ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO TUẦN 5

Môn học: Project II

Chủ đề: Xây dựng mô hình dự đoán giá cổ phiếu bằng mạng Transformer

Giáo viên hướng dẫn:	**Đỗ Tuấn Anh**
Sinh viên thực hiện:	***Lê Minh Triết***
Mã số sinh viên:	**20220045**

Hà Nội - 2025

MỤC LỤC

Contents

1. Công việc đã hoàn thành trong tuần

1.1. Tìm hiểu về bối cảnh xuất hiện của mạng Transformer

Trước Transformer, các kiến trúc mạng nơ ron thường sử dụng cho các bài toán chuỗi là RNN, LSTM, Seq2Seq với Attention đều tồn tại các vấn đề về vanishing gradient và tốc độ xử lý. Transformer ra đời đã giải quyết được các vấn đề này.

Transformer là một kiến trúc mạng nơ-ron sâu được giới thiệu bởi Vaswani et al. trong bài báo *"Attention Is All You Need"* (2017). Đây là một bước đột phá lớn trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), giúp cải thiện hiệu suất dịch máy và nhiều tác vụ khác so với các mô hình trước đó như RNN và LSTM.

Điểm đặc trưng của Transformer là sử dụng cơ chế Attention, đặc biệt là Multi-Head Self-Attention, để xử lý dữ liệu đầu vào song song thay vì theo tuần tự như RNN/LSTM. Điều này giúp Transformer huấn luyện nhanh hơn, không bị mất thông tin dài hạn, và dễ mở rộng trên các GPU mạnh.

Mặc dù Transformer ban đầu được phát triển cho NLP, nhưng sau đó nó đã được mở rộng sang nhiều lĩnh vực khác như xử lý chuỗi thời gian, phân tích dữ liệu tài chính, y sinh, và thị giác máy tính (Vision Transformer - ViT).

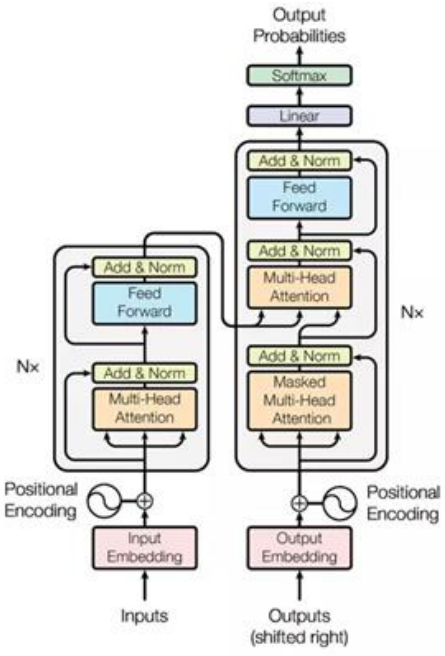
1.2. Tìm hiểu kiến trúc Transformer chuẩn

Transformer bao gồm 2 phần chính:

1. Encoder: Nhận đầu vào, xử lý qua nhiều lớp (layers) có **Self-Attention** và **Feedforward Network**, sau đó tạo ra một tập vector biểu diễn thông tin.

2. Decoder: Nhận thông tin từ encoder và sinh ra đầu ra (ví dụ: dự đoán từ tiếp theo trong dịch máy).

Cấu trúc của một lớp Encoder hoặc Decoder đều gồm 2 phần: Attention vàFeedforwarNetwork.



Self-Attention là cơ chế giúp mô hình học cách tập trung vào các phần quan trọng của dữ liệu đầu vào, thay vì xử lý theo tuần tự như RNN/LSTM. Từng thành phần trong chuỗi đầu vào sẽ quan sát lẫn nhau để thu thập ngữ cảnh của dữ liệu đầu vào.

Đầu vào sẽ được biến đổi bằng cách nhân các ma trận trọng số để tạo thành 3 vector

- Query (Q): Đại diện cho thông tin mà một thành phần đang "hỏi" phần tử khác.
- Key(K): Đại diện cho thông tin được lưu trữ để so sánh với query.
- Value(V): Chứa thông tin thực tế để tổng hợp đầu ra.

Từ đó công thức Self-Attention được tính bằng

$$\text{Attention}(Q,K,V)=\text{softmax}(Q \times K/\text{sqrt}(d\_k))V$$

Trong đó QxK là phép nhân 2 ma trận.

Multi-Head Attention là một thành phần chuẩn trong mạng Transformer ban đầu, nó chạy nhiều Self-Attention song song bằng cách chia nhỏ dữ liệu đầu vào thành các head, sau đó tổng hợp kết quả.

Positional Encoding để giúp Transformer nhận biết thứ tự của dữ liệu đầu vào, vì Transformer không có khả năng tự nhận diện thứ tự như RNN hay LSTM. Nó sử dụng hai hàm sin và cos để encode.

1.3. Phân tích sự phù hợp với bài toán dự đoán giá cổ phiếu

Kiến trúc Transformer ban đầu được thiết kế cho bài toán NLP, đặc biệt là Text Generation, nhưng khi áp dụng vào bài toán dự đoán giá cổ phiếu, có một số điểm không phù hợp:

1. **Tính tự tương quan ngắn hạn của giá cổ phiếu:**
    - Giá cổ phiếu thường chỉ chịu ảnh hưởng từ vài ngày gần nhất, thay vì toàn bộ lịch sử dữ liệu.
    - Việc sử dụng Multi-Head Attention trên toàn bộ dữ liệu có thể làm giảm độ chính xác, trong khi các mô hình tập trung vào tương quan ngắn hạn như LSTM thường hiệu quả hơn.
2. **Positional Encoding không phù hợp:**
    - Positional Encoding trong Transformer dựa trên các hàm tuần hoàn sin, cos, phù hợp với dữ liệu có tính chu kỳ rõ ràng.
    - Tuy nhiên, giá cổ phiếu có biến động mạnh, không tuần hoàn, nên cách mã hóa vị trí này có thể không mang lại lợi ích đáng kể.
3. **Transformer dễ bị quá khớp:**
    - Kiến trúc Transformer có nhiều tham số hơn so với LSTM, trong khi dữ liệu huấn luyện chỉ có khoảng 45,000 mẫu với 2 đặc trưng đầu vào.
    - Việc sử dụng Multi-Head Attention trên toàn bộ chuỗi có thể làm mô hình học quá mức (overfitting) thay vì tổng quát hóa tốt.
4. **Thiếu cơ chế lưu trạng thái như LSTM:**

- LSTM có khả năng ghi nhớ trạng thái, giúp tận dụng tốt hơn thông tin từ những ngày trước.
- Transformer không có trạng thái tái sử dụng, gây lãng phí tài nguyên và có thể bỏ sót thông tin quan trọng.
- Tuy nhiên, Transformer có lợi thế tính toán song song, giúp tăng tốc huấn luyện trên GPU.

5. **Decoder không cần thiết:**
   - Khi chỉ dự báo trung hạn (5 ngày kế tiếp), sử dụng Decoder là không cần thiết và có thể làm tăng lỗi cộng dồn (vì từng dự đoán ngày t+1sẽ ảnh hưởng đến dự đoán ngày t+2, t+3...).

Đây chỉ là các phân tích dựa trên lý thuyết, vì vậy tuần tới em sẽ thực nghiệm mô hình mạng Transformer chuẩn để kiểm chứng.

Từ những phân tích trên, các cải tiến chính được đề xuất:

1. **Chỉ sử dụng Encoder của Transformer**
   - Loại bỏ Decoder, chỉ dùng Transformer Encoder với Multi-Head Attention và Positional Encoding.
2. **Thay thế Positional Encoding**
   - Dùng Learnable Positional Encoding, giúp mô hình tự học cách mã hóa vị trí thay vì dựa vào các hàm tuần hoàn.
3. **Thay thế Multi-Head Attention bằng các Attention phù hợp hơn**
   - Thử nghiệm Local Attention (tập trung vào vùng dữ liệu gần nhất) và Causal Attention (đảm bảo không sử dụng dữ liệu tương lai).
4. **Kết hợp Transformer với LSTM**
   - Sử dụng LSTM để xử lý thông tin ngắn hạn, kết hợp với Transformer Encoder để khai thác mối quan hệ dài hạn và tận dụng song song hóa.
5. **Đánh giá hiệu suất các mô hình cải tiến**
   - Thử nghiệm và so sánh các mô hình theo các tiêu chí đã đề ra trong báo cáo tuần trước, chọn ra kiến trúc tối ưu nhất.

2. Dự kiến các công việc tuần tới

Xây dựng mạng Transformer (Encoder) theo kiến trúc chuẩn và so sánh để chứng minh các phân tích trên