Not All Samples Are Created Equal
Deep Learning with Importance Sampling

Angelos Katharopoulos & Françoís Fleuret

ICML, July 11, 2018

Funded by
Evolution of gradient norms during training

Small CNN on MNIST

Gradient norm CDF

Training loss

Gradient norm

Probability
Evolution of gradient norms during training

Small CNN on MNIST

Gradient norm CDF
Evolution of gradient norms during training

Small CNN on MNIST

Gradient norm CDF
Evolution of gradient norms during training

Small CNN on MNIST

Gradiente norm CDF

85% of the samples have negligible gradient
Related work

- Sample points proportionally to the gradient norm (Needell et al., 2014; Zhao and Zhang, 2015; Alain et al., 2015)
- SVRG type methods (Johnson and Zhang, 2013; Defazio et al., 2014; Lei et al., 2017)
- Sample using the loss
  - Hard/Semi-hard sample mining (Schroff et al., 2015; Simo-Serra et al., 2015)
  - Online Batch Selection (Loshchilov and Hutter, 2015)
  - Prioritized Experience Replay (Schaul et al., 2015)
Related work

- Sample points proportionally to the gradient norm \cite{needell2014, zhao2015, alain2015}
- SVRG type methods \cite{johnson2013, defazio2014, lei2017}
- Sample using the loss
  - Hard/Semi-hard sample mining \cite{schroff2015, simo2015}
  - Online Batch Selection \cite{loshchilov2015}
  - Prioritized Experience Replay \cite{schaul2015}
Contributions

- Derive a fast to compute importance distribution
- Variance cannot always be reduced so start importance sampling when it is useful
Contributions

- Derive a fast to compute importance distribution
- Variance cannot always be reduced so start importance sampling when it is useful
- Package everything in an embarrassingly simple to use library
Deriving the sampling distribution

Similar to Zhao and Zhang (2015) we want to minimize the variance of the gradients.

$$P^* = \arg \min_P \text{Tr} \left( \nabla_P [w_i G_i] \right) = \arg \min \mathbb{E}_P \left[ w_i^2 \| G_i \|_2^2 \right]$$

To simplify, we minimize an upper bound

$$\| G_i \|_2 \leq \hat{G}_i \iff \min_P \mathbb{E}_P \left[ w_i^2 \| G_i \|_2^2 \right] \leq \min_P \mathbb{E}_P \left[ w_i^2 \hat{G}_i^2 \right]$$
Deriving the sampling distribution \(^{(1)}\)

Similar to Zhao and Zhang (2015) we want to minimize the variance of the gradients.

\[ P^* = \arg\min_P \text{Tr} (\mathbb{V}_P [w_i G_i]) = \arg\min_P \mathbb{E}_P \left[ w_i^2 \| G_i \|_2^2 \right] \]

To simplify, we minimize an upper bound

\[ \| G_i \|_2 \leq \hat{G}_i \iff \min_P \mathbb{E}_P \left[ w_i^2 \| G_i \|_2^2 \right] \leq \min_P \mathbb{E}_P \left[ w_i^2 \hat{G}_i^2 \right] \]
Deriving the sampling distribution \(^{(1)}\)

Similar to Zhao and Zhang (2015) we want to minimize the variance of the gradients.

\[
P^* = \arg \min_P \text{Tr} (\nabla_P [w_i G_i]) = \arg \min_P \mathbb{E}_P \left[ w_i^2 \| G_i \|_2^2 \right]
\]

To simplify, we minimize an upper bound

\[
\| G_i \|_2 \leq \hat{G}_i \iff \min_P \mathbb{E}_P \left[ w_i^2 \| G_i \|_2^2 \right] \leq \min_P \mathbb{E}_P \left[ w_i^2 \hat{G}_i^2 \right]
\]
Deriving the sampling distribution \(^{(1)}\)

Similar to Zhao and Zhang (2015) we want to minimize the variance of the gradients.

\[
P^* = \arg \min_P \text{Tr} (\nabla_P [w_i G_i]) = \arg \min \mathbb{E}_P [w_i^2 \|G_i\|_2^2]
\]

To simplify, we minimize an upper bound

\[
\|G_i\|_2 \leq \hat{G}_i \iff \min_P \mathbb{E}_P [w_i^2 \|G_i\|_2^2] \leq \min_P \mathbb{E}_P [w_i^2 \hat{G}_i^2]
\]
We show that we can upper bound the gradient norm of the parameters using the norm of the gradient with respect to the pre-activation outputs of the last layer.

We conjecture that batch normalization and weight initialization make it tight.
Variance reduction achieved with our upper-bound
Variance reduction achieved with our upper-bound
Variance reduction achieved with our upper-bound

![Graph showing empirical variance reduction for Downsampled Imagenet with uniform loss and gradient-norm over iterations.]
Variance reduction achieved with our upper-bound

![Graph showing variance reduction](image)

- **Empirical variance reduction**
- **Downsampled Imagenet**
- **Uniform**
- **Loss**
- **Gradient-norm**
- **Upper-bound (ours)**
Is the upper-bound enough to speed up training?

Not really, because
- a forward pass on the whole dataset is still prohibitive
- the importance distribution can be arbitrarily close to uniform

Two key ideas
- Sample a **large batch** \((B)\) randomly and resample a **small batch** \((b)\) with importance
- Start importance sampling when the variance will be reduced
When do we start importance sampling?

We start importance sampling when the variance reduction is large enough

\[
\text{Tr} (\nabla_u [G_i]) - \text{Tr} (\nabla_P [w_i G_i]) = \frac{1}{B} \sum_{i=1}^{B} \| G_i \|_2^2 \sum_{i=1}^{B} (p_i - u)^2 \propto \sum_{i=1}^{B} (p_i - u)^2
\]

distance of importance distribution to uniform
When do we start importance sampling?

We start importance sampling when the variance reduction is large enough

\[
\text{Tr} (\nabla_u G_i) - \text{Tr} (\nabla_P w_i G_i) = \frac{1}{B} \sum_{i=1}^{B} \| G_i \|^2 \sum_{i=1}^{B} (p_i - u)^2 \propto \sum_{i=1}^{B} (p_i - u)^2
\]

We show that the **equivalent batch increment** \( \tau \geq \left( 1 - \frac{\sum_i (p_i - u)^2}{\sum_i p_i^2} \right)^{-1} \) which allows us to perform importance sampling when

\[
Bt_{\text{forward}} + b(t_{\text{forward}} + t_{\text{backward}}) \leq \tau(t_{\text{forward}} + t_{\text{backward}})b
\]

\( Bt_{\text{forward}} \) Time for importance sampling iteration

\( Bt_{\text{forward}} + b(t_{\text{forward}} + t_{\text{backward}}) \) Time for equivalent uniform sampling iteration
Experimental setup

- We fix a time budget for all methods and compare the achieved training loss and test error.
- We evaluate on three tasks:
  1. WideResnets on CIFAR10/100 (image classification task)
  2. Pretrained ResNet50 on MIT67 (finetuning task)
  3. LSTM on permuted MNIST (sequence classification task)
Importance sampling for image classification

SVRG methods do not work for Deep Learning

Our loss-based sampling outperforms existing loss-based methods

Improvement from $3 \times 10^2$ to $10 \times 10^2$ compared to training loss with uniform sampling

Training loss relative to uniform

Test error relative to uniform

CIFAR-10  CIFAR-100

CIFAR-10  CIFAR-100

CIFAR-10  CIFAR-100
Importance sampling for image classification

- SVRG methods do not work for Deep Learning

![Training loss relative to uniform](image1)

![Test error relative to uniform](image2)

CIFAR-10 CIFAR-100
Importance sampling for image classification

- SVRG methods do not work for Deep Learning
- Our loss-based sampling outperforms existing loss-based methods
Importance sampling for image classification

- SVRG methods do not work for Deep Learning
- Our loss-based sampling outperforms existing loss-based methods
- Improvement from $3 \times$ to $10 \times$ compared to training loss with uniform sampling
Importance sampling for finetuning

- Earlier variance reduction leads to faster convergence

![Graphs showing training loss and test error over time with different variance reduction methods.](image-url)
Thank you for your time!

Check out the code at http://github.com/idiap/importance-sampling.

```python
from importance_sampling import ImportanceTraining
x, y = load_data()
model = load_model()
ImportanceTraining(model).fit(x, y, batch_size=128, epochs=10)
```


