RESERVOIR COMPUTING

학부생인턴 박근아

CONTENTS

RESERVOIR COMPUTING 개요

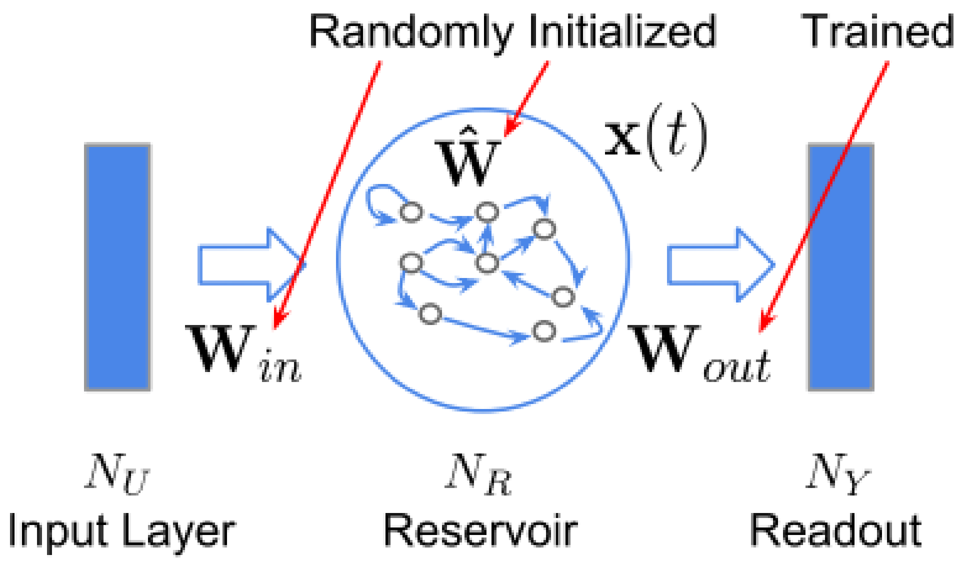

Fig. 1 : The ESN architecture.

Reservoir computing(RC) is an efficient paradigm for the design and implementation of RNNs. The model is composed by two components, a dynamic layer with recurrent connections called **reservoir** and a linear output layer called **readout**.

Usually, the **reservoir** contains a large number of recurrent units, connected to each other in a sparse manner. The recurrent weights of the reservoirs are **randomly initialized** according to specific criteria and then left untrained. In particular, small weights initializations characterize a contractive dynamics of the network state.

The **readout** is composed by linear units and it is trained by direct methods.

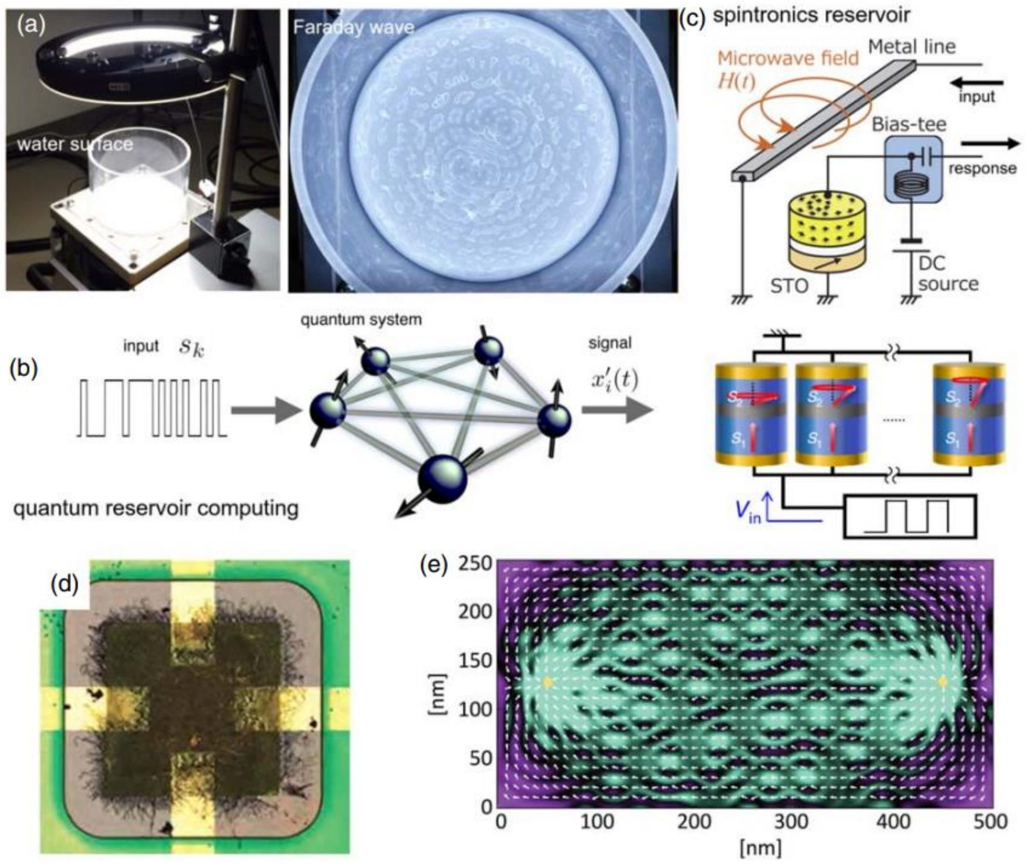

Fig. 2 : Variations of physical reservoirs.

In these models, the computation is characterized by causal, stationary and partially adaptive (not all weights are learned) transductions in which the encoding function is realized by a fixed recurrent layer (the reservoir) and the output function is realized by a linear output layer (the readout). Thereby, **only the non-recurrent part (the readout) is trained.**

Types : Echo State Network(ESN), Liquid State Machine(LSM)

Also **Physical reservoirs** are possible because of the inherent non-linearity of certain natural systems. The interaction between ripples on the surface of water contains the nonlinear dynamics required in reservoir creation, and a pattern recognition RC was developed by first inputting ripples with electric motors then recording and analyzing the ripples in the readout.

Processing scheme in reservoir computing

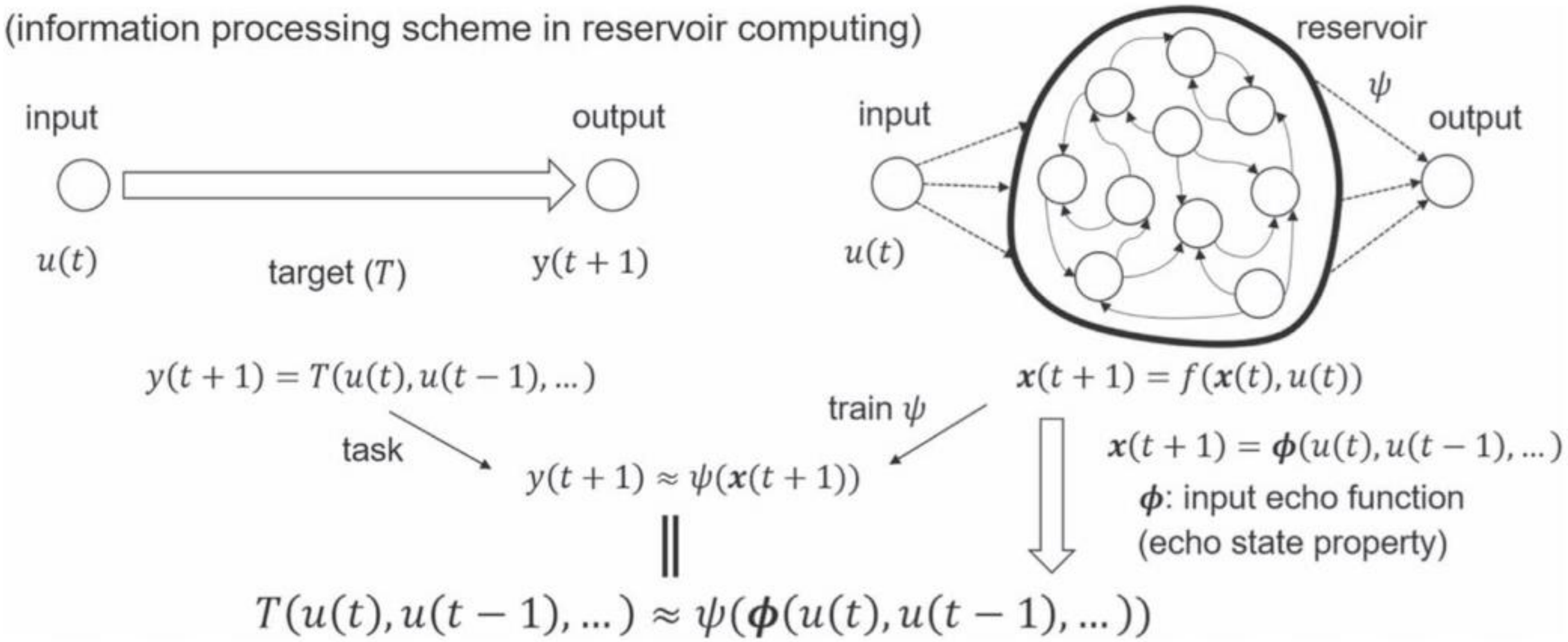

Fig. 3 : How the echo state property works in RC.

Schematics showing how the echo state property works in RC. As can be seen in the diagram, input echo function f is a part intrinsic to the reservoir, and the experimenters can adjust the output using readout function ψ.

Processing scheme in reservoir computing

Consider that we have input **u(t)** and the reservoir state **x(t)** at timestep t, and the reservoir dynamics is expressed as **$x(t+1) = f(x(t),u(t))$.** In general, a task T targeted by RNN is a function of the previous input sequence, which is sometimes called a temporal machine learning task; then, it is expressed **$y(t+1) = T(u(t),u(t-1),...)$.**

In the RC scheme, by tuning the readout ψ (note that this readout function does not have to be linear in general), we aim to approximate the target y(t), which is expressed as **$y = \psi\ (x(t))$.** Now, if the reservoir fulfills the ESP, then **$x(t) = \Phi(u(t-1),u(t-2),\ldots)$,** where Φ is called the input echo function in Ref. 1 and where it is a function intrinsic to the reservoir. This implies that the internal state of the reservoir is completely described by the driven input sequence and is related to the filter concept. In summary, the RC scheme can be expressed as exploiting the function intrinsic to the reservoir Φand adjusting the readout function ψ to approximate the target function T, which is expressed as **$T(u(t), u(t-1),\ldots) = \psi\ (\Phi(u(t),u(t-1),\ldots)\ )$.**

일반적인 RNN, LSTM 계열 신경망과의 차이

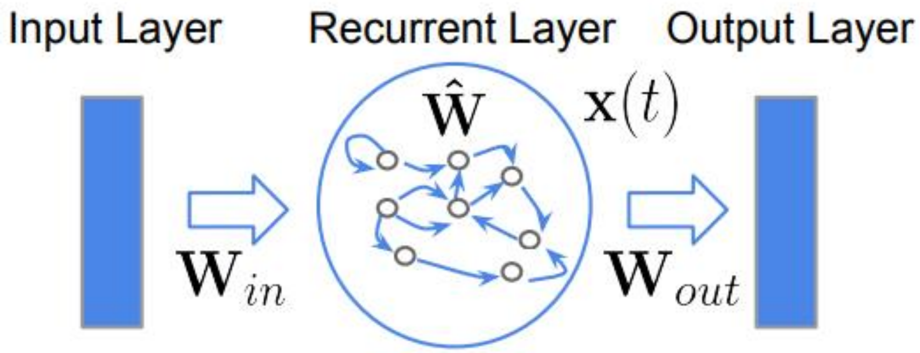

Fig. 4 : Architecture of simple RNN (SRN) model.

Conventionally, to train an RNN, a **backpropagation-through-time** (BPTT) method) is frequently used. In this method, all the weights of the network are basically tuned toward the target function.

In the RC framework, by preparing an RNN equipped with a massive amount of nonlinear elements coupled with one another, called a reservoir, only the readout part is usually trained toward the target function. In the simplest case, this readout part consists of linear and static weights that directly connect the reservoir nodes and output node.

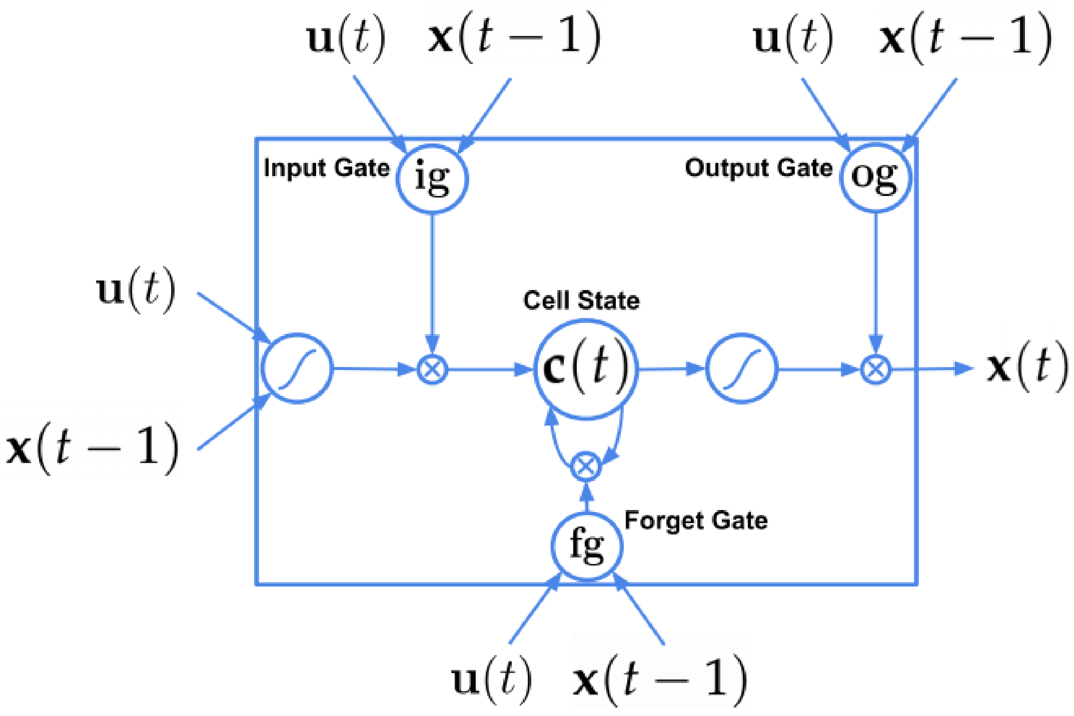

Fig. 5 : Architecture of LSTMmodel.

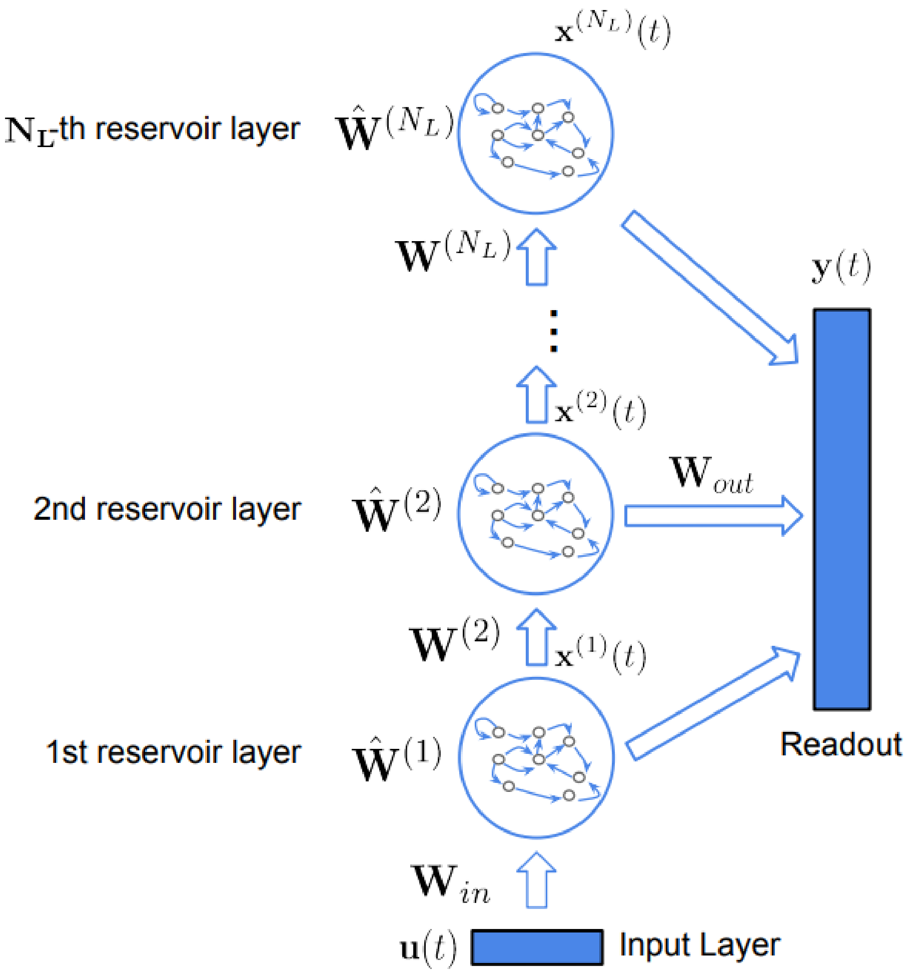

Fig. 6 : Architecture of a DeepESN.

Model	total recurrent units	free parameters	test ACC	computation time
Piano-midi.de				
DeepESN	6000	540088	**33.33 (0.11) %**	**386**
ESN	6000	540088	30.43 (0.06) %	748
SRN	652	540596	29.48 (0.35) %	3185
LSTM	316	539816	28.98 (2.93) %	2333
GRU	369	539566	31.38 (0.21) %	2821
MuseData				
DeepESN	6000	504082	**36.32 (0.06) %**	**789**
ESN	6000	504082	35.95 (0.04) %	997
SRN	632	503786	34.02 (0.28) %	8825
LSTM	307	504176	34.71 (1.17) %	18274
GRU	358	503072	35.89 (0.17) %	18104
JSBchorales				
DeepESN	6000	324052	**30.82 (0.12) %**	**83**
ESN	6000	324052	29.14 (0.09) %	140
SRN	519	323908	29.68 (0.17) %	341
LSTM	254	325172	29.80 (0.38) %	532
GRU	295	323372	29.63 (0.64) %	230
Nottingham				
DeepESN	6000	360058	69.43 (0.05) %	**677**
ESN	6000	360058	69.12 (0.08) %	1473
SRN	545	360848	65.89 (0.49) %	2252
LSTM	266	361286	70.00 (0.24) %	26175
GRU	309	359116	**71.50 (0.77) %**	11844

Table 1 : free parameters and test ACC achieved by DeepESN, SRN, LSTM and GRU. Computation time represents the seconds to complete training and test.

In this comparison we take into account all 4 polyphonic music tasks , namely Piano-midi.de, MuseData, JSBchorales8 and Nottingham9 . These applications concern next-step prediction tasks in which data is composed by 88-, 82-, 52- and 58-dimensional sequences for Piano-midi.de, MuseData, JSBchorales and Nottingham tasks, respectively. Since these datasets are characterized by sequences with high-dimensionality and complex temporal sequences, these challenging tasks are particularly suitable for RNNs evaluation

DeepESN required much less time in computation time with respect to the others models resulting in an extremely efficient model able to compete with the state of the art on challenging time-series tasks.

RESERVOIR COMPUTING의 장단점

Advantages

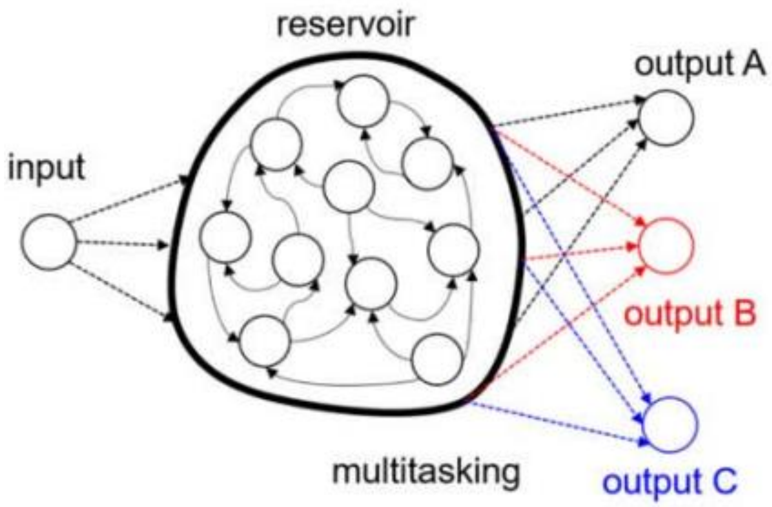

Fig. 7 : In the RC framework, multitasking can be safely implemented in principle, because no interference occurs among the tasks during the learning procedures.

The first advantage comes from the ease in the training procedure, which makes the learning quick and stable.

The second advantage is its ease in multitasking or in sequential learning. In the conventional approach of backpropagation, the entire network is optimized for the task TA first, and then the network is additionally trained for the task TB using the backpropagation method, so these two tasks interfere during the update of weights within the same network. In the RC framework, because the training is basically limited at the readout part, no interference occurs among the tasks, so multitasking can safely be implemented in principle.

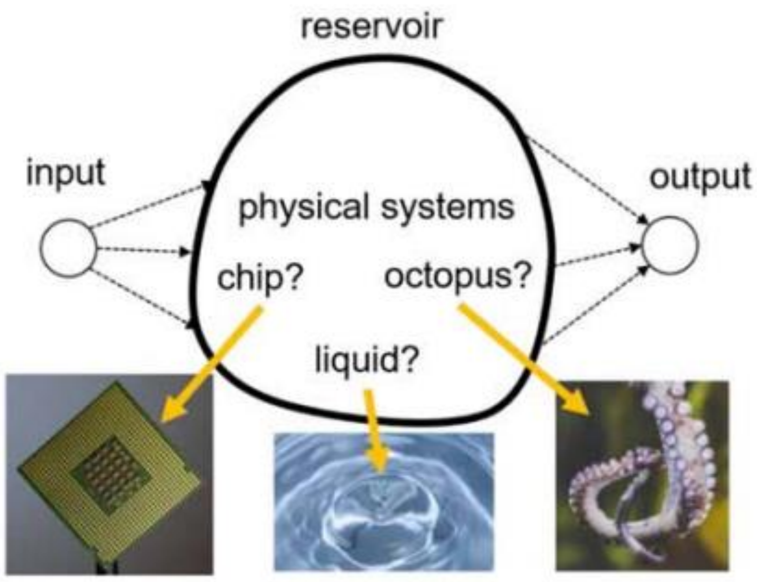

Fig. 8 : Physical reservoir computing, which exploits the physical dynamics as a reservoir.

The third advantage is the arbitrariness and diversity in the choice of a reservoir. That is, **PRC** provides a novel insight not only into the machine learning community, but also into the dynamical systems field, physics, materials science, and biological science.

Disadvantages

RNNs and RC both suffer from scalability problems(“scalable” means having a learning algorithm which can deal with any amount of data, without consuming ever growing amounts of resources like memory.) in high dimensional systems, as the required hidden state size dh to capture the high-dimensional dynamics can become prohibitively large especially with respect to the computational expense of training.

In order to scale the moto capture the high-dimensional dynamics can become prohibitively large especially with respect to the computational dels to high-dimensional systems we employ a parallelization scheme that exploits the local interactions in the state of a dynamical system.

RESERVOIR COMPUTING의 적용사례, 장점 발휘 분야

Automatic Berg Balance Scale Estimation

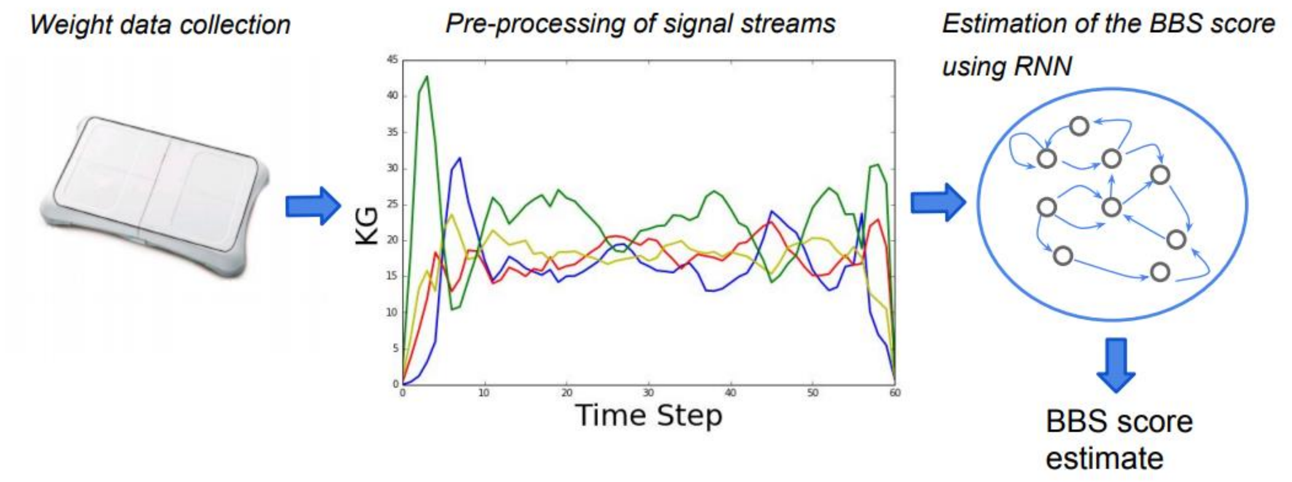

Fig. 9 : Graphical sketch of the overall operation of the proposed system for automatic BBS score estimation.

One of the common and easiest functional tests frequently used in medical practice is the Berg Balance Scale (BBS) test. Initially, this was proposed for balance assessment in elderly population but it has been frequently used in subjects with stroke, Parkinson's disease, brain injury, and multiple sclerosis.

The test is composed by 14 items, in the following also referred to as exercises, with a score ranging from 0 to 4 points. The maximum BBS score is 56. A score of 45 is indicated as a threshold for subjects at high risk of fall. each reduction of 1 point in BBS score is correlated to an increased risk of 6-8% to fall.

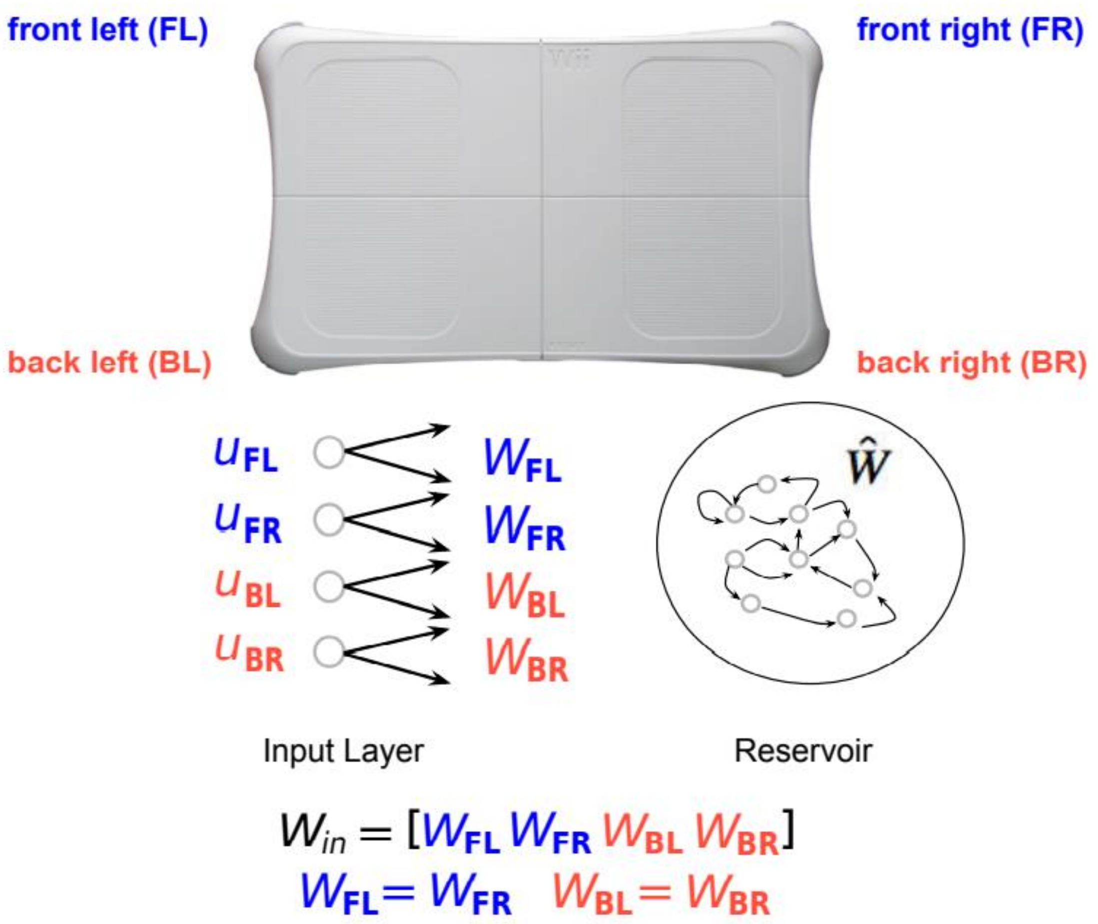

Fig. 10 : Graphical illustration of the adopted weight sharing approach. The input weights pertaining to signals coming from the left and the right side of the balance board are shared.

While a subject executes a BBS exercise on the Wii Balance Board, the sensor stream is gathered and collected into a database as above described. Then the data is used as input for the neural network model that computes the overall BBS score estimate.

RC System for BBS Score Estimation

In particular, we evaluate a weight sharing technique and the use of clinical data in addition to the time-series data gathered from the balance.

Physical reservoir computing using a soft robot arm

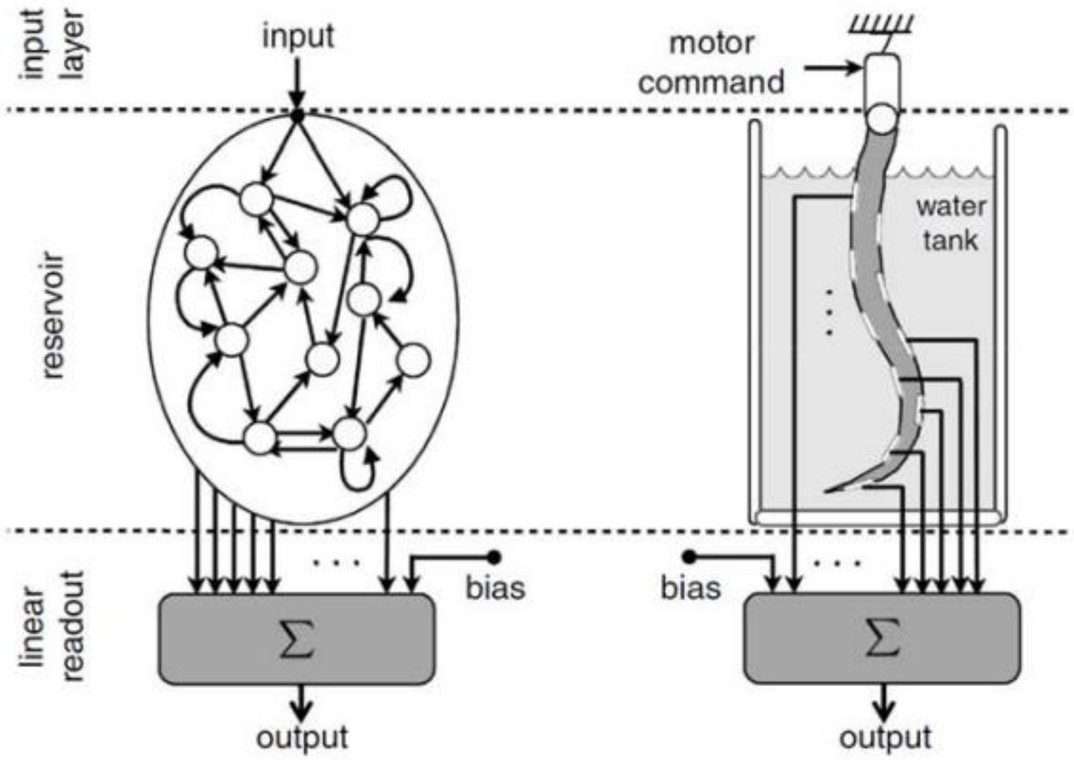

Fig. 11 : Schematics explaining how to exploit the soft robotic arm as a reservoir.

The dynamic model of the muscular-hydrostat system has the computational capacity to achieve a complex nonlinear computation. Furthermore, by incorporating the feedback-loop from the output to the next input (i.e. the next actuation pattern), we have demonstrated that the robot's behavioral control for the next time step can be implemented by using its current state of its body as a computational resource.

This concept has been also applied to the study of a quadruped robot, where the robot exploits its spine dynamics as a physical reservoir to control its actuation patterns and locomotions.

Applications

Especially, good at natural language processing, **continuous speech recognition**

For what regards real-world applications, ESNs obtained good results on tasks characterized by noisy, continuous and heterogeneous signals. Examples of these kind of tasks are represented by **Human Activity Recognition** and **Health Informatics**.

A relevant application field for ESN models concerns **time-series prediction**. Applications for the prediction of Market prices are presented in. Forecasting systems for the prediction of person movements are introduced in. Other works explore spatio-temporal forecasting for meteorological prediction.

Moreover, the ESN models are widely used to develop different **Control System** applications such as, robotic arm control, pneumatic muscle control, oil well control and motor control.

RC오픈 소스코드 확보 여부

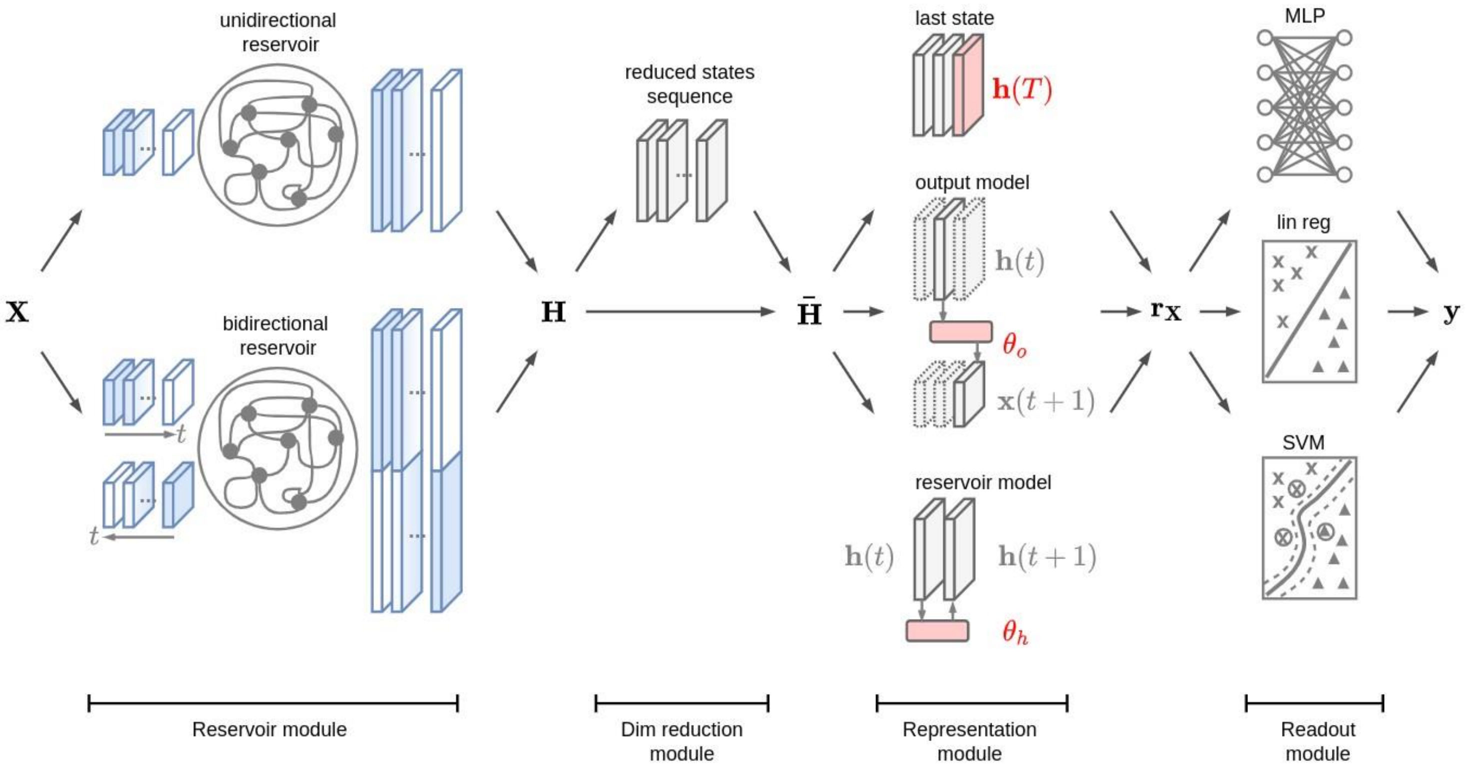

Fig. 12 : Framework overview.

https://github.com/FilippoMB/Time-series-classification-and-clustering-with-Reservoir-Computing

https://github.com/nschaetti/EchoTorch

https://github.com/lucapedrelli/DeepESN

https://github.com/cknd/pyESN

REFERENCE

[1] Kohei Nakajima “Physical reservoir computing—an introductory perspective”
Jpn. J. Appl. Phys. 59 060501(2020), pp.2-7.

[2] Pedrelli, Luca “ *Deep Reservoir Computing: A Novel Class of Deep Recurrent Neural Networks*” (PhD thesis). Università di Pisa. (2019), pp.31-26, ,120-127.

[3] P.R. Vlachas, J. Pathak, B.R. Hunt, T.P. Sapsis, M. Girvan, E. Ott, P. Koumoutsakos, “Backpropagation algorithms and Reservoir Computing in Recurrent Neural Networks for the forecasting of complex spatiotemporal dynamics”, Neural Networks, Volume 126, 2020, Pages 191-217, ISSN 0893-6080, https://doi.org/10.1016/j.neunet.2020.02.016.
(https://www.sciencedirect.com/science/article/pii/S0893608020300708), pp. 206.