A Neuro-Expert System Architecture with Application to Alarm Processing in a Power System Control Centre

R. Khosla       T. Dillon

Department of Computer Science and Computer Engineering
La Trobe University
Bundoora, Victoria-3083, Australia

Abstract

In order to broaden the scope of application of NN's there has been a recent surge of interest in combining artificial neural networks with expert systems to solve real world problems. The two approaches need to be integrated in a way that we exploit their strengths and cover their weaknesses. The generalizing effects of neural nets need to be safely used especially in real-time applications. In this paper we propose a generic neuro-expert system architecture which can overcome difficulties faced by standalone expert systems and artificial neural networks.

1 Introduction

Expert systems (ESs) as they are currently structured suffer from number of weaknesses [3,5]. These include difficulties in defining the initial hypothesis to be investigated for backward chaining, and the need to define too many rules where there are special cases or a large number of combinatorial possibilities. Artificial neural networks (ANNs), on the other hand suffer from lack of structured knowledge representation, lack of inheritance, one step reasoning, an inability to interact with conventional symbolic databases and, inability to explain the reasons for conclusions reached. There is not much discussion in the existing literature [1,2,3] on a generic architecture for combining ESs and ANNs. This becomes important when one deals with real world problems of sufficient complexity. Further one needs to address the question as to what model should be adopted to maximise the strengths of both methodologies? There is little discussion on how to cope with the generalized results of the neural nets safely for real-time applications where cost of a misclassification can be catastrophic.

2 Neuro-Expert System Architecture

At higher (if not the highest) level of most problem domains, one needs some kind of decomposition for effecting the solution process. The ANNs which provide decomposition into micro-features because of their structure are suitable candidates for this problem decomposition at near top level. ANNs learn the heuristics or generalizations employed by domain experts from the microfeatures. ESs, on the other hand use rules or heuristics as interpreted to them by the domain expert. ANNs can thus be looked upon as mechanisms for generating goals and ESs as mechanisms for proving goals. Putting the two together, ANNs can be used at a higher level to explicitly tell an ES at a lower level where to start from or activate an ES or ANN module. This in a way is knowledge sharing between two heterogeneous knowledge structures. ESs, further can be used to interface
with symbolic databases and, knowledge explanation.

The abstraction of knowledge at the highest level should be enough to enable use of ANN's in the classification mode. By classification mode we mean the ANN has been trained for the entire training data set. At lower levels where the solution process has been narrowed down ANNs can be used in the generalizing mode. In the generalizing mode the ANN is not trained for the entire data set. However, it must be ensured that generalization effects can be safely used. Here, we would say that a symbolic checking mechanism would be handy to cross-check the results of the ANN. This is especially important for real-time systems where a wrong generalized output at the top level can make the system pursue a wrong branch in the search space and result in a non-satisfactory solution or time consuming backtracking to other possible candidate modules. Further, things can get more complicated if one ANN generalizer module feeds into another ANN generalizer module. The symbolic cross-checking module can be a rule based diagnosis system, a database of fault-propagation model/s of each component or other application dependent cross-checking module. Further, a symbolic cross-checking module is also useful for purpose of simulation and improving the generalization of the ANN and building up the optimal size of the training data set.

Thus it is better to concentrate more on properties of ANNs like distributed control, massive parallelism, and fast execution time at higher levels. In contrast at lower levels, one could use their capacity to generalize. The above ideas are reflected in Fig 1.

Some of the advantages of using this architecture are: a) This architecture provides us a hierarchical, distributed, heterogeneous knowledge structure; b) A hierarchical representation of NN modules provides for reduction in learning patterns, learning time and memory requirements. c) The decomposed format provides for modularity in the whole structure; d) This architecture tries to preserve the properties of maintenance, modularity, normally associated with expert systems and distributed control, parallelism, generalization, fast execution associated with neural networks; e) In implementation of this architecture, ANNs and rules could be called as methods in an object. ANNs could also be called from antecedent or consequent of a rule.

3 Application

Alarm Processing in power systems has recently generated considerable interest [4,6]. When multiple alarms with multiple and single faults are considered, one gets a proliferation of rules which prevent a timely response from the system. The neuro-expert system architecture was used to develop a real time Alarm Processing System that overcomes these difficulties [5].

4 Conclusion

The neuro-expert system architecture developed here can be applied in various problem domains like engineering, fault diagnosis, which require problem decomposition. Having used it on a real-time application we can recommend it for use in real-time systems. The neuro-expert system architecture implemented by us can be used at can be used at different levels of a power system hierarchy for alarm interpretation and fault diagnosis.

References