Introduction to Deep Learning

Optimization Methods

Optimization Methods for Training a Deep Neural Network

- Slow convergence of gradient descent method
- Problem addressed: How to reduce the number of epochs taken to reach a local minimum?
- Weight update methods that use the past history of updates have been shown to be effective.
- Generalized delta rule that uses momentum factor
- Weight-specific learning rate scheduling methods (Adaptive learning rate methods)
  - AdaGrad
  - RMSProp
  - AdaDelta
  - AdaM
- Second-order methods for optimization

Gradient Descent Method

- Pattern Mode :
  - Stochastic Gradient Descent Method
  - Weights are updated after the presentation of each example.

$$\Delta w(m) = -\eta \frac{\partial \tilde{E}(m)}{\partial w}$$

  - Epoch: Presentation of all the examples once.
- Batch Mode:
  - Weights are updated after the presentation of all the examples once.

$$\Delta w(m) = -\eta \frac{\partial E_{av}(m)}{\partial w}$$

$$\text{where } E_{av} = \frac{1}{N} \sum_{n=1}^{N} \tilde{E}_n$$

Weight Update Rules

- **Weight update at step m:**

$$w_{m+1} = w_m + \Delta w(m)$$

- **Delta Rule:**

$$\Delta w(m) = -\eta \frac{\partial E}{\partial w(m)}$$

η **is the learning rate parameter in the range 0.0 to 1.0**

E **is the error function**

- **Generalized Delta Rule:**

$$\Delta w(m) = -\eta \frac{\partial E}{\partial w(m)} + \alpha \Delta w(m-1)$$

α **is the momentum factor in the range 0.0 to 1.0**

Generalized Delta Rule

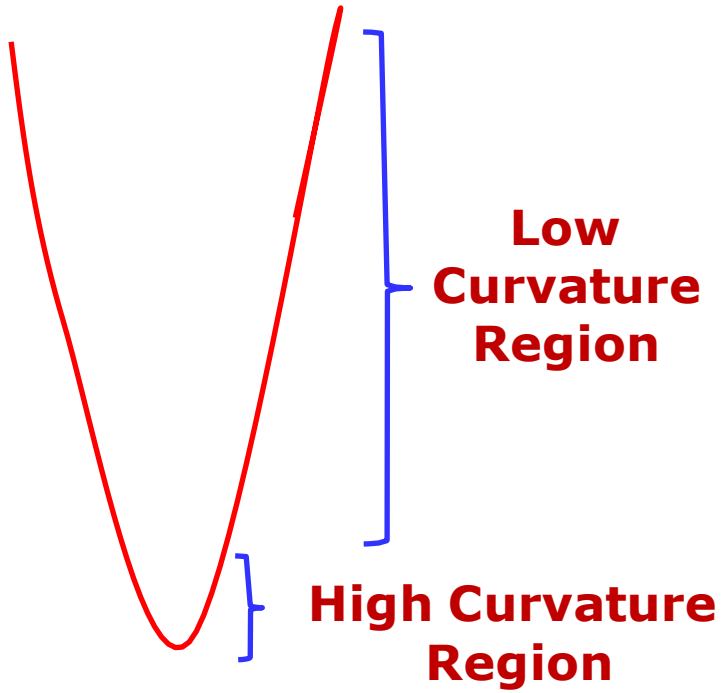
$$\Delta w(m) = -\eta \frac{\partial E}{\partial w(m)} + \alpha \Delta w(m-1)$$

$$\Delta w(m) = -\eta g(m) + \alpha \Delta w(m-1)$$

$$w(1) = w(0) - \eta g(0)$$

$$w(2) = w(1) - \eta g(1) + \alpha \Delta w(0)$$

$$w(3) = w(2) - \eta g(2) + \alpha \Delta w(1)$$



$$w(m+1) = w(m) - \eta g(m) - \eta\, \alpha g(m-1) - \eta\, \alpha^2 g(m-1) - \ldots - \eta\, \alpha^m g(0)$$

Low curvature region of error surface: Change in gradient is very small.

$$w(m+1) = w(m) - \eta(1 + \alpha + \alpha^2 + \cdots + \alpha^m)g$$

$$= w(m) - \frac{\eta}{1-\alpha} g$$

$$= w(m) - \hat{\eta} g$$

η **is the effective learning rate. For** $\alpha = 0.9$, $\hat{\eta} = 10\eta$

Adaptive Learning Rate Methods

AdaGrad:

$$\Delta w(m) = -\frac{\eta}{\varepsilon + r(m)} g(m)$$

$$r^2(m) = g^2(0) + g^2(1) + g^2(2) + \ldots + g^2(m-1)$$

ε is a small positive constant

RMSProp:

$$\Delta w(m) = -\frac{\eta}{\varepsilon + r(m)} g(m)$$

$$r^2(m) = \rho\left(\frac{1}{L}\sum_{l=1}^{L} g^2(m-l)\right) + (1-\rho)g^2(m)$$

Moving average of past gradients

ρ is the weightage given to the average of the past L gradients

Adaptive Learning Rate Methods

AdaDelta:

$$\Delta w(m) = -\frac{p(m)}{\varepsilon + r(m)} g(m)$$

$$p^2(m) = \gamma \left(\frac{1}{L} \sum_{l=1}^{L} \Delta w^2(m-l-1) \right) + (1-\gamma)\Delta w^2(m-1)$$

Moving average of past changes in weight

γ is the weightage given to the average of the past L changes in weight

Learning rate is not used in the AdaDelta method as it is implicitly present in the past change in the weight.

Adaptive Learning Rate Methods

Adaptive Moments (AdaM):

- It uses the first moment and the second moment of gradient
- More effective than Generalized Delta rule, AdaGrad, RMSProp and AdaDelta

$$\Delta w(m) = -\eta \frac{q(m)}{\varepsilon + s(m)} g(m)$$

$$q(m) = (1 - \rho_1) q(m-1) + \rho_1 g(m)$$

$$s(m) = (1 - \rho_2) s(m-1) + \rho_2 g^2(m)$$

ρ_1 is the weightage given to the first moment of the current gradient

ρ_2 is the weightage given to the second moment of the current gradient

$q(m)$: Exponentially weighted moving average of the first moment

$s(m)$: Exponentially weighted moving average of the second moment

Second Order Methods

The second order methods make use of the second order derivatives of error function with weight parameters

Newton's method of approximation:

Let w be the vector of all weight parameters in the model

Let g be the vector of the first derivative of the error function with all weight parameters in the model

Let $\boldsymbol{H}$ be the Hessian matrix. The element h_{ij} of $\boldsymbol{H}$ is the second derivative of the error function E with ith and jth weight parameters.

Second order approximation of error function:

$$E(w + \Delta w) = E(w) + g^{t}\Delta w + \Delta w^{t} H \Delta w$$

Optimal change in w is given by

$$\Delta w^{*} = -H^{-1} g$$

Second Order Methods

- **Levenberg-Marquardt (LM) Method**
  - Suitable for batch mode of learning only
  - It approximates the Hessian matrix by an outer- product matrix, and reduces the computational complexity

- **Broyden-Fletcher-Goldfarb-Shanno (BFGS) Method**
  - It approximates the inverse of Hessian matrix

- **Conjugate-Gradient Method**