Simulation study of Stochastic Gradient Descent Algorithms

Balaji Kumar

STAT 540

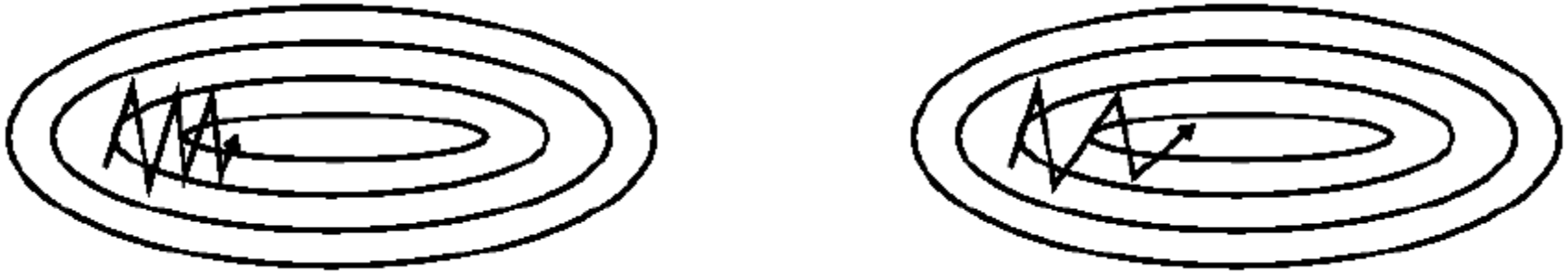
24 April 2018

Stochastic Gradient Descent

- Randomly pick one or more data points to compute $\nabla J(\theta, x^*)$;
- Update: $\theta_{i+1} = \theta_i - \eta \nabla J(\theta, x^*)$
- Challenges:
  - No convergence guaranteed
  - Choosing “ideal” learning rate can be difficult.
  - Learning rate schedule needs to be fine tuned according to data
  - Having same learning rate for all parameters is not optimal
  - SGD gets trapped in saddle points in non-convex optimization (Dauphin et al, 2015)

Momentum

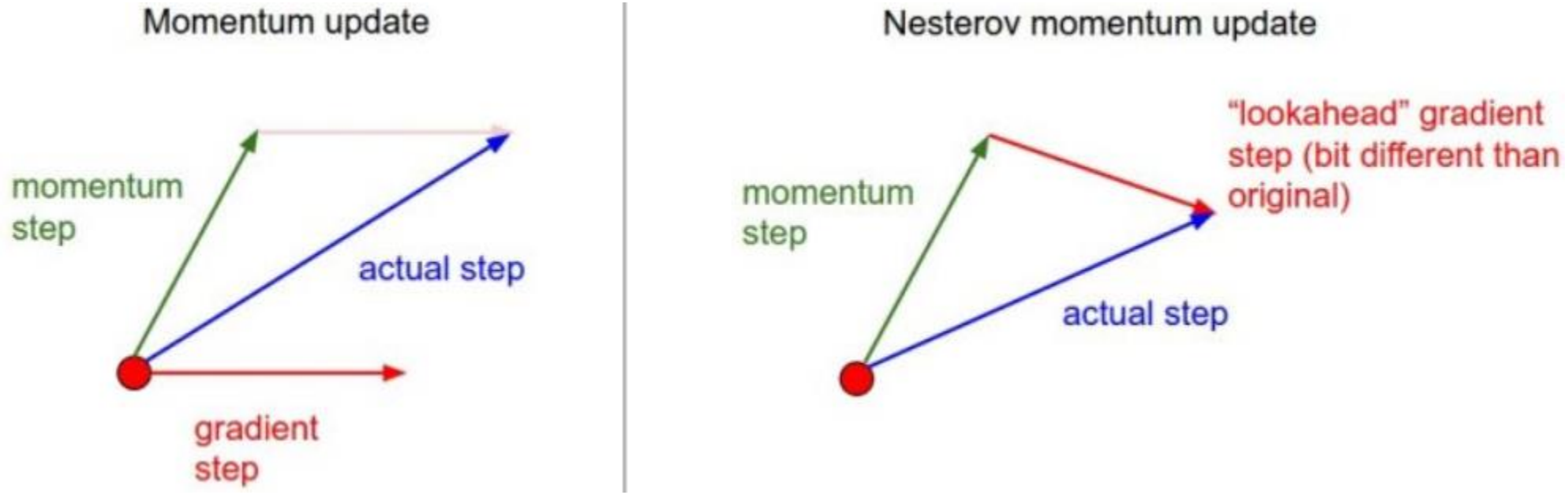
- Ning Qian, 1999
- $v_t = \gamma v_{t-1} + \eta \nabla J(\theta, x^*)$
- $\theta_{i+1} = \theta_i - v_t$
- SGD oscillates in ravines (Sutton et al, 1986)



- Figure left without momentum and right with momentum

Nesterov Accelerated Gradient

- Yurii Nesterov, 1983: Less blind than Momentum
- $v_t = \gamma v_{t-1} + \eta \nabla J(\theta - \gamma v_{t-1}, x^*)$
- $\theta_{i+1} = \theta_i - v_t$
-



Adaptive Learning Methods

- Adagrad (Singer et al, 2011): Per-parameter η
  - $\theta_{t,i+1} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \varepsilon}} \nabla J\left(\theta_{t,i}, x^*\right)$
  - Dean et al 2006: Adagrad more robust than SGD for sparse data
  - No need to tune learning rate.
- Adadelta (Zeiler 2012): More adaptive and less monotonic-decreasing
  - $E[g^2]_t = \gamma E[g^2]_{t-1} + (1-\gamma) g_t^2$
  - $E[\Delta\theta^2]_t = \gamma E[\Delta\theta^2]_{t-1} + (1-\gamma) \Delta\theta_t^2$
  - $\theta_{t,i+1} = \theta_{t,i} - \frac{\sqrt{E[\Delta\theta^2]_{t-1} + \varepsilon}}{\sqrt{E[g^2]_t + \varepsilon}} \nabla J\left(\theta_{t,i}, x^*\right)$

Adaptive Learning Methods Contd.

- Adaptive Moment Estimation (Kingma and Lei Ba, 2015):
- $m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$
- $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$
- $\widehat{m}_t = \frac{m_t}{1-\beta_1^t}$
- $\hat{v}_t = \frac{v_t}{1-\beta_2^t}$
- $\theta_{t+1} = \theta_t - \frac{\eta}{\varepsilon + \sqrt{\hat{v}_t}} \widehat{m}_t$
- Default values for hyper-parameters work well in practice

Studying Update step

- Finding minimum of Beale function:
- $(1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$
- Multimodal, saddle points

Algorithm	x1,x2	F(x)	# Steps
SGD	(-2.51,1.30)	0.97	78
Momentum	(-2.5,1.3)	1	496
NAG	(-2.5,1.3)	**0.9**	**56**
Adadelta	(-1.4,1.4)	1.3	3148
Adam	(-1.6,1.5)	1.7	6362

Simulations

- $Y = \beta_0 + X\beta_1 + N(0,\sigma^2)$
- 1000 data points: 75-25 training-test

Algorithm	MSE	# Steps
SGD	852	12449
Momentum	8.2	2411
NAG	**2.78**	**2756**
Adadelta	4e+9	Stopped
Adam	**1.57**	20976

Conclusions

- Tuning hyper-parameters is not as easy as just backtracking
- Adaptive Learning algorithms sometimes can’t be tuned with backtracking
- NAG is fast and robust. Adam is more expensive and slower to converge and cumbersome to tune.
- Something seems wrong with Adadelta. It is the one algorithm doesn’t need tuning!

Future Work

- Fix Adadelta
- Apply Stochastic Gradient Descent Algorithms to real data