École Publique d'Ingénieurs en 3 ans

Report

OVERSEAS INTERNSHIP

08/26/2024

2nd year internship 2023/2024

Abdelmalek BELGHOMARI
Academic year 2023/2024
Computer Science / ISIA

Tuteur entreprise : **Ryosuke HOSAKA**
Tuteur ENSICAEN : **Thibaud SADÉ**

ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

芝浦工業大學
SHIBAURA INSTITUTE OF TECHNOLOGY

www.ensicaen.fr

TABLE DES MATIERES

ACKNOWLEDGMENT

First of all, I would like to thank Professor Ryosuke Hosaka-san for his pedagogy, patience and kindness towards me. He made me feel at home in this new country and helped me understand the complexities of neuroscience and Reservoir Computing.

Also, I would like to thank all the members (and now friends) of the Neural Information Systems laboratory who welcomed me in Japan and who helped me discover the Japanese culture. I would like to thank particularly Kazuki Sakakibara-senpai who helped me during my research.

I want to express all my gratitude to the Shibaura Institute of Technology Global Learning Commons' offices for the help they gave me during my journey in Japan, either for their guidance and patience regarding administrative procedures or for the activities they planned for our integration into the students' life.

I genuinely thank the laboratory and the Shibaura Institute of Technology that allowed me to participate to the 2024 IEEE World Congress on Computational Intelligence in Yokohama.

Of course, I am grateful to the ENSICAEN teaching team for the quality and rigour of the education they provide. It has been a valuable asset that helped me complete my internship and understand high-level scientific research.

1. Introduction

This internship that I completed in Tokyo, Japan started on May 7th. However, I had to find this internship before the end of my academic year in April. I began looking for an internship in the beginning of the 2023/2024 academic year. Bringing us back to October; when my research for a position as an intern debuted.

My preferences were clear; I wanted to find an internship in a foreign country as mandated by the ENSICAEN, but I wanted to go to Japan since I aspire to work there in the future. The goal of this internship was not only to deepen my knowledge of computer science, but also to improve both my English and Japanese skills. At this time of the year, I was unsure whether I wanted to perform an internship in the domain of AI or Cybersecurity. So, I initially targeted Japanese and global companies based in Japan, but it wasn't successful. Every time I found an offer or had an interview, it was only for six-months long internships.

Consequently, I started contacting several laboratories in some Japanese universities and private research institutions to enhance my chances of having an internship in Japan. Although, having an internship in a laboratory had two advantages; firstly, I would observe how people study and research in a country so different from ours, and secondly, it would definitely clarify whether I want to pursue a PhD after getting my engineer degree. Additionally, I wanted to discover other scientific fields through my expertise of computer science. My years in CPGE had me curious about not only mathematics and physics but also about chemistry, biology and medicine. I aimed to find an internship that would be the crossover between these disciplines. After receiving offers from multiples laboratories, I chose the one that best aligned my interests.

In this report, I will cover all aspects of my internship, the life in a Japanese university, which include the laboratory's purpose, the team, my role within the laboratory and the work I have done. Also, I will highlight the skills I acquired and the passion I developed for neuroscience. Naturally, I will also share my feedback and give an outlook on the year ahead and beyond.

2. My experience in a Japanese University

The Shibaura Institute of technology (SIT) is a Japanese prestigious university established in 1927, having a total of 7,600 students enrolled. SIT have a strong commitment to fostering research and engineering. With a large international community among researchers and professors, cutting-edge laboratories and facilities, SIT is at the forefront of developing solutions to today's challenges.

I came to the SIT with the "overseas intern researcher" title. This allowed me to get access to all the buildings and facilities within the SIT regardless of the campus.

The SIT has two campuses in the area of Tokyo:

- The major campus is the one in the "city" of Toyosu, in the bay of Tokyo. I was spending one to two days a week in the buildings of Toyosu. The University neighbourhood was a plenty of skyscrapers being the offices of the biggest companies in Japan. It felt a little bit impressive to study in those buildings in the middle of the biggest business district in Tokyo.

- The second campus, where my lab was located and where I was spending three to four days a week, is located in Higashiōmiya, in the prefecture of Saitama. The campus is surrounded by a countryside-style city. I had access to the university and the laboratory library, with the possibility to borrow and order any book.

Both campuses offered an access to the University gym for all the students, interns and professors with an access during opening time to the Gymnasium. We also had access to the buildings and school's infrastructures 24/7, like the computers room. They also offer an access for a room full of cutting-edge tools and machines for engineering students and chemistry students, for their educational or personal needs. In addition to the physical and virtual resources made available to provide us with the most efficient work environment possible, we also had access to a school convenience store called a コンビニ (konbini). It gives the possibility for all the people over each campus to get food in they don't want to go the several cafeterias, restaurants or food-trucks over the campus. It is also possible to buy, with a student discount, school supplies like notebooks or writing materials, but also to borrow a book about a specific subject, or even a laptop or a tablet. They also sell the merchandise concerning SIT called お土産 (omiyage) such as ties, pull-overs or biscuits which are really common to offer in the Japanese culture when you go somewhere else.

The SIT is open to the world, and I met plenty of students, interns and PhD/post-PhD students who were coming from different countries. The exchanges were rich between international students, and a lot of activities were proposed by the Global Learning

Commons' offices, which is the International Office of SIT, to exchange between all the students either Japanese or internationals, to learn Japanese language and culture but also the culture of the other countries.

3. Presentation of the Neural Information Systems Laboratory

The Neural Information Systems Laboratory, one of several research labs at SIT, was recently established in 2022 by **PhD Professor Ryosuke Hosaka**. Initially starting with just three undergraduate students in 2022, the laboratory has since grown to include 12 students and one intern (myself).

3.1. Main Mission and Focus

The main mission of the **Neural Information Systems Laboratory** at SIT is to explore the mechanisms of the human intelligence by investigating both biological and artificial systems. The laboratory focuses on understanding how the brain enables intelligent behaviour and whether this form of intelligence can be replicated in machines. Through research in theoretical neuroscience and machine learning, the team aims to demystify the foundations of intelligence, bridging the gap between biological proceeds and computational models. Also, the brain's complexity makes it challenging to fully understand it from a purely biological perspective. To solve this, the laboratory approaches the problem from a computational angle. By mimicking the brain's functioning through computational models and machine learning techniques, the laboratory seeks to gain insights into the mechanisms of intelligence.

3.2. Laboratory Team

The laboratory is formed by SIT master's and undergraduate's students that are for mostly studying Computer Science or Electrical Engineering.

As you can see below in Figure *1*, I was the first and only intern in the laboratory. The laboratory team made me feel at ease really quickly by organising an arrival party for me, where we could have discussed about the differences between our cultures and what I thought about Japan so far.

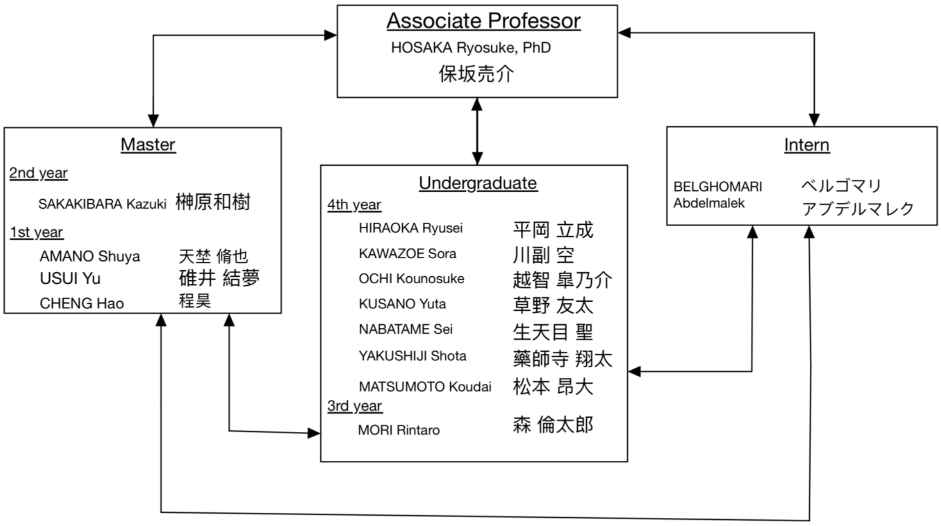

Figure 1 : Diagram of the laboratory team communications

Students of the laboratory; **Sakakibara Kazuki** and **Mori Rintaro** gave me advice on what article to read or lecture video to watch to grasp quickly the subject I had to work on.

3.3. Working schedule and communication inter-laboratories

When I arrived, my supervisor gave me my schedule for the internship, told me the basic rules they apply within the laboratory and the different ways of communication they use between the members. Thereafter, I knew about the different exchange and meetings organized with other Japanese laboratories.

3.3.1. Schedule and organisation

Regarding the schedule, Mr. Hosaka assigned me the same timetable as his students. I was required to be on-site every **Monday, Wednesday, and Friday from 1 pm to 5 pm**. The rest of the time, I had the flexibility to **work remotely** and could organise my time as I wanted, while respecting **the 35-hour workweek.** This arrangement was enriching because it gave me more responsibility for the work I needed to complete and required me to maintain a high level of discipline. It also allowed me to explore Japan while working, which was an incredible privilege.

To ensure that the workflow was maintained, we had to prepare a **report each week** detailing all the work we had done, the books and articles we had read, and the work we planned to do the following week. This report was then presented to all the members of the laboratory, allowing us to share each other's work and enhance our scientific knowledge. It also permitted us to assist others who might be struggling with something we had mastered or seek help if we encountered something we did not understand beforehand.

I used the **Agile methodology** that we learned at ENSICAEN. This approach was essentially based on several two-week sprints aimed at maximising the efficiency. Of course, there were some weeks when the workload was overestimated, allowing me to gain 2 or 3 days of work. However, there were times when I underestimated the amount of work required, which allowed me to benefit from the remaining days more effectively. My methodology was based on trying to achieve each objective. The objectives will be displayed in the 4.1.3.

3.3.2. Communication inter-laboratories

The professor setted up channels of discussions and used platforms and applications to keep us all connected and ready to exchange about our results. In the laboratory, we used **Zoom** to have live meetings and communicate during the weekly report presentation. We used **Slack** to communicate between each other and to stay alert about the laboratory activity. It was moslty used for to alert on events or preventing about an absence or a lateness. **Scrapbox** was the platform used to upload or download files regarding the laboratory or university related documents. This is where we uploaded each report.

Communication with other laboratories was also important. For instance, we were invited to attend and/or participate in a **workshop** that brought together students from various Japanese universities. The main workshop took place at the Nippon Institute of Technology (NIT). It was an incredible opportunity for me to experience what a workshop entails, as it was my first time being involved in such an event. Additionally, it allowed me to interact with Japanese students from other universities. I was the only international student at this event, which was intimidating, but it also pushed me to use some Japanese words that I had recently learned in class.

The laboratory is also keen on encouraging its students to attend or participate in **conferences**. In July, I was given the opportunity to attend a conference thanks to the laboratory’s support. Mr. Hosaka encouraged me to participate in **the IEEE World Congress on Computational Intelligence** (WCCI) in Yokohama. This was an incredible opportunity for me to meet some of the brightest minds in the fields of computing and Artificial Intelligence and. I met numerous researchers and PhD students, attended a full week of meetings, workshops, and conferences, and learned a great deal about AI during that time. It was surreal to discuss ideas with professors whose work we studied in class this year, like Johan A.K. Suykens. Additionally, it was an invaluable opportunity to connect with people working in prestigious laboratories and companies and I made several friends there.

Mr. Hosaka is eager to provide the best environment for his students to conduct research under optimal conditions. His dedication to fostering a supportive and resourceful atmosphere has allowed me to dive deeply into my research. This next section will detail the research I conducted during my internship, including the methods used, the challenges faced, and the key findings.

4. My Research

4.1. Introduction

The primary objective of this internship was to develop and refine a **Machine Learning model** (ML) that simulates brain connectivity and to establish a criterion for ensuring accurate simulations. This model is based on a **Liquid State Machine** (LSM), a type of **Reservoir Computing** (RC) model, and the specific criterion I focused on was the **Echo State Property** (ESP).

To provide context, let’s define these key terms.

4.1.1. What is Reservoir Computing (RC)

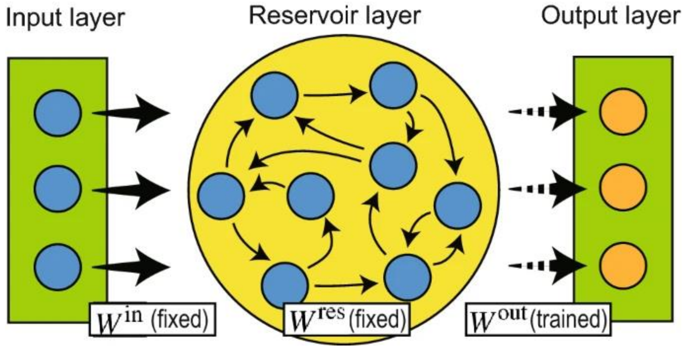

Figure 2 : Scheme of a Reservoir Computer

The main area of research of this internship is about reservoir computing models and their properties.

Reservoir computing (RC) is a computational framework designed to handle complex temporal processing tasks by using dynamic systems.

As defined in (Nakajima & Fischer, 2021), the simplest form of Reservoir Computing is the **Echo State Network** (ESN). ESNs and generally RCs are a type of **Recurrent Neural Networks** (RNNs) where the internal connections, known as the **“reservoir”**, are sparsely and randomly connected, and these connections remain fixed after initialisation(Figure 2). What makes ESNs particularly interesting is their ability to map input sequences into a high-dimensional space, where the dynamics of the reservoir maintain a fading memory of past inputs. This fading memory property is crucial for tasks involving time series data.

The key property and characteristics of an ESN is the **Echo State Property** (ESP), which ensures that the influence of a given input on the network’s state fades away over time like an ”echo”. This property guarantees that the network’s response is stable and reliable. This is what makes it suitable for processing sequential data. Unlike traditional RNNs, the only part of the ESN that undergoes training is the linear read-out layer, which simplifies the learning process and reduces computational costs. Due to their simplicity, ESNs have been successfully applied in various domains, including speech recognition, time series prediction, and robotics (De Azambuja, 2017), where efficient and effective temporal processing is essential.

4.1.2. What is a Liquid State Machine (LSM)

A Liquid State Machine (LSM) can be viewed as an extension of the Echo State Network (ESN) within the realm of reservoir computing. As described by in the paper of (Maass, 2002) and the lecture of (Maass); like an ESN, the LSM is a type of recurrent neural network (RNN) that uses a fixed, randomly connected reservoir to transform input data into a high-dimensional representation, which is then processed by a simple linear read-out layer. However, the key difference lies in the nature of the reservoir.

In an ESN, the reservoir is composed of standard artificial neurons that process continuous input signals. In contrast, the reservoir of an LSM consists of **spiking neurons**, which are more biologically realistic and operate using discrete spikes of activity (similar to the way neurons in the brain communicate). This spiking activity adds an additional layer of complexity and temporal dynamics, allowing the LSM to capture and process information in a way that is more akin to how biological neural systems work. As said in (Zhang & Li, 2015), the concept of LSMs is inspired by the **neocortex**, a vital brain region for functions like sensory perception and conscious thought, characterised by interconnected layers of neurons developed over time.

The introduction of spiking neurons into the reservoir enables the LSM to model time-dependent phenomena more effectively, making it particularly well-suited for tasks that require fine-grained temporal processing.

4.1.3. The Problem to solve

While the Echo State Property (ESP) is well-studied and understood within the context of Echo State Networks (ESNs), its existence and role in Liquid State Machines (LSMs) are less clear. LSMs, which involve more complex neuronal dynamics due to their use of spiking neural networks, hold potential for applications that closely mimic natural brain processes. This report seeks to investigate whether a criterion similar to the ESP can be defined for LSMs, enabling the formulation of necessary and sufficient conditions for their optimal performance.

By exploring the ESP within LSMs, we aim to establish a theoretical framework that enhances our understanding of the computational capabilities of LSMs while providing valuable insights into modelling biological neural networks. Studying the ESP for LSMs could therefore deepen our comprehension of the fundamental principles underlying learning and information processing in both computational and biological neural networks.

My objectives to achieve my internship were:

- **Research** and **understand** what Reservoir Computing is
- **Create** an ESN
- **Experiment** the ESP on an ESN
- **Research** on LSM and the system of neuronal spiking
- **Create** an LSM
- **Experiment** the ESP on an LSM

4.2. The Echo State Property in Echo State Network

To study the existence of the Echo State Property (ESP) in Liquid State Machines (LSMs), I first needed to ensure that I fully grasped the fundamentals. Naturally, my initial step was to examine the ESP within Echo State Networks (ESNs).

4.2.1. Echo State Network Theory

Reading the book *Reservoir Computing* (Nakajima & Fischer, 2021) introduced me to the concept of RC, where training only the readout layer allows for efficient and effective learning from time series data, often requiring less data compared to other machine learning methods. This highlights the potential of ESNs in this area. One aspect of ESNs that I found particularly intriguing is the ability to incorporate feedback loops (Figure *3*). Although not always necessary, these loops can enhance the network's capability to capture dynamic temporal dependencies within the data, making the model even more powerful for certain tasks.

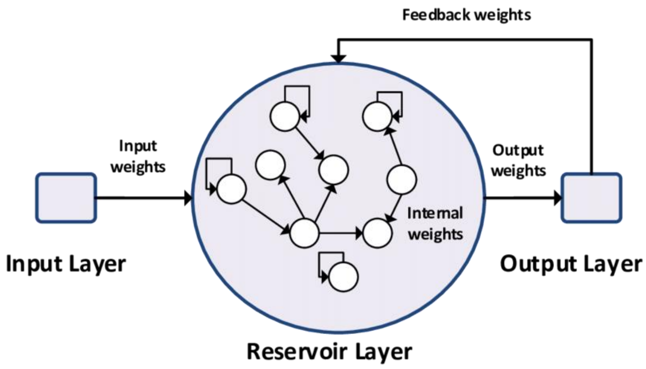

Figure 3: Functioning of an Echo State Network

The equations defining an Echo State Network (ESN) are as follows (Jaeger, 2007):

The basic discrete-time sigmoid unit echo state network with N reservoir units, K inputs, and L outputs is governed by the state update equation:

$$x(n+1) = f\left(Wx(n) + W_{in}u(n+1) + W_{fb}y(n)\right)$$

where $x(n)$ is the N-dimensional reservoir state, f is a sigmoid function (usually the logistic sigmoid or the tanh function), W is the $N \times N$ reservoir weight matrix, W_{in} is the $N \times K$ input weight matrix, $u(n)$ is the K -dimensional input signal, W_{fb} is the $N \times L$ output feedback matrix, and $y(n)$ is the L -dimensional output signal. In tasks where no output feedback is required, W_{fb} is nullified. The extended system state $z(n) = [x(n); u(n)]$ at time n is the concatenation of the reservoir and input states. The output is obtained from the extended system state by:

$$y(n) = g\left(W_{out}z(n)\right)$$

where g is the output activation function.

During my research, I focused on understanding the various **hyper-parameters** that can influence the performance of the reservoir in predicting time-series data. These hyper-parameters include **the spectral radius, input scaling**, and **reservoir sparsity**, each playing a critical role in ensuring that the ESP is maintained and that the network functions optimally.

4.2.2. A closer look at the hyper-parameters

In my exploration of Reservoir Computing, I discovered that specific hyper-parameters significantly influence the model's performance, particularly the **sparsity of connections** in the reservoir and the **spectral radius** of the reservoir weight matrix. Sparsity, which refers to the proportion of non-zero weights in the reservoir, affects the network's connectivity and can lead to more efficient information processing by reducing redundancy and enhancing signal propagation. On the other hand, the spectral radius of the reservoir's weight matrix is critical for maintaining a balance between memory and stability within the network. A **spectral radius close to one** ensures that the reservoir's state dynamics are neither too stable (leading to information decay) nor too chaotic (resulting in instability), thus enabling the network to retain and process temporal information effectively (Jaeger, 2007). By carefully tuning these hyper-parameters, the reservoir can be optimised to capture and predict complex time series data with greater accuracy.

4.2.3. First Experiments

After doing some research on the web on how to implement this kind of model, I found reservoirPy (Xavier & Nathan, 2020), a GitHub repository made and maintained by French researchers. So, my first goal was to implement the ESN using my own code and the one provided by other researchers to test the ESP with time series data and chaotic time series data. I also decided to test all my models with the **Lorenz system** for its chaotic behavior and because it is a canonical example frequently used to test the robustness of dynamical systems models such as ESNs or LSMs

Here are the results I obtained with both ESN implementation:

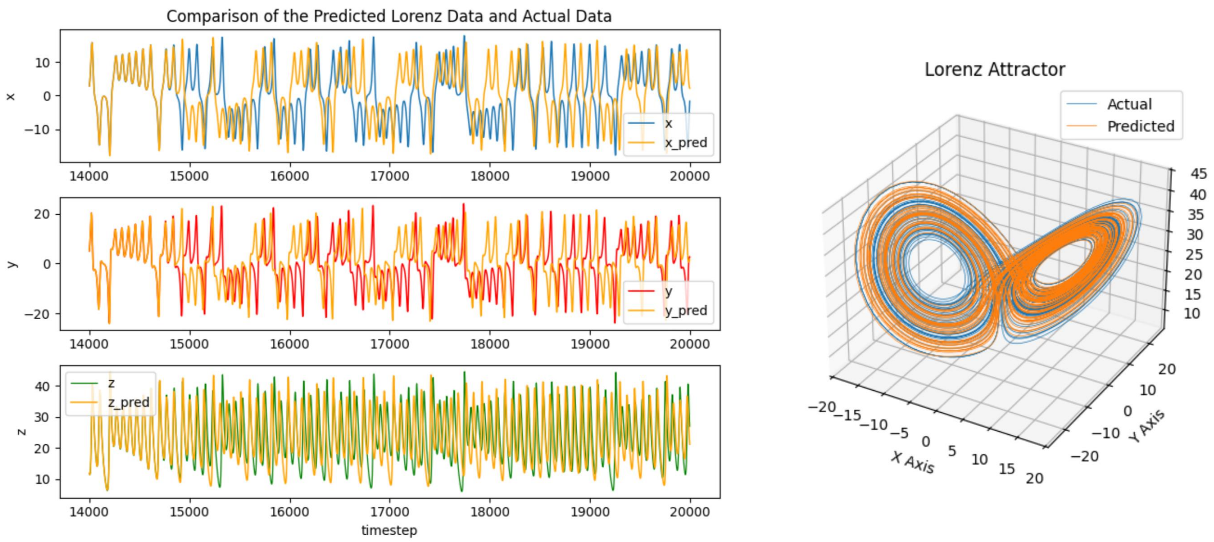

Figure 4: Prediction of the Lorenz System with an ESN

As shown in Figure *4* the ESN's prediction is initially quite accurate, effectively capturing the values and behaviour of the chaotic function. However, as the prediction progresses, the ESN struggles to maintain accuracy and fails to consistently track the chaotic dynamics. The

Echo State Property (ESP) demonstrates its strength in interpreting the values of the predicted function.

The ESP is a fundamental requirement for the reservoir computing principle to work effectively. It states that the reservoir should asymptotically wash out any information from its initial conditions, allowing the input signal to dominate the reservoir dynamics. This property is crucial for ensuring that the network's state is primarily driven by the input rather than lingering initial states.

In practice, the spectral radius of the reservoir weight matrix is a critical factor in achieving the ESP. It is generally observed that when the spectral radius is smaller than one, the ESP is maintained, allowing the network to process inputs effectively without instability. However, contrary to a common misconception, the ESP can sometimes be maintained even with a spectral radius larger than one, particularly when input amplitudes are large (Jaeger, 2007). This suggests that the relationship between the spectral radius and the ESP is more nuanced than simply keeping the spectral radius below one.

As I explored different hyper-parameters in Reservoir Computing, I found that both spectral radius and sparsity of connections within the reservoir significantly impact the model's performance. Adjusting these parameters allows for better control over the reservoir's dynamics, enhancing the network's ability to predict time-series data effectively.

4.3. Echo State Property in Liquid State Machines

Having established the importance of the Echo State Property (ESP) in ensuring the stability and effectiveness of Echo State Networks (ESNs), we can now explore how this property is similarly crucial in the context of Liquid State Machines (LSMs), with some distinct considerations due to the spiking nature of the neurons within the reservoir.

4.3.1. Characteristics of a Liquid State Machine

An LSM processes temporal data through a network of spiking neurons, which are initialised with random weights. These can be **excitatory** or **inhibitory**, influencing the dynamics of the network in distinct ways:

- **Excitatory Neurons**: Increase the likelihood of subsequent neurons firing, promoting activity within the network. They correspond to a positive float in the reservoir's weight matrix
- **Inhibitory Neurons**: Decrease the likelihood of subsequent neurons firing, which helps regulate and balance the network's activity, preventing runaway excitation. On the opposite, they correspond to negatives floats in the reservoir's weight matrix.

As for the ESN, we train only the readout layer only.

We first map the current received by the readout layer for each neuron (Oladipupo):

$$I_y(t) = \sum_i^n w_{oi} \cdot f\big(u(t)\big)$$

Here, $u(t)$ represents the current received by the readout neuron at time t, with w_{oi} being the connection weights from the reservoir neurons to the readout layer, and $f(u(t))$representing the input function over time.

Then, this current is integrated to captures the total input current received by the readout neuron over the interval $[0, T]$:

$$\int_0 I_{y(t)} \ dt = \sum w_{oi} \cdot \int_0 f[u(t)]dt$$

After determining the optimal ratio of inhibitory to excitatory neurons and selecting the appropriate regression function to map the readout layer, I focused my research on the lectures and publications of Professor W. Maass. My attention was directed towards understanding the specific types of **spiking neurons** he employed in his experiments, the methods used to randomise connections within the network, and **the types of synapses** incorporated to introduce noise and delays. These elements are crucial as they mimic the natural variability and timing delays observed in the biological systems of the brain, which are essential for creating more accurate and biologically plausible models of neural processing.

4.3.2. Building a Liquid State Machine

To be able to test the ESP for the LSM, I needed to first take a look at existing Python libraries and tools to help me build one. I first tried to see if I could create a model using my own ESN code or the ReservoirPy library. I documented myself about the Maass model (Maass, 2002) that I wanted to reproduce (because it is the "original" one) to get interpretable results. I had to use some synaptic and neuron properties. For example, I used **Leaky Integrate and Fire** (LIF) neurons (Maass, 2002) (Oladipupo).

What are LIF Neurons?

- LIF neurons are a type of spiking neuron.
- The "leaks" involve a decay of the neuron's electrical potential:

$$\tau_m \frac{dV_m}{dt} = -\left(V_m - V_{resting}\right) + R_m \cdot \left(I_{syn}(t) + I_{background} + I_{inject}(t)\right)$$

where $\tau_m = C_m \cdot R_m$ is the membrane time constant, R_m is the membrane resistance, $I_{syn}(t)$ is the current supplied by the synapses, $I_{background}$ is a constant background current, and $I_{inject}(t)$ represents currents induced by a “teacher.” If V_m exceeds the threshold voltage V_{tresh}, it is reset to V_{reset}, and held there for the length $T_{refract}$, of the absolute refractory period.

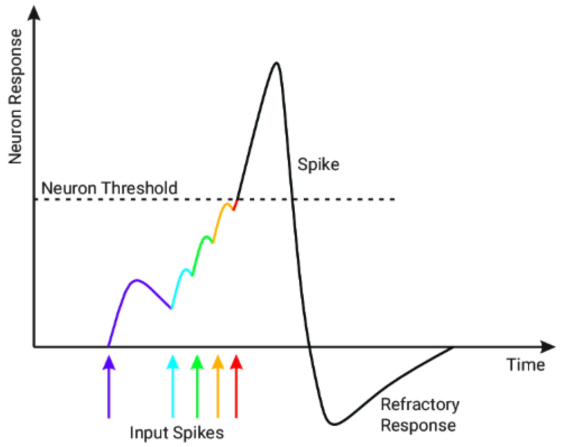

Figure 5: Neuron Respone to spiking activity (Parameshwara, 2021)

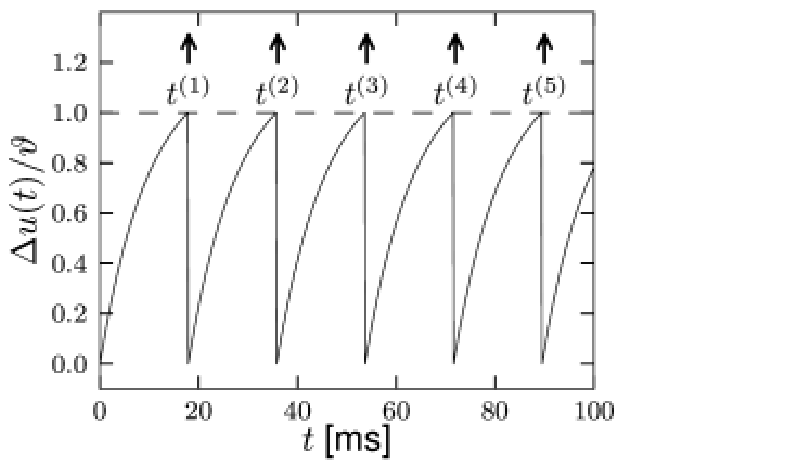

Figure 6: Membrane potential of a LIF in function of time

The Figure 5 is a depiction of the dynamical activity of a spiking neurone. The neuron receives input signals from either the data or lower layers (depicted here as coloured arrows), causing fluctuations in its membrane potential, denoted as $u(t)$ in this paper. When $u(t)$ surpasses a threshold value V_{thresh}(indicated by the dotted line), the neuron generates a spike and then enters a refractory period during which it is less likely to produce another spike for a short duration. In the code, it is translated to the neuron being unable to fire a spike for a refractory period of time $T_{refract}$.

The Figure 6 describes the time evolution of the membrane potential a LIF neuron driven by a constant input current $I_0 = 1.5$. The change in voltage, $\Delta u(t) = u - u_{rest}$, is scaled relative to the threshold value ϑ. The input current units are selected so that $I_0 = 1$ results in a trajectory that eventually reaches the threshold as t → ∞. After a spike occurs, the membrane potential is reset to the resting potential $u_r = u_{rest}$.

Considering this choice of representations, I first wanted to look if someone already made such a model. I contacted the researchers from **INRIA** (one of the biggest French laboratories), that have implemented reservoirPy, and they told me that their library wasn't ready to create an LSM yet. So, I looked up to create one by myself. But that would have been intricate and complex due to the brain mimetic of biological neurons. It would have taken a long time to create a library to do so. So, I looked on the Internet if there was any library to code biological neurons. I found two of them: **Brian Simulator** and **NEST Simulator**.

Brian seemed easier to use and grasp, while **NEST Simulator** was harder to get hands-on with but more comprehensive. As a result, I preferred using **NEST Simulator.**

By doing some research on NEST-simulator, I found the LSM GitHub Repository (Subramoney, 2017) done by the PhD researchers working with W.Maass. As they followed all the settings described in the Maass' experiment, I used the main code to get the settings from the original experiment. I adapted the code to grasp the nest library and get and inject the spikes into the LSM reservoir to get the output and train it.

4.3.3. Experimentation on Liquid State Machine predictions

For the experimentations, I used two types of inputs; one provided by the git repository trying to predict the results of an **XOR function** and the other one being predicting the chaotic **Lorenz function**.

In the meantime, as it took time to understand all the complexities of NEST, I tried to adapt Professor R. Hosaka's code to obtain and interpret the first results. Professor Hosaka's model was simple yet complete. This model uses the same processes as the code that I developed for the ESN, but it added the LIF neuronal spiking model in the reservoir. The weights in the reservoir are either positive or negative, with an inhibitory/stimulating ratio, to replicate the behaviour of a real LSM reservoir. During the experiments, the ratio was maintained at 80%, as it shouldn't be a hyperparameter that influences the ESP. For the NEST model, I used the *iaf_psc_alpha* neuron model[1]. Here are the results of the experiments I obtain with the Lorenz input:

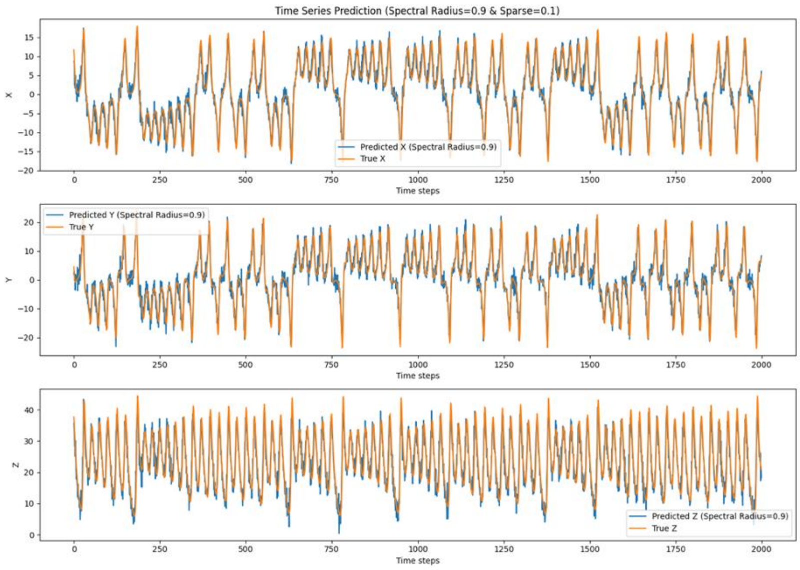

Figure 7: Prediction of the Lorenz system with the simple model

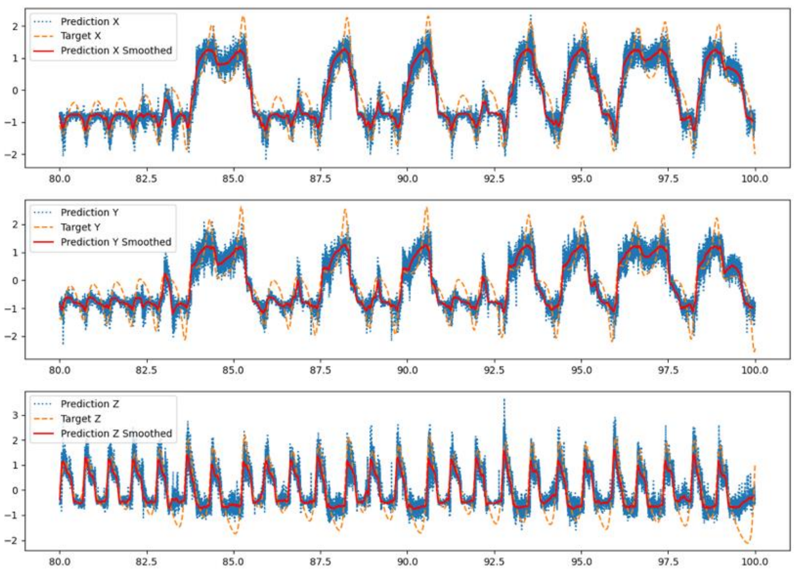

Figure 8: Prediction of the Lorenz system with the NEST model

[1] For more about the documentation, visit NEST's website

4.3.4. Results Interpretations

First of all, the results that we got with both LSM models shows that the prediction we obtained is **better than the one with an ESN**. This improvement is likely due to the reservoir's enhanced ability to capture information **and the biomimetic "memory"** being more effective for chaotic functions.

Second of all, we can see that the predictions are jittery. This comes from the fact that reservoir's neurons use the LIF neuron model. As we saw in Figure 6, the IAF activation function looks like a step function, contributing to this jitteriness.

Finally, the model using NEST is significantly **more complex** than the simpler model. This increased complexity likely accounts for its more noisy predictions, but it also enables the model to better capture the behaviour of the function.

From these results, I experimented to tune the reservoir sparsity or the spectral radius to see if I can find any ESP. As expected, the larger the spectral radius ρ was above one, the worse the prediction was (Figure 9).

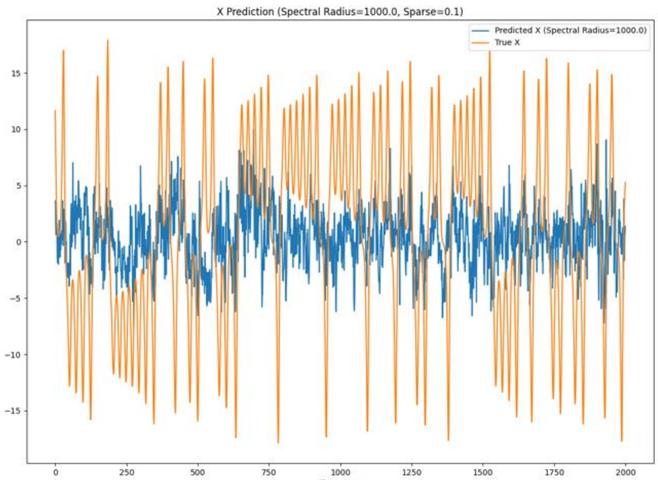

Figure 9: Prediction with a high spectral radius and low sparsity

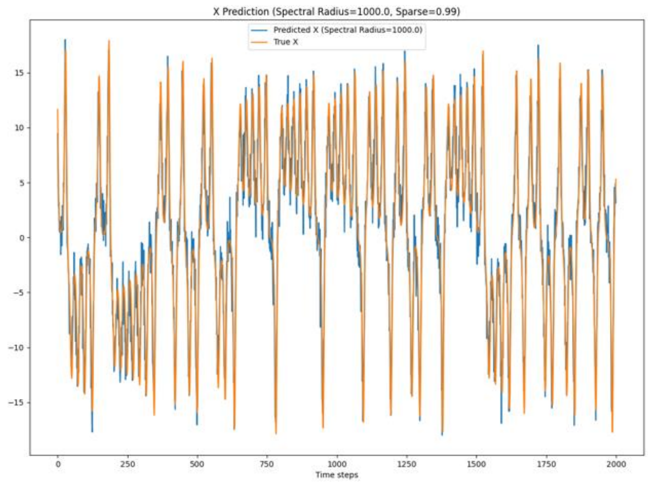

Figure 10: Prediction with a high spectral radius and high sparsity

Additionally, I also noticed something else. If I increase the spectral radius abusively high; $\rho = 1000.0$, increasing the reservoir's sparsity to a very high level, as $sparsity = 0.99$ ($sparsity$ is a value between 0 and 1), helped **cushion** this and resulted in reasonably accurate predictions (Figure 10).

At first, I wanted to find a theoretical way to determine the ESP, but I didn't find any documentation on the internet or in the documents we had in the laboratory. After discussion with some other researchers, they advised me to find experimental methods to determine the ESP empirically. Not wanting to lose more time searching for a mathematical proof, I decided to proceed with **empirical** methods, believing that this approach would eventually help me recognize when a theoretical result might be acceptable or not.

4.3.5. Further Experimentations

I searched for methods to test the ESP, but I haven't found anything conclusive. I then came up with two simple methods:

- Test the same input with different Reservoir initial Conditions:

We give a random initialization to the internal states within the reservoir. By feeding the model with the same input, we can determine the ESP by seeing if the output differs. If the output remains the exact same no matter of the initial conditions, this is when we know that the ESP is respected.

This is a straightforward and **naive** method to test the criterion. It does not require significant modifications to my code and provides **a macro-level interpretation** of the ESP, though it is still **imprecise**.

- Test with the Lyapunov Exponent:

The **Lyapunov exponent** (Wikipedia, 2024) is a quantity that characterizes the separation of infinitesimally close trajectories (Figure 11). Quantitatively, two trajectories in phase space with initial separation vector δZ_0 diverge (provided that the divergence can be treated within the linearized approximation) at a rate given by $|\delta Z(t)| \approx e^{\lambda t}|\delta Z_0|$, where λ is the Lyapunov exponent.

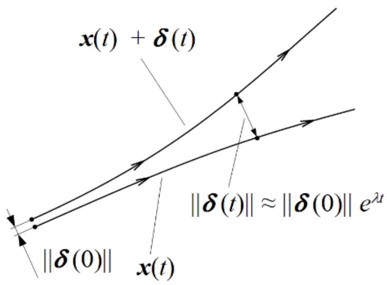

Figure 11: The Lyapunov Exponent defining the separation of two trajectories (Wikipedia, 2024)

This is how I obtain the Lyapunov exponent to test the ESP:

$$|\delta Z(t)| = e^{\lambda}t|\delta Z_0| \Leftrightarrow e^{\lambda}t \approx \left|\frac{\delta Z(t)}{\delta Z_0}\right|$$

$$\Rightarrow \lambda t \approx \ln\left(\left|\frac{\delta Z(t)}{\delta Z_0}\right|\right)$$

$$\Rightarrow \lambda \approx \frac{1}{t}\ln\left(\left|\frac{\delta Z(t)}{\delta Z_0}\right|\right)$$

To keep the starting distance infinitesimal, we must verify:

$$\delta Z_0 \to \epsilon\,,\ \epsilon > 0$$

Thus, we have:

$$\lambda = \lim_{(t\to\infty)} \lim_{(\delta Z_0\to\epsilon)} \frac{1}{t}\ln\left(\left|\frac{\delta Z(t)}{\delta Z_0}\right|\right)$$

Despite my efforts to implement the Lyapunov exponent as a measure for the Echo State Property (ESP), the results remain unreliable as of the time I am writing this paper. Surprisingly, even in degenerate models, such as those with a high spectral radius and low sparsity, the Lyapunov exponent suggests that the ESP is respected, which contradicts my expectations.

However, I have performed macro-level evaluations to assess the quality of the output, and based on these, I know that with a spectral radius close to $\boldsymbol{\rho} = \mathbf{1}$, the results is consistent regardless of sparsity. This is particularly true when predicting the Lorenz chaotic system with a Liquid State Machine (LSM) featuring an excitatory/inhibitory ratio of 80%.

5. Conclusion and Perspectives

Firstly, the last results indicate that my current implementation of the Lyapunov exponent for evaluating the Echo State Property still needs further refinement. However, I successfully created a Liquid State Machine (LSM) using both a basic biological model and a more complex one with the NEST simulator. These models are significant because they allow me to replicate the biological functioning expected from an LSM, which I hope will help my laboratory fellows in the future. I still can empirically evaluate the ESP from a macroscopic scale. Finishing implementing the Lyapunov method could be a next step for this research. I am also interested in switching the functioning of the reservoir neurons from LIF model to Hodgkin-Huxley model (the most realistic one), which is more genuine in the sense of biological brain neurons model. This internship got me better using python, either by creating my own homemade AI from scratch or by implementing a complex and comprehensive library.

Secondly, the results I obtained in these 4 months of research, and experimentations are promising. I gained significant knowledge in Reservoir Computing, Spiking Models, and Brain Information Processing—topics I never imagined understanding when I arrived at the laboratory. I experimented a lot, from trying to understand and create a simple ESN to studying the intricate science of the brain and spiking information communication with the LSM.

As for the environment in which we worked, it was somewhat different from what I was accustomed to at ENSICAEN, even though we remained in an academic context. What I appreciated about the way we worked was the flexibility we had in choosing our schedules and methods. I noticed that I prefer to dedicate the early hours of my day to tasks that require deep focus. This allowed me to be productive early in the morning and reserve the late afternoon for activities like going to the gym. Additionally, I took the opportunity to explore the country by visiting cities around Tokyo, often working on the train or in local coffee shops, which gave me a great balance between work and discovering Japan.

This experience contributed to my personal growth, both mentally and intellectually. In fact, I had to work on a subject I wasn't familiar with. I had to make research on my own, as a thesis student would do, and I managed to get satisfying results. Throughout this journey, I had the chance to exchange with different researchers, and I also went to different conferences in order to share my results, and to learn from others. I have sharpened my skills in the area of machine learning and AI, but also in English and in Japanese. I also developed several relationships within the laboratory and outside it. I successfully made connections and learned how to communicate within the scientific community and with recruiters, particularly at events like the

AI conference in Yokohama. I also learnt how to manage a budget in the context of scientific research like buying specialized books or access to articles, research material and conferences entrance fees.

Thanks to this internship, I know now more than ever that I want to work abroad after my graduation at ENSICAEN. I am even thinking about starting a PhD program in the field of AI applied in neurosciences. I had the opportunity to receive some advice from many researchers, which led me to the consideration of doing a CIFRE. The traineeship helped me to develop and enhance essential qualities for an engineer such as problem-solving skills, adaptability and a technical expertise in the domains of machine learning and neuroscience. I know that a four-months experience is not enough to become an expert, but it has provided me the basis necessary to work in AI after my graduation. These are among the qualities and domains of knowledge I will aim to further improve during my final year at ENSICAEN.

6. Bibliography

De Azambuja, R. (2017). Neurorobotic simulations on the degradation of multiple column Liquid State Machines. *IEEE*.

Jaeger, H. (2007). *Echo State Network*. Retrieved 07 28, 2024, from Scholarpedia: http://www.scholarpedia.org/article/Echo_state_network

Maass, W. (2002). Real-time computing without stable states: a new framework for neural computation. *Neural Computation*.

Maass, W. (s.d.). *Liquid State Machines: Motivation, Theory, and Applications.* Récupéré sur https://igi-web.tugraz.at/PDF/189.pdf

Nakajima, K., & Fischer, I. (2021). *Reservoir Computing: Theory, Physical Implementations, and Applications.* Springer.

Oladipupo, G. G. (n.d.). *Research on the concept of Liquid State Machines.* Archive of Heriot-Watt University.

Parameshwara, C. (2021). SpikeMS: Deep Spiking Neural Network for Motion Segmentation. *IEEE*.

Subramoney, A. (2017). *LSM.* Récupéré sur Github: https://github.com/IGITUGraz/LSM

Wikipedia. (2024). *Lyapunov Exponent*. Récupéré sur Wikipedia: https://en.wikipedia.org/wiki/Lyapunov_exponent

Xavier, H., & Nathan, T. (2020). *ReservoirPy*. Retrieved from https://github.com/reservoirpy/reservoirpy

Zhang, Y., & Li, P. (2015). A Digital Liquid State Machine With Biologically Inspired Learning and Its Application to Speech Recognition. *IEEE*.

APPENDICES

FIGURES TABLE

APPENDICES

Appendix 1 : Non-biblographic sources

Source of the Figure 2: https://www.nature.com/articles/s41598-020-78725-0

Source of the Figure 6: https://neuronaldynamics.epfl.ch/online/Ch1.S3.html

ENSI CAEN

ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

Ecole Publique d'Ingénieurs en 3 ans

6 boulevard Maréchal Juin, CS 45053
14050 CAEN cedex 04