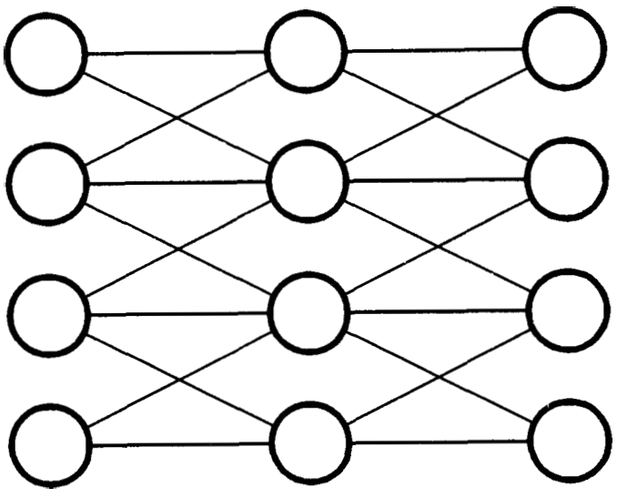
ISSN 0105-3027

A Neural Network Model for Prediction of Sound Quality.



THE ACOUSTICS LABORATORY

TECHNICAL UNIVERSITY OF DENMARK

Report No. 53,1993

A Neural Network Model for Prediction of Sound Quality.

by

Lars Bramsløw Nielsen

Oticon Research Unit ”Eriksholm”

and

The Acoustics Laboratory
Technical University of Denmark

Abstract.

An artificial neural network structure has been specified, implemented and optimized for the purpose of predicting the perceived sound quality for normal-hearing and hearing-impaired subjects. The network was implemented by means of commercially available software and optimized to predict results obtained in subjective sound quality rating experiments based on input data from an auditory model.

Various types of input data and data representations from the auditory model were used as input data for the chosen network structure, which was a three-layer perceptron. This network was trained by means of a standard backpropagation procedure and tested on selected stimuli from the subjective rating experiment. The best results were obtained with an additional input to the network, identifying the listener, and thus allowing different states for each subject.

The performance with previously unseen test was evaluated for two types of test set extracted from the complete data set. With a test set consisting of mixed stimuli, the prediction error was only slightly larger than the statistical error in the training data itself. Using a particular group of stimuli for the test set, there was a systematic prediction error on the test set. The overall concept proved functional, but further testing with data obtained from a new rating experiment is necessary to better assess the utility of this measure.

The weights in the trained neural networks were analyzed to qualitatively interpret the relation between the physical signal parameters and the subjectively perceived sound quality. No simple objective-subjective relationship was evident from this analysis.

Preface.

This report describes the specification, implementation and evaluation of a neural net model for prediction of sound quality in normal hearing and hearing impaired subjects (sensorineural hearing losses).

The neural network model is just one of the elements covered in the entire Ph.D. project "Modeling of sound quality for hearing-impaired listeners". The Ph.D. project is a joint project between Oticon A/S and The Acoustics Laboratory, Technical University of Denmark, and the report has thus been published by both parties: Oticon Internal Report No. 43-8-3 and The Acoustics Laboratory, Report no. 53. This report is meant to cover the neural network aspect of the entire project. However, there may be aspects that are not fully explained here, and there may be overlapping areas between the four reports that together comprise the Ph.D. thesis: Nielsen (1992), Nielsen (1993a), the present report and Nielsen (1993b). The last reference is the project overview and summary report.

The present report contains the following sections:

Section 1 is a brief introduction to artificial neural networks, what they can do and how they are trained and applied. The section deals primarily with supervised learning, which was used for the current project.

Section 2 takes a look at an area related to the current project - neural nets applications for speech processing and speech recognition. Parallels between the applications are made and useful ideas are discussed.

Section 3 summarizes the purpose of the present investigation and its scope within the entire Sound Quality project.

Section 4 describes the architecture of the neural net and the data reduction, from the output of the auditory model, required to make training possible. Various options for data representation are discussed.

Section 5 deals with training and test parameters for the network.

In section 6, the training sessions are listed, with different choices of training data, training parameters. The training and test statistics are presented as well as plots of predicted vs. actual ratings on the subjective scales Clearness and Sharpness.

In section 7, the weights in the trained neural networks are analyzed and the mapping from physical signal parameters to subjective sound quality is touched upon.

Section 8 discusses the results of the report and some suggestions for future work are made.

Section 9, the Conclusion, summarizes the main results

I want to acknowledge my advisors for their support and valuable discussion during the work with this neural network model: Claus Elberling, Oticon A/S, Torben Poulsen, The Acoustics Laboratory and Paul Dalsgaard, Center for Speech Technology, Aalborg University Center. Kim Vejlby Hansen and Helge B.D. Sørensen were able to help me with practical issues on neural networks, in terms that I could understand. I also want to thank colleagues at Oticon for support and constructive criticism during this phase of the project: Graham Naylor and Peter Djørup.

Lars Bramsløw Nielsen
Snekkersten, July 1993

Table of contents.

1. An introduction to neural nets. **9**
2. Speech-related applications. **15**
3. Scope and purpose of neural net application. **21**
4. Model architecture. **23**
4.1 Input data representation. 24
4.1.1. Spectral data reduction. 28
4.1.2. Temporal data reduction. 30
4.1.3. Other inputs. 31
4.2 Output data representation. 33
4.3 Neural network implementation. 34
5. Model training and testing principles. **37**
5.1 Training algorithm. 37
5.2 Training performance. 38
5.3 Test facts. 39
5.4 Test performance. 40
6. Training sessions. **43**
6.1 Training and test schedule. 43
6.2 Single subject. 45
6.3 Subject group (Normal hearing). 50
6.4 Subject group, with subject input. 53
6.4.1. Normal hearing. 54
6.4.2. Hearing impaired. 60
6.5 Test with a class of stimuli. 62
6.5.1. Normal hearing. 63
6.5.2. Hearing impaired. 65
7. Analysis of network weights. **67**
8. Discussion. **73**
9. Conclusion. **77**
10. References. **79**
11. Appendices. **83**
11.1 Calibration and stimulus levels. 83

11.2 Auditory model parameter files. 84
11.3 Example of auditory model output correlation. 86
11.4 List of stimuli and test sets. 88

1 An introduction to neural nets.

Artificial neural networks (ANN), or neural networks (NN), have become quite popular in recent years, as solutions to complicated, non-trivial problems where parametric or rule-based solutions have failed. One such example is in the field of automatic speech recognition, where ANN's have supplemented existing parametric recognition models and made them more robust (Lippmann, 1989). A combination of an ANN combined with a physiological model of the ear (Cochlear Model) as a front-end/preprocessor has successfully been used to identify spoken vowels (Cosi et al, 1990). Similarly, ANN's have been used in speech synthesis, to select the correct phonemes for correct English pronunciation of a text string (Sejnowski and Rosenberg, 1987). These are all examples, where it is difficult to formulate a model, or a set of rules, but easy to provide training examples for a neural network, while leaving the hard task - formulating the model - to the network.

An artificial neural network is composed of many simple, non-linear computational elements operating in parallel and densely interconnected. Hence the term neural nets, since they are inspired from the biological nerve cell structures in the human brain or in any animal, for that matter. Each node in an ANN can be viewed as a nerve cell, and the inter-connections (weights) are the counterparts of synapses in an organic nerve cell structure. The weights are adjustable, and are typically adapted by means of an adaptive algorithm combined with training examples presented during a training (or learning) phase. The high degree of parallelism coupled with the adaptive process allows the network to discover underlying features in the training data and extract relevant features only, while discarding random noise or irrelevant information. The network acquires the ability to generalize based on the training data and, if these represent an entire population (all possible outcomes) well, the network can predict outcomes for new, previously unseen data. As with biological systems, ANN's are not suited for precise numerical calculations (the strong point of an ordinary computer), but are able to generalize by example (what to do). A computer, on the other hand, must be programmed, e.g. it must be told exactly how to do it. The fact that neural nets are often implemented as

computer programs on ordinary computers does not affect the nature of their performance, however it is a very inefficient way to implement them, since the parallel structure is serialized and thus slowed down. Recently, dedicated hardware and chips have appeared that are truly parallel, and typically analog or mixed analog-digital. These nets provide a much larger throughput.

For the types of applications where neural nets are used for regression or classification, there are also alternatives within traditional statistical procedures. However, NN predictors or classifiers are non-parametric and make weaker assumptions concerning the shape of underlying distributions. This should make the NN approach a more robust one, when the nature of the data is not fully understood.

The neural networks perform "distributed processing". The processing of inputs is done by most of the network - thus, the representation of knowledge is distributed throughout the net. Because of this, trained networks are typically very robust, and performance is only degraded slightly if a few units (neurons) are removed. This is similar to biological cell structures, that continue to perform well if a few nerve cells have died.

There are different terms used for NN's depending on the actual application, and thus, the task performed by the NN. Identification of vowels, for instance, is a classification task, where the NN typically uses several discrete-value (or binary) outputs, one for each of the output classes (i.e. ten different American-English vowels (Cosi et al, 1990)). For other purposes, the network is not a classifier, but rather used for prediction of some output vector. This will typically require continuously-valued outputs, and the NN can be trained to perform a multidimensional, non-linear mapping (or regression) from an input space to an output space. This is an example of using the network for function approximation, which is the type of application presented in the current report.

Neural net models are specified by net topology, node characteristics and learning (or training) rules. Of many described structures, one of the most commonly used structures is the multilayer perceptron (Lippmann, 1987), which we shall focus on in the following, since it is the choice for the current project. The node characteristics include type of

input summing (typically addition), which non-linearity is used to form the output and possibly temporal integration or other types of time dependency.

Finally, there is a distinction between un-supervised training, where no desired output is presented to the network in the training phase and, on the other hand, supervised training where output patterns (exemplars) are presented during training. In the following, only supervised training is considered. Lippmann (1987) presents a broader overview and taxonomy of neural nets for the interested reader, and more recent results on supervised learning networks have been presented by Hush and Horne (1993).

In the following, we will present an important and much applied neural network structure for supervised learning, the multi-layer perceptron (MLP). The perceptron is the basic unit (node) in the multi-layer perceptron, which contains one or more layers of nodes between the input and the output nodes. This is illustrated in Figure 1. All inputs are multiplied with a corresponding input weight and summed:

$$u_i = \sum_j w_{ij} x_j \qquad \textbf{(1)}$$

This sum is passed through a non-linearity, commonly a sigmoid function:

$$f(u) = \frac{1}{1+e^{-\beta u}} \qquad \textbf{(2)}$$

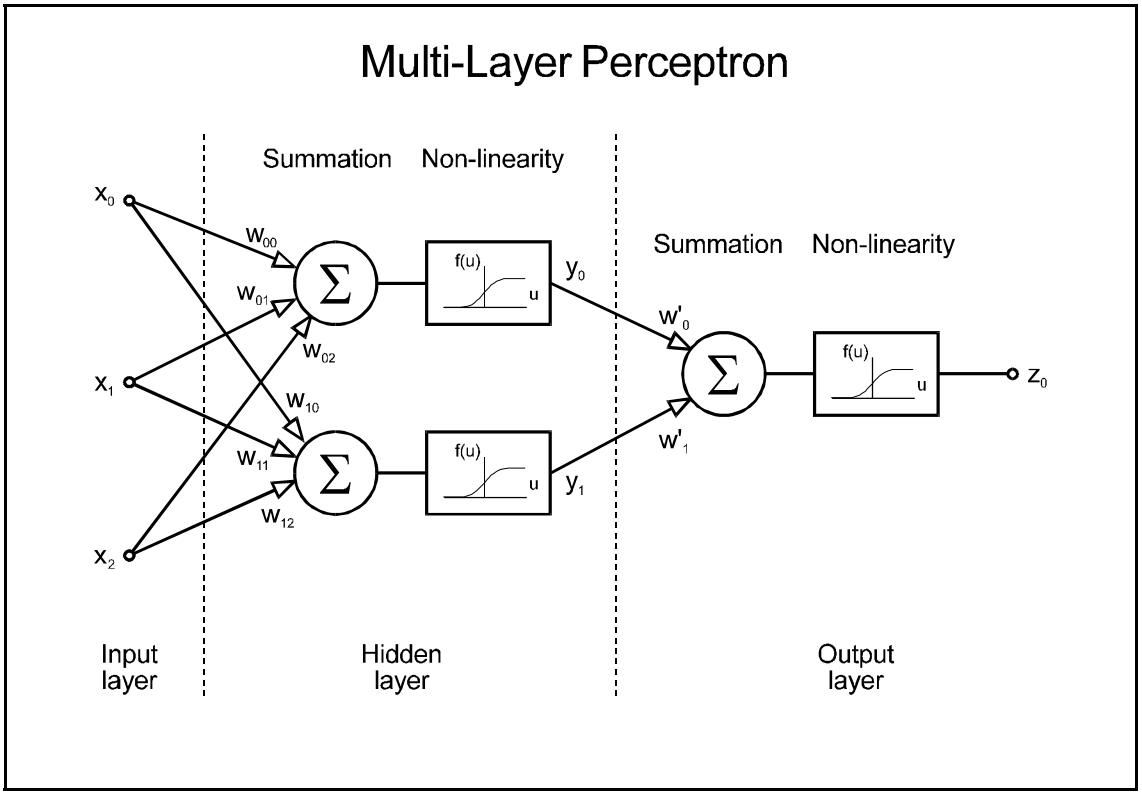
where β is the gain of the sigmoid that determines the steepness of the transition region. The neuron output from the hidden layer is then calculated as:

$$y_i = f\left(\sum_j w_{ij} x_j\right) \qquad \textbf{(3)}$$

Usually all inputs and outputs to the network have been normalized to an internal representation between 0 and 1, and the output is then rescaled to match the range of the training data.

It has been shown that a three-layer perceptron can be used to represent a decision-problem of any shape in the input-space, but the complexity of the problem is

limited by the number of nodes (Lippman, 1987). Until recently, multi-layer perceptron networks were not used, due to the lack of an efficient training algorithm. Now, the back-propagation (BP) algorithm is commonly used for training of the multilayer perceptron (McClelland & Rumelhart, 1986). This is a gradient-search algorithm that attempts to minimize the squared error on the output by adjusting the weights backwards in the net, i.e. starting with the weights in the output layer and propagating the error back through the preceding layers, while adjusting their weights. Descriptions of this are provided by Hush and Horne (1993), Hertz et al (1991) and McClelland & Rumelhart (1986).



1. *Example of a simple three-input, one-output multi-layer perceptron with two neurons in the hidden layer between input and output, and a total of eight weights ($w_{..}$ and $w'_{.}$, indicated by open arrows). The output non-linearity shown is the commonly used sigmoid function.*

One of the difficult aspects in the use of MLP is how to select the proper network size. The network must be large enough to approximate the given problem by developing an internal representation, but not so large that the weights (= degrees of freedom) cannot be estimated reliably from the available training data. A MLP-network consists of an input layer, one or more "hidden" layers (i.e. not visible from the outside) and one output

layer. The input layer is simply the input terminals of the NN, distributing signals to the first hidden layer. The hidden layer(s) performs the feature extraction, i.e. the neurons in a hidden layer, together with their input weights, are where the "intelligence" of the network is stored. Usually, one hidden layer is used, but more hidden layers have provided an advantage in some cases. Finally, there is the output layer, which uses weighted sums of the hidden layer activity to form the network output. This layer usually also employs non-linear neurons, except for certain signal-processing applications, where the output unit(s) are weighted sums without a non-linearity (Hush and Horne, 1993).

After this short introduction to neural networks, we will proceed with the description, implementation and evaluation of the present NN application: Predicting the sound quality of processed sound signals based on an auditory model front-end and training data obtained from subjective listening tests. Good textbooks on neural nets and related topics include: Hecht-Nielsen (1990), Hertz et al (1991), Kohonen (1984, 1988), McClelland & Rumelhart (1986, 1988) and Rumelhart & McClelland (1986).

1. *An introduction to neural nets.*

2 Speech-related applications.

In this section, we will focus on applications of neural networks to speech processing and related areas, since these applications might provide inspiration for the present application. There are no direct examples found in the literature on the use of ANN's in the context of sound quality measures or prediction of sound quality.

There is a crucial difference between the applications in speech synthesis and recognition (ASR) and the current project: These models convey or extract the information carried by the input (speech) signal. In the present project we are interested in some perceptual attributes of the incoming signal, and specifically not the information. Speech recognition by the hearing aid user is deliberately left out in the definition of sound quality used in the present project (Nielsen, 1992). Thus, the focus here is on extracting the quality of the signal (whether this is music, speech or other signals) as opposed to the information. Nevertheless, the speech work summarized below has provided inspiration for choice of window sizes, frame spacing, preprocessing, network topology and more. Many papers on speech recognition are not particularly scientific, but more pragmatic, and many choices are motivated by past experience, rather than strict scientific argumentation. And the present application is not different - one has to experiment and gradually increase the sophistication and performance of the systems.

One of the first large, successful applications of neural nets in the speech area is the NETtalk system, a multilayer perceptron to pronounce English text (Sejnowski & Rosenberg, 1987). The network input was a window of 7 letters, each letter represented by 26 units plus 3 units for punctuation and pauses, i.e. a total of 203 binary-valued units. There were 80 hidden units and 26 output units, representing the phonemes combinations of articulatory features, such as place and manner of articulation, phoneme type, vowel height, stress and punctuation. After training, the output neurons were then used to control a speech synthesizer, that converted the features to the acoustic signal. The network was trained using backpropagation on 1024 words from children's speech and speech was understandable after 10 passes. Adding input groups (i.e. more letters) and an extra layer of hidden units both improved performance. The layer of hidden units

in the trained network was analyzed, but no clear patterns emerged, except in a few cases, where clusters of units seemed to provide the vowel/consonant distinction. The trained network was quite insensitive to damage (i.e. robust) and re-training after damage was quick. This is a classic and quite successful application.

Cosi et al (1990) have used neural nets for a speech recognition problem: classification of spoken vowels. A physiological model of the ear (cochlear model: Seneff, 1985) was used as acoustic pre-processor, that appears to encode phonetic features in a rather straightforward way. 40-channel output vectors from the cochlear model were sampled every 5 ms, and assembled (interpolated) into 10 frames, equally-spaced in time, that contained the entire vowel. The vowel boundaries were determined by using a vowel detection algorithm directly on the speech signal. This resulted in 400 spectral coefficients for the neural network (= 400 input units). 20 hidden units in a single hidden layer (no particular reason for this number) and 10 output units (representing 10 English vowels) were used and the network was trained using back-propagation on 5 samples of each vowel from 13 speakers. The recognition score was approximately 95% for seven new speakers. By comparison, an FFT-based 40-channel mel-scale filterbank achieved a score of 87%. The authors point out that the overall advantage of the cochlear model needs to be investigated further.

Gramss & Strube (1990) did experiments with a psychoacoustic preprocessor and a neural network for recognition of spoken German digits. The preprocessor derived 8,16 or 38 Bark-scale (critical bands) filter outputs from a 512 point FFT. The sampling frequency was 10 kHz, with an overlap of 412 samples = 41.2ms, i.e. one frame was obtained for each 10 ms. The output power from each critical band was converted to loudness, using a power-law with exponents in the range p = 0.2 - 1.0, instead of just the classical exponent of 0.3. The resulting loudness spectrogram is contrasted (enhanced) within a rectangular area of the spectrogram. This is followed by extraction of spectral and temporal features (not described in detail) and the neural network. What neural net structure was used, is unclear from the paper, but a recognition rate of 100% was obtained. It is interesting to note that the score was sensitive to the loudness power

exponent, with optimal results for p = 0.5. Training was done on 11 different utterances of the 10 first digits in German, by one speaker. Subsequent testing was done on a 12th utterance by the same speaker.

A summary of auditory and neural-net models used for automatic speech recognition (ASR) was presented by Lippmann (1989). Typically, a spectrum-based (Fast Fourier Transform (FFT) or Linear Predictive Coding (LPC)) front-end delivers frames every 10 ms to the recognizer. A summary of physiologically-based auditory preprocessors is provided. They have a potential advantage over purely spectrum-based preprocessors, by providing time and synchrony information (intra- as well as inter-channel), which appears to provide more robustness under noisy conditions. Some of these physiological (cochlear) models have been reviewed by Nielsen (1993a).

Lippman (1989) identifies three other elements in ASR: pattern matching and classification (phonemes), time alignment and pattern sequence classification (words). The neural net can do a number of tasks related to ASR, such as computing local distance scores to stored reference patterns on a frame-by-frame basis, perform vector quantization and reduce the dimensionality of input patterns. They may develop internal hidden abstractions in hidden layers that can be related to meaningful acoustic-phonetic speech characteristics, such as formant transitions. Auto-associative unsupervised (e.g. Kohonen, 1984) networks can reduce dimensionality and extract relevant features, similar to principal component analysis. Lippmann presents several examples on multilayer perceptrons for static (time-aligned, pre-segmented) classification of speech segments (e.g. determine which vowel is present). One study found that the hidden nodes (units) often become feature detectors and differentiate between important subsets of sound types such as consonants versus vowels, and it was stressed that choosing the right data representation for speech is crucial. Various network sizes and time-feature extraction schemes have been used. For classification into speech feature maps, combinations of unsupervised and supervised training in hierarchical nets have been suggested and used successfully with rapid training. For dynamic classification of speech segments (e.g. running speech), time-delay neural networks (TDNN) are most common

(notably Waibel et al (1989), Waibel & Hampshire (1989)). TDNN's are multilayer perceptrons with time delays in the input and some form of temporal integration in the output nodes. The TDNN used by Waibel et al contained successive layers, with gradually fewer dimensions to provide specific feature-classification (e.g. classify voiced stops /b,d,g/). Several of these sub-nets can then be trained separately and combined with some "glue"-nets for faster overall training. This type of pre-defined architecture probably has an advantage compared to a large, completely interconnected network, and the resulting weight pattern may be easier to interpret. New training algorithms have also been developed for faster training. Nets with internal memory and recurrent connections have also been used to capture the time structure and discrete states in the speech, and neural nets have been used in combination with traditional techniques (Hidden Markov Models, Dynamic Time Warping) with success. Much progress has been made here since Lippmann (1989) published the summary paper.

Waibel (1992) has provided a more recent summary on neural network approaches for speech recognition. The speech recognition is divided into three levels: The phonemic level, the word level and the language level. Only the first of the three is of relevance to the present project. Phoneme classification networks can be divided into two groups: Temporally static classifiers, that require precise temporal alignment of input tokens, and temporally dynamic classifiers that do not require this. The static classifiers are often multi-layer perceptrons with backpropagation training. The temporally dynamic classifiers include Time Delay Neural Networks (TDNN) and recurrent networks. In recent work with TDNN, it was again possible to "discover" phonetic features in the trained network, given specialized sub-nets for different tasks. Hidden layer activations showed specific response to acoustic-phonetic features such as detectors for unvoiced speech, vowel onsets and rising or falling formants.

Sørensen (1991) used a multi-layer neural network for cepstral noise reduction. The entire system consists of an auditory (psychoacoustic) model for pre-processing, a cepstral noise reduction neural network and a classification neural net. The auditory model output is specific loudness in 30 channels, every 10 ms, which is interpolated to

256 points, fed to an inverse FFT, from which 8 LPC coefficients are derived and subsequently transformed into 8 cepstral coefficients. It is argued that the cepstral domain works better due to the deconvolution property of the cepstrum transform. The cepstral input coefficients were then fed to a four-layer perceptron network with 1 input, 2 hidden and 1 output layers using 8, 32, 32 and 8 neurons in the four layers. In this application, 2 hidden layers gave better results than one. The network was then trained at this stage by presenting speech mixed with fighter plane cockpit noise at various signal-to-noise ratios to the input and using the network outputs for the corresponding clean speech signals as training data. Thus, this network performed an eight-dimensional autoassociative mapping from a "noisy" input space back to a "clean" input space, i.e. a function approximation.

The subsequent classifier was trained to classify 1170 instances of the Danish digits 0 - 9, spoken by 66 speakers, and tested using 510 words from 34 speakers. At 0 dB S/N, there was a 65% improvement in recognition score, using the auditory model front-end and 35% using a traditional LPC-Cepstrum front end. This application is another example of a successful neural net application, and the advantage of an auditory-based front-end is again demonstrated.

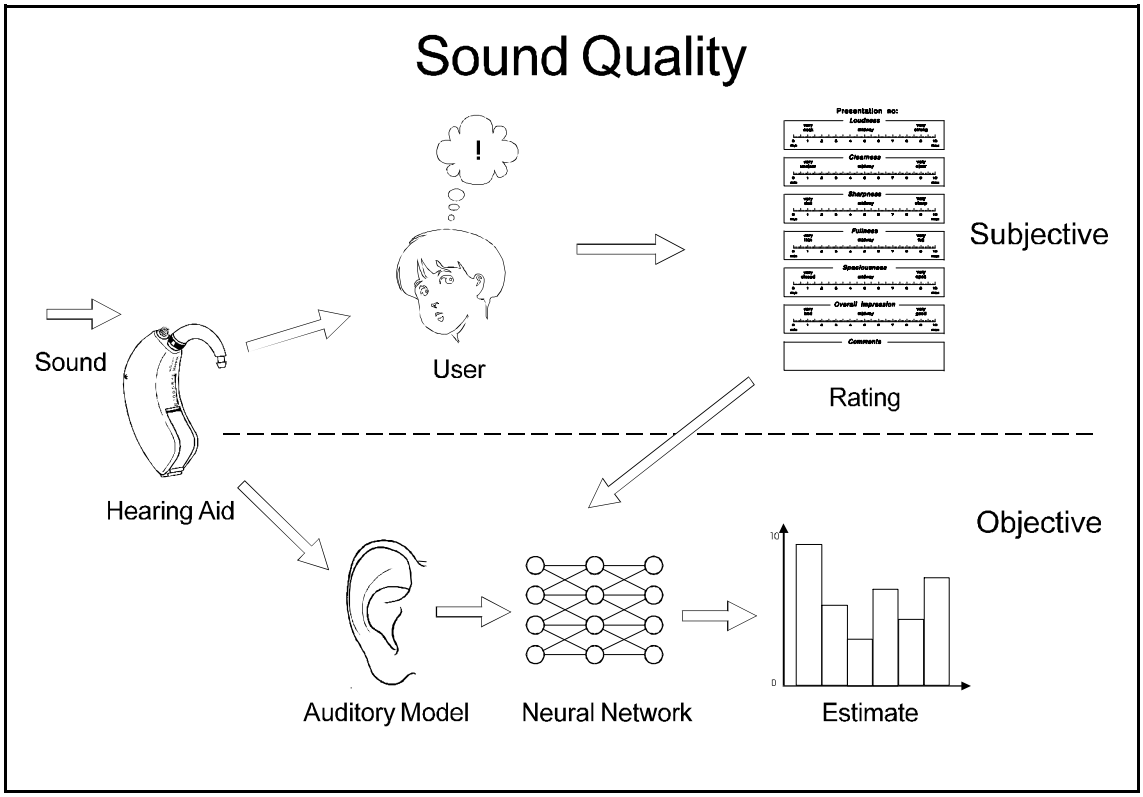
Compared to speech recognition, there is an application that lies closer to the present sound quality application, namely the task of speaker identification. This is for instance used in security systems, automated bank tellers, or elsewhere where a determination of speaker identity is required. The information carried by the signal is irrelevant, but the speaker identity must be determined on speech characteristics, such as fundamental frequency, timbre of the voice, articulation, prosody (rhythm) etc. Bennani et al (1990) presented a neural net approach to this problem. The speech was pre-processed using two different methods: LPC analysis, providing 12 LPC coefficients every 10 ms, and a critical-band type front-end based on FFT analysis followed by 24 triangular filters, from which 8 cepstral coefficients are derived (MFCC: Mel Frequency Cepstral Coefficient). After pre-processing a sentence into N frames, a 12 x N or 8 x N matrix is obtained. The principal components for the matrix are determined (see also section 4.1.1) and these are

passed onto a neural net that performs the speaker classification. The MFCC preprocessing, using the auditory model, showed ~20% higher recognition scores than the traditional LPC preprocessing. The preprocessing and data reduction used by Bennani et al (1990) is interesting for the present investigation.

All the papers presented above present actual, practical applications, proving that neural networks are useful in the speech area. It is characteristic that the decisions on preprocessing, net size, topology etc. are often made with no strong scientific basis, since no stringent theories are available, and that the applications nevertheless are successful in performing the assigned task. With this in mind, we shall proceed with the current neural network application.

3 Scope and purpose of neural net application.

The neural network modeling is the final step in a larger project, aiming for the development and evaluation of objective sound quality measures (Nielsen, 1993b). The overall idea in the project (also shown in Figure 2) is the following: A sound signal is processed through a hearing aid, where it is subject to "desired" signal processing (frequency shaping, output limiting etc.) and "undesired" signal processing (resonance peaks, non-linear distortion etc.). The subjects - one normal-hearing group and one hearing-impaired group - rate the subjective sound quality of the signals on a number of perceptual rating scales (Loudness, Clearness, Sharpness, Fullness, Spaciousness and Overall impression). The same signals are presented to a computer model of the ear - an auditory model - which mimics some of the dominant psychoacoustic properties of the normal or the impaired ear, to the extent that these are known today.



2. *Overall project concept for an objective measure of sound quality. See text for details.*

The auditory model thus implements known properties of the normal and the impaired ear, and the remaining task in predicting sound quality is left for the neural network.

In the actual rating experiment, the signals were not processed through a hearing aid, but through various types of signal operations: addition of background noise, filtering, clipping, compression etc. in various combinations. The purpose of this scheme was to generate signals that were perceptually very diverse and which elicited responses also on the extremes of the different perceptual dimensions (Nielsen, 1992).

The neural network (NN) model must tie these two elements together: subjective rating data and auditory model output. By using the auditory model output as NN input and the subjective rating data as desired NN output, the neural net can be trained to match the rating data. After training on the training set, the combined auditory model and neural network should ideally be able to predict the sound quality of any signal with which they are presented. Most likely, it will predict (that is, mimic) well on the training data, but it should also be able to predict the perceived sound quality of a stimulus, that has never been presented to the model before, i.e. it should be able to generalize. By evaluating performance on the test set, we get some indication of the model's ability to generalize beyond the training data. These results must be treated with caution, however, as the test set used in the following still originates from the same experiment, i.e. the same class of signals, the same groups of hearing losses, the same subjects etc. A true cross-validation can only be performed with results from a new experiment with different types of signals, different subjects, but the same subjective test method.

Analysis of the weights in the trained network can provide interesting information on the features that the network extracted during training. In the current investigation, this might give us some insight into what physical (or auditory) parameters have an effect on the perceived sound quality. The weights may provide a meaningful link between the objective stimulus measures and the subjective impression.

4 Model architecture.

In the present investigation, the neural network (NN) must perform a fitting of a non-linear function to a multidimensional set of input and output data, thus creating a mapping from input to output. This is also referred to as function approximation. The trained network should be able to generalize between data points (interpolation) and for new input data unknown to the network (extrapolation).

There are alternatives to neural networks, such as linear and non-linear regression methods, but these may often require some kind of "qualified guess", in order to estimate a large number of free parameters (Conradsen, 1984a). For the current approach, the neural-network method was chosen as a 'non-parametric' type of modeling, where no initial guess was required, except for the choice of network structure and training method.

Since there are no prevailing models or methods for objective evaluation of sound quality, a neural network can potentially be used to determine the complex mapping from sound signal (processed by a model featuring the known properties of the ear) to subjective sound quality. Ideally, a large pool of data representing all types of signals and listening situations should be presented, unfiltered, to the NN, which should then be able to analyze and extract the relevant features. Such an approach, where as little *a priori knowledge* as possible is used, is attractive, but has severe limitations in practice as we shall see.

Even though an auditory model was developed and used for the preprocessing of the signals, it is not clear whether this type of advanced signal processing is necessary for a good objective estimate of sound quality. One could imagine using 1/3 octave filtering, short-term FFT analysis or other signal measures with more or less perceptual relevance, assuming that the neural network would still be able to establish the underlying relation between signal and perceived sound quality. Nonetheless, the present investigation used a perceptual model of the ear as a signal preprocessor, which includes known properties of the normal and impaired ear (Nielsen, 1993). The literature presented in section 2

also indicated a potential advantage by using auditory models. There are side-benefits from the use of the present auditory model, such as the calculation of total loudness in both the normal and the impaired case, according to established theories. The use of simpler pre-processors is left as a topic for future investigations.

4.1 Input data representation.

For the subjective sound quality ratings, 64 stimuli with different types of input signal, frequency shaping and spectral and temporal distortion were created. These stimuli were of 30 sec. duration, but presented twice in succession, allowing the subject 1 minute for the rating task. See Nielsen (1992) for a complete description of the subjective rating experiment. The 64 stimuli were processed through the auditory model using one parameter file for the normal-hearing group and another one for the hearing-impaired group. The parameter file for the normal-hearing group was set with audiogram = 0 dB HL across all frequencies, and stimulus level set at the most comfortable level (MCL) averaged across the 12 normal-hearing subjects. For the hearing-impaired group, the parameter file was set with audiogram equal to the average audiogram across the 11 subjects (which were deliberately matched closely (Nielsen, 1992)) and the average stimulus level. The calibration procedure for the stimulus levels is described in Appendix 11.1. The parameter files specified a 20.16 kHz sample rate and a 256 point window size, with no overlap, corresponding to windows 12.8 ms in length. The important parameters for the two subject groups are listed in Figure 3, and the two parameter files for the auditory model preprocessor are listed in Appendix 11.2. For a detailed description of the parameter file format, see the report on the auditory model (Nielsen, 1993a).

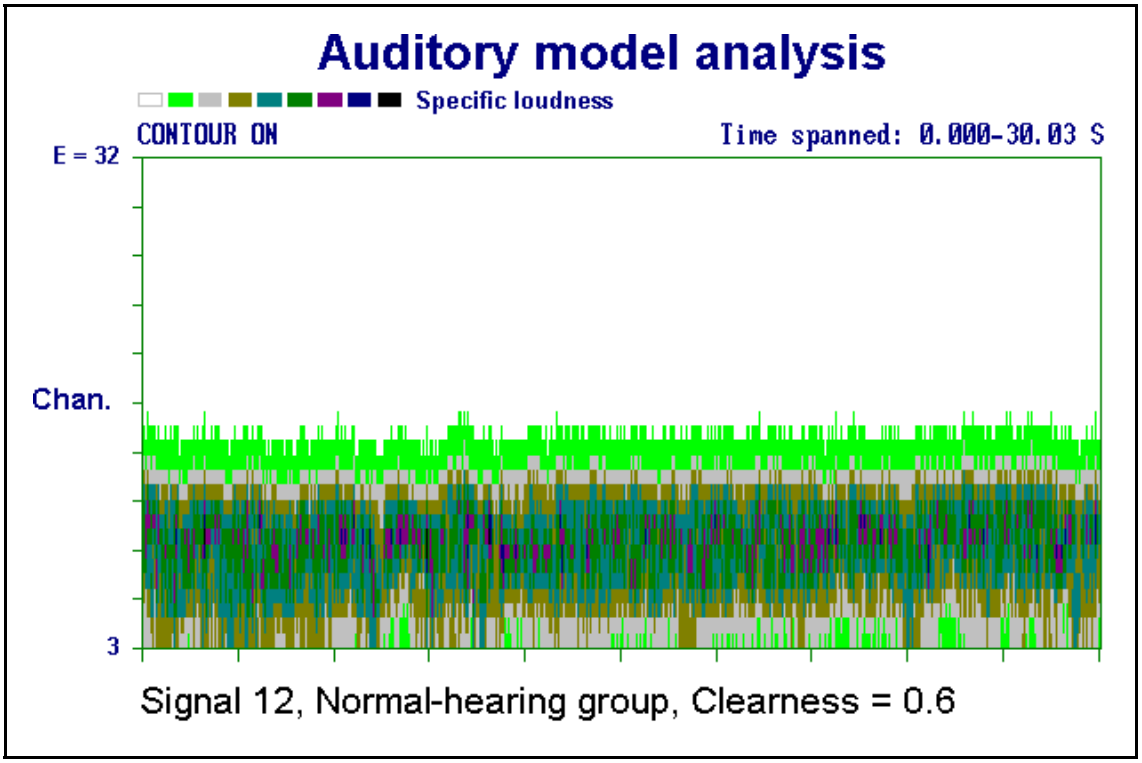
Important Auditory Model parameters.

Number of channels:	30										
Sample rate (kHz)	20										
Frame size:	256										
Overlap:	0										
Stimulus RMS value:	880										
dB SPL (NH)	79.5			dB SPL (HI):			100.2				
Frequency (Hz):	125	250	500	750	1000	1500	2000	3000	4000	6000	8000
NH Audiogram (dB HL):	0	0	0	0	0	0	0	0	0	0	0
HI Audiogram (dB HL):	30.5	35	41.4	44.1	49.1	54.1	57.7	61.4	63.7	69.6	76.4

3. *Table showing the auditory model parameters used for the processing of the 64 stimuli from the listening experiment. Example shown uses average audiogram and average stimulus level for the hearing-impaired subject group.*

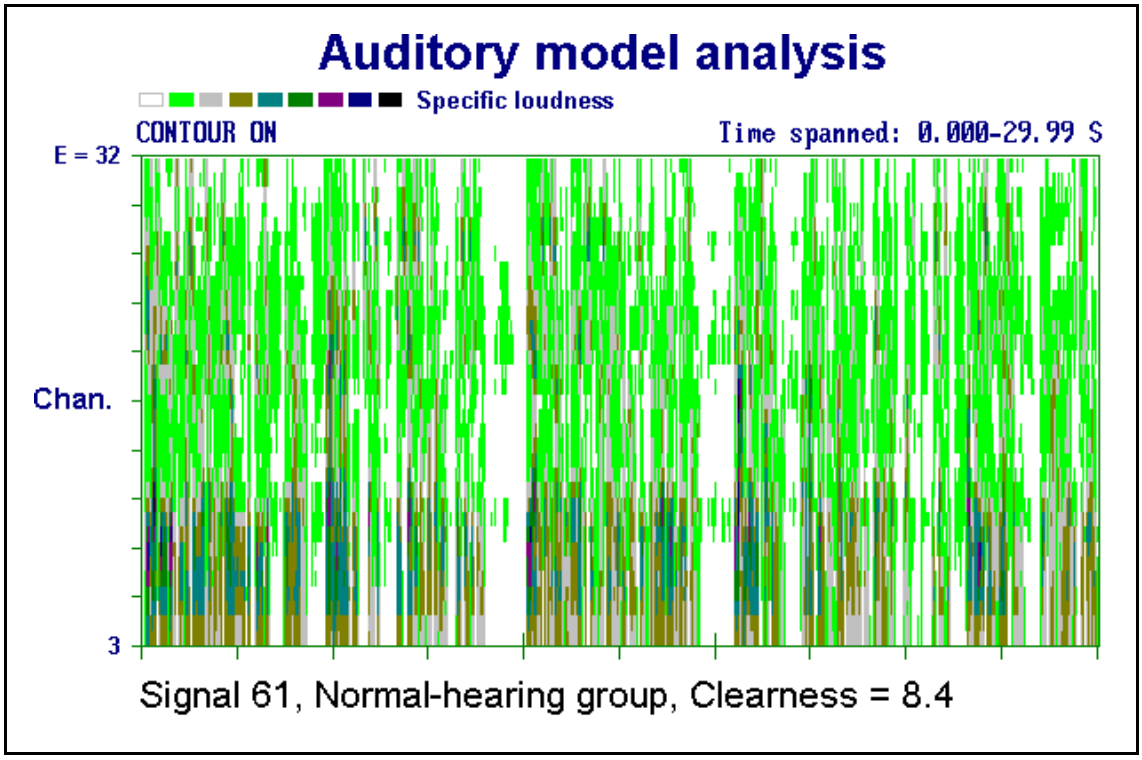
The auditory model of course allows for individual parameter files, such that individual subjects' audiograms and stimulus levels would affect the calculated values of specific loudness. For the present purpose, it was considered too time-consuming and cumbersome to process all 64 stimulus files on an individual subject basis, since the individual variations in audiogram and stimulus level were relatively small.

By processing the stimuli through the auditory model, 64 output files were generated, each containing roughly 2350 frames, 30 channels wide. To get an initial impression of the data, selected output files were plotted in a spectrogram format, e.g. with time along the x-axis, frequency along the y-axis and specific loudness (N') plotted as a grayscale. This was done for selected files, namely the six signal files that received extreme ratings by the normal-hearing group, on each of three rating scales: Clearness, Sharpness and Spaciousness. These are listed in Nielsen (1992). In figures 4 and 5, the spectrograms for the least and the most Clear stimuli, stimulus 12 and stimulus 61, are shown, along with their average subjective ratings.



4. *Spectrographic plot of specific loudness (N') versus time and frequency. N' is indicated by a gray scale, with linearly spaced steps. Signal shown is the least Clear stimulus (on a scale from 0 to 10) as judged on the average by the Normal-hearing group.*

The loudness spectrogram in Figure 4 shows that no mid- or high-frequency information is present and that the loudness over time is almost constant. There is very little contrast or detail in this picture. The particular stimulus was music with background noise, severely compressed in the low-frequency channel (i.e. little temporal information) and switched off in the mid-frequency and high-frequency channel (i.e. 500 Hz lowpass filtered). The most clear signal shown in Figure 5 below, on the other hand, has a rich and detailed structure with contrasts in both the spectral and the temporal domain. This stimulus was speech with no background noise, and no filtering, compression or clipping, see Nielsen (1992) for details.



5. *Spectrographic plot of specific loudness (N') versus time and frequency. N' is indicated by a gray scale, with linearly spaced steps. Signal shown is the most Clear stimulus, on a scale from 0 to 10, as judged on the average by the Normal-hearing group.*

By inspecting Figures 4 and 5, it is obvious, that there is a lot of information (≅ 2350 frames by 30 channels equals ≅ 70500 data points) in such a loudness "spectrogram", corresponding to only one rating on each of the subjective scales. Furthermore, there are in principle only 64 different stimuli , i.e. 64 different input vectors to the network, which is a very small data sample for NN training. Therefore, the size of the network input layer must be kept small, and thus, some type of data reduction of the input data is required. Ideally this data reduction should not discard any perceptually relevant information from the stimulus, with respect to perceived sound quality.

4.1.1 Spectral data reduction.

In the present configuration, the auditory model uses 30 channels, which may be more than required to distinguish between the input stimuli used in the subjective rating experiment (Nielsen, 1992).

One option is to investigate correlation between the network inputs. If there is a large correlation between any of the auditory model channels, we can reduce the effective number of dimensions, before presenting the data to the neural net. Before training, a principal-component analysis (PCA) on input data can reduce the dimensionality and provide orthogonal input vectors. If one can imagine the input vectors in a 30-dimensional vector space, the PCA will determine principal axes in the data "cloud", organized in descending order, based on size of the axes. This is similar to determination of the main axes in an ellipse, for the 2-dimensional case. In mathematical terms, the PCA determines the eigenvectors and the eigenvalues for the covariance matrix of the data, and extracts the eigenvectors for the largest eigenvalues, such that a given percentage of data variance is accounted for. The original input data is then multiplied by the matrix of extracted eigenvectors, i.e. projected onto the principal axes, which are fewer that the original data dimensionality. The network size will be reduced, thus making the size of the training set sufficient. A similar approach has been used by Bennani (1990) for a speaker-identification task. See Hertz et al (1991) or Conradsen (1984b) for further details on PCA.

In the frequency domain, the following analysis was done to investigate redundancy in the auditory model output: The 64 stimuli were analyzed through the auditory model, using the parameters listed in Figure 3. Each output file consisted of $\cong$ 2350 frames by 30 channels equals $\cong$ 70500 data points. For selected output files, a correlation matrix was calculated by means of a software spreadsheet. This is similar to the covariance matrix, but normalized with respect to the internal variance in each variable - the pattern in the two types of matrices are the same, but the correlation matrix is easier to interpret.

The correlation matrices should ideally reflect on the auditory model and not the input stimulus, but the two are intertwined when the auditory model output is analyzed, using the stimuli from the subjective rating experiment as input. As an alternative, one could use idealized input signals, such as pure tones and/or noise signals, but these are difficult to identify properly, since the auditory model performs complex level-dependent operations on the stimulus. If the correlation matrices could be obtained for a cross-section of all stimuli, a more general result would be obtained, but this was not possible due to memory limitations in the spreadsheet software.

Instead, correlation matrices were examined separately for a number of the input stimuli. One example of this is shown in Appendix 11.3. There was a high degree of correlation ($r > 0.7$) along the diagonal of the matrices, with different rates of decay away from the diagonal. Thus, the highest correlation was between adjacent auditory channels. Thus, for a reduction of auditory model output dimensionality, adjacent channels could be grouped, i.e. summed together two-by-two or three-by-three, forming 15- or 10-dimensional output vectors. This approach is meaningful from a psychoacoustic point of view, since we must assume that adjacent channels are correlated to some degree, due to the overlapping filter skirts in the auditory channel. The spectral resolution in the output is obviously reduced, but still large enough to resolve major spectral differences in the input stimulus. Change in bandwidth was found to be the major cause of a changed perceived sound quality (Nielsen, 1992), compared to severe clipping or compression in three frequency bands, and even a model with only 10 outputs should be able to detect these changes. An alternative was to use the auditory model with fewer channels, thus deviating from the critical-band concept. Furthermore, this would leave no freedom for post-processing of the model output, which was faster than running the auditory model again (~ 20 min. per stimulus).

Another means of channel reduction is the aforementioned principal component analysis (PCA). This is a type of "unsupervised" or "unintelligent" data reduction, that in this case will reflect the input stimulus as well as the auditory model. This type of analysis will remove correlated information from the model output, but may also in some cases

discard crucial information for a succeeding neural network . The major problem, however is that the PCA will have different outcomes, depending on the type of input stimulus. Thus, this type of analysis was avoided for the first experiment. Furthermore, experiment indicated that there were severe memory limitations in existing software (Matlab 3.0) which were exceeded by the ≅ 2350 input vectors per stimulus.

In the following, only the 30-to-10 channel data reduction has been used.

4.1.2 Temporal data reduction.

As previously mentioned, a single stimulus (30 sec.) produces ≅ 2350 frames by 30 channels equals ≅ 70500 data points from the auditory model. With the proposed channel reduction in the frequency domain, there will be ≅ 2350 frames by 10 channels equals ≅ 23500 data points. The same stimulus results in 1 quality rating vector (6 scales). Somehow the temporal dimension in the auditory model output must be considered.

One solution to this time problem could be a time-delay neural network (TDNN - Waibel & Hampshire, 1989) or a temporally recurrent network (Dolson, 1989). A TDNN would for instance accept all input frames on successive input nodes to the network, i.e. ≅ 23500 input nodes with frequency domain data reduction, which is clearly not feasible. The temporally recurrent network is a network with internal memory and internal feed-back from previous network states. Such a NN could in principle be used to cycle through the time frames, and settle in a state corresponding to the quality rating for a given stimulus. The theoretical and practical issues with such a model are many, and no clear examples of similar applications were found in the literature.

Thus a much simpler strategy was tried initially, namely to extract certain statistics for each channel in the reduced-channel output, and use these with a static (memoryless) neural net. Examples of statistics are: Mean, standard deviation, median, maximum and minimum. In the following, mean and standard deviation of the specific loudness output has been used. The rationale behind this is: Mean specific loudness characterizes the

long-term spectral balance of the stimulus, and should be able to detect spectral deformations of the stimulus such as band-limiting, resonant peaks and frequency shaping in general as well as harmonic and intermodulation distortion. The standard deviation provides information about the temporal characteristics of the signal, or at least how much the loudness fluctuates over time.

At a late stage during this study, another interesting idea was formed, namely to use the average differential N'(t) as input, i.e. average (N'(t) - N'(t-1)). This quantity should presumably represent the amount of transients in the signal and be of relevance to the Sharpness ratings and possibly also the Clearness ratings, but the idea was not pursued further.

4.1.3 Other inputs.

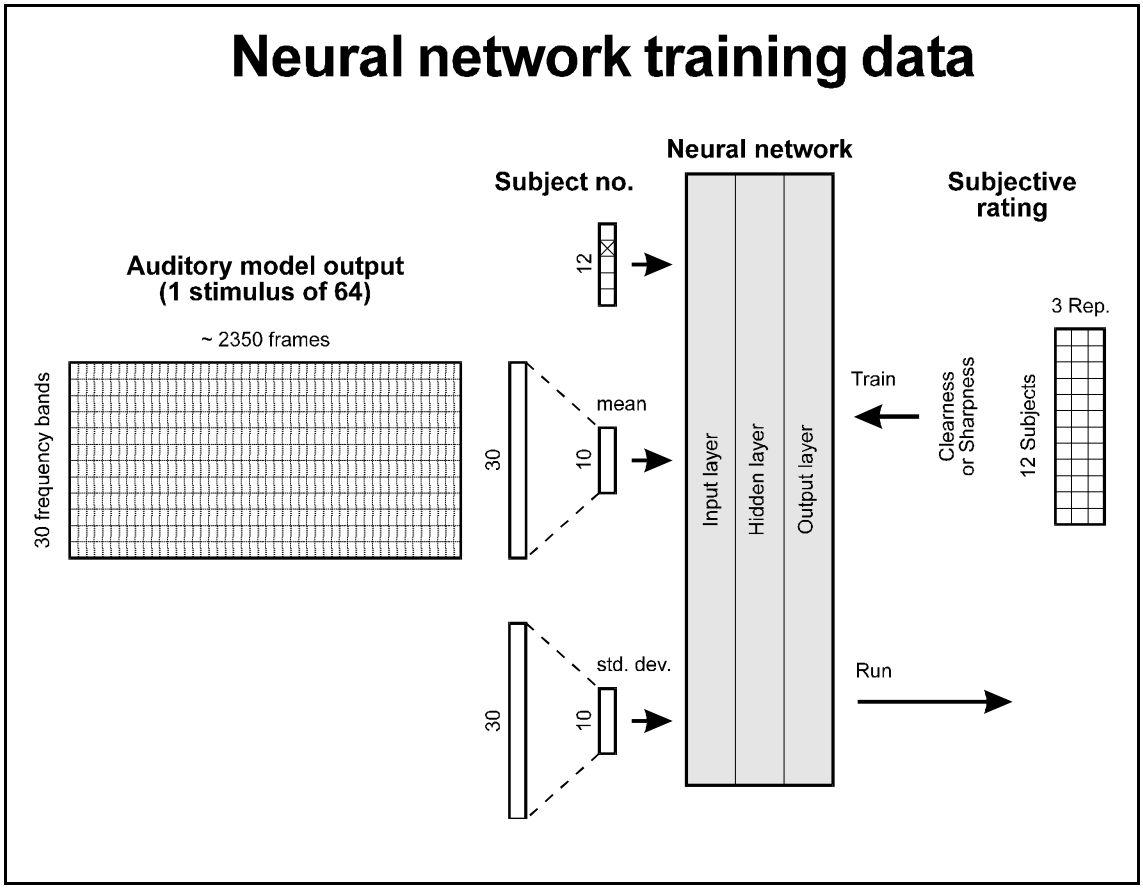
In the subjective rating data, the variance due to changing stimuli is the largest effect (Nielsen, 1992), which is accounted for by presenting the NN with the auditory model output. However, statistical analysis of the results also indicated a large subject effect (Nielsen, 1992). That is, the subjects centered their responses differently on the scales. If the NN input layer sees auditory model (AM) output only (with fixed audiogram and signal level across subjects), the corresponding responses will diverge for the same input. During training, the NN will never converge completely, due to variance in the training output values from the subjects.

The subject effect can be accounted for by providing the network with an input, identifying the listener whose ratings are to be predicted. The best way to do this is to use binary-valued inputs, one for each subject in the group, i.e. 12 inputs for the normal-hearing group and 11 for the hearing-impaired group. The use of 12 (11) binary inputs results in many more network weights, than the use of 1 input neuron with 12 (11) different input values, but despite this, the use of binary inputs generally leads to a better training result (Lawrence et al, 1992).

Since each subject made three ratings on successive days, the day count could of course also be used as a NN input, but this effect is either not significant or very small compared to the stimulus and subject effects (Nielsen, 1992) and was thus not included. The task of averaging this small effect was thus left to the neural net.

During training, the entire data set for the subject group (normal-hearing or hearing-impaired) is cycled through, while the subject inputs are alternating. This is straightforward. The question is: How is the trained network used for prediction, if we must choose one of the original subjects for the prediction? The solution is to present the output of the auditory model for the given stimulus to the NN input layer, while cycling through the subject neurons and successively setting them to 1 followed by 0. The 12 (11) output values from this process are stored and averaged to produce an average rating, in the same way that an average would be calculated from all subjects in a subject group.

The types of data representation used in the following training sessions is summarized in Figure 6 below, indicating both data reduction in the frequency and the time-domain.



6. *Schematic presentation of the data inputs and outputs to the neural network and the types of data reduction used to facilitate training.*

4.2 Output data representation.

The rating results on the network output side can also be represented in different ways. Factor analysis on the data showed that there were two prominent rating scales (Nielsen, 1992): Clearness and Sharpness. For the current experiment, it was decided to train one network for each of the two scales separately in order to keep network complexity down and to ease the interpretation of the weights in the trained network. In principle, this also allows for different types of data representation for the two scales. For instance, if the judgment of Sharpness relies primarily on spectral information, and Clearness depends on temporal information as well.

As indicated in Figure 6, there are 3 replications for each subject and stimulus, giving 36 and 33 ratings per stimulus, for the normal-hearing and the hearing-impaired groups respectively The number of output values are:

NH listeners: 64 stimuli * 12 subjects * 3 replications = 2304

HI listeners: 64 stimuli * 11 subjects * 3 replications = 2112

Using this number of rating data, the network will be trained on conflicting data (due to replication effects and subject effects, if these are not accounted for (Section 4.1.3). However, the network may still be successful in extracting the relevant information while ignoring outliers. This approach was preferred instead of training on mean data only, since the averaging might remove important information or be strongly affected by outliers.

Instead of training one ensemble network for each group, the simpler approach of training one NN for each subject was tried first. The NN would then be individual and not have to adapt to the larger intersubject effect.

Of course, it should also be possible to train one network to cover both groups together, if the perceived sound quality depends on specific loudness only, i.e. if the impaired auditory model correctly includes all aspects of hearing loss. This approach was not tested in the present investigation due to time limitations.

4.3 Neural network implementation.

The neural network is implemented by means of a commercially available software package, BrainMaker Professional version 2.52 from California Scientific Software (Lawrence et al, 1992), which includes BrainMaker for defining, training, testing and running neural nets and NetMaker for building fact files (training data), test files (test data) and network definition files. BrainMaker can implement a standard multilayer perceptron with from 1 up to 6 hidden layers and up to 8192 neurons per layer. During

training and testing, activation in the input and output neurons can be viewed along with training and test statistics.

The basic structure chosen for the current project is a 3-layer multilayer perceptron, thus a single hidden layer was used. The input layer is simply a distribution of signals to all input weights of the first hidden layer. The hidden layer acts as a feature extractor, and should thus contain enough neurons to extract the salient features of the data. The optimal number of hidden units (neurons) in this layer is difficult to determine, but typically it should be chosen between the number of input nodes (10 - 32 depending on the choice of input data) and output units (1) and should be much less than the number of training samples. With these rules-of-thumb, the network sizes have been chosen for the training runs in this study. BrainMaker has a feature to modify the network structure during training, by adding hidden units to improve the network fitting to the training data. This type of constructive learning has been used in all training sessions in section 6.

5 Model training and testing principles.

5.1 Training algorithm.

BrainMaker Professional uses the classic Back-Propagation (BP) algorithm to update all weights. This is a generalization of the least mean squares (LMS) algorithm which uses a gradient search to minimize a cost function equal to the mean square difference between the desired and the actual net output across all *N* data points (in the case of a single output neuron):

$$MS = \frac{1}{N} \sum_{i=1}^{N} (y_i - o_i)^2 \qquad \textbf{(4)}$$

Where y_i is the actual net output and o_i is the desired net output. This cost function is also referred to as the error surface in a *n*-dimensional space (corresponding to *n* weights), where the gradient search attempts to find the global minimum. The net is trained by initially selecting small random weights and then presenting all training data (facts) repeatedly, while updating the weights by means of the BP algorithm (Rumelhart & McClelland, 1986a). Weights are adjusted after each data sample, and the next fact is presented to the net. It is important that there is no trend in the training facts as a function of the data sequence. This is usually handled by randomizing the order of facts before training begins.

When training is commenced, the network will produce random outputs, but gradually it will "learn" the data and converge towards some finite mean squared (MS) error. The weights are adjusted proportionally to the magnitude of the error and a learning rate between 0 and 1. A low learning rate causes the network to converge slowly, thus training may take a long time. A high learning rate may cause the network to speed past the minimum of the error surface and perhaps oscillate around the minimum without ever learning the facts optimally. In BrainMaker, the learning rate can be adjusted during training and slowed down, when the error is small, i.e. when the gradient descent is close

to the minimum on the error surface. This feature was used in most of the training sessions in section 6.

There are also features in BrainMaker for destructive learning by pruning small weights or inactive neurons, and thus "simplifying" the network. Constructive learning can also be applied, where neurons are added to the hidden layer, if the training error does not decrease over a given number of training runs. If constructive learning is used, the initial number of hidden neurons should be small (too few), and for destructive learning, training should start with a high number of hidden neurons (too many). In the current investigation constructive learning was used.

5.2 Training performance.

During training, BrainMaker Professional calculates certain training statistics for each training run, i.e. at the end of the training set. The number of bad facts are the number of facts, where the actual output value differs from desired output by more than a given distance, the training tolerance. This statistic provides a simple count of network training progress, but is not the most useful for determining the optimal network state, since only a 'cube' in output space is considered, whereas the magnitude of the error inside the cube is not considered. Other training statistics are: Training run, Total no. of facts, Number of good and bad facts in the last complete training run and in the training run before that. Training will stop when the bad fact count reaches 0. The training tolerance can also be decreased during training when a certain percentage of facts are matched correctly by the network.

Training statistics can be written to files at specific intervals during training. In addition to the above statistics, this file will contain the average error and the RMS error during the past training run, which is the quantity being minimized by the LMS gradient search. Also, the coefficient of determination, R^2, is computed. This is also referred to as the multiple correlation coefficient, which is the ratio of the variance explained by the model,

i.e. total variance (SS_{tot}) minus residual variance (*SSres*) divided by the total variance of the output (SS_{tot}):

$$R^2 = \frac{SS_{tot} - SS_{res}}{SS_{tot}} \quad \textbf{(5)}$$

R^2 should ideally be close to 1, indicating a perfect match between actual network output and desired output.

The same statistics can be computed for the test set, by interrupting training and testing the network at specific intervals during training. The state of the network can also be saved at regular intervals, and after training, the optimal network can be chosen by inspecting the test and training statistics.

5.3 Test facts.

Test facts are facts that the NN has not seen during training, typically obtained from the same original data set. NetMaker can generate a test set automatically by randomly picking out a certain percentage (e.g. 10 %) of the data from the entire set before training. For the current application, we know that there are 36 (normal-hearing) or 33 (hearing-impaired) different outputs for the same input and only 64 different inputs altogether. Thus, if 10% are picked for testing, it is very likely that the same stimuli are represented in both training and test sets, thus it is important to select all ratings for some of the stimuli. For a test set of 10%, which is common, we must pick on the order of 6 stimuli.

Instead of picking the test facts randomly, it was decided to pick a combination of stimuli, such that all stimuli and processing parameters were equally represented. The 64 stimuli have been generated as a fractional factorial experiment, combining two factors on two levels (signal: speech or music and background noise: on or off) with three factors on four levels (each of three frequency bands: off, linear, clipped or compressed) for a total of 256 combinations (See Nielsen (1992) for further details on the experimental design). Out of these, 64 were picked by means of two defining relations

and subsequently blocked into 4*16 blocks by two more defining relations. To select the test set, three defining relations were found that did not coincide with any of the previous relations - we could term this a factorial pick. The stimuli selected in this manner were: 1, 6, 28, 31, 41, 46, 52, 55. See Appendix 11.4 for a table of all stimuli, listing the signal and processing conditions. This way of picking the test stimuli creates a balanced set: 4 with music, 4 with speech, 4 without noise, 4 with noise etc. This test set is very representative of the data set with all types of processing equally represented.

Another way to pick the test set is to set a certain class of stimuli aside for testing. This will probably cause poorer performance on the test set, since training and test sets represent different types of signals and processing. This type of test set is more similar to an optimal situation, where the test data originate from a different experiment than the training data. As an example of this, all speech signals that are clipped in the mid-frequency band were picked for the test set, a very important group of signals, totaling eight stimuli. All other processing parameters were balanced, i.e. 4 with noise and 4 without, 2 switched off in the LF channel etc., see Appendix 11.4 for details. The 8 test stimuli based on the class pick were: 17, 19, 21, 23, 25, 27, 29, 31.

5.4 Test performance.

The trained network will have some prediction error on the test set. The error is acceptable within certain limits, since the test fact itself is noisy. The amount of error that is acceptable depends on the reliability of the particular subject and must be obtained from the statistical analysis. It was decided to use the confidence intervals for the stimulus means as found in the rating experiment (Nielsen, 1992), thus if the test set predictions were outside of the 95% confidence intervals, the prediction error was larger than the data error.

The best available objective criterion to use is the multiple correlation coefficient R^2 from BrainMaker Professional, calculated from the test set. It is difficult to set criteria for this, but R^2 should at least be close to 0.5.

There are theoretical results in the literature concerning the prediction of test performance and generalization based on the training performance (Moody, 1991), but based on the sensitivity to choice of test set in the present experiments - with very small training and test sets - it did not seem justifiable to apply these models.

6 Training sessions.

6.1 Training and test schedule.

The various options for data representation, training parameters etc., are outlined in a table in Figure 7. The table is a chronological listing of the stepwise refinement of input data, network structure, training algorithm etc. This was a stepwise process of incremental changes rather than a systematic combination of all possible parameters, since this is not feasible with so many options and so few theoretical and practical rules. The resulting evolutional process of learning and optimization is presented in this section.

Sess.	Input data	Hid. units	Output data	Training set	Test set	Opt. run	Hid. units	RMS Error	R^2	Comments
780_1	m	3+	Clear	TS 780	Fact.	400	5	0.25	0.28	Rather poor
780_2	m+s	5+	Clear	TS 780	Fact.	1500	15	0.17	0.46	Good prediction, except stim 55
N_1	m+s	5+	Clear	NH	Fact.	400	8	0.19	0.29	Poor for Clearness > 5
N_3	m+s + noise	5+	Clear	NH	Fact.	3300	36	0.23	0.16	Poorer than N_1
N_4	m+s	2+	Clear	NH	Fact.	750	4	0.28	0.08	Slower training: Poorer than N_1
N_5	m+s	10+	Clear	NH	Fact.	150	10	0.18	0.32	Facts shuffled - better than N_1
N_6	m+s	15+	Clear	NH	Fact.	-	-	-	-	As N_5, poorer convergence
N_10	m+s+TS	20+	Clear	NH	Fact.	2300	27	0.14	0.49	Much scatter, good mean results
N_11	m+s+TS	20+	Sharp	NH	Fact.	100	20	0.15	0.48	Slightly worse than N_10
H_12	m+s+TS	20+	Clear	HI	Fact.	100	20	0.18	0.4	Much scatter, good mean results
H_13	m+s+TS	20+	Sharp	HI	Fact.	400	20	0.14	0.53	As N_12, but better means
N_14	m+s+TS	20+	Clear	NH	Class	1450	24	0.17	0.56	Overestimation on test set
N_15	m+s+TS	20+	Sharp	NH	Class	1000	22	0.18	0.35	Underestimating clustered test set
H_16	m+s+TS	20+	Clear	HI	Class	100	20	0.17	0.52	Good, little overest. - test set
H_17	m+s+TS	20+	Sharp	HI	Class	2750	28	0.16	0.34	Very good means

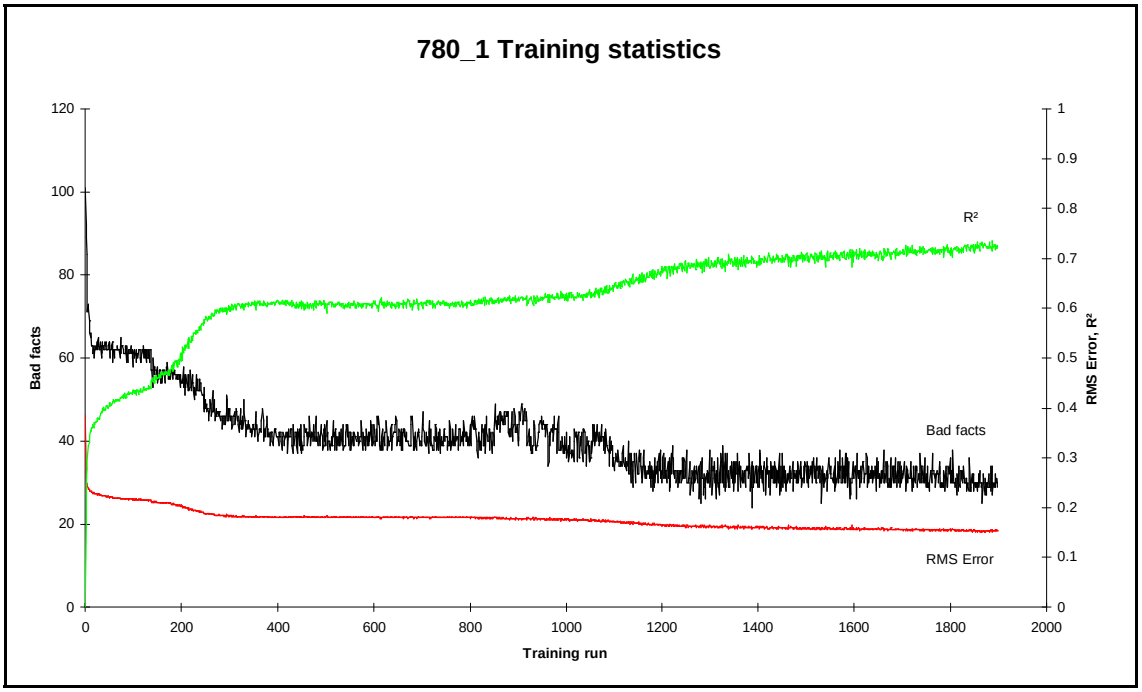
7. *Listing of the conditions tested in a series of training sessions. Explanation of symbols: Input data, m = mean, s = standard deviation, TS = test subject. Hidden units: + = add neurons during training. Training set: TS 780 = Test subject 780 (NH), NH = Normal-Hearing group, HI = Hearing-Impaired group. Test set: Fact = Factorial, Class = One class of stimuli. Optimum = number of training runs for best result on test set: RMS error and multiple correlation, R^2 shown.*

6.2 Single subject.

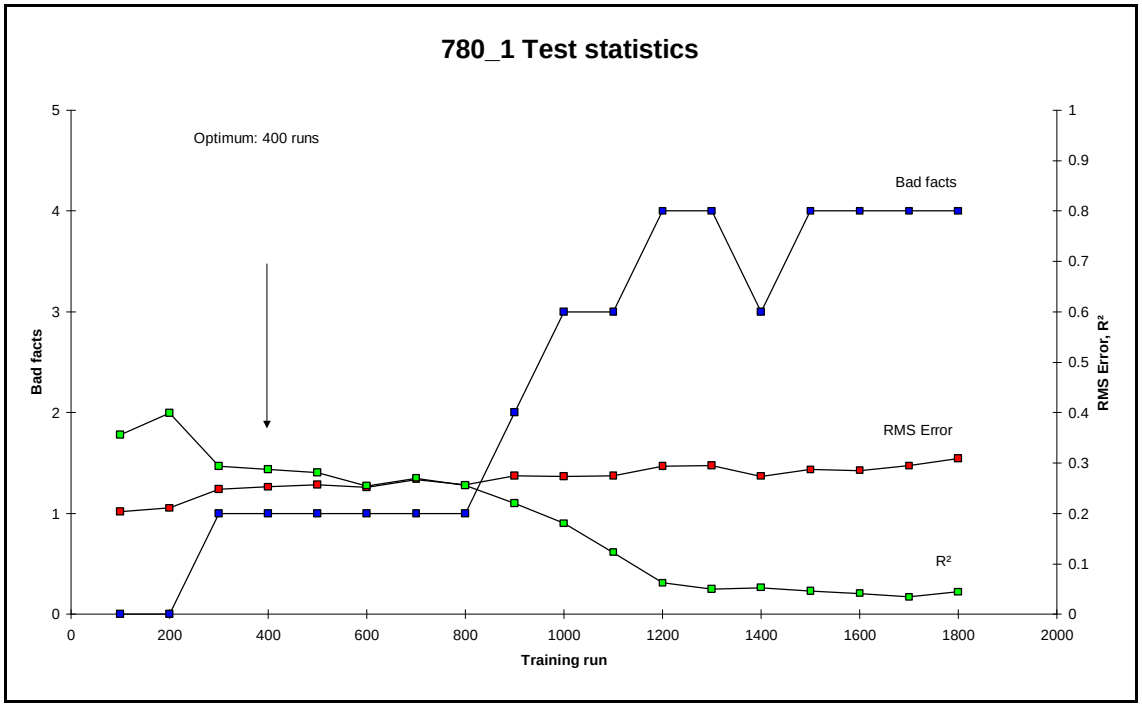
The first training sessions were conducted on a single subject to test the neural net approach on a limited data set.

Test subject 780 (TS 780) was chosen as the most reliable on the Clearness and Sharpness scales combined (Nielsen, 1992). The day-to-day effect for this subject was non-significant ($p > 0.05$) for all scales, except Sharpness, i.e. this subject was very consistent, and therefore provided a relatively well-defined input-output relationship.

The network input consisted of the 10 mean values for specific loudness. The hidden layer contained 3 units (neurons), and the output unit represented Clearness. Units were added to the hidden layer, if the RMS error decreased by less than 0.05 over 100 training runs. The training set had the stimuli arranged in the same order as used for the subjective evaluation, and the test set was picked on the factorial basis (section 5.3). The training and test statistics as a function of training runs are shown below in Figures 8 and 9.



8. *Training progress for session 780_1, subject 780, mean specific loudness only. In some cases, performance is improved, as hidden neurons are added.*

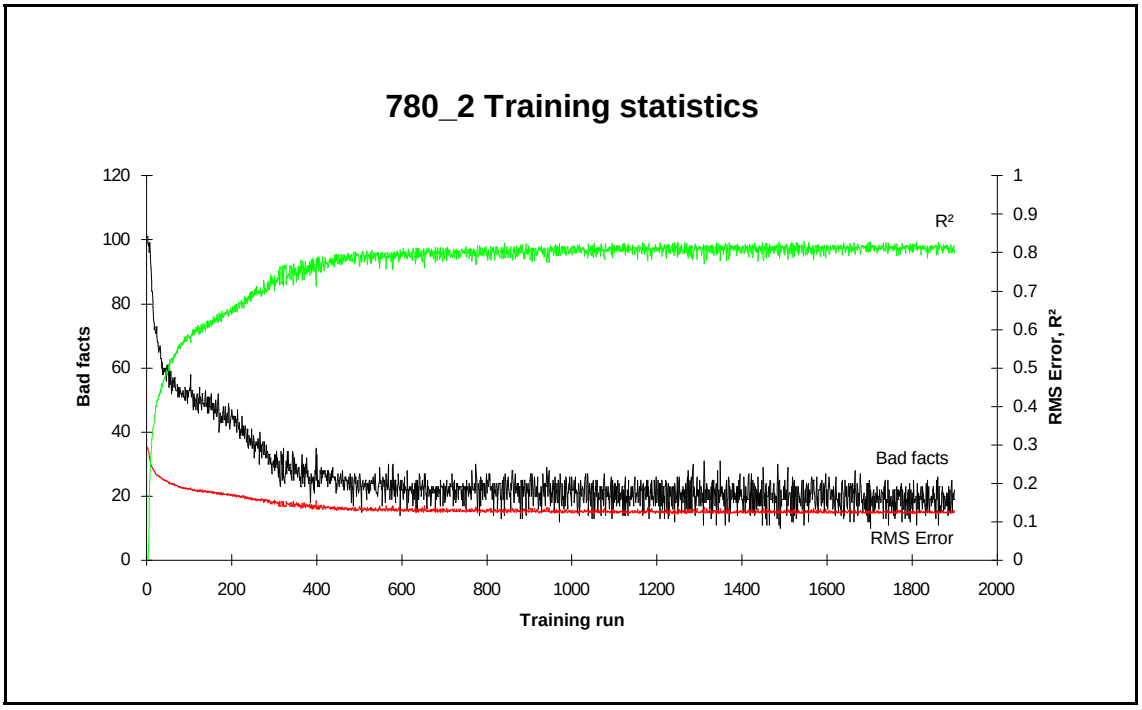


9. *Training progress (test set) for session 780_1, subject 780, mean specific loudness only. Arrow indicates, where optimal performance was reached for training and test set combined.*

The optimal performance of the network was reached around 400 training runs, as a compromise between minimum Bad facts, minimum RMS Error and maximum R^2 on both training and test sets. On the test set alone, the optimum was reached at 200 already, but the training performance had not nearly converged at that point. Addition of hidden neurons improved performance on the training set, while degrading it on the test set. At 400 runs, the network contained 5 hidden neurons.

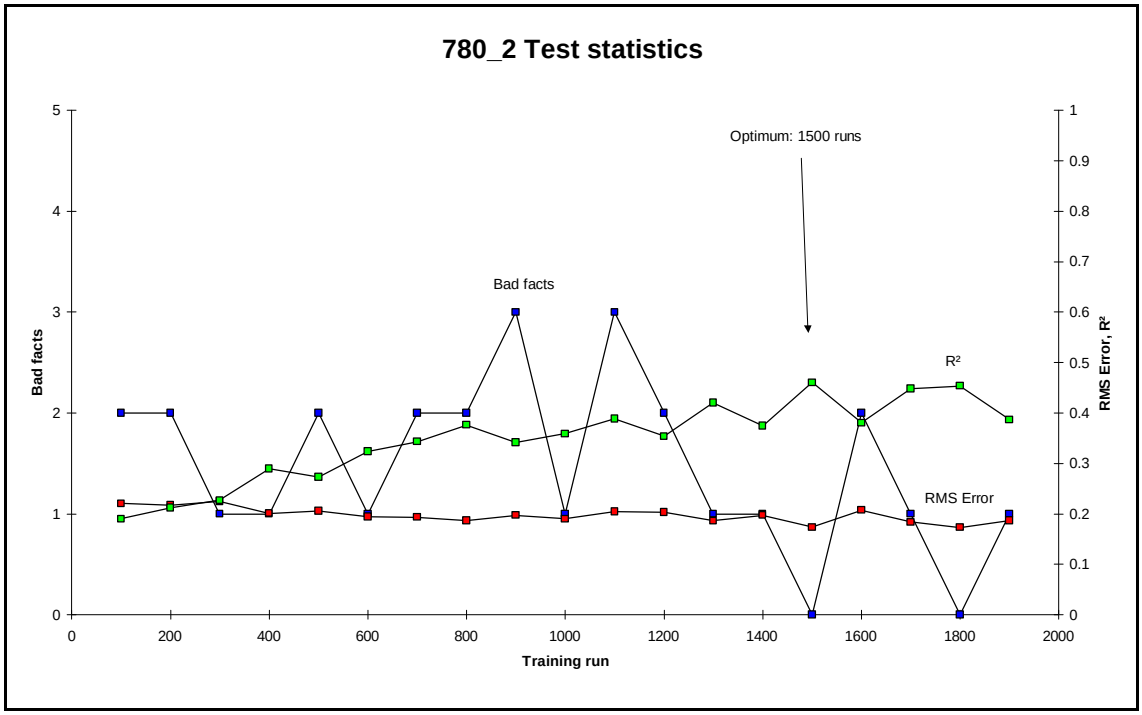
On the test set, the resulting performance was: RMS Error = 0.25 and R^2 = 0.28, which is not good. The divergence between training and test set performance could be due to lack of useful input data, i.e. mean specific loudness is not adequate in accounting for perceived sound quality.

Thus, the next training session, 780_2, included 10 more input units, the standard deviation for each of the 10 bands after reduction from 30. The network initially contained 5 hidden units, with 1 more added after 100 training runs with less than 0.05 decrease in RMS Error. The remaining options were identical to 780_1. The training (Figure 10) showed convergence to constant performance around 500 runs, while the test set performance continued improving (Figure 11).



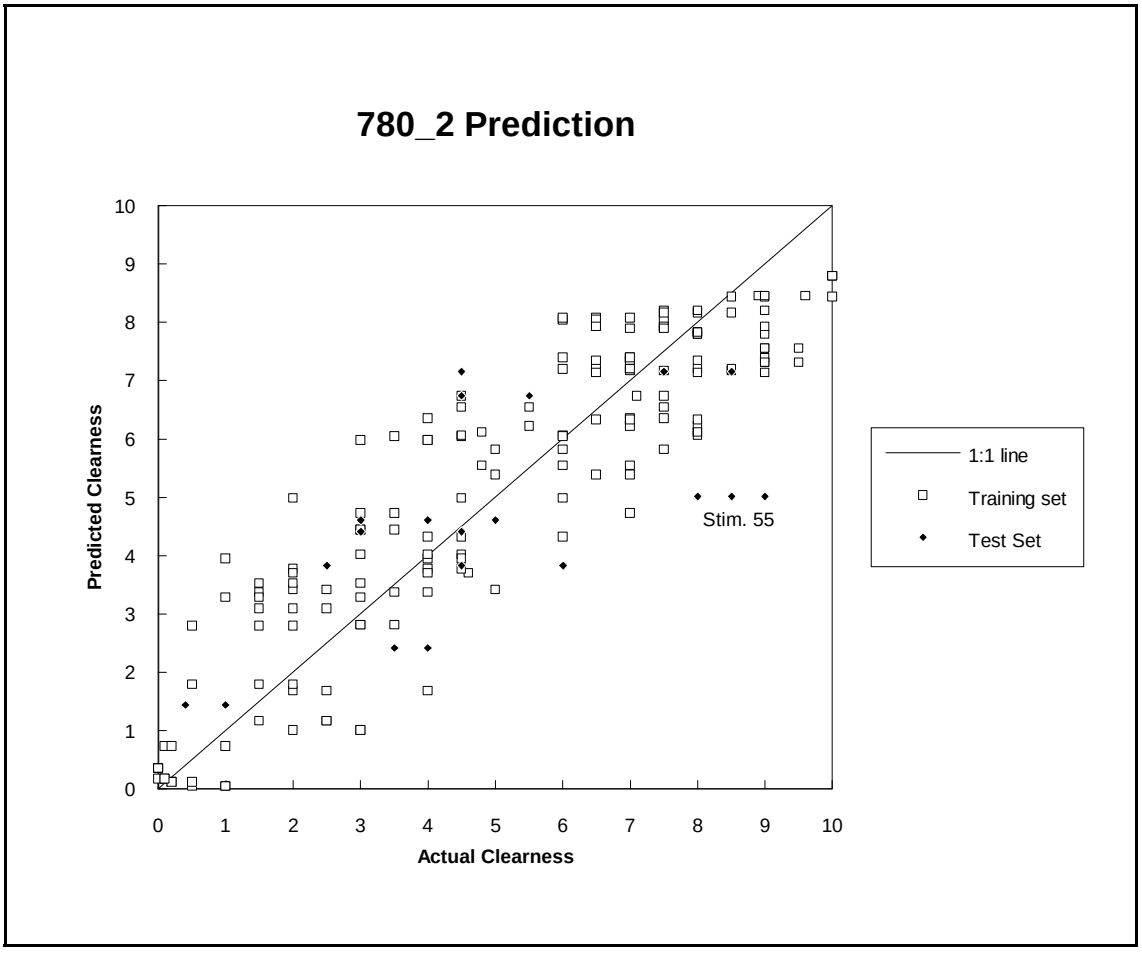
10. *Training progress for session 780_2, subject 780, mean and standard deviation input. Hidden units were added during training.*

Therefore, choosing the optimal training point was based on the test set only, which had 0 bad facts (using a 40% test tolerance) after 1500 training runs. Here, the RMS Error was 0.17 and the R^2 was 0.46, a substantial improvement over session 780_1.



11. *Training progress (test set) for session 780_2, subject 780, mean and standard deviation of specific loudness. Arrow indicates, where optimal performance was reached for the test set.*

Using this optimal network, the training and test vectors were run through the network in order to compare the predicted values from the model with the actual clearness ratings - this is shown in Figure 12. The figure shows that there are often 3 points in a group with same predicted and different actual ratings, corresponding to the three ratings per stimulus. In some cases, there have been identical actual ratings, thus two or three points have been plotted on top of each other. The training set data form a parallel band around the 1:1 line, and the test set points are within that band, except for one group of three points. This group is stimulus 55, which is speech with background noise, clipped in the LF and HF channels and unchanged in the MF band (see appendix 11.4). The model underestimates the Clearness of this stimulus by roughly 2.5. A comparison of mean actual ratings to mean predicted ratings was not done.



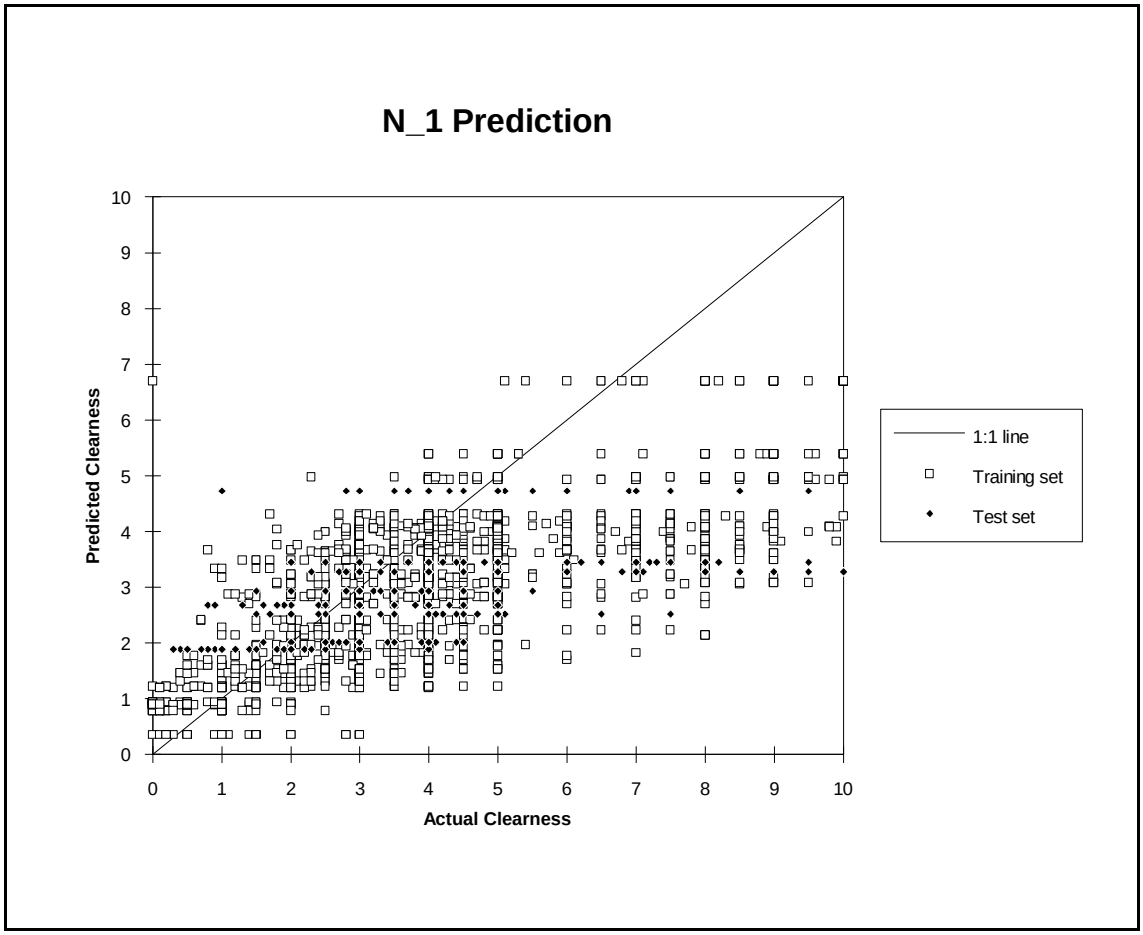
12. *Predicted vs. actual ratings of Clearness for test subject 780, training session 2. The plot contains 64 stimuli * 3 ratings = 192 data points, however some may be plotted on top of each other. The three deviating test set points represent stimulus 55.*

6.3 Subject group (Normal hearing).

The next step was to include an entire group of subjects, which was done for the normal-hearing group in session N_1 through N_6. Using the same 20 inputs for mean and standard deviation of specific loudness, the neural net would likely need to handle more divergent data, since the subjects used the rating scales differently (Nielsen, 1992).

In session N_1, 5 hidden units were used initially, and hidden units were added during training if the RMS error decreased by less than 0.05 during 100 training runs. The output unit represented Clearness. The test set was picked on the factorial basis, i.e. the same 8 stimuli as before, totaling 12 subjects*3 repetitions*8 stimuli = 288 data points for testing and the remaining 12 subjects*3 repetitions*56 stimuli = 2016 data points for

training. The training set was sorted by subject and stimulus, i.e. subject 1, stimulus 1, stimulus 2 etc. The optimum was found at 400 training runs and 8 hidden units, with an RMS Error of 0.19 and $R^2 = 0.29$ - a rather poor multiple correlation coefficient. The poor prediction performance on both training and test sets is shown in Figure 13.



13. *Predicted vs. actual ratings of Clearness for the Normal-Hearing subject group. Clearness is very poorly predicted for Clearness ratings above 5. The data point on the y-axis (y ~ 6.7) represents a missing observation that was erroneously set to 0 by BrainMaker.*

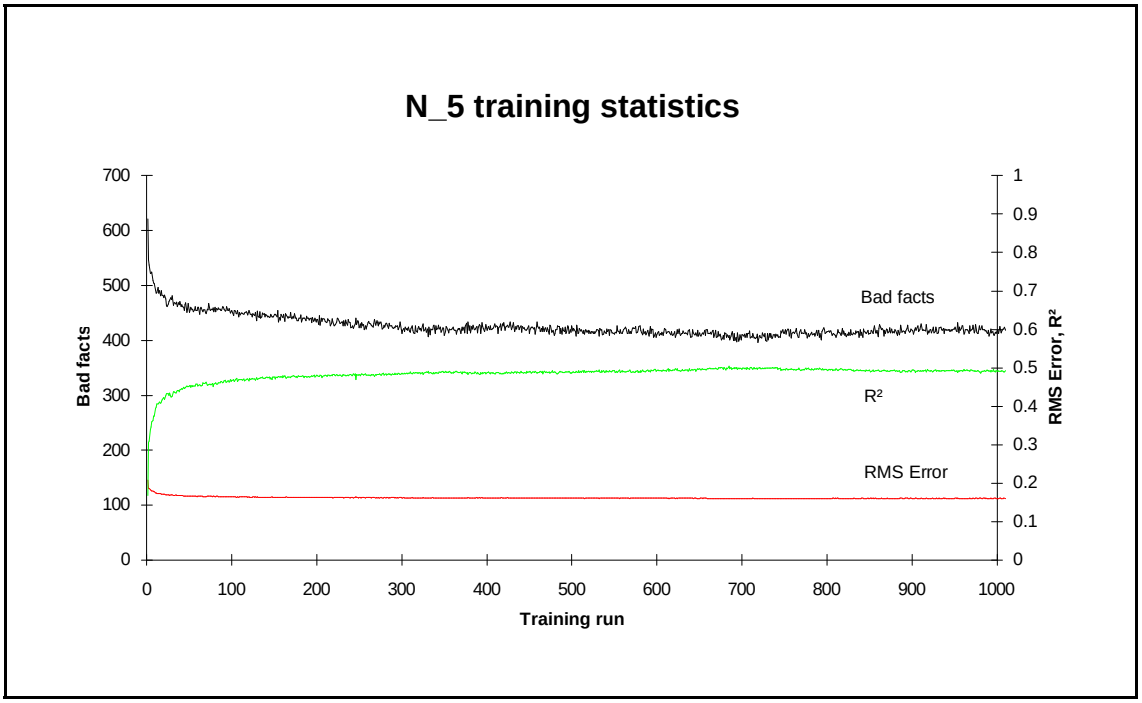
This network is clearly not capable of fitting the diverging data, in particular above 5 on the Clearness scale, where prediction errors are large.

In session N_3, an attempt to improve convergence and generalization was made by means of added input noise. Random noise (amplitude 0.01 re. 1.0 maximum value for normalized inputs) made the input vectors slightly different all the time, qualitatively corresponding to the output noise inherent in the Clearness ratings. However, the input

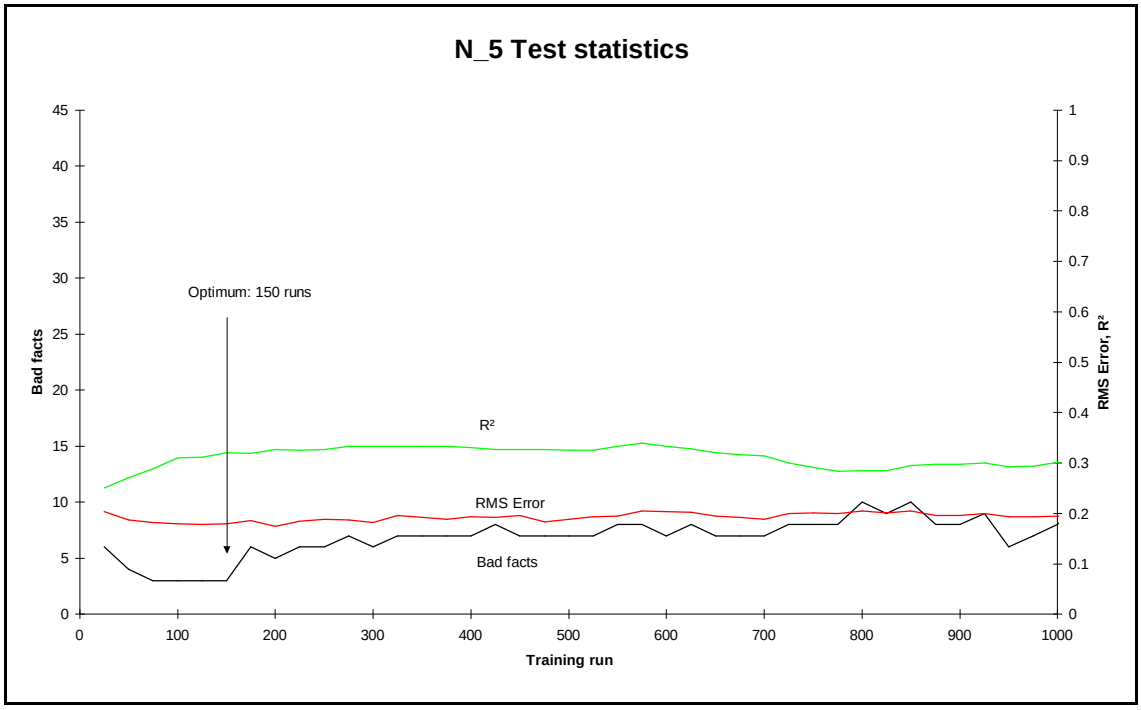
and output noise were obviously not correlated. The training session gave poorer performance on the test set than N_1, and the use of input noise was abandoned.

Session N_4 used a lower learning rate (1, then 0.8 if more than 75% of the facts were correct) and slower addition of hidden units (if RMS error decreased by less than 0.01 over 200 runs) from the initial 2 units, with no improvement in performance.

In session N_5, the training facts were presented to the network in pseudo-random order, by using the "Shuffle rows" feature in the NetMaker program to shuffle the facts prior to training. Furthermore, the network started at 10 hidden units, with the same slow addition of units as in N_4. This led to some improvement in training and test performance and was the best result for the NH group with specific loudness inputs only. The training and test statistics are shown in Figures 14 and 15.



14. *Training progress for session N_5. The network converges fast on the training set.*



15. *Network N_5 performance on the test set. Performance increases quickly and decreases as hidden units are added.*

The optimum for this network was after 150 training runs, with 10 hidden units, resulting in an RMS Error = 0.18 and $R^2 = 0.32$. This result is slightly better than N_1, but not as good as the single-subject result from session 780_2. Adding more hidden units degraded the test performance.

A final attempt was made with session N_6, with a slower learning rate of 0.6 (when more than 75% of training facts are correct), but this degraded the test set performance, probably because too many hidden neurons were added. Therefore, no optimum was found for this session.

To summarize: Training one network on the entire group of normal-hearing subjects, using specific loudness inputs only does not produce satisfying results.

6.4 Subject group, with subject input.

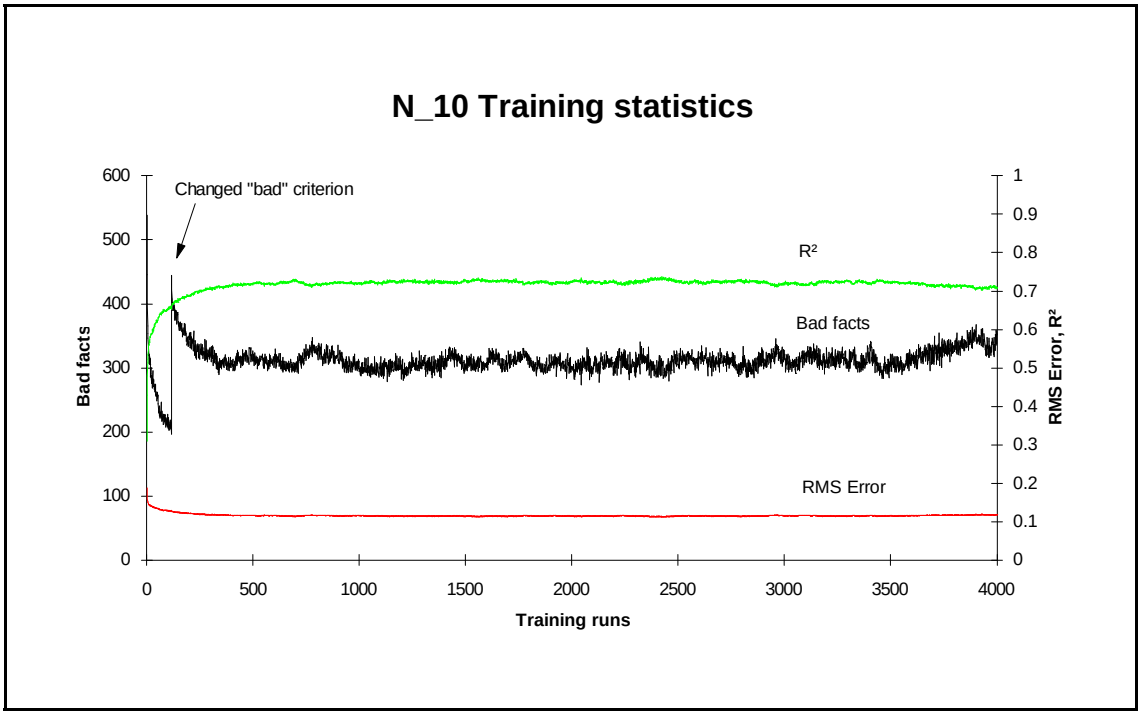
The next step was to account for the subject effect in the ratings by informing the neural net about the current test subject. New training sets were prepared where the subject

number was included and transformed into symbolic information by NetMaker, i.e. the neural net had one binary input for each test subject. When training data for a given subject was used, the corresponding input unit was set to 1 and the remaining test subject inputs were set to 0. See figure 6 for an illustration of this principle.

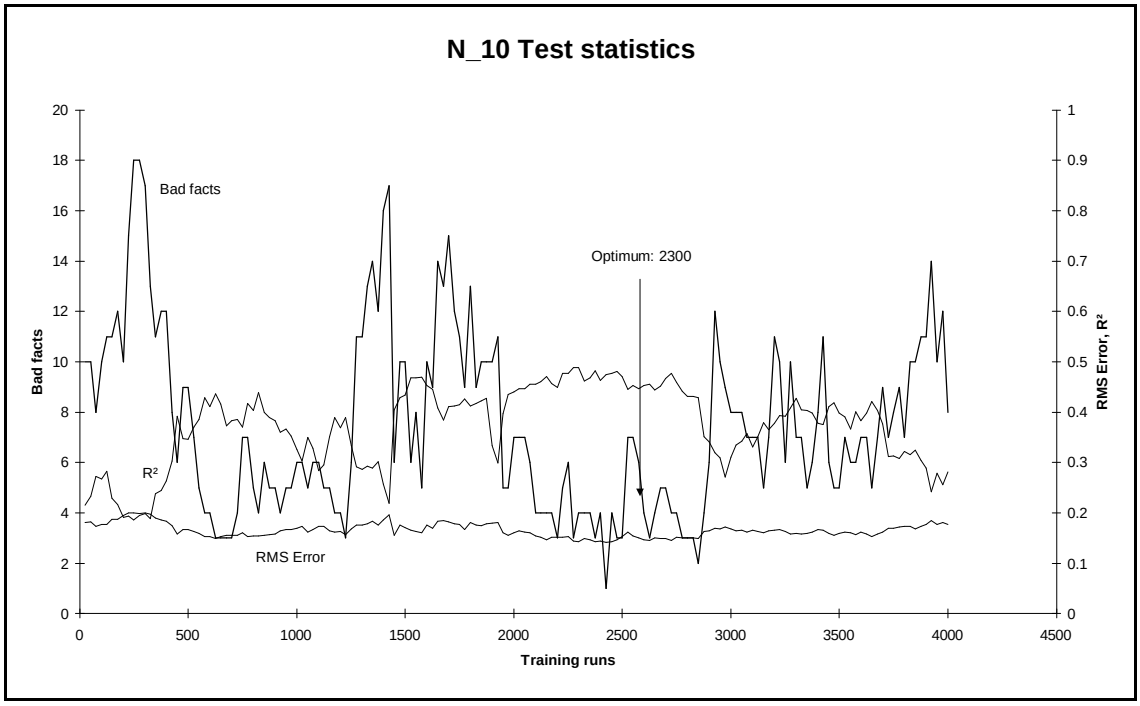
The corresponding training sessions were labeled N_10 through H_13, where N stands for Normal-hearing and H is for Hearing-impaired.

6.4.1 Normal hearing.

Training session N_10 used 32 input units, 20 from the auditory model and 12 representing the 12 test subjects. The initial number of hidden units was 20, with additional units added if the training RMS Error decreased by less than 0.01 in 200 runs. The output unit represented Clearness for the Normal-hearing group. The training and test progress is shown in Figures 16 and 17 below.



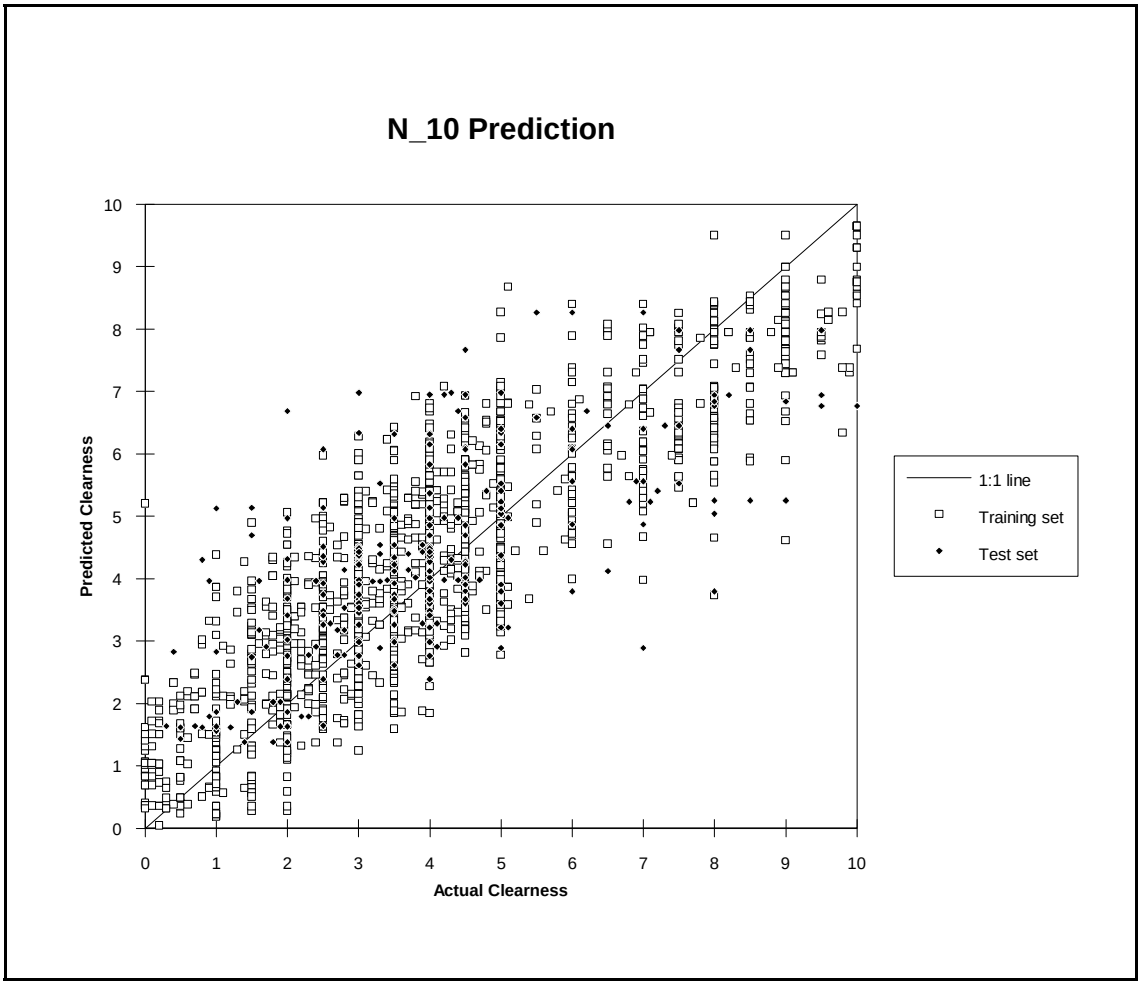
16. *Training progress for session N_10. The network definition file was set up to decrease the training tolerance from 0.2 by a factor 0.8 when more than 90% of the training facts were correct according to present training tolerance.*



17. *Training progress on test set for session N_10. This is a very fluctuating test pattern with the best performance found at 2300 runs. Network state was only saved every 50 runs.*

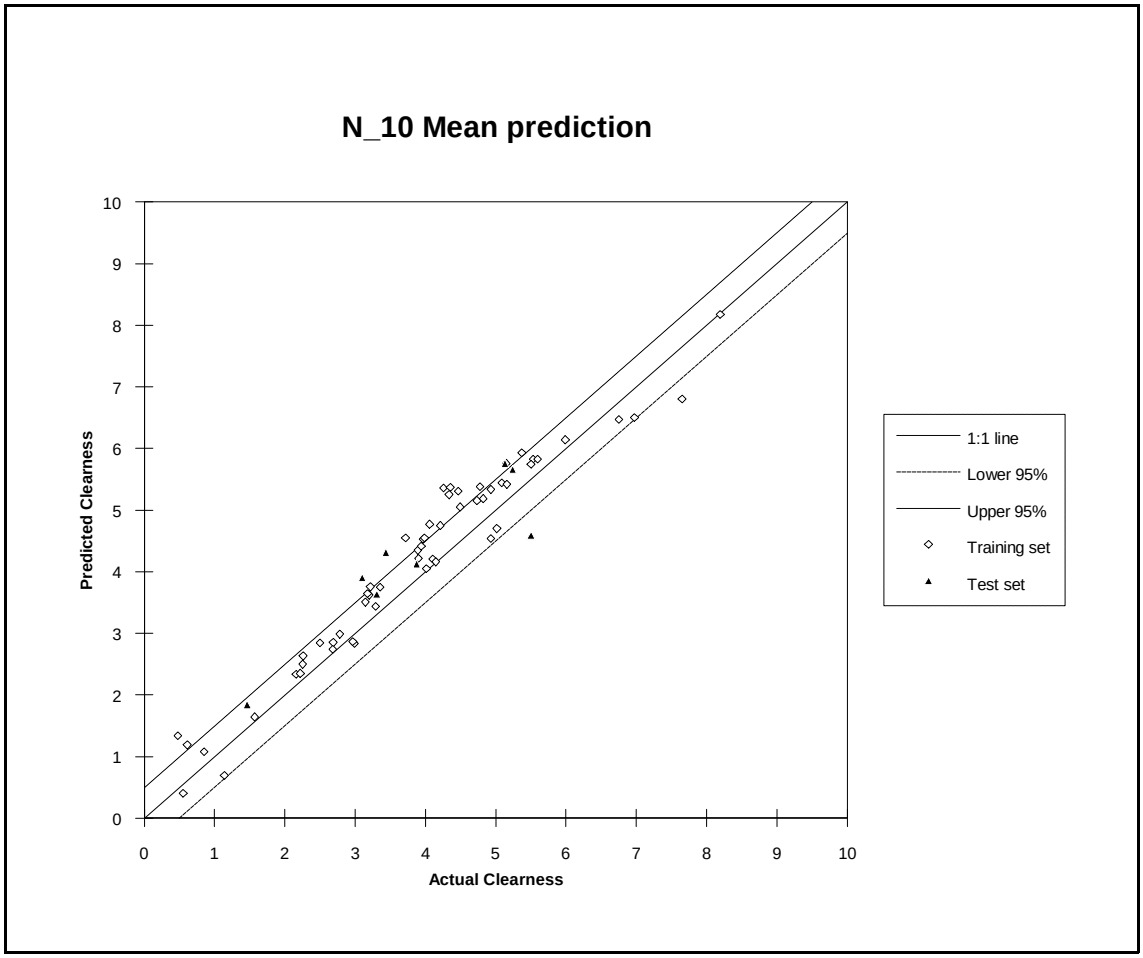
The network converged quickly on the training set, but test performance (generalization) was very fluctuating as training progressed. As a compromise between number of bad facts, RMS Error and R^2, the best point was found after 2300 runs, where RMS Error = 0.19 and R^2 = 0.49. This is better than any of the sessions where subject input was absent (N_1 through N_6) and as good as the single-subject training session 780_2.

The plot of predicted output vs. actual output for network N_10 is shown in Figure 18. Most of the training set is scattered in a symmetrical band around the 1:1 line, but the model underestimates Clearness for actual ratings above 8. This is an area with few samples, most actual ratings are below 5. The test set is mostly within that band with a few test samples outside.



18. *Predicted vs. actual ratings of Clearness for the Normal-Hearing subject group, using a 32-input network with test subject inputs. Clearness is under-predicted for Clearness ratings above 8.*

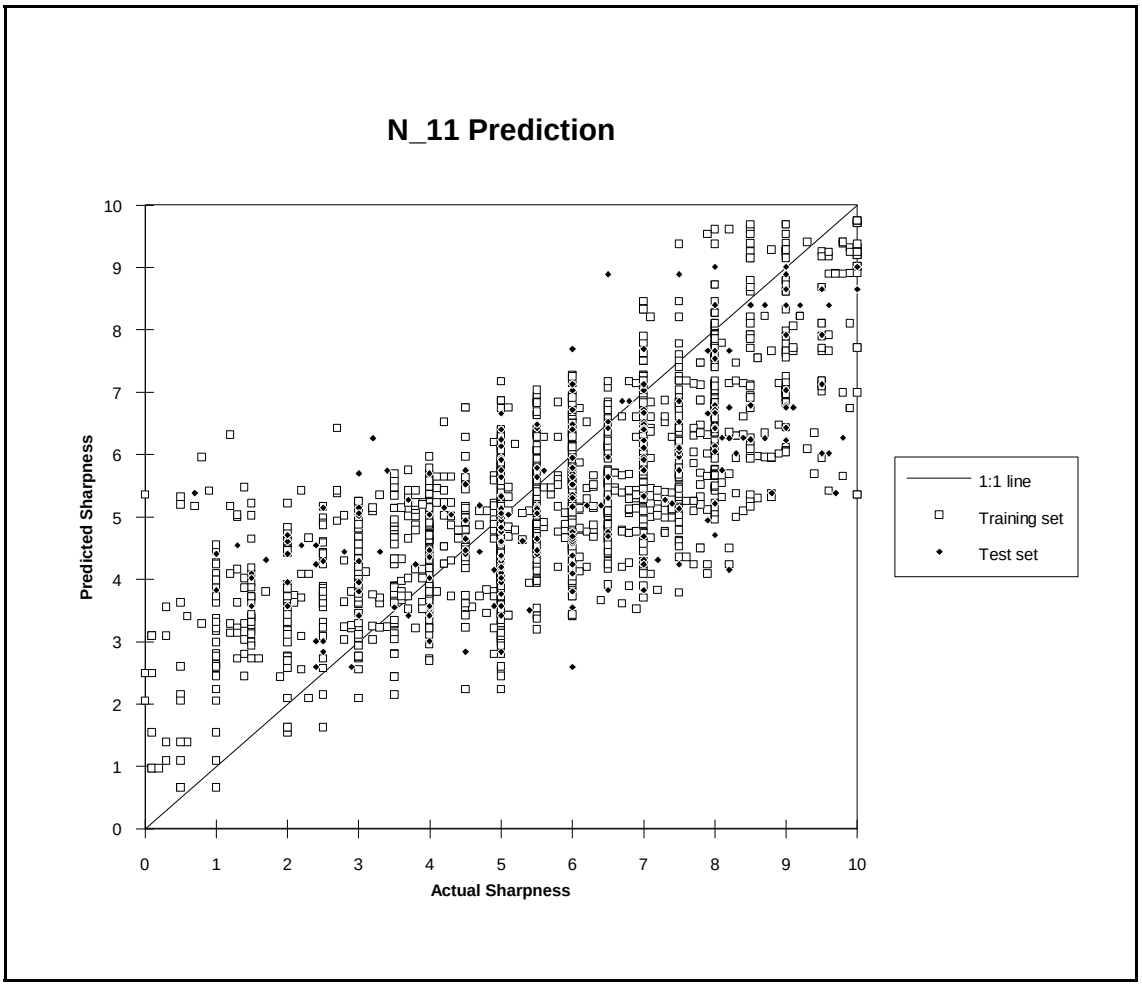
The scatterplot in Figure 18 indicates how well the neural net makes individual predictions, and this can of course be with no more precision than the original noisy rating data. To make predictions based on the entire NH group, predicted and actual means for the 64 stimuli should be compared. The predicted means are calculated by running the input vectors for the 64 stimuli through the NN, once while each of the 12 subject inputs is set high, and the mean is calculated for the resulting 12 outputs. The result is shown in Figure 19.



19. *Predicted vs. actual mean ratings of Clearness for the Normal-Hearing subject group, using a 32-input network with test subject inputs.*

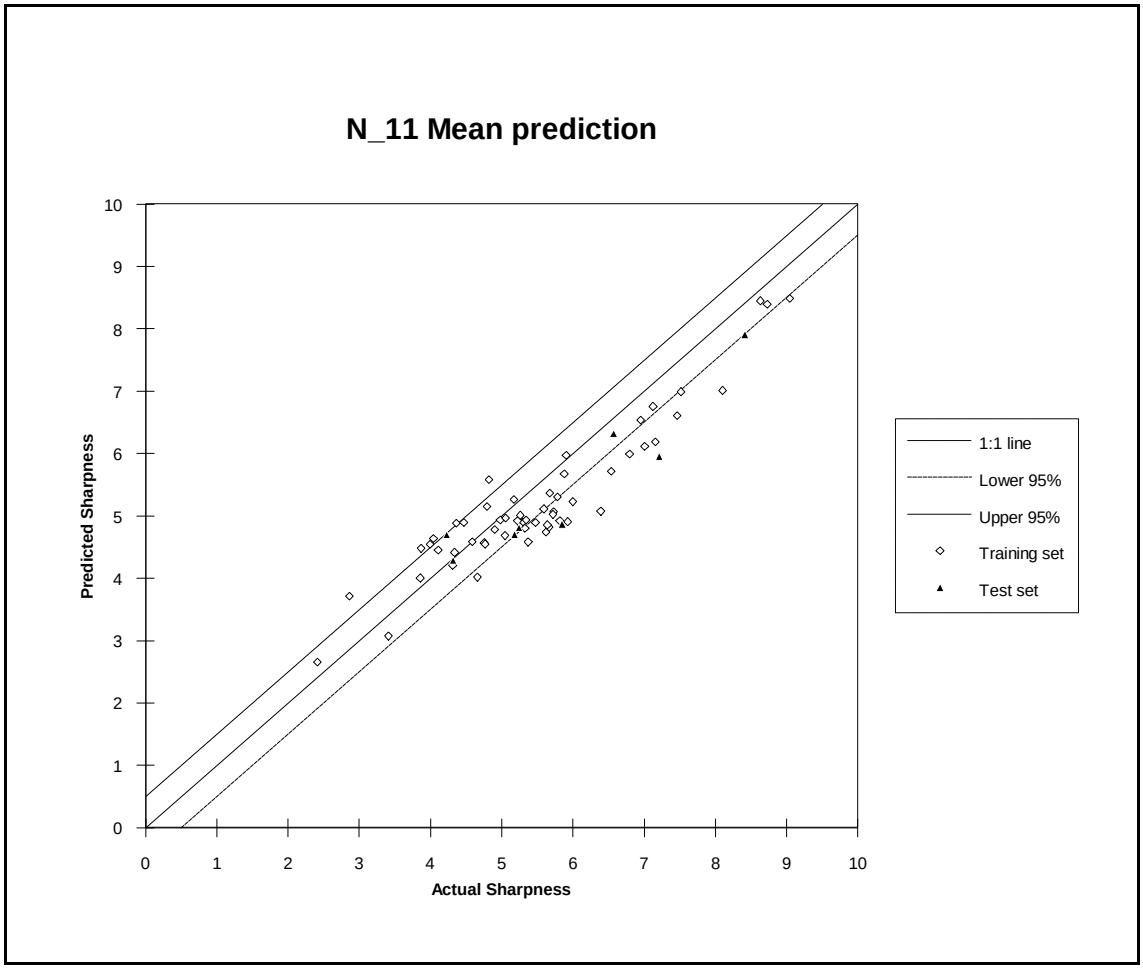
This figure shows very good prediction results for the mean. The training set data points are scattered in a parallel band around the 1:1 line, with some overprediction around 5. The test set points are also in this band, but spread out more. The figure shows that the network is capable of inferring the mean itself, when presented with repeated measures of a noisy rating value. The mean values of the actual rating are of course estimates, with a 95% confidence interval of +/- 0.5. On the figure this could be indicated by a horizontal band, 1 unit wide, around each data point. If this interval does not cross the 1:1 line, then the prediction error is larger than the data set error. In the figure this is shown in a different way, with parallel lines indicating the 95% confidence interval around the 1:1 line. The data points outside of this band have a larger prediction error, but still quite small, on the order of +/- 1.

The same training was then done on the Sharpness scale, with predicted values as shown in Figure 20 below. The test set statistics (figure 7, session N_11) are only slightly worse than for N_10.



20. *Predicted vs. actual ratings of Sharpness for the Normal-hearing subject group, using a 32-input network with test subject inputs.*

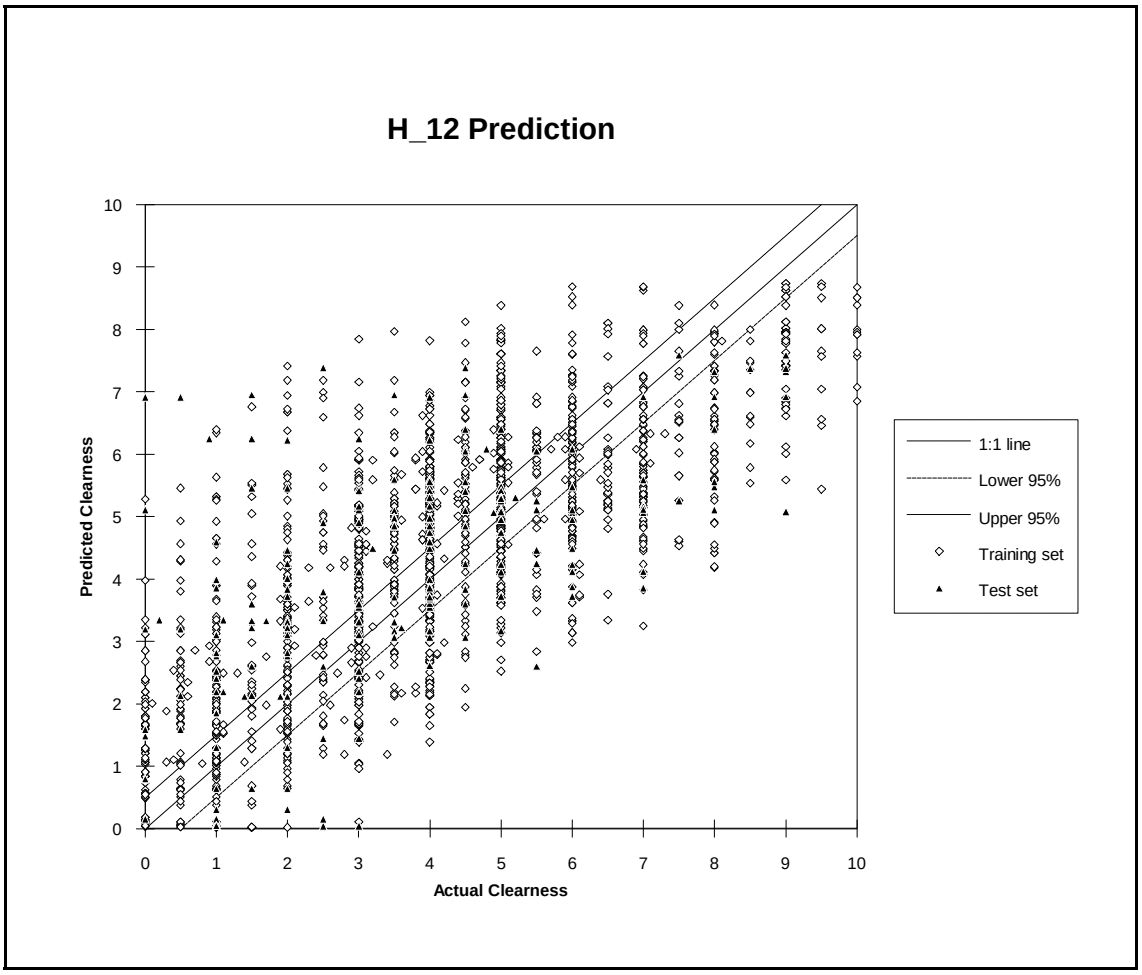
The predictions are generally not quite as good as for Clearness (N_10), and the trend in the scatterplot has a more shallow slope than the 1:1 line. The plot for the mean values, shown in Figure 21, shows mean predictions nearly as good as for N_10.



21. *Predicted vs. actual mean ratings of Sharpness for the Normal-hearing subject group, using a 32-input network with test subject inputs.*

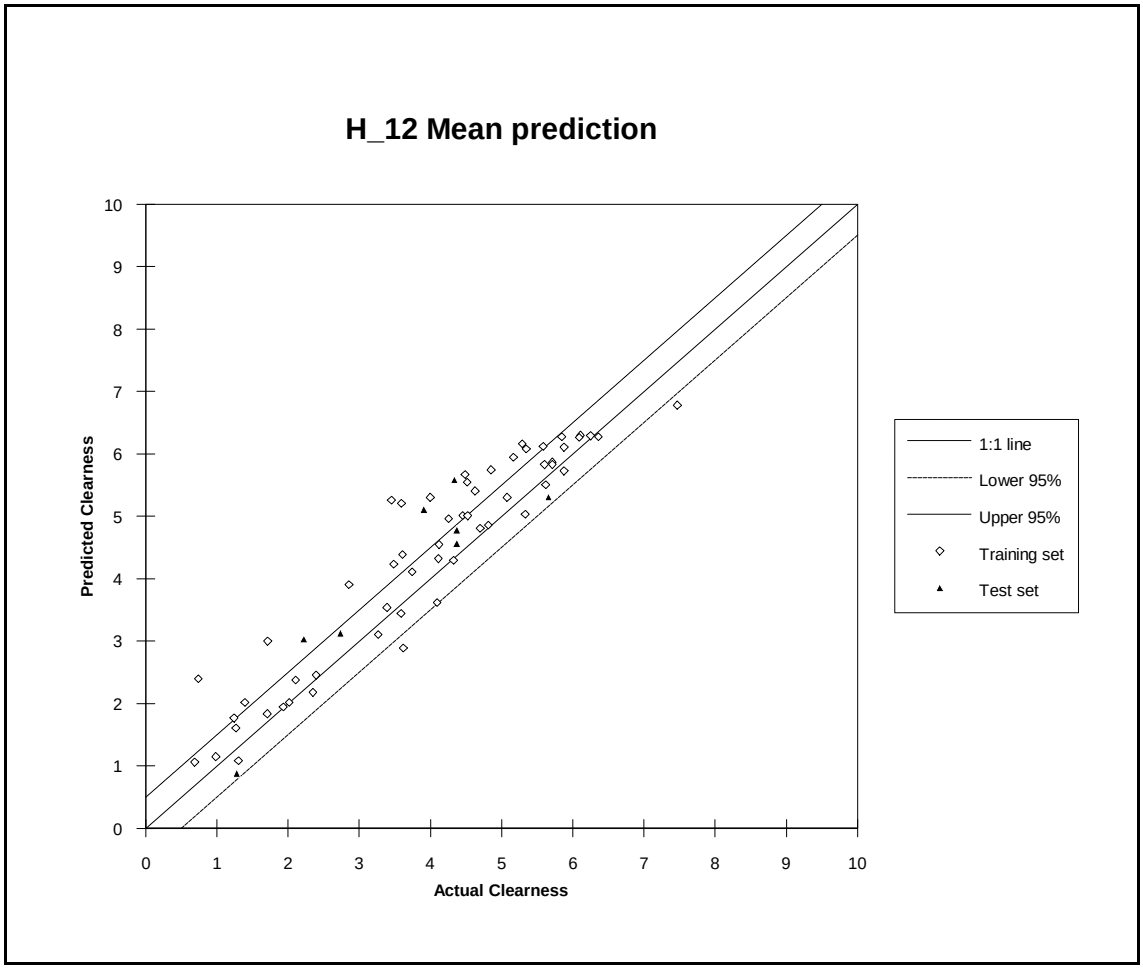
The predicted Sharpness values are below the actual values for ratings above 5, which corresponds to better sound quality (this scale is "inverted" compared with the other rating scales). The underestimation of Sharpness is on the order of 0.5.

6.4.2 Hearing impaired.



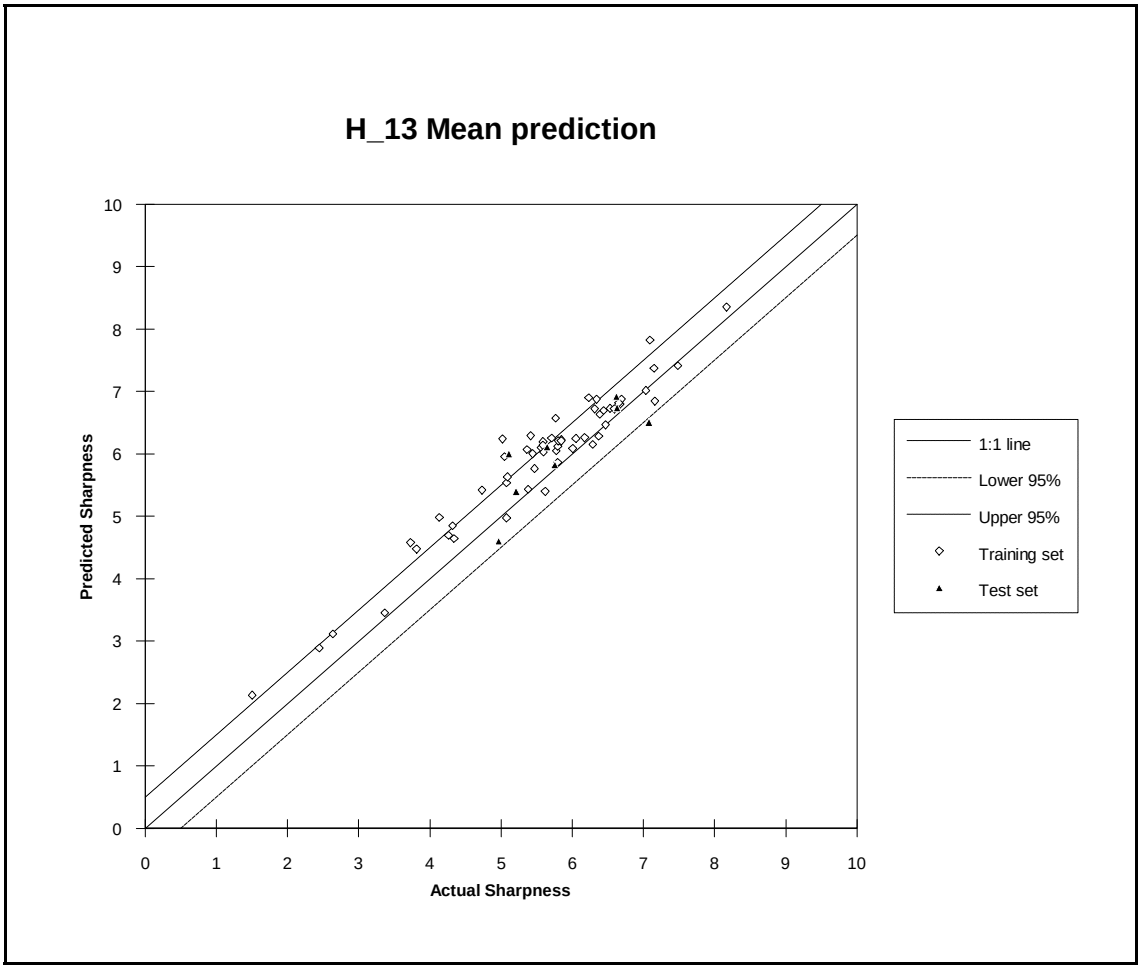
22. *Predicted vs. actual ratings of Clearness for the Hearing-impaired subject group, using a 32-input network with test subject inputs.*

The same network structure was now trained and tested with the rating data from the hearing-impaired group. For Clearness (session H_12), the optimal test set performance was worse than for the NH group (N_10), with RMS Error = 0.18 and $R^2 = 0.4$, not a good correlation coefficient. The scatter plot (Figure 22) also shows a large spread around the 1:1 line.



23. *Predicted vs. actual mean values of Clearness, Hearing-impaired subject group.*

The prediction of mean values, however, is still good, as shown in Figure 23. The band around the 1:1 line is broader compared to the normal-hearing subject group (Figure 19), with some over-prediction of Clearness.



24. *Mean predicted vs. mean actual values of Sharpness for the 64 stimuli, Hearing-impaired group.*

The mean predictions of Sharpness are better than for Clearness (H_13: RMS Error = 0.14 and $R^2 = 0.53$), in fact the mean predictions show a very good fit as shown in figure 24, as good as for the normal-hearing group (Figure 21).

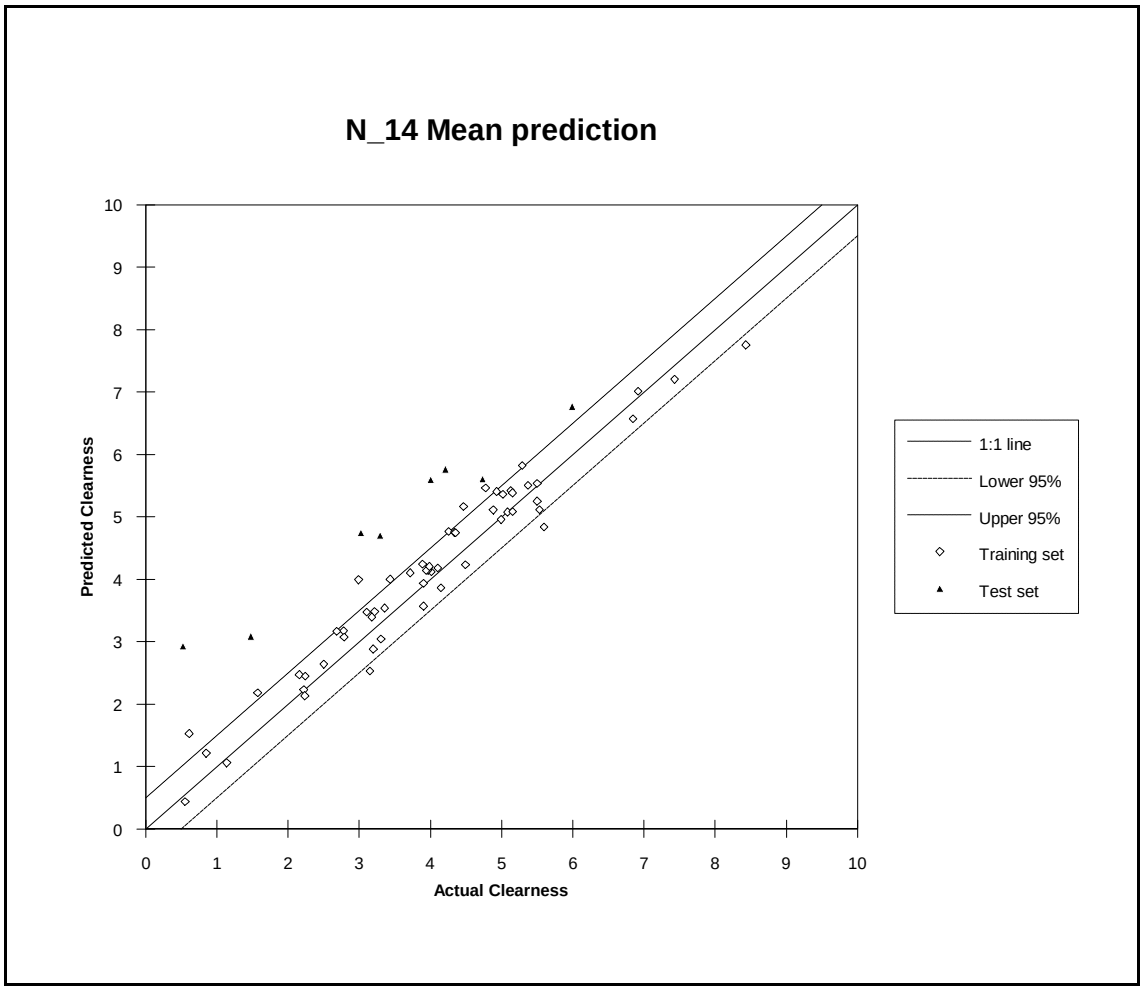
6.5 Test with a class of stimuli.

In the following training sessions, (N_14 through H_17), the training and test sets were made more divergent, by selecting 8 similar stimuli from the original 64 stimuli, namely all speech signals with clipping in the mid-frequency band (see Section 5.3 and Appendix 11.4). This would more closely resemble a situation, where the test data came from an

entirely different experiment. The drawback is that some of the most important stimuli are also omitted from the training set.

6.5.1 Normal hearing.

Two networks were trained: N_14 for Clearness and N_15 for Sharpness. The network for Clearness had slightly larger RMS error than for the same net with a factorial test set (N_10), but also better multiple correlation (0.56 vs. 0.49). The mean predictions for the normal-hearing group are shown below in Figure 25.



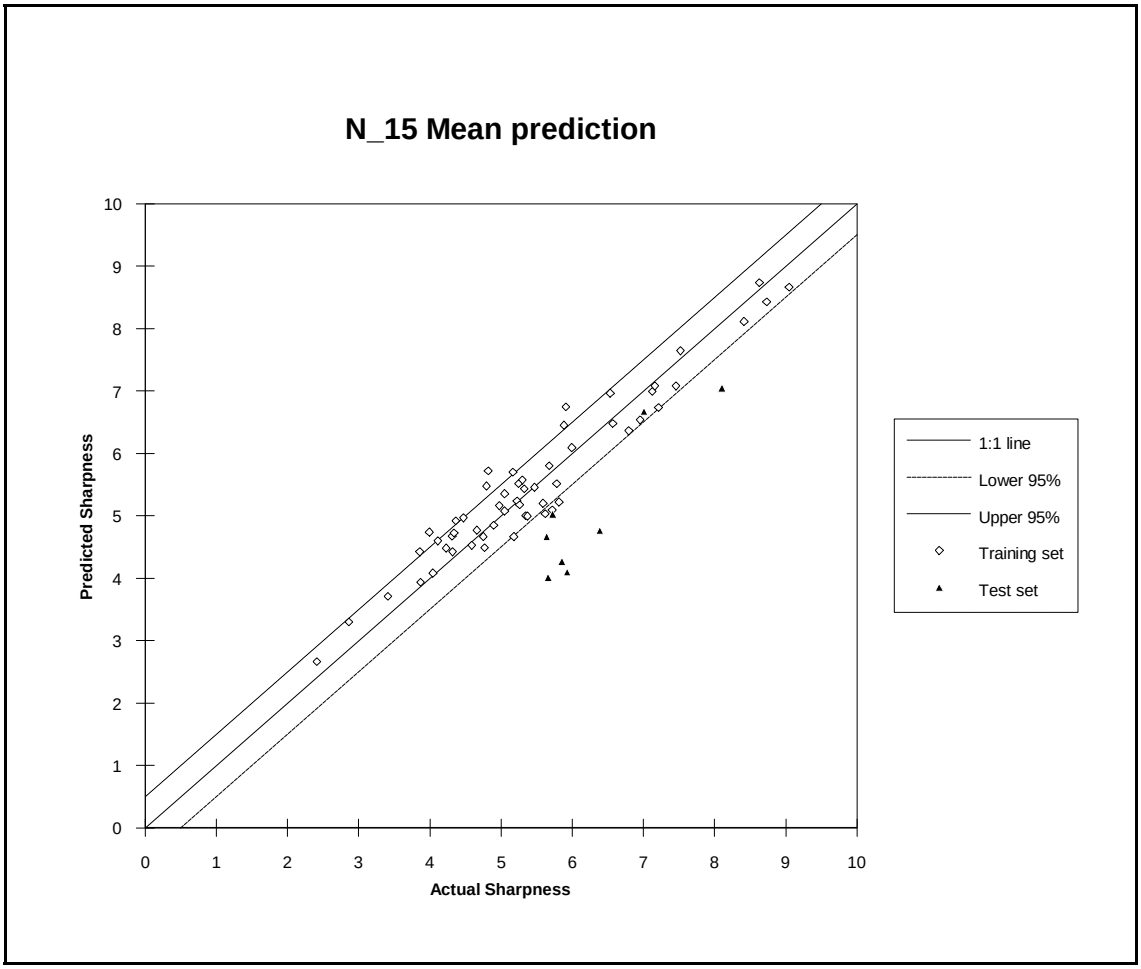
25. *Predicted vs. actual mean Clearness ratings with one class of stimuli in the test set.*

The training performance is as seen previously, with a parallel band around the 1:1 line, but the test set performance is poorer than for the factorial test set N_10 (Figure 19).

All stimuli in the test set are over-predicted beyond the confidence interval of the actual ratings. Performance must be degraded, when an important class of stimuli are removed from the training set.

If a true test set was generated in an independent experiment, the performance would probably lie somewhere between the factorial test set performance (Figure 19) and the class test set performance (Figure 25). These speculations can only be confirmed in a future experiment.

For Sharpness ratings, there is also a larger test set deviation as shown in Figure 26 below, when compared to the factorial test set in session N_11 (Figure 21).

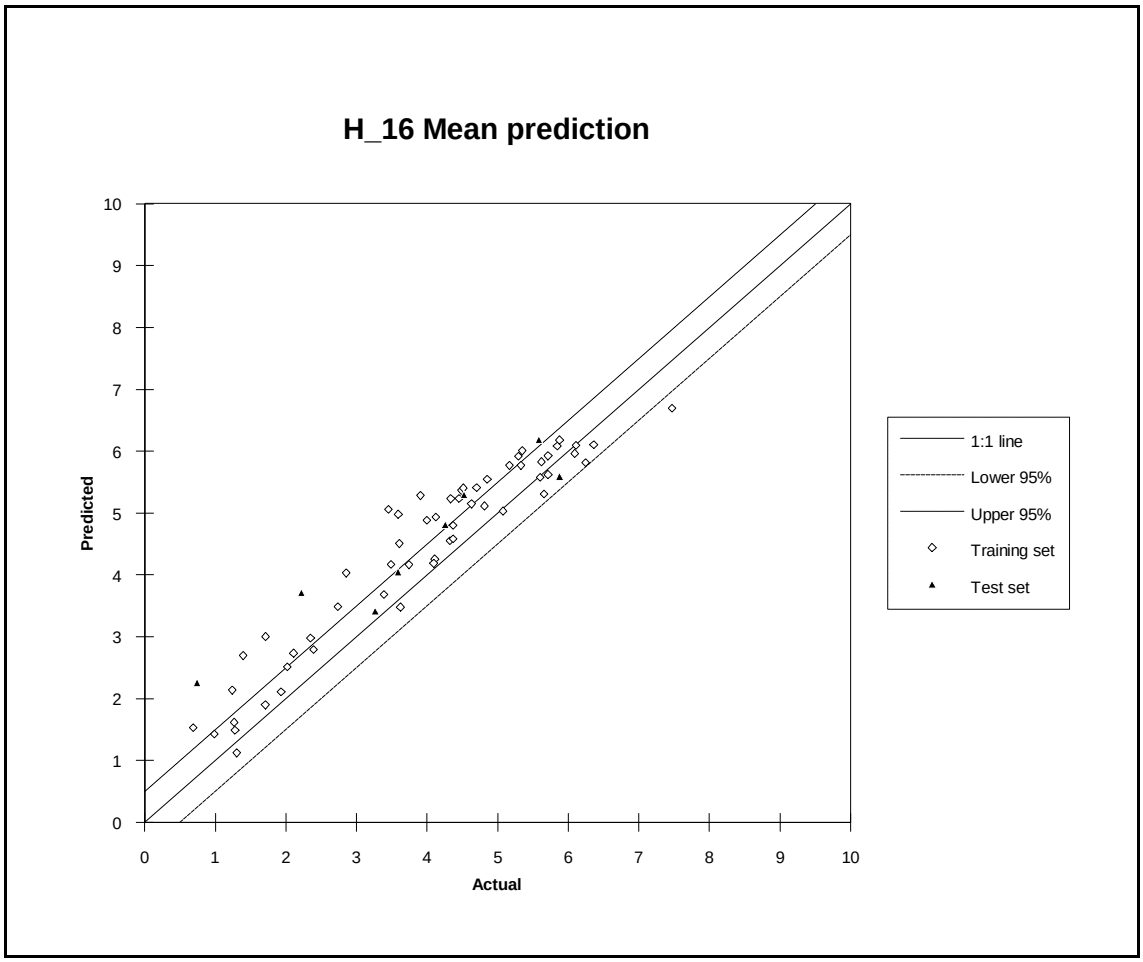


26. *Predicted vs. actual mean Sharpness ratings with one class of stimuli in the test set.*

The Sharpness is under-predicted, but since this scale is oriented opposite of the other scales, the quality is over-predicted, as for the Clearness dimension. For this scale, the test stimuli are not scattered much, but tend to have actual Sharpness ratings around the average. This also explains the low multiple correlation coefficient, $R^2 = 0.35$.

6.5.2 Hearing impaired.

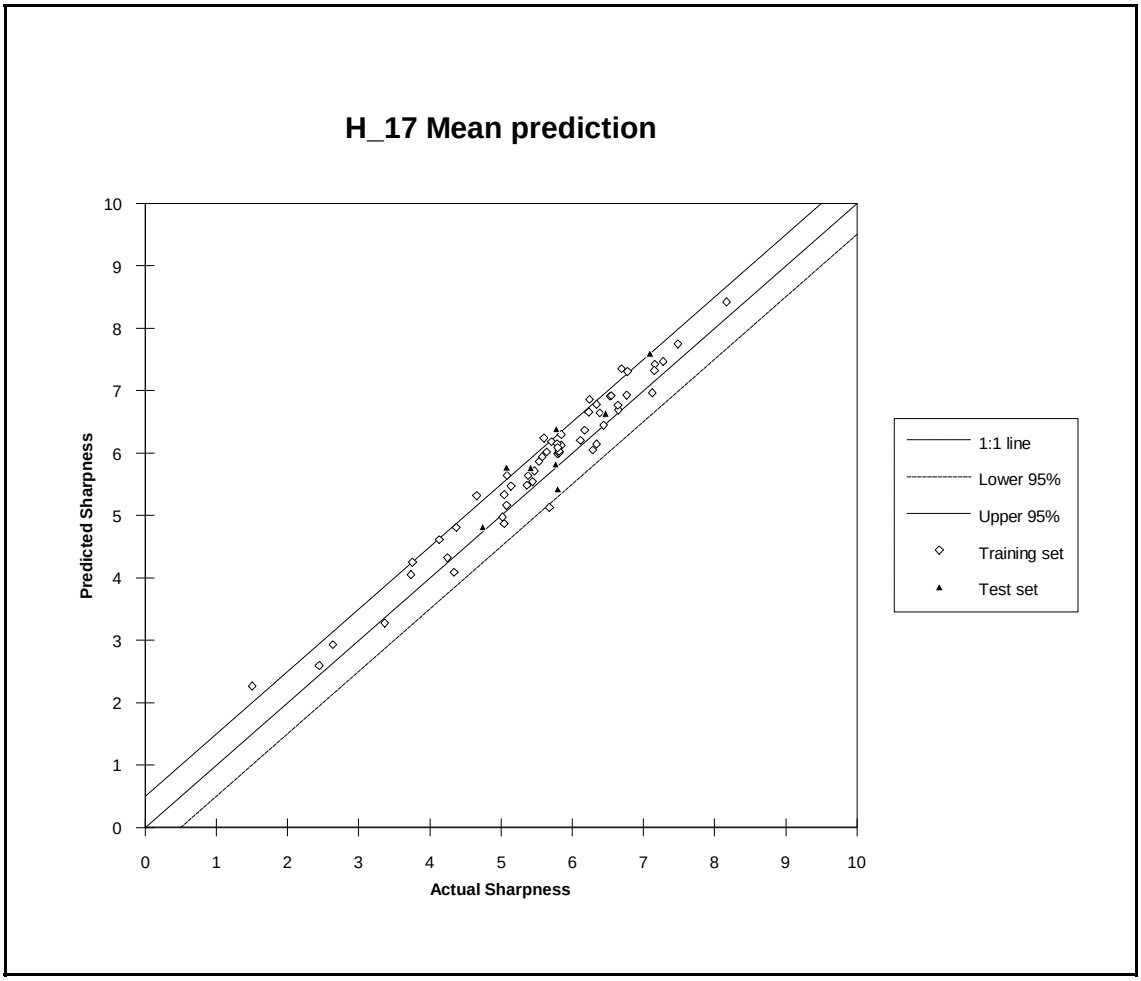
For the Clearness scale, the network (H_16) trained well on the remaining training set after removal of the "class" test set. RMS Error = 0.17 and $R^2 = 0.52$ were better training results than for H_12 (Figure 23). The mean predictions for this model are shown in Figure 27.



27. *Predicted vs. actual mean ratings of Clearness for the Hearing-Impaired group, using one class of stimuli as test stimuli.*

The mean predictions for Clearness are roughly as good for the test set as for the training set, and the spread for both sets is about as large as for the factorial test set in session N_12 (Figure 23).

The Sharpness mean predictions for session N_17 are shown in Figure 28 below.



28. *Predicted vs. actual mean ratings of Sharpness for the Hearing-Impaired group, using one class of stimuli as test stimuli.*

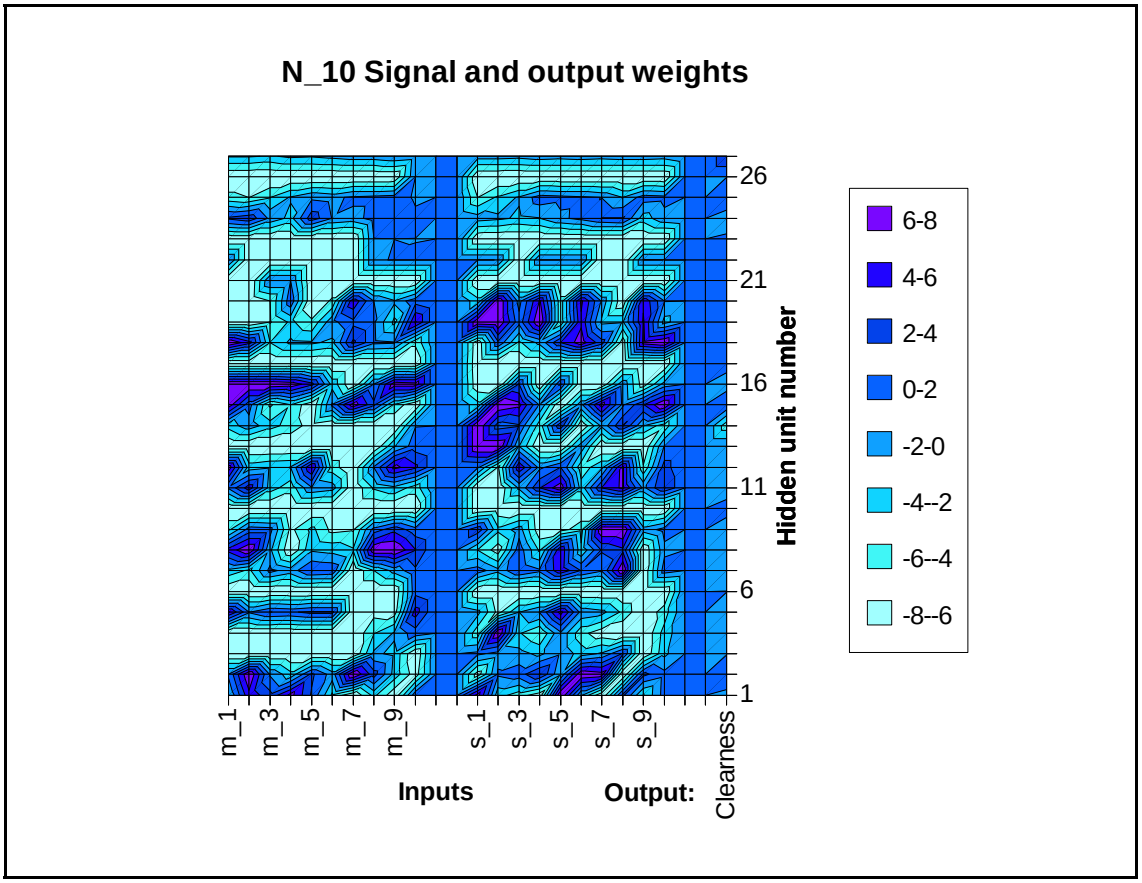
The training statistics for this are: RMS Error = 0.16 and correlation $R^2 = 0.34$, which can be compared to the good predictions on both training and test sets as shown in Figure 24. It should be kept in mind, that the multiple correlation coefficient is a measure of how much of the total variance can be explained by the model, rather than a measure of the correlation between predicted and actual output from the net.

7 Analysis of network weights.

As mentioned previously, the trained networks contain "knowledge" about the underlying input-output relation. This knowledge could potentially be used to infer the mechanism or model that the network has formed, quantitatively or qualitatively. There are examples of such interpretations in the literature, using specialized nets for speech recognition (Waibel & Hampshire, 1989). These are neural applications for classification. Within function approximation, as used in the present report, there have been examples of no clear structure in the network (Sejnowski & Rosenberg, 1987).

Two types of analyses of the trained networks were done to investigate the input-output relation, which tells us something about the mapping from the physical (stimulus) domain to the subjective domain: Plotting of the trained network weights and plotting of neuron activity (i.e. outputs) for selected input stimuli.

The first network to be analyzed was N_10: normal-hearing subjects, binary subject inputs, and factorial pick of test set. The test set performance for this network was very fluctuating as training progressed (Figure 17), and the optimum (2300 runs) was not clearly located at one point. After 2300 runs, 7 hidden units had been added, to a total of 27 units. The network then contained 32 inputs * 27 hidden units = 864 + 1 output * 27 hidden units, totaling 891 weights. In the weight analysis, we were only concerned with the weight connected to the 20 stimulus inputs, i.e. 20 * 27 = 540 + 27 weights = 567 weights. These weights have been visualized in a 3-D contour plot shown in Figure 29.

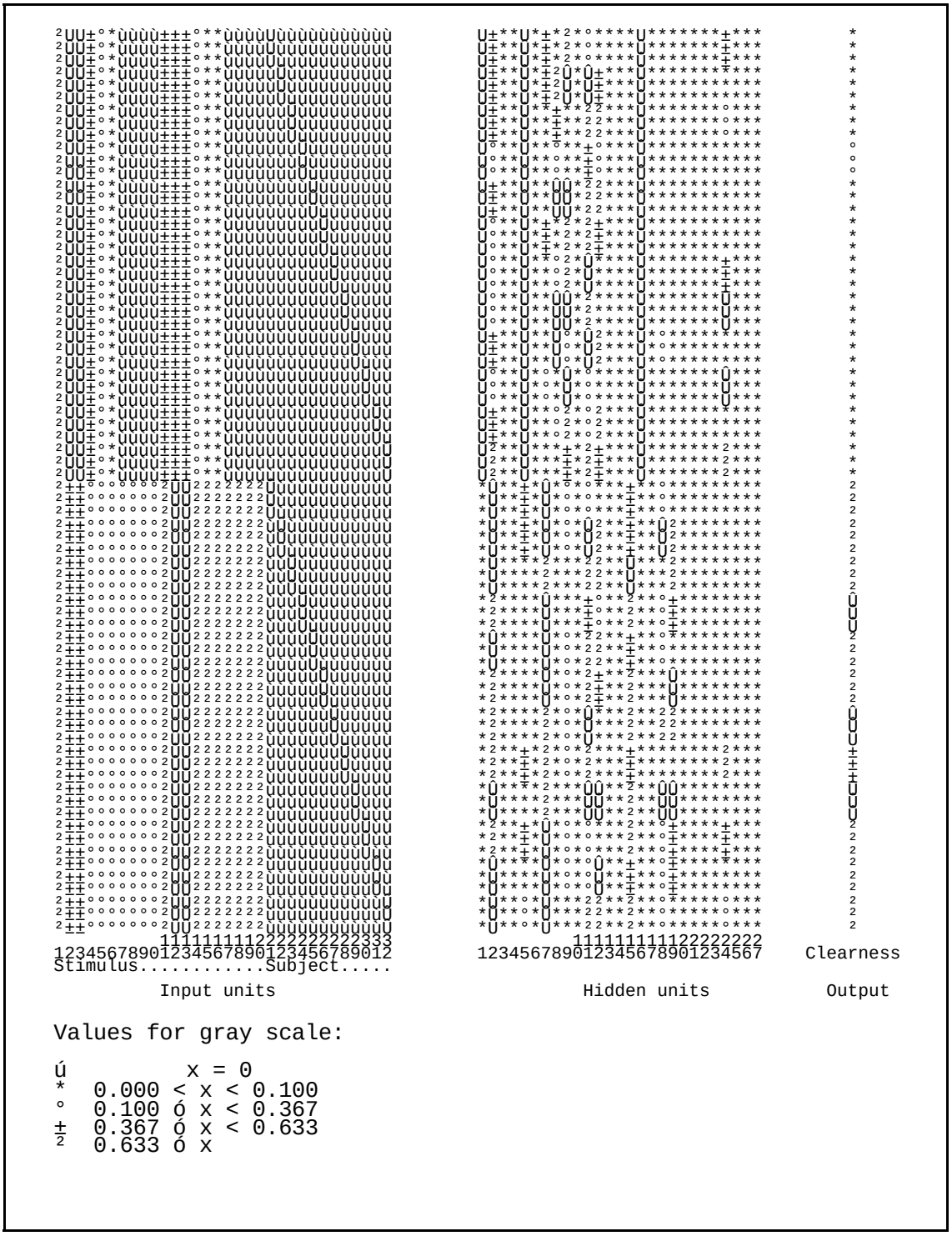


29. *N_10 trained network weights shown in a 3-D contour plot. Along the x-axis are the 20 stimulus inputs plus the output. Along the y-axis are the 27 hidden units. The network weight values are shown by means of a gray-scale.*

The weight values are shown as a gray-scale, covering the range -8 to 8. This is the numerical range that BrainMaker allows for weights, thus they saturate at -8 or 8, which is generally a sign of less than optimal network convergence. In Figure 29, there are horizontal white stripes, i.e. the weights for that particular hidden unit have all saturated at -8. The input values are positive only, thus any small input is multiplied to form a large negative value. A number of these are summed and passed through the sigmoid nonlinearity with an output range from 0 to 1, thus resulting in 0 output (see Figure 1 for example). The hidden unit will never respond and has been trained to be inactive. Many of the units added during training (21 - 27) have not found a function and have been trained into saturation. The plot in Figure 29 has a pattern, but the pattern is too complicated to provide for any simple interpretation. If an assignment of neurons to certain features has taken place, this is random across weights and depends on the initial weight values, which were initialized with small random values before training. A

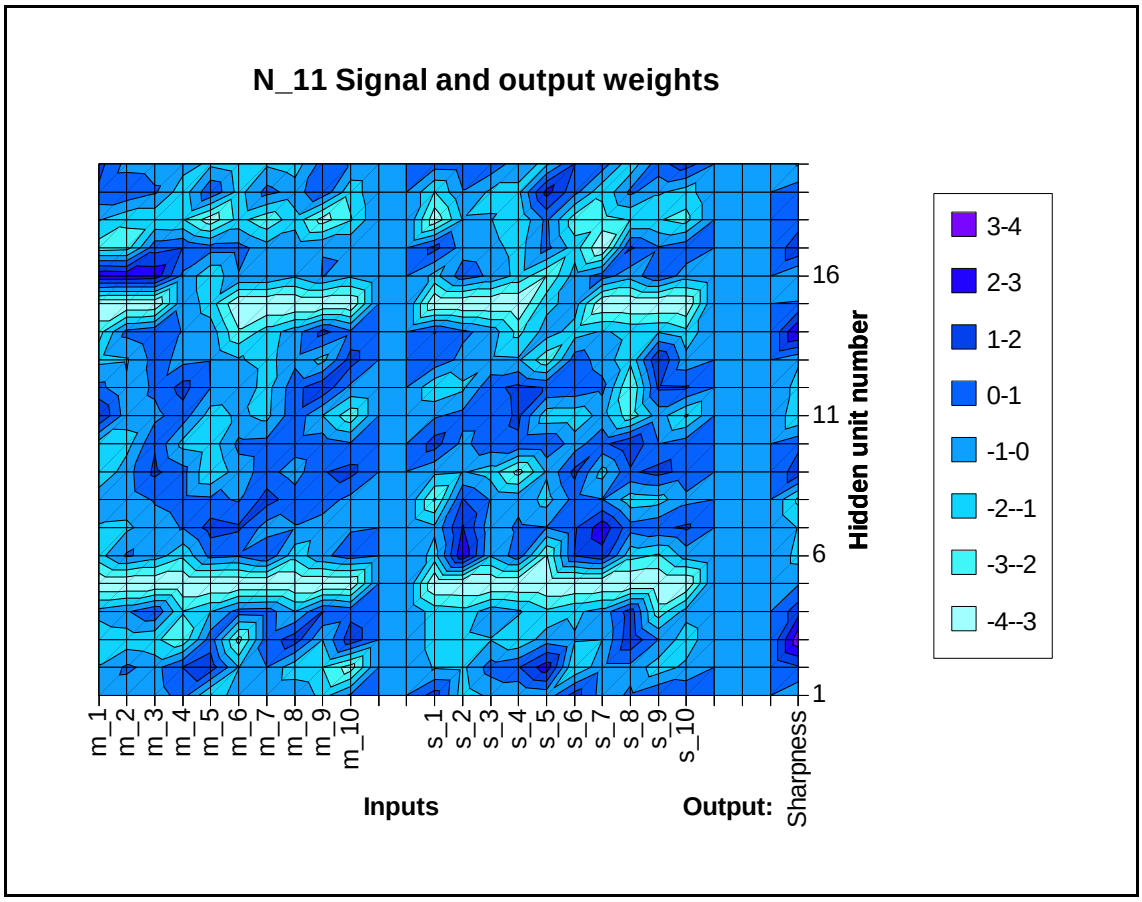
reordering of the hidden units in Figure 29 might provide more information, but sorting after magnitude of the weight to the output layer, did not provide a clearer picture. None of the output weights were saturated, thus they were all active. Another way to investigate the trained network is to visualize the activity within the network as certain stimuli are presented to the network. This has been done for the same network, N_10, and is shown in Figure 30.

The activity patterns show that some hidden units are mostly sensitive to changes in stimulus, by having different patterns for the two stimuli shown here. Some of these hidden units are: 1, 2, 5, 7, 15, 16. Other neurons respond mostly to changing subjects, which is seen as vertical groups of three identical characters - examples are: 11, 12, 24. Some of the remaining units respond to both stimulus and subjects. Finally, some units are never active: 3, 4, 6, 10, 13, 14, 17, 20 - 23 and 25 - 27. These same units have the large negative input weights indicated by the horizontal stripes in Figure 29. Some of these units could of course become active for other input stimuli, but inspection of activity patterns for all 64 stimuli confirmed that these units were in fact always passive.



30. *Diagram of N_10 Network activity for two stimuli: 12 (very unclear) and 61 (very clear). The activity is roughly indicated by a character-based gray-scale as shown in the bottom. Note that the first 20 inputs are for stimulus and the remaining 12 represent the 12 normal-hearing subjects.*

The N_10 network had been trained for a long time, and saturated many weights, thus was far from ideal. Thus, another example was investigated, which converged faster towards optimum test set performance: N_11 reached optimum after 100 training runs, and no hidden neurons had been added at that point. Few weights in this network were saturated, thus the 3-D surface plot was set to cover only the -4 to +4 range, as shown in Figure 31 below.



31. *N_11 trained network weights shown in a 3-D contour plot. Along the x-axis are the 20 stimulus inputs plus the output. Along the y-axis are the 20 hidden units. The network weight values are shown by means of a gray-scale.*

As in Figure 30, we can identify passive units by the horizontal bright stripes, although not as many: 5, 15. A plot of network activity (not shown) confirmed this. Otherwise, there are no clear patterns that can indicate the meaning of individual units.

The examples presented here document that the information is distributed throughout the network, with no clear functionality in individual hidden units. The neural network has

been used as a function approximator, and has accomplished the function fitting by superposition of many non-linear sigmoid functions (Hush & Horne, 1993). A superposition means the sum of many contributions, that are all important to form the result.

The network functionality could also be investigated using idealized network inputs representing various combinations of low- and high-frequency energy with varying shapes in the frequency domain (Mean values: Network inputs 1 - 10) and temporal domain (Standard Deviation values: Network inputs 11 - 20). This option has not been investigated further at present.

8 Discussion.

Section 6 described the training and test set performance, and the prediction performance provided by the neural net in various configurations of input data, subject groups, test set selection etc. These prediction plots indicate surprisingly good results for a quite simple concept of stimulus preprocessing, feature extraction and training data. Best results were obtained by providing the network with a subject input, thus allowing different states in the network for each subject. This provided acceptable results for the factorial (balanced) test set (sessions N_10 through H_13: R^2 = 0.49, 0.48, 0.4, 0.53). For the other type of test set, a selected class of stimuli, the results were as good for Clearness (sessions N_14 and H_16: R^2 = 0.56, 0.52) and poorer for the Sharpness scale (sessions N_15 and H_17: R^2 = 0.35, 0.34).

The prediction performance was only slightly poorer for the hearing-impaired (HI) than for the normal-hearing (NH) group, which is probably due to the slightly poorer reliability and lower sensitivity of the HI group (Nielsen, 1992). This is of course reflected in the subjective rating data that were used for training.

The auditory model developed as part of the overall project, and used here as a pre-processor appeared to provide adequate information for good model prediction performance. A simple scheme for temporal data reduction of the auditory model output was used: calculation of mean and standard deviation across the time axis for each auditory model channel. This seemed to work well, and indicates that temporal effects (temporal integration, post-masking) are not crucial for a good result, since the temporal data reduction used here would ignore the temporal effects to a large extent. Temporal effects are presently not incorporated into the auditory model (Nielsen, 1993a).

The choices made through this work are many and to some extent they just represent the most qualified guesses. There are many other potential ways of matching the models - auditory and neural net - to the subjective ratings. The present results could probably be optimized further.

From a neural net perspective, the models applied here - traditional static multi-layer perceptrons with backpropagation training are very simple compared to some of the sophisticated network topologies, such as recurrent nets, hybrids with Hidden Markov models, etc. As pointed out earlier these models are based on huge research efforts to recognize speech, i.e. to extract the *information* carried by the signal. In the current application we are not concerned with the information, but with the quality of the signal, and most of the current speech models are therefore not relevant here. The theoretical basis and arguments for using more sophisticated models was not present for the work carried out here. As a first step, the present training results look promising.

There is one clear limitation in the evaluation of model performance as presented here. This is the close relationship between training and test data - they originate from the same experiment, meaning the same types of stimuli and the same subjects. The model can only be thoroughly evaluated with new test data from a new experiment with different stimuli and subjects, but the same rating scales. There are theoretical results in the literature concerning the prediction of test performance and generalization based on the training performance, but based on the sensitivity to choice of test set in the present experiments - with very small training and test sets - it did not seem justifiable to apply these models.

When the networks have been trained and evaluated, the next logical step is to examine the established input-output relation further to make at least qualitative statements about the relation between physical parameters (i.e. objective) of the input stimulus and the subjective sound quality rating. Such an examination of the weights may reveal meaningful patterns, but this is not always the case (Sejnowski & Rosenberg, 1987), even when the network provides an acceptable generalization. This was done in section 7 for two of the trained networks, and a simple interpretation of the weight patterns was not possible. The reason is that the network does distributed processing to form the predicted output, and the "knowledge" that the neural net has learned is thus not easy to

deduce. The hope of deducing the relation between physical (signal or auditory) parameters and subjective sound quality parameters was thus not fulfilled.

Another way to view this is to view the neural net approach as a very complex model to solve a prediction problem. Such a complicated model may perform the required prediction well, but will most likely be very difficult to interpret. A simple model (i.e. multiple linear regression) will probably perform worse, but is the easier to interpret.

Other ideas were discussed during this training phase but not tested due to time limitations. Other features could be extracted from the huge amount of data from the auditory model., such as the differential loudness in each channel, representing the transients in the signal. The use of an auditory model to account for some of the known psychoacoustic properties of the impaired ear (frequency selectivity, loudness recruitment) raises the question whether hearing loss affects the perceived quality beyond the psychoacoustic properties. In other words: does equal specific loudness (in frequency and time) for a normal hearing and a hearing impaired person mean equal perceived sound quality? This question might be answered by training and testing on all subjects in the experiment, instead of using the two groups separately.

9 Conclusion.

The neural network model implemented in the present investigation was successful in predicting subjective ratings of sound quality, when used with an auditory model as pre-processor. The output of the auditory model was reduced in both the frequency and time domains to allow for a reasonably small neural network.

Best prediction results were obtained by providing the neural network with a subject input in addition to the auditory model output. This allowed for different states in the network for each subject.

The verification of the network was done with a test set picked from the total data set from the subjective rating experiment. The accuracy of the test set predictions depended on how the test set was picked. Using a mix of stimuli for testing showed prediction errors only slightly larger than the random errors in the subjective rating data itself. Poorer prediction was found, using a specific group of stimuli as test set: clipped speech signals. In this case, the neural network tended to overpredict the sound quality on both of the subjective scales: Clearness and Sharpness. A true verification should be performed using data from a new subjective rating experiment with different stimuli and the same rating procedure.

An analysis of the weights in the trained neural networks showed no simple functional patterns that could be used to deduce the qualitative relation between physical parameters in the sound signal and the perceived sound quality.

10 References.

Bennani, Y., Soulie, F.F. & Gallinari, P. (1990). A connectionist approach for automatic speaker identification. Proc. ICASSP 1990, S1, 265 - 268. Albuquerque, New Mexico.

Conradsen, K. (1984a). En introduktion til statistik, vol 2A (In Danish). The Institute of Mathematical Statistics and Operational Research (IMSOR), Technical University of Denmark.

Conradsen, K. (1984b). En introduktion til statistik, vol 2B (In Danish). The Institute of Mathematical Statistics and Operational Research (IMSOR), Technical University of Denmark.

Cosi, P., Bengio, Y & De Mori, R. (1990). Phonetically-based multi-layered neural networks for vowel classification. Speech Communication 9, 15 - 29.

Gramss, T. & Strube, H.W. (1990). Recognition of isolated words based on psychoacoustics and neurobiology. Speech communication 9, 35 - 40.

Hecht-Nielsen, R. (1990). Neurocomputing. Addison-Wesley, 1990.

Hertz, J., Krogh, A. and Palmer, R.G. (1991). Introduction to the theory of neural computation. Addison-Wesley, Redwood City, CA, 1991.

Hush, D. R. and Horne, B. G. (1993). Progress in Supervised Neural Networks - What's New Since Lippmann?. IEEE ASSP Magazine, January 1993.

Kohonen, T. (1984). Self-organization and associative memory. Springer-Verlag, Berlin.

Kohonen, T. (1988). An introduction to neural computing. Neural Networks 1, 3 -16.

Lawrence, M., Petterson, A. and Lawrence, J. (1992). BrainMaker Professional Users Guide and Reference Manual (1992). 3rd edition, California Scientific Software, April 1992.

Lippmann, R.P. (1987). An introduction to computing with neural nets. IEEE ASSP Magazine, April 1987.

Lippmann, R.P. (1989). Review of Neural Networks for Speech Recognition. Neural Computation 1, 1 - 38.

McClelland, J.L. & Rumelhart, D.E. (1986). Parallel distributed processing, Volume II: Psychological and biological models., MIT press, Cambridge, MA.

McClelland, J.L. & Rumelhart, D.E. (1988). Explorations in parallel distributed processing. MIT Press, Cambridge, MA.

Moody, J.E. (1991). The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems. In: J.E. Moody, S.J. Hanson and R.P. Lippmann (eds.): Advances in Neural Information Processing Systems, vol 4, Morgan Kaufmann Publishers, San Mateo, CA.

Münster-Swendsen, J (1981). Measurements of Kemar Open-Ear-Gain. Internal report no. 9-8-6, Oticon Research Unit, Snekkersten, Denmark.

Nielsen, Lars B. (1992). Subjective evaluation of sound quality for normal-hearing and hearing-impaired listeners. Internal report no. 43-8-1, Oticon Research Unit, Snekkersten, Denmark. Also published as: Technical Report no. 51, The Acoustics Laboratory, Technical University of Denmark, Lyngby, Denmark.

Nielsen, Lars B. (1993a). An Auditory Model with Hearing Loss. Internal report no. 43-8-2, Oticon Research Unit, Snekkersten, Denmark. Also published as: Technical Report no. 52, The Acoustics Laboratory, Technical University of Denmark, Lyngby, Denmark.

Nielsen, Lars B. (1993b). Objective Measures of Sound Quality - Normal-Hearing and Hearing-Impaired Listeners. Internal report no. 43-8-4, Oticon Research Unit, Snekkersten, Denmark. Also published as: Technical Report no. 54, The Acoustics Laboratory, Technical University of Denmark, Lyngby, Denmark. (in preparation)

Rumelhart, D.E. & McClelland, J.L. (1986a). Parallel distributed processing, Volume I: Foundations. MIT press, Cambridge, MA.

Sejnowski, T.J. & Rosenberg, C.R. (1987). Parallel networks that learn to pronounce English text. Complex Systems 1, 145 - 168.

Seneff, S. (1985). Pitch and spectral analysis of speech based on an auditory synchrony model. MIT Technical Report 504, 242 pp.

Sørensen, H.B.D. (1991). A Cepstral Noise Reduction Multi-Layer Neural Network. Proc. ICASSP 1991, S14.14, 933 - 936.

Waibel, A. (1992). Neural Network Approaches for Speech Recognition. In: "Advances in Speech Signal Processing", eds.: Furui, S. and Sondhi, M.M., Marcel-Dekker, New York.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K. & Lang, K.J. (1989). Phoneme recognition using Time-Delay Neural Networks. IEEE Trans ASSP 37(3), 328 - 339.

Waibel, A. & Hampshire, J. (1989). Building blocks for speech. Byte Magazine, August 1989, 235 - 242.

11 Appendices.

11.1 Calibration and stimulus levels.

Procedure: The 64 stimulus files all had the same digital RMS-value: 880, and were referenced to the free field. Four pure-tone files with the same RMS-value (peak amplitude 1245) were created and played back through the play-back setup (see Nielsen (1992) for further details) with the system attenuator set at 40 dB. Sound pressure levels were measured in the KEMAR manikin equipped with the small pinna. The following values were measured.

	500 Hz	1 kHz	2 kHz	4 kHz
file RMS-value	880	880	880	880
mV RMS at phones	88	88	86	80
Measured left (dB SPL)	81.8	81.4	85.4	85.2
Measured right (dB SPL)	80.6	80.9	84.3	85.1
L-R Average (dB SPL)	81.2	81.2	84.9	85.2

Based on average actual attenuator settings, the stimulus levels could be calculated for the two groups. These were referred back to free-field levels by subtracting the free-field gain of KEMAR (Münster-Swendsen, 1981). Subsequently, the 500 Hz value (bold) was used as reference, since the least influence from KEMAR and headphone irregularities was expected here.

Average attenuator (MCL) setting for the NH group: 39.5

	500 Hz	1 kHz	2 kHz	4 kHz
L-R Average	81.7	81.7	85.4	85.7
Free field correction	2.2	3.6	12.8	14.4
Free field SPL	**79.5**	78.1	72.6	71.3

Average attenuator (MCL) setting for the HI group: 18.8

	500 Hz	1 kHz	2 kHz	4 kHz
L-R Average	102.4	102.4	106.1	106.4
Free field correction	2.2	3.6	12.8	14.4
Free field SPL	**100.2**	98.8	93.3	92

11.2 Auditory model parameter files.

```
                AUDITORY MODEL PARAMETERS

                Filename:   nh_anal.aud
                Date:       15.04.93
                Time:       15:00
                Note:       Analyze all NH stimuli with audmod,
                            CSV file output

No. channels:          30
Lower E limit:         3
Upper E limit:         32
Output channel:        0
                       0 for all channels.
Output level:          0
                       0 for end of model.
Input sample rate (Hz):      20000
dB SPL of cal. sinus:  79.5
Peak value of cal. sin:      1245
                       sqr(2)*signal rms value
Recording coupler:     1
                       1: Free field, 2: IEC711/KEMAR, 3: IEC303
Transmission factor:   1
                       1: Zwicker's A0, 2: ELC 100, 3: ELC 100 flat below. 1 kHz
Binaural:              0
                       0: Monaural, else binaural loudness
Output sample rate (Hz): 0
Input frame size:            256
                       Must be power of two and no more than 8192
Overlap:               0
                       0 % overlap
Process:               1
                       0 = all frames, 1 = single frame, n = # frames to average
Output frame size:     100
No. frames to process: 0
                       0 for all frames.
No. zero frames to add:      0
Output format:         12
                       Hypersignal FRQ (10), TXT (11) or CSV (12)
Audiogram (Hz): 125   250   500   750   1000  1500  2000  3000  4000  6000  8000
Audiogram (dB HL):0   0     0     0     0     0     0     0     0     0     0
UCL (dB HL):    120   120   120   120   120   120   120   120   120   120   120
```

32. *Auditory model parameter file for the Normal-hearing subject group used for the processing of the 64 stimuli from the listening experiment. The parameter file uses 0 dB audiogram (per definition) and average signal level for the normal-hearing subject group. See Nielsen (1993a) for further explanation on file format.*

```
                AUDITORY MODEL PARAMETERS

                Filename:   hi_anal.aud
                Date:       27.05.93
                Time:       15:00
                Note:       Analyze all HI stimuli with audmod,
                            CSV file output

No. channels:          30
Lower E limit:         3
Upper E limit:         32
Output channel:        0
                       0 for all channels.
Output level:          0
                       0 for end of model.
Input sample rate (Hz):     20000
dB SPL of cal. sinus:  100.2
Peak value of cal. sin:     1245
                       sqr(2)*signal rms value
Recording coupler:     1
                       1: Free field, 2: IEC711/KEMAR, 3: IEC303
Transmission factor:   1
                       1: Zwicker's A0, 2: ELC 100, 3: ELC 100 flat below. 1 kHz
Binaural:              0
                       0: Monaural, else binaural loudness
Output sample rate (Hz): 0
Input frame size:           256
                       Must be power of two and no more than 8192
Overlap:               0
                       0 % overlap
Process:               1
                       0 = all frames, 1 = single frame, n = # frames to average
Output frame size:     100
No. frames to process: 0
                       0 for all frames.
No. zero frames to add:     0
Output format:         12
                       Hypersignal FRQ (10), TXT (11) or CSV (12)
Audiogram (Hz): 125   250   500   750   1000  1500  2000  3000  4000  6000  8000
Audiogram (dB HL):30.45     35.00 41.36 44.09 49.09 54.09 57.73 61.36 63.64 69.55
76.36
UCL (dB HL):    200   200   200   200   200   200   200   200   200   200   200
```

33. *Auditory model parameter file for the Hearing-impaired subject group used for the processing of the 64 stimuli from the listening experiment. The parameter file uses average audiogram and average signal level for the hearing-impaired subject group. See Nielsen (1993a) for further explanation on file format*

11.3 Example of auditory model output correlation.

This example is for stimulus 44: Music + background noise, compressed in all three frequency bands. This stimulus has relatively little inter-band correlation between the 30 channels.

Chan.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	1														
1	0.92	1													
2	0.64	0.87	1												
3	0.45	0.66	0.8	1											
4	0.21	0.37	0.49	0.89	1										
5	0.04	0.05	0.06	0.37	0.62	1									
6	-0.08	-0.08	-0.06	0.05	0.21	0.59	1								
7	-0.11	-0.11	-0.08	-0.05	0.04	0.23	0.77	1							
8	-0.1	-0.11	-0.09	-0.04	0.01	0.06	0.39	0.8	1						
9	-0.11	-0.11	-0.09	0.01	0.08	0.12	0.1	0.32	0.71	1					
10	-0.1	-0.09	-0.06	0.03	0.1	0.18	0.09	0.11	0.31	0.74	1				
11	-0.09	-0.08	-0.04	-0.04	-0.02	0.03	0.2	0.21	0.17	0.2	0.54	1			
12	-0.05	-0.05	-0.02	-0.05	-0.06	-0.03	0.19	0.25	0.18	0.02	0.1	0.67	1		
13	0.07	0.06	0.03	0.02	0	0	0.07	0.15	0.18	0.06	-0.02	0.16	0.6	1	
14	0.07	0.07	0.04	0.07	0.08	0.08	0	0.03	0.11	0.16	0.04	-0.1	0.02	0.47	1
15	0.06	0.06	0.05	0.08	0.11	0.12	0.03	0	0.04	0.14	0.15	-0.01	-0.05	0.12	0.72
16	0.09	0.11	0.1	0.08	0.06	0.06	0.1	0.09	0.05	0.02	0.08	0.16	0.15	0.15	0.21
17	0.06	0.06	0.06	0.08	0.08	0.07	0.1	0.11	0.11	0.07	0.04	0.1	0.18	0.12	0.14
18	-0.02	-0.03	-0.04	0.03	0.08	0.11	0.07	0.1	0.14	0.18	0.12	-0.01	0.04	0.1	0.14
19	-0.03	-0.03	-0.03	-0.01	0.03	0.06	0.14	0.16	0.17	0.12	0.11	0.06	0.1	0.25	0.11
20	0.04	0.05	0.06	0.07	0.08	0.05	0.12	0.16	0.15	0.09	0.06	0.15	0.17	0.14	0.08
21	0.04	0.06	0.07	0.09	0.1	0.07	0.1	0.12	0.11	0.08	0.09	0.12	0.15	0.07	0.04
22	0.01	0.01	0.02	0.02	0.03	0.02	0.06	0.09	0.1	0.06	0.04	0.12	0.12	0.04	0.03
23	-0.03	-0.02	0	0	0.01	0.01	0.08	0.11	0.1	0.03	0.03	0.13	0.13	0.04	-0.01
24	-0.07	-0.08	-0.06	-0.05	-0.03	-0.02	0.07	0.15	0.14	0.04	0.01	0.12	0.16	0.06	-0.03
25	-0.01	-0.02	-0.03	-0.05	-0.05	-0.03	0.07	0.12	0.09	-0.01	-0.02	0.12	0.18	0.15	-0.03
26	0.05	0.04	0.02	-0.01	-0.03	-0.05	0.04	0.08	0.04	-0.08	-0.08	0.14	0.21	0.13	-0.04
27	0.09	0.08	0.06	0.02	-0.02	-0.06	0.01	0.02	-0.01	-0.11	-0.12	0.08	0.18	0.18	-0.01
28	0.05	0.04	0.03	-0.01	-0.04	-0.08	-0.02	0.01	0	-0.09	-0.12	0.07	0.17	0.18	-0.02
29	0.08	0.06	0.05	0	-0.05	-0.1	-0.05	-0.02	-0.05	-0.14	-0.14	0.09	0.18	0.17	-0.04

Chan.	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
15	1														
16	0.54	1													
17	0.18	0.55	1												
18	0.13	0.15	0.57	1											
19	0.05	0.12	0.19	0.58	1										
20	0.07	0.17	0.21	0.25	0.51	1									
21	0.13	0.18	0.21	0.19	0.25	0.64	1								
22	0.07	0.19	0.19	0.17	0.17	0.31	0.55	1							
23	0.02	0.17	0.22	0.15	0.18	0.29	0.42	0.71	1						
24	-0.01	0.14	0.2	0.18	0.16	0.27	0.37	0.49	0.7	1					
25	-0.03	0.16	0.18	0.15	0.24	0.32	0.31	0.39	0.54	0.74	1				
26	-0.01	0.21	0.22	0.11	0.16	0.34	0.34	0.35	0.46	0.57	0.76	1			
27	-0.03	0.2	0.19	0.11	0.19	0.34	0.33	0.34	0.38	0.42	0.54	0.8	1		
28	-0.03	0.18	0.16	0.09	0.16	0.33	0.34	0.35	0.39	0.44	0.51	0.68	0.87	1	
29	-0.06	0.15	0.14	0.05	0.13	0.29	0.3	0.31	0.36	0.41	0.48	0.62	0.74	0.87	1

11.4 List of stimuli test sets.

See Nielsen (1992) for further details on how to generate this experiment.

Factorial design

2^(8-2) design Resolution V

Alias: G=ABCD
H=CDEF
GH = ABEF

Block: I3=ACF
I4=BDE
I3I4=ABCDEF

Test stim: I5=ACE
I6=BDF
I7=DEF

Stimulus	Signal	Noise	Ch1	Ch2	Ch3
1	Speech	Off	Off	Off	Lin
2	Music	Off	Off	Off	Comp
3	Speech	On	Off	Off	Comp
4	Music	On	Off	Off	Lin
5	Speech	Off	Clip	Off	Off
6	Music	Off	Clip	Off	Clip
7	Speech	On	Clip	Off	Clip
8	Music	On	Clip	Off	Off
9	Speech	Off	Comp	Off	Off
10	Music	Off	Comp	Off	Clip
11	Speech	On	Comp	Off	Clip
12	Music	On	Comp	Off	Off
13	Speech	Off	Lin	Off	Lin
14	Music	Off	Lin	Off	Comp
15	Speech	On	Lin	Off	Comp
16	Music	On	Lin	Off	Lin
17	Speech	Off	Off	Clip	Clip
18	Music	Off	Off	Clip	Off
19	Speech	On	Off	Clip	Off
20	Music	On	Off	Clip	Clip
21	Speech	Off	Clip	Clip	Comp
22	Music	Off	Clip	Clip	Lin
23	Speech	On	Clip	Clip	Lin
24	Music	On	Clip	Clip	Comp
25	Speech	Off	Comp	Clip	Comp
26	Music	Off	Comp	Clip	Lin

27	Speech	On	Comp	Clip	Lin
28	Music	On	Comp	Clip	Comp
29	Speech	Off	Lin	Clip	Clip
30	Music	Off	Lin	Clip	Off
31	Speech	On	Lin	Clip	Off
32	Music	On	Lin	Clip	Clip
33	Speech	Off	Off	Comp	Clip
34	Music	Off	Off	Comp	Off
35	Speech	On	Off	Comp	Off
36	Music	On	Off	Comp	Clip
37	Speech	Off	Clip	Comp	Comp
38	Music	Off	Clip	Comp	Lin
39	Speech	On	Clip	Comp	Lin
40	Music	On	Clip	Comp	Comp
41	Speech	Off	Comp	Comp	Comp
42	Music	Off	Comp	Comp	Lin
43	Speech	On	Comp	Comp	Lin
44	Music	On	Comp	Comp	Comp
45	Speech	Off	Lin	Comp	Clip
46	Music	Off	Lin	Comp	Off
47	Speech	On	Lin	Comp	Off
48	Music	On	Lin	Comp	Clip
49	Speech	Off	Off	Lin	Lin
50	Music	Off	Off	Lin	Comp
51	Speech	On	Off	Lin	Comp
52	Music	On	Off	Lin	Lin
53	Speech	Off	Clip	Lin	Off
54	Music	Off	Clip	Lin	Clip
55	Speech	On	Clip	Lin	Clip
56	Music	On	Clip	Lin	Off
57	Speech	Off	Comp	Lin	Off
58	Music	Off	Comp	Lin	Clip
59	Speech	On	Comp	Lin	Clip
60	Music	On	Comp	Lin	Off
61	Speech	Off	Lin	Lin	Lin
62	Music	Off	Lin	Lin	Comp
63	Speech	On	Lin	Lin	Comp
64	Music	On	Lin	Lin	Lin

Test stimuli (factorial pick):

Stimulus	Signal	Noise	Ch1	Ch2	Ch3
1	Speech	Off	Off	Off	Lin
6	Music	Off	Clip	Off	Clip
28	Music	On	Comp	Clip	Comp
31	Speech	On	Lin	Clip	Off
41	Speech	Off	Comp	Comp	Comp
46	Music	Off	Lin	Comp	Off
52	Music	On	Off	Lin	Lin
55	Speech	On	Clip	Lin	Clip

Test stimuli (class pick):

Stimulus	Signal	Noise	Ch1	Ch2	Ch3
17	Speech	Off	Off	Clip	Clip
19	Speech	On	Off	Clip	Off
21	Speech	Off	Clip	Clip	Comp
23	Speech	On	Clip	Clip	Lin
25	Speech	Off	Comp	Clip	Comp
27	Speech	On	Comp	Clip	Lin
29	Speech	Off	Lin	Clip	Clip
31	Speech	On	Lin	Clip	Off