DEEP LEARNING PROJECT REPORT

BY

SUBHAJIT KARMAKAR

(111801041)

(Team 1)

COURSE NAME:-

Deep Learning

INTRODUCTION

Our deep learning project is made up of three tasks which we need to perform using the Pascal 50S dataset. In this dataset we have 1000 images, each image having 50 descriptions. Each image belongs to one of the 20 classes (person, bird, cat, cow, dog, horse, sheep, aeroplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa, tv/monitor).

In the first task our aim is to use the descriptions for each image and use some scraping techniques to extract labels for each image. Then we are required to build a deep convolutional neural network model to classify the images into its labels.

In the second task our aim is to do a multi-modal image classification using text and image data.

In the third task our aim is to design a model using CNN, RNN and FNN which predicts a description for each image.

TASK 1

Problem Statement

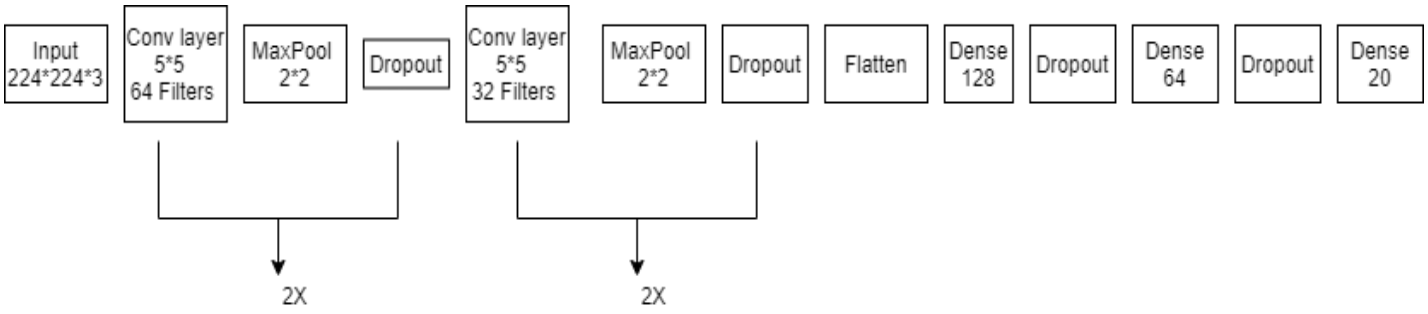
We are given PASCAL 50S dataset consisting of 1000 images each belonging to one of the 20 classes and having 50 descriptions. Our aim is to find the labels of images by scraping and then use the scraped labels and images to perform multi label image classification.

Data Preprocessing

- We extracted the image links and descriptions from the Pascal50s.mat file.
- Obtained images from the links using requests and image library functions.
- Resized each dimension of each image to a fixed size
- Built a dictionary with the 20 labels as the keys. Each label has a list of synonyms, obtained from scraping WordNet
- Converted all images descriptions to lower case, removed punctuations, numbers
- Lemmatized each word using WordNetLemmatizer function
- Scanned through each word of the description and checked whether that is a synonym of any label, if so, we set as 1 index for that corresponding label in an array of size 20.

Model Architecture

- Our model takes in input of shape 224*224*3
- Then there are two sets of Convolutional Layer of size(5,5), no of filters as 64 and with relu activation function, max pool layer of size(2,2) and dropout layer.
- Then there are 2 sets of Convolutional Layer of size(5,5), no of filters are 32, and with relu activation function, Max Pool layer of size(2,2) and dropout layer
- These convolutional layers are followed by a flatten layer, a Dense layer of size 128 with relu activation, a dropout layer, a Dense Layer of size 64 with relu activation and a dropout layer.
- Finally there is a dropout layer of size 20 with sigmoid activation function.
- This sigmoid layer gives a value of 1 for the nodes corresponding to those labels which are present in that image.



Hyperparameters Selection:-

In task 1 we tried various combinations of different kinds of Hyperparameters. The hyperparameters which we decided to vary were the following:-

- The number of blocks of convolutional layers,max pooling and dropout layers.
- The number of filters in convolutional layers, the kernel size in convolutional layers and the activation function in the activation layer.
- The max pool size in max pooling layer
- The optimizers and the loss function
- The number of epochs for which we trained and the batch size.

Results from varying the hyperparameters:-

At first we tried to change the architecture of our deep convolutional neural network model by varying the number of blocks of convolutional layer, max pooling layer and dropout layer. We tried by putting a higher number of blocks like 6,8,12,16 and observed that the validation accuracy decreased drastically, which implied that the model was overfitting. On decreasing the number of blocks to lower numbers like 2,3, we found that the loss train accuracy was really low which implied due to less number of weights, the model was unable to learn properly. So at the end we decided to stick with 4 numbers of blocks of Convolutional, max pooling and dropout layer.

We tried a varied combination of filters in convolutional layers which varied like 256,128,64,32,16,8. We also changed the kernel

size by trying out values like (6,6),(5,5),(4,4),(3,3),(2,2). We tried the activation functions like ReLU, hypertan, etc.

In the max pooling layer we tried a few combinations like (4,4),(3,3),(2,2) and observed varying accuracies.

In optimiser we tried Adam, Stochastic Gradient Descent,etc. and in loss function we tried binary_crossentropy and categorical_crossentropy.

We tried training various numbers of epochs, and observed that a higher number of epochs shows overfitting and a lower number of epochs shows underfitting.

Finally after various combinations of hyperparameters and many trial and errors we arrived at our model which we used for training our train data.

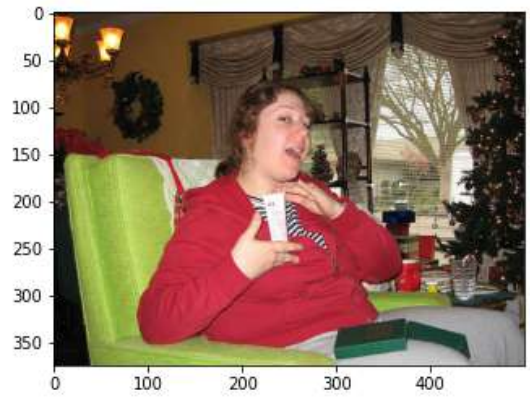
For training we chose the training accuracy, precision and recall as the metrics for measuring the performance of our model.

Result:-

- We splitted our model into an 8:2 train validation split.
- We trained our model on the train set which gave us an accuracy of about 45%.
- Precision is 0.45 and recall is 0.22.

Output:-

On using this model to predict the labels present in few images. The following results were obtained.




```
The predicted labels >>  person, dining table,
The actual labels >>  person, bottle, chair, potted plant, sofa,
```


```
The predicted labels >>  person, dining table,
The actual labels >>  person, dining table,
```


```
The predicted labels >>  person, dining table,
The actual labels >>  person, boat,
```

TASK 2

Problem Statement

We need to use image data and text data and perform multi-modal classification of the images present in PASCAL 50S dataset into the 20 classes.

Data Preprocessing

- Converted all image descriptions to lower case, removed punctuations, numbers.
- Lemmatized each word using WordNetLemmatizer function
- Scanned through a few of the descriptions and used Tokenizer() function to convert the texts to encoded integers, used pad_sequences() function to make the encoded array of fixed size.

Text feature extraction

- We trained a simple RNN model on the encoded descriptions and label data.
- We removed the last layer of this model and used this new model to extract features from descriptions.
- This model helps us extract features from text data and gives us an array of features of length 100 for each image.

Architecture of the model to extract text features

Here in the first layer the size is 44 because we have set 44 as the max length of the input string.

The dense layer is of size 20 because we have 20 labels in our dataset.

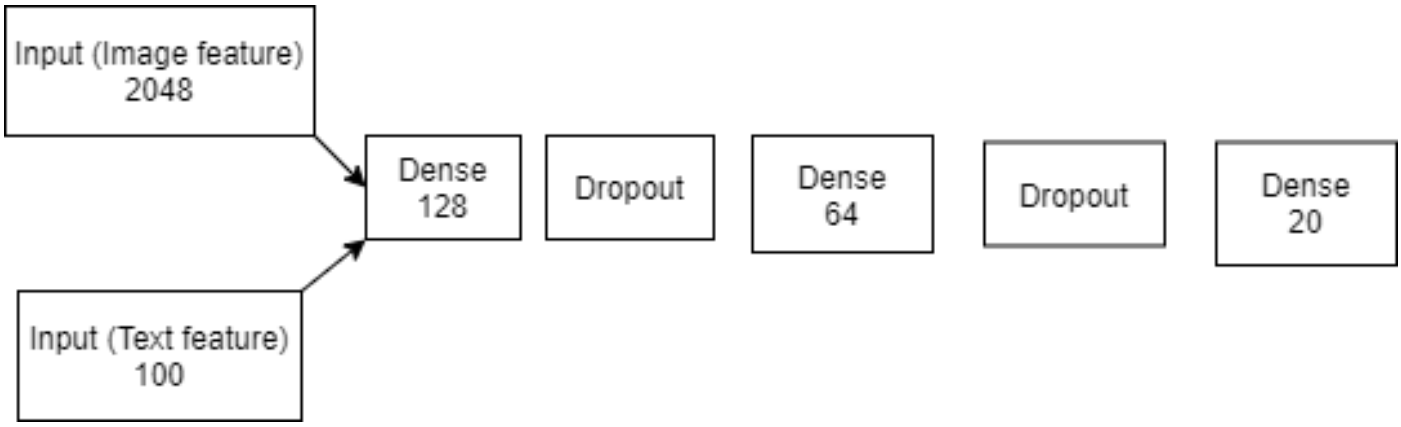
Image Features Extraction

- We used a pretrained model called ResNet .
- We removed the last layer of the ResNet model and passed our image into it, the second last layer returned image features.
- This model helps us extract features from image data and gives us an array of features of length 2048 for each image.

Model combining text features and image features

- For each image we combine the text features and image features obtained from the previous models.
- We passed these combined representations of text and image through a FNN having a dense layer of size 128 with relu activation function, then it has a dropout layer followed by a dense layer of size 64, then it has a dropout layer.
- The model has a final layer having length 20 and a sigmoid function
- Then according to the output of this sigmoid function we get the predicted labels.

- The architecture of the model combining image and text features are as follows.



Details of literature review

We referred to the TieNet paper for making ourselves familiar with the idea of using multimodal image classification. We learnt how the TieNet paper aims to combine the Chest XRay image details and the XRay report and diagnose the disease. It is a particularly interesting methodology as it aims to combine different sources of information and use the various details provided by different sources of data to diagnose the disease. In this paper we found that the text data is converted into text embedding and the image data is passed through pretrained ResNet-50 model which fetches the high label features of the image and then both the different sources of information are passed through FNN to diagnose the disease.

Similarly, we also converted our text data from image descriptions to text embeddings and we passed our images through the ResNet-50 model to obtain the high level features of the image

and then combined two sources of data through our FNN to detect the image labels.

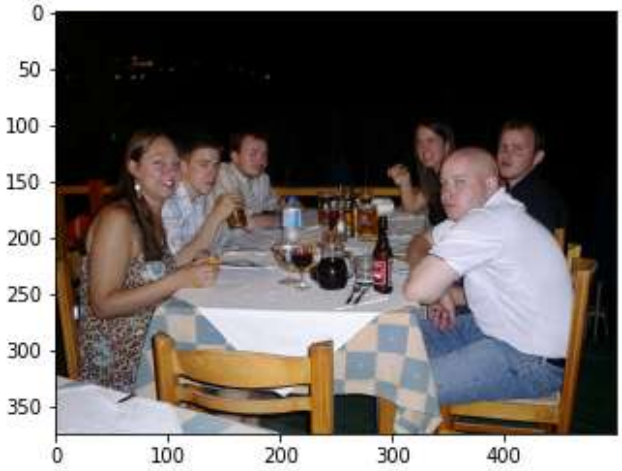
Results

- We splitted our dataset in an 8:2 train validation split.
- After training our model on 80% of image data as input and labels as target, we obtained a train accuracy of about 47%
- Precision is 0.72 and recall is 0.40

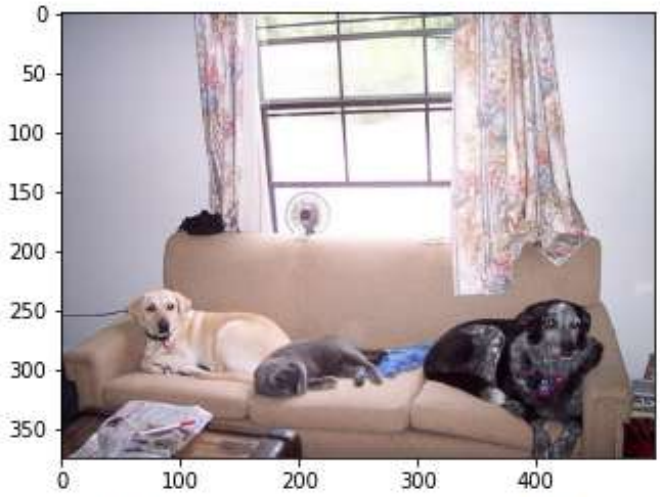
Output:-

On using this model to predict the labels present in few images. The following results were obtained.


```
The predicted labels >>  car, dining table,
The actual labels >>  bus, car,
```



```
The predicted labels >>  person, dining table,
The actual labels >>  person, dining table,
```



```
The predicted labels >>  dog, sofa,
The actual labels >>  cat, dog, sofa,
```

```
The predicted labels >>  car, potted plant,
The actual labels >>  car, potted plant,
```

TASK 3

Problem Statement

We need to build a model for generating image descriptions using a combination of RNN and CNN.

Data Preprocessing

- Converted all image descriptions to lower case, removed punctuation, numbers and added a start word to the beginning of the sentence and added a stop word to the end of the sentence.
- Lemmatized each word using WordNetLemmatizer function
- Scanned through the descriptions and used Tokenizer() function to convert the texts to encoded integers, used pad_sequences() function to make the encoded array of fixed size.

Generating Dataset for training

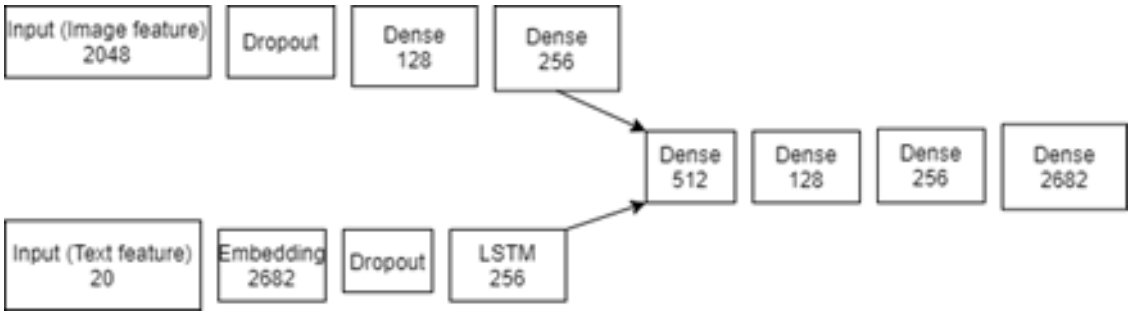
- Took a pretrained ResNet model, removed the last layer and passed each image into the model to obtain image features.
- Scanned through descriptions, converted the words into encodings, splitted each description at a different index.
- The image and the part of the description towards the left of the split is treated as input for training, and the next word after the split is treated as output.

```
X1,     X2 (text sequence),                          y (word)
photo   startseq,                                    little
photo   startseq, little,                            girl
photo   startseq, little, girl,                      running
photo   startseq, little, girl, running,             in
photo   startseq, little, girl, running, in,         field
photo   startseq, little, girl, running, in, field,  endseq
```

As shown in the above example our training features consist of the features of the photo and a portion of the description. Here at first the data point is the photo feature and the word ‘startseq’, then the next data point is the photo and the words ‘startseq’, ‘little’. At every instance the output data is the next word in the sentence before which has been included in training features.

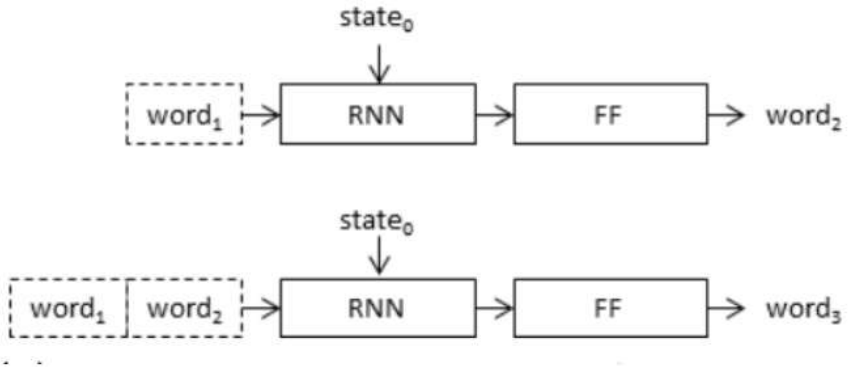
Model combining text features and image features

- For each image, we pass the extracted features through a FNN which consists of two dense layers of size 128 and 256.
- For each image, we pass the text encodings corresponding to every image through a RNN, which has an embedding layer of size 2682(the number of different words present in all of our descriptions), then it is followed by a dropout layer and a LSTM of size 256.
- Then the resultant output obtained from two models are fed into a FNN having 4 dense layers of size 512, 128, 256 each having relu activation.
- The final layer has 2682 nodes with softmax activation, which gives us a single word out of the 2682 nodes present in our descriptions.
- The architecture of the model is as follows-



Generating description

- We take an image and use the modified ResNet model to extract its features.
- Then we take an input array with only the start word
- Then we use the image feature and encoded form of input array to predict the next word.
- We append the sampled word to the end of the input array and continue it till the length of generated description is 20 or the sampled word is stop word.



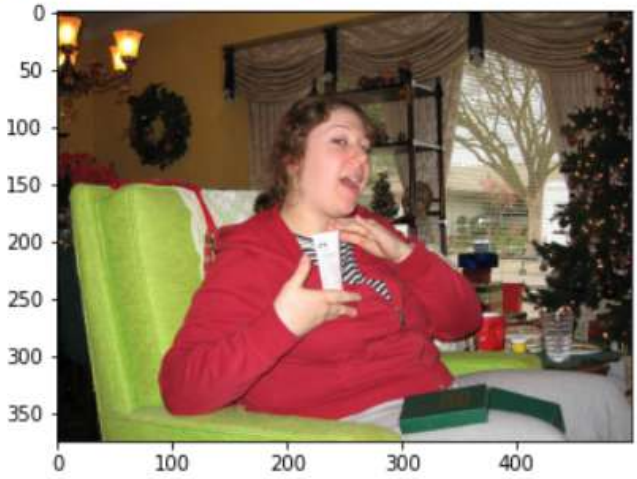
Here the state0 is the image features obtained after passing our image through the ResNet model and the word1 is a start word ‘sl’ which we appended at the end of every description. The word2 which we get as output is appended to word1 and both the words are fed into the model along with the image features state0, we obtain word3. This keeps on continuing till either our length of description reaches 20 or any word sampled is our stop word ‘el’.

Output:-

On using this model to predict the captions of a few images. The following results were obtained.

CAPTION >> a man awaits a train as a train approaches

CAPTION >> a woman is sitting on a chair



CAPTION >> a plane is preparing to takeoff

CAPTION >> a train stoped at the train station

References

- https://arxiv.org/abs/1801.04334#:~:text=In%20this%20paper%2C%20we%20show,distinctive%20image%20and%20text%20representations

Challenges Faced

- Due to the limited size of RAM provided by google Colab, the RAM was exploding frequently, we had to change our model architectures, no of epochs and dataset size at different instances to make our computations less expensive.
- At different instances in our code, we reused the dictionaries and arrays repeatedly but sometimes computing those dictionaries and arrays were really expensive. So we saved those arrays and dictionaries at intermediate stages and imported them again to reuse them.

Learning Outcomes

- We learnt a lot about word scraping, lemmatization and word preprocessing techniques.
- We learnt a lot about designing neural network models and optimizing them by trying various combinations of hyperparameters.
- We learnt about doing multi-modal image classifications.
- We learnt about converting text data to vector form using various techniques.
- We learnt about manipulating pre trained models like VGG16 and ResNet and using the modified models to extract high level features of the image.

Acknowledgement:-

I would like to express my deep sense of gratitude towards my teammate Vishal Rao for his continuous cooperation and support right throughout the entire duration of the project. His sincere efforts and the team spirit were imperative to the successful completion of the project.

I would like to thank our mentor Neha A S for her valuable guidance and feedback throughout the project which helped us improve our work and ultimately led to successful completion of the project.

I would like to extend my sincere gratitude to Dr. Mrinal Kanti Das, our course instructor, for giving us this wonderful opportunity to do this project on such an interesting topic. It helped us apply the knowledge gained in class to build a real life application and learn so many new things.

I also would like to acknowledge the efforts of my friends who have helped me to understand various concepts which led to successful completion of this project.