Gradient Descent Algorithms

Gradient Descent

- Batch gradient descent
  - Makes a parameter update i.e. calculates gradient of cost function, using the entire training data set
  - No online update is possible
- Stochastic gradient descent
  - Parameter updates are done for every training sample
  - Online learning is possible
  - Causes large oscillations in objective function due to frequent updates
- Mini-batch gradient descent
  - Combination of the above
  - Performs update for a mini-batch of training data

$$\boxed{w = w - \alpha \frac{\partial J}{\partial w}}$$

Algorithms

- Momentum
- NAG- Nesterov Accelerated Gradient
- Adagrad
- RMS-Prop

Momentum

- Adds a fraction of the previous update to the current update
- Helps to take larger steps in the relevant direction, preventing oscillations that are common in vanilla SGD

$\gamma = 0.9$

$$\delta w_n = \gamma . \delta w_{n-1} + \alpha \nabla_w J(w)$$

$$w_n = w_{n-1} + \delta w_n$$

Nesterov Accelerated Gradient (NAG)

- We can compute the updated parameter value using the current gradient and treat it like a look ahead
- Evaluate gradient at the new parameter values and then perform an update to the parameters.

$w_n = w_{n-1} + \alpha \nabla J$

$$\delta w_n = \gamma . \delta w_{n-1} + \alpha \nabla_w J(w_{n-1} + \delta w_n)$$

$$w_n = w_{n-1} + \delta w_n$$

Adagrad

- Use a different learning rate for different parameters
- Compute update for every parameter using gradient
- Scale general learning rate at a particular iteration using accumulated squared gradients from previous iterations
- Leads to rapidly diminishing updates

$i \rightarrow$ parameter index

$g_{n,i} \rightarrow \nabla_{w_i}(J)$

timestep n

$$w_{n,i} = w_{n-1,i} + \frac{\alpha}{\sqrt{G_{n,i} + \varepsilon}} g_{n,i}$$

$$G_{n,i} = G_{3,i} = \frac{g_{1,i}^2 + g_{2,i}^2}{2} , \quad G_{4,i} = \frac{g_{1,i}^2 + g_{2,i}^2 + g_{3,i}^2}{3}$$

Adadelta

- Extension of Adagrad
- Instead of storing running sum of squared gradients, use weighted average of squared gradients
  - i.e. the current average depends on the average in the previous iteration and the square of the gradient in the current iteration

$w_n = w_{n-1} + \delta w_n$

$\sqrt{E[\delta w^2]_n}$

$$\delta w_{n,i} = \frac{\alpha}{\sqrt{E[g^2]_n + \varepsilon}} g_{n,i}$$

$g_n \rightarrow \nabla_w J$

w_i

$\gamma = 0.9$

$$E[g^2]_n = \gamma . E[g^2]_{n-1} + (1 - \gamma) g_n^2$$