INFORMATICA, 1996, Vol. 7, No. 4, 495–516

NEW TRENDS IN SPEECH PROCESSING WITH ARTIFICIAL NEURAL NETWORKS

Dalius NAVAKAUSKAS

Department of Radioelectronics, Electronics Faculty, Vilnius Technical University
Aušros Vartų 7a, 2600 Vilnius, Lithuania
E-mail: dn@tuef.vno.osf.lt

Abstract. An analytical review of recent publications in the area of digital speech signal processing is presented. The aim of the given paper is the analysis of these publications, where Artificial Neural Networks (ANNs) were successfully employed. Numerous methods of ANNs employment are discussed due to identify when and why they are reliable alternative to the conventional adaptive signal processing techniques.

Key words: digital speech signal processing, artificial neural networks, multi-layer perceptrons, self-organizing feature map, vector quantization.

1. Introduction. Most important in digital processing of speech signal or simply speech processing (SP) is the fact that it requires *extremely sophisticated DSP algorithms* to perform *complex transformations* (Hunt, 1992). In many cases, required SP tasks are so complex (they must encounter considerable speech variability introduced by, e.g., different speakers, their intonations, accents, acoustic features of room, transmission channel characteristics, etc.), that it is difficult to characterize them (Jekosch, 1993) and formalize their transformations into closed form equations or stable algorithms. Alternative to adaptive signal processing techniques dedicated to overcome these problems, could be learning systems, i.e., systems where Artificial Neural Networks (ANNs) are employed (Simpson, 1990; Mogensen, 1993).

In this paper *we focus on recent (1990–1994) publications dealing with SP applications where ANNs were used. Our aim is to outline and discuss recently discovered numerous methods how ANNs could be used in SP applications, due to identify in which situations and why ANNs are reliable alternative to the conventional adaptive signal processing techniques.*

Material in this paper is organized in a such way. *On the very begin-*

ning, non-experienced reader we acquaint with two mostly used ANN models: multi-layer perceptron and self-organizing feature map. *Then*, we present main discussion about method of ANNs' employment in SP applications. While doing this we outline what task ANNs were employed to do, how they were used and what results were achieved. *Finally*, discussion why ANNs are reliable alternative to the traditional adaptive signal processing techniques in SP is presented.

2. Background of mostly used ANNs. Looking more closely to practical applications in SP area becomes evident, that *in perhaps 80 to 90 percents of all cases, multi-layer perceptrons and self-organizing feature maps are used* (Klimasauskas, *et al.* 1989). Naturally, good understanding of principles how these ANNs work is of primary importance for further discussion. Thus, in this section non-experienced reader will find short introduction of each ANN, where description of network's architecture, learning and recall procedures are given without strict mathematical formalization.

2.1. Multi-layer perceptron. The Elementary multi-layer perceptron (MLP) – introduced by Rosenblatt (1962) – is a three layer ANN with feedforward connections from the F_A processing elements to the F_B processing elements and feedforward connections from the F_B to the F_C processing elements (Fig. 2.1).

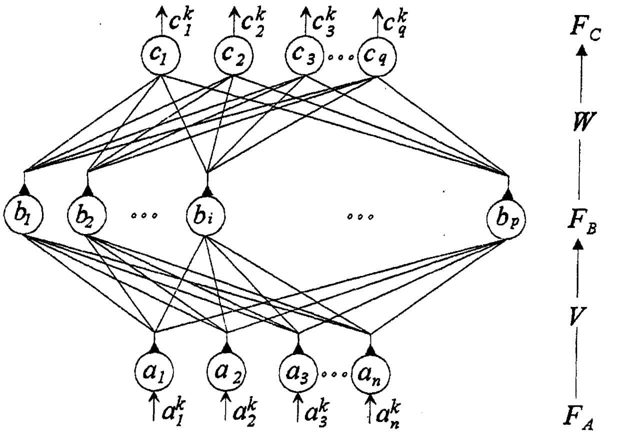

Fig. 2.1. Structure of multi-layer perceptron.

In general it is possible to have several hidden layers (layers between input and output), connections that skip over layers, recurrent connections (connections that loop and connect back to the same processing element), and lateral connections (connections between processing elements in the same layer). Although these advanced topologies are important, they tend to obfuscate the simplicity of MLP. Hence, we will further concentrate only on elementary MLP.

The components of the input vector of MLP may take binary or continuous values. Training is performed in supervised way, that is why, network output has to be specified.

Multi-layer architectures were described at on early date, but they became popular when a learning algorithm was described to train multi-layer feedforward networks (Denning, 1992). This algorithm, the back-propagation learning rule, introduced by Rumelhart, *et al.* (1986), is in fact a generalization of the Widrow–Hoff rule.

The application of the back-propagation rule involves two phases. During the first phase the input is presented and propagated forward through the network to compute the output value c_j^k (the jth element of the actual output pattern produced by the presentation of input pattern k) for each unit. This output is then compared with the targets, resulting in an error signal δ_j^k for each output unit. The second phase involves a backward pass through the network (analogous to the initial forward pass) during which the error signal is passed to each unit in the network and the appropriate weight changes are made. This second, backward pass allows the recursive computation of δ. The first step is to compute δ for each of the output units. This is simply the difference between the actual and desired output values times the derivative of the squashing function (also called as threshold function – is function that map processing elements input to the prespecified range – the output. Examples of these functions could be linear, ramp, step or sigmoid functions). We can then compute weight changes for all connections that feed into the final layer. After this is done, then compute δ's for all units in the penultimate layer. This propagates the errors back one layer, and the same process can be repeated for every layer. The backward pass has the same computational complexity as the forward pass, and so it is not unduly expensive.

The MLP strengths include its ability to store many more patterns than the number of F_A dimensions ($m >> n$) and its ability to acquire arbitrarily complex non-linear mappings. For MLP limitations: its extremely long training time, its offline encoding (training) requirement and the inability to know how to precisely generate any arbitrary mapping procedure, many improvement techniques (e.g., momentum method, adaptive learning rule) were proposed and made MLP training procedure – back-propagation more reliable and faster.

2.2. Self-organizing feature map. The self-organizing feature map (SFM) – introduced by Kohonen (1984) – and also referred as Kohonen network formally consists of two layers of units: an input layer F_A and an output layer F_B (Fig. 2.2). The array of input units operates simply as a flow-through layer for the input vectors and has no further significance. Often this layer is left out, as is depicted in Fig. 2.2. The units in the output layer are ordered in a low-dimensional framework of units, e.g., a one-dimensional array or a two-dimensional matrix. Usually, one- or two-dimensional networks are used, and henceforth, only these two types of SFMs will be considered. Each unit in the network is fed by the input layer and is equipped with a single weight vector. The dimension of the weight vectors is equal to the number of components of the input vectors. No signals are transferred between units in the output layer.

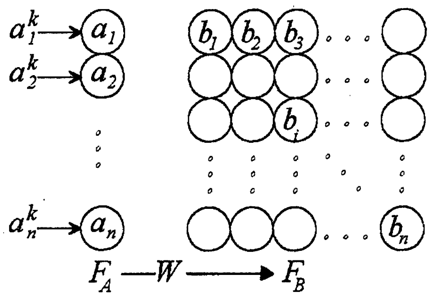

Fig. 2.2. A two-dimensional SFM.

As in multi-layer feedforward networks, the components of the input vectors or patterns may take binary or continuous values. Often the input vectors are scaled in some way, e.g., vector normalization, before these are used for training the network. Because training is performed in an unsupervised way, no network output has to be specified.

Since the introduction of the SFM, several training strategies have been proposed which deal with different aspects of use of the SFM. We will restrict ourselves to the ANN which has been proposed by Kohonen. The SFM learning algorithm can be subdivided into five clearly distinguishable stages which are presented briefly below.

Before the training process is initiated, the weight vectors of the units in the output layer need some preparation. Therefore, for each unit j in the network, random values are assigned to the elements of its associated weight vector w_j. Then an input vector A^k is drawn randomly from the training set. For each unit in the output layer a predefined similarity or distance measure $D(w_j, A^k)$, which operates on the unit's weight vector and the input vector, is determined. The unit in the SFM possessing the most extreme value of $D(w_j, A^k)$, i.e., its weight vector is most similar (the similarity measure $D(w_j, A^k)$ is at a maximum) or close (the distance measure $D(w_j, A^k)$ is at a minimum) to the input vector, is declared as the winner. In addition, all units in the close vicinity of the best matching unit are selected too. Finally, the weight of the winning unit and its neighbours are modified according to:

$$w_j(t+1) = w_j(t) + \eta(t)N(t,r)[A^k - w_j(t)], \tag{2.1}$$

where t and $\eta(t)$ denote the iteration number and the learning rate, respectively. The $N(t,r)$ is a neighbourhood function (as neighbourhood function Kohonen used a block function, however, a triangular, Gaussian Bell-Shaped or Mexican-Hat functions could be used too) in which r indicates the distance in the SFM between the unit j which has to be updated and the winning unit. Note that both the Neighbourhood Function and the learning rate explicitly depend on the iteration number. This completes the processing of one input vector. Next, a new input vector is drawn randomly from the training set and the learning process continues. When all patterns of the training set are presented to the network, one full processing cycle or iteration has been completed. Finally, the learning process stops after a predefined number of processing cycles.

As a result, the weight vectors of the winning unit and its neighbours are gradually moved along the straight line which connects w_j and A^k towards the applied input vector (Fig. 2.3). By this mechanism, input vectors which possess similar features will be mapped onto the same region in the SFM. However, due to the unsupervised character of the learning algorithm, one cannot control

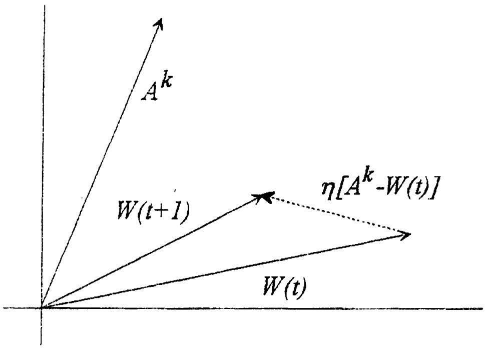

Fig. 2.3. Adaptation of SFM's weight vectors.

which unit or cluster of units in the map will be associated with a specific input pattern or class of input patterns.

The SFM is *a topology-preserving mapping technique.* The Kohonen mapping techniques is primarily used for the examination of data sets for which no or only a little a priori knowledge concerning the internal structure is available. Once the network has been trained, each unit in the SFM might be associated with an object class and then the map may be used for classification purposes.

3. ANNs employment into SP applications. Following our aim to discuss how ANNs are employed into SP applications, we divide this section into several parts, while each of them dedicating to specific *selected* SP application area. Selection of particular SP application area was determined by desire to discuss *main* and *mostly utilized* areas. We realize, that each of discussed SP applications areas is *some how dependent* from the others and even could be merged with others into more complex SP applications (e.g., speech recognition, speech classification, etc.). If this duality introduce new aspects of ANNs' use, we also try to mention it.

3.1. Speech recognition. It is well known that speech recognition problem could be solved in a two different ways. First of them is based on *speech signal features extraction,* while the other – on *speech signal segmentation and classification.* Usually final – recognition task is performed matching outlined speech signal features or classes with templates, utilizing traditional dynamic programming technique. Both approaches will be more thoroughly discussed in the separate chapters below. Hence here we will concentrate on whole speech recognition system implementation with ANNs.

Buniet *et al.* (1993) proved the fact that *entire system for speech recognition could be designed using only ANNs*. Their system consists of three parts: vowel nuclei detection, vowel identification and word identification (Fig. 3.1). In all of these parts authors employed selectively trained MLPs, due to perform small vocabulary connected word recognition task. Preliminary results showed good ANNs' performance in the two first parts of the system. In later chapters we will more detaily discuss ANNs' use in mentioned sub-systems.

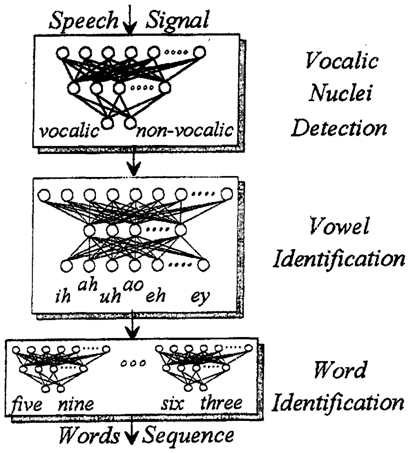

Fig. 3.1. Structure of system for speech recognition entire based on only ANNs (from Buniet *et al.*, 1993).

Another global approach of speech recognition task is a *direct mapping* the acoustic domain into a (large) semantic space. Castano *et al.* (1993) proposed for this purpose to use a simple recurrent network of small size. Experiment performed by authors involved real-voice input. The simple recurrent network adopted to solve required task consisted of 16 input units, one for each of the 16 possible microphonetic labels, and 18 outputs corresponding to a local representation of the 18 possible semantic units (spoken numbers in one million range) of the semantic language categories. Several number of hidden units were tried. Experiment results indicated that the underlying acoustic – syntactic – semantic mapping was actually captured by the learning network.

3.2. Speech signal feature extraction. Speech signal feature extraction procedures usually are used as a part of more complex systems such as speech recognition system, speech coding system, etc. If one considers speech signal feature extraction problem as *nonlinear mapping* from input – speech signal space, which is of high dimension, to output – features space (smaller dimension), then ANNs' employment into speech signal feature extraction applications becomes natural and evident.

Together with employment of *classical ANNs* such as MLP and SFM (for an example of SFM use, see Section 3.3), some authors made attempts to design *specific for this purpose ANNs*. The time-delay neural network developed by Hinton *et al.* (1990) would be as a typical example. The time-delay neural network could be thought as a usual MLP with delay elements included among processing elements in input and hidden layers. Such architecture of ANN while acting enables to consider not only the present features but also the history of these features. Thus time-delay neural network could perform required mapping with time varying signals. Major drawbacks of time-delay neural network trained with the back-propagation learning procedure are slow speed of convergence to the global minimum and occasional instabilities. Besides the fact, that Bappert and Jobst (1993) proposed the method of solving stiff differential equations to ensure the convergence of time-delay neural network without adjustment of the learning rate, additional research must be done to ensure more quick time-delay neural network's convergence speed in order to effectively employ it into speech signal feature extraction problems.

Perspective methods of ANNs' use in speech signal feature extraction application which outperforms usual linear prediction method, was proposed by Kwong and Gang (1993). They introduced a single layer neural network based nonlinear autoregression model (Fig. 3.2). In this case the identification of the model parameters a_i is equivalent to the training of the single layer neural network. Authors proposed fast learning algorithm for single layer neural network utilizing the modified Newton method for the identification of the nonlinear parameters. Experimental results employing proposed model in speech recognition of 10 chinese digits task showed 98,7% of the recognition rate.

3.3. Speech signal labelling, segmentation and classification. Because of inherited classification properties of ANNs, they are widely used for speech recognition tasks where speech signal labelling, segmentation or classification

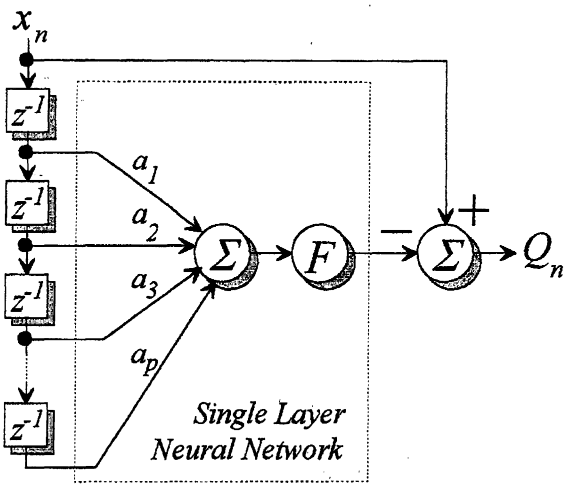

Fig. 3.2. The single layer neural network based nonlinear auto regression model (from Kwong and Gang, 1993).

is performed. All these actions solely or linked together usually form the first step in the speech recognition process, mainly to reduce the order of the selection set.

One of the most recent presentation of ANN as speech signal labeller was made by Cerf and Compernolle (1993), where hybrid – consisting of ANNs and hidden Markov models (HMMs) – system for speech recognition task was introduced. Authors used four MLPs (Fig. 3.3) trained with basic, first and second derivative and energy related parameters respectively due to classify phonemes. Then the label, corresponding to the highest output of each MLP was used by the HMM. Final segmentation of the database was achieved using a well known Viterbi alignment. Experimental results showed that MLPs labelling technique is more efficient than both unsupervised and supervised Euclidean labelling.

Speech signal segmentation is usually followed by classification of input speech into acoustic classes. One of perspective ANN use for this task is clearly identified in the work of Hernaez *et al.* (1993). Here authors used SFM in an acoustic segmentation task and MLP in smoothing the results and finally classifying them (Fig. 3.4). Presented approach enabled to overcome

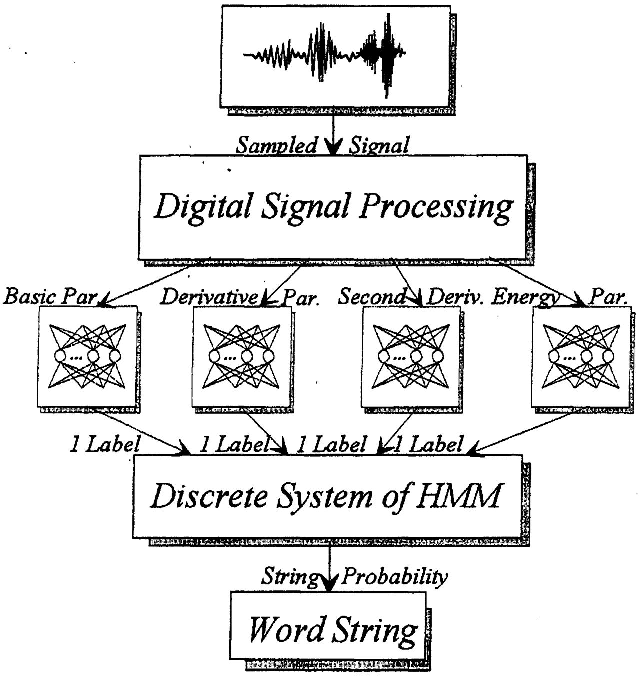

Fig. 3.3. Structure of system for speech recognition based on multi-MLPs labelling for HMMs (from Cerf and Compernolle, 1993).

most problems in this field arousing because of coarticulation between adjacent phonemes. Main idea was based on the fact, that tranzition frames are fundamental to the correct perception of the corresponding phoneme or syllable. Tracking the trajectories followed by input speech frames in the SFM enabled to extract much of this information. For this purpose, MLP with two analogue inputs, corresponding to the $x - y$ coordinates of the winner cell, was trained in a way to learn different trajectories associated with transitions.

For the speech classification problem mostly suited, with excellent classification properties is MLP (example of it's use was presented before). However, a lots of research was done also to create an original ANN for classification purposes. As a typical example – hidden control neural network (Sorensen and Hartmann, 1991) could be. Main problem of ANN's use in classification problem is the optimality of ANN's structure. Thus, a near optimal structure ANNs

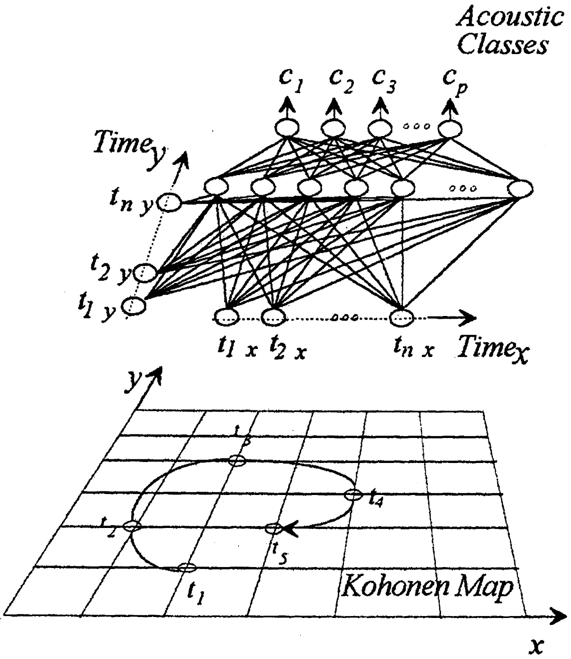

Fig. 3.4. Connection of SFM and MLP dedicated to speech signal segmentation and classification (from Hernaez *et al.*, 1993).

were tried to design. Sorensen and Hartmann (1992, 1993, 1994) introduced self-structuring hidden control neural network possessing required optimality.

3.4. Speech signal detection and prediction. Speech signal detection is distinctly different from the speaker identification in a way, that it requires to identify speech or non-speech signals, but not global speech signal properties. It means, that speech signal detection task is less complicated and in most of the cases could be realized as *simple classification network.* Cryzewski (1994) reports about MLP's use as classifier to perform speech signal detection task in restoration of old records. MLP is trained engaging examples from both – desired signals and disturbing impulses. Single neuron in the output layer has two stages (+1; –1) enabling to classify signals into distorted and non-distorted speech classes.

Solving pitch period detection problem, which also could belong to speech signal detection task, Denzler *et al.* (1993) use different approach. They employ MLP to perform *inverse filtering* of the speech signal. In this case, speech signal is directly mapped to the voice source signal. Authors present one frame of the speech signal to the input layer of MLP and then get one single output value at the output layer. This value is interpreted as one signal point of the voice source signal. By shifting the speech signal point-by point through the input layer, they get the complete voice source signal by concatenating single output values to one signal. Pitch periods are determined from voice source signal more accurately even if laryngealization is present.

Speech signal prediction techniques are widely used when speech signal is corrupted by noise and it's recovery or speech recognition task is required. For these cases, special linked predictive neural networks were introduced by Tebelskis and Waibel. These networks predictively estimate an input sequential signal by mapping present and past spectral frames into predicted spectral frames via a prediction network. However, usual MLP also could be applied here. Cryzewski (1994) used them to restore old records. In this case, MLP was trained on non-distorted speech signal sets. In a lost packets case, MLP performed *forward and backward non-linear prediction.* As a quite accurate result, superposition of resulted signals from MLP was used.

3.5. Speaker normalization and identification. Normalization techniques are widely used in SP tasks, due to exclude variability of speech signal introduced by two main sources: speaker variability and variations in acoustical channel (room) conditions. Two basic approaches to the problem of speaker normalization are: *a nonlinear functional mapping* of a test speaker's feature space to the one of the reference speaker, or *a codebook mapping*, i.e., finding of reliable correspondences between the entries of a test-speaker's codebook and the one of the reference speaker.

According to *first* approach, Mokbel *et al.* (1993) employed MLP trying to reduce speech variability introduced by the channel effects. They trained MLP to estimate the Gaussian mean of acoustical vector's HMM. For this purpose, speech acoustical vectors were aligned on the basis HMM (Fig. 3.5a). As input of MLP the concatenation of the acoustical vector and the channel model was used. Once the MLP was trained to normalize the speech data, it served as a preprocessing module preceding the HMM to be trained with the preprocessed data (Fig. 3.5b). Preliminary experiments gave encouraging results even if the improvements obtained were not as significant as those obtained by cepstral subtraction.

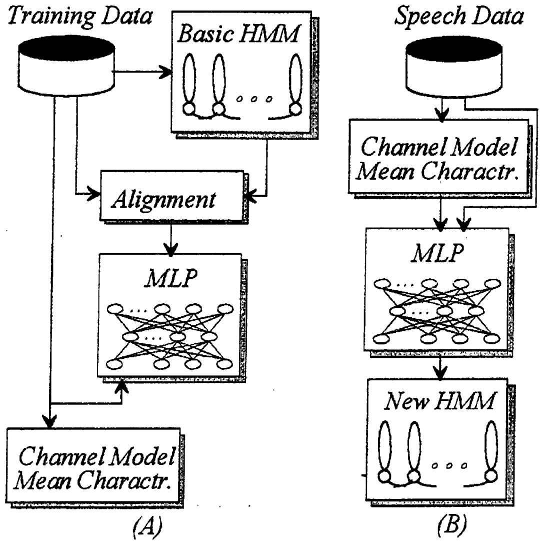

Fig. 3.5. Scheme of the use of an MLP for speaker normalization task by nonlinear functional mapping (from Mokbel *et al.*, 1993).

The *second* approach was explored by Knohl and Rinscheid (1993), which developed speaker normalization method based on mapping of two SFM. Their normalization system (Fig. 3.6.) consists of a reference map (40x25 nodes) that is trained on the reference speaker's map (40x25 nodes), generated by a special topology – maintaining retraining of the reference map. The retraining procedure ensures the topological identity of both maps and thereby implicitly establishes an 1:1 correspondence of the codebooks. This allows for an 1:1 – exchange of the feature vectors represented by the neurons of the reference map for those of the test map in the operation phase. Proposed method works equally well on any abstract feature space apart from the commonly used FFT-spectra, e.g., on neurograms or on even-oriented feature vectors.

Very close problem to speaker normalization is a speaker identification problem. Here, a degree of similarity among speakers must be measured. H.-Mendez and F.-Vidal (1993) investigated three ANN's architectures: MLP, SFM and radial basis function, due to perform this task. All ANNs were employed to perform a non-linear principal component analysis of acoustic vectors. Experimental results show that SFM gives the best identification error rates due to

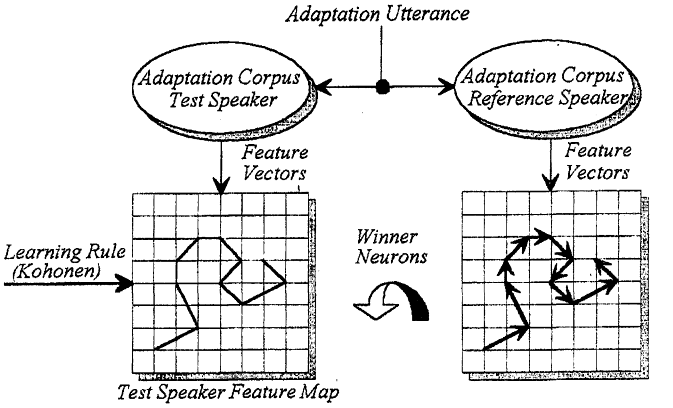

Fig. 3.6. Scheme of the use of an SFM for speaker normalization task by a codebook mapping (from Knohl and Rinscheid, 1993).

employed Learning Vector Quantization algorithm, which reduces the number of misclassificated vectors.

3.6. Noise reduction in speech signal. Noise reduction in speech signal area could be thought as original application area (e.g., noise reduction in old records, noise reduction of channel parameters fluctuations, etc.) or as sub-area (sub-system) in already discussed (Section 3.1–3.5) SP applications (e.g., noise reduction in speech recognition, speaker identification, etc.).

This duality of noise reduction application area dictates two different strategies of ANNs' employment: use ANNs as filters *operating in time or frequency domain or operating in speech feature vectors domain* respectively.

First – straightforward way of ANNs' employment in noise reduction is to design or train them so, that ANNs will perform casual adaptive filters' functions. Such ANNs are called adaptive neural filters.

For operating in the time domain as adaptive neural filters usually are employed single- or multi- layer perceptrons trained with modified back-propagation algorithm (Hoyt and Wechsler, 1990). However, other approaches of such filters' design are also explored. For example Wan (1990) proposed as adaptive neural filters use a network structure which models each synapse by a finite impulse response linear filter. Such network could be thought as a complex nonlinear filter. Yin *et al.* (1991) showed that adaptive neural filters could successfully approximate both linear Finite Impulse Response filters and Weighted Order Statistic filters (median, rank order and weighted median), and even provide better performance then usual adaptive filters (Hoyt and Wechsler, 1990).

Typical example of adaptive neural filters operating in the frequency domain could be found in work of Gao and Haton (1993) or Tsoukalas *et al.* (1993) (for latest see Fig. 3.7). The main idea of their work was to use ANN (MLP) for noise removal from a noisy speech power spectrum by forcing it's output to have the same auditory masking threshold as the clean speech signal. An approximation of the auditory masking threshold of the clean speech signal was calculated using an estimate of the clean signal, taken by a modification of the power spectral subtraction method. Proposed noise reduction method is very promising since employment of ANN allowed a better estimation of the power spectrum of the speech signal than the estimation obtained from the power spectral subtraction method.

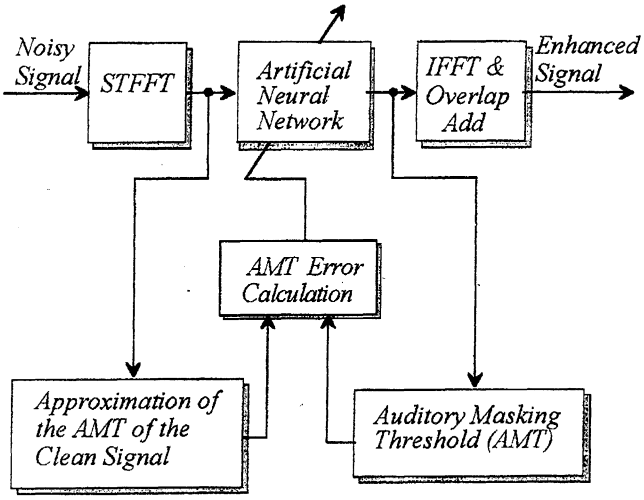
Noisy Signal
STFFT
Artificial Neural Network
IFFT & Overlap Add
Enhanced Signal
AMT Error Calculation
Approximation of the AMT of the Clean Signal
Auditory Masking Threshold (AMT)

Fig. 3.7. Block diagram of ANN's employment as adaptive neural filters operating in the frequency domain (from Tsoukalas *et al.*, 1993).

Second approach of ANN's employment in noise reduction application is to use ANN for noise removal in speech feature vector domain. Mostly in this approach as feature vectors are used linear prediction coding (LPC) cepstral coefficients. Trompf *et al.* (1993) additionally proposed several preprocessing of LPC cepstral coefficients steps: derivative calculation, principal components analysis and principal components analysis variance-sorting (Fig. 3.8.), due to enhance the robustness of the basic feature set. Authors report, that employment of ANN for noise reduction and special training enable to encode in ANN not only noise reduction task but also all required preprocessing steps. Thus, all necessary calculations were held in one step by ANN.

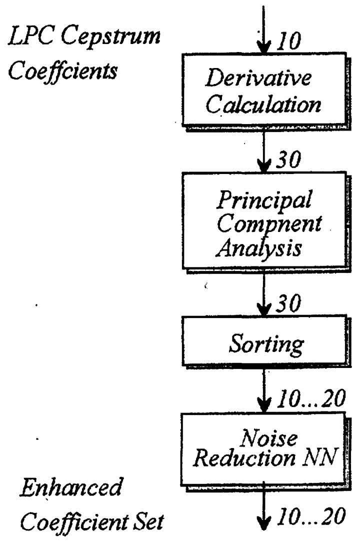

Fig. 3.8. Block diagram of ANN's employment for noise reduction in the speech features vectors domain (from Trompf *et al.*, 1993).

3.7. Speech coding. Let's consider an 2400 bps code-excited LPC (CELP) coder. Such coder generates the speech signal calculating a linear filtered excitation. The filter is usually represented by a set of LPC parameters. Important progress has been made recently including vector quantization techniques for reducing the bit rate for encoding the LPC parameters without introducing any degradation in the speech quality. However, *computational complexity* is one of the major drawbacks of the traditional vector quantization and even improved dynamic vector quantization (Tadj and Poirier, 1993) representation, as they use an unstructured codebook. One of perspective ways to improve this is to use *a suboptimal structure codebook.*

On the other hand the SFM can be seen as an "online" induced topological vector quantization scheme. They quantizate a vector in a generally higher dimensional input space onto a vector in an output space. The optimal output space dimension is considered when neighbourhood relations from input space to output space are preserved, so close input vectors are quantizated as close points in the output space. Thus, SFM leads to an encoding that is a compromise between minimization of the Vector Quantization distortions for the original data and preservation of their spatial ordering.

L.-Gonzalo and H.-Gomez (1993) proposed to use SFM to obtain ordered codebook. Authors designed two maps of 64x64 nodes (the first with the four first LPC's and the second with the last six LPC's) to obtain transparent vector quantization with 24 bits/frame. Proposed method produced ordered vector quantization as good as classical methods for vector quantization training.

In this new approach with SFM, contrary to the traditional ones, neighbourhood relations are preserved in output space. This enabled to follow trajectories in the low-dimentional (output) map and improve LPC speech coder in two different ways. First of all, performance of search procedures was increased, due to employment of fast nearest-neighbour search algorithm. Then, more efficient encoding of LPC envelope during steady-states in map was proposed. It was based on the fact, that LPC trajectories of certain stationary sounds are concentrated in regions in the SFM. In case both bands are described as trajectories concentrated in one cluster of map, a 50% of the bits can be saved.

4. Discussion and final remarks. Now it is clear in which SP tasks and how ANNs were used. Becomes evident also the fact that ANNs are potential to perform arbitrary nonlinear time independent or dependent processing. However, the question – why ANNs are so potential and even in some cases outperform conventional adaptive signal processing techniques, is not clear yet. Thus here we will try to answer it.

Analyzing ANN's employment methods one could notice that all of them can be generalized into three main mapping categories:

a) *feature-to-symbol mapping* (direct mapping presented in Sec. 3.1, speaker classification – Sec. 3.3);

b) *feature-to-feature mapping* (feature extraction – Sec. 3.2, speaker generalization and identification – Sec. 3.5, speech segmentation and labelling – Sec. 3.3;

c) *signal-to-signal mapping* (speech detection and prediction – Sec. 3.4, filtering in time/frequency domain – Sec. 3.6).

Common feature of these mappings is that usually high-to-low dimension space mapping is required (due to reduce variability of signal and/or variable set). In a such process, main interest is not to precisely perform mapping (in general it is not possible), but to extract all main features (structure) of high dimension signal. Inherited ANNs' structures are excellent for required *dimensionality reduction.* ANNs are able not only to represent a variety of statistical properties of data distributions, but also are capable of constructing combinations of features which characterize *higher-order moments* in the data distributions.

Unsupervised ANNs' learning ability lets to perform dimensionality reduction (feature extraction) in *an automatic manner.* It enables a possibility to use ANN as preprocessing modules for speech signal processing and also as modules for speech signal properties examination (Morgan and Scofield, 1994).

During recall process ANNs are acting as associative memories, despite they were trained in unsupervised (autoassociative memories) or supervised (heteroassociative memories) way. It means, that ANNs *respond to not known patterns* in the nearest-neighbor or interpolative fashion (this property of ANN is called generalization). This ANNs' property makes them very useful for SP tasks in noisy environment.

Finally, in that cases when ANNs could not outperform conventional adaptive signal processing techniques in the functional sense, they do outperform them in the sense of computation speed. It is the fact of high connectivity of ANNs' processing elements, which leads to *parallel (distributed) processing order.*

Whereas ANNs' employment in SP applications is emerging and despite ANNs' drawbacks – sensitivity to both the amount and type of training data (scaling problem), becomes evident that in SP application area ANNs found solid background.

REFERENCES

Bappert, V., and M. Jobst (1993). Training of a time-delay neural network for speech recognition by solving stiff differential equations. *Eurospeech'93: Proceedings,* Vol. 2. pp. 1492–1496.

Buniet, L. *et al.* (1993). Selectively trained neural networks for connected word recognition in noisy environments. *Eurospeech'93: Proceedings,* Vol. 2. pp. 841–844.

Castano, M.A., E. Vidal and F. Casacuberta (1993). Learning direct acoustic-to-semantic mappings through simple recurrent networks. *Eurospeech'93: Proceedings,* Vol. 2. pp. 1017–1020.

Cerf, P., and D.V. Compernolle (1993). Speaker independent small vocabulary speech recognition using MLPs for phonetic labelling. *Eurospeech'93: Proceedings,* Vol. 1. pp. 143–146.

Czyzewski, A (1994). Artificial intelligence-based processing of old audio recordings. *An Audio Engineering Society Preprint.* 16 pp.

Denning, P.J (1992). Neural networks. *American Scientist,* **80**, 426–429.

Denzler, J. *et al.* (1993). Going back to the source: inverse filtering of the speech signal with ANNs. *Eurospeech'93: Proceedings,* Vol. 1. pp. 111–114.

Gao, Y., and J.-P. Haton (1993). Noise reduction and speech recognition in noise conditions tested on LPNN-based continuous speech recognition system. *Eurospeech'93: Proceedings,* Vol. 2. pp. 1035–1038.

Hernaez, I. *et al.* (1993). A segmentation algorithm based on acoustical features using a self-organizing neural networks. *Eurospeech'93: Proceedings,* Vol. 1. pp. 661–664.

Hernandez-Mendez, J.A., A.R. Figueiras-Vidal (1993). Measuring similarities among speakers by means of neural networks. *Eurospeech'93: Proceedings,* Vol. 1. pp. 643–646.

Hinton, G.E., A.H. Waibel and K.J. Lang (1990). A time-delayneural network architecture for isolated word recognition. In *Neural Networks,* Vol. 3. pp. 23–43.

Hoyt, J.D., and H. Wechsler (1990). An examination of the application of multi-layer neural networks to audio signals processing. *IJCNN'90: Proceedings.*

Hunt, M.J (1992). The speech signal. In *Digital Speech Processing, Speech Coding, Synthesis and Recognition,* Vol. 2. Prentice-Hall., pp. 43–71.

Jekosh, U (1993). Speech quality assessment and evaluation. *Eurospeech'93: Proceedings,* Vol. 2. pp. 1387–1394.

Klimasauskas, K., J. Guiver and G. Pelton (1989). *Neural Ware: Neural Computing,* Vol. 1. Pittsburgh. pp. 268.

Knohl, L., and A. Rinscheid (1993). Speaker normalization andadaptation based on feature-map projection. *Eurospeech'93:Proceedings,* Vol. 1. pp. 367–370.

Kohonen, T (1984). *Self-Organization and Associative Memory.* Springer–Verlag, Berlin. 314 pp.

Kwong, S.Y (1993). Discrete utterance recognition based onnonlinear model identification with single layer neural networks. *Eurospeech'93: Proceedings,* Vol. 2. pp. 2419–2422.

Lopez-Gonzalo, E., L.A. Hernandez-Gomez (1993). Fast vector quantization using neural maps for CELP AT 2400 bps. *Eurospeech'93: Proceedings,* Vol. 1. pp. 55–58.

Mogensen, C (1993). Adaptive learning networks. *Tech. Material of SWL Inc.,* 40 pp.

Mokbel, C., J. Monne and D. Jouvet (1993). On-line adaptation of a speech recognizer, to variations in telephone line conditions. *Eurospeech'93: Proceedings,* Vol. 2. pp. 1247–1250.

Morgan, D.P., and C.L. Scofield (1994). *Neural Networks and Speech Processing.* Kluwer Academic Publ., 392 pp.

Rosenblatt, F (1962). *Principles of Neurodynamics.* Spartan Books, Washington.

Rumelhart, D.E., G.E. Hinton and R.J. Williams (1986). Learning internal representations by error propagation. In *Neurocomputing: Foundations of Research,* Vol. 41. pp. 318–362.

Simpson, P.K (1990). *Artificial Neural Systems: Foundations, Paradigms, Applications, and Implementations.* Pergamon Press., 210 pp.

Sorensen, H.B.D., and U. Hartmann (1991). A self-structuring neural noise reduction model. In *European Conference on Speech Communication and Technology Proceedings,* Vol. 4..

Sorensen, H.B.D., and U. Hartmann (1992). Self-structuring hidden control neural model for speech recognition. *ICASSP'92: Proceedings,* Vol. 2. pp. 353–356.

Sorensen, H.B.D., and U. Hartmann (1993). Pi-sigma and hidden control based self-structuring models for text-independent speaker recognition. *ICASSP'93: Proceedings,* Vol. 1. pp. 537–540.

Sorensen, H.B.D., and U. Hartmann (1994). Hybrid model decomposition of speech and noise in a radial basis function neural model framework. *ICASSP'94: Proceedings,* Vol. 2. pp. 657–660.

Tadj C., and F. Poirier (1993). Improved DVQ algorithm for speech recognition: a new adaptive learning rule with neurons annihilation. *Eurospeech'93: Proceedings,* Vol. 2. pp. 1009–1012.

Trompf, M. *et al.* (1993). Combination of distortion-robust feature extraction and neural noise reduction for ASR. *Eurospeech'93: Proceedings,* Vol. 2. pp. 1039–1042.

Tsoukalas, D.E., J. Mourjopoulos and G. Kokkinakis (1993). Neuralnetwork speech enhancer utilizing masking properties. *Eurospeech'93: Proceedings,* Vol. 3. pp. 1595–1598.

Wan, E.A (1990). Temporal backpropagation for FIR neural networks. *IJCNN'90: Proceedings*. pp. 575–580.

Yin, L., J. Astola and Y. Neuvo (1991). Adaptive neural filters. *IEEE Workshop on Neural Networks for Signal Processing: Proceedings*. pp. 503–512.

Received March 1995

D. Navakauskas was born in 1969. He received Engineer Diploma in Radioelectronics from Vilnius Technical University in 1992 and M.Sc. degree in Electronics from Vilnius Technical University in 1994. Currently he is a post graduate student at Radioelectronics Department of Vilnius Technical University. His research interests include digital speech signal processing, speech signal restoration and artificial neural networks.

NAUJOS KRYPTYS KALBOS SIGNALŲ APDOROJIME TAIKANT DIRBTINIŲ NEURONŲ TINKLUS

Dalius NAVAKAUSKAS

Straipsnyje apžvelgtos kalbos signalų apdorojimo srities mokslinės publikacijos, kuriose pateikti sėkmingi dirbtinių neuronų tinklų taikymo rezultatai. Glaustai supažindinama su atgalinio sklidimo bei Koheno žemėlapio dirbtinių neuronų tinklais. Pagrindinis dėmesys skiriamas detaliai dirbtinių neuronų tinklų taikymo sričių bei būdų analizei. Išnagrinėti dirbtinių neuronų tinklų taikymo būdai kalbos signalų atpažinime, klasifikavime, kodavime, kalbančiojo identifikavime, kalbos požymių radime, kalbos signalų detektavime ir nuspėjime, triukšmų slopinime kalbos signaluose. Bandoma atskleisti, kada ir kodėl dirbtinių neuronų tinklų taikymas tampa alternatyva klasikiniam adaptyviajam signalų apdorojimui.