TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO HỌC KỲ I NĂM HỌC 2022 – 2023**

**MÔN: XỬ LÝ NGÔN NGỮ TỰ NHIÊN**

**PROJECT 1**

*Người hướng dẫn*: **THẦY LÊ ANH CƯỜNG**

*Người thực hiện*: **TRẦN TRUNG DŨNG -**

**LÊ HUỲNH THỊ NGỌC HÂN – 51900330**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO HỌC KỲ I NĂM HỌC 2022 – 2023**

**MÔN: HỌC MÁY**

**PROJECT 1**

*Người hướng dẫn*: **THẦY LÊ ANH CƯỜNG**

*Người thực hiện*: **TRẦN TRUNG DŨNG -**

**LÊ HUỲNH THỊ NGỌC HÂN – 51900330**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

LỜI CẢM ƠN

Đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến Thầy Lê Anh Cường. Thầy là người hỗ trợ và hướng dẫn tận tình cho chúng em trong quá trình nghiên cứu, tìm hiểu để hoàn thành bài báo cáo môn Xử lý ngôn ngữ tự nhiên.

Tiếp theo, chúng em xin gửi lời cảm ơn đến khoa Công Nghệ Thông Tin trường Đại học Tôn Đức Thắng vì đã tạo điều kiện cho chúng em được học tập và nghiên cứu môn học này. Khoa đã luôn sẵn sàng chia sẻ các kiến thức bổ ích cũng như kinh nghiệm tham khảo tài liệu, giúp ích không chỉ cho việc thực hiện và hoàn thành đề tài nghiên cứu mà còn giúp ích cho việc học tập và rèn luyện trong quá trình thực hành tại trường Đại học Tôn Đức Thắng nói chung.

Cuối cùng, trong quá tình hoàn tất đề tài của mình nhờ vào sự hướng dẫn, giúp đỡ và những kiến thức học hỏi được từ quý thầy cô giảng viên. Do giới hạn về mặt kiến thức và khả năng lý luận nên chúng em còn nhiều thiếu sót, kính mong sự chỉ dẫn và đóng góp của quý thầy cô giảng viên để bài nghiên cứu của chúng em được hoàn thiện hơn. Hơn nữa, nhờ những góp ý từ thầy cô và bạn bè, chúng em sẽ hoàn thành tốt hơn ở các bài nghiên cứu khác sau này. Chúng em mong quý thầy cô giảng viên và bạn bè sẽ luôn quan tâm và hỗ trợ chúng em và chúc quý thầy cô giảng viên và các bạn luôn tràn đầy sức khỏe.

CHÚNG EM XIN CHÂN THÀNH CẢM ƠN!

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của chúng tôi và được sự hướng dẫn của thầy Lê Anh Cường. 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 tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Trần Trung Dũng*

*Lê Huỳnh Thị Ngọc Hân*

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

[TRÌNH BÀY LÝ THUYẾT 2](#_Toc127543791)

[1. Tổng quan phương pháp học biểu diễn từ 2](#_Toc127543792)

[2. Ưu điểm của phương pháp Word2Vec 3](#_Toc127543793)

[3. Phương pháp Word2Vec 3](#_Toc127543794)

[BÀI 2 14](#_Toc127543795)

[2.1 Mô tả dữ liệu 14](#_Toc127543796)

[2.2 Mô tả thuật toán 17](#_Toc127543797)

[2.3 Thực nghiệm 25](#_Toc127543798)

**DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

**CÁC KÝ HIỆU**

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

**DANH MỤC BẢNG**

TRÌNH BÀY LÝ THUYẾT

1. Tổng quan phương pháp học biểu diễn từ

Có nhiều phương pháp học biểu diễn từ (word embedding) hiện tại được sử dụng trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), tuy nhiên các phương pháp phổ biến nhất bao gồm:

* Word2Vec: Đây là một phương pháp học biểu diễn từ phổ biến bằng cách sử dụng mô hình neural network đơn giản với hai kiểu đào tạo (training method) là Continuous Bag-of-Words (CBOW) và Skip-gram. Word2Vec thường tạo ra các biểu diễn từ có số chiều tương đối thấp, từ đó giúp tăng tốc độ xử lý cho các tác vụ NLP.
* GloVe: Phương pháp này là viết tắt của Global Vectors, và dựa trên việc tính toán ma trận liên kết thống kê giữa các từ dựa trên tần suất xuất hiện của chúng trong một văn bản lớn. Ma trận này sau đó được giải quyết để tìm ra biểu diễn từ véc tơ có số chiều tương đối cao hơn so với Word2Vec.
* FastText: Đây là một phương pháp khác sử dụng mạng neural để học biểu diễn từ, tuy nhiên FastText thêm một chiều vào việc đào tạo, tương ứng với việc học các phần nhỏ hơn của từ (subword). Điều này giúp FastText xử lý tốt các từ mới, không nằm trong tập từ điển đã biết trước đó.
* Transformer-based models: Đây là một loại mô hình mới nhất trong việc học biểu diễn từ, dựa trên kiến trúc Transformer và sử dụng phương pháp self-attention để tính toán trọng số cho các từ trong văn bản. Các mô hình Transformer-based như BERT, GPT và RoBERTa hiện nay đang được sử dụng rộng rãi trong các tác vụ NLP.

Các phương pháp trên là các phương pháp học biểu diễn từ phổ biến hiện nay, tùy thuộc vào nhiệm vụ cụ thể mà một phương pháp nào sẽ phù hợp hơn cho từng tác vụ.

Phương pháp Word2Vec là một phương pháp học biểu diễn từ (word embedding) phổ biến trong xử lý ngôn ngữ tự nhiên (NLP). Nó được sử dụng để chuyển đổi các từ thành các vector số học, sao cho các từ có ý nghĩa tương tự nhau trong ngữ cảnh sẽ có các biểu diễn vector gần nhau trong không gian vector.

1. Ưu điểm của phương pháp Word2Vec

Word2Vec sử dụng một mô hình neural network đơn giản với hai kiểu đào tạo (training method) là Continuous Bag-of-Words (CBOW) và Skip-gram. Với CBOW, mô hình cố gắng dự đoán từ đang xét dựa trên các từ lân cận, trong khi với Skip-gram, mô hình cố gắng dự đoán các từ lân cận dựa trên từ đang xét.

Phương pháp này mang lại nhiều lợi ích có thể kể đến bao gồm:

* Tốc độ xử lý nhanh: Word2Vec tạo ra các biểu diễn từ có số chiều tương đối thấp, từ đó giúp tăng tốc độ xử lý cho các tác vụ NLP.
* Hiệu quả và độ chính xác cao: Word2Vec được chứng minh là rất hiệu quả trong việc học biểu diễn từ với độ chính xác cao.
* Khả năng khai thác được ngữ cảnh: Word2Vec có khả năng khai thác được ngữ cảnh của các từ trong văn bản để xác định các từ có ý nghĩa tương đồng nhau.
* Dễ triển khai và sử dụng: Word2Vec đã được triển khai và được sử dụng rộng rãi trong cộng đồng NLP. Nhiều thư viện NLP phổ biến như Gensim, TensorFlow, PyTorch cung cấp các API để triển khai Word2Vec một cách dễ dàng.

Nhìn chung, Word2Vec là một phương pháp học biểu diễn từ hiệu quả, tốc độ xử lý nhanh và dễ triển khai, được sử dụng rộng rãi trong các ứng dụng NLP.

1. Phương pháp Word2Vec

Trước khi đi vào phương pháp biểu diễn, ta cần làm rõ một số khái niệm:

* *Documents (Văn bản):* Là tập hợp các câu trong cùng một đoạn văn có mối liên hệ với nhau. Văn bản có thể được coi như một bài báo, bài văn,….
* *Corpus (Bộ văn bản):* Là một tợp hợp gồm nhiều văn bản thuộc các đề tài khác nhau, tạo thành một nguồn tài nguyên dạng văn bản. Một văn bản cũng có thể được coi là corpus của các câu trong văn bản. Các bộ văn bản lớn thường có từ vài nghìn đến vài trăm nghìn văn bản trong nó. Một số bộ văn bản trong tiếng việt có thể được download từ nguồn Wikipedia, VNCoreNLP.
* *Character (kí tự):* Là tợp hợp gồm các chữ cái (nguyên âm và phụ âm) và dấu câu. Mỗi một ngôn ngữ sẽ có một bộ các kí tự khác nhau.
* *Word (từ vựng):* Là các kết hợp của các kí tự tạo thành những từ biểu thị một nội dung, định nghĩa xác định, chẳng hạn con người có thể coi là một từ vựng. Từ vựng có thể bao gồm từ đơn có 1 âm tiết và từ ghép nhiều hơn 1 âm tiết. Khác với tiếng anh khi các từ chủ yếu là đơn âm. Tiếng việt có rất nhiều những từ ghép 2, 3 âm tiết. Do đó chúng ta cần phải có từ điển để thực hiện tách từ (tokenize) trong câu. Một số package thông dụng trong Tiếng Việt có sẵn chức năng này được sử dụng phổ biến là *underthesea, pyvi, VNCoreNLP, RDRsegmenter, coccoc-tokenizer*. Kết quả tokenize có thể khác nhau tuỳ thuộc vào cách định nghĩa từ ghép ở mỗi package. Khi xử lý ngôn ngữ tự nhiên cho một số lĩnh vực đặc biệt cần phải có từ điển chuyên ngành, vì vậy cần phải customize riêng mà không nên sử dụng từ điển từ package.
* *Dictionary (từ điển):* Là tợp hợp các từ vựng xuất hiện trong văn bản.
* *Volcabulary (từ vựng):* Tợp hợp các từ được trích xuất trong văn bản. Tương tự như từ điển.

Trước khi biểu diễn từ, ta cần xác định từ điển của văn bản. Số lượng từ là hữu hạn và được lặp lại trong các câu. Do đó thông qua từ điển gồm tập hợp tất cả các từ có thể xuất hiện, ta có thể mã hoá được các câu dưới dạng ma trận mà mỗi dòng của nó là một véc tớ one-hot của từ.

1. ***Định nghĩa One-hot vectơ của từ:***

Giả sử ta có từ điển là tập hợp từ vựng . Khi đó, mỗi từ sẽ có một đại diện index riêng lần lượt 0, 1, 2, ... tương ứng với One-hot vectơ của từ vựng thứ(với sẽ là vecto sao cho các phần tử của vecto thỏa mãn:

1. ***Word Embedding***

Sau khi biểu diễn từ dưới dạng one-hot véc tơ, mô hình đã có thể huấn luyện được từ dữ liệu được mã hóa. Tuy nhiên dữ liệu này chỉ đáp ứng được khả năng huấn luyện mà chưa phản ảnh được mối liên hệ về mặt ngữ nghĩa của các từ và các hạn chế khác bao gồm:

* Mối quan hệ tương quan giữa các cặp từ bất kì luôn là không tương quan (tức bằng 0). Do đó không có tác dụng trong việc tìm mối liên hệ về nghĩa.
* Kích thước của véc tơ sẽ phụ thuộc vào số lượng từ vựng có trong bộ văn bản dẫn đến chi phí tính toán rất lớn khi tập dữ liệu lớn.
* Khi bổ sung thêm các từ vựng mới số chiều của véc tơ có thể thay đổi theo dẫn đến sự không ổn định trong shape.

Do đó các thuật toán nhúng từ được tạo ra nhằm mục đích tìm ra các véc tơ đại diện cho mỗi từ sao cho:

* Một từ được biểu diễn bởi một véc tơ có số chiều xác định trước.
* Các từ thuộc cùng 1 nhóm thì có khoảng cách gần nhau trong không gian.

Có nhiều phương pháp nhúng từ khác nhau, có thể kể đến 3 nhóm chính:

* Sử dụng thống kê tần xuất: *tfidf*
* Các thuật toán giảm chiều dữ liệu: *SVD, PCA, auto encoder, word2vec*
* Phương pháp sử dụng mạng nơ ron: *word2vec, ELMo, BERT.*

1. ***Phương pháp SVD***

SVD là phương pháp giảm chiều dữ liệu dựa trên một phép phân tích suy biến nhằm tìm ra một ma trận gần sát với ma trận ban đầu. Đối với word embedding theo SVD, ta sẽ áp dụng phân tích suy biến trên ma trận đồng xuất hiện của các cặp từ input và output. Trong đó input là từ hiện tại và output là các từ liền kề xung quanh nó. Chẳng hạn chúng ta có 2 câu văn như sau:

Tập từ điển sẽ bao gồm các từ:

Khi đó ta sẽ có các từ được biểu diễn trong ma trận đồng nhất sau:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **xử lý** | **ngôn ngữ  tự nhiên** | **là** | **một** | **môn học** | **thú vị** | **tôi** | **rất** | **yêu thích** |
| **xử lý** |  | 2 |  |  |  |  |  |  |  |
| **ngôn ngữ tự nhiên** |  |  | 1 |  |  |  |  |  |  |
| **là** |  |  |  | 1 |  |  |  |  |  |
| **một** |  |  |  |  | 1 |  |  |  |  |
| **môn học** |  |  |  |  |  | 1 |  |  |  |
| **thú vị** |  |  |  |  |  |  |  |  |  |
| **tôi** |  |  |  |  |  |  |  | 1 |  |
| **rất** |  |  |  |  |  |  |  |  | 1 |
| **yêu thích** |  |  |  |  |  |  |  |  |  |

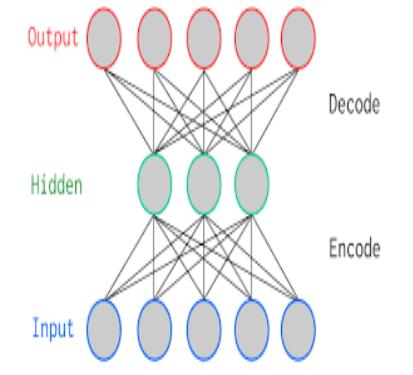
Chúng ta cũng có thể tìm ra biểu diễn của mỗi từ trong từ điển bằng một véc tơ các nhân tố ẩn dựa vào việc lựa chọn một số lượng các giá trị đặc trưng.

Phương pháp SVD (Singular Value Decomposition) là một kỹ thuật toán đại số tuyến tính phân tích một ma trận vuông hoặc chéo khác. SVD được sử dụng rộng rãi trong nhiều lĩnh vực, bao gồm khoa học dữ liệu, xử lý ảnh, xử lý tín hiệu, lý thuyết đồ thị, thống kê và nhiều lĩnh vực khác. Một số ứng dụng của phương pháp SVD bao gồm:

* Giảm chiều dữ liệu: SVD có thể được sử dụng để giảm chiều của các bộ dữ liệu lớn bằng cách chọn các thành phần chính của ma trận và loại bỏ các thành phần ít quan trọng.
* Nén hình ảnh: SVD có thể được sử dụng để nén hình ảnh bằng cách giảm kích thước của ma trận đại diện cho hình ảnh.
* Phân tích tín hiệu: SVD được sử dụng để phân tích các tín hiệu phức tạp, như âm thanh và video, thành các thành phần cơ bản và hiểu được cấu trúc của tín hiệu.
* Phân tích đồ thị: SVD có thể được sử dụng để phân tích cấu trúc của đồ thị và tìm ra các thành phần quan trọng của đồ thị.
* Giải bài toán tối ưu: SVD có thể được sử dụng để giải các bài toán tối ưu như PCA (Phân tích thành phần chính), LSA (Phân tích ngữ nghĩa latents), và Ridge Regression.

1. ***Phương pháp auto encoder***

Autoencoder là một mô hình mạng thần kinh sâu (Deep Neural Network) được sử dụng để giảm chiều dữ liệu hoặc rút trích đặc trưng. Auto encoder được xây dựng trên một mạng nơ ron có 3 layer: input, hidden layer và output. Trong đó số units ở input và output là bằng nhau. Số units ở hidden layer sẽ qui định số chiều của véc tơ biểu diễn từ, thông thường sẽ nhỏ hơn số units ở đầu vào.



*Phương pháp auto encoder với số units ở đầu vào bằng đầu ra.*

Phương pháp Auto Encoder bao gồm các bước thực hiện sau:

* *Xây dựng kiến trúc mạng:* Bước đầu tiên là xác định kiến trúc mạng cho Autoencoder. Mạng bao gồm ít nhất 2 lớp: lớp mã hóa (encoder) và lớp giải mã (decoder). Thông thường, lớp mã hóa có kích thước nhỏ hơn so với kích thước của dữ liệu gốc, trong khi lớp giải mã có kích thước bằng với kích thước của dữ liệu gốc.
* *Xác định hàm mất mát (loss function):* Hàm mất mát được sử dụng để đánh giá sự sai khác giữa đầu ra của mô hình và dữ liệu đầu vào. Trong Autoencoder, hàm mất mát thường là sai số bình phương (mean squared error) giữa đầu ra và đầu vào.
* *Điều chỉnh trọng số:* Mô hình sẽ được đưa qua quá trình huấn luyện để tối thiểu hóa hàm mất mát và điều chỉnh trọng số trong mạng để tạo ra đầu ra xấp xỉ gần với đầu vào. Phương pháp gradient descent thường được sử dụng để tối ưu hoá hàm mất mát và điều chỉnh trọng số.
* *Kiểm tra mô hình:* Sau khi huấn luyện xong, mô hình được kiểm tra trên tập dữ liệu kiểm tra để đánh giá hiệu suất của mô hình. Nếu mô hình không đáp ứng được yêu cầu, ta có thể tiếp tục điều chỉnh kiến trúc mạng hoặc thay đổi tham số huấn luyện.
* *Sử dụng mô hình:* Sau khi huấn luyện mô hình, chúng ta có thể sử dụng mô hình để giảm chiều dữ liệu hoặc rút trích đặc trưng (feature extraction) từ các dữ liệu mới.

1. ***Các phương pháp trong mô hình word2vec***

Mô hình Word2Vec là một phương pháp học không giám sát để chuyển đổi các từ thành các vectơ số thực trong không gian vector có số chiều thấp. Mô hình Word2Vec được xây dựng trên cơ sở giả định rằng các từ xuất hiện gần nhau trong văn bản có nghĩa tương đồng và có thể được sử dụng để dự đoán các từ khác trong văn bản.

Có hai kiểu kiến trúc mô hình Word2Vec bao gồm Skip-Gram và Continuous Bag of Words (CBOW).

* *Continuous Bag of Words (CBOW):*
  + Tạo một tập hợp các từ (word) trong văn bản (corpus).
  + Chuyển đổi các từ thành các one-hot vectors, trong đó mỗi vector có số chiều bằng số lượng từ trong tập hợp.
  + Xây dựng một mô hình neural network đơn giản với một lớp ẩn.
  + Sử dụng các one-hot vectors của các từ xung quanh để dự đoán từ tại một vị trí cụ thể.
  + Huấn luyện mô hình bằng cách tối ưu hóa hàm mất mát (loss function), ví dụ như hàm cross-entropy loss, để đưa ra các vectơ nhúng (embedding vectors) cho mỗi từ trong không gian vector.
* *Skip-Gram:*
  + Tạo một tập hợp các từ trong văn bản.
  + Chuyển đổi các từ thành các one-hot vectors.
  + Xây dựng một mô hình neural network với một lớp ẩn.
  + Sử dụng từ ở vị trí tại một từ cụ thể để dự đoán các từ xung quanh.
  + Huấn luyện mô hình bằng cách tối ưu hóa hàm mất mát, ví dụ như negative sampling, để đưa ra các vectơ nhúng cho mỗi từ.

Sau khi hoàn tất quá trình huấn luyện, các vectơ nhúng cho mỗi từ trong không gian vector có số chiều thấp có thể được sử dụng để đo lường sự tương đồng giữa các từ và thực hiện các tác vụ như phân loại, tìm kiếm và sinh văn bản.

1. ***Phương pháp Skip grams***

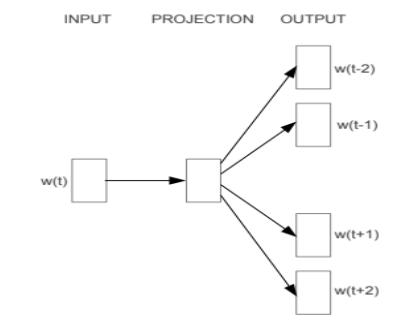
Giả sử chúng ta có một câu văn như sau:

.

Để thu được một phép nhúng từ tốt hơn chúng ta sẽ lựa chọn ra ngẫu nhiên các từ làm bối cảnh (*context*). Dựa trên từ bối cảnh, các từ mục tiêu (*target*) sẽ được xác định nằm trong phạm vi xung quanh từ bối cảnh. Chẳng hạn ta với việc lựa chọn từ làm bối cảnh nếu lấy từ tiếp theo, từ liền trước, từ cách đó liền trước 2, 3 từ ta sẽ lần lượt thu được các từ mục tiêu như sau:

| **Bối cảnh (context)** | **Mục tiêu (target)** |
| --- | --- |
| cốc | màu\_xanh |
| cốc | chiếc |
| cốc | một |
| cốc | muốn |

Các nghiên cứu cho thấy từ mục tiêu sẽ được giải thích tốt hơn nếu được học theo các từ bối cảnh. Do đó mô hình skip-grams tìm cách xây dựng một thuật toán học có giám sát có đầu vào là các từ bối cảnh đến đầu ra là từ mục tiêu:



*Kiến trúc mô hình Skip grams*

*Với w\_t là từ bối cảnh, w\_(t-1),w\_(t-2),w\_(t+1),w\_(t+2) là các từ mục tiêu*

Cũng giống như các cách tiếp cận thông thường khác, mô hình sẽ biểu diễn một từ bối cảnh dưới dạng one-hot vectơ . Vectơ này sẽ trở thành đầu vào cho một mạng nơ ron có tầng ẩn gồm 300 units. Kết quả ở output layer là một hàm softmax tính xác xuất để các từ mục tiêu phân bố vào những từ trong vocabulary (10000 từ). Dựa trên quá trình feed forward và back propagation mô hình sẽ tìm ra tham số tối ưu để kết quả dự báo từ mục tiêu là chuẩn xác nhất. Khi đó quay trở lại tầng hidden layer ta sẽ thu được đầu ra tại tầng này là ma trận nhúng :

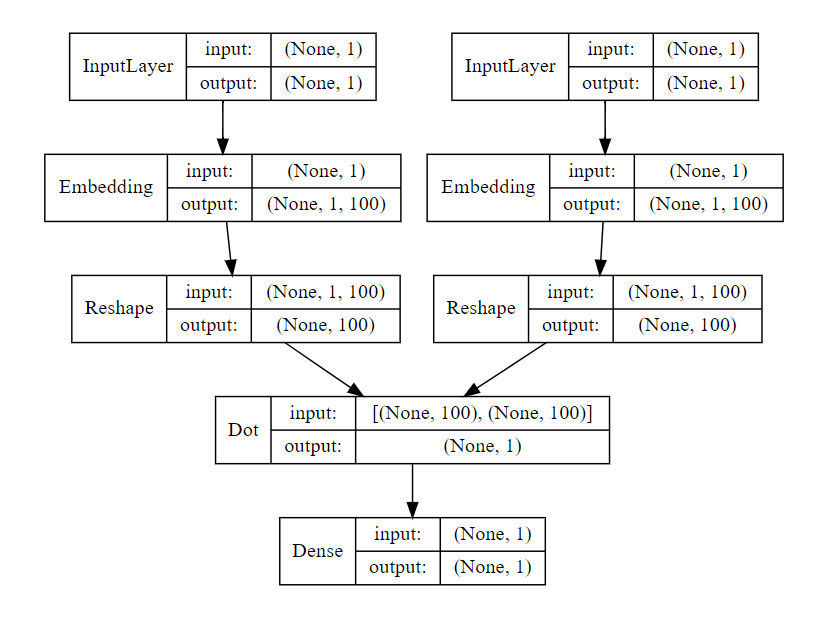
Trong đó, là vectơ nhúng trích xuất từ ma trận tương ứng với bối cảnh . là xác xuất được dự báo của từ mục tiêu.

Khi áp dụng hàm softmax, xác xuất ở đầu ra có dạng: . Trong đó, là các véc tơ tham số thể hiện sự liên kết giữa các units ở hidden layer với output layer.

Kết quả dự báo mô hình mạng nơ ron càng chuẩn xác thì véc tơ nhúng sẽ càng thể hiện được mối liên hệ trên thực tế giữa từ bối cảnh và mục tiêu chuẩn xác. Kết quả cuối cùng ta quan tâm chính là các dòng của ma trận . Chúng là các véc tơ nhúng đại diện cho một từ bối cảnh .

Các bước chính để xây dựng mô hình skipgram bao gồm:

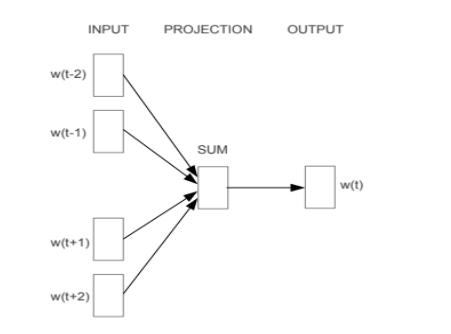
* Bước 1: Chuẩn bị dữ liệu là các cặp *[context, target].*
* Bước 2: Xây dựng mạng nơ ron:



* Bước 3: Huấn luyện mô hình.
* Bước 4: Trích xuất ra véc tơ nhúng ở layer đầu tiên: Tìm các từ gần nghĩa nhất với từ cần tính dựa trên khoảng cách euclidean.

1. ***Phương pháp CBOW***

Cả CBOW và Skip-gram là hai mô hình phổ biến trong bài toán Word2Vec của Machine Learning để học nhúng từ (word embedding) từ dữ liệu văn bản. Tuy nhiên, chúng có một số điểm khác nhau về cách thức hoạt động và ứng dụng. Mô hình skip-grams sẽ rất tốn chi phí để tính toán vì mẫu số xác xuất là tổng của rất nhiều số mũ cơ số tự nhiên. Để hạn chế chi phí tính toán mô hình CBOW (continueos backward model) được áp dụng. Về cơ bản thì CBOW là một quá trình ngược lại của skip-grams. Khi đó input của skip-grams sẽ được sử dụng làm output trong CBOW và ngược lại.



*Kiến trúc CBOW*

Kiến trúc mạng nơ ron của CBOW sẽ gồm 3 layers:

* *Input layers:* Là các từ bối cảnh xung quanh từ mục tiêu.
* *Projection layer:* Lấy trung bình véc tơ biểu diễn của toàn bộ các từ input để tạo ra một véc tơ đặc trưng.
* *Output layer:* Là một dense layers áp dụng hàm softmax để dự báo xác xuất của từ mục tiêu.

Dựa trên phương pháp CBOW, ta xây dựng mô hình nhúng các từ bối cảnh thành những véc tơ có 300 chiều bằng *keras*. Để xây dựng mô hình sẽ đi qua các bước sau đây:

* Bước 1: Tạo bộ từ điển cho toàn bộ các câu trong kinh thánh sao cho mỗi từ được gán giá trị bởi 1 số *index*.
* Bước 2: Mã hoá toàn bộ các câu văn bằng index.
* Bước 3: Xác định các cặp *Context và Target* tương ứng với *input* và output của mô hình. Trong đó từ *Target* là từ hiện tại ở vị trí *index*, các từ *Context* nằm ở khoảng *[index - window\_size, index + window\_size].* Padding giá trị 0 tại những *context* không đủ độ dài là *2\*window\_size.*
* Bước 4: Xây dựng mạng nơ ron gồm 3 layers chính:
  + *Embedding layer:* dùng để mã hoá đầu vào thành các one-hot vectơ. Số lượng từ ở đầu vào chính là *2\*window\_size*. Sau khi mã hoá, qua quá trình training mỗi một từ vựng sẽ được biểu diễn bởi một véc tơ nhúng 100 chiều tương ứng với *embed\_size.*
  + *Mean layer*: Tính véc tơ trung bình của các véc tơ đầu ra ở *Embedding* *layer*. Số lượng véc tơ là *2\*window\_size.*
  + *Dense layer:* Tính phân phối xác xuất của từ *Target* dựa vào hàm *softmax*.
* Bước 5: Huấn luyện mô hình.
* Bước 6: Trích xuất ma trận nhúng tại đầu ra của *hidden* *layer*.

1. Thực nghiệm phương pháp học biểu diễn từ

<https://colab.research.google.com/drive/1KsqUI4EAzk2WhZG9XIC82ZLljQvGq3Xe?usp=sharing>

ỨNG DỤNG HỌC BIỂU DIỄN TỪ

TRONG LANGUAGE MODEL: CHO WORD PREDICTION

<https://colab.research.google.com/drive/1OfyzGjPUsQ2H4MbCvA-ZzeHtHFM-SRNp?usp=sharing>

TÀI LIỆU THAM KHẢO

1. **Giáo trình Nhập môn học máy,** khoa Công nghệ thông tin, trường Đại học Tôn Đức Thắng.
2. **Machine Learning cơ bản** (2017, April 9) Retrieved January 18, 2022.

https://machinelearningcoban.com/2017/04/09/smv/

1. **Machine Learning cơ bản** (2017, January 27) Retrieved January 18, 2022. https://machinelearningcoban.com/2017/01/27/logisticregression/
2. **Phân lớp bằng Random Forests trong Python** (Viblo), Retrieved January 18, 2022. <https://viblo.asia/p/phan-lop-bang-random-forests-trong-python-djeZ1D2QKWz>
3. **Giải thích chi tiết về mạng LSTM** (2018, October 18), Nguyễn Trường Long https://nguyentruonglong.net/giai-thich-chi-tiet-ve-mang-long-short-term-memory-lstm.html