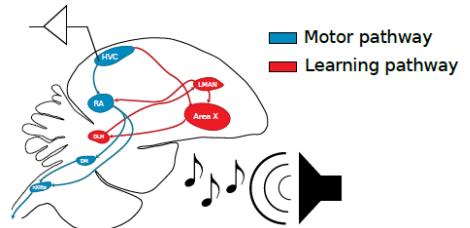
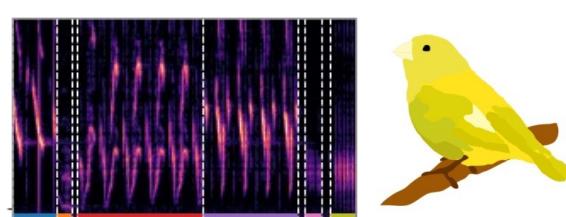
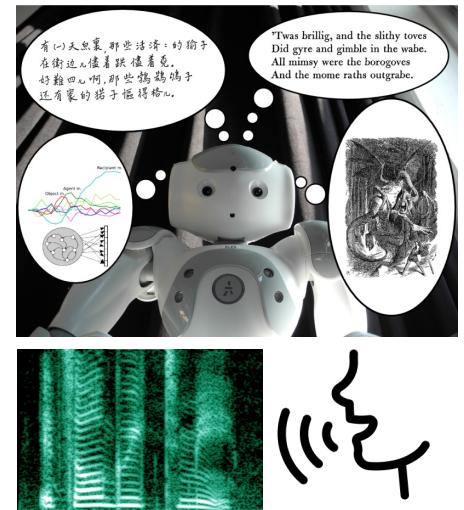
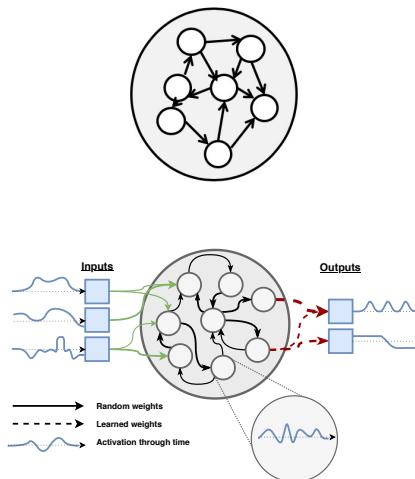


Mercredi 8 Janvier 2025 – Institut d'Optique – M2

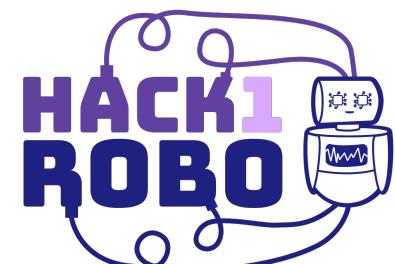


Entraîner un RNN avec le Reservoir Computing

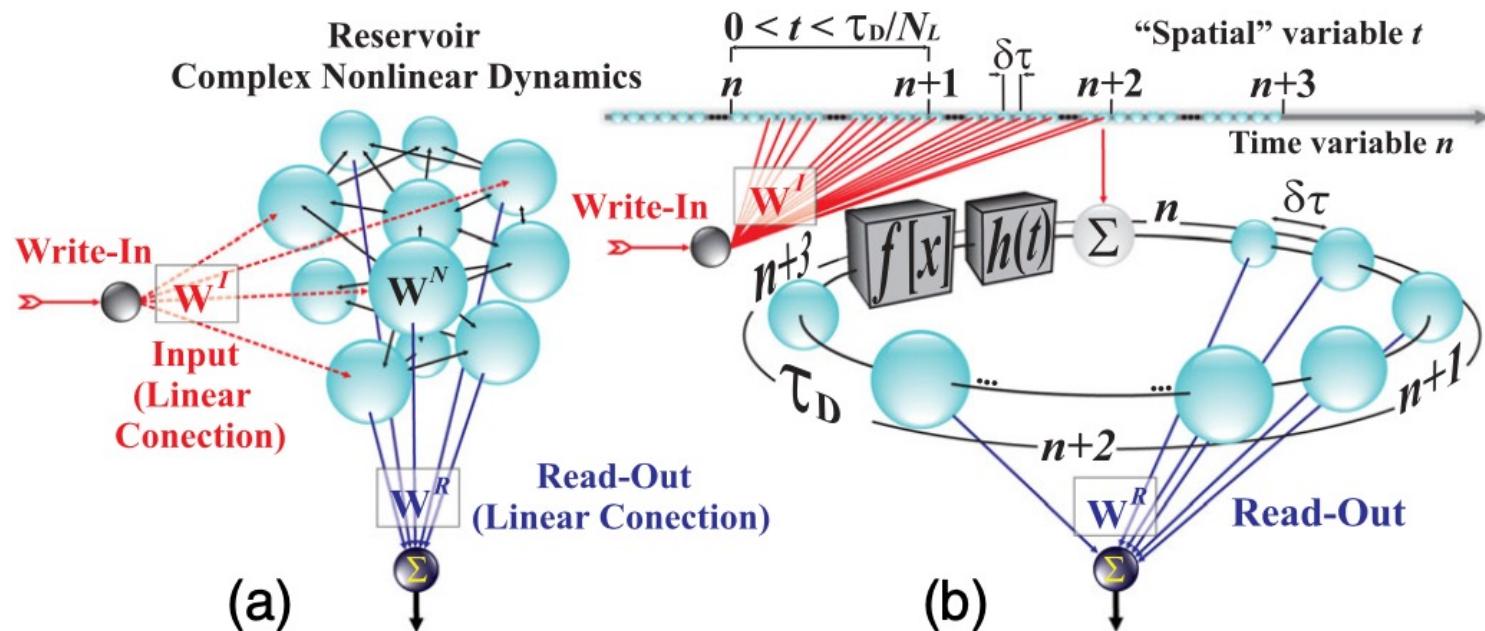
Prédire, générer, discriminer des séquences ...



xavier.hinaut@inria.fr
 @neuronalX



Vers le Reservoir Computing Optique

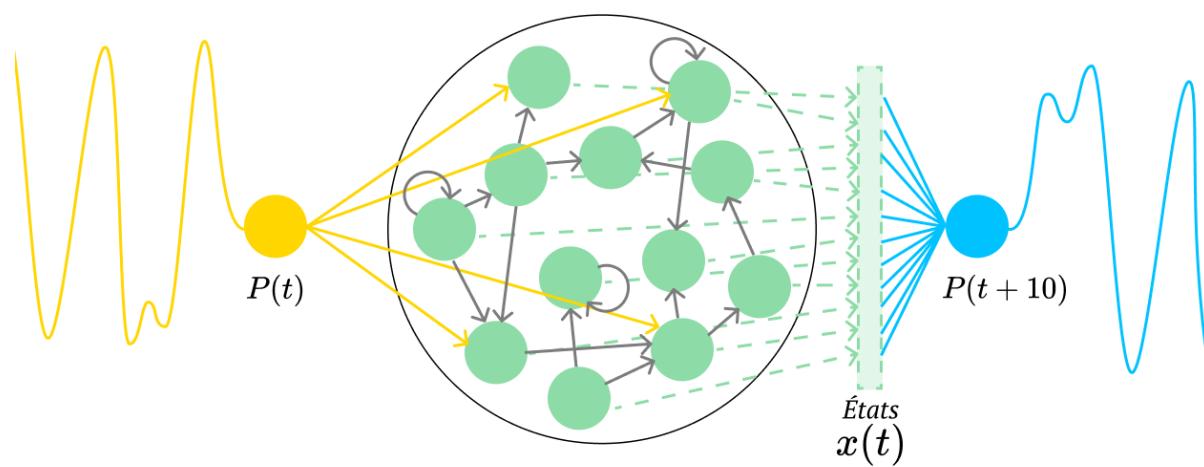


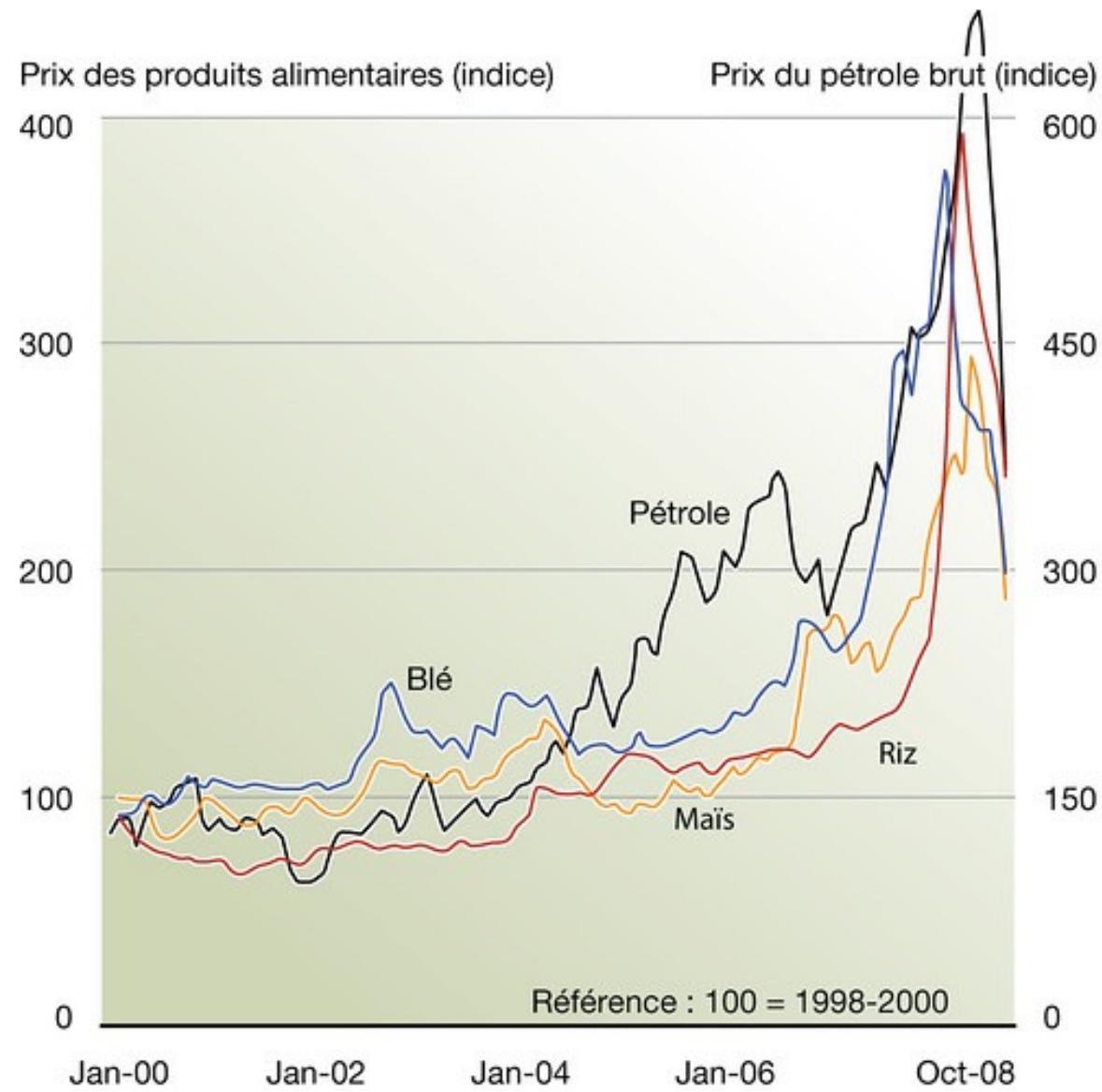
Larger et al. 2017

Xavier HINAUT - Mnemosyne

A quoi servent ces Réseaux de Neurones Récursifs ?

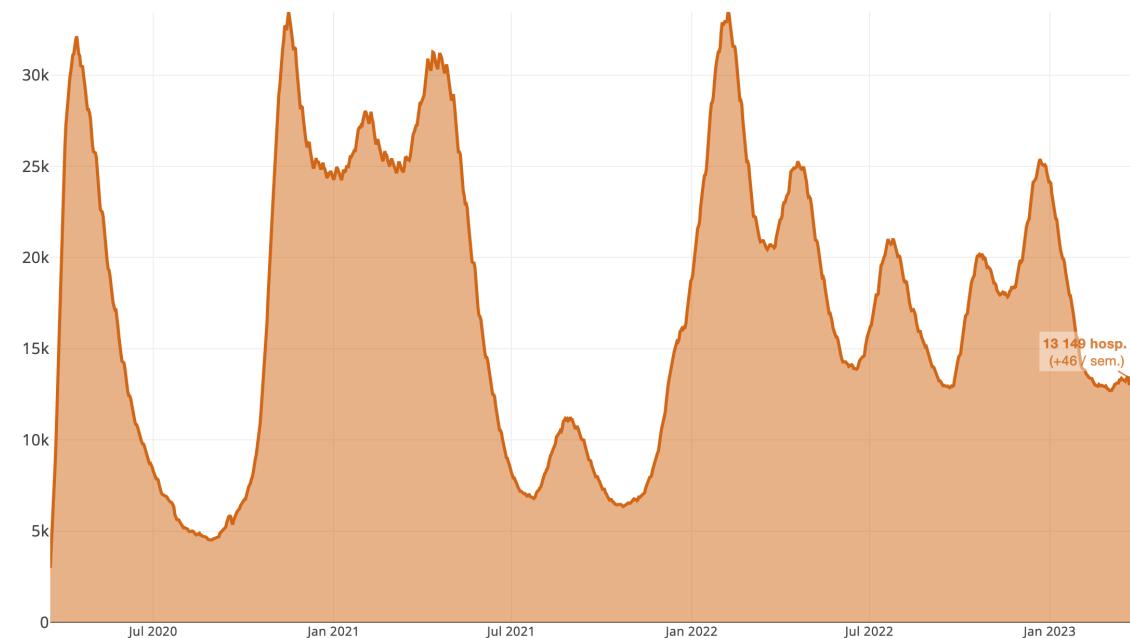
Prédiction de séries temporelles





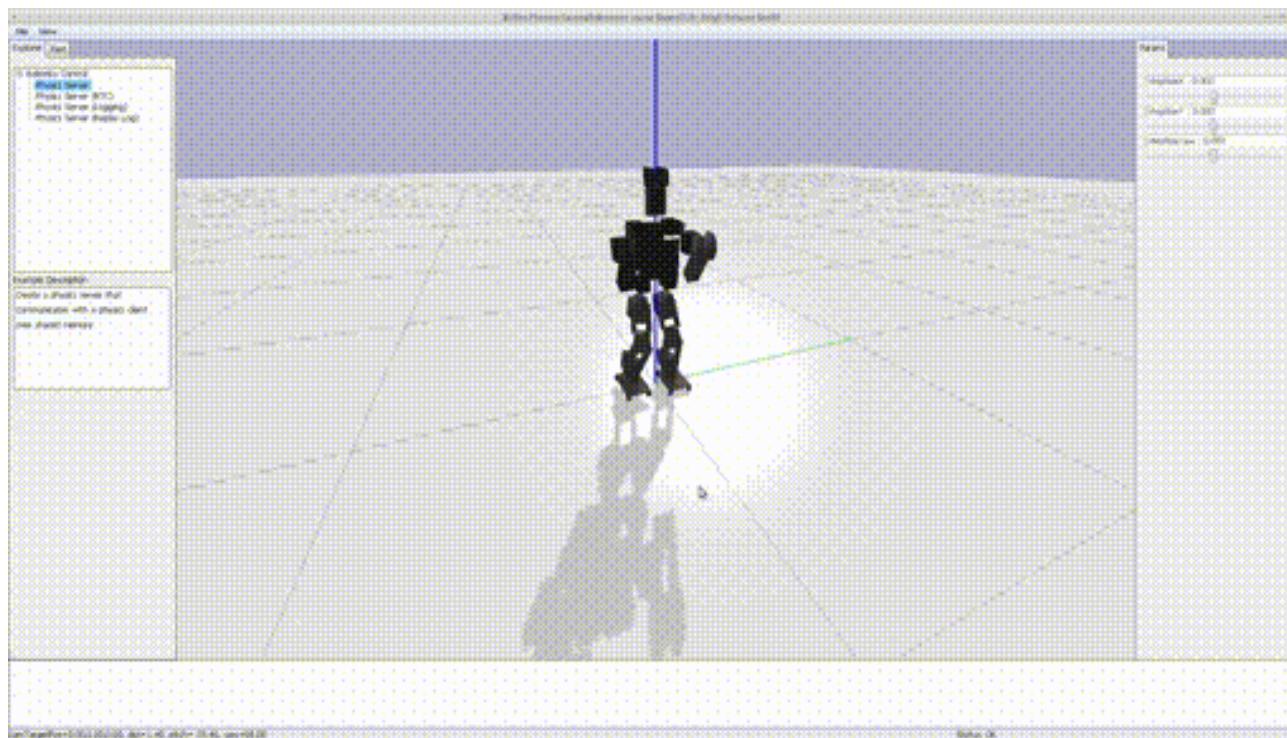
Prédiction de séries temporelles

Nombre de personnes hospitalisées
Nombre de personnes hospitalisées avec Covid19.



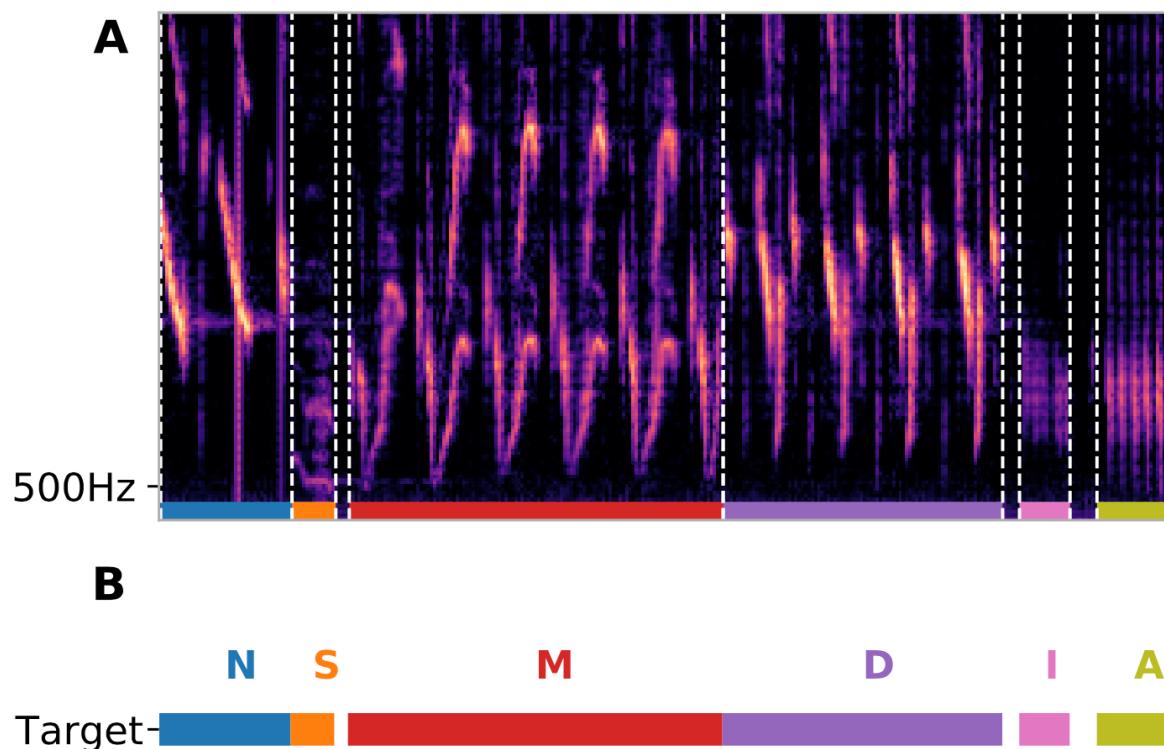
CovidTracker.fr • Données : Santé publique France • Dernière donnée : 31 / 03 / 2023

Prédiction de séries temporelles

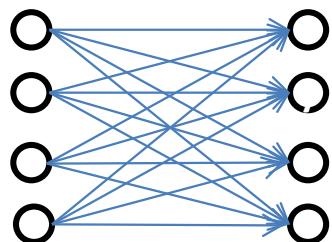


Discrimination de sons

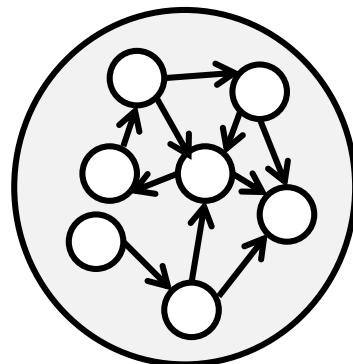
(syllabes de canari)



2 types de réseaux

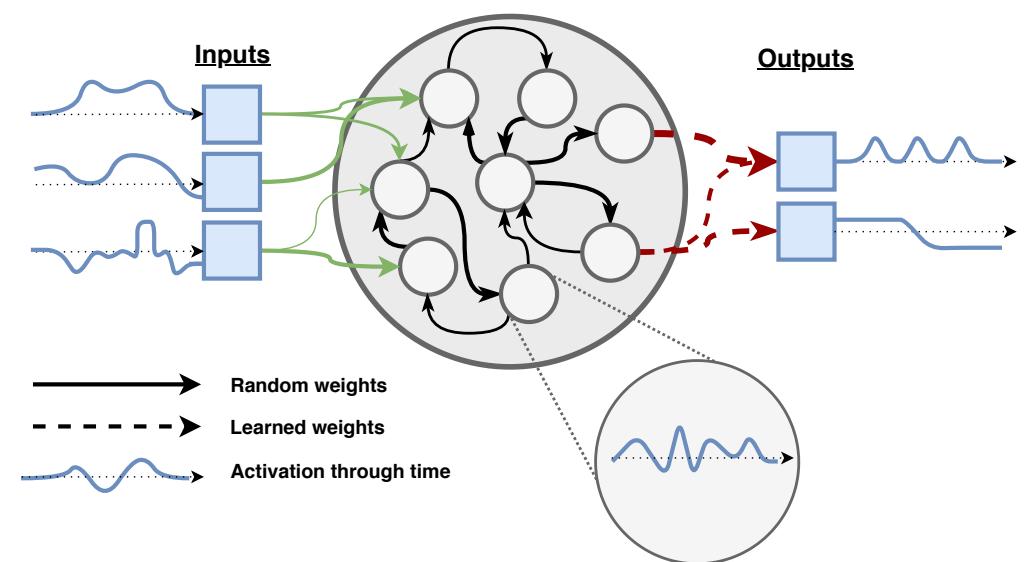
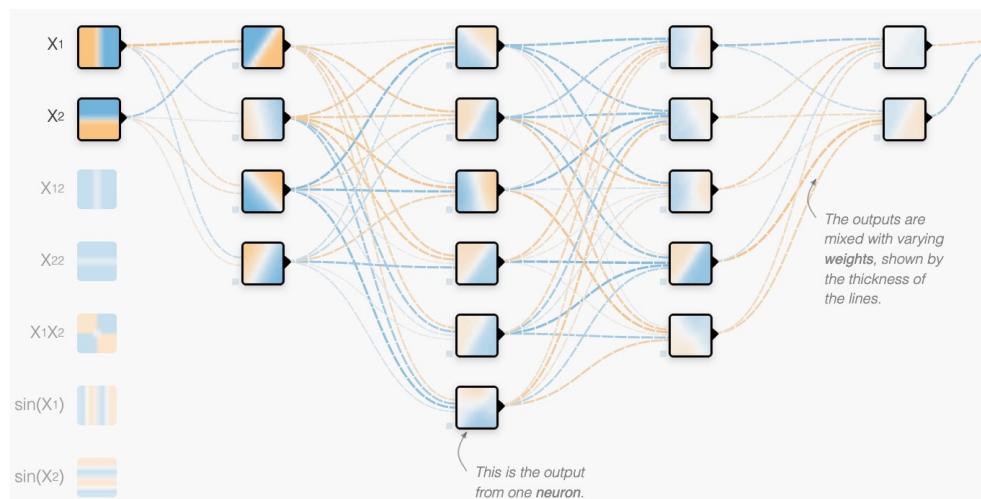


- Réseau de neurones en couches
 - Connexions entre neurones de couches différentes
 - Associations

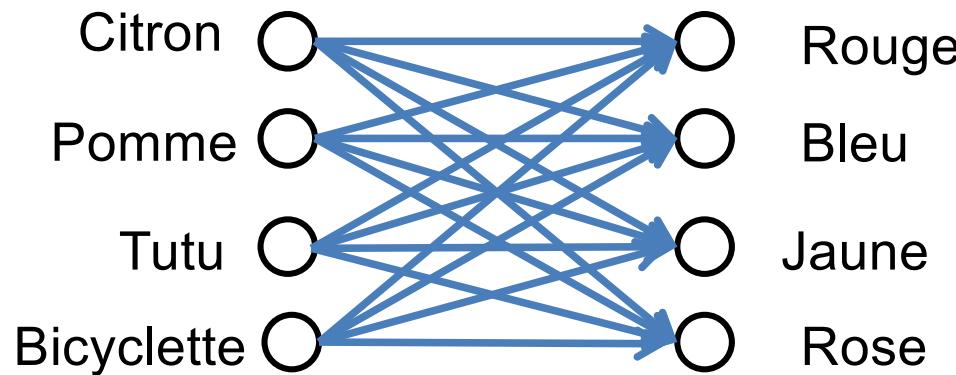


- Réseau de neurones récurrent
 - Mémoire dynamique
 - Représentation du contexte temporel

○ Neurone → Axone (connexion entre deux neurones)



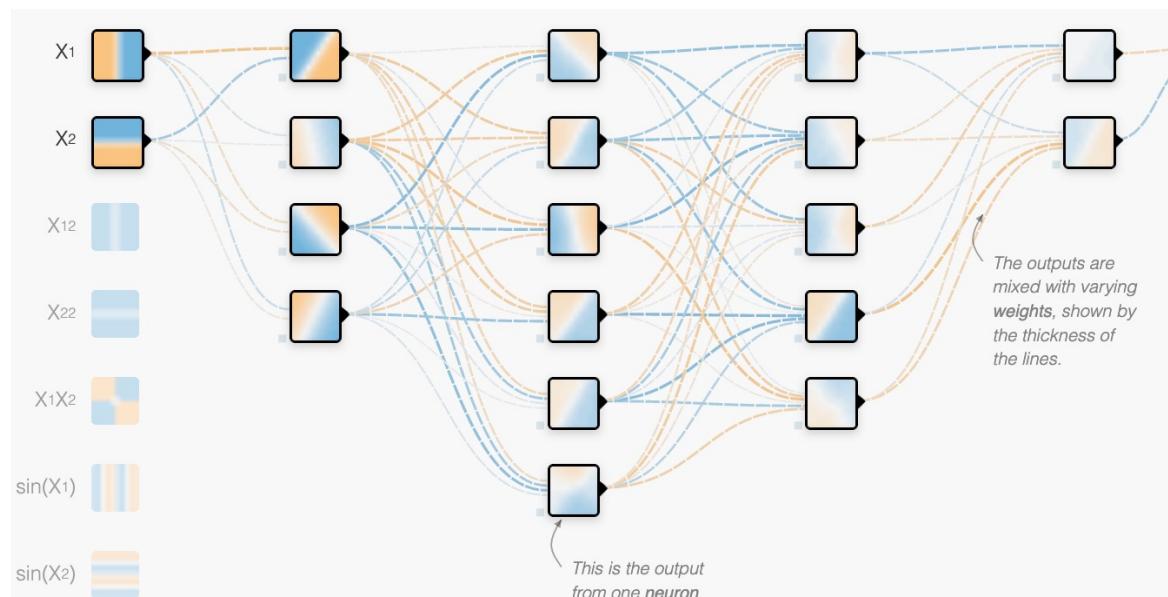
Un réseau de neurones en couches



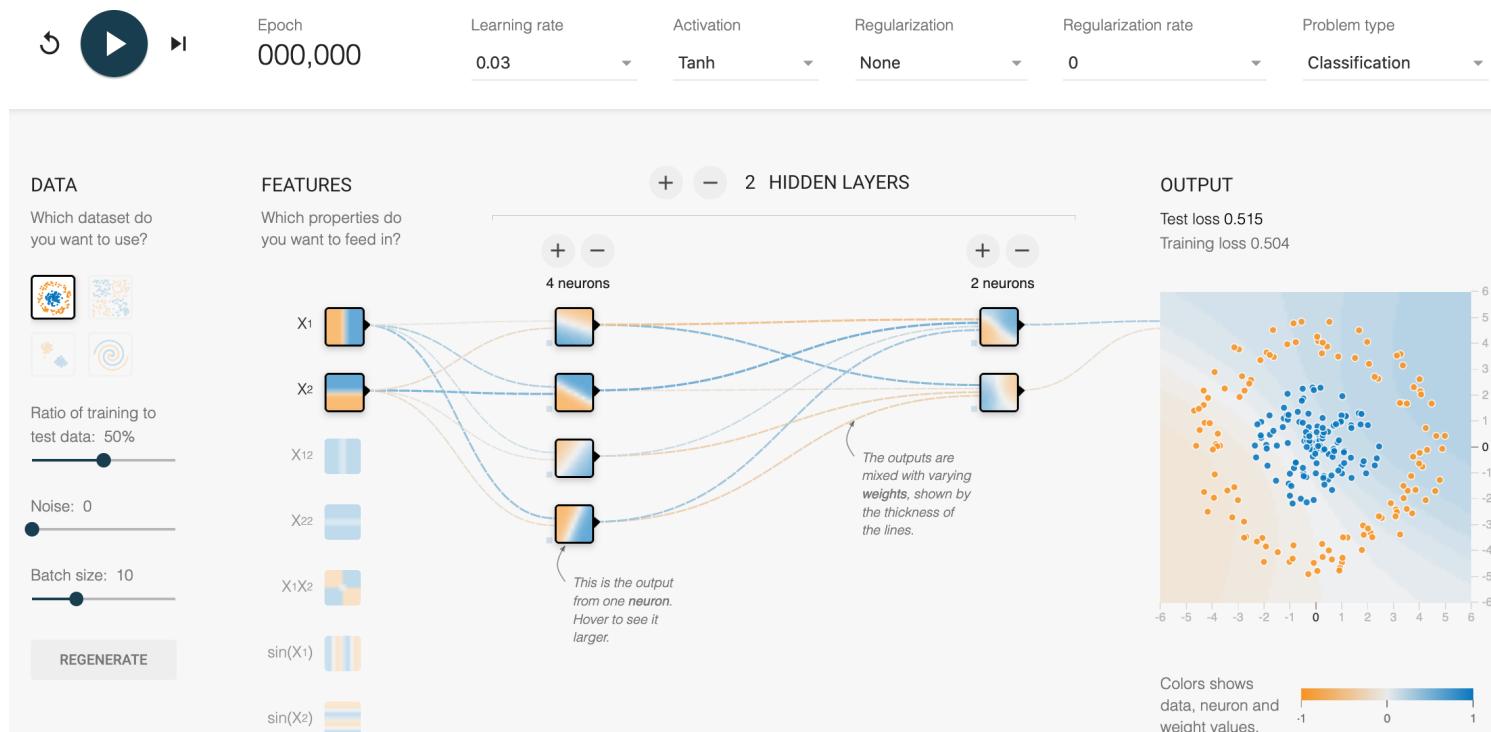
- Loi de Hebb (1949)
 - lorsque 2 neurones sont actifs en même temps leurs connexions sont renforcés
 - « Neurons that fire together, wire together »

Un réseau de neurones en couches

- Avec l'algorithme de rétro-propagation du gradient, on peut apprendre avec un empilement de couches



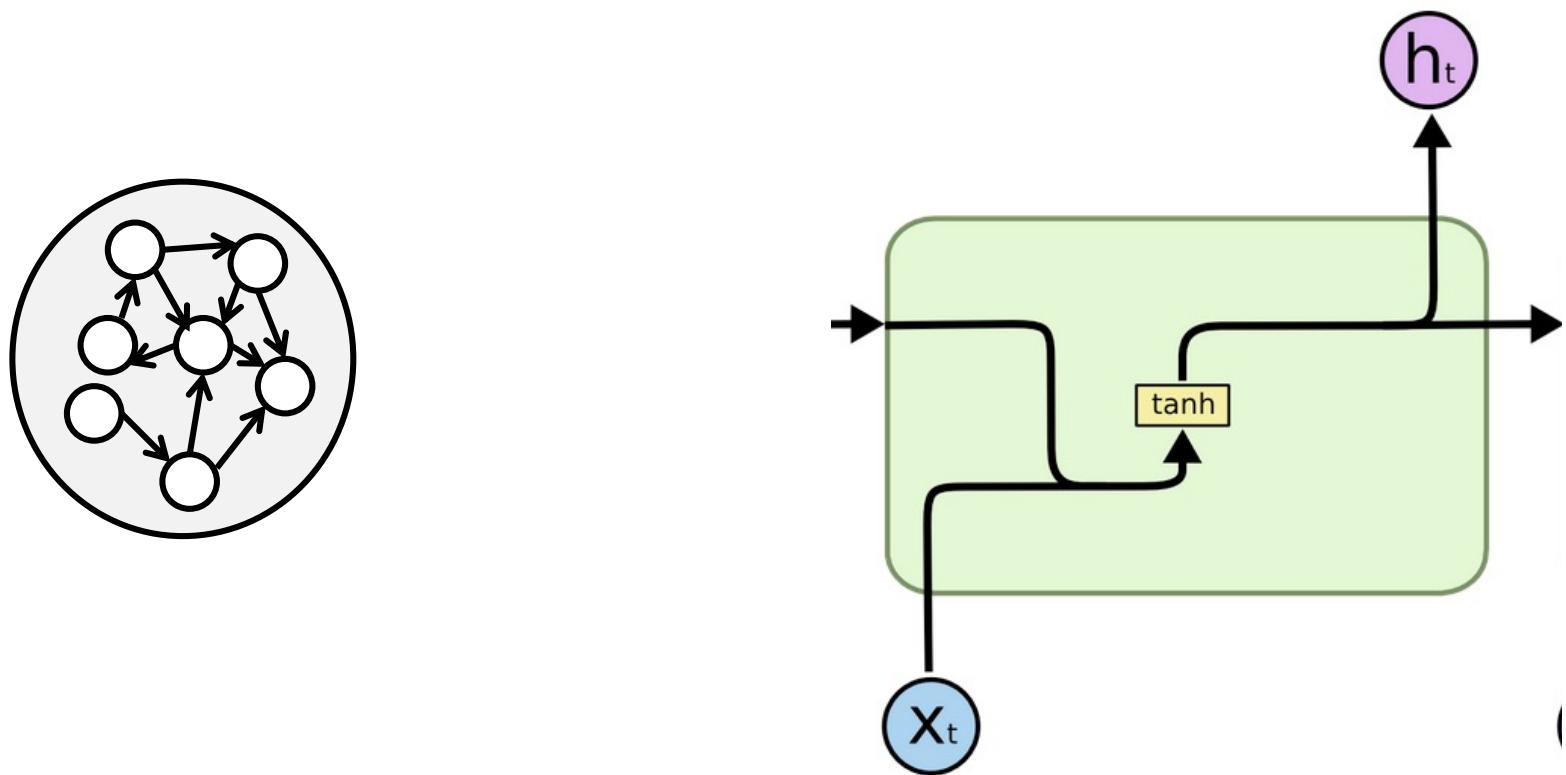
Testez votre intuition des réseaux de neurones



- <http://playground.tensorflow.org>
- <http://binaire.blog.lemonde.fr/2017/10/20/jouez-avec-les-neurones-de-la-machine/>

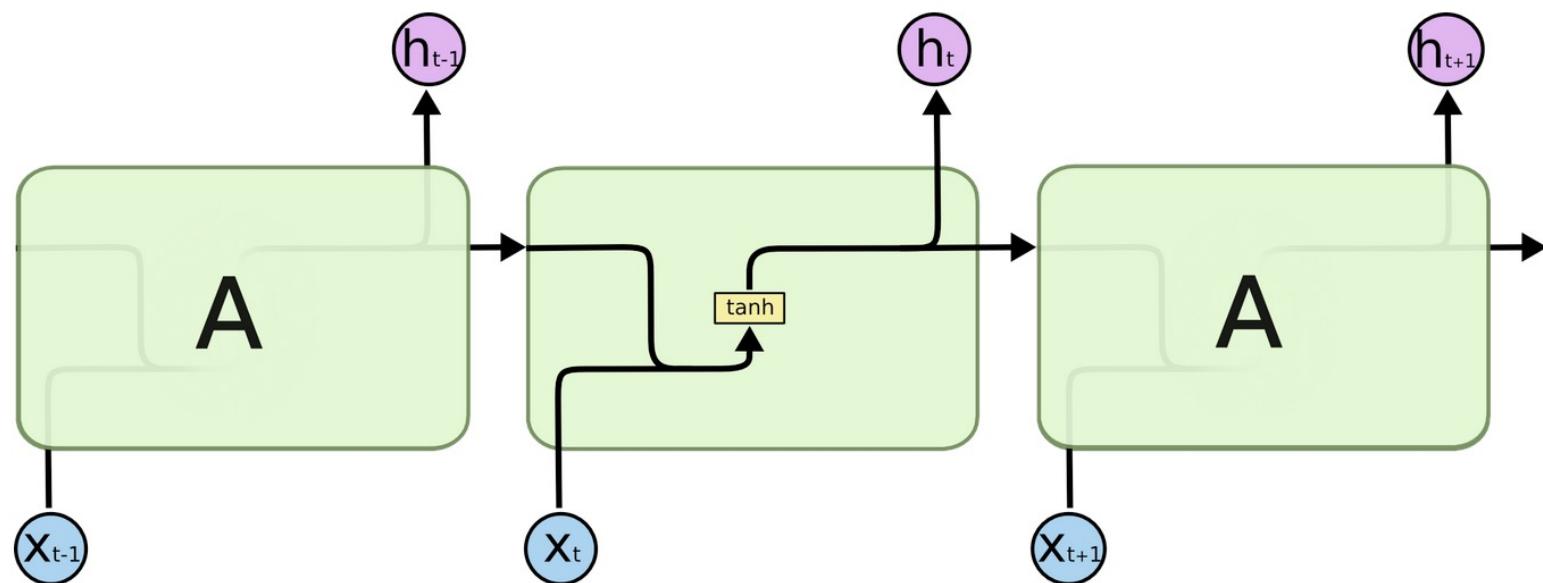
RNN

(Recurrent Neural Network)



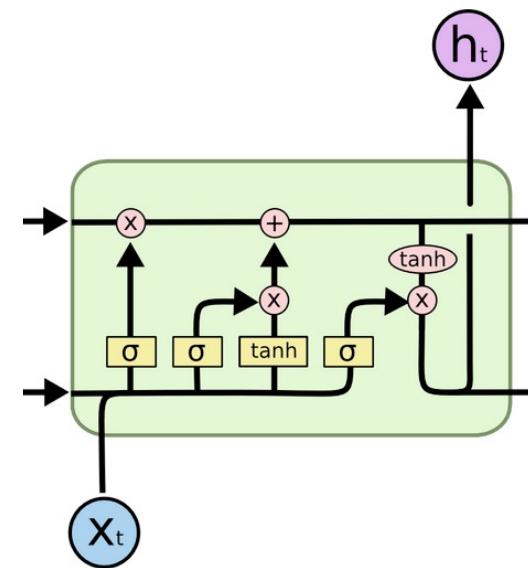
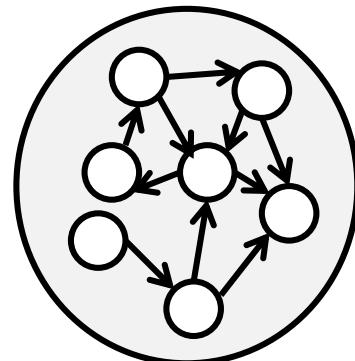
RNN

(Recurrent Neural Network)



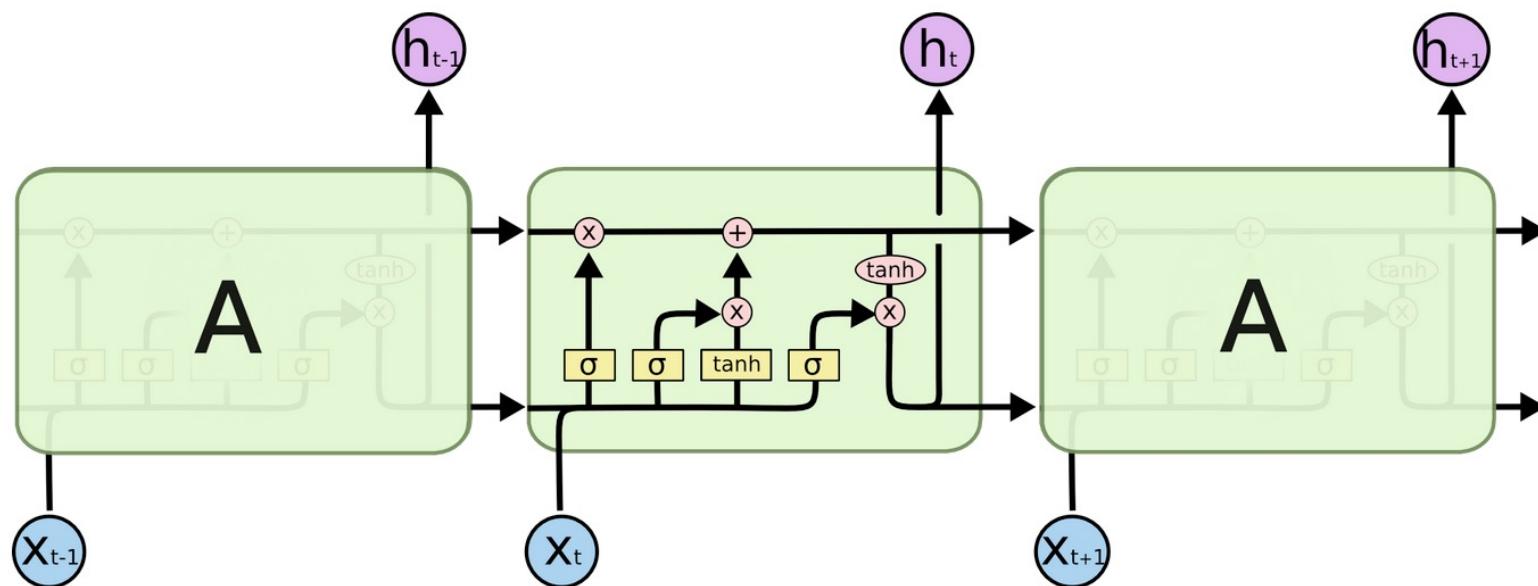
LSTM

(Long Short Term Memory network)

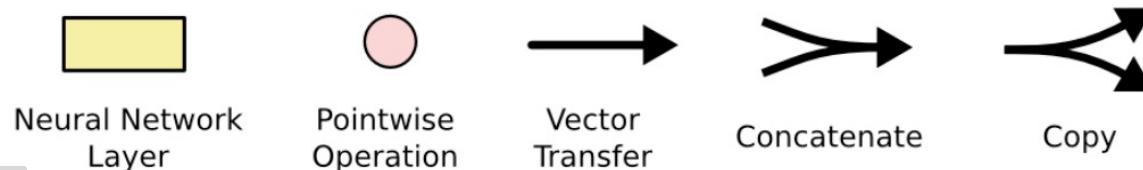


LSTM: Long Short-Term Memory Networks

Hochreiter & Schmidhuber (1997)



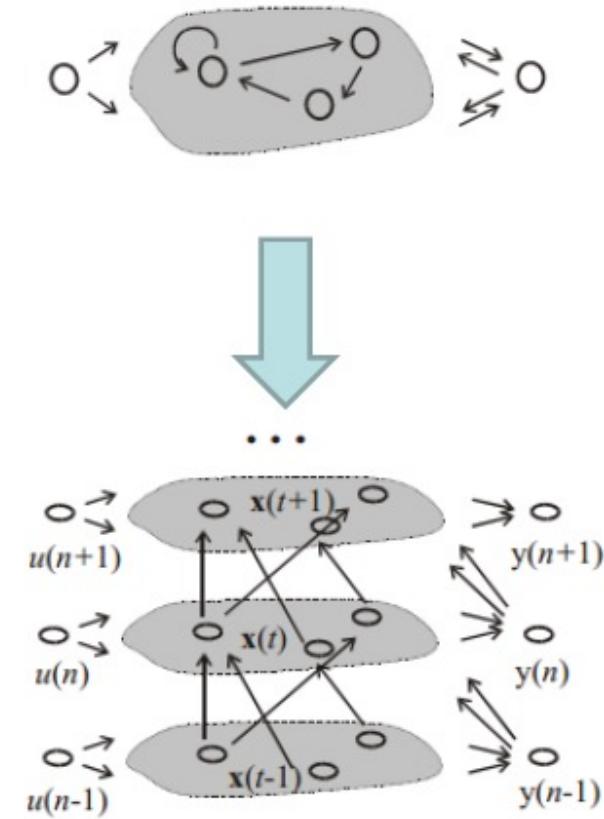
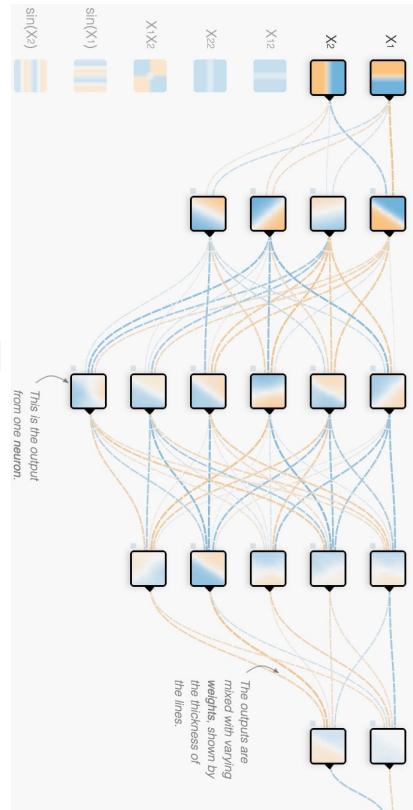
The repeating module in an LSTM contains four interacting layers.



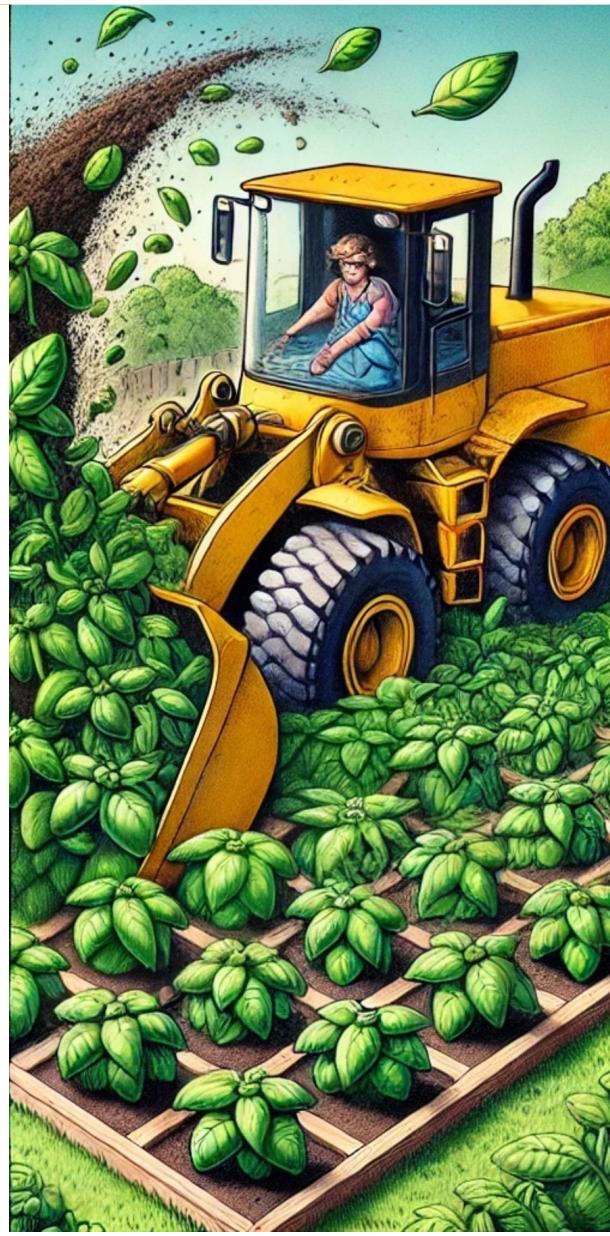
Entraîner un RNN/LSTM avec la rétro-propagation du gradient = virtualiser le temps

Ceci est vrai pour un réseau de neurones récurrent (RNN) classique, pas un réservoir !

Nb couches
= taille du contexte temporel

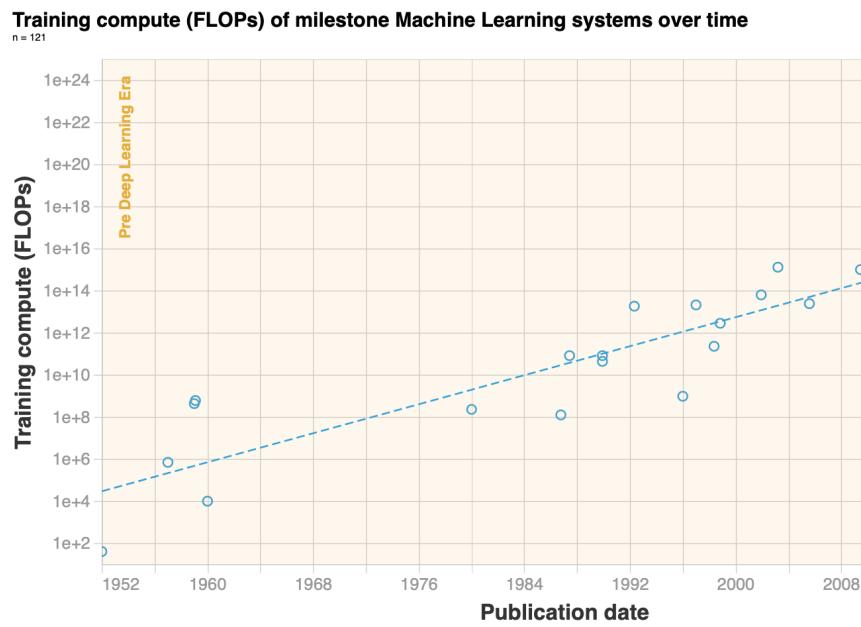










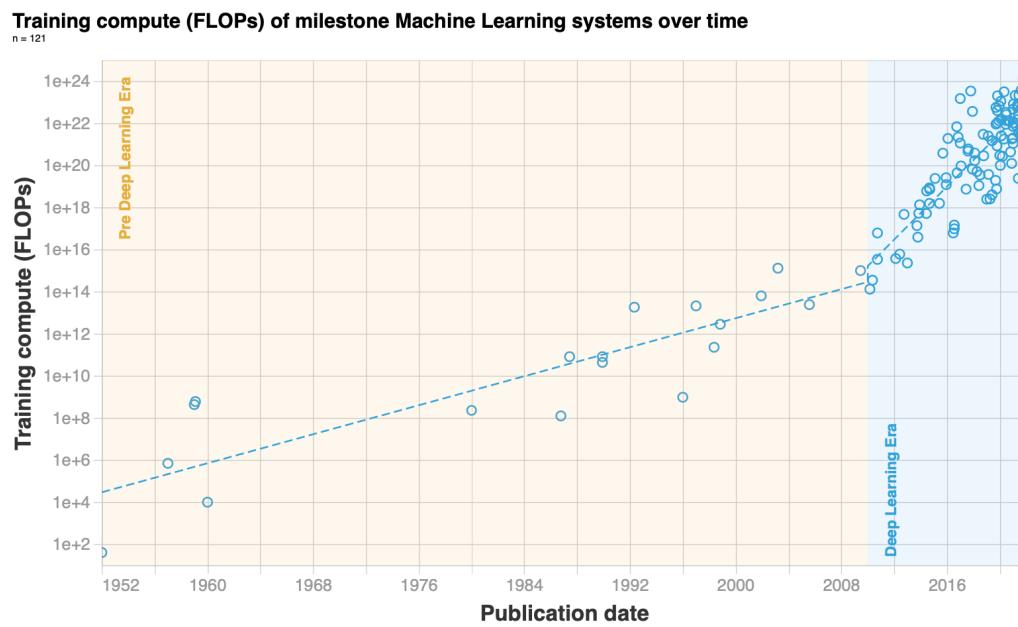


Doublement de la puissance de calcul

Loi de Moore
 $\approx 2 \text{ ans}$

Sevilla, Jaime, et al. "Compute Trends Across Three Eras of Machine Learning." *arXiv e-prints* (2022): arXiv-2202.

Thanks to Claudio Gallicchio for the slide



Doublement de la puissance de calcul

Loi de Moore
 $\approx 2 \text{ ans}$

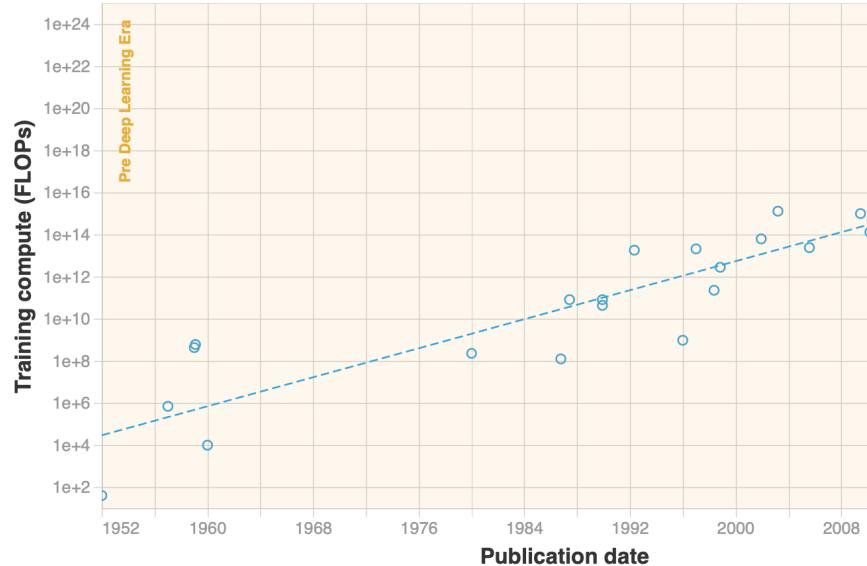
Algo. de DL
 $\approx 3 \text{ mois}$

Sevilla, Jaime, et al. "Compute Trends Across Three Eras of Machine Learning." *arXiv e-prints* (2022): arXiv-2202.

Thanks to Claudio Gallicchio for the slide

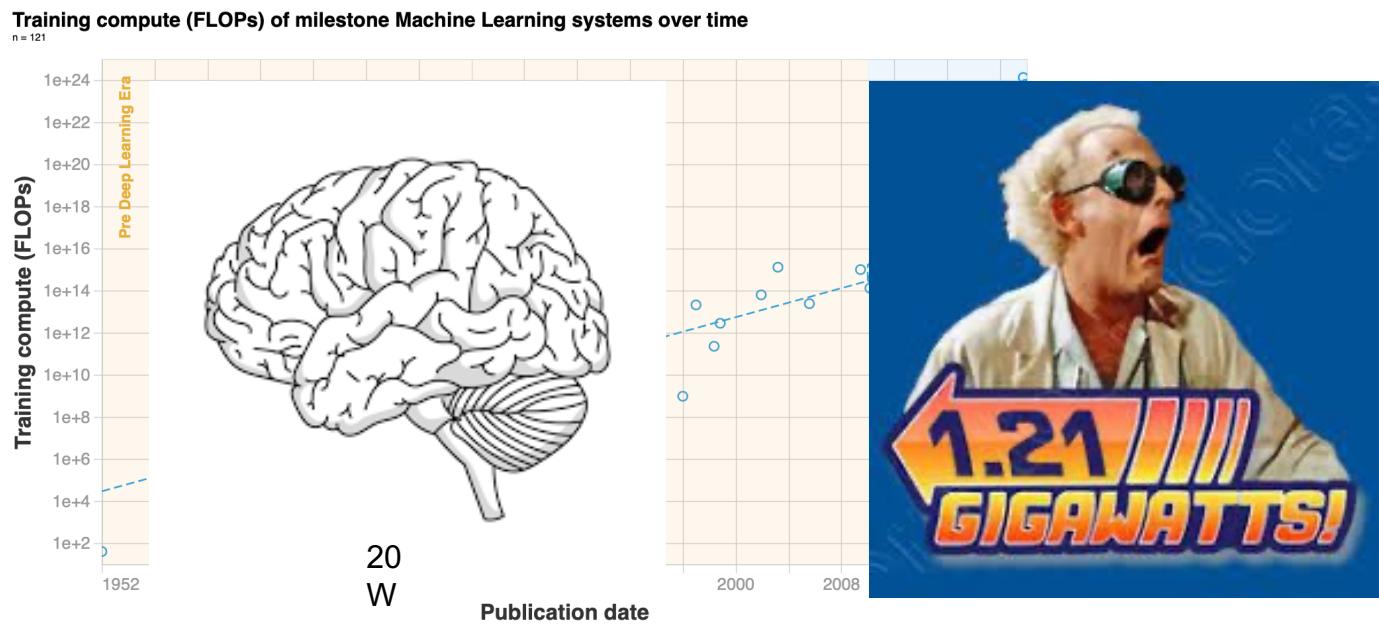
Training compute (FLOPs) of milestone Machine Learning systems over time

n = 121



Sevilla, Jaime, et al. "Compute Trends Across Three Eras of Machine Learning." *arXiv e-prints* (2022): arXiv-2202.

Thanks to Claudio Gallicchio for the slide

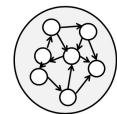


Sevilla, Jaime, et al. "Compute Trends Across Three Eras of Machine Learning." *arXiv e-prints* (2022): arXiv-2202.

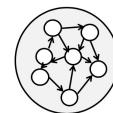
Thanks to Claudio Gallicchio for the slide

Peut-on avoir un bon équilibre entre performances et efficacité énergétique / temps de calcul ?

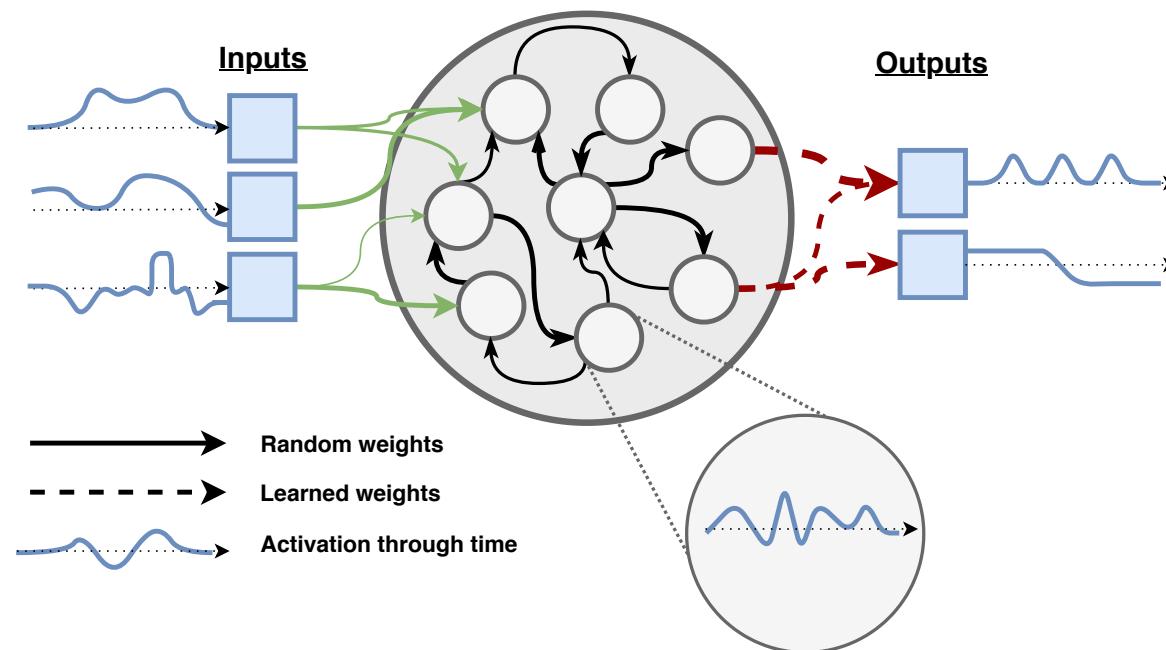
Afin d'avoir de faire des calculs plus soutenables ?
Pour modéliser le cerveau ?



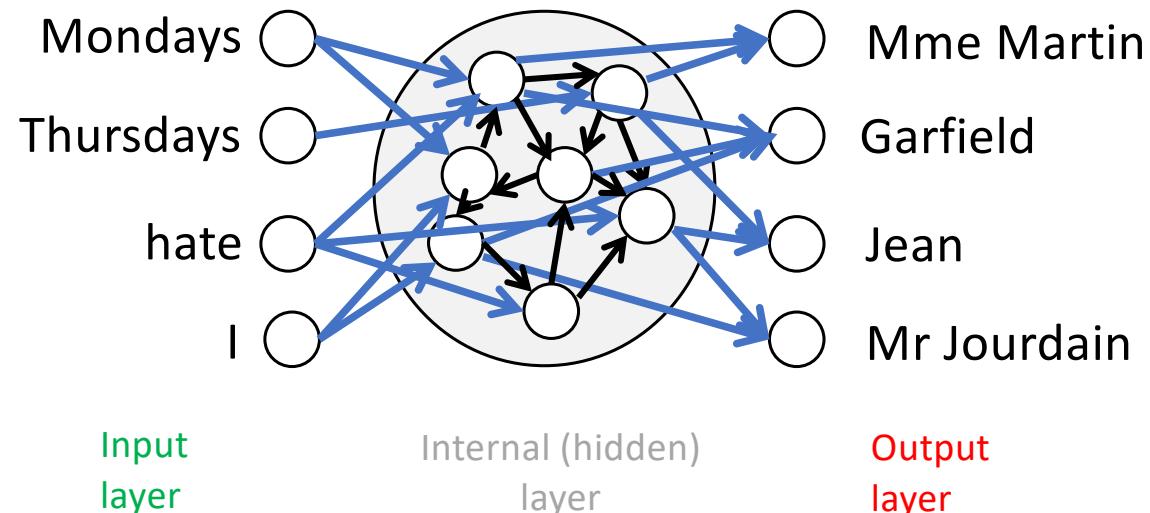
Reservoir Computing



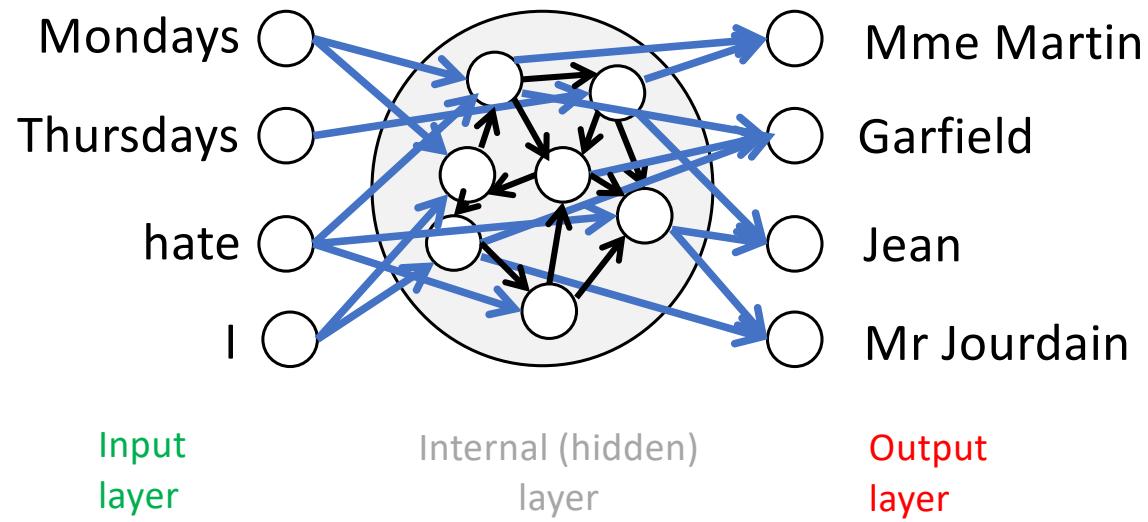
Ceci n'est pas un réseau de neurones récurrent (RNN) classique !

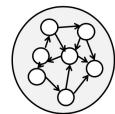


Reservoir Computing : Exploiter les dynamiques « naturelles d'un réseau récurrent »

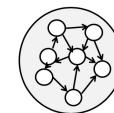


Reservoir Computing : Exploiter les dynamiques « naturelles d'un réseau récurrent »

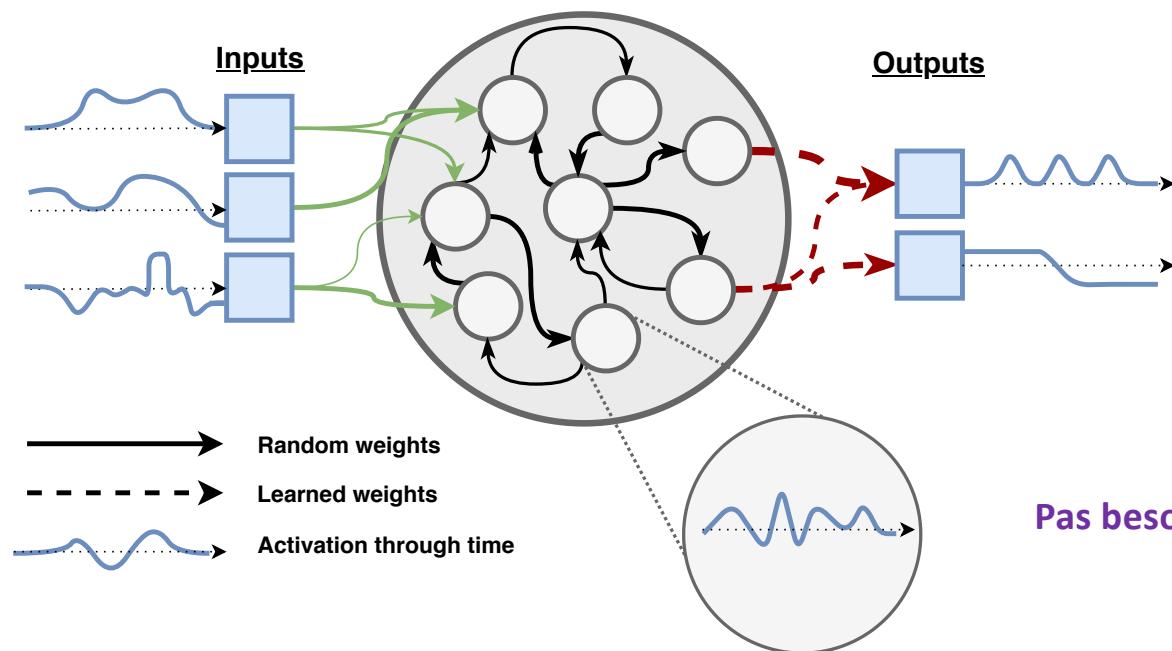




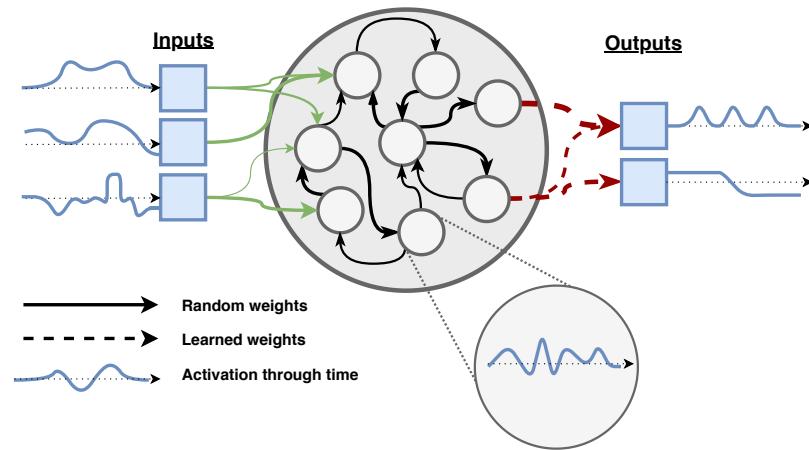
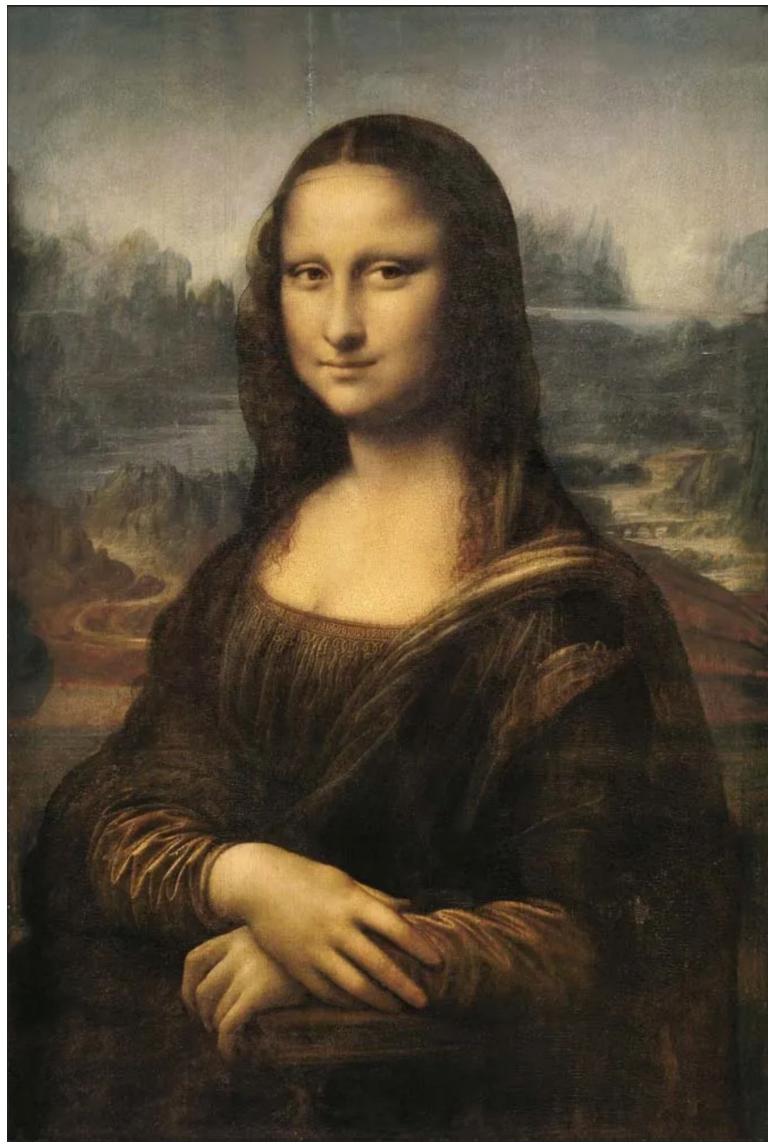
Reservoir Computing



Poids aléatoires, MAIS dynamiques déterministes !

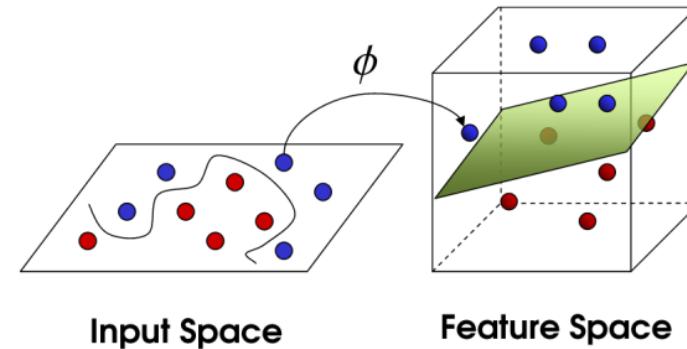
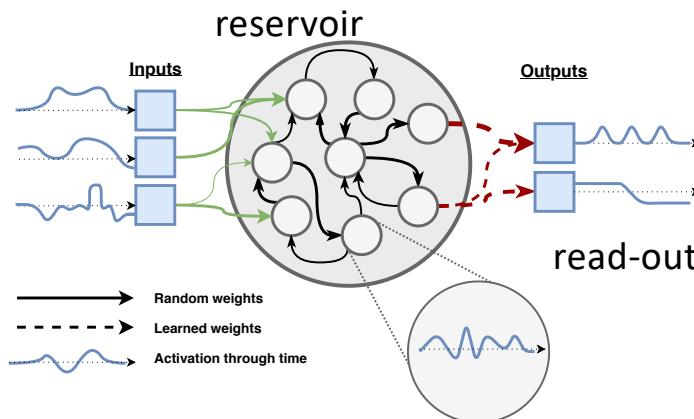


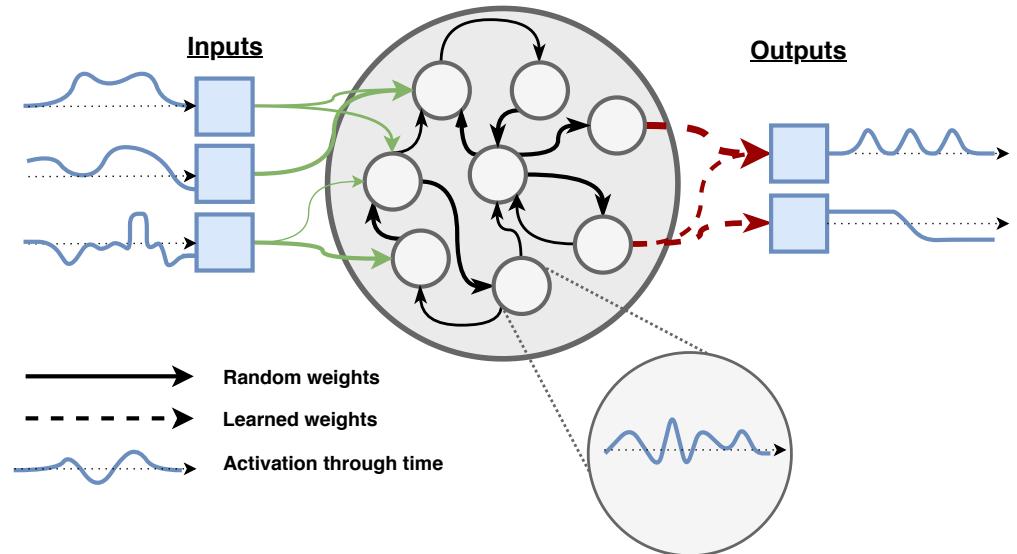
Poids aléatoires = pas d'a priori sur les connexions

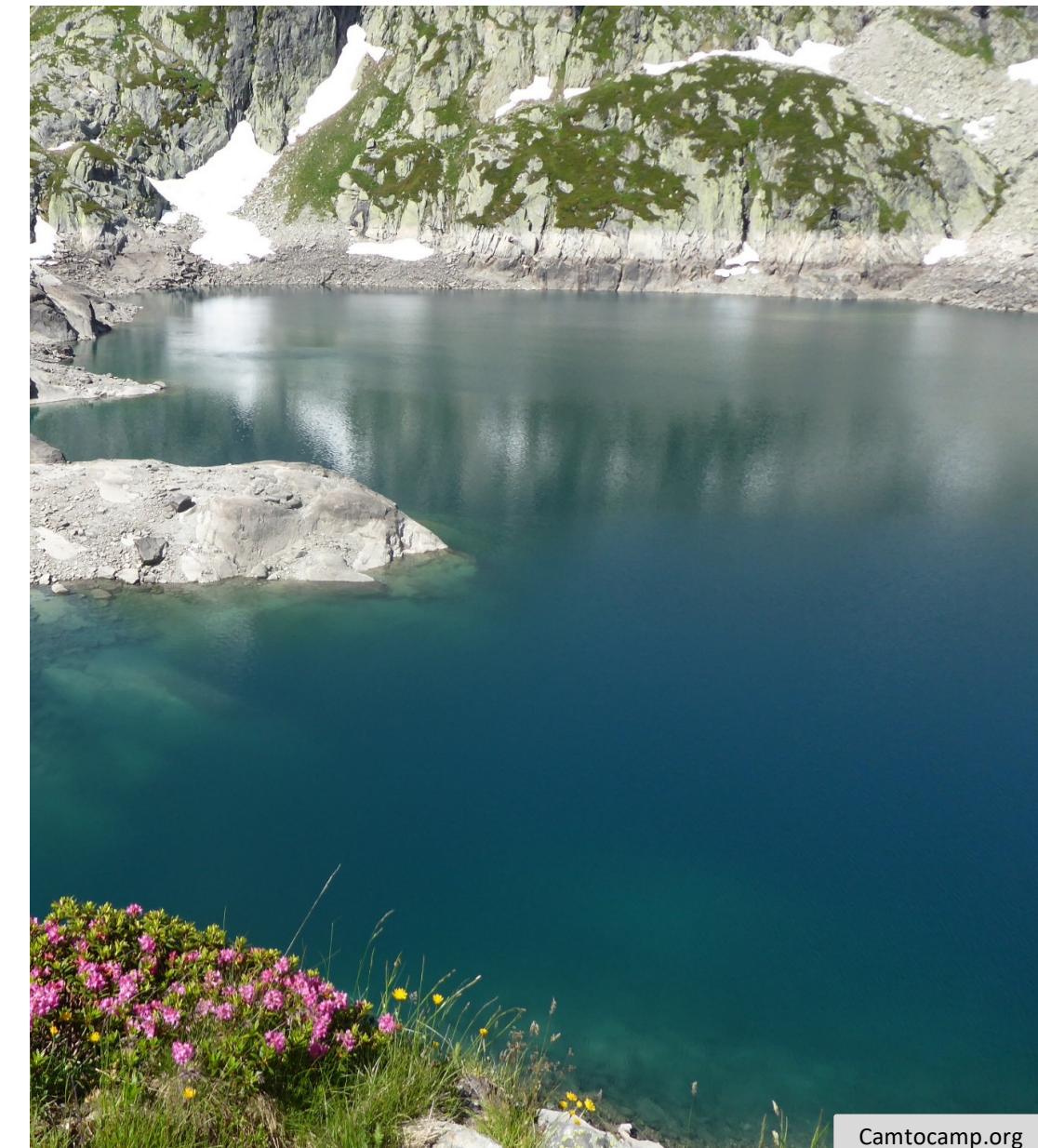


Pourquoi ce nom ?

- Les noms “**reservoir**”, pour la couche récurrente,
- et “**read-out**”, pour la couche de sortie,
- viennent du fait que beaucoup de **combinaisons** des entrées sont faites dans le reservoir (grâce aux projections aléatoires),
- le “reservoir” est donc littéralement un **réservoir de calculs** (= “reservoir computing”) non-linéaires
- dans lequel on vient “**piocher**”/décoder (= “read-out”) linéairement les combinaisons qui vont être utiles pour le problème / la tâche que l’on veut résoudre (= avoir les bonnes valeurs dans la couche de sortie).





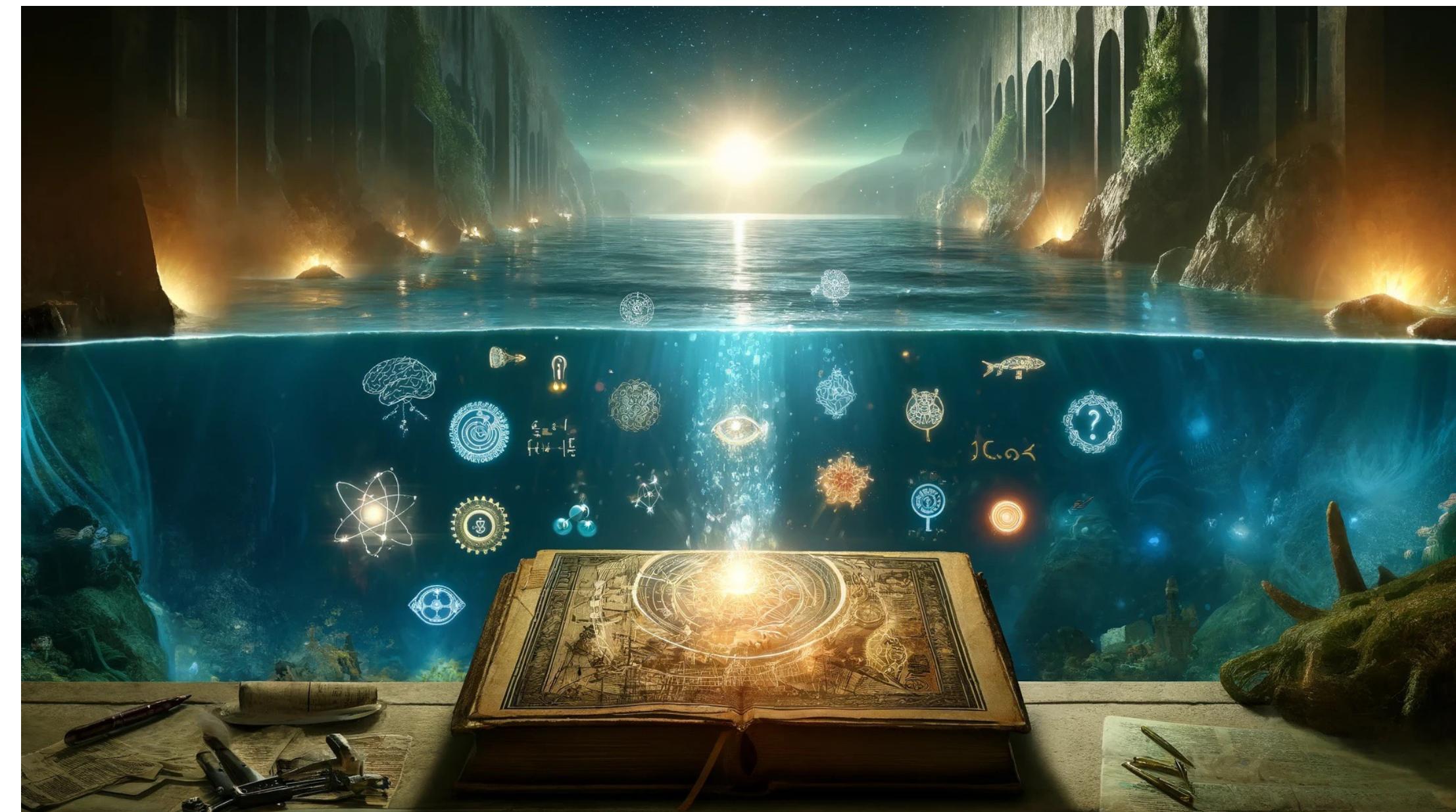


Camtocamp.org



43

Fernando & Sojakka, 2003



Pourquoi est-ce que ça marche ?



Pourquoi est-ce que ça marche ?

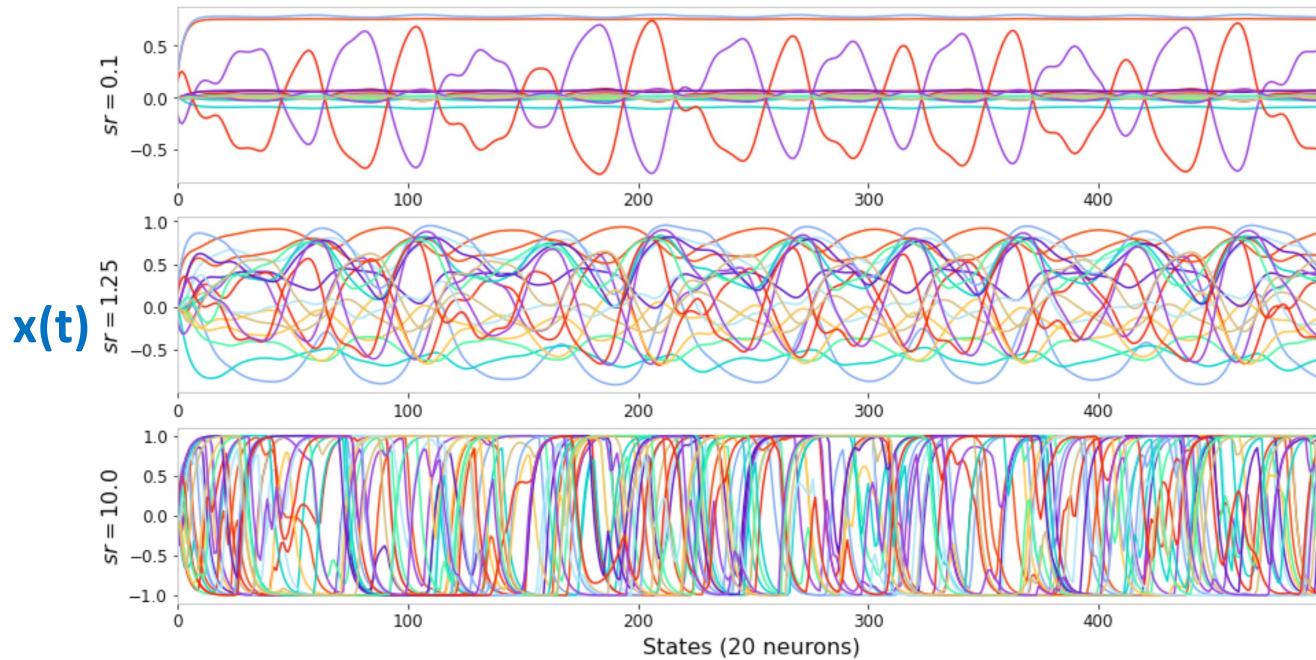


Pourquoi est-ce que ça marche ?



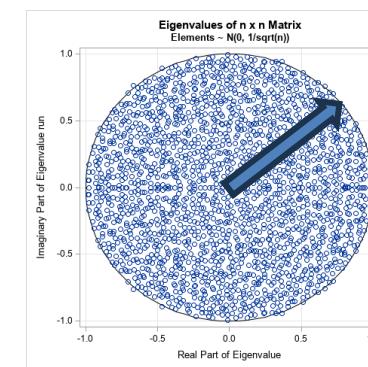
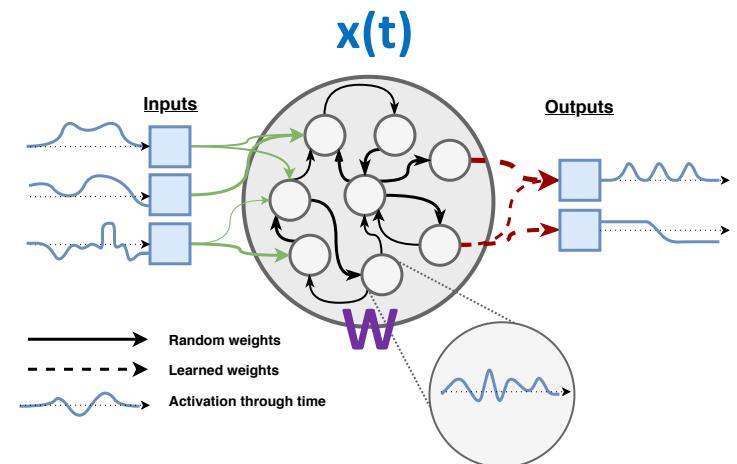
Spectral Radius (rayon spectral)

= scaling des connexions récurrentes



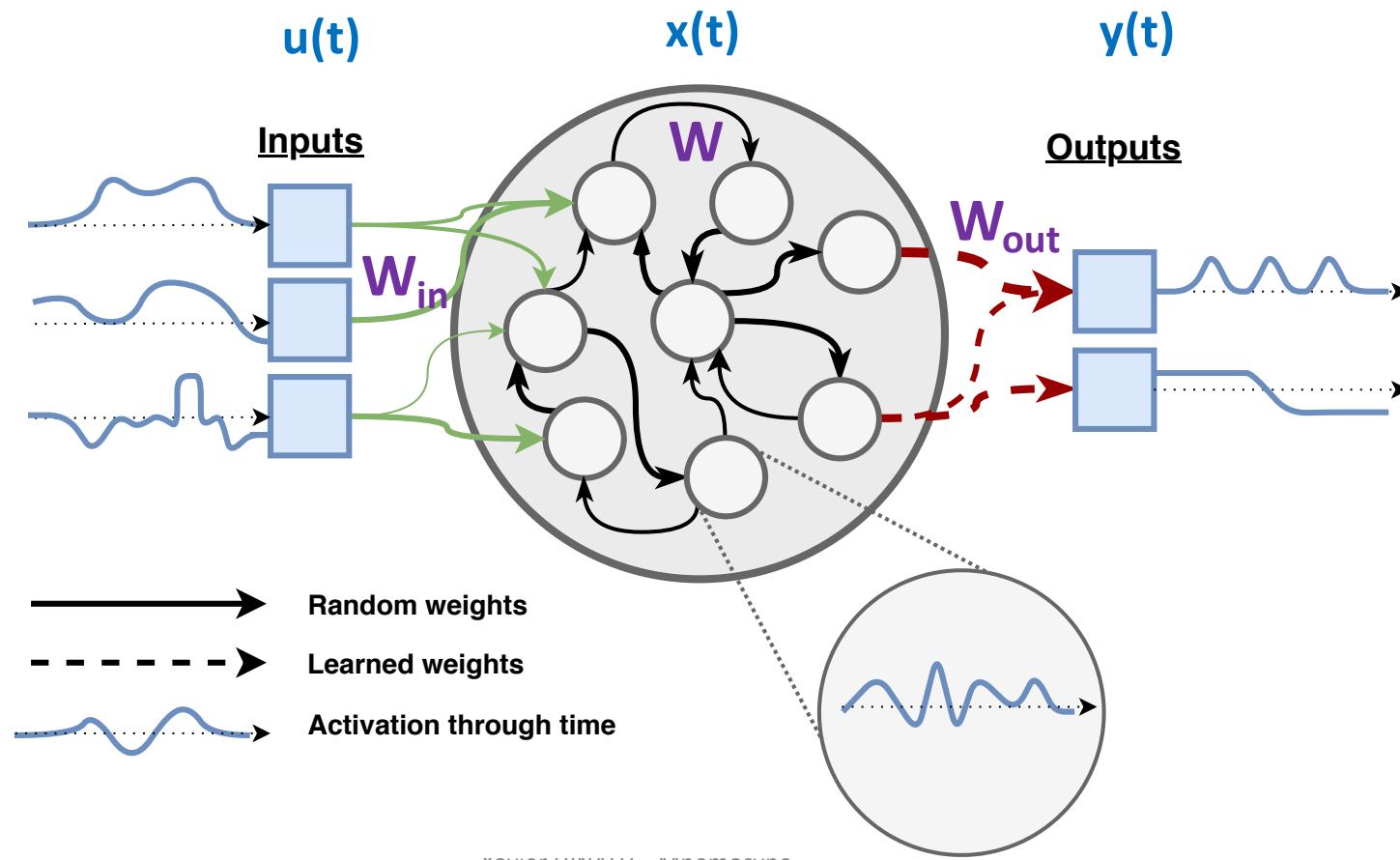
The spectral radius is defined as the maximum eigenvalue of the reservoir matrix.

- spectral radius → **stable** dynamics
- + spectral radius → **chaotic** dynamics



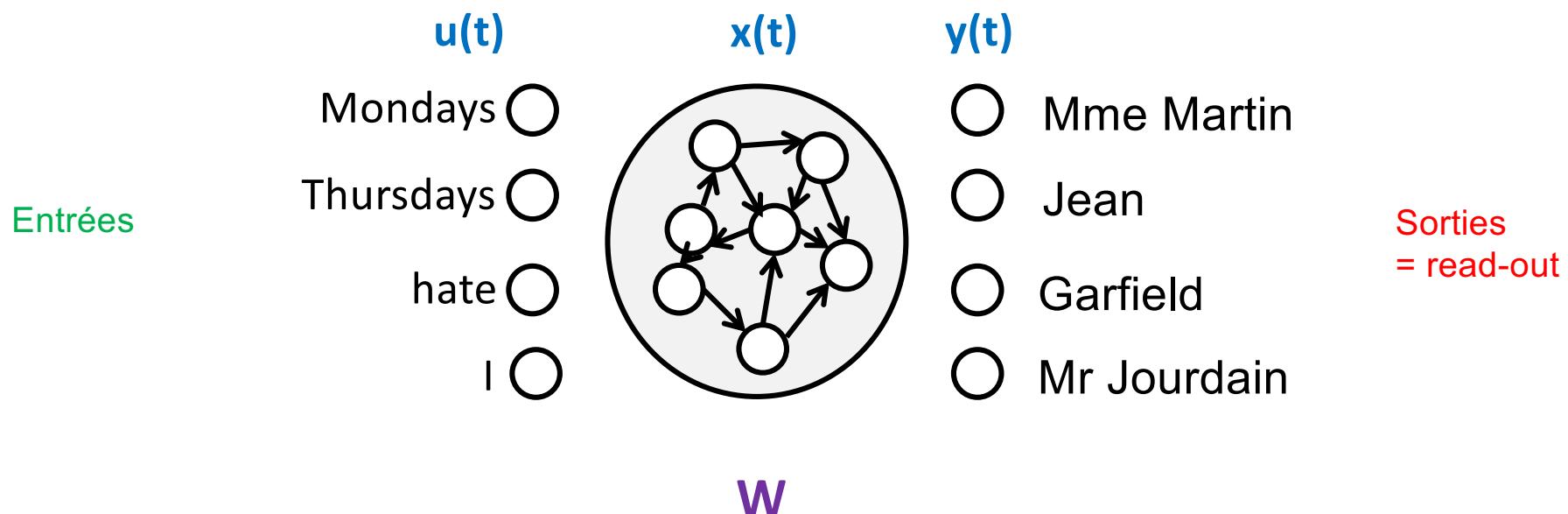
Rayon spectral de la matrice W

Reservoir Computing



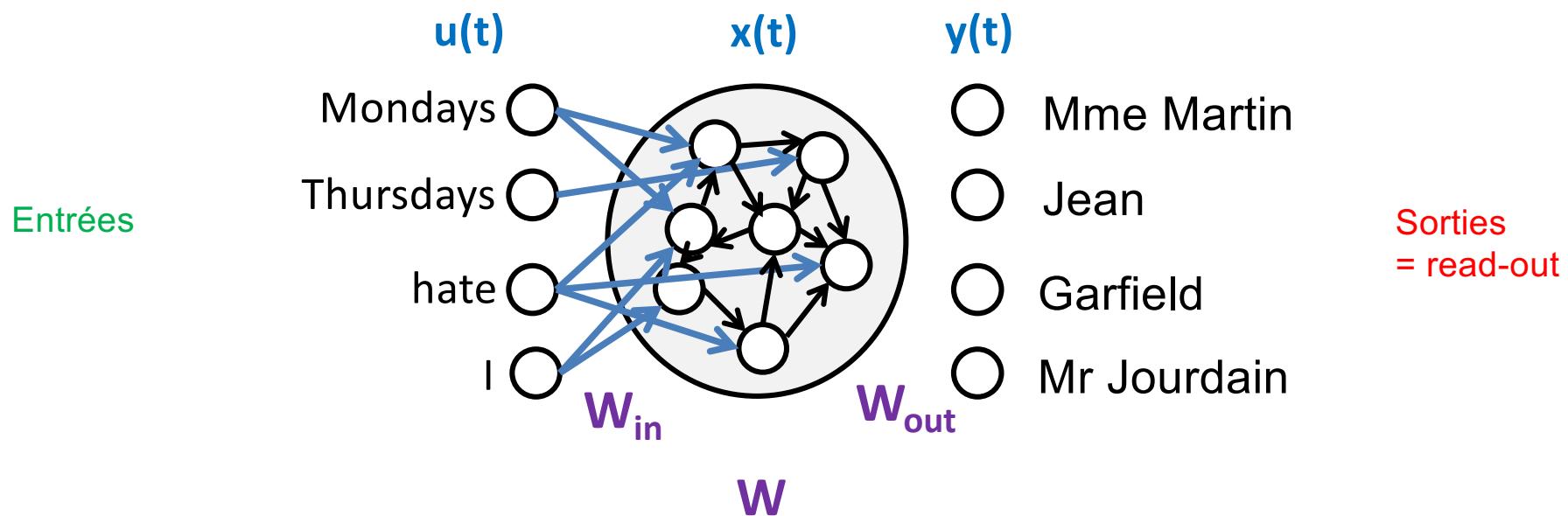
Reservoir Computing

« I hate Mondays »



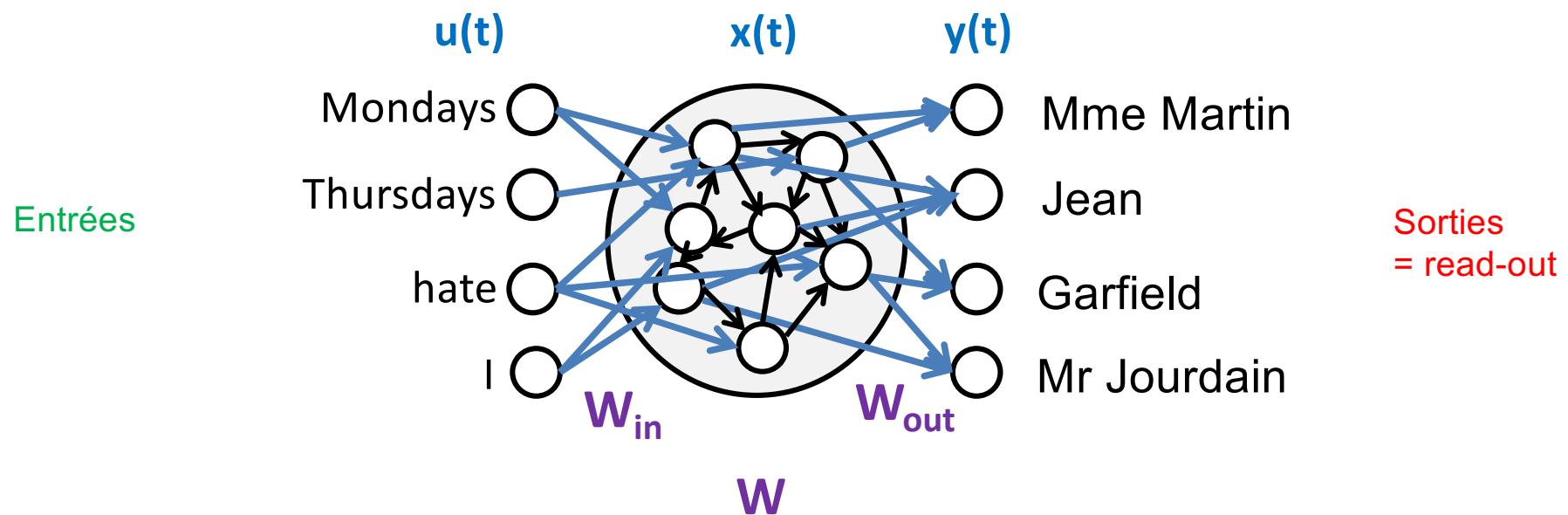
Reservoir Computing

« I hate Mondays »



Reservoir Computing

« I hate Mondays »



NE REGARDEZ PAS LA PROCHAINE DIAPO !

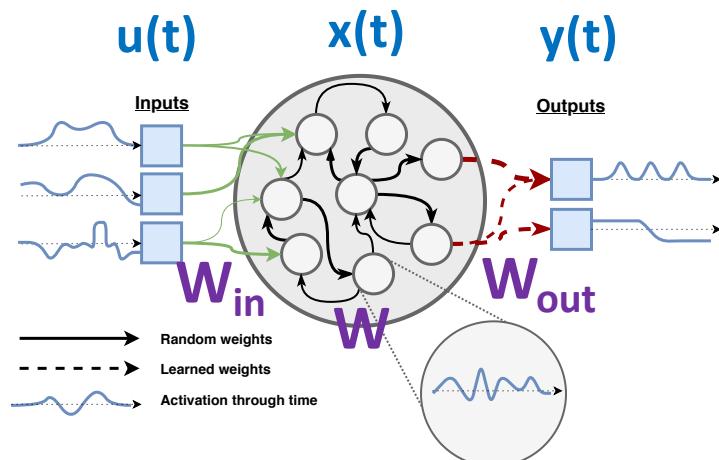
Reservoir Computing: Echo State Network (ESN)

Reservoir state update

$$x(t) = \left(1 - \frac{1}{\tau}\right)x(t-1) + \frac{1}{\tau}f(W^{in}u(t) + Wx(t-1))$$

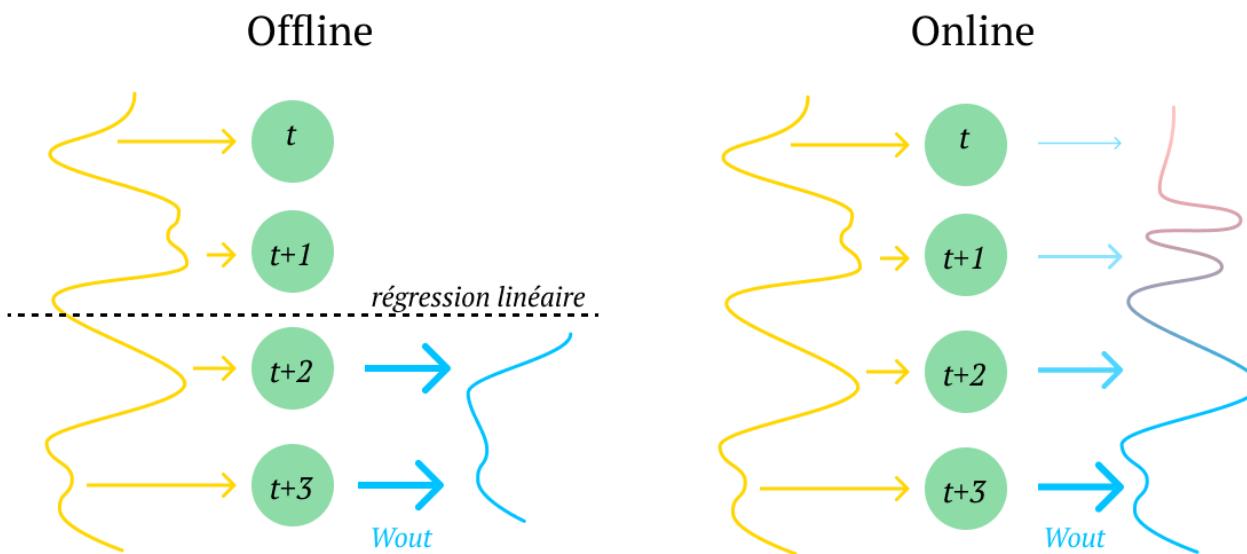
Output (« read-out ») update

$$y(t) = W^{out}x(t)$$

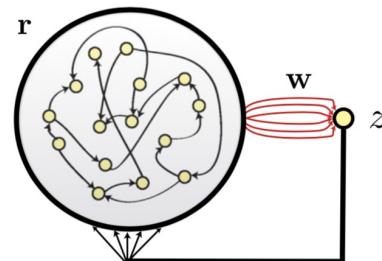


- $u(t)$: inputs
- W^{in}, W, W^{out} : input, recurrent, output matrices
- W^{in} and W matrices are kept random
- Only W^{out} is trained (e.g. ridge regression)
- τ : time constant of reservoir units
 - *leak rate (LR)* is often used instead of $\frac{1}{\tau}$
- f : activation function (usually $tanh$)

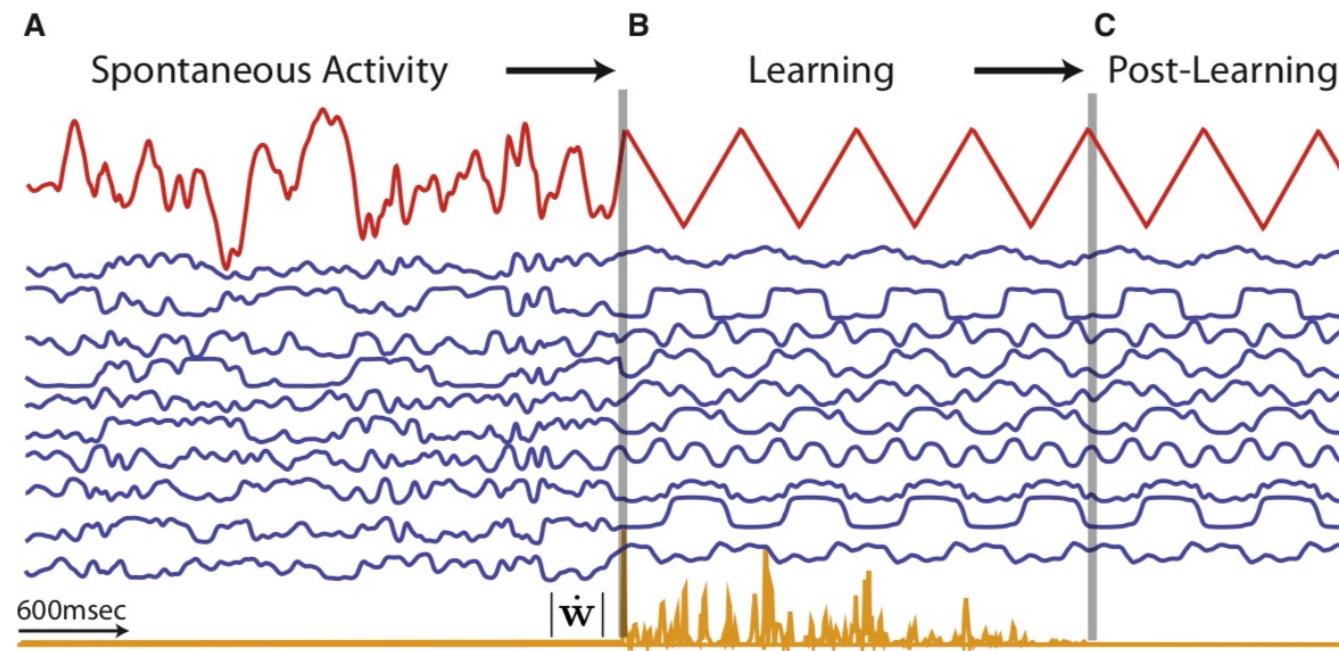
Apprentissage “hors-ligne” vs. “en-ligne”



FORCE LEARNING

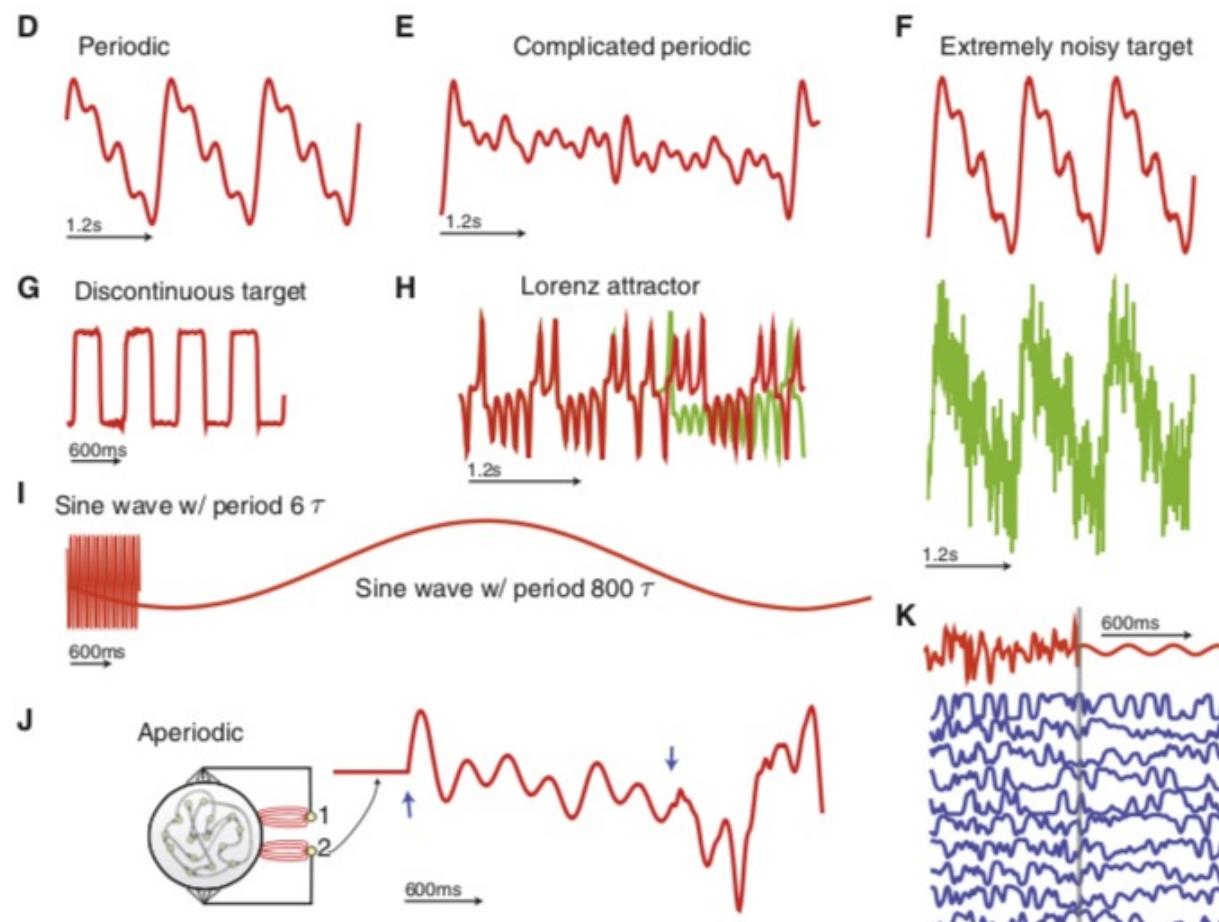


A good example of online learning
(Sussillo & Abbott, Neuron 2009)



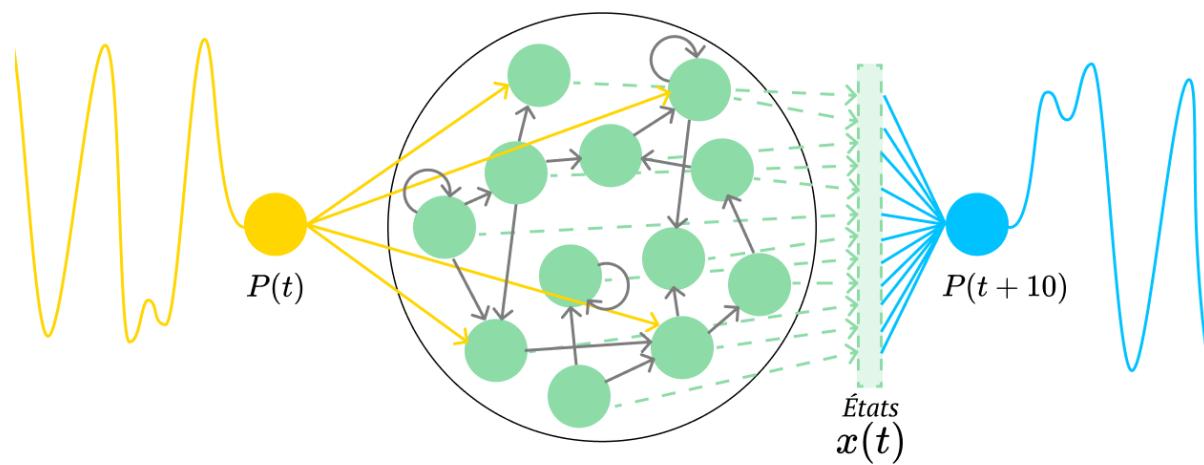
Sussillo & Abbott 2009

FORCE LEARNING



Sussillo & Abbott 2009

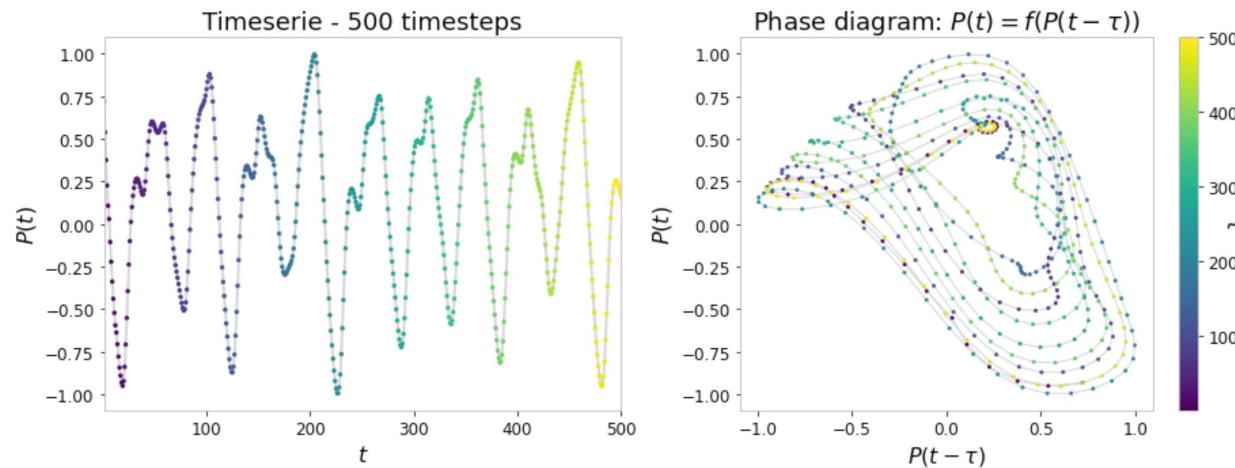
Prédiction de séries temporelles



Prédiction de séries temporelles

- Les réservoirs sont réputés pour leurs bonnes prédition de séries temporelles difficiles
 - ex: séries chaotiques (ici Mackey & Glass)

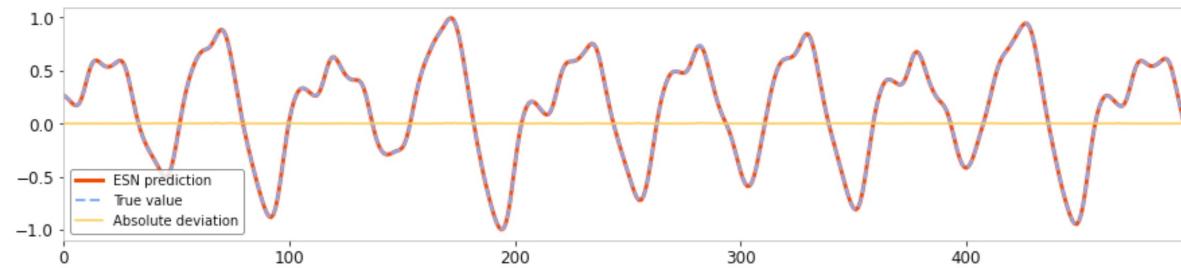
```
plot_mackey_glass(x, t, 500, tau)
```



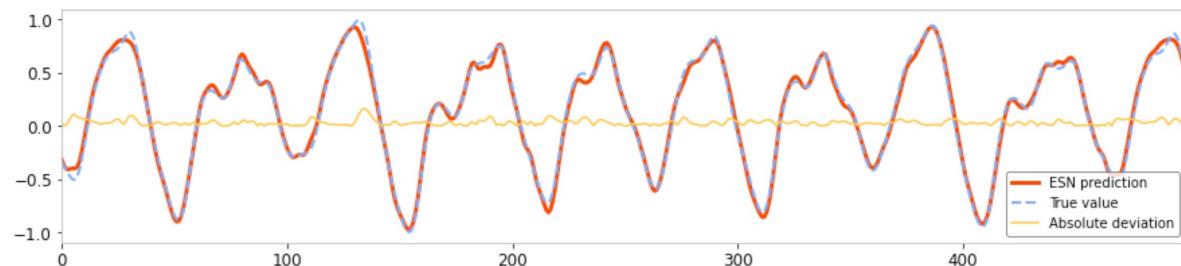
Exemple tiré de github.com/reservoirpy/reservoirpy

Prédiction de séries temporelles

- Ex : série chaotique de Mackey & Glass
 - Prédictions obtenues pour 10 pas de temps dans le futur ($T+10$)



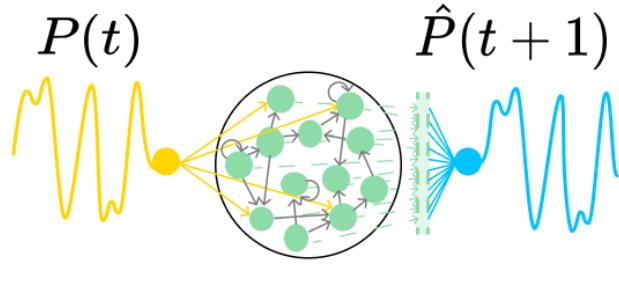
- Prédictions obtenues pour 50 pas de temps dans le futur ($T+50$)



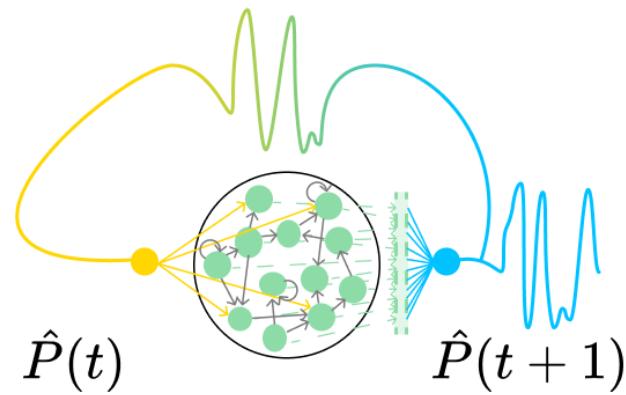
Exemple tiré de github.com/reservoirpy/reservoirpy

Génération de séries temporelles

- Génération de séquences en « faisant boucler » la sortie sur les entrées
- = Mode « Autorégressif » comme ChatGPT



Mode Prédiction



Mode Génération

RESERVOIRPY ET EXEMPLE D'APPLICATION

GÉNÉRATION DE MUSIQUE + INFLUENCE DES HYPERPARAMÈTRES



reservoirpy / reservoirpy

Code Issues Pull requests Discussions Actions Projects Wiki ...

reservoirpy Public Edit Pins Unwatch 17 Fork 110 Starred 449

v0.3.11 Go to file + <> Code About

README MIT license

python 3.8 | 3.9 | 3.10 | 3.11 | 3.12 | 3.13 pypi package 0.3.12 docs passing Testing passing codecov 93% downloads 1.9k/month downloads 57k

ReservoirPy (v0.3.12) 🤖

Simple and flexible code for Reservoir Computing architectures like Echo State Networks (ESN).

[launch binder](#)

🎉 Exciting News! We just launched a new beta tool based on a Large Language Model! 🎉 You can chat with our "ReservoirChat" and ask anything about Reservoir Computing or coding reservoirs! 🎉💡 Don't miss out, it's available for a limited time! 🚧 <https://chat.reservoirpy.inria.fr>

```
from reservoirpy.nodes import Reservoir, Ridge, Input
data = Input(input_dim=1)
reservoir = Reservoir(100, lr=0.3, sr=1.1)
readout = Ridge(ridge=1e-6)
```

banner_bw_small-size.jpg >> reservoir >> readout

About

A simple and flexible code for Reservoir Computing architectures like Echo State Networks

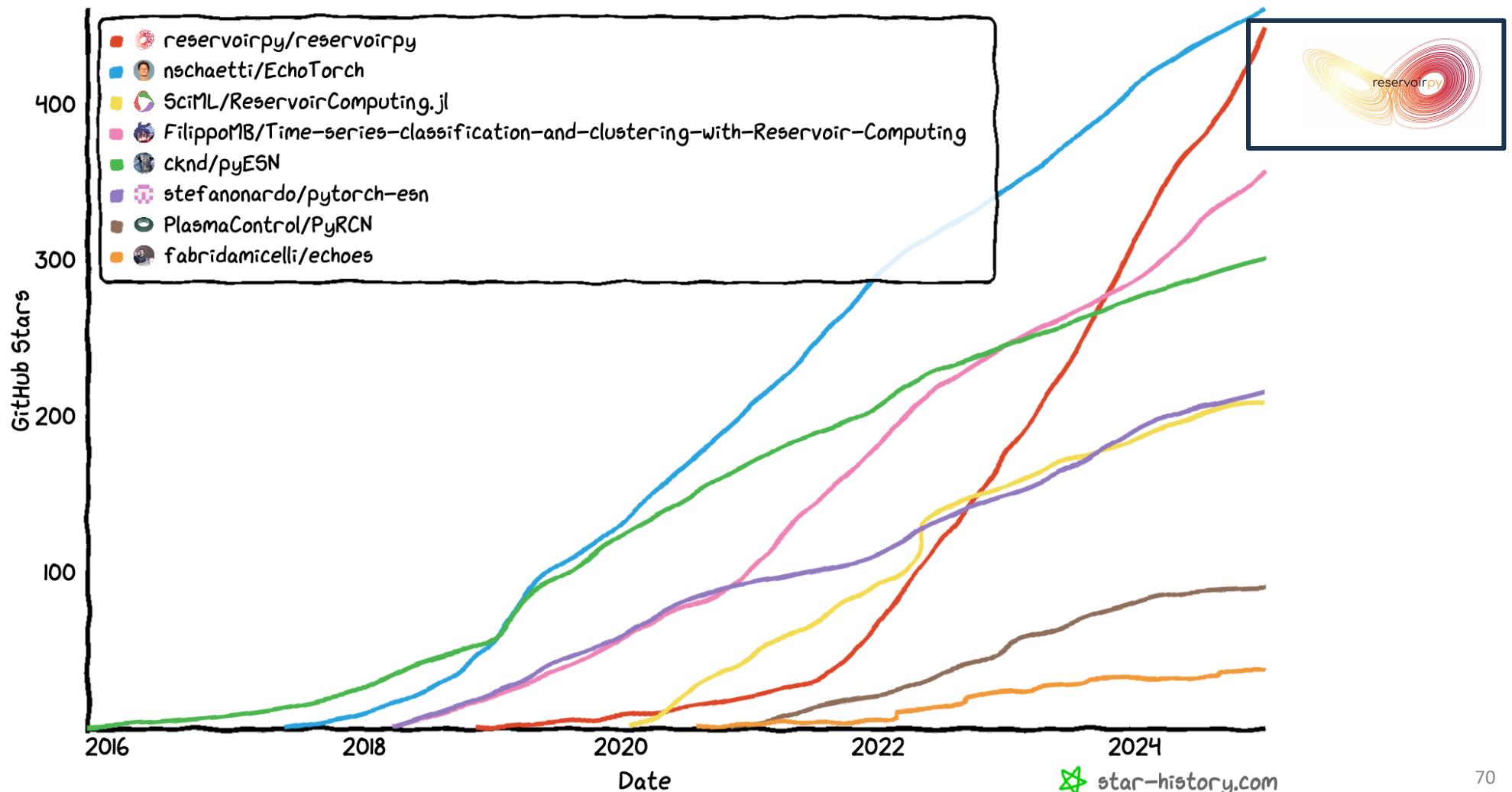
python machine-learning timeseries neural-network machine-learning-algorithms esn recurrent-neural-networks artificial-intelligence reservoir echo-state-networks

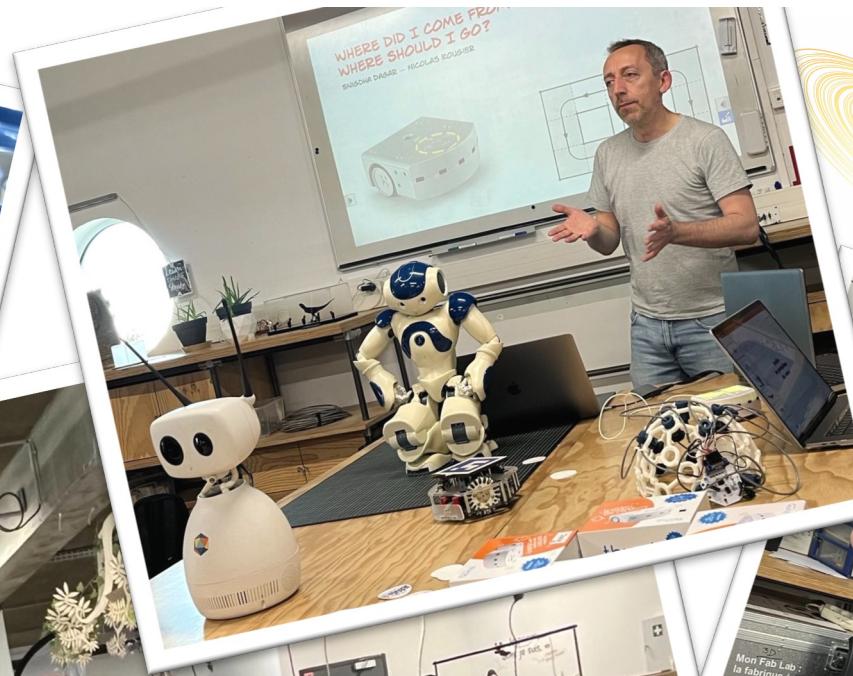
Readme MIT license Activity Custom properties 449 stars 17 watching 110 forks Report repository

Releases 26

ReservoirPy v0.3.12 Latest on Nov 12, 2024 + 25 releases

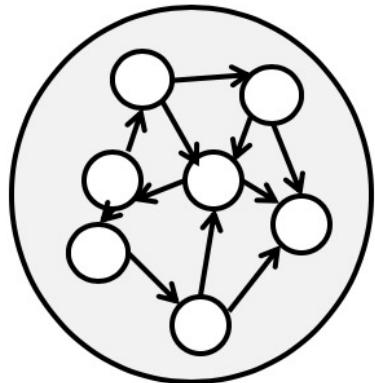
Star History

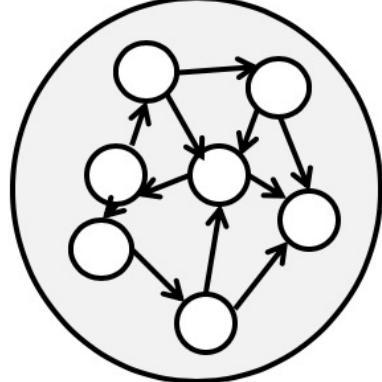




Avez-vous déjà rêvé d'être DJ ?





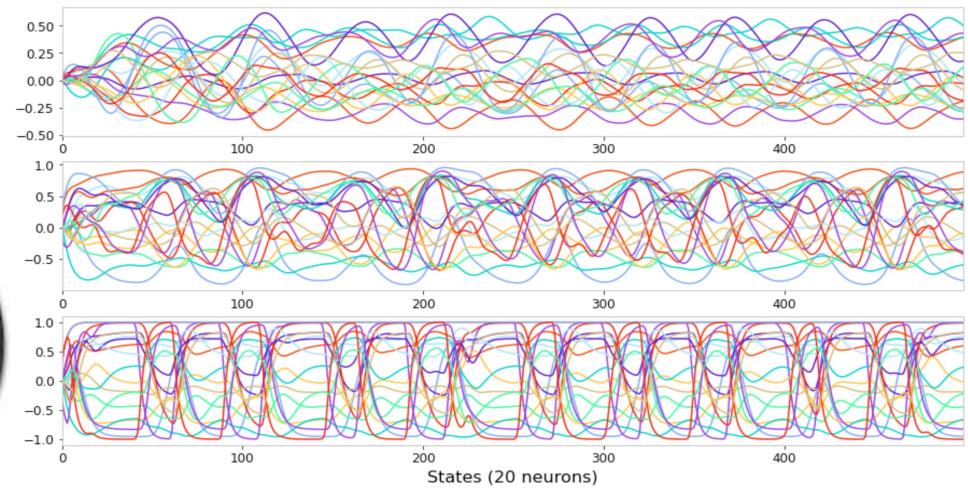


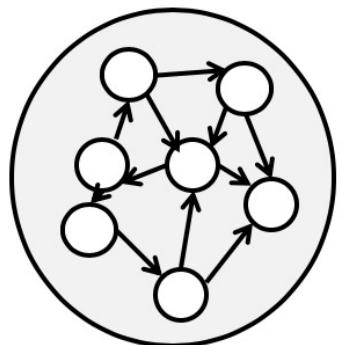
```
from reservoirpy.nodes import Reservoir, Ridge, Input

data = Input(input_dim=1)
reservoir = Reservoir(100, lr=0.3, sr=1.1)
readout = Ridge(ridge=1e-6)

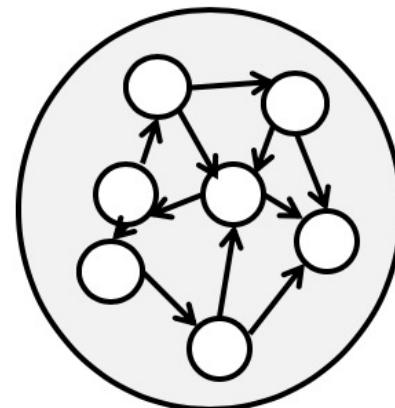
esn = data >> reservoir >> readout

forecast = esn.fit(X, y).run(timeseries)
```

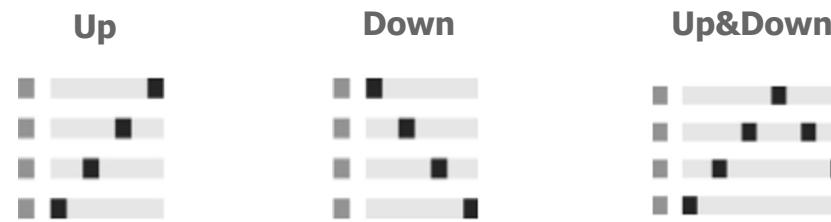




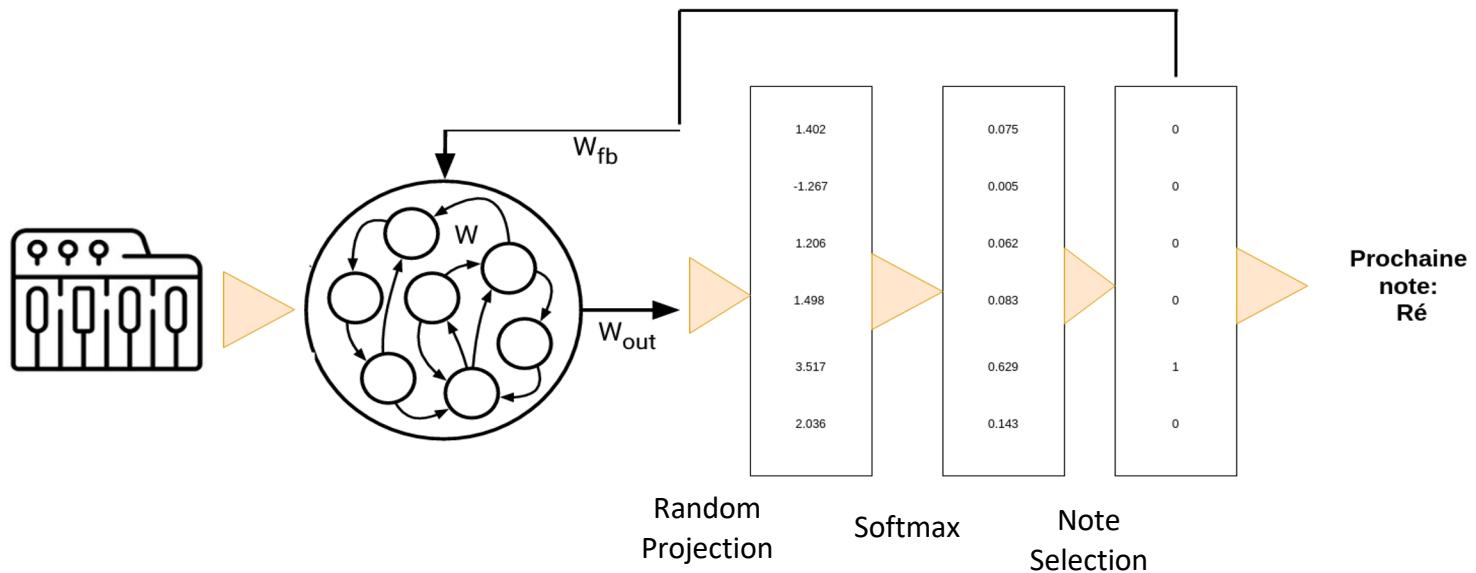
Avez-vous déjà rêvé de jouer d'un réservoir ?

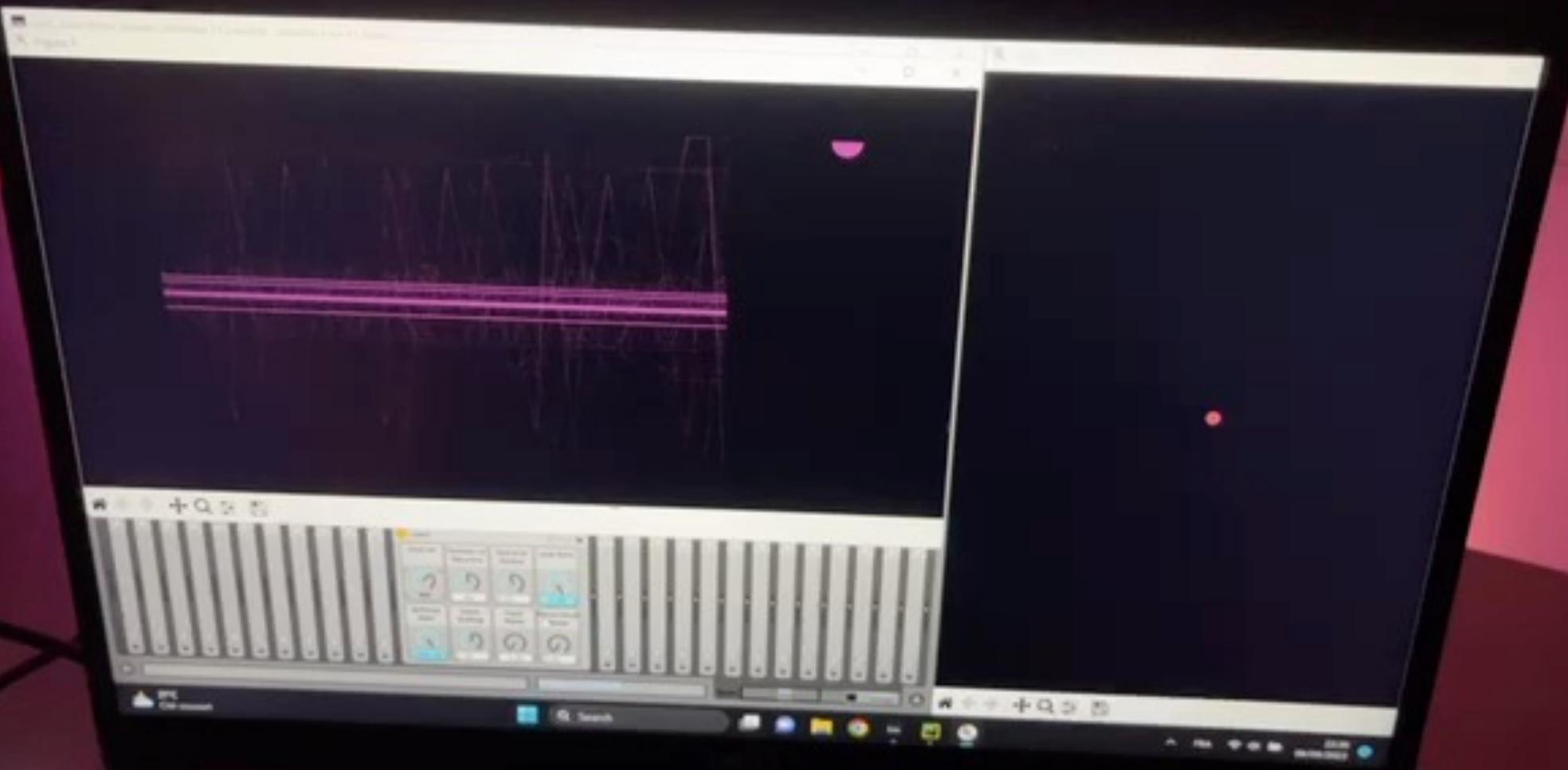


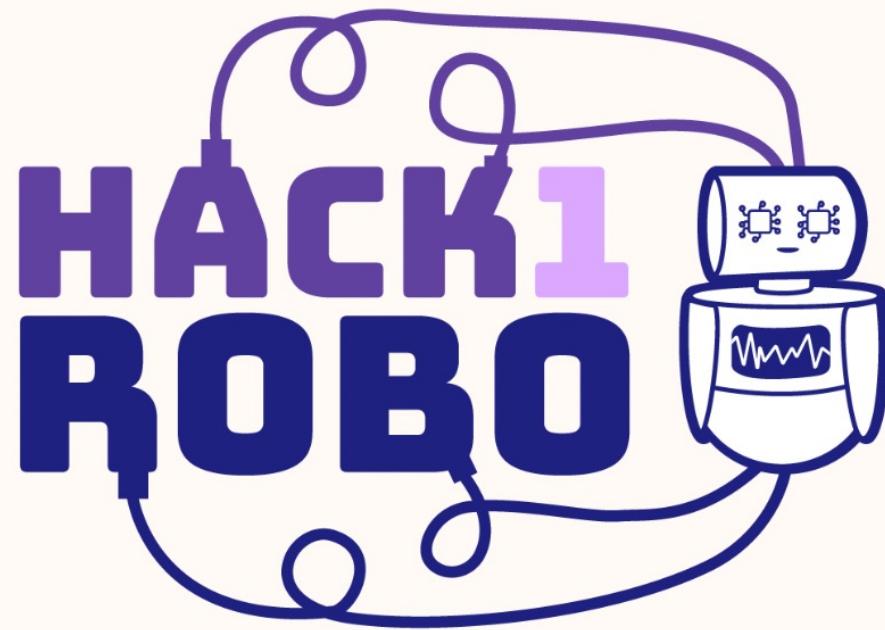
Classical arpegiators



“RéMi”: the Reservoir MIDI





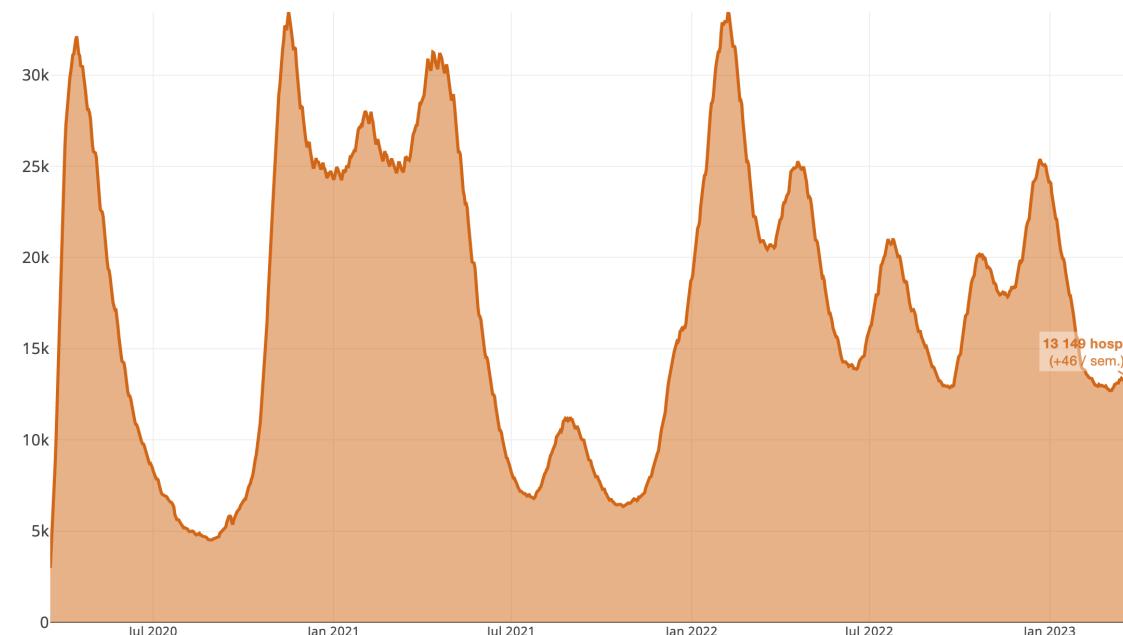


3e
~~DEUXIÈME~~ HACKATHON
EN IA & ROBOTIQUE
DU 9 AU 11 FEVRIER 2024
Novembre



Prédiction hospitalisation COVID à t+14

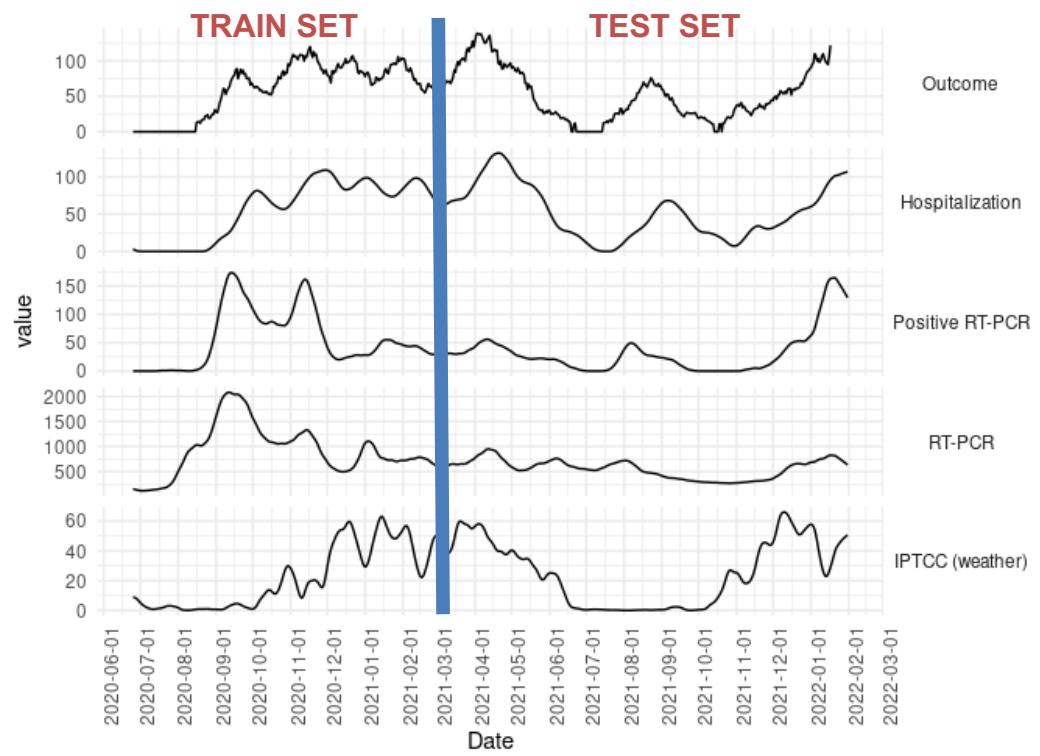
Nombre de personnes hospitalisées
Nombre de personnes hospitalisées avec Covid19.



CovidTracker.fr • Données : Santé publique France • Dernière donnée : 31 / 03 / 2023

Méthodes

- **Données** : Données agrégées provenant des dossiers médicaux électroniques + SARS-CoV-2 publics + météo = 409 *features* sur 586 jours.
- **Haute dimension** : 409 *features* \simeq nombre de variables observées chaque jour.
- **Résultat** : Hospitalisations à 14 jours au CHU de Bordeaux.
- **Métrique** : Erreur Absolue Moyenne (MAE)
- **Modèle** : Reservoir Computing (médiane de prédictions de 40 réservoirs)



Reservoir Computing + Algorithme Génétique

4 Reservoir Computing Hyperparameters (RC-HP) :

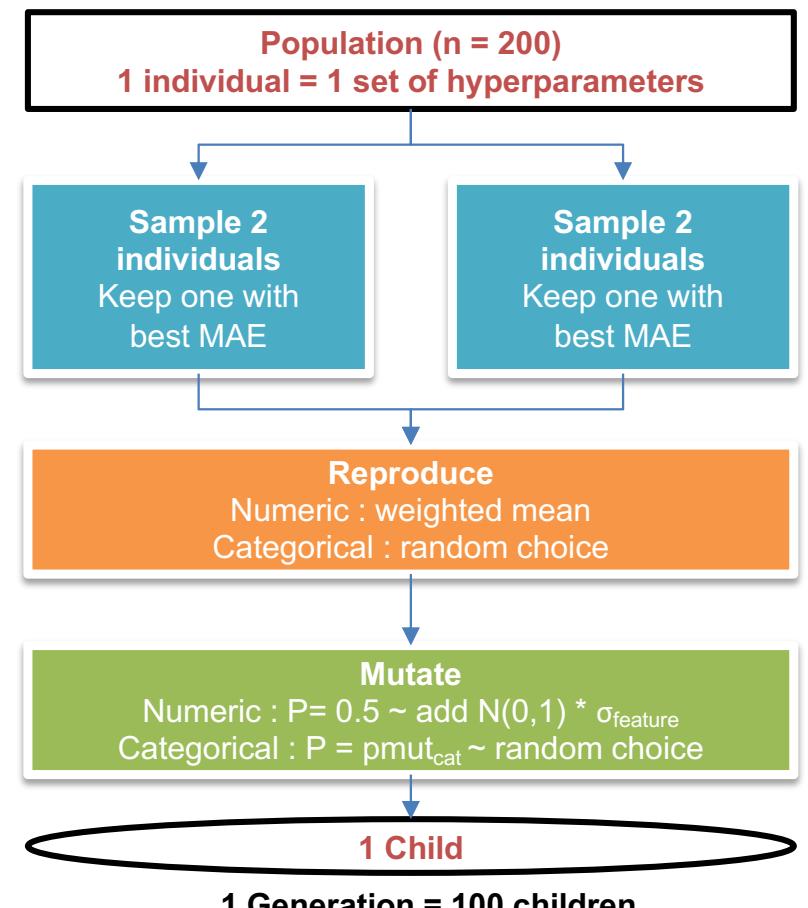
- leaking rate = inertia
- spectral radius = chaoticity
- input scaling = gain of input features
- ridge output layer = limit overfitting.

409 Binary indicator for each feature selection

- Selected/not selected for each feature

Example of 1 individual :

- Leaking rate = 0,5 ; spectral_radius = 1e-2 ;
input_scaling = 1e-3 ; ridge = 1e2 ; hospitalization =
« selected », IPTCC = « not selected » ; RT-PCR 60
yo+ = « selected » ; ...
- Fitness = MAE on training period (median forecast of
3 RC)



Total = 30 Generations (+ 10 generations each month)

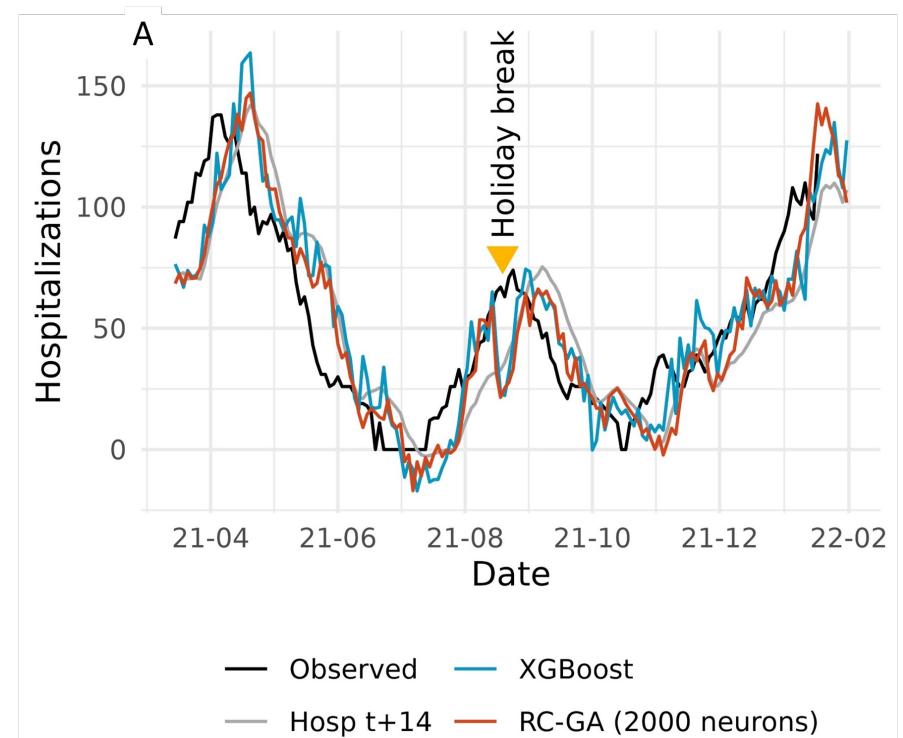
Prédiction hospitalisation COVID à t+14

Model	Mean Absolute Error (\pm std)
RC-GA (500 neurons)	15.27 (\pm 12.88)
RC-GA (2000 neurons)	14.66 (\pm 12.63)
RC-RS (500 neurons)	18.45 (\pm 13.27)
RC-GA-no Feature selection (500 neurons)	17.59 (\pm 13.35)
Elastic-net	15.83 (\pm 12.41)
XGBoost	15.45 (\pm 13.91)
LSTM-PCA	15.74 (\pm 13.98)
Transformers-PCA	19.28 (\pm 16.80)
Informer-PCA	18.70 (\pm 14.11)
PatchTST-PCA	18.34 (\pm 18.14)
Prophet	21.30 (\pm 16.33)

Notre approche

Prédiction hospitalisation COVID à t+14

Model	Mean Absolute Error (\pm std)
RC-GA (500 neurons)	15.27 (\pm 12.88)
RC-GA (2000 neurons)	14.66 (\pm 12.63)
RC-RS (500 neurons)	18.45 (\pm 13.27)
RC-GA-no Feature selection (500 neurons)	17.59 (\pm 13.35)
Elastic-net	15.83 (\pm 12.41)
XGBoost	15.45 (\pm 13.91)
LSTM-PCA	15.74 (\pm 13.98)
Transformers-PCA	19.28 (\pm 16.80)
Informer-PCA	18.70 (\pm 14.11)
PatchTST-PCA	18.34 (\pm 18.14)
Prophet	21.30 (\pm 16.33)



Prédiction hospitalisation COVID à t+14

- Algorithmes génétiques avec 400 variables d'entrées
 - Ferte, ..., Hinaut, ICML 2024

Reservoir Computing for Short High-Dimensional Time Series: an Application to SARS-CoV-2 Hospitalization Forecast

Thomas Ferté, Dan Dutartre, Boris P Hejblum, Romain Griffier, Vianney Jouhet, Rodolphe Thiébaut, Pierrick Legrand, Xavier Hinaut

Published: 02 May 2024, Last Modified: 25 Jun 2024 | ICML 2024 Poster | Everyone | Revisions | BibTeX
CC BY-NC 4.0

Abstract:
In this work, we aimed at forecasting the number of SARS-CoV-2 hospitalized patients at 14 days to help anticipate the bed requirements of a large scale hospital using public data and electronic health records data. Previous attempts led to mitigated performance in this high-dimension setting; we introduce a novel approach to time series forecasting by providing an alternative to conventional methods to deal with high number of potential features of interest (409 predictors). We integrate Reservoir Computing (RC) with feature selection using a genetic algorithm (GA) to gather optimal non-linear combinations of inputs to improve prediction in sample-efficient context. We illustrate that the RC-GA combination exhibits excellent performance in forecasting SARS-CoV-2 hospitalizations. This approach outperformed the use of RC alone and other conventional methods: LSTM, Transformers, Elastic-Net, XGBoost. Notably, this work marks the pioneering use of RC (along with GA) in the realm of short and high-dimensional time series, positioning it as a competitive and innovative approach in comparison to standard methods.

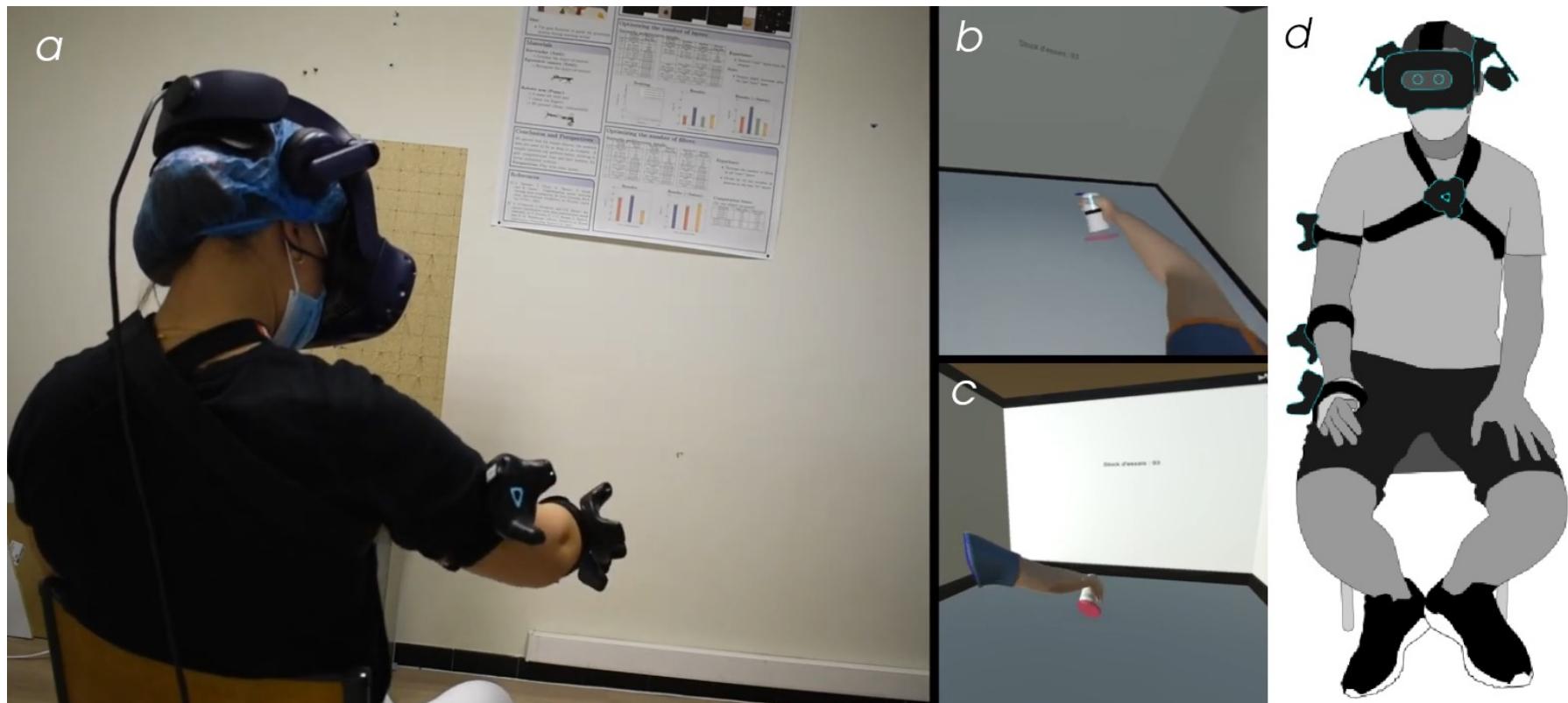
<https://openreview.net/forum?id=CY0lFwD4qx>

90

EXEMPLE D'APPLICATION

CONTROLE DE PROTHÈSE BIOMIMÉTIQUE

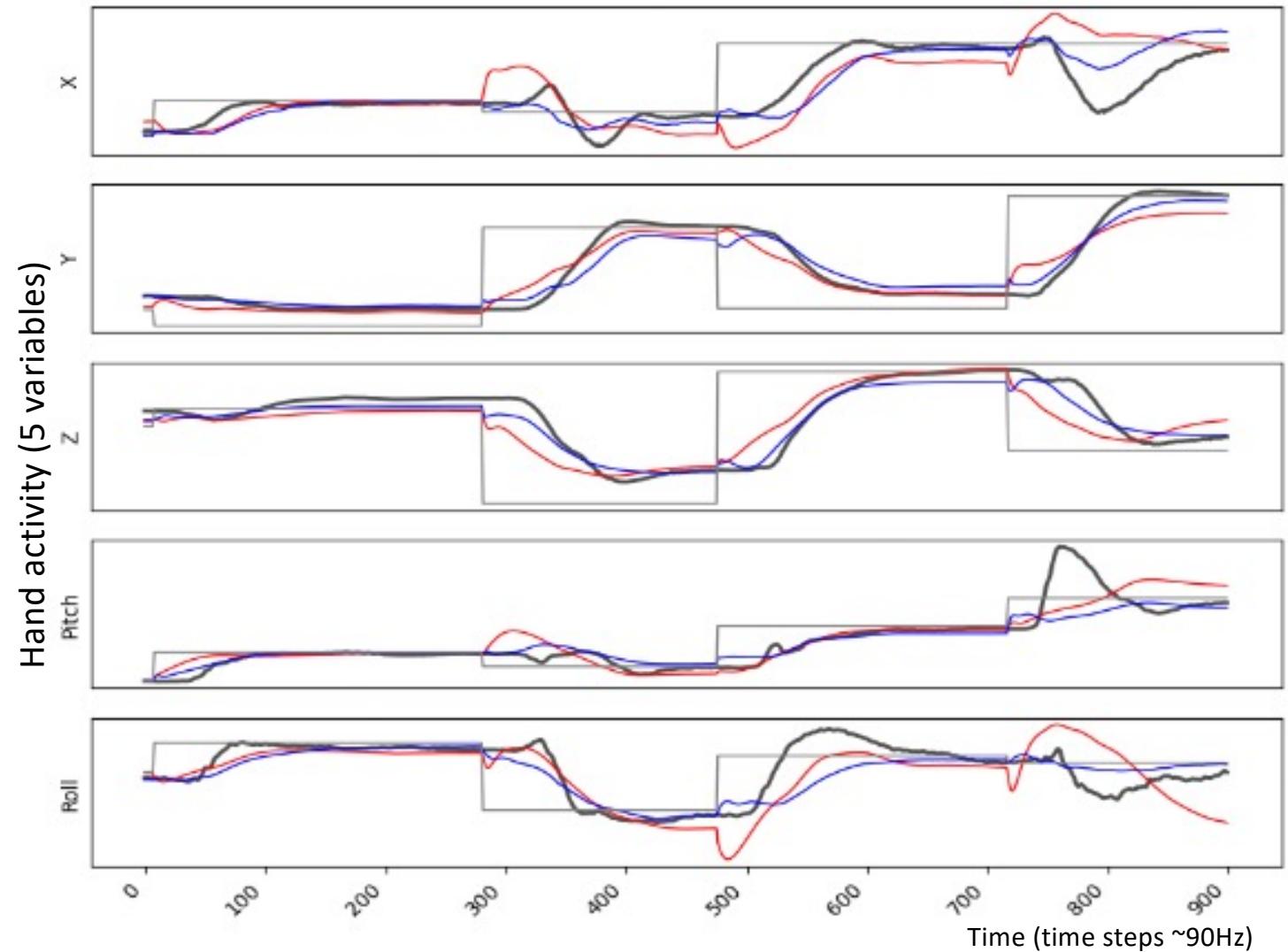
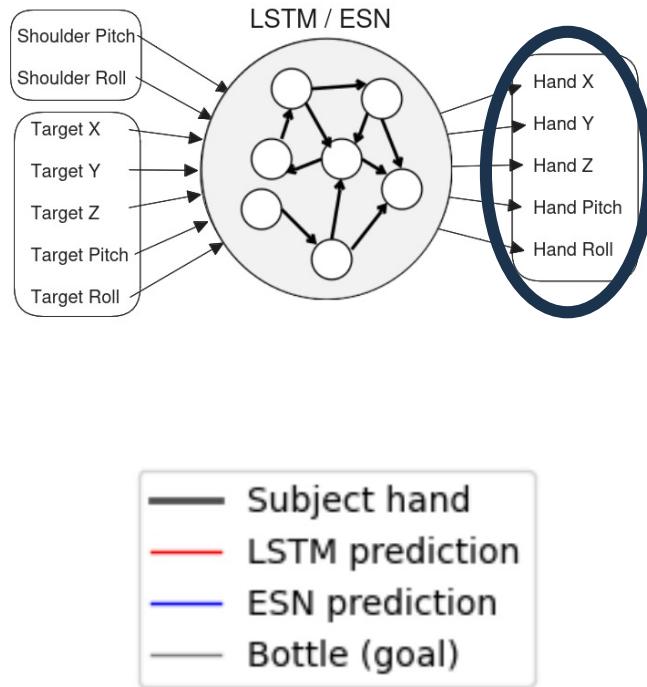
Controle de prothèse biomimétique



Segas, 2023

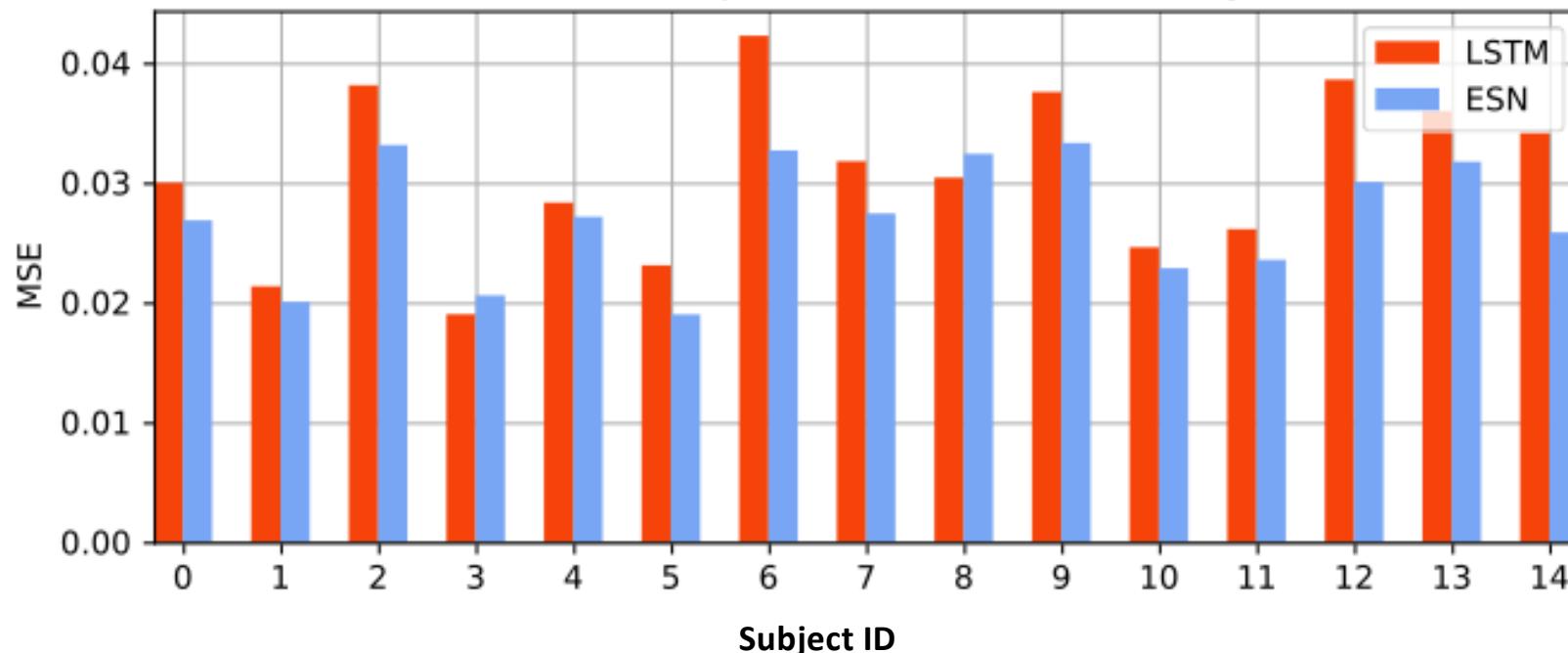


Résultats de l'apprentissage sur un seul sujet

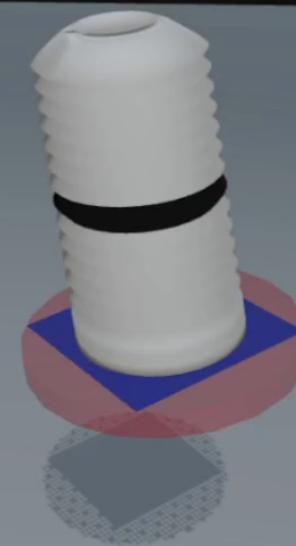


Résultats de l'apprentissage sur un seul sujet

Erreurs du reservoir (ESN) et LSTM pour chaque sujet



Stock d'essais: 49



En pratique ? (Hypster-paramètres)



En pratique ? (Hypster-paramètres)

- Les **hyperparamètres** importants à régler
 - nombre de neurones
 - Input scaling
 - Spectral radius (= rayon spectral)
 - (Feedback scaling si présence de feedback)
 - Leak-rate (= taux de fuite = 1 / constance de temps)
 - Ridge (régularisation de la régression)
 - Warm-up (= « temps de chauffe »)

→ Voir le tutoriel 4 de ReservoirPy sur GitHub

[reservoirpy / tutorials / 4-Understand_and_optimize_hyperparameters.ipynb](#)

VirgileBoraud and PAUL-BERNARD Correction of some minor mistakes on the tutorials (#161) 78b89al

2.29 MB

Understand and optimize ESN hyperparameters

This chapter gives very basic clues on how to interpret the action of some first importance hyperparameters of an networks.

It also present a very basic example of optimization using `hyperopt` and `reservoirpy.hyper` tools.

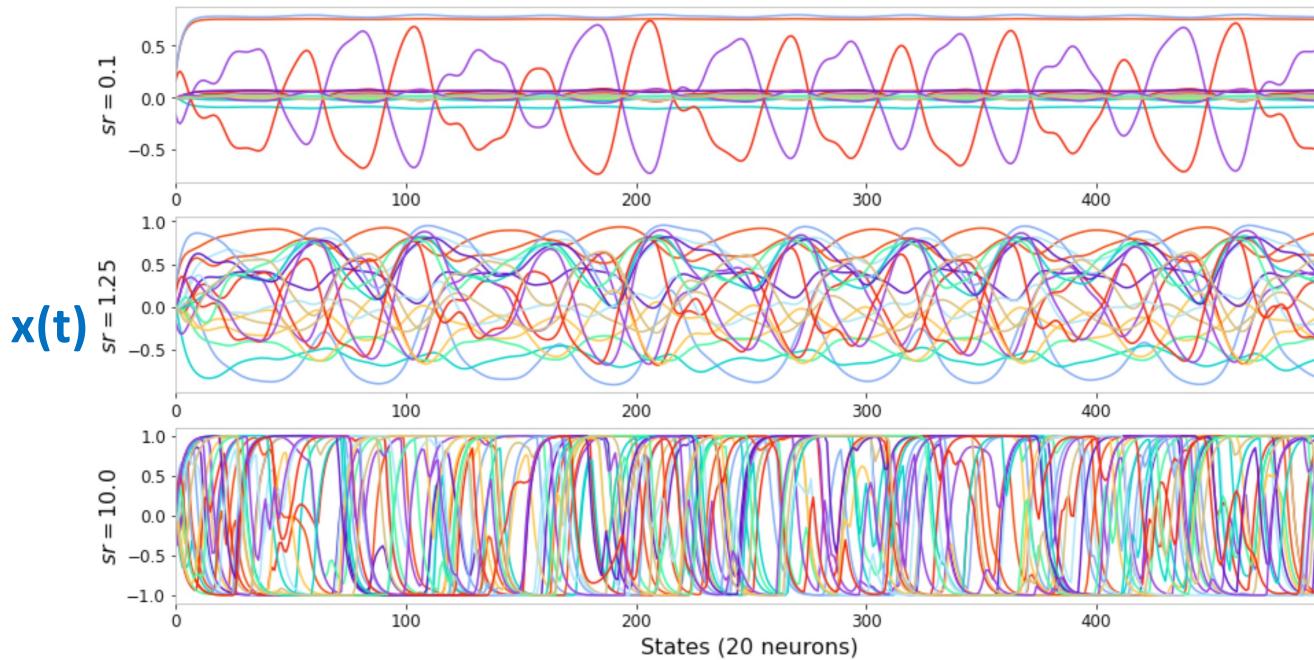
Understand ESN hyperparameters

```
In [1]:  
UNITS = 100          # - number of neurons  
LEAK_RATE = 0.3      # - leaking rate  
SPECTRAL_RADIUS = 1.25  # - spectral radius of W  
INPUT_SCALING = 1.0    # - input scaling  
RC_CONNECTIVITY = 0.1   # - density of reservoir internal matrix  
INPUT_CONNECTIVITY = 0.2  # and of reservoir input matrix  
REGULARIZATION = 1e-8    # - regularization coefficient for ridge regression  
SEED = 1234           # for reproducibility
```

https://github.com/reservoirpy/reservoirpy/blob/master/tutorials/4-Understand_and_optimize_hyperparameters.ipynb

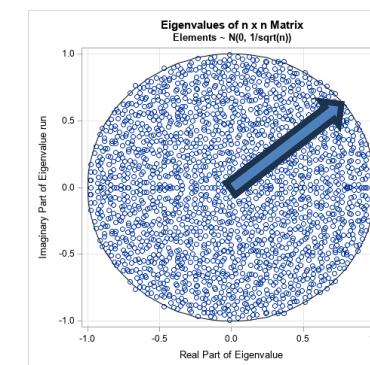
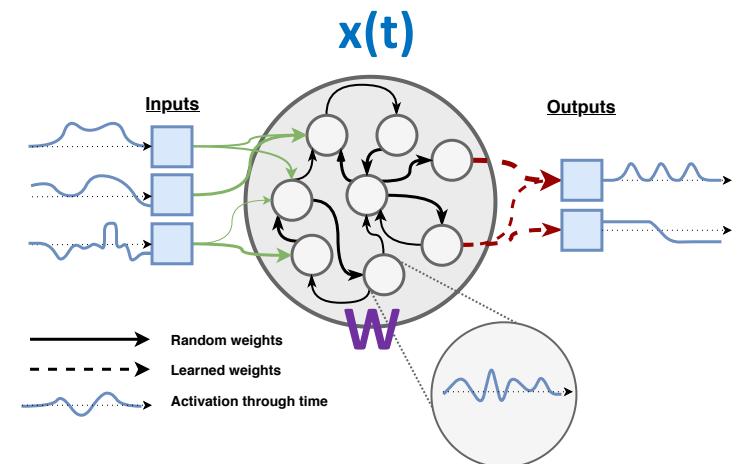
Spectral Radius (rayon spectral)

= scaling des connexions récurrentes



The spectral radius is defined as the maximum eigenvalue of the reservoir matrix.

- spectral radius → **stable** dynamics
- + spectral radius → **chaotic** dynamics



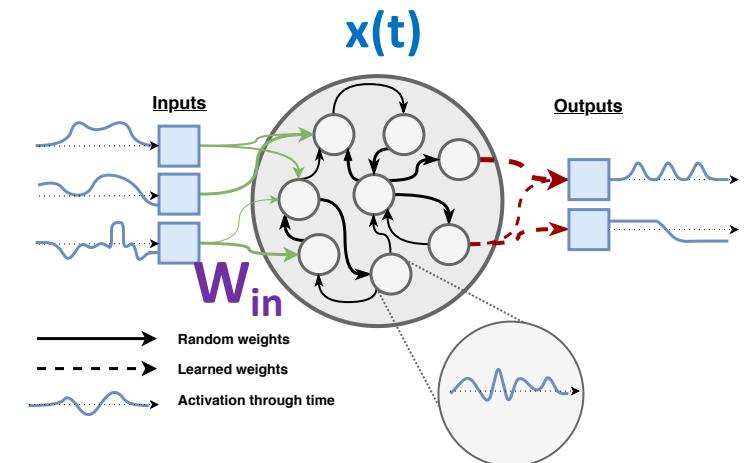
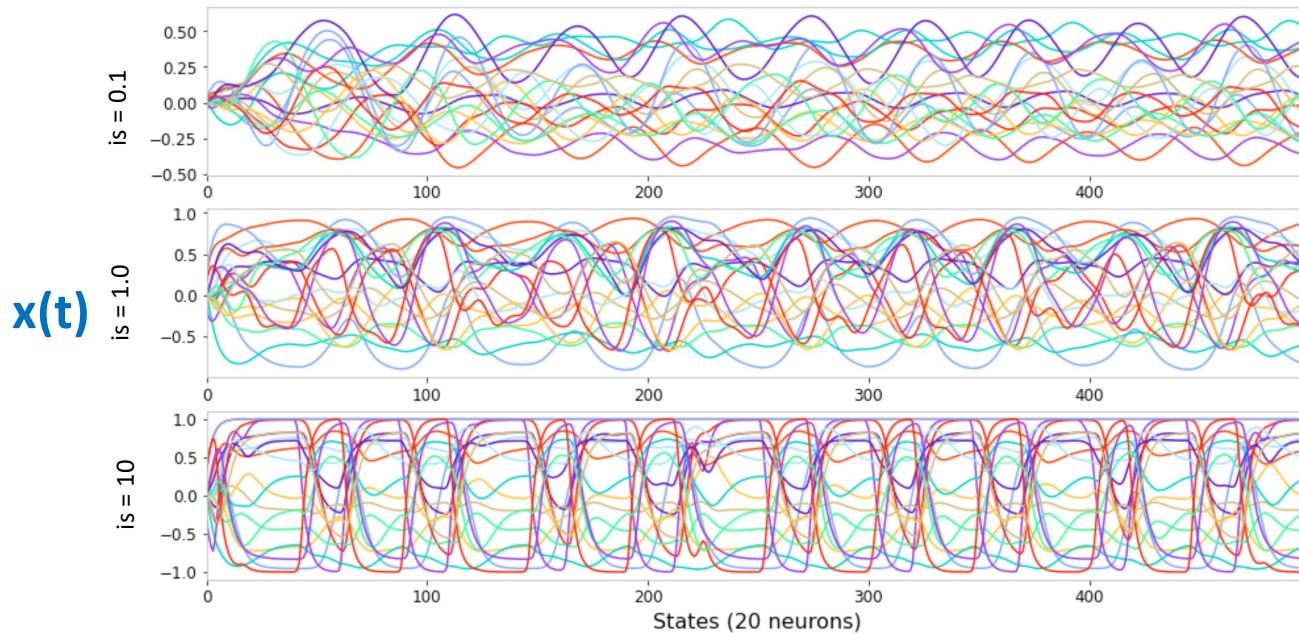
Rayon spectral de la matrice W

100

<https://blogs.sas.com/content/iml/2020/04/27/circular-law-eigenvalues.html>

Input Scaling

= scaling des connexions entrantes



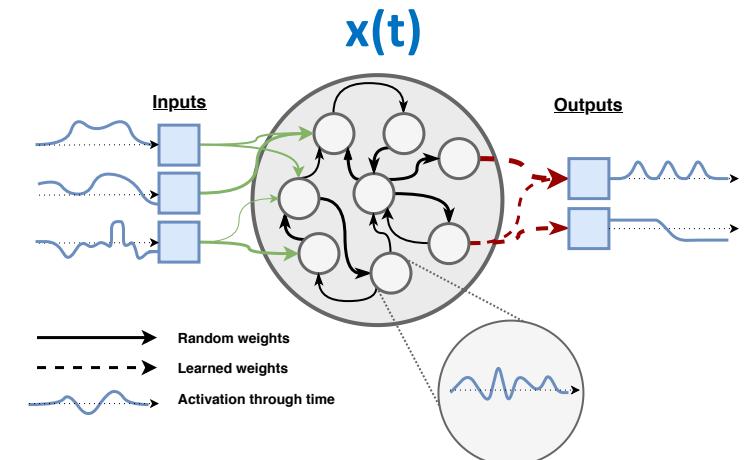
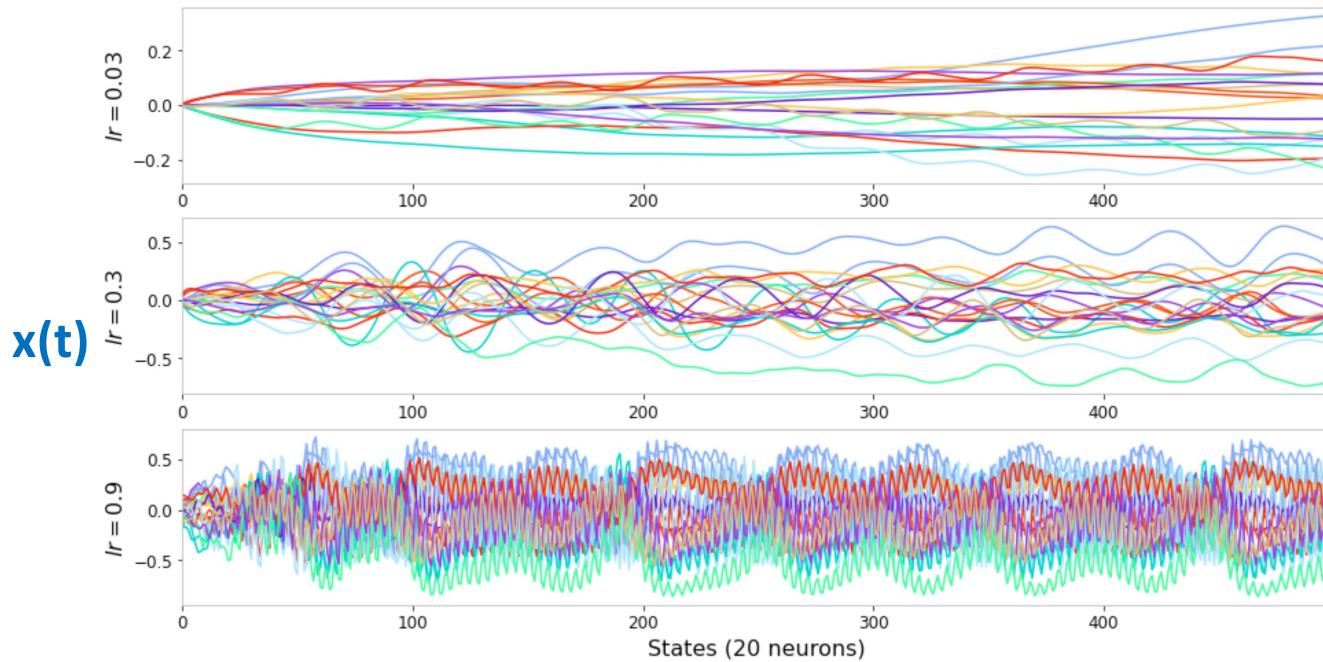
The input scaling controls how the ESN interact with the inputs. It is a coefficient applied to the input matrix W_{in} .

- + input scaling → **input-driven** activities
- - input scaling → **free** activities

The input scaling can also be used to rescale the inputs and adjust their influences.

Leak-rate (=taux de fuite)

= inverse de la constante de temps des neurones



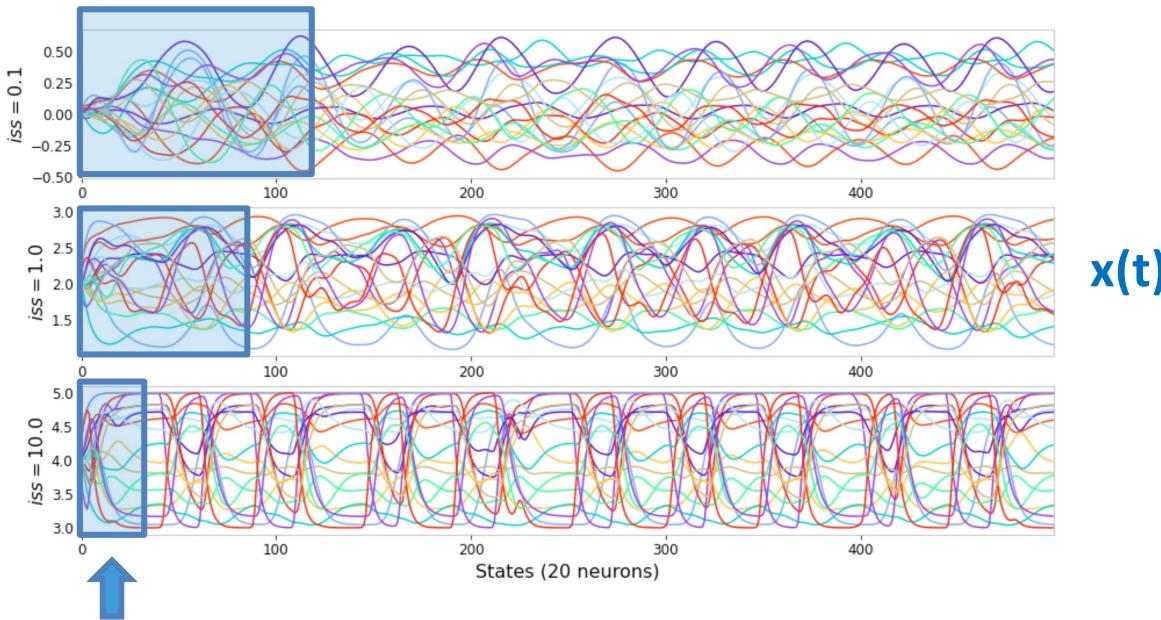
The leaking rate (α) controls the "memory feedback" of the ESN. The ESN states are indeed computed as:

$$s(t+1) = \underbrace{(1 - \alpha)s(t)}_{\text{previous states}} + \underbrace{\alpha f(u(t+1), s(t))}_{\text{new states}}$$

- + leaking rate → **low inertia**, little memory of previous states
- - leaking rate → **high inertia**, big memory of previous states

« Warming up »

warmup
ou
warming up

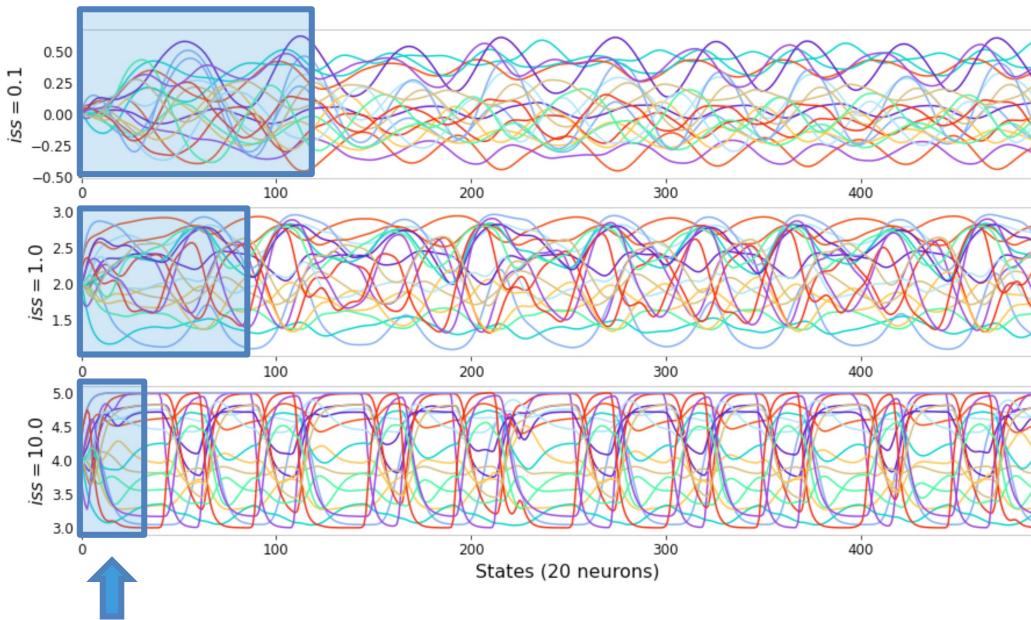


$x(t)$

- Le *warmup* est le « temps de chauffe » nécessaire pour que les dynamiques soient dans un rythme de « croisière ».
 - = zone bleue sur chaque figure
- Cela correspond au temps nécessaire pour que le réseau ne fasse plus vraiment la différence entre le début ou le milieu de la série temporelle (c-à-d après la zone bleue).
- = si je copie/colle un bout de graphique après le *warmup* vous ne serez pas capable de distinguer s'il s'agit du début/milieu/fin du graphique.

« Warming up »

warmup
ou
warming up



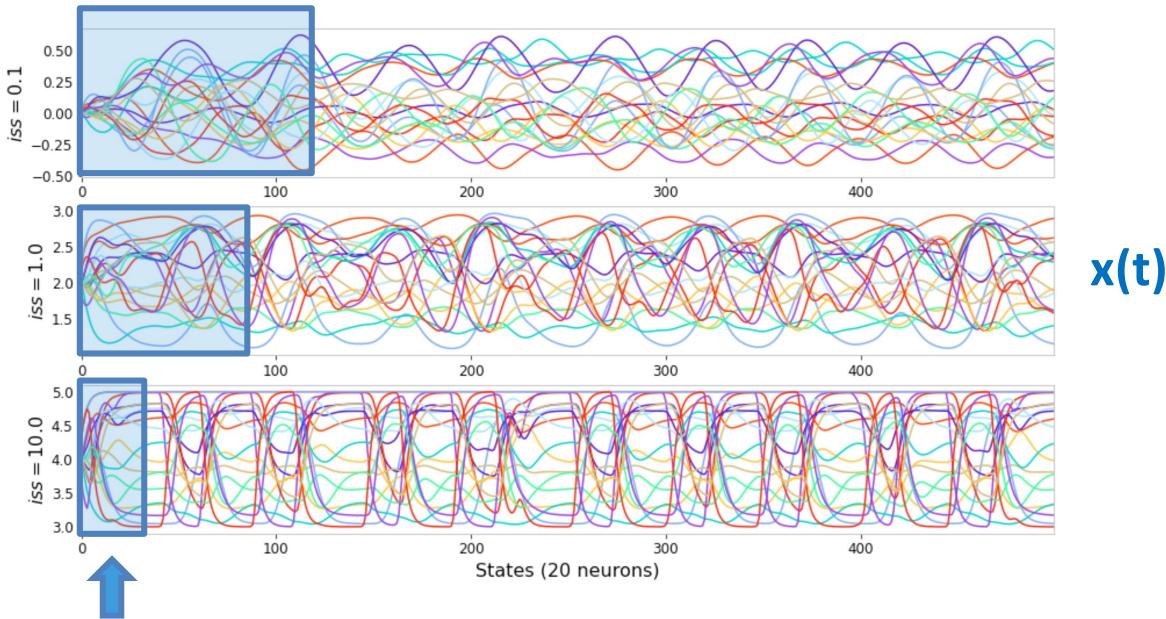
A priori, vous ne voulez pas utiliser ces dynamiques initiales du réservoir qui peuvent perturber l'apprentissage de votre tâche.

→ En conséquence, il faut définir l'hyperparamètre *warmup*

- « à l'œil » par rapport aux activités internes du réservoir
- et/ou en testant par essai/erreur pour quel *warmup* minimal vous avez des performances qui ne changent plus

« Warming up »

warmup
ou
warming up



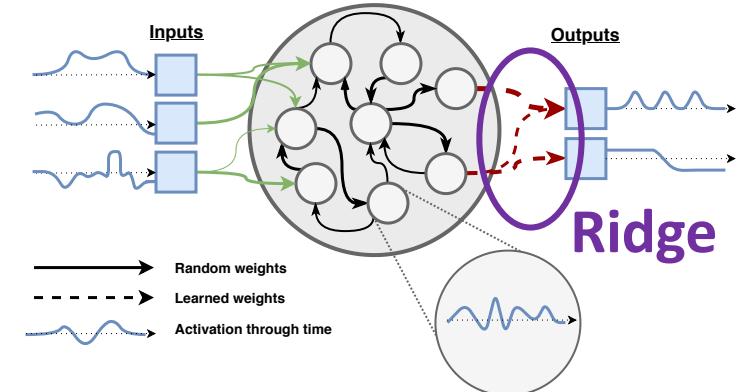
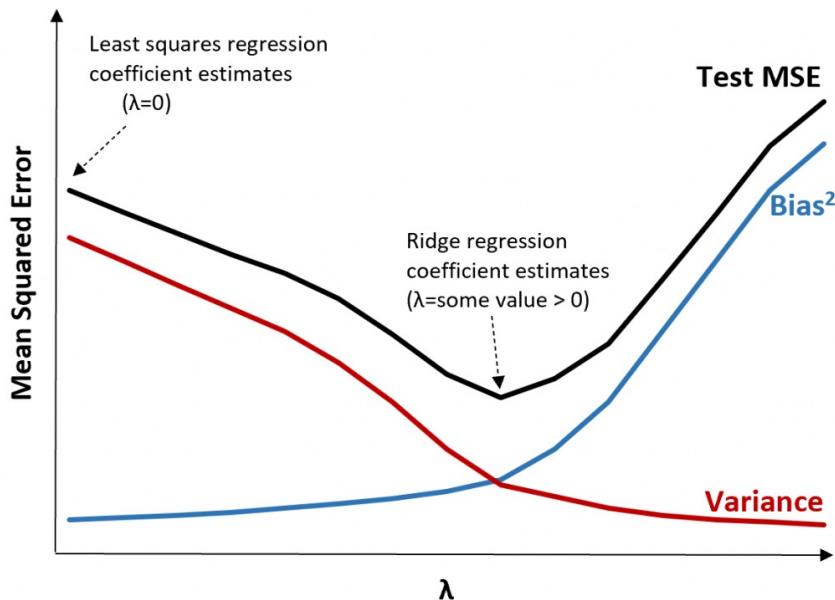
$x(t)$

Ce *warmup* / « temps de chauffe » dépend

- des hyperparamètres du réservoir
 - (ici l'input scaling iss)
- des entrées / des données que vous donnez au réservoir
 - (par ex. si le taux d'échantillonnage change)

Ridge

= régularisation de la régression linéaire



- Modèle avec **variance élevée** est très flexible
 - et s'adapte trop bien aux données d'entraînement, **y compris le bruit**
- Modèle avec **biais élevé** simplifie trop le problème
 - le modèle est **trop simple** pour capturer les relations sous-jacentes dans les données

Méthodes pour la recherche d'hyperparamètres

- « Random search » répétés et adaptés
 - Hinaut & Trouvain, ICANN 2021
 - Plus efficace que « grid search »
- Algorithmes génétiques avec 400 variables d'entrées
 - Ferte, ..., Hinaut, ICML 2024
 - Prédiction hospitalisations COVID à +14 jours

Which Hype for my New Task?
Hints and Random Search for
Echo State Networks Hyperparameters

Xavier Hinaut^{1,2,3,*}[0000-0002-1924-1184] and Nathan Trouvain^{1,2,3}[0000-0003-2121-7826]

¹ INRIA Bordeaux Sud-Ouest, France.
² LaBRI, Bordeaux INP, CNRS, UMR 5800.
³ Institut des Maladies Neurodégénératives,
Université de Bordeaux, CNRS, UMR 5293.

*Corresponding author: xavier.hinaut@inria.fr

Abstract. In learning systems, hyperparameters are parameters that are not learned but need to be set a priori. In Reservoir Computing, there are several parameters that need to be set a priori depending on the task. Newcomers to Reservoir Computing cannot have a good intuition on which hyperparameters to tune and how to tune them. For instance, beginners often explore the reservoir sparsity, but in practice this parameter is not of high influence on performance for ESNs. Most importantly, many authors keep doing suboptimal hyperparameter searches: using

<https://inria.hal.science/hal-03203318>

Reservoir Computing for Short High-Dimensional Time Series: an Application to SARS-CoV-2 Hospitalization Forecast



Thomas Ferté, Dan Dutartre, Boris P Hejblum, Romain Griffier, Vianney Jouhet, Rodolphe Thiébaut, Pierrick Legrand, Xavier Hinaut

Published: 02 May 2024, Last Modified: 25 Jun 2024 [ICML 2024 Poster](#) [Everyone](#) [Revisions](#) [BibTeX](#)
CC BY-NC 4.0

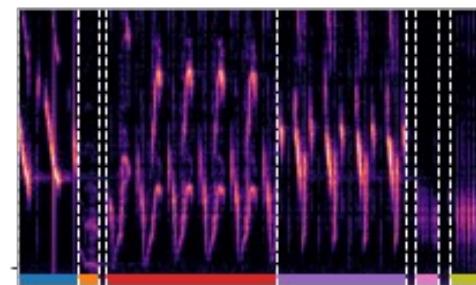
Abstract:

In this work, we aimed at forecasting the number of SARS-CoV-2 hospitalized patients at 14 days to help anticipate the bed requirements of a large scale hospital using public data and electronic health records data. Previous attempts led to mitigated performance in this high-dimension setting: we introduce a novel approach to time series forecasting by providing an alternative to conventional methods to deal with high number of potential features of interest (409 predictors). We integrate Reservoir Computing (RC) with feature selection using a genetic algorithm (GA) to gather optimal non-linear combinations of inputs to improve prediction in sample-efficient context. We illustrate that the RC-GA combination exhibits excellent performance in forecasting SARS-CoV-2 hospitalizations. This approach outperformed the use of RC alone and other conventional methods: LSTM, Transformers, Elastic-Net, XGBoost. Notably, this work marks the pioneering use of RC (along with GA) in the realm of short and high-dimensional time series, positioning it as a competitive and innovative approach in comparison to standard methods.

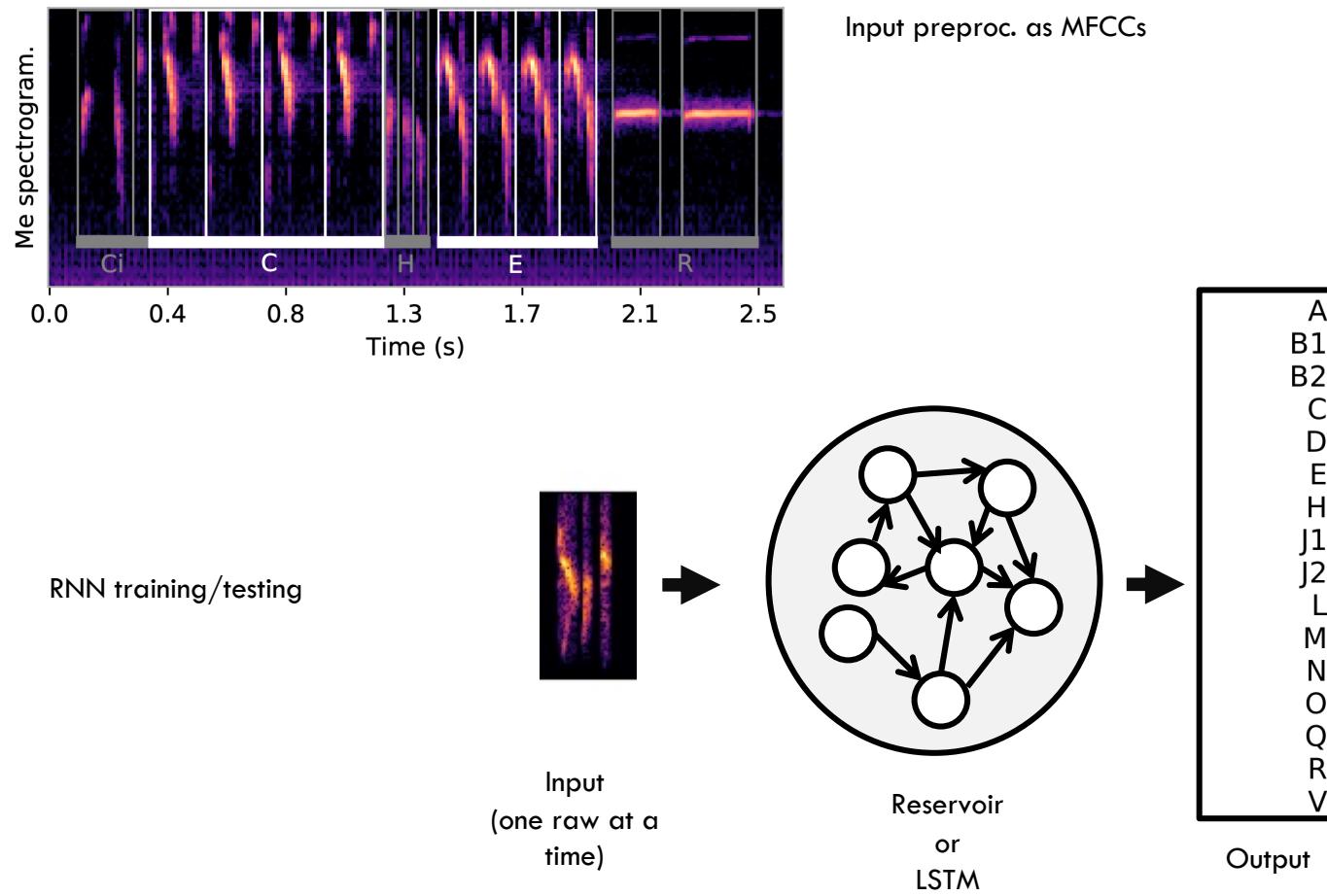
<https://openreview.net/forum?id=CY0lFwD4qx>

EXEMPLE D'APPLICATION

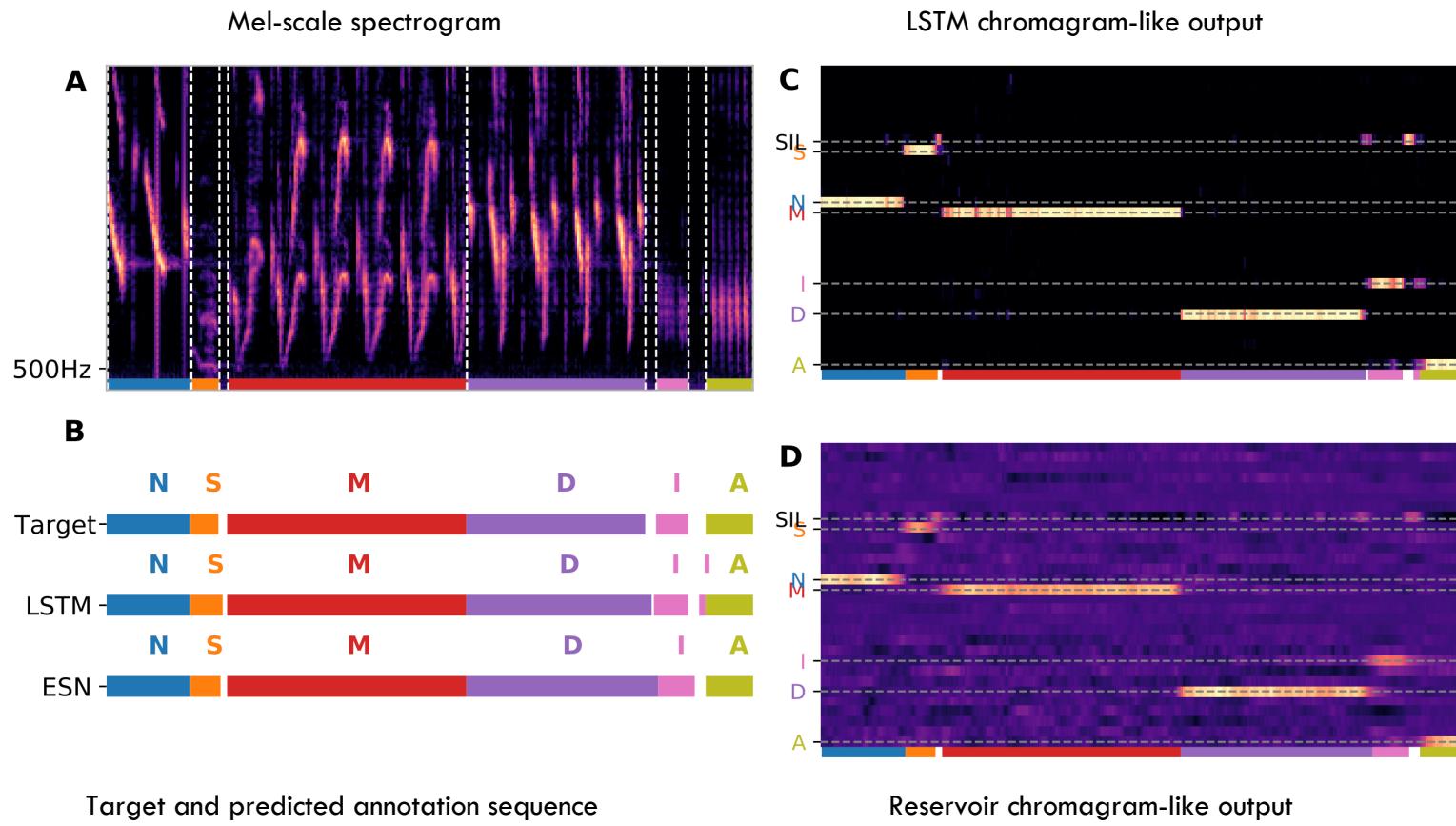
DISCRIMINATION CHANT CANARIS RESERVOIR VS. LSTM



Discrimination de phrases de canaris



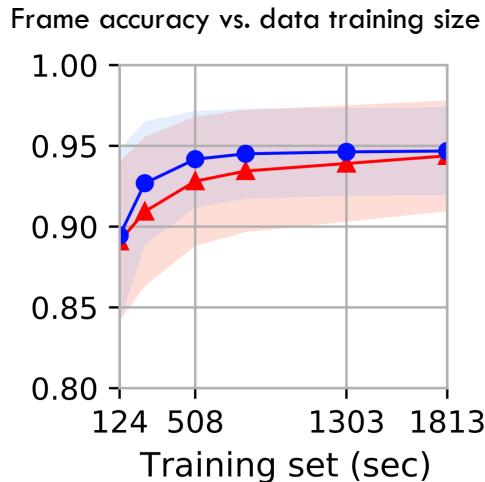
Discrimination de phrases de canaris



Discrimination de phrases de canaris

Table 1. Average scores obtained with a 5-fold cross-validation over all training songs and several models instances.

Model	Average frame accuracy (ACC)	Median frame accuracy	F1 (macro avg.)
LSTM	0.931 ± 0.104	0.951	0.865
ESN	0.935 ± 0.09	0.952	0.877

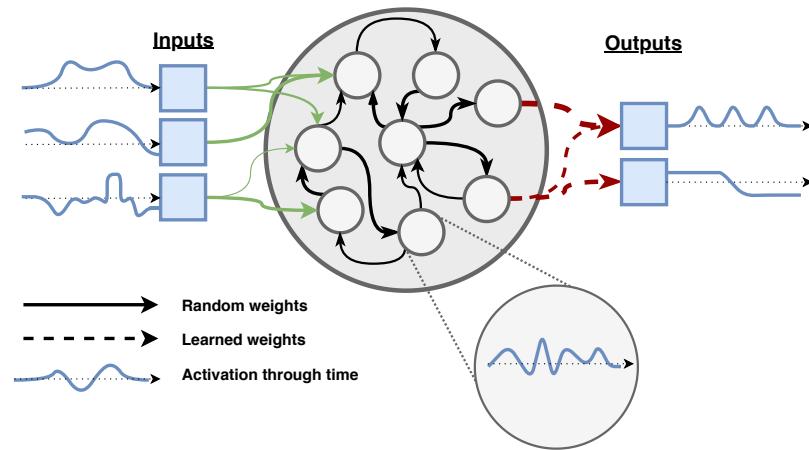
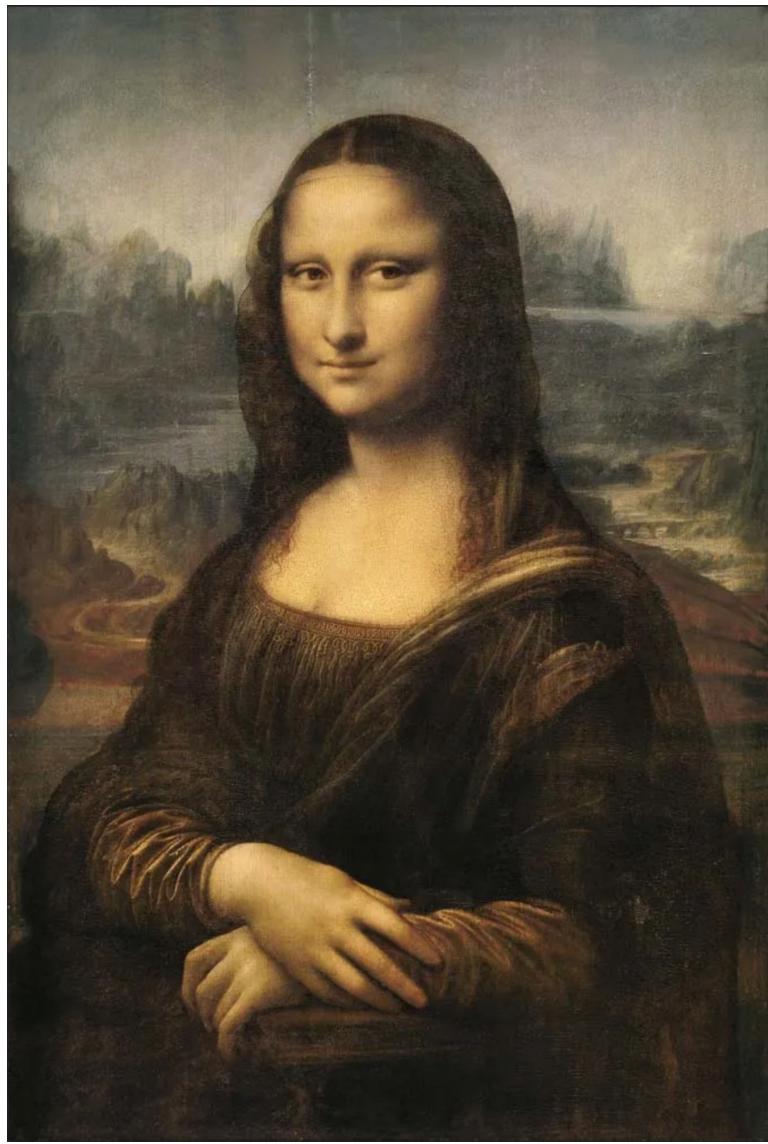


Average training time on multi-core CPU (10-fold CV)

Model	Average training time (s)
LSTM	2930 ± 222
ESN	35 ± 1

Les performances en généralisation sont similaires entre les LSTM et les réservoirs, mais :

- les réservoirs généralisent avec moins de données
- les réservoirs sont bien plus rapides à entraîner





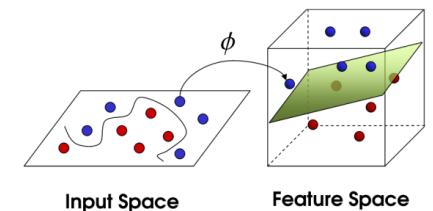
En Résumé

- Le Reservoir Computing permet de
 - Prédire / Générer / Classifier
 - Simuler des systèmes complexes, ...
 - Détection de pannes,
 - Entraînement *online* et *offline*
 - Pour des données temporelles (et non-temporelles)
- **ReservoirPy** propose une **IA frugale**
 - Bonnes performances même avec peu de données nécessaires
 - Economie de calculs / temps
 - Economie d'énergie
- **Intérêts pour la santé**
 - Peu de données nécessaires
 - Applicable à un seul ou quelques patients
- Transférable sur différents types de *hardware*
 - système physique optiques, magnétiques, etc.

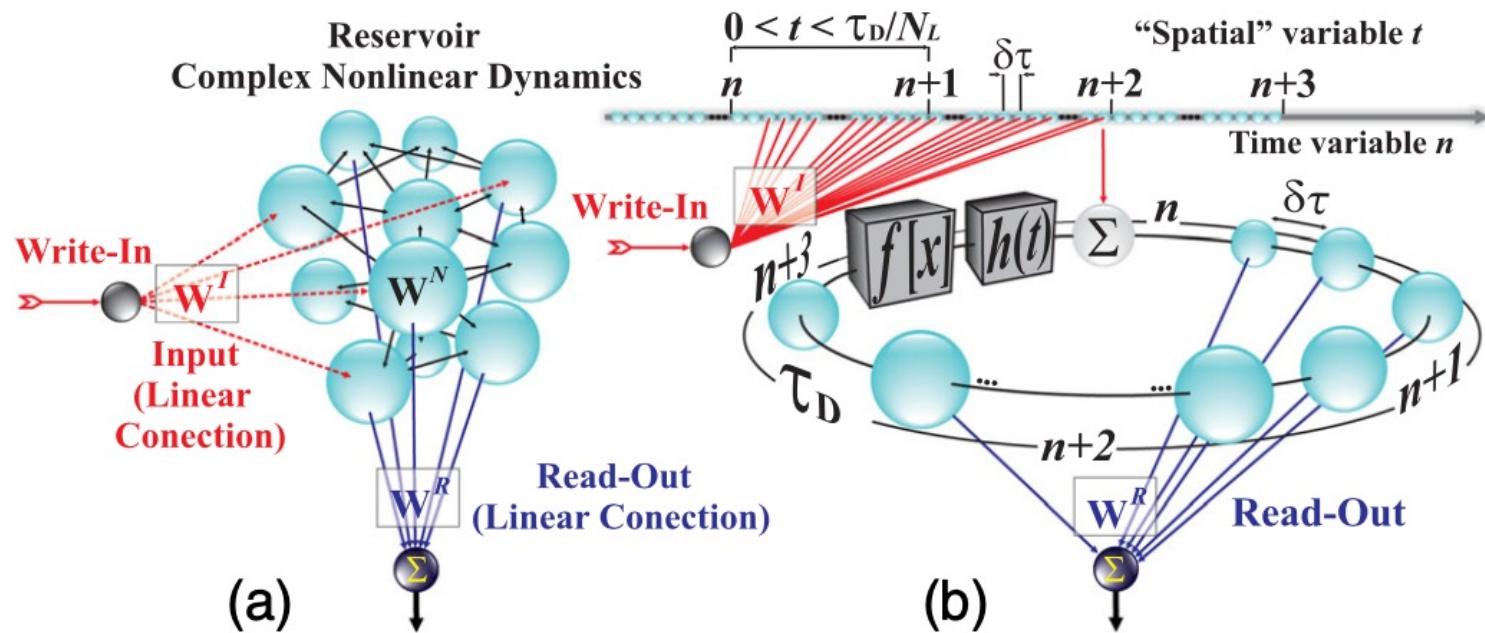


Take-Home message

- Les projections aléatoires en grande dimension sont une resource non négligeable de calculs
 - Hyperparamètres pour "garder du contrôle"
- $RC = Bcp \text{ de calculs "pas cher"} + \text{sélection}$
- Similaire à un SVM temporel
- Intuition sur ce que font les hyperparamètres
- Bcp de réservoirs possible sur hardware



Vers le Reservoir Computing Optique

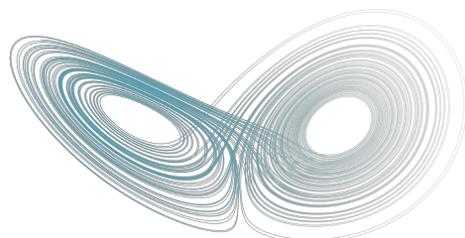


Larger et al. 2017

Xavier HINAUT - Mnemosyne

116

ReservoirChat



chat.reservoirpy.inria.fr

ReservoirChat



Inria

User  What's the point of ridge regression in reservoir computing ? 
10:32

ReservoirChat  Ridge regression is used in reservoir computing to prevent overfitting and improve the generalization of the model. It does this by adding a penalty to the regression equation to reduce high-value coefficients, making the model more stable. This technique is particularly useful when dealing with multicollinearity, where two or more independent variables are highly correlated. By adding a regularization term to the loss function, ridge regression helps to mitigate the risk of overfitting and improve the model's performance on new, unseen data.

In the context of reservoir computing, ridge regression is used to compute the readout map from reservoir states to the output signal. This is typically done by linear regression using the MSE loss, but ridge regression can be used instead to add a penalty to the loss function and prevent overfitting. The regularization term in ridge regression



Send a message

 Send

 Rerun

 Undo

 Clear

Fin du tutoriel Reservoir Computing



Ingénieurs dév. ReservoirPy

Actuel : Paul Bernard

Précédent : Nathan Trouvain



github.com/reservoirpy/reservoirpy



chat.reservoirpy.inria.fr



sites.google.com/view/hack1robo
<https://www.instagram.com/p/C2SkVGKMchu>





- J'ai initié le logiciel *from scratch* en 2019 et poursuivi la supervision de son dév.
- Puis développement poursuivi par 4 H.A. d'ingénieurs, post-doctorants, stagiaires
- Logiciel de *Reservoir Computing* le plus utilisé et mis à jour constamment
 - un ingénieur à temps-plein actuellement
- Licence MIT : *open source* + utilisation commerciale sans contraintes
 - Cela permet une **adoption large et rapide**, afin de se placer comme un **standard de l'industrie**

Suite pour la partie TP



github.com/reservoirpy/reservoirpy

Téléchargez le tutoriel pour le TP
→ tester ReservoirPy sur votre machine

<https://github.com/reservoirpy/presentations>

Installation et lancement du jupyter notebook

- Ouvrir un terminal, et taper cette commande pour installer reservoirpy avec les dépendances pour hyperopt
- pip install reservoirpy[hyper]
- Télécharger le notebook à l'adresse
https://github.com/reservoirpy/presentations/blob/main/Cours-institut-d-optique-2024-2025/01_Introduction.ipynb
- Et le mettre dans le dossier de votre choix
- Lancez Jupyter Notebook dans ce même dossier avec la commande
- jupyter notebook
- Cela ouvre votre navigateur. Sélectionnez ensuite le fichier “01_Introduction.ipynb” pour lancer le notebook.
- Lisez et exécutez le notebook.
- Faites des modifications dans le notebook en testant différents paramètres.

