Summer 2004

Modeling Launch Vehicle Success Using Artificial Neural Networks

Jennifer A. Schuck
Embry-Riddle Aeronautical University - Daytona Beach

Modeling Launch Vehicle Success Using Artificial Neural Networks

By

Jennifer A. Schuck
B.S., Embry-Riddle Aeronautical University, 2002

A Thesis Submitted to the
Department of Human Factors and Systems
In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Human Factors and Systems

Embry-Riddle Aeronautical University
Daytona Beach, Florida
Summer 2004

MODELING LAUNCH VEHICLE SUCCESS USING ARTIFICIAL NEURAL NETWORKS

by

Jennifer A. Schuck

This thesis was prepared under the direction of the candidate's thesis committee chair, Linda Trocine, Ph.D., Department of Human Factors and Systems, and has been approved by the members of the thesis committee. It was submitted to the Department of Human Factors and Systems and has been accepted in partial fulfillment of the requirements for the degree of Master of Science in Human Factors and Systems.

THESIS COMMITTEE:

Linda Trocine

Linda Trocine, Ph.D., Chair

Shawn M Doherty

Shawn Doherty, Ph.D., Member

Mehmet Sozen

Mehmet Sozen, Ph.D., Member

Shawn M Doherty

MS HFS Program Coordinator

Frances A Greene

Department Chair, Department of Human Factors and Systems

Associate Chancellor for Academic Affairs

Acknowledgements

I am indebted to many people who have given their priceless time and energy to helping me complete this project. First of all I would like to thank my committee. You have assisted me in creating a document that I am very proud of. I especially would like to thank my committee chair, Linda Trocine, Ph.D. for driving me to do my best and for being a great mentor and friend.

I also want to send my appreciation to all of the faculty and staff of the Human Factors Department. You do a great job! Thanks for creating such a supportive environment for us graduate students.

I think the unsung heroes on the campus of Embry-Riddle are the members of the library staff. I would not have survived this project without the assistance of the Interlibrary Loan office.

Additionally there were a couple of men outside the university that helped me along the way. Thank you to Martin Steele, Ph.D. for assisting me in my initial research and to William Brown at the National Climatic Data Center for gathering all the weather data for me.

Of course I do need to acknowledge the people that had to put up with me during this whole project—my family and friends. Thank you for all your support, patience, and love.

Abstract

Expendable launch vehicles in the United States currently have a reliability of 92%. The failures that do occur cost millions of dollars in spacecraft replacement, lost revenue, and other expenses. These costs are passed on in higher insurance rates and launch vehicle price. If the launch outcome of the launch vehicles could be better predicted, the overall cost of launching payloads into space would decrease. This study used artificial neural networks to model the overall launch outcome of a launch vehicle so that the results of a launch could be predicted. Two neural network architectures—MLP and fuzzy ARTMAP--were trained on historical launch data of Atlas, Delta, and Titan launch vehicles. The networks were then tested on their ability to generalize to new data. Fuzzy ARTMAP performed slightly better than MLP overall, but neither network can be used during launch countdown today. Future application of the networks in real-time during the vehicle launch countdown will require the use of more launch specific data.

Table of Contents

Acknowledgements iii
Abstract iv
List of Tables vi
List of Figures vii
List of Abbreviations viii
Introduction 1
Neural Networks for Launch Outcome Prediction 6
  Neural Networks for Reliability 6
  Neural Network Overview 8
  Training 18
  Launch Prediction 19
Methods 21
  Apparatus 21
  Data 21
  Procedure 23
Results 28
Conclusions 31
References 34
Appendix 36

List of Tables

Table 1.	Enumeration of Launch Expenditures and their Associated Costs	2
Table 2.	Use of MLP Neural Networks in Classifying Reliability Prediction	6
Table 3.	Comparison of the Performance of Several Neural Network Architectures	10
Table 4.	Comparison of the Characteristics of MLP and Fuzzy ARTMAP	18
Table 5.	Factors Affecting Launch Vehicle Reliability	21
Table 6.	MLP Configuration for the Previously Discussed Studies	24
Table 7.	Percentage of Correctly Classified Launches from the Neural Network Models	28
Table 8.	Percentage of Launches Classified in each Category during the Testing Phase	29
Table 9.	Index to the Following Launch Vehicle Data	36
Table 10.	Complete Launch Vehicle Data	38

List of Figures

Figure 1.	Model of a NN neuron	11
Figure 2.	Model of the multi-layer neural network architecture	13
Figure 3.	Model of the fuzzy ARTMAP architecture	15
Figure 4.	Geometric view of the output patterns as developed during fuzzy ARTMAP training	16
Figure 5.	Bar graph indicating the frequency of use of each number of epochs by fuzzy ARTMAP and MLP	30

List of Abbreviations

ART	Adaptive Resonance Theory
LVQ	Learning Vector Quantization
MATLAB	Matrix Laboratory
MLP	Multi-layer Perceptron
NN	Neural Network
PC	Personal Computer
RBF	Radial Basis Function

Introduction

Expendable launch vehicles, rockets used to launch satellites into space, constitute a large amount of business in the United States. Every year, dozens of payloads or satellites are launched into Earth orbit. Some of the payloads are commercial. These satellites facilitate cell phone signals, Global Positioning System navigation, television broadcasting, and countless other world-wide communication links. Other launch vehicle payloads are classified military projects that help to keep our country safe. Still other payloads consist of scientific packages. These satellites study the Earth, our Solar System, and beyond.

Each launch vehicle payload is the result of years of research, development, manufacturing, and testing. This process constitutes a large commitment on the part of the satellite manufacturer and typically costs hundreds of millions of dollars (Chang, 2000). The satellite makers sink large amounts of time and energy into creating their product—one they are counting on bringing a return back to their company.

In addition to development and manufacturing cost, the actual launch of the payload is very expensive. The current price for launching one pound of payload into Earth orbit is $5,000 (Fragola, 1991). Considering most satellites are a few thousand pounds in weight, launches can run into the range of tens of millions of dollars. For example, a common commercial communications satellite has a mass of 9,480 pounds (Hill, 2000). This would incur an approximate launch cost of $47 million.

It is important to the launch vehicle customer that their payload reaches its destination safely. It is also important to the maker of the launch vehicle that the process comes to a successful completion. Just like the satellites they transport, launch vehicles undergo years of development and testing before they are prepared for use. Any launch failures would be a setback to both the payload manufacturer and the launch vehicle company.

The current United States launch vehicle failure rate is 8% (Fragola, 1991). The consequences of these failures, listed in Table 1, are great. In addition to the monetary risk discussed above, there are several other risks associated with launch vehicle failures.

Table 1
Enumeration of Launch Expenditures and their Associated Costs (from Parkinson, 1998)

Launch Expenditure	**Typical Cost**
Low Earth Orbit Payload	$88 million
Launch Cost	$90 million
Insurance	$18 million
Cost of Failure	$296 million
Lost Business	$79 million

Parkinson (1998) lists the total cost consequences of a vehicle failure. They include paying insurance for the lost payload, the cost of a new launch, the cost of an investigation and recovery, the cost of system maintenance during downtime, and the cost of lost opportunity. These costs are shared between the insurance agency, the launch operator, and the satellite manufacturer.

Investigations into the cause of a launch vehicle failure cause downtime. Launch activities cannot resume until a cause is determined and the problem is fixed. This investigation process may result in schedule delays of several years (Pytanowski, 1999).

Launch vehicle success also affects future business. One factor that is considered when choosing a launch vehicle to raise a payload into orbit is the reliability of the vehicle. If one launch vehicle has been receiving bad publicity because of failures, the vehicle's maker may lose business until the reliability improves (Parkinson, 1998).

Failures, and therefore lower vehicle reliability, add cost to the rest of the launches. Vehicle operators increase the price of launches to cover the costs they ensue from other failures. This cost of risk adds about 18% to the cost of a launch (Parkinson, 1998).

Additionally, there is also a risk to life on the ground if a launch vehicle fails during lift-off. The fuels used to propel launch vehicles into space are dangerous to the environment and to humans. Clean-up after a failure is an extensive and expensive process. Danger is somewhat mitigated by the placement of the two launch facilities in the United States. The Cape Canaveral launch site in Florida launches vehicles over the Atlantic Ocean, and the Vandenberg Air Station in California launches vehicles over the southwestern desert. However, winds may carry toxic fumes to populated areas. Besides launch site placement, there are many other safety considerations in place during launch. Contingency plans are ready in the event of a failure.

For all the reasons stated above, it is important to better predict the outcome of a launch than it is to recover from it afterwards. If a failure can be prevented, time, energy, and money will be saved. There are many factors that contribute to the success or failure of a launch—the weather during launch, the type of engine, the reliability of the engine, the reliability of the internal components of the vehicle, and many others.

In the United States today, there are three major launch vehicles: Delta operated by The Boeing Company and Atlas and Titan operated by Lockheed Martin Astronautics. There are currently three Delta vehicles being used: Delta II, III, and IV. Delta II has been in service since February 1989. The Delta III and IV vehicles are more recent additions to the fleet (Launch Vehicles, 2004). On average, there have been 8 Delta vehicle launches per year since 1989 (Boeing, 2004). Lockheed Martin operates both the Atlas and Titan launch vehicles. The Atlas vehicles presently in use are the IIAS, III, and V. The first launch of the IIAS occurred on December 15, 1993. Titan II and IV are currently active. Titan II has been in service since 1964, and Titan IV was first launched in 1989 (Launch Vehicles, 2004). Altogether there have been 329 Atlas, Delta, and Titan launches since 1979.

The objective of this study was to build a model to predict launch vehicle success. Statistical modeling methods were eliminated because the data are not independent and identically distributed. Neural networks have been used for similar problems in the past, so were selected for this application.

This study was intended to demonstrate whether artificial neural networks are useful for modeling launch outcome. In more general terms, the neural networks modeled the reliability of the launch vehicles. Reliability in this paper is defined as the ability of a launch vehicle to reach orbit without destruction of the vehicle.

Research was done to see if launch outcome could be predicted with an artificial neural network, and how well the network could model launch outcome. If successful, neural network modeling could be added to existing preflight checks as an additional measure of launch safety. Currently, a controller must use information from several sources to decide on a "Go" or "No Go" for launch. A neural network model would be able to sort through all of the controller's information to determine the outcome of the impending launch. The controller would be able to make a better decision having to analyze only one piece of computer output rather than dozens.

The ability to accurately predict launch outcome would save money due to lost payloads. Launch failures would be avoided as would the impact of those failures. Humans and the environment would be protected by avoiding fuel and payload debris from contaminating the earth. Finally, it would avoid the need to clean up hazardous materials.

Neural networks are defined in the next section. Also detailed are how they have been used to predict reliability and what has been done in the past to predict launch vehicle success.

Neural Networks for Launch Outcome Prediction

This chapter discusses neural network literature relevant to this reliability problem. First, recent studies using neural networks to model reliability will be outlined. Second, details about neural networks in general will be discussed. Finally, studies concerning the reliability of launch vehicles will be described.

Neural Networks for Reliability

Neural networks can be used to predict the reliability of systems. In the terms of this paper, reliability is the success rate of launch vehicles. Table 2 lists some of the research that has been done on modeling reliability using neural networks. The literature shows that neural networks are consistently useful for complex reliability problems that may not be solved using statistical methods. Neural networks take large amounts of data and quickly find solutions to a variety of types of problems.

Table 2
Use of MLP Neural Networks in Classifying Reliability Prediction

Author/Date	**Application**	**Scope**	**Success Rate of Prediction**
Adnan, W.A., Yaakob, M., Anas, R., and Tamjis, M.R. (2000)	software	overall system	98 %
Amjady, N. and Ehsan, M. (1999)	power systems	overall system	99 %
Chinnam, R.B. (1997)	drill bits	individual component	time dependent
Coit, D.W. and Smith, A.E. (1995)	genetic algorithms	overall system	99.5 %
Hiebert, S.F. and Chinnam, R.B. (2000)	drill bits	individual component	time dependent
Khaparde, S.A. and Bhattacharyya, K. (1996)	power systems	overall system	99 %
Sinha, S.K. and Pandey, M.D. (2002)	oil and gas pipelines	overall system	89 %

In addition to being powerful and fast, neural networks can be applied to find either the reliability of overall systems or of individual parts. As an example of an overall system view, Sinha and Pandey (2002) studied the reliability of oil and gas pipelines. The neural network utilized eight attributes that were collected during an inspection of the pipeline. The neural network model estimated the probability of pipeline failure based on the data from the inspection. The probability of failure output was categorized into one of five ranges depending on the severity of the probability of failure. The model accurately predicted pipeline failure 89% of the time.

Coit and Smith (1995) also focused on overall system reliability. They found that neural networks were useful in estimating overall system reliability of genetic algorithms based on individual component reliability and design configuration. The resulting neural network correctly classified system reliability 99.5% of the time.

A study by Adnan, et al. (2000) examined the reliability of different types of software. Individually, software such as on-line data entry and flight dynamic applications was modeled. Overall, the neural network models correctly predicted reliability 98% of the time.

Amjady and Ehsan (1999) studied the overall reliability of electrical transmission systems. They found that the neural networks modeled reliability correctly 99% of the time. The authors used estimations of scheduled maintenance to model the systems. Khaparde and Bhattacharyya (1996) also modeled electrical generator systems and found a correct prediction rate of 99% as well.

Two other studies by Chinnam (1997) and Hiebert and Chinnam (2000) stressed the fact that neural networks can be used to analyze one individual component's reliability. The studies focused on individual drill bits. They proposed that end users are interested in their component's reliability, not the average characteristics of an entire batch. The authors used degradation signals from individual drill bits to model the performance reliability of that drill bit. The percent accuracy of reliability prediction of the models increased as the number of holes drilled with the bit increased.

The wide range of reliability problems that have been addressed using neural networks is promising. Appling neural networks to launch outcome requires looking at the overall system and choosing appropriate attributes in the neural network model of the system. Next, a background on neural networks is provided.

Neural Network Overview

Artificial neural networks have existed for sixty years (Hagan, Demuth, and Beale, 1996). Their widespread use, however, has just in the past few years begun to flourish. Within the neural network domain, there are several architectures that can be used for different types of applications. These architectures are used for a variety of learning tasks. Some architectures are more suited for some tasks than others.

Each neural network is taught to perform a specific task. These tasks can be applied to a wide range of problems in a wide spectrum of fields. The following list details neural network tasks (Christodoulou and Georgiopoulos, 2001):

- Approximation—estimate a function given a set of x and y data points.
- Pattern classification—fit the input patterns into a fixed number of categories.
- Prediction—predict present samples given past samples.
- Clustering—group data with common features into categories.

This paper focuses on pattern classification. Input and output pairs of data with known outcomes were applied to the neural networks in order to train them. These input output pairs are referred to as training patterns. After they were trained with the training patterns, the trained network was used to categorize new input patterns. These new input patterns were not part of the training patterns. These have known outcomes to the modeler and are used to test how well the network performs on novel data. These are referred to as test patterns. The launches were classified into either a success or failure category.

Each input pattern is made up of attributes that define that pattern. Some input attributes may have a relatively low correlation to launch outcome, and other input data may have a high correlation. It is not necessary for the researcher to know which attributes are more relevant to launch outcome. The neural network will assign weights to each attribute according to its effect on the outcome.

The best neural network architectures for pattern classification problems are multi-layer perceptron (MLP) and fuzzy ARTMAP (Christodoulou and Georgiopoulos, 2001). Multi-layer perceptron networks are feed-forward networks that are trained with back propagation algorithms and are the most widely used neural networks. Fuzzy ARTMAP networks are newer and thus less widely known. They are based on adaptive resonance theory. Table 3 lists studies that have compared the performance of MLP and fuzzy ARTMAP to each other and to other network architectures.

Table 3
Comparison of the Performance of Several Neural Network Architectures

Authors	**Architectures Studied**	**Success Rate Training Set**	**Success Rate Test Set**
Meneganti, Saviello, and Tagliaferri, (1998)	Fuzzy ARTMAP	100 %	69.7 %
	Fuzzy Basis Functions	65.4 %	59.8 %
	Adaptive Optimal Fuzzy Logic System	70.6 %	67.0 %
	Quasi-Newton Multilayer Perceptron	81.9 %	72.1 %
Llobet, et al., (1999)	Fuzzy ARTMAP	---	90.3 %
	Learning Vector Quantization	---	92.0 %
	MLP	---	82.4 %
Sinha and Pandey, (2002)	Custom Modified Probabilistic Neural Network	96.7 %	91.2 %
	MLP	89.2 %	84.5 %
	General Regression	81.9 %	77.3 %
	Radial Basis Function (RBF)	85.7 %	81.1 %
Trocine, (2002)	MLP	86.1 %	83.6 %
	RBF	100 %	45 %
	Fuzzy ARTMAP	100 %	98.2 %

Meneganti, et al. (1998) compared four architectures. Fuzzy ARTMAP and MLP outperformed Fuzzy Basis Functions and Adaptive Optimal Fuzzy Logic System. Trocine (2002) also found that fuzzy ARTMAP and MLP outperformed another architecture, Radial Basis Function neural networks. In the study by Sinha and Pandey (2002), only their custom-designed Probabilistic Neural Network outperformed MLP. Fuzzy ARTMAP was not included in the study.

Llobet, et al. (1999) found that both fuzzy ARTMAP and Learning Vector Quantization (LVQ) classified better than MLP. LVQ performed well for that application, but currently there are no other examples comparing LVQ to fuzzy ARTMAP. Future studies may focus on this gap in research, however, that question was not considered for this study.

Overall, MLP and fuzzy ARTMAP are considered the best architectures for pattern classification tasks. The next section provides an overview of these two network architectures.

The multi-layer perceptron (MLP) network is a feed-forward network. Feed-forward means the signals between neurons only go from one layer to a higher index layer, not sideways or backwards. Synapses are the connections between individual neurons and each synapse has a different weight associated with it. Figure 1 shows one neuron with three synapses connecting to it. The sum of the weighted signals from the synapses must cross a threshold in order to generate an output from the neuron. Multi-layer means that there are one or more hidden layers between the input and output layers as shown in Figure 2.

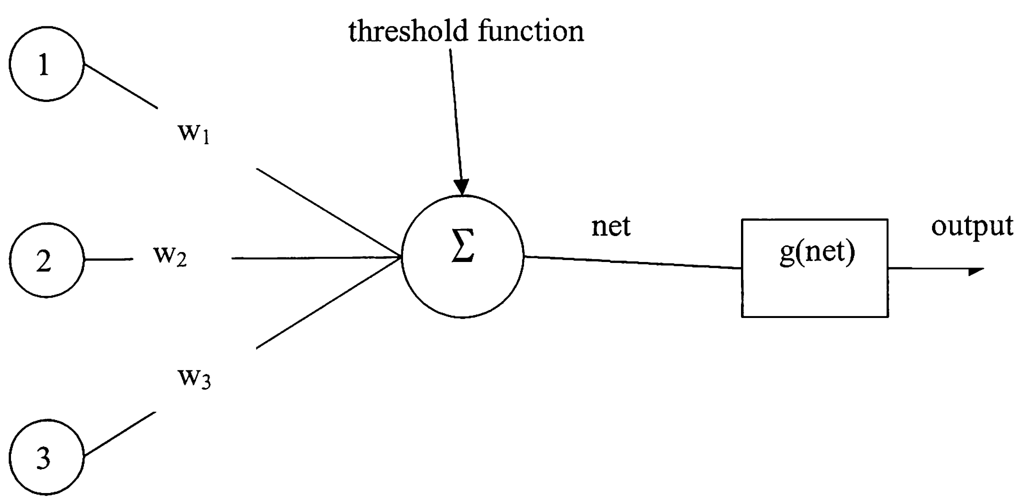

Figure 1: Model of a NN neuron (from Christodoulou and Georgiopoulos, 2001). The neuron receives weighted signals w_1, w_2, and w_3 from nodes 1 through 3 via the synapses. The sum of the input signals must pass some threshold function in order to pass through the neuron. If the threshold is met, the net signal is acted upon by some predetermined function and passed on as output.

There are several transfer functions available for use in MLP networks. Transfer functions are learning rules that are used to get the neuron input/output relationship to meet a specific goal. Specific functions are chosen for the type of problem that needs to be solved. For the model created in this study, the Hyperbolic Tangent Sigmoid (tansig) and Linear (purelin) functions were used, the latter for the transfer from the input layer to the hidden layer and the former for the transfer from the hidden layer to the output layer. The linear transfer function is exactly how it sounds, linear. The output is equal to the input (Hagan, et al., 1996):

$$a = \mathrm{purelin}(n)$$

The tansig transfer function is a type of sigmoid function. Sigmoid functions are the most common activation functions used (Christodoulou and Georgiopoulos, 2001). The function has a range from 0 to +1, however applying the hyperbolic tangent sigmoid stretches the range to -1 to +1. This transfer function forces the output nodes to be nearly integer valued to indicate which class the output belongs to. The input/output relation is (Hagan, et al., 1996):

$$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$$

The network used a gradient descent procedure which means that the synaptic weights were changed by an amount proportional to the negative gradient during training (Christodoulou and Georgiopoulos, 2001). Note that initially the weights are assigned to meaningless values. Through training the weights are gradually changed, via back propagation with gradient descent, to weights that map the input attributes ultimately to the correct outputs.

MLP can either be fully connected or partially connected. In a fully connected network, every node, or neuron, in each layer is connected to every node in the next forward layer. A partially connected network has some missing synapses (Christodoulou and Georgiopoulos, 2001). These weights are determined by the correlation between the input data and the desired output.

Figure 2 shows the layout of an MLP network. The number of inputs is equal to K. Data from each input node is passed to each one of J nodes in the hidden layer. In turn, each hidden layer node passes its outcome to each of the I nodes in the output layer. I is equal to the desired amount of outputs. Currently, MLP is the most widely used architecture for classification and prediction problems (Adnan, et al., 2000).

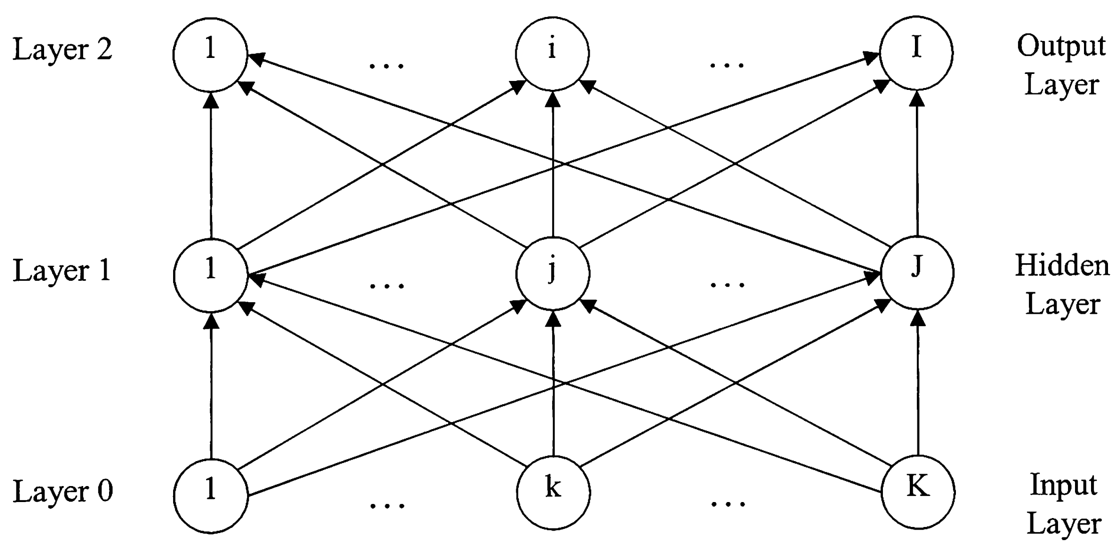

Figure 2: Model of the multi-layer neural network architecture (Christodoulou and Georgiopoulos, 2001).

MLP networks are also called back-propagation networks. Back-propagation is actually the training algorithm used to teach the network. The algorithm starts with the input propagating through the three layers of the network. This initializes the synaptic weights. Next, the sensitivities of those weights are calculated starting at the final output layer working back to the input layer. Finally, the synaptic weights are updated (input layer to output layer) according to the sensitivities (Hagan, et al., 1996). Training will be discussed more in the next section.

Adaptive Resonance Theory (ART) networks were developed by Stephen Grossberg in the 1970s. The name of the network comes from the way the network acts during training. The neuron outputs reverberate back and forth between the node layers until a good pattern is developed. Then the oscillation becomes stable. ART is different from MLP in that it has a "plastic memory." Having a plastic memory means that, after the network is trained on one set of data, more data can be added in without having to retrain with the old data (Christodoulou and Georgiopoulos, 2001). The neural network adapts to the new information without forgetting the old information.

Fuzzy ARTMAP is a particular ART architecture that requires binary input patterns. The fuzzy ARTMAP network is composed of three modules as seen in Figure 3. The ART_a and ART_b modules are fuzzy ART modules with an interART module connecting them. Inputs flow into the ART_a module and corresponding outputs are mapped to the ART_b module. A field within the interART module determines whether the mapping from inputs to outputs is correct. If the mapping is satisfactory, the output is sent through the ART_b module. Otherwise, the process continues with constant communication back and forth among the three modules until the mapping is satisfactory (Christodoulou and Georgiopoulos, 2001).

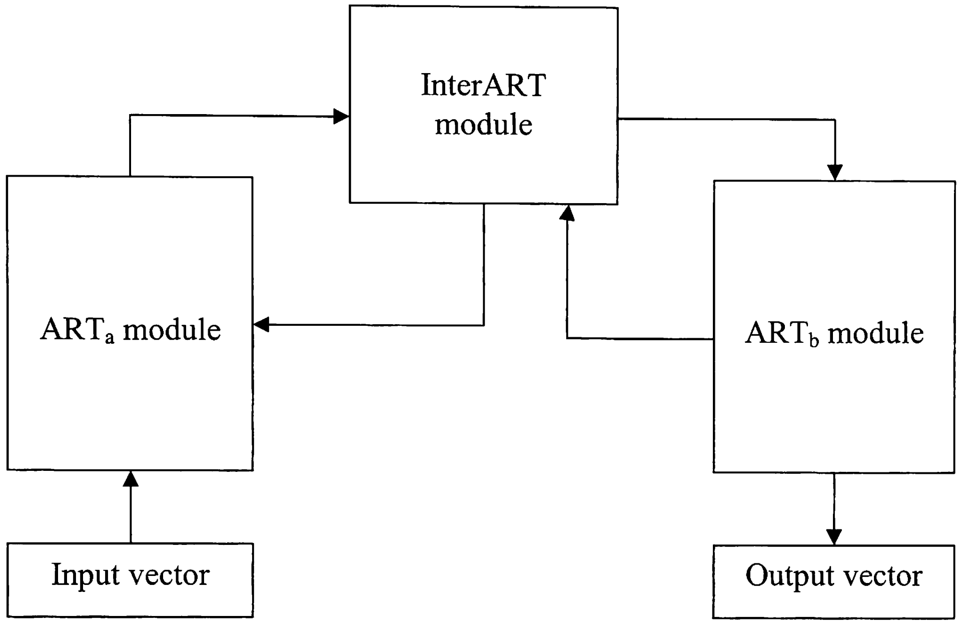

Figure 3: Model of the fuzzy ARTMAP architecture adapted from Christodoulou and Georgiopoulos (2001). The interART module interacts with both the ART$_a$ and ART$_b$ modules to map the input patterns to the output patterns.

Another way to explain how fuzzy ARTMAP classifies data is by using a geometrical view. The weights that are created in the ART$_a$ and ART$_b$ modules are also called *templates*. The templates are represented as rectangles. A training pattern is presented to the network. If the weight of the pattern fits into the previous rectangle template, then it has the same outcome. If the weight does not fit into the rectangle, then a new rectangle template is formed. The size of the rectangles is set by tuning the network parameters with smaller rectangles being more ideal (Christodoulou and Georgiopoulos, 2001). In the end, the rectangles may overlap, but that is allowed. The rectangles represent the output classes that the input data fall into. Figure 4 shows this representation.

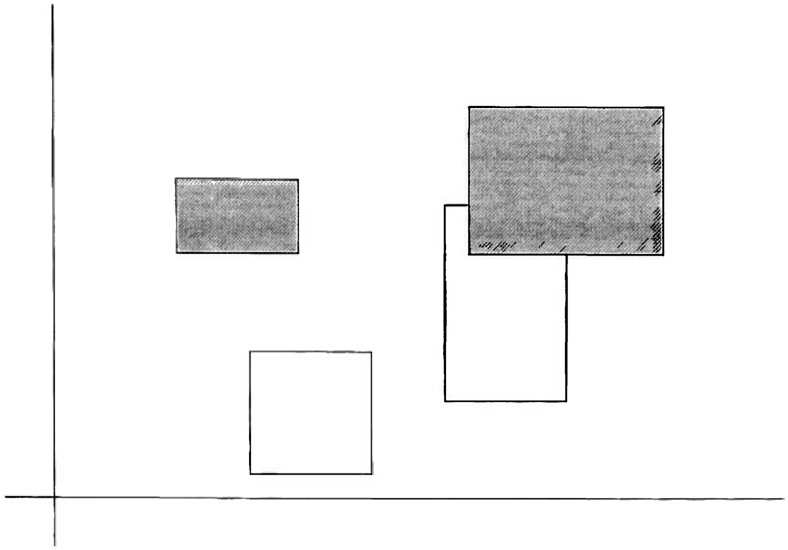

Figure 4: Geometric view of the output patterns as developed during fuzzy ARTMAP training (adapted from Trocine, 2002). The colors show different output classes. The input patterns fall within one of the rectangles or on the border of one.

There are several examples of fuzzy ARTMAP being used for pattern classification. Llobet and others (1999) used an electric nose to determine several characteristics of bananas. They tested the abilities of both MLP and fuzzy ARTMAP to classify the bananas into ripeness categories. Overall, the MLP network had an 83.4% correct classification rate. The fuzzy ARTMAP network classified correctly 90.3% of the time. It was also stated that the fuzzy ARTMAP architecture performed well even with the presence of noise added to the signals from the electronic nose.

Tu, et al. (2001) applied neural network architectures to benchmark datasets in order to compare training time. They found that the training time for fuzzy ARTMAP is relatively fast even with large datasets. The training time required for MLP is comparatively large.

Lee, et al. (2002) applied fuzzy ARTMAP and MLP to channel equalization for digital communications. This application was regarded as a pattern classification problem. The researchers found that fuzzy ARTMAP was not as sensitive to noise as MLP. The training time of the fuzzy ARTMAP network was approximately one-fifth that of the MLP network.

Trocine (2002) compared the performance of MLP and fuzzy ARTMAP on the classification of a benchmark data set. The networks classified wines by their characteristics into one of three categories. MLP correctly classified 83.6% of the wines while fuzzy ARTMAP correctly classified 98.2%.

Another study compared the ability of MLP and fuzzy ARTMAP networks to classify or find anomalies in a cooling system (Meneganti, Saviello, and Tagliaferri, 1998). The networks were run with synthetic, realistic, and real data. Fuzzy ARTMAP had faster computation time for two of the three experiments. The percentages of error with the test sets showed variation among the type of data. With the synthetic data, fuzzy ARTMAP had a percent error of 7.29% and MLP had a percent error of 16.15%. However, the errors with the realistic data for fuzzy ARTMAP and MLP were 0.26% and 0.18% respectively. MLP also performed better than fuzzy ARTMAP with the real data set. These results, however, contradict the other studies presented here.

The details of Fuzzy ARTMAP and MLP have just been discussed. In summary, Table 4 shows the overall differences between MLP and fuzzy ARTMAP. Training the neural networks is discussed in the next section.

Table 4
Comparison of the Characteristics of MLP and Fuzzy ARTMAP

	MLP	**Fuzzy ARTMAP**
Number of Hidden Nodes	Set by programmer	Data driven
Training Time	Long, variable	Short
Memory	New patterns must be presented along with the old patterns in order to generalize	Plastic—new patterns learned without having to relearn the old patterns
Categories	Forces patterns into output classes	Includes an "I Don't Know" class

Training

One of the major benefits of neural networks is their ability to generalize. That is, the network may accurately classify a pattern without having been trained on that pattern. Each of the neural network architectures needs to be trained first, before it can generalize to solve problems. The training procedure allows the network to adapt its synaptic weights to respond to the training patterns. Several iterations, called *epochs*, are required to find the ideal weights. One iteration of all the input patterns is one epoch (Christodoulou and Georgiopoulos, 2001).

Before training, the input data is separated into two categories: a training set and a test set. During training, the training set is applied to the neural network with its appropriate target output. Training is complete when the actual output received from the neural network matches with some high percentage rate the target output. In other words, what the neural network answered (the actual output), matched the actual answers (target output). After training, the test set of data is applied to the neural network. The test set consists of input patterns the network has never seen before. The actual output from the network is compared to the target output that matches the test set. If they match with some high percentage rate, the network has successfully generalized.

Both MLP and fuzzy ARTMAP utilize supervised learning. Supervised learning involves the network adjusting its weights to match the target output. (Unsupervised learning does not require this.) The neural network weights adapt to decrease the error between the target output and the actual output of the neural network. Target responses that the neural network output should reach are provided (Christodoulou and Georgiopoulos, 2001).

According to Christodoulou and Georgiopoulos (2001), generalization is one of the most important considerations when using a neural network for pattern classification. The network must be able to use one data set to generalize to another.

Launch Prediction

There are no previous studies applying artificial neural networks to launch outcome. There are, however, investigations into the causes of the failures after the fact. The failure data is used to find patterns and address the subsystems that fail most often. Chang (2001) summarized the subsystems that failed during launches around the world from 1980 to 1999. The data included manned launch vehicles and unmanned vehicles other than Atlas, Delta, and Titan. The overall problems, however, are constant across all launch vehicles. Propulsion systems fail most often, followed by avionics, stage separation, electrical systems, structural, and other systems.

Pytanowski (1999) presented a case study of the RL10E-1 liquid propellant rocket engine. The study focused on increasing the reliability of the Centaur upper-stage of the Atlas II launch vehicle. Previous launch and engine testing data were used to determine which components of the engine had the highest failure rates. The three most unreliable parts—valves, ignition system, and actuators--were individually redesigned to increase their reliability. The resulting changes reduced the predicted failure rate by three times its original value. The mission reliability was predicted to increase from 0.97 to 0.99.

Recently, a neural network study was conducted on an issue within the launch vehicle industry. Williams, et al. (2004) applied an MLP network to the issue of range safety decisions. Range safety is a portion of launch safety. If a problem develops with a launch vehicle after it is launched, but before it reaches orbit, range safety personnel must decide whether to destroy the vehicle. They must figure out where the vehicle is and where debris would hit the ground if there was an incident.

The model developed by Williams, et al. did not adequately replace the decision-making of the range-safety personnel. It was determined that range-safety was too important of a decision to be left up to a computer.

This study investigated whether overall launch vehicle success could be predicted using an artificial neural network. Two neural network architectures, multi-layer perceptron (MLP) and fuzzy ARTMAP, were compared in doing so. It was expected that the fuzzy ARTMAP network would produce a better model with less error than the MLP network and take less time to train.

The next section details the methods used in this study to model launch prediction.

Methods

The purpose of the study was to determine if artificial neural networks could be used to accurately predict launch vehicle success. In order to accomplish this, two neural network architectures were used to model launch outcome. This chapter details the method in which the models were developed and compared. First, the apparatus needed for the modeling is discussed. Next, the data collection and procedure are outlined. Finally, the specific experimental tests and interpretation of the results are described.

Apparatus

The equipment used for this study was a Dell PC running Windows NT. The software used was Microsoft Excel 2002 Version 10.4524.4219 SP-2 and MATLAB Version 6.5.0.180913a Release 13, by The Math Works, Inc. including the neural network toolbox.

Data

In order to create neural network models of launch prediction, a range of data was gathered from a variety of sources. The data covered system-wide aspects of the launch and the launch vehicle. The input data is listed in Table 5.

Table 5
Factors Affecting Launch prediction

Input Factor	**Source**
Barometric pressure at launch	National Climatic Data Center (2004)
Cloud ceiling at launch	National Climatic Data Center (2004)
Customer country of origin	Isakowitz, et al. (1999)
Intended orbit inclination	Isakowitz, et al. (1999)
Launch date	Isakowitz, et al. (1999)

Launch pad	Isakowitz, et al. (1999)
Launch vehicle manufacturer	Isakowitz, et al. (1999)
Mass of payload(s)	Isakowitz, et al. (1999)
Miles visibility at launch	National Climatic Data Center (2004)
Number of days between launches	Isakowitz, et al. (1999)
Number of engines	Launch Vehicles (2004)
Number of payloads	Isakowitz, et al. (1999)
Payload client	Isakowitz, et al. (1999)
Sky cover at launch	National Climatic Data Center (2004)
Temperature at launch	National Climatic Data Center (2004)
Time of launch	McDowell (2004)
Vehicle model	Isakowitz, et al. (1999)
Wind speed at launch	National Climatic Data Center (2004)

Each of the factors may have a different effect on the success of the launch vehicle. Data concerning the launch date, vehicle model, and payload descriptions were acquired through Isakowitz, et al. (1999). This information, however, only went back as far as 1979 and includes up to 1999. Information about the engines used for the launches was found in Launch Vehicles (2004). The weather data was acquired through the National Climatic Data Center.

Data collection was the most difficult part of this study. Originally, it was planned that data such as the number of preflight anomalies and the value of the payloads would be included. This information, however, is not available to the public. The number of complete data sets was limited, in the end, by the weather data. The weather data used by launch control is not available to the public, so civilian weather observations had to be used. Presently these observations are made every hour, but in the past they were only made a few times a day. This lack of consistent data limited the number of launches with complete data to 125 out of a total of 329 launches. Eight of the 125 launches were failures. The complete data set used for the neural network models is included in the Appendix.

The launch vehicles that this study focused on are the three unmanned vehicles manufactured and currently launched in the United States: Atlas, Delta, and Titan.

Procedure

After the data was compiled in an Excel spreadsheet, the complete data sets were sorted randomly and separated into two groups. Two-thirds of the data sets (or 83 rows) were used as the training set to train the neural networks, while the other third (42 rows) were used as the test set to validate the networks. This is standard practice among researchers applying neural networks (Christodoulou and Georgiopoulos, 2001). Five launch failures were included in the training set, and three failures were in the test set. The sets remained constant for both the MLP and fuzzy ARTMAP networks.

MLP networks can be manipulated in several ways. The number of hidden nodes, the type of transfer function and training algorithm, and the number of epochs can all be changed to find the best MLP network. For this experiment, only the number of hidden nodes was modified. The transfer functions used were "tansig" and "purelin." These are typical classification problem functions. The number of epochs was set to 300. This was to allow sufficient time for the model to converge on a solution. This implementation was a 'plain vanilla' and was not intended to experiment on which MLP configuration would be best.

Before running the programs, the number of hidden nodes to use was analyzed. Table 6 lists works previously cited in this paper and the number of hidden nodes that were used in their models.

Table 6
MLP Configuration for the Previously Discussed Studies

Authors	Number of Training Patterns	Number of Hidden Nodes
Adnan, et al. (2000)	Not stated	Not stated
Amjady and Ehsan (1999)	50	3
Chinnam (1997)	Not stated	50
Coit and Smith (1995)	9600	15
Lee, et al. (2002)	16	8
Llobet, et al. (1999)	44	6
Meneganti, et al. (1998)	380	27
Sinha and Pandey (2002)	350	Not stated
Trocine (2002)	178	7
Tu, et al. (2001)	Not stated	Not stated

From this information, it was decided to try seven different numbers of hidden nodes (3, 6, 9, 12, 15, 18, 21). The optimum number of nodes would be determined by the outcome—the best match to the expected outcome would be the best number of hidden nodes to use.

The program was run with the same training and test sets for each number of hidden nodes. All programs reached the goal before completing 300 epochs except for the 21 hidden node version. Therefore, the 21 hidden node version (and any version with a higher number of hidden nodes) was eliminated. All of the programs from the remaining hidden node versions resulted in the same output—all of the launches in both the training and test sets were classified as successful.

In order to pare down the field even further, 3, 6, and 9 were eliminated because the outputs they produced were just repetitions of the same numbers.

1.1320	0.0448
1.1320	0.0448
1.1320	0.0448
1.1320	0.0448
1.1320	0.0448
...	...

The output of the MLP network is interpreted as follows. The first output column represents a predicted successful launch while the second column represents a predicted failure. Ideally the MLP would output a "1 0" when the launch was actually a success, and a "0 1" when the launch was actually a failure. Instead the MLP network will produce values between 0 and 1 in each of the two columns. The higher value of the two columns is the predicted outcome.

The output for 12, 15, and 18 hidden nodes looked different from above. Though the outputs were the same (all success outcomes), the individual columns of numbers did not just repeat.

0.9372	0.2409
0.9370	0.2414
0.9371	0.2412
0.9372	0.2408
0.9372	0.2409
...	...

Any of these three numbers of hidden nodes could be used. The final number of hidden nodes was chosen to be 15, because it is the mean of these useable numbers.

Fuzzy ARTMAP networks are manipulated when the experimenter defines the values for the network parameters. Standard values were used for all of the parameters. The number of committed and uncommitted nodes in the ART_a and ART_b modules were set to 1. The vigilance parameters for ART_a and ART_b were set to 0 and 1 respectively. Eps was set to 0.001. Only the beta weights were changed during this experiment.

Before the experiment, the best value for the beta weights was determined. The baseline beta weight was set at 0.01 and was to be increased or decreased by powers of ten. The optimum beta weight would be chosen by finding the one that produced the most correct classification of the outcomes. Experimenting with the beta weights would stop when the incremental improvement in the outcome decreased dramatically or if the outcome did not improve.

First, the fuzzy ARTMAP program was run with the beta weights set at 0.01. This resulted in a correct classification of 100% of the training patterns and 90.5% of the test patterns. Next, the beta weights were increased by a power of ten to 0.10. The correct classification decreased to 88.1% of the test patterns. Because the percentage of correct outcome decreased, the beta weights were not increased any more. The next program was run with beta weights of 0.001. The outcome was a correct classification of 90.5% of the test patterns. Beta weights of 0.0001 and 0.00001 also produced the same outcome as with 0.001 and 0.01. Therefore, the beta weights were chosen to be 0.01.

When the optimal neural network programs were completed, the final running of the training and test patterns was begun.

The same training and testing patterns were used for both MLP and fuzzy ARTMAP during each trial. The training and test patterns were randomly ordered before each trial. There were 5 failures in the training pattern and 3 failures in the testing pattern.

The training patterns and outcomes were transferred from an Excel file into text files. The MLP network code for training was run in MATLAB. The code automatically retrieved the training patterns and outcomes from the text files. The neural network was allowed to train on this data for a given number of epochs. A popup screen in MATLAB showed the network coming to convergence. When this step was completed, the code for testing the network was run in MATLAB. Again, the code was set up to retrieve the test patterns from a text file. The outputs from the training and testing phases were saved into new text files. Those outputs were compared to the actual outcomes. When the actual outcome of the launch (success or failure) matched the model outcome (success or failure), the match was a correct prediction.

Training and testing for fuzzy ARTMAP occurred in a similar fashion. The fuzzy ARTMAP model retrieved the training and testing patterns from text files. The text files were set up slightly different, however. Each row of the training pattern included the 18 attributes followed by the complements of those attributes. The outcomes were presented in the same way. The fuzzy ARTMAP code was run in MATLAB. There was no popup screen to show the progress of the model towards convergence because each running of the training program is one epoch.

In order to determine if the network could generalize to new data, the testing phase of fuzzy ARTMAP was implemented next. The code of the testing phase is set up to compare the output of the network to the actual outcome. The model returned information on the MATLAB screen about what percentage of the outputs were correct and specifically which lines in the testing pattern were classified incorrectly.

This procedure was repeated ten times on both the MLP and fuzzy ARTMAP networks. The repetition of the experiment ensured to minimize any side effects of order of presentation.

Results

This study was intended to not only determine if launch outcome could be predicted by a neural network, but also to compare the performance of two kinds of neural networks. The outcomes for both of these objectives are shown next.

The results presented in Table 7 show some differences between the outcomes of MLP and fuzzy ARTMAP. The outcome is split into successes and failures because this more accurately shows the differences. Overall, including successes and failures together, fuzzy ARTMAP performed statistically the same as MLP, $t(18) = 1.085$, $p > 0.05$. This information is distorted, however. The MLP model did not classify any launches (in training or testing) as failures. Therefore, 100% of the successes were correctly classified, and 0% of the failures were correctly classified throughout all ten trials. On the other hand, the fuzzy ARTMAP model correctly classified 100% of the successes and failures during the training phase. In the testing phase, 93.3% of the successes were correctly classified and 20% of the failures were correctly classified.

Table 7
Percentage of Correctly Classified Launches from the Neural Network Models

	Training		**Testing**	
	% of Successes Correct N = 78	**% of Failures Correct** N = 5	**% of Successes Correct** N = 39	**% of Failures Correct** N = 3
MLP	100 %	0 %	100 %	0 %
Fuzzy ARTMAP	100 %	100 %	93 %	20 %

These results can also be analyzed in another way. Table 8 shows the outcome from the two models in the context of Signal Detection Theory. The objective of the models is to maximize hits and correct rejections while minimizing misses and false alarms. As discussed previously, the cost of a miss is about $300 million dollars. A false alarm, however, would just result in a delay of the launch. A delay would cost money in operations and personnel, but the magnitude of the cost a delay is nowhere near that of a launch failure. Though fuzzy ARTMAP did miss 80% of the failures, the network shows possibility in that it correctly rejected 20% of the failures. MLP does not show potential; it missed 100% of the failures.

Table 8
Percentage of Launches Classified in each Category during the Testing Phase

		Predicted Outcome	
		Success	***Failure***
Actual Outcome	***Success***	MLP: 100 % FAM: 93 % Hit	MLP: 0 % FAM: 7 % False Alarm
	Failure	MLP: 100 % FAM: 80 % Miss	MLP: 0 % FAM: 20 % Correct Rejection

Figure 5 shows the difference in The MLP model used between 2 and 6 epochs to train the network; while fuzzy ARTMAP used one epoch each trial. Fuzzy ARTMAP overall trained in about one-fifth of the time that MLP did.

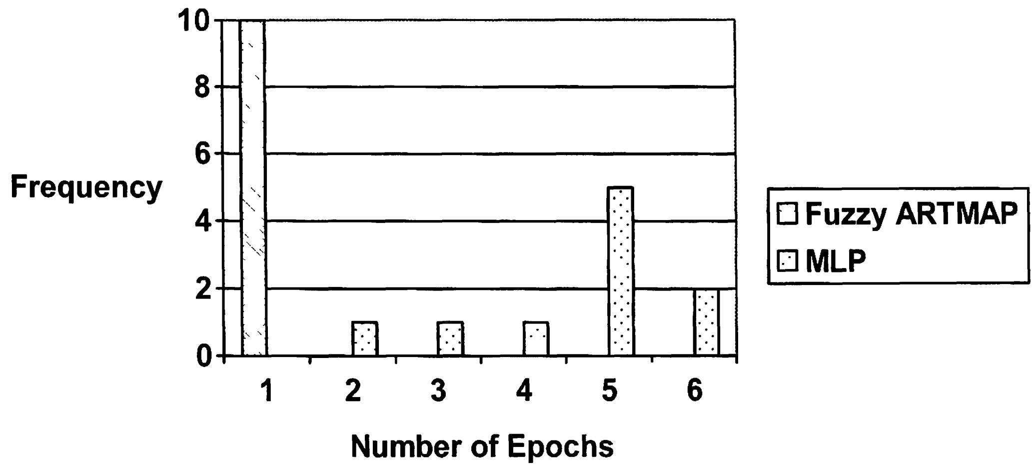

Figure 5: Bar graph indicating the frequency of use of each number of epochs by fuzzy ARTMAP and MLP.

The number of hidden nodes was set to 15 for the MLP model. Fuzzy ARTMAP determines for itself how many hidden nodes are needed. Throughout the trials, fuzzy ARTMAP used between 7 and 11 hidden nodes. Overall, fuzzy ARTMAP was faster and more efficient (used less resources) than MLP. In the space industry it is important to have a fast, efficient model. During launch countdowns, time is very critical.

MLP models do not allow the researcher to discover how the input patterns match up to the output. The model acts as a black box, inside of which the weights assigned to each attribute remain hidden. Fuzzy ARTMAP models, on the other hand, give the researcher information that can be used to determine the weight each attribute has on the outcome. At the end of the training phase, the fuzzy ARTMAP model saves the weights of the trained ART_a, ART_b, and interART modules in three files. These files are then called during the testing phase and used to generalize to the new patterns. The weights may be used to determine the impact each attribute has on the final output. A trace may be done from node to node to determine the weight assigned to the attribute.

Conclusions

Expendable launch vehicles have a great impact on our society. A model that helps predict launch vehicle outcome would be a great asset. It would save money, time, and perhaps lives. Currently, the reliability rate of launch vehicles in the United States is 92%. A model that predicts the launch outcome of a specific launch vehicle—whether it will be in the successful group or in the 8% that fail—would be a valuable asset to the space industry.

Though the models presented do not perform as well as hoped, the fuzzy ARTMAP model does show promise. The MLP network never found a failure. All of the outputs for every trial were successes. The fuzzy ARTMAP model did correctly classify one failure during six of the ten trials. However, all of the failures were misclassified during the rest of the trials. Also some successes were misclassified as failures.

The failure that fuzzy ARTMAP found in six of the ten trials was the same launch. This would indicate that either this launch was very similar to the failures in the training pattern or it was very different from the other failures in the test pattern. Close examinations of these launches found that most attributes of the test pattern failures fell within the range of the attributes of the training pattern failures. By human standards, all eight failures are very similar to each other. However, some difference must exist that sets that one failure apart from the others.

The expected results were that the fuzzy ARTMAP network would produce less error and have a faster training time. The fuzzy ARTMAP network trials did have a lower training time and more efficiency in the number of hidden nodes used.

Statistically, there was no difference between the results of the MLP and fuzzy ARTMAP testing phases; however, the statistics are misleading in this case. When the results are broken down into the number of failures correctly classified, fuzzy ARTMAP does perform better.

There could be several reasons why the models did not work as expected in modeling launch outcome. It is thought that the most likely reason for the unsuccessful models was the lack of more in depth data. Information such as the reliability of the engines, the number of preflight anomalies, and other data was not available to the public. This different data may be more relevant to the outcome of a launch than the data that was included in this study. Including data with a higher correlation to the outcome should produce a better classification model.

The lack of complete weather data also served to limit the number of training patterns applied to the networks. Though networks can be adequately trained on very few data patterns, it was an inconvenience to cut the data set down to less than half because of the lack of the weather data. This also introduced the bigger problem that there were only eight failures in the total data patterns. More failures would have given the models more information. This may have been the problem with MLP. The network was overpowered with successful launches (almost 16 times more successes than failures during training). The failure category was so small that the MLP model did not want to classify any data set into that small category.

This experiment, though not as successful as hoped, does serve to show that there is a possible use for neural networks in the launch vehicle industry. It is recommended to future researchers to attempt to create the models again with different and more specific data. The idea of using neural networks to model launch outcome should not be quelled.

References

Adnan, W.A., Yaakob, M., Anas, R., and Tamjis, M.R. (2000). Artificial Neural Network for Software Reliability Assessment. *Proceedings of the IEEE Region 10 Annual International Conference,* 3, III-446-III-451.

Amjady, N. and Ehsan, M. (1999). Evaluation of power system reliability by an artificial neural network. *IEEE Transactions on Power Systems*, 14(1), 287-292.

Boeing. (2004). *Delta Launch Record*, accessed February 10, 2004. (http://www.boeing.com/defense-space/space/delta/record.htm).

Chang, I. (2000). Overview of world space launches. *Journal of Propulsion and Power*, 16(5), 853-866.

Chang, I.S. (2001). Space Launch Vehicle Reliability. *Crosslink,* 2(1), 22-32.

Chinnam, R.B. (1997). Performance reliability monitoring of physical systems using artificial neural networks. Proceedings of the Industrial Engineering Research Conference, Norcross, GA, 471-476.

Christodoulou, C. and Georgiopoulos, M. (2001). *Applications of Neural Networks in Electromagnetics*. Boston: Artech House, Inc.

Coit, D.W. and Smith, A.E. (1995). Using a neural network as a function evaluator during GA search for reliability optimization. *Intelligent Engineering Systems Through Artificial Neural Networks*, 5, 369-374.

Fragola, J.R. (1991). A second look at launch system reliability. *Aerospace America*, 29(11), 36-39.

Hagan, M.T., Demuth, H.B., and Beale, M. (1996). *Neural Network Design.* Boston: PWS Publishing Company.

Hiebert, S.F. and Chinnam, R.B. (2000). Role of artificial neural networks and wavelets in on-line reliability monitoring of physical systems. Proceedings of the International Joint Conference on Neural Networks, Piscataway, NJ, 369-374.

Hill,B. (2000). *Boeing Delta III Rocket Successfully Places Data-gathering payload into Orbit*, accessed February 10, 2004. (http://www.boeing.com/news/releases/2000/news_release_ 000823h.html).

Isakowitz, S.J., Hopkins Jr., J.P., and Hopkins, J.B. (1999). *International Reference Guide to Space Launch Systems* (3rd ed.). AIAA.

Khaparde, S.A. and Bhattacharyya, K. (1996). Power system reliability evaluation using fuzzy logic and neural networks. *International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications*, 4(4), 197-206.

Launch Vehicles. *Aviation Week and Space Technology*, January 19, 2004.

Lee, J., Choi, Y., Lee, J., and Han, S. (2002). Channel equalization using fuzzy-ARTMAP. *IEICE Transactions on Communications*, E85-B(4), 826-831.

Llobet, E., Hines, E.L., Gardner, J.W., and Franco, S. (1999). Non-destructive banana ripeness determination using a neural network-based electronic nose. *Measurement Science and Technology*, 10(6), 538-548.

McDowell, J. (2004). *Launch Log*. Retrieved May 7, 2004, from http://planet4589.org/space/log/launchlog.txt

Meneganti, M., Saviello, F.S, and Tagliaferri, R. (1998). Fuzzy neural networks for classification and detection of anomalies. *IEEE Transactions on Neural Networks*, 9(5), 848-861.

National Climatic Data Center. (2004). *Surface Weather Observations*. Retrieved May 7, 2004 from http://ols.nndc.noaa.gov/plolstore/plsql/olstore.prodspecific?prodnum=C00483-MAN-S0001

Parkinson, R.C. (1998). The Hidden Costs of Reliability and Failure in Launch Systems. *Journal of Reducing Space Mission Cost*, 1, 263-273.

Pytanowski, D.C. (1999). Rocket-engine control-system reliability-enhancement analysis. Proceedings of the Annual Reliability and Maintainability Symposium, USA, 1-6.

Sinha, S.K. and Pandey, M.D. (2002). Probabilistic neural network for reliability assessment of oil and gas pipelines. *Computer-Aided Civil and Infrastructure Engineering*, 17(5), 320-329.

Trocine, L. (2002). *A brief overview of the fuzzy ARTMAP neural network and its application to an aviation prediction problem*. Unpublished report. Embry-Riddle Aeronautical University: Daytona Beach, Florida.

Tu, F., Wen, F., Willett, P., Pattipati, K., and Jordan, E.H. (2001). Signal processing and neural network toolbox and its application to failure diagnosis and prognosis. Proceedings of the International Society for Optical Engineering, 4389, 121-132.

Williams, G., Keen, J., Ostroff, R., and Richard, D. (2004). Use of neural networks as a range safety decision support system. *Proceedings of the 41st Space Congress*, USA.

Appendix

Table 9
Index to the Following Launch Vehicle Data

M	Month			
Day	Day			
Year	Year			
Time	Time (UTC)			
Intvl	Launch Interval (days)			
Model	Vehicle Model	11—Atlas E, F, and SLV 3D	23—Delta 2310	29—Delta 8930
		14—Atlas H and G	24—Delta 6920 and 6925	30—Delta 7320, 7325, and 7326
		16—Atlas I	25—Delta 4925	51—Titan 2
		17—Atlas II, IIA, and IIAS	26—Delta 5920	52—Titan 3, 3C, 3D, 34B, 34D, and 24B
		21—Delta 2914	27—Delta 7920 and 7925	
		22—Delta 3910, 3913, 3914, and 3920	28—Delta 7420 and 7425	53—Titan 4, 402A, 405A, 403A, 404A, 401A, 402B, 401B, and 404B
Eng	Number of Engines			
Pad	Launch Pad	1—SLC 2W	5—LC 40	9—LC 36A
		2—LC 17A	6—LC 41	10—LC 36B
		3—LC 17B	7—SLC 3W	11—SLC 4E
		4—SLC 2E	8—SLC 3E	
Pylds	Number of Payloads			
Mass	Payload Mass (kg)			
Orbit	Intended Orbit	1—EEO	4—SSO	7—GEO
		2—GTO	5—MEO	8—Polar Orbit
		3—LEO	6—Outside Earth Orbit	
Mrkt	Market	1—Military	2—Commercial	3—Civilian
Cntry	Country of Origin	1—USA	5—Europe	9—Japan
		2—United Kingdom	6—Indonesia	10—Korea
		3—India	7—Germany	11—Norway

		4—Canada	8—International	12—Russia
Speed	Wind Speed (mph)			
Ceiling	Ceiling (feet)			
Sky	Sky Cover	1—Broken	3—Overcast	5—Clear
		2—Scattered	4—Obscured	
Vsblty	Visibility (mi)			
Temp	Temperature (°F)			
Press	Barometric Pressure (in Hg)			
Manu	Vehicle Manufacturer	1—Titan	2—Atlas	3—Delta
Result	Launch Result	1—Success	0—Failure	

Table 10
Complete Launch Vehicle Data

M	Day	Year	Time	Intvl	Model	Eng	Pad	Pylds	Mass	Orbit	Mrkt	Cntry	Speed	Ceiling	Sky	Vsblty	Temp	Press	Manu	Result
3	16	1979	1030	92	52	6	4	2	13360	4	1	1	0	98	1	9.9	54	30.03	1	1
6	27	1979	852	54	11	7	7	1	723	3	3	1	21	197	1	5	60	30.06	2	1
2	7	1980	1310	78	52	6	4	1	13300	3	1	1	18	722	2	9.9	59	29.86	1	1
6	18	1980	1129	132	52	6	4	1	1300	4	1	1	7	14	3	7	57	30.08	1	1
9	3	1981	1129	132	52	6	4	1	13300	4	1	1	0	6	3	6	59	29.92	1	1
12	18	1981	1710	3	11	7	8	1	759	5	1	1	0	128	1	4	58	30.11	2	0
2	28	1981	1115	77	52	6	1	1	3000	4	1	1	23	36	1	7	52	29.86	1	1
7	16	1982	1059	37	22	11	1	1	1972	4	3	1	9	7	3	7	59	29.97	3	1
6	20	1983	1145	66	52	6	4	1	32000	4	1	1	9	14	1	7	64	29.87	1	1
5	26	1983	818	28	22	12	1	1	540	1	3	5	3	6	4	0.9	54	29.83	3	1
3	28	1983	752	47	11	7	7	1	3775	4	3	1	6	197	1	14.9	52	30.12	2	1
3	1	1984	959	161	22	11	1	2	1990	4	3	8	1	722	1	12.9	61	30.12	3	1
9	8	1984	1441	87	11	7	7	1	759	5	1	1	9	722	2	24.9	93	29.89	2	1
6	9	1984	1903	125	14	8	10	1	1928	2	2	8	9	722	5	7	77	30.11	2	0
11	14	1984	1934	54	22	12	2	1	760	2	1	5	0	722	2	7	58	30.34	3	1
8	16	1984	1048	168	22	12	2	3	924	1	3	8	9	722	5	7	87	30.09	3	1
9	21	1984	1818	36	22	11	3	1	1218	2	2	1	9	98	1	7	79	29.95	3	1
12	22	1984	1902	18	52	6	5	1	1170	7	1	1	0	722	2	7	68	30.15	1	1
6	25	1984	1143	69	52	6	4	2	13360	4	1	1	8	722	2	24.9	65	30.05	1	1
3	13	1985	1800	91	11	7	7	1	635	3	1	1	7	722	2	11.9	54	29.98	2	1
3	22	1985	1855	9	14	8	10	1	2013	2	2	8	11	722	2	7	75	29.86	2	1
9	28	1985	1917	90	14	8	10	1	2013	2	2	8	11	30	3	7	78	30.14	2	1
5	3	1986	1818	535	22	12	2	1	838	2	3	1	17	246	1	5	75	30.03	3	0
9	5	1986	1108	125	22	11	3	2	2495	3	1	1	11	722	1	7	85	29.99	3	1
12	5	1986	2130	79	14	8	10	1	2310	2	1	1	17	98	1	7	62	30.22	2	1
9	17	1986	852	220	11	7	7	1	1712	4	3	1	11	10	1	7	64	30.07	2	1
3	20	1987	1722	22	22	11	3	1	1244	2	2	6	16	722	1	7	64	29.89	3	1
2	26	1987	1805	174	22	12	2	1	841	2	3	1	11	79	3	5	70	30.18	3	1
3	26	1987	1622	111	14	8	10	1	2300	2	1	1	0	15	3	1.5	68	29.97	2	0
2	8	1988	1707	325	22	11	3	1	1574	3	1	1	16	12	3	7	52	30.15	3	1
6	10	1989	1819	78	24	12	2	1	1657	5	1	1	14	722	2	7	82	30.16	3	1

12	11	1989	1310	23	24	12	3	1	1664	5	1	1	5	722	2	7	65	30.09	3	1
2	14	1989	1330	372	24	12	2	1	1657	5	1	1	11	722	2	7	76	30.4	3	1
6	14	1989	918	35	53	7	6	1	2355	7	1	1	9	722	5	7	79	30.15	1	1
10	21	1989	531	55	24	12	2	1	1664	5	1	1	11	722	5	7	45	30.24	3	1
3	24	1989	1651	38	22	11	3	1	2637	3	1	1	9	722	2	7	73	30.1	3	1
1	1	1990	1907	117	52	5	5	2	3743	2	1	8	9	722	2	7	51	30.25	1	1
6	1	1990	1748	49	24	12	2	1	2440	3	3	7	9	722	2	7	78	30.16	3	1
12	1	1990	757	129	11	7	7	1	823	4	1	1	11	722	5	7	55	30.13	2	1
4	13	1990	1828	18	24	11	3	1	1241	2	2	6	9	722	2	7	68	30.22	3	1
2	14	1990	1115	21	24	10	3	2	2470	3	1	1	14	722	2	7	74	30.28	3	1
3	14	1990	652	72	52	5	5	1	4215	2	2	8	0	722	2	7	66	30.11	1	0
6	23	1990	719	15	52	5	5	1	4215	2	2	8	9	148	1	7	73	29.99	1	1
1	24	1990	1755	44	24	12	2	1	1664	5	1	1	11	722	2	7	74	30.03	3	1
7	25	1990	1521	105	16	7	10	1	1629	1	3	1	10	722	2	7	90	30.01	2	1
10	30	1990	1816	29	24	12	3	1	1370	2	2	8	7	722	2	7	70	30.2	3	1
12	7	1991	1747	9	17	5	10	1	1841	2	2	5	5	722	2	7	64	30.18	2	1
1	8	1991	1953	43	27	12	3	1	1434	2	1	5	14	5	3	3	62	30.23	3	1
3	8	1991	1803	59	24	12	3	1	1385	2	2	1	9	722	2	7	81	29.86	3	1
5	14	1991	852	26	11	7	7	1	1413	4	3	1	14	722	2	7	57	30.05	2	1
11	28	1991	523	198	11	7	7	1	823	4	1	1	15	722	5	7	50	30.04	2	1
5	29	1991	1855	46	27	12	3	1	1338	2	2	1	5	722	5	7	81	30	3	1
7	3	1992	1754	23	17	5	9	1	2615	2	1	1	0	722	5	4	94	30.03	2	1
6	7	1992	1240	24	24	12	2	1	3249	3	3	1	7	722	2	7	84	30.07	3	1
6	10	1992	2000	88	17	6	10	1	2928	2	2	8	6	148	1	7	78	29.93	2	1
2	11	1992	1941	66	17	5	9	1	2615	2	1	1	7	722	5	5	51	30.23	2	1
10	12	1992	547	33	27	12	3	1	1411	2	2	7	6	148	1	7	66	29.97	3	1
12	18	1992	1716	26	27	12	3	1	1882	5	1	1	7	59	1	7	68	30.13	3	1
2	23	1992	1729	234	27	12	3	1	1882	5	1	1	9	722	2	7	73	29.98	3	1
9	25	1992	1305	153	52	5	5	1	2573	6	3	1	11	98	3	7	86	29.98	1	1
8	31	1992	641	38	27	12	3	1	1402	2	2	1	6	722	2	7	74	30.14	3	1
9	3	1993	717	25	16	7	10	1	2844	2	1	1	0	722	2	7	81	30.16	2	1
10	5	1993	1056	64	51	3	1	1	1740	4	3	1	10	23	1	7	63	30.03	1	1
12	8	1993	1948	43	27	12	3	1	1434	2	1	5	7	722	5	6	56	30.2	3	1
5	13	1993	2007	44	27	12	2	1	1882	5	1	1	7	722	2	7	73	29.78	3	1

12	16	1993	1938	18	17	7	10	1	3375	2	2	1	4	722	5	7	46	30.07	2	1
3	25	1993	1638	215	16	7	10	1	2866	2	1	1	3	148	1	7	71	30.05	2	0
6	26	1993	927	44	27	12	2	1	1882	5	1	1	8	722	2	7	82	30.13	3	1
10	26	1993	1204	57	27	12	3	1	1882	5	1	1	9	10	3	5	72	29.93	3	1
8	30	1993	838	65	27	12	3	1	1882	5	1	1	7	722	2	7	80	30.03	3	1
8	3	1994	1957	40	17	7	9	1	2860	2	2	1	6	722	2	7	78	30.1	2	1
12	22	1994	1719	117	53	7	5	1	2355	7	1	1	4	7	3	4	56	29.81	1	1
6	24	1994	950	72	16	7	10	1	2847	2	1	1	11	722	2	6	83	30.05	2	1
1	25	1994	834	112	51	3	1	1	424	6	1	1	8	722	2	10	47	30.03	1	1
12	30	1994	202	31	11	7	7	1	1712	4	3	1	6	722	5	7	38	29.98	2	1
11	4	1995	622	91	27	12	1	2	2768	4	1	8	6	722	5	7	52	29.94	3	1
8	5	1995	710	277	27	12	3	1	1447	2	2	10	0	722	2	4	77	30.04	3	0
4	7	1995	1947	14	17	6	9	1	2700	2	2	1	7	49	1	7	64	29.97	2	1
12	15	1995	1923	13	17	6	9	1	2980	2	2	1	0	722	5	7	68	30.17	2	1
3	24	1995	605	2	11	7	7	1	830	4	1	1	9	722	2	7	38	30.07	2	1
12	30	1995	848	56	27	12	2	1	3030	3	3	1	0	98	3	6	62	30.27	3	1
5	31	1995	1127	8	17	5	9	1	3015	2	1	1	7	722	2	6	82	30.06	2	1
7	31	1995	1930	61	17	6	9	1	2610	2	1	1	7	722	2	5	76	29.99	2	1
2	1	1996	2015	48	17	7	10	1	2980	2	2	6	6	722	5	7	68	30.07	2	1
4	3	1996	1901	62	17	6	9	1	2066	2	2	8	8	246	1	7	66	30.11	2	1
11	7	1996	1200	56	27	12	2	1	1060	6	3	1	9	722	2	7	84	30.1	3	1
12	18	1996	2057	27	17	6	10	1	2074	2	2	8	0	197	1	7	68	29.96	2	1
11	21	1996	1547	74	17	6	9	1	2910	2	2	5	9	197	1	7	81	29.87	2	1
2	24	1996	324	7	27	13	1	1	1301	1	3	1	14	722	2	7	48	29.96	3	1
9	4	1997	803	38	17	7	9	1	2580	2	2	1	5	197	1	5	79	29.9	2	1
5	5	1997	755	108	27	12	1	5	3450	3	2	1	2	246	1	7	59	30.08	3	1
10	5	1997	1701	31	17	7	10	1	3282	2	2	1	11	197	1	7	81	30.06	2	1
11	6	1997	1930	40	27	12	2	1	1881	5	1	1	0	722	2	7	70	30	3	1
12	8	1997	1852	44	17	7	10	1	3560	2	2	1	3	722	2	7	61	30.03	2	1
7	9	1997	604	50	27	12	1	5	3450	3	2	1	1	0	4	0.2	54	29.93	3	1
11	9	1997	1734	3	27	12	1	5	3450	3	2	1	2	28	1	7	57	29.83	3	1
1	17	1997	1128	44	27	12	2	1	1881	5	1	1	14	722	2	7	46	30.25	3	0
2	17	1997	2042	61	17	7	10	1	3094	2	2	9	5	722	5	7	59	30.41	2	1
5	20	1997	1839	15	27	12	2	1	1248	2	2	11	7	722	2	7	81	29.96	3	1

12	20	1997	516	41	27	12	1	5	3450	3	2	1	7	722	2	7	39	30.02	3	1
8	21	1997	1639	29	27	12	1	5	3450	3	2	1	13	6	3	3	63	30.05	3	1
9	27	1997	1824	33	27	12	1	5	3450	3	2	1	17	722	2	7	64	29.82	3	1
11	6	1998	538	13	27	12	1	5	3450	3	2	1	7	722	2	7	48	30.05	3	1
9	8	1998	1413	12	27	12	1	5	3450	3	2	1	7	11	1	7	68	29.77	3	1
10	9	1998	1850	113	17	6	10	1	2900	2	2	5	5	79	3	5	77	29.93	2	1
12	11	1998	1346	19	28	7	2	1	629	6	3	1	17	49	1	7	77	30.16	3	1
5	13	1998	852	4	51	3	1	1	2232	4	3	1	9	69	1	7	55	29.89	1	1
4	14	1998	934	15	28	12	2	4	1800	3	2	1	14	197	1	7	72	30.18	3	1
3	16	1998	1632	16	17	5	9	1	3200	2	1	1	16	722	2	7	73	30.14	2	1
5	17	1998	1417	23	27	12	1	5	3450	3	2	1	22	722	2	7	59	30.07	3	1
2	18	1998	558	39	27	12	1	5	3450	3	2	1	8	722	5	7	43	30.14	3	1
6	18	1998	1848	94	17	7	9	1	3692	2	2	8	6	722	2	3	88	30.06	2	1
4	24	1998	1838	10	28	12	2	4	1800	3	2	1	7	722	2	7	66	30.04	3	1
10	24	1998	808	46	30	6	2	2	520	6	3	1	11	25	1	7	72	30.25	3	1
3	30	1998	2202	40	27	12	1	5	3450	3	2	1	5	722	2	7	45	30.05	3	1
1	3	1999	1521	23	28	7	3	2	574	6	3	1	11	25	3	7	73	29.98	3	1
2	7	1999	1604	35	28	7	2	1	380	6	3	1	14	722	2	7	79	30.01	3	1
4	9	1999	1301	240	53	7	6	1	2382	7	1	1	11	722	2	7	88	29.99	1	1
6	10	1999	949	36	28	6	3	4	1800	3	2	1	7	20	1	7	82	30.05	3	1
4	12	1999	1850	55	17	7	9	1	3178	2	2	5	6	722	5	7	81	29.97	2	1
4	15	1999	1132	51	27	12	1	1	2200	4	3	1	8	722	2	7	66	30.01	3	1
2	16	1999	2045	119	17	7	9	1	2904	2	2	9	5	722	2	7	66	30.14	2	1
6	19	1999	1915	29	51	3	1	1	970	3	3	1	10	5	3	7	54	29.89	1	1
2	23	1999	230	16	27	11	1	3	2617	4	1	8	10	722	5	7	55	30.25	3	1
6	24	1999	1144	14	30	5	2	1	1360	3	3	1	9	197	1	7	84	30.04	3	1