An Overview of Gradient Descent Optimization Algorithms

by Sebastian Ruder

Group 7

Myung-Hwan Song
Index

1. Gradient Descent Variants
2. Challenges
3. Gradient descent optimization algorithms
   - Momentum
   - Nesterov’s Accelerated Gradient
   - Adagrad
   - RMSprop
   - Adadelta
   - Adam
   - Adamax
   - Nadam
4. Which Optimizer to Use?
5. Reference
Gradient descent variants

- **Batch gradient descent**
  
  Use whole dataset to compute gradient

- **Stochastic gradient descent**
  
  Use one data sample to compute gradient

- **Mini-batch gradient descent**
  
  Use few number of samples (mini-batch) to compute gradient
Challenges

• Choosing step size (learning rate)
  
  Annealing, scheduling

• Same step size applies to all parameters
  
  Adaptive learning rate strategy

• Avoiding local minima / saddle point
  
  Avoiding saddle point is more critical!
Momentum

- Update Rule

\[ m_t = \gamma m_{t-1} + \eta \nabla \theta J(\theta) \]
\[ \theta_{t+1} = \theta_t - m_t \]

- What it does

Use momentum term (m) to compute next move
We can expect higher training speed
Nesterov’s Accelerated Gradient (NAG)

• Update Rule

\[ m_t = \gamma m_{t-1} + \eta \nabla_{\theta} J(\theta - \gamma m_{t-1}) \]
\[ \theta_{t+1} = \theta_t - m_t \]

• What it does

Compute gradient direction after moving to past momentum direction
Adaptive Learning Rate

• Recall the second challenge
  Same learning rate applies to all parameters

• When dataset is sparse this can be a problem
  Some parameters will converge very slowly

• If we apply different step size to different parameter
  -> Adaptive learning rate
Adagrad

Duchi et al., (2010)

- Update Rule

\[ G_{t,i} = G_{t-1,i} + \left( \nabla_{\theta_t} J(\theta_{t,i}) \right)^2 \]

\[ \theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,i} + \epsilon}} \nabla_{\theta_t} J(\theta_{t,i}) \]

- What it does

Moved small in the past → far away from the optimal
Moved a lot in the past → maybe near the optimal
RMSprop

Geoffery Hinton et al., 2012

- **Update Rule**

\[ G_{t,i} = \gamma G_{t-1,i} + (1 - \gamma) \left( \nabla_{\theta_t} J(\theta_{t,i}) \right)^2 \]

\[ \theta_{t+1,i} = \theta_{t,i} - \eta \frac{\nabla_{\theta_t} J(\theta_{t,i})}{\sqrt{G_{t,i} + \epsilon}} \]

- **What it does**

Moving average of squared gradient descent

Therefore, G can get smaller!
Adadelta

Matthew D. Zeiler (2012)

• Update Rule

\[ G_{t,i} = \gamma G_{t-1,i} + (1 - \gamma) \left( \nabla_{\theta_t} J(\theta_{t,i}) \right)^2 \]  \hspace{1cm} \text{Moving average of squared gradient}

\[ s_t = \gamma s_{t-1} + (1 - \gamma) \Delta_{\theta_t}^2 \]  \hspace{1cm} \text{Moving average of squared delta}

\[ \Delta_{\theta_t} = -\frac{\sqrt{s_{t-1} + \epsilon}}{\sqrt{G_{t,ii} + \epsilon}} \nabla_{\theta_t} J(\theta_{t,i}) \]  \hspace{1cm} \text{Delta decides how much you update the parameter}

\[ \theta_{t+1} = \theta_t + \Delta_{\theta_t} \]
Adadelta – approximate Hessian!

Matthew D. Zeiler (2012)

- **Why using squared delta?**
  - In Newton Method...
    
    $$(g = \nabla j, H = \nabla^2 j)$$  
    
    Assume $J$ has no unit

  \[ \Delta_\theta \propto H^{-1} g \propto \frac{\partial J}{\partial \theta} \propto \partial \theta \propto \text{unit}(\theta) \]  

- **However, in Gradient Method**...

  \[ \Delta_\theta \propto g \propto \frac{\partial J}{\partial \theta} \propto \frac{1}{\partial \theta} \propto \frac{1}{\text{unit}(\theta)} \]  

- **Therefore**...

  \[ \Delta_\theta = \frac{\partial J}{\partial \theta^2} \Rightarrow H^{-1} = \frac{1}{\partial^2 J/\partial \theta^2} = \frac{\Delta_\theta}{\partial J/\partial \theta} \]

  \[ \theta_{t+1} = \theta_t - \frac{\sqrt{s_{t-1} + \epsilon}}{\sqrt{G_{t,ii} + \epsilon}} \nabla_{\theta} J(\theta_{t,i}) \]
Adam

Kingma & Ba, 2015

- **Update Rule**

\[ g_t = \nabla_\theta J(\theta) \]

\[ m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad \text{Momentum Term} \]

\[ v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad \text{RMSprop Term} \]

\[ \hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad \text{Bias correction Term} \]

\[ \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \]

\[ \theta_t = \theta_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \]
Adamax

Kingma & Ba, 2015

- **Update Rule**

\[
g_t = \nabla_\theta J(\theta)
\]

\[
m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t
\]

\[
u_t = \max(\beta_2 u_{t-1}, |g_t|)
\]

\[
\hat{m}_t = \frac{m_t}{1 - \beta_1}
\]

\[
\theta_t = \theta_{t-1} - \eta \frac{\hat{m}_t}{u_t}
\]

Max norm of gradient

No need to correct bias
$$m_t = \gamma m_{t-1} + \eta \nabla \theta J(\theta_t)$$

$$\theta_{t+1} = \theta_t - (\gamma m_t + \eta \nabla \theta J(\theta_t))$$

- Modified NAG

- What it does

Use current momentum and current gradient to decide direction
Nadam
Timothy Dozat (2016)

\[ \theta_t = \theta_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}} = \theta_{t-1} - \eta \frac{1}{\sqrt{\hat{v}_t + \epsilon}} \left( \beta_1 \hat{m}_{t-1} + \frac{(1 - \beta_1) \nabla \theta J(\theta_t)}{1 - \beta_1^t} \right) \]

Change momentum term with modified NAG

\[ \theta_t = \theta_{t-1} - \eta \frac{1}{\sqrt{\hat{v}_t + \epsilon}} \left( \beta_1 \hat{m}_t + \frac{(1 - \beta_1) \nabla \theta J(\theta_t)}{1 - \beta_1^t} \right) \]
Which Optimizer to Use?

- Not surprisingly, no exact answer
- But generally…
  - Momentum strategy -> high training speed
  - If dataset is sparse -> adaptive learning rate
- Adam?
  - Kind of default algorithm to use, however…

https://imgur.com/a/Hqolp
Reference


- Timothy Dozat, Incorporating Nesterov Momentum Into Adam, ICLR 2016


- http://cs231n.github.io/optimization-1/
Thank you!