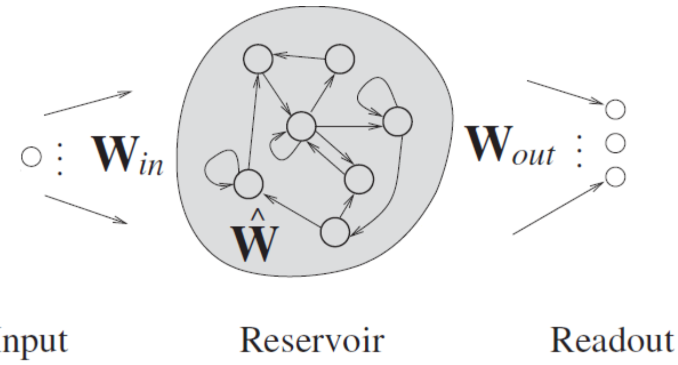
Additional Material (Lab3-2):

Echo State Networks



- Input $\mathbf{u}(t)$: column vector of size N_U
- Reservoir state $\mathbf{x}(t)$: column vector of size N_R
- Output $\mathbf{y}(t)$: column vector of size N_Y

In the case of autoregressive tasks on 1-dimensional time-series, such is the next-step prediction task for the Laser dataset, we use $N_U = N_Y = 1$.

Reservoir

- It is a recurrent, non-linear layer with (typically) sparse connectivity among units.
- It computes the state transition function: how the state at time step *t* is obtained from the input at time step *t* and from the state at time step *t-1*:

$$\mathbf{x}(t) = tanh(\mathbf{W}_{in}[\mathbf{u}(t);1] + \hat{\mathbf{W}}\mathbf{x}(t-1))$$

  - Input bias for the reservoir: concatenate the input at each time step with a constant input bias equal to 1;
  - Use a null state as initial state of the ESN: $\mathbf{x}(0) = \mathbf{0}$

Readout

- It is implemented as a feed-forward, linear layer
- It computes the output function: how the output of the network at time step *t* is obtained from the state at time step *t*

$$\mathbf{y}(t) = \mathbf{W}_{out}[\mathbf{x}(t);1]$$

  - Input bias for the readout: concatenate the state at each time step with a constant input bias equal to 1;

Reservoir Initialization

Initialize the reservoir according to the necessary condition for the Echo State Property:

$$\rho(\hat{\mathbf{W}}) < 1$$

Recall: the spectral radius is the maximum among the eigenvalues in modulus, i.e.
$$\rho(\hat{\mathbf{W}}) = max(|eig(\hat{\mathbf{W}})|)$$

In Matlab: `rho = max(abs(eig(Wr)));`

In the following, we refer to Matlab variables: Win for the input-to-reservoir weight matrix, Wr for the recurrent reservor weight matrix, Wout for the reservoir-to-readout weight matrix.

Initialization of the input-to-reservoir weight matrix Win

Values in matrix **W**in are chosen randomly from uniform distribution in the interval `[-inputScaling, inputScaling]`, e.g.

```
Win = inputScaling*(2*rand(Nr,Nu+1)-1);
```

Initialization of the recurrent reservoir weight matrix Wr

1) start with randomly generated matrix `Wrandom` (e.g. from uniform distribution in [-1,1])
   ```
   Wrandom = 2*rand(Nr,Nr)-1;%in the case of full connectivity
   ```

2) scale the random matrix to the desired spectral radius
   ```
   Wr = Wrandom * (rho_desired/max(abs(eig(Wrandom))));
   ```

Readout Training

Only the readout needs to be trained.

1. Discard an initial transient (run the network for some steps before starting collecting the states)
2. Collect all reservoir states and target values for each time step into matrices (after the initial transient)

$$\mathbf{X} = [[\mathbf{x}(1);1]\dots[\mathbf{x}(N);1]] \quad \mathbf{Y}_{target} = [\mathbf{y}_{target}(1)\dots\mathbf{y}_{target}(N)]$$

3. After having collected all the states, train the linear readout
    - Pseudo-inverse

      $$\mathbf{W}_{out} = \mathbf{Y}_{target}\mathbf{X}^{+}$$

      In Matlab:

      ```
      Wout = Ytarget * pinv(X);
      ```
    - Ridge regression (λ_r is a regularization coefficient)

      $$\mathbf{W}_{out} = \mathbf{Y}_{target}\mathbf{X}^{T}(\mathbf{X}\mathbf{X}^{T} + \lambda_r \mathbf{I})^{-1}$$

      In Matlab:

      ```
      Wout = Ytarget * X’*inv(X*X’+lambda_r*eye(Nr+1));
      ```

<u>note: the readout should not be trained after each time step, but only once after the collection of all the states</u>

Reservoir Guesses

For every reservoir hyper-parametrization the performance (e.g. accuracy for classification task, MSE for regression task) should be averaged over a number of reservoir guesses (different random instantiations of networks with the same values of the hyper-parameters).

Model Selection!

Hyper-parameters to take into account (at least): number of reservoir units (`Nr`), spectral radius (`rho`), input scaling parameter (`inputScaling`), readout regularization (`lambda_r`), …

Leaky Integrator Echo State Network (LI-ESN)

$$\mathbf{x}(t) = (1-a)\mathbf{x}(t-1) + a\tanh(\mathbf{W}_{in}\mathbf{u}(t) + \hat{\mathbf{W}}\mathbf{x}(t-1))$$

Leaking rate parameter $a \in [0,1]$

In this case the condition for the Echo State Property must be imposed to

$$\tilde{\mathbf{W}} = (1-a)\mathbf{I} + a\hat{\mathbf{W}}$$

(after $\widetilde{W}$ has been obtained with the desired spectral radius, you have to properly set the weight matrix $\widehat{W} = \left(\frac{1}{a}\right)(\widetilde{W} - (1-a)I)$)