CH3.4 Neural Networks

→ Add more layers between the input and output layers. This makes your model DEEPER leading to the term Deep learning.

→ If we just pass linear functions to linear functions the final result is again a linear function. To overcome this output of each layer is evaluated at a spesific nonlinear function called "activation function".

→ We can add as many layers we want. Every layer, just like input and output layers, is composed of nodes called "neuron". Every neuron is connected the output of the all neurons in the previous layers and the input of all neurons in the next layer. This is inspried by the actual brain neurons.

→ So we have a network of neurons. That's why we call this structure as Neural Network.

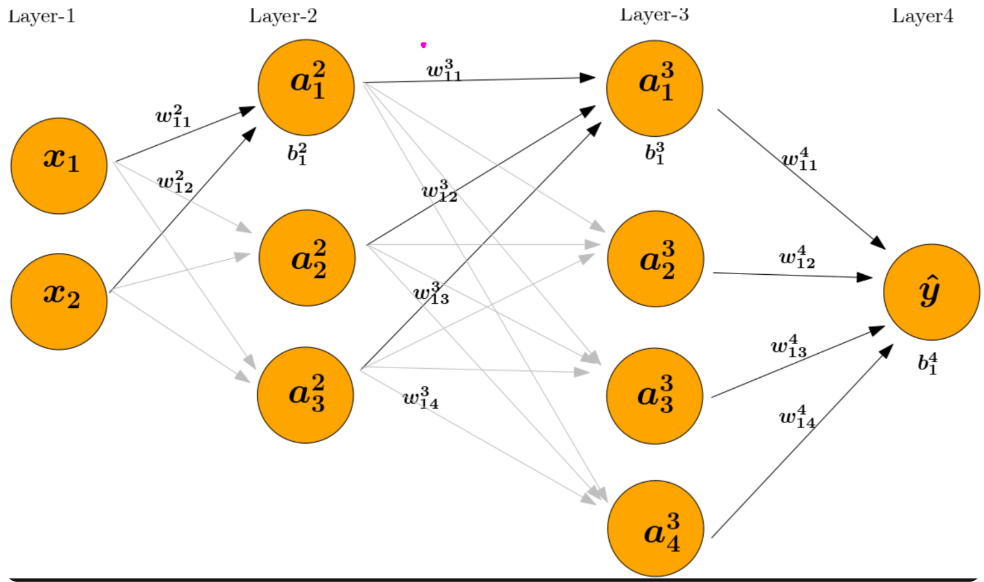
→ "I know a guy know who knows a guy"

→ Almost all neural network models have two components.

(1) Forward Pass : get the input and pass it through the layers all the way to the last layer and compute the cost (loss)

(2) Backpropagation: compute the derivative of the cost function with respect to ALL parameters and update them using gradient decent method

(1) Forward Pass



w^l_{jk} → weight from k^{th} neuron in layer-$(l-1)$ to the j^{th} neuron in layer-l

b^l_j → bias of the j^{th} neuron of layer-l

a^l_j → activation of j^{th} neuron in the l^{th} layer

<u>Ex:</u> w^2_{31} : weight from 3^{rd} neuron in layer-1 to the 1^{st} neuron in layer 2

b^2_1 : bias of the 1^{st} neuron in layer-2

From Layer-1 to Layer-2

$$
\begin{aligned}
z_1^2 &= w_{11}^2 x_1 + w_{12}^2 x_2 + b_1^2 \\
z_2^2 &= w_{21}^2 x_1 + w_{22}^2 x_2 + b_2^2 \\
z_3^2 &= w_{31}^2 x_1 + w_{32}^2 x_2 + b_3^2
\end{aligned}
\Rightarrow
\underbrace{\begin{bmatrix} z_1^2 \\ z_2^2 \\ z_3^2 \end{bmatrix}}_{z^{(2)}} = \underbrace{\begin{bmatrix} w_{11}^2 & w_{12}^2 \\ w_{21}^2 & w_{22}^2 \\ w_{31}^2 & w_{32}^2 \end{bmatrix}}_{W^{(2)}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{a^{(1)}} + \underbrace{\begin{bmatrix} b_1^2 \\ b_2^2 \\ b_3^2 \end{bmatrix}}_{b^{(2)}}
$$

Then apply nonlinear activation to obtain:

$$
\begin{bmatrix} a_1^2 \\ a_2^2 \\ a_3^2 \end{bmatrix} = \sigma \begin{pmatrix} z_1^2 \\ z_2^2 \\ z_3^2 \end{pmatrix} \Rightarrow a^{(2)} = \sigma\left(z^{(2)}\right)
$$

$$
\boxed{z^{(2)} = W^{(2)} a^{(1)} + b^{(2)} \Rightarrow a^{(2)} = \sigma\left(z^{(2)}\right)}
$$

From Layer-2 to Layer-3

$$
\begin{aligned}
z_1^3 &= W_{11}^3 a_1^2 + w_{12}^3 a_2^2 + w_{13}^3 a_3^2 + b_1^3 \\
z_2^3 &= W_{21}^3 a_1^2 + w_{22}^3 a_2^2 + w_{23}^3 a_3^2 + b_2^3 \\
z_3^3 &= W_{31}^3 a_1^2 + W_{32} a_2^2 + W_{33}^3 a_3^2 + b_3^3 \\
z_4^2 &= W_{41}^3 a_1^2 + W_{42} a_2^2 + W_{43}^3 a_3^2 + b_4^3
\end{aligned}
\Rightarrow
\underbrace{\begin{bmatrix} z_1^3 \\ z_2^3 \\ z_3^3 \\ z_4^2 \end{bmatrix}}_{z^3} = \underbrace{\begin{bmatrix} W_{11}^3 & w_{12}^3 & w_{13}^3 \\ W_{21}^3 & w_{22}^3 & w_{23}^3 \\ W_{31}^3 & W_{32}^3 & W_{33}^3 \\ W_{41}^3 & W_{42}^3 & W_{43}^3 \end{bmatrix}}_{W^3} \underbrace{\begin{bmatrix} a_1^2 \\ a_2^2 \\ a_3^2 \end{bmatrix}}_{a^2} + \underbrace{\begin{bmatrix} b_1^3 \\ b_2^3 \\ b_3^3 \\ b_4^3 \end{bmatrix}}_{b^3}
$$

Apply nonlinear activation

$$\begin{bmatrix} a_1^3 \\ a_2^3 \\ a_3^3 \\ a_4^3 \end{bmatrix} = \sigma\begin{pmatrix} z_1^3 \\ z_2^3 \\ z_3^3 \\ z_4^2 \end{pmatrix} \Rightarrow a^{(3)} = \sigma\left(z^{(3)}\right)$$

$$\boxed{z^{(3)} = W^{(3)} a^{(2)} + b^3 \Rightarrow a^{(3)} = \sigma\left(z^{(3)}\right)}$$

From Layer-3 to Layer-4

$$z_1^{(4)} = w_{11}^4 a_1^3 + w_{12}^4 a_2^3 + w_{13}^4 a_3^3 + w_{14}^4 a_4^3 + b_1^4$$

$$\underset{z^4}{z_1^{(4)}} = \underset{W^4}{\begin{bmatrix} w_{11}^4 & w_{12}^4 & w_{13}^4 & w_{14}^4 \end{bmatrix}} \underset{a^3}{\begin{bmatrix} a_1^3 \\ a_2^3 \\ a_3^3 \\ a_4^3 \end{bmatrix}} + \underset{b^4}{\begin{bmatrix} b_1^4 \end{bmatrix}}$$

Apply activation $a^4 = \sigma\left(z^4\right)$

$$\boxed{z^{(4)} = W^{(4)} a^{(3)} + b^{(4)} \Rightarrow a^{(4)} = \sigma\left(z^{(4)}\right)}$$

Summary

$$\boxed{\begin{aligned}
z^{(2)} &= W^{(2)} a^{(1)} + b^{(2)} &\Rightarrow\quad a^{(2)} &= \sigma(z^{(2)}) \\
z^{(3)} &= W^{(3)} a^{(2)} + b^{3} &\Rightarrow\quad a^{(3)} &= \sigma(z^{(3)}) \\
z^{(4)} &= W^{(4)} a^{(3)} + b^{(4)} &\Rightarrow\quad a^{(4)} &= \sigma(z^{(4)})
\end{aligned}}$$

Forward Pass

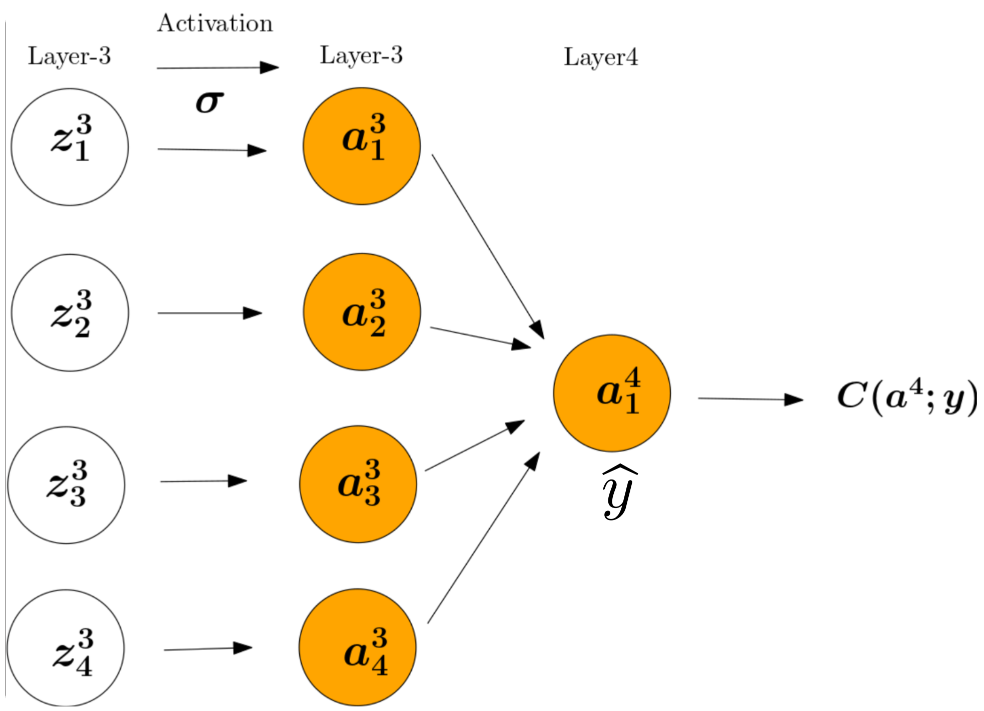
$$\boxed{z^{l} = W^{l} a^{l-1} + b^{l}, \quad a^{l} = \sigma(z^{l}), \; l = 2, 3, \dots L}$$

- In any deep learning publication, you should see pretty much the same form.

- Just like multiple linear regression, we will now compute the derivative of the cost function with respect to all weights w^{l}_{jk} and biases b^{l}_{j} in all layers

② Backpropagation

- Let's consider the picture in the very last layer. We don't need to specify a certain form for the cost function C and the nonlinear activation σ at this point.



- Convince yourself that the cost function C is a function of all weights w_{jk}^{l} and all biases b_j^{l} in the network. The ultimate goal is to find an expression for

$$\frac{\partial C}{\partial w_{jk}^{l}} \quad \text{and} \quad \frac{\partial C}{\partial b_j^{l}} \qquad l = 1, 2, \dots L$$

Let's define

$$\delta_1^{(4)} = \frac{\partial C}{\partial z_1^{(4)}}$$ → derivative of cost function with respect to the output of j^{th} neuron in last layer. In this case we have only 1 neuron. $j=1$

Using chain rule

$a_1^{(4)} = \sigma(z_1^{(4)})$ $\Rightarrow$ $$\boxed{\delta_1^{(4)} = \frac{\partial C}{\partial a_1^{(4)}} \frac{\partial a_1^{(4)}}{\partial z_1^{(4)}} = \frac{\partial C}{\partial a_1^{(4)}} \sigma'(z_1^{(4)})}$$

In general, this expression is a vector since we have more than one output neuron. We can write

$$\boxed{\delta^{(4)} = \nabla_a C \odot \sigma'(z^{(4)})}$$

$\odot$: elementwise multiplication. $\nabla_a C = [\partial C / \partial a^{(4)}]$

Note: Once we know $z_1^{(4)}$, we can compute δ_1^4, the derivative of cost function with respect to the last layer. Now we will propagate this error to the previous layers

Formula: the actual derivation of the formulas below can be found in my lecture notes I will post

$$\delta^l = (W^{l+1})^T \delta^{l+1} \odot \sigma'(z^l), \quad l = L-1, L-2, \cdots 2$$

Then we have

$$\underbrace{\frac{\partial C}{\partial w^l_{jk}} = a^{l-1}_k \delta^l_j \qquad \frac{\partial C}{\partial b^l_j} = 1 \cdot \delta^l_j}_{\text{component-wise}} \quad \Rightarrow \quad \underbrace{\boxed{\begin{array}{ll} \dfrac{\partial C}{\partial W^l} = \delta^l (a^{l-1})^T & \leftarrow \text{matrix} \\ \dfrac{\partial C}{b^l} = \delta^l & \leftarrow \text{vector} \end{array}}}_{\text{layer-wise}}$$

Now we can perform gradient decent "layer-wise"

$$\delta^l = (W^{l+1})^T \delta^{l+1} \odot \sigma'(z^l)$$

$$W^{l+1} = W^l - r \frac{\partial C}{\partial W^l} = W^l - \eta \cdot \delta^l (a^{l-1})^T \leftarrow \text{matrix}$$

$$b^{l+1} = b^l - r \frac{\partial C}{\partial b^l} = b^l - r\, \delta^l \qquad \leftarrow \text{vector}$$