Y19 Artificial Intelligence II
Deep Learning for Natural Language Processing
Project 2

Oikonomou Stylianos - 1115201500243

Announced: November 30, 2022 $\|Due : December 31, 2022$

1 Preprocessing

In the preprocessing stage our goal is to remove as much noise as possible from the data sets that will interfere with the classification. The methods used to "clean up" the reviews are the following:

- Remove the left over HTML tags
- Remove words containing numbers
- Remove all non letter characters
- Clean up words containing apostrophe
- Remove multiple white spaces
- Turn all letters to lower case
- Remove stop words

The urls of the reviews were dropped from the data set as they could not provide significant context about whether a rating is positive or not.

The numbers 0 or 1 replaced the movie rating values to symbolise the negative and positive ratings respectively. Thus the classification turns into a binary problem based on those values.

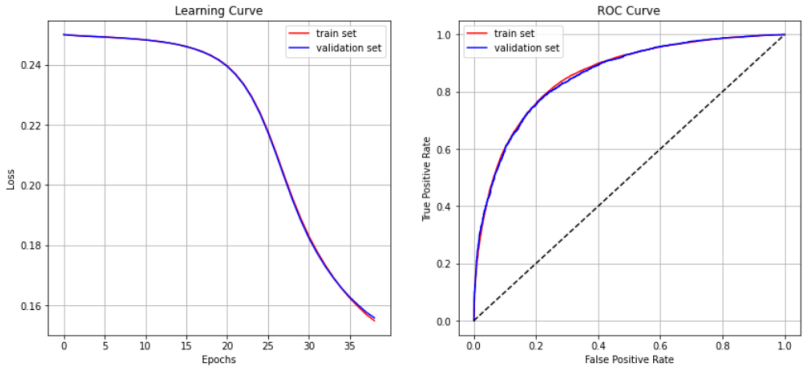
2 Feature Selection

The features were created by utilizing the Standford GloVe embeddings. Each review has a 1 by DIM array of word features were DIM is the feature dimension of the glove file. The feature array was made calculating the mean of the feature columns of the words in the review. If a word did not exist in the dictionary no features were added to the sum of the columns and the divisor would be reduced by one.

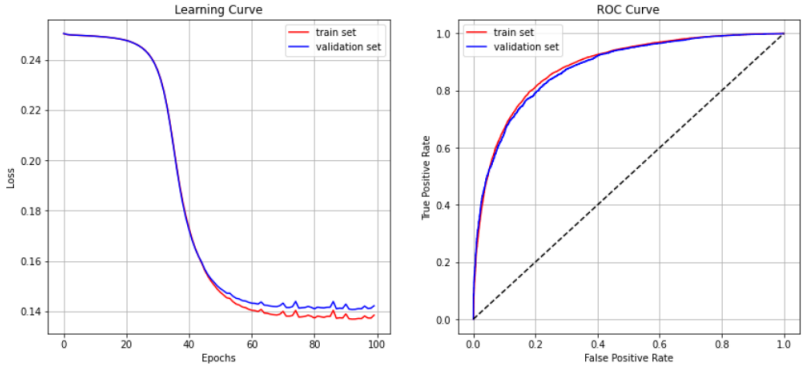
3 Early Stopping

In order to avoid the overfitting and to make the best generalized version of our model an early stopping method was implemented. This method was fine tuned by the two parameters: patience and loss_diff_limit. During the training of our model if the difference between the valid and train set predictions was above the loss_diff_limit a counter was increased by one. If the counter exceed the patience parameter then the training would stop and the model would restore the last best version.

With Early Stopping (dim=100)



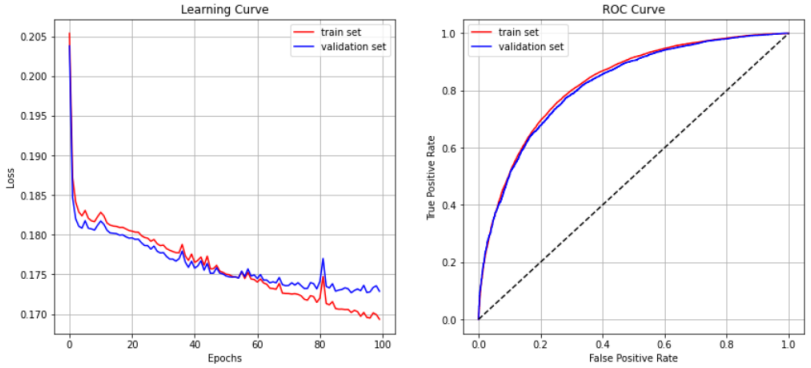
Without Early Stopping (dim=100)



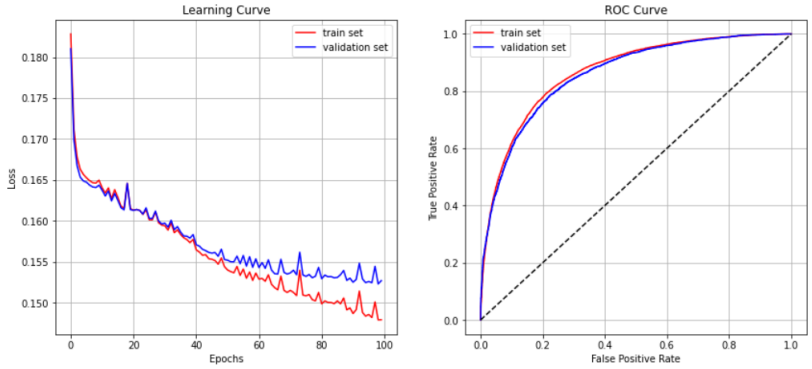
4 Experimentation

Stemming and Lemmatization

Using Stemmer (dim=50)

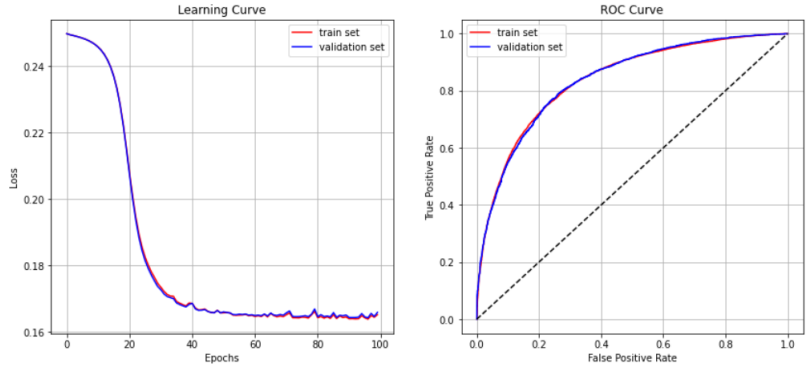


Using Lemmatizer (dim=50)

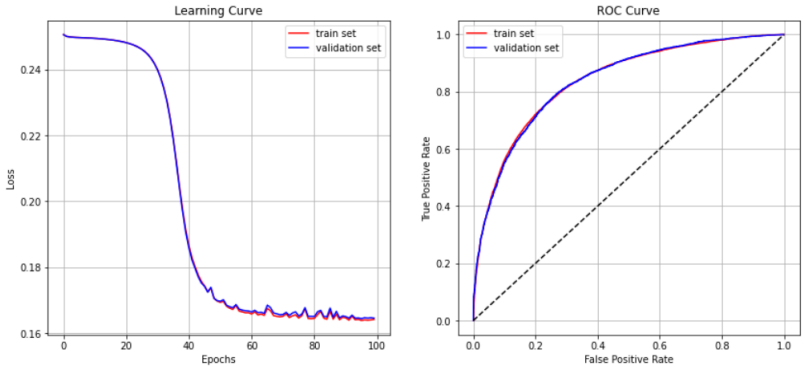


Layers

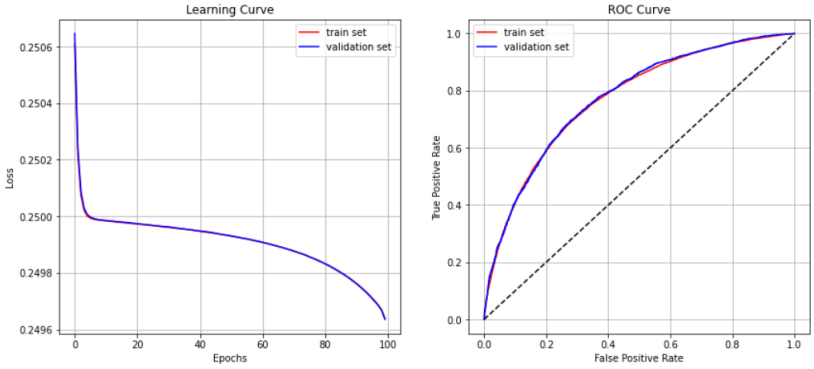
2 Inner Layers (dim=50)



3 Inner Layers (dim=50)



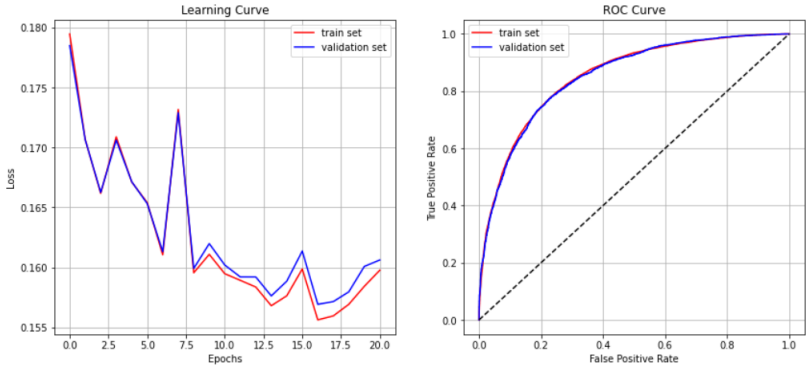
4 Inner Layers (dim=50)



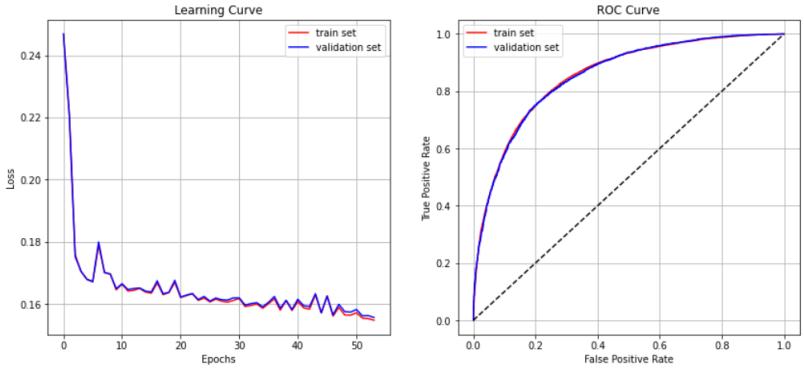
Results : Adding more layers did not increase the performance of the model. In fact it increased the complexity, execution time and as we see in the 4 layer example the loss became much larger than the 2 inner layers example.

Lerning Rate

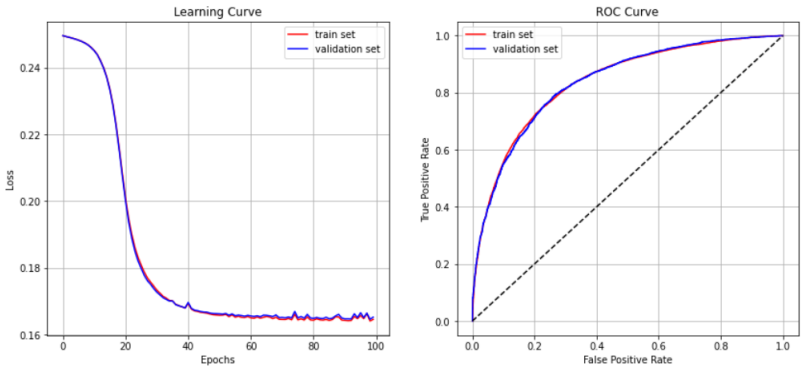
0.01 Learning Rate (dim=50)



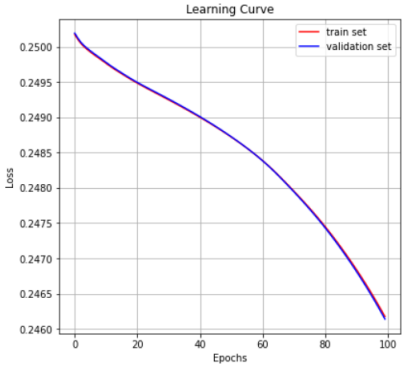
0.001 Learning Rate (dim=50)

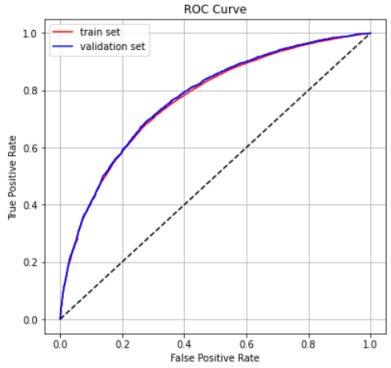


0.0001 Learning Rate (dim=50)



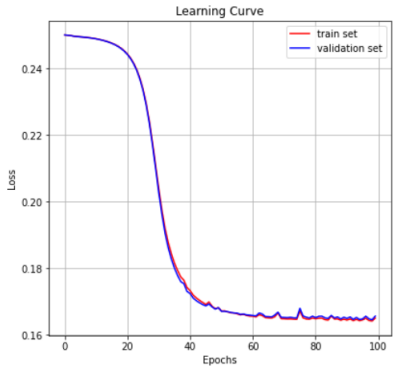
0.00001 Learning Rate (dim=50)

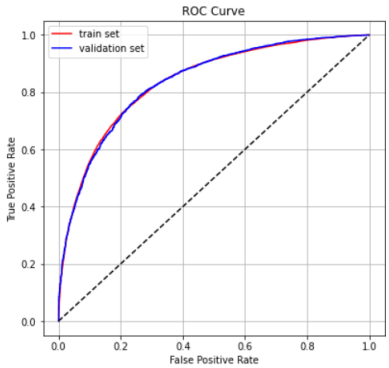




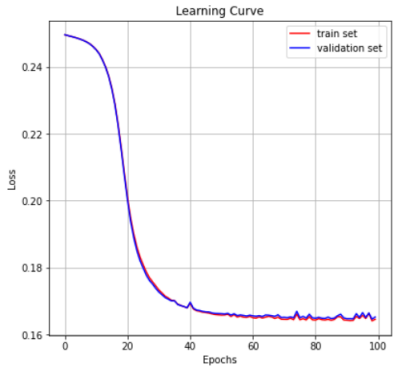
Batch Size

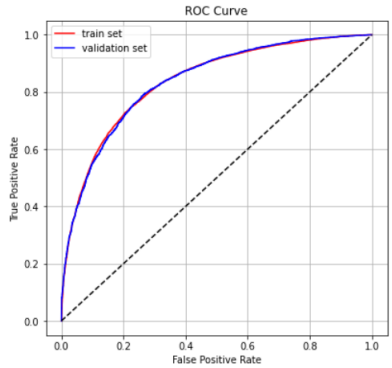
32 Batch Size (dim=50)



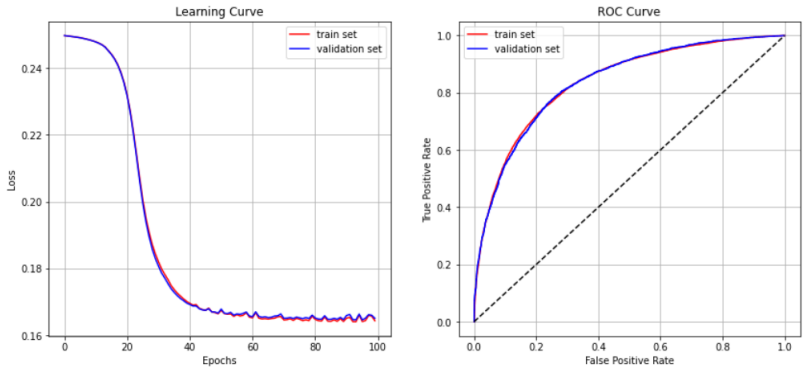


64 Batch Size (dim=50)

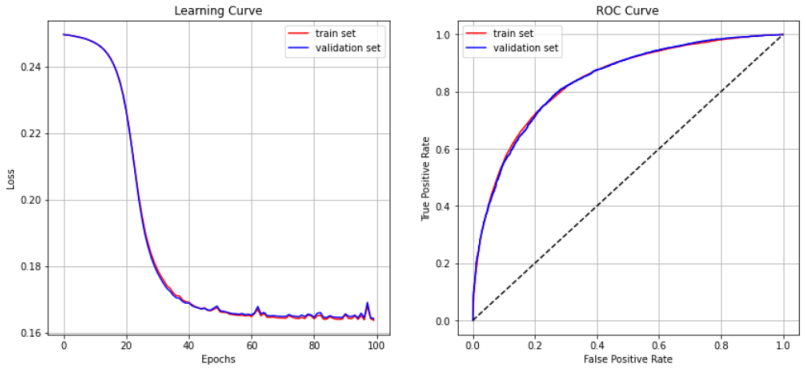




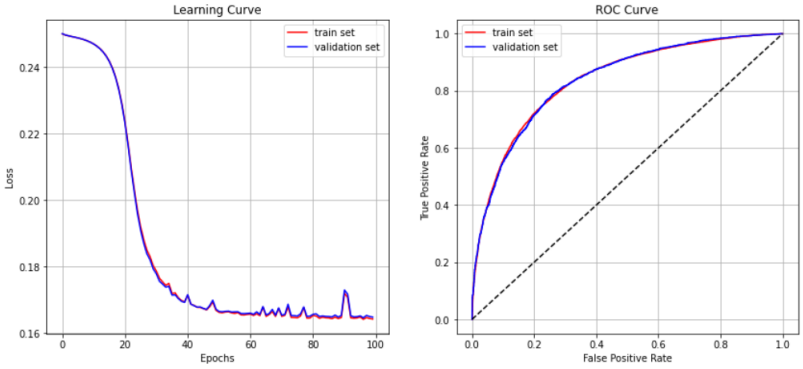
128 Batch Size (dim=50)



256 Batch Size (dim=50)



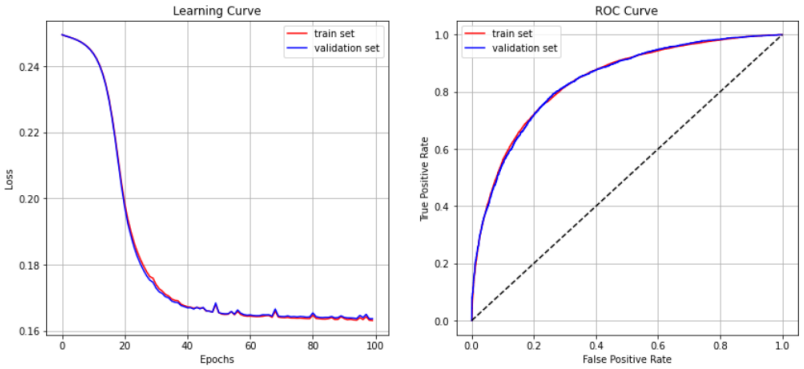
512 Batch Size (dim=50)



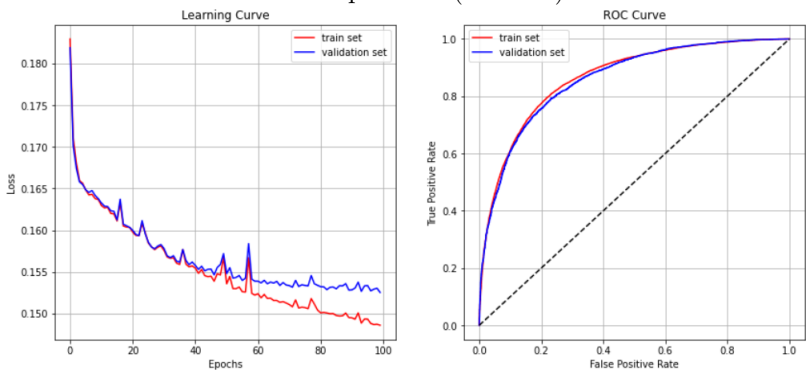
Results : The batch size parameter did not influence the results as much as the other model parameters did

Optimizer

SGD Optimizer (dim=50)



Adam Optimizer (dim=50)



5 Best Model

Inner Layers: 2
Batch Size: 64
Learning Rate: 0.0001
Loss Function: MSELoss
Optimizer: Adam