# Advanced Gradient Descent Methods
# Regularization

# CONTENTS

# CONTENTS

# Batch Gradient Descent

**BGD**

# BGD

Based on Batch(total Data)

$$W := W - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(Wx^i - y^i\right) x^i$$

# Stochastic Gradient Descent

SGD

# SGD

Based on (One Data)

$$W := W - \alpha \frac{1}{m}\left(Wx^{i} - y^{i}\right)x^{i}$$

# Mini-Batch Gradient Descent

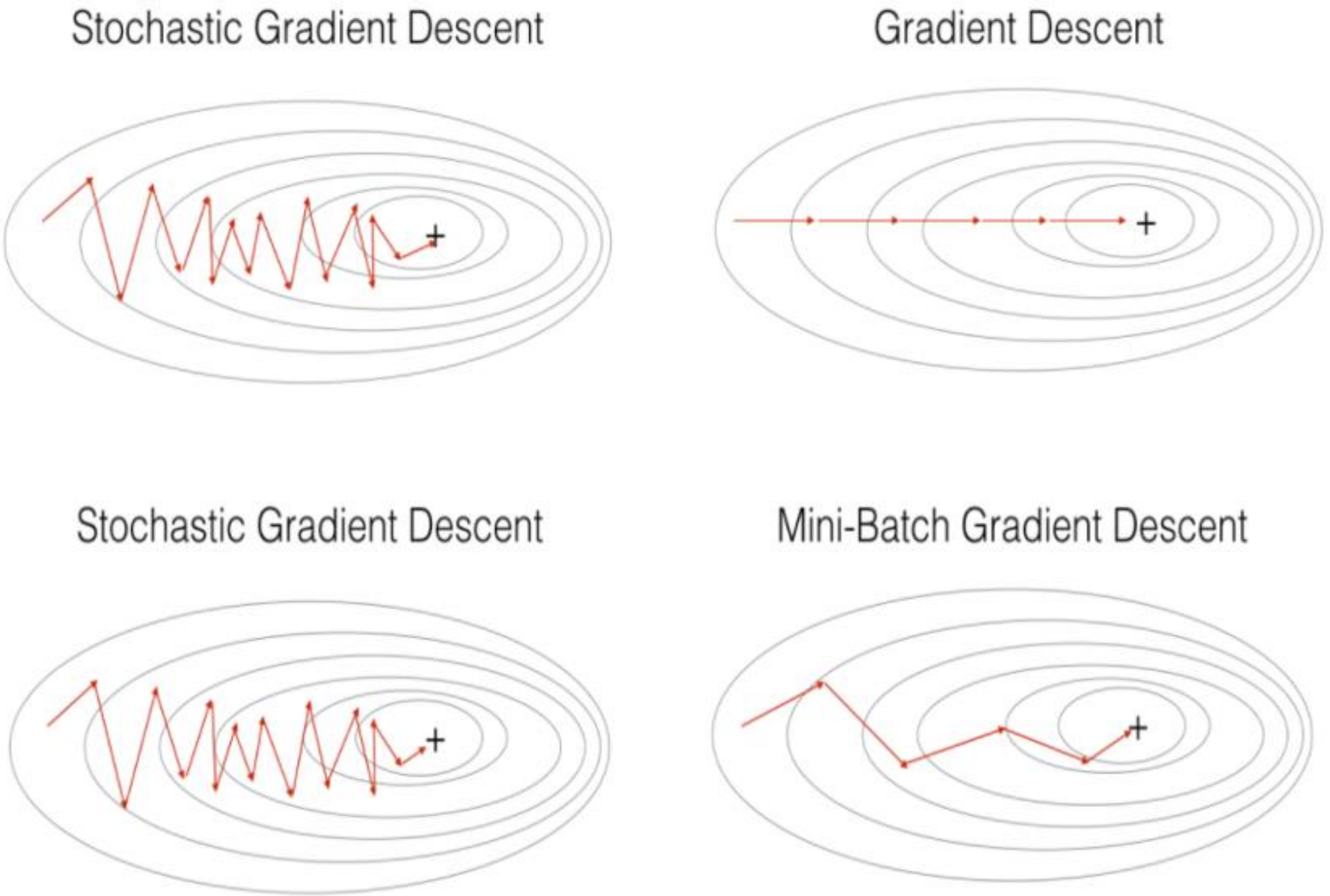
MSGD

# MSGD

Based on Mini Batch (One Data)

$$W \coloneqq W - \alpha \frac{1}{m} \sum_{i=1}^{m} \left( W x^i - y^i \right) x^i$$

# Visualize




출처 : https://light-tree.tistory.com/133

# Momentum

Momentum

# Momentum

Like Acceleration => Rolling a ball

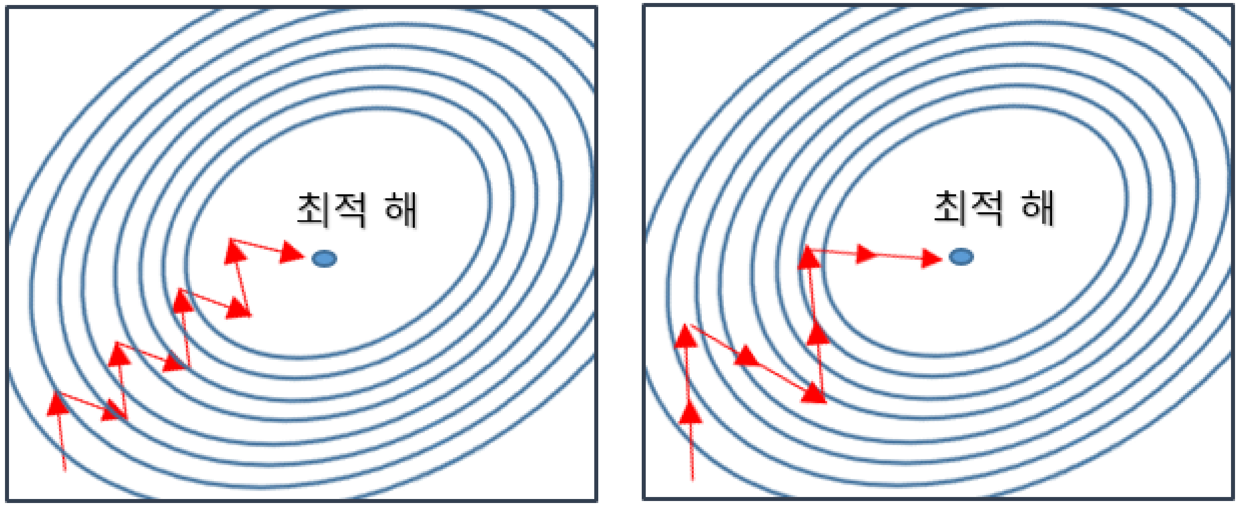
$$V(t) = m * V(t-1) - \alpha \frac{\partial}{\partial w} \mathrm{Cost}(w)$$

V(0)=0

W(t+1)=W(t)+V(t)

W=weight

# Momentum




확률적 경사 하강법　　　　모멘텀

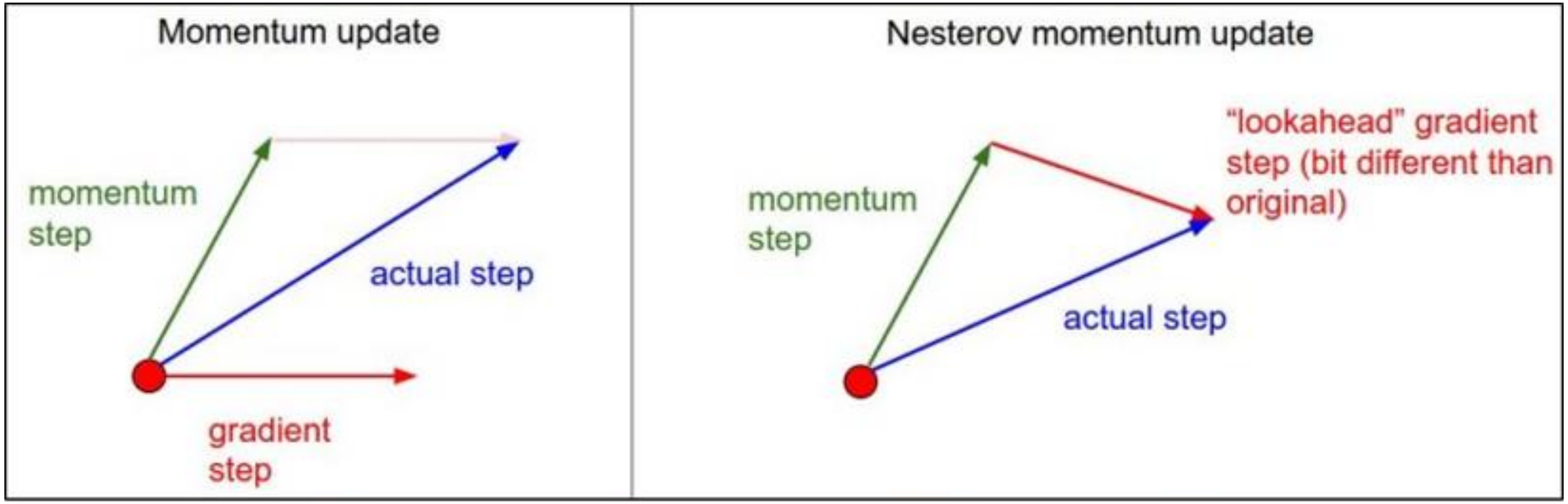
출처 : https://twinw.tistory.com/247

# Nesterov Accelrated Gradient

NAG

# NAG

## In Momentum Step, gradient




Difference between Momentum and NAG. Picture from CS231.

출처 : https://twinw.tistory.com/247

# NAG

$$V(t)=m*V(t-1)-\alpha\frac{\partial}{\partial(w+m*V(t-1))}Cost(w)$$

$$W(t+1)=W(t)+V(t)$$

# NAG

# Adaptive Gradient

Adagrad

# Adagrad

$$G(t)=G(t\text{-}1)+\left(\frac{\partial}{\partial w(t)}Cost(w(t))\right)^2$$ G(t) = Vector W[i] element

$$W(t+1)=W(t)-\alpha\frac{1}{\sqrt{G(t)+\epsilon}}\frac{\partial}{\partial w(i)}Cost(w(i))$$ W(t) = Vector W[i] element

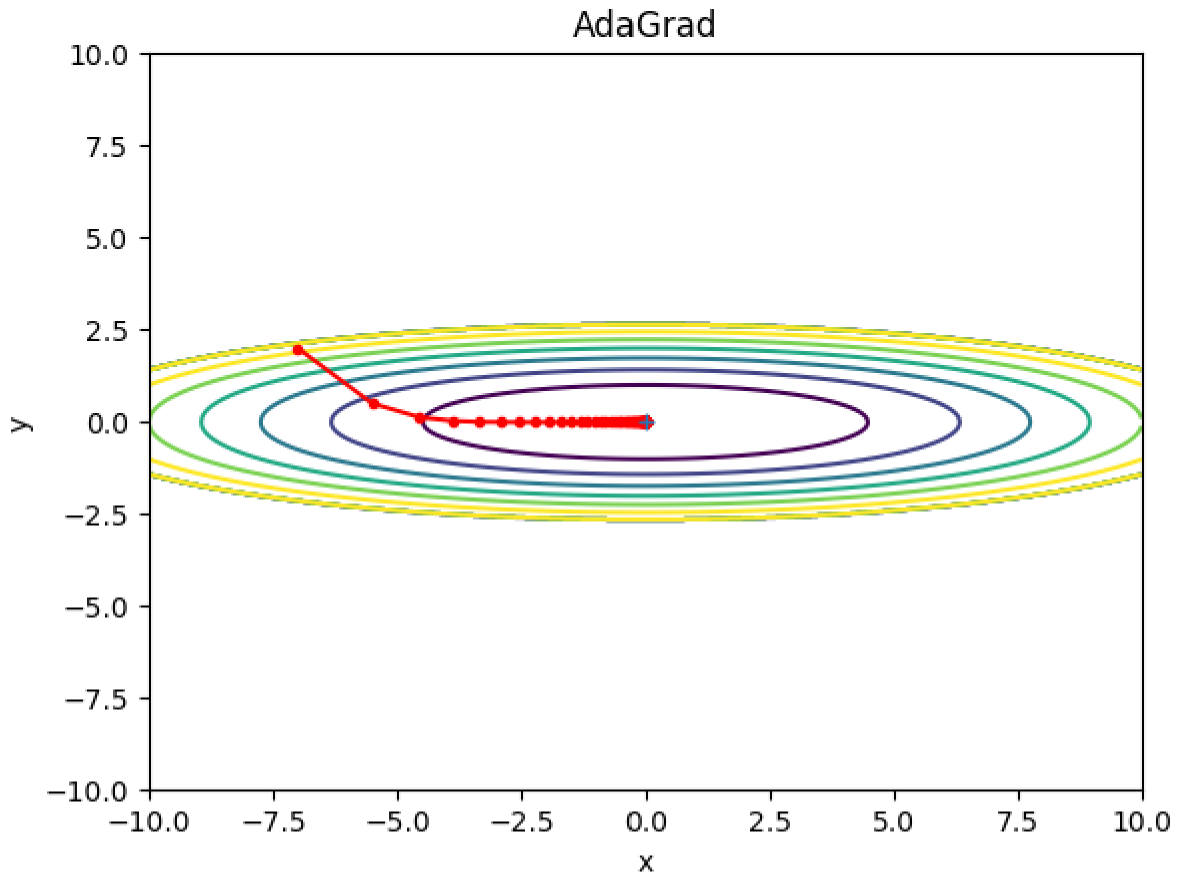
Adagrad

# Problem

1. $G(0)=0$ and $G(t)=0$, insert $\in$

2. Infinite Training $G(t)$ is infinite

# Adagrad




출처 : https://twinw.tistory.com/247

# Adaptive Gradient

RMSProp

# RMSProp

It Complements the adagrad

$$G(t) = \gamma G(t-1) + (1-\gamma)\left(\frac{\partial}{\partial w(t)} Cost(w(t))\right)^2$$

G(t) = Vector W[i] element

$$W(t+1) = W(t) - \alpha \frac{1}{\sqrt{G(t) + \epsilon}} \frac{\partial}{\partial w(i)} Cost(w(i))$$

W(t) = Vector W[i] element

$\gamma$\`s Value => 0.9~0.999

# Adaptive Gradient

## AdaDelta

# Adadelta

Update Parameter W, W`s unit ?
Remove Learning Rate

$$G(t) = \gamma G(t-1) + (1-\gamma)\left(\frac{\partial}{\partial w(t)} Cost(w(t))\right)^2$$

G(t) = Vector W[i] element

$$S(t) = \gamma S(t-1) + (1-\gamma)\Delta\theta^2$$

Hesian Matrix

$$W(t+1) = W(t) - \frac{\sqrt{S(t)+\epsilon}}{\sqrt{G(t)+\epsilon}} \frac{\partial}{\partial w(i)} Cost(w(i))$$

W(t) = Vector W[i] element

$\gamma$`s Value => 0.9~0.999

Hesian Matrix : https://bskyvision.com/661

참고
https://hiddenbeginner.github.io/deeplearning/2019/09/22/optimization_algorithms_in_deep_learning.html

# Animation

애니메이션
http://shuuki4.github.io/deep%20learning/2016/05/20/Gradient-Descent-Algorithm-Overview.html

# Adaptive Moment Estimation

Adam

# Adam

Moment + Adaptive

Moment is not Momentum Probability Moment

What is Moment ? => Kocw 김충락 교수님(수리통계학)

1-Moment=>E[X]

Not Known Moment => Estimation

2-Moment=>E[$X^2$]

참고
https://hiddenbeginner.github.io/deeplearning/2019/09/22/optimization_algorithms_in_deep_learning.html

# Adam

$$m_t = \beta_1 m_{t-1} + (1 + \beta_1) g_t$$

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

If Initial m, v is 0 , weight=>Zero biased

If decay rate is small, $(\beta_1, \beta_2 \ close\ one)$ weight=>biased

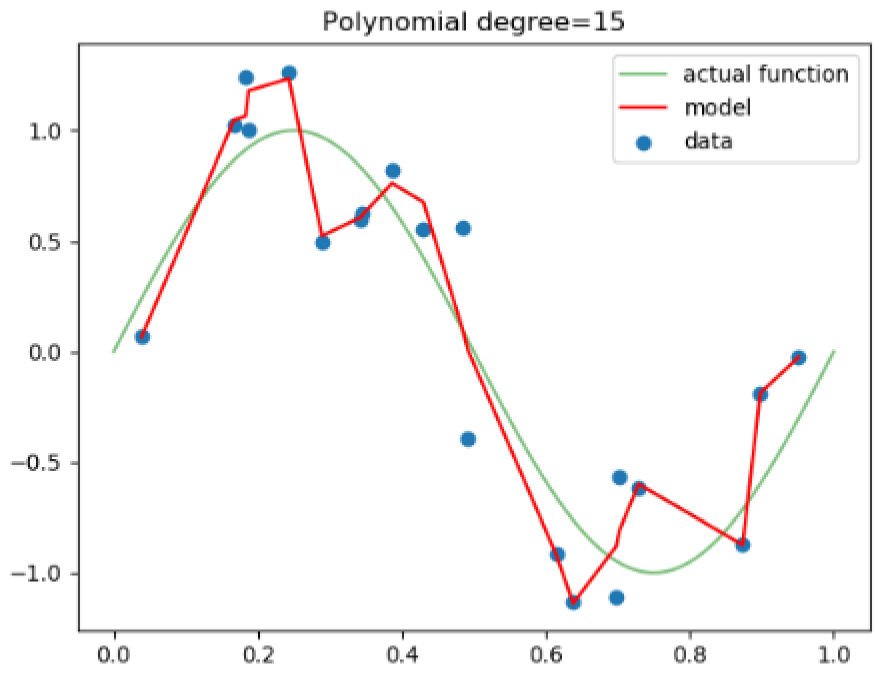
$$m_t = \beta_1 m_{t-1} + (1 + \beta_1) g_t$$

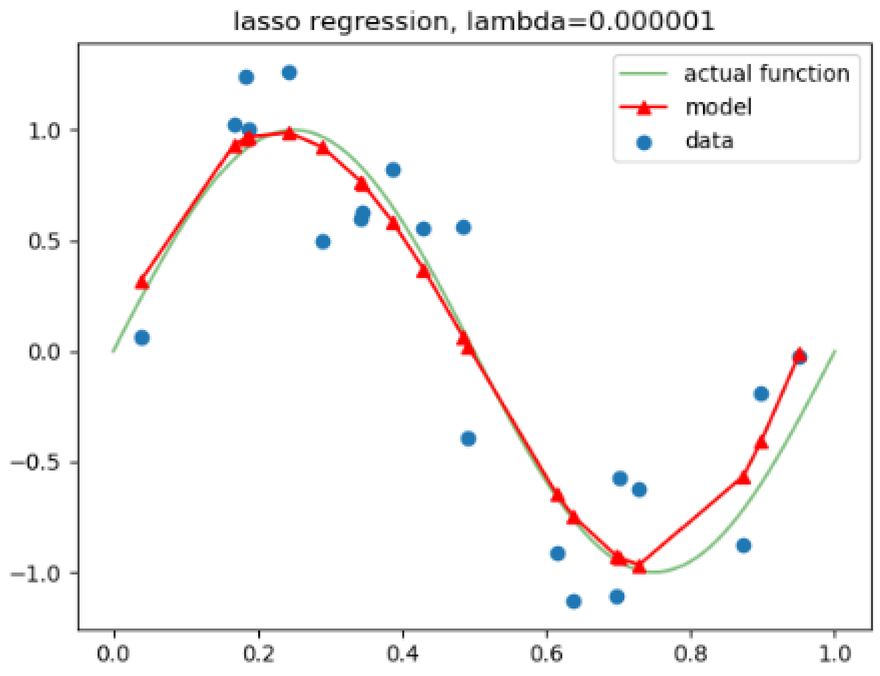
참고
https://hiddenbeginner.github.io/deeplearning/2019/09/22/optimization_algorithms_in_deep_learning.html

# Regularization

Regularization

# Regularization




참고,출처 : https://light-tree.tistory.com/125

# L1-Norm

# L1 − Norm

## Lasso

$$J(\theta) = \frac{1}{2m}\sum\left(h\left(x^i\right) - y^i\right)^2$$

$$J(\theta) = \frac{1}{2m}\sum\left(h\left(x^i\right) - y^i\right)^2 + \frac{\tau}{2}\sum\left|\theta_j\right|$$

# Ridge

L2-Norm

# L2−Norm

## Ridge

$$J(\theta) = \frac{1}{2m}\sum\left(h\left(x^i\right) - y^i\right)^2$$

$$J(\theta) = \frac{1}{2m}\sum\left(h\left(x^i\right) - y^i\right)^2 + \frac{\tau}{2}\sum \theta_j^2$$

참고,출처 : https://light-tree.tistory.com/125

# L2−Norm

## Ridge

$$\theta_j := \theta_j \text{-} \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^i\right) - y^i\right) x_j^i$$

$$\theta_j := \theta_j (1 - \alpha \frac{\tau}{m}) \text{-} \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^i\right) - y^i\right) x_j^i$$

감사합니다

THANK YOU