

PATTERN RECOGNITION OF NUMERIC NUMBERS USING ARTIFICIAL NEURAL NETWORK

Dr. Sunita, Sunjeev Kumar
DRDO, IRDE, Dehradun, India

Abstract

In this paper, an algorithm is modeled and simulated to recognize the numeric numbers using perceptron learning rule. The perceptron learning rule uses an iterative weight adjustment that is more powerful than the other learning rules. The perceptrons use threshold output function and the McCulloch-Pitts model of a neuron. Their iterative learning converges to correct weight, i.e. the weight that produces the exact output value for the training input pattern. To model and simulate the algorithm, numeric numbers from 0 to 9 are taken.

Keywords

Perceptron, Pattern Recognition, Learning Rule

1. Introduction

Pattern recognition is the association of an observation to past experience or knowledge. With the recent advances in the computing technology many pattern recognition tasks have become automated. As computers and the methods of automatic pattern recognition progress, more and more fascinating applications are being discovered in the fields of finance, manufacturing and medicine. Pattern recognition is the science and art of giving names to the natural objects in the real world. A pattern is an essential arrangement or an ordering, in which some organization of underlying structure are exist. So a pattern can be referred to as a quantitative or structural description of an object or some other item of interest. A set of pattern that shares some common properties can be regarded as pattern class. It naturally involves extraction of significant attributes of the data from the background of irrelevant details. The pattern recognitionsolutions involve many stages such as making the measurements, preprocessing and segmentation. Jayanta Kumar Basuet. al. [3] gave review on pattern recognition using Artificial Neural Network. Hiromichi Fujisawa [2] presented an overview about technical advances in the field of character and document recognition. Bidyut Baran Chaudhuri and Sumedha Bera presented [1] preprocessing in handwritten text OCR involves line, word and character segmentation, which takes several iterations to produce correct output. Perceptron learning rule plays an important role in pattern recognition applications. In this paper, modeling and simulation to recognize the numbers 0, 1, 2, 3 ... 9 are carried out using perceptron learning rule.

2 Perceptron Learning Rule Algorithm

The perceptron learning rule uses an iterative weight adjustment that is more powerful than the Hebb rule. The perceptrons use threshold output function and the McCulloch-Pitts model of a neuron. Their iterative learning converges to correct weight, i.e. the weight that produces the exact output value for the training input pattern. The original perceptron is found to have three layers, Input, Weight and Output units as shown in [Fig. 1].

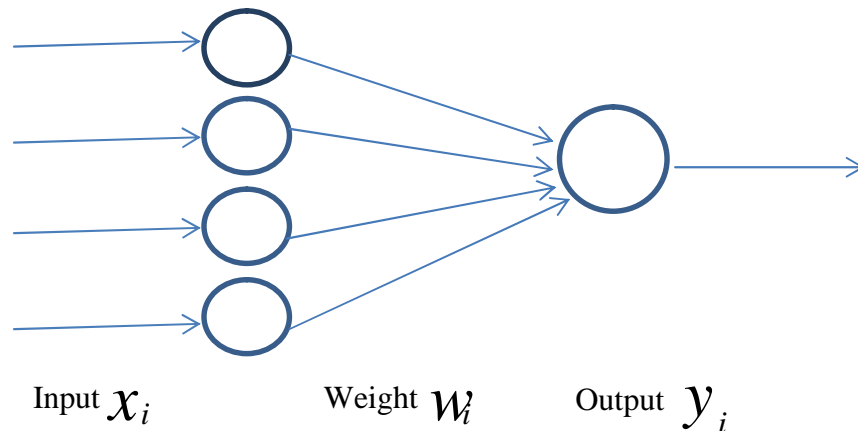


Fig. 1 McCulloch-Pitts model of a neuron

The input and output units have binary activation of +1 or -1 is used for the response unit. All the units have their corresponding weighted interconnections. Training in perceptron will continue until no error occurs. This net solves the problem and is also used to learn the recognition. To start the training process, initially the weights w_i and the bias b_j are set to zero. It is also essential to set the learning rate parameter , which range between 0 to 1. Then the input is presented, the net input is calculated by multiplying the weights with the inputs x_i and adding the result with the bias entity. Once the net input is calculated, by applying the activation function the output of the network y_j is also obtained. This output is compared with the target t_j , where if any difference occurs, we go in for weight updation based on perceptron learning rule, else the network training is stopped. The training algorithm is as follows:

Step1: Initialize the weights (w_i) and biases (b_i). Set the learning rate.

Step2: When stopping condition is false, perform Steps 3-7.

Step3: For each input training pair, do step 4-6.

Step4: Set identity function as an activation function for the input units x_i

$$x_i = s_i \quad \text{for } i=1 \text{ to } n$$

Step5: Compute the activation output of each output unit using binary step function

$$y_{-inj} = b_j + \sum_i x_i w_i \quad \text{for } j=1 \text{ to } m$$

$$y_j = f(y_{-inj}) = \begin{cases} 1, & \text{if } y_{-inj} > \epsilon \\ 0, & \text{if } -\epsilon \leq y_{inj} \leq \epsilon \\ 1, & \text{if } y_{inj} < -\epsilon \end{cases}$$

Step6: The weights and bias are to be updated for $j = 1$ to m and $i = 1$ to n .

If $y_j \neq t_j$ and $x_i \neq 0$, then

$$w_{ij(new)} = w_{ij(old)} + \Gamma t_j x_i$$

$$b_{j(new)} = b_{j(old)} + \Gamma t_j$$

else if $y_j = t_j$

$$w_{ij(new)} = w_{ij(old)}$$

$$b_{j(new)} = b_{j(old)}$$

That is, the biases and weights remain unchanged.

Step7: Test for stopping condition.

3 Methodology to Recognize the Numbers

To recognize the numbers 0, 1, 2, 3 . . . 9, a 5x3 matrix is formed to represent these numbers. For any valid point it is taken as 1 and invalid point it is taken as -1. The Net has to be trained to recognize all the numbers and when the test data is given, the network has to recognize the particular numbers using perception algorithm for several output classes. Input vector X_i (for $i = 0$ to 9) have 15 numbers of neurons to formulate the numbers 0, 1, 2, 3 . . . 9. Representations of all these numbers are shown in table 1.

0	[1 1 1 1 -1 1 1 -1 1 1 -1 1 1 1 1]
1	[-1 1 -1 -1 1 1 -1 -1 1 -1 -1 1 1 1 1]
2	[1 1 1 1 -1 -1 1 1 1 1 1 -1 -1 1 1]
3	[1 1 1 1 -1 -1 1 1 1 1 -1 -1 1 1 1]
4	[1 -1 1 1 -1 1 1 1 1 -1 -1 1 -1 -1 1]
5	[1 1 1 1 -1 -1 1 1 1 1 -1 -1 1 1 1]
6	[1 1 1 1 -1 -1 1 1 1 1 -1 1 1 1 1]
7	[1 1 1 1 -1 -1 1 -1 -1 1 -1 -1 1 -1 -1]
8	[1 1 1 1 -1 1 1 1 1 1 -1 1 1 1 1]
9	[1 1 1 1 -1 1 1 1 1 -1 -1 1 1 1 1]

The weight matrices are initially set to zero. After being trained, the final weight that produces the correct output is obtained. The net has been trained to recognize these numbers by giving

some test data y_i using perceptron learning rule as described in section 2. The number of epochs required for the training is very less.

4 Simulation Result

To simulate the algorithm, maximum numbers of epochs required to train the net are only 54. Network is trained to recognize all the numbers 0, 1, 2, 3 . . . 9. Identity function and binary step function are used as an activation function. The algorithm is simulated in MATLAB.

The results are as follows:

Test Pattern of number 0 is Recognised as 0

Test Pattern 1 is Not Recognised

Test Pattern of number 2 is Recognised as 2

Test Pattern of number 3 is Recognised as 3

Test Pattern of number 4 is Recognised as 4

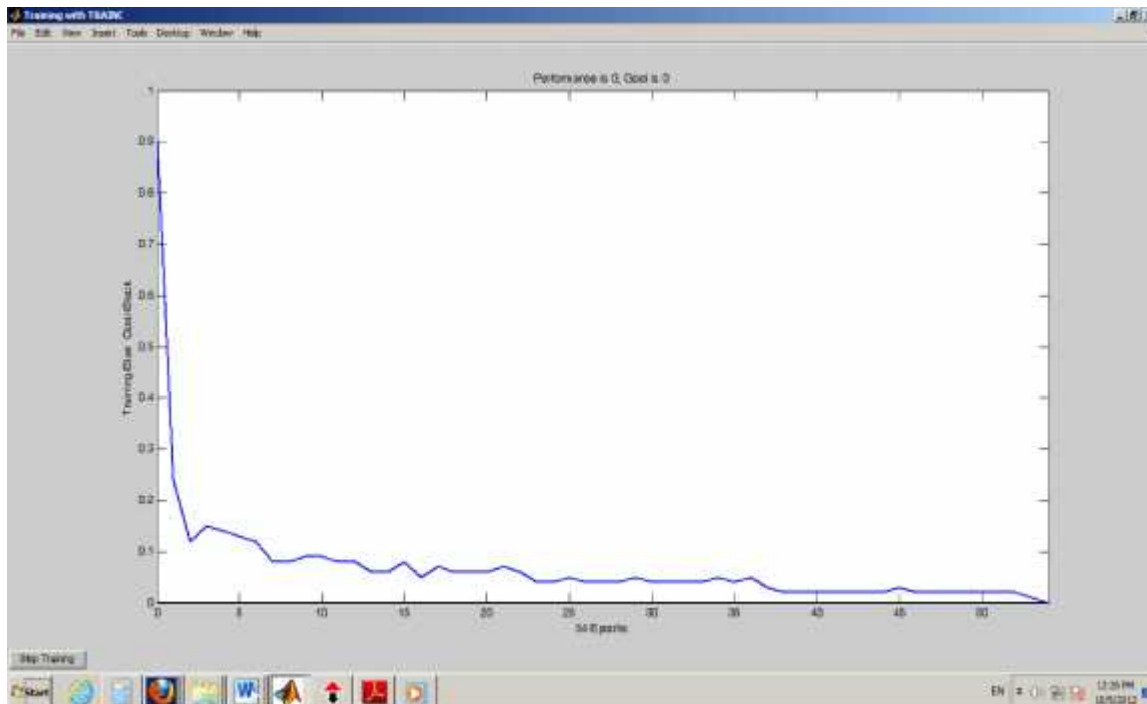
Test Pattern of number 5 is Recognised as 5

Test Pattern of number 6 is Recognised as 6

Test Pattern of number 7 is Recognised as 7

Test Pattern of number 8 is Recognised as 8

Test Pattern of number 9 is Recognised as 9



Graph1: Training Curve (epoch = 54)

5. Conclusion

The algorithm is able to recognize all the numbers from 0 to 9 except number 1 by using different test data only in epoch 54. Identity function is used as an activation function for the input units and binary step function is used as an activation function for the output units.

References

1. Bidyut Baran Chaudhuri and Sumedha Bera: Hand written Text Line Identification in Indian Scripts. *ICDAR' 09 Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*. 636-640
2. Hiromichi Fujisawa. Forty years of research in character and document recognition-an industrial perspective. *Pattern Recognition* 2008, 41, 2435 – 2446.
3. Jayanta Kumar Basu, Debnath Bhattacharyya and Tai-hoon Kim: Use of Artificial Neural Network in Pattern Recognition. *International Journal of Software Engineering and Its Applications*. 2010, 4(2), 23-34.

Book

1. Frank Y. Shih: *Image Processing and Pattern Recognition Fundamentals and Techniques*. A John Wiley & Sons, Inc., Publications. 2010.