ESN course

The reservoir is defined by the tuple (Win, W, α).

The defining global parameters of the reservoir are: the size Nx, sparsity, distribution of nonzero elements, and spectral radius of W; scaling(-s) of Win; and the leaking rate α.

The reservoir can be too big only when the task is trivial and there is not enough data available T < 1+Nu +Nx.

**The size Nx**

Select global parameters with smaller reservoirs, then scale to bigger ones.

Nx should be at least equal to the estimate of independent real values the reservoir has to remember from the input to solve its task. (For i.i.d. inputs u(n), this estimate is Nu times a rough estimate of how many time steps the inputs should be remembered to solve the task.)

**Sparsity of the Reservoir**

In general, sparsity of the reservoir does not affect the performance much and this parameter has a low priority to be optimized.

Connect each reservoir node to a small fixed number of other nodes (e.g., 10) on average, irrespective of the reservoir size. Exploit this reservoir sparsity to speedup computation.

**Distribution of Nonzero Elements**

We usually prefer a uniform distribution for its continuity of values and boundedness. Gaussian distributions are also popular. Both distributions give virtually the same performance.

**Spectral Radius \***

Typically, a random sparse W is generated; its spectral radius ρ(W) is computed; then W is divided by ρ(W) to yield a matrix with a unit spectral radius; this is then conveniently scaled with the ultimate spectral radius to be determined in a tuning procedure.

For the ESN approach to work, the reservoir should satisfy the so-called echo state property: the state of the reservoir x(n) should be uniquely defined by the fading history of the input u(n) [6]. In other words, for a long enough input u(n), the reservoir state x(n) should not depend on the initial conditions that were before the input.

Large ρ(W) values can lead to reservoirs hosting multiple fixed point, periodic, or even chaotic (when sufficient nonlinearity in the reservoir is reached) spontaneous attractor modes, violating the echo state property.

ρ(W) < 1 ensures echo state property in most situations. Even though it is possible to violate the echo state property even with ρ(W) < 1, this is unlikely to happen in practice. More importantly, the echo state property often holds for ρ(W) ≥ 1 for nonzero inputs u(n).

In practice ρ(W) should be selected to maximize the performance, with the value 1 serving as an initial reference point.

As a guiding principle, ρ(W) should be set greater for tasks where a more extensive history of the input is required to perform it, and smaller for tasks where the current output y(n) depends more on the recent history of u(n). The spectral radius determines how fast the influence of an input dies out in a reservoir with time, and how stable the reservoir activations are.

**Input Scaling \***

For uniformly distributed Win we usually define the input scaling a as the range of the interval [−a;a] from which values of Win are sampled; for normal distributed input weights one may take the standard deviation as a scaling measure.

Scale the whole Win uniformly to have few global parameters in ESN. However, to increase the performance:

– scale the first column of Win (i.e., the bias inputs) separately;  
– scale other columns of Win separately if channels of u(n) contribute differently to the task.

It is advisable to normalize the data and may help to keep the inputs u(n) bounded avoiding outliers (e.g., apply tanh(·) to u(n) if it is unbounded).

The input scaling regulates:

– the amount of nonlinearity of the reservoir representation x(n) (also increasing with ρ(W));

– the relative effect of the current input on x(n) as opposed to the history (in proportion to ρ(W)).

**Leaking Rate \***

The leaking rate α of the reservoir nodes in (3) can be regarded as the speed of the reservoir update dynamics discretized in time.

α can be regarded as the time interval in the continuous world between two consecutive time steps in the discrete realization.

Set the leaking rate α in (3) to match the speed of the dynamics of u(n) and/or ytarget(n).

This is one more of the global parameters to be tuned by trial and error.

To eliminate the random fluctuation of performance, keep the random seed fixed and/or average over several reservoir samples.

When manually tuning the reservoir parameters, change one parameter at a time.

Dimensions:

n : discret time [1, T] with T the number of data points in the training set

u(n) 🡪 (Nu, 1)

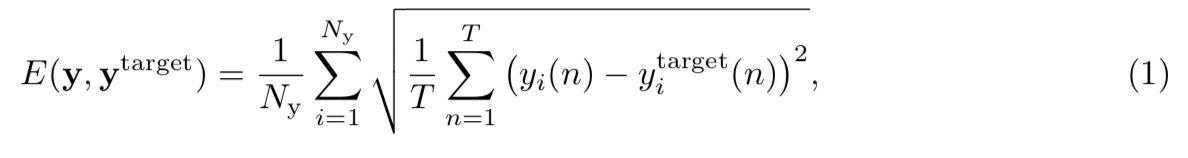
y(n) 🡪 (Ny, 1)

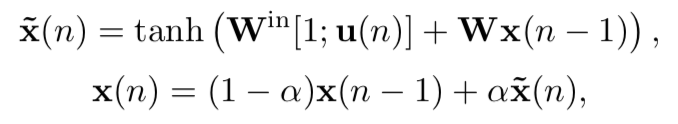
x(n) 🡪 (Nx, 1)

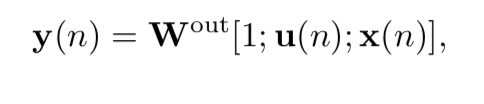
Win 🡪 (Nx, 1 + Nu)

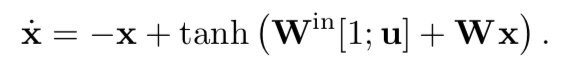
W 🡪 (Nx, Nx)

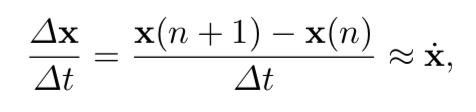
Wout 🡪 (Ny, 1 + Nu + Nx)

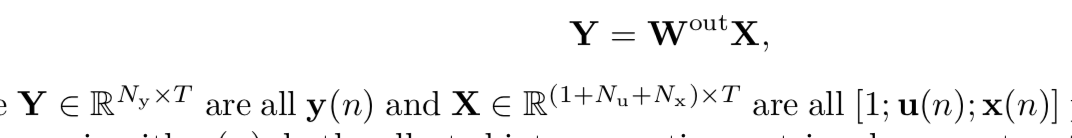


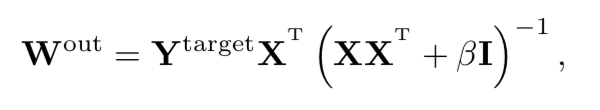


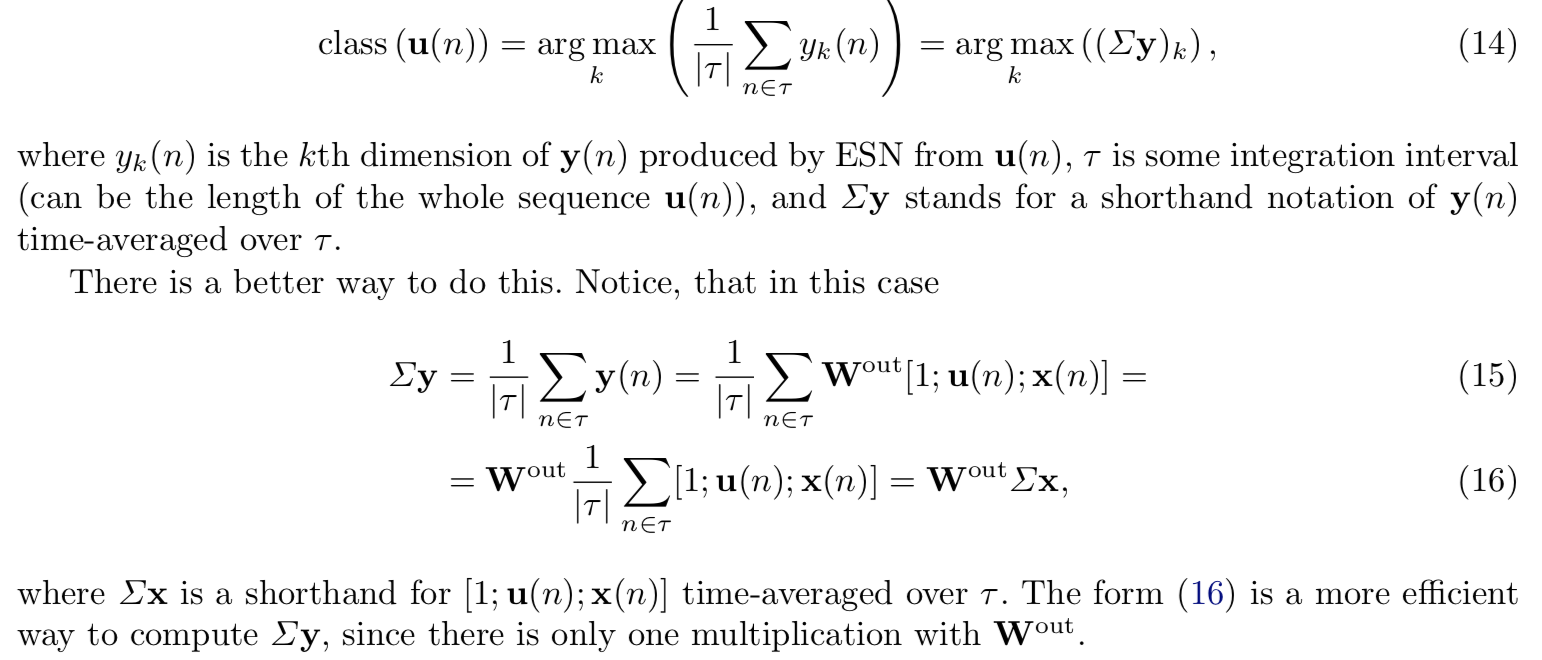












**Spoken digit recognition**. The spoken digit data set consists of five female speakers, uttering the numbers zero to nine with a tenfold repetition for statistics24. Before injecting the information into the laser, we performed standard 25 preprocessing creating a cochleagram of each digit via the Lyon ear model . For simplicity, mask M and connectivity matrix oi were merged into a single matrix (o). In this matrix, 98% of the elements were set to zero in order to realize a sparse connectivity between cochleagram channels and transients. The remaining values were randomly selected among (0.41,0.59). The preprocessing following the digit’s chocleagram is illustrated in Fig. 5. After preprocessing, each cycle of length T2 data of the cochleagram’s 86 frequency channels were injected into the system with 8 bit resolution, corresponding to a rate of 1.1 Gbyte/s. One Byte here refers a unit of injected information, which can comprise different numbers of bits. Out of the entire body of spoken digits, we chose 20 random partitions of 25 samples each, using 475 samples for training the readout weights, keeping the remaining 25 for testing. Each random partition and each sample are used exactly once for testing (20-fold cross-validation). Statistical information is provided by repeating this procedure five times, creating different random combinations of testing and training. At the two data points, where the system approaches 0% error, the cross- validation procedure was repeated 100 times.

The input data matrix Ml (dimension Nf xNs) constructed with

the Lyonâ€™s Cochlear ear model consists of the corresponding Nf =86 frequency channels and a maximum of Ns=130 samples in time. Ml is multiplied with the input connectivity matrix Wi (dimension NNxNf , NN=400 being the number of virtual nodes in the delay line), creating the

data input Mi for the reservoir. Most of the elements of the connectivity matrix Wi are set to

zero, realizing a sparse and random connectivity between the input layer and the reservoir. The

remaining elements are chosen randomly from two discrete mask values, keeping the system in

a transient state for the duration of the spoken digit, while also breaking the symmetry between

the NN nodes. The elements of the connectivity matrix remain constant for the duration of the

node separation Î. For training the output weights we have randomly chosen 475 spoken digits

among a data set of 500, leaving 25 for testing. The read-out weights Ï‰rjk are calculated from a

ridge regression [23] on the system response to the 475 test samples. These weights correspond

to the coefficients of a read-out matrix Wr, which is expected to provide the identification of the

spoken digit in the form of a so-called target function. The entire training and test procedure is

repeated 20 times with different, non-overlapping fragmentations of the 500 speech samples.

By following this approach, we minimize the influence of individual speakers and spoken digits

on our results, as well as providing statistical information.

