

**NAME: Meghashyam Chennur**

### **Introduction:**

The dataset used for this assignment was the same as that given for the homework. NIST preprocessing programs were used to extract the normalized bitmaps of the handwritten digits of preprinted form. The 32x32 bitmaps are further boxed into 4x4 smaller blocks and these are non-overlapping and then the number of pixels' relevant pixels are counted. This generates a smaller 8x8 matrix where the elements range between 1 – 16. This approach reduces distortions.

The neural network built would read the data and predict the appropriate digit. But the algorithm has to be enhanced by further feeding more data.

### **Design:**

- The input is a 64x1 feature vector.
- I used 2 hidden layers with 64 neurons in each layer.

### **Implementation:**

- I represented the final output as a 10x1 neuron where a '1' at the neuron i corresponds to the output digit i.
- I took the initial weights to be random and then used the backpropagation algorithm to update the weights and learn with every incoming data.
- I haven't added the gradient checking to the code.

### **Experiment:**

- Started with a single hidden layer, later added one more to increase performance.
- Used the test data(optdigits\_test.csv) that was provided for the actual testing of the model.
- The overall performance that I got is 46% accuracy.

### **Conclusions:**

- The initial data selected for the weights determine the overall accuracy but in the long run with training over several inputs, that doesn't matter, but it should be random though.
- The code understandability was difficult. It was not sure which section caused incorrect results.
- I could achieve an overall accuracy of 46%.