TTIC 31230, Fundamentals of Deep Learning

David McAllester, April 2017

SGD Variants

Review

- A Computation Graph is a sequence of assignment statements $y = f(x)$.

- In the EDF frameword a computation is implemwnted with assignments $y = F(x)$ where x and y are **objects** with **value attributes** $x.\text{value}$ and $y.\text{value}$.

- Backprogation on computation graphs produces attributes $x.\text{grad} = \partial \ell.\text{value}/\partial x.\text{value}$.

Review

- EDF supports minibatching. For inputs and computed values x we have that x.value and x.grad now **contain an entire batch** where the first index is the batch index.

- For parameters W we have that W.value and W.grad do not have a batch index but are instead averaged over the batch.

- Minibatching is **required** for efficiency.

Central SGD Issues

Consider a parameter vector Θ.

- **Gradient Estimation.** The need to estimate the gradient at a fixed Θ.
- **Gradient Drift.** The fact that the gradient changes as Θ changes.
- **Exploration.** Since deep models are non-convex we need to search over the parameter space. SGD can behave like MCMC.

An Example

Consider the following where $\hat{\mu}$ and y are scalars.

$$\ell_n(\hat{\mu}) = \frac{1}{2}(\hat{\mu} - y_n)^2 \qquad y_n \in \{-1, 1\}$$

For random n repeat:

$$\begin{aligned} \hat{\mu} \texttt{ -= } & \eta(d\ell_n/d\hat{\mu}) \\ = & \ \eta(\hat{\mu} - y_n) \\ \hat{\mu}^* = & \ \mathrm{E}\left[y\right] \end{aligned}$$

The updates reflect the true gradient plus stochastic noise.

This defines a stochastic process with an equilibrium density $p[\hat{\mu}]$.

As $\eta \to 0$ the width of this distribution goes to zero.

Gradient Flow

Gradient Descent: $\Theta \mathrel{-}= \eta \nabla_\Theta \ell(\Theta) \qquad \ell(\Theta) = \frac{1}{N} \sum_{n=1}^{N} \ell_n(\Theta)$

Here we are considering total gradient descent (ignoring the gradient estimation problem).

Take the limit $\eta \to 0$ with the update repeated $\lfloor T/\eta \rfloor$ times.

$$d\Theta = -dt \, \nabla_\Theta \, \ell(\Theta) \qquad \text{or} \qquad \frac{d\Theta}{dt} = -\nabla_\Theta \, \ell(\Theta)$$

The limit integrates this differential equation from $t = 0$ to $t = T$.

Stochastic Gradient Flow

SGD: Θ -= $\eta \nabla_\Theta \, \ell(\Theta, \, x_n, \, y_n)$ for random n

Again take the limit $\eta \to 0$ with the update repeated $\lfloor T/\eta \rfloor$ times.

As $\eta \to 0$ we get an arbitrarily large number of updates with no gradient drift.

The direction of motion then becomes deterministic and we get the same limiting differential equation.

$$\frac{d\Theta}{dt} = -\nabla_\Theta \, \ell(\Theta)$$

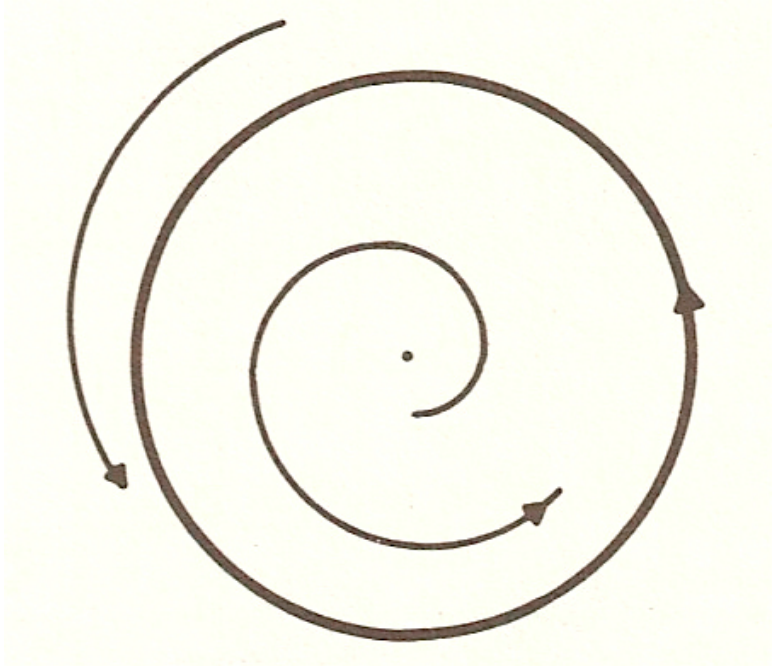
Gradient Flow Guarantees Progress

$$\begin{aligned}
\frac{d\ell}{dt} &= (\nabla_\Theta \, \ell(\Theta)) \cdot \frac{d\Theta}{dt} \\
&= -(\nabla_\Theta \, \ell(\Theta)) \cdot (\nabla_\Theta \, \ell(\Theta)) \\
&= -||\nabla_\Theta \, \ell(\Theta)||^2 \\
&\leq 0
\end{aligned}$$

If $\ell(\Theta) \geq 0$ then $\ell(\Theta)$ must converge to a limiting value.

This does not imply that Θ converges.

Limit Cycles



It is possible that the value converges but the parameters do not.

In practice if the value converges the parameters will also converge.

Figure from the web page "First Order ODEs" by Mike Martin

A Classical Convergence Theorem

Consider

$$\Theta \mathrel{-}= \eta_t \nabla_\Theta \, \ell(\Theta)$$

For $\ell(\Theta)$ "sufficiently smooth" with $\ell(\Theta) \geq 0$ and

$$\eta_t > 0 \quad \text{and} \quad \lim_{t \to 0} \eta_t = 0 \quad \text{and} \quad \sum_t \eta_t = \infty,$$

we have that $\ell(\Theta)$ will converge.

See "Neuro-Dynamic Programming" by Bertsekas and Tsitsiklis proposition 3.5.

Again, there are pathological (unrealistic) cases where Θ enters a limit cycle and fails to converge.

Review of Minibatch SGD

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} \, \mathrm{E}_i \left[\ell_i(\Theta) \right]$$

minibatch SGD:

repeat:

Select a minibatch B at random

$$\Theta \mathrel{-}= \eta \, \frac{1}{|B|} \sum_{i \in B} \nabla_\Theta \, \ell_i(\Theta)$$

Each vector operation in the implementation operates on the entire batch.

Minibatching is **required** for efficiency.

Popular SGD variants

Momentum

Nesterov Momentum

RMSProp

Adam

Momentum

$$\hat{g}^{t+1} = \mu \hat{g}^t + (1-\mu)\nabla_\Theta \, \ell^t(\Theta) \qquad \mu \in (0,1)$$

$$\Theta^{t+1} = \Theta^t - \eta \hat{g}^{t+1}$$

Each $\nabla_\Theta \ell^t(\Theta)$ is an average over a minibatch of size B.

$\hat{g}$ is a running average of $\nabla_\Theta \ell^t(\Theta)$

For $\mu = .9$ we intuitively have that $\hat{g}$ is an average over $10B$ gradients.

Here $\hat{g}$ is averaged over different model parameters Θ.

A Comment on Presentation

Momentum is often presented in the following equivalent way.

$$v^{t+1} = \mu v^t + \eta' \nabla_\Theta \, \ell^t(\Theta) \qquad \mu \in (0, 1)$$

$$\Theta^{t+1} = \Theta^t - v^{t+1}$$

However, setting $\eta = \eta'/(1-\mu)$ gives $v^t = (\eta'/(1-\mu))\hat{g}^t$ and the same sequence Θ^t.

The semantics of $\hat{g}^t$ seems clearer than the semantics of v^t.

A similar comment applies to Nesterov Momentum below.

Nesterov Momentum

$$\hat{g}^{t+1} = \mu \hat{g}^t + (1-\mu) \nabla_\Theta \, \ell^t(\Theta) @ (\Theta^t - \eta \mu g^t)$$

$$\Theta^{t+1} = \Theta^t - \eta \hat{g}^{t+1}$$

This is very similar to standard momentum except that the gradient is measured at a "lookahead" parameter value different from both Θ^t and Θ^{t+1}.

RMSProp

Adaptive Feature-specific Learning Rates.

RMS — Root Mean Square

$$s_i^{t+1} = \beta s_i^t + (1-\beta)\left(\nabla_\Theta \ell^t(\Theta)\right)_i^2$$

s_i^t is a mean square.

$$\Theta_i^{t+1} = \Theta_i^t - \frac{\eta}{\sqrt{s_i^{t+1}} + \epsilon} \quad \left(\nabla_\Theta \ell^t(\Theta)\right)_i$$

Adam — Adaptive Momentum

$$\hat{g}_i^{t+1} = \beta_1 \hat{g}_i^t + (1-\beta_1)\left(\nabla_\Theta \, \ell^t(\Theta)\right)_i$$

$$s_i^{t+1} = \beta_2 s_i^t + (1-\beta_2)\left(\nabla_\Theta \, \ell^t(\Theta)\right)_i^2$$

$$\Theta_i^{t+1} = \Theta_i^t - \frac{\eta}{\sqrt{s_i^{t+1}} + \epsilon} \; \hat{g}_i^{t+1}$$

Review of Issues

- **Gradient Estimation.** The need to estimate the gradient at a fixed Θ.

- **Gradient Drift.** The fact that the gradient changes as Θ changes.

- **Exploration.** Since deep models are non-convex we need to search over the parameter space. SGD can behave like MCMC.

Comments

From empirical experience, Adam is generally recommended.

Adam seems less sensitive to the η parameter.

However, vanilla SGD remains competative when η is carefully tuned.

The exploration performed by SGD (similar to MCMC) seems important.

END