International Journal of Soft Computing 2 (1): 172-175, 2007

Hebbian Neural Network Based Algorithm for Video Processing

[1]Ismail A. Ismail, [2]Nabawia A. El-Ramly, [2]Mohammed Amin and [2]Hani M. Ibrahim
[1]Faculty of Computers and Information, Zagazig University, Egypt
[2]Department of Computer Science, Faculty of Science, Menoufiya University, Egypt

Abstract: Traditional video processing methodology consists in transforming an original video in a processed video, using a well-known transformation function. This function is coded into the application. Our goal is to make a system able to perform image processing tasks without knowing the transformation function as traditional methods. We have used Artificial Neural Networks-ANN (Hebbian learning rule) in order to achieve this goal that is easy and flexible. The main advantage of this approach is that we haven't used predefined algorithmic functions, but video visual features learning. However, in the proposed method each frame is processed independently from the others. This studies goal is to show the viability and usability of Neural Network training, in creating "adaptive intelligent filters", using Hebbian learning rule. The comparison between the proposed technique and the temporally average filter is included.

Key words: Gaussian noise, hebbian learning rule, neural networks, video processing

INTRODUCTION

One aspect of video processing that makes it such an interesting topic of study is the amazing diversity of applications that use video processing or analysis techniques. Virtually every branch of science has sub disciplines that use recording device or sensors to collect image and video data from the universe around us[1-6].

In artificial neural networks (ANNs), learning is achieved mostly (but not exclusively) through the changes in the strengths of connections between neurons. The mechanisms of learning: changes in neural parameters (threshold, time constant), creation of new synapses, elimination of synapses and changes in the synaptic weights or connection strengths are included. The type of learning is determined by the manner in which the parameter changes take place. However, several learning algorithms are used for ANNs training. These algorithms can be classified as: supervised and unsupervised learning[7,8]. We may think of the supervised (teacher) as having knowledge being represented by a set of input-output examples. Suppose now that the teacher and neural network are both exposed to training vector. The network parameters are adjusted under the combined influence of the training vector and the error signal. This adjustment is carried out iteratively in a step-by-step fashion with the aim of eventually making the neural network memic the teacher. Some examples of supervised learning rules are perceptron, delta, Widrow-Hoff, correlation and outstar learning rules[8,9]. In unsupervised learning there is no external teacher or critic to over see the learning process. Examples for unsupervised learning are Hebbian and winner-take-all learning rules[7,9].

On the other hand, (ANNs) are very general function approximators which can be trained based on a set of examples. It would seem as useful tools for image and video processing[10,11]. Instead of designing an algorithm, one could construct an example data set and an error criterion; hence train ANNs to perform the desired input-output mapping. The network input can consist of pixels or measurements in images. On the other hand, the output can contain pixels, decision, labels, etc, as long as these can coded numerically (no assumptions are made). This means that adaptive methods can perform several steps in the video processing chain at once.

In the field of image and video restoration, the most recent publications have suggested Hopfield and back propagation networks applied to different areas of image and video processing and computer vision such as stereo matching, edge detection and image segmentation[12-16]. In this study, the Hebbian artificial neural network (HANN) is developed to filter noisy video degraded with Gaussian noise.

Hebbian learning rule: The Hebbian learning rule is based on the assumption that if two neighboring neurons one is activated and the other is deactivated at the same time, the weight connecting these neurons should increase[15,6]. For neurons operating in the opposite phase, the weight between them should decrease. If there is no correlation, the weight should remain unchanged. This assumption $\Delta w_{ji} = c\, x_j\, o_i$ can be described by the formula

Corresponding Author: Hani M. Ibrahim, Department of Computer Science, Faculty of Science, Menoufiya University, Egypt

Where w_{ji} is the weight from j-th to i-th neuron, c is the learning constant, x_j is the signal on the j-th input and o_i is the output signal. The training process starts usually with values of all weights set to zero. This learning rule can be used for both soft and hard threshold neurons. Since the desired responses of neurons are not used in the learning procedure, this is the unsupervised learning rule, i.e., the resulting output will not affect the learning. The absolute values of the weights are proportional to the learning time, which is an undesired effec.

MATERIALS AND METHODS

The HANN model as shown in Fig. 1 is constructed as full connected ANNs as input layer with four nodes (because of less blurring), two hidden layers, (the first one with three nodes but the second one with two nodes) and output layer, (with one node). Such that W represents the synaptic weight between the input layer and the first hidden layer. The synaptic weight between the two hidden layers is represented by V. Finally, U represents the synaptic weight between the second hidden layer and the output layer The following technique displays HANN method that is used to remove Gaussian noise.

Video processing implementation on HANN: Video processing implementation on neural networks based on Hebbian learning rule is described by the following steps:

- Divide the video into separate frames.
- Initialize the synaptic weights of network, w_{ji}, v_{ji} and u_{ji} to small random values.
- Assign a small positive value to the learning-rate parameter η.
- Assign e_i=4 and e_j=3 .(where the e_i is the number of nodes in the input layer and e_j is the number of nodes in the first hidden layer).

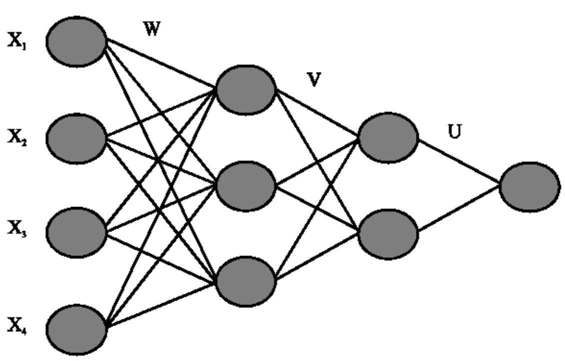

Fig. 1: The model of HANN

- For n=1 ; n is the iteration number.
- For each four neighboring pixels in the same row with only one pixel shift in each step do the following steps until all the frame is finished.
- Assign the four pixels to input data vector X , $X=[x_1,x_2,x_3,x_4]$.
- For j=1,...,e_j ; Compute

$$y_j = \sum_{i=1}^{e_i} w_{ji} x_i$$

- For j=1,…,e_j and For i=1,…,e_i ;Compute

$$\Delta w_{ji} = \eta(y_j\ x_i - y_j \sum_{k=1}^{jj} w_{ki} + y_k)$$

$$w_{ji} = w_{ji} + \Delta w_{ji}$$

- For j=1,…,e_j-1 ;Compute

$$r_j = \sum_{i=1}^{e_i -1} v_{ji} y_i$$

- For j=1,…,e_j-1 and For i=1,…,e_i-1 ; Compute

$$\Delta v_{ji} = \eta(r_j y_i - r_j \sum_{k=1}^{j} v_{ki} + r_k)$$

$$v_{ji} = v_{ji} + \Delta v_{ji}$$

- For j=1,…,e_j-2 ;Compute

$$s_j = \sum_{i=1}^{e_i -2} u_{ji} r_i$$

- For j=1,…,e_j-2 and For i=1,…,e_i-2 ; Compute

$$\Delta u_{ji} = \eta(s_j r_i - s_j \sum_{k=1}^{j} u_{ki} + s_k)$$

$$u_{ji} = u_{ji} + \Delta u_{ji}$$

- Increase n by 1 and continue until the synaptic weights, w_{ji}, v_{ji} and u_{ji} reach their steady-state values.
- Repeate the above steps for the others frames.

Noticing: The value of the processed pixel is equal to s_j

RESULTS AND DISCUSSION

In this result of the proposed method for video degraded by Gaussian is presented in Fig. 2.

In order to illustrate the performance of the proposed method, it is compared with the result of temporally average filter by computing the correlation between processed frames and original frames as shown in Table 1.

Table 1: Comparison between the proposed method and the temporally average filter

Temporally average filter	Proposed method
0.9891	0.9980

CONCLUSION

In this study, we developed and trained HANN with one input layer, two hidden layers and one output layer. Then we used this method to processed video degraded by Gaussian noise. The results indicate that the technique gives good performance which smoothing the background noise while preserving the edges and the fine details with less blurring to improve video appearance for human viewer.

REFERENCES

1. Al Bovik, 2000. Image and video processing, Academic press, Canada.
2. Rafael, C.G. and R.E. Woods, 2002. Digital image processing, Prentice-Hall Inc., USA.
3. Thomas, B. *et al.*, 2001. Parallel Image Processing, Springer-Verlag Berlin Heidelberg, Germany.
4. John, C.R., 2002. The image processing handbook, CRC Press LLC, Acid-Free Paper, Florida.
5. Cirstian, M. and A. Rosa, 2004. Gray-scale image enhancement as an automatic process driven by evalution, IEEE Transactions on systems, Man and Cybernetics, P art B: Cybernetics, 43: 1292-1298.
6. Kim, J.U. *et al.*, 2001. Advanced contrast enhancement using partially overlapped sub-block histogram equalization, IEEE Trans. Circuits and Systems for Vide Tech., 11: 475-484.
7. Simon, H., 1999. Neural networks a comprehensive foundation, Prentice-Hall Inc, USA.
8. Jacek, M.Z., 1995. Introduction to artificial neural systems, PWS Publishing Company, USA.
9. Bogdan, M.W., 2003. Neural network architec-tures and learning, ICIT 2003 Maribor, Slovenia pp: TU1-TU12.
10. Dali, W. and A. Zilouchion, 2001. Application of neural network in design of digital filters, CRC Press LLC.
11. Joae, R. and F.S. Osorio, 2000. Adaptive Filters for Image Processing Based on Artificial Neural Networks, Proceedings of XIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'00) IEEE, Brazil.
12. Abbiss, J., B. Branes and M. Fiddy, 1991. Superresolution algorithms for a modified Hopfield neural network, IEEE Trans. Signal Processing, 39: 1516-1523.
13. Paik, J. and A. Katsaggelos, 1992. Image restoration using a modified Hopfield network, IEEE Trans. Image Processing, 1: 49-63.
14. Mario, A.T.F. and J.M.N. Leito, 1994. Sequential and parallel image restoration: Neural network implementations, IEEE Trans. Image Processing, 3: 789-801.
15. Sun, Y., 2000. Hopfield neural network based algorithms for image restoration and reconstruction-Part I: Algorithms and simulations, IEEE Trans. Signal Processing, 48: 2105-2118.