N93-1

1992

NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

MARSHALL SPACE FLIGHT CENTER
THE UNIVERSITY OF ALABAMA

NEURAL NETWORK ARCHITECTURES TO ANALYZE OPAD DATA

Prepared By: Kevin W. Whitaker, Ph.D.

Academic Rank: Assistant Professor

Institution and Department: The University of Alabama
Department of Aerospace Engineering

NASA/MSFC:

Laboratory: Information and Electronic Systems
Division: Guidance, Control, and Optical Systems
Branch: Sensors

MSFC Colleagues: W. T. Powers
Anita E. Cooper

INTRODUCTION

Plume emission spectroscopy (PES) can be applied to rocket engine testing by treating the engine plume as a precisely-controlled laboratory flame for chemical analysis. Test stand or remotely-mounted telescopes can collect engine plume emissions and direct the light, via a grating spectrometer system, onto a linear array of silicon photodetectors. In a quantitative manner, light from many wavelengths of interest can be compared to identify elements, ratioed to recognize alloys, or monitored as a function of time to establish trends and the onset of significant material erosion.

The space shuttle main engine (SSME) became the subject of plume emission spectroscopy in 1986 when researchers from NASA-Marshall Space Flight Center (MSFC), Arnold Engineering Development Center (AEDC), and Rocketdyne went to the SSME test stands at the NASA-Stennis Space Center and at Rocketdyne's Santa Susana Field Laboratory to optically observe the plume. Since then, plume spectral acquisitions have recorded many nominal tests and the qualitative spectral features of the SSME plume are now well established. Significant discoveries made with both wide-band and narrow-band PES systems led MSFC to promote the Optical Plume Anomaly Detection (OPAD) program with a goal of instrumenting all SSME test stands with customized spectrometer systems.

A prototype OPAD system is now installed on the SSME Technology Test Bed (TTB) at MSFC. The OPAD system instrumentation consists of a broad-band, optical multiple-channel analyzer (OMA) and a narrow-band device called a polychrometer. The OMA is a high-resolution (1.5-2.0 Angstroms) "super-spectrometer" covering the near-ultraviolet to near-infrared waveband (2800-7400 Angstroms), providing two scans per second. The polychrometer consists of sixteen narrow-band radiometers: fourteen monitoring discrete wavelengths of health and condition monitoring species and two dedicated to monitoring background emissions. All 16 channels are capable of providing 500 samples per second. To date, the prototype OPAD system has been used during 35 SSME firings on the TTB, collecting well over 200 megabytes of plume spectral data.

The plume spectral data analysis and database correlation efforts to determine how much of a specie (or element) is present and where it came from require the handling and processing of a massive database. The data analysis is an incredibly labor intensive task and not one to be performed by hand. In the case of engine monitoring for flight operations, which is the ultimate goal of the OPAD technology program, PES contributions to a condition monitoring system must be made in real-time or near real-time in addition to processing a large database. These OPAD system requirements dictate the need for fast, efficient data processing techniques.

To address this need of the OPAD system, a study was conducted into how artificial neural networks could be used to assist in the analysis of plume spectral data.

WHY NEURAL NETWORKS?

The latest trend in artificial intelligence (AI) research is the resurrection of artificial neural networks (often simply called neural networks). Inspired by the working of the human brain and nervous system, neural networks have recently offered breakthroughs in vision and speech processing technology. Additionally, the systems have contributed to the practice of operations research providing insight to decision-making methodology. Neural network research was originally conducted in the 1960's but the research became dormant for almost the next two decades, primarily due to the inability to train a multiple-layered network. This was overcome in the late 1980's and neural network research has started to flourish once again with many successes.

A fundamental difference between the philosophies underlying conventional artificial intelligence techniques (such as expert systems or rule-based systems) and neural networks involves the emphasis on logical versus perceptive abilities. For example, AI in the form of expert systems employs logic to drive its intelligence using rules and their logical combinations. Neural networks, on the other hand, are based on the ability to recognize patterns through experience, rather than the deduction or recollection of rules. In fact, neural networks can also be referred to as adaptive pattern recognition devices but, because of their flexibility and ability to generalize, their applicability typically extends well outside the established realm of pattern recognition. Since neural networks excel at tasks requiring the identification of relationships among large amounts of data, such as is the case with the OPAD data, they are a superior alternative to the conventional expert system approach.

WHAT TYPE OF NEURAL NETWORK?

All neural networks have some type of structure associated with them. In fact, much research has been conducted into the orientation of neuron layers, neuron interconnections, and the interconnection weights. Unfortunately, parameters such as the number of neurons, the interconnection schemes, and the network learning-rate strategy elude *a priori* assignment and require something of a "black art" to determine them. Thus most neural network research centers around finding these network parameters.

Many neural network architectures exist. However, only a few are suited for analyzing OPAD data and these are discussed below.

Traditional Neural Networks

A traditional neural network is based on the concept of the adaptive linear element first studied in detail by Widrow and Angell[1] in the early 1960's. The network consists of simple, highly-interconnected processing elements called neurons. Each input signal to a processing element is amplified or dampened by a weighting factor associated with the path from the signal source to the processing element. The processing element collects all the weighted inputs and sums them to form a total weighted input. This weighted input is passed to an activation threshold function, typically the mathematical function called the sigmoid. If the weighted input exceeds a certain threshold, the neuron fires sending a signal along its output path to another processing element.

Neural systems solve problems by adapting to the nature of the data they receive. This is done primarily through supervised training. In supervised training, a network is told whether its answer (output) to a given input is correct or, if incorrect, what the magnitude of the error is. If the network's answer is not correct, the network begins to adjust the weights along each and every signal path to reduce the error between the predicted output and the actual output. The most popular supervised training method is called backpropagation, which was initially described by Werbos[2] in 1974, and later by Rummelhart et al.[3] in 1987. In short, backpropagation dictates how error at the output layer is fed backwards through the network, adjusting the path-weights along the way. Through repeated applications of a set of training data and subsequent backpropagation of error, the weights eventually converge to an optimal configuration which allows the neural network structure to identify patterns in any new data presented to it. (It is tacitly assumed that the new data lies within or near a domain spanned by the training data.)

Probabilistic Neural Networks

The structure of a probabilistic neural network (PNN) is similar to a traditional network built using a backpropagation training algorithm. The primary difference is that the sigmoidal activation function is replaced by a *class* of functions which includes, in particular, the exponential. Probability is then used to select which function in the class is used to determine whether or not the neuron fires. Theoretically, this allows data patterns to be quickly imbedded in the network structure. It has been demonstrated[4,5] that training a PNN is several orders-of-magnitude faster than training a traditional backpropagation network of comparable size.

Fuzzy Neural Networks

In practice there exists a close relationship between neural networks and fuzzy logic systems since they both work with degrees of imprecision in a space that is not defined by sharp, deterministic boundaries. Fuzzy and neural technologies can be fused into a unified methodology known as fuzzy neural networks in which the conventional control parameters in the neuron and the connection weights are replaced by fuzzy regions[6].

A fuzzy neuron has an architecture that approximates the classical McCulloch-Pitts neuron[7]. The differences between a traditional neuron and a fuzzy neuron are evident when examining their respective control properties. In place of scalar weights, fuzzy neurons use fuzzy sets as strength mediators of input signals. When a signal is receive by a fuzzy neuron, it is summed along with all the other active input signals to generate a cumulative signal. This cumulative signal is than mapped to a fuzzy set region containing possible activation levels. Using a defuzzification methodology, such as centroid or minimum entropy, the fuzzy region is collapsed to a scalar value representing the expected value of the fuzzy region under the signal conditions. A conventional fuzzy alpha-cut threshold is used by the neuron to decide whether or not the cumulative signal strength lies above a minimal gain boundary. If it is, the output of the fuzzy neuron is roughly proportional to the fuzzy compatibility state, mapping the input signal strength to the topology of the fuzzy threshold space. If the cumulative signal strength is below the threshold, the output is zero.

Entropy Networks

As has been repeatedly discussed, a multiple-layer artificial neural network structure is capable of recognizing patterns, or stated more formally, implementing arbitrary input-output mappings. Similarly, hierarchical classifier systems, more commonly known as decision trees, posses the capability to generate arbitrarily complex decision boundaries in an n-dimensional space. Exploiting this similarity, it is possible to restructure a given decision-tree as a multiple-layered neural network. This mapping of decision-trees into multiple-layered neural network structures can be used for the systematic design of a class of layered neural networks called entropy nets[9] which have far fewer interconnections than a traditional neural network. This means that the number of weights contained in the network is minimized while still maintaining the generalizability of the network.

WHAT SHOULD BE DONE?

The raw spectral data obtained by the OPAD system is voluminous, approaching seven megabytes per SSME test-firing. That is the primary difficulty to be tackled by any AI system which hopes to analyze (or help analyze) the data. Much hidden information is contained in each set of data. Patterns which exist may be functions of many variables such as time, engine age, or operating conditions. Also, there is an interplay between species data meaning that spectral signals must be looked at in unison rather that separately. This is a classic problem of complex pattern recognition for which neural networks are well suited.

The task at hand is to identify what type of neural network architecture should be utilized to analyze OPAD data. Unfortunately, even with all the neural network research being conducted there are no reliable rules for selecting neural network architectures for a given problem. Network selection and design is still very much an art rather than a science. To that end, specific recommendations are put forth as how to select, design, and apply neural network technology to the OPAD system.

Recommendation #1

All of the network architectures discussed have advantages and disadvantages when considering them for use with the OPAD system. With no basis for eliminating any of the network types, what should be done is evaluate each type with a systematic approach. An investigation should be undertaken to first construct neural networks of the various architecture types discussed in this report and then train each one on a *simplified subset* of OPAD (or OPAD-like) data to learn how network behavior and architecture interact. Items to be looked at should be ease of training, final size of the trained network, generalizability, and accuracy and speed of predictions.

Recommendation #2

Once each of the network types has been studied using a simplified set of data, they should be designed and trained using full sets of OPAD data. Again, each network will be evaluated on its ease of training, size, and generalizability. It is anticipated that at least one of the network architectures will prove to be superior but until they are all trained and exercised with real OPAD data it is unknown which one that will be.

Recommendation #3

Conduct a parallel research effort into developing new network architectures. It is conceivable that the neural network architectures discussed may not be entirely appropriate for the OPAD system. However, by combining some of the architectures it may be possible to develop a neural network which is, in essence, customized for OPAD data analysis. For example, it may be possible to combine fuzzy set theory with entropy nets to produce a network with an optimum number of nodes and the entropy net aspect will allow existing decision-tree structures from expert systems currently in use at Stennis Space Center to be converted into a neural network.

REFERENCES

1. Widrow, B., and Angell, J. B., "Reliable, Trainable Networks for Computing and Control," *Aerospace Engineering*, Vol. 21, pp. 78-123, 1962.

2. Werbos, P., "Beyond Regression: New tools for Prediction and Analysis in the Behavioral Sciences," Ph.D. Dissertation, Committee on Applied Mathematics, Harvard University, Nov. 1974.

3. Rummelhart, D. E., Hinton, G. E., and Williams, R. J., "Learning Internal Representations by Error Propagation," *Nature*, Vol. 323, pp. 318-362, 1986.

4. Specht, D. F., "Probabilistic Neural Networks," *Neural Networks*, Vol. 3, pp. 109-118, 1990.

5. Specht, D. F., "Probabilistic Neural Networks and the Polynomial Adaline as Complementary Techniques for Classification," *IEEE Transactions on Neural Networks*, Vol. 1, No. 1, pp. 111-121, 1990.

6. Cox, E., "Integrating Fuzzy Logic into Neural Nets," *AI Expert*, pp. 43-47, June 1992.

7. McCulloch, W. W., and Pitts, W., "A Logical Calculus of the Ideas Imminent in Nervous Activity," *Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115-133, 1943.

8. Rummelhart, D. E., "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing*, Vol. 1, pp. 318-362, 1986.

9. Sethi, I. K., "Entropy Nets: From Decision Trees to Neural Networks," *Proceedings of the IEEE*, Vol. 78, No. 10, pp. 1605-1613, 1990.