A Framework For Integrating Fault Diagnosis And Incremental Knowledge Acquisition In Connectionist Expert Systems

Joo-Hwee Lim, Ho-Chung Lui, and Pei-Zhuang Wang
Institute of Systems Science, National University of Singapore
Heng Mui Keng Terrace, Kent Ridge
Singapore 0511, Republic of Singapore

Abstract

In this paper, we propose a framework for integrating fault diagnosis and incremental knowledge acquisition in connectionist expert systems. A new case solved by the Diagnostic Function is formulated as a new example for the Learning Function to learn incrementally. The Diagnostic Function is composed of a neural networks-based Example Module and a symbolic-based Rule Module. While the Example Module is always first invoked to provide the short-cut solution, the Rule Module provides extensive coverage of cases to handle odd cases when Example Module fails. Two applications based on the proposed framework will also be briefly mentioned.

1 Introduction

Neural networks have been recognized as a new and powerful paradigm to fault diagnostic applications in the past few years (Dietz et al. 1989, McDuff & Simpson 1990a-c, Uhrig & Guo 1989, Venkatasubramanian et al. 1990, Yamamoto et al. 1990). This new method, known as Connectionist Expert Systems (Gallant 1988), alleviates the knowledge acquisition "bottleneck" and problem solving "brittleness" faced by traditional rule-based expert systems. It also enjoys the robustness and noise tolerance of similarity-based systems. However, it does not provide complete or near complete coverage of diagnostic cases and is inadequate in providing explanation to the user on its solution.

There are generally two ways to overcome these problems. One active area of research has been to extract or interprete rules from a trained neural network (Fu 1991, Nottola et al. 1991, Towell & Shavlik 1991) so as to understand the behaviours of the network. While this does not lend itself to handle unfamiliar cases, the other approach, favoured by us, supplements the network with a knowledge-based, symbolic module -- Rule Module (hereafter, RM). As similarity-based learning, which requires no domain knowledge model, is not universally applicable (Harandi & Lange 1990), other approches that rely on partial domain models are necessary to make use of the background knowledge. From the viewpoint of models of expertise, connectionist model falls within the implicit models (Slatter 1990). Other models of expertise like deep models and competence models (Slatter 1990) offer as alternatives for our RM. In essence, RM, which reasons from first principles, is useful for solving novel cases while the prototype-based neural network -- Example Module (hereafter, EM) -- which reasons from previous experience, is especially suited for solving frequently encountered problems with fast response. An analogous idea that integrate model-based reasoning and case-based reasoning to solve large novel problem is pursued in (Rajamoney & Lee 1991).

The other aspect of our approach is incremental learning (hereafter, IL). The loose coupling of EM and RM described above is tightened by feeding the new cases that fails the EM but solved by RM to EM to be incorporated into its network knowledge base incrementally. This allows the refinement of both reasoning processes and domain knowledge without looking at all past cases again. This responsiveness in learning will also be useful in real-time operation.

We describe the proposed framework in Section 2. The network representation of EM and the requirement of the learning algorithm are discussed in Section 3. The inference engine and symptom selection scheme in backward chaining follow in the next section. Two applications based on the framework will be briefly discussed in Section 5.

2 Framework

The framework that we proposed is summarized in Figure 1. After constructing the initial knowledge base for EM, the user, the fault diagnostic (hereafter, FD) function, and the IL function form a closed loop that allows the system to learn while it is being used. The system automatically and incrementally acquires the diagnostic knowledge from examples generated by the FD function. The FD invokes the EM to solve a case which matches the symptoms with the prototypes of past cases learned. It provides short-cut solution to a familiar case but fails on a novel case during which the RM is called upon to solve the case by step-by-step reasoning. The RM can follow causal models, rules, troubleshooting flowcharts or even solicit the answer from the expert in case where explicit knowledge is not available. The idea being that the verified fault(s) are used as new examples for EM to learn spontaneously. In this way, the knowledge base of EM grows over time during operation and the chance of failure on new cases decreases. The RM is also useful as a means to provide rigorous explanation when necessary.

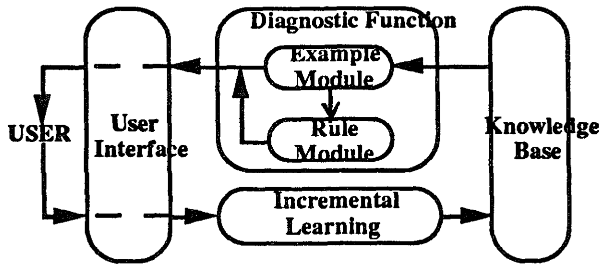

Figure 1: Framework to combine fault diagnosis and incremental learning.

3 Network Representation

The knowledge base of EM is represented as a feedforward 3-layer classification network (Figure 2). The nodes in the input layer and output layer denote the symptoms and the faults (classes) respectively. The hidden layer, whose nodes are recruited when necessary, captures the representative past examples (or their central tendencies) -- prototypes. The input layer is fully connected to the hidden layer while only those prototypes of the same fault class are joined to their respective output node. The input symptoms form the clustering space where each fault class is represented by a union of the regions representing the

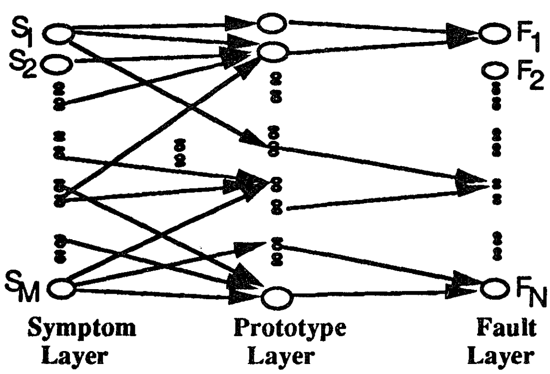

Figure 2: Network architecture.

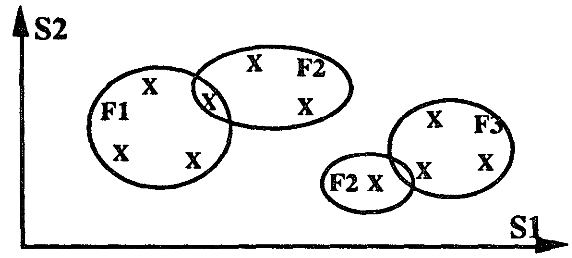

Figure 3: Fault clusters in symptom space.

prototypes of the class after learning. Figure 3 shows three fault classes (F1, F2, F3) with examples represented as 'x' on a 2-dimensional symptom space (S1, S2). Note that the boundary of a prototype can be 'fuzzy' by using the degree of similarity as the degree of belongness.

We require the hidden layer be constructed incrementally as follows. Given an example with input vector and its fault class, the learning algorithm compares the symptoms with those captured by the existing prototypes. If the new example is close enough, by means of similarity measure such as Euclidean distance, to some prototype(s) of the same fault class, the prototype(s) can be refined to capture this example or no action is taken. We can also penalise those well-matched prototypes of different fault classes. Otherwise, a new prototype is recruited to represent this new example with a connection to its fault class. A general formulation of the decisions to make in learning is given below:

> Suppose a new example consists of an input symptom vector, I, and output fault class, F. Let P_j be the pattern encoded by a prototype j, $\Phi(x,y)$ be a distance function and $\Omega(x)$ be a decreasing nonlinear function, and ρ be some predetermined threshold, then,
>
> Refine P_j to match more with I or take no action
>
> if there exists some prototype j of fault F, such that $\Omega(\Phi(I,P_j)) \geq \rho$ (1)
>
> Generate new prototype to encode I and connect it to fault F
>
> if for all prototypes j of fault F, such that $\Omega(\Phi(I,P_j)) < \rho$ (2)
>
> Adjust P_j to differ more from I or take no action
>
> if there exists some prototype j NOT of fault F, such that $\Omega(\Phi(I,P_j)) \geq \rho$ (3)

Note that many variants of the above general formulation are possible. Also if Ω is increasing, the relational operators '≥' and '<' become '≤' and '>' respectively. When only the distance is necessary, Ω reduces to an identity function. With this formulation, learning a new example needs not look at all the past examples learned and thus the border between batch and incremental learning becomes blur.

This kind of *generative* learning (Honavar & Uhr 1991) allows the network connectivity to be determined adaptively. It might also produce networks that learn rapidly without sacrificing the ability to generalize correctly to new input patterns in non-stationary, rapidly changing environments and has potential to solve the stability-plasticity dilemma (Grossberg 1980). There are a number of network architectures that fulfill our incremental learning requirement described above, to mention a few (Alpaydin 1991, Honavar & Uhr 1991, Reilly et al. 1982). We have also developed a new supervised incremental clustering algorithm which approximates the likelihood of the examples by using Gaussian prototypes (ie. Φ is the weighted Euclidean distance, Ω is the Gaussian function). The details of this learning algorithm will be reported elsewhere. We also envision the easiness of this class of learning algorithms to handle exceptions as nested exemplars as in (Salzberg 1991). To further compact the internal representation, a number of network pruning schemes can also be considered (Alpaydin 1991, Le Cun et al. 1990, Mozer & Smolensky 1989, Hanson & Pratt 1989).

4 Consultation

The general flow of consultation, as depicted in Figure 4, is described as follows. The user first enters some initial symptom(s) that he observed to the system. The EM is invoked which performs sequential, repetitive *hypothesize-and-test* cycles, resembling human diagnostic reasoning (Elstein et al. 78, Kassirer & Gorry 78, Rubin 75), until the fault is located. In particular, based on the current set of symptoms, the inference engine infers (*hypothesis generation* or *hypothesis updating*) a list of most probable faults and presents them to the user. The user verifies the fault by replacing the faulty component suggested by the system and testing if the problem disappears. If the user acknowledges that the actual fault has not been found, the system performs backward chaining (*hypothesis testing*) to solicit the values of relevant unknown symptoms which are useful in discriminating the possible faults. Once the real fault has been located, the user can save the case as a new example for IL. However, if the case is unfamiliar to EM, no satisfactory solution will be generated by the EM. The RM can then be activated which follows rules or first principle of the equipment model to pinpoint the fault. Similarly the result can be formulated as a new case for IL.

4.1 Inference Engine

With the proposed network representation, the EM infers the faults from the current symptoms by the nearest neighbor algorithm or its variants as follows:

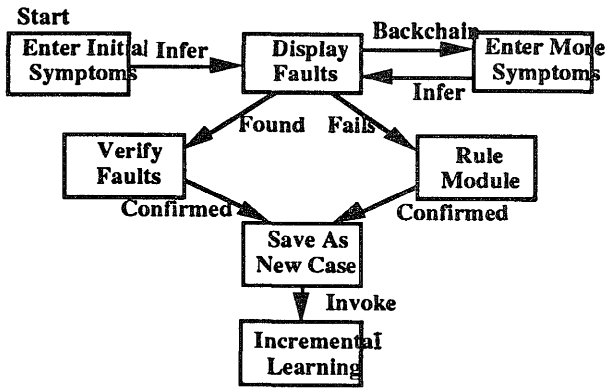

Figure 4: Consultation flow.

Let the input symptom vector be I, P_j be the pattern encoded by a prototype j, $\Phi(x,y)$ be a distance function and $\Omega(x)$ be a decreasing nonlinear function, and θ be some predetermined acceptance threshold, then,
A fault F is 'fired'

$$\text{if } \max_{j \in F} \Omega(\Phi(I,P_j)) \geq \theta \qquad (4)$$

We can regard this maximum activation of the prototypes of fault class F as the confidence measure for F. The fired faults are displayed to the user. As an alternative to using the threshold, we can select the top few faults with the highest confidence measure.

4.2 Backward Chaining

The purpose of backward chaining is to obtain more information (unknown symptom values) in order to better differentiate the fault clusters in a higher dimensional space. We have two possible schemes, namely the Information Heuristic Scheme and the Fuzzy Entropy Scheme.

4.2.1 Information Heuristic Scheme

The Information Heuristic Scheme attempts to select the most informative unknown symptom which roughly appears in half of the representative prototypes of the currently inferred faults with other heuristic knowledge information as follows:

For each undetermined symptom i,
compute $H_i = E_i + W_i - C_i$
Select symptom i* such that H_{i*} is the maximum

where W_i is the weightage (importance) of symptom i and

C_i is the cost incurred to determine the value of symptom i. They are to be supplied by the domain experts. E_i is the 'effectiveness' of symptom i which is obtained as follows:

Let F_j be one of the currently inferred faults, P_j be the pattern encoded by the prototype of F_j which has the maximum activation, and G_j be this maximum activation, then

$$E_i = \begin{cases} \sum_j G_j * A_{ij} & \text{if } \sum_j G_j * A_{ij} \leq \frac{M}{2} \\ M - \sum_j G_j * A_{ij} & \text{if } \sum_j G_j * A_{ij} > \frac{M}{2} \end{cases}$$

where A_{ij} is a boolean flag to indicate whether symptom i appears in prototype pattern P_j, and M is the number of the currently inferred faults.

4.2.2 Fuzzy Entropy Scheme

The Fuzzy Entropy Scheme (Wang 90, Zhang & Wang 88, Zadeh 78) selects the undetermined symptom with the smallest entropy as follows:

For each undetermined symptom i, compute

$$T_i = -K \sum_{s_i \in S} \sum_{F \in F_{s_i}} \Omega(\Phi(I,P_F)) * \log \Omega(\Phi(I,P_F))$$

Select symptom i* such that T_{i*} is the minimum

where s_i, the value of symptom i, ranges over all three possible values of symptom, namely {-1, 0, +1}, for the outer summation, F_{s_i}, is the set of inferred faults F with the addition of s_i into the currently known symptoms, P_F denotes the pattern of the prototype with the maximum activation of fault F, and K is a constant. The computation is intensive as the new set of faults have to be recomputed for each possible value of each unknown symptom considered.

5 Applications

INSIDE (Inertial Navigation System Interactive Diagnostic Expert) (Lim et al. 1991, Lui et al. 1991) is a connectionist dignostic expert system developed for the Singapore Airline (SIA) to assist technicians in diagnosing the avionic equipment known as Inertial Navigation System (INS). In this application, the EM is based on the RCE network (Reilly 1982) and the RM is a Flowchart Module which implements the troubleshooting flowcharts supplied by the INS manufacturer. The backward chaining scheme is similar to the Information Heuristic Scheme described above. There are 209 symptoms and 85 faults. With 990 past cases, 216 prototypes were formed. The diagnostic accuracy of the EM alone is 65%-75%. From the experiments that we carried out, the success rate improves when the training data size increases. Although the number of prototypes increases accordingly, the rate of increment declines. The system, which also has a record-keeping and report-generating module, tooks 20 man-months to complete in 1990 and is now in operation.

In the second quarter of 1991, we also conducted a feasibility study for the Port of Singapore Authority (PSA) to develop a prototype diagnostic system called COINCIDE (COnnectionist INcremental-learning Crane Interactive Diagnostic Expert) to train the new technicians in troubleshooting the quay cranes. In this case, the EM utilizes the supervised incremental clustering algorithm that we developed and due to time constraint, no RM has been implemented. Instead, when the EM fails to give a satisfactory answer, the human expert will instruct the system with the actual fault(s) for the system's IL to update the knowledge base. Both symptom selection schemes discussed in Section 4 have been incorporated, but the latter requires heavy computation that prolongs the response time. In this study, only the most-frequently-breakdown Spreader subsystem of the whole huge quay crane is considered, which consists of 100 symptoms and 93 faults. With limited time, the size and quality of the collected training data is not adequate for a quantitative analysis of the performance of the system at the moment. However, an initial benchmark of the learning algorithm on the INSIDE's data has shown a performance level of 88% (Chng 1992). There is a high possibility that we will extend the prototype to an operational system in the near future.

6 Conclusion

We have proposed a framework for connectionist expert systems to integrate the automatic and incremental construction of knowledge base with the diagnostic capability. A network representation based on prototypical learning is described and the suitable class of learning algorithm is identified. We also discuss the nearest neighbor-like inference engine and two possible backward chaining schemes. Finally, two applications based the framework are presented.

We are now turning our framework into a generic connectionist expert systems kernel which can be customized with different learning algorithms, RMs, and application domains easily. In the forthcoming future, we are also looking into IL algorithms with faster convergence rate and more compact prototype layer as well as neural network-based RM (Lim et al. 1991).

Acknowledgements

We would like to express our gratitudes to Dr Ding Liya, Mr Chen Chung-Chih, and Mr Chng Tiak Jung for contributing ideas in shaping the framework.

References

Alpaydin, E. (1991). GAL: Networks that grow when they learn and shrink when they forget. Tech. Report TR91-032, International Computer Science Institute, Berkeley.

Chng, T.J. (1992). Connectionist Expert System For Fault Diagnosis. Honours Year Project Report. Dept. of Information Systems and Computer Science, National University of Singapore.

Dietz, W.E., Kiech, E.L., & Ali, M. (1989). Jet and Rocket Engine Fault Diagnosis in Real Time. *Journal of Neural Network Computing*, Summer, pp. 5-18.

Elstein, A., Shulman, L., & Sprafka, S. (1978). *Medical Problem Solving - An Analysis of Clinical Reasoning.* Harvard University Press.

Fu, L.M. (1991). Rule Learning by Searching on Adapted Nets. In *Proceedings Ninth National Conference on Artificial Intelligence*, pp. 590-595.

Gallant, S.I. (1988). Connectionist Expert Systems. *Communications of the ACM*, 31(2): 152-169.

Grossberg, S. (1980). How does the brain build a cognitive code? *Psychological Review* 1: 1-51.

Hanson, S.J. & Pratt, L.Y. (1989). Some comparisons of constraints for minimal network construction with backpropagation. In D.S. Touretzky (Ed.) *Neural Information Processing Systems (Vol. 1).* San Mateo, CA: Morgan Kaufmann.

Harandi, M.T. & Lange, R. (1990). Model-based Knowledge Acquisition. In H. Adeli (Ed.), *Knowledge Engineering, Volume 1, Fundamentals.* McGraw-Hill Publishing Company.

Honavar, V. & Uhr, L. (1991). Generative Learning Structures and Processes for Generalized Connectionist Networks. Tech. Report #91-02, Dept. of Computer Science, Iowa State University.

Kassirer, J. & Gorry, G. (1978). Clinical Problem Solving: A Behavioral Analysis. *Ann. Intl. Med.*, 89, pp. 245-255.

Le Cun, Y., Denker, J.S., & Solla, S.A. (1990). Optimal Brain Damage. In D.S. Touretzky (Ed.) *Neural Information Processing Systems (Vol. 2).* San Mateo, CA: Morgan Kaufmann.

Lim, J.H, Lui, H.C., & Teh, H.H. (1991). A Rule Inferencing System Based On Neural-logic. Submitted to *IEEE Transactions on Neural Networks.*

Lim, J.H., Lui, H.C., Tan, A.H., & Teh, H.H. (1991). INSIDE: A Connectionist Case-based Diagnostic Expert System That Learns Incrementally. In *Proceedings of IJCNN'91*, Singapore, Nov. 18-21, 1991.

Lui, H.C., Tan, A.H., Lim, J.H., & Teh, H.H. (1991). Practical Application of a Connectionist Expert System: The INSIDE Story. In *Proceedings of the World Congress on Expert Systems*, Orlando, Florida, Dec. 16-19, 1991.

McDuff, R. & Simpson, P. (1990a). An Investigation of Neural Networks for F-16 Fault Diagnosis I: System Description. In *Conference Record of IEEE Anktestcon 1989*, Philadelphia, PA, pp. 351-357.

McDuff, R. & Simpson, P. (1990b). An Investigation of Neural Networks for F-16 Fault Diagnosis II: System Performance. In *Proceeding of the SPIE Technical Symposium on Aerospace Sensing, Applications of Neural Networks.*

McDuff, R. & Simpson, P. (1990c). An Adaptive Resonance Diagnostic System. *The Journal of Neural Networks*, Summer 1990. (preprint)

McMillan, C., Mozer, M.C., & Smolensky, P. (1991). The Connectionist Scientist Game: Rule Extraction and Refinement in a Neural Network. Tech. Report CU-CS-530-91, Dept. of Computer Science, Institute of Cognitive Science, University of Colorado.

Mozer, M.C. & Smolensky, P. (1989). Skeletonization: a technique for trimming the fat from a network via relevance assessment. In D.S. Touretzky (Ed.) *Neural Information Processing Systems (Vol. 1).* San Mateo, CA: Morgan Kaufmann.

Nottola, C., Condamin, L., & Naim, P. (1991). Hard Neural Networks for Rule Extraction. A Methodological Approach Applied to Business Financial Evaluation. In *Proceeding of Neuro-Nimes 91*, Nimes, France, Nov. 4-8, 1991.

Rajamoney, S.A. & Lee, H-Y. (1991). Prototype-Based Reasoning: An Integrated Approach to Solving Large Novel Problems. In *Proceeding Ninth National Conference on Artificial Intelligence*, AAAI Press/The MIT Press, pp. 34-39.

Reilly, D.L., Cooper, L.N., & Elbaum, C. (1982). A Neural Model for Category Learning. *Biological Cybernetics* 45, pp. 35-41.

Rubin, A. (1975). The Role of Hypotheses in Medical Diagnosis. In Proceeding of IJCAI, 1975, pp. 856-862.

Salzberg, S. (1991). A Nearest Hyperrectangle Learning Method. *Machine Learning* 6, pp. 251-276.

Slatter, P.E. (1990). Models of Experrise in Knowledge Engineering. In H. Adeli (Ed.), *Knowledge Engineering, Volume 1, Fundamentals.* McGraw-Hill Publishing Company.

Towell, G.G. & Shavlik, J.W. (1991). The Extraction of Refined Rules from Knowledge-Based Neural Networks. Machine Learning Research Group Working Paper 91-4, University of Wisconsin-Madison. Submitted (8/91) to *Machine Learning.*

Uhrig, R. & Guo, Z. (1989). Use of Neural Networks in Nuclear Power Plant Diagnosis. In *Proceeding of the SPIE Technical Symposium on Aerospace Sensing, Applications of Artificial IntelligenceVII.*

Venkatasubramanian, V., Vaidyanathan, R., & Yamamoto, Y. (1990). Process Fault Detection and Diagnosis using Neural Networks--I. Steady-State

Processes. *Computers Chem. Engng*, 14(7):699-712.
Wang, Pei-Zhuang. (1990). A Factor Spaces Approach to Knowledge Representation. *Fuzzy Sets and Systems*, 36: 113-124.
Yamamoto, Y. & Venkatasubramanian, V. (1990). Integrating Qualitative and Quantitative Neural Networks for Fault Detection and Diagnosis. Submitted to *AICWEJ*.
Zadeh, L.A. (1978). Fuzzy Sets as a Basis for a Theory of Possibility. *Fuzzy Sets and Systems*, 1: 3-28.
Zhang, Hong-Min & Wang, Pei-Zhuang. (1988). A Fuzzy Diagnostic Expert System - FUDES. FAS, Presented at *7th NAFIPS Conference*, San Francisco, CA, USA.