**PYTHON DEEP LEARNING**

**PROJECT\_BASED\_EXAM-1 REPORT**

**INTRODUCTION:**

Main moto of this project is to implement Deep Learning and machine learning algorithms learnt in Python and Deep Learning class so far. Different algorithms used to in this project are CNN, LSTM, PCA, Text classification, Text generation and other Machine learning algorithms.

**OBJECTIVE:**

To implement different algorithms and models learnt in Python Deep Learning so far. And to apply different Machine Learning algorithms for obtaining efficient results.

**METHODS:**

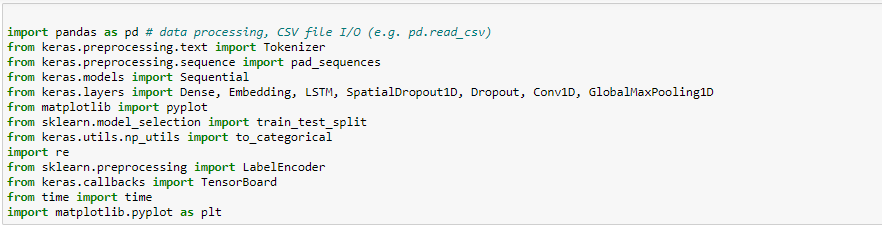
* CNN model: Convolutional neural network is a class of deep neural networks, most generally applied for evaluating visual imagery.
* LSTM model: LSTM model is a type of recurrent neural network that attains state-of-the-art fallouts on challenging prediction issues.
* Image classification.
* Text classification with CNN.
* Text generation with CNN.
* Auto-encoder with CNN.
* PCA

**WORK FLOW:**

**Question 1**:

**Implement text classification on the review’s sentiment dataset using CNN model.**

* Import all the necessary libraries.



* Read train and test files.
* Print the content in train and test files.

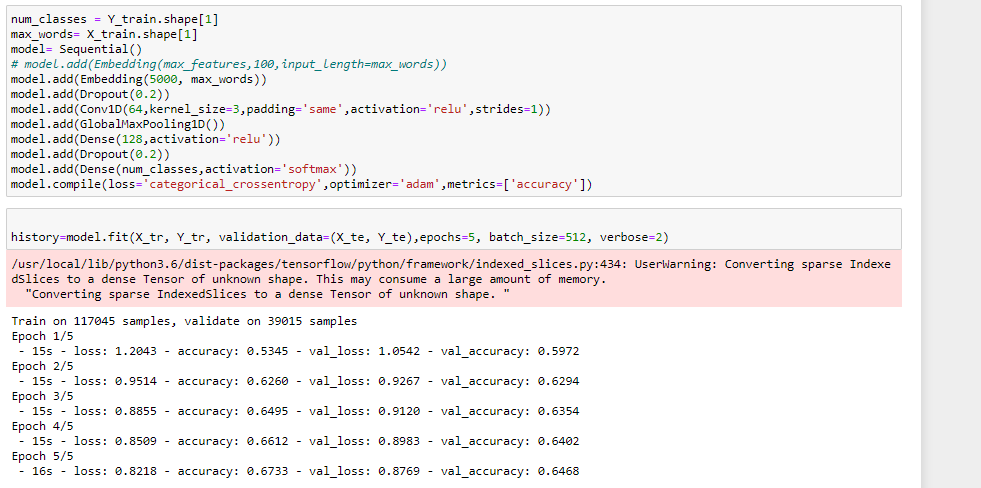


* Drop multiple columns in pandas dataframe.
* Specify maximum number of features needed to be considered.
* Initialize tokenizer for train data, where it obtains top maximum number of features and filters remaining.
* Repeat the same procedure of test data.
* Specify label encoder.
* Fit the model.
* Split the data to train and test and print the content.



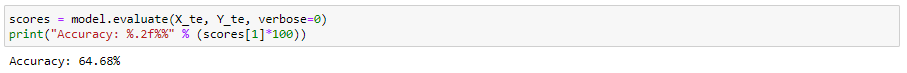
**a. Include Embedding layer in the design of your models and report if that leads to a better performance.**

* Import embedding from keras layers.
* Add embedding hidden layer.
* Fit the data and compile the model.

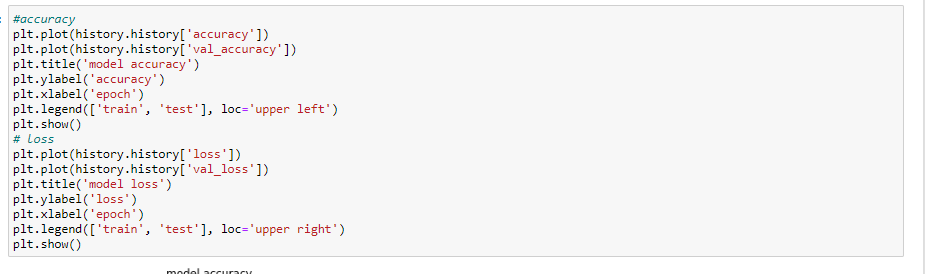


**b. Plot loss of the model and report if you see any overfitting problem.**

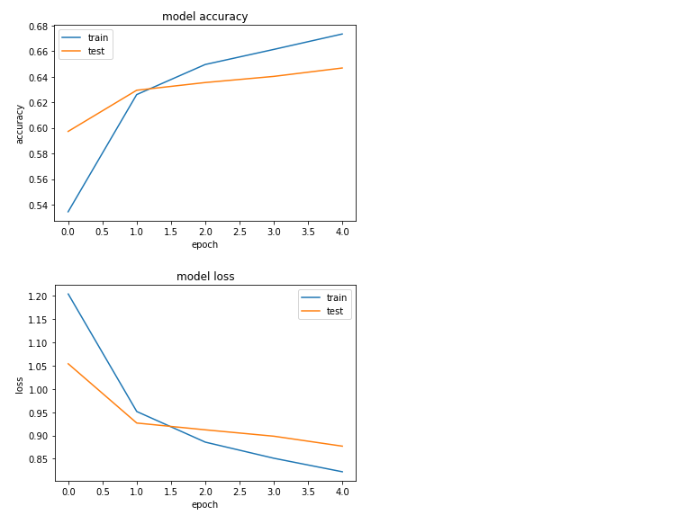
* Calculate accuracy and print accuracy value.



* Plot loss and accuracy.



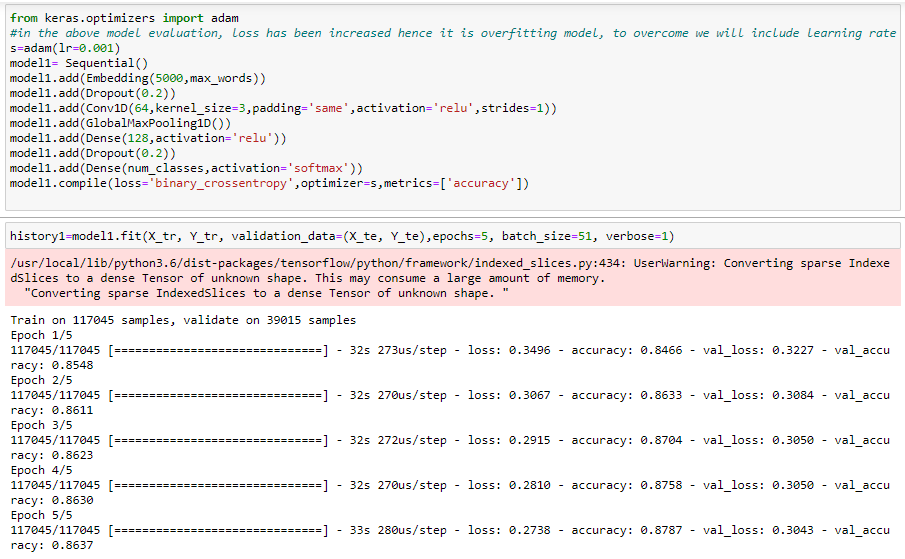
**PLOTS:**



**Inference:** From above procedure, we can draw that loss has increased and hence it is an overfitting model.

**c. What techniques you can apply to fix overfitting model.**

* Adding hidden embedding layers reduces this overfitting problem.



**Question 2: Implement text classification on the 20news\_group dataset using LSTM model**

**a. Include Embedding layer in the design of your models and report if that leads to a better performance**.

* Import the necessary libraries.

![](https://github.com/PallaviArikatla/Python\_Exam\_2/blob/master/Documentation/Question%202/1.import.PNG)

\* Then downloaded the 20 newgroups dataset and then load that into dataframe.

![](https://github.com/PallaviArikatla/Python\_Exam\_2/blob/master/Documentation/Question%202/2.PNG)

\* Apply the tokenizer to the dataset.

![](https://github.com/PallaviArikatla/Python\_Exam\_2/blob/master/Documentation/Question%202/3.tokenization.PNG)

\* Apply labelencoder to convert categorical data into numeric features.

![](https://github.com/PallaviArikatla/Python\_Exam\_2/blob/master/Documentation/Question%202/4.labelencoder.PNG)

\* Intialize the LSTM model and add the embedding layer.

![](https://github.com/PallaviArikatla/Python\_Exam\_2/blob/master/Documentation/Question%202/5.model.PNG)

\* Plot the history object and predict the loss and accuracy.

![](https://github.com/PallaviArikatla/Python\_Exam\_2/blob/master/Documentation/Question%202/5.history.PNG)

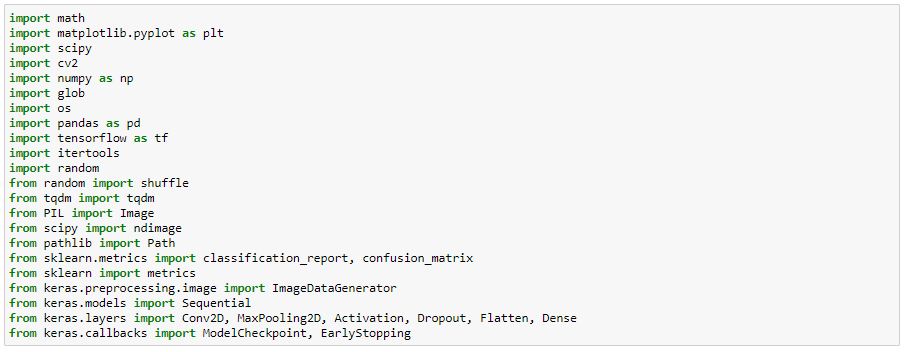
\*\*b. Plot loss of the model and report if you see any overfitting problem\*\*

\* Plot the loss and accuracy

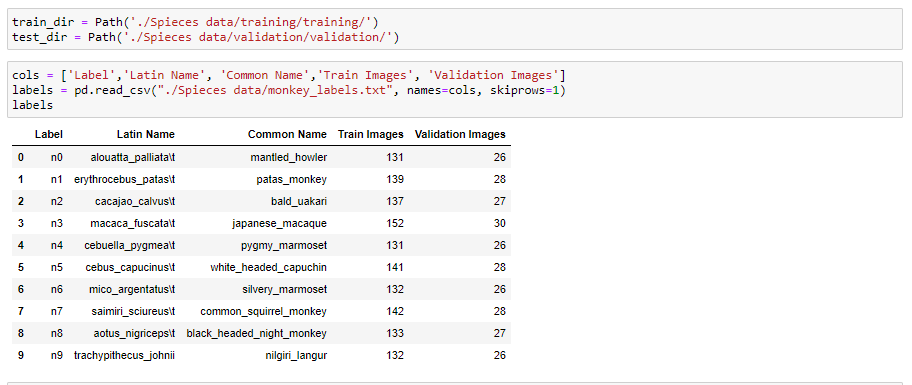
**Question 3: Implement image classification with CNN model, using a dataset.**

1. **Report your classification result with and without doing scaling**

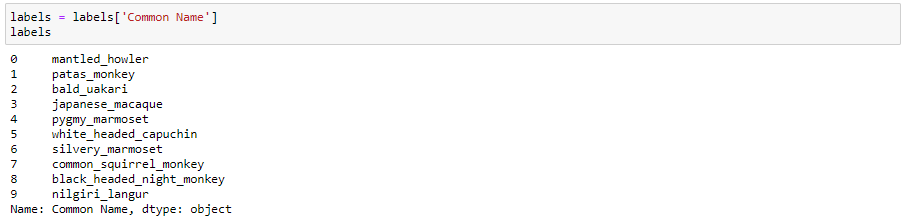
* Install scikit-image and opencv-python.
* Import all the necessary libraries.



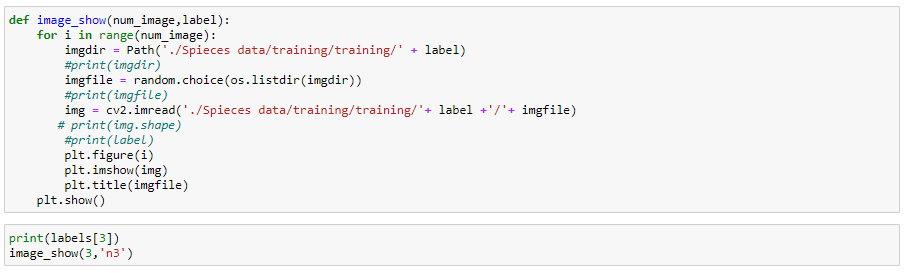
* Now read train and the test files.
* From labels.txt, add the label names to the column.

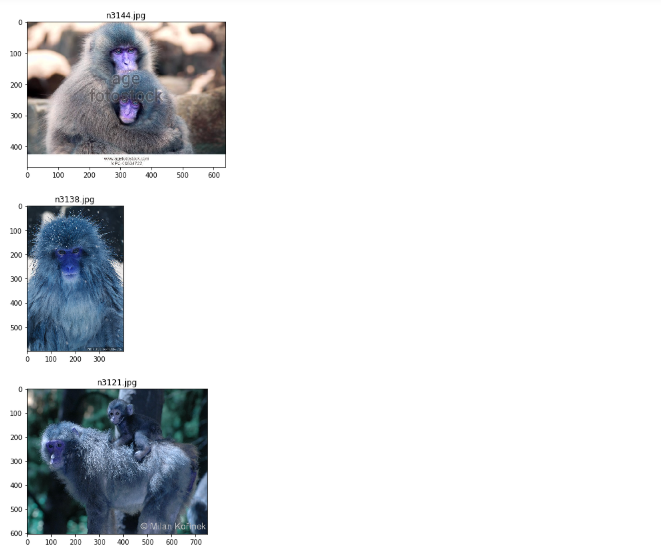


* Print the content in 'common name' column to check.



* Now Print the images with the different labels.

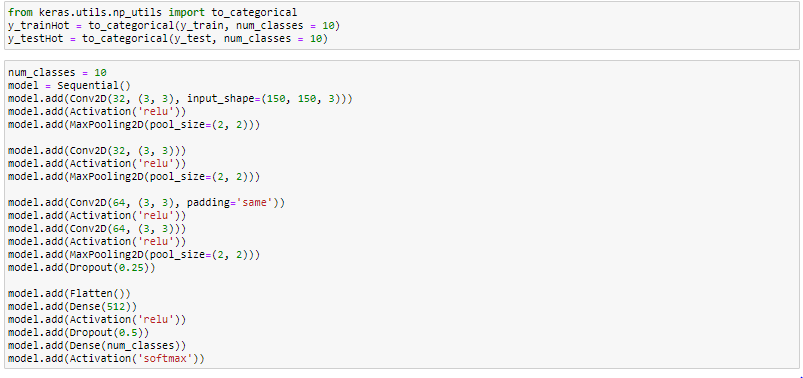




* Import tqdm and then append all the foldernames to read the image data and store in array.

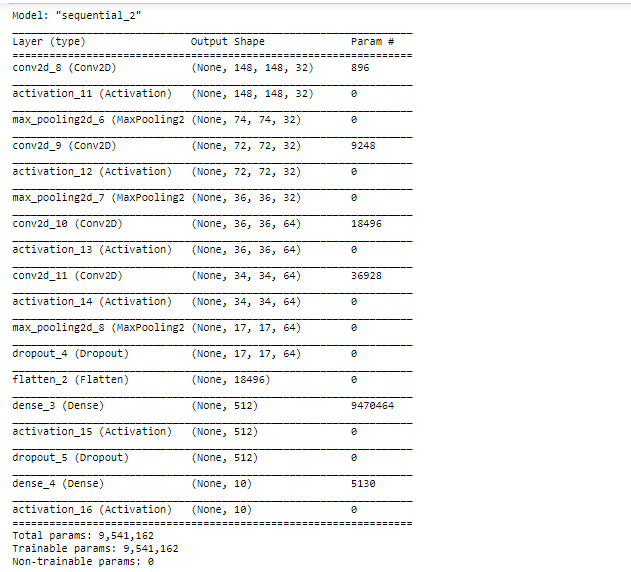


* Using hot encoder, convert categorical columns to numerical columns.
* Initialize the sequential model.

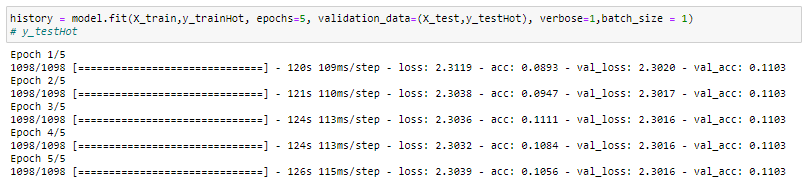


* Compile the model and get the summary.





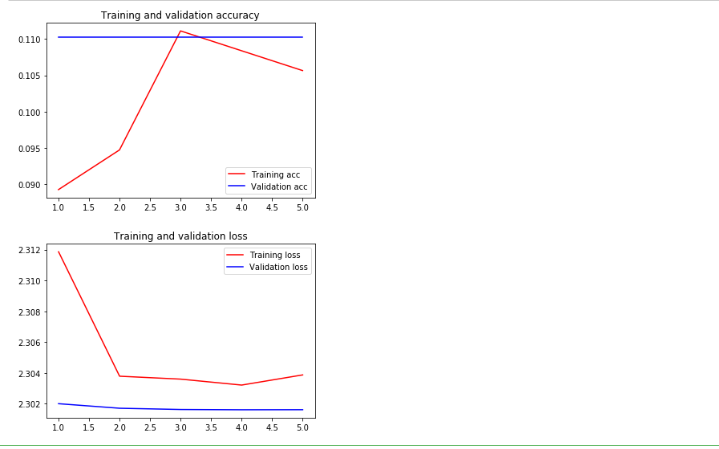
* Plot the history object and predict the accuracy and loss.



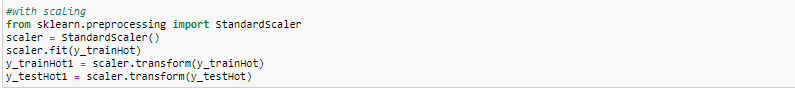


* Now plot the loss and accuracy.

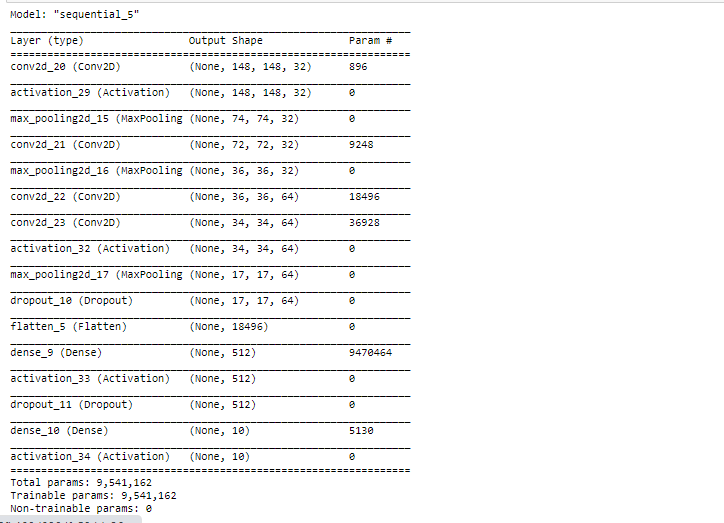




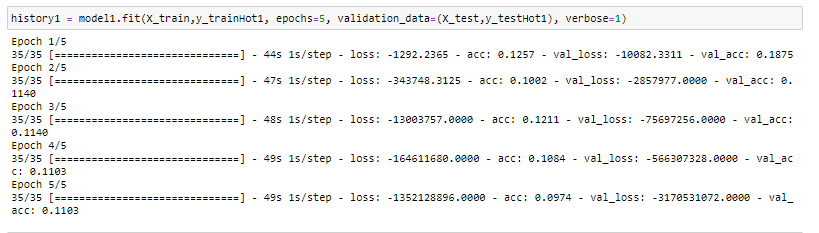
* Fit it to the standard scaler and then initialize the model.



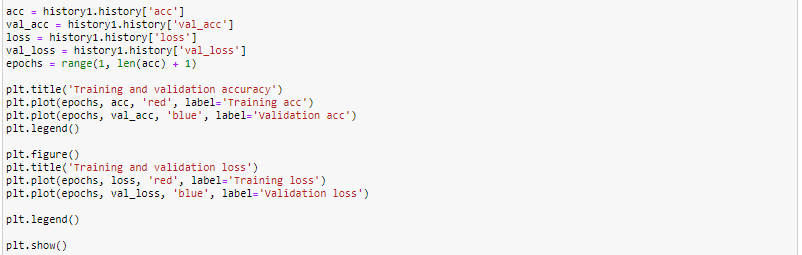


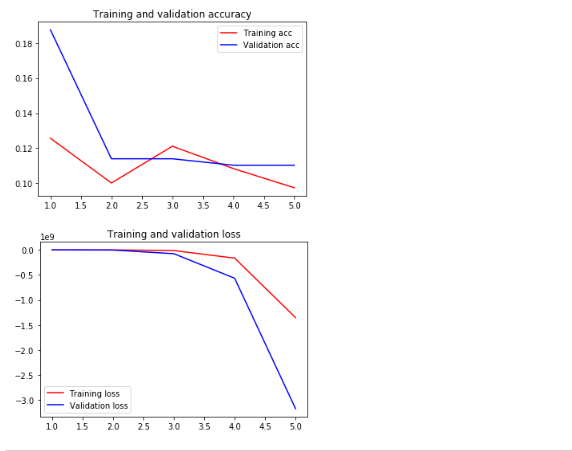


\* plot the history object and predict the loss and accuracy.



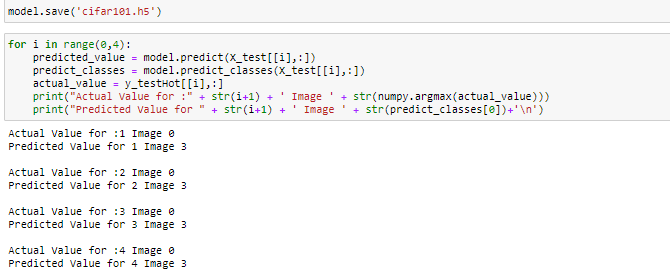
* Plot the loss and accuracy.





**b. Save the model and then predict on one of the test data. Report the prediction and check if it has been predicted correctly or not.**

* Now save the model and print the actual and predicted values for images.)

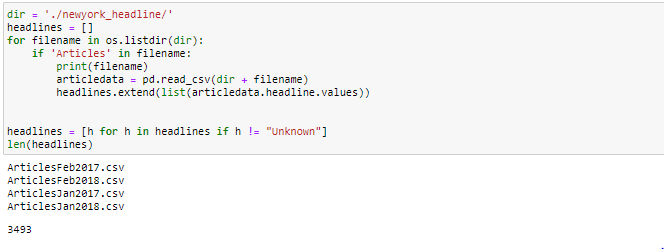


**Question 4: The purpose of this question is to learn about text generation. Use New York Times Comments and Headlines to train a text generation language model which can be used to generate News Headlines.**

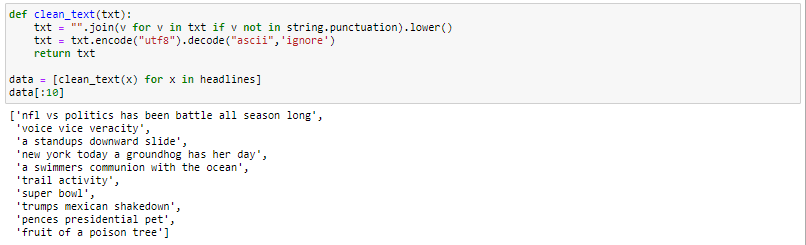
* Import all the necessary libraries.



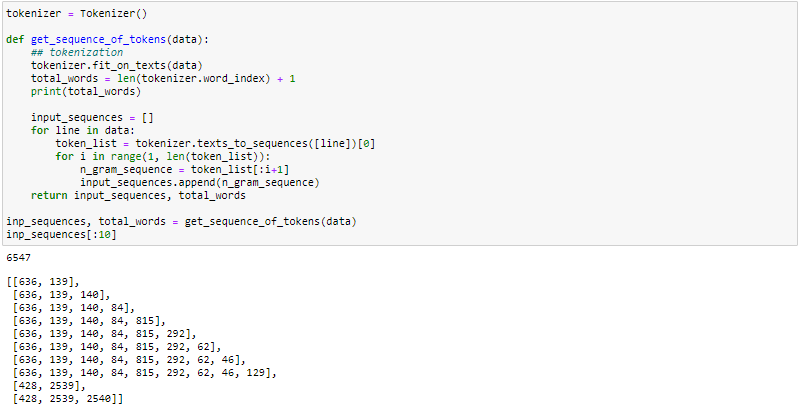
* Read the files.



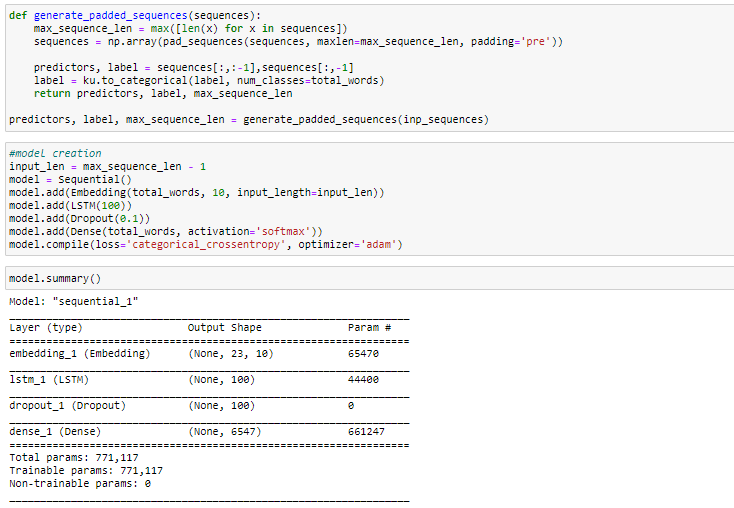
* Obtain train and test data.
* Normalize the text data.
* Print ten headlines.

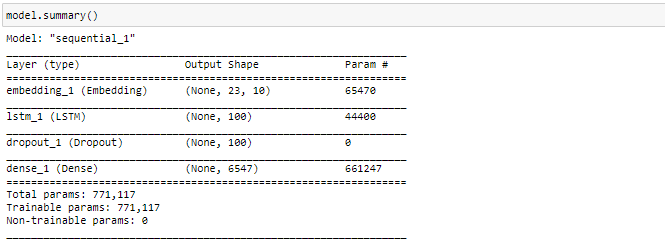


* Initialize tokenizer.
* Input sequences will be created using the list of tokens.
* Obtain sequence of tokens and print ten of them.



* Will be creating predictors and labels.
* Padding the sequences and obtaining the predictors and targets.
* Create a model with input sequence length as one less than maximum sequence length.
* Initialize LSTM model.
* As we have multiple classes initialize softmax activation layer.
* Then obtain model summary.

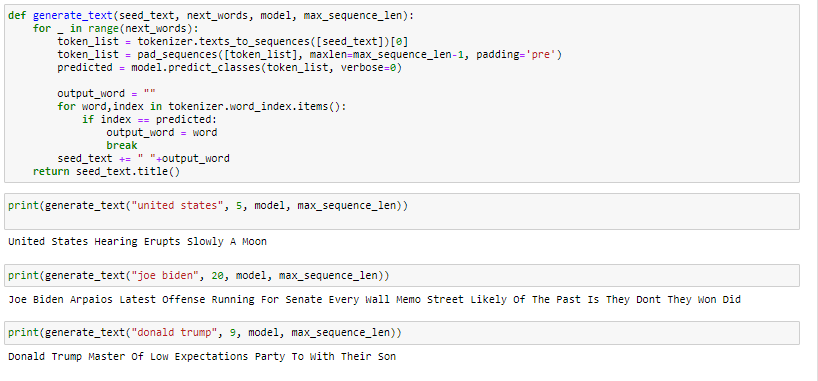




* Now fit the model on training data.
* Return the history object which hold the values of loss and accuracy metrics values.

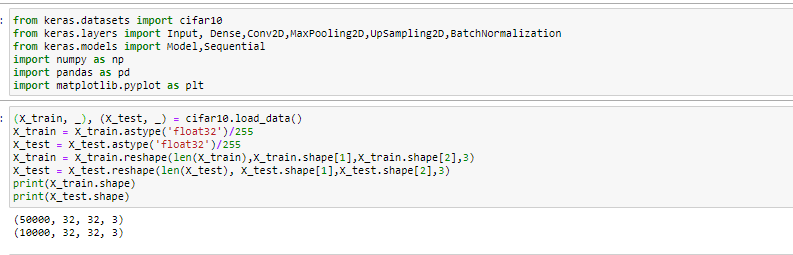


* Create a model and train the model which can generate sequence of next words.
* Consider this model and predict the token list.



**Question 5:** Apply Autoencoder on theCifar\_10datasetand then pass the result of Autoencoder to CNN or LSTM or three layers model to classify data.

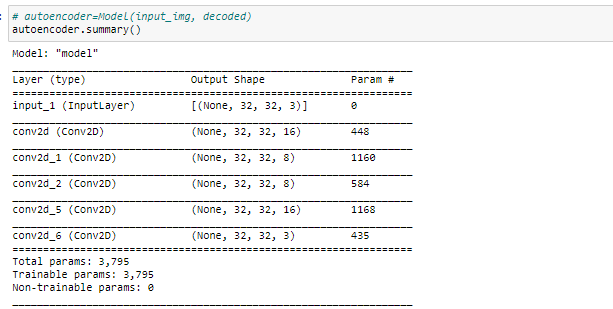
* First import all the necessary libraries.
* Normalize train and test data and then reshape.



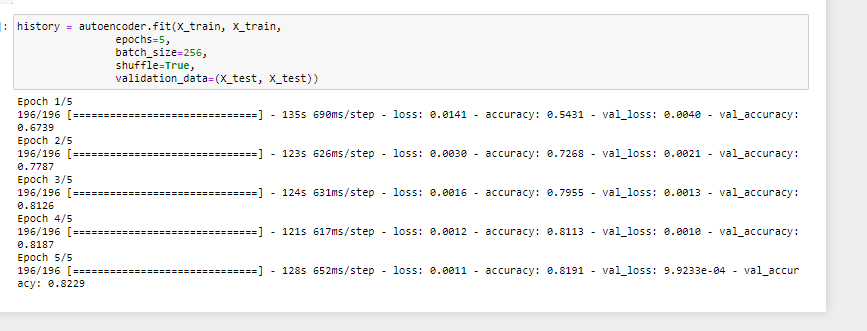
* Add the encoding layer with padding same.
* Also add the decoding layer with activation sigmoid.



* Now compile the autoencoder and print the summary.



* Plot the history object and then predict the loss and accuracy.

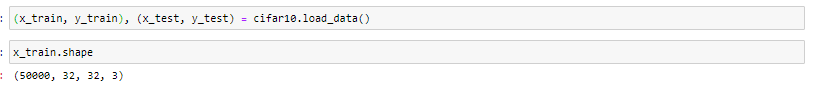


**a) Repeat the same thing with PCA (apply PCA on the dataset and then pass the result to CNN or LSTM or three layers model)**

* Import the necessary libraries.

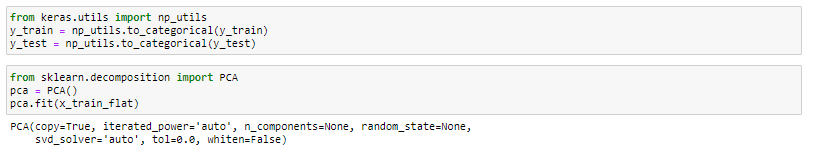


* Read the train and the test data.
* Reshape the train and the test data into two dimensional to fit into PCA.



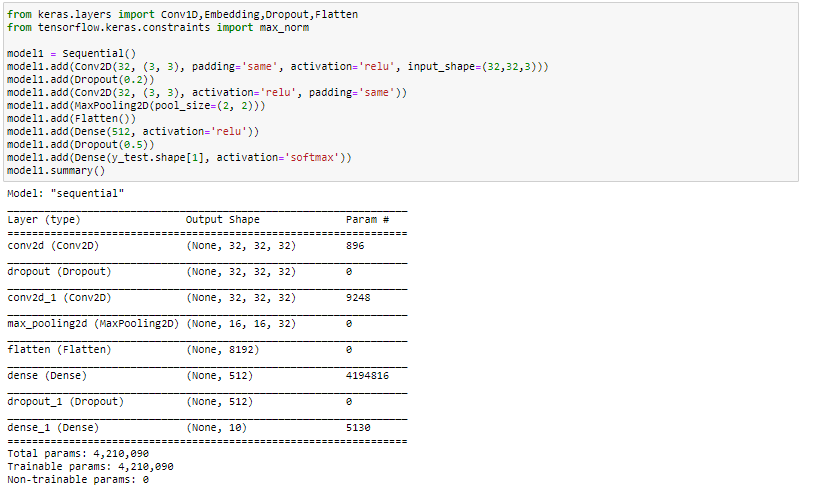


* Now import PCA and then fit x\_train\_flat and x\_test\_flat into PCA

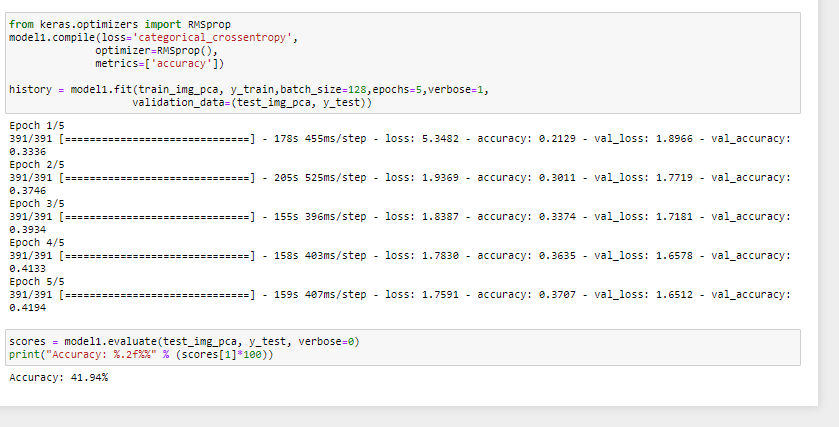




* Initialize the sequential model with the convolutional layers.



* Compile the model
* Plot the history object and then predict the loss and accuracy.



**DATASETS:**

**PARAMETERS:**

Target columns supposed in each question are our considered parameters.

Question 1: Column ‘Phrase’ is considered parameter.

Question 2:

Question 3: Column ‘Label’ is considered parameter.

Question 4: ‘Headlines’ is our considered parameter.

Question 5:

**EVALUATION AND DISCUSSION:**

Question 1:

1. Applied CNN algorithm and text image classification for implementation.
2. Obtained accuracy for this model is 64.68%.
3. After evaluating the model, we can infer that it has an overfitting problem. To resolve this issue, add hidden embedded layers with which metrics value got raised. Final accuracy obtained is 86.37%.

Question 2:

Question 3:

1. Applied image classification with CNN algorithm.
2. With scaling loss increased and accuracy decreased.
3. Without scaling loss decreased and accuracy increased.

Question 4:

1. Applied LSTM model and Text generation for implementation.
2. Used tokenizer to generate input text data sequence.
3. Initialized softmax activation layer as we have multiclass function.

Question 5:

1. Applied CNN algorithm and Auto-encoder in the implementation part.
2. Applied Principal Component Analysis.
3. Using Auto-Encoder loss decreased and accuracy increased, meanwhile using PCA made difference in metrics values, loss has increased comparatively.

**CONCLUSION:**

This project helped us learning and implementing Deep Learning concepts.

**TEAM MEMBERS:**

ARIKATLA, PALLAVI – 4

GENTE, HARUN SAI KUMAR – 15

KOTA, SARIKA REDDY – 25

MAROJU, JAGADEESH – 28