11주차(2/3)

# 로지스틱 회귀 1

파이썬으로 배우는 기계학습

한동대학교
김영섭 교수

# 로지스틱 회귀

- 학습 목표
  - 회귀 분석을 익힌다.
  - 오차 함수의 차이를 인식한다.
  - 다층 신경망에 두 개의 활성화 함수를 사용한다.
  - 로지스틱 회귀 알고리즘을 배운다.

- 학습 내용
  - 회귀와 회귀 분석 정의
  - 오차 함수의 차이
  - 로지스틱 회귀

# 1. 회귀 분석: 정의

- 회귀**(Regression)**
  - 변수 간 관계
- 회귀 분석**(Regression Analysis)**
  - 예측

# 1. 회귀 분석: 정의

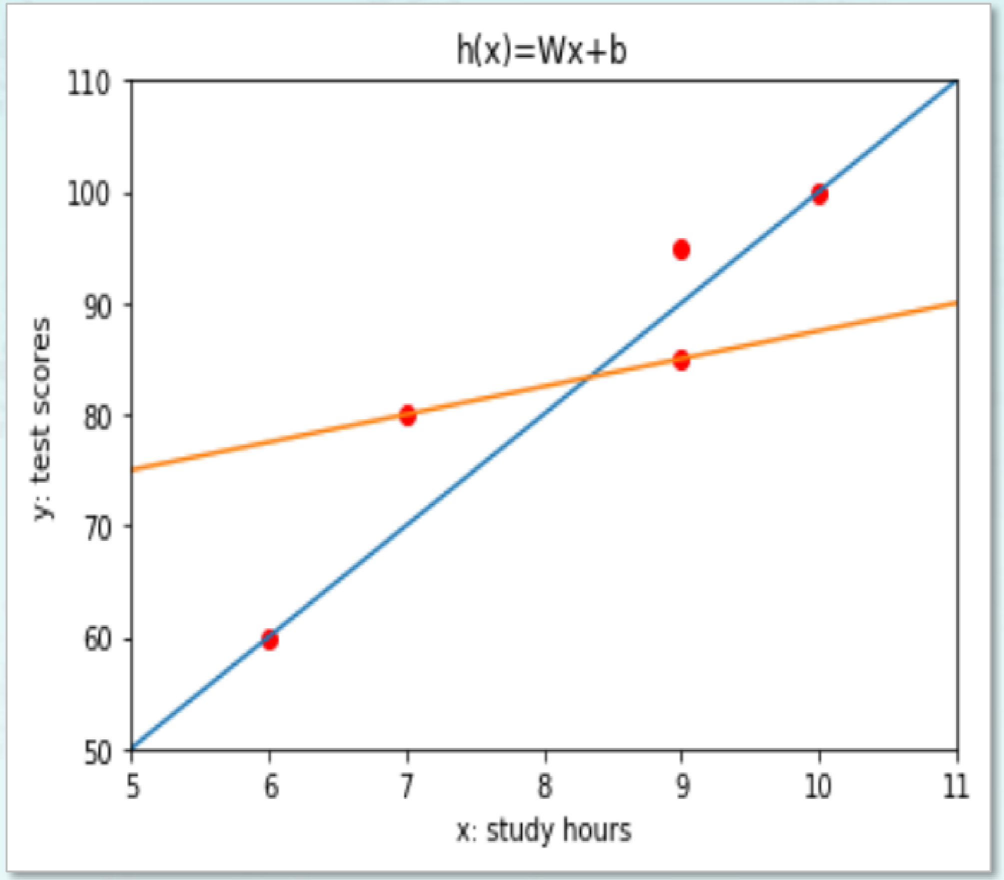
- 회귀(**Regression**)
  - 변수 간 관계
- 회귀 분석(**Regression Analysis**)
  - 예측

| 공부 시간(x) | 시험 성적(y) |
|---|---|
| 10 | 100 |
| 9 | 85 |
| 7 | 80 |
| 6 | 60 |
| 9 | 95 |
| … | … |

# 1. 회귀 분석: 단순 회귀 분석

- 단순 회귀 분석
  - $X, Y$

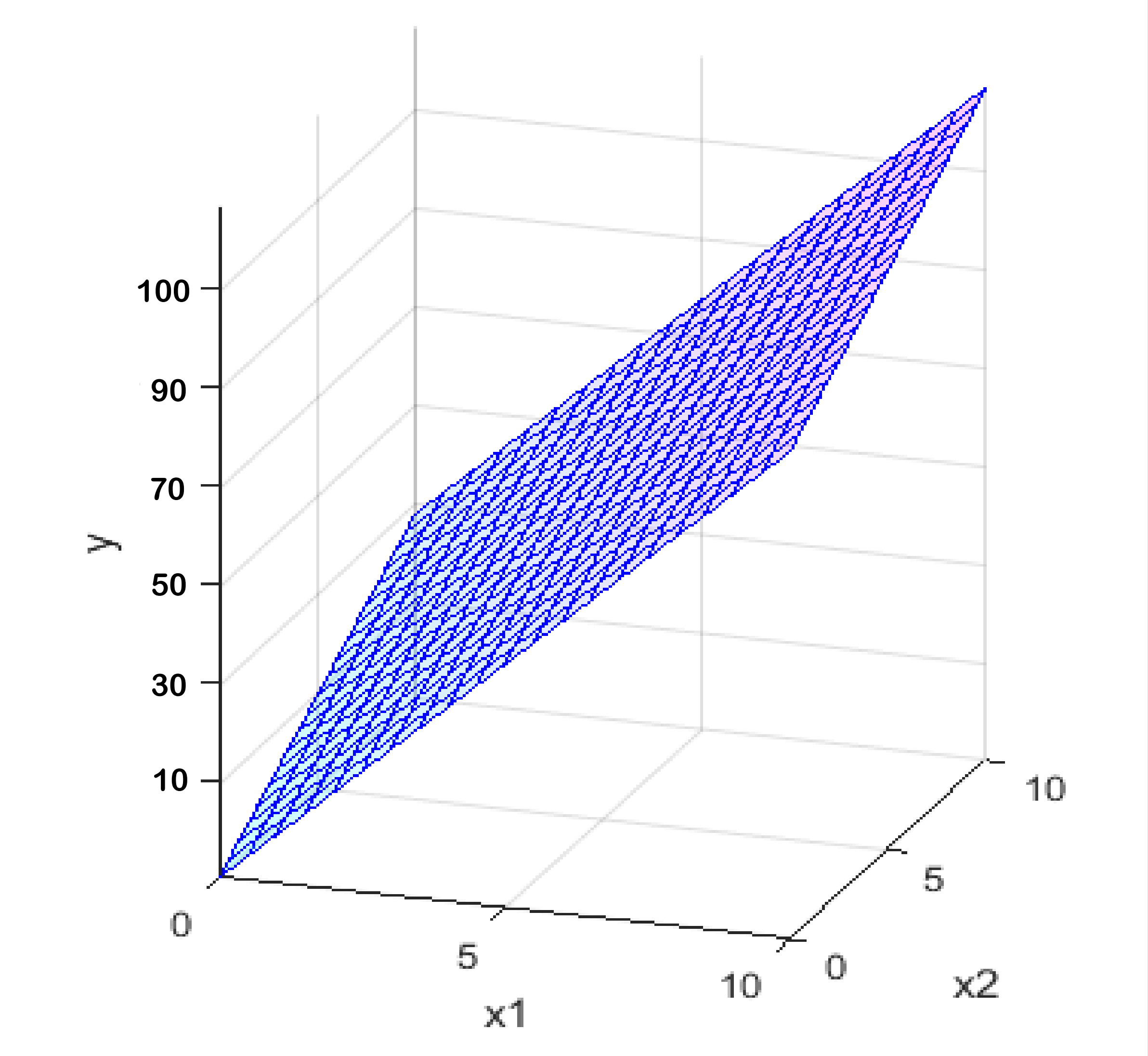
| 공부 시간(x) | 시험 성적(y) |
|---|---|
| 10 | 100 |
| 9 | 85 |
| 7 | 80 |
| 6 | 60 |
| 9 | 95 |
| … | … |



# 1. 회귀 분석: 다중 회귀 분석

- 다중 회귀 분석
  - $X_1, X_2, \ldots, Y$

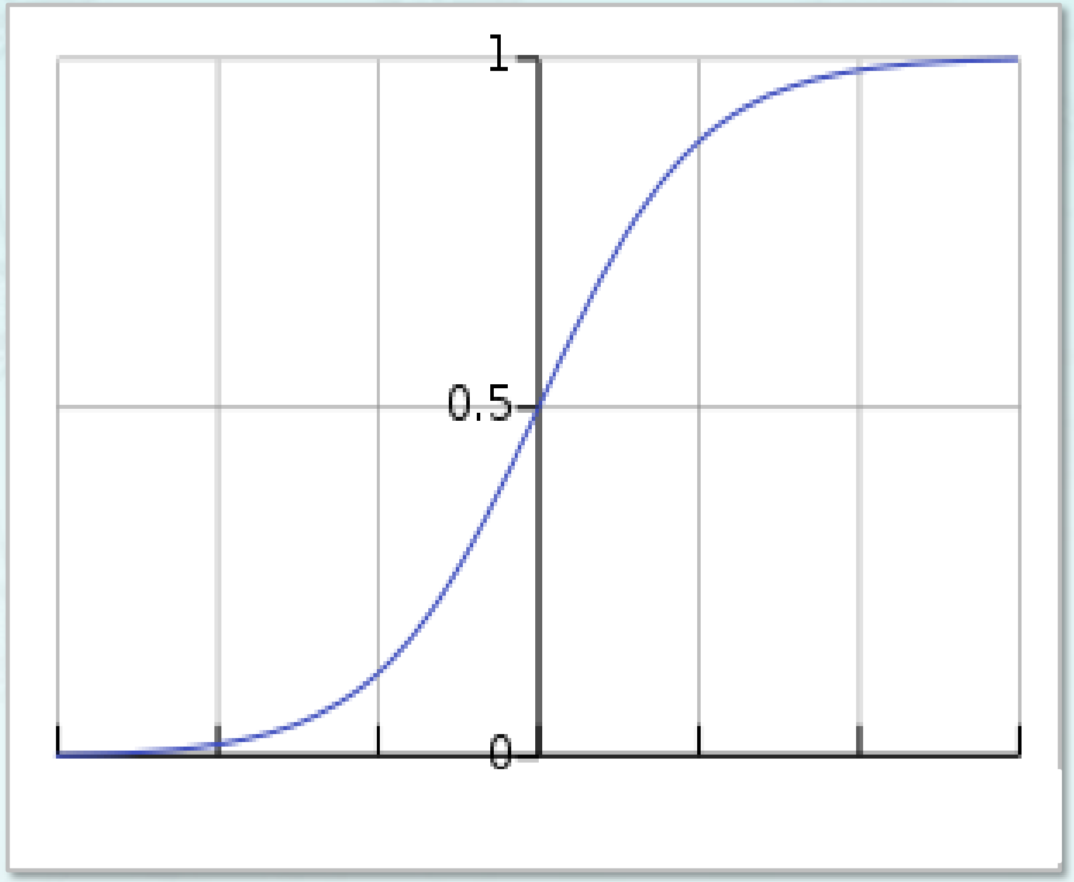
| 공부 시간 ($X_1$) | 집중 시간 ($X_2$) | 시험 성적 (y) |
|---|---|---|
| 10 | 10 | 100 |
| 9 | 8 | 85 |
| 7 | 6 | 80 |
| 6 | 4 | 60 |
| 9 | 9 | 95 |
| … | … | … |



# 1. 회귀 분석: 로지스틱 회귀

- 로지스틱 회귀(Logistic Regression)
  - 0, 1
  - Pass, Fail

| 시험 성적(x) | 합격(y) |
|---|---|
| 100 | P |
| 45 | F |
| 50 | P |
| 30 | F |
| 95 | P |
| … | … |



# 1. 로지스틱 회귀: 학습자료 준비

- 학습자료

자료 준비 단계

학습 단계

검증 단계

평가 단계

실전 배치

# 1. 로지스틱 회귀: 학습자료 준비

- 학습자료
- 시각화
- 이항 분류
  - 두 개의 클래스로 분류

```
import joy
X, Y = joy.planar_data()
m = X.shape[1]
print ('X.shape={} Y.shape={}, m={}'.
       format(X.shape, Y.shape, m))
joy.plot_xyw(X.T, Y.squeeze(),title="PlanaData")
```

```
X.shape=(2, 400) Y.shape=(1, 400), m=400
```

# 1. 로지스틱 회귀: 학습자료 준비

- 학습자료
- 시각화
- 이항 분류
  - 두 개의 클래스로 분류

$$X = \begin{pmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_{400}^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_{400}^{(2)} \end{pmatrix}$$

```python
import joy
X, Y = joy.planar_data()
m = X.shape[1]
print ('X.shape={} Y.shape={}, m={}'.
        format(X.shape, Y.shape, m))
joy.plot_xyw(X.T, Y.squeeze(),title="PlanaData")
```

```
X.shape=(2, 400) Y.shape=(1, 400), m=400
```

# 1. 로지스틱 회귀: 학습자료 준비

- 학습자료
- 시각화
- 이항 분류
  - 두 개의 클래스로 분류

$$X = \begin{pmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_{400}^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_{400}^{(2)} \end{pmatrix}$$

$$Y = \begin{pmatrix} y_1^{(1)} & y_2^{(1)} & \cdots & y_{400}^{(1)} \end{pmatrix}$$

```python
import joy
X, Y = joy.planar_data()
m = X.shape[1]
print ('X.shape={} Y.shape={}, m={}'.
       format(X.shape, Y.shape, m))
joy.plot_xyw(X.T, Y.squeeze(),title="PlanaData")
```

```
X.shape=(2, 400) Y.shape=(1, 400), m=400
```

# 1. 로지스틱 회귀: 학습자료 준비

- 학습자료
- 시각화
- 이항 분류
  - 두 개의 클래스로 분류

```python
import joy
X, Y = joy.planar_data()
m = X.shape[1]
print ('X.shape={} Y.shape={}, m={}'.
        format(X.shape, Y.shape, m))
joy.plot_xyw(X.T, Y.squeeze(),title="PlanaData")
```

```
X.shape=(2, 400) Y.shape=(1, 400), m=400
```

# 1. 로지스틱 회귀: 학습자료 준비

- 학습자료
- 시각화
- 이항 분류
  - 두 개의 클래스로 분류

```python
import joy
X, Y = joy.planar_data()
m = X.shape[1]
print ('X.shape={} Y.shape={}, m={}'.
        format(X.shape, Y.shape, m))
joy.plot_xyw(X.T, Y.squeeze(),title="PlanaData")
```

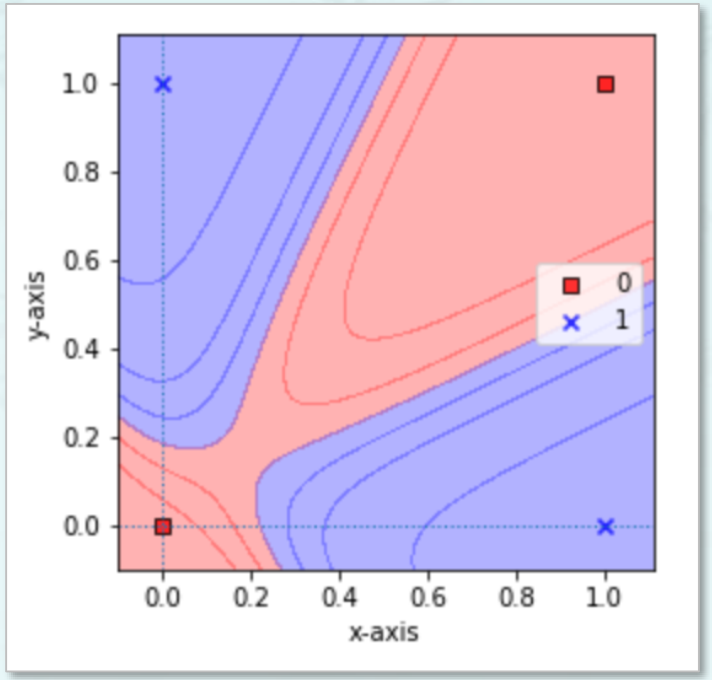
```
X.shape=(2, 400) Y.shape=(1, 400), m=400
```

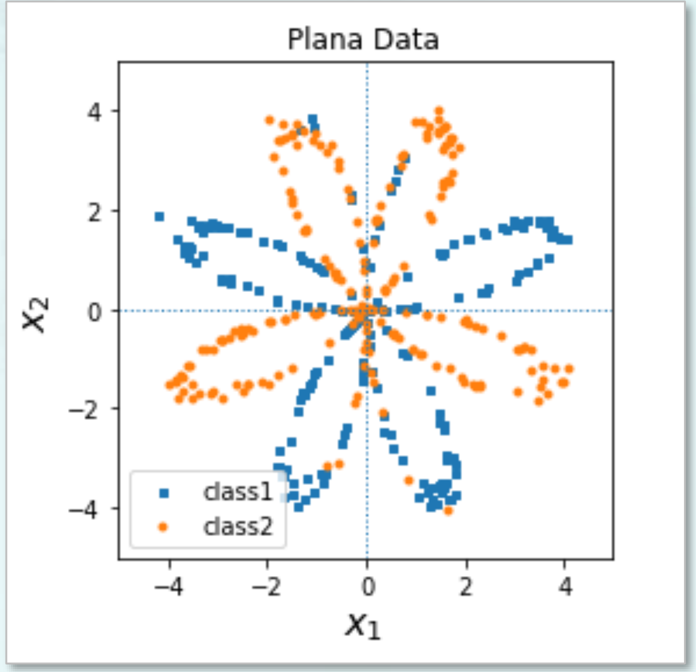
# 1. 로지스틱 회귀: 학습자료 준비

- 학습자료
- 시각화
- 이항 분류
  - 두 개의 클래스로 분류

```python
import joy
X, Y = joy.planar_data()
m = X.shape[1]
print ('X.shape={} Y.shape={}, m={}'.
        format(X.shape, Y.shape, m))
joy.plot_xyw(X.T, Y.squeeze(),title="PlanaData")
```

```
X.shape=(2, 400) Y.shape=(1, 400), m=400
```



# 2. 신경망 모델: 3층 신경망

- 신경망**(2 - 3 - 1)**

```
n_h = 3
net_arch=[2, n_h, 1]
nn = joy.NeuralNetwork(net_arch, eta=0.1, epochs=100)
nn.fit(X, Y)

joy.plot_decision_regions(X.T, Y, nn)
yhat = nn.predict(X.T)
accuracy = float(np.dot(Y, yhat.T) +
                 np.dot(1 - Y, 1 - yhat.T))/Y.size * 100
plt.title('Hidden neurons={}, accuracy={}'.
          format(n_h, np.round(accuracy,2)))
```

# 2. 신경망 모델: 3층 신경망

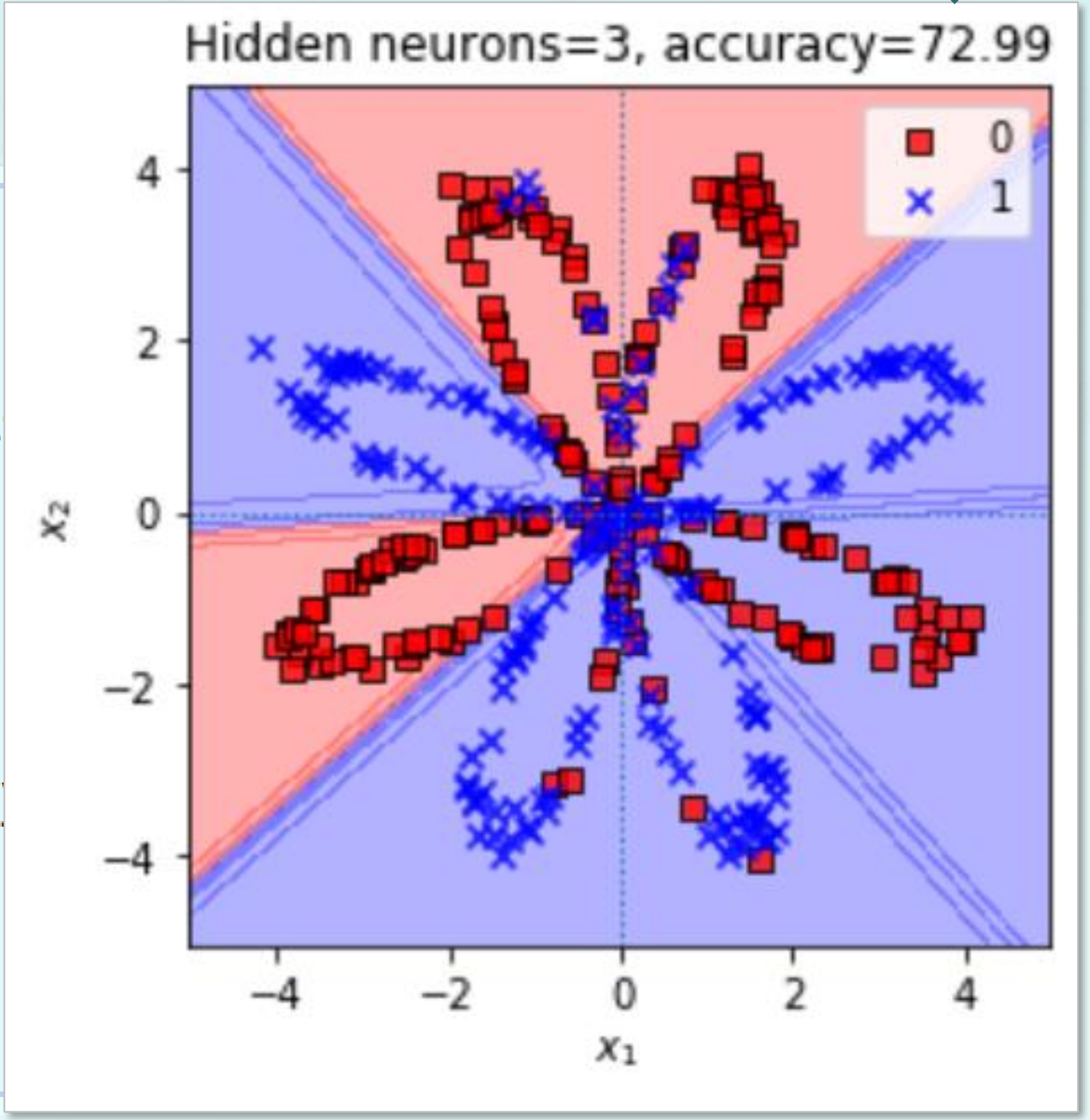
- 신경망**(2 - 3 - 1)**

```
n_h = 3
net_arch=[2, n_h, 1]
nn = joy.NeuralNetwork(net_arch, eta=0.1, epochs=100)
nn.fit(X, Y)

joy.plot_decision_regions(X.T, Y, nn)
yhat = nn.predict(X.T)
accuracy = float(np.dot(Y, yhat.T) +
                 np.dot(1 - Y, 1 - yhat.T))/Y.size * 100
plt.title('Hidden neurons={}, accuracy={}'.
          format(n_h, np.round(accuracy,2)))
```

# 2. 신경망 모델: 3층 신경망

- 신경망**(2 - 3 - 1)**

```
n_h = 3
net_arch=[2, n_h, 1]
nn = joy.NeuralNetwork(net_
nn.fit(X, Y)

joy.plot_decision_regions(X
yhat = nn.predict(X.T)
accuracy = float(np.dot(Y, 
                 np.dot(1 -
plt.title('Hidden neurons={
          format(n_h, np.ro
```



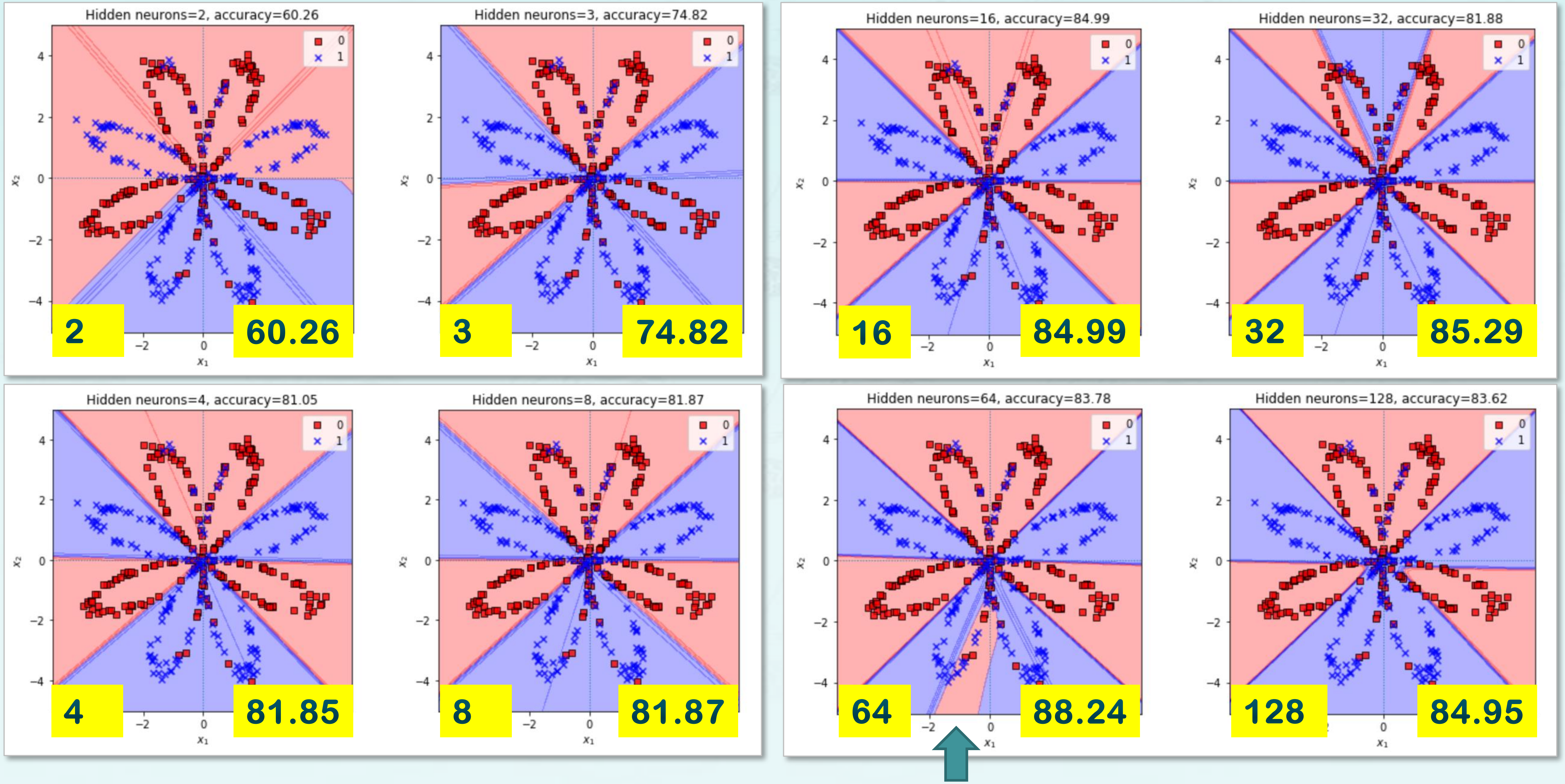
# 2. 신경망 모델: 3층 신경망

```python
X, Y = joy.planar_data()
plt.figure(figsize=(12, 24))
accuracy = []
number_of_neurons = [2, 3, 4, 8, 16, 32, 64, 128]
for i, n_h in enumerate(number_of_neurons):
    print('[{}] Processing {} neurons case....'.format(i, n_h))
    net_arch = [2, 1]
    net_arch.insert(1, n_h)
    nn = joy.NeuralNetwork(net_arch, eta=0.1, epochs=200)
    nn.fit(X, Y)                              # train the net

    plt.subplot(5, 2, i+1)
    joy.plot_decision_regions(X.T, Y, nn)
    yhat = nn.predict(X.T)
    accuracy.append(float(np.dot(Y, yhat.T) +
                          np.dot(1 - Y, 1 - yhat.T))/Y.size * 100)
    plt.title('Hidden neurons={}, accuracy={}'.
              format(n_h, np.round(accuracy[i],2)))
```

# 2. 신경망 모델: 3층 신경망

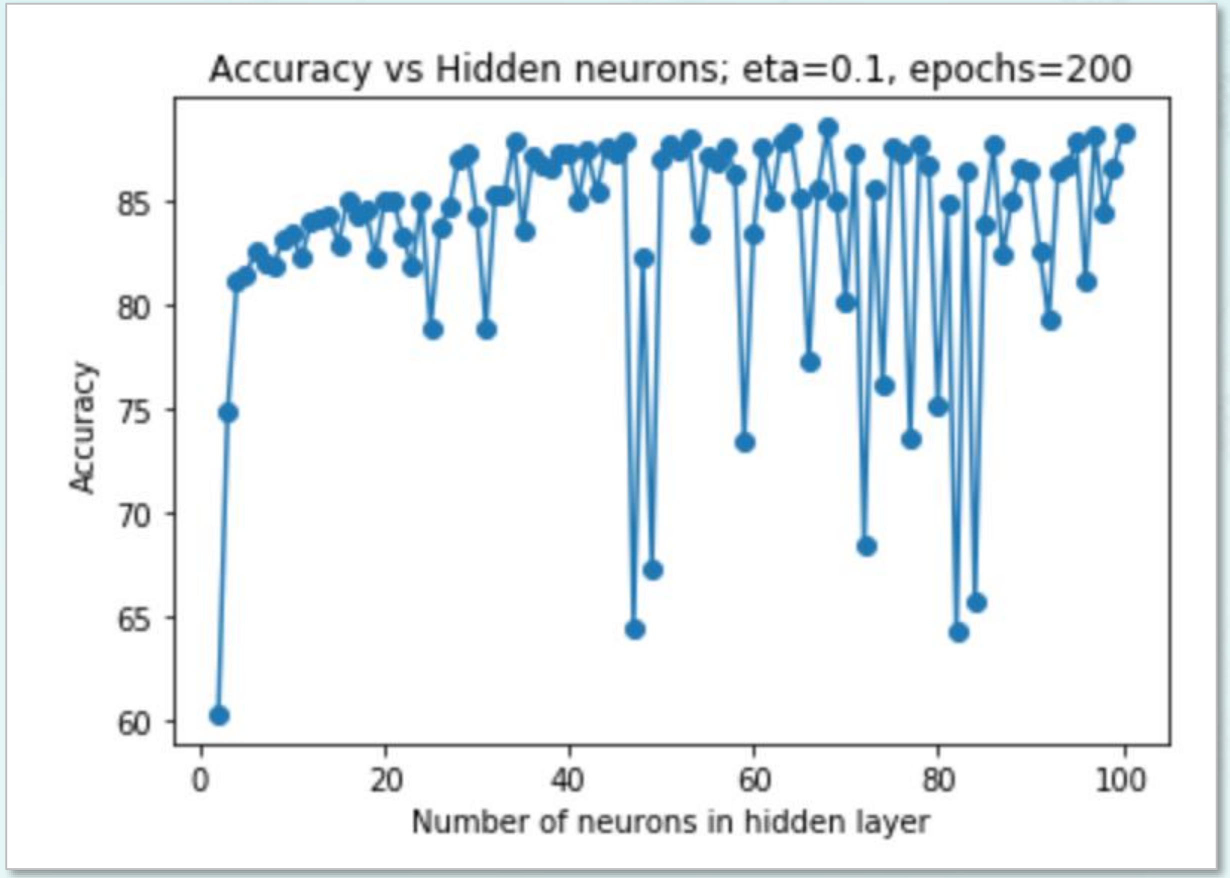
```
X, Y = joy.planar_data()
plt.figure(figsize=(12, 24))
accuracy = []
number_of_neurons = [2, 3, 4, 8, 16, 32, 64, 128]
for i, n_h in enumerate(number_of_neurons):
    print('[{}] Processing {} neurons case....'.format(i, n_h))
    net_arch = [2, 1]
    net_arch.insert(1, n_h)
    nn = joy.NeuralNetwork(net_arch, eta=0.1, epochs=200)
    nn.fit(X, Y)                        # train the net

    plt.subplot(5, 2, i+1)
    joy.plot_decision_regions(X.T, Y, nn)
    yhat = nn.predict(X.T)
    accuracy.append(float(np.dot(Y, yhat.T) +
                          np.dot(1 - Y, 1 - yhat.T))/Y.size * 100)
    plt.title('Hidden neurons={}, accuracy={}'.
              format(n_h, np.round(accuracy[i],2)))
```

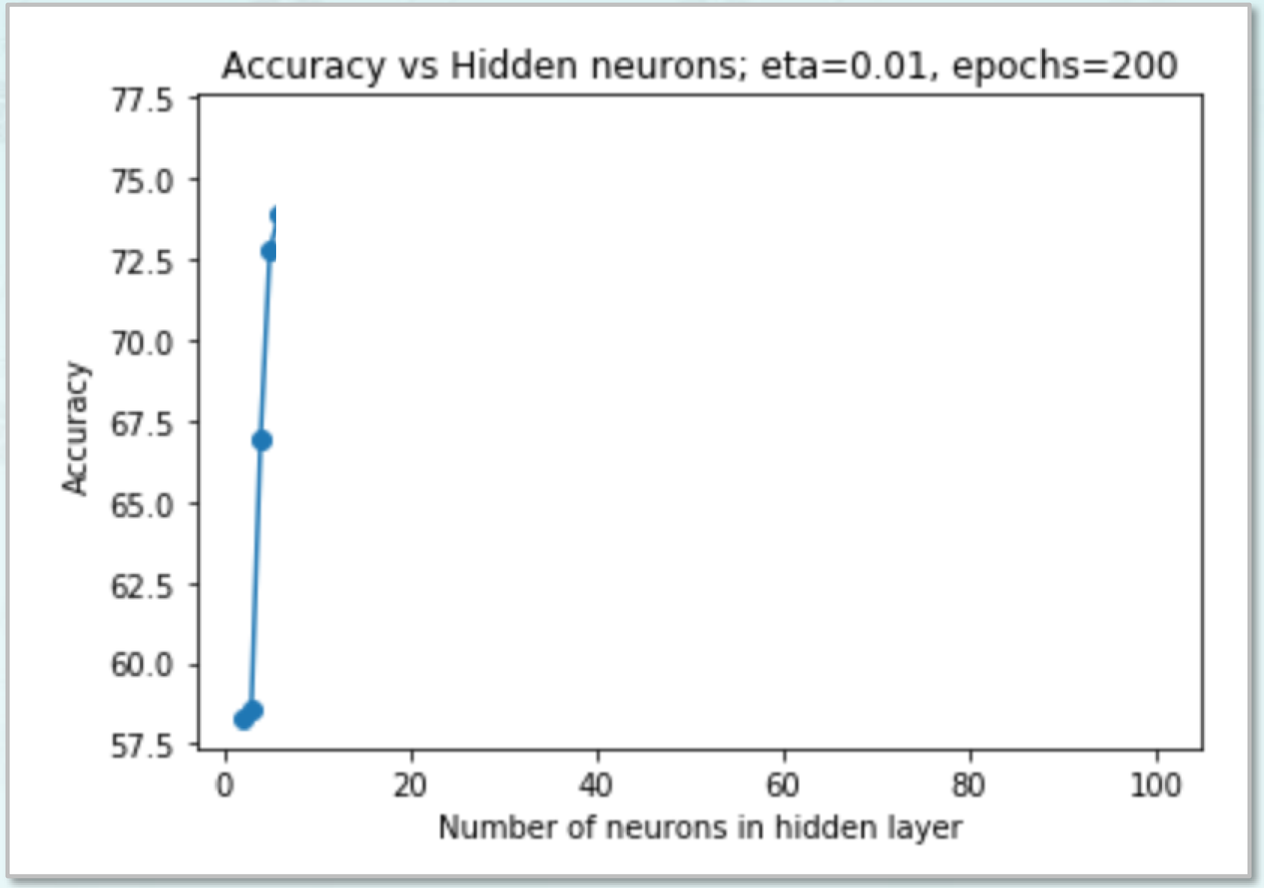
# 2. 신경망 모델: 3층 신경망



# 2. 신경망 모델: 3층 신경망

- 학습률 **eta=0.1, epochs=200**
- 학습률 **eta=0.01, epochs=200**

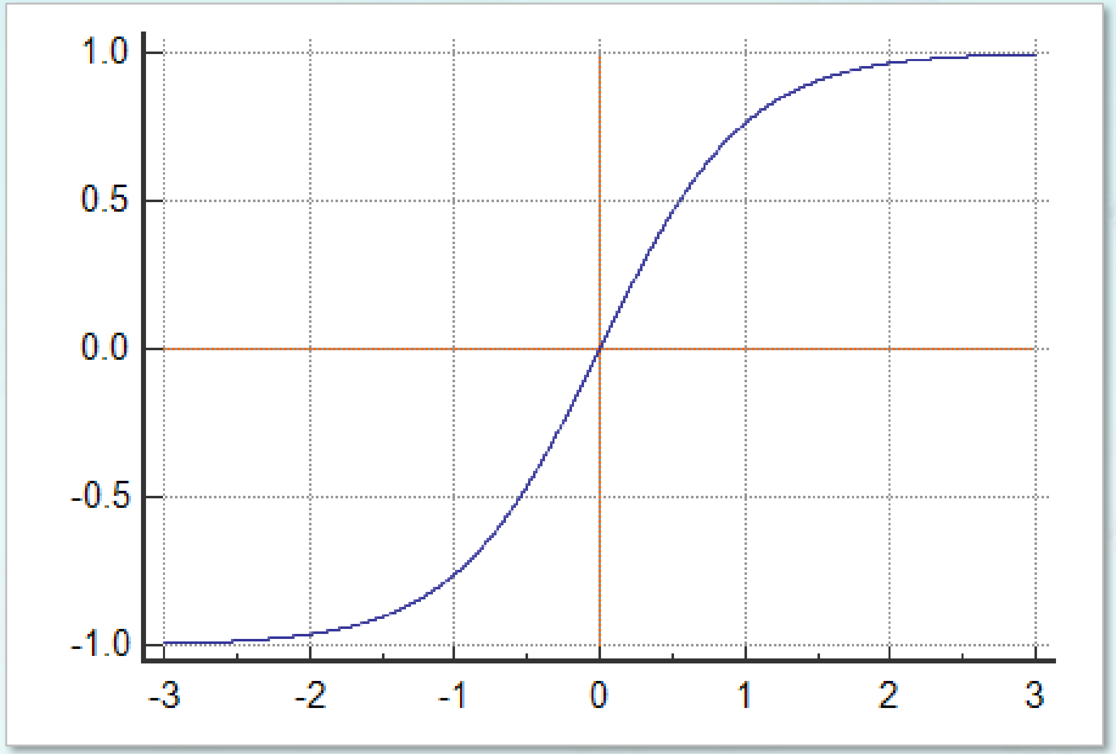


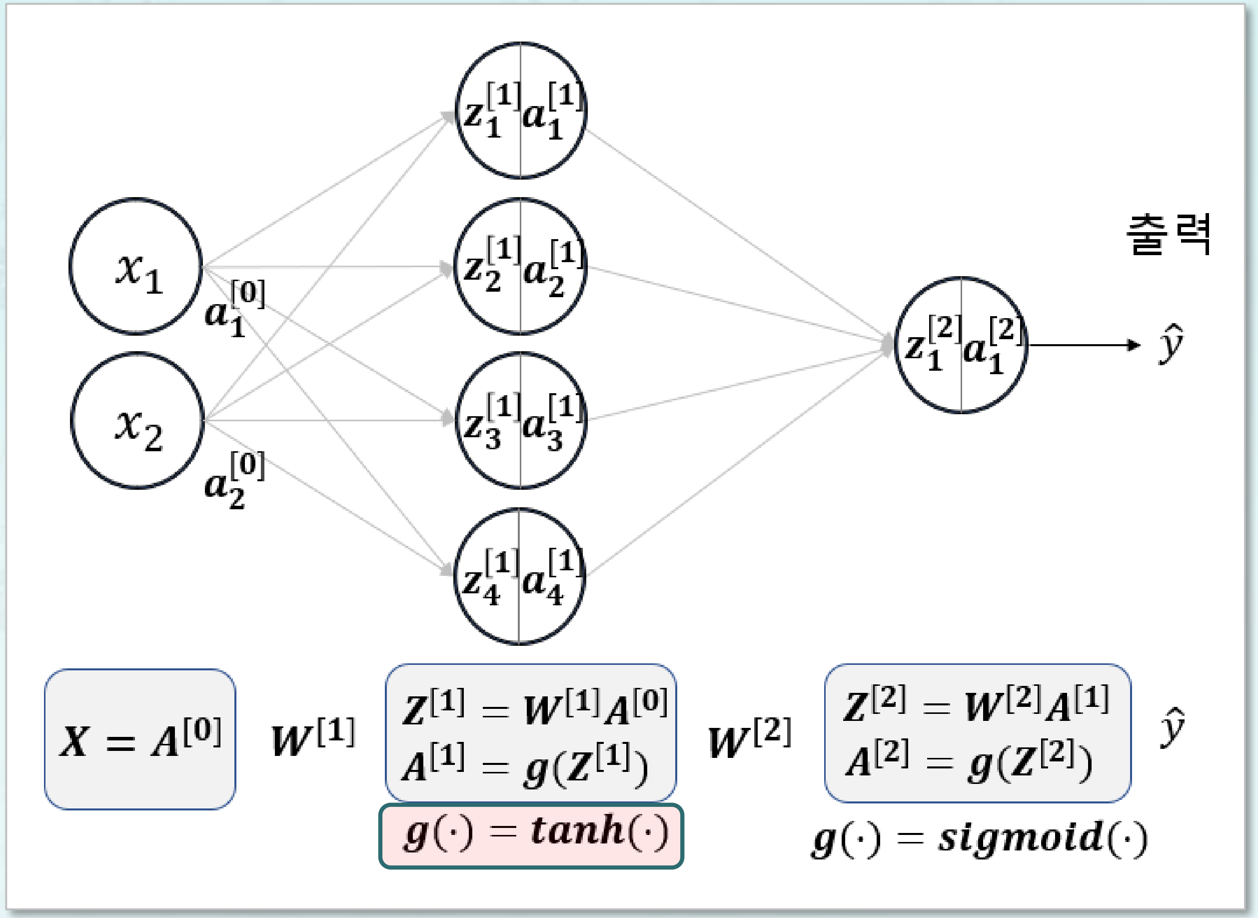



# 2. 신경망 모델: 3층 신경망

# 2. 신경망 모델: 3층 신경망

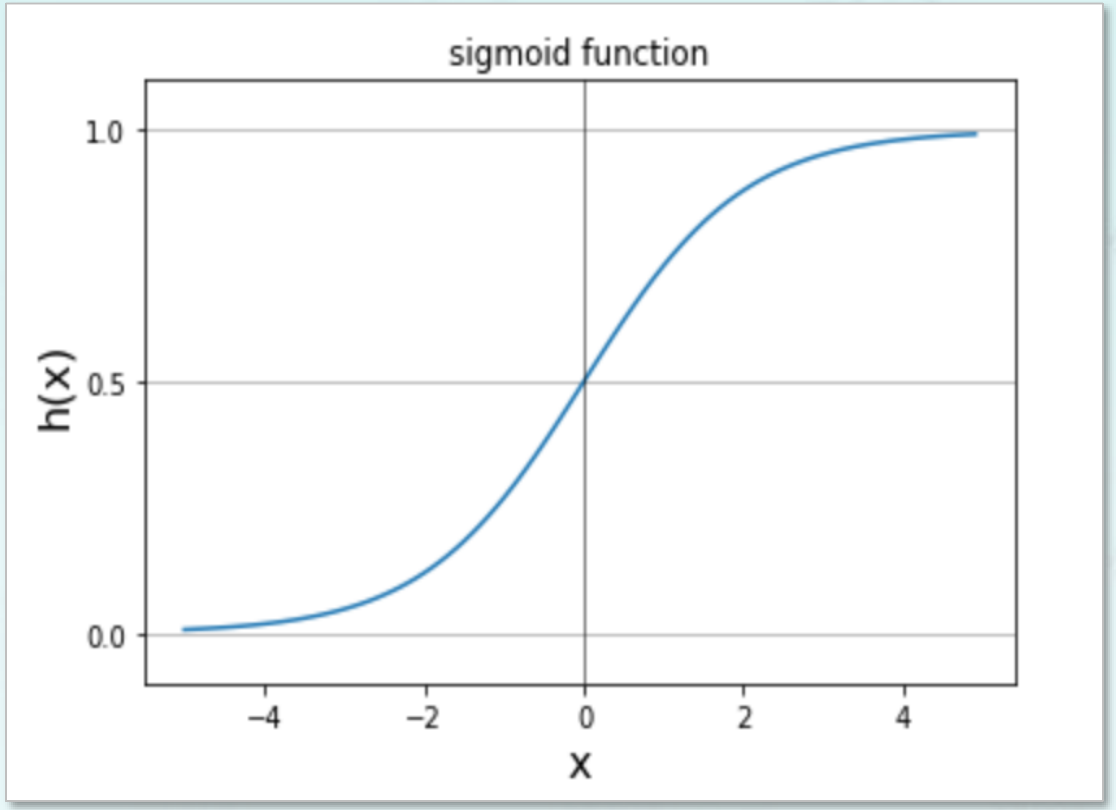
- 쌍곡 탄젠트

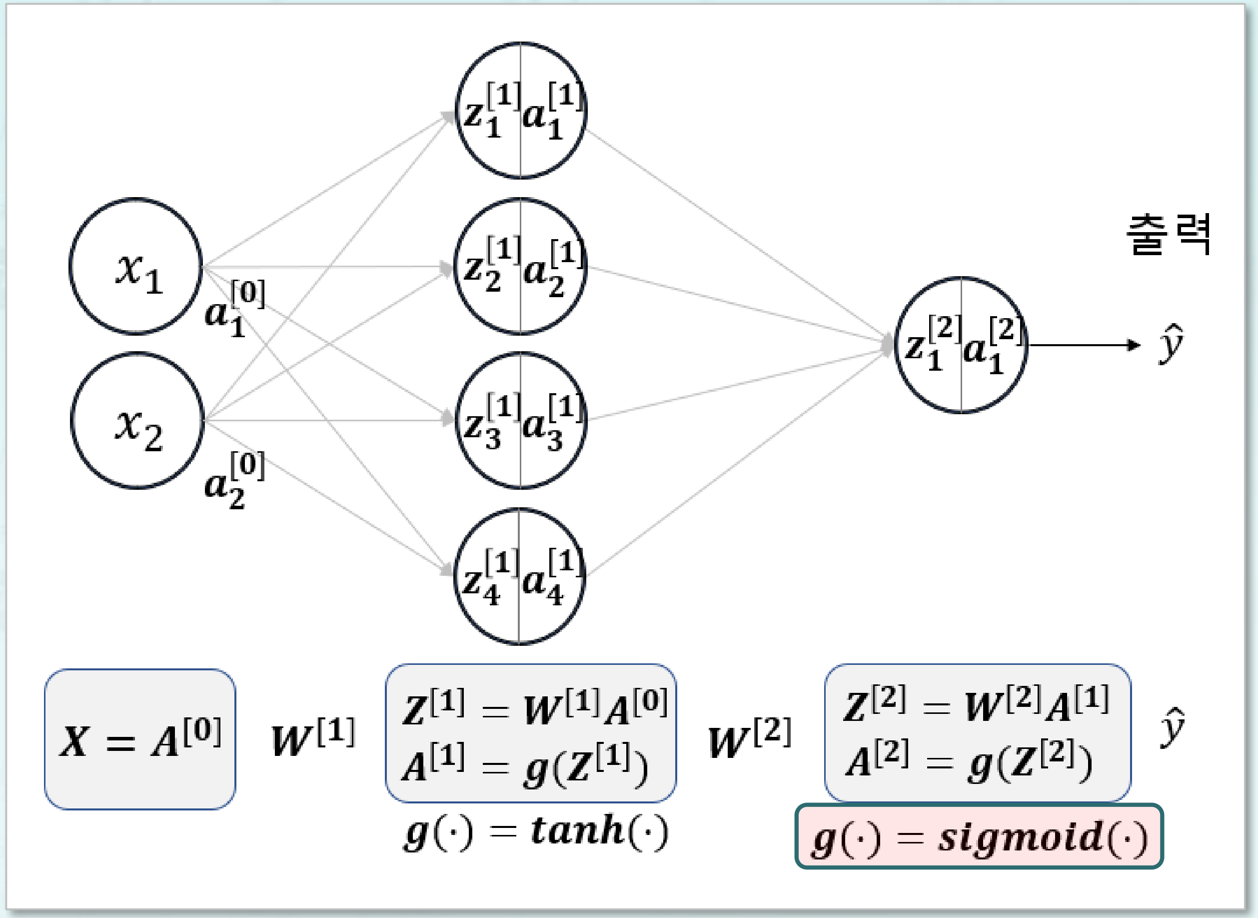


# 2. 신경망 모델: 3층 신경망

- 시그모이드

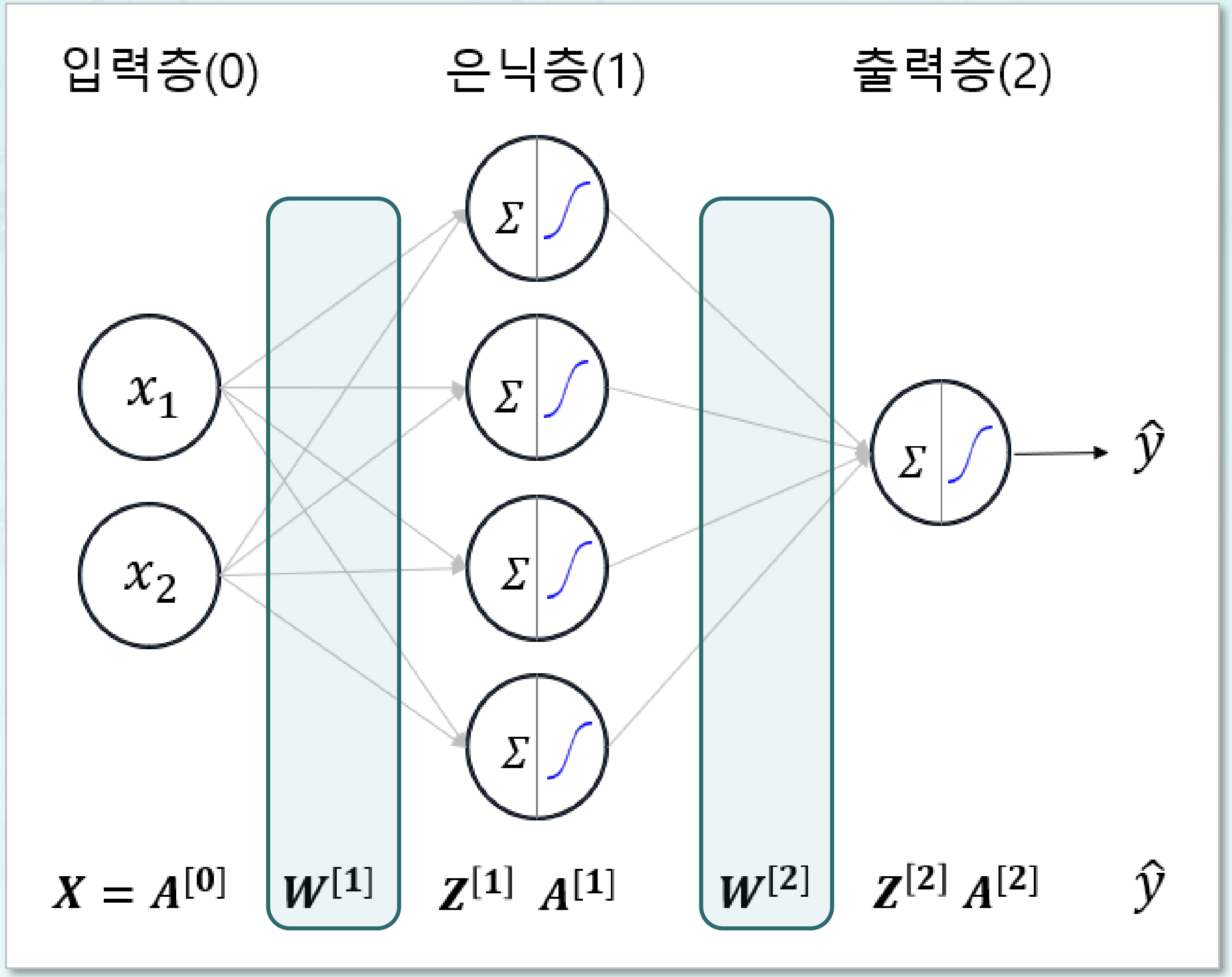


# 2. 신경망 모델: 3층 신경망

- 제곱 합 오차(**SSE)**

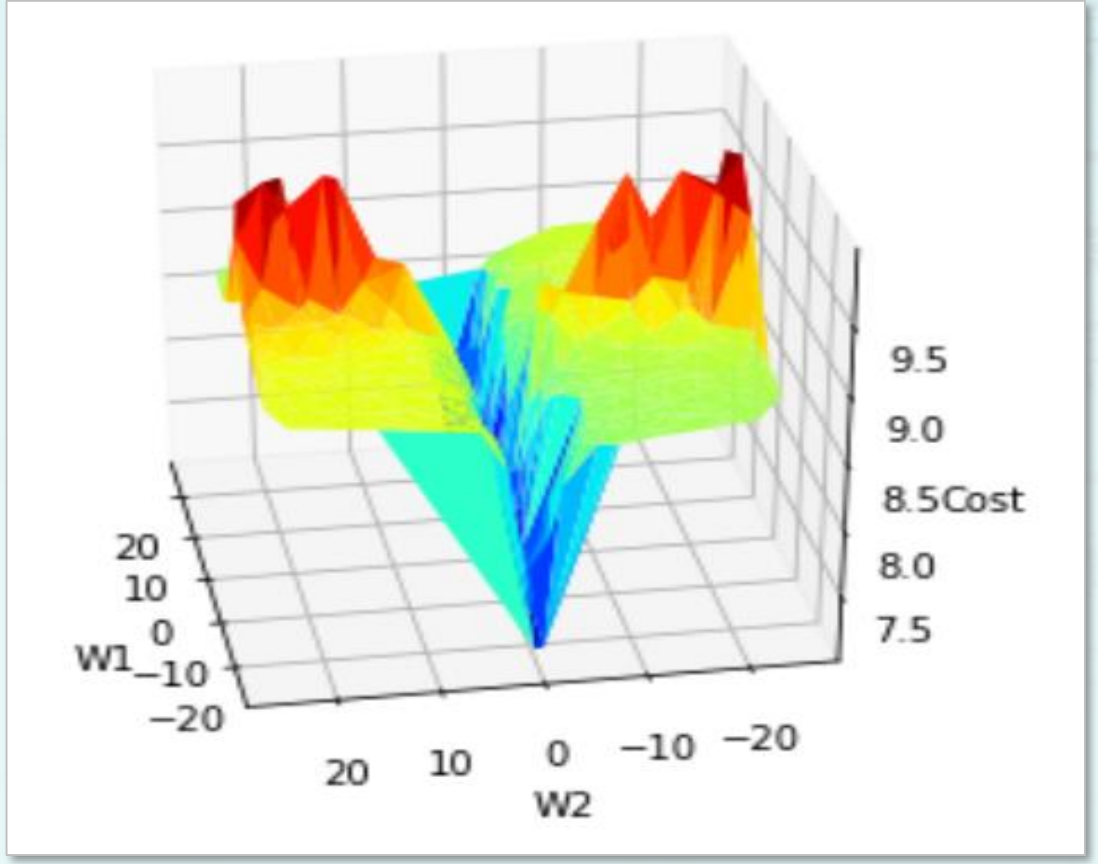
$$J(h(z), y) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\left(h(z) - y^{(i)}\right)^2$$



## 2. 신경망 모델: 3층 신경망

- 제곱 합 오차(**SSE)**

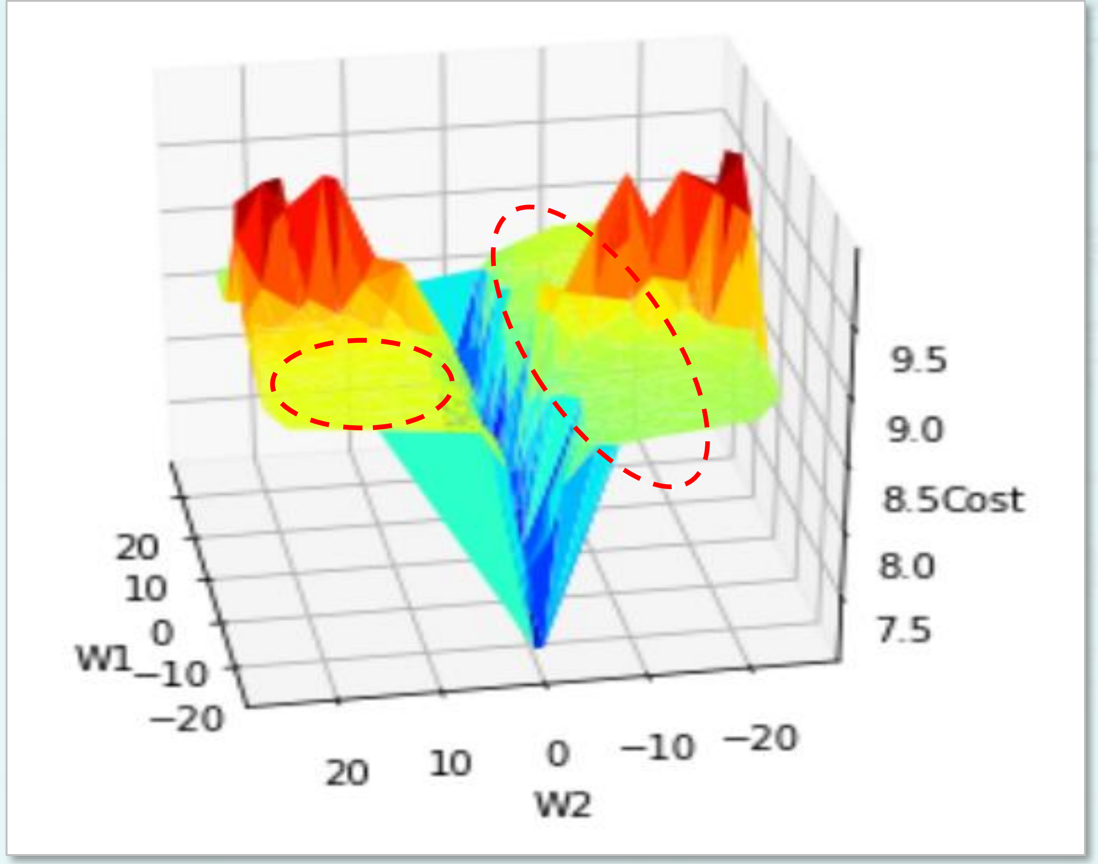
$$J(h(z), y) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} \left( h(z) - y^{(i)} \right)^2$$



# 2. 신경망 모델: 3층 신경망

- 제곱 합 오차(**SSE)**

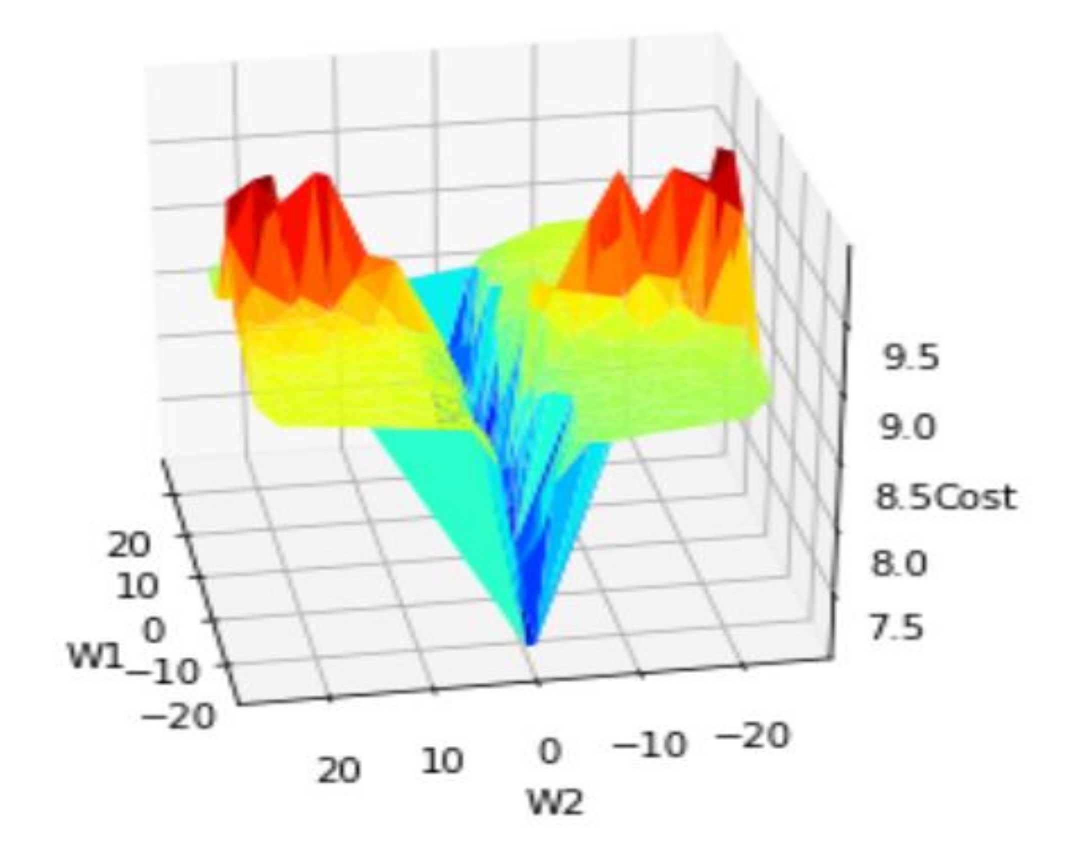
$$J(h(z), y) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\left(h(z) - y^{(i)}\right)^2$$



## 2. 신경망 모델: 3층 신경망

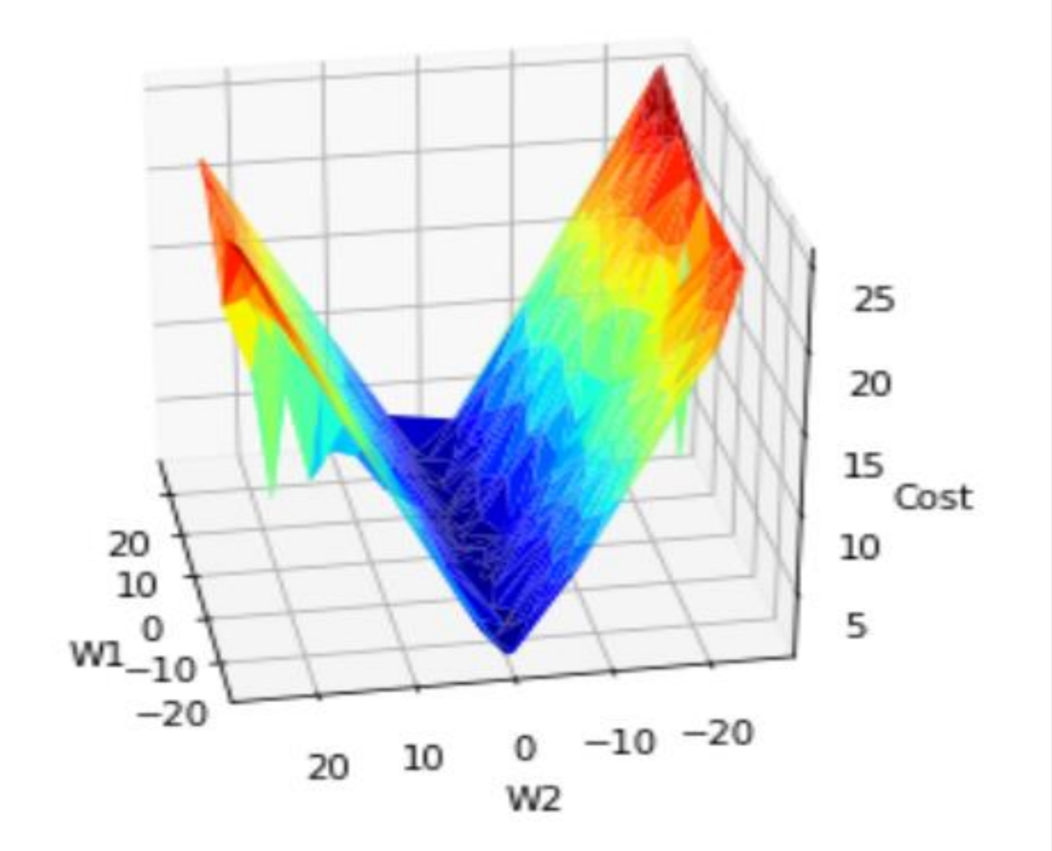
- 제곱 합 오차(**SSE**)

$$J(h(z), y) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\left(h(z) - y^{(i)}\right)^2$$




- 교차 엔트로피(**Cross Entropy**)

$$J(h(z), y) = -\sum_{i} y^{(i)}\log(\hat{y}^{(i)})$$



# 3. 로지스틱 회귀: 로지스틱 vs 선형 회귀

- 로지스틱 회귀
  - 결과 값 : 확률**(0 ~ 1)**

# 3. 로지스틱 회귀: 로지스틱 vs 선형 회귀

- 로지스틱 회귀
  - 결과 값 : 확률**(0 ~ 1)**
  - **Ex.)** 비 올 확률**,** 카드 패턴 진단

# 3. 로지스틱 회귀: 로지스틱 vs 선형 회귀

- 로지스틱 회귀
  - 결과 값 : 확률**(0 ~ 1)**
  - **Ex.)** 비 올 확률**,** 카드 패턴 진단

- 선형 회귀 분석
  - 연속적인 값 예측**(**하나의 값**)**
  - **Ex.)** 집 값 예측

- 선형 회귀 함수

$$Z = W \cdot X$$

# 3. 로지스틱 회귀: 로지스틱 vs 선형 회귀

- 로지스틱 회귀
  - 결과 값 : 확률**(0 ~ 1)**
  - **Ex.)** 비 올 확률**,** 카드 패턴 진단

- 선형 회귀 분석
  - 연속적인 값 예측**(**하나의 값**)**
  - **Ex.)** 집 값 예측

- 선형 회귀 함수

$$Z = W \cdot X$$

# 3. 로지스틱 회귀: 로지스틱 vs 선형 회귀

- 로지스틱 회귀
  - 결과 값 : 확률(0 ~ 1)
  - Ex.) 비 올 확률, 카드 패턴 진단

- 로지스틱 함수

$$h(z) = \frac{1}{1 + e^{-W \cdot X}} = \frac{1}{1 + e^{-Z}}$$

- 선형 회귀 분석
  - 연속적인 값 예측(하나의 값)
  - Ex.) 집 값 예측

- 선형 회귀 함수

$$Z = W \cdot X$$

# 로지스틱 회귀

- 학습 정리
  - 오차 함수들의 분석과 차이
  - **3**층 신경망에 두 개의 활성화 함수를 사용하기
  - 로지스틱 회귀 알고리즘 익히기
- 차시 예고
  - **11-1** 로지스틱 회귀 **2**

11주차(2/3)

# 로지스틱 회귀 1

파이썬으로 배우는 기계학습

한동대학교
김영섭 교수

여러분 곁에 항상 열려 있는 K-MOOC 강의실에서 만나 뵙기를 바랍니다.

# 3. 로지스틱 회귀: 로지스틱 vs 선형 회귀

- 로지스틱 회귀
  - 결과 값 : 확률**(0 ~ 1)**
  - **Ex.)** 비 올 확률,
    카드 패턴 진단
  - 분류
  - 승산비**(odds ratio)**

$$\frac{Sucess}{Fail} = \frac{p}{1-p} \qquad \begin{cases} 0 & \text{if } p \to 0. \\ +\infty & \text{if } p \to 1. \end{cases}$$

- 선형 회귀 분석
  - 연속적인 값 예측**(**하나의 값**)**
  - **Ex.)** 집 값 예측

$$Z = W \cdot X$$

# 3. 로지스틱 회귀: 로짓 함수

- 승산비(odds ratio)

$$\frac{Sucess}{Fail} = \frac{p}{1-p} \qquad \begin{cases} 0 & \text{if } p \to 0. \\ +\infty & \text{if } p \to 1. \end{cases}$$

# 3. 로지스틱 회귀: 로짓 함수

- 승산비(odds ratio)

$$\frac{Sucess}{Fail} = \frac{p}{1-p} \qquad \begin{cases} 0 & \text{if } p \to 0. \\ +\infty & \text{if } p \to 1. \end{cases}$$

- 로짓 함수(logit function)

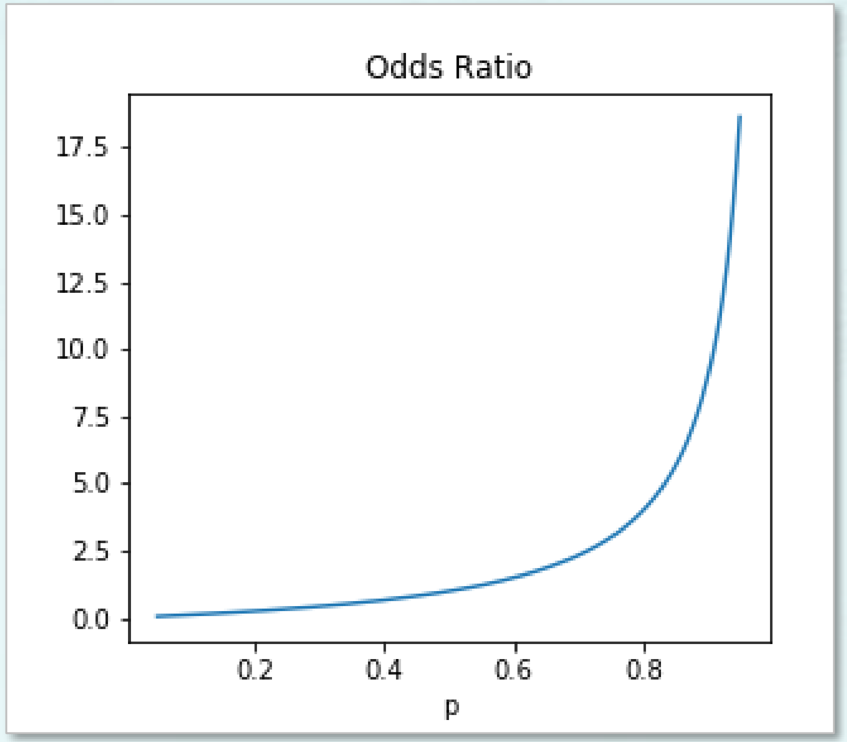
$$logit(odds\ ratio) = \log_e \frac{p}{1-p} \qquad \begin{cases} -\infty & \text{if } p \to 0. \\ +\infty & \text{if } p \to 1. \end{cases}$$

$$e^{-\infty} = 0$$

# 3. 로지스틱 회귀: 로짓 함수

- 승산비(odds ratio)

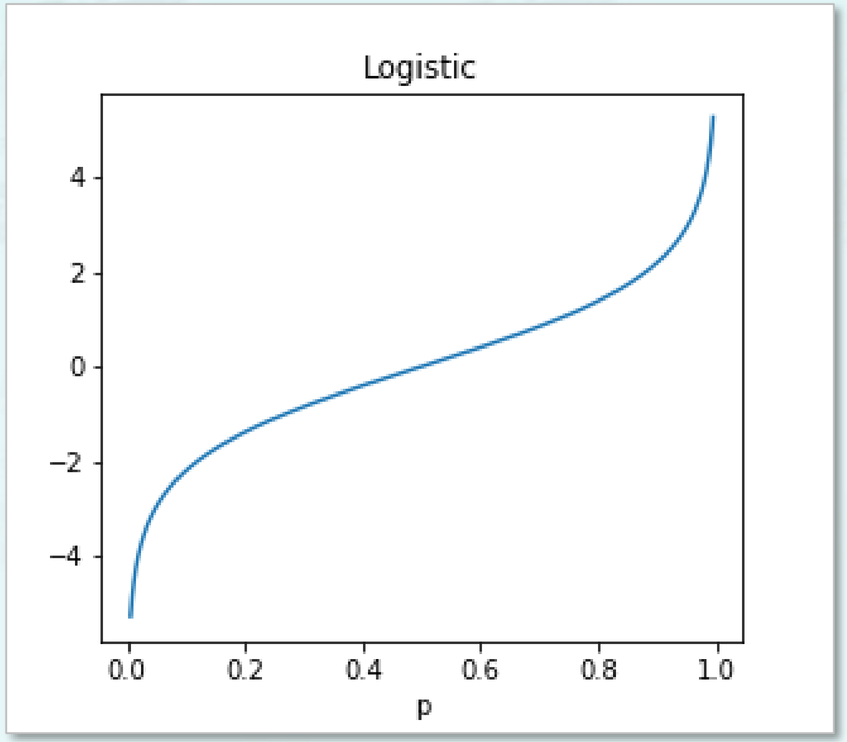
$$\frac{Sucess}{Fail} = \frac{p}{1-p} \qquad \begin{cases} 0 & \text{if } p \to 0. \\ +\infty & \text{if } p \to 1. \end{cases}$$




- 로짓 함수(logit function)

$$logit(odds\ ratio) = \log_e \frac{p}{1-p} \qquad \begin{cases} -\infty & \text{if } p \to 0. \\ +\infty & \text{if } p \to 1. \end{cases}$$

$$e^{-\infty} = 0$$



# 3. 로지스틱 회귀: 로짓 함수

- 로짓 함수**(logit function)**

$$\log_e \frac{p}{1-p} = W \cdot X$$

# 3. 로지스틱 회귀: 로짓 함수

- 로짓 함수**(logit function)**

$$\log_e \frac{p}{1-p} = W \cdot X$$

$$\frac{p}{1-p} = e^{W \cdot X}$$

# 3. 로지스틱 회귀: 로짓 함수

$$
\begin{aligned}
odds\ ratio = \frac{1-p}{p} &= \frac{1}{e^{W\cdot X}} \\
\frac{1}{p} - 1 &= \frac{1}{e^{W\cdot X}} \\
\frac{1}{p} &= \frac{1}{e^{W\cdot X}} + 1 \\
&= \frac{1}{e^{W\cdot X}} + \frac{e^{W\cdot X}}{e^{W\cdot X}} \\
&= \frac{1 + e^{W\cdot X}}{e^{W\cdot X}} \\
p &= \frac{e^{W\cdot X}}{1 + e^{W\cdot X}}
\end{aligned}
$$

- 로짓 함수(**logit function**)

$$\log_e \frac{p}{1-p} = W \cdot X$$

$$\frac{p}{1-p} = e^{W\cdot X}$$

# 3. 로지스틱 회귀: 로짓 함수

$$
\begin{aligned}
odds\ ratio = \frac{1-p}{p} &= \frac{1}{e^{W\cdot X}} \\
\frac{1}{p} - 1 &= \frac{1}{e^{W\cdot X}} \\
\frac{1}{p} &= \frac{1}{e^{W\cdot X}} + 1 \\
&= \frac{1}{e^{W\cdot X}} + \frac{e^{W\cdot X}}{e^{W\cdot X}} \\
&= \frac{1+e^{W\cdot X}}{e^{W\cdot X}} \\
p &= \frac{e^{W\cdot X}}{1+e^{W\cdot X}}
\end{aligned}
$$

- 로짓 함수**(logit function)**

$$
\begin{aligned}
p &= \frac{e^{W\cdot X}}{1+e^{W\cdot X}} \\
&= \frac{1}{\frac{1}{e^{W\cdot X}} + 1} \\
&= \frac{1}{1+e^{-W\cdot X}} \\
h(z) &= \frac{1}{1+e^{-W\cdot X}} = \frac{1}{1+e^{-Z}}
\end{aligned}
$$