Variance Suppression: Balanced Training Process in Deep Learning

T Yi¹ and X Wang¹*

¹ Department of Electronic Engineering, Fudan University, Shanghai 200433, China
* E-mail: wxx07@fudan.edu.cn

Abstract. Stochastic gradient descent updates parameters with summation gradient computed from a random data batch. This summation will lead to unbalanced training process if the data we obtained is unbalanced. To address this issue, this paper takes the error variance and error mean both into consideration. The adaptively adjusting approach of two terms trading off is also given in our algorithm. Due to this algorithm can suppress error variance, we named it Variance Suppression Gradient Descent (VSSGD). Experimental results have demonstrated that VSSGD can accelerate the training process, effectively prevent overfitting, improve the networks learning capacity from small samples.

1. Introduction

Gradient descent is one of the most widely used optimization algorithm in deep learning. Stochastic gradient descent (SGD) has been proved to be an effective approach for deep neuron networks training [1]. For each learning sample \( \mathbf{x}_i \) and label \( d_i \), the network’s parameters \( \theta \) and output \( y_i \), we can accordingly write the error as \( e_i(\theta; \mathbf{x}_i, d_i) \). In general, we apply SGD algorithm on networks training by optimizing the average error \( \text{loss function} \)

\[
E = \frac{1}{n} \sum_{i=1}^{n} e_i(\theta; \mathbf{x}_i, d_i)
\]

where \( n \) is a batch size. In SGD training steps, a random batch be drawn from data set and fed in the networks to compute the loss function’s gradient \( \nabla_{\theta}E \), and updating parameters by gradient descent. Although SGD is effective, we are still faced with two challenging problems: (I) excessive local minimum points, (II) unbalanced training process.

With the expansion of data sets and the deepening of the network structure, nevertheless, optimizing loss function (1) becomes a tough task because the object function has a large number of local extremum points[3]. To address this difficult optimization issue caused by local extremum, in the past few decades, researchers have proposed many gradient descent variants (e.g. Momentum[4], Adagrad[5], Adam[6]). The main idea of these algorithms is that using the first derivatives estimate the first moment or second moment of gradient to select the feasible direction of gradient descent, which is helpful to escape from the local extremum point and accelerate convergence speed. Another helpful method is that adding penalty terms to reduce the feasible direction such as \( L_1, L_2 \) regularization terms[7][8]. In above methods, although it, to a certain extent, solves the problem of easy to fall into local extremum during the training, not fundamentally solves the problem.

In engineering practice (e.g. target detection, image classification), the data we obtained always have unbalanced distribution. The unbalanced distribution of data sets will lead to unbalanced training process[9]. When SGD is used, for the unbalanced data set[10][11], the gradient of loss function derives from the linear sum of the gradient of the single sample loss, which will result in the gradient submergence of the smaller samples in the data set. In addition, even for uniform data sets, the ability of
networks fitting different targets is different. After several training epochs, some targets still have a larger loss value[8][11] that are more difficult to fit. Although the loss value of this part is larger, its value is far less than the sum of the rest loss values of the better fitting samples, which leads to its gradient inundated in the small loss values’ summation, and thus make the training process into a bottleneck.

In this paper, we propose Variance Suppression Stochastic Gradient Descent (VSSGD) to rebalance the training process. On the one hand, VSSGD is helpful to escape from the local extremum in training steps; on the other hand, VSSGD can force training process more balanced and stable. As a new approach to define loss function, VSSGD regards to restrain variance of outputs error signals in a mini-batch as part of learning objectives. It is different from the previous algorithms that variance suppression (VS) can force the networks’ fitting errors equilibrium reduction. By this way, the networks fitting errors have been equilibrated. Moreover, because the trained neuron networks have capacity to fit each sample of the batch simultaneously, we can therefore improve the networks’ generalization ability and accelerate convergence speed of gradient descent.

2. Variance Suppression

2.1. Measure the unbalanced training process by variance

Whether for target detection or image classification, we often encounter the issue of imbalance in the process of training neural networks. Due to the unbalanced training process, the networks we trained will have very different accuracy for different data samples. As shown in figure 1, although each class have the same sample size in CIFAR-10 dataset[12], the test accuracy is very different after dozens of times training.

![Figure 1. CIFAR-10 each class test accuracy](image)

There are three factors that can result in training imbalance. First, the data sets may have a nonuniform distribution. Second, the random mini batch we feed in networks is not always uniform. Third, networks fitting ability to different sample classes is different. In order to measure the unbalanced training process, we put forward the concept of error variance. We define error variance at the following equation

$$\text{var}[e] = \frac{1}{n} \sum_{n=1}^{n} (e_n - \frac{1}{n} \sum_{i=1}^{n} e_i)^2.$$  

As shown in figure 2, we train LeNet on MNIST data set[13] after 50000 times iterations with 64 batch size, and record the network’s fitting error variance for 64 samples after each iteration. We can find that the fluctuation range of the variance is very large, particularly for the point that variance exceeds 10, and it is no downward trend, which demonstrated the training process is extremely unbalanced.

2.2. Add error variance to loss function

In deep learning, the two major problems are classification and regression, and the corresponding two chief approaches to define cost function are distance and cross entropy. By networks’ mapping we get the abstract features in output space, and we can therefore use a measurement to measure the networks fitting capacity, such as cross entropy and Euclidean distance. In a certain measure, networks’ fitting
errors can be reduced by adjusting its parameters. In SGD, parameters updating is performed for each mini batch by gradient descent. And gradient direction of a batch data’s loss comes from each single sample losses’ gradient direction summation. However, this summation produces a large number of local extremum points. And due to the three factors we indicated in above, the training process is unbalanced. For example, to certain training step, the data that be fed in the networks partial to a certain category, which can make the parameters’ updating direction closed to fit this class. Once the networks overfitting to one class that is equaled to give up to fit the rest classes. To prevent the training process goes to wrong, we consequently take the error variance into consideration, if $\lambda$ is a parameter to adjust variance rate, thus the loss function can be written as

$$L = E + \frac{1}{2} \lambda \text{var}[\epsilon].$$

(3)

Consider the contribution of a single sample to the gradient:

$$\frac{\partial L}{\partial e_n} = \frac{1}{n} \left[ 1 + \lambda \left( e_n - \frac{1}{n} \sum_{i=1}^n e_i \right) \right] \frac{\partial e_n}{\partial \theta}.$$

(4)

And we calculate the difference between the following two partial derivatives:

$$\frac{\partial L}{\partial e_n} - \frac{\partial E}{\partial e_n} = \frac{\lambda}{n} \left( e_n - \frac{1}{n} \sum_{i=1}^n e_i \right).$$

(5)

Equation (5) demonstrates that the data with larger fitting error will contribute more to the parameters update, and the data with small fitting error will contribute less to the parameters update. When we use a batch data to train the neural networks, the gradient contribution rate of the high loss samples can be enhanced, and the low loss samples’ gradient contribution can be reduced. Thus, the fitting errors can be reduced symmetrically, the training process is balanced, and the generalization ability of the trained networks will be improved.

Next we analyze VSSGD from another perspective. In order to detailedly describe the problem, we put forward a new concept: gradient space

$$G = \left( \frac{\partial e_1}{\partial \theta}, \frac{\partial e_2}{\partial \theta}, \ldots, \frac{\partial e_n}{\partial \theta} \right).$$

(6)

For SGD and VSSGD, both of the gradients belongs to the same gradient space $G$, $\forall L \in \text{range}(G)$, $\forall E \in \text{range}(G)$, but the coordinates are different. In VSSGD, the difference of gradient contribution of different single sample has been widened. It should be noticed that the weights must be positive, let the coefficient of equation (4) nonnegative we can achieve an inequality:

$$0 < \lambda \leq \frac{1}{\text{min}\{e_n - \frac{1}{n} \sum_{i=1}^n e_i\}_{n=1}}.$$

(7)

To ensure that the denominator is nonzero and the scale adjustment of variance is appropriate, in practice we recommend using

$$\lambda = \frac{\alpha}{\varepsilon - \text{min}\{e_n - \frac{1}{n} \sum_{i=1}^n e_i\}_{n=1}}.$$

(8)

In equation (8), $\alpha$ is a scale regulator and $\alpha \in (0,1)$, $\varepsilon$ is a small positive constant. The value selection of $\alpha$ depends on the degree of datasets imbalance. It should be noted that $\lambda$ is up to each training steps loss value, so we should compute $\lambda$ for each training steps. Another minutia is that $\lambda$ is regarded as a constant when derivations are computed. Therefore, in practice the loss function of VSSGD will be defined as:

$$L(\theta;x,d) = \frac{1}{2n} \sum_{i=1}^n \epsilon_i(\theta;x,d) + \frac{\alpha}{2n\varepsilon - 2n\min\{\epsilon_n,\epsilon_\theta,\epsilon_d\} - \sum_{i=1}^n \epsilon_i(\theta;x,d)]^\alpha - \sum_{i=1}^n \epsilon_i(\theta;x,d)]^\alpha}{\sum_{i=1}^n \epsilon_i(\theta;x,d)}.$$

(9)

A description of the algorithm used in VSSGD technique is given in Table 1. Compared with SGD,
VSSGD does not increase the complexity of the algorithm.

### Table 1. Algorithm description

| Algorithm | VSSGD technique |
|-----------|-----------------|
| **Initialize:** | batch size: $n$ |
| | learning rate: $\eta$ |
| | scale regulator: $\alpha$ |
| | weights and biases: $\theta$ |
| **Input:** | training data: $\{x^{(b)}_{h}^{n}, \{d^{(b)}_{h}\}_{h=1}^{B} \}$ |
| **while** not converge do | |
| | shuffle the data set $\{x^{(b)}_{h}^{n}, \{d^{(b)}_{h}\}_{h=1}^{B} \}$ |
| | for each batch do |
| | compute $e_{n}$, $e_{b_{1}}$, $..., e_{n}$ |
| | $t \leftarrow \min\{e_{n}, \frac{1}{n} \sum_{i=1}^{n} e_{i}^{n}\}$ |
| | $\lambda \leftarrow \frac{\alpha}{e-t}$ |
| | modify loss function $L = \sum_{i} e_{i} + \frac{1}{2} \lambda \text{var}[e]$ |
| | update $\theta$ by optimizer /* Momentum, Adagrad, Adam.et*/ |
| | end for |

3. Experiments and Results

3.1. Experimental conditions

We performed experiments using the MNIST dataset[13] and CIFAR-10 dataset[12], and compared the results by VSSGD technique and that by the standard SGD as a baseline. The MNIST database of handwritten digits has a training set of 60000 examples, and a test set of 10000 examples. Each handwritten image is $28 \times 28$ pixels grayscale image. MNIST images are divided into 10 classes, from 0 to 9. The CIFAR-10 dataset consists of $60000 \times 32 \times 32$ color images in 10 classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck), with 6000 images per class. There are 50000 training images and 10000 test images in CIFAR-10. Compared with MNIST, CIFAR-10 is more complex and more difficult to train. In our experiments, we train LeNet-5 on MNIST dataset, modified VGG16 on CIFAR-10 dataset that the fully-connected layers are revised.

3.2. Experiments on uniform distribution data set

In this section, we use the entire MNIST and CIFAR-10 dataset training networks. Both of the entire dataset have a uniform distribution in train set and test set. Because of networks fitting capacity to different objection is different, uniform data distribution does not mean that the dataset is balanced. Thus, experimenting on uniform distribution data set is essential. To demonstrate our algorithm can suppress variance, we train Le-Net by VSSGD and recorded the error variance after each training step. As shown in figure 3(a), compared with SGD, the error variance of VSSGD are more balanced than SGD and fluctuation range are suppressed more narrow than SGD. In VSSGD, almost all of the variance less than 1. SGD training steps are violent concussion by comparison because a number of points with error variance greater than 10 arising.

We use learning rate 0.01, batch size 32, scale regulator 0.2 to train VGG-Net by VSSGD, and use the same parameters without variance term to train VGG-Net by SGD as a contrast. We obtained the loss variation curve and the test precision curve. As shown in figure 3(b) the training processes of VSSGD are more stable and VSSGD is helpful to accelerate the convergence speed. The experimental results show that VSSGD loss value steadily declines in training process. In contrast, SGD training process is unbalanced that the loss value may fiercely up and down. figure 3(c) demonstrates that VSSGD is earlier
convergence than SGD, and VSSGD improves the networks’ generalization ability.

(a) The error variance in LeNet training process on MNIST
(b) Transition in Mean cross entropy loss
(c) Transition in test accuracy

**Figure 3.** Results of experiments using VSSGD

### 3.3 Experiments on nonuniform distribution data set

In this section, we use the nonuniform distribution dataset training networks. To simulate the extreme disequilibrium of the datasets, the samples of each class in train sets are partly deleted in different degrees, and the test sets remain unchanged. It is worth noting that class 1 in table 1 D1 and table 2 D2 is only reserved for 50 and 10 images, but the total trainset has more than 20000 images.

To show the effect of data distribution, we apply a separated test for different categories. The table 1, table 2 give the performance of our algorithm on the test set of MNIST dataset, and compare it to the SGD. As table 1, table 2 shown, VSSGD have better performance than SGD. Even in table 2 that the class 1 only 10 images, VSSGD still have 62.2% accuracy but SGD can hardly detect this class images. This experiment is similar to training deep object detection models.

#### Table 2. MNIST dataset test accuracy.

| Class | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  |
|-------|----|----|----|----|----|----|----|----|----|----|
| distribution1 (D1) | 300 | 50 | 2000 | 2500 | 3000 | 800 | 4000 | 4500 | 3000 | 4000 |
| VSSGD | 0.950 | 0.882 | 0.985 | 0.986 | 0.993 | 0.958 | 0.992 | 0.980 | 0.982 | 0.981 |
| SGD   | 0.963 | 0.829 | 0.980 | 0.988 | 0.989 | 0.974 | 0.994 | 0.989 | 0.985 | 0.976 |

#### Table 3. MNIST dataset test accuracy.

| Class | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  |
|-------|----|----|----|----|----|----|----|----|----|----|
| distribution2 (D2) | 300 | 10 | 2000 | 2500 | 3000 | 800 | 4000 | 4500 | 3000 | 4000 |
| VSSGD | 0.957 | 0.622 | 0.994 | 0.992 | 0.992 | 0.987 | 0.985 | 0.977 | 0.985 | 0.987 |
| SGD   | 0.966 | 0.011 | 0.987 | 0.986 | 0.990 | 0.971 | 0.994 | 0.991 | 0.989 | 0.985 |

**Figure 4.** Results of experiments using MNIST dataset

Figure 4 shows the performance transition curves in training process. Due to the scale factor we used in this experiment (D1: $\alpha=0.8$, D2: $\alpha=0.95$) is larger than uniform distribution data sets, the VSSGD training loss descending process slightly slow down than SGD. Although the loss descending
process of VSSGD is slow, the test accuracy rises rapidly, and the training processes of VSSGD are more stable. This phenomenon demonstrates that VSSGD have more strong learning capacity than SGD. In the training process, VSSGD can balance the gradient contribution of each class and take small class samples into account.

The table 3, table 4 give the performance of our algorithm on the test set of CIFAR-10 dataset. For this complex datasets and extreme disequilibrium training condition, our algorithm still show an obvious advantage than SGD.

| Class   | airplane | automobile | bird | cat | deer | dog | frog | horse | ship | truck |
|---------|----------|------------|------|-----|------|-----|------|-------|------|-------|
| distribution3(D3) | 5000     | 1000       | 3000 | 4000| 5000 | 2000| 1000 | 3000  | 2000 | 5000  |
| VSSGD   | 0.905    | 0.838      | 0.724| 0.733| 0.862| 0.678| 0.750| 0.834 | 0.871| 0.930 |
| SGD     | 0.902    | 0.788      | 0.714| 0.722| 0.869| 0.675| 0.708| 0.859 | 0.870| 0.936 |

Table 3. CIF-10 dataset test accuracy.

| Class   | airplane | automobile | bird | cat | deer | dog | frog | horse | ship | truck |
|---------|----------|------------|------|-----|------|-----|------|-------|------|-------|
| distribution4 (D4) | 5000     | 100        | 4000 | 5000| 5000 | 5000| 5000 | 5000  | 1000 | 5000  |
| VSSGD   | 0.900    | 0.422      | 0.778| 0.722| 0.838| 0.785| 0.869| 0.856 | 0.814| 0.933 |
| SGD     | 0.892    | 0.330      | 0.761| 0.708| 0.849| 0.768| 0.886| 0.819 | 0.808| 0.942 |

We also use our algorithm to solve binary classification problems, and to compare it with SGD. Selected one class as a positive sample and the rest classes as negative sample, thus the dataset is built for binary classification problems. As table 5 shown, we perform experiments on different proportion of positive and negative samples for different algorithms. Compared with SGD, VSSGD exhibits a more obvious advantage.

| Positive: Negative | 1:10 | 1:50 | 1:80 | 1:150 | 1:300 |
|--------------------|------|------|------|-------|-------|
| VSSGD              | 0.921| 0.776| 0.645| 0.399  | 0.310 |
| SGD                | 0.878| 0.645| 0.501| 0.352  | 0     |

Table 5. binary classification.

4. Discussion and Conclusion

In deep learning, especially metric learning, the definition of error measurement methods is particularly important, because the trained networks’ generalization ability and the training difficulty is up to measurement we defined. As a new approach to define loss function, VSSGD suppresses the error variance in networks training process by adding error variance into loss function and dynamically adjusting the different samples’ gradient contribution.

Our results have indicated that VSSGD can accelerate the training process, effectively prevent overfitting, improve the networks learning capacity from small samples. This learning mechanism shows different effects in different conditions. Due to variance suppression, the training process can be rebalanced, the gradient utilization rate can be improved, which is helpful to escape from local minimum. VSSGD have more strong learning capacity than SGD.

Theoretically, the VSSGD error measurement equal to use weighted error measurement, which is different from the conventional SGD that uses the average errors. This measurement as an effective solution to the issue that the training process is unbalanced caused by the average errors measurement of SGD shows more strong learning capacity than SGD for data distribution imbalanced condition.

In VSSGD, the selecting of scale factor $\alpha$ is a key operation, which can affect the training speed and networks generalization ability. However, this algorithm chief limitation is that scale factor is selected by experience and VSSGD have a narrow advantage on uniform distribution. For example, the LeNet that we use the whole MNIST trainset trained does not show the batter performance than SGD. We guess the reason is that the networks’ representative ability is limited by its architecture. In the next, we will study on the adaptive adjusting of errors weights. We hope this algorithm will apply for more widely machine learning field, such as least square, SVM.
References

[1] Bottou L 2010 Large-Scale Machine Learning with Stochastic Gradient Descent *Proceedings of COMPSTAT’2010* ed Lechevallier Y and Saporta G (Heidelberg: Physica-Verlag HD) pp 177-186

[2] Polyak B T and Juditsky A B 2006 Acceleration of stochastic approximation by averaging *Siam Journal on Control & Optimization* vol 30 (USA: Society for Industrial and Applied Mathematics) pp 838-855

[3] Werbos and Paul J 1994 Supervised learning: can it escape its local minimum? *Theoretical Advances in Neural Computation and Learning* ed Siu KY and Orlitsky A (Boston: Springer) pp 449-461

[4] Qian N 1999 On the momentum term in gradient descent learning algorithms *Neural Networks* vol 12 (Elsevier) pp 145-151

[5] Duchi J, Hazan E, Singer A, Duchi J and Hazan E 2011 Adaptive subgradient methods for online learning and stochastic optimization *Journal of Machine Learning Research* vol 12 ed B Francis and B David et al (USA: Microtome Publishing) p 2121

[6] Kingma D P and Ba J 2014 Adam: a method for stochastic optimization *International Conference on Learning Representations* (ICLR 2015) (Preprint arXiv:1412.6980)

[7] Krogh A, Hertz J A, Hanson J S and Moody J E 1992 A simple weight decay can improve generalization *NIPS’91 Proceedings of the 4th International Conference on Neural Information Processing Systems* ed Moody J E and Hanson S J et al (San Francisco: Morgan Kaufmann Publishers) pp950-957

[8] Tibshirani R 1994 Regression shrinkage and selection via the lasso: a retrospective *Journal of the Royal Statistical Society Series B* vol73 (USA: Wiley for the Royal Statistical Society) pp 273-282

[9] D Masko and P Hensman 2015 The impact of imbalanced training data for convolutional neural networks *degree project, in computer science, first level Stockholm, Sweden.*

[10] Shrivastava A, Gupta A and Girshick R 2016 Training region-based object detectors with online hard example mining *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR) pp 761-769

[11] Lin T Y, Goyal P, Girshick R, He K and Dollar P 2017 Focal loss for dense object detection *IEEE Transactions on Pattern Analysis & Machine Intelligence* (ICCV vol 99) ed S Dickinson (IEE) pp 2999-3007

[12] Krizhevsky A and Hinton G 2009 Learning multiple layers of features from tiny images *Technical report*, University of Toronto

[13] Lecun Y L, Bottou L, Bengio Y and Haffner P 1998 Gradient-based learning applied to document recognition *Proceedings of the IEEE* vol 86 (IEEE) pp 2278-2324