Enhancements to Adam Optimizer

TTIC 31230 Fundamentals of Deep Learning

Amit Pradhan

December 4, 2021

1 Introduction

Adam[1], a popular optimization algorithm, probably most used optimizer in machine learning systems. It is a first order gradient based optimization algorithm which takes adaptive estimates of lower order moments into account. In some sense, it combines the ideas from RMSProp and momentum. Adam also uses bias correction. The method is very straightforward to implement, computationally efficient, requires less memory, invariant to diagonal scaling of gradients and suitable for models with large number of parameters and gradients. It also works well with non-stationary distributions and models with noisy/sparse gradients.

2 Objective

My primary objective is to simplify the Adam algorithm without degrading much in terms of model performance. One simplified version of the Adam algorithm was discussed in the class. We will compare the simplified version of the algorithm with original. I will also play with the bias correction terms and compare the performances of model when we remove either one or both of the bias terms. I will also experiment with higher order moments to see if they offer any performance benefits.

3 Background

Standard Adam algorithm is described below:

$$g_t \leftarrow \nabla_\phi f_t(\phi_{t-1}) \tag{1}$$

$$m_t \leftarrow \frac{\beta_1 m_{t-1} + (1-\beta_1) g_t}{1-\beta_1^t} \tag{2}$$

$$v_t \leftarrow \frac{\beta_2 v_{t-1} + (1-\beta_2) g_t^2}{1-\beta_2^t} \tag{3}$$

$$\phi_{t+1} \leftarrow \phi_t - \frac{\eta m_t}{\sqrt{v_t} + \epsilon} \tag{4}$$

$$\tag{5}$$

where η is step size, $\beta_1, \beta_2 \in [0, 1)$ are the exponential decay rates for the moment estimates, $L(\phi)$ loss function with parameters ϕ, t current step, g_t gradient with respect to parameters ϕ, m_t first order moment, v_t second order moment.

A simplified Adam algorithm as discussed in the class is described below:

$$g_t \leftarrow \nabla_\phi f_t(\phi_{t-1}) \tag{6}$$

$$m_t \leftarrow (1 - \frac{1}{min(t, N_1)}) m_{t-1} + \frac{1}{min(t, N_1)} g_t \tag{7}$$

$$v_t \leftarrow (1 - \frac{1}{min(t, N_2)}) v_{t-1} + \frac{1}{min(t, N_1)} g_t^2 \tag{8}$$

where N_1, N_2 are typically 100 and 1000.

Adam with higher order (k) moment is described below:

$$g_t \leftarrow \nabla_\phi f_t(\phi_{t-1}) \tag{9}$$
$$m_t \leftarrow \frac{\beta_1 m_{t-1} + (1-\beta_1) g_t}{1-\beta_1^t} \tag{10}$$
$$v_t \leftarrow \frac{\beta_2 v_{t-1} + (1-\beta_2) g_t^k}{1-\beta_2^t} \tag{11}$$
$$\phi_{t+1} \leftarrow \phi_t - \frac{\eta m_t}{\sqrt[k]{v_t} + \epsilon} \tag{12}$$
$$\tag{13}$$

Adam algorithm without first order moment correction is described below:

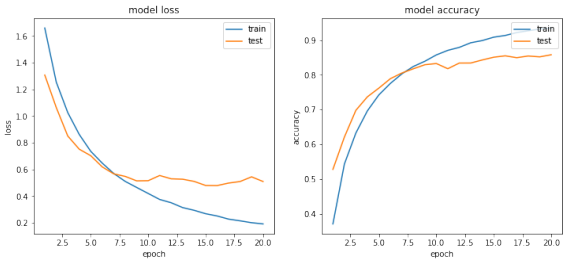

Figure 1: (a) Train and Test Loss vs Epoch, (b) Train and Test Accuracy vs Epoch, using standard Adam optimizer

$$g_t \leftarrow \nabla_\phi f_t(\phi_{t-1}) \tag{14}$$
$$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{15}$$
$$v_t \leftarrow \frac{\beta_2 v_{t-1} + (1-\beta_2) g_t^2}{1-\beta_2^t} \tag{16}$$
$$\phi_{t+1} \leftarrow \phi_t - \frac{\eta m_t}{\sqrt{v_t} + \epsilon} \tag{17}$$
$$\tag{18}$$

Adam algorithm without second order moment correction is described below:

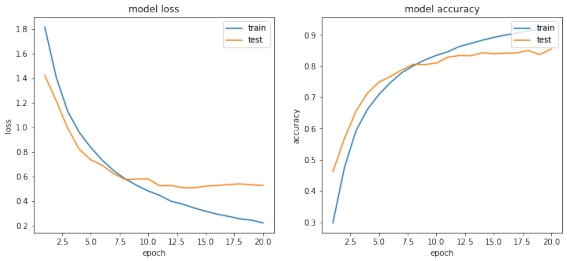

Figure 2: (a) Train and Test Loss vs Epoch, (b) Train and Test Accuracy vs Epoch, using simplified Adam optimizer

$$g_t \leftarrow \nabla_\phi f_t(\phi_{t-1}) \tag{19}$$
$$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{20}$$
$$m_t \leftarrow \frac{\beta_1 m_{t-1} + (1-\beta_1) g_t}{1-\beta_1^t} \tag{21}$$
$$v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \tag{22}$$
$$\tag{23}$$

Adam algorithm without moment correction is described below:

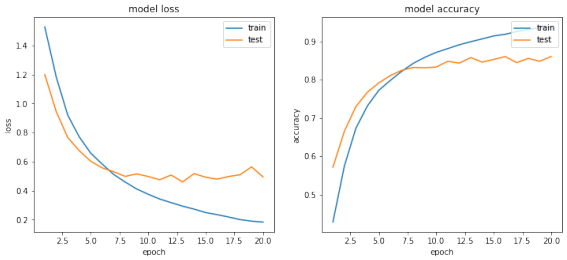

Figure 3: (a) Train and Test Loss vs Epoch, (b) Train and Test Accuracy vs Epoch, using Adam optimizer without bias correction of the first moment

$$g_t \leftarrow \nabla_\phi f_t(\phi_{t-1}) \tag{24}$$
$$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{25}$$
$$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{26}$$
$$v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \tag{27}$$
$$\tag{28}$$

4 Results

I have trained a Resnet like model with CIFAR-10 dataset with the standard image classification objective. For each experiment, I use a different variations of Adam optimizer while keeping all other hyper-parameters constant. For every experiment, the model is trained for 20 epochs. Initial learning rate η_0, β_1 and β_2 is 0.001, 0.99 and 0.999 for all the experiments. For each experiment, I have plotted loss vs epoch and accuracy vs epoch for both train and test datasets.

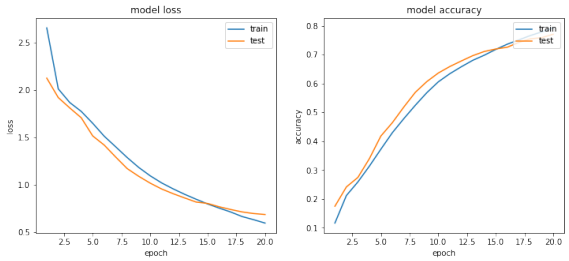

Figure 4: (a) Train and Test Loss vs Epoch, (b) Train and Test Accuracy vs Epoch, using Adam optimizer without bias correction of the second moment

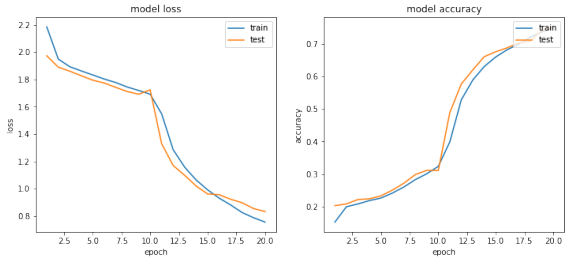

Figure 5: (a) Train and Test Loss vs Epoch, (b) Train and Test Accuracy vs Epoch, using Adam optimizer without bias correction

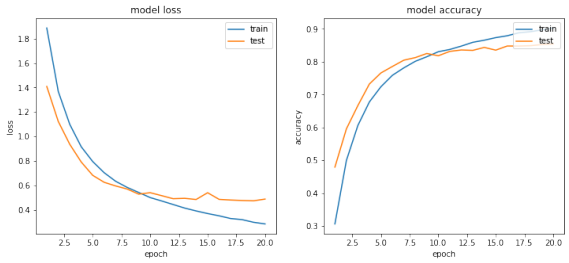

Figure 6: (a) Train and Test Loss vs Epoch, (b) Train and Test Accuracy vs Epoch, using Adam optimizer with third order moment

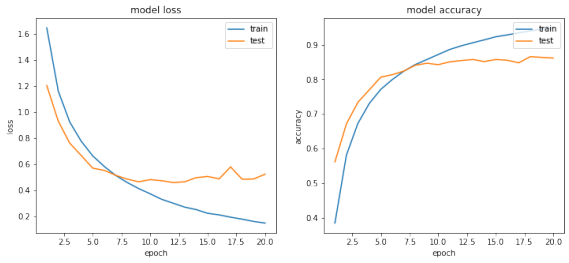

Figure 7: (a) Train and Test Loss vs Epoch, (b) Train and Test Accuracy vs Epoch, using Adam optimizer with fourth order moment

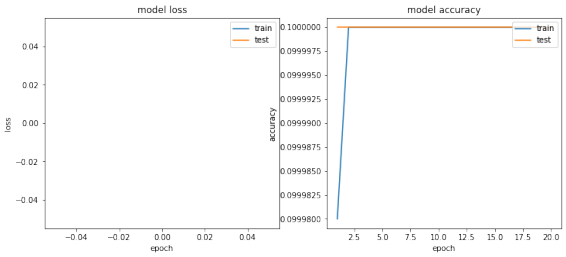

Figure 8: (a) Train and Test Loss vs Epoch, (b) Train and Test Accuracy vs Epoch, using Adam optimizer with fifth order moment (model diverged with infinity loss values)

Conclusions

The simplified version of the Adam optimizer performs close to the original Adam optimizer. It reaches nearly similar test and train accuracy. Among the higher order moment optimizer, Adam with fourth order moment performs better than the original Adam. Adam with fifth order moment diverged with loss values as infinity. Dropping only first moment correction gives better generalization than the original Adam.

From the experiments, we can conclude the simplified version can achieve close to original Adam performance. However, Adam optimizer itself is very straightforward to implement and does not add any computational or memory overhead. So if it is not absolutely necessary to simplify the algorithm, we can live with the original Adam algorithm.

5 Appendix

The experimental results are available in link

References

[1] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Figure 9: Performances of different optimizers after 20 epoch

Optimizer	Train Loss	Test Loss	Train Accuracy	Test Accuracy
Adam	0.19	0.51	93.46	85.77
Simplified	0.22	0.53	92.54	85.52
Third Order	0.28	0.49	90.28	85.42
Fourth Order	0.15	0.52	94.84	86.14
Adam w/o first moment correction	0.18	0.49	93.78	86.07
Adam w/o second moment correction	0.60	0.69	79.28	77.08
Adam w/o moment correction	0.76	0.83	79.94	75.23