Eurographics Workshop on Visual Computing for Biology and Medicine (2023)
T. Höllt and D. Jönsson (Editors)

Short Paper

CDF-Based Importance Sampling and Visualization for Neural Network Training

Alex Knutsson†, Jakob Unnebäck†, Daniel Jönsson, and Gabriel Eilertsen

Linköping University, Sweden

Abstract

Training a deep neural network is computationally expensive, but achieving the same network performance with less computation is possible if the training data is carefully chosen. However, selecting input samples during training is challenging as their true importance for the optimization is unknown. Furthermore, evaluation of the importance of individual samples must be computationally efficient and unbiased. In this paper, we present a new input data importance sampling strategy for reducing the training time of deep neural networks. We investigate different importance metrics that can be efficiently retrieved as they are available during training, i.e., the training loss and gradient norm. We found that choosing only samples with large loss or gradient norm, which are hard for the network to learn, is not optimal for the network performance. Instead, we introduce an importance sampling strategy that selects samples based on the cumulative distribution function of the loss and gradient norm, thereby making it more likely to choose hard samples while still including easy ones. The behavior of the proposed strategy is first analyzed on a synthetic dataset, and then evaluated in the application of classification of malignant cancer in digital pathology image patches. As pathology images contain many repetitive patterns, there could be significant gains in focusing on features that contribute stronger to the optimization. Finally, we show how the importance sampling process can be used to gain insights about the input data through visualization of samples that are found most or least useful for the training.

CCS Concepts

• ***Computing methodologies*** → ***Neural networks;*** • ***Human-centered computing*** → ***Visualization techniques;***

1. Introduction

Deep neural networks (DNNs) are trained to learn complex patterns and relationships from large amounts of data. Deep learning has been shown to achieve high accuracy on a wide range of tasks, from detecting cancer metastases in digital pathology whole slide images [BVvDea17, KGG20] to segmenting objects for autonomous driving [GTCM20]. However, the training process involves solving an optimization problem that is computationally expensive, especially when dealing with large volumes of data and large models. At the same time, training samples contribute differently to the model performance, so that choosing samples that improve the model performance the most in each training iteration can allow for obtaining an as good or better model with less training [KF18]. However, it is not possible to know beforehand exactly which samples will contribute the most. Furthermore, the strategy for determining which samples to use must be computationally efficient to be beneficial. Previous work has applied input data importance sampling with the gradient norm of the loss function [KF18] as a measure of importance. While this metric is directly associated with the optimization in training, it is also expensive to compute.

We propose an importance sampling strategy which focuses on difficult samples while still incorporating some of the easy samples. By doing so, we prevent the model from "forgetting" how to solve the easy problems – similar to the phenomena of catastrophic forgetting [TSC*18] – while gradually focusing more on the hard problems. In practice, we compute the cumulative distribution function (CDF) of the importance metric and sample inputs from this distribution. This results in obtaining more samples with high importance while still maintaining a few samples with lower importance. It also means that we can use the number of times a sample has been picked during the optimization as a measure of overall importance. Using this information, we can visualize input samples with low and high contribution to the training.

An issue not addressed in previous work is that the network weights are most often randomly initialized. This means that metrics such as loss or gradient norm are more or less random and can be detrimental when used for importance sampling. Before using such metrics, it is necessary to first train the network. We present an empirical study on the number of training iterations required before importance metrics provide meaningful information.

We perform experiments on both synthetic data and on a digital pathology dataset. The synthetic dataset is used to analyze the be-

† Authors contributed equally to this work

DOI: 10.2312/vcbm.20231212

havior of the importance sampling, while digital pathology is our application of interest. In digital pathology, the images present a large degree of repetitions, and differences between classes can be subtle. We are interested in exploring if importance sampling can aid in speeding up the training process and gain insight into what features are most relevant for learning to separate between healthy and tumorous tissue. The main contributions of this work are:

- A novel importance sampling strategy that incorporates both easy and hard samples in the training process.
- An empirical study of when to start using loss/gradient norm-based importance sampling for randomly initialized weights.
- A method for visualizing which input samples are considered important/unimportant during training.
- An analysis of using importance sampling for the purpose of detecting malignant tumors in digital pathology data.

2. Related Work

Loshchilov and Hutter [LH15] present a scheme for batch selection based on the history of already evaluated loss values. However, this runs the risk of using values that are not representative of the current stage of training. Katharopoulos et al. [KF18] present multiple techniques for finding important samples using the gradient norm of the loss function. They conclude that the upper bound of the gradient norm of any neural network can be computed in a single forward pass. This, in turn, means that it gives a net positive to training performance. Liu et al. [LWM20] implement importance sampling using a multi-armed bandit algorithm, balancing the computational cost of exploring samples to include in training and the reward of adding their contribution to optimization. Johnson and Guestrin [JG18] propose a robust, approximate, importance sampling procedure (RAIS) using stochastic gradient descent. The method is used to find min-points and prevent overshoot. This can be applied when looking for samples where the loss function is reduced as much as possible. Using their implementation of importance sampling the training phase is sped up by 20%.

A related area of research is active learning, where samples are selected from a pool of unlabeled data and then labeled by a user/oracle [Set09, RXC*21]. Although techniques based on, e.g., gradient norm can be used for this purpose as well, there is a fundamental difference compared to importance sampling, where sampling is performed over the already labeled data points.

Although previous work on importance sampling in deep learning has demonstrated great potential to speed up training, it is not clear if selecting only the most important samples is the best strategy. There could be potential benefits in also including samples of less importance to stabilize the training, which we explore in this work using a CDF-based formulation. Furthermore, we apply the techniques in digital pathology, where to our knowledge the benefits of importance sampling has not yet been explored.

3. Primer on Neural Network Input Importance Sampling

The core idea of importance sampling for DNN training is to pick samples that improve the model the most. The only way to get a specific sample's true importance is to train the network on that particular sample and measure the improvement. However, doing so would not improve the efficiency of the training process due to the computational overhead. Therefore, we have to rely on more cost-efficient approximations of importance. Here, we provide a quick overview of the basic loss and gradient norm sampling techniques described in more detail in [KF18].

Loss Sampling: In theory, samples with high loss have a large impact on the network. Thus, these samples should increase the model accuracy more than samples with lower loss, and the loss is therefore used as a measurement of how important a sample is. The benefit of this method is that it does not require a backward pass as the loss is returned from the forward pass making this sampling method cost-efficient. However, the per-sample loss needs to be calculated as opposed to a batch loss.

Gradient Norm Sampling: The gradient of a single-layer neural network is straightforward to compute since the error can be directly computed as a function of the weights. However, with more complex networks, the loss becomes a complicated composition function of the weights in earlier layers, requiring the backpropagation algorithm [Agg18]. In the backward pass, the loss function is used to update the weights, from which the gradient norm of the updated model parameters can be computed. To get a per-sample gradient norm, the backward pass has to be computed on a per-sample basis. Otherwise, there is no way to distinguish the importance of individual samples but rather the importance of the batch. A gradient with a small magnitude means that the loss function is relatively flat in that region and updates to the parameters will not have a significant effect on the loss. Conversely, a large gradient magnitude indicates that the loss function is steep in that region and that the parameters will be updated more significantly.

4. CDF-based Importance Sampling

Choosing samples solely based on highest loss or gradient norm can be problematic as they are only approximations of the true importance. Such sample selection consequently risks excluding samples actually important for the training. It would therefore be beneficial to choose many samples that are believed to improve the performance the most while still including other samples. For this purpose, we choose samples according to the cumulative distribution function of the loss and the gradient norm.

4.1. CDF Sampling

For a continuous variable x, the cumulative distribution function (CDF) is defined as the integral of the probability density function (PDF) of the distribution $f(t)$ from negative infinity to x:

$$F(x) = \int_{-\infty}^{x} f(t)dt. \tag{1}$$

In the context of importance sampling, we use the CDF to determine the probability of selecting a particular sample from the dataset. More specifically, we compute the importance of each sample K in a large batch. Then, we draw N samples following the CDF to form a mini batch for optimization, where $N < K$. This CDF-based sampling strategy makes it more likely to include samples

with high importance while still including samples with low importance, although with lower probability. Figure 1 demonstrates how our CDF sampling strategy compares to only choosing samples with the highest importance for an inside/outside circles classification problem. In the 1st and 3rd rows, only samples close to the boundary are chosen, which means that the model might forget how to deal with samples from other parts of the input space. The CDF sampling strategy, 2nd and 4th rows, instead feeds the optimization with a wider spread of samples; still sampling more in areas with higher importance around the edges of the circles.

Increasing K yields better importance sampling as there is a higher likelihood of important samples appearing in the large batch while simultaneously costing more computations and memory. Furthermore, if the N is too close to K, there will be little difference compared to uniform random sampling and the gain of the important sampling will be foreshadowed by the computational overhead. In this work, we use $K = 1024$ and $N = 128$ based both on the results in [KF18] as well as good performance in internal tests.

4.2. Importance Sampling Start Criterion

While most previous work consider importance sampling from the initiation of the training process, we observe cases where this causes unwanted behavior. If the network weights are randomly initialized, the notion of importance based on the loss or gradient norm can lead the optimization to focus on random subgroups of samples that hamper the learning. This can have a detrimental impact on the optimization, counteracting the benefits of importance sampling. To overcome this problem, we start the optimization using uniform sampling and introduce a criterion for switching to importance-based sampling. This is defined by the fraction between the current loss and the initial loss. For instance, a threshold of 0.5 would imply that the importance sampling is initiated once the mini batch loss is smaller than half of the original loss. In our experiments in Section 5, we use different selections of thresholds to evaluate the impact of the initiation criteria on the optimization.

4.3. Sample Importance Visualization

As our importance is an approximation of how much the training should improve if the sample is included, we can use it as a way to visualize which samples that are most important to the training.

The importance of each sample in the dataset is constantly changing during training. Some samples might be difficult early in the training process, but not at a later stage. Therefore, there are many different ways of analyzing the training process with respect to importance of samples over time. For simplicity, we here focus on having a single indicator over the entire training process. Aggregates, such as the sum or average of the sample importances can be used for this purpose. However, these can be dominated by single instances of high importance throughout the training. Therefore, to visualize how the model utilizes the samples during the training process, we count the number of times a sample is included and use this as a measure of overall impact. The counting approach treats samples equally independent of if it is early or late in the training phase and avoids emphasis on single high-importance peaks. Displaying the most important and unimportant samples is simplistic

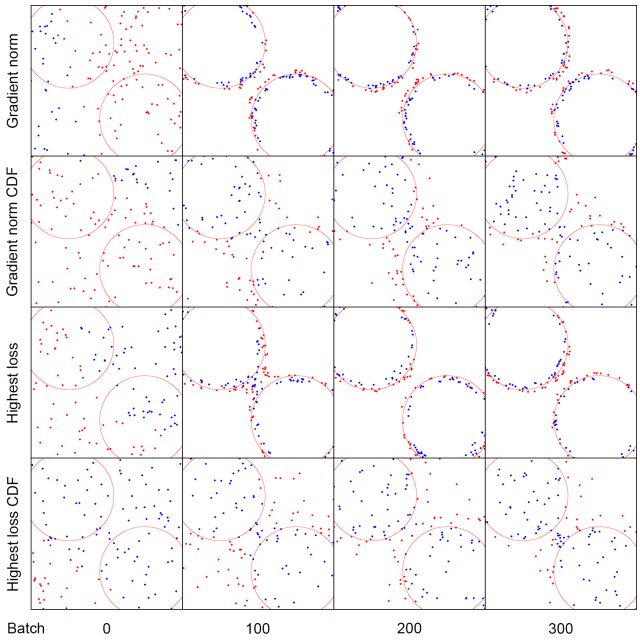

Figure 1: *Examples of sampling patterns captured during the training progress for the Two Circles dataset. Blue and red dots correspond to inside and outside predictions, respectively. Batch zero reflects the weight initialization. The* gradient norm *and* highest loss *techniques only place samples along the difficult regions, which prevents exploration of the whole search space. The CDF-based techniques, on the other hand, choose a broader range of samples across the whole space while still focusing on the edges.*

but can nevertheless reveal interesting insights about what types of input data could improve the model even more or what the model has difficulties with.

5. Results

Results are demonstrated on one synthetic dataset in 2D and on a digital pathology dataset. Additional results on the MNIST and CIFAR-10 datasets are available in the supplementary material.

5.1. Datasets

Two Circles is a synthetically generated binary classification problem designed to facilitate visual comparison of different sampling strategies. Two circles define the boundaries between two classes, with one class being inside either of the two circles and the other being the outer region. The area of the two classes is the same, meaning that there is an equal probability of sampling from either of the classes given a uniform sampling within the square.

Camelyon [BVvDea17] consists of whole slide images (WSIs) of sentinel lymph node sections collected in the Netherlands. We use the PatchCamelyon version of Camelyon [VLW*18], with 262K training patches sampled in 96×96 pixels resolution from the WSIs. Images are separated in two classes with equal number of images, one with healthy tissue and one with cancer metastases. Testing is performed on the test set comprised of 32K images.

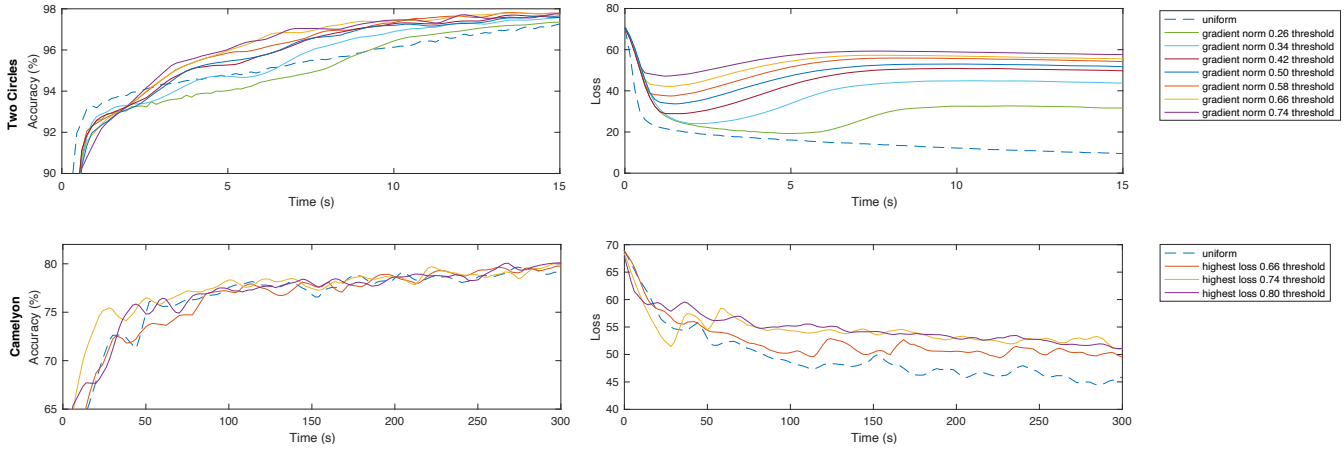

Figure 2: *Test accuracy and training loss for different thresholds determining when to start importance sampling. Lower thresholds are advantageous for the simple Two Circles problem, while higher thresholds yields superior performance on the more complex Camelyon data.*

5.2. Training Setup

Experiments[‡] were conducted using Pytorch on a MacBook Pro M1 Max with 64GB unified memory, a 12 core CPU and 32 core GPU. Performance is averaged across 5-10 successive runs to ensure consistent results. Note that the random nature of the training still means that the baseline can vary between figures. We used the Adam optimizer, whereas networks differ between datasets:

- The Two Circles network is trained with the negative log-likelihood loss and learning rate 0.005. It consists of two linear layers with ReLU activation. The input and output size is two and the hidden layer uses 50 neurons, for at total of 252 weights.
- The Camelyon network is trained with the cross entropy loss and learning rate 0.001. It consists of two convolutional layers, two max-pooling layers, one dropout 2D layer, and two linear layers. The total number of weights is 446,932.

5.3. Analysis of Importance Sampling Behavior

The Two Circles synthetic 2D dataset allows for easy inspection of how the importance sampling schemes prioritize different regions of the data distribution. In Figure 1, the selected samples for every 100 batches can be compared with and without our CDF-based sampling strategy. It can be seen that the gradient norm method and most loss method both focus the sample area to the border of the circles, implying that the highest-loss samples are in those areas. The CDF version of the methods expands the range of interest and also includes samples further from the circle border. A uniform sampling method would continuously sample points randomly over the area instead, giving all areas of the frame equal importance.

5.4. When To Start Importance Sampling?

In Figure 2, we investigate the optimal sampling threshold, e.g., how much the loss needs to decrease before starting to apply importance sampling. Uniform sampling is used as a baseline comparison, indicated by a blue dashed line in all figures.

For the Two Circles dataset it is not beneficial to wait long before applying importance sampling. All tested thresholds outperform uniform sampling at the end of the training with higher accuracy, but the lower threshold of 0.26 takes the longest time to overtake uniform sampling. The loss graph displays a distinct branching pattern at the designated threshold point. The reason is that more samples with high loss are selected for training, consequently resulting in increased loss. For the Camelyon dataset, the higher thresholds (0.74 and 0.80) are beneficial early in training. The difference between the thresholds is less clear later during the training.

5.5. Importance Sampling Strategy Comparison

We used the best thresholds from the analysis of when to start importance sampling. The result of this evaluation can be seen in Figure 3. For the Two Circles (top row), we can see that the *highest loss* and *highest loss CDF* perform best. The *gradient norm* sampling strategy is too slow to compute, and the additional computations needed for the CDF-based method is too costly to introduce benefits for this small models and dataset.

For the significantly larger model trained on the Camelyon dataset (bottom row), we can see that the CDF-based sampling result in significantly higher accuracy compared to not using the CDF-based sampling for the *highest loss* and *gradient norm*. The *gradient norm CDF* is best at the beginning of the training but is later overtaken by the *highest loss CDF* strategy.

5.6. Important Training Samples

Samples that are considered important and unimportant for training on the Camelyon dataset are shown in Figure 4. The most important samples tend to be more difficult and more unique compared to the least important samples, which tend to be easy to classify. For example, we can observe that the most important samples showcase irregular and complex content, which are more challenging to

[‡] Code available at: github.com/3DJakob/ai-importance-sampling

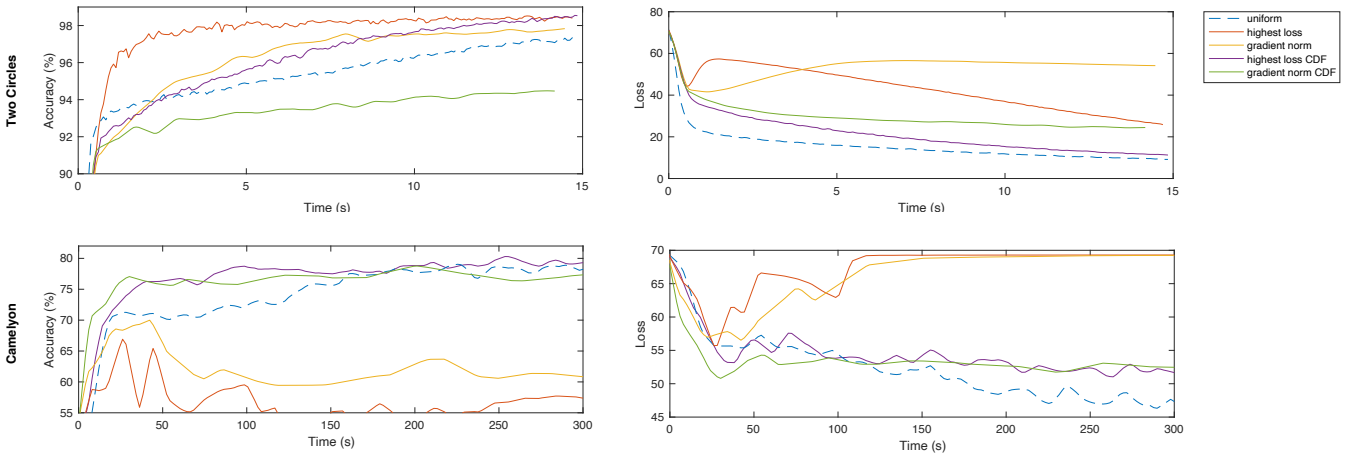

Figure 3: *Graphs depicting the test accuracy and training loss for different sampling methods. A higher accuracy than the dashed line, representing uniform sampling, means better performance and efficiency. Highest loss and gradient norm correspond to [KF18].*

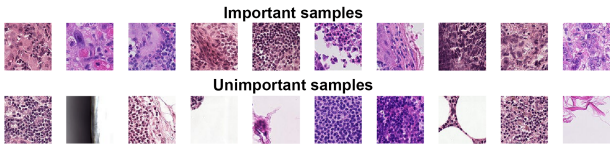

Figure 4: *The most and least important training samples from the Camelyon dataset.*

classify. On the other hand, the unimportant samples either contain larger areas of fat tissue, background or experience more regular cellular patterns that are easier to classify.

6. Conclusions

In this work, we presented a new importance sampling strategy for training DNNs. This builds on previous work that compute the importance based on the loss of each sample. However, instead of choosing the samples with the highest loss for a training batch, we sample from the CDF of the loss. We show how our sampling strategy avoids overemphasis on a subspace of the input data, e.g., the edges in the Two Circles synthetic dataset. Furthermore, we demonstrate that our technique outperforms previous importance sampling schemes when used for tumor classification in digital pathology data, and is significantly faster at reaching high accuracy compared to uniform sampling. However, determining when to start importance sampling using a loss-based threshold is tricky and still has a relatively large impact on performance at the beginning of the training. Here, further research is needed to determine better when to start importance sampling. Nevertheless, importance sampling has a large potential to avoid excessive training on the redundant features commonly seen in this type of data. Our work can potentially also be used in strategies for dataset pruning or subsampling for training using a smaller dataset with little or no reduction in performance. Finally, we also demonstrate how our method can be used to visualize which samples that are most and least important to the training. Such visualization can aid in understanding the training data and we show examples indicating that samples with unique features are especially important to the training.

Acknowledgments: This work was supported by the Zenith career development program at Linköping University and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

[Agg18] AGGARWAL C.: Neural networks and deep learning: A textbook. In *Neural Networks and Deep Learning*. Springer, 2018. 2

[BVvDea17] BEJNORDI B. E., VETA M., VAN DIEST P. J., ET AL.: Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *JAMA 318*, 22 (2017), 2199–2210. 1, 3

[GTCM20] GRIGORESCU S., TRASNEA B., COCIAS T., MACESANU G.: A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics 37*, 3 (2020), 362–386. 1

[JG18] JOHNSON T. B., GUESTRIN C.: Training deep models faster with robust, approximate importance sampling. *Advances in Neural Information Processing Systems 31* (2018). 2

[KF18] KATHAROPOULOS A., FLEURET F.: Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning* (2018), PMLR, pp. 2525–2534. 1, 2, 3, 5

[KGG20] KUMAR N., GUPTA R., GUPTA S.: Whole slide imaging (WSI) in pathology: Current perspectives and future directions. *Journal of Digital Imaging 33*, 5 (May 2020), 1034–1040. 1

[LH15] LOSHCHILOV I., HUTTER F.: Online batch selection for faster training of neural networks. *arXiv preprint arXiv:1511.06343* (2015). 2

[LWM20] LIU R., WU T., MOZAFARI B.: Adam with bandit sampling for deep learning. In *NeurIPS* (2020), vol. 33, pp. 5393–5404. 2

[RXC*21] REN P., XIAO Y., CHANG X., HUANG P.-Y., LI Z., GUPTA B. B., CHEN X., WANG X.: A survey of deep active learning. *ACM computing surveys (CSUR) 54*, 9 (2021), 1–40. 2

[Set09] SETTLES B.: Active learning literature survey. 2

[TSC*18] TONEVA M., SORDONI A., COMBES R. T. D., TRISCHLER A., BENGIO Y., GORDON G. J.: An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159* (2018). 1

[VLW*18] VEELING B. S., LINMANS J., WINKENS J., COHEN T., WELLING M.: Rotation equivariant CNNs for digital pathology. In *MICCAI 2018* (2018), Springer, pp. 210–218. 3