Lecture 19:
Liquid State Machines (LSMs)

Objectives

1. Understand the architecture of Liquid-state Machine
2. Understand the baseline working principle of Liquid-State Machines
3. Understand the properties of a good Liquid-state Machine

Introduction: Liquid State Machines (LSMs)

- Like ESNs, an LSM is a reservoir computing model for processing information in a recurrent neural network architecture.

Introduction: Liquid State Machines (LSMs)

- Like ESNs, an LSM is a reservoir computing model for processing information in a recurrent neural network architecture.
- In biological circuits, individual neurons lack the capacity to process intricate input information. Instead, neural microcircuits serve as a computational foundation. The concept of microcircuit computing, known as the Liquid State Machine (LSM), was introduced by Wolfgang Maass in 2002.

Introduction: Liquid State Machines (LSMs)

- Like ESNs, an LSM is a reservoir computing model for processing information in a recurrent neural network architecture.
- In biological circuits, individual neurons lack the capacity to process intricate input information. Instead, neural microcircuits serve as a computational foundation. The concept of microcircuit computing, known as the Liquid State Machine (LSM), was introduced by Wolfgang Maass in 2002.
- Essentially, the brain cortex or a specific area of it is treated as a "liquid." These cortical microcircuits exhibit remarkable computational abilities when confronted with perturbations. These microcircuits possess diverse elements, such as neurons and synapses, along with a variety of mechanisms and time constants that define their interactions, including recurrent connections.

Introduction: Liquid State Machines (LSMs)

- Like ESNs, an LSM is a reservoir computing model for processing information in a recurrent neural network architecture.
- In biological circuits, individual neurons lack the capacity to process intricate input information. Instead, neural microcircuits serve as a computational foundation. The concept of microcircuit computing, known as the Liquid State Machine (LSM), was introduced by Wolfgang Maass in 2002.
- Essentially, the brain cortex or a specific area of it is treated as a "liquid." These cortical microcircuits exhibit remarkable computational abilities when confronted with perturbations. These microcircuits possess diverse elements, such as neurons and synapses, along with a variety of mechanisms and time constants that define their interactions, including recurrent connections.
- To illustrate the LSM approach, consider a sequence of temporary disruptions applied to an excitable medium, similar to sounds, objects dropped into a liquid or wind gusts. By treating the liquid as an attractor neural network, its resting state becomes the sole attractor. However, perturbed states of the liquid represent both current and past inputs, carrying valuable information for analyzing the surrounding environment.

Introduction: Liquid State Machines (LSMs)

- In contrast to other models, the Liquid State Machine (LSM) is specifically designed for real-time computations on continuous streams of data. These data streams consist of spike trains, which are sequences of action potentials from neurons that serve as inputs to a cortical microcircuit, i.e., a network of interconnected neurons within the cerebral cortex, the brain's outer layer responsible for many higher-order functions such as perception, cognition, and motor control. These microcircuits are the fundamental units of cortical processing, playing a crucial role in how the brain interprets and responds to information.

Introduction: Liquid State Machines (LSMs)

- In contrast to other models, the Liquid State Machine (LSM) is specifically designed for real-time computations on continuous streams of data. These data streams consist of spike trains, which are sequences of action potentials from neurons that serve as inputs to a cortical microcircuit, i.e., a network of interconnected neurons within the cerebral cortex, the brain's outer layer responsible for many higher-order functions such as perception, cognition, and motor control. These microcircuits are the fundamental units of cortical processing, playing a crucial role in how the brain interprets and responds to information.
- Cortical microcircuits are also organized into vertical structures called cortical columns or minicolumns. Each column is thought to represent a basic functional unit of the cortex, processing a specific type of information.

Introduction: Liquid State Machines (LSMs)

- In contrast to other models, the Liquid State Machine (LSM) is specifically designed for real-time computations on continuous streams of data. These data streams consist of spike trains, which are sequences of action potentials from neurons that serve as inputs to a cortical microcircuit, i.e., a network of interconnected neurons within the cerebral cortex, the brain's outer layer responsible for many higher-order functions such as perception, cognition, and motor control. These microcircuits are the fundamental units of cortical processing, playing a crucial role in how the brain interprets and responds to information.
- Cortical microcircuits are also organized into vertical structures called cortical columns or minicolumns. Each column is thought to represent a basic functional unit of the cortex, processing a specific type of information.
- In the LSM framework, both the inputs and outputs are treated as continuous-time data streams. Mathematically, these inputs and outputs are represented as functions $u(t)$ and $y(t)$ that operate in continuous time.

Introduction: Liquid State Machines (LSMs)

- In contrast to other models, the Liquid State Machine (LSM) is specifically designed for real-time computations on continuous streams of data. These data streams consist of spike trains, which are sequences of action potentials from neurons that serve as inputs to a cortical microcircuit, i.e., a network of interconnected neurons within the cerebral cortex, the brain's outer layer responsible for many higher-order functions such as perception, cognition, and motor control. These microcircuits are the fundamental units of cortical processing, playing a crucial role in how the brain interprets and responds to information.
- Cortical microcircuits are also organized into vertical structures called cortical columns or minicolumns. Each column is thought to represent a basic functional unit of the cortex, processing a specific type of information.
- In the LSM framework, both the inputs and outputs are treated as continuous-time data streams. Mathematically, these inputs and outputs are represented as functions $u(t)$ and $y(t)$ that operate in continuous time.
- These functions are typically multidimensional because they capture spike trains from multiple external neurons providing inputs to the circuit and involve various readouts that extract output spike trains. Since an LSM maps input streams $u(\cdot)$ to output streams $y(\cdot)$, it is often described as implementing a functional or operator, akin to a filter. From a mathematical perspective, it represents a function that operates on objects of a higher type than numbers or bits.

Introduction: Liquid State Machines (LSMs)

- In contrast to other models, the Liquid State Machine (LSM) is specifically designed for real-time computations on continuous streams of data. These data streams consist of spike trains, which are sequences of action potentials from neurons that serve as inputs to a cortical microcircuit, i.e., a network of interconnected neurons within the cerebral cortex, the brain's outer layer responsible for many higher-order functions such as perception, cognition, and motor control. These microcircuits are the fundamental units of cortical processing, playing a crucial role in how the brain interprets and responds to information.
- Cortical microcircuits are also organized into vertical structures called cortical columns or minicolumns. Each column is thought to represent a basic functional unit of the cortex, processing a specific type of information.
- In the LSM framework, both the inputs and outputs are treated as continuous-time data streams. Mathematically, these inputs and outputs are represented as functions $u(t)$ and $y(t)$ that operate in continuous time.
- These functions are typically multidimensional because they capture spike trains from multiple external neurons providing inputs to the circuit and involve various readouts that extract output spike trains. Since an LSM maps input streams $u(\cdot)$ to output streams $y(\cdot)$, it is often described as implementing a functional or operator, akin to a filter. From a mathematical perspective, it represents a function that operates on objects of a higher type than numbers or bits.
- One notable characteristic of this higher-type computational processing is that the output stream's target value, $y(t)$, at a given time t, may depend on the input stream's values, $u(s)$, at many (potentially even infinitely many) preceding time points s.

Introduction: Liquid State Machines (LSMs)

- In biological information processing models, a typical readout consists of a single neuron, referred to as a projection neuron in neuroscience. This neuron extracts information from a local microcircuit and transmits it to other microcircuits within the brain, whether in the same region or elsewhere.

Introduction: Liquid State Machines (LSMs)

- In biological information processing models, a typical readout consists of a single neuron, referred to as a projection neuron in neuroscience. This neuron extracts information from a local microcircuit and transmits it to other microcircuits within the brain, whether in the same region or elsewhere.
- Various methods can be used to model this readout neuron, such as a linear gate, a perceptron with a threshold, a sigmoidal gate, or a spiking neuron. The main purpose of the LSM (referred to as the "Liquid") is to act as a pre-processor for this readout neuron, expanding the range of possible functions that it can learn from input streams $u(t)$.

Introduction: Liquid State Machines (LSMs)

- In biological information processing models, a typical readout consists of a single neuron, referred to as a projection neuron in neuroscience. This neuron extracts information from a local microcircuit and transmits it to other microcircuits within the brain, whether in the same region or elsewhere.
- Various methods can be used to model this readout neuron, such as a linear gate, a perceptron with a threshold, a sigmoidal gate, or a spiking neuron. The main purpose of the LSM (referred to as the "Liquid") is to act as a pre-processor for this readout neuron, expanding the range of possible functions that it can learn from input streams $u(t)$.
- This division of computational processing into the Liquid and the readout is highly efficient because the same Liquid can serve multiple readout neurons, each learning to extract a distinct "summary" of information from the same Liquid.
- The need to extract different information summaries from a cortical microcircuit arises due to different computational objectives, such as determining the movement direction of objects or identifying object identities when $u(t)$ represents visual inputs.

Introduction: Liquid State Machines (LSMs)

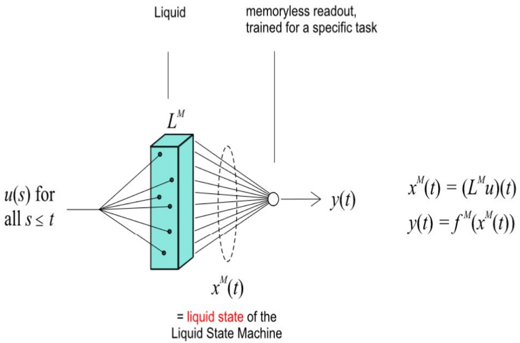
- In biological information processing models, a typical readout consists of a single neuron, referred to as a projection neuron in neuroscience. This neuron extracts information from a local microcircuit and transmits it to other microcircuits within the brain, whether in the same region or elsewhere.
- Various methods can be used to model this readout neuron, such as a linear gate, a perceptron with a threshold, a sigmoidal gate, or a spiking neuron. The main purpose of the LSM (referred to as the "Liquid") is to act as a pre-processor for this readout neuron, expanding the range of possible functions that it can learn from input streams $u(t)$.
- This division of computational processing into the Liquid and the readout is highly efficient because the same Liquid can serve multiple readout neurons, each learning to extract a distinct "summary" of information from the same Liquid.
- The need to extract different information summaries from a cortical microcircuit arises due to different computational objectives, such as determining the movement direction of objects or identifying object identities when $u(t)$ represents visual inputs.
- Neurophysiological data indicates that spike trains from different projection neurons in the same column are generally weakly correlated with natural stimuli. Therefore, the LSM serves as a model for multiplexing diverse computations on a shared input stream $u(t)$.

Introduction: Liquid State Machines (LSMs)

- The model of a LSM is based on a strict mathematical framework that guarantees, under ideal conditions, universal computational power.

Introduction: Liquid State Machines (LSMs)

- The model of a LSM is based on a strict mathematical framework that guarantees, under ideal conditions, universal computational power.
- The figure below shows only one input and output channel for simplicity. A liquid-state machine consists of three main components. The first component is a layer of input neurons where an external stimulus is introduced. The signal from this component is transmitted to the selected neurons of the second component, called the liquid column denoted that L^M, where the proper neural computation takes place.



Introduction: Liquid State Machines (LSMs)

- In mathematical terms, the liquid state is simply the current output of some operator or filter L^M that maps input functions $u(\cdot)$ onto functions $x^M(t)$:

$$x^M(t) = \left(L^M u\right)(t) \tag{1}$$

Introduction: Liquid State Machines (LSMs)

- In mathematical terms, the liquid state is simply the current output of some operator or filter L^M that maps input functions $u(\cdot)$ onto functions $x^M(t)$:

$$x^M(t) = \left(L^M u\right)(t) \tag{1}$$

- The liquid component of the system receives a continuous input stream, denoted as $u(s)$. As time progresses, specifically at a later time $t > s$, the current internal state $x^M(t)$ of the liquid holds a significant amount of information regarding the recent inputs $u(s)$. This current liquid state $x^M(t)$ is then transmitted to the final component of the liquid-state machine. This component is referred to as a memory-less readout map, denoted as f^M, meaning that it has no access to any states prior to time t.

Introduction: Liquid State Machines (LSMs)

- In mathematical terms, the liquid state is simply the current output of some operator or filter L^M that maps input functions $u(\cdot)$ onto functions $x^M(t)$:

$$x^M(t) = \left(L^M u\right)(t) \tag{1}$$

- The liquid component of the system receives a continuous input stream, denoted as $u(s)$. As time progresses, specifically at a later time $t > s$, the current internal state $x^M(t)$ of the liquid holds a significant amount of information regarding the recent inputs $u(s)$. This current liquid state $x^M(t)$ is then transmitted to the final component of the liquid-state machine. This component is referred to as a memory-less readout map, denoted as f^M, meaning that it has no access to any states prior to time t.
- The readout map performs an analysis and interpretation of the liquid's calculations. Ultimately, the input signal is transformed by the readout map, resulting in the system's output $y(t)$ given by:

$$y(t) = f^M\left(x^M(t)\right). \tag{2}$$

Introduction: Liquid State Machines (LSMs)

- In mathematical terms, the liquid state is simply the current output of some operator or filter L^M that maps input functions $u(\cdot)$ onto functions $x^M(t)$:

$$x^M(t) = \left(L^M u\right)(t) \tag{1}$$

- The liquid component of the system receives a continuous input stream, denoted as $u(s)$. As time progresses, specifically at a later time $t > s$, the current internal state $x^M(t)$ of the liquid holds a significant amount of information regarding the recent inputs $u(s)$. This current liquid state $x^M(t)$ is then transmitted to the final component of the liquid-state machine. This component is referred to as a memory-less readout map, denoted as f^M, meaning that it has no access to any states prior to time t.
- The readout map performs an analysis and interpretation of the liquid's calculations. Ultimately, the input signal is transformed by the readout map, resulting in the system's output $y(t)$ given by:

$$y(t) = f^M\left(x^M(t)\right). \tag{2}$$

- The structure of the readout map, f^M, is not explicitly provided, allowing for the utilization of various statistical analysis or pattern recognition methods that are available (e.g. linear regression, stochastic gradient descent, etc).

Properties of a good Liquid State Machines (LSMs)

- The fundamental characteristic of an LSM lies in its capacity to generate distinct responses to different input patterns. This ability, known as "the separation property" of the liquid, plays a pivotal role.
- On the other hand, the "approximation property" refers to the capability of the readout maps to differentiate these responses, generalize them, and establish a connection with the given output.

Properties of a good Liquid State Machines (LSMs)

- The fundamental characteristic of an LSM lies in its capacity to generate distinct responses to different input patterns. This ability, known as "the separation property" of the liquid, plays a pivotal role.
- On the other hand, the "approximation property" refers to the capability of the readout maps to differentiate these responses, generalize them, and establish a connection with the given output.
- The approximation property primarily relies on the adaptive nature of the readout map, while the separation property is directly influenced by the complexity of the liquid itself.

Properties of a good Liquid State Machines (LSMs)

- The fundamental characteristic of an LSM lies in its capacity to generate distinct responses to different input patterns. This ability, known as "the separation property" of the liquid, plays a pivotal role.
- On the other hand, the "approximation property" refers to the capability of the readout maps to differentiate these responses, generalize them, and establish a connection with the given output.
- The approximation property primarily relies on the adaptive nature of the readout map, while the separation property is directly influenced by the complexity of the liquid itself.
- Differentiating between two liquid states or comparing the responses of a Liquid State Machine to two stimuli presents a challenging task. There is no definitive measure of the distance between liquid states or input patterns. Due to the highly dynamic and non-linear nature of liquid states, it becomes evident that a straightforward "one-to-one" mapping of two states is not feasible. Even a slight variation in the input pattern can result in a significant disparity in the liquid's response.

Properties of a good Liquid State Machines (LSMs)

- However, certain basic comparisons, such as the Euclidean norm, can offer insights into the distance between liquid trajectories for two input patterns. The Euclidean distance, formally calculated using the following formula, determines the distance between liquid states for all recorded time steps:

$$E_{I_1 I_2}(t) = \sqrt{\sum_{i=1}^{n} \left(y_i^{I_1}(t) - y_i^{I_2}(t) \right)^2} \tag{3}$$

where n stands for the number of neurons in the liquid, $y_i^{I_1}(t)$ and $y_i^{I_2}(t)$ are the ith neuron activity (the neuron activity is usually the spiking times) measured in time t after presenting I_1 and I_2 inputs, respectively.

Properties of a good Liquid State Machines (LSMs)

- However, certain basic comparisons, such as the Euclidean norm, can offer insights into the distance between liquid trajectories for two input patterns. The Euclidean distance, formally calculated using the following formula, determines the distance between liquid states for all recorded time steps:

$$E_{I_1 I_2}(t) = \sqrt{\sum_{i=1}^{n} \left(y_i^{I_1}(t) - y_i^{I_2}(t) \right)^2} \tag{3}$$

where n stands for the number of neurons in the liquid, $y_i^{I_1}(t)$ and $y_i^{I_2}(t)$ are the ith neuron activity (the neuron activity is usually the spiking times) measured in time t after presenting I_1 and I_2 inputs, respectively.

- In order to investigate the information flow in LSMs a global entropy (denoted as S) concept based on the classical definition of Shannon's informational entropy, can be used. Simulating liquid for a duration of T and obtaining N spikes on the readout, the probability of having n_i spikes occurring during a period t_i of e.g., $t_i = T/100$ is:

$$p_i = \frac{n_i}{N} \tag{4}$$

Properties of a good Liquid State Machines (LSMs)

- However, certain basic comparisons, such as the Euclidean norm, can offer insights into the distance between liquid trajectories for two input patterns. The Euclidean distance, formally calculated using the following formula, determines the distance between liquid states for all recorded time steps:

$$E_{I_1 I_2}(t) = \sqrt{\sum_{i=1}^{n} \left(y_i^{I_1}(t) - y_i^{I_2}(t) \right)^2} \tag{3}$$

  where n stands for the number of neurons in the liquid, $y_i^{I_1}(t)$ and $y_i^{I_2}(t)$ are the ith neuron activity (the neuron activity is usually the spiking times) measured in time t after presenting I_1 and I_2 inputs, respectively.

- In order to investigate the information flow in LSMs a global entropy (denoted as S) concept based on the classical definition of Shannon's informational entropy, can be used. Simulating liquid for a duration of T and obtaining N spikes on the readout, the probability of having n_i spikes occurring during a period t_i of e.g., $t_i = T/100$ is:

$$p_i = \frac{n_i}{N} \tag{4}$$

- Such a probability can be interpreted as a chance of giving the whole information in t_i part of time T.

Properties of a good Liquid State Machines (LSMs)

- Using probability p_i, partial entropy can be defined as:

$$S_0^i = -p_i \cdot ln(p_i) \tag{5}$$

and global entropy as:

$$S = \sum_{i=1} S_0^i = -\sum_{i=1} p_i \cdot ln(p_i) \tag{6}$$

Properties of a good Liquid State Machines (LSMs)

- Using probability p_i, partial entropy can be defined as:

$$S_0^i = -p_i \cdot ln(p_i) \tag{5}$$

and global entropy as:

$$S = \sum_{i=1} S_0^i = -\sum_{i=1} p_i \cdot ln(p_i) \tag{6}$$

- In information theory. Shannon's entropy is a measure that quantifies the amount of uncertainty or unpredictability in a set of data or information. High entropy indicates a high degree of randomness and complexity in the system's states. Systems with high entropy can explore a larger state space, which may improve their ability to learn complex functions.

Properties of a good Liquid State Machines (LSMs)

- Using probability p_i, partial entropy can be defined as:

$$S_0^i = -p_i \cdot ln(p_i) \tag{5}$$

and global entropy as:

$$S = \sum_{i=1} S_0^i = -\sum_{i=1} p_i \cdot ln(p_i) \tag{6}$$

- In information theory. Shannon's entropy is a measure that quantifies the amount of uncertainty or unpredictability in a set of data or information. High entropy indicates a high degree of randomness and complexity in the system's states. Systems with high entropy can explore a larger state space, which may improve their ability to learn complex functions.
- In general, the measure of global entropy helps in evaluating the capability of the system to represent the information contained in the data properly, and its high index is a cue that the system is able to capture the useful information available in input.

Properties of a good Liquid State Machines (LSMs)

- Using probability p_i, partial entropy can be defined as:

$$S_0^i = -p_i \cdot ln(p_i) \tag{5}$$

and global entropy as:

$$S = \sum_{i=1} S_0^i = -\sum_{i=1} p_i \cdot ln(p_i) \tag{6}$$

- In information theory. Shannon's entropy is a measure that quantifies the amount of uncertainty or unpredictability in a set of data or information. High entropy indicates a high degree of randomness and complexity in the system's states. Systems with high entropy can explore a larger state space, which may improve their ability to learn complex functions.
- In general, the measure of global entropy helps in evaluating the capability of the system to represent the information contained in the data properly, and its high index is a cue that the system is able to capture the useful information available in input.
- A neural network model should have a large Euclidean distance $E_{I_1 I_2}(t)$ (good separability property) and a large global entropy S (to capture the useful information from the input) for it to use as a liquid. Research has shown that leaky integrate-and-fire, Morris-Lecar, resonate-and-fire, and Hindmarsh-Rose neuron models are good candidates for LSMs.

Architecture of a Liquid State Machine

- As already mentioned, an LSM consists of three distinct parts: an input layer, a liquid, and an output layer.

Architecture of a Liquid State Machine

- As already mentioned, an LSM consists of three distinct parts: an input layer, a liquid, and an output layer.
- The input layer consists of spike generator neurons having the same dimensionality as the input.

Architecture of a Liquid State Machine

- As already mentioned, an LSM consists of three distinct parts: an input layer, a liquid, and an output layer.
- The input layer consists of spike generator neurons having the same dimensionality as the input.
- Input neurons connect randomly in a feedforward fashion to excitatory neurons in the liquid. We use a constant number of synapses n_{inpsyn} between the input neurons and the liquid.

Architecture of a Liquid State Machine

- As already mentioned, an LSM consists of three distinct parts: an input layer, a liquid, and an output layer.
- The input layer consists of spike generator neurons having the same dimensionality as the input.
- Input neurons connect randomly in a feedforward fashion to excitatory neurons in the liquid. We use a constant number of synapses n_{inpsyn} between the input neurons and the liquid.
- The liquid consists of N $(= n_{\text{exc}} + n_{\text{inh}})$ randomly connected neurons (e.g., leaky integrate-and-fire neurons), 80% of which are excitatory and 20% inhibitory. The visual cortex biologically inspires such proportion. The liquid neuron parameters are usually the standard parameter values given to the neuron model under consideration.

Architecture of a Liquid State Machine

- As already mentioned, an LSM consists of three distinct parts: an input layer, a liquid, and an output layer.
- The input layer consists of spike generator neurons having the same dimensionality as the input.
- Input neurons connect randomly in a feedforward fashion to excitatory neurons in the liquid. We use a constant number of synapses n_{inpsyn} between the input neurons and the liquid.
- The liquid consists of $N\ (= n_{\text{exc}} + n_{\text{inh}})$ randomly connected neurons (e.g., leaky integrate-and-fire neurons), 80% of which are excitatory and 20% inhibitory. The visual cortex biologically inspires such proportion. The liquid neuron parameters are usually the standard parameter values given to the neuron model under consideration.
- Decorrelation (to induce suitable separability property) between liquid neurons is introduced by drawing synaptic delays, current bias, and connection strengths from a probability distribution.

Architecture of a Liquid State Machine

- As already mentioned, an LSM consists of three distinct parts: an input layer, a liquid, and an output layer.
- The input layer consists of spike generator neurons having the same dimensionality as the input.
- Input neurons connect randomly in a feedforward fashion to excitatory neurons in the liquid. We use a constant number of synapses n_{inpsyn} between the input neurons and the liquid.
- The liquid consists of $N\ (= n_{\text{exc}} + n_{\text{inh}})$ randomly connected neurons (e.g., leaky integrate-and-fire neurons), 80% of which are excitatory and 20% inhibitory. The visual cortex biologically inspires such proportion. The liquid neuron parameters are usually the standard parameter values given to the neuron model under consideration.
- Decorrelation (to induce suitable separability property) between liquid neurons is introduced by drawing synaptic delays, current bias, and connection strengths from a probability distribution.
- We connect n_{rec} excitatory liquid neurons all-to-all to perfect linear readout neurons in a feedforward fashion.

Architecture of a Liquid State Machine

- Only these connections (i.e., those between the recorded liquid neurons and the output neurons) are trained in a way that we describe later.

Architecture of a Liquid State Machine

- Only these connections (i.e., those between the recorded liquid neurons and the output neurons) are trained in a way that we describe later.
- The readout neurons map the liquid activity to the desired output. Note that $n_{\text{rec}} = n_{\text{exc}}$ (i.e., an all-to-all connection between internal excitatory liquid neurons and readouts neurons) is a common choice. (*However, it should be noted that sometimes mapping only a fraction of the excitatory neurons to the output lower memory consumption and do not impact the performance of the LSM.*)

Architecture of a Liquid State Machine

- Only these connections (i.e., those between the recorded liquid neurons and the output neurons) are trained in a way that we describe later.
- The readout neurons map the liquid activity to the desired output. Note that $n_{\text{rec}} = n_{\text{exc}}$ (i.e., an all-to-all connection between internal excitatory liquid neurons and readouts neurons) is a common choice. (*However, it should be noted that sometimes mapping only a fraction of the excitatory neurons to the output lower memory consumption and do not impact the performance of the LSM.*)
- The connection strengths between input and liquid neurons and within the liquid can be tuned to make the liquid memory-driven or input-driven.

Architecture of a Liquid State Machine

- Only these connections (i.e., those between the recorded liquid neurons and the output neurons) are trained in a way that we describe later.
- The readout neurons map the liquid activity to the desired output. Note that $n_{\text{rec}} = n_{\text{exc}}$ (i.e., an all-to-all connection between internal excitatory liquid neurons and readouts neurons) is a common choice. (*However, it should be noted that sometimes mapping only a fraction of the excitatory neurons to the output lower memory consumption and do not impact the performance of the LSM.*)
- The connection strengths between input and liquid neurons and within the liquid can be tuned to make the liquid memory-driven or input-driven.
- For example, by increasing the connection strengths between input and liquid neurons, the activity regime of the liquid will depend more on the inputs than its memory.

Training of a Liquid State Machine

- The training part of an LSM consists of finding the proper weights of the connections between the liquid and the readout neurons.

Training of a Liquid State Machine

- The training part of an LSM consists of finding the proper weights of the connections between the liquid and the readout neurons.
- Let $w_i \in \mathbb{R}^{n_{\text{rec}}}$ be the weights between a readout neuron i and the set of recorded liquid neurons. Training consists of finding the proper weights for all readout neurons.

Training of a Liquid State Machine

- The training part of an LSM consists of finding the proper weights of the connections between the liquid and the readout neurons.
- Let $w_i \in \mathbb{R}^{n_{\text{rec}}}$ be the weights between a readout neuron i and the set of recorded liquid neurons. Training consists of finding the proper weights for all readout neurons.
- The liquid state $x(t) \in \mathbb{R}^{n_{\text{rec}}}$ is a vector containing the post-synaptic membrane potentials of the n_{rec} recorded liquid neurons at time t.
- We sample the liquid state at discrete timesteps of Δ_{sample} for $t \in [0, t_{\text{train}}]$. They are accumulated into a matrix $X \in \mathbb{R}^{n_{\text{rec}} \times n_{\text{sample}}}$ where $n_{\text{sample}} = t_{\text{train}}/\Delta_{\text{sample}}$

Training of a Liquid State Machine

- The training part of an LSM consists of finding the proper weights of the connections between the liquid and the readout neurons.
- Let $w_i \in \mathbb{R}^{n_{\text{rec}}}$ be the weights between a readout neuron i and the set of recorded liquid neurons. Training consists of finding the proper weights for all readout neurons.
- The liquid state $x(t) \in \mathbb{R}^{n_{\text{rec}}}$ is a vector containing the post-synaptic membrane potentials of the n_{rec} recorded liquid neurons at time t.
- We sample the liquid state at discrete timesteps of Δ_{sample} for $t \in [0, t_{\text{train}}]$. They are accumulated into a matrix $X \in \mathbb{R}^{n_{\text{rec}} \times n_{\text{sample}}}$ where $n_{\text{sample}} = t_{\text{train}}/\Delta_{\text{sample}}$
- The matrix X contains all the liquid states encountered during training. We, therefore, have the linear system for all readout neurons i:

$$y_i = X \cdot w_i \tag{7}$$

  where $y_i \in \mathbb{R}^{n_{\text{sample}}}$ is the sampled target (output) signals of readout neuron i.
- In other words, each readout neuron solves a linear regression (just as with ESN) to map the accumulated liquid states to accumulate target signals for this neuron.

Training of a Liquid State Machine

- The training part of an LSM consists of finding the proper weights of the connections between the liquid and the readout neurons.
- Let $w_i \in \mathbb{R}^{n_{\text{rec}}}$ be the weights between a readout neuron i and the set of recorded liquid neurons. Training consists of finding the proper weights for all readout neurons.
- The liquid state $x(t) \in \mathbb{R}^{n_{\text{rec}}}$ is a vector containing the post-synaptic membrane potentials of the n_{rec} recorded liquid neurons at time t.
- We sample the liquid state at discrete timesteps of Δ_{sample} for $t \in [0, t_{\text{train}}]$. They are accumulated into a matrix $X \in \mathbb{R}^{n_{\text{rec}} \times n_{\text{sample}}}$ where $n_{\text{sample}} = t_{\text{train}}/\Delta_{\text{sample}}$
- The matrix X contains all the liquid states encountered during training. We, therefore, have the linear system for all readout neurons i:

$$y_i = X \cdot w_i \tag{7}$$

  where $y_i \in \mathbb{R}^{n_{\text{sample}}}$ is the sampled target (output) signals of readout neuron i.
- In other words, each readout neuron solves a linear regression (just as with ESN) to map the accumulated liquid states to accumulate target signals for this neuron.
- To reduce overfitting, we use a regularized linear regression that penalizes strong synaptic weights. The parameter λ is selected with respect to a cross-validation dataset.

Training of a Liquid State Machine

- The training part of an LSM consists of finding the proper weights of the connections between the liquid and the readout neurons.
- Let $w_i \in \mathbb{R}^{n_{\text{rec}}}$ be the weights between a readout neuron i and the set of recorded liquid neurons. Training consists of finding the proper weights for all readout neurons.
- The liquid state $x(t) \in \mathbb{R}^{n_{\text{rec}}}$ is a vector containing the post-synaptic membrane potentials of the n_{rec} recorded liquid neurons at time t.
- We sample the liquid state at discrete timesteps of Δ_{sample} for $t \in [0, t_{\text{train}}]$. They are accumulated into a matrix $X \in \mathbb{R}^{n_{\text{rec}} \times n_{\text{sample}}}$ where $n_{\text{sample}} = t_{\text{train}}/\Delta_{\text{sample}}$
- The matrix X contains all the liquid states encountered during training. We, therefore, have the linear system for all readout neurons i:

$$y_i = X \cdot w_i \tag{7}$$

  where $y_i \in \mathbb{R}^{n_{\text{sample}}}$ is the sampled target (output) signals of readout neuron i.
- In other words, each readout neuron solves a linear regression (just as with ESN) to map the accumulated liquid states to accumulate target signals for this neuron.
- To reduce overfitting, we use a regularized linear regression that penalizes strong synaptic weights. The parameter λ is selected with respect to a cross-validation dataset.
- Once the weights $w_i \in \mathbb{R}^{n_{\text{rec}}}$ are sufficiently well trained, the test and prediction phases are similar to the ones described for the ESN.

Evaluation of performance of LSM

- We evaluate the performance of the method with respect to the predictions generated by the liquid. We define a metric to evaluate the predictions. This metric is general and consists of computing the normalized error for all tests or predictions:

$$E(W) = \frac{1}{n_{\text{samples}}^{\text{test}}} \sum_{i=1}^{n_{\text{samples}}^{\text{test}}} \left\| X^{\text{test}} \cdot w_i - y_i \right\|, \tag{8}$$

where X^{test} the accumulated liquid states during the test or prediction period, and $n_{\text{samples}}^{\text{test}}$ the number of samples in the test set. The residual error is the one we implicitly minimize when solving the linear equations for all readout neurons, see Eq. (7) above.