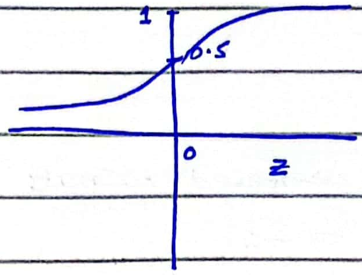
Lecture : 09

Key Activation function

→ An activation function in a neural network decides whether a neuron should be activated or not, including non-linearity and enabling the model to learn complex patterns from the data.

→ Sigmoid

$$\sigma(z) = \frac{1}{1+e^{-z}}$$



→ sigmoid is use for binary classification

→ maps between 0 to 1

→ usually used as an output layer neuron

- at $z=0$ sigmoid 0.5
- as z moves toward $-\infty$ sigmoid value get closer to 0
- as z moves toward $+\infty$ sigmoid value closer to 1

Pros :

⇒ Reduce abrupt changes in prediction

⇒ Directly interpretable as the probability of a class

Cons

⇒ computationaly expensive because of exponentiation

⇒ Non-zero centered output (leads to convergence problem)

⇒ vanishing gradient (function shows straight line at particular values of z so rate of change becomes zero, it cause problem in weights optimization)

→ Tanh (Scaled version of sigmoid)

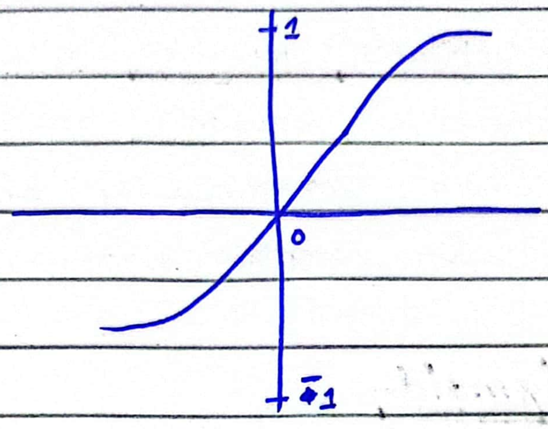
⇒ use for Binary classification

⇒ It range is (-1 to +1) making it zero centered

⇒ useful for output layer

$$\tanh(z) = \frac{e^{2z} - 1}{e^{2z} + 1}$$

$$= 2\sigma(2z) - 1$$



Pros:

⇒ zero centered outputs

⇒ (-1 to +1)

⇒ use with SVM LOSS

Cons:

⇒ computationaly expensive

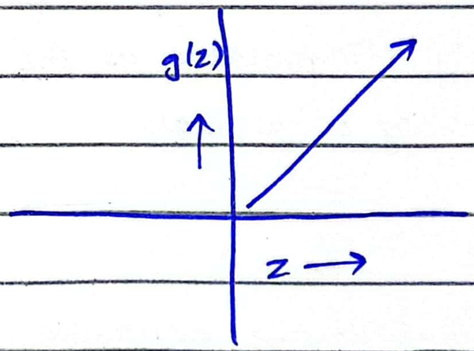
⇒ vanishing gradient

→ RELU (Rectified Linear unit)

⇒ It's the most important neural net for contemporary neural networks

$$g(z) = \max(0, z)$$

$$\frac{d}{dz} g(z) = \begin{cases} z & z > 0 \\ 0 & z \leq 0 \end{cases}$$



⇒ use in hidden layers to avoid vanishing gradient problem

⇒ for all positive values, it returns the input value z

⇒ for all negetive values it returns zero

Pros

⇒ No vanishing gradient

⇒ Computationally effective

⇒ commonly used in hidden layers

⇒ deactivate neuron on negative numbers

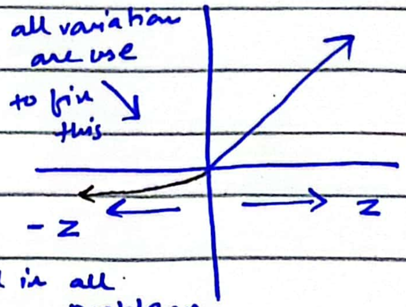
Cons

⇒ Dying Relu problem (neuron can die if the input is negative, leading to inactive neurons that no longer update during training)

→ Leaky Relu

★ All Relu variations are use to address dying relu problem

$$g(z) = \begin{cases} z & z > 0 \\ \alpha z & z \le 0 \end{cases}$$



where $\alpha = 0.01$

con: fixed α value doesn't perform well in all problems

→ Parametric Relu

$$g(z) = \begin{cases} z & z > 0 \\ \alpha z & z \le 0 \end{cases}$$

here α is a parameter (learn from data)

→ Exponential Relu

$$g(z) = \begin{cases} z & z > 0 \\ \alpha(e^{z} - 1) & z \le 0 \end{cases}$$

Pros: zero centered output

Cons: computationaly expensive