1. Chuẩn bị Corpus

- Bộ văn bản: “Tôi học. Tôi chơi”

-Từ điển (dictionary): “Tôi”, “học”, “chơi”

2. Xây dựng ma trận đồng xuất hiện (Co-occurrence Matrix)

* Cửa sổ ngữ cảnh (window size) bằng 1 (chỉ xét các từ liền kề).
* Đếm số lần từ  xuất hiện trước từ :

|  |  |  |  |
| --- | --- | --- | --- |
|  | Tôi | Học | Chơi |
| Tôi | 0 | 1 | 1 |
| Học | 0 | 0 | 0 |
| Chơi | 0 | 0 | 0 |

Giải thích:

* “Tôi” xuất hiện cạnh (phí trước) “học” 1 lần
* “Tôi” xuất hiện cạnh (phía trước) “chơi” 1 lần

Ma trận X:

Giả sử ta tính toán chi tiết với đầu vào cụ thể là vector .

Vì vector đầu vào có 3 chiều, chúng ta sẽ thiết kế một Autoencoder có kiến trúc 3-2-3 (Input 3, Latent 2, Output 3).

1. Đầu vào:

Vector đầu vào cho bước tính toán này là:

2. Kiến trúc Autoencoder (3-2-3):

* Lớp đầu vào: 3 nơ-ron.
* Lớp tiềm ẩn: 2 nơ-ron.
* Lớp đầu ra: 3 nơ-ron.
* Hàm kích hoạt: Sigmoid cho cả lớp tiềm ẩn và lớp đầu ra.
* Hàm mất mát: MSE.

3. Tham số ban đầu (Trọng số và Độ lệch):

Chúng ta sẽ sử dụng các giá trị khởi tạo mới cho kiến trúc này:

* Bộ mã hóa (Encoder):
  + Ma trận trọng số Wenc​ (3x2): ​​
  + Vector độ lệch benc​ (1x2):
* Bộ giải mã (Decoder):
  + Ma trận trọng số Wdec​ (2x3):
  + Vector độ lệch bdec​ (1x3):
* Tốc độ học:

4. Hàm kích hoạt Sigmoid và đạo hàm:

5. Tính toán chi tiết cho vector đầu vào x=(0​1​1​):

a) Forward Pass:

* Bước 1: Tính toán đầu ra tuyến tính của Encoder

Đầu vào:

* Bước 2: Tính toán đầu ra của Encoder

Áp dụng hàm Sigmoid cho zlinear​:

Đây là vector biểu diễn nén 2 chiều.

* Bước 3: Tính toán đầu ra tuyến tính của Decoder

Đầu vào cho Decoder là :

* Bước 4: Tính toán đầu ra được tái tạo ()

Áp dụng hàm Sigmoid cho :

Đây là vector được tái tạo.

b) Tính toán hàm mất mát (Loss):

So sánh đầu vào và đầu ra tái tạo

Hàm mất mát cho mẫu này.

c) Truyền ngược (Backward Pass) và Cập nhật tham số:

* Bước 1: Gradient của Loss theo đầu ra () ()
* Bước 2: Gradient của Loss theo đầu ra tuyến tính của Decoder (​):
* Bước 3: Gradient của Loss theo trọng số Decoder (​) ( là 2x1, ​ là 1x3. Kết quả 2x3)
* Bước 4: Gradient của Loss theo độ lệch Decoder (​)
* Bước 5: Gradient của Loss theo lớp tiềm ẩn (z): ​ (​ là 1x3, ​ là 3x2. Kết quả 1x2)
* Bước 6: Gradient của Loss theo đầu ra tuyến tính của Encoder (​)): (​ là 1x2, là 1x2. Kết quả 1x2)
* Bước 7: Gradient của Loss theo trọng số Encoder (​) ( là 3x1, là 1x2. Kết quả 3x2)
* Bước 8: Gradient của Loss theo độ lệch Encoder (​)
* Bước 9: Cập nhật Tham số

Sử dụng tốc độ học :

* + Cập nhật ​:
  + Cập nhật ​:
  + Cập nhật ​:
  + Cập nhật bdec​:

Quá trình này là một bước huấn luyện cho vector đầu vào (0​1​1​). Để huấn luyện Autoencoder, bạn sẽ lặp lại quá trình Forward Pass, tính Loss, Backward Pass và cập nhật tham số cho từng mẫu dữ liệu trong tập dữ liệu của bạn (hoặc theo mini-batch) qua nhiều kỷ nguyên. Mục tiêu là giảm thiểu Loss trung bình trên toàn bộ tập dữ liệu.