Deep Learning

Assignment 3

Classification Problem

Abstract

To perform multi class classification on the dataset UCI car dataset and to evaluate against various optimizers and losses. Finally the best model is chosen and is regularized to get more optimized model if necessary.

Problem Statement

The car dataset has 1728 instances and each instance has 6 attributes. These instances are classified against four classes based on the 6 attributes. The instances are words that are to be preprocessed and then classified. Finally various optimizers and losses are chose and the efficient model is chose. This one is then regularized if needed for a better model.

Proposed Solution

The dataset is chose and is preprocessed. This step includes converting the data to numerical, splitting the data into test and train, splitting the training data further to validate and training, popping the labels out and finally encoding the label.

Then various loss functions are evaluated. They are

Loss function used:

Binary cross entropy – This loss builds up an entropy and measures the difference between the two probability distributions. It is mostly used in binary classification but since there are only 4 classes in this dataset, it is used here too.

Categorical cross entropy – This loss is similar to a cross entropy or a softmax function. It predicts the difference between the actual and predicted labels across all the instances.

Hinge loss – Hinge loss focuses more on the enforcement of distribution in classification. It determines a margin beside the dividing boundary in a classification and adds penalty to the elements that cross these margins. By this way it makes the classification better.

Squared hinge loss - This loss is similar to the hinge loss but squares the value obtained by the hinge loss. The aim of this loss is to smoothen the curve of the hinge loss.

KL divergence – Kullback – Leibler divergence also known as relative entropy is a ratio between the distributions given. It is a combination of both entropy and cross entropy. When the entropy of different instances are same, it is equal to cross entropy.

Optimizers used:

Rmsprop – It is a gradient based optimization technique to optimize the neural network model. It decreased the gradient that increases to stop from exploding and increases the gradient that decreases to prevent from vanishing.

Sgd – At each iteration, this optimizer performs a parameter update. It finds the local minima and updates it to the parameters frequently to get the better results. One demerit of this optimizer is the frequent updates in the variance can cause a fluctuation in the loss function.

Adam – It is an alternative to sgd optimizer. It combines the properties of rmsprop and adagrad. It keeps the exponentially decaying average of the past gradients. One advantage of this optimizer is it converges faster and easily.

Adagrad – This optimizer is better than optimizers like rmsprop as it includes another hyper parameter in addition to the existing ones called the learning rate. Learning rate can be defined as the update or modification given to a parameter once there is a change in the gradients. This parameter makes a less update for a frequent parameter and a big update for a less frequent parameter.

Regularization

Regularization includes certain methods followed to prevent the data from overfitting and under fitting once the correct loss functions and optimizers are determined.

The techniques used are:

Dropout – After each iteration the weight of the input or the parameter is decayed or decreased to prevent overfitting.

Batch Normalization – In this case the output of each layer of the neural network is normalized in batches to give a scalable input to the following layers.

Ensemble classifier – Finally multiple models are trained on parts of the data and all the models are combined to obtain an averaged accuracy of all the models. By this method many incompletely trained models are combined to form a completely trained model.

Evaluation metrics

Rmsprop and binary_crossentropy

Hyper parameter change:

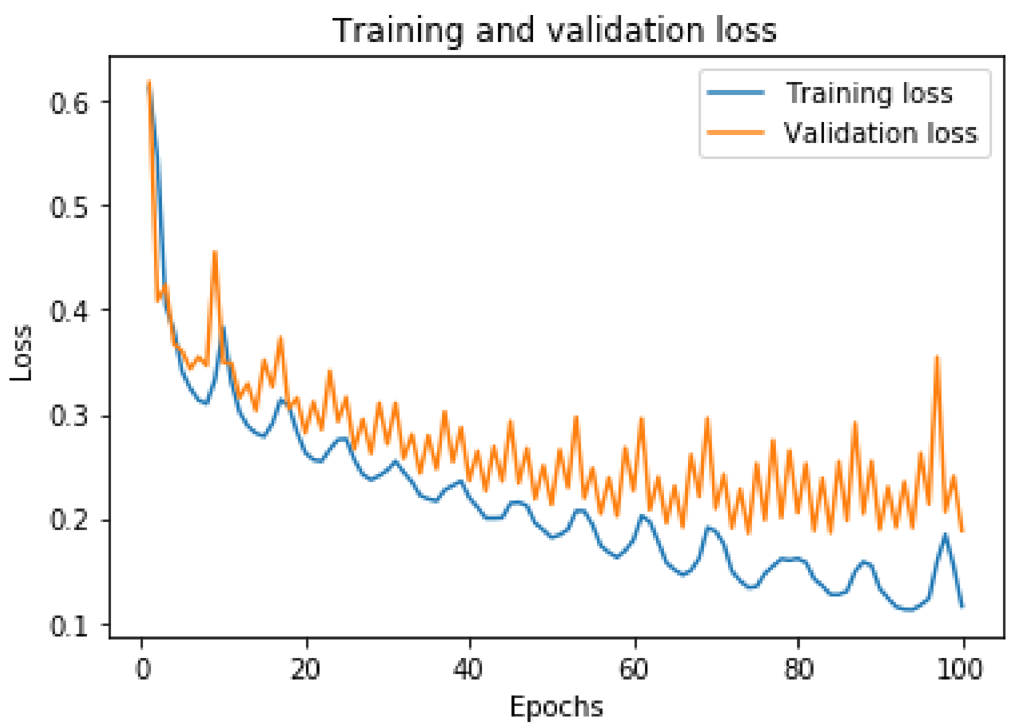
Layers = 4

Epoch = 100

Hidden units = 256

```
Epoch 96/100
```

```
210/210 [==============================] - 0s 24us/step - loss: 0.1231 - a
ccuracy: 0.9357 - val_loss: 0.2136 - val_accuracy: 0.8975
Epoch 97/100
210/210 [==============================] - 0s 28us/step - loss: 0.1585 - a
ccuracy: 0.9286 - val_loss: 0.3546 - val_accuracy: 0.8720
Epoch 98/100
210/210 [==============================] - 0s 24us/step - loss: 0.1845 - a
ccuracy: 0.9107 - val_loss: 0.2063 - val_accuracy: 0.9032
Epoch 99/100
210/210 [==============================] - 0s 24us/step - loss: 0.1540 - a
ccuracy: 0.9357 - val_loss: 0.2400 - val_accuracy: 0.8938
Epoch 100/100
210/210 [==============================] - 0s 24us/step - loss: 0.1159 - a
ccuracy: 0.9393 - val_loss: 0.1883 - val_accuracy: 0.9133
```




```
Accuracy:
518/518 [==============================] - 0s 17us/step
[0.19525637238872556, 0.9074710512161255]
```

Rmsprop and categorical cross entropy

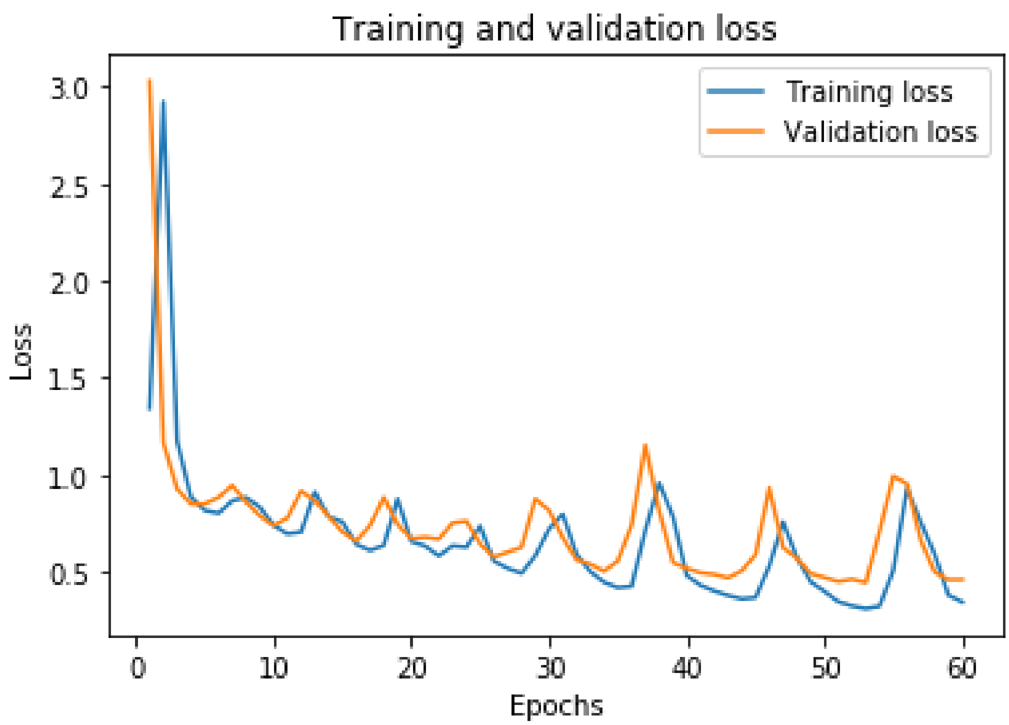
Hyper parameter change

Layers = 5

Epoch = 60

Hidden units = 512

```
Epoch 56/60
210/210 [==============================] - 0s 95us/step - loss: 0.9300 - accu
racy: 0.5619 - val_loss: 0.9538 - val_accuracy: 0.7210
Epoch 57/60
210/210 [==============================] - 0s 90us/step - loss: 0.7565 - accu
racy: 0.7571 - val_loss: 0.6622 - val_accuracy: 0.7450
Epoch 58/60
210/210 [==============================] - 0s 90us/step - loss: 0.5928 - accu
racy: 0.7619 - val_loss: 0.5002 - val_accuracy: 0.7680
Epoch 59/60
210/210 [==============================] - 0s 90us/step - loss: 0.3807 - accu
racy: 0.8238 - val_loss: 0.4603 - val_accuracy: 0.8040
Epoch 60/60
210/210 [==============================] - 0s 90us/step - loss: 0.3448 - accu
racy: 0.8905 - val_loss: 0.4609 - val_accuracy: 0.8030
```

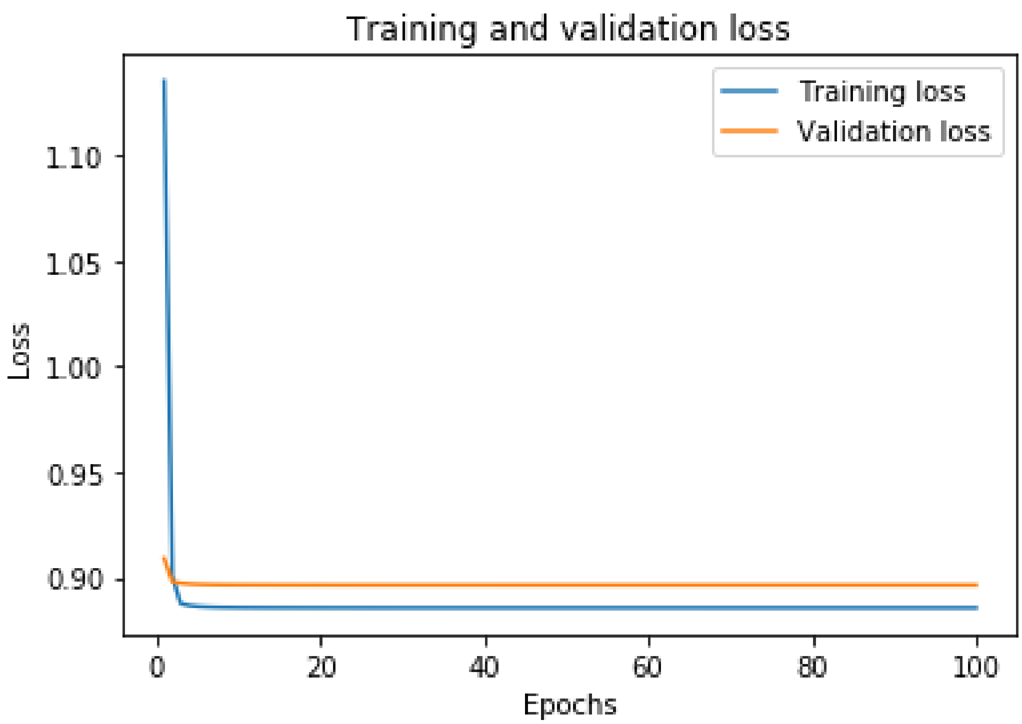



```
Accuracy
518/518 [==============================] - 0s 31us/step
[0.4253678929391515, 0.8243243098258972]
```

Rmsprop and hinge

```
Epoch 96/100
210/210 [==============================] - 0s 24us/step - loss: 0.8857 - a
ccuracy: 0.7286 - val_loss: 0.8965 - val_accuracy: 0.7070
```

```
Epoch 97/100
210/210 [==============================] - 0s 28us/step - loss: 0.8857 - a
ccuracy: 0.7286 - val_loss: 0.8965 - val_accuracy: 0.7070
Epoch 98/100
210/210 [==============================] - 0s 28us/step - loss: 0.8857 - a
ccuracy: 0.7286 - val_loss: 0.8965 - val_accuracy: 0.7070
Epoch 99/100
210/210 [==============================] - 0s 28us/step - loss: 0.8857 - a
ccuracy: 0.7286 - val_loss: 0.8965 - val_accuracy: 0.7070
Epoch 100/100
210/210 [==============================] - 0s 28us/step - loss: 0.8857 - a
ccuracy: 0.7286 - val_loss: 0.8965 - val_accuracy: 0.7070
```

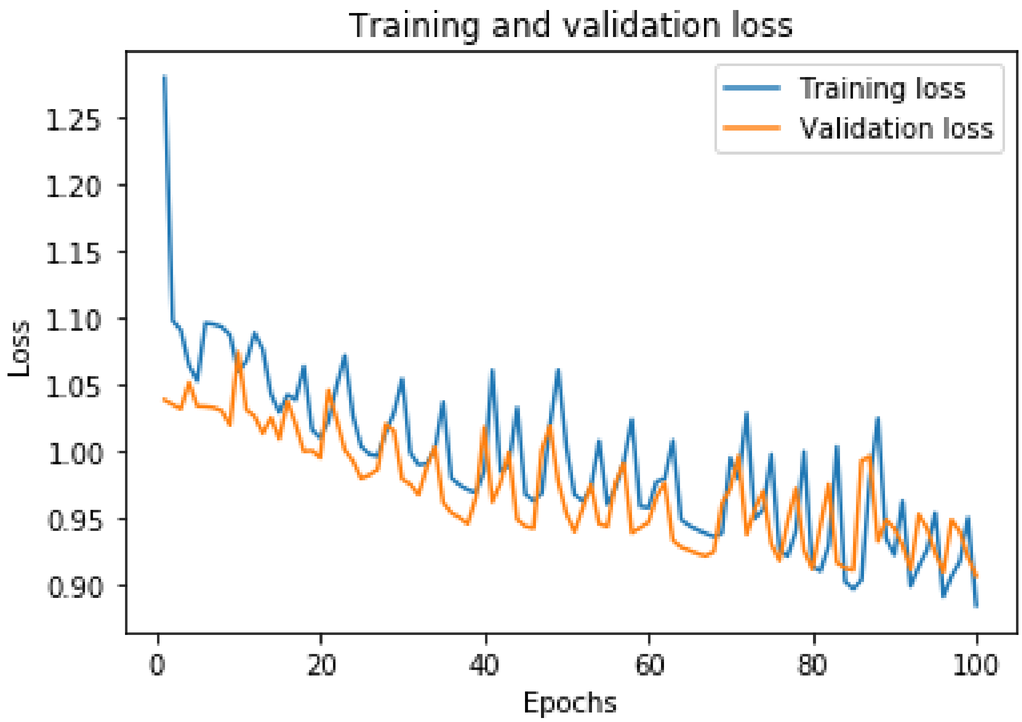



```
Accuracy
518/518 [==============================] - 0s 19us/step
[0.9121622615799481, 0.6756756901741028]
```

Rmsprop and squared hinge

```
Epoch 96/100
210/210 [==============================] - 0s 24us/step - loss: 0.8908 - a
ccuracy: 0.8905 - val_loss: 0.9092 - val_accuracy: 0.8350
Epoch 97/100
210/210 [==============================] - 0s 24us/step - loss: 0.9062 - a
ccuracy: 0.8381 - val_loss: 0.9484 - val_accuracy: 0.7970
Epoch 98/100
210/210 [==============================] - 0s 24us/step - loss: 0.9169 - a
ccuracy: 0.8333 - val_loss: 0.9394 - val_accuracy: 0.7890
```

```
Epoch 99/100
210/210 [==============================] - 0s 24us/step - loss: 0.9499 - a
ccuracy: 0.7762 - val_loss: 0.9202 - val_accuracy: 0.8270
Epoch 100/100
210/210 [==============================] - 0s 28us/step - loss: 0.8840 - a
ccuracy: 0.9048 - val_loss: 0.9061 - val_accuracy: 0.8410
```




```
Accuracy
518/518 [==============================] - 0s 19us/step
[0.9225932640458626, 0.8166022896766663]
```

Rmsprop and kl loss

```
Epoch 96/100
210/210 [==============================] - 0s 24us/step - loss: 0.2687 - a
ccuracy: 0.8429 - val_loss: 0.6225 - val_accuracy: 0.7490
Epoch 97/100
210/210 [==============================] - 0s 24us/step - loss: 0.3666 - a
ccuracy: 0.8286 - val_loss: 0.7233 - val_accuracy: 0.7620
Epoch 98/100
210/210 [==============================] - 0s 24us/step - loss: 0.4643 - a
ccuracy: 0.7905 - val_loss: 0.6662 - val_accuracy: 0.7320
Epoch 99/100
210/210 [==============================] - 0s 24us/step - loss: 0.3978 - a
ccuracy: 0.8190 - val_loss: 0.5484 - val_accuracy: 0.8000
Epoch 100/100
210/210 [==============================] - 0s 24us/step - loss: 0.2951 - a
ccuracy: 0.8524 - val_loss: 0.4591 - val_accuracy: 0.8320
```


```
Accuracy
518/518 [==============================] - 0s 19us/step
[0.4597517472896797, 0.806949779429245]
```

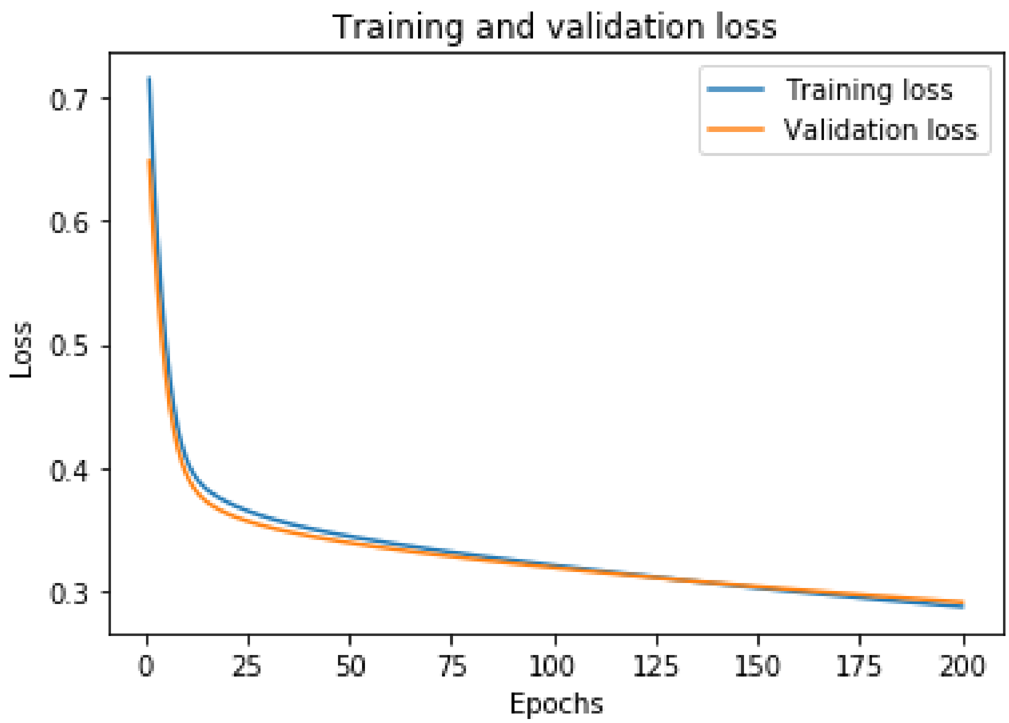
Sgd and binary cross entropy

Hyper parameter change

Hidden units = 512

Epoch = 200

```
Epoch 196/200
210/210 [==============================] - 0s 57us/step - loss: 0.2885 - a
ccuracy: 0.8738 - val_loss: 0.2920 - val_accuracy: 0.8790
Epoch 197/200
210/210 [==============================] - 0s 62us/step - loss: 0.2882 - a
ccuracy: 0.8750 - val_loss: 0.2917 - val_accuracy: 0.8792
Epoch 198/200
210/210 [==============================] - 0s 66us/step - loss: 0.2879 - a
ccuracy: 0.8750 - val_loss: 0.2915 - val_accuracy: 0.8790
Epoch 199/200
210/210 [==============================] - 0s 66us/step - loss: 0.2876 - a
ccuracy: 0.8750 - val_loss: 0.2913 - val_accuracy: 0.8792
Epoch 200/200
210/210 [==============================] - 0s 62us/step - loss: 0.2874 - a
ccuracy: 0.8762 - val_loss: 0.2910 - val_accuracy: 0.8790
```




```
Accuracy
518/518 [==============================] - 0s 25us/step
[0.3145194946568905, 0.86583012342453]
```

Sgd and categorical cross entropy

Hyper parameter change

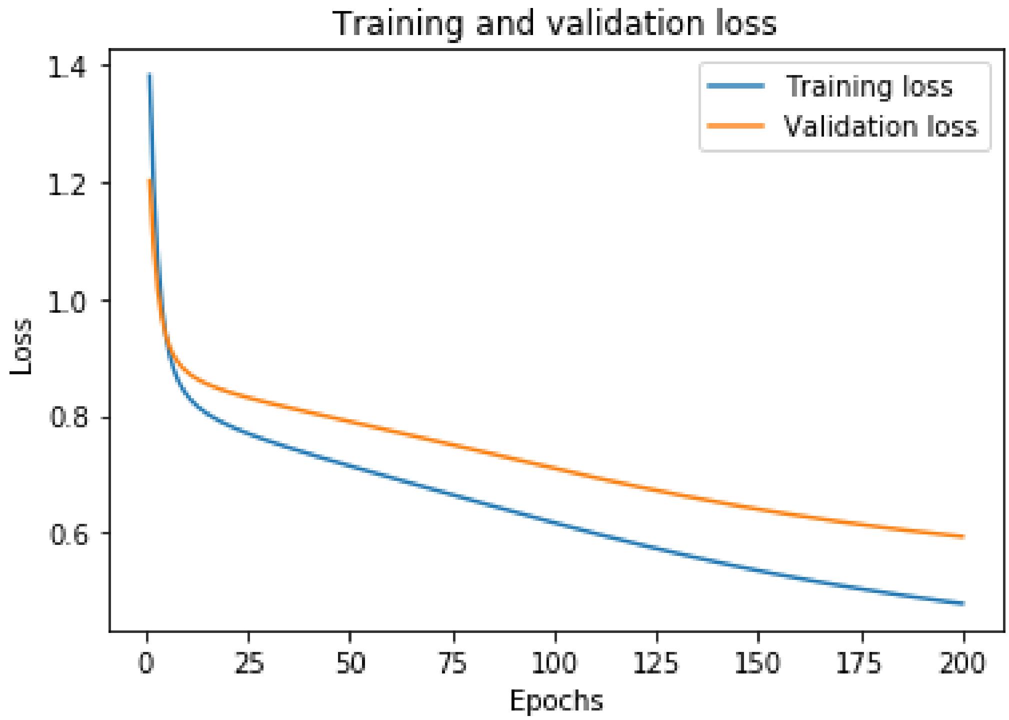
5 layers

512 hidden units

200 epoch

```
Epoch 196/200
210/210 [==============================] - 0s 85us/step - loss: 0.4817 - a
ccuracy: 0.8286 - val_loss: 0.5959 - val_accuracy: 0.7810
Epoch 197/200
210/210 [==============================] - 0s 85us/step - loss: 0.4808 - a
ccuracy: 0.8286 - val_loss: 0.5950 - val_accuracy: 0.7810
Epoch 198/200
210/210 [==============================] - 0s 81us/step - loss: 0.4798 - a
ccuracy: 0.8286 - val_loss: 0.5944 - val_accuracy: 0.7800
Epoch 199/200
210/210 [==============================] - 0s 81us/step - loss: 0.4789 - a
ccuracy: 0.8286 - val_loss: 0.5935 - val_accuracy: 0.7830
Epoch 200/200
```

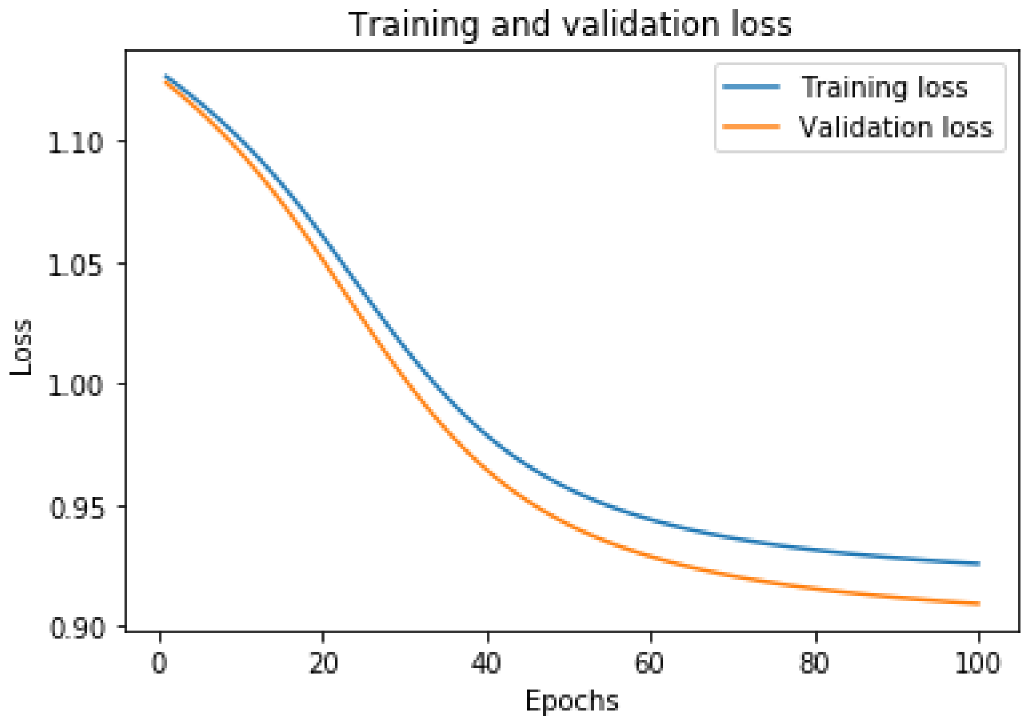
```
210/210 [==============================] - 0s 81us/step - loss: 0.4780 - a
ccuracy: 0.8286 - val_loss: 0.5929 - val_accuracy: 0.7810
```




```
Accuracy
518/518 [==============================] - 0s 31us/step
[0.6397814548153675, 0.739382266998291]
```

Sgd and hinge

```
Epoch 96/100
210/210 [==============================] - 0s 19us/step - loss: 0.9401 - a
ccuracy: 0.6714 - val_loss: 0.9251 - val_accuracy: 0.7080
Epoch 97/100
210/210 [==============================] - 0s 19us/step - loss: 0.9394 - a
ccuracy: 0.6714 - val_loss: 0.9243 - val_accuracy: 0.7080
Epoch 98/100
210/210 [==============================] - 0s 28us/step - loss: 0.9387 - a
ccuracy: 0.6714 - val_loss: 0.9236 - val_accuracy: 0.7080
Epoch 99/100
210/210 [==============================] - 0s 19us/step - loss: 0.9380 - a
ccuracy: 0.6714 - val_loss: 0.9229 - val_accuracy: 0.7080
Epoch 100/100
210/210 [==============================] - 0s 24us/step - loss: 0.9374 - a
ccuracy: 0.6714 - val_loss: 0.9222 - val_accuracy: 0.7080
```



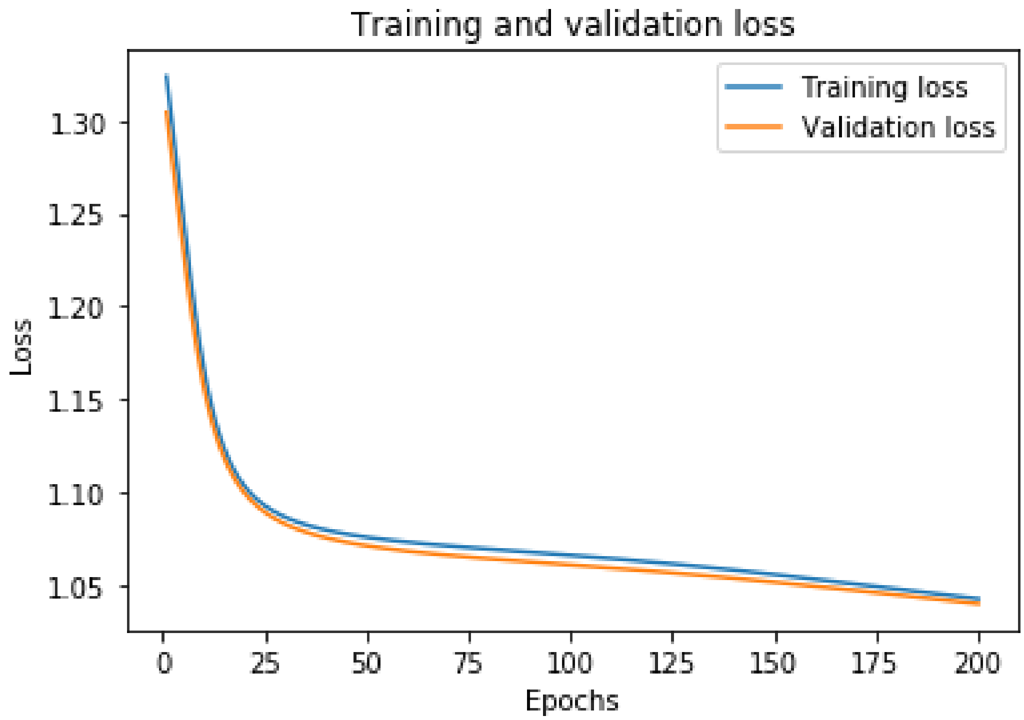

```
Accuracy
518/518 [==============================] - 0s 17us/step
[0.9146607879506115, 0.6969112157821655]
```

Sgd and square hinge

Hyper parameter change

Epoch 200

```
Epoch 196/200
210/210 [==============================] - 0s 24us/step - loss: 1.0436 - a
ccuracy: 0.6810 - val_loss: 1.0409 - val_accuracy: 0.6960
Epoch 197/200
210/210 [==============================] - 0s 19us/step - loss: 1.0434 - a
ccuracy: 0.6810 - val_loss: 1.0407 - val_accuracy: 0.6960
Epoch 198/200
210/210 [==============================] - 0s 24us/step - loss: 1.0431 - a
ccuracy: 0.6810 - val_loss: 1.0404 - val_accuracy: 0.6970
Epoch 199/200
210/210 [==============================] - 0s 24us/step - loss: 1.0428 - a
ccuracy: 0.6810 - val_loss: 1.0402 - val_accuracy: 0.6970
Epoch 200/200
210/210 [==============================] - 0s 19us/step - loss: 1.0426 - a
ccuracy: 0.6810 - val_loss: 1.0399 - val_accuracy: 0.6970
```




```
Accuracy
518/518 [==============================] - 0s 17us/step
[1.0257120459236233, 0.7297297120094299]
```

Sgd and kl loss

Hyper parameter change

5 layers

512 hidden units

200 epoch

128 batch size

```
Epoch 196/200
210/210 [==============================] - 0s 142us/step - loss: 0.4054 -
accuracy: 0.8476 - val_loss: 0.4801 - val_accuracy: 0.8100
Epoch 197/200
210/210 [==============================] - 0s 142us/step - loss: 0.4035 -
accuracy: 0.8429 - val_loss: 0.4773 - val_accuracy: 0.8080
Epoch 198/200
210/210 [==============================] - 0s 142us/step - loss: 0.4090 -
accuracy: 0.8571 - val_loss: 0.4768 - val_accuracy: 0.8090
Epoch 199/200
210/210 [==============================] - 0s 147us/step - loss: 0.4085 -
accuracy: 0.8333 - val_loss: 0.4801 - val_accuracy: 0.8120
Epoch 200/200
```

```
210/210 [==============================] - 0s 147us/step - loss: 0.4009 -
accuracy: 0.8476 - val_loss: 0.5391 - val_accuracy: 0.7810
```


```
Accuracy
518/518 [==============================] - 0s 27us/step
[0.5552696226646541, 0.7741312980651855
```

Adam and binary cross entropy

Hyper parameter change

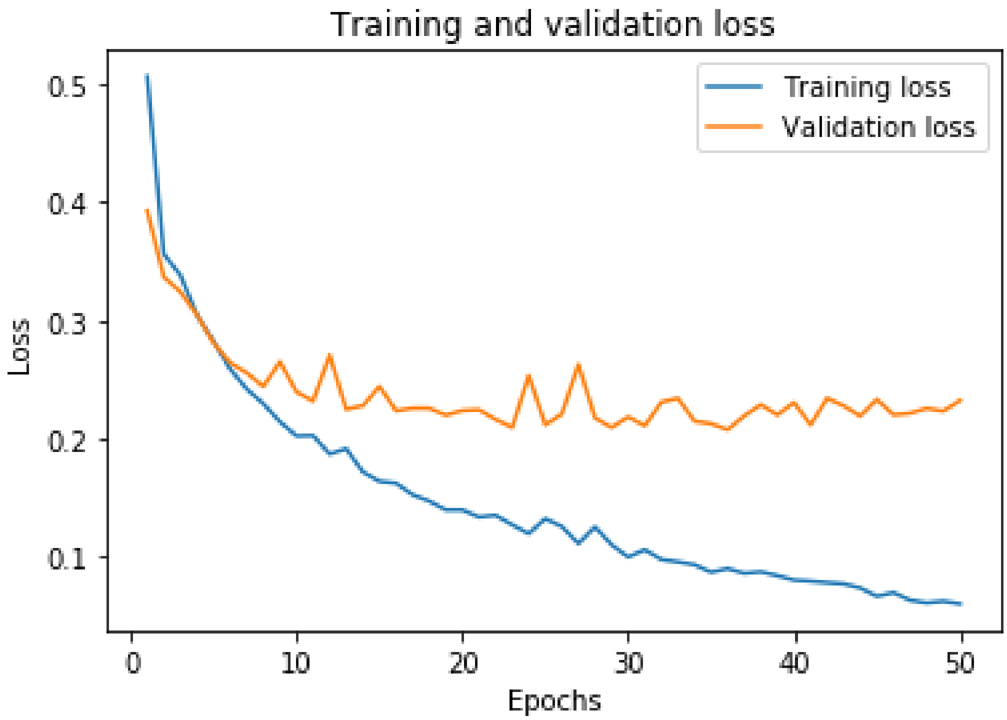
512 hidden units

50 epoch

128 batch size

```
Epoch 46/50
210/210 [==============================] - 0s 123us/step - loss: 0.0696 -
accuracy: 0.9786 - val_loss: 0.2202 - val_accuracy: 0.9208
Epoch 47/50
210/210 [==============================] - 0s 133us/step - loss: 0.0632 -
accuracy: 0.9833 - val_loss: 0.2215 - val_accuracy: 0.9200
Epoch 48/50
210/210 [==============================] - 0s 133us/step - loss: 0.0609 -
accuracy: 0.9833 - val_loss: 0.2256 - val_accuracy: 0.9193
Epoch 49/50
210/210 [==============================] - 0s 128us/step - loss: 0.0623 -
accuracy: 0.9810 - val_loss: 0.2233 - val_accuracy: 0.9175
```

```
Epoch 50/50
210/210 [==============================] - 0s 128us/step - loss: 0.0601 -
accuracy: 0.9810 - val_loss: 0.2322 - val_accuracy: 0.9202
```




```
Accuracy
518/518 [==============================] - 0s 25us/step
[0.19941595493021158, 0.9100193190574646]
```

Adam and categorical cross entropy

Hyper parameter change

5 layers

512 hidden units

70 epoch

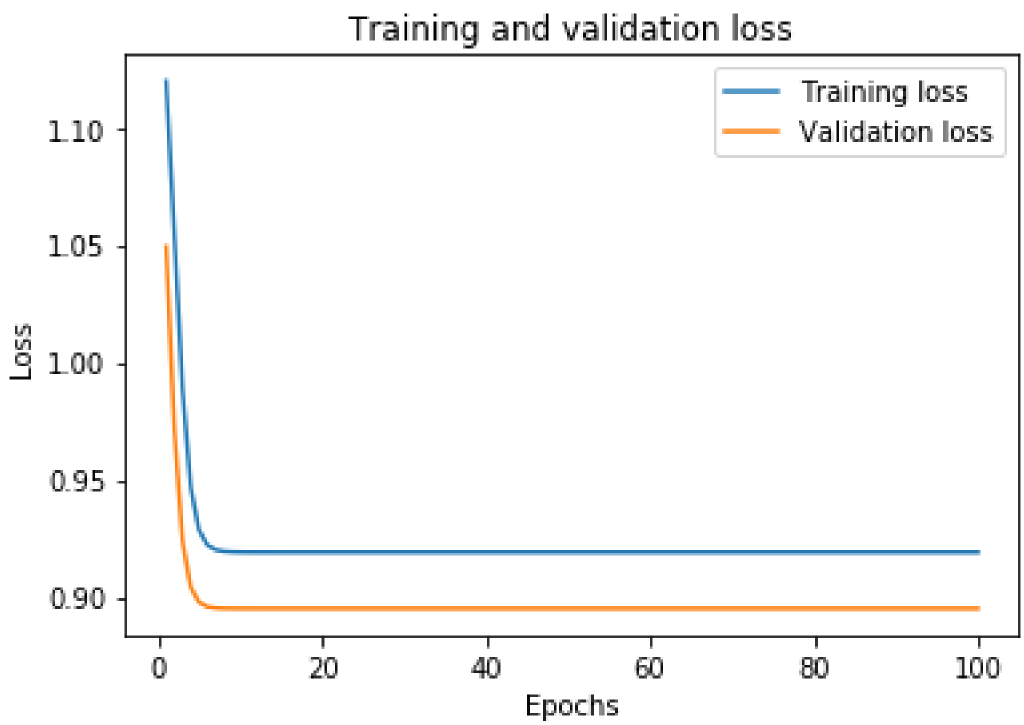
```
Epoch 66/70
210/210 [==============================] - 0s 95us/step - loss: 0.0319 - a
ccuracy: 0.9952 - val_loss: 0.7611 - val_accuracy: 0.8140
Epoch 67/70
210/210 [==============================] - 0s 104us/step - loss: 0.0325 -
accuracy: 0.9952 - val_loss: 0.8119 - val_accuracy: 0.8160
Epoch 68/70
210/210 [==============================] - 0s 100us/step - loss: 0.0320 -
accuracy: 0.9952 - val_loss: 0.7943 - val_accuracy: 0.8150
Epoch 69/70
```

```
210/210 [==============================] - 0s 95us/step - loss: 0.0279 - a
ccuracy: 0.9952 - val_loss: 0.8158 - val_accuracy: 0.8140
Epoch 70/70
210/210 [==============================] - 0s 95us/step - loss: 0.0259 - a
ccuracy: 0.9952 - val_loss: 0.8476 - val_accuracy: 0.8130
```


```
Accuracy
518/518 [==============================] - 0s 33us/step
[0.7488315112342245, 0.8359073400497437]
```

Adam and hinge

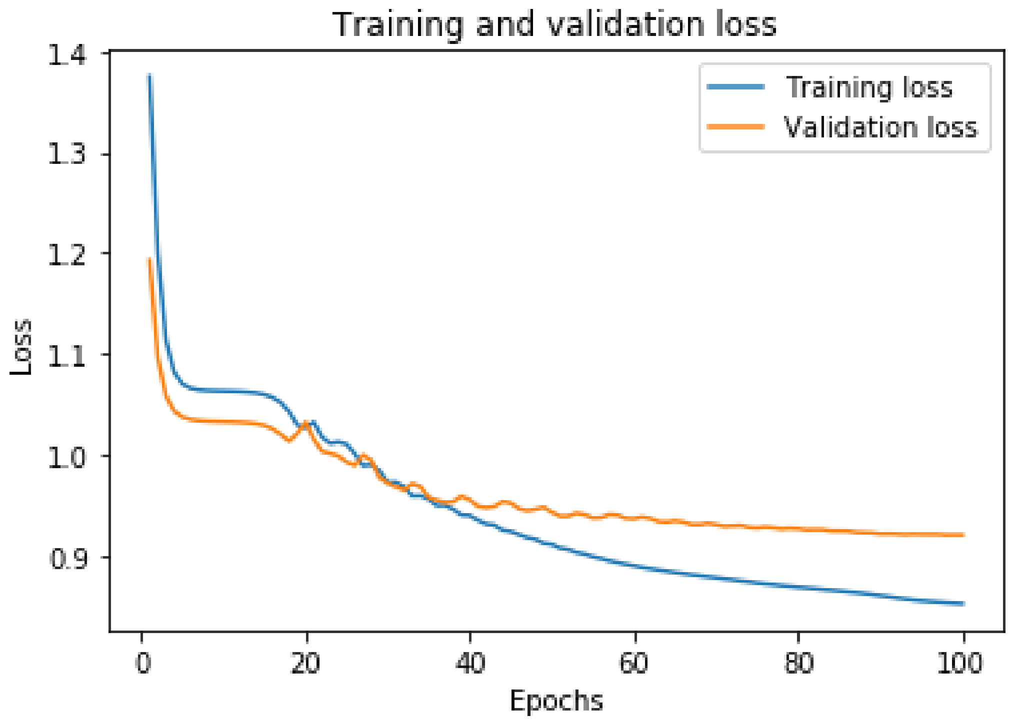
```
Epoch 96/100
210/210 [==============================] - 0s 28us/step - loss: 0.9190 - a
ccuracy: 0.6619 - val_loss: 0.8950 - val_accuracy: 0.7100
Epoch 97/100
210/210 [==============================] - 0s 24us/step - loss: 0.9190 - a
ccuracy: 0.6619 - val_loss: 0.8950 - val_accuracy: 0.7100
Epoch 98/100
210/210 [==============================] - 0s 24us/step - loss: 0.9190 - a
ccuracy: 0.6619 - val_loss: 0.8950 - val_accuracy: 0.7100
Epoch 99/100
210/210 [==============================] - 0s 24us/step - loss: 0.9190 - a
ccuracy: 0.6619 - val_loss: 0.8950 - val_accuracy: 0.7100
Epoch 100/100
210/210 [==============================] - 0s 24us/step - loss: 0.9190 - a
ccuracy: 0.6619 - val_loss: 0.8950 - val_accuracy: 0.7100
```




```
Accuracy
518/518 [==============================] - 0s 19us/step
[0.9015444015444015, 0.6969112157821655]
```

Adam and squared hinge

```
Epoch 96/100
210/210 [==============================] - 0s 28us/step - loss: 0.8544 - a
ccuracy: 0.9048 - val_loss: 0.9208 - val_accuracy: 0.8250
Epoch 97/100
210/210 [==============================] - 0s 24us/step - loss: 0.8538 - a
ccuracy: 0.9048 - val_loss: 0.9208 - val_accuracy: 0.8240
Epoch 98/100
210/210 [==============================] - 0s 24us/step - loss: 0.8532 - a
ccuracy: 0.9048 - val_loss: 0.9203 - val_accuracy: 0.8260
Epoch 99/100
210/210 [==============================] - 0s 24us/step - loss: 0.8526 - a
ccuracy: 0.9048 - val_loss: 0.9204 - val_accuracy: 0.8250
Epoch 100/100
210/210 [==============================] - 0s 24us/step - loss: 0.8521 - a
ccuracy: 0.9048 - val_loss: 0.9203 - val_accuracy: 0.8260
```




```
Accuracy
518/518 [==============================] - 0s 17us/step
[0.9375913373291722, 0.8146718144416809]
```

Adam and kl loss

Hyper parameter change

3 layers

512 hidden units

60 epochs

```
Epoch 56/60
210/210 [==============================] - 0s 71us/step - loss: 0.1098 - accu
racy: 0.9714 - val_loss: 0.5108 - val_accuracy: 0.8300
Epoch 57/60
210/210 [==============================] - 0s 71us/step - loss: 0.1044 - accu
racy: 0.9619 - val_loss: 0.5026 - val_accuracy: 0.8310
Epoch 58/60
210/210 [==============================] - 0s 71us/step - loss: 0.0930 - accu
racy: 0.9762 - val_loss: 0.5053 - val_accuracy: 0.8200
Epoch 59/60
210/210 [==============================] - 0s 109us/step - loss: 0.0984 - acc
uracy: 0.9714 - val_loss: 0.5181 - val_accuracy: 0.8320
Epoch 60/60
```

```
210/210 [==============================] - 0s 81us/step - loss: 0.0907 - accu
racy: 0.9762 - val_loss: 0.5161 - val_accuracy: 0.8360
```


```
Accuracy
518/518 [==============================] - 0s 25us/step
[0.5017269678759069, 0.8397683501243591]
```

Adagrad and binary cross entropy

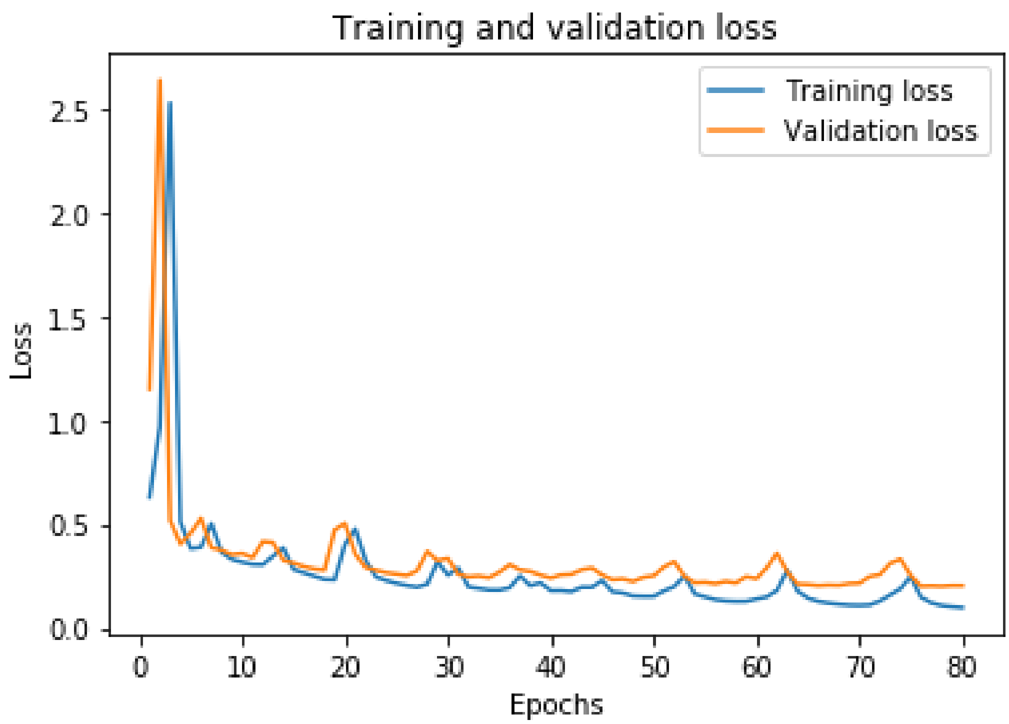
Hyper parameter change

5 layers

80 epoch

```
Epoch 76/80
210/210 [==============================] - 0s 28us/step - loss: 0.1489 - a
ccuracy: 0.9333 - val_loss: 0.2016 - val_accuracy: 0.9105
Epoch 77/80
210/210 [==============================] - 0s 28us/step - loss: 0.1219 - a
ccuracy: 0.9560 - val_loss: 0.2031 - val_accuracy: 0.9190
Epoch 78/80
210/210 [==============================] - 0s 28us/step - loss: 0.1089 - a
ccuracy: 0.9631 - val_loss: 0.2013 - val_accuracy: 0.9165
Epoch 79/80
210/210 [==============================] - 0s 28us/step - loss: 0.1031 - a
ccuracy: 0.9690 - val_loss: 0.2046 - val_accuracy: 0.9193
Epoch 80/80
```

```
210/210 [==============================] - 0s 28us/step - loss: 0.0986 - a
ccuracy: 0.9714 - val_loss: 0.2042 - val_accuracy: 0.9185
```



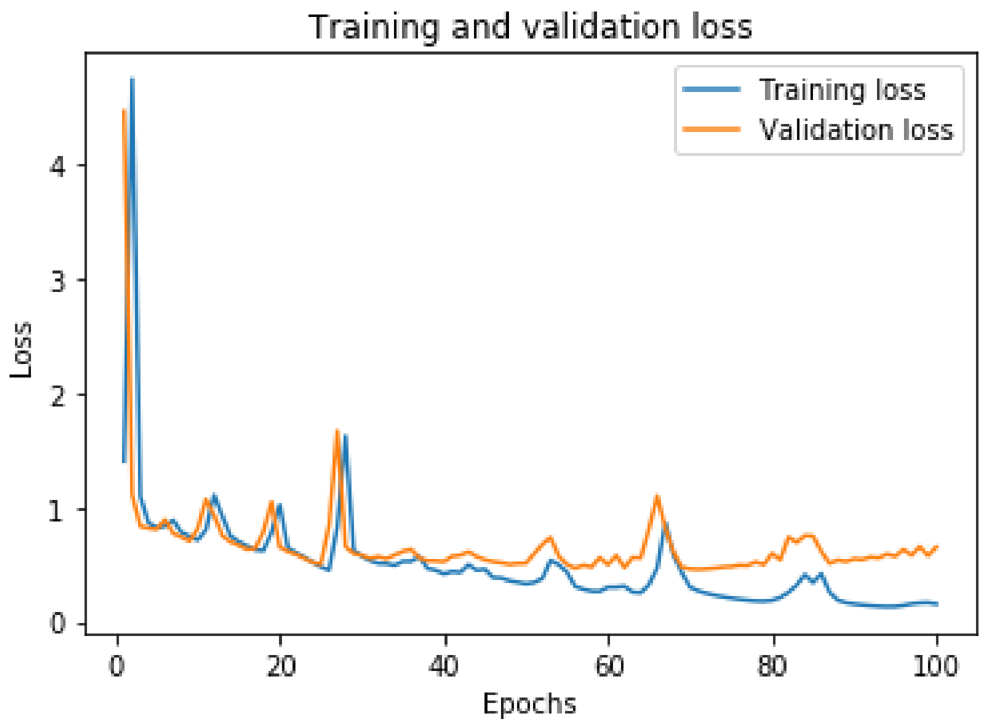

```
Accuracy
518/518 [==============================] - 0s 21us/step
[0.22147977029955065, 0.9020270109176636]
```

Adagrad and categorical cross entropy

Hyper parameter

5 layers

```
Epoch 96/100
210/210 [==============================] - 0s 33us/step - loss: 0.1431 - a
ccuracy: 0.9667 - val_loss: 0.6324 - val_accuracy: 0.8220
Epoch 97/100
210/210 [==============================] - 0s 38us/step - loss: 0.1569 - a
ccuracy: 0.9286 - val_loss: 0.5824 - val_accuracy: 0.8160
Epoch 98/100
210/210 [==============================] - 0s 28us/step - loss: 0.1651 - a
ccuracy: 0.9524 - val_loss: 0.6572 - val_accuracy: 0.8180
Epoch 99/100
210/210 [==============================] - 0s 28us/step - loss: 0.1676 - a
ccuracy: 0.9286 - val_loss: 0.5777 - val_accuracy: 0.8030
Epoch 100/100
210/210 [==============================] - 0s 24us/step - loss: 0.1557 - a
ccuracy: 0.9476 - val_loss: 0.6535 - val_accuracy: 0.8250
```



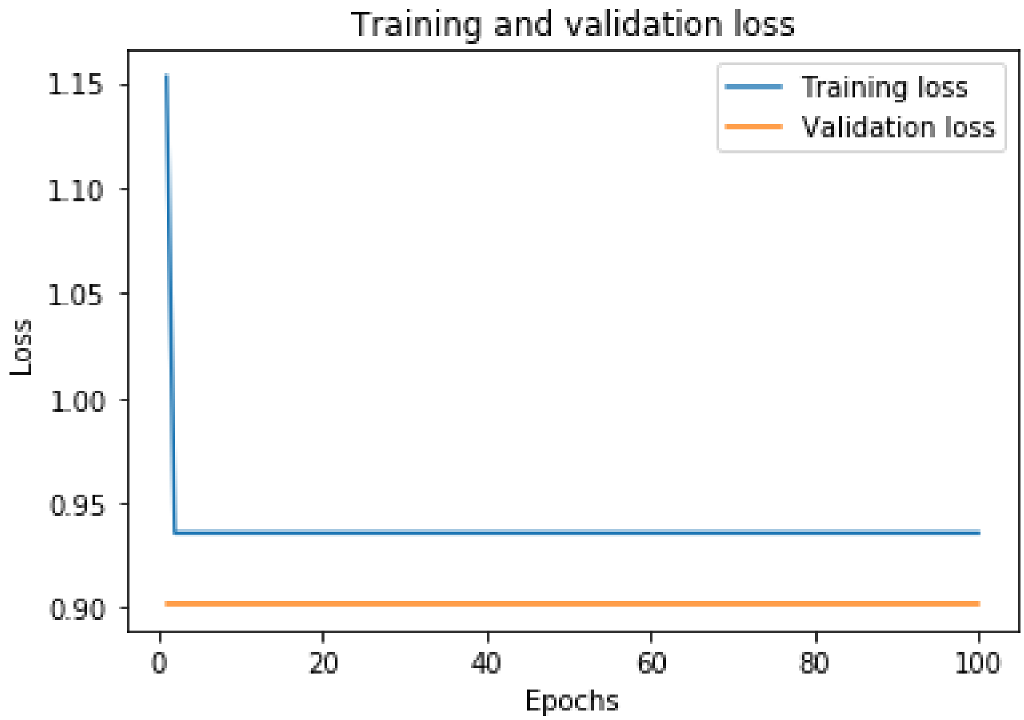

```
Accuracy
518/518 [==============================] - 0s 35us/step
[0.6461128441753535, 0.8262548446655273]
```

Adagrad and hinge

Hyper parameter change

512 hidden units

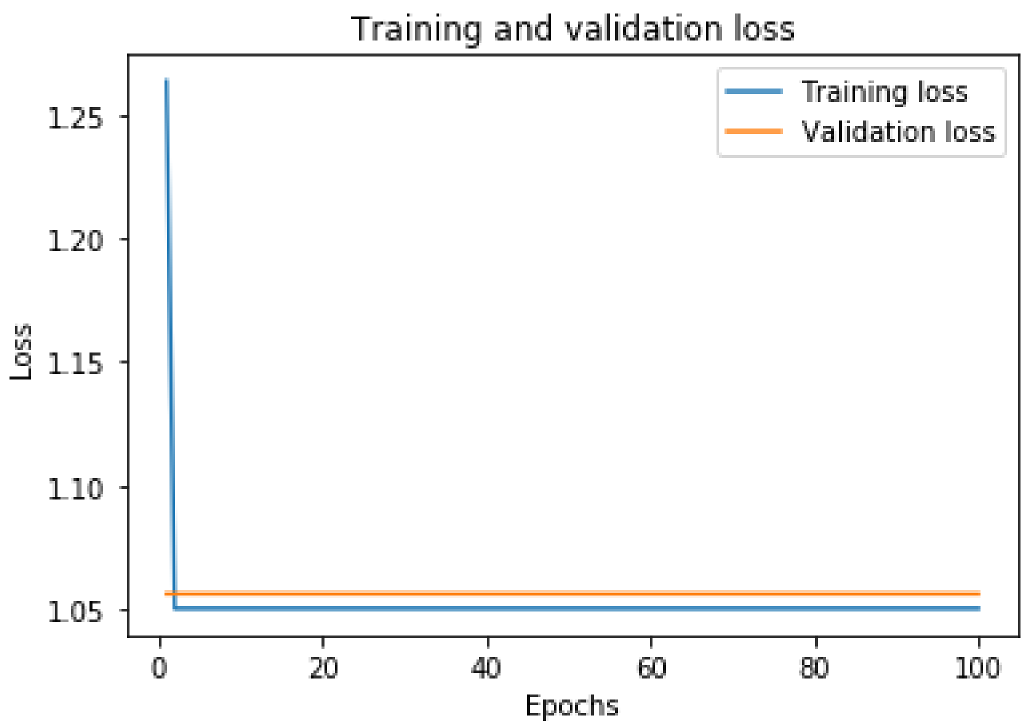
```
Epoch 96/100
210/210 [==============================] - 0s 71us/step - loss: 0.9357 - a
ccuracy: 0.6286 - val_loss: 0.9020 - val_accuracy: 0.6960
Epoch 97/100
210/210 [==============================] - 0s 66us/step - loss: 0.9357 - a
ccuracy: 0.6286 - val_loss: 0.9020 - val_accuracy: 0.6960
Epoch 98/100
210/210 [==============================] - 0s 71us/step - loss: 0.9357 - a
ccuracy: 0.6286 - val_loss: 0.9020 - val_accuracy: 0.6960
Epoch 99/100
210/210 [==============================] - 0s 62us/step - loss: 0.9357 - a
ccuracy: 0.6286 - val_loss: 0.9020 - val_accuracy: 0.6960
Epoch 100/100
210/210 [==============================] - 0s 62us/step - loss: 0.9357 - a
ccuracy: 0.6286 - val_loss: 0.9020 - val_accuracy: 0.6960
```




```
Accuracy
518/518 [==============================] - 0s 25us/step
[0.8812741310439975, 0.7374517321586609]
```

Adagrad and squared hinge

```
Epoch 96/100
210/210 [==============================] - 0s 24us/step - loss: 1.0500 - a
ccuracy: 0.7000 - val_loss: 1.0560 - val_accuracy: 0.6940
Epoch 97/100
210/210 [==============================] - 0s 24us/step - loss: 1.0500 - a
ccuracy: 0.7000 - val_loss: 1.0560 - val_accuracy: 0.6940
Epoch 98/100
210/210 [==============================] - 0s 24us/step - loss: 1.0500 - a
ccuracy: 0.7000 - val_loss: 1.0560 - val_accuracy: 0.6940
Epoch 99/100
210/210 [==============================] - 0s 24us/step - loss: 1.0500 - a
ccuracy: 0.7000 - val_loss: 1.0560 - val_accuracy: 0.6940
Epoch 100/100
210/210 [==============================] - 0s 24us/step - loss: 1.0500 - a
ccuracy: 0.7000 - val_loss: 1.0560 - val_accuracy: 0.6940
```



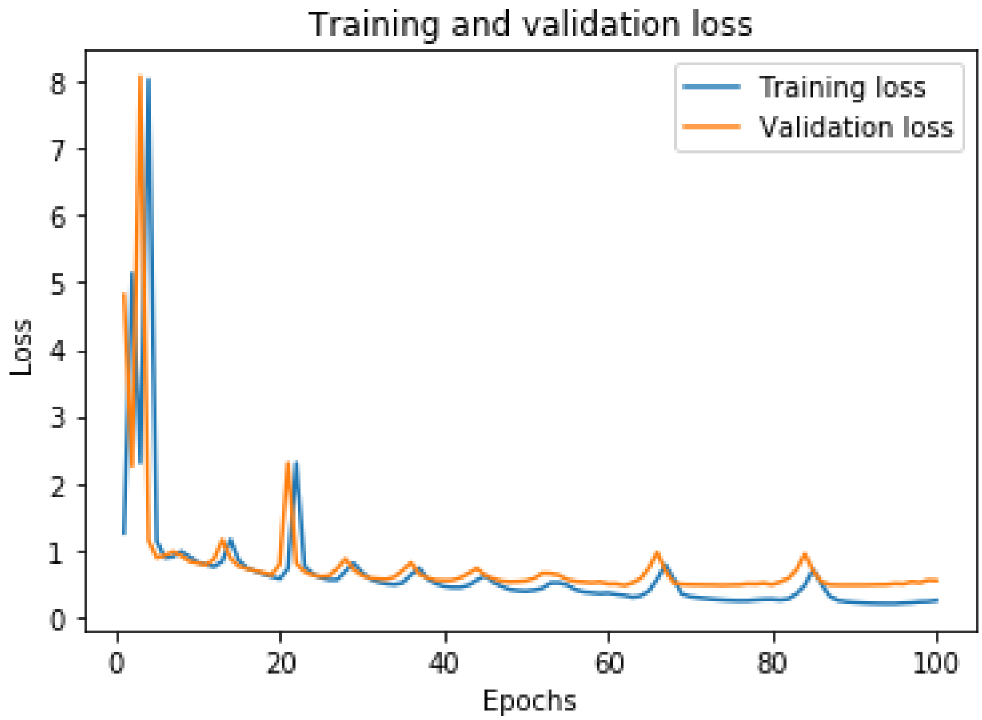

```
Accuracy
518/518 [==============================] - 0s 17us/step
[1.0376447986912083, 0.7123551964759827]
```

Adagrad and kl loss

Hyper parameter change

5 layers

```
Epoch 96/100
210/210 [==============================] - 0s 28us/step - loss: 0.2094 - a
ccuracy: 0.9048 - val_loss: 0.4959 - val_accuracy: 0.8200
Epoch 97/100
210/210 [==============================] - 0s 33us/step - loss: 0.2159 - a
ccuracy: 0.9286 - val_loss: 0.5265 - val_accuracy: 0.8100
Epoch 98/100
210/210 [==============================] - 0s 28us/step - loss: 0.2294 - a
ccuracy: 0.8952 - val_loss: 0.5133 - val_accuracy: 0.8070
Epoch 99/100
210/210 [==============================] - 0s 33us/step - loss: 0.2337 - a
ccuracy: 0.9048 - val_loss: 0.5516 - val_accuracy: 0.8010
Epoch 100/100
210/210 [==============================] - 0s 28us/step - loss: 0.2482 - a
ccuracy: 0.8952 - val_loss: 0.5464 - val_accuracy: 0.7860
```




```
Accuracy
518/518 [==============================] - 0s 17us/step
[0.4395031399708457, 0.8301158547401428]
```

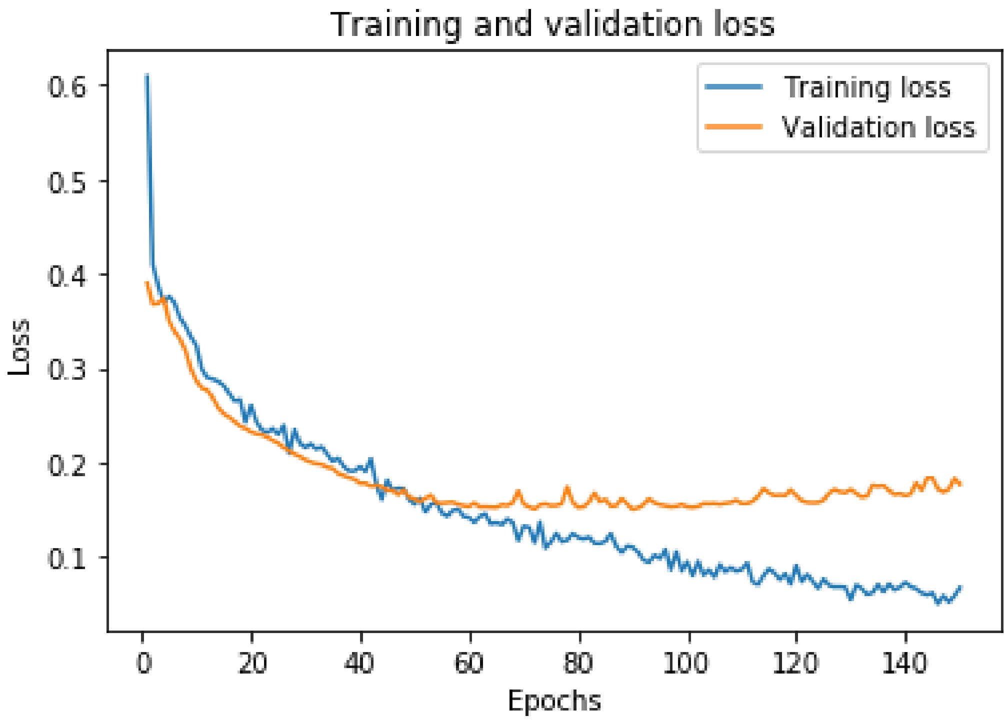
Applying Regularization

The best combination of optimizer and loss found were adam and binary cross entropy with an accuracy of 91% (approx.).

A dropout of 0.5 was added to get a better model. Any other regularization such as batch normalization started decaying the accuracy of the model.

Thus the final output is:

```
Epoch 146/150
210/210 [==============================] - 0s 81us/step - loss: 0.0505 - a
ccuracy: 0.9821 - val_loss: 0.1719 - val_accuracy: 0.9355
Epoch 147/150
210/210 [==============================] - 0s 85us/step - loss: 0.0589 - a
ccuracy: 0.9714 - val_loss: 0.1687 - val_accuracy: 0.9360
Epoch 148/150
210/210 [==============================] - 0s 76us/step - loss: 0.0522 - a
ccuracy: 0.9845 - val_loss: 0.1719 - val_accuracy: 0.9362
Epoch 149/150
210/210 [==============================] - 0s 81us/step - loss: 0.0588 - a
ccuracy: 0.9845 - val_loss: 0.1833 - val_accuracy: 0.9365
Epoch 150/150
210/210 [==============================] - 0s 81us/step - loss: 0.0672 - a
ccuracy: 0.9738 - val_loss: 0.1771 - val_accuracy: 0.9390
```




```
518/518 [==============================] - 0s 27us/step
[0.21079988462641894, 0.9324324131011963]
```

The model now is identified to converge faster and to give little better result than the previous one.

Ensemble classifier:

For better optimization an ensemble model can be developed by combining the models

1. RMSprop and binary cross entropy
2. Adam and binary cross entropy
3. Adagrad and binary cross entropy

(0.9100193190574646 + 0.9074710512161255 + 0.9020270109176636) / 3

=2.7195173811912537 / 3

= 0.9065057937304179

= 90 % (approx.)

It is found that the ensemble value is not as good as the regularized value. Thus an ensemble classifier is not required.

Implementation details

Implemented using anaconda and jupyter.

The issues included

1. Large datasets took a longer time.
2. Time consumption due to evaluation against all optimizers and loss.
3. The code was ran on a CPU that consumed more time.
4. Not every model was saved because the models were misinterpreted by the jupyter kernel. Thus only the best model code is provided.
5. Since it was difficult to save multiple models ensemble classifiers were not implemented.

References:

1. My own assignment 1 code
2. Code given by the professor