Report no. changed (Mar 2006): SM-261-UU

SACLANTCEN MEMORANDUM
serial no.: SM-261

SACLANT UNDERSEA
RESEARCH CENTRE

MEMORANDUM

Application of real data to sonar detection by neural networks

P. Nielsen

November 1992

Page count for SM-261
(excluding covers)

Pages	Total
i–vi	6
1–14	14
	20

SACLANT Undersea Research Centre
Viale San Bartolomeo 400
19138 San Bartolomeo (SP), Italy

tel: 0187 540 111
fax: 0187 524 600
telex: 271148 SACENT I

NORTH ATLANTIC TREATY ORGANIZATION

Report no. changed (Mar 2006): SM-261-UU

Application of real data to sonar detection by neural networks

P. Nielsen

The content of this document pertains to work performed under Project 20 of the SACLANTCEN Programme of Work. The document has been approved for release by The Director, SACLANTCEN.

Gordon Murdoch.

Issued by:
Systems Research Division

R. Weatherburn
Division Chief

intentionally blank page

Application of real data to sonar detection by neural networks

P. Nielsen

Executive Summary: Artificial neural networks are being studied for a variety of applications in ASW relating to the detection of target echoes by means of pattern recognition. This memorandum is a status report on an attempt to detect low-frequency submarine echoes in a set of real received time-domain beam outputs.

The reason for studying the application of a neural network to the problem of detection stems from the fact that neural networks use cumulative *a priori* knowledge to perform a specific task rather than (as in the case of a simple peak detector) taking each signal as an isolated event.

Before applying a neural network to real data, studies using simulated data were conducted on the effect that parameters such as the echo signal-to-noise ratio (SNR) have on network performance. Results indicated that the best network performance is achieved when the SNR is the same for both the training and the test set, almost independent of the value of the SNR. The network was then trained on a set of 16 real target echoes that were strong, and also tested against a set of 50 possible echoes. The result was a probability of detection (P_{det}) of 0.84 and a probability of false alarm ($P_{\mathrm{f.a.}}$) of 0.14.

It would be valuable to investigate how the performance of a network against real data could be improved if a larger training set were used. It would also be instructive to compare the performance of a trained network against a simple peak detector under carefully controlled conditions.

intentionally blank page

Application of real data to sonar detection by neural networks

P. Nielsen

Abstract: This memorandum describes the results of applying an artificial neural network for the detection of target echoes. The present study is a continuation of work conducted in 1989 in which a three-layer artificial neural network was used to detect target echoes in simulated sonar data. The use of real data here has made it necessary to investigate the influence of various network-dependent parameters, such as learning rate and other parameters that describe noise conditions. In an attempt to speed up the learning process and to improve performance, a network architecture that contains direct connections from input to output is investigated. This modified architecture allows the number of hidden nodes to be reduced while still maintaining the same network performance. The real data include 66 received signals, 16 of which contain clear target detections and are used for training. The remaining 50 form a test set. The detection performance of the network for the test set is characterized by a probability of detection $P_{\mathrm{det}} = 0.84$ and a probability of false alarm $P_{\mathrm{f.a.}} = 0.14$.

Keywords: artificial intelligence ○ target detection

Contents

1

Introduction

This memorandum describes the use of an artificial neural network for the detection of target echos in real and simulated active sonar signals. The present study is a continuation of previous work [1] performed with simulated data only.

The results presented in this memorandum are divided into three parts. The first two parts involve the use of simulated data to explore the effect of various network and data parameters on detection performance. The first part describes the effect of changing signal-to-noise ratio (SNR) and learning rate. The second part considers a modified network architecture that contains direct connections from input to output. The purpose of the modification is to realize a more efficient network: one with a shorter training phase and better overall performance. Results from these two parts are then used in the third part to train and test a network using real data.

2

Influence of network parameters

In this Section we consider the influence of network parameters on overall detection performance. The analysis is based on simulated data, as were the studies reported in [1]. Of particular interest is the effect of different SNR conditions in the training and testing phases. This is done because low SNR values were expected to be encountered in the real data. In particular, a study was conducted to find out just how low the SNR of a training set can be while still remaining adequate to train a network. In addition, work was conducted to determine the influence of the learning rate ϵ on network performance [1]. This parameter adjusts the weights of the back-propagation algorithm during the learning phase.

2.1. NOISE CONDITIONS

Table 1 contains the results for three three-layer networks trained under different SNR conditions and tested with data containing strong target echoes and with data containing noise only. The test signals have different SNR values as well as different limits between which the noise varies. All networks have 300 input units, 30 hidden units and one output unit, and all are trained until 100% performance is achieved. The test signals are time segments of the simulated output for an acoustic beam. Each set tested contains 50 separate signals. The signal amplitude varies between 0 and 5 V. Bracketed numbers following the word 'Noise' indicate the amplitude range of the noise portion of the signal (given in volts); unbracketed numbers indicate that the noise amplitude is constant at that value. The parameter $\overline{P}$ is the average of the output values from the network for a given set of signals.

The results of the tests in Table 1 show, as expected, that the best network performance is achieved when testing under the same SNR conditions as occur during training. Another measure of performance is the difference between the probability of detection P_{det} and the probability of false alarm $P_{\mathrm{f.a.}}$ (see [1]). To maximize this difference, the difference between the maximum output values for testing with noise and signal has to be large. It is also important for the standard deviation to be low, because a high standard deviation will result in a high $P_{\mathrm{f.a.}}$ and a low P_{det}.

Table 1 also shows that it is possible to achieve reasonable performance with test signals that contain noise conditions different from those used to train the network. However, the standard deviation often increases causing a degradation in the performance. These effects may be seen, for example, when a network trained with an SNR of 4 and noise amplitudes of 2–4 V is tested with test signals that contain

Table 1 *Performance when trained and tested with various noise conditions*[1]

Test signal conditions	Network training conditions					
	SNR: 2 (6 dB), Noise: [2,4]		SNR: 4 (12 dB), Noise: [2,4]		SNR: 6 (15.9 dB), Noise: [2,4]	
	$\overline{P}$	σ_{p}	$\overline{P}$	σ_{p}	$\overline{P}$	σ_{p}
SNR: 2 Noise: [2,4]	0.9699	0.011	0.8007	0.2525	0.5329	0.2160
SNR: 2 Noise: 2	0.8548	0.0797	0.3537	0.3071	0.1994	0.1825
SNR: 4 Noise: [2,4]	0.9769	0.0034	0.9673	0.0286	0.8487	0.1354
SNR: 4 Noise: 2	0.9757	0.0039	0.9570	0.0468	0.8053	0.1380
SNR: 6 Noise: [2,4]	0.9805	0.0013	0.9915	0.0017	0.9735	0.0170
SNR: 6 Noise: 2	0.9786	0.003	0.9899	0.0017	0.9521	0.0197
Noise-only						
Noise: [2,4]	0.8941	0.1078	0.3914	0.2677	0.1615	0.1062
Noise: 2	0.6932	0.1941	0.0929	0.0213	0.0494	0.0142
Noise: 4	0.9719	0.0030	0.6154	0.0940	0.2970	0.0530
Noise: [1,5]	0.7519	0.3133	0.3565	0.2951	0.1681	0.1403

[1] Noise is described by a SNR and noise amplitude that is given in volts (V).

noise amplitudes fixed at 2 V. The problem is very clear when the noise amplitude interval of the test set exceeds the interval for the training set. Thus, it is advisable that the range of noise amplitude variation in the test set should not lie outside the interval for the noise in the training set.

Variation in performance is also investigated when the testing set has an SNR other than the one used during training. The first idea is to use a training set with a low SNR and then test the network on signals with a larger SNR. This does not work well because if the smallest possible echo value is near the highest value of the noise used in the training phase it becomes very difficult to make a correct detection. An example of this can be seen in Table 1 for the network trained with data containing an SNR of 2 and noise amplitudes of 2–4 V. Table 1 also shows the opposite case when very good performance is achieved using a network trained with an SNR that is higher than the SNR for the test signal.

As another example, compare the test set with an SNR of 4 and noise amplitudes of 2–4 V to the three noise-only cases tested using networks trained with the same

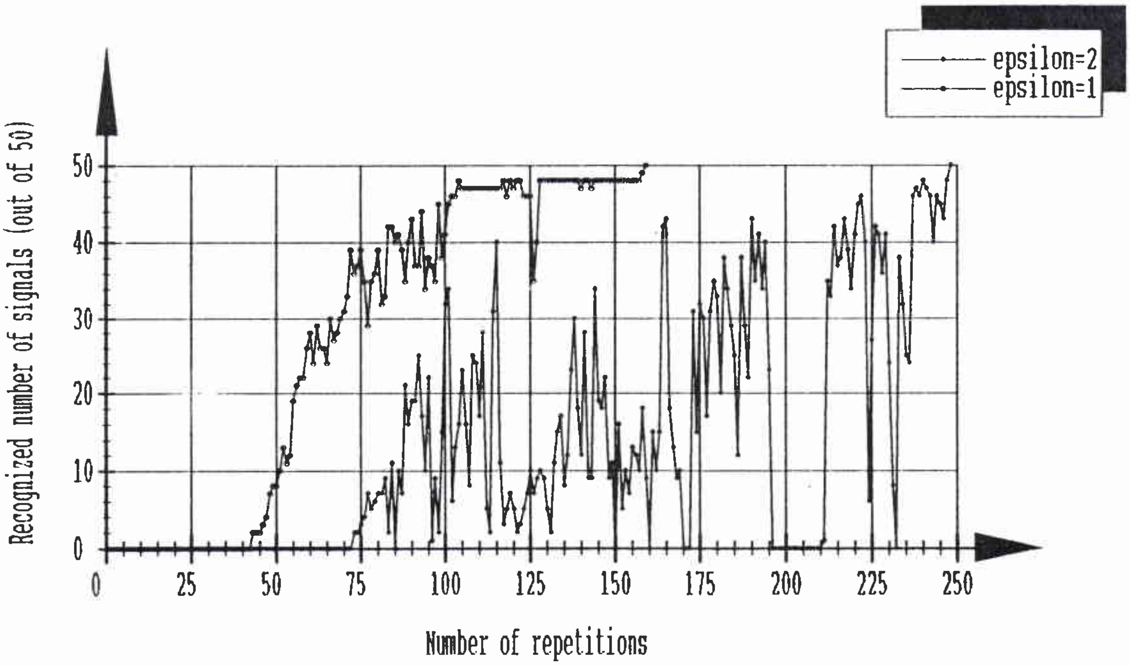

Figure 1 *Example of training networks with SNR of 4 and noise amplitudes between 2 and 4 V.*

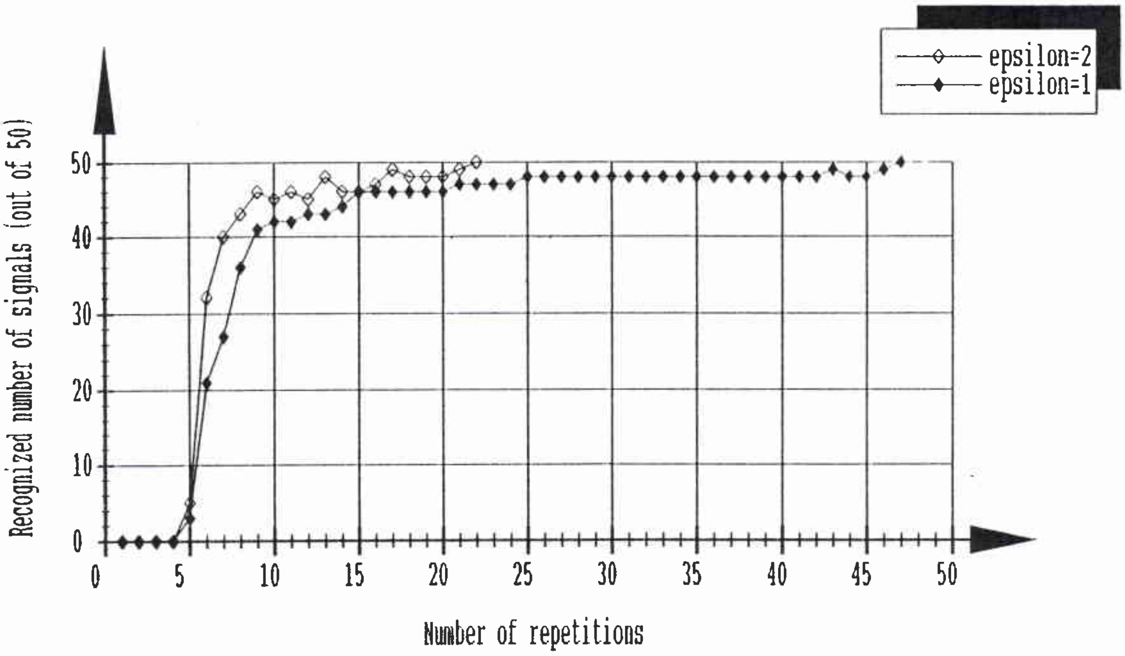

Figure 2 *Example of training networks with SNR of 8 and noise amplitudes between 4 and 5 V.*

noise interval and with SNR values of 4–6. When testing on the one trained with an SNR of 4, a threshold of 0.887 is found to be optimal and gives $P_{\text{det}} = 0.943$ and $P_{\text{f.a.}} = 0.029$. For the network trained with an SNR of 6, a threshold of 0.469 gives the very same probabilities. The fact that it is possible to train with an SNR that is higher than the SNR for the test set is of importance when using real data for training the network. This will be discussed further in Sect. 4.

2.2. INFLUENCE OF LEARNING RATE

From the studies described above, it was discovered that it is sometimes impossible to train a network to recognize all target echoes in a training set. The output values approach the maximum value of 0.5 for all signals in the training set after a few repetitions and do not change with further training. This is especially true when the training set has a low SNR. The observation of this fact results in some studies of the influence of the learning rate ϵ for the training process.

The parameter ϵ, as described in [1], is a scaling factor for the changes in the network weights, where the changes are decided from the back-propagating algorithm. Figures 1 and 2 show examples of training networks with different sets of signals and for two different values of the learning rate. Figure 1 represents a more difficult problem than Fig. 2 because its training data have a lower SNR. The figures show the number of correctly recognized signals as a function of the number of times the training set is repeated.

When training the difficult problem shown in Fig. 1, $\epsilon = 1$ worked better than $\epsilon = 2$. For $\epsilon = 1$, 100% recognition is achieved after 159 repetitions, whereas 248 repetitions are necessary for $\epsilon = 2$. The increase in fluctuations in the number of recognized signals with increasing number of repetitions suggests that the changes in the weights that result when $\epsilon = 2$ are too large for the problem shown in Fig. 1. For the simpler problem in Fig. 2, $\epsilon = 2$ is seen to result in the fastest learning.

3

Three-layer network with direct connections

In an attempt to make the training process faster and to improve performance, a different network design was studied. The new design again contained three layers and the same connections as the network discussed in Sect. 2 (and described in [1]), but direct connections were added between the input layer and the output layer as seen in Fig. 3. The values of the direct weights are decided by training a simple two-layer network such as the one shown in Fig. 4. The two-layer network is trained until a certain performance is achieved or until the training set has been repeated a specified number of times. This simple network will try to make the best possible linear discrimination between the signals in the training set. The weights for connections between the input and output were then fixed and hidden nodes were added between the two layers to take care of the remaining discrimination. The training was then continued, and this adjusted only the weights connecting the hidden nodes to the input and output nodes.

To test this architecture, networks are trained with signals containing noise amplitudes of 4–5 V and with signals containing target echoes with an SNR of 4. Two types of network are compared. One type is trained as a normal three-layer network (with 100 input units, 50 hidden units and one output unit) to 100% recognition. The other type (100 input units, one output unit) is trained first as a simple two-layer network (100 repetitions), and then the resulting weights are assigned to a three-layer network with 25 hidden nodes, and trained to 100% recognition. These weights are held fixed and the remaining weights are optimized by training the three-layer network to 100% recognition.

Table 2 *Performance for various network architectures*

Network	Target present		P_{det}	$P_{\mathrm{f.a.}}$	$P_{\mathrm{det}} - P_{\mathrm{f.a.}}$	Noise-only	
	$\overline{P}$	σ_{p}				$\overline{P}$	σ_{p}
Normal 3-layer	0.5525	0.1597	0.72	0.14	0.4043	0.1250	0.58
3-layer + direct connections	0.6049	0.1052	0.75	0.19	0.4893	0.0799	0.56
2-layer (trained to 100%)	0.5944	0.1002	0.69	0.25	0.5150	0.1004	0.44

As seen in Table 2, the performance of these networks is very similar. The training times are also nearly equal. However, the network with the fewer hidden nodes is faster in the testing phase. For the two-layer network some unexpected results

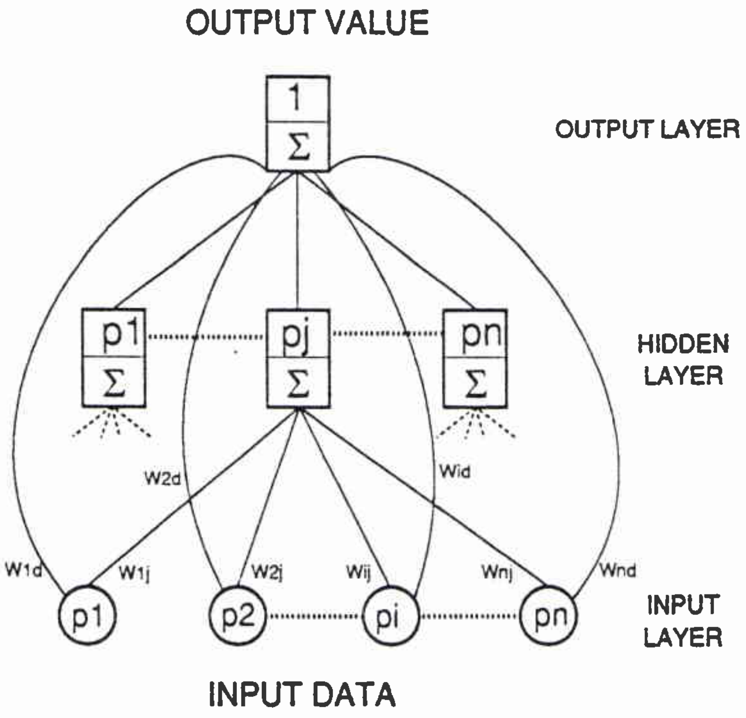

Figure 3 *Three-layer network with direct connections.*

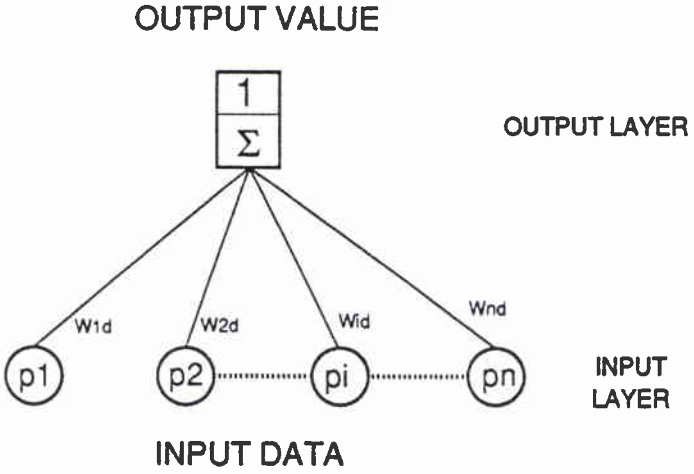

Figure 4 *Two-layer network.*

are found. It appears that the network can be trained to 100% recognition of the training set, and its performance is close to that of the two three-layer networks. Also, even though the two-layer network needs 318 repetitions of the training set, the training phase is much shorter than for the three-layer networks. The good performance of the two-layer network suggests that a simple linear detector might perform as well as a three-layer network for detection on these kinds of signals.

4

Application to real data

The real data used in these investigations were all selected from one run in a trial performed by SACLANTCEN in 1984. The run contains 66 separate signals ('pings'), each one associated with the return of a possible target echo following the transmission of a coded pulse. Of these 66, 16 are selected for training because they contain a strong target echo that is over 10 dB higher than the noise. These 16 are found using a simple peak-amplitude detector and are classified as target signals. They are used here for training normal three-layer networks. Each signal consists of 200 numerical samples. To create noise-only signals, the first 70 samples from a target signal are taken and repeated until a full 340 samples (i.e. the number of input units) of pure noise is obtained.

This study with real data is divided into three parts:

- The network is trained with simulated data (the same data as were used in the previous studies), and then tested on a portion of the real data.
- The network is trained with 8 of the 16 preselected real target signals, and then tested using the remaining 8 signals.
- Finally, networks are trained with all 16 target signals, and the remaining 50 signals are used as a test set.

All the networks have 340 input units, 30 hidden units and one output unit.

In the first part, a network is trained to 100% recognition using a training set containing 50 simulated signals with noise amplitude values that vary from 2 to 4 V and for target signals with an SNR of 12 dB. For this network, a test is made using the 16 preselected target signals. The real signal values had to be multiplied by a factor of 3 to make them fall within the amplitude range of the training data. Following this multiplication, the noise amplitude values of the test signals vary from 1.5 to 3 V.

Table 3 gives the results of the test. With a threshold of 0.2805 (found as optimal) the test results in $P_{\mathrm{det}} = 0.94$ and $P_{\mathrm{f.a.}} = 0.06$. Considering that all the test signals have an SNR better than 10 dB, this performance is not extraordinarily good.

Table 4 shows the results of the second part for training with the first 8 (1–8) preselected real target signals and testing with the last 8 (9–16). The opposite case was also tried: training with the last 8 and testing with the first 8. As before, the network has 340 input units, 30 hidden units and one output unit. Training is also

Table 3 *Test with real data on network trained with simulated data*

Noise-only		Target present	
$\overline{P}$	σ_p	$\overline{P}$	σ_p
0.1152	0.070	0.6587	0.2023

Table 4 *Training with eight out of sixteen preselected test pings and testing with remaining eight*

Training set	Test set	Reps	Noise-only		Target present	
			$\overline{P}$	σ_p	$\overline{P}$	σ_p
Ping 1–8	Ping 9–16	20	0.1556	0.0637	0.6448	0.2234
Ping 9–16	Ping 1–8	63	0.1581	0.0637	0.8837	0.0710

Table 5 *Training with all 16 preselected pings and testing with the remaining 50, the original training set and a 'noise-only' set*

	Noise-only		Target present	
	$\overline{P}$	σ_p	$\overline{P}$	σ_p
Test with 50 remaining signals	0.2648	0.2043	0.7262	0.2450
Test with training set	0.2020	0.1364	0.9379	0.1291

made on networks with more hidden units (50 and 70), but there is no improvement in performance. With thresholds of 0.2971 and 0.5019 for the first and second tests, respectively, the network gives $P_{\text{det}} = 0.875$, $P_{\text{f.a.}} = 0$ and $P_{\text{det}} = 1$, $P_{\text{f.a.}} = 0$, respectively.

The performance of the network is better when trained with pings 9–16 than with pings 1–8. Also, for training to 100% recognition, the former needs 63 repetitions and the latter only 20. This indicates that pings 1–8 are very alike, but different from some of pings 9–16. Pings 9–16 are not that alike, and thus more repetitions are required. However, pings 9–16 provide more information to the network, which results in improved performance.

Next, a network of the same dimensions as the previous one is trained with all 16 preselected target signals. A test set consisting of the remaining 50 signals is then created, and the results of testing with this set are shown in Table 5 under the heading 'Target present'. Using the threshold of 0.490 on the test results for the first test gives $P_{\mathrm{det}} = 0.820$ and $P_{\mathrm{f.a.}} = 0.140$. Or in other words, the network indicates that 41 out of the 50 contain a target. Table 5 also shows the results of using the training set itself as a testing set.

A 'Noise-only' set in Table 5 is created from the first 70 samples in the target signal. Here the network indicates a target in 7 out of 50 signals. From Table 5, it is seen that the standard deviation of the output values is quite large. This is probably caused by the fact that the noise conditions are changing from ping to ping. A scaling of both training and test signals to a certain value of the noise would probably improve the network performance. Because of insufficient time, no tests were made of this. However, with the computer programs made during these studies it would be easy to perform a more detailed study.

5

Discussion and suggestions for further work

In Sect. 2 the influence of various parameters for learning speed and performance was investigated. Other attempts to improve the network are described in [2]. These include ways of approximating the activation function that can decrease the number of calculations and speed up the learning phase. This is not important as long as the training sets do not contain more than the number of signals used here. However, for a large training set, a reduction in training time would be important.

The networks with direct connections described in Sect. 3 were not studied in sufficient detail to apply to real data. However, it seems possible to reduce the number of hidden nodes when using direct connections and still obtain the same performance, although the use of fixed direct connections slows down the training. A modification which might improve this is to use a different value for β in the sigmoidal functions for the direct connections [1] than that used in the sigmoidal functions for the hidden layer. The output would then be calculated as the sum of two activities instead of summing all contributions into one activity level and then applying the sigmoidal function with $\beta = 1$. From the real data studies it must be concluded that it is difficult in general to use simulated data for training a network intended for use on real data. The results obtained for networks trained with real data were more favourable, yielding good performance with a small training set.

Two suggestions for further work are made here. The first is to investigate whether the performance of a network against real data can be improved if a larger training set is used. Second, to compare the performance of a simple peak detector against a trained network. This comparison could be made for SNR thresholds lower than 10 dB, the value used in the present study.

References

[1] Meek, H. Target detection using a three-layered neural network trained by supervised back-propagation, SACLANTCEN SM-235. La Spezia, Italy, SACLANT Undersea Research Centre, 1990.

[2] Jacobson, J.Z. et al. Artificial Neural Network Applications in Naval Warfare, The Bureau of Management Consulting, Project 2-6803, March 1988.

Appendix A

Important programs

READ_TRAIN.FOR	Creates a training set file to use for the program TRAIN. It reads 342 samples from the ping files with the form name# (where # = ping number) and converts to a form which can be used by train. Every second signal is a noise only signal which is created from the first 70 samples.
READ_TRAIN1.FOR	Like the program READ_TRAIN but reads from US3:[BOVIO.NIELSEN.INDOIN.PO#] (# specifies ping number). It looks for a file called B018.MF1 and reads 342 signal target samples and 342 noise samples.
READ_TRAIN2.FOR	As READ_TRAIN1 but the noise signals are here created as a repetition of the first 70 samples of the previous target signal.
READ_CONV.FOR	Works like TRAIN_CONV but reads only target signals into the test file converted to the format for shift.
READ_REP.FOR	Read instead from US3:[BOVIO.NIELSEN.INDOIN.PO#] (# specifies ping number). It looks for a file called B018.MFI. It is possible here to achieve a certain length of the signal.

Document Data Sheet

Report no. changed (Mar 2006): SM-261-UU

Security Classification NATO UNCLASSIFIED	*Project No.* 20

Document Serial No. SM-261	*Date of Issue* November 1992	*Total Pages* 20 pp.

Author(s)

P. Nielsen

Title

Application of real data to sonar detection by neural networks

Abstract

This memorandum describes the results of applying an artificial neural network for the detection of target echoes. The present study is a continuation of work conducted in 1989 in which a three-layer artificial neural network was used to detect target echoes in simulated sonar data. The use of real data here has made it necessary to investigate the influence of various network-dependent parameters, such as learning rate and other parameters that describe noise conditions. In an attempt to speed up the learning process and to improve performance, a network architecture that contains direct connections from input to output is investigated. This modified architecture allows the number of hidden nodes to be reduced while still maintaining the same network performance. The real data include 66 received signals, 16 of which contain clear target detections and are used for training. The remaining 50 form a test set. The detection performance of the network for the test set is characterized by a probability of detection $P_{\mathrm{det}} = 0.84$ and a probability of false alarm $P_{\mathrm{f.a.}} = 0.14$.

Keywords

artificial intelligence, target detection

Issuing Organization

North Atlantic Treaty Organization
SACLANT Undersea Research Centre
Viale San Bartolomeo 400, 19138 La Spezia, Italy

[From N. America: SACLANTCEN CMR-426
(New York) APO AE 09613]

tel: 0187 540 111
fax: 0187 524 600
telex: 271148 SACENT I

Report no. changed (Mar 2006): SM-261-UU

Initial Distribution for SM-261

SCNR for SACLANTCEN	
SCNR Belgium	1
SCNR Canada	1
SCNR Denmark	1
SCNR Germany	1
SCNR Greece	1
SCNR Italy	1
SCNR Netherlands	1
SCNR Norway	1
SCNR Portugal	1
SCNR Spain	1
SCNR Turkey	1
SCNR UK	1
SCNR US	2
French Delegate	1
SECGEN Rep. SCNR	1
NAMILCOM Rep. SCNR	1

National Liaison Officers	
NLO Canada	1
NLO Denmark	1
NLO Germany	1
NLO Italy	1
NLO Netherlands	1
NLO UK	1
NLO US	4

Total external distribution	27
SACLANTCEN Library	10
Stock	23
Total number of copies	60