Proceedings of the 7th WSEAS International Conference on Neural Networks, Cavtat, Croatia, June 12-14, 2006 (pp1-6)

Image denoising and Fuzziness Measures

VINCENZO NIOLA
Department of Mechanical Engineering for Energetics
University of Naples "Federico II"
Via Claudio, 21 - 80125 Napoli
ITALY

GIUSEPPE QUAREMBA
University of Naples "Federico II"
Via S. Pansini, 5 – 80131 Napoli
ITALY

Abstract: - A self-organizing multilayer neural network suitable for image processing applications is proposed. The output of the neurons in the output layer has been viewed as a fuzzy set and measures of fuzziness have been used to model the error (instability of the network) of the system. Various mathematical models for calculation of fuzziness of this fuzzy set have been described. The weight updating rules under each model have been developed. This error is then back-propagated to correct weights so that the system error is reduced in the next stage. A comparative study (both analytical and experimental) on the rate of learning for different error measures is also done. Results also show that the rate of learning affects the output, especially when the noise level is very high.

Key-Words: - Neural network, fuzzy sets, computer vision, image processing.

1 Introduction

Many scientific datasets are contaminated with noise, either because of the data acquisition process or because of naturally occurring phenomena. A first pre-processing step in analyzing such datasets is denoising, that is, estimating the unknown signal of interest from the available noisy data. There are several different approaches to denoise images.[1][2]

Fuzzy set-based techniques are an important ingredient in the development of information technologies.

In the field of information processing fuzzy sets are important in clustering, data analysis and data fusion, pattern recognition and computer vision. Fuzzy rule-based modeling has been combined with other techniques such as neural nets and evolutionary computing and applied to systems and control engineering, with applications to robotics, complex process control and supervision. In the field of information systems, fuzzy sets play a role in the development of intelligent and flexible man-Bmachine interfaces and the storage of imprecise linguistic information. In Artificial Intelligence various forms of knowledge representation and automated reasoning frameworks benefit from fuzzy set-based techniques, for instance in interpolative reasoning, non-monotonic reasoning, diagnosis, logic programming, constraint-directed reasoning, etc.

The scope of the paper is to analyze the learning rate with computing time and robustness of results with reference to the selection of error measures and activation function in order to improve the response of a neural network.

2 Fuzzy Sets: Measures of Information

First of all consider that the perceptron can be thought like a net composed of elementary processors organized in such a way to recreate the biological neural connections. Observe that, in our study, the nodes of two consecutive levels are connected by one link (or weight) but no connection exists among nodes belonging to the same level (Fig. 1).

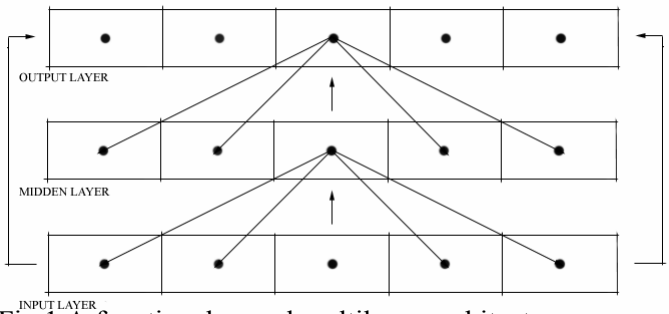

Fig.1 A functional neural multilayer architecture

Usually, the level where the nodes of input are present is named input layer, while the level which shows the output is said output layer. The layers which lies between the input and the output layers are named hidden layers.[3][4]
The output of nodes of one layer is transmitted to the correspondent nodes of the following layers by means of links (weights) which can amplify, attenuate or inhibit such output through weighted factors. With the exception of nodes of the input layer, the total input for each node is the sum of the weighted output of nodes belonging to previous layer. Each node is activated in agreement with the input received from both the other nodes and the activation function. The total input of the *i*-th node of one layer is

$$I_i = \sum_j w_{ij} o_j \quad (1)$$

where o_j is the output of *j*-th neuron of the previous layer and w_{ji} is the weighted link between the *i*-th node of one layer and *j*-th node of the previous layer.
The output of the *i*-th node is

$$o_j = f\left(I_j\right) \quad (2)$$

where $f(\cdot)$ is the activation function.
In our application the activation function was set as follows

$$f(\cdot) = \frac{1}{\sqrt{2\pi}} e^{\frac{\bar{X}-x}{5\pi\sigma}} \quad (3)$$

where $\bar{X}$ and σ were the mean and the standard deviation respectively of the values of the nodes selected by the neighbourhood system.
For a specific target $\{t_i\}$ the error was estimated as

$$E = \frac{1}{2}\sum_i (t_i - o_i)^2 \quad (4).$$

The problem is to adjust the set of weights in the connecting links such that the desired output $\{t_i\}$ is obtained at the output nodes.
In particular, in the following we describe two different fuzziness measures used for the weight updating rules employed during the back-propagation of the network.

2.1 Weight correction for Linear Index of Fuzziness

Let us consider

$$E = g\left(\nu_l\right) = \nu_l \quad (5)$$

where the linear index of fuzziness is defined as

$$\nu_l = \frac{2}{n}\sum_j \left\{\min\left(o_j, \overline{1-o_j}\right)\right\}, \quad (6)$$

n being the number of neurons in the output layer.
Thus if we consider that (see Appendix)

$$\Delta w_{ji} = \begin{cases} \eta\left(-\frac{\partial E}{\partial o_j}\right) f'\left(I_j\right) o_i \\ \eta\left(\sum_j \delta_k w_{kj}\right) f'\left(I_j\right) o_i \end{cases} \quad (7)$$

for the output layer and other layers respectively, where

$$f'\left(I_j\right) = \frac{\partial o_j}{\partial I_j} = o_j\left(1-o_j\right) \quad (8)$$

and

$$\delta_j = -\frac{\partial E}{\partial I_j} = -\frac{\partial E}{\partial o_j}\frac{\partial o_j}{\partial I_j} = -\frac{\partial E}{\partial o_j} f'\left(I_j\right) \quad (9)$$

we obtain

$$\Delta w_{ji} = \begin{cases} \eta_l\left(-\frac{2}{n}\right) f'\left(I_j\right) o_i \quad if\ 0 \le o_j \le 0.5 \\ \eta_l\left(\frac{2}{n}\right) f'\left(I_j\right) o_i \quad if\ 0.5 \le o_j \le 1. \end{cases} \quad (11)$$

Here, in fact,

$$-\frac{\partial E}{\partial o_j} = \begin{cases} -\frac{2}{n} \quad if\ 0 \le o_j \le 0.5 \\ \frac{2}{n} \quad if\ 0.5 \le o_j \le 1. \end{cases} \quad (12)$$

Note that η is a proportional constant: a large value corresponds to rapid learning but might result in oscillations.

2.2 Weight correction for Exponential Entropy

If we consider

$$E = g(H) = H \quad (13)$$

where H is the exponential entropy of a fuzzy set with

$$H(S) = \frac{1}{n\left(\sqrt{e}-1\right)} \sum_{j=1}^{n} \left\{ o_j e^{1-o_j} + \left(1-o_j\right) e^{o_j} - 1 \right\} \quad (14)$$

S being a fuzzy set, then

$$\frac{\partial H}{\partial o_j} = \frac{1}{n\left(\sqrt{e}-1\right)} \left\{ \left(1-o_j\right) e^{1-o_j} - o_j e^{o_j} \right\} \quad (15).$$

Applying an argument similar to that of Linear Index for Fuzziness, for Exponential Entropy also we get

$$\Delta w_{ji} = -\eta' \frac{1}{\frac{\partial E}{\partial o_j}} f'\left(I_j\right) o_j$$

$$= -\eta \frac{1}{\left(1-o_j\right) e^{1-o_j} - o_j e^{o_j}} f'\left(I_j\right) o_j \quad (16)$$

with $\eta = \eta' \times n\left(\sqrt{e}-1\right)$.

In other words

$$\Delta w_{ji} = \begin{cases} -\eta \dfrac{1}{\left(1-o_j\right) e^{1-o_j} - o_j e^{o_j}} f'\left(I_j\right) o_i & \text{if } 0 \le o_j \le 0.5 \\ \eta \dfrac{1}{o_j e^{o_j} - \left(1-o_j\right) e^{1-o_j}} f'\left(I_j\right) o_i & \text{if } 0.5 \le o_j \le 1.0 \end{cases} \quad (17)$$

Considering the expression for the Linear Index of Fuzziness and the Exponential Entropy, one can see that in each of the two cases, the expression for Δw_{ji} has a common factor (*i.e.*, $\eta f'\left(I_j\right) o_i$).

3 The Neural Architecture

In more details, the neural network was developed by constructing 3 layers. In the Fig. 2 is depicted the scheme of the neural multilayer architecture.

Each layer has MxN neurons (the image is MxN), each neuron corresponding to one pixel. Between the input and output layer there exists one layer indeed. The neurons belonging to the same level do not have any link between them. Each neuron of one layer is connected to the correspondent neuron of the previous layer and to its nearest neurons. Moreover, each neuron of the output layer, is connected to the correspondent neuron of the input layer.

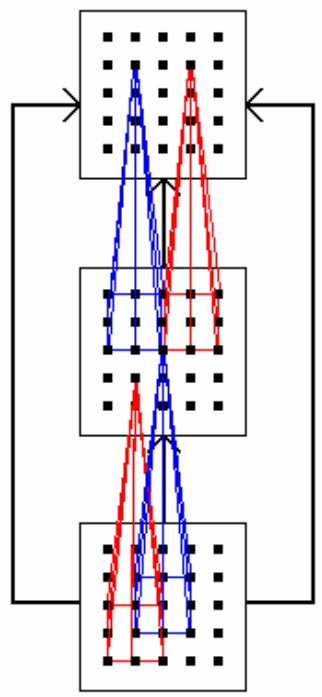

Fig.2 A schematic representation of neural network.

The input to a neuron belonging to the input layer is given from a real number in the range $[0,1]$ proportional to the gray value of corresponding pixel. In fact, first of all the image was converted from RBG to gray levels and in order to save time computing it was resized from 640 x 480 to 320 x 240 pixels.[5]

Since we are interested to eliminate the noise and to extract the original spiked signal, all the weights, initially, were put to be equal to 1.

The input value I_i for each neuron belonging to the *i*-th layer (except the input layer) was calculated through (1).

The goal is to obtain as output, the greatest number of neurons, set from 0 to 1, proportional to the target image. Therefore we will say that the state of the output level can be thought like a fuzzy set. The measure of fuzziness of such a set can be considered like the error of instability of the entire system (*i.e.*, neural network). Therefore we can use

the fuzziness value as a measure of the error produced by the system and use the back-propagation in order to adjust the weights until to eliminate the error (fuzziness). The measure of E can be adopted as a meaningful function of fuzziness index

$$E = g(I),$$

where I is the measure of fuzziness of a fuzzy set. After a first adjustment of the weights, the output of the neurons belonging to the output layer is used as feedback to the correspondent neurons belonging to the input layer. In the same way the second iteration will proceed. The iteration of weight adjustment shall continue until the net becomes stabilized (*i.e.*, the fuzziness error/index becomes minimum/negligible). When the net shall be stabilized, the state of output of neurons belonging to the output layer shall assume the values from 0 to 1 proportional to the target image.
The mathematical rules for the weight adjustment are the (11) for weight correction for Linear Index of Fuzziness and (17) for weight correction for Exponential Entropy.
Calculations were made using MATLAB® 7.0, The MathWorks, Inc, Natick, Mass, Simulation Toolbox Version 2.1.2.

4 Results

In the Fig. 3 is depicted the original image while the corrupted input image is shown in Fig. 4. In the Fig. 5 an 6 are shown the extracted object with Linear Index of Fuzziness and Exponential Entropy, respectively.

Fig. 3 Original RBG image, 640x480 pixels

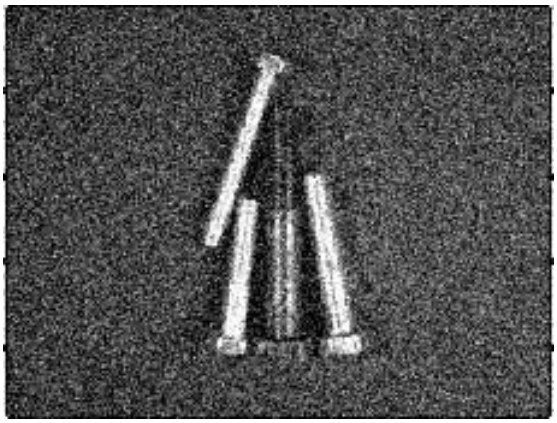

Fig. 4 Corrupted image, 640x480 pixels

The corrupted image of Fig. 4 was obtained by adding noise form $N(0,\sigma^2)$ with $\sigma = 40$.[6][7]

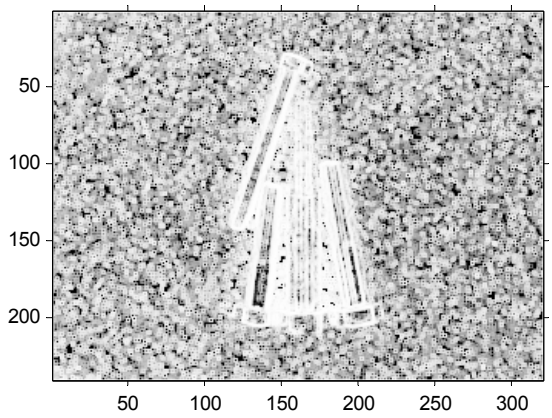

Fig. 5 Output image obtained by Linear Index of Fuzziness resized to 320 x 240 pixels

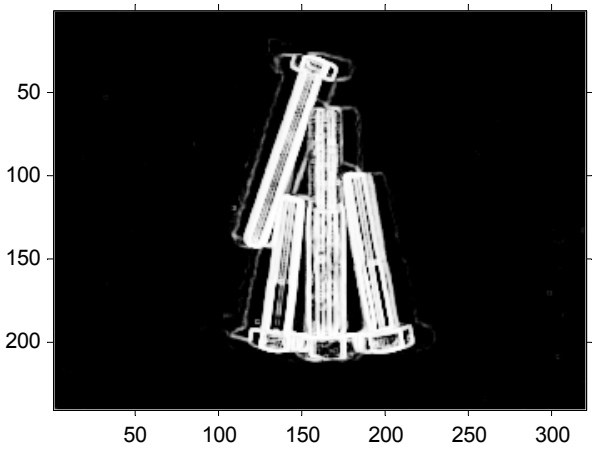

Fig. 6 Output image obtained by Exponential Entropy resized to 320 x 240 pixels

For the simulation η was put equal to 0.35.

By the comparison of the results, it is noticed that the objects extracted with entropy measures are better than those extracted by linear fuzziness measure. Of the two measures the exponential entropy seems to be more noise immune. This also for different values of η. A critical examination of the results reveal that the entropy measure enable the network to preserve the object boundaries as learning rate is very high near the most ambiguous region $\left(o_j \simeq 0.5\right)$.

For all the cases the convergence of the system was faster with reference to the entropy (*i.e.*, 2 epochs) and linear index (*i.e.*, 3 epochs).

4 Conclusion

In this paper was proposed a fuzziness measure to denoise a corrupted image by means of a neural network.

The output of the neurons in the output layer has been viewed as a fuzzy set. The instability of the network was modeled by means of measures of fuzziness. The weight updating rules were developed in order to reduce the error of the system. A comparative study was done in order to test the response of a network on the reduction of noise.

New investigations will be done in order to apply a multiscale approach based on wavelet transform.[8][9][10][11]

Appendix

The incremental change for a pattern is

$$\Delta w_{ji} \propto -\frac{\partial E}{\partial w_{ji}} = -\eta \frac{\partial E}{\partial w_{ji}} = -\eta \frac{\partial E}{\partial I_j}\frac{\partial I_j}{\partial w_{ji}}.$$

The input and the output of a node *i* are obtained, respectively, as

$$I_i = \sum_j w_{ji} o_j$$

and

$$o_i = f\left(I_i\right).$$

For the links connected to the output layer the change in weight is given by

$$\Delta w_{ji} = \eta\left(-\frac{\partial E}{\partial o_j}\right) f'\left(I_j\right) o_i \ .$$

For the links between the input and the hidden layer and also between two consecutive hidden layers the factor $\frac{\partial E}{\partial o_j}$ can be calculated as [12]

$$\frac{\partial E}{\partial o_j} = \sum_k \frac{\partial E}{I_k}\frac{\partial I_k}{\partial o_j} = \sum_k \frac{\partial E}{\partial I_k}\frac{\partial}{\partial O_j}\sum_i w_{ki} o_i$$

$$= \sum_k \frac{\partial E}{\partial I_k} w_{kj} = \sum_k \left(-\delta_k w_{kj}\right),$$

where

$$\delta_k = -\frac{\partial E}{\partial I_k} = -\frac{\partial E}{\partial o_k}\frac{\partial o_k}{\partial I_k} = -\frac{\partial E}{\partial o_k} f'\left(I_k\right).$$

Hence

$$\Delta w_{ji} = \begin{cases} \eta\left(-\dfrac{\partial E}{\partial o_j}\right) f'\left(I_j\right) o_i \\ \eta\left(\sum_j \delta_k w_{kj}\right) f'\left(I_j\right) o_i \end{cases}$$

for the output layer and other layers, respectively.

References:

[1]Weeks, A.R., *Fundamentals of Electronic Image Processing*, SPIE Optical Engineering Press and IEEE Press, 1996.

[2]Weickert, J., Ter Haar Romeny, B.M., and Viergever, M., Efficient and Reliable Schemes for Nonlinear Diffusion Filtering, *IEEE Trans. Image Processing*, N. 7, 1998, pp. 398-410.

[3] Ghosh, A. and Pal, S.K., Neural network, self-organization and object extraction, *Pattern Recog. Lett.*, Vol. 13, N. 5, 1992, pp 387-397.

[4] Chua, L.O., and Yang, L., Cellular neural network: Theory, *IEEE Trans. Circuits Syst.*, Vol. 35, 1988, pp. 1257-1272.

[5] Egiazarian, K., Astola, J., Helsingius, M., and Kuosmanen, P. Adaptive denoising and lossy compression of images in transform domain,

Journal of Electronic Imaging, 8/3, July, 1999, pp. 233-245.

[6] Black, M., Sapiro, G., Marimont, D. and Heeger, D., Robust Anisotropic Diffusion, *IEEE Trans. Image Processing*, N. 7, 1998, pp. 421-432.

[7] Bruce, A.G. and Gao, H.Y., Understanding WaveShrink: Variance and Bias Estimation, *Tech. Rep. 36*, StatSci Division of MathSoft, Inc., 1995.

[8] Chambolle, A., DeVore, R.A., Lee, N., and Lucier, B.J., Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelets, *IEEE Trans. Image Processing*, N. 7, 1998, pp. 319-335.

[9] Weaver, J.B., Yansun, X.,Healy, D.M.Jr. and Cromwell, L.D., Filtering noise from images with wavelet transforms, *Magnetic Resonance in Medicine*, N. 24, 1991, pp. 288-295.

[10] Romberg, J.K., Choi, H. and Baraniuk, R.J., Shift-Invariant Denoising using Wavelet-Domain Hidden Markov Trees, in *Conference Record of The Thirty-Third Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, October 1999.

[11] Starck, J.-L., Murtagh, F., and Bijaoui, A., *Image Processing and Data Analysis. The Multiscale Approach*, Cambridge University Press, 1998.

[12] Rumelhart, D.E., McClelland, J. and P.D.P. research group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1, Cambridge, MA: MIT Press, 1986.