EVEN LARGER BATCH SIZE WITH ADDITIVE GRADIENT NOISE

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning rate scaling rules have shown to be effective in allowing stochastic gradient descent with large batch sizes to attain good generalization performance. However there is still a maximum batch size with which these scaling rules is effective. We add mean-zero noise to the gradient, and demonstrate that good generalization performance can be achieved with even larger batch sizes than previously thought. Adding artificial noise to the gradient allows us to tune the signal-to-noise ratio of the stochastic gradients more precisely than with the learning rate. We demonstrate on several benchmark data sets that adding noise to the gradient significantly improves the performance on test for comparable to the training set size. Our results indicate that there is still room for training speedup through large batch parallel implementations.We take the large batch training to its logical extreme by performing full-batch gradient descent with added noise, effectively simulating stochastic gradient descent (SGD). We are still able to achieve good performance on test with this simulated SGD.

1 INTRODUCTION

Stochastic gradient descent (SGD) is the dominant optimization algorithm in deep learning. It was originally developed in a serial setting, however there has been much recent interest in adapting SGD to a parallel setting due to increasing data set sizes or a desire for faster training times. One way to parallelize SGD is by sharing the computation of the mini-batch gradient across many workers. To effectively utilize the parallel resources, the batch size must be significantly increased. This has been dubbed Large Batch SGD in the literature, where the batch size is significantly larger than what would be normally used.

However an increase in batch size has been shown to lead to a decrease in test accuracy (Keskar et al 2016, Goyal et al 2017). This behavior can be understood by interpreting SGD as a stochastic differential equation (Smith & Le 2017). There is a noise scale $g = \epsilon(\frac{N}{B} - 1)$ where ϵ is the learning rate, N the training set size, and B the batch size. There is an optimal noise scale g which maximizes the test accuracy (with a constant learning rate). When $B \ll N$ we can approximate $g \approx \epsilon N/B$. This insight has motivated learning rate scaling rules, the main tool which have allowed Large Batch SGD to attain good performance on test (Smith et al 2018, Goyal et al 2017, Hoffer et al 2017, You et al 2017). Increasing the batch size decreases the noise scale g. Thus to maintain the optimal noise scale the learning rate is proportionally increased. There are several proposed scaling rules: linear, square root, log.

While the learning rate scaling has had much success, there is still a limit to how much the batch size can be increased. For example, it is common to use a batch size $B = 64$ while training MNIST which has training set size $N = 60k$. Let ϵ be the tuned learning rate. If we calculate the noise scale, $g_{64} = 937\epsilon$. But if we want to increase the batch size to $B = 30k$, using the same ϵ the noise scale becomes $g_{30k} = \epsilon$. If we want to maintain the noise scale, we would need to increasing the learning rate by almost $1,000$ times! From this simple example it is clear that simple scaling the learning rate will not work for significantly large batch sizes. One could ask: is it even possible to attain good generalization performance when using batch size approximately half of the training set?

Adding artificial noise to the stochastic gradients has long history in SGD. Stochastic gradient langevin dynamics have been used for bayesian learning (Welling & Teh 2011). The noisy SGD algorithm uses uniform noise to escape saddle points (Ge et al 2015). Added gradient noise has been shown to improve the training for very deep networks (Neelakantan et al 2016). We add artificial noise to the stochastic gradients to improve deep learning training on large batch sizes. Let g_t be the stochastic gradient at time step t. We assume that we can decompose $g_t = f_t + e_t$ into the true gradient f_t and the stochastic noise e_t. By scaling the learning rate $\epsilon g_t = \epsilon(f_t + e_t)$, both the true gradient and stochastic noise are increased. With high learning rate ϵ this can cause divergence issues the true gradient will be scaled too greatly. Let $\sigma_t \sim N(0, 1)$ be a source of artificial gaussian noise. If we instead add scaled artificial noise

$$g_t = f_t + e_t + \alpha_t \sigma_t$$

we can control the noise scale *independently* of the scaling on the true gradient. With the artificial noise we can also encourage Here we show,

- Adding mean-zero noise to the gradient significantly increases the generalization performance for Large Batch SGD. We conduct experiments on MNIST and CIFAR10, and show an increase of up to 10% in test accuracy and batch size up to half of the training set size.
- Adding mean-zero noise to the gradient significantly stabilizes the training of Large Batch SGD. The test set accuracy oscillates significantly at large batch sizes for high enough values of the learning rate. We show that by adding gradient noise removes essentially all oscillation in the training curves.

To the best of our knowledge, this paper is the first which combines artificial gradient noise and Large Batch SGD.

2 METHOD

We adapt the additive noise procedure from Neelakantan et al (2016). We add standard normal artificial noise $\sigma_t \sim N(0, 1)$ that is scaled by a time-dependent constant,

$$g_t \leftarrow g_t + \alpha_t \sigma_t. \tag{1}$$

In our experiments using a decaying constant worked better than using a fixed Gaussian noise. The decay schedule of α_t is inspired from Welling & Teh (2011)

$$\alpha_t = \frac{\eta}{(1 + t)^\gamma} \tag{2}$$

with $\eta \in \{0.005, 0.01, 0.3\}$ and $\gamma = 0.55$. Higher noise early on encourages exploration of the loss landscape.

3 INDEPENDENT CONTROL OF TRUE GRADIENT AND STOCHASTIC NOISE

In its most simple form the update equation for SGD is

$$\theta_t \leftarrow \theta_{t-1} - \epsilon_t g_t$$

with ϵ_t the learning rate and $g_t = \nabla \ell(\theta_{t-1}, \xi_t)$ an unbiased estimate of the true gradient. ξ_t is typically the source of randomness due to the randomly sampled mini-batch. We assume that the stochastic gradient can be decomposed

$$g_t = f_t + e_t$$

into the true gradient f_t and stochastic noise e_t. As the batch size increases the variance of g_t decreases. This translates to e_t decreasing. The learning rate scaling rules use a larger learning rate ϵ' to compensate for the decrease in variance. The scaled stochastic gradient becomes $\epsilon' g_t = \epsilon' f_t + \epsilon' e_t$. Both the true gradient and noise are increased because the scaling is used on the stochastic gradient. This can be disadvantageous if the scaling ϵ' is too large. We can see that if ϵ' is too large the true gradient f_t will be scaled too greatly, causing divergence.

The issue is that the learning rate affects both true gradient and stochastic noise. If instead artificial noise is added as in Equation 1, the true gradient and noise level can be controlled *independently*. We can now write the update equation for SGD with artificial noise as

$$\theta_t \leftarrow \theta_{t-1} - \epsilon_t(f_t + e_t) + \alpha_t \sigma_t.$$

Adding artificial noise to the gradient can be seen as controlling the signal to noise ratio (SNR) of the stochastic gradients g_t. The learning rate scaling rules can be seen as re-tuning the SNR to account for the increase in batch size. Increasing the batch size increases the strength of the signal, i.e. the true gradient f_t. However a certain amount of noise in the gradient is essential to find minima which generalize well. The learning rate scaling rules increase the noise strength by increasing the learning rate. But this increases both the signal and noise at the same time. By adding artificial noise to the gradient, the noise can be strengthened without touching the signal.

4 Related work

List batch sizes used by other papers (mostly on CIFAR10). The current literature on Large Batch SGD utilizes learning rate scaling rules to attain good generalization performance on larger batch sizes. Smith et al (2018) show that decaying the learning rate and increasing the batch size are equivalent. On their experiments on CIFAR10, the use a maximum batch size of 5120. Goyal et al (2017) use a linear scaling rule and warmup scheme to attain good performance for mini-batch size 8192 on the ImageNet data set. You et al (2017) employ a layer-wise adaptive update to train Resnet-50 on ImageNet with up to batch size 32K.

5 Experiments

We conduct experiments with Large Batch SGD on MNIST digit classification and CIFAR10 object recognition. We show that adding artificial gradient noise has two benefits. First, we can improve performance on the test set by as much as 10%. Second, we can effectively stabilize the training curves. When the batch size is very large, we observe that the performance on the test set swings widely over successive iterations, by as much as 40%. The addition of artificial noise to the gradients eliminates all of this oscillation.

5.1 MNIST

We train a standard convolutional neural network with ReLU activation function (Nair & Hinton 2010) on the MNIST handwritten digit classification dataset (LeCun et al 1998). MNIST has training set with $60,000$ examples and test set with $10,000$ examples. The network has two convolutional layers followed by two fully connect layers. This network can be found in the pytorch documentation `https://github.com/pytorch/examples/tree/master/mnist`.

We add gradient noise sampled from a standard normal distribution which is scaled by the schedule in Equation 2 with $\eta = \{0.005, 0.01, 0.3\}$. We use SGD with momentum parameter set to 0.5, and learning rates in a range from 0.01 to 0.60. We run the training for 15 epochs and record the final test accuracy. At lower batch size we find it better to use smaller values of η. As the batch size increases, we find it to use larger values of η.

Batch Size	Final Test Accuracy (%)	Artificial Noise Final (%)
2,048	98.61	98.59
4,096	96.45	
8,192	95.66	96.53
16,384	94.37	95.72
32,768	86.08	92.51
45,000	86.48	92.19

Table 1: MNIST with Artificial Gradient Noise

The results can be seen in Table 5.1. We conduct experiments up to batch size of 45k, which is three quarters of the training set size. These are truly large batch sizes. The additional gradient noise only really helps when the batch size is extremely large. For MNIST, this occurs when the batch size is about half that of the training set. Here we are able to achieve performance on test of above 90% for up to batch size of 45k, which is three quarters of the training set!

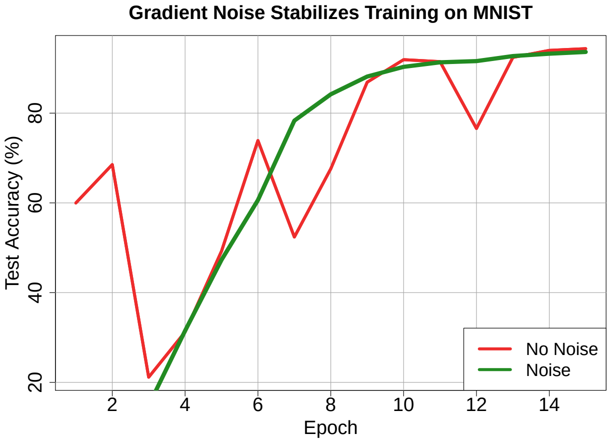

Figure 1: Test accuracy for MNIST, with and without artificial gradient noise.

The training can be quite unstable for Large Batch SGD. In Figure 1 we plot the training accuracy per epoch for training runs with and without artificial gradient noise. We set the learning rate $\epsilon = 0.1$, batch size $B = 16,384$, and noise decay $\eta = 0.3$ if used. The test accuracy oscillates wildly when no artificial noise is used, where the drop is 50%. The test accuracy continues to swing significantly until the accuracy settles in the low 90s. When artificial noise is added to the gradient, the test accuracy smoothes out completely.

5.2 CIFAR10

We train the ResNet18 model (He et al 2015) on the CIFAR10 dataset (Krizhevsky 2009). The CIFAR10 dataset consists of 60,000 32x32 color images in 10 classes, with 50,000 training images and 10,000 test images. We add gradient noise sampled from a standard normal distribution which is scaled by the schedule in Equation 2 with $\eta = \{0.005, 0.01, 0.3\}$. We use SGD with momentum parameter set to 0.5, and learning rates in a range from 0.01 to 0.2. In our initial experiments we run the training for 20 epochs and record the final test accuracy.

The results for batch size up to 8,192 can be seen in Table 2. Like in the MNIST experiments, the benefits of artificial noise manifest when the batch size gets sufficiently big. The improvement in final test accuracy that artificial noise provides also increases as the batch size increases.

Batch Size	Final Test Accuracy (%)	Artificial Noise Final (%)
1,024	86.36	84.88
2,048	84.54	84.16
4,096	79.33	82.28
8,192	65.64	71.08

Table 2: CIFAR10 with Artificial Gradient Noise

[Still working on getting more results for CIFAR10]

5.3 WikiText-2

Much of the literature on Large Batch SGD has focused on image recognition tasks using datasets. We perform language modeling experiments on the WikiText-2 dataset using a LSTM model.

[Planning on running these experiments, have not yet]

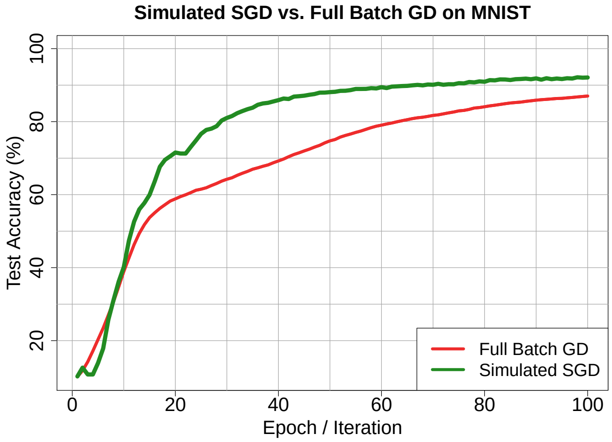

Figure 2: Training full-batch gradient descent with and without artificial noise on MNIST. By adding artificial noise to the full-batch gradient we are simulating SGD.

6 Full batch gradient descent with added noise

We take Large Batch SGD to its logical conclusion by running full-batch gradient descent with added artificial noise. The source of randomness of the stochastic gradient typically comes from the randomly sampled mini-batch. However when the full batch is used to compute the gradient, there is no more randomness with respect to the empirical loss function. This randomness in the gradients is important because it helps SGD to reach minima that generalize well. We re-introduce randomness to the gradient with artificial noise as in Equation 1. This is essentially “simulating” SGD.

It is uncertain if the noise which naturally occurs in the stochastic gradients has the same structure as Gaussian noise. We conduct experiments on the MNIST data set, using the same network described in the previous section. A learning rate of 0.01 is used, with the noise scale schedule as in Equation 2 with $\eta = 1$. Both full batch gradient descent with and without additional noise are run for 100 epochs.

We compare simulated SGD with regular full-batch gradient descent in Figure 2. Adding noise to the true gradient significantly accelerates training over the full-batch gradient descent. In addition the final achieved test accuracy is higher by 5%. We believe that the added noise helps the optimization procedure to more efficiently explore the non-convex loss landscape. Because the full training set is used in each iteration, each epoch consists of exactly one iteration. The simulated SGD procedure is able to converge in 100 iterations.

Because of the success of the simulated SGD procedure, Gaussian noise can be reasonable effective in simulating the natural noise in the stochastic gradients of SGD. If the simulated SGD procedure is run for 500 epochs / iterations and allowed to converge, and the test accuracy is reported in Table 3. This accuracy is very close to the state-of-art results, which are just above 99% depending on the network configuration `http://yann.lecun.com/exdb/mnist/`.

Epochs	Simulated SGD Test Accuracy (%)
500	97.13

Table 3: Test Accuracy for Converged Simulated SGD

It remains to be seen if state-of-the-art results can be achieved with simulated SGD by using different types of noise, potentially conditioned on the iterates (Zhu et al 2019). Doing so would provide a convincing argument that the noise used to achieve such results would have a functional equivalent to the noise introduced through randomly sampled mini-batches.

7 Conclusion

[Tentative]

We show that adding artificial noise to the gradient can improve the generalization performance of Large Batch SGD. Additive noise can tune the signal-to-noise ratio of the stochastic gradients in a more careful way than scaling the learning rate. We propose a simple to implement noise procedure, and present numerical experiments on image classification and language modeling tasks. In addition, we take Large Batch SGD to its logical conclusion and show that SGD can be effectively simulated by performing full-batch gradient descent and adding artificial noise to the gradients.