Reservoir Computing : traitement efficace de séries temporelles avec ReservoirPy

Nathan Trouvain, nathan.trouvain@inria.fr, Inria, LaBRI, IMN, Bordeaux (Orateur)

Xavier Hinaut, xavier.hinaut@inria.fr, Inria, LaBRI, IMN, Bordeaux (Orateur)

Thématique : 6 – Intelligence Artificielle

Résumé

De la météo au langage, extraire l'information de flux de données est un enjeu primordial en Intelligence Artificielle. Le Reservoir Computing (RC) est particulièrement adapté pour bien prendre en compte ces dynamiques temporelles. C'est un paradigme d'apprentissage automatique sur des données séquentielles où un réseau de neurones artificiel n'est que partiellement entrainé. Un des intérêts majeurs de ces réseaux de neurones récurrents est leur coût computationnel réduit et la possibilité d'apprendre aussi bien "en-ligne" que "hors-ligne". Leurs applications sont très variées, qu'il s'agisse de prédiction/génération de séries chaotiques ou de discrimination de séquences audio, comme la reconnaissance de chants d'oiseaux. Nous verrons les aspects théoriques du RC : comment ce "réservoir de calculs" fonctionne grâce à des projections aléatoires en grandes dimensions, et s'apparente ainsi à des Machines à Vecteurs Supports (SVM) temporelles.

Nous présenterons également ReservoirPy : une bibliothèque Python à la fois simple et efficace basée sur la pile logicielle scientifique de Python (Numpy, Scipy, Matplotlib). ReservoirPy met l'accent sur les Echo State Networks (ESN), l'instance la plus connue de RC. Elle permet la conception d'architectures développées dans la littérature, allant des plus classiques aux plus complexes. ReservoirPy embarque plusieurs règles d'apprentissage (online et offline), une implémentation distribuée de l'entraînement des ESN, la possibilité de créer des réseaux hiérarchiques comportant des boucles de feedback complexes, et des outils d'aide à l'optimisation des hyperparamètres. La documentation, des tutoriels, des exemples et des jeux de données sont disponibles sur la page GitHub de ReservoirPy : github.com/reservoirpy

Mots clés : *Séries temporelles, Réseaux de neurones récurrents, Prédiction, Génération, Classification, Reconnaissance de la parole*

1. Introduction

Appréhender la dimension temporelle d'un jeu de données est un problème particulièrement complexe en apprentissage machine. Elle est cependant primordiale dans de nombreux domaines, comme le traitement du langage naturel ou la prédiction de séries temporelles. En particulier, appliquer à des séries temporelles les techniques fructueuses du *deep learning* impose une complexité algorithmique et mathématique grandissante, des LSTM [4] aux Transformers [8]. Les réseaux de neurones récurrents mettent ainsi à profit une version modifiée de l'algorithme de descente du gradient d'erreur pour apprendre des relations

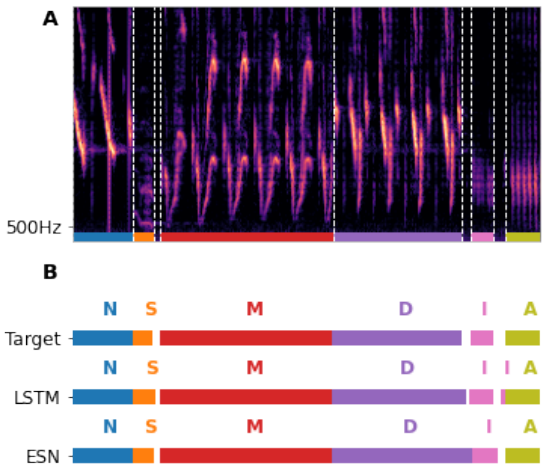
temporelles complexes. Ces techniques sont souvent très coûteuses en calculs et en mémoire et sont en général moins robustes face au bruit ou à des comportements chaotiques, observés dans de nombreux problèmes de la recherche comme de l'industrie.

Nous nous proposons de présenter le *Reservoir Computing* [5], une technique de *machine learning* permettant de tirer parti de réseaux de neurones récurrents en utilisant des mécanismes d'apprentissage plus rapides et plus simples que ceux utilisés en *deep learning.* Le Reservoir Computing offre de nombreuses perspectives en traitement robuste des séries temporelles, utile pour la robotique, le traitement de données audio, ou la prédictions de systèmes chaotiques par exemple.

2. Méthodologie

Le Reservoir Computing est une méthode tirant parti d'un réseau de neurones récurrents aléatoirement construit pour produire une grande variété de représentations des données en entrée, en y intégrant une dimension temporelle. L'apprentissage n'a pas lieu dans la partie récurrente du réseau, appelée *reservoir*, mais uniquement à sa sortie, où une unique couche de neurones, appelée *readout,* est chargée de produire le signal désiré. Ainsi, il est possible d'effectuer l'apprentissage à l'aide d'une simple régression linéaire [5].

Nous présenterons des applications du Reservoir Computing à des tâches considérées comme complexes dans la littérature : prédiction de séries temporelles chaotiques, classification de sons… Nous introduirons également les concepts clés de cette technique, et quelques exemples issus de la littérature, au travers notamment des *Echo State Networks* (ESN), une technique courante de Reservoir Computing. Nous présenterons également le bibliothèque Python *ReservoirPy* [7], développée au sein de l'Inria pour diffuser et appliquer



ces techniques.

Figure 1 *- Exemple de résultats obtenus à l'aide de techniques de Reservoir Computing (ESN) appliqué à une tâche d'annotation de chant de canaris [6]. La tâche consiste à identifier chaque type de motif sonore dans le chant (tâche similaire à la reconnaissance de la parole chez l'humain). La figure A montre le type de données traitées (ici, les composantes spectrales du son). La figure B montre l'adéquation entre la séquence de motifs identifiés par le reservoir (ESN) et la séquence attendue (Target). On y voit également la comparaison avec les performances d'un LSTM.*

3. Originalité / perspective

Quoique plus discrète et à contre courant des politiques d'augmentation du nombre de paramètres des modèles d'apprentissage profond rencontrés de nos jours, le Reservoir Computing offre une alternative pertinente et économique (voire écologique) aux réseaux de neurones récurrents plus classiques. Le Reservoir Computing est également appliqué avec succès dans sa version *hardware,* où des composants neuromorphiques ou optiques [1] remplacent les transistors électroniques pour produire un *reservoir* physique.

Dans un autre registre, le Reservoir Computing offre également de nombreuses perspectives en neurosciences computationnelles, où certains indices semblent indiquer que le cerveau possède des structures neuronales semblables lui permettant de traiter la dimension temporelle de son environnement [2, 3]. C'est dans ce cadre que les membres de l'équipe Mnemosyne l'utilisent, où cette technique permet de modéliser divers processus cognitifs et sensorimoteurs.

Références

[1] Brunner, D., Soriano, M. C., & Van der Sande, G. (Eds.). (2019). *Photonic Reservoir Computing: Optical Recurrent Neural Networks*. De Gruyter.

[2] Dominey, P. F. (2021). Cortico-Striatal Origins of Reservoir Computing, Mixed Selectivity, and Higher Cognitive Function. In K. Nakajima & I. Fischer (Eds.), *Reservoir Computing: Theory, Physical Implementations, and Applications* (pp. 29–58). Springer.

[3] Enel, P., Procyk, E., Quilodran, R., & Dominey, P. F. (2016). Reservoir Computing Properties of Neural Dynamics in Prefrontal Cortex. *PLOS Computational Biology*, *12*(6), e1004967. https://doi.org/10.1371/journal.pcbi.1004967

[4] Gers, F. A., Schmidhuber, J., & Cummins, F. (1999). Learning to forget: Continual prediction with LSTM. *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, *2*, 850–855 vol.2.

[5] Jaeger, H. (2001). The" echo state" approach to analysing and training recurrent neural networks-with an erratum note'. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, *148*

[6] *Trouvain, N., & Hinaut, X. (2021). Canary Song Decoder: Transduction and Implicit Segmentation with ESNs and LTSMs. In I. Farkaš, P. Masulli, S. Otte, & S. Wermter (Eds.), Artificial Neural Networks and Machine Learning – ICANN 2021 (pp. 71–82). Springer International Publishing*

[7] Trouvain, N., Pedrelli, L., Dinh, T. T., & Hinaut, X. (2020). ReservoirPy: An Efficient and User-Friendly Library to Design Echo State Networks. In I. Farkaš, P. Masulli, & S. Wermter (Eds.), *Artificial Neural Networks and Machine Learning – ICANN 2020* (pp. 494–505). Springer International Publishing.

[8] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All you Need. *Advances in Neural Information Processing Systems*, *30*.