Comparison of Deep Learning Approaches for IMDB Sentiment Classification

Written by Preeti Singh (1002013566), Sai Sharath Reddy Koppula (1002081785), Renu Aakanksha Veesam (1002113666)

[1]Department of Computer Science
University of Texas at Arlington, TX, USA
pxs3566@mavs.uta.edu, sxk1785@mavs.uta.edu, rxv3666@mavs.uta.edu

Abstract

The key to comprehending and measuring public opinion is sentiment analysis, which gives companies and organizations important insights into the opinions and preferences of their customers. The public expresses their ideas on a variety of internet venues, including social media and reviews. Organizations can make well-informed decisions to improve customer satisfaction and their products or services by studying reviews that are posted on various platforms. Mining movie reviews is equally important as mining reviews information for a business as movie reviews offer valuable information to the makers on the aspects that the audience are liking and the aspects where one should improvise in order to cater the audience a better movie watching experience. In this research paper we developed four different Deep Learning model architectures based on Long-Short Term Memory units, Convolutional Neural Networks and Transformer based BERT architecture to inspect which architecture is offering best classification accuracy on unseen IMDB movie reviews dataset. Our experiments reveal that LSTM-CNN based model is offering better classification accuracy within limited epochs of training the model.

Introduction

The key to comprehending and measuring public opinion is sentiment analysis, which gives companies and organizations important insights into the opinions and preferences of their customers (1). The public expresses their ideas on a variety of internet venues, including social media and reviews. Organizations can make well-informed decisions to improve customer satisfaction and their products or services by studying reviews that are posted on various platforms. Mining movie reviews is equally important as mining reviews information for a business as movie reviews offer valuable information to the makers on the aspects that the audience are liking and the aspects where one should improvise in order to cater the audience a better movie watching experience.

The amount of data being generated is rapidly increasing and hence are the computational requirements to process the data to gain valuable insights. Thanks to modern day processors based on Graphical Processing Units (GPU), we are now able to process large amounts of data to gain valuable insights on the perspective of customers towards a business. This is one of the possible applications of Deep Learning based technique given the large amounts of dataset and modern day processors.

In this research paper we developed four different Deep Learning model architectures based on Long-Short Term Memory units, Convolutional Neural Networks and Transformer based BERT architecture to inspect which architecture is offering best classification accuracy on unseen IMDB movie reviews dataset. Our experiments reveal that LSTM-CNN based model is offering better classification accuracy within limited epochs of training the model. The dataset utilized in this regard is the Internet Movie Review Database (IMDB).

The next section in this research paper talk about the dataset description, model development and the analysis of the results obtained.

Dataset Description

The dataset used as part of this comparison study is the IMDB movie reviews dataset. The IMDB movie reviews dataset is publicly available for download at (2; 3; 4). This dataset is comprised of 50,000 reviews. This is a balanced dataset meaning that there are 25,000 reviews belonging to the positive class and another 25,000 reviews belonging to the negative class. Figure xx shows a count plot capturing the data distribution between the two class: positive and negative classes. From the figure 1 it can be seen that the number of samples between the two classes is same. It is interesting to note that this dataset is divided into training and test set with 25,000 reviews in each set.

The IMDB movie reviews dataset in original form may need additional preprocessing in order to apply any of the Deep Learning techniques to develop a sentiment classification model. Figure 2 captures a raw version of reviews for both the positive and negative classes.

In general, the following are the text preprocessing steps that are applied employed in preparing the movie reviews dataset:

Data Standardization

In this phase, all the review samples are converted into lowercase letters. Figures 3 and 4 capture the text data before

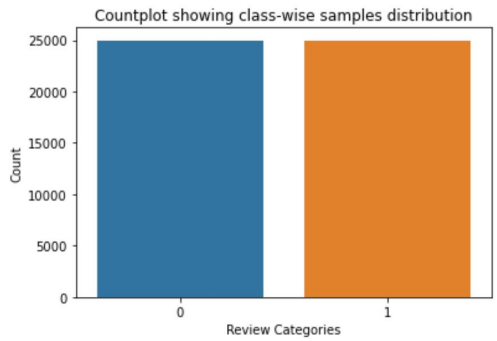


Figure 1: Reviews count for Positive and Negative Class Labels

If you like original gut wrenching laughter you will like this movie. If you are young or old then y...	positive
Phil the Alien is one of those quirky films where the humour is based around the oddness of everythi...	negative

Figure 2: Raw samples of IMDB reviews

and after performing the text data standardization.

```
df["text"].head()

0    Bromwell High is a cartoon comedy. It ran at t...
1    Homelessness (or Houselessness as George Carli...
2    Brilliant over-acting by Lesley Ann Warren. Be...
3    This is easily the most underrated film inn th...
4    This is not the typical Mel Brooks film. It wa...
Name: text, dtype: object
```

Figure 3: Date before applying Standardization Technique

Retaining Valid Characters

In this phase, all the non-alphabet characters were removed from the movie reviews dataset. Figures 5 and 6 capture the text data before and after retaining only alphabet characters.

Removing Stop words

In this phase, all the high frequency words in English language such as the, a, is, an etc., were removed from the movie reviews dataset. A list of stop words as available in the NLTK package were utilized for this purpose. The following figures capture the text data before and after removing stop words.

```
0    bromwell high is a cartoon comedy. it ran at t...
1    homelessness (or houselessness as george carli...
2    brilliant over-acting by lesley ann warren. be...
3    this is easily the most underrated film inn th...
4    this is not the typical mel brooks film. it wa...
Name: cleaned_data, dtype: object
```

Figure 4: Data after applying Standardization Technique

```
0    bromwell high is a cartoon comedy. it ran at t...
1    homelessness (or houselessness as george carli...
2    brilliant over-acting by lesley ann warren. be...
3    this is easily the most underrated film inn th...
4    this is not the typical mel brooks film. it wa...
Name: cleaned_data, dtype: object
```

Figure 5: Date before retaining valid characters only

Stemming

In English language it is common to use a word in different tenses and a standard procedure to extract the root words is called Stemming. For obtaining the root words across all the vocabulary in the IMDB movie reviews dataset, we have used the Porter Stemmer available in the NLTK package. Figures 9 and 10 capture the text data before and after applying the porter stemmer.

After processing the IMDB reviews dataset in step-by-step manner applying the above preprocessing steps, we have created a word cloud using the words retained in the positive and negative classes respectively. Word clouds are a technique to display the words in proportional to their frequency of occurrence. Figures 11 and 12 are the word clouds generated based on the positive and negative reviews dataset.

Model Development

As part of the Model Development for classifying the IMDB movie review sentiments, we have explore four different model architectures ranging across various concepts such as Long Short Term Memory (LSTM) networks, LSTM layers followed by Convolutional layers, Convolutional layers followed by the LSTM layers and lastly the BERT based architecture. We conducted hyper-parameter tuning to identify the best set of parameters and concluded on the best set of hyper-parameters for each the four models and evaluated these models to identify the best performing model in classifying the IMDB movie reviews.

In the following sections, we have outlined the details of each of these four models, hyper-parameter tuning process and the results obtained on the unseen or the test dataset. Lastly, all the models were developed in Python programming language using tensorflow package.

Long-Short Term Memory Units Based Model

This model is an implementation of the architecture proposed in the Research Paper(5). The core of this model is to utilize the LSTM units for capturing the sequential patterns in the text dataset. This model utilizes two layers of LSTM units to predict the sentiment class of a given review.

```
0    bromwell high is a cartoon comedy it ran at th...
1    homelessness or houselessness as george carlin...
2    brilliant overacting by lesley ann warren best...
3    this is easily the most underrated film inn th...
4    this is not the typical mel brooks film it was...
Name: cleaned_data, dtype: object
```

Figure 6: Date after retaining valid characters only

```
0    bromwell high is a cartoon comedy it ran at th...
1    homelessness or houselessness as george carlin...
2    brilliant overacting by lesley ann warren best...
3    this is easily the most underrated film inn th...
4    this is not the typical mel brooks film it was...
Name: cleaned_data, dtype: object
```

Figure 7: Date before removing stop words

The first layer of this model architecture utilizes LSTM units as mentioned above and a number of fifty LSTM units were utilized. Further, in the second layer hundred and one LSTM units were utilized. These hundred and one LSTM units were connected to an output or a Dense layer to obtain the sentiment value of a given review.

As mentioned earlier, this model was implemented using the tensorflow package and below is the model summary, which depicts the model architecture:

For hyper-parameter tuning, we have opted to identify the best values for batch size and embedding lengths. For each of these parameters, we have iterated over a set of values of 32 and 64. Across four combinations of batch size and embedding vector lengths, it is identified that a batch size of 64 and an embedding length of 32 were yielding best results. The following plots capture the loss values and classification accuracy obtained over the training process.

Our analysis of the results is that the model is under-learning, meaning that the training set accuracy is less than the testing set accuracy. Further, it is taking a lot of time for the model to converge. If we see even after 20 epochs the training set accuracy is still around 0.5 (as that of a random guess in a binary classification setting.

LSTM-CNN Based Model

This model is an implementation of the architecture proposed in the Research paper (6). This architecture utilizes a combination of LSTM units, which are used in general in handling text data, as it contains sequential information and 1 dimensional Convolution units, which establish the local dependency between the text sequence.

As mentioned earlier, this model was implemented using the tensorflow package and below is the model summary, which depicts the model architecture:

Similar to first model, we have conducted hyper-parameter tuning and the best combination obtained is an embedding length of 64 and a batch size of 64. With the best parameters we have re-trained the model for 10 epochs and below are training loss, training accuracy scores and the test set accuracy obtained at the end of training.

```
import nltk
nltk.download('stopwords')

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
True


stopwords = list(stopwords.words('english'))
def remove_stopwords(text):
    return " ".join([word for word in text.split() if word not in stopwords])


df["cleaned_data"] = df["cleaned_data"].apply(remove_stopwords)

print(df["cleaned_data"].head())
```

Figure 8: Date after removing stop words

```
0    bromwell high cartoon comedy ran time programs...
1    homelessness houselessness george carlin state...
2    brilliant overacting lesley ann warren best dr...
3    easily underrated film inn brooks cannon sure ...
4    typical mel brooks film much less slapstick mo...
Name: cleaned_data, dtype: object
```

Figure 9: Date before removing stop words

Unlike the first model, a combination of LSTM followed by 1D convolution network is yielding better results. It can be seen that the loss value is consistently decreasing and the accuracy values are increasing without any sudden spikes. Lastly, the training and test set accuracy values are: 0.9958 and 0.99435 respectively. Which means the model is generalizing well without much over fitting.

CNN-LSTM Based Model

The inspiration for the third model was from the research work (7). This work is similar to that of model2 except in this model we are applying the 1D convolutions first, followed by the LSTM layers. Hence the name CNN-LSTM. We have performed hyper-parameter tuning here as well and the best combination obtained was a batch size of 32 and an embedding length of 64.

As mentioned earlier, this model was implemented using the tensorflow package and below is the model summary, which depicts the model architecture:

Below are the training results of the final model with best hyper-parameter value. It can be seen that this model is over fitting, where the training accuracy is close to 80 percent at the end of the 10th epoch but the testing accuracy is 55 percent.

BERT Model

The last model that we explored was the BERT family of models. BERT family models were based on the Transformer based Encoder- Decoder architecture. We have trained in the Small-BERT model available in the tensorflow library and here is the Small-BERT model architecture (8).

Below are the BERT model results, when trained for 10 epochs. The loss and accuracy values are a lot better in the training phase compared to the model1 and model3, where there are spikes during the training phase. However, the model appears to be over fitting to the training data, we can

```
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize

ps = PorterStemmer()

def stemming(text):
  return " ".join([ps.stem(word) for word in text.split()])

df["cleaned_data"] = df["cleaned_data"].apply(stemming)

print(df['cleaned_data'].head())
```

Figure 10: Date after removing stop words

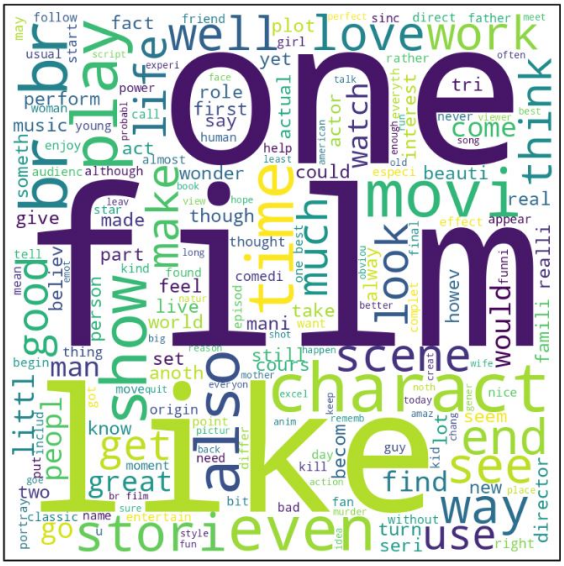


Figure 11: Word Cloud of Terms in Positive Reviews Class

see that the training accuracy at the end of 10 epochs was close to 98, however the test set accuracy was 85.

Analysis

Comparison of Four Models

Overall, we have tested four different architectures for a Binary Text Classification problem. The first model is a pure LSTM based model and the model was under fitting on the training dataset. The second model is a hybrid model based on LSTM and CNN layers. This is the best model obtained of all the four different architectures. The third model is a hybrid model too, where CNN layers are first applied followed by the application of LSTM layers. This model was over fitting on the training data. The last model we explored was BERT model, this model is performing better than the model1 and model3 but slightly over fitting on the training data. Overall, model2 is the best model and if fine-tuned BERT model may also provide on-par results with model2.

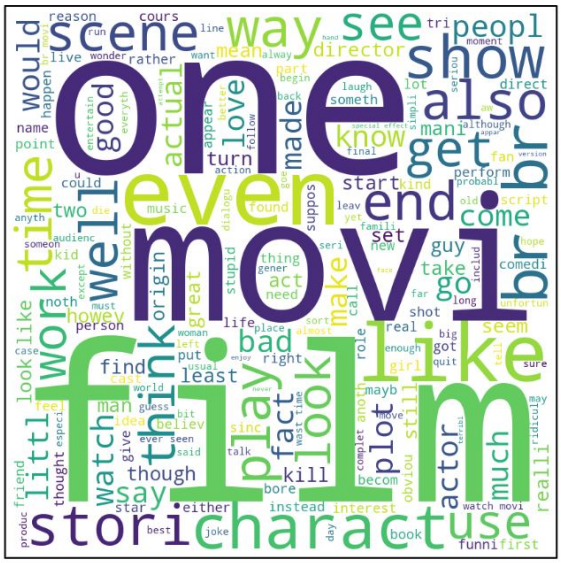


Figure 12: Word Cloud of Terms in Negative Reviews Class

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 610, 32)	2834752
lstm (LSTM)	(None, 610, 50)	16600
lstm_1 (LSTM)	(None, 101)	61408
dense (Dense)	(None, 1)	102

Total params: 2912862 (11.11 MB)
Trainable params: 2912862 (11.11 MB)
Non-trainable params: 0 (0.00 Byte)

Figure 13: LSTM Model Architecture

Conclusion

With the advancements in Computing power and Deep Learning techniques, we are increasingly analyzing the customer reviews data to gain valuable insights on to what actions to take for a better customer engagement and experience. In this work, we studied various architectures and concluded that LSTM-CNN is yielding better results on the IMDB sentiment classification dataset. In the future, we would like to understand in grate detail on what aspects on a model architecture influece the results and how LSTM-CNN is yielding better results in comparison to other models.

References

[1] https://www.techtarget.com/searchcustomerexperience/tip/Sentiment-analysis-Why-its-necessary-and-how-it-improves-CX

[2] https://ai.stanford.edu/~amaas/data/sentiment/

[3] https://www.tensorflow.org/api_docs/python/tf/keras/datasets/imdb/load_data

[4] https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

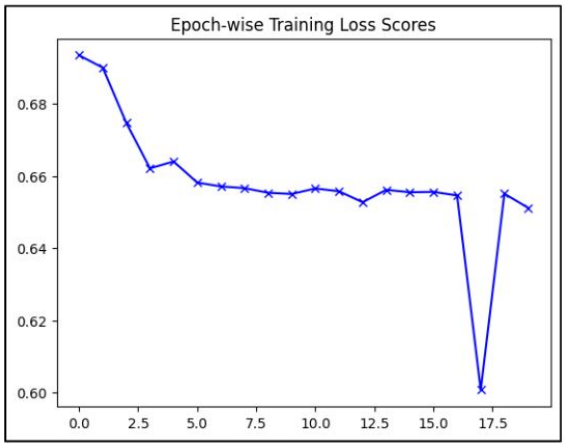


Figure 14: Model1 Training Loss Curve

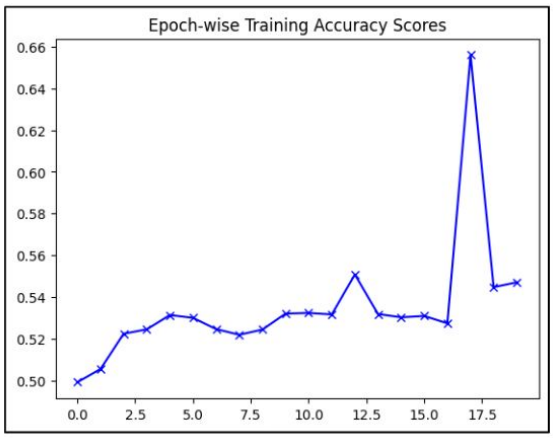


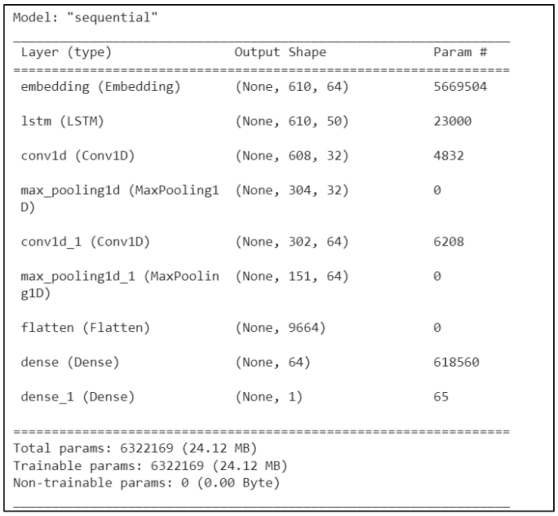
Figure 15: Model1 Training Accuracy Curve

[5] Sentiment Analysis of IMDb Movie Reviews Using Long Short-Term Memory by S. M. Qaisar.

[6] Performance Analysis of Different Neural Networks for Sentiment Analysis on IMDb Movie Reviews by M. R. Haque et. al.

[7] Sentiment Prediction of IMDb Movie Reviews Using CNN-LSTM Approach" by M. Mishra et. al.

[8] https://www.tensorflow.org/tfmodels/nlp/fine_tune_bert



Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 610, 64)	5669504
lstm (LSTM)	(None, 610, 50)	23000
conv1d (Conv1D)	(None, 608, 32)	4832
max_pooling1d (MaxPooling1D)	(None, 304, 32)	0
conv1d_1 (Conv1D)	(None, 302, 64)	6208
max_pooling1d_1 (MaxPooling1D)	(None, 151, 64)	0
flatten (Flatten)	(None, 9664)	0
dense (Dense)	(None, 64)	618560
dense_1 (Dense)	(None, 1)	65

Total params: 6322169 (24.12 MB)
Trainable params: 6322169 (24.12 MB)
Non-trainable params: 0 (0.00 Byte)

Figure 16: Model2 Architecture

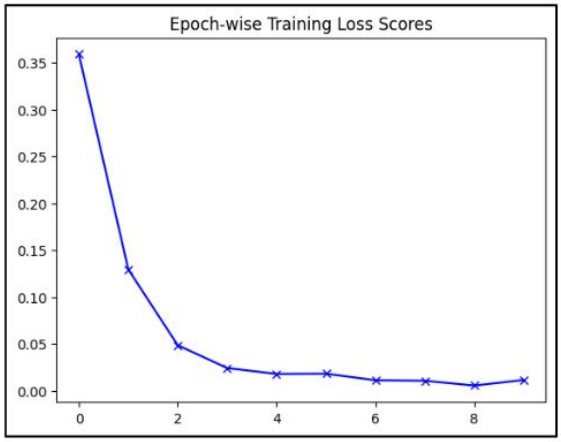


Figure 17: Model2 Training Loss Curve

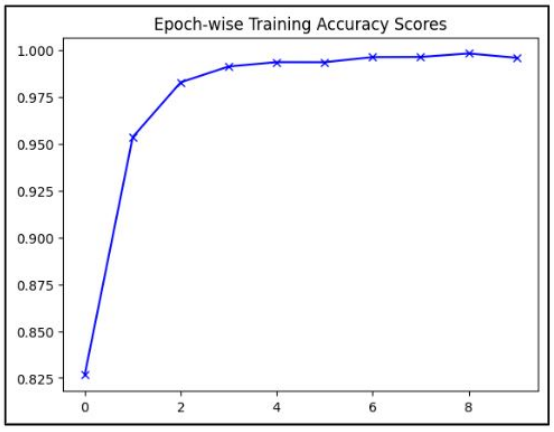


Figure 18: Model2 Training Accuracy Curve

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 610, 32)	2834752
conv1d (Conv1D)	(None, 608, 32)	3104
max_pooling1d (MaxPooling1D)	(None, 304, 32)	0
lstm (LSTM)	(None, 50)	16600
dense (Dense)	(None, 1)	51

Total params: 2854507 (10.89 MB)
Trainable params: 2854507 (10.89 MB)
Non-trainable params: 0 (0.00 Byte)

Figure 19: Model3 Architecture

Figure 20: Model3 Training Loss Curve

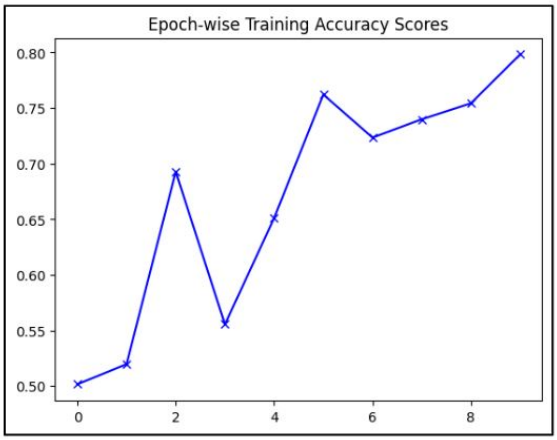


Figure 21: Model3 Training Accuracy Curve

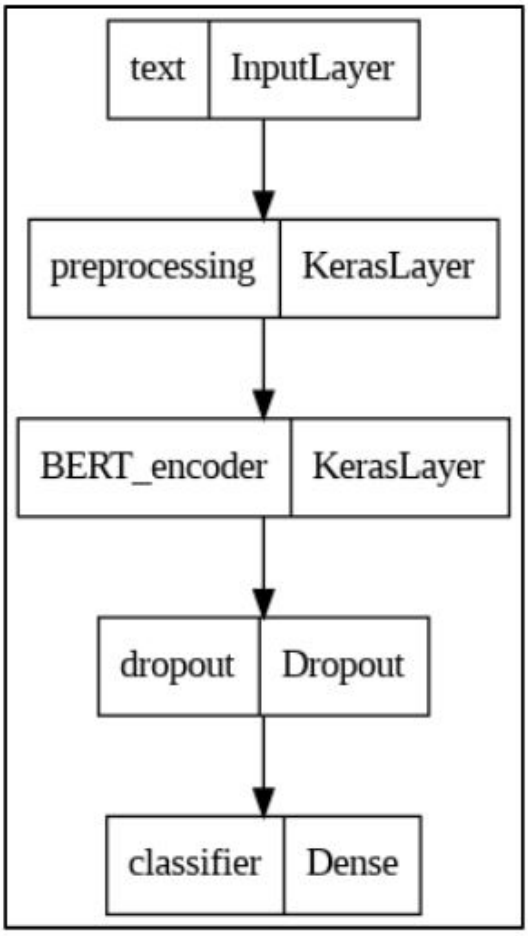


Figure 22: Model4 Architecture

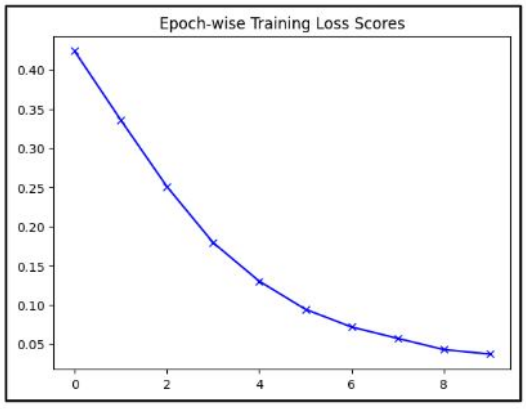


Figure 23: Model4 Training Loss Curve

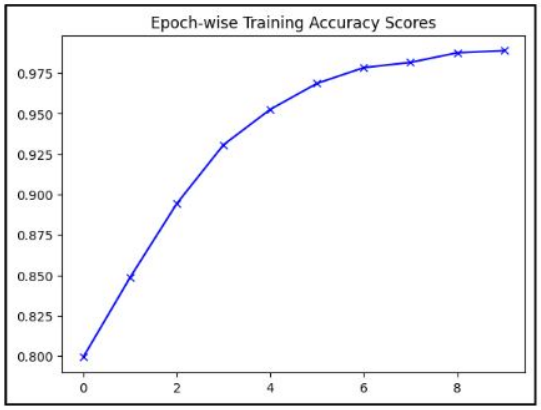


Figure 24: Model4 Training Accuracy Curve