Deep Neural Networks

Week 3

Content

Deep Neural Networks

Neural Network Notation

Forward Propagation

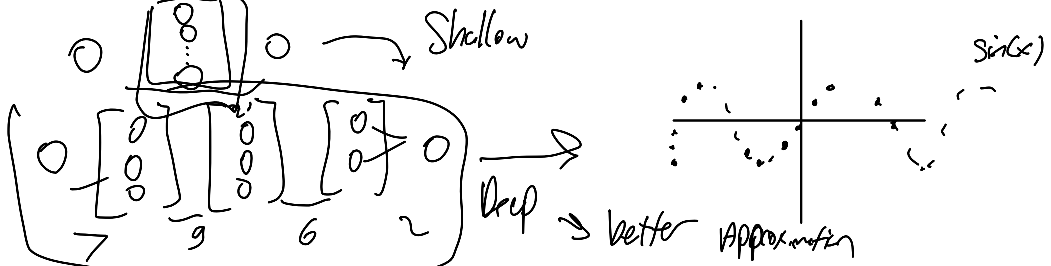
Backward Propagation

Hyperparameters

Universal Approximation Theorem

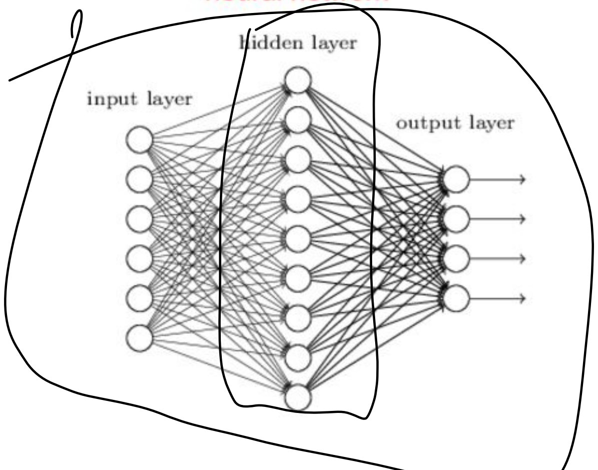
Deep Neural Networks

Deep Neural Networks

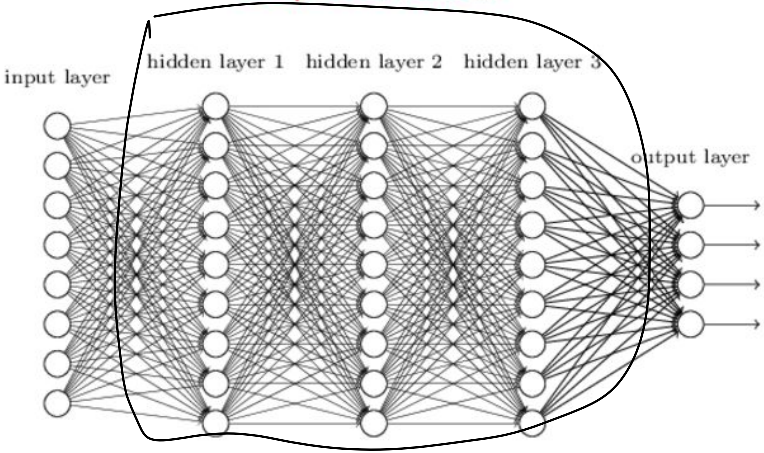


- Shallow neural networks is a term used to describe NN that usually have only one hidden layer while the term deep neural networks is used to describe NN that have several hidden layers.
- The deep NN with the right architectures achieve better results than shallow ones that have the same computational power.

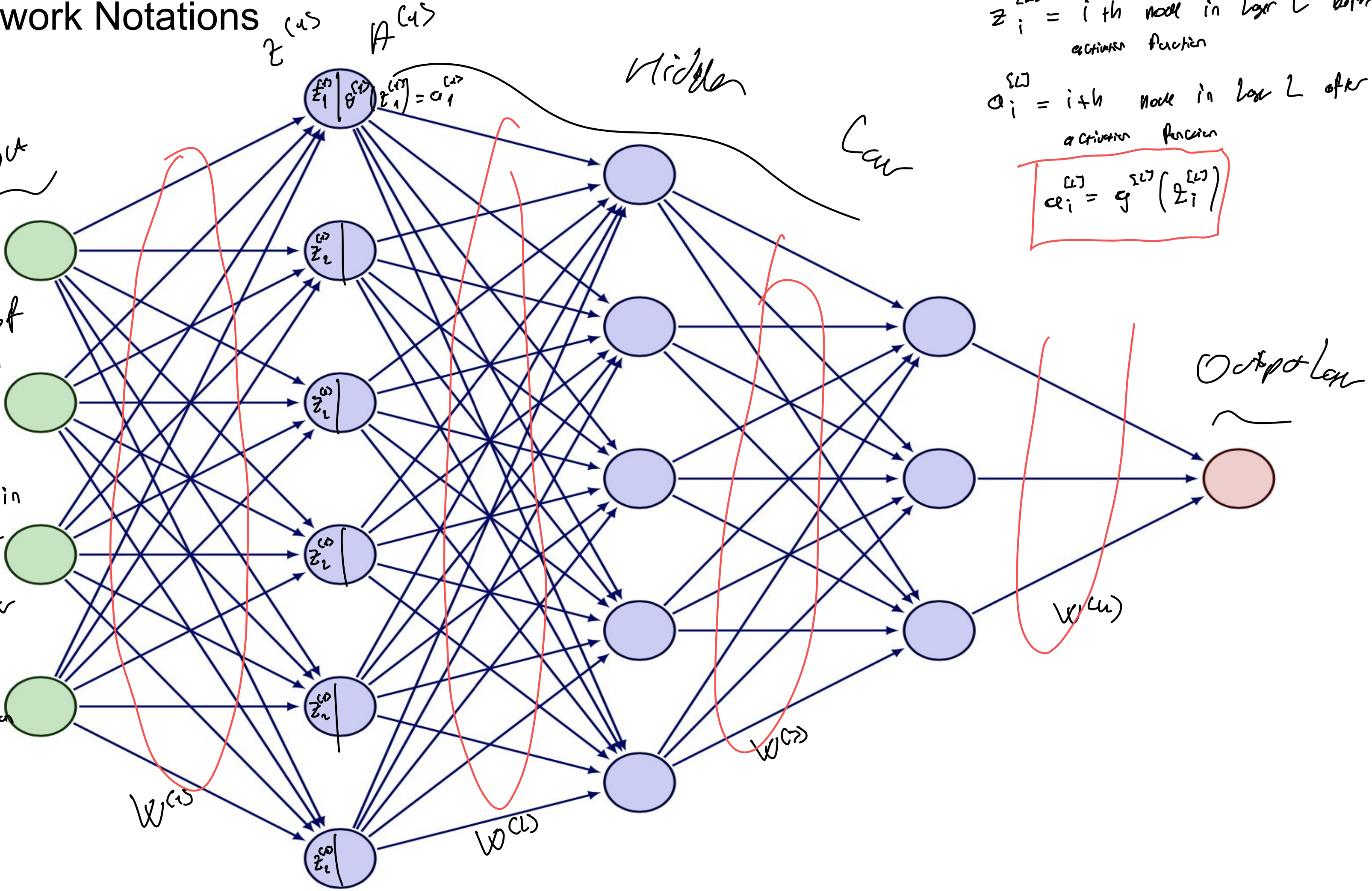
"Non-deep" feedforward neural network



Deep neural network



Neural Network Notations



$n^{[L]}$: number of units in Layer L

$W^{[L]}$ = Weight matrice in Layer L

$b^{[L]}$ = bias vector in Layer L

$g^{[L]}$ = activation function in Layer L

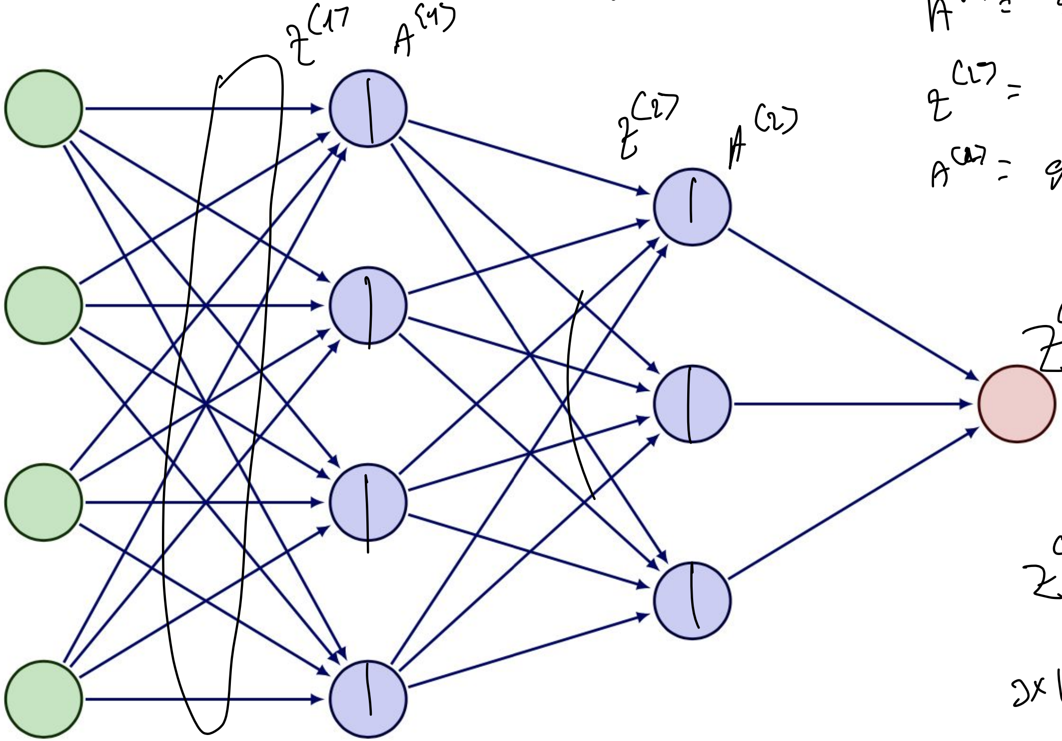
$z_i^{[L]}$ = i th node in Layer L before activation function

$a_i^{[L]}$ = i th node in Layer L after activation function

$\boxed{a_i^{[L]} = g^{[L]}(z_i^{[L]})}$

Forward Propagation

$g^{[L]}$ = activation func on layer L



Assume $A^{[0]}$.shape = (4, 1)
↓
one sample

$Z^{[1]} = W^{[1]} A^{[0]} + b^{[1]}$

$A^{[1]} = g^{[1]}(Z^{[1]})$

$Z^{[2]} = W^{[2]} A^{[1]} + b^{[2]}$

$A^{[2]} = g^{[2]}(A^{[2]})$

$$\boxed{\begin{aligned} &W.\text{shape} = (n^{[l]}, n^{[l-1]}) \\ &Z.\text{shape} = (n^{[l]}, m) \\ &A.\text{shape} = (n^{[l]}, m) \\ &b.\text{shape} = (n^{[l]}, 1) \end{aligned}}$$

$$Z^{[1]} = \underbrace{W^{[1]}}_{4\times 4}\ \underbrace{A^{[0]}}_{4\times 1} + \underbrace{b^{[1]}}_{4\times 1}$$

$4\times 1 = 4\times 1 + 4\times 1$

$$Z^{[2]} = \underbrace{W^{[2]}}_{3\times 4}\ \underbrace{A^{[1]}}_{4\times 1} + \underbrace{b^{[2]}}_{3\times 1}$$

3×1

$3\times 1 + 3\times 1$

$$Z^{[3]} = \underbrace{W^{[3]}}_{1\times 3}\ \underbrace{A^{[2]}}_{3\times m} + \underbrace{b^{[3]}}_{1\times 1}$$

$1\times m + 1\times 1$

$1\times m$

$Z^{[3]} = 1\times m$

Forward Propagation

$$\left[A^{[1]} \right] =$$

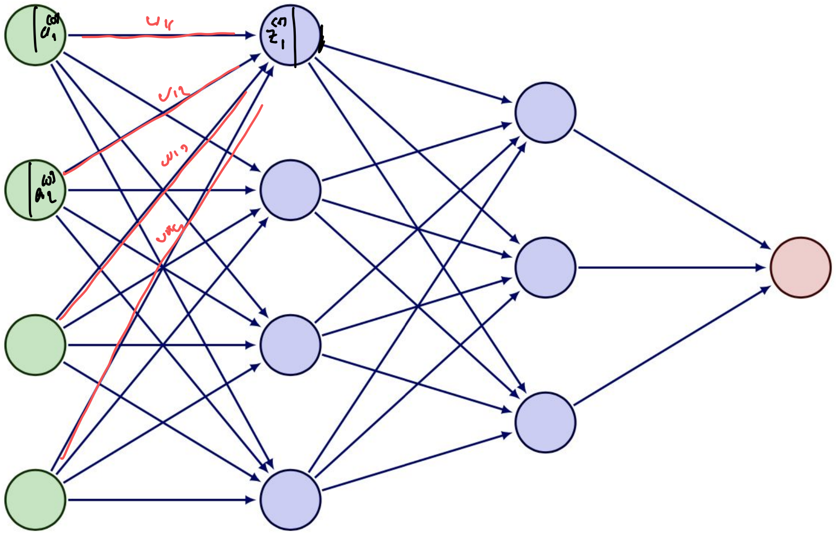
$$a_1^{[1]} = g(z_1^{(1)}) = \left(w_{11} a_1^{(0)} + w_{12} a_2^{(0)} + w_{13} a_3^{(0)} + w_{14} a_4^{(0)} + b_1^{(1)} \right) g^{(1)}$$

$$a_2^{[1]} = g(z_2^{(1)}) = \left(w_{21} a_1^{(0)} + w_{22} a_2^{(0)} + w_{23} a_3^{(0)} + w_{24} a_4^{(0)} + b_2^{(1)} \right) g^{(1)}$$

$$a_3^{(1)} = g(z_3^{(1)}) = \left(w_{31} a_1^{(0)} + w_{32} a_2^{(0)} + w_{33} a_3^{(0)} + w_{34} a_4^{(0)} + b_3^{(1)} \right) g^{(1)}$$

$$a_4^{(1)} = g(z_4^{(1)}) = \left(w_{41} a_1^{(0)} + w_{42} a_2^{(0)} + w_{43} a_3^{(0)} + w_{44} a_4^{(0)} + b_4^{(1)} \right) g^{(1)}$$

$A^{(1)}$



$$w_{11} a_1^{(0)} + w_{12} a_2^{(0)} + w_{13} a_3^{(0)} + w_{14} a_4^{(0)} + b_1^{(1)} = z_1^{(1)}$$

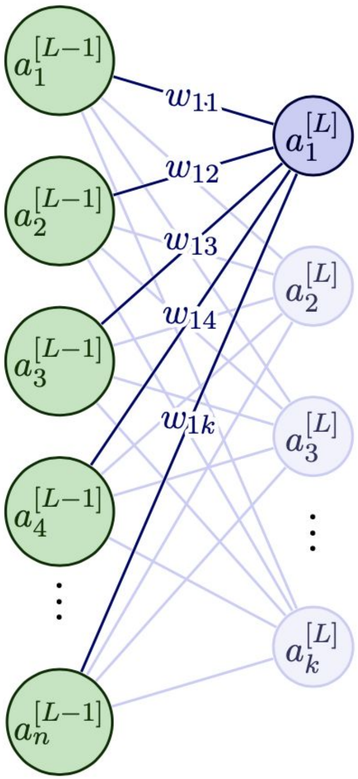
$$w_{21} a_1^{(0)} + w_{22} a_2^{(0)} + w_{23} a_3^{(0)} + w_{24} a_4^{(0)} + b_2^{(1)} = z_2^{(1)}$$

$$w_{31} a_1^{(0)} + w_{32} a_2^{(0)} + w_{33} a_3^{(0)} + w_{34} a_4^{(0)} + b_3^{(1)} = z_3^{(1)}$$

$$w_{41} a_1^{(0)} + w_{42} a_2^{(0)} + w_{43} a_3^{(0)} + w_{44} a_4^{(0)} + b_4^{(1)} = z_4^{(1)}$$

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \\ w_{41} & w_{42} & w_{43} & w_{44} \end{bmatrix} \begin{bmatrix} a \end{bmatrix} + b^{[1]} = \begin{bmatrix} z_1^{[1]} \\ z_2^{(1)} \\ z_3^{(1)} \\ z_4^{(1)} \end{bmatrix}$$

Forward Propagation



$$a_1^{[L]} = g(z_1^{[L]}) = (w_{11}^{[L]} a_1^{[L-1]} + w_{12}^{[L]} a_2^{[L-1]} + \dots + w_{1n}^{[L]} a_n^{[L]})$$

$$a_2^{[L]} = g(z_2^{[L]}) = g(w_{21}^{[L]} a_1^{[L-1]} + w_{22}^{[L]} a_2^{[L-1]} + \dots + w_{2n}^{[L]} a_n^{[L]})$$

$$= g^{[L]}\left(\sum_{i=1}^{n} w_{2i}^{[L]} a_i^{[L-1]}\right)$$

$$A^{[L]} = W^{[L]} A^{[L-1]} + b^{[L]}$$

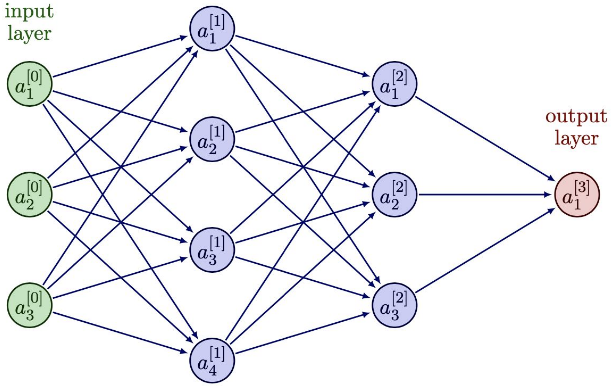
$$k\times m = k\times n \;\; n\times m \;\; + k\times 1$$

$$a_k^{[L]} = g(z_k^{[L]}) = g(w_{k1}^{[L]} a_1^{[L-1]} + w_{k2} a_2^{[L-1]} + \dots w_{kn} a_n^{[L-1]})$$

$$= g^{[L]}\left(\sum_{i=1}^{n} w_{ki}^{[L]} a_i^{[L-1]}\right)$$

Backward Propagation

Backward Propagation



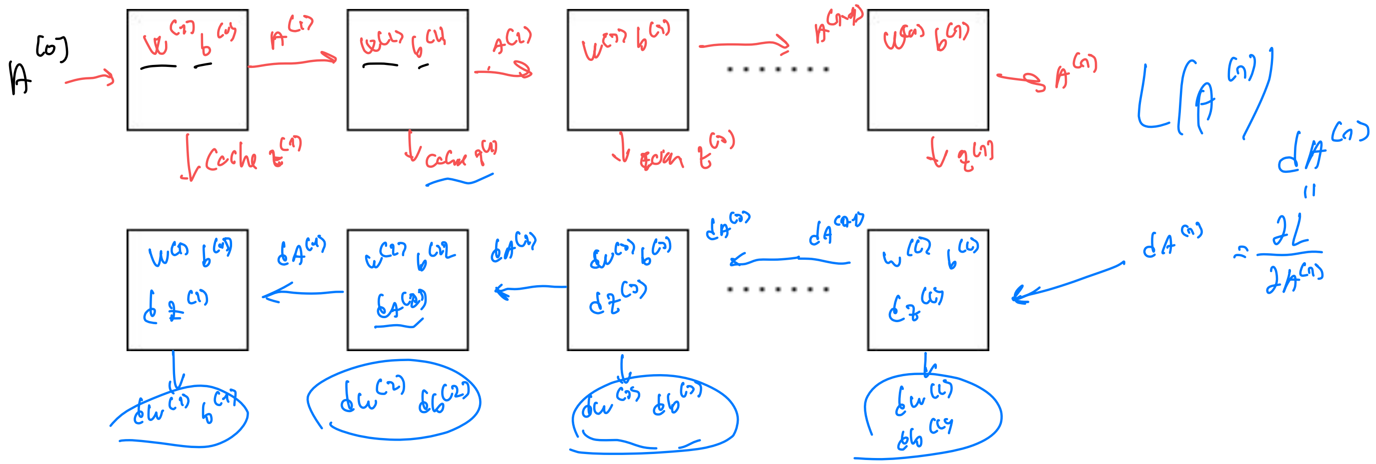
Forwar Prop Input : $A^{[l-1]}$

Output : $A^{[l]}$

Backwar Prop Input : $dA^{[l]}$

cache ($z^{[l]}$)

Output : $dW^{[l]}$ $db^{[l]}$ $dA^{[l-1]}$



Backward Propagation

$$L(a^{(3)}) = -\left(y \log(a^{(3)}) + (1-y)\log(1-a^{(3)})\right)$$

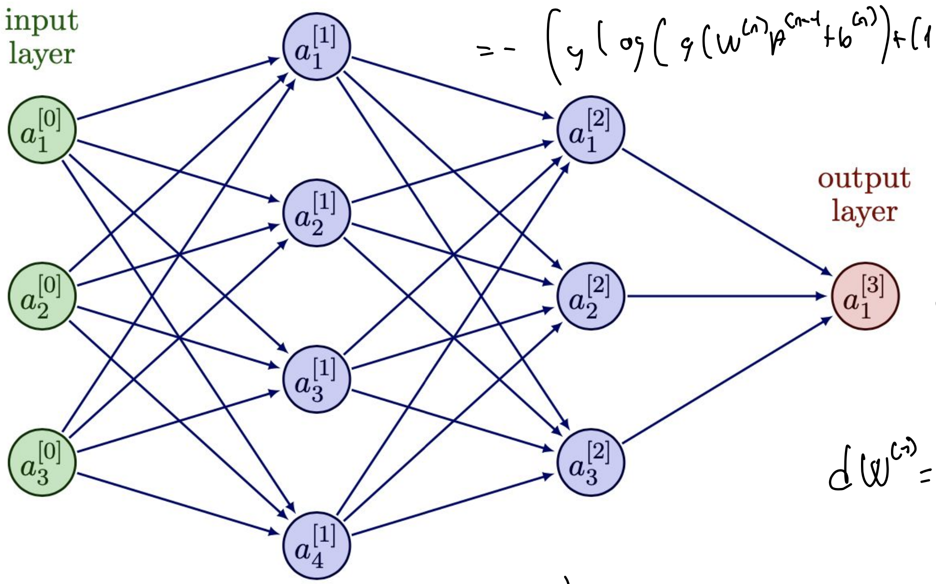
$$= -\left(y \log(g(z^{(3)})) + (1-y)\log(1-g(z^{(3)}))\right)$$

$$= -\left(y \log(g(W^{(3)}A^{(n-1)}+b^{(3)})) + (1-y)\log(1-g(W^{(3)}A^{(n-1)}+b^{(3)}))\right)$$

$$-\left(\frac{y}{a} + \frac{(1-y)}{1-a}\right) = \qquad \frac{d\sigma(x)}{dx} = \sigma'(x)$$

$$\frac{\partial(\sigma(z^{(3)}))}{\partial(z^{(3)})} = \sigma(z^{(3)})(1-\sigma(z^{(3)})) = \sigma'(z^{(3)})$$

$$\partial(g(z^{(3)}))$$

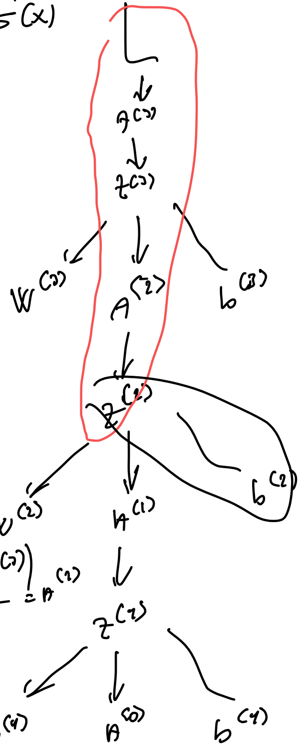


$$dz = \frac{\partial L}{\partial z}$$

$$\partial z^{(3)} = \frac{\partial L}{\partial z^{(3)}} = \underbrace{\frac{\partial L}{\partial A^{(3)}}}_{dA^{(3)}} \cdot \underbrace{\frac{\partial A^{(3)}}{\partial z^{(3)}}}_{g'(z^{(3)})}$$

$$\frac{\partial(W^{(3)}A^{(2)}+b^{(3)})}{\partial W^{(3)}} = A^{(2)}$$

$$dW^{(3)} = \frac{\partial L}{\partial W^{(3)}} = \underbrace{\frac{\partial L}{\partial A^{(3)}} \cdot \frac{\partial A^{(3)}}{\partial z^{(3)}}}_{dz^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial W^{(3)}}$$



1

$$db^{(2)} = \frac{\partial L}{\partial b^{(2)}} = \underbrace{\underbrace{\underbrace{\frac{\partial L}{\partial A^{(3)}} \cdot \frac{\partial A^{(3)}}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial A^{(2)}}}_{da^{(2)}} \cdot \underbrace{\frac{\partial A^{(2)}}{\partial z^{(2)}}}_{g^{(2)'}(z^{(2)})}}_{dz^{(2)}}} \cdot \frac{\partial z^{(2)}}{\partial b^{(2)}}$$

$$z^{(2)} = W^{(2)}A^{(1)} + b^{(2)}$$

$f(x) = x$

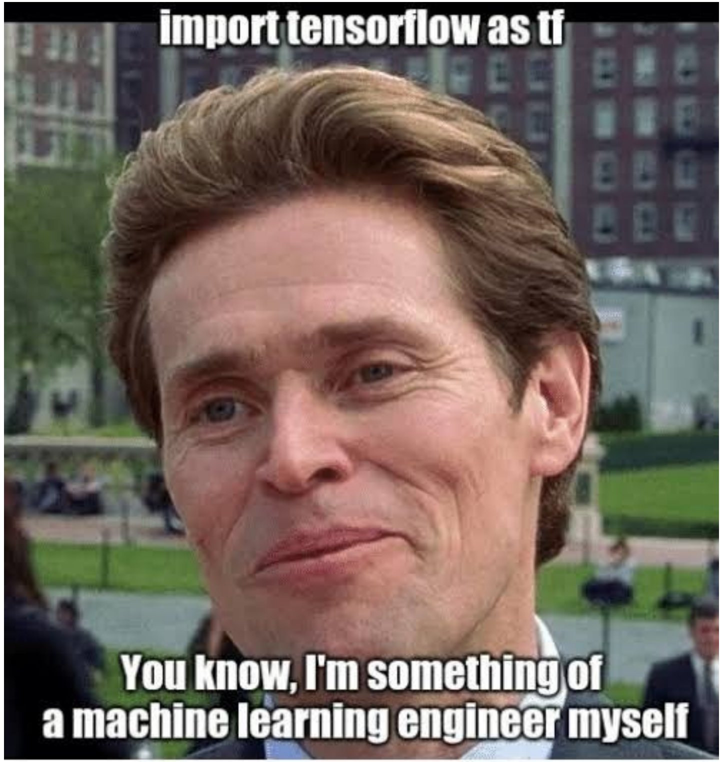
$f'(x) = 1$

$$dz^{[L]} = da^{[L]} * g^{[L]'}(z^{[L]})$$

$$dW^{[L]} = dz^{[L]} \cdot a^{[L-1]}$$

$$db^{[L]} = dz^{[L]}$$

Hyperparameters



Hyperparameters

data 500

- batch = 500
- batch = 250
- batch = 100

- 1 iteration : 1 epoch
- 2 iteration : 1 epoch
- 5 iteration : 1 epoch

Hyperparameters effect parameters

Hyperparameter examples:

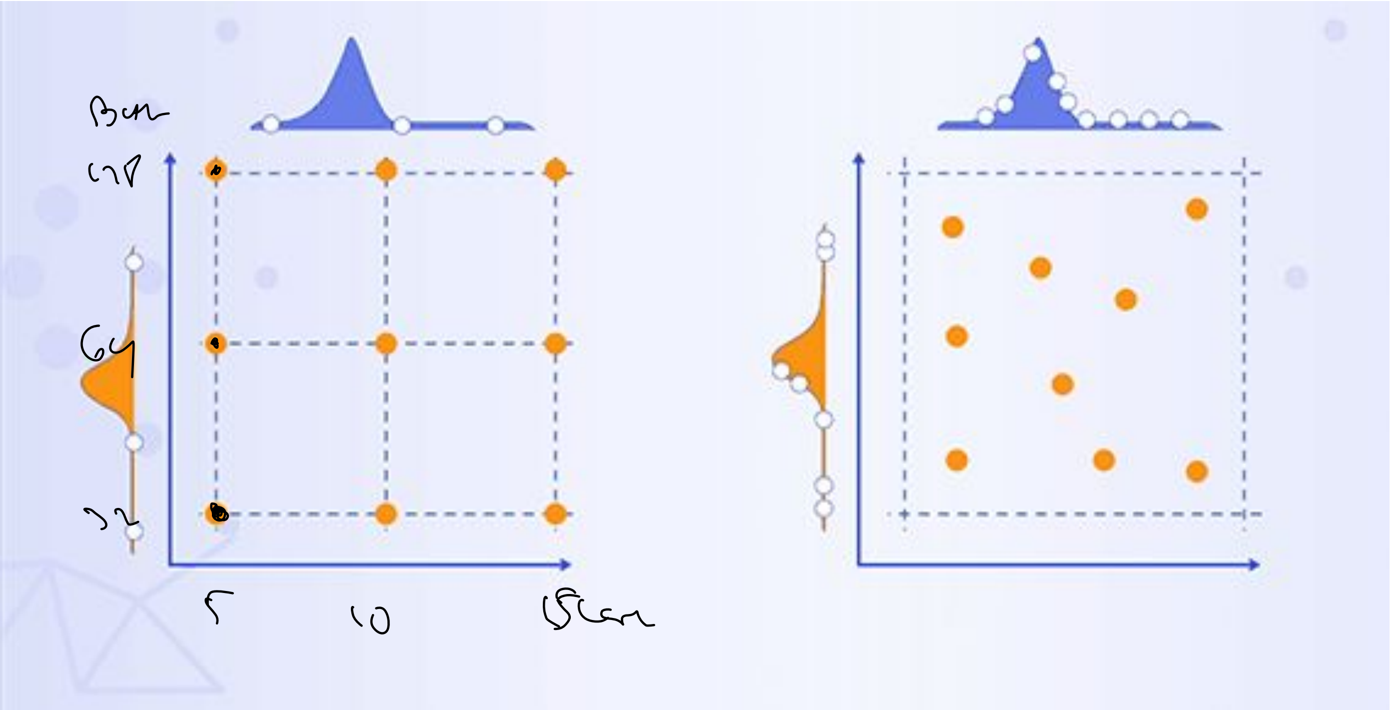
- Learning Rate
- #Units
- #Iterations
- #Layers
- Batch size

epoch

We can select hyperparameters using several methods

Hyperparameter Tuning

Grid Search | Random Search



Universal Approximation Theorem

Some conditions $\Rightarrow$ Borel measurable: $\mathbb{R}^n$

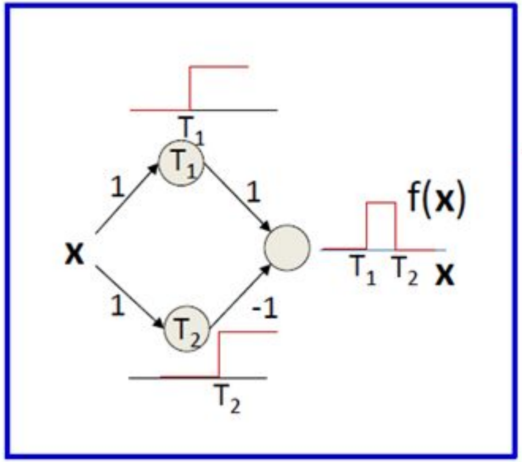
$\downarrow$ subset $\rightarrow$ closed

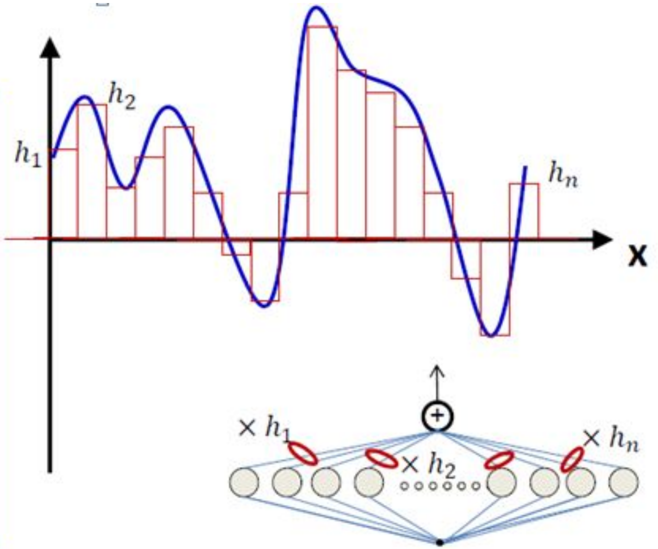
$\swarrow$ continous $\searrow$ bounded

The Universal Approximation Theorem means that regardless of what function we are trying to learn, we know that a large MLP will be able to represent this function

$f: NN \longrightarrow \mathbb{R}^n \Rightarrow f$ is onto

NN: set of neural nets





inzva: *brings the AI fellows together*

inzva: