Rank Ordered Neural Network Filters for Image Restoration

Hung-Hsu Tsai, Shih-Che Lo and Pao-Ta Yu

Department of Computer Science and Information Engineering
National Chung Cheng University
Chiayi, Taiwan 62107, R.O.C.
E-mail: csipty@ccunix.ccu.edu.tw, Fax: 886-5-272-0859

Abstract

In this paper, we propose a new class of nonlinear filters called *rank ordered neural network* (*RONN*) filters based on a detection-estimation strategy and neural network technique. *RONN* filters not only inherit the adaptive capability from neural networks but also exploit the rank information from selected observation samples in contrast to that of traditional *neural network* (*NN*) filters. As expected, the experimental results reveal that *RONN* filters are superior than that of other existing filters.

1 Introduction

In this paper, we propose a new class of nonlinear filters called *rank ordered neural network* (*RONN*) filters based on a detection-estimation strategy and neural network technique. *RONN* filters not only inherit the adaptive capability from neural networks but also exploit the rank information from selected observation samples in contrast to that of traditional *neural network* (*NN*) filters [1]. As known, *NN* filters must be trained in advance when they are ready to be employed. However, the training phase is required to spend a lot of computational operations in order to achieve the local minima. *RONN* filters also are required to be trained, but their training time can be reduced to arrive the local minima rapidly. The reason is that the number of training samples of *RONN* filters is less than that of *NN* filters. The filtering capability of *RONN* filters is further improved to invoke the selection of training samples to train *RONN* filters, whereas *NN* filters do not. The fact is justified in the experimental results.

2 Basic Concepts

2.1 Definitions and Concepts

Definition 2.1 *Let $M_{original}$ and M_{noisy} be two given images. Suppose that both $M_{noisy} = [x_{ij}]_{n\times n}$ and $M_{original} = [d_{ij}]_{n\times n}$ are given as well as the size of the sliding window is $r \times r$. We have*

$$S = \{P_{ij} = (X_{ij}, d_{ij}) : 1 \le i \le n, 1 \le j \le n\} \tag{1}$$

where X_{ij} represents an observation vector when the sliding window masks on M_{noisy} at position (i, j). In general, we are able to regard S as

$$S = T \cup C \tag{2}$$

where S is called as the set of observation patterns, T is called as the set of training patterns, and C is called as the set of checking patterns. □

However, the exact formula of input-output mapping is not easily found just by using the mathematical approach while the training patterns are available. The structure of this input-output mapping can be pictorially shown in Fig. 1. In this paper, F is de-

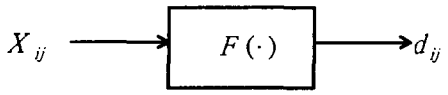

Figure 1: The desired function F.

fined as

$$F : \{0,1,2,\cdots,255\}^{r\times r} \rightarrow \{0,1,2,\cdots,255\}, \tag{3}$$

that is,

$$F(X_{ij}) = d_{ij}. \tag{4}$$

As before, it is difficult to obtain the formulation of F. Our aim, therefore, is triggered to approximate F by virtue of neural networks which play a role as the estimated function $\widehat{F}$ depicted in Fig. 2. $\widehat{F}$ is expected to minimize a sum of squared errors between its actual output $\widehat{d_{ij}}$ and the desired output d_{ij} when the mean squared error (MSE) criterion is applied to find the best possible mapping (or estimated function) for the set of training patterns.

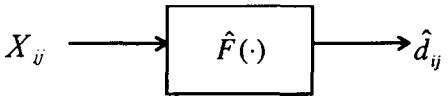

Figure 2: The estimated function $\widehat{F}$.

2.2 NN Filters

2.2.1 The Architecture of NN Filters

A p-q-1 multilayer perceptrons is employed to the design of the NN filter, its architectural layout shown in Fig. 3. It accepts p observation samples, such as $x_{i1}, x_{i2}, ..., x_{ip}$, has only one hidden layer with q neurons, and has a single output neuron in the output layer. In Fig. 3, the square stands for one of the input observation samples, and the circle represents the neuron. The neuron model is pictorially depicted in Fig. 4. Those neurons, are shown in Fig. 3, form a fully connected

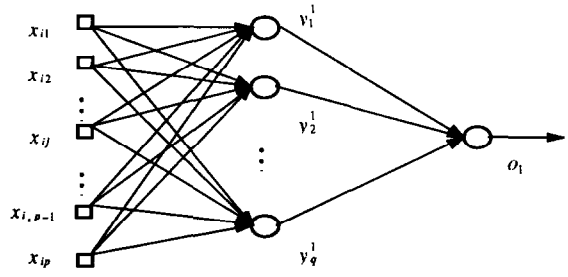

Figure 3: A Multillayer perceptron with a single hidden layer and output layer.

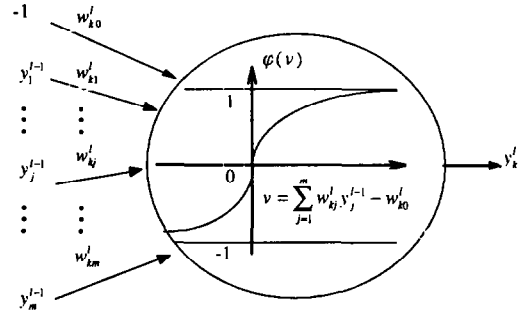

Figure 4: The neuron model of neuron k at layer l.

graph which means that each neuron is connected by either all inputs or all neurons located at the previous layer. Each connection edge between two neurons has a corresponding synaptic weight. To be specific, each neuron must be given a model depicted as Fig. 4. The neural model includes three basic terms. The first term is a set of synaptic weights, such as $w_{k0}^l, w_{k1}^l, ..., w_{km}^l$. The second term is a linear combiner,

$$v_k^l = \sum_{j=1}^{m} w_{kj}^l y_j^{l-1} - w_{k0}^l \tag{5}$$

where w_{k0}^l represents the threshold value. The third term is an activation function φ. Note that the output of the linear combiner is called internal value, namely v_k^l, and is taken as the input of φ. Then the output of φ is the actual output of the neuron. The details about the model of neuron can refer to [5]. As men-

tioned before, the function of neural network is decided by a set of synaptic weights. In this paper, Fig. 2 can be expressed as

$$\widehat{F}(X_{ij}, W) = \widehat{d}_{ij} \tag{6}$$

where W is a set of synaptic weight vectors. Consequently, it is important to apply a learning algorithm to obtain W with respect to a neural network with the specific structure $MLPs$. The network is expected to be the best possible input-output mapping for the training pattern set. The learning algorithm used in the experiment is introduced in the following subsection.

2.2.2 The Learning Algorithm of NN Filters

Actually, the neural network not only must include a fixed structure $MLPs$ but also must need a learning algorithm to decide W in order to perform well behavior for the set of training patterns. In analogy to the well-known back-propagation learning algorithm, our learning algorithm is derived based on the back-propagation learning algorithm, but the activation function is different from [5]. Our activation function is a hyperbolic tangent function, a sigmoidal nonlinear function, which is asymmetric corresponding to origin as well as for its amplitude of output between -1 and 1. It is shown in Fig. 4. Our learning algorithm adopts a hyperbolic tangent function, φ, to be the activation function. φ is defined as

$$\varphi(v) = a\tanh(bv) = a\left[\frac{1 - e^{(-bv)}}{1 + e^{(-bv)}}\right] \tag{7}$$

where the parameters a and b denotes the amplitude of $\varphi's$ outputs and $\varphi's$ slope, respectively. The shape of φ is depicted in Fig. 4 if parameter a is equal to 1.

Although, NN filters possess detail-preserving and adaptive capability when they are applied to restore noisy images. Only the local statistics of selected observation samples are taken as inputs of NN filters, but the rank information of selected observation samples is not utilized to improve further performance.

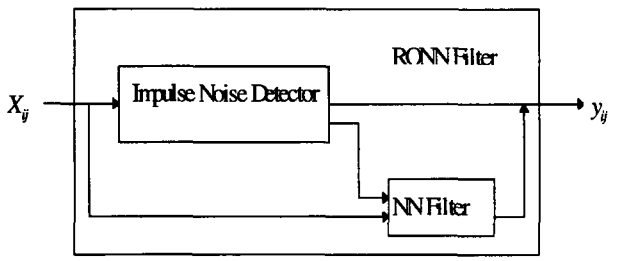

Figure 5: The RONN filter model

3 *RONN* Filters

The model of $RONN$ filters, shown in Fig. 5, is composed of two components. One component is the impulse noise detector, and the other is an NN filter. The filtering algorithm begins to exploit the impulse noise detector to decide whether the target pixel is a noise or not. The target pixel is immediately outputted with no change if it is not a noisy pixel. Otherwise, the NN filter is triggered to estimate the desired value with respect to the noisy pixel. Thus, the dash line denotes that the control of whether the NN filter is fired or not. In other words, the actual output of $RONN$ filter, y_{ij}, can be represented as

$$y_{ij} = \begin{cases} x_{ij}, & \text{if } x_{ij} \text{ is not a corrupted sample,} \\ \widehat{F}(X_{ij}), & \text{otherwise,} \end{cases} \tag{8}$$

where $\widehat{F}(\cdot)$ denotes the NN filter, and the observation sample vector X_{ij} is defined as

$$\begin{aligned} X_{ij} = {} & (x_{i-1,j-1}, x_{i-1,j}, x_{i,j+1}, x_{i,j-1}, x_{ij}, \\ & x_{i,j+1}, x_{i+1,j-1}, x_{i+1,j}, x_{i+1,j+1}) \end{aligned} \tag{9}$$

when a 3×3 sliding window is available and covers on a corrupted image at position (i, j).

The model we propose herein can further improve the filtering performance of NN filters, since it is more prone to be a best mapping associated with the training samples (i.e., input-output pairs) than that NN filters can be. NN filters are without the feature of the selecting training samples which is the most prominent difference from $RONN$ filters. That is, NN filters arrive at a worse

mapping associated with the training samples, since they are without invoking the selection of training samples when they are fed into network in the training phase. However, *RONN* filters employ the local statistics of selected observation samples to pass those training samples which may make a worse mapping. Hence, we believe that *RONN* filters achieve a best possible mapping for the given training samples, and perform a better behavior of noise cancellation than that of other proposed filtering techniques.

4 Experimental Results

The image "Baboon" with size 512×512 is one of the testing images in the computer simulation. The corrupted version of the original "Baboon" is corrupted by 10% impulse, that is, 10% positive impulse and 10 % negative impulse. In addition, the height of the impulse is ±100. The thresholds used in the *SD-ROM* filter are exerted by the *RONN* filter for impulse noise detection [2]. The filtering component of the *RONN* filter adopts the same architecture and learning algorithm as the *NN* filter [1].

We propose a new quantitative measurement as comparison index with respect to *MSE*,

$$\overline{MSE} = \frac{\sum_{p_e \in \mathrm{Q}} MSE_{p_e}}{\#(\mathrm{Q})} \qquad (10)$$

where $\mathrm{Q} = \{p_e = 0.2k | k = 2, \cdots, 10\}$, $\#(\mathrm{Q})$ is the cardinality of the set Q, and MSE_{p_e} stands for the *MSE* performance of a filtered image, when the noise model is the impulse model with an error probability p_e. Using the similar concept, $\overline{SNR}$ can be defined as

$$\overline{SNR} = \frac{\sum_{p_e \in \mathrm{Q}} SNR_{p_e}}{\#(\mathrm{Q})} \qquad (11)$$

The results using $\overline{MSE}$ and $\overline{SNR}$ as comparison indices are shown as Table 1, respectively.

The "Baboon" image corrupted by additive impulse noise $p_e = 0.12$ (i.e., [6%, 6%]) is shown in Fig. 6(a). The results of restored images are shown in Figs. 6(b) to (f) with 3×3 filters, respectively. We only show the results of restored images associated with $p_e = 0.12$ to present in the paper. The explanation is that the quantitative performances of $\overline{MSE}$ and $\overline{SNR}$ are close to that of the case when the impulse noise model with $p_e = 0.12$ is applied.

	$\overline{MSE}$	$\overline{SNR}$
Median [3]	238.55	18.72 dB
SD-ROM [2]	141.62	21.04 dB
RCRS (M=1) [4]	146.31	21.13 dB
NN (3000 epochs)	89.79	23.41 dB
RONN (3000 epochs)	65.39	24.92 dB

Table 1: Comparison results using different techniques via distinct comparison indices defined as Eqs. (10) and (11)

5 Conclusions

In this paper, a new approach of nonlinear filters associated with *NN* filters called *RONN* filters, which is controlled by the impulse noise detector related to whether an impulse noise is detected or not, has been proposed. Due to *NN* filters cooperate with the impulse noise detector to yield *RONN* filters. *RONN* filters not only employ the rank information from local statistics of observation samples but also reduce the training time in contrast to that of the traditional *NN* filters. Moreover, *NN* filters possess more adaptive capability than that of other existing filters. The computer simulation results reveal the fact that *RONN* filters provide better filtered performance than that of the others.

References

[1] P.-T. Yu and H.-H. Tsai, "On the design and study of adaptive capability for neural network filters in image restoration," *ICS*, NSYSU, Kaoshiung, R.O.C., 1996.

[2] E. Abreu and S. K. Mitra, "A signal-dependent rank ordered mean (SD-ROM) filter: a new approach for removal of impulses from highly corrupted images," in

Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, ICASSP-95, Detroit, MI, May 1995.

[3] I. Pitas and A. N. Venetsanopoulos, *Nonlinear Digital Filters - Principles and Applications*, Kluwer Academic Publishers, 1990.

[4] R. C. Hardie and K. E. Barner, "Rank conditioned rank selection filters for signal restoration," *IEEE Trans. on Image Processing*, vol. 3, no. 2, pp.192-206, March 1994.

[5] S. Haykin, *Neural Networks*, Macmillan College Publishing Company, 1995.

Figure 6(a) Noisy image.

Figure 6(b) Median filter.

Figure 6(c) *SD-ROM* filter.

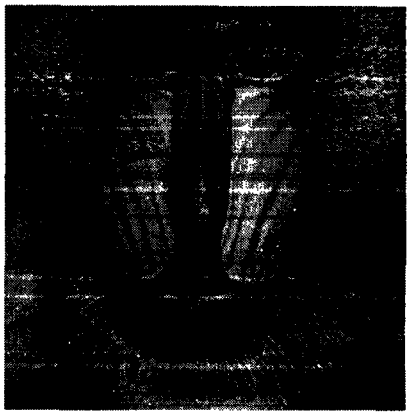

Figure 6(d) *RCRS* filter.

Figure 6(e) *NN* filter.

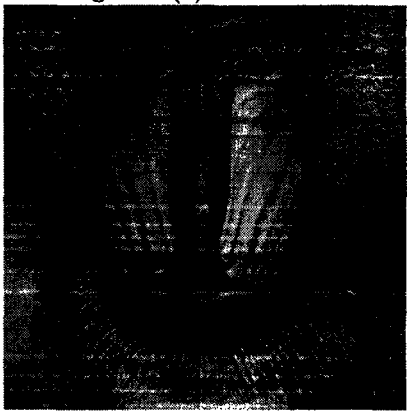

Figure 6(f) *RONN* filter.