International Journal of Current Trends in Engineering & Technology
www.ijctet.org, ISSN: 2395-3152
Volume: 06, Issue: 04 (July- August, 2020)

Multi-script Handwritten Numerical Recognition Using Multilayer Perceptron Algorithms

Madhulika Goswami[1], Pankaj Savita[2], Shital Gupta[3]
CSE Department, Research Scholar, SORT Peoples University
[1]madhu.priya108@gmail.com, [2]er.pankajsavita@gmail.com, [3]email4sgupta@gmail.com

Abstract: - *Optical character recognition, usually abbreviated to OCR, is the mechanical or electronic translation of images of handwritten, typewritten or printed text into machine-editable text. In this Paper, we presents create the Character Recognition System, in which Creating a Character Matrix and a corresponding Suitable Network Structure is key. The central objective of this project is demonstrating the capabilities of Artificial Neural Network implementations in recognizing extended sets of optical language symbols. The applications of this technique range from document digitizing and preservation to handwritten text recognition in handheld devices. An emerging technique in this particular application area is the use of Artificial Neural Network implementations with networks employing specific guides (learning rules) to update the links (weights) between their nodes. Such networks can be fed the data from the graphic analysis of the input picture and trained to output characters in one or another form. Specifically some network models use a set of desired outputs to compare with the output and compute an error to make use of in adjusting their weights. Such learning rules are termed as Supervised Learning. One such network with supervised learning rule is the Multi-Layer Perception (MLP) model. It uses the Generalized Delta Learning Rule for adjusting its weights and can be trained for a set of input/desired output values in a number of iterations. The project has employed the MLP technique mentioned and excellent results were obtained for a number of widely used font types.*

Keywords: Optical character recognition, Artificial Neural Network, supervised learning, Multi-Layer Perception algorithm.

1. INTRODUCTION

Optical character recognition, usually abbreviated to OCR, is the mechanical or electronic translation of images of handwritten, typewritten or printed text (usually captured by a scanner) into machine-editable text. OCR [2] is a field of research in pattern recognition, artificial intelligence and machine vision. Though academic research in the field continues, the focus on OCR has shifted to implementation of proven techniques. Optical character recognition (using optical techniques such as mirrors and lenses) and digital character recognition (using scanners and computer algorithms) were originally considered separate fields. Because very few applications survive that use true optical techniques, the OCR term has now been broadened to include digital image processing as well. Early systems required training (the provision of known samples of each character) to read a specific font. "Intelligent" systems with a high degree of recognition accuracy for most fonts are now common. Some systems are even capable of reproducing formatted output that closely approximates the original scanned page including images, columns and other non-textual components. The accurate recognition of Latin-script, typewritten text is now considered largely a solved problem. Typical accuracy rates exceed 99% although certain applications demanding even higher accuracy require human review for errors. Other areas—including recognition of hand printing, cursive handwriting, and printed text in other scripts (especially those with a very large number of characters)--are still the subject of active research. Note that accuracy rates can be measured in several ways, and how they are measured can greatly affect the reported accuracy rate. For example, without the use of word context (basically a dictionary of words) to correct "spelling" errors, an error rate of 1% (or 99% accuracy) measured letter-by-letter may result in an error rate of 5% or more (or 95% accuracy), if the measurement is based instead on whether each whole word was recognized with no incorrect letters. Optical Character Recognition (OCR) [5] is sometimes confused with on-line character recognition. OCR is an instance of off-line character recognition, where the system recognizes the fixed static shape of the character, while on-line character recognition instead recognizes the dynamic motion during handwriting. For example, on-line recognition, such as that used for gestures in the Pinpoint OS or the Tablet PC can tell whether a horizontal mark was drawn right-to-left, or left-to-right. On-line character recognition is also referred to by other terms such as dynamic character recognition, real-time character recognition, and Intelligent Character Recognition or ICR. On-line systems for recognizing hand-printed text on the fly have become well-known as commercial products in recent years. Among these are the input devices for personal digital assistants such as those running Palm OS. The Apple Newton pioneered this product. The algorithms used in these devices take advantage of the fact that the order, speed, and direction of individual lines segments at input are known. Also, the user can be retrained to use only specific letter shapes. These methods cannot be used in software that scans paper documents, so accurate recognition of hand-printed documents is still largely an open problem. Accuracy rates of 80% to 90% on neat, clean hand-printed characters can be achieved, but that accuracy rate still translates to dozens of errors per page, making the technology useful only in very limited applications. Recognition of cursive text is an active area of research, with recognition rates even lower than that of hand-printed text. Higher rates of recognition of general cursive script will likely not be possible without the use of contextual or grammatical information. For example, recognizing entire words from a dictionary is easier than trying to parse individual characters from script. Reading the Amount line of a cheque (which is always a written-out number) is an example where using a smaller dictionary can increase recognition rates greatly. Knowledge of the grammar of the language being scanned can also help determine if a word is likely to be a verb or a noun, for example, allowing greater accuracy. The shapes of individual cursive characters themselves simply do not contain enough information too accurately (greater than 98%) recognize all handwritten cursive script. It is necessary to understand that OCR technology is a basic technology also used in advanced scanning applications. Due to this, an advanced scanning solution can be unique and patented and not easily copied despite being based on this basic OCR technology.

2. LITERATURE ON OCR

2.1. Optical character recognition

Optical Character Recognition usually abbreviated to OCR, is the mechanical or electronic translation of images of handwritten, typewritten or printed text (usually captured by a scanner) into machine-editable text. The problem of OCR is fairly simple:-

Input: scanned images of printed text,
Output: Computer readable version of input contents

There are several existing solutions to perform this task for English text. The potential benefits of this approach is its flexibility, since it makes no prior assumptions on the language of From the computational point of view, there are three major tasks involved in our approach to performing OCR. Segmentation: Given input image, identify individual glyphs, Feature Extraction From each glyph image; extract features to be used as input of ANN. This is the most critical part of this approach, since it is not at all clear how this can be done Classification Train the ANN using training sample.

2.1.1 Segmentation

Segmentation is important in two phases of the process. Obtaining training samples the easiest way to obtain training samples is to segment an image and ask a human supervisor to classify each glyph recognizing new image after training as a first step for trying to recognize a new input image, it must be segmented into glyphs. An additional requirement here is to obtain the glyphs in correct order as well. To make this easier, the input image is first divided into lines and then segmented into glyphs

2.2.2 Feature Extraction: Why does it?

Segmented glyphs are binary image matrices. MLP needs moderately low-dimensional input feature vector unfortunately; there is no obvious way to reduce the dimensionality in a way guaranteed to preserve the distinctiveness of glyphs. Classification once the features are extracted, we can go ahead and train a neural network using the training data for which we already know the true classes. After training, recognizing a new scanned image [11], involves, reading in the image, segmenting the image into lines, segmenting each line into glyphs, classify each glyph by extracting the feature set and using the already trained neural network to predict its class given new glyph, classify it. Optical Character Recognition (OCR) is sometimes confused with on-line character recognition). OCR is an instance of off-line character recognition [12], where the system recognizes the fixed static shape of the character, while on-line character recognition instead recognizes the dynamic motion during handwriting On-line systems for recognizing hand-printed text on the fly have become well-known as commercial products in recent years. Among these are the input devices for personal digital assistants such as those running Palm OS. The Apple Newton pioneered this product. The algorithms used in these devices take advantage of the fact that the order, speed, and direction of individual lines segments at input are known. Also, the user can be retrained to use only specific letter shapes. These methods cannot be used in software that scans paper documents, so accurate recognition of hand-printed documents is still largely an open problem. Accuracy rates of 80% to 90% on neat, clean hand-printed characters can be achieved, but that accuracy rate still translates to dozens of errors per page, making the technology useful only in very limited applications. Recognition of cursive text is an active area of research, with recognition rates even lower than that of hand-printed text. Higher rates of recognition of general cursive script will likely not be possible without the use of contextual or grammatical information. For example, recognizing entire words from a dictionary is easier than trying to parse individual characters from script. Reading the Amount line of a cheque is an example where using a smaller dictionary can increase recognition rates greatly. Knowledge of the grammar of the language being scanned can also help determine if a word is likely to be a verb or a noun, for example, allowing greater accuracy. The shapes of individual cursive characters themselves simply do not contain enough information to recognize all handwritten cursive script. For more complex recognition problems, intelligent character recognition systems are generally used, as artificial neural networks can be made indifferent to both affine and non-linear transformations.

2.3 Types of Recognition Engines

2.3.1 Optical Character Recognition (OCR)

OCR engines turn images of machine-printed characters into machine-readable characters. Images of machine-printed characters are extracted from a bitmap. Forms can be scanned through an imaging scanner, faxed, or computer generated to produce the bitmap. OCR is less accurate than optical mark recognition but more accurate than intelligent character recognition.

2.3.2 Intelligent Character Recognition (ICR)

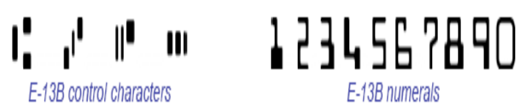
ICR reads images of hand-printed characters (not cursive) and converts them into machine-readable characters. Images of hand-printed characters are extracted from a bitmap of the scanned image. ICR recognition of numeric characters is much more accurate than the recognition of letters. ICR is less accurate than OMR and requires some editing and verification. However, proven form design methods outlined later in this paper can minimize ICR errors.

2.3.3 Optical Mark Recognition (OMR)

OMR technology detects the existence of a mark, not its shape. OMR forms usually contain small ovals, referred to as 'bubbles,' or check boxes that the respondent fills in. OMR cannot recognize alphabetic or numeric characters. OMR is the fastest and most accurate of the data collection technologies. It is also relatively user-friendly. The accuracy of OMR is a result of precise measurement of the darkness of a mark, and the sophisticated mark discrimination algorithms for determining whether what is detected is an erasure or a mark.

2.3.4 Magnetic Ink Character Recognition

MICR is a specialized character recognition technology adopted by the U.S. banking industry to facilitate check processing. Almost all U.S. and U.K. checks include MICR characters at the bottom of the paper in a font known as E-13B. Many modern recognition engines can recognize E-13B fonts that are not printed with magnetic ink. However, since background designs can interfere with optical recognition, the banking industry uses magnetic ink on checks to ensure accuracy.



2.3.5 Barcode Recognition

A barcode is a machine-readable representation of information. Barcodes can be read by optical scanners called barcode readers or scanned from an image using software. A 2D barcode is similar to a linear, one-dimensional barcode, but has more data representation capability.

3. OCR, Neural Networks and other Machine learning Techniques

There are many different approaches to solving the optical character recognition problem. One of the most common and popular approaches is based on neural networks, which can be applied to different tasks, such as pattern recognition, time series prediction, function approximation, clustering, etc. In this section, we'll review some OCR approaches using Neural Networks (NNs). A Neural Network (NN) is a wonderful tool that can help to resolve OCR type problems. Of course, the selection of appropriate classifiers is essential. The NN is an information-processing paradigm inspired by the way the human brain processes information. Neural Networks are collections of mathematical models that represent some of the observed properties of biological nervous systems and draw on the analogies of adaptive biological learning. The key element of an NN is its topology. Unlike the original Perceptron model, shown by Minsky and Papert to have limited computational capability, the NN of today consists of a large number of highly interconnected processing elements (nodes) that are tied together with weighted connections (links). Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true for NNs as well. Learning typically occurs by example through training, or exposure to a set of input/output data (pattern) where the training algorithm adjusts the link weights. The link weights store the knowledge necessary to solve specific problems [10]. Originated in late 1950's, neural networks didn't gain much popularity until 1980s. Today NNs are mostly used for solution of complex real world problems. They are often good at solving problems that are too complex for conventional technologies (e.g., problems that do not have an algorithmic solution or for which an algorithmic solution is too complex to be found) and are often well suited to problems that people are good at solving, but for which traditional methods are not. They are good pattern recognition engines and robust classifiers, with the ability to generalize in making decisions based on imprecise input data. They offer ideal solutions to a variety of classification problems such as speech, character and signal recognition, as well as functional prediction and system modeling, where the physical processes are not understood or are highly complex. The advantage of NNs lies in their resilience against distortions in the input data and their capability to learn.

3.1 Back Propagation Neural Network

A popular and simple NN approach [1] to the OCR problem is based on feed forward neural networks with back propagation learning [3]. The basic idea is that we first need to prepare a training set and then train a neural network to recognize patterns from the training set. In the training step, we teach the network to respond with the desired output for a specified input. For this purpose, each training sample is represented by two components: possible input and the desired network's output given that input. After the training step is done, we can give an arbitrary input to the network and the network will form an output, from which we can resolve a pattern type presented to the network.

4. Example of OCR-based NN Font learning using Bitmap Image

For example, let's assume that we want to train a network to recognize 26 capital letters, represented as images of 16x16 pixels. One of the most obvious ways to convert an image to an input part of a training sample is to create a vector of size 256 (for our case), containing a "1" in all positions corresponding to the letter pixels and "0" in all positions corresponding to the background pixels. In many NN training tasks, it's preferred to represent training patterns in a so called "bipolar" way, placing into the input vector "0.5" instead of "1" and "-0.5" instead of "0". This sort of pattern coding will often lead to a greater learning performance improvement. For each possible input we need to create a desired network's output to complete the training samples. For the OCR task at hand, it's very common to code each pattern as a vector of size 26 (because we have 26 different letters), placing into the vector "0.5" for positions corresponding to the pattern's type number and "-0.5" for all other positions. After having such training samples for all letters, we can start to train our network. But, the last question is about the network's structure. For the above task we can use one layer of neural network, which will have 256 inputs corresponding to the size of input vector and 26 neurons in the layer corresponding to the size of the output vector. At each learning epoch, all samples from the training set are presented to the network and the summary squared error is calculated. When the error becomes less than the specified error limit, then the training is done and the network can be used for recognition.

4.1 Example of OCR-based NN Font learning Using Feature based Classifiers

The approach described above works fine, but is limited in its extensibility. There are some issues that a generalized, robust NN-based OCR system needs to handle, which include font and scale variations. Giving an NN system bitmaps as input is somewhat problematic since humans don't see characters at the pixel level, nor is the "essence" of a character font conveyed by this pixelized representation. When there are considerable bitmap variations in the definition of each font character, a better set of inputs to represent the data would be a set of classifiers, computable from the bitmap images, such that these classifiers are invariant to changes in font and point size. Such classifiers might include topological characteristics, such as Euler number, compactness, and geometric properties, e.g., concave up. Of course, these features now need to be computed from the input images and given as input to the NN system. In addition, the system is invariant to changes in font and point size, so it cannot classify beyond labeling an input bitmap as say an "e", when we may want additional information such as the font and point size, e.g. "e", point size: 12, font: Times Roman. The point is that features typically provide some level of invariance, but at the same time, limit the degree of recognition.

5. Introduction to Techniques for English OCR

5.1 Artificial Neural Network

An artificial neural network (ANN) [4], usually called "neural network" (NN), is a mathematical model or computational model that tries to simulate the structure and/or functional aspects of biological neural networks. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. In more practical terms neural networks are non-linear statistical data modeling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data. Neural networks have seen an explosion of interest over the last few years, and are being successfully applied across an extraordinary range of problem domains, in areas as diverse as finance, medicine, engineering, geology and physics. Indeed, anywhere that there are problems of prediction, classification or control, neural networks are being introduced. This sweeping success can be attributed to a few key factors:

Power: Neural networks are very sophisticated modeling techniques capable of modeling extremely complex functions. In particular, neural networks are nonlinear[r. For many years linear modeling has been the commonly used technique in most modeling domains since linear models have well-known optimization strategies. Where the linear approximation was not valid (which was frequently the case) the models suffered accordingly. Neural networks also keep in check the curse of dimensionality problem [8] that bedevils attempts to model nonlinear functions with large numbers of variables.

Ease of use: Neural networks learn by example. The neural network user gathers representative data, and then invokes training algorithms to automatically learn the structure of the data. Although the user does need to have some heuristic knowledge of how to select and prepare data, how to select an appropriate neural network, and how to interpret the results, the level of user knowledge needed to successfully apply neural networks is much lower than would be the case using (for example) some more traditional nonlinear statistical methods.

5.2. Algorithm

5.2.1 Multilayer perceptron Algorithm

A multilayer perception is a feed forward artificial neural network model that maps sets of input data onto a set of appropriate output. It is a modification of the standard linear perceptron in that it uses three or more layers of neurons (nodes) with nonlinear activation functions, and is more powerful than the perceptron in that it can distinguish data that is not linearly separable, or separable by a hyper plane. To capture the essence of biological neural systems, an artificial neuron is defined as follows:

It receives a number of inputs (either from original data, or from the output of other neurons in the neural network). Each input comes via a connection that has a strength (or weight); these weights correspond to synaptic efficacy in a biological neuron. Each neuron also has a single threshold value. The weighted sum of the inputs is formed, and the threshold subtracted, to compose the activation of the neuron (also known as the post-synaptic potential, or PSP, of the neuron). The activation signal is passed through an activation function (also known as a transfer function) to produce the output of the neuron. If the step activation function is used (i.e., the neuron's output is 0 if the input is less than zero, and 1 if the input is greater than or equal to 0) then the neuron acts just like the biological neuron described earlier (subtracting the threshold from the weighted sum and comparing with zero is equivalent to comparing the weighted sum to the threshold). Actually, the step function is rarely used in artificial neural networks, as will be discussed. Note also that weights can be negative, which implies that the synapse has an inhibitory rather than excitatory effect on the neuron: inhibitory neurons are found in the brain is describes an individual neuron. The next question is: how should neurons is connected together? If a network is to be of any use, there must be inputs (which carry the values of variables of interest in the outside world) and outputs. Inputs and outputs correspond to sensory and motor nerves such as those coming from the eyes and leading to the hands. However, there also can be hidden neurons that play an internal role in the network. The input, hidden and output neurons need to be connected together. A typical feed forward network [6] has neurons arranged in a distinct layered topology. The input layer is not really neural at all: these units simply serve to introduce the values of the input variables. The hidden and output layer neurons are each connected to all of the units in the preceding layer. Again, it is possible to define networks that are partially-connected to only some units in the preceding layer; however, for most applications fully-connected networks are better.

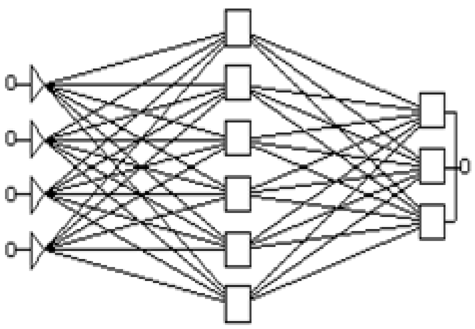


Figure 5.1 Typical Feed forward Network

The Multi-Layer Perceptron Neural Network is perhaps the most popular network architecture in use today. The units each perform a biased weighted sum of their inputs and pass this activation level through an activation function to produce their output, and the units are arranged in a layered feed forward topology. The network thus has a simple interpretation as a form of input-output model, with the weights and thresholds (biases) the free parameters of the model. Such networks can model functions of almost arbitrary complexity, with the number of layers, and the number of units in each layer, determining the function complexity. Important issues in Multilayer Perceptrons (MLP) design include specification of the number of hidden layers and the number of units in each layer [7].

6. Conclusion and Future Work

At the current stage of development, the software does perform well either in terms of speed or accuracy but not better. It is unlikely to replace existing OCR methods, especially for English text. However, it does show some promise. The basic idea of using extracted features to train an ANN seems to work although the success rate is not impressive, it could have been worse. There are several possible changes that the current bottleneck for speed is the feature extraction stage. With some work, it should be possible to speed this up considerably by re implementing it in VB.NET. The other obvious step is to increase the training data set. This requires some effort, but clearly more training data will lead to a more robust and accurate ANN.

References

[1]. Bigus, Joseph P, “Data Mining with Neural Networks”, McGraw Hill, 1996. 24. Wenke Lee, Sal Stolfo and Kui Mok. ‘CA, May 1999.
[2]. Chaudhuri, B.B., Pal, U., "A complete OCR system".
[3]. Hastie, T., Tibshirani, R., and Friedman, J. The Elements of Statistical Learning, Springer 2001
[4]. Neural Network, A. CS229 Lecture Notes, Fall 2008.
[5]. Suzuki et al. INFTY - An Integrated OCR System for Mathematical Documents, DocEng '03, Grenoble, France. 2003.
[6]. Muller et al. An Introduction to Kernel-Based Learning Algorithms, IEEE Transactions on Neural Networks, Vol. 12, No. 2, March 2001.
[7]. Barber, D. Learning from Data: Dimensionality Reduction, 2004.
[8]. Shlens, J. A Tutorial on Principle Component Analysis, Institute for Nonlinear Science, UCSD, 2005
[9]. Dailey, M. Principal Component Analysis Tutorial, Asian Institute of Technology, 2008.
[10]. Le Cun, Y., Bottou, L., and Hanger, P. (1998). Gradient-based learning applied to document recognition, Proceedings of the IEEE, 86(11), 2278-2324.
[11]. Mardia, Kanti V. "Statistics and Images", Vol 1, Vol 2 Cartax Publishing Company, Oxford, UK, 1995.
[12]. T. Matsuoka, H. Hamada and R. Nakatsu, "Syllable Recognition Using Integrated Neural Networks," in Proceedings of the IEEE, IJCNN, Vol. I, pp. 251-258, 1989.