TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**PHÂN TÍCH XÁC SUẤT VÀ GIẢI THUẬT NGẪU NHIÊN**

**TIẾP CẬN DỊCH MÁY BẰNG**

**PHƯƠNG PHÁP ENCODER-DECODER**

*Người hướng dẫn*: **Thầy Nguyễn Chí Thiện**

*Người thực hiện*: **Nguyễn Duy Hàn Lâm (MSHV: 196005004)**

Khoá  **: 2019**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**PHÂN TÍCH XÁC SUẤT VÀ GIẢI THUẬT NGẪU NHIÊN**

**TIẾP CẬN DỊCH MÁY BẰNG**

**PHƯƠNG PHÁP ENCODER-DECODER**

*Người hướng dẫn*: **Thầy Nguyễn Chí Thiện**

*Người thực hiện*: **Nguyễn Duy Hàn Lâm (MSHV: 196005004)**

Khoá  **: 2019**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020**

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi và được sự hướng dẫn của thầy Nguyễn Chí Thiện. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 05 tháng 04 năm 2020*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Nguyễn Duy Hàn Lâm*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

MỤC LỤC

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN ii](#_Toc35716930)

[MỤC LỤC 1](#_Toc35716931)

[CHƯƠNG 1: GIỚI THIỆU VỀ MẠNG NEURAL 4](#_Toc35716932)

[1.1 Perceptron 5](#_Toc35716933)

[1.2 Sigmoid neuron 8](#_Toc35716934)

[1.3 Cho mô hình học bằng kĩ thuật gradient descent 10](#_Toc35716935)

[1.3.1 Training dataset, testing dataset 10](#_Toc35716936)

[1.3.2 Gradient descent 11](#_Toc35716937)

[1.3.3 Áp dụng gradient descent vào mạng neural 13](#_Toc35716938)

[1.3.4 Cải tiến gradient descent 14](#_Toc35716939)

[1.4 Tóm tắt chương 1 15](#_Toc35716940)

[CHƯƠNG 2: GIẢI THUẬT BACKPROPAGATION 16](#_Toc35716941)

[2.1 Matrix-based algorithm 16](#_Toc35716942)

[2.2 Phép nhân Hadamard 17](#_Toc35716943)

[2.3 Bốn yếu tố cơ bản đằng sau giải thuật backpropagation 17](#_Toc35716944)

[2.3.1 Yếu tố thứ 1: Biểu thức tính mức độ sai số ở output layer (lớp cuối cùng) 18](#_Toc35716945)

[2.3.2 Yếu tố thứ 2: Biểu thức tính mức độ sai số ở các lớp ẩn 19](#_Toc35716946)

[2.3.3 Yếu tố thứ 3: Biểu thức tính mức độ sai số của phần bù đại số 19](#_Toc35716947)

[2.3.4 Yếu tố thứ 4: Biểu thức tính mức độ sai số của các trọng số 20](#_Toc35716948)

[2.4 Một số phép tính thay thế 20](#_Toc35716949)

[2.5 Cách chứng minh các biểu thức (27), (30), (31), (32) 21](#_Toc35716950)

[2.5.1 Chứng minh biểu thức (27) 21](#_Toc35716951)

[2.5.2 Chứng minh biểu thức (30) 21](#_Toc35716952)

[2.5.3 Chứng minh biểu thức (31) 22](#_Toc35716953)

[2.5.4 Chứng minh biểu thức (32) 23](#_Toc35716954)

[2.6 Giải thuật backpropagation 25](#_Toc35716955)

[2.7 Một góc nhìn khác 26](#_Toc35716956)

[2.8 Tổng kết chương 2 26](#_Toc35716957)

[CHƯƠNG 3: THỰC HÀNH LẬP TRÌNH 27](#_Toc35716958)

[3.1 Khởi tạo mạng neural 27](#_Toc35716959)

[3.2 Stochastic gradient descent 28](#_Toc35716960)

[3.3 Chạy mạng neural 32](#_Toc35716961)

[3.4 Tổng kết 34](#_Toc35716962)

[TÀI LIỆU THAM KHẢO 35](#_Toc35716963)

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 1.1. Cấu trúc cơ bản của 1 neural [1] 4](#_Toc35716964)

[Hình 1.2. Cấu trúc chi tiết 1 tầng của perceptron [2] 5](#_Toc35716965)

[Hình 1.3. Cấu trúc thu gọn của perceptron 1 tầng [3] 6](#_Toc35716966)

[Hình 1.4. Cấu trúc thu gọn của perceptron nhiều tầng [3] 7](#_Toc35716967)

[Hình 1.5. Minh họa về việc thay đổi [3] 8](#_Toc35716968)

[Hình 1.6. Đồ thị của hàm sigmoid [3] 9](#_Toc35716969)

[Hình 3.1. Hàm khởi tạo cho đối tượng mạng neural 27](#_Toc35716970)

[Hình 3.2. Khởi tạo đối tượng network 27](#_Toc35716971)

[Hình 3.3. Hàm tính sigmoid và đạo hàm của sigmoid kèm theo 28](#_Toc35716972)

[Hình 3.4. Hàm thực thi chiều thuận của mạng neural 28](#_Toc35716973)

[Hình 3.5. Hàm thực hiện huấn luyện cho mạng neural 29](#_Toc35716974)

[Hình 3.6. Hàm update các giá trị trọng số, bias 29](#_Toc35716975)

[Hình 3.7. Hàm giải thuật backpropagation 30](#_Toc35716976)

[Hình 3.8. Hàm tính đạo hàm của loss function 31](#_Toc35716977)

[Hình 3.9. Hàm tính giá trị mất mát 31](#_Toc35716978)

[Hình 3.10. Đồ thị minh họa cho tập dữ liệu đầu vào 32](#_Toc35716979)

[Hình 3.11. Khởi tạo đối tượng mạng neural 32](#_Toc35716980)

[Hình 3.12. Chuẩn hóa và phân tập tập dữ liệu thành tập train và tập test 33](#_Toc35716981)

[Hình 3.13. Chạy và huấn luyện mạng neural 33](#_Toc35716982)

[Hình 3.14. Kết quả RMSE sau khi chạy chương trình 33](#_Toc35716983)

[Hình 3.15. Đồ thị biểu diễn xu hướng dự đoán 34](#_Toc35716984)

CHƯƠNG 1: SƠ LƯỢC VỀ DỊCH MÁY NEURAL

Dịch máy là công cụ giúp chúng ta thực hiện việc dịch một cách tự động từ một ngữ này sang một ngôn ngữ khác.

Dịch máy dựa trên mạng neural (dịch máy neural) là một mô hình dịch máy mà trong đó có sử dụng các hệ thống mạng neural. Các mô hình dịch máy neural có cấu trúc gồm 2 phần chính: phần thứ nhất là bộ mã hóa (encoder), phần thứ hai là bộ giải mã (decoder). Kiến trúc này ta gọi là encoder-decoder hay seq2seq.

Bộ mã hóa (encoder) là một mô hình mạng neural mà trong đó các neural của nó được tính toán hay thực hiện dựa trên một trong các giải thuật deep learning (RNN, LSTM,…) dùng để biểu diễn tập dữ liệu đầu vào (tập dữ liệu ngôn ngữ gốc) thành 1 tập dữ liệu số đặc biệt. Trong tài liệu này các neural của encoder sẽ thực hiện trên Recurrent Neural Network (RNN).

Bộ giải mã (decoder) tương tự cũng là một một mô hình mạng neural và chức năng của nó từ chuyển hóa từ tập dữ liệu số đặc biệt của encoder thành tập dữ liệu đầu ra mà ta mong muốn (tập dữ liệu ngôn ngữ mong muốn).

Ví dụ khi ta muốn dịch từ tiếng Anh sang tiếng Việt, encoder sẽ chuyển tập dữ liệu tiếng Anh thành một dạng tập dữ liệu số đặc biệt và decoder sẽ có nhiệm vụ chuyển tập dữ liệu số đặc biệt này thành tập dữ liệu tiếng Việt tương ứng ở đầu ra.

Tài liệu này sẽ tập trung vào việc phân tích mô hình dịch máy neural bằng cách sử dụng hai phương pháp. Phương pháp thứ nhất là RNN sẽ được sử dụng cho encoder và decoder. Và phương pháp thứ hai là Convolutional Neural Network (CNN) sẽ được dụng cho encoder-decoder.

* 1. Giới thiệu về dịch máy neural

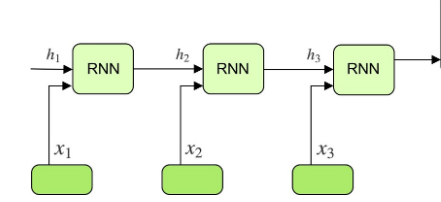
Ý tưởng dịch máy dựa trên hệ thống mạng neural được xuất phát từ ý tưởng ban đầu của Kalchbrenner và Blunsom vào năm 2013. Được lấy cảm hứng bằng việc áp dụng các kĩ thuật deep learning vào trong dịch máy. Tất cả các giải thuật đều được giới thiệu trong các tài liệu của Kalchbrenner, Blunsom vào năm 2013, của Sutskever vào năm 2014,… Trong tất cả các tài liệu này đều biểu diễn mô hình dịch thành 2 phần cơ bản là encoder và decoder.

Trong đó, encoder có chức năng chuyển hóa các tập dữ liệu ngôn ngữ gốc riêng biệt thành 1 tập dữ liệu số chung đặc biệt. Từ đó decoder có nhiệm vụ giải mã tập dữ liệu số đặc biệt này thành các tập dữ liệu ngôn ngữ muốn dịch tương ứng.

Cụ thể tài liệu này sẽ trình bày cách thực hiện làm sao ta có thể dịch từ tiếng anh sang tiếng việt.

* 1. Recurrent Neural Network trong mô hình

Ta có tập dữ liệu đầu vào:



1. Sơ lược cấu tạo hoạt động của RNN [1]

Tại mỗi bước (lớp) t, neural sẽ tính toán ra được một lượng giá trị gọi là ht. Giá trị ht này sẽ được đem đi tính chung với dữ liệu xt+1 tại bước t + 1. Giá trị ht tại mỗi bước t được tính như sau:

Trong đó:

* là kết quả của neural tại bước (lớp) t.
* là hàm activation.
* dữ liệu từ tập dữ liệu gốc

RNN là một kĩ thuật huấn luyện hiệu quả để tìm phân phối xác suất cho toàn tập bằng cách tính phân phối xác suất cho lớp tiếp theo . Trong trường hợp tập dữ liệu đầu vào là một chuỗi các vector từ 1 đến K thì phân phối xác suất của toàn tập ở đầu ra có thể được tính như sau:

Trong đó:

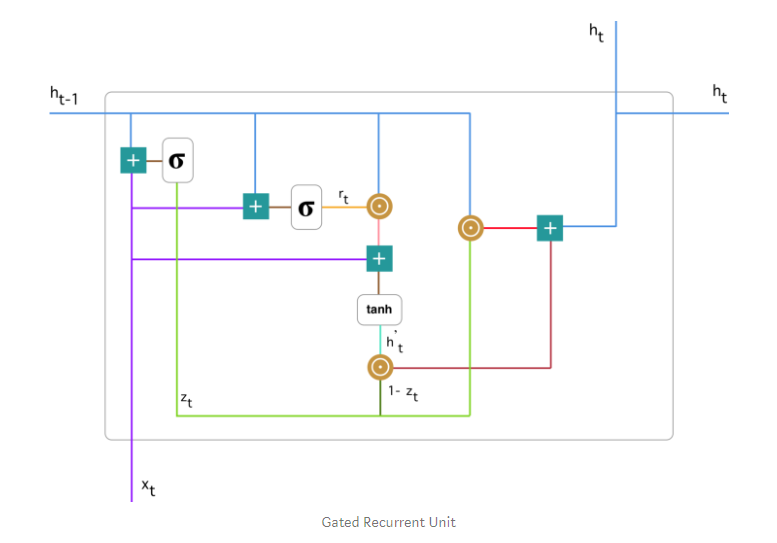
* là vector dữ liệu của neural thứ j ở bước thứ t.
* là trọng số tương ứng của neural thứ j.

Từ đó ta tính được phân bố xác suất cho toàn tập:

Trên hình 1.1 ta có thể thấy ở mỗi neural đều có 2 nguồn dữ liệu đầu vào (input). Input thứ nhất đó là dữ liệu gốc của tập dữ liệu và input thứ 2 là kết quả h tính được ở lớp phía trước (t – 1).

Dựa trên ý tưởng của LSTM, có một phương pháp tính mới cho mỗi neural được gọi là Gated Recurrent Unit (GRU). Tài liệu này sẽ trình bày dựa trên phương pháp tính toán này.

Trong GRU, ngoài 2 input đầu vào, ta sẽ có thêm 2 thành phần mới đó là cổng update z (update gate z) và cổng reset (reset gate r).



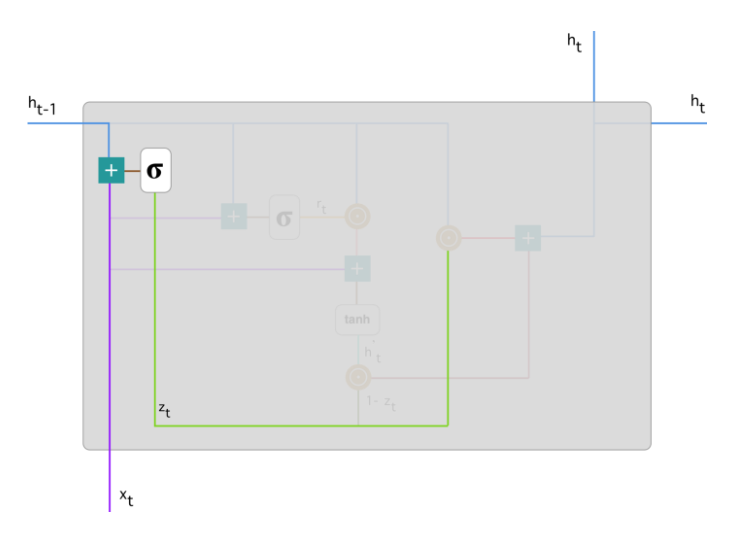
1. Cấu trúc neural theo GRU [2]
   * 1. Cổng update

Khi mới vào neural sẽ thực hiện việc tính toán các cổng trước.

Đầu tiên ta sẽ đi tìm hiểu về cổng update z. Cổng z được tính như sau:

Trong đó:

* trọng số ứng với
* trọng số ứng với



1. Mô tả cách tính của cổng update [2]

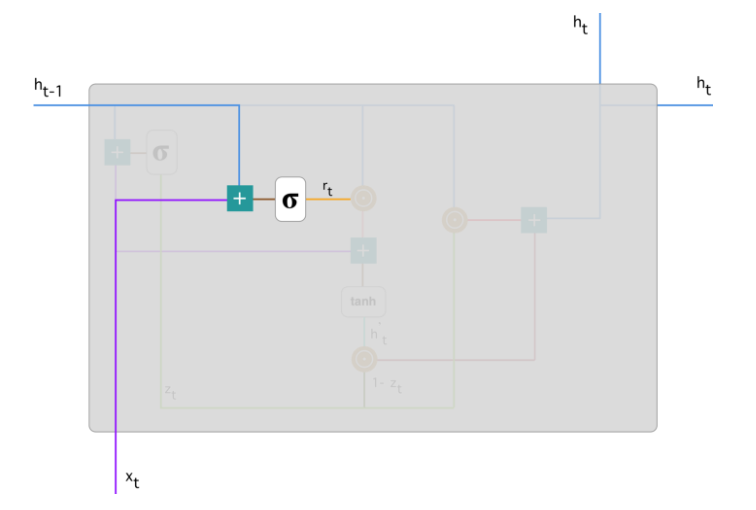
Cổng update tượng trưng cho việc tại mỗi neural, neural sẽ quyết định nên giữ lại bao nhiêu phần từ input để truyền lại cho neural ở các lớp tiếp theo. Điều này giúp neural chỉ giữ lại những phần cần thiết nhằm tránh gây ra hiện tượng vanishing gradient về sau.

* + 1. Cổng reset

Sau khi tính cổng update, ta sẽ đi thực hiện tính cổng reset r.

Trong đó:

* trọng số ứng với
* trọng số ứng với

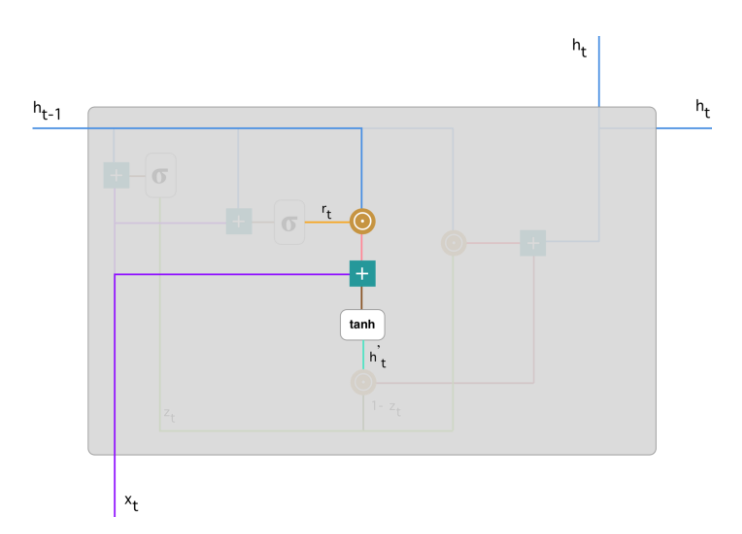


1. Mô tả cách tính cổng reset [2]

Cổng reset giúp neural quyết định xem nên bỏ đi bao nhiêu phần của input.

* + 1. Xuất kết quả

Đầu tiên ta sẽ tính giá trị :



1. Mô tả cách tính [2]

Cuối cùng ta sẽ tính giá trị đầu ra của neural:

Ta sẽ áp dụng mạng neural RNN với mỗi neural là GRU cho cả 2 bộ là encoder và decoder.

Ở encoder đầu vào của mỗi neural sẽ là tập dữ liệu văn bản của ngôn ngữ gốc kèm với kết quả đầu ra của neural phía trước. Đầu ra của mỗi neural là kết quả tính được của mỗi neural.

Ở decoder đầu vào của mỗi neural gồm kết quả đầu ra của neural phía trước và kết quả ta dịch được tại bước phía trước (t – 1). Còn đầu ra sẽ gồm kết quả ta dịch được kèm với kết quả ta tính được ở neural hiện tại.

* 1. Convolutional Neural Network trong mô hình

Ngoài việc sử mô hình mạng neural RNN cho quá trình dịch máy, ta còn có thể sử dụng mô hình mạng neural Convolutional Neural Network (CNN) vào quá trình dịch.

Ta có tập dữ liệu đầu vào:

Ở bước 0, hidden unit được tính như sau:

Trong đó U là trọng số tại bước 0.

Mạng CNN của ta sẽ có 4 trọng số chính là: , , và . Tại mỗi bước , đầu vào của neural j được tính dựa trên 3 giá trị đầu vào cơ bản: giá trị đầu vào bên trái (left - l), bên phải (right - r) và ở giữa (center - c).

Giá trị ở giữa được tính như sau:

Giá trị của neural hiện tại được tính như sau:

Các giá trị , và được tính như sau:

Trong đó và Z được tính như sau:

Theo kích hoạt này, người ta có thể nghĩ đến việc kích hoạt một nút duy nhất ở cấp độ đệ quy t là sự lựa chọn giữa một kích hoạt mới được tính toán từ cả trẻ trái và phải, kích hoạt từ trẻ trái hoặc kích hoạt từ trẻ phải. Lựa chọn này cho phép cấu trúc tổng thể của tích chập đệ quy thay đổi thích ứng đối với mẫu đầu vào. Xem hình 2 (b) cho một minh họa.

Về mặt này, chúng tôi thậm chí có thể coi grConv được đề xuất là thực hiện một loại phân tích cú pháp không giám sát. Nếu chúng tôi xem xét trường hợp đơn vị gating đưa ra quyết định khó khăn, tức là ,! tuân theo mã hóa 1-K, dễ dàng thấy rằng mạng thích nghi với đầu vào và tạo thành một cấu trúc giống như cây (Xem hình 2 (cọ d)). Tuy nhiên, chúng tôi để lại nghiên cứu sâu hơn về cấu trúc mà mô hình này học được cho nghiên cứu trong tương lai.

* 1. Tóm tắt chương 1

TÀI LIỆU THAM KHẢO

**Tiếng Anh:**

1. <https://towardsdatascience.com/understanding-encoder-decoder-sequence-to-sequence-model-679e04af4346>
2. <https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be>
3. Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.
4. <https://www.simplilearn.com/what-is-perceptron-tutorial>
5. <http://neuralnetworksanddeeplearning.com/chap1.html>
6. <http://neuralnetworksanddeeplearning.com/chap2>