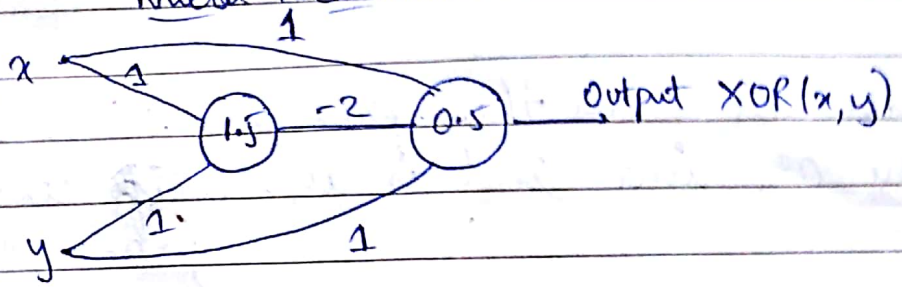
Theory - Ass - 3

Shubham Khanna

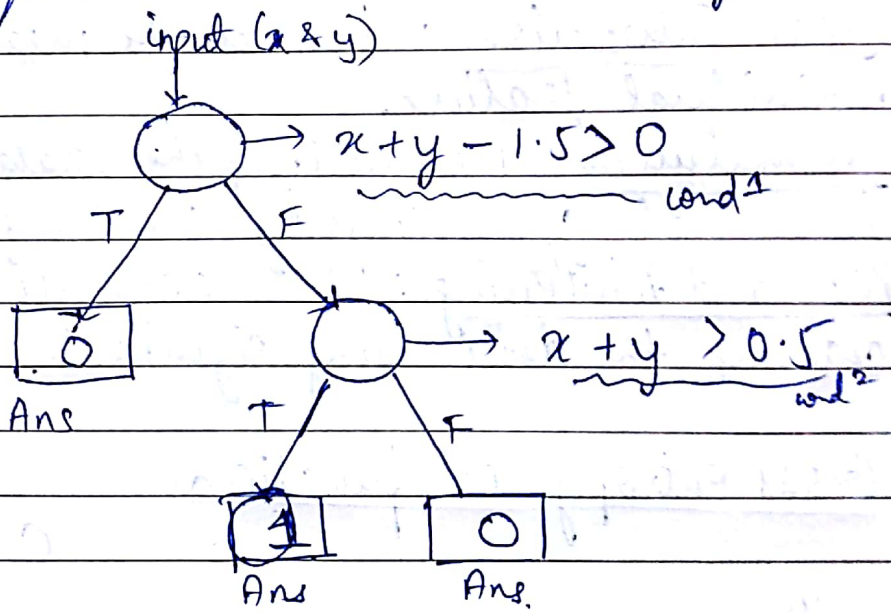
1 YES

Nueral Network



Eq of network $x+y-2(x+y-1.5>0)-0.5>0$

The above equation can be also be made using a decision tree.



Both the models represent the same ~~model~~ mathematical equation.

2 Sigmoid saturate and kill gradients → The problem is if the ~~sigmoid~~ nueron's activation saturates at either tail of 0 or 1, the gradient at the regions is almost zero. Thus during backpropogation, the small gradient will kill itself.

Sigmoid outputs are not zero-centered. This creates problems on the dynamics during gradient decent.

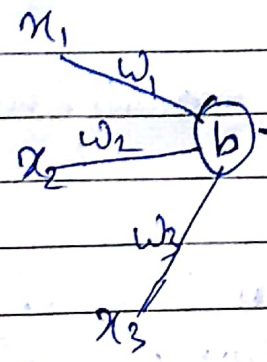
ReLU does not have these problems since it doesn't kill the gradient and is also zero-centred.

We can use some Pre-processing techniques to counter the problem.

- Mean Subtraction : Subtract the mean across every individual feature.
- Normalization : Normalising the data dimensions so they scale approximately the same way.
- PCA and Whitening : This normalises the data, after centering the data using eigenbasis.

3. Cross entropy loss function.

$$C = \frac{-1}{n} \sum_x y \ln a + (1-y) \ln (1-a)$$



- n → no. of items of training data
- x → input
- y → o/p, a → o/p from of Neuron

$$\frac{\partial C}{\partial w_j} = \frac{-1}{n} \sum_x \left(\frac{y}{\sigma(z)} - \frac{(1-y)}{1-\sigma(z)} \right) \frac{\partial \sigma(z)}{\partial w_j} =$$

$$= \frac{-1}{n} \sum_x \left(\frac{y}{\sigma(z)} - \frac{(1-y)}{1-\sigma(z)} \right) \sigma'(x)\, x_j$$

$$\frac{\partial C}{\partial w_j} = \frac{1}{n} \sum_x \frac{\sigma'(z)\, x_j}{\sigma(z)(1-\sigma(x))} (\sigma(z) - y) = \frac{1}{n} \sum_x (\sigma(z) - y)\, x_i$$

Thus we observe that the rate of learning is dependent on $(\sigma(z) - y)$ ie the error of the output. It avoids the learning slowdown caused by $\sigma'(z)$ in the quadratic cost function.