Harnessing Nonlinearity: **Predicting Chaotic Systems and Saving Energy in Wireless Communication**

Publishde in Science Magazine, 2004

Siamak Saliminejad

Overview

- Eco State Networks
- How to build ESNs
- Chaotic Systems
- Equalization
- Conclusions

Authors

Department of *Electrical Engineering* and *Computer Science*
Jacobs University Bremen

Dr. Herbert Jaeger (*Computer Science*)

http://www.faculty.jacobs-university.de/hjaeger/

Dr. Harald Haas (*Electrical Engineering*)

https://www.jacobs-university.de/directory/hhaas

Echo state Networks (ESN)

ESN Is a Recurrent Neural Network (**RNN**s)

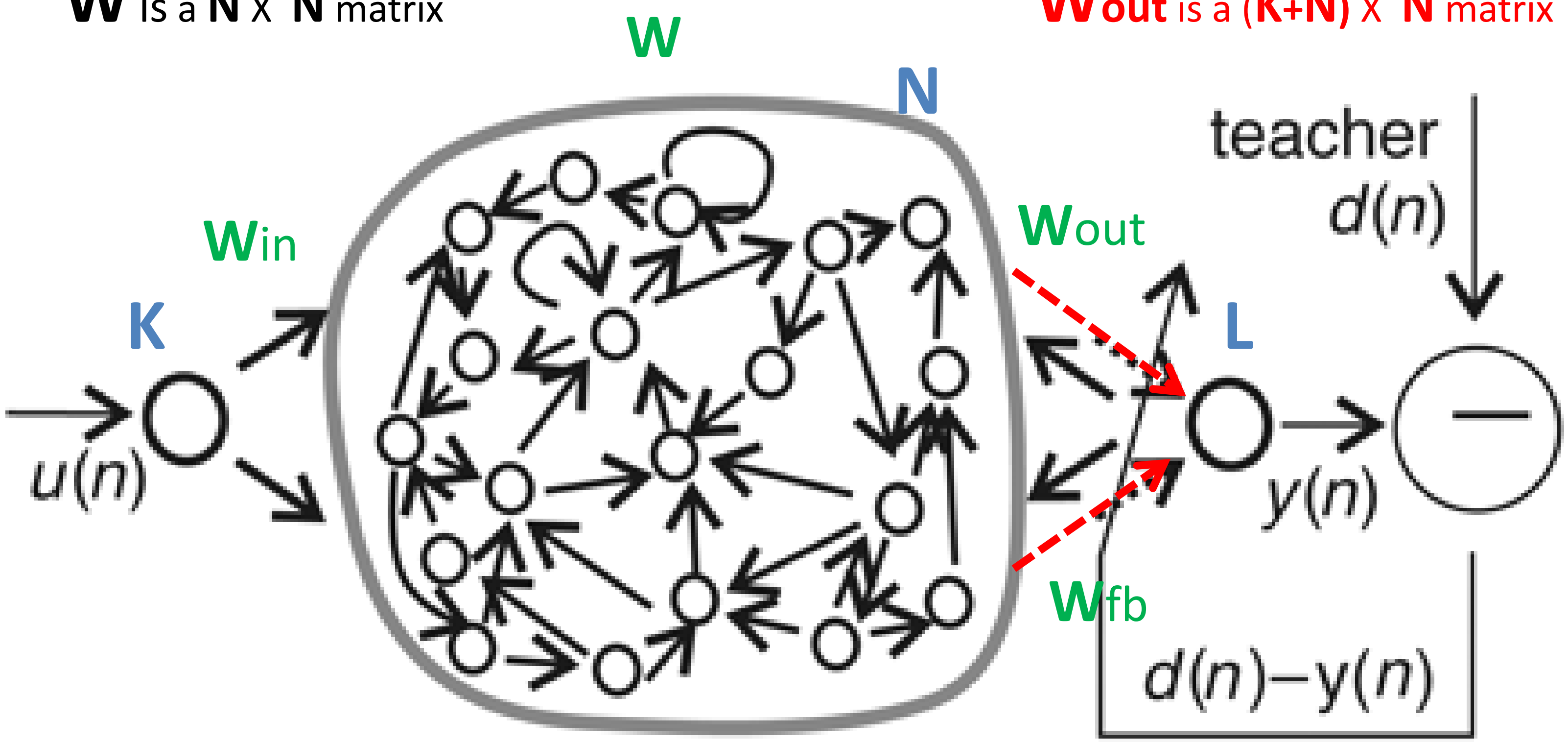
RNN: is a class of NN where connections between units form a directed cycle.

W$_{in}$ is a **K** X **N** matrix

W is a **N** X **N** matrix

W$_{fb}$ is a **L** X **N** matrix

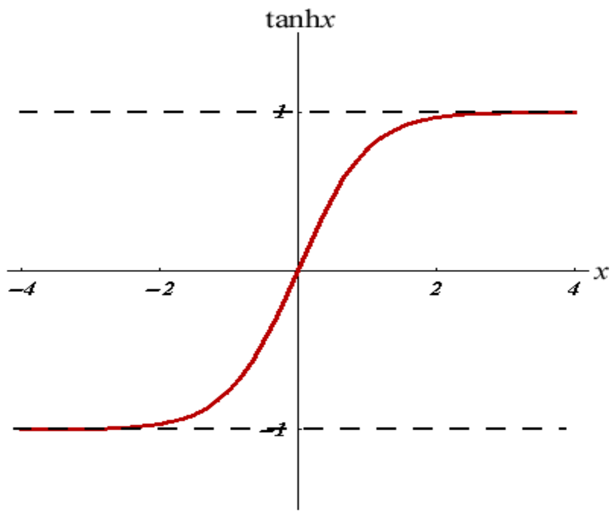
W$_{out}$ is a **(K+N)** X **N** matrix



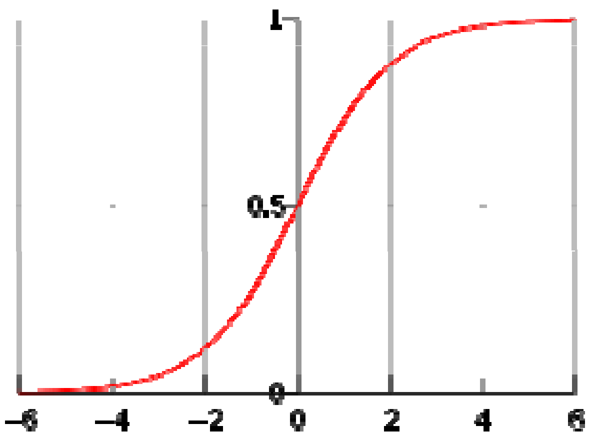
ESN-Units

Unit activation functions are typically **sigmoid**

$$y = \tanh(x)$$



$$y = \frac{1}{1 + e^{-x}}$$



ESN-System Equations

$$X(n+1) = f(W_{in}U(n+1)+WX(n)+W_{fb}y(n))$$

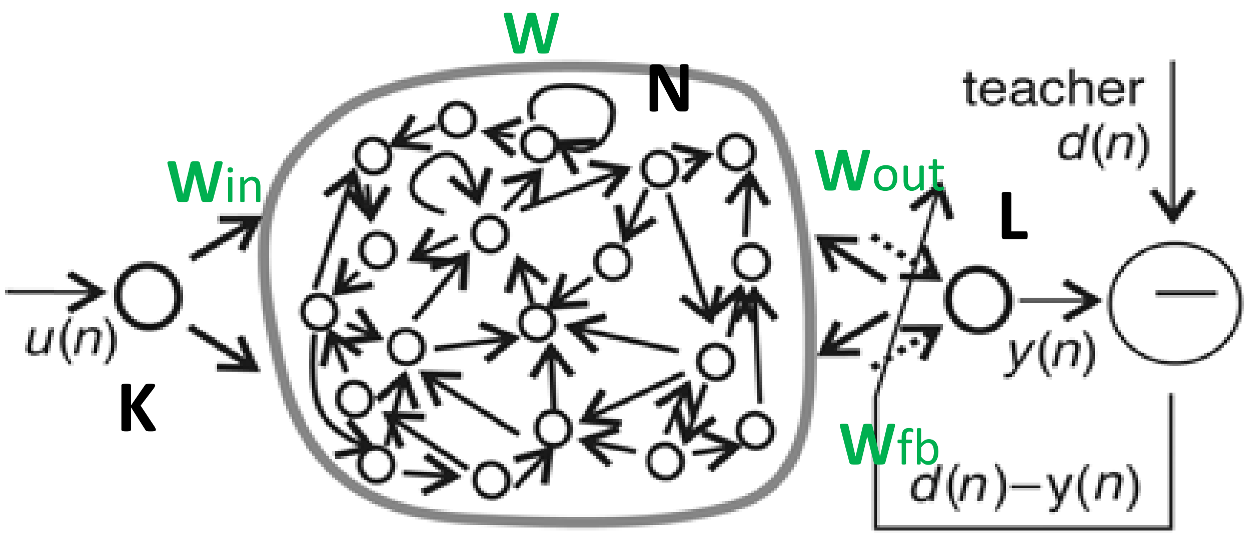
X: N dimensional Reservoir state

n: time

f :Sigmoid function

U: K-dimensional input signal

y: L-dimensional output signal



ESN-System Equations

$\mathbf{y}(n) = \mathbf{g}(W_{out}\, \mathbf{Z}(n))$ where $\mathbf{Z}(n)=[\mathbf{X}(n); \mathbf{u}(n)]$

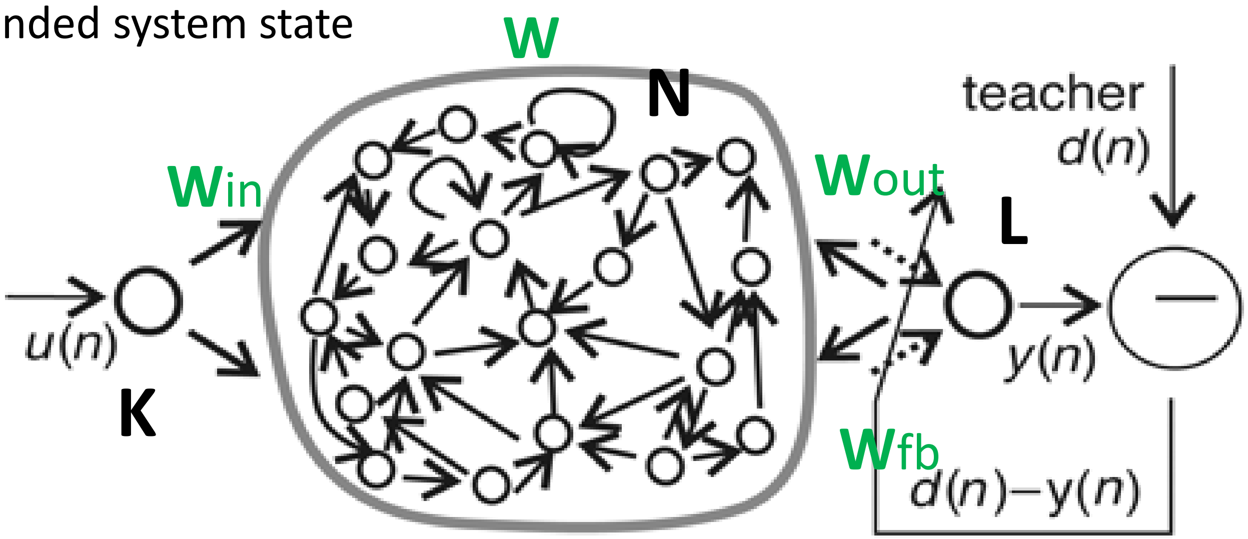
y: L-dimensional **output** signal

X: N dimensional **Reservoir** state

U: K-dimensional **input** signal

g: Output **activation** function (typically the identity or a sigmoid)

Z: extended system state



Echo State Property

This condition in essence states that the effect of a previous state x(*n) and a previous input u(n) on a* future state x(*n + k) should* ***vanish gradually as time passes*** *(i.e., k ( )), and not persist or even get* amplified.

Having echo states (or not having them) is a property of the network prior to training, that is, a property of the weight matrices **Win, W,** and *(optionally, if they exist)* $\mathbf{W_{fb}}$

The property is also relative to the **type of training data**: the same untrained network may have echo states for certain training data but not for others.

Unfortunately, there is **no known necessary and sufficient** algebraic condition which allows one to decide, given (W*in, W, Wfb), whether the network has the echo state* property.

Building ESNs

Step by step approach

Step1: Create Reservoir (W)

There are many Reservoir recepies. However, **weights** (W) and **topology** are selected **randomly**, in all of them:

I. Classic

1. Big: sufficiently large, with order ranging from tens to thousands.

2. Sparsely: the weight matrix W is sparse, with up to 20 % of possible connections.

3. Randomly: the weights of the connections are usually generated randomly from a uniform distribution symmetric around the zero value.

II. Different topologies *of the reservoir* from sparsely randomly connected ones.

III. Modular reservoirs*:* dividing the reservoir into sub-reservoirs

IV. Layered Reservoir *and …*

Optimizing reservoirs for a particular task or a class of tasks in an automated fashion is currently the most important field of ESN research.

Step 2: Attach input units to the reservoir (W_{in})

Absolute size of input weights

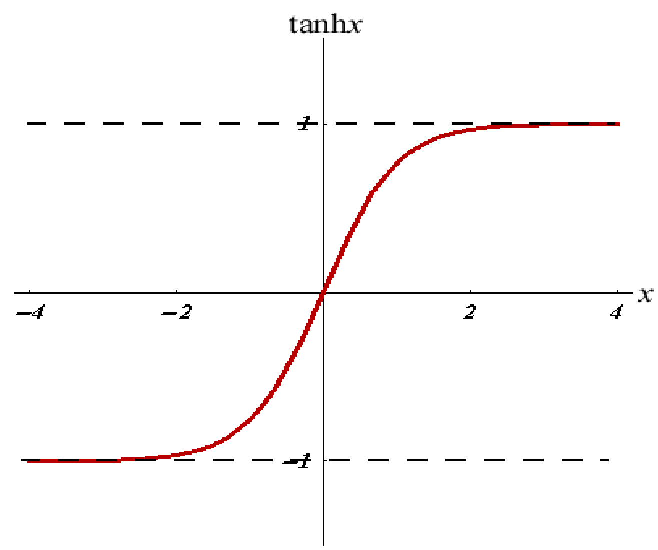
Random all-to-all connections

Small: the network state is only slightly excited around the DR's resting (zero) state. the network units operate around the linear central part of the sigmoid, i.e. one obtains a network with an almost **linear dynamics**.

Large: network is strongly driven by input and the internal units go closer to the saturation of the sigmoid, which results in a more nonlinear behavior of the resulting model.

Very large: the internal units will be driven into an almost pure – 1 / +1 valued, binary dynamics.

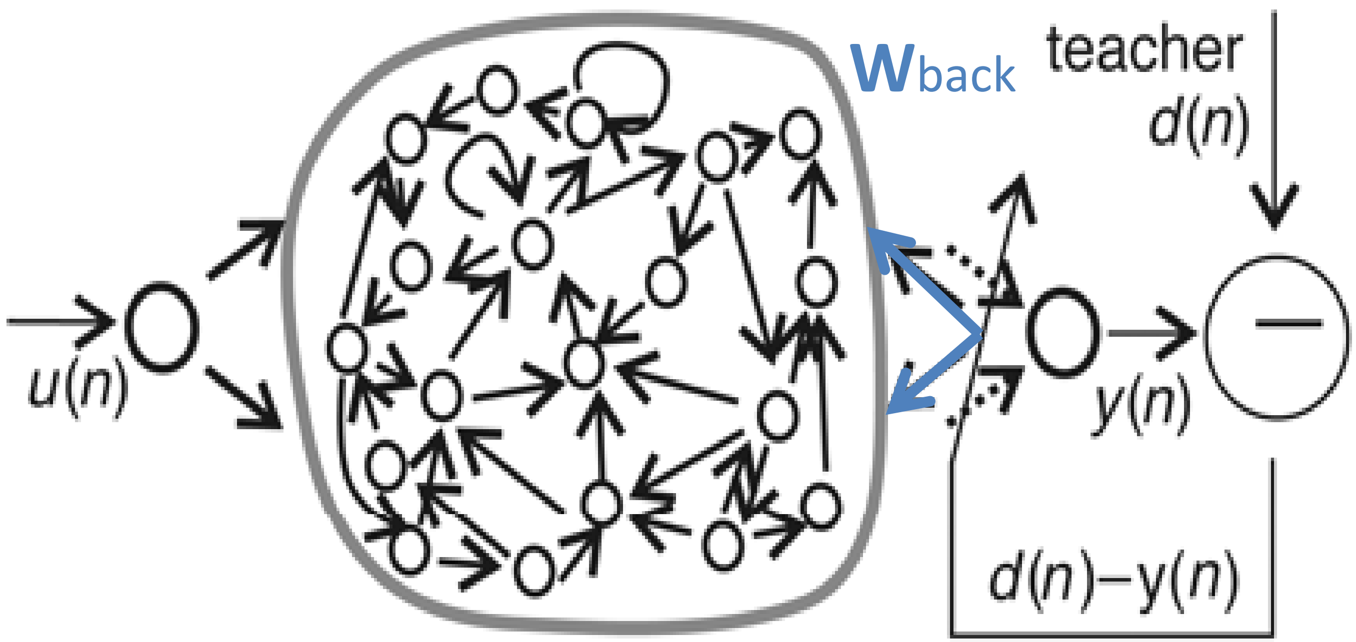
Manual adjustment and repeated learning trials will often be required to find the task appropriate scaling.



Step 3: W_{fb}

If the task requires **output feedback**

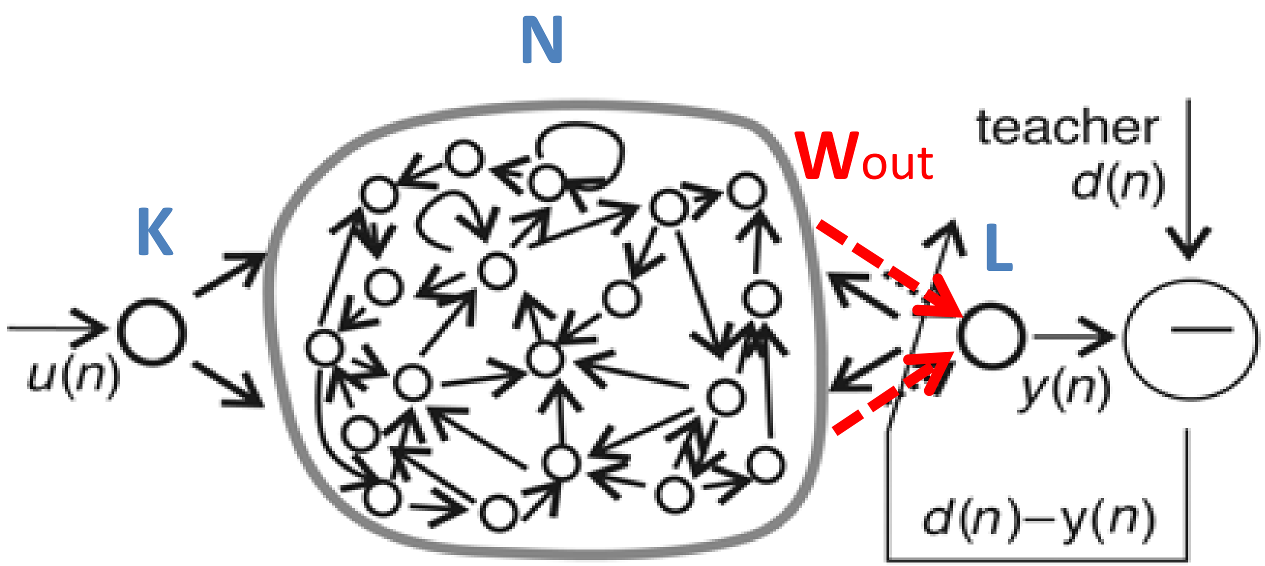
install randomly generated output-to-reservoir connections (all-to-all).



Step 4: Training

Given: A training input/output sequence **(u(1), d(1)), ..., (u(*T*), *d*(*T*)).**

Wanted: A trained ESN ($\mathbf{W}_{out}$) whose output y(n) approximates the teacher output d(n), when the ESN is driven by the training input u(n). In other words, Minimize error.



Step 4: Training

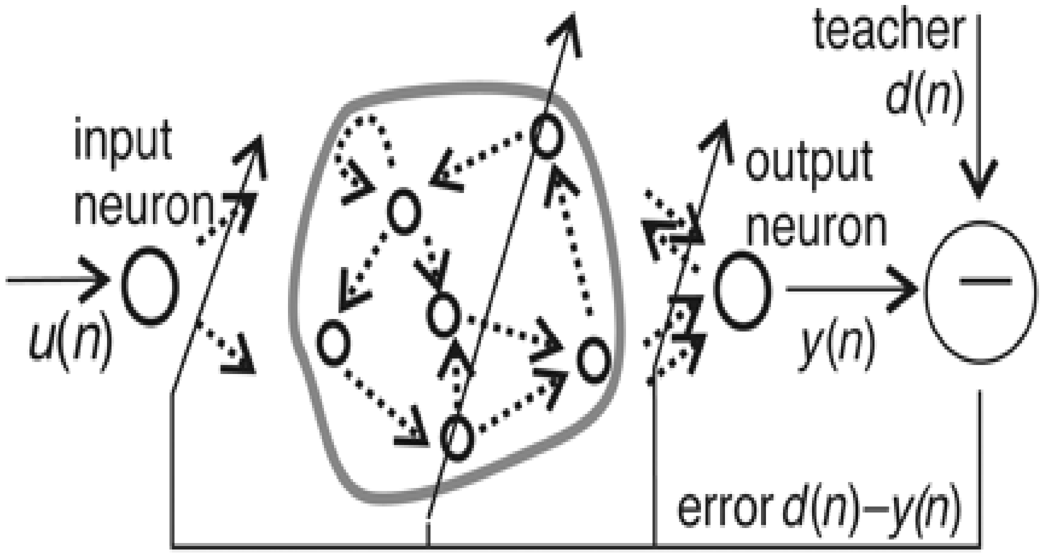
Minimize $Error = \sum_{n=1}^{T} (d(n) - y(n))^2$

While $y(n) = \sum_{i=1}^{N} w_i x_i(n)$

n:time
N: reservoir size
T: # of Training pairs
y: L-dimensional output signal
X: N dimensional Reservoir state

Any method for computing linear regressions can be used to obtain **W**

Previous Works



Minimize error (d(n)-y(n))

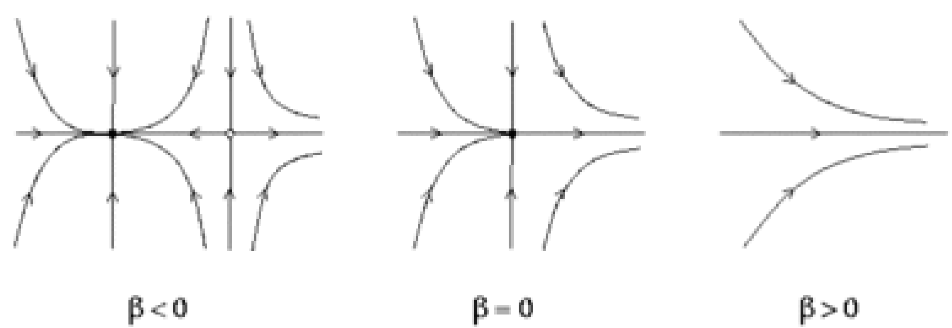
Recurrent

adapt all connections (input, recurrent, output) by some version of gradient descent.

Reservoirs have 5-10 neurons

The learning process is **slow**, may find **suboptimal** solutions, and is prone to become disrupted by **bifurcations**

A **bifurcation** of a dynamical system is a qualitative change in its dynamics produced by varying parameters.



Harnessing Nonlinearity:

Predicting Chaotic Systems and Saving Energy in Wireless Communication

Chaotic Systems

Chaotic systems: Time-dependent (dynamical) systems that are highly sensitive to initial conditions.

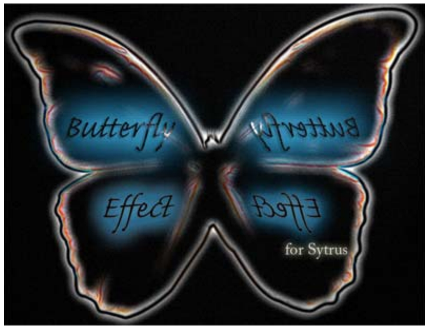
Arbitrarily small perturbation of the current trajectory may lead to significantly different future behavior (***Butterfly*** Effect)

*"Does the Flap of a **Butterfly**'s Wings in Brazil set off a Tornado in **Texas**?" By* Edward Lorenz (1970)

Chaotic systems are not periodic, not forever increasing and not approaching a fixed point. But, they are not random!

Long-term prediction of chaotic systems is impossible (weather)

Lasers, oscillating chemical reactions, and fluid dynamics are other examples of chaotic systems



Other Methods

wavelet networks (Liangyue Cao, et al., 1995)

Likelihood and Bayesian (Berliner, 1991)

Regression using Support Vector Machines
(Mukherjee , 1997) and RBF (Rosipal etal, 1998)

feedforward network trained with backpropagation
Chakroporty et al (1992), Elsner et al (1991), Andresia etal(2000)

Recurrent Neural Networks → Echo State Networks (ESN)

Previous Works

- *real-time recurrent learning* (Williams and Zipser 1989)
- *backpropagation through time* (Werbos 1990)
- *extended Kalman filtering* based methods (Puskorius etal 2004),
- Atiya-Parlos algorithm (Atiya and Parlos 2000)

Mackey-Glass System

It is a standard benchmark system

Reservoir size (**N**) = 1000 ⟶ A **1000 X 1000** matrix was constructed

The connectivity was **1**% — Random weights drawn from a **uniform** distribution over **(–1,1)**

Uses **feedback** connections

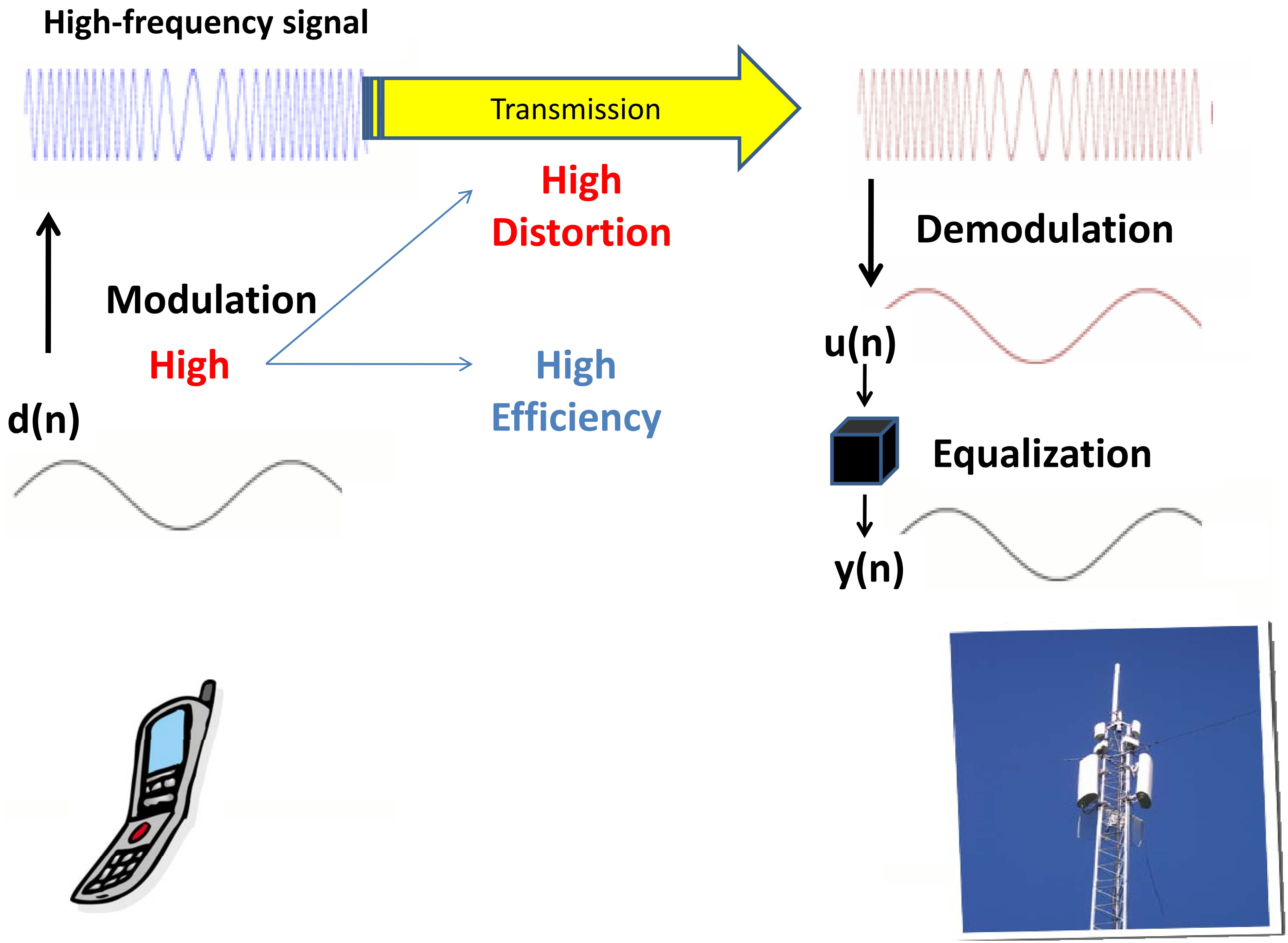
First **1000** steps were discarded to wash out initial transient.

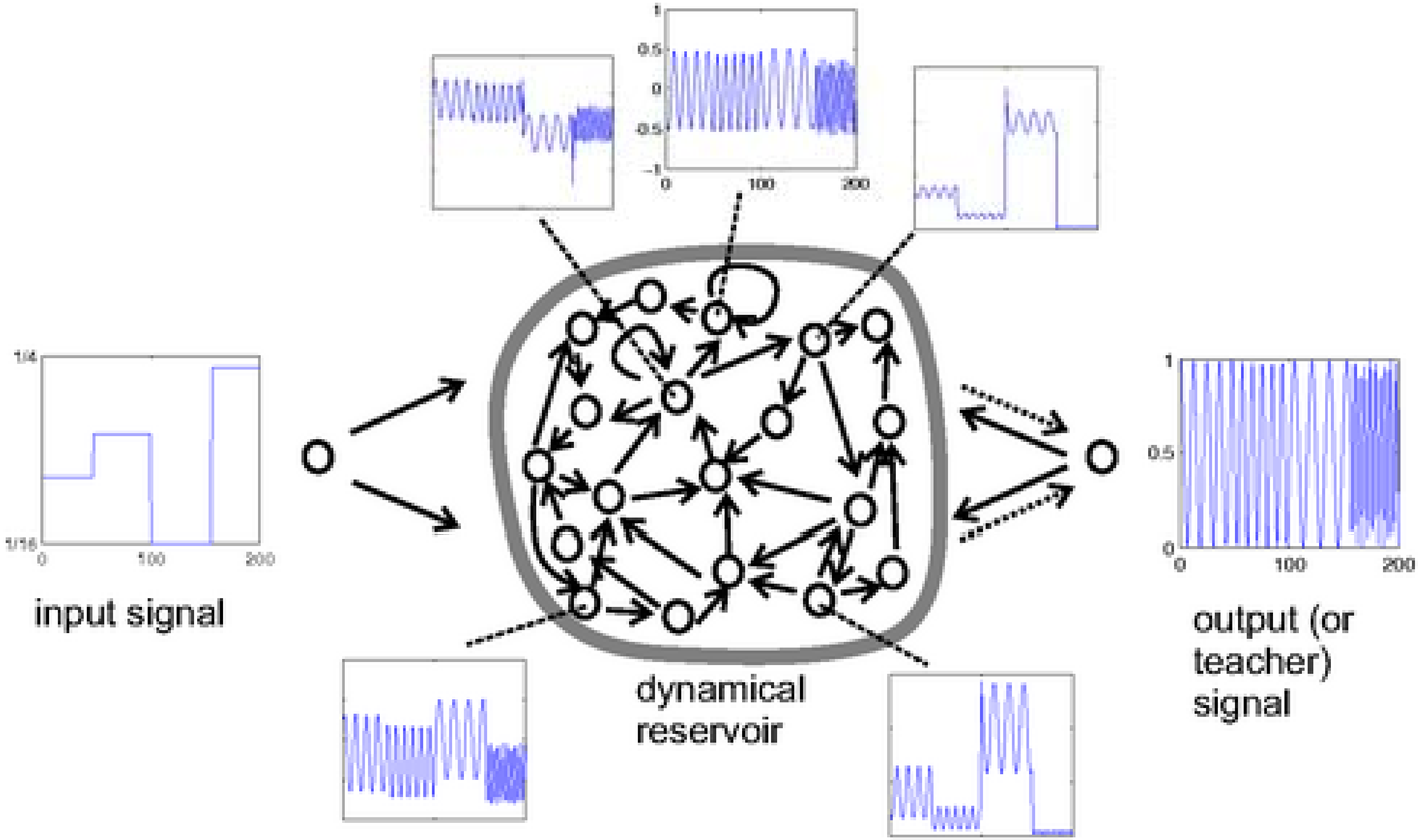
Echo signals x(n) were sampled from remaining **2000** steps and the network trained by them

The results show a **jump in modeling accuracy** with respect to previous models

Harnessing Nonlinearity:

Predicting Chaotic Systems and Saving Energy in Wireless Communication


High-frequency signal
Transmission
High
Distortion
Demodulation
Modulation
High
u(n)
High
Efficiency
d(n)
Equalization
y(n)



input signal
dynamical
reservoir
output (or
teacher)
signal

Equalization of wireless channels

A **46**-neuron reservoir and **W** were randomly generated with a **connectivity** of **20%**, and nonzero connection weights drawn from a **uniform** distribution over **(-1, 1).**

Output neuron was a **linear** neuron

Results Showed an improvement of two magnitudes for high signal-to-noise ratios.

Summary and conclusion

- The mathematical properties of large RNNs such that they can be used with a linear, trainable readout mechanism for general blackbox modeling are elucidated.

- ESN is faster and more applicable than previous methods that try to train all the connections.

- ESNs can be applied to all basic tasks of signal processing and control

- ESNs have been developed from a mathematical and engineering perspective, but exhibit typical features of biological RNNs