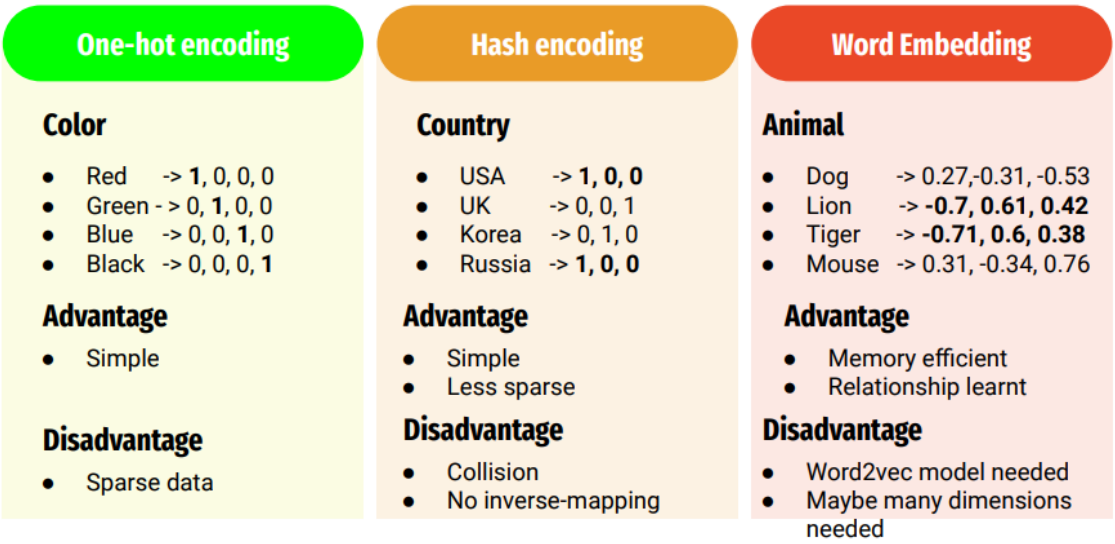
Word embedding

1.Biểu diễn từ bằng vector

Máy tính không hiểu **từ ngữ**, **chữ cái**, hay **ngôn ngữ tự nhiên**. Nó chỉ hiểu **số**.

Vậy nên nếu ta muốn máy tính hiểu từ *"king", "queen", "apple", "banana"* thì ta phải biến chúng thành **vector số**.



One-hot encoding hay hash encoding là những lựa chọn tệ

2.Word embedding

Word Embedding là kỹ thuật ánh xạ mỗi từ thành một vector nhiều chiều (ví dụ: 50, 100 hoặc 300) sao cho:

- Các từ có nghĩa giống nhau → vector gần nhau
- Các quan hệ ngữ nghĩa có thể biểu diễn bằng phép toán vector

Ngôn ngữ tự nhiên là một hệ thống phức tạp được dùng để biểu đạt ý nghĩa. Trong hệ thống này, **từ** là đơn vị cơ bản của ý nghĩa. Như tên gọi, **word vector** là các vector dùng để biểu diễn từ, và cũng có thể được xem như **feature vector** hoặc **biểu diễn của từ**.

Kỹ thuật ánh xạ từ sang vector thực được gọi là **word embedding**. Trong những năm gần đây, **word embedding** đã dần trở thành kiến thức cơ bản trong xử lý ngôn ngữ tự nhiên.

3.Word2Vec

Word2Vec là một trong những phương pháp đầu tiên và nổi tiếng nhất để học Word Embedding.

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.

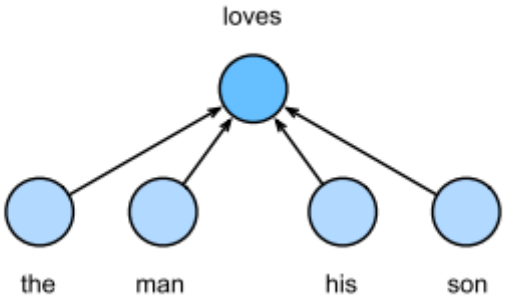
Có hai cách khác nhau xây dựng mô hình word2vec:

- Skip-gram: Dự đoán những từ xung quanh dựa vào một từ cho trước
- CBOW (Continuous Bag of Words): Dự đoán từ mục tiêu từ những từ xung quanh

3.1. Skip-gram

Ví dụ, với chuỗi từ: “the”, “man”, “loves”, “his”, “son”.
Chọn “loves” làm từ trung tâm và đặt kích thước cửa sổ ngữ cảnh bằng 2. Với từ trung tâm “loves”, skip-gram xem xét xác suất có điều kiện để sinh ra các từ ngữ cảnh: “the”, “man”, “his”, “son”, tức là những từ cách không quá 2 từ so với từ trung tâm:

$$P(\text{"the"}, \text{"man"}, \text{"his"}, \text{"son"} \mid \text{"loves"}).$$



Giả định rằng các từ ngữ cảnh được sinh ra độc lập với nhau dựa trên từ trung tâm (độc lập có điều kiện). Khi đó, xác suất trên có thể viết lại thành:

$$P(\text{"the"} \mid \text{"loves"}) \cdot P(\text{"man"} \mid \text{"loves"}) \cdot P(\text{"his"} \mid \text{"loves"}) \cdot P(\text{"son"} \mid \text{"loves"})$$

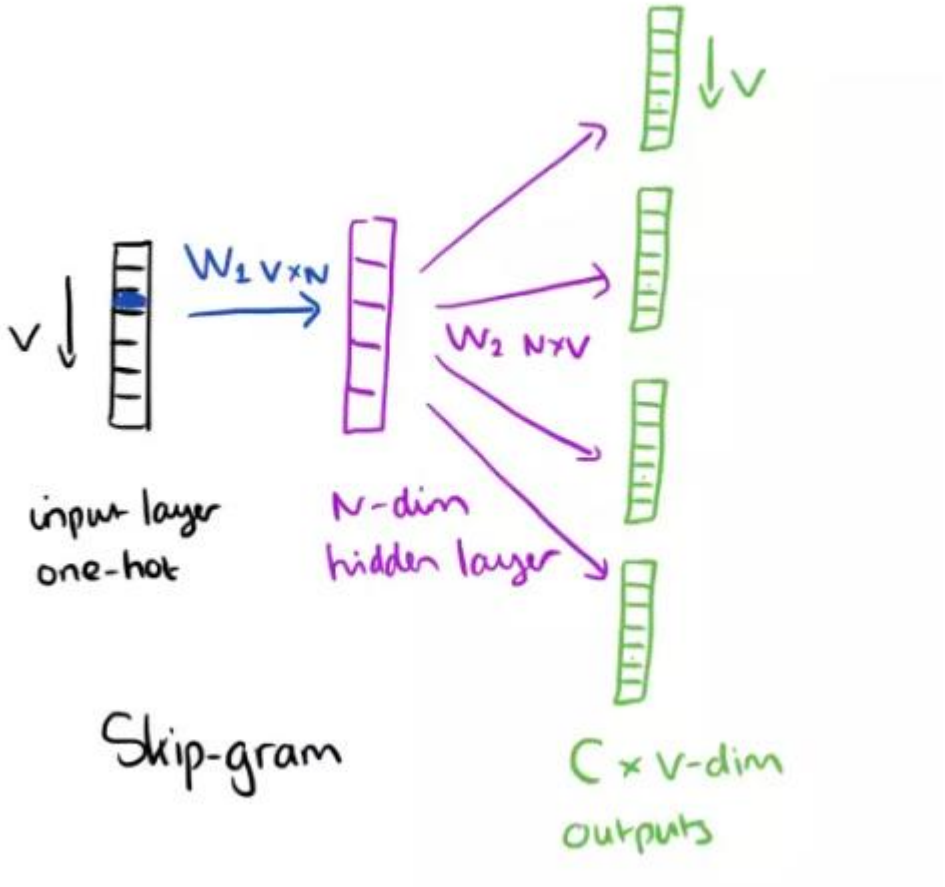
Trong skip-gram, mỗi từ có hai vector □ chiều:

- $v_i \in R^d$: vector khi là **center word**
- $u_i \in R^d$: vector khi là **context word**

Xác suất sinh một context word w_o (chỉ số o trong từ điển) từ center word w□ (chỉ số c) được mô hình hóa bằng softmax trên tích vô hướng:

$$P(w_o \mid wc) = \frac{\exp(u_o^T vc)}{\sum_{i \in V} \exp(u_i^T vc)}.$$

Biểu diễn dưới dạng mạng neural:



Hàm Loss (negative log-likelihood):

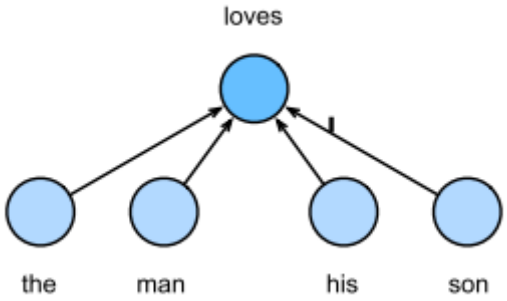
$$\mathcal{L}_{Skip} = -\sum_{j=-m, j \neq 0}^{m} \log P(w_{t+j} \mid w_t)$$

3.2. CBOW (Continuous Bag of Words)

Mô hình CBOW ngược lại so với skip-gram

Ví dụ, với chuỗi “the”, “man”, “loves”, “his”, “son”, từ trung tâm là “loves”, kích thước cửa sổ ngữ cảnh = 2. CBOW sẽ xem xét xác suất có điều kiện:

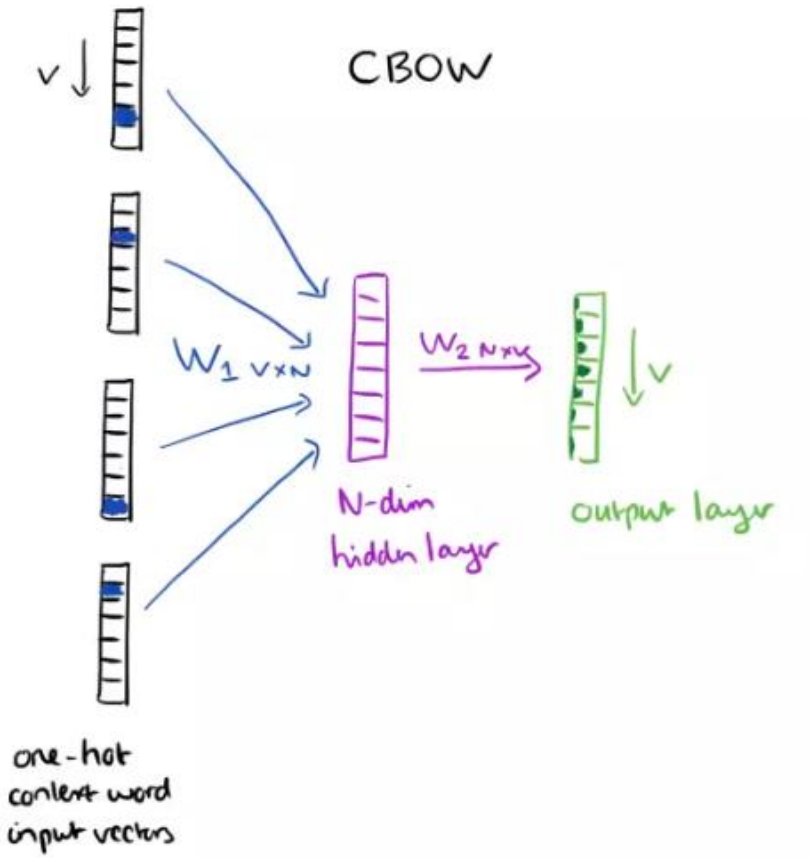
$$P(\text{"loves"} \mid \text{"the"}, \text{"man"}, \text{"his"}, \text{"son"}).$$



Trong CBOW, các vector context được lấy trung bình rồi đưa vào softmax. Sau huấn luyện, **vector context word** thường được dùng làm biểu diễn từ.

$$P(w_c \mid \mathcal{W}_o) = \frac{\exp\left(\mathbf{u}_c^\top \bar{\mathbf{v}}_o\right)}{\sum_{i \in \mathcal{V}} \exp\left(\mathbf{u}_i^\top \bar{\mathbf{v}}_o\right)}.$$

Biểu diễn dưới dạng mạng neural:



Hàm Loss (negative log-likelihood):

$$\mathcal{L}_{CBOW} = -\log P(w_c \mid w_{t-m}, \ldots, w_{t+m})$$

4.GloVe (Global Vectors for Word Representation)

Là một phương pháp Word Embedding do Stanford phát triển (2014), kết hợp **ưu điểm của Word2Vec** (mô hình dự đoán) và **ưu điểm của TF-IDF / thống kê đồng xuất hiện** (mô hình đếm).

GloVe dựa trên **ma trận đồng xuất hiện** (co-occurrence matrix), do đó GloVe mang tính “toàn cục” hơn là Word2vec vì GloVe tính toán xác suất từ dựa trên toàn bộ tập dữ liệu còn Word2vec học dựa trên các ngữ cảnh đơn lẻ

	king	queen	man	woman	cat
king	—	53	113	2	1
queen	47	—	1	105	0
dog	0	0	0	0	200

Mô hình:

GloVe xây dựng từ **ma trận đồng-xuất-hiện** X với phần tử Xij = số lần từ j xuất hiện trong ngữ cảnh của từ i (cửa sổ trượt).

Với một cặp có $X_{ij} > 0$:

1. Lấy $w_i,\ \tilde{w}_j,\ b_i,\ \tilde{b}_j$.
2. Dự đoán: $\hat{s}_{ij} = w_i^\top \tilde{w}_j + b_i + \tilde{b}_j$.
3. Target: $t_{ij} = \log X_{ij}$.
4. Error: $e_{ij} = \hat{s}_{ij} - t_{ij}$.
5. Loss cho cặp: $\ell_{ij} = f(X_{ij}) \cdot e_{ij}^2$.

Toàn bộ loss là tổng $\sum_{i,j} \ell_{ij}$.

Với f(Xij): trọng số để:

- **Giảm ảnh hưởng các cặp cực hiếm** (đếm = 1) vì chúng noisy.
- **Không để các cặp siêu phổ biến** áp đảo (ví dụ "the", "of").

$$f(x) = \begin{cases} \left(\frac{x}{x_{\max}}\right)^\alpha & \text{if } x < x_{\max}, \\ 1 & \text{otherwise.} \end{cases}$$

Thông thường x_max=100, a=3/4

Hàm Loss:

Hàm mất mát (loss) tiêu chuẩn của GloVe là **weighted least squares:**

$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij}) \left(w_i^\top \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij} \right)^2.$$