Handwritten Digit Recognition

ECE407: NEURAL NETWORKS AND FUZZY CONTROL
PROJECT REPORT

Aman Sardana (13BEC0107) [E1]
Rakshith Manandi (13BEC0390) [E2]
Vinay Gupta (13BEC0749) [E1]

SLOT: E1, E2 | FACULTY: PROF. SANKAR GANESH S.

Introduction

In a generation of computers, the advantages of digitalizing data are vast in terms of storing, processing and access. Yet there are several industries/areas where there is a large amount of handwritten data available/being worked upon and computerizing all of it manually would prove to be tedious task. This project serves as a prototype to read and detect handwritten digits with the help of Artificial Neural Networks. The prototype can be adapted to different datasets and is completely modular (in the sense, it can be made more efficient with improved algorithms).

Using Neural Networks/Machine Learning

We make use of the publicly available Mixed National Institute of Standards & Technology (MNIST) database of handwritten digits. We keep the first 80% data for training & the final 20% data for testing. Training is done using conjugate gradient algorithm (which is an advanced version of gradient descent), and the network is a backpropagation network with the sigmoidal activation function. Backpropagation is a generalization of the delta rule to multi-layered feedforward networks, made possible by using the chain rule to iteratively compute gradients for each layer.

We also make use of (image) pre-processing techniques to bridge the process of obtaining an image (from a camera source) and optimizing it for processing. Basically, the colour images taken from a regular camera are converted to gray-scale and smoothened, and finally *thresholded* appropriately to feed into the network.

Network Design

The MNIST database contains images of size 28 x 28 (=784 in total/image) pixels each. They are 8-bit grayscale images with intensities ranging from 0-255. There are 10 digits (0-9) therefore, 10 output nodes. We use a single hidden layer with 100 nodes in our Backpropagation Neural Network. Additionally, each layer except the output layer has a bias unit with input value fixed at 1. A schematic of the network is shown below:

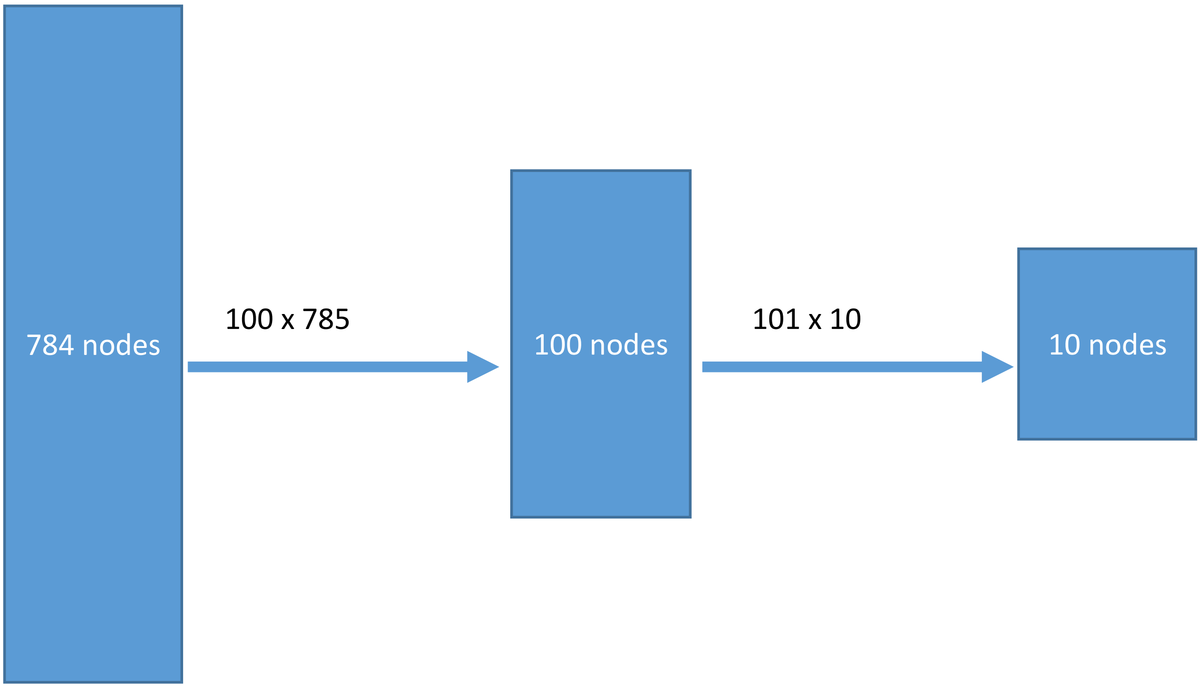

Figure 1: Design of the Backpropagation Neural Network

We implement a log-likelihood cost function for a two-layer neural network (classification), given by:

$$L(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N}\left[y_n \log \hat{y}_n + (1-y_n)\log(1-\hat{y}_n)\right]$$

The weight updates are calculated from the partial derivatives as follows:

$$\Delta w_{ji} = \alpha(t_j - y_j)g'(h_j)x_i \,.$$

where *alpha* is the learning rate, *t* is the target, *y* is the output. *g* is the sigmoid activation function.

The weights of the Neural Network are optimized using the Conjugate Gradient

Function, using Back Propagation Algorithm. We also used regularization with a parameter *lambda* to control & prevent overfitting.

Results & Real Life Experimentation

Our implementation based on the MNIST Database achieves a **95% training accuracy** & **93% Testing Accuracy** which is *appreciably good* for a simple back-propagation network used. We also investigate network performance with real pictures of digits, with pre-processing as required.

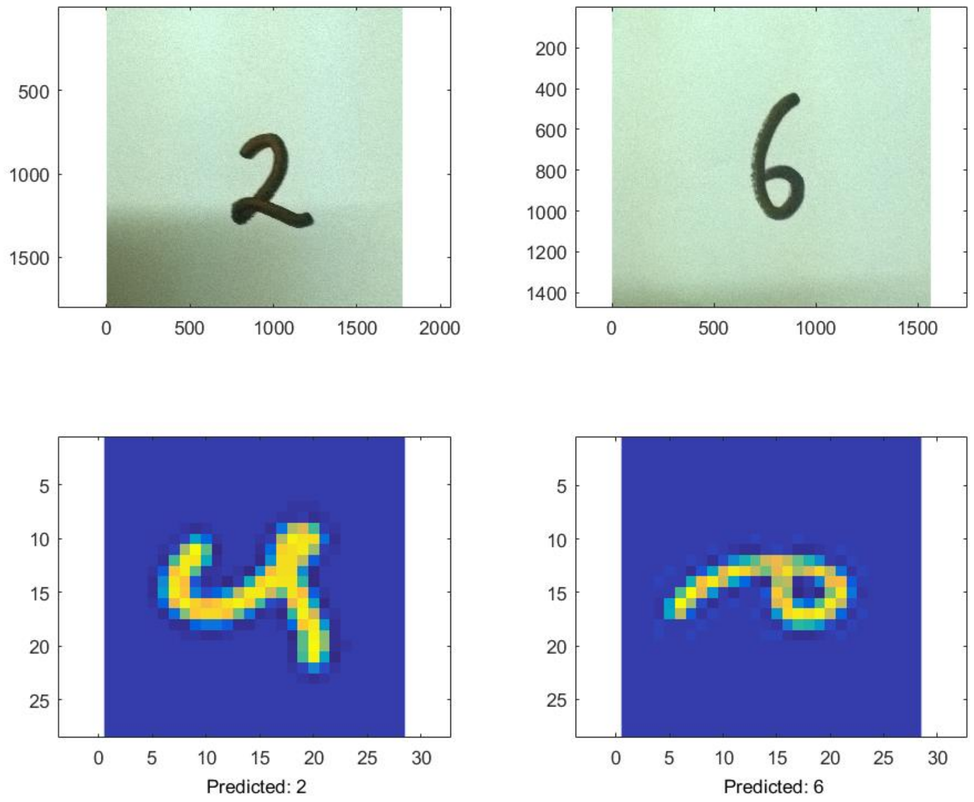

Figure 2: Predictions with real pictures of digits

We believe that this project serves as a prototype for similar but large-scale applications such as Automated Phone Number Scanning & Digital Storing in case of large phone book databases, Scanning & Digitalizing Postal Codes on envelopes/postal mail, Vehicle Number Plate Scanning on Highways etc. In the future, the accuracy & efficiency of the system could be improved further with the use of newer Neural Network/Machine Learning Algorithms, to achieve faster response & access times.

Acknowledgement & References

We would like to express our immense gratitude to **Prof. Sankar Ganesh S**, for the time and effort he put into teaching us while also giving us the chance to apply theoretical concepts learned in the classroom thru this project.

We also greatly benefited from Andrew N G's online Machine Learning course at Coursera & MATLAB's official documentation while building this project.