

# Practice with List - Positional Embedding

*Hoàng-Nguyên Vũ*

## 1 Positional Embedding trong Transformer

Positional embedding là một phương pháp để mã hóa thông tin về vị trí của các từ trong câu. Transformer không có cấu trúc tuần tự, do đó cần có cách để cung cấp thông tin về vị trí của từ trong câu cho mô hình.

### 1.1 Công thức tính Positional Embedding

Với một từ tại vị trí  $pos$  trong câu và chiều của embedding vector là  $d_{\text{model}}$ , positional embedding được tính như sau:

$$PE_{(pos, 2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \quad (1)$$

$$PE_{(pos, 2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \quad (2)$$

Trong đó:

- $pos$  là vị trí của từ trong câu (bắt đầu từ 0).
- $i$  là chỉ số của chiều trong vector embedding.
- $d_{\text{model}}$  là độ dài của vector embedding.

### Giải thích Công thức

- Các giá trị  $PE_{(pos, 2i)}$  sử dụng hàm sin để mã hóa thông tin vị trí cho các chỉ số chẵn.
- Các giá trị  $PE_{(pos, 2i+1)}$  sử dụng hàm cos để mã hóa thông tin vị trí cho các chỉ số lẻ.
- Số mũ của 10000 chia cho  $d_{\text{model}}$  để điều chỉnh tần số của các giá trị sin và cos sao cho các từ ở các vị trí khác nhau có embedding khác biệt rõ rệt.

### Ưu điểm của Positional Embedding

- **Tính khả vi:** Hàm sin và cos là các hàm khả vi, giúp cho mô hình có thể học thông qua quá trình lan truyền ngược.
- **Mã hóa tuần tự:** Các giá trị sin và cos cho phép mô hình phân biệt được các vị trí khác nhau của từ trong câu, đảm bảo thông tin về thứ tự từ được duy trì.
- **Khả năng tổng quát hóa:** Các hàm tuần hoàn như sin và cos giúp mô hình tổng quát hóa tốt hơn cho các câu dài hơn hoặc ngắn hơn các câu trong tập huấn luyện.

## 2 Bài tập

Giả sử chúng ta có một câu với độ dài tối đa là 10 từ ( $max\_len = 10$ ) và vector embedding có độ dài là 16 ( $d_{model} = 16$ ). Hãy viết hàm tính Positional embedding cho câu này.

```
1 def create_position_matrix(seq_length, embed_size):
2     position_matrix = [[0 for _ in range(embed_size)] for _ in range(
3         seq_length)]
4     # Your Code Here #
5     return position_matrix
6
7 # Test
8 seq_length = 10
9 embed_size = 16
10 position_matrix = create_position_matrix(seq_length, embed_size)
11 print(position_matrix)
```

Kết quả:

```
1 Đầu ra:
2 [[ 0.00000000e+00  1.00000000e+00  0.00000000e+00  1.00000000e+00
3     0.00000000e+00  1.00000000e+00  0.00000000e+00  1.00000000e+00
4     0.00000000e+00  1.00000000e+00  0.00000000e+00  1.00000000e+00
5     0.00000000e+00  1.00000000e+00  0.00000000e+00  1.00000000e+00]
6 [ 8.41470985e-01  5.40302306e-01  3.10983593e-01  9.50415280e-01
7     9.98334166e-02  9.95004165e-01  3.16175064e-02  9.99500042e-01
8     9.99983333e-03  9.99950000e-01  3.16227239e-03  9.99995000e-01
9     9.99999833e-04  9.99999500e-01  3.16227761e-04  9.99999950e-01]
10 [ 9.09297427e-01 -4.16146837e-01  5.91127117e-01  8.06578410e-01
11     1.98669331e-01  9.80066578e-01  6.32033979e-02  9.98000667e-01
12     1.99986667e-02  9.99800007e-01  6.32451316e-03  9.99980000e-01
13     1.99999867e-03  9.99998000e-01  6.32455490e-04  9.99999800e-01]
14 [ 1.41120008e-01 -9.89992497e-01  8.12648897e-01  5.82753611e-01
15     2.95520207e-01  9.55336489e-01  9.47260913e-02  9.95503374e-01
16     2.99955002e-02  9.99550034e-01  9.48669068e-03  9.99955000e-01
17     2.99999550e-03  9.99995500e-01  9.48683156e-04  9.99999550e-01]
18 .....
19 .....
20 [ 9.89358247e-01 -1.45500034e-01  5.74317769e-01 -8.18632457e-01
21     7.17356091e-01  6.96706709e-01  2.50292358e-01  9.68170303e-01
22     7.99146940e-02  9.96801706e-01  2.52955229e-02  9.99680017e-01
23     7.99991467e-03  9.99968000e-01  2.52981943e-03  9.99996800e-01]
24 [ 4.12118485e-01 -9.11130262e-01  2.91259121e-01 -9.56644200e-01
25     7.83326910e-01  6.21609968e-01  2.80778353e-01  9.59772638e-01
26     8.98785492e-02  9.95952733e-01  2.84566569e-02  9.99595027e-01
27     8.99987850e-03  9.99959500e-01  2.84604605e-03  9.99995950e-01]]
```