Digit Categorisation Neural Net

. Introduction

Using the UCI dataset, 8x8 pixels handwritten digits from 0 to 9.

Write an algorithm to most accuractly from the training dataset is able to predict the digits from a training set.

Splitting the original dataset in two parts and performing training and testing using each part in order to perform kfold testing where k=2 to better verify the accuracy of our model. .

The input are integers in the range from 0 to 16 and output is a digit from 0 to 9.

. Neural Networks

A neural network-based classifier, called Multi-Layer perceptron (MLP), was used in the project to classify handwritten digits. The MLP consists of three layers which are the input layer, hidden layer and output layer. Each of these layers contain a certain number of nodes which are also called neurons and each node in a layer is connected to all other nodes to the next layer. This can also be referred to as the feed forward network. The number of nodes in the input layer depends upon the number of attributes present in the dataset.

Using the Sigmoid function allow input signals to pass through the neuron if the input is big enough but it limits the output if the input is too small.

. Weights and bias

In MLP, the connection between two nodes consists of a weight. The number of hidden layers is hard to determine as the numbers are selected experimentally.

Using a dot product function each of the m features in the input layer is multiplied with a weight and summed. Then, the output from the neurons are used as input data that has n features.

The sigmoid function from the neural network introduces non-linearity into the neural network model which means that the output from the neuron, which is the dot product of inputs x and weights w plus bias and then put into a sigmoid function, cannot be represented by a linear combination of the input x. This non-linear function produces a new representation of the original data.

. Categorisation

The training process was done using Backpropagation algorithm to calculate and adjust the weights of the neural network based on the error we get from comparing the labeled result to the neural network output.

Backpropagation is a very useful as it allows us to use multiplayer layers of neurons feeding information to between layers.

. Conclusion

After training our model and settled on the weigths, bias and settling values 26 and 15 for the first and second layer accordingly were found to be good enough as our model was able to score an accuracy of >96% training on the first split and testing on the second and >94% doing the opposite. achieved with the current setup an accuracy.

We can now formulate more accurate answers from more complex systems using neural networks to compress data.

Rodrigo Ferreira
Middlesex University
CST3170 CW2
M00736217