Course: "Advanced Machine Learning"
Classifying " IMDB" data set
"Mohamed Megahed"
date: "February 2, 2020"

This Report summarizes the results of the R codes to build models for IMDB classification.
IMDB is a set of 50,000 highly polarized reviews from the Internet Movie Database. They are split into 25,000 reviews for training and 25,000 reviews for testing, each set consisting in 50% negative and 50% positive reviews.
The original network was based on 2 hidden layers, now we try to use 1 and 3 hidden layers and see how it affects validation and test accuracy.
The next figures showing a brief summary "validation accuracy and test accuracy" about the results we get by building different models with different hyperparameters

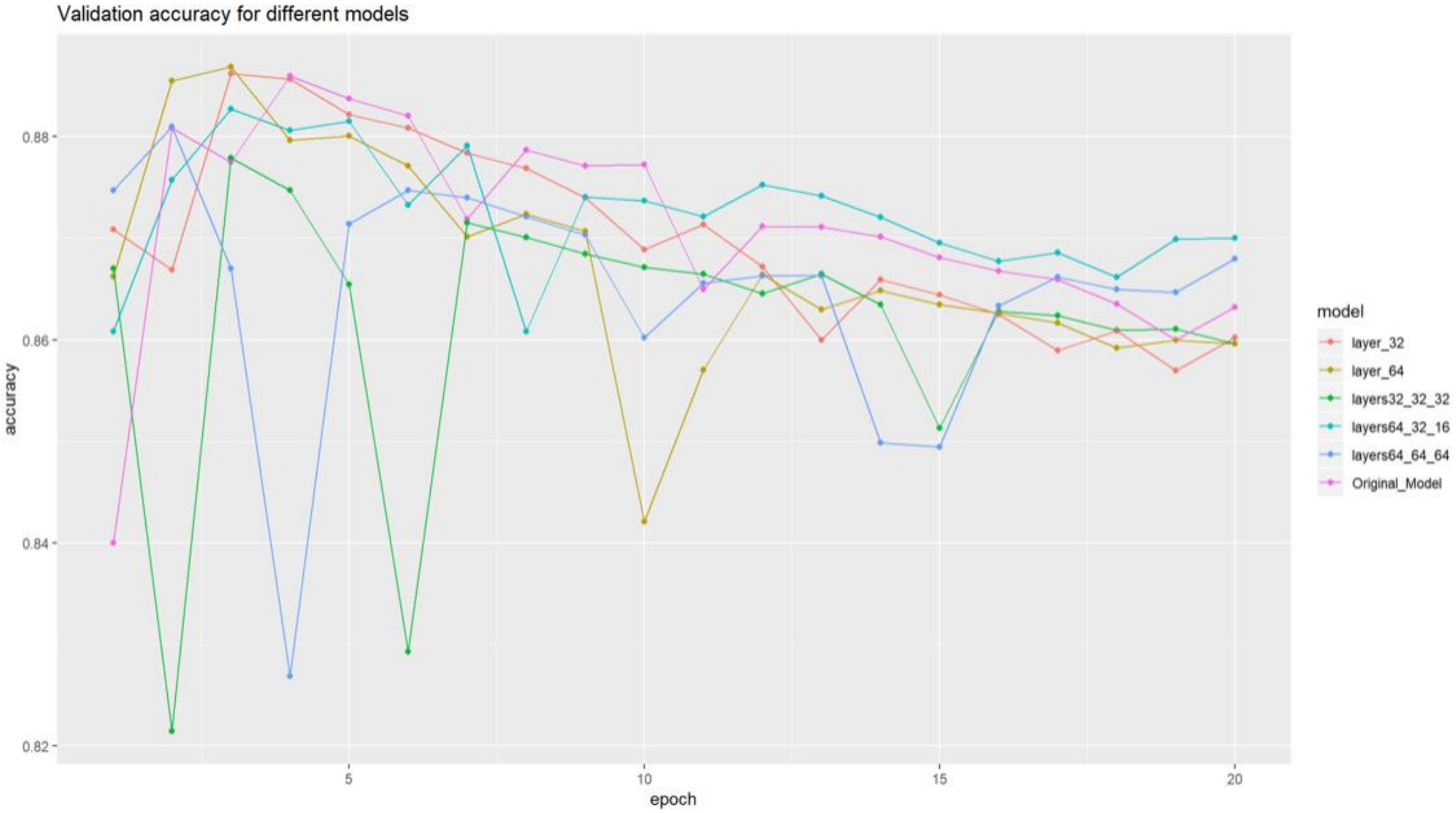
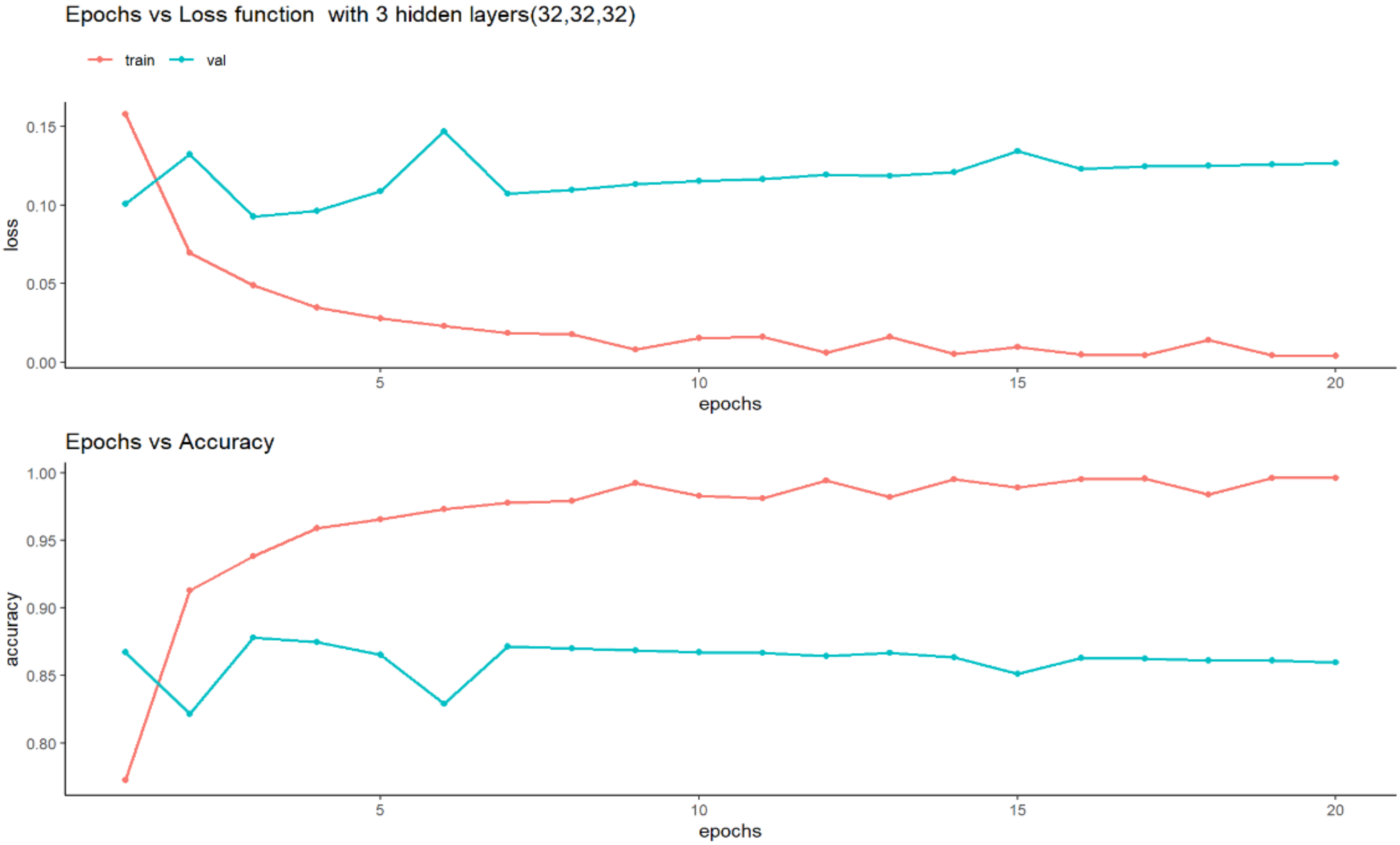


Figure (1)

1. According to the above figure (1) we can notice that the almost all models are reaching a high validation accuracy at epoch 2, 3 or 4 then the val. Accuracy starts to drop down as the model starts to overfit for the training data.

We start our analysis as follows:

We Build a bigger neural network model with 3 Layers 32 units, 20 epochs, and a batch size of 512


Epochs vs Loss function with 3 hidden layers(32,32,32)

Epochs vs Accuracy

Figure (2)

The bigger network (3 layers) starts overfitting almost right away, after just one epoch, and overfits much more severely. Its validation loss is also noisier.

As you can see, the bigger network gets its training loss near zero very quickly. The more capacity the network has, the quicker it will be able to model the training data (resulting in a low training loss), but the more susceptible it is to overfitting (resulting in a large difference between the training and validation loss).

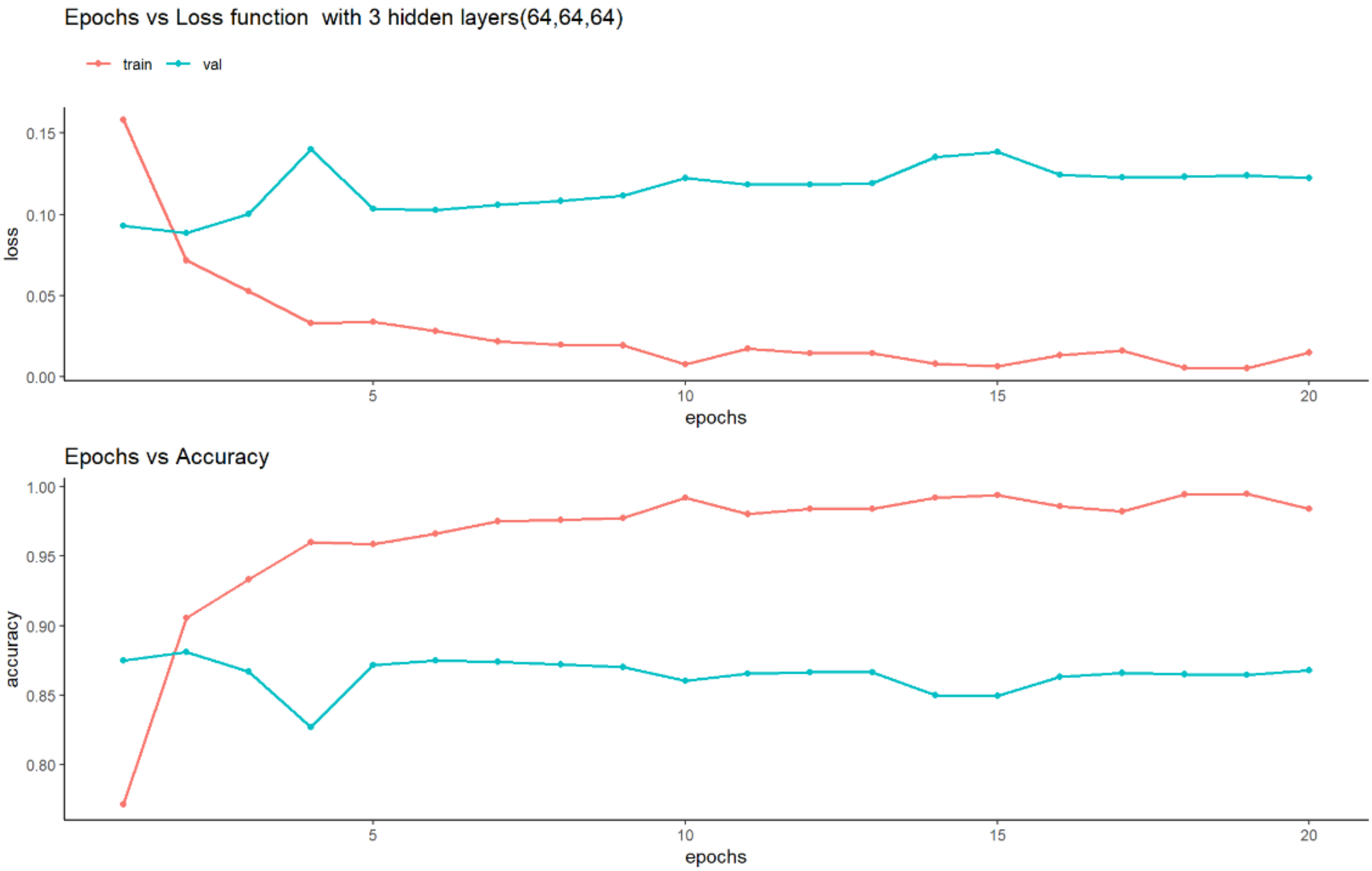
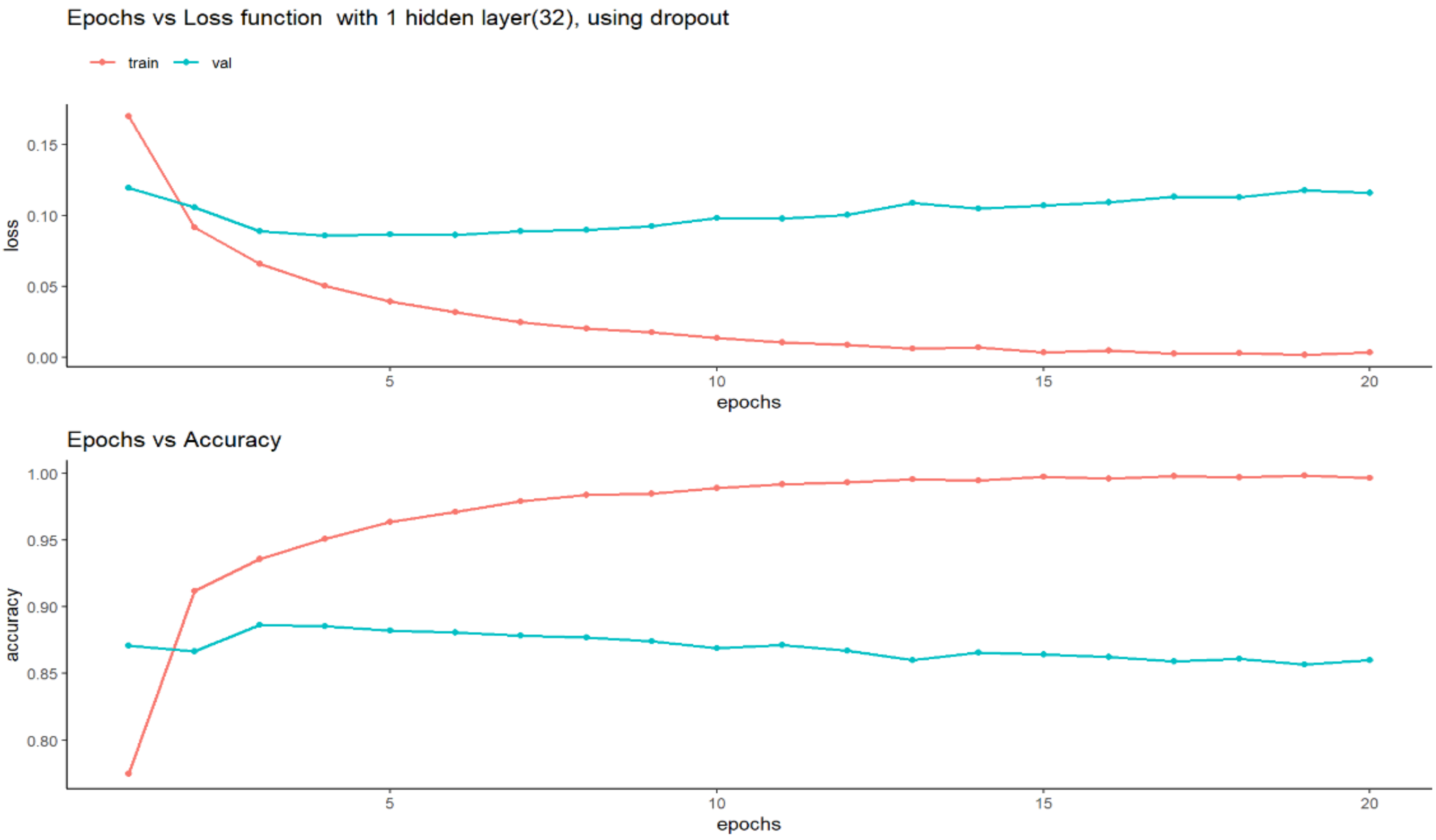


Figure (3)

In this chart we changed the number of units it has a high-test accuracy 88.5% but still the model overfitting and the validation accuracy is not consistent along many epochs

The simplest way to prevent overfitting is to reduce the size of the model, i.e. the number of learnable parameters in the model (which is determined by the number of layers and the number of units per layer). In deep learning, the number of learnable parameters in a model is often referred to as the model's "capacity". Intuitively, a model with more parameters will have more "memorization capacity" and therefore will be able to easily learn a perfect dictionary

Let's add a dropout layer in our IMDB network to see how well they do at reducing overfitting using 32 units:



We Observed that this model with only one layer with 32 units tend to perform well in both the validation accuracy consistency and has a high accuracy on test data 88.1%. Hence this model is my best model.

On the other hand, if the network has limited memorization resources, it will not be able to learn this mapping as easily, and thus, in order to minimize its loss, it will have to resort to learning compressed representations that have predictive power regarding the targets -- precisely the type of representations that we are interested in.

Here is a table summarizing the test accuracy for the different models:

# Hidden Layers	*Units*	*epoch*	*Test accuracy*
Two Layers	16,16	4	0.87828
One Layer	32	4	0.88144
	64	3	0.86068
Three Layers	64, 32, 16	4	0.87988
	64, 64, 64	2	0.88516
	32, 32, 32	5	0.87168