

A Simple Classification of Binary Document into Vector Image or Scalar Image using Feed Forward Neural Networks with Back Propagation Training

G.Sudha
Madurai Kamaraj University College,
Madurai, TamilNadu, India.

ABSTRACT

This paper describes a simple method to classify the binary document into scalar type or vector type. Any image can be basically classified into

1. Vector image
2. Scalar Image.

Vector images are represented as display file containing set of drawing primitive commands (like move, line, curve, rectangle, circle etc) along with related parameters (like coordinates, angle, radius etc.). Scalar images are represented as pixel information ex. Bitmap. Thus memory required to store vector image is very less than of scalar image. Vector image is best to represent text since letters of text are generally formed by set of primitive drawing commands. If the vector image portions of a binary document are identified, then it can be represented as display file by the display file Generating-programs. Remaining portion of the file i.e. scalar portion can be compressed and kept/sent separately.

Initially the binary document images are segmented using Constrained Run Length Algorithm (CRLA) which splits the entire binary image into labeled segment blocks. Next step is to calculate the features for each block according to the block size and data transition (zero-to-one / one-to-zero). Next, the block – features are fed into input layer of Feed Forward Neural Network (FFNN) and it is trained using back propagation method to train the network to classify content of each block into scalar or vector type. Hidden layer and neurons in hidden layer can be developed using constructive approach. FFNN training can be improved using momentum, learning rate and bias if necessary.

General Terms

Neural Networks for Image Classification.

Keywords

Neural Network: Back Propagation (BP), Feed Forward Neural Networks, Image Processing: Block run, Block Features, Constrained Run Length Algorithm (CRLA).

1. INTRODUCTION

Most of the Paper based document contains text, graphs, tables, and other graphical information. Such type of paper based information is converted routinely done for records management, automated document delivery, document archiving, distribution and other related applications. Binary document processing

includes scanning, displaying, quality assurance, image processing etc. A binary image refers to two-dimensional binary function

$$F(i,j) = 0 \text{ or } 1$$

Where, i is Horizontal Co-ordinate of the pixel
 j is Vertical Co-ordinate of the pixel

The origin starts at top-left-corner of the image. A Block in binary document image is connected component (BLOb). Each block is represented as collection of block-runs that are 8-connected as shown in the following picture

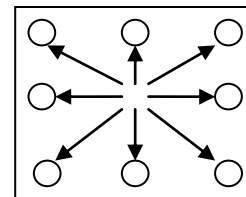


Fig.1 : A connected block run

Vector type of data is usually present as text, line, chart, graph, basic drawing shapes in a document. These vector types of data can be added with additional information like shading, color, grayscale, style etc. Block data type information is useful for Optical Character Recognition (OCR) systems because it improve the accuracy rate as well as speed up the recognition process by handling only the textual data block.

1.1. Biological Neuron and Artificial Neuron:

The fundamental processing element of a neural network is a neuron. This building block of human awareness encompasses a few general capabilities. Basically, a biological neuron receives inputs from other sources (like sensory organs), combines them in some way, performs a generally nonlinear operation on the result, and then outputs the final result. Natural neurons have the same four basic components. These components are known by their biological names - dendrites, soma, axon, and synapses. Dendrites are hair-like extensions of the soma which act like input channels. The soma processes these incoming signals over time. The soma turns processed value into an output which is sent out to other neurons through the axon and the synapses.

The following figure 2 shows the relationship of these four parts.

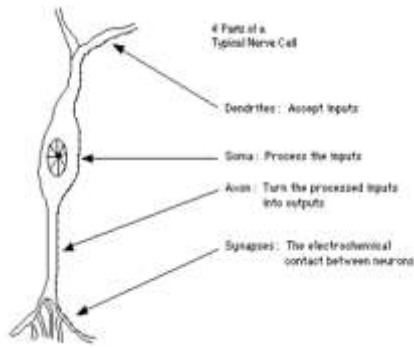


Fig.2: Biological neuron

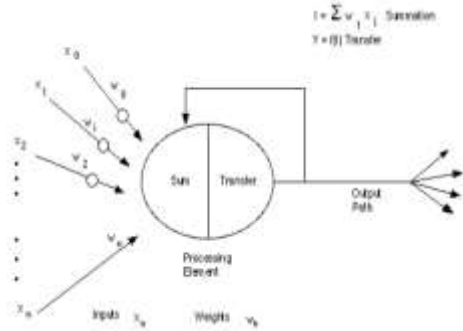


Fig.3: Artificial Neuron

The simulated - artificial neuron is represented as shown in Figure.3. Various inputs to the artificial neural network are represented by the mathematical symbol, $x(n)$. Each of these inputs is multiplied by a connection weight which are represented by $w(n)$. In the simplest case, these products are simply summed ($\sum X_i W_i$), fed through a transfer function to generate a result, and it is the output of the neuron.

Transfer function is used to turn the sum ($\sum X_i W_i$) number into a real output such as, zero or a one, a minus one or a one, or some other number, which is smaller in magnitude. The transfer functions that are commonly supported are sigmoid, sine, hyperbolic tangent, etc. This transfer function also can scale the output or control its value via thresholds. The result of the transfer function is usually the direct output of the processing element.

1.1. Feed Forward Neural Network With Back Propagation Training Method

The Feed Forward Back-Propagation architecture was developed in the early 1970's by several independent sources (Werbor; Parker; Rumelhart, Hinton and Williams). Currently, this synergistically developed back-propagation architecture is the most popular, effective, and easy to learn model for complex, multi-layered networks and suitable for different types of applications.

The typical back-propagation network has an input layer, an output layer, and at least one hidden layer. There is no theoretical limit on the number of hidden layers but typically there is just one or two. The input and output layers indicate the flow of information during recall. Recall is the process of presenting the input data into a trained network and receiving the answer. Back-propagation is not used during recall, but only when the network is learning a training set. The number of layers and the number of

processing element per layer has an impact over the learning speed and recall process.

Number of Input – Output patterns are created in order to train the network and it is called as test / training set. The test set is split into Set_{Train} and Set_{Test} , which are used to train the network and test the network respectively. Training inputs are applied to the input layer of the network, and desired outputs are compared at the output layer. During the learning process, a forward sweep is made through the network, in which the input is transformed by hidden layer neurons and reaches the output layer. The difference between the output of the final layer and the desired output is back-propagated to the previous layer(s), usually modified by the derivative of the transfer function, and the connection weights are normally adjusted using the Delta Rule. This process proceeds for the previous layer(s) until the input layer is reached. Grippo [4] have explained that application of training pattern can be (i) batch mode – where error is accumulated after applying all training patterns (2) on-line/ pattern mode – where, network is trained for each single pattern. And he concluded that batch mode is best for supervised learning.

There are many variations to the learning rules for back-propagation network. Different error functions, transfer functions, and even the modifying method of the derivative of the transfer function can be used. The concept of momentum error was introduced to allow for more prompt learning while minimizing unstable behavior. Thus the error function is modified so that a portion of the previous delta weight is fed through to the current delta weight. This acts, in engineering terms, as a low-pass filter on the delta weight terms since general trends are reinforced whereas oscillatory behavior is canceled out. This allows a low, normally slower, learning coefficient to be used, but creates faster learning. Another technique that has an effect on convergence speed is to only update the weights after many pairs of inputs and their desired outputs are presented to the network, rather than after every presentation. This is referred to as cumulative back-propagation because the delta weights are not accumulated until the complete set of pairs is presented. The number of input-output pairs that are presented during the accumulation is referred to as an epoch. This epoch may correspond either to the complete set of training pairs or to a subset.

2. THE PROPOSED METHOD FOR IMAGE CLASSIFICATION

The process consists of four steps:

1. Block Run Lengths Detection: This step segments the entire binary document into blocks using CRLA method.
2. Block Labeling: This step labels each segmented blocks using component labeling procedure. This label is used to identify a block.
3. Block Features Calculation: This step calculates the features needed for classification for each block.
4. FFNN construction and Training: This step selects training data and test data, models the network, learns and classifies the block into vector / scalar image block based on the block feature.

2.1. Block Run Length Detection

A binary image can be segmented into blocks using CRLA developed by F.M.Wahi and et.al. [2]. Binary 0's and 1's

represents white and black pixels respectively. CRLA uses the predefined constraint C, which replaces 0's by 1's if the number of adjacent 0's is less than or equal to C. Ex. If C=3, and consider the following bitstring

```
00111000100001111100110000011
would become
11111111100001111111110000011
```

This bit string operation is applied to each row and each column to a bitmap of a binary document image. Constraint used for horizontal processing (row wise) is termed as C_{hor} . Similarly C_{ver} denotes the constraint to be used in vertical (Column) processing. In addition to C_{hor} and C_{ver} , two more constraint C_{hs} (horizontal smoothing constraint), C_{vs} (Vertical smoothing Constraint) can be used to omit the space between images / line spacing in case of text. Now apply logical AND operation to the resultant of C_{hor} and C_{ver} .

2.2. Block Labeling

A block is collection of block run. A block run is 8-connected as given in introduction and it is assigned a label. Block run belonging to same block connected with block and labeled same as black run. Number of block run in each block may vary because the application of smoothing constraint removes the blank block runs in the previous step.

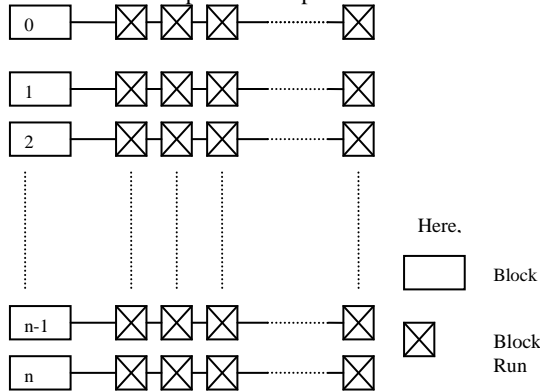


Fig.4: Block structure

2.3. Block Features Calculation

The features of each block in a binary image are calculated based on coordinates as block's extremities. The coordinates of a block's extremities provides information about the approximate size of a block in binary image. Extreme coordinates of each block is defined with respect to the origin of binary image.

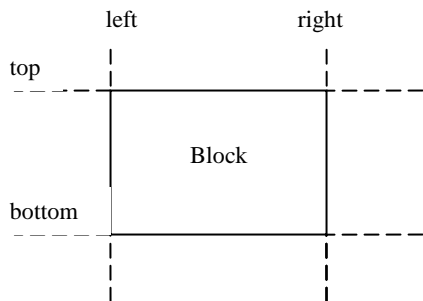


Fig 5: Block Extrimities

Block can be represented by (Xstart,Ystart) and (Xend,Yend).

Where ,

Ystart denotes Minimum left Column i.e. left most pixel of the block

Yend denotes maximum right column i.e. right most pixel

Xstart denotes minimum top row i.e. top most pixel

Xend denotes maximum bottom row i.e. bottom most pixel.

Now Let us discuss the general qualities possessed by vector and scalar image.

Property	Vector Image	Scalar Image
Height of the block	< (in case of text)	>
Total number of pixel used over the block area	<	>
Data Transition (zero to one / one to zero)	<	>

Table 1: Properties of vector and scalar portion of the image

Based on the properties of the image type we can derive the following block features

Feature 1: Height of each block

$$H = Dx = (Yend - Ystart) + 1$$

Width of each block

$$W = Dy = (Xend - Xstart) + 1$$

(Here, the H & W are added with additional one, since first block starts with value zero).

Feature 2: Ratio of height to width of each block

$$E = Dx / Dy$$

Feature 3: Ratio S, denoting (information coverage of pixel in the block) number of black pixel with respect to block area

$$S = BC / (Dx * Dy)$$

Where BC is number of block pixel in the block

Feature 4: The ratio R of the mean horizontal length of black runs of the original data within each block

$$R = DC / TC$$

Feature 5: The product HR of the block height and the ratio R

$$HR = H * R$$

Feature 6: The product ER of the ratio E and the ratio R

$$ER = E * R$$

Feature 7: The product SR of the ratio S and the ratio R

$$SR = S * R$$

The seven block features are normalized to have unit length and are adjusted to be within the [-0.5, 0.5] range.

2.4. Constructing FFNN (using constructive approach) and training

Back-propagation (BP) is a multi-layer neural network using sigmoidal activation functions. The network is made up of an input layer, hidden layers, and an output layer and nodes in each layer are fully connected to those in the layers above and below. Each connection is associated with a synaptic weight.

As Abid.S, Fnaiech.F and Nahim.M[3] states that gradient descent method which is based on the least squared error or Mean Square Error (MSE) between the desired and the actual response of the network along with derivate of activation function will improve the learning. During the training process, the synaptic weights are adjusted in such a way that the actual response gets increasingly closer to the desired response. When an input pattern vector of a training data set is presented to the input layer, the sigmoidal activation values (layers' responses) are propagated forward through the network from the input layer to the output layer. The network's actual response is obtained at the network output layer. The difference between the actual and the desired response is then used to modify the network's weights backward through the network from the output layer to the input layer using a gradient descent method.

To improve the learning speed and control the network we can use

Learning Rate: A portion of weight needed to be adjusted.

Momentum: Fraction of previous weight adjustment that will be added to current weight adjustment.

Bias: Fixed neuron in hidden layer with fixed weight to control the network learning acceleration.

The two-layer BP network is implemented with an input layer - seven block features H, E, S, R, HR, ER and SR, an output layer - one binary output representing vector or scalar image.

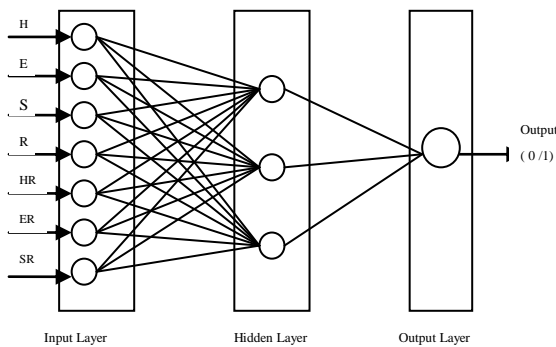


Fig 6: Two Layer Feed Forward Neural Network

The hidden layer is developed using “constructive approach”.

Step 1 : Create 7 neurons in input layer, one neuron in hidden layer and one neuron in output layer. Assign randomized weights to the synapses with in a range like [-5 to +5].

Step 2: Select the training pattern.

Step 3 : Measure the network output

Apply the pattern ‘p’ through the network for each layer and calculate

- i) Summation outputNET = $U_j^s = \sum W_{ji}^s y_i^{s-1}$
- ii) Output (OUT) $y_i^s = f(u_j^s) = 1 / (1 + e^{-u_j^s})$

Step 4: Measure the error

Calculate the desired summation $ld_j^L = f^{-1}(d_j^L)$

Calculate the error $e_j^L = ld_j^L - u_j^L$

Step 5: Modify the weights

For every node ‘j’ of layer s (1 to L), apply the following equation and change the weight

$$\Delta w_{ji}^s = \mu y_i^{s-1} f'(u_j^s) + \mu \lambda y_i^{s-1} e_j^s$$

Step 6: Train the network

Repeat the steps 2 to 5 until the desired accuracy or plateau is reached

Step 7: cascade neuron with existing network

Add a neuron in hidden layer and make a connection with previous layer & next layer neurons. Assign random weights to it. To train the extended network, repeat the steps 2 to 6 for number of epochs (until convergence with desired accuracy is reached).

Note: Here, $f'(y) = \log(y/(1-y))$. μ, λ are optional parameters to improve the learning speed.

3. CONCLUSION

This paper presents a simple method of classification of scalar image and vector image (including text) of binary document. Further the classification can be extended to classify scalar image into color, formats, graphics, dithering, 3D concepts, Transformation, etc. Vector image can be further classified into text font, text properties, text formats, etc. The blocks may be compressed in suitable format according to the classification.

Based on the classification output, corresponding software can take over the information and process accordingly. The transfer of binary document over the network can be split into vector portion – transfer and scalar portion - transfer to speed up the data transfer.

This proposed method can be used in other area of image processing like Hand written text recognition, Component identification from hand drawn image, Language recognition and other related applications.

REFERENCES

- [1] D.X.Le G.Thoma, H.Weschier, “ Classification of Binary images into texture or non-texture data blocks using neural network models”
- [2] F.M.Wahi, K.Y. Wong and R.G. Casey (1982), “Block segmentation and text extraction in mixed text / image documents”, Computer and Image processing 20: 375-390.
- [3] Abid.S, Fnaiech.F and Nahim.M, “ A fast feed forward training algorithm using a modified form of the standard back propagation algorithm” IEEE Transactions on Neural Networks, 12 (2001), 424-430.
- [4] Grippo.L, “Convergent on-line algorithm for supervised learning on neural networks” IEEE Transactions on Neural Networks, 11(2000), 1284-1292.
- [5] Wasserman.P.D. “Neural Computing Theory and Practice”, Van nosrand and Reinhold, NewYork, 1989.
- [6] T. Pavlidis and J. Zhou (1992), Page Segmentation and Classification. Computer Vision Graphics and Image Processing 54(6): 484-496.