AIST4010: Foundation of Applied Deep Learning
Lecture 9 Scribing: Optimization

Lecturer: Professor Yu LI Scriber: Man Ho LAM (1155159171)

17 February 2024

1 Introductory

1.1 Terminologies

- **Epoch:** the time that we iterative through all the training data.
- **Batch Size:** the number of data points that we feed to the algorithm every iteration.

1.2 Gradient Descent

Given the steps of gradient descent algorithm,

1: Initialize the weights.
2: For all the training data: forward pass and calculate the output(s) Y.
3: Update the weights (gradient descent):
- $\Delta w_{ij}^{(k)} = \alpha \frac{\partial Div(Y,d)}{\partial w_{ij}^{(k)}}$
- $w_{ij}^{(k)} = w_{ij}^{(k)} - \Delta w_{ij}^{(k)}$.

4: Repeat from step 1.

1.3 Batch Gradient Descent

We can use GPU to accelerate the computations, but we cannot fit all the data into the GPU memory as the GPU is small. Therefore, we consider feeding a batch of data each time.

- **Min-Batch Gradient Descent:** This algorithm uses a batch of samples ($|\beta|$ samples) to update the weights every iteration.
- **Stochastic Gradient Descent:** This algorithm uses 1 sample to update the weights every iteration.

However, if the model is too large while the batch is too small, the training can be very unstable.
Any solutions can make the convergence more stable?

1.4 Accumulate Gradient

We can accumulate the gradient for like 10 iterations to update the weights.

for $t = 1 : 10$ **do**
- $\Delta w_{ij}^{(k)} \mathrel{+}= \alpha \frac{\partial Div(Y_{B_t}, d_{B_t})}{\partial w_{ij}^{(k)}}$
- $w_{ij}^{(k)} = w_{ij}^{(k)} - \Delta w_{ij}^{(k)}$.

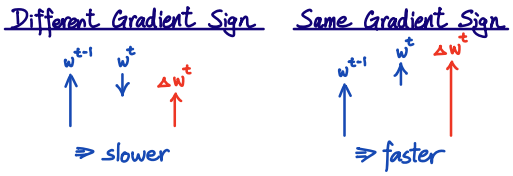
Any solutions can make the convergence faster and better?

2 Momentum Optimization

2.1 Momentum

Learning Rate: It is hard to set the learning rate, if too small will take more steps to converge, while too large may lead to oscillation which slows down the convergence. Momentum extension focuses on improving the selection of the learning rate.

Idea: By remembering the update of the previous step, the learning rate α is affected by the sign of both previous and current gradients. If they have the same gradient sign, the update step is longer in directions; otherwise, the update step is shorter in directions.



Algorithm

for each iteration, **do**

1. Compute the gradient at the current location
2. Add the scaled previous step

$$w^t = w^{t-1} - \alpha \frac{\partial Loss}{\partial w^{t-1}} + \beta \Delta w^{t-1}$$

- $\Delta w^{t-1} = w^{t-1} - w^{t-2}$
- Typical $\alpha = 0.001$, $\beta = 0.9$

Why momentum can conduct a smoother and faster convergence?

It actually maintains a moving average of the gradient, which can amplify the learning rate in the correct direction, which smoothes the updates and enlarges the learning rate if the gradient signs are the same.

2.2 Nestorov's Momentum

The difference between Momentum and Nesterov's Momentum is in the gradient computation phase. The Nesterov's Momentum will take an extension of the previous step first, then compute the gradient at the location (after extension).

for each iteration, **do**

1. Extend the previous step
2. Compute the gradient at the current location
3. Sum them up

$$w^t = w^{t-1} - \alpha \frac{\partial Loss}{\partial (w^{t-1} + \beta \Delta w^{t-1})} + \beta \Delta w^{t-1}$$

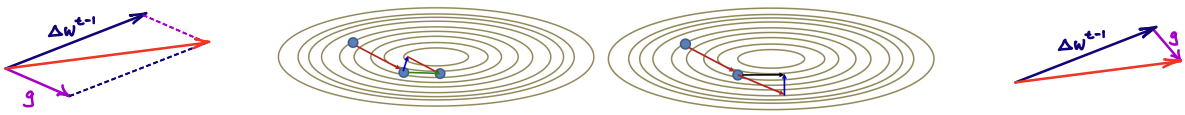


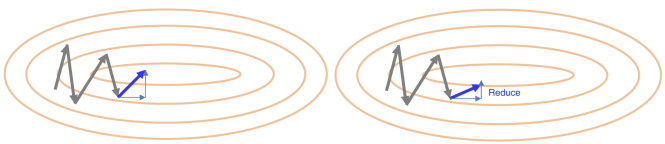
Figure 1: Momentum Method (Left), Nestorov's Momentum (Right)

Any further improvement?

Observation: The steps in the oscillatory direction still show a large total movement.

Improvement: We want to reduce the step size in directions with high motion and increase it in directions with slow motion.

Methodology: We can scale the updates in every component in inverse proportion to the total movement of that component in recent past.



3 Scaling

3.1 Inverse mean square derivative

Recall the update of weights: $w^t = w^{t-1} - \alpha \frac{\partial Loss}{\partial w^{t-1}}$.
Let $\partial_{w^{t-1}} L = \frac{\partial Loss}{\partial w^{t-1}}$ be the derivative of loss with respect to the parameter w^{t-1} (i.e., the gradient).

- $(\partial_{w^{t-1}} L)^2$ denotes the squared derivative, not the second derivative.
- $E[(\partial_w L)^2]$ denotes the expectation of the square derivative.
- $E[(\partial_w L)^2]_{t-1}$ denotes the expectation of the square derivative until step $t-1$.

Now the step is scaled by the inverse mean square derivative,

$$w^t = w^{t-1} - \frac{\alpha}{E[(\partial_w L)^2]_{t-1}} \cdot \partial w^{t-1} L$$

Since the number of iterations may be very large, it is expensive to store all previous squared derivative. Instead of accurately compute the expectation of the square derivative until the step, we apply a running average to estimate it,

$$E[(\partial_w L)^2]_t = \begin{cases} \gamma E[(\partial_w L)^2]_{t-1} + (1-\gamma)(\partial_{w^t} L)^2 & \text{if } t \neq 0 \\ 0 & \text{if } t = 0 \end{cases}$$

To prevent the divide by 0 problem, we add ε to $E[(\partial_w L)^2]_{t-1}$,

$$w^t = w^{t-1} - \frac{\alpha}{E[(\partial_w L)^2]_{t-1} + \varepsilon} \cdot \partial w^{t-1} L$$

3.2 Root Mean Squared Propagation (RMS Prop)

$$w^t = w^{t-1} - \frac{\alpha}{\sqrt{E[(\partial_w L)^2]_{t-1} + \varepsilon}} \cdot \partial w^{t-1} L$$

- $E[(\partial_w L)^2]_{t-1} = \gamma E[(\partial_w L)^2]_{t-2} + (1-\gamma)(\partial_{w^{t-1}} L)^2$
- $E[(\partial_w L)^2]_0 = 0$
- Typical $\gamma = 0.9$, $\alpha = 0.001$, $\varepsilon = 10^{-7}$

3.3 Adaptive moment estimation

3.3.1 RMS Prop + Momentum

Recall the update of momentum method: $w^t = w^{t-1} - \alpha \frac{\partial Loss}{\partial w^{t-1}} + \beta \Delta w^{t-1}$, where $\Delta w^{t-1} = w^{t-1} - w^{t-2}$.
We can determine w^t as following,

$$\begin{aligned} w^t &= w^{t-1} - \alpha \frac{\partial Loss}{\partial w^{t-1}} + \beta \Delta w^{t-1} \\ &= w^{t-1} - \alpha \frac{\partial Loss}{\partial w^{t-1}} + \beta (w^{t-1} - w^{t-2}) \\ &= w^{t-1} - \alpha \big(\frac{\partial Loss}{\partial w^{t-1}} - \frac{\beta}{\alpha} (w^{t-1} - w^{t-2}) \big). \end{aligned}$$

We find the recursive structure,

$$\begin{aligned} w^t - w^{t-1} &= -\alpha \frac{\partial Loss}{\partial w^{t-1}} + \beta (w^{t-1} - w^{t-2}) \\ w^{t-1} - w^{t-2} &= -\alpha \frac{\partial Loss}{\partial w^{t-2}} + \beta (w^{t-2} - w^{t-3}) \\ &\dots \\ w^2 - w^1 &= -\alpha \frac{\partial Loss}{\partial w^1} + \beta (w^1 - w^0). \end{aligned}$$

We can determine the w^t by apply the recursive structure,

$$\begin{aligned} w^t &= w^{t-1} - \alpha \Big(\frac{\partial Loss}{\partial w^{t-1}} - \frac{\beta}{\alpha} \big(-\alpha \frac{\partial Loss}{\partial w^{t-2}} + \beta (w^{t-2} - w^{t-3}) \big) \Big) \\ &= w^{t-1} - \alpha \big(\frac{\partial Loss}{\partial w^{t-1}} + \beta \frac{\partial Loss}{\partial w^{t-2}} - \frac{\beta^2}{\alpha} (w^{t-2} - w^{t-3}) \big) \\ &\dots \\ &= w^{t-1} - \alpha \big(\frac{\partial Loss}{\partial w^{t-1}} + \beta \frac{\partial Loss}{\partial w^{t-2}} + \dots + \beta^{t-1} \frac{\partial Loss}{\partial w^0} + 0 \big). \end{aligned}$$

Note that the terms $\frac{\partial Loss}{\partial w^{t-1}} + \beta \frac{\partial Loss}{\partial w^{t-2}} + \dots + \beta^{t-1} \frac{\partial Loss}{\partial w^0} + 0$ related to ALL previous derivatives, which is the running average.
Therefore, we can unify the expression,

$$w^t = w^{t-1} - \frac{\alpha}{\sqrt{E[(\partial_w L)^2]_{t-1} + \varepsilon}} \cdot E[\partial_w L]_{t-1}$$

- $E[(\partial_w L)^2]_{t-1} = \gamma E[(\partial_w L)^2]_{t-2} + (1-\gamma)(\partial_{w^{t-1}} L)^2$.
- $E[\partial_w L]_{t-1} = \delta E[\partial_w L]_{t-2} + (1-\delta)\partial_{w^{t-1}} L$, which is similar to the RMS Prop.
- The boundary cases: $E[(\partial_w L)^2]_0 = 0$ and $E[\partial_w L]_0 = 0$.

3.3.2 The expectation of the running average

Let m_t be the running average until step t.
Let g_t be the value of each observation at step t.

$$\begin{aligned} m_t &= \delta m_{t-1} + (1-\delta) g_t \\ &= \delta^2 m_{t-2} + \delta(1-\delta) g_{t-1} + (1-\delta) g_t \\ &\cdots \\ &= (1-\delta)\delta^{t-1} g_1 + \cdots + (1-\delta)\delta g_{t-1} + (1-\delta) g_t \\ &= (1-\delta) \sum_{i=0}^{t} \delta^{t-i} g_i. \end{aligned}$$

Take the expectation on both sides,

$$E[m_t] = E[(1-\delta) \sum_{i=0}^{t} \delta^{t-i} g_i]$$

We believe that the observation of g_i is around g_t, so we let $g_i = g_t + \varsigma$,

$$\begin{aligned} E[m_t] &= E[(1-\delta) \sum_{i=0}^{t} \delta^{t-i} (g_t + \varsigma)] \\ E[m_t] &= E[g_t] \cdot (1-\delta) \sum_{i=0}^{t} \delta^{t-i} + \varsigma' \\ E[m_t] &= E[g_t] \cdot (1-\delta^t) + \varsigma' \end{aligned}$$

If the running average is good, $E[m_t] = E[g_t]$.

3.3.3 Running average correction

Since running average is not an unbiased estimator, we need to add a correction term.

$$\hat{E}[(\partial_w L)^2]_{t-1} = \frac{E[(\partial_w L)^2]_{t-1}}{1-\gamma^{t-1}}, \hat{E}[\partial_w L]_{t-1} \frac{E[\partial_w L]_{t-1}}{1-\delta^{t-1}}.$$

If t is small, $1-\gamma^{t-1}$ is small, which means not correlate.

3.3.4 Adam formula

$$w^t = w^{t-1} - \frac{\alpha}{\sqrt{\hat{E}[(\partial_w L)^2]_{t-1} + \varepsilon}} \cdot \hat{E}[\partial_w L]_{t-1}$$

- $\hat{E}[(\partial_w L)^2]_{t-1} = \frac{E[(\partial_w L)^2]_{t-1}}{1-\gamma^{t-1}}$ where $E[(\partial_w L)^2]_{t-1} = \gamma E[(\partial_w L)^2]_{t-2} + (1-\gamma)(\partial_{w^{t-1}} L)^2$.
- $\hat{E}[\partial_w L]_{t-1} \frac{E[\partial_w L]_{t-1}}{1-\delta^{t-1}}$ where $E[\partial_w L]_{t-1} = \delta E[\partial_w L]_{t-2} + (1-\delta)\partial_{w^{t-1}} L$.
- The boundary cases: $E[(\partial_w L)^2]_0 = 0$ and $E[\partial_w L]_0 = 0$.
- Typical $\alpha = 0.001$, $\gamma = 0.999$, $\delta = 0.9$, $\varepsilon = 10^{-7}$