BACKPROPAGATION EXPERIMENT REPORT

GANESHREDDY ANNAPAREDDY

50442295

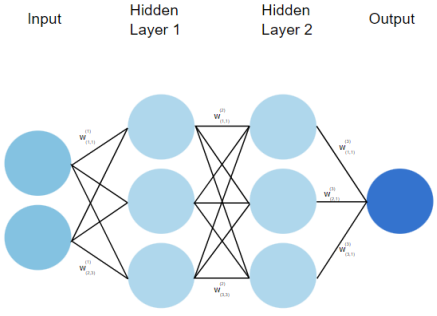
ganeshre@buffalo.edu

“ When you want to increase the accuracy of your neural network model, backpropagation works well to assist you do so. You may improve the accuracy of your neural network model predictions by using the backpropagation technique. The backpropagation algorithm will be thoroughly explained in this experiment report, along with how I used Python to implement it. ”

1. BACKPROPAGATION

For artificial neural networks, the backpropagation method is a form of supervised learning process where we may adjust the weight functions and raise the model's accuracy. The cost function is decreased using the gradient descent approach. The mean-squared difference between the expected and actual data is decreased. When training feed-forward neural networks with a set of data whose classifications are known to us, this kind of procedure is typically utilized. Backward propagation is also known as the spreading of mistakes backward to increase precision. The backpropagation procedure must be used whenever the forecast we obtained from a neural network model differs significantly from the actual output in order for us to gain improved accuracy.

WORKING OF BACKPROPAGATION :



Let's now gain an understanding of how the algorithm truly operates. There are three main layers for the backpropagation. They are input layer, hidden layer which are of two layers and an output layer as we can see from the figure on the right. The process can be explained in form of the steps

I. At first the input layer receives the input.

II. Overweight's are then averaged over the input.

III. Every concealed layer processes the result. Here, each output is referred to as a "Error," which really refers to the discrepancy between the output that is produced and the output that is sought.

IV. In order to improve the weights and lower the error, the algorithm returns to the hidden layers in this

TYPES *of* BACKPROPAGATION :

The two primary categories of backpropagation techniques are recurrent backpropagation and static backpropagation. Let's examine what the two sorts of language truly imply. Static inputs lead to static outputs in static backpropagation. For static categorization issues like Optical Character Recognition, this is especially employed. Recurrent propagation, on the other hand, continues until it hits a certain value or threshold value. The fault propagates backwards after it reaches the fixed value.

EXPERIMENT :

Dataset :

For the experiment backpropagation I have chosen a pretty standard not linearly separable dataset made of two classes 1's and 0's. In this model we will go through the back propagation which is the main aspect. At first I have imported the libraries that are needed for the experiment. They are math, numpy, matplotlib and expit as sigmoid. Second I set the seed and generated a random dataset with 0's and 1's and then visualized it which looks like the figure 1

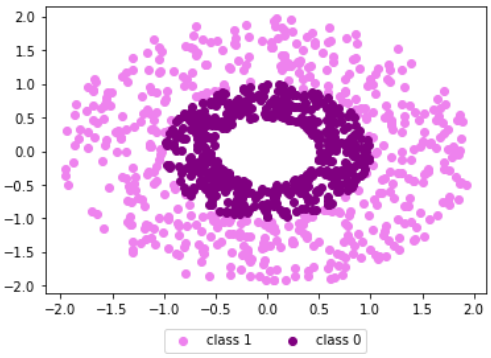
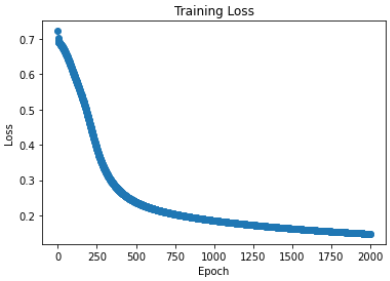
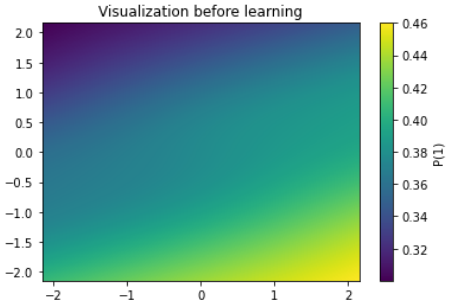


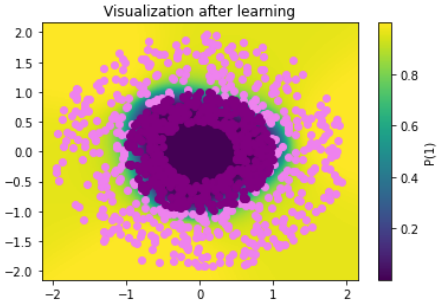
Fig 1



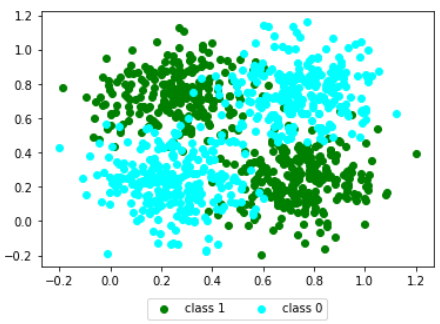
I am splitting the dataset for training and testing purpose and then I am initializing the weights. After that I am doing both the forward propagation and backward propagation after setting the weights, by doing this I am getting the training loss which is crucial to know.

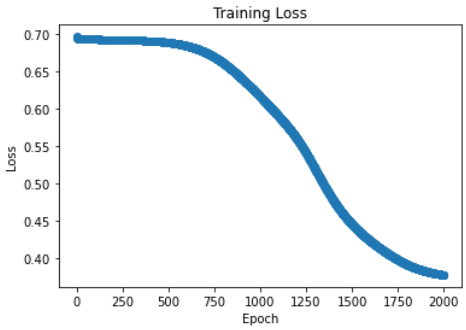
I have visualized the data before and after the training (learning) which looks like below 2 figures. And we can see a clear difference within them.

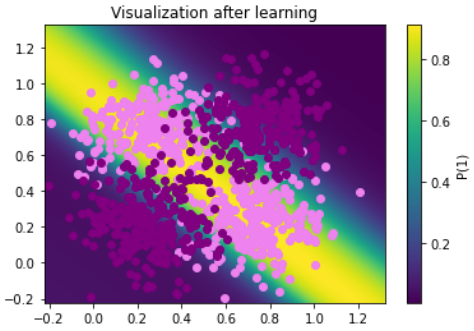




I have implemented the same experiment with different attributes where I got the following below figures which has a different random dataset generated as well as the training loss and the visualization after training.







ADVANTAGES *of* BACKPROPAGATION :

- The implementation of this method is quite easy and quick.
- It is extensively used in the field of voice and face recognition.
- It is a versatile approach because it doesn't require any previous experience with neural networks.

DISADVANTAGES *of* BACKPROPAGATION :

- For erratic and noisy data, the technique is not detrimental.
- The input has a significant impact on how well the backpropagation performs.

CONCLUSION :

In conclusion, we discovered that using backpropagation to enhance feed-forward neural network model accuracy is an excellent idea. Although it is a simple and adaptable algorithm, it struggles with noisy data. It is a fantastic technique to lower the error and increase the model's precision. Gradient descent is used to minimize the loss function in order to optimize the weights by working backwards.