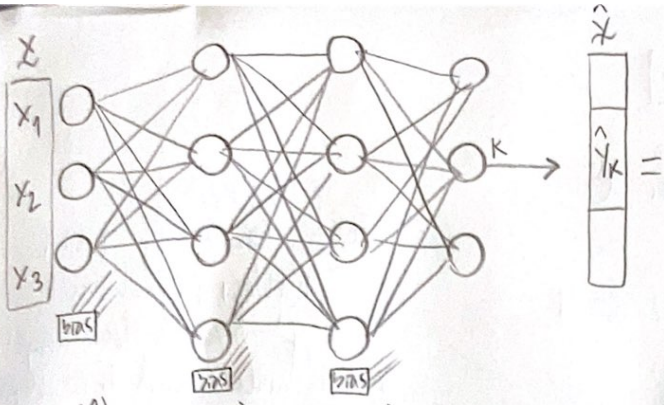


$\underset{\sim}{a}^{(1)}$ INPUT — $\underset{\sim}{a}^{(2)}$ $\underset{\sim}{a}^{(3)}$ HIDDEN — $\underset{\sim}{a}^{(4)}$ OUTPUT

layer 1 layer 2 layer 3 layer 4

X: input

a: output after activation function

z: output before activation function

f: activation function, sigmoid

$\underset{\sim}{W}^{(l)}$: weight matrix from layer l to layer $l+1$ of size $S_{l+1} \times (S_l + 1)$

S_{l+1}: units in layer $l+1$; S_l: units in layer l; +1: bias

FORWARD PASS

$\underset{\sim}{a}^{(1)} = \underset{\sim}{x}$

↳ append bias: $\underset{\sim}{a}^{(1)} \leftarrow [a_0^{(1)} = 1, \underset{\sim}{a}^{(1)T}]^T = [1\ x_1\ x_2\ x_3]^T$ $(3+1)\times 1$, 4

$\underset{\sim}{z}^{(2)} = \underset{\sim}{W}^{(1)} \cdot \underset{\sim}{a}^{(1)}$ $(4\times 4)\times(4\times 1) = 4\times 1$

$\underset{\sim}{a}^{(2)} = f(\underset{\sim}{z}^{(2)})$

↳ append bias: $\underset{\sim}{a}^{(2)} \leftarrow [a_0^{(2)} = 1, \underset{\sim}{a}^{(2)T}]^T$ $(4+1)\times 1$, 5

$\underset{\sim}{z}^{(3)} = \underset{\sim}{W}^{(2)} \cdot \underset{\sim}{a}^{(2)}$ $(4\times 5)\times(5\times 1) = 4\times 1$

$\underset{\sim}{a}^{(3)} = f(\underset{\sim}{z}^{(3)})$

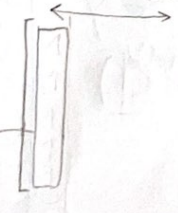
↳ append bias: $\underset{\sim}{a}^{(3)} \leftarrow [a_0^{(3)} = 1, \underset{\sim}{a}^{(3)T}]^T$ $(4+1)\times 1$, 5

$\underset{\sim}{z}^{(4)} = \underset{\sim}{W}^{(3)} \cdot a^{(3)}$ $(3\times 5)\times(5\times 1) = 3\times 1$

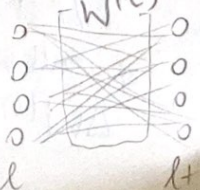
$\underset{\sim}{a}^{(4)} = f(z^{(4)}) = h_W(\underset{\sim}{x}) : 3\times 1$

↳ model / neural network.

Note: $\underset{\sim}{W}^{(l)}$ =



$\underset{\sim}{W}^{(l)}$ is the weight matrix of connections that go from units in l to units in $l+1$



BACKWARD PASS

$\delta_j^{(l)}$: delta error of node/unit j in layer l

- For each output unit j in last layer $l=4$

$$\delta_j^{(4)} = \overbrace{(y_j - \hat{y}_j)}^{e^{(4)}} \cdot f'(z_j^{(3)})$$

In Udacity it's multiplied by $f'()$; Andrew Ng didn't use f' in the last layer, but I understand that depends in the activation function!

note: for the sigmoid: $f' = f \cdot (1-f)$; if no activation: $f(x) = x$, $f' = 1$.

in Udacity, the es are the errors, δ is the delta

↳ assemble: $\underset{\sim}{\delta}^{(4)} = [\delta_1^{(4)}, \dots, \delta_j^{(4)}, \dots]$ (num classes)

- Propagate errors to all units of all layers

$$\underset{\sim}{\delta}^{(3)} = \overbrace{\left(\underbrace{(\underset{\sim}{W}^{(3)})^T}_{5\times 3} \cdot \underbrace{\underset{\sim}{\delta}^{(4)}}_{3\times 1}\right)}^{e^{(3)}} .* \underbrace{[1, f'(\underset{\sim}{z}^{(3)})^T]}_{1\times 4}$$

5×1

multiply 1 by 1

extend because of bias, so that dims match, but then it's removed; I understand we could extend the bias at the end of the vector, too...

$$= [\underbrace{\delta_0^{(3)}}_{\text{bias}}, \delta_1^{(3)}, \dots]^T$$: each unit in layer 3 has an error.

$\delta^{(3)} \leftarrow \delta^{(3)}[1:]$ remove bias component : 4×1

$$\underset{\sim}{\delta}^{(2)} = \overbrace{\left(\underbrace{(\underset{\sim}{W}^{(2)})^T}_{5\times 4} \cdot \underbrace{\underset{\sim}{\delta}^{(3)}}_{4\times 1}\right)}^{e^{(2)}} .* \underbrace{[1, f'(z^{(2)})^T]}_{4\times 1}$$

5×1

$\delta^{(2)} \leftarrow \delta^{(2)}[1:]$ remove bias component : 4×1

initialized as $\Delta W = 0$

Note: if we are doing complete batch gradient descend, we update weights after each epoch, but if stochastic, we can update weights for every example pass

- Weight changes: For each sample:

$\hat{y}$ = forward $(\underset{\sim}{X}) \rightarrow a^{(1)}, a^{(2)}, a^{(3)}, a^{(4)} = \hat{y}$

backward $(\hat{y}) \rightarrow \delta^{(4)}, \delta^{(3)}, \delta^{(2)}$ (no $\delta^{(1)}$!)

$$\Delta \underset{\sim}{W}^{(l)} = \Delta \underset{\sim}{W}^{(l)} + \underbrace{\delta^{(l+1)}}_{s_{l+1}\times 1} \cdot \underbrace{a^{(l)T}}_{1\times(s_l+1)}$$

Then, after each epoch:

$$\underset{\sim}{W}^{(l)} = \underset{\sim}{W}^{(l)} + \underbrace{\frac{\alpha}{m} \Delta \underset{\sim}{W}^{(l)}}_{\text{step}}$$

old