ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 43-48

Negative Initial Weights Improve Learning in Recurrent Neural Networks

Davor PAVISIC Jean-Philipe DRAYE Roberto TERAN
Gustavo CALDERON Guy CHERON Gaëtan LIBERT

Parallel Information Processing Laboratory
Faculté Polytechnique de Mons – B-7000 Mons (BELGIUM)

Abstract. In this paper we explore the effect that a negative initial weight distribution has on the learning time and learning quality of recurrent neural networks. We shortly introduce the recurrent models used in this research and then we present some experimental results which suggest a dependency of the learning phase with the initial mean of the weight distribution; a frequency domain analysis follows which gives us an idea of the initial weight influence on the frequential characteristics of the temporal sequences generated by recurrent networks. We finally offer a statistical analysis of the neural transformation to show that a negative mean inital weight distribution has, indeed, a positive impact on the network behaviour.

1 Introduction

Recurrent neural networks have shown a capability to exhibit complex dynamical behaviour: fixed point or non-autonomous non-converging dynamics; networks in these latter group have time varying inputs and/or outputs and are more appropriate for adaptative temporal processing such as signal processing or control.

Although, recurrent networks can frequently outperform static networks, they are significantly more difficult and expensive to train. Their learning algorithms usually compute repeatedly the gradient of a cost function with respect to the adaptative parameters of the neural model which tends to be computationally very expensive; moreover, the learning algorithms do not guarantee a global minimum and the system may easily settle in sub-optimal solutions [1]. Techniques to improve learning speed and learning quality are therefore a common field of research in the artificial (recurrent) neural network community.

The technique of weight initialization inspired by Rumelhart et al [7] states that initial weights of exactly zero cannot be used since symmetries in the environment are not sufficient to break the symmetries of the initial weights. Since the publication of their article, the convention on the field has been to initialize weights with an uniform distribution between $-a$ and a with $a \leq 0.5$. Kolen and Pollack have latter remarqued the sensitivity that backpropagation has to the initial conditions to which it is subjected and experimented this by varying the parameter a over a certain range (from 0.1 to 10.0) [5].

In this paper, we show that a slightly negative mean of the initial weight distribution may have a very good influence on the learning speed and quality of recurrent neural networks (independently of the type of initial weight distribution or learning algorithm). This fact was also noticed by Bush et al. [2] in recent experimental and biophysical network. They state that the role of cortical inhibition is more complex than only to oppose excitation; for example, it was suggested that the role of inhibition is to synchronize the firing of groups of pyramidal cells and that excitation and inhibition assume a synergistic action to shape the optimal response of a cortical neuron to a specific stimuli.

2 Recurrent Network Models

We consider recurrent networks governed by the following equations:

$$T_i \frac{dy_i}{dt} = -y_i + F(x_i) + I_i \tag{1}$$

$$F(\alpha) = \frac{1 - \exp^{-\alpha}}{1 + \exp^{-\alpha}} = \tanh\left(\frac{\alpha}{2}\right) \tag{2}$$

$$x_i = \sum_j w_{ji}\, y_j \tag{3}$$

where y_i is the state or activation level of unit i; $F(\alpha)$ is the squashing function defined between -1.0 and $+1.0$; I_i is the bias and x_i is the total or effective input of the neuron. We have considered continuous time networks and have associated to each neuron an adaptative time constant T_i. The error function is defined as a functional:

$$E = \int_{t_0}^{t_1} q(\mathbf{y}(t), t) \cdot dt \tag{4}$$

where t_0 and t_1 define the temporal interval during which the learning process occurs. To train networks governed by equations (1) to (3), we use either the *Real-Time Recurrent Learning* algorithm presented by Williams and Zipser [8] or the *Time-Dependent Recurrent Backpropagation* algorithm derived by Pearlmutter [6].

3 Experimental Results

We present here, from a series of experiments we conducted, a typical application which uses a fully connected dynamic recurrent network with the time dependent recurrent backpropagation learning algorithm to show that, indeed, there might be a dependency

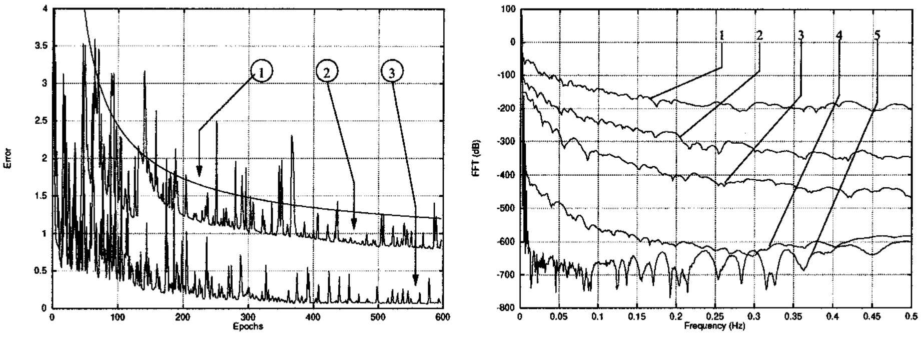

Figure 1: On the left, learning curves 1, 2 and 3 for the first 400 iterates of the Mackey-Glass series with a initial weight mean of $+1.0$, 0.0 and -1.0 respectively. On the right, characterization fo the frequential behaviour of the network spectral analysis. Curves 1, 2, 3, 4 and 5 represent uniform weight distributions of $[-20,-10]$, $[-1,0]$, $[-0.5,+0.5]$, $[0,+1]$ and $[+10,+20]$ respectively.

of the learning process with the initial mean of the weight distribution. We then consider a different learning algorithm, real-time recurrent learning, and two different types of initial weight distributions, uniform and gaussian, to see if the dependency is algorithm and/or weight distribution dependent. Finally we observe the network power spectra with different initial weight-distributions.

The curves on the left graph of figure 1 correspond to the average epochs vs. total error curve, over 10 runs of the learning process for the prediction of the first 400 iterates of the Mackey-Glass series with a 20 neuron network [4]. Each curve corresponds to a Gaussian initial weight distribution with means of: $+1$, 0 and -1 with a standard deviation of 1 in all three cases: 1, 2 and 3 respectively. Note that the best learning curve, 3, which corresponds to the -1 average initial weight distribution, is far below the two other cases during and at the end of the learning process. All other experiences we tried with these particular type of networks gave the same type of result strongly suggesting that there is a benefic influence of a small negative mean distribution of weights on the learning phase of recurrent network. Further experimental essays suggest that this influence is independent of the type of learning algorithm and initial weight distribution.

The graph on the right of figure 1 is one of the various similar results we obtained on frequency domain for different network sizes. It shows five power spectra curves of a 10 neuron recurrent network. Each curve corresponds to uniform weight distributions of: $[-20,-10]$, $[-1,0]$, $[-0.5,+0.5]$, $[0,+1]$ and $[+10,+20]$ (curves 1, 2, 3, 4 and 5 respectively) and is an average of 10 runs in each case. No learning took place. The power spectrum of the output node was computed using a 512-point fast Fourier transform where the maximum detectable period equals 256. Note that in all five cases, the mean amplitude of the frequency response decreases as the weight mean value becomes more positive. This shows that a negative weight mean value allows a recurrent network to exhibit a richer dynamical behaviour whereas a positive weight distribution mean will drive the network quickly to the saturation zones of the sigmoid.

4 Statistical Analysis of the Neural Transformation

To quantify theoretically the effects described in the previous sections, we develop an analytical model that gives the evolution of a macroscopic state variable of the network versus time. This allows to consider the impact of the weight mean value on the evolution of the network mean activity level A_y which will be defined as the mean value of the neurons' activations.

We consider a totally random network where the weights w_{ji} are the realization of independent random variables subject to a gaussian distribution $N(\bar{w}, \sigma_w)$. We can now determine the distribution of the variables x_i from equation (3). These variables are independently, identically and normally distributed with:

$$\bar{x} = E\left[\sum_j w_{ji} y_j\right] = \sum_j y_j \bar{w} = n\bar{w}A_y \qquad (5)$$

$$\sigma_x^2 = \sigma^2\left[\sum_j w_{ji} y_j\right] = \sum_j y_j^2 \sigma^2[w_{ji}] = n\sigma_w^2 A_{y^2} \qquad (6)$$

where $A_y = 1/n \sum y_j$ is the mean activity of the network and $A_{y^2} = 1/n \sum y_j^2$ is the mean power of the network.

We can now determine the distribution of the variables y_i. We will use equation (7) which is the discretized version of equation (1) (see [3]) with a proper time step Δt. Note that here we will limit our study to the effect of the weight distribution and therefore, the time constants T_i will be set to 1.0 and remain unchanged.):

$$y_i(t + \Delta t) = (1 - \Delta t) y_i(t) + \Delta t F(x_i(t)) \qquad (7)$$

The density function f_{sigmoid} of the variables $F(x_i)$, since the distribution of x is given by $N(\bar{x}, \sigma_x)$, can be calculated:

$$f_{\text{sigmoid}}(u) = \frac{1}{\sqrt{2\pi}} \frac{2}{(1+u)(1-u)} \exp\left(-\frac{\left(\ln\left(\frac{1+u}{1-u}\right)\right)^2}{2}\right) \quad \text{with} \quad u \in [0,1] \qquad (8)$$

This density function can be approximated by a Gaussian distribution with:

$$\text{mean:} \quad \left[\frac{1-\exp^{-1}}{1+\exp^{-1}} \cdot \bar{x} + 1\right] \qquad (9)$$

$$\text{variance:} \quad \left[\frac{1-\exp^{-1}}{1+\exp^{-1}}\right] \sigma_x^2 \qquad (10)$$

These results give the distribution of the last term of equation (7). If we replace equations (5) and (6) into (9) and (10) and if we use these parameters in equation (7),

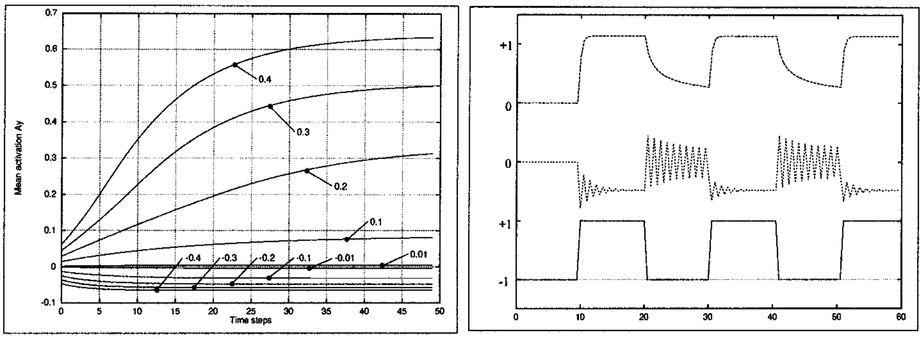

Figure 2: Left, evolution of the mean activity of a 200 neuron network ($A_y = 1/200\sum_j y_j$). The weights have Gaussian distributions with mean vaules ranging from -0.4 to $+0.4$ and a standard deviation of $+1$. Right, Activity of a single recurrent neuron: bottom, stimulation pulse; middle, response of a neuron with negative weights; top, reponse of a neuron with positive weights; for clarity purposes, all three cases have been scaled to a 0 to 1 range and the two outputs have been shifted.

we can approximate the mean activity of the network, A_y, versus time[1]:

$$A_y(t+\Delta t) = (1-\Delta t)A_y(t) + \Delta t\left(\frac{1-\exp^{-1}}{1+\exp^{-1}} n\bar{w}A_y(t) + 1\right) \tag{11}$$

This equation gives the temporal evolution of A_y as a function of the mean value of the weights $\bar{w}$ and the number of neurons n. Figure 2 (left) shows the evolution of the macroscopic variable A_y versus time for a 200 neuron network. Ten different values of $\bar{w}$ are shown in the figure: 0.4, 0.3, 0.2, 0.1, 0.01, −0.01, −0.1, −0.2, −0.3, −0.4. These curves prove that positive weight mean values drive the network towards the saturation zone whereas negative values keep A_y at an acceptable level. This effect can be observed also in figure 2 (right) which shows the evolution of a single recurrent neuron vs time when stimulated with a pulse; the neuron has an external input, a recurrent connection, an activation level of 0 at $t = 0$ and a sigmoid activation function (range $[-1.0 : 1.0]$); the botom curve is the stimulus (input) to the neuron; the middle curve is the response of the neuron (with offset $+2$) when its two weights are set to -1 and the top curve (offset $+3$) is the response of the neuron with its weights equal to 1. Note how the neuron quickly saturates when the weights are positive whereas it presents a complex behaviour around the zero axis when the weights are negative.

5 Conclusion

We have explored an innovative way to improve the learning process in recurrent networks by studying the effects of initial weights selection. By ajusting the mean of the

[1] This equation was developed further to take into account the saturation effect (which was lost in the approximation) of the sigmoid (see [3])

initial weights to a slightly negative value, important improvements in quality and performance were obtained. This fact became significant when various different cases of recurrent networks were tried with the same general results.

By looking at random networks' power spectra, we see that inhibitory weights increase the dynamics of the network but results indicate that too negative values will increase learning time to a point that training would be unfeasible. Looking at random networks' mean activity we see that a positive weight-mean value would quickly saturate the network whereas negative weight-means will keep the network away from the cliping zones of the activation function; the statistical model we developed correctly predicted the mean activity of networks with different sizes and different weight distributions.

We therefore conclude by saying that a negative initial weight mean has a good influence on the speed and quality of learning. We highlighted the fact that this influence is independent of the learning algorithm and of the initial weight distribution type. This paper provides an easy addition to the classical analytical acceleration techniques used for learning in recurrent networks.

References

[1] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is dificult. *IEEE Trans. on Neural Networks*, 5(2):157–166, 1994.

[2] P.C. Bush and T.J. Sejnowski. Effects of inhibition and dendritic saturation in simulated neocortical pyramidal cells. *Journal of Neurophysiology*, 71(6):2183–2193, 1994.

[3] J.P. Draye, D. Pavisic, G. Cheron, and G. Libert. Dynamic recurrent neural networks: a dynamical analysis. *IEEE Transactions on Systems, Man and Cybernetics*. in press.

[4] J.P. Draye, D. Pavisic, G. Cheron, and G. Libert. Adaptive time constants improve the prediction capability of recurrent neural networks. *Neural Processing Letters*, 2:12–16, 1995.

[5] J.F. Kolen and J.B. Pollack. Backpropagation is sensitive to initial conditions. Technical Report TR 90-JK-BPSIC, Laboratory for Artificial Intelligence Research - Ohio State University, 1990.

[6] B.A. Pearlmutter. Learning state space trajectories in recurrent neural networks. *Neural Computation*, 1:263–269, 1989.

[7] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations of the Microstructure of Cognition*, volume 1. Bradford Books, 1986.

[8] R.J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1:270–280, 1989.