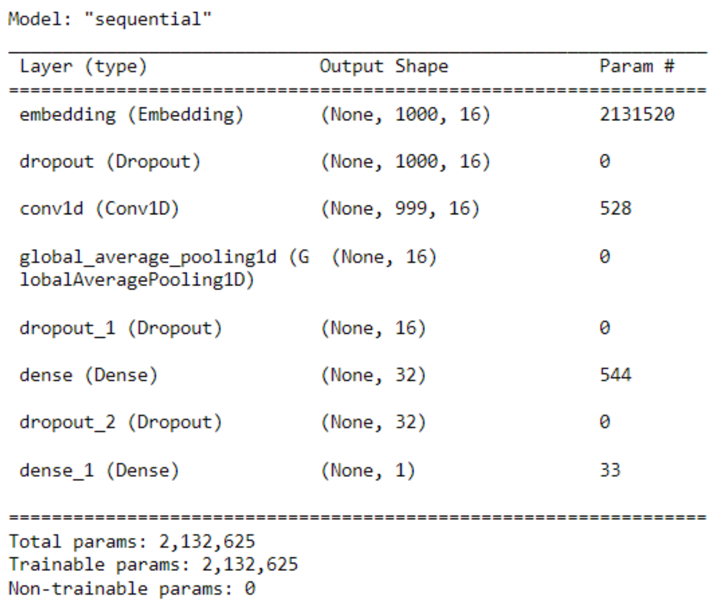
Problem 3 – Sentiment Analysis

Data Preprocessing:

The dataset was downloaded as 'aclImdb' directory which consists of train and test directories separately. Using glob library, each of the text files were read and its data was stored into a list of all training and test reviews. The labels were stored in a separate list with 1's for positive class and 0 for negative class. After this, we performed a cleanup of the texts to remove all punctuations, html tags, extra whitespace and then converted the whole text to lowercase. Then the whole textual data was converted into words using regular expressions tokenizer from NLTK and stopwords were removed. We have applied lemmatization to convert words into their base forms using WordNetLemmatizer from NLTK library. Following that, we fit a text tokenizer from the Keras library onto the training set which first tokenizes the text and assigns a unique integer to each word in the vocabulary and then converts the text into a sequence of integers based on the learned vocabulary. The average length of all reviews at this point was 121 words and the maximum length was 1429. Since we need the input length to be the same for all reviews, the shorter sequences were padded with zeros at the end and longer sequences were trimmed to a maximum of 1000. After executing all the above mentioned steps, we split the set into 80% training set and 20% validation set.

Network Architecture:

The following Figure 1 shows the network design chosen, which is a sequential CNN model.

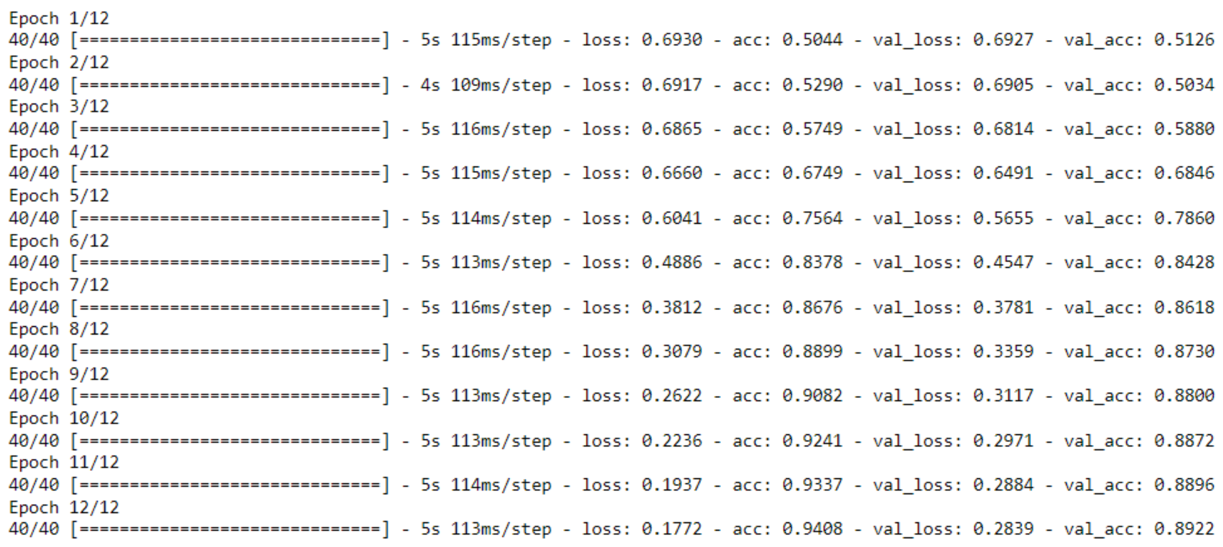


```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 1000, 16)          2131520

 dropout (Dropout)           (None, 1000, 16)          0

 conv1d (Conv1D)             (None, 999, 16)           528

 global_average_pooling1d (G  (None, 16)               0
 lobalAveragePooling1D)

 dropout_1 (Dropout)         (None, 16)                0

 dense (Dense)               (None, 32)                544

 dropout_2 (Dropout)         (None, 32)                0

 dense_1 (Dense)             (None, 1)                 33

=================================================================
Total params: 2,132,625
Trainable params: 2,132,625
Non-trainable params: 0
```

Figure 1: CNN structure

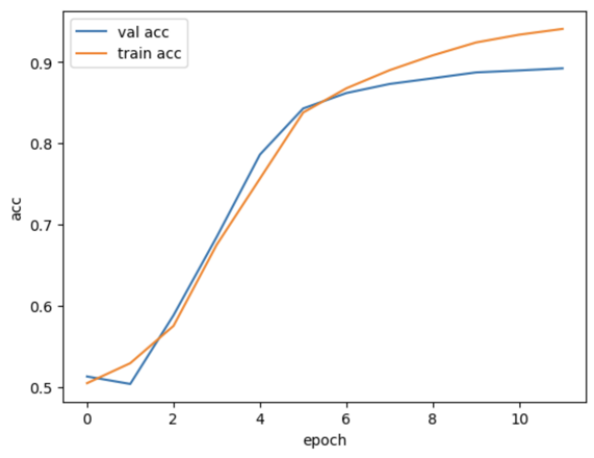
First layer is an Embedding layer which is used to convert integer sequences into vectors of fixed size. After that, there is a one-dimensional convolutional layer with 16 filters, kernel size of 2, no padding and ReLU activation function. Average pooling layer of one dimension has been added that reduces spatial dimensions of data. Dense adds a fully connected layer with 32 units and tanh activation function. The final layer has 1 unit and sigmoid activation function since this is a binary classification problem. Dropout layers have been added in between operations to regularize the model and avoid overfitting.

Training and Testing Accuracies:

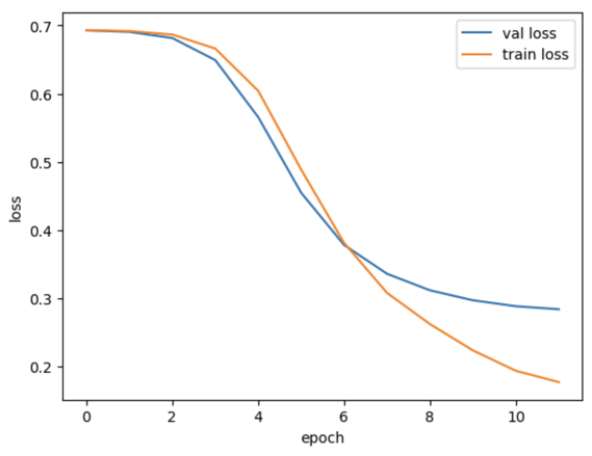


```
Epoch 1/12
40/40 [==============================] - 5s 115ms/step - loss: 0.6930 - acc: 0.5044 - val_loss: 0.6927 - val_acc: 0.5126
Epoch 2/12
40/40 [==============================] - 4s 109ms/step - loss: 0.6917 - acc: 0.5290 - val_loss: 0.6905 - val_acc: 0.5034
Epoch 3/12
40/40 [==============================] - 5s 116ms/step - loss: 0.6865 - acc: 0.5749 - val_loss: 0.6814 - val_acc: 0.5880
Epoch 4/12
40/40 [==============================] - 5s 115ms/step - loss: 0.6660 - acc: 0.6749 - val_loss: 0.6491 - val_acc: 0.6846
Epoch 5/12
40/40 [==============================] - 5s 114ms/step - loss: 0.6041 - acc: 0.7564 - val_loss: 0.5655 - val_acc: 0.7860
Epoch 6/12
40/40 [==============================] - 5s 113ms/step - loss: 0.4886 - acc: 0.8378 - val_loss: 0.4547 - val_acc: 0.8428
Epoch 7/12
40/40 [==============================] - 5s 116ms/step - loss: 0.3812 - acc: 0.8676 - val_loss: 0.3781 - val_acc: 0.8618
Epoch 8/12
40/40 [==============================] - 5s 116ms/step - loss: 0.3079 - acc: 0.8899 - val_loss: 0.3359 - val_acc: 0.8730
Epoch 9/12
40/40 [==============================] - 5s 113ms/step - loss: 0.2622 - acc: 0.9082 - val_loss: 0.3117 - val_acc: 0.8800
Epoch 10/12
40/40 [==============================] - 5s 113ms/step - loss: 0.2236 - acc: 0.9241 - val_loss: 0.2971 - val_acc: 0.8872
Epoch 11/12
40/40 [==============================] - 5s 114ms/step - loss: 0.1937 - acc: 0.9337 - val_loss: 0.2884 - val_acc: 0.8896
Epoch 12/12
40/40 [==============================] - 5s 113ms/step - loss: 0.1772 - acc: 0.9408 - val_loss: 0.2839 - val_acc: 0.8922
```

Figure 2: Results per epoch



(a)



(b)

Figure 3: (a) Training vs Validation set accuracy comparison, (b) Training vs validation set loss comparison

After training the model for 25 epochs, it was noticed that the model was overfitting beyond 10-12 epochs and thus, the final model has been trained over 12 epochs with batch size of 512. As per the figures above, the training loss is 0.1772, validation loss is 0.2839, training accuracy reaches 94.08% and validation set accuracy reaches 89.22%. The test set accuracy is 88.136%.