



HAL
open science

Reservoir Computing: théorie, intuitions et applications avec ReservoirPy

Nathan Trouvain, Xavier Hinaut

► **To cite this version:**

Nathan Trouvain, Xavier Hinaut. Reservoir Computing: théorie, intuitions et applications avec ReservoirPy. Plate-Forme Intelligence Artificielle (PFIA), Jun 2021, Bordeaux, France. hal-03533731

HAL Id: hal-03533731

<https://inria.hal.science/hal-03533731v1>

Submitted on 19 Jan 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Reservoir Computing : théorie, intuitions et applications avec *ReservoirPy*

Nathan Trouvain^{1,2,3}, Xavier Hinaut^{1,2,3}

¹ INRIA Bordeaux Sud-Ouest, France

² Bordeaux INP, CNRS, UMR 5800, LaBRI

³ Université de Bordeaux, CNRS, UMR 5293, IMN

nathan.trouvain@inria.fr, xavier.hinaut@inria.fr

1 Description courte

Le *Reservoir Computing* (RC) est un paradigme d'apprentissage automatique sur des données séquentielles dans lequel les dynamiques temporelles non-linéaires sont centrales, et le coût computationnel réduit par rapport à d'autres réseaux de neurones récurrents. Ce tutoriel aborde la théorie du RC au travers de son application la plus connue : l'utilisation des *Echo State Networks* pour le traitement, la prédiction et la production de séries temporelles, à l'aide de l'outil Python *ReservoirPy*.

2 Description longue

Le temps joue un rôle central dans la plupart de nos interactions avec notre environnement : le langage nécessite de retenir des informations dans le temps pour être compris ; nos mouvements, notre physiologie, sont dépendants du temps qui passe et de notre adaptation à divers événements. Manipuler efficacement des données temporelles représente donc un défi très important pour le cerveau. Pour permettre à une machine comme un robot ou un ordinateur de répondre également à ce défi, de nombreuses méthodes ont été mises en œuvre, comme les réseaux de neurones artificiels récurrents. Ce tutoriel présente une coloration particulière des réseaux de neurones récurrents, le *Reservoir Computing* (RC).

Nous aborderons les aspects théoriques qui font du RC une technique de choix dans de nombreuses situations, de part sa légèreté en terme de calculs, sa robustesse face au bruit et sa plausibilité d'un point de vue biologique. Une mise en pratique démontrera les capacités des *Echo State Networks*, l'outil le plus commun de RC, à résoudre des tâches complexes comme la prédiction de séries chaotiques, le traitement du son ou du langage, tout en expliquant par l'expérience comment choisir leurs hyperparamètres. Ce tutoriel présentera également *ReservoirPy*, un logiciel en langage Python simple à prendre en main permettant de rapidement appliquer le RC à toutes sortes de données temporelles. *ReservoirPy* permettra d'approcher le RC sans connaissance préalable d'outils d'apprentissage automatique, au sein d'un environnement scientifique Python classique basé sur *Numpy/Scipy*.

3 Déroulé

Le tutoriel sera composé de deux exposés ouverts aux questions/réponses durant les présentations :

- *Reservoir Computing pour l'apprentissage rapide de séquences : théorie, intuitions et exemples d'applications*. Xavier HINAUT. 1h30 avec questions.
 - Introduction au RC et plausibilité biologique.
 - Application 1 : prédiction de séries temporelles
 - Application 2 : discrimination de sons
 - Application 3 : modéliser l'apprentissage du langage
 - Intuitions pratiques sur l'influence des hyperparamètres
 - Architectures dérivées et perspectives
- *ReservoirPy : prise en main d'une bibliothèque Python de RC flexible et polyvalente*. Nathan TROUVAIN. 1h30 avec questions. Travaux pratiques.
 - Introduction au RC avec *ReservoirPy* : approche de base des *Echo State Networks* (ESN).
 - Application du RC sur un jeu de données synthétique simple : la série chaotique de Mackey-Glass.
 - Exploration des espaces d'hyperparamètres définissant un réseaux de neurones en RC.
 - Exploration des capacités de génération des ESN.
 - Application du RC à des séries temporelles multivariées, comme la série de Lorenz.
 - Application du RC à des données terrain réelles.

Les auditeurs du deuxième exposé sont invité à suivre en testant la librairie *ReservoirPy* préalablement installée sur leur machine (seulement des dépendances standards sont nécessaires).

4 Public cible et pré-requis

Ce tutoriel s'adresse à un public sensibilisé à la théorie des réseaux de neurones artificiels. Aucun bagage théorique fort n'est cependant requis. Il peut en particulier concerner des étudiants (doctorants, master) et chercheurs intéressés par le traitement des séries temporelles comme le langage, les séries chaotiques, les événements géologiques et biolo-

giques, le son, la vidéo, les données boursières et économiques, etc. Il est recommandé pour les industriels traitant des données temporelles et cherchant des méthodes avec un coût computationnel ou un besoin en données moindre que les méthodes d'apprentissage profond classiques. Ce tutoriel vise en particulier à sensibiliser le public sur la facilité d'application du RC comme alternative à des techniques plus lourdes de traitement des séquences temporelles, mais peut également servir d'introduction et construire des intuitions sur le fonctionnement général des réseaux de neurones récurrents.

Concernant la mise en pratique durant une partie du second exposé, une connaissance basique de Python et de ses bibliothèques scientifiques (*Numpy* notamment) est nécessaire. Si les conférences doivent se faire par visioconférence uniquement les exposés seront probablement réduits en temps car les interactions sont plus difficile dans ce contexte.

5 Intérêt

Ce sujet présente un intérêt pour toutes les personnes, industriels, académiques ou étudiants, qui ont à traiter des données temporelles dans leurs applications ou modélisations. Par exemple, le RC peut-être utilisé pour :

- construire des modèles robustes permettant d'effectuer des tâches de prédiction sur des données où la prise en compte des dynamiques temporelles est importante ;
- produire des modèles capables d'expliquer certains processus cognitifs, comme la mémoire de travail ou la compréhension et la production du langage ;

Ce tutoriel vise donc à faire connaître cette technique à un public large souvent habitué à utiliser des outils plus classiques et souvent plus coûteux en données comme les réseaux de neurones *Long-Short Term Memory* pour traiter des séquences temporelles, voire à ne pas traiter cette dimension temporelle par soucis de simplicité. Il vise à démontrer que des solutions plus légères comme le RC peuvent être facilement mise en œuvre dans de nombreuses situations tout en offrant des perspectives de résultats similaires, et sans nécessiter de maîtriser d'outil dédié à l'apprentissage automatique comme *Tensorflow* ou *Pytorch*.

6 Informations sur les orateurs

- Xavier Hinaut est chargé de recherche à Inria Bordeaux Sud-Ouest depuis 2016, dans l'équipe Mnemosyne (Neurosciences Computationnelles). Il a soutenu son doctorat à l'Université de Lyon en 2013. Il utilise le *Reservoir Computing* dans ses recherches depuis une dizaine d'années, il enseigne sur ces thématiques depuis plusieurs années, à Bordeaux et à Hambourg (Allemagne) notamment. Ses thématiques de recherches sont liées à la modélisation du traitement, de l'apprentissage et de la production de séquences dans le cerveau, notamment le langage et les chants d'oiseaux. Il modélise également certaines fonctions cognitives de haut niveau (comme la mémoire de travail) et applique ses modèles de

langage aux interactions homme-robot dans la perspective d'ancrer le langage à travers l'interaction du robot avec le monde réel.

- Nathan Trouvain est ingénieur de recherche à Inria Bordeaux Sud-Ouest depuis 2020, dans l'équipe Mnemosyne également. Il est actuellement le principal développeur de la bibliothèque *ReservoirPy*. Il enseigne la pratique des techniques d'apprentissage automatique dans le master « Ingénieur spécialité Cognitive » de l'École Nationale Supérieure de Cognitive (ENSC).