Application of Neural Networks Analysis in Image Feature Extraction

MAHER I. RAJAB
Computer Engineering Department
University of Umm Al-Qura
P.O. Box 5555, Mecca, KSA
KINGDOM OF SAUDI ARABIA

Abstract: - Analysing neural network edge detection (NNED) is being presented in a new method in order to gain a close insight into their internal functionality. A training sets consisting of a limited number of prototype edge patterns is proposed in order to analyze the problem of edge detection. The generated set of vectors simply corresponds to standard situations that are clearly understood as edges or non-edges and also permit a controlled distribution of the edges enclosed. The behavior of neural network edge detector's hidden units, as templates, were analyzed into three gradient component: low pass, gradient, and second-order gradients. Although the above only gives some analysis results for the units in the hidden units, it should be clear that a characterization of the neural network as a whole could also be derived from these results. Visual comparisons are conducted here, by image convolution of the weights templates with a real test image to demonstrate the feature extracted by each hidden node of NNED. Different image features extracted by image convolution could provide a simple verification of the investigated gradient analysis method.

Keywords: - Neural networks analysis, domain base function, edge detection, pattern recognition, feature extraction.

1 Introduction

Researchers have tried to analysis the internal functionality of neural networks. This analysis was to gain a clear understanding of the internal behavior of different neural networks systems for different applications [1-3]. A neural network analysis method was introduced and presented in earlier publications by van der Zwang [2,3]. 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. In a recent publication by van der Zwang [1], the case of image edge detection was investigated and in particular the 2-dimensional differential operators (base functions). It was also found that it is feasible to analyse 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 [1]. Zwang et al. analysed a neural network edge detector; the neural networks were of the feed-forward error-backpropagation type. 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 analyse the order of this template according to equation (1) in [1].

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

From Equation (1) we can realise 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. 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 Zwang et al. method while analyzing various neural network architectures.* Moreover, various neural networks that have been analysed only consist of one single output, whereas we have analysed similar neural network architectures but with multi-outputs.

This paper is organized as follows. Section 2 starts with an introduction of the method used to generate different noisy edge patterns. The images that are formed by these patterns and also other adopted tools are described to achieve the aims of this work. Section 3 summarizes and discusses the results found when evaluating the success rate of the investigated neural network edge detector. Finally, in Section 4 conclusions are considered.

2 Method

In our case, the noise is chosen to be additive [4] to a prototype edge patterns. Therefore, a generated noise image $n(i, j)$ added to the true image, free-of-noise, $I(i, j)$ will yield $I_n(i, j)$ as

$$I_n(i, j) = I(i, j) + n(i, j)$$

The model of Gaussian noise deviates [5] is chosen because: (i) One can define the Gaussian noise values as normally distributed deviates with zero mean and at fixed standard deviations, σ_n as $n(i, j; \sigma_n)$. These noise values are completely uncorrelated of each other and also, of the true image e.g. $I(i, j)$. Noise is considered to be additive; *additive noise*. (ii) The symmetrical nature of this distribution is a practical approximation. In contrast impulsive noise, randomly alters the pixels so that their values deviate considerably from the true values $I(i, j)$. Two Gaussian deviates are generated by the algorithm introduced by Press et al. [5]. This algorithm has been developed to generate these normal deviates at the desired σ_n and in two dimensions i and j to yield finally a noise image, file, of size *nxm*.

In the present work, the training set consists primarily of prototype edge patterns. Figure 1 shows sixteen edge patterns as a base for the neural network training set. It should be noticed that these patterns are subject to further tuning in the proceeding work until successful training is achieved. The work presented here was based on a neural network model of three layer multi-layer perceptron architecture. An example of 9-9-7 multi-layer perceptron MLP is shown in Figure2.

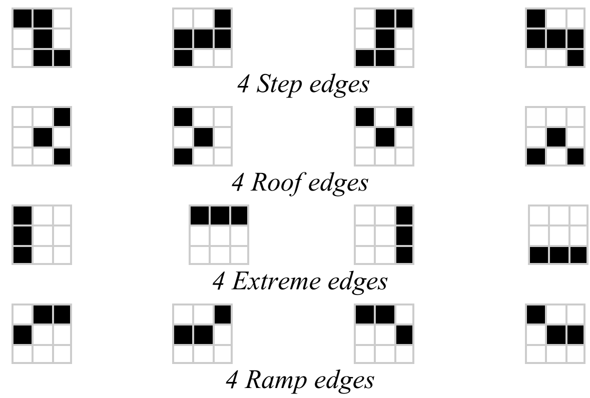


Fig.1. An example of sixteen different prototype edge patterns in a 3x3 pixels grid representing four edge profiles.

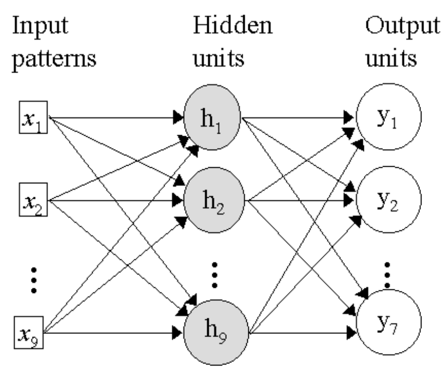


Fig.2. 9-9-7 multi-layer perceptron.

It should be noted that total size is calculated according to the following equations:

$$Total\ noisy\ patterns = P \times R \times N \quad (2)$$

$$Total\ size = P \times R \times N \times W \quad (3)$$

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. Rotation is also considered when generation edge patterns. 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 [6].

3 Results and Discussions

Scion image software [7] is used as a development environment to generate prototype edge patterns [8]. After a successful training of the neural network with various edge patterns, a testing phase is carried out and a decoding procedure is required to decode the winner edges from the neural network outputs. An output code which matches the following criterion, is considered as a winner edge [8]:

$$\textbf{\textit{If}}\ Output \in \{EdgePttern\,Codes\}$$
$$\quad winner\ edge = 1$$
$$\textbf{\textit{else}}$$
$$\quad winner\ edge = 0$$

The above described method (section 2) was used to generate a preliminary training set consisting of 1,312 edge patterns which are similar to the roof edge patterns shown in Figure 1. These patterns are classified into two categories: edges and non-edges. The training set mainly consists of 8 edge patterns or $P = 8$; the original roof edge pattern 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. As an attempt to increase the variability in the training phase, we have considered randomly reordering the sequence of the generated edge patterns.

The recognition accuracy of five NNets with 4,5,6,7, and 8 hidden nodes, and one output in the output layer, have been trained with ten noisy training sets (consisting of the above 1,312 roof patterns). These sets are additive with Gaussian noise 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 decreases for all NNets. The recognition accuracy of the five NNets is demonstrated in Figure 3. Therefore, the increase of hidden units could support the robustness of NNets against noise and hence improve the recognition accuracy as SNR's increase.

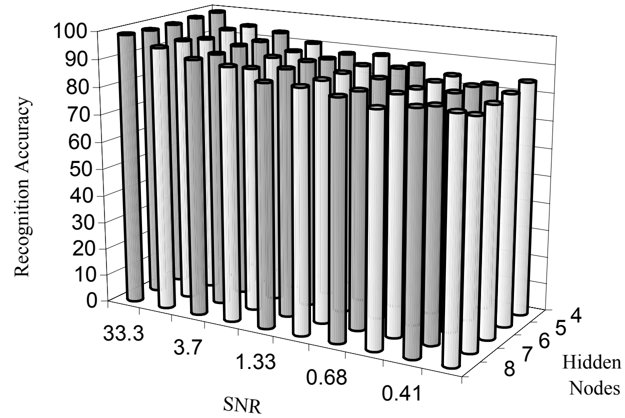


Fig.3. Recognition accuracy via SNR's for five neural network.

In order to test our analysis method, we trained several NNs for edge detection. The NNs are of feed-forward error-back propagation type, with 3x3 inputs, 4 to 20 units in the single hidden layer, and 7 outputs. All units used sigmoid activation functions. Some networks are trained with training sets containing sharp edges only (noise free edge patterns), others are trained with training sets that contained sharp edges as well as additional Gaussian noise. In these experiments three SNR are chosen: (i) no noise, (ii) SNR=12, and (iii) SNR=6. This reduction in noise levels makes the neural network capable of detecting edges and also noise patterns. However, it is difficult to arbitrate between the behaviors of any two successful NNED with similar recognition accuracy. Also, similar classes of synthetic images are needed to arbitrate between the most successful NNED. The use of real images in the arbitration process is not an easy task because of two main reasons: (i) the real life images possess diverse of features, (ii) each of the tested NNED could be able to detect a cluster of these features. The full training set consisting of the above four edge patterns categories is used in the training process of the neural network, (Figure 1). Experiments are applied using the neural network model trained with different sizes of training sets. It is also seen that the training sets that are free of noise make the NNED converge with the lowest error. As an attempt to support the arbitration process between the successful NNED's, the analysis method introduced by van der Zwaag et al. [1] is adopted. The weights between a hidden units and the output unit represents the importance of the hidden unit's edge detection outcome.

The internal behavior of the neural network outputs with 9 hidden nodes and 7 outputs will be analysed here. The noise-free training set is used to train the NNED. Each of the seven templates of size 3×3 will be convolved with a test image to demonstrate the general behavior of each hidden node. Figure 4 shows an example outputs produced from seven templates. It is clearly seen from this figure that the NNED has low-pass or averaging behavior (nodes 1 and 2), which makes the network less sensitive to noise and improves its edge detection ability. We also produce for outputs 3 to 7, the corresponding example of edge strength after thresholding each output at a maximum threshold. These can also check how much details are still contained in these outputs. Low-pass or averaging behaviour makes the network less sensitive to noise and improves the edge detection ability of the second-order network. The weights analysis of the seven NNED outputs is shown in Table 1. These results have a good match with the results found by template convolution in Figure 4. The first two outputs has the lowest first-order behavior that will agree to visible and manifest zero-order behaviors. NNED Outputs 3 to 7, Table 1, show different edge detection behaviors of first and second order gradients. These behaviors are manifested by the amount of image features shown in Figure 4.

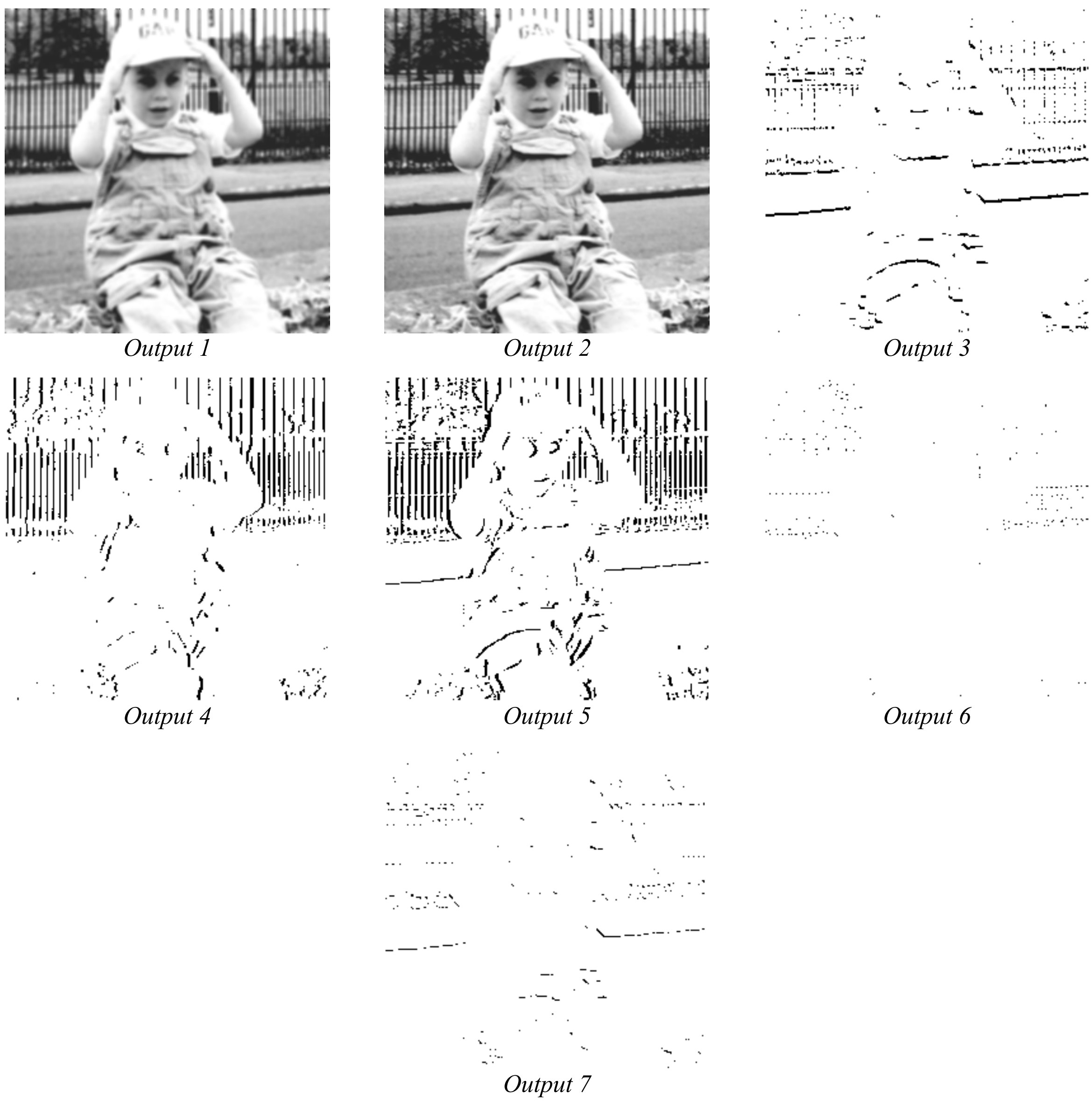
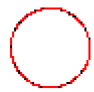
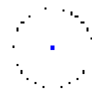
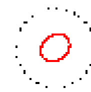
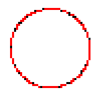
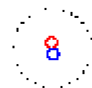
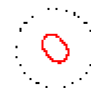
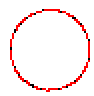
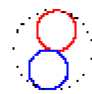
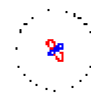
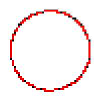
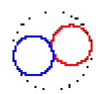
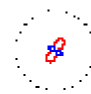
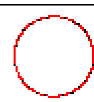
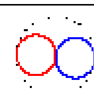
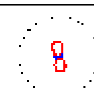
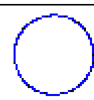
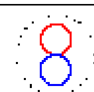
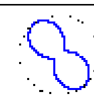
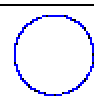
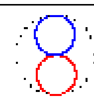
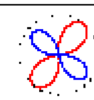


Fig.4. NNED outputs with 9 hidden nodes. Outputs number 3 to 7 are thresholded to verify sharp edges. Outputs number 1 and 2 show smooth images of "Ammar".

Table 1: Low pass, gradient, and second-order gradient analysis results for all 7 hidden nodes of a 9x16x7 neural network edge detector.

NNED output No.	Zero order	First order	Second order
1:			
2:			
3:			
4:			
5:			
6:			
7:			

4 Conclusions

The work presented in this paper proposes the analysis of different neural network edge detection (NNED). Small training sets consisting of limited number of prototype edge patterns are used to analyse BP NNED. These generated set of vectors simply correspond to standard situations that are clearly understood as edges or non-edges and also permit a controlled distribution of the edges enclosed. A variety of NNED's consisting of 4 to 20 hidden units are analysed. Hidden units, whose weights are regarded as templates, were analyzed into three gradient component: low pass or averaging, gradient, and second-order gradients. Although the above only gives some analysis results for the units in the hidden units, it should be clear that a characterization of the neural network as a whole could also be derived from these results. We have analysed the weights between the hidden units and the output units which may suggest an importance of the hidden unit's edge detection outcome. The template analysis of the NNED with the larger number of hidden nodes shows more variety in their behaviours. Visual comparisons are conducted here, by image convolution of the weights templates with a real test image to demonstrate the feature extracted by each hidden node of NNED. Different image features extracted by image convolution could provide a simple verification of van der Zwang gradient analysis method of NNED hidden units.

References

[1] B.J. van der Zwang, K. Slump, Analysis of neural networks for edge detection, in: Proceedings of the ProRISC Workshop on Circuits, Systems and Signal processing (Veldhoven, the Netherlands, 2002) 580-586.

[2] B.J. van der Zwaag, C. Slump, et al., Process identification through modular neural networks and rule extraction, in: D. Ruan, P. D'hondt, E.E. Kerre (Eds.), Computational Intelligent Systems for Applied Research: Proceedings of the 5th International FLINS Conference (World Scientific, Singapore, Ghent, Belgium, 16-18 Sept., 2002) pp. 268-277.

[3] B.J. van der Zwang, L. Spaanenburg, 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, 2002) 237-240.

[4] M. Garcya-Silvente, J.A. Garcya, et al., A new edge detector integrating scale-spectrum information, Image and Vision Computing 15 (1997) 913-923.

[5] W.H. Press, B.P. Flannery, et al., Numerical Recipes in Pascal (Cambridge University Press, 1989).

[6] P. Ilardi, Neural Network Models of Sound Localization (Boston University, 1999).

[7] W. Rasband, Scion image for windows, in: (Scion Corp., USA, 2000).http://www.scioncorp.com. wayne@codon.nih.gov

[8] M.I. Rajab, M.S. Woolfson, Analysis of neural network edge detection, in: V.M. Mladenov (Ed.), WSEAS Transactions on Systems; 4th WSEAS International Conference on Neural Networks and Applications, Vol. 2 Issue(3), (WSEAS, Vouliagumeni,Athens, Greece, July 2003) 649-654.