Artificial Intelligence 2, Project 2

Georgios Nikolaou

April 12, 2023

Tuning

F1-Score Tables

One Layer NN

SGD	ReLU	LeakyReLU	GELU
No shceduler	84.27%	84.28%	83.95%

Adam/Adamax	ReLU	LeakyReLU	GELU	ReLU	LeakyReLU	GELU
No scheduler	84.46%	85.03%	85.09%	85.28%	85.17%	85.23%
CyclicLR	85.15%	85.29%	84.29%	84.30%	84.25%	84.05%
ReduceOnPlateuLR	84.50%	84.64%	84.49%	84.40%	84.43%	84.13%
LinearLR	84.55%	84.66%	84.40%	84.39%	84.41%	84.01%

Dense NN

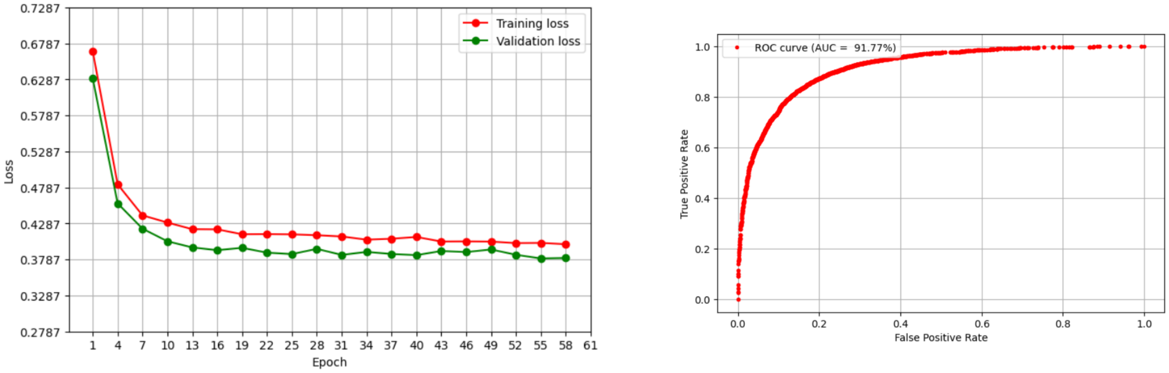
SGD	ReLU	LeakyReLU	GELU
No scheduler	84.60%	84.73%	84.16%

Adam/Adamax	ReLU	LeakyReLU	GELU	ReLU	LeakyReLU	GELU
No scheduler	84.72%	84.70%	84.45%	84.95%	84.98%	84.38%
CyclicLR	85.10%	85.06%	84.29%	84.78%	84.73%	84.10%
ReduceOnPlateuLR	84.83%	85.09%	84.66%	84.21%	84.23%	84.09%
LinearLR	83.93%	84.83%	83.50%	84.63%	84.47%	83.91%

Curves and Observations

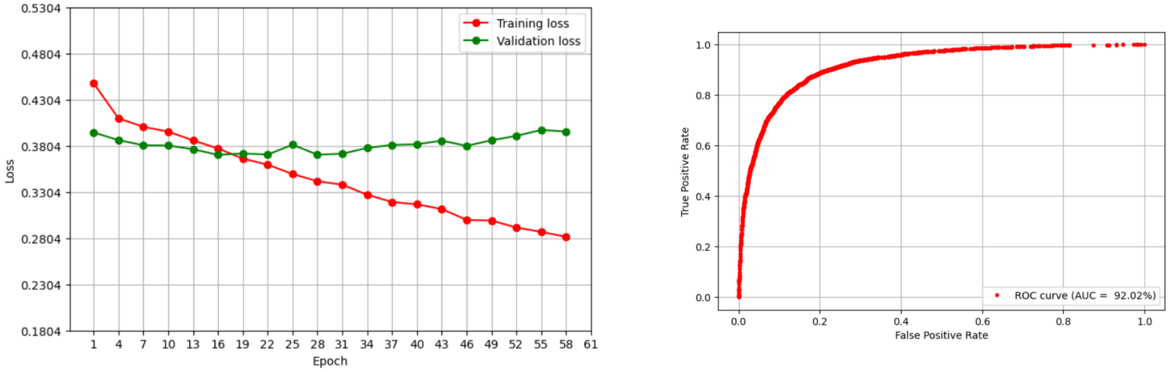
One Layer NN (One hidden layer with 1024 neurons)

Figure 1: SGD, No Scheduler



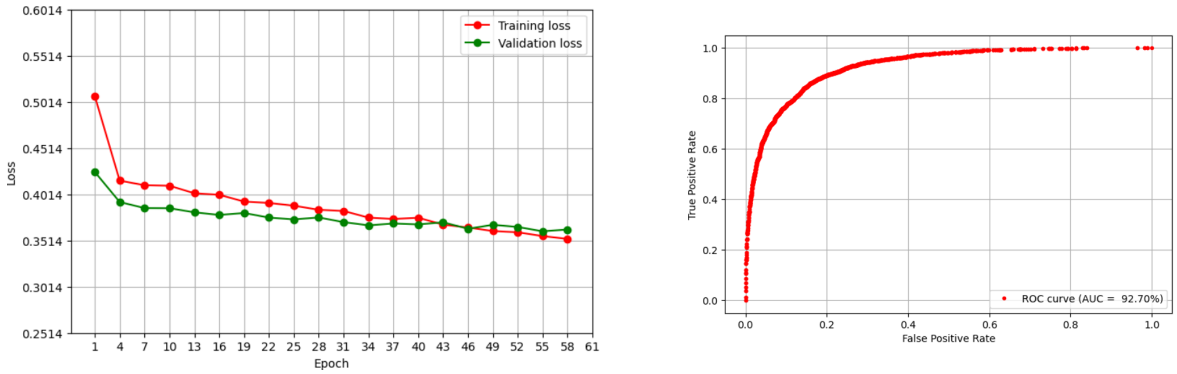
We get these curves for the base model.
The learning curve indicates that there is no significant overfitting, and given the score it achieves, we can conclude that despite its simplicity, it is a good model.

Figure 2: Adam, No Scheduler



There is clearly overfitting, starting after around 20 epochs. Here, early stopping should be applied to prevent further training, as it is clear that the model's performance would not improve.
If we do not follow the approach of selecting the best epoch, the AUC is approximately 90%, which is lower than every other test where the approach was followed. Therefore, we can more clearly see the effects of overfitting.

Figure 3: Adam, CyclicLR



Simply having a scheduler gave us an impressive learning curve.
We notice that in contrast with the base model, the loss is significantly lower in the beginning and is steadily reducing for many more epochs.
We also notice that if we continued there would probably be overfitting and therefore we should implement early stopping if we'd like to do so.
We conclude that the model is very good and should generalize well.

Figure 4: Adamax, No Scheduler, ReLU

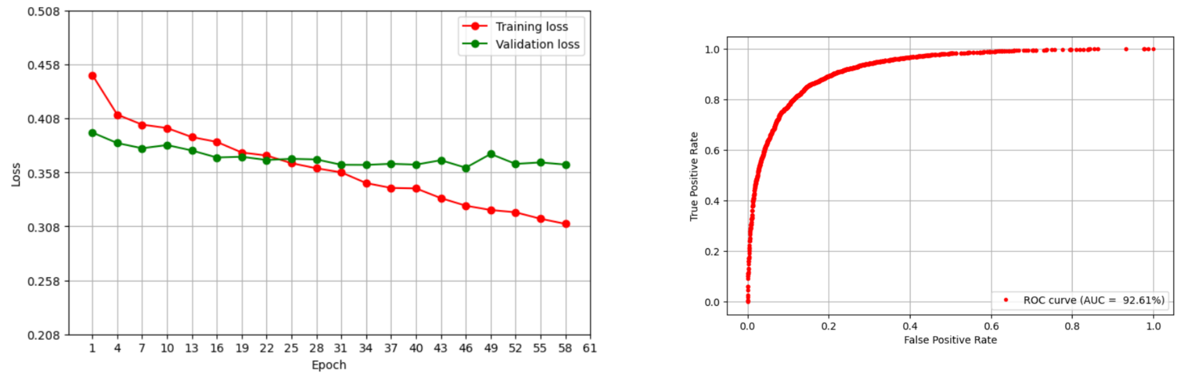
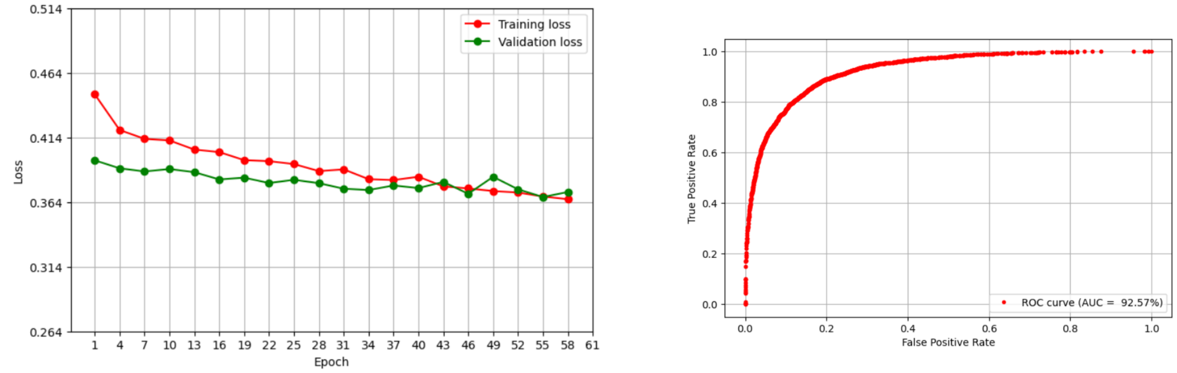


Figure 5: Adamax, No Scheduler, GELU



Here we have a model that the activation function played a significant role on the degree to which the model is overfitting. For the first case see figure 2 and for the second see figure 3.
GELU is smoother than ReLU around zero and suffers less from too much neurons becoming zero.

Dense NN (3 hidden layers with 1024, 256 and 32 neurons respectively)

Figure 6: SGD, No Scheduler, ReLU

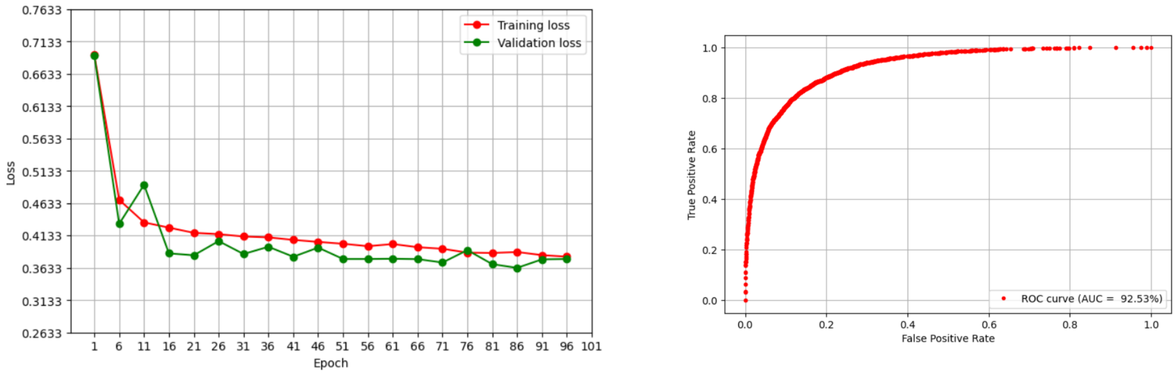
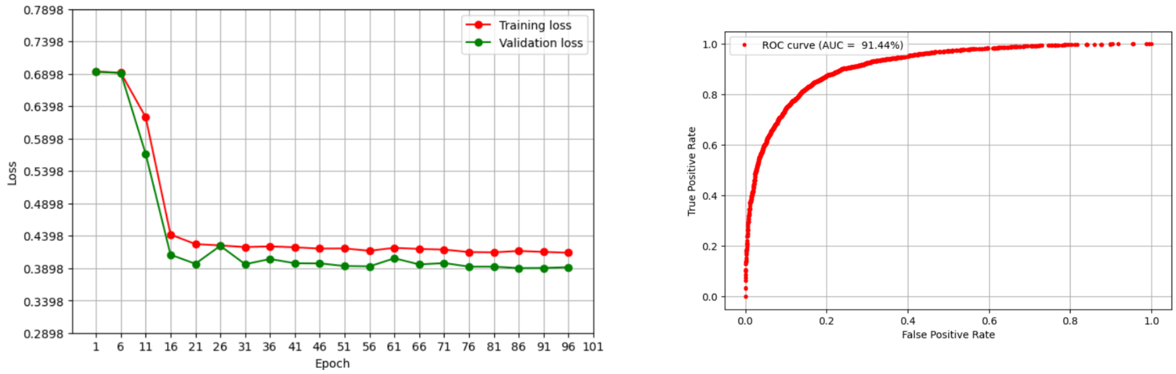


Figure 7: SGD, No Scheduler, GELU



We see that for the dense network, the base model presents an interesting behavior.
The validation set loss goes up and down throughout the training. I increased the number of epoch from 60 to 100 in order to confirm that this effect would slowly die down as it did. Nevertheless, it is not a desired property and we should consider another model.

We notice that with a different activation function this effect is not present, and the learning curve plateaus despite the large number of epoch and therefore we conclude that this model is good.

Figure 8: Adam, No Scheduler, ReLU

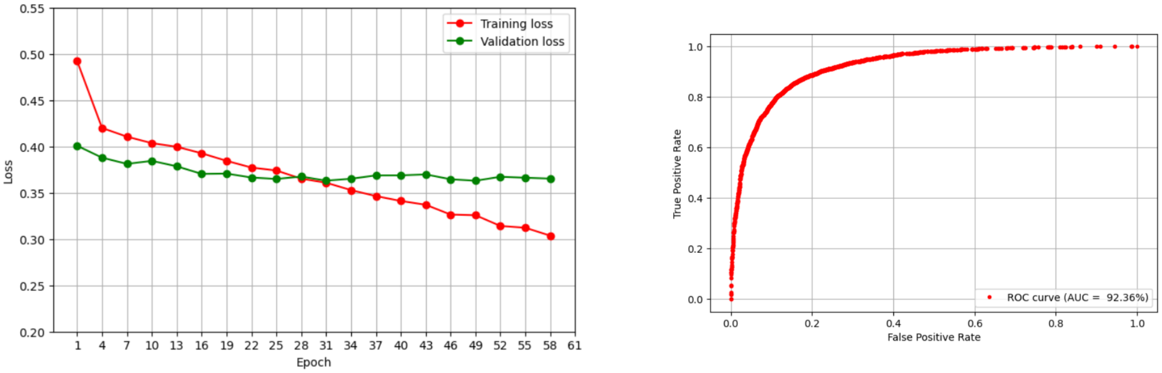
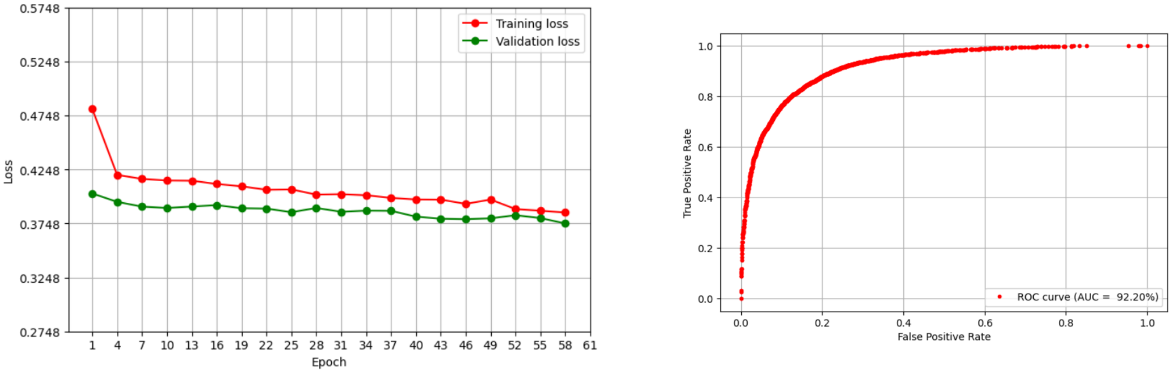
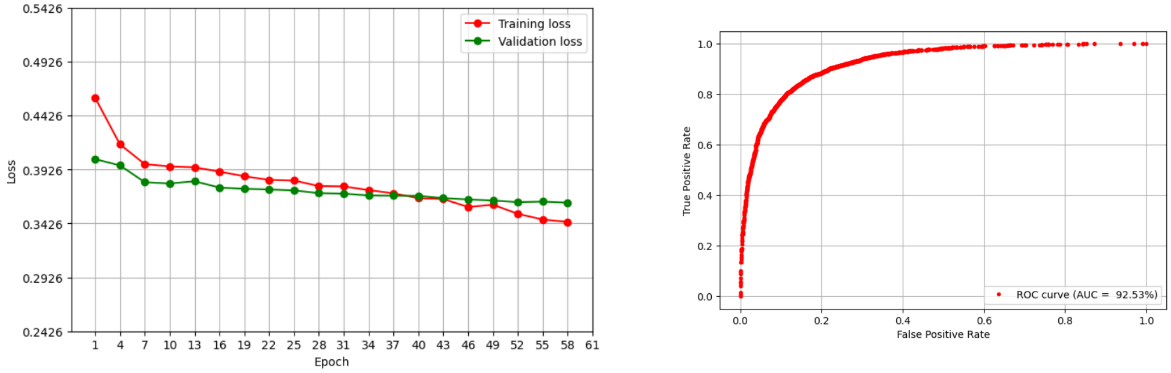


Figure 9: Adam, No Scheduler, GELU



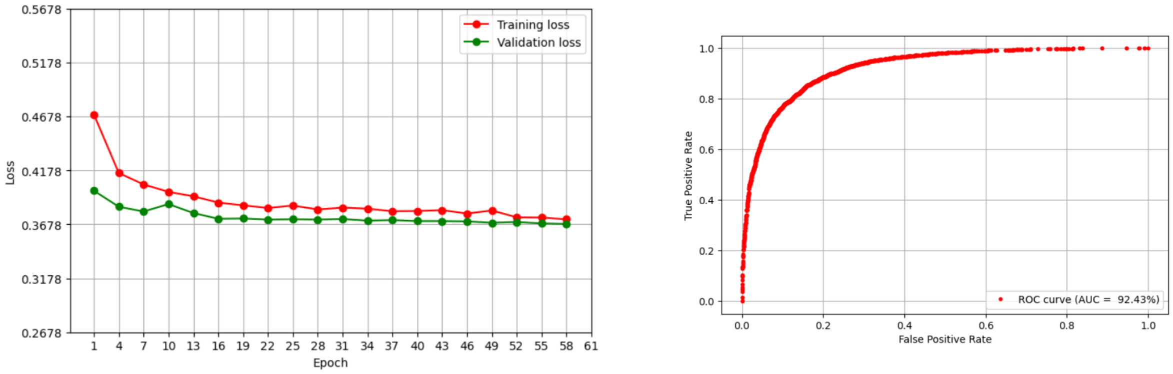
This case is completely identical to the respective one for the one-layer network. This shows that activation functions play a huge role, regardless of the number and shape of the hidden layers in a network.

Figure 10: Adam, CyclicLR



Since a plateau is reached very quickly, early stopping would have saved both time and resources and should have been used.
Overall, it is a good fit, but there is a possibility of overfitting if more epochs were used.

Figure 11: Adamax, LinearLR



The learning curve indicates that the model is a good fit, but it is important to note that good performance on a training set does not guarantee good generalization. Nevertheless, we can be optimistic that the model will generalize well on unknown datasets.

Testing

Unknown movies data split

Created a train, test set pair with the property that every review on the test set is for a movie that doesn't exist on the train set.

I chose two models, one with a single layer and one dense, that from my observations seemed to not overfit and generalize well. We train and test both models on the above datasets and we get the following results:

	Accuracy	Precision	Recall	F1-Score
One Layer	84.43%	83.05%	87.11%	85.03%
Dense	84.10%	83.88%	85.03%	84.45%

Figure 12: One Layer Network

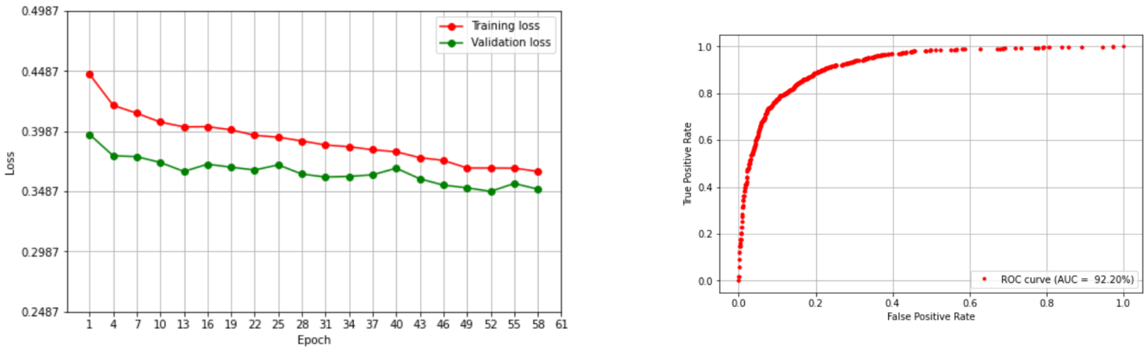
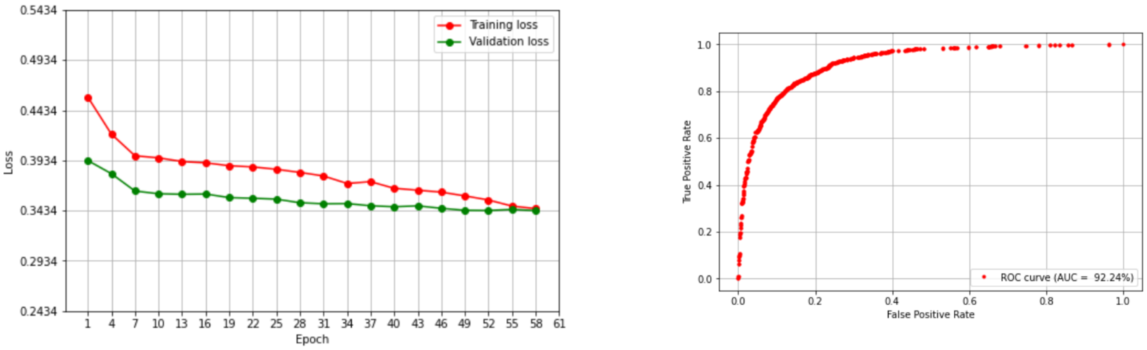


Figure 13: Dense Network



We observe that the models chosen have not been overfit to any data set used during the tuning process. We get excellent learning curves and every metric on the test set is about where it was expected from the respective results during tuning.