**CS5590 APLS- DEEP LEARINING PROGRAMMING**

**LAB 2**

**Team 1**

**TEAM MEMBERS:**

**ESWAR VALLURU-27**

**BHAVYA SRI SIRASANAGANDLA-21**

**MANIDEEP NADIMPELLI -16**

**Introduction:**

This lab mainly focuses whether we are fully aware of the concepts such as Convolution Neural Networks, Autoencoder, TensorBoard, Optimizers, Loss functions, LSTM and so on.

**Objective:**

The main objective of this lab assignment is to get good experience in Keras library by using neural networks like LSTM, CNN and also machine learning algorithms like Logistic regression and Linear regression.

**DataSets:**

* Boston Housing
* Heart
* 10 monkey species
* Movie review
* Mnist

**Problem 1:**

**Problem Statement**

A screenshot of a social media post

Description automatically generated

Step 1:

Importing all the packages that are necessary for the execution of linear regression sequential model.

A screenshot of a cell phone

Description automatically generated

Step 2:

Loading the boston data set.



Step 3:

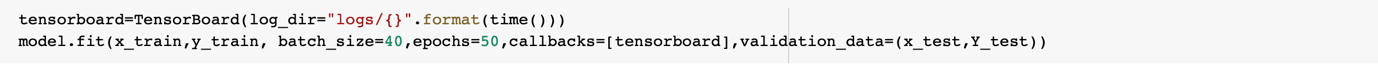
Defining the required optimizer, and creating the sequential model. We have chosen relu and softmax as activation functions. We have complied the model using rmsprop as optimizer and mae as loss function.

A screenshot of a cell phone

Description automatically generated

Step 4:

Initialising the tensor board and fitting the model using the tensorboard as call backs.

A screenshot of a social media post

Description automatically generated

Step 5:

Using the following commands to load the tensor board logs and generate the graph.

A screenshot of a cell phone

Description automatically generated

Step 6:

Below are the tensor board graphs that are generated.

A screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

Description automatically generated

Step 7:

After changing the hyper parameters

Leaning rate from 0.1 to 0.01, the loss got increased.

Batch size from 40 to 50, there is no much change in the loss.

Optimizer from adam to rmsprop, the loss is less.

Activation function from sigmoid to softmax, then the loss has increased.

Below are the plotted losses.

A screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

Description automatically generated

**Problem 2:**

**Problem Statement**

A screenshot of a cell phone

Description automatically generated

Step 1:

Importing all the packages that are required for the logistic regression.

A screenshot of a cell phone

Description automatically generated

Step 2:

Read the data from heart.csv using read csv method.

A screenshot of a cell phone

Description automatically generated

Step 3:

Here we are using sequential model by adding few dense layers with activation function as relu, and sigmoid in the output layer. Now we compile the built model using the binary crossentropy as the loss function and adam as the optimizer.

A screenshot of a cell phone

Description automatically generated

Step 4:

After compiling, we fit the model using epochs as 50 and batch size as 30. After running it, we got an accuracy of 85%

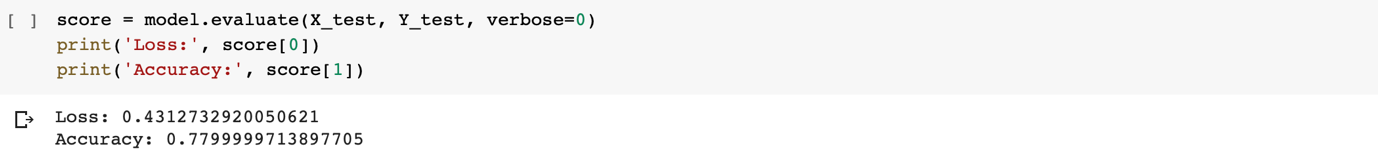
A screenshot of a cell phone

Description automatically generatedA close up of a piece of paper

Description automatically generated

Step 5:

We evaluate the model and the obtained score is loss of 43% and accuracy of 77%



Step 6:

Below are the commands to plot the graphs using the tensor board.



Step 7:

Below are the graphs for the accuracy, loss, val\_accuracy and val\_loss after and before changing the hyper parameters.

A screenshot of a cell phone

Description automatically generatedA screenshot of a social media post

Description automatically generated

**Problem 3:**

**Problem Statement:**

A screenshot of a cell phone

Description automatically generated

Step 1:

Import all the required packages for image classification.

A screenshot of a cell phone

Description automatically generatedA screenshot of a social media post

Description automatically generated

Step 2:

Load the data by reading the csv using pandas and read csv.

A screenshot of a social media post

Description automatically generated

Step 3:

Here we are reading the image and plotting it by using matplotlib package.

A close up of a logo

Description automatically generatedA screenshot of a social media post

Description automatically generated

A close up of a logo

Description automatically generated

Step 4: As there are only few images, we have used imageDataGenerator to add few more images.

Step 5: We have used a sequential model and conv2d layers, used relu and softmax as activation layers and maxPooling2D.

A screenshot of a social media post

Description automatically generatedA screenshot of a social media post

Description automatically generated

Step 6:

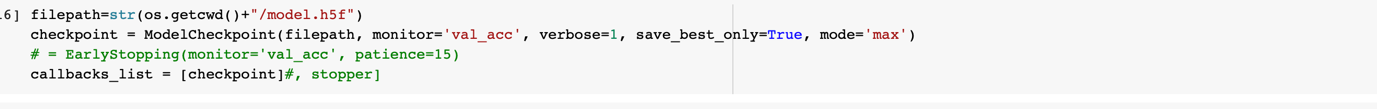
Have compiled the model using adam as optimizer and categorical\_crossentropy as loss function. And have printed summary.

A screenshot of a social media post

Description automatically generated

Step 7:

Now we have fit the model using fit\_generator.



A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

Step 8:

Plotted the training and testing accuracy and loss. At last have also printed the classification report.

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

**Problem 4:**

**Problem Statement:**



Step1 :

First we have to import all the packages that are necessary for the LSTM model.

A screenshot of a cell phone

Description automatically generated

Step 2:

We use pandas read csv function to read the csv files into training and test data frames. Next we drop few unnecessary columns along with the target column. Now we apply lambda function which will get applied for all the rows of the mentioned column. Here this function changes all the phrases that come from movie dataset into the lower case.

A screenshot of a cell phone

Description automatically generated

Step 3:

We consider the phrase column along which contains the reviews, we now tokenize the sentences into words. Then we transform these tokens into sequences, and also add some padding to make every word to a constant or fixed size. And we print the shape.

A screenshot of a social media post

Description automatically generated

Step 4:

Here in this step we add layers such as embedding, LSTM and output layers. While compiling we have used categorical crossentropy as loss function and adam as optimizer. We have used label encoder to transform the non numeric data to numeric data and perform one hot encoding. Then we will split the data into training data and testing data.

A screenshot of a social media post

Description automatically generated

Step 5:

Now we fit the model with all the above parameters, taking epochs as 5 and batch size as 40. After running the epochs, accuracy is obtained as 66%.

A screenshot of a cell phone

Description automatically generated

Step 6:

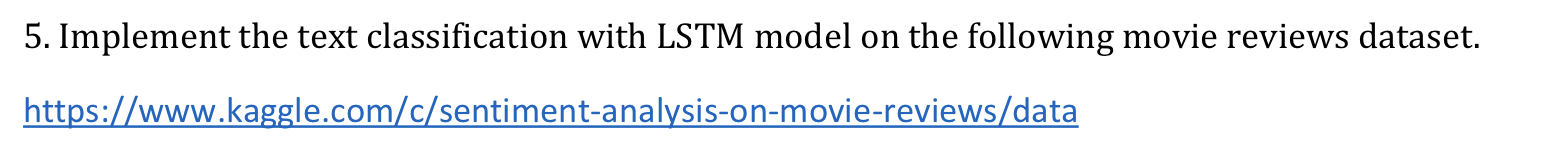
After fitting the model, we have to test it by predicting the sentiment. Now we take a phrase, predict the class of it, then compare it with the results provided in the sampleSubmission.csv. the actual and the predicted value is matched, which means our model is a good fit.

A screenshot of a cell phone

Description automatically generated

**Problem 5:**

**Problem Statement**



Step1 :

First we have to import all the packages that are necessary for the LSTM model.

A screenshot of a cell phone

Description automatically generated

Step 2:

We use pandas read csv function to read the csv files into training and test data frames. Next we drop few unnecessary columns along with the target column. Now we apply lambda function which will get applied for all the rows of the mentioned column. Here this function changes all the phrases that come from movie dataset into the lower case.

A screenshot of a cell phone

Description automatically generated

Step 3:

We consider the phrase column along which contains the reviews, we now tokenize the sentences into words. Then we transform these tokens into sequences, and also add some padding to make every word to a constant or fixed size. And we print the shape.

A screenshot of a social media post

Description automatically generated

Step 4:

Here in this step we add layers such as embedding, LSTM and output layers. While compiling we have used categorical crossentropy as loss function and adam as optimizer. We have used label encoder to transform the non numeric data to numeric data and perform one hot encoding. Then we will split the data into training data and testing data.

A screenshot of a social media post

Description automatically generated

Step 5:

Now we fit the model with all the above parameters, taking epochs as 5 and batch size as 40. After running the epochs, accuracy is obtained as 65%.

A screenshot of a cell phone

Description automatically generated

Step 6:

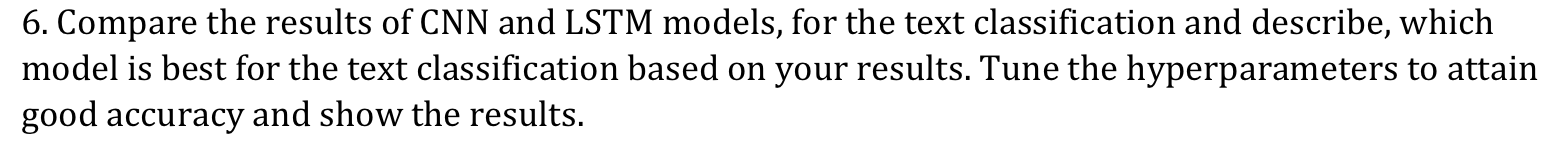
After fitting the model, we have to test it by predicting the sentiment. Now we take a phrase, predict the class of it, then compare it with the results provided in the sampleSubmission.csv. the actual and the predicted value is matched, which means our model is a good fit.

A screenshot of a cell phone

Description automatically generated

**Problem 6:**

**Problem Statement:**



Step 1:

Have used the same code of problem 4, but have changed a few hyper parameters to tune the data. We have used CNN model here, changed the loss function to binary crossentropy, modified the classifier to adam with learning rate, decreased the embed dimension. Have also reduced the batch size. Using all this we have built the model. The accuracy has increased from 66% to 86.05%.

A screenshot of a social media post

Description automatically generated

Step 2:

Have used the same code of problem 5, but have changed a few hyper parameters to tune the data. We have used LSTM model here, changed the loss function to binary crossentropy, modified the classifier to adam with learning rate, decreased the embed dimension and lstm\_out. Have also reduced the batch size. Using all this we have built the model. The accuracy has increased from 65% to 86.21%.

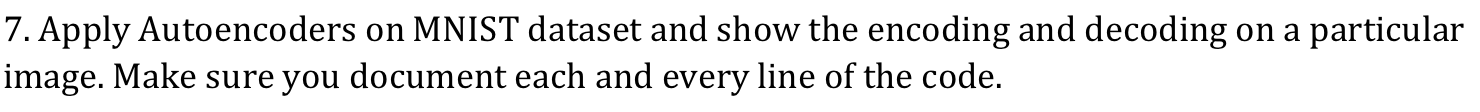
A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

**Problem 7:**

**Problem Statement:**



Step1:

From Keras Library Import the Input, Dense, Model packages

A screenshot of a cell phone

Description automatically generated

Step 2:

Encode the input image and decoded is the lossy reconstruction of the input image. Create a separate encoder and decoder model. Compiling the model .

A screenshot of a cell phone

Description automatically generated

Step 3:

Import mnist and load data to test and train data respectively to x\_train and x\_test. We have scaled the training and testing data. After scaling we are reshaping the data and fitting the data using autoencoder. Here we have used batch size as 256 and epochs as 20.

A screenshot of a cell phone

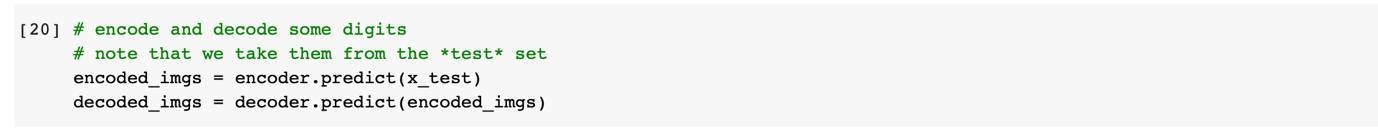
Description automatically generated

A close up of a logo

Description automatically generated

Step 4:

Using the encoder and decoder models to get the encoded images and decoded images



Step 5:

Display the original image which is in the dataset using imshow and decoded image using decoded\_imgs[ ]

A screenshot of a cell phone

Description automatically generated

**Evaluation & Discussion:**

Have performed all the objectives as mentioned and have met all the necessary criteria. All the programs were successfully executed.

**Conclusion:**

We learned how to apply deep learning concepts and algorithms to analyze different machine learning problems. We have also learned how to do the predictions on different datasets using appropriate algorithms.