Introduction to Neural and Cognitive Modelling
Monsoon ’22, IIIT-H

Find implementation here.

Abhinav S Menon (2020114001)

Real-Time Computing Without Stable States

A New Framework for Neural Computation Based on Perturbations

Maass, Wolfgang, Thomas Natschläger, and Henry Markram. "Real-time computing without stable states: A new framework for neural computation based on perturbations." *Neural computation* 14.11 (2002): 2531-2560.

Abstract

- New paradigm for real-time computing
  - Alternative to TMs, attractor NNs
  - Based on principles of high-dimensional dynamic systems
- Liquid State Machines
  - Universal computational power
  - Separation, approximation

Overview

- Part I – Motivation and Foundation
    - Introduction
    - Computing without Attractors
- Part II – Theory
    - **Liquid State Machines**
    - UCP of LSMs
- Part III – Practical Considerations
    - Neural Microcircuits for LSMs
    - **Exploring Computational Power**
    - Adding Computational Power
- Part IV – Details
    - **Parallel Computing in Real-Time**
    - Readout-Assigned Equivalence
- Part V
    - Discussion

Part I
Motivation and Foundation

- Introduction
- Computing without Attractors

Introduction

- Complex neural microcircuits carry out computation in the brain

- Difficult to model
  - High-dimensional
  - Real-time input

- Existing attempts
  - Adaptive mechanisms to control high dimensionality
  - Simulating TMs
  - Attractor NNs

Introduction (contd.)

- Adaptive mechanisms to control high dimensionality
  - Do not apply to spiking neurons
- Simulating TMs
  - Requires central clock
  - Constructed, not evolved
  - Break down under noise
- Attractor NNs
  - Memory inefficient
  - Less suitable for real-time
- NOTA allow multiple parallel computations on same input

Introduction (contd.)

- Readout neurons extract info from neural microcircuits
  - Learns to identify salient features
  - Transform transient states into stable readouts

- Multiple readout modules can run in parallel

- Does not require convergence to stable states
  - Perturbations capture past info

Computing Without Attractors

- Analogy to liquid
    - Transient perturbations in excitable medium
    - As attractor NN, useless

- Perturbations contain present and past information
    - Sensitive to different inputs
    - Nonchaotic

Computing Without Attractors (contd.)

- Neural microcircuits are ideal liquids for computation
  - Diversity of elements
  - Large variety of mechanisms, time constants

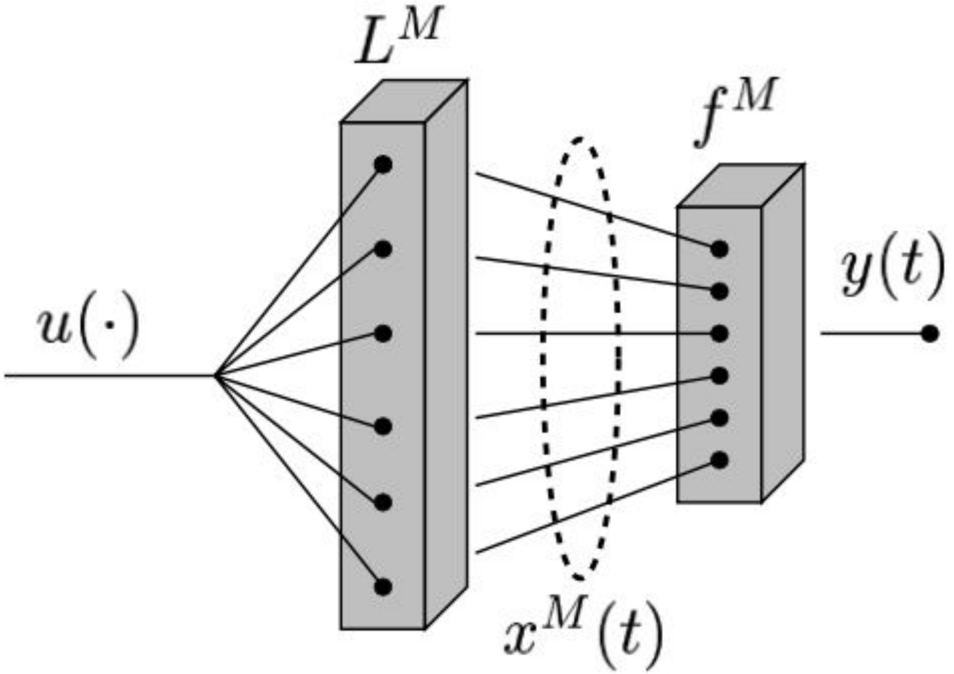
- Rigorous model: Liquid State Machine
  - Separation property (liquid)
    - Between trajectories of states caused by different input streams
  - Approximation property (readout)
    - Distinguish and transform internal states to target outputs

Part II
Theory

- Liquid State Machines
- Universal Computational Power of LSMs

Liquid State Machines

- Based on rigorous mathematical framework like TMs
  - Universal computational power
  - Real-time computing with fading memory on analog functions in continuous time
- Map input functions $u(t)$ to output functions $y(t)$
  - Liquid
    - All inputs $u(s \le t)$ generate a liquid state $x_u(t)$
    - $x_u(t)$ contains all info accessible to readout
  - Readout
    - Takes $x_u(t)$ to $y(t)$
    - Memoryless
- Separation property is a proxy for stable memory



Liquid State Machine Framework

Universal Computational Power of LSMs

- Universal power of computation with fading memory on functions of time
  - Any filter $F : u(t) \rightarrow y(t)$ can be approximated
    - Fading memory: MSBs of $(Fu)(0)$ depend only on MSBs of $(Fu)(T \leq t \leq 0)$
    - Time-invariant: $F(u(t + t_0)) = (Fu)(t + t_0)$
  - Arguably, all that is necessary for living organism
- Can be proved that LSMs have UCP
  - $L : u(t) \rightarrow x(t)$ satisfies SP
  - $f : x(t) \rightarrow y(t)$ satisfies AP
- No requirement for filters, unlike other models

Part II Practical Considerations

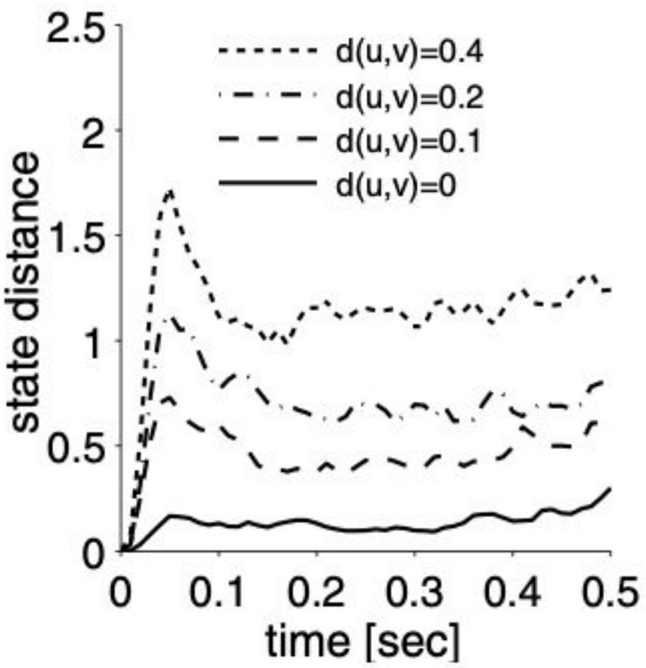
- Neural Microcircuits for LSMs
- Exploring Computational Power
- Adding Computational Power

Neural Microcircuits as Implementations of LSMs

- (Simulation of) Generic recurrent circuit of IF neurons employed as liquid filter
- Evaluated performance on tasks to test UCP

- Input to circuit: Spike trains
- Liquid state: Vector of membrane voltages of liquid neurons
- Readout map: Population of readout neurons w/o lateral/recurrent connections
    - Analog output = firing activity p(t) in 20ms bin
    - Trained by adjusting liquid-readout synapse weights

- Evaluated SP on spike train inputs
    - Distance between state trajectories well above noise level
    - Proportional to distance between input trains

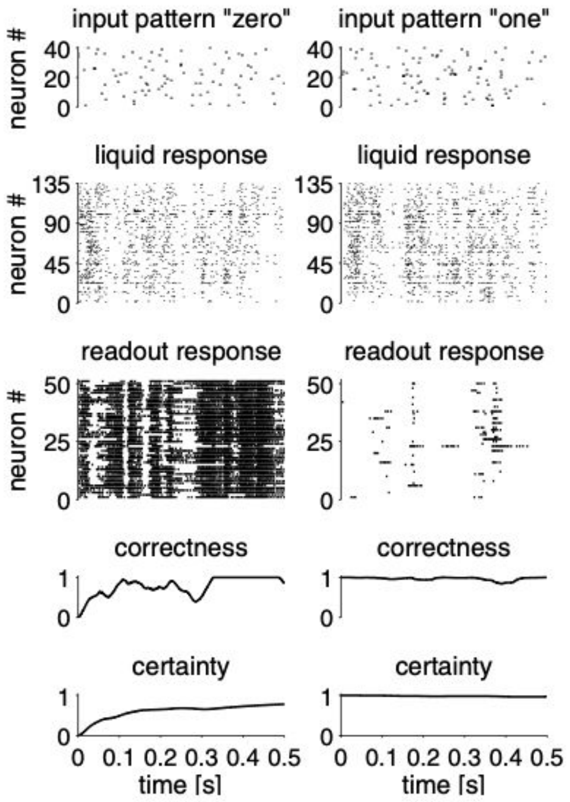


Average Distance of Liquid States for Two Different Input Spike Trains

Exploring the Computational Power of Models for Neural Microcircuits

- First test: classification task
  - Spoken words (noise-corrupted spike train)
  - Previously solved by a network that allows only 1 spike/channel

- Current method:
  - More general (multiple spikes per channel)
  - Has continuous output (becomes correct as soon as enough information is absorbed)
  - Invariant to large class of noise
- Correctness
  - $c(t) = 1 - |p(t) - y(t)|$
- Certainty
  - $v(t) = \operatorname{avg}_{s \leq t} c(s)$

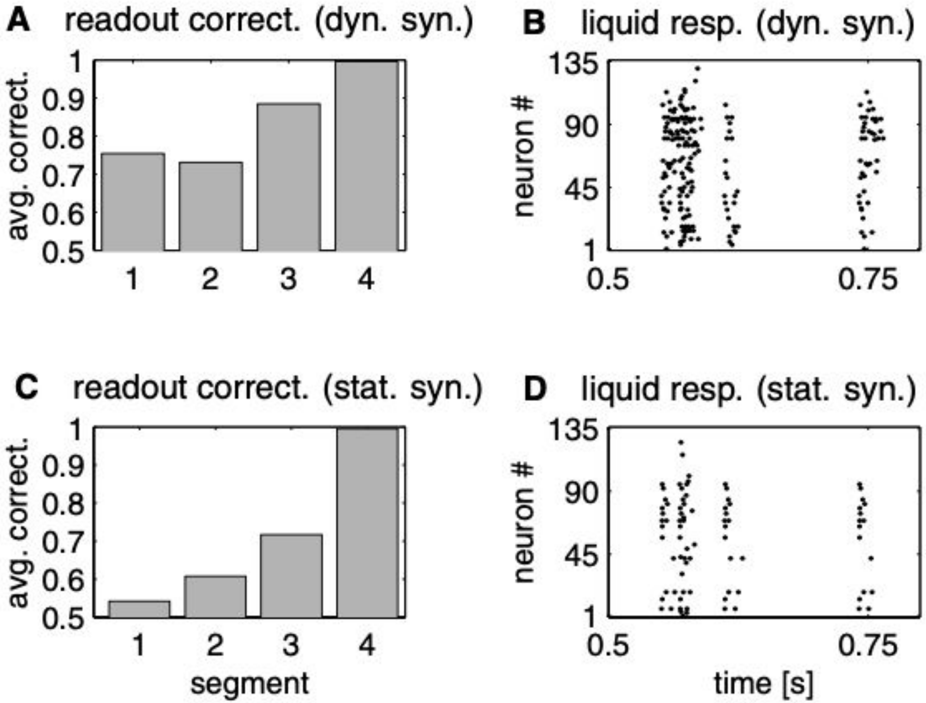


Response of LSM (trained to identify "zero") on Noisy Inputs

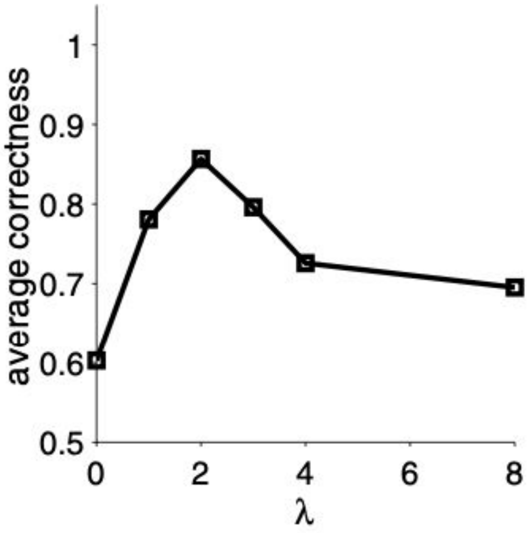
Exploring the Computational Power of Models for Neural Microcircuits (contd.)

- Second task: All information contained in interspike intervals
    - Predefined eight templates of 250ms spike trains
    - Input
        - 4×250 ms spike trains
        - Each segment = template + jitter
    - Four readout modules at $t = 1000$ms to classify each preceding segment
- Circuit is able to remember
    - Even though $\tau = 30$ms
    - When synapses are static, significantly less accurate

- Statistical distribution of connection lengths
    - $\lambda = 0$: no recurrent connections
    - $\lambda >> 0$: homogenised, chaotic
- Main bottleneck: SP
    - Improving AP did not help



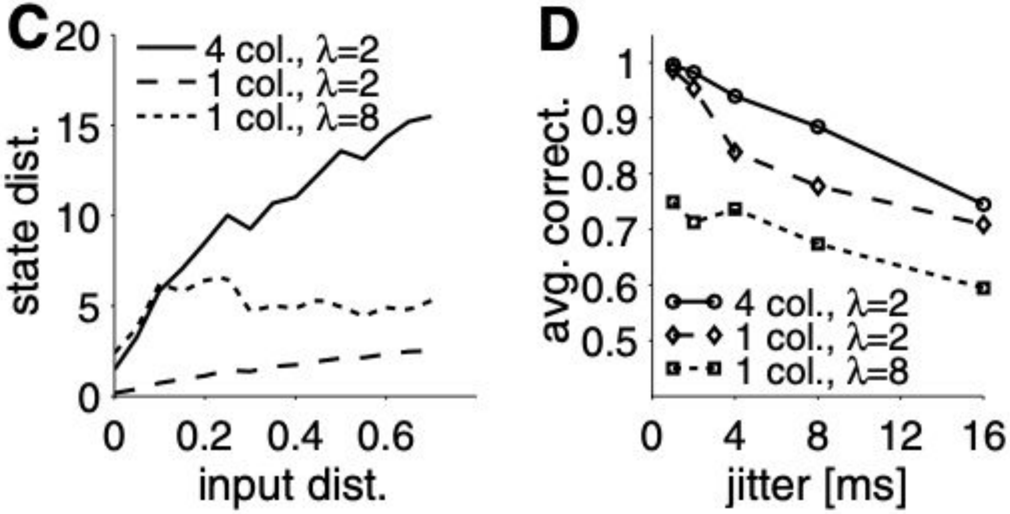
Accuracy of Segment Template Prediction



Effect of Length of Connections on f_3 Classification

Adding Computational Power

- Current artificial mechanisms do not reflect increase in power on addition of circuitry
  - Without need to retrain existing circuitry
- Explored increase in SP on
  - Adding more columns
  - Adding more connections to column
- Adding more columns increases SP in a desirable manner
- Adding more connections increases SP quasi-chaotically
  - Increases sensitivity to task as well as to noise



Effect on SP of More Connections vs. More Columns

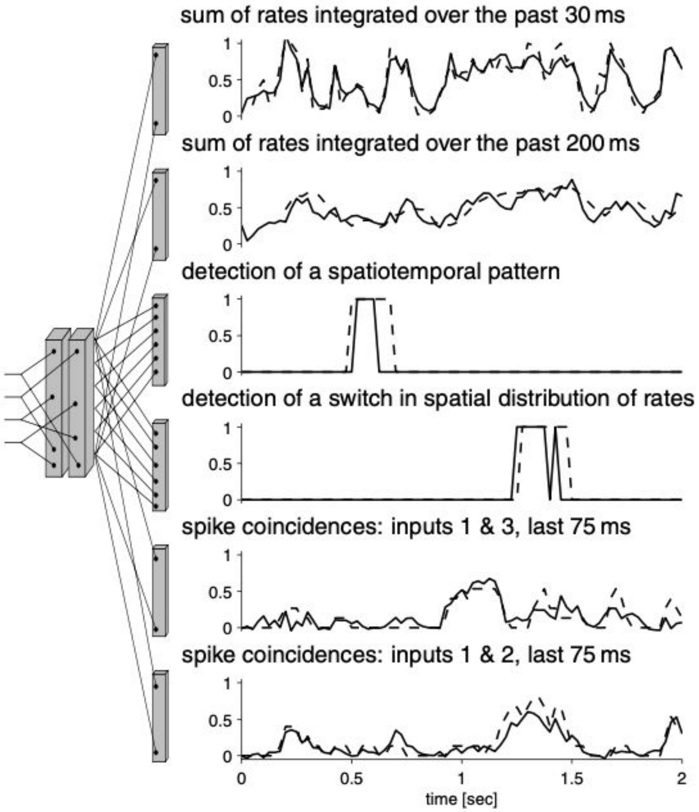
Part IV
Details

- Parallel Computing in Real-Time
- Readout-Assigned Equivalence

Parallel Computing in Real-Time on Novel Inputs

- Liquid supports parallel computing
  - Task-invariant

- Tested on six real-time tasks
  - Two-column liquid
  - Six readout modules
  - Diverse and rapidly changing y(t)
  - Novel input

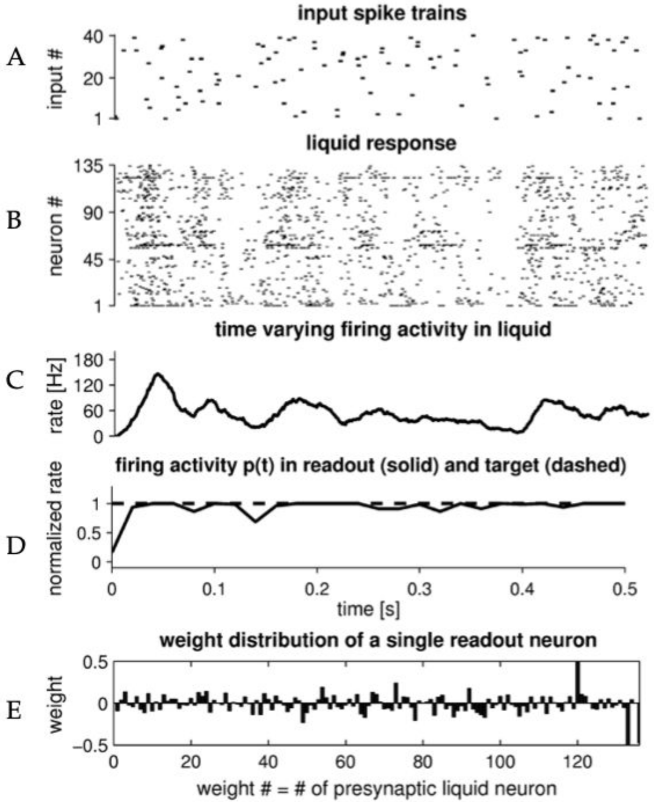


Accuracy of Multiple Simultaneous Real-Time Tasks

Readout-Assigned Equivalent States of a Dynamical System

- Readout must be able to generate stable output
  - Liquid state may never repeat
  - Dynamics of readout modules can significantly vary from liquid state

- Readout neurons define equivalence for states of liquid
  - Inevitable since states of HDDS are collapsed into one dimension
  - Meaningful notion



Invariance of Readout under Non-Constant Liquid Input

Part V

- Discussion

Discussion

- Introduced LSM
  - Does not require construction of circuit/program
  - Relies on principles of HDDSs and learning theory
  - Computing w/o stable states or attractors

- Demonstrated UCP of IF neurons
  - First stable & generally applicable method
  - Explanation for distribution of connection lengths
  - Role of dynamic synapses
  - Enhanced rather than hampered by diversity
  - Biologically realistic model of temporal aspects

Discussion (contd.)

- New approach for computational modelling
  - Need not encode all inputs through FF pathways
  - Global information about preceding inputs stored in HDDS trajectories
  - Basic units are columns, not neurons
- Impact on other fields
  - Real-time processing of input streams: robotics
    - Nonlinear projection of inputs into HD space
    - Linear readout
    - Easy to train in an on-line manner
  - Neuromorphic engineering
  - Analog VLSI