deeplearning.ai

Multi-class classification

Trying a softmax classifier

Understanding softmax

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

"hard max"

softmax라는 이름은 hard max의 반대라는 의미에서 가져온 것인데 hard max는 벡터 z를 보고 가장 큰 element의 자리만 1로 하고 나머지는 0

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

반면 softmax는 z에서 이 확률 값으로 가는 more gentle mapping이다.

$$a^{[L]} = g^{[L]}(z^{[L]}) = \begin{bmatrix} e^5 / (e^5+e^2+e^{-1}+e^3) \\ e^2 / (e^5+e^2+e^{-1}+e^3) \\ e^{-1} / (e^5+e^2+e^{-1}+e^3) \\ e^3 / (e^5+e^2+e^{-1}+e^3) \end{bmatrix} = \begin{bmatrix} 0.842 \\ 0.042 \\ 0.002 \\ 0.114 \end{bmatrix}$$

Softmax regression generalizes logistic regression to C classes.

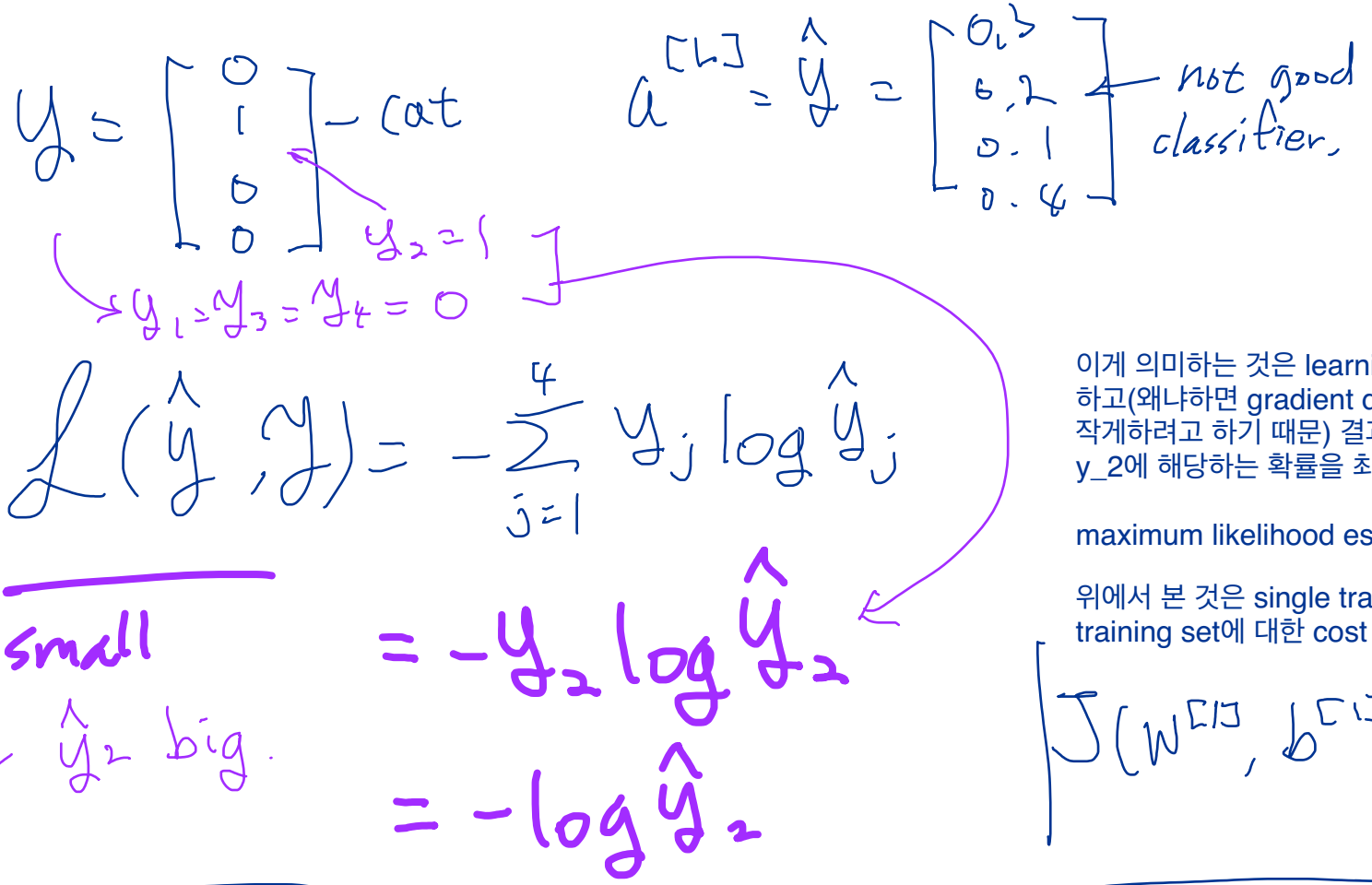
If C=2, softmax reduces to logistic regression.

$a^{[L]} =$ [0.842, ~~0.158~~] ↳ redundant computation

Andrew Ng

이제 softmax output layer로 neural network를 어떻게 학습하는지 알아보자. 그러려면 학습에 사용할 loss function을 정의해야된다.

Loss function



이 때 loss function은 아래와 같이 정의된다.

이게 의미하는 것은 learning algorithm이 L(y_hat, y)를 작게 만들려고 하고(왜냐하면 gradient descent가 트레이닝 데이터셋 상에서 loss를 작게하려고 하기 때문) 결과적으로 y_hat_2 즉, ground truth가 1인 y_2에 해당하는 확률을 최대한 크게 만드는 것이다.

maximum likelihood estimation의 형태임

위에서 본 것은 single training example에 대한 loss이다. entire training set에 대한 cost function J의 경우는 어떨가?

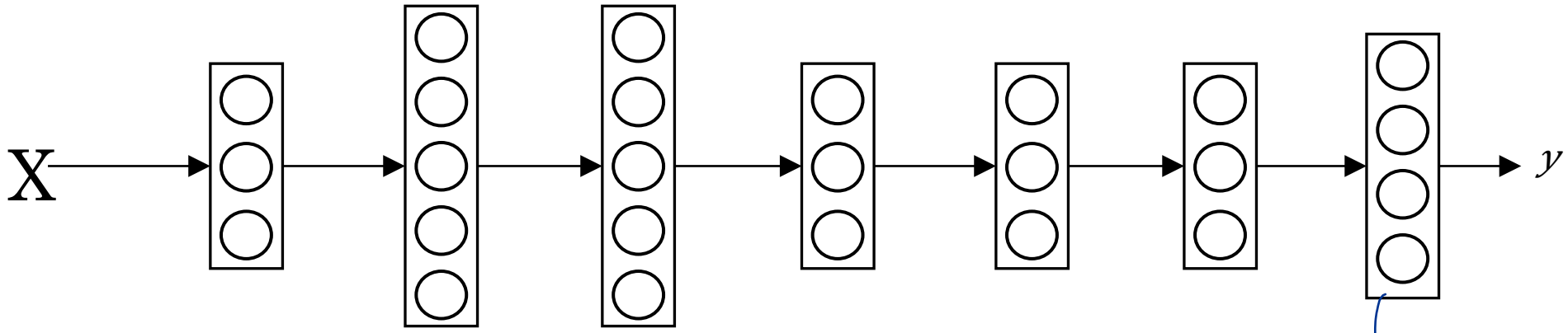
$$J(W^{[1]}, b^{[1]}, \ldots) = \frac{1}{m} \sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)})$$

$$Y = [y^{(1)}, y^{(2)}, \ldots y^{(m)}] = \begin{bmatrix} 0 & 0 & 1 & \ldots \\ 1 & 0 & 0 & \\ 0 & 1 & 0 & \\ 0 & 0 & 0 & \end{bmatrix}_{(4,m)}$$

$$\hat{Y} = [\hat{y}^{(1)}, \ldots, \hat{y}^{(m)}] = \begin{bmatrix} 0.3 & \ldots \\ 0.2 & \\ 0.1 & \\ 0.4 & \end{bmatrix}_{(4,m)}$$

Andrew Ng

Summary of softmax classifier



$z^{[L]}$ → softmax activation function → $a^{[L]} = \hat{y}$ → $\mathcal{L}(\hat{y}, y)$

$z^{[L]}$: $(4,1)$

Backprop : $dz^{[L]} = \hat{y} - y$ $(4,1)$

gradient descent를 위한 back propagation 과정에서 key step은 last layer에서 dz에 대한 derivative를 구하는 것이다.

$\frac{\partial J}{\partial z^{[L]}}$ (partial derivative of the cost function with respect to z[L])

근데 사실 forward 과정만 잘 정의하면 프레임워크에서 backward 과정은 알아서 잘 해준다.

Andrew Ng