Handwritten Digit Recognition using various Neural Network Approaches

Sakshica¹, Dr. Kusum Gupta²
Banasthali Vidyapith, Jaipur Campus, Jaipur¹²

Abstract: Handwritten digit recognition is one of the important problems in computer vision these days. There is a great interest in this field because of many potential applications, most importantly where large number of documents must be dealed such as post mail sorting, bank cheque analysis, handwritten form processing etc. So a system should be designed in such a way that it is capable of reading handwritten digits and provide appropriate results. This paper presents a survey on various neural network approaches to recognize handwritten digits.

Keywords: Artificial Neural Network (ANN), Handwritten Digit Recognition, Back-propagation (BP), Single Layer Perceptron (SLP), Hopfield Neural Network (HNN).

I. INTRODUCTION

Human beings have been given a natural tendency to recognize images of isolated handwritten digits (0-9). More specifically, the problem is equivalent to find a model, which takes image of handwritten digit as input, and output the predicted class label of the image [2]. Furthermore, the ideas and methods to solve this problem would be very useful in various fields of pattern recognition problems where large volumes of real-world data is used [1].

Handwritten digits recognition is an old and important problem of machine learning. The objective is to recognize images in the same way as the human brain does. The network is composed of a large number of highly interconnected processing elements working in parallel to solve a specific problem. Neural networks learn by example [8].

In this paper the section III gives the brief introduction of the various stages of handwritten digit recognition. Section IV gives a brief introduction of various artificial neural network techniques used for the classification of handwritten digits. Section V gives the conclusion and future scope of the described work.

III. VARIOUS STAGES OF RECOGNITION

The Handwritten digit recognition system has to go through following steps [3]:

- Image acquisition
- Preprocessing
- Segmentation
- Feature extraction
- Classification

Image acquisition implies obtaining the image either by scanning documents or by capturing photograph or by directly writing using mouse [3].

Pre-processing of the image means applying a number of procedures for the purpose of thresholding, smoothing, filtering, resizing, and normalizing so that further algorithm for final classification can be made simple and more accurate [4].

Image segmentation means to divide the image into various segments so that it can be made more meaningful and easy to analyze [3].

Feature Extraction is the process to capture appropriate characteristics of the target object. It is mainly related to dimension reduction. The extracted features from an image are applied as an input to the train classifier like neural network [3].

After obtaining feature vector from the input image next step is to decide which classifier to use for classifying the class of handwritten digits. The most traditional classifier to be used is neural network [3].
IV. CLASSIFICATION USING VARIOUS NEURAL NETWORK APPROACHES

Classification determines the region or area of feature space in which an unknown pattern falls. The various neural network approaches used for classification are described as follows [6]:

A. Single layer perceptron model

The basic structure of a single layer perceptron is described with the help of Fig.2. We can consider it as a single “neuron” which has various input signals that provide an output signal [6].

\[
x_j = \sum_{i=1}^{N} w_{ij} y_i
\]

The value of this output depends on the relative strengths of weighted input signals. The perceptron output can be expressed as:

\[
y(n) = f[w^T(n)x(n)+b]
\]

where, \(w(n) = [w_1(n), \ldots, w_N(n)]\) is the adaptive weight vector \(x(n) = [x_1(n), \ldots, x_N(n)]^T\) is the input signal vector, and \(b\) is the bias term. The most commonly used activation functions are sigmoid & hard limiter. The perceptron weights are updated according to:

\[
w(n+1) = w(n) + \eta [d(n) - y(n)]x(n)
\]

where, \(\eta\) denotes the learning rate parameter less than 1 and \(d(n)\) denotes the desired output or target.

Limitation of Single Layer Perceptron

The single layer perceptron model does not work in case of X-OR, we need more complex network that combine various simple networks together or can use different activation functions.

B. Hopfield Neural Network

Hopfield neural network is a form of recurrent artificial neural network which was invented by John Hopfield. In this network the output of each unit is fed as input to all other units except itself. The Hopfield network is used for pattern storage task. The main motive in a pattern storage task is to store a given set of patterns, and recall the corresponding or closest match pattern when an approximate version of the corresponding pattern is given to the network [8].

For this purpose, the features and their spatial relationship in the pattern need to be stored. The recall of a pattern should also take place when due to noise or distortion or due to variation of pattern generating process the feature and their spatial relations are slightly disturbed [8].

The Hopfield algorithm for storing and recalling a pattern is given below [8]:

Let us consider a network consisting of \(N\) fully connected units with each unit having a hard-limiting bipolar threshold output function. Let \(a_1, a_2, \ldots, a_L\) be the vectors that we want to store. The vector \(\{a_i\}\) are assumed to have bipolar components, i.e., \(a_i = \pm 1, i = 1, 2, \ldots, N\).

1. Initialize the network output with the given unknown input pattern

\[
s_i(0) = a_i, \quad \text{for } i = 1, 2, \ldots, N
\]

where \(s_i(0)\) is the output of unit \(i\) at time \(t=0\).
Back-propagation algorithm is described as follows [7]:
1. Initialize input layer which includes an input for bias, $I_i$, $W_i$, $T_i$, $Y_i$, Where, $I_i$ = input neurons, $W_i$ = random weights, $T_i$ = target values, $Y_i$ =output at each neuron.
2. Propagate activity forward through input layer to output layer.
   
   \[ I => H => O \]
3. Calculate output at each neurons at each layer
   \[ O_i = \sum a_i a_j \quad \text{for} \quad i \neq j \]
4. Apply activation function to neurons and collect final output at each neuron
   \[ Y_i = 1/(1+e^{-a_i}) \]
5. Calculate the error in the output layer
   \[ Error = 1/2 (Y_i - T_i)^2 \]
6. Back propagate the error through layer
   \[ \frac{dE}{dW_i} = \frac{d(Error)}{dW_i} \]
7. Update the weights
   \[ \Delta W_i = \varepsilon (\frac{dE}{dW_i}) + \alpha (\Delta W_{i,1}) \]

Where, $\varepsilon$ = learning rate
$\alpha$ = momentum

**V. CONCLUSION AND FUTURE SCOPE**

In this paper three approaches of neural network have been presented to recognize the handwritten digits. Among the all three approaches Back-propagation is the most successful in the recognition process. Back-propagation is fast and efficient as compared to the other two approaches and gives quick convergence on satisfactory local minima in case of error. It has a simple implementation and computing time is reduced if the weights chosen are small at the beginning.

In future we are expecting to explore the Back-propagation algorithm to make recognition of digits more fast and efficient and improve the overall performance.

**REFERENCES**