

SISTEMA NEURONAL RÁPIDO DE DECISÃO BASEADO EM
CALORIMETRIA DE ALTAS ENERGIAS

André Rabello dos Anjos

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS
PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE
FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS
EM ENGENHARIA ELÉTRICA.

Aprovada por:

Prof. José Manoel de Seixas, D.Sc.

Prof. Luiz Pereira Calôba, Dr.Ing.

Prof. Alexandre Santos de la Vega, D.Sc.

Prof. Márcio Portes de Albuquerque, Dr.Ing.

RIO DE JANEIRO, RJ - BRASIL

ABRIL DE 2001

ANJOS, ANDRÉ RABELLO DOS

Sistema Neuronal Rápido de Decisão
Baseado em Calorimetria De Altas Ener-
gias [Rio de Janeiro] 2001

XVII, 167 p. 29,7 cm (COPPE/UFRJ,
M.Sc., Engenharia Elétrica, 2001)

Tese - Universidade Federal do Rio de
Janeiro, COPPE

1. Física de Altas Energias 2. Calorime-
tria 3. Redes Neurais 4. Processamento
Rápido 5. Processador Digital de Sinais
(DSP)

I.COPPE/UFRJ II.Título (série)

Ao grande amor de minha vida, Aninha:
tu és minha glória. A minhas mães Léa e
Ira e meus eternos irmãos Vanessa e Gus-
tavo: vocês me fortificam.

Agradecimentos:

Agradeço ao apoio de minha noiva, minha mãe e meus irmãos pela paciência eterna. A meu orientador, pela resolução de minhas incansáveis dúvidas e pelas longas horas de discussão. Aos professores e alunos do LPS, pela inestimável contribuição que deram à minha formação e sobretudo a meu caráter pela excelente forma que se relacionam e trabalham.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

SISTEMA NEURONAL RÁPIDO DE DECISÃO BASEADO EM CALORIMETRIA DE ALTAS ENERGIAS

André Rabello dos Anjos

Abril/2001

Orientador: José Manoel de Seixas

Programa: Engenharia Elétrica

Este trabalho desenvolve um classificador neuronal rápido para o problema da discriminação de partículas de alta energia (elétron/jato) no segundo nível de filtragem *on-line* do experimento ATLAS, no CERN, Suíça. O classificador é alimentado pela informação de apenas um dos detetores do ATLAS, o calorímetro, detetor altamente segmentado que mede a energia das partículas com alta resolução. A informação é pré-processada de forma inteligente, formando-se anéis concêntricos de energia, reduzindo-se, assim, significativamente a dimensionalidade da informação original. Apesar da compactação da informação, obtém-se, ainda assim, uma elevada eficiência de discriminação (97% para elétrons e 95,1% para jatos), superando as soluções clássicas atualmente implementadas no sistema de filtragem. Uma implementação do sistema em um processador digital de sinais (DSP) rápido é apresentada, avaliando-se o desempenho do classificador quanto ao tempo de processamento e a qualidade de discriminação.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

FAST NEURAL DECISION SYSTEM BASED ON HIGH ENERGY CALORIMETRY

André Rabello dos Anjos

April/2001

Advisor: José Manoel de Seixas

Department: Electrical Engineering

This work develops a fast neural classifier for high energy particle discrimination (electron/jet) at the second level trigger of ATLAS, at CERN, Switzerland. The classifier is fed by the information from one of the ATLAS detectors, the calorimeter, a highly segmented detector which measures the energy of particles with high resolution. The information is preprocessed in a clever way, by building concentric energy ring sums, which reduces significantly the input dimensionality. Despite the high information compactation rate, the designed system achieves a very high discrimination efficiency (97% for electrons and 95,1% for jets), outperforming the classical solution implemented nowadays at the second level trigger. A system implementation on a fast digital signal processor (DSP) is presented, and its performance is evaluated in both speed and accuracy.

Conteúdo

Conteúdo	vii
Lista de Figuras	xi
Lista de Tabelas	xv
1 Introdução	1
1.1 Contexto	2
1.2 Implementação	3
1.3 Visita guiada	5
2 O ambiente de física de partículas	6
2.1 Técnicas atuais de detecção	6
2.1.1 A aceleração das partículas	8
2.1.2 Detecção: vendo o que ocorreu após a colisão	9
2.1.2.1 Composição dos detetores	10
2.2 A Física de Partículas de hoje	11
2.3 O Laboratório Europeu para a Física de Partículas - CERN	14
2.3.1 O CERN de hoje	15
2.4 O LHC	15
2.4.1 Por que colidir prótons?	17
2.4.2 Decaimentos: as assinaturas do bóson de Higgs	18
2.5 O Experimento ATLAS	19
2.5.1 Calorimetria e o ATLAS	19
2.5.1.1 Calorímetros do ATLAS	22
2.5.2 O Sistema de Filtragem de Eventos do ATLAS	30

2.5.2.1	O primeiro nível de filtragem	31
2.5.2.2	O segundo nível de filtragem (L2)	33
2.5.2.3	O Filtro de Eventos	34
3	O Segundo Nível de Filtragem do ATLAS	36
3.1	Recursos \times tempo de processamento no L2	37
3.2	Organização Lógica do L2	39
3.3	Estruturas e Fluxo de Dados no L2	42
3.4	Os Decisores Locais	43
3.4.1	Decisores Locais para os Calorímetros	44
3.5	Decisão Local com objetos EM	45
3.5.1	Decodificação da posicionamento das células	46
3.5.2	Supressão de valores indesejados	47
3.5.3	Verificação de dados faltantes	48
3.5.4	Discriminação e Extração de Características	49
4	O Software de Referência	51
4.1	O Refsoft e este trabalho	52
4.2	O que é o RefSoft?	52
4.3	Estrutura do RefSoft	53
4.4	O RefSoft e os decisores locais no L2	54
5	Decisores Locais Neurais	61
5.1	O espaço de entrada para os discriminadores de objetos e.m.	62
5.1.0.1	A terceira camada e.m.	66
5.1.1	Os arquivos de entrada	68
5.2	O Processamento clássico de objetos e.m.	70
5.2.1	Eficiência de discriminação	72
5.2.2	Discriminador Linear	73
5.2.3	Discriminador por Análise Combinatória	76
5.2.3.1	Resultados	78
5.2.4	Discriminador Neuronal	79
5.2.4.1	Resultados	80
5.3	Compactação dos dados de entrada usando anéis	83

5.3.1	Normalização	87
5.4	Treinamento Neural	89
5.4.1	Critério de Parada: o produto SP	90
5.4.2	Treinamento	92
5.5	Resultados	93
5.5.1	Resultados para outras normalizações	95
5.6	Informação Relevante	96
5.6.1	Relevância em discriminadores lineares	98
5.6.2	Relevância das quantidades clássicas	100
5.6.3	Relevância dos anéis na separação neuronal	101
5.6.4	Cortes de dimensionalidade baseados na relevância	105
5.6.4.1	As variáveis clássicas	106
5.6.4.2	Eliminando anéis	109
6	Implementação do Decisor Neuronal Local	119
6.1	Pré-processamento de dados	121
6.1.1	Aquisição de Dados	121
6.1.2	Verificação de Integridade da RoI	122
6.1.3	Pré-processamento específico	124
6.1.4	Resultados da implementação	124
6.2	Discriminação neuronal de objetos e.m.	126
6.2.1	Implementação da ativação neuronal	127
6.2.1.1	Acuidade	127
6.2.1.2	Tempo de processamento	129
6.3	Implementação em DSP's	131
6.3.1	O hospedeiro	133
6.3.2	A aplicação no DSP	134
6.3.3	Resultados da implementação	135
7	Conclusões	138
7.1	Somas em anel	139
7.2	Relevância	140
7.3	Implementação	142

7.4 Extensões	143
A Trabalhos publicados	147
B O Sistema de Coordenadas do ATLAS	151
C Orientação a Objetos e as Linguagens C e C++	153
C.1 OO × Estruturação	153
C.2 Relacionamento de objetos	155
C.3 Implementação e C++	156
D O ADSP-21062	159
D.1 Desenvolvimento	160
Referências Bibliográficas	163

Lista de Figuras

2.1	O <i>setup</i> de Rutherford.	7
2.2	A aceleração de partículas.	9
2.3	O diagrama de um detetor cilíndrico moderno.	11
2.4	A interação de partículas com os detetores modernos.	12
2.5	As interações fundamentais da natureza.	13
2.6	Uma visão aérea do CERN, em Genebra na Suíça.	14
2.7	Diagrama esquemático do LEP e seus experimentos.	16
2.8	A simulação da deteção de um bóson de Higgs no CMS.	19
2.9	Diagrama esquemático do detetor ATLAS.	20
2.10	O Calorímetro e.m. do ATLAS em um corte transversal	24
2.11	Diagrama de um dos gomos do barril da seção e.m. do ATLAS.	26
2.12	O calorímetro de telhas do ATLAS (em verde ao centro), em sua posição final, envolvido pelo detetor de múons (em roxo e marrom).	28
2.13	Configuração de leitura das células da seção do barril do Calorímetro de Telhas.	30
2.14	Um esquema do sistema de filtragem do ATLAS.	35
3.1	Ocupação da rede e dos processadores do L2 contra o tempo necessário de processamento para cada evento.	38
3.2	Esquema simplificado de processamento no L2.	41
3.3	Como o posicionamento de cada célula do calorímetro é empacotado em um campo de 25 bits.	47
3.4	Resumo gráfico das operações para a discriminação $e/\gamma \times \text{jato}$	50
4.1	Diagrama esquemático da organização do RefSoft.	54
4.2	Uma análise gráfica da decisão local no L2.	54

4.3	O projeto OO extraído a partir da Figura 4.2. Repare que aqui o supervisor não é modelado. O esquema segue a UML.	56
5.1	O número médio de células por RoI's de diversos objetos e.m..	64
5.2	Uma RoI de um elétron típico interagindo com o calorímetro.	65
5.3	Uma RoI de um jato típico interagindo com o calorímetro.	66
5.4	Uma RoI de um jato cujos padrões de deposição de energia fazem-no parecer um elétron interagindo com o calorímetro.	67
5.5	Variáveis clássicas em planos bi-dimensionais.	72
5.6	Variáveis clássicas em planos tri-dimensionais.	73
5.7	A evolução do EMQ para o discriminador linear	75
5.8	A curva de eficiência para um separador linear ótimo aplicado às 4 variáveis físicas descritas.	76
5.9	O esquema que determina o separador por análise combinatória das 4 variáveis clássicas quando agrupadas em pares.	78
5.10	Comparação entre o discriminador linear e o neural (teste 3).	83
5.11	Evolução do Erro Médio Quadrático durante o treinamento da rede 3.	84
5.12	Idealização da deposição de energia.	85
5.13	A extração de anéis em uma RoI nos calorímetros do ATLAS.	86
5.14	A distribuição da energia contida nos últimos anéis de cada camada do calorímetro (normalizada para área unitária).	87
5.15	Evolução comparativa entre o EMQ e o produto SP em um treinamento neural.	92
5.16	A evolução do EMQ para os conjuntos de treinamento (superior) e teste (inferior) da rede neuronal cuja entrada é o vetor de anéis (teste 20).	95
5.17	Diagrama de um discriminador linear.	99
5.18	Relevância as quatro quantidades clássicas no melhor discriminador neuronal (teste 20).	101
5.19	Relevância das quatro quantidades clássicas no pior discriminador neuronal (teste 18).	102
5.20	Relevância dos 58 anéis utilizando vários tipos de normalização.	103

5.21	A deposição de energia nas 3 camadas hadrônicas, por objetos e.m. que representam elétrons (em cima) e jatos (embaixo).	104
5.22	Relevância para o teste 17 (Tabela 5.5).	105
5.23	Eficiência comparada entre o discriminador linear e os discriminadores neurais.	107
5.24	Eficiência comparada entre os melhores discriminadores neurais usando os 58 anéis, 20 dos 58 anéis e as quatro quantidades clássicas.	112
5.25	As relevâncias para os 20 anéis remanescentes do corte de dimensio- nalidade do espaço de entrada original.	113
5.26	Eficiência comparativa entre os melhores discriminadores neurais quando o número de entradas varia.	116
6.1	Interpolação e decimação bi-dimensional da RoI para ajuste de gra- nularidade.	123
6.2	Organização percentual do tempo durante o pré-processamento de objetos e.m..	125
6.3	A distribuição dos erros quando compara-se a implementação da tan- gente hiperbólica usando a função nativa ao sistema (64 bits) e uma implementação por tabela, de 32 bits, cuja resolução é 0,01 (final em 2,0).	130
6.4	Comparação entre as saídas da rede 58-15-1 quando utiliza-se a imple- mentação da tangente hiperbólica usando a função nativa ao sistema (64 bits) e uma implementação por tabela, de 32 bits, cuja resolução é 0,01 (final em 2,0).	130
6.5	Um fluxograma mostrando a atuação conjunta da aplicação hospedeira com a aplicação no DSP. Este diagrama mostra também os pontos que foram utilizados como referências nas medidas de tempo.	136
6.6	Histograma mostrando a diferença entre a saída da aplicação <i>off-line</i> e o decisor local rodando no DSP.	137
6.7	Tempo de execução do decisor local em um ADSP 21061 da Analog Devices.	137

7.1	Eficiência comparativa entre o melhor discriminador usando 58 anéis e aqueles baseados nas quatro quantidades físicas.	140
B.1	O sistema de coordenadas do ATLAS.	152
C.1	Diagrama do “Ordenador”, uma técnica de projeto OO que permite que algoritmos sejam mudados em um objeto, sem alterá-lo.	156
D.1	Diagrama de blocos do ADSP-21062 SHARC.	161
D.2	Processo de desenvolvimento e projeto em um ADSP-21062.	162

Lista de Tabelas

2.1	A granularidade, camada a camada, dos calorímetros e.m. do ATLAS.	27
2.2	A granularidade, camada a camada, dos calorímetros hadrônicos do ATLAS.	29
3.1	Tipos de algoritmos no L2 e objetos relacionados no L1.	45
3.2	Número de ROBs por detetor	45
5.1	Configurações para um possível discriminador por análise combinatória das variáveis duas a duas (<i>clássico</i>).	77
5.2	Os resultados do treinamento de 23 redes neurais para separação elétron/jato utilizando as quatro variáveis clássicas.	82
5.3	O número de anéis por camada do calorímetro.	86
5.4	Normalização sequencial.	90
5.5	Os resultados do treinamento de 20 redes neurais para separação elétron/jato utilizando anéis.	94
5.6	Os resultados do treinamento de algumas redes neurais usando os 58-anéis.	97
5.7	Resultados obtidos com algumas redes neuronais cujo vetor de entrada são as componentes clássicas de discriminação.	108
5.8	As variáveis do espaço original de anéis de dimensão 58, após o corte baseado na relevância.	110
5.9	Resultados de alguns discriminadores neuronais utilizando o espaço reduzido de anéis (dimensão 20).	111
5.10	As variáveis do espaço original de anéis de dimensão 58, após o corte baseado na relevância aplicado ao teste 6 (Tabela 5.9).	113

5.11	Resultados obtidos com cortes de dimensionalidade aplicados aos 20 anéis restantes utilizando a técnica de cortes sequenciais.	115
5.12	Resultados obtidos com diversos cortes de dimensionalidade aplicados aos 58 anéis utilizando a técnica de corte direto.	117
5.13	As variáveis do espaço original de anéis de dimensão 58, após vários cortes baseados na relevância.	118
6.1	Erro Médio Quadrático por tipo de variável/ativação. Estes testes utilizam uma rede neuronal 58-5-1.	128
6.2	Tempo de processamento médio para cada RoI usando uma rede 58-5-1.	132

Lista de Implementações

4.1	O Coletor de Dados (<code>Data Collector</code>)	57
4.2	Os algoritmos de discriminação e extração de características.	58
4.3	O decisor local (<code>CaloEMLocalProcessor</code>).	59
5.1	Os campos de dados dos calorímetros.	69
C.1	As classes <code>Dictionary</code> e <code>Organizer</code>	158

Capítulo 1

Introdução

Sistemas de filtragem são necessários em ambientes onde se deseja separar o sinal de interesse de uma grande quantidade de outros sinais ordinários. Geralmente, existem restrições no tempo de execução para este tipo de processamento, pois os sistemas de filtragem operam, freqüentemente, em ambientes que geram altas taxas de eventos. Eventualmente, os eventos gerados podem requerer uma quantidade de memória elevada, exigindo uma discriminação *on-line* de alta velocidade e tornando todo o processo de filtragem ainda mais complexo.

As soluções procuradas nestes sistemas devem, via-de-regra, incluir:

- A utilização de métodos de discriminação muito eficientes;
- A simplicidade na implementação (mapeamento em *hardware* ou *software*);
- Flexibilidade de programação, alteração do código e parâmetros de funcionamento;
- Velocidade compatível com o sistema-alvo;
- Robustez, no caso do sistema funcionar por longo período;
- Integrabilidade, no caso do filtro funcionar em conjunto com outros sistemas de filtragem.

1.1 Contexto

Todos os requisitos acima são tipicamente encontrados na filtragem de eventos para experimentos com colisionadores de partículas de altas energias. Em especial, no ATLAS (*A Toroidal LHC ApparatuS*), um laboratório de detecção para o experimento colisionador de última geração, LHC (*Large Hadron Collider*), a se tornar funcional no CERN, Suíça, em 2005, o sistema de filtragem deverá atender a taxas de eventos jamais atingidas. Cerca de 40 milhões de eventos serão gerados por segundo, e cada um destes possuirá cerca de 1 *megabyte* de dados, produzindo-se, portanto, 40 *gigabytes* de dados por segundo. Esta taxa não pode ser gravada em mídia permanente e, mesmo que pudesse, não seria viável sua análise *off-line*, pois o sinal de interesse para a experiência estaria imerso numa densa quantidade de eventos que formam o ruído de fundo da experiência. Sendo assim, é necessário projetar-se um sistema de filtragem eficiente e suficientemente rápido, que consiga reduzir a taxa de eventos do ATLAS para valores mais realísticos.

A física de interesse no experimento LHC é o bóson de Higgs. Esta partícula poderá ser observada no ATLAS umas poucas vezes ao longo de vários dias, nas suas condições de operação. O Higgs (se existir), além de raro, é extremamente instável, decaindo em partículas mais estáveis e menos energéticas durante sua interação com o laboratório ATLAS. O sistema de filtragem deverá identificar os padrões de decaimento do Higgs (sinal) para poder localizá-lo na massa de eventos com física ordinária, produzida pelas interações mal-sucedidas (interações que não produzem o Higgs, mas sim canais físicos já conhecidos e que, portanto, significam ruído) durante o experimento.

O sistema de filtragem do experimento ATLAS é composto de 3 níveis conectados em cascata, com complexidade e tempo de processamento crescentes. O primeiro nível será basicamente implementado em *hardware* de baixa programabilidade, reduzindo a taxa de eventos de entrada para não mais que 100 mil por segundo (100 kHz). O terceiro nível será implementado com processores comuns (PC's) interligados por redes rápidas.

O segundo nível de filtragem (L2), em especial, deve reduzir a taxa de eventos dos 100.000 por segundo, que lhe será entregue pelo primeiro nível, para não mais que 1.000 eventos por segundo. Para isto, contará com uma rede de computadores

que processarão algoritmos de busca especializados nos diversos sub-detetores do ATLAS. Esta busca visa encontrar elementos que representem decaimentos do bóson de Higgs.

Dentre os algoritmos especializados, encontram-se os algoritmos separadores de elétrons e jatos, que se baseiam em dados dos *calorímetros* do ATLAS. Os calorímetros são equipamentos responsáveis por medir a energia de partículas, sendo capazes de produzir uma resposta rápida e de permitir uma discriminação eficiente das partículas que com ele interagem. Elétrons podem representar a física de interesse do experimento e estão contidos em cerca de 60% das prováveis *assinaturas* de Higgs.

1.2 Implementação

Neste trabalho, implementa-se um sistema de discriminação neuronal elétron-jato para o segundo nível de filtragem do ATLAS, basendo-se nos dados de seus calorímetros. Os dados de cada objeto (elétron ou jato) correspondem a uma pequena parte do detetor completo, aquela efetivamente sensibilizada pela interação física dos sub-produtos gerados na colisão. Isto representa uma matriz tri-dimensional de células. Cada camada da matriz possui uma granularidade característica. O número de células total, por objeto a ser discriminado estará, em torno de 1000.

Para reduzir a elevada dimensão dos sinais de entrada, inspirou-se no conhecimento da física do experimento e aplicou-se um processamento em soma de anéis concêntricos de energia, construídos ao redor dos picos de deposição energética nas camadas, resultando em um total de 58 quantidades que possam descrever o fenômeno.

O vetor de entrada, agora com 58 componentes, alimenta uma rede neuronal totalmente conectada, com apenas 1 neurônio na saída, que realiza a discriminação entre elétrons e jatos. A rede foi treinada com diferentes parâmetros, visando encontrar a melhor configuração de discriminação. Para o critério de parada, ao invés de utilizar-se o Erro Médio Quadrático (EMQ), utilizou-se uma quantidade que se baseia na eficiência de discriminação elétron/jato, já que a ênfase para o processamento neuronal é no reconhecimento de padrões.

Após vários testes, confirmou-se que uma rede com 5 neurônios na camada intermediária poderá efetuar a separação elétron/jato com 97% de eficiência na discriminação de elétrons, para menos de 5% na probabilidade de falso alarme, a qual corresponde à probabilidade com que jatos venham a ser erroneamente classificados como elétrons.

Para tentar reduzir ainda mais a dimensão de entrada da rede classificadora, apresenta-se um estudo da relevância de cada entrada do discriminador neuronal. Para fazê-lo, substituiu-se o valor de cada variável de entrada (um dos 58 anéis formados) pelo seu valor médio, avaliando o impacto dessa substituição na resposta do classificador e, portanto, o significado de cada componente do vetor de entrada para o processo de discriminação.

Com base nesta análise, variáveis cujas relevâncias venham a ser menores que um dado patamar podem ser eliminadas da entrada. A partir dos sub-conjuntos das variáveis de entrada, cada um tomado em relação a um patamar de eliminação das variáveis menos relevantes, novos discriminadores neurais mais compactos com 20 ou até 5 entradas foram propostos, conseguindo-se manter um desempenho superior, em relação ao método clássico desenvolvido pela comunidade no CERN. Um discriminador com 5 dos 58 anéis iniciais, e tendo apenas 4 neurônios na camada intermediária, consegue cerca de 95% de eficiência na classificação de elétrons, contra apenas 5% de falso alarme.

Na fase atual de desenvolvimento do sistema de filtragem do ATLAS, o discriminador operará dentro de um ambiente de emulação, desenvolvido com nossa participação, de forma que este sistema seja independente da plataforma na qual é executado (*Software* de Referência). Este ambiente é implementado usando-se o paradigma da Orientação a Objetos (OO) e provê várias vantagens ao sistema de filtragem, embora determine também algumas restrições de funcionamento. O sistema que foi desenvolvido, embora não totalmente conforme, implementa uma parte significativa das restrições de operação propostas pelo *Software* de Referência e utiliza também algumas de suas vantagens, como será visto.

Por fim, o método de discriminação neuronal baseado em anéis foi implementado na tecnologia dos processadores digitais de sinais (DSP's). Testes de desempenho mostram que todo o processamento exigido poderá ser executado em aproxi-

madamente 1 ms, mantendo a acurácia necessária e cumprindo todos os requisitos essenciais do sistema de filtragem do ATLAS.

1.3 Visita guiada

O trabalho está dividido em outros 6 capítulos. No Capítulo 2, descreve-se o detetor ATLAS em detalhes, assim como conceitos de calorimetria e do sistema de filtragem de eventos são fornecidos, em linhas gerais. No Capítulo 3, detalha-se o segundo nível de filtragem do ATLAS, berço dos discriminadores elétron-jato desenvolvidos, explicando a importância na detecção de elétrons e as limitações técnicas do problema.

O Capítulo 4 introduz o *Software* de Referência, um sistema complexo de emulação do L2. Nesse capítulo, mostra-se ao leitor os detalhes de implementação do sistema de filtragem e justificam-se as escolhas com relação ao tipo de processamento e a abordagem orientada a objetos. O Capítulo 5 aborda o problema da discriminação neuronal na classificação elétron/jato, avaliando as diferentes configurações e enfoques de treinamento da rede. Esse capítulo também detalha o pré-processamento dos dados de entrada em anéis concêntricos ao pico de energia e o estudo de relevância das entradas, como forma de redução adicional da dimensionalidade do sinal de entrada.

No Capítulo 6, uma visão mais detalhada da implementação de um decisor local é dada, utilizando as configurações discutidas nos capítulos anteriores. Nesse capítulo, discute-se também a implementação do decisor local em um processador digital de sinais (DSP). Medidas de tempo e acuidade são discutidas. O Capítulo 7 conclui este documento, resumizando os principais pontos do trabalho e resultados conseguidos. No final desse capítulo, possíveis extensões deste trabalho são discutidas.

Capítulo 2

O ambiente de física de partículas

A física de partículas estuda a composição da matéria a nossa volta, a nível atômico. Tudo que é possível tocar ou sentir é essencialmente, como nos é ensinado, composto por átomos. Átomo (do grego, *sem divisão*) era considerado a partícula fundamental até que no final do século XIX e princípio do século XX, cientistas como J.J. Thompson e E. Rutherford constataram que, na realidade, átomos são composições de outras partículas eletricamente carregadas [1]. Estas partículas estão organizadas de forma a neutralizar a carga elétrica do átomo, e por isto não são perceptíveis aos métodos triviais de detecção de carga elétrica. Os átomos são compostos por um núcleo maciço, constituído de nêutrons e prótons, e cercados por elétrons, que giram ao redor deste núcleo.

Por estarem envoltos por uma barreira de potencial (causada pelo movimento dos elétrons), os núcleons (isto é, prótons e nêutrons) não podiam ser observados naturalmente. Para extraí-los seria necessário que houvesse energia suficiente para que pudessem vencer a barreira de potencial¹.

2.1 Técnicas atuais de detecção

Rutherford utilizou-se de um feixe de partículas alfa, emitido por uma fonte radioativa, uma folha de ouro e um simples detetor feito de Sulfeto de Zinco para

¹Ou que ocorresse um tunelamento, para que algumas partículas nucleares pudessem escapar, ou seja, uma redução da barreira de potencial que houvesse em algum ponto da periferia do núcleo, permitiria que núcleons pudessem escapar.

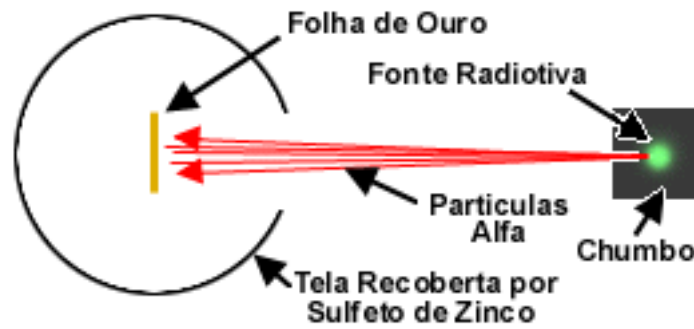


Figura 2.1: A configuração do experimento de Rutherford para a constatação de que o núcleo atômico era denso e positivo.

testar sua teoria [1]. As partículas alfa, ao baterem no detetor, marcavam-no (veja a configuração do experimento de Rutherford na Figura 2.1). Embora não pudesse ver o que ocorria no mundo sub-atômico, Rutherford podia teorizar, testar sua hipótese e então analisar os dados experimentais para verificar se sua hipótese estava correta. A única hipótese cabível, após seus experimentos, é que o átomo tem que ser composto de um núcleo compacto e positivo e uma periferia negativamente carregada, de partículas mais leves e menores que aquelas no núcleo.

De forma parecida a Rutherford, os físicos atuais usam um feixe de partículas aceleradas. Tal feixe pode colidir com um alvo fixo ou com um outro feixe de partículas, que é acelerado em direção contrária ao feixe primário. Para visualizar eletronicamente os sub-produtos de tais interações físicas, utilizam-se múltiplos detetores.

A aceleração das partículas resolve dois problemas que os físicos de hoje encontram em seus experimentos:

1. Comprimento de Onda - O comprimento de onda determina a acurácia do que é possível observar [2].

Uma vez que partículas também apresentam características de onda, não é possível obter uma medida acurada usando-se partículas comuns, como um elétron, na observação de partículas muito pequenas. Um elétron não serve, nem mesmo, para observar outro elétron. A aceleração da partícula, no entanto, aumenta seu momento, diminuindo² seu comprimento de onda e permitindo

²O comprimento de onda e o momento de um corpo são inversamente proporcionais.

que medidas acuradas possam ser tomadas usando-se partículas maiores, como elétrons.

2. Energia Cinética - Deseja-se, nos experimentos modernos, que o impacto seja o mais aniquilador possível. Isto é interessante, pois ao se aniquilar matéria, liberando muita energia, partículas mais massivas e menos estáveis são geradas. Ao se acelerar uma partícula, aumenta-se sua energia cinética, tornando a colisão com o alvo mais eficiente (melhor aniquilação).

2.1.1 A aceleração das partículas

A aceleração de partículas é um processo bastante simples: inicialmente deve-se escolher partículas eletricamente carregadas para um experimento - elétrons, pósitrons, prótons, anti-prótons ou íons, num geral, são aconselháveis. As partículas eletricamente carregadas são posicionadas no interior de um túnel e excitadas a acelerar por pulsos eletromagnéticos. A Figura 2.2 exemplifica como elétrons são acelerados por pulsos eletromagnéticos (e.m.): as ondas e.m. aceleram as partículas, pois elementos eletricamente carregados adquirem força (aceleração) quando envolvidos por um campo eletromagnético.

Os aceleradores podem ter dois formatos: linear e circular. Num acelerador linear, as partículas são injetadas em uma extremidade e percorrem uma reta até que colidam com outras partículas ou com um alvo fixo. A outra possibilidade é ter um acelerador circular ou síncrotron. Num acelerador circular, as partículas são injetadas em um ponto do anel de aceleração e lá permanecem até que tenham adquirido velocidade suficiente ao experimento a que se destinam. A aceleração circular exige que magnetos poderosos curvem a trajetória das partículas injetadas. Aceleradores circulares também permitem que vários experimentos sejam conduzidos em pontos de sua circunferência, simultaneamente. Aceleradores circulares possuem uma desvantagem: quanto maior a energia necessária no centro de impacto, maior deve ser o perímetro de sua circunferência.

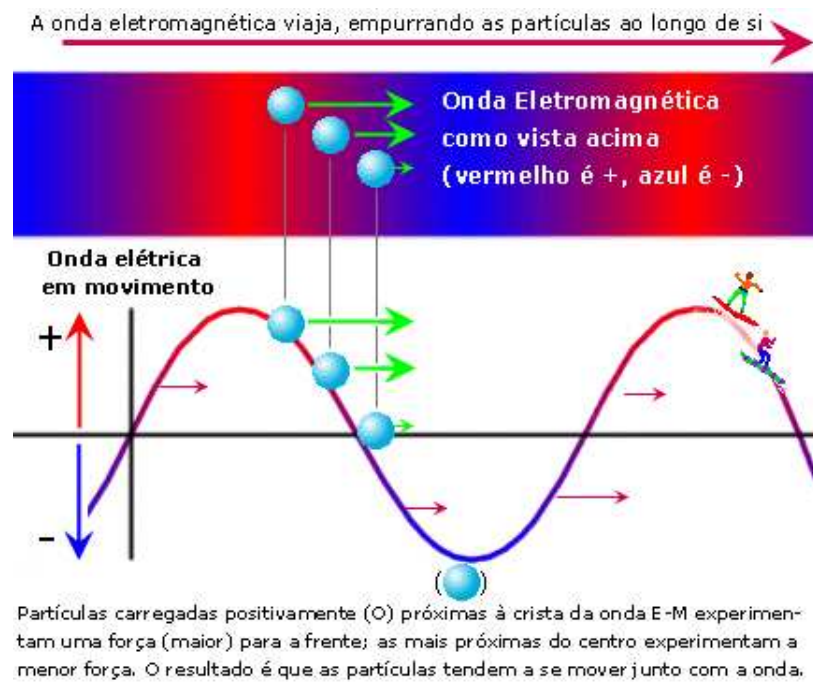


Figura 2.2: A aceleração de partículas eletricamente carregadas usando-se de pulsos e.m..

2.1.2 Detecção: vendo o que ocorreu após a colisão

Depois de um acelerador ter “bombeado” energia suficiente para suas partículas, elas colidem com um alvo fixo ou, então, com as partículas de outro feixe acelerado. Cada uma dessas colisões forma um *evento* físico. O objetivo dos físicos é isolar cada evento, coletar dados a seu respeito e verificar se o processo do qual a partícula participou está de acordo com a teoria que está sendo testada no experimento em questão.

A análise de cada evento é muito complexa, porque muitas partículas são produzidas. A maioria dessas partículas (ou “objetos”) têm tempo de vida tão diminuto que viajam por distâncias extremamente curtas, antes de decaírem em outras partículas, sem deixar pistas detetáveis.

Para procurar esses vários objetos e os produtos de seu decaimento, os físicos projetam detetores com multi-componentes, que testam diferentes aspectos de um evento. Cada componente de um detetor moderno é usado para medir vários parâmetros das partículas provenientes de um evento, e/ou distinguir os diferentes tipos de objetos gerados. Quando todos esses componentes funcionam juntos

para detetar um evento, partículas individuais podem ser distinguidas da multidão para efeito de análise.

Seguindo cada evento, os sistemas de processamento coletam e interpretam a vasta quantidade de dados dos detetores e apresentam os resultados extrapolados aos físicos.

Os físicos interessam-se pelos eventos que ocorrem durante ou mesmo depois da colisão das partículas. Por essa razão, colocam detetores em regiões nas quais os objetos resultantes daquela interação passarão. Os detetores são construídos de diferentes maneiras, dependendo do tipo de colisão analisada:

Alvo fixo Num experimento envolvendo um alvo fixo, as partículas produzidas geralmente projetam-se para frente; por isso, os detetores são na forma de cones e são colocados ao longo da direção do feixe;

Feixes de colisão Durante um experimento envolvendo feixes em colisão, as partículas são espalhadas em todas as direções; assim, o detetor mais adequado é esférico ou, mais comumente, cilíndrico.

2.1.2.1 Composição dos detetores

Os detetores modernos são feitos de peças distintas, que testam diferentes aspectos de um evento. Esses vários componentes são organizados de tal maneira que os físicos possam obter o máximo de informação sobre as partículas geradas durante um evento. A Figura 2.3 mostra um diagrama esquemático de um detetor cilíndrico moderno. Uma figura humana é mostrada em escala, indicando a enormidade dos detetores típicos de altas energias.

Na Figura 2.3, é possível observar que um detetor moderno é composto de 4 sub-detetores principais. O primeiro detetor, de dentro para fora (tomando o feixe de partículas como origem), tem a função de determinar a rota e curvatura de partículas, por isto é chamado de detetor de trajetória ou traços. A curvatura das partículas sobre o campo magnético a que são expostas no detetor dá informações sobre a carga da partícula e seu momento. O segundo detetor é chamado calorímetro eletromagnético (e.m.) e tem a função de determinar a energia total de elétrons, pósitrons e fótons (raios γ). Estas partículas decaem, formando chuviscos de outros

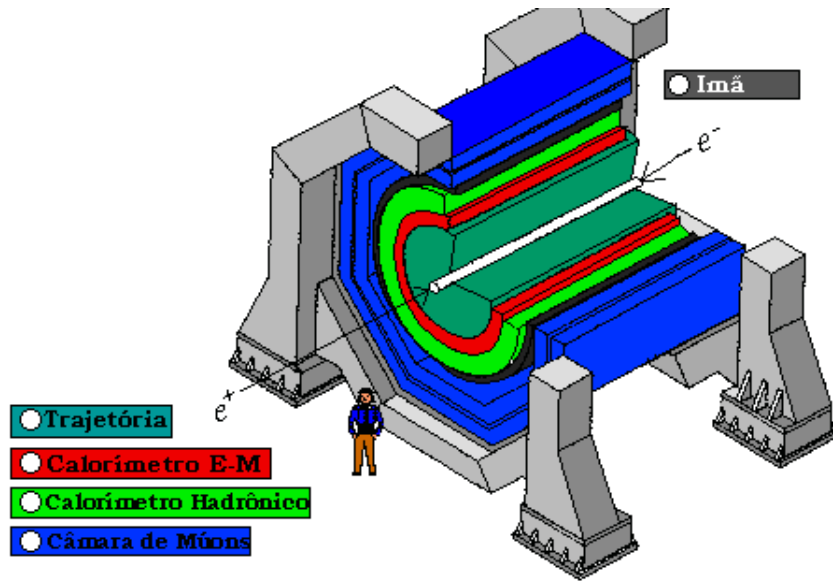


Figura 2.3: O diagrama de um detector cilíndrico moderno.

elétrons, pósitrons e fótons por dentro do calorímetro e.m.. O terceiro detector, o calorímetro hadrônico, mede a energia total de chuveiros originados por hádrons (prótons, nêutrons ou mésons). É composto de material pesado, que força o decaimento dos hádrons num chuveiro hadrônico, bem mais largo e profundo que o eletromagnético. O quarto detector é um detector de múons. Somente estas partículas e neutrinos escapam dos outros sistemas de detecção. Neutrinos, infelizmente, nem por este último são detetados. Sua massa pode ser calculada, no entanto, através da energia faltante no evento. A Figura 2.4 mostra como alguns tipos de partículas interagem com estes detectores. Fótons, por exemplo, têm carga nula e não são detetados pelos detectores de traços, mas interagem com os calorímetros e.m.. Elétrons e pósitrons interagem tanto com os detectores de traços quanto com o calorímetro e.m., desenvolvendo cascata e.m. neste último. Prótons são detetáveis pelos detectores de traços, mas desenvolvem cascata somente nos calorímetros hadrônicos. Nêutrons somente desenvolvem cascata nos detectores hadrônicos.

2.2 A Física de Partículas de hoje

Por volta da década de 30, com o advento de novos detectores e colisionadores mais eficientes, para a surpresa de muitos físicos, dezenas de outras partículas foram

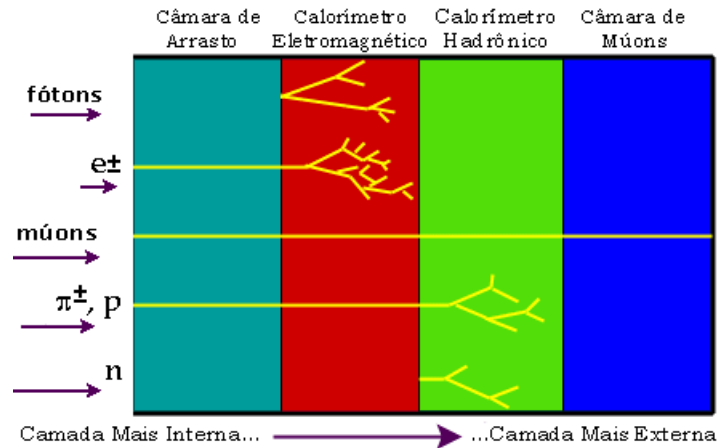


Figura 2.4: A interação de partículas com os detectores modernos.

descobertas, dentre elas píons e káons. Estas partículas possuem uma vida média muito pequena, o que tornava impossível sua detecção natural.


Na época, decidiu-se por organizar as partículas em 3 grupos quanto a sua massa: bárions (partículas pesadas), mésons e léptons (partículas mais leves). Neste ponto, ficou difícil entender como o núcleo atômico não explode, pois a força eletromagnética de seus prótons os impeliria a tal. Finalmente, em 1964, M. Gell-Mann e G. Zweig propuseram uma teoria para entender a composição dos núcleons e dos mésons recém-descobertos, baseando-se na existência de outra partícula chamada *quark*, e quotando Gell-Mann:

“Não há méson ou bárion conhecido cujas propriedades não possam ser entendidas em termos de uma combinação apropriada de quarks. Inversamente, não há combinação possível de quarks sem uma correspondência com um méson ou um bárion observado.”

Os bárions e mésons seriam então formados por 3 ou 2 quarks, respectivamente, enquanto léptons como os múons, elétrons e neutrinos seriam partículas fundamentais. Os quarks, e somente eles, interagem através de uma força muito poderosa de coesão chamada de *força forte*. Foram previstos e encontrados em experimentos no passado 6 tipos ou *sabores* de quarks: up, down, top, bottom, charm e strange.

As partículas fundamentais, i.e., quarks e léptons, e num geral **tudo** a nossa volta [2] se relaciona através das interações (ou forças) fundamentais: eletromagnética,

THE FORCES IN NATURE			
TYPE	INTENSITY OF FORCES (DECREASING ORDER)	BINDING PARTICLE (FIELD QUANTUM)	OCCURS IN
STRONG NUCLEAR FORCE	~ 1	GLUONS (NO MASS)	ATOMIC NUCLEUS
ELECTRO-MAGNETIC FORCE	$\sim \frac{1}{1000}$	PHOTON (NO MASS)	ATOMIC SHELL ELECTROTECHNIQUE
WEAK NUCLEAR FORCE	$\sim \frac{1}{100000}$	BOSONS Z^0, W^+, W^- (HEAVY)	RADIOACTIVE BETA DESINTEGRATION
GRAVITATION	$\sim 10^{-38}$	GRAVITON ?	HEAVENLY BODIES



THE EXCHANGE OF PARTICLES IS RESPONSIBLE FOR THE FORCES

Figura 2.5: As interações fundamentais presentes na natureza e suas respectivas partículas. Em seguida, uma ilustração de como a troca de partículas pode transmitir forças.

forte, fraca e gravitacional. Cada tipo de força interage com elementos específicos. Por exemplo, a força forte é aquela que faz com que núcleons, ou seja prótons e nêutrons, permaneçam coesos no núcleo atômico, sem “explodir”. A força eletromagnética, sendo este um segundo exemplo, dá origem à interação eletromagnética entre prótons e elétrons no átomo.

Uma *interação*, no entanto, não é um ente presente, mas uma teorização de como dois corpos se relacionam. Para explicar concretamente como as partículas interagem, deve-se entender que existem partículas transportadoras de força. A Figura 2.5 mostra uma tabela de todas as partículas transportadoras de força previstas e as forças com as quais estão associadas. Abaixo da tabela, ilustra-se como os corpos interagem trocando partículas de força através de um exemplo didático. Imagine duas pessoas em embarcações distintas, segurando uma bola de basquete. Ao lançar a bola de basquete à outra pessoa, a primeira se distanciará do ponto inicial e idem para a segunda ao receber a bola. Ao enviar uma partícula para um outro corpo, estes se separam. Mecanismo equivalente ocorre quando duas partículas se atraem.

É interessante notar, no entanto, que embora existam todos estes tipos de partículas, somente algumas formam o mundo que conhecido: elétrons, o neutrino de elétrons e os quarks *up* e *down*. Isto se deve ao fato que partículas mais pesadas



Figura 2.6: Uma visão aérea do CERN, em Genebra na Suíça.

que estas decaem rapidamente para formar partículas de menor massa. Estes decaimentos, em geral, implicam em uma partícula *transformar-se* em outra, liberando partículas transportadoras de força.

2.3 O Laboratório Europeu para a Física de Partículas - CERN

O Laboratório Europeu para a Física de Partículas (CERN, Genebra, Suíça) vem colaborando, direta e indiretamente, desde a década de 40, nos avanços nas áreas de física de partículas com altas energias. Ele pertence a um conjunto de países europeus e por eles é financiado.

No CERN, dezenas de experimentos com alvos fixos e colisionadores são realizados todos os anos [3]. Graças a sua existência, os físicos teóricos encontram um campo fértil de idéias e os experimentais, recursos para testar teorias de forma bastante eficiente. A Figura 2.6 mostra uma foto aérea do CERN.

2.3.1 O CERN de hoje

O CERN, atualmente, não é apenas um laboratório europeu, mas sim um centro de estudos internacional. Embora não façam parte dos países integrantes do CERN, países não-europeus podem contribuir financeiramente e intelectualmente para alguns dos experimentos do CERN. Milhares de cientistas em todo o mundo estão reunidos a sua volta, hoje em dia, pesquisando em muitas áreas de física de partículas e áreas associadas, como engenharia eletrônica, computação e materiais, entre outras.

Os experimentos mais conhecidos do CERN, no entanto, são aqueles que estão fazendo averiguações com altas energias. Para isto, o CERN conta com o LEP (*Large Electron-Positron Collider*). O LEP é um colisionador elétron-pósitron que visa, entre outros aspectos, depurar os valores de massa de quarks.

O LEP possui 26,7 km de perímetro e encontra-se sob a fronteira franco-suíça. Nele, situam-se 4 pontos de colisão que abrigam 4 experimentos distintos: ALEPH, L3, DELPHI e OPAL [4]. O diagrama da Figura 2.7 mostra como o LEP está distribuído sobre a fronteira entre a França e a Suíça e onde os 4 experimentos estão situados. O anel é colocado a cerca de 100 metros de profundidade para evitar vazamentos radio-ativos para a superfície e também para diminuir o ruído no experimento. A sensibilidade do feixe é tanta que seu direcionamento é levemente alterado com a maré do lago Lemano, na Suíça.

2.4 O LHC

O LHC (*Large Hadron Collider*) é a máquina colisionadora de última geração que utilizará o mesmo túnel construído para o LEP, de tal forma que hádrons (na verdade prótons), possam ser colididos. O LHC terá, como o LEP, 4 pontos com diferentes experimentos: ATLAS, CMS, ALICE e LHCb [5]. Está previsto que o término de construção do LHC será em 2005. É possível citar como características desse colisionador:

Pacotes Os feixes de partículas não serão contínuos no LHC, e sim divididos em pacotes, ou *bunches*. Isto dá tempo para que o sistema de detecção se reajuste

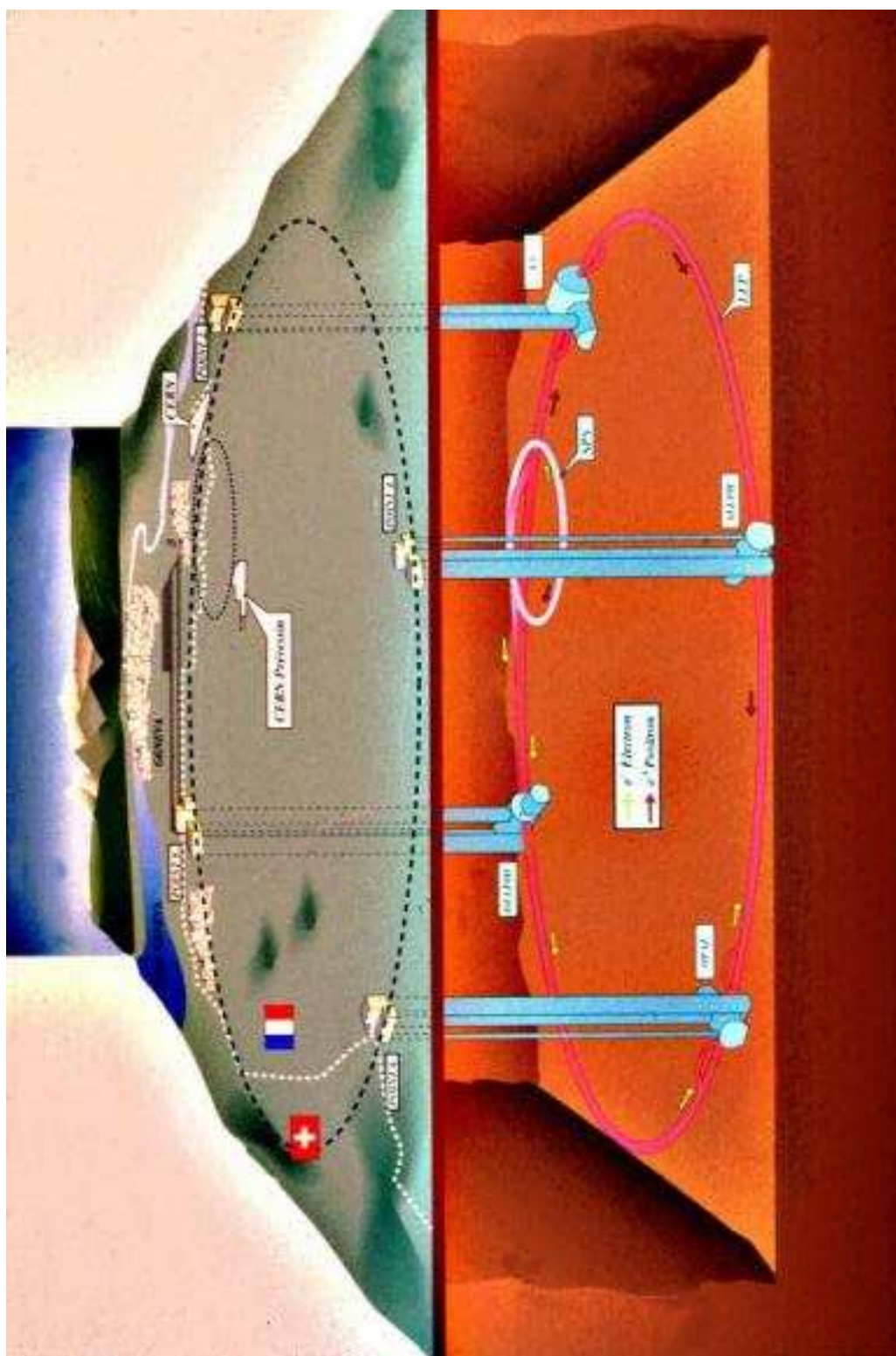


Figura 2.7: Um diagrama esquemático do LEP e seus 4 experimentos. O LEP fica a cerca de 100 metros abaixo da superfície.

para novas colisões, evitando o acúmulo de informações nos detetores ao longo do tempo. Cada pacote terá cerca de 10^{11} prótons.

Luminosidade Ao colocar pacotes de partículas em rota de colisão há uma certa probabilidade de algumas das partículas colidirem. Colimando-se o feixe de partículas esta chance aumenta. Por outro lado, se diminui-se o tempo entre cada pacote, esta chance também aumentará. E ainda, se aumenta-se o número de partículas por pacote, esta chance também aumentará. Por esta razão, é comum medir o tempo de cada pacote por área de sua seção ortogonal e concentração. Esta medida é chamada luminosidade do feixe, ela é proporcional ao número de partículas em cada pacote (N), a área da seção ortogonal do feixe (S) e o tempo (t) entre cada pacote de partículas no colisionador. A quantidade é expressa em $cm^{-2}s^{-1}$.

$$\text{Luminosidade} \propto \frac{N}{S \times t} cm^{-2}s^{-1} \quad (2.1)$$

Assim, quanto maior a luminosidade menor a área da seção ortogonal ao feixe e/ou menor o período para o qual pacotes chegarão ao centro do detetor e maior o número de partículas por pacote. Quanto maior a luminosidade, maior a probabilidade de partículas colidirem. No LHC, a luminosidade do feixe será de $10^{34} cm^{-2}s^{-1}$; ou seja, pretende-se atingir a maior luminosidade dentre todos os colisionadores. O número de eventos esperado para esta luminosidade será de, aproximadamente, 40 milhões por segundo!

Energia no centro de impacto Como colocado na Seção 2.1, a energia cinética no centro do impacto é fundamental na formação de elementos muito massivos. A energia cinética, em experimentos de física de partículas é medida em elétron-Volts. No LHC, a energia no centro de impacto será de 14 TeV.

2.4.1 Por que colidir prótons?

No LHC deseja-se verificar a existência dos bósons de Higgs. Os bósons são partículas cujo o momento angular é um múltiplo inteiro na mecânica quântica [2]. O

bóson de Higgs, segundo teóricos, é responsável por interagir com algumas partículas, gerando massa. Isto se dá de forma parecida à interação de partículas eletricamente carregadas com campos e.m., gerando força. A descoberta desta partícula promete ser um dos próximos prêmios Nobel na área de física.

Caso exista, o bóson de Higgs encontra-se em energias de 100 MeV a 1 TeV, sendo, portanto, muito massivo. O bóson de Higgs, segundo previsões, é uma partícula muito instável, e portanto, decairá rapidamente em outras partículas mais estáveis e partículas portadoras de força. Para detetá-lo é preciso conhecer **todas** as possíveis interações próton-próton, definindo a existência do bóson por eliminação de possibilidades, como fez Rutherford em seu primeiro experimento sobre os núcleos atômicos.

Por possuírem 3 quarks (2 quarks *up* e 1 quark *down*), prótons são elementos muito massivos. Ao serem acelerados a velocidades próximas a da luz e colocados em rota de colisão, há uma possibilidade em aproximadamente 1 milhão de que um bóson de Higgs seja gerado. Havendo cerca de 40 milhões de interações por segundo, é esperado que de 10 a 100 iterações produzam a física de interesse do experimento.

2.4.2 Decaimentos: as assinaturas do bóson de Higgs

Após um evento gerar um bóson de Higgs, é esperado que ele decaia em partículas menos massivas, liberando partículas transmissoras de força ao redor do centro de impacto. É possível entender os decaimentos de bósons de Higgs como assinaturas que identificam a existência da partícula. Numa assinatura, três fatores são de extrema importância: o tipo de partícula, sua carga e sua energia.

Dentre as assinaturas possíveis, dado o número de eventos que irão compor o sinal de fundo (aproximadamente 40 milhões), os físicos têm que escolher as mais fáceis de separar com algoritmos rápidos, já que não poderão gravar ou analisar **todos** os eventos gerados. Por isto, físicos selecionam assinaturas que geram léptons como elétrons e múons, pois a detecção destes elementos é bastante eficiente com a tecnologia atual de detetores. A Figura 2.8 mostra a simulação da detecção de um Higgs em um dos experimentos do LHC, o CMS (*Compact Muon Solenoid*) [6], indicando a complexidade da tarefa de detecção.

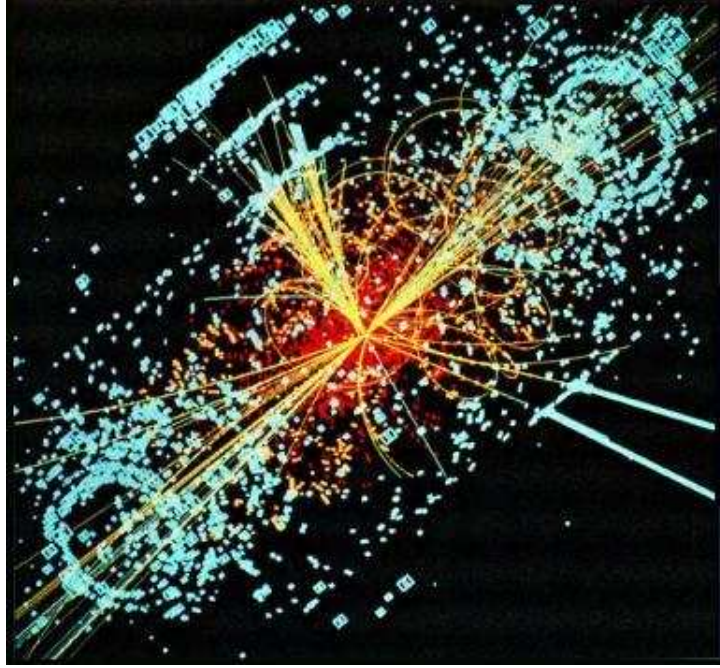


Figura 2.8: A simulação da detecção de um bóson de Higgs no CMS.

2.5 O Experimento ATLAS

O experimento ATLAS (do inglês *A Toroidal LHC ApparatuS*) deseja investigar a existência do bóson de Higgs e determinar com boa acurácia sua massa, caso exista. O ATLAS está sendo projetado no CERN.

O ATLAS pretende investigar as interações próton-próton e detetar o bóson de Higgs. Para isto, seu detetor será formado por 4 sistemas de detecção independentes, posicionados cilindricamente ao redor da colisão. No interior, 2 detetores de traço serão colocados. A seguir, virão os calorímetros e.m. e hadrônico e por fim os detetores de múons. A Figura 2.9 mostra esquema do detetor ATLAS. No canto inferior esquerdo, é possível notar o desenho de pessoas em escala. O ATLAS terá altura equivalente a um prédio de 3 andares.

2.5.1 Calorimetria e o ATLAS

Para medir a energia de partículas, usam-se calorímetros. O princípio de detecção é simples [7]. Basicamente, um calorímetro é um bloco de matéria com

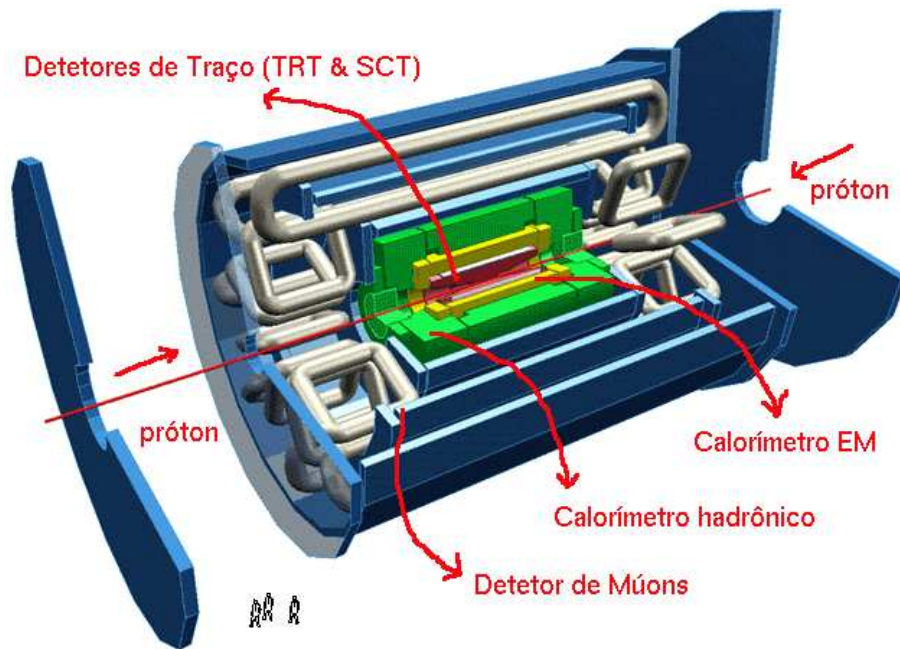


Figura 2.9: Diagrama esquemático do detector ATLAS.

o qual a partícula que será medida interage³ e transforma (parte de) sua energia numa quantidade mensurável. É muito importante, no entanto, que o sinal gerado seja proporcional à energia que deseja-se medir, o que em muitos casos é difícil de atingir.

Dentre as razões pelas quais os calorímetros emergiram como detetores-chave em praticamente todos os experimentos em física de partículas estão :

1. Calorímetros são sensíveis a partículas neutras e carregadas;
2. Devido a diferenças na forma de deposição de energia das partículas, a identificação de partículas com alta eficiência pode ser atingida usando-se calorímetros;
3. Quanto maior a energia da partícula, mais acurado é o resultado. Isto não acontece com outros tipos de detetores;
4. Para conter o desenvolvimento de cascatas dos objetos a serem medidos, a profundidade dos calorímetros aumenta logarithmicamente com a energia, o que permite o projeto de detetores mais compactos;

³Neste caso, interagir significa decair em partículas mais leves e estáveis, formando uma cascata ou chuva.

5. Não precisam de campos magnéticos (como os detetores de traços);
6. Podem ser segmentados, o que permite acurada medida da energia e a visualização da trajetória das partículas;
7. Resposta rápida (melhor que 50 ns) pode ser atingida, o que é importante num ambiente com alta taxa de eventos;
8. A informação de energia pode ser usada para filtrar eventos interessantes com alta seletividade;

Tradicionalmente, muitos experimentos empregam dois tipos de calorímetros: eletromagnético (e.m.) e hadrônico. O primeiro tem a função de medir com precisão a energia de elétrons (pósitrons) e γ 's (fótons). O segundo, como o nome indica, tem a função de detetar hádrons. Esta distinção ocorre pois estes dois tipos de partículas (e.m. e hadrônica) tem propriedades fundamentais distintas e interagem de forma diferente com estes detetores:

Absorção de elétrons e fótons O melhor mecanismo conhecido de perda energética é a interação eletromagnética experimentada por partículas carregadas que atravessam a matéria. As partículas carregadas ionizam o meio. Este processo forma o princípio sobre o qual muitos detetores se baseiam, já que os elétrons liberados no processo podem ser coletados por campos elétricos, gerando um sinal elétrico. Outros processos também podem gerar outros tipos de sinais que não serão averiguados aqui [7].

Exceto possivelmente em energias muito baixas, elétrons e fótons são absorvidos num processo de vários passos, no qual ocorre uma multiplicação de partículas (desenvolvimento de *chuveiro*), chamado de *bremssstrahlung* [8] e posterior ionização das partículas do chuveiro. Este fenômeno é a principal fonte de perda energética de elétrons e pósitrons em altas energias. Sua principal característica é a formação de fótons que podem decair em pares elétron-pósitron, que podem gerar fótons, dando continuidade ao chuveiro, que perde, então, sua energia exponencialmente.

As grandes chances de desenvolvimento do chuveiro ou cascata e.m., fazem do elétron, do pósitron e do fóton elementos muito sensíveis aos detetores. Sendo assim,

para conseguir maior precisão na estimação da energia destas partículas, posiciona-se um calorímetro e.m. altamente segmentado num plano anterior ao do calorímetro hadrônico.

Absorção de hádrons As partículas hadrônicas somente decaem quando interagem com a força forte. O desenvolvimento de cascatas hadrônicas é um processo semelhante ao *bremsstrahlung* para partículas como elétrons. Segundo Wigmans em [7]:

“Quando um hádron altamente energético penetra um bloco de matéria, ele, em algum ponto, interagirá com algum núcleo atômico. Neste processo, mésons são usualmente gerados (píons, káons, etc.). Outra fração da energia inicial da partícula é transferida para o núcleo com o qual o hádron interagiu. Este núcleo excitado, liberará esta energia emitindo um certo número de núcleons (prótons ou nêutrons) e num estado posterior, γ 's de baixa energia, perdendo sua energia cinética por ionização. As partículas produzidas nesta reação (mésons, núcleons e γ 's), por sua vez, podem perder sua energia cinética por ionização ou induzir novas reações formando uma cascata ou chuva.”

Em outras palavras, as cascatas derivadas de decaimentos de elétrons e fótons são menores, mais estreitas e penetram pouco na matéria, enquanto que as cascatas de hádrons tendem a penetrar mais nos detetores, produzindo chuvas mais longas e radialmente mais largas. Há, no entanto, a possibilidade de hádrons decaírem como elétrons e fótons, o que torna difícil, ou até mesmo impossível sua detecção usando apenas calorímetros.

2.5.1.1 Calorímetros do ATLAS

De modo a detetar de forma ótima as duas classes de partículas, calorímetros especializados na detecção e.m. e hadrônica são desenvolvidos para cada experimento, de acordo com a física de interesse. No ATLAS, as principais características destes calorímetros são:

Calorímetros Eletromagnéticos: São calorímetros cujo o material absorvedor é chumbo, e os elementos amostradores são eletrodos de chumbo imersos em Argônio Líquido, com a forma de “acordões” ao longo da direção η (pseudo-rapidez, veja Apêndice B). Os eletrodos amostram a energia da partícula através da ionização das células do calorímetro. O sistema necessita de elementos resfriadores para a redução do ruído induzido pelos campos magnéticos e pela eletrônica de detecção.

A seção e.m. do ATLAS é dividida em 3 camadas, das quais a segunda é a mais profunda. Cada camada possui uma granularidade específica, que ajuda a determinar alguns aspectos dos objetos que interagem com este detetor. Além disso, a seção e.m. é dividida em duas partes: o barril (do inglês *barrel*) e a tampa (*endcap*). O barril, e as duas tampas, completam totalmente o espaço ao redor da colisão até um valor de $\eta = 3,2$ (para maiores referências sobre o sistema de coordenadas do ATLAS, leia o Apêndice B). A Figura 2.10 mostra o posicionamento do calorímetro eletromagnético no detetor ATLAS. Sua granularidade não é mostrada nessa figura. Ao invés, mostram-se os valores de η que definem os limites geométricos da seção e.m.. Pode-se perceber que a porção do barril de tal calorímetro estende-se de $\eta = 0$ até $\eta = 1,475$. Em $\eta = 1,375$ o barril começa a sobrepor a tampa, que é dividida entre tampa exterior (até $\eta = 2,5$) e interior (de $\eta = 2,5$ até $\eta = 3,2$). Há também um calorímetro que abrange valores de $\eta > 3,2$, denominado de *Forward*. Este calorímetro não será abordado neste trabalho, em nenhum de seus níveis, uma vez que é comumente utilizado somente para efeitos de análise *off-line*.

O calorímetro e.m. do ATLAS também inclui um pré-irradiador (do inglês, *pre-sampler*), que funciona praticamente como um calorímetro muito fino, posicionado antes dos calorímetros de argônio líquido, com a função de recuperar a informação perdida no material *morto* da seção e.m. (ou seja, fios, encapamentos, etc.). O pré-irradiador também é importante para detetar píons, já que este tipo de partícula deposita bastante energia nesta camada do calorímetro e.m., ao contrário do que acontece com elétrons. O pré-irradiador pode ser observado, na figura, de $\eta = 0$ até $\eta = 1,5$ no barril, e depois, de $\eta = 1,5$ até $\eta = 1,8$ nas tampas.

Uma observação mais apurada da Figura 2.10 revela um *buraco* entre o barril e a tampa da seção e.m.. Esta falha existe para que seja possível passar a fiação acoplada aos sensores dos detetores internos. Para que a perda nessa região seja

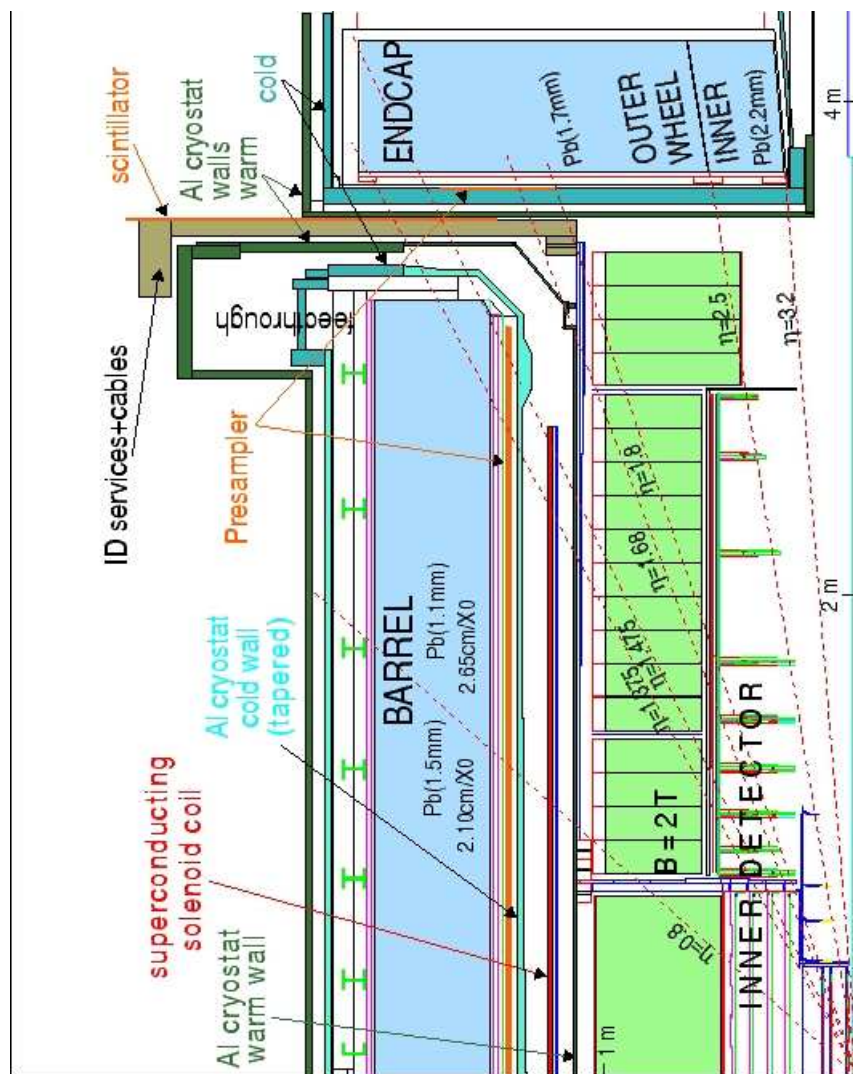


Figura 2.10: O Calorímetro e.m. do ATLAS em um corte transversal.

minimizada, decidiu-se por colocar um cintilador (indicado na figura). Cintiladores são detetores que se excitam pela passagem das partículas energéticas e produzem luz. Cintiladores são normalmente bastante compactos e finos.

Granularidade da seção e.m. O calorímetro e.m. do ATLAS possui uma granularidade⁴ constante com relação à rotação (eixo ϕ), mas variável com relação ao eixo η . Este calorímetro também é dividido em 3 camadas, com granularidades independentes. Isto quer dizer que ao longo do eixo z, a granularidade pode variar. A Figura 2.11 exemplifica a diversificação da granularidade ao longo do eixo ϕ . Cada camada é formada por células de diferentes tamanhos. Nessa figura, também verifica-se que a segunda camada é a que possui células mais profundas. É plausível esperar que mais energia seja amostrada nesta camada.

A Tabela 2.1 resume as informações de granularidade para a seção e.m., incluindo informações sobre o *pre-sampler* e as tampas. Nota-se, a partir da tabela, que a granularidade de algumas camadas varia bastante com η . A coluna da extrema direita indica o número de células numa área de $0,1 \times 0,1$ no plano $\eta \times \phi$. Esta área é uma referência para os níveis de filtragem, como será visto mais adiante.

Calorímetros Hadrônicos Os calorímetros hadrônicos do ATLAS são formados pelo TileCal ou Calorímetro de Telhas e a Tampa Hadrônica. O TileCal é um calorímetro de amostragem, cujo material pesado é uma liga com aço e os elementos amostradores são telhas cintilantes [9]. As telhas são posicionadas em planos perpendiculares aos feixes de partículas colididas e conectadas a fibras ópticas em duas de suas extremidades. Estas fibras coletam o sinal luminoso, gerado pela telha ao interagir com a partícula, e transportam-no até tubos fotomultiplicadores (PM), onde o sinal é convertido em sinal elétrico. Somadores rápidos [10] se encarregam de adicionar os sinais das telhas entre si formando células de detecção, de forma equivalente à seção e.m.. O TileCal é posicionado após a seção e.m., como é possível verificar na Figura 2.12.

Uma peculiaridade dos calorímetros hadrônicos do ATLAS é que o barril e a tampa são feitos de formas diferentes, ao contrário da seção e.m.. O calorímetro

⁴A “granularidade” de um detetor é a resolução deste no plano $\eta \times \phi$.

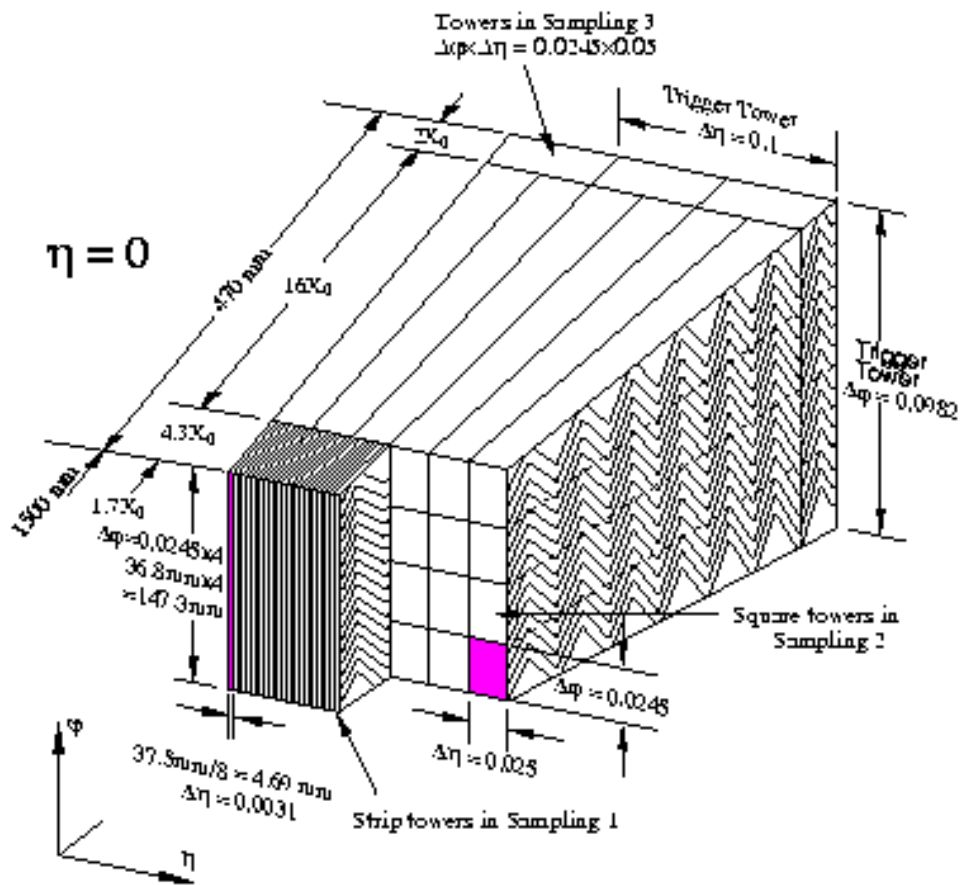


Figura 2.11: Diagrama de um dos gomos do barril da seção e.m. do ATLAS.

Tabela 2.1: A granularidade, camada a camada, dos calorímetros e.m. do ATLAS.

Camada	Peça	$\eta_{\text{início}}$	η_{fim}	$\Delta\eta \times \Delta\phi$	$N_\eta \times N_\phi$
<i>Pre-sampler</i>	Barril	0	1,5	$0,025 \times 0,1$	4×1
	Tampa	1,5	1,8	$0,025 \times 0,1$	4×1
Camada 1	Barril	0	1,4	$0,003125 \times 1$	32×1
		1,4	1,475	$0,025 \times 0,025$	4×4
	Tampa	1,375	1,5	$0,025 \times 0,1$	1×4
		1,5	1,8	$0,003125 \times 0,1$	32×1
		1,8	2,0	$0,004167 \times 0,1$	24×1
		2,0	2,4	$0,00625 \times 0,1$	16×1
		2,4	2,5	$0,025 \times 0,1$	4×1
		2,5	3,2	$0,1 \times 0,1$	1×1
Camada 2	Barril	0	1,4	$0,025 \times 0,025$	4×4
		1,4	1,475	$0,075 \times 0,025$	1×4
	Tampa	1,375	2,5	$0,025 \times 0,025$	4×4
		2,5	3,2	$0,1 \times 0,1$	1×1
Camada 3	Barril	0	1,35	$0,05 \times 0,025$	2×4
	Tampa	1,5	2,5	$0,05 \times 0,025$	2×4

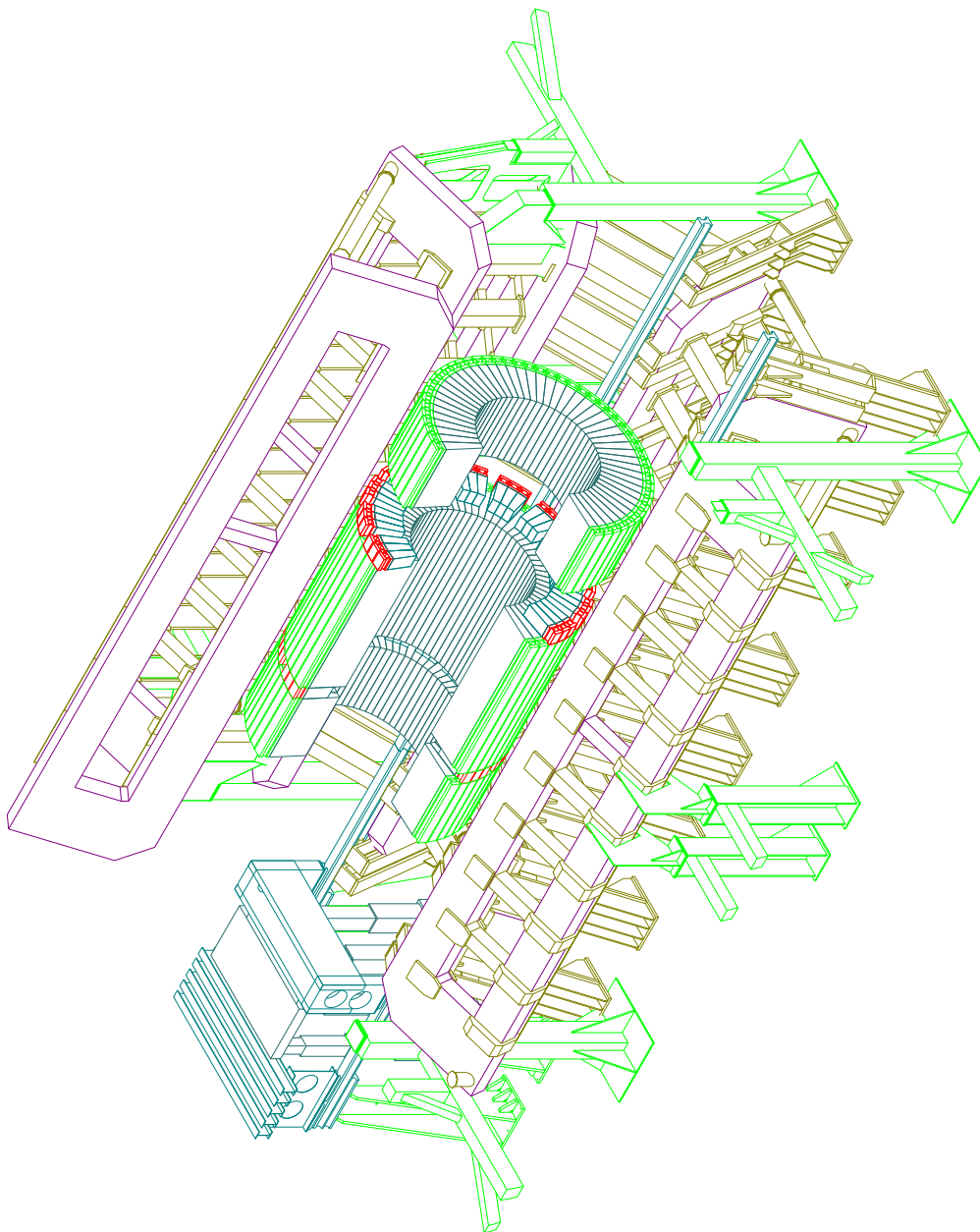


Figura 2.12: O calorímetro de telhas do ATLAS (em verde ao centro), em sua posição final, envolvido pelo detetor de múons (em roxo e marrom).

Tabela 2.2: A granularidade, camada a camada, dos calorímetros hadrônicos do ATLAS.

Camada	Peça	$\eta_{\text{início}}$	η_{fim}	$\Delta\eta \times \Delta\phi$	$N_\eta \times N_\phi$
Camadas 1 e 2	Barril (TileCal)	0	1,0	$0,1 \times 0,1$	2×2
	Barril Ext. (TileCal)	0,8	1,7	$0,1 \times 0,1$	2×2
	Tampa	1,5	2,5	$0,1 \times 0,1$	2×2
	(LAr)	2,5	3,2	$0,2 \times 0,2$	1×1
Camada 3	Barril (TileCal)	0	1,0	$0,2 \times 0,1$	1×2
	Barril Ext. (TileCal)	0,8	1,7	$0,2 \times 0,1$	1×2
	Tampa	1,5	2,5	$0,1 \times 0,1$	2×2
	(LAr)	2,5	3,2	$0,2 \times 0,2$	1×1

de telhas (TileCal) abrange as porções do barril ($0 < |\eta| < 1,0$) e sua extensão ($0,8 < |\eta| < 1,7$) da seção hadrônica dos calorímetros. A tampa desta seção é feita como os calorímetros e.m., no formato de acordeões, usando Argônio líquido.

A Tabela 2.2 resume a informação de granularidade da seção hadrônica dos calorímetros do ATLAS, de forma equivalente a da Tabela 2.1. Nessa tabela é possível perceber que o tamanho das células, em média, é bem maior que o valor equivalente no calorímetro eletromagnético. A granularidade é também mais uniforme que na seção e.m. dos calorímetros do ATLAS. Isto se deve ao fato dos chuveiros hadrônicos serem mais largos e profundos, provocando maiores flutuações nas medidas de energia e, portanto, não necessitando de uma granularidade tão fina.

Outra diferença é na área de referência. Na seção e.m., considera-se $0,1 \times 0,1$ - aqui a área de referência é de $0,2 \times 0,2$, até porque há células maiores que a área de referência no calorímetro e.m..

A Figura 2.13 mostra uma seção transversal da parte do Barril do Calorímetro de Telhas. Nessa figura, verifica-se que o sistema de leitura agrupa as células deste detetor em 3 camadas. A granularidade no sentido de η é mantida constante ainda assim.

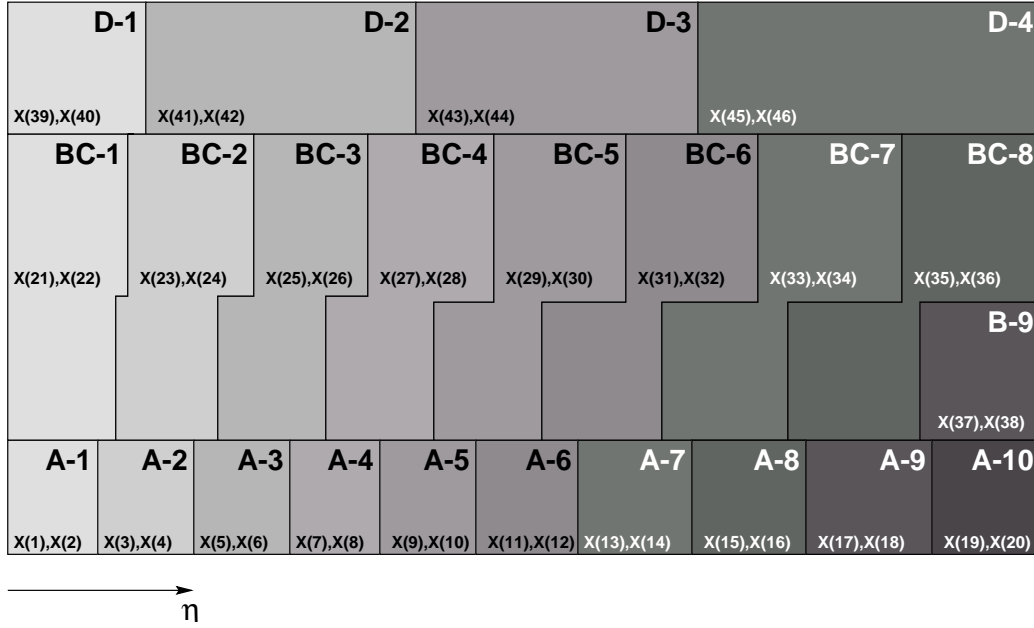


Figura 2.13: Configuração de leitura das células da seção do barril do Calorímetro de Telhas.

2.5.2 O Sistema de Filtragem de Eventos do ATLAS

Cada evento registrado no ATLAS é composto dos dados de **todos** os detectores e ainda dados de controle. Isto representa um volume considerável de dados, que está na ordem de alguns megabytes [11]. Ademais, a taxa inicial de eventos de (aproximadamente) 40 MHz do experimento ATLAS torna impraticável o acúmulo de tanta informação num curto espaço de tempo. No tempo de vida do experimento, ou seja, 10 anos, o número de eventos gravados em mídia permanente seria de 146 bilhões, ou cerca de 200 milhões de terabytes! Este universo tornaria impraticável qualquer avanço científico, já que a ocorrência de um bóson de Higgs é extremamente rara. Com a taxa do ATLAS espera-se que apenas alguns bósons de Higgs sejam criados por dia...

Para resolver esse problema, pretende-se acoplar aos detectores do ATLAS um sistema de filtragem extremamente eficiente, que rejeite eventos representantes da física ordinária (ruído) e retenha os eventos que sejam potenciais decaimentos de bósons de Higgs (sinal). Para atender às taxas de entrada, o sistema deverá ser dividido em três níveis, que serão conectados em cascata, com complexidade operacional crescente e velocidade de processamento decrescente. Cada nível eliminará

de forma mais eficiente eventos que o nível predecessor, apesar de lidar com eventos mais difíceis de serem identificados como ruído.

2.5.2.1 O primeiro nível de filtragem

O primeiro nível de filtragem do ATLAS (L1) será construído a partir de lógica de baixa programabilidade (ASICs e FPGAs [12]), mas extremamente veloz. Sua meta é reduzir a taxa inicial de eventos de 40 MHz para não mais do que 100 kHz. Repare que cada evento deve ser analisado num espaço de tempo inferior a $1/40MHz = 25ns$.

Para executar suas rotinas de filtragem, o primeiro nível utilizará somente os calorímetros e os detetores de múons do ATLAS. Ele identificará as assinaturas básicas da física de interesse, baseando sua decisão na multiplicidade dos seguintes objetos encontrados:

1. Objetos locais:

- múons;
- objetos e.m., em que haja algum isolamento (a energia na periferia do objeto deve estar abaixo de algum patamar);
- jatos de partículas colimados (podendo significar decaimentos de hádrons ou taus);
- jatos.

2. Objetos globais:

- energia transversal⁵ faltante (devido a neutrinos) acima de um determinado patamar;
- energia transversal total acima de um determinado patamar.

Os objetos globais não serão estudados neste trabalho. A referência [12] pode ser consultada para mais informações sobre estes objetos.

O primeiro nível deverá detetar os objetos locais usando sistemas de discriminação baseados nos dados dos calorímetros e detetores de múons do ATLAS. Por

⁵A energia é dita transversal quando somente as componentes x e y da energia são contabilizadas.

exemplo, objetos do tipo e.m. e jato são detetados usando somente os dados dos calorímetros. Para isto, o primeiro nível secciona o calorímetro segundo blocos de células de $0,1 \times 0,1$ em $\eta \times \phi$ (*Trigger Towers*⁶) e desloca uma janela móvel de $0,2 \times 0,2$ sobre toda a área do calorímetro. Os valores de energia depositados em cada janela são comparados a um patamar. Quando o valor de energia excede ao valor do patamar, considera-se que um objeto foi encontrado. Uma região do detetor é destacada após o centro do objeto tenha sido determinado. Esta região é chamada de Região de Interesse (*Region of Interest*, RoI). Extraí-se do objeto, i.e., da Região de Interesse, as seguintes características:

- A energia total do objeto a partir dos valores de energia amostrados em cada *trigger tower*;
- O isolamento do objeto, medindo as diferenças energéticas entre o centro e a periferia do objeto;
- A penetração do objeto, a partir da trajetória do objeto pelas camadas do calorímetro.

Os objetos encontrados são classificados dependendo das quantidades de energia, isolamento e penetração. Paralelamente a este esforço, uma análise no detetor de múons verifica se há múons no evento.

Um evento é aceito basendo-se na multiplicidade e qualidade dos objetos locais encontrados. Para isto, compara-se a listagem de objetos locais do evento com uma tabela de assinaturas (decaimentos de Higgs). Eventos que se adequam às restrições dessa tabela são aceitos e repassados ao segundo nível de filtragem para uma análise mais depurada.

Durante o processamento do L1, os dados de todos os detetores são guardados em memórias *pipeline*. Quando o evento é aceito, os dados são lidos do *pipe*, formatados, opcionalmente pré-calibrados, e gravados nos *buffers* de leitura (do inglês *ReadOut Buffers*, ou ROB's) para serem lidos pelos posteriores níveis de filtragem.

⁶A utilização de *trigger tower* no processamento dos dados dos calorímetros é fundamental no primeiro nível de filtragem, pois reduz o tempo de processamento.

Os ROB's estão diretamente associados a regiões dos detetores, de tal forma que posterior consulta dos dados de uma região do detetor deverá ser feita observando-se os *buffers* associados àquela localidade.

2.5.2.2 O segundo nível de filtragem (L2)

O L2 será composto de processadores comercialmente disponíveis (como computadores pessoais – PC's) e conectados usando-se uma tecnologia de rede que seja suficiente aos propósitos do projeto. A cooperação do L2 tem em mente, no entanto, preço e disponibilidade comercial como parâmetros de escolha primários. Sendo assim, redes *gigabit-ethernet* ou ATM (*Asynchronous Transfer Mode*) são consideradas as melhores opções atuais.

Os processadores do segundo nível de filtragem avaliarão, com maior precisão, os objetos encontrados pelo L1. Para isto, usarão toda a granularidade disponível nos detetores do ATLAS, destacando janelas ao redor dos pontos de interação assinalados pelo L1. Desta forma, somente uma fração dos dados do evento será transferida dos ROB's para o processador designado, reduzindo a necessidade de banda-passante na rede de conexão L2-ROBs e de poder de processamento neste nível. A seguir, descreve-se as etapas executadas durante o processamento de cada evento no L2:

Análise Local No L2, cada RoI é examinada no detetor na qual foi originada (no detetor de múons ou nos calorímetros) de forma a confirmar a validade do objeto. Em paralelo ou após esta confirmação do objeto, características adicionais podem ser procuradas em outros detetores, como os detetores de traço. Este é o caso de múons, objetos e.m. e taus. As RoI de jatos são somente processadas com dados dos calorímetros. Há duas formas de executar a análise local:

- **Extração de Características:** Neste método extrai-se um certo número de características do objeto, tendo como base os dados dos diversos detetores. Após a extração das características, os objetos são confirmados.
- **Análise Direta** Neste caso, um discriminador define a validade do objeto selecionado pelo L1. A extração de características pode ocorrer em paralelo ou após a confirmação do objeto. Como será visto, discriminadores especializados

podem conseguir um desempenho **muito** superior neste caso, em relação ao uso das características físicas dos objetos.

Decisão Global As informações colhidas são encaminhadas para um sistema de decisão global, que compara os objetos encontrados no evento com um conjunto de assinaturas, mais acuradas que no L1, representativas de bósons de Higgs. O sistema de decisão global responde a um supervisor, indicando a validade do evento, que é repassado ao nível de filtragem seguinte ou apagado dos *buffers*.

2.5.2.3 O Filtro de Eventos

O Filtro de Eventos (do inglês *Event Filter*, ou EF) completa o sistema de filtragem do ATLAS. Ele representa o terceiro nível de processamento. Aqui, o evento é totalmente coletado dos ROBs e analisado, levando-se em conta toda a granularidade disponível. O tempo de processamento máximo permitido está na ordem de alguns segundos. Uma análise global é possível aqui. Fenômenos mais complexos poderão ser estudados e filtrados, tais como a reconstrução de vértices, *fitting* de traços e recuperação de *bremsstrahlung* para elétrons.

A tecnologia empregada no terceiro nível de filtragem é equivalente à do segundo nível. Atualmente, existem claras intenções de acoplamento de sua funcionalidade ao L2, tornando sua existência explícita desnecessária.

A Figura 2.14 mostra um esquema representativo da conectividade do sistema de filtragem do ATLAS como um todo. Na parte superior, um bloco representa os detetores. Conectado às memórias *pipeline*, que se ligam aos calorímetros e ao detetor de múons, o primeiro nível de filtragem consegue reduzir a taxa de eventos inicial cerca de 1000 vezes. Os dados são re-organizados e acumulados nos ROBs, no caso de terem sido aceitos pelo L1. A informação de regiões de interesse é repassada ao L2, que, com acesso à granularidade total do detetor, pode refinar a decisão do L1. A taxa de eventos de saída do L2 não é maior que 1 kHz. Os eventos remanescentes são repassados ao EF, para o processamento global e, se for o caso de serem aceitos, gravados em mídia permanente. A taxa de eventos final não é maior que 100 Hz.

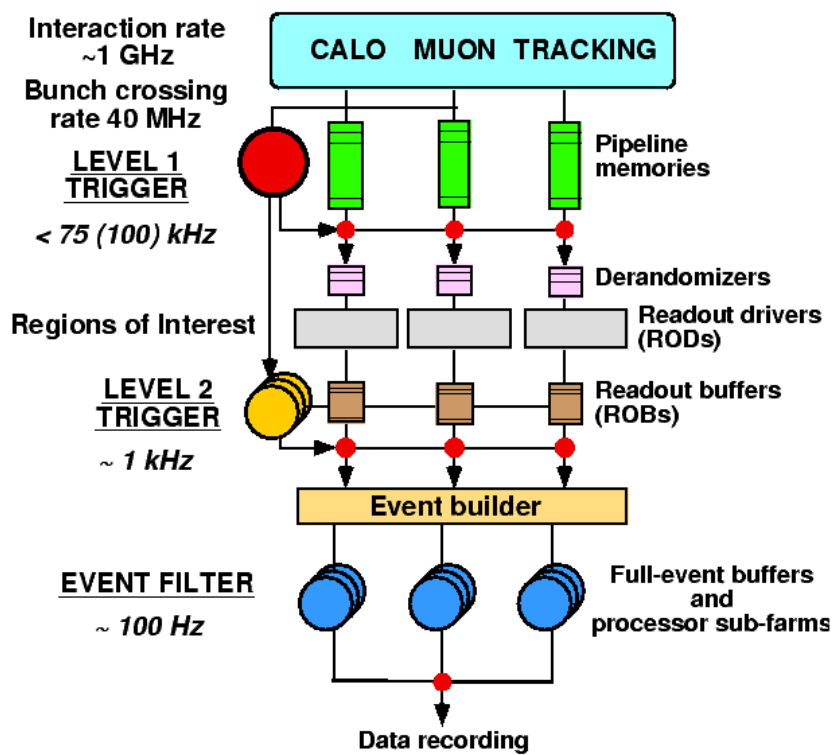


Figura 2.14: Um esquema do sistema de filtragem do ATLAS.

Capítulo 3

O Segundo Nível de Filtragem do ATLAS

Este trabalho desenvolve técnicas de processamento para o segundo nível de filtragem (L2). Neste nível é feita a primeira análise do evento tendo em vista suas características globais [11]. Ainda assim, tal análise é chamada de análise local. O processamento é assim designado pois leva em conta somente áreas destacadas pelo primeiro nível de filtragem no detetor.

Espera-se que o L2 seja capaz de reduzir sua taxa de entrada de 100 kHz para não mais que 1 kHz na saída para o Filtro de Eventos (EF). Esta é, sem dúvida, uma restrição muito genérica, levando-se em consideração a diversidade de operações que este nível de filtragem deve realizar. Alguns algoritmos serão mais velozes que outros, e portanto, alguns canais¹ tenderão a ser processados de forma mais eficiente que outros. Como então garantir que o sistema todo funcionará na taxa de eventos esperada? Basta garantir que nenhum dos canais seja processado num intervalo de tempo maior do que a taxa de primeiro nível para aquele canal. Assim sendo, se espera que dos 100 kHz de entrada do primeiro nível, 10% representem o mesmo tipo de decaimento, o que indica que o tempo de processamento para este canal no segundo nível não poderá ser maior que $\frac{10}{100\text{kHz}} = 100\mu\text{s}$.

Neste caso, é possível imaginar o segundo nível como um conjunto de “tubos” (*pipes*). Cada “tubo” é responsável pelo processamento de um canal. Eventos de um

¹Um canal representa uma “assinatura de Higgs”, seguindo as restrições para os objetos daquela assinatura, num determinado nível de filtragem.

canal chegam ao L2, segundo uma taxa determinada pelo primeiro nível de filtragem. É claro que esta taxa é um valor estatisticamente tomado, então uma margem de segurança pode ser aplicada tanto na taxa dos canais de entrada do L2 (aumentar) ou no tempo de processamento para cada canal (reduzir). Um sistema “supervisor” é o responsável por receber os dados do L1 e repassá-los ao “tubo” respectivo.

É claro que a idéia dos “tubos” é apenas representativa do conceito de processamento orientado à taxa de eventos de cada canal do experimento ATLAS. Realisticamente, gostaria-se de possuir um sistema mais complexo que pudesse se configurar sob-demanda, ou seja, houvesse uma distribuição da carga de trabalho no L2 de tal forma que os “tubos” se re-dimensionassem em intervalos de tempo, conforme o número de eventos de cada canal naquele intervalo. Neste caso, o tempo de processamento seria limitado pelo sistema supervisor do L2, e dependeria do número de eventos de cada canal.

3.1 Recursos \times tempo de processamento no L2

O tempo de processamento de cada canal não é um fator somente de sua taxa de entrada associada. Há também o poder de processamento associado a cada canal. Isto quer dizer que o mesmo fluxo de eventos pode ser tratado por mais de um processador, mais rápido ou reciprocamente, um maior fluxo de eventos pode ser tratado por mais processadores no mesmo espaço de tempo anterior. Neste caso, é possível dizer que o número de processadores associados a cada canal, depende da carga de cada canal, ou da dificuldade de processá-lo. Canais mais complexos podem ter mais processadores a eles disponibilizados, e canais mais simples, menos.

A Figura 3.1 mostra um gráfico que explora este conceito, tomando por base, a taxa de eventos de entrada no L2. Suponha que só exista um canal de entrada para o L2, e portanto somente um tipo de assinatura a ser identificada. Neste caso, todos os processadores (e também a rede que os interliga) no L2 estariam dedicados ao processamento de cada evento de entrada. Se a taxa de ocupação² da rede e dos

²A *taxa de ocupação* mede o percentual do recurso que está sendo utilizado. Portanto, uma *taxa de ocupação* de 0,5 ou 50% quer dizer que o recurso está com 50% de sua capacidade de processamento aproveitada.

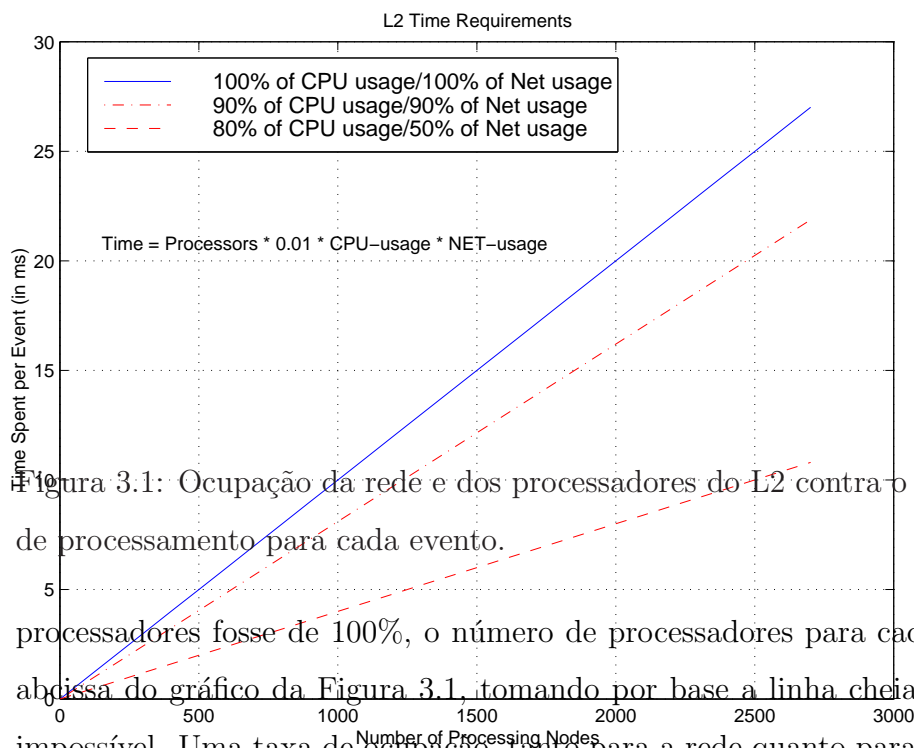


Figura 3.1: Ocupação da rede e dos processadores do L2 contra o tempo necessário de processamento para cada evento.

processadores fosse de 100%, o número de processadores para cada evento está na abscissa do gráfico da Figura 3.1, tomando por base a linha cheia. Isto, de fato, é impossível. Uma taxa de ocupação, tanto para a rede quanto para os processadores de 100% é no mínimo impraticável. Valores mais realísticos apontam para a segunda (traço-e-ponto) ou para a terceira reta (tracejada), que estão abaixo desta primeira. Na segunda hipótese, a ocupação da rede e dos processadores seria de 90%. Já no caso da terceira a ocupação da rede seria de 50%, enquanto que dos processadores, de 80%. É claro que estes valores hipotéticos representam médias.

Então, para calcular-se o tempo de processamento de cada evento no L2, é necessário levar em consideração a taxa de ocupação de seus recursos. Não é, no entanto, difícil de ver que o número de processadores estará na casa dos milhares. Atualmente estima-se que, em média, 10 ms serão necessários para cada evento no L2 [11]. Isto quer dizer: 1000 processadores para a primeira curva, em torno de 1200 para a segunda e 2500 para a terceira (!), considerando-se as taxas de ocupação mostradas na Figura 3.1.

3.2 Organização Lógica do L2

Não é uma atividade simples controlar milhares de processadores operando simultaneamente. Por exemplo, as chances de um processador falhar aumentam muito com o número de processadores envolvidos no sistema. Sendo assim, é necessário que se crie um sistema redundante e capaz de distribuir o trabalho sob-demanda, como mencionado anteriormente.

A Rede O primeiro problema de projeto a ser resolvido, no entanto, é a rede. Sua ocupação está seriamente comprometida necessitando-se conectar tantos processadores. Uma solução é utilizar sistemas de processamento simétrico (*Symmetric Multi-Processing*, ou SMP) . Nestes sistemas, muitas CPUs idênticas têm acesso ao mesmo conteúdo de memória, e de fato, pertencem ao mesmo nó de computação. Muitos sistemas operacionais, dentre eles Linux e Windows NT suportam este tipo de arquitetura. Muitos fabricantes já possuem versões multi-processadas de suas CPUs.

Adotando-se SMPs seria possível ter nós de processamento com muitos processadores. Assim sendo, alguns nós corresponderiam a uma unidade lógica (UL) de processamento. Desta forma, se reduziria e otimizaria o uso da rede para acesso a outros nós. Ainda, a comunicação entre processadores do mesmo nó também seria ótima, pois se daria através da memória compartilhada. A distribuição da carga em cada processador dos nós é feita pelo sistema operacional, e os programadores não têm que se preocupar com isto. Correlacionando com nosso exemplo anterior usando “tubos”, uma UL seria equivalente a um “tubo”.

Existem 3 tipos de ULs no L2:

- **Decisor Local:** Os decisores locais são as unidades responsáveis por confirmar os objetos selecionados pelo L1. Eles podem ser divididos, dependendo da estratégia de processamento adotada em:
 - **Unidade de Pré-processamento:** Esta unidade é responsável por extrair dos ROB's os dados necessários para a confirmação da RoI destacada pelo L1. Os dados são pré-analisados e organizados nesta unidade;

- **Discriminador Local:** Nesta unidade ocorre a confirmação do objeto destacado pelo L1. O discriminador local pode empregar qualquer método de identificação de objetos;
- **Extrator de Características:** Nesta unidade ocorre a extração de características do objeto, por exemplo, seu momento transverso, energia, isolamento, etc.

Existem decisores locais para os vários tipos de objetos locais e globais que podem ser indicados pelo L1;

- **Decisor Global:** Os decisores globais reúnem os dados de todos os decisores locais para um evento, comparando seus objetos contra uma tabela de assinaturas que representem a física que se deseja observar. Hipoteticamente, qualquer conjunto de objetos poderia ser configurado, permitindo altíssima flexibilidade na escolha do “cardápio” de um determinado teste;
- **Supervisor:** Os supervisores são os agentes que se comunicam com o L1 e o EF. Eles recebem os dados do L1, e ordenam atividades às UL’s no L2. Após receber o aval do decisor global sobre um evento, o supervisor decide, dependendo de outras variáveis, se aceitará ou rejeitará o evento, repassando-o ao EF.

A Figura 3.2 mostra um esquema simplificado da interconexão entre as ULs do L2. Na parte superior-direita da figura é possível observar o detetor ATLAS, em escala. Conectado diretamente ao ATLAS, encontra-se o L1 e os ROB’s. O L1 pré-filtra os dados gerados no ATLAS, dando o sinal de controle para o armazenamento dos dados nos ROBs, caso o evento seja aceito naquele nível de filtragem. O L1 então comunica-se com o L2, repassando os dados colhidos. O L2, através de seu supervisor, ativa suas UL’s, gerando uma decisão sobre o evento observado. Se a decisão de manter o evento for aceita, o L2 recolhe os dados do evento e repassa-os, todos, ao EF. A legenda na parte inferior-esquerda do gráfico define os elementos existentes no L2.

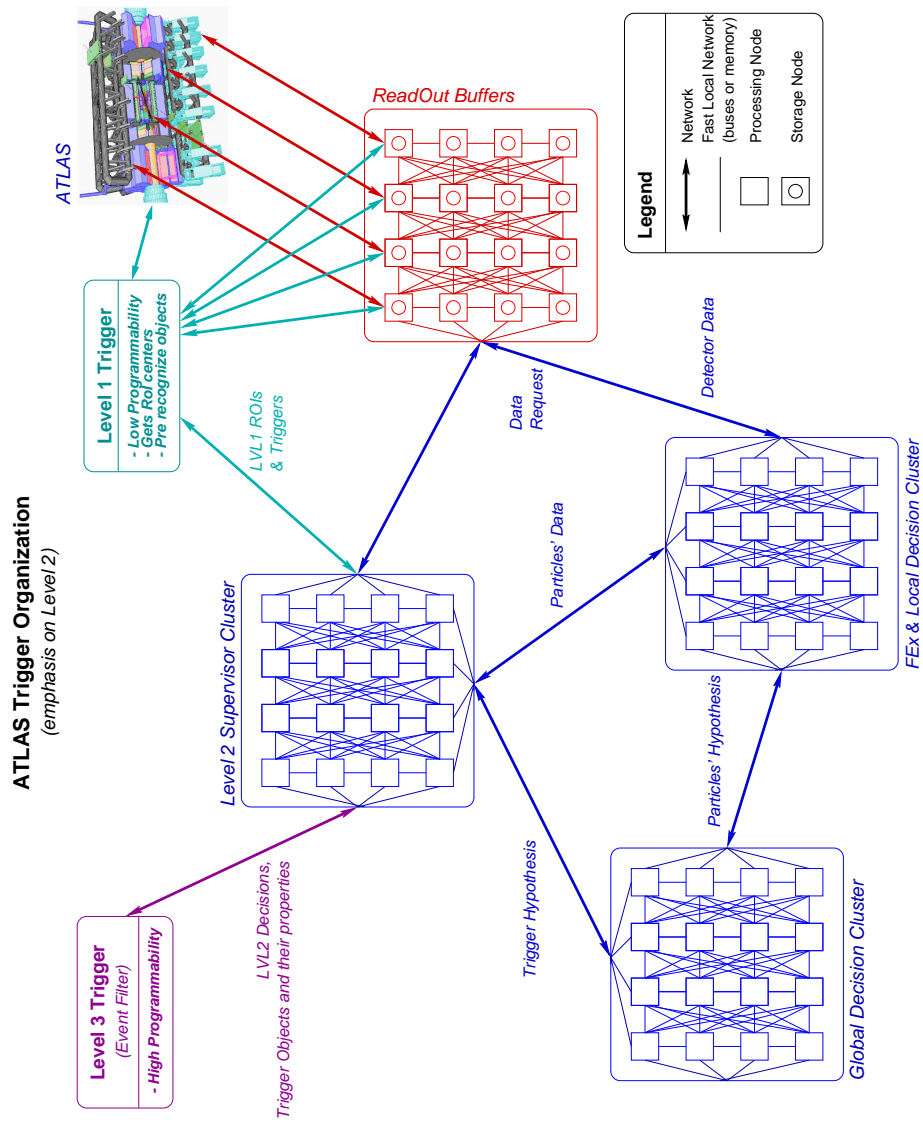


Figura 3.2: Esquema simplificado de processamento no L2.

3.3 Estruturas e Fluxo de Dados no L2

Até agora, introduziu-se os diversos aspectos da organização lógica do L2. Nesta seção, focalizar-se-á a atenção dos dados que trafegarão por esta estrutura lógica. A Figura 3.2 pode ser usada como referência.

Quando um evento é aprovado pelo L1, é “empacotado” numa estrutura que contém elementos de controle, as coordenadas do centro de cada objeto local (centro da RoI) e as características de isolamento, energia, etc, de cada objeto local e global que foi definido. Os dados dos detetores são gravados nos ROBs, de forma que níveis subsequentes de processamento possam utilizá-los.

O L1, então, sinalizará ao supervisor do L2 que há um evento aprovado, descarregando os dados nesta UL. Ao receber a informação, o supervisor irá assinalar um ou mais decisores locais para confirmar a filtragem do L1. Estes decisores produzirão um conjunto de características daquele objeto e o avaliarão. Após esta etapa, o supervisor do L2 é novamente acionado, designando unidades de decisão global para colher os dados dos diversos decisores locais e avaliar o evento como um todo. Os dados que são repassados ao decisor global são valores mais acurados de isolamento, energia transversa e momento dentre outros. Essas quantidades que descrevem o objeto e são denominadas *características* do objeto (*features*), e representam a física do elemento sob análise.

Os decisores globais comparam os objetos e suas características com um conjunto de assinaturas que representam a física de interesse. Independente de casarem ou não com as assinaturas, o decisor global repassará os dados de sua operação para o supervisor, onde a decisão final de aceitar ou rejeitar o evento será tomada.

O supervisor do segundo nível receberá o aval do decisor global designado para aquele evento. Dependendo dos limites operacionais configurados, ele poderá aceitar o evento ou rejeitá-lo. Caso aceite o evento, o supervisor deverá reunir **todos** os dados que foram extraídos do evento, seus objetos locais e globais, a assinatura com o qual o evento *casou*, os valores de controle do evento, etc. Estes dados devem ser encaminhados ao EF.

O tempo 10 ms discutido na Seção 3.1, representa o intervalo de tempo que o L2 terá para executar todos os passos descritos aqui. Isto inclui tempos de comunicação entre processos e processadores, de execução de algoritmos e, obviamente,

de quaisquer outras atividades que devam ser conduzidas pelo L2.

3.4 Os Decisores Locais

As unidades de decisão local são, sem dúvida, o maior gargalo, em termos de tempo de processamento, no L2. Ou seja, de todas as atividades desempenhadas pelo L2, as unidades de decisão local representam a maior quota de tempo de processamento. Isto é evidente analisando-se a estrutura do L2: o supervisor é tão somente uma unidade de organização do trabalho e os decisores globais, unidades de consulta de um banco de dados, i.e., a tabela com as assinaturas da física de interesse.

Os algoritmos de reconhecimento e extração de características no L2 devem analisar os dados de cada RoI, de tal forma a avaliar da melhor maneira os objetos que interagiram naquela porção do detetor. Este trabalho pode ser bastante difícil, já que, em muitos casos [11], não é evidente que um objeto seja confirmado baseando-se somente nos dados de um detetor. Estes objetos necessitam muitas vezes de extrapolações de seus traços nos detetores internos para a confirmação de valores de momento ou da identificação de uma possível multiplicidade³.

Por outro lado, devido aos requisitos de tempo, seria importante uma discriminação eficiente através dos algoritmos de decisão local, pois:

- Quanto mais eficientes são os algoritmos de confirmação de objetos, baseados em dados dos calorímetros e do detetor de múons, menor a quantidade de dados que são transferidos dos ROBs para o L2;
- Quanto mais eficientes os algoritmos, menor o tempo de processamento de cada objeto;

Independentemente da eficiência dos algoritmos, uma estratégia de processamento deve ser escolhida. A mais eficiente, como já colocado na Seção 2.5.2.2, é a confirmação inicial dos objetos nos processadores locais, usando-se somente o

³Pode acontecer que um objeto destacado pelo L1 venha ser, na verdade, a ocorrência de 2 ou mais objetos muito próximos. O L2 deve esclarecer tais casos.

detetor pelo qual o objeto foi inicialmente destacado no L1. Se o objeto for confirmado e medidas mais precisas forem necessárias, outros decisores locais associados a diferentes detetores são disparados. Sendo assim, por exemplo, quando deseja-se confirmar um objeto EM, o processo deverá ser conduzido usando-se os dados dos calorímetros do ATLAS. Se a confirmação é para um múons, utiliza-se o detetor de múons.

3.4.1 Decisores Locais para os Calorímetros

Os decisores locais, que analisam os objetos destacados pelo L1 usando os calorímetros, executam diferentes algoritmos. A Tabela 3.1 correlaciona os tipos de algoritmos no L2 aos objetos destacados pelo L1. Com exceção do primeiro, os dois outros algoritmos precisam consultar outros detetores para a definição de seus objetos. O primeiro algoritmo tem que identificar, dos objetos EM destacados pelo L1, aqueles que realmente são resultantes da interação de elétrons ou fótons (γ). O segundo classifica jatos estreitos, definindo se são provenientes de decaimentos de hádrons (representados por p/\bar{p} na tabela) e taus (τ) e separando-os de outros tipos de jatos. Para isto, consulta os detetores de traços para mais informações sobre os eventos. O terceiro algoritmo destina-se à classificação de jatos quanto à sua energia e multiplicidade. Neste último quesito, deseja-se compreender se há mais de um jato agrupado na mesma RoI.

Outra informação mostrada na Tabela 3.1 é o tamanho da RoI que deve ser transferida para os decisores locais. A dimensão da RoI depende do algoritmo a ser transferido, mas é sempre um múltiplo do tamanho das torres de filtragem do primeiro nível. Os valores mostrados são para os algoritmos atualmente utilizados pela colaboração do sistema de filtragem do ATLAS [11]. A Tabela 3.2 indica o número de *buffers* de leitura associado a alguns detetores do ATLAS. A segunda coluna desta tabela indica o número de ROB's que deverão ser ativados, em média, por RoI (com tamanho $\Delta\eta \times \Delta\phi = 0,4 \times 0,4$). A coluna seguinte indica o número máximo de ROB's por RoI, e por final, a última coluna indica o tamanho, em bits, da informação que será transferida dos ROB's para o L2 se uma RoI naquele detetor for requerida.

De forma a transferir uma RoI para análise em um decisor local para os

Tabela 3.1: Tipos de algoritmos no L2 e objetos relacionados no L1.

Objeto no L1	Algoritmo no L2	Tamanho da RoI $\eta \times \phi$
EM	discriminação $e^{+-}/\gamma \times$ jato	$0,4 \times 0,4$
Jatos Estreitos	discriminação $\tau/p/\bar{p} \times$ jato	$0,4 \times 0,4$
Jatos	classificação de jatos	$1,0 \times 1,0$

Tabela 3.2: Número de ROBs por detetor. Esta tabela também mostra o volume de dados que devem ser guardados em cada ROB.

Detetor	Número de ROBs	ROB/RoI (média)	ROB/RoI (máximo)	RoI data (kilobit)
SCT	194	3,3	8	13
TRT	256	5,2	6	31
ECAL	208	4,4	6	46
HCAL	24	2,0	4	14

dados dos calorímetros, transfere-se $46 + 14 = 60$ kbits, ou seja, 7,5 quilobytes. O número médio de ROBs que deverá se consultar será $4,4 + 2,0 = 6,4$ ROB's e o número máximo $6 + 4 = 10$ ROB's. Repare que, por possuir menos ROB's acopladas, o calorímetro hadrônico do ATLAS (HCAL na tabela) terá cada um de seus ROB's consultado mais vezes [13]. Sendo assim, algoritmos que priorizam a utilização do calorímetro EM são mais eficientes. Para os dados dos calorímetros há mais uma característica interessante: é garantido que os dados de uma torre de filtragem estejam concentrados em apenas um ROB. As torres de filtragem, como mencionou-se na seção 2.5.2.1, representam a máxima granularidade visível no L1.

3.5 Decisão Local com objetos EM

Os objetos EM são importantes pois podem representar elétrons e fótons. Elétrons e fótons, por sua vez, representam canais muito comuns no experimento [11]. Estes elementos, no entanto, sofrem freqüentemente a contaminação por jatos

de partículas, que interagem com os calorímetros e falseiam assinaturas de física significativa. Estes jatos colimados confundem o L1, pois têm características de deposição e isolamento muito parecidas com a de elétrons e fótons, dada a granularidade de processamento no L1. Uma análise mais apurada somente pode ser conduzida no L2, usando-se toda a granularidade disponível nos calorímetros.

Para realizar a discriminação entre elétrons, fótons e jatos e a respectiva extração de características, o decisor local do L2 poderá carregar os dados dos calorímetros para aquele objeto destacado pelo L1. Os dados, no entanto, devem ser interpretados e pré-analisados antes da execução dos algoritmos de filtragem. Nesta seção, discuti-se o formato dos dados e formas de pré-processamento destes dados para que se adequem às necessidades dos algoritmos de decisão local. Dentre as etapas que serão destacadas tem-se:

1. Decodificação do posicionamento das células;
2. Supressão de valores indesejados;
3. Verificação de dados faltantes e sua compensação.

3.5.1 Decodificação da posicionamento das células

As células do calorímetro são identificadas por um campo determinado e pelo valor de energia amostrado para um determinado evento, formando uma estrutura com 2 campos de 32 bits. Assim sendo, antes de executar qualquer outro algoritmo, o decisor local terá que identificar a posição das células que recebe, de forma a processá-las.

A completa identificação das células inclui seu posicionamento no plano $\eta \times \phi$, o tipo de calorímetro e a camada a qual pertencem. Estas informações estão empacotadas em um campo de 25 bits, organizado como mostra a Figura 3.3. Nesta figura, observa-se que os 8 bits menos significativos guardam a posição da célula em ϕ , enquanto que os 9 bits seguintes, a posição em η , juntamente com um bit de sinal. Os campos seguintes, continuando da direita para a esquerda, guardam o número da camada e o tipo de calorímetro a qual a célula pertence. Os últimos 7 bits são inseridos apenas para completar uma palavra de 32 bits.

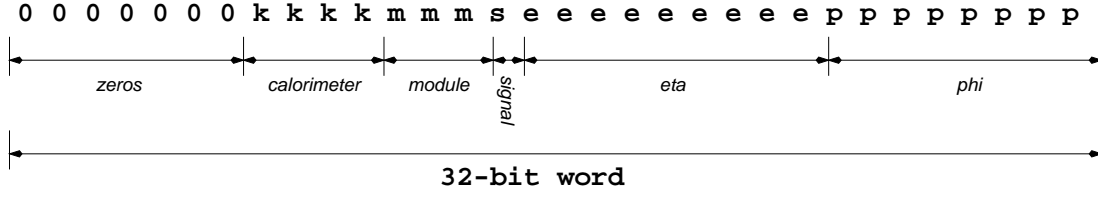


Figura 3.3: Como o posicionamento de cada célula do calorímetro é empacotado em um campo de 25 bits.

Após a decodificação do campo de bits, o processador poderá posicionar as células no plano $\eta \times \phi$ e prosseguir com a supressão de valores indesejados.

3.5.2 Supressão de valores indesejados

O tamanho de uma RoI depende diretamente do algoritmo que realizará a discriminação, sendo este sempre múltiplo do tamanho das torres de filtragem no L1. Esta é uma escolha evidente, já que os dados de uma mesma torre de filtragem encontram-se concentrados no mesmo ROB. Portanto, através da consulta de uma tabela, é possível descobrir a quais ROB's se devem requisitar os dados.

A requisição de dados dos ROB's acarreta um problema: o ROB deverá transferir todos os seus dados ou somente os dados relativos às torres de filtragem requisitadas? Para poder selecionar dados, os ROB's teriam que possuir alguma capacidade de processamento, o que aumentaria a sua complexidade. Caso o ROB não venha a ter alguma capacidade de seleção de dados, então transferirá todos os seus dados para os decisores locais, possivelmente congestionando a rede. O fluxo de informação, nos dois casos, poderia ser representado assim:

1. ROB sem capacidade de processamento
 - (a) Um processador local do L2 requer os dados;
 - (b) Algum processador simples, localizado no grupo de ROB's, traduz a posição no plano $\eta \times \phi$ do centro da RoI requisitada para a identificação dos ROB's que devem ser consultados;
 - (c) Todos os dados dos ROB's são transferidos para o processador no L2. Neste local, dados que não pertençam à RoI que se deseja processar são destacados.

2. ROB com capacidade de pré-processamento

- (a) Um processador local do L2 requer os dados;
- (b) Algum processador simples, localizado no grupo de ROB's, traduz a posição no plano $\eta \times \phi$ do centro da RoI requisitada para a identificação dos ROB's que devem ser consultados;
- (c) O ROB destaca, dentro de seus dados, os dados das *trigger towers* que foram requisitadas;
- (d) Os dados das *trigger towers* são transferidos para o processador no L2.

Atualmente, considera-se que a supressão de valores indesejados será executada nos decisores locais, ainda que a decisão final não tenha sido tomada.

3.5.3 Verificação de dados faltantes

Após receber as células, decodificá-las e suprimir a informação de regiões indesejadas, o decisor local ou uma unidade de pré-processamento com este propósito terá que verificar a completude dos dados. Essa não é garantida pois, como os dados chegam de forma assíncrona, problemas diversos podem ocorrer. Dentre eles, é possível citar:

- Falha de célula(s) do calorímetro;
- Falha de parte dos ROBs conectados ao calorímetro;
- Falha de algum nó de processamento ou da rede que abastece o sistema de extração de características;
- Demora excessiva para que um certo conjunto de dados requisitados pelo decisor local chegue, etc.

A forma de se remediar esta situação dependerá do algoritmo de processamento que será acoplado ao decisor local.

3.5.4 Discriminação e Extração de Características

Depois de aplicado o pré-processamento, se faz necessária a confirmação do objeto destacado pelo L1 (discriminação) e a extração de algumas características do objeto. As características extraídas são utilizadas no processo de decisão global para adjetivar o objeto. Por exemplo, “este objeto é um elétron com 33,5 GeV de energia transversa...”

Dois problemas foram propostos aqui:

1. Extração de Características - Nesta fase, deseja-se calcular algumas das características do objeto EM sendo atualmente processado. Estas características são utilizadas na fase de decisão global, na verificação das assinaturas, e também podem ser utilizadas na fase de decisão local, para a confirmação do objeto. Elas incluem o isolamento e energia transversa (E_t) do objeto de estudo;
2. Discriminação $e/\gamma \times \text{jato}$ - Nesta fase, deseja-se certificar que o objeto EM destacado pelo primeiro nível é um elétron ou fóton, e não um jato colimado.

Estas duas fases podem ser realizadas em qualquer ordem, contanto que respeitem o sistema de decisão global e suas necessidades. A realização da extração de características em primeiro lugar deverá ser seguida da discriminação do objeto. Se primeiramente deseja se realizar discriminação, deve-se prosseguir, confirmado o objeto, com a extração de características.

O fluxograma na Figura 3.4 resume os passos aqui descritos. Nesta figura estão representados 2 fluxogramas que representam as duas formas descritas no parágrafo anterior de se realizar o processamento no decisor local. Na forma mais à esquerda (a), nota-se que o sistema discriminador atua diretamente sobre a informação pré-processada, dando a capacidade de se interromper o processamento antes da extração de características, caso o objeto naquele decisor local não seja confirmado. No segundo caso à direita (b), o sistema de discriminação conta com a extração de características para dar seu aval. Nesta hipótese, não é possível interromper o processamento de um objeto indesejado antes do final do algoritmo. Algoritmos mais eficientes são aqueles que realizam as fases de extração de características e discriminação de forma ótima, ou seja, representem as melhores alternativas para um dado tempo de processamento *versus* desempenho de discriminação.

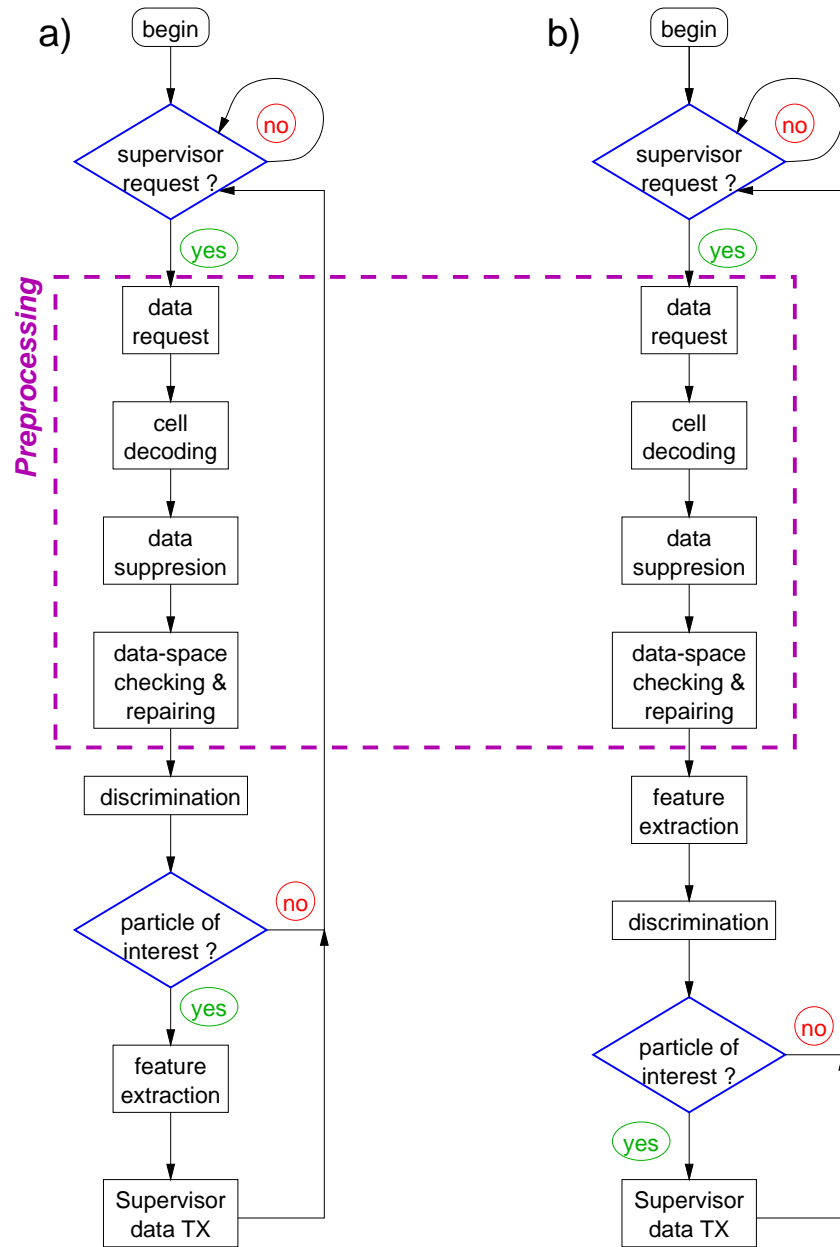


Figura 3.4: Resumo gráfico das operações para a discriminação e $\gamma \times$ jato.

Capítulo 4

O Software de Referência

O segundo nível de filtragem do ATLAS é, de fato, um sistema bastante sofisticado. Há dois níveis de complexidade que se pode destacar no projeto de tal sistema:

Complexidade dos algoritmos as rotinas que o L2 tem que executar são bastante complexas, possuindo, normalmente muitos parâmetros de configuração, e executando algoritmos bastante sofisticados;

Complexidade de *hardware* os programas estarão sendo executados sob nós de computação interligados por uma rede. A gestão deste sistema é, por si, uma tarefa complexa.

Os trabalhos de desenvolvimento de algoritmos são conduzidos pela colaboração que projeta o sistema de filtragem (ATRIG), enquanto que as plataformas de processamento recebem a atenção da colaboração do sistema de aquisição de dados (DAQ). Embora correlatos, os dois trabalhos são normalmente desenvolvidos sob diferentes escopos. A cooperação ATRIG deseja otimizar os algoritmos de filtragem no aspecto físico e os membros do DAQ desejam fazer a melhor escolha de *hardware* para o ambiente.

Esta descentralização tem que evoluir harmonicamente ao longo do desenvolvimento, pois, ao final, os algoritmos têm que rodar sobre alguma plataforma e devem ser ajustados para tal. Ademais, muitas plataformas estão disponíveis: como escolher a melhor? A solução para estes e para outros problemas está no *Software de Referência* (RefSoft), que é detalhado neste capítulo.

4.1 O Refsoft e este trabalho

No desenvolvimento de um sistema de discriminação que participe do projeto ATLAS deve-se considerar sua integração ao Software de Referência. Esta integração pode ser feita de forma fácil, caso o desenvolvedor conheça as limitações e facilidades do software. A atenção será focalizado, no entanto, nas restrições tecnológicas do Decisor Local, berço dos sistemas de discriminação de objetos no segundo nível de filtragem do ATLAS. Em específico, o sistema de discriminação de objetos relacionados ao calorímetro será detalhado, pois constitui a base para o discriminador neuronal desenvolvido.

Esta parte do trabalho foi desenvolvida pelo autor no CERN [14, 15, 16, 17, 18], considerando-se as restrições e características do Software de Referência e suas interfaces.

4.2 O que é o RefSoft?

O RefSoft é um conjunto de programas escritos em C/C++, que formam uma espécie de moldura de trabalho, onde é possível *encaixar* os diversos componentes que devem operar no contexto do L2, ou seja, os equipamentos e as rotinas de filtragem [14, 19].

O RefSoft foi desenvolvido usando o paradigma da orientação a objetos (OO) (veja o apêndice C) [15]. Desta forma, é possível idealizar os diversos componentes como peças de um grande sistema. Se, por exemplo, deseja-se substituir o componente “Sistema Operacional”, é possível fazê-lo substituindo somente uma parte do RefSoft, i.e., aquela relacionada ao sistema operacional corrente, e re-compilar os programas. Esta flexibilidade resolve grandes problemas presentes no desenvolvimento do L2:

1. Permite a integração entre as colaborações ATRIG e DAq, pois os algoritmos agora podem ser desenvolvidos sobre o mesmo projeto, onde os testes de equipamento são realizados;
2. Desenvolve uma forma justa de comparar algoritmos, plataformas e sistemas operacionais, criando um critério unificado de análise;

3. Por ser desenvolvido com o enfoque *re-usabilidade* e independência tecnológica, pois testes de algoritmos podem ser feitos sem alterar a estrutura de operação do software;
4. Reduz a duplicação de esforços em áreas que são compartilhadas por todos os usuários;
5. Permite a emulação de toda a funcionalidade do L2 em sistemas reduzidos, somente desenhados para testes específicos com diversos tipos de algoritmos.

4.3 Estrutura do RefSoft

Para que os algoritmos sejam totalmente independentes da plataforma na qual são executados, objetos representativos do *hardware* e do sistema operacional devem mascarar as diferenças entre as diversas possibilidades disponíveis.

A melhor forma de conceber a organização do RefSoft é imaginá-lo como sendo um conjunto de objetos associados a camadas . Cada camada é responsável por uma funcionalidade. Os elementos de camadas superiores dependem dos elementos das camadas inferiores. As camadas inferiores isolam as superiores das diferenças operacionais de sistema para sistema. A Figura 4.1 mostra um diagrama em blocos representativo desta idéia. Na parte inferior encontra-se a camada mais básica, o *hardware* e os sistemas de controle associados, ou seja, o sistema operacional (SO). Eles representam a escolha de sistema do usuário para *rodar* o L2. Para tornar as camadas superiores, isto é, os sistemas de controle e algoritmos do L2 independentes (supervisor, decisores locais e globais, etc), deve-se mascarar todos os componentes da plataforma e do SO que são ali utilizados. Este mascaramento garante que os algoritmos e o controle do L2 jamais necessitem de mudanças no código, caso a plataforma ou sistema operacional com o qual estamos trabalhando sejam trocados. As setas à esquerda indicam o crescimento da abstração e complexidade do L2 (à extrema esquerda) e a dependência do *hardware* (à direita).

Por exemplo, desejando-se rodar o sistema em um ambiente Linux, a simples configuração e compilação sobre a nova plataforma deveria ser suficiente, sem ajustes nos algoritmos físicos e de controle do L2.

The Reference Software Structure

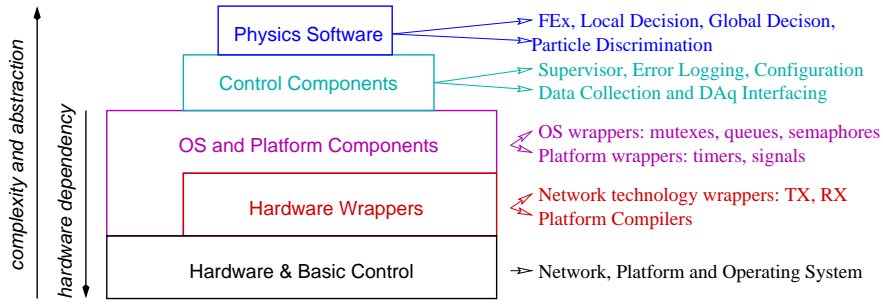


Figura 4.1: Diagrama esquemático da organização do RefSoft.

<EM Objects> Local Decision

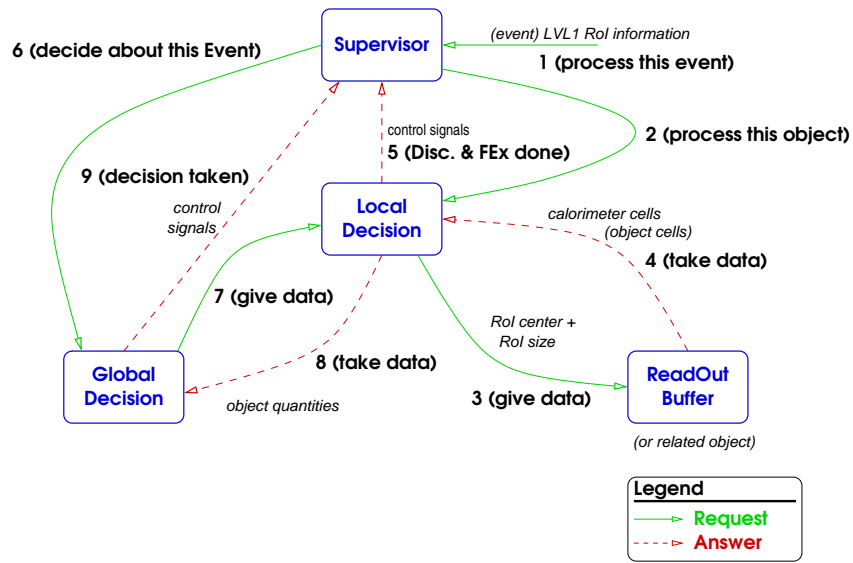


Figura 4.2: Uma análise gráfica da decisão local no L2.

4.4 O RefSoft e os decisores locais no L2

É cabível introduzir as diferentes nuances de implementação dos decisores locais no L2, pois trata-se da moldura deste trabalho. Os decisores locais, como foi visto no Capítulo 3, são responsáveis por grande parte do processamento físico no L2. Mais especificamente, analisar-se-á decisores locais de objetos EM, usando os calorímetros do ATLAS, embora o modelo possa ser aplicado a qualquer decisor local no L2.

É possível modelar a decisão local (elétron/jato, por exemplo) a partir dos seguintes passos (veja Figura 4.2) [15, 17]:

1. O supervisor do L2 recebe um evento para ser tratado;
2. Um decisor local é alocado, pelo supervisor do L2, para processar uma RoI;
3. O decisor local, que encontra-se em uma UL qualquer, faz o pedido dos dados a serem processados para uma RoI que foi indicada pelo supervisor. Neste caso, são transmitidos aos ROB's o centro da RoI, seu tamanho e um identificador do evento;
4. O decisor local recebe tais dados e realiza o pré-processamento, como indicado na seção 3.5;
5. Realiza-se a discriminação e^-/jato do objeto;
6. Realiza-se a extração de características do objeto, caso seja confirmado, e um sinal de controle é repassado ao supervisor;
7. O supervisor aloca um decisor global para tratar o evento;
8. Estas características são repassadas ao decisor global que, possivelmente recebendo dados sobre o mesmo e /ou outros objetos do evento, decide sobre a natureza do evento;
9. O decisor local transmite o sinal de controle e, possivelmente, os dados de todos os objetos processados.

O projeto OO pode ser extraído quase que diretamente desta análise e está descrito na Figura 4.3. As classes com letras em itálico são abstratas, suas funções são virtuais e apenas definem as interfaces de alguns algoritmos para o decisor local (*Local Processor*). À direita, na parte superior da figura, é possível ver duas possíveis classes que representam implementações da classe abstrata *Data Collector*. Na primeira hipótese (*File Data Collector*), os dados são recolhidos de um arquivo, trata-se portanto de uma simulação. Na segunda hipótese no entanto, recolhem-se os dados diretamente de um ROB. Para o decisor local, o processo é indiferente. O pré-processamento de dados ocorrerá neste coletor de dados.

Após receber os dados, o decisor local deverá discriminar o objeto e extrair suas características usando instâncias de implementação de classes herdeiras de

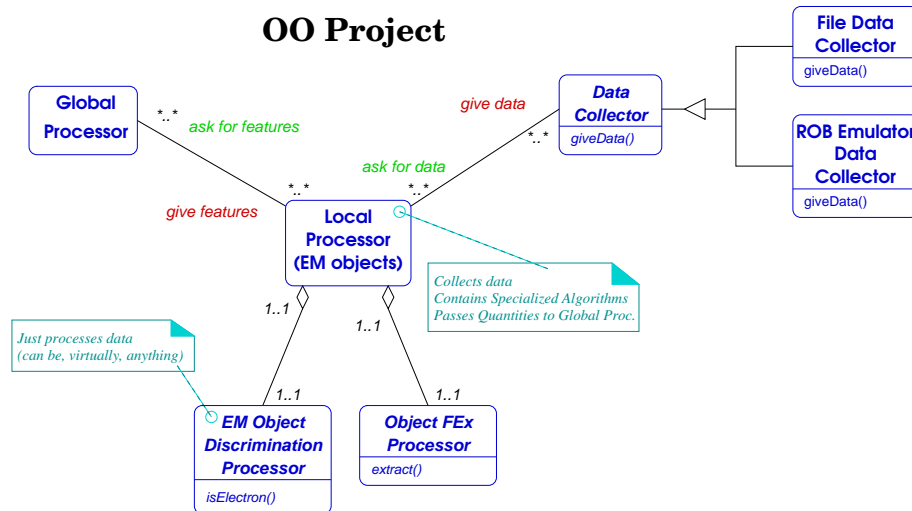


Figura 4.3: O projeto OO extraído a partir da Figura 4.2. Repare que aqui o supervisor não é modelado. O esquema segue a UML.

Discrimination Processor e FEx Processor (parte inferior do diagrama). Após o processamento e confirmação da RoI, o decisor global (Global Processor) poderá requerer os dados do objeto analisado. O decisor local deve então repassá-los à instância apropriada desta classe. As Implementações 4.1, 4.2 e 4.3 mostram como é possível sintetizar a transparência de operação requerida no projeto usando-se C++.

Nesta visão resumida da implementação, inicialmente, demonstrou-se um dos grandes trunfos de C++: a criação de modelos de classes. Modelos (do inglês, *templates*) são usados nas muitas definições de classes do L2. Nas implementações 4.1 e 4.2, é possível ver modelos definindo um padrão de comportamento para coletores de dados, discriminadores. Os argumentos do modelo são utilizados para criar a definição final da classe que será instanciada ou referenciada. Na implementação 4.1, o primeiro argumento identifica a classe de dados que será retornada por uma instância de uma classe deste modelo, enquanto que o segundo, o tipo de especificação. No L2 ambos parâmetros variam de decisor local para decisor local.

A classe criada, finalmente, tem definida sua interface para a coleção de dados. Imagina-se aqui que o sistema irá simular o L2 usando arquivos de entrada com os dados que hipoteticamente viriam dos ROBs. Prossegue-se com esta filosofia de implementação usando modelos, e definir discriminadores e extratores de características como foi visto na Implementação 4.2. Somente parte da definição das

```

// O coletor de dados é um modelo para a definição da interface
TEMPLATE <CLASS DATA, CLASS ROISPEC>
CLASS DataCollector {
PUBLIC:
    VIRTUAL DATA* collect (CONST ROISPEC*) = 0;
    ...
};

// Os tipos que definirão os dados de nossa classe abstrata.
TYPEDEF VECTOR<Cell> Cluster;
TYPEDEF STRUCT CRS {
    FLOAT eta;
    FLOAT phi;
    FLOAT etaSize;
    FLOAT phiSize;
    LONG INT eventNo;
} CaloRoISpec;

// Define-se a interface que irá coletar os dados
// automaticamente junto a sua implementação. Entre "<" e ">"
// defini-se as classes que serão utilizadas pelo modelo na
// criação desta nova classe. A primeira classe define o tipo
// que será retornado pela rotina collect(), o segundo,
// o tipo que define a especificação da RoI.
CLASS CaloFileDataCollector :
    PUBLIC CLASS DataCollector<Cluster, CaloRoISpec> {
PUBLIC:
    CaloFileDataCollector(const String*);
    VIRTUAL Cluster* collect (CONST CaloRoISpec*);
    ...
PRIVATE:
    stream* _sp;
    ...
};

// Deve-se implementar a função collect()
Cluster* CaloFileDataCollector::collect (CONST CaloRoISpec*) {
    // Aqui, descreve-se como coletar dados de arquivos.
    ...
};

```

Implementação 4.1: O Coletor de Dados (Data Collector).

```

// Os algoritmos podem ser definidos da maneira semelhante ao
// DataCollector usando-se do artifício de modelos
// (template) descrito anteriormente.
TEMPLATE <CLASS DATAIN>
CLASS Discriminator {
PUBLIC:
    VIRTUAL bool confirm (CONST DATAIN*) = 0;
    ...
};

TEMPLATE <CLASS DATAIN, CLASS DATAOUT>
CLASS FEx {
PUBLIC:
    VIRTUAL DATAOUT* extract (CONST DATAIN*) = 0;
    ...
};

// As quantidades que serão retornadas pelo FEx.
TYPEDEF STRUCT EMOBJECT {
    FLOAT Et;
    FLOAT Isolation;
    ...
} EMOBJECT;

CLASS MyEMDiscriminator : PUBLIC Discriminator<Cluster> {
    // Aqui a função confirm será declarada.
    // Sua implementação poderá vir depois
    ...
};

CLASS MyFEx : PUBLIC FEx<Cluster, EMOBJECT> {
    // Aqui a função extract será declarada.
    // Sua implementação poderá vir depois
    ...
};

```

Implementação 4.2: Os algoritmos de discriminação e extração de características.

```

// Por fim, define-se o decisor local agregando
// os algoritmos e o coletor de dados. Ao instanciar
// estas classes, cria-se objetos que representam os
// blocos do diagrama mostrado:
CLASS CaloEMLocalProcessor {
PUBLIC:
    CaloEMLocalProcessor(DataCollector<Cluster, CaloRoISpec>&,
                          FEx<Cluster, EMOBJECT>&,
                          Discriminator<Cluster>&);
    EMOBJECT* run (CONST CaloRoISpec*);
    ...
PRIVATE:
    DataCollector<Cluster, CaloRoISpec> _collector;
    MyEMFEx _fex;
    MyEMDiscriminator _discriminator;
    ...
};

// A Implementação de run() fica simples:
EMOBJECT* CaloEMLocalProcessor::run (const CaloRoISpec* spec) {
    EMData* data = this->_collector.collect(spec);

    IF ( this->_discriminator.confirm() )
        RETURN this->_fex.extract(data);
    ELSE
        RETURN 0;
}

// Durante o programa, é possível criar um decisor local
// e operá-lo!
    CaloFileDataCollector collector;
    MyEMDiscriminator discriminator;
    MyEMFEx fex;
    CaloEMLocalProcessor* local = NEW CaloEMLocalProcessor(
        collector, discriminator, fex);
    ...

```

Implementação 4.3: O decisor local (CaloEMLocalProcessor).

classes que serão realmente implementadas está mostrada.

Por final, define-se o decisor local (`CaloEMLocalProcessor`). A implementação da função `run()` é mostrada por último. Repare a simplicidade como é conduzida a requisição de dados, discriminação e extração de características. Por utilizar ponteiros para classes mais genéricas, para o decisor local não é importante saber que algoritmos está usando.

De fato, o decisor local não faz parte dos algoritmos de discriminação e extração de características no topo da pilha da Figura 4.1. É um dos sistemas de controle do L2. Os algoritmos **são** elementos do topo daquela pilha no entanto.

Levando-se em consideração que C é um sub-conjunto da linguagem C++, estes algoritmos podem ser implementados de forma estruturada ou orientada o objetos, como está colocado no apêndice C. Além disso, seu desenvolvimento pode ser feito de forma quase que independente do L2. Os únicos laços definidos são as interfaces com decisor local.

Capítulo 5

Decisores Locais Neurais

O Segundo Nível de filtragem do experimento ATLAS pode utilizar redes neurais artificiais (RNA's) em muitas de suas tarefas [20, 21]. Por exemplo, durante a fase de classificação de objetos locais. Dentre as características de RNA's que são importantes no L2, destaca-se as seguintes [22]:

Robustez Em ambientes extremamente *agressivos* (sujeitos a falhas e a radiotividade), como é o caso da operação do L2, RNA's podem manter um excelente desempenho, mesmo quando parte dos dados (canais de aquisição) de entrada ou neurônios são perdidos;

Generalização RNA's podem extrair a informação relevante escondida sob uma grande quantidade de ruído. Os objetos a serem identificados possuem, invariavelmente, esta característica;

Deteção de novos fenômenos RNA's podem detetar a ocorrência de novos objetos de forma bastante eficaz. Isto é extremamente importante em ambientes que podem revelar resultados inesperados (nova física).

Simples implementação Uma rede neuronal artificial pode ser implementada em *hardware* ou sistemas programáveis. Embora RNA's sejam mais rápidas se modeladas em *hardware* dedicado, esta opção é inevitavelmente cara e de pouca flexibilidade [23]. Usando-se processadores mais genéricos, é possível baratear os custos de produção, simplificar a implementação e manter a flexibilidade necessária nos sistemas de segundo nível de filtragem do ATLAS.

Nesse trabalho, propõe-se o uso de RNAS's enquanto discriminadores de objetos eletromagnéticos. Em particular, a discriminação $e^-, e^+/\text{jato}$ será analisada em detalhes. Os dados de entrada, no entanto, possuem uma dimensionalidade bastante alta. Propõe-se então, uma redução inteligente da dimensão de entrada, baseando-se na forma topológica com que as partículas depositam sua energia ao longo de sua trajetória nos calorímetros. Esse pré-processamento leva em conta os detalhes operacionais do ambiente de calorimetria e o conhecimento da física que deseja-se detetar.

5.1 O espaço de entrada para os discriminadores de objetos e.m.

Como anotado na Tabela 3.1, os objetos e.m. são analisados em uma região $0,4 \times 0,4$ no plano $\eta \times \phi$. Isto compreende uma área de 4×4 torres de filtragem para os calorímetros e.m. do ATLAS e de 2×2 torres de filtragem para os calorímetros hadrônicos.

Devido à granularidade variante de cada camada com η , e de camada para camada, nos calorímetros do ATLAS, o número de células para uma RoI do calorímetro pode variar bastante. Além disso, como mostram as Tabelas 2.1 e 2.2, há regiões nas quais algumas camadas não estão presentes. Por exemplo, o *presampler* somente existe até $\eta = 1,8$. Este tipo de situação deve ser tratada, de forma a se compor um vetor de entrada compatível com o sistema neuronal que atuará como discriminador. A seguir lista-se algumas soluções que podem ser dadas às diferenças de granularidade de RoI para RoI:

1. **Vários pré-processadores, 1 classificador** Dependendo do centro da RoI, um esquema de pré-processamento é adotado. O pré-processamento, independente da localização do centro da RoI, é responsável por apresentar, em sua saída, um número fixo de componentes que são encaminhadas, então, a um classificador neuronal. Nesse esquema, há somente um processador neuronal treinado para classificar elétrons e jatos a partir das componentes na saída do pré-processador;

2. **Vários pré-processadores, vários classificadores** Haverá, como no caso anterior, um esquema de pré-processamento para cada tipo de RoI nos calorímetros do ATLAS. As saídas destes pré-processadores não seguem um esquema fixo e, portanto, a melhor forma de classificar a sua saída é realizada com vários classificadores neuronais, treinados para a identificação de elétrons e jatos de forma a se adequar às saídas produzidas por cada tipo de pré-processamento.

A Figura 5.1 mostra o número médio de células por RoI, extraído de alguns arquivos de dados contendo elétrons e jatos, que interagem com os calorímetros do ATLAS. As barras de erro representam a falta de estatística ao longo de ϕ ¹. No eixo das abscissas é possível ver os valores de η , e no eixo das ordenadas, o número de células na seção e.m., hadrônica e total (somando-se o número de células nas duas seções).

O contorno dos gráficos é coerente, considerando-se as Tabelas 2.1 e 2.2. O número de células da RoI, na seção hadrônica, é a que mais varia, como é possível ver. Os picos, começando em $|\eta| = 0,8$ e se estendendo até $|\eta| = 1,0$, representam o início do barril estendido e o final do barril (TileCal). Na região central ($\eta = 0$), o número de células é $4 \times 4 \times 3 = 48$, pois há 3 camadas, cada uma com 16 células, como mostra a Tabela 2.2. O número de células nas laterais da figura cai vertiginosamente. O pico, situado aproximadamente em $|\eta| = 1,5$, representa o início da tampa, que também perde granularidade rapidamente com o aumento de $|\eta|$.

O número de células e.m. varia menos que o equivalente na seção hadrônica. Há uma constância no número de células até aproximadamente $|\eta| = 1,3$, quando a tampa começa, mostrando um pico que culmina em $|\eta| = 1,475$, ou seja, no início da terceira camada na tampa. Após $|\eta| = 1,5$ o número de células cai vertiginosamente com o aumento do módulo da pseudo-rapidez, até $|\eta| = 2,5$, quando a tampa acaba.

Existem algumas soluções para o problema de granularidade variante de RoI para RoI, como foi visto. Este trabalho, no entanto, limita o seu escopo na inves-

¹A curva foi traçada criando-se um histograma bi-dimensional ao longo das direções η e ϕ , e depositando em cada caixa (*bin*), o número médio de células de todas as RoI que tinham seu centro naquele canal. As diversas curvas ao longo de ϕ foram somadas e produziram uma média (linha cheia) e um desvio padrão (barras de erro) para cada um de seus pontos em η .

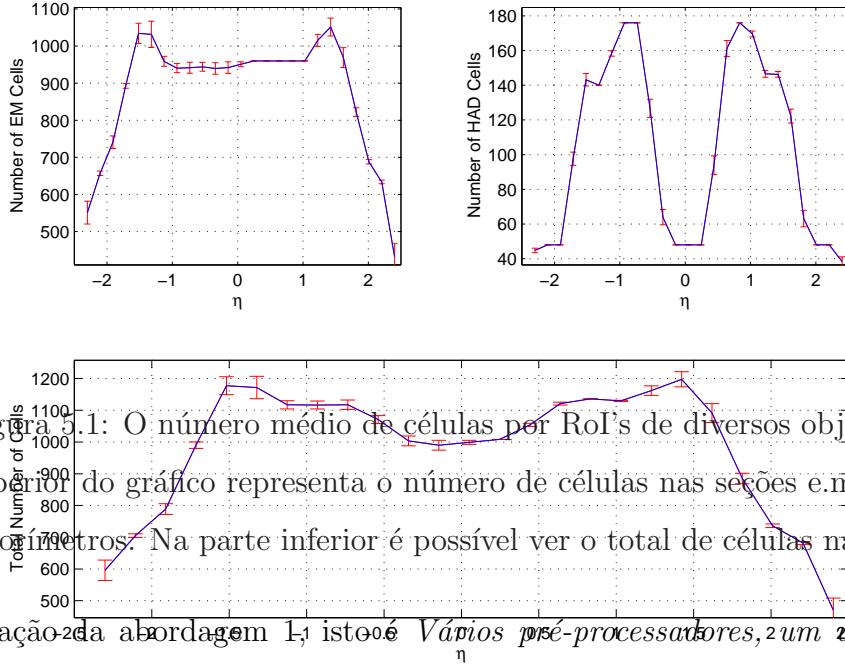


Figura 5.1: O número médio de células por RoI's de diversos objetos e.m.. A parte superior do gráfico representa o número de células nas seções e.m. e hadrônica dos calorímetros. Na parte inferior é possível ver o total de células nas duas seções.

Esta abordagem em 1, isto é, vários pré-processadores, 2 em 1, único classificador. Esta abordagem permite que se simplifique o estágio de discriminação nos processadores locais. Ademais, como será visto mais a frente, cortes de dimensionalidade nos dados de entrada podem tornar o pré-processamento genérico o suficiente para atender um grande número de possibilidades, sem perdas significativas no desempenho de classificação.

Inicialmente, trabalhar-se-á somente com aquelas RoI's cujo centro possui valores de $|\eta_{\text{centro}}| < 1,3$. Desta forma não se tem que lidar com problemas mais complicados de granularidade e é possível focalizar os estudos no método em si. Mais tarde será visto que é possível reduzir estas restrições facilmente, considerando-se a relevância dos dados de entrada. Ademais, para reduzir-se a dimensão de entrada, as células do *plug* conectado ao barril estendido são somadas às células correspondentes naquele detetor. As células de regiões onde há uma sobreposição entre o barril e sua extensão também são somadas. A Figura 5.2 mostra uma RoI de um elétron típico, interagindo com as diversas camadas do calorímetro. As somas propostas acima já foram efetuadas. Cada bloco representa a energia depositada naquela célula. Na

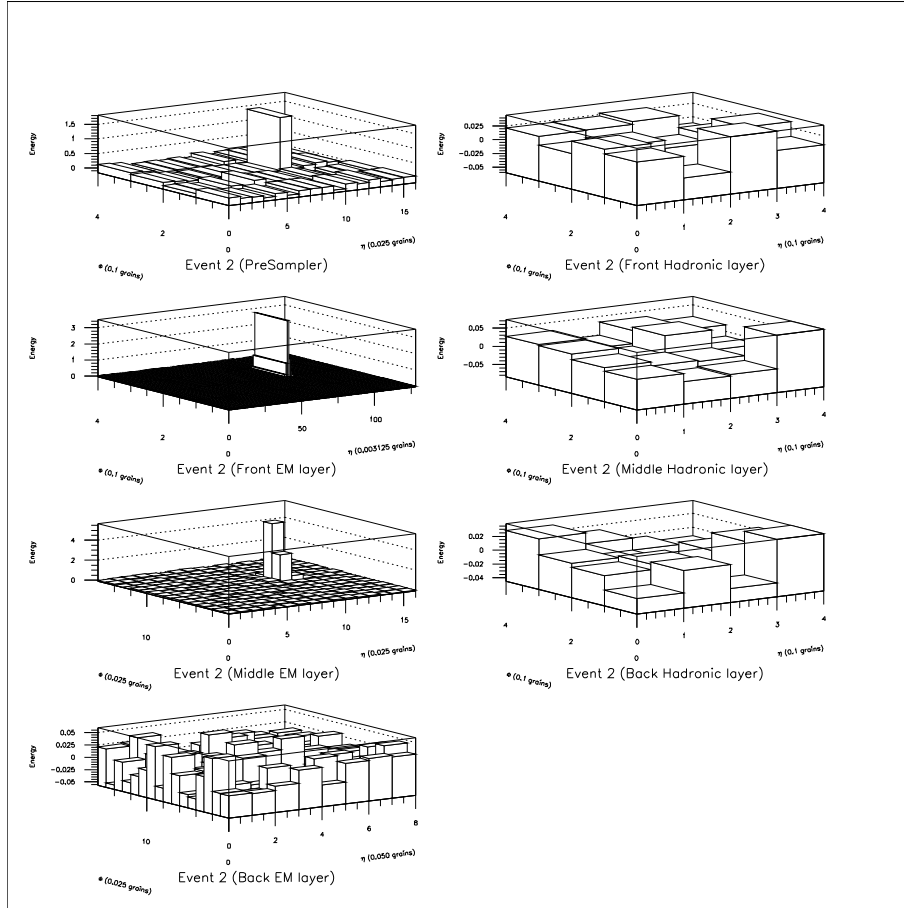


Figura 5.2: Uma RoI de um elétron típico interagindo com o calorímetro.

parte superior desta figura, é possível ver o *presampler* seguindo-se das 3 camadas eletromagnéticas e das 3 camadas hadrônicas. Como é possível ver, um elétron desenvolve sua cascata de forma bem concentrada, ao redor do ponto de impacto. Da terceira camada e.m. em diante, não há praticamente informação relevante, predominando o ruído, já que o objeto perde energia rapidamente devido à forma específica com que decai.

A Figura 5.3, por sua vez, mostra uma RoI típica de um jato. Repare que a deposição de energia no calorímetro hadrônico torna este objeto bastante óbvio para qualquer classificador. A Figura 5.4 mostra, no entanto, um jato bem mais difícil de ser corretamente classificado pois sua cascata possui um padrão de deposição que o faz muito parecido com um elétron. Devido à alta eficiência do primeiro nível de filtragem, os eventos (jatos) que falsearam o disparo do sistema para elétrons, tendem a ser predominantemente deste tipo, tornando a classificação correta dos

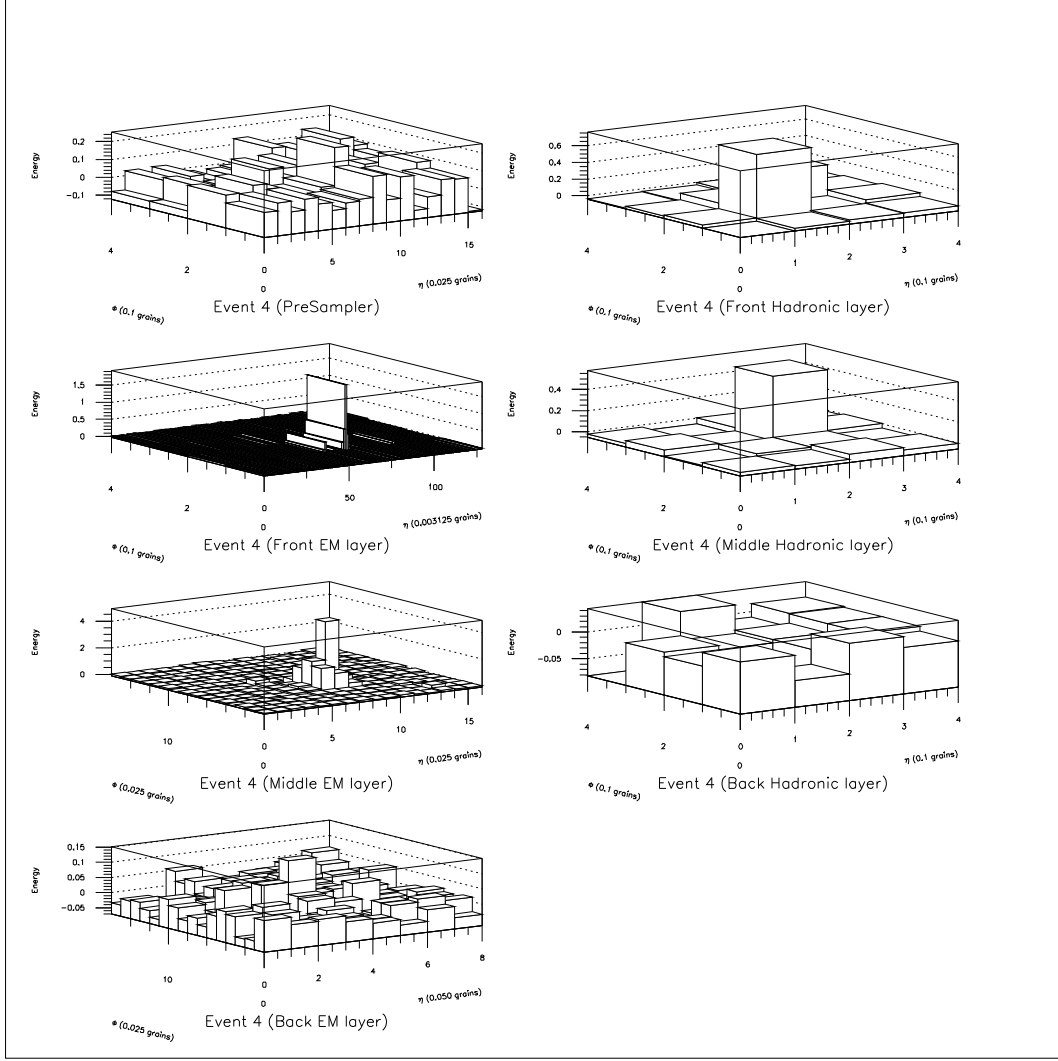


Figura 5.3: Uma RoI de um jato típico interagindo com o calorímetro.

eventos uma tarefa desafiadora. Somente um discriminador que explore muito bem as nuances no perfil de deposição desse tipo de objeto conseguirá classificá-lo corretamente.

5.1.0.1 A terceira camada e.m.

A 3^a camada e.m. possui uma granularidade que varia bastante com η . Para evitar maiores complicações durante a implementação do discriminador neuronal, eliminar-se-á a informação desta camada no processo. Espera-se que o sistema consiga, através de correlações de ordens mais elevadas, recuperar esta informação. Isto também representará uma redução significativa no tempo de processamento da RoI, já que os dados desta camada não serão movidos para o processador local ou

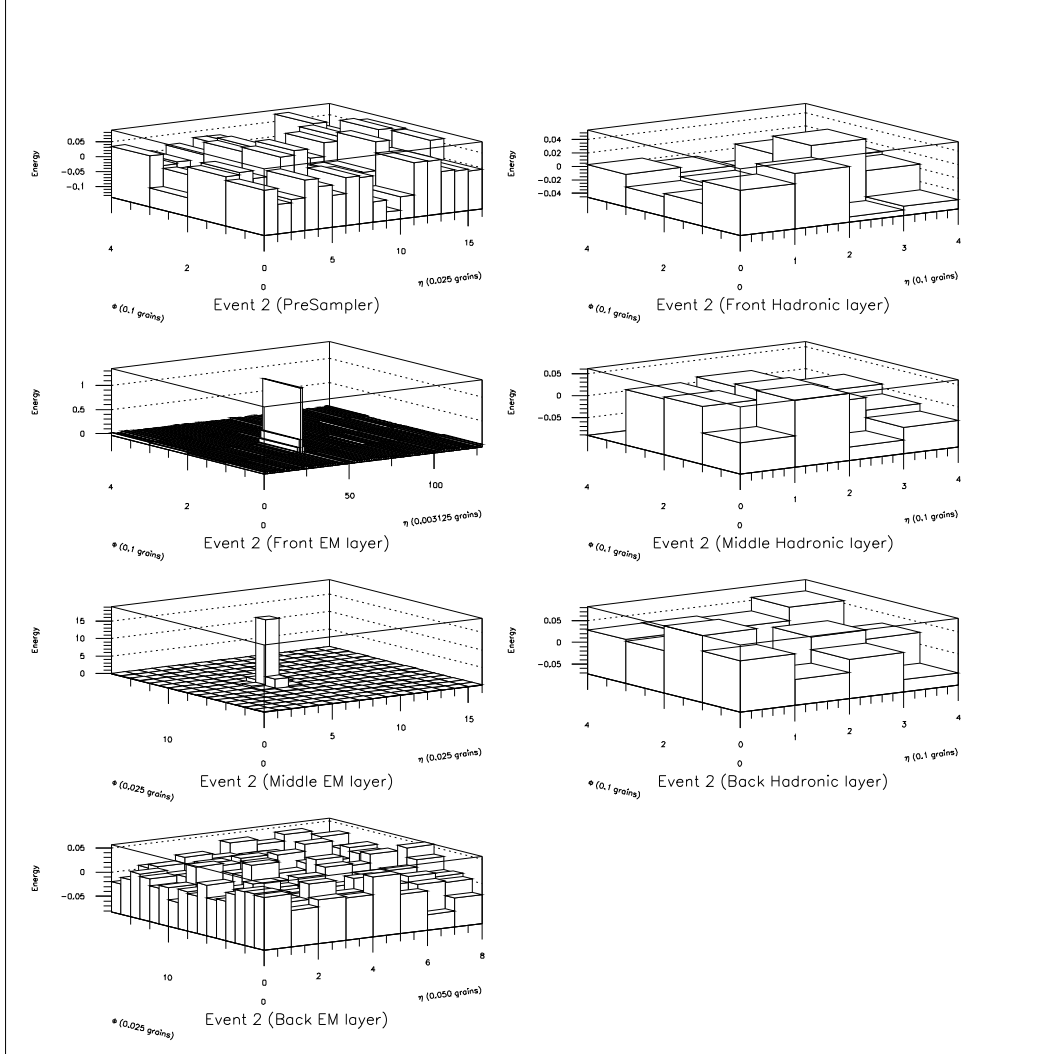


Figura 5.4: Uma RoI de um jato cujos padrões de deposição de energia fazem-no parecer um elétron interagindo com o calorímetro.

processados pelo sistema.

5.1.1 Os arquivos de entrada

A simulação dos sistemas de pré-processamento, discriminação e extração de características no L2, necessita de bancos de dados simulados com eventos completos do ATLAS. De cada evento, é possível extrair as regiões de interesse e, de cada região de interesse, os dados dos calorímetros.

A partir do Capítulo 4, conclui-se que a forma apropriada de utilizar dados é através da classe `DataCollector`. No entanto, porque este projeto foi inicialmente desenvolvido de forma estruturada, decidiu-se utilizar diretamente uma biblioteca de funções que podem consultar arquivos contendo eventos simulados do ATLAS. Atualizações do discriminador proposto deverão ser conduzidas de forma a embutir este enfoque de processamento no Software de Referência.

Os eventos se encontram organizados através de etiquetas que marcam seus diversos campos. A organização dos campos dos calorímetros encontra-se na Implementação 5.1. Os arquivos têm um formato parecido com uma codificação em linguagem C. A biblioteca `spec` pode ler arquivos com esta estrutura e retornar as células de todas as RoI's no arquivo, através de uma interface simples. Cada célula compõe-se do valor de energia depositado pela partícula em cada célula do sistema de calorimetria e seu número de identificação, o campo de bits.

Considerando todos os eventos disponíveis, dispõe-se de cerca de 600 RoIs de elétrons e cerca de 3600 de jatos. Isto pode apresentar um segundo problema: redes neurais tendem a se especializar, por falta de estatística suficiente, particularmente se o número de eventos de uma das classes é bem menor do que na outra. Este problema foi remediado, sorteando-se um evento de cada classe para cada vez que apresenta-se um padrão à rede, durante o treinamento. Em outras palavras, apresenta-se à rede um par de padrões a cada passo de treinamento. A seguir, discutem-se algumas das características dos dados disponíveis:

Energia Transversa Os elétrons têm energia transversa aproximadamente igual a 20 GeV. A energia transversa para o conjunto de jatos varia, pois a técnica de discriminação empregada no L1 não leva em consideração a energia depositada

```

/* Inicialmente define-se o que são células ou digs */
STRUCT emCalDigiType {
    INT id;
    FLOAT Et;
};

STRUCT hadCalDigiType {
    INT id;
    FLOAT Et;
};

/* Define-se então um campo que conterà todas as células */
TAG CALDIGI {
    INT nEmDigi;
    emCalDigiType emDigi[nEmDigi];
    INT nhadDigi;
    hadCalDigiType hadDigi[nhadDigi];
};

/* A RoI, dentre outros campos terá os dados do calorímetro */
TAG ROI {
    ROIHEAD header; /* O cabeçalho da RoI */
    CALDIGI calDigi; /* As nossas células */
    L2CALEM l2CalEm; /* Informação de depuração */

    ...
};

/* Os Eventos contêm qualquer número de RoIs */
TAG EVENT {
    EVHEAD header;
    KINE kine;
    INT nroi;
    ROI roi[nroi];
};

```

Implementação 5.1: Os campos de dados dos calorímetros.

nos calorímetros hadrônicos, onde este tipo de objeto decai, depositando a maior parte de sua energia.

Objetos e.m. com baixa energia (ou seja, na faixa de dezenas de GeV) são mais difíceis de serem classificados [11]. Por isto, testes com discriminadores são realizados prioritariamente com elementos nessa faixa energética.

Empilhamento Por causa da taxa elevada de eventos, o detetor ATLAS sofrerá efeitos de empilhamento (do inglês, *pile-up*). Empilhamento é o efeito causado nos detetores devido a sobreposição de eventos (um evento que ainda se desenvolve no detetor tem seu padrão de deposição de energia distorcido por um novo evento que chega e se sobrepõe ao evento). Ele gera um *pedestal*, ou seja, um ruído muito forte, aos eventos sendo analisados. Os dados disponíveis não têm a simulação deste efeito. Alguns estudos [24, 21], no entanto, sugerem que haja uma perda de aproximadamente 2% no desempenho de discriminação, ao adicionar o efeito pedestal aos eventos, quando se utiliza de processamento neuronal para as RoI's.

5.2 O Processamento clássico de objetos e.m.

A colaboração ATRIG (*ATLAS Trigger*) desenvolveu um algoritmo para realizar a separação elétron/jato nas condições do LHC. Este algoritmo é baseado no conhecimento especialista de altíssimo nível de físicos experimentais. Ele extrai quatro quantidades físicas, que são altamente discriminantes. Neste caso, a extração de características será ainda feita a parte. A este algoritmo dá-se o nome de *clássico*, pois não utiliza métodos otimizados de discriminação neuronal como será visto. Chamar-se-á estas quatro quantidades de *quantidades clássicas* no decorrer do texto.

As quatro variáveis extraídas para a discriminação se encontram descritas a seguir. Como é possível ver, são, na maior parte, derivadas de somas energéticas, o que é implementável de forma extremamente veloz.

$E_{T_{3 \times 7}}^{e.m.}$ Esta quantidade representa a soma das energias das células da seção e.m. dos calorímetros, cujo centro recai sobre uma região de $0,075 \times 0,175$ no plano

$\eta \times \phi$, ao redor do centro da célula da segunda camada e.m. que apresenta o maior valor de energia depositada. A janela é irregular por motivos da física de interesse [11]. A região na qual esta quantidade é extraída equivale, usando-se a granularidade da segunda camada e.m., a uma região de 3 células na direção η por 7 células na direção de ϕ . Espera-se que a energia de elétrons esteja totalmente contida nesta janela; por seu turno, jatos devem apresentar, hipoteticamente, uma fração menor de sua energia depositada nesta área;

$E_{T_{0,2 \times 0,2}}^{\text{HAD}}$ Esta quantidade representa a soma das células da seção hadrônica do calorímetro dentro de uma região de $0,2 \times 0,2$ no plano $\eta \times \phi$, ao redor do centro da célula na segunda camada e.m. que apresenta o maior valor de energia depositada. Para elétrons, a quantidade de energia depositada na camada hadrônica deve ser próxima de zero, enquanto que para jatos espera-se que seja bem elevada;

R_{η}^{shape} Representa a razão entre a soma das células na segunda camada e.m., numa área de 3×7 células por 7×7 células ao redor da célula na segunda camada que apresenta o maior valor de energia depositado. Esta quantidade mede o espalhamento da cascata formada pelo decaimento do objeto de estudo. No caso do objeto ser um jato, espera-se que a cascata tenha um espalhamento maior que no caso de elétrons, resultando em um valor menor que 1 para esta quantidade;

$R_{e.m.1}^{\text{strip}}$ Uma vez que jatos de partículas interagem de forma mais espalhada do que elétrons, espera-se que na primeira camada e.m. sejam observados vários picos. Definindo um pico (E) como sendo a ocorrência de um máximo registrado pela célula central de um agrupamento de três células adjacentes, então, a quantidade $R_{e.m.1}^{\text{strip}}$ representa a razão de energias $R_{e.m.1}^{\text{strip}} = \frac{E_1 - E_2}{E_1 + E_2}$. Ou seja, a razão da subtração pela soma dos valores de energia dos dois picos mais energéticos na primeira camada e.m..

Se o objeto for um elétron (objeto único), espera-se que $E_2 = 0$, já que a cascata de partículas que o elétron forma na sua interação com o calorímetro é, tipicamente, bastante estreita, e portanto $R_{e.m.1}^{\text{strip}} = 1$. Para jatos, normalmente, $R_{e.m.1}^{\text{strip}}$ será menor que 1, já que $E_2 \neq 0$.

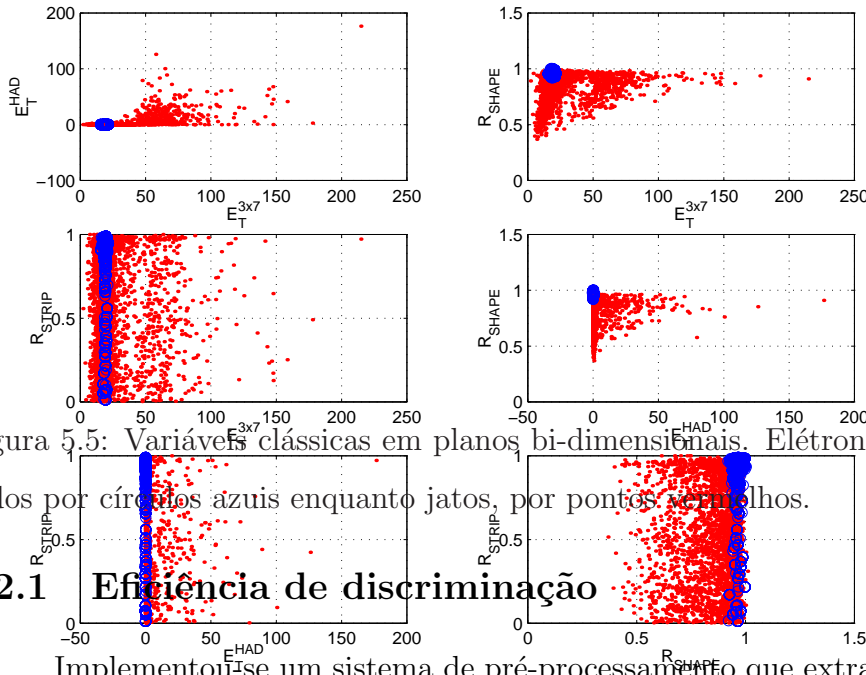


Figura 5.5: Variáveis clássicas em planos bi-dimensionais. Elétrons estão representados por círculos azuis enquanto jatos, por pontos vermelhos.

5.2.1 Eficiência de discriminação

Implementou-se um sistema de pré-processamento que extrai as quantidades descritas acima dos arquivos de dados. Nas Figuras 5.5 e 5.6, é possível ver projeções bi e tri-dimensionais destas 4 variáveis. É possível notar que algumas variáveis possuem uma variância pequena para elétrons e grande para jatos. A partir destas quantidades, desenvolveu-se alguns discriminadores, para avaliar o desempenho da formulação clássica na separação elétron/jato.

Os testes foram feitos utilizando-se cerca de 270 elétrons e 3600 jatos (separados em duas metades, uma para treino e outra para teste), que estavam disponíveis nos arquivos de dados². Os elétrons utilizados nestes testes são um sub-conjunto dos 600 elétrons que foram mencionados na Seção 5.1.1. Os jatos são exatamente os mesmos. A razão de ter se reduzido o conjunto de dados repousa no fato que o al-

²Os dados são obtidos através de simulações de Monte Carlo, utilizando a configuração atual de detetores e as técnicas de filtragem de eventos do ATLAS. Por esta razão, o número de elétrons gerados (provenientes de decaimentos de bósons de Higgs) é bastante menor que o número de jatos, já que a ocorrência de um Higgs é um processo raro.

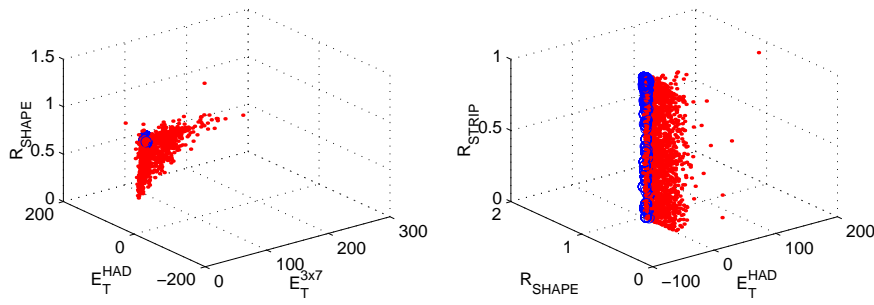


Figura 5.6: Variáveis clássicas em planos tri-dimensionais. Elétrons estão representados por círculos azuis enquanto jatos por pontos vermelhos.

goritmo necessita da maior granularidade disponível para maximizar sua eficiência de discriminação e os 330 elétrons restantes não possuem a informação da terceira camada bem definida.

O projeto dos discriminadores é conduzido utilizando-se metade dos dois conjuntos de dados, ou seja, 135 elétrons e 1800 jatos. O segundo conjunto de dados somente é usado para testar os discriminadores encontrados.

A seguir, se enumera e detalha alguns possíveis discriminadores baseados nestas quatro quantidades físicas.

5.2.2 Discriminador Linear

Nesta modalidade, calcula-se o hiper-plano no espaço das 4 variáveis que melhor discrimina elétrons e jatos, utilizando o algoritmo LMS (*Least Mean Squares*, ou método dos mínimos quadrados iterativo). Utilizando um neurônio linear, o mínimo global está bem-definido no espaço proposto pelas 4 variáveis e é alcançado facilmente com a escolha correta da taxa de aprendizado [25, 22].

Esta solução não explora bem situações nas quais o domínio das variáveis de

uma das classes está dentro do domínio das variáveis da outra classe (como é possível ver nos gráficos das Figuras 5.5 e 5.6). Dois ou mais patamares de separação por dimensão, neste caso, seriam mais apropriados. Por esta razão, este discriminador não produz resultados tão bons como na separação através da análise combinatória das variáveis duas a duas, conforme é feito pelos físicos especialistas, ou separadores neurais, que podem esboçar recortes mais sofisticados no subespaço formado por estas quatro variáveis.

O discriminador, em si, consiste apenas de um neurônio, cuja função de ativação é a identidade (neurônio *linear*). Os valores iniciais de pesos para a entrada e para a polarização (*bias*) são escolhidos aleatoriamente, durante a inicialização do treinamento. Os alvos (t_i) para as classes de elétrons e jatos são, respectivamente, $+1$ e -1 . O sistema é treinado em bateladas de 100 eventos (usando-se o algoritmo LMS), até que o erro médio quadrático das saídas (o_i) da rede, calculado sobre **todos** os (N) vetores de entrada disponíveis (EMQ, Equação 5.1), se estabilize. Para isto, o conjunto de treinamento é dividido nas classes elétron e jato e 50 eventos de cada classe, escolhidos aleatoriamente, são apresentados à rede para que se dê um passo de treinamento.

$$\text{EMQ} = \frac{1}{N} \sum_{i=1}^N (t_i - o_i)^2 \quad (5.1)$$

Após 30000 passos de treinamento, o erro médio quadrático para o conjunto de treinamento estabilizou-se em 0,63 e para o conjunto de teste em 0,67. O resultado se repetiu para diferentes inicializações (o teste foi realizado 3 vezes). A Figura 5.7 mostra a evolução do erro médio quadrático para os conjuntos de teste e treinamento. Nessa figura, é possível ver que o sistema converge de forma suave para o valor de EMQ final.

À saída do discriminador linear (agora treinado), aplica-se um patamar de separação (*threshold*). Objetos cujas saídas estão à esquerda do patamar, são considerados como jatos, todos os outros, como elétrons. Cada patamar que é proposto implica num valor de eficiência de discriminação para ambas as classes que estão sendo separadas, formando a curva característica do detetor (ROC, do inglês, *Receiver Operation Characteristics*) [26]. Patamares mais à esquerda (aproximando-se de -1) privilegiarão elétrons, aumentando a probabilidade de detecção do sinal de

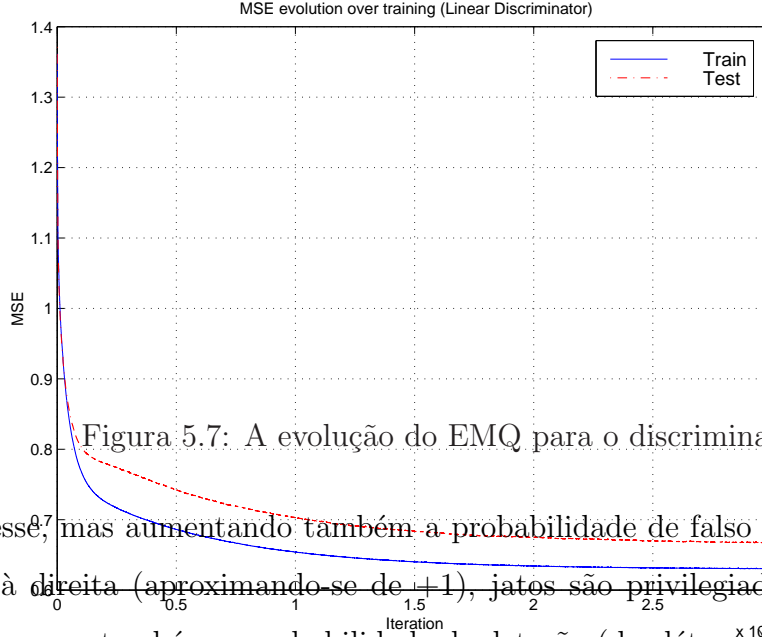


Figura 5.7: A evolução do EMQ para o discriminador linear

interesse, mas aumentando também a probabilidade de falso alarme, enquanto que mais à direita (aproximando-se de $+1$), jatos são privilegiados, reduzindo o falso alarme, mas também a probabilidade de deteção (de elétrons).

É possível determinar um conjunto de coordenadas que representem a eficiência de discriminação de elétrons e jatos, à medida que desloca-se o patamar de separação, da esquerda para a direita. Analogamente, à este conjunto de coordenadas, existe outro que representa a eficiência de discriminação em elétrons contra o erro na classificação de jatos. A Figura 5.8 mostra a curva de eficiência obtida considerando-se os diferentes patamares de separação. Neste caso, os valores no eixo das abscissas foram multiplicados por 25 (kHz)³, pois há, de fato, interesse na eficiência de classificação de elétrons, contra a taxa de falso alarme real (i.e., do sistema de filtragem) em jatos, ou seja, o número de jatos erroneamente classificados como elétrons (vide equação 5.2). Portanto, por exemplo, para 86% de eficiência em elétrons, a eficiência para jatos é 28,2%.

$$\text{Falso Alarme Real} = 25\text{kHz} \times (1 - \text{Eficiência para jatos}) \quad (5.2)$$

³Este número é extraído de [27]. Ele representa a taxa de objetos que são erroneamente classificados como elétrons pelo L1 e, portanto, a taxa de elétrons “falsos” que o L2 receberá.

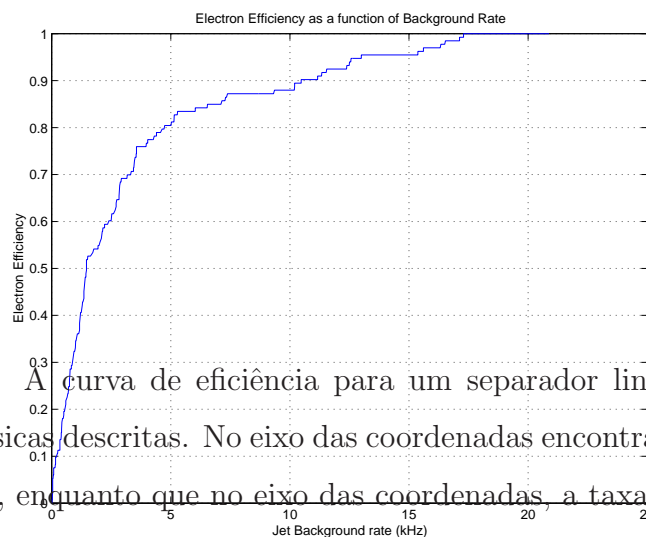


Figura 5.8: A curva de eficiência para um separador linear ótimo aplicado às 4 variáveis físicas descritas. No eixo das coordenadas encontra-se a eficiência no acerto de elétrons, enquanto que no eixo das coordenadas, a taxa de ruído (jatos erroneamente classificados), em kHz.

5.2.3 Discriminador por Análise Combinatória

Neste caso, estudam-se as correlações entre as variáveis duas a duas, determinando um separador linear por plano de análise. O processo continua até que todas as permutações tenham sido analisadas. É o método atualmente utilizado no CERN e classicamente usado pelos físicos, como forma de superar as dificuldades em se analisar o comportamento das variáveis de discriminação em espaços de dimensionalidade elevada, onde a visualização completa do comportamento discriminante dessas variáveis não é possível. Espera-se que o resultado aqui seja um pouco melhor que no caso anterior, pois este enfoque explora o conhecimento da física do evento. Ademais, esta técnica permite que um conjunto de recortes mais sofisticados seja aplicado, de forma a classificar melhor as duas classes de objetos.

É claro que a ordem na qual averiguam-se os 6 espaços bi-dimensionais é extremamente relevante para um desempenho satisfatório do discriminador, já que não é possível anteceder com acurácia que alguns eventos poderão ser separados de forma mais eficaz pelos próximos níveis de separação. O projetista deve ter um bom conhecimento das variáveis e suas distribuições para operar desta forma. Para a referência do leitor, realizou-se o exercício de discriminação analisando as

Tabela 5.1: Configurações para um possível discriminador por análise combinatória das variáveis duas a duas (*clássico*).

Etapa	Configuração 1	Configuração 2
1	$E_{T_{3 \times 7}}^{e.m.} \times E_{T_{0,2 \times 0,2}}^{HAD}$	$R_{e.m.1}^{strip} \times R_{\eta}^{shape}$
2	$E_{T_{3 \times 7}}^{e.m.} \times R_{\eta}^{shape}$	$R_{e.m.1}^{strip} \times E_{T_{0,2 \times 0,2}}^{HAD}$
3	$E_{T_{3 \times 7}}^{e.m.} \times R_{e.m.1}^{strip}$	$R_{e.m.1}^{strip} \times E_{T_{3 \times 7}}^{e.m.}$
4	$E_{T_{0,2 \times 0,2}}^{HAD} \times R_{\eta}^{shape}$	$R_{\eta}^{shape} \times E_{T_{0,2 \times 0,2}}^{HAD}$
5	$E_{T_{0,2 \times 0,2}}^{HAD} \times R_{e.m.1}^{strip}$	$R_{\eta}^{shape} \times E_{T_{3 \times 7}}^{e.m.}$
6	$R_{\eta}^{shape} \times R_{e.m.1}^{strip}$	$E_{T_{0,2 \times 0,2}}^{HAD} \times E_{T_{3 \times 7}}^{e.m.}$

combinações de variáveis duas a duas segundo à ordem descrita na Tabela 5.1. A primeira dessas configurações é a ordem mais intuitiva de separação de objetos e.m., baseando-se nas descrições das variáveis e sua relevância em [27, 11]. A segunda configuração é a primeira com sua ordem invertida e com a 3^a e 4^a etapas trocadas.

Para construir os 6 separadores lineares, utiliza-se do sistema mostrado na figura 5.9. Na parte inferior dessa figura é possível averiguar que o sistema de filtragem nesse caso estará configurado em cascata. Cada etapa de filtragem eliminará mais jatos, tentando reter o maior número de elétrons possível. A parte superior da figura mostra o funcionamento de cada bloco da cascata. Esses blocos são compostos de um sistema entrada gráfica e um algoritmo de treinamento usando LMS.

O projeto de cada bloco inicia-se escolhendo a polarização e os pesos através de um sistema gráfico ou de forma aleatoriamente. Os eventos de entrada são divididos em conjuntos de treinamento e teste e são normalizados⁴. Os valores iniciais de peso e polarização então, são utilizados para a inicialização do algoritmo de treinamento, que calcula o plano separador com o menor EMQ possível usando como entrada para rede as duas variáveis escolhidas para a etapa de treinamento atual. O treinamento é parado quando os valores de EMQ para os conjuntos de teste e treino atingem seus mínimos ou quando o EMQ se mostra com uma variação muito me-

⁴As variáveis $E_{T_{3 \times 7}}^{e.m.}$ e $E_{T_{0,2 \times 0,2}}^{HAD}$ tiveram as médias de classe removidas e foram divididas pelo desvio padrão da respectiva distribuição. As demais ($R_{e.m.1}^{strip}$ e R_{η}^{shape}) não sofreram alterações, pois se encontravam contidas no intervalo entre 0 e 1, não sendo necessário uma normalização adicional.

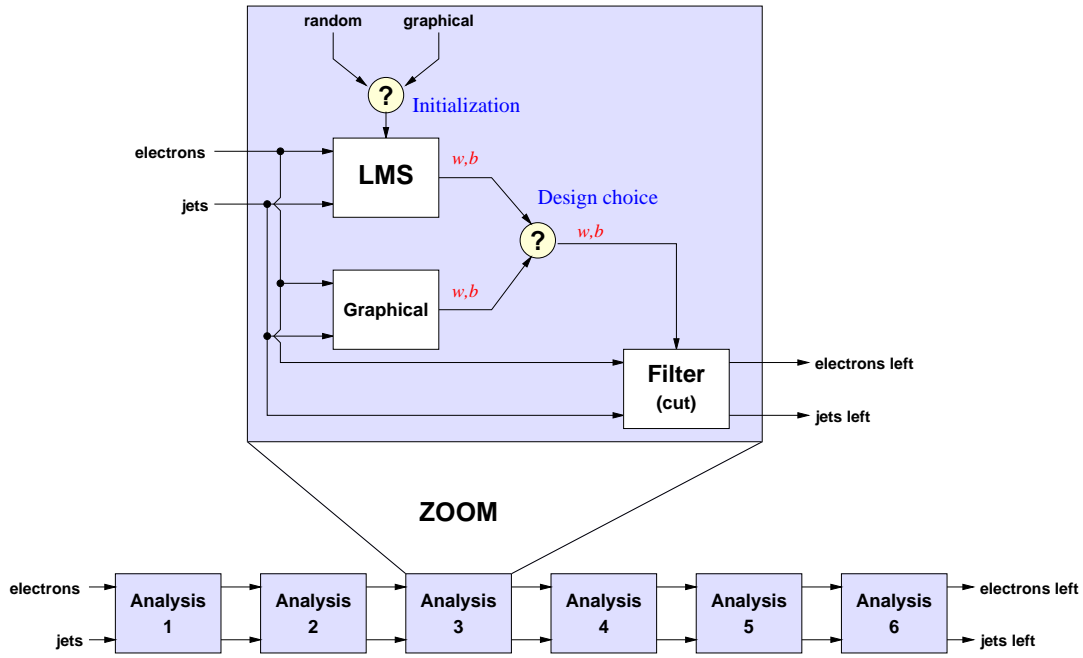


Figura 5.9: O esquema que determina o separador por análise combinatória das 4 variáveis clássicas quando agrupadas em pares (veja texto).

nor (10000 vezes) que sua amplitude entre dois passos de treinamento consecutivos. Utilizou-se o treinamento em bateladas de 10 eventos.

Após minimizar o EMQ, o sistema desenhará no plano escolhido os vetores relativos a cada objeto sob análise e confirma para o projetista o plano de separação. Nesse momento, o projetista poderá alterar o plano graficamente, se assim desejar, invalidando a minimização pelo EMQ. Independente do escolhido, um separador será utilizado por um filtro linear, já na saída do bloco, para discriminar os objetos na entrada do sistema. À saída do filtro, um patamar de separação em 0 (zero) é aplicado: eventos cujas saídas são maiores ou iguais a zero são resguardados, enquanto que os demais são descartados. Os eventos remanescentes são utilizados para um processo análogo de projeto da próxima etapa de filtragem.

5.2.3.1 Resultados

A curva de eficiência deste separador é difícil de ser traçada, já que deve-se sintonizar 6 retas ao invés de um único plano no caso linear! Repare que um esforço extenuante poderia esboçar alguns pontos no plano eficiência *versus* falso alarme; pontos esses que representam projeções da superfície no hiper-plano de

dimensão 6 que realmente define a curva de eficiência deste discriminador. Já que este trabalho foge do escopo deste projeto, limita-se a descrever as eficiências de separação elétron/jato para os dois casos descritos, com o patamar de separação afixado em 0 para todos os 6 discriminadores lineares:

1. Configuração 1 - Conseguiu-se, ao final, que 95% dos elétrons fossem corretamente classificados enquanto que apenas 2,91 kHz de jatos fossem admitidos (88,4% de eficiência na detecção de jatos);
2. Configuração 2 - Neste caso termina-se o processo com 93% de eficiência de elétrons contra apenas 3,58 kHz (85,7% de eficiência) de taxa de jatos (jatos).

Para o caso linear, a mesma eficiência de detecção de elétrons implica em uma taxa de jatos de 12,9 kHz (48,3% de eficiência) e 12,2 kHz (51,3% de eficiência) respectivamente para as configurações 1 e 2. Esses resultados são obtidos a partir da curva na Figura 5.8, e confirmam a melhoria obtida com os novos cortes.

5.2.4 Discriminador Neuronal

Neste caso, treinou-se uma rede neuronal sem realimentação e totalmente conectada, com 4 neurônios não-lineares (com função de ativação tipo *tangente hiperbólica*) na camada intermediária, para que se faça a separação entre elétrons e jatos. Os dados de entrada foram normalizados da mesma forma que na seção anterior, de tal forma que não mais que 10% dos eventos se encontrassem fora do intervalo $-1, +1$.

A rede foi treinada em bateladas, usando-se o método de retro-propagação dos erros, com taxa de aprendizado variável e momento [22, 28] usando-se o pacote JETNET, versão 3.0 [29]. As equações 5.3 e 5.4 descrevem a utilização do momento (μ) e do decaimento no aprendizado (γ) da rede, respectivamente. Nestas equações E representa o erro médio quadrático (EMQ) e a matriz \mathbf{w} o valor dos pesos da rede [29]. Na equação 5.3 é possível ver que valores muito altos de momento (próximos a unidade) podem trazer instabilidade ao sistema, já que o valor efetivo da atualização dos pesos sinápticos torna-se muito alto [28].

$$\Delta \mathbf{w}_{t+1} = -\mu \frac{\partial E}{\partial \mathbf{w}_{t+1}} + \alpha \cdot \Delta \mathbf{w}_t \quad (5.3)$$

$$\mu_{t+1} = \begin{cases} \mu_t \cdot \gamma & \text{se } E_{t+1} > E_t \\ \mu_t \cdot \left(1 + \frac{1-\gamma}{10}\right) & \text{caso contrário} \end{cases} \quad (5.4)$$

5.2.4.1 Resultados

Um total de 23 redes foram treinadas com parâmetros de taxa de aprendizado (μ), momento (α) e decaimento (γ) diferentes. A inicialização dos pesos foi feita de forma aleatória para cada teste, escolhendo-se um número real entre -1 e $+1$. O critério de parada utilizado é o produto SP, que será explanado em detalhes mais a frente (Seção 5.4.1). Basicamente, o produto SP é a soma das eficiências de discriminação de elétrons e jatos multiplicado pelo produto das mesmas. Utilizar o erro médio quadrático para este fim, neste caso, não é conveniente, pois se está interessado na eficiência de discriminação da rede [22].

A Tabela 5.2 mostra os resultados obtidos após o treinamento dessas várias redes. A primeira coluna mostra o número do teste. As colunas seguintes indicam, respectivamente, a taxa de aprendizado, o momento, a taxa de decaimento da taxa de aprendizado, o valor do produto SP, o número total de passos de treinamento, o tamanho da batelada, o número da iteração onde o melhor produto SP foi obtido, o erro médio quadrático ao final do treinamento, o patamar de separação entre as classes (que garante o produto SP ótimo apresentado), as eficiências de discriminação para elétrons e jatos naquele patamar e o ruído de fundo (jatos) resultante do erro de discriminação de jatos. Os resultados (valores de eficiência e EMQ) são representativos do conjunto de teste, ou seja, aquele que **não** foi utilizado para treinar a rede neuronal. O tamanho da época para todos os casos foi mantido fixo em 100 (passos de treinamento).

Durante esses testes, verificaram-se as seguintes propriedades do sistema:

- Tamanho da batelada e taxa de aprendizado - Utilizando uma taxa de aprendizado alta, a rede tende a obter resultados melhores (maior produto SP). Como consequência, há maior oscilação nos valores de EMQ durante o treinamento, o que pode ser reduzido optando-se por um treinamento mais longo usando uma taxa de aprendizado bem menor (em torno de 0,01) ou aumentando o tamanho da batelada e inserindo um amortecimento à taxa de aprendizado (γ).

Utilizou-se bateladas com diferentes tamanhos (100, 400, 600, 800 e 1000). Dentre estas opções, a rede que apresentou convergência mais suave, tinha sua batelada igual a 1000;

- Suscetibilidade ao momento - Ao adicionar à atualização dos pesos a informação do passo anterior, através do momento (μ), o sistema apresenta um forte comportamento oscilatório. Valores de momento acima de 0,9 são suficientes para que o fenômeno ocorra, o que já era esperado [28];
- Produto SP e o EMQ - Analisando a Tabela 5.2, é possível ver que o EMQ não tem, de fato, total correlação com o valor do produto SP. Na grande maioria dos testes, o valor do produto SP foi acima de 1,5, chegando a um valor máximo de 1,57 em alguns casos. Já o valor do EMQ, para os mesmos testes, variou bastante, havendo casos onde dobrava, embora o desempenho de discriminação da rede permanecesse constante;
- Patamar de separação - O patamar de separação⁵ está, na grande maioria das vezes, à direita do zero, indicando que as redes treinadas conseguem identificar bem elétrons, mas não são tão eficazes na identificação de jatos. Este resultado era esperado, pois os jatos que chegam ao segundo nível são de difícil classificação e apresentam uma flutuação estatística maior que o sinal de elétrons, bem mais estável. O fato do patamar estar muitas vezes à direita do zero também pode indicar que a classe de elétrons tem um EMQ menor que a de jatos;

A Figura 5.10 compara as curvas características do teste 3 com a do discriminador linear. Este teste foi escolhido por possuir alta eficiência de discriminação (com patamar de separação próximo à zero) e um baixo erro médio quadrático, se comparado aos outros classificadores.

No caso do separador que utiliza planos bi-dimensionais (a opção atual no CERN), os valores dos produtos SP para os testes 1 e 2 descritos anteriormente é respectivamente 1,53 e 1,43. Estes resultados são piores que a maioria dos resultados

⁵O melhor patamar de separação representa o ponto no intervalo $[-1, +1]$ que gera o maior produto SP.

Tabela 5.2: Os resultados do treinamento de 23 redes neurais para separação elétron/jato utilizando as quatro variáveis clássicas.

Teste	μ	α	γ	SP	# Passos	# Batelada	Iteração	EMQ	Patamar	% e ⁻	% jatos	Falso alarme (kHz)
1	0,15	0,5	1	1,55	1000	100	954	0,368	0,243	95	89,07	2,73
2	0,15	0,7	1	1,57	1000	100	900	0,572	0,715	94	90,45	2,39
3	0,15	0,9	1	1,57	1000	100	757	0,311	0,039	96	88,96	2,76
4	0,15	0,99	1	1,54	1000	100	149	0,210	-0,696	96	88,02	2,99
5	0,05	0,9	1	1,57	1000	100	726	0,452	0,550	93	91,39	2,15
6	0,02	0,9	1	1,54	1000	100	953	0,419	0,281	95	88,63	2,84
7	0,02	0,95	1	1,57	1500	100	1366	0,287	0,148	93	91,11	2,22
8	0,02	0,995	1	1,55	1500	100	348	0,411	0,559	93	90,34	2,41
9	0,01	0,95	1	1,56	2000	100	1845	0,300	0,133	94	90,07	2,48
10	0,09	0,75	1	1,56	1000	400	706	0,344	0,125	96	87,80	3,05
11	0,07	0,75	1	1,55	1000	400	925	0,319	0,069	96	87,75	3,06
12	0,1	0,75	1	1,55	1000	600	616	0,345	0,128	96	87,75	3,06
13	0,1	0,75	1	1,56	1000	1000	573	0,330	0,067	96	87,80	3,05
14	0,1	0	1	1,47	1000	1000	791	0,493	0,195	94	86,53	3,37
15	0,15	0	1	1,52	1000	1000	787	0,290	0,418	92	90,73	2,32
16	0,15	0	1	1,53	1000	800	886	0,378	0,466	93	90,34	2,41
17	0,15	0	0,95	1,53	1000	800	941	0,548	0,491	93	90,34	2,41
18	0,15	0	0,9	1,53	1000	800	946	0,578	0,528	93	90,45	2,39
19	0,1	0	0,7	1,51	1500	600	999	0,412	0,262	93	89,79	2,55
20	0,15	0,75	0,7	1,56	1300	600	757	0,250	-0,083	95	89,24	2,69
21	0,15	0,9	0,7	1,56	1300	600	360	0,301	0,098	95	89,29	2,68
22	0,15	0,5	0,6	1,55	1300	600	1179	0,363	0,164	96	87,69	3,08
23	0,05	0,75	0,7	1,55	2500	800	2190	0,283	0,115	93	90,62	2,35

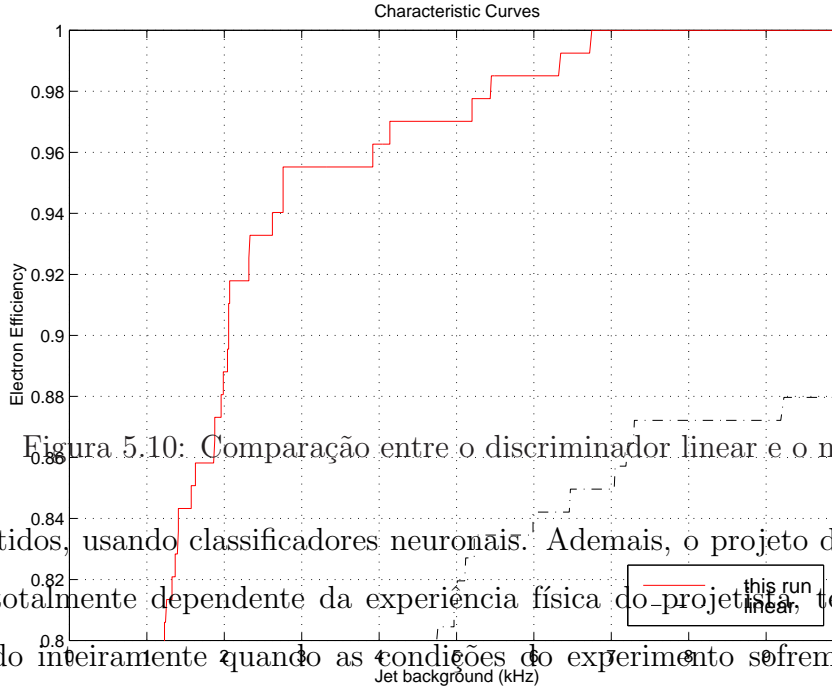


Figura 5.10: Comparação entre o discriminador linear e o neural (teste 3).

obtidos, usando classificadores neurais. Ademais, o projeto de tal discriminador é totalmente dependente da experiência física do projetista, tendo que ser adaptado inteiramente quando as condições do experimento sofrem modificações. No caso de ocorrerem tais modificações, o procedimento neuronal, entretanto, pode ser atualizado a partir de novo treinamento, tornando este enfoque mais flexível.

Na Figura 5.11 apresenta-se a evolução do EMQ para o treinamento da rede do teste 3. Como é possível ver, a curva do erro para o conjunto de teste acompanha a do conjunto de treinamento.

5.3 Compactação dos dados de entrada usando anéis

No processamento clássico, as cerca de 1000 células da RoI na entrada do discriminador são reduzidas a quatro quantidades, facilmente processáveis por métodos clássicos ou neurais como foi visto. No entanto, querendo-se processar em tempo real estas células, utilizando toda a granularidade disponível, há um problema: a dimensionalidade do espaço de entrada torna proibitivo qualquer método de dis-

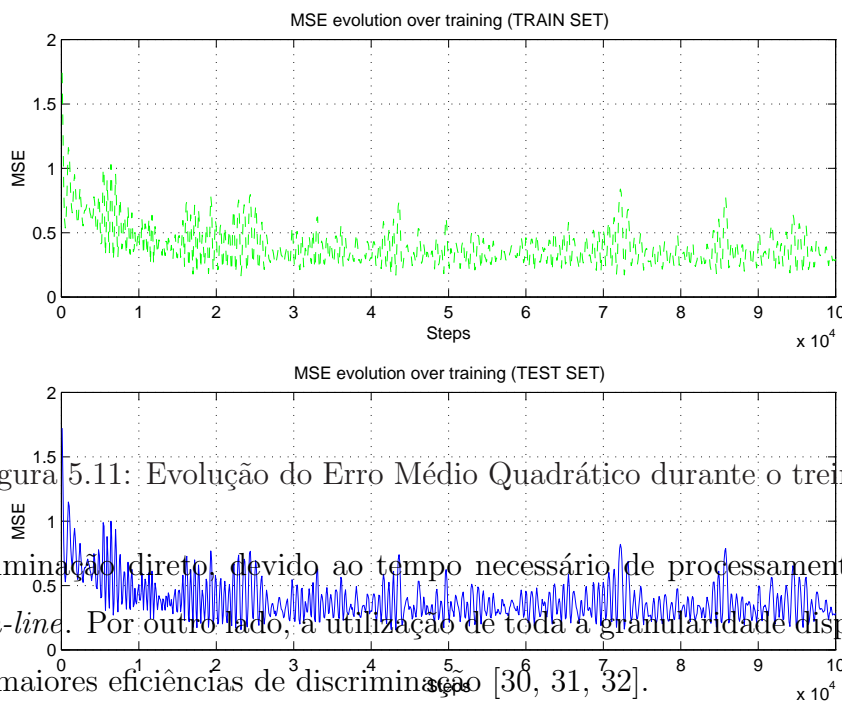


Figura 5.11: Evolução do Erro Médio Quadrático durante o treinamento da rede 3. criminação direto, devido ao tempo necessário de processamento para a operação *on-line*. Por outro lado, a utilização de toda a granularidade disponível, poder levar a maiores eficiências de discriminação [30, 31, 32].

A solução para esta dificuldade pode estar numa compactação eficiente dos dados de entrada. Neste trabalho, propõe-se uma versão modificada de compactação em anéis descrita em [33], que será adaptada às condições de granularidade variante da configuração atual de calorímetros do ATLAS.

A compactação em anéis [21, 33] aproveita-se do isotropismo na deposição de energia por parte dos objetos que interagem com o calorímetro. A tendência é que as partículas desenvolvam suas cascatas rumando para o exterior dos calorímetros, com raio crescente, de forma semelhante a um cone. A Figura 5.12 exemplifica o que idealmente ocorreria. As setas ilustram o decaimento do objeto em objetos menos energéticos, enquanto este interage com o detetor. Num detetor segmentado, é possível pressupor que este fenômeno será observável a cada camada, representando ao cone, uma de suas seções transversais.

Já que os calorímetros são divididos em camadas com granularidade variante, é natural propor que cada camada tenha seu conjunto de anéis. Assim, pode-

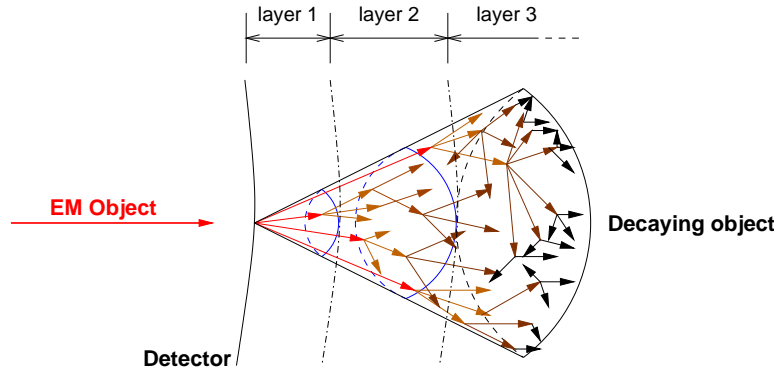


Figura 5.12: Idealização da deposição de energia.

se explorar eficientemente a informação detalhada oferecida pelos calorímetros do ATLAS.

Os anéis de cada camada são extraídos localizando-se a célula que registrou a maior deposição de energia na camada e somando-se o valor das células concêntricas a este ponto. Cada conjunto de células vizinhas⁶ é um anel. A Figura 5.13 ilustra como extrair os anéis de algumas camadas dos calorímetros do ATLAS. Após identificar a célula com maior deposição de energia, procura-se as células vizinhas que a circundam, formando um anel *quadrado*. Prossegue-se com este algoritmo, buscando o anel seguinte, até que se esgotem todas as células na camada. Durante este processo, é possível definirem-se anéis incompletos. Em casos extremos, como no *Presampler* e na terceira camada do calorímetro e.m., os anéis nem mesmo são conectados. Ainda assim, os valores destes anéis incompletos são somados para dar lugar a uma única quantidade. Por exemplo, na terceira camada do calorímetro e.m. (na parte esquerda inferior da Figura 5.13), marcou-se as células que somadas produzirão as quantidades de interesse. Cada quantidade representará um anel para o discriminador. A extração de anéis é uma atividade que deverá ser adicionada ao sistema de pré-processamento.

Construiu-se o pré-processador para os dados de simulação descritos anteriormente. Após processadas, as cerca de 1000 células iniciais são reduzidas a 58 somas em anel, como organizadas na Tabela 5.3.

O leitor atento notará que o número de anéis em algumas camadas, por exemplo no Pré-irradiador, é menor que o necessário para extrair todas as quantidades,

⁶Chama-se vizinha, a célula que se encontra adjacente à parte externa de um anel.

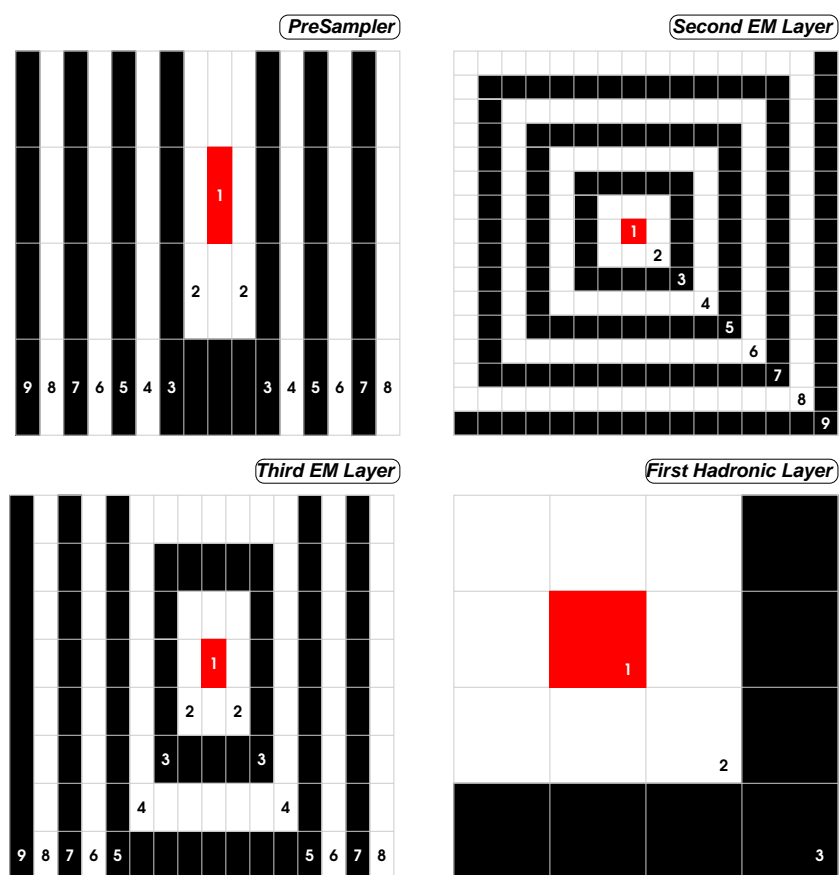


Figura 5.13: A extração de anéis em uma RoI nos calorímetros do ATLAS.

Tabela 5.3: O número de anéis por camada do calorímetro.

Camada	Número de Anéis
Pré-irradiador	6
1 ^a camada e.m.	33
2 ^a camada e.m.	10
1 ^a camada Hadrônica	3
2 ^a camada Hadrônica	3
3 ^a camada Hadrônica	3

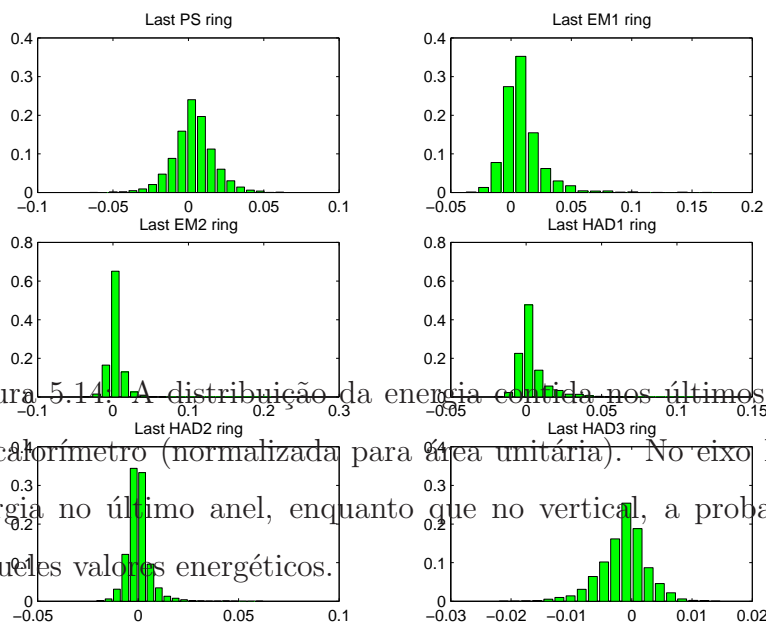


Figura 5.14 – A distribuição da energia contida nos últimos anéis de cada camada do calorímetro (normalizada para área unitária). No eixo horizontal observa-se a energia no último anel, enquanto que no vertical, a probabilidade de ocorrência daqueles valores energéticos.

ou anéis, possíveis daquela camada. No entanto, a quantidade de ruído na borda é tipicamente grande, como é possível ver na Figura 5.14. Essa figura mostra o histograma (normalizado para ter área unitária) dos últimos anéis de cada camada. No eixo das abcissas é possível ver o valor energético normalizado usando toda a energia da RoI, enquanto que no eixo das coordenadas é possível ver a probabilidade de se ter aquele valor energético no último anel da camada.

Desta forma, para se aproveitar a possível informação existente nestes anéis, soma-se os valores dos últimos anéis formando uma quantidade única, aumentando, possivelmente, a relação sinal/ruído nesta região.

5.3.1 Normalização

A energia dos objetos que devem ser discriminados pode variar bastante [11]. As células do calorímetro respondem a esta variação influenciando no valor de cada uma das somas (dos anéis) extraídas. Por outro lado, classificadores neuronais são muito sensíveis a variações bruscas da amplitude de suas entradas, podendo

eventualmente vir a saturar, perdendo desempenho.

Para evitar este efeito, é possível normalizar os dados de cada RoI, tendo por base a energia (transversa) do objeto sendo discriminado. Realizando-a, se está, artificialmente, eliminando as diferenças energéticas entre os diversos objetos e.m. que serão processados, permitindo que o discriminador concentre-se nas diferenças nos padrões de deposição energética. Esta forma de normalização permite se obter um discriminador único para objetos com diferentes níveis energéticos, embora a resolução em energia no calorímetro melhore com o aumento da energia do objeto, como foi visto no Capítulo 2, e, portanto, espera-se que a qualidade de discriminação melhore com a energia. Neste trabalho, a análise está limitada à faixa de menor energia (pior caso), nas condições do ATLAS: objetos e.m. com 20 GeV.

A normalização em energia pode ser feita de várias formas de acordo com a segmentação do sistema de calorimetria:

Energia do Evento Nesta modalidade soma-se a energia de todas as células da RoI. Divide-se o valor de energia em cada célula da RoI pelo valor total encontrado;

Energia da Seção Aqui divide-se o calorímetro do ATLAS em duas seções, e.m. e hadrônico, calculando a energia depositada em cada uma delas. Cada célula é dividida pela energia total de sua respectiva seção;

Energia da Camada O mesmo dos dois outros tipos de normalização, mas leva-se em consideração a energia em cada camada do calorímetro;

Sequencial Este tipo de normalização propõe um esquema de amplificação da energia dos anéis periféricos ao 1^o anel da camada, de forma que a informação contida nestes anéis seja realçada, possivelmente melhorando o desempenho do discriminador.

Nesta modalidade, para cada camada, calcula-se a soma das energias de cada célula e divide-se o valor da energia do primeiro anel por esta soma. A energia do segundo anel é dividida pelo valor da soma menos a energia do primeiro anel. A energia do terceiro anel, pela soma da energia total menos a energia do primeiro e segundo anéis, e assim sucessivamente até que o número de anéis se esgote.

Um problema ocorrerá: os valores energéticos nos últimos anéis de cada camada, aqueles nos quais a razão sinal-ruído é a menor, terão o menor fator normalização e, portanto, a maior amplificação. Isto é prejudicial ao desempenho de qualquer discriminador, pois beneficiar-se-ia canais ruidosos àqueles que realmente contêm informação relevante à discriminação. A solução é limitar a variação do fator de normalização por camada. Assim, por exemplo, na segunda camada e.m. somente os primeiros 7 anéis serão normalizados utilizando este esquema variável. Os demais anéis serão normalizados por um fator constante geralmente igual ao fator de normalização do 7^o anel. A Tabela 5.4 exemplifica a execução do algoritmo. Na primeira coluna é possível ver a posição do anel que será normalizado, na segunda o fator de normalização e na terceira a soma em anel correspondente. Nessa situação hipotética, há N anéis na camada e somente os $(N - 3)$ primeiros serão normalizados com o esquema variante. Os demais (a partir de $N - 2$) serão normalizados por um valor constante, igual ao fator de normalização do anel $(N - 3)$.

Como variantes deste processo, ao invés de se usar a energia total do objeto, pode-se utilizar a energia da seção ou da camada, relativas ao segmento para o qual se está aplicando a normalização, ou ainda esquemas híbridos. Ademais, a identificação em cada camada do ponto crítico (a partir do qual a normalização será feita de acordo a um fator constante) não é trivial [33]. Desta forma, optou-se por não realizar o estudo com normalização seqüencial neste trabalho, reservando-o para futuras atualizações do mesmo.

5.4 Treinamento Neural

Após compactar os dados usando a técnica de anéis, adaptada aqui para a granularidade variada camada a camada, é possível começar a projetar nosso discriminador neuronal. Lembrando, têm-se um conjunto de 58 somas em anel, que estão normalizadas de acordo com cada uma das técnicas descritas⁷ na Seção 5.3.1. Este fato estabelece imediatamente um dos parâmetros da rede neuronal: o número de nós de entrada. Como identificar-se-ão apenas duas classes de objetos, apenas um

⁷Exceto normalização seqüencial.

Tabela 5.4: Normalização sequencial.

Anel	Normalização	Soma em anel
1	E (total da camada)	E_1
2	$E - E_1$	E_2
3	$E - E_1 - E_2$	E_3
4	$E - E_1 - E_2 - E_3$	E_4
...	...	
N-4	$E - E_1 - E_2 - \dots - E_{N-5}$	E_{N-4}
N-3	$E - E_1 - E_2 - \dots - E_{N-5} - E_{N-4}$	E_{N-3}
N-2	$E - E_1 - E_2 - \dots - E_{N-5} - E_{N-4}$	E_{N-2}
N-1	$E - E_1 - E_2 - \dots - E_{N-5} - E_{N-4}$	E_{N-1}
N	$E - E_1 - E_2 - \dots - E_{N-5} - E_{N-4}$	E_N

nó na saída será suficiente. A função de ativação escolhida é a tangente hiperbólica. Durante o treinamento supervisionado por retro-propagação de erros, o alvo para elétrons será mantido em $+1$, enquanto que para jatos, -1 .

Um dos primeiros parâmetros a ser escolhido é, sem dúvida, o número de neurônios na camada intermediária. Estudos com componentes principais [34] indicam que cerca de 10 neurônios são suficientes para realizar discriminação similar. Isto motiva testes de redes tendo entre 5 e 15 neurônios na camada intermediária (escondida).

5.4.1 Critério de Parada: o produto SP

O critério de parada deve estar intimamente relacionado com a finalidade do projeto. Embora se estabeleça um conjunto de alvos, desejando que a rede convirja para $+1$ (quando elétrons aparecerem) ou -1 , se existirem jatos em sua entrada, se está fundamentalmente interessado no poder de discriminação desta rede. Sendo assim, o erro médio quadrático pode não traduzir facilmente tal objetivo, pois mede a distância euclidiana entre a saída do classificador e os valores-alvo, e não diretamente o poder de discriminação da rede neuronal. Desta maneira, é possível utilizar uma medida de eficiência do classificador como critério de parada do treinamento da rede

neuronal [22].

O poder de discriminação está relacionado à eficiência com que separa-se as duas classes de eventos. Desta forma, é necessário levar em conta os valores de eficiência para ambas as classes, pois deseja-se que ambos atinjam o máximo. Seria possível, por exemplo, considerar a soma de ambas as eficiências (S). Neste caso, valores próximos de 2 representariam uma excelente separação.

Considerando a soma das eficiências como critério de parada, entretanto, seria possível deparar com situações nas quais a eficiência de discriminação de uma das classes é alta, mas a da outra classe é baixa. Imagine, por exemplo, que se encontrou o valor da soma das eficiências igual a 1,3. Isto quer dizer que a primeira classe foi discriminada com 80% de acerto e a segunda com 50%? Talvez. Isto quer dizer que as eficiências juntas somam 1,3 e nada mais. A soma das eficiências, portanto, pode produzir eficiências de classificação não uniforme para as duas classes de interesse, o que não é interessante nesta aplicação, pois deseja-se atingir uma elevada eficiência de discriminação (em elétrons) com o mais baixo nível de falso alarme (detecção errônea de jatos).

O produto das eficiências, no entanto, restringe a existência de tal desequilíbrio, uma vez que valores baixos para a eficiência numa dada classe implicarão em valores baixos para tal produto. Neste trabalho, conjuga-se as duas medidas de desempenho do classificador, adotando o produto SP, i.e., a soma vezes o produto das eficiências (equação 5.5), como critério de parada no treinamento do discriminador neuronal.

$$SP = (efic_{classe_1} + efic_{classe_2}) \times (efic_{classe_1} \times efic_{classe_2}) \quad (5.5)$$

No método de treinamento que foi utilizado, guarda-se sempre a rede que, durante o treinamento, apresenta o melhor produto SP, levando-se em consideração, para um mesmo classificador, o melhor produto SP obtido observando-se a sua curva característica ponto-a-ponto. É claro que se espera que esta rede também apresente um baixo erro médio quadrático, embora esta variável não seja utilizada como critério de parada no processo de treinamento da rede. O erro médio quadrático, não obstante, é usado como função objetivo a ser minimizada pelo processo de treinamento da rede. A título de exemplo, a Figura 5.15 mostra a evolução do EMQ

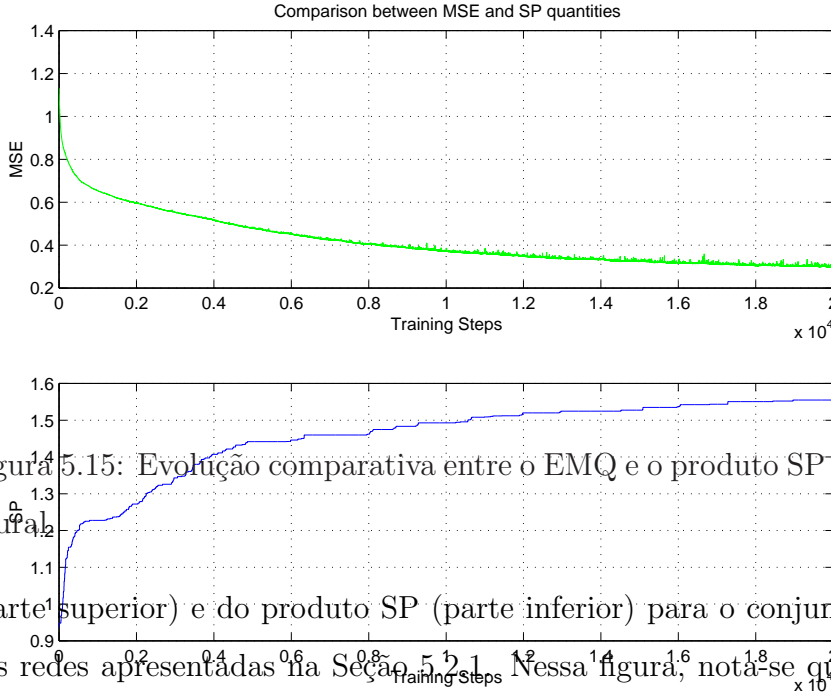


Figura 5.15: Evolução comparativa entre o EMQ e o produto SP em um treinamento neural (parte superior) e do produto SP (parte inferior) para o conjunto de teste de uma das redes apresentadas na Seção 5.2.1. Nessa figura, nota-se que o valor de SP de fato aumenta com a diminuição do EMQ, ainda que não se observe uma correlação plena entre as duas quantidades.

5.4.2 Treinamento

O método de treinamento a ser adotado é a retro-propagação de erros com momento e decaimento da taxa de treinamento, como na Seção 5.2.1. Também será adotado *épocas de treinamento*. O uso de *épocas* atenua a oscilação durante a convergência da rede neuronal [22, 29]. O tamanho da época terá que ser escolhido, assim como a taxa de aprendizado, seu decaimento e o momento. O número de passos de treinamento deverá ser grande o suficiente para a estabilização do produto SP e do erro médio quadrático.

O treinamento das redes neurais que executarão a separação elétron/jato foi feito dividindo-se o conjunto de dados em 2 sub-conjuntos, cada um contendo a metade do número de eventos total. Isto se traduz em cerca de 300 elétrons e 1800

jatos para cada conjunto.

As figuras de mérito que serão utilizadas para julgar o comportamento da rede são o EMQ para os conjuntos de treino e teste, a evolução do produto SP para os conjuntos de treino e teste e a comparação da curva característica obtida no classificador encontrado com a obtida pelo melhor discriminador neuronal usando as quantidades clássicas.

5.5 Resultados

A Tabela 5.5 resume os resultados encontrados, depois do treinamento de cerca de 20 redes neurais com parâmetros diferentes, para o vetor de entrada normalizado, usando-se o enfoque de normalização pela soma das energias de todas as células da RoI (energia total). A organização desta tabela é semelhante a da Tabela 5.2, com exceção da última coluna, que define o número de neurônios escondidos usados durante o treinamento.

Esta tabela mostra claramente que redes com muitos neurônios na camada intermediária são mais difíceis de serem treinadas, vindo a apresentar um EMQ acima do valor encontrado para redes com menos neurônios. Possivelmente, a estatística disponível não tenha permitido (ao treinamento neuronal) o ajuste adequado de tantas sinapses, já que o número de graus de liberdade do sistema está na ordem do número de eventos disponíveis nestes casos. Por exemplo, ao construir uma rede neuronal com 58 entradas, 15 neurônios na camada intermediária e 1 neurônio na saída, obriga-se o treinamento a ajustar $(15 \times 58) + 15$ sinapses e mais 16 polarizações, ou seja, 900 diferentes valores. Em contrapartida, o número de eventos disponível está na ordem de 2000 (300 elétrons e 1800 jatos). Embora não caracterize uma situação clássica de *over-fitting* [26], pode-se esperar que haja uma maior dificuldade no processo de treinamento usando-se mais neurônios na camada intermediária, resultando, como é possível ver, num pior classificador. O valor do produto SP é mais baixo que em outros testes, confirmando a tendência apontada pelo EMQ.

A redução do número de neurônios aumentou consideravelmente as oscilações na convergência do processo. Para atenuar estas oscilações, inseriu-se um valor de momento diferente de zero em testes de redes cujo número de neurônios na camada

Tabela 5.5: Os resultados do treinamento de 20 redes neurais para separação elétron/jato utilizando anéis. A normalização do vetor de entrada se utiliza da soma das energias de todas as células da RoI.

Teste	μ	α	γ	SP	# Passos	Bat.	Iteração	EMQ	Patamar	% e ⁻	% jatos	Tx.(kHz)	# esc.
1	0,1	0	1	1,72	2500	400	2341	0,206	0,170	95,38	94,76	1,31	15
2	0,15	0	1	1,72	2500	400	2047	0,182	0,070	96,04	94,26	1,43	15
3	0,15	0	0,9	1,72	2500	400	2405	0,154	-0,050	95,71	94,59	1,35	15
4	0,15	0	0,9	1,73	2500	200	1129	0,262	0,410	95,38	95,03	1,24	15
5	0,02	0	0,95	1,70	20000	25	14365	0,245	0,180	95,05	94,43	1,39	15
6	0,05	0,9	0,95	1,74	2000	100	1996	0,159	0,060	96,04	94,92	1,27	12
7	0,05	0,9	0,95	1,75	3000	100	2944	0,213	0,550	95,05	96,14	0,97	12
8	0,05	0,9	0,95	1,75	5000	100	4776	0,148	-0,030	96,70	94,70	1,32	12
9	0,05	0,9	0,95	1,76	5000	100	5000	0,150	-0,230	98,35	93,38	1,66	14
10	0,1	0,9	0,95	1,75	1000	100	741	0,145	-0,010	96,04	95,14	1,21	7
11	0,08	0,95	0,95	1,77	5000	100	3949	0,156	-0,130	97,69	94,26	1,43	7
12	0,08	0,95	0,95	1,77	5000	100	4736	0,129	-0,360	97,36	94,54	1,37	8
13	0,08	0,95	0,95	1,77	5000	100	4158	0,130	-0,290	97,03	94,87	1,28	6
14	0,12	0,95	0,95	1,76	4000	150	2633	0,285	0,690	97,03	94,70	1,32	5
15	0,09	0,99	0,95	1,77	3000	200	2761	0,160	-0,220	98,02	93,87	1,53	5
16	0,05	0,99	0,95	1,76	5000	200	2562	0,109	-0,830	97,36	94,43	1,39	5
17	0,01	0,99	0,95	1,76	5000	200	2882	0,133	-0,110	96,70	95,09	1,23	5
18	0,09	0,95	0,8	1,76	5000	100	1632	0,216	0,400	96,70	94,92	1,27	5
19	0,09	0,99	0,8	1,75	5000	200	3462	0,168	-0,040	97,03	94,37	1,41	5
20	0,09	0,9	0,8	1,76	5000	200	3467	0,162	0,020	97,03	94,65	1,34	5

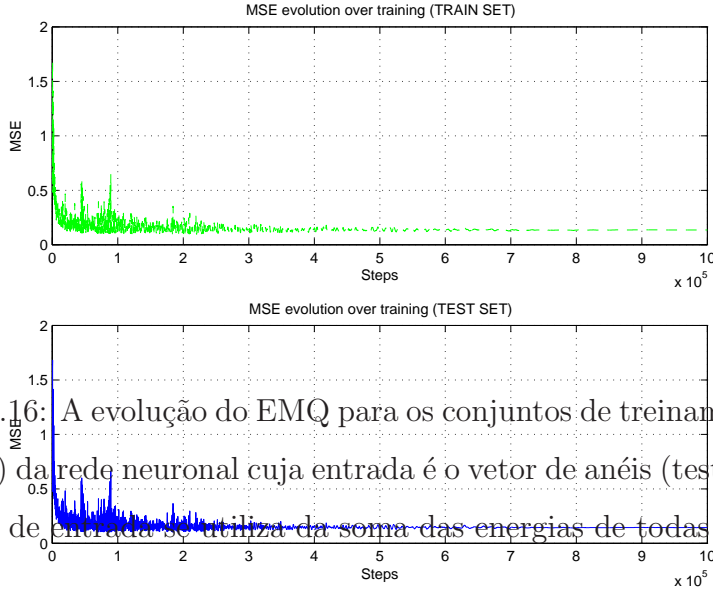


Figura 5.16: A evolução do EMQ para os conjuntos de treinamento (superior) e teste (inferior) da rede neuronal cuja entrada é o vetor de anéis (teste 20). A normalização do vetor de entrada se utiliza da soma das energias de todas as células da RoI.

intermediária era menor que 15. Juntamente com a redução do decaimento da taxa de treinamento para 0,95, foi possível chegar a redes com apenas 5 neurônios na camada intermediária sem degradação do desempenho, seguindo nosso critério de parada (o produto SP). A evolução do EMQ nessas redes, no entanto, ainda se mostrou muito ruidoso.

Ao diminuir o valor do decaimento para 0,8, procura-se uma convergência suave de nosso discriminador, em direção ao mínimo que garanta o desempenho obtida nos testes com 5 neurônios na camada intermediária. Os últimos dois testes confirmam os resultados obtidos nos testes 14 a 18: baixo erro médio quadrático e alto poder de discriminação. A evolução do EMQ nestes testes se dá de forma suave, como é possível observar na Figura 5.16. Essa figura é referente ao teste 20.

5.5.1 Resultados para outras normalizações

Testou-se também o discriminador neuronal utilizando as normalizações por camada e seção, discutidas anteriormente, sem melhoria nos resultados. De fato, não foi possível atingir o mesmo desempenho com os dados, enquanto normalizados a partir da energia total da RoI. A princípio, acredita-se que a rede *dê* uma

importância bastante grande as diferenças energéticas entre as duas seções dos calorímetros do ATLAS. Ao normalizar os dados de entrada utilizando a energia da camada ou seção, se está, de fato, destruindo esta diferença. Isso influencia no comportamento do discriminador neural, movendo-o de seu ponto ótimo. É possível que a utilização de uma normalização híbrida [33] nesses casos seja mais conveniente que o uso de somente um enfoque no processo de normalização dos valores energéticos dos anéis, já que assim beneficia-se as diferenças de granularidade em cada camada e respeitando suas funções no experimento. Por exemplo, o pré-irradiador é um calorímetro muito fino e é utilizado somente para aumentar a precisão da localização do centro da cascata do objeto. Uma normalização que utiliza toda a energia da RoI prejudica o treinamento neuronal neste caso, à medida que praticamente elimina a informação contida naquelas células. Este enfoque híbrido e preocupado com as funções de cada camada do calorímetro e suas respectivas granularidades não foi desenvolvida neste trabalho, tendo sido reservado para investigações futuras.

A Tabela 5.6 mostra alguns resultados conseguidos utilizando as normalizações por camadas e por seção, respectivamente. Como é possível ver, o EMQ tende a aumentar muito quando utiliza-se esses tipos de normalização. A utilização de mais neurônios na camada intermediária tende a piorar o resultado. Isto confirma as tendências dos testes na Tabela 5.5. Mantendo a configuração do discriminador e do treinamento que rendeu-nos os melhores resultados, quando utilizou-se como fator de normalização toda a energia da RoI, os novos discriminadores gerados a partir do treinamento com vetores normalizados pela energia da seção ou da camada não se mostraram tão eficientes.

5.6 Informação Relevante

No ambiente de física experimental, determinar a informação relevante é de grande importância, podendo revelar canais físicos interessantes ao experimento. Por outro lado, conhecendo-se a informação relevante, pode-se compactar mais ainda a dimensionalidade dos sinais originais de entrada, ou mesmo manter a informação redundante, de forma a garantir a robustez do discriminador, em caso da ausência de alguns canais dos calorímetros.

Tabela 5.6: Os resultados do treinamento de algumas redes neurais usando os 58-anéis. Na tabela superior mostra-se os testes utilizando a energia da camada como fator de normalização dos dados. Na inferior, a energia da seção.

Teste	μ	α	γ	SP	# Passos	Bat.	Iteração	EMQ	Patamar	% e ⁻	% jatos	Tx.(kHz)	# esc.
1	0.090	0.900	0.800	1.42	5000	200	4300	0.3880	-0.430	96.37	82.56	4.36	5
2	0.200	0.900	0.800	1.43	5000	200	1923	0.4470	-0.170	97.36	81.90	4.53	5
3	0.050	0.900	0.700	1.45	5000	200	4735	0.3820	0.100	91.09	88.52	2.87	5
4	0.050	0.900	0.700	1.47	5000	200	3368	0.3640	-0.080	94.39	86.31	3.42	8
5	0.050	0.900	0.700	1.40	5000	200	3337	0.3710	-0.350	91.42	86.09	3.48	12

Teste	μ	α	γ	SP	# Passos	Bat.	Iteração	EMQ	Patamar	% e ⁻	% jatos	Tx.(kHz)	# esc.
1	0.090	0.900	0.800	1.60	5000	200	2086	0.2130	-0.070	93.40	92.44	1.89	5
2	0.050	0.900	0.800	1.60	2500	200	1664	0.2880	0.170	94.06	91.78	2.06	5
3	0.010	0.900	0.800	1.60	5000	200	4489	0.2500	0.130	92.41	93.10	1.72	5
4	0.010	0.900	0.800	1.48	5000	200	4112	0.3140	-0.080	92.08	88.85	2.79	10

Para identificar a informação relevante, pode-se analisar a resposta do classificador neuronal em termos da sua sensibilidade à informação contida em cada canal. Para tal, o mapeamento de relevância [35] mede a variação na saída do discriminador, quando substitui-se uma componente do vetor de entrada pelo seu valor médio. Esta operação é repetida para todos os vetores de entrada. Tais variações são calculadas ao quadrado, somadas e normalizada pelo número total de padrões considerado. O valor encontrado mede, então, a importância da variável no desempenho de saída do classificador. A este valor dá-se o nome de *relevância*. A equação 5.6 mede a relevância da *i*-ésima componente de entrada. Nessa equação, N representa o número total de eventos (padrões) disponíveis para a rede, e x_j um evento particular.

$$R_i = \frac{1}{N} \sum_{j=1}^N [\text{saída}(\vec{x}_j) - \text{saída}(\vec{x}_j |_{x_{j,i}=\bar{x}_i})]^2 \quad (5.6)$$

5.6.1 Relevância em discriminadores lineares

Como é possível notar, a equação 5.6 não leva em conta o projeto do discriminador. É possível dessa forma, aplicá-la à discriminadores lineares, ou seja, aqueles que funcionam através da definição de patamares que melhor discriminam o conjunto de dados de entrada, observando o espaço M-dimensional onde esses dados se encontram.

Um discriminador linear pode ser representado por um neurônio cuja função de ativação é a identidade. A Figura 5.17 representa os elementos que são encontrados em um discriminador linear. A saída do discriminador é o somatório, ponderado pelos pesos w_i , de todas as entradas x_i .

Ao substituir-se uma das entradas do discriminador linear por sua média, calcula-se a relevância desta variável. Definindo então:

$$\begin{aligned} x_{ij} &= x_i \text{ para a } j\text{-ésima componente de entrada} \\ \bar{x}_i &= E[x_i] \\ y &= x^T w \\ y_i &= x |_{x_i=\bar{x}_i}^T w \end{aligned} \quad (5.7)$$

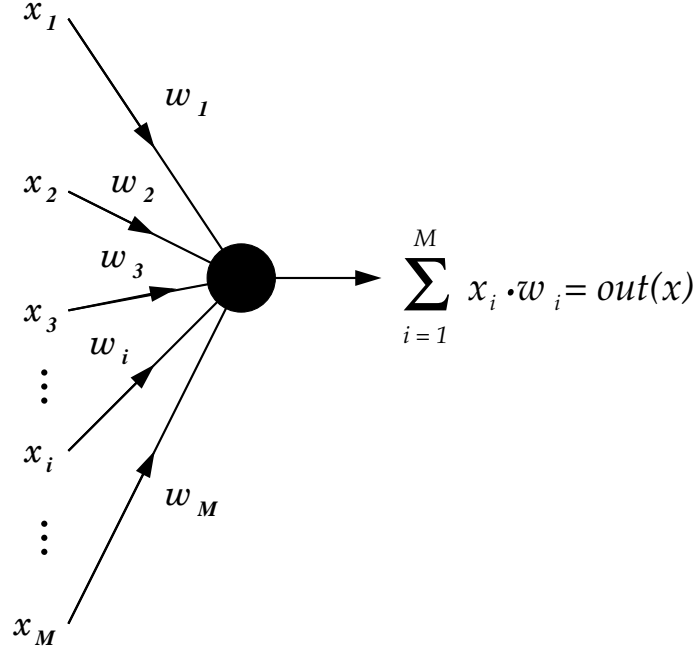


Figura 5.17: Diagrama de um discriminador linear.

Prossegue-se com as seguintes equações:

$$\begin{aligned}
 R_i &= \frac{1}{N} \sum_{j=1}^N [y_j - y_{ji}]^2 \\
 &= \frac{1}{N} \sum_{j=1}^N [x_j^T w - x_j^T |_{x_{ij}=\bar{x}_i} w]^2 \\
 &= \frac{1}{N} \sum_{j=1}^N [(x_j^T - x_j^T |_{x_{ij}=\bar{x}_i}) w]^2
 \end{aligned}$$

O valor das diferenças de todos os termos de x , exceto o i -ésimo será 0, portanto:

$$\begin{aligned}
 R_i &= \frac{1}{N} \sum_{j=1}^N [(x_{ij} - \bar{x}_i) w_i]^2 \\
 &= \frac{w_i^2}{N} \sum_{j=1}^N [x_{ij} - \bar{x}_i]^2 \\
 &= \frac{(N-1) \cdot w_i^2}{N} \cdot \sigma_i^2 = w_i^2 \cdot \sigma_i^2, \text{ se } N \longrightarrow \infty
 \end{aligned} \tag{5.8}$$

Ou seja, para um discriminador linear, calcular a relevância de suas entradas é o mesmo que estimar a variância dessas entradas, quando estão ponderadas pelo peso da sinapse que conecta cada entrada ao neurônio linear. Este resultado

é razoável, pois, para um separador linear, uma variável altamente discriminante é aquela cuja função de densidade de probabilidade (f.d.p.) possui um domínio bastante extenso. Uma variável que possua uma pequena variância será inevitavelmente pouco discriminante, já que um patamar não poderá ser fixado sem que haja uma perda muito grande em uma das classes de eventos a serem separadas.

No caso de redes neurais, não há como traçar um paralelo com a estatística de forma simples, como feito com o discriminador linear, já que operações não-lineares estão envolvidas. Não há sentido em definir a qualidade de discriminação das classes através da variância das componentes, ainda que cortes mais suaves possam ser esboçados em variáveis cuja variância seja maior. Isto equivale a dizer que é mais fácil separar eventos cujas variáveis que os representam estejam espalhadas por um grande domínio, do que separar eventos cujas variáveis representativas possuam uma baixa variância.

5.6.2 Relevância das quantidades clássicas

A Figura 5.18 mostra o valor da relevância para cada uma das 4 quantidades clássicas, quando o mapeamento relevante é aplicado ao discriminador do teste 20 (Tabela 5.2). Nessa figura, mostra-se o valor de relevância para os conjuntos de treino e teste, de forma que o leitor possa verificar a concordância entre os valores obtidos para os dois agrupamentos de dados. Nesse discriminador, se obteve um alto desempenho de discriminação associado a um baixo EMQ. De baixo para cima, na figura, é possível ver as barras de relevância para as variáveis $E_{T_{3 \times 7}}^{e.m.}$, $E_{T_{0,2 \times 0,2}}^{HAD}$, R_{η}^{shape} e $R_{e.m.1}^{strip}$. É possível notar que a variável mais relevante é $E_{T_{0,2 \times 0,2}}^{HAD}$.

Em último lugar no quesito relevância para aquele discriminador, têm-se $R_{e.m.1}^{strip}$. Como foi visto nas Figuras 5.5 e 5.6, esta variável possui uma pequena variância, tornando-se extremamente difícil a separação entre as duas classes de eventos. Todos os testes realizados concordam com esta afirmação.

A Figura 5.19 mostra um gráfico similar ao da Figura 5.18, mas tem como fonte o discriminador do teste 18 da Tabela 5.2. Este teste foi escolhido, uma vez que o discriminador gerado ali apresenta baixa eficiência de discriminação e alto EMQ, comparando-se aos dos outros testes. Como é possível ver, a relevância da quantidade 1 ($E_{T_{3 \times 7}}^{e.m.}$) é bem menor para este discriminador que no teste 20,

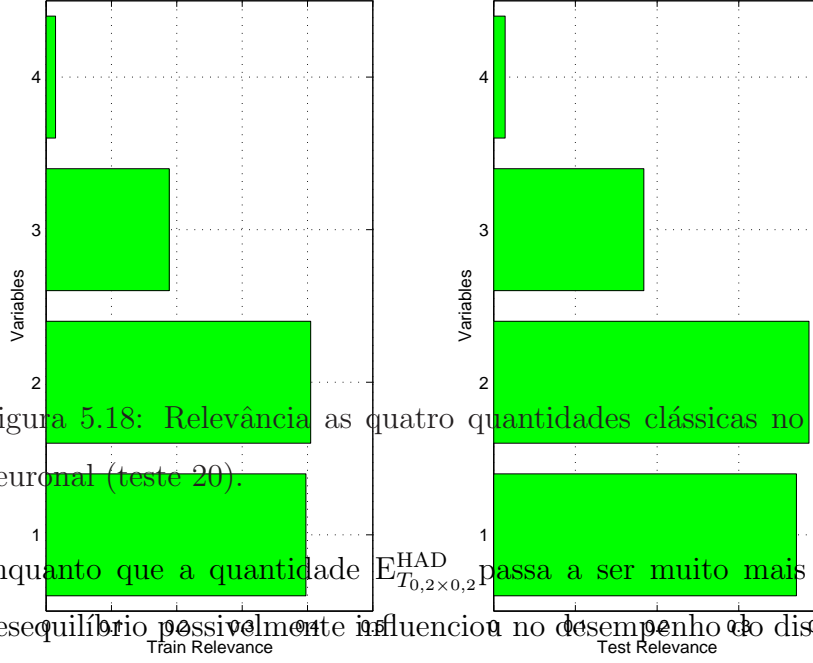
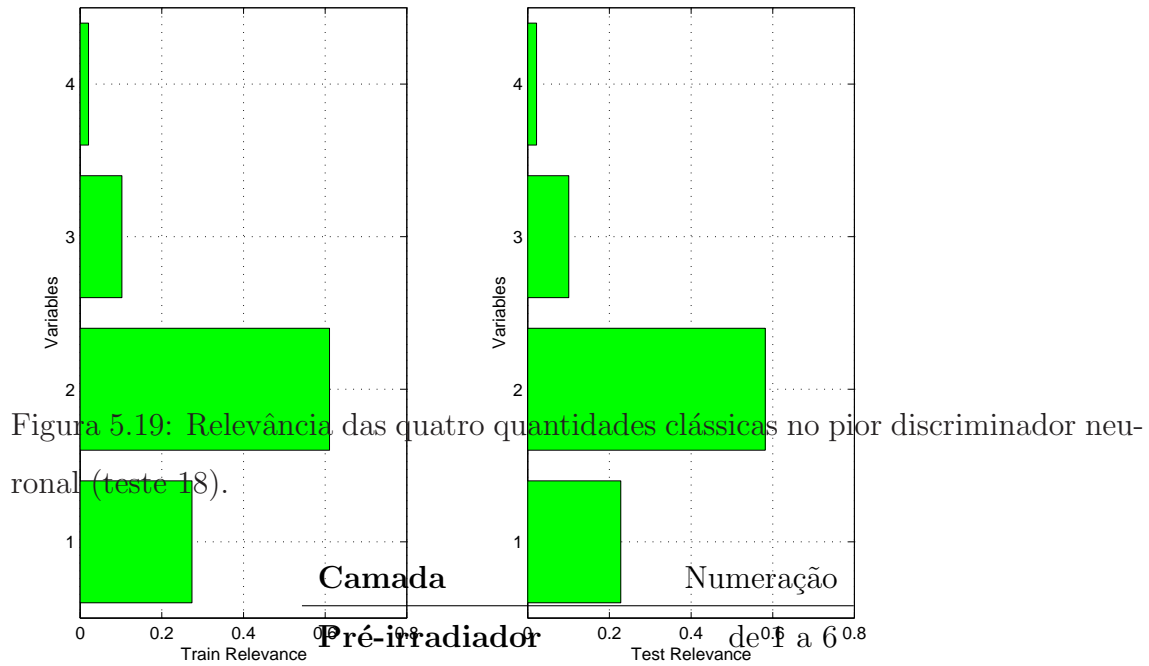


Figura 5.18: Relevância as quatro quantidades clássicas no melhor discriminador neuronal (teste 20).

enquanto que a quantidade $E_{T_{0,2 \times 0,2}}^{\text{HAD}}$ passa a ser muito mais relevante aqui. Este desequilíbrio possivelmente influenciou no desempenho do discriminador, ainda que uma análise dos eventos erroneamente classificados pelos dois discriminadores não mostrasse qualquer tendência.

5.6.3 Relevância dos anéis na separação neuronal

Aplicou-se o estudo de relevância aos discriminadores baseados em anéis. A Figura 5.20 mostra, empilhadas, as relevâncias dos anéis para os melhores discriminadores cujos fatores de normalização baseiam-se na energia total da RoI (teste 17), na energia da camada (teste 4) e na energia da seção (teste 1), respectivamente. Os valores, de baixo para cima, representam as relevâncias de:



1 ^a camada e.m.	de 7 a 39
2 ^a camada e.m.	de 40 a 49
1 ^a camada Hadrônica	de 50 a 52
2 ^a camada Hadrônica	de 53 a 55
3 ^a camada Hadrônica	de 56 a 58

Nesta figura, observa-se que o discriminador, cuja entrada é normalizada pela energia total da RoI, dá bastante importância ao centro de energia no Pré-irradiador, ao centro energético (1^o anel) e ao segundo anel na primeira camada e ao primeiro e segundo anéis da segunda camada e.m.. Os discriminadores cujas entradas são normalizadas pela energia da camada e energia de seção valorizam notavelmente de forma mais intensa a deposição de energia nos anéis da seção hadrônica.

A figura 5.21 mostra a deposição de energia nas camadas da seção hadrônica para objetos e.m. (elétrons e jatos). É possível ver que, para elétrons, a deposição de energia é praticamente nula. A flutuação que é percebida é devida ao ruído instrumental nas camadas (subtraída de fatores de correção). Para jatos, mostra-se

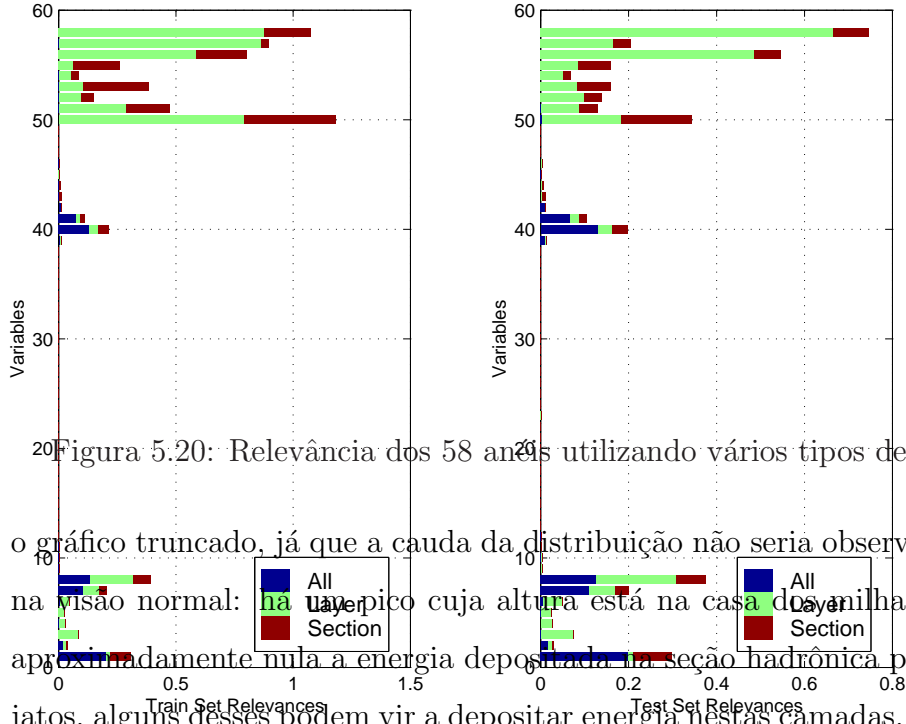


Figura 5.20: Relevância dos 58 anos utilizando vários tipos de normalização.

o gráfico truncado, já que a cauda da distribuição não seria observada corretamente na visão normal: na camada 50, o pico cuja altura está na casa dos milhares. Embora seja aproximadamente nula a energia depositada na seção hadrônica para a maioria dos jatos, alguns desses podem vir a depositar energia nestas camadas. Quando o fazem, são facilmente discrimináveis de elétrons.

Conclui-se que a normalização baseada na energia da seção ou na energia da camada onde se encontra a célula amplifica em demasiado o ruído contido nas camadas da seção hadrônica. Isso piora em muito a eficiência de discriminação entre elétrons e jatos, já que a maioria das RoI's não possui informação nestas camadas. É importante frisar, no entanto, que a seção hadrônica contém alguma informação de discriminação, possivelmente contida no pico de sua primeira camada. De fato, ao se redesenhar a relevância para os dados de entrada do discriminador que utiliza como fator de normalização a energia contida na RoI, percebe-se que a quantidade 50 (1 anel na primeira camada hadrônica) não é totalmente irrelevante (vide Figura 5.22).

As diferenças nos valores de relevância entre os discriminadores baseados em anéis e nas quatro quantidades é marcante. No primeiro, a discriminação baseia-se nos perfis de deposição nos centros energéticos da seção e.m. basicamente, enquanto

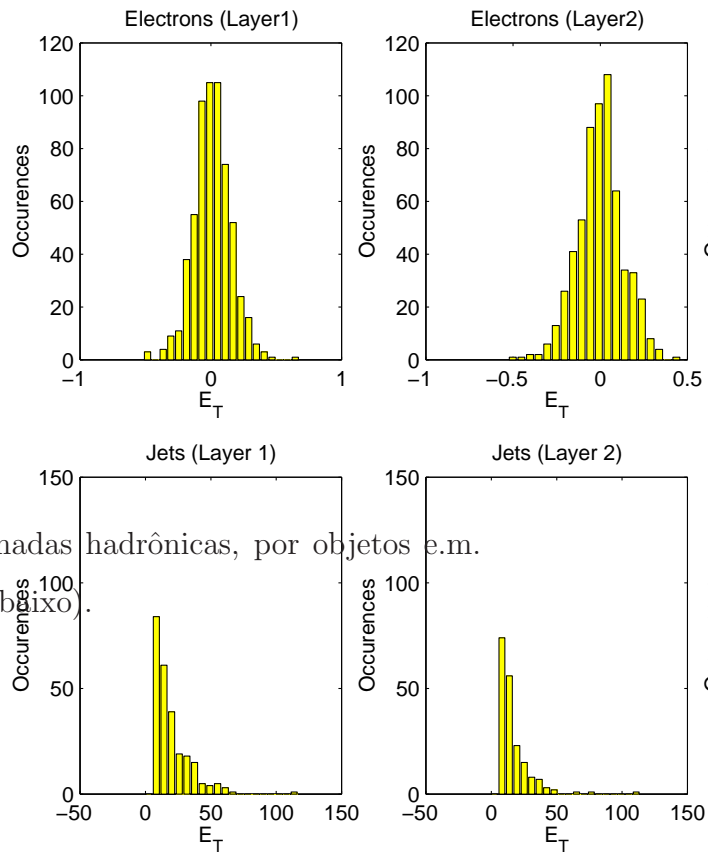


Figura 5.21: A deposição de energia nas 3 camadas hadrônicas, por objetos e.m. que representam elétrons (em cima) e jatos (embaixo).

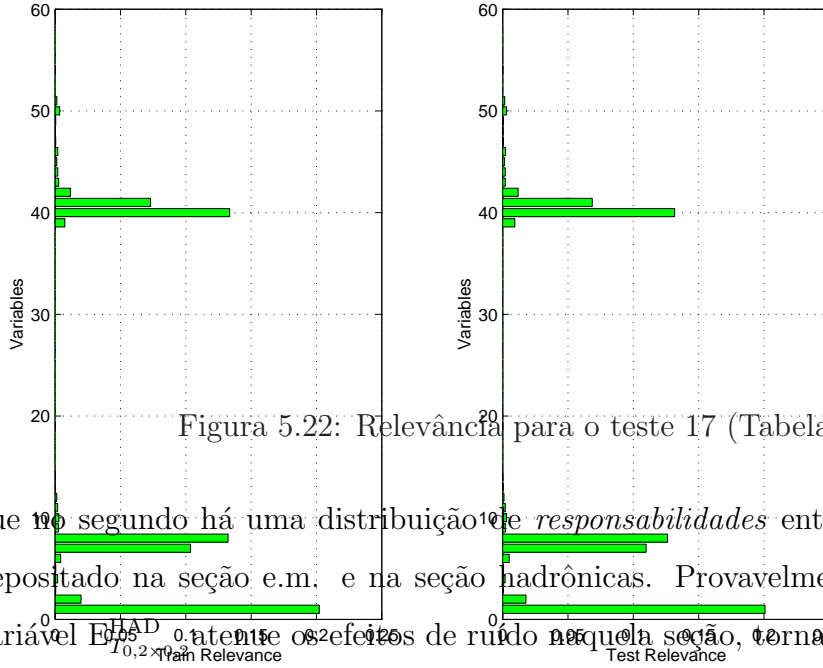


Figura 5.22: Relevância para o teste 17 (Tabela 5.5).

que segundo há uma distribuição de *responsabilidades* entre o valor energético depositado na seção e.m. e na seção hadrônica. Provavelmente a construção da variável E_{HAD} atenua os efeitos de ruído naquela seção, tornando-a altamente discriminante. Por outro lado, possivelmente, a deposição de tamanha responsabilidade em uma variável extremamente ruidosa acaba por degradar, em algumas unidades percentuais, o desempenho do discriminador usando as quatro quantidades clássicas.

5.6.4 Cortes de dimensionalidade baseados na relevância

Embora apresente um melhor desempenho, o discriminador neuronal, que utiliza os anéis montados a partir da RoI, representa uma opção potencialmente lenta para L2, já que a montagem dos anéis e o processamento neuronal são atividades mais elaboradas que a extração das quatro quantidades clássicas e a execução de um algoritmo de separação bi-dimensional baseado em cortes. Ainda assim, este método apresenta uma robustez intrinsicamente maior, já que utiliza um maior número de variáveis na discriminação, adicionando redundância ao sistema.

A partir do estudo de relevâncias, um método de *poda* de rede é proposto: dado um patamar, eliminar-se-á, da entrada, todas as variáveis cuja relevância seja

menor que o patamar. Treina-se uma nova rede neuronal mais compacta, para que aprenda a separar elétrons e jatos utilizando somente as variáveis que restaram.

Espera-se que a eficiência de discriminação com este processo de poda não deteriore-se de forma muito acentuada. De fato, espera-se que a robustez do classificador seja suficientemente grande, para que se possa suprir com a deficiência inserida no sistema. Por outro lado, espera-se estar eliminando informação potencialmente ruidosa (das bordas do objeto predominantemente) e não seria inesperado que a eficiência de discriminação da rede aumentasse após o retreinamento.

5.6.4.1 As variáveis clássicas

Inicialmente, o método de redução de dimensionalidade do vetor de entrada será aplicado ao discriminador neuronal que é alimentado pelas quatro quantidades clássicas. A poda, neste caso, é óbvia: eliminar-se-á a variável $R_{e.m.1}^{\text{strip}}$, que tem a menor relevância. A dimensionalidade dos dados de entrada cairá, desta forma, de 4 para 3.

Para se treinar a nova rede neuronal, utilizar-se-á, inicialmente, a configuração do teste 3 da Tabela 5.2, variando-a até atingir-se o desempenho equivalente a esse teste. Utilizou-se os resultados deste teste, pois é o que conjuga o maior valor de produto SP na tabela, um patamar de separação aproximadamente em zero (indicando boa separação das classes) e um baixo erro médio quadrático, comparando-se com os outros resultados naquela tabela.

A variação do número de neurônios na camada escondida poderá, eventualmente, melhorar o desempenho da rede, pois garantirá um maior espaço de acomodação para os dados, quando projetados na saída dessa camada. Lembra-se que a saída da camada intermediária para cada padrão de entrada pode ser entendida como um vetor que está no interior do hipercubo cuja dimensão é igual ao número de neurônios daquela camada. Um pequeno aumento no número de neurônios na camada intermediária, portanto, aumenta a área disponível para a acomodação dos padrões de entrada, facilitando o posicionamento do plano separador que será traçado pelo neurônio na saída do classificador neuronal.

A Tabela 5.7 resume os resultados obtidos. A Figura 5.23 mostra a curva de eficiência obtida no teste 4 (cut-34). Este corte é baseado nas relevâncias obtidas

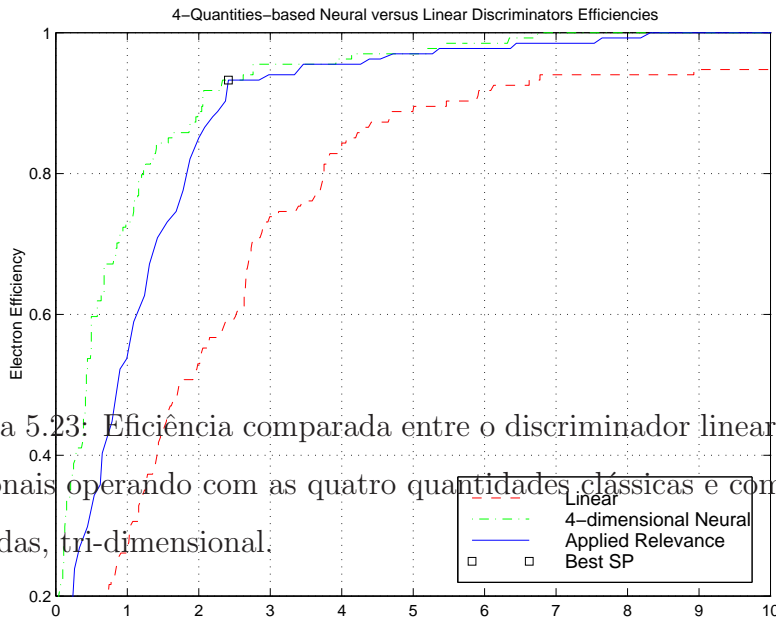


Figura 5.23: Eficiência comparada entre o discriminador linear e os discriminadores neurais operando com as quatro quantidades clássicas e com o vetor reduzido de entradas, tri-dimensional.

no teste 3, mostradas na Figura 5.18, quando o patamar de eliminação é 0,2, i.e., eliminando a componente $R_{e.m.1}^{\text{strip}}$.

Como é possível ver nesta figura (5.23), a curva de eficiência está, para a maior parte da abscissa, abaixo da curva anterior, mas acima da curva gerada por um discriminador linear. Entretanto, o ponto com maior produto SP está muito próximo da curva de eficiência obtida para o teste utilizando as 4 componentes clássicas, indicando que o desempenho é compatível com os discriminadores neurais utilizando o espaço completo de variáveis clássicas. O número de neurônios na camada escondida para este teste teve que ser aumentado de 4 para 5. Isso representa um aumento no número de operações para discriminação de 4 multiplicações e uma tangente hiperbólica, mas a eliminação desta quarta variável reduz a complexidade de pré-processamento das características do objeto e.m., tornando desnecessária a procura de picos na primeira camada e.m. e uma divisão.

Tabela 5.7: Resultados obtidos com algumas redes neurais cujo vetor de entrada são as componentes clássicas de discriminação. A quarta componente ($R_{e.m.1}^{\text{strip}}$) foi eliminada do vetor de entrada, que agora é tri-dimensional.

Teste	μ	α	γ	SP	# Passos	Bat.	Iteração	EMQ	Patamar	% e ⁻	% jatos	Tx. (kHz)	# esc.
cut-31	0.050	0.900	0.800	1.54	1000	100	719	0.2510	-0.030	93.28	90.07	2.48	4
cut-32	0.050	0.700	0.800	1.55	1000	100	655	0.4870	0.340	93.28	90.40	2.40	5
cut-33	0.050	0.500	0.800	1.55	1000	100	960	0.4920	0.310	93.28	90.29	2.43	5
cut-34	0.050	0.500	0.800	1.55	1500	100	973	0.4630	0.280	93.28	90.34	2.41	5
cut-35	0.050	0.500	0.800	1.51	1500	100	1491	0.4620	0.360	93.28	88.91	2.77	7

5.6.4.2 Eliminando anéis

O prosseguimento natural é aplicar os cortes de dimensionalidade baseados na relevância em discriminadores que utilizam os anéis. Reduzindo-se o número de anéis de entrada para o discriminador, sem perda de desempenho, se reduz a carga computacional do pré-processamento de dados. Por outro lado, é possível ver através da Figura 5.22 que grande parte do vetor de entrada é pouco relevante para a rede. Neste caso, a eliminação destes anéis não representará somente um ganho no tempo de pré-processamento dos dados, mas também no tempo de processamento do classificador neuronal, que efetuará menos produtos internos.

Para realizar este estudo, utilizou-se o teste 17 da Tabela 5.5, considerando-o o melhor resultado de discriminação até aqui alcançado. A relevância de cada uma das 58 componentes de entrada já foi discutida e pode ser vista na Figura 5.22. É possível selecionar vários patamares diferentes neste caso. Estes patamares nos levarão a cortes mais severos, ou não, no número utilizado de anéis. Neste trabalho, dois métodos para o corte de dimensionalidade dos dados de entrada são propostos:

- Corte seqüencial - define-se um patamar conservador, reduz-se o conjunto de variáveis de entrada e nova análise de relevância é feita. Um novo corte é feito e assim é feito sucessivamente até que o desempenho do discriminador apresente degradação acentuada. Em outras palavras, analisa-se o novo conjunto de dados com relação à sua relevância para o melhor discriminador neuronal encontrado. É possível aplicar um novo corte e re-análise ou parar, se houver uma queda muito grande no desempenho. Este procedimento de re-análise permite que a relevância das variáveis, que foram mantidas, seja reavaliada.
- Corte direto - a partir da análise inicial de relevância, usando-se os 58 anéis, é possível estabelecer cortes mais severos, reduzindo o número de componentes de entrada.

Corte seqüencial Estabeleceu-se, inicialmente, um corte em 0,001. Este corte poupará apenas 20 das 58 variáveis iniciais. Estas variáveis estão listadas na Tabela 5.8. Como é possível ver, a maior parte das variáveis que restaram pertencem ao calorímetro e.m.. Este resultado corresponde ao fato que, na maior parte dos

Tabela 5.8: As variáveis do espaço original de anéis de dimensão 58, após o corte baseado na relevância.

Calorímetro	Anéis Restantes
Pré-irradiador	1, 2, 4 e 6
1 ^a camada e.m.	7, 8, 9, 10, 11, 12 e 39
2 ^a camada e.m.	40, 41, 42, 43, 44, 45 e 46
1 ^a camada hadrônica	50 e 51

dados, no calorímetro hadrônico a relação sinal/ruído é baixa, ou seja, há pouca informação relevante à discriminação, considerando a maior parte dos dados que se possui⁸.

A Tabela 5.9 resume alguns resultados obtidos após o treinamento de redes neurais baseadas no vetor de entrada, que se encontra descrito na Tabela 5.8. A Figura 5.24 mostra a curva de eficiência de um dos testes dessa tabela (teste 6). O resultado é bastante semelhante ao original (teste 17). No entanto, utiliza-se um sub-espaço dos dados de entrada e 1 neurônio a menos na camada intermediária, o que representa um ganho computacional significativo. Percebe-se através desta tabela, que o erro médio quadrático pouco se altera em relação aos testes da Tabela 5.5, que utilizavam todo o espaço de entrada (os 58 anéis). Um dos discriminadores, o do teste 8, utiliza somente 3 neurônios na camada intermediária, sua curva de eficiência também foi bastante próxima a curva do discriminador do teste 17 da Tabela 5.5, ainda que o EMQ atingido ao final do treinamento não tivesse sido o menor de todos na Tabela 5.9.

Para prosseguir com os cortes na dimensionalidade do espaço de entrada, selecionou-se o discriminador do teste 6, pois além de apresentar o menor valor de EMQ obtido nos testes, apresenta também alta eficiência. A partir desse discriminador, recalculou-se as relevâncias das entradas para o discriminador neuronal. A Figura 5.25 mostra o resultado destes cálculos. As entradas foram renumeradas neste gráfico, de forma que a primeira quantidade apontada na Tabela 5.8 equivale a entrada 1, a segunda a entrada 2 e assim sucessivamente. A última variável (o anel número 51) equivale a 20^a entrada. Este gráfico mostra que as variáveis

⁸De fato, este método de compactação não leva em conta que alguns dos eventos podem ser bem discriminados usando a informação da seção hadrônica.

Tabela 5.9: Resultados de alguns discriminadores neuronais utilizando o espaço reduzido de anéis (dimensão 20).

Teste	μ	α	γ	SP	# Passos	Batelada	Iteração	EMQ	Patamar	% e ⁻	% jatos	Tx.(kHz)	# esc.
1	0.010	0.990	0.950	1.75	5000	200	4746	0.1720	-0.170	98.02	93.32	1.67	5
2	0.010	0.900	0.950	1.74	5000	200	4763	0.1670	0.030	96.04	94.92	1.27	5
3	0.010	0.900	0.950	1.73	5000	200	4025	0.2030	0.150	96.37	94.26	1.43	6
4	0.010	0.900	0.950	1.73	5000	200	4857	0.1760	0.150	95.38	95.09	1.23	4
5	0.020	0.900	0.950	1.73	5000	200	3494	0.1630	0.110	95.38	95.14	1.21	4
6	0.020	0.900	0.950	1.73	5000	200	3628	0.1490	0.030	95.38	95.25	1.19	4
7	0.020	0.900	0.900	1.73	5000	200	4653	0.1840	0.230	95.38	95.20	1.20	4
8	0.020	0.900	0.950	1.72	5000	200	4998	0.1740	0.070	95.71	94.43	1.39	3

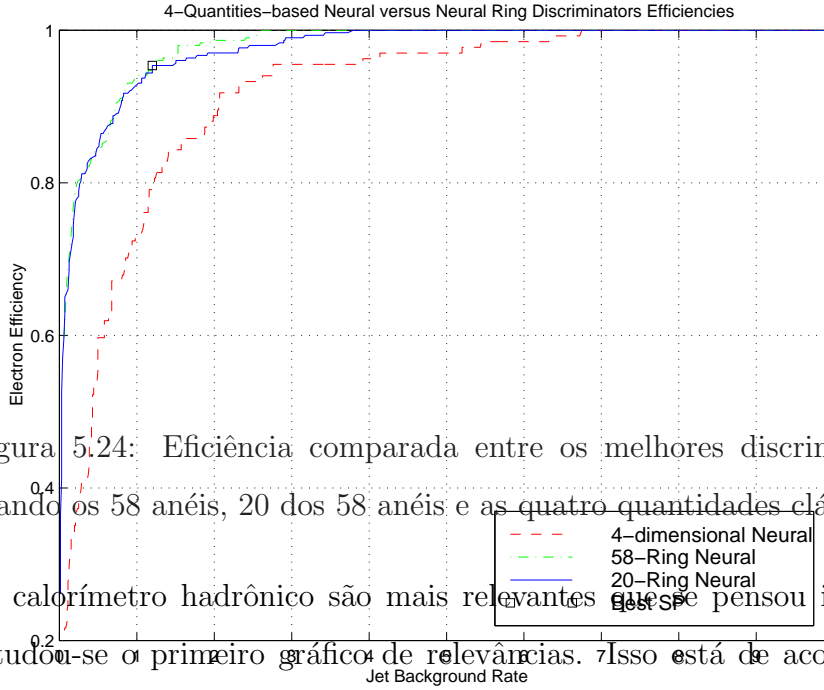
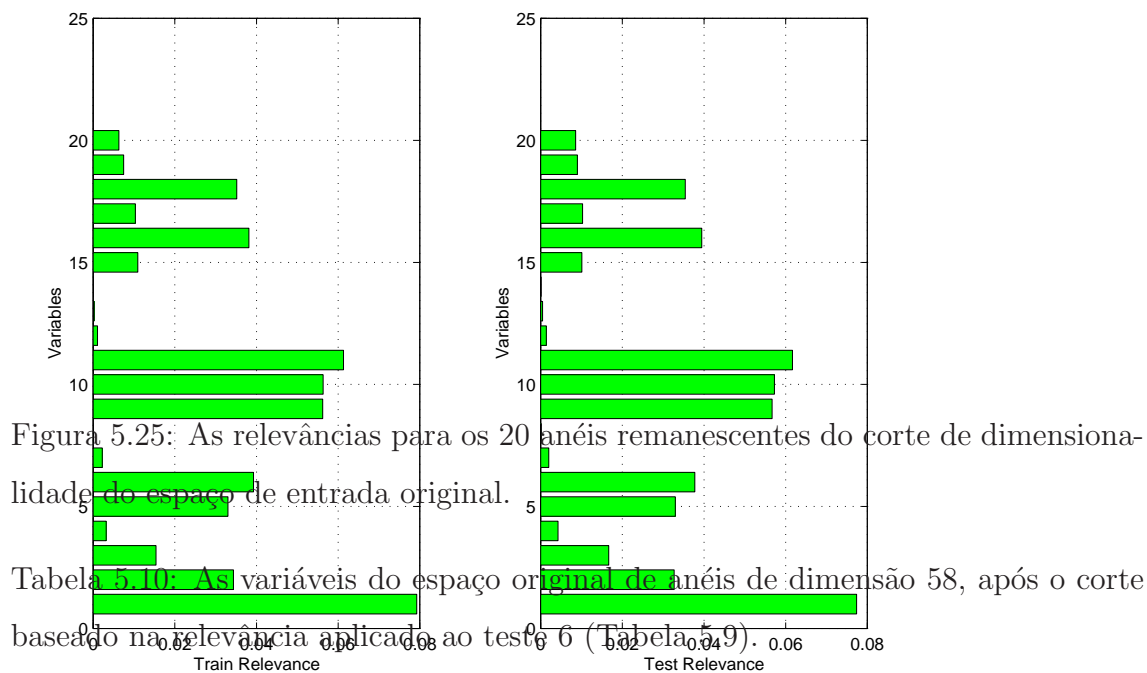


Figura 5.24: Eficiência comparada entre os melhores discriminadores neurais usando os 58 anéis, 20 dos 58 anéis e as quatro quantidades clássicas.

no calorímetro hadrônico são mais relevantes que se pensou inicialmente, quando estudou-se o primeiro gráfico de relevâncias. Isso está de acordo com a física do experimento, já que jatos são bem discriminados quando há parte significativa da energia do objeto na seção hadrônica.

Este gráfico também mostra que algumas das variáveis que são utilizadas são, possivelmente, supérfluas. Alguns testes eliminando diferentes quantidades de variáveis foram realizados. A Tabela 5.11 resume os resultados e configurações desses testes. Esta tabela é um pouco diferente das anteriores, pois inclui a dimensionalidade do vetor de entrada junto com o número de neurônios escondidos na coluna “# *ent-esc*”. Percebe-se que o EMQ mantém-se baixo em todos os testes, mas que a capacidade de discriminação, aqui avaliada pelo produto SP, diminui bastante com a redução da dimensionalidade da entrada. A Tabela 5.10 mostra os anéis que não foram podados e portanto servem de entrada para os discriminadores da Tabela 5.11.

O aumento do número de neurônios na camada intermediária novamente implicou na perda no desempenho da rede. Observa-se isto no teste 2, por exemplo, onde o número de neurônios naquela camada foi aumentado para 6. O EMQ deste



Patamar	# Anéis	Calorímetro	Anéis Restantes
0,0005	17	<i>Presampler</i>	1, 2, 4 e 6
		1ª camada e.m.	7, 8, 9, 11, 12 e 39
		2ª camada e.m.	40, 43, 44, 45 e 46
		1ª camada hadrônica	50 e 51
0,003	15	<i>Presampler</i>	1, 2, 4 e 6
		1ª camada e.m.	7, 8, 11, 12 e 39
		2ª camada e.m.	43, 44, 45 e 46
		1ª camada hadrônica	50 e 51
0,005	14	<i>Presampler</i>	1, 2 e 4
		1ª camada e.m.	7, 8, 11, 12 e 39
		2ª camada e.m.	43, 44, 45 e 46
		1ª camada hadrônica	50 e 51
0,009	12	<i>Presampler</i>	1, 2 e 4
		1ª camada e.m.	7, 8, 11, 12 e 39
		2ª camada e.m.	44, 45 e 46
		1ª camada hadrônica	50
0,02	9	<i>Presampler</i>	1 e 2
		1ª camada e.m.	7, 8, 11, 12 e 39
		2ª camada e.m.	44 e 46

teste é o maior de todos.

Nestes testes, manteve-se a taxa de aprendizado baixa, porque se deseja uma convergência mais suave ao mínimo. Verificou-se em testes anteriores que o treinamento deste discriminador neuronal pode apresentar uma convergência com características fortemente oscilatórias, no caso de valores muito grandes serem utilizadas para a taxa de aprendizado. O momento foi mantido constante em todos os testes e o decaimento da taxa de aprendizado foi variado, quando desejava-se que a convergência durante o treinamento fosse suavizada no final desse período.

De certo, não é possível re-avaliar a relevância das variáveis nos testes de 1 a 14, já que apresentaram um rendimento abaixo do esperado. Esses discriminadores, ainda assim, se mostraram mais eficientes que a solução neuronal utilizando as quatro quantidades clássicas. Decidiu-se por re-avaliar a relevância no teste 15, já que representa a maior capacidade de discriminação aliada ao menor EMQ em todos os testes. Realizou-se testes, mas nenhum apresentou desempenho comparativo aos resultados até agora obtidos.

Corte direto Nesta modalidade sempre se partirá da dimensionalidade inicial do problema, ou seja, 58 (anéis). Espera-se atingir eficiência na discriminação compatível com testes anteriores.

A Tabela 5.12 mostra os resultados obtidos para diversos discriminadores baseados em podas de dimensionalidade do conjunto de relevâncias dos 58 anéis para o teste 17 da Tabela 5.5. Percebe-se que estes testes apresentam desempenho, num geral, melhores que quando utilizou-se a técnica de poda seqüencial. É interessante notar também que os valores de EMQ estão bastante baixos, inclusive em testes com apenas 5 dos 58 anéis de entrada iniciais. É claro, nota-se também um decaimento no desempenho, quando medida através do produto SP, conforme se reduz o número de entradas. A redução do número de entradas para o discriminador oferece entretanto, uma redução exponencialmente maior no tempo de pré-processamento específico do método de discriminação e no tempo de discriminação em si.

A Figura 5.26 mostra um resultado comparativo entre as curvas de eficiência nos testes 1, 9, 17 e 21 e a eficiência do melhor discriminador usando as 58 entradas iniciais (teste 17 da Tabela 5.5). Os testes selecionados da Tabela 5.12 representam os melhores discriminadores do ponto de vista do produto SP quando se reduz o

Tabela 5.11: Resultados obtidos com cortes de dimensionalidade aplicados aos 20 anéis restantes utilizando a técnica de cortes sequenciais.

Teste	μ	α	γ	SP	# Passos	Bat.	Iteração	EMQ	Patamar	% e ⁻	% jatos	Tx.(kHz)	# ent-esc
1	0.030	0.900	0.950	1.60	10000	200	2841	0.2220	0.090	92.08	93.65	1.59	9-4
2	0.050	0.900	0.900	1.60	10000	200	1360	0.3020	0.330	92.08	93.60	1.60	9-6
3	0.050	0.900	0.900	1.61	10000	200	2031	0.2300	0.140	92.08	93.93	1.52	9-3
4	0.050	0.900	0.900	1.66	5000	200	4032	0.2090	0.060	94.72	93.27	1.68	12-5
5	0.030	0.900	0.950	1.66	10000	200	6633	0.2180	0.100	94.72	93.27	1.68	12-5
6	0.050	0.900	0.900	1.66	10000	200	3909	0.2050	0.020	94.72	93.21	1.70	12-5
7	0.050	0.900	0.900	1.65	10000	200	1843	0.2170	-0.060	96.04	91.50	2.12	12-4
8	0.050	0.900	0.900	1.66	10000	200	2851	0.1880	0.030	93.73	94.26	1.43	12-3
9	0.020	0.900	0.900	1.63	10000	200	5552	0.2210	-0.010	95.05	91.83	2.04	12-3
10	0.020	0.900	0.900	1.66	10000	200	9999	0.2270	0.030	96.04	92.05	1.99	14-5
11	0.020	0.900	0.900	1.67	10000	200	5321	0.2350	0.120	95.71	92.66	1.83	14-4
12	0.020	0.900	0.900	1.67	10000	200	4636	0.2130	0.050	95.38	92.94	1.77	14-3
13	0.020	0.900	0.900	1.65	10000	200	4506	0.2290	0.190	93.73	93.93	1.52	15-5
14	0.020	0.900	0.900	1.66	10000	200	6106	0.2210	0.020	95.71	92.27	1.93	15-4
15	0.020	0.900	0.900	1.71	10000	200	5980	0.1920	0.110	95.38	94.43	1.39	17-4
16	0.020	0.900	0.900	1.70	10000	200	2269	0.1900	0.050	95.38	94.09	1.48	17-3
17	0.050	0.900	0.800	1.71	10000	100	995	0.2140	0.210	95.38	94.32	1.42	17-4

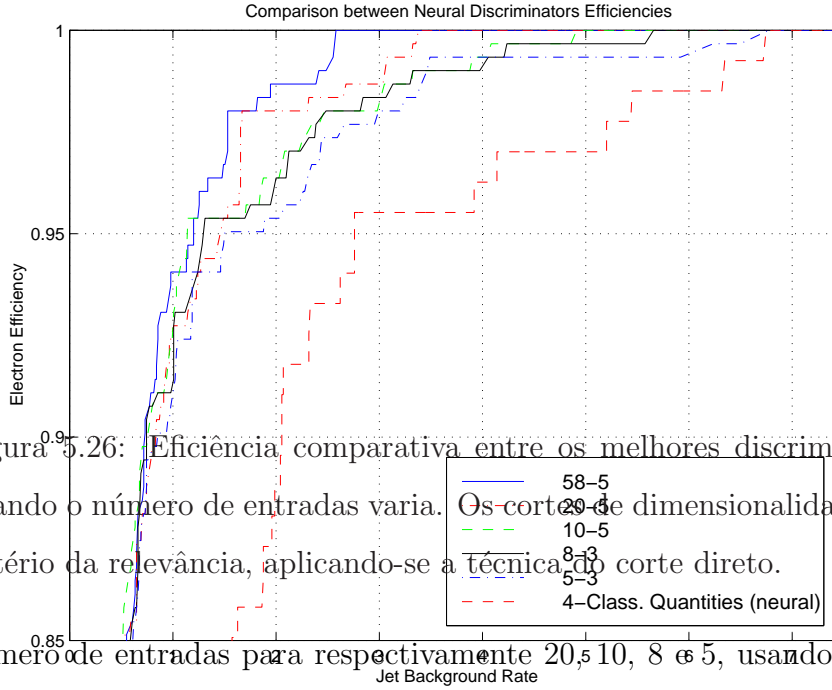


Figura 5.26: Eficiência comparativa entre os melhores discriminadores neurais quando o número de entradas varia. Os cortes de dimensionalidade são baseados no critério da relevância, aplicando-se a técnica do corte direto.

número de entradas para respectivamente 20, 10, 8 e 5, usando o critério de poda por relevância. Os anéis preservados nestes testes estão enumerados na Tabela 5.13. Nesta figura, observa-se que até um nível de eficiência de classificação de elétrons de 95%, todos os discriminadores baseados em anéis apresentam comportamento similar, superando em **muito** o resultado baseado nas quantidades clássicas. Para valores mais altos de eficiência na discriminação de elétrons, algumas das soluções usando anéis perdem bastante a capacidade de discriminação de jatos em relação às soluções anteriores. A configuração com 20 anéis de entrada, no entanto, mantém-se acima de todas as outras curvas, excetuando-se a primeira, que se utiliza dos 58 anéis de entrada.

Tabela 5.12: Resultados obtidos com diversos cortes de dimensionalidade aplicados aos 58 anéis utilizando a técnica de corte direto.

Teste	μ	α	γ	SP	# Passos	Bat.	Iteração	EMQ	Patamar	% e ⁻	% jatos	Tx.(kHz)	# ent-esc
1	0.010	0.990	0.950	1.75	1000000	200	949800	0.1720	-0.170	98.02	93.32	1.67	20-5
2	0.010	0.900	0.950	1.74	1000000	200	592600	0.1670	0.030	96.04	94.92	1.27	20-5
3	0.010	0.900	0.950	1.73	1000000	200	805000	0.2030	0.150	96.37	94.26	1.43	20-6
4	0.010	0.900	0.950	1.73	1000000	200	971400	0.1760	0.150	95.38	95.09	1.23	20-4
5	0.020	0.900	0.950	1.73	1000000	200	698800	0.1630	0.110	95.38	95.14	1.21	20-4
6	0.020	0.900	0.950	1.73	1000000	200	725600	0.1490	0.030	95.38	95.25	1.19	20-4
7	0.020	0.900	0.900	1.73	1000000	200	930600	0.1840	0.230	95.38	95.20	1.20	20-4
8	0.020	0.900	0.950	1.72	1000000	200	999600	0.1740	0.070	95.71	94.43	1.39	20-3
9	0.050	0.900	0.800	1.74	1000000	100	80700	0.2160	0.310	95.38	95.42	1.15	10-5
10	0.050	0.900	0.800	1.73	1000000	100	84200	0.2120	0.290	95.38	95.36	1.16	10-4
11	0.050	0.900	0.800	1.73	1000000	100	72400	0.2790	0.460	95.38	95.25	1.19	10-3
12	0.050	0.900	0.700	1.73	500000	100	96500	0.2000	0.250	95.38	95.20	1.20	10-3
13	0.050	0.900	0.700	1.73	500000	200	162400	0.1680	0.070	95.38	95.25	1.19	10-3
14	0.050	0.900	0.900	1.73	500000	500	338500	0.1690	0.050	95.71	94.92	1.27	10-3
15	0.050	0.900	0.700	1.73	500000	500	355000	0.2160	0.250	95.38	95.09	1.23	10-3
16	0.050	0.900	0.950	1.73	1000000	1000	706000	0.1850	0.160	95.38	95.09	1.23	10-3
17	0.050	0.900	0.950	1.72	1000000	1000	553000	0.2350	0.310	95.38	94.76	1.31	8-3
18	0.050	0.900	0.950	1.71	1000000	1000	1000000	0.1960	0.020	96.70	92.99	1.75	8-4
19	0.050	0.900	0.950	1.69	1000000	1000	968000	0.2150	0.200	94.72	94.43	1.39	5-4
20	0.050	0.900	0.950	1.69	1000000	1000	983000	0.2600	0.380	94.39	94.70	1.32	5-3
21	0.050	0.900	0.950	1.70	1000000	500	965000	0.1680	0.090	94.06	95.25	1.19	5-3
22	0.050	0.900	0.950	1.70	2000000	500	1421500	0.1840	0.210	94.06	95.36	1.16	5-3

Tabela 5.13: As variáveis do espaço original de anéis de dimensão 58, após vários cortes baseados na relevância.

Patamar	# Anéis	Calorímetro	Anéis Restantes
0,001	20	<i>Presampler</i>	1, 2, 4 e 6
		1 ^a camada e.m.	7, 8, 9, 10, 11, 12 e 39
		2 ^a camada e.m.	40, 41, 42, 43, 44, 45 e 46
		1 ^a camada hadrônica	50 e 51
0,003	10	<i>Presampler</i>	1, 2 e 6
		1 ^a camada e.m.	7, 8 e 39
		2 ^a camada e.m.	40, 41 e 42
		1 ^a camada hadrônica	50
0,005	8	<i>Presampler</i>	1 e 2
		1 ^a camada e.m.	7, 8 e 39
		2 ^a camada e.m.	40, 41 e 42
0,05	5	<i>Presampler</i>	1
		1 ^a camada e.m.	7 e 8
		2 ^a camada e.m.	40 e 41

Capítulo 6

Implementação do Decisor Neuronal Local

Durante o processamento local no L2, é possível se distinguir as seguintes fases, como discutido no Capítulo 3:

Aquisição de Dados Nesta fase do processamento local interessa-se em agrupar as informações contidas nos diversos ROBs, de forma a poder reconstruir a RoI de interesse;

Pré-processamento Os dados recolhidos na fase de aquisição são processados nesta fase. Este pré-processamento tem por objetivo:

- Supressão de dados excedentes (discutido na Seção 3.5.2);
- Verificação da integridade da RoI e tratamento de dados faltantes (discutido na Seção 3.5.3);
- Pré-processamento específico para o discriminador, ou seja, formatar os dados que se alimente o discriminador elétron/jato ;

Discriminação Após os pré-processamento, aplica-se o sistema de discriminação adequado à deteção do objeto, como definido no Capítulo 5;

Extração de Características Nesta fase interessa-se em extrair as quantidades físicas que definem o objeto, tais como energia transversa, momento, isolamento, etc. Estas características serão utilizados pelo L2 para definir os eventos que serão re-passados ao L3.

Neste capítulo se introduz alguns detalhes de implementação de um decisor local para objetos e.m., utilizando-se das técnicas de discriminação neuronal introduzidas no Capítulo 5. Apresenta-se também resultados de testes de desempenho, acuidade e tempo de resposta.

O projeto foi dividido em duas tarefas distintas: o projeto de um pré-processador para os dados do calorímetro, que terá a responsabilidade de conduzir todas as etapas de pré-processamento descritas anteriormente, e um discriminador de objetos e.m., neural, que tem por objetivo a separação elétron/jato. Ao final, une-se as duas aplicações em uma implementação para um processador digital de sinais (DSP) da família 21k da Analog Devices, formando um decisor local completo.

A extração de características não é abordada neste trabalho, por se tratar de um procedimento padronizado de processamento e onde maiores otimizações não serão necessárias. Para maiores referências, ao leitor é indicado [11]. Ainda assim, esta implementação traz uma vantagem em relação aos demais decisores locais implementados, já que é a mais completa. Espera-se que este trabalho seja utilizado como base de comparação no futuro para sistemas que integrem toda a funcionalidade do decisor local para objetos e.m. no segundo nível de filtragem do ATLAS.

Durante os 10 anos de funcionamento a que se destina o experimento ATLAS, será necessário que não somente, mas principalmente, o sistema de filtragem sofra alterações tanto a nível de manutenção quanto à inclusão e exclusão de procedimentos, ou seja, novas técnicas de filtragem, seleção de eventos, etc. Desta forma, o sistema de filtragem, deverá ser construído de forma que seja robusto e flexível a mudanças, o que dificilmente é atingido quando se utiliza *hardware* de baixa programabilidade. Infelizmente, para L1, os requisitos de tempo são mais importantes que a flexibilidade do sistema, o que obriga os projetistas deste sub-sistema a implementarem-no usando recursos menos flexíveis. No entanto, em níveis de filtragem mais altos, como no L2, os requisitos de tempo igualam-se à flexibilidade desejada e portanto, computadores multi-processados e linguagens populares de programação serão utilizados em sua implementação. Em específico, C e C++ foram as linguagens de programação escolhidas tanto para a implementação do L2 quanto do L3. Para a implementação do decisor local neste trabalho, utilizou-se a linguagem de programação C. Esta implementação, embora não esteja totalmente integrada no ambiente do L2,

respeita todas as suas limitações e utiliza todas as suas vantagens. Acredita-se que com um esforço mínimo, as rotinas de filtragem sejam portadas para um ambiente 100% compatível com o *Software* de Referência. Atualmente, o sistema de filtragem aqui proposto roda *stand-alone*.

6.1 Pré-processamento de dados

Nesta fase interessa-se na aquisição e tratamento dos dados de forma que possam ser encaminhados ao discriminador. Adotou-se o pré-processamento em anéis como já discutido. A normalização das somas em anel a ser utilizada aqui é a que apresentou os melhores resultados no Capítulo 5, i.e., usando toda a energia contida no objeto. A seguir, as etapas envolvidas nesta fase de pré-processamento são discutidas com maior detalhamento.

6.1.1 Aquisição de Dados

O *Software* de Referência (Capítulo 4) emula as condições reais do segundo nível do sistema de filtragem do ATLAS como foi visto. Isto acontece pois não há condições de se construir um sistema *on-line*, já que os detetores ATLAS ainda vêm sofrendo alterações e adaptações.

Assim sendo, a forma atual de aquisição de dados para este ambiente emulado do L2, é a leitura de bancos de dados contendo eventos simulados e gravados em mídia permanente, como discos rígidos. Para a consulta do banco de dados, utilizou-se uma biblioteca especial, desenvolvida pelo time do L2, com este propósito.

Esta etapa é uma das partes do decisor local que terá que ser adaptada, caso o sistema seja integrado no *Software* de Referência. Para isto, o projetista poderá utilizar a classe `CaloEMLocalProcessor` transplantando o algoritmo de discriminação para dentro do objeto `_discriminator` e utilizando o objeto `_collector` para a aquisição de dados automática, como descrito na Implementação 4.3 no Capítulo 4.

Supressão de dados Nos arquivos de dados, os eventos e RoI's associadas não contém informação que excede ao seu domínio. Em outras palavras, somente os dados do evento estão presentes nos arquivos de dados, não havendo a necessidade

da supressão de regiões indesejadas, como colocado na Seção 3.5.2.

6.1.2 Verificação de Integridade da RoI

Após a aquisição, é necessário verificar se os dados de todo o domínio da RoI estão presentes. Caso não estejam, opta-se por substituir os dados por seus valores médios (normalizados), substituir a região faltante com zeros ou ainda, para efeitos de estudo, eliminar a RoI de nosso conjunto de dados.

O pré-processador que foi construído é bastante rígido neste aspecto: caso não encontre o número de células que espera, não pré-processa a RoI. Além disso, o processamento em anéis exige que haja uma granularidade constante na camada. Sabe-se, por outro lado, que os calorímetros do ATLAS, em sua configuração atual, possuem granularidade variante, não somente camada a camada, mas também dentro de uma mesma camada.

Isto nos coloca um segundo problema: como garantir granularidade contínua? Uma solução simples seria interpolar bi-dimensionalmente cada camada da RoI, de tal forma que a granularidade camada a camada permaneça constante e igualmente fina. Por outro lado, este pode ser um processo bastante trabalhoso do ponto de vista do decisor local, sendo uma solução mais prática a decimação da RoI, até que a granularidade seja constante e menos fina naquela camada.

A Figura 6.1 exemplifica os processos de interpolação e decimação descritos. Na figura, à esquerda é possível ver um esboço das células de uma camada. Percebe-se que à direita da camada há uma descontinuidade de granularidade, que passa a ser menos fina. As regiões modificadas que estão representadas à direita da Figura 6.1 mostram como ficariam as camadas após o processo de interpolação (*up-sampling*) e decimação (*down-sampling*). Na primeira possibilidade, a interpolação, regiões com maior granularidade seriam preservadas, enquanto que regiões menos granulares seriam re-dimensionadas para que sua granularidade seja igual a da primeira. Assim, por exemplo, se uma célula originará, após a interpolação, duas novas, divide-se a energia daquela célula por dois, para representar a energia de cada uma das novas células criadas. Neste caso, aplica-se tantas divisões quanto forem necessárias para a uniformização da granularidade na camada. Na decimação é possível se realizar o processo inverso. É possível empregar técnicas tradicionais para este processo [36],

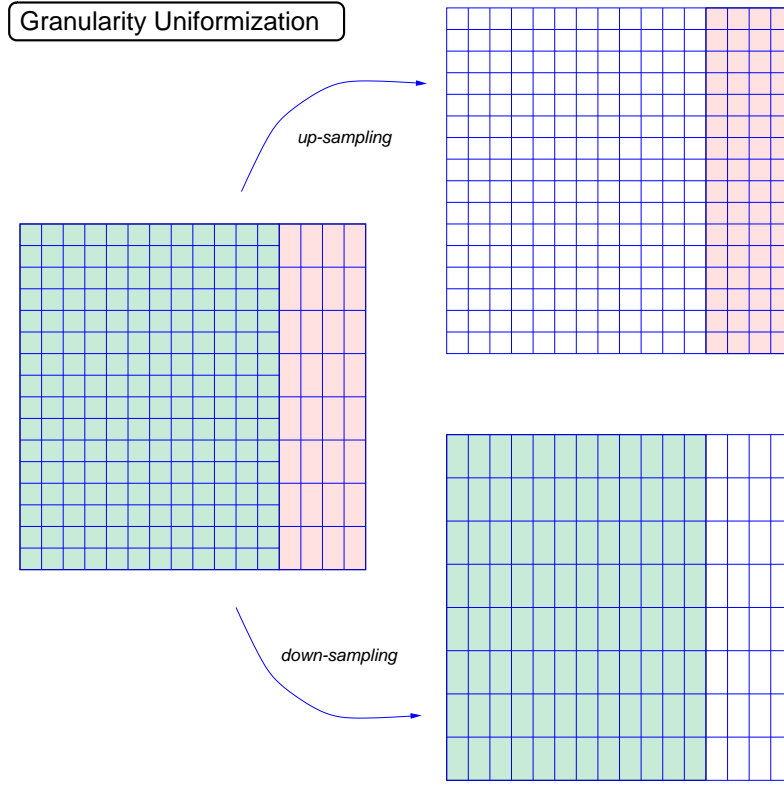


Figura 6.1: Interpolação e decimação bi-dimensional da RoI para ajuste de granularidade.

como uma pré-filtragem por um passa-baixas para evitar *aliasing*, ou simplesmente dividir a energia da célula pelo número de células na nova granularidade. Ou seja, as energias das células que serão agrupadas através da decimação, em uma única, são, então, somados.

Há vantagens e desvantagens em ambos os enfoques. No primeiro (interpolação) aplica-se divisões, cujo o processamento é notoriamente lento, e no segundo, perde-se a informação de granularidade. Este estudo deve ser conduzido de forma criteriosa e levando-se em consideração o desempenho do discriminador, tanto em tempo quanto eficiência de separação das classes.

Neste trabalho, limitou-se o domínio de estudo a RoI's que têm seus centros com $|\eta| < 1,1$. Dentro desta região não há mudanças de granularidade que demandem processamento específico. Por outro lado, por ser um estudo comparativo, os resultados não sofrerão tendências, já que o mesmo conjunto de dados foi aplicado a ambos, ao processamento clássico (usando as quatro variáveis introduzidas no Capítulo 5) e ao processamento baseado em anéis.

De fato, mostra-se em [11] que há uma piora de algumas unidades percentuais no algoritmo clássico, quando aplicado à regiões de interesse cujo o centro está em regiões nos quais $|\eta|$ é maior que 1,1. Sendo assim, compara-se estes estudos com o máximo desempenho do algoritmo clássico. Estudos posteriores, entretanto, deverão considerar o desempenho para todas as possíveis RoI's.

6.1.3 Pré-processamento específico

Nesta fase, interessa-se por gerar os anéis a partir das células da RoI. Aproveitando o estudo de relevâncias, é possível avaliar o tempo de processamento para diferentes quantidades de anéis. Estes testes investigarão a eficácia do algoritmo, verificando o impacto no tempo de processamento no requerido para o pré-processamento.

O número de anéis é, de fato, uma variável importante no processo de discriminação, o processo de normalização, é aplicado diretamente aos anéis formados. Neste caso, possuindo-se 58 anéis, têm-se que normalizar 58 variáveis. A redução no número de quantidades a serem normalizadas irá, portanto, diminuir o tempo de processamento.

As rotinas implementadas possuem um dispositivo que permite que o número de anéis e camadas que é utilizado no pré-processamento da região de interesse seja controlado, de forma a produzir um pré-processador mais compacto ou mais extenso, reduzindo e aumentando o tempo de pré-processamento específico, respectivamente.

6.1.4 Resultados da implementação

Inicialmente, implementou-se um sistema de pré-processamento *off-line* utilizando os recursos listados. Esta implementação servirá como ponte para uma implementação em *tempo real*, como será visto mais a frente.

Através de um *profiler*¹, é possível entender a percentagem do tempo gasto em cada atividade do pré-processamento e estimar, com os cortes baseados na relevância, o tempo de processamento final da aplicação.

¹*Profiler* é um programa que permite que um arquivo de dados especial, produzido após a execução de outro programa, possa ser interpretado, produzindo os intervalos de tempo (relativos e absolutos) que cada sub-rotina ocupou no tempo de processamento da aplicação.

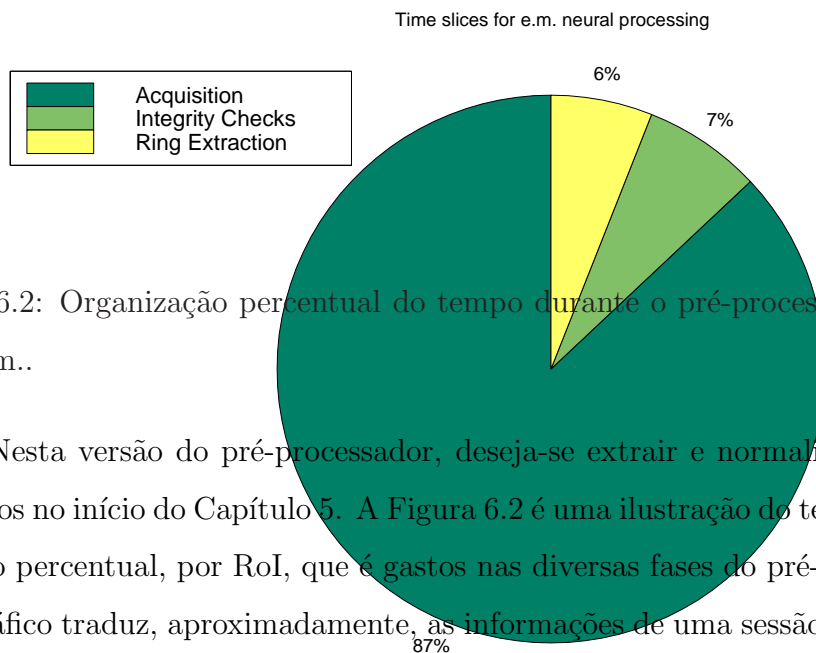


Figura 6.2: Organização percentual do tempo durante o pré-processamento de objetos e.m..

Nesta versão do pré-processador, deseja-se extrair e normalizar os 58 anéis propostos no início do Capítulo 5. A Figura 6.2 é uma ilustração do tempo de processamento percentual, por RoI, que é gastos nas diversas fases do pré-processamento. Este gráfico traduz, aproximadamente, as informações de uma sessão de *profiling* no pré-processador, tendo sido compilado tanto com e sem otimização. Neste gráfico, é possível ver que a maior parte do tempo de processamento está na aquisição das células. O motivo do esforço computacional advém do fato que as células colhidas dos ROB's (aqui emulados pela consulta ao banco de dados) são enviadas fora de ordem e codificadas. O sub-sistema de aquisição deve organizar as células de acordo com sua posição na RoI antes, ou paralelamente a verificação da integridade dentro da região de interesse. É interessante notar que apenas 6% do tempo é, de fato, gasto com a extração de anéis.

O tempo médio de processamento encontrado para esta parte do processador local, em um PC, com processador Pentium compatível e 450 MHz de *clock* é de 13 milissegundos por RoI². Porém, o sistema de pré-processamento construído possui um conjunto de opções que podem ser mudadas a cada vez que é executado. Estas

²Estes resultados advém de um código otimizado pelo compilador utilizado (GNU)

opções são utilizadas para modificar o tipo de saída, normalização e a forma como a checagem de integridade é conduzida pelo programa. Estes recursos introduzem uma latência que prejudica a eficiência no tempo.

6.2 Discriminação neuronal de objetos e.m.

A fase seguinte ao pré-processamento é a discriminação dos objetos que foram adquiridos pelo sistema de pré-processamento. Esta tarefa é simples e consiste basicamente de uma rede neural, com 58 entradas e uma saída. A rede deverá ser totalmente conectada, como apontam os estudos do Capítulo 5. Para a implementação que será aqui estudada, utilizar-se-á o projeto obtido no teste 17 (5 neurônios na camada escondida), cuja configuração e resultados estão na Tabela 5.5.

Implementou-se este sub-sistema utilizando-se da linguagem de programação C, como no caso do pré-processador. Aqui, além do tempo de desempenho, se está preocupado também com a precisão da saída, comparando-se o resultado da implementação com aqueles da simulação *off-line*. Em termos da função de ativação dos neurônios, diferentes realizações podem ser testadas:

1. Implementação através da diretiva `tanh()` nativa do sistema, com 64 bits de precisão;
2. Implementação através da diretiva `tanh()` nativa do sistema, com 32 bits de precisão;
3. Implementação por tabela (*Look-up Table*). Nesta versão, os limites da tabela e a precisão da conversão serão variáveis de interesse.

Tomando por base a implementação descrita no item 1, pode-se avaliar o tempo de pré-processamento em função da qualidade de separação alcançada através de todas as técnicas em estudo. É possível também avaliar o impacto no tempo de processamento, mudando-se a configuração da rede e as opções de compilação do código.

6.2.1 Implementação da ativação neuronal

Ao construir um sistema neuronal, que acarreta em um grande número de operações e acumulações, deve-se preocupar com a precisão e velocidade de cada operação. Na codificação em C, dois tipos de variáveis operam sobre números reais: `float` e `double`. Utilizando um sistema GNU/Linux em computadores compatíveis com a linha Intel 386, `float`'s têm 32 bits de precisão, enquanto que `double`'s, 64 bits. Com relação aos tipos de operações, é possível também definir duas possibilidades: utilizar as funções da biblioteca (*built-in*) ou um procura em tabela (do inglês, *Look-up Table*), implementada diretamente no programa. Com excessão da ativação neuronal, todas as outras funções serão programadas com componentes da biblioteca.

No caso de optar-se pela implementação da tangente hiperbólica por meio de tabelas de procura (TP), deve-se controlar duas variáveis: o fim da TP e a sua resolução. O fim da TP marcará o valor do último argumento com o qual a função retornará um valor de sua tabela interna, indicando, assim, a saturação da função tangente hiperbólica.

6.2.1.1 Acuidade

A Tabela 6.1 mostra o erro médio quadrático referido a implementação 1 (lógica de 64 bits/função de ativação interna à biblioteca), quando utiliza as várias possibilidades averiguadas para a ativação dos neurônios de uma rede neuronal com 58 nós de entrada, 5 nós na camada intermediária e apenas um nó na saída. Cada vez que propaga-se a entrada até a saída, a rede executará $58 * 5 = 290$ multiplicações da entrada até a camada escondida da rede, seguindo-se de 5 somatórios, cada um com 58 entradas. O resultado dos 5 somatórios são ativados de acordo com o método indicado nesta tabela. A transmissão sináptica da camada escondida para a camada de saída acontecerá de forma similar. No total, utiliza-se 6 ativações, 295 multiplicações e 289 somas. Este representa o pior caso que será enfrentado em termos de complexidade, pois todas as demais configurações de redes neuronais (que se destacaram pelo desempenho no Capítulo 5) requerem um menor número de operações.

Nos testes realizados não observou-se nenhuma mudança do desempenho

Tabela 6.1: Erro Médio Quadrático por tipo de variável/ativação. Estes testes utilizam uma rede neuronal 58-5-1.

Variáveis	Ativação	Fim da TP	Resolução	EMQ
64 bits	nativa	-	-	0
64 bits	TP	6.0	10^{-2}	$(5, 28 \pm 0, 23) \times 10^{-5}$
64 bits	TP	6.0	10^{-3}	$(5, 30 \pm 0, 23) \times 10^{-7}$
64 bits	TP	6.0	10^{-4}	$(5, 14 \pm 0, 22) \times 10^{-9}$
64 bits	TP	4.0	10^{-2}	$(5, 28 \pm 0, 23) \times 10^{-5}$
64 bits	TP	4.0	10^{-3}	$(5, 46 \pm 0, 24) \times 10^{-7}$
64 bits	TP	4.0	10^{-4}	$(2, 10 \pm 0, 09) \times 10^{-8}$
64 bits	TP	2.0	10^{-2}	$(1, 28 \pm 0, 05) \times 10^{-4}$
64 bits	TP	2.0	10^{-3}	$(7, 28 \pm 0, 31) \times 10^{-5}$
64 bits	TP	2.0	10^{-4}	$(6, 98 \pm 0, 29) \times 10^{-5}$
32 bits	nativa	-	-	$(4, 71 \pm 0, 20) \times 10^{-14}$
32 bits	TP	6.0	10^{-2}	$(5, 28 \pm 0, 23) \times 10^{-5}$
32 bits	TP	6.0	10^{-3}	$(5, 30 \pm 0, 23) \times 10^{-7}$
32 bits	TP	6.0	10^{-4}	$(5, 14 \pm 0, 22) \times 10^{-9}$
32 bits	TP	4.0	10^{-2}	$(5, 28 \pm 0, 23) \times 10^{-5}$
32 bits	TP	4.0	10^{-3}	$(5, 46 \pm 0, 24) \times 10^{-7}$
32 bits	TP	4.0	10^{-4}	$(2, 10 \pm 0, 09) \times 10^{-8}$
32 bits	TP	2.0	10^{-2}	$(1, 25 \pm 0, 05) \times 10^{-4}$
32 bits	TP	2.0	10^{-3}	$(7, 28 \pm 0, 31) \times 10^{-5}$
32 bits	TP	2.0	10^{-4}	$(6, 98 \pm 0, 29) \times 10^{-5}$

na discriminação de elétrons e jatos utilizando nas configurações indicadas na Tabela 6.1. Ou seja, independente da resolução e forma de implementação, o sistema final discrimina com a mesma eficiência elétrons e jatos. Utilizou-se uma patamar fixo em 0, para estes testes, reconhecendo elétrons quando a saída da rede neuronal for menor do que zero.

A Figura 6.3 mostra o erro na saída da rede quando compara-se a saída da implementação em 64 bits e usando a função `tanh()` nativa ao sistema e a implementação em 32 bits, usando a tabela de procura com resolução 0,01 e final em 2. Como verifica-se na Figura 6.3, o maior erro obtido está abaixo de 0,05, o que concorda com o esperado³.

A Figura 6.4 mostra um gráfico de dispersão (*scatter plot*) comparando-se as duas saídas mencionadas. Na legenda na parte superior esquerda da figura, é possível ver as componentes do *fitting* linear (usando o método dos mínimos quadrados) para aquela distribuição de pontos. Como é possível notar, a aproximação é razoavelmente boa mesmo com a baixa resolução da tabela. Nessa figura também é possível distinguir que os elementos cujas saídas estão próximas a zero estão mais dispersos, confirmando a questão do erro ser maior para ativações próximas a zero.

6.2.1.2 Tempo de processamento

O tempo de processamento para cada conjunto de 58 anéis (ou seja, para cada RoI) varia com o nível de otimização e precisão utilizada nos testes. O tempos de execução para cada RoI (em um Pentium II @ 300 MHz) estão listados na tabela 6.2. Não há distinções entre as diversas configurações de TP, já que o esforço computacional indifere neste caso. Os valores da tabela foram obtidos medindo-se

³O maior erro cometido para cada neurônio, ou seja, para cada ativação está localizado próximo a zero, pois é onde a função tangente hiperbólica possui maior taxa de variação (derivada = 1). Um erro de 0,01 nas proximidades da origem, portanto, traduz-se num erro de 0,01 na ativação. Se todos os neurônios tiverem suas ativações próximas a zero, então o erro no evento considerado será aproximadamente $6 \times 0,01 = 0,06$. Este valor é aproximado pois não considera-se o erro devido ao truncamento na acumulação de um número real em uma variável de 32-bits (que varia com a mantissa do número sendo armazenado e cujo máximo é $1,2 \times 10^{-6}$ quando a mantissa se aproxima de 10), que é pelo menos 4 ordens de grandeza menor que o erro cometido usando-se uma tabela com 0,01 de resolução.

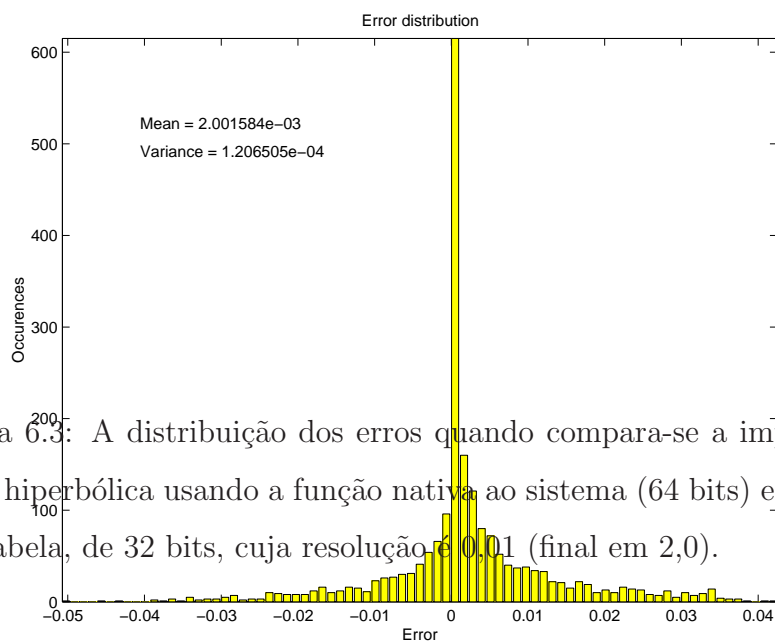


Figura 6.3: A distribuição dos erros quando compara-se a implementação da tangente hiperbólica usando a função nativa ao sistema (64 bits) e uma implementação por tabela, de 32 bits, cuja resolução é 0,01 (final em 2,0).

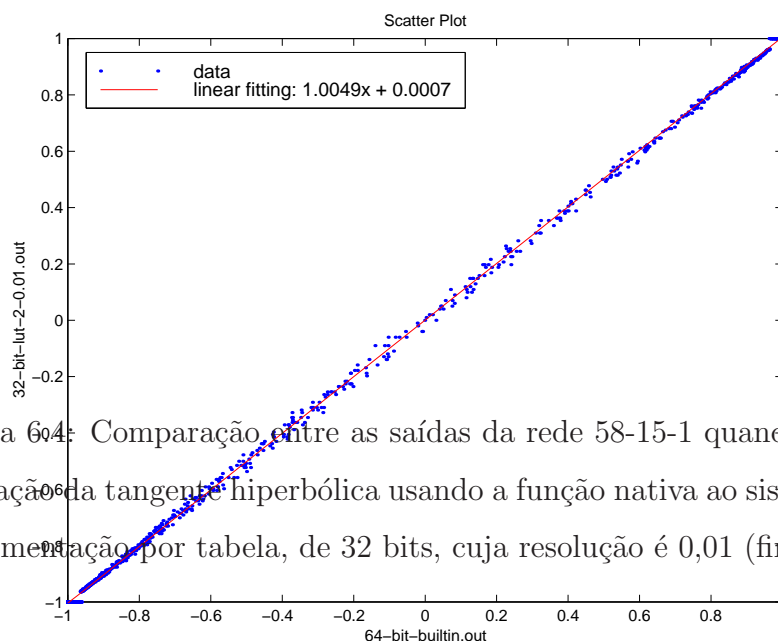


Figura 6.4: Comparação entre as saídas da rede 58-15-1 quando utiliza-se a implementação da tangente hiperbólica usando a função nativa ao sistema (64 bits) e uma implementação por tabela, de 32 bits, cuja resolução é 0,01 (final em 2,0).

o tempo de processamento sobre um total de cem mil iterações e computando-se a média. Nesse caso, a medida de tempo é feita para cada conjunto de 100.000 iterações já que os recursos no PC impossibilitam uma medida precisa de tempo para cada evento individualmente. O erro na medida é dado em função da resolução do cronômetro ($1\mu s$) do número de eventos: $\epsilon = 1\mu s / \sqrt{100000} \approx 3,16ns$. A tabela também contém a descrição dos *flags* de otimização utilizados durante a compilação.

Na Tabela 6.2 nota-se que a diferença entre os diversos tempos é bastante grande quando se aplica a otimização automática (via compilador) do código, pois o tempo de processamento, neste caso, cai bruscamente. Não há diferenças marcantes entre as implementações utilizando tabelas de procura ou as funções padrões da biblioteca, quando compara-se o mesmo nível de otimização e precisão. Isto se deve ao fato de que cerca de 92% do tempo é gasto com a propagação pela rede. Uma análise de uma sessão de *profiling* do discriminador, no entanto, revela que cerca de 6% do tempo é gasto com a ativação quando utiliza-se tabelas de procura e **apenas** 3% do tempo de processamento é gasto com ativação quando utiliza-se as funções nativas do sistema. Das atividades realizadas durante a procura na tabela, 50% do tempo é gasto com a procura da posição correta na tabela⁴, que resulta na melhor aproximação para o argumento dado. Vale notar, também, que utilizando uma codificação com lógica de 64 bits e código otimizado, o sistema é mais rápido que utilizando 32 bits para o mesmo nível de otimização. Possivelmente, isso se deve às características de implementação do compilador para o processador que foi utilizado nos testes.

6.3 Implementação em DSP's

Nesta seção, descreve-se a implementação em linguagem C no ambiente de um processador digital de sinais (DSP) de 32 bits e com 25 nanossegundos de ciclo, o ADSP-21062 da Analog Devices (veja detalhes deste processador no Apêndice D). Este DSP pode ser programado usando-se a linguagem C, mas possui algumas li-

⁴Isto é feito quantizando-se o argumento da função com respeito a resolução da tabela sendo utilizada. A posição correta na tabela (índice) é considerada a que está mais próxima do valor quantizado.

Tabela 6.2: Tempo de processamento médio para cada RoI usando uma rede 58-5-1.

Otimização	Flags	Precisão	Ativação	Tempo
Não	-g -pg	32-bits	nativa	46,12 μ s
Não	-g -pg	32-bits	TP	46,83 μ s
Não	-g -pg	64-bits	nativa	55,29 μ s
Não	-g -pg	64-bits	TP	55,06 μ s
Sim	-O3	32-bits	nativa	23,79 μ s
Sim	-O3	32-bits	TP	25,68 μ s
Sim	-O3	64-bits	nativa	17,69 μ s
Sim	-O3	64-bits	TP	19,82 μ s

mitações, por exemplo, o suporte limitado a variáveis reais.

Por possuírem um suporte limitado à tarefas mais genéricas de processamento, mas grande capacidade de processamento paralelo, os DSP's podem ser alocados como processadores auxiliares aos nós de processamento principal do L2. Neste caso, deseja-se utilizá-los na tarefa de pré-processamento e discriminação de objetos, utilizando os calorímetros do ATLAS e as técnicas descritas no Capítulo 5. Algumas das vantagens de se usar DSP's como co-processadores para a discriminação elétron/jato são:

1. Os processadores principais ficariam para tarefas menos complexas, que não coubessem aos DSP's, como controle de fluxo e algoritmos mais simples e sequenciais;
2. Os DSP's são, em geral, mais baratos e consomem menos energia que os processadores convencionais. Isto reduziria a carga sobre estes dispositivos, aumentando sua vida útil e reduzindo o consumo geral;
3. DSP's são especializados em operações do tipo S.I.M.D (*Single Instruction, Multiple Data*), vitais nos algoritmos do segundo nível; Neste caso, sendo também mais rápidos que processadores convencionais;
4. DSP's são programáveis em C, o que simplifica o desenvolvimento e manutenção dos programas desenvolvidos no contexto do segundo nível de filtragem,

já produzido em C/C++;

5. DSP's são especializados em produtos internos, o que é muito conveniente para a implementação do discriminador neural,

No sistema de desenvolvimento que se trabalhou, o DSP está alojado em uma placa conectada ao sistema hospedeiro (no nosso caso um PC comum, rodando Windows) através de uma interface ISA. Nesta configuração, o DSP não possui contato com os recursos de entrada e saída do sistema hospedeiro, como os discos rígidos e a tela que são necessários para a emulação do decisor local, havendo a necessidade de se dividir a aplicação em 2 sub-sistemas especializados. O primeiro realiza o discriminador em si, rodando no DSP. O segundo será uma aplicação que roda no sistema hospedeiro, com o intuito de prover à aplicação no DSP, os recursos necessários à simulação.

6.3.1 O hospedeiro

A comunicação entre as aplicações é feita através de acesso direto à memória do DSP, pela aplicação hospedeira. Cuidados extras devem ser tomados para que os recursos compartilhados entre o DSP e o sistema hospedeiro (neste caso a memória) não se tornem um motivo para o mal-funcionamento do sistema, gerando travamentos (*dead-locks*) ou resultados inesperados.

Durante a execução, o hospedeiro verifica se o DSP encontrou problemas em sua fase de inicialização, finalização ou operação, de forma automática, reportando ao operador da simulação o problema encontrado, assim como alguns detalhes que levaram a aplicação ao erro. Desta forma, embora não desfrute dos recursos plenos do hospedeiro, o DSP poderá se comunicar com o operador.

O hospedeiro também tem a função de cronometrar o tempo necessário para o processamento de cada RoI que é entregue ao DSP. Estas medidas darão bases para concluir sobre a velocidade de processamento neste tipo de implementação. Em resumo, o hospedeiro executará os seguintes passos:

1. Inicializa o DSP;
2. Carrega a aplicação no DSP;

3. Abre os arquivos de entrada e saída no hospedeiro;
4. Configura o discriminador neuronal no DSP;
5. Inicializa a aplicação no DSP;
6. Lê todos eventos nos arquivos de entrada e envia-os para o DSP, um-a-um e recebe sua saída e o tempo que o DSP levou para executá-la, guardando no arquivo de saída todos os dados;
7. Termina todas as variáveis do programa;
8. Termina a aplicação no DSP;

6.3.2 A aplicação no DSP

Desenvolveu-se uma versão do sistema de pré-processamento dos dados de RoI's do calorímetro do ATLAS para o ADSP-21062 citado acima. Esta versão do pré-processador pode executar:

1. Decodificação das células - a decodificação é feita basicamente por controle de fluxo;
2. Organização das células em *Trigger Towers* - nesta fase organiza-se os eventos recebidos segundo sua granularidade no ROB. Desta forma, células faltantes podem ser identificadas tão pronto um determinado ROB envie seus dados;
3. Verificação da integridade da RoI - nesta etapa tomam-se todos os cuidados necessários quanto a completude da RoI. Caso sejam identificadas regiões faltantes, é nesta fase que seria possível sua correção;
4. Cálculo dos anéis baseando-se no centro de deposição de cada camada;
5. Normalização dos dados de entrada para a rede neuronal;
6. Discriminação neuronal, utilizando uma rede 58-5-1, conforme descrita no Capítulo 5 (teste 17 da Tabela 5.5);

A Figura 6.5 mostra um fluxograma dos passos descritos, mostrando a troca de mensagens entre o sistema hospedeiro e a aplicação no DSP, assim como as medidas tomadas para estimar o tempo de processamento, tanto da aplicação no DSP quanto para a troca de mensagens entre o sistema hospedeiro e o DSP. Nessa figura, é possível entender a complexidade adicional ao envolvermos um nó de processamento auxiliar no sistema.

Na implementação da rede neuronal (passo 6), utilizou-se como função de ativação a implementação da função tangente hiperbólica nativa do DSP. Optou-se por esta alternativa pois o tempo de execução do classificador neuronal representa somente cerca de um milésimo do tempo que o algoritmo de pré-processamento de dados leva para executar. Desta forma, o provável ganho em tempo de execução não justificaria o aumento na complexidade da aplicação.

6.3.3 Resultados da implementação

Para verificar o funcionamento da implementação do decisor local no DSP, comparou-se suas saídas com as do processo *off-line*. A Figura 6.6 mostra as diferenças entre as saídas geradas pelo decisor local rodando no DSP e a versão *off-line* com 64-bits de precisão e implementação da ativação neuronal usando a função tangente hiperbólica nativa do sistema. Como é possível observar nessa figura, o erro médio é bastante pequeno e compatível com o que foi observado na Tabela 6.1.

A Figura 6.7 mostra uma distribuição do tempo que o decisor local leva para executar no DSP para cerca de 600 eventos. A figura foi obtida medindo-se o tempo de execução da aplicação no DSP através de variáveis internas e transmitindo o resultado ao sistema hospedeiro. A Figura 6.5 mostra em que momentos o cronômetro do DSP foi inicializado e parado (*DSP timer*).

A média para o tempo de processamento de 1,22 ms representa a décima-terceira parte do que foi encontrado para a versão *off-line* do sistema! Este tempo está enquadrado dentro do que é desejável para um decisor local real, já que o tempo de processamento por evento do ATLAS não deverá exceder ao valor de 10 ms (Capítulo 3). A flutuação encontrada no tempo de processamento se deve a variações no tamanho (número de células) das RoI's analisadas.

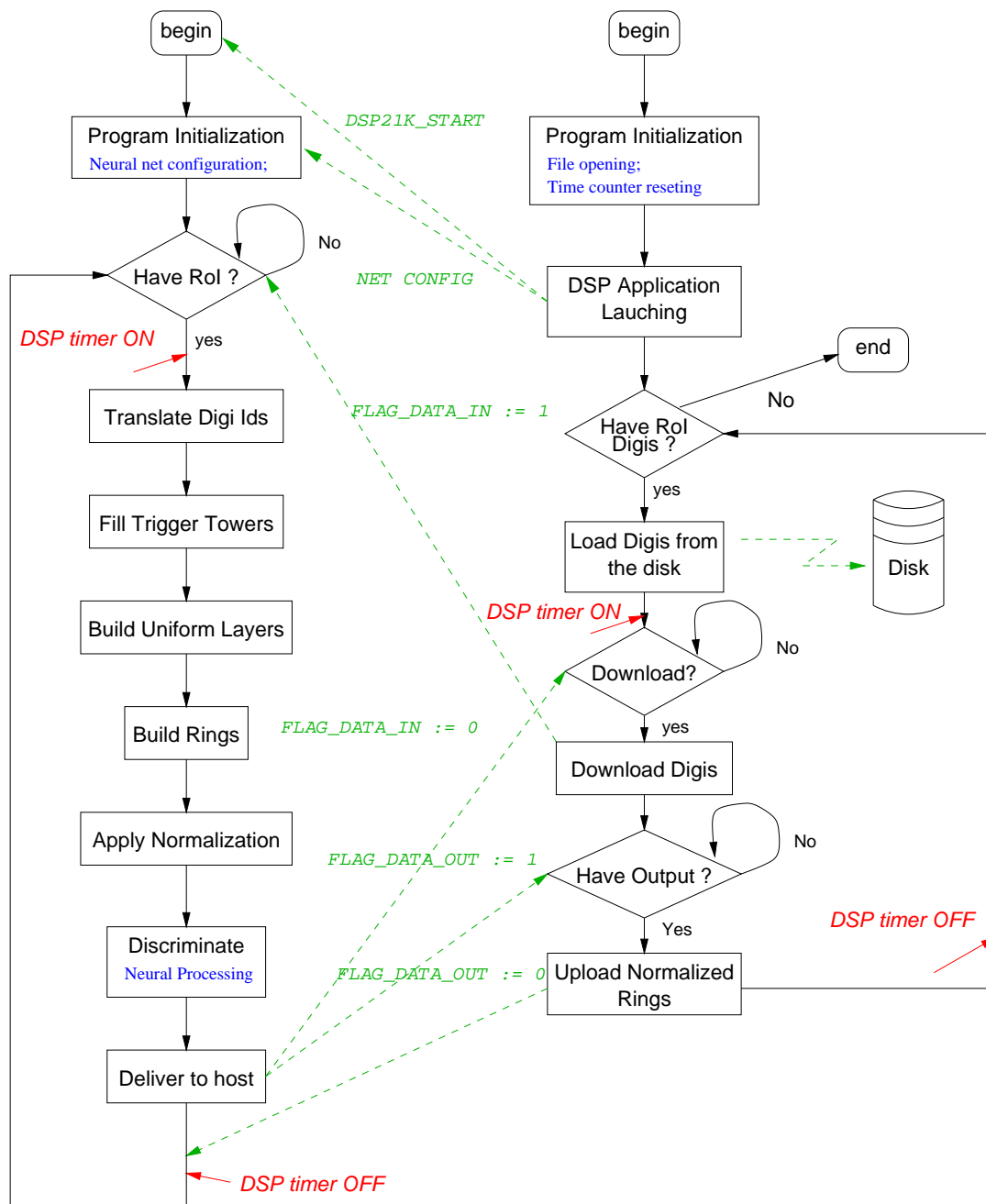


Figura 6.5: Um fluxograma mostrando a atuação conjunta da aplicação hospedeira com a aplicação no DSP. Este diagrama mostra também os pontos que foram utilizados como referências nas medidas de tempo.

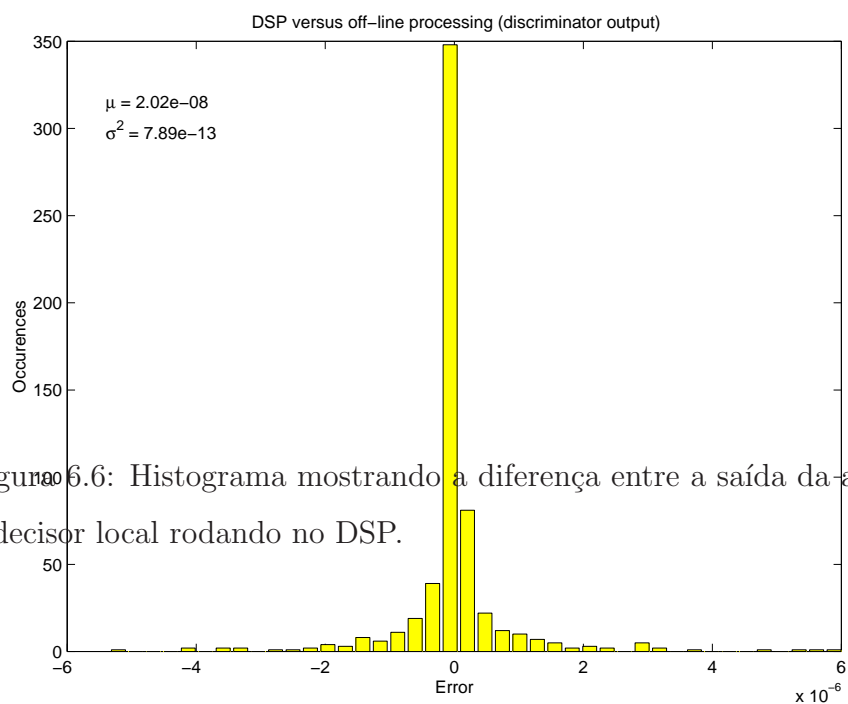


Figura 6.6: Histograma mostrando a diferença entre a saída da aplicação *off-line* e o decisor local rodando no DSP.

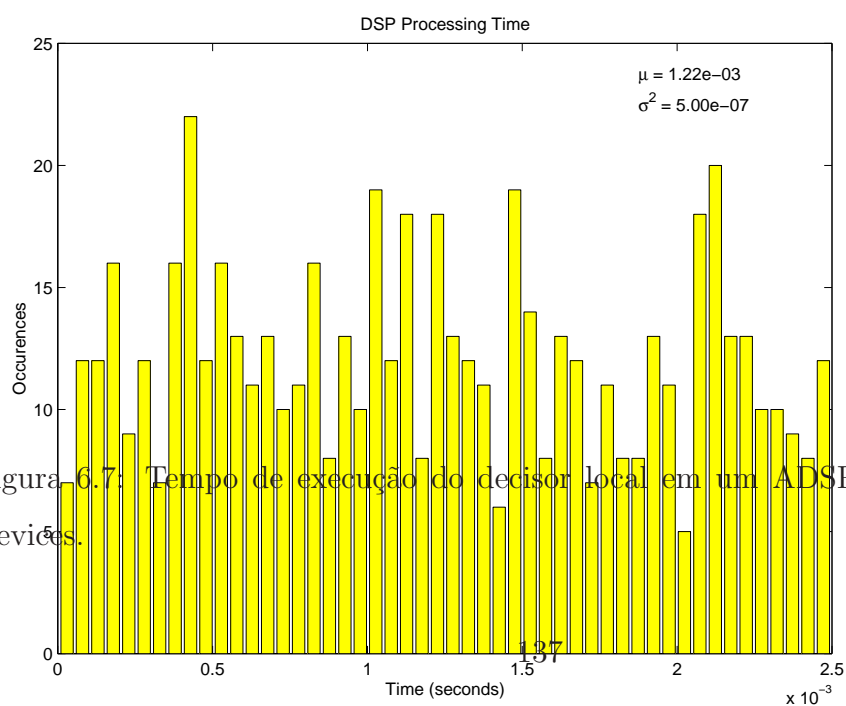


Figura 6.7: Tempo de execução do decisor local em um ADSP 21061 da Analog Devices.

Capítulo 7

Conclusões

Neste trabalho, apresentou-se, para o experimento ATLAS, um discriminador neuronal elétron/jato que representa, tanto em tempo de execução quanto em classificação, uma opção mais eficiente para o segundo nível do sistema de filtragem de eventos *on-line* do experimento.

O detetor ATLAS (Capítulo 2) é um complexo laboratório que tem por objetivo a detecção do raro bóson de Higgs. Para isto, se utiliza de modernas técnicas de detecção e conta com o auxílio de um sistema de filtragem de eventos muito eficiente, que reduzirá a taxa de eventos inicial de 40 MHz para cerca de 10 a 100 Hz, eliminando, durante essa filtragem, eventos que indicam a ocorrência de física ordinária (ruído), mas mantendo aqueles eventos que, potencialmente, representem decaimentos de bósons de Higgs (sinal).

O sistema de filtragem para o experimento é composto de 3 níveis que estão conectados em cascata, com complexidade e tempo de processamento de eventos crescente. O primeiro nível (L1) tem por objetivo eliminar os eventos baseando-se na ocorrência de alguns objetos, enquanto os níveis posteriores (segundo e terceiro) farão a eliminação analisando cada evento como um todo. O primeiro nível, enquanto filtra, seleciona as regiões nos detetores (RoI's) nas quais o detetor foi sensibilizado. Esta informação é utilizada para iniciar os demais níveis de filtragem.

O segundo nível de filtragem (Capítulo 3) é o primeiro da série filtros que analisará o evento tendo em vistas suas características globais. Para cada objeto destacado no primeiro nível (RoI), o segundo nível executará rotinas de classificação, visando confirmar os objetos destacados pelo primeiro nível e, assim, classificar um

evento. Dentre os objetos mais importantes para o experimento ATLAS, encontram-se os elétrons, representando cerca de 60% dos objetos que serão analisados pelo L2. Elétrons, por sua vez, são comumente falseados por jatos, quando esses decaem de forma similar. Portanto, um classificador elétron/jato mais eficiente se faz necessário no segundo nível (L2).

O classificador elétron/jato tem o objetivo de discriminar jatos e elétrons, preservando o máximo de elétrons que for possível, já que estas partículas são portadoras da informação que se deseja detetar no ATLAS: o bóson de Higgs. Elétrons e jatos são comumente distinguidos pela forma que depositam energia nos calorímetros do ATLAS (Capítulo 5). Jatos, por serem um conglomerado de partículas hadrônicas, usualmente penetram com mais profundidade nos calorímetros, também produzindo uma cascata mais espalhada, enquanto elétrons produzem cascatas menores e mais colimadas (Figuras 5.2, 5.3 e 5.4).

7.1 Somas em anel

Métodos clássicos de detecção aproveitam-se das características físicas da deposição de energia nos calorímetros para obter uma alta eficiência de classificação (95% para elétrons e 88,4% para jatos, como visto na Seção 5.2.3.1), extraindo quatro variáveis físicas com alto poder de discriminação. Os métodos clássicos, no entanto, desconsideram jatos que possuem um padrão de decaimento extremamente semelhante ao de elétrons, e se diferenciam apenas por sutilezas no seu processo de interação com os calorímetros do ATLAS. Para classificar corretamente estes eventos, um sistema tão simplificado não pode ser empregado.

Embora não seja uma solução que aproveite-se da máxima granularidade dos calorímetros do ATLAS, a discriminação usando as quatro variáveis clássicas compacta o vetor de entrada de cerca de 1000 células para 4 quantidades, mantendo um bom desempenho. Re-avaliando o espaço de entrada inicial (as 1000 células) e observando-se a física de decaimento dos objetos de interesse, no entanto, é possível elaborar um sistema também compacto, mas que preserve melhor as diferenças nos padrões de decaimento entre as duas classes de partículas. Nesse novo sistema, identifica-se o pico de deposição energética em cada camada e executa-se um pro-

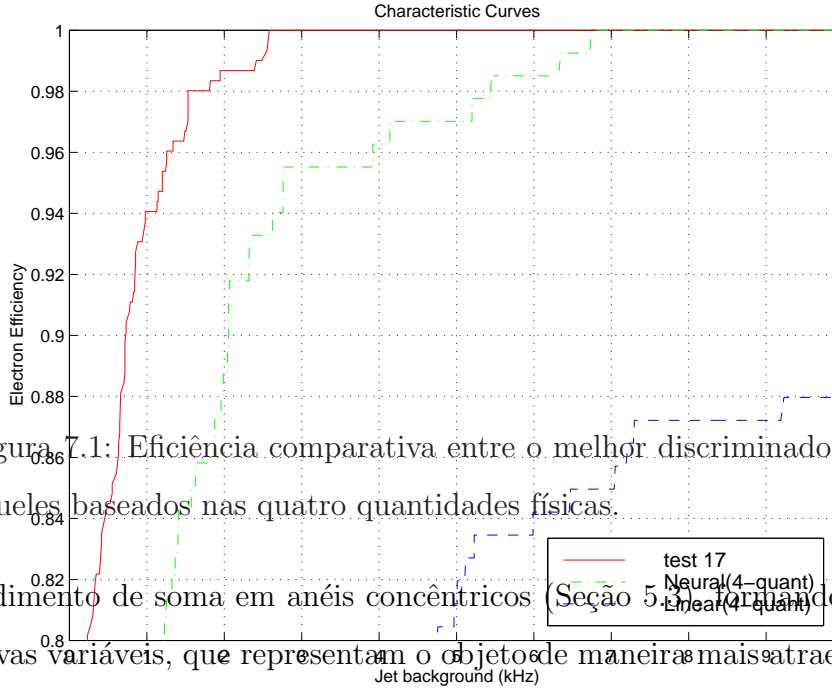


Figura 7.1: Eficiência comparativa entre o melhor discriminador usando 58 anéis e aqueles baseados nas quatro quantidades físicas.

cedimento de soma em anéis concêntricos (Seção 5.3). Quando se um total de 58 novas variáveis, que representam o objeto de maneira mais atraente para um classificador.

Projetando-se, então, um classificador neuronal para este novo espaço de entrada (de dimensão 58), é possível obter uma eficiência de classificação bastante acima do que foi obtido usando-se as quatro variáveis discutidas. A figura 7.1 mostra a curva característica do Teste 17 da Tabela 5.5, quando comparada ao melhor discriminador neuronal utilizando as quatro variáveis clássicas e ao discriminador linear, também para variáveis clássicas. Como é possível ver nessa figura, a eficiência de classificação usando anéis é bastante superior àquelas usando as quantidades clássicas.

7.2 Relevância

O espaço gerado a partir das 58 somas em anel pode ainda ser reduzido, observando-se a relevância (Seção 5.6) de cada uma das entradas para o classificador

neuronal proposto. A relevância, como foi visto, mede o impacto na classificação, quando substitui-se uma das variáveis de entrada pela estimativa de sua média estatística.

Aplicando-se o estudo de relevância, tanto aos classificadores baseados nas variáveis clássicas quanto aqueles baseados nas somas em anéis, é possível ver que nem todas as variáveis disponíveis para a classificação são, de fato, relevantes para o discriminador neuronal. Para o caso das variáveis clássicas, a remoção da variável $R_{e.m.1}^{\text{strip}}$ e o treinamento de um novo classificador (agora baseado em um novo espaço tri-dimensional) não demonstra um impacto profundo no poder de classificação de elétrons e jatos, como é possível notar na Figura 5.23.

No classificador baseado nas somas em anel, a redução da dimensão do espaço de entrada, de 58 para 20 (eliminando 38 anéis!) também não influencia de forma significativa (Figura 5.24) a discriminação elétron/jato. Algum impacto, no entanto, pode ser observado continuando-se a reduzir a dimensionalidade do espaço de entrada. Ainda assim, classificadores com apenas 5 dos 58 anéis mantiveram desempenho acima do melhor desempenho que foi obtido com as 4 variáveis clássicas usando um neuronal (Figura 5.26). Estes resultados indicam que há melhores maneiras de compactar os dados de uma RoI, de forma a discriminar mais eficientemente elétrons e jatos.

O excelente desempenho dos discriminadores baseados nas somas em anel advém do aproveitamento da máxima granularidade do detetor, sem que as sutilezas nos padrões de decaimento dos objetos sejam perdidas, o que não é possível observar quando aplica-se o estudo com as variáveis clássicas. Esta compactação em anéis, ademais, permite um estudo mais detalhado da relevância das diversas camadas dos calorímetros do ATLAS, quando expostas ao problema da discriminação de objetos e.m.. Através deste estudo, é possível estabelecer critérios de simplificação (cortes de dimensionalidade) do discriminador de forma a torná-lo suficientemente rápido, mas ainda mantendo uma eficiência de discriminação acima das possíveis soluções com o sub-espaço de variáveis clássicas, como foi possível observar na Figura 5.26.

A **não** utilização dos dados da terceira camada e.m. dos calorímetros não impediu que uma eficiência maior fosse atingida, utilizando a soma em anéis. Possivelmente, a inclusão dos dados desta camada em estudos posteriores possa mostrar

uma melhora significativa no método proposto, acentuando ainda mais as diferenças entre o método de discriminação hoje empregado no segundo nível de filtragem do ATLAS e aquele desenvolvido neste trabalho.

7.3 Implementação

A implementação do sistema de decisão local *off-line*, baseado nas somas em anel, revelou que grande parte do tempo de pré-processamento de dados (cerca de 87%), fase que antecede a filtragem propriamente dita, é gasto com a aquisição de dados (Figura 6.2), e que apenas 6% deste tempo é gasto com a soma em anéis. Estas estimativas indicam que mesmo um sistema baseado em um pré-processamento mais simples, como aquele usando as quatro variáveis clássicas, terá problemas com relação à latência (de eventos) no segundo nível de filtragem, atualmente estimada em 10 ms. Além disso, estas estimativas indicam também que os esforços para reduzir o tempo de processamento do decisor local (para objetos e.m.) devem ser focalizados na aquisição de dados para esta parte do sistema de filtragem e não com a classificação dos objetos propriamente dita que, como foi visto, pode ser executada de forma extremamente veloz usando-se um sistema neuronal, mesmo quando utiliza-se um PC e não se aplica a otimização automática na compilação do código (Tabela 6.2).

Neste contexto, a utilização de tabelas (*Look-up tables*) não se mostra necessária, já que os cálculos que podem ser baseados neste método representarão uma ínfima parte do tempo global de processamento. Por outro lado, a tecnologia de processamento a ser utilizada no sistema de decisão local representará um fator extremamente importante para o segundo nível: a simples conversão das rotinas de filtragem *off-line* implementadas para que rodem em um DSP (Apêndice D) reduziu o tempo de processamento de 13 ms para cerca de 1 ms (Figura 6.7)! É possível que com poucos ajustes e utilizando funções nativas ao DSP, seja possível atingir uma redução ainda maior no tempo de processamento do decisor neuronal local, tornando-o uma opção ainda mais atraente para o segundo nível de filtragem.

Lembra-se ao leitor que esta é a primeira vez que um sistema tão completo em termos de decisão local é implementado. Em [37], um sistema menos complexo

de classificação de objetos e.m. foi desenvolvido, tendo por base as quatro variáveis clássicas. Esta versão foi capaz de executar a filtragem do evento em cerca de 50 μ s, usando um Pentium II @ 300 MHz. Comparando-se este resultado com o obtido para o classificador neuronal (aproximadamente 18 μ s), foi possível perceber que a classificação proposta neste trabalho, além de mais eficiente, é também mais veloz. O pré-processamento dos dados, sem verificação de integridade dos dados recebidos e descontando-se o tempo de aquisição levará cerca de 175 μ s na versão usando as quatro variáveis clássicas, ainda segundo [37]. Estimando que apenas 6% do tempo é gasto com a extração das somas em anel (“pré-processamento”equivalente ao proposto pelo trabalho citado), o tempo gasto para a mesma atividade é de cerca de 780 μ s, ou seja, cerca de 4,5 vezes maior. Ainda assim, lembra-se ao leitor que a versão que se está utilizando para comparação não foi otimizada. Em uma versão mais otimizada do sistema, rodando em um DSP com apenas 40 Mhz de clock (ADSP-21062 da Analog Devices, Apêndice D), a extração das somas em anel equivaleria a 60 μ s, ou seja, 1/3 do tempo da versão apresentada em [37], considerando-se a projeção percentual do tempo vista na Figura 6.2.

O sistema implementado poderá ser facilmente integrado ao *Software* de Referência (Capítulo 4), como foi explanado na Seção 6.1.1. Este ambiente inclui uma completa interface de emulação do segundo nível de filtragem do ATLAS, fornecendo também, meios mais acurados de comparação entre as diversas opções de processamento disponíveis atualmente.

7.4 Extensões

Nesta seção, discute-se possíveis extensões para este trabalho, com o intuito de melhorar os resultados obtidos aqui ou reduzir o tempo de processamento necessário para a correta classificação de objetos e.m., no contexto do segundo nível de filtragem do ATLAS:

1. No trabalho apresentado não efetuou-se um estudo para a inclusão dos dados da terceira camada e.m.. A anexação desta parte do calorímetro poderá aumentar a capacidade de filtragem (usando redes neurais) do decisor local. Para isto, deve-se averiguar como tratar situações onde há mudanças de

granularidade dentro da própria camada (Figura 6.1), de forma a aplicar a compactação resultantes das somas em anel. Tanto a interpolação quanto a decimação podem ser soluções favoráveis. Este estudo também permitirá que RoI's cujo o centro está em $|\eta| > 1,1$ possam ser avaliadas pela técnica de classificação neuronal aqui proposta;

2. A redução do número de anéis usando-se o método da relevância poderá diminuir ainda mais o tempo de pré-processamento (incluindo a aquisição de dados), aliviando a carga no segundo nível de filtragem, mas mantendo o desempenho do discriminador. Uma das possíveis extensões deste trabalho poderá averiguar o impacto no tempo de processamento e na robustez, decorrentes da redução do número de anéis utilizado para a discriminação. A redução da dimensão do vetor de entrada para o classificador neuronal também poderá simplificar o estudo do item 1 desta lista, já que regiões que deveriam sofrer ajuste de granularidade podem não ser tão relevantes à classificação, podendo, portanto, ser eliminadas para este efeito;
3. Para realizar estudos mais elaborados quanto ao tempo de processamento do decisor local, deve-se portar a aplicação proposta para dentro do *Software de Referência*, como proposto acima.
4. O pré-processamento dos dados usando a soma em anéis é uma atividade paralelizável, à medida que as diversas camadas do calorímetro podem ser processadas de forma independente. Portanto, a utilização de um ambiente multi-processado, preferencialmente com DSP's, poderá diminuir significativamente o tempo de processamento para o decisor local;
5. A Figura 5.20 mostra os valores da relevância, de cada entrada, para o processo de classificação de objetos e.m., quando utilizou-se diversos tipos de normalização. Como foi visto, há um desequilíbrio bastante grande na ênfase que os diversos classificadores dão às entradas, indicando, possivelmente, que nenhuma das soluções estudadas representa a forma ótima de normalização dos dados. Estudos utilizando soluções híbridas de normalização devem ser conduzidos procurando o enfoque ótimo para esta etapa do pré-processamento;

6. Os dados utilizados no trabalho exposto não incluem efeitos de empilhamento, como enfatizado na Seção 5.1.1. As próximas etapas devem incluir eventos com este tipo efeito, para que seja possível estimar o desempenho do sistema de validação quando da operação plena (alta luminosidade) do colisionador LHC;
7. Como colocado na Seção 3.5.3, uma forma de tratamento de dados faltantes deverá ser introduzida no sistema, gerando um classificador mais robusto, com capacidade atender as reais expectativas do segundo nível de filtragem do ATLAS;
8. Nesse trabalho, utilizou-se a soma em anéis não-circulares ou quadrados para a compactação dos dados de entrada do decisor local. Uma possibilidade a ser averiguada é a compactação da informação da RoI em anéis circulares. Nesta modalidade, a soma em anéis respeita raios crescentes em relação ao centro de deposição energética de cada camada. A energia das células é dividida de forma ponderada pela contribuição da área da célula em cada anel a que pertence;
9. A utilização do produto SP como figura de mérito para o treinamento e avaliação do desempenho de uma rede neuronal leva em conta somente um ponto na curva característica do detetor, como foi visto no Capítulo 5. Uma forma mais eficiente de mensurar o desempenho da rede neuronal pode ser proposto observando-se a área abaixo da curva característica. Esta área é tão maior quanto mais eficiente é o discriminador e pode ser considerado uma figura de mérito mais robusta, pois leva em conta não somente um ponto na curva característica, mas um conjunto de pontos. A avaliação da área também pode ser reduzida a um determinado espaço de operação, já que usualmente baixas eficiências de discriminação não são relevantes. Desta forma, o cálculo da nova figura de mérito pode ser feito de forma mais rápida e o discriminador encontrado ao final do treinamento melhor sintonizado para as necessidades do processo a que se destina.

A área abaixo da curva característica do detetor também pode ser usada para a definição de um novo critério que estude a importância (ou relevância) das

variáveis de entrada para o discriminador. O aumento dessa área resultante da substituição da variável por sua média indica maior importância na discriminação, enquanto que a diminuição desta área, menor importância. Este critério poderá indicar variáveis que são muito, pouco ou até anti-relevantes à discriminação, o que não é possível realizar utilizando diretamente o critério da relevância exposto aqui.

Apêndice A

Trabalhos publicados

Este apêndice lista algumas notas e artigos publicados pelo autor, que tiveram impacto neste trabalho ou dele são resultantes. Os resumos (*abstracts*) destes trabalhos estão inclusos na listagem.

1997 *Fast Neural Decision System Based on DSP's and Parallel Processing* - International Conference on Signal Processing Applications and Technologies

Um protótipo para um sistema *on-line* de validação é desenvolvido para aplicações em experimentos com colisionadores de altas energias. O sistema, primordialmente, usa redes neurais para extrair eventos raros com significado físico de um ruidoso ambiente. Tal sistema é baseado no processamento da informação coletada de diferentes detetores que são posicionados ao redor da colisão. Combinando uma fase de extração de características para cada detetor com uma fase de decisão global para uma decisão final sobre eliminar ou não um evento, o sistema atua em eventos previamente selecionados pelo primeiro nível de análise que reduz a taxa de eventos para 100 kHz. Para cobrir com esta frequência de entrada, o sistema proposto está sendo emulado em um computador baseado em *transputers* com 16 nós de processamento que, por sua vez, possuem um processador digital de sinais acoplado, funcionando como co-processadores para os *transputers*.

- 1998 • ***A Neural Online Triggering System based on Parallel Processing - IEEE Transactions on Nuclear Science***

O estudo de um protótipo do segundo nível de filtragem que operará nas condições do LHC é endereçado através de uma implementação baseada em uma máquina paralela. O computador baseado em 16 nós, contendo em cada um *transputer*, usa um processador digital de sinais que atua como co-processador para otimizar o processamento de sinal nas aplicações. Um ambiente de desenvolvimento em linguagem C é usado para rodar todas as aplicações em sua forma mais veloz. A implementação é baseada na informação dada por quatro detetores e inclui duas fases distintas de operação: extração de características e decisão global. A extração de características para calorímetros e a decisão global são executadas usando-se redes neuronais. O pré-processamento e os parâmetros das redes (neuronais) são guardados em memória e a função de ativação é implementada usando uma tabela. Dados simulados para o segundo nível de filtragem são utilizados para a avaliação da performance.

- ***Requirements, Constraints, Use-Cases and an Object Oriented Analysis of the Steering and Feature Extraction Systems for ATLAS/Trigger Level 2 - CERN - Testbed Technical Note 8***

Esta nota reúne um conjunto de demandas, restrições e “casos-de-uso” para o Supervisor e os Sistemas de Processamento de Dados (SPD) do segundo nível de filtragem (L2). Um Análise Orientada a Objetos (AOO) é conduzida na sequência, respeitando abstração, encapsulamente e as demandas e restrições do sistema. O objetivo deste documento não prover um conjunto completo de demandas, restrições e “casos-de-uso”, mas formar uma base para desenvolvimentos futuros e ampliações. As demandas foram coletadas de notas diferentes e organizadas. Entidades (predecessores de classes) são abstraídas dessas demandas e relacionamentos são baseados nos “casos-de-uso”.

- ***OO Design of the Steering System*** - CERN - *Testbed Technical Note 17*

Esta nota sumariza o projeto orientado a objetos que foi utilizado no sistema de supervisão para o *Software* de Referência do segundo nível de filtragem. Além da definição das interfaces para o supervisor e para o sistema de extração de características, a definição das classes mais relevantes para o supervisor é dada. Ademais, a relação dessas classes é definida e as mensagens passadas entre estes objetos são explicadas de forma que opere como esperado durante o processamento de um evento.

- ***Design of the Feature Extraction System*** - CERN - *Testbed Technical Note 20*

Este artigo apresenta um projeto orientado a objetos e definições de classe da interface entre o sistema de Extração de Características (“*FEx System*”, contendo o Coletor de Dados e o algoritmo de FEx propriamente dito) e as outras partes do sistema de filtragem de segundo nível diretamente conectadas ao FEx (Supervisor, ROB’s e a Descrição Geométrica do Detetor).

- **Um Protótipo do Sistema de Validação de Nível 2 para as Condições do LHC** - XIX Encontro Nacional de Física de Partículas e Campos

O experimento ATLAS pretende comprovar a existência do bóson de Higgs. Para tal, um grande sistema de detecção e aquisição vem sendo projetado. O sistema aquisição tem a função de e separar em tempo real interações originárias do decaimento de um Higgs de física ordinária.

A filtragem de eventos no sistema de aquisição é concebida em três níveis de complexidade crescente e velocidade decrescente. O segundo nível pretende utilizar redes de computadores pessoais (PC’s) interconectados por rápidos sistemas de rede. A escolha de fabricantes, sistemas operacionais e algoritmos de processamento ainda não foi feita, mas esforços em prol desta decisão

vem sendo realizados. Neste trabalho desenvolve-se uma fração do filtro de segundo nível utilizando-se de processamento paralelo, redes neurais artificiais e DSP's. Conclusões e resultados são endereçados no final.

1999 Integrando Plataformas e Algoritmos para o Segundo Nível de Trigger do Experimento ATLAS - XX Encontro Nacional de Física de Partículas e Campos

No ano de 2005 entrará em funcionamento o experimento ATLAS, que pretende verificar interações a 16 Tev. A física de interesse é rara e o número de eventos gerados é enorme. Um sistema de filtragem será, então, responsável pela separação da física de interesse da física ordinária. Este sistema é dividido em níveis de processamento. No segundo nível de filtragem estão sendo investigados vários algoritmos e plataformas. Aqui apresentamos desenvolvimentos para o Software de Referência, que visa integrar os estudos de algoritmos com realização tecnológica neste nível de filtragem.

2000 *Neural Electron-Jet Separation Using Multi-layered and Multi-segmented Calorimeters* - XXI Encontro Nacional de Física de Partículas e Campos (submetido ao *Brazilian Journal of Physics*)

Este trabalho descreve um enfoque neuronal para a discriminação elétron/jato no segundo nível de filtragem do ATLAS. Estudos são baseados numa compressão rápida usando o perfil de deposição energética das partículas nos calorímetros. Uma comparação entre este enfoque e o atual (clássico) para a mesma tarefa mostra melhores resultados. O método proposto aqui consegue atingir 95,67% de eficiência na discriminação de elétrons com menos de 4,26% de jatos sendo erroneamente classificados.

2001 Os resultados deste trabalho serão submetidos ao *Computer Physics Communications*.

Apêndice B

O Sistema de Coordenadas do ATLAS

O sistema de coordenadas usados em experimentos com feixes não é o sistema polar. É um sistema adequado ao formato cilíndrico dos detetores dispostos ao redor do ponto de impacto, ou seja, um sistema que *acompanha* a direção dos feixes de partículas provenientes da colisão. As coordenadas empregadas são η , ϕ e z em contraposição a x , y e z . η e ϕ seguem a uma transformação não-linear de x e y :

$$\phi = \text{atan}\left(\frac{x}{y}\right) \quad (\text{B.1})$$

$$\eta = -\log\left(\tan\left(\frac{\phi}{2}\right)\right) \quad (\text{B.2})$$

A figura B.1 pode ser explicativa quanto ao sistema. Em sua parte superior, é possível ver um esquema do barril e da tampa de um detetor, mostrando como se comportam as coordenadas tomando por referência as coordenadas cartesianas x , y e z (marcadas em pontilhado). Nota-se que a variável ϕ representa a rotação e a variável η (também chamada de pseudo-rapidez) representa a direção de projeção das partículas, após a colisão.

Os valores dados das variáveis η e ϕ são apenas para referência do leitor. A variável ϕ , como é possível ver no canto direito da parte superior da figura, possui uma região em que dois valores são possíveis: 0 e 2π . Esta área é chamada de região *wrap-around*. Cálculos utilizando esta variável devem atentar para este fato. Os detetores são simétricos, com relação ao eixo ϕ . A construção dos dispositivos é

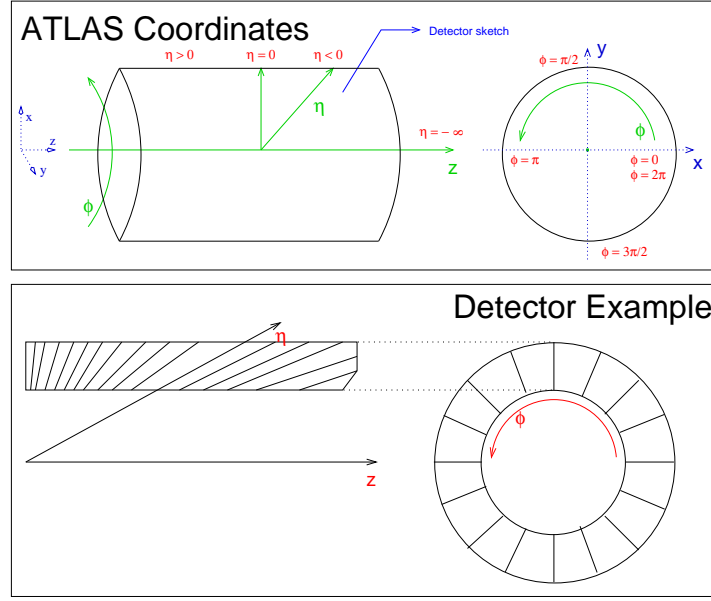


Figura B.1: O sistema de coordenadas do ATLAS.

feita em gomos.

Repare que quando alcança o eixo z , $\eta = \infty$, isto significa que objetos com valores grandes em η representam colisões onde as partículas do feixe apenas se desviaram, não havendo, usualmente informações interessantes de análise pois representam choques elásticos. É comum utilizar-se detetores com baixa resolução quando $\eta > 3$.

Na parte inferior da figura B.1, é possível ver um exemplo de como um detetor genérico é segmentado, acompanhando as coordenadas η e ϕ , tanto para o barril, quanto para uma tampa.

Apêndice C

Orientação a Objetos e as Linguagens C e C++

Não há, sem dúvida, uma melhor forma de desenvolver soluções para **todos** os problemas que podem ser resolvidos com programação. Há problemas que possuem ótimas soluções estruturadas ou orientadas a texto. Ainda assim, existe um grande número de problemas que podem ser resolvidos analisando-se a forma com que alguns elementos deste problema interagem. Este tipo de análise extrai os “objetos” do problema, e é comumente chamada de análise orientada a objetos ou OOA.

Após a OOA, desenvolve-se um projeto que respeite o conjunto de objetos encontrados e suas interações. Nesta fase, possivelmente, elementos abstratos para suprir carências do projeto podem ser criados. Esta fase é chamada de projeto orientado a objetos ou OOP. Na OOP deseja-se resolver o problema através da interação de objetos existentes na OOA com novos e abstratos elementos.

C.1 OO × Estruturação

O leitor pode estar se perguntando neste momento: “Quando o *programa* surgirá?”. A diferença marcante entre projetos OO e os antigos projetos estruturados baseados em linguagens como Pascal, Fortran ou o próprio C, é que não deve se programar nada antes do entendimento do problema ser *o mais completo possível*. Não é possível escrever código eficientemente sem que se conheça boa parte dos objetos que interagem no sistema. Isto pode levar a resultados bastante desastrosos

[38].

Não se está dizendo aqui que a programação estruturada deve ser feita de forma mal planejada! Frisa-se apenas que um mal planejamento num projeto OO pode levar a piores resultados que usando outras formas de programação.

Quando deseja-se resolver problemas usando o paradigma da estruturação (de funções), deseja-se decompor o problema original em pequenos blocos que possuem um única finalidade e são facilmente programáveis. Estes pequenos blocos, quando invocados pelo programa principal ou por outros blocos interagem resolvendo o problema original. Neste caso, a análise do problema deve ser feita tomando-se por base a funcionalidade de seus componentes. Ou seja, deve-se deixar o espaço original de informação (ou seja os componentes) e criar um espaço-solução, composto agora de várias funções. Invariavelmente as soluções para um mesmo problema variam de analista para analista.

Quando analisa-se um problema usando-se técnicas OO matém-se no espaço original do problema, somente extraindo os componentes mais evidentes e que mais se iteragem dentro deste. Invariavelmente, os objetos extraídos são os mesmos de analista para analista. Isto não é um espanto, pois a sensatez impede-nos de extrair objetos que não existam no escopo do problema. Abaixo estão anexados vários trechos de um monólogo hipotético de uma dona de restaurante que deseja integrar por computador vários aspectos de seu restaurante. No monólogo, a senhora descreve como funciona seu restaurante. Os objetos estão sublinhados. Veja como é fácil identificá-los:

... os clientes podem chegar ao restaurante diretamente ou reservar uma mesa com nosso serviço de reservas. Caso optem por reservar uma mesa, devem notificar a hora de chegada e o número de pessoas que serão servidas durante a refeição. O número de telefone do cliente que reserva a mesa também é anotado.

... os garçons anotam os pedidos dos clientes e os levam até a cozinha. Os cozinheiros devem atender o pedido. Caso falte algum item pedido o cozinheiro deve informar imediatamente à gerência, para que providencie o item nas próximas compras.

Durante uma análise mais depurada ou na fase de projeto, muitos dos objetos

destacados podem se revelar inúteis. Neste caso, estes elementos são eliminados da solução do problema. É importante notar, entretanto, que provavelmente você teria destacado o mesmo conjunto de objetos que foram marcados ali. Por causa disto, o entendimento e a manutenção de sistemas OO é bem mais fácil, pois é natural.

Em contrapartida, analisando-se o problema para resolvê-lo usando-se a estruturação de funções muitas das funções poderiam ter nomes estranhos, além de acumular a funcionalidade de outras partes, tornando o simples entendimento e a re-utilização de software quase impossível.

C.2 Relacionamento de objetos

Objetos são basicamente caracterizados pelas suas qualidades. Por exemplo, um garçom tem um salário, uma frequência e é responsável por um número de mesas. Os objetos podem ser relacionar de várias formas:

- Possuindo outros objetos, como características próprias, por exemplo: a cozinha possui fogões;
- Cooperando com outros objetos, por exemplo: o garçom atende ao cliente;
- Sendo um “herdeiro” de algum tipo de objeto. Por exemplo: um cozinheiro e um garçom são funcionários. Um funcionário possui salário, logo um cozinheiro e um garçom também têm salário.
- etc.

A forma com que os objetos se relacionam implementa a solução de determinados problemas. Há, no entanto, formas comuns de relacionamento entre objetos que podem ser facilmente adaptadas a uma vária gama de problemas [39].

Suponha, por exemplo, que deseja-se organizar várias palavras (ou seja, um dicionário) em diversos formatos, utilizando, portanto, diversos algoritmos diferentes. O diagrama da figura C.1 exemplifica uma solução bastante comum. Neste diagrama é possível ver os objetos representados segundo a UML (do inglês *Universal Modelling Language*), uma linguagem de diagramação para projetos orientados a objetos. O objeto `Dictionary` possui um conjunto de palavras como seus atributos. Eventualmente, o usuário do `Dictionary` poderá querer organizá-lo usando

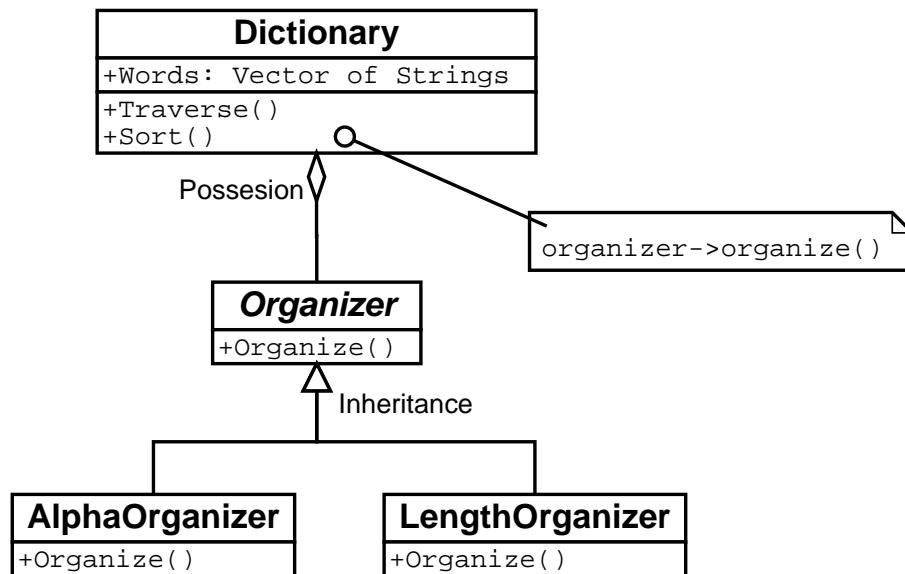


Figura C.1: Diagrama do “Ordenador”, uma técnica de projeto OO que permite que algoritmos sejam mudados em um objeto, sem alterá-lo.

ordenação alfabética ou pelo tamanho das palavras. Para que isto seja indiferente a este objeto, encapsula-se o algoritmo em uma classe que define sua interface: **Organizer**. Esta classe, não existe pois é uma classe abstrata. Uma classe abstrata não tem implementação, ela somente define um padrão de comportamento, ou seja, uma *interface* para suas herdeiras.

C.3 Implementação e C++

Durante o projeto, deve-se escolher uma linguagem de implementação. Este *dialeto* de programação deverá conter elementos que tornem o projeto OO simples de ser implementado. Linguagens de programação que possuem tais elementos são chamadas “orientadas a objetos”, pois permitem que o paradigma de programação adotado seja codificado diretamente, ou seja, não implique no desenvolvimento de soluções de implementação personalizadas.

A título de exemplo, ver-se-á como a implementação das classes **Dictionary** e **Organizer** serão conduzidas em C++. Esta linguagem é uma derivativa do C, criado em 1978 por Dennis M. Richie [40, 41]. C++ contém diretivas que permitem a simples programação de sistemas OO.

Em C++, a definição dos objetos destacados acima transformam-se em classes. Ao *instanciar* uma classe, cria-se um objeto daquele tipo. *Instanciar* significa declarar uma variável, exatamente da mesma forma que é feito em C. Na implementação C.1 encontra-se o trecho de programa em C++ que implementa as classes descritas acima e sua interação.

Nesta implementação, as palavras-chave em C++ estão sublinhadas. Inicialmente implementou-se a classe abstrata **Organizer**, que na verdade somente contém a definição da interface, o que fica determinado ao acrescentar o trecho “= 0” na declaração de seu único método. A implementação somente ocorrerá em classes derivadas daquela, como em **AlphaOrganizer**. A função que *constrói* objetos da classe **Dictionary** recebe como parâmetro um ponteiro para objetos do tipo **Organizer**. Desta forma, é possível inicializar nosso dicionário com qualquer tipo de ordenador de palavras. Veja que também é possível *trocar* o ordenador de palavras dinamicamente, já que se trata apenas de um ponteiro.

A implementação, em cada herdeiro da classe **Organizer** deverá ser conduzida, como indicado no final deste trecho de código. Como C é um sub-conjunto de C++, i.e., os comandos de C também são interpretáveis por um compilador C++, a implementação do algoritmo em si pode ser conduzida utilizando-se tanto uma análise estruturada, quanto orientada a objetos. Esta escolha fica ao encargo do programador.

```

// Define-se, inicialmente, a classe abstrata Organizer
// O termo "= 0" no final da declaração garante que não há
// implementação para esta função aqui.
CLASS Organizer {
PUBLIC:
    // Define-se a interface da função que ordena as palavras
    virtual void Organize(VECTOR<STRING>*) = 0;
PROTECTED:
    Compositor();
};

// Como escrever a classe Dictionary
CLASS Dictionary {
PUBLIC:
    Dictionary(Organizer*); // cria nova instância do dicionário
    VOID Sort(); // re-organiza o dicionário
PRIVATE:
    Organizer* _organizer;
    VECTOR<STRING> _words;
    ...;
};

// Agora é possível definir vários tipos de algoritmos de ordenação,
// herdando da classe Organizer a interface.
CLASS AlphaOrganizer : public Organizer {
PUBLIC:
    AlphaOrganizer();
    VIRTUAL VOID Organize(VECTOR<STRING>*);
    ...
PRIVATE:
    // Posso colocar aqui, os atributos que achar
    // necessário para realizar a organização das palavras
    ...
};

// Implementação da função
VOID AlphaOrganizer::Organize(VECTOR<STRING>* words)
{ // O que precisar fazer para organizar as palavras ... }

// É possível, agora, durante o programa, ter uma declaração
// do seguinte tipo:
Dictionary* dict = NEW Dictionary( NEW AlphaOrganizer() );

// O conjunto de palavras será organizado de acordo com o
// algoritmo selecionado na inicialização:
dict->sort();

```

Implementação C.1: As classes Dictionary e Organizer

Apêndice D

O ADSP-21062

O ADSP-21062 SHARC (*Super Harvard Architecture*) [42] é um processador digital de sinais para números reais (“pontos-flutuantes”) com 32-bits de precisão, 2 megabits de memória SRAM *on-chip* e suporte para periféricos integrados através de um barramento de entrada e saída (E/S) dedicado. Com seu *cache* de instruções *on-chip*, o processador pode executar cada instrução em somente 1 ciclo de seu *clock*. Quatro barramentos internos independentes podem ser usados para o tráfego de variáveis, instruções ou E/S, permitindo operações em paralelo.

O DSP também inclui outras características *on-chip*, como interface para um sistema hospedeiro, um controlador DMA (*Direct Memory Access*), portas seriais e portas específicas que podem ser usadas para estabelecer um ambiente de trabalho multi-processado, com vários DSP’s da mesma família.

A Figura D.1 ilustra a arquitetura SHARC do ADSP-21062:

Unidades de Computação O processador de um ADSP-21062 contém 3 unidades de computação independentes: uma ULA (Unidade de Lógica Aritmética), um multiplicador com acumulador em ponto-fixe e um *shifter*. As unidades podem processar dados em 3 formatos: ponto-fixe de 32-bits e ponto-flutuante de 32 e 40 bits. As operações em ponto-flutuante são compatíveis com o padrão IEEE;

Arquivo de Registradores de dados é usado para a transferência de dados entre as unidades de computação e os barramentos de dados e para o armazenamento de resultados imediatos;

Geradores de Endereços e Seqüenciador dois geradores de endereços dedica-

dos (DAG 1 e 2) e o seqüenciador de programas fornecem endereços para acessos à memória. Juntos, permitem que operações computacionais sejam geradas com máxima eficiência já que as unidades de computação serão exclusivamente destinadas ao processamento de dados. Com seu *cache* de instruções, o ADSP-21062 pode acessar uma instrução (do *cache*) e dois operandos (da memória) em um único ciclo de seu *clock*.;

Temporizador Para a geração de interrupções;

Memória SRAM O ADSP-21062 inclui 2 megabits de memória divididos em 2 bancos, que podem ser configurados usando diferentes combinações para o armazenamento de código e dados. Cada banco de memória possui 2 portas, possibilitando o acesso paralelo dos dados pelas unidades de computação.

Porta para memória externa *off-chip* e periféricos ;

Porta para um sistema hospedeiro Permitindo que o processador seja acoplado a um sistema hospedeiro através de um barramento padronizado (ISA), com o mínimo acréscimo de equipamento;

Interface para ambiente multi-processado Oferecendo um ambiente com espaço de endereçamento unificado, permitindo um acesso direto a memória de outros nós de processamento em um ambiente simétrico (com processadores idênticos);

Controlador de DMA ;

Portas seriais ;

Porta de teste (JTAG)

D.1 Desenvolvimento

Aplicações para o ADSP-21062 são desenvolvidas através um conjunto de ferramentas de desenvolvimento (compiladores de depuradores). Este conjunto de programas provêm ao desenvolvedor, a possibilidade de construir aplicações em assembly nativo ou C. Um emulador (EZ-ICE) permite a depuração da aplicação.

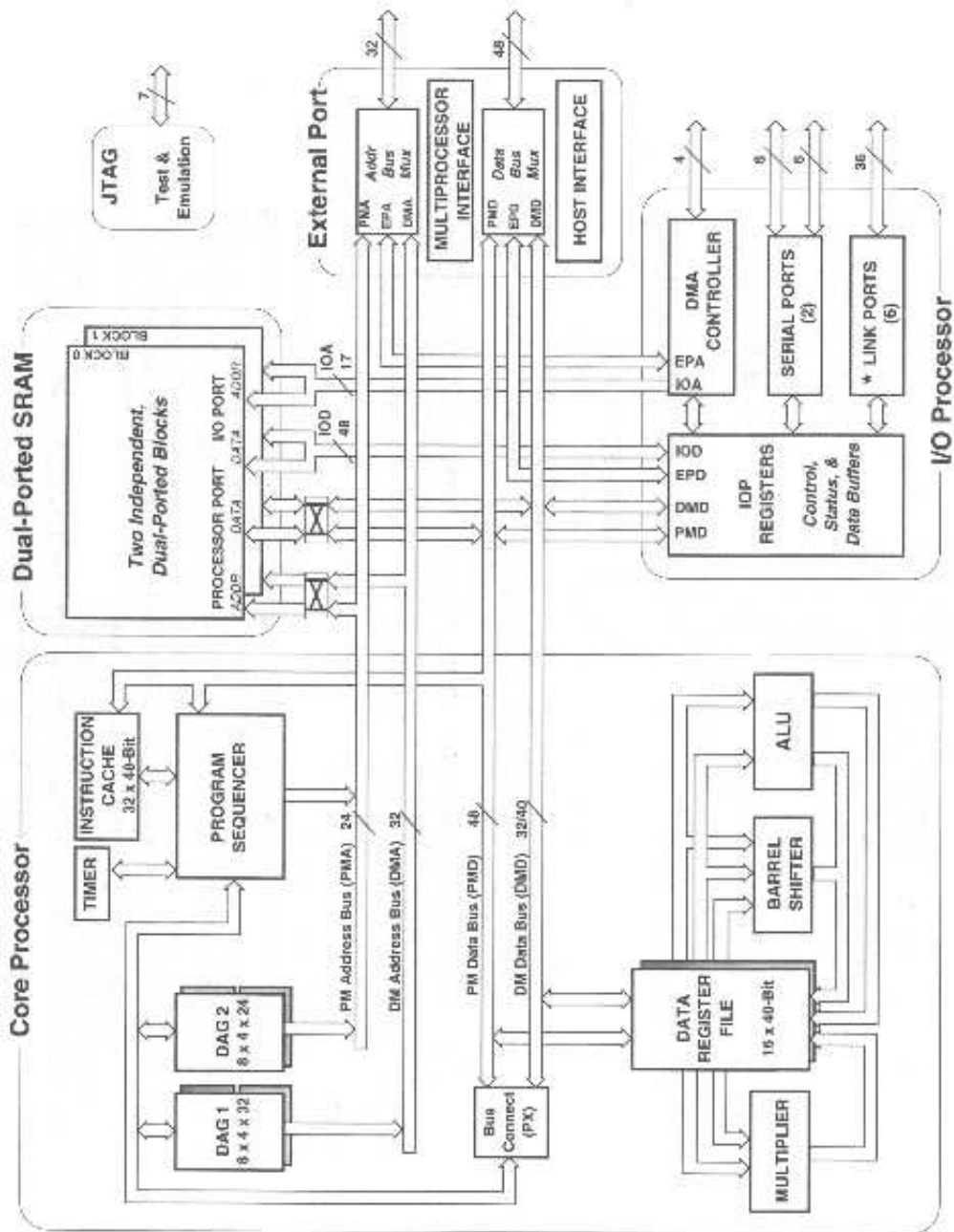


Figura D.1: Diagrama de blocos do ADSP-21062 SHARC. Extraído de [42].

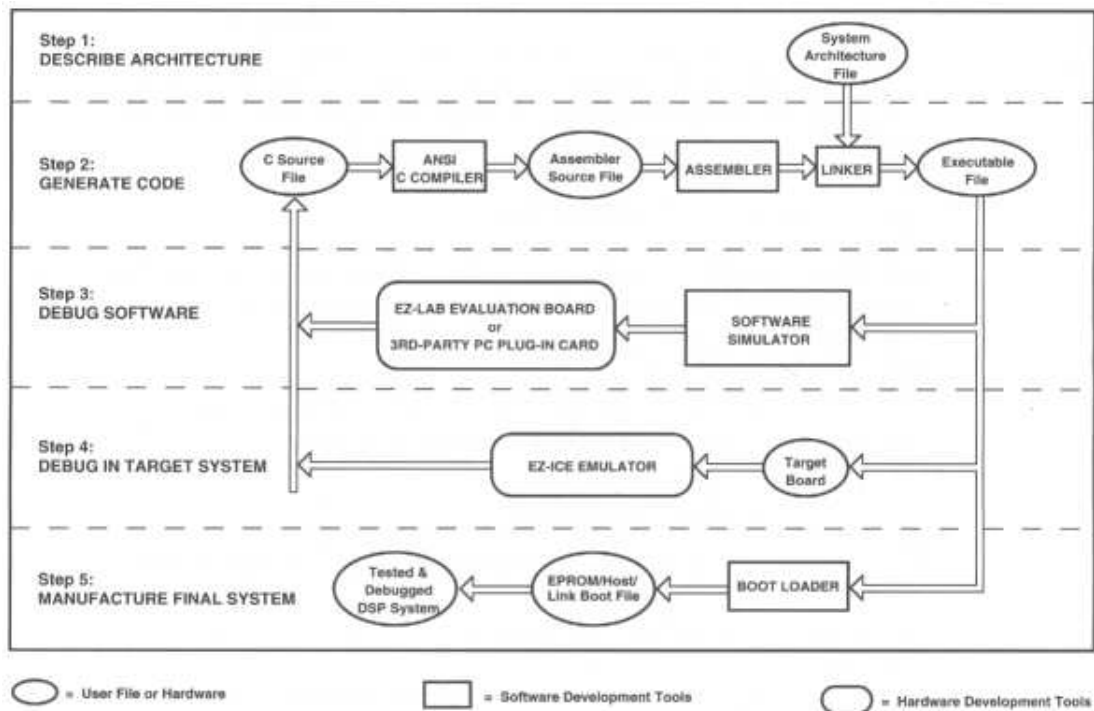


Figura D.2: Processo de desenvolvimento e projeto em um ADSP-21062. Extraído de [42].

A figura D.2 mostra o processo de desenvolvimento de aplicações no ADSP-21062. Essa figura mostra que o processo é composto de 5 passos:

1. Descrição da arquitetura - onde o desenvolvedor definirá como operará o sistema, a configuração da memória, interrupções, etc;
2. Codificação - onde escreve-se a aplicação em C ou Assembly nativo;
3. Depuração *off-line* - Este passo é opcional caso o desenvolvedor possua um simulador do DSP;
4. Depuração *on-line* - onde o desenvolvedor irá depurar a aplicação rodando no DSP, eliminando erros fatais, vazamentos de memória, etc.
5. Manufaturação - onde a aplicação, já construída e depurada será compilada diretamente em memória ROM para ser acoplada a um sistema dedicado.

Referências Bibliográficas

- [1] HALLIDAY, RESNICK, WALKER, *Fundamentos da Física*. 4 ed. Livros Técnicos e Científicos Editora, 1995.
- [2] “The Particle Adventure”, **World Wide Web:** <http://ParticleAdventure.org>.
- [3] “The European Laboratory for Particle Physics”, **World Wide Web:** <http://www.cern.ch>.
- [4] “Large Electron-Positron Collider”, **World Wide Web:** [http://cern.web.cern.ch/CERN/Divisions/% -SL/welcome.html](http://cern.web.cern.ch/CERN/Divisions/%20SL/welcome.html).
- [5] “Large Hadron Collider”, **World Wide Web:** <http://www.cern.ch/LHC>.
- [6] “Compact Muon Solenoid”, **World Wide Web:** <http://cmsinfo.cern.ch/Welcome.html>.
- [7] WIGMANS, R., “Advances in Hadron Calorimetry”, *Submitted to the Annual Review of Nuclear and Particle Science*, , 1991.
- [8] BOCK, R., A.VASCILEU, *The Particle Detector BriefBook*. Springer, 1999.
- [9] ATLAS/Tile Calorimeter Collaboration, *Tile Calorimeter - Technical Design Report*, Report, CERN, Dezembro 1996.
- [10] SEIXAS, J., CERQUEIRA, A., CALÔBA, L., “A High-Performance Active Adder”, *IEEE International Conference on Electronics, Circuits and Systems*, pp. 377–380, 1998. Lisboa, Portugal.
- [11] ATLAS Collaboration, *ATLAS DAq, EF, LVL2 and DCS Technical Progress Report*, Report, CERN, 1998.

- [12] ATLAS Collaboration, *Trigger Performance Technical Progress Report*, Report, CERN, 1999.
- [13] BOCK, R., KASTRUP, B., *Realistic Calorimeter Feature Extraction: Algorithm, Benchmarks and Implementation options*, ATLAS DAQ Note 65, CERN, Março 1997.
- [14] ANJOS, A., BOCK, R., BOGAERTS, A., *et al.*, *Software Framework and Testbed Architectures*, Testbed Technical Note 1, CERN, 1998.
- [15] ANJOS, A., HAUSER, R., SESSLER, M., *et al.*, *Requirements, Constraints, Use-Cases and an Object Oriented Analsis of the Steering and Feature Extrac-tion Systems for ATLAS/Trigger Level 2*, Testbed Technical Note 8, CERN, 1998. **Disponível em inglês:** <http://www.lps.ufrj.br/~rabello/publications/index.html>.
- [16] ANJOS, A., HAUSER, R., SESSLER, M., *et al.*, *OO Design of the Steering System*, Testbed Technical Note 17, CERN, July 1998. **Disponível em inglês:** <http://www.lps.ufrj.br/~rabello/publications/index.html>.
- [17] ANJOS, A., HAUSER, R., SESSLER, M., *et al.*, *Design of the Feature Extrac-tion System*, Testbed Technical Note 20, CERN, 1998. **Disponível em inglês:** <http://www.lps.ufrj.br/~rabello/publications/index.html>.
- [18] ANJOS, A., *Implementation of the Steering and Calorimeter systems for the Reference Software (status note)*, Testbed Technical Note 25, CERN, 1998. **Disponível em inglês:** <http://www.lps.ufrj.br/~rabello/publications/index.html>.
- [19] ANJOS, A., SEIXAS, J., “Integrando Plataformas e Algoritmos para o Segundo Nível de Trigger do Experimento ATLAS”, *Encontro Nacional de Física de Partículas e Campos*, , Outubro 1999. São Lourenço, Minas Gerais, **Disponível em português:** <http://www.lps.ufrj.br/~rabello/publications/index.html>.
- [20] BOCK, R., CARTER, J., LEGRAND, I., *What can Artificial Neural Networks do for the Global Second Level Trigger*, ATLAS/DAQ Note 11, CERN, Março

1994. **Disponível em inglês:** <http://atlasinfo.cern.ch/Atlas/GROUPS/DAQ-TRIG/NOTES/note11/daq-11.ps>.
- [21] SEIXAS, J., CALÔBA, L., SOUZA, M., *et al.*, “Neural Second-Level Trigger System Based on Calorimetry”, *Computer Physics Communications*, v. 95, pp. 143–157, 1996.
 - [22] HAYKIN, S., *Neural Networks: A comprehensive foundation*. Prentice Hall, 1999.
 - [23] ROGERS, J., *Object-Oriented Neural Networks in C++*. Academic Press, 1997.
 - [24] SEIXAS, J., CALÔBA, L., SOUZA, M., *et al.*, “Neural Networks Applied to a Second Level Trigger System Based on Calorimeters”, *Third International Workshop on Software Engineering*, pp. 353–358, 1993. Artificial Intelligence and Expert Systems for High Energy and Nuclear Physics, Oberammergau, Alemanha.
 - [25] WIDROW, B., SEARNS, S., *Adaptive Signal Processing*. Prentice Hall, 1985.
 - [26] TREES, H. V., *Detection, Estimation and Modulation*. John Wiley & Sons, 1966.
 - [27] GONZALEZ, S., HANSL-KOZANECKA, T., WIELERS, M., *Selection of high- p_T electromagnetic clusters by the level-2 trigger of ATLAS*, ATLAS-Trigger DAq Note 002, CERN, Fevereiro 2000.
 - [28] HERTZ, J. A., *Introduction to the Theory of Neural Computation*. Addison-Wesley, 1991.
 - [29] PETERSON, C., RÖGNVALDSSON, T., *JETNET 3.0 A versatile artificial neural network package*. CERN, December 1993.
 - [30] SEIXAS, J. M., KASTRUP, B., CALÔBA, L. P., *et al.*, “Neural Feature Extraction for Calorimeters in a Distributed Processing Environment”, , 1997.
 - [31] ANJOS, A., SEIXAS, J., “Neural Electron-Jet Separation Using Multi-layered and Multi-segmented Calorimeters”, *XXI Encontro Nacional de*

- Física de Partículas e Campos*, , Outubro 2000. **Disponível em inglês:**
<http://www.lps.ufrj.br/~rabello/publications/index.html>.
- [32] SEIXAS, J., ANJOS, A., DANTAS, A., *et al.*, “Um Protótipo do Sistema de Validação de Nível 2 para as Condições do LHC”, *XIX Encontro Nacional de Física de Partículas e Campos*, pp. 32–33, Dezembro 1998.
 - [33] SEIXAS, J., ISIDRO, I., “A Neural Electron/Jet Discriminator Based on a Highly-Segmented Calorimeters”, *Artificial Intelligence in High-Energy and Nuclear Physics (AIHENP)*, , 1999.
 - [34] VASSALI, M., SEIXAS, J., “Principal Component Analysis for Neural Electron/Jet Discrimination on Highly Segmented Calorimeters”, *VII International Workshop on Advanced Computing & Analysis Techniques in Physics Research*, , October 2000.
 - [35] GRUBER, A., MöCK, J., FRÖCHTENICHT, J., *et al.*, “Performance of Back-propagation mentworks in the Second Level Trigger of the H1-Experiment”, *New Computing Techniques in Physics Research*, , 1994.
 - [36] MITRA, S. K., *Digital Signal Processing: A Computer-Based Approach.* , 1998.
 - [37] S. GONZALEZ, B. GONZALES-PIÑERO, T. S., *First Implementation of Calorimeter FEX Algorithms in the Level-2 Reference Software*, Lvl2 testbed technical note, CERN, Maio 1999. **Usuários AFS podem pegar uma cópia em:** [/afs/cern.ch/atlas/project/LVL2testbed/www/doc/physics/calor/notes/-caloalgo.ps](http://afs/cern.ch/atlas/project/LVL2testbed/www/doc/physics/calor/notes/-caloalgo.ps).
 - [38] BOOCH, G., *Object Oriented Analysis and Design with Applications*. Addison and Wesley, 1994.
 - [39] GAMMA, E., HELM, R., JOHNSON, R., *et al.*, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994.
 - [40] KERNIGHAN, B. W., RITCHIE, D. M., *C, a linguagem de programação: padrão ANSI*. 2 ed. Editora Campus, 1990.

- [41] STROUSTRUP, B., *The C++ Programming Language*. 3 ed. Murray Hill, New Jersey, Addison-Wesley, 1997.
- [42] *ADSP-2106x Sharc User's Manual*. Segunda edição ed. Analog Devices.