

Recurrent Neural Prediction Model for Digits Recognition

Tellez Paola

Abstract— There are three models of recurrent neural networks that can be used for speech recognition, but we do not know which one is the best suited for Spanish digits recognition. Each language uses different parts of the mouth to create sounds, so it is logical that we develop different parts of the brain to recognize these sounds. The best recurrent neural network to recognize Spanish digits is Jordan recurrent neural network. The next step is to improve the performance of Jordan recurrent neural network for Spanish recognition.

Index Terms— Recurrent neural network, speech recognition, optimization, prediction.

1 INTRODUCTION

Spanish is a romance language that is very different to the English. One difference is the way to pronounce the ten digits. It is not known which parts of the brain are used to recognize Spanish, and recurrent neural networks are models that try to simulate how it could be done. In this paper, the recurrent neural prediction model is used to predict the recognition of each one of ten Spanish digits. This model is suitable by its dynamic performance that can model the speech changing characteristics in the time.

Artificial neural networks are computational models with particular properties such as the ability of learning, generalizing and organizing data. Its operation is based on parallel processing as an oversimplification of biological models.

One of the successful research about artificial neural networks is the Neural Prediction Model (NPM) [1]. This model provides discriminant-based learning. Each network is trained to suppress incorrect classification as well to model each class separately. One of the NPM are the Multilayer Perceptrons (MLP). MLP can acquire pattern knowledge by learning, and can then recognize similar patterns to those presented in the learning set.

In the multilayer perceptron, the mapping between the network inputs and outputs is a static function. By this reason, it is necessary to make use of all the available input information up to the current time frame t_c to predict the output [2]. For performing this task it is necessary to have an external memory to make use of all information.

The processing time of back propagation plus an external memory in the NPM can be reduced with a new model: the recurrent neural prediction model (RNPM) [3]. This model is particularly suited for time sequences by its ability to keep an internal memory of the past. The training processing of RNPM only uses back propagation without dynamic programming, so it is a reduction of calculation time.

2 RECURRENT NEURAL PREDICTION MODEL

Recurrent neural prediction model is used for classification of patterns. There is one recurrent neural network to each pattern to recognize. The input of each network is a frame in the time (such as a word), and the output is the next frame in the time. The number of patterns to recognize is equal to the number of networks to train.

The recurrent neural model is created by the addition of recurrent connections to a back propagation network. This will increase the computational capability because the network now becomes a dynamical system that has a temporal evaluation of states. There are several ways to do it: Jordan, Elman, VSRN+ recurrent neural networks.

2.1 Recurrent Neural Networks Models

2.1.1 Elman Model

The Elman network was introduced by Elman in 1990 [4]. Elman's network has connections from the hidden layer to a context layer that also has connections as input to the hidden layer, it is shown in the **Error! Reference source not found.**

The role of the context units is to provide dynamic memory so as to encode the information contained in the sequence of phonemes, necessary to the prediction. Yousef used it to recognize the ten Arabic digits [5].

2.1.1 Jordan Model

Jordan developed a network model capable of computing time series during run-time as the result of a dynamic process. Jordan model emphasizes that processors do not need to store and to retrieve output vector sequences in lined lists or in any other abstract data structure [6]. Jordan's network has connections, shown in the **Error! Reference source not found.**, from the output layer to the context layer. This layer also has connections as input to the hidden layer.

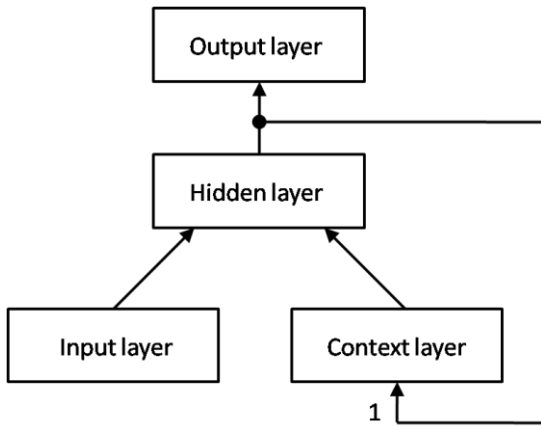


Fig. 1. Simple Recurrent Network. Recurrent link between hidden and context layer.

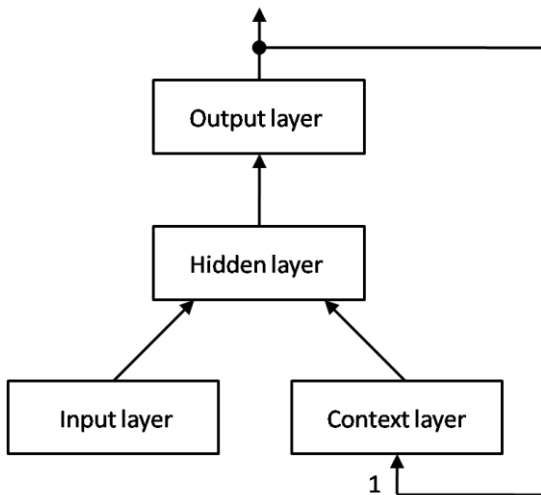


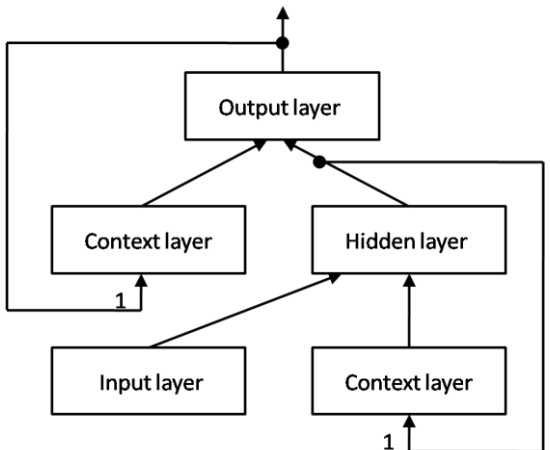
Fig. 2. Jordan Recurrent Network. Recurrent link between output and context layer.

2.1.1 VSRN+ Model

The hidden layer is connected to its context layer, and the output layer is connected to its context layer in this model. Both context layers have input connections to the same layers that feed them. It is shown in the figure 3.

2.1. Training method.

The training method for recurrent networks is similar to the back propagation learning rule, but now we must take also the input of the context layer in the rule. The training method will train the input signal close to output signal to minimize the



prediction error of output layer. This rule involves two phases:

(1) During the first phase the input is propagated forward through the network to compute the output values for each output layer. It is showed in the left side of the Fig. 2. This output is compared with its desired value, resulting in an error signal for each output layer.

(2) The second phase is a backward pass through the network during which the error signal is passed to each layer in the network and appropriate weight changes are calculated. It is showed in the right side of the Fig. 2.

Fig. 3. VSRN+ Recurrent Network. Recurrent link between output and context layer, and between hidden and context layer.

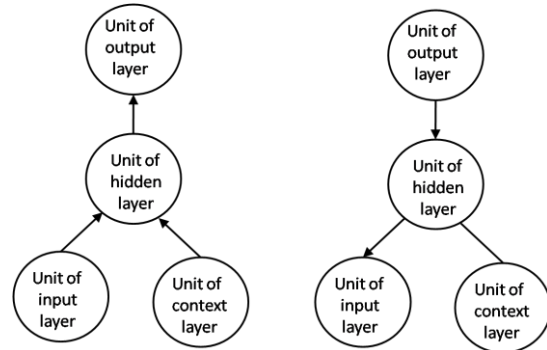
Fig. 2. RNN Training. The arrows show the forward pass and the backward pass in the training.

2.2. Evaluation method.

The digits classification in RNPM shown in Fig. 3, follows the next process:

- (1) Each recurrent neural network was trained to recognize a Spanish digit. N time sequenced feature vectors of length T: a_1, a_2, \dots, a_T are the input of each one of he recurrent neural networks,
- (2) Each recurrent neural network gives a prediction value for each one of the N input vectors: $\hat{a}_1, \hat{a}_2, \dots, \hat{a}_T$. The square difference between the prediction value and the real value gives un prediction error E_c , in the equation 1 for each recurrent neural network.

$$E_c = \frac{1}{2} \sum_{t=1}^T \sum_j^n (a_{t+1,j} - \hat{a}_{t+1,j})^2 \quad (1)$$



- (3) The recurrent neural network that has the smallest prediction error is used to classify the category of the input signal. In this case, the digit to be recognized

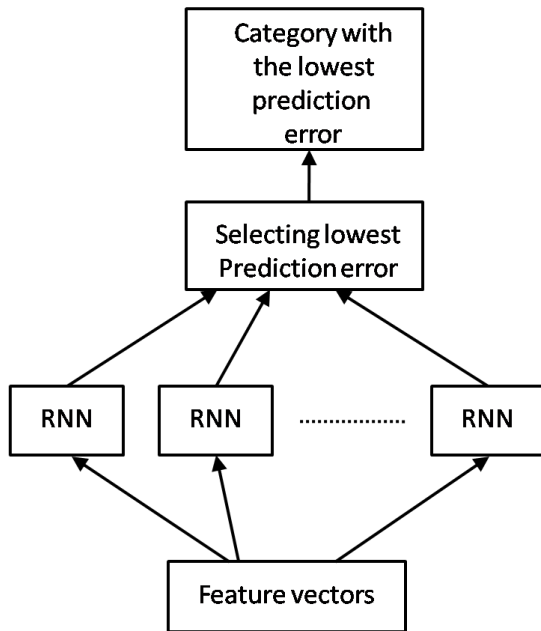


Fig. 3. Evaluating method of digits recognition. The recurrent neural network with lowest prediction error gives the pattern to choose.

3 RESULTS

The speech of the ten spanish digits was recorded with voices from three woman and three men. Each one of them repeated each digit 4 times. The sampling frequency was 8 Khz, this is shown in the Table 1. The results of each recurrent neural network model are showed in the Table 2, with different random number seeds.

Table 1. Voice analysis for digits.

Sampling rate	8 Khz
Window	Blackman-Harris
Window width	5 ms
Frame shift	2.5 ms
Analysis	16th order LPC
Auditory Scale	17 ch Bark scale

Table 2. Results for digits recognition

Seed	Jordan %	SRN %	VSRN+ %
30	96.25	90.00	95.00
31	96.25	93.75	92.50
32	95.00	88.75	93.75
33	96.25	92.50	95.00
34	97.50	91.25	96.25
35	95.00	90.00	93.75
36	93.75	92.50	93.75
37	97.50	90.00	93.75
38	97.50	93.75	95.00

5 CONCLUSIONS

The performance of three recurrent neural networks for the 10 Spanish digits was watched. The best results were obtained with the Jordan model with a medium recognition of 96.11%.

It was proved that the neural prediction model provides less computation time with good results than normal dynamic programming. The Jordan recurrent neural network is a good model of the process of learning of neurons in the brain to recognize Spanish language.

ACKNOWLEDGMENT

The authors wish to thank Conacyt by its support by a grant.

REFERENCES

- [1] Speaker-independent word recognition using a neural prediction model. Watanabe, Ken-ichi Iso and Takao. s.l. : IEEE Proc. ICASSP, 1990, Vol. 8, pp. 441-444.
- [2] Phoneme recognition using time-delay neural network. A. Waibel, T. Hanazawa, G. Hinton and K. Shikano. s.l. : IEEE Transactions Acoustics, Speech, Signal Processing., 1989, Vol. 37, pp. 328-339.
- [3] Recurrent Neural networks. Haruhisha, Takahashi. s.l. : SP93-111, 1993.
- [4] Finding structure in time. Elman, J.L. s.l. : Cognitive Science, 1990, Vol. 14, pp. 179-211.
- [5] Spoken arabic digits recognizer using recurrent neural networks. Ajami, Yousef. s.l. : Proceedings of the 4th IEEE International Symposium, 2004, pp. 195-199.
- [6] Jordan, M.I. Serial Order: a parallel distributed processing approach. Institute for Cognitive Science, Report 8604. University of California, San Diego.
- [7] Japanese digits recognition using recurrent neural prediction model. Uchiyama Toru, Takahashi Haruhisha. s.l. : Tech report of Universidad de Electrocomunicaciones , 1999, Vols. J82-D-II, pp. 1-7.