deeplearning.ai

Multi-class classification

Softmax regression

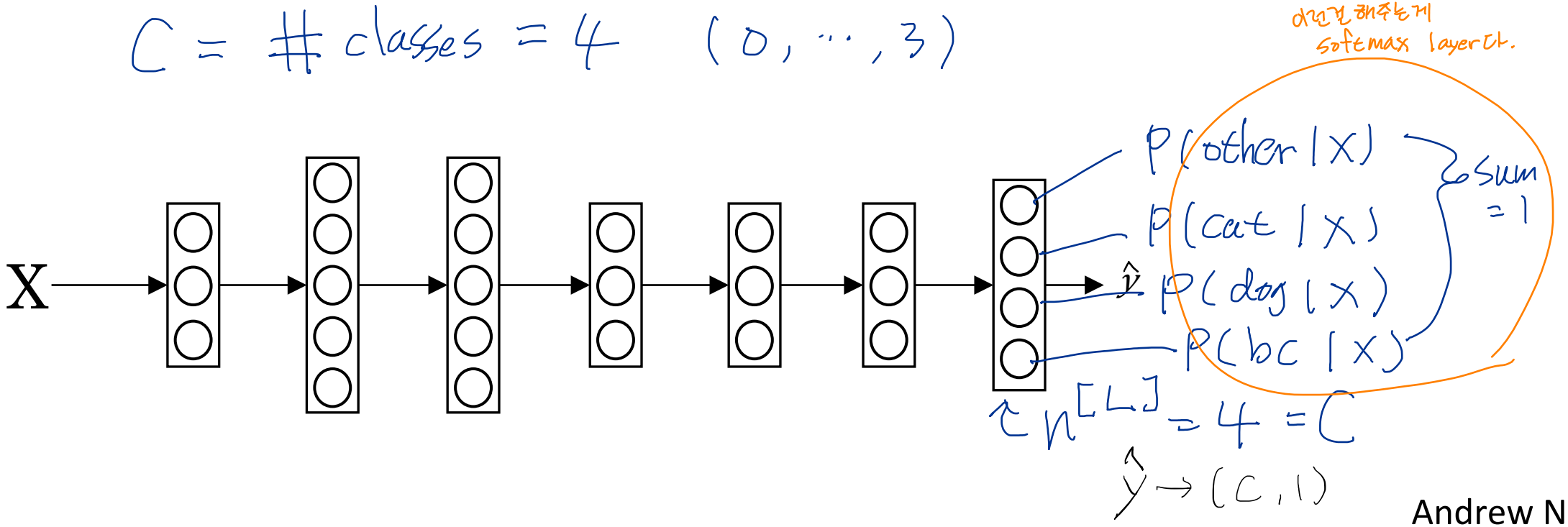
softmax regression : generalization of logistic regression

Recognizing cats, dogs, and baby chicks, other

1 2 3 0

3 1 2 0 3 2 0 1

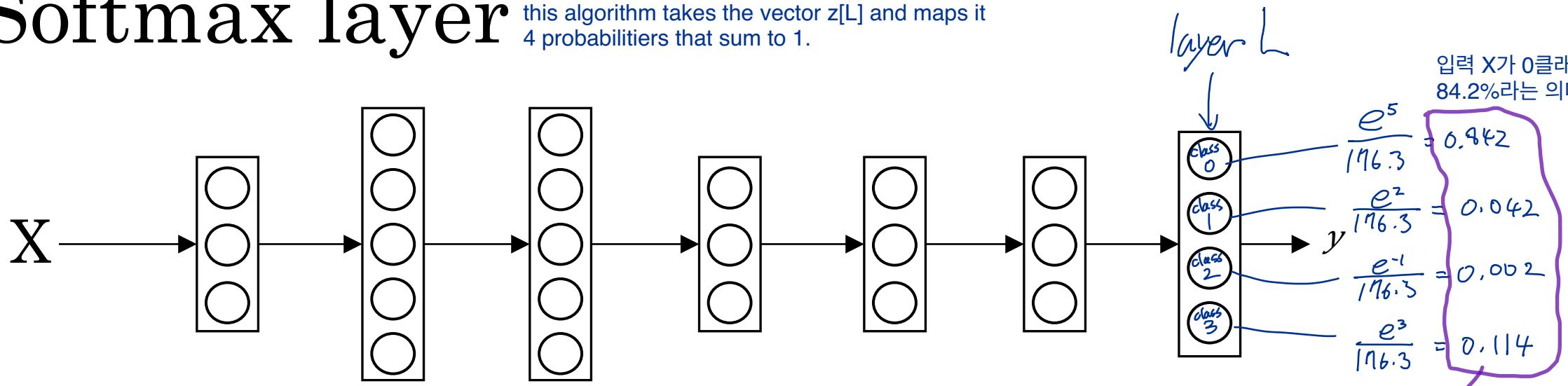
$C = \#\text{classes} = 4 \quad (0, \cdots, 3)$



Andrew Ng

Softmax layer

this algorithm takes the vector z[L] and maps it 4 probabilitiers that sum to 1.



$$z^{[L]} = W^{[L]} a^{[L-1]} + b^{[L]} \quad (4,1)$$

$$z^{[L]} = \begin{bmatrix} 5 \\ 2 \\ -1 \\ 3 \end{bmatrix}$$

(softmax) Activation function:

$$t = e^{(z^{[L]})} \quad (4,1) \quad \text{(element-wise 연산)}$$

$$a^{[L]} = \frac{e^{z^{[L]}}}{\sum_{j=1}^{4} t_i}, \quad a_i^{[L]} = \frac{t_i}{\sum_{j=1}^{4} t_i}$$

(sum이 1이 되도록 normalize 하기위함)

$$t = \begin{bmatrix} e^5 \\ e^2 \\ e^{-1} \\ e^3 \end{bmatrix} = \begin{bmatrix} 148.4 \\ 7.4 \\ 0.4 \\ 20.1 \end{bmatrix} \quad \sum_{j=1}^{4} t_j = 176.3$$

$$a^{[L]} = \frac{t}{176.3}$$

$a^{[L]}$: (4,1) → $\hat{y}$

마지막으로 이 layey의 특이한 점은, 우리가 여태껏 봤던 activation function들은 single row value input을 받았지만 softmax activation function은 (4,1)벡터를 받아서 (4,1) 벡터를 출력한다. normalization 과정이 필요하기 때문이다.

Andrew Ng

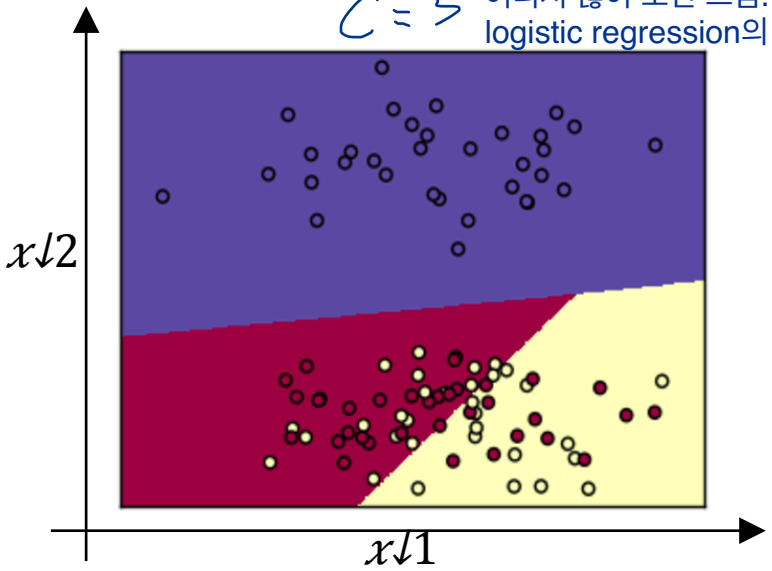
Softmax examples

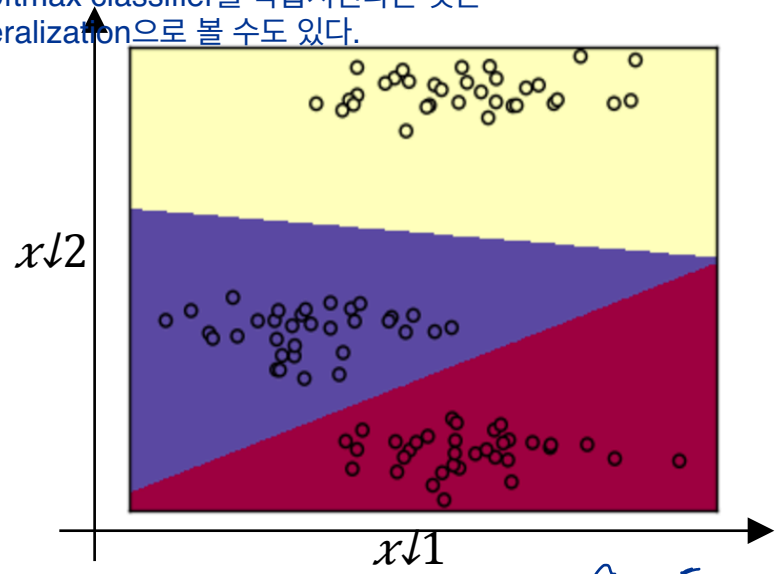
$x_1 \rightarrow \boxed{\quad} \rightarrow \hat{y}$
$x_2 \nearrow$

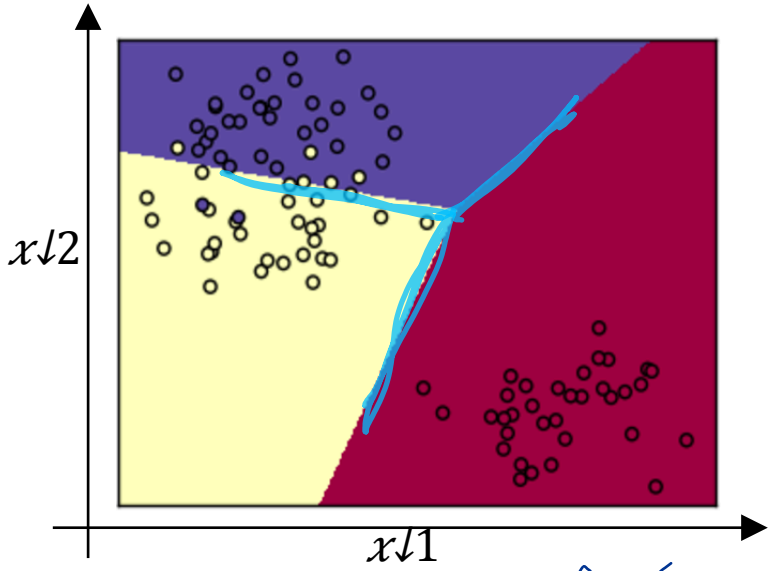
$z^{[1]} = W^{[1]}x + b^{[1]}$
$a^{[1]} = \hat{y} = g(z^{[1]})$

hidden layer가 없는 단순한 case를 가지고 softmax function이 어떤 representation을 가질수있는지 살펴보자.

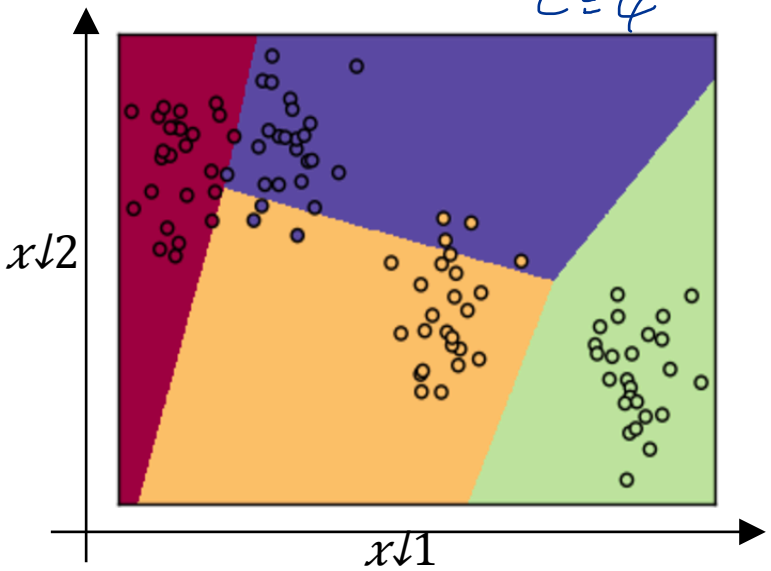
$C = 2$ 어디서 많이 보던 그림. 즉, softmax classifier를 학습시킨다는 것은 logistic regression의 generalization으로 볼 수도 있다.



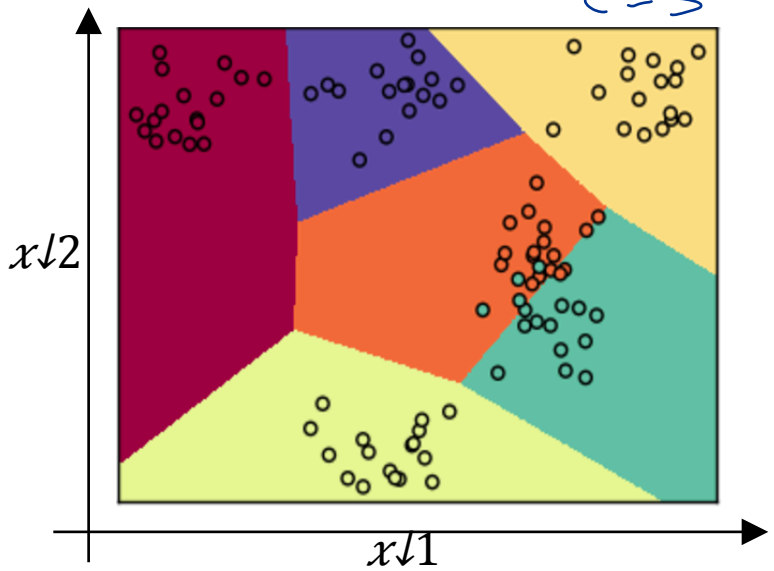




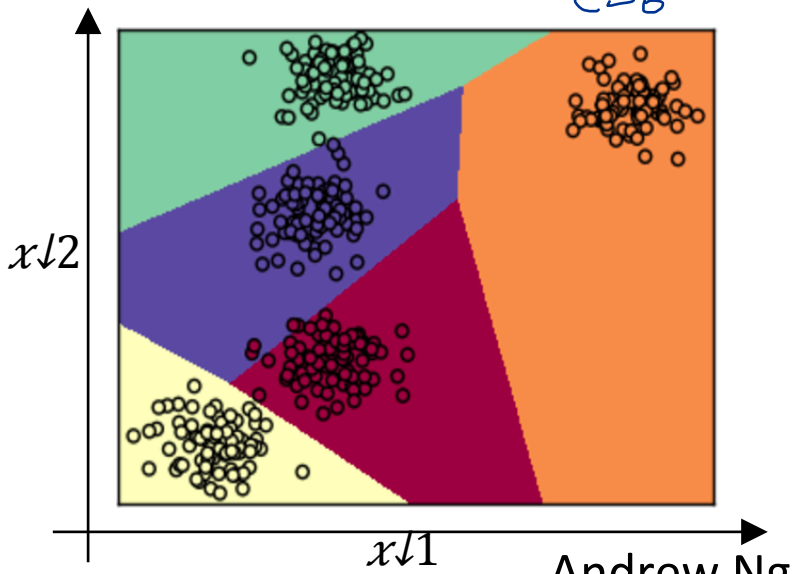
$C = 4$



$C = 5$



$C = 6$



히든 레이어들을 추가해주면 complex non-linear decision boundaries를 학습할 수 있을 것이다.

Andrew Ng