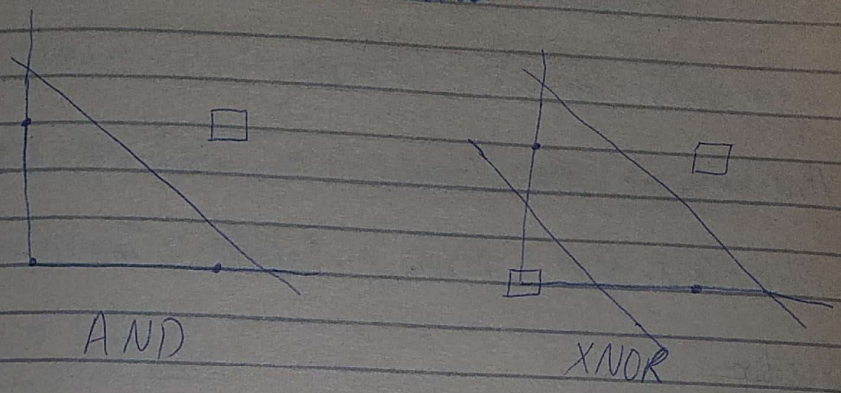
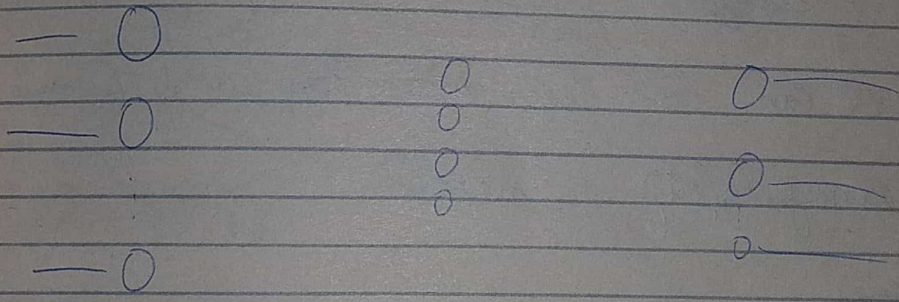
18-10-2021 Monday

week #8 Lecture #11

Neural Networks



Each logistic unit = Neuron



2 layered Neural Network.

we don't count Input layer in no.'s.

~~Deep learning.~~

If more than 1 hidden layer, we say it is deep learning.

2^{Power} = Neurons in each layer usually.

Question:-

If we increase No. of Neurons in one hidden layer, what does it Change?

Ans: we can now learn more complicated Polynomials. More accuracy. Capacity is Increased.

Question

If we increase No. of layers in NN, what does it change?

Ans) we can now learn more complicates Polynomials More accuracy, Capacity is Increased.

Problem:

If we have only 2 classes & many layers & Neurons, then its Problem is that it overfits & memorizes the data, tries to fit every example Individually, also covers noise in the data.

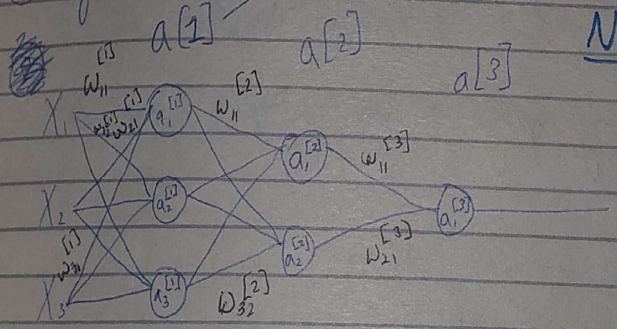
Solution

W

If no. of classes, is ~~small~~ less, then don't make deep learning NN or make more Neurons & layers.
or if Dataset is small, then also don't make deep learning NN or make more Neurons & layers.

Techniques to Solve this:

(1) Regularization → layer



Naming Convention:-

Fully Connected NN

↗ Where all inputs are connected to input layer Neurons

$$Z_1^{[1]} = W_{11}^{[1]} X_1 + W_{21}^{[1]} X_2 + W_{31}^{[1]} X_3$$

$$Z_2^{[1]} = W_{12}^{[1]} X_1 + W_{22}^{[1]} X_2 + W_{32}^{[1]} X_3$$

$$Z_3^{[1]} = W_{13}^{[1]} X_1 + W_{23}^{[1]} X_2 + W_{33}^{[1]} X_3$$

Every Neuron calculates weighted Sums, Then it applies Activation function.

$$a_1^{[1]} = \sigma(z_1^{[1]})$$

$$a_2^{[1]} = \sigma(z_2^{[1]})$$

$$a_3^{[1]} = \sigma(z_3^{[1]})$$

Question:

If we don't apply this activation function in hidden layers & only apply it on output layer, what does it imply?

Solution:- our learning slows down.

$$f = W_3(W_2(W_1(x)))$$

$$W = W_1 * W_2 * W_3$$

$$f = Wx \quad \text{// It becomes linear classifier}$$

Activation function acts as a linear classifier if it adds non-linearity.

Question:

If we use Sigmoid as an Activation function, what does it Imply?

Solution:

Local Gradient

$$\frac{dL}{dw_{11}^{[1]}} = \sigma(Z_1^{[1]})\left(1-\sigma(Z_1^{[1]})\right) * X_1 * \underbrace{\text{upstream gradients}}$$

if we have $Z_1^{[1]} = +\infty$, then

$$\sigma(Z_1^{[1]}) = 1$$

$$\frac{dL}{dw_{11}^{[1]}} = 1(1-1) = 0$$

So then our weights won't be updated ever.

$$W_{11}^{[1]} = W_{11}^{[1]} - \alpha \underbrace{dW_{11}^{[1]}}_{0}$$

So weights won't be update

Sigmoid Problem:

It kills gradients of weights.

If weighted sum value is $-4 \rightarrow$ Grad or Grad $> +4$

Vanishing Gradients Problem:

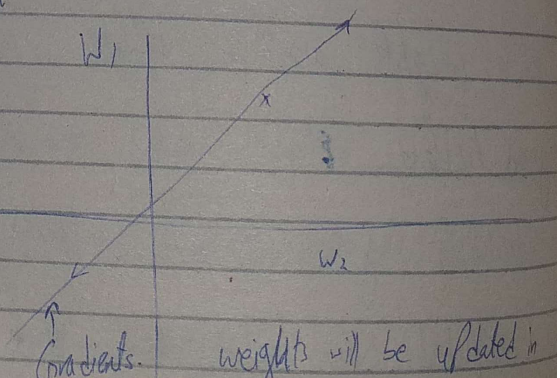
If we have more layers meaning a bigger NN, then weights multiplied together becomes really small.

② Problem of Sigmoid: ~~Exploding Gradient Problem~~

Weights are positive, & Gradients are positive, Sigmoid output is also positive so our weights are updated in either positive or Negative so our learning is not improved & gets affected

It can get Negative bcz we back propagate from upstream gradients.

It Saturates the output.



weights will be updated in negative direction

$W_1 = W_1 - \alpha \frac{dW}{+}$

W_1 decreases.

we only explore weights in one direction.