APPLIED MACHINE LEARNING

ASSIGNMENT 3

Srinikethan Pusthay (SXP210162)

Goal: Implement Artificial Neural Networks (ANN) on the dataset to predict the rented bike count.

Dataset:

The dataset contains count of public bikes rented at each hour in Seoul Bike haring System with the corresponding Weather data and Holidays information.

Tasks:

We have used the data set from previous assignment, and we have modified the features same as previously. To convert the dataset into binary classification problem we have converted the output to class label. We have considered the mean value of bike count 704 as the threshold and transformed the column to 1 if the values are above threshold and 0 if the values are below threshold. The data is then split into train and test by 70% and 30%.

We have used TensorFlow and keras to develop the Artificial Neural Network learning algorithm. We have performed various experiments by changing the number of layers, number of nodes and activation functions. Then we have plotted graphs of loss and accuracy for best models. Also, calculated confusion matrix for them.

Experiments and Results:

To develop ANN algorithm, we have used packages from TensorFlow and keras. We have developed the model and experimented with various parameters like number of layers, number of nodes and activation functions.

We have considered four activation functions in our experiments, Rectified Linear Unit (relu), Hyperbolic Tangent (tanh), Scaled Exponential Linear Unit (selu) and softmax. For all the experiments, output layer is using sigmoid activation function due to its property. For small values (<5), sigmoid returns a value close to zero, and for large values (>5) the result of the function gets close to 1.

Other parameters which are fixed across all the experiments are epochs, batch size, learning rate, validation split, binary cross entropy as loss etc. After developing the model, we have performed testing using the test data and calculated respective Accuracy. Below are the results of various experiments:

Accuracy Score for model with 5 layers and 64 nodes initially	
Activation Function	Accuracy (%)
relu	69.52
tanh	71.27
selu	70.77
softmax	46.31

Accuracy Score for model with 8 layers and 64 nodes initially	
Activation Function	Accuracy (%)
relu	71.65
tanh	71.95
selu	71.11
softmax	46.31

Accuracy Score for model with 8 layers and 128 nodes initially	
Activation Function	Accuracy (%)
relu	69.93
tanh	71.46
selu	71.04
softmax	46.31

Accuracy Score for model with 8 layers and 128 nodes initially	
Activation Function	Accuracy (%)
relu	69.59
tanh	71.84
selu	72.14
softmax	46.31

Accuracy Score for model with 12 layers and 256 nodes initially	
Activation Function	Accuracy (%)
relu	71.53
tanh	**75.22**
selu	**74.16**
softmax	46.31

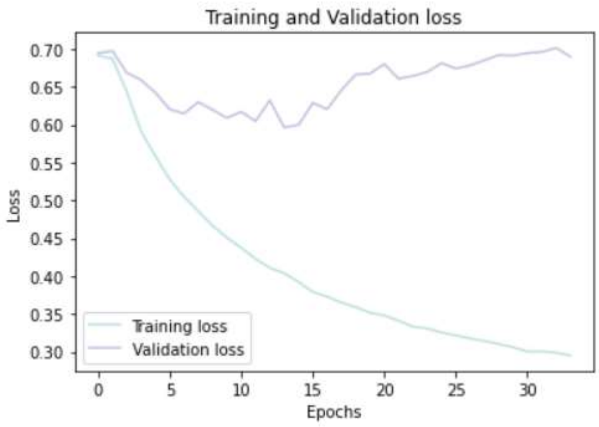
Accuracy Score for model with 16 layers and 1024 nodes initially	
Activation Function	Accuracy (%)
relu	46.31
tanh	75.41
selu	46.31
softmax	46.31

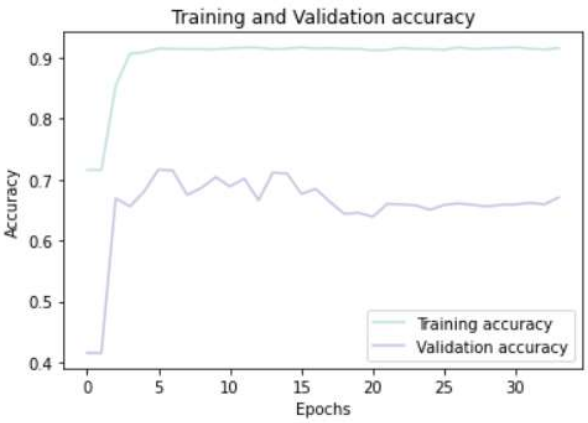
Experimenting with various number of layers and nodes across the activation functions has produced the above results. There is an early stopping check while running the epochs, if there is not much improvement from the models, the model is stopped processing further.

We have gradually increased the number of layers and nodes in different steps. We have obtained the maximum accuracy at 12 layers and 256 nodes in the first layer for tanh and selu activation functions. We have also performed validation with a validation split of 0.2, and have a constant learning rate.

We have selected these two models to continue with our experiments. Using these models, we have plotted the loss vs epochs and accuracy vs epochs graphs.

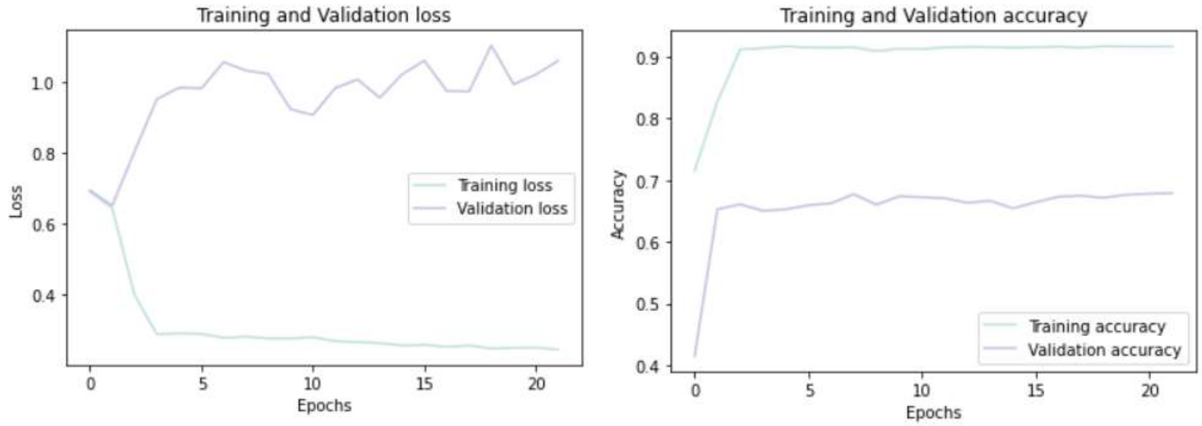
Activation Function: tanh





From both the graphs, we can see that training data has performed very well. But with validation data, the results are not the same. As number of epochs is increasing, loss for training has been decreasing. But for validation it has decreased a bit initially but later it has increased. Accuracy trend is similar for both validation and training data, increased initially and remained constant with the increase in number of epochs. Training accuracy is higher than the validation accuracy.

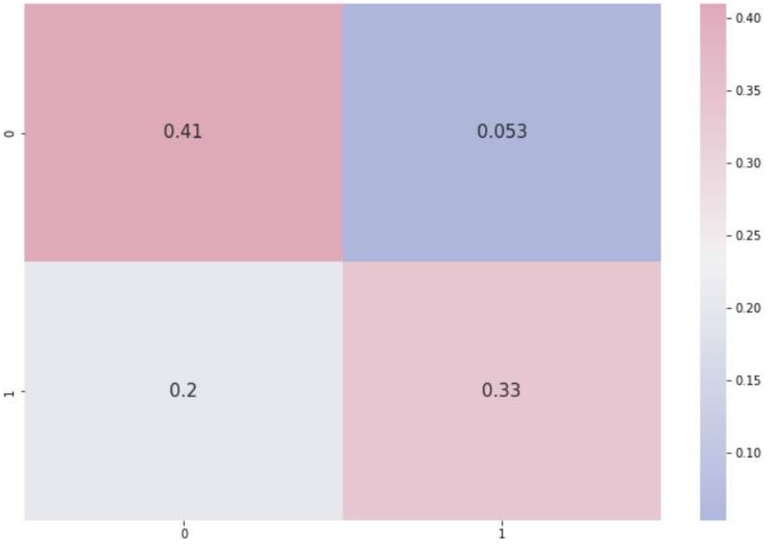
Activation Function: selu

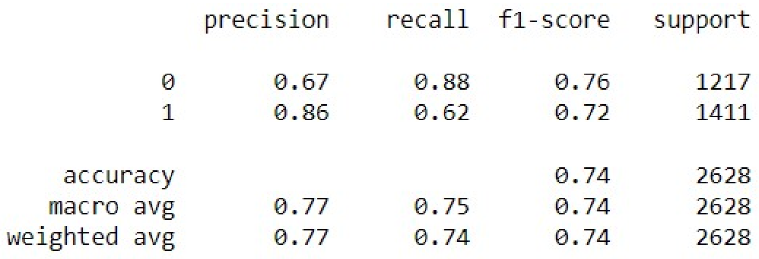


Results of curves are like tanh function curves; model has performed well on training data than validation data. As number of epochs are increasing, training loss has decreased whereas validation loss has been increasing. Trends for accuracy is same for both, increased initially and remained constant. And training accuracy is higher than validation accuracy.

In both the activation functions, training data has performed well than validation data. Next, we have performed testing using the test data with these models. Below are the results of the confusion matrix of the tests.

Confusion Matrix and Performance Measures for tanh:

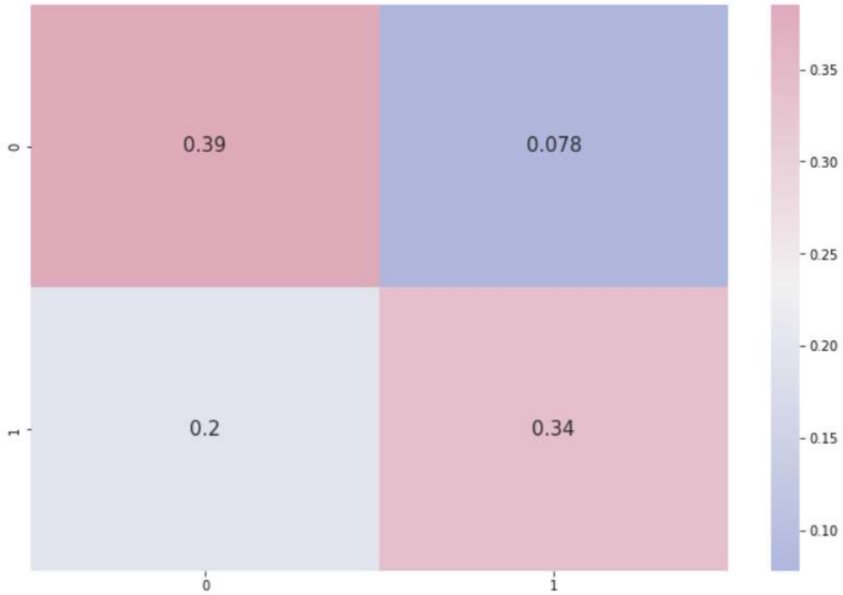


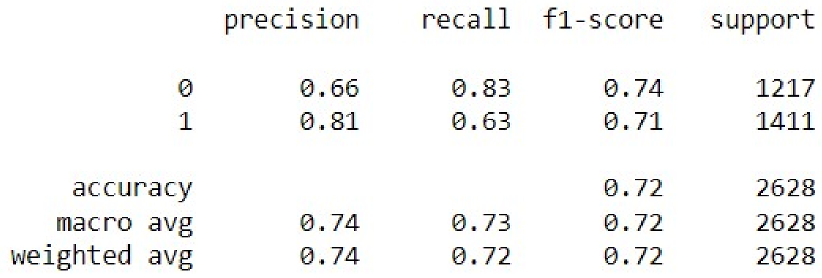


```
              precision    recall  f1-score   support

           0       0.67      0.88      0.76      1217
           1       0.86      0.62      0.72      1411

    accuracy                           0.74      2628
   macro avg       0.77      0.75      0.74      2628
weighted avg       0.77      0.74      0.74      2628
```

Various performance measures and confusion matrix can be observed from the above results. Accuracy for the test data is 74%.

Confusion Matrix and Performance Measures for selu:





```
              precision    recall  f1-score   support

           0       0.66      0.83      0.74      1217
           1       0.81      0.63      0.71      1411

    accuracy                           0.72      2628
   macro avg       0.74      0.73      0.72      2628
weighted avg       0.74      0.72      0.72      2628
```

Accuracy for test data obtained is 72%.

Comparison of different learning algorithms test accuracy:

Learning Algorithm	Accuracy
Artificial Neural Network	74
Decision Tree	80.47
SVM (Kernel = RBF)	79.53

From various learning algorithms, accuracy of test data is as shown above in the table. Decision Tree has the highest accuracy, next to SVM and ANN at the last. (Decision Tree > SVM > ANN)

There can be many factors influencing the accuracies. We can perform many more experiments in ANN to improve the model efficiency and obtain optimal parameters which can outperform other algorithms.