Hyomin Seo

Title : Compare and Contrast on Different Types of Gradient Descent Algo

Explain the key difference between Adam and the basic gradient descent algorithm. Prose or mathematics are equally acceptable.

Adam

Adaptive Moment Estimation

Efficient stochastic optimization that only requires first order gradient with relatively smaller memory requirement

Uses Momentum and Adaptive Learning Rates to converge faster

Momentum

Algorithm that converges faster to local minimum with addition of temporal element to update parameters of a neural network, the element of time (acceleration the gradient of the past step to determine the direction to go)

Adaptive Learning Rate

The rate of converge (steps) here is proportional to the learning rate.The formula below shows that the the bigger the division is, the smaller the learning rate η is.

$$\theta_{4,i} = \theta_{3,q} - \frac{\eta}{\sqrt{\varepsilon+g(\theta_{1,i})^2+g(\theta_{2,i})^2+g(\theta_{3,i})^2}}(delta)J(\theta_{3,i})$$

Update rule of Adam (with two decay terms, β_1, β_2)

$\theta_{t+1} = \theta_t - \frac{\eta \cdot \widehat{m}_t}{\sqrt{\widehat{v}_t + \varepsilon}}$, *where* $\widehat{m_t} = \frac{m_t}{1-\beta_1^t}$, $\widehat{v_t} = \frac{m_t}{1=\beta_2^t}$

and where $m_t = (1-\beta_1)g_t + \beta_1 m_t - 1, v_t = (1-\beta_2)g_t^2 + \beta_2 v_t - 1$

Basic gradient descent algorithm : first order optimization algorithm.

Each iteration, the system updates the parameters in the opposite direction of the gradient that gives the steepest ascent to the target (local minimum). The simplicity of the algorithm is cin exchange with relatively slow process rate and bigger memory requirement.

Update rule of typical gradient descent algorithm

$$\theta_j = \theta_j - \eta \cdot \frac{\partial C}{\partial \theta_j} \text{ repeat until convergence } (\text{ reach local minimum}),$$

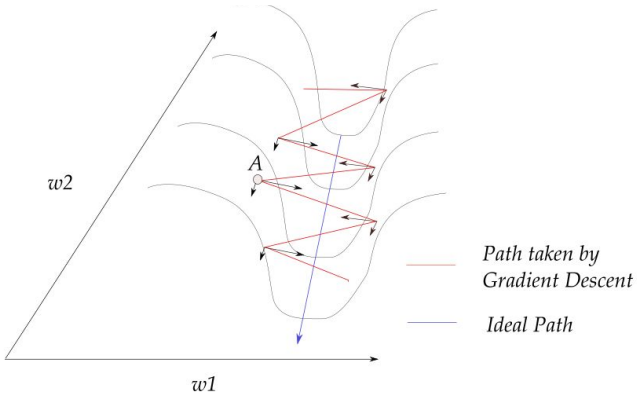
How does momentum actually work

When typical gradient descent meets a pathological curvature (where the region of the function is not scared properly), it is most likely to be delayed greatly. Momentum added the 'acceleration' to gradient descent in such a scenario.

$$'Acceleration';\ =$$

$$z^{k+1} = \beta z^k + (gradient)\, f(w^k)$$

$$w^{k+1} = w^k - \alpha z^{k+1}$$



Momentum update

With this addition, the previous gradient does remain in subsequent updates but the weight of the most recent precious gradient is the biggest. Which decreases the magnitude of oscillation quickly.