BÀI TẬP

Đồ án chuyên ngành

THỜI GIAN: 90 PHÚT

THÔNG TIN SINH VIÊN

MÃ SỐ SINH VIÊN: 3122411100

HỌ VÀ TÊN SINH VIÊN: LÂM QUANG KHÔI

LỚP MÔN HỌC: DCT122C5

QUY ĐỊNH

Sử dụng:

- Unicode
- Font: Times New Roman
- Font size: 13 hoặc 14

Đặt tên file: MSSV_HỌ-LÓT-TÊN_LỚP

Nộp bài:

- Moodle SGU: hoctructuyen.sgu.edu.vn

ĐỀ BÀI

Công bố trên hệ thống: hoctructuyen.sgu.edu.vn

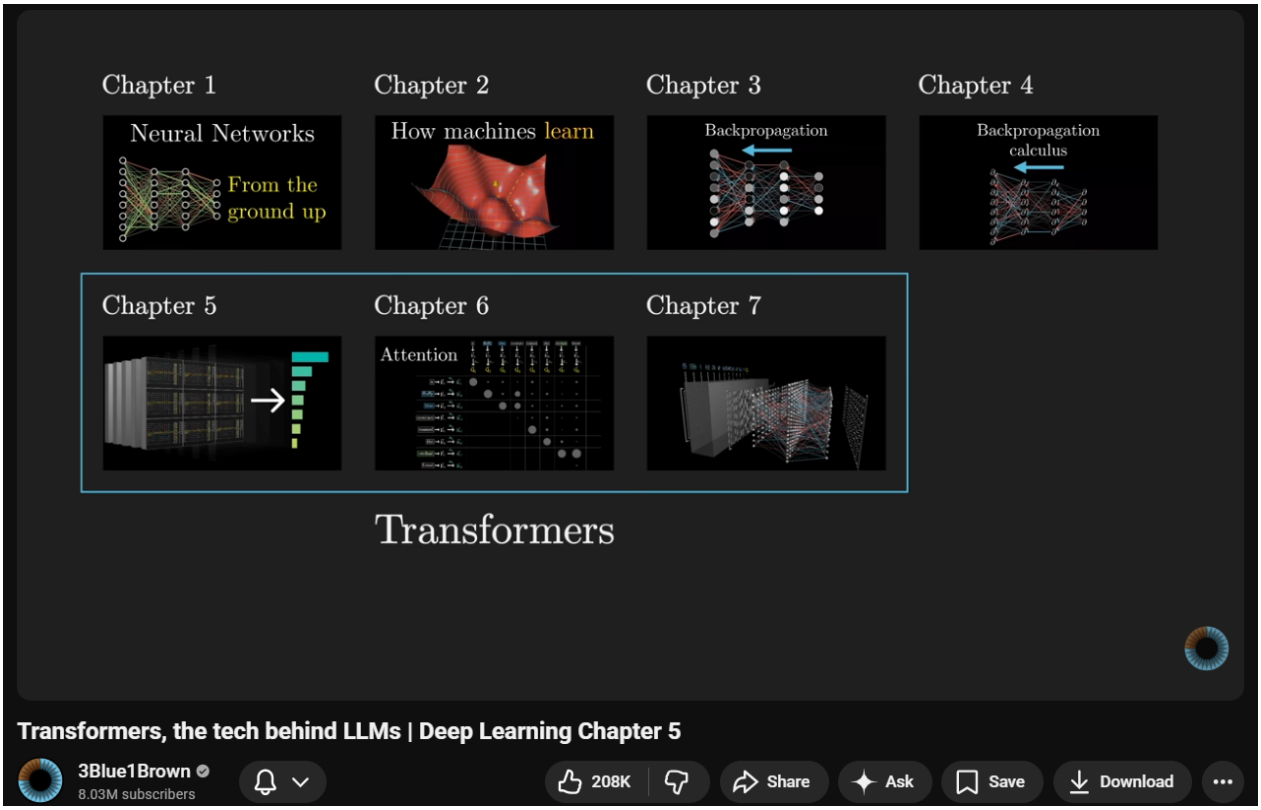
BÀI LÀM

Bài 1. Những chi tiết nào trong bài báo không được các tác giả chứng minh hoặc giải thích rõ nên làm anh/chị thấy cơ sở khoa học của chúng không vững? Anh/chị đã làm gì để hiểu được những nội dung đó và hiểu như thế nào? (5 điểm)

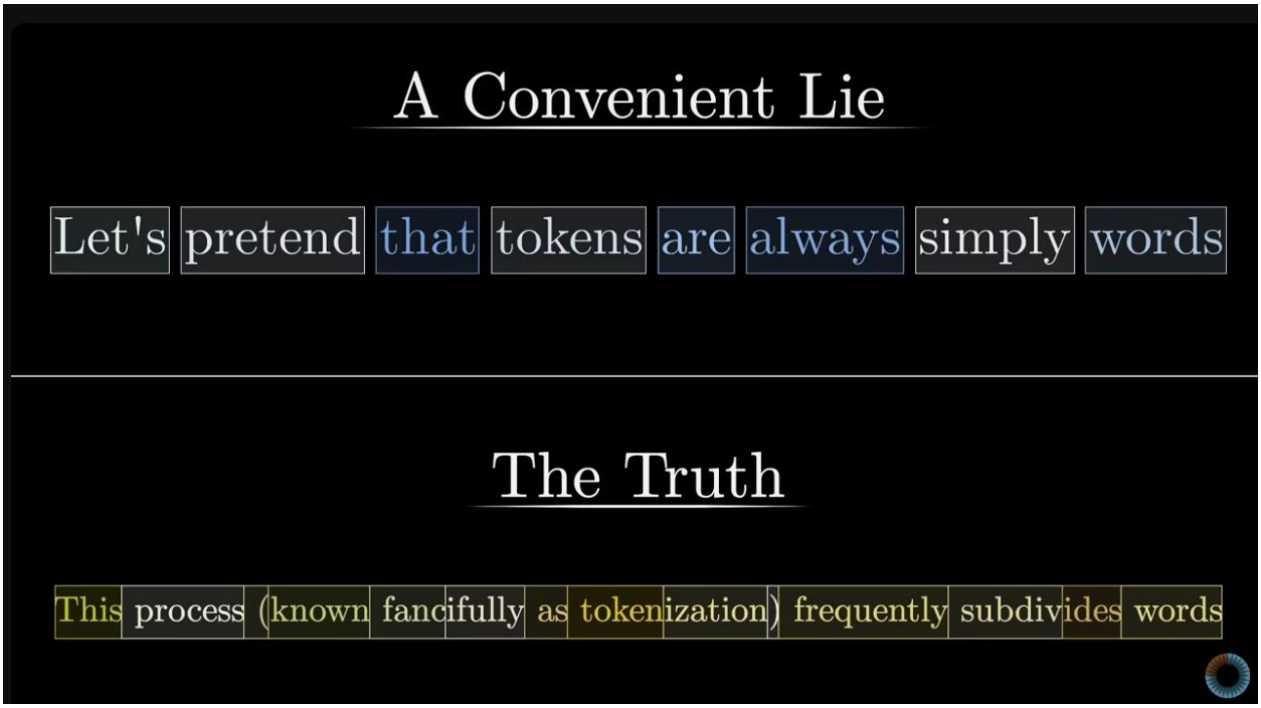
Đối với em bài báo mang tính gói trọn phần lớn các công trình nghiên cứu xử lý dữ liệu để trả về kết quả đầu ra theo mong muốn và mang tính học thuật rất nặng để có thể hiểu rõ được mục tiêu, quá trình và cả đóng góp của chuyên ngành trí thông minh nhân tạo sau này. Em hiện tại đến với bài nghiên cứu này với tâm thế chưa thực sự hiểu toàn bộ các khái niệm chi tiết về Attention cũng như các công nghệ trước để có thể nắm bắt được toàn bộ. Ngoài các khái niệm và em đã nắm được qua các mô học trước về xử lý ngôn ngữ tự nhiên em có thể hiểu được 1 chút về encoder, decoder, mô hình học máy, …. Tuy vậy các chi tiết mô tả về cơ chế Attention của bài báo ban đầu khiến em rất bối rối vì những khái niệm này chưa được giải thích quy trình rõ ràng, đặc biệt lấy đơn cử như việc tính ra giá trị của 1 head của 1 lớp Attention, nếu chỉ nhìn qua công thức:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Thời điểm này em vẫn chưa hình dung được Q (Query), K (Key) và V (Value) mang mục đích gì trong quá trình tính toán và bản thân attention này có quy trình và khả năng ra sao mà lại được đánh giá cao hơn các mô hình cấu trước tuần tự trước đó. Tất nhiên, để hiểu hơn 1 chút về các thuật ngữ này em đã có xem qua youtube các video hướng dẫn liên quan và có prompt Gemini để giải đáp 1 số thắc mắc của cá nhân trong quá trình học. Bài giải thích em chọn xem đó là của một youtube có tiếng chuyên nghiên cứu về chuyên ngành toán học 3Brown1Blue để xem anh ta giải thích 1 phần nào đó về mô hình Transformer dưới góc nhìn vào mô hình là một chương trình xử lý các siêu tham số để cho ra được kết quả theo huấn luyện:



Bài giảng khá đầy đủ và chi tiết đã cung cấp cho em được cái nhìn về việc tính toán 1 head của attention, Q, K, V hiểu cơ bản là gì và chúng được đưa vào quy trình gì để tính toán ra được ma trận vector kết quả cuối cùng. Ví dụ như việc tính head của 1 attention, anh ta giải thích một cách đơn giản hóa hơn 1 quy trình từ việc mỗi token thay vì phải nhúng vào từ một ma trận khác để có được ngữ cảnh của 1 từ thì anh ta coi 1 chữ là 1 token luôn để có thể giải thích một quy trình 1 cách đơn giản hơn. Từ đó em hiểu được quy trình và mục đích của công thức đề ra là về cơ bản tìm được ma trận được cập nhật mới từ Query bản thân token đó với các Key khác, từ đó lấy ngữ cảnh các của 1 token với toàn bộ các token khác và có được ngữ cảnh khác nhau, từ đó thay đổi giá trị mới theo toàn bộ các token để ra được nghĩa đúng. Ví dụ như từ “model” trong câu “an AI model” và “Fashion model” khi được đưa qua attention sẽ mang ý nghĩa số hóa khác nhau để mô hình hiểu được các ngữ nghĩa khác nhau của từ “model”.



Bài giảng giải thích khá chi tiết các khái niệm và thành phần cốt lõi của mô hình Transformer nhưng chưa đi qua hết toàn bộ mọi thứ, đơn cử như các vấn đề quá trình gì diễn ra trong khi training, quá trình add và norm là làm gì nhưng bài giảng đã giúp em hiểu được cốt lõi cơ chế attention đang được sử dụng như thế nào trong bài báo.

Còn vấn đề khác mà bài báo cũng chưa chỉ rõ ra mà em có thể nêu lên như bài báo giải quyết vấn đề overfitting như thế nào khi thời gian training lâu đến như vậy

Bài 2. Hãy trình bày cách xác định kích thước và tính các ma trận K, Q, V trong Self-Attention. (2điểm)

Nhằm xác định kích thước và tính các ma trận K, Q, V, trước hết phải hiểu rằng các ma trận này ảnh hưởng nhiều bởi các siêu tham số đầu vào như: chiều của 1 token sau khi nhúng (Embedding), số lượng head attention để tính đồng thời song song mà vẫn giữ được tốc độ xử lý,….

Đầu tiên, kích thước đầu vào cho Q và K có thể được tính bằng kích thước Embedding của mô hình, lấy ví dụ trong bài báo lấy d = 512, số lượng đầu attention tính song song là 8, nên ta có thể suy ra chiều của đầu vào Query, Key và Value là d_ k = d_v = d_model/8 = 512/8 = 64.

Như vậy ta có ma trận trọng số cho từng thành phần K, Q, V là:

- Ma trận trọng số cho Query (W_i^Q): Kích thước d_model x d_k → 512 x 64

- Ma trận trọng số cho Key (W_i^K): Kích thước d_model x d_k → 512 x 64
- Ma trận trọng số cho Value (W_i^V): Kích thước d_model x d_v → 512 x 64

Bài 3. Hãy cho một ví dụ cụ thể với dữ liệu giả lập và thực hiện các tính toán thủ công (không dùng công cụ và không viết chương trình) để mô phỏng quá trình tính toán Scaled Dot Product Attention. Chỉ dùng 1 head Attention. (3 điểm)

Em sẽ thực hiện bài tập đã có sẵn của thầy đã đăng trên group facebook:

Đề bài

Giả sử có câu: "Tôi thích đọc sách" gồm 4 từ. Mỗi từ được ánh xạ thành một vector có kích thước 3.

Cho các ma trận **Query (Q)**, **Key (K)**, và **Value (V)** cho mỗi từ như sau:

Từ	Query (Q)	Key (K)	Value (V)
A	[1, 0, 1]	[0, 1, 0]	[1, 0, 2]
B	[0, 1, 0]	[1, 0, 1]	[0, 2, 1]
C	[1, 1, 0]	[0, 0, 1]	[2, 1, 0]
D	[0, 0, 1]	[1, 1, 0]	[1, 2, 3]

Đầu tiên tính kết quả nhân ma trận Q với K của toàn bộ các từ khác:

Query A với các Key khác:

	Q.K^T
A.B	2
A.C	1
A.D	1

Query B với các Key khác:

	Q.K^T

B.A	1
B.C	0
B.D	1

Query C với các Key khác:

	Q.K^T
C.A	1
C.B	1
C.D	2

Query D với các Key khác:

	Q.K^T
D.A	0
D.B	1
D.C	1

Kết quả chia cho dot product:

A	B	C	D
1.154701	0.57735	0.57735	0
0.57735	0	0.57735	0.57735
0.57735	0.57735	1.154701	0.57735

Tính Softmax:

EXP(A)	EXP(B)	EXP(C)	EXP(D)
3.173073	1.781312	1.781312	1
1.781312	1	1.781312	1.781312
1.781312	1.781312	3.173073	1.781312

Kết quả softmax của A:

	EXP	Softmax (Weight)
	1	0.129271
	3.173073	0.410186
	1.781312	0.230272
	1.781312	0.230272
sum	7.735697	1

Cuối cùng nhân với V:

Kết quả Vector Z_A	V1	V2	V3
	0.820086	1.511187	1.359543

Reference:

[Attention in transformers, step-by-step | Deep Learning Chapter 6]