Алгоритмы оптимизации

Основы Deep Learning

Gradient Descent

$$Loss(\hat{y}, y) = \frac{1}{2n}\sum_{i=1}^{n}(\hat{y_i} - y_i)^2 = \frac{1}{2n}\sum_{i=1}^{n}(\sigma(w \cdot X_i) - y_i)^2$$

$$\frac{\partial Loss}{\partial w} = \frac{1}{n}X^T(\sigma(w \cdot X) - y)\sigma(w \cdot X)(1 - \sigma(w \cdot X))$$

Mini-batch Gradient Descent

$$Loss(\hat{y}, y) = \frac{1}{2 \cdot \text{batch\_size}} \sum_{i=1}^{\text{batch\_size}} (\hat{y_i} - y_i)^2 = \frac{1}{2 \cdot \text{batch\_size}} \sum_{i=1}^{\text{batch\_size}} (\sigma(w \cdot X_i) - y_i)^2$$

$$\frac{\partial Loss}{\partial w} = \frac{1}{\text{batch\_size}} X_{batch}^T (\sigma(w \cdot X_{batch}) - y) \sigma(w \cdot X_{batch})(1 - \sigma(w \cdot X_{batch}))$$

Momentum

Обозначим: $J = Loss$, $\theta = w$

Тогда: $\frac{\partial Loss}{\partial w} = \nabla_\theta J$

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta - \gamma v_{t-1})$$

$$\theta = \theta - v_t$$

https://habr.com/post/318970/

Adagrad (adaptive gradient)

$$g_t \equiv \nabla_\theta J(\theta_t)$$

$$G_t = G_t + g_t^2$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} g_t$$

http://jmlr.org/papers/volume12/duchi11a/duchi11a.pdf

https://arxiv.org/abs/1002.4908

RMSProp (root mean square propagation)

$$g_t \equiv \nabla_\theta J(\theta_t)$$

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1-\gamma) g_t^2$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t$$

$$RMS[g]_t = \sqrt{E[g^2]_t + \epsilon}$$

https://www.youtube.com/watch?v=76lj_cKBvmg

Adam (adaptive moments estimation)

$$g_t \equiv \nabla_\theta J(\theta_t)$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \qquad v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \; \hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

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

https://arxiv.org/abs/1412.6980