Pattern Recognition Using Pulse-Coupled Neural Networks and Discrete Fourier Transforms

Raul C. Mureşan
Personal Research Center, P-ta Cipariu, Nr. 9, Ap. 17, Cluj-Napoca, Romania
raulmuresan@personal.ro, http://www.raulmuresan.home.ro

Abstract

A novel method for pattern recognition using Discrete Fourier Transforms on the global pulse signal of a pulse-coupled neural network (PCNN) is presented in this paper. We describe the mathematical model of the PCNN and an original way of analyzing the pulse of the network in order to achieve scale- and translation-independent recognition for isolated objects. We also analyze the error as a result of rotation. The system is used for recognizing simple geometric shapes and letters.

Keywords: Pattern Recognition; Pulse-coupled neural networks; Discrete Fourier Transform; Multilayer perceptron.

1. Introduction

Pulse coupled neural networks (PCNN) were introduced as a simple model for the cortical neurons in the visual area of the cat's brain. Important research in the 80's and 90's led to the establishment of a general model for PCNN [3]. Such models proved to be highly applicable in the field of image processing, a series of optimal procedures being developed for contour detection and especially image segmentation [10].

At the other end, image processing is faced with harder problems such as the pattern recognition. Many recognition systems are based on saliency techniques or on feature extraction followed by matching. However, such models are either subject to problems determined by geometric transforms (scaling, translation or rotation) or to high computational complexity. Moreover, it is known today that parallel processing could solve computational complexity but in order to take advantage of it we need parallelisable models. Neural models fit this requirement.

We describe in the next sections a system that evaluates the global pulse of a PCNN in order to find correlation in the pulse signal and achieve pattern recognition. We prove that the pulse signal captures the morphological information from the stimulus image.

2. The architecture of the model

The model proposed here is based on three modules of processing: the pulse-coupled neural network, the Discrete Fourier Transform (DFT) module and the multilayer perceptron (MLP) classifier (Fig. 1). Information flow is mainly feed-forward but there are also lateral interactions between the pulse-coupled neurons.

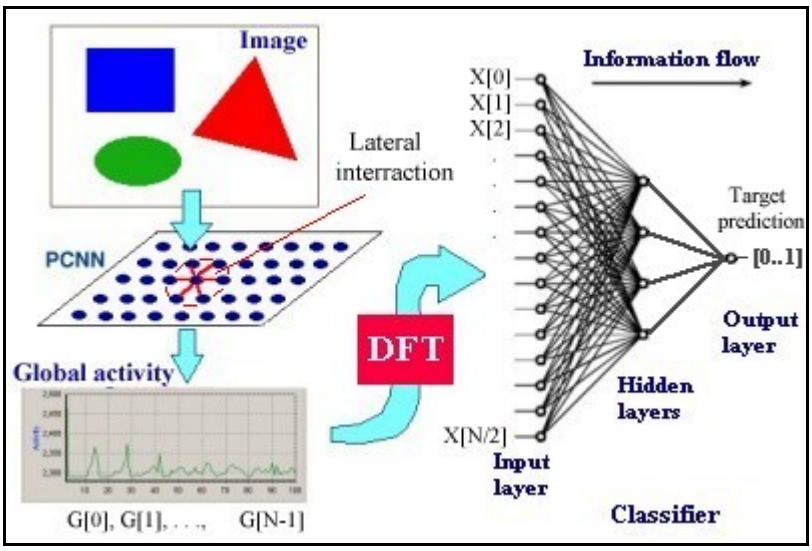

Fig. 1. The architecture of the recognition system

2.1. The pulse-coupled neural network

The key of the entire system lies in the neural analyzer that, in our case, is made of pulse-coupled neurons, which act like local analyzer cells (Fig. 1. - PCNN). The pulse train generated by the neurons is a direct result of stimulus excitation and lateral interaction between neurons. Lateral interaction and further stimulation determine the neurons to fire in synchrony in the homogenous areas associated to the image. These effects can be exploited in image segmentation. However, our assumption is that the pulse train of the neurons captures somehow morphological information from the image.

The model we used for the network had been proposed by T. Lindblad and J.M. Kinser [10]. The pulse-coupled neuron is a particular type of leaky integrator neuron [2, 8]. The leak is modeled by the exponential terms in equations (1) and (2). The refractory period is simulated by increasing the threshold when the neuron fires and decreasing it exponentially after firing.

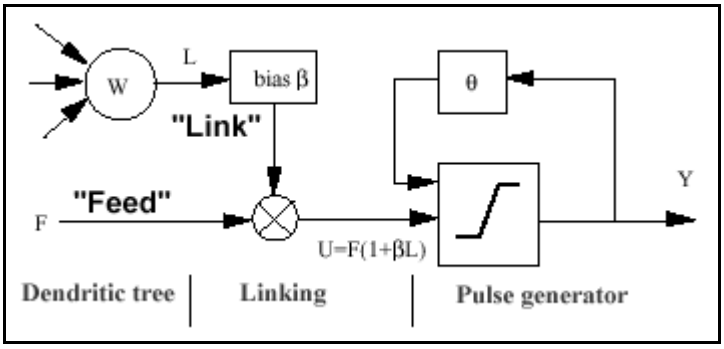

Fig. 2. The structural model of the pulse-coupled neuron

In the next equations we will refer to “n” as being the current iteration (discrete time step) where "n" varies from 1 to N-1 (N - is the total number of iterations; n = 0 is the initial state). The dendritic tree can be described by the following equations:

$$Fij[n] = e^{-\alpha_F} \cdot F_{ij}[n-1] + V_F \cdot \sum_{kl} M_{kl} S_{ijkl} \quad (1)$$

$$Lij[n] = e^{-\alpha_L} \cdot L_{ij}[n-1] + V_L \cdot \sum_{kl} W_{kl} Y_{ijkl}[n-1] \quad (2)$$

The two main components F and L are called feeding and linking. The (i,j) pair stands for the position of the neuron in the map. α_F and α_L are time constants for feed and link. S_{ijkl} is the stimulus component computed from the pixel intensity (<i+k, j+l>, "<x,y>" meaning the intensity of the pixel with coordinates x and y) in the input image. Usually this value is normalized. V_F and V_L are normalizing constants and M and W represent the constant synaptic weights. M and W are computed by using the inverse square rule [10]: $f(k,l) = 2 / \sqrt{k^2 + l^2}$. Y stands for the output of the neuron and can only take a binary value of 0 or 1.

The linking effect can be modeled as follows:

$$U_{ij}[n] = F_{ij}[n] \cdot (1 - \beta \cdot L_{ij}[n]) \quad (3)$$

$U_{ij}[n]$ represents the internal activation of the neuron and β is the linking weight parameter (bias in Fig. 2).

The pulse generator determines the firing events in the model. In fact, the pulse generator is also responsible for the modeling of the refractory period. As the neuron produces a spike, its threshold is raised to prevent it from firing again in the near future (established by the parameter settings). The threshold is then decreased to allow the neuron to fire when its activation is increased.

$$Y_{ij}[n] = \begin{cases} 1, if\ U_{ij}[n] > \Theta_{ij}[n-1] \\ 0, otherwise \end{cases} \quad (4)$$

$$\Theta_{ij}[n] = e^{-\alpha_\Theta} \cdot \Theta_{ij}[n-1] + V_\Theta \cdot Y_{ij}[n] \quad (5)$$

In equations (4) and (5) $\Theta_{ij}[n]$ represents the dynamic threshold of the neuron while α_Θ and V_Θ are the time constant and the normalization constant respectively.

During the simulation, each iteration updates the internal activity and the output for every neuron in the network, based on the stimulus signal from the image and the previous state of the network. For each iteration the total number of firings (Equation 6) over the entire PCNN is computed and stored in a global array G (see Fig. 1).

$$G[n] = \sum_{ij} Y_{ij}[n], \quad \text{where n is the iteration } (n = 0...N-1) \quad (6)$$

The global array is then used at the next levels of the system (to compute the DFT of the global pulse signal).

2.2. The Discrete Fourier Transform

We used the standard analysis equations to calculate the DFT [16]:

$$\mathrm{Re}\, X[k] = \sum_{i=0}^{N-1} G[i]\cos(2\pi ki / N), \quad k = 0...N/2 \qquad (7)$$

$$\mathrm{Im}\, X[k] = -\sum_{i=0}^{N-1} G[i]\sin(2\pi ki / N), \quad k = 0...N/2 \qquad (8)$$

Computing the DFT means basically correlating the input signal with each basis function. The DFT yields two shorter signals to be analyzed. We used only the imaginary part of the DFT in further processing but a combination may be possible as well. Our choice had been motivated by experimental observations that show a relative stability of the real part over all the shapes used for testing. We also enhanced speed by using only the imaginary part in the higher levels.

2.3. The classifier

Our classifier is basically a multilayer perceptron (MLP). The neural architecture consists of one input layer, one hidden layer and one output neuron. The input layer contains a number of inputs equal to the samples in the imaginary part of the DFT signal (Im X in eq. (8)). Then, a hidden layer has an extension of about 10 to 20% of the input layer.

Because of the specific tasks used to test the system, the output layer contained only one neuron (target detection). An output value of 1 is equivalent to target detection whereas a value of 0 means no target detection. A standard backpropagation algorithm is used for supervised training [15].

3. Results and discussion

In order to test the system 100 samples of the pulse signal are computed (100 iterations). As a consequence, the imaginary part of the DFT has only 51 samples to be classified. The system is used only for target detection. The first test deals with scale and position independence and we use simple geometric shapes as input images (Fig. 3). Only one shape is presented at a time. We tested the capability of the system to detect the target shape it was trained on. For scale-independence testing we used the circle as target.

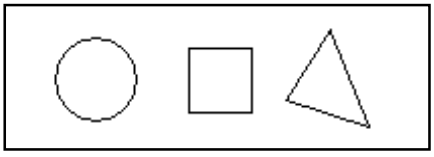

Fig. 3. Shapes used for testing.

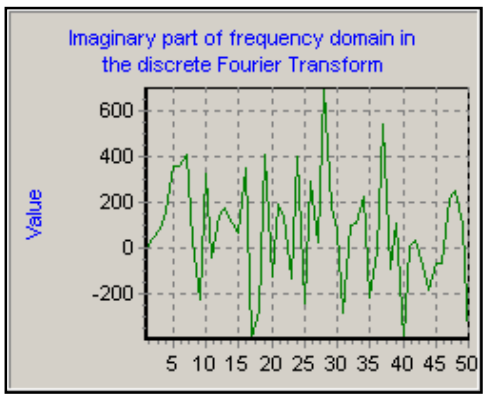

Fig. 4. The DFT transform of the pulse signal

Stimulus images used were grayscale bitmaps sized 100 x 100 pixels.

As expected, the system showed total translation independence. Then we tried to determine the resistance to the scaling of the stimulus. The shape used for testing was a circle and 7 different scales were used. The supervised training procedure was composed of two phases: a learning phase, when only one target (circle with a diameter of 40 pixels) had been presented and a rejection phase which consisted of training for non-target prediction over all the other shapes (rectangle, triangle). Prediction rate never dropped below 75% (which is good enough considering information loss when downscaling the stimulus to 1 / 4 the original size). Up scaling produces a much better rate (90%).

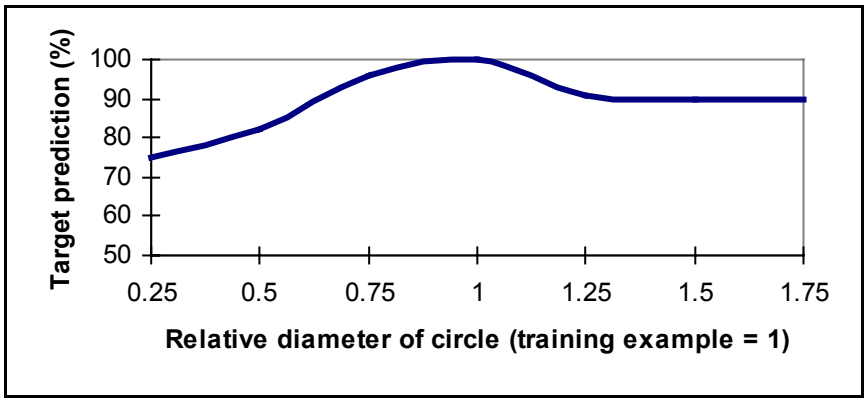

Fig. 5. Target prediction for different scales of stimulus (1 - training example)

As for the rotation error, the stimulus (square shape) had been rotated at different angles and the normalized medium error computed. Results shown in Fig. 6 prove that the maximum error is obtained for the two principal diagonals (45^0 and 135^0). This is a clue indicating that rotation error is caused by pixel discretization (the diagonal of the pixel is $l \cdot \sqrt{2}$ so the diagonal distance between pixels is larger then the distance along the axes).

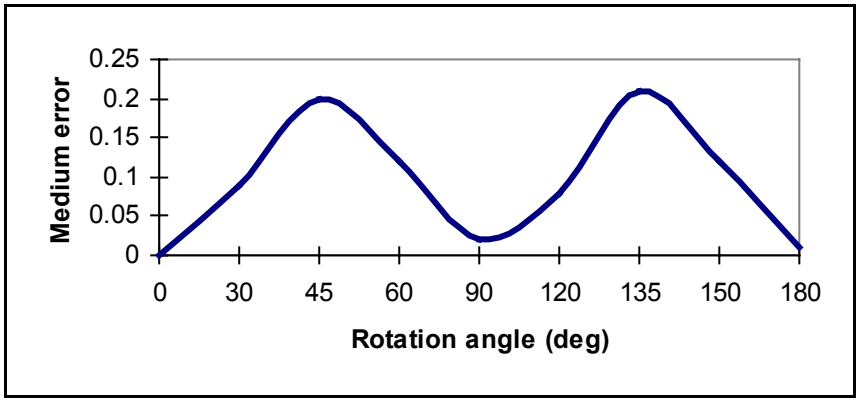

Fig. 6. The normalized medium error for different rotation angles

We have to mention that the main purpose of this study was to show the existing correlation between the pulse signal of the PCNN and the shapes presented as stimulus images. We did not test recognition performance with multiple or overlaid shapes but such a mechanism could be constructed. The system should split the image into smaller windows, perform a local analysis

finding the most probable location of the target and then enlarge or shrink the attention window to fit the target.

References

[1] E.L. Brown, B.M. Wilamowski, Spatial to Temporal Conversion of Images Using A Pulse-Coupled Neural Network, International Joint Conference on Neural Networks - IJCNN'99, Washington, DC, July 10-16, 1999, #521, Session 7.1.

[2] R. Eckhorn, H.J. Reitboeck, M.Arndt, P.Dicke, Feature Linking via Stimulus-Evoked Oscillations: Experimental results from Cat Visual Cortex and Functional Implications from a Network Model., Proc. Int JCNN'89, Washington D.C., Vol. I, 1989, pp. 723-730.

[3]. R. Eckhorn, H.J. Reitboeck, M. Arndt, Feature Linking via Synchronization among Distributed Assemblies: Simulations of Results from Cat Visual Cortex., Neural Comp. Vol. 2, 1990, pp. 293-307.

[4] A.K. Gray, C.M. Konig, P. Engel, W. Singer, Stimulus-Dependant Neural Oscillations in the Cat Visual Cortex Exhibit Inter-Columnar Synchronization which Reflects Global Stimulus Properties., Nature, Vol. 338, 1989, pp. 334-337.

[5] J.J. Hopfield, Neural Networks and Physical Systems with Emergent Collective Computational Abilities., Proceedings of the National Academy of Sciences USA, Volume 79, April, 1982, pp. 2554-2558.

[6] J.L. Johnson, Waves in Pulse Coupled Neural Networks, Proc. of the World Congress on Neural Networks, Vol. 4, INNS Press, 1993.

[7] J.L. Johonson, Pulse-Coupled Neural Nets: translation, rotation, scale, distortion, and intensity signal invariance for images, Appl. Opt., Vol.33, No.26, Sep. 1994, pp.6239-6253.

[8] P.E. Keller, D. McKinnon, Pulse-Coupled Neural Networks for Medical Image Analysis, Proceedings of the SPIE, v. 3722 (47), 1999, pp. 444-451.

[9] J.M. Kinser, A Simplified Pulse-Coupled Neural Network, Proceedings Applications and Science of Artificial Neural Networks II, Orlando, FL, Vol. 2760, April 1996, pp. 563-567.

[10] T. Lindblad, J.M. Kinser, Image Processing using Pulse-Coupled Neural Networks, Springer, London, 1998.

[11] J.W. McClurkin, J. A. Zarbock and L. M. Optican, Temporal Codes for Colors, Patterns and Memories., Cerebral Cortex, Vol. 10, A. Peters and K. S. Rockland [eds], Plenum Press, NY, 1994, pp. 443.

[12] Y. Ota, B.M. Wilamowski, Analog Implementation of Pulse-Coupled Neural Networks, IEEE Transactions on Neural Networks, vol. 10, no. 3, May 1999, pp. 539-544.

[13] H.J. Ranganath, G. Kuntimad, J.L.Johnson, Pulse Coupled Neural Networks for Image Processing, PCNN International Workshop, Huntsville, Alabama, April 5, 1995.

[14] S.K. Rogers and M. Kabrisky. An Introduction to Biological and Artificial Neural Networks for Pattern Recognition. Vol. TT4., SPIE Optical Engineering Press., Bellingham, WA, 1991.

[15] S.J. Russell, P. Norvig, Artificial Intelligence. A Modern Approach., Prentice-Hall, Upper Saddle River, New Jersey, 1995.

[16] S.W. Smith, The Scientist and Engineer's Guide to Digital Signal Processing, California Technical Publishing, San Diego, CA, 1997.

[17] T. Watanabe, M. Tanaka, T. Kurita**,** Autonomous foveating system based on the Pulse-Coupled Neural Network, Proceedings of the ITC-CSCC'99, vol.1, 1999, pp.197-200.

[18] M.E. Zaghloul, J.L. Meador, R.W. Newcomb, Silicon implementation of pulse coded neural networks, Kluwer, Boston, 1994.

[19] J.M. Zurada, Introduction to Artificial Neural Systems., West Publishing Company, St. Paul, Minnesota, 1992.