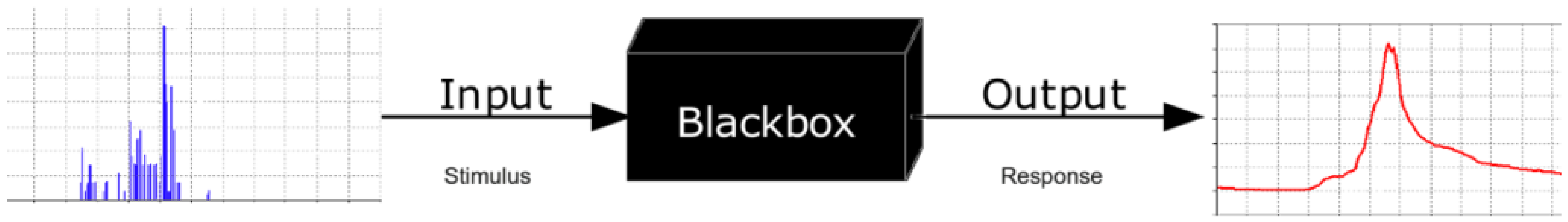
Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communications

Herbert Jaeger and Harald Haas

Presented by: Andrew Van

The Motivation - Modeling Nonlinear systems

- Linear systems follow 2 properties:
  - Superposition: $f(x+y) = f(x) + f(y)$
  - Homogeneity: $f(\alpha x) = \alpha f(x)$
- Non-linear systems are those that are missing one or all of these properties.
  - Because of this, it is generally difficult to study and obtain analytical models of non-linear systems
  - However, we can still study/utilize non-linear systems through black boxes (given some input to my mystery box, what is the output?)

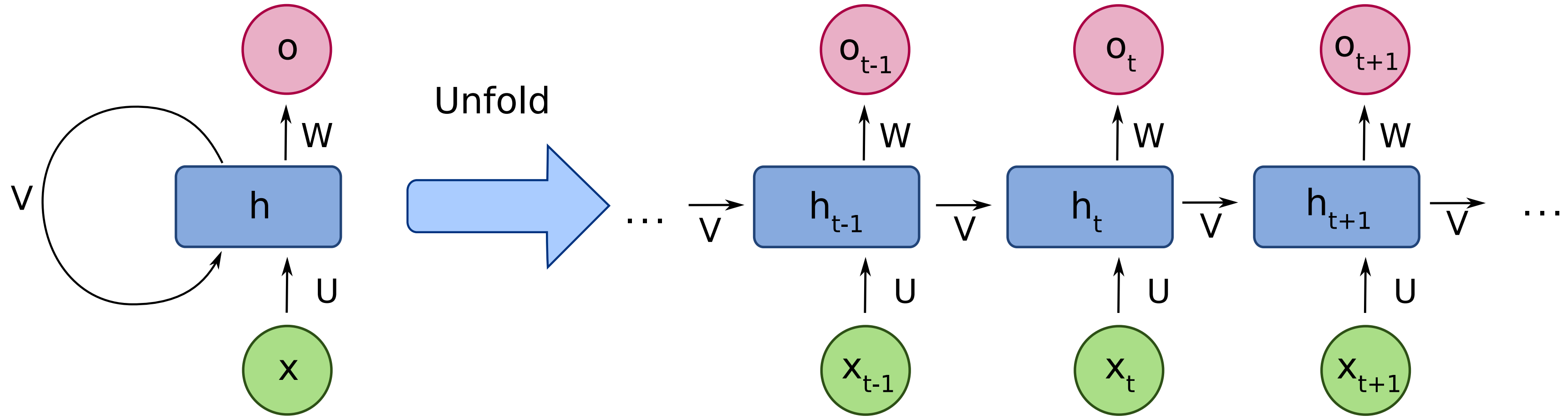


Nonlinear systems (continued)

- Most technical systems become non-linear at higher operational points (closer to saturation).
    - More energy efficient, but because of non-linearities they become unpredicatable
    - We use inefficient linear systems because of this
    - Biomechanical systems use their full dynamic range (up to saturation) and are very efficient, and also throughly nonlinear.
        - Inspirations from biology?

Echo State Networks (ESNs)

- An approach to learning black-box models of nonlinear systems
- A type of artificial recurrent neural network
  - Feedback loops in their synaptic connections
  - Can maintain activation in absence of input
  - Exhibit dynamic memory
  - Can learn to mimic any target system with arbitrary accuracy

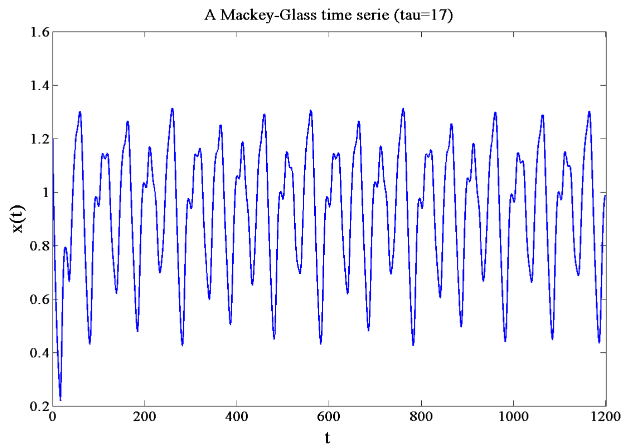


What (was) novel about ESNs?

- At the time of publication (2004), recurrent neural networks were very difficult to train
    - Vanishing Gradient Problem
    - Suboptimal Solutions due to contrained model complexity
- ESNs eschew backprojection
    - Create a large "reservoir" of neurons (50 - 1000) of random connections and only modify the input/output neurons
    - No cyclic dependencies between trained readout connections, and training ESN becomes a simple linear regression task

Demonstrating the utility of ESNs

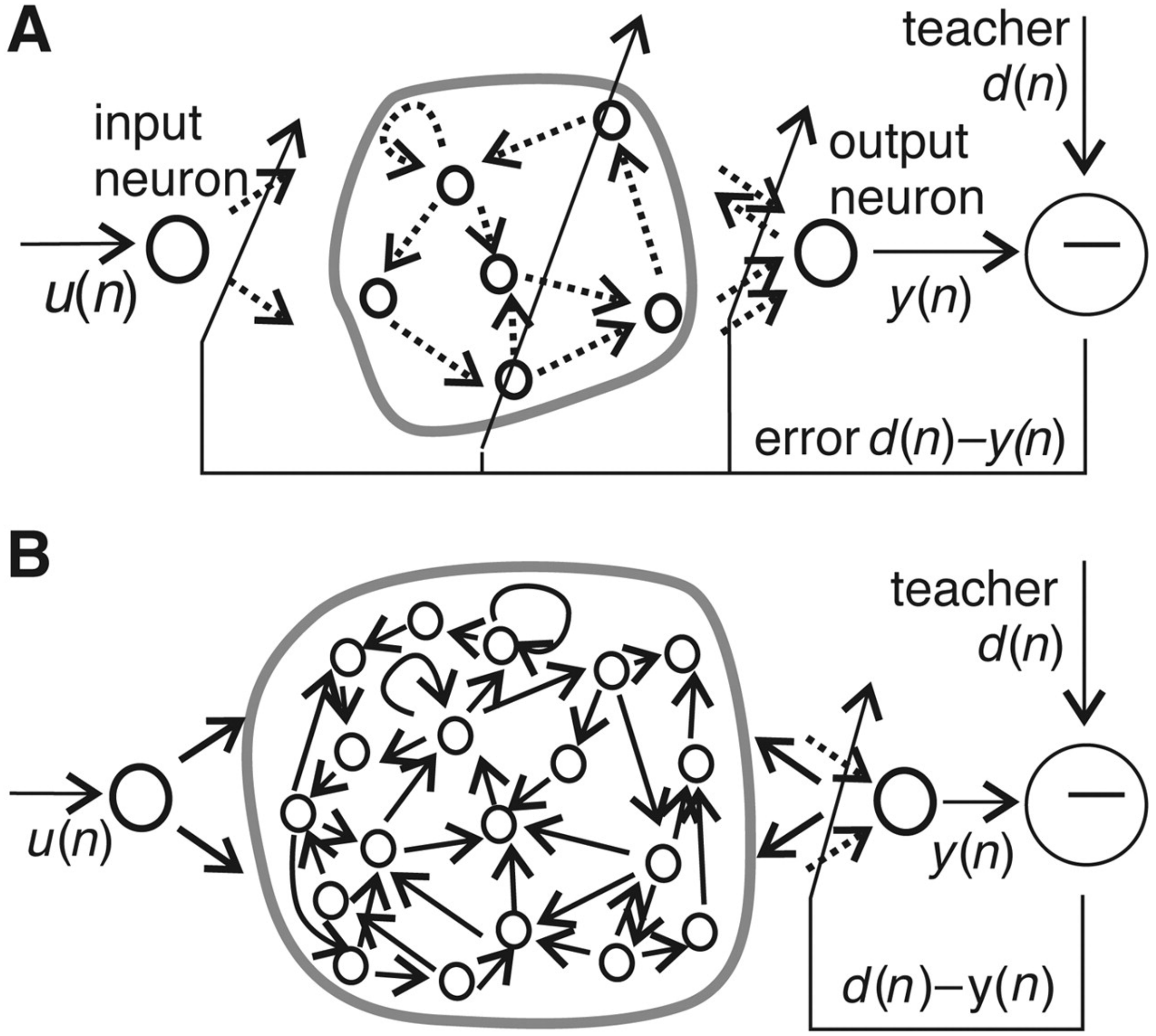
- Demonstrated ESN approach for a Mackey-Glass system (MGS)
  - Standard benchmark system for time series prediction studies
  - Generates an irregular time series
- Two steps for using ESN
  - Train with signal generated from original MGS as teacher signal
  - Use it to predict original signal some steps ahead



Training

- Create 1000 neurons ("reservoir") with sparse interconnections (1%) and one output neuron with random connections back into reservoir
- 3000 step teacher sequence generated from original MGS, and fed into the output neuron
    - Excites the internal neurons through the output feedback connections
    - After initial transient period, exhibit systematic individual variations of the teacher sequence
        - Act as "echo functions" for the driving signal
        - Sparsity of interconnections lets reservoir decompose into many loosely coupled subsystems

Training (continued)



Training (continued)

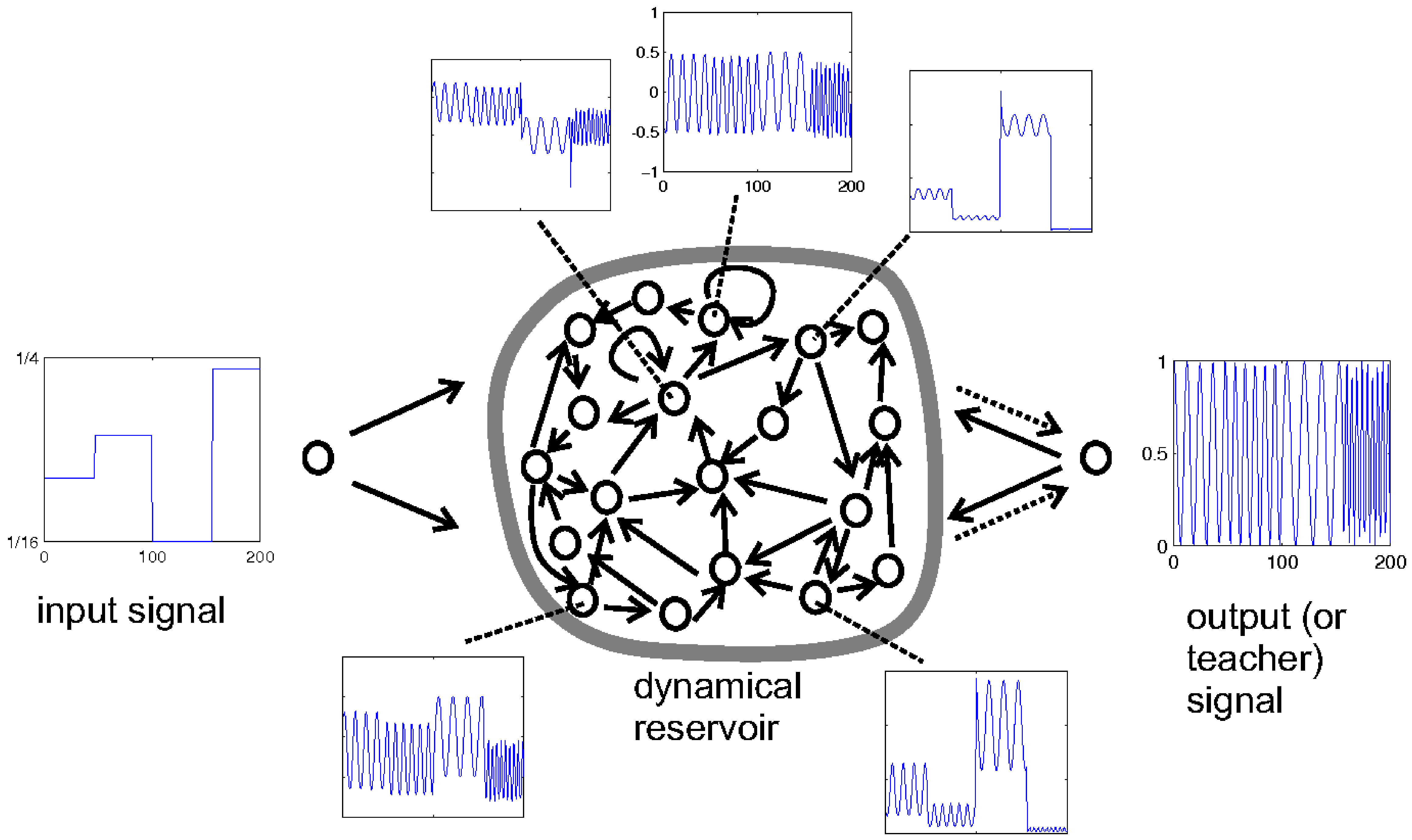
- After time n = 3000, output connection weights, w_i were computed from the last 2000 steps (n = 1001,...,3000) of the training run such that MSE was minimized

$$MSE_{train} = \frac{1}{2000} \sum_{n=1001}^{3000} \left(d(n) - \sum_{i=1}^{1000} w_i x_i(n) \right)^2$$

where $x_i(n)$ is the activation of the ith internal neuron at time n.

- This is simple linear regression!

Another Example (Signal Generator)



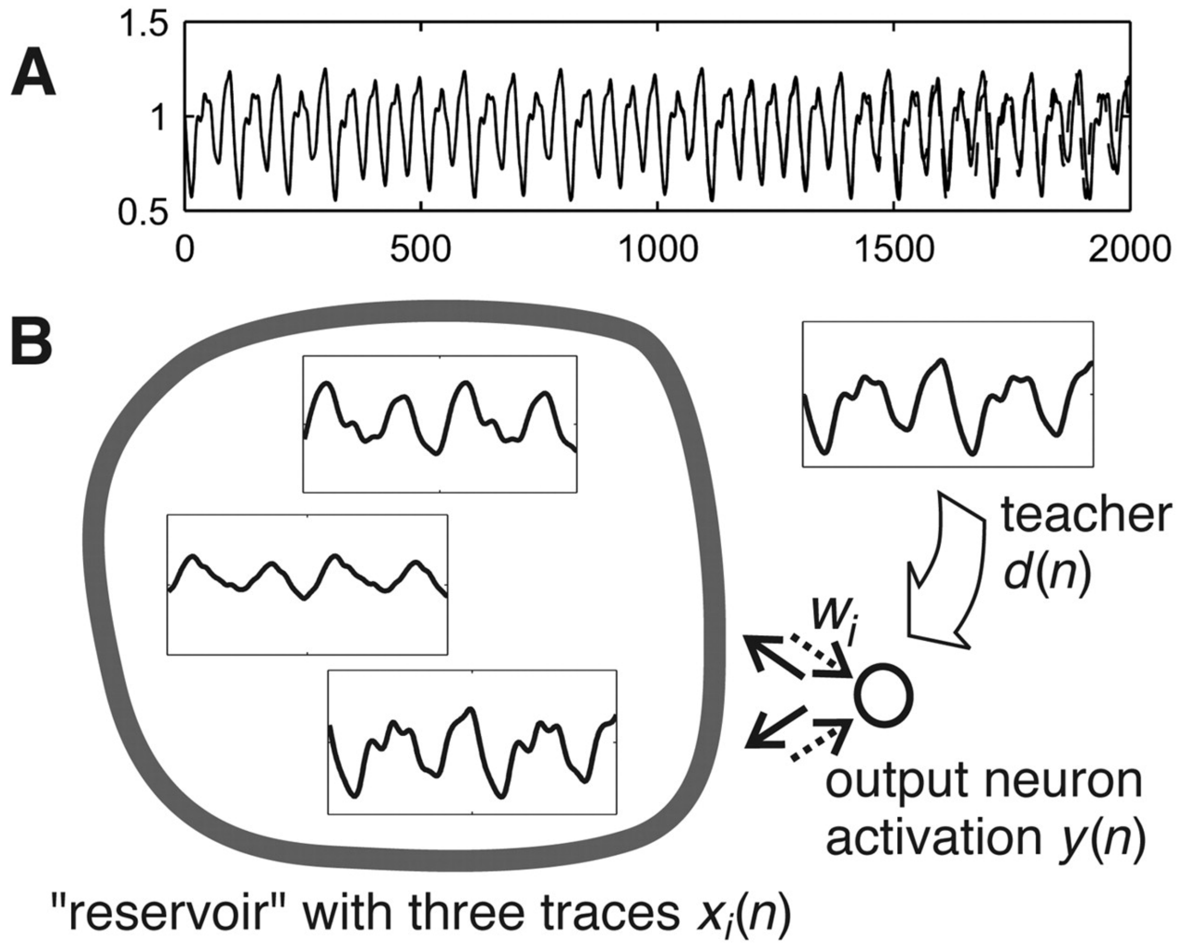
Validation

- Disconnect after 3000 steps and left running freely
- Output created through: $y(n) = \sum_{i=1}^{1000} w_i x_i(n)$
- Looked at next 84-steps of original signal for comparison vs. generated signal.
  - Averaged results over 100 independent trials, and calculated normalize root mean square error:

$$NRMSE = \left(\sum_{j=1}^{100} (d_j(3084) - y_i(3084))^2 / 100\sigma^2 \right)^{1/2} \approx 10^{-4.2}$$

- Improvement in performance by a factor of **700** when compared to previous techniques!
- Deviations noted after about 1300 steps
  - Refinements to model showed improvement factors up to **2400**

MGS Results



A
1.5
1
0.5
0
500
1000
1500
2000
B
teacher
$d(n)$
w_i
output neuron
activation $y(n)$
"reservoir" with three traces $x_i(n)$

Why this jump in accuracy when compared to previous methods?

- ESNs capitalize on a massive short-term memory.
- Authors showed analytically, that under certain conditions, an ESN of size N can "remember" a number of previous inputs that is of the same order of magnitude as N
    - Further reading here
    - Significantly more memory than any other techniques (up until this point)

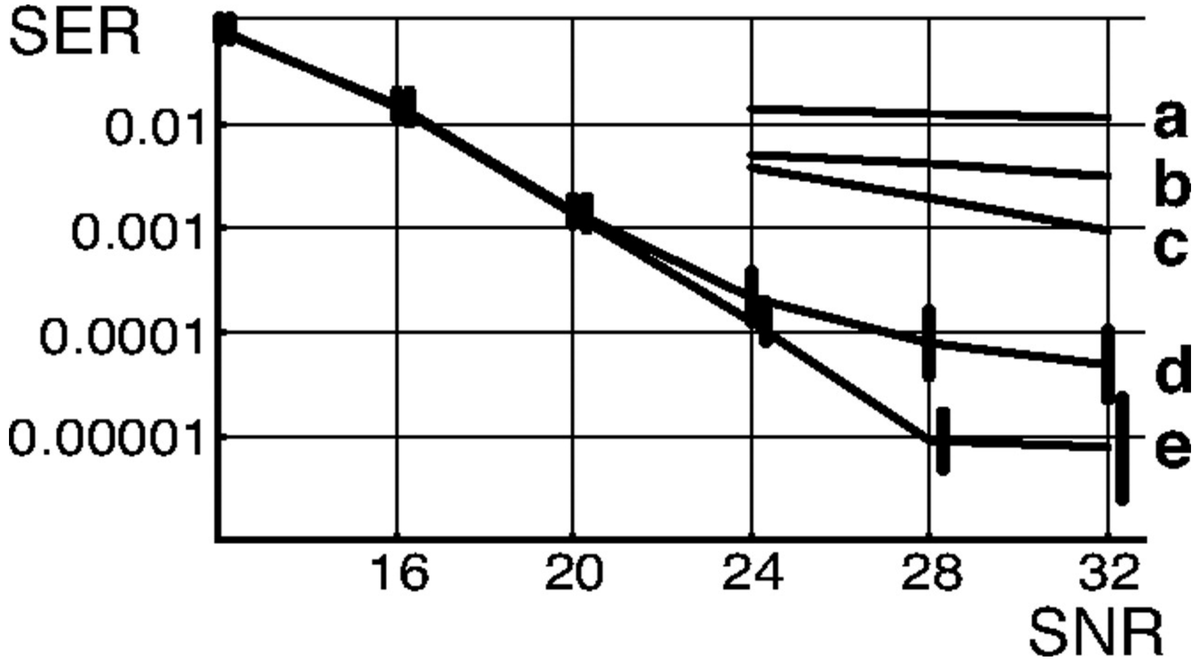
A practical application of ESNs: Wireless Communication Channel Equalization

- A sender wants to communicate a symbol sequence $s(n)$
- This sequence is transformed in an analog envelope signal $d(n)$, then modulated on a high frequency carrier signal and transmitted
- Receiver receives an analog signal $u(n)$, which is a corrupted version of $d(n)$
    - Nonlinear distortion by operating the sender's power amplifier in the high gain region
    - Avoided by running power amplification at below maximum amplification
        - Imposed inefficiency due to nonlinear distortions!
        - Nonlinear modeling can help run wireless communications more efficiently! (Cellphones/Satellites use less power)

Wireless Communication Channel Equalization

- Equalizing filter to restore $u(n)$ as closely as possible to $d(n)$, restored signal called $y(n)$
    - Reconstruct symbol sequence from $y(n)$
- Can we build a better equalizing filter that can handle nonlinear distortions?
    - Authors used message sequences of 5000 symbols for training of ESN and compared against conventional methods of equalization
    - Used 46 neurons to have comparable number of parameters as conventional models

Results for Nonlinear Channel Equalization



Echo State Networks

- ESNs are highly configurable networks than can be used to model all kinds of nonlinear systems
    - Signal and Processing Control, Time Series Prediction, Inverse Modeling, Pattern Generation, Event Detection and Classification, Modeling distributions of stochastic processes, filtering, and nonlinear control.
- Very fast learning (~ few seconds to minutes depending on size of network)
- Simple training process through linear regression
- Biologically Inspired
    - ESN principles not uncommon in biological networks
    - Similar to models of prefrontal cortex function in sensory-motor sequencing tasks, models of birdsong, models of cerebellum, and general computational models of neural oscillators

Limitations and the Present

- ESNs have been mostly supplanted by deep learning techniques, namely LSTM Networks
  - ESNs were developed at a time when Vanishing Gradients were still an issue
  - Compared to LSTMs, ESNs tend to perform worse on higher dimensional modeling tasks
- Still useful?
  - ESNs still popular when blended with non-digital computational substates
    - Optical microchips, neuromorphic microchips, artificial soft limbs, etc.
    - ESNs useful because interconnected ensemble of such devices (reservoir) can be built, and thus trained with ESN principles.