Hierarchical Architecture of Expert Systems for Database Management

Manjunath R.
Bangalore University, India

INTRODUCTION

Expert systems have been applied to many areas of research to handle problems effectively. Designing and implementing an expert system is a difficult job, and it usually takes experimentation and experience to achieve high performance. The important feature of an expert system is that it should be easy to modify. They evolve gradually. This evolutionary or incremental development technique has to be noticed as the dominant methodology in the expert-system area.

Knowledge acquisition for expert systems poses many problems. Expert systems depend on a human expert to formulate knowledge in symbolic rules. It is almost impossible for an expert to describe knowledge entirely in the form of rules. An expert system may therefore not be able to diagnose a case that the expert is able to. The question is how to extract experience from a set of examples for the use of expert systems. Machine learning algorithms such as “learning from example” claim that they are able to extract knowledge from experience. Symbolic systems as, for example, ID3 (Quinlan, 1983) and version-space (Mitchell, 1982) are capable of learning from examples. Connectionist systems claim to have advantages over these systems in generalization and in handling noisy and incomplete data. For every data set, the rule-based systems have to find a definite diagnosis. Inconsistent data can force symbolic systems into an indefinite state. In connectionist networks, a distributed representation of concepts is used. The interference of different concepts allows networks to generalize.

A network computes for every input the best output. Due to this, connectionist networks perform well in handling noisy and incomplete data. They are also able to make a plausible statement about missing components. A system that uses a rule-based expert system with an integrated connectionist network could benefit from the described advantages of connectionist systems.

BACKGROUND

Maintenance of databases in medium-size and large-size organizations is quite involved in terms of dynamic reconfiguration, security, and the changing demands of its applications. Here, compact architecture making use of expert systems is explored to crisply update the database. An architecture with a unique combination of digital signal processing/information theory and database technology is tried. Neuro-fuzzy systems are introduced to learn “if-then-else” rules of expert systems. Kuo, Wu, and Wang (2000) developed a fuzzy neural network with linguistic teaching signals.

The novel feature of the expert system is that it makes use of a large number of previous outputs to generate the present output. Such a system is found to be adaptive and reconfigures fast. The expert system makes use of a learning algorithm based on differential feedback. The differentially fed learning algorithm (Manjunath & Gurumurthy, 2002) is introduced for learning. The learning error is found to be minimal with differential feedback. Here, a portion of the output is fed back to the input to improve the performance. The differential feedback technique is tried at the system level, making the system behave with the same set of learning properties. Thus, control of an expert system controls the entire system.

KNOWLEDGE EXTRACTION FROM DIFFERENTIALLY FED NEURAL NETWORKS

The expert systems are organized in a hierarchical fashion. Each level controls a unique set of databases. Finally, the different expert systems themselves are controlled by a larger expert system. This ensures security of databases and selective permissions to their access, (i.e., some of the data needs to be public and the rest has to be private and protected; a concept borrowed from object-oriented programming). Thus, the master expert system can have access to public information. The neural networks are integrated with a rule-based expert system. The system realizes the automatic acquisition of knowledge out of a set of examples. It enhances the reasoning capabilities of classical expert systems with the ability to generalize and the handle incomplete cases. It uses neural nets with differential feedback.
Hierarchical Architecture of Expert Systems for Database Management

algorithms to extract regularities out of case data. A symbolic-rule generator transforms these regularities into rules governing the expert system. The generated rules and the trained neural nets are embedded into the expert system as knowledge bases. In the system diagnosis phase it is possible to use these knowledge bases together with human experts’ knowledge bases in order to diagnose an unknown case. Furthermore, the system is able to diagnose and to complete inconsistent data using the trained neural nets exploiting their ability to generalize.

It is required to describe a possible approach for the optimization of the job scheduling in large distributed systems, based on self-organizing neural networks. This dynamic scheduling system should be seen as adaptive middle-layer software, aware of current available resources and making the scheduling decisions using past experience. It aims to optimize job-specific parameters as well as resource utilization. The scheduling system is able to dynamically learn and cluster information in a large dimensional parameter space and at the same time to explore new regions in the parameter’s space. This self-organizing scheduling system may offer a possible solution for providing an effective use of resources for the off-line data-processing jobs.

Finding and optimizing efficient job-scheduling policies in large distributed systems, which evolve dynamically, is a challenging task. It requires the analysis of a large number of parameters describing the jobs and the time-dependent state of the system. In one approach, the job-scheduling task in distributed architectures is based on self-organizing neural networks. The use of these networks enables the scheduling system to dynamically learn and cluster information in a high-dimensional parameter space. This approach may be applied to the problem of distributing off-line data processing. These jobs need random access to very large amounts of data, which are assumed to be organized and managed by distributed federations of OODB (object-oriented database) systems. Such a scheduling system may also help manage the way data are distributed among regional centers as a function of time, making it capable of providing useful information for the establishment and execution of data replication policies. A hybrid combination of neural networks and expert systems was tried by Apolloni, Zamponi, and Zanaboni (2000). Fdez-Riverola and Corchado (2003) used unsupervised learning for prediction of parameters during the learning process.

SYSTEM DESCRIPTION

Case data that are presented to an expert system are usually stored in a case database. A data-transformation module encodes such cases in a suitable way to be learned by neuronal networks. This module performs as follows:

First, it transforms or preprocesses the data so that the components with equal scopes or a hierarchy is formed. It has been shown that discretization of data (i.e., preprocessing of data in to several subdivisions) makes the neural network converge faster (Abu Bakar, Jaaman, Majid, Ismail & Shamsuddin, 2003). At the same time, hierarchy in the data can be maintained.

Second, the transformation module encodes the data into a binary input pattern because some neural networks, as, for example, the competitive learning model (Rumelhart & Zipser, 1985) processes only binary inputs. To do this, the intervals of the components are subdivided into different ranges. These ranges are adapted according to the distribution of the components. So, every component of a vector is represented by the range its value belongs to. Depending on the kind of representation, the ranges could be encoded locally.

With the transformed data, different neuronal networks with unsupervised learning algorithms, such as competitive learning (Rumelhart & Zipser, 1985), ART, and Kohonen, are trained. These networks have the ability to adapt their internal structures (i.e., weights) to the structure of the data. In a rule-generation module, the structures learned by the neuronal networks are detected, examined, and transformed into expert systems rules. These rules can be inspected by a human expert and added to an expert system. When a case is presented to the expert system, the system first tries to reason with the rules that have been acquired from the human expert to produce a suitable diagnosis. If this fails to produce a diagnosis, the new rules produced by the process described can be used. If the case can be handled in such a way, all steps of the reasoning process may be inspected by and explained to a user of the system.

If the system, however, is not able to produce a suitable diagnosis in this way, be it, that data is missing, or the input is erroneous or no rule fits the data, since such a case has not been considered while building the knowledge base, the expert system can turn the case over to the networks. The networks, with their ability to associate and generalize, search for a most suitting case that has been learned before. The diagnosis that has been associated with that case is then returned as a possible diagnosis.

PROPOSED ARCHITECTURE

In recent years, neural networks have been extensively used to simulate human behavior in areas such as vision, speech, and pattern recognition. In large scales, they
Related Content

An Empirical Analysis of the Object-Oriented Database Concurrency Control Mechanism O2C2
www.igi-global.com/article/empirical-analysis-object-oriented-database/51215?camid=4v1a

Applying Learner-Centered Design Principles to UML Sequence Diagrams
www.igi-global.com/article/applying-learner-centered-design-principles/3399?camid=4v1a

On the Implementation of a Logic Language for NP Search and Optimization Problems
www.igi-global.com/chapter/implementation-logic-language-search-optimization/20765?camid=4v1a

Similarity Search in Time Series Databases
www.igi-global.com/chapter/similarity-search-time-series-databases/11218?camid=4v1a