

Massively Parallel Implementation of Neural Network Architectures

O. M. Omidvar, University of the District of Columbia
Washington, DC 20008

C. L. Wilson, National Institute of Standards and Technology
Gaithersburg, MD 20899

Abstract

In recent years neural networks have been used to solve some of the difficult real time character recognition problems. These SIMD implementations of the networks have achieved some success, but the real potential of neural networks are yet to be utilized. Several well known neural network architectures have been, modified, and implemented. These architecture are then applied to character recognition. The performance of these parallel character recognition systems are compared and contrasted. Feature localization and noise reduction are achieved using least squares optimized Gabor filtering. The filtered images are then presented to an FAUST based learning algorithm which produces the self-organizing sets of neural network generated features used for character recognition. Implementation of these algorithms on highly parallel computer with 1024 processors allows high speed character recognition to be achieved at a speed of 2.3ms/image, with greater than 99% accuracy on machine print and 89% accuracy on unconstrained hand printed characters. These results are achieved using identical parallel processor programs demonstrating that the method is truly font independent. The backpropagation is included to allow comparison with more conventional neural network character recognition methods. The network has one hidden layer with multiple concurrent feedback from the output layer to the hidden and from hidden layer to the input layer. This concurrent feedback and weight adjustment is only possible on a SIMD computer.

1 Introduction

Recent advances in neural network research resulted in various implementations of well known learning algorithms for varieties of applications. The learning process is used for the adjustment of connection weights toward a stable state. The stable state insures convergence, which will create locally optimal results with minimal error. Most of these implementations have been done on serial machines which have taken up to a week of execution time for training to solve a small size problem. Neural networks operate in parallel and distributed fashion, and the weight adjustments need not be done one at the time. This makes massively parallel implementation of these algorithms a logical step in the development of neural network technology. Biological neurons are too slow to process things in serial, therefore the parallel implementation is a natural choice if we are to simulate biological neural networks.

Neural network parallel algorithms show great promise for providing highly accurate, and noise resistant image recognition. One specific area of image recognition, the conversion of images of hand written and

Machine print characters to computer representation, has been studied in detail in the past. Both special purpose hardware [1,2] and software [3,4] approaches have been used on the character recognition problem with promising results.

1.1 Statement of The Problem

The problem of computer recognition of document content from images is usually broken down into three operations. First the relevant areas containing text are located. This is usually referred to as field isolation. The global image containing images of one or more characters is broken into images of individual characters. This process is usually referred to as segmentation. These isolated characters form the input for the character recognition operation. Only this final part of the problem is discussed in this paper. It is important to recognize that the success of all of the later steps in this process is dependent on the success of all of the previous steps. Character recognition is substantially more difficult if the images of characters can not be successfully segmented, or if extraneous graphic information such as part of the box surrounding a field is captured during field isolation.

In the character recognition operation, images of individual characters from either hand or machine print are recognized as specific characters or are rejected. An isolated character image is presented to the recognition device and a specific character identity or set of identities with associated confidence factors is returned. For a character recognition technology to be of commercial interest, both accuracy and the ability to detect low confidence recognitions must be present. If human intervention is required to detect failures or if the failure rate is too high existing data entry methods will be more cost effective.

Neural networks, as the most promising new method of character recognition, have several advantages. They are designed, from the beginning, to be parallel and can easily be used on massively parallel computers. They provide a much simpler method of recognition than strictly local image processing methods or rule based methods. This simpler, learning based, structure makes for simpler less brittle programs, easier software development, and greater speed. Finally, recognition rates are being speeded up by the introduction of parallel computers. For both image processing and neural networks, parallel computers can result in significant increases in speed.

An image of a standard page contains about eight million pixels. The rotation of this size image take approximately 20 minutes on a Sun 3/470 workstation or on a Sun/4¹. The same rotation can be carried out on the massively parallel computer, between 4.5 and 0.9 seconds, depending on the angle of rotation [5]. This basic image processing operation has been speeded up by between 266 and 1300 times. Similar speed increases are obtained on other image processing and neural network character recognition.

1.2 Parallel Hardware Architecture

The Single Instruction Multiple Data (SIMD) architecture used for this study was a Active Memory Technology 510 Distributed Array Processor; DAP510. This machine consists of a 32 by 32 grid of 1 bit processor elements (PE). Operation of the PE array is controlled by a 4 MIPS RISC master control unit (MCU). All program instructions are stored in a separate program memory and are passed to the PE array through the MCU. A block diagram of this architecture is shown in figure 1.

All data is stored in a separate array memory. The array memory is organized in 32 by 32 bit planes with each of the bits in each plane connected to one PE. Data can also be passed between PEs along the

¹Certain commercial equipment may be identified in order to adequately specify or describe the subject matter of this work. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the equipment identified is necessarily the best available for the purpose.

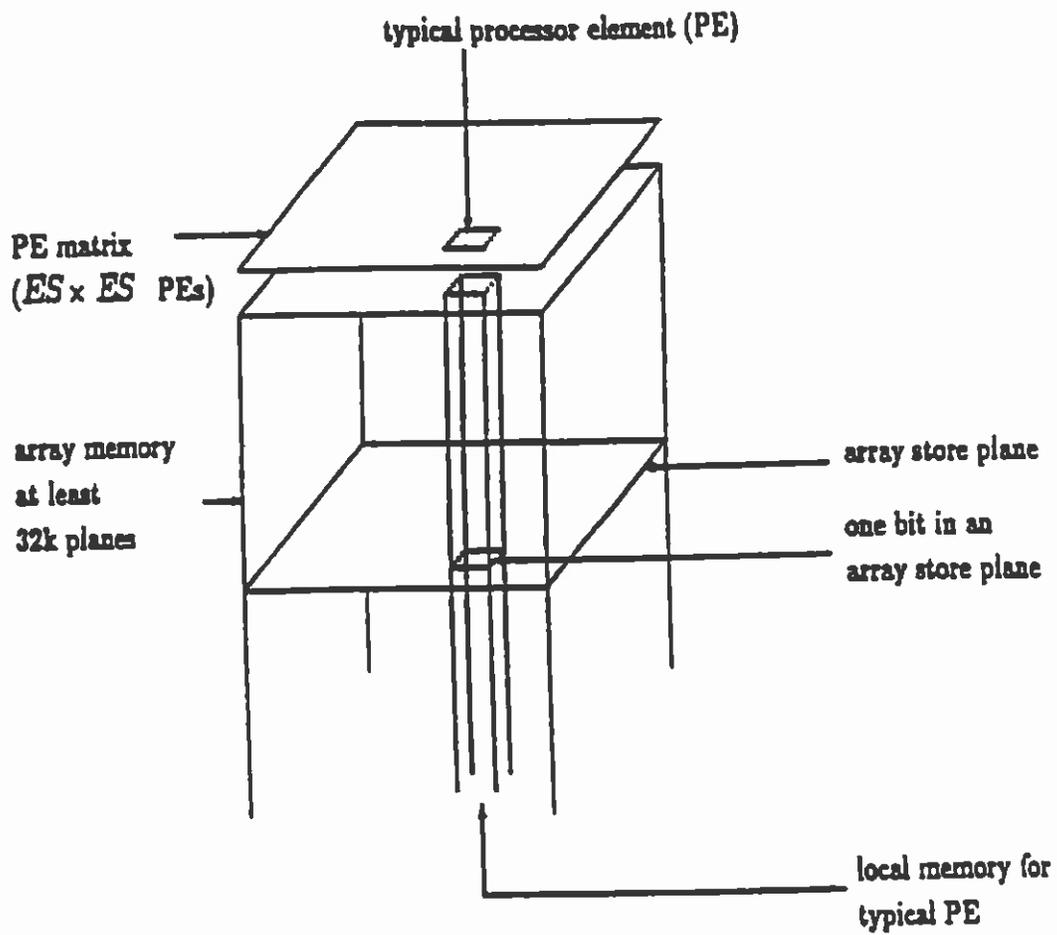


Figure 1: Array processor architecture for massively parallel computer.

grid. The cycle time of all PEs is 100 ns. This processor configuration is capable of performing ten billion binary operations per second; processing speed increases proportionally with the length of data items used. Two data mappings are particularly well suited to the DAP structure: a vector mode in which successive bits of a single word are mapped into a row of the array, and a matrix mode in which successive bits of a word are mapped into layers of the array memory vertically. Both of these modes of operation are used in the backpropagation implementation presented in this paper. The optimized least squares filtering makes extensive use of the vector mode for the calculation of optional filter coefficients and of the matrix mode for the generation of the Gabor function filter kernels.

The backpropagation network model makes extensive use of binary matrix mode operations. The most important considerations for character filtering applications are that the Gabor functions provide the minimum combination of uncertainty in position and spatial frequency resolution, that the profiles of Gabor functions match the visual receptor field profiles of mammalian eyes[6], and that the resulting filtering process is well suited to the parallel data mapping of the array processor. For character recognition applications, only sixteen 8-bit Gabor filter coefficients are required to approximate a 1024 pixel image file.

2 Neural Character Recognition

The usual method for designing character recognition systems has been top down. A method of feature extraction is selected which provides features and the resulting classification problem is solved by a neural network. In this work, we have take a different approach. The general form of input receptor fields which are used in tasks such as stereo vision have been modeled using parallel Gabor functions [6]. The output of these receptor fields is coupled to small networks for detection of position. The model was not specifically designed for character recognition and could be used for any set of signals which could be represented by the available set of Gabor functions. Gabor functions are well suited to this application because they allow reasonable quality image reconstruction with small numbers of basis functions.

Image reconstruction using other methods of equivalent quality requires approximately 100 basis functions. Any decrease in the basis function size reduces the number of connections needed in the classification network by the product of the basis function difference times the number of hidden nodes in the classification network. The system uses parallel processing for both the feature extraction process and the character recognition process. The image is transmitted to a set of Gabor receptor field(GRF) modules and the combined optimal response is determined by least squares. This method of feature extraction is clearly not biological but is well understood in terms of conventional numerical processing. The outputs of the GRF's are the inputs of a back propagation network. The activation of the output nodes of the back propagation network are used to classify the input images.

3 Gabor Receptor Fields

The methods used in this paper have been applied to neural network image processing and pattern recognition applications but have not been applied together previously for character recognition applications. Gabor function image representation is a set of incomplete nonlinear functions, which are used for input image feature extraction. These functions reduce random image noise and smooth irregularities in image structure by acting as spatially localized low-pass filters.

The Gabor filter has been used for image compression and image texture analysis. The motivation for use of incomplete functions, which add considerable complexity to the representation problem are discussed in detail [6]. Gabor functions provide the minimum combination of uncertainty in position and spatial frequency resolution, in feature extraction . The profiles of Gabor functions also match the visual

receptor field profiles of primate eyes. Therefore the resulting feature extraction process is well suited to the parallel data mapping of an array processor.

The Gabor filtering section is accomplished using a least squares fit of each image. The kernel functions used are Gabor functions. The least squares fitting of the filter coefficients is necessitated by the non-orthogonal nature of the Gabor functions. Adopting the convention that bold upper case variables represent array processor matrix data types and bold lower case variables represent array processor vector data types, the Gabor functions are defined as:

$$\mathbf{G}_j(\mathbf{X}, \mathbf{Y}) = \exp(-\mathbf{R}^2) \begin{cases} \sin(\omega_j \mathbf{X}') \\ \cos(\omega_j \mathbf{X}') \end{cases} \quad (1)$$

where the matrix variables \mathbf{R} , \mathbf{X}' , and \mathbf{Y}' are given by:

$$\mathbf{R}^2 = (\mathbf{X}'^2 + \mathbf{Y}'^2) / \sigma_j^2 \quad (2)$$

$$\begin{pmatrix} \mathbf{X}' \\ \mathbf{Y}' \end{pmatrix} = T \begin{pmatrix} \mathbf{X} - x_{0j} \\ \mathbf{Y} - y_{0j} \end{pmatrix}. \quad (3)$$

The \mathbf{X} and \mathbf{Y} matrices in the array processor are row and column expanded in the form:

$$\mathbf{X} = \begin{pmatrix} x_1 & x_2 & \dots & x_{32} \\ x_1 & x_2 & \dots & x_{32} \\ & & \dots & \\ x_1 & x_2 & \dots & x_{32} \end{pmatrix} \quad (4)$$

$$\mathbf{Y} = \begin{pmatrix} y_1 & y_1 & \dots & y_1 \\ y_2 & y_2 & \dots & y_2 \\ & & \dots & \\ y_{32} & y_{32} & \dots & y_{32} \end{pmatrix} \quad (5)$$

A typical scalar transformation to be applied to each element of the matrix variables is a rotation of the form:

$$T = \begin{pmatrix} \cos \theta_j & \sin \theta_j \\ -\sin \theta_j & \cos \theta_j \end{pmatrix}. \quad (6)$$

The matrix function \mathbf{G}_j is then expressed as a function of the scalar variables: ω_j , which is the spatial frequency of the function; σ_j , the spatial extent of the function; (x_{0j}, y_{0j}) , the origin of the function; and θ_j , the orientation of the function.

3.1 Tiling

Since the Gabor basis functions are an infinite set, it is necessary to select a specific subset of them to be used as the filter elements which cover the character image. This selection process is referred to as tiling the image. For the class of filter discussed here each set of image origins has twice the sample density of the previous level and the number of directions selected, n_θ , is fixed. This results in a filter with directional sensitivity and positional sensitivity determined by the choice of the level parameter, i . The character images used in this study are 32×32 so that using large values of i would result in massive over-sampling of the image. The Gabor filter for the lowest value of i , on the other hand, is approximately a directional bar detector and adds little to the filter's spatial resolution. At each level the frequency and spatial resolutions, ω_i and σ_i , are adjusted to allow small overlaps in extent and provide octave spatial frequency response.

After extensive experimentation, it was found that a reasonably good approximation to the image could be obtained by using only level 2 Gabor functions. Reasonable directional selectivity was obtained with four fold symmetry, $n_\theta = 4$. When the even and odd frequency components are included this results in a Gabor function set with 32 functions. All filter operations described in this paper are carried out with these 32 function filters.

The results of the experiments are easily explained. For $i = 1$, the Gabor functions form a directional filter and provide only field-centered spatial location. As i increases the resolution of the filter increases. At level 3 the spatial and frequency resolution exceed the stroke size (line width) of the character and provide limited improvement in resolution. All experiments also suggest that, given the complex structure of equations (1)-(6), sampling an image containing less than 16 pixels for each d_i interval is not an efficient use of Gabor functions.

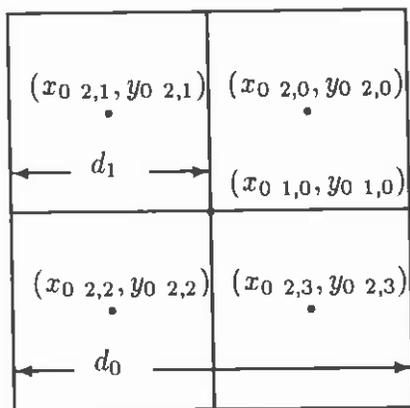


Figure 2: Location of the first two levels of tiling points for the Gabor functions. The full set of locations is given in table 1; in general $d_{i+1} = d_i/2$.

| i | x_{0i} | y_{0i} | ω_i | σ_i | θ_i |
|-----|--|--|---------------------------|------------------------|---|
| 1 | $\frac{d_0}{2}$ | $\frac{d_0}{2}$ | $\frac{\pi}{\sqrt{2}d_0}$ | $\frac{d_0}{\sqrt{2}}$ | $\frac{2\pi}{n_\theta}, \frac{4\pi}{n_\theta}, \dots, 2\pi$ |
| 2 | $\frac{d_1}{2}, \frac{3d_1}{2}$ | $\frac{d_1}{2}, \frac{3d_1}{2}$ | $\frac{\pi}{\sqrt{2}d_1}$ | $\frac{d_1}{\sqrt{2}}$ | $\frac{2\pi}{n_\theta}, \frac{4\pi}{n_\theta}, \dots, 2\pi$ |
| ... | | | | | |
| n | $\frac{d_n}{2}, \frac{3d_n}{2}, \dots, \frac{(n+1)d_n}{2}$ | $\frac{d_n}{2}, \frac{3d_n}{2}, \dots, \frac{(n+1)d_n}{2}$ | $\frac{\pi}{\sqrt{2}d_n}$ | $\frac{d_n}{\sqrt{2}}$ | $\frac{2\pi}{n_\theta}, \frac{4\pi}{n_\theta}, \dots, 2\pi$ |

Table 1: Table of the possible Gabor functions used to tile the image. Each level, i , contains $2i^2 n_\theta$ possible Gabor functions. The values of d_i are obtained by dividing the image as shown in figure 2; $d_{i+1} = d_i/2$.

3.2 Image Reconstruction

Once the Gabor functions are selected, the filtering operation starts by converting the binary image to an 8-bit image with a step height between levels of -127 and 127 with $\sum q = 0$. Since the set of Gabor functions is non-orthogonal, the filtering must be performed by least squares optimization. On the small images discussed here, direct methods are far more efficient for this operation than the neural net method proposed for data compression [4]. Given n different G_j 's the filtering operation is based on obtaining a

least squares fit to the image \mathbf{q} by forming the matrix \mathbf{A} , each component of which is the inner product of the form:

$$a_{ij} = \mathbf{G}_i \cdot \mathbf{G}_j \quad (7)$$

and the vector,

$$b_i = \mathbf{q} \cdot \mathbf{G}_i \quad (8)$$

and solving

$$\mathbf{b} = \mathbf{A} \mathbf{c} \quad (9)$$

for the filter coefficients, \mathbf{c} . Since the matrix \mathbf{A} is the same for any given set of n Gabor functions, the matrix is factored once, and only generation of \mathbf{b} and back substitution of the factored \mathbf{A} matrix is required to obtain each \mathbf{c} . The image is converted to its filtered form:

$$\mathbf{q}' = \sum_{j=1}^n c_j \mathbf{G}_j \quad (10)$$

and then thresholded at zero making the image binary again.

For character recognition applications, only sixteen 8-bit Gabor reconstruction coefficients are required to approximate a 1024 pixel image. The most important of these justifications is the biological one. The known properties of the primate vision system can provide vital clues about the recognition process used to distinguish characters[7].

4 Neural Network Architecture

Neural network based methods for image filtering and feature extraction are combined to develop a font independent character recognition on a massively parallel array processor. Feature localization and extraction are also achieved using optimized Gabor filtering. Resultant feature sets are then classified using a backpropagation network. Implementation of these algorithms on a highly parallel computer with 1024 processors allows high speed character recognition to be achieved at a speed of 87 μs /image, with greater accuracy on machine print and reasonable accuracy on unconstrained hand printed characters. These results are achieved using identical neural network algorithms implemented on a massively parallel processor demonstrating that the method is truly font independent.

The backpropagation learning algorithm is modified to adapt a massively parallel environment. The network has 1024 neurons in the first layer. The hidden layer has 256 neurons and the output layer can recognize up to 64 different classes. The network operates in a concurrent parallel manner. In the present implementation the number of output terminal is variable. All the incoming inputs are fed to the input layer at the same time, this simulates primate vision in collecting information from the environment. The connection weights are created at random with normal distributions. Two sets of random numbers are generated and used which makes convergence easier. one set for the connections between the input and hidden layer and another set for connections between the hidden layer and output layer.

The error is calculated not one at a time but simultaneously for all the classes. Then the error matrix is used for calculation of the total error in a concurrent manner. This is a major departure from the serial implementation of the backpropagation algorithms where errors are calculated and reduced one at a time for each output class being recognized in the network. Weight adjustment is done in a novel manner.

Several strategies are used to accomplish the concurrent weight adjustment. One is to adjust the connection weights between the hidden and output layer one at a time and then calculate the network outputs after each adjustment [8]. If the error is changing in the right direction (the minimum difference)

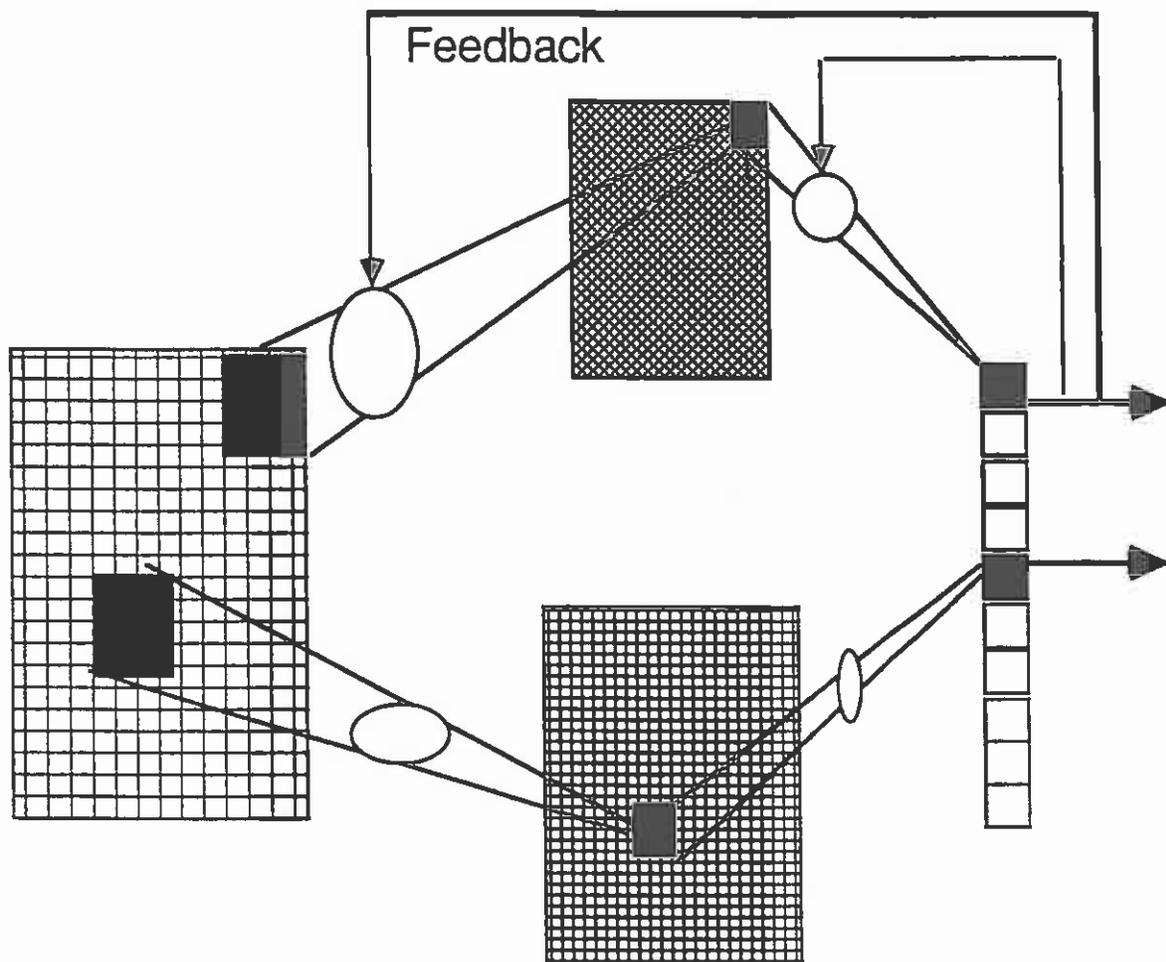


Figure 3: Neural Network Architecture.

then we continue the weight adjustment. Another method is to adjust all the connection weights between the input layer and the hidden layer simultaneously one set at a time [9]. Then the error is checked. If the error is changing in the desired direction the process continues until the time when the decrease in the size of error falls below a set tolerance level. The overview of the of the network architecture is shown in figure 3. The network connections are omitted and the layers shown in the figure are not the actual size.

The present implementation uses the third strategy which is to adjust all the connected weights in the network at one time. This strategy will generate a higher level of parallelism in the network operation, while it has the higher potential to suffer from the traditional local minima problem. Another difficulty is the convergence to a stable state. The network could be trapped in a constant weight change cycle with no change in error direction even with continuous weight adjustment. There are several methods to prevent the above problems. One is to shake the network out the local minima by using techniques such as conjugate gradient, the Boltzmann machine and simulated annealing.

A unique feature of this implementation is simulating the excitatory and inhibitory functions of real neural systems. The program has the capability to check all the connections between two layers simultaneously, then activate the inactive connections in either excitatory and inhibitory directions. This feature is not feasible in serial backpropagation implementations. This makes the implemented network converge faster and create stability regardless of the network applications. We can also increase the number of hidden layers to simulate higher functions.

The Gabor function approach provides a mathematically concise method for performing feature extraction on image data. Parallel back propagation networks provide a very effective method for performing supervised nonlinear classification [8,9]. In any pattern classification problem, the efficiency of the network is strongly effected by the quality of the input features used to train the network. In back propagation networks, the weights in the fully connected network increase as the product of the number of input nodes and hidden nodes plus the product of hidden nodes and output nodes. As the number of weights increases the cost of calculation increases and the numerical stability of the network decreases [10].

In this implementation of the backpropagation network, effective character recognition can be performed with as few as sixteen hidden nodes. This requires only 256 connections which greatly reduces cost and increases network stability.

5 Training and Performance

We have chosen to use a parallel implementation of the Gabor function as an input filter for feature extraction. Since the calculation time for the Gabor filter is 14 ms, this provides an order of magnitude time saving. Noise and small translation and rotation errors in the input images cause the number of required connections in the network to become large and redundant. The Gabor filter discussed above was chosen to provide an adaptive mechanism which would remove noise, normalize the image, and provide the directional and positional sensitivity [5]. The comparison of training time between parallel and serial implementations with variable number of patterns is shown in Figure 4.

After the training process, the parallel backpropagation has improved the speed of the recognition process thirty five times. This enable us to implement more elaborate variations and new and improved version of the backpropagation algorithm on parallel machine and achieve greater accuracy and higher speed. The performance based on classification time for parallel implementation is shown in Figure 5.

The effectiveness of the several biologically motivated character recognition methods are compared in table 2. All of the calculations were performed on an Active Memory Technology DAP 510. The data for the first three methods is taken from [11,12,13]. All of the timing data is for recognition of low quality machine printed digits with 100% recognition. The learning times for the ART-1 and FAUST methods

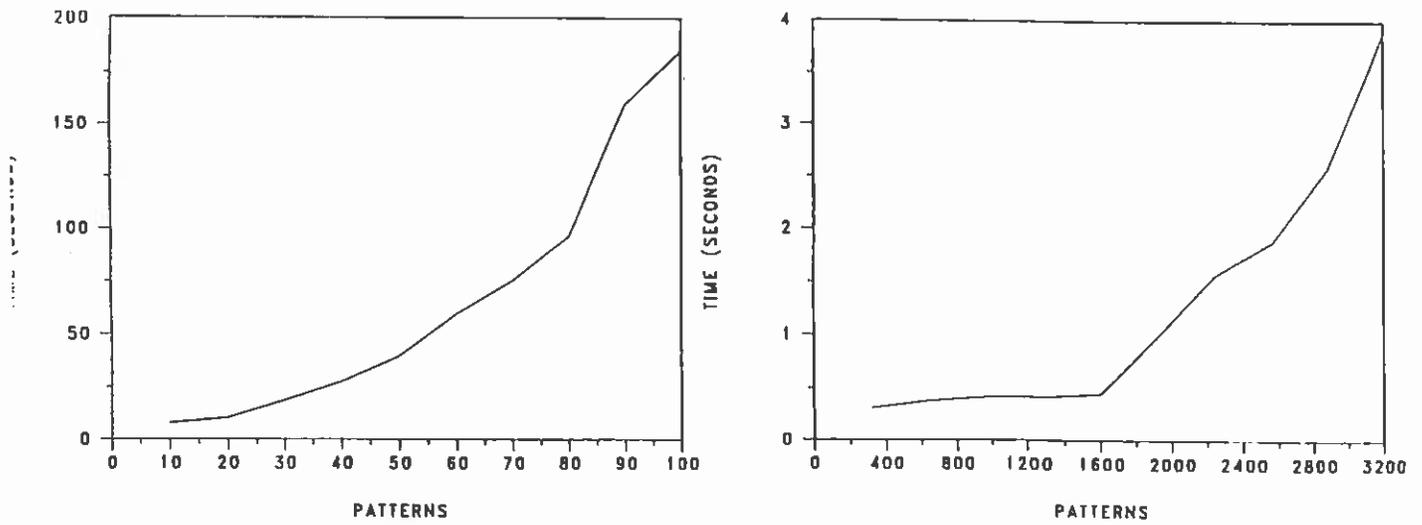


Figure 4: Training Time Comparison.

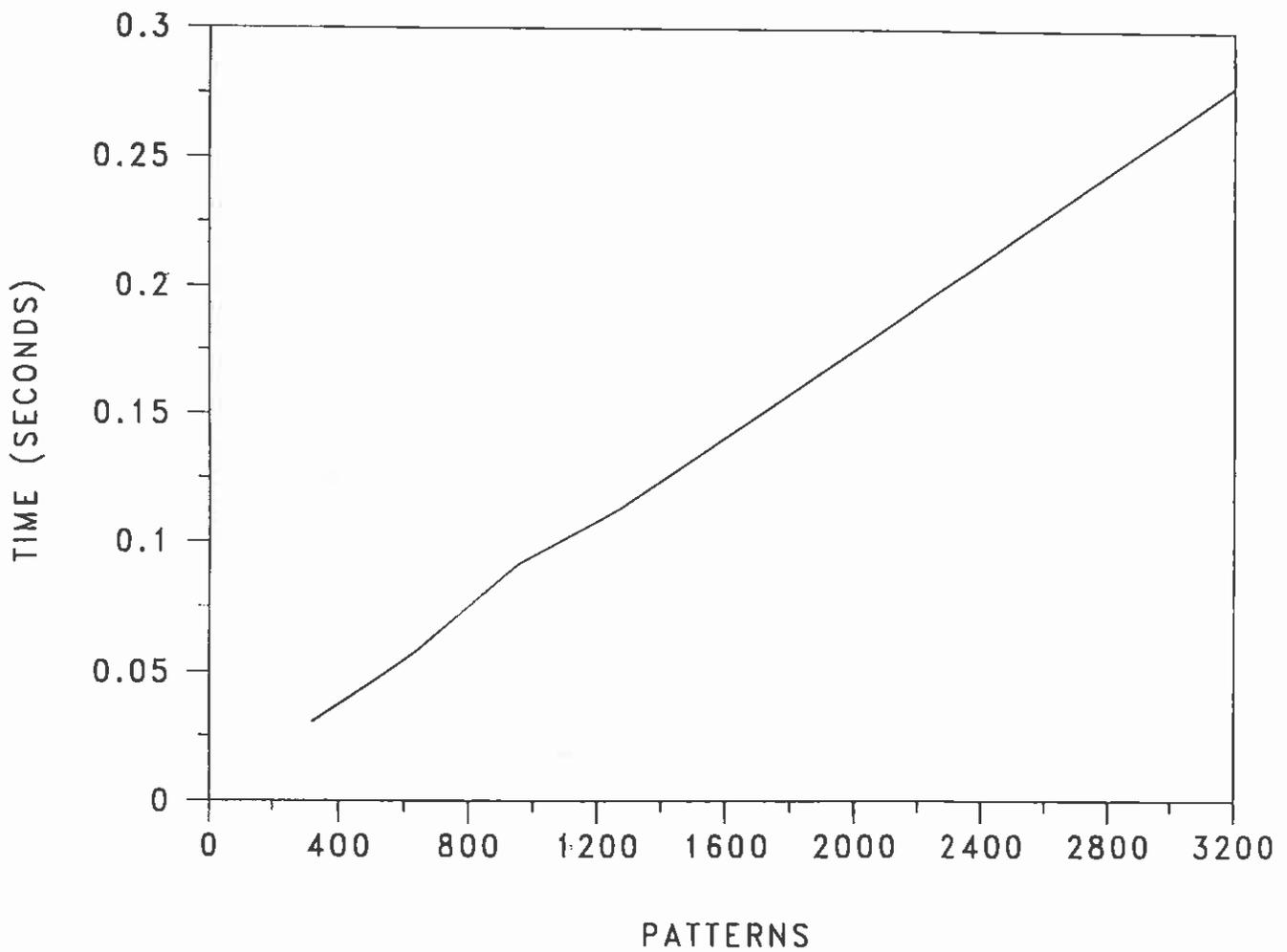


Figure 5: Parallel Classification Performance.

are comparable to the recognition times. The learning times for the ART-2 and CORT-X methods are approximately 10 times the recognition times. The first four methods are unconditionally stable and always converge. The advantage of the FAUST method over ART-2 based method is obtained by using Gabor filtering methods instead of a nonlinear filter. The advantage over ART-1 based methods is that the filter can be eliminated for simple recognition tasks, such as machine print, because edge and stroke variations are learned by the associative relevance memory. This advantage is also apparent in comparing both serial and parallel implementations of Backpropagation. The parallel method is much faster than the serial method for classification but both methods are dominated by the Gabor filtering time.

| System | Filter | Assoc. | Trigger | Learn | Class. | Speed |
|----------|------------|--------|---------|----------|----------|----------|
| ART-1 | external | corr. | max P | replace | Bayesian | 7.4ms |
| ART-2 | nonlinear | corr. | max P | average | Bayesian | ≈ 1400ms |
| CORT-X | ART-2 & RF | corr. | max P | average | Bayesian | ≈ 1400ms |
| FAUST | external | 5 | Assoc. | 6 types | Assoc. | 2.5ms |
| | 4 types | types | Space | of RF | Space | |
| Backprop | Gabor | | Error | Serial | Bayesian | 16ms |
| Backprop | Gabor | | Error | Parallel | Bayesian | 14ms |

Table 2: Comparison of the performance of various recognition methods on low quality machine printed digits.

6 Summary and Results

A massively parallel neural network architecture for size and local shape invariant digit recognition has been developed. The network is based on known biological data on the structure of vertebrate vision but is implemented using more conventional numerical methods for image feature extraction and pattern classification.

The input receptor field structure of the network uses Gabor function feature selection. The parallel neural network implementation presented here provides a highly accurate, noise resistant image recognition. The present work also addresses the problem of using a specific class of computer architecture, an array of 1024 processors arranged in a 32 by 32 grid and operated in a parallel mode, as a neural network character recognition device.

The classification is done using the parallel backpropagation network. Using the Gabor generated features as neuron input, an implementation of back propagation on serial machine achieved 100% accuracy when trained and tested on a single font size and style while classifying at a rate of 87 μ s per character. Taking the same trained network, recognition greater than 99.9% accuracy was achieved when tested with digits of different font sizes.

At the present time the technology of character recognition is advancing rapidly for several reasons. First, the introduction of document imaging systems has made the incremental cost of introducing character recognition much lower by providing a large supply of scanned images. Second, new methods of character recognition, such as neural networks, are increasing the accuracy of the recognition process. Finally, many of the new recognition methods, and image processing methods in general, are being made more practical by parallel processing computers which have significantly reduced computing costs and increased recognition speed.

Imaging of documents makes character recognition cost depend on the recognition process without incurring the additional cost required to collect and store image data. The volume of images being collected is very large. One credit company now images four trillion bytes of images each year and stores the data on optical media. The total key data entry segment of the United States economy is estimated at 20 billion dollars a year. The existence of large volumes of accessible images and a high potential economic return make the climate for the introduction of recognition technology very favorable.

References

- [1] M. Verleysen, G. Paul, "VLSI Chips for Neural Networks" in O.M. Omidvar(ED), Progress in Neural Networks, Vol I, Ablex Pub Co, Norwood, NJ, 1990.
- [2] L. D. Jackel, H.P. Graf, W. Hubbard, J.S. Denker, D. Henderson and Isabelle Guyon, "An Application of Neural Net Chips: Handwritten Digit Recognition", IEEE/ IJCNN Proc, San Diego, Vol. II, pp. 107-115, 1988.
- [3] A. Rajavelu, M. T. Musavi, and M. V. Shirvaikar, "A Neural Approach to Character Recognition", Neural Networks, 2, pp. 387- 393,1989.
- [4] O. M. Omidvar, "Formation of Visual Perception Via Synaptic Reinforcement", Proc of INNS Conf, Paris, France, 1990.
- [5] C. L. Wilson, R. A. Wilkinson, and M.D. Garris, "Self- Organizing Neural Network Character Recognition on a Massively Parallel Computer" Proc. of the IJCNN, II, PP. 325-329, June 1990.
- [6] J.G. Daugman, "Complete Discrete 2-D Gabor Transform by Neural Networks for Image Analysis and Compression". IEEE Trans. on Acoustics, Speech, and Signal Processing, 36, pp. 1169-1179, 1988.
- [7] J.E. Dayhoff, "Neural Network Architecture", Van Nostrand Rienhold", New York, NY, 1990
- [8] A. H. Kramer and A. S. Vincentelli, "Efficient Parallel Algorithms for Neural Networks", in D. S. Touretzkey(Ed), Advances in Neural Information Processing Systems I, Morgan Kaufman, pp. 40- 48, 1989.
- [9] A. Kremer and A. Sangcovanni, "Efficient Parallel Algorithms for Neural Networks" Tech Report: Department of EE-CS, UC Berkeley, CA, 1989.
- [10] J. Yadegar and R. Thanakig, "The DAP Neural Simulator", Tech Report: Active Memory Technology, Irvine, CA, 1989.
- [11] G. A. Carpenter, S. Grossberg and C. Mehanian, "Invariant Recognition of Cluttered Scenes by a Self-Organizing ART Architecture: CORT-X Boundary Segmentation," Neural Networks, 2, pp. 169-181, 1989.
- [12] G. A. Carpenter and S. Grossberg, "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine," Computer Vision, Graphics, and Image Processing, 37, pp 54-115, 1987.
- [13] C. L. Wilson, "A New Self-Organizing Neural Network Architecture for Parallel Multi-Map Pattern Recognition - FAUST," submitted to Progress in Neural Networks.