REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
May 1990	Conference paper	

4. TITLE AND SUBTITLE

See report.

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

See report.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

See report.

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

See report.

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Distribution Statement A - Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

Presented at the IEEE 1990 National Aerospace and Electronics Conference (NAECON 1990) held in Dayton, Ohio, on 21-25 May 1990.

14. ABSTRACT

See report.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	UU		**19b. TELEPHONE NUMBER** *(include area code)*
Unclassified	Unclassified	Unclassified			

NEURAL NETWORK BASED HUMAN PERFORMANCE MODELING

Capt Edward Fix
US Air Force
Harry G. Armstrong
Aerospace Medical Research Laboratory

Abstract

Neural networks provide an alternative method of building models of human performance. They can learn behavior from examples, reducing the need for many identical repetitions and intensive analysis. A properly trained net can be very robust in its response to a novel stimulus. This opens the door to modeling performance in the presence of an interactive stimulus. Neural networks provide the possibility of robust models that can operate interactively in real time, depending on the size and architecture of the net and the application.

A neural network architecture derived from recurrent back propagation is presented which learns to mimic human behavior and performance in a sample task. It shows operating characteristics similar to those of human subjects, and even makes the same kinds of mistakes. Possible applications are discussed.

1 Introduction

This paper presents a new technique for modeling human performance based on neural networks. Neural nets have several characteristics that lend themselves to modeling. They learn from examples, which expands the types of data that can be used for model building. Many repetitions of a standard set of tasks is not required. They can interpolate between, and extrapolate somewhat beyond the training examples, which greatly reduces the manual analysis required. A properly trained net can emulate the training examples to virtually any degree of accuracy, make the same kinds of mistakes, and show uncertainty at appropriate times.

The Manned Threat Quantification program at the Aerospace Medical Research Laboratory's Human Engineering Division has been involved with modeling human operation of air defense systems for more than 15 years. It has developed human operator models for many purposes including stand-alone air defense system models and integration into existing models. MTQ models are unique in that they involve human models derived from real-time man-in-the-loop simulations using realistic operating positions. These models have been used for many purposes including tactics development and survivability analyses.

Although there are many different techniques for modeling and emulating human performance, and each technique has its own strengths and applications, there are some weaknesses shared by all. The analysis that goes into building a human operator model is very intensive and expensive. The data upon which the model is based is a statistical representation of a human's performance through many repetitions of a set of tasks. This means the model is actually only correct for the exact tasks in the statistical base, and its applicability to other conditions is only as good as the manual analysis that went into the model. Finally, standard models do not handle uncertainty well. Under conditions where a human operator might make a mistake, the model may not make a mistake, or it may make an inappropriate mistake.

The goal of this task was to develop a neural network which, when trained on data derived from a human subject performing a task, emulates that subject's performance and style. The task was to be interactive, and the conditions uncontrolled. In an environment like this, standard modeling techniques are difficult to apply at best.

2 Experimental Task

The task the subjects performed was based on a popular neural network demonstration concept (Restrepo). It is a computer generated display showing a two-lane circular track and five "cars" (Fig 1). One car is controlled by the subject, and the other four

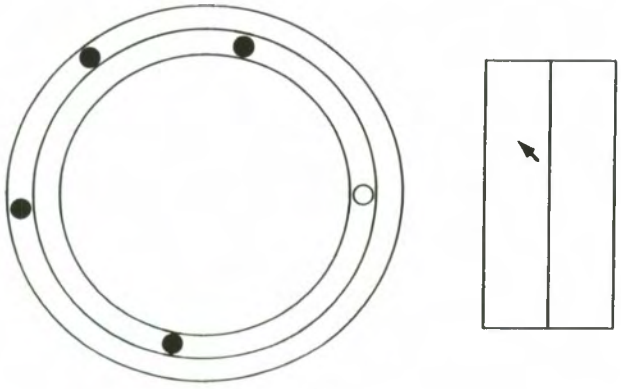

Figure 1: Experimental Task Presentation Screen

are controlled by the computer. The cars all travel in a counter clockwise direction. The perspective is adjusted so that the subject's car is always at the 3 o'clock position on the track, and everything else moves relative to the controlled car.

The subject's task was to drive his car around the track, switching lanes and adjusting speed as necessary to avoid collisions. Although the subject was not instructed to pass the other cars, there is a score presentation that increments one point each time the subject passes another car, and decrements by three whenever the subject bumps another car. The instructions were vague in order to elicit the as many driving techniques as possible from the subjects.

The subject controls his car through a mouse interface. The mouse pointer moves around on a "control panel". Putting the pointer in the left half of the panel puts the car in the left (inside) lane, and the right half puts it into the right lane. Moving the mouse to the top of the panel accelerates the car to full speed, and moving it to the bottom stops the car.

2.1 Performance

There are two different scenarios. In one scenario, called "Variable Cars", the other cars start at random positions on the track, two in each lane, traveling at random speeds. At infrequent, random times the cars change speed, and if one car approaches another from behind, it switches lanes to pass. In the second scenario, called "Hostile Cars", the computer controlled cars change speeds and lanes as in the first, but in addition they actively try to prevent the subject from passing. They speed up as the subject approaches, and switch lanes, or match speeds with a nearby car in the other lane to block the subject.

Driver training for the subjects was accomplished by exposing the subject to increasing levels of difficulty. The subject first ran a version where the other cars never change speed or lane until he felt comfortable with controlling the simulation. When the subject was consistently passing the other cars without collisions, the demonstration program was stopped and the "Variable Cars" program started. The subject was given as much time as necessary to become accustomed to the new situation, and then data was gathered for 4 minutes. This was repeated for the "Hostile Cars" case.

2.2 Data

While the subject was performing the task, the position, lane, and speed of each car, including the subject's car, were recorded each time the screen animation was updated. After a net was trained, it was tested by presenting a novel starting position and letting it control the simulation, and its output was gathered in the same way. Then the data from both the subject and net were analyzed for operating style. Parameters gathered included the distance the subject or net was behind the nearest car in the same lane when it switched lanes to pass (D_L), the closest approach to the nearest car ahead in the same lane when the controlled car slowed enough to allow the computer car to pull further ahead (D_B), the distance between cars that were close ahead and close behind in opposite lanes when the controlled car switched lanes to pass between them (D_C), and number of cars passed versus number of collisions.

In the case of D_L, D_B, and D_C, there are many examples in the data from both the subject and net. These can be statistically tested for significant differences.

3 Modeling Network Architecture

The basic architecture of the net is a feed-forward net with two active layers as described in Rumelhart, et al. However, a simple feed forward net trained by back propagation could not handle the complexities of emulating the behavior of human subjects on this task. The biggest problem was that while the net learned behavior it saw frequently, it had problems with rare behavior. Another limitation of feed forward nets trained by back propagation is that they have no time sense. This made it impossible to emulate reaction time. Following is a description of the architecture and training used to overcome these problems. The net has 42 input nodes, illustrated

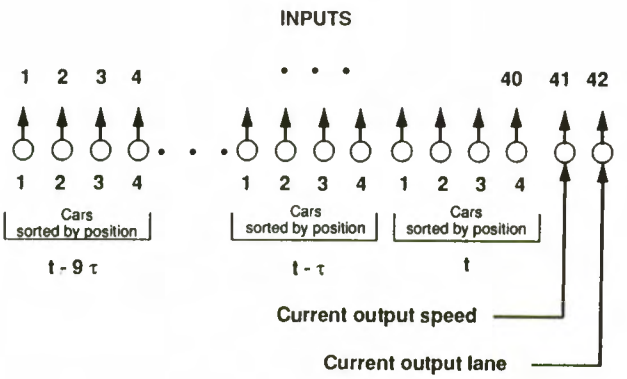

Figure 2: Presentation of Input Parameters

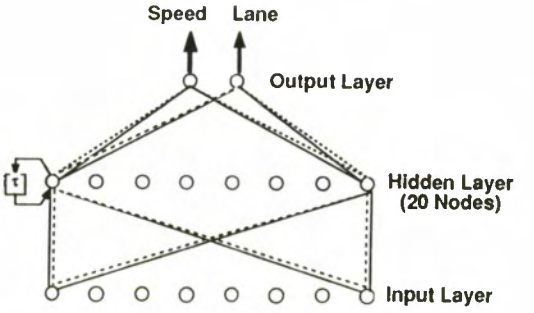

Figure 3: Network Structure

in Fig 2. The position values of the four computer controlled cars are sorted by position relative to the subject- or net- controlled car. That is, input 1 is the nearest car in the inside lane, 2 is the nearest car in the outside lane, 3 is the farthest car in the inside lane, and 4 is the farthest in the outside lane. This input format is presented for the latest 10 time slices to give the pattern over recent time. This allows the net to discern relative velocity in much the same way that a human would. The final two input values are the present value of the subject's car's speed and lane. The hidden layer contains 20 nodes, and the output layer contains 2 nodes; one is a continuous value representing speed and the other represents lane with the value either 0 or 1 (inside or outside lane.)

Two major modifications were made to the standard feed-forward net architecture described in Rumelhart. First, recursion was used in the hidden layer to introduce a time delay to emulate reaction time. This architecture caused the net to behave with realistic reaction time, and sometimes a rapid change would overshoot the target value, just as would be observed in human performance.

The second modification to the net architecture was made necessary by the net's continuing failure to recognize and treat rare occurrences adequately. It was discovered that subjects tended to act quite differently in two different circumstances. In the case where the other cars were spaced out ahead of the subject car, the driver tended to drive full speed and merely switch lanes to avoid collisions. This occurred most of the time. However, when there were cars in both lanes ahead of the subject's car, with insufficient room to pass between them, the subject would slow down to their speed or less and wait for them to separate. The subject would sometimes abruptly slow too much and overshoot the desired slower speed. This was especially true in the case where the other cars were actively working against the subject. Training on the whole subject data set, the net was unable to emulate this characteristic.

The solution was to use two redundant sets of feed forward weights; in effect use two nets with a common recursive connection (Fig 3). One set of weights emulates behavior in the frequent condition and the other in the rare condition. This architecture is essentially a neural network set controlled by a simple expert system. Arrangements of this type have been described before (Holden), and appear to combine the strengths of neural and expert systems.

4 Results

A net was considered successfully trained if there was no significant difference between the data gathered from a subject and that from the net trained on that subject. In addition, data from the subjects and nets could be played back in the original visual format and contrasted subjectively with each other. The first time this was accomplished with a successfully trained net, the experimenter spent considerable time checking to make sure the correct file was being displayed, until he realized that his uncertainty indicated the program was passing the Turing test.

After successfully training this network architecture to mimic three subjects, several preliminary results can be reported. In general, only those parameters where there is a significant difference between subjects (at the p=0.05 level) are considered. In the case of the hostile cars, there was a significant difference in mean D_L between subject A and both B and C, and the respective nets showed similar differences. In the case of variable cars, there was significant difference in mean D_B of subjects A and B, again with

similar difference between the nets trained from them.

Hostile Cars, D_L

	Subject Data			Net Data		
Subject	$\overline{X}$	SD	N	$\overline{X}$	SD	N
A	.099	.140	179	.080	.093	118
B	.168	.208	86	.134	.170	90
C	.189	.226	126	.176	.272	66

Variable Cars, D_B

	Subject Data			Net Data		
Subject	$\overline{X}$	SD	N	$\overline{X}$	SD	N
A	.093	.069	40	.073	.054	32
B	.158	.062	26	.143	.068	33

5 Applications

This type of model could be used where realistic behavior is required from a simulation of a human operated system. An example would be the use of intelligent adversaries in simulators for training military aircrews. It would be desirable for simulations of human-operated threats to act like the real thing, to give the aircrews realistic training as if they were working against actual people. This is especially pertinent to the modeling of ground-based air defense systems.

It might also be useful for modeling operations where multiple repetitions of a situation are not possible, making standard statistical modeling techniques impractical. Data could conceivably be taken from almost any source and used to build a model, with the requirement that it be broadly representative of the overall operation. Laboratory measurements would not be necessary, and control of conditions becomes less important.

Finally, the amount of analysis required for building a model is much less than for other techniques. This means that less human labor may be required, lowering the cost of building a model. Models of human operation may become practical in applications where they have always been desirable, but prohibitively expensive.

In the future, MTQ will use this technique to model a human-operated SAM system for insertion into an aircrew training-type simulator. Existing data is being used to refine the specific network architecture, and data is being gathered from linked SAM and aircraft simulators. This refined technique could then be used to transform data from training ranges (Red Flag, Navy Fallon, etc.), for instance, into models to control simulations or unmanned threat emitters on the ranges.

References

Almeida, Luis B., A Learning Rule for Asynchronous Perceptrons with Feedback in a Combinatorial Environment, Proc IEEE First Annual International Conference on Neural Networks, 609-618, 1987

Holden, Alistair D.C., Fast Learning in Symbolic/Neural Models using External Constraints and Automatic Re-Structuring, Proc IEEE Systems, Man, and Cybernetics, **1**, 2-4, 1989.

Pineda, Fernando J., Generalization of Back-Propagation to Recurrent Neural Networks, Physical Review Letters, **59** (19), 2229, 1987

Restrepo, Pedro, Personal communication, Alphatech, Inc., 1988

Rumelhart, D.E., G.E. Hinton, and R.J. Williams, Learning Internal Representations by Error Propagation. In D. E. Rumelhart and J.L. McClelland (Eds.), *Parallel Distributed Processing Volume 1: Foundations* (pp. 318-362). Cambridge, MA: MIT Press/Bradford, 1986.