TTIC 31230, Fundamentals of Deep Learning

David McAllester, Autumn 2020

Stochastic Gradient Descent (SGD)

RMSProp and Adam

RMSProp and Adam

RMSProp and Adam are “adaptive” SGD methods — they use different learning rates for different model parameters where the parameter-specific learning rate is computed from statistics of the data.

Adam is variant of RMSProp with momentum and “debiasing”.

RMSProp

RMSProp was introduced in Hinton's class lecture slides.

RMSProp is based on a running average of $\hat{g}[i]^2$ for each scalar model parameter i.

$$s_t[i] = \left(1 - \frac{1}{N_s}\right) s_{t-1}[i] + \frac{1}{N_s}\hat{g}_t[i]^2 \quad N_s \text{ typically 100 or 1000}$$

$$\Phi_{t+1}[i] = \Phi_t[i] - \frac{\eta}{\sqrt{s_t[i]} + \epsilon}\; \hat{g}_t[i]$$

RMSProp

The second moment of a scalar random variable x is $E\ x^2$

The variance σ^2 of x is $E\ (x - \mu)^2$ with $\mu = E\ x$.

RMSProp uses an estimate $s[i]$ of the second moment of the random scalar $\hat{g}[i]$.

If the mean $g[i]$ is small then $s[i]$ approximates the variance of $\hat{g}[i]$.

There is a "centering" option in PyTorch RMSProp that switches from the second moment to the variance.

RMSProp Analysis

For sufficiently small ϵ, RMSProp makes the update independent of the scale of the gradient.

$$\Phi[i] \mathrel{-}= \eta \, \frac{\hat{g}[i]}{\sqrt{s[i]}}$$

If the gradient $\hat{g}_i$ is scaled up by a factor of α we have that $\sqrt{s[i]}$ is also scaled by α.

This makes the learning rate η dimensionless.

RMSProp Analysis

$$\Phi[i] \mathrel{-}= \eta\, \frac{\hat{g}[i]}{\sqrt{s[i]}}$$

It is not completely clear why making the update invariant to gradient scaling is a good thing.

It would be nice to have an analysis directly tied to convergence rates.

Adam — Adaptive Momentum

Adam combines momentum and RMSProp.

PyTorch RMSProp also supports momentum. However, it presumably uses the standard momentum learning rate parameter which couples the temperature to both the learning rate and the momentum parameter. Without an understanding of the coupling to temperature, hyper-parameter optimization is then difficult.

Adam uses a momentum parameter that is naturally decoupled from temperature.

Adam also uses "bias correction".

Bias Correction

Consider a standard moving average.

$$\begin{aligned} \tilde{x}_0 &= 0 \\ \tilde{x}_t &= \left(1 - \frac{1}{N}\right)\tilde{x}_{t-1} + \left(\frac{1}{N}\right)x_t \end{aligned}$$

For $t < N$ the average $\tilde{x}_t$ will be strongly biased toward zero.

Bias Correction

The following running average maintains the invariant that $\tilde{x}_t$ is exactly the average of $x_1, \ldots, x_t$.

$$\tilde{x}_t = \left(\frac{t-1}{t}\right)\tilde{x}_{t-1} + \left(\frac{1}{t}\right)x_t$$

$$= \left(1 - \frac{1}{t}\right)\tilde{x}_{t-1} + \left(\frac{1}{t}\right)x_t$$

We now have $\tilde{x}_1 = x_1$ independent of any x_0.

But this fails to track a moving average for $t >> N$.

Bias Correction

The following avoids the initial bias toward zero while still tracking a moving average.

$$\tilde{x}_t = \left(1 - \frac{1}{\min(N,t)}\right)\tilde{x}_{t-1} + \left(\frac{1}{\min(N,t)}\right)x_t$$

The published version of Adam has a more obscure form of bias correction which yields essentially the same effect.

Adam (simplified)

$$\tilde{g}_t[i] \;=\; \left(1 - \frac{1}{\min(t, N_g)}\right)\tilde{g}_{t-1}[i] + \frac{1}{\min(t, N_g)}\hat{g}_t[i]$$

$$s_t[i] \;=\; \left(1 - \frac{1}{\min(t, N_s)}\right)s_{t-1}[i] + \frac{1}{\min(t, N_s)}\hat{g}_t[i]^2$$

$$\Phi_{t+1}[i] \;=\; \Phi_t[i] - \frac{\eta}{\sqrt{s_t[i]} + \epsilon}\;\; \tilde{g}_t[i]$$

Decoupling η from ϵ

$$\Phi_{t+1}[i] = \Phi_t - \frac{\eta}{\sqrt{s_t[i]} + \epsilon}\;\; \tilde{g}_t[i]$$

The optimal ϵ is sometimes large. For large ϵ it is useful to set $\eta = \epsilon\eta_0$ in which case we get

$$\Phi_{t+1}[i] = \Phi_t - \frac{\eta_0}{1 + \frac{1}{\epsilon}\sqrt{s_t[i]}}\;\; \tilde{g}_t[i]$$

We then get standard SGD as $\epsilon \to \infty$ holding η_0 fixed.

Making Adam Adapt to the Batch Size B

Adam alreaady adapts to momentum by using and EMA of the gradient as momentum. Adapting to batch size requires some analysis.

$$\Phi[i] \mathrel{-}= \eta \, \frac{\hat{g}[i]}{\sqrt{s[i]}}$$

Since we are taking a long moving average of $\hat{g}[i]^2$ we can assume $s[i] = E\ \hat{g}[i]^2$ and we have

$$\begin{aligned} s[i] &= E\ \hat{g}[i]^2 \;\; = \;\; \mu^2 + \sigma^2/B \\ \mu &= E\ \hat{g}[i] \; = \; E\ g[i] \\ \sigma^2 &= E(g[i] - \mu)^2 \end{aligned}$$

Making Adam Adapt to the Batch Size B

$$\Phi[i] \mathrel{-=} \eta \, \frac{\hat{g}[i]}{\sqrt{\mu^2 + \sigma^2/B}}$$

$$E\ \Delta\Phi \ = \ -\eta \frac{\mu}{\sqrt{\mu^2 + \sigma^2/B}}$$

For $\mu^2 >> \sigma^2/B$ we want $\eta = B\eta_0$ as with vanilla SGD.

For $\sigma^2/B >> \mu^2$ we want $\eta = \sqrt{B}\ \eta_0$.

Making Adam Independent of B

The previous analysis looses information. We could estmate $g[i]^2$ more accurately by computing $g[i]^2$ for each batch element rather than measuring $\hat{g}[i]^2$ where $\hat{g}[i]$ is alread averaged over the batch.

This would require adding a batch index to the parameter gradients.

END