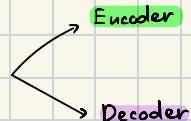


Attention Mechanism

- Cỗ chế' dịch máy -

I / Basic model

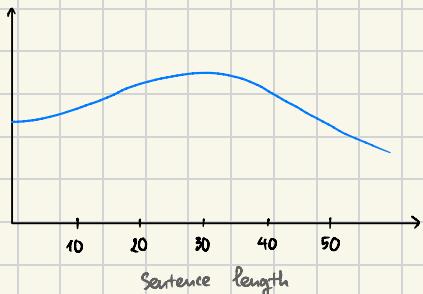
- Ta thường sử dụng mô hình "Seq2seq" với 2 khối RNN chính là 

Encoder: xử lý thông tin đầu vào và đầu ra là một khái vector biểu diễn duy nhất → quá trình nén thông tin.

Decoder: khôi encoder mang toàn bộ thông tin nén để decoder
tạo ra câu dịch (output)

II / Problem

* Khi độ dài của câu càng dài thì chất lượng của mô hình sẽ giảm



Why? → Đối với chuỗi dài thì RNN gấp 2 vẫn để

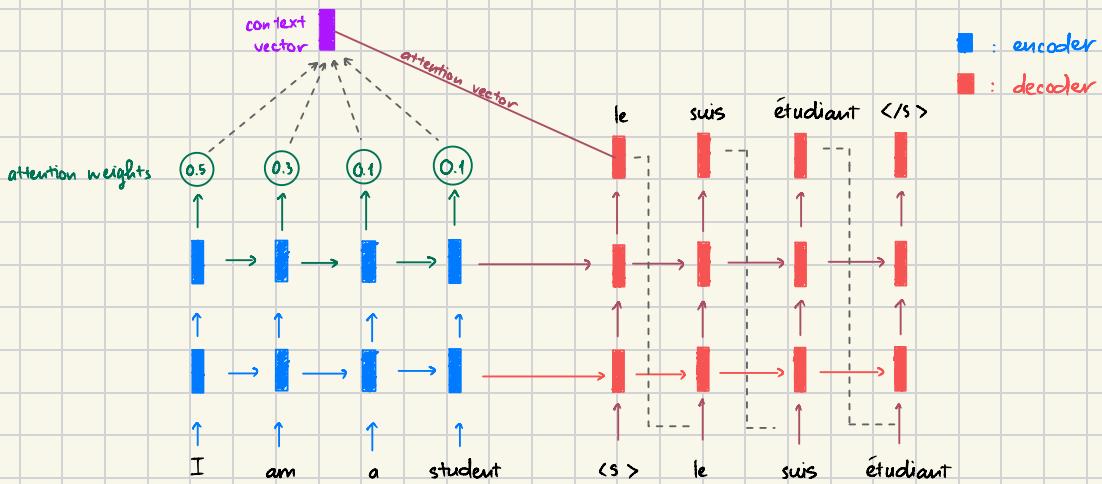
1. vanishing gradient (tiêu biến gradient)
2. exploding gradient (bùng nổ gradient)

* Việc sử dụng LSTM có những hạn chế:

- Khi huấn luyện, thời gian huấn luyện lâu do gradient path rất dài (chuỗi 100 từ có gradient như 100 layers)
- transfer learning không hoạt động với LSTM → với một bài toán mới thì ta cần huấn luyện lại mô hình với bộ dữ liệu riêng biệt cho nhu cầu đề ra (tốn kém)

III / Attention

- Cố chế' attention để ra đối để giải quyết các vấn đề của mô hình seq2seq (transformer và cố chế' attention ra đời để thay cho seq2seq → k° cùi đèn các mạng neuron hồi tiếp)



H: Mô hình seq2seq khi áp dụng cố chế' attention

Cách hoạt động của mô hình:

1. Encoder (bộ mã hóa):

- + Nhận đầu vào là chuỗi các từ ("I'm a student.")
- + Mỗi từ được biến thành một vector bằng phương pháp Embedding or One-hot encoding
- + Chuỗi vector này được đưa qua mạng RNN, GRU (Gate Recurrent Unit), or LSTM
- + Target là tạo biểu diễn (embedding) cho cả câu đầu vào.
- + Cuối cùng, output của bộ encoder (bộ mã hóa) là một chuỗi các vector biểu diễn trong thời của mỗi từ trong chuỗi đầu vào

2. Attention mechanism : (giải quyết vấn đề "bottleneck" của mô hình seq2seq khi toàn bộ thông tin phải dc nén vào một vector)

+ Khi bộ giải mã (decoder) tạo ra từ tiếp theo của câu dịch, thay vì chỉ đưa vào trạng thái cuối cùng của encoder, có chế' attention sẽ xem xét tất cả các trạng thái ẩn của encoder và tính toán trọng số' ẩn attention cho mỗi từ đầu vào dựa trên mức độ liên quan đến từ hiện tại đang dc dịch.

ATTENTION WEIGHT (rd: 0.5, 0.4, 0.1, ...)

→ chỉ ra sự đóng góp của từng từ trong chuỗi đầu vào

giải thích :

Khi bộ giải mã đang cò' gắng tạo ra từ "Je", nó có thể tập trung nhiều vào từ "I" (rdi trong số' 0.5), ít hơn vào từ "am" (0.3), và ít hơn nữa vào các từ còn lại.



một vector ngữ cảnh (context vector) được tạo ra bằng cách kết hợp tất cả các trạng thái ẩn của bộ mã hóa với attention weight tương ứng. Vector này sẽ dc dùng để giúp bộ giải mã tạo ra từ tiếp theo.

3 Decoder (Bộ giải mã)

- Bộ giải mã là một mạng RNN (hoặc GRU / LSTM) tương tự như bộ mã hóa.
- Sử dụng context vector và trạng thái ẩn trước đó để tạo ra từ tiếp theo trong chuỗi đầu ra.
- Quá trình này tiếp tục cho đến khi bộ giải mã sinh ra kí hiệu kết thúc câu </s>