

# 7. 다중 클래스 분류


## Softmax Regression

### Softmax

- 여러 개의 클래스를 분류
  - ex. 고양이, 개, 병아리 분류
- output layer의 unit이 클래스 수와 같음
- output: 각 클래스의 확률 (합계 1)

Recognizing cats, dogs, and baby chicks, *other*

*1 2 3 0*



*3 1 2 0 3 2 0 1*

$C = \#classes = 4$  *(0, ..., 3)*

$X \rightarrow$  [Diagram of a neural network with 6 hidden layers and 1 output layer]

$\hat{y}$

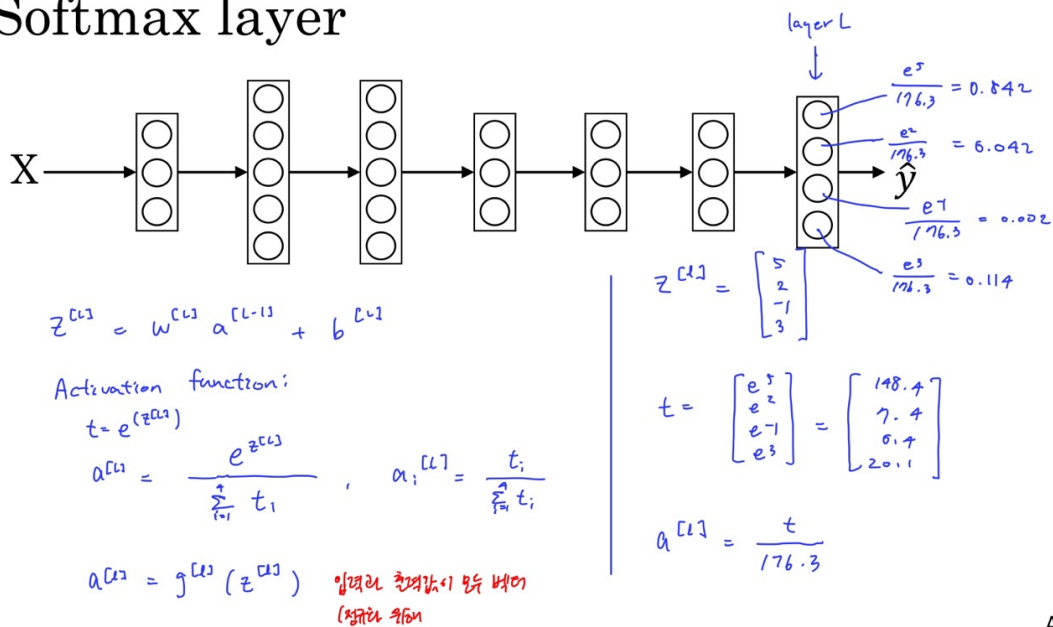
$P(other|x)$   
 $P(cat|x)$   
 $P(dog|x)$   
 $P(bel|x)$

$\hat{C}_n^{[L]} = 4 = C$

Andrew Ng

### Softmax layer

# Softmax layer

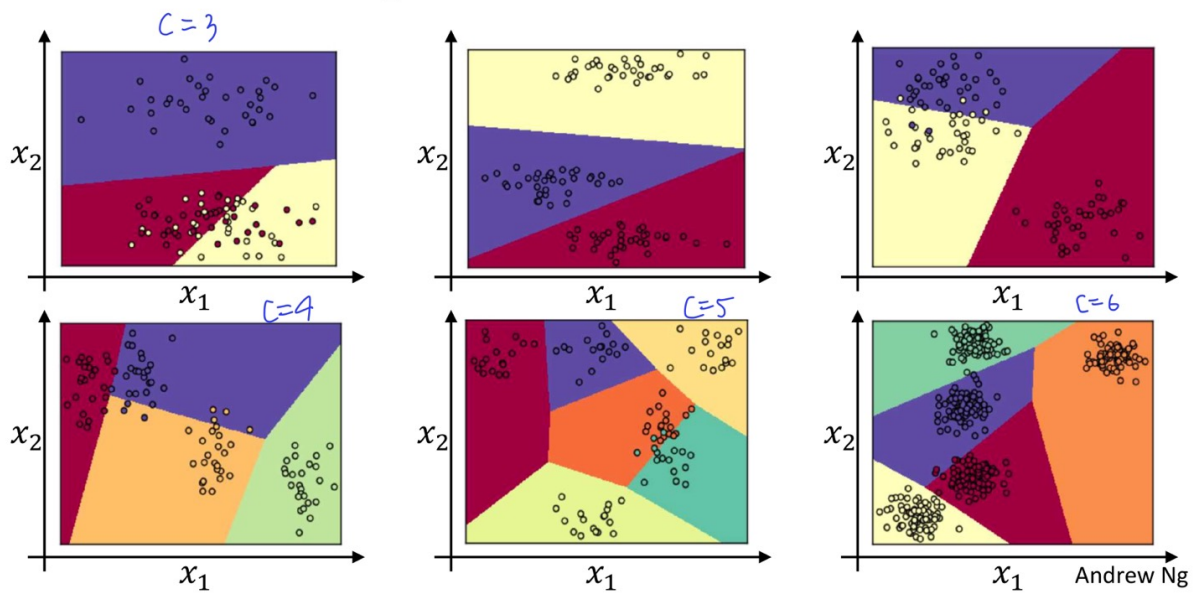


Andrew Ng

- 마지막 층의 출력값이 주어졌을 때 해당 클래스에 속할 확률을 Softmax 층을 통해 구할 수 있음
- 마지막 선형 출력값( $z$ )들을 지수화해  $t = e^Z$ 를 만들고, 모든 값들의 합이 1이 될 수 있도록 모든  $t$ 의 합으로 나눠서 정규화
- $$a_i = \frac{e^{z_i}}{\sum_{j=1}^C e^{z_j}}$$

## Softmax examples

## Softmax examples



## Training Softmax Classifier

### Loss function

Loss function

$$y = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, a^{[2]} = \hat{y} = \begin{bmatrix} 0.3 \\ 0.2 \\ 0.2 \end{bmatrix}$$

$$L(\hat{y}, y) = - \sum_{j=1}^C y_j \log \hat{y}_j \quad \left| \quad J(\omega^{[1]}, b^{[1]}, \dots) = \frac{1}{m} \sum_{i=1}^m L(\hat{y}^{(i)}, y^{(i)}) \right.$$

$$\Rightarrow -y_2 \log_2 \hat{y}_2 = -\log \hat{y}_2$$

$\Rightarrow$  make  $\hat{y}_2$  big as possible

$$Y = [y^{(1)} \ y^{(2)} \ \dots \ y^{(n)}]$$

$$= \begin{bmatrix} p & 0 & & \\ 1 & 0 & & \\ 0 & 1 & \dots & \\ 0 & 0 & & \end{bmatrix}$$

$$\bullet \quad L(\hat{y}, y) = - \sum_{j=1}^4 y_j \log y_j$$

## Gradient descent

- Backpropagation
  - $dz^{[L]} = \hat{y} - y$