Báo cáo Nhập môn trí tuệ nhân tạo - IT3160

Nhóm 9

Tháng 6 2024

Mục lục

1 Giới thiệu thành viên

Họ và tên	MSSV	Phân chia công việc	Phần trăm công việc
Nguyễn Đình Hiếu	20215049	Tìm hiểu thuật toán SGD, so sánh với thuật toán GD truyền thống Code phần training bằng SGD để tối ưu 2 hàm loss của Word2VEc Làm báo cáo	33.33%
Hoàng Đức Gia Hưng	20215062	Tìm hiểu hàm loss phần naive softmax và ưu nhược điểm Code phần naive softmax của word2vec Làm báo cáo	33.33%
Mai Minh Khôi	20210492	Tìm hiểu hàm loss phần negative sampling và đánh giá word2vec Tìm dữ liệu và code phần negative sampling Làm slide	33.33%

2 Mô tả bài toán

Word2vec là một mô hình đơn giản và nổi tiếng giúp tạo ra các biểu diễn embedding của từ trong một không gian có số chiều thấp hơn nhiều lần so với số từ trong từ điển. Ý tưởng cơ bản của word2vec có thể được gói gọn trong các ý sau:

- Hai từ xuất hiện trong những văn cảnh giống nhau thường có ý nghĩa gần với nhau.
- Ta có thể đoán được một từ nếu biết các từ xung quanh nó trong câu. Ví dụ, với câu "Hà Nội là ... của Việt Nam" thì từ trong dấu ba chấm khả năng cao là "thủ đô". Với câu hoàn chỉnh "Hà Nội là thủ đô của Việt Nam", mô hình word2vec sẽ xây dựng ra embeding của các từ sao cho xác suất để từ trong dấu ba chấm là "thủ đô" là cao nhất.

Với ý tưởng trên, bọn em đề xuất một bài toán như sau

- Input: Dữ liệu dạng văn bản lấy từ Wikipedia
- Output: Một biểu đồ biểu diễn từ trong đó những từ có ngữ nghĩa gần giống nhau thì gần nhau

3 Phương pháp

3.1 Mô hình Skip-gram

Skip-gram là một mô hình dựa trên kiến trúc neural network, được sử dụng để tạo ra các biểu diễn vector cho từ vựng. Nó cố gắng dự đoán từ ngữ cảnh (context words) dựa trên từ trung tâm (center word) trong một văn bản. Mô hình này được đề xuất bởi Tomas Mikolov và các đồng nghiệp tại Google.

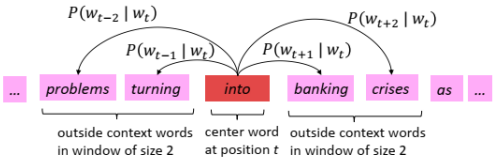
3.2 Hàm mục tiêu

Đối với mỗi vị trí t = 1,..., T, ta dự đoán các từ ngữ cảnh ở trong một cửa sổ với kích thước cố định m, khi đã cho trước từ trung tâm w_j. Xác suất dữ liệu:

$$\text{Likelihood} = L(\theta) = \prod_{t=1}^{T} \prod_{\substack{-m \leq j \leq m \\ j \neq 0}} P(w_{t+j} \mid w_t; \theta) \tag{1}$$

3.3 Naïve softmax

Giả sử ta có *context window* như sau: Để biểu diễn sự tương đồng giữa center



Hình 1: Context window

word w_t với outside context words w_{t+j} ta cần tối đa xác xuất $P(w_{t+j}|w_t)$ và cần giảm xác xuất các từ ngoài *context window*. Với mỗi từ w trong từ điển $\mathcal{V}$ ta sử dụng 2 embedding vector: v_w khi w là center word và u_w khi w là context word. Tương ứng với đó, ta có hai ma trận embedding U và V cho các context word và các center word. Mô hình Skipgram word2vec sẽ tối đa hoá xác xuất của context word o trong *context windows* với center word c cho trước.

$$\prod_{o \in \mathcal{C}_t} P(o \mid c) \tag{2}$$

Để tránh các sai số tính toán khi nhân các số nhỏ hơn 1 với nhau, bài toán tối ưu này thường được đưa về bài toán tối thiểu đối số của log (thường được gọi là *negative log loss*):

$$-\sum_{o\in\mathcal{C}_t}\log P\left(o\mid c\right) \tag{3}$$

Xác suất $P(o \mid c)$ được định nghĩa bởi:

$$P(o\mid c)=\frac{\exp\left(u_o^T v_c\right)}{\sum_{w\in\mathcal{V}}\exp\left(u_w^T v_c\right)} \tag{4}$$

Như vậy hàm mất mát cần tối ưu là:

$$\mathcal{L}\left(\mathbf{U},\mathbf{V};c\right)=-\sum_{o\in\mathcal{C}_t}\log\frac{\exp\left(u_o^T v_c\right)}{\sum_{w\in\mathcal{V}}\exp\left(u_w^T v_c\right)} \tag{5}$$

Tiến hành tối ưu hàm mất mát theo Gradient Descent. Xét riêng số hạng:

$$\log\frac{\exp\left(u_o^T v_c\right)}{\sum_{w\in\mathcal{V}}\exp\left(u_w^T v_c\right)}=u_o^T v_c-\log\sum_{w\in\mathcal{V}}\exp\left(u_w^T v_c\right)$$

Đạo hàm theo v_c:

$$\frac{\partial\log P\left(o\mid c\right)}{\partial v_c}=u_o-\sum_{x\in\mathcal{V}}\left(\frac{\exp\left(u_x^T v_c\right)u_x}{\sum_{w\in\mathcal{V}}\exp\left(u_w^T v_c\right)}\right)=u_o-\sum_{x\in\mathcal{V}}P\left(x\mid c\right)u_x \tag{6}$$

Ở mỗi bước cập nhật 2 ma trận U và V như sau:

$$\mathbf{U}^{(i+1)}=\mathbf{U}^{(i)}-\alpha\nabla_{\mathbf{U}}\mathbf{L}(\mathbf{U}^{(i)},\mathbf{V}^{(i)}) \tag{7}$$

$$\mathbf{V}^{(i+1)}=\mathbf{V}^{(i)}-\alpha\nabla_{\mathbf{V}}\mathbf{L}(\mathbf{U}^{(i)},\mathbf{V}^{(i)}) \tag{8}$$

3.3.1 Gradient Descent

Gradient của một hàm số là vectơ của các đạo hàm riêng của hàm số đó theo từng biến số. Nó cho biết hướng và mức độ tăng/giảm nhanh nhất của hàm số tại một điểm cụ thể.

Gradient Descent là một thuật toán tối ưu hóa dựa trên việc điều chỉnh các tham số của một mô hình theo hướng và khoảng cách được xác định bởi gradient của hàm mục tiêu. Thuật toán hoạt động bằng cách cập nhật các tham số của mô hình theo hướng ngược với gradient, nhằm giảm thiểu giá trị của hàm mục tiêu.

$$\theta = \theta - \eta \nabla J(\theta) \tag{9}$$

Với:

- θ là vectơ các tham số của mô hình cần cập nhật.
- η là tỷ lệ học (learning rate), quyết định tốc độ học của thuật toán.
- $\nabla J(\theta)$ là gradient của hàm mục tiêu J đối với các tham số θ.

3.3.2 Stochastic Gradient Descent (SGD)

Định nghĩa: Stochastic Gradient Descent (SGD) là một thuật toán tối ưu hóa được sử dụng rộng rãi trong máy học để tối ưu hóa các hàm mục tiêu bằng cách cập nhật các tham số của mô hình dựa trên gradient của hàm mục tiêu được ước lượng từ một mẫu dữ liệu ngẫu nhiên.

SGD thường được sử dụng trong các bài toán lớn hoặc dữ liệu lớn, nơi việc tính toán gradient trên toàn bộ dữ liệu huấn luyện là không khả thi. Thay vào đó, SGD chỉ cần một mẫu dữ liệu ngẫu nhiên tại mỗi bước cập nhật để ước lượng gradient, giúp giảm thời gian huấn luyện.

$\nabla J(\theta) \in \mathbb{R}^n$ có thể có dạng:

$$\begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & \theta_{22} & 0 & \theta_{24} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \theta_{m2} & 0 & \theta_{mn} \end{bmatrix}$$

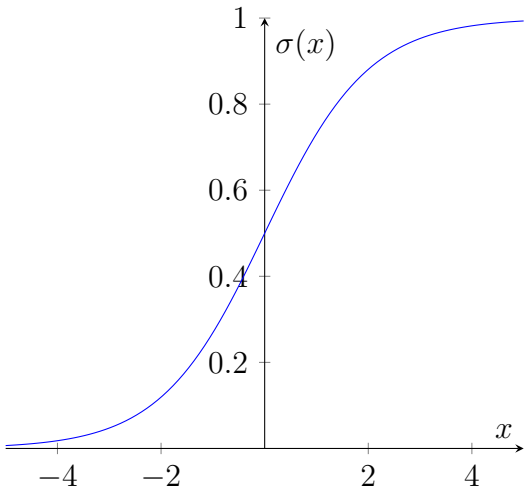
Trong đó: - Các phần tử không phải 0 là những phần tử tương ứng với từ ngữ cảnh trong cửa sổ. - Các phần tử còn lại là 0 do các từ không nằm trong cửa sổ ngữ cảnh không ảnh hưởng đến dự đoán của từ trung tâm.

Ma trận này biểu diễn các tham số của mô hình cần cập nhật bằng SGD. Mỗi phần tử trong ma trận tương ứng với một tham số trong mô hình.

3.4 Negative sampling

Ta có thể thấy trong gradient của biểu thức 6 chúng ta vẫn phải tính xác xuất $P(x \mid c)$, xác xuất này có phần mẫu phải tính toán trên toàn bộ từ điển $\mathcal{V}$ sẽ rất tốn kém. Vì vậy, để tránh việc phải tính toán nhiều, chúng ta có thể mô hình mỗi xác xuất $P(x \mid c)$ là một hàm sigmoid.

$$P(o \mid c) = \sigma(x) = \frac{1}{1 + \exp(-u_o^T v_c)}$$



Bản chất của bài toán tối ưu ban đầu là xây dựng mô hình sao cho với mỗi center word, xác suất của context word (từ ngữ cảnh) xảy ra là cao trong khi xác suất của toàn bộ các từ ngoài context window đó là thấp – việc này được thể hiện trong hàm softmax. Để hạn chế tính toán, trong phương pháp này ta chỉ lấy mẫu ngẫu nhiên một vài từ ngoài context window đó để tối ưu. Các từ trong context window được gọi là "từ dương", các từ ngoài ngữ cảnh được gọi là "từ âm"; vì vậy phương pháp này còn có tên gọi khác là "lấy mẫu âm" (negative sampling).

Từ ý tưởng đó ta xây dựng hàm loss như sau:

$$\mathcal{L}\left(\mathbf{U}, \mathbf{V}; c\right) = -\log \sigma(u_o^T v_c) - \sum_{k \notin \text{context window}} \log \sigma(-u_k^T v_c) \tag{10}$$

3.5 Mô hình Skip-gram với Negative Sampling:

Định nghĩa: Mô hình Skip-gram với Negative Sampling là một biến thể của mô hình Skip-gram trong xử lý ngôn ngữ tự nhiên, sử dụng phương pháp huấn luyện Negative Sampling để cải thiện hiệu suất tính toán.

Công thức: Công thức của mô hình Skip-gram với Negative Sampling thường được biểu diễn bằng hàm mục tiêu như sau:

$$J_{NEG}(u_o, v_c, U) = -\log \sigma(u_o^T v_c) - \sum_{k=1}^{K} \log \sigma(-u_k^T v_c)$$

Trong đó:

- $J_{NEG}(\theta)$ là hàm mục tiêu với Negative Sampling.
- $\sigma(x)$ là hàm sigmoid: $\sigma(x) = \frac{1}{1+e^{-x}}$.
- u_o là vector biểu diễn của từ mục tiêu.
- v_c là vector biểu diễn của từ ngữ cảnh.
- u_k là vector biểu diễn của một từ mẫu âm (negative sample).
- K là số lượng các từ âm mẫu được lấy mẫu trong mỗi lần huấn luyện.

Minh họa: Trong quá trình huấn luyện, các từ mục tiêu và từ ngữ cảnh được chọn kèm với các từ mẫu âm (negative samples) để cập nhật các biểu diễn vector sao cho mô hình dự đoán chính xác hơn.

4 Đánh giá Word Vectors

Đối với câu hỏi "a:b::c:?": Mối quan hệ của từ a tương đường với từ b thì sẽ như mối quan hệ của từ c tương đương với gì. Ví dụ: Nếu chúng ta có "man:woman::king:?", mối quan hệ giữa "man"và "woman"là mối quan hệ giới tính (gender relationship), nghĩa là "man"tương đương với "woman"như "king"tương đương với gì? Trong phương pháp tích vô hướng, chúng ta sử dụng công thức sau để đánh giá word vectors:

$$d = \text{argmax}_i \left(\frac{(x_b - x_a + x_c)^T x_i}{||x_b - x_a + x_c||} \right)$$

Trong đó:

- Từ "d"là từ cần tìm
- x_a, x_b, x_c là các vector biểu diễn của các từ đã biết "a", "b"và "c".
- x_i là các biểu diễn vector của các từ trong từ điển.
- argmax_{x_i} là toán tử argmax, chọn ra x_i sao cho giá trị của $\frac{(x_b - x_a + x_c)^T x_i}{||x_b - x_a + x_c||}$ là lớn nhất.

5 Quá trình thực hiện

5.1 Bộ dữ liệu

Bộ dữ liệu Stanford Sentiment Treebank- là một kho văn bản có các cây phân tích được dán nhãn đầy đủ cho phép phân tích đầy đủ các tác động cấu thành của cảm xúc trong ngôn ngữ. Kho ngữ liệu này dựa trên tập dữ liệu do Pang và Lee (2005) giới thiệu và bao gồm 11.855 câu đơn được trích từ các bài phê bình phim. Nó được phân tích cú pháp bằng trình phân tích cú pháp Stanford và bao gồm tổng cộng 215.154 cụm từ duy nhất từ các cây phân tích cú pháp đó, mỗi cụm từ được chú thích bởi 3 giám khảo con người.

5.2 Huấn luyện mô hình

Kết hợp sử dụng SGD với mô hình word2vec, với các tham số như sau:

- bath size: 50
- learning rate: 0.3
- iterations: 40000
- context window size: 5

Lưu check point với mỗi lần iteration đạt 5000, 10000, 15000,...400000.

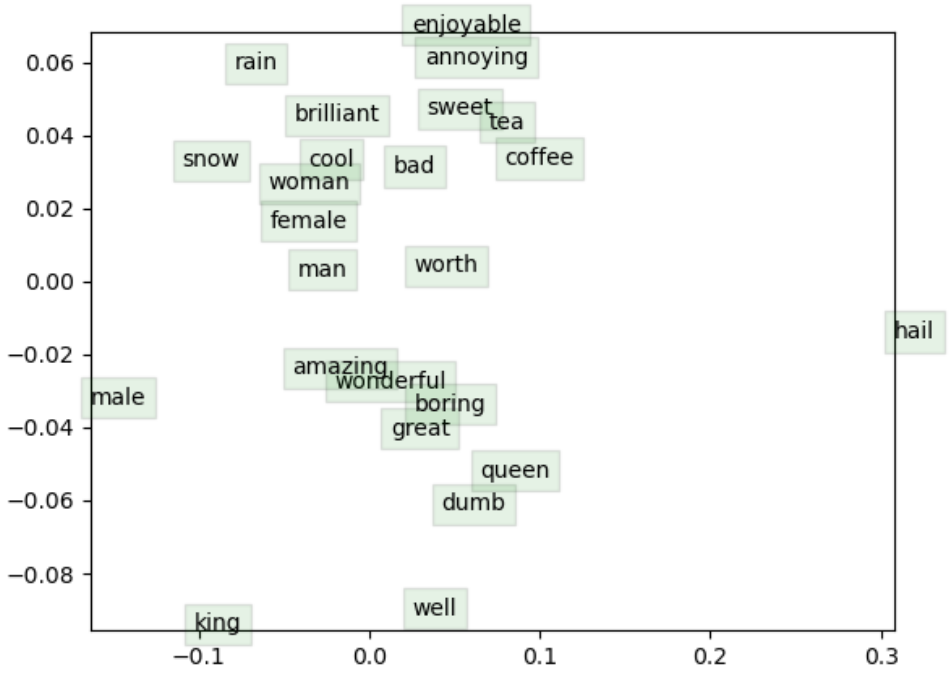
Source code thầy có thể xem trong link github gửi đính kèm.

5.2.1 Kết quả

Ta được biểu diễn các vector từ như sau đối với những từ "great", "cool", "brilliant", "wonderful", "well", "amazing", "worth", "sweet", "enjoyable", "boring", "bad", "dumb", "annoying", "female", "male", "queen", "king", "man", "woman", "rain", "snow", "hail", "coffee", "tea"

6 Nhận xét

- **Các từ có ngữ cảnh sử dụng tương tự hoặc liên quan đến nhau thì sẽ nằm gần nhau.Ví dụ:**
  - "enjoyable," "annoying," "sweet," "tea," "coffee" được nhóm lại gần nhau, gợi ý rằng chúng có những điểm tương đồng trong ngữ cảnh sử dụng.
  - "amazing," "wonderful," và "great" cũng nằm gần nhau, cho thấy chúng có thể được sử dụng trong các ngữ cảnh tương tự.
- **Các thuật ngữ liên quan đến giới tính**:
  - Các từ như "male" và "female" được đặt riêng biệt, nhưng cả hai đều gần các từ như "man" và "woman," chỉ ra rằng mô hình đã nắm bắt được sự khác biệt về giới tính.
  - "king" và "queen" gần nhau, phản ánh mối quan hệ của chúng như là các từ đối ngẫu.



Hình 2: Vector biểu diễn ma trận kết quả

- **Sắc thái tích cực và tiêu cực**:
  - Các từ có sắc thái tích cực như "brilliant," "amazing," "wonderful," và "great" được nhóm lại với nhau.
  - Các từ tiêu cực như "boring" và "dumb" cũng nằm trong một cụm riêng.
- **Thuật ngữ liên quan đến thời tiết**:
  - Các từ liên quan đến thời tiết như "rain," "snow," và "hail" được đặt riêng biệt, có thể phản ánh các loại thời tiết khác nhau nhưng có cách sử dụng ngữ cảnh tương tự.
- **Từ ngữ trung lập**:
  - Các từ như "well" và "worth" nằm riêng biệt, có thể do sự sử dụng ngữ cảnh trung lập hoặc đa dạng.

7 Kết luận

Mô hình Word2Vec đã nắm bắt hiệu quả các mối quan hệ ngữ nghĩa giữa các từ, như thể hiện qua các mẫu cụm trong biểu đồ. Các từ có ý nghĩa tương tự hoặc cách sử dụng ngữ cảnh tương tự được đặt gần nhau, cho thấy khả năng của Word2Vec trong việc học các đại diện từ ngữ có ý nghĩa từ dữ liệu văn bản mà nó đã được huấn luyện. Biểu đồ trên có thể giúp hiểu rõ hơn về cách mô hình nhận thức các mối quan hệ giữa các từ khác nhau và có thể hữu ích cho các nhiệm vụ tiếp theo như cải thiện các embedding từ hoặc phân tích các cụm cụ thể để có cái nhìn sâu sắc hơn.