TECHNICAL RESEARCH REPORT

SYSTEMS RESEARCH CENTER

Supported by the
National Science Foundation
Engineering Research Center
Program (NSFD CD 8803012),
the University of Maryland,
Harvard University,
and Industry

Integrating Neural Network and Expert Reasoning: An Example

by J. Hendler and L. Dickens

TR 91-11

To appear: AISB Conference, April 1991.

Integrating Neural Network and Expert Reasoning: An Example

James Hendler[1]
Leonard Dickens

Department of Computer Science
University of Maryland
College Park, MD 20742

ABSTRACT

In this paper we describe a shell which has been developed to allow an integration of neural network and expert systems technology. The system, SCRuFFy, is based on an analysis of the different abilities and time courses of NN and AI systems. Critical to the processing of this system is a temporal pattern matcher which is used to mediate between the two subsystems, providing a "signal to symbol" mapping. This mapping allows the expert system to reason about the time course of signals which are classified by a connectionist network which is trained via classical back-propagation of error during a separate training phase. An example of the simulated control of the temperature of an underwater welding robot is presented to demonstrate these capabilities.

INTRODUCTION

Steels (1989) has pointed out that connectionist networks can perform heuristic classification tasks in a matter different than, but comparable to, the heuristic classification used in many expert systems. He argues that the distinction of which to use should be based on an analysis of features of the domain, rather than based on pre-theoretical biases. In addition he outlines a set of criteria to determine those situations in which a statistically-based classification method (such as the connectionist back-propagation algorithm) can be used: the cost of observation should be low, the features of the class may or may not all be present, the set of classes is known prior to classification time, the categorizations are based on statistical criteria, and hierarchical structure of the classes is either not necessary or used in only a limited manner.

One type of operation fitting these criteria is the categorization of sensor signals into classes. Typically, sensor data is obtained cheaply at a high rate of speed, features of the class may be underdetermined (necessitating a learning algorithm), classification of the signals is desired to match known categories, classification is based on high-order statistical correlations between features in the signal, and no hierarchical categorization of the signals is typically performed. Thus, it is no surprise that one of the greatest successes of connectionist models is in the classification of sensor signals.

Making intelligent decisions based on signals received from sensors,however,

[1]e-mail: hendler@cs.umd.edu. Dr. Hendler is supported in part by ONR grant N00014-88-K-0560 and NSF grant IRI-8907890 and by the UM Systems Research Center.

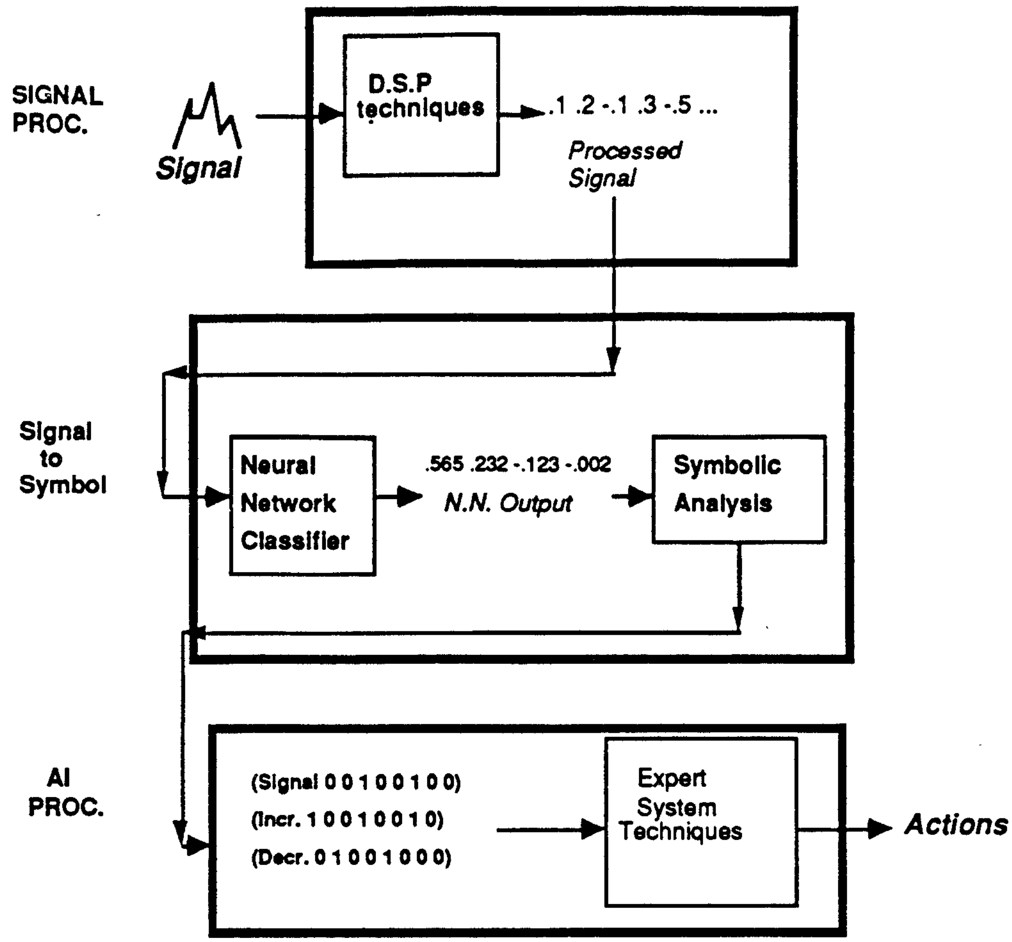

Figure 1: Flow of Control in SCRuFFy

of the flow of control in the system is shown in Figure 1. A sensor signal feeds through a digital signal processing program which converts it into a vector of discrete values. A set of known signals is gathered, and these are used to train a back propagation network to perform the signal classification task. Once the network is trained, the flow of control is for a new signal to be received, to be sampled, and to be classified using the connectionist network. The output of this process is then fed into a symbolic analysis program – a program which tracks the changes in signal classification over time and produces symbolic information describing the time course of the signal. This symbolic information is put onto a blackboard (or working memory) and thus a traditional blackboard-based expert reasoning system can make control decisions based on this information.

The heart of the SCRuFFy system is a "temporal pattern matcher", which provides an interface between the continuous signal sampling and classification, provided by the analog connectionist system, and the expert system used in making the control decisions. The pattern matcher is based on the observation that the particular activation patterns from the connectionist component for a given signal will usually be less interesting to the expert system-based reasoning than will the pattern of a set of signals over time. For example, in a simple application, we might have 10 outputs coming from the connectionist system. Two of these might be indicators of normal operation, while the 8 others represent different kinds of errors. If a controller takes no action until one of the error signals is the not yet been attempted, instead, diagnostic messages describing what is occuring and what actions are suggested are the outputs of the system.

The following is an example of SCRuFFy running on acoustic data. Comments are indicated in *italics*, output from the expert system component (i.e. system control decisions) are indicated by SMALL UPPERCASE text. The example uses signals from a simulator designed to show particular behaviors. In this case, we see SCRuFFy taking actions to control "streaming," a particular error condition arising in the welding.

```
; The expert system is initialized

********** INITIALIZING EXPERT SYSTEM
********** INPUT WATER TEMPERATURE: 021
********** INPUT WATER DEPTH: 030
; the context information provided by water depth and temperature
; is used by the SCRuFFy's expert system component.
===============================================
; the first signal is sampled
Signal (after DSP) is:

0.727 0.672 0.508 0.738 0.56 0.633 0.144 0.429 0.196 0.554 0.658 0.483
0.602 0.41 0.371 0.955 0.225 0.796 0.021 0.922 0.447 0.255 0.549 0.855
0.97 0.807 0.77 6 0.641 0.834 0.674 0.581 0.726 0.005 0.716 0.073
0.664 0.69 0.191 0.905 0.367 0.427 0.844 0.783 0.078 0.403 0.453 0.775
0.989 0.292 0.726 0.355 0.335 0.61 0.12 0.723 0.629 0.857 0.889 0.864
0.63 0.369 0.406 0.246 0.617 0.552 0.014 0.09 3 0.633 0.704 0.682
0.592 0.615 0.769 0.945 0.893 0.512 0.68 0.587 0.473 0.681 0.732 0.475
0.956 0.142 0.986 0.947 0.827 0.82 0.602 0.764 0.209 0.884 0.788 0.208
0.301 0.036 0.304 0.351 0.347 0.249 0.124 0.352 0.922 0.272 0.618
0.768 0.92 7 0.554 0.925 0.057 0.321 0.09 0.171 0.527 0.86 0.231
0.768 0.111 0.798 0.78 0.891 0.786 0.905 0.678 0.751 0.981 0.957 0.732
; This signal is input to the pre-trained connectionist network
Output from Connectionist network is:
.9296 .0145 .0759 .0465
; And the output is recorded for use by the temporal pattern matcher

The temporal pattern matcher finds any patterns that match
Output from temporal pattern matcher:
  Normal operation.
  A is highest, and no increasing minors
; Control is then transferred to the expert systems
Transferring control to Expert System:
_________
********** NORMAL OPERATION OF WELDER
; and diagnostic or control messages are produced
===============================================
... ; Several normal signals are omitted
===============================================
; another signal is sampled
Signal (after DSP) is:
  ... ; signals are deleted to save space
Output from Connectionist network is:
.9405 .0323 0.005 .2729
Output from temporal pattern matcher:
  Normal operation.
```

sidered to be quite a limitation. The overall system is a "loosely-coupled" hybrid model[6] and cannot take self-modifying actions.

The current system provides a mapping from signals through to "control" actions proposed by the expert system reasoning component. There is no reason that these control actions could not be self-modifying, that is, changing the various parameters and rules used by the system itself. Thus, in addition to control of an external system, the current model can be extended for self control. The major forms of self-modification we are currently examining include:

- Changing the temporal patterns: Upon the receipt of external information (or via internal decisions), the system can modify the set of temporal patterns of interest.

- Changing network parameters and/or output multipliers: Neural networks are governed by a set of parameters (learning rate, momentum, etc.). Controlling these parameters dynamically is an area just starting to be explored. As control models are developed, "intelligent" control (using the expert system) can be implemented, and experimented with, in this framework.

- Changing network configuration: Certain sorts of network models (such as Hopfield networks for minimal cost paths) have structural constraints depending on external information (path cost between nodes, etc.) When information is discovered during processing, it may reflect a change in this information, and thus necessitate dynamic reconfiguration of the network.

In addition, the current model provides for a single network reporting to a single temporal pattern-matcher. However, the blackboard architecture on which SCRuFFy is based provides a natural mechanism for the integration of multiple knowledge sources. A hierarchical model of the temporal patterns (in which patterns found for a given network can be compared with other patterns for other networks) can provide for a merging of the data from various sensors. By exploring the use of such patterns, coupled with the parameter changes described above, we believe this framework will allow for experimentation with varying models of multi-sensor systems.

CONCLUSIONS

In this paper we have described a shell which was developed to allow an integration of neural network and expert systems technology. The system, SCRuFFy, is based on an analysis of the different abilities and time courses of neural network and AI systems. Critical to the processing of this system is a temporal pattern matcher which is used to mediate between the two subsystems, providing a "signal to symbol" mapping. This mapping allows the expert system to reason about the time course of signals which are classified by a connectionist network which is trained via classical back-propagation of error during a separate training phase. An example of the simulated control of the temperature of an underwater welding robot was presented to demonstrate these capabilities.

[6]See (Hendler, 1989b) for a discussion of loosely and tightly coupled hybrid models.