**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**



**BÁO CÁO ĐỒ ÁN**

**MÔN CÁC MÔ HÌNH HỌC SÂU VÀ ỨNG DỤNG**

**Đề tài: All Attention You Need**

GVHD: TS. Nguyễn Vinh Tiệp

Nhóm sinh viên thực hiện:

1. Lê Thanh Dũng MSSV: 230101074

2. Nguyễn Lê Nam Anh MSSV: 230101070

3. Trần Quốc Huy MSSV: 230101048

4. Nguyễn Xuân Trường MSSV: 230104027

Tp. Hồ Chí Minh, 26/06/2024

**LỜI MỞ ĐẦU**

Sau đây, nhóm chúng em sẽ trình bày về Giới thiệu mô hình Transformer qua các chương sau:

* Chương 1: Giới Thiệu Mô Hình Transformer
* Chương 2: Kiến Trúc Của Transformer
* Chương 3: So Sánh Transformer ới RNN Và CNN
* Chương 4: Kết luận

# GIỚI THIỆU MÔ HÌNH TRANSFORMER

## Giới thiệu

Trong những năm gần đây, lĩnh vực trí tuệ nhân tạo (AI) và học máy (Machine Learning) đã chứng kiến sự phát triển vượt bậc với sự ra đời của nhiều mô hình học sâu (Deep Learning). Các mô hình như Recurrent Neural Networks (RNN) và Convolutional Neural Networks (CNN) đã từng thống trị trong các nhiệm vụ xử lý ngôn ngữ tự nhiên (NLP) và xử lý ảnh. Tuy nhiên, sự xuất hiện của Transformer đã mang đến một cách tiếp cận mới, mang lại hiệu suất vượt trội trong nhiều bài toán phức tạp.

## Định nghĩa

Transformer là một mô hình học sâu được thiết kế để phục vụ giải quyết nhiều bài toán trong xử lý ngôn ngữ và tiếng nói, ví dụ như bài toán dịch tự động, bài toán sinh ngôn ngữ, phân loại, nhận dạng thực thể, nhận dạng tiếng nói, chuyển văn bản thành tiếng nói. Tuy nhiên, khác với RNNs, Transformer không xử lý các phần tử trong một chuỗi một cách tuần tự. Nếu dữ liệu đầu vào là một câu ngôn ngữ tự nhiên, Transformer không cần phải xử lý phần đầu câu trước rồi mới tới phần cuối câu. Do tính năng này, Transformer có thể tận dụng khả năng tính toán song song của GPU và giảm thời gian xử lý đáng kể. Khác với các mô hình RNN và CNN, Transformer không dựa vào cấu trúc tuần tự để xử lý dữ liệu mà sử dụng cơ chế Attention để học và nắm bắt mối quan hệ giữa các từ trong câu.

# KIẾN TRÚC CỦA TRANSFORMER

Kiến trúc của Transformer bao gồm hai phần chính: Encoder (bộ mã hóa) và Decoder (bộ giải mã), mỗi phần bao gồm nhiều lớp (layers).

## Encoder

Encoder gồm nhiều lớp giống nhau, mỗi lớp bao gồm hai thành phần chính:

* **Multi-Head Self-Attention Mechanism:** Cho phép mô hình tập trung vào các phần khác nhau của câu đầu vào. Cơ chế này bao gồm nhiều đầu Attention (Multi-Head) để mô hình có thể chú ý đến nhiều phần khác nhau của dữ liệu.
* **Feed-Forward Neural Network:** Sau khi qua cơ chế Attention, dữ liệu được truyền qua một mạng nơ-ron truyền thẳng để thực hiện các phép biến đổi phi tuyến.

## Decoder

Decoder cũng gồm nhiều lớp giống nhau, với các thành phần tương tự như Encoder nhưng có thêm một lớp Attention:

* **Masked Multi-Head Self-Attention Mechanism:** Tương tự như trong Encoder, nhưng được điều chỉnh để đảm bảo mô hình không nhìn thấy các từ tương lai trong quá trình dự đoán.
* **Multi-Head Attention:** Tập trung vào các phần quan trọng của dữ liệu đầu vào từ Encoder.
* **Feed-Forward Neural Network:** Như trong Encoder, để biến đổi dữ liệu sau khi qua cơ chế Attention.
* **Masked Multi-Head Self-Attention Mechanism:** Tương tự như trong Encoder, nhưng được điều chỉnh để đảm bảo mô hình không nhìn thấy các từ tương lai trong quá trình dự đoán.
  1. **Cơ chế Attention**

Cơ chế Attention là điểm mấu chốt của Transformer, cho phép mô hình xử lý toàn bộ dữ liệu cùng một lúc thay vì tuần tự như trong RNN. Cơ chế này hoạt động như sau:

* **Scaled Dot-Product Attention:** Đầu tiên, tính toán sản phẩm dot giữa Query và Key, sau đó chia cho căn bậc hai của kích thước của Key (để tránh các giá trị quá lớn). Kết quả được đưa qua hàm Softmax để lấy trọng số Attention, sau đó nhân với Value để tạo ra đầu ra.
* **Multi-Head Attention:** Thay vì tính toán Attention một lần duy nhất, Multi-Head Attention thực hiện quá trình này nhiều lần song song với các trọng số khác nhau (head khác nhau), rồi kết hợp lại kết quả. Điều này giúp mô hình có thể tập trung vào các phần khác nhau của câu cùng một lúc.

# SO SÁNH TRANSFORMER VỚI RNN VÀ CNN

* 1. **Recurrent Neural Networks (RNN)**
* **Ưu điểm:** RNN thích hợp cho các bài toán liên quan đến dữ liệu tuần tự như dịch máy, dự đoán chuỗi thời gian và nhận dạng giọng nói nhờ khả năng ghi nhớ các trạng thái trước đó.
* **Nhược điểm:** RNN gặp vấn đề với các chuỗi dữ liệu dài do gradient biến mất (vanishing gradient) và gradient bùng nổ (exploding gradient). Việc tính toán tuần tự cũng làm giảm tốc độ huấn luyện.
  1. **Convolutional Neural Networks (CNN)**
* **Ưu điểm:** CNN nổi bật trong xử lý dữ liệu hình ảnh nhờ khả năng phát hiện các đặc trưng cục bộ qua các lớp convolution. CNN cũng được sử dụng trong một số bài toán NLP để phát hiện các mẫu đặc trưng.
* **Nhược điểm:** CNN không phù hợp lắm cho việc xử lý các chuỗi dữ liệu dài hoặc có mối quan hệ phức tạp giữa các phần tử không liền kề.
  1. **Transformer**
* **Ưu điểm:** Transformer xử lý các chuỗi dữ liệu dài một cách hiệu quả nhờ cơ chế Attention. Không cần xử lý tuần tự, Transformer có thể song song hóa quá trình tính toán, giúp tăng tốc độ huấn luyện. Mô hình này cũng nắm bắt được mối quan hệ dài hạn trong dữ liệu một cách xuất sắc.
* **Nhược điểm:** Yêu cầu tài nguyên tính toán và bộ nhớ lớn do số lượng tham số và độ phức tạp của cơ chế Attention.

# KẾT LUẬN

Transformer đã đem đến một bước tiến lớn trong lĩnh vực học sâu và xử lý ngôn ngữ tự nhiên. So với các mô hình trước đó như RNN và CNN, Transformer không chỉ cải thiện hiệu suất mà còn mở ra nhiều cơ hội mới cho các ứng dụng trí tuệ nhân tạo. Tuy nhiên, việc áp dụng Transformer cũng đòi hỏi tài nguyên tính toán lớn và cần phải được tối ưu hóa để phù hợp với từng bài toán cụ thể.

Tương lai của nghiên cứu và ứng dụng Transformer rất hứa hẹn, với khả năng tiếp tục cải thiện và mở rộng trong nhiều lĩnh vực khác nhau.

# TÀI LIỆU THAM KHẢO

1. <https://www.ibm.com/topics/transformer-model>
2. <https://baiblanc.github.io/2020/06/21/RNN-vs-CNN-vs-Transformer/>

# 