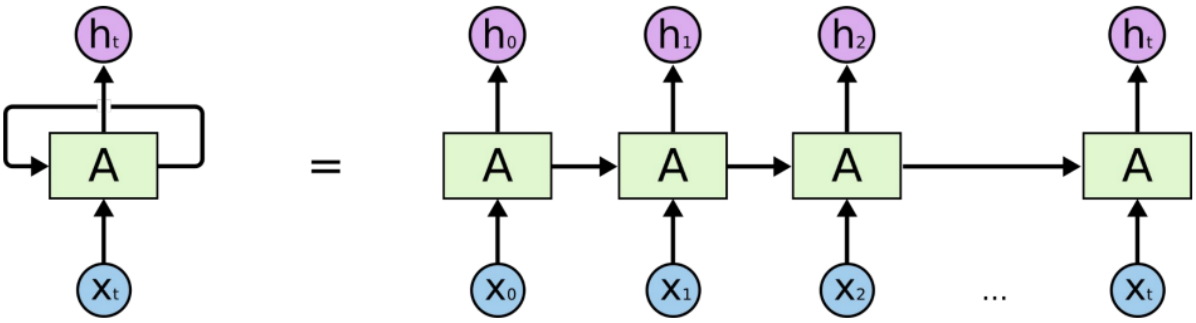
RNNs và LSTMs

1.RNNs

1.1. Recurrent Neural Network là gì?

Trong lý thuyết về ngôn ngữ, ngữ nghĩa của một câu được tạo thành từ mối liên kết của những từ trong câu theo một cấu trúc ngữ pháp. Nếu xét từng từ một đứng riêng lẻ ta không thể hiểu được nội dụng của toàn bộ câu, nhưng dựa trên những từ xung quanh ta có thể hiểu được trọn vẹn một câu nói. Như vậy cần phải có một kiến trúc đặc biệt hơn cho các mạng nơ ron biểu diễn ngôn ngữ nhằm mục đích liên kết các từ liền trước với các từ ở hiện tại để tạo ra mối liên hệ xâu chuỗi. Mạng nơ ron truy hồi đã được thiết kế đặc biệt để giải quyết yêu cầu này:



Vòng lặp ở thân mạng nơ ron là điểm mấu chốt trong nguyên lý hoạt động của mạng nơ ron truy hồi. Đây là chuỗi sao chép nhiều lần của cùng một kiến trúc nhằm cho phép các thành phần có thể kết nối liền mạch với nhau theo mô hình chuỗi.

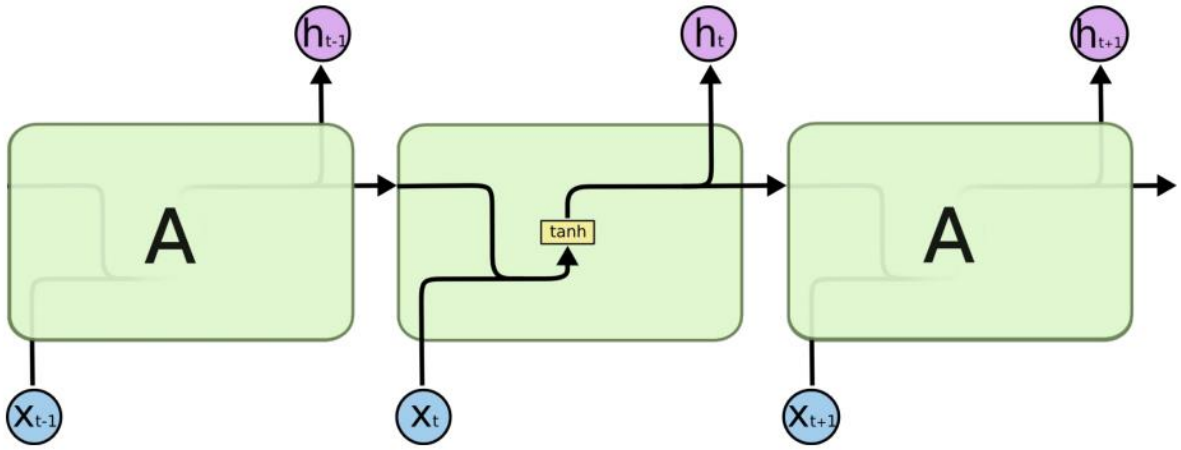
1.2.Cấu trúc của RNN tiêu chuẩn

Với chuỗi đầu vào x1,x2,...,xT, RNN duy trì một **hidden state** hth_tht tại mỗi thời điểm:

$$h_t = \tanh(W_h h_{t-1} + W_x x_t + b)$$

Và nếu bài toán có đầu ra tại mỗi bước:

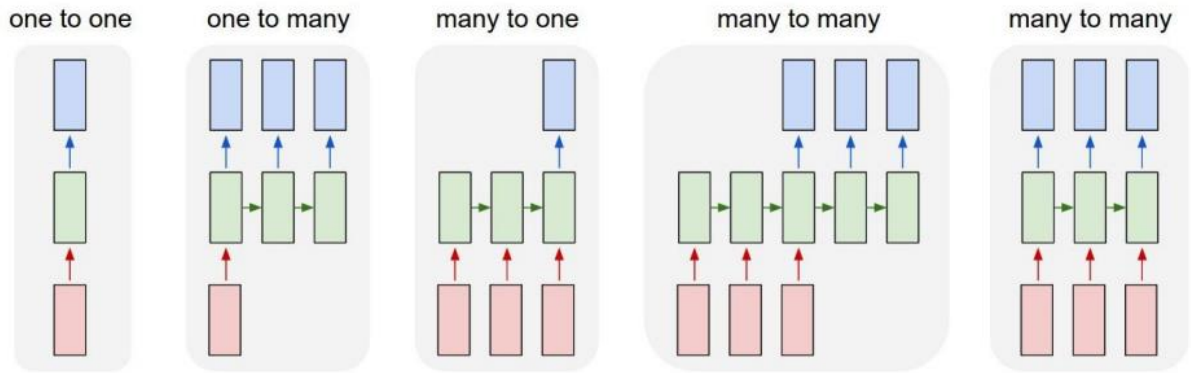
$$y_t = W_y h_t + c$$



RNN về bản chất giống như **một mạng nơ-ron lặp lại theo thời gian**, chia sẻ trọng số tại mỗi bước.

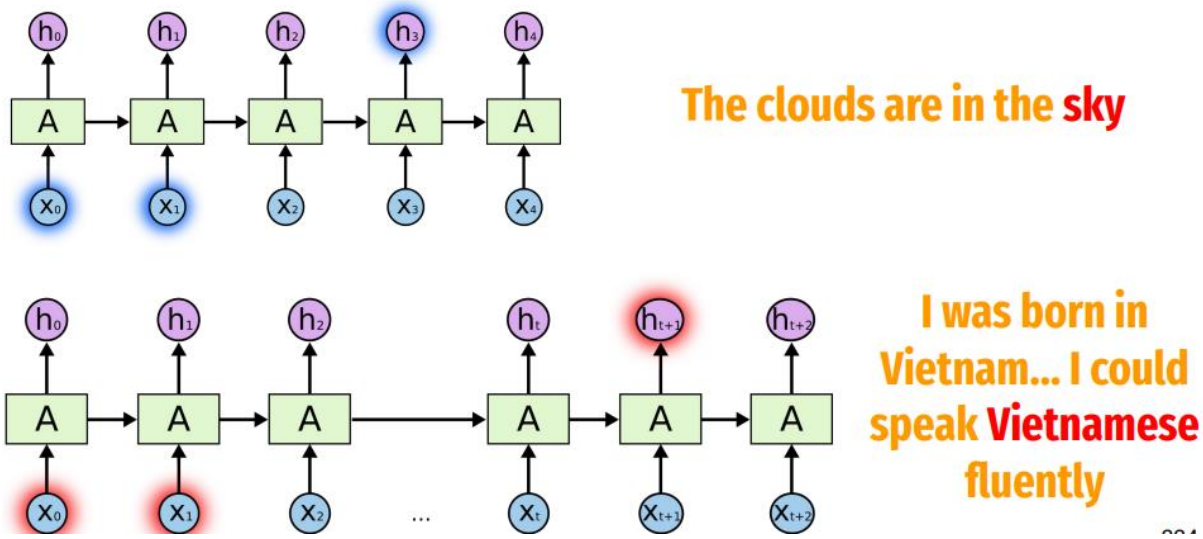
1.3.Phân loại bài toán RNN

One to one: mẫu bài toán cho Neural Network (NN) và Convolutional Neural Network (CNN), 1 input và 1 output, ví dụ với bài toán phân loại ảnh MNIST input là ảnh và output là ảnh đấy là số nào.
One to many: bài toán có 1 input nhưng nhiều output, ví dụ với bài toán caption cho ảnh, input là 1 ảnh nhưng output là nhiều chữ mô tả cho ảnh đấy, dưới dạng một câu.
Many to one: bài toán có nhiều input nhưng chỉ có 1 output, ví dụ bài toán phân loại hành động trong video, input là nhiều ảnh (frame) tách ra từ video, ouptut là hành động trong video
Many to many: bài toán có nhiều input và nhiều output, ví dụ bài toán dịch từ tiếng anh sang tiếng việt, input là 1 câu gồm nhiều chữ: "I love Vietnam" và output cũng là 1 câu gồm nhiều chữ "Tôi yêu Việt Nam". Để ý là độ dài sequence của input và output có thể khác nhau.



1.4.Vấn đề của RNN

Không thể ghi nhớ thông tin xa (Long-Term Dependency Problem)



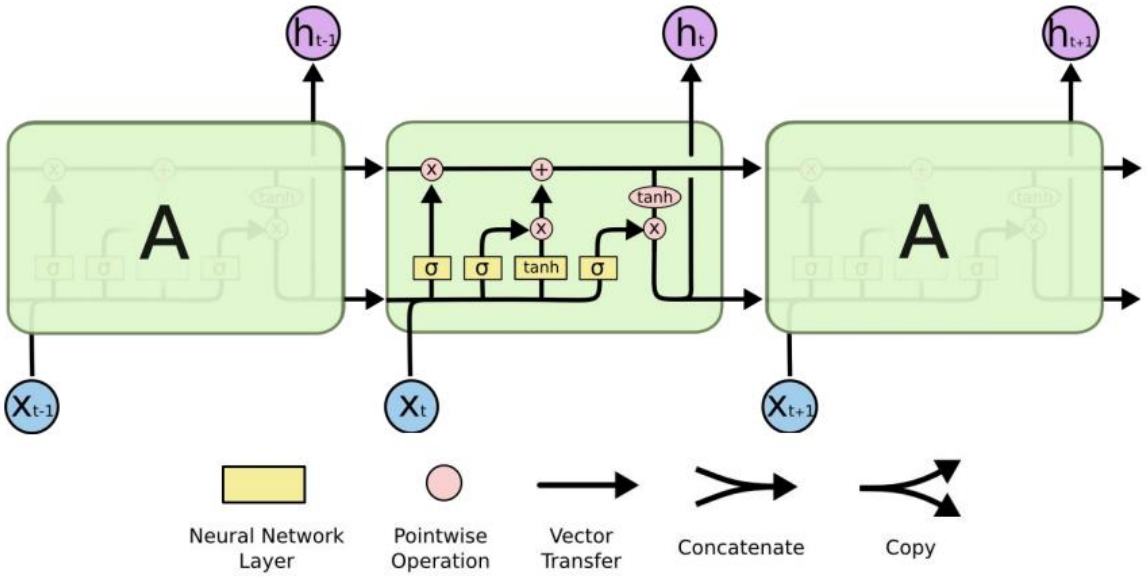
Đây là hậu quả của vanishing gradient:

Trong quá trình huấn luyện, tín hiệu lỗi (gradient) phải quay ngược từ bước t đến các bước trước đó.
Nhưng khi chuỗi quá dài, gradient bị nhỏ dần về 0 → không học được mối quan hệ xa.

2.Long Short Term Memory(LSTMs)

2.1.Tổng quát

LSTM được thiết kế để giải quyết vấn đề **mất trí nhớ dài hạn** của RNN thông qua một **cơ chế bộ nhớ có kiểm soát bằng các "cổng" (gates)**.

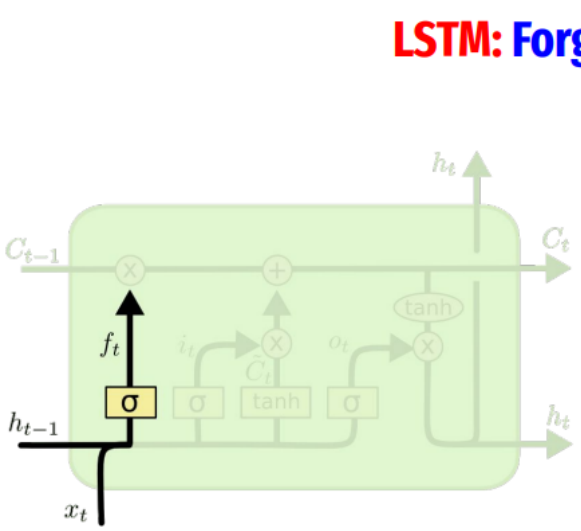


 LSTM = RNN + Bộ quản lý trí nhớ (Memory Controller)
Nó **tự quyết định cái gì cần nhớ, cái gì cần quên, và khi nào sử dụng**.

2.2 Cấu trúc của LSTMs

Thành phần	Chức năng
Forget Gate f_t	Quyết định *quên gì* từ quá khứ
Input Gate i_t	Quyết định *nhớ thêm gì mới*
Cell State C_t	Bộ nhớ dài hạn
Output Gate o_t	Xuất ra thông tin cho bước tiếp theo

2.2.1: Quyết định *quên gì* → Forget Gate

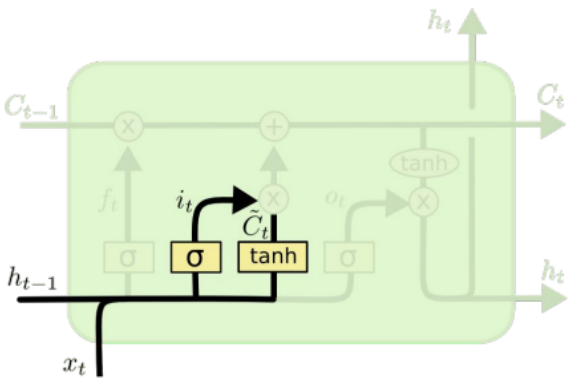


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

-Nếu ft≈0 → xoá bỏ ký ức cũ ở bước trước.

-Nếu ft≈1 → giữ lại hoàn toàn.

2.2.2: Quyết định *nhớ thêm gì* → Input Gate + Candidate



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

-it giống như **cửa cho phép thông tin mới đi vào**.

-C~t là **thông tin mới được đề xuất để nhớ**.

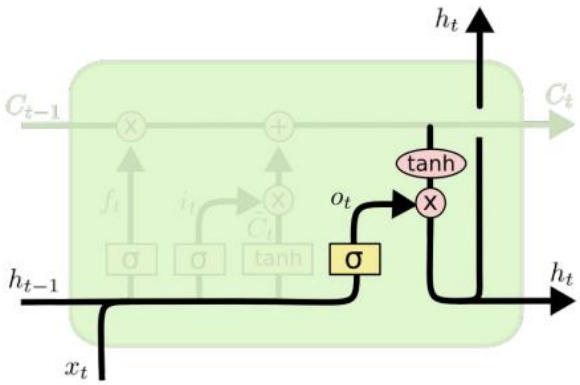
2.2.3: Cập nhật bộ nhớ mới

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Kết hợp giữa cái cũ (đã chọn lọc) + cái mới (đã chọn lọc)

2.2.4: Xuất thông tin → Output Gate

LSTM: Output

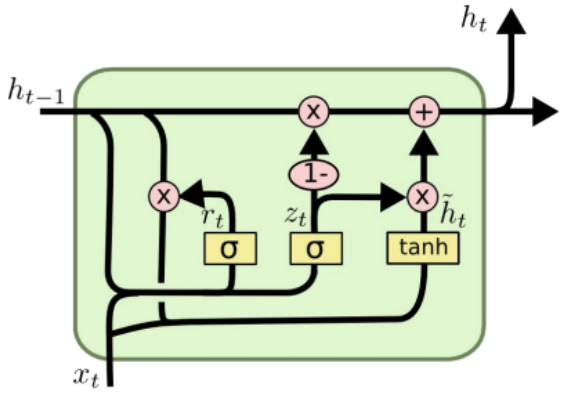


$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

 Chỉ cho **một phần bộ nhớ** được tiết lộ ra làm ngữ cảnh tiếp theo.

3. Gated Recurrent Unit(GRU)

GRU là một biến thể LSTM, nhưng đơn giản hơn (ít cổng hơn, ít tham số hơn), do đó train nhanh hơn nhưng vẫn đạt hiệu quả tương đương trong nhiều bài toán.



$$z_t = \sigma\left(W_z \cdot \left[h_{t-1}, x_t\right]\right)$$
$$r_t = \sigma\left(W_r \cdot \left[h_{t-1}, x_t\right]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot \left[r_t * h_{t-1}, x_t\right]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$