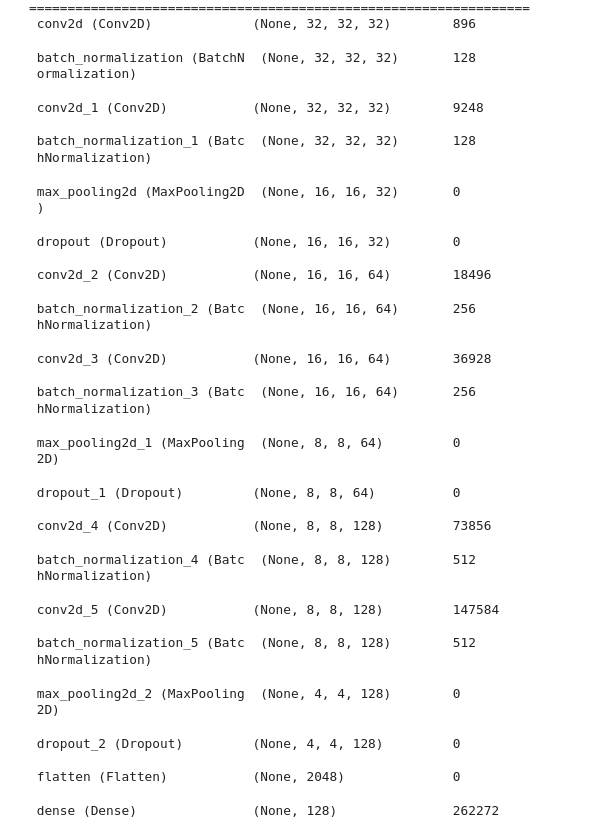
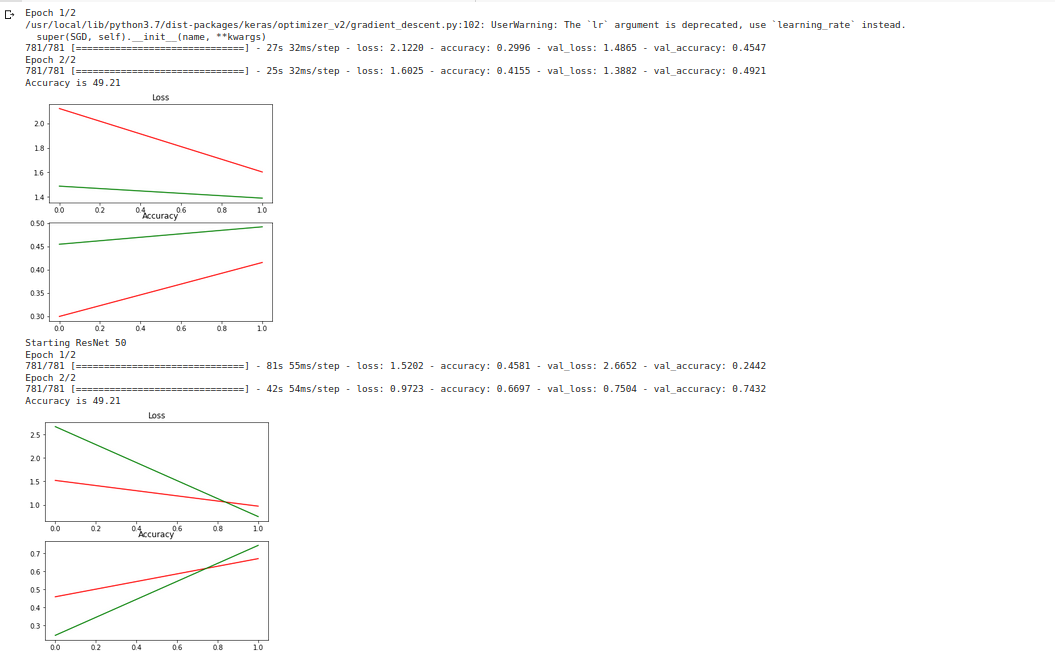
**Report**

**Task - I**

The first task was about building a CNN model and comparing it with one of the models with help of transfer learning so we start the task by loading the cifar 10 datasets with Keras into our notebook after reading we have to apply some preprocessing to the data which mean we need to convert our all number to float and then we normalize all number between zero and one for removing noise from out dataset after preprocessing it time to build the model and we mostly used Keras and TensorFlow library for building a sequential model and then providing some other parameter like the optimizer of our choice the loss by which we will compute our change in gradient and many other each layer are fully connected with it next and previous layer so the other term which we refer this to is fully connected convolutional neural network as each layer is connected in a linear manner to it next layer. The summary of our model is shown below as you can see we have used several convolutional layers along with dropout, normalization, and dense layer since cifar-10 has 10 possible outputs so the last dense layer should have 10 outputs one for each class. After building model we have to pass our dataset.

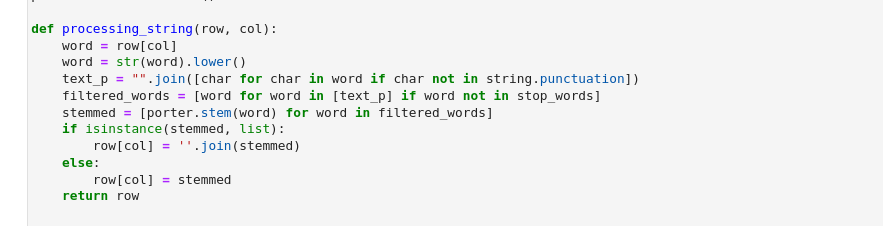
Our dataset can be first passed through some data argumentation where it will be an argument with some common image processing like rotation, shifting, and many more so that we will have more images to train and more the data the better our model will be in predicting unless it overfits the dataset. So after applying argumentation we will pass it to the model in two ways firstly we will see what happen when we pass it through our manual network then we used the same dataset out model will change we will use famous CNN model named Resnet50 and it can be easily used with help of transfer learning so we check our result in two models and compare there result. The result is pasted below and since I don’t have a very heavy machine I used it very less number of epochs we can try it for larger epochs because it can be seen with epochs our accuracy is increasing and loss is getting less so we can in that direction for a future experiment.

So our conclusion here can be using a pre-built model like ResNet50 show way better accuracy as compared to our model so we need to add more layers and as we can see both the model after each iteration show an increase in accuracy so this I also an important insight which mean if we continue our epochs model will learn better also as you can there is not very much difference b/w training and testing dataset so these mean our model is neither overfitting nor underfitting



**Task - II**

The second task is much related to a natural language problem where we have given a dataset and we have to process it and at the end, we need to add a machine learning model which will simply detect the label or output. After looking into the given dataset we came to the conclusion that it is a classification problem and we need to preprocess our sentences since they contain a lot of stop words and many noises or non-useful information which can be simply ignored so we start by reading the CSV file into a notebook with help of pandas and then we apply common string preprocessing like converting into to lowercase removing of stop words then removing punctuation and at last applying Porter Stemmer which stems our words for better understanding.



After processing each row it time we need to convert it into a number because at last, our model can only understand the number it is enabled to identifies the string so we need to find some word mapping or way to convert our string to words so we use famous algorithm know as Term Frequency-Inverse Document Frequency which takes some parameter and then each word into a one-hot encoding and also we apply two thresholds which as remove any words which are occurring less than 5 items in our whole dataset because that will not provide much information and also remove any words which have frequency rater greater than 80% because they are repeating to form so can be classified as stop words for this problem.

The final steps here will be to pass it to the machine learning model and we choose the two most used models Random Forest Classifier to check our accuracy, and the reason I choose random forest because as we know we have high dimension dataset because of one hot encoding so there are chances of overfitting also it a tree-based algorithm which works on decision making so it is quite simple for it to predict the label as it has to simply check if any particular word is present in the sentences or not. We can also try with different classification problems like support vector machine, decision tree classifier

To check which model is best as compared to others.As you can see we are achieving 98% accuracy with random forest that is quite good in our first trial.

