CSCI 5922 -Neural Networks and Deep Learning
Lab - 1 Solutions

Gowri Shankar Raju Kurapati
Student ID 110568555

September 11, 2022

1 Neural Network Hyperparameters

1.1 Dataset & Hyperparameters Used

- CIFAR 10 Dataset
  - Source : From *torchvision.datasets*
  - The CIFAR-10 dataset consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images.
  - Since these are color images, each image has the tensor dimension of 3 * 32 * 32 where the first dimension is RGB colors.
  - The 10 classes (labels) are $airplane(0), automobile(1), bird(2), cat(3), deer(4), dog(5), frog(6), horse(7), ship(8), truck(9)$
- Hyperparameters
  - training/test split is 70 : 30
  - Number of neurons in each hidden layer is 1024 is held constant for all hidden layers used for the experiment.
  - The input size to the neural network is 3 * 32 * 32 image which is reshaped to a vector of length 3072 and is fed to the network.
  - As stated above, the output layer consists of 10 neurons to predict 10 classes of the CIFAR 10 dataset.
  - The learning rate is set to 0.001 with a batch size of 128 and is held constant for the experiment.
  - The models are trained for 20 epochs.
- Methods:
  - All neural networks are Vanilla Neural Networks (no CNNs Used) with input Layer, hidden Layer, and output Layer with softMax.
  - Loss function used for the experiment is the Cross Entropy.
  - Adam Optimizer with a learning rate of 0.001 is used.

- As specified, I have considered three activation functions ReLU, Tanh and Sigmoid.
- Also, I have iterated the experiment with 1,2& 3 hidden layers while also iterating with activation functions as mentioned above, which leaves us with nine neural network models.
  * The models are named as Lab1_P1 _HiddenL_<no of hidden layers>_Activation _<FunctionName> to capture the activation function and the number of hidden layers.

1.2 Results & Analysis

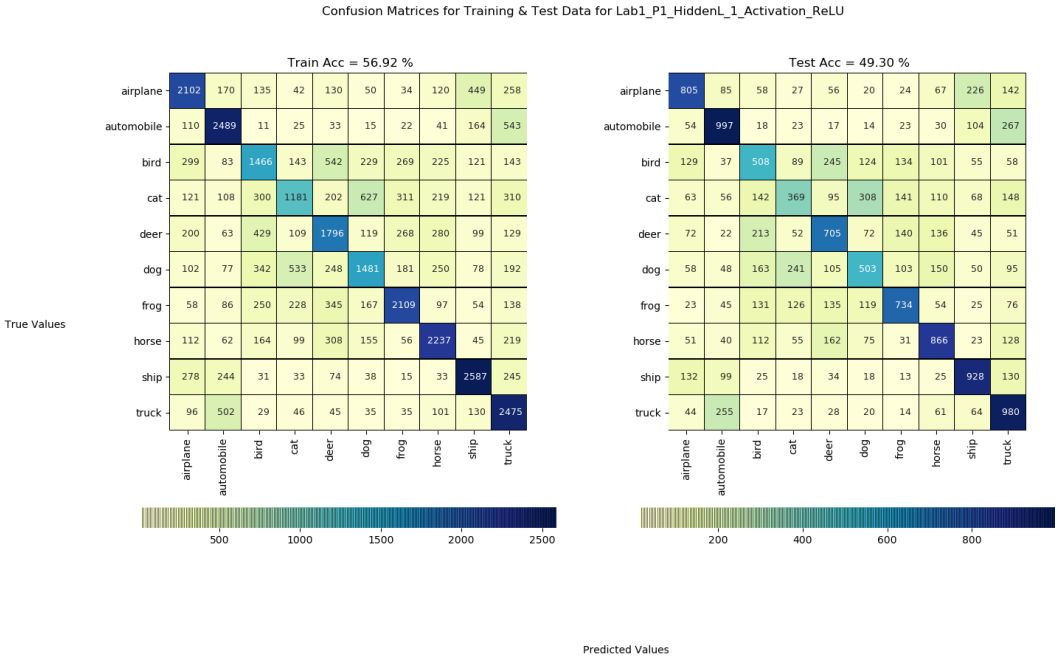
Reporting **Test Accuracies** for the NINE NN Models

ActivationF / Hidden Layers	ReLU	Tanh	Sigmoid
1	49.30	43.96	47.49
2	50.73	40.12	47.62
3	50.21	36.49	45.95

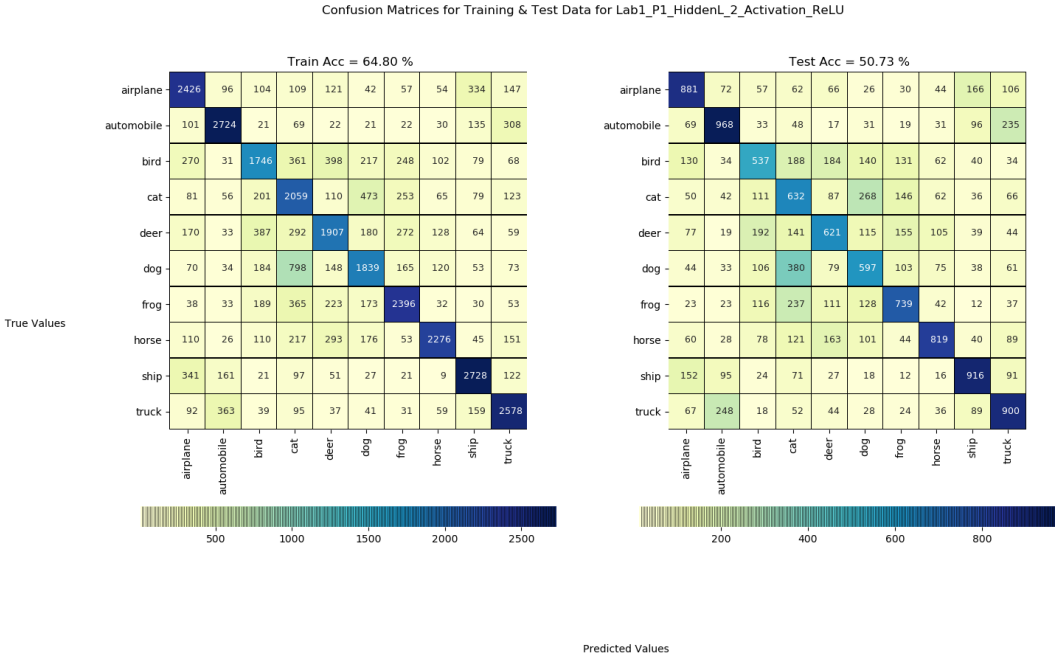
Reporting **Training Accuracies** for the NINE NN Models for comparison

ActivationF / Hidden Layers	ReLU	Tanh	Sigmoid
1	56.92	48.36	55.27
2	64.80	42.63	55.09
3	70.52	38.37	51.37

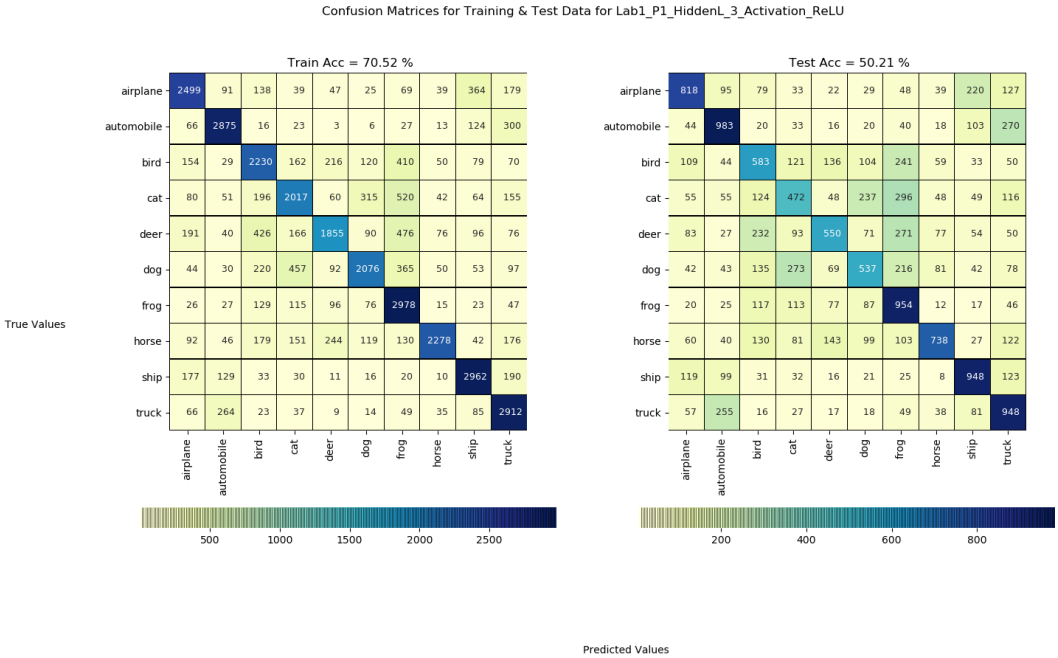
- From the tables above, we could infer that,
  - [**ReLU Performs Better**] From the test accuracies table, ReLU activation gives accuracies better than the Tanh and Sigmoid functions when compared against a configuration of hidden layers. Since it is an image dataset (which is reshaped) the model may be trying to pick up small features from the pixel (like tire, headlights etc) and its surroundings and then try to map them to a bigger feature (like a car). Since ReLU gives zero value before a threshold and a positive value after a threshold, at each neuron in the hidden layers, it might capture whether a small feature exists or not and if it exists, how persistent it is. This analogy is in perfect sync with the ReLU function. This can also be strengthed by the fact of poor accuracies when using continuous functions like tanh and Sigmoid, though sigmoid seems to do better than tanh.
  - [**Overfitting With More Hidden Layers**] When considering the better performing ReLU, we can see that as the number of hidden layers is increased, the training accuracy increases significantly(65 -> 71) but at the same time we can see the testing accuracy decreasing (50.73 -> 50.21). This shows that though the model is performing well on training data with a significant amount of accuracy increase, the test accuracy almost remains stagnant. This is a clear sign of the model trying to overfit the training data. Sigmoid, Tanh activation functions seem to struggle to capture the features from the image pixel data and even the increase in the hidden layers isn't helping the model to perform better, and sometimes the training accuracy itself is going down by underfitting the training data and thereby having an impact on test accuracy as well.



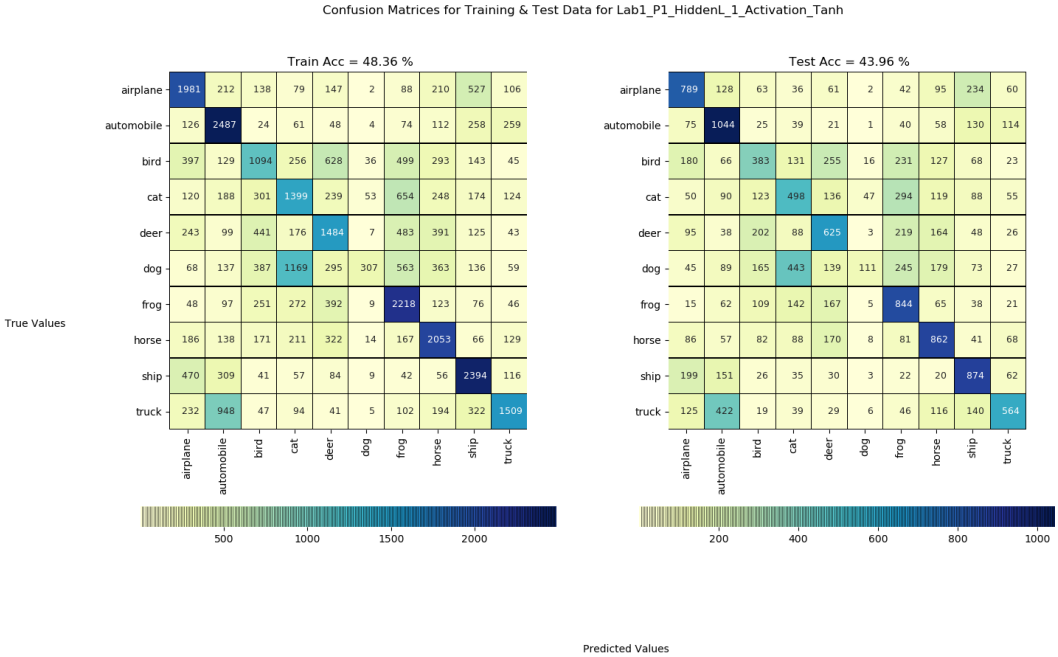
(a) 1 Hidden Layer with ReLU



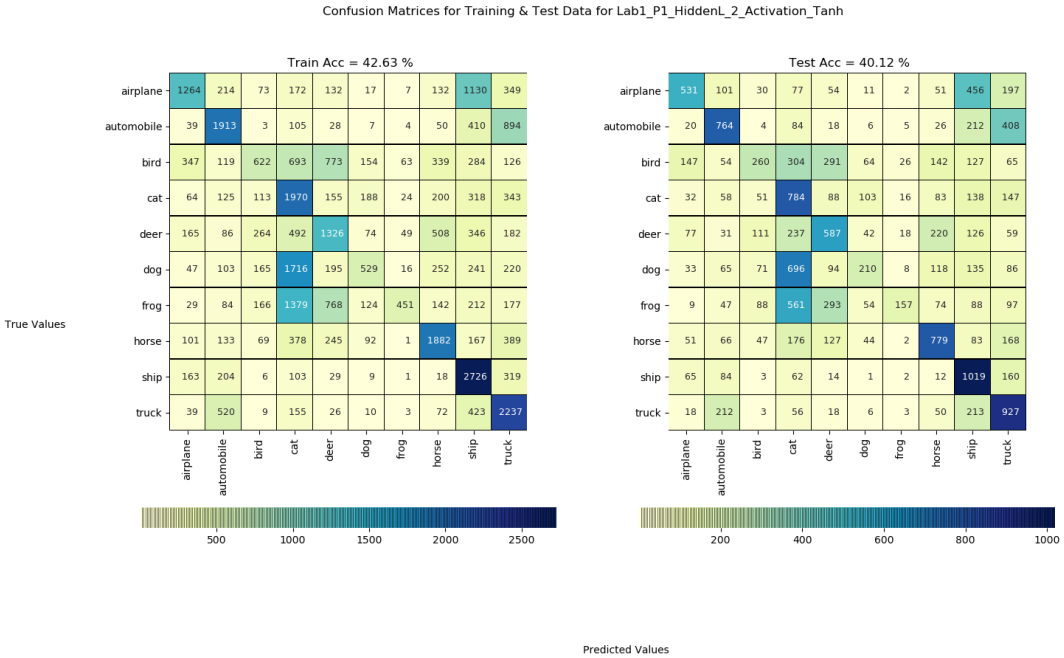
(b) 2 Hidden Layers with ReLU



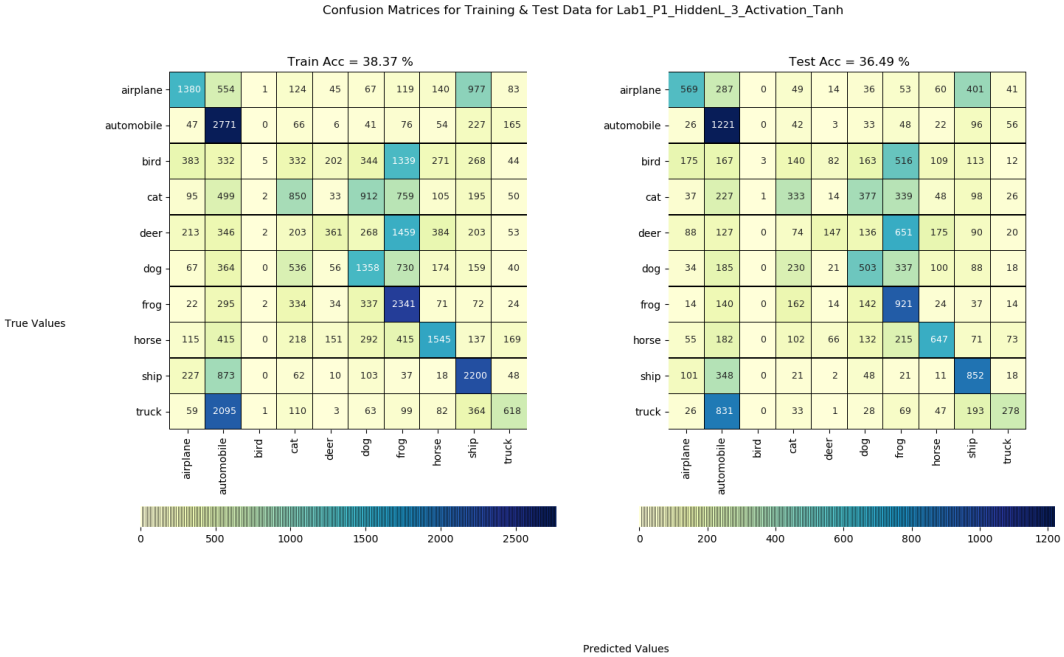
(a) 3 Hidden Layer with ReLU



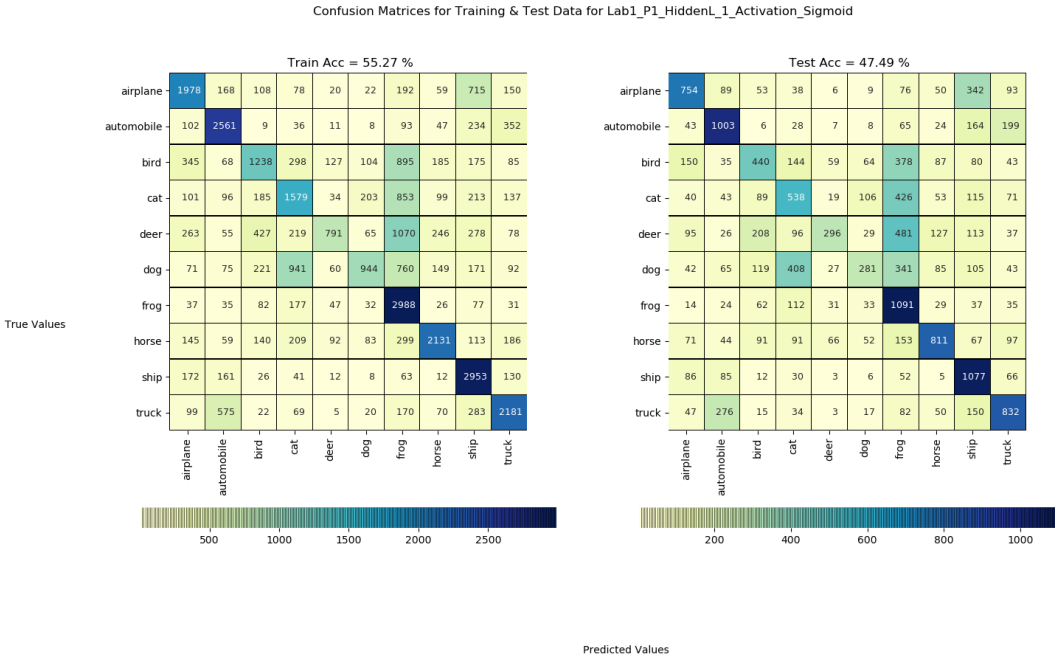
(b) 1 Hidden Layer with Tanh



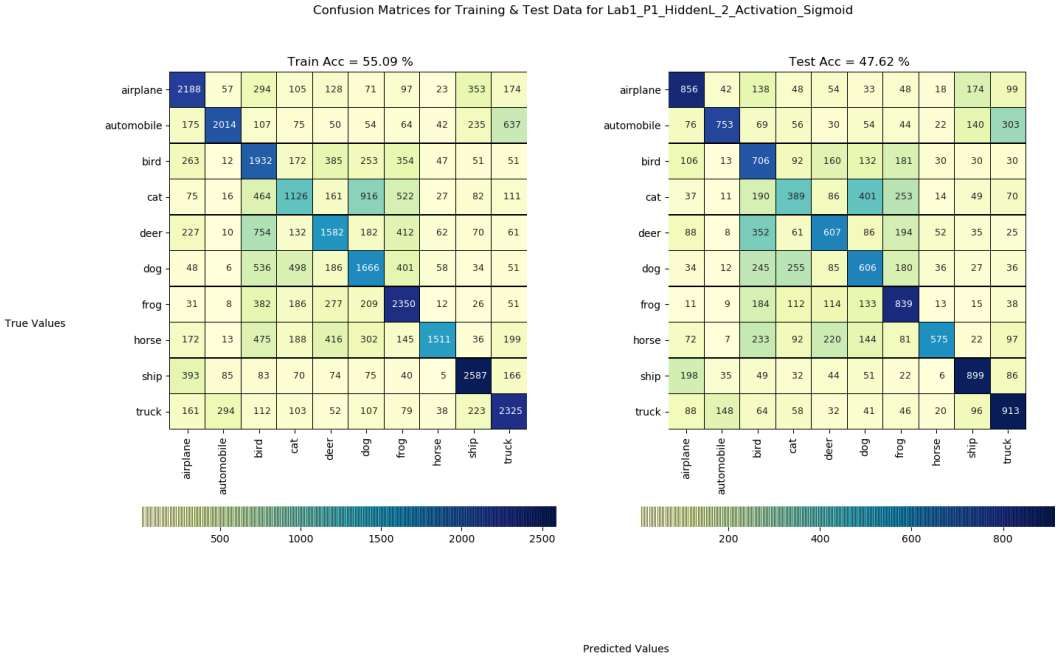
(a) 2 Hidden Layers with Tanh



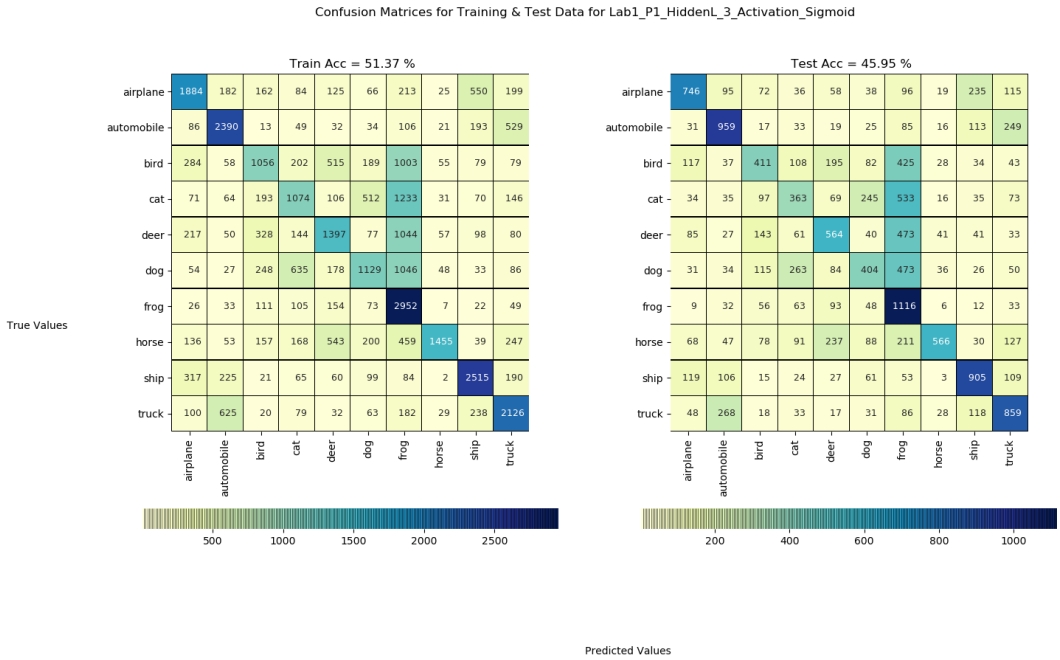
(b) 3 Hidden Layers with Tanh



(a) 1 Hidden Layer with Sigmoid



(b) 2 Hidden Layers with Sigmoid



(a) 3 Hidden Layers with Sigmoid