Understanding Stochastic Optimization Behavior at the Layer Update Level (Student Abstract)

Jack Zhang,[1*] **Guan Xiong Qiao,**[1†] **Alexandru Lopotenco,**[1†] **Ian Tong Pan** [1†]

[1]512 Technologies

jzhang0@sas.upenn.edu, tompan@wharton.upenn.edu, lopo@sas.upenn.edu, alanqiao@seas.upenn.edu

Abstract

Popular first-order stochastic optimization methods for deep neural networks (DNNs) are usually either accelerated schemes (e.g. stochastic gradient descent (SGD) with momentum) or adaptive step-size methods (e.g. Adam/AdaMax, AdaBelief). In many contexts, including image classification with DNNs, adaptive methods tend to generalize poorly compared to SGD, i.e. get stuck in non-robust local minima; however, SGD typically converges slower. We analyze possible reasons for this behavior by modeling gradient updates as vectors of random variables and comparing them to probabilistic bounds to identify "meaningful" updates. Through experiments, we observe that only layers close to the output have "definitely non-random" update behavior. In the future, the tools developed here may be useful in rigorously quantifying and analyzing intuitions about why some optimizers and particular DNN architectures perform better than others.

Introduction

Stochastic optimization is critical to many modern machine learning methods. Typically, these methods rely on some variant of stochastic gradient descent, which have certain useful convergence properties in the convex regime (Nesterov 1983). However, these properties do not necessarily translate into the non-convex regimes frequently encountered in DNNs and related optimization tasks such as saddle points, plateaus, and local minima (Dauphin et al. 2014). A lack of precise understanding of reasons for models failing to converge toward high accuracy and robustness precludes the construction of methods that effectively target these issues. Our research studies the fundamentals of how DNNs behave during training, offering possibly new perspectives for the high-dimensional, non-convex optimization of DNNs at the parameter update level.

Background

Many recent advances in DNN optimization come in the form of functions $g(\nabla_\theta(f_t(\theta_{t-1})), h)$ applied to the estimated gradient $\nabla_\theta(f_t(\theta_{t-1}))$ at time t with history, h, of gradients, updates, parameters, etc. (e.g. SGD with momentum, Adam, AdaBelief, Lookahead) (Qian 1999; Kingma and Ba 2017; Zhuang et al. 2020; Zhang et al. 2019). These are rooted in the human intuition of how DNNs learn the training distributions, and have proven effective. However, empirical analyses of precisely why these methods work better than simple SGD remain sparse. Overall, there has been relatively little focus on studying the behavior of the gradients and parameter updates that are at the core of all stochastic optimization methods.

Motivation

We can apply certain, well-studied, and useful probabilistic results about vectors in general and use them to guide our intuition about the nature of gradients. For instance, a standard observation is that for two vectors $x, y \in R^n$ drawn randomly from the n-ball B^n, they are likely to be approximately orthogonal for high dimensions (Arora 2013)

$$|\cos(\Theta_{x,y})| = O\left(\frac{\sqrt{\log(n)}}{n}\right).$$

Equivalently, for $\epsilon = \frac{1}{n}$:

$$Pr\left(|\cos(\theta_{x,y})| > \sqrt{\frac{-\log(\epsilon)}{n}}\right) < \epsilon.$$

Thus, if two high-dimensional vectors are not orthogonal, they likely were not generated randomly and independently.

We believe that intuition-based analysis of the gradient estimates are insufficient. We desire a more empirical analysis of the gradient estimates. Consider the following example: one might expect if estimated gradients are in the same direction then they likely approximate the true gradient well; indeed, this is the intuition behind AdaBelief (Zhuang et al. 2020), which demonstrates strong performance across multiple tasks.

However, it is worth noting that their method compares the current gradient to an exponential moving average of past gradients. Although we agree there is strong human intuition behind this concept, we do not think this lends itself to rigorous analysis in terms of likelihood that gradient estimates are approaching the true gradient. While we do not propose an alternative to AdaBelief, leaving that to future

*3335 Woodland Walk, Philadelphia PA 19104; +1 (603)-834-5923

†These authors contributed equally.

work, we attempt to move toward quantifying the behavior of optimizers at the gradient level in order empirically justify and explain novel methods like AdaBelief.

Experiments

We observed the gradient estimation behavior of various DNN architectures using SGD during training on the standard CIFAR-10 image classification task as a proof of concept for our general idea (Krizhevsky 2009). We consider the layer-specific gradient updates. In particular, we look at the following, for a given layer l, the cosine similarity $\cos\left(\nabla_{\theta,l}(f_t(\theta_{t-1})), \nabla_{\theta,l}(f_{t-1}(\theta_{t-2}))\right)$. We also then define non-random as having a large cosine similarity value with the previous update, where $|\theta_l|$ should be understood as the number of parameters in a given layer of parameters θ_l $\quad |\cos(\nabla_{\theta,l}(f_t(\theta_{t-1})), \nabla_{\theta,l}(f_{t-1}(\theta_{t-2})))| > \sqrt{-\log\left(\frac{1}{\theta_l}\right)|\theta_l|}$.

In particular, we consider ResNets (18, 34, 50) using SGD and SGD with momentum (He et al. 2015) with batch size 128. We use consistent parameters (lr = 0.05, no learning rate scheduler) and repeat each experiment three times with random initialization weights for replicability purposes.

Results and Observations

In the case of SGD with no momentum, we observe strange behavior dependent upon layer depth across all ResNets tested, which has never been noted previously (Figure 1).

In particular, we observe that only certain layers, namely those close to the output layer or skip connections, undergo highly non-random updates. This suggests non-random updates occur primarily near the output layer.

Although a cosine similarity value being below the threshold does not necessarily denote the absence of a "meaningful" update, only certain layers exhibit behavior consistent with randomness while others do not.

This is especially surprising considering the presence of skip connections in ResNets, which were originally designed to combat the vanishing gradient problem, and should intuitively ensure meaningful updates even in layers distant from the output (Pascanu, Mikolov, and Bengio 2012).

Even with a strong momentum term of 0.9, many hidden layers distant from the output layer appeared to have, on average, cosine similarity values near zero. This is unexpected, since a momentum term should intuitively push cosine similarities toward positive values.

Conclusion

Layers in ResNets near the output had highly non-random parameter updates during training. This implies that meaningful updates occur in the layers closer to the output during gradient-based optimization. It is unclear what causes this discrepancy between layers close to the output and layers distant from the output, even in DNNs with skip connections (e.g. backpropagation of error, sampling issues, batch sizes, etc.), and we encourage further research into these promising avenues toward rigorous understanding of DNN training.

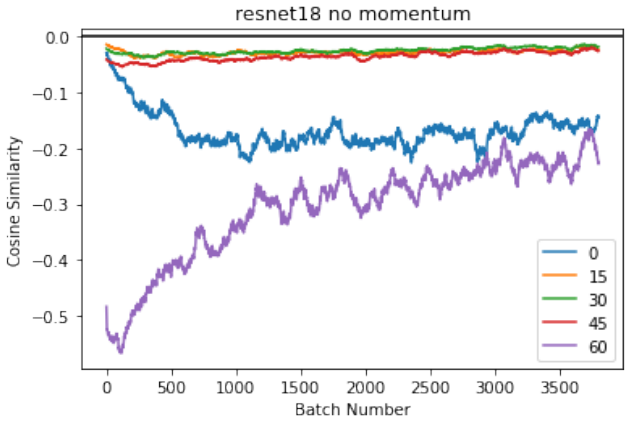

Figure 1: Cosine similarities between gradient updates, smoothed for visualization purposes.

References

Arora, S. 2013. Lecture 11: High Dimensional Geometry, Curse of Dimensionality, Dimension Reduction. University Lecture, Princeton University.

Dauphin, Y.; Pascanu, R.; Gulcehre, C.; Cho, K.; Ganguli, S.; and Bengio, Y. 2014. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. arXiv:1406.2572.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385.

Kingma, D. P.; and Ba, J. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980.

Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto.

Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 2278–2324.

Nesterov, Y. 1983. A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$. *Doklady AN USSR*, $269 : 543 - 547$.

Pascanu, R.; Mikolov, T.; and Bengio, Y. 2012. Understanding the exploding gradient problem. *CoRR*, abs/1211.5063.

Qian, N. 1999. On the Momentum Term in Gradient Descent Learning Algorithms. *Neural Networks*, 12: 145–151.

Simonyan, K.; and Zisserman, A. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556.

Zhang, M. R.; Lucas, J.; Hinton, G.; and Ba, J. 2019. Lookahead Optimizer: k steps forward, 1 step back. arXiv:1907.08610.

Zhuang, J.; Tang, T.; Ding, Y.; Tatikonda, S.; Dvornek, N.; Papademetris, X.; and Duncan, J. S. 2020. AdaBelief Optimizer: Adapting Stepsizes by the Belief in Observed Gradients. arXiv:2010.07468.