A novel Echo State Network algorithm

Aleix Espuña Fontcuberta and Saikat Chatterjee

June 8, 2020

This documentation presents a novel variation of the classical Echo State Network (ESN) algorithm, which is used to predict signal continuations.

Equations

The novel algorithm we are presenting takes a time series of M scalars $Y = \{y_0, y_1 ... y_{M-1}\}$ and it tries to make a jump τ in the future and predict q values $\{\hat{y}_{M+\tau-q+1}, \hat{y}_{M+\tau-q+2} ... \hat{y}_{M+\tau}\}$, where $\tau \geq q-1$. We use an Echo State Network (ESN) architecture: a neural network made of a large reservoir of N neurons ($N \approx 1000$) whose state $\mathbf{x}_n \in \mathcal{R}^N$ evolves in time. The equation that updates a state $\mathbf{x}_n$ to a new state $\mathbf{x}_{n+1}$ is the following:

$$\mathbf{x}_{n+1} = \tanh(W_{res}\mathbf{x}_n + W_{fb}\mathbf{y}_n^p) \tag{1}$$

The matrix $W_{res} \in \mathcal{R}^{N\times N}$ contains all the inner connections of the reservoir. It must be sparse, to speed up computation, and with a spectral radius lower than 1. Its non-zero values remain fixed and they must be initialized from a uniform or normal random distribution between $[-1, 1]$. The feedback matrix $W_{fb} \in \mathcal{R}^{N\times p}$ contains fixed random weights between $[-1, 1]$ (uniform or normal distribution) connecting the vector $\mathbf{y}_n^p \in \mathcal{R}^p$ to the reservoir. The spectral radius of the product by its transpose $W_{fb}^T W_{fb}$ must be lower than 1, which helps to provide stability to the update equation (not guaranteed). Finally, the vector $\mathbf{y}_n^p \equiv \{y_n, y_{n-1}, y_{n-2}, ... y_{n-(p-1)}\}$, with $n \geq p-1$, contains the present value y_n and its history of size $p-1$.

1 Teacher forcing

The goal of teacher forcing is to find a prediction function capable to map a state $\mathbf{x}_n$ to a vector $\mathbf{y}_{n+\tau}^q \equiv \{y_{n+\tau}, y_{n+\tau-1}, y_{n+\tau-2}, ... y_{n+\tau-(q-1)}\}$, with $\tau \geq q-1$. We will do it by choosing a linear predictor $h(\mathbf{x}_n) = \hat{\mathbf{y}}_{n+\tau}^q = W_{out}\mathbf{x}_n + \mathbf{b}$. The matrix $W_{out} \in \mathcal{R}^{q\times N}$ and the intercept $\mathbf{b} \in \mathcal{R}^q$ are found by solving a ridge regression:

$$W_{out}, \mathbf{b} = argmin\Big(\sum_{n=p}^{M-\tau-1} \big(||\mathbf{y}_{n+\tau}^q - \hat{\mathbf{y}}_{n+\tau}^q||^2\big) + \beta\big(||W_{out}||^2 + ||\mathbf{b}||^2\big)\Big) \tag{2}$$

To be able to solve Eq.(2) we must first generate all states $\mathbf{x}_n$ from $n = p$ to $n = M - \tau - 1$. This is done by initializing $\mathbf{x}_{p-1} = 0$ and running Eq.(1) from $n = p - 1$ to $n = M - \tau - 1$. It is convenient however, to run Eq.(1) until we obtain the final state $\mathbf{x}_M$, as we will use it to predict $\hat{\mathbf{y}}^q_{M+\tau}$ once Eq.(2) has been solved.

2 Predicting

Once we have generated the final state $\mathbf{x}_M$ and also solved Eq.(2), we can obtain the set of predicted values $\hat{\mathbf{y}}^q_{M+\tau} = \{\hat{y}_{M+\tau}, \hat{y}_{M+\tau-1}, \hat{y}_{M+\tau-2}, ...\hat{y}_{M+\tau-(q-1)}\}$ by just computing $h(\mathbf{x}_M) = \hat{\mathbf{y}}^q_{M+\tau} = W_{out}\mathbf{x}_M + \mathbf{b}$.

3 Algorithm overview

Training data:

- The training time series $Y = \{y_0, y_1...y_{M-1}\}$.

Goal:

- Predict the values $\hat{Y} = \{\hat{y}_{M+\tau-q+1}, \hat{y}_{M+\tau-q+2}...\hat{y}_{M+\tau}\}$

Input parameters:

- The time series history p sent to the reservoir.
- The jump in the future τ.
- The history q of the prediction.
- The regularization β.

Restrictions:

- $\tau \geq q - 1$
- $p + \tau \leq M - 1$.
- Matrix $W_{res} \in \mathcal{R}^{N\times N}$ must be sparse (e.g. 10 non-zero values per row)
- $\rho(W_{res}) < 1$ (Spectral radius)
- $\rho(W^T_{fb}W_{fb}) < 1$, with $W_{fb} \in \mathcal{R}^{N\times p}$
- W_{res}, W_{fb} initialized from a uniform or random distribution $\in [-1, 1]$

Definitions:

- $\mathbf{y}^p_n \equiv \{y_n, y_{n-1}, y_{n-2}, ...y_{n-(p-1)}\}$
- $\mathbf{y}^q_{n+\tau} \equiv \{y_{n+\tau}, y_{n+\tau-1}, y_{n+\tau-2}, ...y_{n+\tau-(q-1)}\}$

- $\hat{\mathbf{y}}^q_{n+\tau} = W_{out}\mathbf{x}_n + \mathbf{b}$; with $W_{out} \in \mathcal{R}^{q\times N}$ and $\mathbf{b} \in \mathcal{R}^q$.

Compute:

1. Initialize $\mathbf{x}_{p-1} = 0$, W_{fb}, W_{res}.
2. From $n = p - 1$ to $n = M - 1$, run: $\mathbf{x}_{n+1} = \tanh(W_{res}\mathbf{x}_n + W_{fb}\mathbf{y}^p_n)$
3. Solve: $W_{out}, \mathbf{b} = argmin\Big(\sum_{n=p}^{M-\tau-1} ||\mathbf{y}^q_{n+\tau} - \hat{\mathbf{y}}^q_{n+\tau}||^2 + \beta(||W_{out}||^2 + ||\mathbf{b}||^2)\Big)$
4. Evaluate $\hat{\mathbf{y}}^q_{M+\tau} = W_{out}\mathbf{x}_M + \mathbf{b}$

The vector $\hat{\mathbf{y}}^q_{M+\tau}$ contains the sequence $\hat{Y}$ in reversed order.