CS 6375
ASSIGNMENT -2
(Neural Networks)

Names of students in your group:

Siddhant Suresh Medar (ssm200002)
Adithya Sundararajan Iyer (asi200000)

Number of free late days used: _____1_____

Note: You are allowed a **total** of 4 free late days for the **entire semester**. You can use at most 2 for each assignment. After that, there will be a penalty of 10% for each late day.

2 Programming Part (60 points)

Please list clearly all the sources/references that you have used in this assignment:

1) Perceptron slides and Artificial Neural Network slides from E-Learning CS6375 Course Contents Page
2) https://www.tensorflow.org/tutorials/keras/classification
3) https://keras.io/guides/keras_tuner/getting_started/
4) https://www.tensorflow.org/api_docs/python/tf/keras

Dataset Used:

Iris Data Set (https://archive.ics.uci.edu/ml/datasets/iris)

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Libraries Used

- Numpy
- Pandas
- TensorFLow
- Matplotlib
- Scikit-learn
- Keras
- Warnings

Data Preprocessing:

1. Check for null or NA values and remove
2. Remove redundant rows or duplicate values
3. Encode categorical attributes
4. Standardize the data

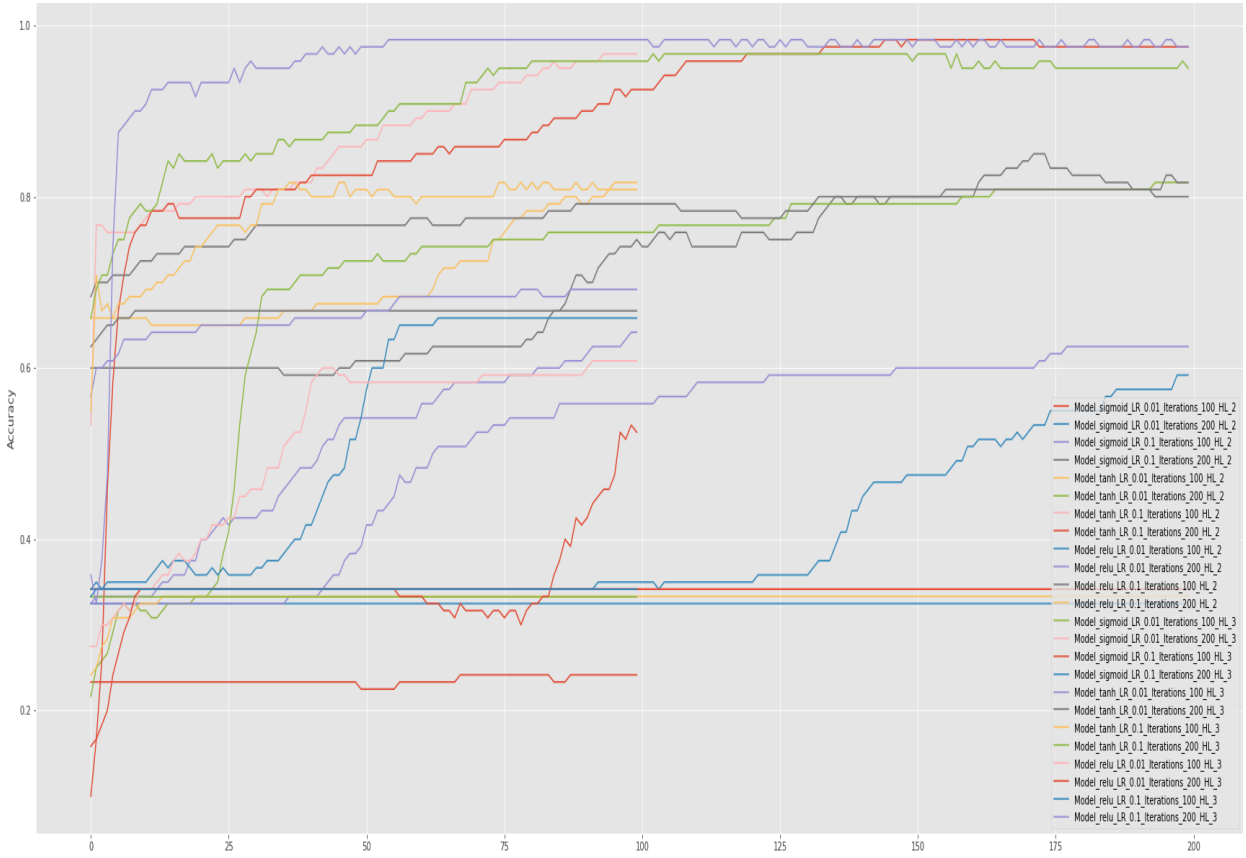
Hyperparameter Permutations

- activations = ['sigmoid', 'tanh', 'relu'] #NOTE: In keras, Logistic activation is nothing but sigmoid
- learning_rate = [0.01, 0.1]
- iterations = [100, 200]
- number_of_hidden_layers = [2, 3]

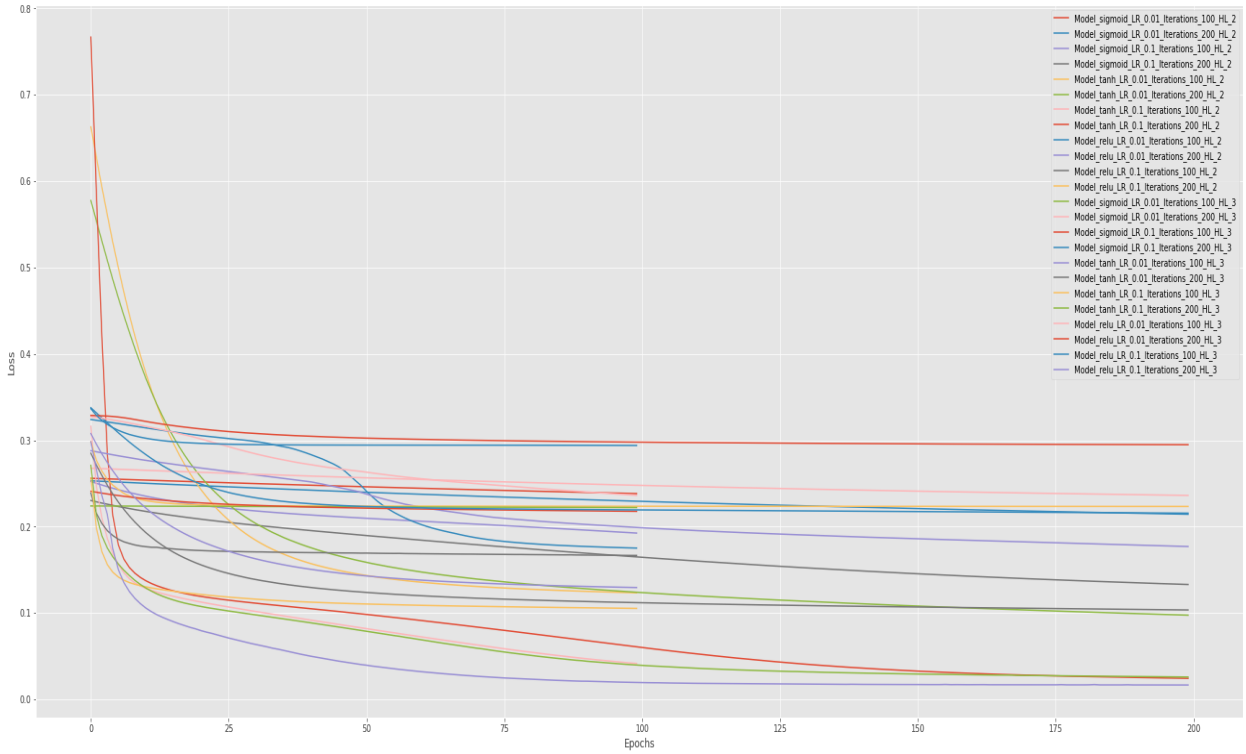
Overall, this gives us 24 combinations for training and testing our model on.

Model History Plots

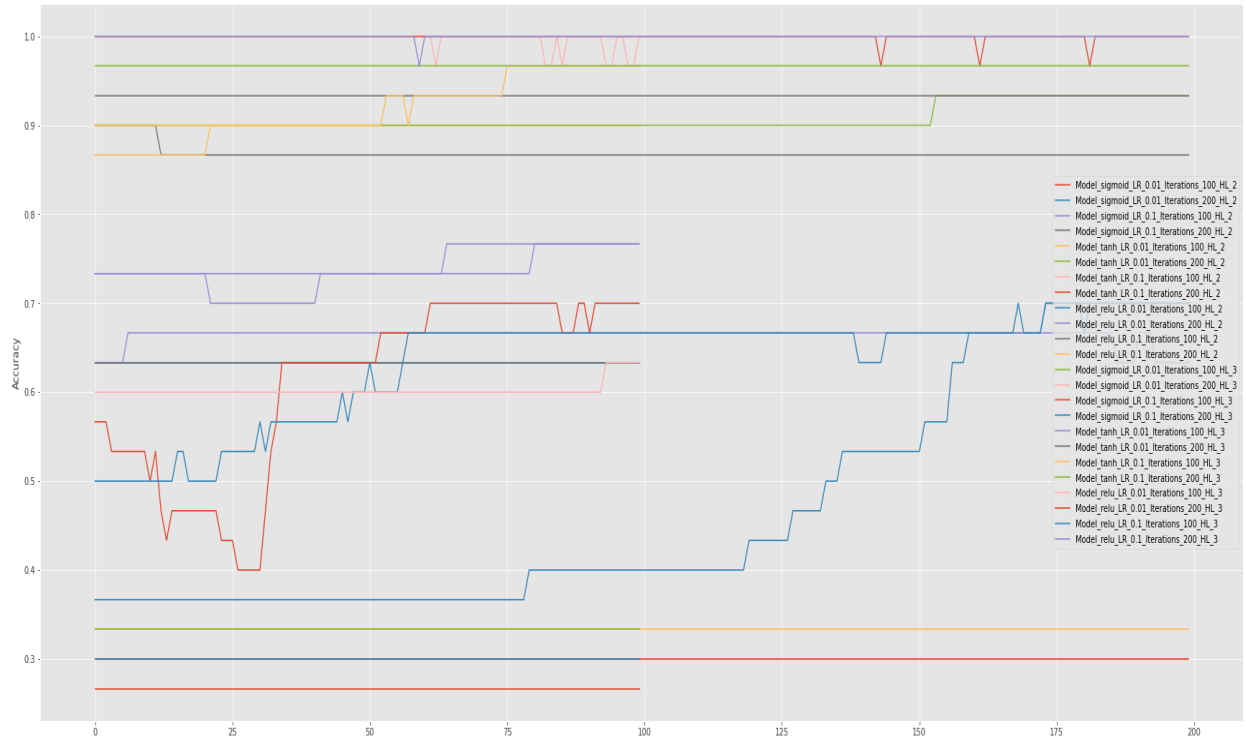
Training accuracy vs number of epochs



Training Loss vs number of epochs



Testing Accuracy vs number of epochs



Testing Loss vs number of epochs

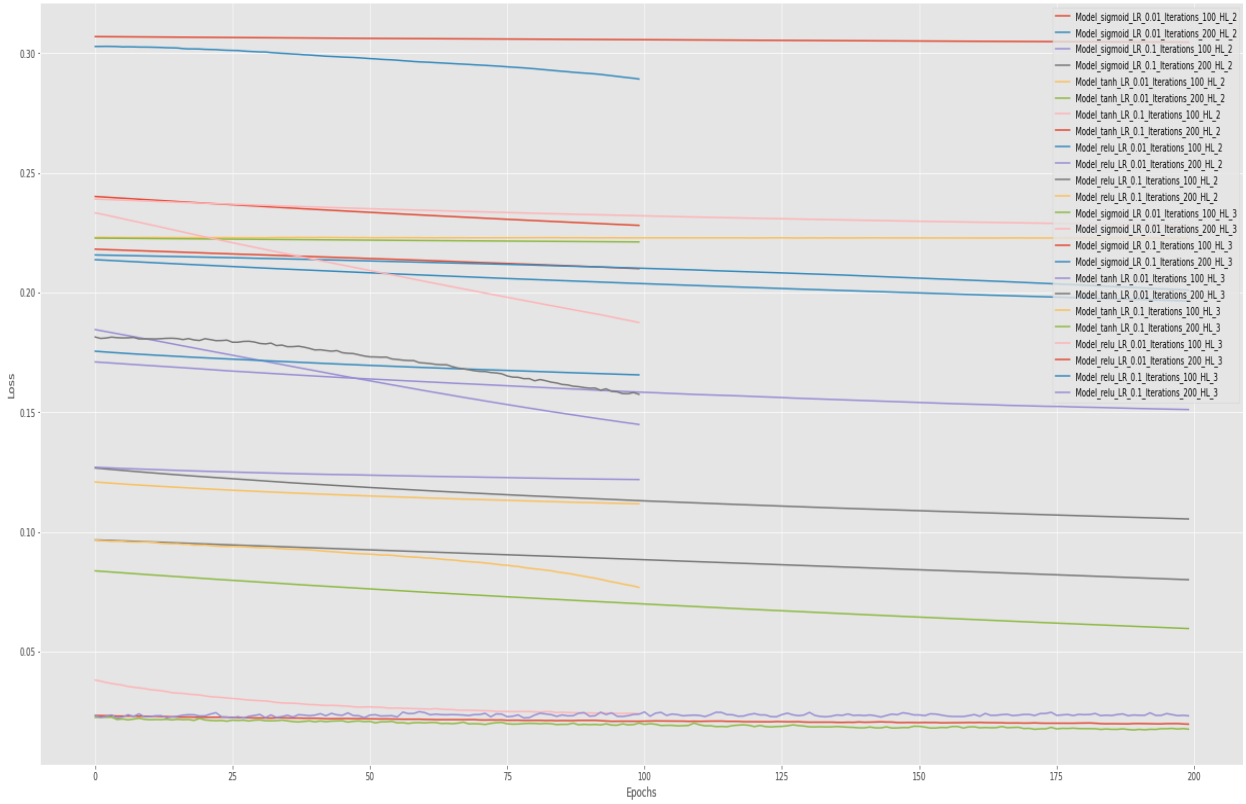


Table of Results

Data for table represented directly in notebook and pdf files after discussion with the professor.

Conclusion

From the results obtained and the graphs plotted, we are able to see that the models that use the RELU (Rectified Linear Unit) Activation Unit give better accuracy when compared to those using sigmoid or tanh activation. The RELU function can be considered better in this case due to the fact that it gives the acceptable performance faster. This happens due to the advantages of RELU over the Sigmoid or the Hyperbolic Tangent Activation Functions. The Vanishing Gradient Problem is completely removed when we use RELU. The model learning becomes faster due to this. RELU also has the drawback of not being upper bounded but we can use a regression function like sigmoid or tanh in these cases. Sigmoid and tanh have similar performance with tanh only being marginally better.