Adaptive Learning Rate

Thang Vu

Stochastic Gradient Descent

- Gradient Descent:

$$\theta_i \leftarrow \theta_{i-1} - \eta \nabla C(\theta_{i-1})$$

$$\nabla C(\theta_{i-1}) = \frac{1}{R}\sum_r \nabla C^r(\theta)$$

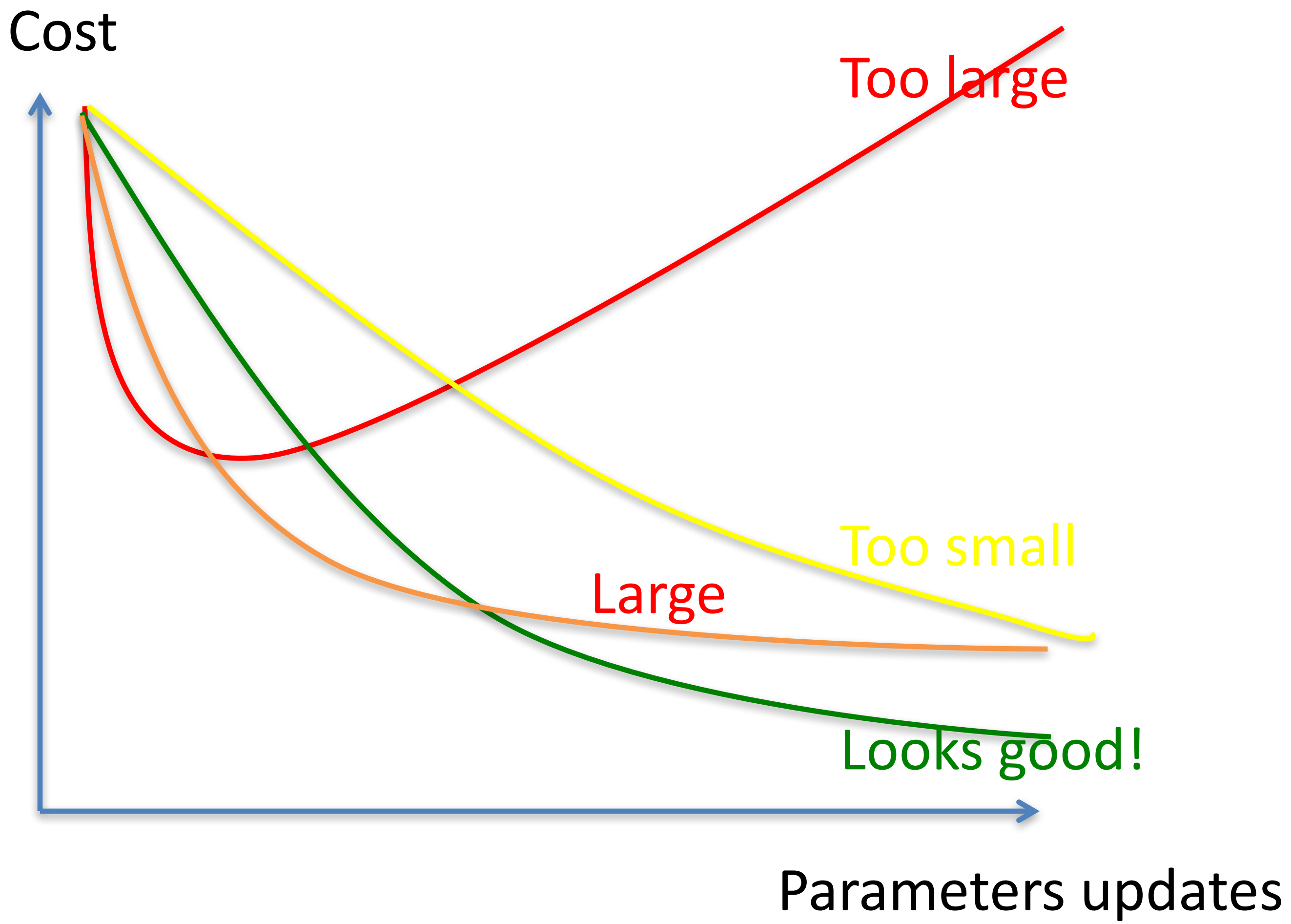
- Stochastic Gradient Descent:
  - Pick an example x^r

$$\theta_i \leftarrow \theta_{i-1} - \eta \nabla C^r(\theta_{i-1})$$

- Mini-batch Gradient Descent:
  - Pick B examples as a batch b
  - B is the batch size

$$\theta_i \leftarrow \theta_{i-1} - \eta \frac{1}{B}\sum_{x_r \in b} \nabla C^r(\theta_{i-1})$$

Learning Rate



Learning Rate

- Popular & Simple idea: Reduce the learning rate by some factor every few epochs
  - At the beginning, larger learning rate
  - After several epochs, reduce the learning rate

$$\eta^t = \eta / (t+1)$$

When to reduce the learning rate?

How much should we reduce the learning rate?

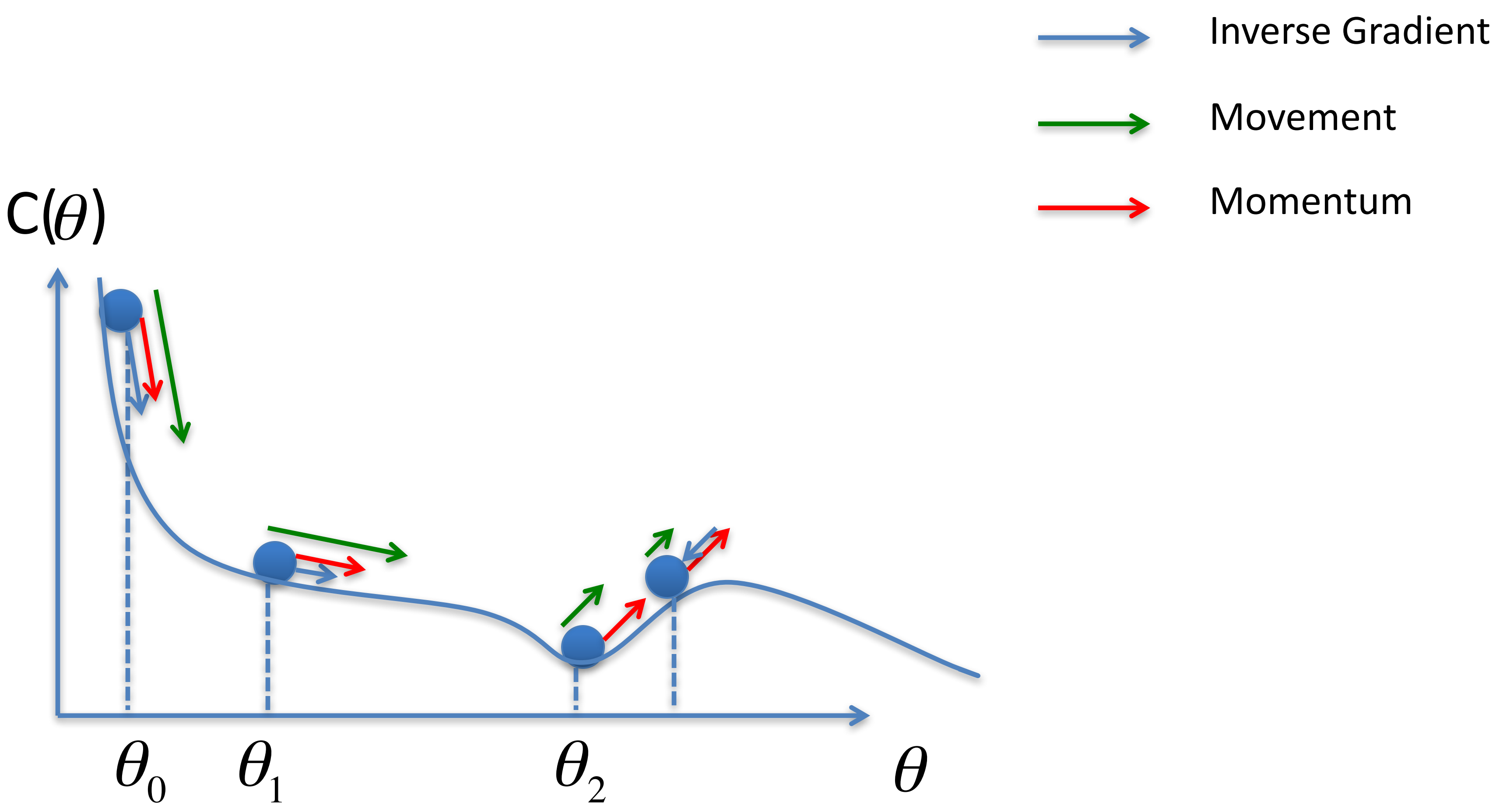
Challenges of Gradient Descent

- Choosing a proper learning rate can be difficult
- Learning rate schedules are helpful but need to be defined in advance
- The same learning rate for all parameter updates is problematic
- The problem of saddle points:
  - they are not local minima
  - the gradients there are close to zero, i.e. difficult to escape

Adaptive Learning Rate

- Adaptive learning rate
  - AdaGrad (Duchi et al. 2011)
  - AdaDelta (Zeiler 2012)
  - RmsProp (Hinton, 2012)
  - Adam (Kingma & Ba, 2014)
  - and more …

Gradient Descent with Momentum



Gradient Descent with Momentum

- Start at point θ_0
- Momentum $v_0 = 0$
- Computer the gradient at θ_0
- Momentum $v_1 = \lambda v_0 - \mu \nabla C(\theta_0)$
- Move to $\theta_1 = \theta_0 + v_1$
- Compute gradient at θ_1
- Momentum $v_2 = \lambda v_1 - \mu \nabla C(\theta_1)$
- Move to $\theta_2 = \theta_1 + v_2$
- …

AdaGrad – Duchi et al 2011

- Divide the learning rate by the average gradient

$$w^{t+1} \leftarrow w^t - \frac{\eta}{\sigma} g^t$$

σ : Average gradient of parameter w

- If w has small average gradient

➡ Larger learning rate

- If w has large average gradient

➡ Smaller learning rate

AdaGrad – Duchi et al 2011

$$w^1 \leftarrow w^0 - \frac{\eta}{\sigma^0} g^0 \qquad \sigma^0 = g^0$$

$$w^2 \leftarrow w^1 - \frac{\eta}{\sigma^1} g^1 \qquad \sigma^1 = \sqrt{\frac{1}{2}\left[(g^0)^2 + (g^1)^2\right]}$$

$$w^3 \leftarrow w^2 - \frac{\eta}{\sigma^2} g^2 \qquad \sigma^2 = \sqrt{\frac{1}{3}\left[(g^0)^2 + (g^1)^2 + (g^2)^2\right]}$$

$$\vdots$$

$$w^{t+1} \leftarrow w^t - \frac{\eta}{\sigma^t} g^t \qquad \sigma^t = \sqrt{\frac{1}{t+1}\sum_{i=0}^{t}(g^i)^2}$$

AdaGrad – Duchi et al 2011

- Divide the learning rate by the average gradient

$$w^{t+1} \leftarrow w^{t} - \frac{\eta}{\sigma^{t}} g^{t}$$

$$\sigma^{t} = \sqrt{\frac{1}{t+1} \sum_{i=0}^{t} (g^{i})^{2}}$$

$$\eta^{t} = \frac{\eta}{\sqrt{t+1}}$$

$$w^{t+1} \leftarrow w^{t} - \frac{\eta^{t}}{\sqrt{\sum_{i=0}^{t} (g^{i})^{2}}} g^{t}$$

RMSProp – Hinton et al 2012

$$w^1 \leftarrow w^0 - \frac{\eta}{\sigma^0} g^0 \qquad \sigma^0 = g^0$$

$$w^2 \leftarrow w^1 - \frac{\eta}{\sigma^1} g^1 \qquad \sigma^1 = \sqrt{\alpha(\sigma^0)^2 + (1-\alpha)(g^1)^2}$$

$$w^3 \leftarrow w^2 - \frac{\eta}{\sigma^2} g^2 \qquad \sigma^2 = \sqrt{\alpha(\sigma^1)^2 + (1-\alpha)(g^2)^2}$$

$$\vdots$$

$$w^{t+1} \leftarrow w^t - \frac{\eta}{\sigma^t} g^t \qquad \sigma^t = \sqrt{\alpha(\sigma^{t-1})^2 + (1-\alpha)(g^t)^2}$$

Adam – Kingma & Ba, 2014

- On iteration t:
  - Compute δW on current mini batch
    - $V_{\delta W} = \beta_1 V_{\delta W} + (1 - \beta_1)\delta W$
    - $S_{\delta W} = \beta_2 S_{\delta W} + (1 - \beta_2)\delta W^2$
    - $V_{\delta W}^{corrected} = {}^{V_{\delta W}}/_{(1-\beta_1^t)}$
    - $S_{\delta W}^{corrected} = {}^{S_{\delta W}}/_{(1-\beta_2^t)}$

Adam – Kingma & Ba, 2014

- On iteration t:
  - Compute δW on current mini batch
    - Derive $V_{\delta W}^{corrected}, S_{\delta W}^{corrected}$
    - Update parameters:

$$w := w - \alpha \frac{V_{\delta W}^{corrected}}{\sqrt{S_{\delta W}^{corrected}} + \varepsilon}$$

  - $\beta_1 = 0.9$; $\beta_2 = 0.999$; $\varepsilon = 10^{-8}$, α needs to be tuned

Adam – Kingma & Ba, 2014

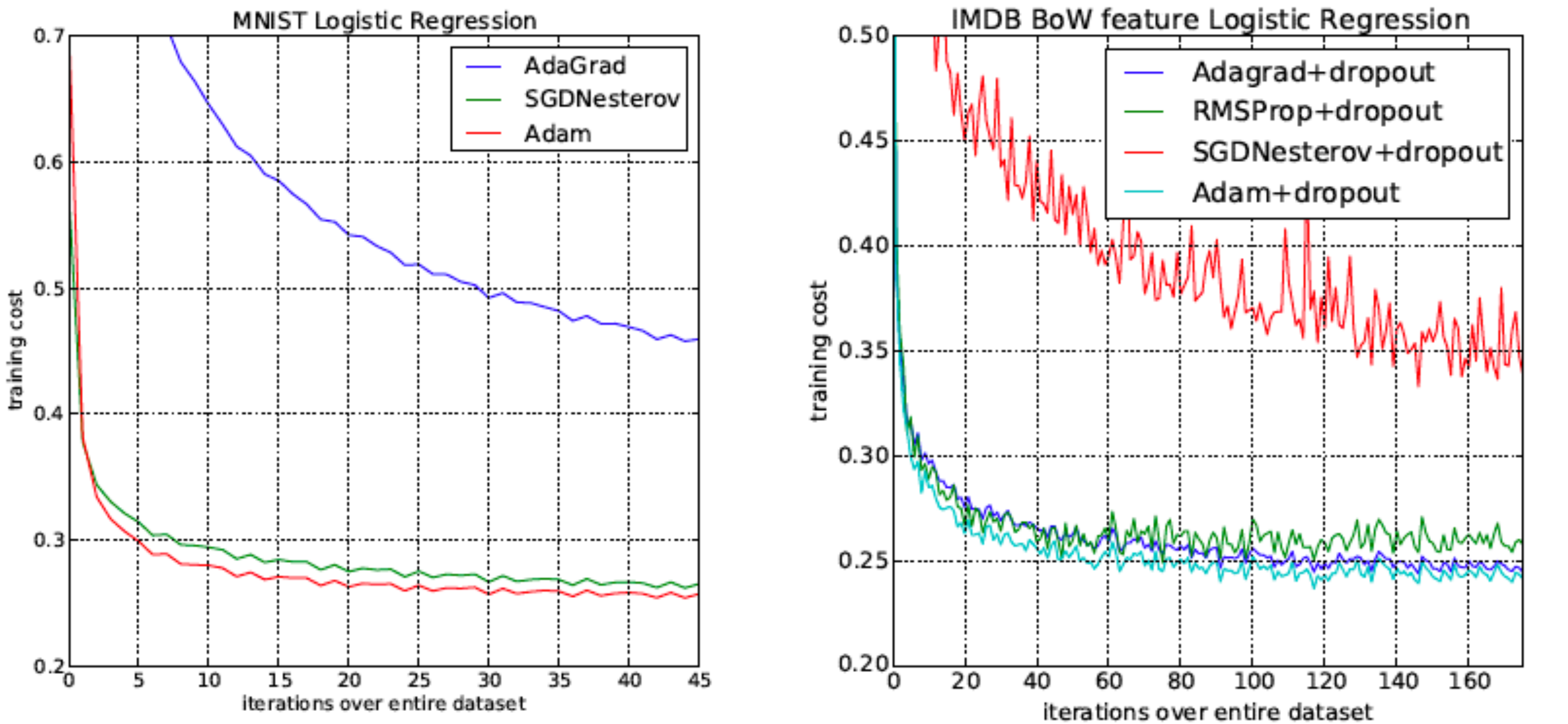


Figure 1: Logistic regression training negative log likelihood on MNIST images and IMDB movie reviews with 10,000 bag-of-words (BoW) feature vectors.

Learning Rate

- Popular & Simple idea: Reduce the learning rate by some factor every few epochs
  - At the beginning, larger learning rate
  - After several epochs, reduce the learning rate

$$\eta^t = \eta / (t+1)$$

- Adaptive learning rate:
  - AdaGrad (Duchi et al. 2011)
  - AdaDelta (Zeiler, 2012)
  - RmsProp (Hinton, 2012)
  - Adam (Kingma & Ba, 2014)
  - and more …