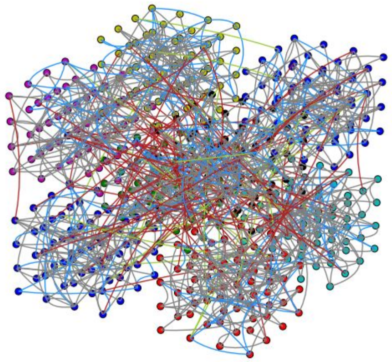
An Overview of Reservoir Computing



Neelesh Kumar

CS 425/525-Brain Inspired Computing

Lecture 13
28-March-2019

Motivation

Solving temporal problems:

- Prediction (weather, dynamic systems, financial data)
- Robotics (Planning)
- Vision and Speech

Common Approaches: Time delayed feed forward networks, Recurrent Networks, Attractor based Networks

Motivation

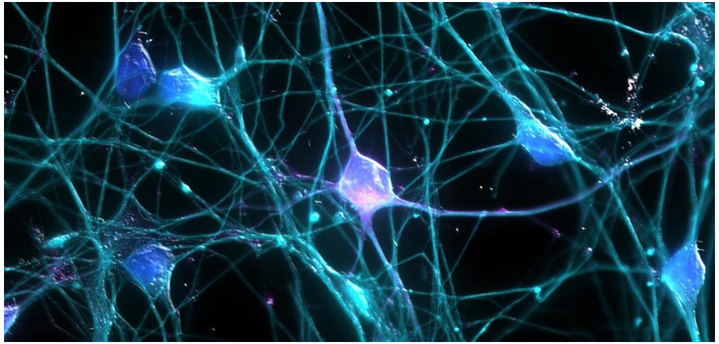
Common Approaches:

- Time delayed feed forward networks
  - Not a natural way to represent time
- Recurrent Networks
  - Difficult to train, Vanishing/Exploding Gradients
- Attractor based Networks
  - Require storing stable states

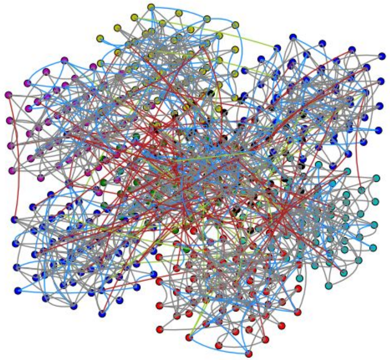
Goal

- Real time computations
- Not task-specific
- Parallel Computations

Inspiration: Neural Microcircuits

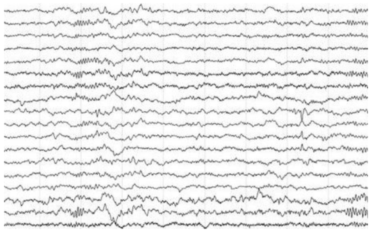


Neural Microcircuits

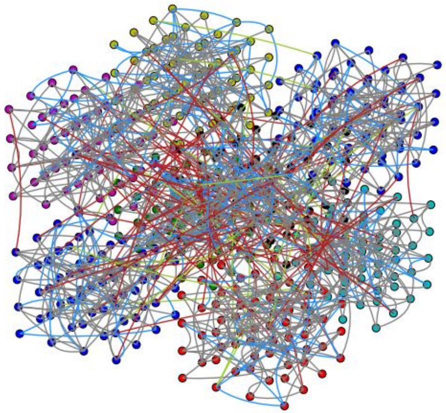


Recurrent randomly connected Network

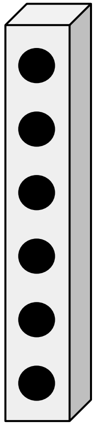
Proposed Approach



Input Signal



Reservoir



Readout Layer

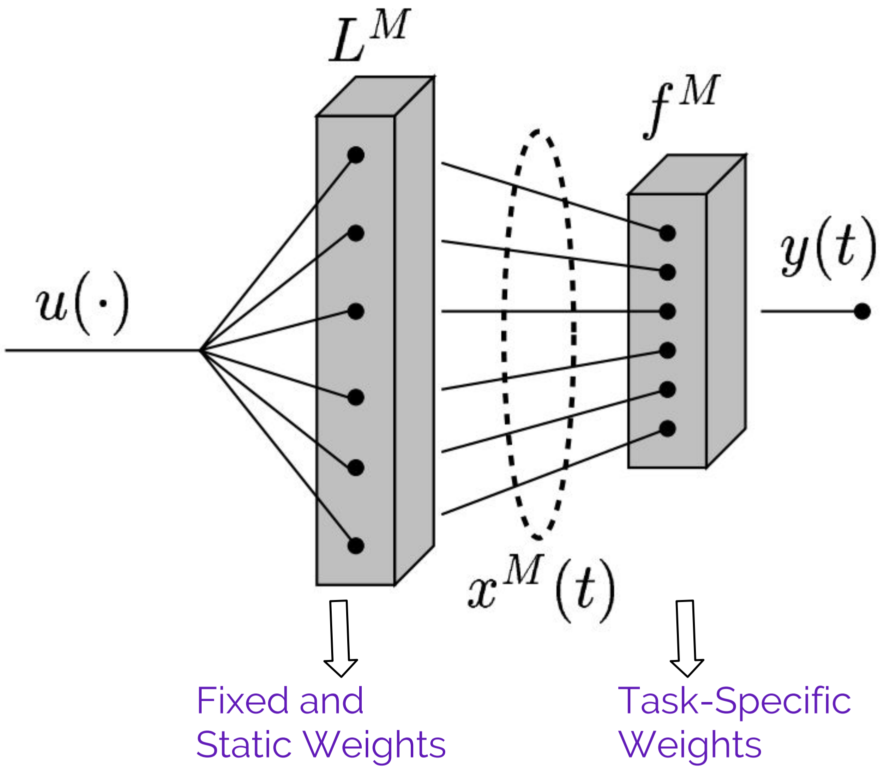
Output State

Analog Neurons: Echo State Network

Spiking Neurons: Liquid State Machine

Reservoir Computing

Liquid State Machines



- Time varying input signal **u**
- Target output **y**
- Machine **M** that maps **u** to **y** in 2 stages:
  - Network $\mathbf{L^M}$ generates "liquid state" $\mathbf{x^M(t)}$

    $$x^M(t) = (L^M u)(t)$$

  - Memoryless readout transforms $\mathbf{x^M}$ to **y**

    $$y(t) = f^M(x^M(t))$$

Theoretical Justifications

Universal Computational Power:

Any filter **F** that is any map from functions of time **u** to functions of time **y**, that is time invariant and has fading memory can be approximated by machines from class M to any degree of precision.

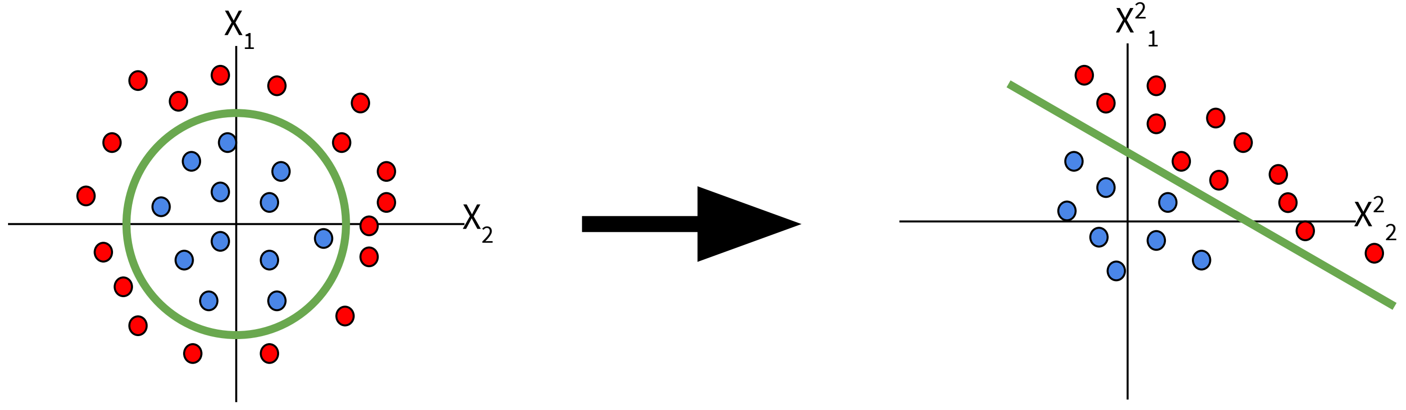
Properties of Filter:

- Time invariant: A temporal shift in the input causes a temporal shift in the output by same amount.
- Fading Memory: Current state depends upon input from some finite time window in the past.

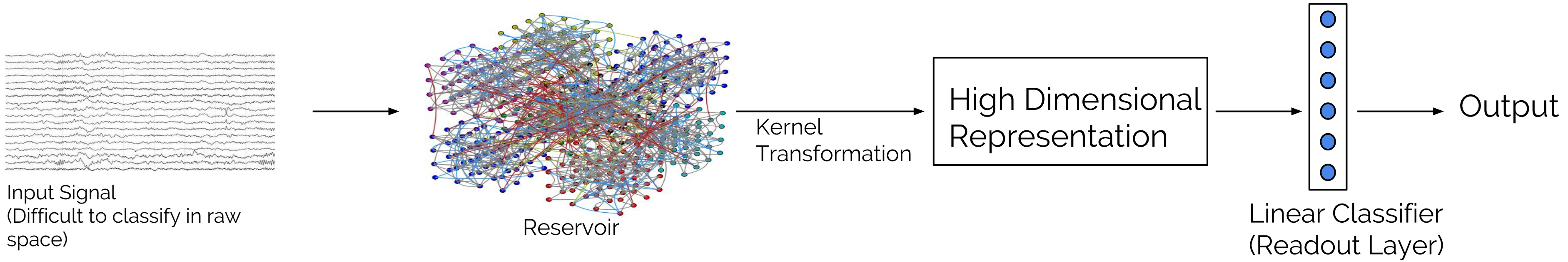
Necessary and Sufficient Conditions for universal computational power:

- Separation
  - Between the trajectories of internal states of the system for two different input streams.
- Approximation
  - Capability of Readout to distinguish and transform different internal states of the system into given target outputs.

Intuition: Kernel Transformation



$$X_1^2 + X_2^2 = 1$$



Implementation of LSM

- Randomly connected recurrent circuit of 135 IF neurons, 20% Inhibitory
- Single column: 135 neurons located on integer points of a 15x3x3 column in space
- Probability of synaptic connection from neuron a to neuron b:

$$P = C.e^{-(D(a,b)/\lambda)^2}$$

  - D(a,b): Eucledian distance between a and b
  - λ: Parameter that controls average number of connections and average distance between connected neurons
- C = 0.3 (EE), 0.2 (EI), 0.4 (IE), 0.1 (II)
- Readout Neurons: can be regular IF neurons or Perceptrons
- Deciding connectivity structure and synaptic weights: Later

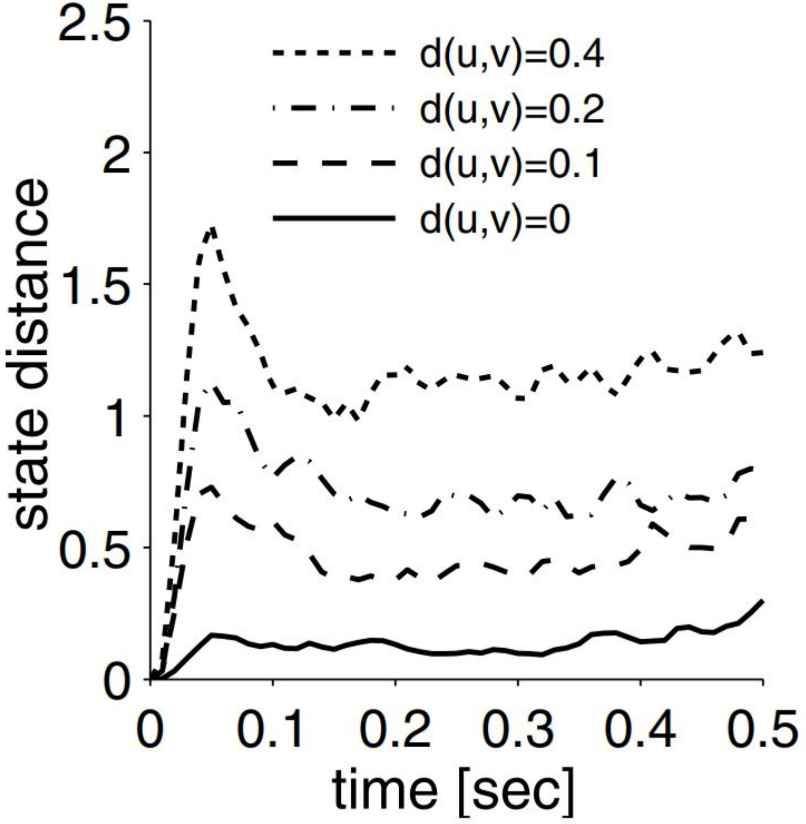
Measuring Separation

Experiment:

- Apply two different input spike trains.
- Measure the distance between the reservoir states for the two inputs.
- Reservoir state is a vector containing activity of each neuron in the reservoir.

Observation:

- State distance increases with increasing distance between input spike trains.
- Distance between liquids states well above noise level.



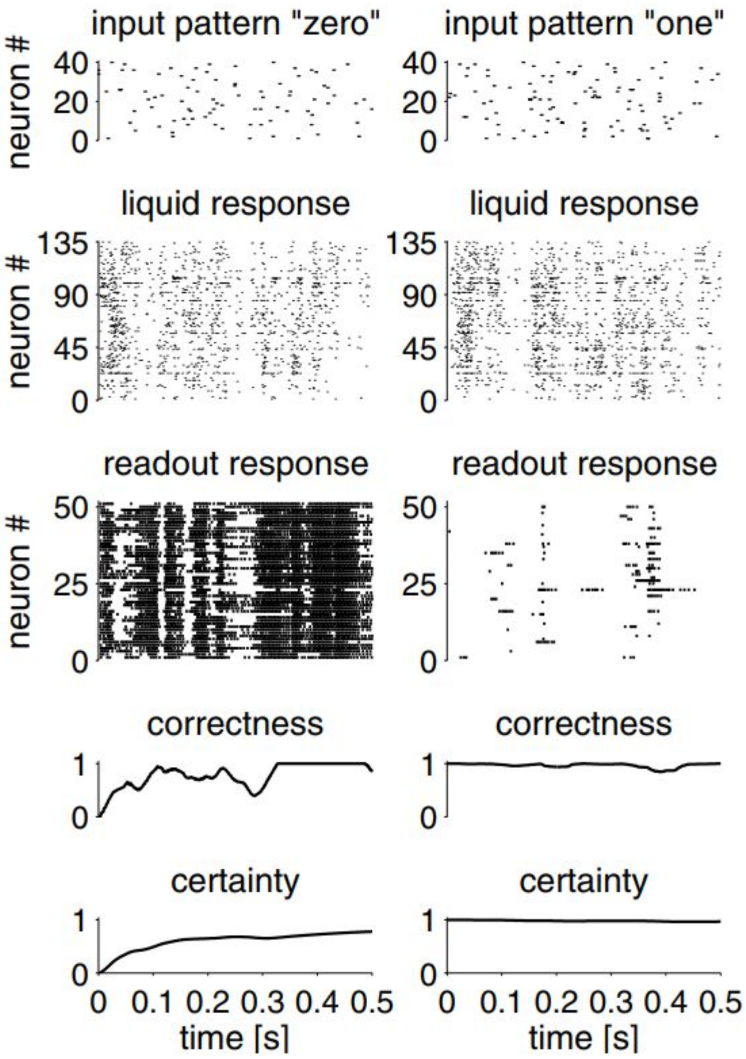
Classifying Speech

Experiment:

- Represent spoken words by noise-corrupted spatiotemporal spike patterns.
- Binary Classification.

Observation:

- Output of network available at any time and usually correct as soon as the liquid state absorbs enough information.
- Robust to noise.



Fading Memory

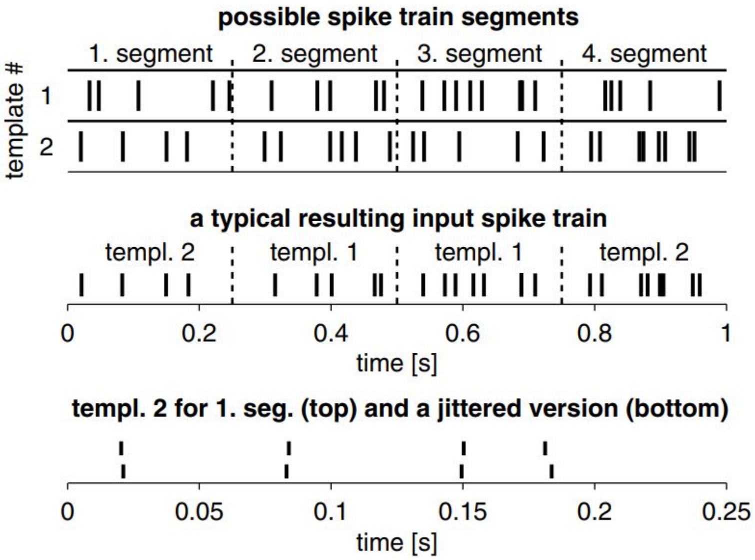
Experiment:

- Encode all information in interspike intervals of single input spike train.
- Create a spike train consisting of blocks from the two templates.
- Classify which template does each block come from at the end of the presentation of spike train.

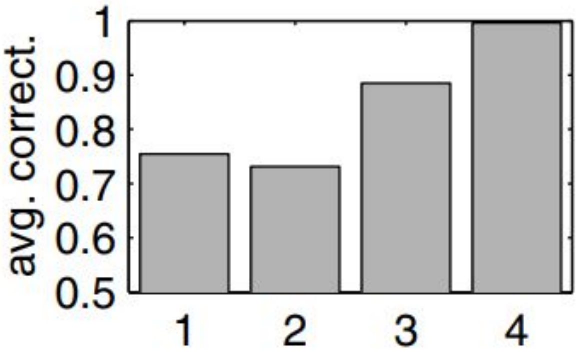
Observation:

- Information about inputs that occurred several hundred ms ago can be recovered even after the input was overwritten.
- Short-term dynamics.

Experiment



Result



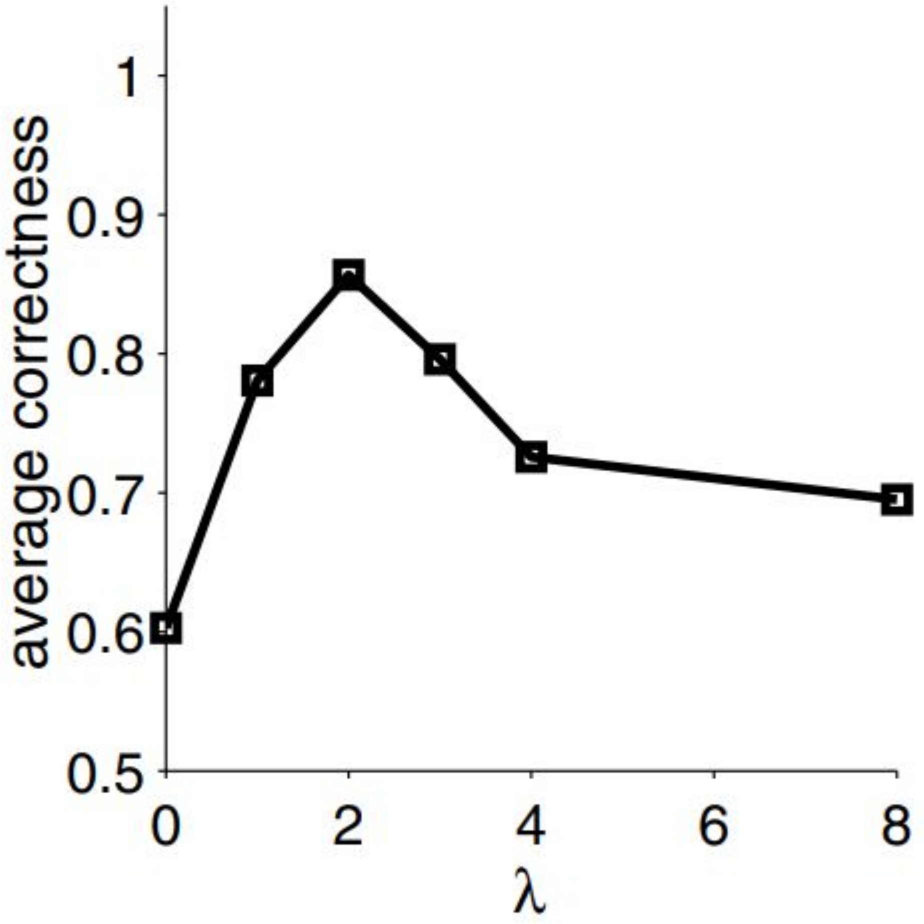
Dependence on Connectivity

Experiment:

- Speech recognition for different values of connectivity parameter.

Observation:

- Recurrent connections are essential for achieving separation in neural microcircuits.
- Large values of connectivity parameter decrease the performance.



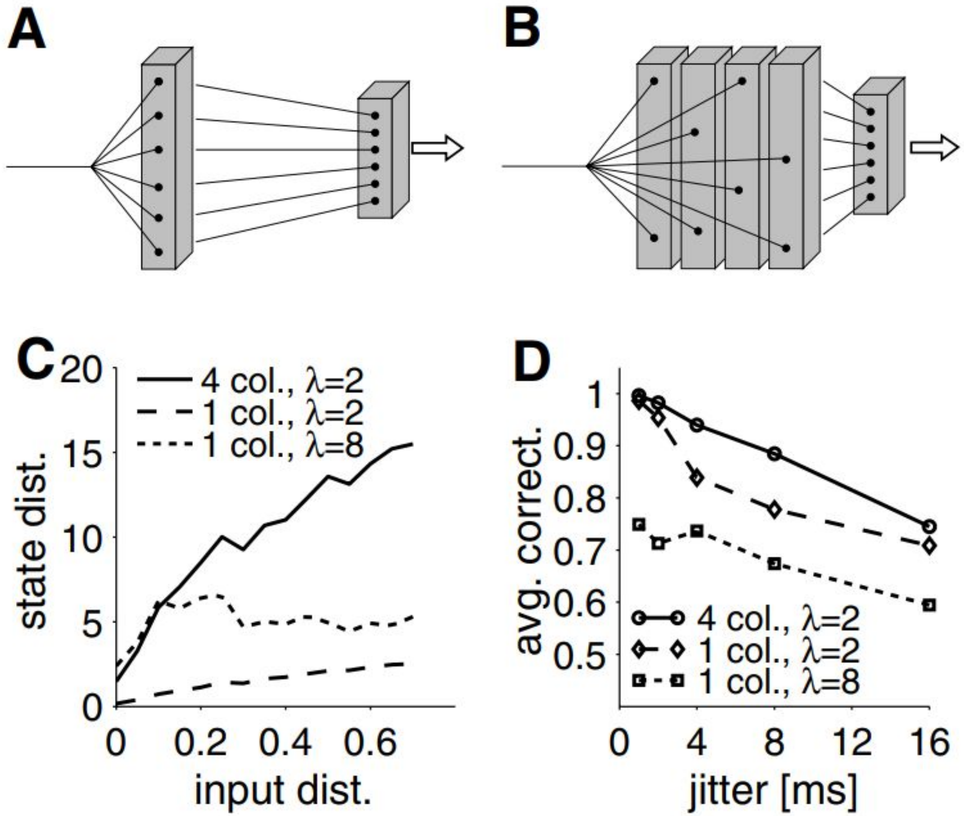
Adding More Neurons

Experiment:

- Add more columns.
- Add more connections.

Observation:

- High internal connectivity achieves better separation but poor performance.
- Several columns (not-interconnected) with low internal connectivity yield better performance.



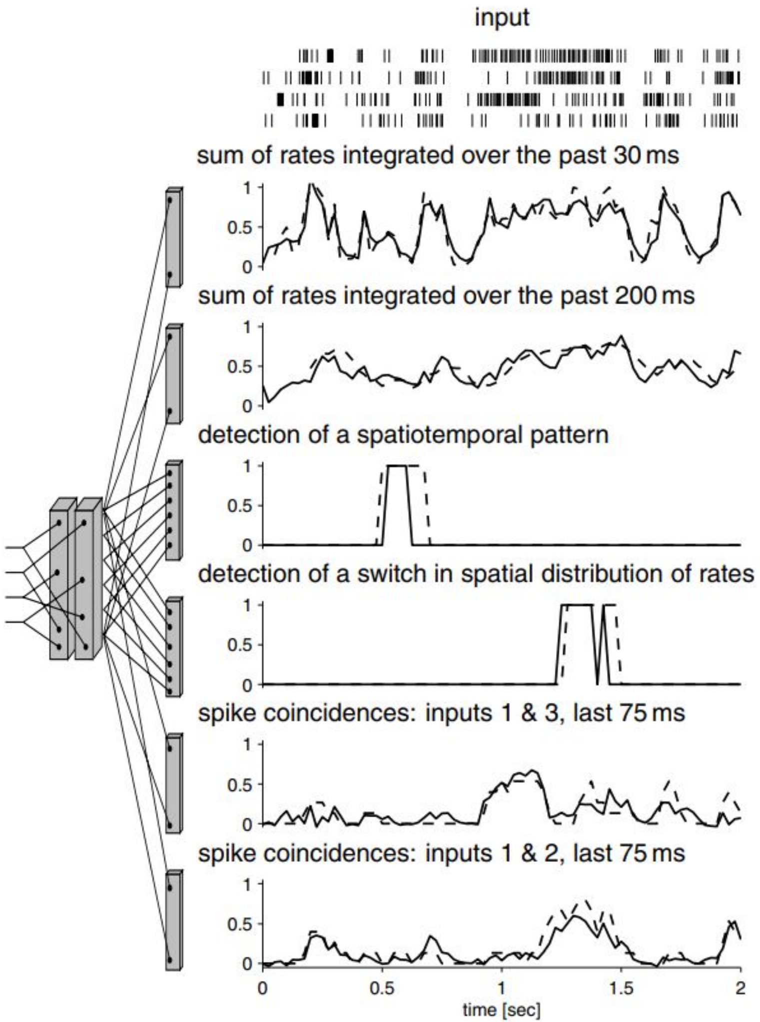
Parallel Computing

Experiment:

- Inject multiple spike trains into the liquid.
- Train multiple readout layers to perform different tasks in parallel.

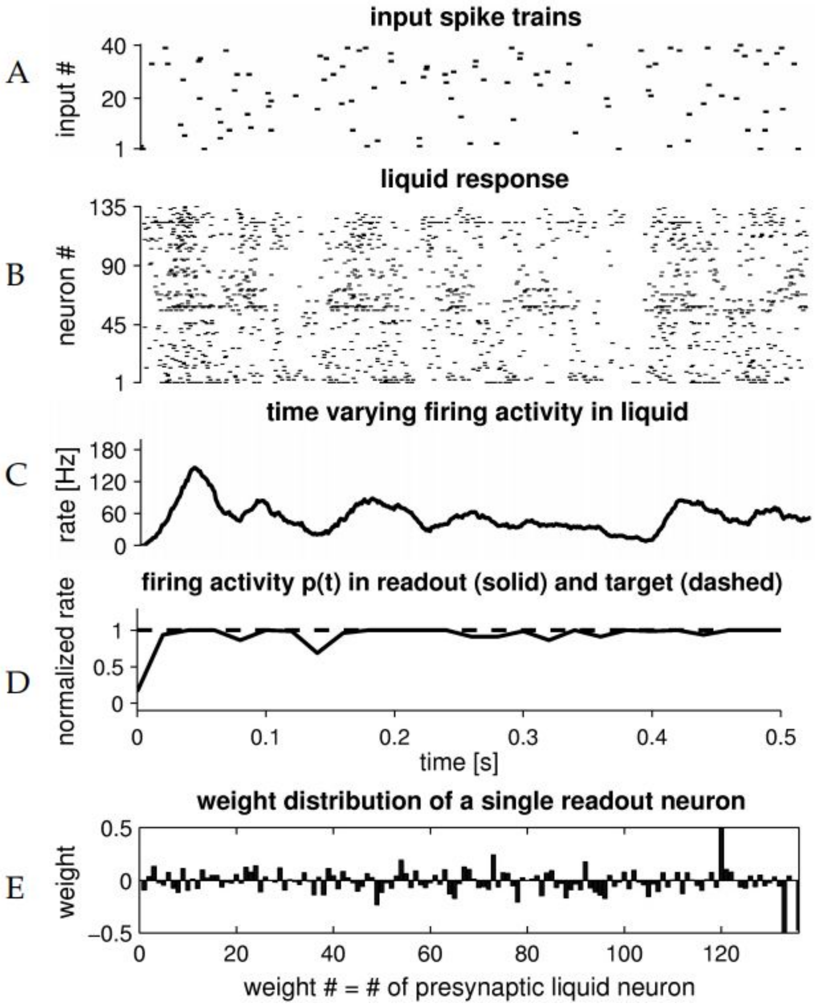
Observation:

- Each task can be performed in real time with high accuracy.



Readout

- Readout response much more stable than liquid response.
- Readout samples only few liquid neurons in response to input.



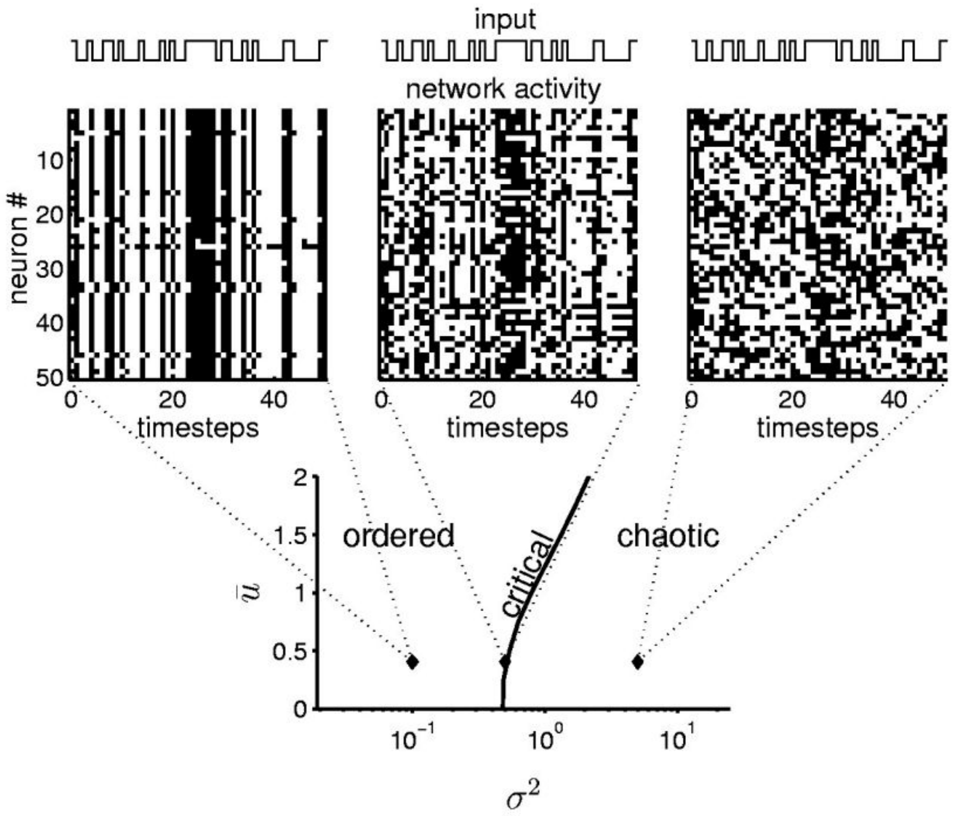
Computational Power and Dynamics

- Computation: An algorithm that assigns outputs to inputs.
- Computational Power: Complexity and diversity of associations of inputs to outputs.
- Analyzing computational power of a random network requires analyzing its dynamics:
  - Depending on connectivity, the networks can exhibit very different types of dynamics from completely ordered to chaotic.

A network is chaotic if arbitrary small differences in initial network state are highly amplified and do not vanish.

A totally ordered network forgets immediately about initial network state and the current network state is determined to a large extent by the current input.

Phase Transitions in Neural Network



Maximum computational capability at the edge of chaos (critical state)

Why Operate at Critical State

3 Viewpoints:

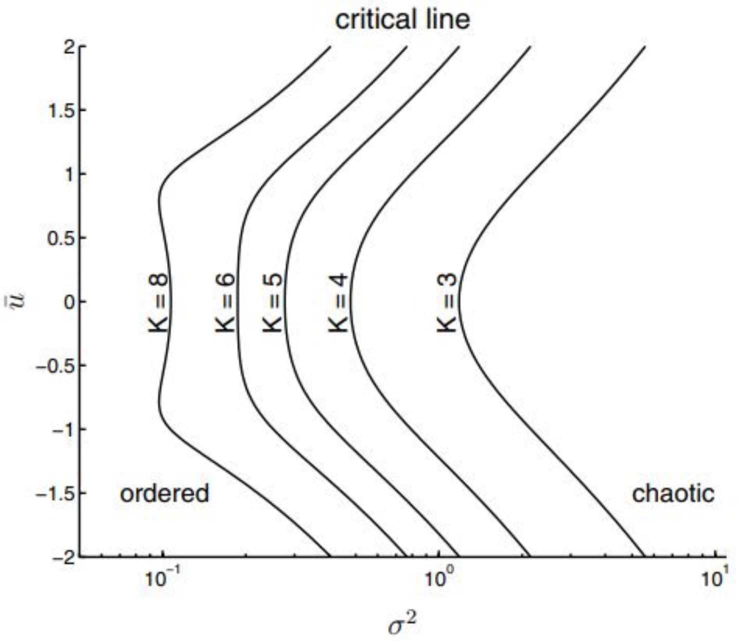
1. Separation:
   a. In an ordered state, response of the network is same to different inputs.
   b. In a chaotic state, network is very sensitive to difference in inputs (even very small difference).
2. Fading Memory:
   a. Want network dynamics to be in a regime where memory about recent inputs is available and past inputs do not interfere with that memory.
   b. Ordered phase: memory needed is not provided by the reservoir.
   c. Chaotic phase: older memories interfere with the recent ones.
3. Complexity of Readout layer:
   a. Near critical line, the encoding of information is such that a linear readout layer is sufficient.

Identifying Points of Criticality

No single approach- depends on how you define criticality.

An example:

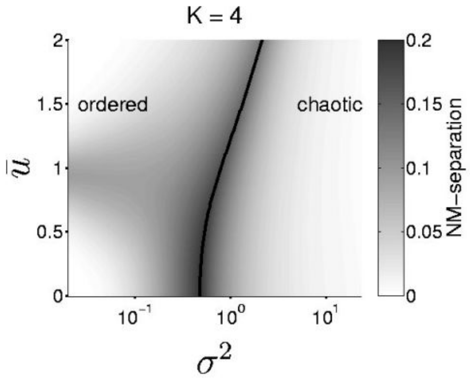
- Define notion of distance between network states (eg: Hamming distance)
- Quantifying the evolution of distance with time
- As time progresses, the distance increases for chaotic state, and goes to 0 for ordered states.
- Critical points are in between.

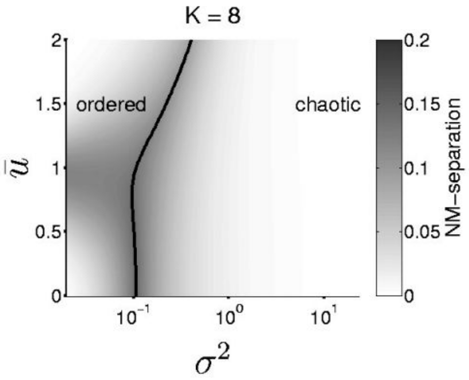


Network State depends upon the connectivity of the network, synaptic weight distribution, and the input signal statistics.

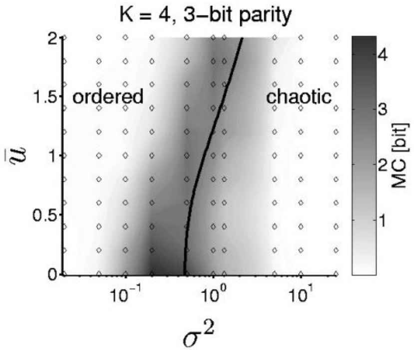
Computational Power at Critical Line

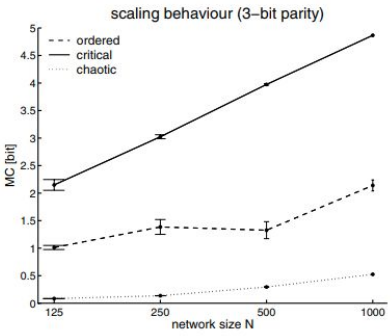
Separation:



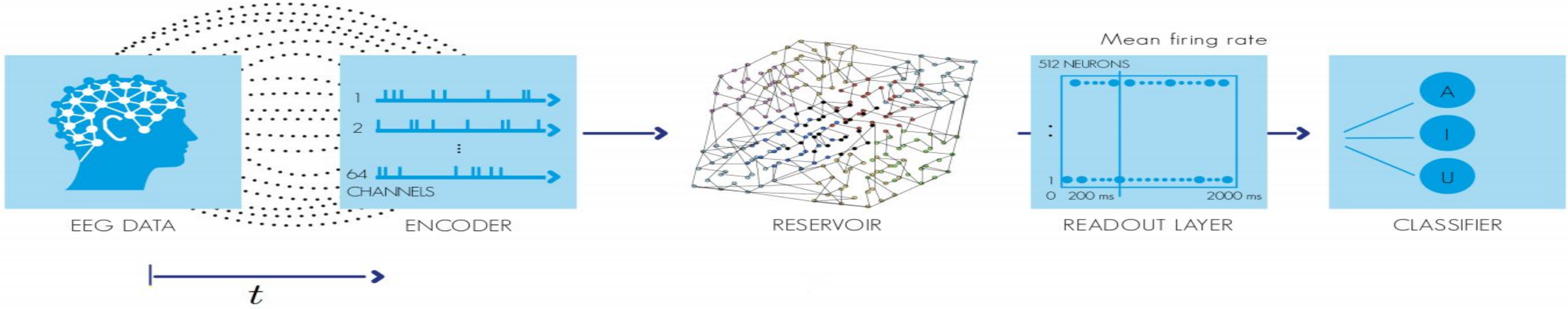


Computational Power (Memory Capacity):





Application: EEG Classification



Subjects	DNN	RNN
1	55.7%	88.8%
2	42.9%	67.7%
3	45.6%	87.7%
4	58.4%	84.6%
5	70.8%	78.5%
6	43.7%	79.3%
7	30.8%	88.2%
8	40.6%	90.4%
Average	48.6%	83.2%

Questions?

[1] **Real-Time Computing Without Stable States**
Maass W., Natchlager T., Markram H.

[2] **An Overview of Reservoir Computing**
Schrauwen B., Verstraeten D., Van Campenhout J.

[3] **Regulation Toward Self-Organized Criticality in a Recurrent Spiking Neural Reservoir**
Brodeur S., Rouat J.

[4] **Real-Time Computation at the Edge of Chaos in Recurrent Neural Networks**
Bertschinger N., Natschlager T.

[5] **Classification of Auditory Stimuli From EEG Signals With a Regulated Recurrent Neural Network Reservoir**
Plourde et. al.