Hearen's Light is Our Guide

Computer Science & Engineering
Rajshahi University of Engineering & Technology

Course No.: CSE 4204
Course Title: Sessional based on CSE 4203

Experiment No. 3

Name of the Experiment: Design and implementation of Multi-layer Neural Networks algorithm (i.e., Back-propagation learning neural networks algorithm).

Course Outcomes: CO1

Learning Domain with Level: Cognitive (Applying, Analyzing, Evaluating & Creating)

Contents

1 Dataset

I've Selected Diabetes Dataset from Kaggle [1] .This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective is to predict based on diagnostic measurements whether a patient has diabetes.

1.1 Dataset Analysis

The Dataset have 8 attributes and 768 instances. The Features are :

1. **Pregnancies:** Number of times pregnant
2. **Glucose:** Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3. **BloodPressure:** Diastolic blood pressure (mm Hg)
4. **Insulin:** 2-Hour serum insulin (mu U/ml)
5. **BMI:** Body mass index (weight in $kg/(heightinm)^2$)
6. **DiabetesPedigreeFunction:** Diabetes pedigree function
7. **Age:** Age (years)
8. **Outcome:** Class variable (0 or 1)

Class Distribution: class value 1 is interpreted as "tested positive for diabetes". In Class 1 there were 268 instances and in class 0 , there were 500 instances.

1.2 Training-Test Ratio

80% of data were used for training and 20% of data for validating .

2 Preprocessing

Multi layer Neural Networks (perceptron) learning algorithm was applied for only 3 selected features. As the dataset was imbalanced, an Undersampling was done. Then z-score normalization was performed to the Dataset. Normalizing the inputs makes it easier for the algorithm to find the optimal weights.

2.1 Correlation Matrix

For Feature selection a correlation matrix was printed for understating the correlation between different features . From the correlation matrix it is shown that the 'Glucose' attribute is highly correlated with the class variable. Then 'BMI' and age attributes are also more correlates with class variable than others . The correlation matrix is shown below:

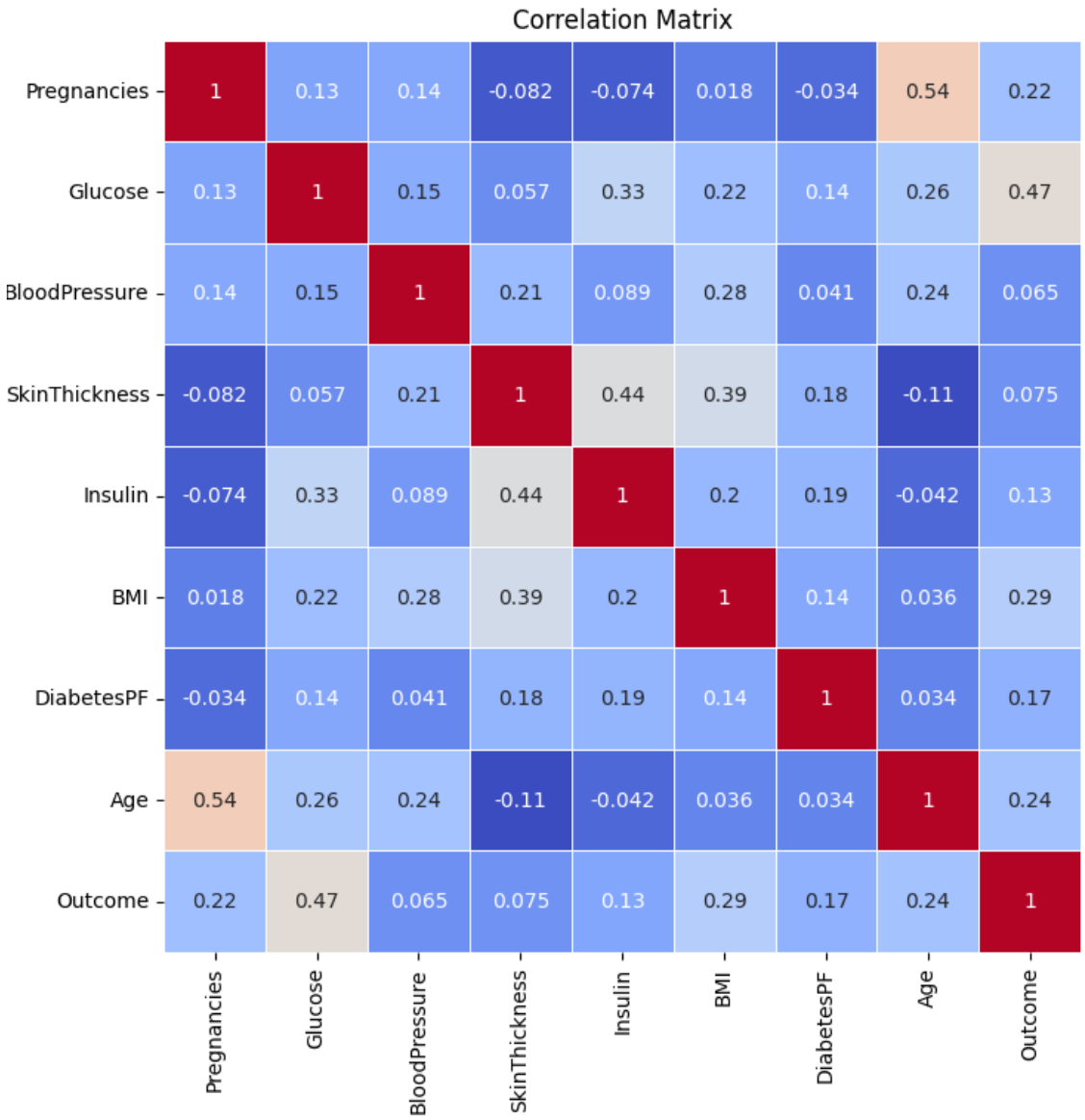

Figure 1: Correlation Matrix

Based of Correlation Matrix It was shown that "Glucose " ,"BMI" , "Age" Attributes are more correlated with output than others. After Plotting it was clear that the Dataset was not fully linearly separable . There were overlapping and Noise in the Dataset.

2.2 Balancing The Dataset

As shown in the class distribution in dataset analysis, the instances of class 0 was 500 and class 1 was 268. Here class 0 is almost double of class 1. So it was needed to be undersampled. Undersampling was done by **RandomUnderSampler** provided by **'imbalanced-learn'** library in python. After Sampling instances of both classes were equal.

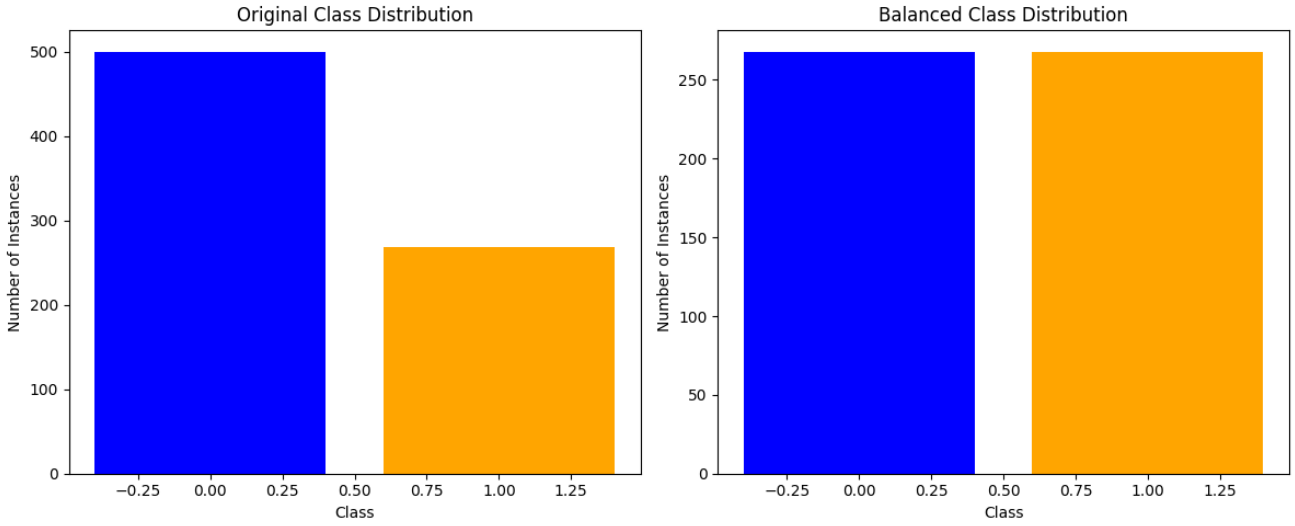

Figure 2: Original Class Distribution vs Balanced Class Distribution

2.3 Normalization

A z-score normalization was done by using this formula:

$$Z = \frac{X - \mu}{\sigma}$$

where:
Z is the is the standardized value,
X is the original value of the feature,
μ is the mean of the feature,
σ is the standard deviation of the feature.

3 Multilayer Perceptron Learning Algorithm

The Algorithm were described as follows[2] :

1. Initialise weights and threshold
   Set all weights and thresholds to small random values.

2. Present input and desired output Present input $X_p = x_0, x_1, x_2, \ldots, x_{n-1}$ and desired output $T_p = t_0, t_1, \ldots, t_{m-1}$ where n is the number of input nodes and m is the number of output nodes. Set w_0 to be $-\theta$, the bias, and x_0 t o be always l. For pattern association, X_p and T_p represent the patterns to be associated. For classification, T_p is set to zero except for one element set to 1 that corresponds to the class that X_p is in.

3. Calculate actual output
   Each layer calculates

$$y_{pj} = f\left[\sum_{i=0}^{n-1} w_i x_i\right]$$

and passes that as input t o the next layer. The final layer outputs values o_{pj}

4. Adapt weights

   Start from the output layer, and work backwards.

$$w_i(t+1) = w_i(t) + \eta \delta_{pj} o_{pj}$$

   $w_{ij}(t)$ represents the weights from node i to node j at time t, η is a gain term,and δ_{pj} is an error term for pattern p on node j .

   For output units

$$\delta_{pj} = k o_{pj}(1 - o_{pj})(t_{pj} - o_{pj})$$

   For hidden units

$$\delta_{pj} = k o_{pj}(1 - o_{pj}) \sum_k \delta_{pk} w_{pk}$$

where the sum is over the k nodes in the layer above node j .

4 Accuracy Analyzing

In my selected dataset for applying multilayer Neural Networks (Perceptron) algorithm was applied. In this case the input layer had 3 nodes and the output layer had 1 node . There were 50 hidden nodes in the network. The Mean Squared Error (MSE) per epoch is shown in Figure 3 . After 50000 epoch the MSE was 0.0428 ,which was quite well. But in testing with 20% of data the accuracy was 0.694 . By seeing this, it can be said that the Multi Layer Perceptron Algorithm Overfitted the training dataset. Thats why the testing accuracy is low but the Training accuracy was high. There could be another reason that the selected dataset was noisy , after overfitting the training data the model could not perform well to new data.

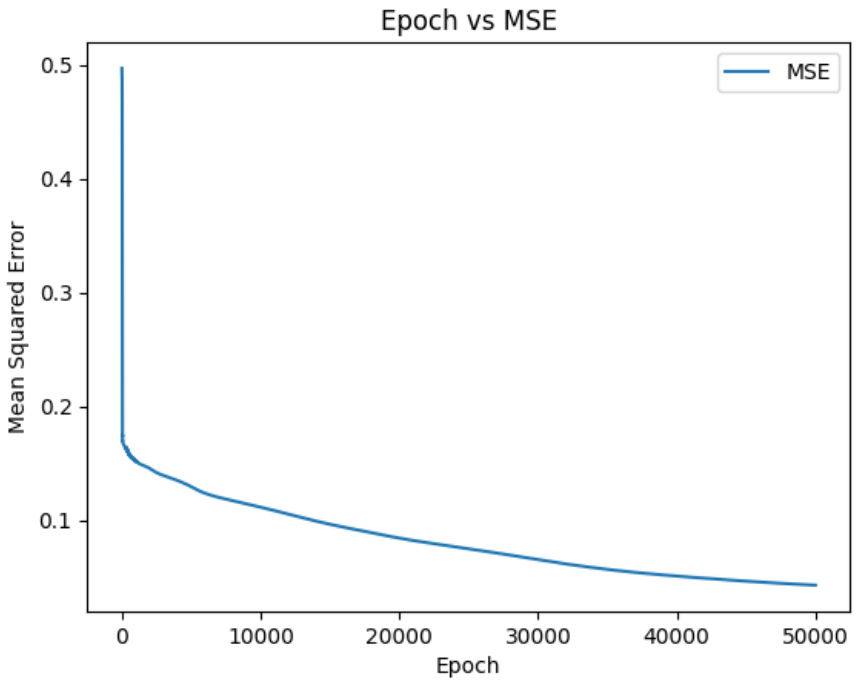

Figure 3: MSE vs Epoch

5 The XOR Problem

The single layer perceptron algorithm was unable to solve the XOR problem. The input and output is given below in Table 1. After Plotting the XOR table in Fugure 4 , where

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Table 1: XOR Table

yellow dots are indicating 1 and other dots indicating 0 . Which can not be separable by using a single line. So it is not linearly separable. Here the single layer perceptron fails to classify the XOR as it can classify when data can be separable by a single line.

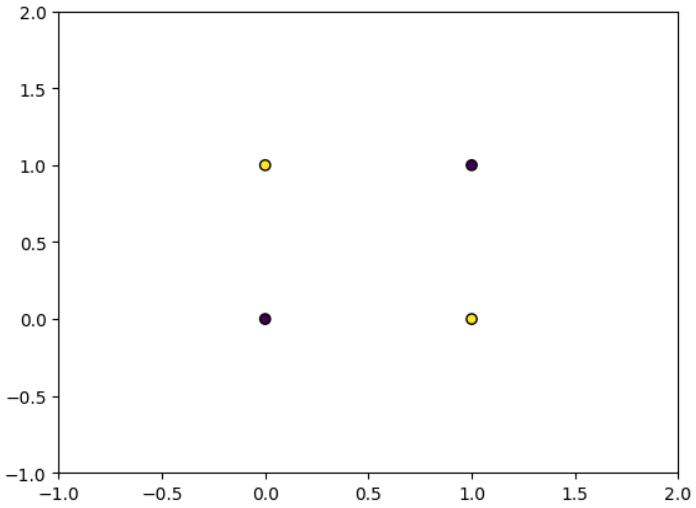

Figure 4: The XOR problem

5.1 Solution of the XOR Problem

The Single layer perceptron is failed to solved the XOR problem , The Multi Layer perceptron can solve this problem. It can draw multiple line between the classes to separate them and classify them more accurately than Single Layer Perceptron Algorithm. For solving the XOR problem a network was made by 2 input, 3 hidden and 1 output layer.

5.2 Analysis of XOR Problem Solution

After Applying the Mulilayer Perceptron on the XOR function it was able to classify successfully. After 10000 epochs the MSE was 0.002 . Decision boundary was plotted based on the network's output is shown below :

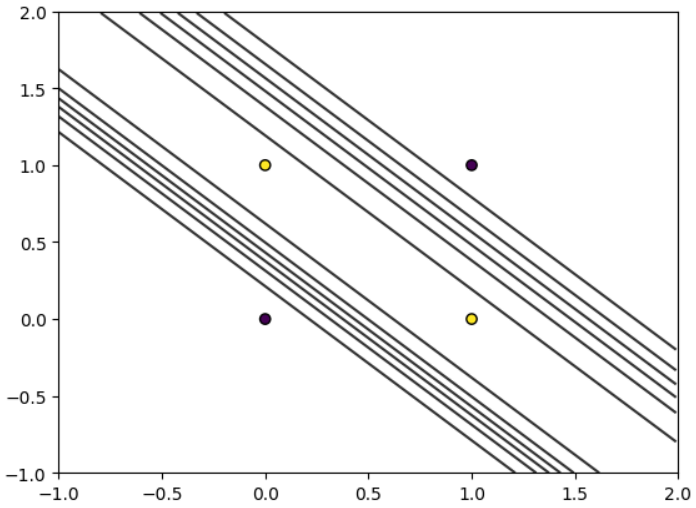

Figure 5: Decision Boundary Based on Multi Layer Perceptron on XOR Problem

6 Conclusion

In conclusion, the analysis and implementation of the neural network models on the selected Diabetes dataset and the XOR problem provide valuable insights into the challenges and capabilities of these algorithm.The multilayer perceptron, a kind of learning algorithm, did well during training, learning complicated patterns. But, when it was tested, there were some challenges, like it might have learned too much from the training data. The Limitations of Single Layer perceptron model for solving the XOR problem was overcomed by the Multi Layer Neural Network (Perceptron) Algorithm .

References

[1] Diabetes Dataset, [Online]. Available at: `https://www.kaggle.com/datasets/mathchi/diabetes-data-set`

[2] R. Beale and T. Jackson, *Neural Computing: An Introduction*, CRC Press, 1990.