

Neural Networks and Structured Knowledge: Rule Extraction and Applications

FRANZ J. KURFESS

Department of Computer Science, Concordia University, Montreal, Quebec H3G 1M8, Canada

Franz.Kurfess@computer.org

Abstract. As the second part of a special issue on “Neural Networks and Structured Knowledge,” the contributions collected here concentrate on the extraction of knowledge, particularly in the form of rules, from neural networks, and on applications relying on the representation and processing of structured knowledge by neural networks. The transformation of the low-level internal representation in a neural network into higher-level knowledge or information that can be interpreted more easily by humans and integrated with symbol-oriented mechanisms is the subject of the first group of papers. The second group of papers uses specific applications as starting point, and describes approaches based on neural networks for the knowledge representation required to solve crucial tasks in the respective application.

Keywords: neural networks, rule extraction, knowledge representation, structured knowledge, connectionism, hybrid systems

1. Introduction

Over the last few decades, neural networks have become acceptable tools for solving a large variety of tasks, frequently as an alternative to conventional statistical techniques. The ability to train a network from a set of example data, and then use the trained network to generalize new data frequently lies at the core of such tasks. For some applications, however, the fact that neural networks are capable of solving a particular task may not be sufficient: A formal verification or validation of the way the task is solved can be necessary in order to show with high confidence that the task can be solved in a satisfactory manner. Formally proving some properties of the algorithm that underlies the actual implementation of a neural network does not really accomplish that goal, since the successful use of a neural network for a particular task depends to a large degree on the selection of the sample data, the presentation of the data during the learning process, and other parameters usually determined by the network

designer or end user. Thus, neural networks are frequently treated as “black boxes” where the actual inner workings are not visible, or do not exhibit useful information when inspected. Merely exposing the inner workings of a network is only the first step, however. For some applications, what we want is a useful representation of the relationships between the items used as input and output for the neural network. Techniques have been developed to extract such relationships and convert them into a higher-level representation of the knowledge contained in a neural network [2–6]. Such a representation should be reasonably understandable by humans, and be formally treated by expert systems or inference engines. This formal treatment can include the proof of certain properties of the rules, or the validation through comparison with other collections of rules (e.g. those derived from a domain expert’s knowledge). One of the most frequently used forms of representing knowledge extracted from neural networks are if-then rules, which are also a common representation mechanism for expert systems.

The approaches for knowledge extraction start from the information contained in a neural network, and concentrate on the transformation of that information into a different representational format, often a rule-based one. Reversing the direction of the transfer leads to rule refinement: An existing set of rules is converted into a neural network, which then is trained with sample data. The goal is the utilization of existing knowledge, and the adaptation of sets of rule to actual data. The availability of prior information can also be used to improve the training process of neural networks. Since the starting configuration of the network is not randomly chosen, but often reflects important aspects of the learning task, the time to train the network may be substantially reduced, or the resulting network may offer better generalization capabilities.

A good overview of techniques used for knowledge initialization, rule extraction, and rule refinement is given in [7].

Neural networks are frequently applied to tasks where the information to be processed consists of sets or sequences of sample data, typically vectors. Thus the internal structure of the items to be processed is very rigid, and only implicitly relevant for the processing tasks. Systems dealing with knowledge, on the other hand, explicitly utilize the interconnections between and within items to be processed. This is usually a more cumbersome task, and most types of neural networks are not very well suited for it. The first part of this special issue [1] deals with some of the fundamental questions and techniques for the representation and processing of structured knowledge with neural networks. This part contains some articles that use neural networks in applications dealing with structured knowledge. One specific problem which is quite difficult to handle for conventional, symbol-oriented approaches is the approximate matching of graphs [8]. The basic idea is to have a "quick glance" at two graphs, and determine if they are similar. Two problems need to be overcome here: First, the computational complexity of comparing two arbitrary graphs is quite high already; second, an appropriate similarity measure has to be defined. Neural networks can be used for the quick comparison of two items, e.g. via associative memories. A simple similarity measure is also available in this case, namely the overlap in the features of the two items. Associative memories in their simple form, however, can only deal with vectors, not with graphs.

Recursive auto-associative memories and related approaches are able to represent graphs, and they are used by Anna Maria Bianucci, Alessio Micheli, Alessandro

Sperduti and Antonina Starita to analyse the structural properties of chemical compounds. Another approach to deal with the complex structure of knowledge is to utilize neural networks with a more complex internal structure, such as the dynamic tree-structured networks used by N. Davey, R.G. Adams and S.J. George for hierarchical classification of data sets. A third approach finally is to use different types of neural networks in collaboration, possibly together with other, symbol-oriented components, to tackle the structural complexity in the task to be processed. An example for this is the contribution of M. Pfister, S. Behnke, and R. Rojas on handwritten ZIP code recognition in a letter sorting system.

The remainder of this introduction to the second part of the special issue on neural networks and structured knowledge contains brief overviews of the individual contributions.

2. Contributions to this Part of the Special Issue

2.1. *FERNN: An Algorithm for Fast Extraction of Rules from Neural Networks*

Many applications of neural networks require the mapping of input patterns onto output patterns, based on sets of training examples. A popular type of neural networks for such tasks are feed-forward networks with one or more hidden layers. In such networks, the information for the mapping function is represented in the weights affiliated with the connections between the nodes of the network. For humans, it is practically impossible to understand the functioning of such a network on the basis of the information represented by these weights: Frequently, there are hundreds or even thousands of weights to be considered, they might show mutual influences, and in addition other aspects like the activation function of the nodes must be taken into account. The mapping of input patterns to output patterns can also be achieved by rules describing relevant properties of patterns. Such a set of rules is much easier to understand for humans, and as a consequence, rule extraction techniques for converting the internal representation of neural networks into sets of rules have been investigated for quite some time [2, 4]. In their contribution, Rudy Setiono and Wee Kheng Leow describe an algorithm that allows for the fast extraction of rules from feedforward networks with one hidden layer. In contrast to most previous approaches, which rely on the removal of less relevant connections and

units and a subsequent retraining of the network, their algorithm identifies relevant hidden units based on information gains, and removes irrelevant connections to such a hidden unit. After training the network with the goal of minimizing the cross entropy function augmented by a penalty term that helps separating relevant from irrelevant connections, a decision tree is generated, and rules can be derived from the decision tree. Experiments show that the predictive accuracy and the tree size of the algorithm are similar to others that require the computationally expensive network pruning and retraining phases.

2.2. *Knowledge Extraction from Transducer Networks*

The way neural networks perform their tasks is not always easy to comprehend, especially for more complex ones like recurrent networks. From the very beginnings of neural network research, intellectual curiosity as well as the need to improve performance have driven attempts at inspecting the internal status and inner workings of neural networks. Since a lot of small, low-level activities are taking place simultaneously in a neural network, it is quite difficult to identify meaningful representations and interpretations of these activities, related artefacts, or their result in a human-understandable way. The analysis of the different basic entities of a network, the nodes and weights, can be used to identify some low-level phenomena, e.g. grouping of nodes, or weights with particularly strong values. These phenomena, however, are very difficult to interpret, and only show a static glimpse into the network at a given point in time. At a slightly higher level, the values of the activations of nodes can be analysed, e.g. by hierarchically clustering patterns into a tree structure, or through a principal component analysis to identify similar patterns. Although more insight can be gathered at this level, both the representation of more complex knowledge as well as information involving changes over time is not satisfactory. In his contribution, Stefan Wermter explores the formation of categories during the learning phase of a network, and the extraction of knowledge from recurrent neural networks via synchronous sequential machines, or transducers. A transducer takes an input together with the current state of the machine, and calculates the new state and an output. Especially the transducer extraction technique is very helpful for understanding the operation and internal representation of recurrent networks, but also for the integration of neural networks

with symbol-oriented techniques. The different knowledge extraction techniques are illustrated with examples from the linguistic domain.

2.3. *Extracting Phonetic Knowledge from Learning Systems: Perceptrons, Support Vector Machines and Linear Discriminants*

Human languages are a very familiar method of representing knowledge in a structured way: Words are grouped into sentences according to grammatical rules and pragmatic usage patterns, and sentences are put together to form paragraphs or larger entities. Although language by nature exhibits a sequential structure, which is more restrictive than general graphs, for example, it is one of the best knowledge representation mechanisms that we humans have, and in many cases out-shines artificial knowledge representation mechanisms. The use of language in its spoken form also illustrates the extraction of knowledge from an analog signal into discrete, meaningful entities—the words of the language. In their contribution, Robert I. Damber, Steve R. Gunn and Mathew O. Gore examined the capabilities of various types of neural networks to convert continuous, analog sound signals into low-level symbolic labels, the phonemes, which are then grouped together into words. In their work, they model the distinction between “voiced” and “unvoiced” consonants, as evident in human and animal listeners, with different networks. These networks display the same systematic behavior as their natural counterparts, and have the advantage that they can be analysed in detail. The knowledge extracted from these networks indicates that voicing is directly derivable from the intermediate auditory representations, which reflect the physiological aspects of the auditory apparatus, but not from the raw acoustic representation. The use of neural networks for knowledge extraction here not only allowed a much more well-founded analysis that would be impossible in living beings, but also lead to the discovery of new features relevant for speech perception.

2.4. *Unsupervised Extraction of Structural Information from High-Dimensional Visual Data*

The use of vision as an input method for a knowledge processing system, be it human or artificial, presents quite a few challenges: the sheer quantity of information can be immense, and the information and

knowledge contained in visual data is extremely varied, ranging from relatively low-level features like color or boundaries over depth or the shape of objects, to the recognition of complex objects like handwritten characters or fingerprints. One basic requirement for tackling such a task is the identification of relevant features, thus reducing the large amount of raw data into a smaller set of features, which then can be used for further processing. Stephen McGlinchey, Darryl Charles, Pei Ling Lai, and Colin Fyfe describe the use of unsupervised neural networks for the extraction of structural features like the orientation of objects and depth information from visual data. One task is the simultaneous identification of multiple features, for which the authors have developed a considerably simpler network than previously reported. This network consists only of an input and an output layer with a single layer of synaptic weights, and uses sparse coding as an effective means for the identification of features like horizontal and vertical bars in visual data. In another task, the movement of a bar across a field of vision is handled by a network forming a subspace map. In this approach, nodes are grouped into modules, which perform principal component analysis on a subset of the visual data, and extract structural information from two-dimensional visual data. The goal of the third task is to extract shared information between different sensors. A neural network implementation of canonical correlation analysis is used as the basis for a method to predict the position of a moving object, where the position changes over time represent temporal coherence. This extraction of multiple features representing structural information about the visual data set then builds the basis for identifying regularities at higher levels.

2.5. The Architecture and Performance of a Stochastic Competitive Evolutionary Neural Tree Network

A very elementary method of identifying underlying structural information in raw data is to group data sets into clusters, assuming that the elements in such a cluster exhibit some similarities. Many types of neural networks have been proposed and successfully applied for the basic cluster analysis, which assigns elements of the data set to a predefined number of clusters. In many applications, however, it is very difficult to predict the number of clusters. Although approaches have been developed that create new clusters as needed, this is a very challenging task, and computationally quite

expensive. Neil Davey, R.G. Adams and S.J. George describe an approach that combines such a dynamic clustering scheme with a hierarchical one, resulting in a tree structure of clusters that can be expanded as needed: Their model, in contrast to similar ones, requires no initial parameter setting, and consistently produces trees of high quality; it should be noted, however, that the criteria to judge the quality of a hierarchy are somewhat subjective. The model produced a hierarchical structure reflecting the visual differences of images showing various animate and inanimate objects, and grouped the elements of the Zoo data set for machine learning into groups in a quite natural way. Such hierarchical classification models are very useful for the quick analysis of data sets, and for the identification of highly relevant features at various levels of the hierarchy.

2.6. Recognition of Handwritten ZIP Codes in a Real-World Non-Standard-Letter Sorting System

A practical application of the challenges posed by the processing of visual data is the recognition of characters. Especially for the case of handwritten characters, this task is quite difficult to solve, considering the myriads of variations in script, shape, size, orientation, and other features. Over the last few decades, the quality achieved by optical character recognition systems has been quite high, and is used in the automated sorting of letters on a routine basis. M. Pfister, S. Behnke, and R. Rojas report on their work performed by Siemens AG in collaboration with the Free University of Berlin for the German postal service. Their task is to perform the recognition of handwritten ZIP codes on non-standard letters, which is significantly harder than for standard letters. This is due to the larger size and greater liberty in the placement of the address block, resulting in a larger variety of handwriting. In addition, this has to be done in real-time, at a speed of about 6 letters per second. This speed can only be achieved by dedicating a dual Pentium computer with additional neural network and image processing hardware. Before the actual interpretation of the ZIP codes can take place, the location of the address block and then the ZIP code itself have to be identified. This in itself is not an easy task, considering that items like brochures, catalogs, or magazines, which frequently contain display text, graphics, or pictures, are to be sorted as well. The main emphasis of their paper, however, lies on the actual

recognition of the handwritten digits constituting the ZIP code. For this, the underlying task is to extract relevant features from the unstructured pixel patterns, and to classify a shape described by a set of features as a digit. Two different classifiers are used to tackle this task, and their results are combined to increase the accuracy of the overall recognition system. One of them is based on the time-delayed neural network (TDNN) approach [9], and scans the pixel pattern horizontally. The other one extracts structural information, and tries to match the extracted structure with prototypes for the respective digits. To do this, the pixel image is first converted into a line drawing, from which a structural graph is obtained. This structural graph is matched against prototypes, which sometimes already yields a classification result. Otherwise, extracted quantitative features can be used to distinguish between digits with the same structure. Due to the tight time constraints, it is not efficient to run the two classifiers in parallel, and combine their results. Instead, the structural classifier is run first since it is simpler and faster. The more complex TDNN classifier is run only when needed, in about 20% of the cases. The combination of these two classifiers results in recognition rates of up to 99.5%, depending on the admissible rate of substitutions.

2.7. *Application of Cascade Correlation Networks for Structures to QSPR and QSAR*

A natural representation of chemical compounds is as labeled graphs, indicating the relationships between the basic elements that constitute a more complex compound. By their very nature, however, these graph-based representations are high-level abstractions of the actual structure of the compound, and thus can not express many of the interesting properties of the compound. In addition to physical properties, the biological activities of a compound can be of very high interest, e.g. for using the compound as a drug. Skilled experts can predict certain properties or biological activities from the structure of a compound, but this is quite hard to do, and rather expensive. There is reason to believe that the structure of compounds has a close correlation with its activities, and approaches like Quantitative Structure-Activity Relationship (QSAR) have been developed to automatically evaluate compounds. The basic idea is to catalog the structures and activities of known active compounds, and to try to predict the activities of unknown compounds by identifying the most similar known compounds. In many cases, specific

activities actually can be correlated to sub-structures of compounds, and it may be sufficient to perform a partial matching of the structural graphs. Traditional approaches to this problem rely on an encoding of the properties of a compound as a vector, and using this vector for the matching of compounds. The problem here is that the encoding of the properties is very critical, and has to be performed by an expert. It is also frequently not very straightforward, and involves substantial experimentations to find a good encoding. Anna Maria Bianucci, Alessio Micheli, Alessandro Sperduti and Antonina Starita use a generalization of recurrent neural networks for the processing of graphs, which in this case represent the chemical compounds. Such networks transform a labeled graph into a vector of real numbers in such a way that the original graph can be recovered. In essence, the network performs a similar task as the expert: trying to find a simple representation for a complex structure without losing essential properties. The network approach, however, optimizes the prediction task through an automated learning mechanism on the basis of samples with known properties, whereas the expert's optimization relies heavily on trial and error.

3. **Looking Back to the First Part: Knowledge Representation and Reasoning**

The collection of articles in the first part of this special issue [1] of the collection of articles on neural networks and structured knowledge has an emphasis on knowledge representation and reasoning mechanisms based on neural networks. Two articles [10, 11] describe the use of graphs for representing knowledge and for reasoning with neural networks. In two other articles, the relationship between logic programs and neural networks is investigated [12, 13]. The two final articles describe representation and reasoning mechanisms exhibiting interesting similarities to the way reasoning is performed by humans [14, 15].

Acknowledgments

The papers published in these two special issues have been selected from around forty contributions submitted in response to the call for papers. I would like to thank all contributors for their efforts, especially those whose contributions could not be accepted here due to space restrictions. My thanks also go to the more

than 150 referees. Without their help it would have been impossible to put this collection together. Many of them offered valuable suggestions for improving the quality and presentation of the reviewed contributions.

This set of two special issues is an outcome of a number of activities pursued over the last few years. Most directly related is a workshop on "Neural Networks and Structured Knowledge" held during the European Conference on Artificial Intelligence (ECAI '96) in Budapest, Hungary [16]. Similar workshops and symposia took place in combination with other conferences like the International Joint Conference on Artificial Intelligence (IJCAI '95) in Montreal [17], the German Conferences on Artificial Intelligence in Berlin (KI '93) and Saarbrücken (KI '94) [18–20] the Fall Schools on Connectionism and Neural Networks (HekoNN '94 and '95) [21, 22], and the MIX '97 Fall Symposium on Hybrid Systems organized by Wolfgang Ertel and Bertram Fronhöfer. I would like to thank the attendees of these workshops, the authors of papers, and of course the organizers.

Most of my work in this area was performed during my employment at the University of Ulm, in the Neural Information Processing Department directed by Prof. Günther Palm, with partial funding from the German Ministry for Research and Technology. While at New Jersey Institute of Technology, related activities have been funded by the State of New Jersey in the SBR program.

References

1. F.J. Kurfeß, Special issue on "Neural Networks and Structured Knowledge: Representation and Reasoning" (guest editor), *Applied Intelligence*, vol. 11, no. 1, 1999.
2. J. Denker, D. Schwartz, B. Wittner, S. Solla, R. Howard, L. Jacket, and J. Hopfield, "Automatic learning, rule extraction and generalization," *Complex Systems*, vol. 1, no. 5, pp. 877–922, 1987.
3. J.-S. Roger Jang, "Rule extraction using generalized neural networks," in *Proc. of the 4th IFSA World Congress* (in the Volume for Artificial Intelligence), July 1991, pp. 82–86.
4. C. McMillan, M.C. Mozer, and P. Smolensky, "The connectionist science game: Rule extraction and refinement in a neural network," in *Proceedings of the 13th Annual Conference of the Cognitive Science Society*, 1991.
5. R. Setiono and H. Liu, "Understanding neural networks via rule extraction," edited by Chris S. Mellish, in *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, San Mateo, August 20–25, 1995, Morgan Kaufmann, pp. 480–487.
6. R. Andrews and J. Diederich (Eds.), *Rule Extraction Workshop*, Neural Information Processing Systems (NIPS) 9, 1996.
7. R. Andrews, J. Diederich, and A.B. Tickle, "Survey and critique of techniques for extracting rules from trained artificial neural networks," *Knowledge Based Systems*, vol. 8, no. 6, pp. 373–389, December 1995.
8. J. Köbler, U. Schöning, and J. Toran, *The Graph Isomorphism Problem: Its Structural Complexity*, Birkhäuser: Boston, 1993.
9. A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K.J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Transactions on Acoustics, Speech, & Signal Processing*, vol. 37, no. 3, pp. 328–339, 1989.
10. K. Schädler and F. Wysotzki, "Comparing structures using a hopfield-style network," *Applied Intelligence*, vol. 11, no. 1, pp. 15–30, 1999.
11. P. Myllymäki, "Massively parallel probabilistic reasoning with boltzmann machines," *Applied Intelligence*, vol. 11, no. 1, pp. 31–44, 1999.
12. S. Hölldobler, Y. Kalinke, and H.-P. Störr, "Approximating the semantics of logic programs by recurrent neural networks," *Applied Intelligence*, vol. 11, no. 1, pp. 45–58, 1999.
13. A.S. d'Avila Garcez and G. Zaverucha, "The connectionist inductive learning and logic programming system," *Applied Intelligence*, vol. 11, no. 1, pp. 59–78, 1999.
14. L. Shastri, "Advances in SHRUTI—a neurally motivated model of relational knowledge representation and rapid inference using temporal synchrony," *Applied Intelligence*, vol. 11, no. 1, pp. 79–108, 1999.
15. R. Sun, T. Peterson, and E. Merrill, "A hybrid architecture for situated learning of reactive sequential decision making," *Applied Intelligence*, vol. 11, no. 1, pp. 109–127, 1999.
16. F.J. Kurfeß (Ed.), *Neural Networks and Structured Knowledge*, European Coordinating Committee for Artificial Intelligence (ECCAI), European Conference on Artificial Intelligence (ECAI '96), Workshop Proceedings, Budapest, 1996.
17. R. Sun and F. Alexandre (Eds.), *Connectionist-Symbolic Integration*, Lawrence Erlbaum, 1997.
18. G. Paaß and F.J. Kurfeß (Eds.), *Wissensverarbeitung mit neuronalen Netzen (Knowledge Processing with Neural Networks)*, number 221 in GMD-Studien, Schloß Birlinghoven, 53757 Sankt Augustin, Germany, September 1993, Gesellschaft für Mathematik und Datenverarbeitung (GMD), Workshop KI '93.
19. G. Paaß and F.J. Kurfeß, *Wissensverarbeitung mit neuronalen Netzen*, O. Herzog, T. Christaller, and D. Schütt (Eds.), in *Grundlagen und Anwendungen der Künstlichen Intelligenz-17, Fachtagung für Künstliche Intelligenz (KI '93)*, Informatik aktuell, Subreihe Künstliche Intelligenz, Springer Verlag, Berlin, pp. 217–225, 1993.
20. F.J. Kurfeß and G. Paaß (Eds.), *Integration Neuronaler und Wissensbasierter Ansätze*, number 242 in GMD-Studien, D-53754 Sankt Augustin, September 1994, Gesellschaft für Informatik (GI), Gesellschaft für Mathematik und Datenverarbeitung (GMD), Workshop at the KI '94 Conference, Saarbrücken, Germany.
21. I. Duwe, F.J. Kurfeß, G. Paaß, and S. Vogel (Eds.), *Konnektionismus und neuronale Netze—Beiträge zur Herbstschule HeKoNN 94*, number 242 in GMD-Studien, D-53754 Sankt Augustin, Oktober 1994.
22. F.J. Kurfeß, *Wissensverarbeitung mit neuronalen Netzen*, edited by G. Dorffner, K. Möller, G. Paaß, and S. Vogel, in *Konnektionismus und neuronale Netze—Beiträge zur Herbstschule HeKoNN '95*, GMD-Studien, D-53754 Sankt Augustin, Oktober 1995, Gesellschaft für Mathematik und Datenverarbeitung (GMD), pp. 211–223.



Franz J. Kurfess was the director of the Software Engineering Lab at the Computer and Information Sciences Department, New Jersey

Institute of Technology (NJIT) until July 1999, when he joined the Computer Science Department at Concordia University in Montreal, Canada. His research activities are centered around knowledge management systems, in particular hybrid systems combining various methods for storing, processing, accessing, and presenting knowledge. Before joining NJIT, he worked in the areas of hybrid systems, neural networks, and parallel inference mechanisms at the University of Ulm, Germany, the International Computer Science Institute in Berkeley, California, and the Technical University in Munich, Germany.