Neural Network Modeling of a Power Generation Gas Turbine

William E. Cole
Northeastern University

ABSTRACT

Over the past several years, I have supervised students creating Neural Network computer models of operating processes for their senior project. Processes modeled include a gas turbine power generator, a furnace, and building energy use. Models were created and used for parametric analysis within the scope of a one semester course. This modeling effort brought the actual operating process into the classroom, demonstrated to the students the value of computer modeling, and demonstrated that fundamental principles taught in the classroom apply to actual operating processes. This paper focuses on using neural networks to model processes, what students can learn from developing a neural network model, and one student's model of a gas turbine power generator.

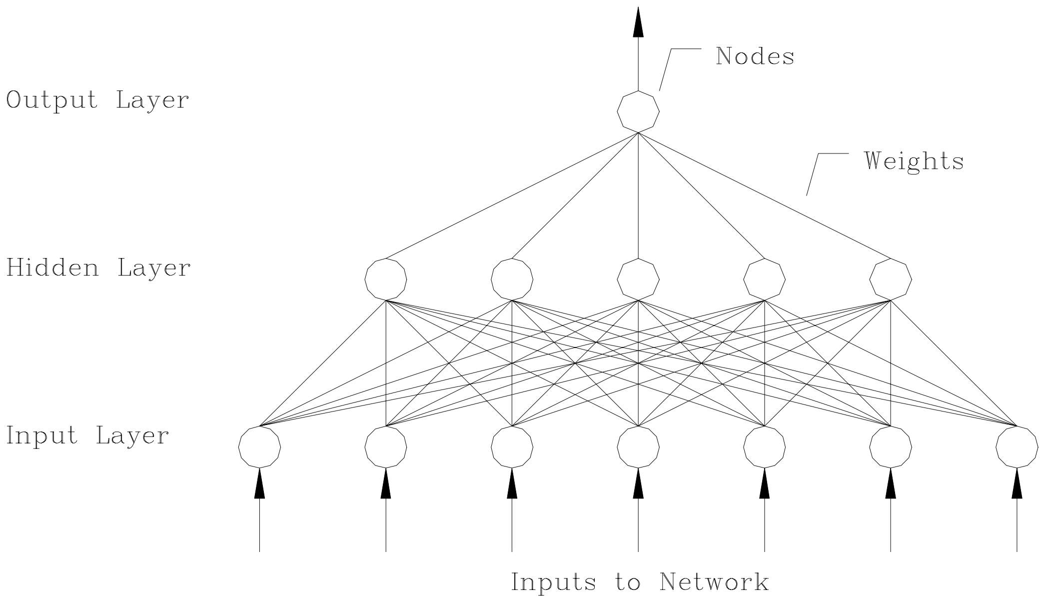
INTRODUCTION

The complexity of operating processes and the inherent difficulty of modeling real equipment makes modeling of industrial processes extremely difficult. The real equipment does not necessarily perform exactly as characterized by the idealized equations used in the models. Consequently computer models created from first principles are complex and frequently do not fit the operating data very well. Additionally, these models cannot account for the individual nuances of operating equipment and are not able to accommodate changes as the equipment ages. Consequently realistic models cannot be created from first principles within the scope of a one semester project.

An alternative technique to model complex processes is to utilize neural networks. Neural network modeling contrasts with conventional computer modeling in that a detailed understanding of the process is not required. The neural network uses operating data to create the model. Neural networks have been used to model complex processes such as distillation columns,[1] nuclear reactors,[2] and automotive fuel injection.[3] Additionally, realistic models can be created within the scope of a one semester project. Neural networks have also been used in a graduate level course at Tufts University.[4]

A neural network is composed of processing elements and connections as shown in Figure 1. The processing elements are arranged in three layers. In the first layer, each element represents one of the input parameters and the element in the third layer represents the output parameter. For educational purposes, each network should have only one output element. Additional processing elements are arranged in a second hidden layer. Each input processing element is connected to each hidden element and in turn each hidden element is connected to the output element. The processing elements first sums the values of the inputs applied to it and then takes the resulting

Figure 1: Neural Network Topology



value and transforms it with the sigmoid function. The sigmoid function is given by $1/(1+e^{-s})$ where s is the sum input to the element. The resulting value is the output of the processing element. The connections between processing elements each have independent coefficients called weights. This weight amplifies, attenuates, and can change the sign of the signal sent over the connection.

To calculate an output value, input values are placed in the input processing elements. New values are calculated by these processing elements and then multiplied by the appropriate weight to become inputs to the hidden elements. The hidden elements each process the sum of their inputs and output a value to the output element. The output element sums the inputs from the hidden elements and calculates the output parameter. By training the network to have the appropriate weights in the connections, the network can mimic the operational database.

The mathematics of neural networks is well developed and many computer programs are available to create and train neural networks.[5,6,7] For students purposes, Neuralyst by Cheshire Engineering Corporation[8] works well. This program is a series of macros within Excel. Thus the students are able to use a familiar spreadsheet program to analyze the operating data. Neuralyst handles the network formulation and training. Thus the students can focus on the operating data, data analysis, and network modeling rather than the mathematics of creating a neural network or learning a new computer program.

In creating a neural network model, the overall topology is first selected. This includes the input parameters to be considered and the output parameter. Two hidden Elements should be selected for the initial model. It has been shown that two hidden elements in a single hidden layer can model any complex function[9]. Next the weights in the connections between processing elements must be determined to model the database. This is performed by training the network.

Training the network determines the appropriate weights for the connections to minimize the error between the predicted and actual outputs. A sample data point is first presented to the network by loading the input values into the input elements. These values are then propagated through the network by the connections and processing elements to the output element. The value in the output element is then compared to the known value of the data point. The weights of the connections between the elements are then adjusted to minimize the sum of squares error between predicted and actual values. A second example is then selected and the process repeated. Thus through the successive use of the examples, the network is trained to learn what the output parameters should be for values of the input parameters.

Through training, the characteristics of the database are learned by the network through the imbedded values of the connection weights. This trained network can be used to predict values of the output parameter for specific values of the input parameters. Thus the network provides a computation model of the process which can be used to predict alternative operating conditions.

NEURAL NETWORK MODEL OF A POWER GENERATION TURBINE

One specific model developed was for a gas turbine power generating unit. The turbine was a ABB Type 11-D2 gas turbine rated at 65 Mw at ISO conditions. Input parameters used in formulating the model were turbine inlet temperature, air temperature, air pressure, and steam injection rate. Additionally, we were interested in optimizing the interval for cleaning the compressor of the unit. Therefore time since cleaning the compressor was an additional input parameter. Parameters modeled were the power output and the unit heat rate.

The first step in creating this model was to obtain a database. Getting this data provided the students an opportunity to visit the plant, make a presentation to station personnel, and follow up visits gave him the opportunity to review his results with operating personnel. This was more meaningful to the student than a plant tour since he had the opportunity to discuss specific plant problems with the operating personnel and he had a specific objective in mind for the visit.

Operating data spanning a one year period was obtained from plant personnel. This data included hourly readings of the turbine operating conditions over two thousand hours of operation. From this database, approximately eight hundred data points were used to train the network.

Students need to make several decisions in creating the model. The first of these is what parameters to use as input parameters and which to model as output parameters. For each of these data points, the turbine inlet temperature, air temperature, air pressure, steam injection rate, time since cleaning, power output, and heat rate were entered into the database created in Excel. Two separate neural networks were created: One to determine the power output and a second to determine the heat rate. For process modeling a separate network should be created for each output parameter. This makes a simple model that is easy for the students to understand and use. Though neural networks can be created with multiple outputs, i.e. both power output and heat rate from the same network, the additional complexity makes the network harder to train, interpret, and use. Only the power output network is discussed in this paper.

Figure 2: Power Output Neural Network Topology

Input Parameters		Hidden Nodes	Output Node	
Turbine Temp	o			
		o		
Air Temp	o			
Air Pressure	o		o	Power
Operating Time	o			
		o		
Steam Flow Rate	o			

The first step is to determine the topology of the model. The topology for the power output model is shown in Figure 2. There are five input parameters and one output parameter. Two hidden elements were selected for the initial model. To train the network, the neural network program takes the first data point in the database and places its values in each of the input processing elements. It then calculates the weights to minimize the error between the calculated power output and the actual power output. It then selects another data point and repeats this process. This is repeated for hundreds of repetitions through the database (one hundred repetitions was 80,000 data points in this example), creating a best fit model in the network. Though this process is mathematically complex, it is performed automatically by the Neuralyst program and transparent to the student. Computation time is only a few minutes on a 486 computer.

If the neural network is considered as a statistical routine determining the best fit of the model to the database, then statistically reasoning can be used to determine the topology and training of the model. If the model is given sufficient degrees of freedom, then it can be trained to perfectly fit the data. The exception to this rule is it cannot be trained to fit two data points with identical input parameters but a different output parameter. Rather it learns the average of the two values. However, if the model is given too many degrees of freedom, then it is fitting the noise of the database rather than the actual real dependencies. Thus the trick is to create the simplest model which can fit the real dependencies of the data with the minimum training. Any further complexity in the model is training it to learn the noise in the data and inhibits generalization of the model to other operating conditions. Much work is underway to determine the optimum way to train a network to obtain the best generalization of the results.[10,11] However, for the purposes of student projects, I have developed a simplified method of determining the network training and topology that, although not mathematically rigorous, does produce good models.

Figure 3: Neural Network Training

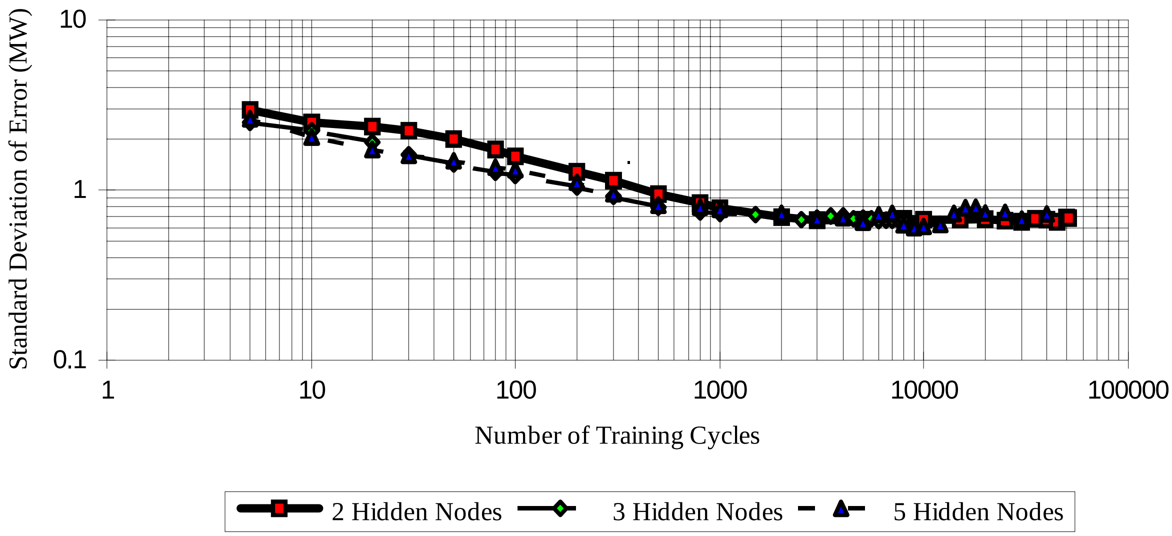
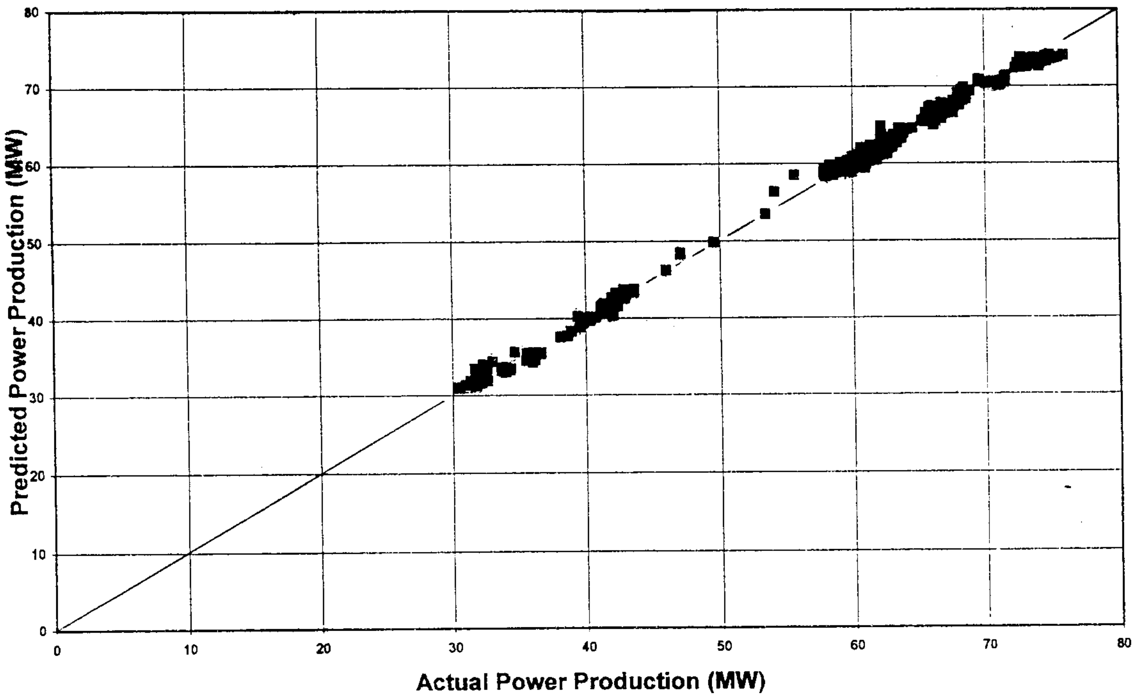


Figure 3 shows training results for the model. The standard deviation of the error decreases with training up to approximately one thousand training cycles. The standard deviation of the error is the standard deviation of the difference between the network predicted power output and the actual power output over the database. Each training cycle represents presentation of every data point to the network once. Figure 3 shows that increasing the training improves the network modeling of the data. However, after about 1000 cycles, this improvement in characterizing the database stops. This is because the network has learned the general dependencies and is now learning the noise in the data. If training continues, the standard deviation decreases slightly as the model learns the noise in the data. This plateau in the learning curve, also represents the standard deviation of the power output. For identical operating conditions, the power output was found to have a standard deviation of about 0.9 Mw compared to the plateau value of 0.7 Mw.

The different curves in Figure 3 represent training for a different number of hidden elements. Two hidden elements is the minimum number for any network. Increasing this number to three or five hidden Elements did not improve the learning of the process. However, increasing this number allowed the network to learn the noise in the data. Hence a better model is created with two Elements than with three or five nodes, even though the level of learning is the same.

Thus the process for creating the model is to first estimate the standard deviation of the output parameter. Then select two hidden elements in the network and train the network. Stop the training periodically and determine the standard deviation of the error. This deviation should reduce and then reach a plateau as shown in Figure 3. The error of this plateau should be approximately the standard deviation of the output parameter. Then the number of hidden nodes should be increased and the process repeated. The final network configuration should represent the minimum number of hidden nodes that produce a standard deviation of the model error approximately equal to the standard deviation of the output parameter with the minimum amount of training. In this case, two hidden nodes with about one thousand training cycles.

Figure 4: Actual and Predicted Power Output



The fit of this model to the operating data is shown in Figure 4. Plotted in this figure is the predicted power output as a function of the actual power output. A good comparison was obtained, showing that the model fits the operating data. The standard deviation of the error between the model prediction and the actual measured power output is 0.7 Mw.

The purpose of creating a model, however, is to better understand how the gas turbine operates and how operating conditions effect the gas turbine. To test this, input conditions are provided to the model and the power output predicted. Figure 5 shows the power output of the turbine for different turbine inlet temperatures with the atmospheric pressure, temperature, and hours since cleaning held constant. The model predicts that the power output increases with increasing turbine inlet temperature. The next step is to compare this observed effect with what the students are taught in thermodynamics.

In thermodynamics the students are taught that the power output from a turbine is given by:

Thus the power output should be proportional to the turbine inlet temperature, as shown in Figure 5. Thus the model confirms, using actual operating data, the variation predicted from first principles. This is an important step in validating a model and in the students understanding of neural network modeling. The model should show the students the validity of the material taught to them in the basic engineering courses.

Figure 5: Predicted Power Output variation with Turbine Inlet Temperature

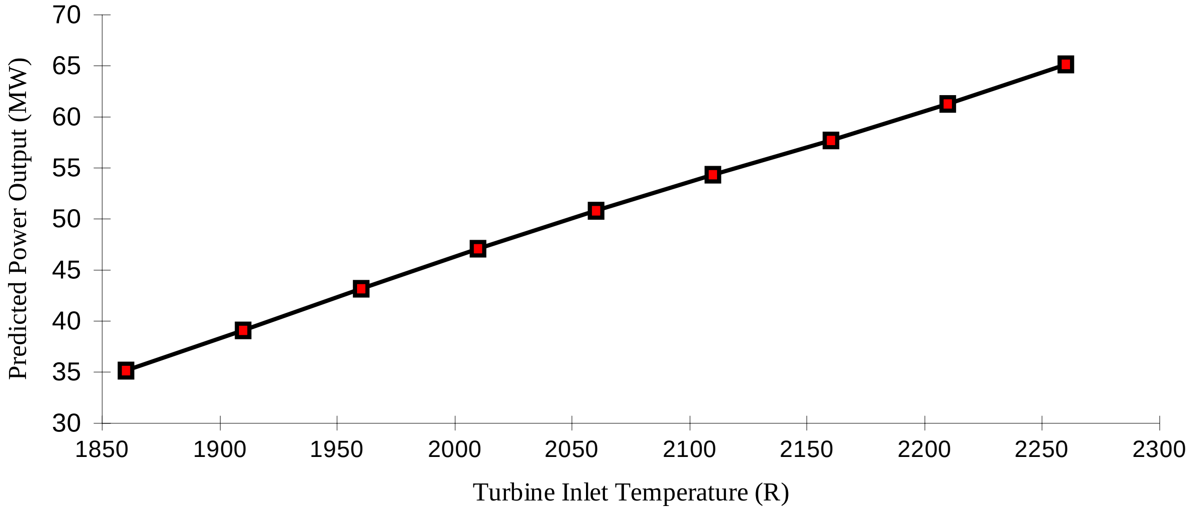


Figure 6 shows the model prediction for the power output as a function of the ambient air temperature. The power output decreases as the ambient temperature increases. This can also be determined from first principles. This is because the power output from the power turbine is constant while the power required by the compressor increases with increasing ambient temperature. Thus the power output should decrease linearly with ambient temperature, as shown in figure 6.

Figure 6: Predicted Power Output variation with Ambient Air Temperature

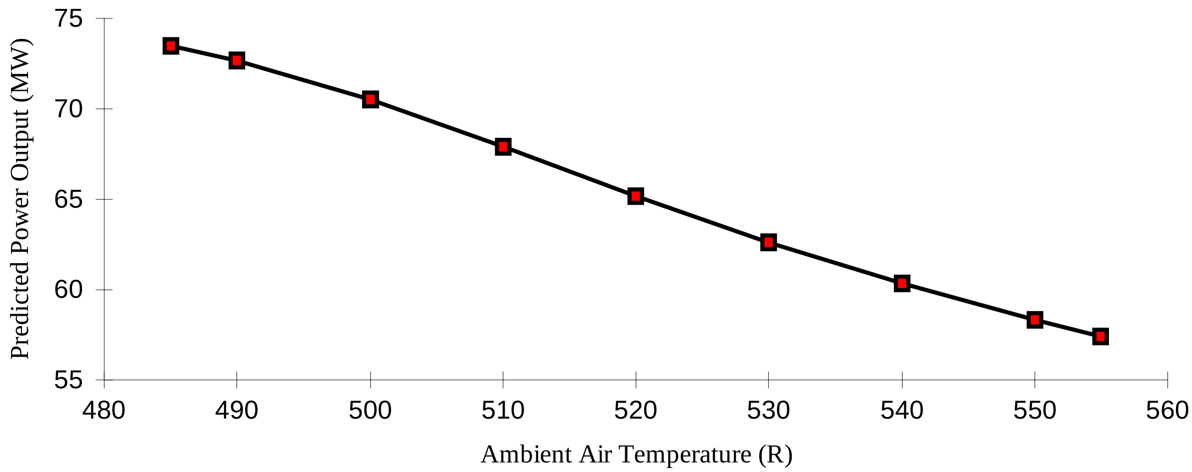


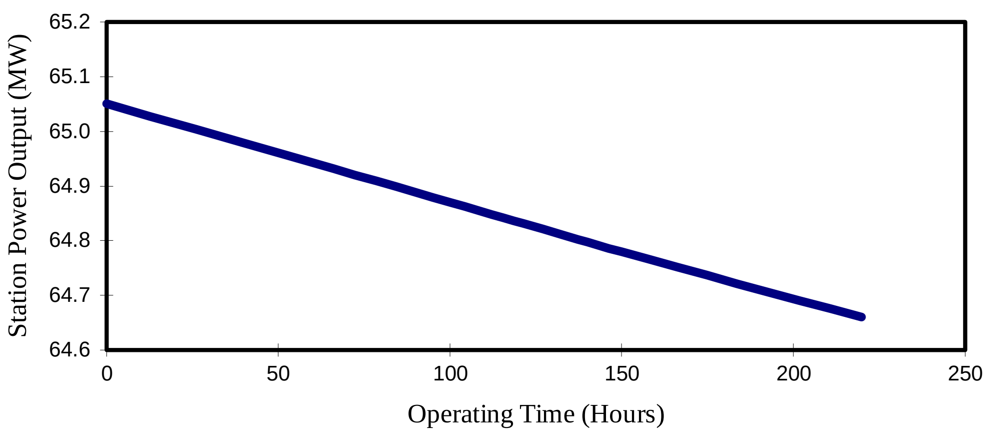
Figure 7 shows the model prediction for the power output as a function of the ambient air pressure. The power output increases linearly with air pressure. This can also be verified from first principles.

The previous effects observed in the model predictions, could all be determined from first principles about a baseline operating point. This helps to verify the model and give the student confidence that the model correctly predicts turbine operation. The power of the neural network modeling, however, is its ability to determine parameters that cannot be derived from first principles. One of these parameters is the importance of fouling of the compressor blades.

A gas turbine ingests ambient air ladened with contaminants including oil, pollen, and dirt. These materials tend to stick to the compressor blades, especially when the air is heated as it passes through the compressor. This fouling of the compressor blades is not fully understood and cannot be predicted from first principles. Studies reported in the literature discuss how fouling occurs,[12] how quickly it occurs and how to clean the compressor.[13, 14] Cleaning is performed by washing the compressor blades with a soap solution. This can be done either while the unit is running or shut down. However, even with online cleaning the compressor must occasionally be cleaned while shut down.

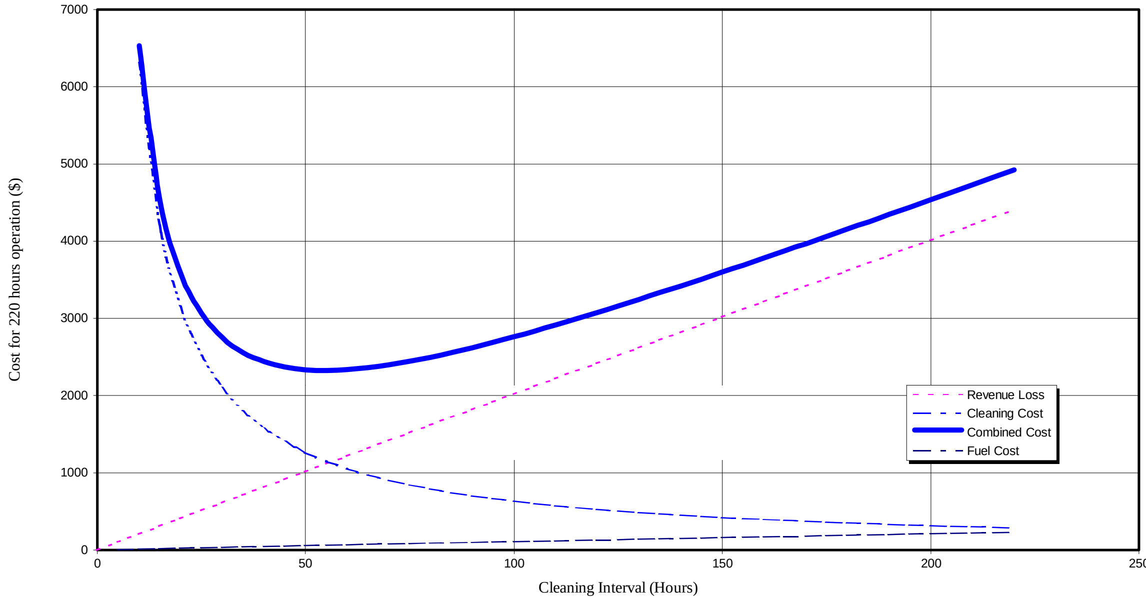
Figure 8 shows the predicted power output from the gas turbine as a function of the time since the compressor has been washed. The power output declines as the operating hours increase. This effect cannot be predicted from first principles. However, it was known to turbine operators. They did, however, not know the magnitude of the reduction with operating time. This prediction is also consistent with the observations reported by Haub[14] where a two percent reduction was reported over 10 days operation. Hence the neural network model was able to increase the understanding of the turbine unit by the operators. The other factor that was not known to the operators was the optimum interval for cleaning the compressor blades.

Figure 8: Predicted Power Output Variation with Time



The cost of cleaning the compressor is shown in Figure 9. Shown on this curve are the various costs associated with the fouling of the compressor. The lowest curve is cost associated with the increased fuel burned as the unit heat rate is increased by the compressor fouling. The increase in unit heat rate was determined by a second neural network similar to the one for power production. The second curve is the loss of revenue due to the reduced power output of the unit. The third curve represents the cost of cleaning the compressor. The final curve represents the sum of these three costs. This curve shows that the total costs are minimized when the compressor is cleaned every 50 hours. For intervals less than this, the cost of cleaning is more expensive than the loss of performance. For cleaning less frequently, the cost of the performance loss exceeds the cost of cleaning. Thus the optimum cleaning interval is somewhere between 50 and 100 hours. This was new information to the unit operators.

Figure 9: Cost of Compressor Fouling



CONCLUSIONS

This study showed the power of using neural networks to model a combined cycle power generation unit. From a technical perspective, results were consistent with operator experience and led to a definition of parameters that were unknown to the operators. From a student perspective, use of neural networks allowed the student to create a computer model of an actual operating unit in one semester. This model was then used to gain a further understanding of station operations and show that the general principles taught in thermodynamics applied to operating plants. It also provided the student with access to an operating power station and a chance to work with the plant personnel. Neural network modeling provides an excellent medium for teaching students about industrial processes as well as providing them with an opportunity to meet with plant personnel and learn about operational problems.

REFERENCES

1. MacMurray, Joel and Himmelblau, David, “Identification of a Packed Distillation Column for Control Via Artificial Neural Networks,” Proc. of the Am. Control Conference, San Francisco, Ca., June 1993, p1455.
2. Uhrig, Robert, “Potential Use of Neural Networks in Nuclear Power Plants,” Proceeding of the 8th Power Plant Dynamics Control and Testing Symposium, Knoxville, Tn, May 1992.
3. Shiraishi, Hitoshi, Ipri, Susan L., and Cho, Dan, “CMAC Neural Network for Fuel-Injection Control,” Proceeding of the American Control Conference, San Francisco, Ca., June 1993, p1773.
4. Winston, Arthur W., “Use of Neural Networks in Engineering Applications and the Education of Engineering Leaders,” Proceeding of the 1995 ASEE Annual Conference, p2828.
5. Baharin, Iskandar B., and Hasan, Md Mahmud, “A neural networks software for the control engineers,” Advances in Engineering Software, 22, 1995, p191.
6. Goh, A. T., “Back-propagation neural networks for modeling complex systems,” Artificial Intelligence in Engineering, 9, 1995, p143.
7. “Neural Network Resource Guide,” AI Expert, June 1995, p35.

8. Neuralyst™, Cheshire Engineering Corporation, 650 Sierra Madre Villa, Suite 201, Pasadena, Ca. 91107.
9. Mhaskar, H. N. "Neural Networks for Optimal Approximation of Smooth and Analytic Functions," Neural Computation, 8, 1996, p164.
10. Lehtokangas, Mikko, et al, "Predictive Minimum Description Length Criterion for Time Series Modeling with Neural Networks," Neural Computation, 8, 1996, p583.
11. Versaggi, Matthew R., "Understanding Conflicting Data," AI Expert, April 1995, p21.
12. Aker, G. F., and Saravanamuttoo, H. I. H., "Predicting Gas Turbine Performance Degradation Due to Compressor Fouling Using Computer Simulation Techniques," Journal of Engineering for Gas Turbines and Power, Vol. 111, April 1989, p343.
13. Harris, Howard and Calabrese, Marc, "On-Line Detergent Fluid Evaluation on a TF40B Gas Turbine Engine," Inter. Gas Turbine and Aeroengine Congress and Exp., The Hague, June 1994, ASME 94-GT-452.
14. Haub, Gary and Hauhe, Wm., "Field Evaluation of On-Line Compressor Cleaning in Heavy Duty Industrial Gas Turbines," Gas Turbine and Aeroengine Congress and Exp., Brussels, June 1990, ASME 90-GT-452.

WILLIAM E. COLE

Dr. Cole received his Bachelors of Mechanical Engineering Degree from Stevens Institute of Technology and his Doctor of Philosophy from the Pennsylvania State University. He has over twenty years of industrial experience developing industrial process equipment at The United Technologies Research Center and Thermo Electron Corp. Dr. Cole is now using this industrial experience to help educate the next generation of engineers.