2014 International Joint Conference on Neural Networks
(World Congress on Computational Intelligence)

An Effective Criterion for Pruning Reservoir's Connections in ESNs

Scardapane S., Nocco G., Comminiello D., Scarpiniti M. and Uncini A.

intelligent signal processing
and multimedia lab

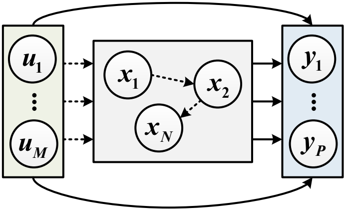
WCCI 2014

Table of Contents

What is an Echo State Network?

An *Echo State Network* (ESN) is a *Recurrent Neural Network* (RNN), whose processing is partitioned in two components [LJ09]:

- A recurrent reservoir, which is *fixed in the beginning*.
- A *static* readout, trained using linear regression techniques.



ESN Update

Given an input $\mathbf{u}(n)$, the internal state $\mathbf{x}(n)$ of the reservoir's neurons is computed as:

$$\mathbf{x}(n) = \tanh(\mathbf{W}_r^r \mathbf{x}(n-1) + \mathbf{W}_i^r \mathbf{u}(n)) \tag{1}$$

where $\mathbf{W}_r^r$ and $\mathbf{W}_i^r$ are *randomly* generated (albeit with some caveats [LJ09]). Similarly, the output is computed as:

$$y(n) = \mathbf{w}_r^o \mathbf{x}(n) + \mathbf{w}_i^o \mathbf{u}(n) \tag{2}$$

where $\mathbf{w}_r^o$ and $\mathbf{w}_i^o$ must be learned. We also define the "extended" state $\mathbf{s}(n) = \left[\mathbf{u}(n)^T \; \mathbf{x}(n)^T\right]^T$.

ESN Training

1. We fed the network with a sequence of inputs $\{\mathbf{u}(1), \ldots, \mathbf{u}(S)\}$ and compute the corresponding states.
2. We concatenate the states in the matrix $\mathbf{A} = [\mathbf{s}(1), \ldots, \mathbf{s}(S)]$, and the desired outputs in $\mathbf{d} = [d(1), \ldots, d(S)]$.
3. In batch mode, we can compute the optimal output weights as:

$$\mathbf{W} = \left(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I}\right)^{-1}\mathbf{A}^T\mathbf{d} \tag{3}$$

where $\mathbf{I}$ is the identity matrix and λ a suitable positive scalar term. In *online learning*, we can use the *Recursive Least-Square* (RLS) algorithm [Jae03].

Online learning for ESNs

At time instant n, the error is computed as:

$$e(n) = d(n) - \mathbf{W}(n-1)\mathbf{s}(n)$$

Denoting by λ the *forgetting factor* of the filter, a *gain vector* is computed as:

$$\mathbf{g}(n) = \mathbf{P}(n-1)\mathbf{s}(n)\left\{\lambda + \mathbf{s}^T(n)\mathbf{P}(n-1)\mathbf{s}(n)\right\}^{-1}$$

where $\mathbf{P}(n-1)$ is updated as:

$$\mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \mathbf{g}(n)\mathbf{s}^T(n)\lambda^{-1}\mathbf{P}(n-1)$$

Finally, the weights are updated according to:

$$\mathbf{W}(n) = \mathbf{W}(n-1) + e(n)\mathbf{g}(n)$$

Characteristics of a Reservoir

A reservoir must possess the following properties:

1. It must be *stable*. This is expressed in term of the so-called *echo state property* [LJ09].
2. It should be "big enough" to ensure optimal generalization capabilities.
3. Additionally, its connections (*synapses*) should be sparse to have heterogeneous states.

Few research has been devoted to the last problem, of obtaining optimally sparse reservoirs in an efficient manner [DSV+09].

Significance of a Synapse

We define the *significance* of a synapse at time-instant n as:

$$s_{ij}(n) = \frac{1}{T} \sum_{z=n-T}^{n} \frac{(x_i(z-1) - \hat{\mu}_x)(x_j(z) - \hat{\mu}_x)}{\hat{\sigma}_x^2} \tag{4}$$

Using it, the probability of removing a given synapse is provided by:

$$p_{ij}(n) = \exp\left\{ -\frac{|s_{ij}(n)|}{t(n)} \right\} \tag{5}$$

where $t(n)$ is monotonically decreasing in n. This is inspired to the *Simulated Annealing* optimization algorithm. In our experiments, the exponential profile for $t(n)$ worked very well:

$$t(n) = \alpha^{(n/Q)-1} t_0 \tag{6}$$

NARMA system

On the 10-th order NARMA system, we achieve a 52% decrease in the synapses, *without altering the performance*:

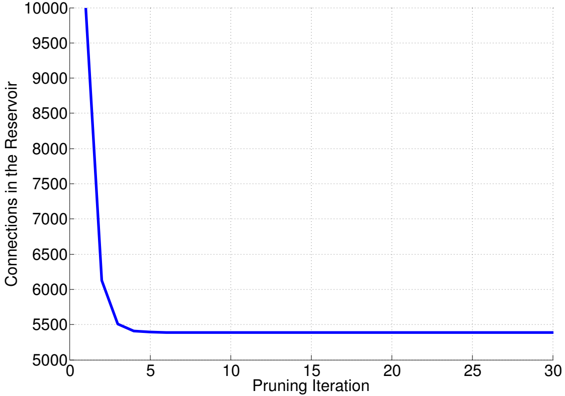
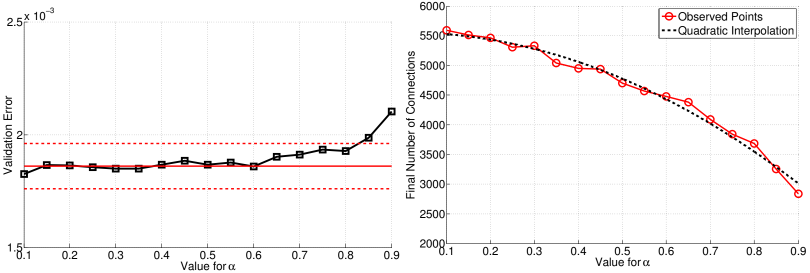


Figure : Evolution of the reservoir density after each pruning operation.

Robustness

The algorithm is robust to a change of the scaling factor α of the temperature profile:



(a) Validation error (b) Resulting number of connections

Figure : Resulting number of connections and average validation error for each choice of the scaling factor.

Extendend Polynomial

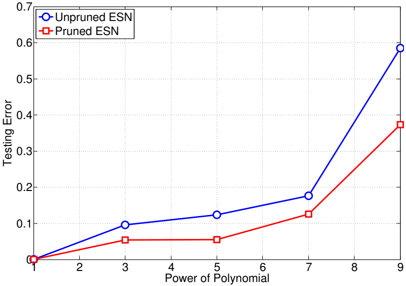
We consider the extended polynomial detailed in [BVS+13]. The input is a random number extracted from an uniform distribution over $[-1, +1]$. The output is given by:

$$d(n) = \sum_{i=0}^{p} \sum_{j=0}^{p-i} c_{ij} u^i(n) u^j(n-d) \tag{7}$$

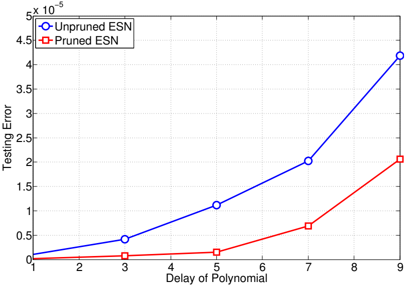
where the coefficients c_{ij} are randomly distributed over the same distribution as the input data, and the two parameters p and d control the requirements of the task in term of memory and non-linearity.

Mean-Squared error

The algorithm is robust to a change in the level of memory and non-linearity of the task:



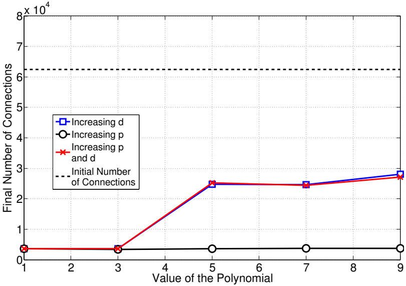
(a) Increasing non-linearity



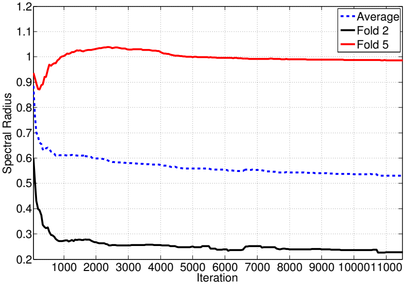
(b) Increasing memory

Figure : MSE of the networks for increasing memory and non-linearity of the task.

Other results



(a) Final number of synapses



(b) Spectral radius

Figure : On the left: final number of connections. On the right: evolution of the spectral radius for three representative folds.

Conclusions and Future Works

We presented an algorithm for deleting connections in a randomly generated reservoir of an Echo State Network (ESN). The resulting ESN has always higher or comparable performance, but a smaller density of connections, resulting in lower computational and hardware requirements.

We are working on combining our pruning strategy with other existing approaches. Moreover, we are interested in developing additional criteria, possibly leading to the direct pruning of neurons, or which are based on small neighborhoods of each synapse. Finally, we aim at proving some strict criteria to ensure the echo state property, and to fully automate the final pruning process.

References

J.B. Butcher, D. Verstraeten, B. Schrauwen, C.R. Day, and P.W. Haycock.
Reservoir computing and extreme learning machines for non-linear time-series data analysis.
Neural networks, 38:76–89, February 2013.

X. Dutoit, B. Schrauwen, J. Van Campenhout, D. Stroobandt, H. Van Brussel, and M. Nuttin.
Pruning and regularization in reservoir computing.
Neurocomputing, 72(7-9):1534–1546, March 2009.

H. Jaeger.
Adaptive nonlinear system identification with echo state networks.
In *Advances in Neural Information Processing Systems*, 2003.

M. Lukoševičius and H. Jaeger.
Reservoir computing approaches to recurrent neural network training.
Computer Science Review, 3(3):127–149, 2009.