Artificial Neural Networks in Elevator Dispatching

Bruce A. Powell
Otis Elevator Company, USA

David J. Sirag, Jr.
United Technologies Research Center, USA

Bradley L. Whitehall
Information Resources, Inc., USA

Keywords: Artificial neural networks, perceptron, elevator dispatching

ABSTRACT

This paper describes a "real world" application of neural networks that has enhanced the control of elevators in high rise buildings. A range of artificial neural networks (ANNs)--from a simple perceptron to complicated networks with hidden layer--have been developed and installed in advanced technology elevator systems. It has been demonstrated that the ANNs provide improved accuracy in estimating the response time for a given elevator to reach the floor of a waiting passenger. This greater accuracy has led to improvement in group control performance as measured generally by passenger waiting times. This paper defines the elevator dispatching problem and some of the research work that has led to the implementation of a neural network based elevator dispatcher in the real world.

1. INTRODUCTION

Otis Elevator Company has developed a proprietary dispatching process in which an artificial neural network (ANN) plays a major role in decision making. Tests have indicated that this learning process has the potential to provide significant improvement in performance of the group control scheme. From the customer's point of view, this improvement translates into fewer long waits for the elevator.

In this paper, we explore these issues in the context of developing an ANN to improve the performance of an elevator dispatching system. The process discussed here has resulted in a new product for Otis Elevator Company and has been running in job sites for some time now. The following points will be made in the body of this paper:

- Even the simplest of neural network technologies can yield improvements in product performance.

- There is no single correct way to introduce a neural network into a product. The simple neural network (e.g., a perceptron) is easy to implement but may not perform as well. The complex network (e.g., with hidden units) may perform better but may be more difficult to train and to maintain in the field.

- Even if a complex neural network can be shown to be better in the laboratory, the resulting improvements in the product performance may not be worth the cost over simpler network architectures.

2. BACKGROUND

The function of giving orders to each elevator as to where and when to travel is commonly referred to as ***dispatching***. This process is illustrated in Figure 1, which shows a snapshot of a hypothetical 12-floor building at the moment when a person on the sixth floor presses the down button to call for an elevator. The dispatching decision is simply stated:

Which of the two cars should stop at floor 6?

Figure 1

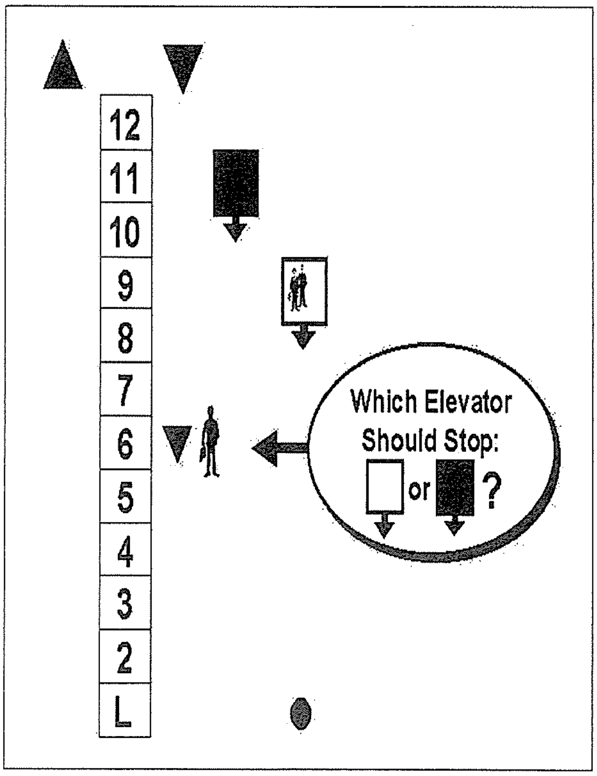
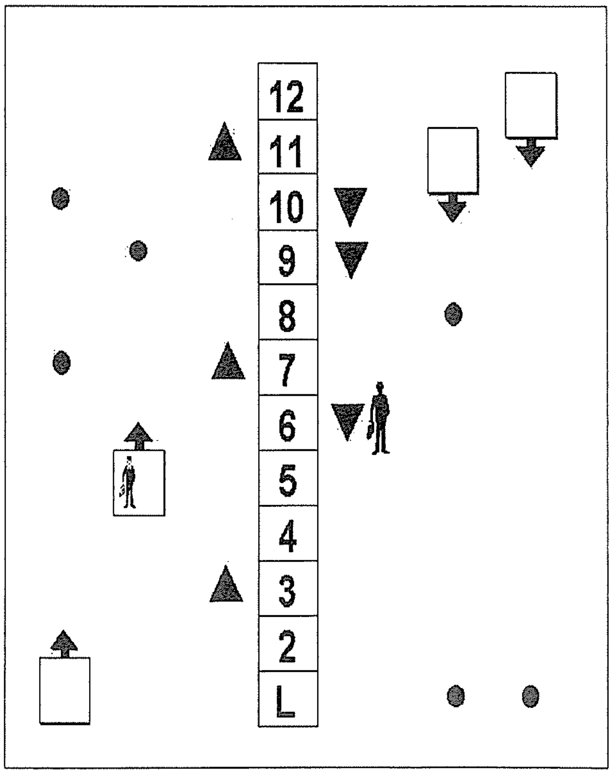
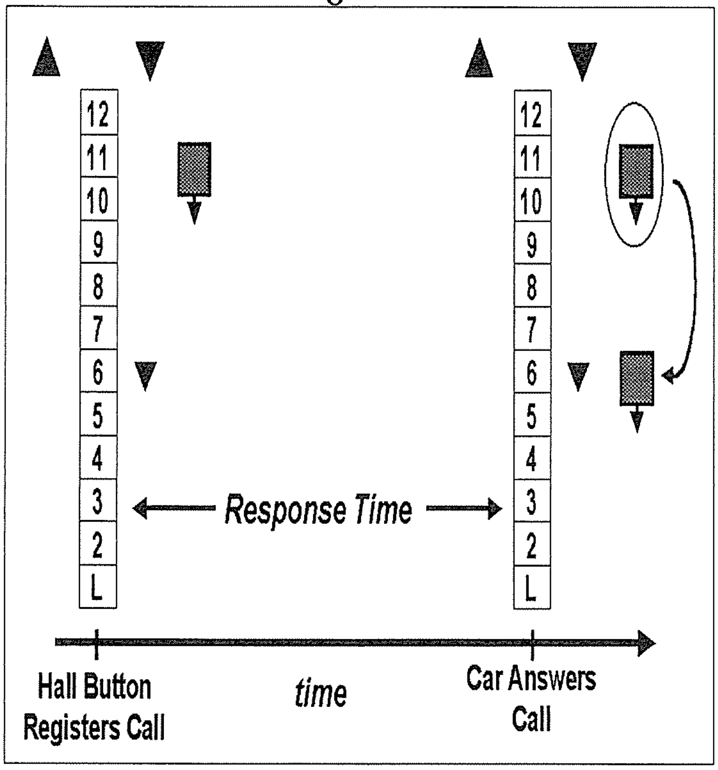


Figure 2



A busy building can have four or more elevators in a group and have many unanswered hall calls and car calls. Figure 2 shows a four-car group with six waiting hall calls, and each car has its own set of car call commitments. In contrast to the situation in Figure 1, there are 4^6 or 4,096 possible assignments of cars to hall calls

Figure 3

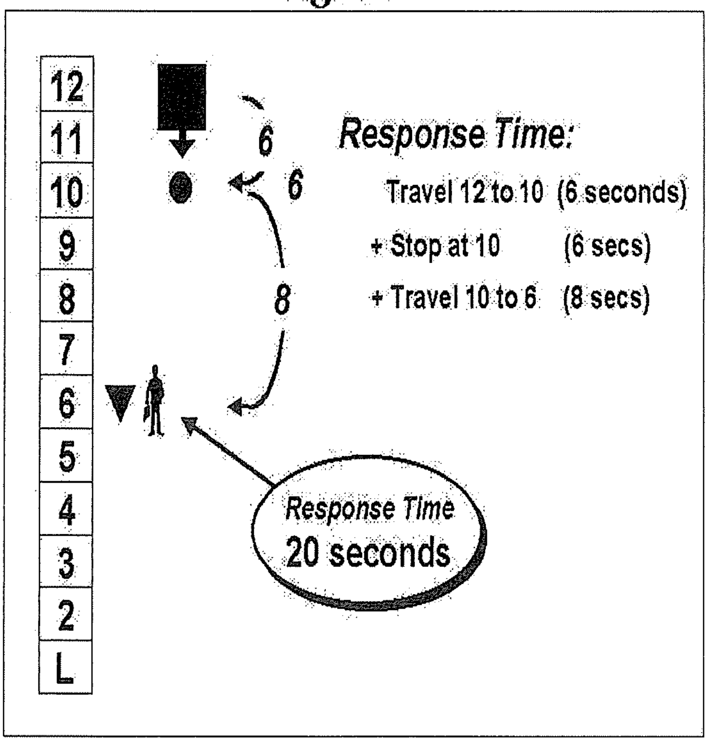


2.1 Conventional Estimation of Response Time

The dispatching decision is based on a number of important factors, one of which is an estimate of the time (in seconds) that each candidate car would require to reach the call. We call this factor Remaining Response Time (or RRT). Clearly, a good accurate estimate of RRT is needed to make a good decision.

Remaining Response Time is illustrated in Figure 3. Estimated response times are determined for each eligible car, and often--but certainly not always--the car with the shortest response time is selected. The conventional method for estimating response time is very straightforward. Considering only known commitments, one can allocate a certain amount of time for each trip segment. Figure 4 shows these deterministic methods for estimating the response time for a car to reach a down hall call on floor 6. Before reaching floor 6, this car must make a stop at floor 10 for a car call (i.e., to allow passengers to exit the elevator). A total of six seconds is allocated for this stop. In addition, each travel time segment is calculated by allocating five seconds for the first floor and one second for each subsequent floor in the same travel segment. The estimated response time for the car to reach the down hall call on floor 6 is thus 20 seconds.

Figure 4



Once an initial car assignment has been made, the elevator proceeds toward the hall call floor, making required stops along the way. As often happens, however, the car's trip can be interrupted. For example, while on the way to the hall call floor, another hall call might be assigned to the car. This new stop, unanticipated some few seconds ago, will result in the original response time estimate for the first hall call to be incorrect.

As a hall call waits for an elevator, the assigned car sometimes experiences significant delays so that the original estimate of response time is very much in error. Under these circumstances, the dispatcher would consider switching to another car. A decision about switching to another car would involve a comparison of the remaining response time of the presently assigned car to the estimated response times of the other cars.

2.2 Inherent Error in Conventional Estimation Procedure for Response Time

Consider the dispatching example of Figure 5a below, where a new down hall call has been registered from floor 8. A car traveling up, currently at the lobby, and has a passenger on board wanting to go to floor 8. The predicted response time for this car to reach floor 8 is a mere 11 seconds. Also, there are coincidental hall call and car calls at floor 8, making this car an ideal car to select.

Figure 5a

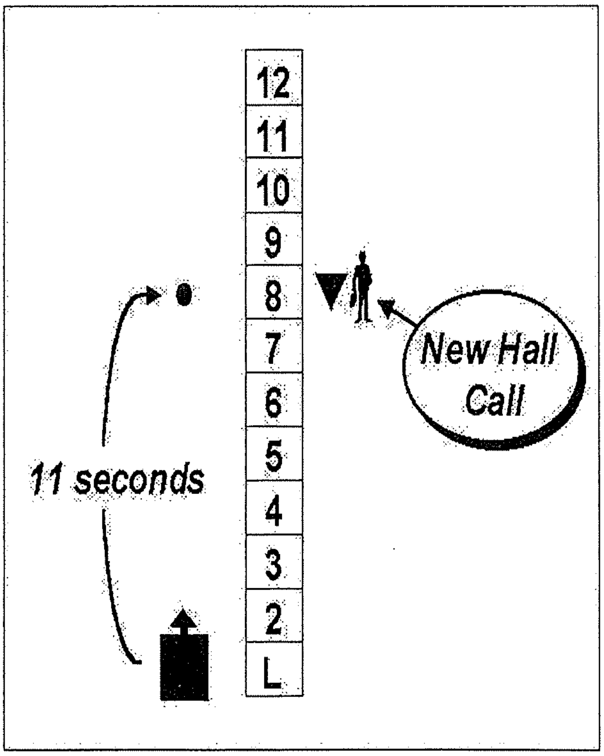
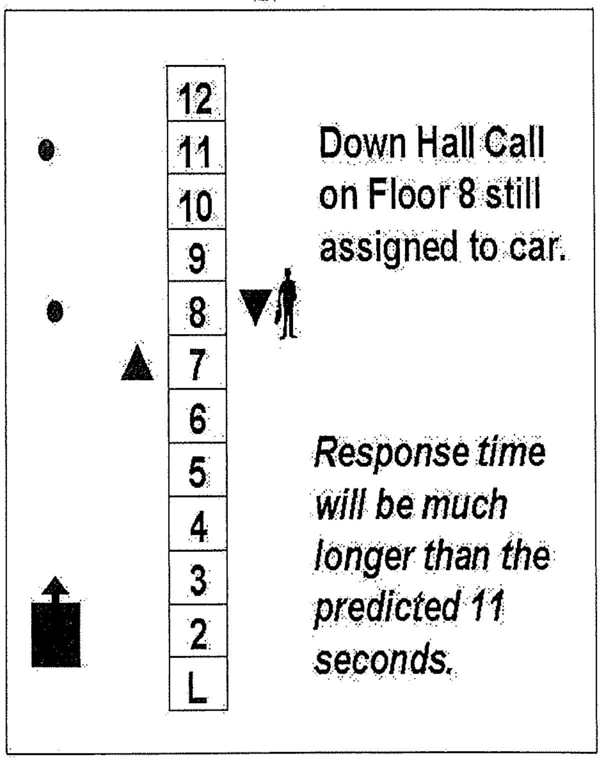


Figure 5b



Some five seconds later, as shown in Figure 5b, the car is seen to have received two additional commitments that were not known at the time that the dispatcher assigned the car to the down hall call on floor 8. These new commitments (a new UP hall call on floor 7, and a car call to floor 11) will cause the car to be severely delayed in reaching its assigned DOWN hall call at floor 8. Of course, if these new commitments were known in advance, the predicted response time would have been longer, and most likely another car (not shown) would have been selected.

Such errors in prediction are very common because the dispatcher cannot know with certainty about future events. The task of the neural network to be described below is to reduce the error by considering such unanticipated delays in the original estimate of response time.

2.3 Importance of Accurate Response Time Predictions—Early Car Announcement

For the case shown in Figures 5a and 5b, the reader might well argue that the dispatcher should merely switch the assigned hall call on floor 8 to another car, once it is determined that the assigned car will be much later than predicted. In most elevator systems in the world, such assignment switches are done very often. This is acceptable because the passenger waiting in the hallway has not been notified as to which car will arrive.

However the problem is that a relatively new protocol is becoming a standard feature for high profile in Japan and is being offered for sale in the rest of the world. This involves the immediate announcement as to which elevator will serve a floor is made to the waiting passenger immediately upon the passenger's pressing the hall call button. This so called Early Car Announcement (ECA) feature allows the passenger to leisurely walk towards the doorway of the elevator prior to the car's landing.

There are serious implications of Early Car Announcement toward dispatching. With reference to the above example, if an assignment is switched from one car to another--and sometimes to another and another--the waiting passenger might be bouncing from one announced car to another as the assignment is changed during the response time period. For this reason alone, assignment switching is severely discouraged and in some cases forbidden. It thus becomes critical that the first car assignment be the best assignment. This means that the prediction of response time must be very accurate. The neural network technology discussed below provides the needed accuracy in response time prediction.

3. NEURAL NETWORKS

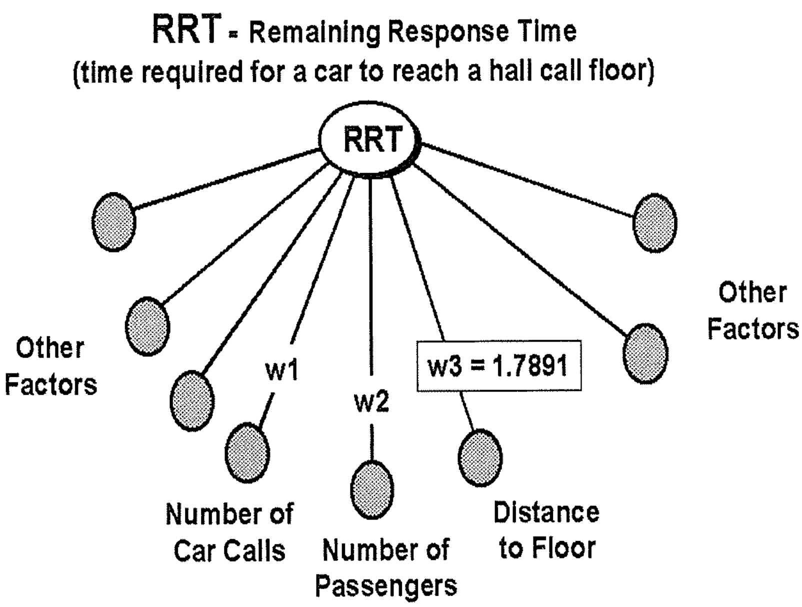
Artificial neural networks are constructed by connecting a number of simple processing elements (called Input nodes) together to perform a calculation. As in the human counterpart ... a network of neurons in the brain ... the power of a network is derived from the connection weights between processing elements. It is the determination of the values of these weights that constitutes the process of learning.

Otis has created a neural network to predict the amount of time that an elevator would require to reach a hall call floor. This new feature is able to learn the characteristics of individual buildings and provide a more accurate measure of RRT than is possible with present methods. In addition to learning how to make better estimates of RRT, it has been shown that the neural network yields better dispatching performance (shorter waiting times) than the current technology.

3.1 Linear Perceptron for RRT

A linear perceptron with 47 inputs and a single output was developed for the prediction of RRT. Figure 6 shows the architecture of this perceptron.

Figure 6



A common metric of performance of a neural network is the absolute error, which is the difference between the estimate from the ANN and the actual RRT. As shown in the table below, the linear perceptron ANN cut the average prediction error by nearly 50%. A good way to visualize this improvement is to plot predicted vs. actual. If the system did a perfect job of prediction, all points would lie on a straight line with slope 1.0, i.e., the line $y = x$. The plot of data points for these two approaches is provided in Figures 7a and 7b.

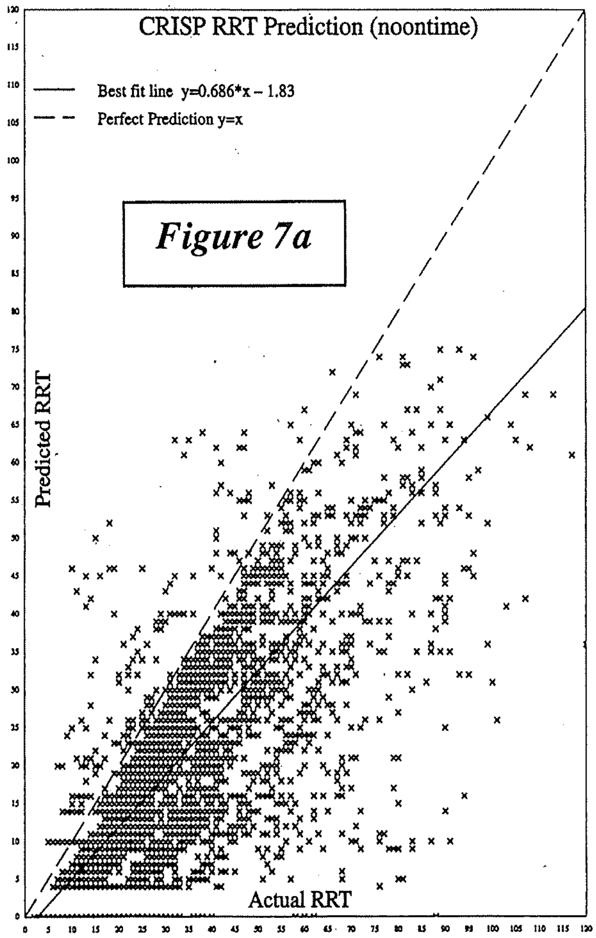


Figure 7a

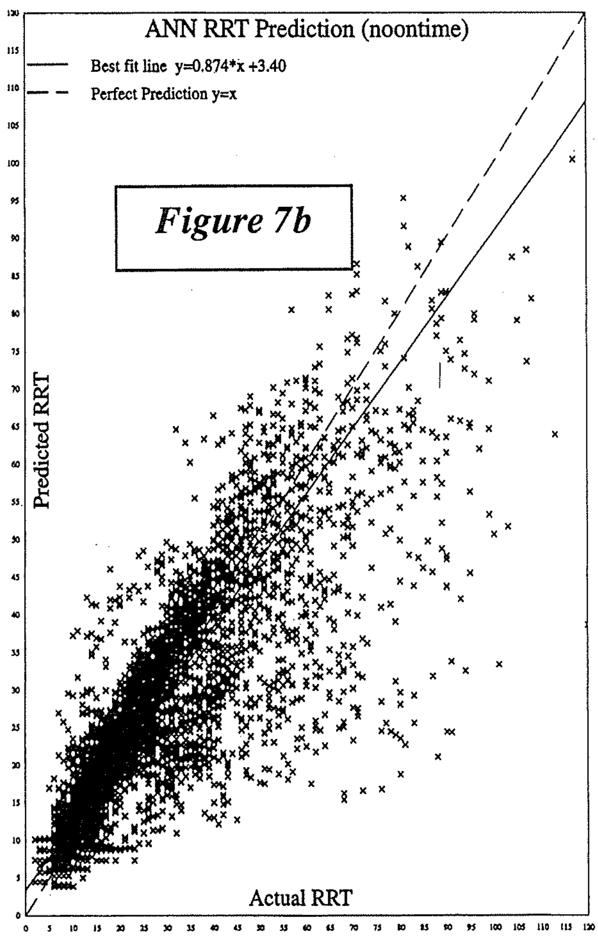


Figure 7b

The greater accuracy of RRT was shown to provide an improvement in dispatching performance, especially in the reduction of long waiting calls. Dispatching with the perceptron was run with three different traffic patterns. The average registration time was improved in each situation: 7.4% for down-peak, 8.9% for noontime, and 20.8% for up-peak. Similar reductions in the frequency of long calls were also obtained.

	Average Absolute Error
Current RRT Approach	9.8 seconds
Linear Perceptron	5.1

3.2 Training the Network

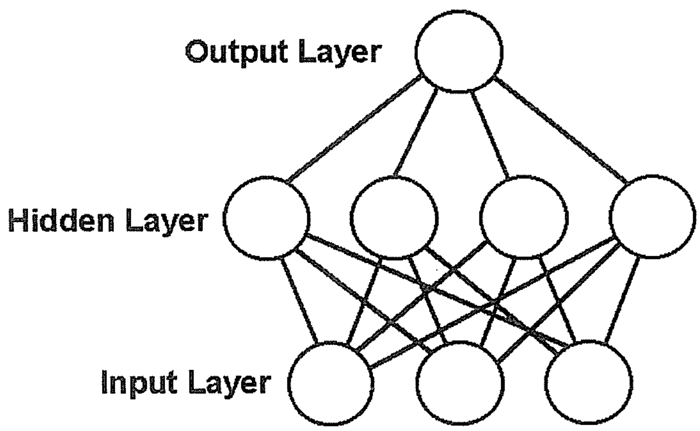
For some network architectures, the connection weights can be learned empirically by training the network on a set of examples that possess desired properties. Single layer networks are effectively weighted sums of the system inputs. For a single layer linear network such as used for RRT prediction, the weight selection process can be considered to be a best hyperplane fitting process.

The quality of the predictions of a neural network is directly related to the nature of the training data. The initial training set examples for predicting RRT were obtained by using the OTISPLAN®simulation program. Two-hour periods were simulated under various traffic conditions and appropriate training data (the state of the system when an assignment was made and the actual RRT determined after the passenger was picked up) recorded. Training took place and optimum values for each of the weights was determined.

The most important part in the design of a good ANN is to determine exactly what information would be useful for input nodes to learn the RRT. Obviously, information about the floor and direction of the new hall calls and state information about a car was important. However, there are hundreds of values that can be used to fully describe the state of an elevator, but the goal was to make the system as small as possible for real-time computational purposes. It was determined that 47 inputs adequately captured the important information. A special representation was designed that allowed the same number of inputs to be used regardless of the number of floors in a building. This feature was patented in 1997.

Field experience with this neural network has been very positive. Once deployed at a job site, a learning procedure was used to provide continuous learning after each completed hall call. Thus the values of the weights reflect the very latest states in building configuration and passenger traffic.

Figure 8



3.3 ANNs with Hidden Units

The perceptron is the simplest of neural networks. The question naturally arises as to how much better and quicker a more complex ANN could learn. A logical extension of the perceptron is the multi-layer ANN with hidden units. Given enough hidden units, an ANN can learn any function and thus should be able to predict RRT better than the linear perceptron. The difficulty in using such networks is determining the appropriate number of hidden units. Too many hidden units can increase the time needed to train the network and decrease the generality of the result. Too few hidden units prohibit the network from making accurate generalizations. The important issue was to determine how much better multi-layer networks could perform than the perceptron.

We studied several network architectures that are variations of the ANN in Figure 8. However, they all had the same 47 nodes in the input layer as the linear perceptron and one node in the output layer. The number of the nodes in the hidden layer varies. A few tests were conducted with networks that have more than one hidden unit layer. In addition to changing the architecture, networks were constructed that use different transfer functions and/or learning techniques. The technical descriptions of each of these different techniques will not be presented here but can be found in [Haykin].

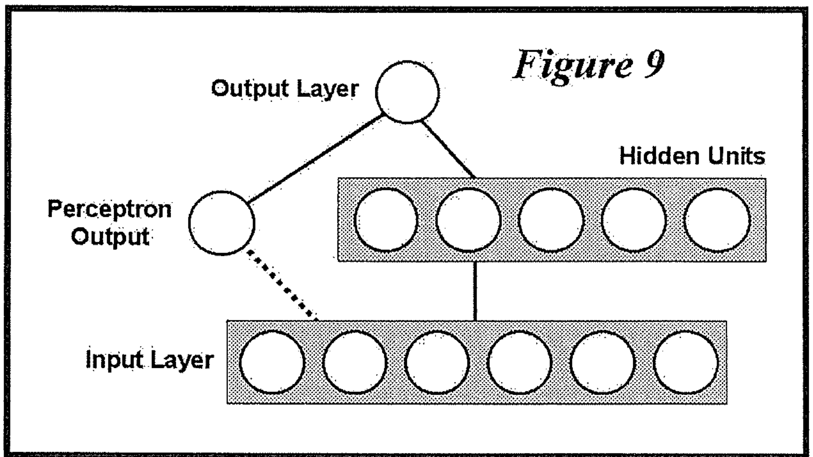


Figure 9

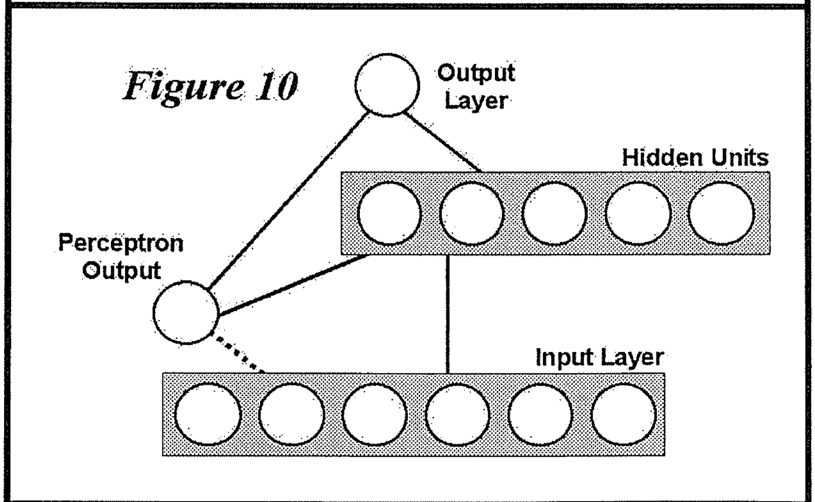


Figure 10

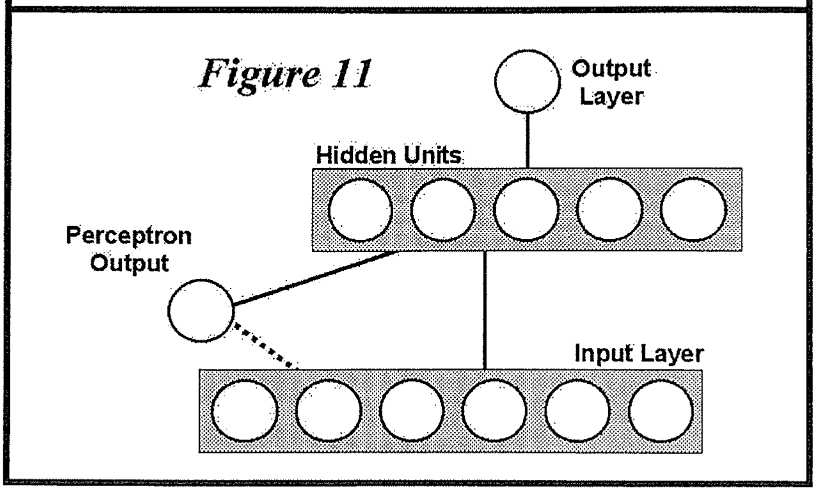


Figure 11

The results of these tests depend a great deal on the number of hidden layers, the number of units in each layer, the transfer function used, and the learning algorithm used. The average absolute error varied from 4.9 seconds to 8.2 seconds, which is about the same as the improvement in estimation with the perceptron. Thus, we found it difficult to provide significantly better accuracy in RRT prediction than the linear perceptron. This was a surprising result.

3.4 Other Special ANNs

Several other significantly different network architectures were considered. They are briefly discussed here.

The first architecture captures the idea of having the hidden units work in conjunction with the linear perceptron. As shown in the architecture overview of Figure 9, the precariously learned perceptron is put into the network. The dashed lines indicate weights that are fixed and are not trained during the learning phase of the system.

The second architecture is built upon the idea of using the perceptron output as an input to the hidden units that can be corrected. Figure 10 shows this architecture.

The third architecture removes the output of the linear perceptron from the network output node and treats it as just another input into the hidden layer. This architecture is illustrated in Figure 11. This network was only tried with five hidden units but allowed to train for different numbers.

Although we were able to show better results than those obtained with the previously discussed networks, these more complex networks were not implemented into the system. In any case, we have already shown that a improvement in error prediction of one second or less will not yield significant improvement in dispatching performance.

4. CONCLUSION

When developing applications using neural networks, prediction accuracy is not the only consideration. Although some of the more complex networks had (slightly) better prediction accuracy, the added complexity involved in creating these networks and putting them in production could not be justified on the expected gains in dispatching performance.

In this paper we have shown how neural networks can be applied to improve commercial products. ANNs can be used to get better predictions of RRT for elevator dispatching, and this in turn can improve the overall dispatcher's performance. We have also demonstrated that more accurate and complex ANNs do not necessarily provide significant improvements over less complicated techniques.

5. REFERENCES

Haykin, S., Neural Networks A Comprehensive Formulation, IEEE Press (Macmillan College Publishing Company, Inc.) 1994.

Powell, B. A., "Elevator Banking for High Rise Buildings," in Transportation Science, Vol. 11, No. 2, pp. 200-210, 1975.

Powell, B. A., "Mathematical Modeling In Elevator Systems," in Case Studies In Mathematical Modeling, editor W. E. Boyce, Pitman Publishing, 1981.

Powell, B. A., and D. J. Sirag Jr., "Fuzzy Logic: A New Way of Thinking About the Complexities of Dispatching Elevators," Elevator World, Vol. 41, No. 9, pp. 78-84, 1993.

Whitehall, B. L., D. J. Sirag, Jr., and B. A. Powell, "Elevator Control Neural Network," U.S. Patent No. 5,672,853, September 30, 1997.

6. BIOGRAPHIES

Bruce A. Powell is an Otis Technical Fellow at Otis Elevator Company's Engineering Center in Farmington, CT, USA. He and his team are responsible for the application of advanced technology to the dispatching of elevators. He earned a BS in Mathematics from Denison University and an MS and Ph.D. in Operations Research from Case Western Reserve University. He has spent more than 30 years applying mathematical optimization and simulation to elevator systems and has numerous publications and some 34 issued patents.

David J. Sirag, Jr., is Manager, Systems & Software at United Technologies Research Center. He earned a BS in Information and Computer Science from the University of California, Irvine and an MS in Computer and Communication Science from the University of Michigan. He spent more than ten years developing Machine Learning and Expert System applications in UTRC's Artificial Intelligence group, including advanced elevator dispatching logic now being used in many of the world's landmark buildings.

Bradley L. Whitehall is Senior Vice President, Knowledge Systems Technologies, at Information Resources, Inc. He has a BS from Northern Illinois University and an MS and Ph.D. in Computer Science from the University of Illinois. He has more than 50 publications in the area of Artificial Intelligence and application of AI techniques and 4 patents. Over a period of five years while working for United Technologies Research Center, he focused on applying AI and Neural Network technologies to elevator dispatching problems.