Echo State Networks for time series analysis

Ponnuthurai Nagaratnam Suganthan [1], Filippo Maria Bianchi [2,3]

[1]Nanyang Technological University, Singapore
[2]Department of Physics and Technology, UiT the Arctic University of Norway,
[3]Università della Svizzera Italiana

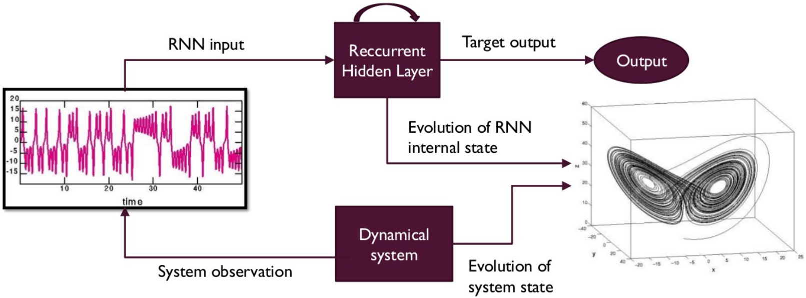
IEEE World Congress on Computational Intelligence 2018

8-13 July 2018, Rio de Janeiro, Brazil

Table of Contents

Relation to dynamical systems

- A recurrent neural network (RNN) is a universal approximator of dynamical systems.
- It can be trained to reproduce any target dynamics, up to a given degree of precision.



Relation to dynamical systems

- An RNN naturally processes new inputs of any lengths.
- RNNs capture sequential information, by modeling temporal dependencies in the inputs.

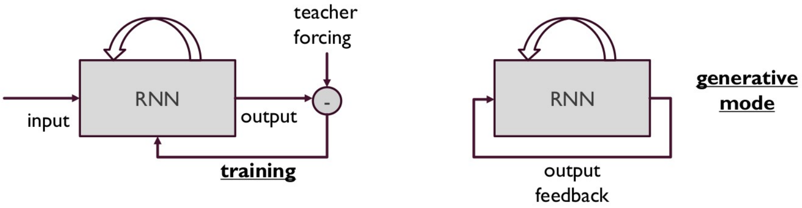
Example

prediction of next item in a sequence depends on the previously seen items

- Network output depends on current input and internal state.
- The internal state maintains a memory about history of past inputs.
- RNNs can capture arbitrarily long sequences, but in practice their memory is limited.

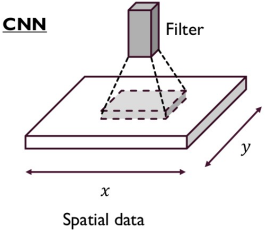
Teacher forcing and generative setting

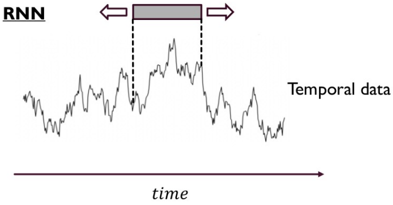
- RNN can be trained to predict future values of the driving input.
- Model parameters can be adapted to obtain a desired output (*teacher forcing*)
- A side-effect we get a *generative model*, which produces new samples by feeding back generated outputs.



Analogies with CNN

- Convolution in space (CNN) VS convolution in time (RNN) .
- **CNN** models relationships in space. Filter slides along x and y dimensions.
- **RNN** models relationships in time. A network with same weights is slided along *time* dimension.





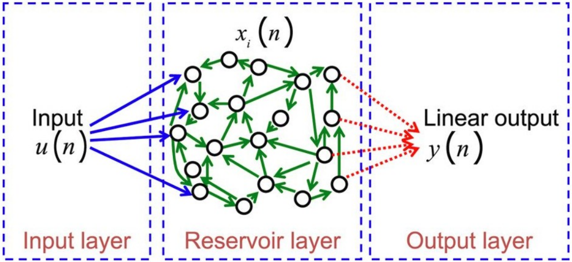
Training issues in traditional RNNs

- High computational training costs and potentially slow convergence.
- Back propagation through time is hard to parallelize on GPU.
- Big dataset are required to train large models.
- Local minima of the error function (which is generally non-convex).
- Vanishing of the gradients and problem of learning long-term dependencies.

Table of Contents

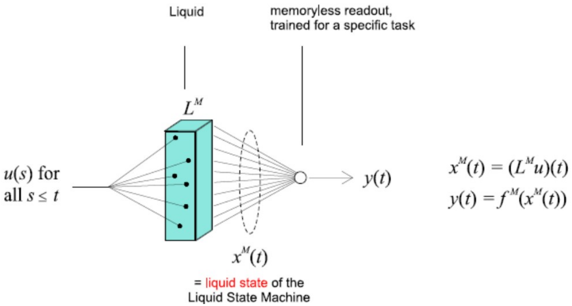
Reservoir computing

- Large, randomly generated and untrained recurrent reservoir layer.
- Tasks are solved by training only a memory-less readout (output weights).
- Fast and efficient (implemented on embedded devices).



Duport et al. S.R. 22381 (2016)

RC architectures (1/3) – Liquid State Machine[1]

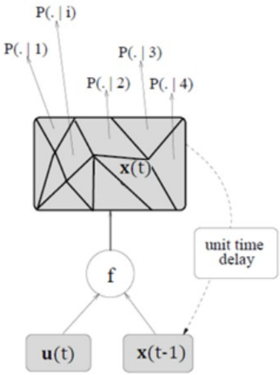


- Dynamics provided by a pool of spiking neurons with bio-inspired architecture.
- The liquid state should satisfy a point-wise separation property.

[1] Wolfgang Maass, Thomas Natschläger, and Henry Markram. "Real-time computing without stable states: A new framework for neural computation based on perturbations". In: *Neural computation* 14.11 (2002), pp. 2531–2560.

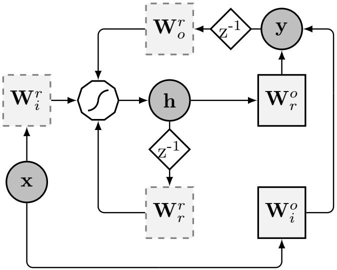
RC architectures (2/3) – Fractal Prediction Machines[2]

- Transform the training sequence into a unit hypercube with *n*-blocks.
- The longer is the common part in a sequence, the closer their representations.
- Prediction is done by detecting clusters in the training *n*-block representation via vector quantization.



[2] Peter Tino and Georg Dorffner. "Predicting the future of discrete sequences from fractal representations of the past". In: *Machine Learning* 45.2 (2001), pp. 187–217.

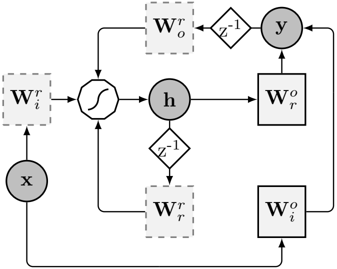
RC architectures (3/3) – Echo State Networks[3]



- Large recurrent reservoir is general as it produces many dynamical features of the input.
- Echo state property guarantees a vanishing effect of the inputs (stability).
- Linear readout trained with ridge regression.

[3] Herbert Jaeger. "The echo state approach to analysing and training recurrent neural networks-with an erratum note". In: *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report* 148.34 (2001), p. 13.

RC architectures (3/3) – Echo State Networks



$$\text{(untrained)} \quad h(t) = f(\mathbf{W}_i^r x(t) + \mathbf{W}_r^r h(t-1) + \mathbf{W}_o^r y(t-1))$$

$$\text{(trained)} \quad y(t) = \mathbf{W}_i^o x(t) + \mathbf{W}_r^o h(t)$$

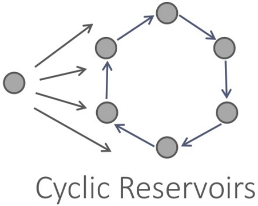
ESN variants (1/4): Leaky integrator units[4]

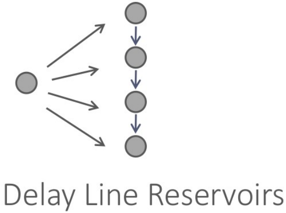
$$h(t) = \alpha h(t-1) + (1-\alpha) f(x(t), h(t-1), y(t-1))$$

- Apply an exponential moving average to reservoir states.
- Leaky integrator $\leftrightarrow$ low pass filter.
- Large α: fast dynamics are filtered.
- Small α: reservoir dynamics react faster to input changes.

[4] Herbert Jaeger et al. "Optimization and applications of echo state networks with leaky-integrator neurons". In: *Neural networks* 20.3 (2007), pp. 335–352.

ESN variants (2/4) – Deterministic reservoirs[5]



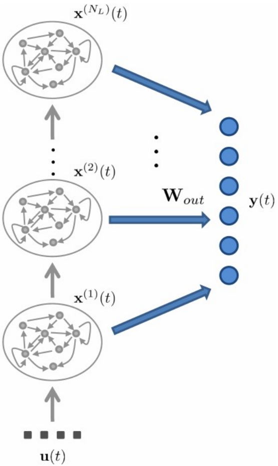


- Reservoir connections and weights are no longer randomly generated.
- Connections and weights have fixed, predefined structure.
- More control on the internal dynamics.
- Less computational units are necessary.

[5] Ali Rodan and Peter Tino. "Minimum complexity echo state network". In: *IEEE transactions on neural networks* 22.1 (2011), pp. 131–144.

ESN variants (3/4) – Deep reservoirs[6]

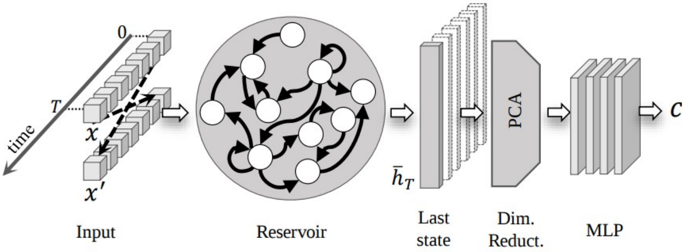
- Reservoir implemented as a stack of untrained recurrent layers.
- Explicit multiple time-scales representations.
- Each layer has different hyperparameters, yields different dynamics and models a specific time-scale.



[6] Claudio Gallicchio, Alessio Micheli, and Luca Pedrelli. "Deep reservoir computing: a critical experimental analysis". In: *Neurocomputing* 268 (2017), pp. 87–99.

ESN variants (4/4) – Deep readout, bidirectional reservoir[7]

- Input sequence are processed both in forward and backward direction.
- Earlier time dependencies are better captured by latter states.
- Dimensionality reduction (PCA) compress large reservoir output.
- Readout implemented by a deep neural network.



[7] FM Bianchi et al. "Bidirectional deep-readout echo state networks". In: *European Symposium on Artificial Neural Networks*. 2018.

Table of Contents

Hyperparameters in ESN (1/2)

- ESN trades precision of gradient descent with the “brute force” redundancy of random reservoirs.
- ESN models are more sensitive to hyperparameters selection.
- Main hyperparameters that control reservoir behavior:
  1. # processing units (reservoir size);
  2. spectral radius;
  3. input scaling.

Hyperparameters in ESN (2/2)

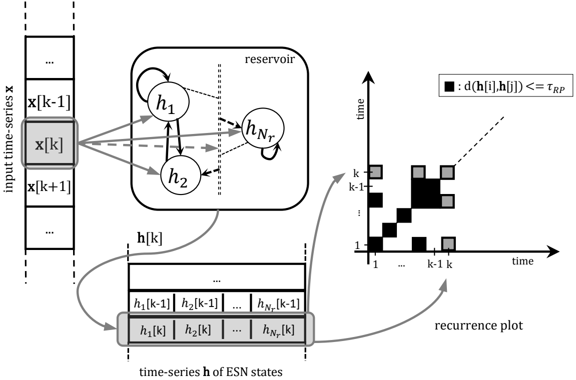
Two main approaches:

- Supervised model selection with cross-validation.
- Unsupervised model selection, based on reservoir dynamics.
  - Identification of the Edge of Chaos.

Edge of Chaos in ESN

- **Contractive dynamic:** stable system, short memory of past inputs, few dynamic features.
- **Chaotic dynamic:** unstable system, no Echo-State-Property, overfit.
- **Edge:** configuration between contractive and chaotic dynamics.
- ESN achieves best performance on the edge.
- Edge can be identified unsupervisedly (independent of the task).
- Proposed approaches: Recurrence Plots, Fisher information maximization, Visibility Graphs.

Example: Recurrence Plots (RP) analysis[8]

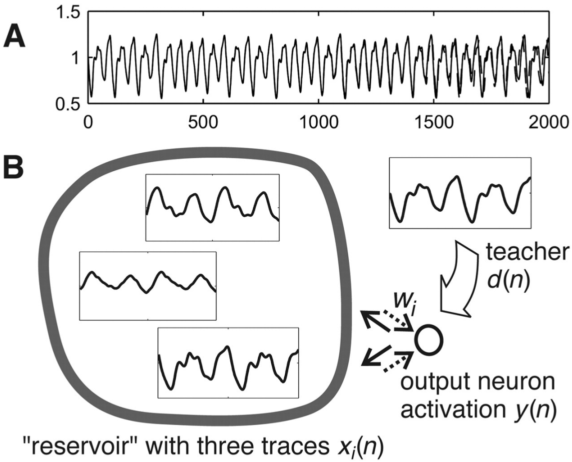


- RP measures similarities between ESN states in time and characterizes reservoir dynamics.
- ESN hyperparameter tuned by maximizing different statistics on the RP.

[8] Filippo Maria Bianchi, Lorenzo Livi, and Cesare Alippi. “Investigating echo-state networks dynamics by means of recurrence analysis”. In: *IEEE transactions on neural networks and learning systems* (2016).

Table of Contents

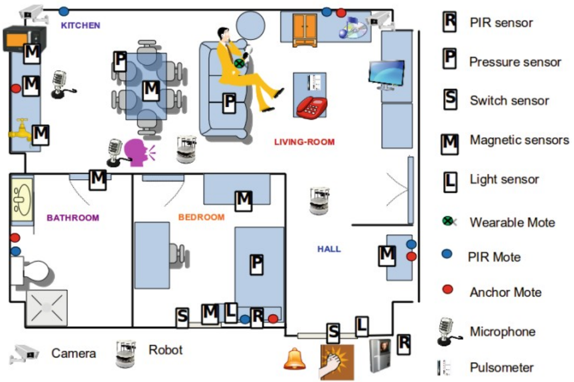
Prediction of chaotic time series[9]

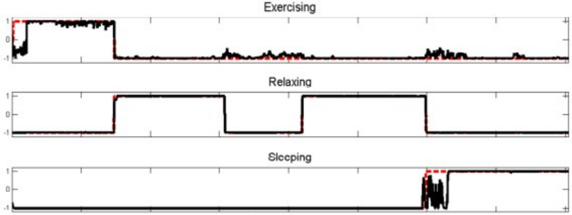


- ESN can emulate chaotic systems by learning their dynamic attractors.
- Difficult task: values in chaotic systems never repeats.
- Mackey-glass time series can be generated by an autonomous ESN.

[9] Herbert Jaeger and Harald Haas. "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication". In: *science* 304.5667 (2004), pp. 78–80.

Human Activity Recognition (HAR) and Localization[10]



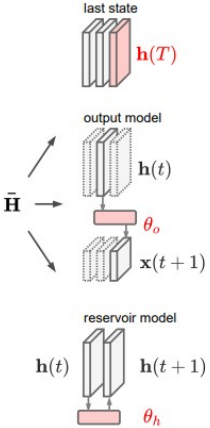


- Input from heterogeneous sensor sources (data fusion).
- Predicting events occurrence and confidence.
- Effectiveness in learning a variety of HAR tasks.

[10] Giuseppe Amato et al. "A benchmark dataset for human activity recognition and ambient assisted living". In: *Ambient Intelligence-Software and Applications–7th International Symposium on Ambient Intelligence (ISAmI 2016)*. Springer. 2016, pp. 1–9.

Time series classification[11,12]

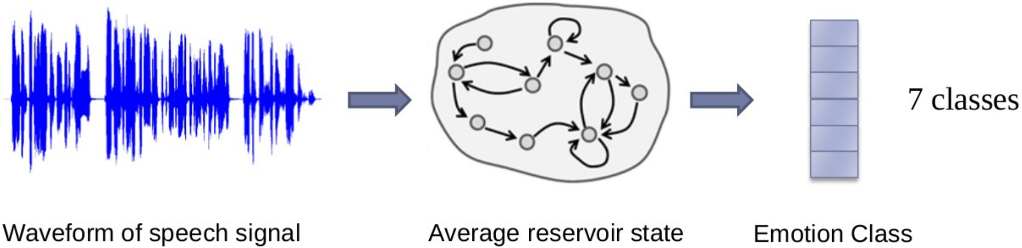
- Sequence of reservoir states $\bar{\mathbf{H}}$ is generated while input is processed.
- The input can be represented by
    1. last state $\mathbf{h}(T)$,
    2. average state $\frac{1}{T}\sum_{t=0}^{T}\mathbf{h}(t)$,
    3. parameters θ_o, s.t. $\mathbf{x}(t+1) = \theta_o \mathbf{h}(t)$,
    4. parameters θ_h, s.t. $\mathbf{h}(t+1) = \theta_h \mathbf{h}(t)$



[11] Huanhuan Chen et al. "Learning in the model space for cognitive fault diagnosis". In: *IEEE transactions on neural networks and learning systems* 25.1 (2014), pp. 124–136.

[12] Filippo Maria Bianchi et al. "Reservoir computing approaches for representation and classification of multivariate time series". In: *arXiv preprint arXiv:1803.07870* (2018).

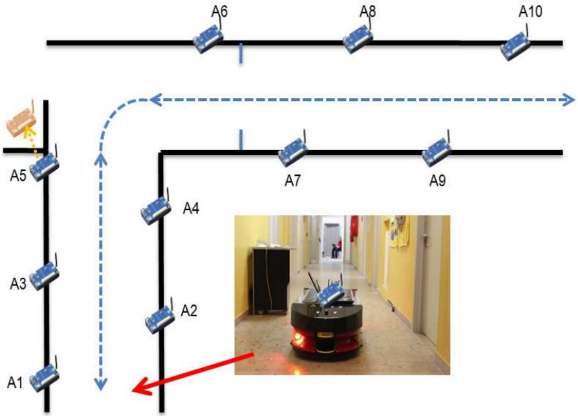
Example: Speech and Text Processing[13]

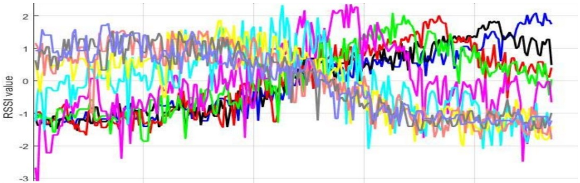


- The reservoir encodes temporal inputs in fixed-size vectors.
- Each vector is obtained as average of the reservoir states.
- Vectors are classified in 7 different sentiments.

[13] Claudio Gallicchio and Alessio Micheli. "A preliminary application of echo state networks to emotion recognition". In: *Fourth International Workshop EVALITA 2014*. Vol. 2. Pisa University Press. 2014, pp. 116–119.

Localization in Robotics[14]



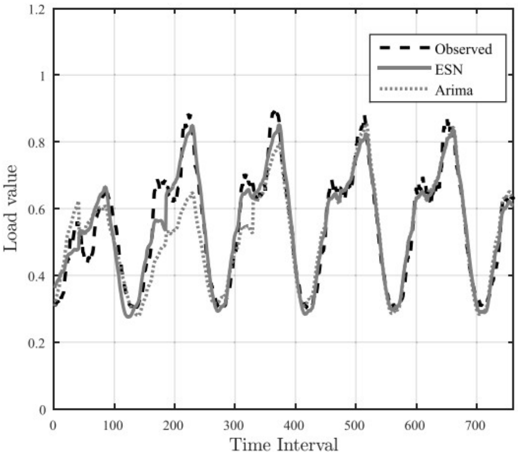


- ESN inputs: noisy radio signal strength index (RSSI) data generated by a network of sensors.
- ESN output: estimated coordinates (x, y) of the position of the robot.

[14] Mauro Dragone et al. "Rss-based robot localization in critical environments using reservoir computing". In: *Proceedings of the 24th European Symposium on Artificial Neural Networks (ESANN)*. 2016, pp. 71–76.

Short Term Load Forecasting[15],[16]

- Prediction of electricity, telephonic activity, load on a distribution network.
- Usually, 1h, 12h, 2h, or 48h ahead prediction.
- Outperform standard models (ARIMA), while using comparable computational resources.



[15] Filippo Maria Bianchi et al. "Prediction of telephone calls load using echo state network with exogenous variables". In: *Neural Networks* 71 (2015), pp. 204–213.

[16] Filippo Maria Bianchi et al. "Short-term electric load forecasting using echo state networks and PCA decomposition". In: *Ieee Access* 3 (2015), pp. 1931–1943.

Thanks for the attention

Contacts

EPNSugan@ntu.edu.sg
filippo.m.bianchi@uit.no

Sources

"*An Introduction to Echo State Networks*"
by Claudio Gallicchio,
https://sites.google.com/view/filippombianchi