

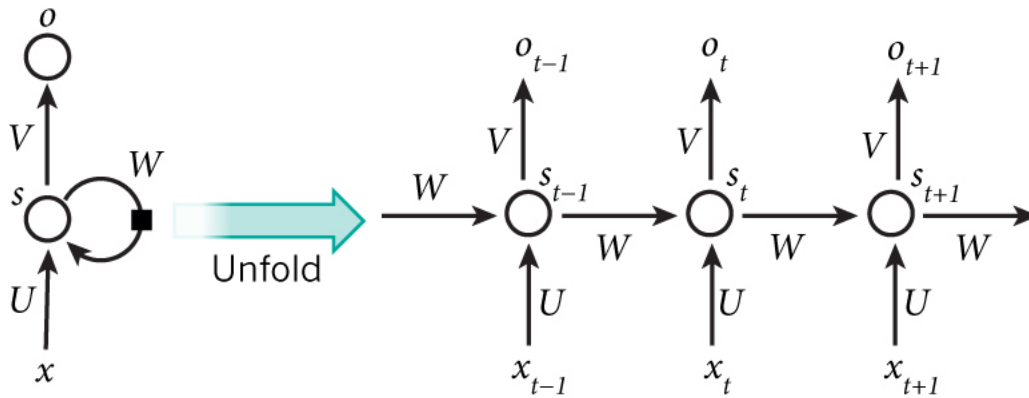
RNN - Recurrent Neural Network

Bùi Quý Bảo

July 3, 2019

1 Giới thiệu về RNN

RNN - Recurrent Neural Network là một thuật toán được chú ý khá nhiều trong thời gian gần đây bởi các kết quả trong lĩnh vực xử lý ngôn ngữ tự nhiên. Ý tưởng chính của RNN là sử dụng chuỗi các thông tin. Trong các mạng neural truyền thống, các đầu vào và đầu ra là độc lập với nhau (không liên kết thành chuỗi). Tuy nhiên trong một số bài toán phức tạp, đòi hỏi thông tin về những thành phần phía trước như bài toán đoán từ tiếp theo xuất hiện trong câu thì mô hình truyền thống này không đáp ứng được. RNN được ra đời với ý nghĩa chúng thực hiện các tác vụ cho các phần tử của một chuỗi với đầu ra phụ thuộc vào thông tin nhận được từ các bước trước. Hay nói cách khác RNN có khả năng nhớ thông tin. Trên lý thuyết, RNN có khả năng sử dụng được thông tin của đoạn văn bản dài, tuy nhiên thực tế nó chỉ có khả năng nhớ một vài bước trước (Short-term memory) mà thôi.



Hình 1: RNN and the unfolding in time of the computation involved in its forward computation. Nguồn: Nature

Mô hình miêu tả ở Hình 1, cho phép triển khai nội dung của một RNN với cấu trúc chuỗi tuần tự. Ví dụ ta có một câu gồm 4 từ "MaSSP thật thú vị" thì RNN sẽ được triển khai gồm 4 tầng nơ-ron tương ứng với mỗi từ một tầng. Việc tính toán cụ thể bên trong được thực hiện như sau:

- x_t là đầu vào tại bước t . Nếu sử dụng one-hot encode thì x_t là vector one-hot tương ứng với từ nằm trong từ điển.

- s_t là trạng thái ẩn tại bước t . Nó chính là bộ nhớ của mạng, xuyên suốt, tuần tự liên kết với nhau. s_t được tính toán dựa trên đầu vào tại bước đó và trạng thái ẩn của bước trước

$$s_t = f(U \cdot x_t + W \cdot s_{t-1})$$
Hàm f thường là một hàm non-linear như hàm tanh, hay ReLU. Để thực hiện được phép toán cho phần tử ẩn đầu tiên s_0 ta thường thêm vào phần tử s_{-1} , và khởi tạo cho nó giá trị bằng 0.
- o_t là đầu ra tại bước t . Kết quả o_t thường là vector xác suất các từ trong danh sách từ vựng của mình

$$o_t = softmax(V \cdot s_t)$$

2 Ứng dụng của RNN

2.1 Mô hình hóa ngôn ngữ và sinh văn bản

Mô hình ngôn ngữ cho phép ta dự đoán được xác suất của một từ xuất hiện sau một chuỗi các từ ở phía trước nó. Điểm lý thú là từ đó chúng ta có thể xây dựng được một mô hình tự sinh từ cho phép máy tính có thể tạo ra các văn bản mới từ tập mẫu và xác suất đầu ra của mỗi từ. Do đó, tùy thuộc vào mô hình ngôn ngữ mà chúng ta có thể tạo ra nhiều văn bản khác nhau.

Trong mô hình ngôn ngữ, đầu vào thường là một chuỗi các từ và đầu ra là một chuỗi các từ dự đoán được. Khi huấn luyện mạng, ta sẽ gán $o_t = x_{t+1}$ vì ta muốn đầu ra tại bước t chính là từ tiếp theo của câu.

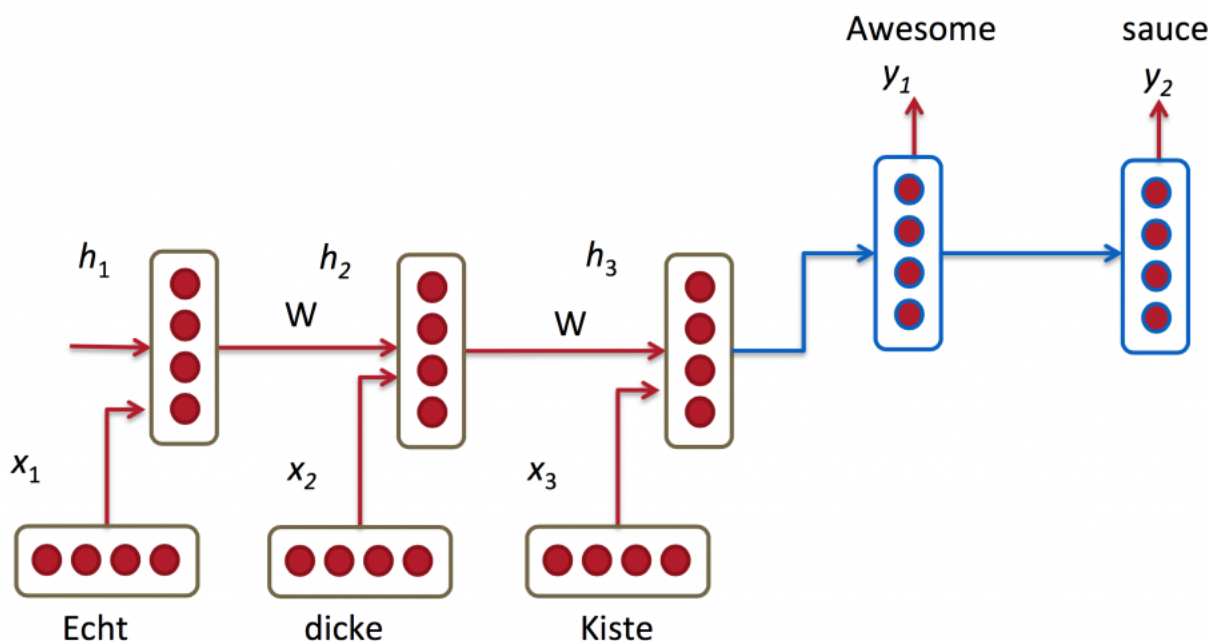
Một số nghiên cứu về mô hình hóa và sinh văn bản ứng dụng RNN:

- Recurrent neural network based language model [3].
- Extensions of Recurrent neural network based language model [4].
- Generating Text with Recurrent Neural Networks [5].

2.2 Machine Translation

Machine Translation (Dịch máy) tương tự như mô hình hóa ngôn ngữ ở điểm đầu vào là chuỗi các từ trong ngôn ngữ nguồn cần dịch. Đầu ra sẽ là chuỗi các từ trong ngôn ngữ đích. Điểm khác nhau so với phương pháp ở trên là chúng ta chỉ xử lý sau khi đã xem xét toàn bộ chuỗi đầu vào. Vì từ dịch đầu tiên của câu cần phải có đầy đủ thông tin đầu vào cần dịch mới có thể suy luận, dịch chính xác được. Một số nghiên cứu về machine translation ứng dụng RNN:

- A Recursive Recurrent Neural Network for Statistical Machine Translation [2].
- Sequence to Sequence Learning with Neural Networks [6].
- Joint Language and Translation Modeling with Recurrent Neural Networks [1].



Hình 2: RNN for Machine Translation. Nguồn: CS224 Stanford

2.3 Huấn luyện RNN

Huấn luyện mạng RNN cũng tương tự như các mạng neural truyền thống khác, tuy nhiên giải thuật lan truyền ngược (backpropagation) phải thay đổi một chút. Vì các tham số trong mạng RNN được sử dụng chung cho tất cả các bước trong mạng (U, W, V) do đó đạo hàm tại mỗi đầu ra của các lớp trong RNN không chỉ phụ thuộc vào các tính toán tại bước đó, mà còn phụ thuộc vào các bước trước. Do đó để tính đạo hàm tại $t = 3$ ta phải tính lan truyền ngược cả 2 bước phía trước rồi cộng tổng đạo hàm của chúng lại với nhau. Việc tính đạo hàm kiểu này được gọi là lan truyền ngược liên hồi (BPTT - Backpropagation Through Time).

Một lưu ý của việc huấn luyện RNN là khi các bước phụ thuộc càng xa thì càng dễ xuất hiện hiện tượng đạo hàm bị triệt tiêu (vanishing). LSTM đã được nghiên cứu để giải quyết vấn đề này của RNN truyền thống.

Tài liệu

- [1] Michael Auli et al. “Joint Language and Translation Modeling with Recurrent Neural Networks”. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*. 2013, pp. 1044–1054. URL: <http://aclweb.org/anthology/D/D13/D13-1106.pdf>.

- [2] Shujie Liu et al. “A Recursive Recurrent Neural Network for Statistical Machine Translation”. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*. 2014, pp. 1491–1500. URL: <http://aclweb.org/anthology/P/P14/P14-1140.pdf>.
- [3] Tomas Mikolov et al. “Recurrent neural network based language model”. In: *INTER-SPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*. 2010, pp. 1045–1048. URL: http://www.isca-speech.org/archive/interspeech%5C_2010/i10%5C_1045.html.
- [4] T. Mikolov et al. “Extensions of recurrent neural network language model”. In: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. May 2011, pp. 5528–5531. DOI: 10.1109/ICASSP.2011.5947611.
- [5] Ilya Sutskever, James Martens, and Geoffrey E. Hinton. “Generating Text with Recurrent Neural Networks”. In: *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*. 2011, pp. 1017–1024. URL: https://icml.cc/2011/papers/524%5C_icmlpaper.pdf.
- [6] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. “Sequence to Sequence Learning with Neural Networks”. In: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. 2014, pp. 3104–3112. URL: <http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks>.