A Modified Recurrent Cascade-Correlation Network for Radar Signal Pulse Detection

N. Karunanithi
2E-378, Bellcore
445, South Street
Morristown, NJ 07692
Phone: (201) 829-4466
Fax: (201) 829-5888
karun@faline.bellcore.com

D. Whitley
Dept. of Computer Science
Colorado State University
Fort Collins, CO 80523
whitley@cs.colostate.edu

D. Newman
Texas Instruments
5825 Mark Dabling Blvd.
MS 3712
Colorado Springs, CO 80919

1 Problem Statement

The signal detection problem that we studied involves identifying radar signal pulses in one of several channels that span a frequency range. The signals are such that a valid signal causes "false signals", or "splatter" to appear in surrounding channels. These "false signals" should not be detected. Also it is necessary that there should not be a "false alarm" whenever there is no signal. The problem is further complicated by the fact that more than one valid signal may simultaneously exist across multiple channels.

The input signal for the detector is a logarithmic envelope of the amplitude of prefiltered outputs of radar returns. The actual pulse may occur anywhere within the envelope and can be of variable duration. There are 24 basic "signal events" in our data set and each signal event can occur somewhere across 21 channels (labeled 1 through 21). Of the 24 basic signal events, 10 are multiple pulse cases while the remaining 14 are single pulse cases. A signal event is called a *single pulse* (SP) if the pulse occurs on only one of the 21 channels at any given instance. A signal event is said to be a *multiple pulse* (MP) case if the useful pulses occur simultaneously on more than one channel. All signal events are of equal duration and each signal event is represented in terms of approximately 2000 equally spaced samples that are 1 nanosecond apart. (However, the actual pulses within the signal events can be of variable duration.) If all 21 channels are considered then all signal events can be represented in terms of 1.008 million time slices. The signal space can be further characterized as highly sparse because: 1) not all signal events contain target pulses, and 2) only a small fraction of the samples within a channel contain useful information (i.e., those samples that represent the region where the pulse is present). The problem requires not only identifying whether the pulse is present (amplitude detection) in the signal event but also to locate the position as well as the duration of the pulse within the channel. There are several outputs in the signal detector corresponding to the width of the pulse, time of arrival and frequency information. The outputs from the detector are then fed to a postprocessing unit for further analysis. The objective of this study was to develop a connectionist network that can detect the presence of pulse(s) in signal events.

Over the past 3 years we have trained neural networks for the radar signal pulse detection problem using 1) traditional back-propagation (Rumelhart, Hinton and Williams 1986), 2) a genetic hill-climbing algorithm (Whitley et al. 1990), 3) a non-recurrent Cascade-Correlation learning architecture (Fahlman and Lebiere 1990) and 4) a variant of Jordan-style recurrent net (Jordan 1986) developed by the Cascade-Correlation algorithm. The back propagation algorithm was a standard form using momentum. Our empirical data indicate that the Cascade-Correlation algorithm produces superior results compared to the other two algorithms, both in terms of learning speed (approximately 50 times faster that the backpropagation) and in terms of generalization (Whitley and Karunanithi 1991; Whitley 1992). Here we present results from only the Cascade-Correlation correlation algorithm.

2 Experimental Approach

2.1 A Training Data Selection Method

In this study, it is assumed that all 24 signal events occur on either channel 10 or channel 11. Thus, it became necessary to test for all 24 signal events on both channels 10 and 11. (Note that this results in a total of 48 signal events to be tested.) Among 24 signal events on each channel, only 8 were used for selecting training data because they typify most of the peculiarities present in the signal space. Of these 8 selected signal events 6 were single pulse cases and the remaining 2 were multiple pulse cases. Selecting samples from these selected signal events at random may not produce a meaningful training set because majority of the samples belong to noninformative regions. In order to study the influence of training data selection method, five training sets (of sizes 364, 434, 546, 634 and 815 respectively) were constructed by manually selecting samples from both transition and nontransition regions. Furthermore, the number of samples from each signal event was also varied depending on the duration and nature of the pulse as well as the noise characteristics of the signal events.

Since the signals occurring on a particular channel can be affected by the interference from the adjacent channels, an equal number of samples from the two side channels (one above and one below) were also included in the training set. These samples could be helpful in identifying "splatters". In the single pulse case, channel 11 was considered as the main channel and channels 10 and 12 as the side channels whereas in the multiple pulse case, both channels 10 and 11 were considered as the main channels. (In the multiple pulse case, if channel 10 was considered as the main channel then channels 9 and 11 would act as the side channels; on the other hand, if channel 11 was the main channel then channels 10 and 12 would be the side channels.) Some of the signal events are shown in solid lines at the top of Figures 1 through 4. The dash lines at the bottom of these figures represent the target pulses.

2.2 Network Models Used

In order to evaluate the generalization performance of connectionist networks, both feedforward networks and a variant of Jordan style recurrent networks were examined. The input layer of the feedforward network had 8 inputs (1 signal + 7 delays) corresponding to the main channel and 4 inputs (1 signal + 3 delays) for each side channel. The output layer of the networks had six units corresponding to the width of the pulse, time of arrival and frequency information. The tapped delay line was added to the input because the actual hardware implementation of the signal detector would incorporate delay lines on the input terminals.

Since the training sets have more samples from the transition regions, the feedforward network was successful in detecting both the rising and falling edges of the pulse but had difficulty in recognizing the middle of the pulse, especially if the pulse had a long duration. This problem can partially be addressed by using a training set that has a large number of midpulse samples. However, if the network has to detect pulses based only on amplitude information and if there are two or more inputs that are similar but their outputs are opposite then it is possible for the network to produce an incorrect output. Increasing the size of the training set is of no avail because the feedforward networks cannot perform one-to-many mapping. To address this situation, the network must develop contextual information in the form of memory based on the previous state (or, output) values. The recurrent network models can be used to address this issue.

One logical choice for the recurrent network model would be to use Elman's recurrent network (Elman 1990) which is simple and is capable of developing memory based on hidden unit activations of the previous time steps. However, the Elman network can produce meaningful result only when the samples are presented in a continuous fashion. Hence the sampling strategy used for constructing the training sets precludes the use of Elman network because samples were selected only from a subset of discrete locations. So, as an alternative we used a variant of Jordan style recurrent networks with "teacher forced" training outputs. In this network the recurrent connections from the output layer were fed to the hidden units. All the recurrent connections had a fixed weight of strength 1.0. Under teacher forced training, the teacher outputs at the time $t-1$ were used as the feedback at time t. The output of the network at time t was a function of the current input and the previous output. Thus, training this style of Jordan network is equivalent to training a feedforward network in which the input consists of both the actual input at time t and the target output at time $t-1$. The resulting Jordan network had 16 input units for the tapped delay inputs and 6 additional input units for the feedback from the output layer. However, when a Jordan network is tested for generalization the actual output of the network (not the target output) are recirculated as the input to the feedback input units. In this study, both the feedforward network and the Jordan style networks were developed and trained using Fahlman et al's Cascade-Correlation algorithm (Fahlman et al. 1990).

2.3 Hardware Implementation Consideration

Since the Cascade-Correlation algorithm adds one unit for each hidden layer and each hidden unit receives fan-in connections from all the earlier hidden units as well as the input layer the resulting networks can be very deep and may not be appropriate for analog hardware implementation because of the noise problem. To facilitate hardware implementation, a modification was made to the Cascade-Correlation algorithm such that it builds a 2 hidden layer network instead of a deep multilayer network (Karunanithi 1992). The modified algorithm constructs networks by adding a predefined number of hidden units to the first hidden layer and as many units in the second hidden layer as needed. In the experiments reported here, approximately 20 hidden units were added to the first hidden layer. (The size of the first hidden layer was empirically determined by trial-and-error.) While constructing hidden layers, the algorithm adds hidden units only in a lateral fashion. Thus, the hidden units in the first hidden layer receive fan-in connections only from the input layer and the units in the second hidden layer receive fan-in connections from both the input and the first hidden layers. (We also experimented with an another version of the Cascade-Correlation algorithm which developed nets with 1 hidden layer. However, the resulting generalization and the size of the network were not satisfactory.)

3 Results

In order to evaluate the generalization performance of different network models it is necessary to use a proper test set. Since the networks were trained using a very small subset of signals on the main (channel 11) and the side channels, it would be appropriate to test the networks by using the remaining samples that were not part of the training set. So a test set containing 96,000 samples from all 48 signal events (24 signal events on channels 10 and 11 respectively) was constructed.

To evaluate the influence of different training sets, the generalization performance of feedforward networks trained using 5 training sets were compared. The results reported here are from the networks that produced the best results. In order to compare the performance of the networks, the outputs are classified into different categories such as "Correct +", "Correct -", "Noisy +", "Noisy -", "False +" and "False -". A response is classified as "Correct +" if the network's output matches the target pulse in terms of its location, amplitude and duration. A "Correct -" response occurs if the output of the network is low for the entire duration of the signal event in the absence of a pulse. Responses that are positive, but either the duration is wrong or turns on and off during the signal are indicated as "Noisy +". "Noisy -" responses occur when the network very briefly indicates a signal; these responses appear to be brief enough that they can be identified and ignored. A "False +" response occurs when the network comes on for a sufficient duration at places where there is no target pulse. A "False -" occurs if the network does not indicate a signal at places where there is a pulse. Some typical outputs of a multilayer feedforward network trained using 815 samples are illustrated in Figures 1 through 4. The topmost 4 graphs in each figure (solid lines) represent the actual signals that appeared on channels 9 through 12. The "dotted" lines represent the six outputs of the network and the "dashed line" represent the envelope of the target events.

Figure 1 represents the output of the network for the single pulse signal event 1 on channel 11 (SP1-Ch11). This is one of the single pulse signal events from which training samples were selected. Of 2000 samples in SP1-Ch11, only 42 are selected for training. This signal event is included in this illustration in order to show how well the network learned to detect the entire pulse only from limited training samples. In this case, the response of the network was classified as "Correct +" because both the actual output and the target envelope of the pulse are exactly the same.

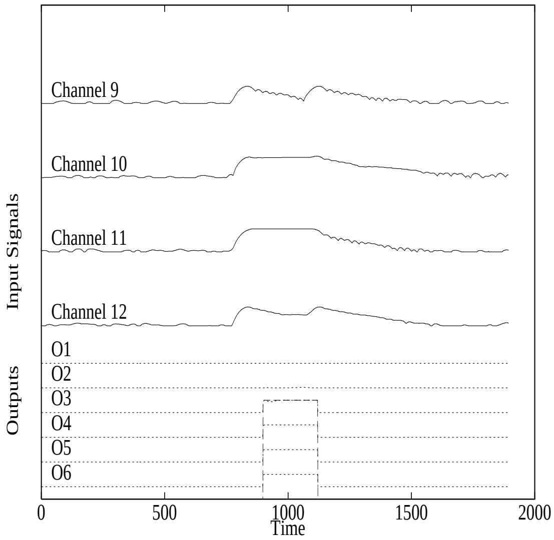

Figure 1: Generalization result of a multilayer FFN network trained with 815 samples for SP1-CH11.

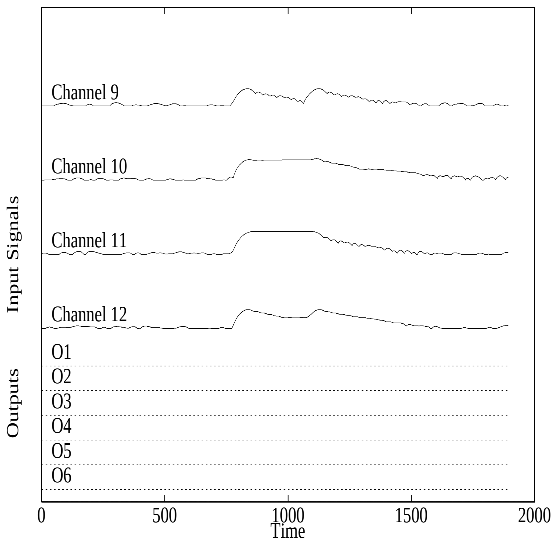

Figure 2: Generalization result of a multilayer FFN network trained with 815 samples for SP1-CH10.

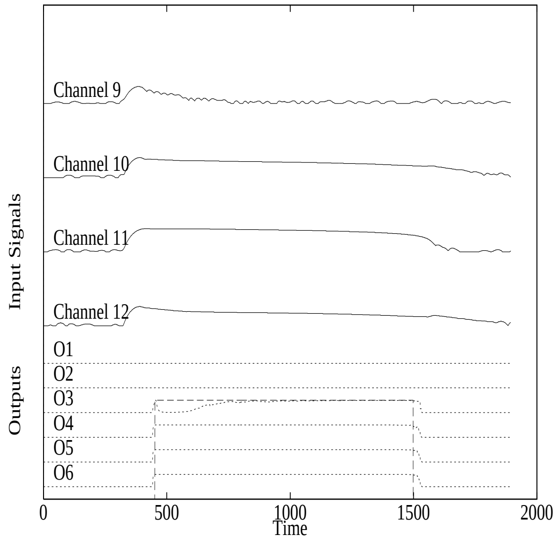

Figure 3: Generalization result of a multilayer FFN network trained with 815 samples for SP7-CH11.

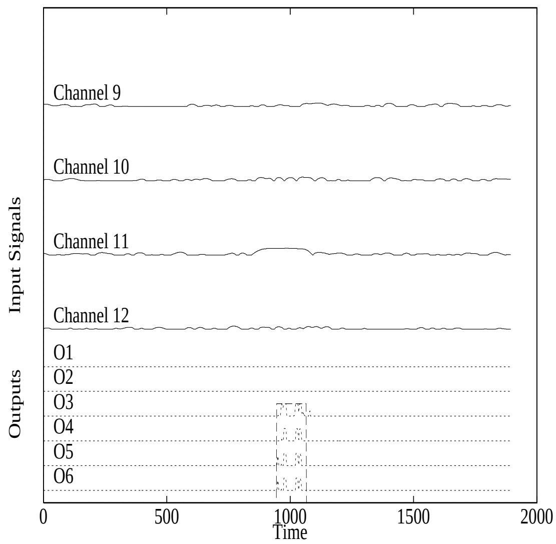

Figure 4: Generalization result of a multilayer FFN network trained with 815 samples for SP8-CH11.

The output of the network for one of the side channels (SP1-CH10) is illustrated in Figure 2. Even though the side channel signal also has similar amplitude as that of the main channel signal, the network was very successful in detecting that the signal event is a single pulse case in an adjacent channel. Figure 3 illustrates the "Correct +" response of the network for a single pulse signal event (SP7-CH11) that was not part of the training set. Figure 4 represents another test case (SP8-CH11) in which the network produced a "Noisy +" response. SP8-CH11 is one of the difficult signal events because the amplitude of the signal was not high.

Figures 1 through 4 illustrate the response of the network only for 4 of 48 test cases. A summary of the network response for all 48 signal events is presented in Table 1. In this table, the signal events that are marked ⊗ were sampled for training data. The "X" in each row denotes the response of the network for that particular signal event. Table 2 shows the effect of training size. The first column represents the signal types and the channels on which the network was tested. (For example, SP*-Ch11 indicates that *all* SP signal events on channel 11 were considered in obtaining these results.) The values in each row represents the number of times the network classified the signal events into that particular category and were obtained by combining both the + and - responses. By comparing the values across each column, the following observations can be made: i) the network response in majority of the test cases were correct, and ii) as the training set size was increased the generalization performance of the network has also increased considerably across all signal types. This trend can be seen not only across different pulses but also across the channels. Thus, these results suggest that generalization can be improved by properly selecting training samples from the regions in which the actual pulses occur.

Signal Event	Classification						Signal Event	Classification					
	Correct		Noisy		False			Correct		Noisy		False	
	+	−	+	−	+	−		+	−	+	−	+	−
SP1-CH11 ⊗	X						SP1-CH10 ⊗		X				
SP2-CH11 ⊗	X						SP2-CH10 ⊗				X		
SP3-CH11	X						SP3-CH10				X	X	
SP4-CH11				X			SP4-CH10		X				
SP5-CH11			X	X			SP5-CH10				X		
SP6-CH11 ⊗	X						SP6-CH10 ⊗				X		
SP7-CH11	X						SP7-CH10				X	X	
SP8-CH11			X				SP8-CH10		X				
SP9-CH11 ⊗	X						SP9-CH10 ⊗		X				
SP10-CH11⊗	X						SP10-CH10⊗		X				
SP11-CH11		X					SP11-CH10		X				
SP12-CH11			X				SP12-CH10					X	
SP13-CH11⊗		X					SP13-CH10⊗				X		
SP14-CH11		X					SP14-CH10		X				
MP1-CH11		X					MP1-CH10		X				
MP2-CH11		X					MP2-CH10		X				
MP3-CH11			X				MP3-CH10		X				
MP4-CH11				X	X		MP4-CH10		X				
MP5-CH11 ⊗	X						MP5-CH10⊗	X					
MP6-CH11			X				MP6-CH10				X		
MP7-CH11 ⊗	X						MP7-CH10⊗		X				
MP8-CH11		X					MP8-CH10		X				
MP9-CH11		X					MP9-CH10		X				
MP10-CH11		X					MP10-CH10		X				

Table 1: Generalization results of a multilayer feedforward network trained with 815 samples.

In order to study the generalization performance of the two hidden layer network another learning experiment was conducted using feedforward nets constructed by the modified Cascade-Correlation algorithm. A summary of comparative test results are shown Table 3. The results in Table 3 suggests that the difference in performance between the multilayer feedforward network and the 2 hidden layer feedforward network is not significant. The 2 hidden layer feedforward network has improved its performance over the multilayer network in terms of the number of "Correct" and "Noisy" classifications for Sp*-Ch11, SP*-Ch10 and MP*-Ch11. However, the 2 hidden layer network did not reduce the number of "False" classifications in all cases. On the other hand, the multilayer network's performance is slightly better than the 2 hidden layer network in all categories of MP*-CH10. Thus, these results suggest that one could use a 2 hidden layer feedforward network to get almost the same performance as that of a deep network.

Signal Event	Classification	Training Set Size				
		364	434	564	634	815
SP*-CH11	Correct	7	8	9	10	10
	Noisy	8	7	5	5	5
	False	0	1	0	1	0
SP*-CH10	Correct	5	5	6	7	7
	Noisy	8	7	6	6	6
	False	5	5	5	5	3
MP*-CH11	Correct	6	7	7	7	7
	Noisy	5	4	2	3	3
	False	1	1	2	1	1
MP*-CH10	Correct	5	7	7	7	9
	Noisy	5	3	3	2	1
	False	1	1	1	1	0

Table 2: Generalization results of multilayer feedforward networks trained with different training sets.

Signal Event	Classification	Network Model Used		
		Multilayer FFN	FFN with 2 hid. layer	JN with 2 hid. layer
SP*-CH11	Correct	10	11	13
	Noisy	5	2	1
	False	0	1	1
SP*-CH10	Correct	7	11	12
	Noisy	6	1	1
	False	3	3	1
MP*-CH11	Correct	7	7	9
	Noisy	3	2	1
	False	1	1	0
MP*-CH10	Correct	9	8	9
	Noisy	1	2	1
	False	0	1	0

Table 3: Generalization results of different network models trained with 815 samples.

To evaluate how well the modified Jordan network performs, another experiment was conducted using modified Jordan nets constructed by the modified Cascade-Correlation algorithm. The modified Jordan network also had the almost same number of hidden units as that of the feedforward networks with 2 hidden layers. The performance of the two hidden layer Jordan network is shown in the last column of Table 3. It is clear that the performance of the two hidden layer Jordan network is better than both the multilayer feedforward network and the two hidden layer feedforward network in SP*-CH10, SP*-CH11 and MP*-CH11 and as good as that of the best feedforward network (i.e., the multilayer feedforward network) in MP*-CH10. The number of "Correct" classifications of the modified Jordan network has considerably increased in single pulse cases (on both channel 10 and 11) as well as in multiple pulse cases on channel 11. Also, the number of "False" and "Noisy" classifications has been considerably reduced in all signal events. Thus, these results suggest that the modified Jordan network model is better than the feedforward networks in this signal detection application. This improvement in performance of the modified Jordan network may be due to the fact that the network is able to maintain the output signal until it detects the trailing edge of the pulse using the previous state information. The use of the previous state information is helpful not only in identifying correct duration of the pulse but also in suppressing noisy inputs.

4 Summary

This paper demonstrated the applicability of training data selection methods for improving generalization in connectionist networks using a problem that is closely related to a real world application. The results presented here show that the performance of the connectionist networks can be improved by increasing the training set size with samples from regions in which useful information is available, Thus selecting training data according to their importance should be preferred over a random sample. However, this data selection method can be applied only to domains in which the problem space is known and where the data is readily available.

It is also demonstrated how different network models can be used to improve generalization in connectionist networks. The results from the modified Jordan style recurrent network suggested that a simple feedback from the output layer can be valuable in developing a limited contextual information for the network. Furthermore, this style of recurrent network is simple to implement and allows the network to process inputs that are temporally discontinuous.

The applicability of the modified Cascade-Correlation algorithm is also demonstrated. One main advantage with the modified Cascade-Correlation algorithm is that it can produce a two hidden layer network which is easy to implement in hardware. Furthermore, the 2 hidden layer networks have simpler layered connections than the multilayer networks developed by the standard Cascade-Correlation algorithm. One drawback of the modified algorithm is that the number of hidden units in a 2 hidden layer network may be larger than an equivalent multilayer network. However, it is quite straight forward to identify the redundant units in such networks and prune them without too much additional effort.

References

Elman, J. L. (1990). "Finding Structure in Time", *Cognitive Science*, no. 14, pp. 179-211, 1990.

Fahlman, S. E., and Lebiere, C. (1990). "The Cascaded-Correlation Learning Architecture". School of Computer Science, Carnegie Mellone University, Tech. Rep. CMU-CS-90-100, Feb. 1990.

Jordan, M. I. (1986). "Attractor Dynamics and Parallelism in a Connectionist Sequential Machine", *Proc. 8th Annual Conf. of the Cognitive Science*, pp. 531-546, 1986.

Karunanithi, N. (1992). "Generalization in the Cascade-Correlation Architecture: Some Experiments and Applications". *Ph.D. Dissertation*, Computer Science Dept., Colorado State Unive rsity, Fort Collins, CO, Fall 1992.

Rumelhart, D., Hinton, G., and Williams, R. (1986). "Learning Internal Representations by Error Propagation", **Parallel Distributed Processing, Vol. I**, MIT Press, pp. 318-362, 1986.

Whitley, D. (1992). "Solving Signal Detection and Sonar Problems Using Neural Networks". Final Report, *The Colorado Institute of Artificial Intelligence*, 1992.

Whitley, D., and Karunanithi, N. (1991). "Generalization in Feedforward Neural Networks". *Proc. Int. Joint Conf. on Neural Networks, Seattle, WA.*, Vol. II, 77-82, July 1991.

Whitley, D., Starkweather, T., and Bogart, C. (1990). "Genetic Algorithms and Neural Networks: Optimizing Connections and Connectivity", *Parallel Computing*, vol. 14, pp. 347-361, 1990.