Artificial Neural Networks

Name: Alekhya Devi Ranabothu

UID: U00924587

In this assignment Neural Networks is implemented from scratch using python. Gradient Descent algorithm is implemented to find out the best parameters.

Note:

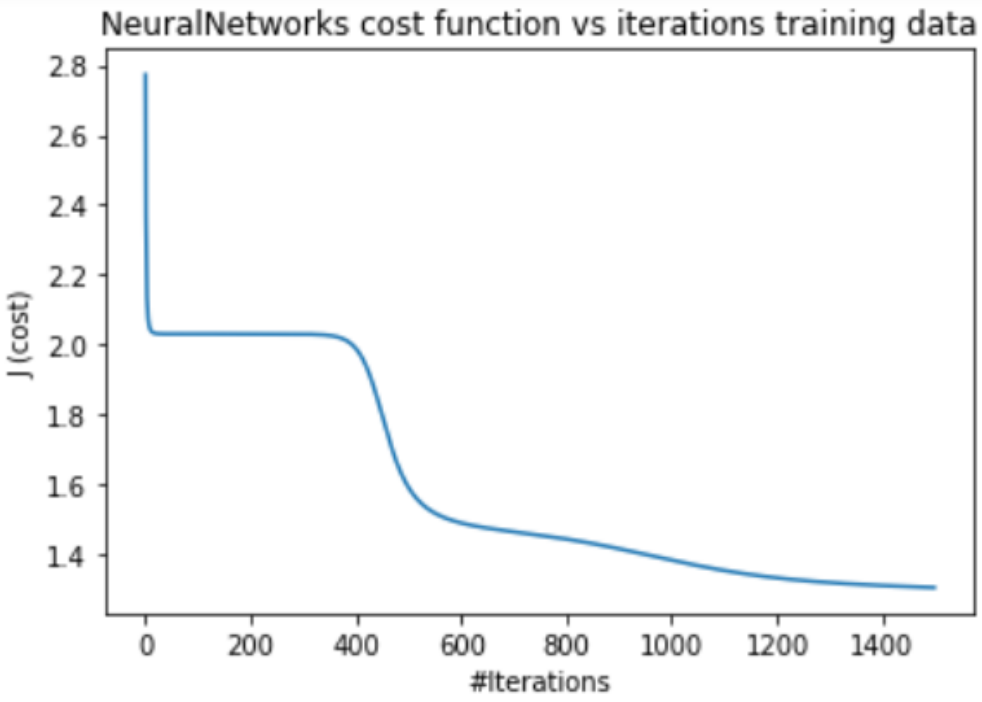
- Convergence condition is reached when the cost in the previous iteration is equal to the cost in the current iteration.As the cost is decreasing very slow, a very small difference between the costs (0.0000001) is used as convergence condition. Number of iterations used are 1500.
- The parameters(thetas) are initialized with random values.
- The BSOM data is split into training data (80%) and test data (20%) and the metrics on test data are used for performance evaluation.
- Missing values in the data are handled by removing the rows containing the missing values.
- Ridge Regularization is used.
- As this is imbalanced data, weighted average is used for calculating precision, recall and F1-score.
- Number of layers are considered including input layer and output layer.
- Feature scaling is done for the variables

1) Neural Network with BSOM Dataset

a) Neural Network is implemented with 4 independent variables ('all_mcqs_avg_n20' 'all_NBME_avg_n4', 'CBSE_01' and 'CBSE_02') and target variable 'LEVEL' with single hidden layer with 5 hidden nodes. The target variable has 4 classes – A, B, C, D.
Gradient Descent algorithm is implemented with different learning rates on training data and the learning rate that gives more F1 score (accuracy) is considered as good learning rate to train the data and the parameters obtained after training the data with this good learning rate are used to predict the test data.

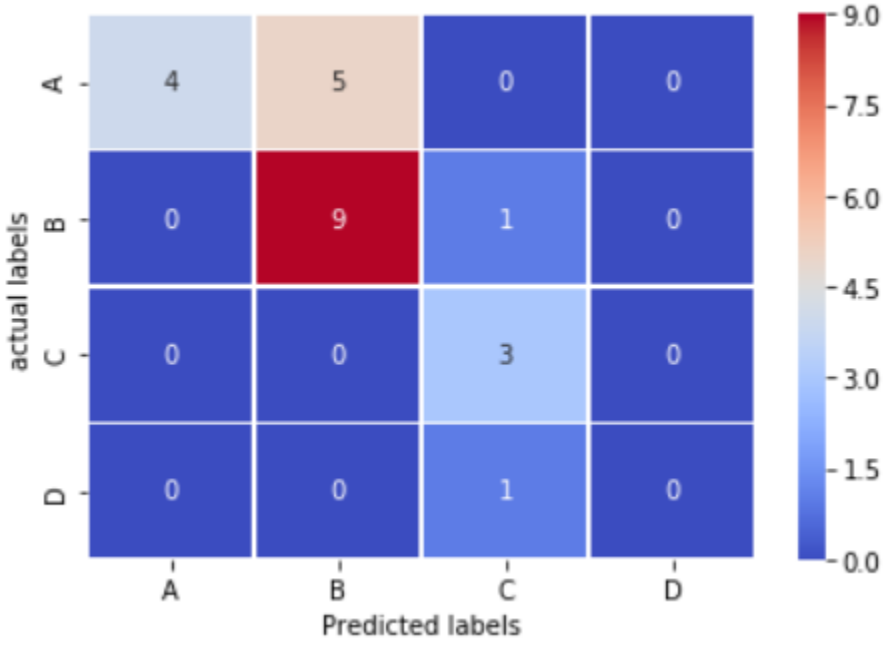
Learning Rate	F1-score
0.01	0.2329
0.1	0.2329
0.5	0.6612
0.6	0.6616
0.7	0.6724

From the above table the F1 score for learning rate 0.7 is more. So, we take 0.7 as good learning rate to apply on training data.
Below is the plot of cost function vs number of iterations for the neural network with 0.7 as learning rate, to verify that the cost decreases with the increase in number of iterations.



The optimum parameters obtained from training data are used to predict the test data.
Below metrics are used to evaluate the performance of test data.

Precision: 0.74906
Recall: 0.69565
F1 score: 0.66471
AUC score: 0.75769
Confusion Matrix:



Out of 23 values, 16 values are predicted correctly. Class D have 0 predicted values because of very less training samples for class D.

b) Now the neural network is tested for different number of hidden nodes with single hidden layer with learning rate 0.7.
Below are the evaluation metrics on the test data.

#hidden nodes	Precision	Recall	F1-score	AUC score
2	0.70082	0.60869	0.59574	0.70769
3	0.72408	0.65217	0.63021	0.73269
5	0.74906	0.69565	0.66471	0.75769
7	0.74906	0.69565	0.66471	0.75769
9	0.74906	0.69565	0.66471	0.75769
50	0.18903	0.43478	0.26350	0.5

From the above example it can be observed that with the increase of number of hidden nodes, the performance of the model is increased to a certain number of nodes (till 5 nodes) and after that it is constant till few increases (for 7 and 9) and decreased with very large number of nodes(100 nodes).
This is because as the number of nodes increases, the model learns more complex features and the performance increases. But after certain number of nodes, the complexity of the model will be more compared to that of data. So, the performance will not increase or decreases after certain number of nodes.
Here, good number of hidden nodes is 5.

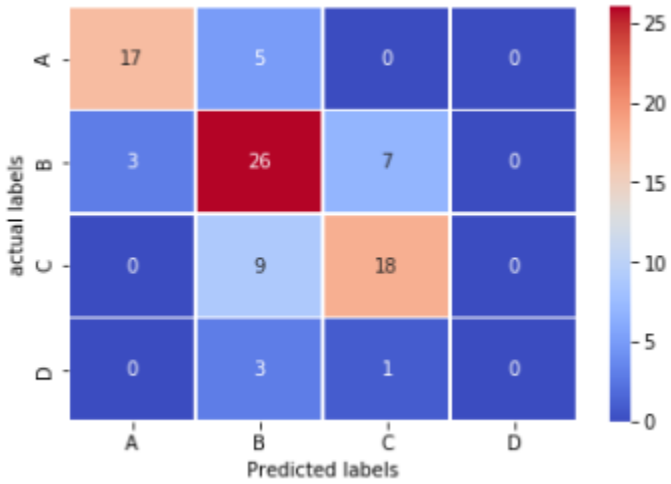
c) Now the neural network is tested for different number of hidden layers and each hidden layer has 5 nodes with learning rate 0.7.
Below are the evaluation metrics on the test data.

#hidden layers	Precision	Recall	F1-score	AUC-score
1	0.74906	0.69565	0.66471	0.75769
2	0.18903	0.43478	0.26350	0.5
3	0.18903	0.43478	0.26350	0.5
4	0.18903	0.43478	0.26350	0.5

As the number of hidden layers increases, performance of the model decreases because with increase in the hidden layers the complexity of the model increases, and we have very less data to train these complex models with many hidden layers.

d) The data set is imbalanced data where class D samples are very less, and class B samples are more compared to other classes.
Below are the metrics of training data.
Precision: 0.66471
Recall: 0.68539
F1 score: 0.67241
AUC score: 0.75271

Confusion Matrix:



From the confusion matrix, it can be observed that the class with more samples (class B) has a greater number of correct predictions and the class with least number of samples (class D) are not predicted correctly. But the AUC score is 0.75. So, I believe AUC is not a better metric for this classification.
By classifying this data set, we should be able to correctly identify or predict the low performing students so that their academics can be taken care before they fail. For this it is very important to correctly predict class D. That means out of 4 samples of class D (last row of confusion matrix), all the 4 should be predicted correctly. In the above confusion matrix, class D has 4 False Negatives which are False Positives for class B and C. That means cost of both False Positives and False Negatives, is important in this classification. Hence, both precision and recall scores are important. So, I believe F1- score is a better metric for this classification as it provides trade off between precision and recall. Weighted average is considered for calculating the metrics as it takes into account the number of samples of each class which is important for imbalanced data.

2) Regularization

a) The best model selected from section 1 is the neural network with single hidden layer with 5 hidden nodes. Ridge regularization is applied to this model with learning rate 0.7 with different regularization parameters. Below are the evaluation metrics on test data.

#Regularization Parameter	Precision	Recall	F1-score
0.01	0.74906	0.69565	0.66471

0.1	0.74906	0.69565	0.66471
1	0.70496	0.60869	0.57246
5	0.18903	0.43478	0.26350
10	0.18903	0.43478	0.26350

From the metrics in the above table, it is observed that the performance of the model is not improved by regularization.

b) All the weights of the model are initialized with same value ('3') and tested with different nodes and different layers with learning rate 0.7. The model here is with single hidden layer and 5 hidden nodes. Below are the metrics on test data.

Table 1:

#hidden nodes	**Precision**	**Recall**	**F1-score**
2	0.72388	0.65217	0.64761
3	0.70082	0.60869	0.59574
5	0.60609	0.56521	0.55031
7	0.60609	0.56521	0.55031
9	0.60609	0.56521	0.55031
50	0.15311	0.39130	0.22010

Table 2:

#hidden layers	**Precision**	**Recall**	**F1-score**
1	0.60609	0.56521	0.55031
2	0.18903	0.43478	0.26350
3	0.18903	0.43478	0.26350
4	0.18903	0.43478	0.26350

By comparing the Table 1 with the table in section 1.b, it can be observed that when all the weights are initialized to the same value, the model with a smaller number of nodes have good performance. And the performance is reduced when compared to the model with randomly initialized weights.
By comparing the Table 2 with the table in section 1.c, it can be observed that when all the weights are initialized to the same value, the performance of the model is reduced compared to the model with randomly initialized weights.

If the weights are initialized with same value, the model does not have symmetry breakage. That means all the nodes in all the layers will perform the same calculation and gives same output. So, this makes the deep neural network as a simpler model.

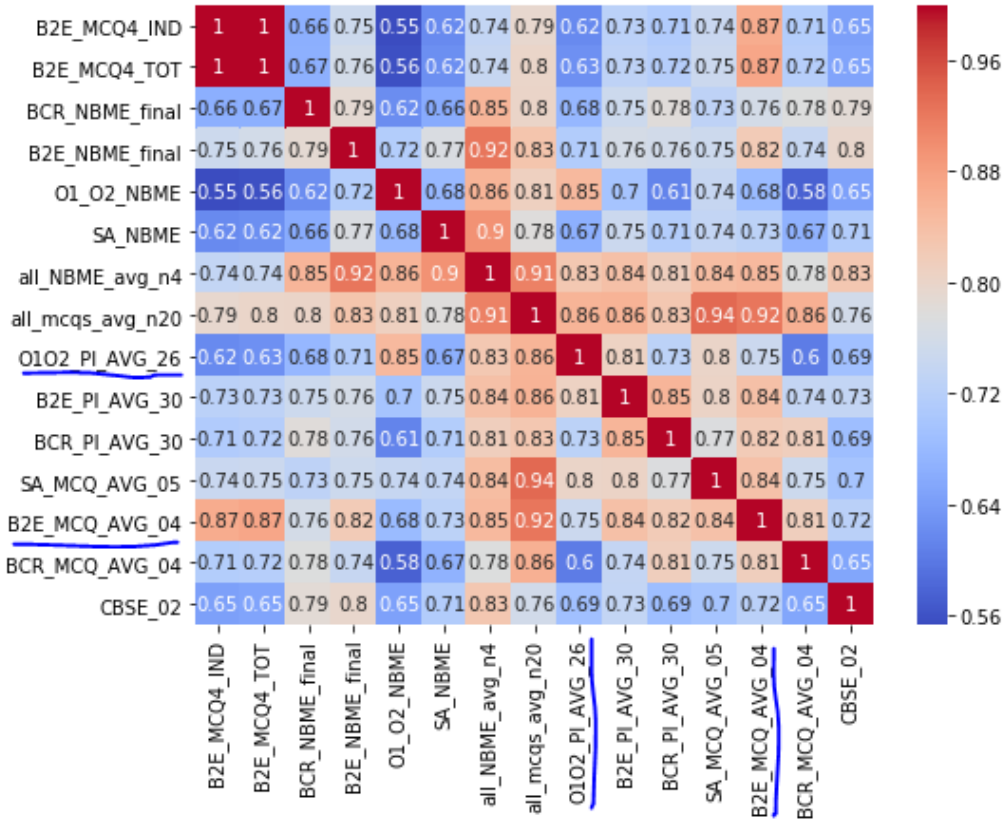
c) Precision Recall and F1- scores are evaluated and compared on the best model selected from question 1. In question 2 a, Ridge regularization is applied to the best model with different values of regularization parameter and metrics on test data are compared and observed that regularization did not improve the performance.
In question 2b, all the weights of the best model are initialized to same value and found that symmetric breakage is a critical issue.

3) <u>Neural Network with more variables</u>

From the variables 'STEP_1' and 'LEVEL' it can be observed that the students with 'STEP_1' scores above 240 has LEVEL 'A', the students with 'STEP_1' scores from 220 to 239 has LEVEL 'B', the students with 'STEP_1' scores from 200 to 219 has LEVEL 'C' and the students with 'STEP_1' scores below 200 has LEVEL 'D'. That means if the model knows the 'STEP_1' score, it can learn it while training and fail to generalize. So, while selecting the features, we will not consider the variable 'STEP_1'. As the target variable 'LEVEL' is a categorical variable and is strongly correlated with 'STEP_1', we can select the features that are highly correlated with the variable 'STEP_1'.
Below are the few variables which are correlated with 'STEP_1'in descending order of correlation values.

all_NBME_avg_n4	0.825868
CBSE_02	0.814663
B2E_NBME_final	0.809040
all_mcqs_avg_n20	0.757584
BCR_NBME_final	0.745923
B2E_MCQ_AVG_04	0.735672
B2E_PI_AVG_30	0.721486
B2E_MCQ4_TOT	0.682846
O1_O2_NBME	0.678500
B2E_MCQ4_IND	0.672767
SA_MCQ_AVG_05	0.671104
SA_NBME	0.667767
BCR_MCQ_AVG_04	0.659748
BCR_PI_AVG_30	0.649210
O2_MCQ_AVG_03	0.648281
O1O2_PI_AVG_26	0.647610
O2_PI_AVG_13	0.644639
SA_MCQ4_TOT	0.617616
SA_PI_AVG_26	0.617490

Now among these variables, we select randomly the variables which may increase the performance.
Below is the correlation matrix of few of the above listed variables.



a) The neural network model with single hidden layer and 5 hidden nodes is trained with 5 variables- 'all_mcqs_avg_n20','all_NBME_avg_n4','O1O2_PI_AVG_26','CBSE_01' and 'CBSE_02', with learning rate 0.7, regularization parameter 0.1. Below are metrics on test data.

Precision: 0.73602
Recall: 0.69565
F1 score: 0.68012

b) The neural network model with single hidden layer and 5 hidden nodes is trained with 6 variables- 'all_mcqs_avg_n20','all_NBME_avg_n4','O1O2_PI_AVG_26', 'B2E_MCQ_AVG_04','CBSE_01' and 'CBSE_02', with learning rate 0.7, regularization parameter 0.1. Below are metrics on test data.

Precision: 0.79968
Recall: 0.782608
F1 score: 0.75362

c) By comparing the metrics of question 2 with regularization parameter 0.1 and the metrics in questions 3a and 3b, it can be observed that the performance has been increased by adding the variables.
Below is the comparison of the metrics on test data.

#variables	**precision**	**recall**	**F1-score**
4(question2)	0.74906	0.69565	0.66471
5(question 3a)	0.73602	0.69565	0.68012
6(question 3b)	0.79968	0.782608	0.75362