Image Filtering Using Functional Link ANN

Raghunath Rout
Assistant Professor, DRIEMS

Abstract -**This project work looks into the application of Functional Link ANN to Adaptive Digital Image Filter. In this project work, a functional link neural network based approach for filtering of grey image was carried out. An image gets noisy at the time of transmission. Different noises added in this image are additive white Gaussian noise, Salt & Pepper noise, Poisson noise, Speckle noises. To avoid this noise, adaptive image filters are developed, which may be utilized in online application. The filter used here is FLANN based adaptive filter. In contrast to a feed forward ANN structure i.e. a multiplayer perceptron (MLP) the FLANN is basically a single layer structure in which non-linearity is introduced by enhancing the input pattern with nonlinear function expansion. Through experimentation with different noisy images the following observation is made that FLANN structure requires much less computation time than that of MLP.**

1. INTRODUCTION

Image processing is a major field of use in present day. When an image is taken, stored, processed and sent to destination, it gets noisy due to different reasons at the time of acquisition, processing and at the time of transmission of the images. Image is a two-dimensional function $f(x, y)$ where x and y spatial (plane) coordinates and the amplitude of $'f'$ at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y and the amplitude value of $'f'$ are all finite, discrete quantities we call the image as a digital image. When x, y and the amplitude values of $'f'$ are all finite, discrete quantities i.e. 0 or 1 (for a black and white image respectively), we call the image a digital image. Digital image processing is electronic data processing on a 2-D array [1]of numbers. The array is a numeric representation of an image. A real image is formed on a sensor when an energy emission strikes the sensor with sufficient intensity to create a sensor output. The energy emission can have numerous possible sources (e.g., acoustic, optic, etc.). When the energy emission is in the form of electromagnetic radiation within the band limits of the human eye, it is called visible light. Some objects will reflect only electromagnetic radiation. Others produce their own, using a phenomenon called radiancy. Radiancy occurs in an object that has been heated sufficiently to cause it to glow visibly. Visible light images are a special case, and it appears with great frequency in the image processing literature. Another source of images includes the synthetic images of computer graphics. These images can provide controls on the illumination and material properties that are generally unavailable in the real image domain. This chapter reviews some of the basic ideas in digital image processing. In acquiring image with CCD camera the low light level, sensor temperatures, sensor noise, poor illumination, and high temperature of the surroundings are the major factors affecting the amount of noise in the image. Again when this image transmitted to the destination, it corrupted due to interference in the channel use for transmission. Image transmitted from in wireless network might be corrupted as a result of lighting or other atmospheric disturbances. Different noises added in this image are additive white Gaussian noise, salt and pepper noise, Both Gaussian and salt and pepper noise, Rayleigh Noise, Erlang Noise, Exponential Noise, Uniform Noise, and Periodic Noise etc.

In this thesis various noise conditions are studied and an efficient nonlinear and adaptive digital image filters are designed to suppress Salt & Pepper noise. The developed filter may use for offline or for online applications.

2. ADAPTIVE DIGITAL IMAGE FILTERS AND RESULTS

The fixed filters are good for offline applications where an image is filtered by using computer and some software algorithm. Here the expert knows the type of noise and the noise power level. So he can apply a specific filtering operation depending upon the requirement. He may change the filtering operation depending upon the requirement. But such human decision cannot be taken for an online and real time operation. For example when data is transmitted in channel the noise added varies from time to time.

Again it changes in fraction of second. A human expert can't take decision to choose a filter at that small time.

To avoid different limitations of fixed filters, adaptive filters are designed that adapt themselves to the changing conditions of signal and noise. The filter characteristics change as the signal statistics or noise type or noise power level very from time to time. There are several types of nonlinear filter which uses the following techniques for developing image filter. They are (i) Neural Network based filter (ii) Fuzzy-Logic based filter (iii) Bacteria Technology. (iv) Ant Colony. (v) Genetic Algorithm based filter. Two broad categories of adaptive image filters developed for efficient noise suppression and presented in this chapter are (i) MLP-BP neural network adaptive filter (ii) FLANN based filter. An adaptive filter is a filter whose behavior changes based on the statistical characteristics of the image inside the filter region defined by $m \times n$ rectangular window s_{xy} . Adaptive filter has performance far better than the non-adaptive types but filter became more complex.

A. *2.1 Introduction to ANN*

Artificial Neural Networks are being touted as the wave of the future in computing. They are indeed self-learning mechanisms which don't require the traditional skills of a programmer. But unfortunately, misconceptions have arisen. Writers have hyped that these neuron-inspired processors can do almost anything. These exaggerations have created disappointments for some potential users who have tried, and failed, to solve their problems with neural networks. These application builders have often come to the conclusion that neural nets are complicated and confusing. Unfortunately, that confusion has come from the industry itself. An avalanche of articles has appeared touting a large assortment of different neural networks, all with unique claims and specific examples. Currently, only a few of these neuron-based structures, paradigms actually, are being used commercially. One particular structure, the feed forward, back-propagation network, is by far and away the most popular. Most of the other neural network structures represent models for "thinking" that are still being evolved in the laboratories. Yet, all of these networks are simply tools and as such the only real demand they make is that they require the network architect to learn how to use them.

B.

C. *2.2 Architecture of neural network.*

D. *2.2.1 Feed-forward networks.*

1) *Feed-forward ANNs allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-forward ANNs tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition.*

2) *2.2.2 Feed-back networks.*

Feedback networks can have signals traveling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organizations.

E. 2.3 Supervised Learning

The vast majority of artificial neural network solutions have been trained with supervision. In this mode, the actual output of a neural network is compared to the desired output. Weights, which are usually randomly set to begin with, are then adjusted by the network so that the next iteration, or cycle, will produce a closer match between the desired and the actual output. The learning method tries to minimize the current errors of all processing elements. This global error reduction is created over time by continuously modifying the input weights until acceptable network accuracy is reached.

F. *2.4 Unsupervised Learning*

Unsupervised learning is the great promise of the future. It shouts that computers could someday learn on their own in a true robotic sense. Currently, this learning method is limited to networks known as self-organizing maps. These kinds of networks are not in widespread use. They are basically an academic novelty. Yet, they have shown they can provide a solution in a few instances, proving that their promise is not groundless. They have been proven to be more effective than many algorithmic techniques for numerical aerodynamic flow calculations. They are also being used in the lab where they are split into a front-end network that recognizes short, phoneme-like fragments of speech, which are then passed on to a back-end network. The second artificial network recognizes these strings of fragments as works.

1. Present a training sample to the neural network.
2. Compare the network's output to the desired output from that sample. Calculate the error in each output neuron.
3. For each neuron, calculate what the output should have been, and a *scaling factor*, how much lower or higher the output must be adjusted to match the desired output. This is the local error.
4. Adjust the weights of each neuron to lower the local error.
5. Assign "blame" for the local error to neurons at the previous level, giving greater responsibility to neurons connected by stronger weights.
6. Repeat the steps above on the neurons at the previous level, using each one's "blame" as its error.

Fig-1: Steps of Backpropagation Algorithm

G. 3.BACKPROPAGATION ALGORITHM

Backpropagation is a supervised learning technique used for training artificial neural networks. It is most useful for feed-forward networks (networks that have no feedback, or simply, that have no connections that loop). The term is an abbreviation for "backwards propagation of errors". Backpropagation requires that the transfer function used by the artificial neurons (or "nodes") be differentiable.

H. 3.1WEIGHT UPDATION IN ACKPROPAGATION

Consider the network above, with one layer of hidden neurons and one output neuron. When an input vector is propagated through the network, for the current set of weights there is an output *Pred*. The objective of supervised training is to adjust the weights so that the difference between the network output *Pred* and the required output *Req* is reduced. This requires an algorithm that reduces the absolute error, which is the same as reducing the squared error, where: Network Error=*Pred – Req* =E

The algorithm should adjust the weights such that E^2 is minimized. Back-propagation is such an algorithm that performs a gradient descent minimization of E^2. In order to minimize E^2, its sensitivity to each of the weights must be calculated. In other words, we need to know what effect changing each of the weights will have on E^2. If this is known then the weights can be adjusted in the direction that reduces the absolute error. The approximation used for the weight change is given by the delta rule [2][3][5]

$W_{AB}(new)) = W_{AB}(old) - \eta \frac{\partial E^2}{\partial W_{AB}}$ Where η is the learning rate parameter, which determines the rate of learning, and $\frac{\partial E^2}{\partial W_{AB}}$ is the sensitivity of the a error, E^2, to the weight W_{AB} and determines the direction of search in weight space for the new weight $W_{AB(new)}$

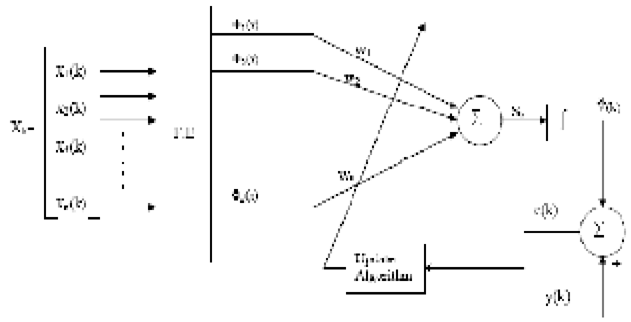
3.2 FILTER DESIGN USING FUNCTIONAL ARTIFICIAL NEURAL NETWORK

The FLANN, which is initially proposed by Pao, is a single layer artificial neural network structure capable of forming complex decision regions by generating nonlinear decision boundaries. In a FLANN the need of hidden layer is removed.

In contrast to linear weighting of the input pattern produced by the linear links of a MLP, the functional link acts on an element or the entire pattern itself by generating a set of linearly independent functions. Here the functional expansion block comprises of exponential series or subset of Chebyshev polynomials. Let input to this structure is $X = [x_1 x_2]'$. An enhanced pattern obtained by using functional expansion is given by $X = [1 x_1 T_2(x) .... x_2 T_2(x_2) ....]'$. [7]

Here input pattern of the s iteratively till all pattern of the image gets completed. The whole process continues fornoisy image is sent in the input node of the FLANN structure and the enhanced pattern is obtained. The target will be the corresponding single pixel. This process continue 10 to 100 times to find out the error plot.

3.3 THE STRUCTURE OF THE FLANN NETWORK



F.E: - Functional Expansion

Structure of FLANN with a single output

The function approximation capability of FLANNs can be understood as follows: Take a MLP with d = 2 input units and h = 3 sigmoidal hidden units in the lone hidden layer as an example. The output function calculated for this neural network is[9]

$$y_k = G_k \Sigma_{j=1}^{h} w_{jk} * \phi_j(x)$$

Where W $_{jk}$ is the weight connecting hidden unit j with output unit k and Gk the activation function employed by the output layer neurons.

3.4 LEARNING WITH THE FLANN

3.4.1 ALGORITHM:

Let K number of patterns be applied to the network in a sequence repeatedly. Let the training sequence be denoted by $\{X_k, Y_k\}$ and the weight of the network be W(k), where k is the discrete time index given by k=k+λk, for λ=0,1,2…, and k=0,1,2….,K. At k^{th} instant, the n-dimensional input pattern and the m-dimensional FLANN output are given by $X_k=[x_1(k)\ x_2(k) \ldots\ldots x_n(k)]^T$ and $\check{Y}(_k)=[\ \check{Y}_1\ (k)\ \check{Y}_2\ (k) \ldots\ldots\ \check{Y}_m(k)]^T$ respectively. Its corresponding target pattern is represented by $Y(_k)=[\ y_1\ (k)\ y_2\ (k) \ldots\ldots\ y_m(k)]^T$. The dimension of input pattern increases from n to N by a basis function Φ given by $\Phi(X_k)=[\ \Phi_1(X_k)\ \Phi_2(X_k) \ldots\ldots\ \Phi_N(X_k)]^T$. The (mxN)- dimensional weight matrix is given by $W(k)=[W_1(k)W_2(k) \ldots\ldots W_m(k)]^T$ where $W_j(k)$ is the weight vector associated with j^{th} output and is given by $W_j(k)=[W_{j1}(k)W_{j2}(k) \ldots\ldots W_{jN}(k)]$. The j^{th} output of the FLANN is given by

$$y\,^\wedge{}_j(k) = \rho(\Sigma_{i=1}^{N} W_{ji}(k)\phi_i(X_k)$$

$$= \rho(w_{ji}(k)\phi^T(x_k))$$

for j=1,2,3….m.

the corresponding error be denoted by $e_j(k)=y_j(k)$- $\check{Y}_j(k)$. Using the BP algorithm for single layer, the update Let rule for all the weights of the FLANN is given by [2][3][]10]

$w(k+1) = w(k) + \mu\delta(k)\phi(x_k)$ where $W(k)=[W_1(k)W_2(k) \ldots\ldots W_m(k)]^T$ is the m x N dimensional weight matrix of the FLANN at the kth time instant , $\delta(k)=[\ \delta_1(k)\ \delta_2(k)\ \ldots\ldots\ \delta_m(k)]^T$, and $[\delta_j(k)=(1-\check{Y}_j(k)^2)e_j(k)]$.

4.SIMULATION STUDY

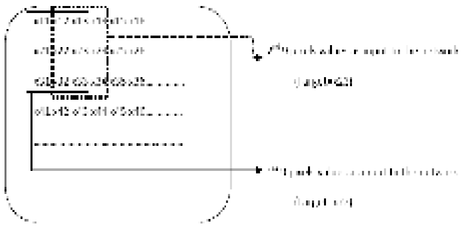
4.1 PLATFORM SPECIFICATION

In this proposed thesis the simulations are carried out with MatLab ver7.0 used to run in Pentium-4 processor, 2.8 GHz, 512MB RAM.

4.1 OBJECTIVE ANALYSIS

In this proposed work simulations were carried out to evaluate the performance of Functional link neural network with Trigonometric functional expansion and Chebyshev functional expansion for denoising of image

corrupted with different noises. Here the 9 inputs of this network are the 3 x 3 window of the noisy image and the target was the middle value of the matrix window.[4][5][6]



3*3 Sliding window

Figure represents an image in matrix format. So here the first 3 x 3 window is selected as input, and target is selected as the middle value of the window, i.e. x22.. These 9 pixels values are given as input to the FLANN and it is then expanded up to 45 using trigonometric or chebyshev expansion. The initial weights have been taken from the range of -0.5 to +0.5 and randomly distributed between 45 layers. The network has been trained by using various parameters like bias=1/+0.5/-0.5, learning rate=0.001.

A text image has been taken and corrupted with different noise of 10% density. Figure 1 to figure 5 shows the weight updation plot between MLP and FLANN. The plots shows that after approximately 20 iterations the weight updation remains constant and nearly to zero in FLANN. After training, testing has been done with the noise of 20% density. After getting the filtered image, the comparison between MLP and FLANN were carried out in terms of Noise Reduction in DB (NRDB).[7][8]

$$NRDB = 10 \times \log_{10}(MSE_{in} / MSE_{out})$$

$$MSE = \frac{1}{mn} \Sigma_{i=0}^{m-1} \Sigma_{j-0}^{n-1} \| I(i.j) - K(i,j) \|^2$$

MSEin-> Error between original and noisy image.
MSEout-> Error between original and filtered image.

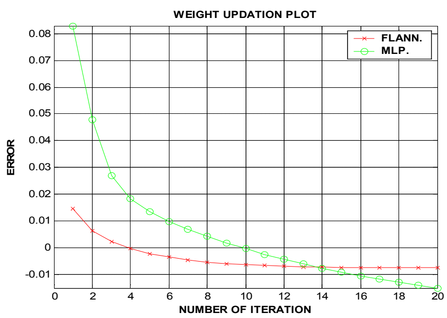
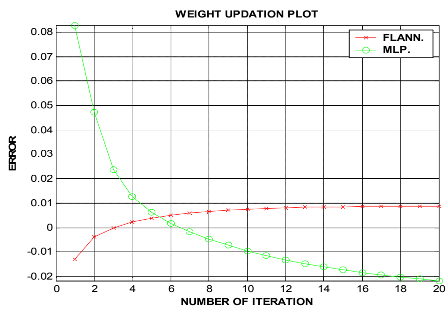
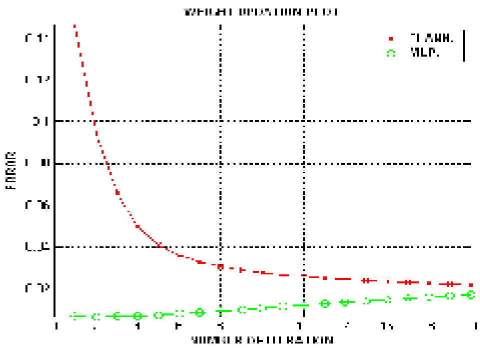


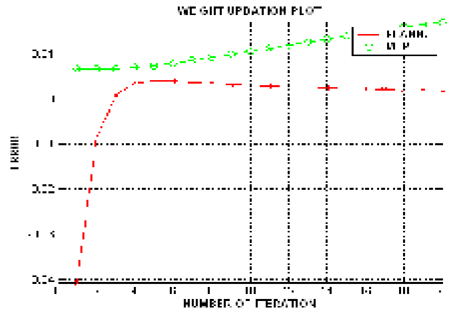
Figure-4.1



Noise: Gaussian, Training=0.1, Testing=0.2, Expansion=Chebyshev Wt=+0.5 to -0.5, Bias=1



Noise: salt & pepper, Training=0.1, Testing=0.2, Expansion=Trigonometric, Wt=+0.5 to -0.5, Bias=1



Noise: Speckle, Training=0.1, Testing=0.2, Expansion=Trigonometric Wt=+0.5 to 1.5, Bias=-0.5

5.Conclusion

The objective of the paper was to present the FLANN algorithm as better than MLP. It also be observed that if we will change the noise factor, biases and weights, then also FLANN gives better accuracy in trigonometric expansion or chebyshev.

In presence of noise at input pattern FLANN performs better than MLP. It has better convergence speed that that of MLP. Due to absence of Hidden layer, it's time complexity is less. It has less number of interconnection weights and biases, which also computationally cheap. Due to this reason it is suitable for on-line applications. It is able to approximate linear as well as non-linear functions. From the results obtained from the work performed in this project, it is concluded that FLANN is computationally cheap and has less convergence time and denoises a digital image more efficiently that of MLP. It gives improved approximation results than that of MLP. Therefore, it is expected that FLANN is likely to re-ignite the interest of research community.

References

[1] G. L. Sicuranza and A. Carini, "On a class of nonlinear filters," in Festschrift in Honor of Jaakko Astola on the Occasion of his 60th Birthday, ser. TICSP series No. 47, I. Tabus, K. Egiazarian, and M. Gabbouj, Eds. Tampere, Finalnd: TICSP, 2009, pp. 115–144.

[2] V. J. Mathews and G. L. Sicuranza, Polynomial Signal Processing. New York: Wiley, 2000.

[3] J. Kim and K. Konstantinou, "Digital predistortion of wideband signals based on power amplifier model with memory,"Electron. Lett., vol. 37, pp. 1417–1418, Nov. 2001.

[4] Y. H. Pao, Adaptive Pattern Recognition and Neural Networks. Reading, MA: Addison-Wesley, 1989.

[5] J. C. Patra and R. N. Pal, "A functional link artificial neural network for adaptive channel equalization," Signal Process., vol. 43, pp. 181–195, May 1995.

[6] D. P. Das and G. Panda, "Active mitigation of nonlinear noise processes using a novel filtered-s LMS algorithm," IEEE Trans. Speech Audio Process., vol. 12, no. 3, pp. 313–322, May 2004.

[7] G. L. Sicuranza and A. Carini, "On the accuracy of generalized Hammerstein models for nonlinear active noise control," in Proc. IEEE Instrum. Meas. Technol. Conf., Sorrento, Italy, Apr. 24–27, 2006, pp. 1411–1416.

[8] G. L. Sicuranza and A. Carini, "Piecewise-linear expansions for nonlinear active noise control," in Proc. IEEE ICASSP'08, Toulouse, France, May 14–19, 2006, vol. III, pp. 209–212.

[9] E. L. O. Batista, O. J. Tobias, and R. Seara, "A fully LMS adaptive interpolated Volterra structure," in Proc. IEEE ICASSP'06, Las Vegas, NV, Mar. 30–Apr. 4 2008, pp. 3613–3616.

[10] D. R. Morgan, Z. Ma, J. Kim, M. G. Zierdt, and J. Pastalan, "A generalized memory polynomial model for digital predistortion of RF power amplifiers," IEEE Trans. Signal Process., vol. 54, no. 10, pp. 3852–3860, Oct. 2006