Adam: A Method For Stochastic Optimization

1 Citation

Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

`https://arxiv.org/pdf/1412.6980.pdf`

2 Abstract

Adam is a variant of stochastic gradient descent that is invariant to diagonal rescaling of gradients, has intuitive hyperparameters, and converges quickly.

3 Introduction

First order optimization techniques like SGD are nice because they have the same runtime complexity as evaluating the function they are trying to optimize. Adam (from the name Adaptive Moments) aims to combine Adagrad (works well for sparse gradients) and RMSProp (works well for non-stationary functions).

4 Algorithm

Our function is $f(\theta)$ and the gradient at step t is $g_t = \nabla_\theta f_t(\theta)$. We keep an exponential moving average (EMA) of the gradient (m_t) and its square (v_t) - these are the moments. We initialize the moments with the zero vector (we'll show how to correct this bias). β_1 and β_2 are the EMA weights for m_t and v_t, respectively. α is our learning rate. ϵ is a small constant to prevent division by zero. At time step t, our update rule is:

$$g_t = \nabla_\theta f_t(\theta_{t-1}) \tag{1}$$

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

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)(g_t \odot g_t) \tag{3}$$

$$\hat{m}_t = m_t/(1-\beta_1^t) \tag{4}$$

$$\hat{v}_t = v_t/(1-\beta_2^t) \tag{5}$$

$$\theta_t = \theta_{t-1} - \alpha \hat{m}_t/(\sqrt{\hat{v}_t} + \epsilon) \tag{6}$$

The $\hat{m}_t$ and $\hat{v}_t$ are the bias corrected moments. Roughly, we draw a region of size α around the parameters and our step is proportional to the size of the gradient $\hat{m}_t$ but inversely proportional to the uncertainty in this estimate $\sqrt{\hat{v}_t}$.

5 Initialization Bias Correction

Initializing $m_0 = v_0 = 0$ introduces bias. Let's consider v_t and see how to correct the bias.

$$v_t = (1 - \beta_2) \sum_{i=1}^{t} \beta_2^{t-i} (g_t \odot g_t) \tag{7}$$

$$\mathbb{E}[v_t] = E[g_t \odot g_t](1 - \beta_2^t) + \zeta \tag{8}$$

ζ is a small constant, so we can throw it away. Thus, our bias correction is $1/(1 - \beta_2^t)$.

6 Convergence Analysis

Given unknown, arbitrary, sequence of convex cost functions $f_1(\theta), f_2(\theta), ..., f_T(\theta)$, we aim to minimize regret:

$$R(T) = \sum_{t=1}^{T} [f_t(\theta_t) - f_t(\theta^*)] \tag{9}$$

$$\text{where } \theta^* = \text{argmin}_{\theta \in \mathcal{X}} \sum_{t=1}^{T} f_t(\theta) \tag{10}$$

Adam has a $O(\sqrt{T})$ regret bound. This only holds for convex functions, but the algorithm still works quite well for nonconvex functions.

7 Related Work

RMSProp uses gradient rescaling to compute momentum instead of keeping track of the first and second moments. It also doesn't do bias correction. Adagrad does $\theta_{t+1} = \theta_t - \alpha g_t / \sqrt{\sum_{i=1}^{t} g_i \odot g_i}$, so it doesn't do an EMA on the first moment (gradient) or second moment (squared gradient).

8 Experiments

We consider logistic regression, feedfoward nets, and convnets. Hyperparameters are chosen with a validation set. We measure how many steps it takes for the optimizers to converge.

We do logistic regression on MNIST. Adam converges similarly to SGD with momentum and both are faster than Adagrad. We also consider IMDB movie rating prediction on a bag-of-words of the reviews. Adam ties with Adagrad and beats SGD with momentum.

On a feedforward net with two hidden layers, we beat the SFO optimizer (which is also really slow), even if we use dropout.

On a convnet for image classification, Adam does a little better than SGD with momentum and a lot better than Adagrad.

If you remove the bias correction, Adam does a lot worse.

9 Extensions

We consider the second moment for v_t, but we could also consider the "infinite" moment instead. This gives the Adamax algorithm:

$$
\begin{align}
g_t &= \nabla_\theta f_t(\theta_{t-1}) \tag{11} \\
m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{12} \\
u_t &= \max(\beta_2 u_{t-1}, |g_t|) \tag{13} \\
\theta_t &= \theta_{t-1} - (\alpha/(1 - \beta_1^t))(m_t/u_t) \tag{14}
\end{align}
$$

Another extension is to keep a running average of θ_t values with an EMA (and bias correct it bebcause it is initialized with the zero vector).

10 Conclusion

Adam combines ideas from Adagrad (good for sparse gradients) and RMS prop (good for nonstationary objectives) to yield a fast-converging optimization algorithm for machine learning.