Personal Note

|  |  |
| --- | --- |
| <https://www.tensorflow.org/text/tutorials/transformer?authuser=0> | Material |
|  |  |
|  |  |

Review lại một số note từ week 6:

|  |  |  |  |
| --- | --- | --- | --- |
| Transformer | Encoder | nhiều lớp, mỗi lớp bao gồm một lớp self-attention và một lớp feedforward (học các features phức tạp từ input) |  |
| Decoder | tương tự như Encoder nhưng có thêm một lớp Masked Multi-Head Attention để kết nối với output của encoder |
| Thành phần chính | Self-Attention | => tập trung vào các phần khác nhau của đầu vào; nắm bắt các mối quan hệ dài hạn |
| Positional Encoding | => Transformer không có cấu trúc tuần tự như RNN, thêm positional encoding vào để cung cấp tt về vị trí của các token trong chuỗi |
| Feedforward Networks |  |

Dataset trong bài Material: TED Talks Open Translation Project, ngôn ngữ Portuguese-English từ tensorflow\_datasets (50000 train, 1100 validate và 2000 test)

Các step:

|  |  |  |
| --- | --- | --- |
| Test the Dataset | Input: (pt,en) Label: chuỗi (en) được dịch chuyển 1 vị trí (=> mỗi token đầu vào tiếng Anh sẽ khớp với nhãn là token tiếp theo)   * “Teacher forcing”: bất kể đầu ra của mô hình là gì, chuỗi tiếp theo sẽ luôn dùng giá trị đúng (ground truth) làm đầu vào cho bước tiếp theo * mô hình huấn luyện nhanh hơn vì không cần xử lý từng bước tuần tự, mà các đầu ra ở các vị trí khác nhau trong chuỗi có thể được tính toán song song | (trong bài material không có ví dụ, nên em lấy tạm trong code phần assignment) |
| Define the components | * seq-2-seq * encoder, decoder * self-attention |  |
| Embedding và Positional Encoding | * Dùng lớp tf.keras.layers.Embedding để chuyển đổi từ chuỗi token => vector * Sau khi vector, thêm positional encoding vào vector embedding. Positional encoding dùng hàm sin và cos ở các tần số khác nhau để mã hóa vị trí từ trong chuỗi. * đảm bảo các vị trí gần nhau có encoding tương đồng, giúp mô hình hiểu thứ tự từ trong câu * en\_emb.\_keras\_mask: | PositionalEmbedding:   * vocab\_size: kích thước từ vựng * d\_model: chiều sâu của vector embedding. * Lớp tf.keras.layers.Embedding dùng để tìm vector embedding của các token. * pos\_encoding là giá trị positional encoding được tạo ra trước đó. * Cộng vector embedding và positional encoding lại để có đầu ra hoàn chỉnh. * Dùng en\_emb.\_keras\_mask để mô hình bỏ qua các giá trị padding khi huấn luyện.   Padding là việc thêm các giá trị đặc biệt (thường là số 0) vào cuối (hoặc đầu) chuỗi để các chuỗi đều có độ dài như nhau. Ví dụ, nếu câu "how are you" được thêm padding để đạt chiều dài 6, ta có thể biểu diễn nó thành "how are you [0] [0] [0]".   * bỏ qua (mask) các vị trí chứa padding khi tính toán, đảm bảo mô hình chỉ học từ các giá trị thật trong chuỗi.   en\_emb.\_keras\_mask là một mask được tạo ra tự động để đánh dấu các vị trí có giá trị padding. Giá trị True trong mask cho biết token thật, còn False cho biết vị trí padding. |
| Add & Normalize (Residual Connections và Layer Normalization) | | |
| Base Attention Layer |  | |
| The cross attention layer |  | |
| The global self-attention layer |  | |
| The causal self-attention layer |  | |
| The feed forward network |  | |
| The encoder layer |  | |
| The encoder |  | |
| The decoder layer |  | |
| The decoder |  | |
| **The transformer** | | |
|  | Create the Transformer by extending tf.keras.Model | |
| Hyperparameters | num\_layers: số lượng các lớp encoder và decoder trong mô hình Transformer. | Trong bài báo gốc dung 6 num\_layers, có thể dung 4 lớp để nhẹ hơn |
|  | d\_model: kích thước của các vector embedding | 512 trong mô hình gốc, có thể giảm xuống 128 |
|  | dff: kích thước của lớp FeedForward trong Transformer. | mô hình gốc là 2048 (trong bài material dung 512) |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |