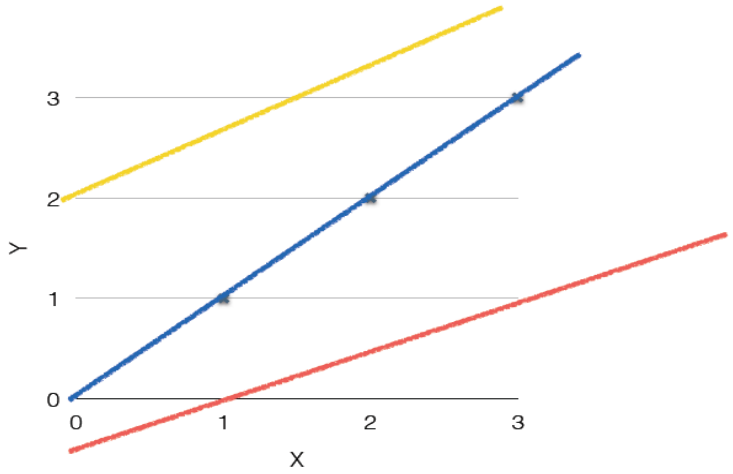
Machine Learning 1주차

발표자 : 박재형

Lec2. Linear regression

- 어떤 선이 Data에 맞는 선일까?

  -> 학습을 통해 찾는 것



- Step1. H(x) = Wx+ b 로 Hypothesis

- Step2. cost function으로 최적 Linear찾기

Cost Function

- Cost function basis form =>$H(x) - y$
- $H(x)$는 원하는 가설, y는 실제 값
- $cost(W, b) = \frac{1}{m}\sum\left(H\left(x^i\right) - y^i\right)^2$ 로 정리
- 목표 : minimize $cost(W, b)$

Full code (less than 20 lines)

```python
import tensorflow as tf

# X and Y data
x_train = [1, 2, 3]
y_train = [1, 2, 3]

W = tf.Variable(tf.random_normal([1]), name='weight')
b = tf.Variable(tf.random_normal([1]), name='bias')

# Our hypothesis XW+b
hypothesis = x_train * W + b

# cost/loss function
cost = tf.reduce_mean(tf.square(hypothesis - y_train))

# Minimize
optimizer = tf.train.GradientDescentOptimizer(learning_rate=0.01)
train = optimizer.minimize(cost)

# Launch the graph in a session.
sess = tf.Session()
# Initializes global variables in the graph.
sess.run(tf.global_variables_initializer())

# Fit the line
for step in range(2001):
    sess.run(train)
    if step % 20 == 0:
        print(step, sess.run(cost), sess.run(W), sess.run(b))
```

중복?

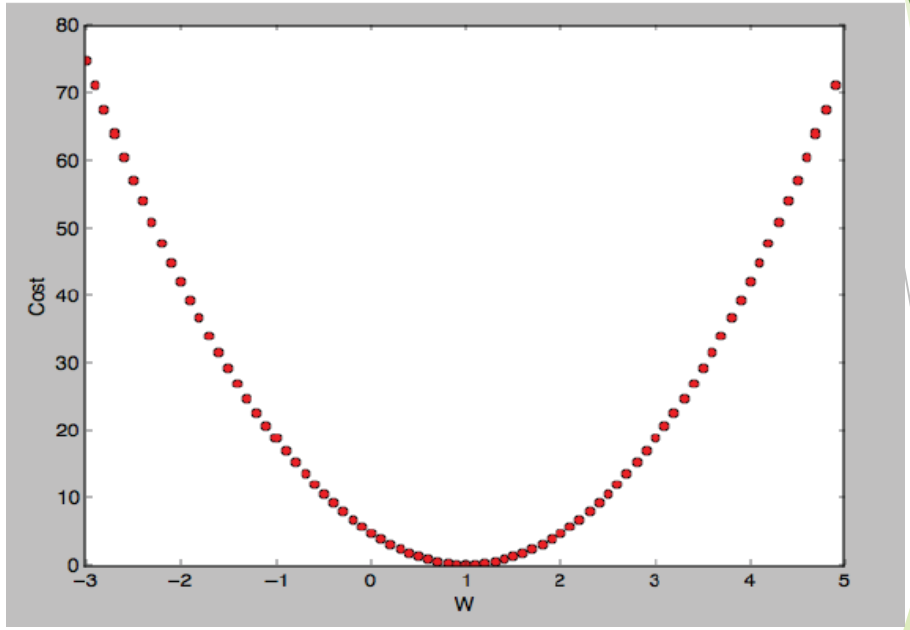
```
'''
0 2.82329 [ 2.12867713] [-0.85235667]
20 0.190351 [ 1.53392804] [-1.05059612]
40 0.151357 [ 1.45725465] [-1.02391243]
...

1920 1.77484e-05 [ 1.00489295] [-0.01112291]
1940 1.61197e-05 [ 1.00466311] [-0.01060018]
1960 1.46397e-05 [ 1.004444] [-0.01010205]
1980 1.32962e-05 [ 1.00423515] [-0.00962736]
2000 1.20761e-05 [ 1.00403607] [-0.00917497]
'''
```

Lec3. How to minimize cost

- 우선 Simplified hypothesis

- $H(x) = Wx + b$

- $cost(W) = \frac{1}{m}\sum(Wx^i - y^i)^2$



Gradient descent algorithm

- $cost(W) = \frac{1}{m}\sum(Wx^i - y^i)^2$

- $cost(W) = \frac{1}{2m}\sum(Wx^i - y^i)^2$

중간에 사라지는 이유??

- $W := W - \alpha \frac{\partial}{\partial W} cost(W)$

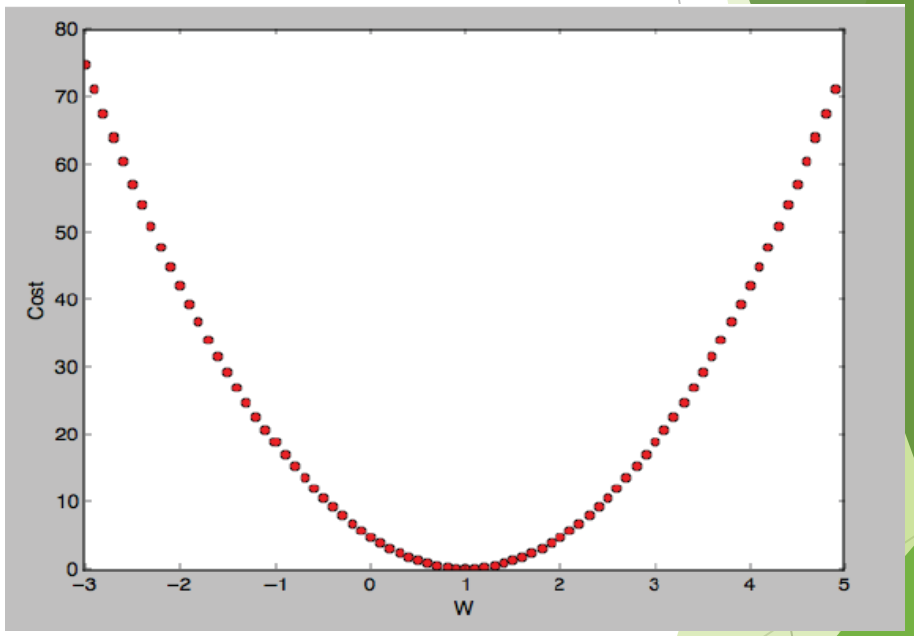
Formal definition

충분히 작지 않으면?

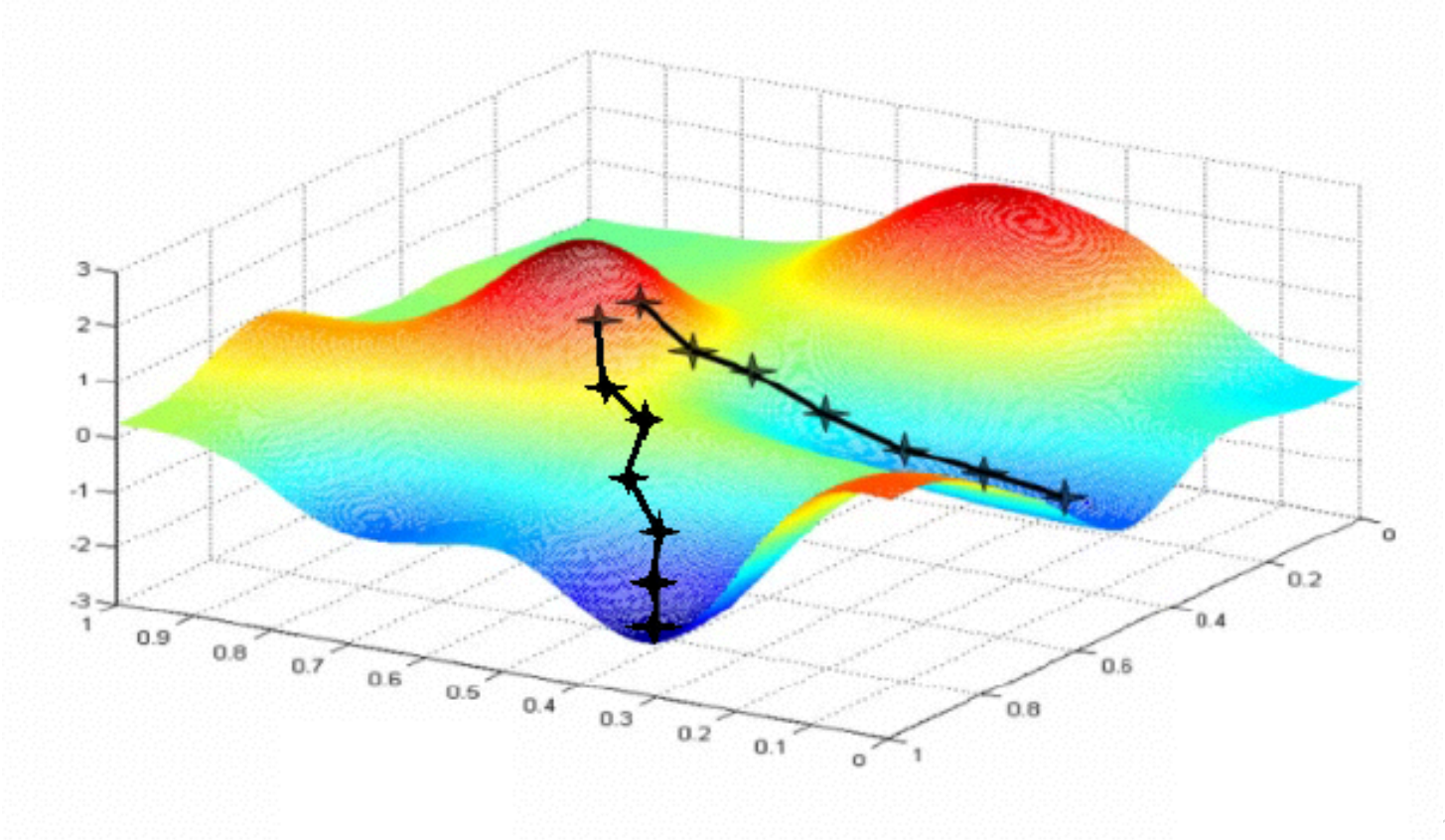
$$W := W - \alpha \frac{\partial}{\partial W} \frac{1}{2m} \sum_{i=1}^{m} (Wx^{(i)} - y^{(i)})^2$$

$$W := W - \alpha \frac{1}{2m} \sum_{i=1}^{m} 2(Wx^{(i)} - y^{(i)})x^{(i)}$$

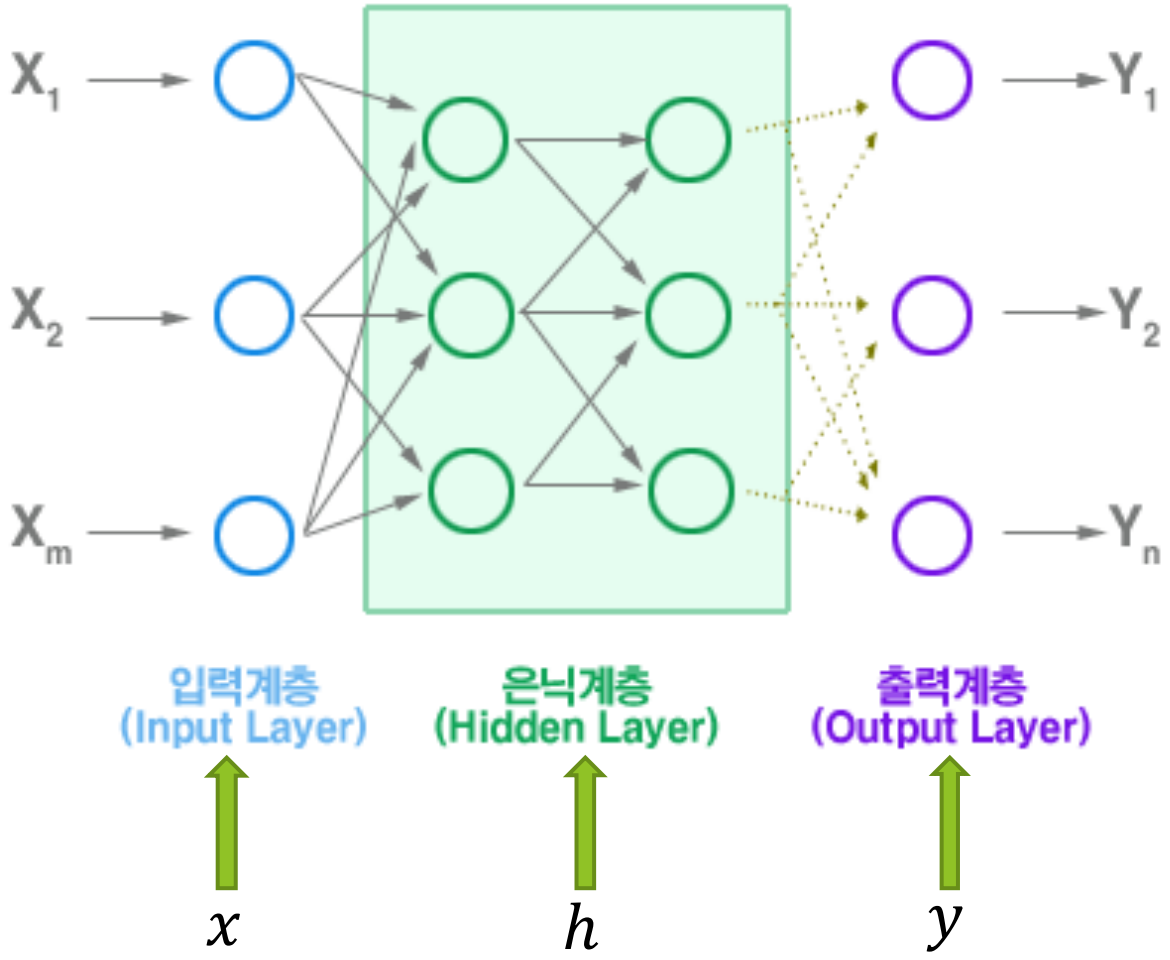
$$W := W - \alpha \frac{1}{m} \sum_{i=1}^{m} (Wx^{(i)} - y^{(i)})x^{(i)}$$



Non-convex



Lec4. 종합 정리

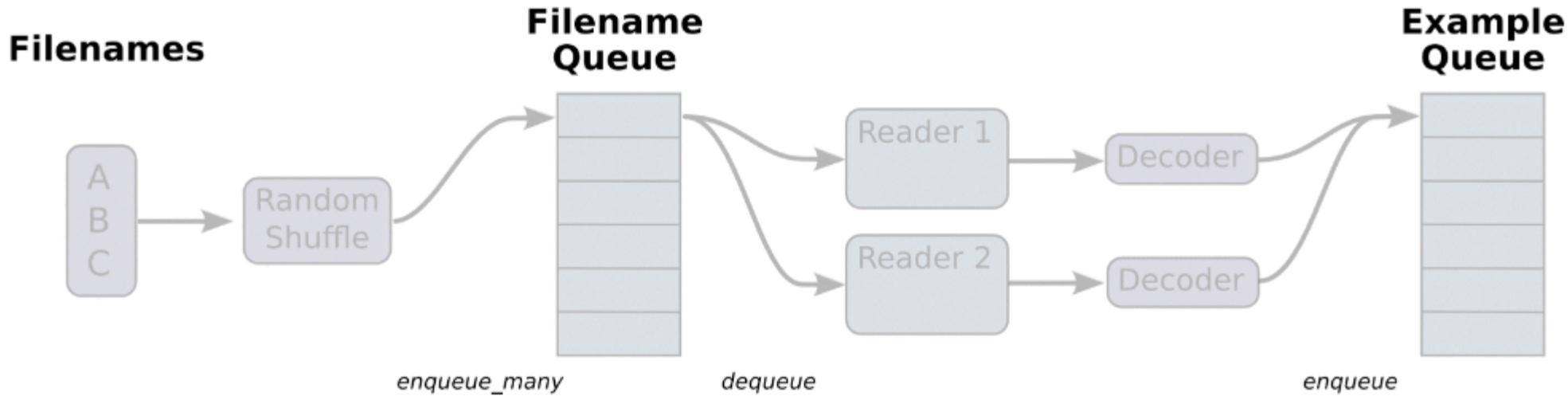


$h_1 = f(w_1x_1 + w_2x_2 + w_3x_3 + b)$;(b는 bias, 평행이동)

```
OSError: data-01-test-score.csv not found.

In [14]: xy = np.loadtxt('c:\\Users\\Jee\\Desktop\\data-01-test-score.csv', delimiter=',',dtype=np.float32)

In [15]: x_data = xy[:, 0:-1]

In [16]: y_data = xy[:, [-1]]

In [17]: print(x_data.shape, x_data, len(x_data))
(6, 3) [[ 73.  80.  75.]
 [ 93.  88.  93.]
 [ 89.  91.  90.]
 [ 96.  98. 100.]
 [ 73.  66.  70.]
 [ 53.  46.  55.]] 6

In [18]: print(y_data.shape, y_data)
(6, 1) [[152.]
 [185.]
 [180.]
 [196.]
 [142.]
 [101.]]

In [19]: X = tf.placeholder(tf.float32, shape=[None, 3])

In [20]: Y = tf.placeholder(tf.float32, shape=[None, 1])

In [21]: W = tf.Variable(tf.random_normal([3, 1]), name='weight')

In [22]: b = tf.Variable(tf.random_normal([1]), name='bias')

In [23]: hypothesis = tf.matmul(X, W) + b

In [24]: cost = tf.reduce_mean(tf.square(hypothesis - Y))

In [25]: optimizer = tf.train.GradientDescentOptimizer(learning_rate=1e-5)

In [26]: train = optimizer.minimize(cost)

In [27]: sess = tf.Session()
2018-03-14 17:50:52.023087: I C:\tf_jenkins\workspace\rel-win\M\windows\PY\36\tensorflow\core\platform\cpu_feature_guard.cc:140] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2

In [28]: sess.run(tf.global_variables_initializer())

In [29]: for step in range(2001):
    ...:     cost_val, hy_val, _ = sess.run([cost, hypothesis, train],feed_dict={X: x_data, Y: y_data})
    ...: if step % 10 == 0:
    ...:         print(step, "Cost: ", cost_val,"\nPrediction:\n", hy_val)
    ...:
2000 Cost:  2.7586195
Prediction:
 [[154.27278 ]
 [183.32866 ]
 [181.96426 ]
 [194.5939  ]
 [142.35019 ]
 [ 99.376816]]

In [30]: print("Your score will be ", sess.run(hypothesis,feed_dict={X: [[100, 70, 101]]}))
Your score will be  [[171.15341]]
```

1
```
filename_queue = tf.train.string_input_producer(
    ['data-01-test-score.csv', 'data-02-test-score.csv', ... ],
    shuffle=False, name='filename_queue')
```

3
```
record_defaults = [[0.], [0.], [0.], [0.]]
xy = tf.decode_csv(value, record_defaults=record_defaults)
```



2
```
reader = tf.TextLineReader()
key, value = reader.read(filename_queue)
```

Generalized expression?(irls)

Iteratively reweighted least squares

From Wikipedia, the free encyclopedia

For iterated weighted least squares, see Feasible generalized least squares.

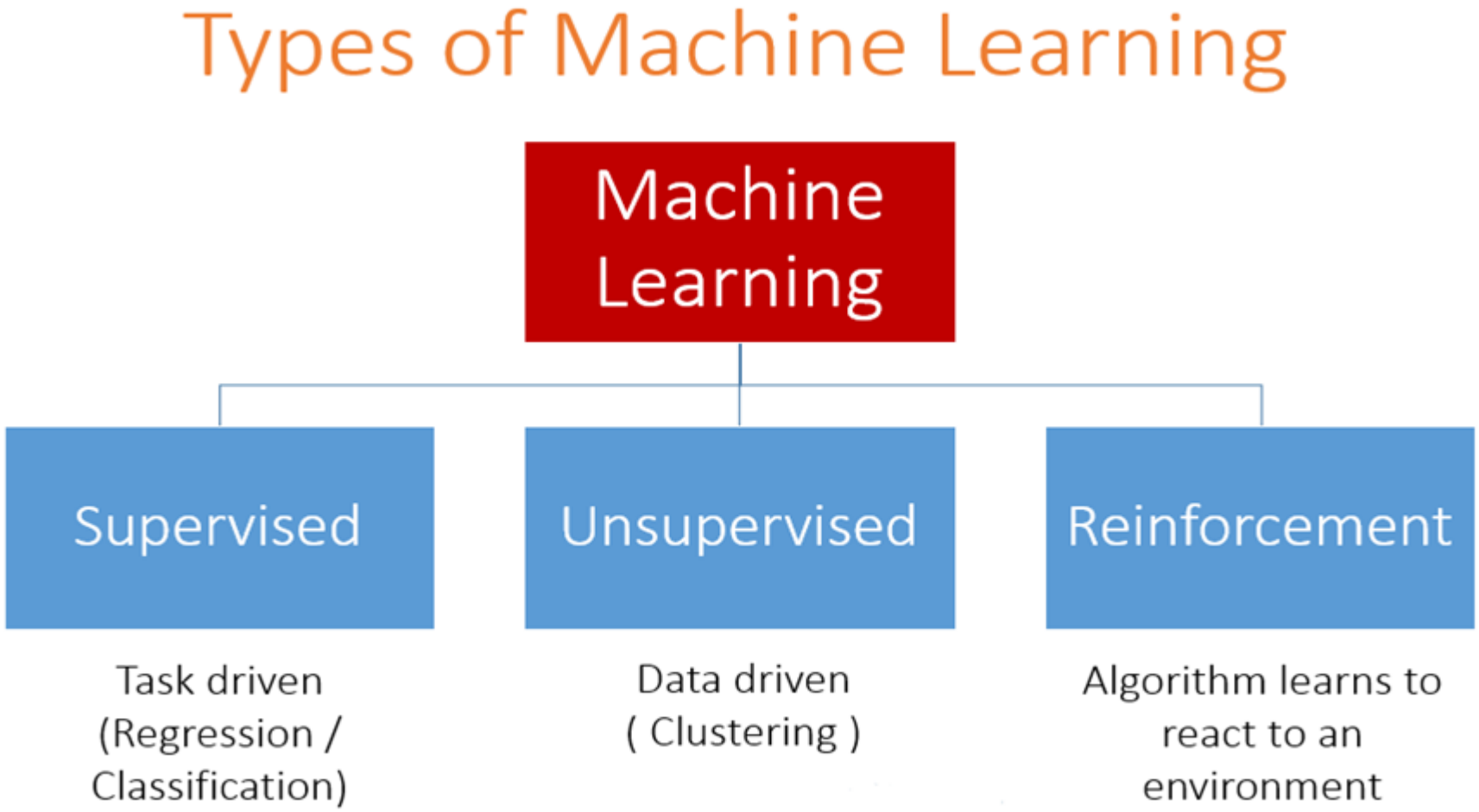
The method of **iteratively reweighted least squares** (**IRLS**) is used to solve certain optimization problems with objective functions of the form:

$$\arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left| y_i - f_i(\boldsymbol{\beta}) \right|^p,$$

by an iterative method in which each step involves solving a weighted least squares problem of the form:[1]

$$\boldsymbol{\beta}^{(t+1)} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} w_i(\boldsymbol{\beta}^{(t)}) \left| y_i - f_i(\boldsymbol{\beta}) \right|^2.$$

Intro(수업x)


Types of Machine Learning
Machine Learning
Supervised
Unsupervised
Reinforcement
Task driven (Regression / Classification)
Data driven (Clustering)
Algorithm learns to react to an environment

수업 정리

- 보통은 fully connected : 이전 단계 모든 뉴런과 연결
- CNN은 fully connected 하지 않다.

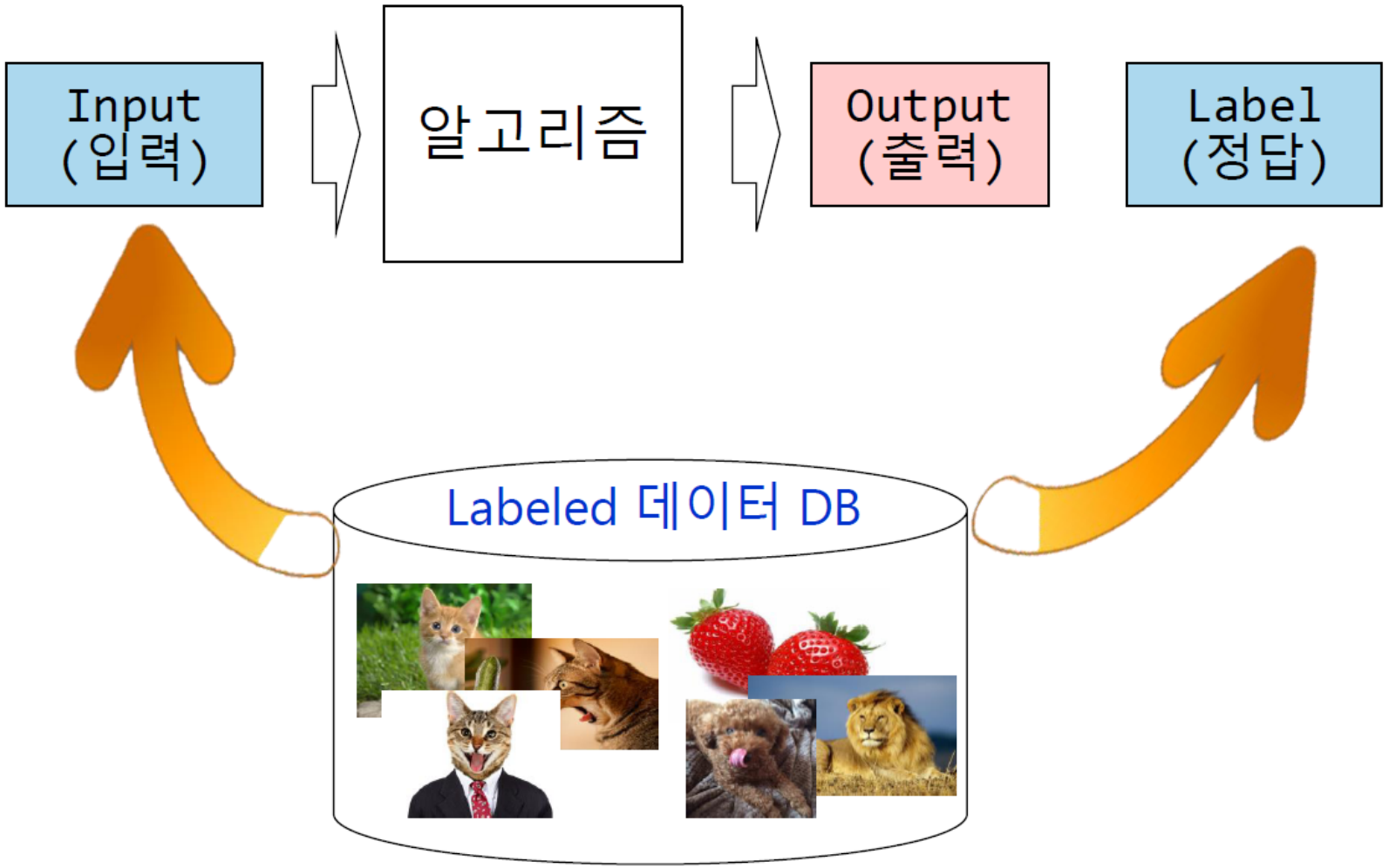
- DNN은 더 복잡한 함수 표현 가능하다.
- (chain rule??)
- (nonlinear 함수를 복잡하게)

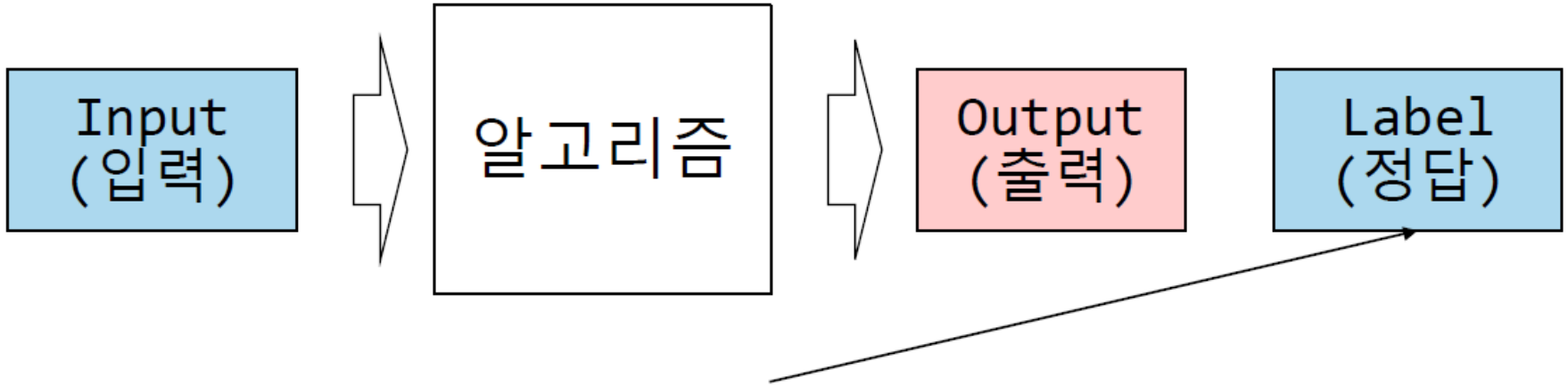
수업 정리

- Supervised learning
  - 데이터가 충분히 필요하다.(labeling또한 필요)
  - 정답도 같이 수집(high cost)

- Unsupervised learning
  - 데이터 label이 없는 것(feature 입력 x)
  - 데이터 축약(feature)->데이터 복원
  - data driven approach->딥러닝이 feature expression (특징추출)

Supervised Learning(지도 학습)

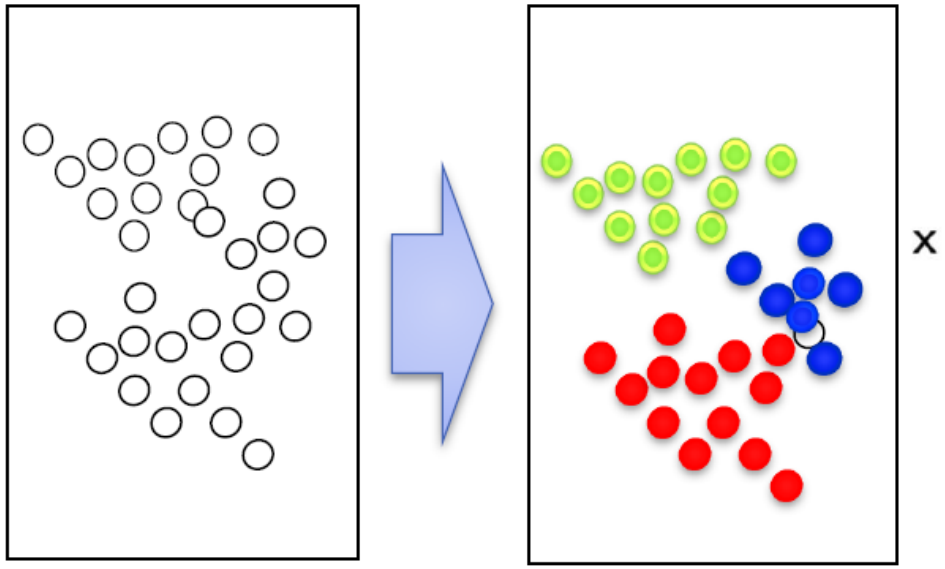




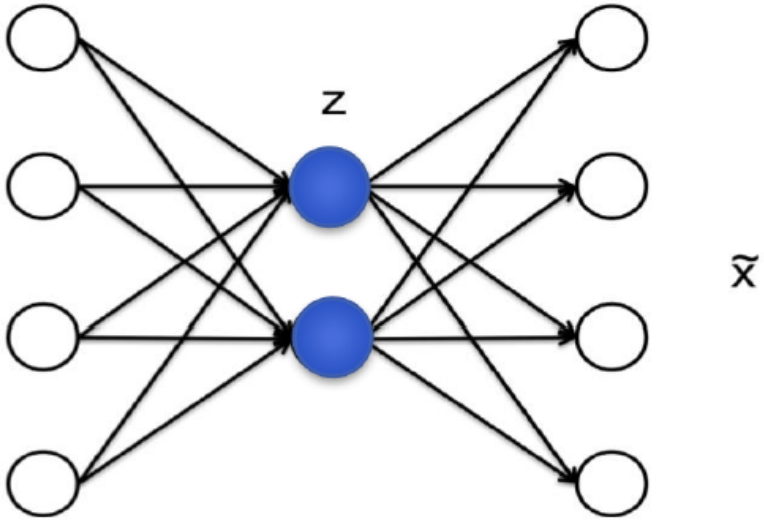
- **Label은 대개 정확한 해답이나 정확한 분류(class)를 말함**
- **Classification : 정확한 class**
  - 예) 0~9이미지 분류시 label은 0~9
- **Regression : 정확한 수치**
  - 환율 예측시 환율이 label

Unsupervised Learning (비지도 학습)

- 데이터의 label이 없는 데, 패턴을 알고 싶을 때
- **Clustering 등**
- 데이터 압축



Clustering



Reconstruction Error

수업 정리

- 1) overfitting-> 너무 크게 잡는 것,
- (이미 뭐든 데이터 샘플을 암기)
- training sample작을 때 overfitting일어난다.

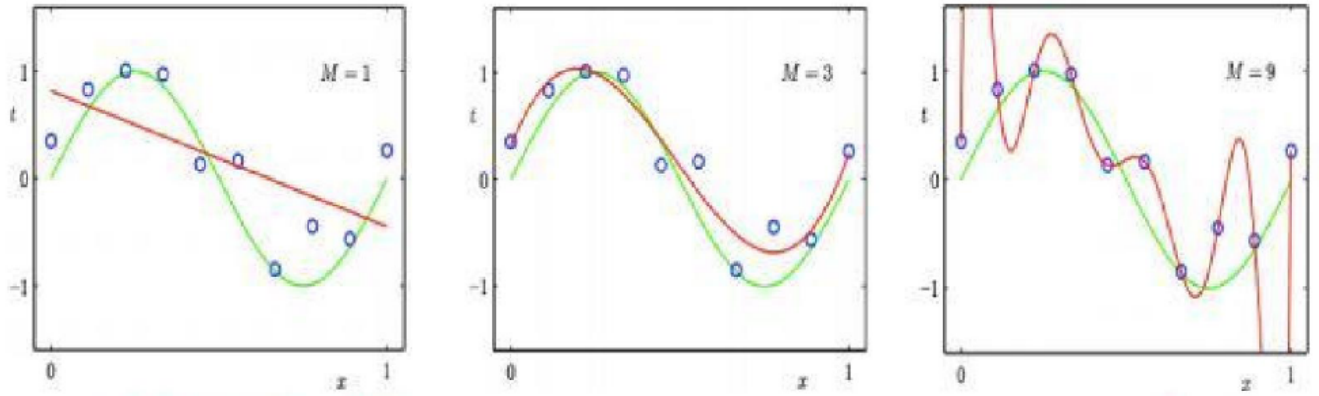
- 2) underfitting-> network가 충분히 깊고 크기 않아서 혹은 데이터가 작아서
- Deep learning인 경우 충분히 layer가 깊어야 효율UP (bottleneck 제거)

Under- and Over-fitting examples

underfitting

overfitting

Regression:



predictor too inflexible:
cannot capture pattern

predictor too flexible:
fits noise in the data

Classification: