A Robust Sampling for Optimization in Deep Learning

Aurora Cobo Aguilera [1]

Abstract

We are studying a method based on a robust risk minimization to obtain better performance in stochastic optimization. Traditional methods are implemented as an uniform sampling which does not take into account the underlying characteristics of the data. The novel regularized estimation is focus on an alternative measure by minimizing the variance, achieving an optimal and computationally efficient solution. From this motivation, we try to show the improvements by the transformation of this point estimation into another pre-processing step in the configuration of a neural network in order to select the samples of the mini-batch. Through this step, we prove the reduction of the computational cost and the increase of the accuracy.

1. Introduction

One of the most important strategies in machine learning since its beginnings is the Empirical Risk Minimization (ERM) due to its ability to approximate some function over the data using a limited number of observations (Vapnik, 2013). With this mechanism, if we have a set of samples and a family of functions to be adjusted, we can find the parameters that minimize the empirical risk as defined in equation 1.

$$R_{emp}(\theta) = \frac{1}{n}\sum_{i=1}^{n} Q\left(x_i, \theta\right) \tag{1}$$

However, every time a statistical model is used to approximate some real phenomenon, it is necessary to make a set of assumptions that may end in conflict with the problem indeed. In machine learning, with the attempt to model unknown structures, it is a key issue to take a robust approach, capable of generalize well and fit new observations.

[1]Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Madrid, Spain. Correspondence to: Aurora Cobo Aguilera <acobo@tsc.uc3m.es>.

The regularized ERM face this issue and improve situations where the optimal solution was overfitting the training set.

In fact, the best way to be in agreement with the practical problem considered is to use the information that it provides. Therefore, at the time of choosing the constraints, it will be useful not only to take robust approximations but also study the hidden information of the samples.

With this line of research, several methods have been studied to make robust modeling, such as the work of (Wang et al., 2016). They took a probabilistic approach in order to make inference by raising the likelihood of each data point to a weight. This idea underweighted the observations that disagree with the assumptions, achieving a model less sensitive to corrupted data.

In our paper it is studied the opposite idea. That is, to reduce the variance of the estimator by giving more 'weight' to the samples that get more error. Considering the situations with relevant quantity of outliers, it will be also proposed a probabilistic approach to avoid them to undermine performance.

In the literature, different authors have studied several variance reducers that will serve as a reference. The closer case is the proposal of (Borsos et al., 2018), who developed a non-uniform importance sampling technique to solve an online optimization problem with bandit feedback. Other alternatives are (Namkoong et al., 2017) and (Salehi et al., 2017), who devise using prior knowledge on the gradients of each observation.

In the same way, we are applying our method inspired by a recently proposed robust regularization to the optimization problem in Deep Learning. However, due to its unfeasibility for the training of a complex model with millions of parameters in a large database, we propose a scheme with the same basics translated in a mini-batch selection problem.

2. Variance-based robust regularization

(Namkoong & Duchi, 2016) proposed an alternative to risk minimization and stochastic optimization that provides an optimal and computationally efficient solution. Particularly, it is based on a robust regularization of the empirical risk by adding a variance term.

They are motivated by the bias-variance tradeoff in statistical

learning in order to minimize a quantity trading between approximation and estimation error. Moreover, they provide a tractable convex formulation -whenever the loss function is convex- that approximates closely to the penalized risk.

We can consider the alternative in the study as a **min-max problem**, that is, an optimization with two steps.

- First, **the minimization of the traditional risk**, $\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} p_i q_i(\theta, x_i)$, where θ are the parameters to be computed, x_i a set of n samples and p_i is the weight associated to each sample, so the samples with higher contribution to the loss function are the more valuable in the model.
- Second, **the maximization of the robust objective**, $\max_{p} \sum_{i=1}^{n} p_i q_i$.

As a constraint, they propose the equation 2, where ρ is a parameter to select the confidence level.

$$p \in \mathcal{P}_n = \left\{ p \in \mathbb{R}_+^n : \frac{1}{2} \|np - \mathbf{1}\|_2^2 \leq \rho, \langle \mathbf{1}, p \rangle = 1 \right\} \tag{2}$$

They give a number of theoretical guarantees and empirical evidences in order to show the optimal performance of the estimator with faster rates of converge and the improvement of out-of-sample test performance.

3. Application on Deep Learning

Nowadays, Deep Learning is known as a powerful framework for supervised learning (Goodfellow et al., 2016). It allows the implementation of Neural Networks with as many layers and units as it is desired, providing a more or less sophisticated function to fit a specific dataset. The description of such algorithms is followed by the specification of a cost function, an optimization procedure and a model, what makes the robust objective proposed a direct application in the step of the risk minimization of this kind of tools.

Moreover, neural networks sometimes require long training times when the graph architecture is some how complex. As a consequence, a small improvement at each iteration in the optimization could make huge differences in the performance at the end.

3.1. Classification based on CNNs

In a classification problem, the robust approach proposed by (Namkoong & Duchi, 2016) can be applied as an alternative to improve the performance on unusual classes, where the traditional empirical risk minimization would sacrifice the accuracy to focus on the common classes. For this purpose, **Convolutional Networks** (LeCun & Bengio, 1995) appear to be a promising alternative. More specifically, in the image classification task, they are one of the most popularly chosen models, with competitive results in international problems proposed to the scientific community as the **ImageNet Large-Scale Visual Recognition Challenge** (ILSVRC) (Russakovsky et al., 2015). In addition to this huge dataset, the ImageNet (Deng et al., 2009) with 1000 categories, the community has trained other simpler ones as the MNIST (LeCun, 1998), the SVHN (Netzer et al., 2011), the LSUN (Yu et al., 2015) or the CIFAR10 (Krizhevsky & Hinton, 2009) in order to explore CNN's performance.

Because of this reason, we propose the application to some models based on a state-of-the-art Convolutional Neural Network (CNN) called VGG (Simonyan & Zisserman, 2015). This CNN is based on a study of increasing the depth of the network using an architecture with very small convolutional filters. They show a significant improvement in performance when the number of layers increase from 11 to 19 and prove their results in different datasets as the one of the ILSVRC.

The main idea of the work is to take advantage of this idea as a regularization reflected in an intelligent way of selecting the samples of the mini-batch in this CNN. This approach has been taken to avoid dealing with the intractable min-max problem in such a complex scenario by giving different weights to the samples in the empirical risk.

3.2. Variance reducer regularization

When the number of samples, n, is very large, the traditional optimization recurs to the empirical risk minimization by solvers which divide the dataset in several subsets and perform one iteration per each of them. As an iteration is considered the fact of computing the gradients and uploading the parameters of the model.

For this purpose, traditional methods sample a few points uniformly from the entire dataset in order to satisfy an unbiased estimation of the loss function, what provokes a high variance solution. However, the methodology proposed in this work is based on a non-uniform sampling technique, allowing a balance between the bias and variance and as consequence a more robust solution.

Through the variance reduction it is possible to lead the convergence of the problem to a more generalized solution, better to explain out-layers or avoid overfitting. That is why it is called a regularization mechanism.

In image classification by CNN, the non-uniform sampling is translated in a non-uniform selection of the samples in the mini-batches. More precisely, the resampling procedure benefits the worst scored samples in the loss function, increasing the probability of being duplicated in an iteration.

3.3. Objectives

The purpose of this work is the study of the robust approach for sampling in the stochastic optimization of CNN in order to achieve one or more of the following characteristics.

- A better performance in those samples that are considered out-layers.
- An improvement in the performance of the overall model.
- A faster convergence of the model to an optimal solution.

The main contribution of this paper is to supply the numerical evidence that the proposal performs well on real data and show that its use is competitive in state-of-the-art problems and datasets.

4. Methodology

The way of selecting the samples of the mini-batch at each iteration in the training of the network has been studied with different approaches:

1. Repeat a percentage of the worst performed samples from one mini-batch to the next one.
2. Repeat a percentage of the worst performed samples from one epoch to the next one.
3. Resample half of the data points from the worst performed ones each mini-batch or epoch.

In all cases, the repeated samples are those ones that get the worst score, that is, the highest value in the loss function at the previous iteration. However, while the two first ideas repeat all the worst classified samples of the previous mini-batch or epoch respectively, the last proposal takes a probabilistic approach by resampling from the fixed subset according to the previous cases. In other words, if in the first method it is repeated the 25% of the samples sorted by highest value in the loss function, with the probabilistic approach, this repeated subset is resampled half its size.

With the probabilistic approach it is followed to cover the cases where there are corrupted samples in the dataset which could degrade the quality of the system. It is a way of avoiding them to contribute at all the iterations.

In order to define the methods in the figures of the experiments, we are using the descriptions **Variance Reducer per Mini-batch** (VR-M) and **Variance Reducer per Epoch** (VR-E) for the models number 1 and 2 respectively, and **Probabilistic Variance Reducer per Mini-batch** (PVR-M) or **per Epoch** (PVR-E) for the last case.

5. Experiments

5.1. Model

5.1.1. Dataset

Between all the available datasets from the literature, it has been chosen the MNIST, composed of 60000 training samples and 10000 test samples of handwritten digits as shown in the figure 1. The images are of size 28x28 pixels and in gray scale, what allows an easier and faster training of the experiments in comparison with larger bases as the ImageNet. For this reason, the extension of this work in more complex scenarios will be a future task.

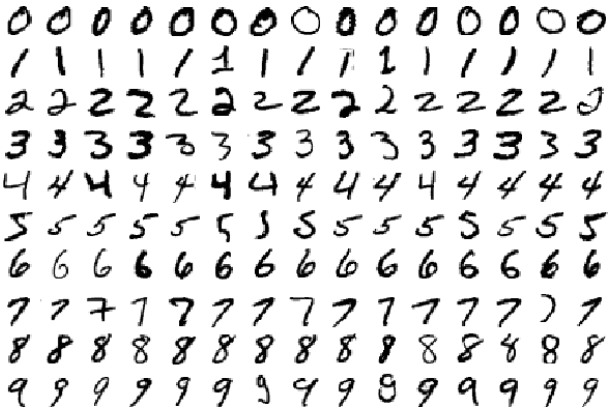


Figure 1. Examples of images from the MNIST dataset

5.1.2. Architecture

As it was mentioned in the section 3.1, we are proving the behavior of our method in a CNN based on the VGG implemented by (Simonyan & Zisserman, 2015). The motivation of this choice is the validation of the method in a complex model where the improvements are considerably more cost efficient. The original architecture has been modified according to the size of our data, resulting in a neural network of 11 layers. It has 3 levels, the first one with two convolutional layers of output 16, the second with other two of output 32 and the third with 4 layers of output 64. All levels are ended with a pool and finally it is applied three fully connected layers of size 1024, except the last one, with size the number of classes. The architecture is shown in the figure 2. All the experiments have been trained with *tensorflow*.

5.1.3. Configuration

The networks' parameters, as the rectification, pooling, etc, has been maintained. Except for the dropout, that was set to 1.0 in some experiments to evaluate the effect of removing all kind of regularization but ours.

For the initialization, the bias was set to zero and the weights

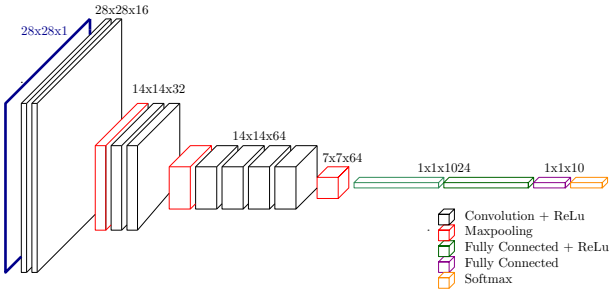
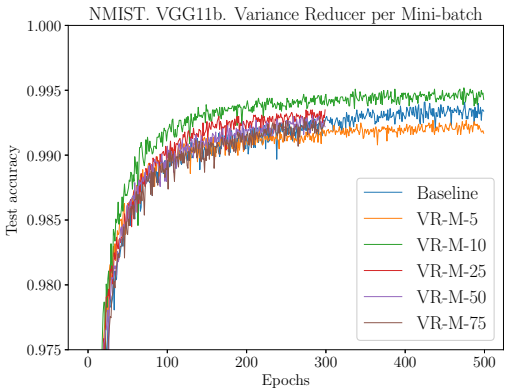


Figure 2. Architecture of the CNN based on the VGGNet

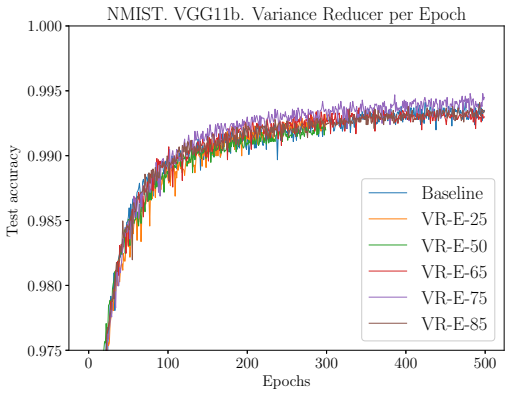
followed a normal distribution with zero mean and standard deviation equal to 0.1.

5.2. Results

In the figures 3 and 4 it is shown the test accuracy obtained for different configurations in the percentage of samples that are repeated for each method. In both cases, the performance of the original scenario (baseline), without the robust implementation, is overcomed.



(a)



(b)

Figure 3. Test accuracy for different percentages of repeated samples in the non-probabilistic methods.

While the VR-M behaves better with smaller percentage of repetition, the VR-E improves with larger percentage until they saturate as it can be appreciated with VR-M-5 and VR-E-85 with the worst accuracies. This is because the former makes changes at each iteration and the other one waits until a complete pass over the dataset. Moreover, one of the advantages of the VR-M is the faster convergence as it can be highlighted in the green curve of subfigure (a) in 3 and at first iterations, the results differ more from the baseline case than the other options.

With the figure 4, we show that the probabilistic method can improve the baseline results too, and it is the best case to deal with out-layers that could degrade the performance of our system.

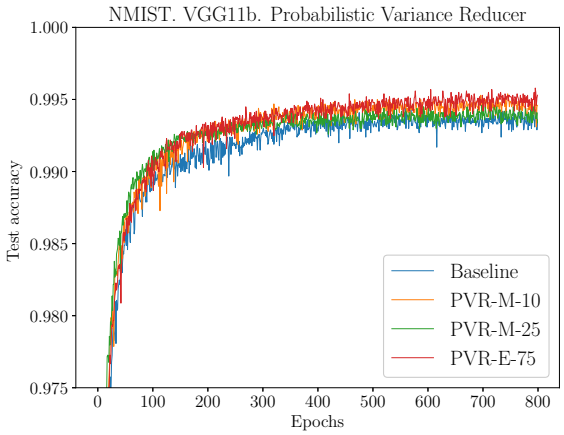


Figure 4. Test accuracy for different percentages of repeated samples in the probabilistic methods.

Lastly, in figure 5 we compare the best configurations with and without dropout. As both our method and the dropout are ways of regularize, we should isolate them to appreciate the differences. It is shown that without dropout, the VR-M-10 is the only configuration valid between the chosen best ones with dropout, but it stands out from the others. However, the best results are the ones with both regularizations integrated.

With the purpose of understanding better what we are doing, the figure 6 represents a histogram with the distribution of the number of times that an image is sampled in the complete training. First, the baseline method performs an uniform sampling so it is defined as an unique bar with all the samples similar number of contributions. Second, the VR-E distributes around the baseline one. And last, the VR-M, with the more important differences, resamples some set of samples much more times than the rest. Those samples are the ones that are worse classified in the previous mini-batch. As it was expected, with the probabilistic approach,

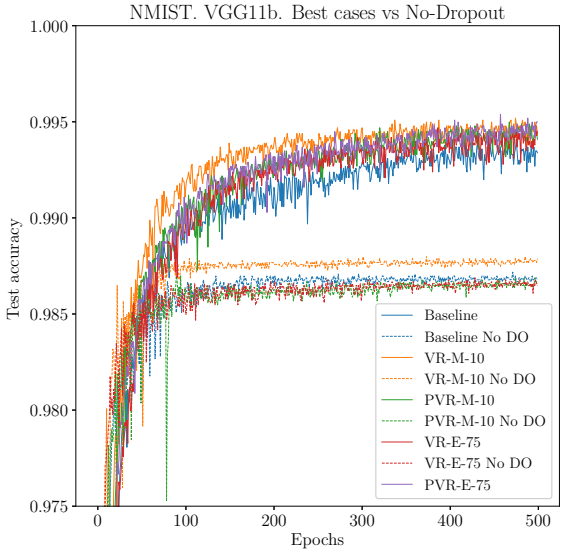


Figure 5. Test accuracy for the best configurations and the comparison without dropout.

this quantity still appears but in less proportion.

6. Conclusions

In this paper we have studied the performance of a novel robust estimation based on a reduction of the variance. First, we have described the original work and then we have proved the theoretical evidences through experiments in deep learning with the training of CNNs.

The table 1 resume the results. Although it has been proved that all the methods proposed are able to overcome the baseline, the characteristics that they offer are different. On the one hand, the VR-M achieves a faster convergence with remarkable improvements from the beginning of the optimization. On the other hand, the VR-E obtains worse accuracy in the earlier stage but at the end get the best score. In any case, the baseline results are always improved.

Table 1. Test accuracy at different epochs with the best configurations.

EPOCH	10	100	300	500
BASELINE	95,49%	98,91%	99,26%	99,33%
VR-M-10	**96,03%**	**99,10%**	**99,44%**	99,43%
PVR-M-10	95,10%	99,06%	99,37%	99,41%
VR-E-75	95,56%	98,93%	99,35%	99,45%
PVR-E-75	95,43%	98,96%	99,36%	**99,50%**

The results presented in this work are a preliminary study,

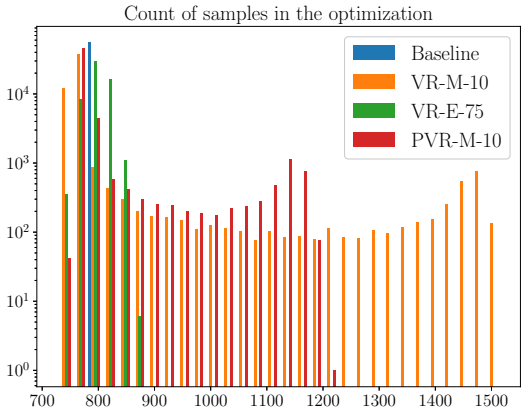


Figure 6. Histogram of sampling for the best configurations and without dropout.

but it is necessary a deeper knowledge of how powerful the proposed tool can be. Currently, we are considering other scenarios with larger datasets and more complex networks.

References

Borsos, Z., Krause, A., and Levy, K. Y. Online variance reduction for stochastic optimization. *arXiv preprint arXiv:1802.04715*, 2018.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255. Ieee, 2009.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT press, 2016.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

LeCun, Y. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

LeCun, Y. and Bengio, Y. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10), 1995.

Namkoong, H. and Duchi, J. C. Variance-based regularization with convex objectives. arXiv:1610.02581, 2016.

Namkoong, H., Sinha, A., Yadlowsky, S., and Duchi, J. C. Adaptive sampling probabilities for non-smooth optimization. In *International Conference on Machine Learning*, pp. 2574–2583, 2017.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 5, 2011.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Salehi, F., Celis, E., and Thiran, P. Stochastic optimization with bandit sampling. *arXiv preprint arXiv:1708.02544*, 2017.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *In Proc. ICLR*, 2015.

Vapnik, V. *The nature of statistical learning theory*. Springer science & business media, 2013.

Wang, Y., Kucukelbir, A., and Blei, D. M. Robust probabilistic modeling with bayesian data reweighting. *arXiv preprint arXiv:1606.03860*, 2016.

Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., and Xiao, J. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.