Proc. 3rd IEEE Benelux Signal Processing Symposium (SPS-2002), Leuven, Belgium, March 21–22, 2002

ANALYSIS OF NEURAL NETWORKS IN TERMS OF DOMAIN FUNCTIONS

Berend Jan van der Zwaag, Cees Slump

Signals & Systems, Dept. of Electrical Engineering, University of Twente
P.O.Box 217, 7500 AE Enschede
Netherlands
{b.j.vanderzwaag,c.h.slump}@el.utwente.nl

Lambert Spaanenburg

Rijksuniversiteit Groningen, Dept. of Mathematics and Computing Science, IWI
P.O.Box 800, 9700 AV Groningen
Netherlands
l.spaanenburg@cs.rug.nl

ABSTRACT

Despite their success-story, artificial neural networks have one major disadvantage compared to other techniques: the inability to explain comprehensively how a trained neural network reaches its output; neural networks are not only (incorrectly) seen as a "magic tool" but possibly even more as a mysterious "black box."
Although much research has already been done to "open the box," there is a notable hiatus in known publications on analysis of neural networks. So far, mainly sensitivity analysis and rule extraction methods have been used to analyze neural networks. However, these can only be applied in a limited subset of the problem domains where neural network solutions are encountered.
In this paper we propose a wider applicable method which, for a given problem domain, involves identifying basic functions with which users in that domain are already familiar, and describing trained neural networks, or parts thereof, in terms of those basic functions. This will provide a comprehensible description of the neural network's function and, depending on the chosen base functions, it may also provide an insight into the neural network's inner "reasoning." It could further be used to optimize neural network systems. An analysis in terms of base functions may even make clear how to (re)construct a superior system using those base functions, thus using the neural network as a construction advisor.

1. INTRODUCTION

In the past 15 years artificial neural networks have gained renewed popularity as "universal problem solvers." Thousands of articles are published on the subject every year. However, one of the strongest weaknesses of neural networks is their inexplicability. This is an important aspect of the functionality of any technology, as users will be interested in "how it works," before trusting it completely. In particular, in safety critical systems (e.g., airlines, power stations, hospitals) it is imperative to know the behavior of an application under all possible input conditions. How can users trust a machine if they do not know how it works or what its behavior will be under extreme circumstances? Neural networks learn from examples, and examples of extreme circumstances are rare by their very nature.
Since the early development of artificial neural networks, but especially in the past 10 years, researchers have tried to analyze them to provide insight into their behavior. For certain applications and in certain problem domains this has been successful. In particular in decision making systems and other systems that can easily be expressed in sets of rules, great advances have been made by the development of so-called rule extraction methods [3]. Neural network systems with relatively few inputs can sometimes be analyzed by means of a sensitivity analysis [8], which is a non-parametric statistical analysis technique.
However, most neural network systems are so high-dimensional that an extracted rule base would become too large to be easily interpreted, or so nonlinear that a sensitivity analysis would only be valid for a small part of the input space. For this reason, we propose domain-specific neural network analysis methods that utilize domain-specific base functions that are easy to interpret by the user and that can even be used to optimize neural network systems. An analysis in terms of base functions may also make clear how to (re)construct a superior system using those base functions, thus using the neural network as a construction advisor.

2. PROBLEM STATEMENT

Despite their success-story, neural networks have one major disadvantage compared to other techniques: the inability to explain comprehensively how a trained neural network reaches its output; neural networks are not only (incorrectly) seen as a "magic tool" but possibly even more as a mysterious "black box" [13]. This is an important aspect of the functionality of any technology, as users will be interested in "how it works" before trusting it completely. In particular, in safety critical systems (e.g., airlines, power stations, hospitals) [5] [12] it is imperative to know the behavior of an application under all possible input conditions.

3. ANALYSIS OF NEURAL NETWORKS

The merits of the analysis of neural networks include: [1]

- provision of a "user explanation" capability
- extension of neural network systems to "safety-critical" application domains
- software verification and debugging of neural network components in software systems
- improving the generalization of neural network solutions
- data exploration and the induction of scientific theories
- knowledge acquisition for symbolic AI systems

These merits clearly indicate how users can benefit from the analysis of neural networks. They will see improved functionality, and using the results of the analysis they can determine which type of neural network is best suited for solving their particular problems.

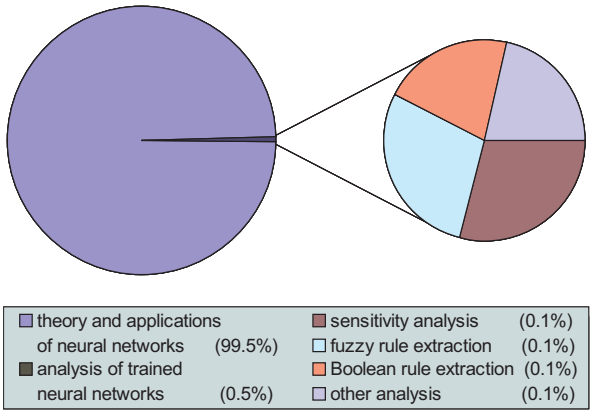


Figure 1: *Results of a search in several major patent databases, illustrating the relatively small amount of literature on the analysis of neural networks.*

Compared to the amount of literature on theory and applications of neural networks, information on the analysis of neural networks is scarce (see Figure 1), and mostly limited to either *sensitivity analysis* or *rule extraction*. Sensitivity analysis is a technique from the field of nonparametric statistical analysis, whereas rule extraction originates from the field of symbol processing methods, which is rule-based rather than case-based. In the following subsections these methods will be treated in more detail.

A few examples that cannot directly be placed under sensitivity analysis or rule extraction are found in [6], [10], [15], and [16]. Feng [6] uses multilayer perceptrons for parameter estimation and also takes a first step to analyze the hidden units. Worth and Spencer [15] describe a neural network for tactile sensing. After learning, the weight vectors of all hidden units have the same structure, but the authors did not find any correlation between the relative sizes of the weight vectors and the outputs of the hidden units. Mitchell [10] presents a method for interpreting the role of the hidden neurons geometrically, when using neural networks for function approximation problems, in terms of regions of the input space. The relationship between this interpretation and the weights of the network connections is highlighted. For the case in which the approximation is of most interest over a bounded region of the input space, the author shows that this interpretation may be used to check for redundancy among hidden units, and to remove any such units found.

In general, neural networks with unsupervised training merely reorganize the input space, so analyzing them after training becomes fairly simple: an investigation into the reorganized input space reveals how the network has restructured the input space.

Analyzing neural networks trained under supervision is far more complicated, for input and output spaces are usually in different domains (e.g., a character recognition system has an image as input, and a character as output), whereas in the unsupervised case, input and output spaces are basically the same, although they are organized in different ways.

3.1. Sensitivity Analysis

Sensitivity analysis (see, e.g., [11]) is a technique from the field of nonparametric statistical analysis. It is occasionally applied to neural networks, but more often in other mathematical modeling and decision-making systems. The general idea of sensitivity analysis is to investigate the effect that perturbations in the inputs have on the outputs, thus determining the sensitivity of the outputs to the inputs [9]. This is usually done for every input and every output, resulting in a matrix of sensitivities. These can then be used to determine whether any insignificant inputs can be ignored. In the neural network case, if a sensitivity analysis is performed for the hidden neurons, it could be used for pruning if some hidden neurons appear to have no influence on the network outputs [4]. Fu and Chen [7] use sensitivity analysis to investigate the generalization capability and error-correcting property of a neural network. Other examples of the use of sensitivity analysis for neural networks can be found in [2] and [8].

A weakness of sensitivity analysis of neural networks is the fact that many applications of neural networks operate in high-dimensional input spaces, which would result in large sensitivity matrices that are hard to interpret. Another drawback is caused by the often strong nonlinearity of neural networks. This can cause outputs to have many different sensitivities over the full range of particular inputs. Different analysis tools are needed in order to be able to analyze neural networks with high-dimensional input or output spaces or with strong nonlinear behavior. In fact, most neural networks have one or both of these properties, as they form some of the strengths of neural networks. The cases in which sensitivity analysis could be applied to neural networks are mainly those cases where the networks are small, those where sensitivity analysis is only applied to a (small) part of the neural network, and those where the dimensionality of the network has been greatly reduced. The latter case could, for example, be realized with modular neural networks such as in [14].

3.2. Rule Extraction

Rule extraction from neural networks (see, e.g., [1]) originates from the field of symbol processing methods, which is rule-based rather than case-based. Neural network behavior described in sets of rules can provide an insight into how the neural network comes to an answer. Craven and Shavlik [3] distinguish two approaches to testing rules for multilayer neural networks: the decompositional approach and the pedagogical approach. The decompositional approach is to extract rules for each hidden and output unit separately, thus providing a certain transparency, whereas the pedagogical approach enables the extraction of rules that directly map inputs to outputs for a network as a whole, thus basically not opening the “black box.” Pedagogical techniques are typically used in conjunction with a symbolic learning algorithm and the basic motif is to use the trained neural network to generate examples for the learning algorithm.

Andrews et al. [1] classify rule extraction techniques into the following categories:

- Boolean rule extraction using decompositional approaches;
- Boolean rule extraction using pedagogical approaches;
- Extraction of fuzzy rules.

Due to its nature, Boolean rule extraction is mainly used in problems with discrete-valued features. Fuzzy rule extraction can be applied to both problems with discrete-valued features and those with real-valued features. However, there are problem domains where solutions cannot easily or comprehensively be described in sets of rules or decision trees, due to, e.g., high dimensionality of

the input space or very large sets of independent features. In those domains, different tools for analysis are needed, either in conjunction with rule extraction or separately.

4. PROPOSED METHOD: ANALYSIS IN TERMS OF DOMAIN-SPECIFIC BASE FUNCTIONS

There is a hiatus in known publications on analysis of neural networks. So far, mainly sensitivity analysis and rule extraction methods have been used to analyze neural networks. However, the previous sections make clear that these can only be applied in a limited subset of the problem domains where neural network solutions are encountered.

We need to investigate in which problem domains trained neural networks cannot satisfactorily be analyzed with sensitivity analysis or rule extraction techniques, and we need to develop new methods to analyze trained neural networks in those domains.

We therefore propose a method which, for a given problem domain, involves identifying basic functions with which users in that domain are already familiar, and describing trained neural networks, or parts thereof, in terms of those basic functions. This will provide a comprehensible description of the neural network's function and, depending on the chosen basic functions, it may also provide an insight into the network's inner "reasoning."

Domain-specific analysis of neural networks through base functions will not only provide insight into the in- and external behavior of neural networks and show their possible limitations in particular applications, but it will also lower the acceptability threshold for future users unfamiliar with neural networks. Further, domain-specific neural network analysis methods that utilize domain-specific base functions can also be used to optimize neural network systems. An analysis in terms of base functions may even make clear how to (re)construct a superior system using those base functions, thus using the neural network merely as a construction advisor. If a user does not want to trust a neural network for any reason whatsoever, he may still trust a non-neural system that would have been nearly impossible to construct without using a neural network as an advisor.

For many problems in certain domains, such as linguistics and decision theory, the common, domain-dependent base functions could be chosen to be *if–then* rules or decision trees, in which case the analysis reduces to rule extraction. For the image processing domain, an example of neural network analysis using base functions is given in the following subsection. Table 1 lists a few problem domains where neural networks have been successfully applied. For each of these domains possible base functions are presented. Investigating the applicability of these base functions for the purpose of neural network analysis is part of the proposed project.

Table 1: *Some application domains with potential domain-specific base functions.*

application domain	potential base functions
signal processing (1-D)	basic operational filters
digital image processing (2-D)	differential operators
general classification problems	feature map regions (compare Kohonen's self-organizing feature map)
decision theory	if-then rules (fuzzy or not) (i.e., rule extraction as a special case of the proposed method)
control theory	basic control operators

5. CASE STUDY

As an example, we can show that an edge detector realized by a neural network can be analyzed in terms of differential filter operators, which are common in the digital image processing domain. Digital image processing filters can be written as two-dimensional vectors, so, if the weights of the connections into a neuron can be written as a two-dimensional vector as well, then this neuron can be considered a particular type of image filter, provided a suitable transfer function is used. Because the working of a filter is difficult to comprehend given a vector of numbers, the vectors are transformed to a description in terms of the filter's differential components, which can then be presented graphically.

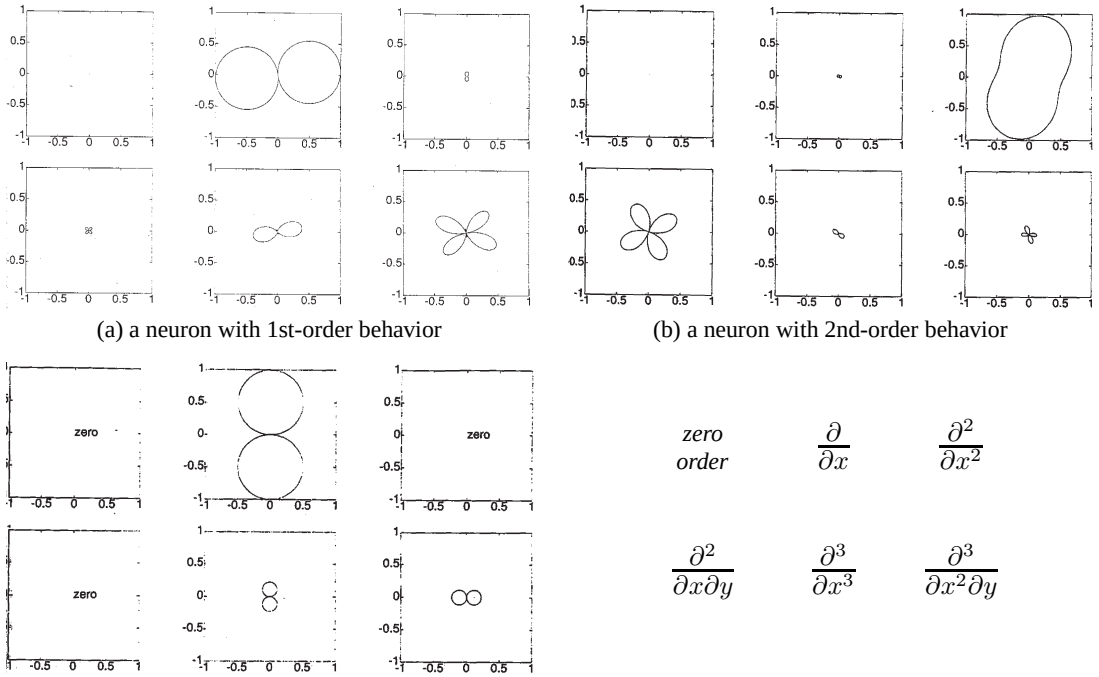
To this extent, the filter's two-dimensional feature vector is first Fourier-transformed. The transformed description consists of a series of sinusoidals, which are easily differentiated to get the Taylor series components. This results in a component-wise description of the filter in the frequency domain. A back Fourier transformation brings us back into the spatial domain yielding an expansion into gradients of different orders. As the filters are often directional, i.e., under some angles their operation is stronger than under others, the Cartesian coordinates are transformed to polar coordinates before the Fourier transformation. This not only provides which orders of differential operators the filter consists of, but also in which directions these operators work optimally.

We then have a description of the edge detection filter unit as a collection of differential operator factors $\beta_{\theta ij}$, where i and j are differential orders in two dimensions, and θ the angle of rotation. Graphically, we can show in which directions these operators work optimally, i.e., for which angles θ $\beta_{\theta ij}$ is maximal, for a given i and j. The differential operators are shown as graphs in which the absolute value of $\beta_{\theta ij}$ drawn as function of θ is represented by the distance from the center of the graph in the direction of θ. See Figure 2 for some examples.

The same method was applied to some well-known image filters, enabling comparison of conventional edge detectors known from literature and the neural network edge detectors. Some partial results are shown in Figure 2. The difference between this comparison and more commonly used methods for comparison lies herein that this comparison was based directly on the detectors' filter operations rather than on their performance on a given (benchmark) example. The latter is a more indirect method of comparison and does not provide any insight into the neural network's functionality.

6. REFERENCES

[1] R. Andrews, J. Diederich, and A.B. Tickle, "A survey and critique of techniques for extracting rules from trained artificial neural networks," Neurocomputing Research Centre report, Queensland University of Technology, Australia, 1995.



(a) a neuron with 1st-order behavior

(b) a neuron with 2nd-order behavior

(c) a differential-of-Gauss filter

(d) the differential orders shown

Figure 2: *Differential operator behavior of two hidden neurons from two different neural networks trained for edge detection, and of a differential-of-Gauss filter, all taking inputs from a 5×5 window. (From [16], with permission.)*

[2] J.Y. Choi and C.-H. Choi, "Sensitivity analysis of multilayer perceptron with differential activation functions," *IEEE Tr. on Neural Networks*, vol. 3, no. 1, pp. 101-107, 1992.

[3] M.W. Craven and J.W. Shavlik, "Using sampling and queries to extract rules from trained neural networks," in *Machine Learning: Proceedings of the Eleventh International Conference*, San Francisco, CA, 1994.

[4] A.P. Engelbrecht and I. Cloete, "A sensitivity analysis algorithm for pruning feedforward neural networks," in *IEEE Int. Conf. on Neural Networks*, Washington, vol. 2, pp. 1274-1277, 1996.

[5] ERA Technology Ltd., "Neural Networks Producing Dependable Systems," ERA Report 97-0365, Leatherhead, April 1997. Http://www.era.co.uk/techserv/pubs/p970365.htm, 2 May 2001.

[6] T.-J. Feng, "Backpropagation networks for parameter estimation (an overall report)," University of Twente, the Netherlands, August 1991.

[7] L. Fu and T. Chen, "Sensitivity analysis for input vector in multilayer feedforward neural networks," in *Proceedings IEEE Int. Conf. on Neural Networks*, San Francisco, CA, vol. I, pp. 215-218, 1993.

[8] S. Hashem, "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, vol. 1, pp. 419-424, 1992.

[9] C.C. Klimasauskas, "Neural nets tell why," *Dr. Dobbs's Journal*, pp. 16-24, April 1991.

[10] J. Mitchell, "A geometric interpretation of hidden layer units in feedforward neural networks," *Network*, vol. 3, pp. 19-25, 1992.

[11] D. Rios Insua, *Sensitivity Analysis in Multi-Objective Decision Making*, Lecture Notes in Economics and Mathematical Systems No. 347, Springer Verlag, Berlin, 1990.

[12] A.J.C. Sharkey, N.E. Sharkey, and O.C. Gopinath, "Diversity, neural nets and safety critical applications," in L.F. Niklasson and M.B. Boden (eds.), *Current Trends in Connectionism*, Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 165-178, 1995.

[13] J. Sjoberg, Q. Zhang, L. Ljung, A. Benveniste, B. Deylon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsky, "Nonlinear black-box modeling in system identification: a unified overview," *Automatica*, vol. 31, no. 12, pp. 1691-1724, 1995.

[14] L. Spaanenburg, C. Slump, R. Venema, and B.J. van der Zwaag, "Preparing for knowledge extraction in modular neural networks," in *Proc. 3rd IEEE Benelux Signal Processing Symposium (SPS-2002)*, Leuven, Belgium, March 21-22, 2002.

[15] A.J. Worth and R.R. Spencer, "A neural network for tactile sensing: the Hertzian contact problem," in *International Joint Conference on Neural Networks*, part I, pp. 267-274, 1989.

[16] B.J. van der Zwaag, "Analysis of neural network edge detectors," report no. 92M185, University of Twente, the Netherlands, 1992.