Experiment- 7

LSTM for Sentiment Analysis

1. Aim

To model longer dependencies in text for classification by training an LSTM on an IMDB subset and evaluating performance using learning curves, confusion matrices and a brief comparison with a simple RNN.

2. Theory

Introduction to Sentiment Analysis

Sentiment Analysis is a supervised natural language processing (NLP) task that determines the emotional polarity (positive or negative) expressed in text. It is widely used in applications such as opinion mining, recommendation systems, and social media analysis.

In this experiment, sentiment analysis is formulated as a binary classification problem:

$$y \in \{0,1\}$$

where,

- 0 = Negative review
- 1 = Positive review

Dataset Description- IMDB Movie Reviews

The experiment uses a subset of the IMDB Large Movie Review Dataset, which is a widely used benchmark dataset for sentiment analysis. It is used due to its real-world text complexity and long-term dependency modelling.

Dataset characteristics:

- Total reviews: 50,000 (original dataset).
- Binary sentiment labels: positive and negative.
- Reviews are pre-labelled and evenly balanced.

Subset used in this experiment:

- Training + Validation: 8,500 reviews.
- Test set: 1,500 reviews.
- Balanced distribution between positive and negative samples.

Movie reviews are particularly challenging because:

- Sentiment may depend on long-range word dependencies.
- Negations and context significantly affect meaning.
- Reviews vary greatly in length.

Hence, sequential models such as RNNs and LSTMs are well suited for this task.

Recurrent Neural Networks (RNNs)

Recurrent Neural Networks process sequential data by maintaining a hidden state that captures information from previous time steps. For a simple RNN, the hidden state and output are computed as:

$$h_t = \tanh(W_h p_t + U_h h_{t-1} + b_h) y_t = \sigma(W_y h_t + b_y)$$ where,

- p_t is the input vector at time step *t.*
- h_{t-1} is the hidden state from the previous time step.
- W_h, U_h, and W_y are weight matrices.
- b_h and b_y are bias vectors.
- $\tanh()$ is the hyperbolic tangent activation function.
- $\sigma()$ denotes the sigmoid activation function used for binary classification.

Despite their conceptual simplicity and ability to model sequences, standard RNNs suffer from several well-known limitations:

- **Vanishing and Exploding Gradient Problem:**
  During backpropagation through time (BPTT), gradients can either shrink exponentially (vanish) or grow uncontrollably (explode), making it difficult for the network to learn long-range dependencies.
- **Difficulty in Learning Long-Term Dependencies:**
  Due to unstable gradient flow, simple RNNs tend to focus only on recent inputs and fail to retain information from earlier time steps, especially in long sequences such as movie reviews.
- **Unstable Training Dynamics:**
  Training deep or long RNNs often requires careful initialization, gradient clipping, and learning rate tuning to prevent divergence.

These limitations motivated the development of more advanced recurrent architectures such as Long Short-Term Memory (LSTM) networks, which introduce gating mechanisms to regulate information flow and improve long-term memory retention.

Long Short-Term Memory (LSTM)

LSTM is a specialized RNN architecture designed to overcome the limitations of standard RNNs by introducing gating mechanisms. It was introduced by Hochreiter and Schmidhuber in 1997 with the explicit purpose of helping address the unstable gradient problem. The gates allow LSTM to retain relevant information and discard irrelevant data over long sequences.

LSTM Cell Components-

- Forget Gate: It decides the type of information that should be thrown away or kept from the cell state. This process is implemented by a sigmoid activation function.

$$f_t = \sigma(W_f[h_{t-1}, p_t] + b_f)$$

- Input Gate: It controls what new information will be added to the cell state from the current input. This gate also plays the role to protect the memory contents from perturbation by irrelevant input.

$$i_t = \sigma(W_i[h_{t-1}, p_t] + b_i)$$

- Candidate Cell State: This is the key to LSTMs and represents the memory of LSTM networks. The LSTM block removes or adds information to the cell state through the gates, which allow optional information to cross.

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, p_t] + b_c)$$

- Cell State Update: This additive update mechanism allows gradients to flow across long sequences, enabling long-term dependency learning.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

- Output Gate: It controls which information to reveal from the updated cell state (C_t) to the output in a single time step. In other words, the output gate determines what the value of the next hidden state should be in each time step.

$$o_t = \sigma(W_o[h_{t-1}, p_t] + b_o)$$

- Hidden State: The hidden state represents the output of the LSTM cell at time step t. It is computed by applying a non-linear transformation to the updated cell state and modulating it with the output gate.

$$h_t = o_t \cdot \tanh(C_t)$$

The neural network architecture for an LSTM block given in Figure 1 demonstrates that the LSTM network extends RNN's memory and can selectively remember or forget information by structures called cell states and three gates. Thus, in addition to a hidden state in RNN, an LSTM block typically has four more layers. These layers are called the cell state (C_t), an input gate (i_t), an output gate (o_t), and a forget gate (f_t). Each layer interacts with each other in a very special way to generate information from the training data.

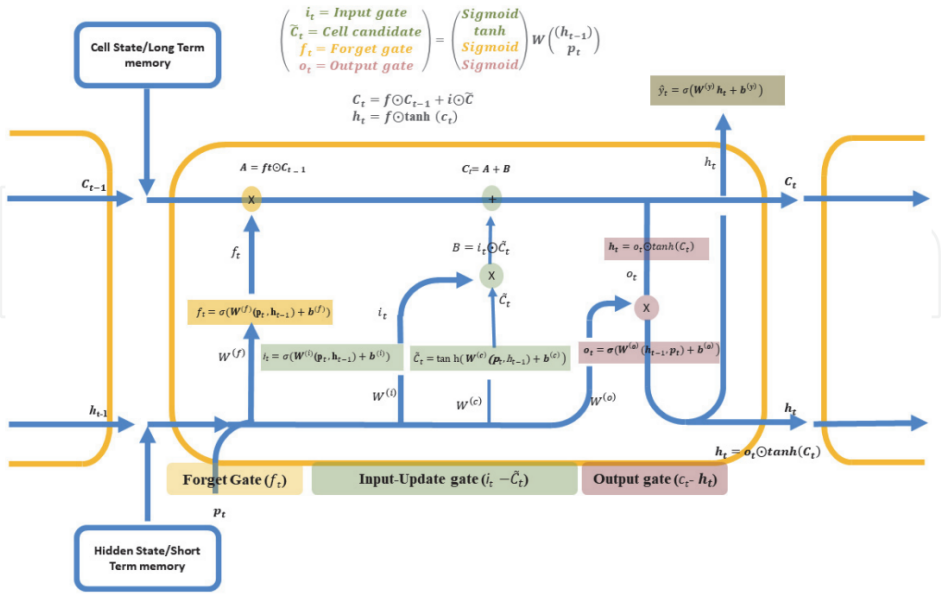

Figure 1- Architecture of LSTM

(Source: H. Okut, "Deep Learning for Subtyping and Prediction of Diseases: Long Short-Term Memory," IntechOpen)

The p_t, h_{t-1}, and C_{t-1} correspond to the input of the current time step, the hidden output from the previous LSTM unit, and the cell state (memory) of the previous unit, respectively. The information from the previous LSTM unit is combined with current input to generate a newly predicted value. The LSTM blocks are mainly divided into three gates: forget, input-update, and output. Each of these gates is connected to the cell state to provide the necessary information that flows from the current time step to the next.

Merits of Long Short-Term Memory

- **Handles Long-Term Dependencies:**

  LSTM networks are capable of learning long-term dependencies in sequential data, overcoming the vanishing gradient problem present in traditional RNNs**.**

- **Effective Memory Management:**

  The use of forget, input, and output gates allows LSTM to selectively store, update, or discard information, leading to better sequence modelling.

- **Suitable for Sequential Data:**

  LSTMs perform well on time-series, text, speech, and sentiment analysis tasks where data has temporal dependencies.

- **Stable Training:**

  Due to controlled gradient flow through the cell state, LSTMs provide more stable training compared to simple RNNs.

- **Better Performance on Contextual Tasks:**

  LSTMs capture contextual information over longer sequences, improving performance in tasks such as language modelling and text classification.

Demerits of Long Short-Term Memory

- **High Computational Cost:**
  LSTM networks involve multiple gate computations, making them computationally expensive compared to standard RNNs.

- **Longer Training Time:**
  Due to their complex structure, LSTMs require more time to train, especially on large datasets.

- **Large Memory Requirement:**
  The presence of multiple weight matrices and gates increases memory consumption.

- **Risk of Overfitting on Small Datasets:**
  When trained on small datasets, LSTMs may overfit if proper regularization techniques are not applied.

- **Complex Architecture:**
  The internal structure of LSTM cells makes them harder to understand, tune, and debug compared to simpler models.

3. Pre-Test (MCQs)

1. What is Sentiment Analysis?

a) Identifying the genre of a movie
(Incorrect because sentiment analysis focuses on opinion or emotion, not genre classification.)

b) Detecting emotions or opinions expressed in text
(Correct because sentiment analysis determines whether the sentiment in text is positive or negative.)

c) Translating text from one language to another
(Incorrect because translation is a different NLP task.)

d) Compressing text documents
(Incorrect because sentiment analysis does not deal with data compression.)

Answer: b

2. In sentiment analysis, a movie review labelled as “1” usually indicates:

a) Neutral sentiment
(Incorrect because this experiment considers only binary sentiment classes.)

b) Negative review
(Incorrect because negative sentiment is usually encoded as 0.)

c) Positive review
(Correct because in binary sentiment classification, 1 represents a positive review.)

d) Unlabelled data
(Incorrect because labels are predefined in supervised learning.)

Answer: c

3. Why are movie reviews challenging for sentiment analysis?

a) Long-range dependencies and negations affect meaning
(Correct because context and negation can completely change sentiment.)

b) Sentiment depends only on individual words
(Incorrect because sentiment often depends on context and word order.)

c) Reviews always have fixed length
(Incorrect because reviews vary greatly in length.)

d) Reviews contain only numerical data
(Incorrect because reviews are text-based.)

Answer: a

4. What is the main limitation of a simple Recurrent Neural Network (RNN)?

a) It cannot process sequential data
(Incorrect because RNNs are designed for sequential data.)

b) It suffers from vanishing and exploding gradient problems
(Correct because this limits the ability of RNNs to learn long-term dependencies.)

c) It requires labelled image data
(Incorrect because RNNs are used for sequence data, not necessarily images.)

d) It cannot be trained using backpropagation
(Incorrect because RNNs are trained using backpropagation through time.)

Answer: b

5. What is the primary purpose of gates in an LSTM network?

a) To increase the number of neurons
(Incorrect because gates control information flow, not network size.)

b) To randomly drop connections
(Incorrect because dropout is a separate regularization technique.)

c) To convert text into numerical vectors
(Incorrect because tokenization and embeddings handle that task.)

d) To control what information is stored, forgotten, and output
(Correct because LSTM gates regulate memory flow in the cell state.)

Answer: d

6. Which dataset is commonly used for benchmarking sentiment analysis models?

a) MNIST
(Incorrect because MNIST is used for handwritten digit recognition.)

b) CIFAR-10
(Incorrect because CIFAR-10 is an image classification dataset.)

c) IMDB Movie Reviews Dataset
(Correct because IMDB is a standard dataset for binary sentiment classification.)

d) ImageNet
(Incorrect because ImageNet is used for large-scale image classification.)

Answer: c

7. Sentiment analysis is an example of which type of learning?

a) Unsupervised learning
(Incorrect because sentiment labels are provided during training.)

b) Reinforcement learning
(Incorrect because there is no reward-based interaction.)

c) Supervised learning
(Correct because the model learns from labelled positive and negative reviews.)

d) Self-supervised learning
(Incorrect because explicit labels are used.)

Answer: c

8. What does the term “binary classification” mean in this experiment?

a) Classifying data into two distinct classes
(Correct because reviews are classified as positive or negative.)

b) Predicting numerical values
(Incorrect because outputs are class labels, not continuous values.)

c) Classifying text into multiple categories
(Incorrect because only two sentiment classes are considered.)

d) Clustering text documents
(Incorrect because clustering is an unsupervised task.)

Answer: a

9. Why is word order important in sentiment analysis?

a) Because models ignore punctuation
(Incorrect because punctuation handling does not define sentiment.)

b) Because sentiment depends on context and sequence of words
(Correct because phrases like “not good” and “good” have opposite meanings.)

c) Because reversing words does not change meaning
(Incorrect because word order can completely change sentiment.)

d) Because all words have equal importance
(Incorrect because some words influence sentiment more than others.)

Answer: b

10. Why is the IMDB dataset suitable for LSTM-based sentiment analysis?

a) It contains only short sentences
(Incorrect because reviews vary in length.)

b) It is balanced and contains long text sequences
(Correct because LSTMs handle long-term dependencies well.)

c) It has image and text data combined
(Incorrect because the dataset contains only text.)

d) It requires no preprocessing
(Incorrect because text preprocessing is necessary.)

Answer: b

4. Procedure

1. Importing Libraries:

All necessary libraries for data handling, text preprocessing, model construction, training, and performance visualization are imported.

2. Dataset Loading:

The IMDB movie review dataset is loaded by reading text files from the positive and negative review directories. Each review is assigned a sentiment label, where positive reviews are labelled as 1 and negative reviews as 0.

3. Text Preprocessing:

The raw text reviews are cleaned to remove noise and irrelevant characters. This includes converting text to lowercase, removing punctuation marks, digits, special characters, and extra whitespaces.

4. Data Splitting:

The cleaned dataset is divided into three subsets:

- Training set for learning model parameters.

- Validation set for hyperparameter tuning and model selection.
- Test set for final performance evaluation.

The split is performed in a balanced manner to maintain equal representation of positive and negative reviews.

5. Text Tokenization:

The processed text is converted into numerical format using tokenization. A fixed vocabulary size is defined, and each unique word is mapped to a corresponding integer index. Words outside the vocabulary are replaced with an out-of-vocabulary token.

6. Sequence Padding:

Since movie reviews vary in length, all tokenized sequences are padded or truncated to a fixed maximum sequence length. This ensures uniform input size for the neural network models.

7. Embedding Layer Construction:

An embedding layer is added to both models to transform word indices into dense vector representations. This layer helps capture semantic relationships between words and improves model performance.

8. Simple RNN Model Construction:

A Simple Recurrent Neural Network (RNN) model is built using an embedding layer followed by one or more recurrent layers. The final layer uses a sigmoid activation function to perform binary sentiment classification.

9. LSTM Model Construction:

An LSTM model is constructed using an embedding layer followed by LSTM layers with gating mechanisms. Dropout and recurrent dropout are applied to reduce overfitting. A sigmoid-activated output layer is used for binary classification.

10. Model Compilation:

Both RNN and LSTM models are compiled using:

- Binary Cross-Entropy as the loss function.
- Adam optimizer for efficient gradient-based optimization.
- Accuracy as the evaluation metric.

11. Model Training:

The models are trained on the training dataset for 150 epochs with 128 batch size. Learning rate scheduling is applied to reduce the learning rate when validation loss plateaus. Model checkpointing is used to save the best-performing model based on validation accuracy.

12. Model Selection:

After training, the saved checkpoints corresponding to the highest validation accuracy are loaded. This ensures that the evaluation is performed using the best version of each model.

13. Model Evaluation:

The trained RNN and LSTM models are evaluated on the unseen test dataset. Performance metrics include all accuracies, confusion matrix, and classification report consisting of precision, recall, and F1-score.

14. Performance Visualization:

Learning curves for training and validation loss and accuracy are plotted to analyse model convergence and overfitting behaviour. Additionally, ROC curves and precision–recall curves are generated to assess classification performance more comprehensively.

15. Performance Comparison:

Finally, the performance of the Simple RNN and LSTM models is compared based on accuracy, loss trends, ROC–AUC scores, and classification metrics to highlight the effectiveness of LSTM in handling long-term dependencies in sentiment analysis tasks, by analysing individual curves for both RNN and LSTM, as well as the combined curves.

5. Post-Test (MCQs)

1. Why is text preprocessing necessary before training the RNN and LSTM models?
a) To remove noise such as punctuation, digits, and extra spaces
(Correct because cleaning improves model learning by reducing irrelevant information.)
b) To increase the number of words in each review
(Incorrect because preprocessing aims to reduce noise, not increase text size.)
c) To convert labels into numeric form
(Incorrect because labels are already numeric and preprocessing is applied to text.)
d) To increase the training epochs automatically
(Incorrect because preprocessing does not control training duration.)
Answer: a

2. What is the purpose of tokenization in the experiment?
a) To directly classify text into positive or negative classes
(Incorrect because tokenization only converts text into numerical format.)
b) To map words into integer indices for neural network input
(Correct because neural networks require numerical input, not raw text.)
c) To remove stop words permanently from the dataset
(Incorrect because tokenization does not necessarily remove stop words.)
d) To evaluate model accuracy
(Incorrect because evaluation is done after model prediction.)
Answer: b

3. Why are sequences padded to a fixed maximum length?
a) To increase vocabulary size
(Incorrect because padding does not affect vocabulary.)
b) To ensure all inputs have the same length for batch processing
(Correct because neural networks require uniform input dimensions.)
c) To improve GPU memory usage
(Incorrect because padding may increase memory usage.)
d) To convert labels into binary format
(Incorrect because padding applies to input sequences, not labels.)
Answer: b

4. What role does the Embedding layer play in the models?
a) It converts probabilities into class labels
(Incorrect because classification is done by the output layer.)

b) It reduces the number of epochs required
(Incorrect because embedding does not directly control training duration.)
c) It performs sentiment classification directly
(Incorrect because embeddings are feature representations, not classifiers.)
d) It learns dense vector representations of words
(Correct because embeddings capture semantic relationships between words.)
Answer: d

5. Why is binary cross-entropy used as the loss function in this experiment?
a) Because the dataset has multiple sentiment classes
(Incorrect because the task is binary classification.)
b) Because it reduces overfitting automatically
(Incorrect because regularization techniques address overfitting.)
c) Because it is suitable for binary classification problems
(Correct because binary cross-entropy measures error between predicted and actual binary labels.)
d) Because it works only with RNNs
(Incorrect because it is used across many binary classifiers.)
Answer: c

6. What does a confusion matrix help evaluate in sentiment analysis?
a) Word frequency distribution
(Incorrect because it evaluates prediction outcomes, not vocabulary.)
b) Relationship between training and validation loss
(Incorrect because loss curves handle that.)
c) Correct and incorrect classification counts
(Correct because it shows true positives, true negatives, false positives, and false negatives.)
d) Sequence length distribution
(Incorrect because padding controls sequence length.)
Answer: c

7. Why are ROC and Precision–Recall curves plotted in this experiment?
a) To visualize training time
(Incorrect because they visualize classification performance.)
b) To analyse classifier performance across different thresholds
(Correct because these curves evaluate trade-offs between true positives, false positives, precision, and recall.)
c) To compare vocabulary sizes
(Incorrect because curves do not depend on vocabulary.)
d) To determine batch size
(Incorrect because batch size is a training parameter.)
Answer: b

8. Why does the LSTM model generally perform better than a Simple RNN in this experiment?
a) LSTM has fewer parameters
(Incorrect because LSTM has more parameters than RNN.)
b) LSTM removes vanishing gradients completely
(Incorrect because it mitigates but does not eliminate the problem.)
c) LSTM requires less training data
(Incorrect because data requirements depend on task complexity.)

d) LSTM can capture long-term dependencies in text
(Correct because gating mechanisms allow LSTM to retain relevant information.)
Answer: d

9. What does validation accuracy indicate during training?
a) Model performance on unseen test data
(Incorrect because validation data is separate from the test set.)
b) Model performance on training data only
(Incorrect because training accuracy reflects training performance.)
c) Model generalization performance during training
(Correct because validation accuracy estimates how well the model generalizes.)
d) Vocabulary learning quality
(Incorrect because vocabulary is fixed during tokenization.)
Answer: c

10. What is the final objective of comparing Simple RNN and LSTM models in this experiment?
a) To demonstrate the importance of long-term dependency modelling in sentiment analysis
(Correct because movie reviews require contextual understanding over long sequences.)
b) To prove that RNN is always better than LSTM
(Incorrect because results depend on task complexity.)
c) To reduce dataset size
(Incorrect because comparison does not affect dataset.)
d) To eliminate the need for preprocessing
(Incorrect because preprocessing remains essential.)
Answer: a

6. References

1. I. Goodfellow, Y. Bengio, and A. Courville, “Deep Learning,” MIT Press, Cambridge, MA, USA, 2016.
2. S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, MIT Press, Cambridge, MA, USA, 1997.
3. M. A. Nielsen, “Neural Networks and Deep Learning,” Determination Press, 2015.