## RNN và LSTM

* A diagram of a network

  Description automatically generated

1. **Cấu trúc và cơ chế cổng**:

* **RNN**: Có cấu trúc đơn giản, không có cơ chế cổng. Mỗi bước thời gian chỉ dựa vào trạng thái ẩn từ bước trước đó.
* **LSTM**: Có cấu trúc phức tạp với các đơn vị nhớ và ba cơ chế cổng (forget, input, output), giúp kiểm soát luồng thông tin và duy trì gradient không đổi qua nhiều bước thời gian.

1. **Giải quyết vấn đề vanishing gradient**:

* **RNN**: Dễ gặp vấn đề vanishing gradient khi xử lý các chuỗi dữ liệu dài.
* **LSTM**: Được thiết kế để giải quyết vấn đề vanishing gradient, giúp mô hình học được các mối quan hệ dài hạn trong dữ liệu.

1. **Khả năng ghi nhớ dài hạn**:

* **RNN**: Thường gặp khó khăn trong việc ghi nhớ thông tin dài hạn do vấn đề vanishing gradient.
* **LSTM**: Có khả năng ghi nhớ thông tin dài hạn tốt hơn nhờ cấu trúc đơn vị nhớ và các cơ chế cổng.

## LSTM

Điểm chính tạo nên sự khác biệt chính là dòng đỏ dưới đây: là trạng thái bộ nhớ hoặc ô cho phép LSTM ghi nhớ thông tin rất cũ.

* A diagram of a diagram

  Description automatically generated

LSTM chứa các cổng có thể cho phép hoặc chặn thông tin đi qua.(Forget gate, input gate, output gate)

Các cổng bao gồm một lớp mạng nơ-ron sigmoid cùng với một phép nhân theo điểm.

Đầu ra của sigmoid dao động từ 0 đến 1:

* 0: Không cho phép bất kỳ dữ liệu nào đi qua
* 1: Cho phép mọi thứ đi qua

Forget gate:

* Quyết định thông tin nào cần bỏ đi khỏi trạng thái của ô nhớ (cell state).
* Nó lấy đầu vào hiện tại và đầu ra từ bước thời gian trước đó để xác định thông tin nào không còn liên quan và nên bị quên đi.
* Đầu ra là một số dao động từ 0 đến 1.

Công thức:

* + 
  + A diagram of a diagram of a diagram

    Description automatically generated

Input gate

* Quyết định thông tin mới nào sẽ được lưu trữ trong trạng thái của ô nhớ (cell state).
* Nó xem xét đầu vào hiện tại và đầu ra từ bước thời gian trước đó.
* Quyết định giá trị nào chúng ta sẽ cập nhật.
* **Lớp Tanh:**
  + Tạo một vector các giá trị ứng viên mới.
  + A close-up of a mathematical equation

    Description automatically generated
  + A diagram of a tank

    Description automatically generated

Update cell

* Quá trình là sự kết hợp của 2 quá trình trên là forget và input.
  + A black text on a white background

    Description automatically generated
* Quên những điều chúng ta quyết định quên trước đó:
  + A black symbols on a white background

    Description automatically generated
* Thêm thông tin chúng tôi quyết định thêm:
  + 
  + A diagram of a diagram of a flowchart

    Description automatically generated

Output gate:

* Lớp cổng đầu ra chịu trách nhiệm quyết định phần nào của trạng thái ô nhớ (cell state) sẽ được xuất ra. Nó thực hiện điều này bằng cách xem xét đầu vào hiện tại và đầu ra từ bước thời gian trước đó, sau đó áp dụng một hàm sigmoid để quyết định mức độ của từng phần thông tin sẽ được phép đi qua. Đầu ra của hàm sigmoid sẽ là một giá trị từ 0 đến 1, biểu thị mức độ thông tin nào nên được giữ lại và truyền đi.

**Lớp Tanh:**

* Lớp Tanh sau đó được áp dụng để đẩy giá trị của trạng thái ô nhớ (cell state) vào khoảng từ -1 đến 1. Điều này giúp chuẩn hóa giá trị và đưa chúng vào một dải giá trị có thể xử lý tốt hơn bởi các lớp tiếp theo trong mạng nơ-ron. Lớp Tanh này tạo ra một vector các giá trị mới, các giá trị này sẽ được kết hợp với kết quả từ lớp cổng đầu ra để tạo ra đầu ra cuối cùng của mạng LSTM cho bước thời gian hiện tại.

Nhờ sự kết hợp của các lớp cổng đầu vào, cổng quên, và cổng đầu ra cùng với lớp Tanh, LSTM có khả năng lưu trữ và xử lý thông tin một cách linh hoạt và hiệu quả, giúp nó vượt qua các hạn chế của các mạng nơ-ron truyền thống trong việc xử lý các chuỗi dữ liệu dài.

A black text on a white background

Description automatically generated

A diagram of a flowchart

Description automatically generated

## BiLSTM

* Bidirectional LSTM là một biến thể của LSTM trong đó có hai lớp LSTM chạy theo hai hướng ngược nhau: một lớp theo chiều xuôi (forward) và một lớp theo chiều ngược (backward). Điều này giúp mô hình có khả năng sử dụng thông tin từ cả quá khứ và tương lai của chuỗi dữ liệu

A group of math equations

Description automatically generated

Một BiLSTM bao gồm hai lớp LSTM hoạt động song song:

1. **LSTM tiến (Forward LSTM)**: Lớp này xử lý chuỗi từ đầu đến cuối (từ trái sang phải).
2. **LSTM lùi (Backward LSTM)**: Lớp này xử lý chuỗi từ cuối đến đầu (từ phải sang trái).

Mỗi từ trong chuỗi đầu vào sẽ được xử lý bởi cả hai lớp LSTM này. Đầu ra của hai lớp LSTM sau đó được kết hợp lại (thường là bằng cách nối hoặc cộng) để tạo ra một vector đặc trưng tổng hợp cho mỗi từ trong chuỗi. Điều này giúp BiLSTM có thể hiểu rõ hơn về ngữ cảnh của từ dựa trên cả thông tin phía trước và phía sau.

A diagram of a diagram

Description automatically generated

A diagram of a software system

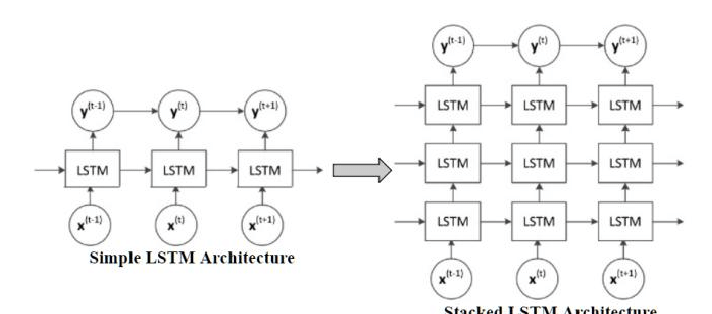
Description automatically generated with medium confidence

A diagram of a computer language

Description automatically generated with medium confidence

## Stack LSTM

* Stacked LSTM là một cấu trúc LSTM nhiều lớp, trong đó đầu ra của lớp LSTM dưới cùng sẽ là đầu vào của lớp LSTM tiếp theo. Việc này giúp mô hình có khả năng học được các biểu diễn dữ liệu ở nhiều mức độ trừu tượng khác nhau.
* A white background with black text

  Description automatically generated
* 

## Vấn đề Vanishing Gradient trong RNN

LSTM giải quyết vấn đề "vanishing gradient" (mất dần gradient) bằng cách cho phép gradient (đạo hàm) chảy ngược qua thời gian mà không bị mất dần hoặc bùng nổ.

Trong các mạng nơ-ron truyền thống, đặc biệt là mạng nơ-ron hồi tiếp (RNN), khi huấn luyện qua nhiều bước thời gian, gradient có thể trở nên cực kỳ nhỏ (vanishing gradient) hoặc cực kỳ lớn (exploding gradient). Điều này gây khó khăn cho việc điều chỉnh các trọng số của mạng, làm cho mạng khó học được các phụ thuộc dài hạn trong chuỗi dữ liệu.

LSTM sử dụng cấu trúc cổng đặc biệt (như cổng quên, cổng đầu vào, và cổng đầu ra) để kiểm soát dòng chảy của thông tin và gradient qua các bước thời gian. Các cổng này cho phép gradient duy trì độ lớn hợp lý khi lan truyền ngược qua thời gian, ngăn chặn việc gradient bị mất dần hoặc bùng nổ.

Nhờ vậy, mạng LSTM có khả năng học các phụ thuộc dài hạn bằng cách hiệu quả lan truyền và điều chỉnh gradient qua nhiều bước thời gian. Điều này giúp LSTM trở nên mạnh mẽ hơn trong việc xử lý các chuỗi dữ liệu dài và phức tạp so với các mô hình mạng nơ-ron truyền thống.

Ảnh có chứa ảnh chụp màn hình, đồng hồ, văn bản, thiết kế

Mô tả được tự động tạo

**Cổng quên (Forget gate)**: Giúp điều chỉnh lượng thông tin từ trạng thái trước đó được giữ lại, tránh việc gradient bị giảm quá nhanh.

**Cổng đầu vào (Input gate)**: Điều chỉnh lượng thông tin mới được thêm vào, giúp mô hình học các thông tin quan trọng hơn.

**Cổng đầu ra (Output gate)**: Điều chỉnh lượng thông tin từ trạng thái nhớ được sử dụng để tạo đầu ra.