Đại học Quốc gia TP.HCM
Trường Đại học Công nghệ Thông tin

Môn học: CS115 – Mathematics for Computer Science

CƠ CHẾ ATTENTION VÀ TRANSFORMER

(Phân tích Toán học và Thực nghiệm)

Sinh viên thực hiện:
Mai Quốc Anh – 24520002
Đặng Phú Duy – 24520010
Hà Bùi Trọng Nghĩa – 24520020

Giảng viên hướng dẫn:
Lương Ngọc Hoàng

Contents

1 Tổng quan

Mô hình Transformer giải quyết các hạn chế của RNN/LSTM trong việc xử lý chuỗi: khả năng tính toán song song và mô hình hóa phụ thuộc xa (long-range dependency). Cốt lõi của kiến trúc này là cơ chế Attention, cho phép xác định trọng số quan hệ giữa các cặp phần tử bất kỳ trong tập dữ liệu đầu vào thông qua các phép chiếu không gian vector.

2 Cơ sở lý thuyết và Động lực

Trước khi đi vào chi tiết mô hình, ta xem xét hai hạn chế toán học của các phương pháp tiền nhiệm:

1. **Embedding tĩnh:** e_{bank} là hằng số bất kể ngữ cảnh ("river bank" vs "bank account"), dẫn đến $\text{Sim}(w_i, w_j)$ không phản ánh đúng quan hệ ngữ nghĩa trong câu cụ thể.
2. **Hạn chế của Recurrence:** Trong RNN/LSTM, trạng thái ẩn $h_t = f(h_{t-1}, x_t)$ buộc quá trình tính toán phải tuần tự ($O(n)$ bước). Hơn nữa, tích của nhiều đạo hàm nhỏ hơn 1 qua thời gian dẫn đến $\lim_{k\to\infty} \prod_{i=1}^{k} \frac{\partial h_t}{\partial h_{t-i}} \to 0$ (Vanishing Gradient), làm mất thông tin ở các vị trí xa.

Cơ chế Attention giải quyết việc này bằng cách tính toán song song và thiết lập đường dẫn trực tiếp $O(1)$ giữa mọi cặp từ.

3 Biểu diễn Toán học của Attention

3.1 Mã hóa đầu vào (Input Embedding)

Để chuyển đổi từ không gian từ vựng rời rạc $\mathcal{V}$ sang không gian vector liên tục, mô hình sử dụng ma trận trọng số học được $W_E \in \mathbb{R}^{|\mathcal{V}| \times d_{model}}$.

Với mỗi token tại vị trí t được biểu diễn bởi vector one-hot $x_t \in \{0,1\}^{|\mathcal{V}|}$, vector embedding e_t tương ứng được tính bằng phép chiếu tuyến tính:

$$e_t = x_t W_E \tag{1}$$

Toàn bộ câu gồm n token sẽ tạo thành ma trận đầu vào $E \in \mathbb{R}^{n \times d_{model}}$:

$$E = [e_1; e_2; \ldots; e_n]^\top \tag{2}$$

Ma trận E này chứa các vector trù mật (dense vectors), đóng vai trò là dữ liệu đầu vào cho các khối Attention kế tiếp.

3.2 Không gian vector Query, Key, Value

Cho ma trận đầu vào $E \in \mathbb{R}^{n \times d_{model}}$. Mỗi token được chiếu tuyến tính sang ba không gian vector khác nhau thông qua các ma trận trọng số học được $W^Q, W^K \in \mathbb{R}^{d_{model} \times d_k}$ và $W^V \in \mathbb{R}^{d_{model} \times d_v}$:

$$Q = EW^Q, \quad K = EW^K, \quad V = EW^V \tag{3}$$

Việc tách biệt Q và K giúp phá vỡ tính đối xứng ($e_i^\top e_j = e_j^\top e_i$), cho phép mô hình học các mối quan hệ có hướng (directed relations).

3.3 Scaled Dot-Product Attention

Đầu ra của khối Attention là tổng có trọng số của các vector Value:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{4}$$

Cơ sở lý thuyết của hệ số tỷ lệ $\frac{1}{\sqrt{d_k}}$ Xét hai vector $q, k \in \mathbb{R}^{d_k}$ có các phần tử là biến ngẫu nhiên độc lập với trung bình 0 và phương sai 1. Tích vô hướng $z = q \cdot k = \sum_{i=1}^{d_k} q_i k_i$.

- Kỳ vọng: $\mathbb{E}[q_i k_i] = \mathbb{E}[q_i]\mathbb{E}[k_i] = 0$.
- Phương sai: $\text{Var}(q_i k_i) = 1$.

Theo tính chất cộng của phương sai các biến độc lập:

$$\text{Var}(z) = \text{Var}\left(\sum_{i=1}^{d_k} q_i k_i\right) = \sum_{i=1}^{d_k} \text{Var}(q_i k_i) = d_k \tag{5}$$

Độ lệch chuẩn là $\sqrt{d_k}$. Khi d_k lớn (ví dụ 512), tích vô hướng có biên độ lớn, đẩy hàm Softmax vào vùng bão hòa (gradient $\to 0$). Việc chia cho $\sqrt{d_k}$ giúp chuẩn hóa phương sai về 1 ($\text{Var}(z/\sqrt{d_k}) = 1$), ổn định quá trình lan truyền ngược.

3.4 Multi-Head Attention

Để mô hình học được nhiều không gian biểu diễn khác nhau (representation subspaces), ta thực hiện h phép Attention song song:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{6}$$

trong đó mỗi head được tính độc lập:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Với $W_i^Q, W_i^K \in \mathbb{R}^{d_{model} \times d_k}, W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ và $W^O \in \mathbb{R}^{hd_v \times d_{model}}$.

4 Mã hóa vị trí (Positional Encoding)

Do Self-Attention có tính chất bất biến hoán vị (permutation invariant), thông tin thứ tự được đưa vào thông qua phép cộng: $x_{pos} = e_{pos} + p_{pos}$.

4.1 Sinusoidal Encoding

Transformer sử dụng các hàm lượng giác với tần số khác nhau:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{7}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{8}$$

4.2 Tính chất dịch chuyển tuyến tính

Với mọi độ lệch cố định k, vector vị trí PE_{pos+k} có thể được biểu diễn như một hàm tuyến tính của PE_{pos}. Xét cặp kích thước $(2i, 2i+1)$ tương ứng với tần số ω_i:

$$\begin{bmatrix} \sin(\omega_i(pos+k)) \\ \cos(\omega_i(pos+k)) \end{bmatrix} = \begin{bmatrix} \cos(\omega_i k) & \sin(\omega_i k) \\ -\sin(\omega_i k) & \cos(\omega_i k) \end{bmatrix} \begin{bmatrix} \sin(\omega_i pos) \\ \cos(\omega_i pos) \end{bmatrix} \tag{9}$$

Đây là phép quay vector (Rotation Matrix). Tính chất này cho phép mô hình học được vị trí tương đối (relative position) dễ dàng hơn so với các phương pháp embedding vị trí học được.

5 Các thành phần kiến trúc khác

5.1 Position-wise Feed-Forward Networks (FFN)

Mỗi vị trí được xử lý độc lập qua hai lớp tuyến tính và hàm kích hoạt ReLU:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{10}$$

Lớp này thực hiện việc trộn thông tin giữa các kênh đặc trưng (channel mixing).

5.2 Add & Norm

Mỗi lớp con (Attention, FFN) được bao bọc bởi kết nối tắt (residual) và chuẩn hóa lớp (LayerNorm):

$$\text{Output} = \text{LayerNorm}(x + \text{Sublayer}(x)) \tag{11}$$

Trong đó LayerNorm chuẩn hóa theo chiều đặc trưng:

$$\hat{x} = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \cdot \gamma + \beta$$

6 Thực nghiệm: Disaster Tweets

Thực nghiệm so sánh khả năng biểu diễn ngữ cảnh giữa Embedding tĩnh (Mean Pooling) và BERT (Transformer Encoder) trên tập dữ liệu phân loại Disaster Tweets.

Kết quả định lượng:

Phương pháp	**F1 Score**
Simple Embedding (Static)	0.75237
BERT (Contextual)	**0.82500**

Phân tích: Mô hình BERT đạt độ chính xác cao hơn đáng kể nhờ cơ chế Attention, cho phép phân giải sự đa nghĩa của từ (ví dụ: "fire" trong ngữ cảnh thảm họa vs. "fire" trong ngữ cảnh sa thải) dựa trên các từ xung quanh, điều mà phép cộng vector đơn giản không thực hiện được.

7 Kết luận

Báo cáo đã trình bày hệ thống toán học của kiến trúc Transformer, trọng tâm là:

1. Cơ chế **Scaled Dot-Product Attention** với phép chuẩn hóa phương sai, giúp mô hình hội tụ ổn định.
2. **Multi-Head Attention** mở rộng khả năng biểu diễn trên nhiều không gian con.
3. **Positional Encoding** dạng hình sin cung cấp khả năng ngoại suy vị trí tương đối thông qua phép quay vector.

Các kết quả thực nghiệm khẳng định ưu thế vượt trội của việc mô hình hóa sự phụ thuộc ngữ cảnh so với các phương pháp truyền thống.