CS231n Lecture 3

- BOAZ 10기 박성현
- BOAZ 11기 김태희
- BOAZ 11기 홍지민
- BOAZ 10기 김용규

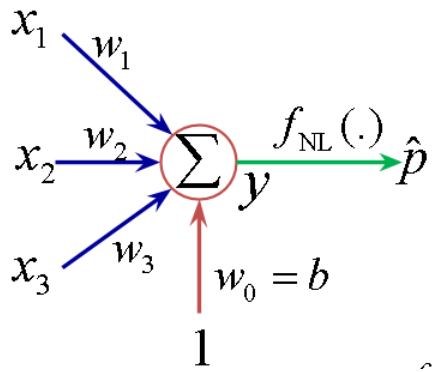
Loss Function & Optimization

Train은 길어도 상관없어! 새로운 인풋에 대해 빠르고 성능 좋게 표현할만한 방법이 없을까?

Parametric approach!

- 데이터의 분포를 가장 잘 설명하는 파라미터에 인풋 값을 넣으면 예측값이 바로 나오도록 하기! (Maximum Likelihood Estimation)

즉 모델은 설명변수와 종속변수 간의 함수관계를 만들어 결과를 도출하는 일!



$$L(\theta) = p(\vec{y}|X;\theta)$$

01 Loss Function & Optimization

How to find?

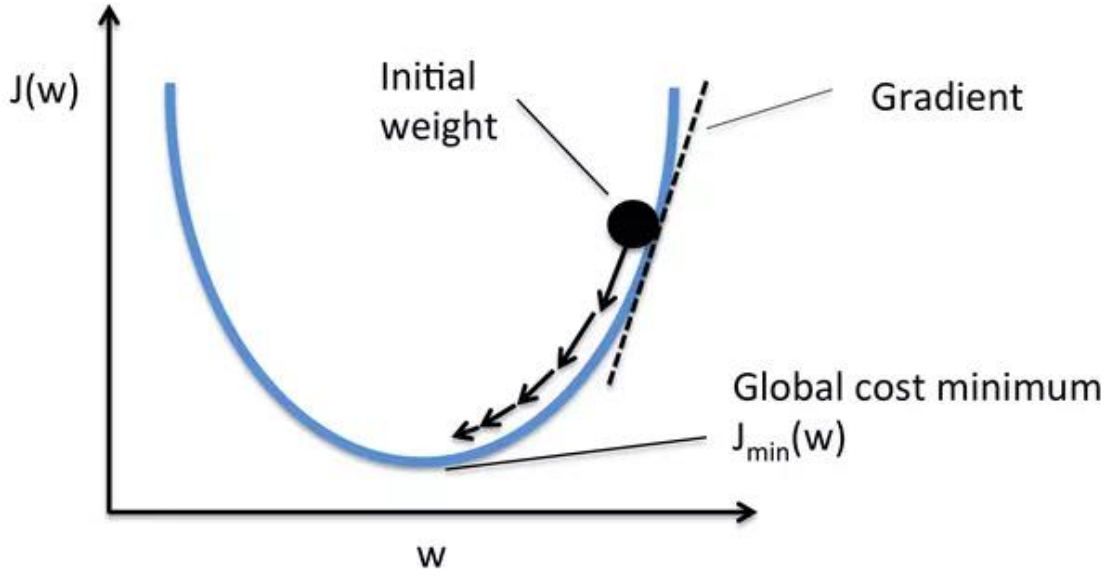
1. Define Loss function과 optimization

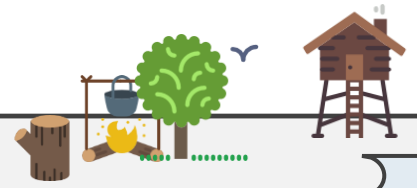
대부분의 모델에서는 에러를 나타내는
Loss function과 그것을 최소화 시키는
Optimization 과정이 존재

물론 Optimization 과정이 없는 모델도 존재!

Ex) random forest, decision tree 등등
voting 방식의 greedy algorithm
(Tree 모델은 1/31에 11기 조단비 발표 때 상세히!)

대표적인 optimization : Gradient Descent



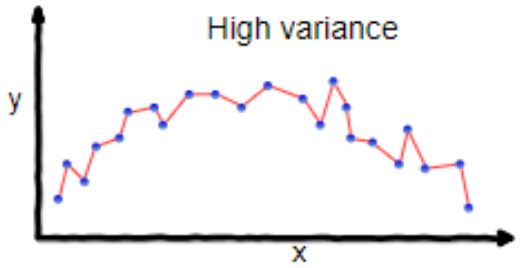


Loss Function & Optimization

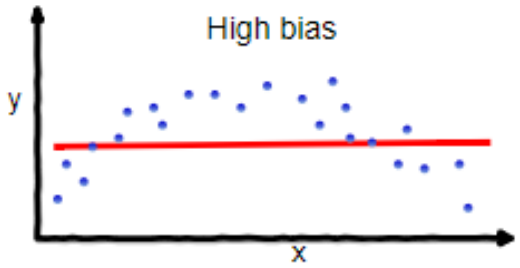
최적화된 모델의 성능 저하를 유발하는 error의 3가지 구성요소

Error =Bias + Variance + Noise

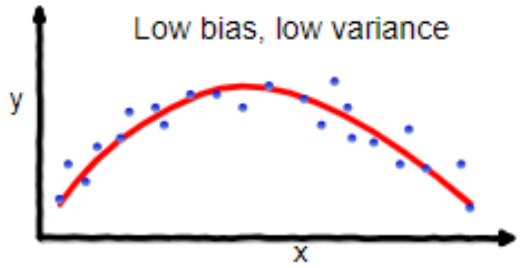
Bias와 variance는 서로 tradeoff 관계



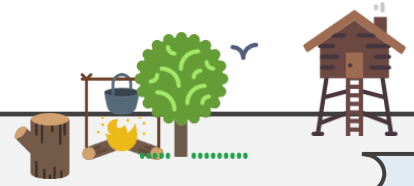
overfitting



underfitting



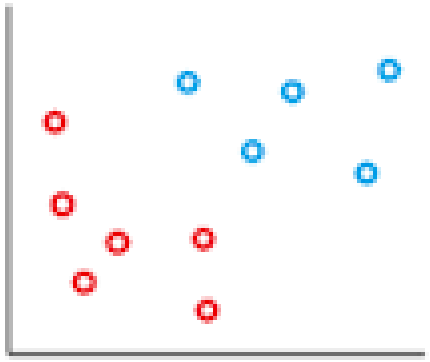
Good balance

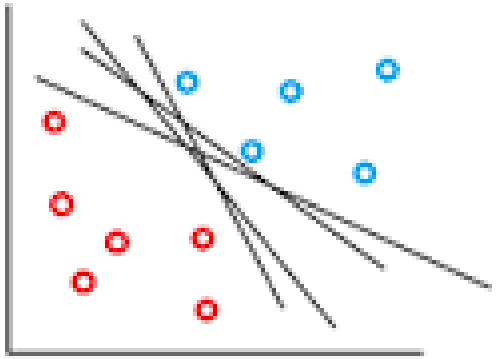


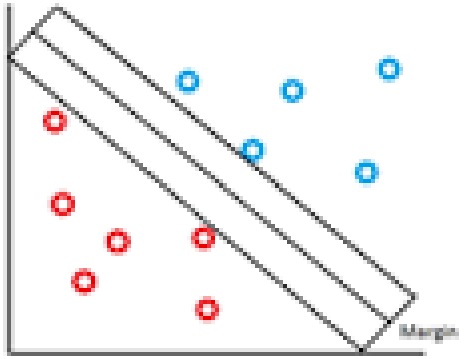
Compare Loss Function

[Support Vector Machine]

Linearly separable한 decision Boundary는 여러 개지만 그 중 가장 좋은 건 뭘까?



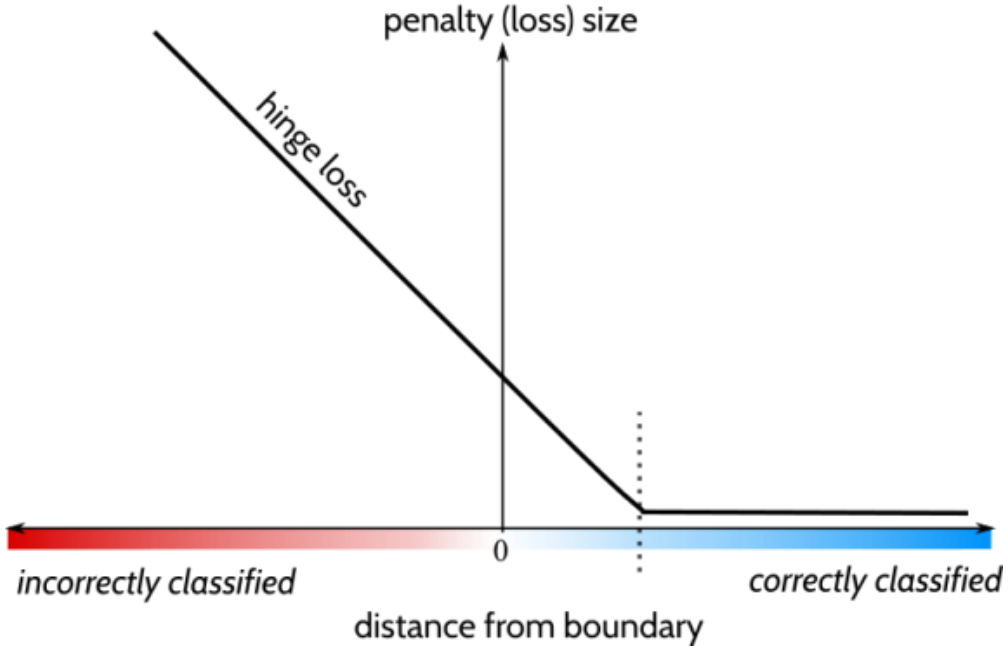


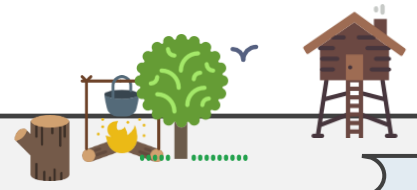


Compare Loss Function

[Hinge Loss]

$$\arg\min_{w,w_0} \sum_{i=0}^{n-1} \max(0, 1 - y_i(w^T x_i + w_0))$$





Compare Loss Function

[Softmax]

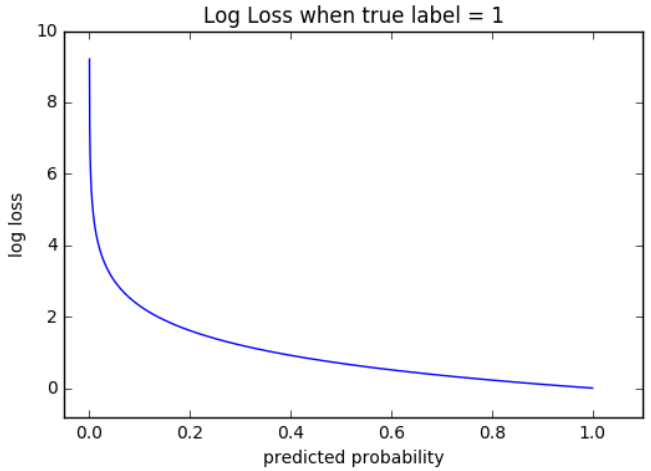
$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$

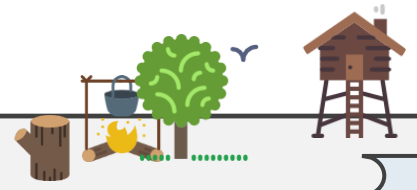
Softmax로 normalize한 이후 cross entropy 수식에 대입

$\hat{\mathbf{y}}$ $\qquad$ $\mathbf{y}$

$$\begin{bmatrix} 0.1 \\ 0.5 \\ 0.4 \end{bmatrix} \quad D(\hat{\mathbf{y}}, \mathbf{y}) = -\sum_j y_j \ln \hat{y}_j \quad \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

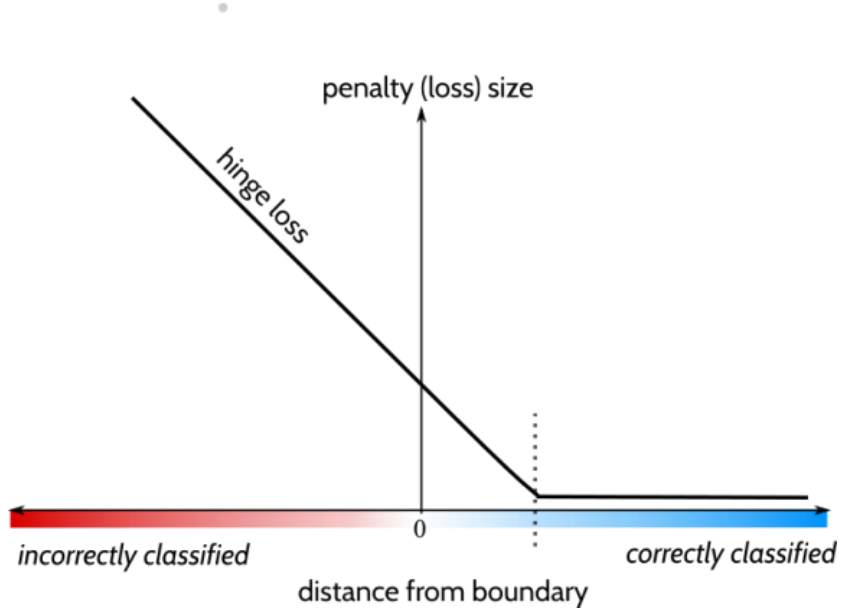
[Cross-entropy Loss]

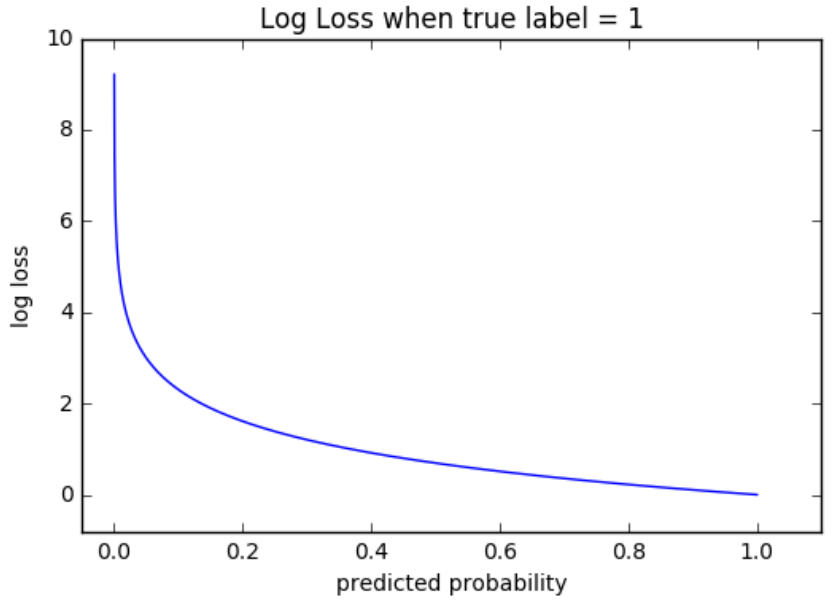




Compare Loss Function

Hinge Loss와 Cross-entropy Loss function 비교





결론: loss function을 어떻게 정의하느냐에 따라 모델의 성능과 Parameters가 바뀐다!

Overfitting & Regularization

모델에 과적합(Overfitting)을 방지! (Variance High)

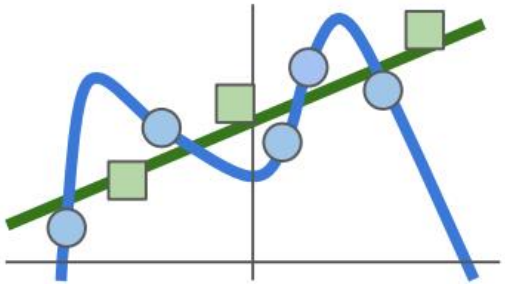
$$L(W) = \underbrace{\frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W), y_i)}$$

Data loss: Model predictions should match training data

$$L(W) = \underbrace{\frac{1}{N}\sum_{i=1} L_i(f(x_i, W), y_i)}_{\text{Data loss}} + \underbrace{\lambda R(W)}_{\text{Regularization}}$$

Data loss: Model predictions should match training data

Regularization: Model should be “simple”, so it works on test data



L1 vs L2 Regularization

L1 regularization on least squares:

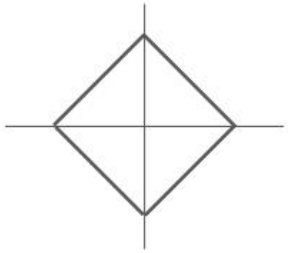
$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_j \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2 + \lambda \sum_{i=1}^{k} |w_i|$$

L2 regularization on least squares:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_j \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2 + \lambda \sum_{i=1}^{k} w_i^2$$

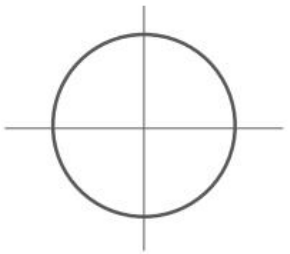
L1 (Manhattan) distance

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$

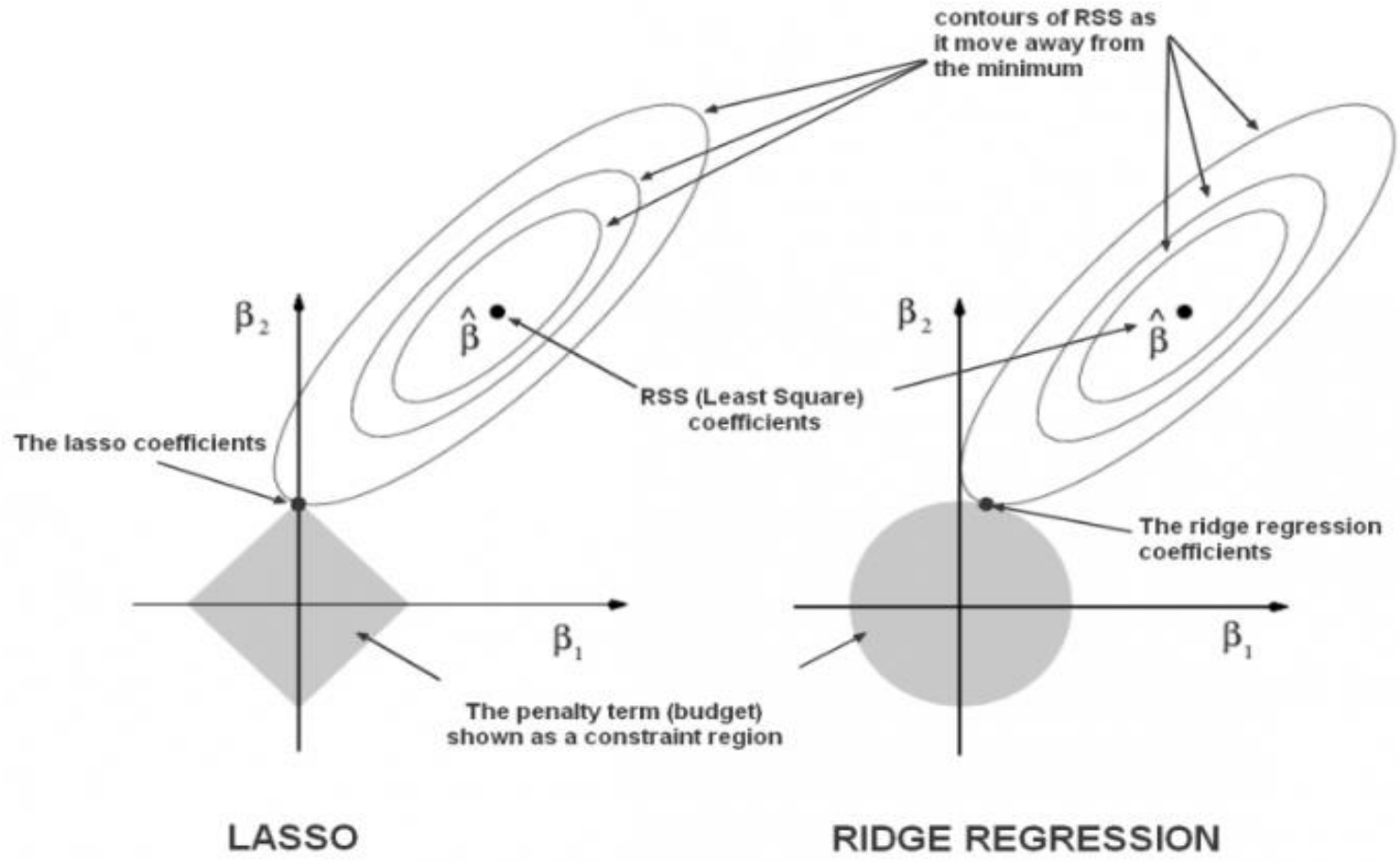


L2 (Euclidean) distance

$$d_2(I_1, I_2) = \sqrt{\sum_p (I_1^p - I_2^p)^2}$$

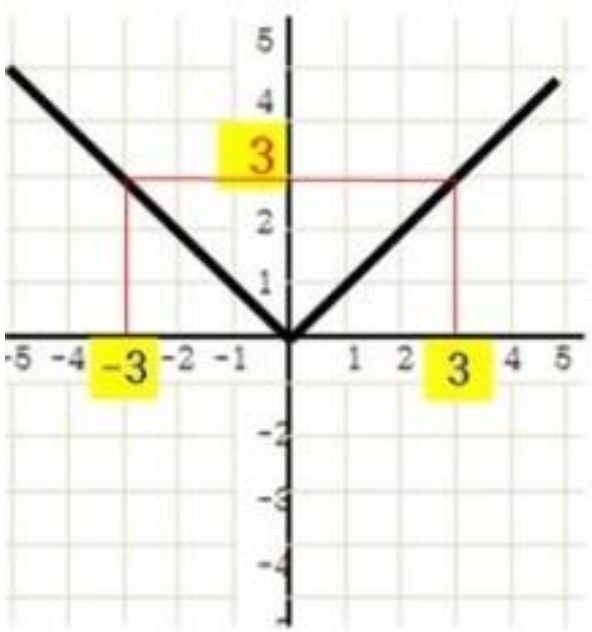


L1 vs L2 Regularization



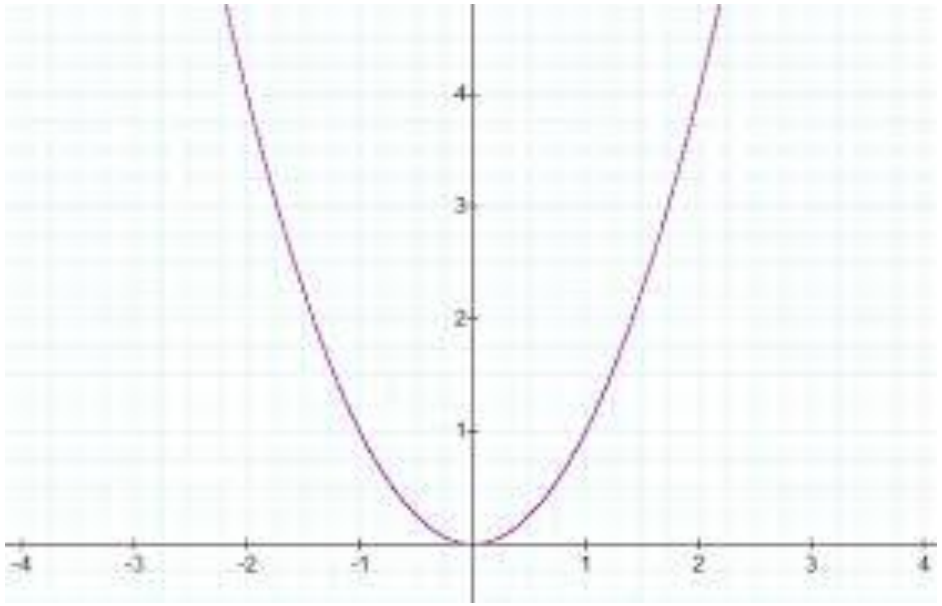
L1 vs L2 Regularization

L1 - Numerical한 방식으로 최적화



y=|x|

L2 - 미분을 통한 최적화



Conclusion

In common use:

L2 regularization $R(W) = \sum_k \sum_l W_{k,l}^2$

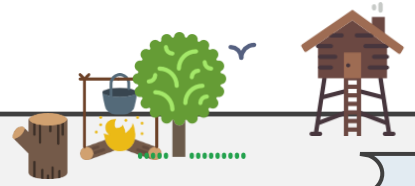
L1 regularization $R(W) = \sum_k \sum_l |W_{k,l}|$

Elastic net (L1 + L2) $R(W) = \sum_k \sum_l \beta W_{k,l}^2 + |W_{k,l}|$

Max norm regularization (might see later)

Dropout (will see later)

Fancier: Batch normalization, stochastic depth



Optimization

각 모델에서 고유한 optimization이 있겠지만
Gradient based optimization이 가장 보편적임!

$$\Theta_{n+1} = \Theta_n - \alpha \frac{\partial}{\partial \Theta_n} J(\Theta_n)$$

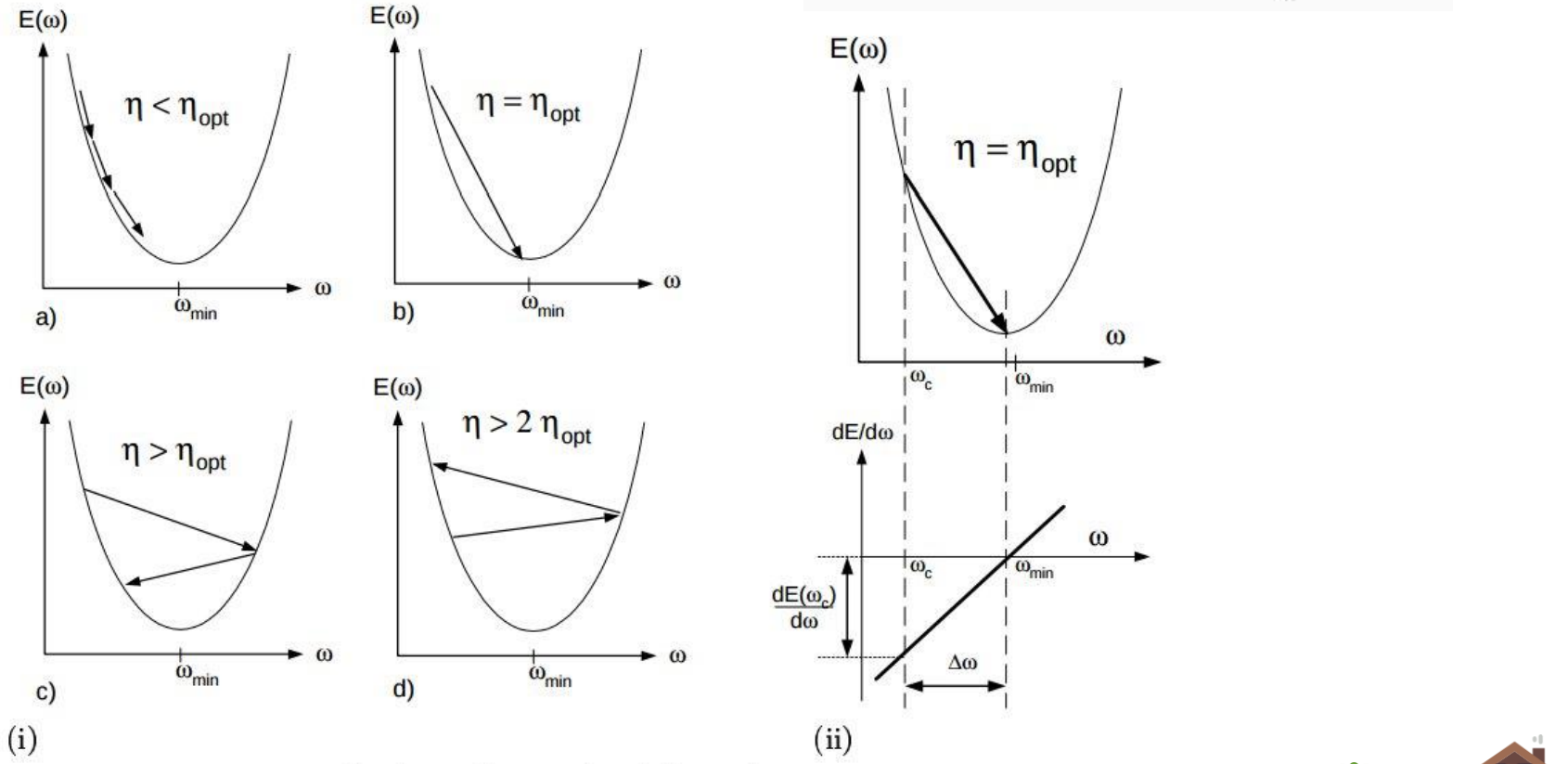


Fig. 6. Gradient descent for different learning rates.

Numerical vs Analytic Gradient

[Numerical Gradient]
수식 h에 직접 작은 값을 대입

(1.25322 - 1.25347)/0.0001
= -2.5

$$\frac{df(x)}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

[Analytic Gradient]
함수 짤 때 그 자체에 수식 대입!

SOL.

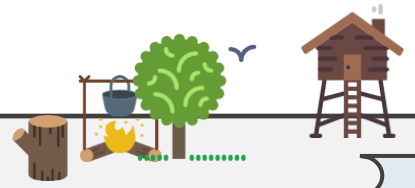
$$\frac{\partial f}{\partial x} = \cos(x+y^2) \times \frac{\partial}{\partial x}(x+y^2) = \cos(x+y^2) \times 1$$

$$\frac{\partial f}{\partial y} = \cos(x+y^2) \times \frac{\partial}{\partial y}(x+y^2) = \cos(x+y^2) \times 2y$$

$f(x,y) = h(g(x,y))$의 편미분은

$$\frac{\partial f}{\partial x} = h'(g(x,y)) \times \frac{\partial}{\partial x} g(x,y)$$

$$\frac{\partial f}{\partial y} = h'(g(x,y)) \times \frac{\partial}{\partial y} g(x,y)$$



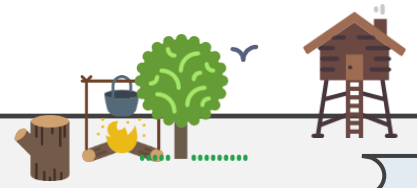
End

참고 자료

CS231n : http://cs231n.stanford.edu/syllabus.html

CS231n 한글 : http://aikorea.org/cs231n/neural-networks-3/

Bias-Variance Trade-off 강의 : https://www.youtube.com/watch?v=FOu8bXV15F8&t=508s



BOAZ