Comparaison des RNN, LSTM et des GRU dans le cadre du pos-tagging de texte

NOELE Elodie, TABOGA Vincent, WOZNIAK Anne-Laure

[elodie.noele@polymtl.ca](mailto:elodie.noele@polymtl.ca)

[vincent.taboga@polymtl.ca](mailto:vincent.taboga@polymtl.ca)

[anne-laure.wozniak@polymtl.ca](mailto:anne-laure.wozniak@polymtl.ca)

Résumé

Ce rapport vise à étudier les performances de différents modèles de RNN, de GRU et de LSTM dans le cadre du pos-tagging de texte. Le pos-tagging de texte consiste à associer chaque mot d’un texte à une entité grammaticale bien définie. Parmi les structures testées on retrouve les modèles vanilla RNN, RNN à deux couches et bidirectionnels, les LSTM simples, bidirectionnels et à plusieurs couches ainsi que les vanilla GRU, les GRU bidirectionnels et les GRU à deux couches. Nous avons aussi ajouté pour certaines architectures un Dropout. Les tests réalisés montrent que les LSTM et les GRU performent mieux pour le POS tagging, comme on pouvait si attendre car c’est une opération qui nécessite de mémoriser des séquences. Il n’y a cependant pas de différences significatives de performance entre les GRU et les LSTM.

Introduction*[[1]](#footnote-1)*

Le POS-Tagging de texte est une étape préliminaire à la traduction de texte d’une langue à l’autre. C’est une des tâches courantes du traitement du langage. Cette technique peut aussi être utilisée pour identifier les mots clés d’une phrase sur un moteur de recherche et ainsi mieux cibler la demande d’un utilisateur.

Le POS-Tagging est un domaine couramment étudié en intelligence artificielle, puisqu’il donne généralement de bien meilleurs résultats que le pos-tagging avec des algorithmes « classiques » comme celui proposé par la bibliothèque NLTK de Python.

Dans cet article, nous proposons différents types de réseaux neuronaux récurrents que nous évaluons sur une expérience de POS-tagging : Les vanilla RNN, les RNN à deux couches, les RNN bidirectionnel, les LSTM simples, les LSTM à deux couches, les LSTM bidirectionnel, les GRU simples, les GRU bidirectionnel et les GRU à deux couches. Nous évaluons ces réseaux le corpus de texte « brown » de la librairie NLTK de python.

L’article est donc organisé de la manière suivante : en premier lieu nous présentons quelques travaux antérieurs, puis nous explicitons les modèles retenus pour notre étude. Nous décrivons ensuite les expériences menées sur ces modèles ainsi que nos résultats avant de discuter des améliorations possibles de notre travail.

Travaux antérieurs

Les travaux antérieurs à notre projet font état de l’utilisation de réseaux neuronaux de type LSTM pour le pos-tagging de texte. En effet, ce sont ces réseaux avec les GRU qui montrent les meilleurs résultats dans le cadre du pos-tagging. Plus particulièrement, les réseaux neuronaux de type LSTM bidirectionnel couplés à des méthodes statistiques telle que le CRF (Wei Xu et al.) obtiennent de bien meilleurs résultats que les GRU simples ou les LSTM bidirectionnel (Xuezhe Ma and Eduard Hovy)

A compléter si nécessaire

This Is an Example of a Figure Caption. Use of the figure caption style is required.

Modèles retenus

RNN

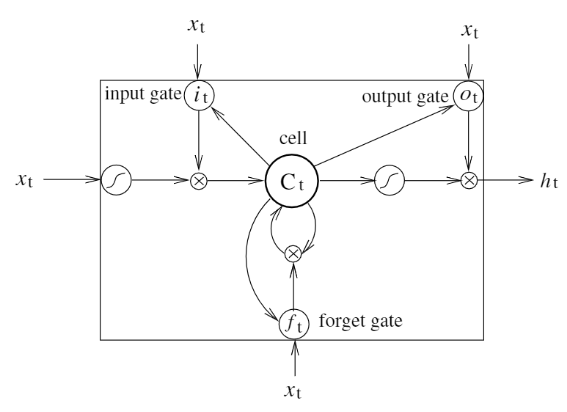
Les réseaux neuronaux récurrents sont des réseaux dont les connexions forment des boucles. Chaque neurone a ainsi en entrée l’état du système à l’instant t ainsi que les informations des cellules au temps t-1. Des tels réseaux possèdent un état interne, ce qui en font des candidats propices pour des tâches de reconnaissance séquentielle comme le traitement du langage naturel. Au sein d’un réseau des opérations matricielles sont appliqués sur l’observation du système (noté o) et sur les états cachés (notés h) de l’état passé. Les résultats de ces opérations sont ensuite passés en entrée d’une fonction d’activation (ici log-softmax).

Figure 1: Représentation d'un cellule LSTM

Avec :

, la matrice du réseau

, la matrice des poids de l’unité cachée

, la matrice des poids de la sortie

, le terme de biais de l’unité cachée

, le terme de biais de la sortie

LSTM

Les réseau Long Short Term Memory (LSTM) sont des RNN dont les cellules ont une structure particulière. Nous avons vu précédemment que deux informations principales sont contenues dans une même variable au sein d’un réseau de neurones récurrents : la valeur retournée au réseau pour faire des prédictions et la séquence de données utilisées jusqu’ici. Une cellule LSTM va séparer stocker ces deux informations dans deux variables séparées : une variable de mémoire c et une variable de l’état de la cellule h. La mise à jour de ces variables se fait à l’aide de trois portes : entrée sortie et de réinitialisation de la mémoire (en anglais input gate, ouput gate et forget gate). Une telle architecture est plus complexe mais présente de très bons résultats pour beaucoup d’applications. Elles permettent notamment de résoudre les problèmes de gradient qui explose ou qui disparait lors de la rétropropagation dans le temps. L’information gardée en mémoire ne sera pas altérée au fur et à mesure des itérations. Une telle architecture est représentée sur le figure 1. Les fonctions classiquement utilisées pour les LSTM, et que nous avons retenues pour nos modèles, sont les suivantes :

Avec comme variables i : input gate, o : ouput gate, f : forget gate, j : new memory content, c : update memory

GRU

Les réseaux à « Gated Recurrent Unit » (GRU) sont basés sur le même principe que les LSTM. Ils ont cependant une architecture un peu plus simple car ils ne possèdent que deux portes : entrée et sortie. Il n’y a donc pas de moyen de contrôler l’information qui doit être gardé en mémoire. Ils permettent tout de même d’éviter les difficultés d’apprentissage liées à l’explosion ou la disparition du gradient. En contrepartie de la moins grande flexibilité ils sont plus faciles à entraîner car ils contiennent moins de paramètres. On notera de plus que le gain d’une troisième porte sur les performances n’est pas significatif pour beaucoup de taches, nous vérifierons plus tard si c’est le cas pour du POS tagging.

Réseau à deux couches

Les réseaux à deux couches sont un type de réseau neuronal où l’information circule de la couche d’entrée, vers les couches cachées puis la couche de sortie. C’est un réseau de type feedforward. Comparé au réseau neuronal dit “simple”, les réseaux à deux couches, couplé à la rétropropagation du gradient, permettent aux réseaux neuronaux d’effectuer des taches plus abstraites, comme l’exercice du POS-Tagging qui est un problème non linéaire.

Il aurait été possible d’utiliser des réseaux à plus de deux couches (cf. Figure 2), nos expériences nous ont montré que les résultats n’étaient pas forcément meilleurs, notamment dans le cas des RNN où nous pourrions faire face à un problème de gradient explosif ou évanescent.

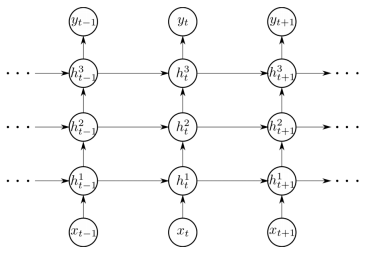


Figure 2 - Réseau RNN à 3 couches cachées : x l’entrée, h l’état de la couche cachée, y la sortie.

Réseau bidirectionnel

Dans le cadre du POS-tagging, il peut être bénéfique de savoir quels sont les mots avant et quels sont les mots après le mot que l’on souhaite classifier. C’est pourquoi l’utilisation des réseaux neuronaux bidirectionnels se prête volontiers à cette tâche.

Dans un réseau bidirectionnel, les informations passées et futures sont mémorisées dans deux sous-réseaux avant et arrière. Il fonctionne en trois étapes :

* Le réseau arrière est utilisé pour déterminer les tags et les états de la couche cachée (s’il y en a, par exemple pour les LSTM)
* Ensuite, ces tags et ces états de la couche caché sont utilisés comme données d’entrée pour le réseau avant.
* Enfin, le modèle global parcourt la séquence complète en utilisant les tags et les états des réseaux avant et arrière.

L’avantage de ce type de réseau est qu’il permet d’avoir une « compréhension » plus fine de la phrase à tagger et donc d’éviter les erreurs. Cette procédure a été définie dans l’article plus précisément dans l’article de de **Schuster & Paliwal, 1997**.

Expériences menées

Pour tester nos réseaux neuronaux récurrents, nous avons choisi d’utiliser la bibliothèque NLTK de Python qui propose un corpus de texte Brown dont le pos-tagging a déjà été réalisé. Ce corpus de texte possède plus de 50 000 phrases issues de la littérature anglaise. Nous avons donc fait le choix de tester nos modèles sur ces corpus en prenant **un ensemble d’entraînement de 5000 phrases**, **un ensemble de validation de 2000 phrases** et **un ensemble de tests de 1000 phrases**. Toutes ces phrases ont des longueurs aléatoires.

En outre, pour avoir une représentation plus fine de la performance de nos modèles nous avons choisi de tester nos modèles sur deux autres ensembles, le premier avec 1000 phrases courtes (moins de 5 entités grammaticales) et le second avec 1000 phrases longues (plus de 40 entités grammaticales). Nous pouvons ainsi comparer les performances de nos architectures selon que la séquence à mémoriser soit longue ou courte.

Conformément à ce qui est généralement fait en POS tagging, la performance sur les jeux de tests est évaluée avec le score f1, qui pénalises certaines erreurs plus que d’autres. Les résultats sont présentés dans le tableau si dessous :

Résultats

|  |  |
| --- | --- |
| **Architecture** | **F1score Test set** |
| **Vanilla RNN** | 75,46 % |
| **RNN à deux couches** | 75,00 % |
| **RNN bidirectionnel** | 78,11 % |
| **LSTM simples** | 82,03% |
| **LTSM à deux couches** | 85,02% |
| **LSTM bidirectionnel** | 85,12% |
| **LSTM à deux couches bidirectionnel** | 85,88% |
| **GRU simple** | 75,36 % |
| **GRU bidirectionnel** | 75,59 % |
| **GRU à deux couches** | 80,92 % |

Si nous comparons nos résultats par classes de modèles, on peut retenir les RNN bidirectionnel avec un résultat de 78,11 %, les LSTM bidirectionnel à deux couches avec un résultat de 85,88%, les GRU à deux couches avec un résultat de 80,92%. Les réseaux bidirectionnels sont naturellement plus performants, car ils permettent une meilleure « compréhension » de l’environnement des mots au sein d’une phrase comme nous l’expliquions dans une des sections précédentes.

A complexité de réseaux équivalente, i.e. en ne considérant que les réseaux bidirectionnels, ou uniquement les réseaux à deux couches, etc., on s’aperçoit que les LSTM sont vraiment les plus performants, quelle que soit la catégorie.

Discussion

Comparaison avec l’existant

Dans la littérature sur le pos-tagging de texte avec les réseaux neuronaux, la majorité des réseaux qui obtiennent les meilleurs scores sont de types LSTM. Sur un corpus similaire au notre, Wei Xu et al. obtiennent un score de près de 98% en utilisant un LSTM-CRF, c’est-à-dire un LSTM couplé à des champs aléatoires conditionnels, permettant de mieux prendre en compte la contribution des différentes mos d’une phrase.

Dans les autres articles du même acabit, les réseaux obtenant les meilleurs résultats sont ceux de type LSTM bidirectionnel et GRU simple.

Dans notre cas, on retrouve globalement ces résultats avec les réseaux LSTM bidirectionnels qui performent bien mieux que les autres types de réseaux. Nous trouvons cependant des résultats qui différent des articles que nous citons en ce qui concerne les GRU. Dans nos expériences, ces derniers sont plus intéressants lorsqu’ils comportent deux couches cachées.

Limites de notre approche

Notre approche se limite à des modèles de réseaux neuronaux « simples », cependant nous avons réalisé des tests avec des modèles de type RNN, LSTM et GRU à 10 couches et notre F1-score accuracy ne dépassait les 30 %.

Nous aurions pu tester nos modèles sur des ensembles plus grands et sur plus d’ensembles, cependant, les contraintes ont été le manque de temps, puisque chacun des modèles met environ 3 heures à s’entraîner voire plus dans le cas des modèles de type LTSM. Aussi, nous avons été limités par la performance de nos ordinateurs.

Conclusion

Explication de pourquoi les RNN sont moins bons que les LTSM et les GRU

Les modèles de type purement RNN ont des performances en général moins bonnes que les modèles utilisant des LSTM. Ceci est notamment lié au fait que les RNN requièrent un corpus bien plus grand que les LSTM afin d’être correctement entrainé. Dans ce projet, nous utilisons un ensemble d’entrainement de 5000 phrases, ce qui nous permet d’avoir tout de même de bons résultats. Il faudrait cependant augmenter encore la taille de l’ensemble pour déterminer à quel moment un RNN performe au même niveau qu’un réseau LSTM.

Références

Juan Antonio Pérez-Ortiz, Mikel L. Forcada, *Part-of-Speech Tagging with Recurrent Neural Networks*, Departament de e Llenguatges i Sistemes Informatics, Universitat d’Alacant, E-03071 Alacant, Spain

Ashish Vaswani, Yonatan Bisk, University of Southern California, Kenji Sagae, Kitt.ai, Ryan Musa, University of Illinois at Urbana-Champaign, *Supertagging with LSTMs*

Xuezhe Ma and Eduard Hovy*, End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF,* Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Zhiheng Huang, Wei Xu, Kai Yu, *Bidirectional LSTM-CRF Models for Sequence Tagging*, Baidu research

Raj Nath Patel, Prakash B. Pimpale, Sasikumar M., KBCS, CDAC Mumbai, *Recurrent Neural Network based Part-of-Speech tagger for Code-Mixed*, Nov 2016.

M. Schuster and K. K. Paliwal*, Bidirectional recurrent neural networks*, in IEEE Transactions on Signal Processing*,* vol. 45, no. 11, pp. 2673-2681, Nov 1997.

Ian H.Witten, Eibe Frank, Mark A. Hall, Christopher J. Pal, Data Mining (Fourth Edition) *Pratical Machine Learning Tools and Techniques*, Morgan Kaufmann, 2017.

1. Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved. [↑](#footnote-ref-1)