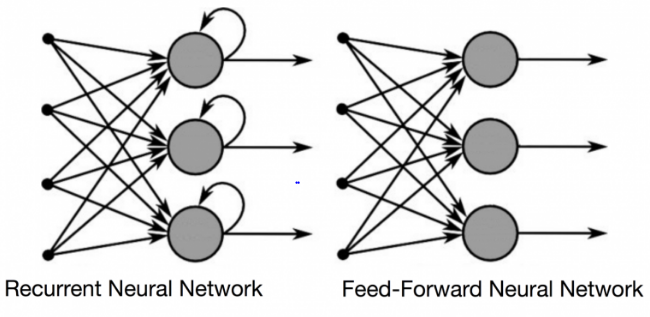
# Recurrent Neural Networks(RNN):

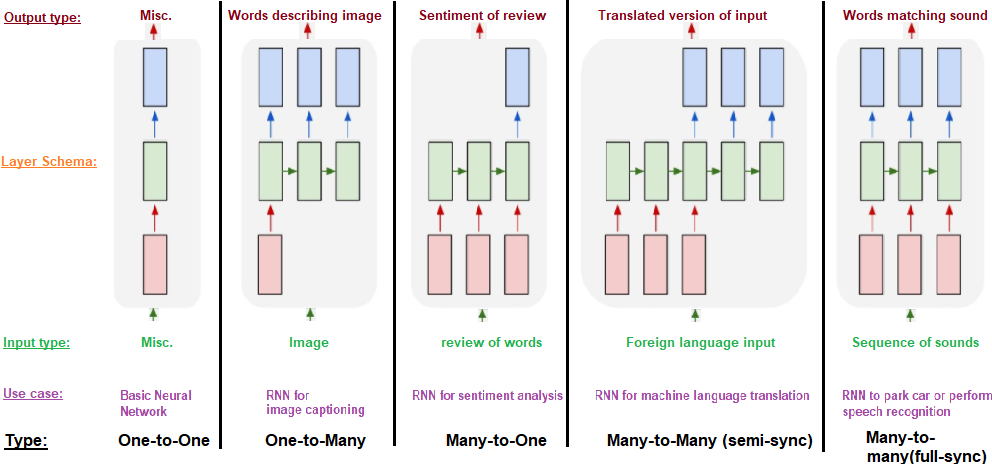
## Overview:

* RNN là một mô hình neural network có thể nhớ input của nó. Nhờ vào bộ nhớ nội, nó phù hợp với các bài toán học máy liên quan dữ liệu dạng chuỗi sequence.

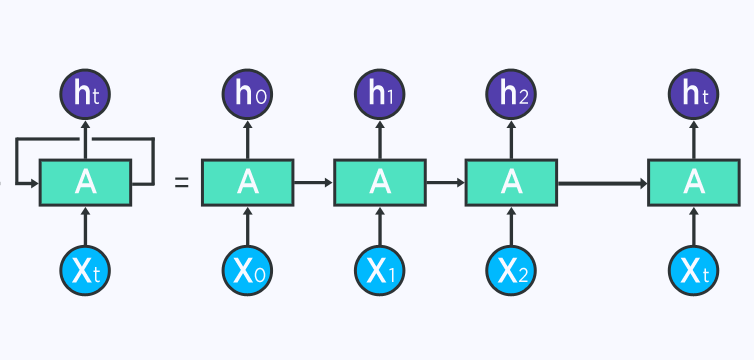
## Cách hoạt động:

* Khác với cách feed forward neural network hoạt động khi thông tin chỉ di chuyển theo 1 hướng (input layer->hidden layers->output layer). Thông tin trong RNN di chuyển qua một vòng lặp và có thể map input-output theo mô hình 1-1, 1-n, n-1, n-n.





* RNN có thể xem như là một chuỗi neural network



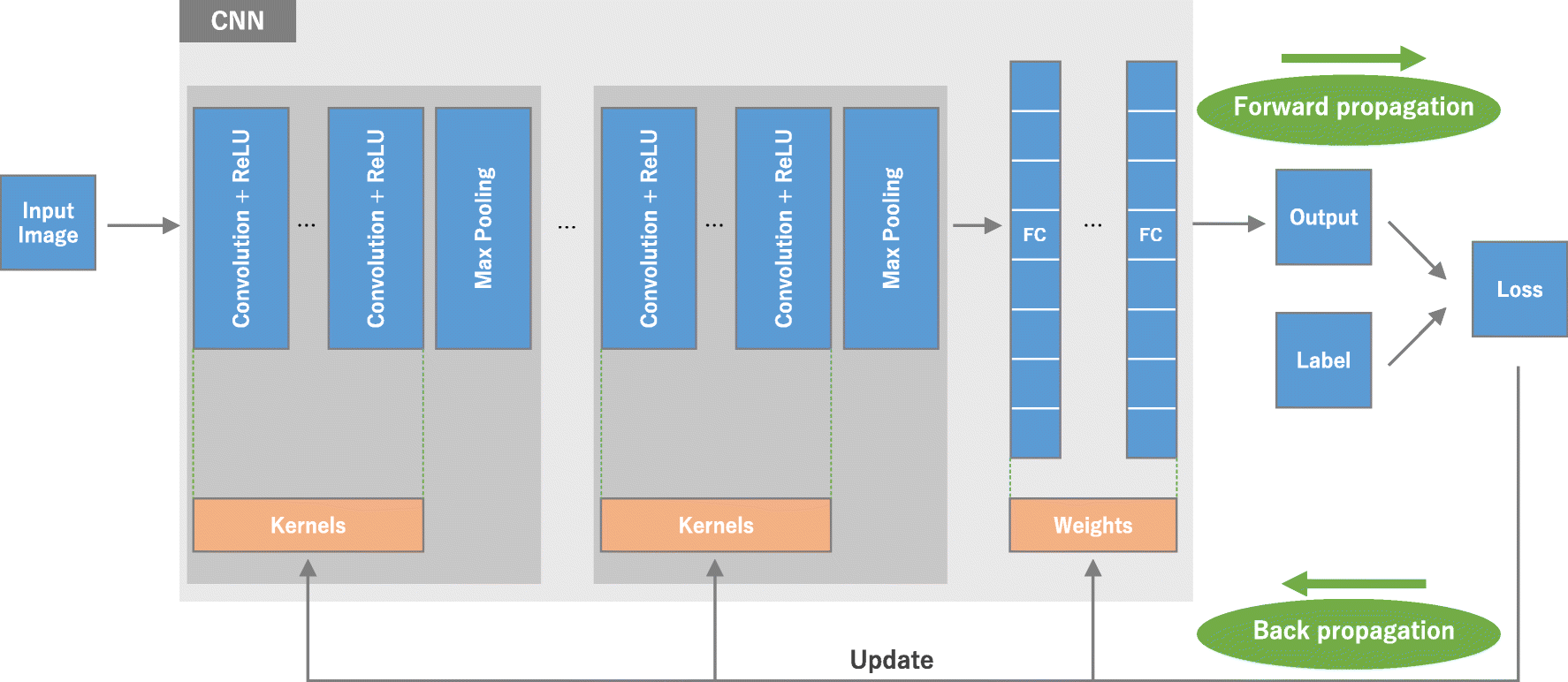
# Convolutional Neural Networks(CNN):

## Overview:

* CNN là một lớp Neural Network nổi tiếng trong việc giải quyết các bài toán thị giác máy tính. Được thiết kế để có thể tự động học phân cấp không gian các thuộc tính qua back propagation bằng cách sử dụng nhiều kỹ thuật như convolution layers, pooling layers và fully connected layers.

## Cách hoạt động:

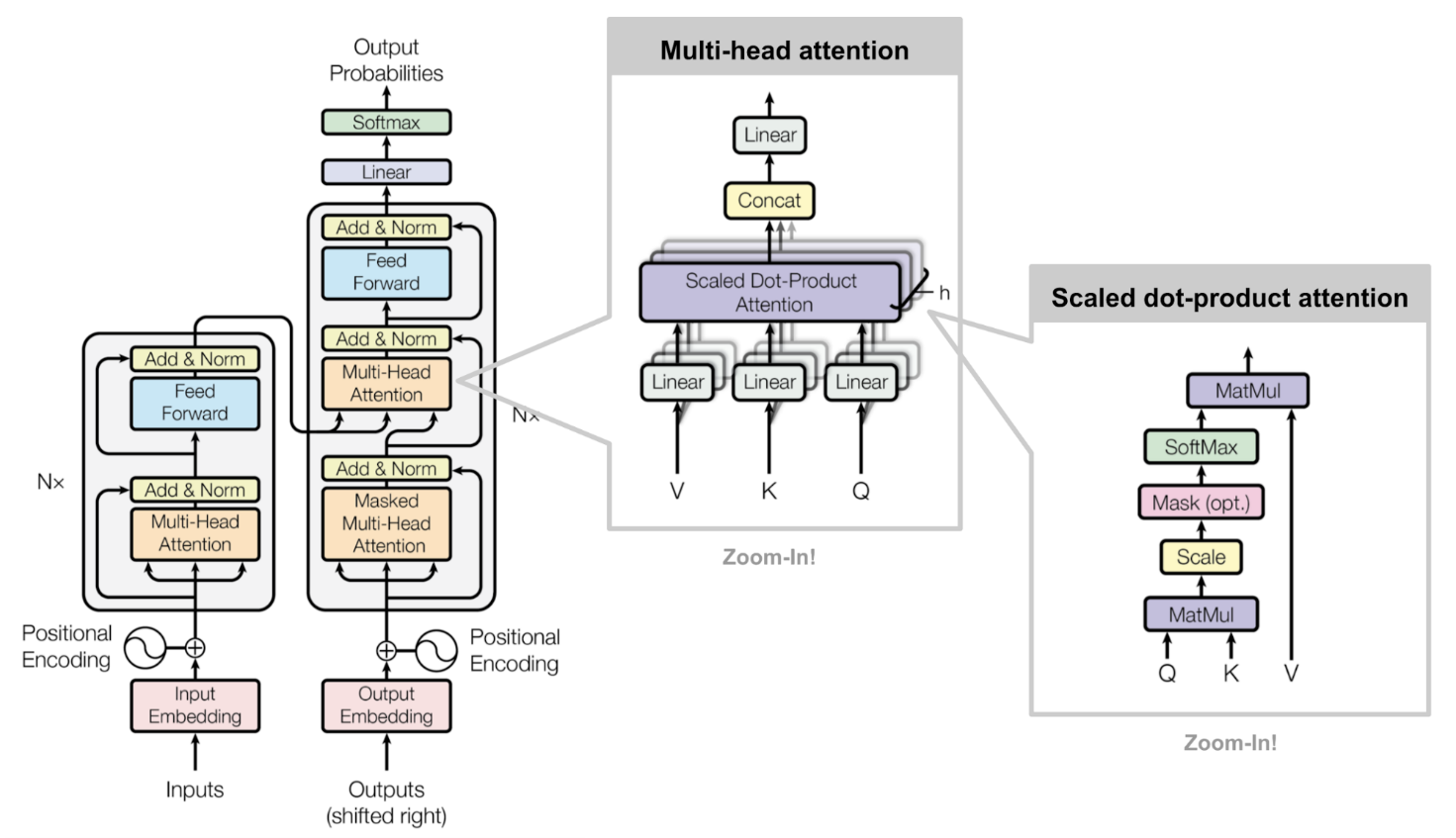
* CNN là 1 cấu trúc toán học bao gồm 3 lớp: convolution, pooling và fully connected. 2 lớp đầu (convolution layer và pooling layer) thực hiện chiết xuất thuộc tính còn lớp cuối cùng (fully connected layer) map từ thuộc tính được chiết xuất thành output.



# Transformer:

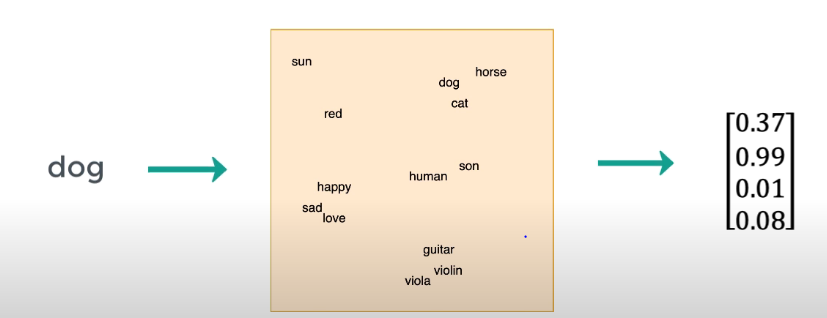
## Overview:

* Là một mô hình học sâu sử dụng cơ chế attention và có thể thực hiện tính toán song song, được ứng dụng chủ yếu trong các bài toán xử lý ngôn ngữ tự nhiên NLP.



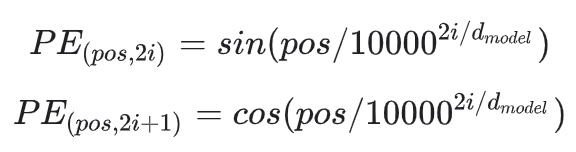
## Chi tiết kiến trúc:

* + 1. Kiến thức cần biết:
  + Embedding: một cách map từ một kiểu dữ liệu sang một vector gồm các số thực.

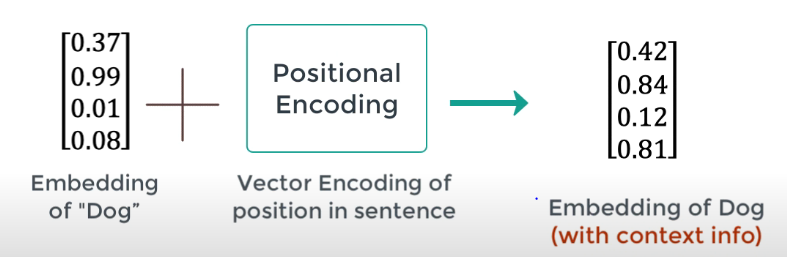


* + Positional Encoder: thuật toán tạo ra một vector mang giá trị ngữ cảnh dựa trên vị trí của từ trong câu.

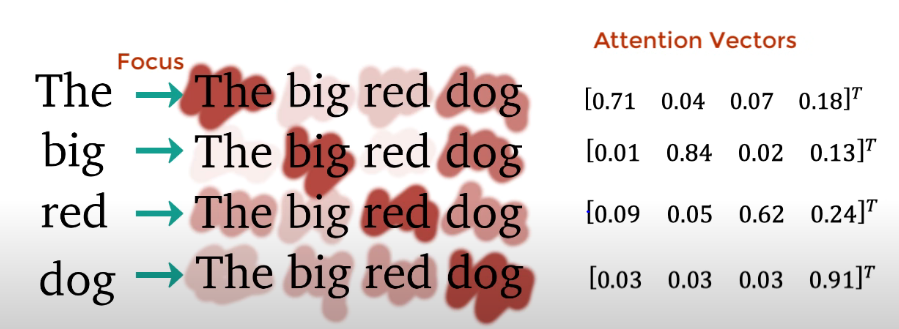




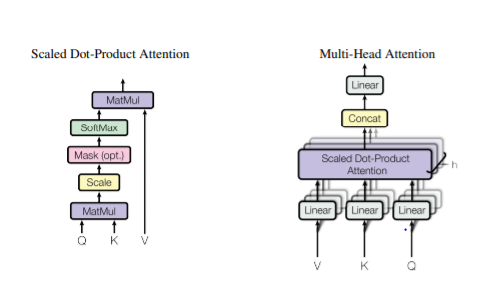
* + Khi kết hợp Embedding và Positional Encoder của một từ trong câu nào đó ta sẽ có kết quả Embedding của nó mang theo giá trị ngữ cảnh.



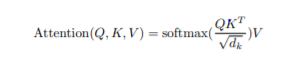
* + Attention:
    - Trả lời câu hỏi phần nào trong input ta nên tập trung vào (độ tương quan giữa các từ trong câu với nhau)



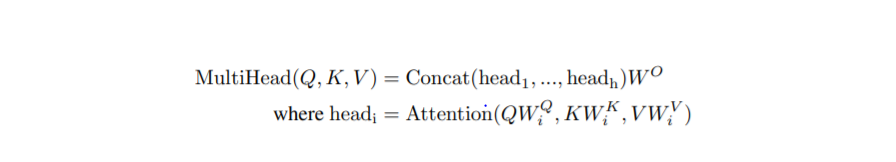
* + - Là một hàm map từ một *query*(Q) và một tập các cặp giá trị *key*(K) *- value*(V)thành một *output* (*query, key, value, output* tất cả đều là vector). *Output* là tổng trọng số của *value*, trọng số được tính bằng một hàm compatibility giữa *query* và *key* tương ứng (theo paper)
  + Scaled Dot-Product Attention và Multihead Attention:



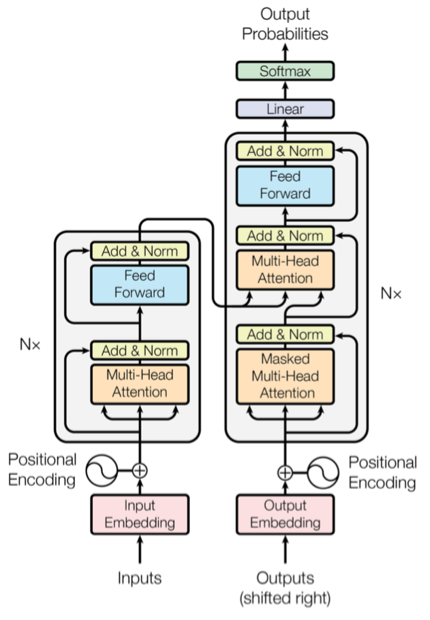
* + - Trong Scaled Dot-Product Attention, input bao gồm query, các keys và các value. Ta nhân query với tất cả các key và chia mỗi kết quả cho .



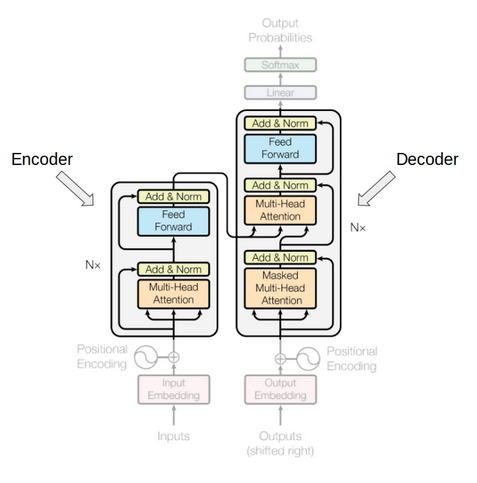
* + - Multihead Attention: thay vì truyền lần lượt từng giá trị query, key và value vào thì ta truyền đồng thời các giá trị và gộp chúng lại



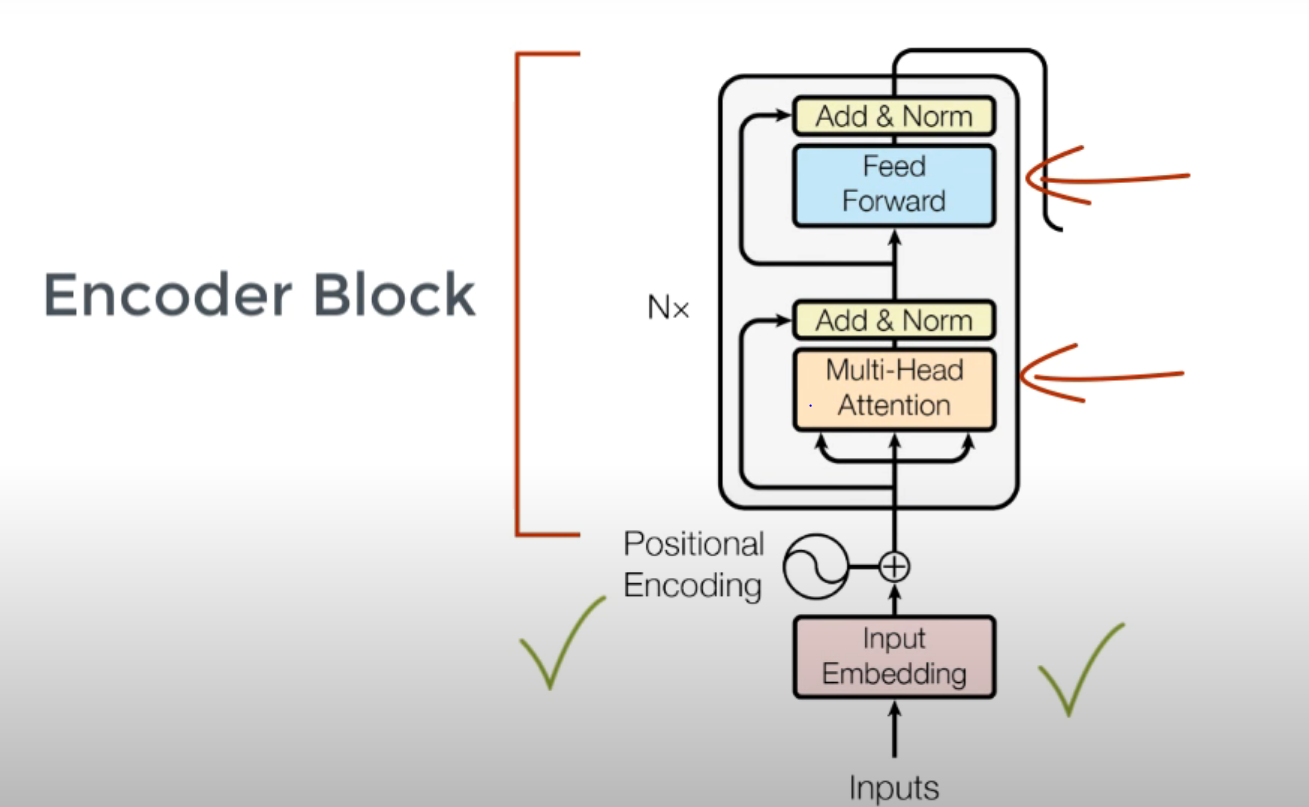
* + 1. Phân rã kiến trúc mô hình:



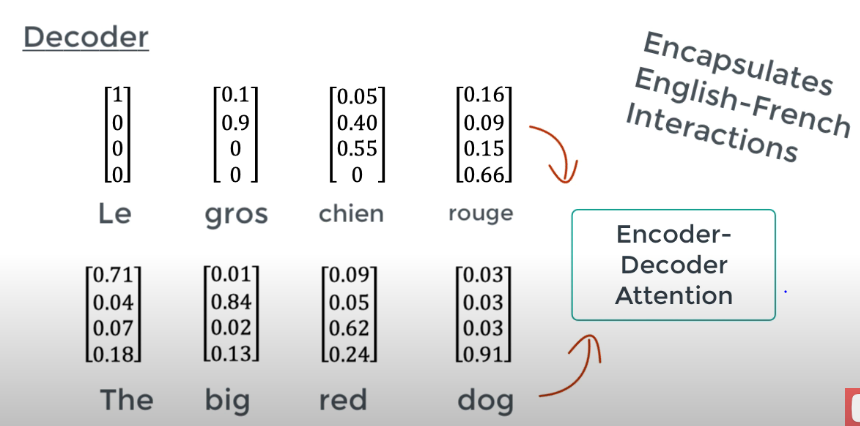
* + Khối Encoder chứa một lớp **Multi-Head Attention** sau đó kết quả được truyền vào một lớp **Feed Forward Neural Network**.
  + Khối Decoder them vào một lớp **Masked** **Multi-Head Attention** so với khối Encoder.

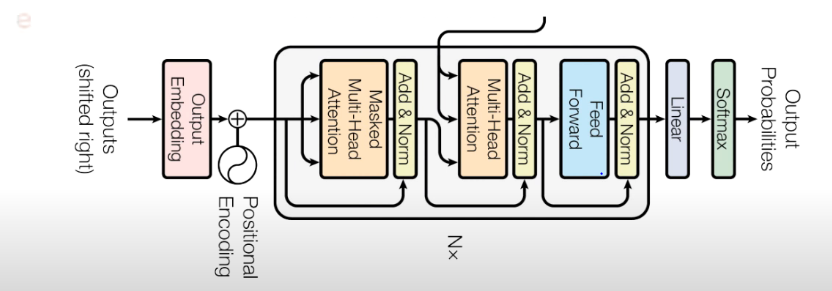


* + 1. Quy trình hoạt động trong trường hợp dịch ngôn ngữ từ Anh sang Pháp:
* Ở block Encoder: Input (các từ trong câu tiếng Anh) sau khi chạy qua Input Embedding (bao gồm Embedding kết hợp với Positional Encoder) cho ra kết quả là một tập các embedding của các từ trong input (mỗi từ là 1 input vào Input Embedding và sẽ cho ra 1 vector embedding ở output). Các vector này sẽ chạy qua một tầng Multi-head Attention Layer cho ra các attention vector. Các attention vector này rồi sẽ chạy qua một tầng Feed Forward Layer.



* Ở block Decoder: Input (các từ trong câu tiếng Pháp) truyền vào qua Output Embedding (bao gồm Embedding kết hợp với Positional Encoder) để thu được từ các từ trong câu dưới dạng embedding vector. Các vector này sẽ chạy qua một tầng Multi-head Attention Layer cho ra các attention vector. Các attention vector ở block decoder này sẽ cùng với các attention vector ở block encoder chạy qua một tầng Encoder\_Decoder attention sẽ nữa để cho ra các vector attention thể hiện sự tương quan giữa các từ tiếng Anh và tiếng Pháp trong câu. Kết quả từ Encoder\_Decoder attention chạy qua tầng Feedforward và Linear và ở Output sẽ dự đoán ra từ đúng tiếp theo cho tới khi gặp endtoken.





Tham khảo:

<https://builtin.com/data-science/recurrent-neural-networks-and-lstm>

<https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks>

<https://insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9#:~:text=CNN%20is%20a%20type%20of,%2D%20to%20high%2Dlevel%20patterns>.

<https://en.wikipedia.org/wiki/Transformer_(machine_learning_model)>

<https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html>

https://www.youtube.com/watch?v=TQQlZhbC5ps