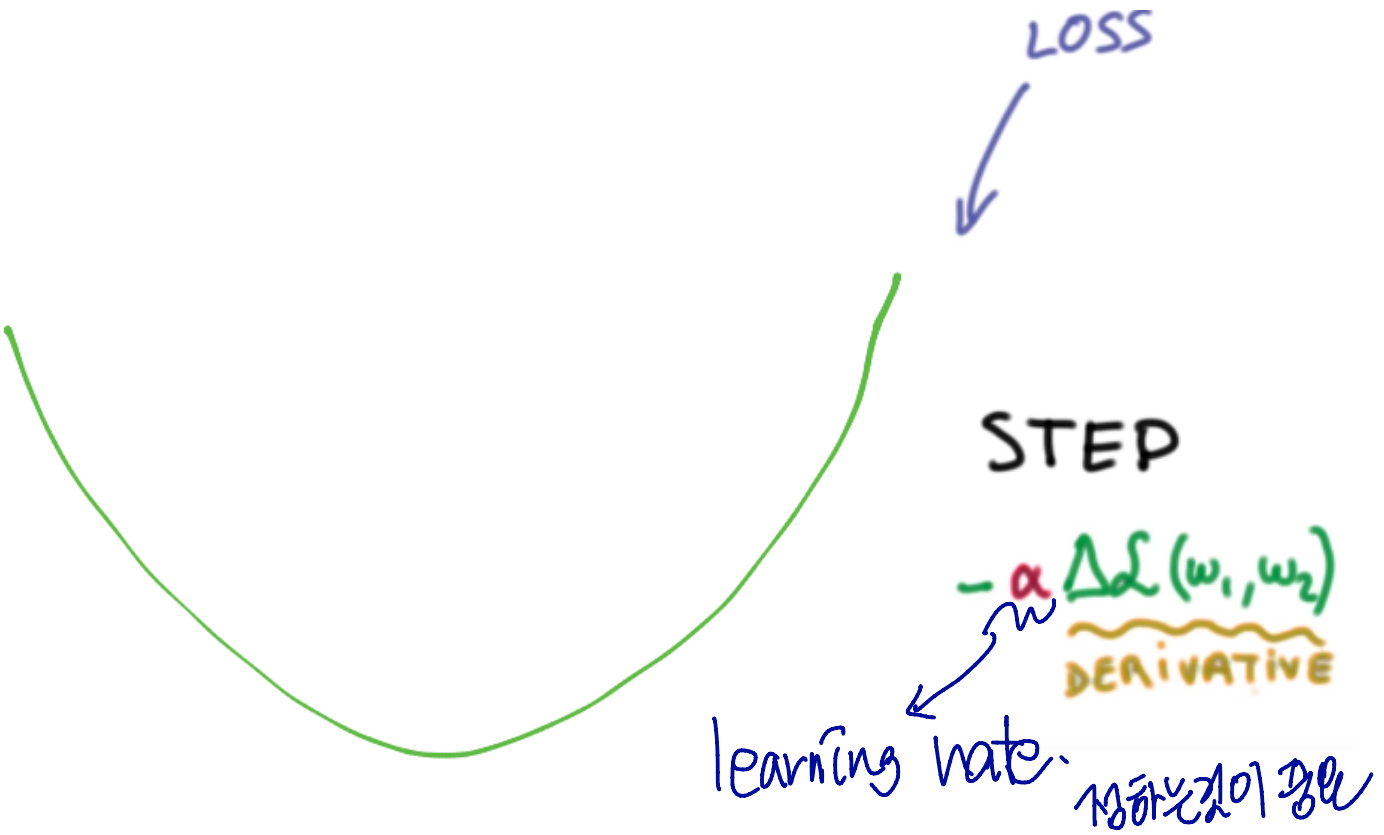
Lecture 7-1

Application & Tips:
Learning rate, data preprocessing, overfitting

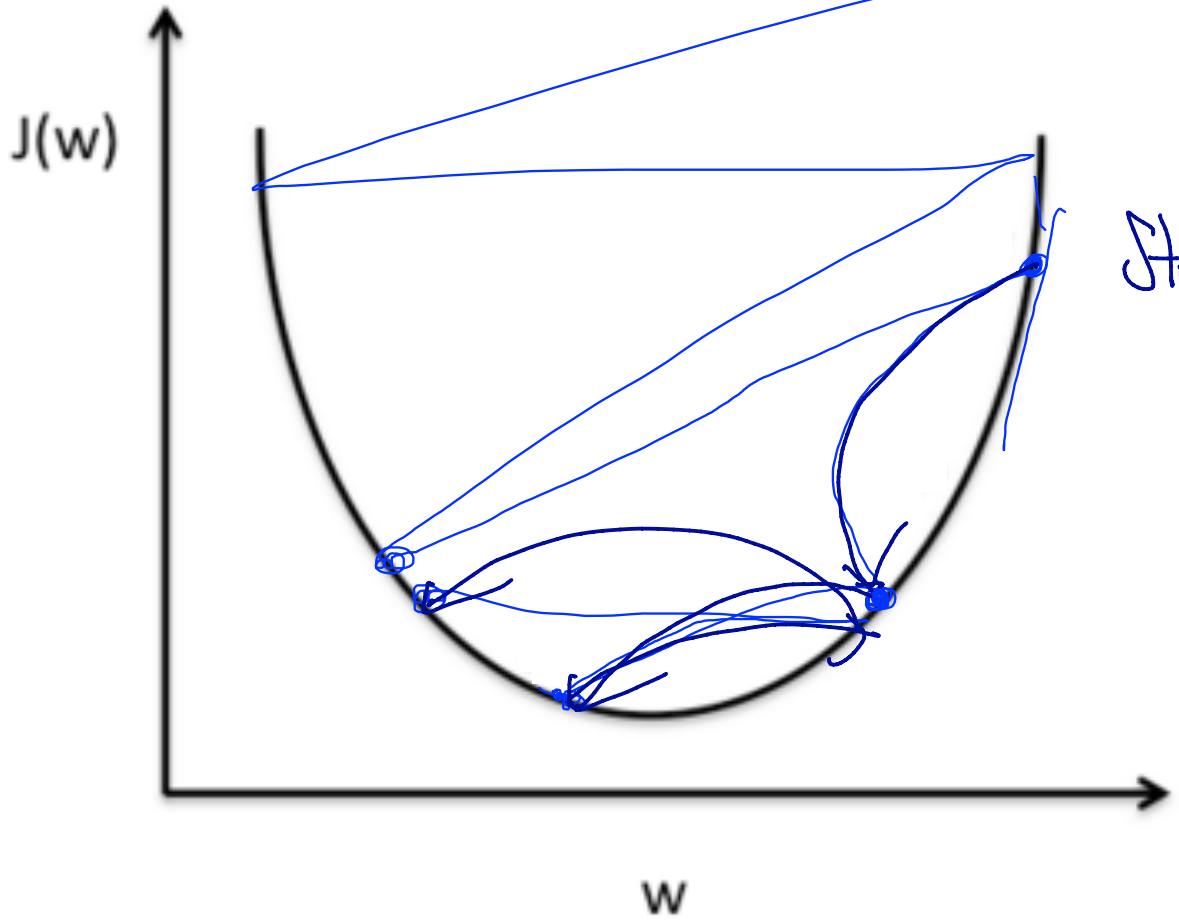
Sung Kim <hunkim+mr@gmail.com>

```python
# 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
```



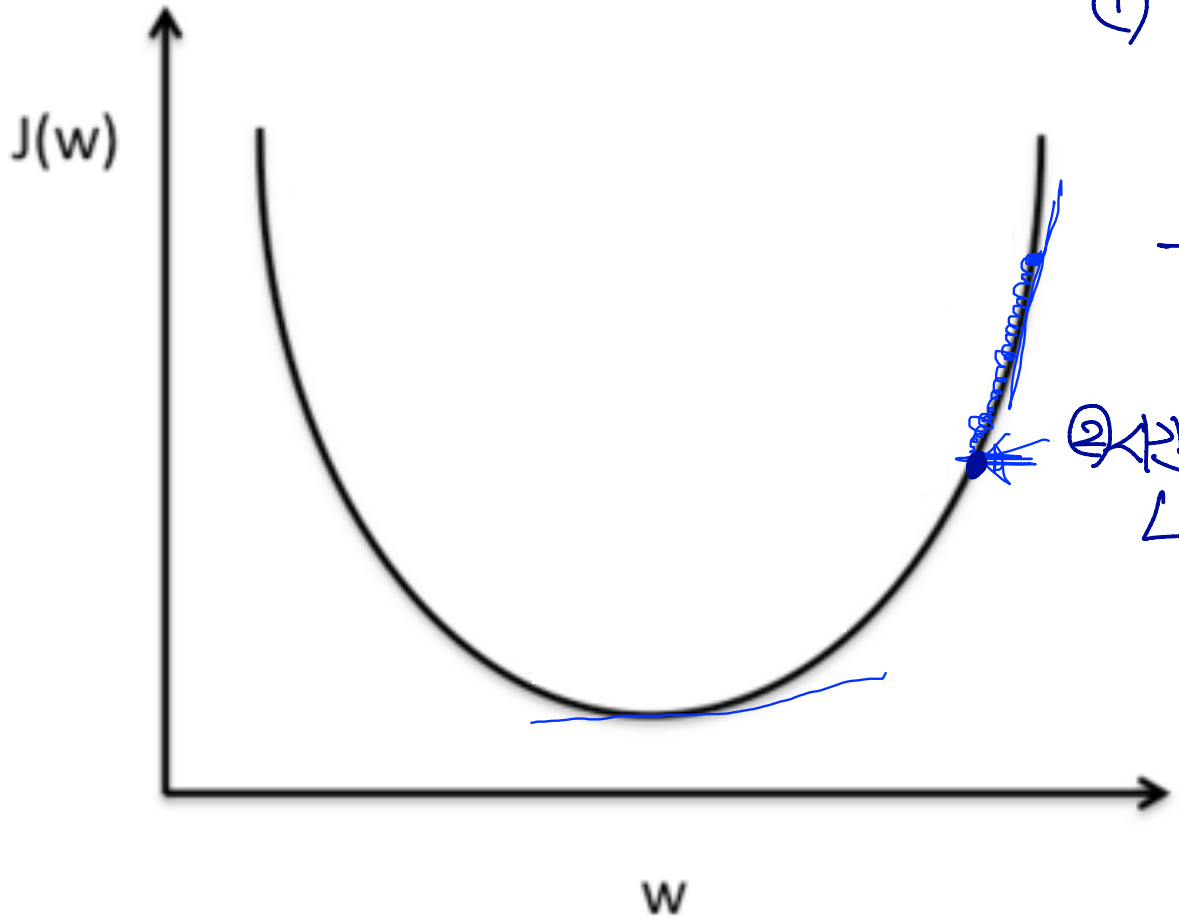
https://www.udacity.com/course/viewer#!/c-ud730/l-6370362152/m-6379811827

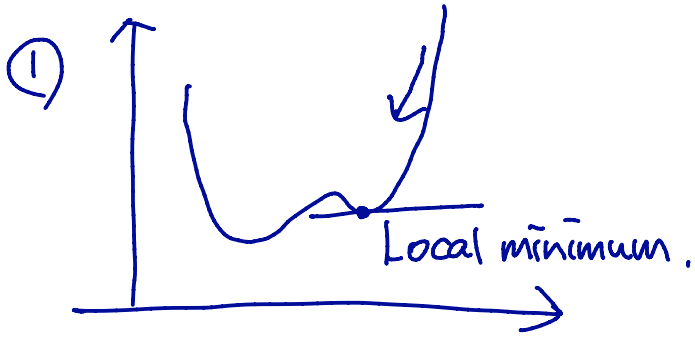
Large learning rate: overshooting



Small learning rate:

takes too long, stops at local minimum



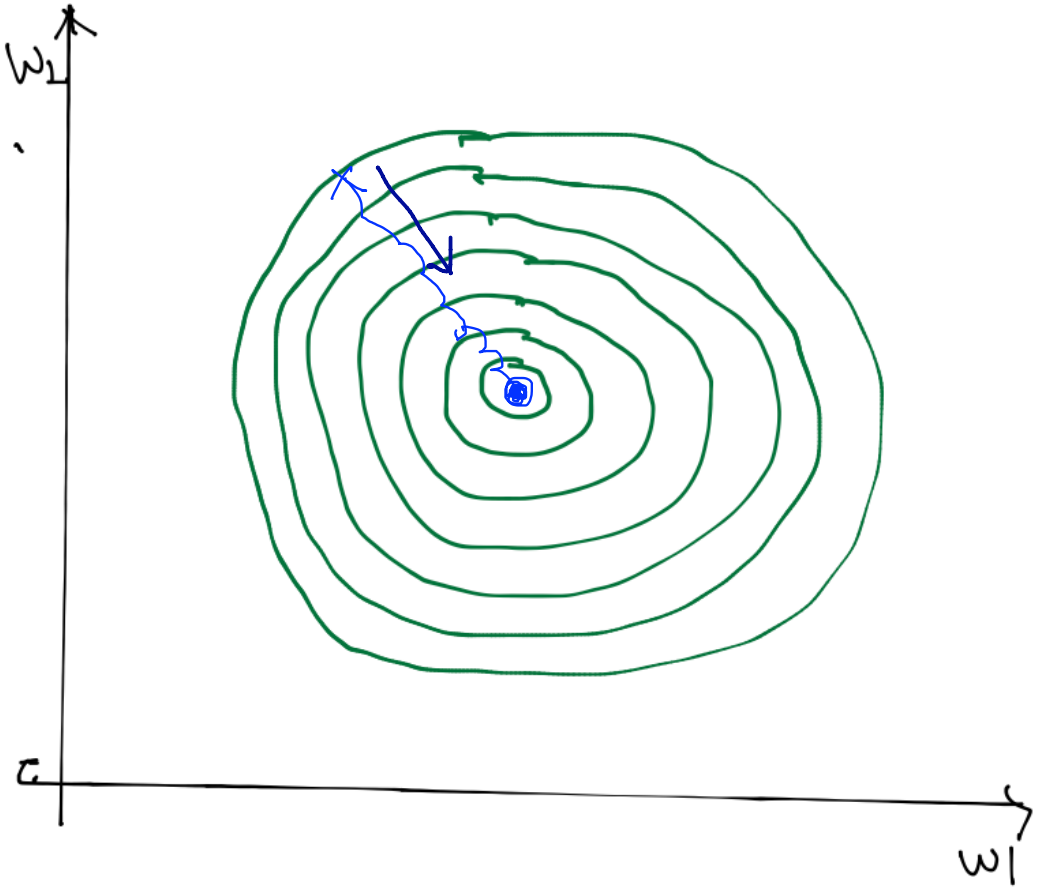


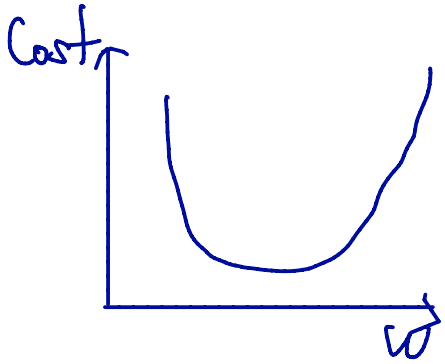
②
Local minimum 아님.

Try several learning rates

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

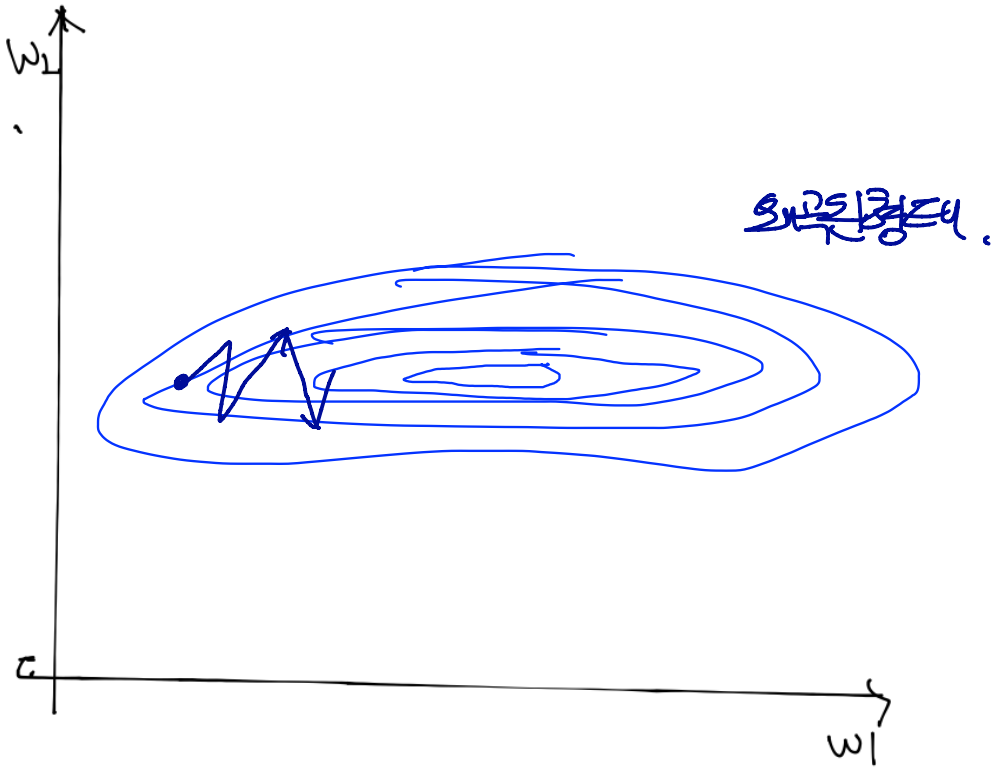
Data (X) preprocessing for gradient descent





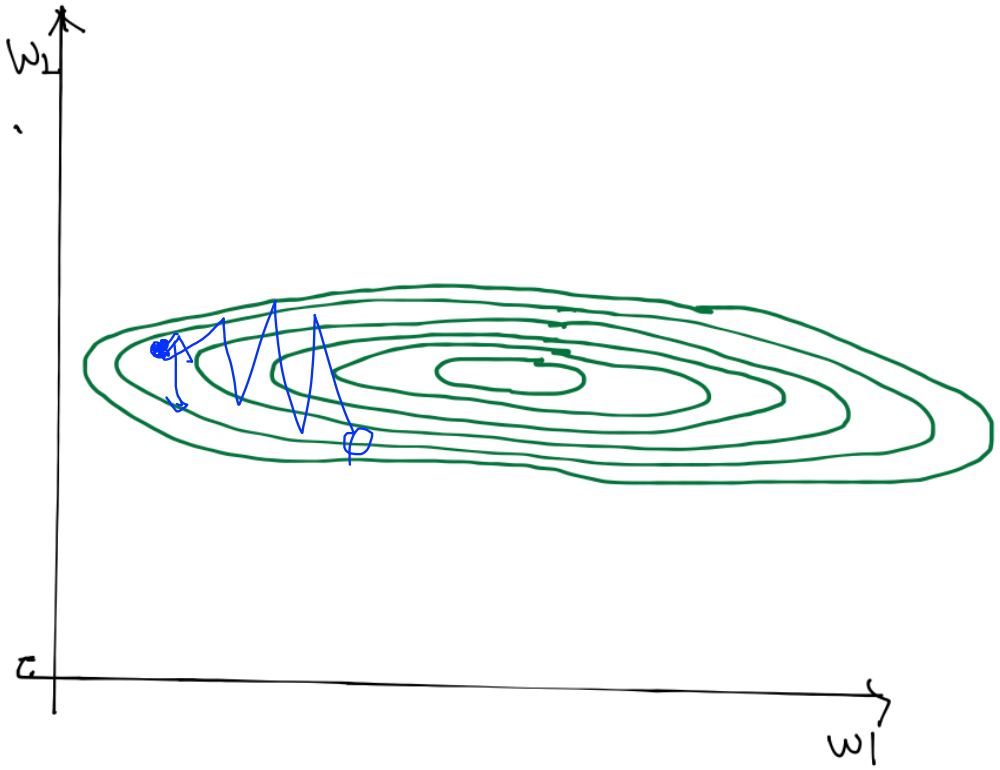
Data (X) preprocessing for gradient descent

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

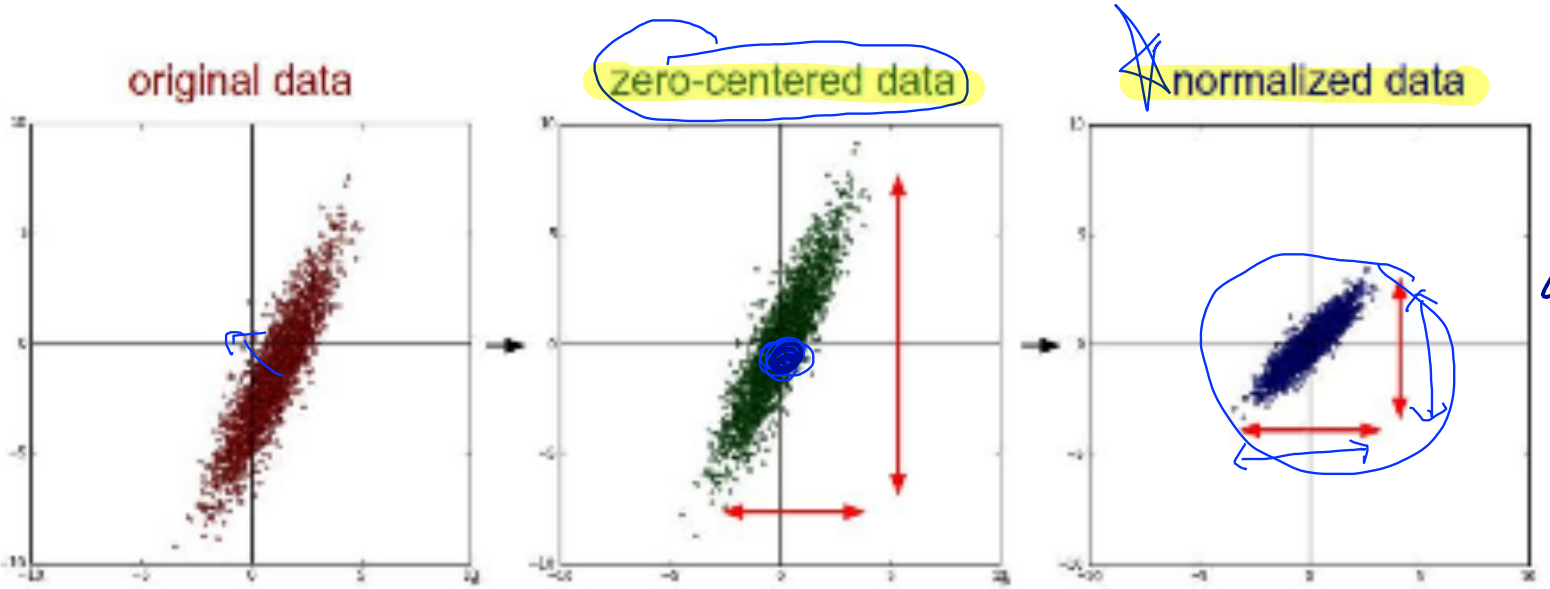


Data (X) preprocessing for gradient descent

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



Data (X) preprocessing for gradient descent



Standardization

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

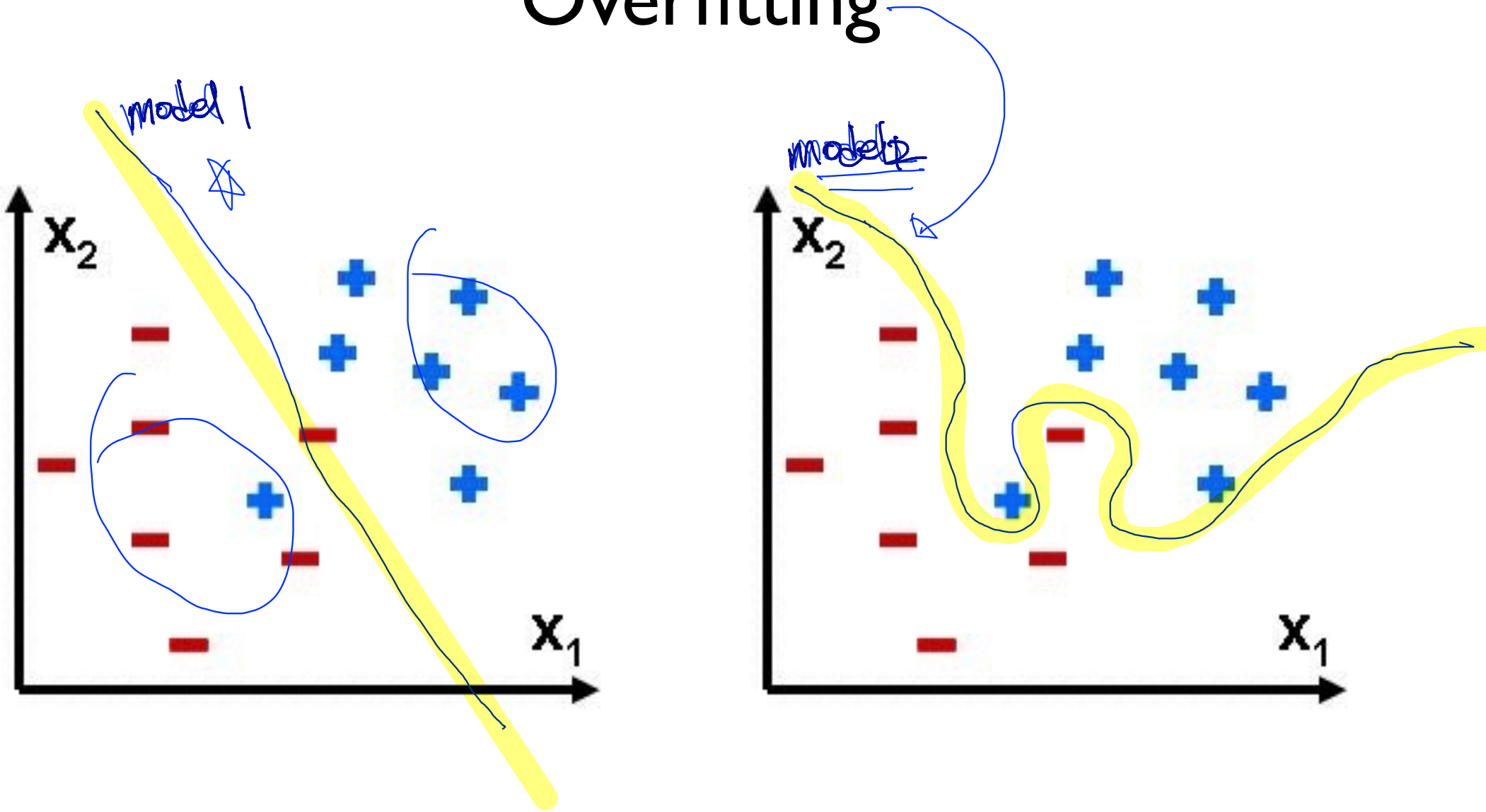
```
X_std[:,0] = (X[:,0] - X[:,0].mean()) / X[:,0].std()
```

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

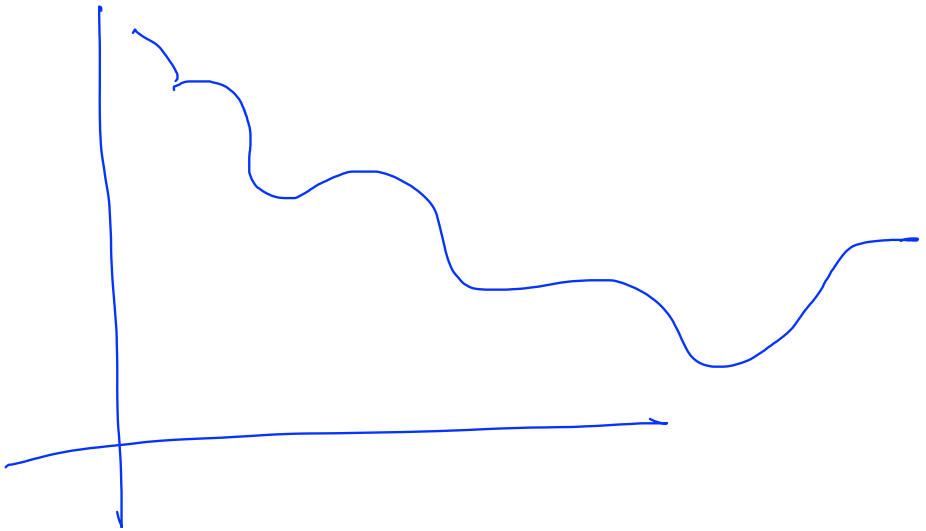


Solutions for overfitting

- More training data!
- Reduce the number of features
- **Regularization**

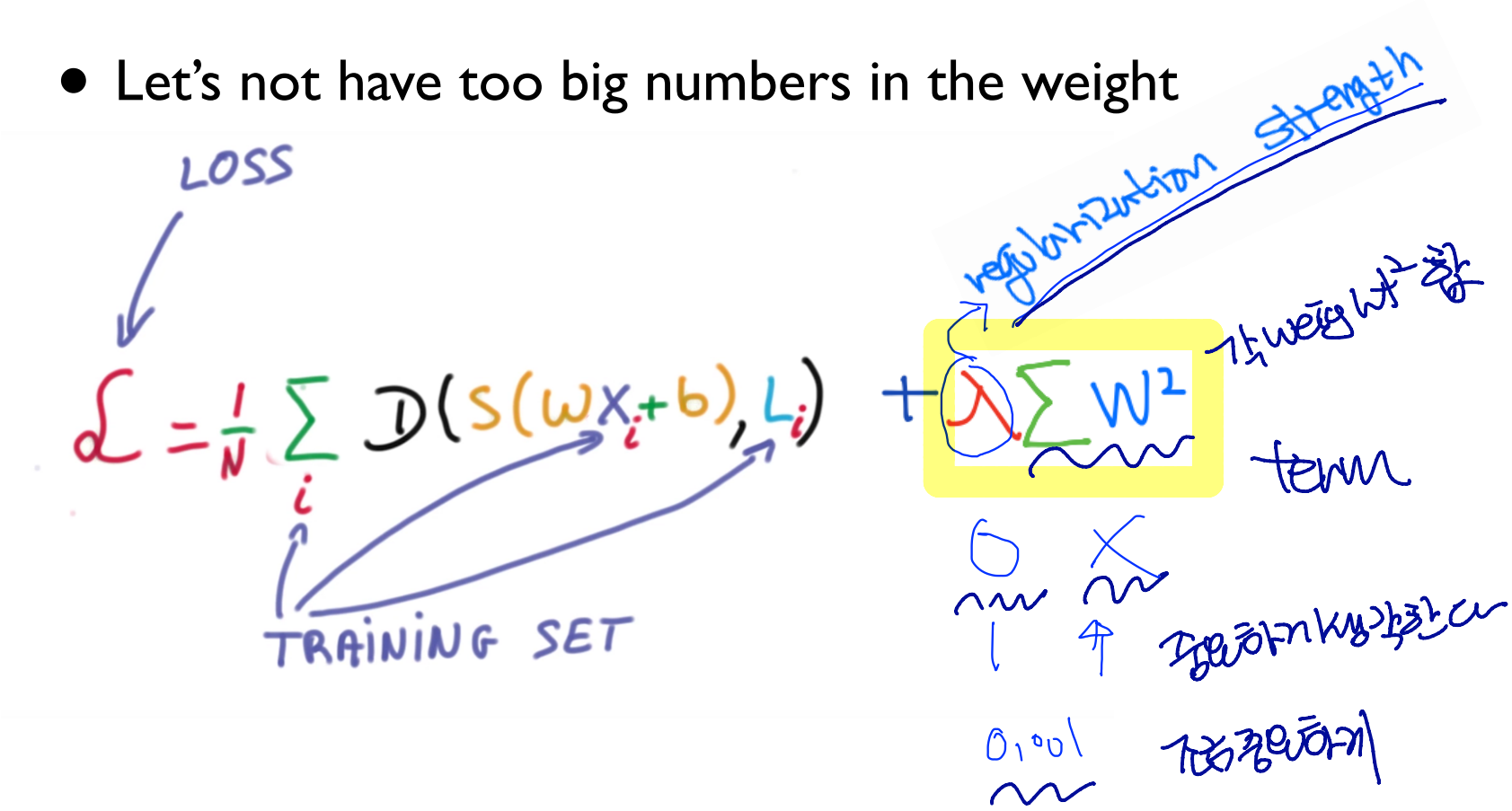
Regularization

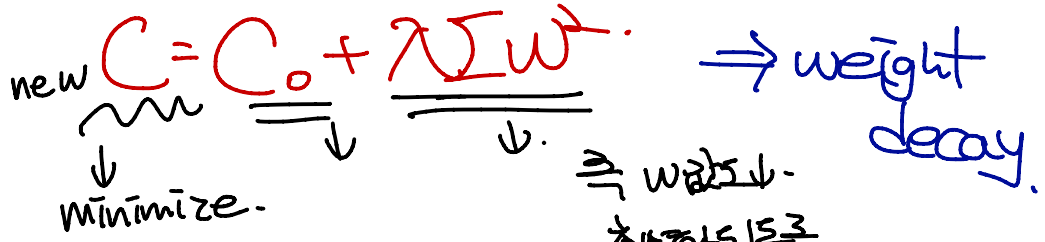
- Let's not have too big numbers in the weight



Regularization

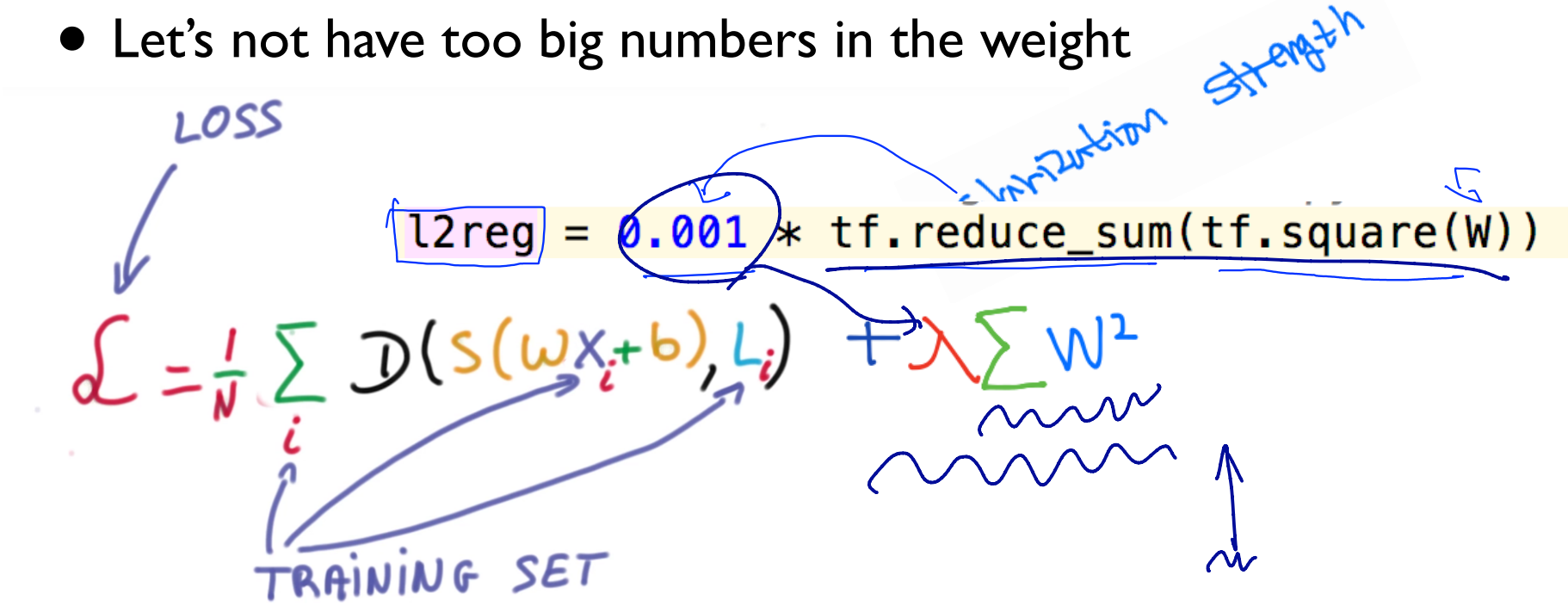
- Let's not have too big numbers in the weight





Regularization

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



Summary

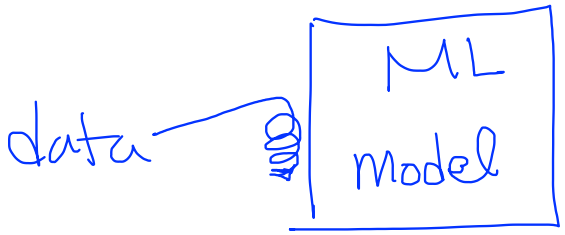
- Learning rate
- Data preprocessing
- Overfitting
  - More training data
  - Regularization

Lecture 7-2

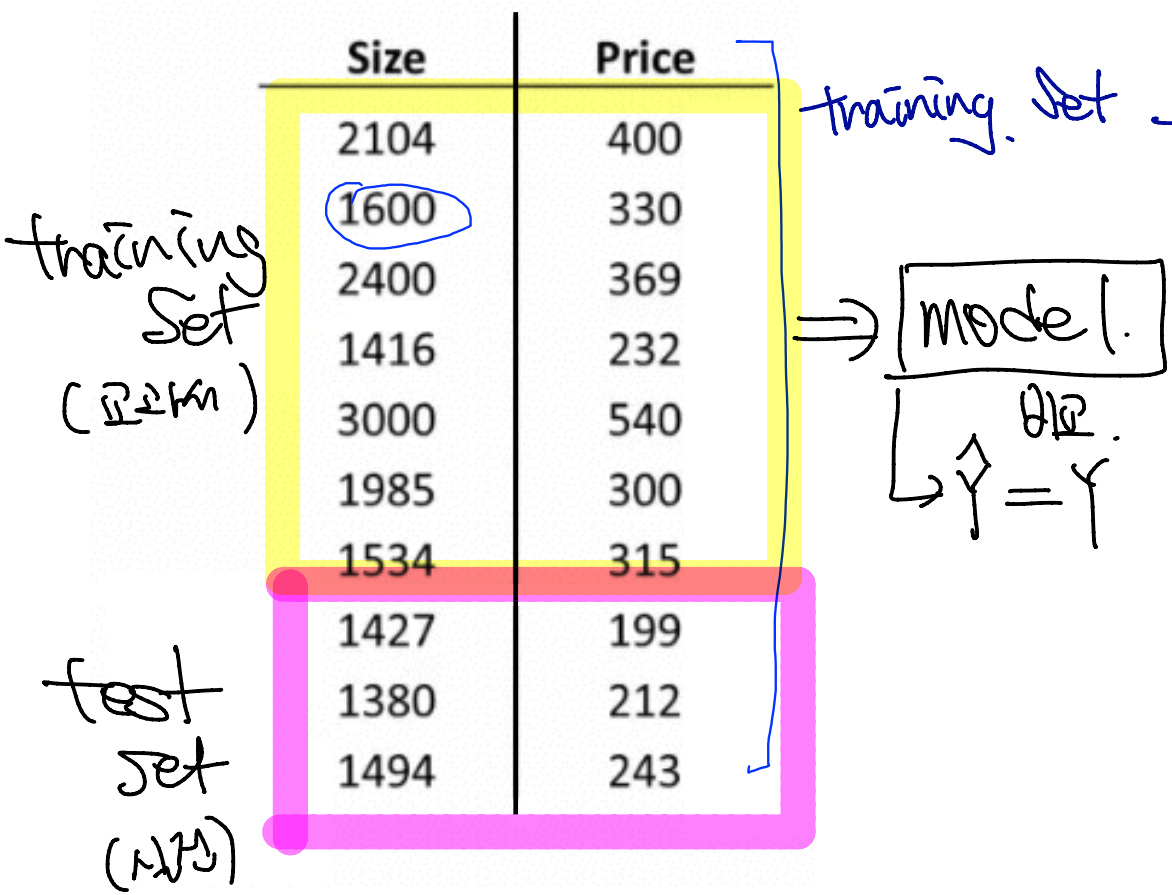
Application & Tips:
Learning and test data sets

Sung Kim <hunkim+mr@gmail.com>

Performance evaluation: is this good?



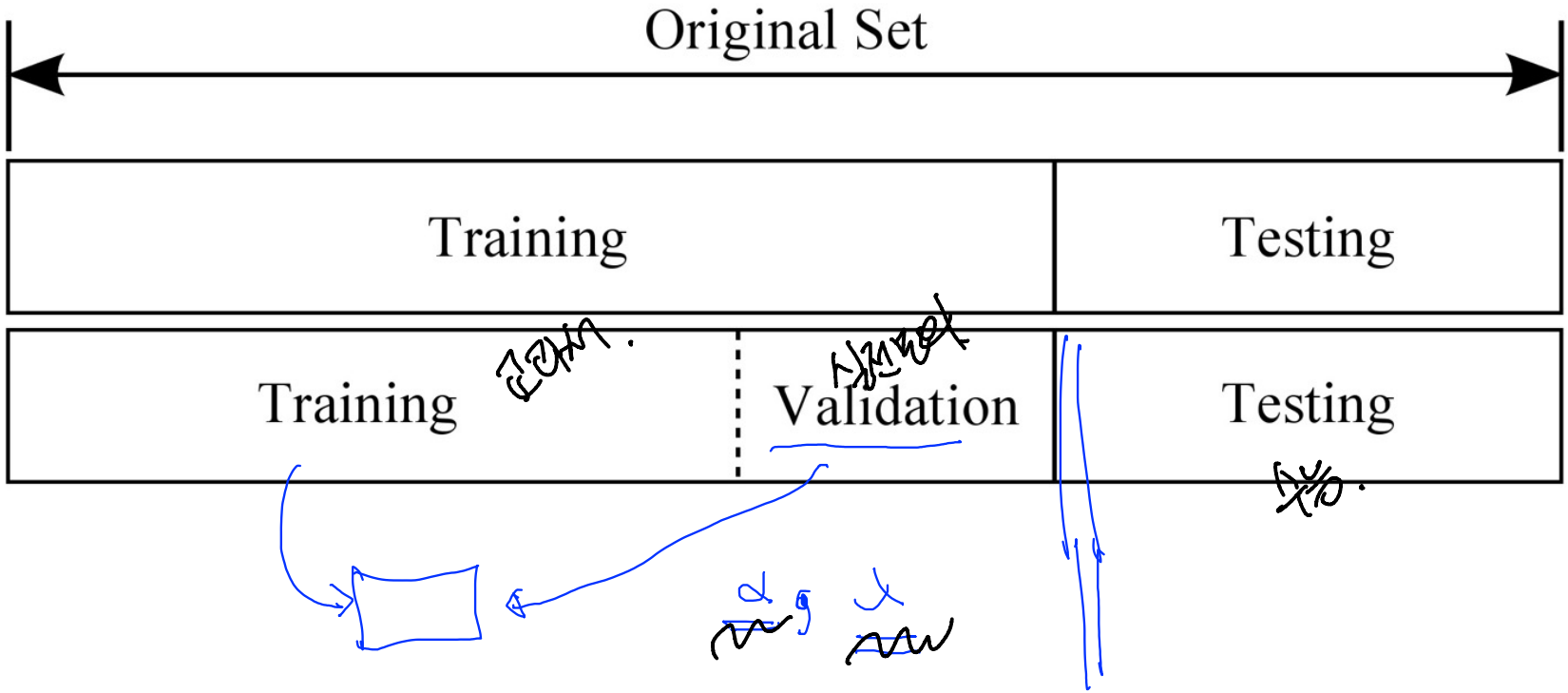
Evaluation using training set?



X. Input : training Data.
traindata (model)

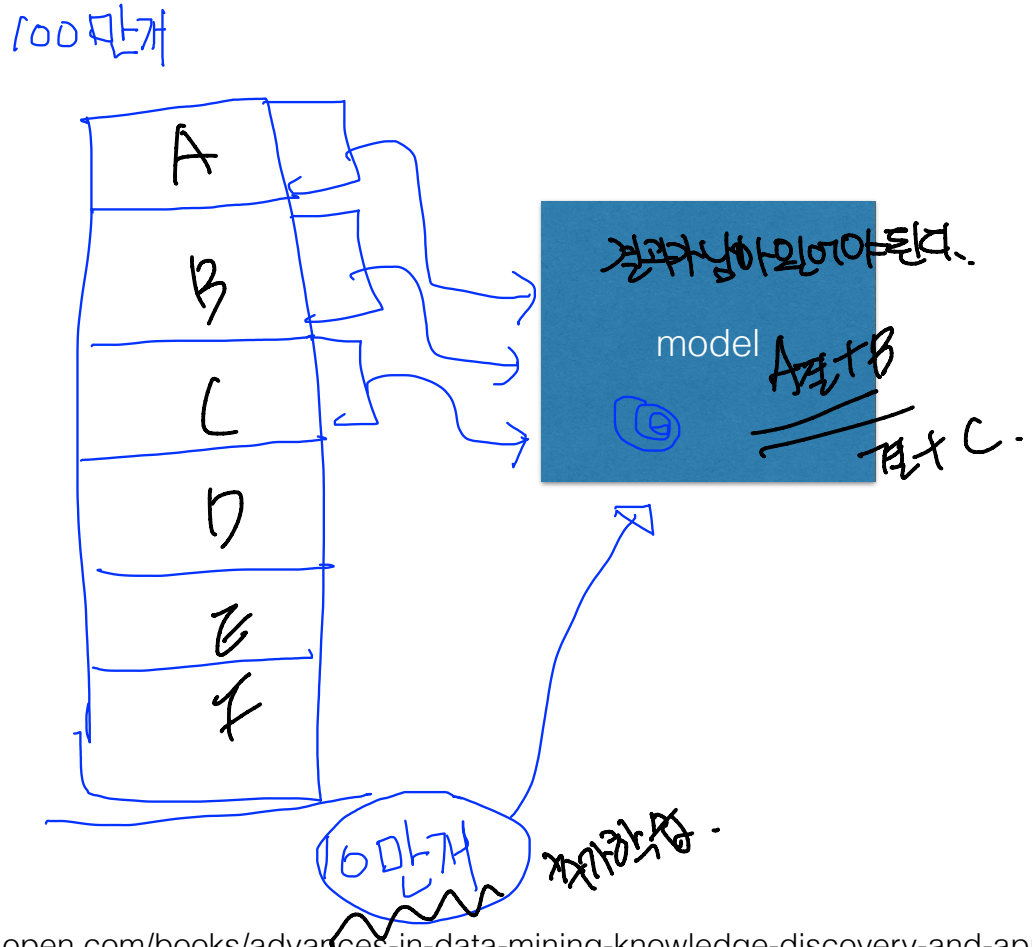
- 100% correct (accuracy)
- Can memorize

Training, validation and test sets



http://www.intechopen.com/books/advances-in-data-mining-knowledge-discovery-and-applications/selecting-representative-data-sets

Online learning



http://www.intechopen.com/books/advances-in-data-mining-knowledge-discovery-and-applications/selecting-representative-data-sets

MNIST Dataset

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

- How many of your predictions are correct?
- 95% ~ 99%?
- Check out the lab video