# CS564: Foundations to Machine Learning

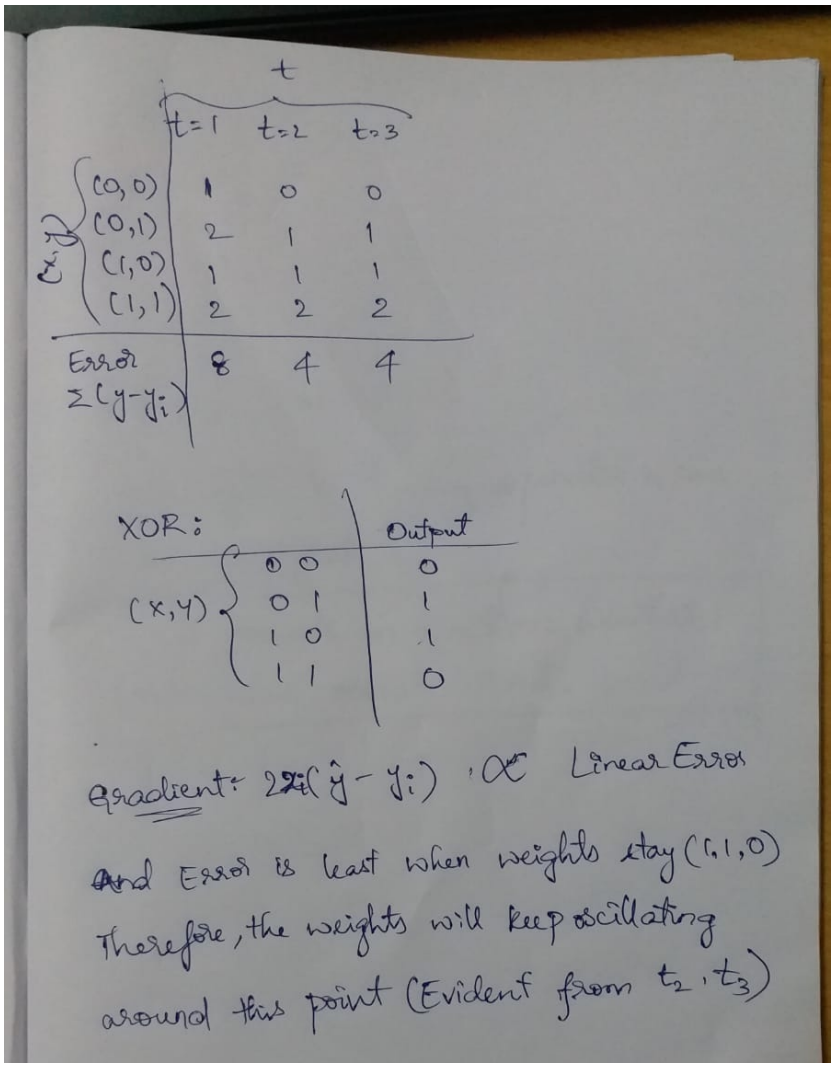
End Semester Quiz

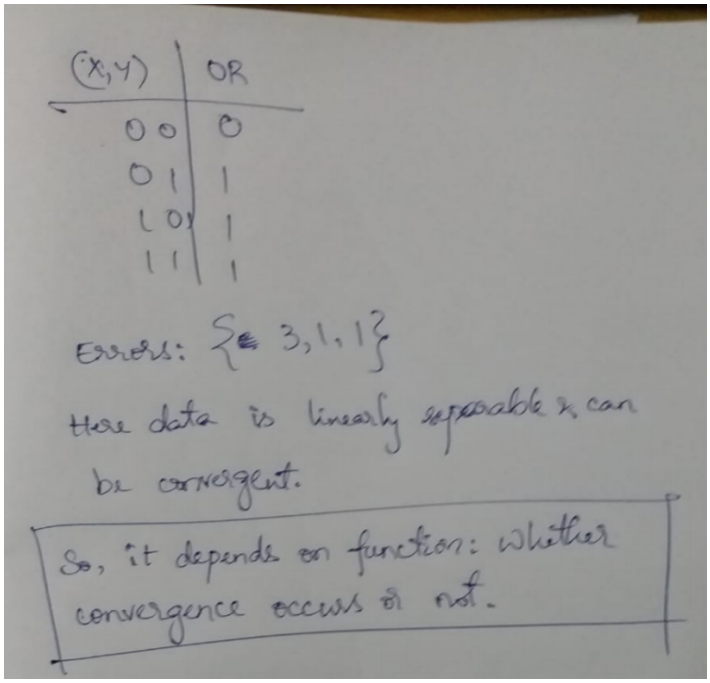
| Name: M. Maheeth Reddy | Roll No. 1801CS31 | Date: 25-Nov-2021 |
|---|---|---|

---

## Ans 1:

**For any perceptron, the weights for the consecutive three iterations are observed as follows: <0,1,1>, <1,1,0>, <1,1,0>. Will this perceptron converge? Justify your answer by proper intuitions.**



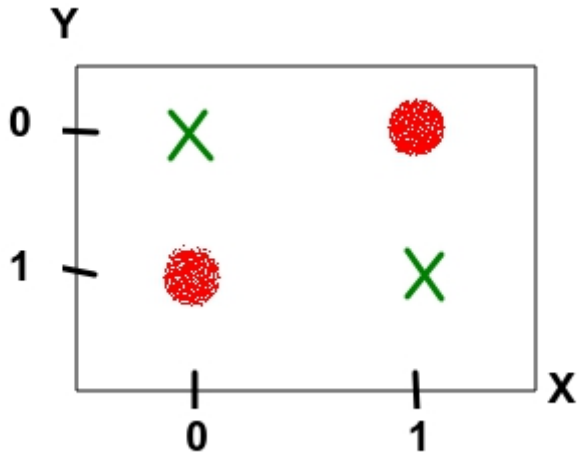



**Can a linear neuron compute X-OR? Explain your answer with appropriate intuitions.**

No, a linear neuron cannot compute XOR logic.

Definition of XOR: XOR(x,y) = {0 if x=y, 1 if x!=y}

Plot of data points in co-ordinate plane will look like this:




Clearly, we cannot seperate the data points linearly. Due to this reason, linear neuron cannot compute XOR. **Hence** XOR cannot be computed using single linear neuron.

## Ans 2:

**Does ensemble always perform better than the single classifier?**

**YES**, ensemble always performs better than single classifier, provided the following:

1. Insufficient training data
2. Learning algorithm leads to different local optimals easily

3. Potentially valuable information may be lost by discarding the results of less-successful classifiers E.g., the discarded classifiers may correctly classify some samples

**Reason**:

- A single classifier leads to a model which is not very accurate, whereas ensemble learning employs multiple learners and combines their predictions.
- Ensemble gives the aggregated output from various classifiers which includes the results of many other features and parameters.
- This leads to an improvement in predictive accuracy.

**Among Bagging and Boosting, which one performs better than the other and under what conditions?**

**Boosting**: Learns sequentially and adaptively to improve model predictions of a learning algorithm.

**Bagging**: Learns from each other independently in parallel and combines them for determining the model average.

Bagging is only effective only when we are using unstable nonlinear models.It decreases the variance and helps to avoid overfitting. It is usually applied to decision tree methods.

Boosting is an ensemble modeling technique that attempts to build a strong classifier from the number of weak classifiers.

Boosting is used to learn complementary classifiers so that instance classification is realized by taking the weighted sum of the classifiers

- Bagging always uses re-sampling rather than re-weighting
- Bagging does not modify the distribution over examples or mislabels, but instead always uses the uniform distribution
- In forming the final hypothesis, bagging gives equal weight to each of the weak hypotheses

Generally, bagging is better than boosting.

## Ans 3:

**Explain with examples how vanishing gradient or exploding gradient causes problems in modeling a vanilla RNN.**

However there are some problems with respect to vanilla RNN approaches. They are:

1. RNNs are not very good at capturing long-term dependencies (Vanishing Gradient Problem): When weights are initialised too small, the resultant gradients shrink exponentially and there cant be a proper gradient descent, hence no learning.

2. RNNs could potentially result in highly variant outputs (Exploding Gradient Problem): When weights are initialised too large, the outputs from each step will

vary highly and gradients grow exponentially, resulting in fluctuating weights and instability.

We typically use variations of Vanilla RNNs like GRU, LSTM to deal with these problems.

**How can the parallel training be implemented in RNN?**

**Parallel Training in RNNs**

The property of RNN limits it to be sequent in nature. Therefore, the model has to wait for the past words to be processed before the processing of the current word. However, there is potential to optimise this approach.

This could be done via parallel training using data paralleling where forward propagation produces outputs for multiple inputs at a time, and the backward-propagation uses all these outputs to propagate gradients from time-step T to 1. This is called mini-batch training of neural networks.

**Explain the following statement with appropriate intuition: HMM and RNN both are efficient sequence learning algorithms, but the expressibility of RNN is higher than HMM.**

**Expressibility of RNN is higher than HMM:**

Although HMMs are good models for sequential learning tasks, they suffer with the major limitation that it can remember only log(N) amount of bits about the past, given that the number of hidden states are N. This issue is taken care of in RNNs.
Also, HMMs are inherently much simpler than RNNs in design and rely on strong assumptions which may not always be true. An RNN may perform better if we have a very large dataset, since the extra complexity can take better advantage of the information in our data. The major loophole is that the state transitions in HMM depend only on the current state and not on the previous states. This assumption is not always true. In many real world examples the Markov assumption cannot hold true. In such cases RNNs perform better than HMMs are hence preferred.

**Give insightful explanation to the following statement: even though none of LSTM and GRU is clearly the winner, but they have their own appealing characteristics?**

**LSTM and GRU appealing characteristics:**

The architectural differences in the construction of LSTM and GRUs, make them suited for different applications. Both LSTM and GRUs were proposed with the aim of improving the problems seen in vanilla RNNs. The key architectural differences between LSTM and GRUs are:

- The GRU has two gates, LSTM has three gates
- GRU does not possess any internal memory, they don't have an output gate that is present in LSTM

- In LSTM the input gate and target gate are coupled by an update gate and in GRU reset gate is applied directly to the previous hidden state. In LSTM the responsibility of reset gate is taken by the two gates i.e., input and target.

We can see experimentally that GRU takes lesser time to train than LSTM, whereas LSTMs are more accurate in general. We can go with LSTM when we are dealing with large data and accuracy is important.