Automatic Recognition and Assignment of Missile Pieces in Clutter

Cheryl Resch*, Fernando J. Pineda, and I-Jeng Wang
Johns Hopkins University Applied Physics Laboratory
{cheryl.resch, fernando.pineda,I-jeng.wang}@jhuapl.edu

Abstract

The ability to discriminate a reentry vehicle (RV) from booster parts and other debris is critical to theater ballistic missile defense (TBMD). As it travels along its trajectory, a threat missile separates into a reentry vehicle (RV) and clutter. The latter consists of several tanks, separation debris and fragments of hot fuel. Interception of the RV requires discrimination of the RV from the clutter. The required discrimination must be performed no later than 30 seconds before intercept. A time-delay neural network (TDNN) is proposed for discrimination of the RV from other missile parts debris. The rate of change of the IR signature over several seconds is used as a discriminant. The performances of two different approaches are compared: 1) A TDNN that employs back-propagation weight updates is used to calculate activation levels for output nodes. The RV is selected via winner-take-all. This TDNN has been previously described in Reference [1]. 2) A TDNN that updates weights using a cross-entropy error function with a softmax activation function is used to estimate assignment probabilities. The RV is subsequently selected via a probabilistic assignment algorithm that imposes the constraint that there can only be a single RV. We found that the TDNN employing back propagated softmax learning performed better than the TDNN employing back propagated least mean square learning.

Introduction

During its trajectory, a threat missile splits into an RV (with payload), the booster tank and the ACM. Separation debris and bits of hot fuel are also generated. The RV contains the warhead; the RV must be discriminated from other missile pieces and debris before it can be intercepted and prevented from hitting its target. An infrared (IR) sensor is used to guide the interceptor to the threat during the terminal homing phase. The data to be classified are features extracted from the irradiance of the missile pieces or debris incident on an IR sensor on board an interceptor.

The target must be chosen about 30 seconds before intercept. Figure 1 shows an IR image four seconds before intercept.

Figure 1 - IR Image 4 Seconds Before Intercept, 256 x 256 pixels, 3° Field of View

Thirty seconds before intercept, the threat complex is about 100 km away from the IR sensor. At this range, even relatively large pieces are represented as single pixels on the IR image. Larger pieces cover a larger portion of the pixel, so they appear as brighter pixels. Hot pieces return a larger irradiance to the sensor, so they look brighter than cooler pieces. Thus, hot and small pieces of fuel may be confused with cooler and larger missile pieces at this range. As a result, static IR images are insufficient for discrimination. Temporal characteristics serve as a potential discriminant because different pieces cool at varying rates, thereby causing their IR signatures to change at different rates. We are using the rate of change of the IR signature over several seconds as a discriminant[1]. The performance of two different

* corresponding author

approaches is compared. 1) A TDNN with a least mean squares error function, a linear activation function and a sigmoid transfer function is used to calculate output values. The RV is selected via winner-take-all. 2) A TDNN with a cross-entropy error function and a softmax activation function is used to estimate assignment probabilities. The RV is subsequently selected via a probabilistic assignment algorithm that imposes the constraint that there can only be a single RV.

Method

Irradiance is the derivative of the radiative power incident on the sensor with respect to area; it is proportional to the square of inverse of the range from the sensor to the target. Thus, the irradiance returned to the interceptor's IR sensor increases as the missile fragments get closer to the interceptor, so irradiance is time-dependent. If the irradiance value returned to the sensor by a piece is used as a feature, the discrimination routine will not be robust to changes in range and time-to-go before intercept. Therefore, the fraction of the total irradiance returned to the sensor that each missile fragment represents is used as a feature. The second feature input to the TDNN is the rate of change of the fraction of irradiance. The warhead will cool at a different rate than debris or empty tanks. The fraction of the irradiance represented by the RV should change relative to that of debris or empty tanks. The debris pieces have the smallest rate of change of fraction of irradiance, followed by RVs, followed by tanks. The final values for this feature are obtained by taking the logarithm of the rate of change of irradiance. Taking the logarithm exaggerates the difference between debris and RVs.

A time-delay neural network (TDNN) is similar to a back-propagation neural network, except that the TDNN keeps track of data in the time domain[2]. During each iteration, data for the current time step and the previous six time steps are used for classification. Figure 2 shows a schematic of a TDNN. After a set of inputs is introduced and fed forward through the neural network, the input is sent to a time-delay node (t-Δt) for use during the next iteration. After subsequent iterations, information in the time-delay nodes is sent to subsequent time-delay nodes, for example, from (t-Δt) to (t-2Δt). Each input time-delay layer is fully connected to the hidden layer. Each hidden time-delay layer is fully connected to the output layer.

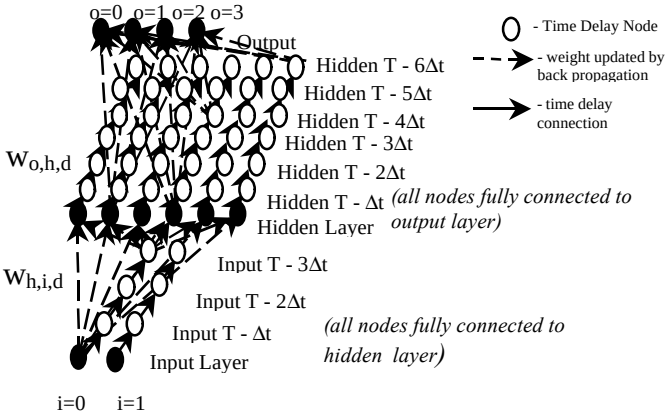

Figure 2 - Schematic of TDNN

Mean Squared Error Training

Error functions for neural networks arise from the principle of maximum likelihood. The likelihood may be written as

$$L = \prod p(x,t) = \prod p(t \mid x)\, p(x) \tag{1}$$

where x is the input data and t is the desired output. Minimizing the negative logarithm of the likelihood is equivalent to maximizing the likelihood. Therefore, a basic form of the error function is as follows.

$$E = -\ln(L) = -\sum \ln(p(t \mid x)) - \sum \ln(p(x)) \tag{2}$$

The second term in this equation does not depend on the neural network parameters, so can be dropped from the error function minimized by the neural network. Thus we minimize the following.

$$E = -\sum \ln(p(t \mid x)) \tag{3}$$

If the errors are assumed to be normally distributed, then

$$p(t \mid x) = 1/(2p s^{2})^{1/2} \exp(-(y-t)^{2}/2s^{2}) \tag{4}$$

where y is the output from the neural network and σ is the standard deviation of the data. Thus, the error function is as follows.

$$E = 1/(2s^{2})\sum (y-t)^{2} + Nc\ln(s) + N(c/2)\ln(2p) \tag{5}$$

The second and third terms and the standard deviation do not depend on neural network parameters, so the error function to be minimized can be expressed as follows.

$$E = 1/2 \sum (y - t)^2 \qquad (6)$$

Thus minimizing the squared error is optimal for Gaussian distributed errors. This leads to a convenient form for the weight update function for the output nodes.

$$\begin{aligned} d &= g' \partial E / \partial y \\ &= g'(y - t) \end{aligned} \qquad (7)$$

where g is a transfer function. The simplicity of the sum of the squares error function makes it very popular for back propagation neural networks.

Normalized Exponential (Softmax) Error Function

Consider the case where data have a 1 of N coding scheme with one output node for each class. For each set of inputs one of the output nodes should have value of one and the others should have a value of zero. The value of the conditional probability for this type of data can be expressed as

$$p(t \mid x) = \prod_k (y_k)^{t_k} \qquad (8)$$

where y_k is class k's output node activation and t_k is that node's target output (one or zero). Taking the negative logarithm as before, the error function is

$$E = -\sum_{k=1}^{N} t_k \ln(y_k) \qquad (9)$$

If the output activation levels of the network are to be interpreted as probabilities, they must lie in the range (0,1) and they must sum to one. To accomplish this, the logistic sigmoid activation function is used:

$$y_k = g(a) = \exp(a_k) / \sum_{k'=1}^{N} \exp(a_{k'}) . \qquad (10)$$

where a_k is the activation level for output node k. Again, the weight update function is

$$d_k = g' \partial E / \partial y_k \qquad (11)$$

The error function, E, and the activation function both contain a summation over all the output nodes. Thus, all output nodes need to be considered in order to evaluate the weight update function.

$$d_k = \sum_{k'=1}^{N} \partial E / \partial y_{k'} \partial y_{k'} / \partial a_k \qquad (12)$$

The expression for the partial derivative of $y_{k'}$ with respect to a_k is [3],

$$\begin{aligned} \partial y_{k'} / \partial a_k &= \partial(\exp(a_{k'}) / \sum_{k''=1}^{N} \exp(a_{k''})) / \partial a_k \\ &= -y_{k'} y_k \quad for \quad k \neq k' \\ &= y_k - y_k^2 \quad for \quad k = k' \end{aligned} \qquad (13)$$

Next,

$$\begin{aligned} \partial E / \partial y_{k'} &= \partial(-\sum_{k=1}^{N} t_k \ln(y_k)) / \partial y_{k'} \\ &= -t_{k'} / y_{k'} \end{aligned} \qquad (14)$$

Substituting equation (13) and equation (14) into equation (12) results in

$$\begin{aligned} d_k &= \sum_{k'=1}^{N} (-t_{k'} / y_{k'}) \partial y_{k'} / \partial a_k \\ &= \sum_{k \neq k'} (-t_{k'} / y_{k'})(-y_{k'} y_k) + (-t_k / y_k)(y_k - y_k^2) \end{aligned} \qquad (15)$$

Since $t_{k'} = 0$ when $k \neq k'$ and $t_k = 1$,

$$d_k = y_k - t_k \qquad (16)$$

Thus, using the logistic sigmoid activation function also results in a simple form of the weight update function.

Training and Testing

The data include simulated and measured data. The simulated data consist of 350 cases for which the missile pieces have varying tumble rate, spin rate, coning angle, trajectory, et cetera. Three different types of scenarios are created. 200 of the simulated scenarios contain one RV, one booster tank, one ACM, and one piece of debris. 100 of the simulated scenarios contain one RV, either one booster or one ACM, and two pieces of debris. The remaining 50 simulated scenarios contain one RV and three pieces of debris. Thus, the classifier should be robust to the number of each type of piece in the scenario. Each scenario consists of 25 time steps in 0.1 second increments starting 60 to 30 seconds before intercept.

The measured data are more limited and consist of eight flights of a missile that expelled hot fuel. This data consist of IR signatures of the RV, a tank, and hot fuel debris. Figure 3 shows the simulated data and flight data for all time steps.

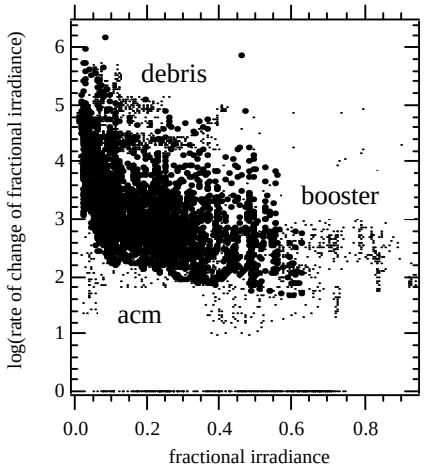

Figure 3 - Missile Piece Features Large dots are RV features, smaller dots are ACM, Booster, or debris.

The TDNN is trained using the eight flight cases and 250 of the simulated scenarios. 100 of the simulated scenarios contain one RV, one booster tank, one ACM, and one piece of debris. 100 of the simulated scenarios contain one RV, either one booster or one ACM, and two pieces of debris. The other 50 simulated scenarios contain one RV and three pieces of debris.

Since it is most important that the TDNN properly identify the RV, the training set is set up so that the RV is the most prevalent item. The training set for each scenario contains the data for the RV, and the data for one other piece; the training set contains 258 RV realizations, 137 debris realizations, 63 booster realizations, and 58 ACM realizations. The TDNN is tested using leave-one-out cross validation. For the TDNN trained using a least mean squares error function, the RV is selected via winner-take-all. The outputs from the TDNN trained using the softmax activation function represent probabilities, which are input to an assignment algorithm. The assignment algorithm determines which of the two observations is more likely to be the RV. Since for each scenario the only data included in the training set are that for the RV and one other piece, the constraints for the assignment algorithm are simple: one observation is an RV, the other is not.

The TDNN employing the softmax activation function is then tested on the 100 simulated cases that are not included in the training set. All of these cases include one RV, one ACM, one booster, and one piece of hot fuel debris. Data for all four pieces are input to the TDNN. The probabilities for the four pieces output from the TDNN are then used by an assignment algorithm to assign observations to the RV, the booster, and the ACM. The constraints for the assignment algorithm are that there is one RV, one ACM, and one booster.

Results

Figure 4 shows the results of leave-one-out cross-validation on the 258 cases as a function of time step. The TDNN that uses a softmax error function produced better results than the TDNN that uses the least mean squares error function. Only 7% of the ACMs are incorrectly classified as RVs for the TDNN trained using a softmax error function, while about 40% of the ACMs are incorrectly classified as RVs for the TDNN trained using least mean squares. For both TDNNs, leass than 5% of boosters or debris are misclassified as RVs.

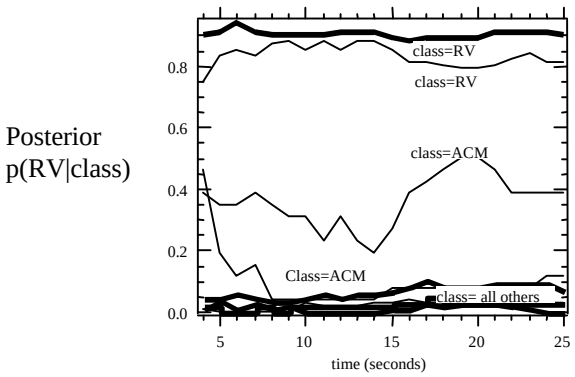

Figure 4 – Results of Leave-one-out Cross-Validation for 258 cases. Probability that a given piece type will be classified as an RV when there are two pieces to choose from. Thick lines use softmax error function, thin lines use least mean squares.

Figure 5 shows the results of testing on 100 cases for which all four observations for each scenario are included in the test set. The fraction of the cases for which a particular piece is chosen as the RV is plotted as a function of time step. At the last time step, 85% of the RVs are classified as an RV. 15% of the ACMs are incorrectly classified as RVs. None of the boosters or debris were classified as RVs.

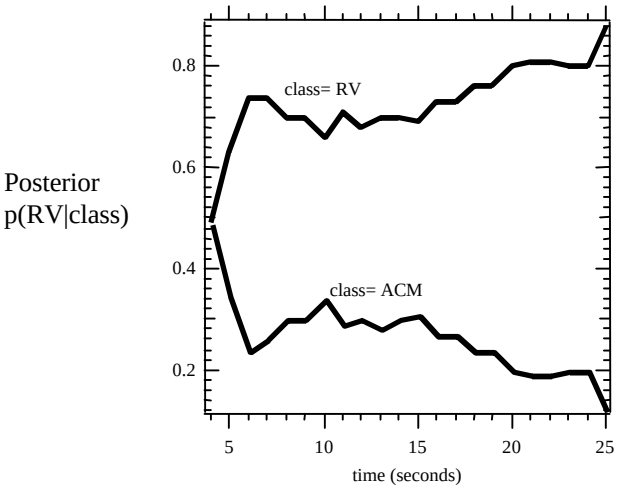

Figure 5 - Results of Testing on 100 Cases with 4 Observations Each. Probability that a given piece type will be classified as an RV when there are four pieces to choose from.

Conclusions

An attempt has been made to build an IR signature classifier for Theater Ballistic Missile Defense that is robust to range, number and types of objects, and dynamic parameters of objects. IR signature classifiers currently proposed require a priori knowledge of these parameters[5].

A TDNN classifier based on softmax training is used to estimate assignment probabilities. The TDNN and assignment algorithm combination incorrectly classifies about 7% of ACMs as RVs for test sets with two observations per case, and incorrectly classifies about 15% of RVs for test sets with four observations per case. The TDNN classifier did not incorrectly classify boosters or hot fuel debris. Currently implemented classifiers incorrectly classify about 10% of ACMs as RVs [5]. These classifiers, however, require a priori knowledge of the range and time to go before intercept.

The TDNN trained using a least mean squares error function incorrectly classified about 40% of ACMs as RVs for test sets with two observations per case. This indicates that the time histories of the features are not Gaussian distributed and a softmax error function is more appropriate than an error function based on least mean squares. Also, the 1 of N scheme of softmax procedure is appropriate for the problem of choosing one RV from among the missile pieces and debris.

References:

1. Resch, Cheryl L. "Exo-atmospheric Discrimination of Thrust Termination Debris and Missile Segments," Johns Hopkins APL Tech. Dig., Volume 19, Number 3, 1998.

2. Lin, D., Dayhoff, J. E., and Ligomenides, P. A., "Trajectory Recognition with a Time-Delay Neural Network," in *Proc. International Joint Conference on Neural Networks*, pp. III-197-III-202, Baltimore, MD, 1992.

3. Bishop, Christopher M. Neural Networks for Pattern Recognition, Oxford University Press, New York, 1995.

4. Bridle, John S., "Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition," Fogelman-Soulie, F. and Herault, J. eds. Neurocomputing: Algorithms, Architectures and Applications pp. 227-235, Springer-Verlag, 1990.

5. Silberman, G. L., "IR Selection Results for the NTW Trade Studies," JHU/APL A1F(1)-99-U-029, March 1999.