LOAN DOCUMENT

PHOTOGRAPH THIS SHEET

LEVEL

INVENTORY

(0)

DTIC ACCESSION NUMBER

Target Discrimination with Neural . . .

DOCUMENT IDENTIFICATION

Sep 96

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DISTRIBUTION STATEMENT

H A N D L E W I T H C A R E

ACCESSION FOR	
NTIS GRA&I	☑
DTIC TRAC	☐
UNANNOUNCED	☐
JUSTIFICATION	
BY	
DISTRIBUTION/	
AVAILABILITY CODES	
DISTRIBUTION	AVAILABILITY AND/OR SPECIAL
A-1	

DISTRIBUTION STAMP

DATE ACCESSIONED

19970114 033

DATE RETURNED

DATE RECEIVED IN DTIC

REGISTERED OR CERTIFIED NUMBER

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-FDAC

DTIC FORM JUN 90 70A

DOCUMENT PROCESSING SHEET

PREVIOUS EDITIONS MAY BE USED UNTIL STOCK IS EXHAUSTED.

LOAN DOCUMENT

DTIC QUALITY INSPECTED 1

17 JAN 1997

TARGET DISCRIMINATION WITH NEURAL NETWORKS

Daw-Tung Lin
University of Maryland
College Park, MD

Judith Dayhoff
University of Maryland
College Park, MD

Cheryl Resch
Johns Hopkins University Applied Physics Laboratory
Laurel, MD

Abstract

The feasibility of discriminating the warhead from an intentionally segmented exo-atmospheric threat missile is demonstrated by applying the time-delay neural network (TDNN) and the adaptive time-delay neural network (ATNN). Exo-atmospheric threats are especially difficult to distinguish using currently available techniques because all threat segments follow the same trajectory. Thus, classification must be done using infrared sensors that record the signal over time. Results have demonstrated that the trained neural networks were able to successfully identify warheads from other missile parts on a variety of simulated scenarios, including differing angles and tumbling. The network with adaptive time delays (the ATNN) performs highly complex mapping on a limited set of training data and achieves better generalization to overall trends of situations compared to the TDNN, which includes time delays but adapts only its weights. The ATNN was trained on additive noisy data, and it is shown that the ATNN possesses robustness to environment variations.

1. Introduction

Automatic target recognition (ATR) is a highly challenging problem. It involves extraction and discrimination of critical information from complex and uncertain data. Because of target signature variability, environmental changes, and a limited database, the traditional approaches of signal processing and rule-based expert systems have been only partially successful.[1] Neural network technology offers a number of tools, such as learning, adaptation, generalization and robustness, feature extraction, and hardware implementation, that could form the bases of a fruitful approach to the ATR problem.[1-3] Neural network architectures with dynamic and temporal capabilities are more promising for signal analysis.

The time delay neural network (TDNN) proposed by Waibel et al.[4] employs time delays on connections and has been successfully applied to phoneme recognition.[5-7] The TDNN also classifies spatio-temporal patterns and provides robustness to handling noise and allowing graceful degradation.[8] In the TDNN architecture, each neuron takes into account not only the current information from its input neurons of the previous layer but also a certain amount of past information from those neurons as a result of delays on interconnections. Typically, the time delays are evenly spaced over a time interval called the frame window, although arbitrary time delays may be used. Training is done with spatio-temporal patterns, and the classification of those patterns is reported at each time step by the output layer. After training, weights are strengthened along interconnections whose time delays are important to recognition.

The adaptive time-delay neural network (ATNN), which adapts time delays as well as weights during training, is a more advanced version of the TDNN. The result is a dynamic learning technique for spatio-temporal classification.[9] We use an algorithm with adaptive time delays, [10,11] which is a powerful tool for dynamic learning. The ATNN model employs modifiable time delays along the interconnections between two processing units, and both time delays and weights are adjusted according to system dynamics in an attempt to achieve the desired optimization. The adaptation of the delays and weights are derived on the basis of the gradient descent method to minimize the energy or cost function during training. Weight modification is based on error back-propagation,[12] and the mathematical derivation of the time delay modifications is done with a gradient descent approach.[9,10,11,13] The weights and time delays are updated step by step proportional to the opposite direction of the error gradient, respectively. Processing units do not receive data through a fixed time window but gather important information from various time

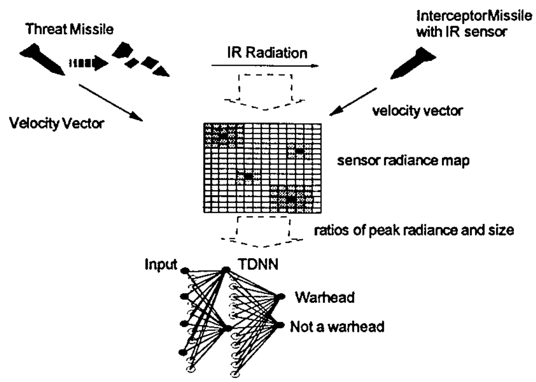


Figure 1 - Schematic for Discrimination

delays that are adapted via the learning procedure. With these mechanisms, the network implements the dynamic delays along the interconnections of the ATNN.

We report here the application of the ATNN to exo-atmospheric target discrimination. The scenario of the problem is schematically illustrated in Figure 1. An intentionally segmented exo-atmospheric threat missile will be detected by an infrared sensor. The infrared sensor records the intensity map over time. The intensity map is then reduced to the peak intensity of each threat segment as a function of time. The neural network then uses this information to discriminate the warhead from the other segments regardless of different aspect angles, tumbling, or number of segments.

This is an important, yet difficult, task in discrimination, as Resch[14] states, "The ability of a kinetic kill vehicle (KKV) to discriminate between warhead and booster parts or decoys is critical to theater ballistic missile defense (TBMD). Exo-atmospheric targets are especially difficult to distinguish using currently available techniques because all target parts follow the same trajectory during the exo-atmospheric portion of the flight." Furthermore, only one sensor is assumed to be available.

The sensors used in this analysis have dedicated specifications. The maximum radiance versus time is obtained from these sensors for all four components (denoted in this context as warhead, oxidizer tank, fuel tank, and tail, in which "warhead" is is the target we are interested in). Furthermore, these components may tumble with different aspect angles or may break into several pieces. One of the many things that makes ATR so hard is that the same threat can vary widely in appearance depending on aspect angles, atmospheric effects, and other variables.

We address this problem by using both the TDNN and the ATNN. We conclude that the TDNN has good performance and that the ATNN can make further improvements in this performance.

2. TDNN Technique

2.1 Simulation Set-ups

The TDNN was first used to perform the discrimination task of distinguishing the warhead from the other missile segments. We used a TDNN with an input layer with six inputs, one hidden layer with three hidden units, and an output layer with two units. The number of time delays in the first layer is four and on the second layer is six. Time delays on the connections between neurons are fixed and are consecutively and equally spaced as $0, \tau, 2\tau, \ldots$, where τ is the data sampling interval. Data are input to the network sequentially starting before intercept and ending very shortly before intercept. The input to the neural network are the ratios of a segment's maximum radiance to the maximum radiance of the other segments, in turn. The target value is 1 or 0, representing either true or false targets for each particular training set composed of the ratios described above. The following techniques were used on this data-discrimination problem:

-A symmetric sigmoid function was used on all processing units. This appears to be advantageous in convergence and recognition performance.

-The data scaling procedure was revised so that the same scaling factor F_s was used on all data. The data were scaled and offset to be symmetric about 0 with range [-0.5:0.5]. The scaling factor F_s was calculated from the training set by Equation 1:

$$F_s = \max(S_{train}) - \min(S_{train}), \quad (1)$$

where S_{train} is the training data space. The training data are scaled and offset to be symmetric about 0 by Equation 2:

$$\text{network input} = (S_{train} - p_{mid})/F_s, \quad (2)$$

where $p_{mid} = \min(S_{train}) + F_s/2$.

2.2 Simulation Scheduling and Analysis

Two training schedules were used and tested on five different test sets. Each set contained more than 23 trajectories in time. Each test set contained a representative set of data with different aspect angles. These first training and test sets were for a 0.5 second sampling rate. Those schedules were as follows:

Schedule 1: This training schedule was to begin adapting weights after the first segment of data entered the network.[15] However, we did not allow data from a previous run to remain in the network at the same time that the data from a new run entered the network.

Schedule 2: A new training schedule was used in which data filled the network before weight adaptation began. Performance improved with this training schedule, as shown in Table 1. Fewer iterations of training were required (only 5,565 as compared with 14,323; see Table 2).

Table 1 - Successful identification rate of different test sets by using TDNN on 0.5 second increment data (percent).

Data Set	Schedule 2	Schedule 1
Training set	100	89.61
Test set #1	100	89.14
Test set #2	59.4	53.18
Test set #3	59.4	50.00
Test set #4	93.4	81.82
Test set #5	97.9	96.59

Table 2 - Overall performance of TDNN on 0.5 second increment data.

Overall	Schedule 2	Schedule 1
Iterations	5,565	14,323
RMSE*	0.044	0.105

* Root mean square error

For both training schedules, there was considerably less possibility of confusion of the components "fuel tank" or "tail" with the warhead in contrast with previous studies.[15] Usually the fuel tank and tail were clearly identified very early as not being the warhead. This is an improvement on previous results.

We also used various numbers of hidden units to see if performance would improve. The TDNN was trained to perform target discrimination with the scaling procedure described above and with training by Schedule 2. Table 3 summarizes its performance for three hidden units (3h') and five hidden units (5h'), as compared with our previous results labeled 3h (with three hidden units). Performance was better for the second, third, and fourth data sets.

Another experiment was performed to include additional data in the training set, with different aspect angles for different components. Previous training experiments were limited to data with the same aspect angle for different components. Performance improved on the third, fourth, and fifth data sets compared with the benchmark 3h run, but only the fifth data set showed an improvement over other runs with the revised scaling procedure. Performance for this experiment is labeled 3h'' in Table 3. As a whole, the network identified the warhead correctly in all cases. The results show that the smaller the aspect angle of the warhead, the more difficulty the network had in distinguishing the warhead as the target.

Table 3 - Comparison of discrimination performance (percent) with different data sets, various training methods, and different network topologies.

Data Set	3h'	5h'	3h	3h''
Training set	89.61	89.61	89.61	95.61
Test set #1	89.14	88.89	89.14	88.13
Test set #2	66.36	65.00	53.18	64.09
Test set #3	72.72	70.45	50.00	72.73
Test set #4	82.72	79.55	81.82	83.64
Test set #5	96.59	94.89	96.59	89.77

2.3 Improvements with Higher Sampling Rate Data

We found that improvements could be made in performance using data with a higher sampling rate. Whereas previous training and test sets were sampled in 0.5 second increments, the newer training and test sets were sampled in 0.1 second increments. When using 0.1 second data, the time delays at the first layer were initially set to (0.0, 0.1, 0.2, 0.3) seconds, and with the 0.5 second data, they were initially set

Intercept angle	wh aspect angle	ot aspect angle	ft aspect angle	tail aspect angle	ot broken	ft broken	tail broken
30°							
60°							
90°							
90°	90°	60°	60°	30°			
90°	60°	90°	90°	30°			
90°	30°	60°	90°	30°			
90°					front 1/3		
30°					front 1/3		
30°					back 2/3		front 1/3 c2*
90°					front 1/3		middle 1/3 c2*
90°					front 1/3	front 1/3	middle 1/3 c2*
30°					front 1/3		middle 1/3 c2*
90°	Later	sample	time				

Table 4. Variation of 0.1 second increment data.

to (0.0, 0.5, 1.0, 1.5) seconds. Time delays were evenly spaced on the hidden layer similarly.

We tested whether faster time sampling improved discrimination of the warhead (wh) from the other segments for more difficult data, described in Table 4, where different segments were at different aspect angles and the threat missile is broken into more than four segments by breaking the oxidizer tank (ot), fuel tank (ft), and/or tail themselves into several pieces.

We obtained 100% performance on the faster sampling data. Identification occurred much faster for the 0.1 second sampling than on the previous data set with 0.5 second sampling. In all cases, with the 0.1 second sample data, the activation became high rapidly for the warhead and low rapidly for the other segments.

For comparison, Figures 2-5 show some of the simulation results on the same scenarios but with two different sampling intervals: Figures 2a-5a show the results from the 0.1 second sampling interval; Figures 2b-5b show the results of the 0.5 second interval data.

All figures from 0.1 second interval data show quick and correct identification. Several of the scenarios with 0.5 second interval data had some confusion, at least at first, in the identity of the warhead parts on 0.5 second interval data, but with the 0.1 second data, this confusion was eliminated. Thus, the new network that depends on 0.1 second time sampling overcomes limitations in the previous network and data. We conclude that in the application of the time delay neural network, time sampling of every 0.1 second significantly improves performance in terms of speed of identification and percentage of correct recognition

2.4 Number of Time Steps versus Sampling Rate

In Section 2.3 we showed that data sampled at 0.1 second intervals gave superior performance to data sampled at 0.5 second intervals. We have explored the reason for this difference and tested the hypothesis that the total number of samples influences performance. There were twice as many samples in the 0.1 second data sets as in the 0.5 second data sets (the duration of the 0.1 second data set is two seconds, while the duration of the 0.5 second data set is five seconds).

We trained the network on only half of the time samples from the 0.1 second data. The resulting performance was still 100%, and the 0.1 second data still performed in a superior fashion compared to the 0.5 second data.

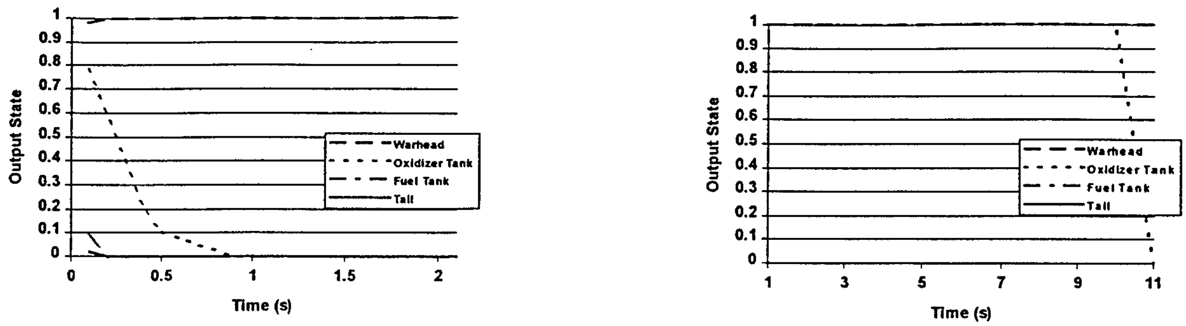


Figure 2. Recognizing warhead with intercept angle 30°; (a) Sampling time interval 0.1 s; (b) sampling time interval 0.5 s.

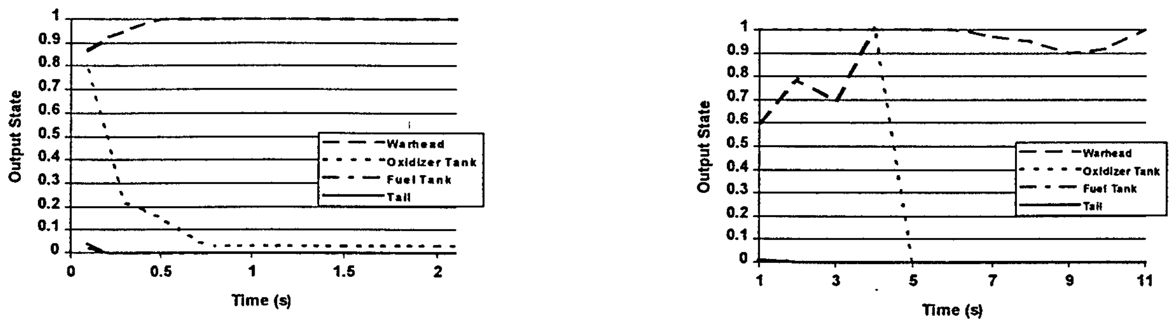


Figure 3. Recognizing warhead with intercept angle 60°; (a) Sampling time interval 0.1 s; (b) sampling time interval 0.5 s.

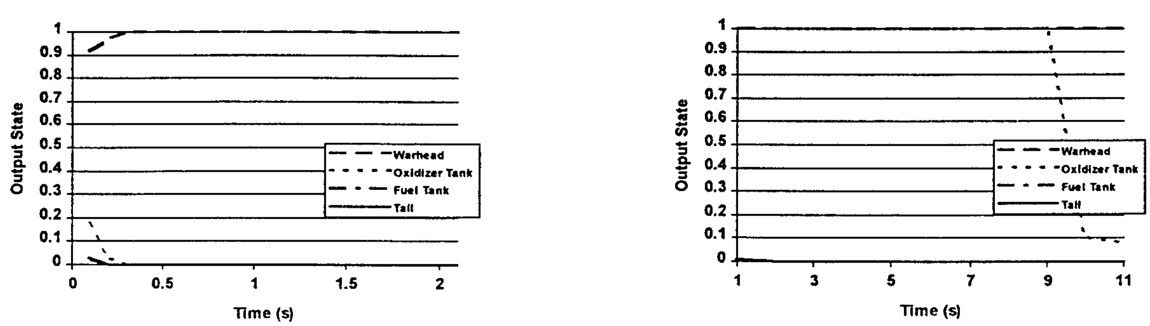
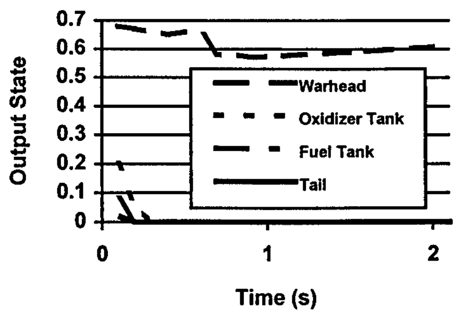


Figure 4. Recognizing warhead with intercept angle 90°; (a) Sampling time interval 0.1 s; (b) sampling time interval 0.5 s.



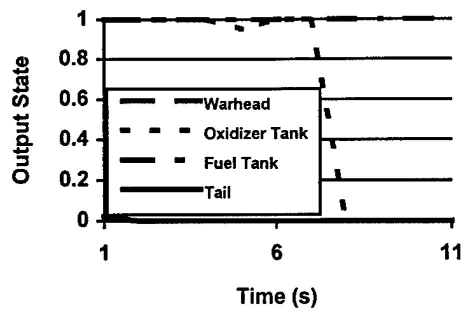


Figure 5. Recognizing target with intercept angle 90°; components 1 to 4 at aspect angle 30°, 60°, 90°, and 30°, respectively. (a) Sampling time interval 0.1 s; (b) sampling time interval 0.5 s.

The conclusion is that the number of samples is **not** the cause of superior performance in the 0.1 second data. The superior performance results from the higher sampling rate. The radiance data used for the analysis of 0.1 second increment data are much smoother than those used in the 0.5 second increment data. Furthermore, after seven seconds before intercept, the segments begin to take up more than one of the 512x512 pixels in the sensor. This causes a change in slope in the data when changing from one pixel to two pixels. During the 0.1 second increment data, these changes do not occur. The TDNN in this analysis performs well possibly because the data are smoother and the sensor stays locked on one pixel. More realistic two pixel data will be analyzed in the next section.

3.0 ATNN: Enhanced Performance

3.1 Earlier Discrimination on More Realistic Sensor Data

In Section 2.3, we showed that data sampled at 0.1 second intervals gave superior performance to data sampled at 0.5 second intervals with the TDNN. Next, the ATNN was applied to the data. Since the ATNN network needed a longer data scenario to allow more possibilities for time delays internal to the network, data was next obtained in 0.1 second intervals for a duration of four seconds (previous analyses were for a duration of two seconds). We applied both the ATNN and TDNN to this long data. The performance of the ATNN was superior to that of the TDNN. The reason for this improvement was that the ATNN has the ability to adapt time delays in addition to weights, whereas the TDNN only adapts weights (and leaves time delays fixed). Performance overall was very promising. For comparison, the ATNN's results (Figures 6a-8a) are shown with the TDNN's results (Figures 6b-8b) on the same observation angles, correspondingly.

The data consisted of three threat break-up scenarios, taken from 30°, 60°, and 90° trajectory intercept angles. For this newer data, the segment can move around on the pixels of the sensor so that the resulting measurements are noisy. The segment can be located either on one pixel or over two pixels (half of the segment lies in one pixel and half lies in an adjacent pixel). This scenario is more realistic than assuming the sensor can lock the segment on only one pixel at all times.[15] Thus, the data on which the neural network is trained and tested randomly jump up and down from one time step to the next. This is more realistic than the smooth data we worked with previously. The noise made the recognition problem more challenging for the network.

The TDNN was trained and tested on these bumpy data, and the result was not as good as before (with the smooth data). The TDNN results on the new data are shown in Figures 6b-8b; the TDNN could not resolve two pixel data with slope changes. The ATNN was then trained on the new data and tested. The ATNN provided more flexibility because time delay elements are adapted in the ATNN according to the characteristics of the data and better time delay values are chosen. Several time delays were increased to be over five time steps in the hidden layer.

The ATNN achieved 100% correct recognition for all three aspect angles. When the

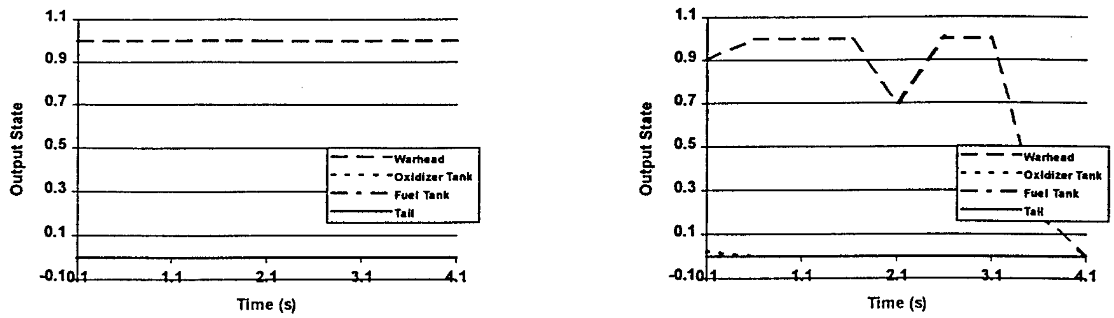


Figure 6. Performance on bumpy data with intercept angle 90°, sampling time interval 0.1 s. (a) ATNN; (b) TDNN

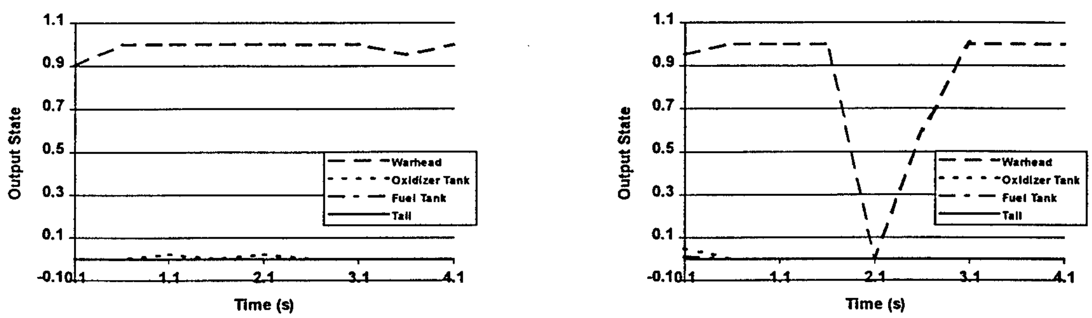


Figure 7. Performance on bumpy data with intercept angle 60°, sampling time interval 0.1 s. (a) ATNN; (b) TDNN

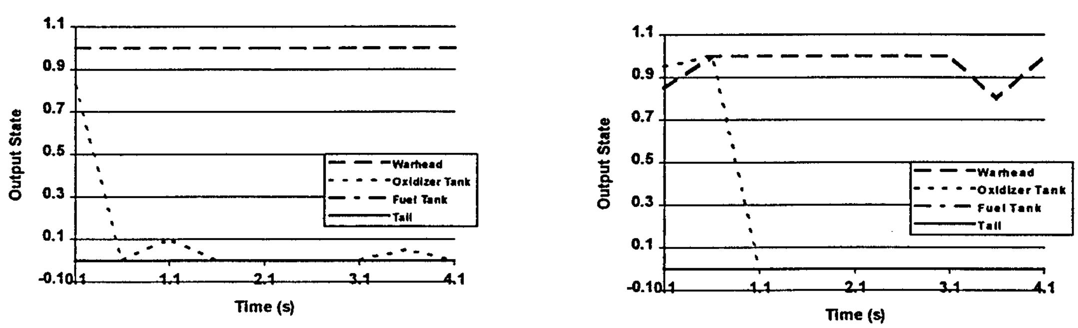


Figure 8. Performance on bumpy data with intercept angle 60°, sampling time interval 0.1 s. (a) ATNN; (b) TDNN

networks were given the entire time (four seconds) to discriminate the warhead and the discrimination was performed at the end of the data stream, the ATNN performed with 100% correct discrimination whereas the TDNN had only 33% discrimination at the end of the data stream.

It is important to consider a scenario in which the network is required to identify the warhead before the end of the data stream. Thus, we computed a stepwise performance percentage, which consists of the percent correct recognition averaged over all possible time steps in the data. On this basis, the ATNN achieved a 99.46% correct recognition rate whereas the TDNN had 85% correct identification over all possible time steps. This means that if the ATNN were requested to make a decision at any time during the incoming data stream, it would have been correct 99% of the time. Early identification is thus made possible with the ATNN. The performance of the ATNN was far superior to that of the TDNN and was far more resilient to noise in the data.

3.2 Environmental Variations and Robustness

Two of the obscurities that degrade the performance of automatic target recognition are environmental changes and noise. Sources of changes and noise include blurred images, camera vibrations, heat atmospheric effects, and more. The effects of noise and methods to perform recognition in spite of noise will be important for eventual deployment of this technology. Current target recognition systems are unable to modify their behavior on the basis of the dynamic environmental changes occurring around them. In order to perform robustly, the system must be able to adapt to this dynamic environment while maintaining acceptable performance[1]. We evaluated our neural network's performance in the presence of noise, and improved the neural network's robustness by training on noisy data.

To study the effects of noise on performance, we used various scenarios in which noise was added during training or recall or both. Simulation results show that the ATNN is robust to moderate amounts of noise and that its robustness improves when training is done with noisy data. This approach is important for eventual implementation in a real-world environment, where substantial noise is expected. The data set is based on a simulation of a more realistic scenario than previous data sets, in which the scenario includes 0.1 second samples with some noise added during simulation by the missile simulator. These initial results showed that the ATNN recognized the warhead in spite of noise during the simulation. We are concerned, however, that the real world environment would have additional noise; the initial runs did not consider the effects of such additional noise. We address these effects here.

Our evaluation of noisy scenarios first required simulated data consisting of a simulated threat scenario. We chose to start with noisy simulated data and to add various amounts of additional noise. Thus, sources of noise include the simulator or addition of noise to simulated data. Two different types of training scenarios were tested:

1. The ATNN was trained with data direct from the simulation.
2. The ATNN was trained with noisy data consisting of simulation data with additional noise added.

The amount of additive noise was varied in each scenario.

The relationship between the original raw data (simulated) and the noisy data is

$$r_n = \text{radiance} + K \times 0.001 \times N_r \qquad (3)$$

where K varies from 1 to 10, N_r denotes a normal distribution random number with a mean of zero, a range of -0.5 to 0.5, and a variance of 1.0, and r_n is an instance of radiance data with additional noise added.

First, the network was trained on simulated data and tested on data with additional noise added. For each value of K (i.e.., each level of additive noise), ten runs were performed. Table 5 shows the results. In general, the performance degraded as the amount (K) of noise increased. Figure 9 shows the property of graceful degradation, as performance is quite high (98-100%) until K>2.

Next the network was trained with data that included additive noise. This data set contained the original simulated data in addition to the data with additive noise.

The trained ATNN was again tested on various amounts of noisy data as described above. The results of 10 runs are shown in Table 6. Compared with the average performance in Table 5, the network trained with

Run	K=0	1	2	3	4	5	6	7	8	9	10
1	99.77	100.0	98.64	87.16	75.00	25.00	31.30	25.00	75.00	25.00	25.22
2	99.77	99.09	95.27	25.00	25.00	75.00	75.00	25.22	75.00	75.22	74.77
3	99.77	99.77	99.09	91.21	75.00	25.00	75.00	25.00	74.77	25.00	74.32
4	99.77	99.32	97.52	96.39	76.12	75.00	75.00	25.00	74.77	75.00	36.93
5	99.77	100.0	99.54	89.86	75.00	75.00	75.00	25.45	80.63	75.00	25.00
6	99.77	99.32	96.62	85.36	25.00	75.00	84.00	75.00	25.22	75.00	25.00
7	99.77	99.54	96.62	80.85	83.78	75.00	75.00	75.00	25.22	25.00	75.00
8	99.77	99.77	98.42	97.74	75.00	68.69	25.00	25.00	75.00	25.00	25.00
9	99.77	100.0	98.87	92.34	25.00	75.00	25.00	75.00	75.00	26.57	75.00
10	99.77	99.54	100.0	75.22	75.00	25.00	75.00	25.00	75.00	75.00	76.57
Av	99.77	99.63	98.06	82.11	60.99	59.36	61.53	40.06	65.56	50.18	51.28

Table 5. Performance of ATNN trained on pure data and tested with various amounts of noise added

Run	K=0	1	2	3	4	5	6	7	8	9	10
1	94.94	93.18	96.71	96.21	87.62	72.47	77.02	73.48	30.30	69.44	66.66
2	94.94	94.69	92.17	79.79	72.22	54.79	30.05	31.31	50.25	70.95	26.76
3	94.94	95.95	95.95	96.21	64.39	72.72	29.79	73.48	72.22	77.02	59.34
4	94.94	95.45	95.45	94.19	87.12	66.66	78.28	28.28	71.46	30.80	66.16
5	94.94	96.96	96.96	91.41	90.30	83.83	77.77	75.25	73.48	39.14	61.61
6	94.94	94.19	94.69	90.65	86.86	32.82	80.80	71.46	65.90	31.56	68.93
7	94.94	94.44	93.18	92.17	90.40	41.91	57.82	76.01	72.72	69.19	28.53
8	94.94	94.44	95.45	95.20	28.53	84.09	73.98	71.21	27.27	67.42	70.20
9	94.94	95.45	95.20	94.69	83.58	63.38	71.21	26.26	30.30	65.65	35.85
10	94.94	94.44	96.46	85.60	83.33	91.96	30.30	66.41	31.56	53.78	76.01
Av	94.94	94.92	95.22	91.61	76.43	64.46	60.70	59.31	52.55	57.50	56.01

Table 6. Performance of ATNN trained on noisy data and tested with various amounts of noise added

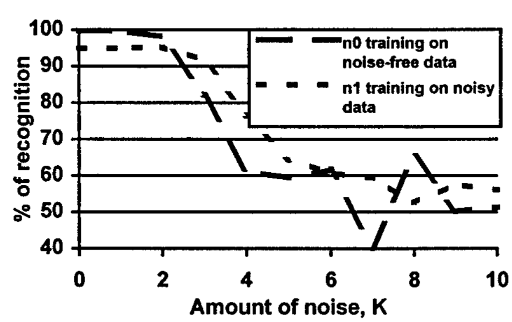


Figure 9. Performance of the ATNN as a function of the amount of noise in the data. For high noise environments, performance is superior when training is done on noisy data.

additive noise improved the identification capability. The comparison is shown in Figure 9. For low noise ($K \leq 2$), there was slightly lower performance when the network was trained on noisy data. For higher noise ($2 < K < 8$), better performance was attained when the network was trained with additive noise. For even larger amounts of noise ($K \geq 8$), there was less jitter in performance when the network was trained with additive noise, i.e., a more stable performance. We can conclude that the ATNN is more robust in noise situations when it is trained with additive noise. Similar studies and conclusions can be found in Reference 16. The lower range of noise, where the networks performed better, are not likely to occur in the real world.

Conclusions

High performance in discrimination is presented here using a neural network approach. The TDNN and ATNN were shown to be promising in recognizing threat warheads regardless of different threat types, different aspect angles, tumbling situations, or the break-up of the threat into more than four segments. The recognition performance of the ATNN outstrips that of the TDNN. Furthermore, the ATNN possesses much earlier discrimination and noise resilience. We have suggested that the ATNN is well suited for spatio-temporal and temporal domains of problems because it can handle multiple channel data and correlate relationships among these data.[17]

Specifically, we have shown that for a threat in an exo-atmospheric trajectory, and with a single sensor, the following conclusions can be made:

1. We have previous results that the TDNN is capable of high performance for discrimination of warhead versus other threat missile segments.
2. Performance can be improved by appropriate scaling and training schedule.
3. Higher sampling rate data gave better discrimination performance.
4. Training the TDNN with adaptive time delays (i.e., as an ATNN) enhanced performance in the trained networks.
5. Training with noise produced a network that is more robust to variations in real world scenarios.

Future work should include extending the discrimination approach by developing one or more neural networks that can identify a warhead regardless of the type of threat.

References

1. Roth, M. W., "Survey of neural network technology for automatic target recognition," *IEEE Trans. On Neural Networks*, **1** (1), 28-43 (1990).
2. Gorman, R. P. and Sejnowski, T. J., "Analysis of hidden units in a layered network trained to classify sonar targets," *Neural Networks*, **1,** 75.89 (1988).
3. Bai, B., and Farhat, N. H., "Learning networks for extrapolation and radar target identification," *Neural Networks*, **5,** 507-529 (1992).
4. Waibel, A., Hanazawa, T., Hinton G., Shikano, K., and Lang, K., "Phoneme recognition: Neural networks versus hidden Markov models," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 107-110 (April 1988).
5. Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K., "Phoneme recognition using time-delay neural networks," *IEEE Trans. Acoust., Speech, Signal Processing,* **37**(3), 328-339 (1989).
6. Waibel, A., Lang, K.J., and Hinton G.E., "A time-delay neural network architecture for isolated word recognition," *Neural Networks*, **3**, 23-43 (1990).
7. Unnikrishnan, K.P., Hopfield, J.J., and Tank, D.W., "Connected-digit speaker-dependent speech recognition using a neural network with time-delayed connections," *IEEE Trans. Signal Processing*, **39**(3), 698-713 (1991).
8. Lin, D.-T., Dayhoff, J.E., and Ligomenides, P.A., "Trajectory recognition with a time delay neural network, " in *Int. Joint Conf. On Neural Networks*, IEEE, New York, Vol. 3, pp. 197-202 (1992).
9. Lin, D.-T., Dayhoff, J.E., and Ligomenides, P.A., "Adaptive time delay neural network for temporal correlation and prediction," in *Intelligent Robots and Computer Vision XI: Biological, Neural Net, and 3-D Methods, Proc. SPIE* Boston, **1826**, 170-181, (Nov. 1992).
10. Lin, D.-T., Dayhoff, J.E., and Ligomenides, *A Learning Algorithm for Adaptive Time Delays in a Temporal Neural Network,* Technical Report SRC-TR-92-59, Systems Research Center, University of Maryland, College Park, MD (15 May 1992).
11. Day, S.P., and Davenport, M.R., "Continuous-time temporal back-propagation with adaptive time delays," *IEEE Trans. Neural Networks*, **4**(2), 348-354 (1993).
12. McClelland, J.L., Rumelhart, D.E., and the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition,* Vol. 2, MIT Press, Cambridge, MA. (1986).
13. Day, S.P., and Davenport, M., "Continuous time temporal back-propagation with adaptive time delays," Neuroprose archive, Ohio State University. Accessible on Internet via anonymous ftp on archive.cis.ohiostate.edu, in pub/neuroprose/day.tempora.ps (Aug. 1991).
14. Resch, C.L. *New Time Delay Neural Network to Distinguish Exo-atmospheric Warheads,* Technical Report AM-93-E141, The Johns Hopkins University Applied

Physics Laboratory, Laurel, MD (Aug. 1993).

15. Resch, C.L., *Effects of Jitter on the Ability of a Time Delay Neural Network to Distinguish Exo-atmospheric Warheads,* Technical Report AM-94-E010, The Johns Hopkins University Applied Physics Laboratory, Laurel, MD (Jan 1994).

16. Holmstrom, L., and Koistinen, P., "Using additive noise in back-propagation training," *IEEE Trans. Neural Networks,* **3**(1), 24-38 (1992).

17. Lin, D.-T., *The Adaptive Time-Delay Neural Network: Characterization and Applications to Pattern Recognition, Prediction, and Signal Processing,* Ph.D. thesis, University of Maryland at College Park (1994).