Real-time computing without stable states: a new framework for neural computation based on perturbations

Wolfgang Maass
Thomas Natschlager
Henry Markram

Presented by Qiong Zhao

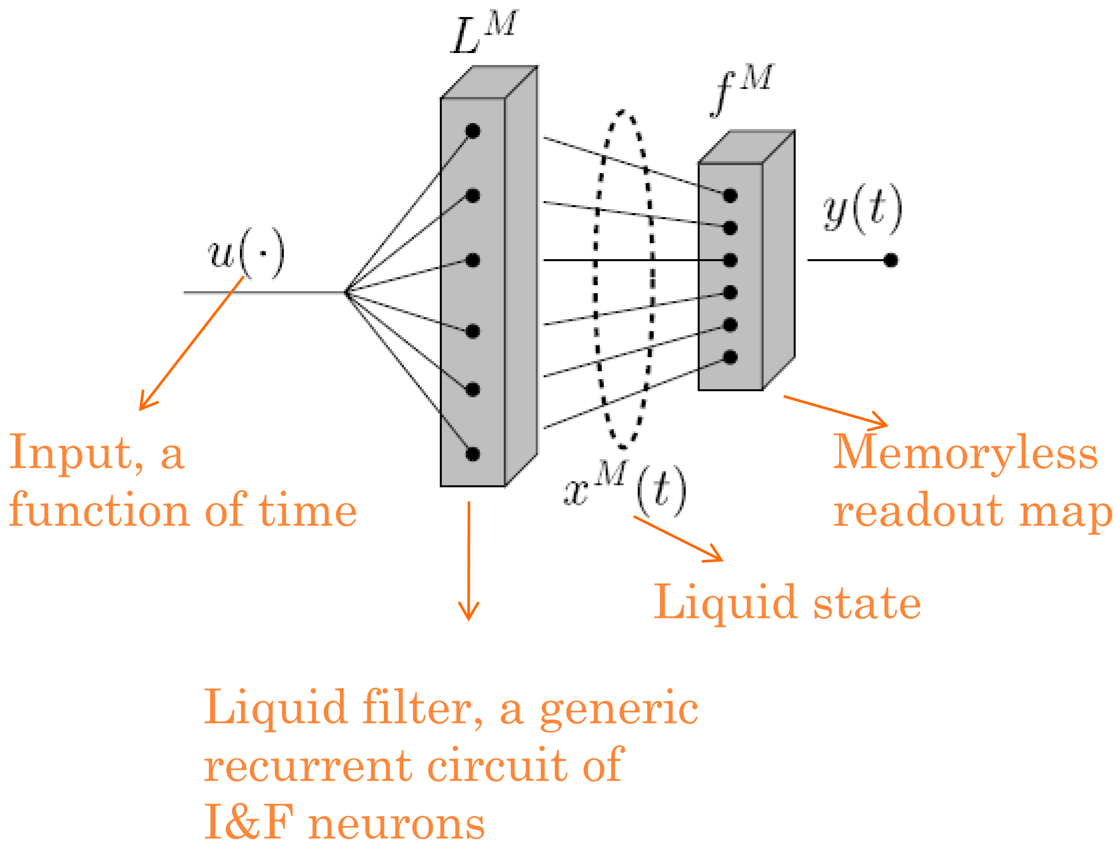
April 28th, 2010

OUTLINE

- Introduction
- Implementations of LSMs
- Exploring the computational power
- Parallel computing
- Readout-assigned equivalent states
- Conclusion

INTRODUCTION

- Liquid state machine



- Mathematical terms

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

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

- Advantages
  - Does not require a task-dependent construction of neural circuits
  - Can be implemented on generic evolved or found recurrent circuitry
  - Stable internal states are not required
  - Supports parallel computing

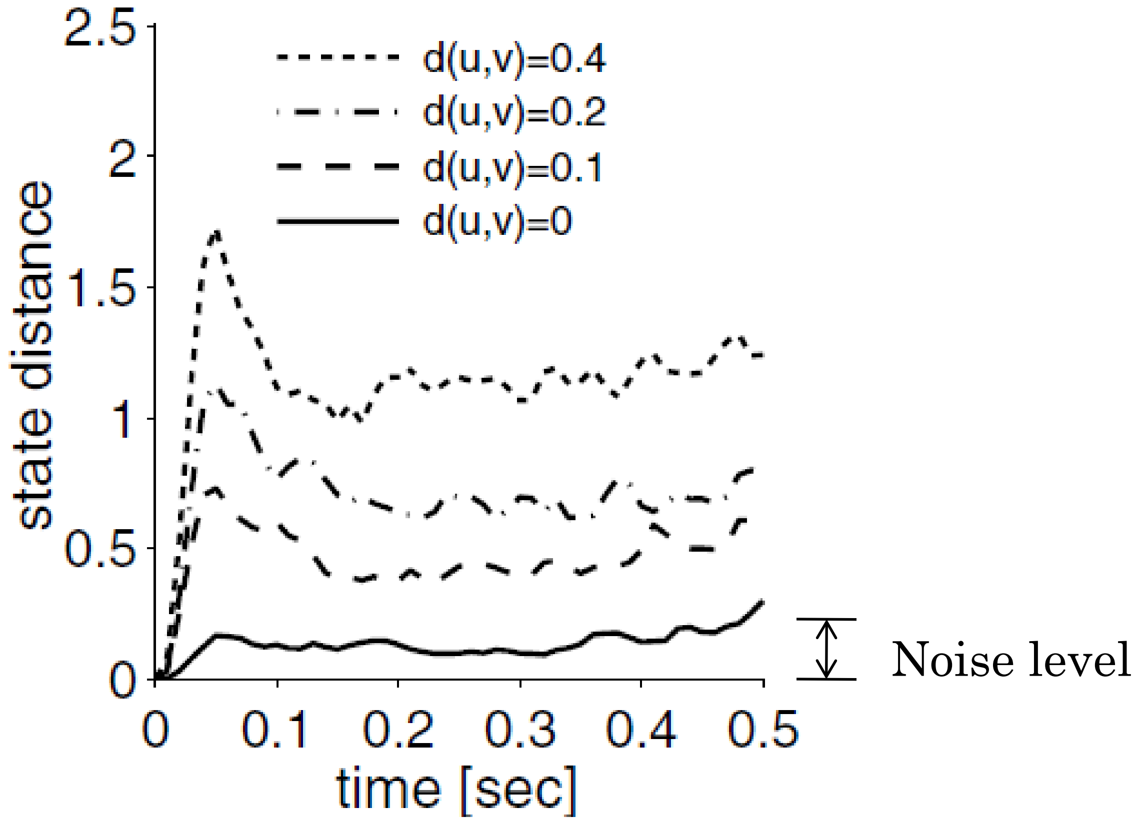
- Necessary and sufficient conditions for powerful real-time computing on perturbations:

  - Separation property (SP)

    for $d(u,v) \neq 0, \quad d(x_u^M, x_v^M) \neq 0$

  - Approximation property (AP)

    A class CF of functions has the AP if any continuous function can be approximated by one of the functions in CF.

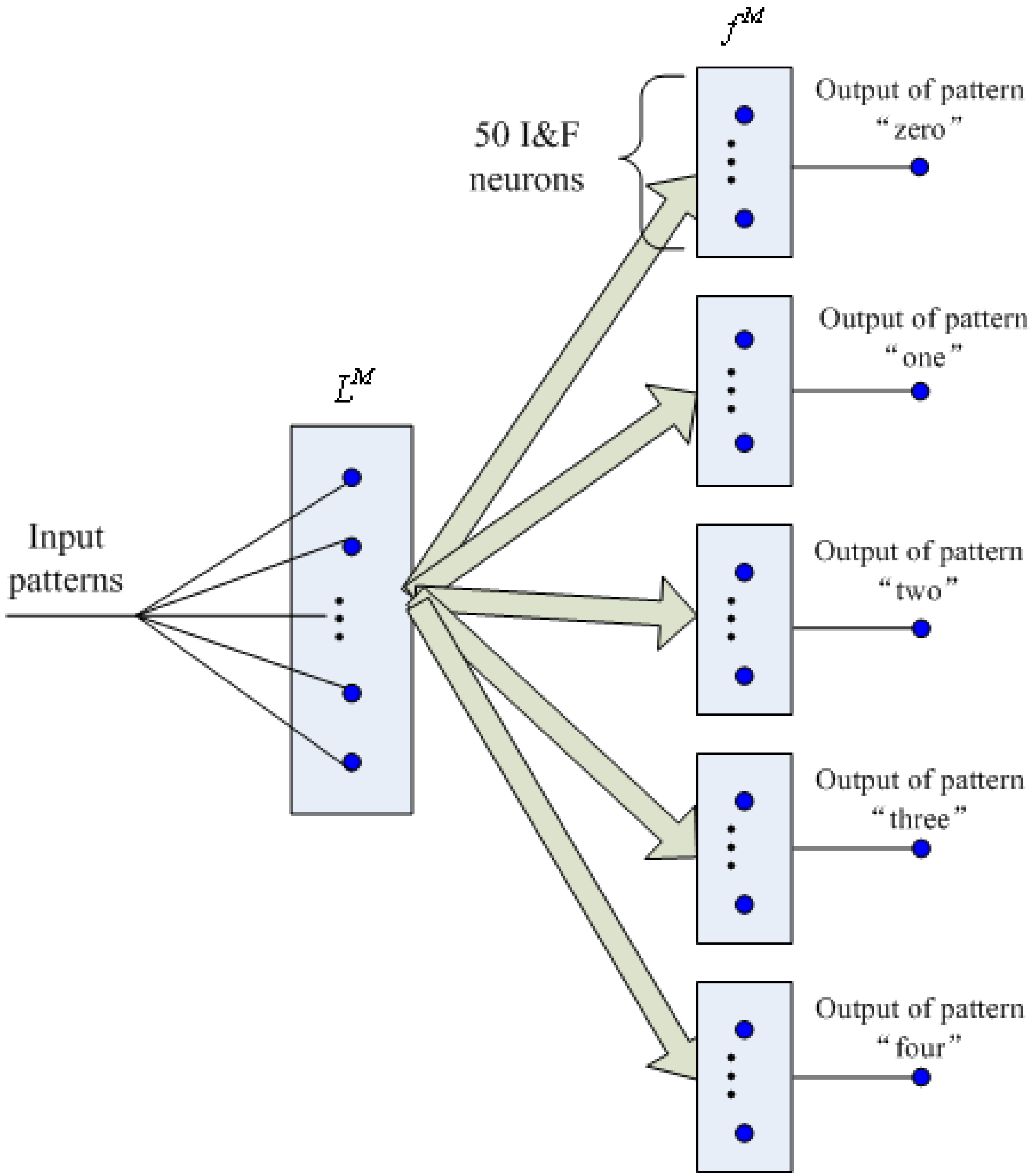


- Clearly shows the separation property SP
- The state distance increases with the input distance

Exploring the computational power - test 1

- Test case (Hopfield & Brody, 2001)

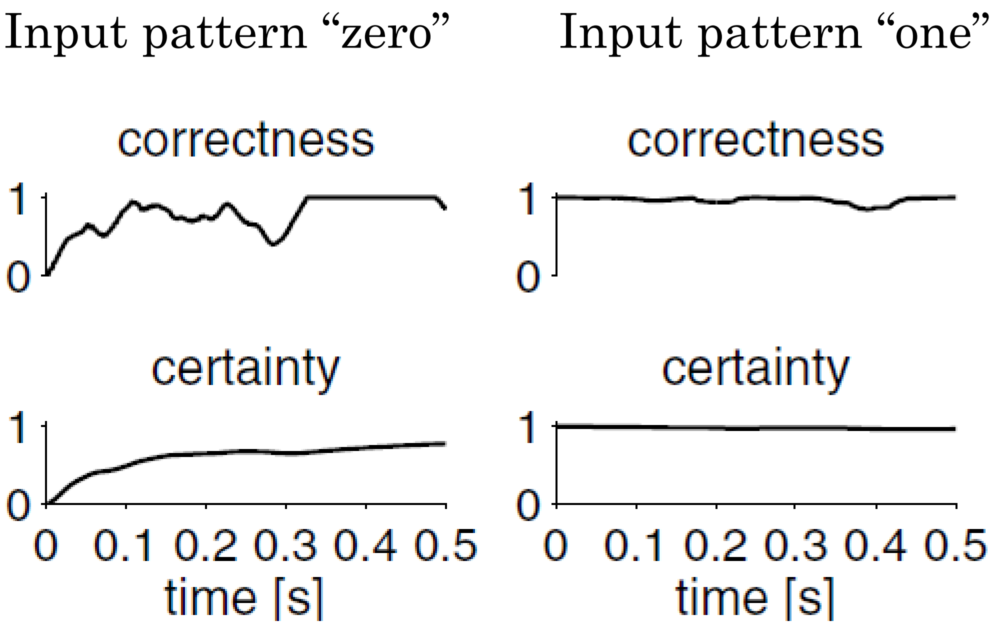
  Five randomly drawn patterns (“zero”, “one” , “two”,…), each consisting of 40 parallel Poisson spike trains over 0.5 sec.
- Network model



f^M
50 I&F neurons
Output of pattern "zero"
Output of pattern "one"
Output of pattern "two"
Output of pattern "three"
Output of pattern "four"
L^M
Input patterns

- Advantages
  - Allow several spikes per input channel.
  - The output of this network was available at any time and was usually correct as soon as the liquid state of the neural circuit had absorbed enough information about the input.
- Definition of correctness

$$correctness = 1 - \left| \underbrace{p(t)}_{\text{actual}} - \underbrace{y(t)}_{\text{target}} \right|$$

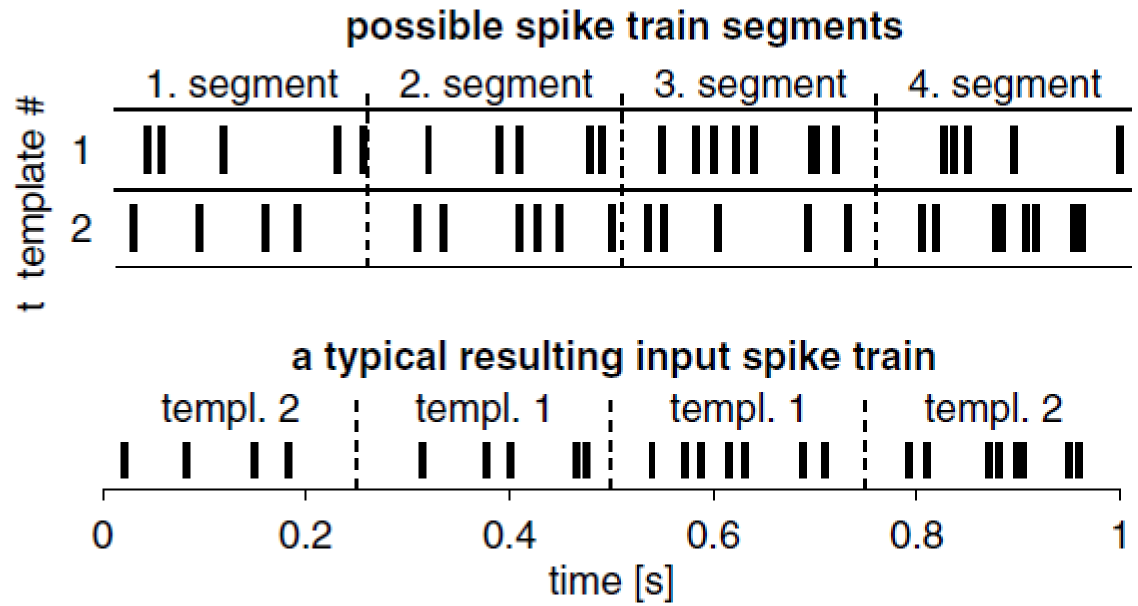


- Pattern “zero”: correctness starts at level of 0; readout pool is supposed to become active.
- Pattern “one”: correctness starts at level of 1; readout pool starts in an inactive state.

Exploring the computational power - test 2

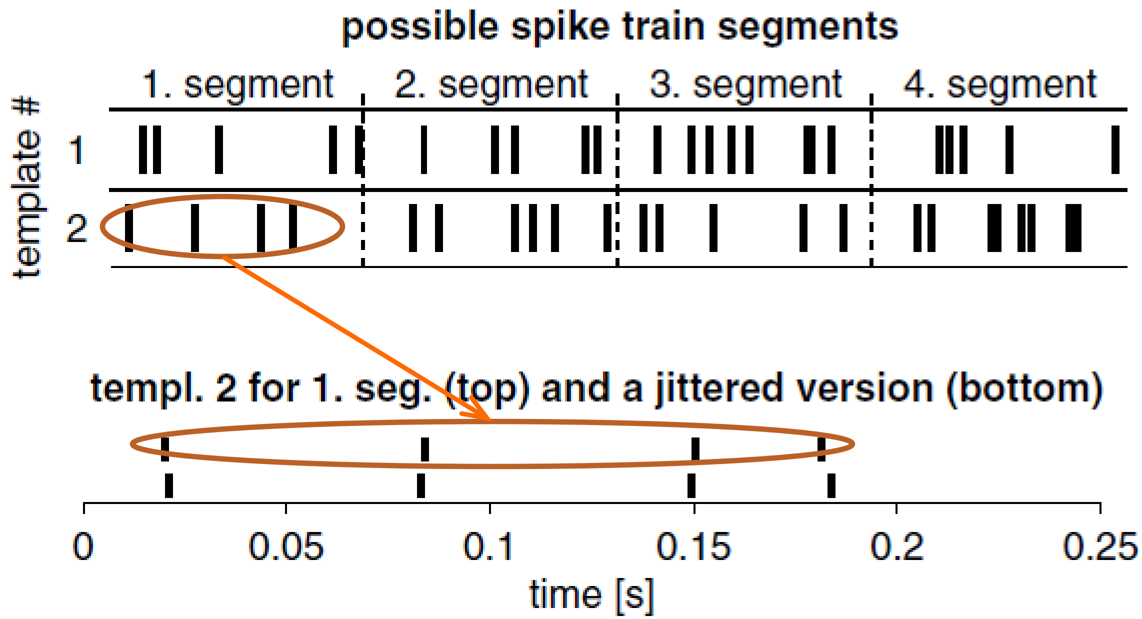
- Giving a constant output for a time-varying liquid state is a serious challenge for an LSM.
- It cannot rely on attractor states, and the memoryless readout has to transform the transient and continuously changing liquid states into a stable output.
- This test is supposed to explore the limits of this neural implementation of an LSM.

- Test case



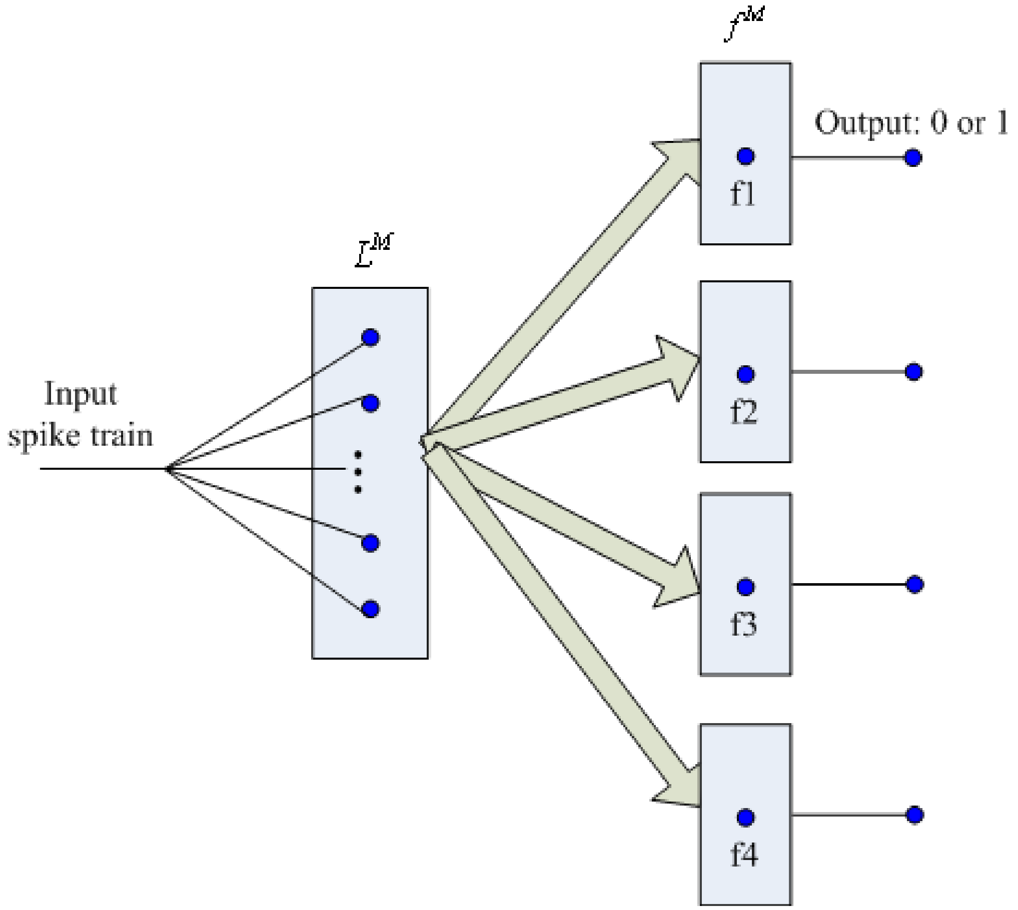
Spike trains of length 1000ms and consist of four segments of 250ms. Each segment could be constructed from either template 1 or 2.

- Jitter



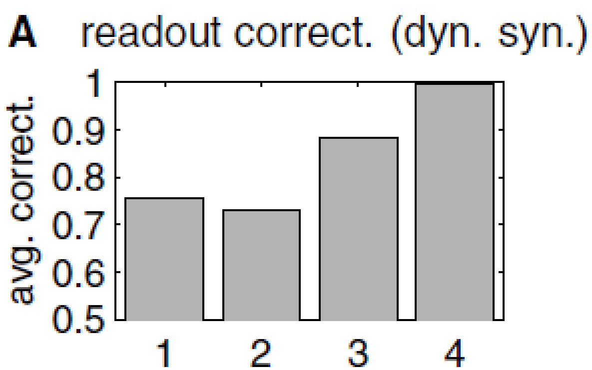
Each spike was moved by an amount drawn from a Gaussian distribution with mean 0 and a SD.

- Network model



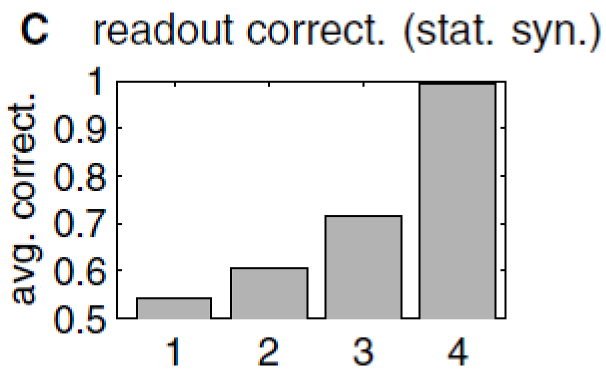
- Correctness

Experimental group



Segment

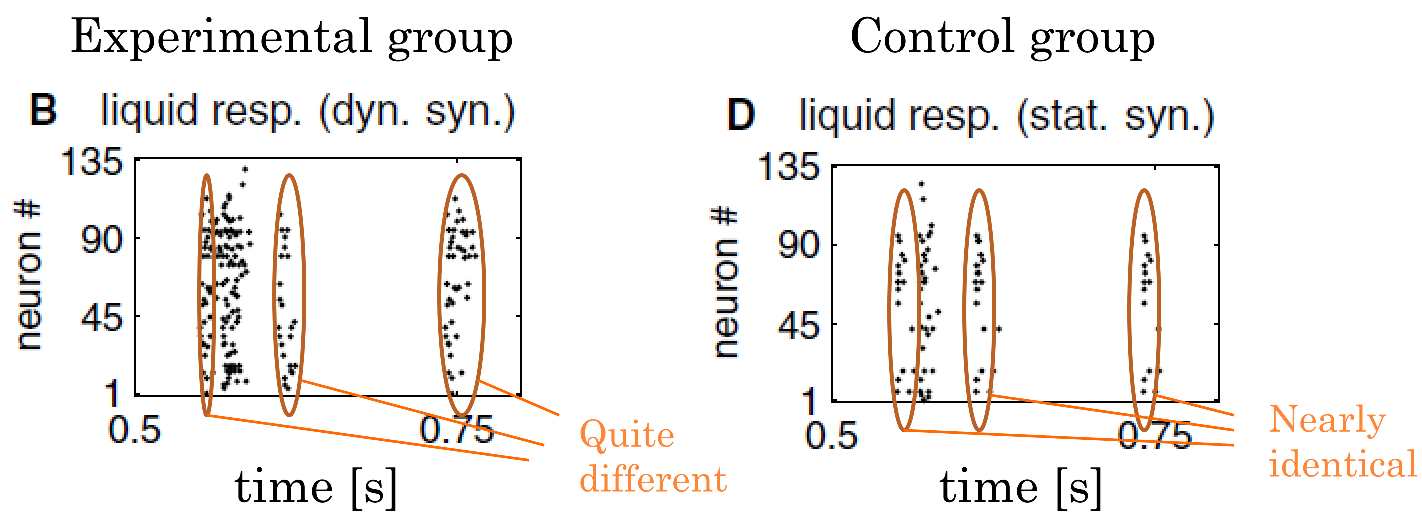
Control group



Segment

The liquid circuit without dynamic synapses contained less information about the earlier input segments.

- Firing activity in the liquid circuit



Dynamic synapses endow circuits with the capability to process new input differently depending on the context set by preceding input.

This helps in the integration of information.

Exploring the computational power - test 3

- Test case

  The same task as task 2.

  Six types of liquid circuits were tested, each consisting of 135 I&F neurons but with different values of the parameter λ.

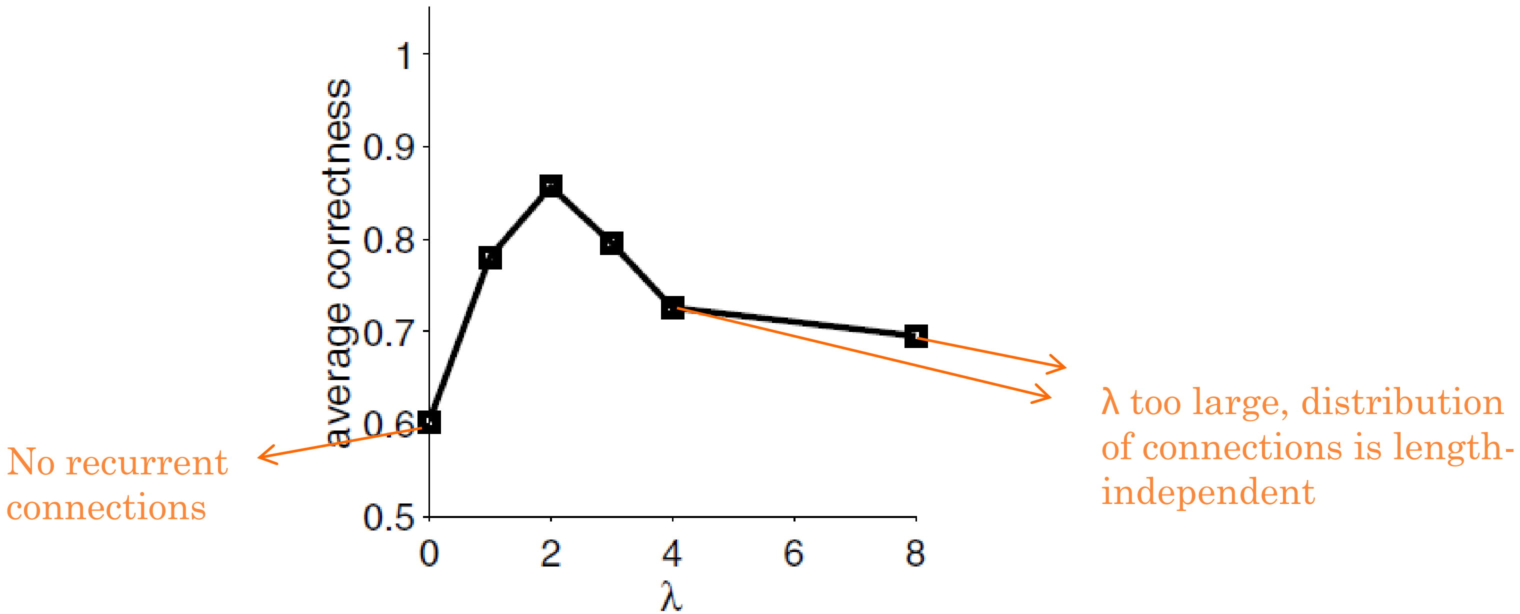
  (λ regulates the average number of connections and the average spatial length of connections.

  The probability of a synaptic connection from neuron a to neuron b is defined as

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

  where $D(a,b)$ is the distance between a and b)

- Correctness vs λ



Recurrent connections are essential for achieving a satisfactory SP. But values of λ that are too large facilitate chaotic behavior.

Adding computational power

- Computational power increases with any improvement in SP or AP.

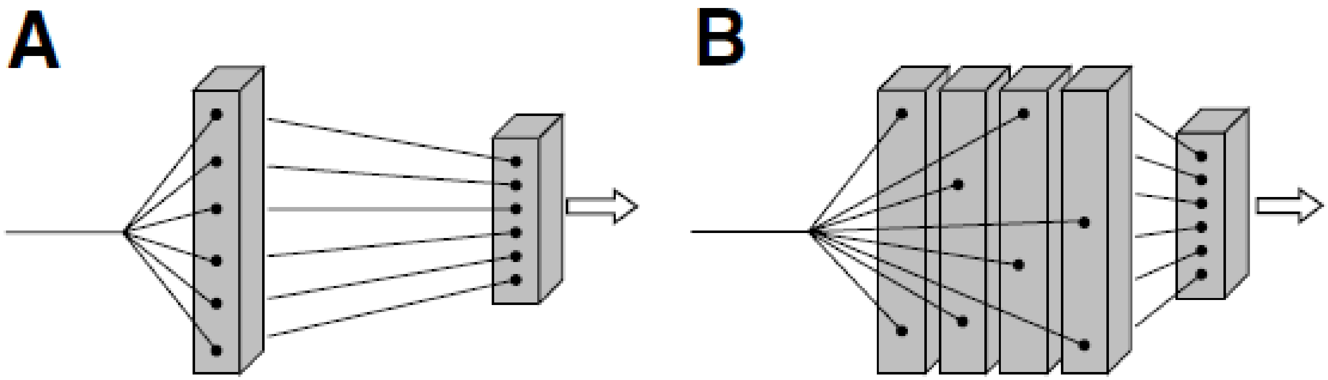
- The primary limitation in performance lay in SP, since AP was already close to optimal.

- Several ways of improving SP:

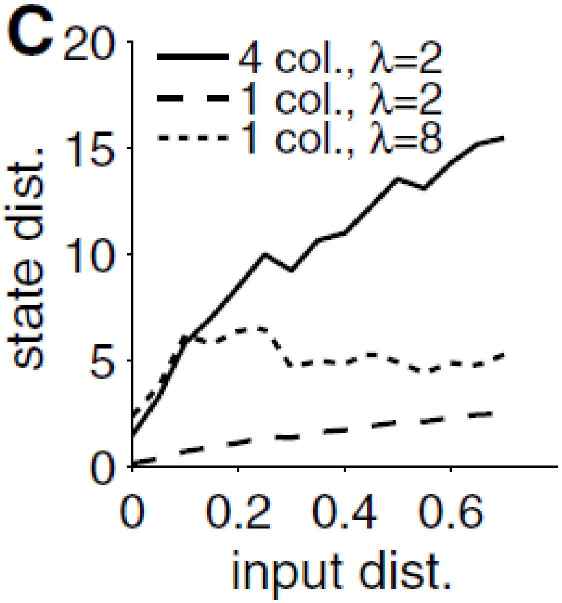
  incorporating neuron diversity, implementing specific synaptic architectures, altering microcircuit connectivity, or recruiting more columns.

ADDING COMPUTATIONAL POWER

- Recruiting additional columns



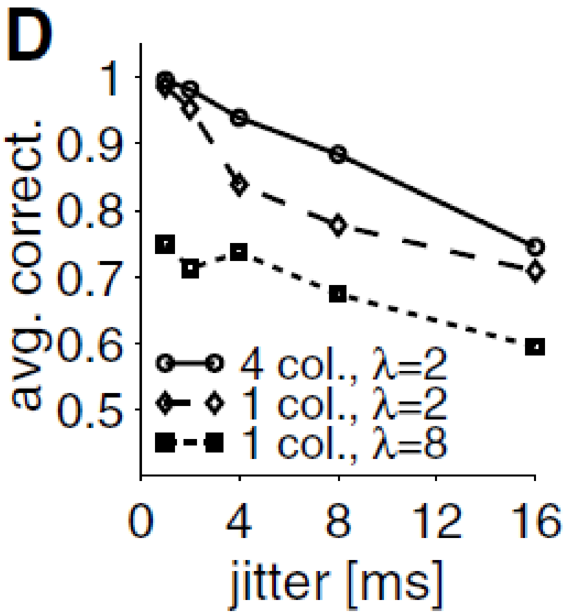
- Separation property



λ=2: 4 column performed better than 1 column.

λ=8: Achieved higher separation, but lead to chaotic behavior.

- Correctness



The 4 column model outperformed the other two.

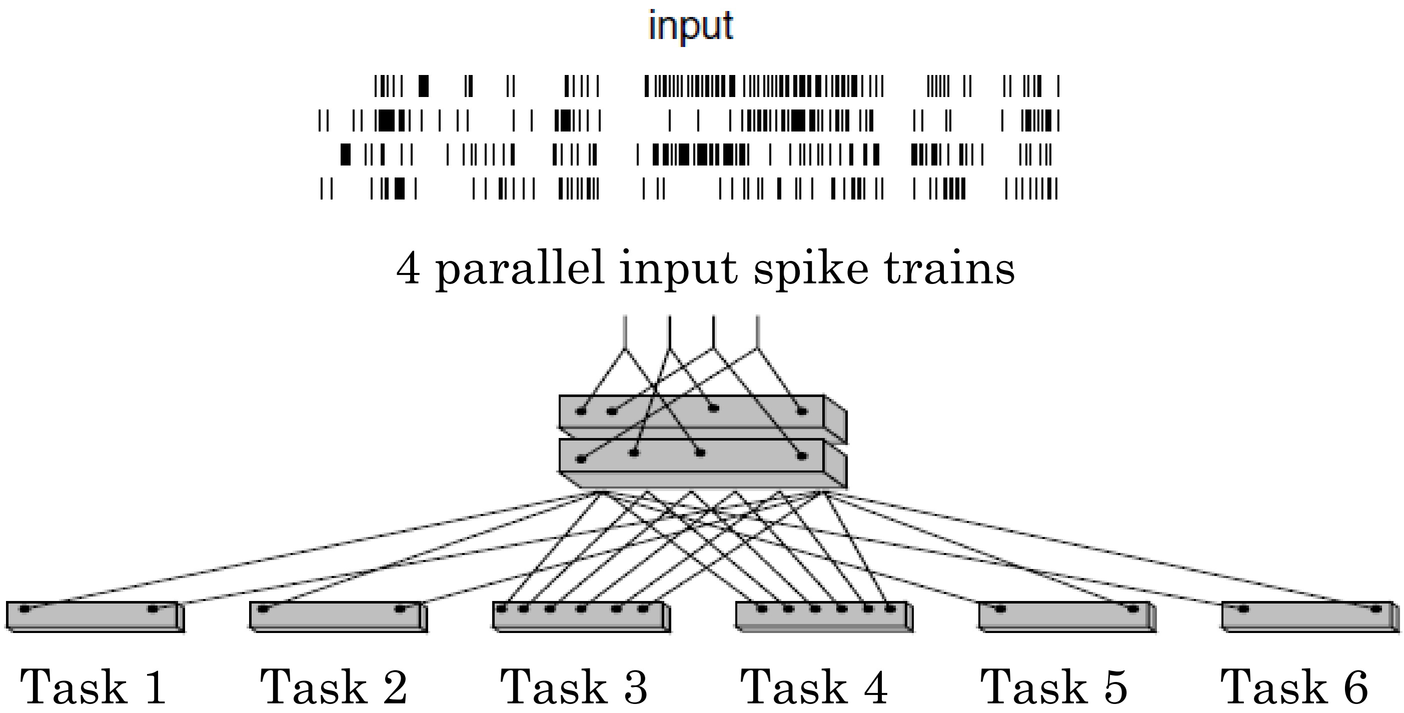
Parallel computing in real-time on novel inputs

- The LSM supports parallel computing in real time.
- Test model:

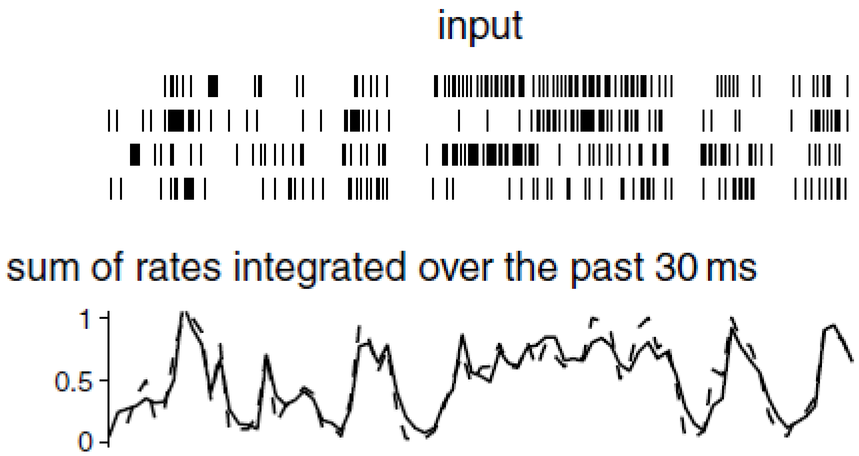
  Four input spike trains and six readout modules;

  The liquid module consists of two columns, one with $\lambda=2$, the other with $\lambda=8$.

- Structure

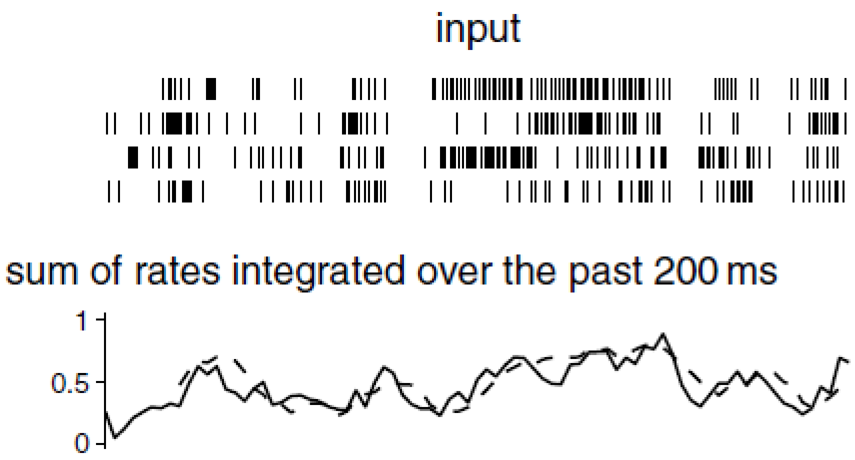

input
4 parallel input spike trains
Task 1
Task 2
Task 3
Task 4
Task 5
Task 6

- Task 1 (readout 1)



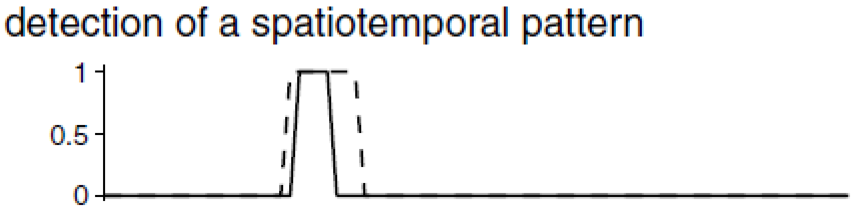
Represent the sum of rates: at time t, output the sum of firing rates of all four input spike trains within the last 30ms.

- Task 2 (readout 2)



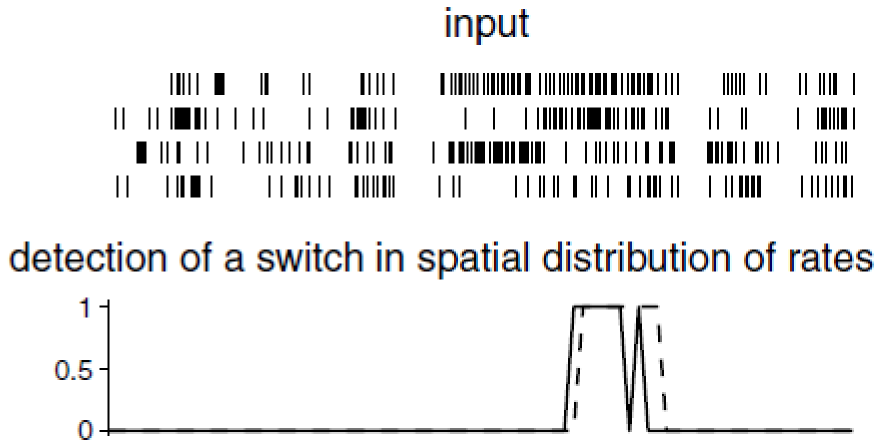
Represent the integral of the sum of rates: at time t, output the total activity in all four inputs integrated over the last 200ms.

- Task 3 (readout 3)



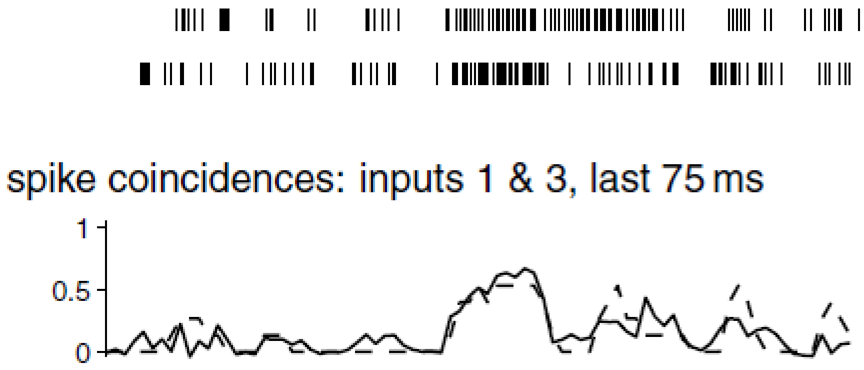
Pattern detection: output a high value if a specific spatiotemporal spike pattern appears.

- Task 4 (readout 4)



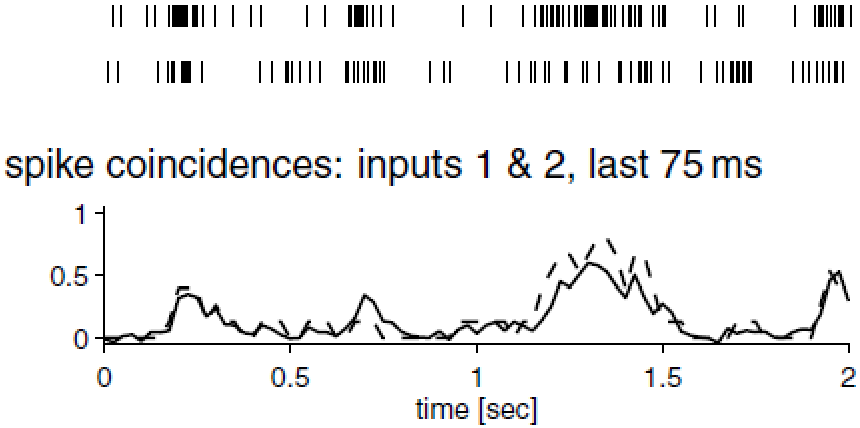
Represent a switch in spatial distribution of rates: output a high value if a specific input pattern occurs where the rate of input spike trains 1 and 2 goes up and simultaneously the rate of input spike trains 3 and 4 goes down.

- Task 5 (readout 5)



Represent the firing correlation: at time t, output the number of spike coincidences during the last 75ms for inputs 1 and 3.

- Task 6 (readout 6)

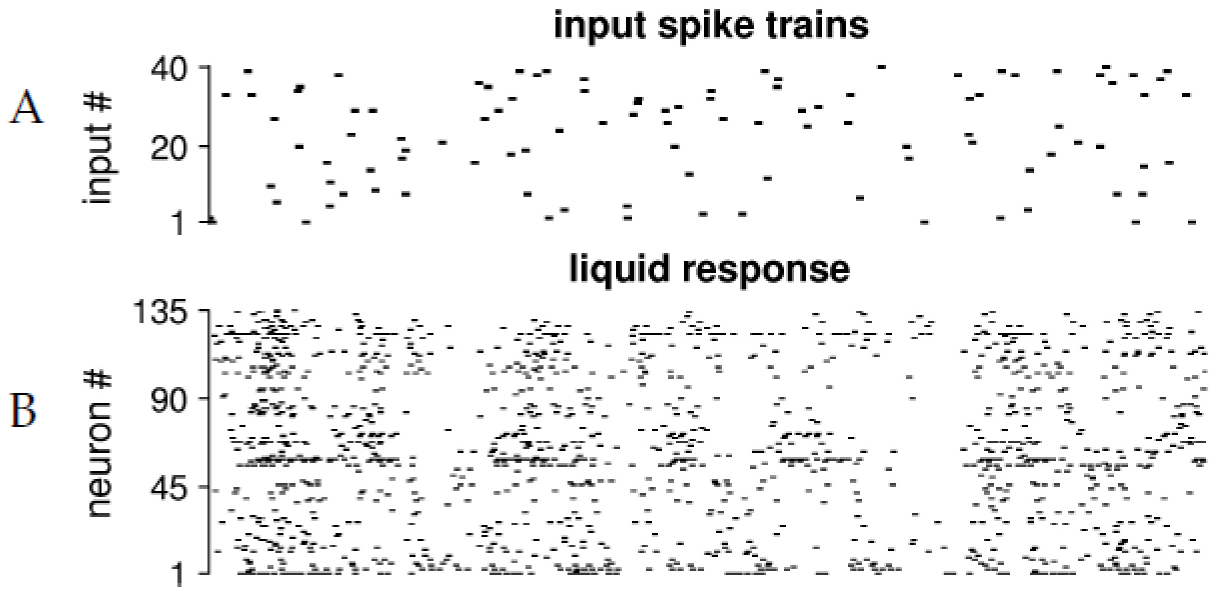


Represent the firing correlation: at time t, output the number of spike coincidences during the last 75ms for inputs 1 and 2.

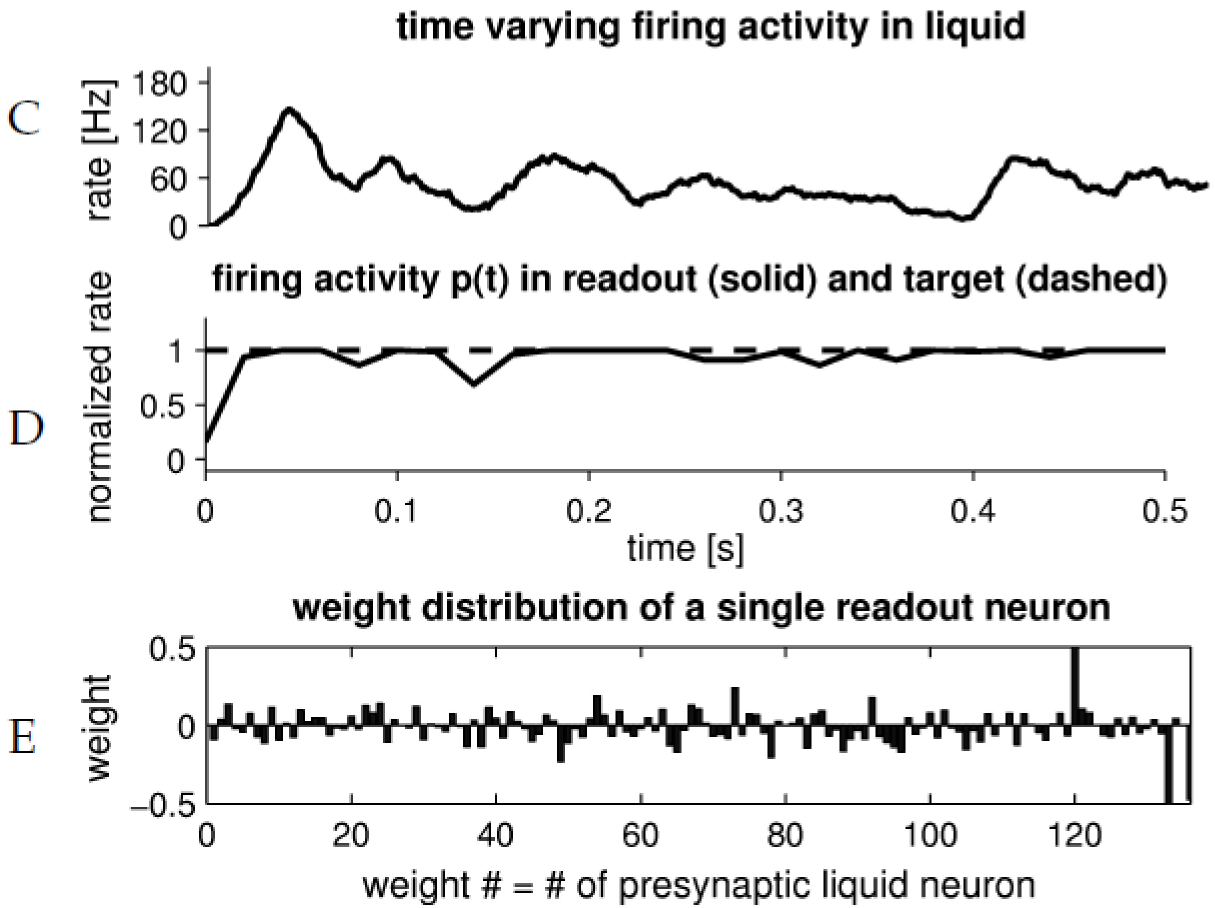
Readout-assigned equivalent states of a dynamical system

- How does the readout transfer time-varying liquid states to stable response?
- The readout neurons have learned to define a notion of equivalence for dynamic states of the liquid.
- Reexamine the classification test.

- Variety of liquid states



- Comparison between liquid and readout



Discussion and conclusion

- The LSM differs from other computational models in that it does not require task-specific circuit or program.
- Only the readouts have to be adapted for specific computational tasks, so the recurrent circuit can support parallel computations.
- Provides possible explanations for the computational role of both connectivity structure and connection length.
- Reveals that readout elements can establish their own equivalence relationships on high-dimensional transient states.