Temporal Processing Neural Networks for Speech Recognition

Alexej V. Ivanov[2], Alexander A. Petrovsky[1]
Computer Engineering Department, Belarusian State
University of Informatics and Radioelectronics
6, P.Brovky Str. 220027 Minsk, Belarus,
tel.: +375-17-2312910, fax.: +375-17-2310914
e-mail: palex@it.org.by[1]
e-mail: alwork@glasnet.ru[2]

Abstract

Application of the temporal processing neural networks (TPNNs) to the speech recognition is justified by the nature of the task. Indeed ASR is a sequence recognition problem and assumes incorporation of time into decision process. Static models treat elements of sequence as independent patterns, which is clearly unrealistic. On the other hand temporal processing nets, built on the basis of multilayer perceptrons give us a hope to dismiss this assumption.

1. Introduction

In our attempt to review the application of the neural networks in the task of speech recognition let us first make use of Marr theory briefly reviewed in [1]. In accordance with it "any complex information-processing system can be studied with respect to three distinct levels of description":

1. *Computational* level – one for description of *the goal* of computation and justification why this goal is appropriate. Here we would try to formulate the task of speech recognition. As we will see later although these formulations are quite diverse they share the common ground of pattern recognition problem, which can be effectively solved with help of neural network approach.

2. *Algorithmic* level – specification of particular algorithms served to achieve the task, specified at the previous level. Here we will restrict ourselves to consideration of temporal processing neural networks (TPNNs) as opposed to static nets, which are already significantly covered in literature. We will present several views onto temporal processing networks, describe their meaning from the signal processing, theory of finite-state automata and probabilistic points. We would try to discuss the drawbacks of chosen approach also.

3. *Implementational* level – specification of the details of realization of the chosen algorithm. Here we will briefly discuss training algorithms developed for temporal processing neural networks.

2. The Speech Recognition Task

2.1 Phonetic Decoding

The earliest attempts to extract information from the speech utterances are at least one-century-old. They had their origin in works of linguists who declared phoneme as the most elementary speech building block. From the

point of view of modern generative phonetics this is not quite true. But let us formulate this task as follows: *Phonetic Decoding – static pattern recognition task with the aim to correctly classify samples coming from one of the limited amount phone classes, phonetic transitions are assumed to take place instantly.*

Practically, there are about sixty different phones distinguished in TIMIT corpus, which constitute the major allophonic realizations of phonemes in the English language. Decoding assumes that each phone has stable target, reached in the process of the generation. There is the problem with stops, their targets are very short periods of time with no airflow and thus no sound, that is why it is a common practice to distinguish two intervals of a stop: closure and release. There are also difficulties with recognition of diphthongs and triphthongs: they have respectively two and three targets successively reached during generation. In any case turning the temporal signal sequence into spatial pattern and manipulating the amount of data fed to the classifier at a time one can find the optimal width of a pattern for phonetic decoding.

2.2 Isolated word recognition

In this formulation our job is to discriminate between speech utterances, which represent isolated words coming from the limited vocabulary. The word boundaries are assumed to be prespecified or found with the help of some external pause detector. There are two ways to extract information from the speech flow:

1. Extract limited amount of features from each utterance, which is in general case is of variable length.
2. Extract limited amount of features per fixed time interval (phone or syllable duration), discriminate between those intervals and generate the word guess with the help of some kind of word temporal structure model e.g. HMM.

There are a number of drawbacks associated with this approach, among the most discouraging we can name:

1. Difficulties with outperforming coarticulation effects at the word boundaries in informal fast speech.
2. Difficulties with introduction of new words to the predefined vocabulary.
3. Impossibility to adequately model word pronunciation variations with the limited amount of features per word.
4. Grammatical forms are treated as different words.

Difficulties with introduction of new words can be outperformed with the help of representation of words as concatenations of syllables, which are modeled at the training stage.

2.3 "Voice control"

This is a task to recognize small amount of simple commands, which are in most cases short sentences built from limited number of words, with some amount of variation of the exact formulation. This task has little difference from isolated word recognition problem, besides the fact that some additional information can be gained from simple grammar. Current state of speech recognition systems allows developing quite reliable "voice control" applications.

2.4 Continuous Speech Dictation Systems

This is the most difficult task of speech recognition. It presupposes usage of well-developed grammar and taking advantage of semantic context. Here we set up a *goal to correctly recognize an arbitrary "well-formed" sentence.* Current continuous speech recognition systems produce a word sequence, which best fits to the perceived utterance as an output. The basic disparity of such formulation with the human way to recognize speech in the fact, that humans able to correct possible grammatical errors of the sentence while understanding the overall meaning, in other words "wellformedness" is not a general precondition for correct recognition.

2.5 Meaning Extraction

This task is frequently associated with researches in the field of artificial intelligence, but from the speech recognition point of view meaning extraction task *constitutes the most abstract representation of utterance at the semantic level.*

2.6 Various improvements

There were several attempts to compare human and machine performance in various speech recognition tasks. The results show that while computers outperform humans in simple

phonetic decoding tasks, humans are much more superior in more complicated dictation and meaning extraction tasks. The main advantages of the human way to recognize speech are:

1. Large vocabulary;
2. Robustness to the environmental noise (so called "cocktail party effect").
3. Speaker Independence – The ability to effectively cope with pronunciation variations from speaker to speaker, regional dialects, foreign accents, etc.
4. Independence of speech rate, ranging from quite slow and clear dictation to fast and often not complete informal conversation utterances.

2.7 Tier representation of speech

Most of the modes employed in recognition treat speech as sequence of some elementary events (patterns, which are classified by recognizer) sequentially concatenated to represent entire message. But linguists for a long time already regard the speech as the communication of information represented at the several tiers: articulatory-acoustic, phonological, grammatical, prosodic, and semantic. These tiers tightly interact with each other in both production and perception processes.

Modeling of such interaction would allow significant improvement of the recognition performance compared to the already developed systems. Many research groups at the time focussed their efforts on this problem [2].

3. Usage of Neural Networks in Speech Recognition

3.1 The Basis

As one can see from the previous formulations of the speech recognition task, all of them share the common idea of statistical pattern recognition: *To label incoming patterns with the probability of misclassification being minimal.* From the statistical decisions theory we know that classifier, which posses this property must assign to the incoming pattern X a class C if the value of posterior probability $P(C \mid X)$ is maximum upon all possible classes. For proof see [3].

Following facts lay in the foundation of usage NN in speech recognition:

- Neural networks trained in classification mode happen to estimate such posterior probabilities (see [4] for proof).
- Neural networks can learn, in other words their parameters can be estimated from some training set automatically with the help of some learning algorithm, without explicit construction by the designer.
- NNs are massively parallel structures, and once properly implemented they can perform their computation very fast.

Neural network models form the broad class of semi-parametric models which is laying between two extremes: parametric models and non-parametric models. Semi-parametric models can be viewed as a compromise characterized by making less constrained assumptions about the process to be modeled than parametric approach while having moderate number of free parameters significantly smaller than in non-parametric modeling.

Many NN architectures were tried in the problem of Speech Recognition, among these we can name MLPs [4], RBF [5], TDNN [5], Recurrent Networks [5] and many other.

Static MLPs, their combination with HMM, RBF networks significantly covered in literature [3], [6], [7], [8] (as well as authors [9], [10], [11]) and lay beyond the scope of this paper.

As the drawback of mentioned approaches we can name the fact that neighboring patterns in the sequence are treated as independent, it is a quite unrealistic assumption. We can at least potentially dismiss this assumption by incorporation time into network operation. Further we'll consider only networks, which are built on the basis of multilayer perceptrons.

3.2 Temporal Processing with Neural Networks

Unlike static pattern classification in sequence recognition we understand that input spatial patterns come as a temporal sequence (figure 1), and as a response we receive temporal sequence of network outputs. The relation between these two sequences is defined by the structure of neural network.

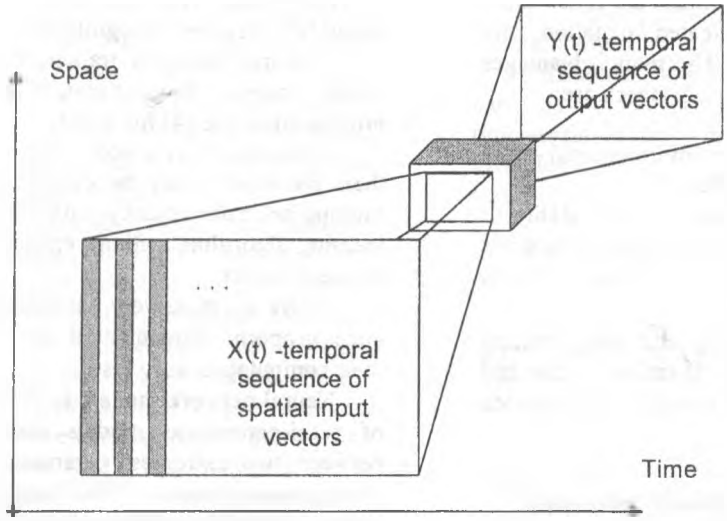

Figure 1. Schematic representation of the temporal processing

There are basically two methods of incorporating time into pattern processing:

1. First way is to present a feed-forward network with some temporal window of input signal (figure 2). Each time the output of such network is computed as a function of input pattern sequence of some finite length:

Thus there is a guarantee that network response for the input pattern sequence of finite length would be also finite.

We have to mention that there are two versions of TLFN: *focussed* and *distributed.* Focussed nets have temporal window only at the input, while each layer of distributed net possesses it's own window, in other words time dependence is distributed through network. Distributed nets can be "unfolded through time" to become focussed at the cost of additional weights.

Universal Myopic Mapping Theorem describes the computational power of the focussed TLFN. It can be stated as follows: *Any shift-invariant myopic dynamic map can be uniformly approximated arbitrary well by a structure consisting of two functional blocks: a bank of linear filters feeding a static neural network.* (After [12])

2. Second way comprises an introduction of recurrent connections between the temporal window of chosen neuron output and neuron inputs of the same or previous level (figure 3). In this situation feed-forward flow is no longer the only direction in which information can be transmitted within a network and each time network output is computed as a function of current input and internal state of the network.

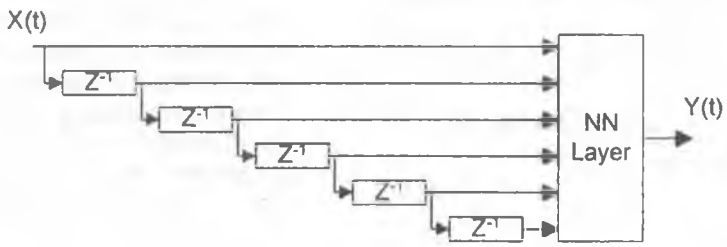

Figure 2. Time Lagged Feed-forward Network (TLFN)

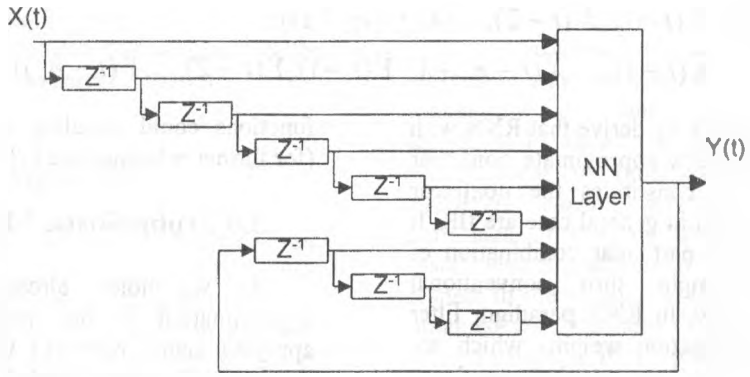

Figure 3. Recurrent Neural Network (NARX model)

Similar to TLFN we can introduce two versions: *globally recurrent* networks only use their outputs as a feedback signal, while in *locally recurrent* networks recurrent connections are introduced at the neuronal level.

Bounded input – bounded output (BIBO) stability criterion is not suitable for RNN, their outputs are always bounded because of neuron saturating output nonlinearity (sigmoid function). This means that RNN's are *always BIBO stable*. That is why discussion of stability RNN as any other dynamical nonlinear system must be done in the Lyapunov sense.

3.3 Delays: Tapped vs. Gamma.

Tapped delay line, characterized by transfer function $G(z) = z^{-1}$ (showed in figures 2 and 3) isn't the only possible way to incorporate short-term memory into network operation.

Let us define the *memory depth* D as a first time moment of total impulse response h_p of the delay line of order p:

$$D = \sum_{t=0}^{\infty} t h_p(t) \qquad (1)$$

Memory depth characterizes ability of a delay line to keep information about the past with a time flow.

Let us further define *memory resolution* R to be a number of taps per unit time interval. Memory resolution defines the quality of the representation of the past.

It can easily be seen that for tapped delay line we have $D = p$, $R = 1$ and their product $DR = p$.

Let us now replace conventional tapped delays with a single pole discrete time filter with a transfer function:

$$G(z) = \frac{\mu}{z - (1 - \mu)}, \quad \text{in which}$$

$0 < \mu < 2$, for the filter to be stable. (2)

This structure would constitute *gamma memory* delay line introduced in [13], it could be shown (see [13], [7]) that in this case $D = p/\mu$, $R = \mu$ and their product remains $DR = p$.

With $\mu = 1$ gamma memory represents ordinary tapped delay line.

With $\mu < 1$ gamma memory is able to store more distant occasions in the past (i.e. increase memory depth) with coarser memory resolution than conventional tapped delay line. The parameter μ can be adapted to achieve the maximum performance.

3.4 Filtering Model.

Here one can clearly see the analogy with FIR and IIR filters. Even more, neural networks of such configurations can be viewed as a generalization of standard filters to the nonlinear filtering. As it was already discussed, the power of neural networks lies in possibility to approximate any finite nonlinear function with arbitrary precision. It should be noted also *that spatial dimensionalities of input and output signals are not restricted* and can be chosen independently one from another.

Thus focussed or distributed TLFN are the approximations of nonlinear non-recurrent filters, which are always FIR, which can be described by the function (3).

$$\bar{Y}(t) = f(\bar{X}(t), \bar{X}(t-1), \bar{X}(t-2), ..., \bar{X}(t-n_x+1)); \quad (3)$$

$$\bar{Y}(t) = g(\bar{X}(t), \bar{X}(t-1), ..., \bar{X}(t-n_x+1), \bar{Y}(t-1), \bar{Y}(t-2), ..., \bar{Y}(t-n_y)) \quad (4)$$

Going further we may derive that RNN with global or local feedback approximate nonlinear function (4), which constitutes the nonlinear recurrent filters, which in general case are IIR. It should be noted that particular combination of the coefficients might turn conventional recurrent filter to FIR. In RNN paradigm filter coefficients are connection weights which are not fixed during training by any relation and it is more convenient to think of RNN as "IIR filter in general".

3.5 State-Space Model

Dynamically driven recurrent networks may be viewed as some sort of finite-state automata. In this case vector $\bar{S}(t)$ is the *state vector* of the model, i.e. a number of internal variables used to store information about past behavior of the model, needed in combination with external input to fully describe its future behavior.

$$\bar{S}(t+1) = f_1(W_i \bar{X}(t) + W_s \bar{S}(t)) \quad (5)$$

$$\bar{Z}(t) = f_2(W_o \bar{S}(t)) \quad (6)$$

in which $\bar{X}(t)$ - external input to the model at a time t, $\bar{Z}(t)$ - model output, W_i, W_s, W_o - connection weight matrixes for inputs, feedback state vector and output respectively. Here we treat multiply delayed output vector $\bar{Y}$ (figure 3) as single state vector $\bar{S}(t)$ and output vector $\bar{Z}(t)$ is computed by as single feed-forward layer with activation function f_2.

If we would brake feedback connections we will have a simple feed-forward MLP computing function f_1 of its inputs, which is capable to approximate any finite nonlinear function with arbitrary precision. Thus our original system can approximate wide class of nonlinear dynamical systems. It should be noted that this approximation holds for compact subsets of input space and finite time intervals.

We also should mention that fully connected recurrent networks with sigmoid activation functions could simulate any Turing machine (for further reference see [7])

3.6 Probabilistic Meaning

As we noted already that probability approximation is our major concern while applying neural networks to the ASR task we have to define the probabilistic meaning of the recurrent networks.

TLFN trained in classification task would approximate the posterior probability of visiting state C at a time t in the form of (7), where $\{\bar{X}\}$ refers the full input sequence, $\{C\}^{-1}$ - state sequence visited at the previous time steps.

It is also possible to shift input window of the focussed TLFN to equally represent future and past contexts. This was done in the NetTalk experiments [14], in the experiments of Morgan and Bourlard [4]; authors also tried such configuration [10]. Such system is not casual (output value depends upon some future values of input) or we can say that there is some delay between the moment the input first time appears in the processing and the time it is associated with some particular class. But human hearing system also posses this property. It is proved in the experiments with temporal masking [15] that during periods of 20-50 ms *before* and 100-200ms *after* loud masker sound faint test sounds perceived attenuated.

Recurrent neural networks trained in the same task will approximate posterior probability as (8), in which $\hat{C}(t-1)$ - represents a network estimate of the state visited at time $t-1$, which might not coincide with the true one.

The problem of replacement of true state with expected one gives rise of a technique called "teacher forcing", which can be described as substitution during the training stage of the possibly wrong network estimate with desired state obtained from the training set. This accelerates training because at some time network may have correct weights, but occasionally be at the wrong place at the state space. On the other hand "teacher forcing"

$$P(C \mid \{\bar{X}\}, \{C\}^{-1}) = P(C \mid X(t), X(t-1), X(t-2), ..., X(t-n+1)), \quad (7)$$

$$P(C \mid \{\bar{X}\}, \{C\}^{-1}) = P(C \mid X(t), X(t-1), ..., X(t-n_x+1), \hat{C}(t-1), ..., \hat{C}(t-n_y)) \quad (8)$$

removes all information about previous errors made by network. Globally this would lead to optimizing error function different from "unforced" case.

3.7 Training algorithms

A number of training algorithms was developed for TLFNs and recurrent MLPs on the basis of standard backpropagation algorithm.

Focused TLFNs can be trained with standard backpropagation algorithm converting temporal input signal into spatial input vector. This procedure can be done once at the stage of preparing training or testing sets.

Any distributed TLFN can be "unfolded" in time to become equivalent focussed network with much bigger input window size and some amount of shared weights and trained with "static" version of backpropagation, but this procedure seems to be impractical. A special procedure called *Temporal Back-Propagation* algorithm was proposed by Wan [16] for the case of distributed TLFN.

Now, that we turn to the brief discussion of the training algorithms for the recurrent networks let us first define two properties:

Algorithm is thought to be *local in time* if it can be executed as temporal input sequence arrives and allows learning using only information contained in the temporally neighboring frames of input signal. Such algorithm can be used to learn sequences of arbitrary length.

Algorithm is *local in space* if weight updates of each neuron can be computed only form the information about immediate neighbors of chosen neuron. Such algorithms can be easily implemented in parallel mode.

For the globally recurrent network locality in space and time are alternatives and can't be achieved simultaneously without any simplifying assumptions.

Back-Propagation Through Time was proposed in [17] as recurrent extension of the standard algorithm and can be summarized as follows:

1. Forward propagation of the input sequence of fixed length, memorizing each neuron's activation at every time step.
2. Backward computation through space and time of correction values for each weight in the network.

This algorithm resembles backpropagation through equivalent "unfolded" feed-forward network. This "unfolding" procedure is possible because input sequence is restricted to limited length. Even more a truncated version of BPTT algorithm was introduced [18], which rejects longer time dependencies than some predefined length. BPTT algorithm is local in space.

A local in time alternative (*Real-Time Recurrent Learning*) was proposed in [19]. The core idea of this algorithm is in the use of instantaneous gradients of the cost function with respect to the weights in the network. Gradients obtained with the help of RTRL would deviate around values of BPTT gradients. This deviation is exactly analogous to the behavior of on-line update technique in front of batch update in conventional static backpropagation.

Further comparison of BPTT and RTRL reveals that while BPTT is computationally simpler and more effective than RTRL, RTRL is casual and therefore suitable for continuous learning without explicitly predefined training set.

Recently several other training algorithms (Recursive Backpropagation, Casual Recursive Backpropagation) were introduced for the case of recurrent networks with local feedback [20]. The main advantage of such algorithms is the fact that training algorithm can be simultaneously local in time and local in space.

3.8 Universal Time flow Rate

But in spite of generality of presented models, one important limitation can be noted in these formulations. All models mentioned above have a property of *universal time flow rate*, i.e. one time step at the input strictly cause exactly one time step at the output. Besides the fact that this brings extra computational complexity (all parts of the network should operate at the time scale of the input signal), it is closely related to the "vanishing gradients" problem (for complete description [5]), in short, recurrent network fails to memorize long term dependencies.

To illustrate time scale problem, let us consider the problem of phoneme identification. If we build a classifier based on TLFN or RNN (for particular examples see [4], [5], [10]) we are forcing the network to produce its outputs at a rate of input preprocessed acoustic signal, which can be considered as a piecewise stationary process through a time period of measurement, but anyway at the phoneme boundaries speaker articulators are in transition to the next stable target configuration. Additional models should be used (such as tri-state phone HMM) to generate global decisions about produced

utterance. In spite of the fact that in classification mode the network approximates posterior probabilities on the basis of which the optimal classification should be done, it does this "too fast".

3.9 Why temporal processing neural networks did not solve speech recognition problem

Feed-forward and recurrent MLPs possess many useful properties for speech recognition, these models have well understood signal-processing state-space and probabilistic meaning, they naturally adopt training, that is why they can't be easily rejected as candidate models for speech recognition. But from this point we already can see some drawbacks of this approach comparing to the problem to be solved. Let us briefly review the basic disparities between the general speech recognition task and recurrent network made of sigmoid neurons:

1. "Linguistic tiers" representation of speech assumes sequential decoding at various levels (i.e. transformation of acoustic signal to phones, further transformation from phone sequences to syllables, syllable sequences to words, word sequences to sentences and so on up to the meaning.). Linguists insist (with some evidences) on non-feed-forward information flow between such tiers (so called "tier interaction"). Moreover, time flow rate is *slowing down* in the direction of higher abstraction levels; in other words one phone constitutes some sequence of acoustic vectors. On the contrary, TPNN provide us with uniform time flow rate mapping between input and output sequences. This problem was addressed from the various positions (Dynamic Time Warping, statistical models like Hidden Markov Models, e.t.c.) but all this approaches suffered from the difficulties of introduction of new higher abstraction entities, such as problems with incorporating new words into a limited dictionary speech recognition system.

2. The size of output alphabet grows very fast from tier to tier from less then hundred different phones to several thousand syllables (There are about 8000 distinct syllabic structures in English language in accordance with [2]), tens of thousands words, virtually unlimited number of shades of meaning. It clearly becomes impractical to encode each possible output class with separated neuron of the output level of the network at the higher processing stages.

3. Learning algorithms developed for MLPs assume more or less equal representation of samples coming from different classes at the training stage in order to achieve equal modeling power for the various output classes. This requirement is clearly unrealistic in continuous real-time learning procedure.

The majority of the problems could possibly be overcome with the help of Pulse-Coupled Neural Networks (PCNN) (for comprehensive foundation look [21], [22]), an approach with biological grounds frequently mentioned as more precise model of biological neuron introduced by Reinhard Eckhorn in 1990. Despite the fact that works of Eckhorm were inspired by specific activity in *visual cortex* of small mammals and the most of known applications of PCNN are in the field of image processing PCNNs posses several useful features for speech processing also:

1. Output neuronal activity represents series of short spikes with a rate proportional to a sigmoid function of the feeding input.

2. Synchronous groups of neurons act as "bigger neurons", operating at the slower time scale, firing not a single spike at a time, but series of spikes with different amplitude.

3. Signatures of that spike series can be viewed as a way to encode output classes, which gives coding capacity bounded by the total number of neurons (not only in the output layer), *coding capacity grows with time scale slowing down*. This is exactly what we have observed for higher speech processing stages.

4. Signature output coding gives an opportunity to introduce a kind of distance measure between different output classes as opposed to "one from m" coding, where there is no possibility to gain any similarity measure from class labels themselves.

Unfortunately learning algorithms for PCNN are not well understood and developed yet, but the principal possibility of such algorithms exists.

4. Conclusion

Temporal processing neural networks provide more suitable framework for speech recognition problem comparing to the conventional static MLPs. They use short-term context information in more effective way than their static counterparts. They provide us with the models, which have less free parameters to be estimated during training.

In their current state they can replace MLPs in the HMM/MLP hybrid recognition system at the phonemic decoding stage.

But even though they have all these useful properties, TPNNs can't be considered as homogenous devices for solving speech recognition problem in general due to the reasons discussed here.

References.

[1] R. M. Golden "Mathematical Methods for Neural Network Analysis and Design" The MIT press, Cambrige, 1996

[2] S. Greenberg "On The Origins of Speech Intelligibility in the Real World", Proc. Of the ESCA Workshop on Robust Speech Recognition for Unknown Communication Channels, April, 1997

[3] B. Ripley "Pattern Recognition and Neural Networks" Cambridge University Press, 1996.

[4] H. Bourlard, N. Morgan "Connectionist Speech Recognition, A Hybrid Approach" Kluwer Academic Publishers, Boston, Dordrecht, London 1994.-312p.

[5] Y. Bengio "Neural Networks for Speech and Sequence Recognition"// International Thomson Computer Press, 1996.

[6] C. Bishop "Neural Networks For Pattern Recognition" Clarendon Press, Oxford 1995.-482p.

[7] S. Haykin "Neural Networks: A Comprehensive Foundation" Prentice Hall Inc., 1999

[8] F. L. Luo R. Unbenhauen "Applied Neural Networks for Signal Processing" Cambridge University Press, 1998

[9] A. Ivanov, A. Petrovsky "Training Multi-Layer Perceptrons in the problem of Static phoneme Identification with the use of TIMIT Speech Corpus", Proc. Of 6th International Workshop on Systems, Signals and Image Processing, 1999 June 2-4, Bratislava, Slovakia;

[10] A. Ivanov, A. Petrovsky "Experiments with Neural Networks for Sequence Recognition in Application to Automatic Speech Recognition" 5th International Conference on Pattern Recognition and Information Processing, May 18-20, 1999, Minsk, Belarus;

[11] A. Ivanov, A. Petrovsky "MLPs and Mixture Models for the Estimation of the Posterior Probabilities of Class Membership", A Workshop on Text, Speech, Dialog TSD'99 Sept. 13-17, 1999, Plzen, Czech Republic.

[12] I. W. Sandberg L. Xu "Uniform approximation of multidimensional myopic maps", IEEE Trans. on Circuits and Systems, vol.44, pp. 477-485, 1997

[13] B. De Vries, J. C. Principe "A gamma model – A new neural model for temporal processing" Neural Networks, vol.5, pp. 565-576, 1992

[14] T. J. Sejnowski, C.R. Rosenberg "Parallel networks that learn to pronounce English Text", Complex Systems, vol. 1, pp. 145-168, 1987

[15] E. Zwicker, H. Fastl, "Psychoacoustics: facts and models" Berlin: Springer-Verlag, 1990, - 354 p.

[16] E. A. Wan "Temporal backpropagation for FIR neural networks" IEEE Int. Joint Conf. On Neural Networks, vol. 1, pp. 575-580, San Diego, CA, 1990

[17] D. Rumelhart, G. Hinton, R. Williams "Learning Internal Representations by Error Propagation" Parallel Distributed Processing, v. 1, ch. 8, pp. 318-362, MIT Press, Cambridge, 1986

[18] R. Williams J. Peng "An efficient gradient-based algorithm for on-line training of recurrent network trajectories" Neural Computation, vol. 2, pp. 490-501, 1990

[19] R. Williams, D. Zipser, "A learning algorithm for continually running fully recurrent neural networks" Neural Computation, vol. 1, pp. 270-280, 1989

[20] P. Campolucci, A. Uncini, F. Piazza, B.D. Rao "On-line Learning Algorithms for Locally Recurrent Neural Networks" IEEE Trans. on Neural Networks, vol. 10, n. 2, March 1999

[21] O. Omidvar, J. Dayhoff (ed.) "Neural Networks and Pattern Recognition" Academic Press, 1998

[22] Special Issue on Pulse Coupled Neural Networks, IEEE Trans. on Neural Networks, vol. 10, n. 3, May 1999