GENERALIZED LOCALLY RECURRENT PROBABILISTIC NEURAL NETWORKS FOR TEXT-INDEPENDENT SPEAKER VERIFICATION

*T. Ganchev, N. Fakotakis**

Wire Communications Laboratory,
Dept. of Electrical and Computer Engineering,
University of Patras, GR-26500 Patras, Greece

D.K. Tasoulis, M.N. Vrahatis

Department of Mathematics,
Artificial Intelligence Research Center,
University of Patras, GR-26500 Patras, Greece

ABSTRACT

An extension of the well-known Probabilistic Neural Network (PNN), to Generalized Locally Recurrent PNN (GLRPNN) is introduced. This extension renders GLRPNNs, in contrast to PNNs, sensitive to the context, in which events occur. A GLRPNN is therefore, able to identify time or spatial correlations. This capability can be exploited to improve performance on classification tasks. A fast three-step algorithm for training GLRPNNs is also proposed. The first two steps are identical to the training of traditional PNNs, while the third step exploits the Differential Evolution optimization method. The performance of the proposed methodology on the task of text-independent speaker verification is contrasted with that of Locally Recurrent PNNs, Diagonal Recurrent Neural Networks, Infinite Impulse Response and Finite Impulse Response MLP-based structures, as well as with Gaussian Mixture Models-based classifier.

1. INTRODUCTION

The locally recurrent global-feedforward architecture was originally proposed by Back and Tsoi [1], who considered an extension of the Multi-Layer Perceptron Neural Network (MLP NN) to exploit contextual information. In their work, they introduced the Infinite Impulse Response (IIR) and Finite Impulse Response (FIR) synapses, able to explore time dependencies in the input data. Ku and Lee [2] proposed Diagonal Recurrent Neural Networks (DRNN) for the task of system identification in real-time control applications. Their approach is based on the assumption that a single feedback from the neuron's own output is sufficient. Therefore, they simplify the fully connected neural network architecture, in order to manage with the training easier. A comprehensive study of several MLP-based Locally Recurrent Neural Networks is also available in Campolucci et al. [3]. The authors of [3] introduced a unifying framework for the gradient calculation techniques, called Causal Recursive Back-Propagation. All approaches, mentioned here, consider gradient based training techniques for neural networks, which, as it is well-known, require transfer functions to be differentiable.

The work presented draws on the concept of a local recurrent global-feedforward architecture, and the locally recurrent layer we propose is similar to the IIR synapse introduced in [1] and the DRNN defined by Ku and Lee. Our approach differs from the aforementioned, primarily, because we consider PNNs instead of MLP NN. Most importantly, however, in the architecture proposed here each summation unit in the recurrent layer receives as input not only current and past values of its inputs, but also the N previous outputs of all neurons in the recurrent layer. In previous work [4], we extended the traditional PNN architecture, proposed by Specht [5], to Locally Recurrent PNN, in order to capture the inter-frame correlations present in speech signals. Here, we generalize the locally recurrent global-feedforward PNN architecture, by adding time-lagged values of its inputs.

2. THE GLRPNN ARCHITECTURE

The GLRPNN is derived from the PNN by including a hidden recurrent layer, which consists of summation neurons with feedbacks. The GLRPNNs (as PNNs) implement the Parzen window estimator by using a mixture of Gaussian basis functions (see [5] for details). If a GLRPNN for classification in K classes is considered, the probability density function $f_i(x_p)$ of each class k_i is defined by:

$$f_i(x_p) = \frac{1}{(2\pi)^{\frac{d}{2}} \sigma_i^d M_i} \sum_{j=1}^{M_i} \exp\left[-\frac{1}{2\sigma_i^2} (x_p - x_{ij})^T (x_p - x_{ij}) \right] \quad (1)$$

for i=1, ..., K, where x_{ij} denotes the j-th training vector from class k_i; x_p is the p-th input vector; d is the dimension of the speech feature vectors; and M_i is the number of training patterns in class k_i. Each training vector x_{ij} is assumed to be a center of a kernel function, and consequently

*This work was supported by the "Infotainment management with Speech Interaction via Remote microphones and telephone interfaces" - INSPIRE project (IST-2001-32746).

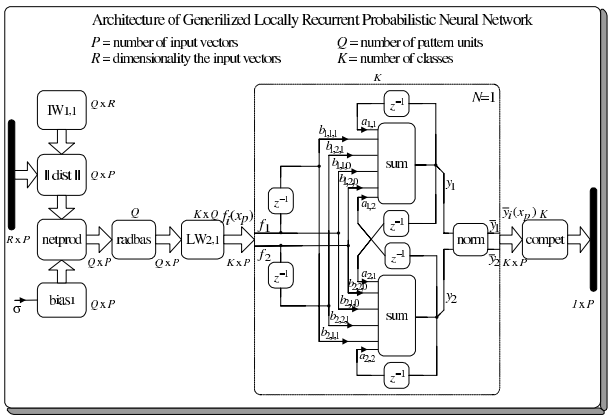

Fig. 1. Architecture of the Generalized Locally Recurrent Probabilistic Neural Network

the number of pattern units in the first hidden layer of the neural network is given by the sum of the pattern units for all the classes. The variance σ_i acts as a smoothing factor, which smooths the surface defined by the multiple Gaussian functions. The value of σ_i can be identical for all pattern units from a specific class, or, as it was originally proposed by Specht, it can be one and the same for all pattern units irrespective of the class. The architecture of the GLRPNN for the case of two classes (K=2) and recurrence depth N=1, is shown in Fig. 1. The locally recurrent layer is delineated by a dashed line. More generally, the recurrent layer can be considered as a form of IIR filter, which smooths the probabilities generated for each class, by using one or more past values of the summation outputs.

The output, $y_i(x_p)$, of the summation units located in the locally recurrent layer is computed by:

$$y_i(x_p) = \sum_{t=0}^{N}\sum_{k=1}^{K} b_{i,k,t} f_k(x_p) z^{-t} + \sum_{t=1}^{N}\sum_{k=1}^{K} a_{i,k,t} y_k(x_p) z^{-t} \tag{2}$$

for i=1, ..., K. $f_k(x_p)$ is the probability density function of each class k_i; x_p denotes the input vector; K is the number of classes; N stands for the recurrence depth; z^{-t} denotes a time delay of t steps; and finally $a_{i,k,t}$ and $b_{i,k,t}$ represent weight coefficients. The output $y_i(x_p)$ of each summation unit is subject to a regularization transformation:

$$\overline{y}_i(x_p) = sgm(y_i(x_p)) / \sum_{j=1}^{K} sgm(y_j(x_p)) \tag{3}$$

which is imposed to retain an interpretation of the output of the recurrent layer in terms of probabilities. The notation sgm denotes the sigmoid activation function.

Finally, the Bayesian decision rule (4) is applied to distinguish the class k_i, to which the input vector x_p belongs:

$$D(x_p) = argmax\{h_i c_i \overline{y}_i(x_p)\}, i = 1, ..., K \tag{4}$$

where h_i is a-priori probability of occurrence of the patterns of category k_i, and c_i is the cost function in case of misclassification of a vector belonging to class k_i. The conditional probability $P(k_i|X)$, all test vectors of set X=$\{x_p\}$, $p = 1, ..., P$ to belong to class k_i, is computed by:

$$P(k_i|X) = \frac{N_{x_p,k_i}}{P} \tag{5}$$

where N_{x_p,k_i} is the number of vectors x_p classified by (4) as belonging to class k_i.

When the task of speaker verification is considered, a speaker independent threshold is applied over the score (5), and a final decision is made. The speaker is rejected as impostor when the probability is below a predefined value, or otherwise is accepted with the identity claimed.

3. THE GLRPNN TRAINING

Similar to the LRPNN training method presented in [4], a three-step training procedure for the GLRPNN is proposed. By analogy to the original PNN, the first training step creates the actual topology of the network. In the first hidden layer, a pattern unit for each training vector is created, by setting its weight vector equal to the corresponding training vector. The outputs of the pattern units associated with the class k_i are then connected to one of the second hidden layer summation units.

The second training step is to compute the smoothing parameter σ_i for each class. According to [6], σ_i is determined as proportional to the mean value of the minimum distances between the training vectors in class k_i:

$$\sigma_i = \lambda \frac{1}{M_i} \sum_{j=1}^{M_i} d_{ij}, \tag{6}$$

where d_{ij} is the minimum Euclidean distance of each pattern unit from class k_i, with all other pattern units from that class. The constant λ is usually chosen in the range $[1.1, 1.4]$. In case σ_i is common for all classes it is chosen empirically, or it is computed by applying (6) on a set, composed by merging the training data for all classes.

The third step is adjusting the weights of the locally recurrent layer by using the same data, exploited at the Radial Basis layer training step. Supervised training of the recurrent layer is equivalent to minimization of the error function:

$$E(w) = \sum_{i=1}^{K} c_i P(Miss|k_i) P(k_i), \tag{7}$$

where the parameter c_i is the relative cost of detection error for the corresponding class k_i, $P(Miss|k_i)$ is the post probability of misclassification of the patterns belonging to class k_i, and the $P(k_i)$ is the a-priori probability of occurrence of the patterns of class k_i in the training data set. The

values of $P(Miss|k_i)$ are obtained in the following way: For a given weight vector $w = \{a, b\}$, the values of y_i are computed, according to (2) and (3), and then (4) is applied. Finally, the post-probability $P(Miss|k_i)$ is computed as $P(Miss|k_i) = 1 - P(k_i|X)$, where $P(k_i|X)$ is obtained from (5) for the case of the training data set.

The total error $E(w) = E(a, b)$ is reduced by adjusting the weight vectors b and a by means of the Differential Evolution (DE) algorithm [7]. From all the five variation operators proposed in [7], we have observed that (8) provides the best performance in the weight optimization procedure. The new candidate for weight vector v^i_{g+1} is generated by:

$$v^i_{g+1} = \omega^{r1}_g + \mu(\omega^{r1}_g - \omega^{r2}_g), \tag{8}$$

where ω^{r1}_g, ω^{r2}_g, ω^{r3}_g and ω^{r4}_g are randomly selected vectors, different from ω^i_g, ω^{best}_g is the best member of the current generation, and the positive mutation constant μ controls the magnification of the difference between two weight vectors. The trial weight vectors obtained at the crossover step of the DE algorithm are accepted for the next iteration only if they yield a reduction of the value of the error function, otherwise the previous weights are retained. The training process ends when the target error margin is reached, or after completing a predefined number of iterations.

4. EXPERIMENTS AND RESULTS

Our text-independent speaker verification system [8], a participant in the 2002 NIST Speaker Recognition Evaluation, was used as a platform to compare the performance of the GLRPNN with that of other locally recurrent architectures like LRPNN, DRNN, and IIR and FIR MLP NNs.

In the speaker verification task, two classes (enrolled user and a reference) are considered. Fifty male speakers, extracted from the PolyCost v1.0 telephone-speech speaker recognition corpus [9], were enrolled as authorized users. As a training data, ten utterances (about 17 seconds of voiced speech) obtained from the first session of each speaker, containing both numbers and sentences, were used. The reference model, was build by combining all users' training data. Depending of the probabilities computed for the user's and reference models, a binary decision is made for every speech frame. These decisions are averaged over a whole test utterance, and a final decision for acceptance or rejection is made. Utterances from all the 74 male speakers (50 users + 24 unknown to the system) available in the database were used to perform test trials. Each user model was tested by four target trials from the second session of the corresponding enrolled user, and by 292 trials from both unknown impostors and pseudo-impostors. About 1.3 seconds of voiced speech per test utterance were available.

Fig. 2 presents the normalized distributions of the speaker scores, generated by the traditional PNN (left) and the GLRPNN with a recurrence depth N=1 (right), trained by the variation operator (8). The enrolled users are represented by dashed line and the impostors by solid one. The considerable spread of the distribution of both users' and impostors' scores for the PNN is obvious. In contrast, as Fig. 2 demonstrates, the GLRPNN classifier produces a smaller deviation from the mean value for both the users and the impostors. In about 60% of the cases, a zero probability for the impostor trials was produced, which is a major improvement compared to the only 40% of the traditional PNN. Moreover, the GLRPNN exhibited a significant concentration of the enrolled users' scores at the maximum probability point (more than 50% of all trials), in contradistinction to the PNN, where the user scores were spread out over a much wider area in the upper part of the scale. A better separation of the two classes was observed, that was expressed in the terms of the Equal Error Rate (EER) as 3.43% and 3.07%, for the PNN and the GLRPNN, respectively.

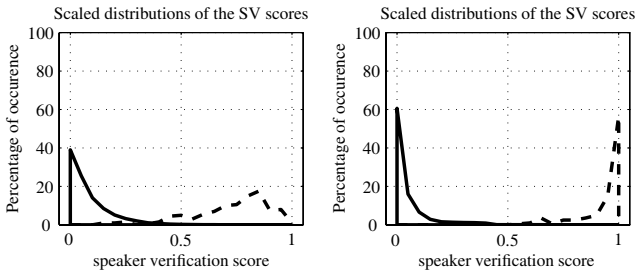

Fig. 2. Distribution of user (dashed line) and impostor (solid line) scores for the traditional PNN (left), and the GLRPNN for N=1 (right)

Table 1 presents the EER obtained for a GLRPNN-based speaker verification and various values of the recursion depth N. As expected, when N increases – the EER decreases,

Table 1. The EER depending on the recursion depth N

N	0	1	2	3	4	5
EER[%]	3.43	3.07	3.04	2.96	2.88	9.00

because a larger part of the inter-frame correlation is identified and exploited. The major increase of the EER, observed for N=5, is mainly due to the insufficient amount of training data. The number of weight coefficients ($(2N + 1)K^2$) in the recurrent layer depends in linear manner from N, but for large N more training data are required. When such data are not available, the neural network becomes specialized on the training set and is not able to generalize well on unknown data. Important constraint that limits the recurrence depth is also the time window size. For large values of N the time window could spread across two or more phonemes, and even across syllables. In that case, the neural network becomes sensitive to the linguistic information carried by the training data, which can be very useful in the

Table 2. The EER in [%] depending on the architecture

Arch	FIR	IIR	DRNN	LRPNN	GLRPNN
EER	3.66	3.58	3.44	3.24	3.07
# w^*	4	6	10	8	12

case of speech recognition or text-dependent speaker verification, but decreases the speaker verification performance when a text-independence is considered.

Table 2 presents a performance comparison among the: FIR and IIR structures [1], DRNN [2], LRPNN [4] and GLRPNN, over a common data set, and a time-window N=1. Exploiting one past value of the input data, these structures also exploit (with exception of the FIR) one past value of their own output, or the outputs of all classes. Thus, for the same time window, a different number of weighted connections are available in each structure. The symbol "*", next to the number of weight coefficients w, suggests that no biases were considered. As shown in Table 2, the best speaker verification performance is achieved for the GLRPNN, followed by LRPNN, DRNN, IIR and FIR at the end. An increasing EER is observed, as the number of weighted connections decreases. The only deviation here is the LRPNN which possesses less connections than DRNN, but exhibits better performance. In our opinion that is due to the presence of cross-class feedbacks from the past outputs of all classes in the LRPNN architecture. Thus, for the same size of the time-window, the linkage of the LRPNN is better suited to capture the dynamics of the process.

For the sake of comparison the PNN classifier was replaced by one, based on Gaussian Mixture Models (GMM), with an equivalent complexity – spherical kernels, 128 mixtures for the user models, and 256 mixtures for the Universal Background Model, and EER=3.01% was obtained. Thus, the GMM system significantly outperformed the baseline one. When the GMM result is compared to the GLRPNN ones (see Table 1), however, a lower EER for N=3, and N=4 (2.96% and 2.88%, respectively) was observed. An important advantage of the GLRPNNs is that they concurrently keep faster training times than the GMMs.

In conclusion, the experimental results support the claim that the new GLRPNN architecture outperforms traditional PNN, LRPNN, DRNN, IIR, and FIR architectures. For the specific cases of recurrence depth N=3, and N=4, the GLRPNNs demonstrate better performance than GMMs with an equivalent complexity. Consequently, on one hand the GLRPNN effectively improves the speaker verification performance, with only a limited increase of the complexity, when compared to the other recurrent structures studied here, and on the other hand, the GLRPNN are able to achieve better performance (for N=3 and N=4) then the GMMs, while keep faster training times.

5. CONCLUSIONS

Introducing the Generalized Locally Recurrent PNN, we extended the traditional PNN architecture to exploit the inter-frame correlation among the features extracted from successive speech frames. In addition to the GLRPNN architecture, a fast three-step training method was proposed. Comparative experimental results for text-independent speaker verification confirmed the practical value of the proposed GLRPNN. A superior performance, in comparison to other recurrent structures, was achieved. A relative reduction of the EER by 10.5% was observed, in contrast to the one for the PNN, without significantly increasing the complexity of the network, and without requiring additional training data.

6. REFERENCES

[1] A. D. Back and A. C. Tsoi, "FIR and IIR synapses, a new neural network architecture for time series modeling," *Neural Computations*, vol. 3, pp. 375–385, 1991.

[2] C. C. Ku and K. Y. Lee, "Diagonal recurrent neural networks for dynamic system control," *IEEE Transactions of Neural Networks*, vol. 6, no. 1, pp. 144–156, 1995.

[3] P. Campolucci, A. Uncini, F. Piazza, and B. D. Rao, "On-line learning algorithms for locally recurrent neural networks," *IEEE Transactions of Neural Networks*, vol. 10, no. 2, pp. 253–271, 1999.

[4] T. Ganchev, D. K. Tasoulis, M. N. Vrahatis, and N. Fakotakis, "Locally recurrent probabilistic neural networks for text independent speaker verification," in *Proc. of the EuroSpeech*, 2003, vol. 3, pp. 1673–1676.

[5] D. F. Specht, "Probabilistic neural networks," *Neural Networks*, vol. 3, no. 1, pp. 109–118, 1990.

[6] B. J. Cain, "Improved probabilistic neural networks and its performance relative to the other models," in *Proc. SPIE, Applications of Artificial Neural Networks*, 1990, vol. 1294, pp. 354–365.

[7] R. Storn and K. Price, "Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

[8] T. Ganchev, N. Fakotakis, and G. Kokkinakis, "Text-independent speaker verification based on probabilistic neural networks," in *Proc. of the Acoustics 2002*, Patras, Greece, Sept. 30th – Oct. 1st, 2002, pp. 159–166.

[9] J. Hennebert, H. Melin, D. Petrovska, and D. Genoud, "Polycost: A telephone-speech database for speaker recognition," *Speech Communication*, vol. 31, pp. 265–270, 2000.