Leaky Echo State Network

Sergio Peignier

May 3, 2017

1 Definitions

Echo State Networks (ESN) are a family of Recurrent Neural Networks (RNN) initially created by Herbert Jaeger and Wolfgang Maass in 2002 [Jaeger, 2003, Jaeger et al., 2007, Jaeger, 2007, Lukoševičius, 2012, Jaeger, 2002]. ESN are used as supervised learning algorithms. These networks are constituted by:

- A large, random and immutable neural network, called the "reservoir". Each one of the N neurons from the reservoir receives input signals and produces high-dimensional nonlinear transformed versions (i.e., *Echo States*).
- A second layer produces a desired output by performing a linear (or even a nonlinear) combination of the reservoir signals. This layer can also send information back to the reservoir (feedback).
- Only the weights between the reservoir and the output are trainable.

Let the vector x denote the states of the neurons belonging to the reservoir, let y denote the feedback vector received from the output layer and let the matrices W^{in}, W and W^{fb} denote respectively the synaptic weights between the inputs and the reservoir neurons, between the reservoir neurons themselves and between the output layer and the reservoir (feedback signal).

The state of the reservoir neurons x is governed by the following ODE:

$$\frac{dx}{dt} = \frac{1}{\tau}(-a \times x + f(W^{in} \cdot u + W \cdot x + W^{fb} \cdot y))$$

- $\tau > 0$ corresponds to a global time constant.
- $a > 0$ is a parameter called leaking rate.
- f is the activation function used for the reservoir neurons.

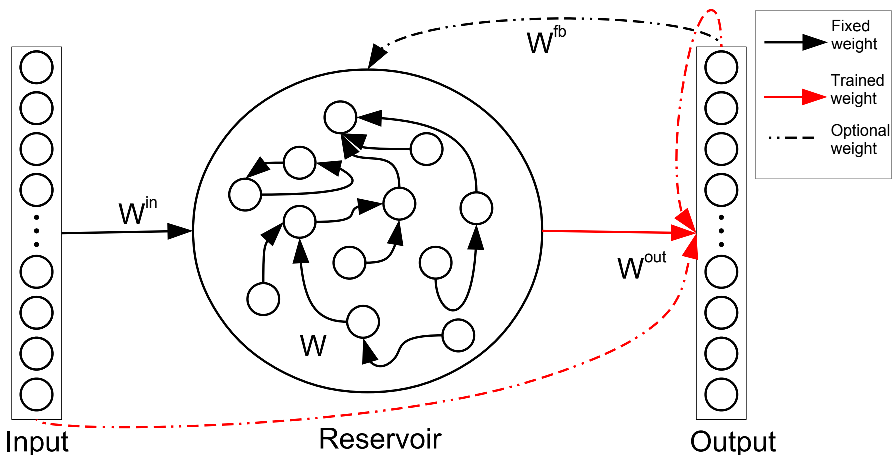

Figure 1: Echo State Network general scheme

The output of the ESN it computes using the following equation:

$$y = f^{out}(W^{out} \cdot x)$$

f^{out} corresponds to the transfer function and W^{out} is the transfer matrix (only this matrix is trained). Some variants of the output equation exist. For instance, $y = f^{out}(W^{out} \cdot [u; x])$ or $y = f^{out}(W^{out} \cdot [1; u; x])$, where $[a; b]$ corresponds to the concatenation of vectors a and b.

- Using the Euler integration formula, write the equation giving the state of the system at time $t + \delta t$ from the values of the system at time t.
- The discretization rate δt is called the "stepsize". Give the intuition about this parameter. What happens if δt is too small or too high? Instead of considering τ and δt separately we will consider $\gamma = \frac{\delta t}{\tau}$. The system is under the constrain $a\gamma \leq 1$.
- Which kind of function would you use for f?

2 Sine wave generator

In this example an ESN is used to generate sine waves, and subsequently the main parameters of this algorithm are analyzed. The objective function is defined as (taking n integer between 0 and 1000):

$$y^{target}(n) = 0.5 \times sin(n * 0.2)$$

- Build a reservoir neural network.
  - $N = 20$ neurons
  - The elements of W^{fb} are i.i.d and follow a uniform distribution between -0.5 and 0.5.
  - The elements of W are i.i.d and follow a uniform distribution between $-b$ and b (here we take $b = 0.2$).
  - This ESN does not receive inputs, and only receives the feedback signal.
  - The tanh function computes the activations of the reservoir neurons.
  - The transfer function used here is the identity function.
- Visualize the temporal evolution of the states of the neurons from the reservoir without any feed back input.
- Iterate the ESN 100 times (with feedback inputs, forcing $y(n) \leftarrow y^{target}(n)$).
- Iterate 200 times the ESN and save at each step the state of the network in a matrix (one row per record) and also save in a different vector the corresponding values of the target function. During the training, the ESN receives the feedback signal from y^{target} ($y(n) \leftarrow y^{target}(n)$). let Z and Y denote the matrix and the vector keeping track of the previous results.
- Compute the (trained) matrix W^{out} using the pseudo-inverse of Z:

  $$W^{out} = Z^{-1} \cdot Y$$

- Take an unknown sequence of length 300 (test set), run the algorithm with $y(n) \leftarrow y^{target}(n)$ for a few iterations and then let the system use its own signal $y(n)$. Plot the target signal and the signal found by the ESN.

3 Parameters

- In this section we assess the quality of the results through the mean sum of square distances between the predicted values $y(n)$ and the target values $y^{target}(n)$ for all n in the test set.

Leaking Rate and time constant

- Try different values for the Leaking Rate a and the time constant γ, for different objective functions (change the period of the sine function). What does this parameter control?

Spectral radius As we already saw during the first practical course, that the reservoir system can exhibit a chaotic behavior (the state of the neurons can oscillate, each neuron may reach a different optimum, ...). In order to use such a neural network in a ESN, we need to avoid this kind of behavior. In practice, there exist a procedure that seems to guarantee this for ESN.

- Generate a random internal weight matrix W.
- Normalize W by a factor $\frac{\alpha}{|\lambda_{max}|}$. Where $|\lambda_{max}|$ is the spectral radius of W (the largest eigen value of W). So λ is the new spectral radius of W. Test different values for α for different objective functions (change the period of the sine). What does this parameter control?

Number of neurons

- Test different values for N (number of neurons in the reservoir). What does this parameter control?

4 Tunable sine wave generator

In this case the period of the sine function that is used as an objective function $y^{target} = 0.5 \times sin(n \times u(n))$ changes according to function $u(n) = 0.2 \times sin(n \times 0.001)$. In this case the ESN received the values of $u(n)$ as an input. Repeat the previous operations with this new problem.

References

[Jaeger, 2002] Jaeger, H. (2002). *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the" echo state network" approach*, volume 5. GMD-Forschungszentrum Informationstechnik.

[Jaeger, 2003] Jaeger, H. (2003). Adaptive nonlinear system identification with echo state networks. *networks*, 8(9):17.

[Jaeger, 2007] Jaeger, H. (2007). Echo state network. *Scholarpedia*, 2(9):2330. revision #151757.

[Jaeger et al., 2007] Jaeger, H., Lukoševičius, M., Popovici, D., and Siewert, U. (2007). Optimization and applications of echo state networks with leaky-integrator neurons. *Neural networks*, 20(3):335–352.

[Lukoševičius, 2012] Lukoševičius, M. (2012). A practical guide to applying echo state networks. In *Neural networks: Tricks of the trade*, pages 659–686. Springer.