ESANN'2007 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2007, d-side publi., ISBN 2-930307-07-2.

Adapting reservoirs to get Gaussian distributions

David Verstraeten[1] and Benjamin Schrauwen[1] and Dirk Stroobandt [1] *

Ghent University - Electronics and Information Systems
Sint Pietersnieuwstraat 41, 9000 Gent - Belgium

Abstract. We present an online adaptation rule for reservoirs that is inspired by Intrinsic Plasiticity (IP). The IP rule maximizes the information content of the reservoir state by adapting it so that the distribution approximates a given target. Here we fix the variance of the target distribution, which results in a Gaussian distribution. We apply the rule to two tasks with quite different temporal and computational characteristics.

1 Introduction

Reservoir Computing is a computational concept that uses a recurrent neural network (the *reservoir*) without adjusting the internal weights. Instead, the reservoir is randomly constructed at the beginning of an experiment, and the weights of an external linear classifier or regression function are trained to generate the desired output using the dynamic response (i.e. the activation levels of the neurons) of the reservoir as input. Many reservoir implementations have been described in literature, but for this contribution we will focus on sigmoid-type reservoirs (Echo State Networks or ESNs) [1].

Various ways of constructing reservoir topologies and weight matrices have been described (see [2] for a brief overview), but for ESNs, one usually creates a network with a certain sparsity and assigns random weights drawn from a normal or uniform distribution or from a discrete set. Next, the weight matrix is globally scaled to set the spectral radius (largest absolute eigenvalue) to a certain value. While Jaeger proposes an optimal spectral radius of around 0.9 for certain small-scale problems, this is not generally applicable [3]. So far, the search for an optimal reservoir remains partly based on experience, and partly on a brute-force search of the parameter space. Moreover, the variance on the performance across different reservoirs with the same spectral radius is still quite substantial which is also undesirable. It is clear that a computationally simple way to adapt the reservoirs to the task at hand without requiring a full parameter sweep or hand tuning based on experience would be welcome.

2 Intrinsic plasticity for tanh activation functions

Intrinsic plasticity (IP) [4] is an unsupervised, bio-inspired, local adaptation technique that adjusts the neuron's excitability in order to maximize the infor-

*David Verstraeten is sponsored by the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT Vlaanderen). Benjamin Schrauwen is sponsored by the FWO Flanders project G.0317.05.

mation content in the neuron output, given certain constraints. Here, information content is expressed as the mutual information between the distribution of the neuron input $p_x(x)$ and the distribution $p_y(y)$ of the neuron activation level:

$$I(p_y, p_x) = H(p_y) - H(p_y|p_x) \tag{1}$$

Here, $H(p_y(y))$ is the entropy or 'uncertainty' of the variable y and $H(p_y|p_x)$ can be interpreted as a noise factor inside the neuron. If we assume the latter term to be constant, maximizing the mutual information reduces to maximizing the entropy of the variable y:

$$H_{\max}(p_y) = \max_p \sum_{i=1}^{n} p_y(y_i) \log(p_y(y_i)), \tag{2}$$

where n is the number of observations of y.

The activation function is tuned so that the distribution of the neuron's activation approximates a desired distribution. The similarity between the desired theoretical distribution f_t and the effective empirical distribution f_y is expressed in terms of the Kullback-Leibler divergence:

$$D_{KL}(f_y, f_t) = \int f_y \log\left(\frac{f_y}{f_t}\right) dy. \tag{3}$$

As mentioned before, IP tries to maximize the information content. Given the constraints, certain distributions maximize the entropy of a stochastic variable. In [5], IP is derived for neurons with a Fermi-type activation function $y_{\text{fermi}} = \frac{1}{1+e^{-ax-b}}$, using the biologically inspired constraint that the neuron tries to maximize information while bounding the metabolical requirements, which means that the mean activation level is set to a desired value. Among all positive distributions with a fixed mean, the exponential distribution is the one with the maximum entropy. Thus, IP will try to drive the distribution of the neuron's activation levels towards a desired distribution by minimizing the Kullback-Leibler distance between the actual and the desired distributions.

When applied to Reservoir Computing, IP has been shown to increase performance on a variety of benchmark problems [5]. However, previous work on IP focuses on the combination of the Fermi activation function and the exponential output distribution, while traditionally analog reservoirs are built from tanh-type neurons. This is not a significant issue, since both activation functions can easily be transformed into each other using $\tanh(x) = 2y_{\text{fermi}}(2x) - 1$. However, while the exponential target distribution is a direct consequence of the biologically plausible constraint on the mean, it might be beneficial to abandon this constraint. Indeed, if we fix the desired variance instead of the mean, the maximum-entropy output distribution is the Gaussian.

Another possible advantage of using Gaussian IP, is the fact that the dynamic evolution of the adapted reservoir might result in a Gaussian process. If this is the case, it would mean that the theoretical results on Bayesian classifiers and regressors that operate on Gaussian processes are applicable, such as the

fact that the confidence of a classification can be evaluated immediately and accurately. However, further research is needed to substantiate this.

There is another difference between fermi neurons adapted with IP and tanh neurons adapted with Gaussian IP: in the former case, if one wants to have sparse activity (meaning an exponential output distribution), the output activation has to be close to zero. This is mainly caused by the bias of the fermi function being afjusted. For these fermi neurons, this means they operate in a highly non-linear regime. The tanh non-linearity, on the other hand, is mainly active in the linear regime (around zero) due to the Gaussian distribution. Thus, previous research regarding the spectral radius of the connection matrix is not applicable for fermi-type activation functions once they have been adapted using IP, while it is much more relevant in the case of the tanh neurons.

3 Deriving the learning rule

To adjust the IP learning rule for tanh activation functions and a Gaussian target distribution, we formulate the following minimization problem:

$$f_{y,\text{opt}} = \arg\min_{f_y} D_{KL}(f_y, f_t), \tag{4}$$

where f_t is the Gaussian distribution. For the Kullback-Leibler divergence in this case this yields :

$$D_{KL}(f_y, f_t) = \int f_y \log\left(\frac{f_y}{\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(y-\mu)^2}{2\sigma^2}}}\right) dy \tag{5}$$

$$= \int f_y \log(f_y) dy + \int f_y \frac{y^2}{2\sigma^2} dy - \int f_y \frac{2\mu y}{2\sigma^2} dy + C \tag{6}$$

$$= E\left(\log(f_x) - \log\left(\frac{\partial y}{\partial x}\right) + \frac{1}{2\sigma^2}y^2 - \frac{\mu}{\sigma^2}y\right) + C, \tag{7}$$

where C are constant terms, and the relation $f_y \partial y = f_x \partial x$ is used.

We use a generalised activation function $y_i = \tanh(a_i x_i + b_i)$, where x_i is the total weighted input to the neuron i, a_i the local slope of the sigmoid and b_i the local bias. We use stochastic gradient descent for the a and b parameters as follows:

$$\frac{\partial D_{KL}}{\partial a} = E\left(-\frac{\mu x}{\sigma^2} + \frac{xy}{\sigma^2}(2\sigma^2 + 1 - y^2 + \mu y) - \frac{1}{a}\right) \tag{8}$$

$$\frac{\partial D_{KL}}{\partial b} = E\left(-\frac{\mu}{\sigma^2} + \frac{y}{\sigma^2}(2\sigma^2 + 1 - y^2 + \mu y)\right) \tag{9}$$

From this, we get the following online learning rule using a learning rate η:

$$\Delta b = -\eta \left(-\frac{\mu}{\sigma^2} + \frac{y}{\sigma^2}(2\sigma^2 + 1 - y^2 + \mu y) \right) \tag{10}$$

$$\Delta a = \frac{\eta}{a} + \Delta b x \tag{11}$$

Thus, we now have an online reservoir adaptation rule inspired by intrinsic plasticity, for tanh activation functions and a Gaussian target distribution.

4 Experimental results

We investigated the influence of Gaussian IP for two different benchmark tests with quite different computational and temporal properties. The first is a tenth-order NARMA system identification task, where the readout is trained to model the following system:

$$y(k+1) = 0.3y(k) + 0.05y(k)\left[\sum_{i=0}^{9} y(k-i)\right] + 1.5u(k-9)u(k) + 0.1 \tag{12}$$

where in both testing and training the input $u(k)$ are drawn from a uniform distribution over [0 0.5]. [1]. This task is rather difficult since it is highly nonlinear and it requires a substantial amount of memory to accurately reproduce the output. The performance is measured as normalized root mean square error (NRMSE), using tenfold crossvalidation over the training set consisting of 10 examples of 1100 timesteps each, of which the first 100 are used to warm up the reservoir and were not regarded for the error measure.

For these experiments, we first constructed a reservoir of 100 neurons with a given spectral radius and sparse connectivity of .1, without output feedback, and evaluated the performance on the benchmark tests. Next, we applied Gaussian IP during 5 epochs (i.e. every example of the dataset was presented five times) with a learning rate of $\eta = 0.001$ in order to determine the optimal slopes a and biases b for each neuron. Finally, we re-evaluated the reservoirs on the same task.

Figure 1 shows the effect that Gaussian IP has on the distribution of the activations of the neurons in the reservoir. Before Gaussian IP, the histogram of the reservoir activation values is very spiky and irregular, while after only five epochs of Gaussian IP the activation values nicely follow a Gaussian distribution.

Figure 2 shows the performance on the NARMA task as a function of the desired variance of the output distribution is (left, with Gaussian IP) and the spectral radius (right, without Gaussian IP). Note that both figures have the same minimal error. Further, when we regard both variables as tunable parameters, the figures show that Gaussian IP reduces the dependence of the performance on the tuning parameter. More importantly, the results also show that for an

[1] Only $u(k)$ is supplied as input to the reservoir, and not $u(k-9)$.

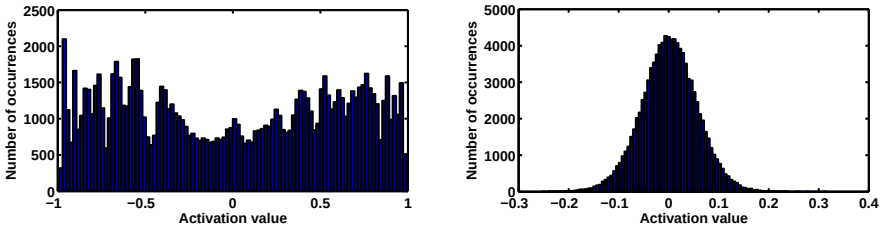

Fig. 1: Histograms of activation values of a reservoir of 100 neurons for a uniform random input before (left) and after (right) 5 epochs of Gaussian IP.

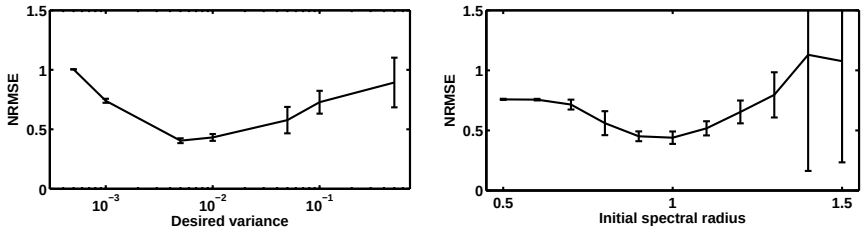

Fig. 2: Performance on the narma task as a function of desired variance (using Gaussian IP) and spectral radius (without Gaussian IP).

individual parameter setting, the variation of the performance is also reduced when using Gaussian IP. Thus, far less random reservoirs need to be constructed to accurately evaluate certain reservoir settings.

The second task is speech recognition of isolated digits, where the training set consists of 500 samples spoken by five different female speakers. Due to space limitations we refer to [2] for more details on both benchmark tests. We have run similar experiments as for the NARMA case, and we can again conclude that the optimal values lie very close to each other but that the variance on the performance is smaller for the Gaussian IP case : we get an error of $2.04 \pm .91$ by setting the optimal spectral radius of 1.2, and $2.42 \pm .6$ by using Gaussian IP with a desired variance of .1.

By changing the slope a of each neuron, the effective spectral radius of the connection matrix is also changed, since the total input to each neuron is scaled. Thus, in order to compute the effective spectral radius ρ_{eff}, we need to multiply every row of the connection matrix with the slope of the corresponding neuron. Figure 3 evaluates the influence of the desired variance on the effective spectral radius. The figure on the left compares ρ_{eff} for both a traditional, random topology and a ring-shaped 1D lattice, where each neuron is only connected to its eight nearest neighbours and only one neuron receives the external input. The latter topology is very specific, and as such the spectral radius is useless as a global scaling measure: when increasing the spectral radius, the behavior

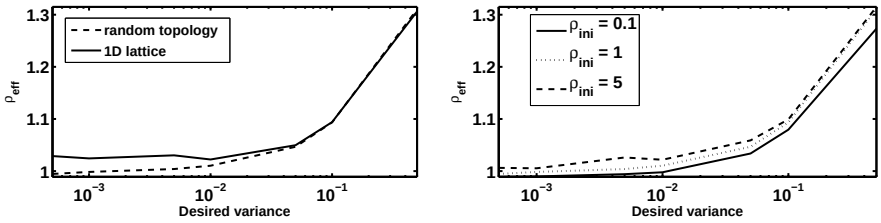

Fig. 3: Effective spectral radius vs. desired function, for a random and 1D lattice topology (left), and for a random topology with different initial spectral radii (right).

of these reservoirs abruptly switches between only one active neuron and the whole network exhibiting chaotic, oscillatory behavior. Nonetheless, the results show that Gaussian IP has similar effects on the effective spectral radius of both topologies, and we have found that with Gaussian IP useful dynamics can always be obtained. Both figures show a strongly nonlinear relation between the desired variance and ρ_{eff} which means that the former is not just a reformulated measure of the latter. Also, note that ρ_{eff} depends only slightly on the initial spectral radius as is apparent from the figure on the right, but this might be due to the fact that the rule hasn't fully converged yet.

5 Conclusions and future work

We have presented and derived an online adaptation rule for reservoirs that maximizes the information content of the reservoir states based on a constraint on the variance of the state distribution. This results in normally distributed reservoir states, perhaps yielding a Gaussian process. We have evaluated reservoir performance and effective spectral radius on two benchmark tests with very different characteristics. Future work includes a thorough investigation of the influence of IP on the reservoir dynamics, the search for a way to decrease the influence of the learning rate on the convergence and further research into the applicability of theoretical results from the field of Gaussian processes.

References

[1] H. Jaeger. The "echo state" approach to analysing and training recurrent neural networks. Technical Report GMD Report 148, German National Research Center for Information Technology, 2001.

[2] David Verstraeten, Benjamin Schrauwen, Michiel D'Haene, and Dirk Stroobandt. A unifying comparison of reservoir computing methods. *Neural Networks*, 2007. submitted.

[3] D. Verstraeten, B. Schrauwen, and D. Stroobandt. Reservoir-based techniques for speech recognition. In *Proceedings of WCCI*, 2006.

[4] J. Triesch. Synergies between intrinsic and synaptic plasticity mechanisms. *Neural Computation*, in press, 2007.

[5] Jochen Steil. Online reservoir adaptation by intrinsic plasticity for backpropagation-decorrelation and echo state learning. *Submitted to Neural Networks: special issue on Reservoir Computing*, 2006.