Multilayer Perceptron Approach for Character Classification

Anita Venugopal[1], Mukesh Madanan[2] and Akhil Venugopal[3]

[1] Dhofar University, Salalah Postal Code 211, Sultanate of Oman
[2] Dhofar University, Salalah Postal Code 211, Sultanate of Oman
[3]Concordia University,H3G 1MB,Montreal,Quebec,Canada

Abstract

Multilayer perceptron (MLP) is a classification of feed forward artificial neural network which utilizes the processing techniques of human reasoning. Multilayer perceptron consists of numerous linear layers of trained network cells called as perceptron along with threshold function. In this paper, we are evaluating particle swarm and multilayer perceptron computational models. The first approach, particle swarm optimization (PSO) algorithm deals with principle components of artificial neural network model to recognize handwritten characters. Handwritten character strokes are collected as feature vectors and based on these vectors, classification of characters takes place. The second approach, bounding box technique is used by MLP (Multi-Layer Perceptron) neural network. The results are compared and their performance and stability in handwritten recognition is evaluated.

Keywords: *Character Recognition, Multi-Layer Perceptron, Particle Swarm Optimization.*

1 Optical Character Reading

Handwritten characters can be written in several ways. Humans can perceive characters easily without making any errors. Humans perceive characters based on features like, strokes and shapes of the characters. These shapes and characters define characters uniquely are called feature vectors. These feature vectors are used for training the machines so that machines develop the ability to recognize human written characters.Machine learning is gaining popularity and is one of the promising field for researchers. To match human performance, better machine training algorithms need to be devised. There is still a lot way to go where machines need to be designed to predict output to achieve human like accuracy. This could be achieved using supervised learning approach. OCR technology is used to scan handwritten documents, convert it into digital format so that it can be stored in computer and manipulated later. Scanned documents need to be converted to grayscale format. Some of the challenges faced by OCR and handwritten documents are poor image quality, noise in the document, rich set of character set, varied writing styles, presence of compound character, vertical lines of varied length at different position and length, all leads to complexity in identifying written documents. Like how human brain sees and interprets things, OCR's performance also is depended on its input. Optical character recognition gives machines the ability to automatically detect texts in documents.

2 Preprocessing and Recognition

A character can be written in multiple ways by different people, character reading and recognition by OCR and AI has been an area of study by researchers. Machines can identify char-

acters offline or online. Online handwritten provides higher accuracy rate compared to offline recognition techniques.

A character needs to be preprocessed before it is recognized. Preprocessing is an important phase in character recognition as characters to be recognized need to be free from noise and bumps in the document during scanning. The gaps in the broken characters need to be filled. Only integral part of the image is taken for recognition to avoid computational and functional loss. Touching and fused characters must be correctly recognized by segmenting them. To minimize storage space, the image is compressed and normalized.

Preprocessing stage can be divided into five stages. Handwritten or printed characters are scanned and converted to grayscale form. Grayscale median equation is used to convert the image into grayscale as each color contributes differently. The image thus obtained will be brighter and in grayscale form. The image is eroded and dilated to extract only the integral part of the character and eliminate pixels that are not relevant. Skew may be present in the document due to scanning or during writing. The skew is detected and the slant is eliminated using skew detection algorithms.

In this work, Projection profile method is used to correct the slant. Image is then binarized. This process converts the image into foreground and background (1 or 0). Various global thresholding methods are popular for performing threshold. Otsu's method is used in this work to find the threshold value. The minimum of the standard variance of all the thresholds are taken and threshold value. This value is applied on the image and those above or below threshold value is regarded as foreground and background. Thus black and white images are obtained. The image is normalized so that all the characters are of standard size. The output of preprocessing is sent for segmentation. Segmentation is the process of decomposing the image into lines, words, characters and zones. Horizontal and vertical profile approaches are used for detecting line and characters. Fused characters are separated. In this work, score value of characters is calculated. The image thus obtained is normalized so that all characters are of uniform size. Statistical and structural features can be extracted for feature extraction. In this work, 16 segment display is used to segment characters. Equal threshold is applied in all the segments and features are extracted using shadow projection method. The 16 bits extracted as features are crucial and it plays important role in character recognition. Other feature such as, curves, loops, contour and junction point of the image is extracted using 8 connectivity neighborhood method. Each pixel in 3X3 matrix form have eight neighbors.

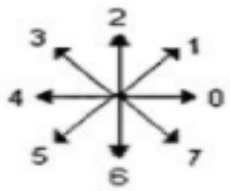

Fig.1. 8-connectivity of neighborhood pixels

Next phase of preprocessing is classification of characters. Now powerful hardware and powerful software methodologies are used for character recognition. Few of the classification techniques such as decision tree, support vector machine, decision tree and neural networks are very popular techniques.

In this paper, fuzzy rules are formulated on the basis of the position of vertical bar components and junction points. 7 fuzzy conditions are formed and based on these conditions char-

acters are classified into 7 classes. In fuzzy system, three elements required are fuzzification, fuzzy rules and defuzzification. Input is given in fuzzification process and based on fuzzy logic developed output is generated. The output generated will consider all the intermediate vague values that are in between the binary values 0 and 1. Computational models like fuzzy logic system and multilayer perceptron networks have the ability to design a system with the ability of human reasoning.

Input are identified in classification. Feature vectors gathered from the input image during feature extraction is compared with the objects in the 7 classes based on the fuzzy rules and patterns are identified. In this stage classifiers are decision making knowledge based system that train the system with the help of classification algorithm [11].

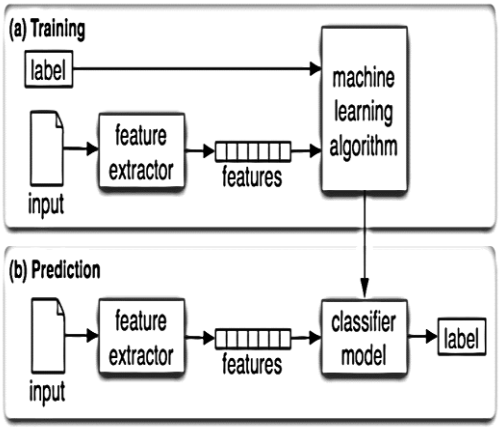

Fig.2. Classifier model

Non-parametric approach is followed by decision tree and knowledge based hierarchical rule algorithms for recognition. Designing and training is simpler compared to other techniques but is complex when different values are interconnected. SVM model is based on group of planes in the dimensional space. Training procedure in SVM is time consuming. ANN works on human reasoning capabilities by using artificial neurons that takes input, does computation and generates output.

In recent years, complex pattern recognition [2, 3, 4, 5, 6, 7] algorithms like particle swarm optimization (PSO) are popular. In this paper, performance of MLP is compared with PSO algorithm in classification problems.

3 Classification

3.1 MLP classifier

Multilayer Perception (MLP) architecture due to its simplicity is the most common neural network used. It is a form of feed-forward neural network. A perceptron in network accepts input, does computation and produces output. Output is checked with input. Weights are randomly generated and added to the neurons during computation till the difference between the input and out is marginal or nil.

Compression takes place in the first layer for the character input. The original image is reduced to fewer bits without affecting the quality. The artificial neurons have the ability to predict values without human intervention. Hidden layers are placed between input and out-

put layers. Till the features are recognized, weights and activation functions are added to the neurons in hidden layer. More hidden layers can be added and trained depending upon the input.

After original data is compressed, computation is performed to adjust weight and error values are calculated in the hidden layer. The data is decompressed and reconstructed to its original form at the output layer. The neurons in the network are trained so as to recognize input [12].

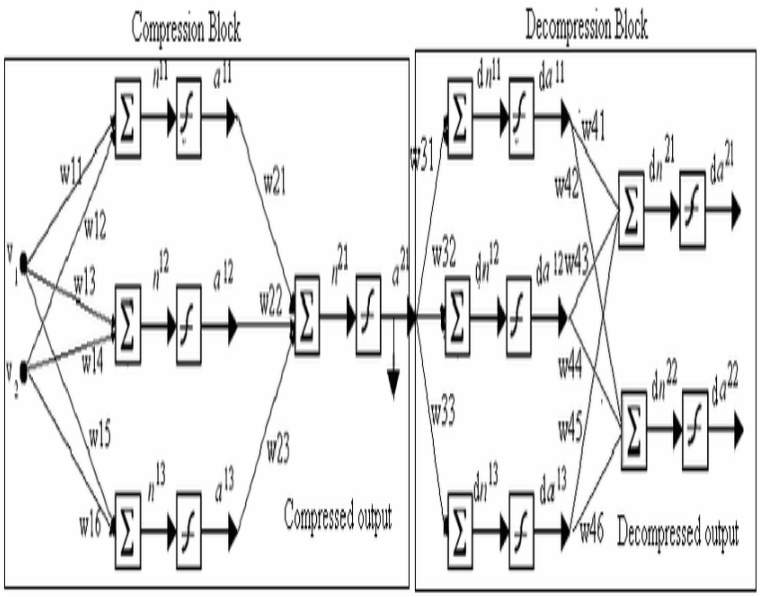

Fig.3. Compression and decompression

Bounding box technique is used and the character to be recognized is enclosed in this box which is divided into many sub boxes.

The neurons in these layers are interconnected to carry the data only in forward direction. It is unidirectional, hence it is called feed forward network. More layers can be added to hidden layer but this architecture cannot forward data in bidirectional mode.

For neuron in second layer after input layer, j = 1, ..., m; input = sum of weights of all inputs in the network. A transfer function f is applied to change given signals to an output using the formula:

$$z_j = f\left(w_{j0} + \sum_{i=1}^{n} w_{ji}x_i\right)$$

Neuron weight = (w1 * x1 + the bias unit)

Bias values are applied to for each neuron. Weights are adjusted till output is accurately recognizes. This is an iterative process and the experiment is carried out in number of cycles. “Learning or training” takes place in this iterative steps [8].

Weights remain fixed in offline learning systems, once the system starts. But in online systems, weights are variable which imparts it with the capability of human reasoning decision

making ability. In this propagation, no loops and feed backs are formed. Sigmoid function is the control tool.

3.2 PSO Classifier

PSO algorithm architecture is based on swarm/animal depicting intelligence concept. Research studies show that its ability to handle class-related issues with improved accuracy rate.

It is based on the assumption of birds as group, a bird as particle; moving synchronically till they find a place to land, 'search space'. In terms of classification, this search space describes algorithm to solve complex optimization problems calculated as:

where,
v -> velocity of particle (vi)
i -> particle
d -> dimension
x -> position of particle (xi)
c1 -> cognitive learning weight
c2 -> social learning weight
r1 & r2 -> random generated numbers (0 &1)

$$w = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} * iter$$

To get better result, in this paper PSO is to be trained by decreasing w from a maximal value w_{max} to a minimal value w_{min} linearly [9,10].

$$v_{id}(t+1) \leftarrow w^{*} v_{id}(t) + c_1 r_1 (p_{id}(t) - x_{id}(t)) + c_2 r_2 (p_{gd}(t) - x_{id}(t)) \quad (1)$$

$$x_{id}(t+1) \leftarrow x_{id}(t) + v_{id}(t+1) \quad (2)$$

Given set of handwritten inputs with Nc classes and database (Z): Z = {z1, z2,..., zp, ...,zNp}; zp = pattern in the Nd-feature space; Np = patterns in Z. PSO encoding steps are:

(1) Initialize individual particles so that it has Nc selected class centroids.

(2) for iter = 1 to $iter_{max}$

(a) for each particle

(i) Using Euclidean distance, each character is assigned to the nearest centroid.
(ii) Based on the following equation, calculate fitness value:

$$f(i) = \frac{w_1 d_{sum}(i) + w_2 \ell(i)}{N_p}$$

(b) Find best position for p_{best} and g_{best}.

(c) Update the classes centroids based on:

$$v_{id}(t+1) \leftarrow w * v_{id}(t) + c_1 r_1 (p_{id}(t) - x_{id}(t)) + c_2 r_2 (p_{gd}(t) - x_{id}(t))$$

$$x_{id}(t+1) \leftarrow x_{id}(t) + v_{id}(t+1)$$

5. Result Analysis

Dataset consists of 60 handwritten characters stored in 7 fields.

The network is designed and trained. When N = 1, 10 patterns were tested and remaining were used for learning purpose. 50 particles were used for PSO training for 500 iterations.

inertia weight w=0.415; w1 = w2 = 0.5;
upper bound velocity = 0.055

Experiment shows performance of PSO with an accuracy rate of 85% for 95% for FFN dataset training.

6. Conclusion

PSO and FFN architecture is proposed in this paper Results are compared. Efforts will continue to further explore and enhance the accuracy level of this algorithm.

References

1. Author, Martin T. Hagan, Howard B. Demuth and Mark Beale, “Neural Network Design”, China Machine Press, 2002.
2. Author, X. Cui, E.T. Potok, and P. Palathingal, “Document Clustering using Particle Swarm Optimization”, Proceedings 2005 IEEE on Swarm Intelligence Symposium, IEEE, June 8- 10, 2005, pp. 185-191.
3. Author, D. I. Falco, D. A. Cioppa, and E. Tarantino, “Facing classification problems with Particle Swarm Optimization”, Applied Soft Computing, Elsevier, Vol. 7, 2007, pp. 652 – 658.
4. Author, B. Fumin, L. Aiguo, and Q. Zheng, “Photo Time-Stamp Recognition Based on Particle Swarm Optimization”, Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI’04), IEEE, September 20-24, 2004, pp. 529-532.
5. D. V. D. Merwe, and A. Engelhrecht, “Data Clustering using Particle Swarm Optimization”, The 2003 IEEE congress on Evalutionary computation, IEEE, Vol. 1 December 8-12, 2003, pp. 215 – 220.

6. Author, S. L. Oliveira, S. A. Britto, and R. Sabourin, “Optimizing Class-Related Thresholds with Particle Swarm Optimization”, Proceedings of International Joint Conference on Neural Networks, IEEE, Montreal, Canada, July 31 - August 4, 2005, pp. 1511-1516.
7. Author, H. G. M. Omran, Particle Swarm Optimization Methods for Pattern Recognition and Image Processin, PhD thesis, University of Pretoria, 2005.
8. Author, Rafid Ahmed Khalil, “Digital image compression enhancement using bipolar back propogation neural networks”, Al Rafidain Engineering, 2007, 15(4), 40–52.
9. Author, R. Eberhart, and Y. Shi, “Particle swarm optimization: Developments, applications and resources”, In Proceedings of the 2001 Congress on Evolutionary Computation. IEEE, Vol. 1, 2001, pp. 81-86.
10. Author, Y. Shi, and R. Eberhart, “A Modified Particle Optimizer”. Proceedings of the 1998 IEEE World Congress on Computational Intelligence, IEEE, May 1998, pp. 69-73.
11. Chapter from, Natural Language Processing with Python, by Steven Bird, Ewan Klein and Edward Loper, 2019 Copyright the authors.
12. Author, Cyril Prasanna Raj P, “Design and Analog VLSI Implementation of Neural Network Architecture for Signal Processing”. European Journal of Scientific Research ISSN 1450-216X Vol.27 No.2 (2009), pp.199-216.