CSE 574: Introduction to Machine Learning(Fall 2017)

Project 3 - Classification: Report

November 20, 2017

Group members:

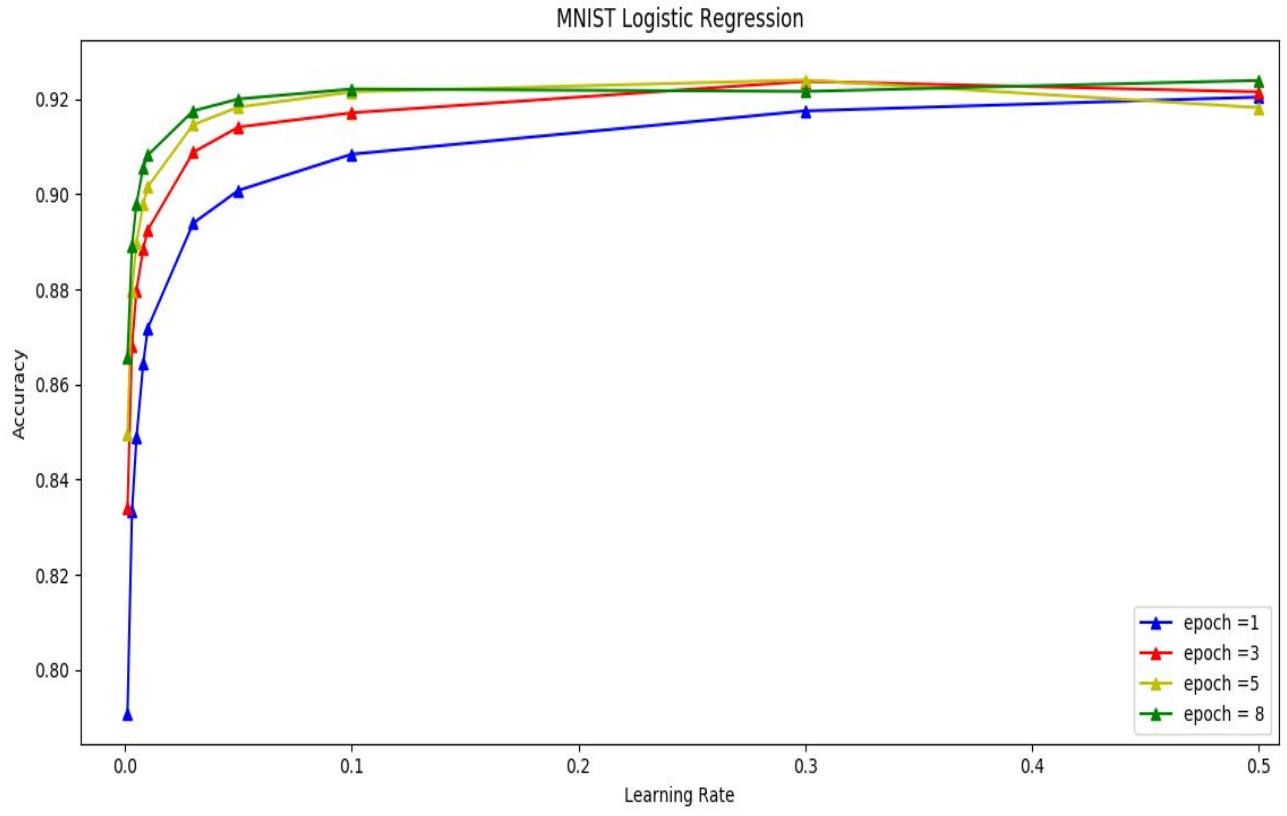
1. karanhor - 50249274
2. pbisht2 - 50247429
3. rakeshsi - 50249135

Task 1 : Implement logistic regression, train it on the MNIST digit images and tune hyperparameters

Tuned values of hyper parameters on best accuracy of MNIST test data using logistic regression.
Epoch: 5
Learning rate: 0.3
Optimizer: Gradient Descent
Logistic Accuracy on test MNIST data: 0.924

In logistic regression we use sigmoidal function because there we only had two classes.
But in our current project we have multiple classes so we cannot use sigmoidal function.
So we are using softmax function. If you want to assign probabilities to an object being one of several different things, softmax is the thing to do, because softmax gives us a list of values between 0 and 1 that add up to 1. Even later on, when we train more sophisticated models, the final step will be a layer of softmax.

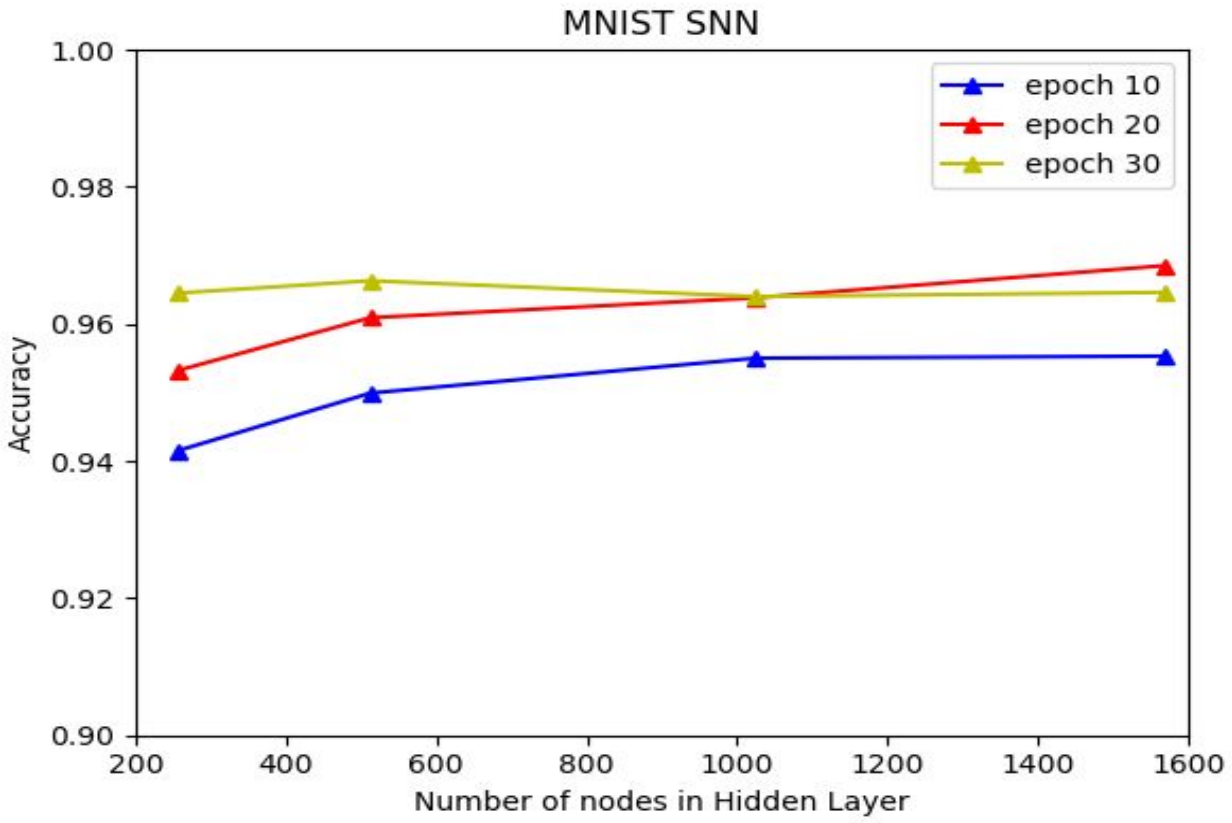
Accuracy is the (number of correct predictions)/(total number of predications). It's range is from 0 to 1.
Accuracy = 1 - Error.



Task 2: Implement single hidden layer neural network, train it on the MNIST digit images and tune hyperparameters such as the number of units in the hidden layer

Tuned values of hyper parameters on best accuracy of MNIST test data using single layer neural network.
Epoch: 20
Learning rate: 0.01
Optimizer: Adam Optimizer
No of units in hidden layer: 1568
Single NN Accuracy on test MNIST data: 0.9685

We have used one hidden layer in the neural network and varied the no. of units in that layer. We have used reLU as the activation function in this case as it simplest non-linear activation function. When we get positive input, the derivative is just 1, so there isn't the squeezing effect we meet on back propagated errors from the sigmoid function or softmax function.
The choice of no. of units is made upto a maximum of twice the actual size of the inputs (2*Image size = 2 * 28* 28 = 1568). Range of no. of units in hidden layer used to tune { 256, 512, 1024, 1568 }. Range of learning rate used { 0.003, 0.01, 0.1, 1 }. Range of epochs used { 10(Blue), 20(Red), 30(Yellow) }



Task 3: Use a publicly available convolutional neural network package, train it on the MNIST digit images

Learning rate: 1e-4
Optimizer: Adam Optimizer
CNN MNIST test accuracy 0.9909
CNN USPS accuracy 0.66555

The following is the structure of our CNN:

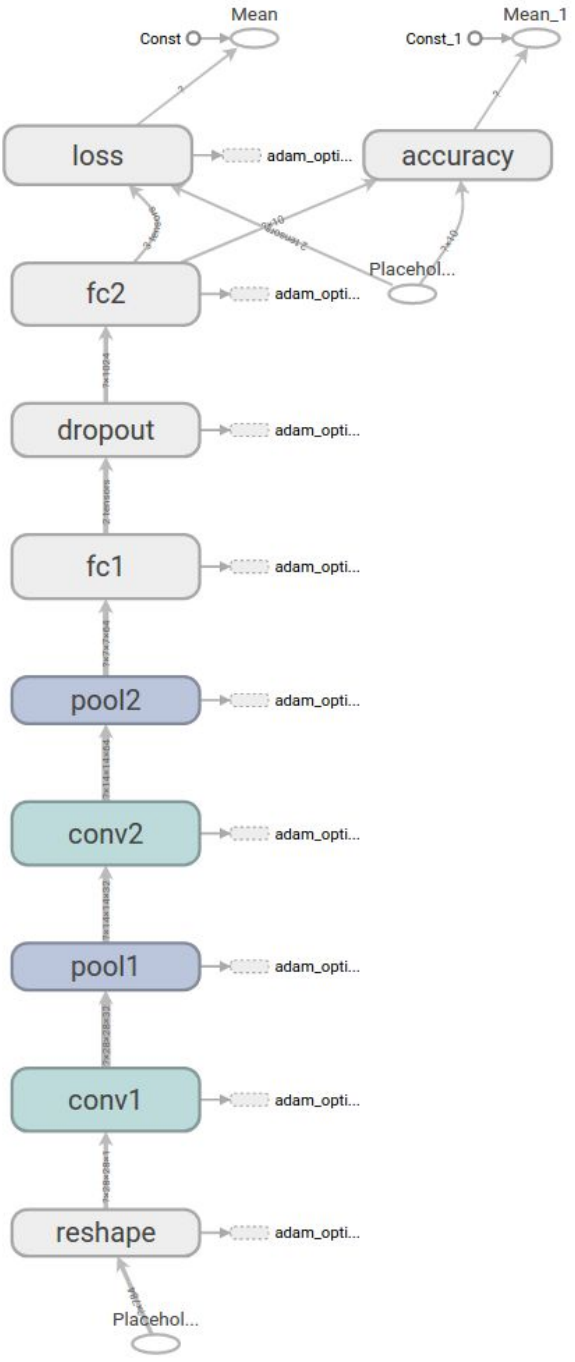
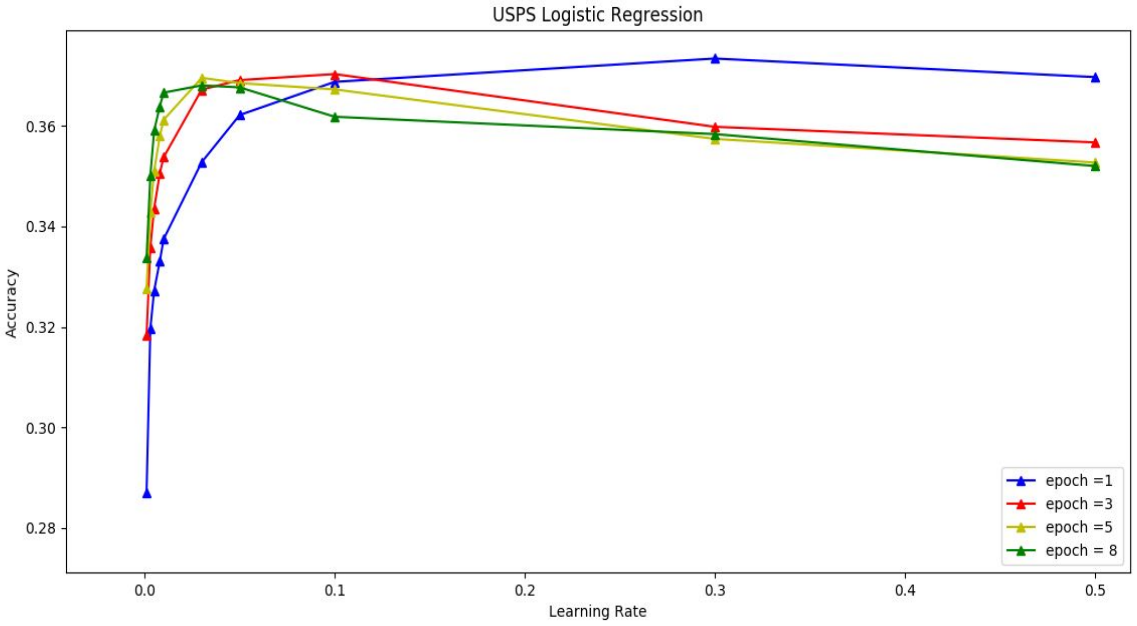


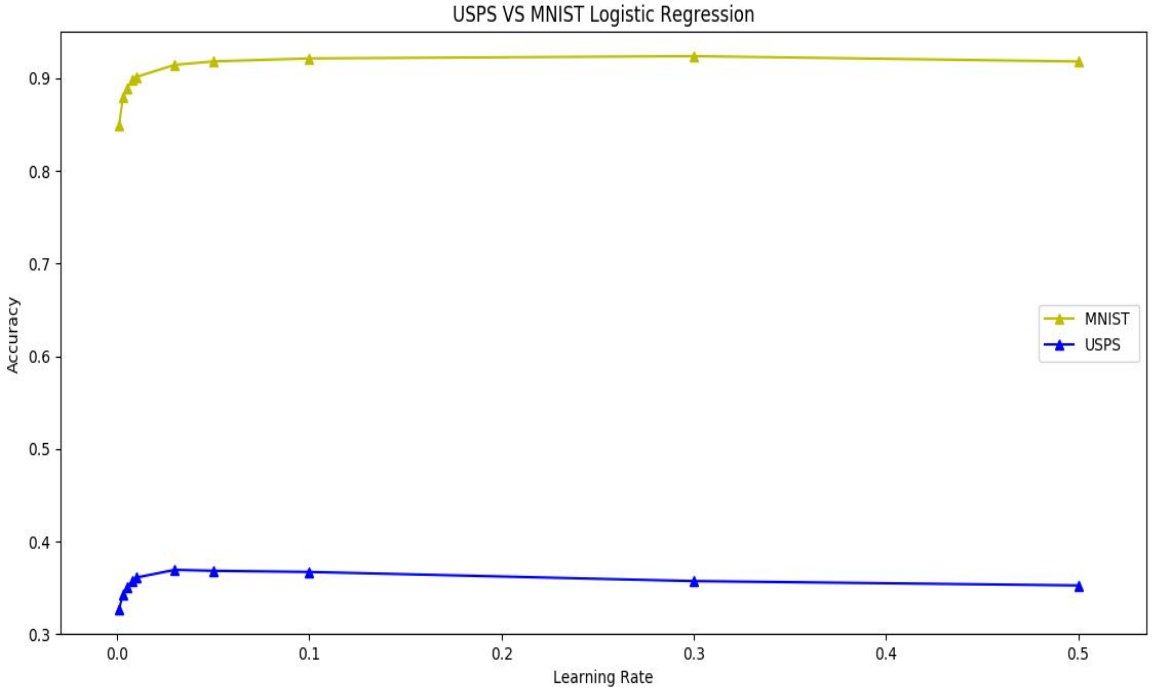
Fig: Structure of CNN (source: tensorflow.org/tutorials)

Task 4: Test your MNIST trained models on USPS test data and compare the performance with that of the MNIST data. Does your finding support the "No Free Lunch" theorem?

The **"No Free Lunch" theorem** states that there is no one model that works best for every problem. The USPS data is first converted to grayscale and then all the images are resized to 28*28 pixels.

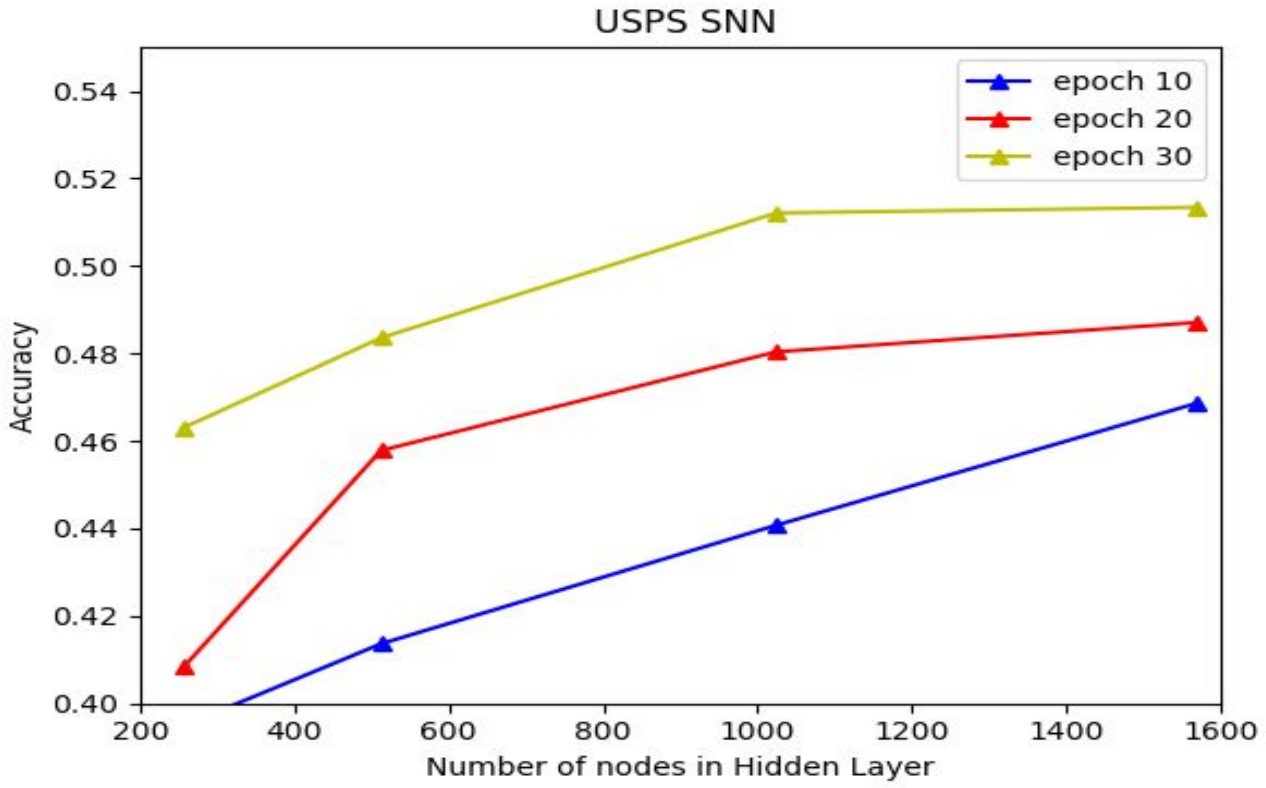
Logistic Accuracy on USPS data: 0.35745, Epoch: 5, Learning Rate: 0.3
It supports No free lunch theorem as the max value(tuned) parameters of MNIST data does not give the max value for USPS data. The max value for USPS **0.37345** occurs at learning rate 0.3 and Epoch 1.

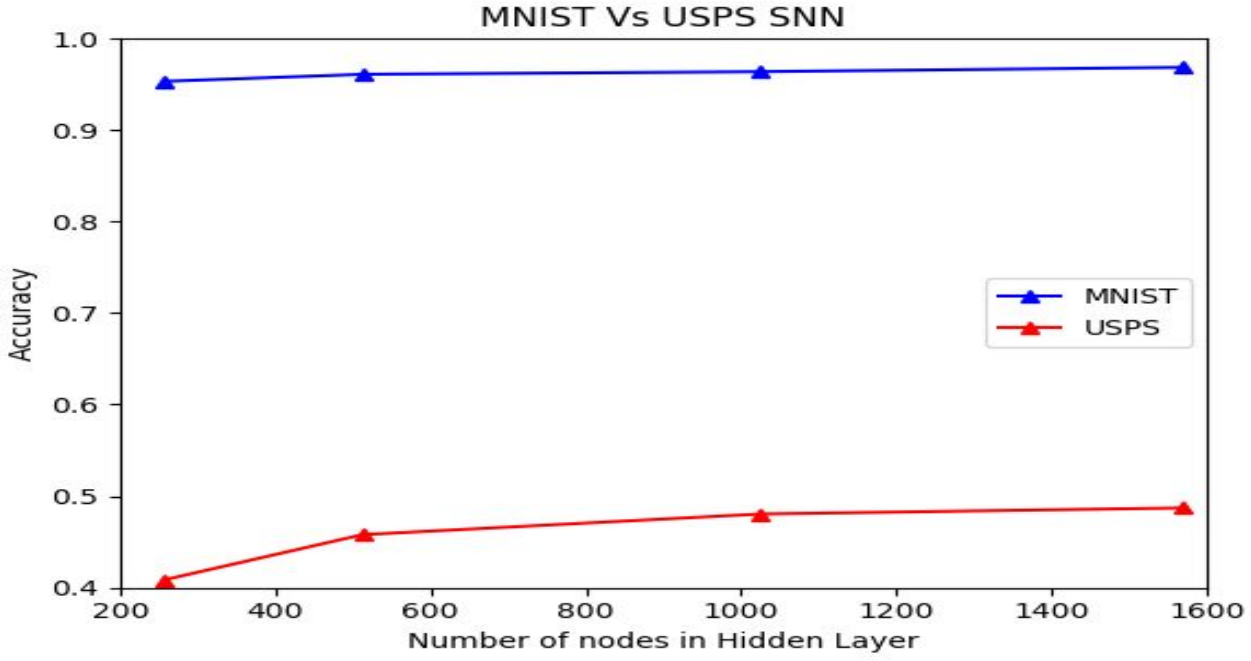




Single NN Accuracy on test USPS data: 0.48705, Epoch: 20, Learning Rate: 0.01
No. of units hidden layer: 1568

It supports No free lunch theorem as the max value(tuned) parameters of MNIST data does not give the max value for USPS data. The max value for USPS **0.5134** occurs at learning rate 0.01, units in hidden layer is 1568 and Epoch 30.





CNN USPS accuracy: 0.66555

Output:

```
D:\Anaconda3\python.exe D:/Users/Rakesh/PycharmProjects/proj3code/main.py
Extracting MNIST_data/train-images-idx3-ubyte.gz
Extracting MNIST_data/train-labels-idx1-ubyte.gz
Extracting MNIST_data/t10k-images-idx3-ubyte.gz
Extracting MNIST_data/t10k-labels-idx1-ubyte.gz
2017-11-19 23:12:52.529412: I
C:\tf_jenkins\home\workspace\rel-win\M\windows\PY\36\tensorflow\core\platform\cpu_feature_guard.cc:137] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX AVX2
Logistic Accuracy on test MNIST data: 0.924
Logistic Accuracy on USPS data:  0.35745
Single NN Accuracy on test MNIST data: 0.9685
Single NN Accuracy on test USPS data: 0.48705
CNN MNIST test accuracy: 0.9909
CNN USPS accuracy: 0.66555
```