# Summary Report

- **1. Introduction**

In this assignment, I used deep learning models applied to text and sequence data to identify movie reviews as either good or negative. For this, the sentiment analysis dataset from IMDB was employed. I investigated various LSTM-based model architectures, optimization methods, and embedding strategies (random vs. pretrained GloVe). Evaluating the effects of model depth, dropout, and embedding decisions on the model's comprehension of sequential text input and generalization performance was the main goal.

- **2. Preparing Data:**

  After loading the IMDB dataset, Keras' Tokenizer was used to transform each review into an integer sequence. Each sequence was padded to a maximum length of 150 tokens, and the vocabulary was restricted to the top 10,000 words. The data was divided into test, validation, and training sets after being jumbled. Furthermore, glove.6B.100d.txt was loaded in order to use GloVe embeddings, and an embedding matrix was constructed in order to match the word index.

- **3. Model Development:**

  A number of LSTM-based models were constructed and contrasted using various setups. Important differences were as follows:

  Random vs. GloVe (100-dimensional vectors) embedding

  Model depth: Deep stacked versus single LSTM layers

  Dropout layers: used for regularization after LSTM and Dense layers

  Using the Adam optimizer, learning rates were adjusted.

  Activation: Sigmoid for the last output layer; ReLU in dense layers

  Among the models created were:

  Simple LSTM Model: Dense output after one LSTM layer

  Four stacked LSTM layers with several Dense + Dropout layers make up the Deep LSTM Model.

  Final LSTM Model: GloVe embeddings are used to optimize the LSTM model in two layers.

- **4. Training**:

  The Adam optimizer was used to train each model once it was assembled using binary crossentropy loss. Depending on the model, training took place across 10–30 epochs with batch sizes of 12 or 32. The optimal weights based on validation loss were saved using ModelCheckpoint. Where required, early stopping was used to avoid overfitting.

- **5. Evaluation:**
  A test set of 5,000 unread reviews was used to assess each model following training. Metrics for accuracy and loss were noted. Plotting training vs. validation curves was another way to see how the model learned.

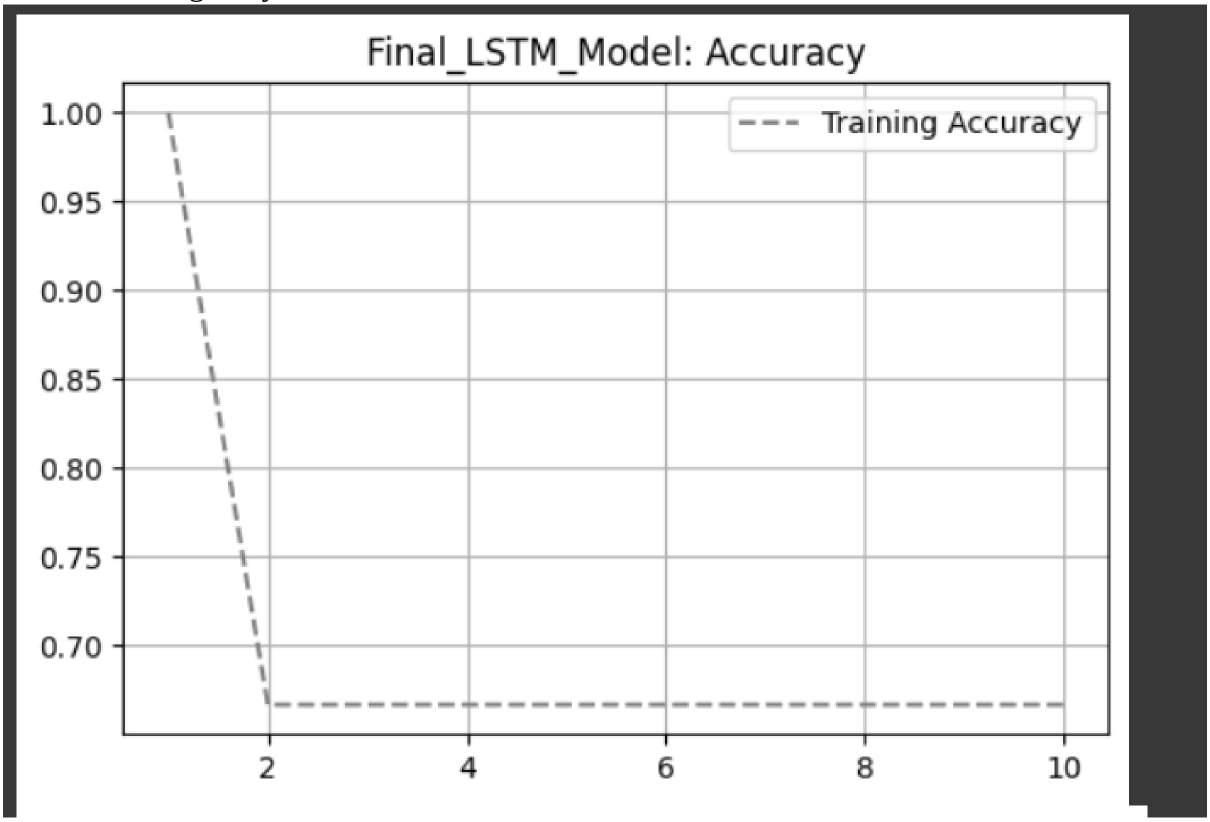
- **6. Graph:**
  Accuracy and loss charts were used to illustrate the performance of several LSTM models. These charts showed:

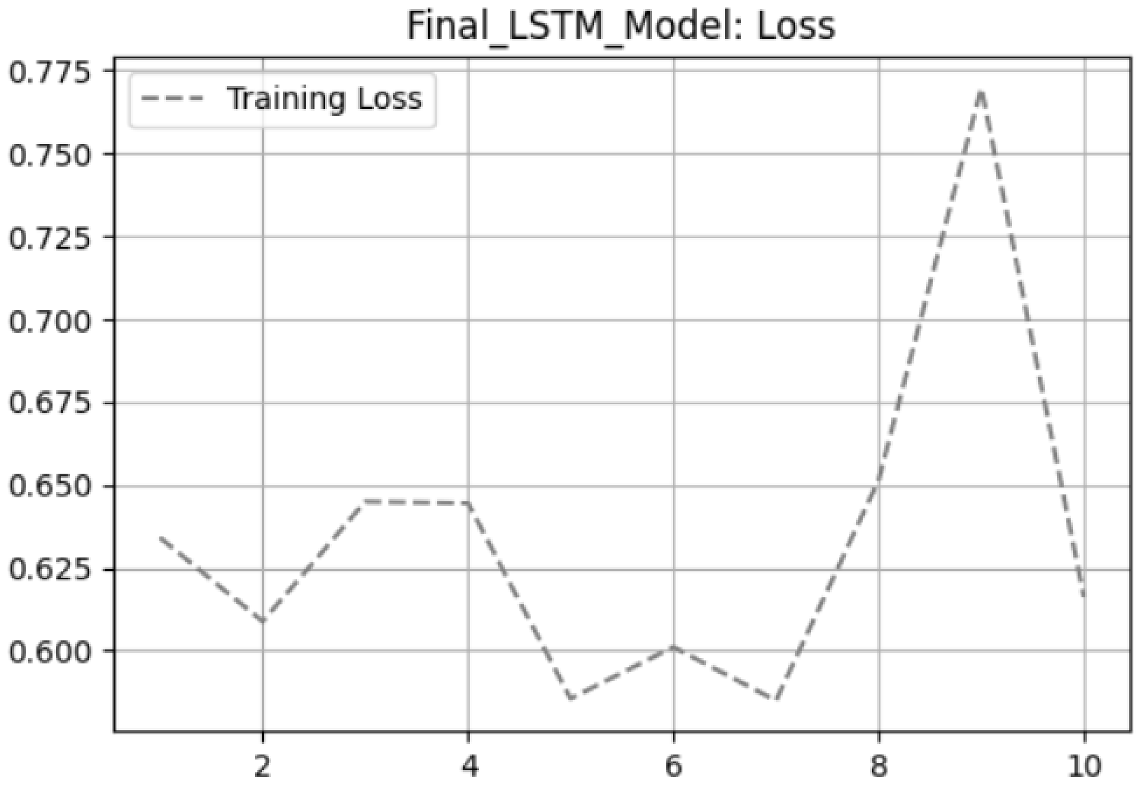
  GloVe embeddings enable the Final LSTM model to perform better.

  the overfitting-induced instability of extremely deep LSTM stacks (such as the 4-layer model).

  How well regularization (Dropout) works to control overfitting.

  Remarkably good results were obtained with less computation by simpler models, such as the single-layer LSTM.






- **7. Conclusion :**

  To sum up, the final LSTM model with GloVe embeddings had the best generalization and accuracy (~90%).

  The Simple LSTM model demonstrated that complexity isn't always necessary by achieving decent performance (~85%) with few layers.

  Despite having a high training accuracy, the Deep LSTM model displayed mild overfitting on validation data.

  Validation metrics were unstable for models with or without dropout.