Analysis of Neural Network Edge Pattern Detectors in Terms of Domain Functions

[1]MAHER I. RAJAB AND [2]KHALID A. AL-HINDI
Department of Computer Engineering
University of Umm Al-Qura
P.O. Box 5555, Mecca
KINGDOM OF SAUDI ARABIA
[1]rajabmah@uqu.edu.sa; [2]alhindi@uqu.edu.sa; http://www.uqu.edu.sa/ce

Abstract: - This paper investigates the analysis of feed-forward BP neural network that has been trained to detect noisy edge patterns, so as to achieve close insight into their internal functionality. The analysis of neural network edge detector's hidden units, as templates, were analysed into three gradient components: low pass or averaging, gradient, and second-order gradients. The weights between NNets hidden units and their output units represent the importance of the hidden unit's edge detection outcome. To this purpose, the elements of the NNets, that have been trained to detect prototype noisy edge patterns with various angle of operation, were analysed in terms of domain functions. The results show that the NNets analysis using the domain functions method could confirms the results of NNets recognition accuracies. Although the work presented only gives some analysis results for the units in the NNets hidden units, it should be clear that a characterization of the neural network as a whole could also be derived from these results.

Key-Words: - Neural networks analysis, domain functions, BP, recognition accuracy, domain-specific base functions, Taylor series coefficients.

1 Introduction

A neural network analysis method was introduced and presented in earlier publications by van der Zwang [1]. This method utilizes domain-specific base functions that are easy to interpret by the user and can even be used to optimize neural network systems. For example, a problem domain, in a recent publication by van der Zwang [2], the case of edge detection - a digital image processing technique - was investigated and in particular the 2-dimensional differential operators (base functions). It was also found that it is feasible to analyze the trained neural network's edge detection in terms of gradient filter components (zero-order or low-pass, gradient, and second-order gradients; in a similar manner when analyzing other digital image operators) and this will enhance the understanding of the neural net's functionality [3].

Zwang et al. used the two training images to synthetic images to analyze a neural network edge detector; the neural networks were of the feed-forward error-backpropagation (BP) type. The aim was to support the analysis of neural network edge detection so as to gain insight into their internal functionality. Hidden nodes, whose weights were regarded as a template, which is similar to any image filter (e.g. Kirsch or Sobel templates), and its Taylor series coefficients, were used to analyze the order of this template. Interesting results were found when some small neural networks edge detectors were trained with sharp edges whilst others were trained with sharp, blurred, and noisy variants of the same images. For the first category of neural network, the Taylor series coefficients analysis shows that most hidden units act as both a gradient filter and also have second-order gradient behaviour, but they do not have a significant low-pass (zero-order gradient) component. When analysed the second category of network's hidden units have similar gradient behaviors as the previous one, while the second-order gradient components are somewhat stronger. On the other hand, the low-pass components are notably present. It is worth mentioning that the amount of noise was not precisely specified in the method of Zwaag et al. method while analyzing various neural network architectures.

In contrast, we have evaluated the performance of BP neural network, in the recognition of prototype noisy edge patterns, over a range of signal to noise ratios. Moreover, neural networks template analysis, of hidden units; whose weights were regarded as a template, has been chosen in this paper for two main reasons: (1) to analyze the order of this template, (2) as an attempt to find out if the results of template analysis method could support the corresponding results of overall performance of neural network pattern recognition.

2 Neural Network Analysis Background

In the past 10 years, researchers have attempted to analyse the internal functionality of neural networks. The main purpose of this analysis was to arrive at a closer understanding of the internal behavior of different neural networks systems for different applications. For example, great improvements have been made by the rule extraction method [4] in decision-making systems and other systems that can easily be expressed as a set of rules. Another example of the so-called sensitivity analysis (nonparametric statistical analysis method) [5] was also used to analyse neural networks systems with relatively few inputs. However, for both methods it would be difficult to analyze most of neural network systems with so high a dimension. This is because extracting rules becomes too large to interpret by the rule extraction method, or so nonlinear for the sensitivity analysis method (perhaps part of the input space could be valid for this analysis) [2-3]. A neural network analysis method was introduced and presented in earlier publications by van der Zwang [1-3]. This method utilizes domain-specific base functions that are easy to interpret by the user and can even be used to optimize neural network systems. For example, a problem domain, in a recent publication by van der Zwaag [2-3], the case of edge detection - a digital image processing technique - was investigated and in particular the 2-dimensional differential operators (base functions). It was also found that it is feasible to analyze the trained neural network's edge detection in terms of gradient filter components (zero-order or low-pass, gradient, and second-order gradients; in a similar manner when analysing other digital image operators) and this will enhance the understanding of the neural net's functionality [2-3].

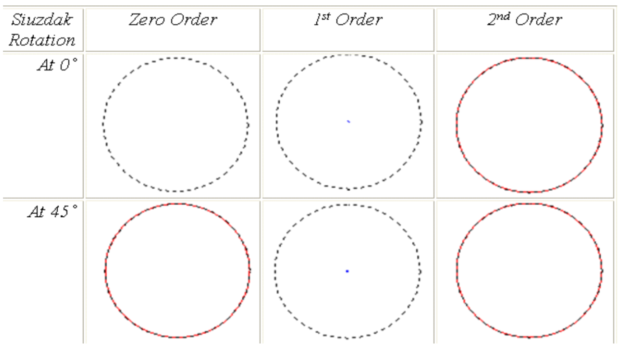
The work presented by Zwaag et al. [2] has been intended to analyze different sizes of neural networks edge detection. Two synthetic images were used in the training set of neural networks. The aim was to gain insight into the internal functionality of the neural networks edge detection responses (zero, gradient, and second-order gradients). The neural networks were of the feed-forward error-backpropagation (BP) type. Weights of hidden nodes were regarded as a template; which is similar to any mask or template of a spatial filter (e.g. templates of Kirsch or Sobel spatial filters). The Taylor series coefficients were used to analyze the order of Siuzdak template filter [6] according to [2]:

$$\beta_{\theta,i,j} = (-1)^{i+j} \sum_{n=-N}^{N} \sum_{m=-M}^{M} w_{n,m} \sum_{k=0}^{i} \sum_{l=0}^{j} \frac{(-1)^{l} n^{k+l} m^{i+j-k-l}}{k!\,l!\,(i-k)!\,(j-l)!} (\sin\theta)^{i-k+l} (\cos\theta)^{j+k-l} \tag{1}$$

From (1) we can realize of which orders of the differential operators the filter consists, i.e. those i and j that give the larger $\beta_{\theta,i,j}$, and in which direction(s) these operators work optimally. In other words, the possible angel(s) θ such that $\beta_{\theta,i,j}$ is maximal for certain i and j .This can be represented graphically by a polar graph, see Table 1, to draw the value of $\beta_{\theta,i,j}$ as a function of θ for various i and j .

0	0	-1.5	-1.5	-1.5	-1.5	0	0
0	-1.5	-0.25	-0.25	-0.25	-0.25	-1.5	0
-1.5	-0.25	1	2.5	2.5	1	-0.25	-1.5
-1.5	-0.25	2.5	2.5	2.5	2.5	-0.25	-1.5
-1.5	-0.25	2.5	2.5	2.5	2.5	-0.25	-1.5
-1.5	-0.25	1	2.5	2.5	1	-0.25	-1.5
0	-1.5	-0.25	-0.25	-0.25	-0.25	-1.5	0
0	0	-1.5	-1.5	-1.5	-1.5	0	0

(a)



(b)

Table 1. (a) Siuzdak filter. (b) Low pass or zero-order (first column of shapes), gradient (second column of

shapes), and second-order (third column of shapes) gradient analysis results.

3 Methodology

In order to test the analysis method investigated, we trained several neural networks for edge detection. The NNs are of BP type, with 3x3 inputs, 4 to 8 units in the single hidden layer, and 1 output. In BP type all hidden units used sigmoid activation functions. Initial experiments were performed using the following technique in generating additive noise to various prototype edge patterns: noise is generated at one low level according to the noise signal itself i.e. noise generated at a standard deviation value: e.g. sN = 0.1. It should be noted that total size is calculated according to the following equations [7]:

Total noisy patterns = P×R×N (2)

Total size = P×R×N×W (3)

Total size (Kbytes) = P×R×N×W/1024 (4)

where
P = no. of patterns,
R = repetition of patterns,
N = noise levels,
W = window size

Since each edge pattern occupies a 3x3 window then W here is calculated as 9 bits. Repetition here means that the same prototype edge pattern is repeated R times. The aim of this is to assist in providing enough information to the input layers of a neural network; i.e. the subset of input data should be chosen so that is adequately represents the input set of a problem domain. This case is similar to "overfitting" problem; if too few input/output pairs are chosen to train the NN, then the network will "memorize" those examples and the network will not be able to generalize to new i/o pairs that it has not been explicitly trained to recognize [8].

Rotation is also considered when generation edge patterns. Fig. 2 illustrates the operation to obtain eight rotated versions of the original pattern shown in Fig. 2(a).

1	0	1
0	1	0
0	0	0

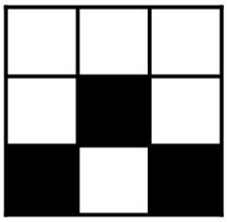
0	1	0
0	1	1
0	0	0

(a) (b)

Fig. 1. Illustrating the operator $R_{\pi/4}$. (a) input image and (b) image after applying the operator.

3.1 SNR Analysis for prototype edge patterns

As an example, when the training set consist of 1,312 roof edge patterns in a 3x3 window (Fig. 2), then the generated noise at standard deviation $\sigma_N = 0.3$ is required to have signal to noise ratio $SNR = 33.3$. This can be shown by simple substituting of the values of S*NR* [7], *N*, and S_{ij} in (4) and solving for σ_N.



$S_{ij} = \{0,0,0,0,1,0,1,0,1\}$
Black represents 1
White represents 0

Fig. 2. Prototype pattern of the roof edge.

$$SNR = \frac{1}{N\,\sigma_N^2}\sum_{j=1}^{J}\sum_{i=1}^{I} S_{ij}^2 \qquad (5)$$

where

N = total number of pixels,
S_{ij} = signal pixel value,
i = pixel index,
j = *pattern* index,
σ_N = standard deviation of noise

Therefore,

$$\sigma_N^2 = \frac{1}{N}\frac{1}{SNR}\sum_{j=1}^{J}\sum_{i=1}^{I} S_{ij}^2 = \frac{1}{50\times 9}\frac{1}{6}\sum_{j=1}^{50}\sum_{i=1}^{9} S_{ij}^2 = 0.01$$

and

$$\sigma_N = 0.1$$

4 Results and Discussion

The above described method was used to generate a training set consisting of 1,312 edge patterns which are similar to the patterns shown in Fig.2. These patterns are classified into two categories: edges and non-edges. The training set consists of 8 types of edge patterns or *P*=8. The original pattern shown in Fig.2(a) is rotated by $\pi/4$ (to generate these patterns with *R* = 100. The non-edge patterns consist of 256 patterns to represent simple grey scale levels of background. Each non-edge pattern is generated twice. Therefore, the total number of patterns included in the training set is 1,312 patterns. Gaussian noise were also added to these patterns at low σ_N = 0.1 or SNR = 33.3. In is interesting to mention here that in order to enhance the performance of the training phase, we have considered randomly reordering the sequence of the generated patterns [9].

Fig. 3 shows the recognition accuracy of five NNets (hidden nodes between 4 to 8). New test patterns have been generated with the same size of 1,312 at ten discrete signal to noise ratios at: SNR = 33.3, 8.33, 3.70, 2.08, 1.33, 0.93, 0.68, 0.52, 0.41, and 0.33. At high SNR the recognition accuracy reaches 98.88 % of the edge patterns for NNets with 8 hidden units. At low SNR, the recognition rate for all NNets begin to decrease, but the NNets with larger hidden units still achieves the highest rate. It is interesting to mention here that the neural network would memorize the prototype noisy edge patterns when an image contains very noisy patterns. Experimentally, we have found that the noise levels (at σ_N = 50 and 75) would make the neural network fail to detect edges.

Fig.4 and Fig.5 show the analysis of neural network edge detector's hidden units, whose weights can be regarded as a template, using the method suggested by van der Zwaag [2]. It is clearly seen from the two figures that all of the analysed templates show low pass behaviour (zero-order gradient). Low-pass or averaging behaviour makes the network less sensitive to noise and improves the edge detection ability. First and second order gradient behaviors are increased with the increase of hidden units. When comparing these results with results shown in Fig.3, the recognition accuracy is also increased with the increase of hidden units.

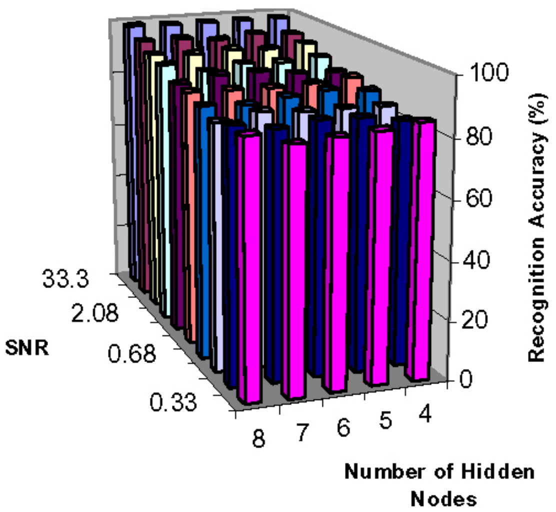


Fig. 3. Comparisons between recognition accuracies via SNR's for five neural networks.

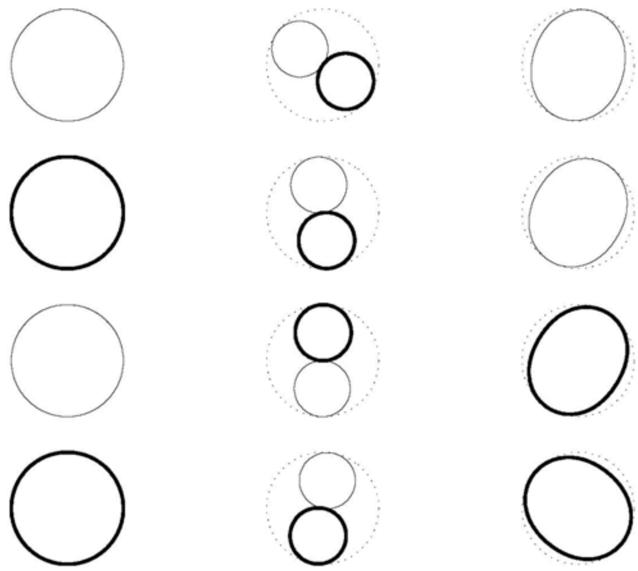


Fig. 4. Low pass or zero-order (first column of shapes), gradient (second column of shapes), and second-order (third column of shapes) gradient analysis results for all 4 hidden units of a 9×4×1 neural network edge detector.

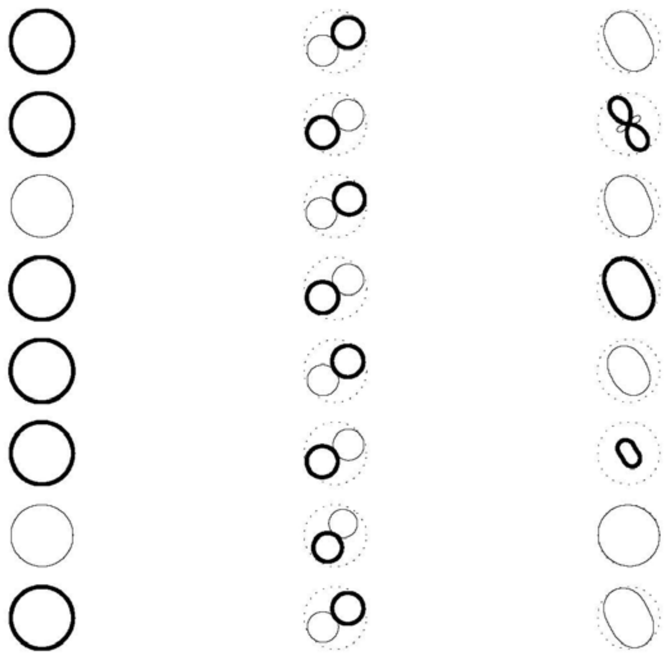


Fig. 5. Low pass or zero-order (first left column of shapes), gradient (second column of shapes), and second-order (third column of shapes) gradient analysis results for all 8 hidden units of a 9×8×1 neural network edge detector.

4 Conclusion

A limited number of noisy edge patterns, is proposed to analyze the capability of BP neural networks edge detection in noisy edge patterns. The experimental results are applied to five different sizes of neural networks (at hidden units 4 to 8). The analyses of neural network edge detector's hidden units, as templates, were analysed into three gradient components: low pass or averaging, gradient, and second-order gradients. A quantitative comparison of the neural networks is enabled by the use of 1,312 noisy test patterns, edge and non-edge patterns to which Gaussian noise are added. The increase of hidden units could support the NNets to robust noise and hence improve the recognition accuracy as SNR's increase. Experimental results of NNets have shown that domain function analysis method supports and confirms the results of pattern recognition accuracy. Further work, using this analysis method, can be reached by analyzing the behavior of different neural network architectures; e.g. RBF (Radial Basis Function).

References:

[1] van der Zwaag BJ, Spaanenburg L, et al., "Analysis of neural networks in terms of domain functions," in Proceedings IEEE Benelux Signal Processing Symposium SPS-2002, (Leuven, Belgium, 21-22 March), March 2002, pp. 237-240.

[2] van der Zwaag BJ and Slump K, "Analysis of neural networks for edge detection," in Proceedings of the ProRISC Workshop on Circuits, Systems and Signal processing, Veldhoven, the Netherlands, 2002, pp. 580-586.

[3] van der Zwang BJ and Slump K, "Analysis of neural networks for edge detection," in Proceedings of the ProRISC Workshop on Circuits, Systems and Signal processing, Veldhoven, the Netherlands, 2002, pp. 580-586.

[4] Craven MW and Shavlik JW, "Using sampling and queries to extract rules from trained neural networks," in Machine Learning: Proceedings of the Eleventh International Conference, San Franscisco, CA, 1994 .

[5] Hashem S, "Sensitivity analysis for feedforward artificial neural networks with differentiable activation functions," in Proceedings of the 1992 International Joint Conference on Neural Networks, IEEE Press, Piscataway, NJ, USA, 1992, vol. 1, pp. 419-424.

[6] Siuzdak J, "A single filter for edge detection," Pattern Recognition, vol. 31, No. 11, pp. 1681-1686, 1998.

[7] Rajab MI, "Neural Network Edge Detection and Skin Lesions Image Segmentation Methods: Analysis and Evaluation," PhD, School of Electrical and Electronic Engineering, The University of Nottingham, Nottingham, 2003.

[8] Ilardi P, "Neural Network Models of Sound Localization," BA thesis, Boston University, 1999.

[9] Haykin S, Neural Networks A Comprehensive Foundation. USA: Prentice Hall, 1999.