Neural Networks with IMDB Dataset

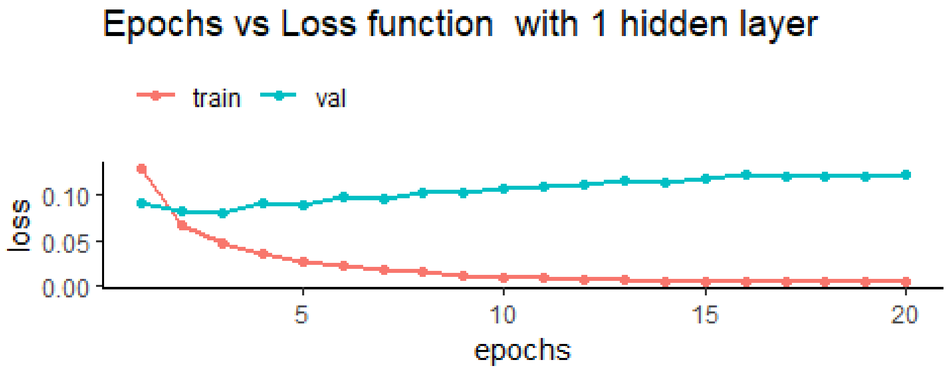
Mallikarjun Sasnur

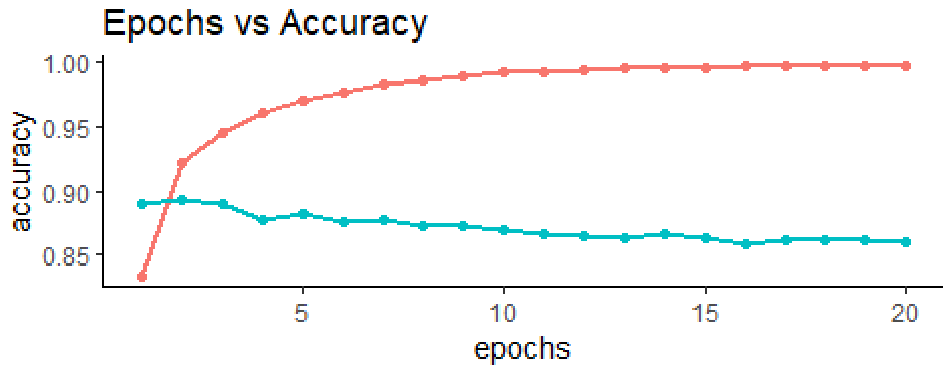
2/6/2020

Initially we import IMDB Dataset from keras library. This Dataset contains set of 50,000 reviews. Then we split into training and testing set with 25,000 reviews each and containing equal number of negative and positive review in each set.

```
imdb <- dataset_imdb(num_words = 10000)
#spliting the data into train and test
c(c(train_data, train_labels), c(test_data, test_labels)) %<-% imdb
```

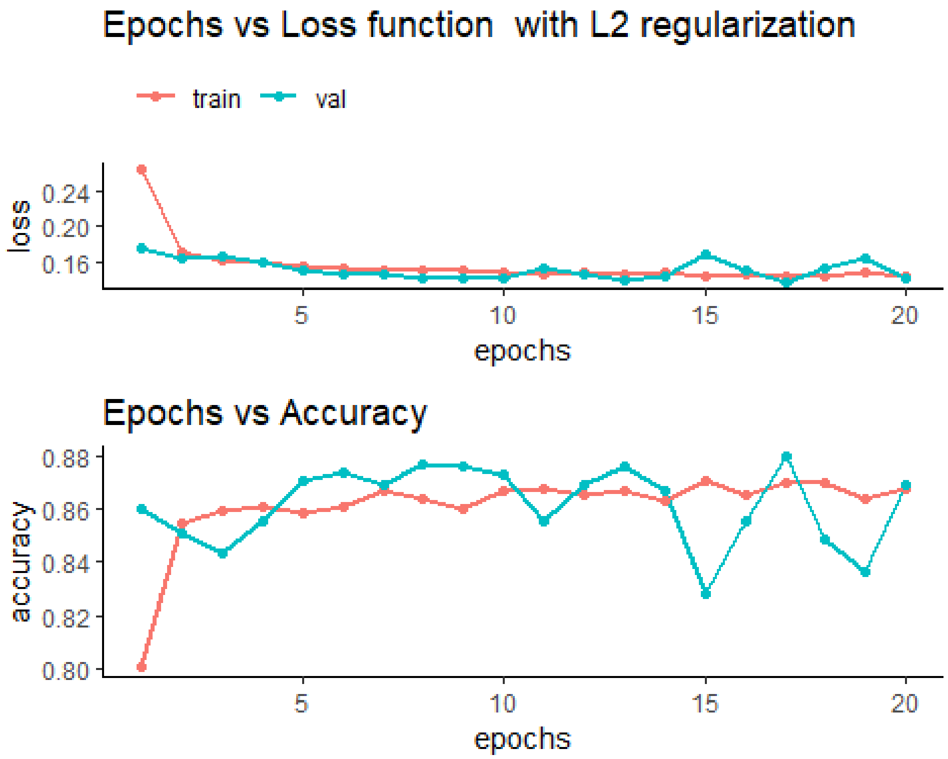
We build our first model with only single hidden layer. This layer is built with tanh activation and 32 units





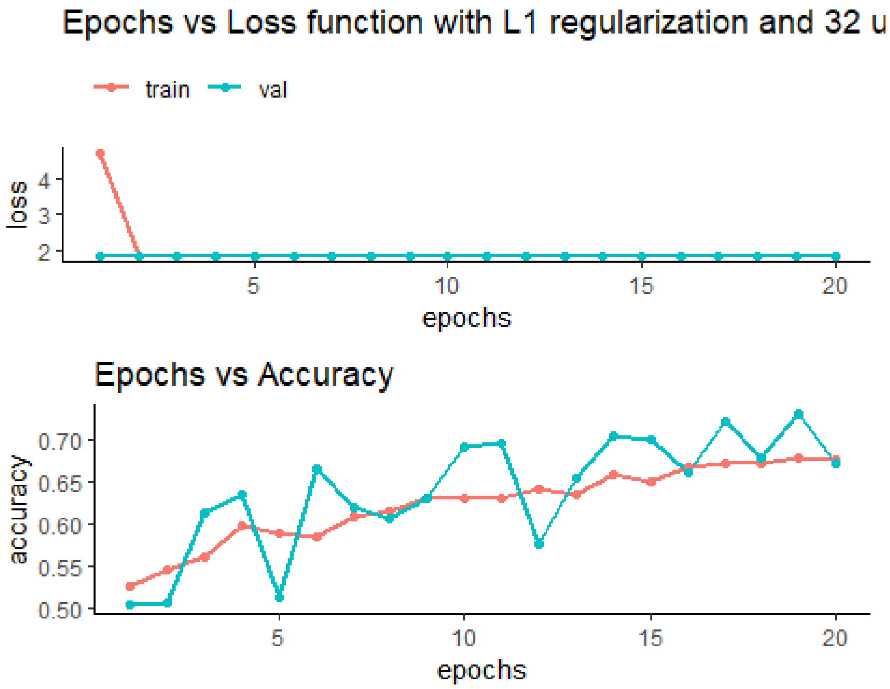
We can observe from the above plot that training set has more accuracy than validation set. So, the model overfits and needs further tuning.

We build our second model same as first one with only single hidden layer. This layer is also built with tanh activation, and 32 units. But we add l2 form regularization.



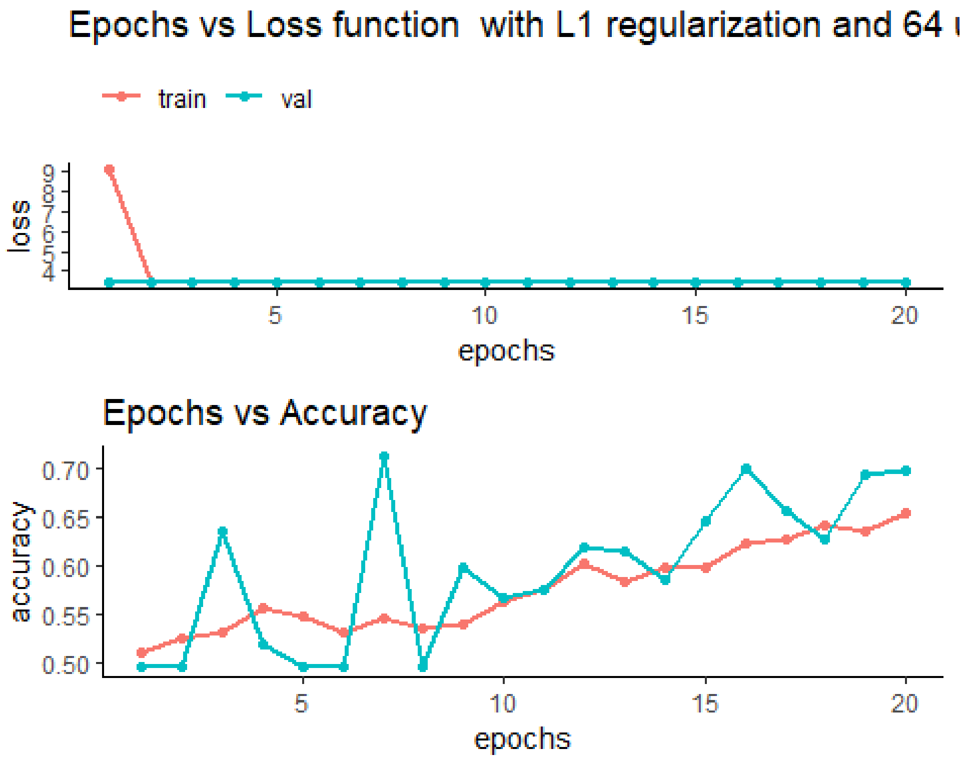
We can observe from the above plot that it overfits even with l2 form regularization also.

3) We build our third model like second one with only single hidden layer. This layer is also built with tanh activation, and 32 units. But we add l1 form regularization instead of l2 form.

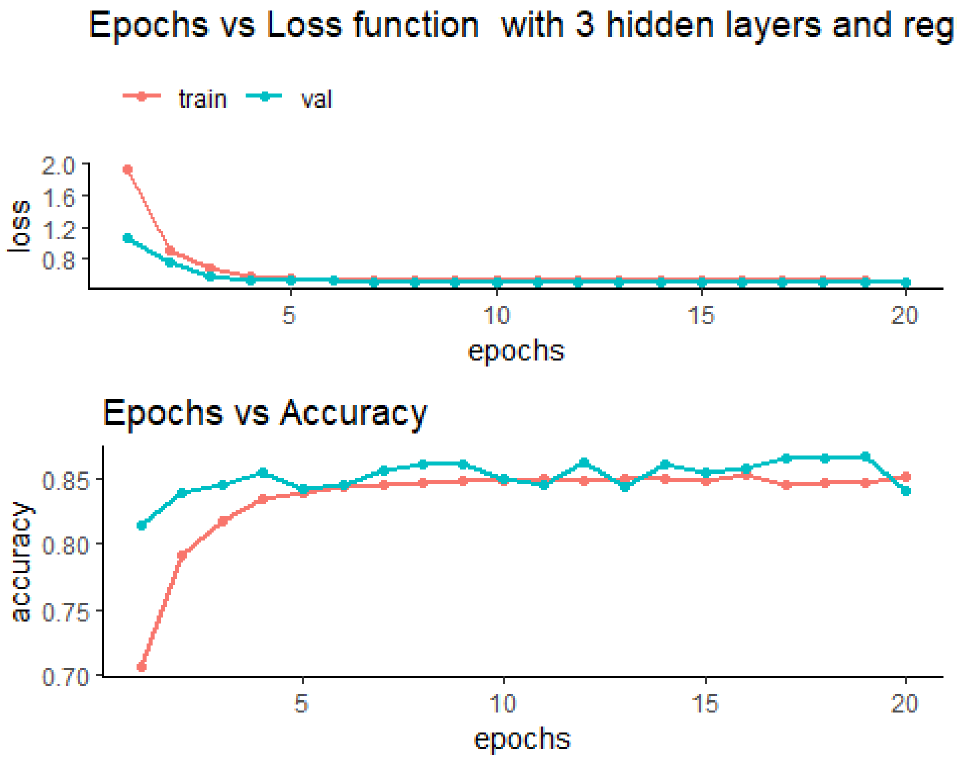


Here, we can observe that this model seems to perform well.

Now, we will try to fine tune the model by inputting different values to weights.



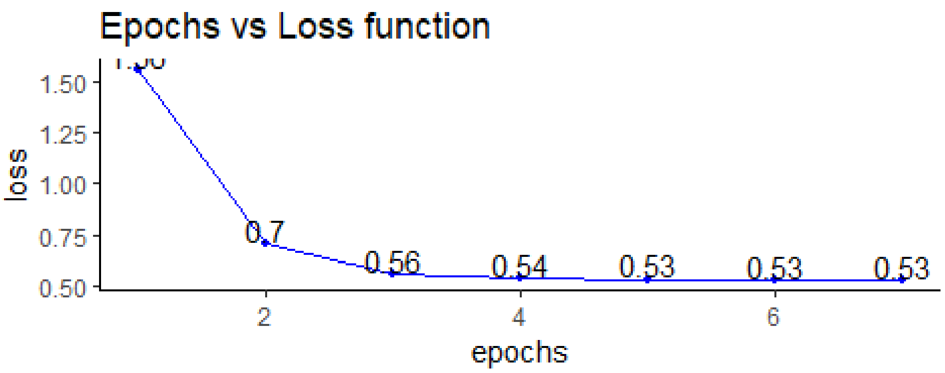
We add more hidden layers to the model. So that we can verify if the model performs well or not .

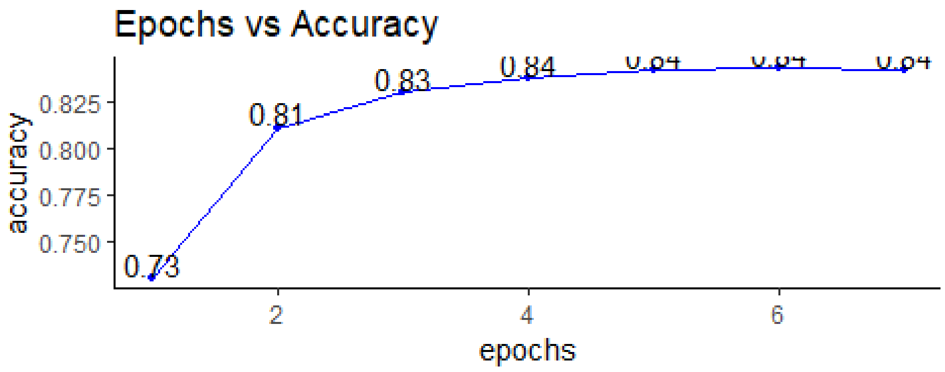

Epochs vs Loss function with 3 hidden layers and reg

Epochs vs Accuracy

We can observe from the above plot that with using 32- units and 64-units combination we get better performing model. So, we will be choosing this above model as best one.

Now we build a model using tuned parameters to evaluate the test data. Three hidden layers, 64 units in each layer, batch size set to 250 and epoch value as 7. We choose epoch value as 7 because we can observe from the above plot that validation started peaking at 7th epoch.





```
result<- best_model %>% evaluate(x_test,y_test)
result
```

```
## $loss
## [1] 0.5209477
##
## $accuracy
## [1] 0.84528
```