Experiments with Noise Reduction Neural Networks for Robust Speech Recognition

Michael Trompf

TR-92-035, May 1992

International Computer Science Institute, 1947 Center Street, Berkeley, CA 94704

SEL ALCATEL, Dept. ZFZ/SC3, Lorenzstr. 10, 7000 Stuttgart 40, Germany,
email: trompf@rcs.sel.de

Abstract

Speech recognition systems with small and medium vocabularies are used as natural human interface in a variety of real world applications. Though they work well in a laboratory environment, a significant loss in recognition performance can be observed in the presence of background noise. In order to make such a system more robust, the development of a neural network based noise reduction module is described in this paper. Based on function approximation techniques using multilayer feedforward networks (Hornik et al. 1990), this approach offers inherent nonlinear capabilities as well as easy training from pairs of corresponding noisy and noise-free signal segments. For the development of a robust nonadaptive system, information about the characteristics of the noise and speech components of the input signal and its past and future context is taken into account. Evaluation of each step is done by a word recognition task and includes experiments with changing signal parameters and sources to test the robustness of this neural network based approach.

1.0 Introduction

Various methods have been developed for the enhancement of a noisy speech signal; for a list of references see e.g. Sorensen (1991). The choice of a particular method highly depends on the application at hand. The approach investigated in this work is to consider noise reduction as a continuous mapping of the noisy input data space to a space of noise-free output data. The optimal mapping function is unknown and can be continuous or discontinuous, linear or nonlinear and variant or invariant in time depending on the input signal characteristics and the complexity of the task. Hornik et al. (1990) and Hecht-Nielsen (1989) have shown that function approximation in high-dimensional spaces can be done by a three-layer feedforward neural network theoretically within any predefined mean squared error accuracy.

The results reported from recent applications are encouraging: Tamura et al. (1988, 1989 and 1990) successfully trained a four-layer connectionist model for noise reduction on the speech signal waveform. As a result, they got improvements in listening tests as well as

from spectrogram analysis. The training was time-consuming and took three weeks on a supercomputer. Less CPU-expensive approaches operate in different signal domains after previous data rate reduction. Sorensen (1991) and Sorensen and Hartmann (1991) found a significant increase in word recognition rate from neural network based noise reduction using the noisy sequence of cepstral vectors as input signal representation, and Barbier and Chollet (1991) concluded from experiments in the same signal domain that even a speaker-insensitive noise reduction mapping might exist.

For a given problem, however, there are still open questions concerning the choice of design parameters such as the optimal network topology, the selection of a representative training data set, or the choice of the learning parameters. This paper focuses on the development of a neural noise reduction network by considering the application-related requirements for a word recognition task. In order to isolate the additive noise problem from connected phenomena like the speaker-stress related Lombard effect, all experiments are done with additive noise from different sources.

In the next section, the general approach is described and the requirements to a noise reduction system are summarized from a task-oriented point of view. Several network topologies and variants of the training method are evaluated and compared in section 3. Problems related to the robustness of neural noise reduction in changing signal environment are addressed and evaluated in section 4. Finally, the results are summarized and conclusions are drawn.

2.0 Approach

The motivation for this neural network based approach is twofold: 1. from a theoretical point of view, neural networks with one hidden layer are universal approximators and can be trained from example data, and 2. from a practical point of view there are application-related requirements which can also be met by neural networks. In this section, both points of view will be discussed.

2.1 Function Approximation and Noise Reduction

For the following considerations, we assume that all signals are processed framewise, and each signal frame can be represented by a n-dimensional vector. The type of coefficients and their number depend on the signal representation. If we have a corresponding n-dimensional noise-free version $\underline{y}$ for each n-dimensional noisy vector $\underline{x}$ in the training set, we can estimate the relation between both. However, we have only access to both versions during training (see section 2.2.2); after the training is finished, we apply the mapping function learned from the L training pairs $(\underline{x}_l, \underline{y}_l)$, $l = 1, \ldots, L$, to the test data in order to map each new noisy input vector $\underline{x}$ to $\hat{\underline{y}}$, an estimate of the noise-free version of $\underline{x}$. For a discussion about the influence of the training data on the approximation error, see Geman et al. (1992).The optimal solution f_{opt} to this problem in the mean squared error sense is the regression of $\underline{y}$ on $\underline{x}$,

$$f_{opt}: \quad f_{opt}(\underline{x}) = E[\underline{y} | \underline{x}] \quad , \qquad \text{(EQ 1)}$$

the expected value of $\underline{y}$ given $\underline{x}$. An approximation to this regression can be learned by feedforward networks from representative training examples (Hecht-Nielsen 1989) by minimizing the squared output error. To calculate the output error, pairs of successive noisy fea-

ture vectors at the input and noise-free vectors at the output (target vectors) are presented to the network. After the forward pass of each noisy input vector through the network, the squared difference between output and target is used for weight modification towards the steepest gradient descend. Using error backpropagation (EBP, Rumelhart et al. 1986), the squared error is fed back through the hidden layer(s) until all weights are adjusted. This is done repeatedly until the minimum is reached or no further improvement can be observed.

The approximation capability of the network is closely related to Kolmogorov's superposition theorem, which states that any continuous function with multiple inputs is representable by sums and superpositions of continuous functions of only one variable (Kurkova 1991). Because the mapping between vector pairs can be considered as a superposition of n mapping networks with n inputs and one single output for each coefficient, it is sufficient to look at only one of these networks (figure 1). The signal representation at the outputs of the hidden layer units is a nonlinear function of a weighted sum of the inputs plus an additive constant term (not shown in figure 1). The desired output function is finally obtained from a linear combination of the weighted outputs of the hidden units. The relation between the input vector and the output for one coefficient $f_c(\underline{x})$ in a network with h hidden layer units is then

$$f_c(\underline{x}) = \sum_{j=1}^{h} v_j \cdot \psi\left(\sum_{i=1}^{n} w_{ji} \cdot x_i + b_i \right), \qquad \text{(EQ 2)}$$

with $\underline{x}$ as input vector, w_{ji} as connection weights between input and hidden layer, the offset b_i as the additional input to the hidden units, $\psi(\,)$ as the nonlinear hidden layer activation function, and v_j as weights from the hidden units to the output. A common choice for $\psi()$ is e.g. a sigmoid-type function. One single hidden layer unit is shown in figure 2. Its contribution to the whole network is the calculation of the nonlinear activity function of the weighted sum of its inputs plus an additional offset.

Some important hints for the practical realization of an appropriate network topology can be found in Hecht-Nielsen (1989): the units of subsequent layers should be fully connected with each other, and three layers are - at least theoretically - sufficient. An upper limit for the number of hidden units h is $h \leq 2n + 1$.

2.2 Application-Related Requirements

2.2.1 Application Environment and System Properties

Some properties of the noise reduction system can be formulated by considering the characteristics of the input signal components, the complexity of the task, and the application related design goals. They are shown in table 1 and affect the network development as follows:

- **Signal complexity** determines the linearity or nonlinearity of the task to be learned. Analysis of the signal segment based squared error and experimental word recognition results with different network types were used to develop an appropriate network structure.
- **Context dependency** of the present signal segment from its past and future neighbors may require a network topology with a temporal input window for adjacent signal segments, leading to larger networks with more units and connections.

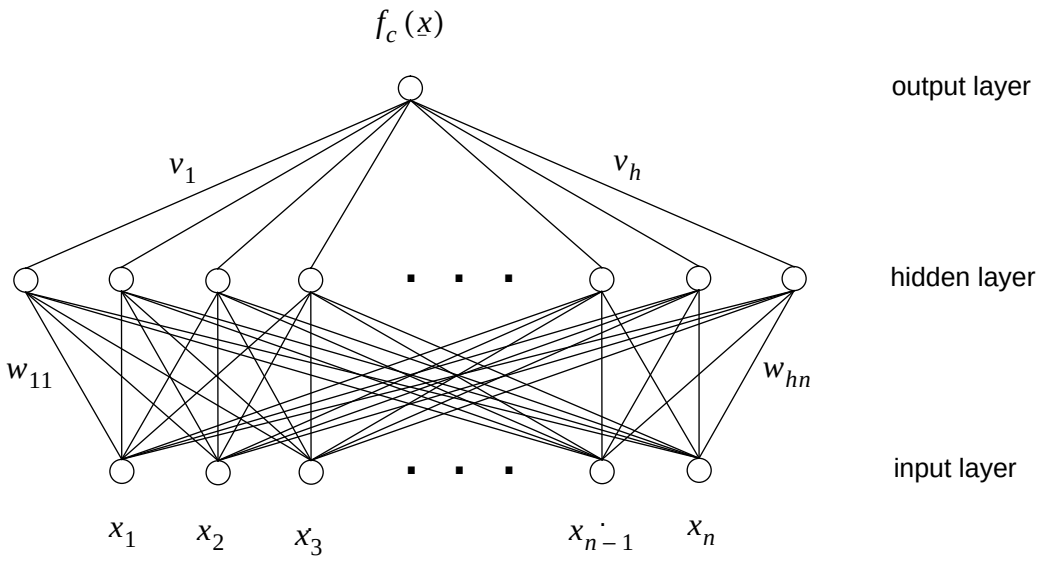

Figure 1: Multilayer feedforward network with n input, h hidden and one output unit.

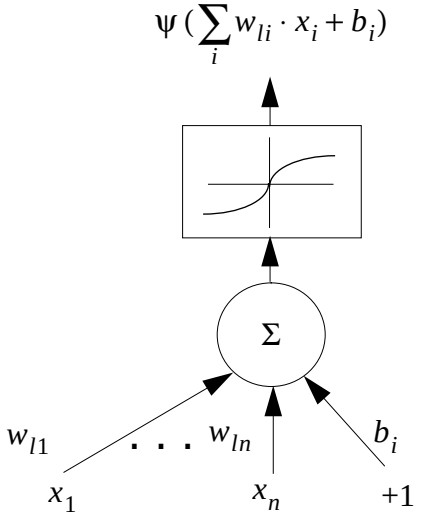

Figure 2: The l-th hidden layer unit with nonlinear activation function $\psi()$

- **Signal dynamics** and robustness of the pretrained network against parameter changes determine the need for either adaptive or nonadaptive networks. Tests with different signal-to-noise ratio (SNR) levels of the noisy speech signal and changing signal sources after training show the performance of nonadptive neural noise reduction in vaying signal environment.
- **Realtime capability** requires a moderate network size and low data rate to limit the computational power needed for the mapping. Therefore, a feature vector domain based approach was chosen which leads to a continuous mapping of relatively small amounts of already preprocessed input data.

Signal Properties	System Requirements
Signal Complexity	(Non)linearity
Context Dependency	Temporal Input Window
Signal Dynamics	(Non)adaptivity, Robustness
Application Environment	
Realtime Capability	Minimal System Complexity (Signal Domain, Data Rate)
Application Independence	Segment Based Processing, Task Independent Optimization

Table 1: System requirements derived from the signal properties and the application environment.

- **Conceptual application independence** can be reached by a frame-based mapping before signal segmentation into task dependent linguistic units. During training, the mean squared error function is used as a task independent objective function.

2.2.2 Signal Characteristics and Noise Reduction Mapping

As we want the data rate as low as possible, we chose to perform noise reduction in the domain of the domain of lpc-cepstral coefficients. After segmentation and signal preprocessing, we denote $\underline{y}_k$ the noise-free and $\underline{x}_k$ the noisy feature vector in the k-th signal segment. The optimal noise reduction mapping between the noisy and noise-free vector pairs is then given by (EQ 1).

The training task is to find an approximation $\tilde{f}$ to the optimal unknown mapping function f_{opt} to obtain an estimation $\hat{\underline{y}}_k$ of the noise-free feature vector of segment k

$$\tilde{f}: \quad \hat{\underline{y}}_k = \tilde{f}(\hat{\underline{x}}_k) \quad , \qquad \text{(EQ 3)}$$

and the approximation $\tilde{f}$ can then be learned from representative example data (see section 2.1).

If we assume that the (nonlinear) approximation to be found is continuous and differentiable, we know from series expansion techniques that it can be separated into a linear part li and a nonlinear part nl as follows:

$$\tilde{f}(\underline{x}_k) = li(\underline{x}_k) + nl(\underline{x}_k) \quad . \qquad \text{(EQ 4)}$$

As speech is only stationary in short segments and important information is contained in the dynamics of the speech signal, adjacent past and future segments also bear information about the present one. Therefore, it is expected to be advantageous to look at a wider contextual input window when restoring the present segment of the noise contaminated speech signal. In this case, (EQ 4) can be modified for a time window containing i future and j past signal segments as input. The result is the context dependent mapping function

$$\tilde{f}_{con}: \quad \underline{\hat{y}}_k = \tilde{f}_{con}(\underline{x}_{k+i},...,\underline{x}_k,...,\underline{x}_{k-j}) \quad . \qquad \text{(EQ 5)}$$

The mapping function in (EQ 5) can be separated into a linear and a nonlinear component in the same way as in (EQ 4). (EQ 5) represents an interpolation task in which a current signal segment is estimated by considering the present signal segment as well as its signal environment. The separation of the linear and the nonlinear part is independent of a particular realization; it is known from optimal filter theory that for linear problems the Wiener filter approach (e.g. Reich 1985) represents the optimal solution of the problem in the least mean squared error sense. However, there are reasons to assume that parts of the problem are of nonlinear nature. Townshend (1991) has shown that nonlinear systems work better than linear ones for the prediction of future signal samples, and this might also be true for the restoration of a present speech segment from its distorted context dependent input. Furthermore, additional nonlinearities are introduced by the signal preprocessing. Hence the complexity of the noise reduction task is unknown and should not be restricted to linear systems. In the experiments described below, the capability of neural networks to model linear as well as nonlinear problems is used to compare the performance of both for the given application.

3.0 Network Design

Based on the general considerations above, different noise reduction experiments for the isolated word recognition task are described in this section. Their goal is to answer the questions abo16ut the network topology, the training algorithm and training data selection and presentation. The test environment is shown in section 3.1. Though developed in parallel, the network topology related experiments (section 3.2) are described separately from the training algorithm related experiments (section 3.3).

3.1 Test Environment

The multi-speaker database used in all experiments contains 30 isolated German words: 20 words from an office environment and the ten digits. They were spoken by five male and five female speakers, with five noise-free repetitions for each speaker. In order to obtain a Lombard-free noisy speech signal, printer noise and computer room noise were recorded, digitized and added to the speech signal samples with different SNR's in the time domain. For comparison reasons, computer generated white noise was also added to the noise database. All signals were lowpass filtered with a cutoff frequency of 3.4 kHz, and ten LPC-cepstral coefficients were extracted every 10 ms from overlapping time segments.

After feature extraction the feature vector sequence is passed to the neural noise reduction network, noise-reduced and finally segmented into a word sequence. After time normalization to 40 feature vector long units, speaker dependent word classification is done using a previously trained neural network with "scaly" architecture. A detailed description of this net-

work as well as its word recognition results were described by Krause and Hackbarth (1988). The architecture of the test environment is depicted in figure 3.

The speech data set was divided into three partitions; the first two repetitions of the 30 words of each speaker were used as training set for both the mapping and the classification network. The third repetition was taken as verification set for the cross validation test (CV, e.g. Morgan and Bourlard 1989) during training (see section 3.3), and repetitions four and five were used as a test set for the evaluation of the noise reduction and the classification net. Therefore, all error rates shown in the result tables are obtained from 60 test words per speaker and averaged over 10 speakers. For all topology and training algorithm related experiments additive printer noise was chosen as noise signal; verification of these results was done with computer room noise and computer generated white noise. The signal waveforms and the spectra of a printer noise and a computer room noise segment are plotted in figure 4.

Two different performance measures were used to evaluate the experiments: the frame-based squared error signal and the error rate from the word classification task. The advantage of the frame based error is its independence from the classification system; however, we are also interested in evaluating the influence of modifications to the noise reduction network in terms of word error rates. For comparison of both error measures, several noise reduction experiments with different network topologies were evaluated in terms of the averaged squared frame error and the word error rates as well. Preliminary results for test data with 10 dB SNR indicate, that both performance measures are closely correlated. Therefore only the word error rates are shown in the following result tables.

The notation for the network topologies and activation functions described in this paper is as follows: “50-20-10 sig, sig” e.g. denotes a three layer network with 50 input units, 20 hidden units and 10 output units; the units in the hidden layer and in the output layer have sigmoid activation functions in this example.

3.2 Topology

The linear part $li(\underline{x}_k)$ in (EQ 4) is a first order approximation of $f(\underline{x}_k)$, and additional accuracy can be obtained from the nonlinear part $nl(\underline{x}_k)$. To compare the performance of purely linear versus nonlinear systems with and without context input for the current task, two groups of word recognition experiments with different mapping network topologies were made: 1. noise reduction of single input vectors with linear and nonlinear networks and 2. the same experiments with past and future input context. According to the signal representation, all networks have ten output and a multiple of ten input units depending on the number of context vectors. Linear networks with just one input and one output layer and linear activity function as well as nonlinear networks with one or two hidden layers, different number of hidden units and sigmoid activity functions were evaluated. After initial tests, it was found that one hidden layer with 20 hidden units and sigmoid output activation function are appropriate for the nonlinear context dependent network. Five input frames worked fine, and further increase of the number of input context frames gave only little improvement.

Training was performed by using EBP together with CV, variable learning rate and random presentation of noisy and noise-free vector pairs with or without input context. Both training and verification data sets were contaminated with additive printer noise at SNR levels of 20, 10 and 6 dB for each recording. The results in table 2 are grouped into categories to allow for easy comparison of the different experiments: the first row shows the word error rates without noise reduction, and rows 2 and 3 the results from the basic linear and nonlinear net-

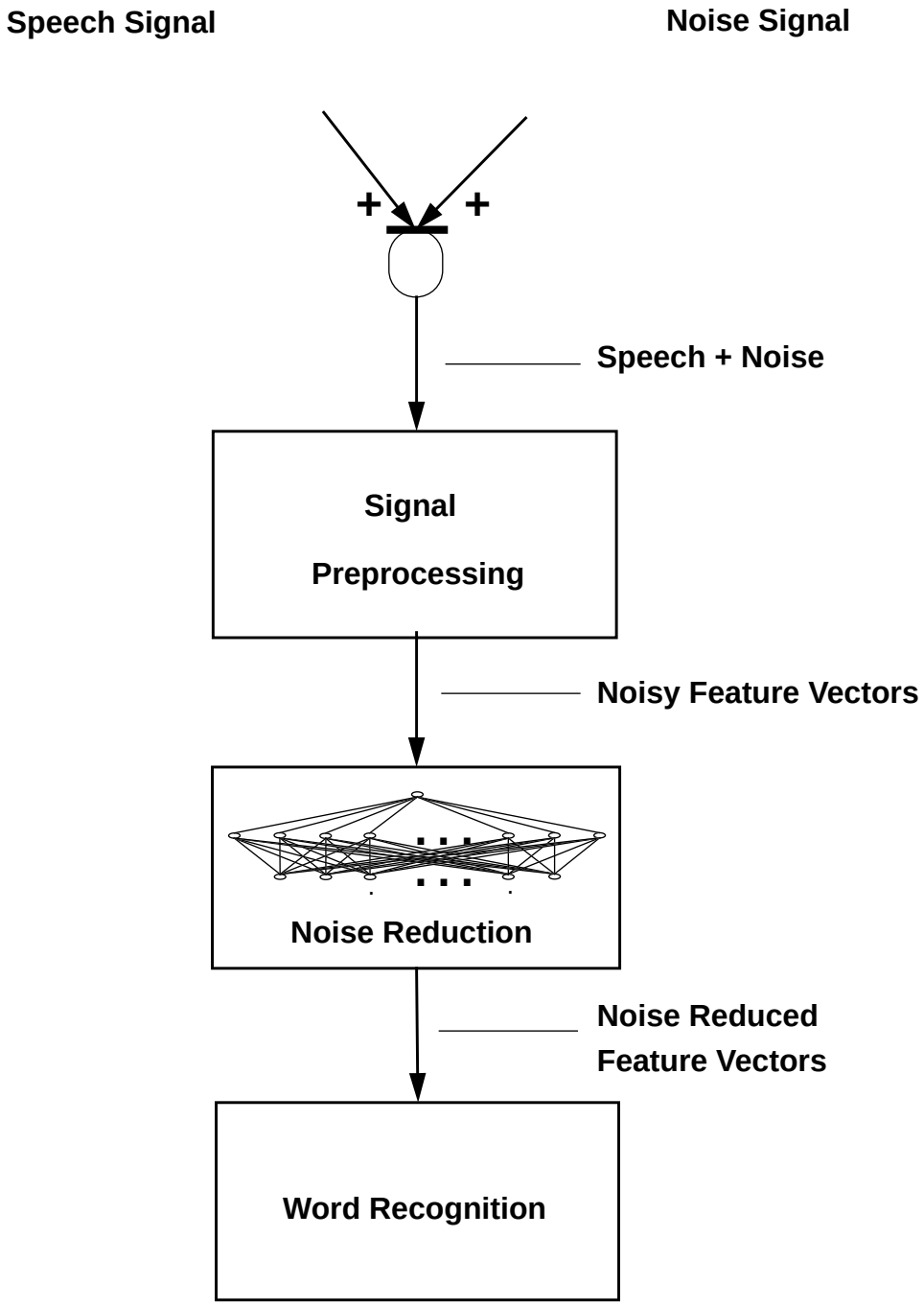

Figure 3: Signal preprocessing and test environment for the neural noise reduction experiments.

works without context input. Significant reduction of the word error rate could be obtained

network topology	# it	test data SNR [dB]			
		20	10	6	0
no noise reduction	-	3.0	12.7	28.2	58.2
linearity					
10-10 lin	7	2.0	4.3	8.8	31.5
10-10-10 sig,sig	90	1.7	4.2	7.5	28.5
context					
50-10 lin	6	1.7	3.3	6.7	25.8
50-20-10 sig,sig	71	1.6	1.7	3.5	16.5

Table 2: Comparison of different noise reduction network topologies in terms of word error rates [%]. Test conditions see text.

with both noise reduction networks, with better performance for the nonlinear network especially for low SNR. The impact of context input is shown in rows 4 and 5, with high improvement over both context-free networks. In total, the word error rate could be reduced by more than 40% in average for printer noise contaminated speech with 0 dB SNR.

However, the performance increase from the linear network without context input to the network with the highest performance is computationally expensive, because the number of training iterations (see table 2, column 2) increases with the network complexity. A rough measure for the comparison of the training times is the number of connections multiplied by the number of training iterations. Normalization of this measure to the result for the linear 10-10 network leads to an increase in training time by a factor of approximately 4 for the linear 50-10 net, and factors of 26 and 122 for the nonlinear 10-10-10 and 50-20-10 networks, respectively.

3.3 Training

The accuracy of the approximation highly depends on the selection of representative training data and the correct estimation of its parameters such as the SNR. However, every deviation from the optimal mapping function can be considered as a source of additional distortions in the feature vector domain. Possible origins are parameter misestimation as well as specialization on the training data and hence insufficient generalization ability of the network. Their impact on the word error rate is determined by the system's tolerance against parameter variations during test. One of these varying parameters is the training data SNR which was initially set to 10 dB after preliminary tests. In this section, results from different variants of the EBP algorithm for the function approximation task with the 50-20-10 sig,sig network topology are compared.

Initial experiments lead to a single frame error based weight modification after each forward pass (per sample learning) instead of the accumulated error after the presentation of the

whole training set (batch learning). The following additions to EBP training were made: 1. variable learning rate (var LR) to allow for larger weight modification steps at the beginning and smaller steps as training proceeds, and 2. CV in order to test the generalization ability after each iteration on the verification set and as stop criterion. Hence, weight modification is determined by the training set error and the adjustment of the learning rate as well as the stop criterion by the verification set error. As soon as the squared error difference after two subsequent training iterations indicates the neighborhood of a minimum, the search is continued after dividing the learning rate by two. The results for this initial configuration are shown in table 3, row 2.

Several modifications were made which affect the learning algorithm as well as the training data selection and presentation. These modifications include:

- **Random pattern selection** instead of sequential presentation of adjacent frame pairs: this technique has only minor effect on the mapping performance (see table 3, row 3), but reduces the training time by more than a factor of two because of faster convergence as can be seen from the number of iterations in column 2.
- **Multi-SNR training**: instead of applying noisy speech recordings at only one average SNR level as training data, random pattern selection allows for increasing the variance of parameters such as the SNR in the training set by randomly selecting frame pairs from differently distorted recordings. As a result, the word error rates decreased in the whole range of test data SNR's (table 3, row 4). However, this could only be reached by increasing the amount of training data by a factor of three, because the whole training set was presented with additive noise at 6, 10 and 20 dB SNR levels. At the same time the number of training iterations remained nearly the same, which led to an increase in training time roughly by a factor of three.

training algorithm	**# it**	**test data SNR [dB]**			
		20	**10**	**6**	**0**
no mapping	-	3.0	12.7	28.2	58.2
EBP+var LR+CV	192	2.0	2.5	4.7	20.5
+ rand. present.	82	2.0	2.0	4.9	20.5
+ multi SNR	71	1.6	1.7	3.5	16.5

Table 3: Comparison of different training variants in terms of word error rates [%] and number of training iterations.

Two additional modifications were investigated: *Weight averaging* (Guillerm and Cotter 1990) in order to smooth the effect of gradient descent into direction of local error decay, and a *modified sigmoid prime* (Fahlman 1988) to avoid the problem of "flat spots" during the search for a minimum in the error function by introducing an additive term (set to 0.1) in the sigmoid prime. However, the results from these experiments were in the same range as those shown in the last row of table 3.

3.4 Verification with Additional Test Data

The experiments described above were done in order to examine the performance of the noise reduction mapping and also to optimize the network topology and the training algorithm. However, the optimization should not be dependent on a particular type of noise. Therefore, the experiments were repeated with two different noise signal components. The nonlinear 50-20-10 network, which was optimized for printer noise, was taken for these experiments and no further design parameter modification was made. The two additional noise signal components used for these tests were the recording of computer room noise and the computer generated white noise already mentioned in section 3.1. As can be seen from the plots in figure 4, the main differences in the spectrum between noise signals are the harmonics, which can be noticed in the printer noise spectrum. The computer room recording consists of superposed components from multiple noise sources like hard disk drives, ventilation, air condition and others, whereas the printer noise is generated by one single source. Their main similarity is the spectral shape with a decay towards the high frequencies. White noise is different from both. It has a flat spectrum, and adjacent noise samples are not correlated with each other. The signal waveforms as well as the spectra of the printer and the computer room noise segments are shown in figure 4.

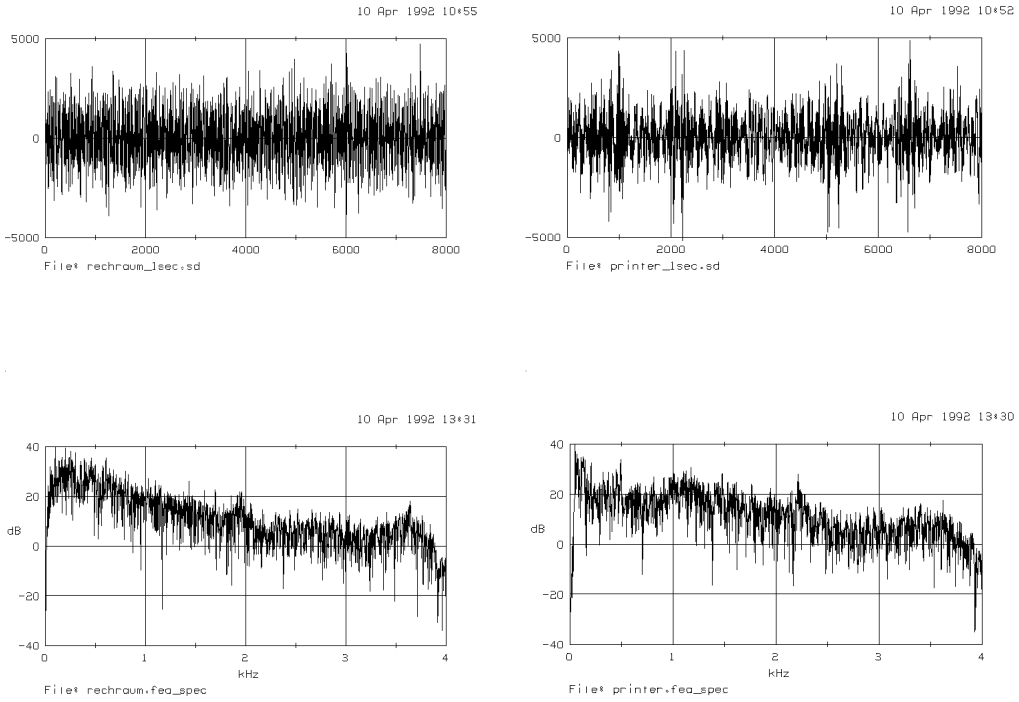

Figure 4: Time signal and spectra of computer room noise (left) and printer noise (right).

As can be seen in table 4, the noise reduction experiments were successful for the two additional test signals at different SNR levels. The word error rates after noise reduction were similar for printer noise and computer room noise. Since the original error rates for the printer noise signal were higher, the impact of noise reduction was slightly higher in this

Test Data SNR [dB]	20	10	6	0
Printer				
No Noise Reduction	3.0	12.7	28.2	58.2
With Noise Reduction	1.6	1.7	3.5	16.5
Computer Room				
No Noise Reduction	2.2	7.8	17.0	48.2
With Noise Reduction	1.7	2.7	3.5	16.1
White Noise				
No Noise Reduction	9.5	46.2	61.0	83.5
With Noise Reduction	1.8	3.5	8.2	32.1

Table 4: Word Recognition error rates from speech with additive noise from different noise sources.

case. This might be due to the optimization of the network to the printer noise signal. Sincethe goal of the experiments is to develop a robust noise reduction system, no attempt was made to further optimize the network for the new test data. For additive white noise, the initial error rate without noise reduction is significantly worse. Though the performance gain obtained from noise reduction was the highest of all three test signals (up to 52%), the final word error rates are still the worst among the results from differently distorted speech signals in the experiments. These results confirm, that noise reduction with neural networks is highly effective in stationary signal environments. This was shown for experiments with speech and different types of computer-added noise signals.

3.5 Automatic network design

So far, network development has been a time-consuming and experiment-driven process. Two development steps were necessary: at first the choice of an appropriate network topology and second, the adjustment of the weights. The future goal is the automatic design of an appropriate network structure and the training of its weights in one step. Two different classes of automatic network design algorithms are known from literature, namely algorithms with constructive and with destructive strategy. Algorithms of the first class add hidden units and layers automatically to an already existing initial network and train the appropriate weights according to a given error criterion until no further improvement is obtained. Examples for this class are Cascade Correlation (Fahlman and Lebiere 1989) and Recurrent Cascade Correlation (Fahlman 1990). For the second class of algorithms, a temporary network topology is trained initially and optimized afterwards by applying rules for deleting or merging connections in order to minimize the number of free parameters in the network and also enhance the generalization ability. Representants of this technique are Optimal Brain Damage (Le Cun and Denker 1989) and Soft Weight Sharing (Nowlan and Hinton 1991).

Because of its similarity with the experiment-driven approach described in the last few sections and its promising results in different applications, the group of constructive algorithms seems more promising for the current approach. According to comparisons with MLP's (Fahlman and Lebiere 1989), the Cascade Correlation algorithm is also expected to give improvements over MLP's in terms of computing time and network size.

4.0 Robustness of Neural Noise Reduction

The robustness of the nonadaptive noise reduction network can be defined as insensibility against changes of the input signal parameters after the training is completed. Since later adaptation is impossible for nonadaptive systems, its operation is only reliable as far as a certain parameter range is not exceeded. Outside this range the system has either to be adapted, or it must be switched off in order to avoid a decrease of performance. The experiments related to SNR changes have already been evaluated and shown in section 3. The following experiments cover changes of the noise component as well as of the speech component of the input signal.

The following two questions arise in connection with the use of nonadaptive systems for noise reduction: 1. what happens if the signal context changes after the training is completed?, and 2. can we already cover expected changes in the input signal during the training of the network? A change in a signal component between training and test time is denoted as “cross-signal” test in the following experiments (“signal’” is either related to the noise or to the speech component), and the inclusion of various signal characteristics in the training data set is denoted as “signal-pool”-experiments. All these experiments were done with the 50-20-10 network already described in section 3. An upper and a lower limit for the experimental results are given by the “signal”-dependent training mode and the “no noise reduction” results, respectively.

4.1 Noise Signal Variability

The experiments in this sections are performed in changing noise environment. Results from three different training situations are compared to the word error results without noise reduction. According to the prior knowledge we have about the expected noise environment during test, we can either include different expected signal sources into the training data (noise pool tests), or a change to a non-trained noise signal source causes complete misestimation of the noise signal properties during training (cross-noise tests. The training set for the noise-pool experiments contains a mixture of all three noise signal types. All three training data sets contain additive noise at a 20, 10 and 6 dB SNR levels, and the results shown in table 5 were averaged over five male and five female speakers. Printer noise as well as white noise distorted speech were chosen as test signals.

The results from noise dependent training are already described in section 3.4 (table 5, rows 4 and 8).In both cases, the noise-pool results give a reasonable improvement over the word error rates without mapping, see rows 3 and 6. On the other hand, considerable losses compared to the noise dependent situation have to be taken into account. Not surprisingly, the results from cross-noise reduction are worse. Whereas they are still better than without noise reduction for noise signals with similar spectral shape (row 2), completely different noise sources during test and training result in a decrease of performance (row 6). From the comparison of the noise dependent results with the noise-pool and cross-noise results it is obvious, that adaptive networks were required for further improvement of the noise reduction performance in these situations. On the other hand, noise-pool training seems to be a good compromise if the test signal environment is only partly known.

.

Test data SNR [dB]	20	10	6	0
Test data: printer noise				
no noise reduction	3.0	12.7	28.2	58.2
cross-noise reduction (computer room)	2.5	5.3	17.0	47.5
noise-pool reduction	2.2	3.3	5.0	21.7
noise dependent reduction	1.6	1.7	3.5	16.5
Test data: white noise				
no noise reduction	9.5	46.2	61.0	83.5
cross-noise reduction (computer room)	10.5	46.2	64.7	84.2
noise-pool reduction	1.5	5.3	13.5	46.0
noise dependent reduction	1.8	3.5	8.2	32.1

Table 5: Noise variability tests for printer noise and white noise. For the description of the experiments see text.

4.2 Speech Signal Variability

Similar to the noise variability experiments, noise reduction tests with changing speakers and stationary noise component were performed. These experiments help to clarify questions concerning noise reduction in speaker-dependent and speaker-pooled recognition systems. Table 6 shows the results: the speaker-pooled noise reduction mapping (row 2) was trained from all ten speakers in the data base and gives already good results in comparison

Test data SNR [dB]	20	10	6	0
Test noise: printer				
no noise reduction	3.0	12.7	28.2	58.2
one speaker pool	2.2	3.7	6.0	24.0
gender dependent speaker pool	1.7	2.6	4.0	22.6
speaker dependent	1.6	1.7	3.5	16.5

Table 6: Speech variability tests with different speakers during training and test. explanations see text.

to the tests without noise reduction. Additional knowledge about the speech signal properties, e.g. the gender of the speaker, can help to improve the mapping. As with noise dependent mapping, the best results are obtained when speech and noise signal characteristics are known in advance (row 4).

5.0 Summary

Mapping neural networks represent an efficient approach for the reduction of stationary additive noise in the feature vector domain. They are able to approximate the unknown optimal mapping function between the noisy and the noise-free signal space by learning from representative examples. Training data selection and presentation is crucial, since robustness against parameter variations can be enhanced significantly by incorporating them into the training set. Tests with different SNR's during training and test suggest that the network be trained from noisy speech signals at multiple SNR levels. At the same time, training speed can be accelerated by applying random pattern presentation.

The topology of the network is determined by the signal representation and the need of contextual input information. Though linear mapping already gives a reasonable first order approximation for low distorted speech, the nonlinear capability highly improves noise reduction performance especially in connection with context input. However, the training time increases by two orders of magnitude between a single input frame based linear and a context dependent nonlinear mapping. Network development is still an iterative heuristic process, and automatic network design would be desirable. Since the training of these systems is time consuming, acceleration by applying faster training algorithms would be helpful.

This approach is conceptually application independent, since the optimization criterion during training is the squared frame-based error; no segmentation into linguistic units is necessary and speech pause detection is only required during the supervised training.

The robustness of the approximation learned during training is of crucial importance in a changing signal environment. In order to determine the operation range, tests with changing speech and noise component characteristics were performed separately. “Cross noise” and “noise pool” experiments include either different or additional noise sources during training and recognition in order to test the system behavior in unexpected signal environment. Furthermore, expected changes can be included into the training set in advance. In general, the results from these noise robustness tests were surprisingly good. Speaker variations can be dealt with by including several speakers into the training set. Additional knowledge such as gender or the data of a particular speaker helps to further improve the mapping results. In some cases, adaptive systems are necessary to deal with a changing signal environment such as a change between completely different noise sources during training and test. The development of such networks will be a matter of future work.

6.0 Acknowledgements

This work has benefited from discussions on noise reduction techniques and neural network design by Nelson Morgan and Steve Renals from ICSI. I also wish to thank Heidi Hackbarth for her continuous encouragement and Guillaume Angleys and Harald Eckhardt from SEL ALCATEL for software encoding and providing the test data.

7.0 References

Barbier L., Chollet G. (1991) Robust Speech Parameter Extraction for Word Recognition in Noise using Neural Networks. IEEE ICASSP 1991, pp. 145-148

Fahlman S. (1988) An Emperical Study of Learning Speed in Back-Propagation Networks. CMU-CS-88-162

Fahlman S, Lebiere C (1989) The Cascade-Correlation Learning Architecture, NIPS 1989, Vol 2, pp 524-532

Fahlman S (1990) The Recurrent Cascade-Correlation Architecture, NIPS 1990, pp190 196

Geman S, Bienenstock E, Doursat R (1992) Neural Networks and the Bias/Variance Dilemma. Neural Computation 4, No. 1, pp 1-58, The MIT Press

Guillerm T., Cotter N. (1990) Neural Networks in Noisy Environment: A Simple Temporal Higher Order Learning for Feed-Forward Networks, IJCNN 1990, Vol3, pp.105-112

Hecht-Nielsen R., Neurocomputing, pp. 132, Addison-Wesley Publishing Company, 1990

Krause A., Hackbarth H. (1988) Scaly Artificial Neural Networks for Speaker-Independent Recognition of Isolated Words. IEEE ICASSP 1988.

Kurkova V. (1991) Kolmogorov's Theorem Is Relevant. Neural Computation 3, pp. 617-622, The MIT Press

Le Cun Y, Denker J, Solla S (1989) Optimal Brain Damage, NIPS 1989, Vol 2, pp 598-605

Morgan N., Bourlard H. (1989) Generalization and Parameter Estimation in Feedforward Nets: Some Experiments. International Computer Science Institute, Berkeley, TR-89-017

Nowlan S, Hinton G (1991) Simplifying Neural Networks by Soft Weight-Sharing. Computational Neuroscience Laboratory, The Salk Institute, and Department of Computer Science, Univerity of Toronto.

Reich W. (1985) Adaptive Systeme zur Reduktion von Umgebungsgeräuschen bei Sprachübertragung, Dissertation, Universität Karlsruhe, Germany

Rumelhart D. and McClelland J. and The PDP Research Group (1986) Parallel Distributed Processing, Vol 1, MIT Press

Sorensen H. (1991) A Cepstral Noise Reduction Multi-Layer Neural Network. IEEE ICASSP 1991, pp. 933-936

Tamura S., Waibel A. (1988) Noise Reduction Using Connectionist Models, IEEE ICASSP 1988, pp. 53-56

Tamura S. (1989) An Analysis of an Noise Reduction Neural Network, IEEE ICASSP 1989, pp. 2001-2004

Tamura S., Nakamura M. (1990) Improvements to the Noise Reduction Neural Network, IEEE ICASSP 1990, pp. 825-828

Townshend B. (1991) Nonlinear Prediction of Speech, IEEE ICASSP 1991, pp. 425-428