

1 Logistic Regression Model

For this task, we have been provided with the IMDB Sentiment Dataset. This dataset has been split into training, validation, and test sets in the ratio 7:3:10.

The dataset is loaded into a DataLoader for training, validation, and testing with a custom batching function specified using the `collate fn` parameter. Before loading the dataset into the DataLoader, the input vectors have been padded with `< pad >` token, which maps to dictionary index '0' to ensure all input vectors in a batch are of the same length.

The model used is a *Logistic Regression* with an embedding layer that takes in text and outputs a sentiment score, processed by a fully connected layer for the final output.

The model is trained using Adam optimizer with a learning rate of $1e-2$. The learning rate is a hyperparameter that controls the step size of the optimization algorithm, with a smaller rate resulting in slower convergence but a higher chance of reaching the global minimum.

Logistic Regression Model:

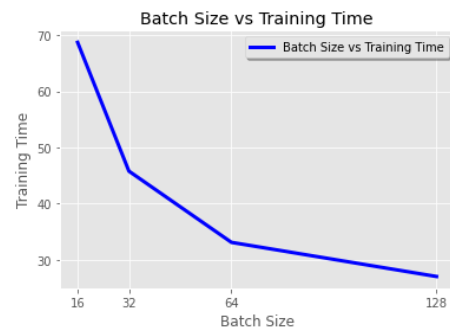
LR((embedding): Embedding(25002, 100)

(fc): Linear(in_features=100, out_features=1, bias=True))

1.1 Batch Training Time Data

S.No	Batch Size	Training Time (sec)
1.	16	68.71
2.	32	45.78
3.	64	33.09
4.	128	27.01

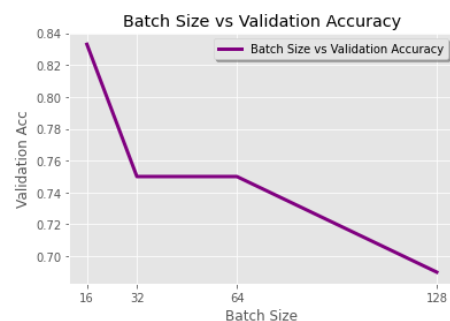
Table 1: Batch Size vs Training Time



1.2 Validation Accuracy Data

S.No	Batch Size	Validation Accuracy
1.	16	0.833
2.	32	0.75
3.	64	0.75
4.	128	0.69

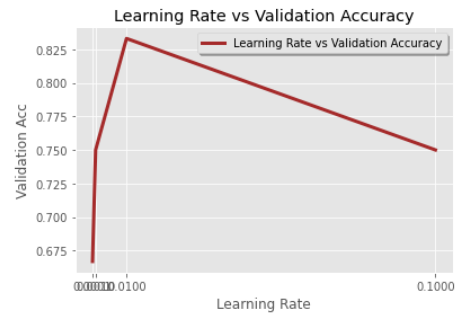
Table 2: Batch Size vs Validation Accuracy



1.3 Learning Rate vs Validation Accuracy

S.No	Learning Rate	Validation Accuracy
1.	1e-1	0.75
2.	1e-2	0.833
3.	1e-3	0.75
4.	1e-4	0.667

Table 3: Learning Rate vs Validation Accuracy



1.4 Best Model

Fixed Parameters:

Input Dimension: Vocab Size = 25002

Output Dimension: 1

Loss Function: BCEWithLogitsLoss (Binary Cross Entropy)

Hyper-Parameters:

S.No	Hyper-Paramter	Value
1.	Epochs	7
2.	Optimizer	Adam
3.	Learning Rate	1e-2
4.	Embedding Dimension	100
5.	Batch Size	16

Table 4: Hyper-Parameters for Logistic Regression

Training Accuracy: 0.917

Validation Accuracy: 0.833

Test Accuracy: 0.75



Figure 1: Plot of Training vs Validation Loss

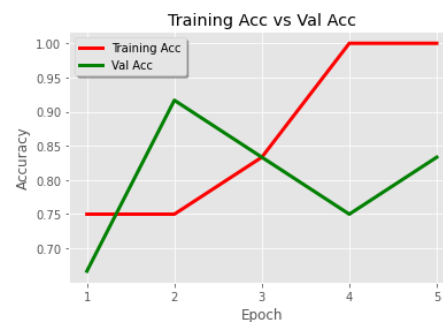


Figure 2: Plot of Training vs Validation Accuracy

2 LSTM Model

Before loading the dataset into the DataLoader, we need to pad the input sequences the same way we did for our Logistic Regression model, but there is a catch when we do padding for LSTM. Since we are building an LSTM model, we need it to process and backpropagate loss for only the non-padded elements

to reduce the computational overhead. So we also pass a length tensor for each batch that contains the length of each input vector in a batch that helps the LSTM cell recognize the non-padded elements for each input sequence. This is done by packing the padded input sequence (using *pack_padded_sequence*) before feeding it into the LSTM cell and unpacking the output (using *pad_packed_sequence*) of the LSTM cell.

The model used is an *LSTM* with an embedding layer that takes in the text to give an embedded output. Dropout is applied to this embedded output, and it is fed to an LSTM cell. Since we are using a Bi-LSTM in this case, we concatenate the final *forward* (*hidden[-2,:]*) and *backward* (*hidden[-1,:]*) hidden layers and apply dropout, processed by a fully connected layer for the final output.

The model is trained using Adam optimizer with a learning rate of 1e-2 and using Binary Cross Entropy Loss Function.

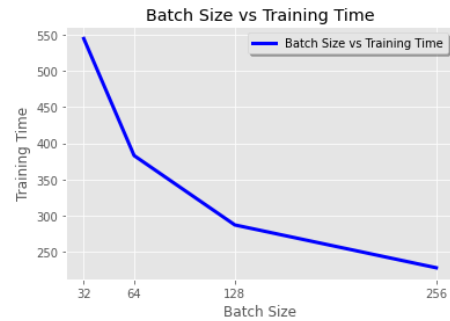
LSTM Model:

LSTM((embedding): Embedding(25002, 100)
 (lstm): LSTM(100, 256, num_layers=2, dropout=0.2, bidirectional=True)
 (fc): Linear(in_features=512, out_features=1, bias=True)
 (dropout): Dropout(p=0.2, inplace=False))

2.1 Batch Training Time Data

S.No	Batch Size	Training Time (sec)
1.	32	545.12
2.	64	382.94
3.	128	286.81
4.	256	227.46

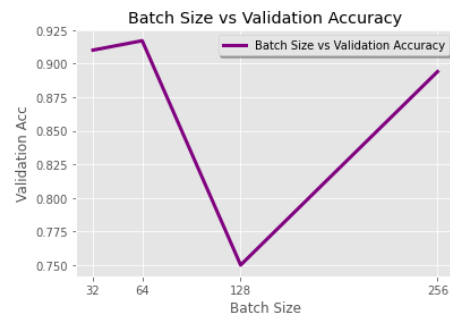
Table 5: Batch Size vs Training Time



2.2 Validation Accuracy Data

S.No	Batch Size	Validation Accuracy
1.	32	0.91
2.	64	0.917
3.	128	0.75
4.	256	0.894

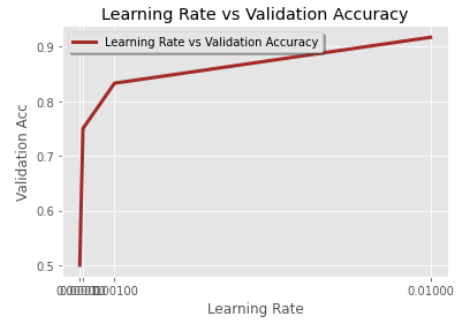
Table 6: Batch Size vs Validation Accuracy



2.3 Learning Rate vs Validation Accuracy

S.No	Learning Rate	Validation Accuracy
1.	$1e-2$	0.917
2.	$1e-3$	0.833
3.	$1e-4$	0.75
4.	$1e-5$	0.50

Table 7: Learning Rate vs Validation Accuracy



2.4 Best Model

Fixed Parameters:

Input Dimension: Vocab Size = 25002

Hidden Dimension: 256

Output Dimension: 1

Number of Layers: 2

Type of LSTM Cell: Bi-Directional

Dropout: 0.2

Loss Function: BCEWithLogitsLoss (Binary Cross Entropy)

S.No	Hyper-Paramter	Value
1.	Epochs	5
2.	Optimizer	Adam
3.	Learning Rate	$1e-2$
4.	Embedding Dimension	100
5.	Batch Size	64

Table 8: Hyper-Parameters for LSTM

Hyper-Parameters:

Training Accuracy: 0.93
Validation Accuracy: 0.917
Test Accuracy: 0.95

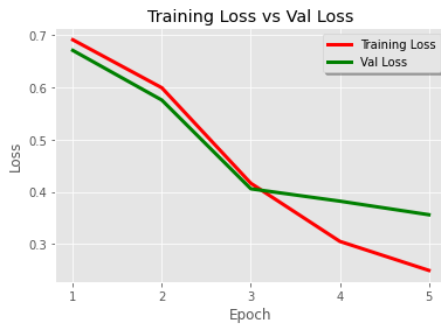


Figure 3: Plot of Training vs Validation Loss

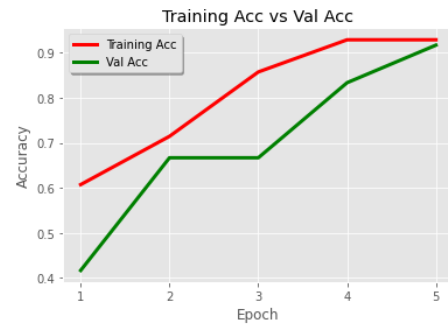


Figure 4: Plot of Training vs Validation Accuracy