

Những mục đầu (1, 2, 3 và một phần nhỏ của 4) được ghi chép từ bài học đầu tiên của khóa CS224N của Standford (Youtube). Những phần còn lại từ nhiều nơi.

### 1. Từ - biểu diễn dưới dạng rời rạc:

- Những từ như “Seattle motel” và “Seattle hotel” đều chỉ đến một khái niệm: “Khách sạn Seattle”.

- **One-hot encoding** là phương pháp biểu diễn thưa thớt (sparse representation) đối với dữ liệu.
- Nhãn là một vector chỉ bao gồm giá trị 1 và 0. Trong đó, nhãn của điểm dữ liệu được đặt là 1, phần còn lại đều là giá trị 0.
- **One-hot encoding** không chỉ giúp mã hóa nhãn thành một chuỗi số cho máy tính, mà nó còn có ý nghĩa về mặt tính toán khi nhân ma trận / vector: giúp tập trung vào nhãn đang xét (với giá trị là 1) và bỏ qua những nhãn còn lại (với giá trị là 0).
- **One-hot encoding** thường được kết hợp với hàm argmax() hoặc argmin(). Khi cập nhật nhãn ở K-means, ta dùng các phép tính ma trận để có được một vector khoảng cách từ điểm đang xét đến từng tâm điểm đang có, sau đó dùng argmin() để chọn ra tâm điểm gần nhất và gán nó bằng 1; các nhãn còn lại thì gán giá trị 0.
- **One-hot encoding** được dùng làm nhãn cho các mô hình phân loại như Softmax Regression, các mô hình học sâu,...

#### Hình 1. Về One-hot encoding (file K-mean).

- Tuy nhiên, khi ta biểu diễn chúng dưới dạng One-hot encoding (hình 1):
  - “motel” = [0, 0, 1, 0, 0]
  - “hotel” = [1, 0, 0, 0, 0]
  - + Đây là hai vector trực giao.
  - + Vector one-hot encoding không thể hiện rõ sự giống nhau giữa các từ vựng.
  - + Số chiều của vector sẽ tương đương số từ vựng (rất lớn).

### 2. Ngữ nghĩa phân phối (Distributional Semantic):

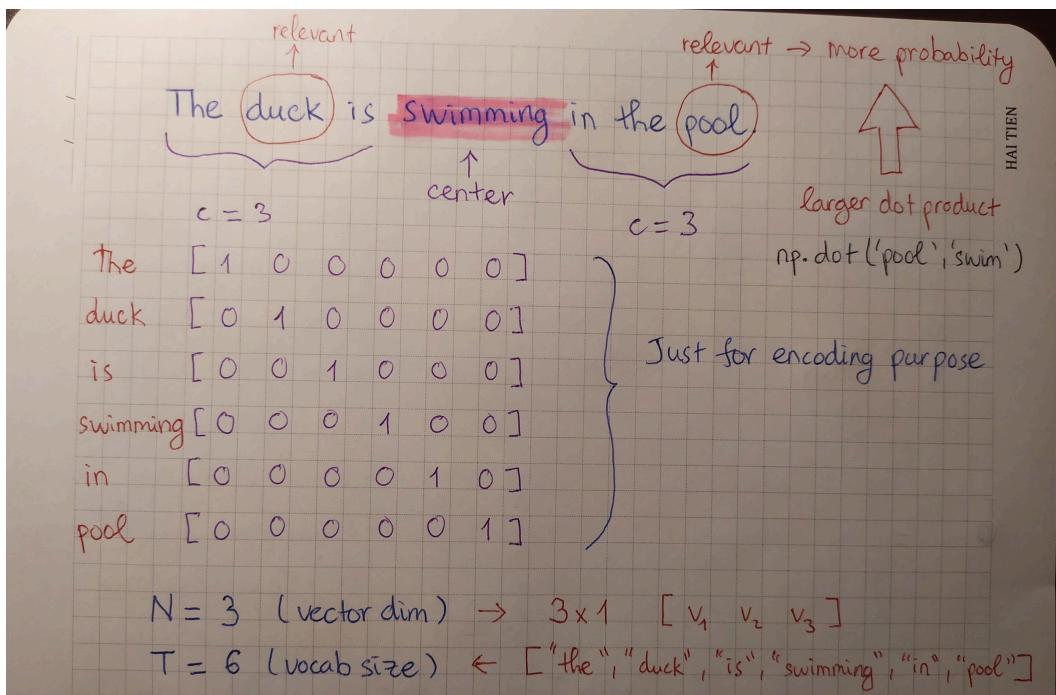
- Đối với phương pháp này, nghĩa của từ được xác định bởi những từ xuất hiện gần nó nhất.
- Khi một từ  $w$  xuất hiện trong văn bản, ngữ cảnh (context) của nó là tập hợp những từ nằm gần nó (trong một context window được định sẵn).

### 3. Word embedding:

- Với phương pháp “nhúng” từ (word embedding), vector từ sẽ là vector dày đặc (dense vector) và bao gồm các số thực.
- AKA word vectorization hoặc (neural) word representation.

### 4. Word2Vec - Skip gram:

- Word2Vec (Mikolov et al., 2013) là một mô hình có thể học các vector từ.
- Word2Vec bao gồm 2 mô hình chính là Skip gram và CBOW.
  - + Chúng ta sẽ tìm hiểu mô hình Skip-gram đầu tiên.
  - + Mô hình Skip-gram sẽ dùng vector center (còn gọi là vector target) để tính toán ra vector context trong quá trình huấn luyện.
  - + Ngược lại, mô hình CBOW sẽ dùng vector context để tính toán vector target.



Hình 2. Đầu vào của Skip-gram sẽ là một từ center, nhằm dự đoán các từ context xung quanh nó.

- Ta gọi:
  - + c là cửa sổ ngữ cảnh (với c = 2 thì ta sẽ lấy 4 từ, bao gồm 2 từ bên trái và 2 từ bên phải). Với một context window, ta sẽ có nhiều vị trí, mỗi

vị trí gọi là một panel. Phần tính toán gradient ta sẽ gọi C là tổng số panel.

- + T là số từ vựng. Phần tính toán gradient ta sẽ gọi số từ vựng là W.
- + N là số chiều của vector từ.

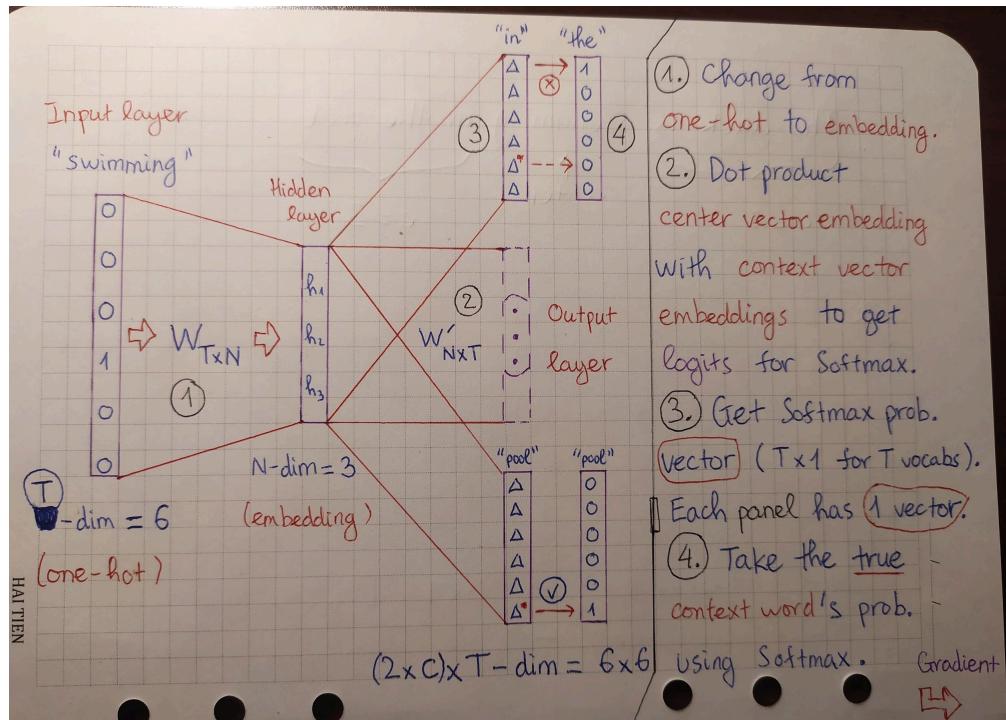
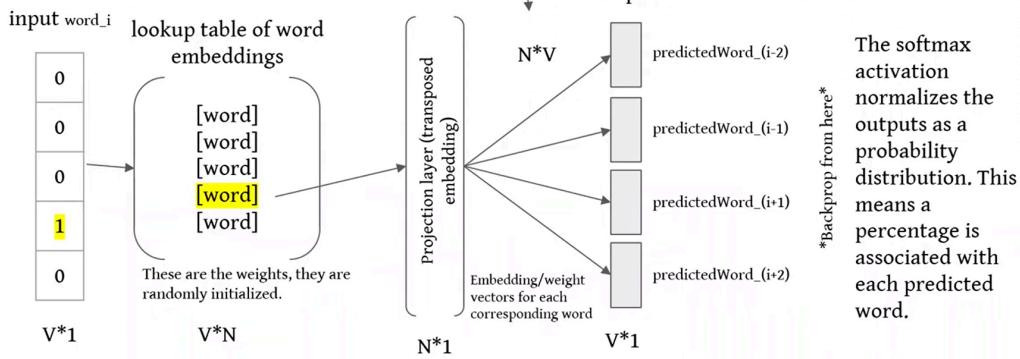
### a) Pass forward:

A closer look...

The probability of a predicted word occurring given a center word:

$$P(\text{predictedWord}_n | \text{centerWord}) = \frac{e^{\text{dot}(\text{predictedWord}_n, \text{centerWord})}}{\sum_i e^{\text{dot}(\text{predictedWord}_i, \text{centerWord})}}$$

One hot vector in:



**Hình 3. Kiến trúc mô hình Skip-gram (ảnh dán by Fu).**

- **One-hot encoding đầu vào:** Đầu vào Input cho mô hình sẽ là một từ center dưới dạng one-hot encoding.
- **Chuyển thành vector center:**
  - Ta nhân pair-wise vector one-hot encoding với ma trận “input → hidden”:  

$$h = x * W$$
  - Ma trận “input → hidden” hay **ma trận W** (Xin, 2016):
    - + Chiều:  $T \times N$ .
    - + Hàng: Có  $T$  hàng, đại diện cho  $T$  từ vựng.
    - + Cột: Mỗi hàng có  $N$  giá trị, tức là  $N$  chiều của mỗi vector embedding.
    - + Các vector trong ma trận này là vector của các từ khi chúng đóng vai trò là vector center.
  - Kết quả của bước này là ta lấy được vector center của từ input, với số chiều  $N$ .
- **Nhân vô hướng vector target với lần lượt các vector context có trong từ vựng:**
  - Sau khi có được vector center rồi, ta **nhân vô hướng** (dot product) vector center với các vector context (trong ma trận “hidden → output”) **từng vị trí trong window context**.
  - Ma trận “hidden → output” hay **ma trận W'** (Xin, 2016):
    - + Có số chiều là  $N \times T$ .
    - + Về cơ bản là ma trận transposed của ma trận “input → hidden”.
  - Mục đích của transpose chủ yếu là vì:
    - + Để thực hiện phép nhân vô hướng ma trận A với B thì số chiều của hai ma trận phải là  $A^{a \times b}$  và  $B^{b \times c}$ . Trong đó, chiều  $b$  bằng nhau.
    - + Vector nhân với **ma trận W** là vector one-hot encoding với chiều là  $T$  (số từ vựng). Do đó **ma trận W** có chiều là  $T \times N$ .
    - + Mặt khác, vector nhân với **ma trận W'** là vector center (đã được embed) với chiều là  $N$  (số chiều). Do đó **ma trận W'** có chiều là  $N \times T$ .
  - Kết quả của bước **nhân vô hướng** này là **C** vector logit với chiều  $T$  (ứng với **mỗi panel**, nhưng đều là một - do nhân với cùng một vector center):
    - + Đây là  $T$  giá trị logit ứng với mỗi từ trong từ vựng lấy từ dot product.
    - + Phép nhân vô hướng sẽ có giá trị lớn nhất khi hai vector có cùng hướng, tức  $\cos(0^\circ) = 1$ . Từ đây, ta thấy hai từ càng có nhiều điểm tương đồng với nhau thì giá trị phép nhân vô hướng càng cao. Ngoài

ra, dot product còn có thể xét đến **độ lớn** của vector (vector càng lớn thì tầm quan trọng càng lớn và tần suất xuất hiện càng nhiều).

- + Ta sẽ đưa các logit ấy vào hàm softmax.

- **Đưa các giá trị nhân vô hướng vào hàm softmax để tính giá trị:**

- Ta sẽ đưa vector logit (bao gồm các giá trị trong khoảng  $-\infty$  đến  $+\infty$ ) vào hàm Softmax để ánh xạ thành vector xác suất (bao gồm các giá trị trong khoảng 0 đến 1, với tổng bằng 1). Mỗi panel (vị trí trong context window) sẽ có một vector như thế (do tất cả share chung **vector center**). Đầu ra sẽ là ma trận  $C \times T$  (với  $C = 2 * c$  và  $T$  là số từ vựng).

*Ta sẽ tạm dùng hàm Softmax cho việc học lí thuyết. Nhưng trong thực tế, và trong cả bài báo gốc của Word2Vec (Mikolov et al., 2013), ta sẽ dùng Hierarchical Softmax. Bài viết này tạm chưa đề cập Hierarchical Softmax.*

$$\frac{\exp(u_i^T \cdot v)}{\sum_{j=1}^w (u_j^T \cdot v)}$$

**Hình 4. Giá trị  $p(w_O | w_I)$  tính thông qua Softmax.**

- **Giải thích:**

- + Kết quả  $p(w_O | w_I)$  : xác suất từ context  $w_O$  khi ta biết từ center  $w_I$  ( $O$  là Output,  $I$  là Input).
- +  $u_i^T$ : Vector embedding của từ context  $w_O$ . Ta thực hiện transpose vector để thực hiện nhân vô hướng (chuyển từ  $N \times 1$  thành  $1 \times N$ )
- +  $v$  : Vector embedding của từ center  $w_I$ .
- +  $u_i^T \cdot v$  : là giá trị logit ta vừa đề cập đối với từ context  $w_O$ .

**b) Gradient:**

- **Ta gọi:**

- +  $wI$ : là từ center.
- +  $C$  là tổng số panel context (tức ta có  $C$  từ context).
- +  $W$ : tổng số từ **HOẶC** ma trận embed của từ center (input  $\rightarrow$  hidden).
- +  $W'$ : ma trận embed của từ context (hidden  $\rightarrow$  output).
- +  $S$ : đầu ra hàm softmax.
- +  $v$ : vector center (input).
- +  $u$ : vector context (output).
- +  $\exp$ : hàm exponential ( $e^x$ ).

$$\begin{aligned}
 L &= -\log P(\widehat{\omega}_1, \widehat{\omega}_2, \dots, \widehat{\omega}_c | \omega_I) \\
 &= -\log \prod_{i=1}^c P(\widehat{\omega}_i | \omega_I) \\
 &= -\log \prod_{i=1}^c \frac{\exp(u_i^\top \cdot v)}{\sum_{j=1}^w (u_j^\top \cdot v)} \\
 &= -\sum_{i=1}^c \log \left[ \frac{\exp(u_i^\top \cdot v)}{\sum_{j=1}^w (u_j^\top \cdot v)} \right]
 \end{aligned}$$

Let  $d = u^\top \cdot v$ .

$$\Rightarrow L = -\sum_{i=1}^c \log \left[ \frac{\exp(d_i)}{\sum_{j=1}^w \exp(d_j)} \right]$$

According to the chain rule:

$$\frac{\partial L}{\partial d_i} = \frac{\partial L}{\partial s_i} \cdot \frac{\partial s_i}{\partial d_i}$$

①  $\frac{\partial L}{\partial s_i}$ :

$$\begin{aligned}
 \frac{\partial L}{\partial s_i} &= \frac{\partial}{\partial s_i} \cdot \left( -\log \prod_{j=1}^c s_j \right) \\
 &= \frac{\partial}{\partial s_i} \cdot (-\log s_i)
 \end{aligned}$$

$$= -\frac{1}{s_i}, \text{ with label } = 1$$

HAI TIEN

Hình 5a. Tính đạo hàm (by Fu).

1) :

ec

$$\begin{aligned}\frac{\partial L}{\partial s_i} &= \frac{\partial}{\partial s_i} \left[ -\log(1-s_i) \right] \\ &= -\frac{-1}{1-s_i} = \frac{1}{1-s_i}, \text{ with label}=0.\end{aligned}$$

②  $\frac{\partial s_i}{\partial d_i}$  :

$$\begin{aligned}\frac{\partial s_i}{\partial d_i} &= \frac{\partial}{\partial d_i} \cdot \frac{\exp(d_i)}{\sum_{j=1}^w \exp(d_j)} \\ &= \frac{[\exp(d_i)]' \cdot \sum_{j=1}^w \exp(d_j) - \exp(d_i) \cdot [\sum_{j=1}^w \exp(d_j)]'}{\left[ \sum_{j=1}^w \exp(d_j) \right]^2} \\ &= \frac{\exp(d_i) \sum_{j=1}^w \exp(d_j) - \exp(d_i) \cdot \exp(d_i)}{\left[ \sum_{j=1}^w \exp(d_j) \right]^2} \\ &= \frac{\exp(d_i) [\sum_{j=1}^w \exp(d_j) - \exp(d_i)]}{\left[ \sum_{j=1}^w \exp(d_j) \right]^2} \\ &= s_i \cdot (1 - s_i) \\ \Rightarrow \frac{\partial L}{\partial d_i} &= \frac{\partial L}{\partial s_i} \cdot \frac{\partial s_i}{\partial d_i} \\ &= \begin{cases} s_i - 1, & \text{with label} = i \\ s_i, & \text{with label} = 0 \end{cases}\end{aligned}$$

Hình 5b. Tính đạo hàm (by Fu).

After having  $\frac{\partial L}{\partial d_i}$ , which is the loss

gradient with respect to  $d_i$  - the logit got from the dot product of  $u_i^T$  and  $v$ , we

"define a  $W$ -dimensional vector (1)

$E = [E_1, E_2, \dots, E_w]$  as the sum of prediction over all context panels (positions in context window).

$$E_j = \sum_{i=1}^c \frac{\partial L}{\partial d_i} = \begin{cases} \sum_{i=1}^c (s_i - 1), & \text{label=1} \\ \sum_{i=1}^c s_i, & \text{label=0} \end{cases}$$

We have the gradient in  $W'$  matrix ( $u$ ).

$$\begin{aligned} \frac{\partial L}{\partial u} &= \frac{\partial L}{\partial d} \cdot \frac{\partial d}{\partial u} \\ &= E \cdot \frac{\partial}{\partial u} (u^T v) \\ &= E \cdot v \end{aligned}$$

We will use this gradient to calculate the update of vectors in  $W'$  matrix.

HAI TIEN

**Hình 5c. Tính đạo hàm (by Fu).**

$$w_{new} = w_{old} - \mu \cdot E \cdot v$$

- $\mu$ : learning rate.
- $E$ : prediction error
- $v$ : center vector used for update.

The intuition behind this is that:

- + When the word is the true context,  $E = \sum_{i=1}^c (s_i - r)$ , which means  $E$  is always negative, and the update will add "some  $v$ " to context vector, make it closer to vector  $v$  (Xin, 2016).
- + Vice versa.

We have: Let  $F$  be the update ~~matrix~~  
vector for input  $\rightarrow$  hidden layer's matrix (center  
vectors).

$$F = \sum_{j=1}^w (E_j \cdot u_{i,j})$$

$F$  has  $N$  dimensions.

$$c_{new} = w_{old} - \mu \cdot F$$

Hình 5d. Tính đạo hàm (by Fu).