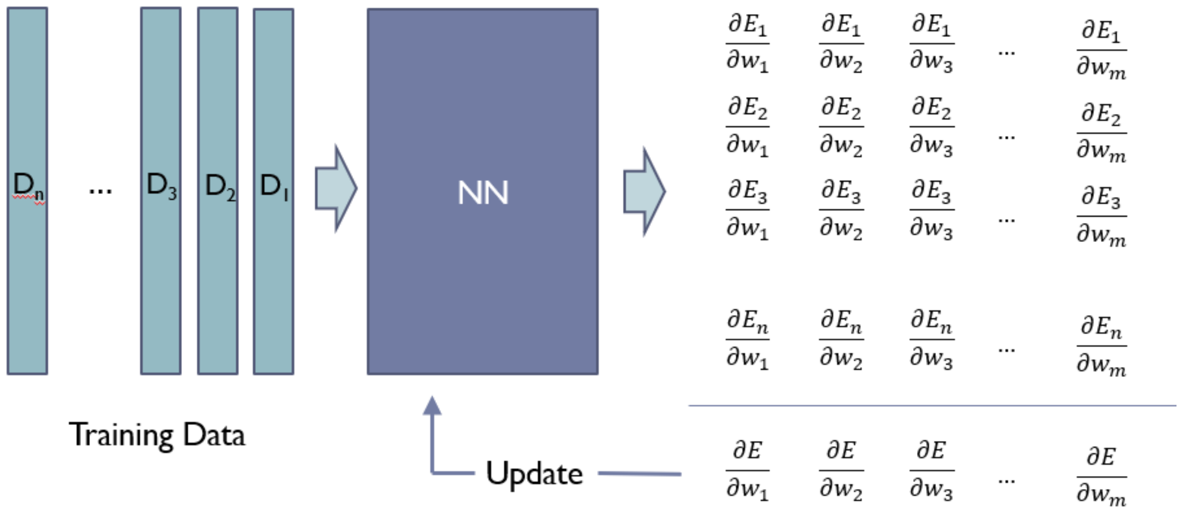
Gradient Descent Optimizer

Gradient Descent Method

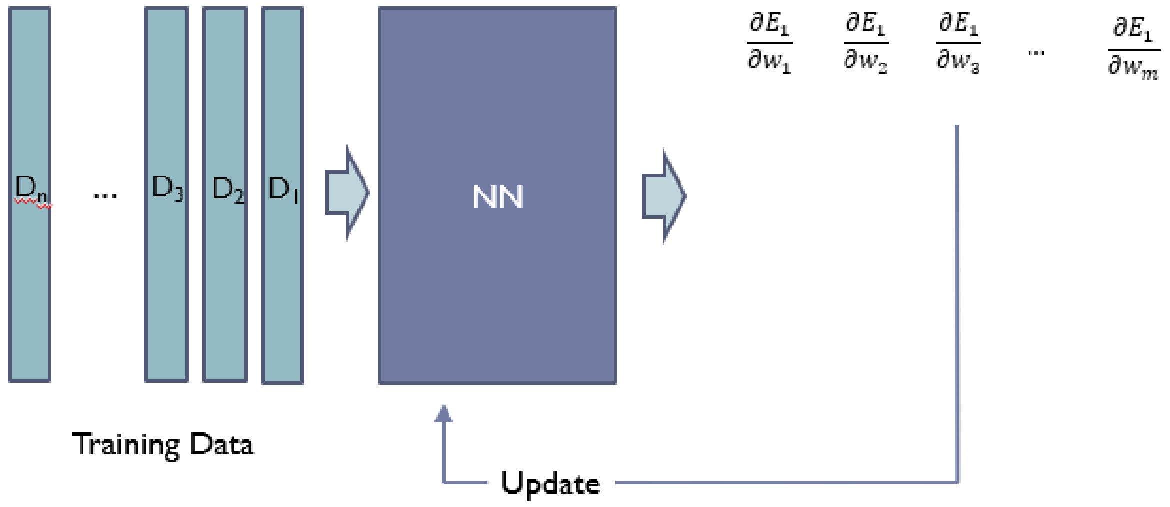
>Error Back Propagation

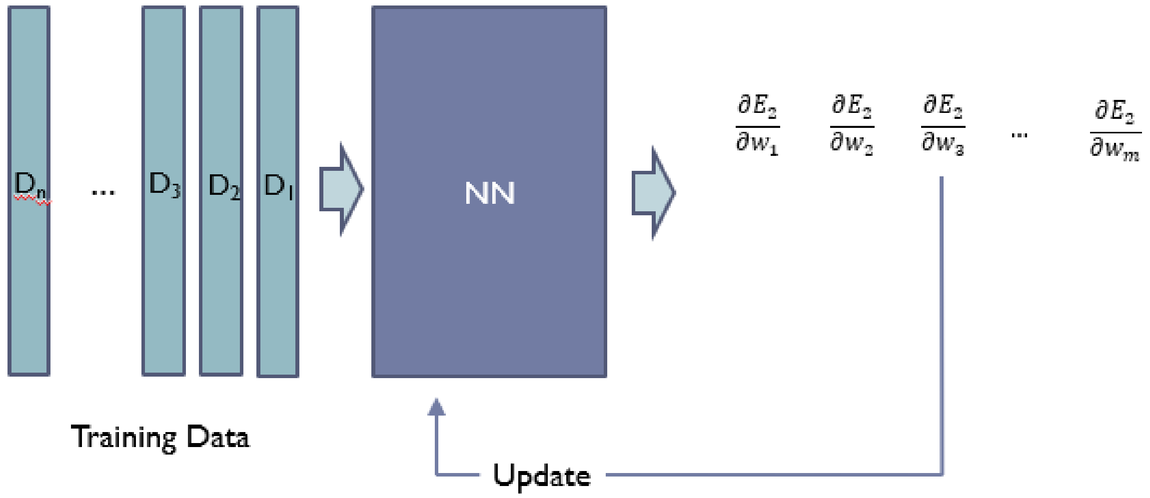
Batch Gradient Descent



- For one update, gradients are calculated for the whole dataset
- Batch gradient descent is guaranteed to converge to a local minimum
- Redundant computations for large redundant dataset

Stochastic Gradient Descent

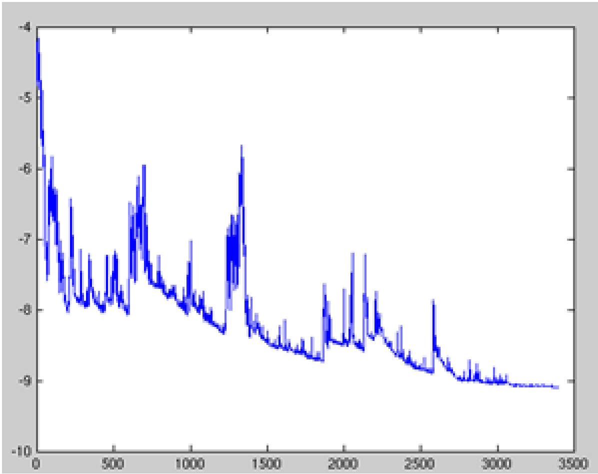




- For one update, gradients are calculated for one sample
- Usually faster
- Fluctuations : Maybe good or maybe bad
- With a small rate,show similar performance

```
Repeat
    for n =1 to N (for all training data)


     end
Until end condition satisfied
```



특징

Usual Batch Size

>Dependent on datasets

Advantage

>Good estimation of real gradient

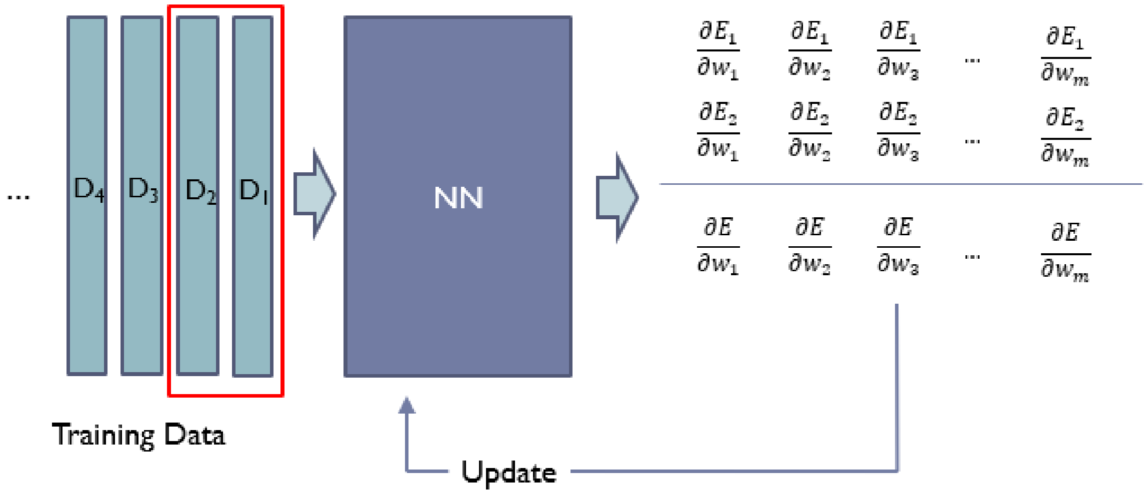
>High throughput

>Faster convergence : Good estimation + High throughput

Disadvantage

>Inaccurate, if dataset with large variances

Mini-batch Gradient Descent Method



- For an update, gradients are calculated for a batch

Better Gradient Descent Methods

1. Learning Rate

   In Neural Network, there are too many parameters

   When learning starts, a large learning rate is preferred but, when learning is also done, small learning rate is preferred,

   2. How to Escape Local Optimum

   Cross the hills to good better places

   >>Momentum

   All connection weights are not equally updated

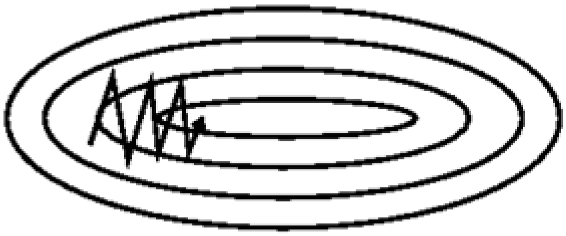
   >>Adaptive learning rates

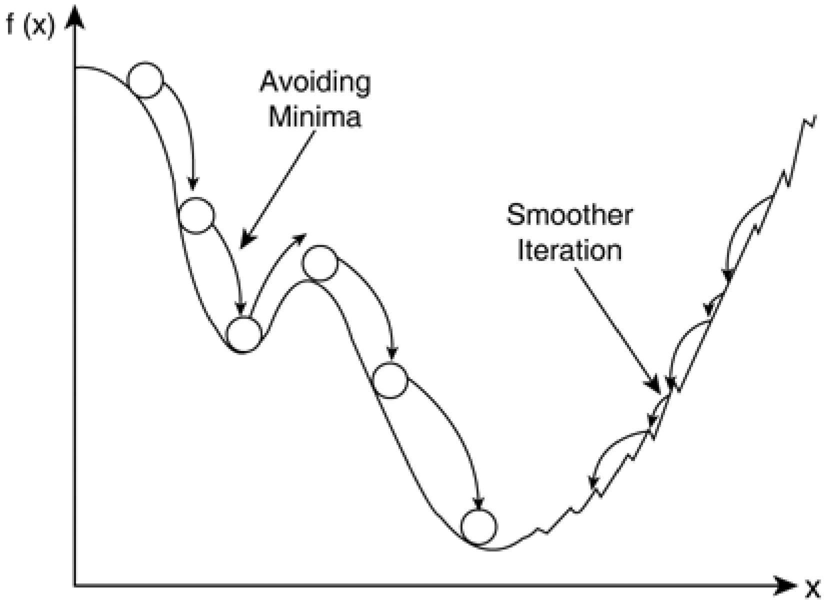
Momentum

Simple gradient method depends on the **current position**

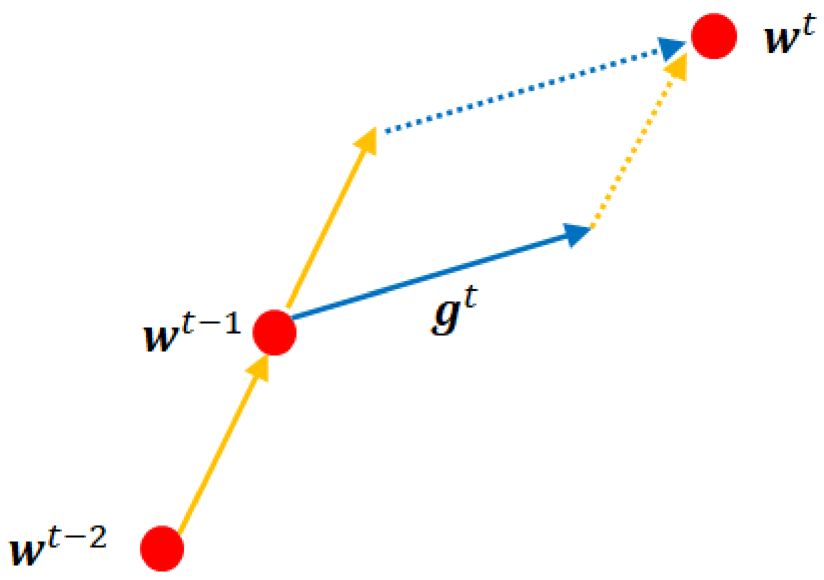
>Hard to avoid local minimum

>Gradient can vary much(oscillation)





How to cross the hill



g^t: 시간 t에서의 기울기

Momentum rate

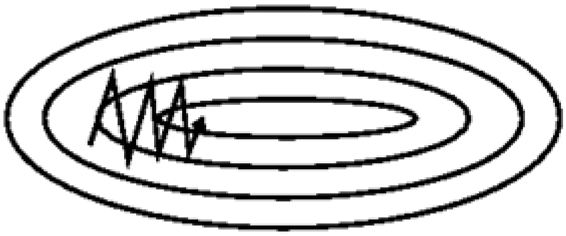
Learning rate

표준 방법	Momentum
$w^t = w^{t-1} - \eta g^t$	$m^t = \gamma\, m^{t-1} + \eta g^t$ $w^t = w^{t-1} - m^t$

- **Update parameters considering both the momentum and the gradient of the current position**

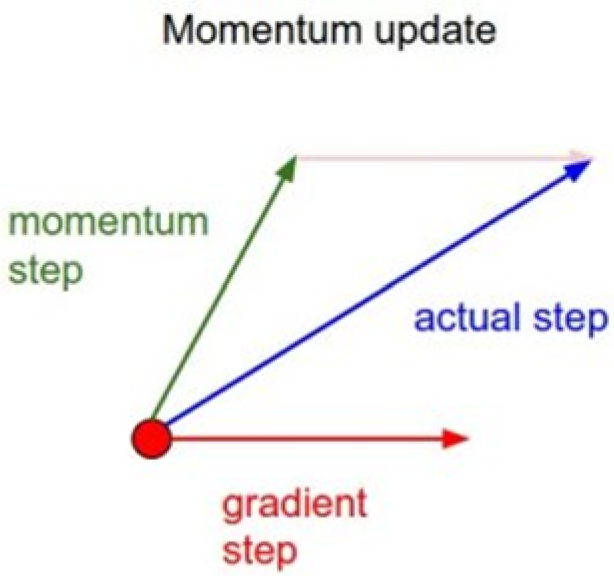
Momentum is the exponential average of the past gradient

$$m^t = \eta g^t + \gamma\, \eta g^{t-1} + \gamma^2 \eta g^{t-2} + \cdots$$



Disadvantage of Momentum

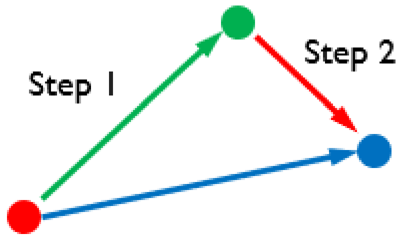
- simple addition of momentum may cause excessive update
- We may miss the position where to stop



Nesterov Accelerated Gradient(NAG)

Not a simple addition

1. Update the current position considering the momentum
2. Evaluate the gradient at the new position
3. Update the position consider the gradient



Adaptive Learning Rate

Adagrad

- 변수별로 학습률이 달라지게 조절하는 알고리즘
- Update much less-updated parameters
- Update less much-updated paramerters

$$G_i^t = G_i^{t-1} + \left(g_i^t\right)^2$$

$$w_i^t = w_i^{t-1} - \frac{\eta}{\sqrt{G_i^t + \epsilon}} g_i^t$$

저 제곱은 원소별 제곱이다. 저 값이 분모에 들어가니까 크기가 큰것은 gradient가 조금 변화, 작은것은 큰 변화!!

Eventually Gi becomes large as time goes on

Parameters are rarely updated at some time

학습을 진행할수록 갱신 강도가 약해진다.

이문제를 개선한 기법으로 RMSProp가 있다.

RMSProp

Instead of considering the total amount of updates,

let's consider the amount of recent updates!

g_i^t: parameter i의 시간 t에서의 기울기

G_i^t: parameter i가 update된 양

$$G_i^t = G_i^{t-1} + \left(g_i^t\right)^2$$
$$w_i^t = w_i^{t-1} - \frac{\eta}{\sqrt{G_i^t + \epsilon}} g_i^t$$

Adagrad

$$G_i^t = \gamma G_i^{t-1} + (1-\gamma)\left(g_i^t\right)^2$$
$$w_i^t = w_i^{t-1} - \frac{\eta}{\sqrt{G_i^t + \epsilon}} g_i^t$$

RMSProp

Very small value

먼 과거의 기울기는 서서히 잊고 새로운 기울기 정보를 크게 반영한다.

Adam

RmsProp + Momentum

g_i^t: parameter i의 시간 t에서의 기울기

G_i^t: parameter i가 update된 양

Momentum

$$m_i^t = \gamma m_i^{t-1} + \eta g_i^t$$
$$w_i^t = w_i^{t-1} - m_i^t$$

$$\rightarrow \quad m_i^t = \beta_1 m_i^{t-1} + (1-\beta_1) g_i^t$$

RMSProp

$$G_i^t = \gamma G_i^{t-1} + (1-\gamma)\left(g_i^t\right)^2$$
$$w_i^t = w_i^{t-1} - \frac{\eta}{\sqrt{G_i^t + \epsilon}} g_i^t$$

$$\rightarrow \quad G_i^t = \beta_2 G_i^{t-1} + (1-\beta_2)\left(g_i^t\right)^2$$

Unbiased expectation

$$m_i^t = \beta_1 m_i^{t-1} + (1-\beta_1) g_i^t$$

$$G_i^t = \beta_2 G_i^{t-1} + (1-\beta_2)\left(g_i^t\right)^2$$

$$\widehat{m}_i^t = \frac{m_i^t}{1-(\beta_1)^2} \qquad \hat{G}_i^t = \frac{G_i^t}{1-(\beta_2)^2}$$

$$w_i^t = w_i^{t-1} - \frac{\eta}{\sqrt{\hat{G}_i^t + \epsilon}} \widehat{m}_i^t$$