

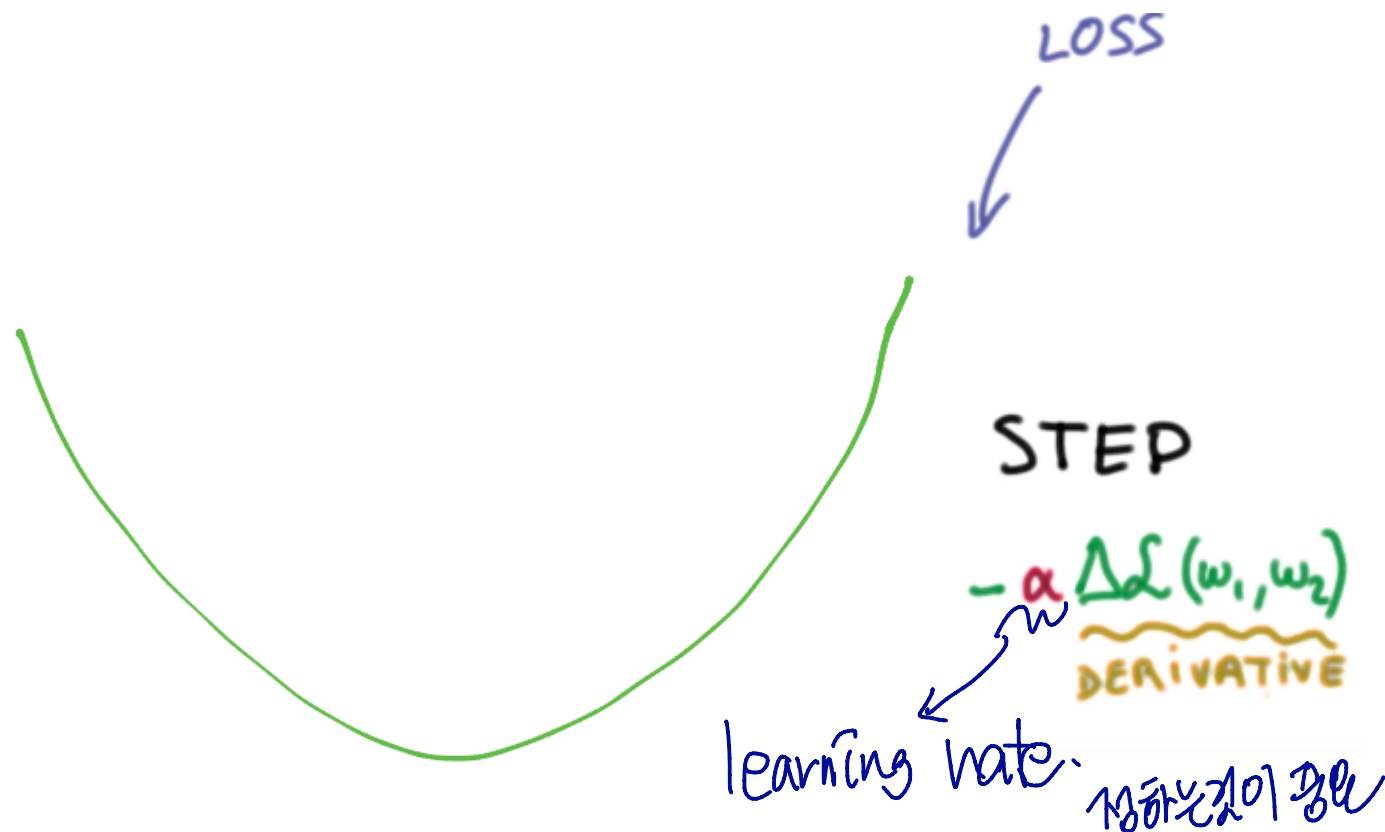
Lecture 7-I

Application & Tips:

Learning rate, data preprocessing,  overfitting

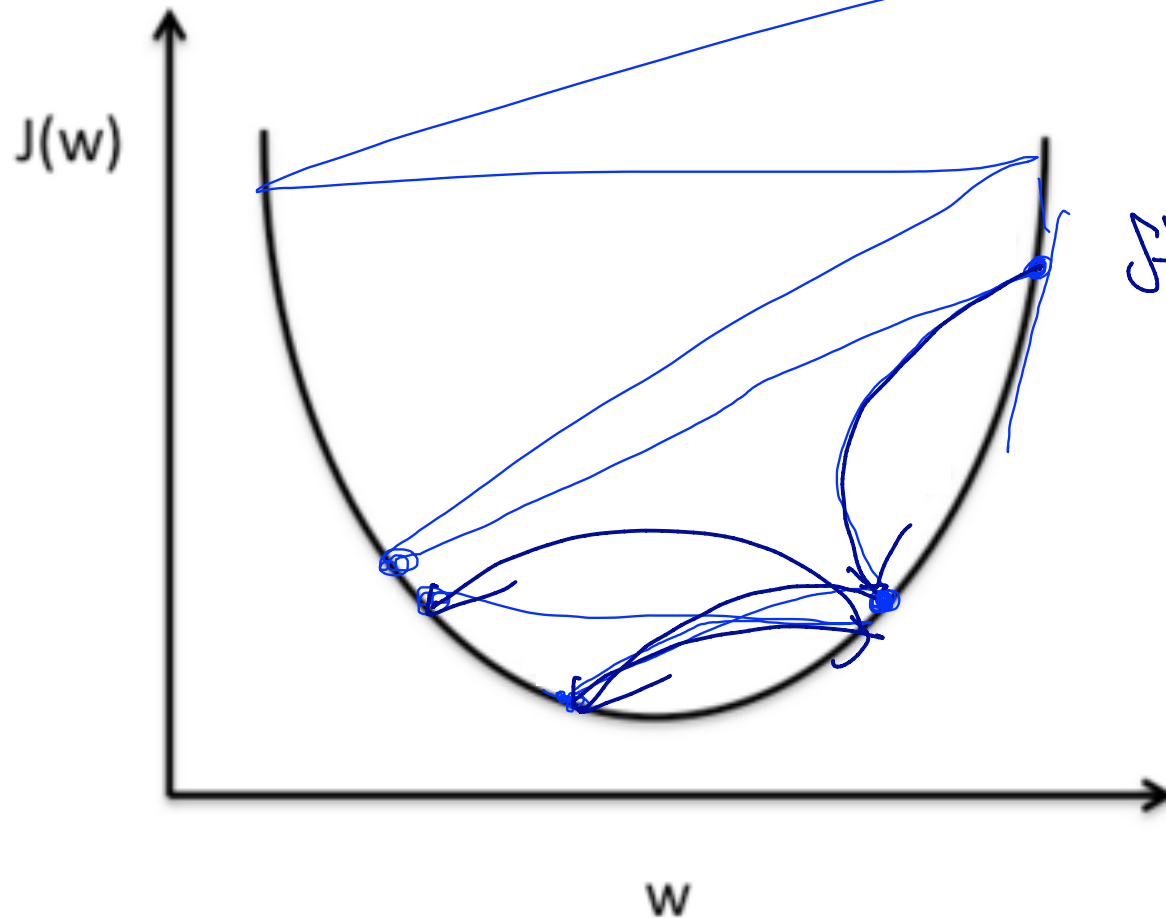
Sung Kim <hunkim+mr@gmail.com>

```
# Minimize error using cross entropy
learning_rate = 0.001
cost = tf.reduce_mean(-tf.reduce_sum(Y*tf.log(hypothesis), reduction_indices=1)) # Cross entropy
optimizer = tf.train.GradientDescentOptimizer(learning_rate).minimize(cost) # Gradient Descent
```



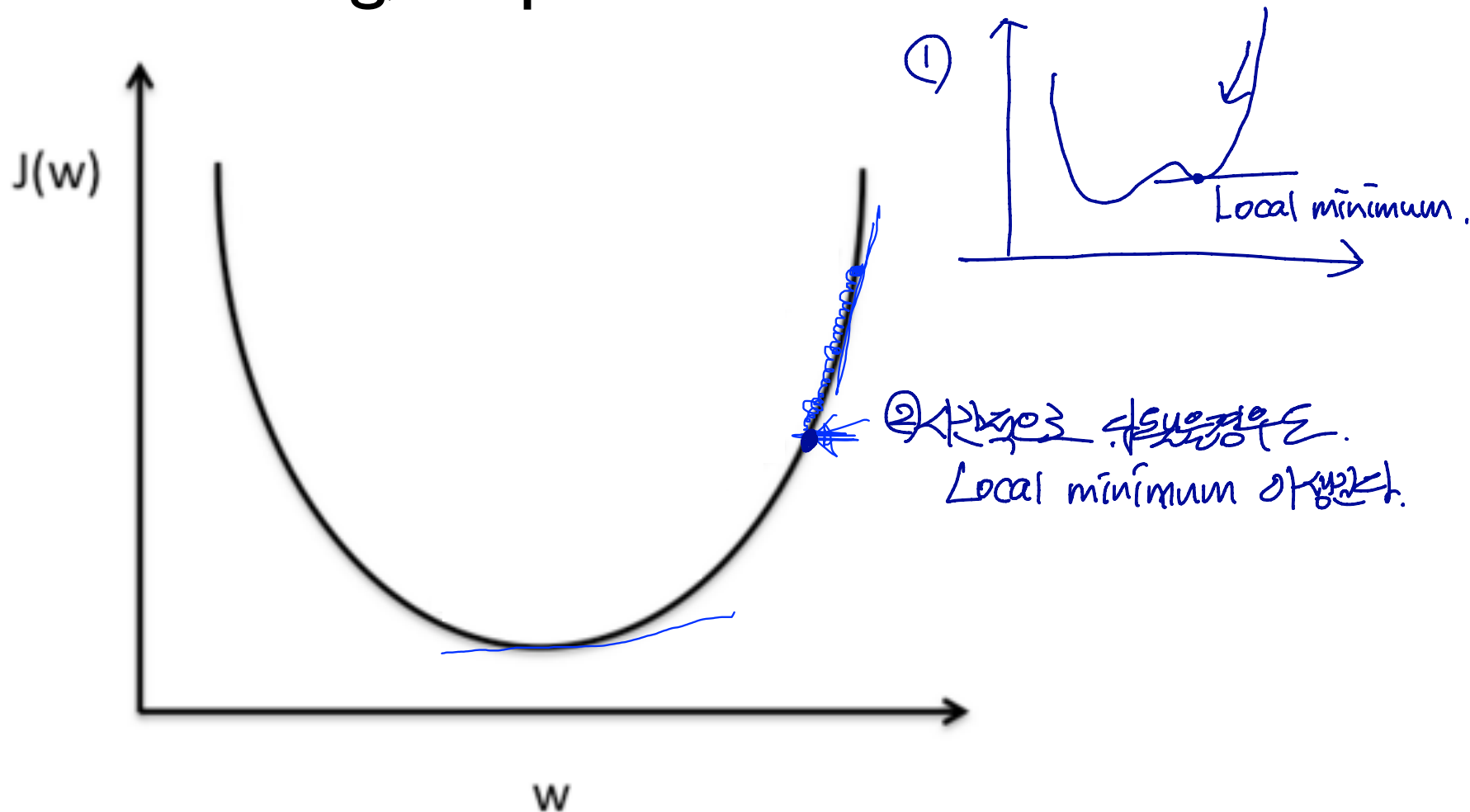
Large learning rate: overshooting

Cost가 증가함, 최소값을 찾지 못함



Step \uparrow (learning rate \uparrow)

Small learning rate:
takes too long, stops at local minimum

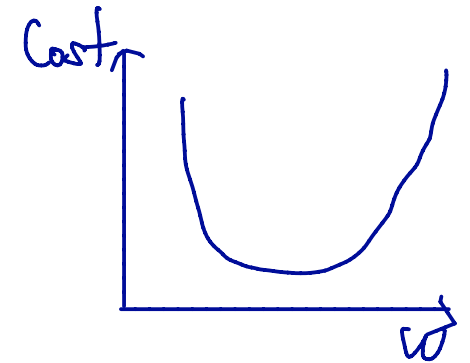
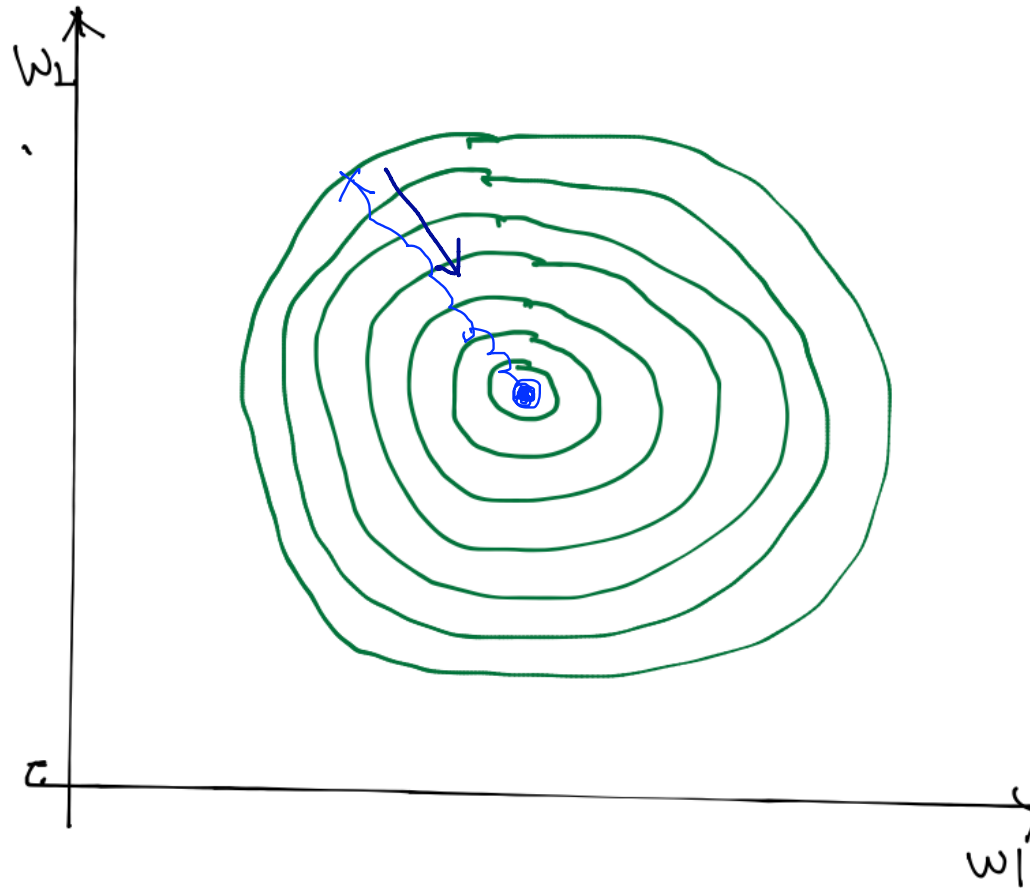


Try several learning rates

0.01

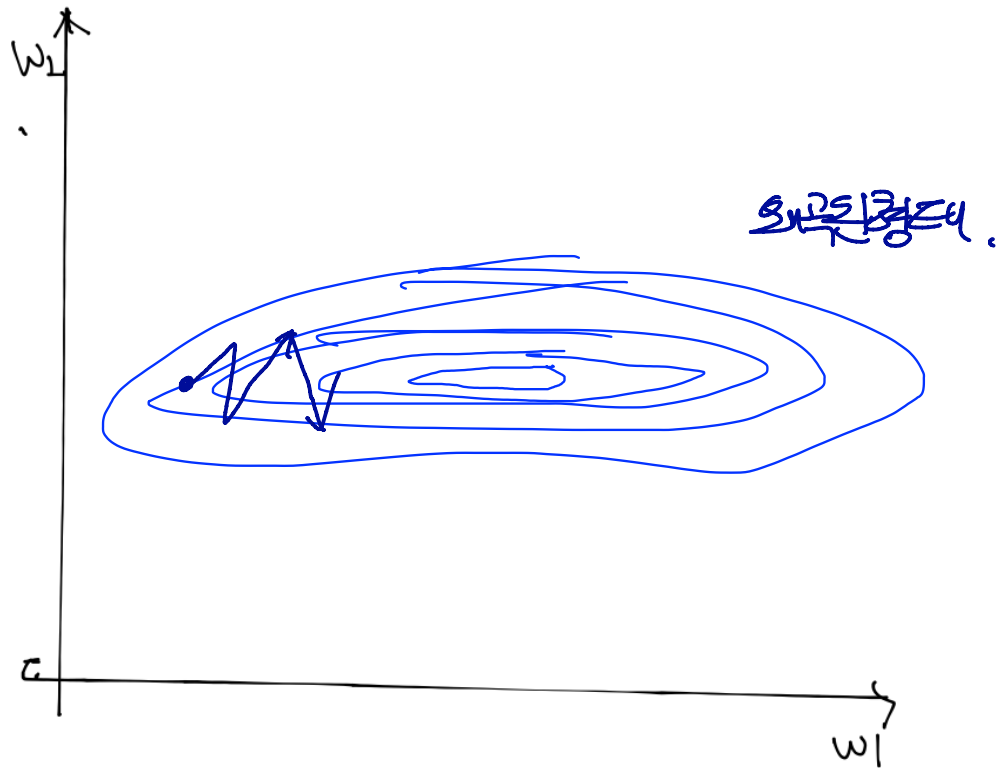
- Observe the cost function
- Check it goes down in a reasonable rate

Data (X) preprocessing for gradient descent



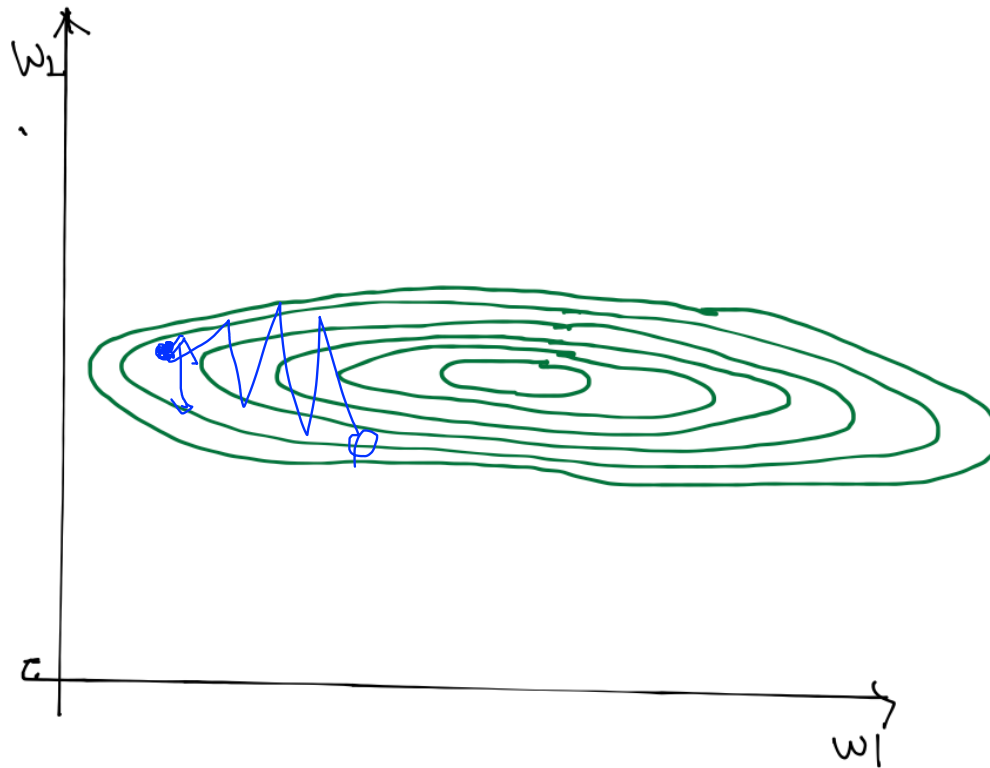
Data (X) preprocessing for gradient descent

x1	x2	y
1	9000	A
2	-5000	A
4	-2000	B
6	8000	B
9	9000	C



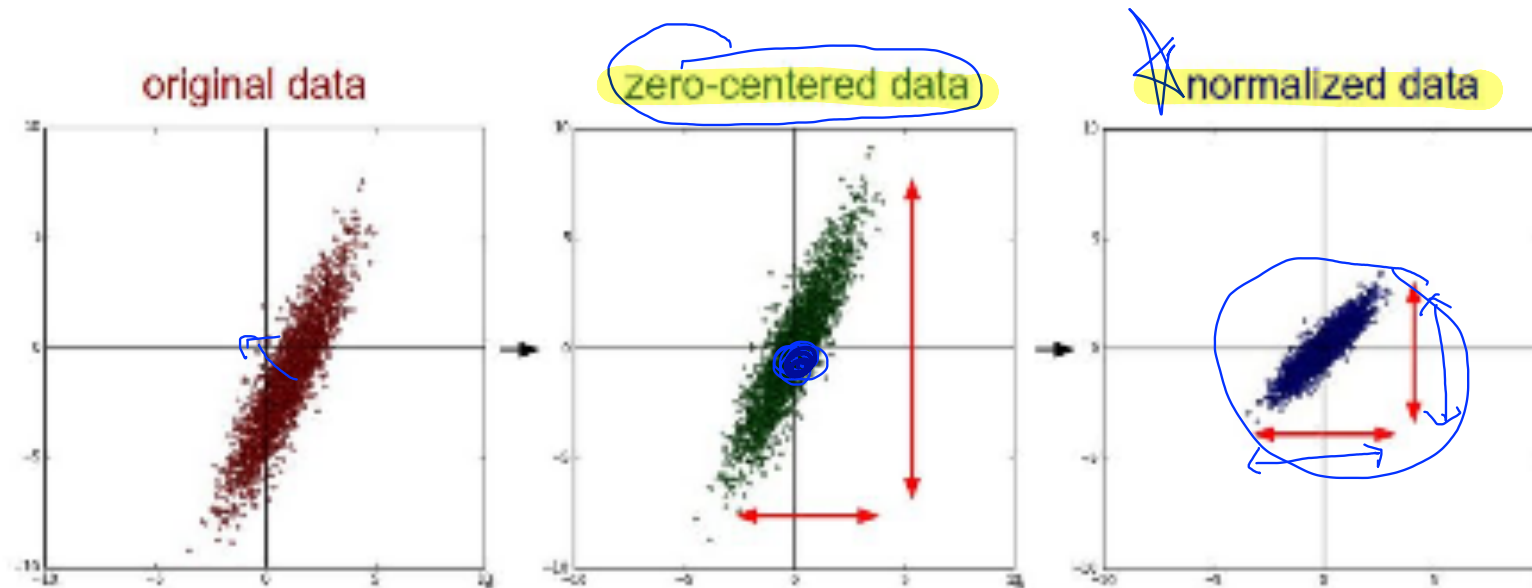
Data (X) preprocessing for gradient descent

x1	x2	y
1	9000	A
2	-5000	A
4	-2000	B
6	8000	B
9	9000	C



Normalizer.

Data (X) preprocessing for gradient descent



이런 방법의 단점이
평균값이 0이
않을 경우

Datamining 2015 48

Standardization (Normalization)

$$\boxed{x'_j = \frac{x_j - \mu_j}{\sigma_j}} \quad \star$$

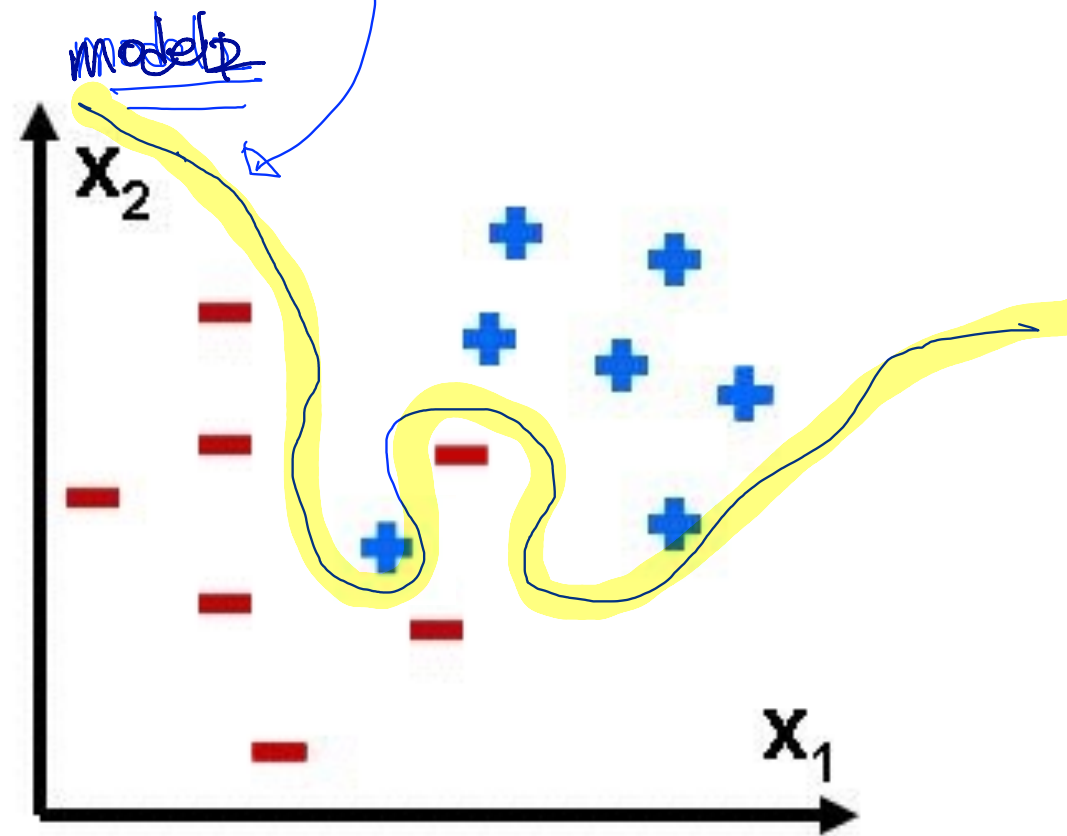
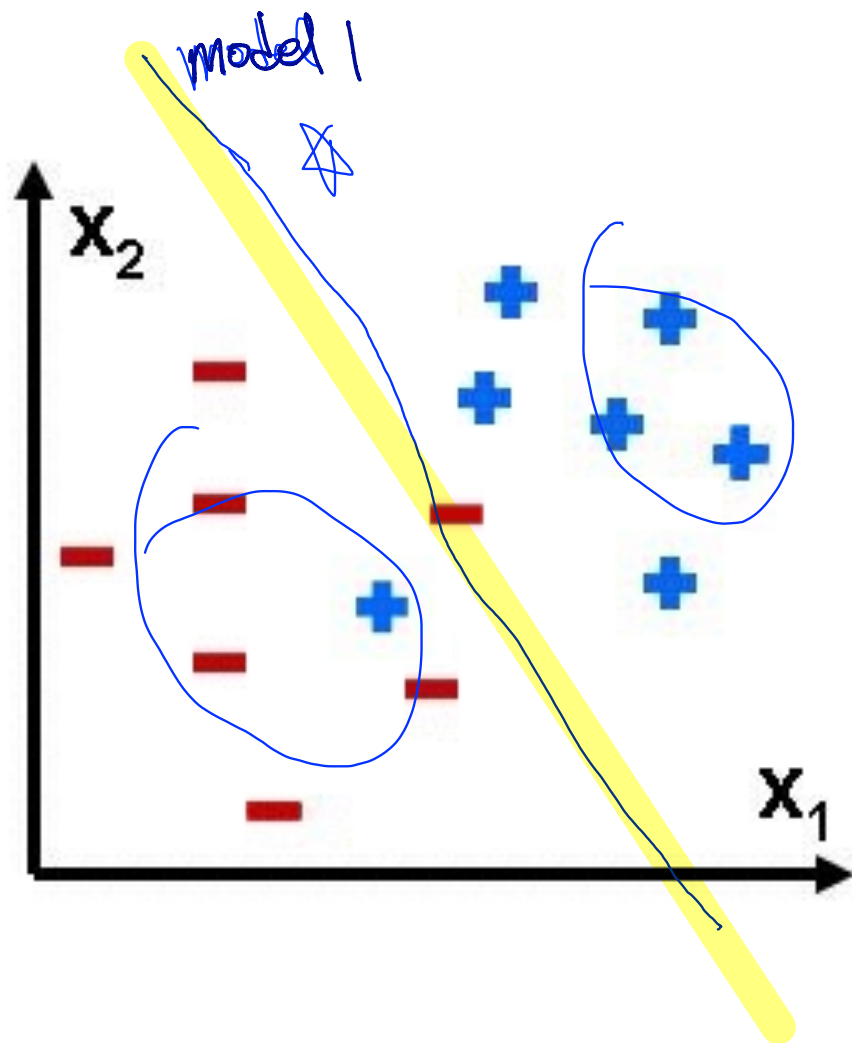
```
x_std[:,0] = (X[:,0] - X[:,0].mean()) / X[:,0].std()
```

http://sebastianraschka.com/Articles/2015_singlelayer_neurons.html

Overfitting

- Our model is very good with training data set (with memorization)
- Not good at test dataset or in real use

Overfitting

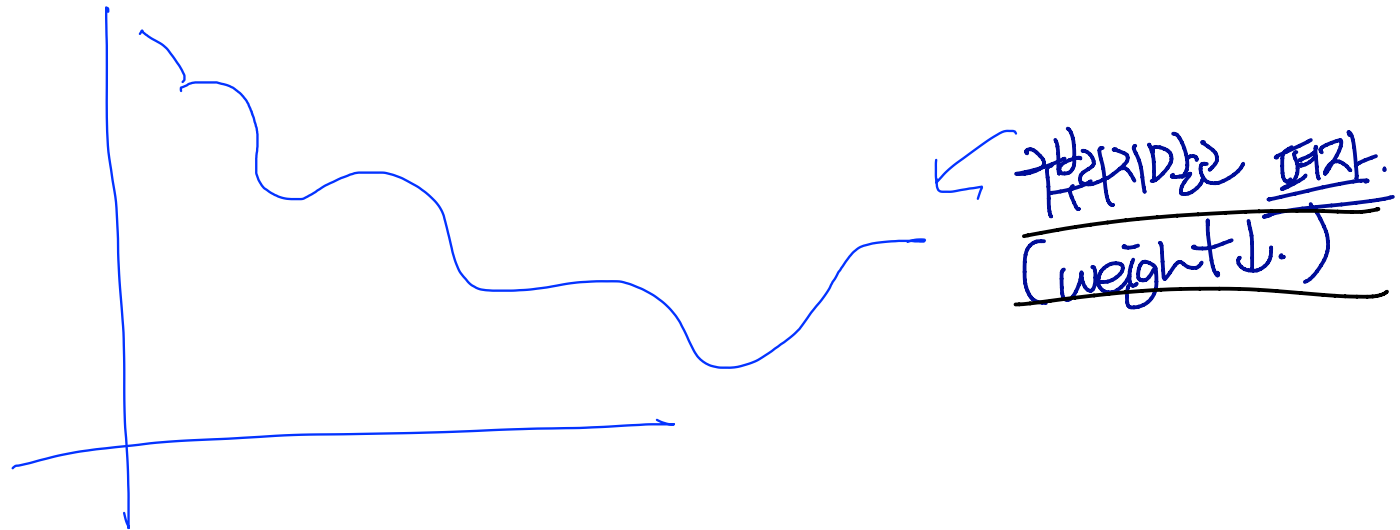


Solutions for overfitting

- More training data!
- Reduce the number of features
- Regularization 제한.

Regularization

- Let's not have too big numbers in the weight Weight은 너무 커지지 X.



Regularization

유한한 파라미터가 hypothesis가
기여할 수 있도록 하는 것.

- Let's not have too big numbers in the weight

LOSS

$$\mathcal{L} = \frac{1}{N} \sum_i \mathcal{D}(s(w x_i + b), L_i) + \lambda \sum W^2$$

TRAINING SET

regularization strength

각 weight w^2 합

term

0 \nearrow X

1 \nearrow 증가하기

0.001 \nearrow 감소하기

$$\text{new } \underbrace{C}_{\downarrow \text{minimize.}} = \underbrace{C_0}_{\downarrow} + \underbrace{\lambda \sum W^2}_{\downarrow} \Rightarrow \text{weight decay.}$$

$\lambda \sum W^2 \Rightarrow$ λ is \downarrow \Rightarrow W is \downarrow \Rightarrow λ is \downarrow \Rightarrow W is \downarrow

λ is \downarrow \Rightarrow W is \downarrow

λ is \downarrow \Rightarrow W is \downarrow

Regularization

- Let's not have too big numbers in the weight

LOSS


`l2reg = 0.001 * tf.reduce_sum(tf.square(W))`

regularization strength

$$\mathcal{L} = \frac{1}{N} \sum_i \mathcal{D}(s(w x_i + b), L_i) + \lambda \sum W^2$$

TRAINING SET

Summary

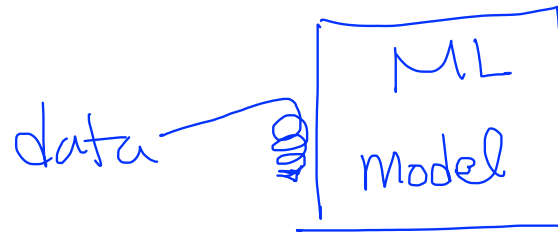
- Learning rate ✓
 - Data preprocessing
 - Overfitting ✕
 - More training data
 - Regularization ?
- 
- A handwritten blue bracket is drawn to the right of the 'Overfitting' section, spanning from the 'Overfitting' bullet point down to the 'Regularization ?' sub-bullet. An arrow points from the bottom of the bracket towards the 'Regularization ?' text.

Lecture 7-2

Application & Tips: Learning and test data sets

Sung Kim <hunkim+mr@gmail.com>

Performance evaluation: is this good?



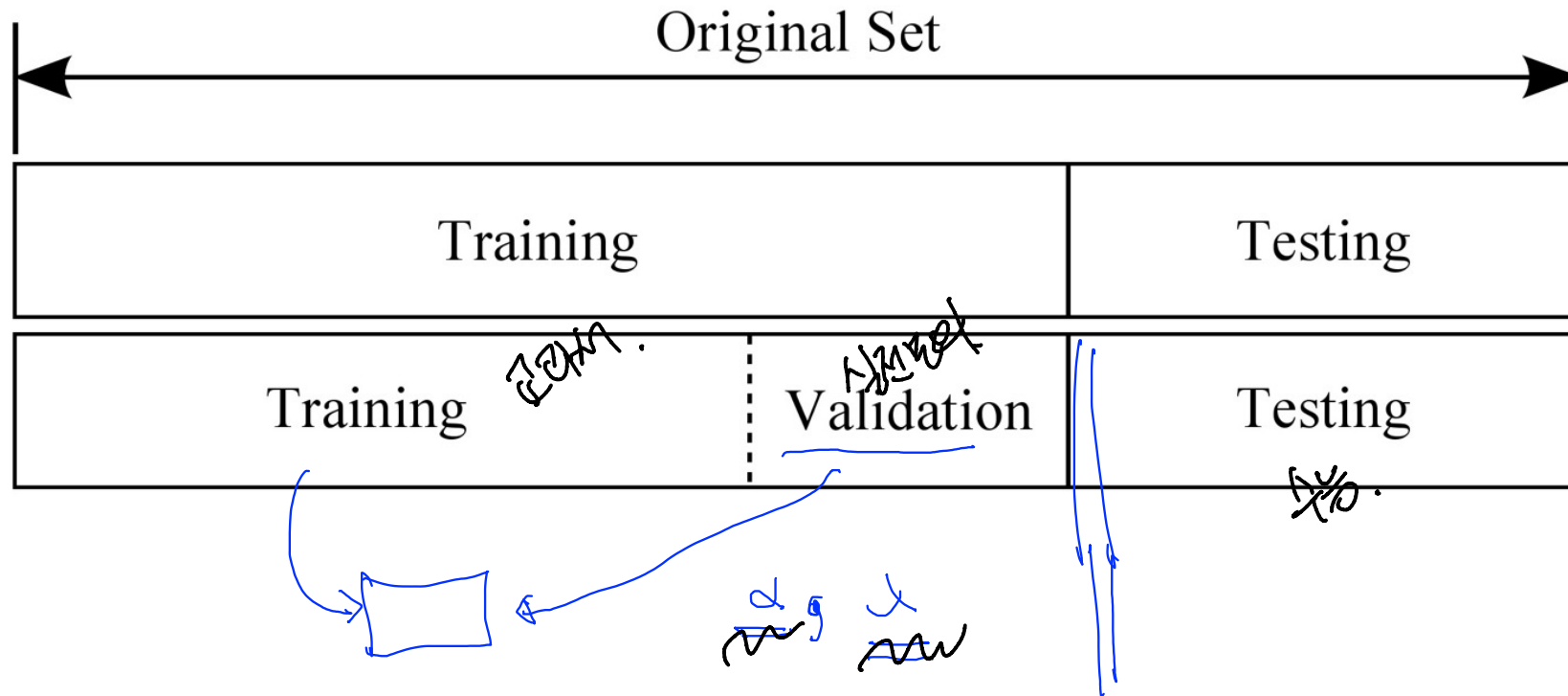
Evaluation using training set?

	Size	Price	
training Set (RM)	2104	400	training set \rightarrow model.
	1600	330	
	2400	369	
	1416	232	
	3000	540	
	1985	300	
	1534	315	
test set (RM)	1427	199	\Rightarrow model. $\hat{y} = y$
	1380	212	
	1494	243	

X. Input: training Data.
traindata(model)

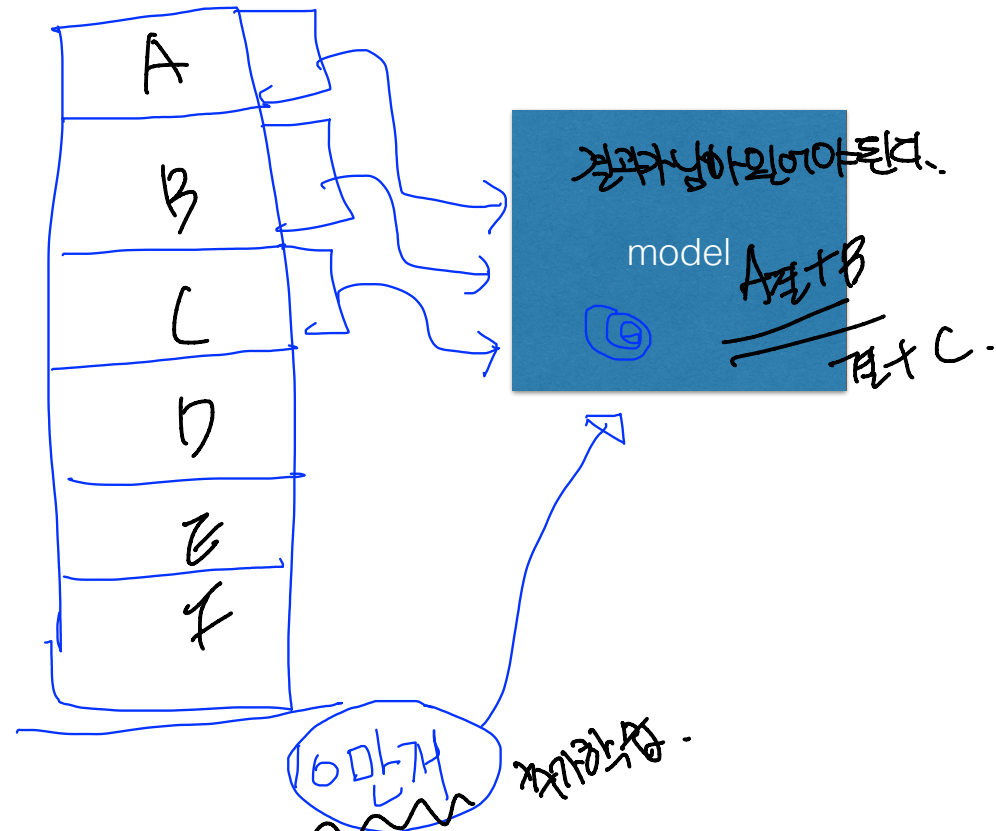
- 100% correct (accuracy)
- Can memorize

Training, validation and test sets



Online learning

100만개



MNIST Dataset

Zip: 03 63



0
1
2

[train-images-idx3-ubyte.gz](#): training set images (9912422 bytes)

[train-labels-idx1-ubyte.gz](#): training set labels (28881 bytes)

[t10k-images-idx3-ubyte.gz](#): test set images (1648877 bytes)

[t10k-labels-idx1-ubyte.gz](#): test set labels (4542 bytes)

<http://yann.lecun.com/exdb/mnist/>

Accuracy

정확도

- How many of your predictions are correct?

- 95% ~ 99%?

- Check out the lab video

Y
=

