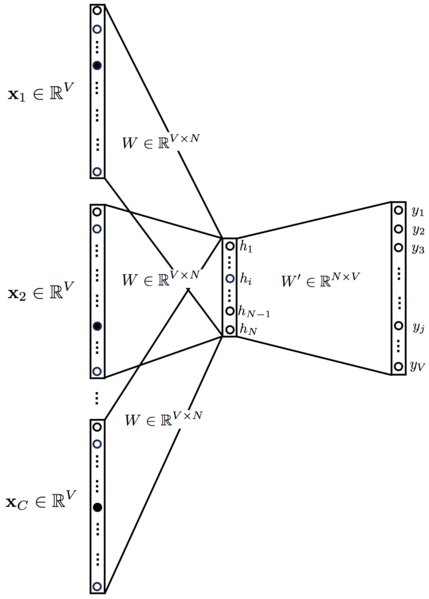
1. **CBOW *(Continous Bag Of Words):***

Đễ dễ dàng tiếp cận vấn đề, ta có ví dụ một câu đã được tách ra như sau: {"The", "cat", ’over", "the’, "puddle"}

Dựa vào ngữ cảnh của câu trên và vị trí của các từ được phân bố trong câu. Ta có thể dự đoán rằng câu trên thiếu một từ và vị trí của nó là ở giữa đó là “junped”. Và mô hình dùng để dự đoán mô hình trên chính là CBOW.

CBOW được xây dựng dựa trên mô hình neural 1 tầng ẩn:

* **Input:** Là một tập các onehot vectors hoặc ngữ cảnh được buổi diễn dưới dạng x(c).
* **Output:** Ở đây ta chỉ có duy nhất 1 đầu ra tương ứng là y(c)



Tiếp thao ta cho 2 ma trận đầu vào và đầu ra như sau:

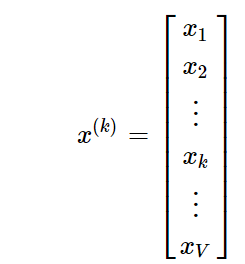
Trong đó:

* là kích thước của không gian nhúng *(embedding space) hay là kích thước của hidden layer, các unit thuộc các layer cận nhau được kết nối theo kiểu fully connected*
* là ma trận các từ đầu vào sao cho cột thứ của là vectơ nhúng n chiều cho từ khi nó là đầu vào của mô hình
* là đầu ra của ma trận sao cho cột thứ của là vectơ nhúng n chiều cho từ khi nó là đầu ra của mô hình, tương tự mỗi hàng của là

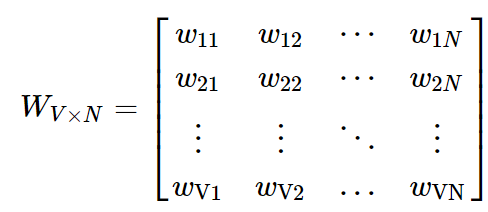
*Mô tả quá trình hoạt đông:*

* Tạo *các onehot vector của các từ đầu vào xung quoanh từ trung tâm: cho* ngữ cảnh *(context)* đầu vào với kích thước

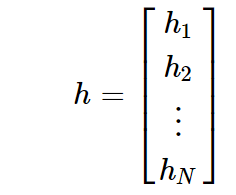
*Một từ đầu vào vị trí thứ k được biểu diễn dưới dạng one –hot như sau, trong đó chỉ có xk có giá trị bằng 1 còn lại đều bằng 0.*



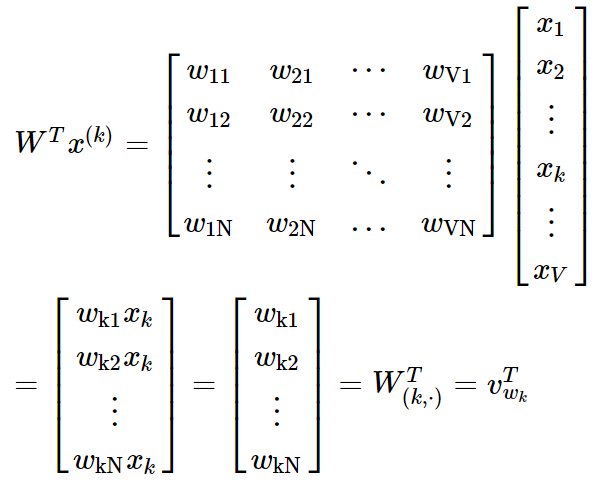
* Khởi tạo các ma trận trọng số giữa input layer và hidden layer với kích thước mỗi ma trận là *VxN,* trong đó mỗi hàng tương ứng với một từ trong tập từ vựng,



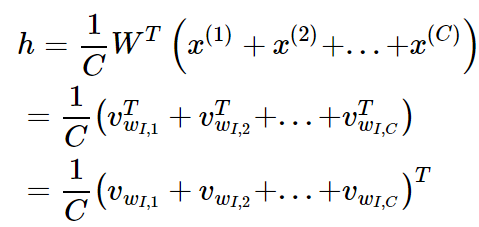
* Tiếp theo ta sẽ định nghĩa các ma trận h tương ứng với đầu ra của mổi hidden unit với kích thước ***N x 1*** có dạng như sau:



* Tiếp theo ta sẽ tiến hành đưa các ma trận ***h*** này qua các hàm activation (ta sử dụng các hàm tuyến tính): 

**9**

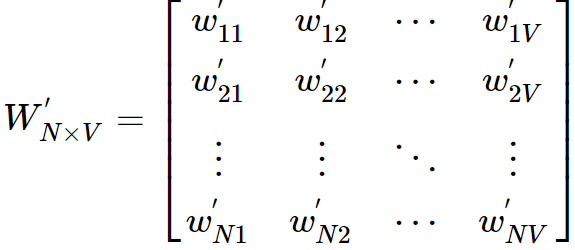
* Áp dụng công thức phía trên vào ***h*** ta khai triển được công thức sau:



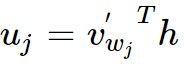
*Trong đó:*

*Các*  tương ứng với các từ ngữ cãnh đầu vào  và Wo tương ứng đầu ra dự đoán của mô hình.

Quay lại ta có, ma trận trọng số kích thước N x V từ hidden layer đến output layer có dạng như sau:



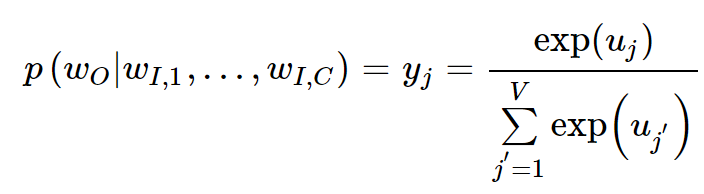
Chúng ta sẽ thực hiện việc tính score ***uj*** tương ứng với mỗi từ thứ *jththeo công thức sau:*



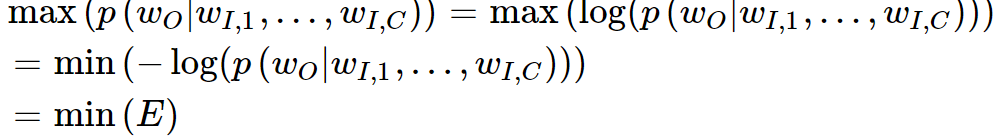
Trong đó:

*  *là ma trận tương ứng với cột thứ j trong ma trận W’*
* *Uj là độ đo khớp giữa C đầu vào và* ***wj***

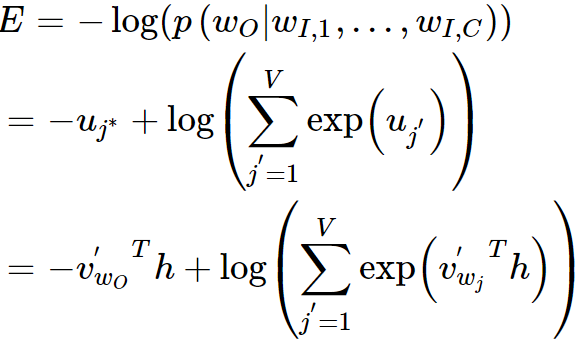
Tuy nhiên do ta sử dụng gradient descent để điều chỉnh trong số nên mô hình sẽ ưu tiên sử dụng hàm softmax để chuẩn hóa đầu ra.



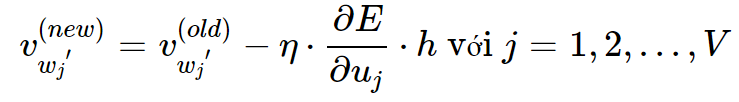
Mục tiêu của mô hình là cực đại hóa trọng số giữa từ dự đoán wo so tương ứng với các từ ngữ cảnh cho trước, thể hiện qua công thức:



Tiếp theo ta sẽ tiến hành minimize loss function tương ứng ở đây là giá trị cross-entropy, với J\* là vị trí hiện tại của wo



Chúng ta sẽ sử dụng gradient descent Để cập nhật trọng số từ hidden layer đến output layer với phương trình sau:



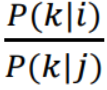
Trong đó:

*  là input vector tương ứng của từ thứ c trong C từ ngữ cành đầu vào.
* là learning rate



***Glove***

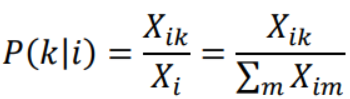
**GloVe** *(Global Vector)* là một trong những phương pháp mới để xây dựng vec-tơ từ *(được giới thiệu vào năm 2014)*, nó thực chất được xây dựng dựa trên Co-occurrence Matrix. GloVe có bản chất là xác suất, ý tưởng xây dựng phương pháp này đến từ tỉ số sau:

 (1)

Trong đó:

* ***P(k|i)*** là xác suất xuất hiện của từ ***k*** trong ngữ cảnh của từ ***i*** , tương tự với ***P(k|j)*** .

Công thức của ***P(k|i)*** :



Trong đó:

* ***Xik*** : số lần xuất hiện của từ k trong ngữ cảnh của từ i (hoặc ngược lại).
* ***Xi :*** số lần xuất hiện của từ i trong ngữ cảnh của toàn bộ các từ còn lại ngoại trừ i.

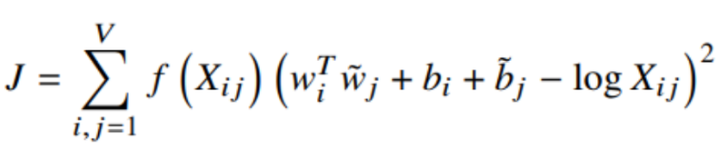
(Các giá trị này chính là các mục nhập của Co-occurrence Matrix)

Ý tưởng chính của ***GloVe***: độ tương tự ngữ nghĩa giữa hai từ **i, j** có thể được xác định thông qua độ tương tự ngữ nghĩa giữa từ ***k*** với mỗi từ ***i, j***, những từ **k** có tính xác định ngữ nghĩa tốt chính là những từ làm cho ***(1) >>1*** hoặc xấp chỉ bằng 0. Ví dụ, nếu *i* là *“table”*, *j* là *“cat”* và k là “chair” thì (1) sẽ khá lớn do “chair” có nghĩa gần hơn với “table” hơn là “cat”

ở trường hợp khác, nếu ta thay k là “ice cream” thì (1) sẽ xấp xỉ bằng 1 do “ice cream” hầu như chẳng lên quan gì tới “table” và “cat”.

Dựa trên tầm quan trọng của tỉ số ***(1),*** GloVe khởi đầu bằng việc là nó sẽ tìm một hàm F sao cho nó ánh xạ từ các vec-tơ từ trong vùng không gian V sang một giá trị tỉ lệ với ***(1)*** .

Việc tìm F không đơn giản, tuy nhiên, sau nhiều bước đơn giản hóa cũng như tối ưu, ta có thể đưa nó về bài toán hồi quy với việc ***minimum cost functio***n sau:



Trong đó:

* ***wi***,***wj***là các vector từ.
* ***bi***, ***bj*** là các ***bias*** tương ứng (được thêm vào ở các bước đơn giản hóa và tối ưu).
* ***Xij***: mục nhập tương ứng với cặp từ ***i***,***j*** trong ***Co-occurrence Matrix***.
* Hàm ***f*** được gọi là ***weighting function***, được thêm vào để giảm bớt sự ảnh hưởng của các cặp từ xuất hiện quá thường xuyên, hàm này thỏa 3 tính chất:
* Có giới hạn tại 0.
* Là hàm không giảm.
* Có giá trị nhỏ khi x rất lớn.