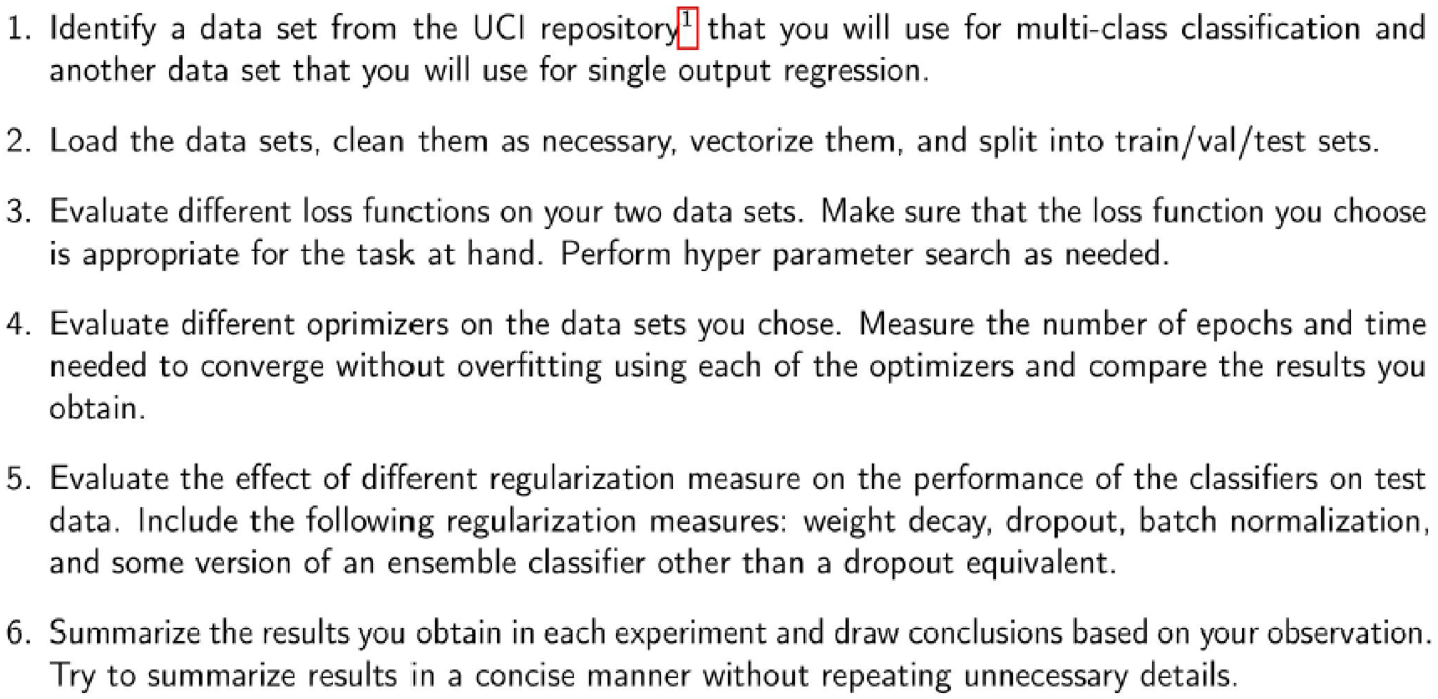
Problem Statement:



1. Identify a data set from the UCI repository[1] that you will use for multi-class classification and another data set that you will use for single output regression.
2. Load the data sets, clean them as necessary, vectorize them, and split into train/val/test sets.
3. Evaluate different loss functions on your two data sets. Make sure that the loss function you choose is appropriate for the task at hand. Perform hyper parameter search as needed.
4. Evaluate different oprimizers on the data sets you chose. Measure the number of epochs and time needed to converge without overfitting using each of the optimizers and compare the results you obtain.
5. Evaluate the effect of different regularization measure on the performance of the classifiers on test data. Include the following regularization measures: weight decay, dropout, batch normalization, and some version of an ensemble classifier other than a dropout equivalent.
6. Summarize the results you obtain in each experiment and draw conclusions based on your observation. Try to summarize results in a concise manner without repeating unnecessary details.

We are trying to improve the performance of multiply neuron and multiple data model which use UCI Data Set.

Proposed Solution UCI Repository:

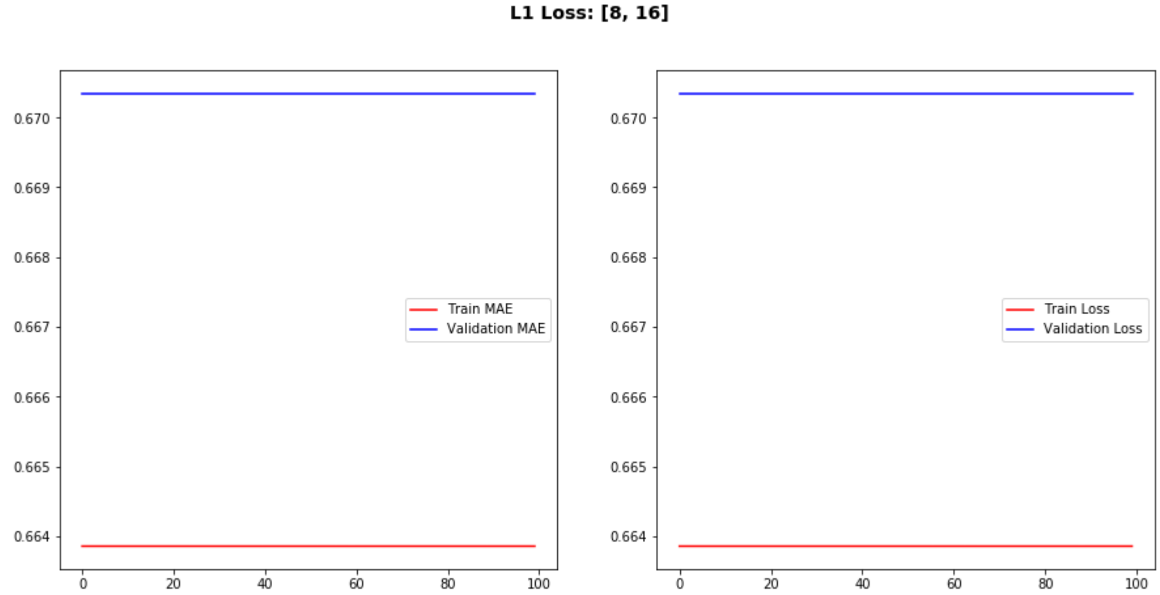
Data: - http://archive.ics.uci.edu/ml/datasets/communities+and+crime.

Implementation details:

1. Data Cleaning:
- Change the value to Categorical.
- Drop columns those have too many missing values.
- Replace missing values with average of the column.

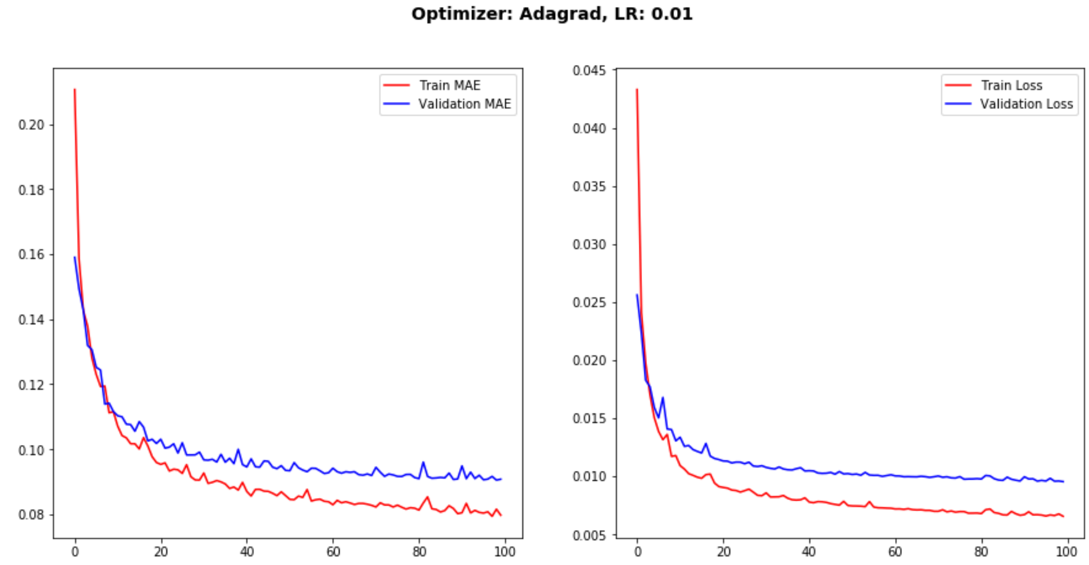
2. Loss Functions:

- I used ‘L1 Loss', 'L2 Loss', 'Huber Loss', and 'logcosh Loss'



3. Optimizer:

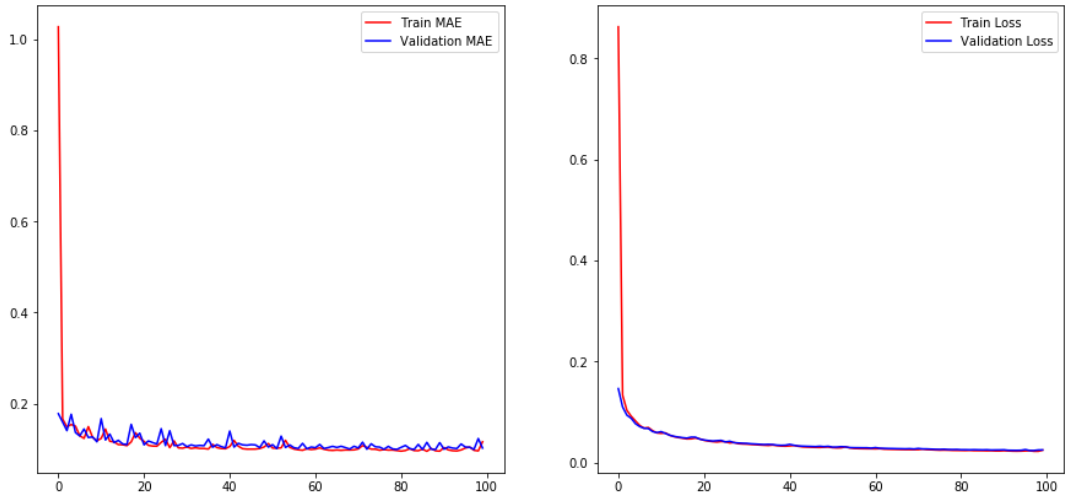
- In my program I used Stochastic gradient descent (SGD), Adagrad and RMSprop .



4. Regularization:

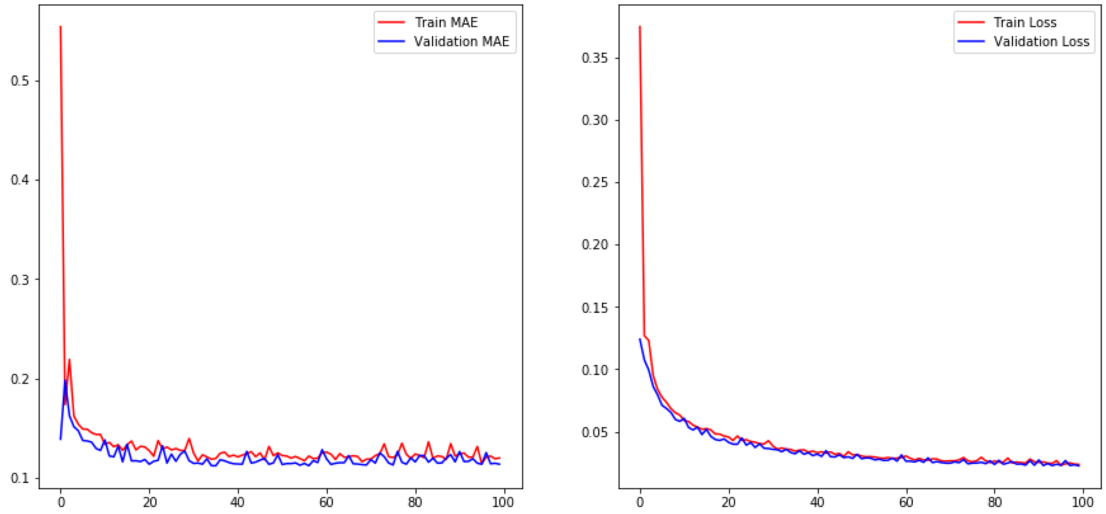
- Evaluating the regularization measure on the performance on the test data.

Regularizer: l2, Regularization Rate: 0.01

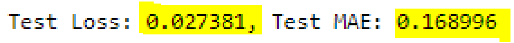


- I used Weight Decay, Dropout, and Batch Normalization.

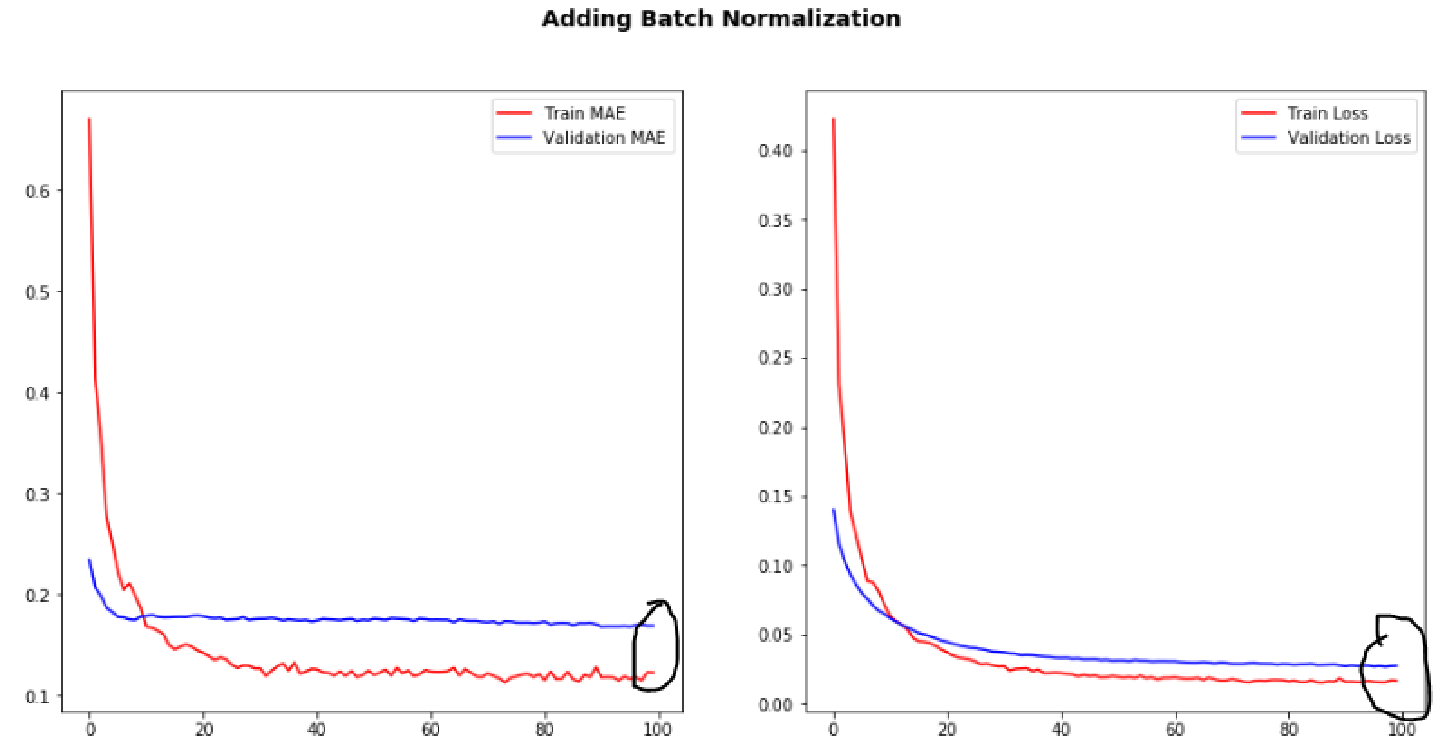
Dropout Level: 0.25



Results and Discussion:



```
Test Loss: 0.027381, Test MAE: 0.168996
```



Final Mode Details for UCI Repository:

1. Number of Epochs = 100
2. Learning Rate = 0.01 (1%)
3. Regularization Rate: 0.01
4. Regularize = L2
5. Hidden Layers = [8, 16]
6. Batch Size = 512
7. Optimizer = Adagrad
8. DropOut = 0.25
9. 80-20 percentage of partition train and test data in the case of UCI data set.
10. Test Loss: 0.**027381**, Test MAE: 0.**168996**

Proposed Solution cifar10:

CIFAR_10 Dataset

#Keras built in dataset was used for this assignment:

#(x_train_all, y_train_all), (x_test_all, y_test_all) = cifar10.load_data()

Implementation details:

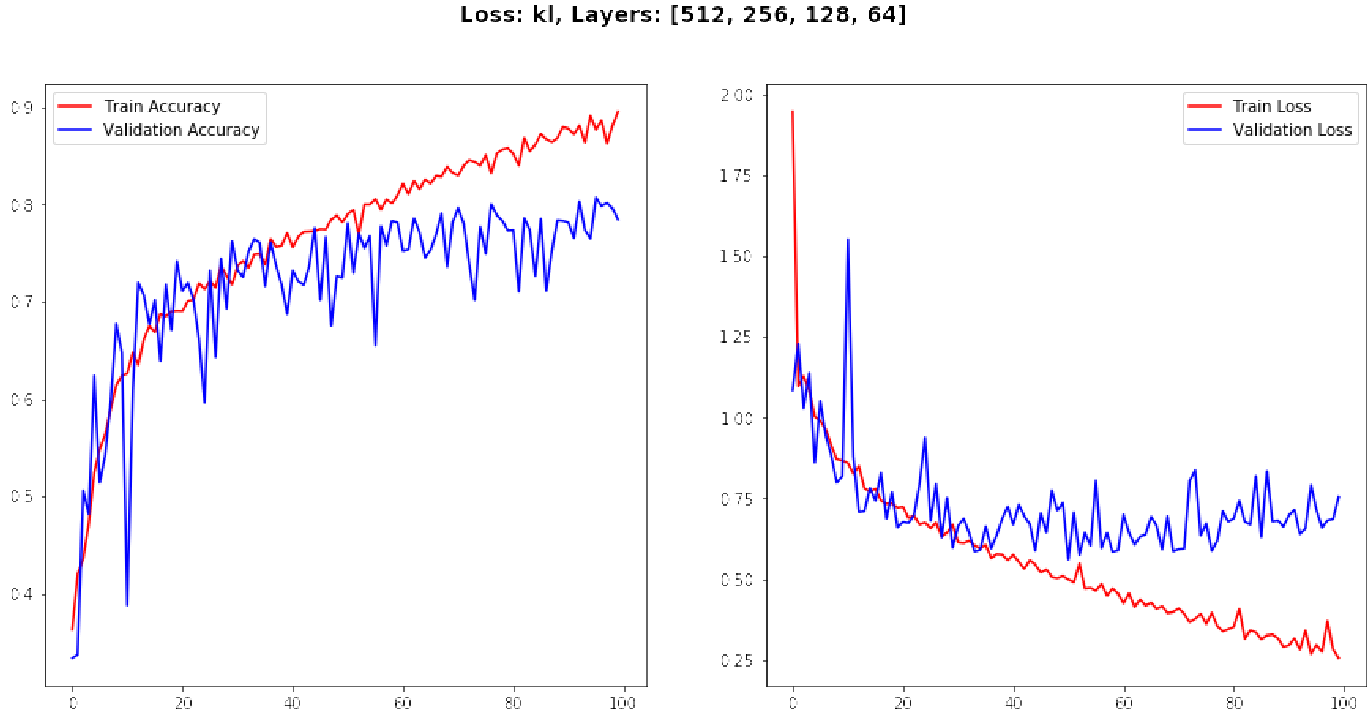
1. Data Cleaning: Not required.

- I used 3 classes for modes.

2. Loss Functions:

- I used ‘KL(Kullback-Leibler) Loss', 'Hing Loss', and ‘Categorical crossentropy Loss'

  Best Model: Loss: kl, Hidden Layers: [512, 256, 128, 64]

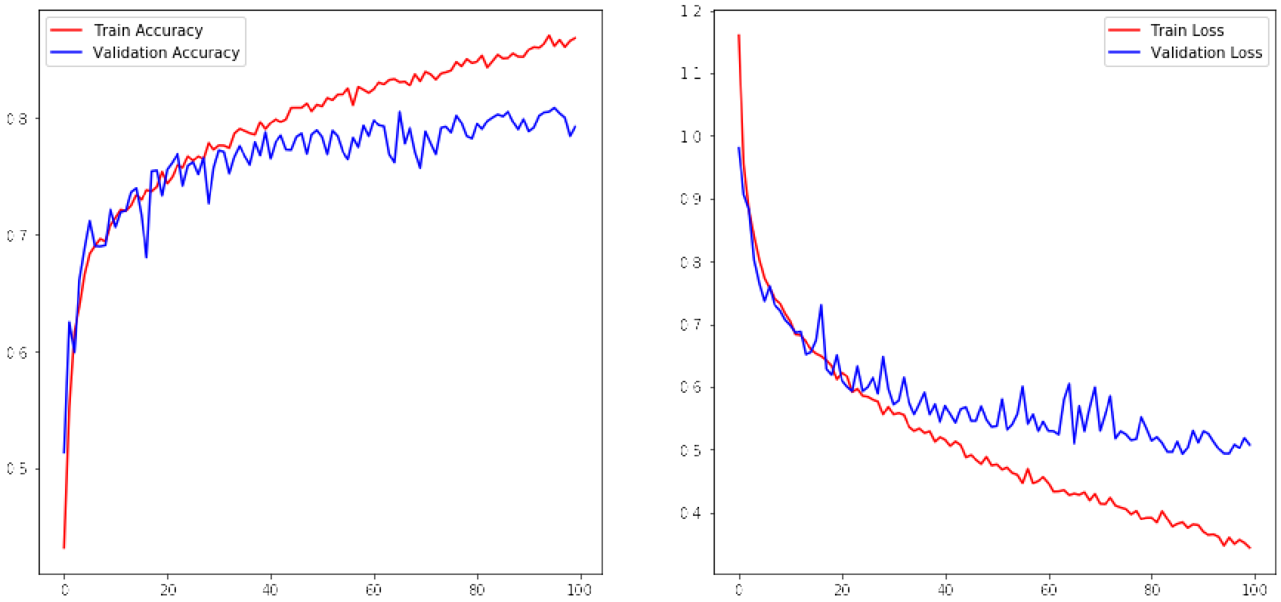


3. Optimizer:

- In my program I used Stochastic gradient descent (SGD), Adagrad and RMSprop .

Best Model: Optimizer: Adagrad, Learning Rate: 0.001

Optimizer: Adagrad, LR: 0.001



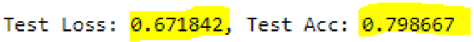
4. Regularization:

- Evaluating the regularization measure on the performance on the test data.
- I used Weight Decay, Dropout, and Batch Normalization.

Results and Discussion:

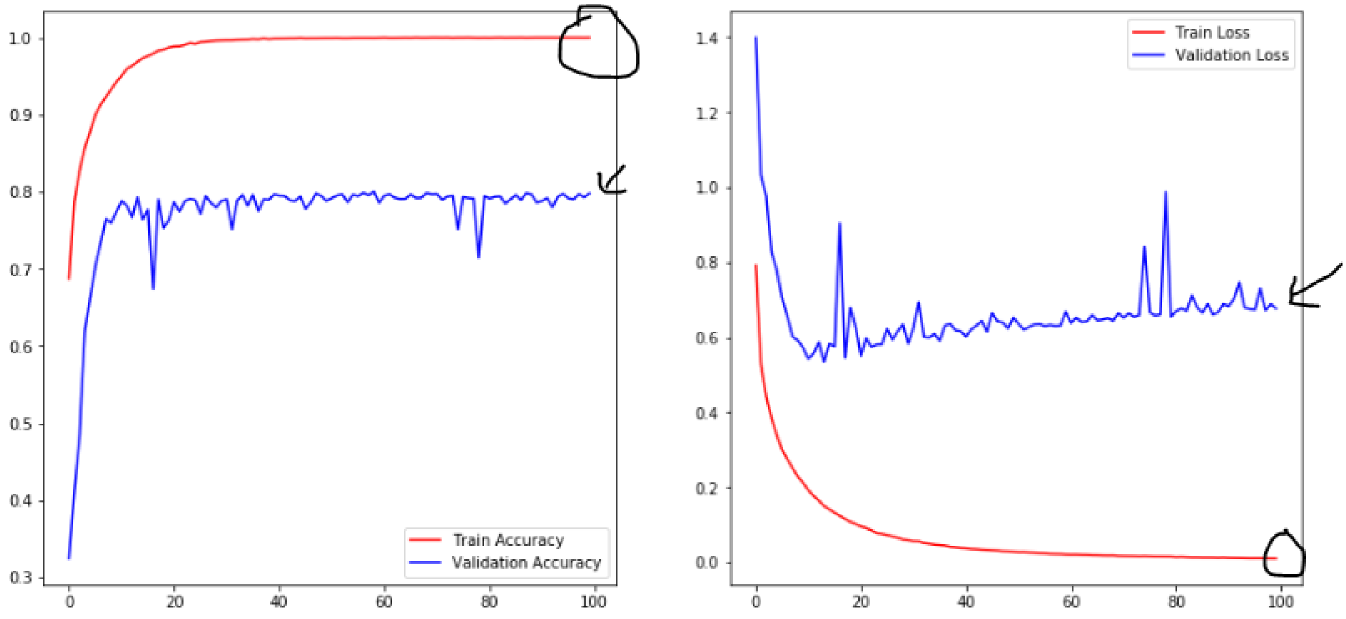
Final Mode Details for CIFAR_10:

1. Number of Epochs = 100
2. Learning Rate = 0.001 (0.01%)
3. Regularization Rate: does not improve the performance of the models.
4. Regularize = L2
5. Hidden Layers = [512, 256, 128, 64]
6. Loss function = kl (Kullback-Leibler)
7. Batch Size = 512
8. Optimizer = Adagrad
9. DropOut = has not improved the performance.
10. 80-20 percentage of partition train and test data in the case of CIFAR_10 data set.
11. Test Loss: 0.671842, Test Acc: 0.798667



```
Test Loss: 0.671842, Test Acc: 0.798667
```

Batch Normalization


Train Accuracy
Validation Accuracy
Train Loss
Validation Loss