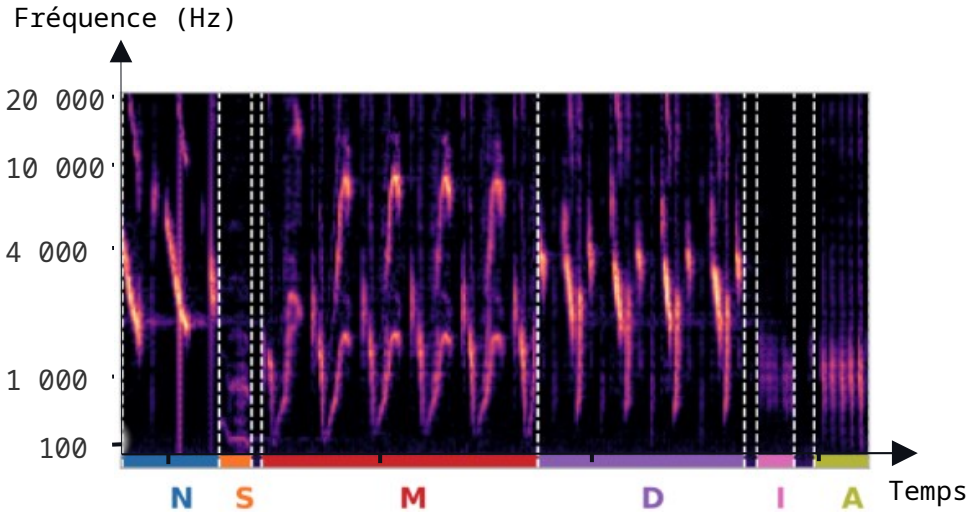
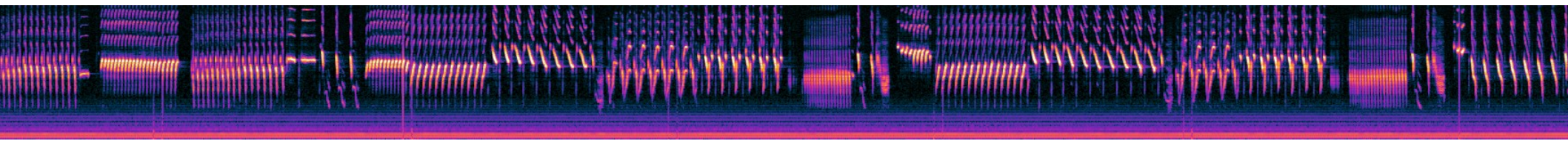
Reservoir Computing : de la théorie à la pratique avec **ReservoirPy**

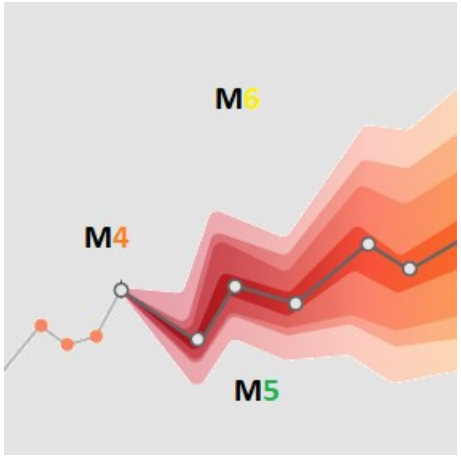
Paul BERNARD – Séminaire Phimeca 2023

Séries temporelles : Classification





Statistiques vs Machine learning



github.com/Mcompetitions

Problème de taille :
Les méthodes de ML généralisent mieux avec plus de données

Conclusion de la compétition Makridakis 3 :

Statistiques > Machine learning
(sur la prédiction)

Cerqueira, 2019, arXiv

Le coût du deep learning en terme d'énergie

284 MWh

Le coût du deep learning en terme d'énergie

ChatGPT (GPT-3)

284 MWh

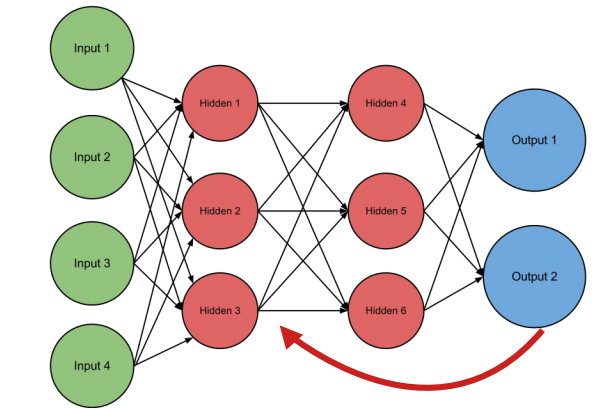
4 MWh

(20W × 23 ans)

Comment apprendre de manière plus économe ? (en calcul, en énergie, en données)

L'entraînement de réseaux de neurones classiques

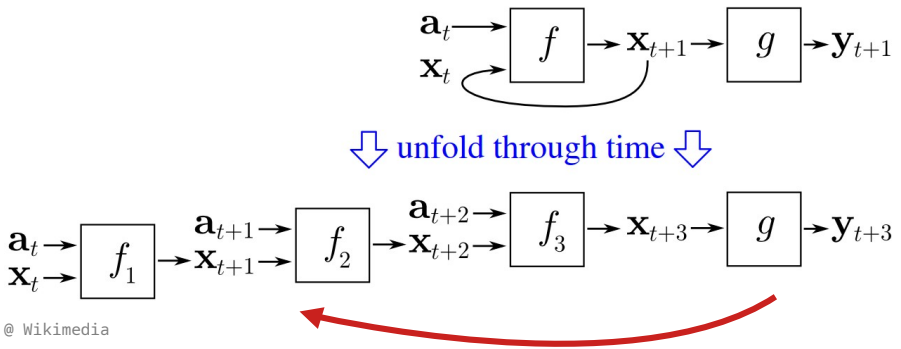
Réseaux « feed-forward »



@ Wikimedia

Rétropropagation

Réseaux récurrents

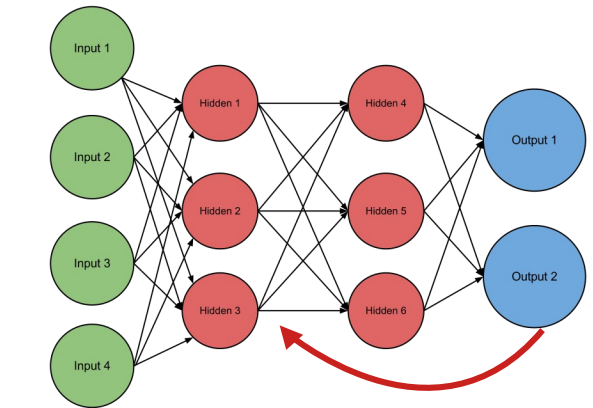


@ Wikimedia

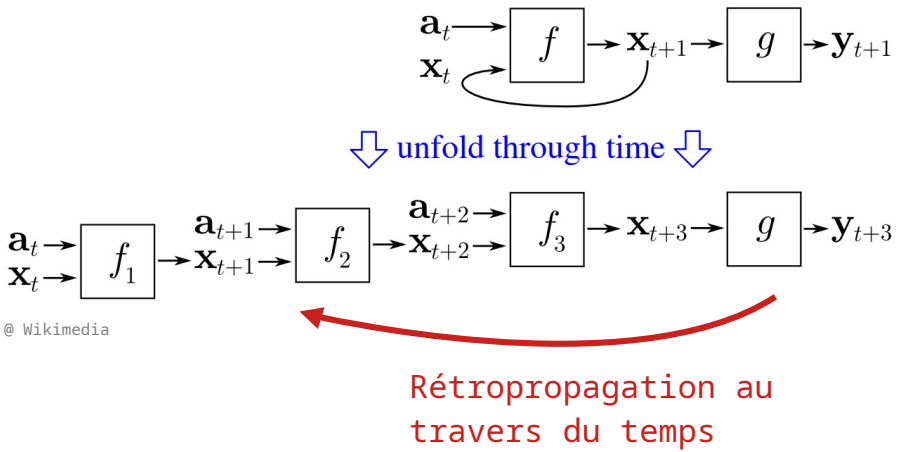
Rétropropagation au travers du temps

L'entraînement de réseaux de neurones classiques

Réseaux « feed-forward »

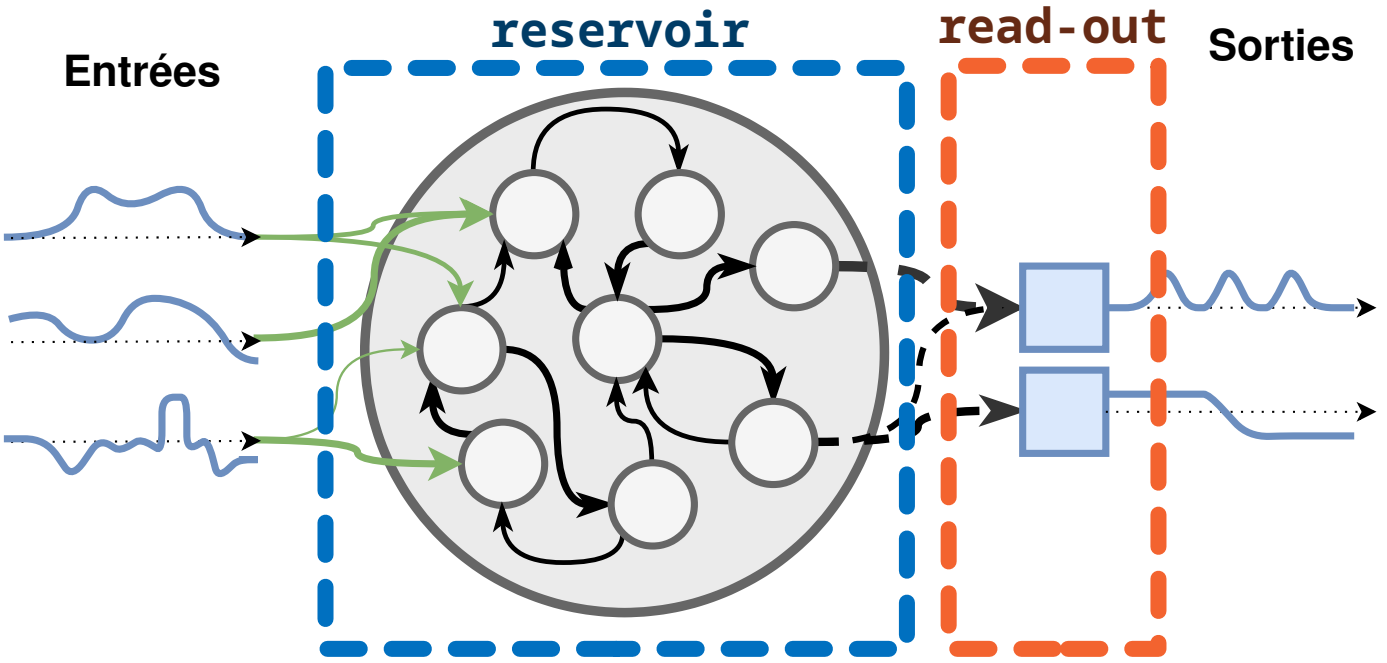


Réseaux récurrents



→ **C'est coûteux !**

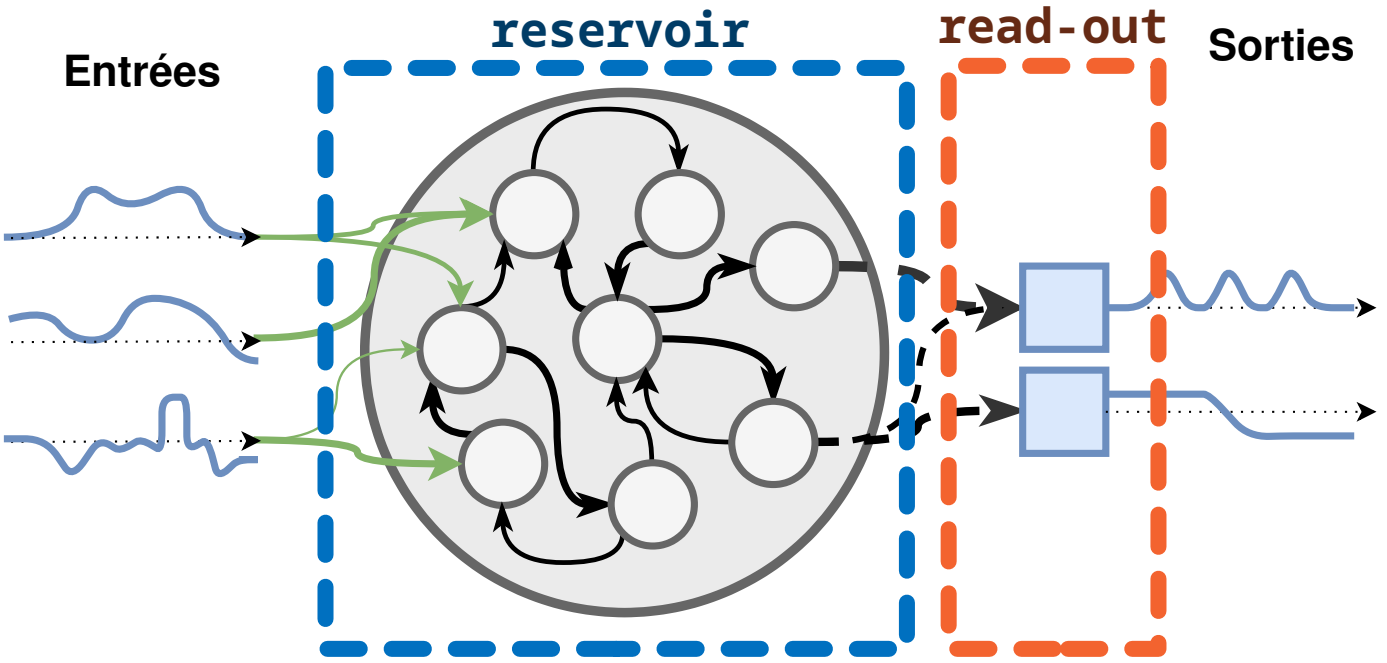
Le **Reservoir Computing**



- Partie récurrente
- Paramètres fixes

- Partie entraînée
- Régression linéaire

Le **Reservoir Computing**

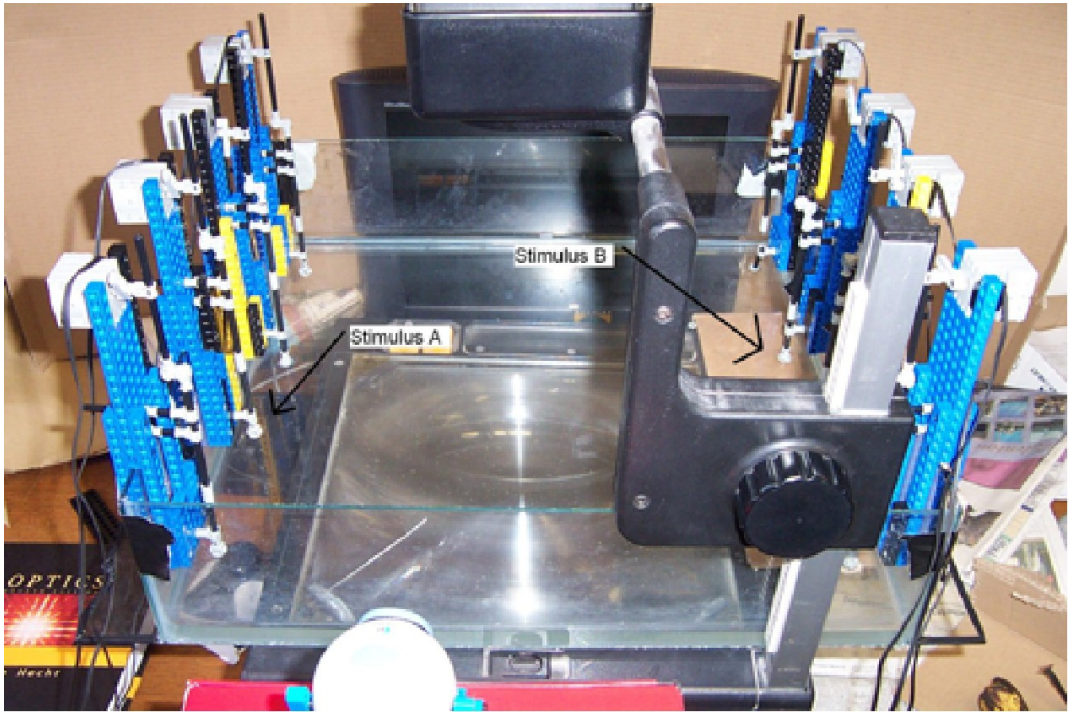


- Partie récurrente
- **Paramètres fixes**

- **Partie entraînée**
- Régression linéaire

→ **Très peu de paramètres appris, simple régression linéaire**

De l’informatique non-conventionnelle



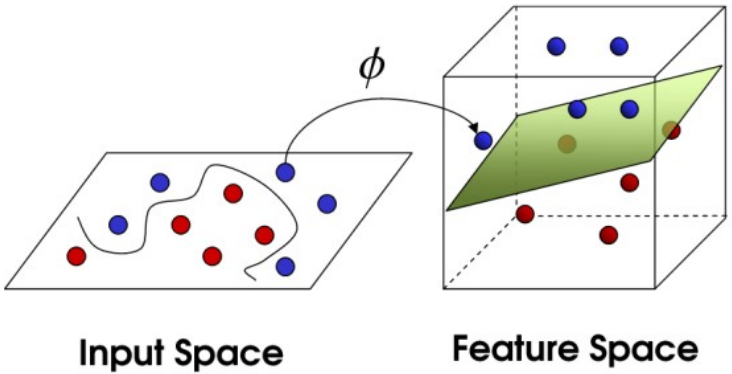
- **Moteurs** : l’entrée
- **Le bac d’eau** : le réservoir
- **La caméra** : le read-out

- **La surface de l’eau** : l’état du réservoir

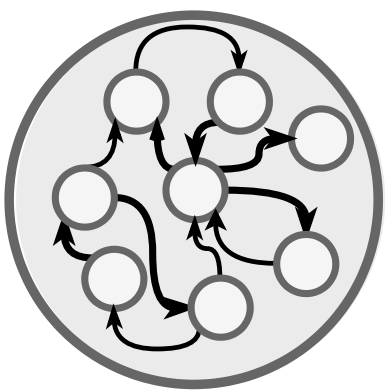
Un **kernel trick** temporel

On applique une fonction non-linéaire complexe qui projette les entrées dans un espace de grande dimension

→ Le problème est rendu linéaire !



Drew Wilimitis

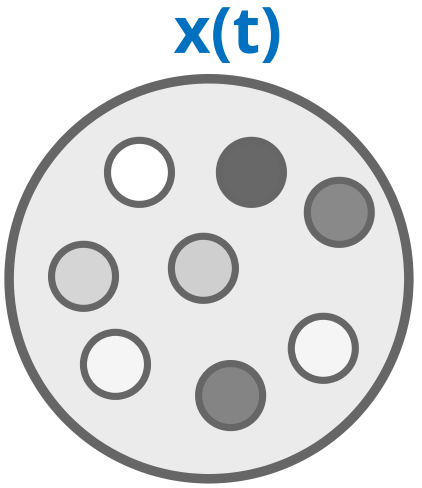


Ici, le réservoir transforme le signal d'entrée ($\ldots, u_{t-2}, u_{t-1}, u_t$) dans un espace de haute dimension (l'état des neurones)

→ Le read-out peut appliquer une régression linéaire !

Echo State Network (ESN)

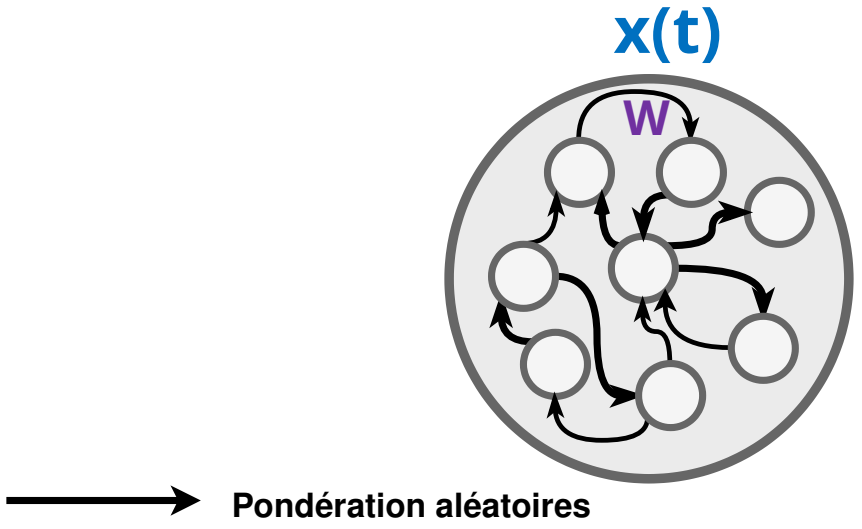
Un tas de neurones



$$x(t)$$

Echo State Network (ESN)

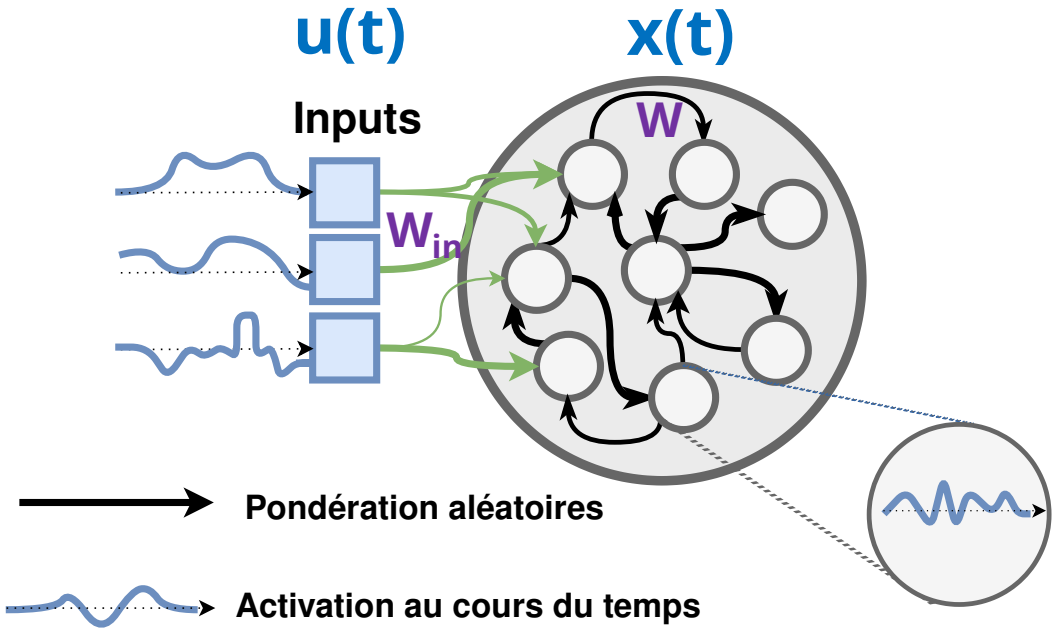
Un tas de neurones interconnectés



$$x(t) = \quad Wx(t-1)$$

Echo State Network (ESN)

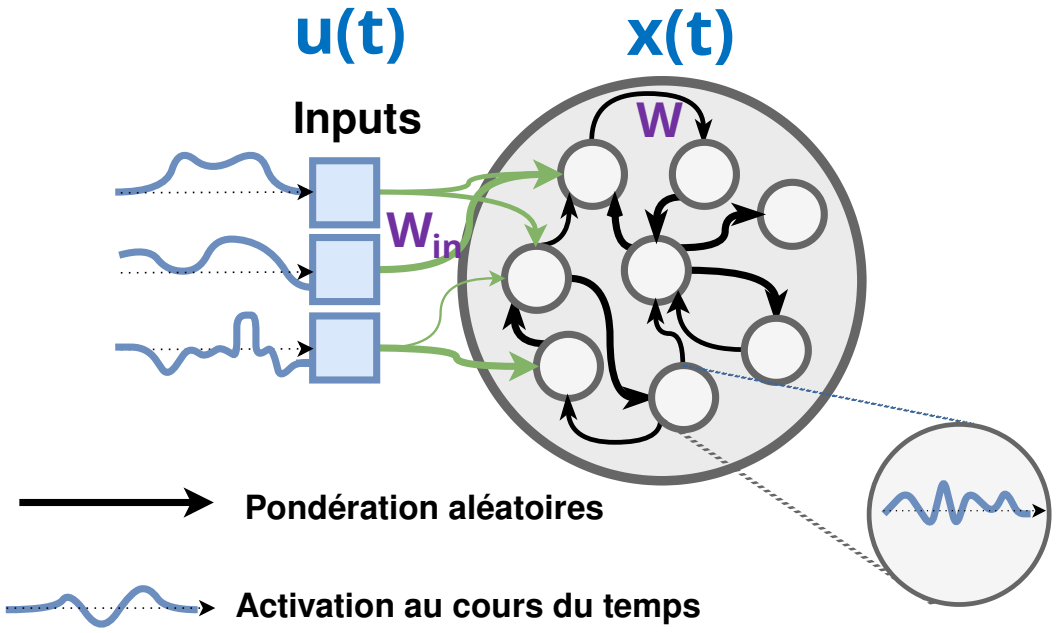
Entrées du réservoir



$$x(t) = \quad Wx(t-1) + W_{in}u(t)$$

Echo State Network (ESN)

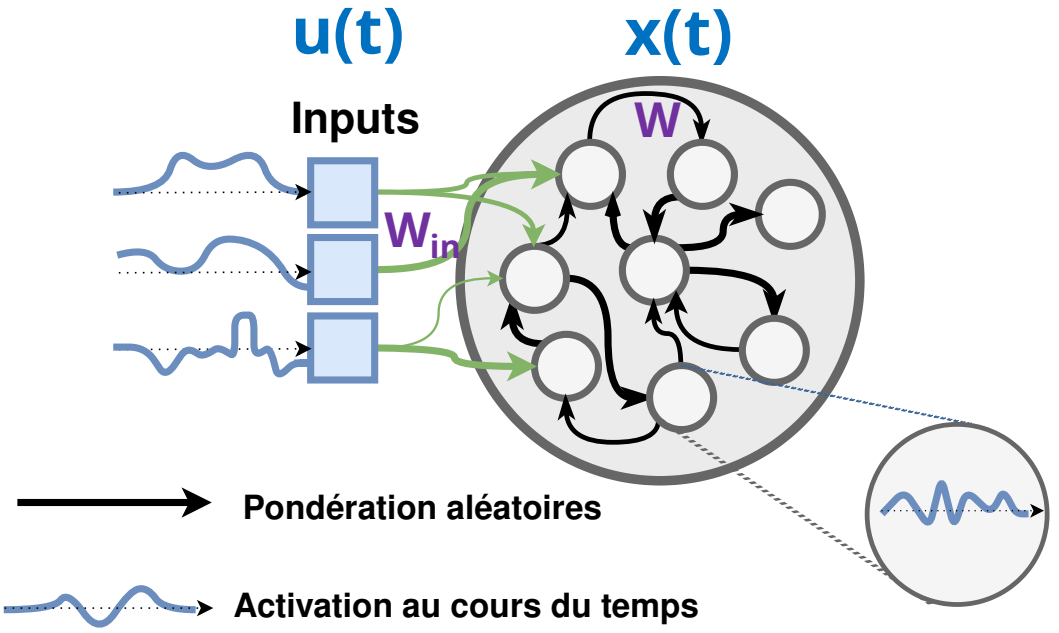
Non-linéarité



$$x(t) = f(Wx(t-1) + W_{in}u(t))$$

Echo State Network (ESN)

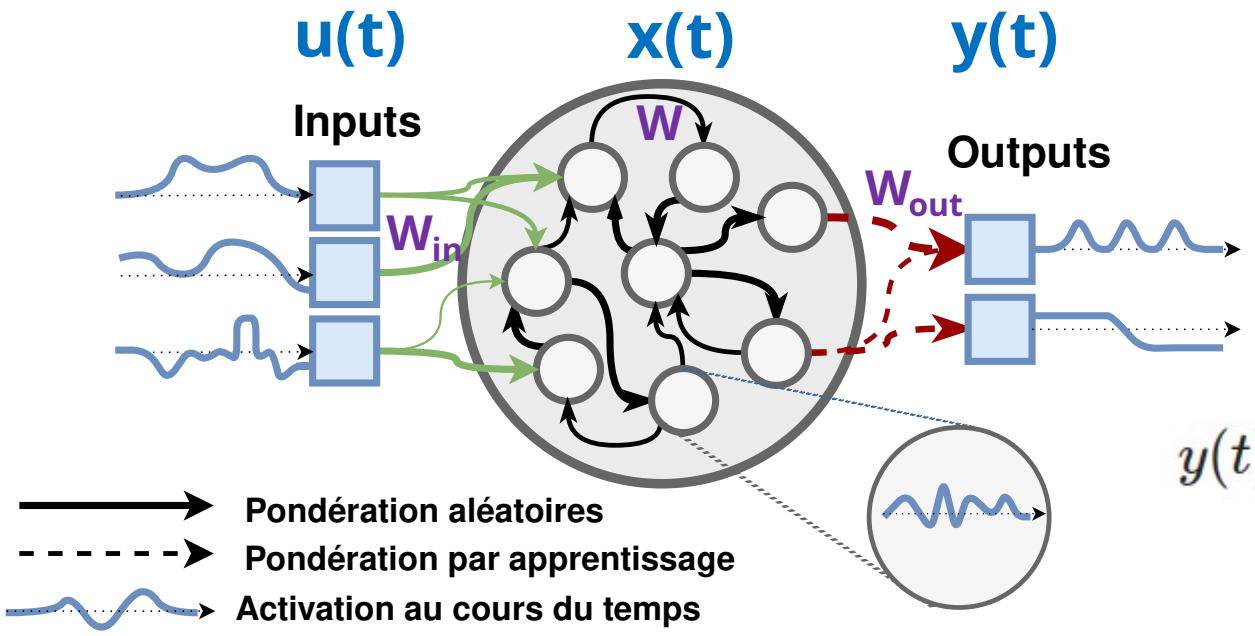
Neurones à fuite



$$x(t) = \frac{1}{\tau} f(W x(t-1) + W_{in} u(t)) + (1 - \frac{1}{\tau}) x(t-1)$$

Echo State Network (ESN)

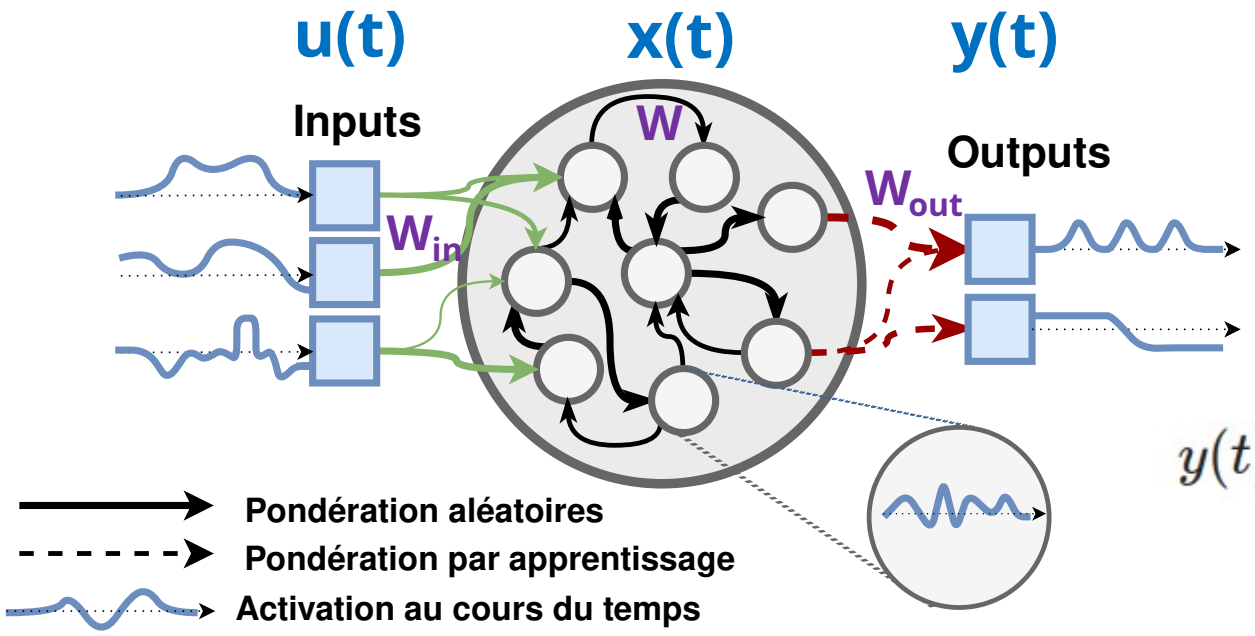
La couche de sortie (read-out)



$$y(t) = W_{out}x(t)$$

$$x(t) = \frac{1}{\tau}f(Wx(t-1) + W_{in}u(t)) + (1 - \frac{1}{\tau})x(t-1)$$

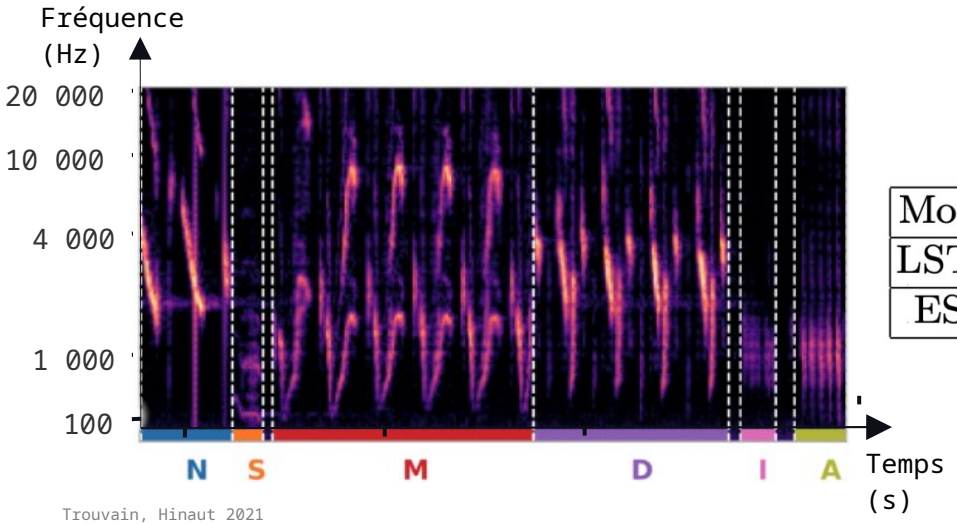
Echo State Network (ESN)



$$y(t) = W_{out}x(t)$$

$$x(t) = \frac{1}{\tau}f(Wx(t-1) + W_{in}u(t)) + (1 - \frac{1}{\tau})x(t-1)$$

Echo State Network et LSTM



Model	Average frame accuracy (ACC)	Average training time (s)
LSTM	0.931 ± 0.104	2930 ± 222
ESN	0.935 ± 0.09	35 ± 1

Performances similaires mais :

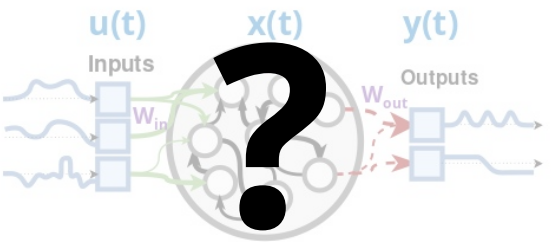
- les réservoirs généralisent avec moins de données
- les réservoirs sont bien plus rapides à entraîner

Quel outil Python pour le reservoir computing ?

SciPy

scikit learn

TensorFlow



PyTorch

ReservoirPy: un module Python pour le reservoir computing

Libre et open-source (licence MIT)

Basé sur l'écosystème NumPy / SciPy

Entièrement documenté

Maintenu à jour

https://github.com/reservoirpy/reservoirpy

Une architecture en nœuds

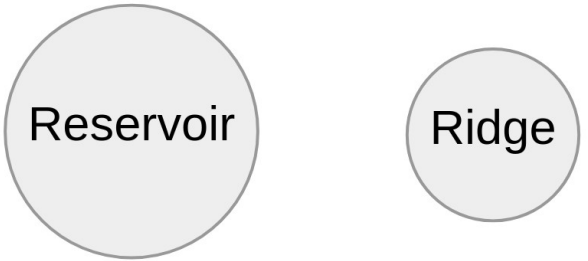
Création de nœuds

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

reservoir = Reservoir(units=100, lr=0.9, sr=0.9)
```

Une architecture en nœuds

Création de nœuds

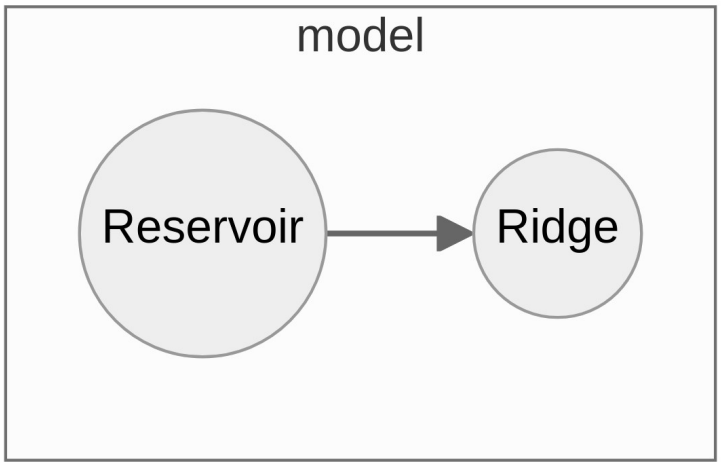
```python
from reservoirpy.nodes import Reservoir, Ridge

reservoir = Reservoir(units=100, lr=0.9, sr=0.9)
readout = Ridge(ridge=0.001)
```



Une architecture en nœuds

Création de modèle

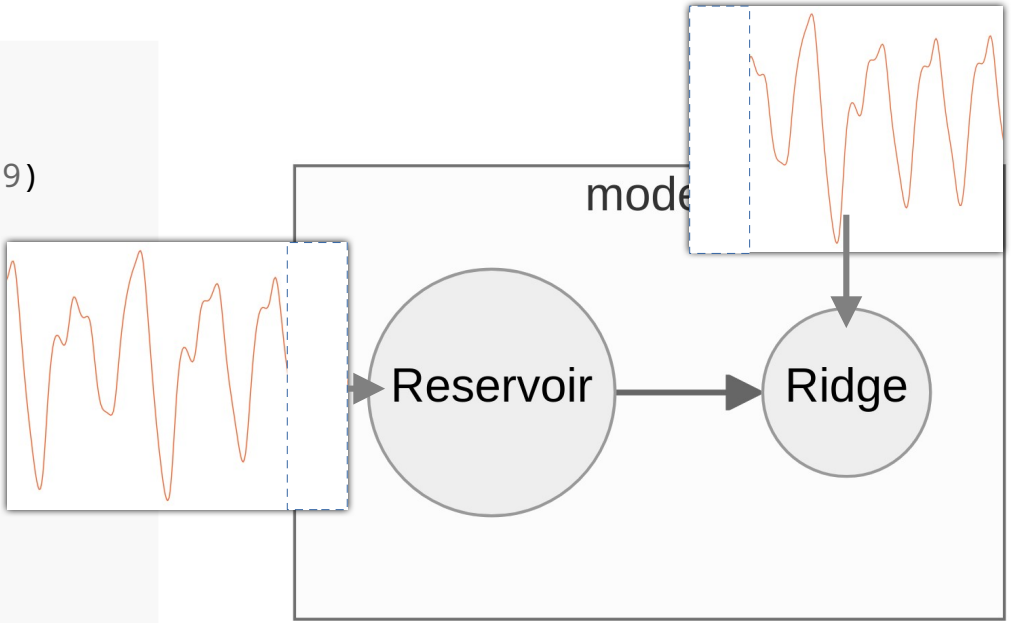
```python
from reservoirpy.nodes import Reservoir, Ridge

reservoir = Reservoir(units=100, lr=0.9, sr=0.9)
readout = Ridge(ridge=0.001)

model = reservoir >> readout
```



Une architecture en nœuds

Entraînement

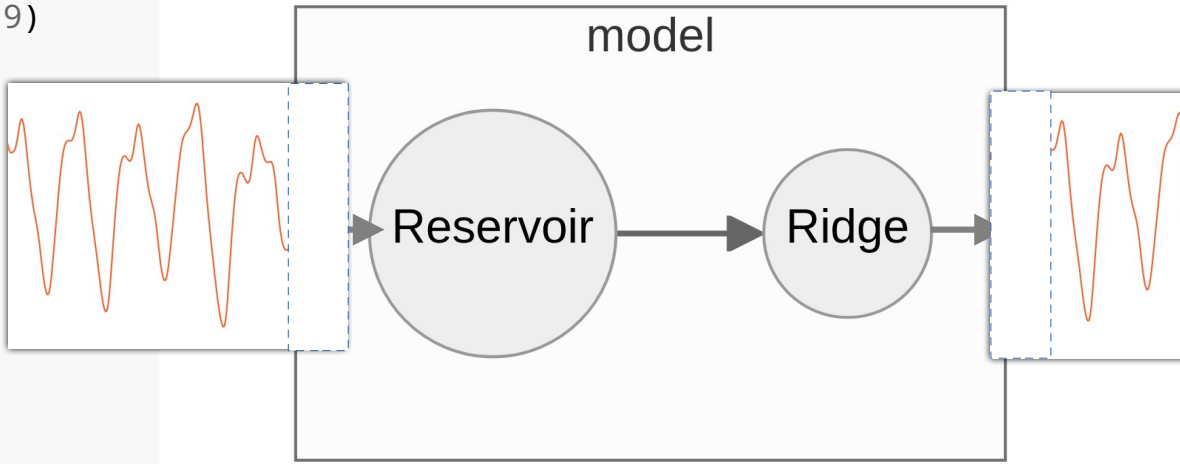
```python
from reservoirpy.nodes import Reservoir, Ridge

reservoir = Reservoir(units=100, lr=0.9, sr=0.9)
readout = Ridge(ridge=0.001)

model = reservoir >> readout

from reservoirpy.datasets import mackey_glass
X = mackey_glass(2500)


model.fit(X[:500], X[1:501], warmup=100)
```



Une architecture en nœuds

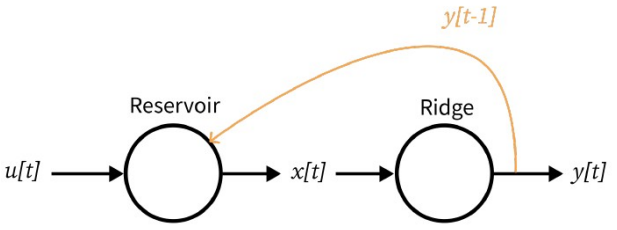
Prédiction t+1

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

reservoir = Reservoir(units=100, lr=0.9, sr=0.9)
readout = Ridge(ridge=0.001)

model = reservoir >> readout

from reservoirpy.datasets import mackey_glass
X = mackey_glass(2500)


model.fit(X[:500], X[1:501], warmup=100)

Y_pred = model.run(X[501:-1])
```

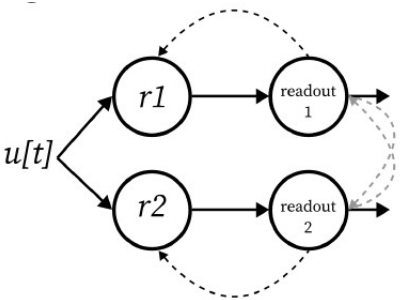


Une bibliothèque riche

- Possibilité de feed-back



- Créer des modèles complexes



- Interface pour R

Perspectives

Implémentation de fonctionnalités

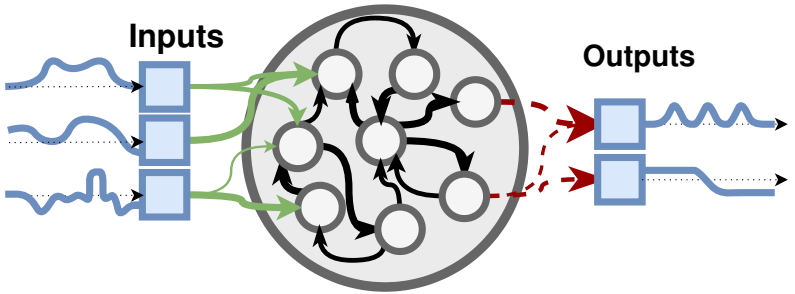
- Intégration de nouvelles méthodes
- Calculs sur GPU
- Interfaçage scikit-learn, PyTorch, ...

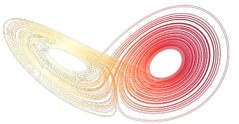
Aspect communautaire

- Rédaction de tutoriels
- Implémentation de papiers

Ouverts aux suggestions et nouveaux cas d'applications !

Résumé



reservoirpy 

Un paradigme :

- pour des **séries temporelles**
- qui **sépare récurrence et apprentissage**
- **économique**

Un outil :

- **accessible**
- **flexible**
- **maintenu**

Démonstration

https://paul.bernard-candaele.com/phimeca/

Séances de questions

Dépôt GitHub
https://github.com/reservoirpy/reservoirpy

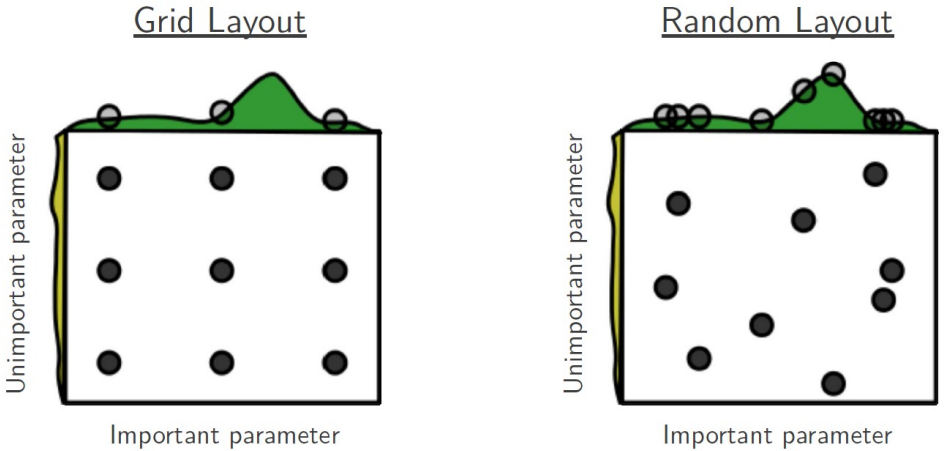
Contributeurs ReservoirPy
Paul BERNARD - Paul.Bernard@inria.fr
Xavier HINAUT - Xavier.Hinaut@inria.fr
Nathan TROUVAIN – Nathan.Trouvain@inria.fr
...

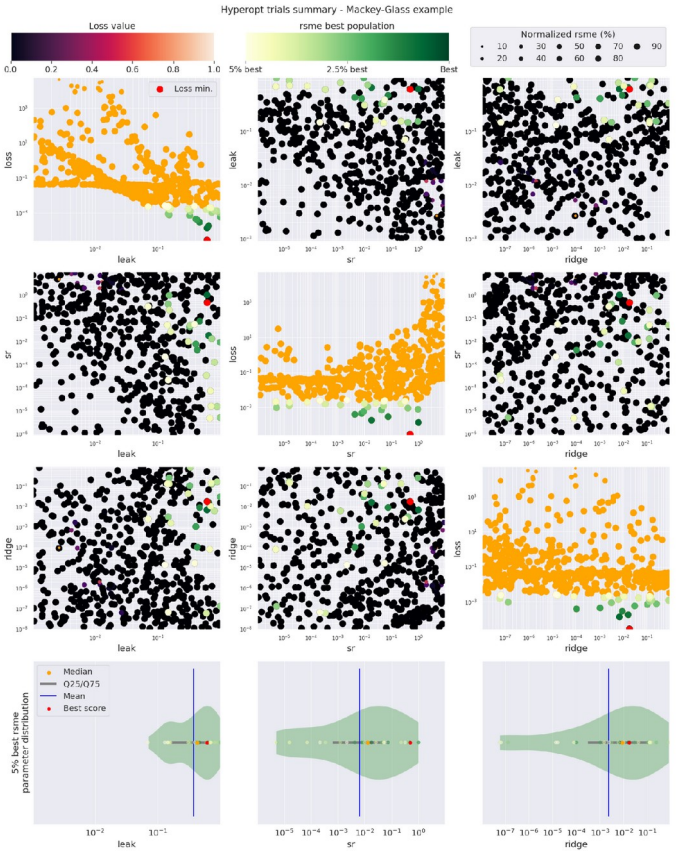
Compléments

Library	Language	Main dependency	Last activity	Package	Doc.	Tests	Off.	On.	Fb.	Model type	Deep
PyRCN	Python 3	Scikit-learn	Nov 2022	pip	✓	✓	✓	x	x	ESN	✓
EchoTorch	Python 3	PyTorch	Sep. 2021	pip	✓	✓	✓	x	x	ESN, Conceptors	✓
★Res.Comp.jl [3]	Julia	Julia	Sept 2023	Pkg	✓	✓	✓	x	x	ESN	x
Pytorch-esn	Python 3	Pytorch	Feb 2022	x	x	x	✓	x	x	ESN	✓
DeepESN	Matlab	Matlab	Feb. 2019	Matlab	✓	?	✓	x	x	ESN	✓
RCNet	C#	C#	Aug. 2021 (closed)	x	partial	x	✓	x	x	ESN, LSM	x
LSM	Python 3	Nest	Nov. 2020	x	x	x	✓	x	x	LSM	x
Oger	Python 2	mdp	2012 (obsolete)	x	x	✓	✓	✓	✓	LSM, ESN	✓
★`reservoirpy` **(this package)**	Python 3	Numpy	Sept 2023	pip	✓	✓	✓	✓	✓	ESN	✓

Table 1: Comparative table of some open source software for Reservoir Computing. This table might not be exhaustive. **Doc.** Complete documentation. **Off.** Offline learning strategies included. **On.** Online learning strategies included. **Fb.** Feedback and delayed connections. **Deep** The software allows the design of complex models where basic RC elements such as reservoirs and readouts can be stacked to form so-called "deep" networks.

Optimisation des hyper-paramètres





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

Publications avec ReservoirPy

HAL publications related to this software

HAL id	HAL citation
hal-03699931	Nathan Trouvain, Xavier Hinaut. reservoirpy: A Simple and Flexible Reservoir Computing Tool in Python. 2022. ⟨hal-03699931⟩
hal-02595026	Nathan Trouvain, Luca Pedrelli, Thanh Trung Dinh, Xavier Hinaut. ReservoirPy: an Efficient and User-Friendly Library to Design Echo State Networks. *ICANN 2020 - 29th International Conference on Artificial Neural Networks*, Sep 2020, Bratislava, Slovakia. ⟨hal-02595026v2⟩
hal-03533731	Nathan Trouvain, Xavier Hinaut. Reservoir Computing : théorie, intuitions et applications avec ReservoirPy. *Plate-Forme Intelligence Artificielle (PFIA)*, Jun 2021, Bordeaux, France. ⟨hal-03533731⟩
hal-03203318	Xavier Hinaut, Nathan Trouvain. Which Hype for my New Task? Hints and Random Search for Reservoir Computing Hyperparameters. *ICANN 2021 - 30th International Conference on Artificial Neural Networks*, Sep 2021, Bratislava, Slovakia. ⟨hal-03203318v2⟩
hal-03482372	Silvia Pagliarini, Arthur Leblois, Xavier Hinaut. Canary Vocal Sensorimotor Model with RNN Decoder and Low-dimensional GAN Generator. *ICDL 2021- IEEE International Conference on Development and Learning*, Aug 2021, Beijing, China. ⟨hal-03482372⟩
hal-03203374	Nathan Trouvain, Xavier Hinaut. Canary Song Decoder: Transduction and Implicit Segmentation with ESNs and LTSMs. *ICANN 2021 - 30th International Conference on Artificial Neural Networks*, Sep 2021, Bratislava, Slovakia. pp.71--82, ⟨10.1007/978-3-030-86383-8_6⟩. ⟨hal-03203374v2⟩
hal-03761440	Nathan Trouvain, Nicolas P. Rougier, Xavier Hinaut. Create Efficient and Complex Reservoir Computing Architectures with ReservoirPy. *SAB 2022 - FROM ANIMALS TO ANIMATS 16: The 16th International Conference on the Simulation of Adaptive Behavior*, Sep 2022, Cergy-Pontoise / Hybrid, France. ⟨hal-03761440⟩
tel-03946773	Xavier Hinaut. Reservoir SMILES: Towards SensoriMotor Interaction of Language and Embodiment of Symbols with Reservoir Architectures. Artificial Intelligence [cs.AI]. Université de Bordeaux (UB), France, 2022. ⟨tel-03946773⟩
hal-03628290	Subba Reddy Oota, Frédéric Alexandre, Xavier Hinaut. Cross-Situational Learning Towards Robot Grounding. 2022. ⟨hal-03628290v2⟩
hal-03780006	Xavier Hinaut, Nathan Trouvain. ReservoirPy: Efficient Training of Recurrent Neural Networks for Timeseries Processing. *EuroSciPy 2022 - 14th European Conference on Python in Science*, Aug 2022, Basel, Switzerland. ⟨hal-03780006⟩
hal-03945994	Nathan Trouvain, Xavier Hinaut. Reservoir Computing : traitement efficace de séries temporelles avec ReservoirPy. *Dataquitaine 2022*, Feb 2022, Bordeaux, France. ⟨hal-03945994⟩