Assignment 8 - Adam, Adaptive Moment Estimation

Nisim Hurst

Tuesday 6 March 2018

Abstract

The authors present *Adam* on (Kingma and Ba 2014), an algorithm that uses lower-order moments estimates for first-order gradient optimization. Related stochastic algorithms that inspired Adam are exposed, along with the theoretical convergence properties to provide an approximate lower bound for convergence. However, empirical results are also presented. Future-work comments include AdaMax, a variant of Adam that uses the infinity norm.

Hypothesis

Adam is an acronym that stands for adaptive moment estimation. In the *Abstract* section the presented hypothesis is that it will help to be applied on large data or many parameters problems:

1. Use a small search space
2. Simple to implement
3. Hyper parameters are intuitive and typically require little tuning
4. Is invariant to rescaling of the gradients
5. Helps to smooth noisy and sparse gradients

Evidence and Results

Adam is an algorithm that has stood up in many domains and deep learning architectures. It takes the concepts of momentum (AdaGrad) and RMSprop and adds them together. The algorithm consist on the following steps:

1. Get first order gradients stochastic objective at time step t (deltas)
2. Update both biased first moment estimate and biased second moment raw estimate (exponentially weighted moving average)
3. Compute both bias-corrected first moment estimate and bias-corrected second moment raw estimate
4. Update parameter vector
5. Loop steps 1 to 6 while the parameter vector has not converged

The algorithm also uses initialization bias correction. This is a procedure that allows a direct correspondence between Adam and Adagrad by bounding parameter updates on the early steps. Bias correction works reducing the effect of early lack of data on the exponentially weighted average by dividing the raw momentums by $(1 - \beta^T)$ where T is the current step.

A theoretical convergence analysis is presented, where a theorem and a corollary prove that the algorithm converges on a $O(\sqrt{T})$. This analysis consist on proving the following:

Regret. sum of differences between the online prediction and the one produced by the optimal weights

1. **Theorem 1.** Assume that the function f has bounded gradients then $\sum_{i=1}^{d} \|g1:T,i\|_2 << dG_\infty\sqrt{T}$
2. **Corollary 1.** If $\sum_{i=1}^{d} \|g1:T,i\|_2 << dG_\infty\sqrt{T}$ then $\lim_{x\to\infty} \frac{R(T)}{T} = 0$

Experiments

Many models were compared with Adam. The hyperparameters are searched using dense grid and given.

Versus Logistic Regression

Two datasets were tested, multiclass MNIST and sparse IMDB BOW (bag of words). For the first test SGDNesterov and AdaGrad were compared. Adam achieved faster training cost decay. For the second test, a 50% dropout noise was added and RMSProp also was included in the competitors list. Adam performed equally well that Adagrad and better than the rest, in terms of convergence with lower training costs. This is due to the capability of Adam to take advantage of sparse features.

Versus Multi-layer Neural Networks

Although a direct comparison is not possible between non-convex objective function models, some results were obtained using sum-of-functions method (Sohl-Dickstein et al., 2014) using the MNIST dataset. A range of 5-10x improvement was found with Adam, please see the original paper for the details of the implementation. A possible explanation given by the authors is that SFO assumes deterministic subfunctions while Adam can work equally well with stochastic regularization.

Versus Convolutional Neural Networks

The components of the architecture are show in Table 1 :

Table 1: CNN architecture

Element	Quantity
5x5 convolution filters	3
3x3 max pooling	2
fully connected layer of 1000 ReLU	1

In every case Adam outperforms it's competitors except for the case in which is combined with dropout and then SGD with momentum. This owns to the fact that the second moment estimate vanishes in the geometry of CNNs.

Using Bias-correction Term

A grid of different curves for each hiperparameter was made to evaluate the effect of bias-correction removal. Gradients tend to become sparser on arranges with high β_2 and low β_1. Adam performed at least as well as RMSProp when bias correction was removed.

Contribution

The paper provides an benchmark of algorithms for optimization of stochastic objectives with high dimensional parameter spaces, and an algorithm called Adam that excels in this area. The real contribution of Adam lies on the bias correction procedure that allows Adagrad and RMSProp work consistently and converge quickly.

Adam present an algorithm that combines both the first raw moment moving average and the second raw moment (the square of the derivative) for generating a procedure that will converge. It shows a theoretical improvement of $O(\sqrt{dt})$ over non adaptive momentum methods.

The most important difference between AdaGrad is that it can be considered as a special case of Adam in which β_1 equals 0 and thus, doesn't take into account the second order momentum.

The most important difference between RMSProp with momentum from Adam is that RMSProp updates its parameters by rescaling the gradient, while Adam uses the exponentially weighted average of the second

moment of the gradient. RMSProp also doesn't uses bias-correction, in case of sparse gradients it can lead to divergence.

Weaknesses

The paper does not compare efficiency on consistent and compact gradients datasets except for the MNIST multiclass. Even though a brief theoretical convergence analysis was made, it is important not only to evaluate the effect on training costs, but also processing time.

Also the comparison with multilayer neural networks can vary between each architecture and it's sparseness. Future work could include a dense grid search for hyperparameter comparison.

Future Work

Other extensions to *Adam* besides *AdaMax* can also be made for example calculating the Nesterov momentum for the gradient vector and then changing the bias-correction accordingly. *AdaMax* uses the infinity norm which is a generalization on the update rule that scales the gradients on a norm of current and past gradients. A recursive function reduces the update rule to $u_t = \max(\beta_2 * u_{t-1}, |g_t|)$, similar to the Minkowski distance generalization. In this case no bias-correction is needed.

An interesting field of study would be to explore the effect of the infinity norm modification on weight decay and its geometric interpretation compared to Adam.

References

Kingma, Diederik P., and Jimmy Ba. 2014. "Adam: A Method for Stochastic Optimization," 1–15. doi:http://doi.acm.org.ezproxy.lib.ucf.edu/10.1145/1830483.1830503.