

EE-613: Machine Learning for Engineers

Practical session 10

(Multi-layer perceptron)

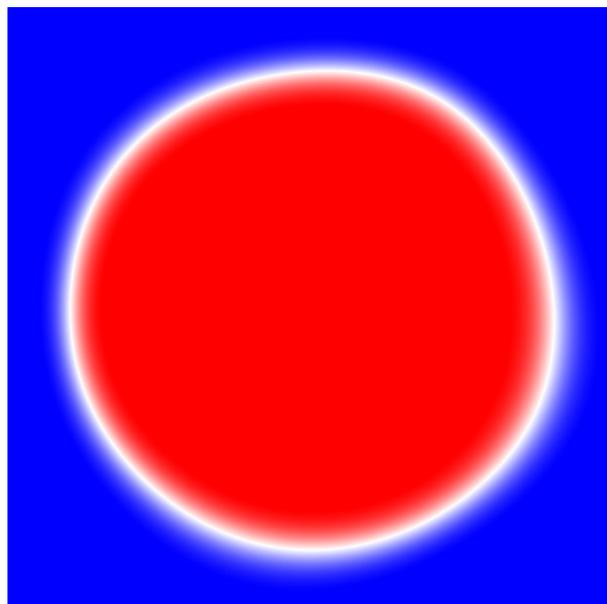
1 Download and test

```
1 | wget http://www.idiap.ch/~fleuret/files/EE613/EE613-pw10.tgz
2 | tar zxvf EE613-pw10.tgz
3 | cd EE613/pw10
4 | ./do.sh example
```

after a few seconds, you should have obtained at the bottom of the console

```
LOSS 33 375.619
LOSS 34 375.594
TRAIN_LOSS 727.659
TEST_LOSS 719.488
Writing MLP activation image ... done. Wrote "example.png".
```

and the following image should be displayed



2 Programming

2.1 Compilation

You can compile the source code with `make`. Running the `do.sh` script will always recompile the source if the executable is not up-to-date.

It is suggested to implement each question its corresponding function

```
1 | void function_question1() {  
2 |     ...  
3 | }
```

so that the `do.sh` script can run it and call the appropriate visualizing tools.

2.2 Classes

A **DataSet** Contains the input and output vectors corresponding to a set of samples.

```
1 | class DataSet {  
2 | public:  
3 |     int input_dim, output_dim;  
4 |     int nb_samples;  
5 |     scalar_t **input_activations;  
6 |     scalar_t **output_activations;  
7 |  
8 |     DataSet(int input_dim, int output_dim, int nb_samples);  
9 |     ~DataSet();  
10 | };
```

A **MLP** is a simple implementation of a multi-layer perceptron.

```
1 | class MLP {  
2 | public:  
3 |  
4 |     MLP(TransferFunction *transfer_function,  
5 |         LossFunction *loss_function,
```

```
6         int input_dim, int nb_layers, const int *layer_dim);
7
8     ~MLP ();
9
10    void randomize_weights_and_biases (scalar_t epsilon);
11
12    void train (DataSet *train_set);
13
14    void test (scalar_t *input_activations,
15              scalar_t *output_activations);
16 };
```

3 Questions

Question 1: XOR with and without hidden layer

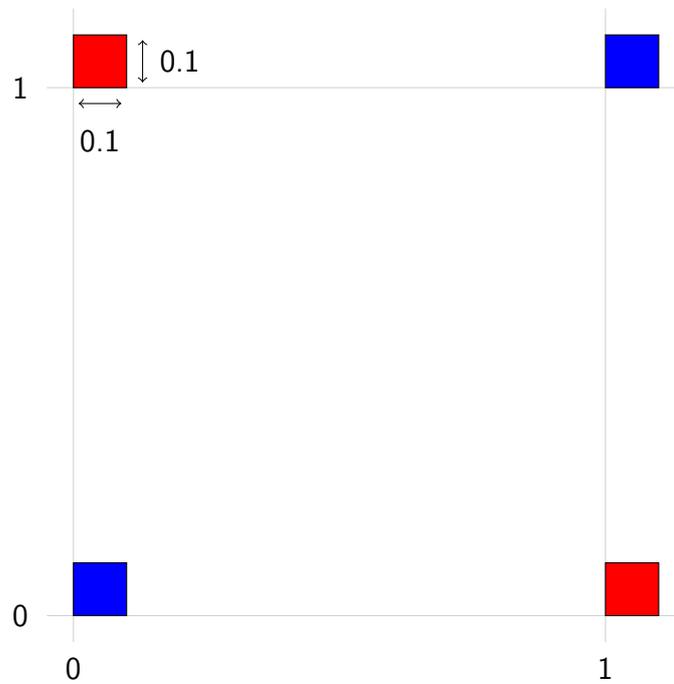
Generate a training set

$$(x_n, y_n) \in \mathbb{R}^2 \times \{-1, 1\}, n = 1, \dots, 1000$$

with

$$P(Y = 1) = P(Y = -1) = \frac{1}{2}$$

and $X|Y=1$ (resp. $X|Y=-1$) uniform over the two blue squares (resp. red squares) on the scheme below.



Train a MLP with no hidden layer, and a MLP with two hidden units, print the train loss in both cases.

Help:

Use the `random_0_1()` function to generate scalar uniformly distributed in $[0, 1]$.

Question 2: Real function regression

Generate a training set

$$(x_n, y_n) \in \mathbb{R}^2, n = 1, \dots, 1000$$

with $x_n \simeq \mathcal{U}[-\pi, \pi]$ and $y_n = \sin(x_n)$.

Train a MLP with 100 hidden units, compute the prediction for all the values

$$x = \pi \left(\frac{n-1}{500} - 1 \right), n = 1, \dots, 1000$$

and write in a file `prediction.dat` the pairs of test `xs` and `ys`, one pair per line, separated by a space.

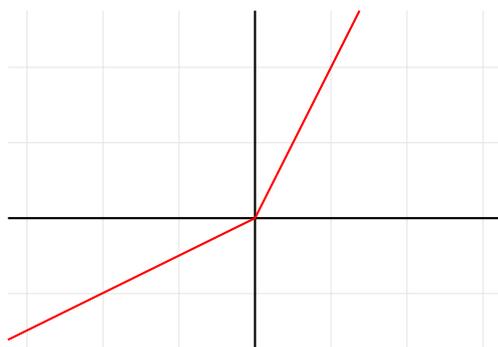
Question 3: Different transfer function and loss

Create two classes

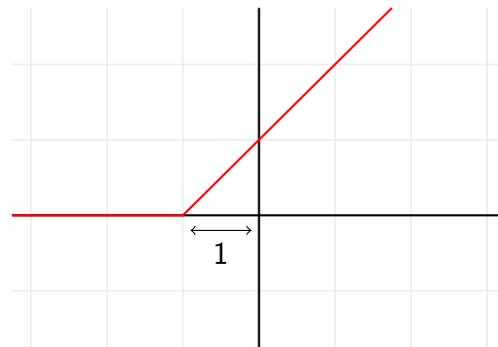
```
class TransferFunctionDoubleRectified : public TransferFunction {  
    ...  
}
```

```
class LossFunctionHinge : public LossFunction {  
    ...  
}
```

that implement respectively the double rectified transfer function and the hinge loss



Double rectified transfer



Hinge loss

Run an experiments similar to the example one with a MLP using these two classes.