A Simulation of Hand Written Characters using Neural Networks Techniques

Hamza A. Ali,
Visiting Professor, C I T E C, University of Aizu, Japan.

Mohamed Hamada
Language Processing Laboratory, University of Aizu, Japan

Key words:
Artificial neural network, neocognitron, character recognition, pattern recognition

Abstract:

The lay out of any character in any language can be considered as a collection of lines and dots. These lines and dots can be looked at as special features or marks, gathered together to give the character its final shape. Although, character shapes and meanings vary from one language to the other, they may be obtained by using the same set of feature patterns.

A statistical study of the feature availability in various characters of one chosen language and their frequency of occurrence is presented. The circuit used was implementing a neural network model based on Fukushima's neocognitron (learning without tutor) for its feature extracting stage.

This study is conducted toward the aim of full and accurate character recognition of hand written script. It must be stated that the training patterns used can be extended further for more concrete recognition rate by increasing the number of training patterns as well as using more pixels at the input layer.

1. Introduction:

The automatic recognition of typed characters has long been implemented. However, research work on hand written characters is still focused on many of the life languages around the world [1,2]. Most difficulties in hand written character recognition arise due to various distortion and noise resulting from personal writing styles, habits, moods, etc. Much interest in the use of neural network has grown tremendously. Artificial Neural networks, ANN's, lend themselves to be highly applicable for character recognition as compared with statistical, syntactical or structural approaches. That is because, ANN's are computing systems having many simple, highly connected processing elements that process information by its dynamic state response to external inputs [3]. Therefore, they have certain characteristics with great similarities to those of biological neural systems [4].

Many approaches are being implemented using ordinary feed forward neural networks, like Fukushima neucognitron model [5,6,7] or feed back, called back-propagation NN's [8] or a combined system of both [9,10].

Generally, Neural Networks, NN's are constructed of many processing elements, as shown in figure (1). Each element may have many inputs but it is limited to one output signal. In layered neural network system, any input signal, which is not an output from another processing element must have been coming from the outside world. The relationship between input and output signals are determined usually by first order ordinary differential equations. Auto-adjustment to the coefficients of the differential equations gives the neural networks its ability to adjust its internal variables. This ability of self-adapting dynamic system that can notify its own response to external forces is the outstanding feature of the NN's. The basic phenomena of neural learning, is that each link has its own small local memory, which stores the result of some previous computation as adaptive coefficient.

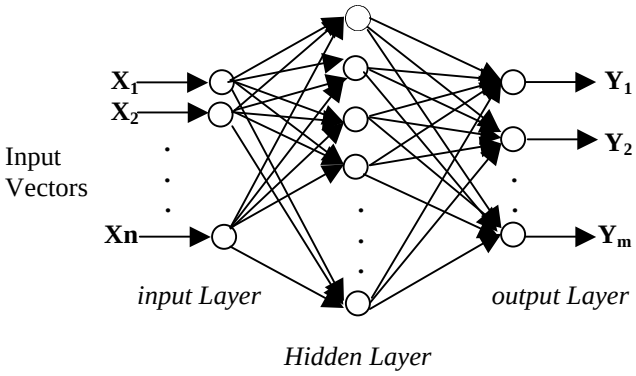


Figure (1) General structure of layered Neural Network

The building block of the NN's is the Neuron, illustrated in figure (2)[11]. It has a set of continuous or discrete inputs, X's, connected through links from previous neurons. Each link has an adaptive coefficient called weight, *a* assigned to it. Each weights might generally be used for amplification, attenuation or possibly changing the sign of the signal in the link. The output of a neuron is calculated by:

$$S_j = \sum_{i=1}^{n} a_{ij} * X_i \quad \ldots (1)$$

X_n , a_{ij} is the weight of the i-th input vector to the j-th neuron.

S_j is further processed by the activation function $f(\ )$ giving the final neuron's output signal Y_j , i.e.

$$Y_j = f(S_j) \quad \ldots (2)$$

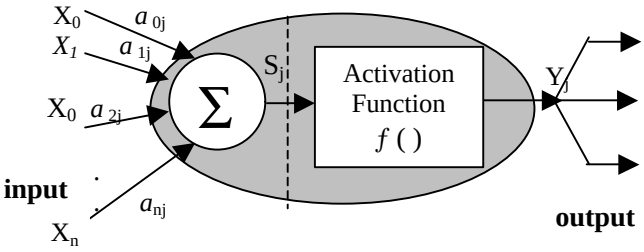


Figure (2) Processing functions inside the Neuron.

The activation function determines the processing inside the neuron. It might be linear or non-linear function depending on the proposed network. However, the function limits the output of the neuron to the desired numerical range. Typically, limiting the output between 0 and 1 or between –1 and +1 for binary or bipolar function, respectively. Numerous numbers of such processing elements forms what is referred to as Artificial Neural Network, ANN. The learning ability of these networks is the basic feature of intelligence [12]. It implies that the processing element somehow changes its input/output behavior in response to the environment. In a similar manner to the way that a child learn to identify various things, NN learns by example [13], i.e. by repeatedly trying to match that set of input data to the corresponding required output. Therefore, after a sufficient number of learning iterations, the network modifies the weights in order to obtain the desirable behavior pattern for new input conditions [3]. ANN's can be classified by their learning scheme as supervised, un-supervised and batch learning which correspond to learning with a tutor, no tutor and encoding, respectively.

In this study, Mathematica [17] is used, as it lends itself for dealing with large number of matrices and matrix operations without the need for considerable memory space and complicated programming. Moreover, mathematical functions can be implemented in neural way as it appears in the books. The nested cell feature of Mathematica system enables us to deal easily with intermediate computations, which makes easier program debugging.

2. Feature Extraction:

NN principle is implemented in this paper using a technique that is based on Fukushima neocognitron model [5]. His complete pattern recognition model, generally consists of cascaded modular structures preceded by input layer. Each modular structure composed of two layers of cells, simple cells and complex cells, successively. The total number of layers was nine and it covers three stages, feature extracting stage, classification stage and recognition stage. However, recent work [9,10] suggested a combined structure consisting of Fukushima neocognitron principle while the classification and recognition stages are based on back propagation technique. This paper adapts the combined structure and concentrates on the feature extracting stage only with the aim of an in-depth study of the feature contents in hand written script.

The Circuit for the feature extracting stage consists of two layers, simple, (S-layer) and complex (C-layer) preceded by an input layer. The input signal is taken as 20 x 20 cells matrix, which corresponds to the eye retina in human visual system. Each of these layers is taken to consist of number of planes equal to the number of features under study. While each plane is a two dimensional matrix of r x r points as detailed in figure (3). The receptive field for the S-planes cells is 3 x 3 while that for the C-plane is 5 x 5. Generally, less care is taken for the edges, leading to reduce the size of the cell planes, in order to concentrate the signal in smaller areas. The principle motivation for dimensionality reduction is that it can help to alleviate the worst effect of the curse of dimensionality [14].

Each cell in the S-planes has excitatory inputs signal $Us_{j,n}$ through connections having weights $\boldsymbol{a}_{j,\nu}$ and inhibitory input signal Vs_n through connection of weight $\boldsymbol{b}_j$. Mathematically, the output signal of the Us, for any cell in S-planes is calculated by equation (3).

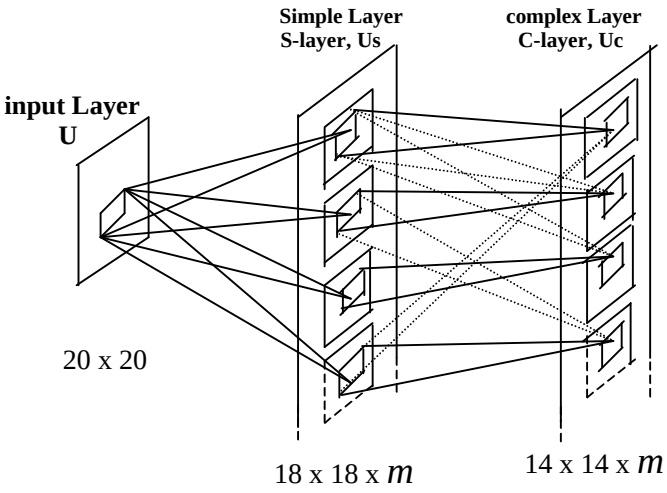


Figure (3): Schematic diagram for the synaptic connections in the feature extracting stage.

$$Us_{j,n} = r.\phi\left\{\frac{1+\sum_{k=1}^{K}\sum_{\nu\,\text{“}A}\boldsymbol{a}_{j,\nu}\,.\,U_{n+\nu}}{1+(r/(r+1)).\boldsymbol{b}_j.Vs_n} - 1\right\} \quad \ldots \quad (3)$$

where: $\phi(x) = \begin{cases} x & \text{if } x >= 0 \\ 0 & \text{if } x < 0 \\ 1 & \end{cases}$

$\boldsymbol{a}_{j,\nu}$ is the strength of the modifiable excitatory synapse coming afferently from cell $U_{n+\nu}$ in the preceding layer, and A denotes the summation range of ν, b_j is the strength of the modifiable inhibitory synapse coming from inhibitory cells V_n. n is two dimensional co-ordinates indicating the position of the cells receptive field center in

the input layer U and k = 1,2,..,K is a serial number indicating the type of feature which the cell extract.

It must be noted that all S-cells in any S-plane have an identical set of input interconnections, therefore, $\boldsymbol{a}_{j,v}$ & $\boldsymbol{b}_j$ do not contain any argument related to the position n of the receptive field.

The parameter r represent the inhibitory signal efficiency. An inhibitory cell Vs-cell sends signal to this Us-cell, yielding an output equal to the weighted root square of signals from the pre-synaptic, U-cells which is proportional to the Euclidean norm of signal sent by the input units, that is:

$$Vs_n = \sqrt{\sum_{k=1}^{K} \sum_{v \in A} C_v . \{ U_{n+v}\}^2} \quad \ldots \ (4)$$

Where: C_v represents the strength of excitatory un-modifiable synapses [15], which is a monotonically decreasing function of |v|, and satisfies:

$K \Sigma C_v = 1$, So that each simple cell is sensitive to a restricted area of the input pattern, i.e. its receptive field. The size of A, which determines the spatial spread of the excitatory input connections of an S-cell (also of a V cell), corresponds to the size of features to be extracted by the S-cell. If the density of local features contained in a pattern is high, the size of A has to be relatively small, but if the local features are spares, the size can be large. The density of the local feature is highly correlated with the complexity of input patterns to be recognized. It is obvious, that the more complex the pattern, the smaller the size of A [15].

As for the C-layer, each C-plane accept input connection from the corresponding S-layer. The output of each complex cell, $Uc_{j,n}$ in the network is mathematically calculated by equation (5).

$$Uc_{j,n} = \Psi\{ [1 + \sum_{ks=1}^{K} L_{j,I} \sum_{v \in D} D_v \, Us_{j,n+v}] / [1 + Vs_n] \} \ldots (5)$$

Where $\psi(x)$ is a function specifying the characteristic of saturation, and is defined by: $\psi(x) = \{ \phi(x)/(\phi(x)+\alpha)\}$, and $L_{j,I}$ indicates that the output of some S-cell planes are joined together and made to converge to a single C-cell plane. The value of $L_{j,i}$ is 1 if the j-th C-cell plane receives signal from the i-th S-cell plane, otherwise it is 0.

The parameter α (>0) determines the degree of saturation of the output. The parameter D_v denotes the strength of the excitatory un-modifiable synapses and is monotonically decreasing function of |v| [16]. Its size, which determines the spatial spread of the fixed excitatory input connection of a C cell, also has a tendency similar to that of A. If the density of features in a pattern is large, the size of D has to be small. Otherwise, detecting the configuration of local features becomes ambiguous.

3. Recognition:

Recognition is achieved by two stages, classification stage and output stage. Back-propagation technique with biased neurons is adopted for these purposes [10,16]. Classification stage corresponds to the hyper-complex layer in the biological visual system and can be represented by two layers, the lower-order hyper-complex U_{LOH} and higher-order hyper-complex U_{HOH}, as shown in figure (4). The neurons of these layers decode the output error tolerated feature of the previous stage and memorize the information about the input patterns, depending on the input features in a fully distributed fashion into its internal structure. Finally the output stage consists of a number of cell-planes (equal to the patterns under consideration), having single cell each. Each neuron receives its input from all planes in the previous layer.

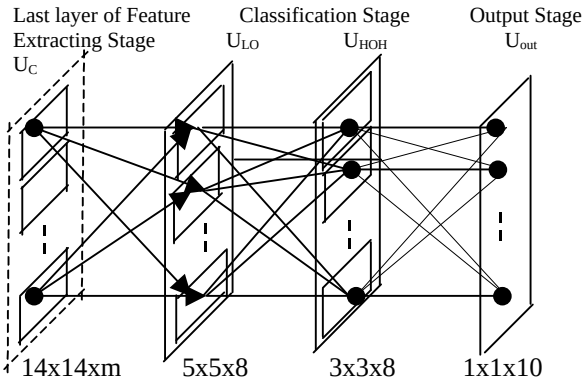


Figure (4): The basic struture of the classification and output stage for the back-propagation recognition model [10].

All the cell's connections are modifiable, but for simplification purposes, cells having comparatively larger output values are chosen as representative for each cell-plane, and only connections to those cells are modified. The adjustment of the weights is achieved by calculating the error, which is the difference between the calculated output and the expected target signal for each training pattern. This error is propagated backward through the layers Uout, UHOH and ULOH. The calculated error δ, is used to compute the changes in the input connection weights of the cells, or the bias weight for each cell in the back-propagation layers. i.e Δ **bias** = $\eta \, \delta(k_i,n_i)$, where η is the learning factor ($0< \eta <1$) and $\delta(k_i,n_i)$ is the calculated error for the n_i-th cell in the k_i-th layer, while i, is the order of the layer in this part of the network (i =3,4,5). This bias difference is added to the weight of any signal. Then the next pattern of training set is presented in the input layer. This process completes one cycle of training, called epoch. This process is repeated until the error is reduced to negligible value, and now the circuit is ready for recognition.

4. Calculations, Results and Discussion:

The feature extraction circuit is trained first by presenting a set of 20 training patterns in the input layer. The group of training patterns, used in this study is shown in the second column of table (1). They are selected to be the most expected pattern feature that can be found in most characters to be considered in this study. They included small lines of different shapes and orientations. One of these patterns is taken at a time in the center of the input layer and the model is trained for a single S-plane. A computer program is

written in the powerful Mathematica system to achieve the training as follows.

First, the initial values for the weights are chosen as random small values, ***a***'s between 0.01 & 0.001 and ***b***'s = 0. Then, the output response is calculated for all cells in that plane using equations (3) and (4). The highest value is chosen as representative of that particular feature pattern, then, the variations in the weights are calculated according to the following:

$$\Delta \boldsymbol{a}_{j,v} = \boldsymbol{q}\, C_v\, U_{n+v} \quad \& \quad \Delta \boldsymbol{b}j = \boldsymbol{q}\, Vs_n \quad \ldots (6)$$

where ***q*** *is a positive constant determining the speed of reinforcement, which shall have sufficiently large value for supervised training (in the order of 104), in order to have fast convergence.*

These variations are used to adjust the old weights, and the output response, then taking the highest as representative cell. The variations in the weights are recalculated again. This completes one epoch. This process is repeated until variations in the weights are less than a predefined small value. Now, this completes the training for one cell plane and the weights values are frozen for that S-plane. This process is repeated for all the training patterns, each pattern is used to train one S-cell plane.

Table (1): The S-planes of the feature extracting stage that have values for the cells > 0.9 for the selected training templates as compared with those for Fukushima model.

S. N	Pat-tern	Responding S-Plane number			S.N	Pat-tern	Responding S-Plane number	
		Fuk-ushima	Com-bined Model				Fukushima Model	Com-bined Model
1	000 **111** 000	1	1		11	000 **11**0 00**1**	11	11 , 19
2	0**1**0 0**1**0 0**1**0	2 , 10	2 , 10		12	**1**00 0**11** 000	12	11 , 12
3	00**1** 0**1**0 **1**00	3	3		13	000 **11**0 00**1**	-	10 , 13
4	**1**00 0**1**0 00**1**	4	4, 19		14	00**1** **11**0 000	-	10 , 14
5	00**1** **11**0 000	5	5		15	000 0**11** 0**1**0	-	15
6	000 0**11** **1**00	6	6		16	0**1**0 **1**00 0**1**0	-	16
7	0**1**0 0**1**0 **1**00	7 , 10	7 , 10		17	**01**0 00**1** 0**1**0	-	17 , 19
8	00**1** 0**1**0 0**1**0	8	8		18	000 **1**0**1** 0**1**0	-	18
9	**1**00 0**1**0 0**1**0	9 , 10	9 , 10, 19		19	0**1**0 **1**0**1** 000	-	19
10	0**1**0 0**1**0 00**1**	10	10		20	000 **11**0 0**1**0	-	20

Now, for each training pattern in the input layer, the response of all the S-planes are calculated. It is found that some S-planes are more responsive to certain feature than others. However, the response can be illustrated by showing the S-planes that have cell values greater than certain amount. Tables (1) and (2) show these results for planes having cell values greater than 0.9 and 0.5, respectively. This clearly indicates that some features activate more than one S-plane, which should be considered when calculating the cell values of the C-planes, i.e. the values of the expected links $L_{j,i}$ between the S-planes and the C-planes of the feature extracting stage. They also show the sensitivity of various S-planes for the training patterns used by Fukushima [15] for comparison reason. However, if only planes with higher cell values (for example, greater than 0.9) are considered, less planes will be activated for each feature pattern.

Table (2): The S-planes of the feature extracting stage that have values for the cells > 0.5 for the selected training templates as compared with those for Fukushima model.

	Pat-tern	Responding S-Plane number			S.N	Pat-tern	Responding S-Plane number	
		Fukushima Model	Combined Model				Fuku-shima	Combined Model
1	000 **111** 000	1, 5	1, 5, 14		11	000 **11**0 0 0**1**	1, 4, 9, 11, 12	1, 4, 9, 11, 12, 14, 17, 18, 19, 20
2	0**1**0 0**1**0 0**1**0	2, 8, 9, 10	2, 8, 9, 10		12	**1**00 0**11** 000	1, 4, 5, 11,12	5, 11, 12, 13, 16, 19
3	00**1** 0**1**0 **1**00	3, 5, 6, 7, 8	3, 5, 6, 7, 8, 14, 15, 17,18		13	000 **11**0 00**1**	-	10 , 13
4	**1**00 0**1**0 00**1**	4, 9, 11, 12	4,9,11,12, 13,16,17, 18, 19, 20		14	00**1** **11**0 000	-	10 , 14
5	00**1** **11**0 000	1, 3, 5, 6	1, 3, 5, 6, 14, 15, 16		15	000 0**11** 0**1**0	-	2, 15
6	000 0**11** **1**00	1, 5, 6, 7, 8	1, 3, 5, 6, 7, 8, 14, 16, 18		16	0**1**0 **1**00 0**1**0	-	3, 4, 7, 9, 11, 13, 16,17, 18, 19
7	0**1**0 0**1**0 **1**00	2, 3, 5, 6, 7, 8, 9, 10	2, 3, 5, 6, 7, 8, 9, 10, 14, 15, 16		17	**01**0 00**1** 0**1**0	-	3, 4, 5, 6, 8, 12, 14,15,16, 18, 19, 20
8	00**1** 0**1**0 0**1**0	2, 3, 7, 8	2, 3, 7, 8, 17, 18		18	000 **1**0**1** 0**1**0	-	3, 4, 6, 11, 13, 15, 16, 17, 18
9	**1**00 0**1**0 0**1**0	2, 4, 9, 10, 12	2, 4, 9, 10, 16, 18, 19, 20		19	0**1**0 **1**0**1** 000	-	3, 4, 5, 7, 8, 9, 12, 14, 16, 17, 18, 19, 20
10	0**1**0 0**1**0 00**1**	2, 4, 8, 9, 10, 11	2, 4, 8, 9, 11, 13, 17, 19		20	000 **11**0 0**1**0	-	2, 19, 20

Another way of showing the sensitivity of each S-plane for any specific feature pattern is the sum of the resulting values for the cells in each plane or more specifically their percentage as shown in table (3). Such measure looks more representative for the effect of each plane on any subsequent calculation.

The values for the C-cell planes in the Complex layer are then calculated using equation (5).

As this study is meant for the search for certain pre-selected features in numerals and characters, different samples of hand written numerals are selected, a sample of which is shown in figure (5). Each of these numerals is placed in the input layer of the model, and the output response for all the feature extraction stage cells are calculated. The training patterns used are numbered from 1 to 20 as was listed in table (1).

A statistical summary of the availability of the included features in the numerals understudy (i.e. 0 to 9) of this study is shown in tables (4) illustrating the number of cells in the S-planes having values > 0.5. This table shows that, all features might be available in all the numerals, but with varying levels or sensitivity. For more accurate calculations, more S-planes are interconnected to the subsequent C-planes, but this would be at the cost of speed. A balance must be achieved between recognition speed and accuracy.

Table (3) The percentage of the total contents of the S-planes cells in the feature extraction stage for the numerals 0 - 9.

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
8.11	2.57	10.71	5.93	10.30	13.68	5.72	14.02	3.84	3.61
8.07	21.61	4.96	8.47	6.33	8.63	12.63	2.39	4.94	8.59
4.40	3.87	8.83	4.30	9.53	4.30	3.55	9.33	5.91	5.00
4.40	.66	1.41	4.30	.07	2.13	3..39	0.76	5.91	5.16
4.91	2.72	6.57	4.30	6.38	7.57	4.35	7.95	3.92	3.91
4.55	1.36	6.72	4.02	7.22	5.54	4.29	7.75	5.13	4.20
4.55	6.51	7.24	5.26	7.85	3.94	4.38	6.65	5.86	5.31
4.91	11.31	5.27	5.62	5.90	4.62	6.38	4.44	4.85	5.42
4.91	10.29	3,01	6.10	3.00	4.02	6.49	2.09	4.85	5.79
5.34	16.77	3.75	6.66	4.43	4.63	7.07	3.10	3.97	6.76
4.56	0	2.61	4.07	1.82	3.64	3.92	2.23	5.13	4.40
4.56	0.29	2.66	4.61	1.50	3.86	4.21	2.29	5.13	4.42
5.11	2.52	2.80	4.68	2.67	3.96	4.81	2.95	5.34	5.21
4.79	4.11	6.85	4.52	6.70	6.87	3.98	8.37	3.96	3.89
5.11	3.69	7.28	4.82	8.07	6.29	5.20	8.46	5.34	4.65
3.86	2.91	4.48	4.05	4.56	2.94	2.88	4.59	4.74	4.99
3.86	1.57	4.11	4.07	4.51	2.98	3.49	4.17	4.74	4.11
4.26	1.79	5.29	3.81	5.10	3.15	3.67	4.28	5.18	4.60
4.57	1.23	2.08	4.78	1.27	2.69	4.08	1.03	5.88	5.28
5.11	4.29	3.77	5.42	2.78	4.57	5.56	3.13	5.34	4.70

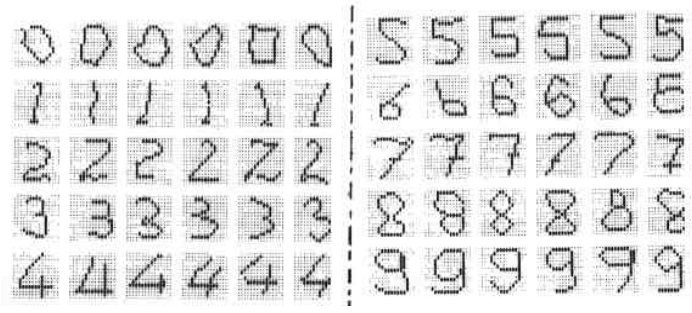


Figure (5) Samples of the hand-written numerals used in testing the model.

Table (4): Number of cells having values >0.5 for the simple planes of the feature extraction stage for numerals from 0 t0 9.

S-Plane number	numerals									
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	8	1	12	5	11	15	4	13	4	2
2	10	13	6	10	6	11	15	1	8	11
3	4	1	8	3	9	3	2	7	6	4
4	4	0	1	3	0	2	2	0	6	5
5	1	1	1	2	2	3	2	2	2	2
6	5	0	8	4	9	6	4	8	7	5
7	5	3	10	4	10	4	3	8	8	6
8	1	2	2	2	3	1	2	2	3	1
9	1	1	1	3	2	1	2	1	3	2
10	9	14	5	9	5	7	10	3	6	12
11	5	0	1	4	1	3	5	1	7	6
12	5	0	2	4	1	3	4	0	7	6
13	4	0	0	2	1	2	3	2	5	5
14	0	3	1	1	2	0	0	3	1	1
15	4	0	7	3	9	5	3	8	5	4
16	0	1	0	1	0	1	0	1	0	1
17	0	0	0	0	1	0	0	0	0	0
18	8	1	11	5	9	6	6	6	10	8
19	6	0	2	6	0	3	5	0	10	7
20	4	1	2	3	1	2	3	1	5	4

Various circuit configurations are studied using different number of cell planes for the classification stage for different training factors. In the following we report the resulted calculation for the clearness rate, rejection rate and recognition rate as a function of the number of iterations for various learning factors and cell-planes, see figures (6 , 7 & 8). They are defined as number of patterns clearly recognized / total number of patterns tested, number of patterns rejected / total number of patterns tested and number of patterns correctly recognized / total number of patterns tested, respectively. For all of the three above measurement criteria, it is found that fastest convergence occurs when using η=0.2, U_{LOH} =7 and U_{HOH}=9.

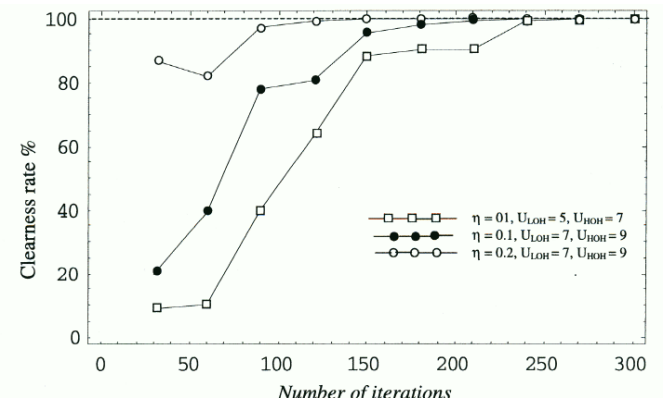


Figure (6): The Clearness rate percentage for the circuit versus the number of iterations, using biased neurons back-propagation for the classification and output stages.

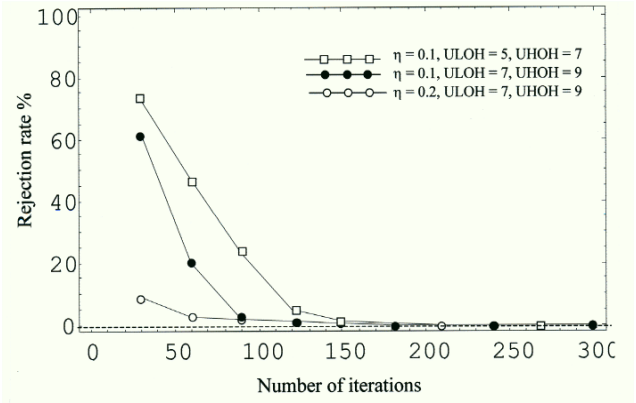
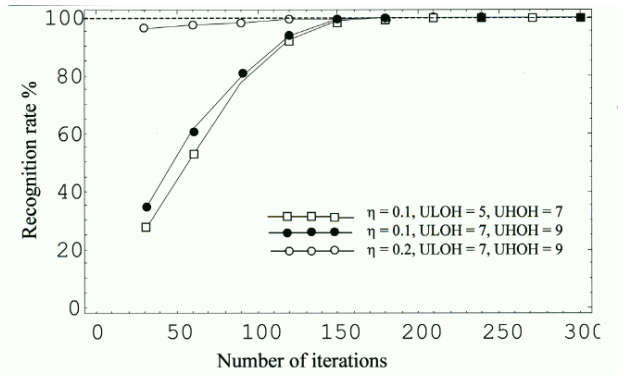


Figure (7): The Rejection rate percentage for the circuit versus the number of iterations, using biased neurons back-propagation for the classification and output stages.



F

Figure (8): The Recognition rate percentage for the circuit versus the number of iterations, using biased neurons back-propagation for the classification and output stages.

5. Conclusions:

Twenty different training patterns are used in the training of a neural circuit that uses both Fukushima and back-propagation principles for character recognition. The availability of these patterns or features in characters and numerals varies considerably. Although, all the features extracting planes are affected for each input character, it is found that a reasonable trade off can be decided to achieve a balance between the required accuracy and speed of circuit training. This can be manifested by the fast convergence speed achieved, which is shown in calculating clearness, rejection and recognition rates.

Large size of input layer and bigger number training patterns used in the learning process would increase the model ability for accurate response to characters having complex features and distorted or shifted, such as hand written, without much increase in circuit complexity.

It should be noted that, biased neurons back-propagation circuits are used for the classification stage and the output stage rather than Fukushima's neucognitron only [10].

6. References:

1. Fukushima, K. Nagahara, K. and Shouno, H., "Training Neocognitron to Recognize Handwritten Digits in Real World", IEEE Computer Society Press, and Aizu International Symposium on Parallel Algorithm / Architecture Synthesis, March, 1997.
2. Gouhara, K.Imai, K. and Uchikawa, Y., "Position and size Representations by Neural Networks", Control and Computers, Vol. 17, No. 2, 1989.
3. Cho, S, "Neural Network Classifiers for Recognizing Totally Unconstrained Hand Written Numerals", IEEE Trans. on Neural Networks, Vol. 8, No. 1, Jan. 1997.
4. Caudill, M., "Neural Networks Primer; Part I", Artificial Intelegence Expert, Vol. 2, No. 12, Dec. 1987.
5. Fukushima, K. Miyake, S. and Takayukiito, "Neocognitron: A Neural Network Model for Mechanism of Visual Pattern Recognition", IEEE Trans. on Systems, Man and Cybernetics", Vol. SMC-13, No. 5, Oct. 1983.
6. Fukushima, K. Miyake, S., "Neocognitron: A new Algorithm for Pattern Recognition Tolarant of Deformation and Shifts in Position", Pattern Recognition, Vol. 15, No. 6, 1982.
7. El-Sayegh, S., "A System for Handwriting Character Recognition Based on the Neocognitron neural Network", M.Sc. Thesis in Computer Science, University of Technology, Baghdad, Iraq, 1996.
8. Fukumi, M. and Omatu, S., "A New Bach-Propagation with Coupled Neuron", IEEE Trans. on Neural Networks, Vol. 2, No. 5, 1991.
9. Mahmmod, M. A. B., "An Equivalent Model as Medium Scale Neocognitron-Like Brain Model", M.Sc. Thesis in Electrical Engineering, University of Basrah, Iraq, 1999.
10. Ali,H. A. and Mahmmod, M. A. B., "Character Recognition Medium Scale Neocognitron- Like, Neural Network Combined Model", Submitted to the Transaction on Asian Language Information Processing (TALIP), Dec. 2000.
11. Haykin, S., "Neural Networks: A Comprehensive Foundation", Prentice Hall,1999.
12. Tank, D. and Hopfield, J., "Collective Computation in Neural-like Circuits", Scientific American, Dec. 1987.
13. Sousek, B. and Soucek, M., "Neural and Massively Parallel Computers: the Sixth Generation", John Wiley & sons Inc. 1988.
14. Bishop, C. M., "Neural Networks for Pattern Recognition", Oxford University Press, 2nd Ed., 1999.
15. Fukushima, K and Wake, N, "Handwritten Alphanumeric Character Recognition by the Neocognitron", IEEE Trans. on Neural Networks, Vol. 2, No. 3, May 1991.
16. Ali, H. A. and Mahmmod, M. A. B., " A Hybrid Neural Network Model for Character Recognition of Hand Written Numerals", Sub. For Publication in the AMSE Journal of the Association for Modeling and Simulation Techniques in Enterprises, March, 2001.
17. Wolfram, S., "The Mathematica Book", 4th Edition, Wolfram Media, Cambridge, 1999.