

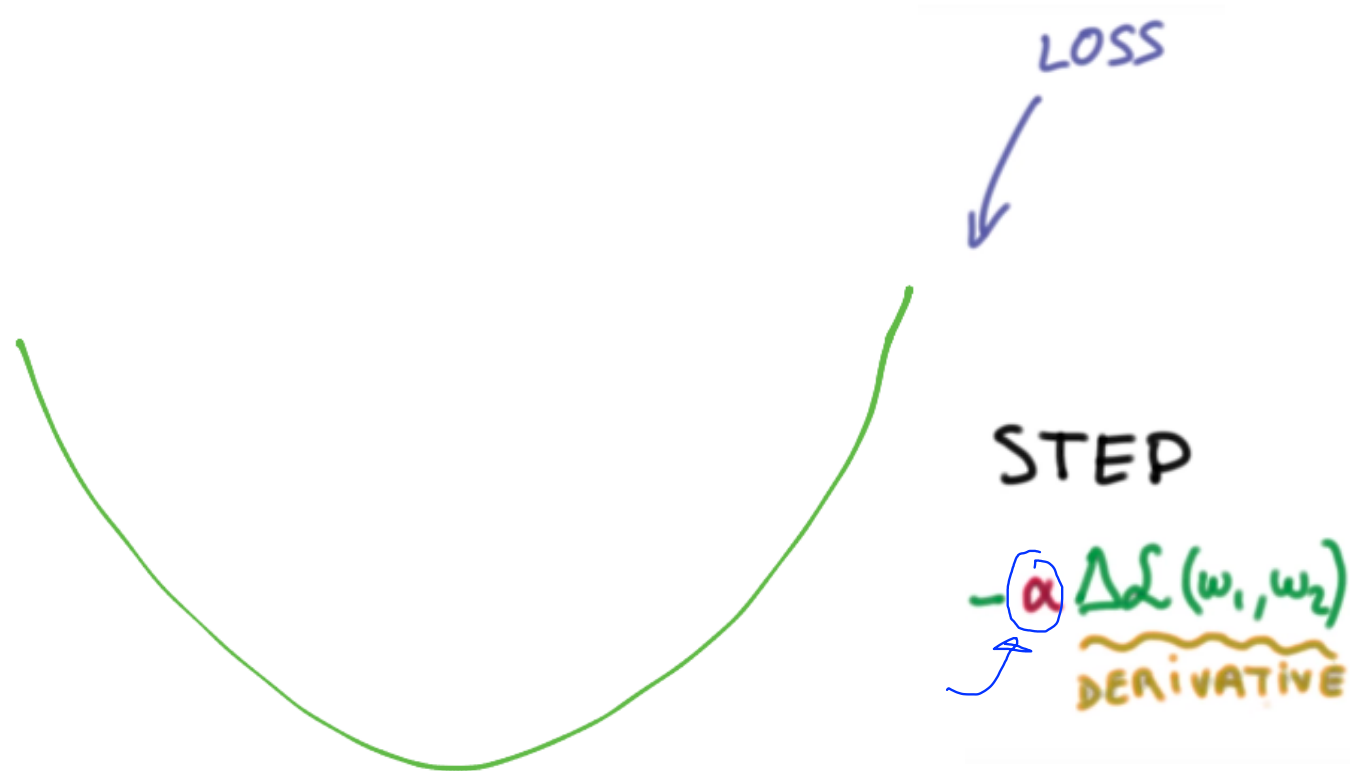
Lecture 7-I

Application & Tips:

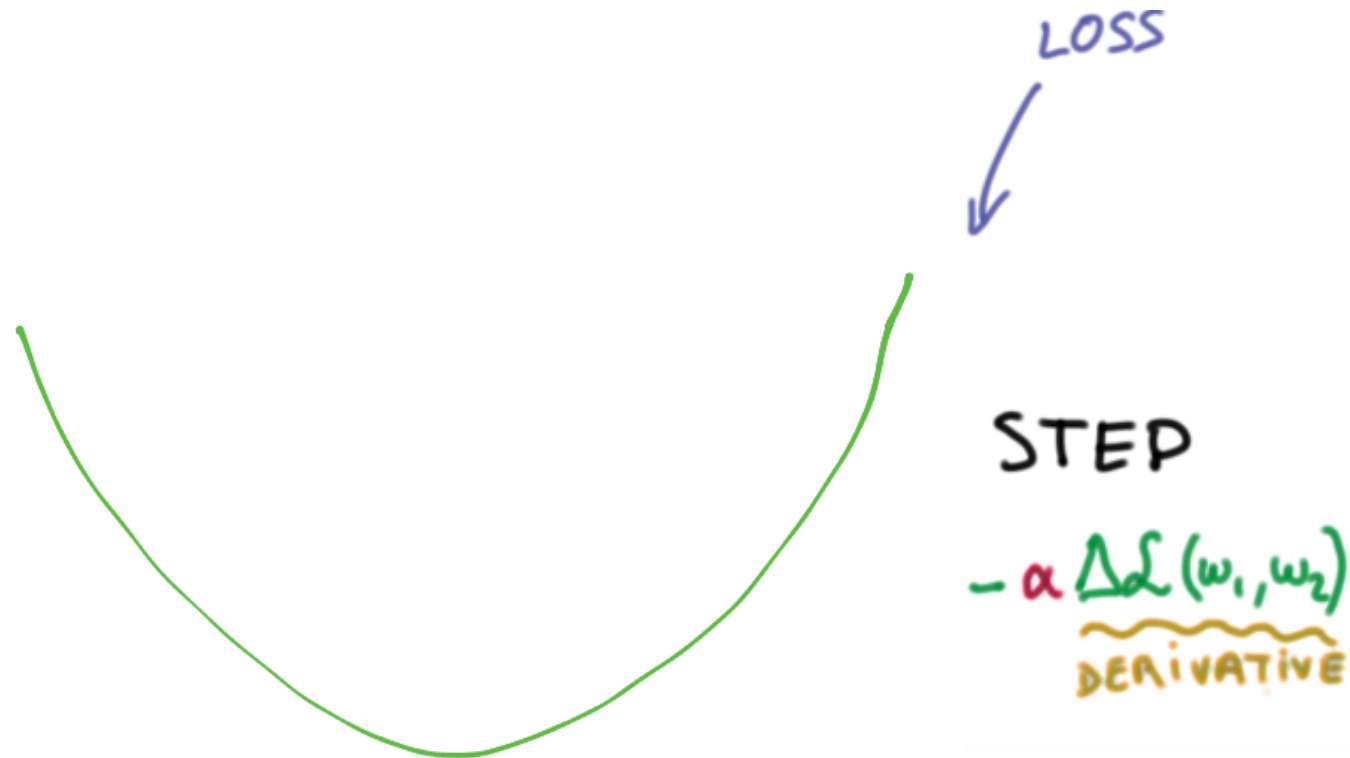
Learning rate, data preprocessing,  overfitting

Sung Kim <hunkim+mr@gmail.com>

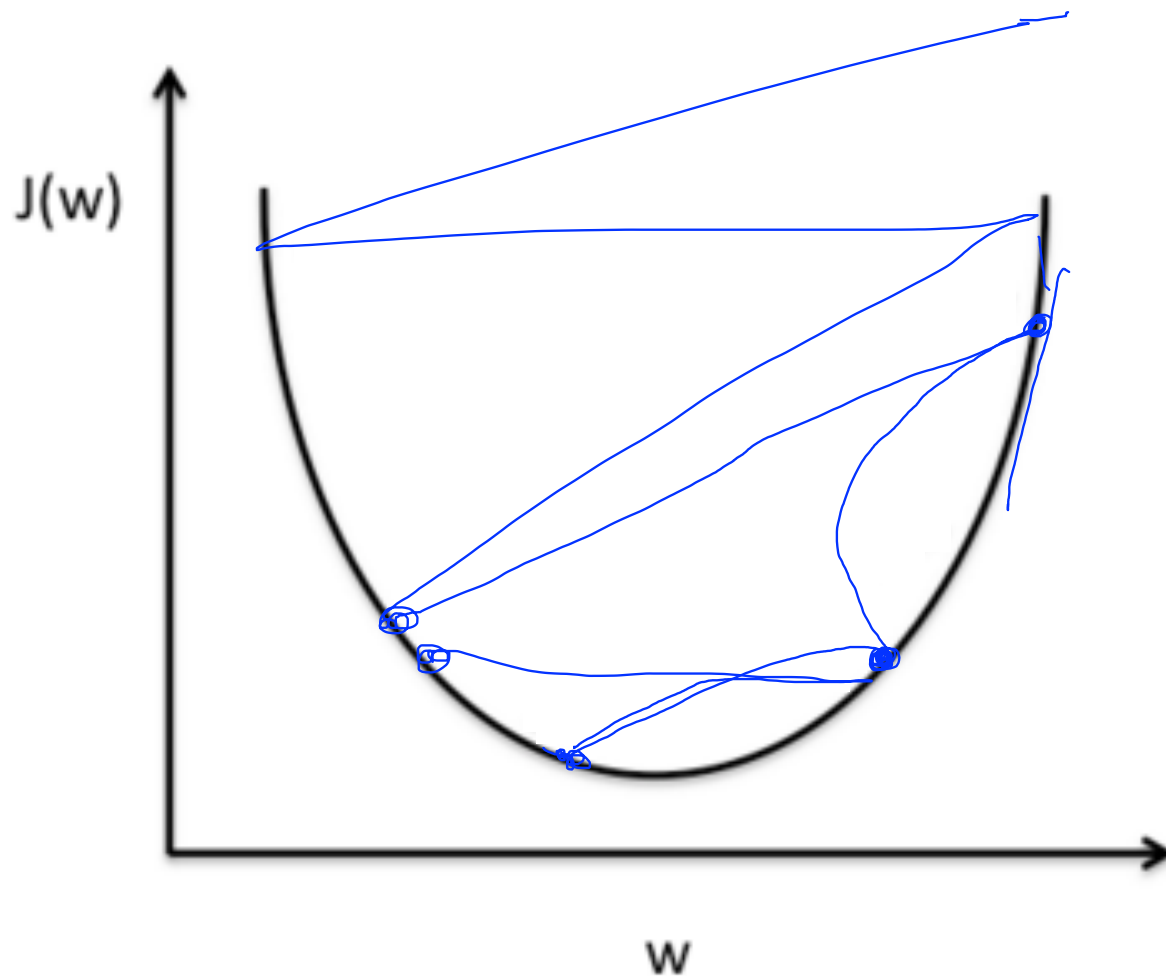
Gradient descent



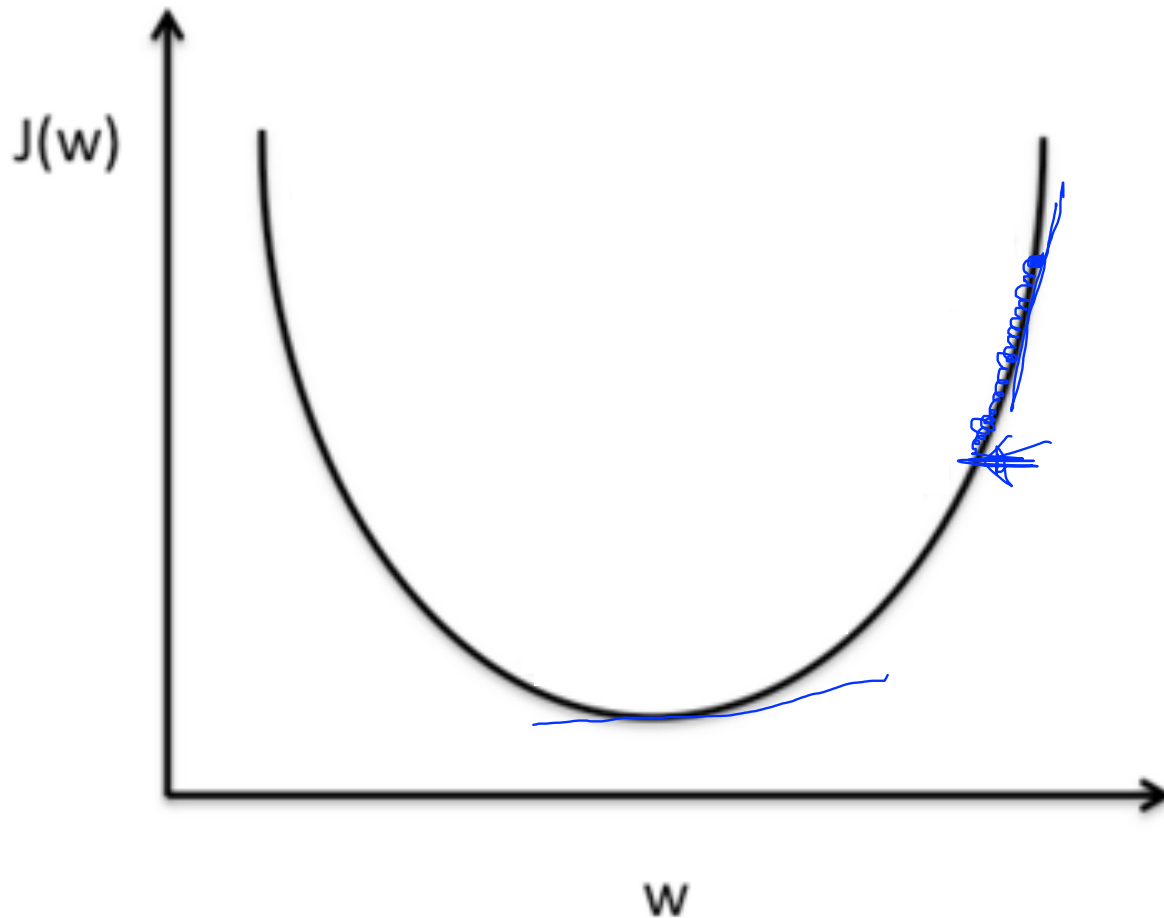
```
# Minimize error using cross entropy
learning_rate = 0.001
cost = tf.reduce_mean(-tf.reduce_sum(Y*tf.log(hypothesis), reduction_indices=1)) # Cross entropy
optimizer = tf.train.GradientDescentOptimizer(learning_rate).minimize(cost) # Gradient Descent
```



Large learning rate: overshooting



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

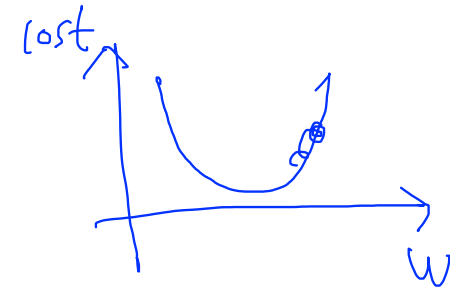
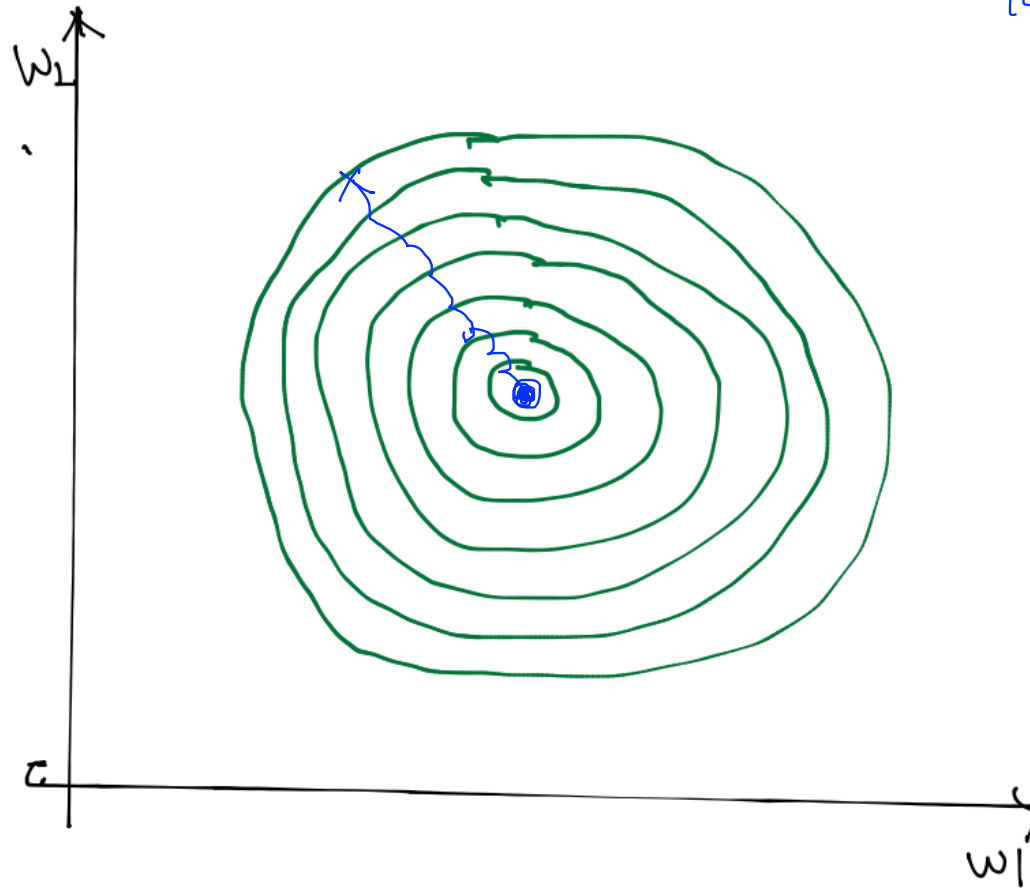


Try several learning rates

0.01

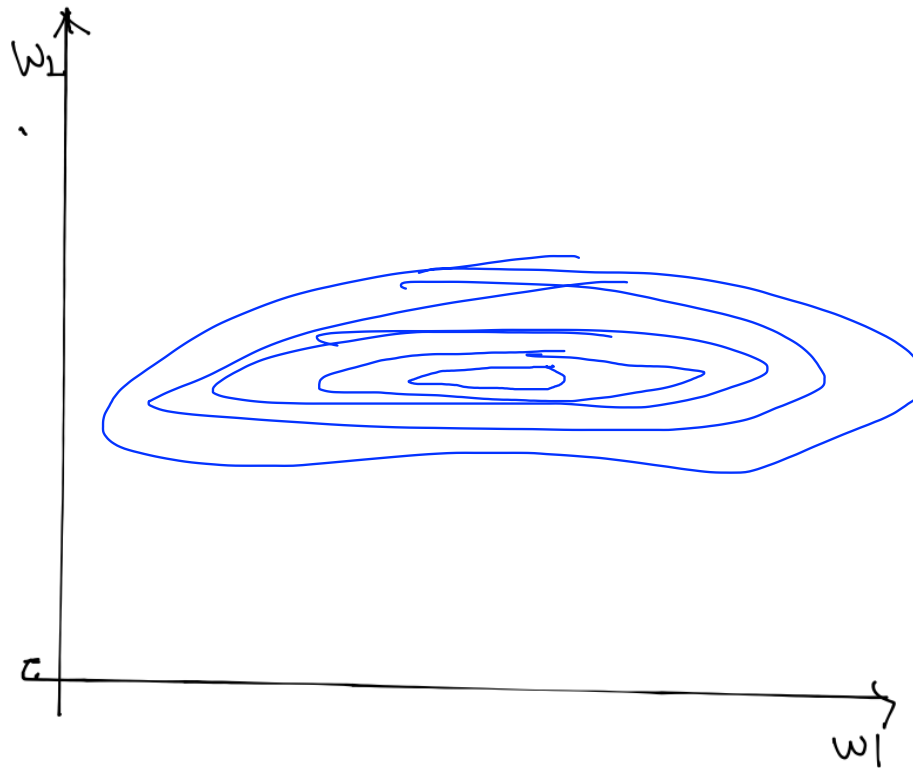
- Observe the cost function
- Check it goes down in a reasonable rate

Data (X) preprocessing for gradient descent



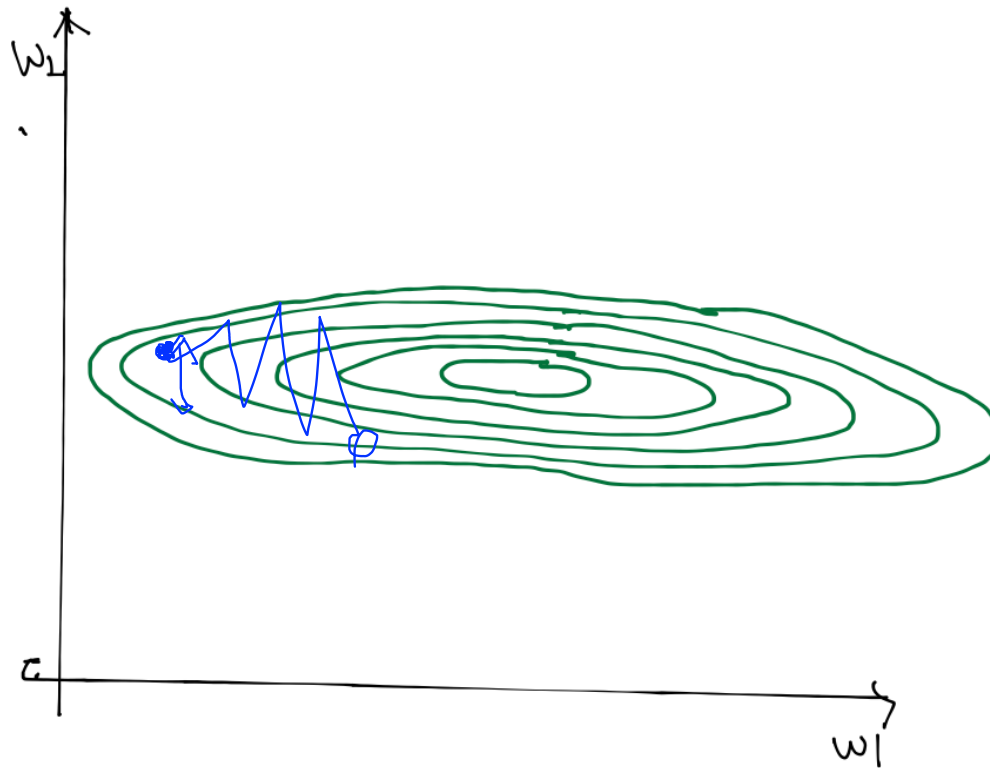
Data (X) preprocessing for gradient descent

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

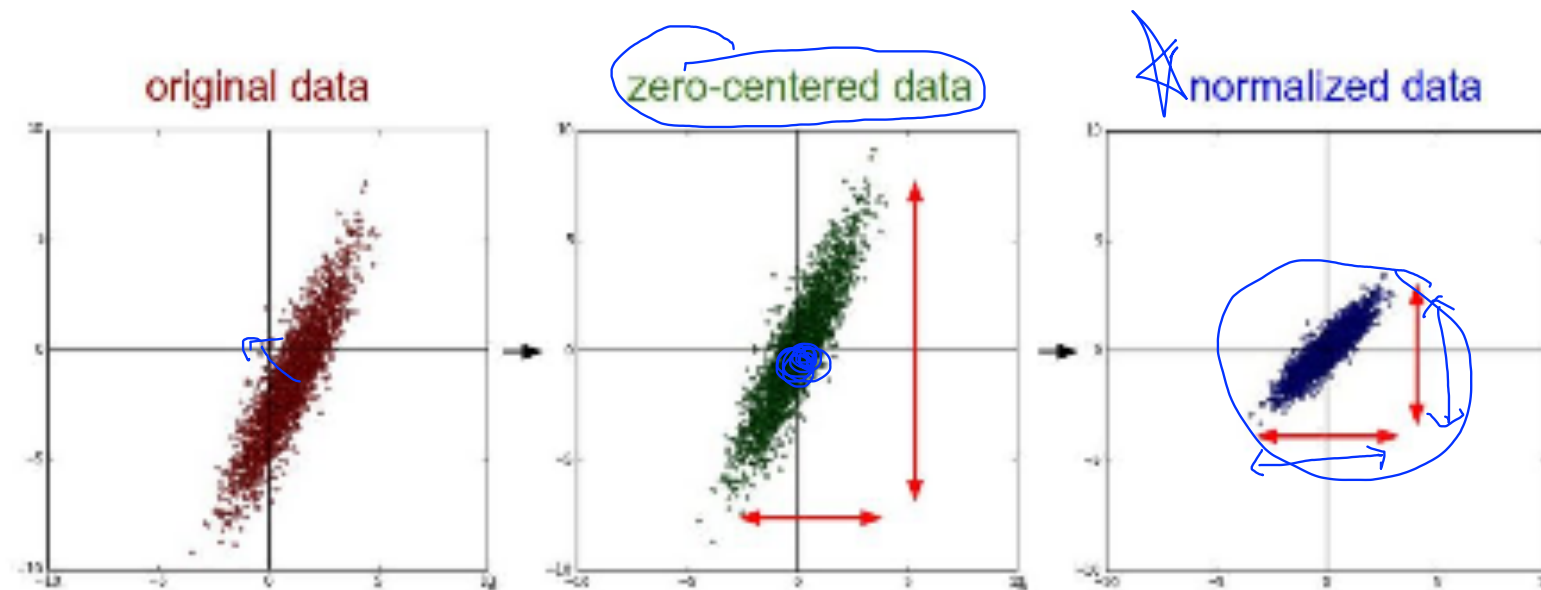


Data (X) preprocessing for gradient descent

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



Data (X) preprocessing for gradient descent



Standardization

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

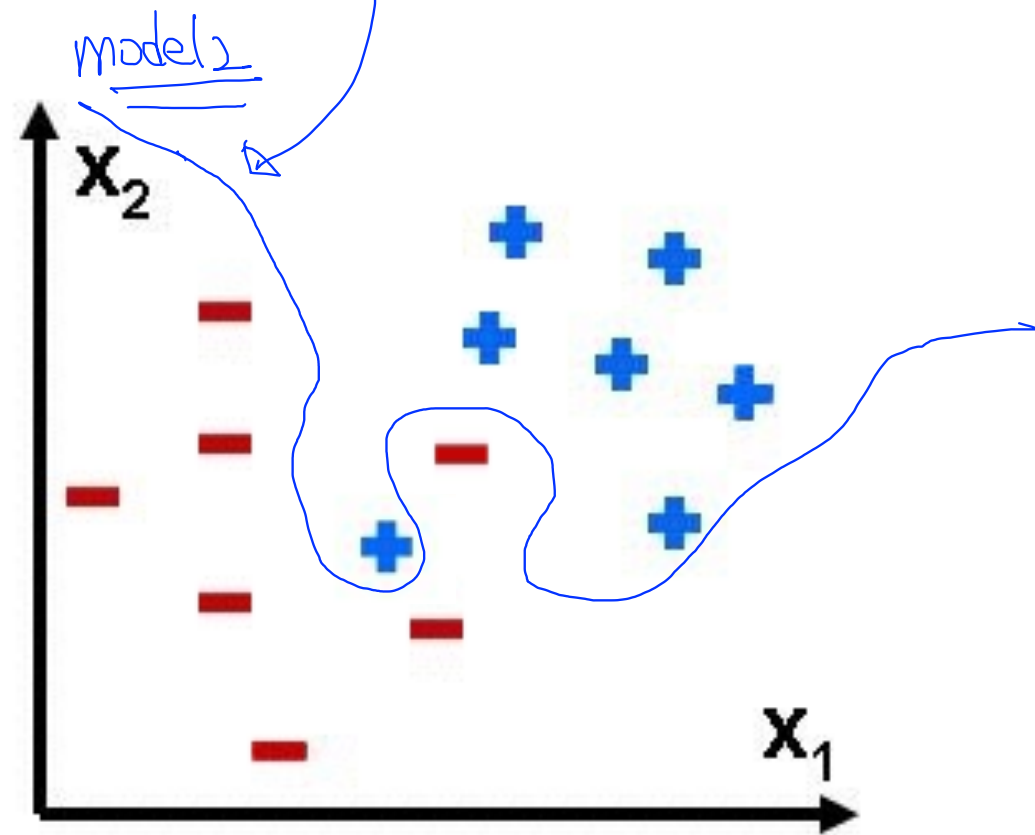
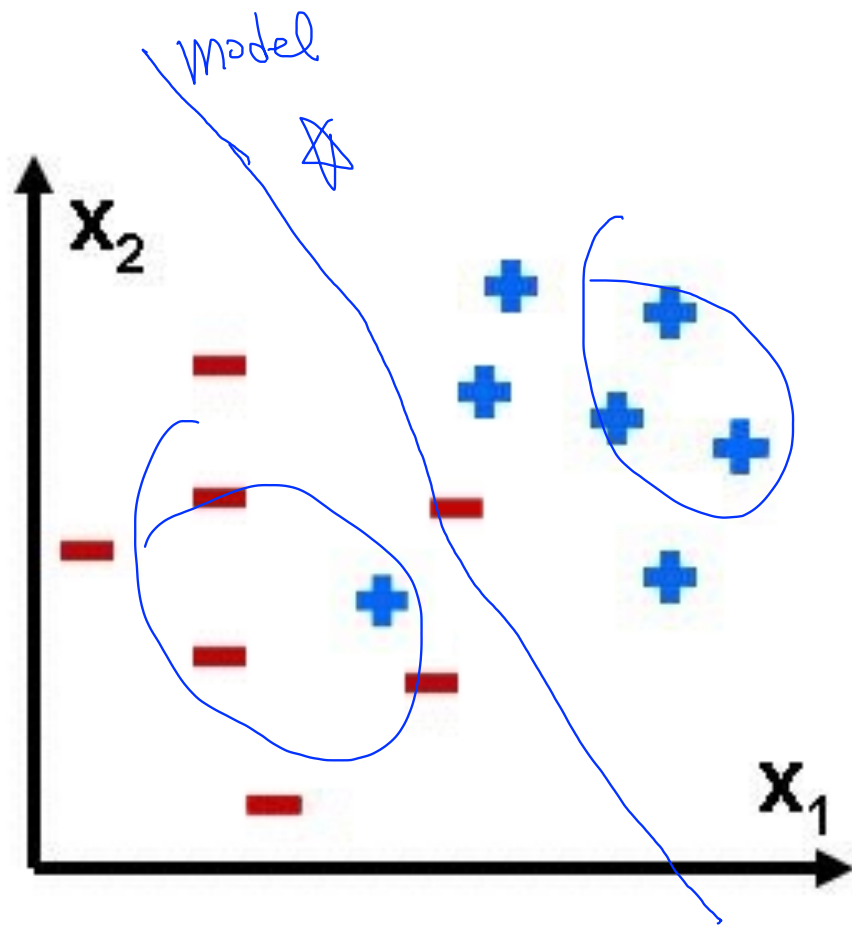
```
x_std[:,0] = (X[:,0] - X[:,0].mean()) / X[:,0].std()
```

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



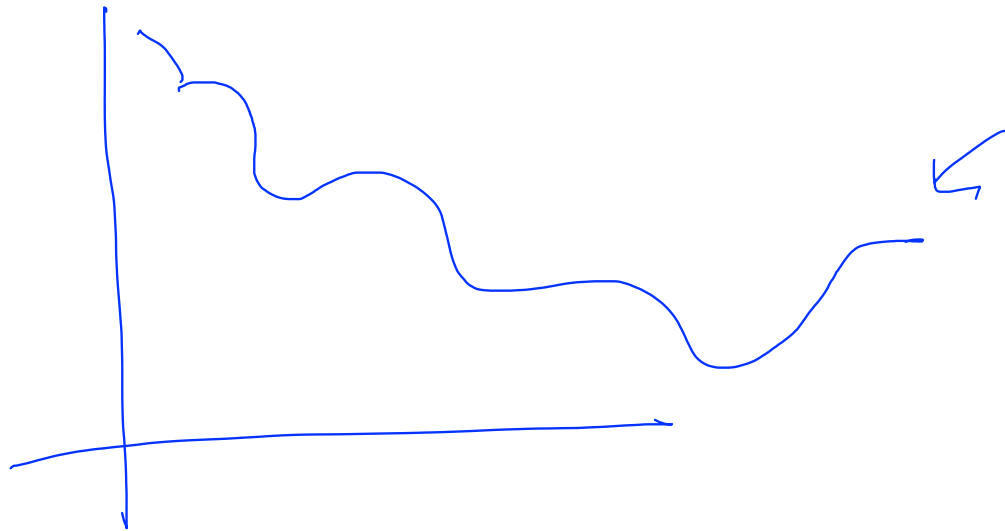
Solutions for overfitting

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



Regularization

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



Regularization

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

A handwritten diagram illustrating the loss function. The equation is $\mathcal{L} = \frac{1}{N} \sum_i \mathcal{D}(s(w x_i + b), L_i)$. Annotations include: a blue arrow labeled "LOSS" pointing to the \mathcal{L} term; a blue arrow labeled "TRAINING SET" pointing to the index i in the summation; and two blue arrows pointing from the "TRAINING SET" label to the x_i and L_i terms in the distance function \mathcal{D} .

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

Regularization

- 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

0 X
↓ ↑
0.001

Regularization

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

LOSS

$\text{l2reg} = 0.001 * \text{tf.reduce_sum}(\text{tf.square}(W))$

Regularization Strength

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

TRAINING SET

The diagram illustrates the components of the loss function and their corresponding code. The word 'LOSS' points to the overall loss equation. The 'TRAINING SET' points to the data points x_i and L_i in the loss equation. The 'Regularization Strength' points to the λ parameter in the loss equation and the 0.001 coefficient in the TensorFlow code. The TensorFlow code $\text{l2reg} = 0.001 * \text{tf.reduce_sum}(\text{tf.square}(W))$ is shown in a yellow box, with l2reg highlighted in a pink box. The loss equation is $\mathcal{L} = \frac{1}{N} \sum_i \mathcal{D}(s(w x_i + b), L_i) + \lambda \sum W^2$.

Summary

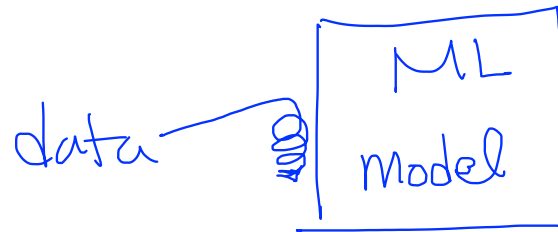
- Learning rate ✓
- Data preprocessing
- Overfitting ✗
 - More training data
 - Regularization

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?

Size	Price
2104	400
1600	330
2400	369
1416	232
3000	540
1985	300
1534	315
1427	199
1380	212
1494	243

training set

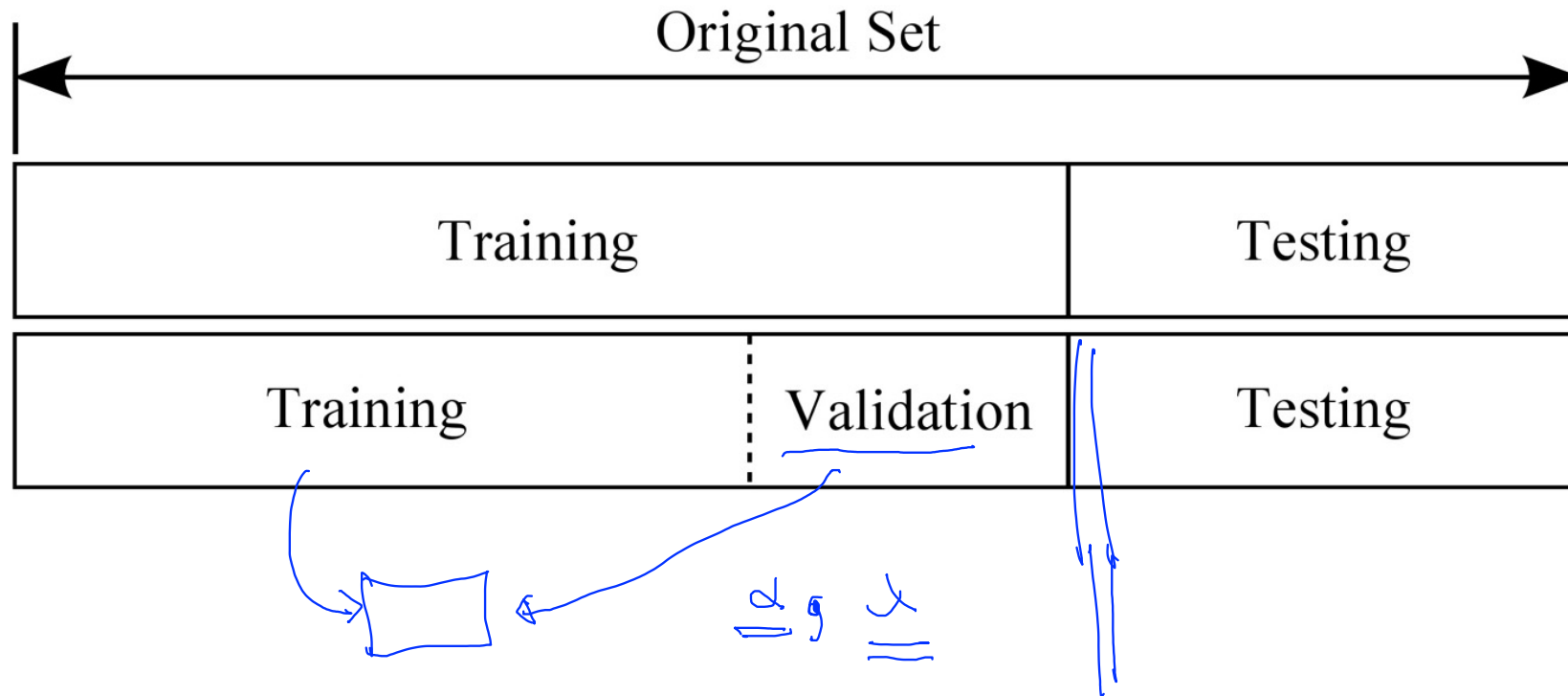


- 100% correct (accuracy)
- Can memorize

Training and test sets

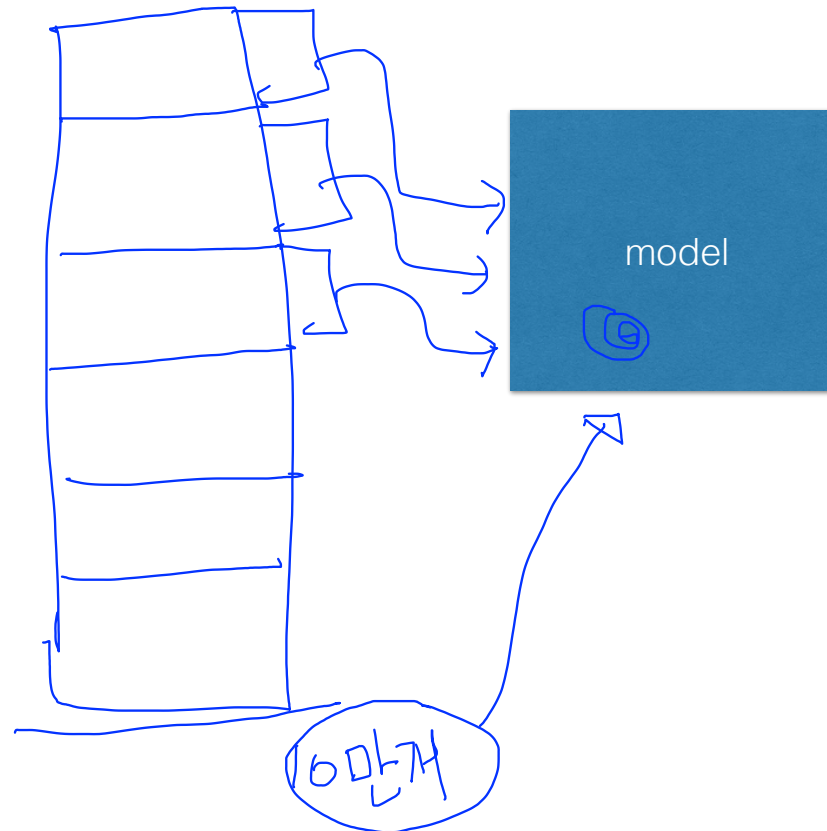
	Size	Price
training (교과서)	2104	400
	1600	330
	2400	369
	1416	232
	3000	540
	1985	300
	1534	315
test set (시험)	1427	199
	1380	212
	1494	243

Training, validation and test sets



Online learning

100만개



M NIST 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

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

