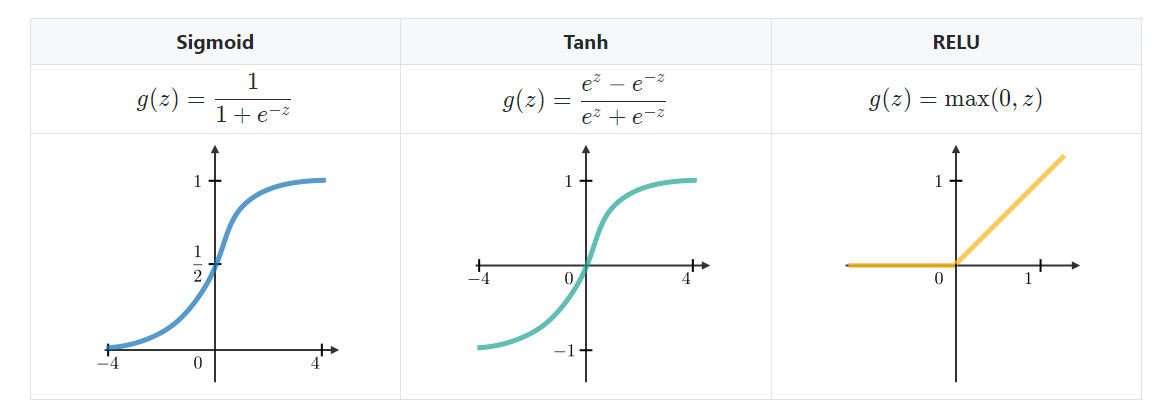
Diagram

Description automatically generated

Diagram

Description automatically generated

Hình biểu diễn cách hoạt động của RNN



Hình Các hàm kích hoạt thường dùng

Vậy vanishing gradient(Đạo hàm bị triệt tiêu) trong rnn là gì ?

Vì hàm kích hoạt (tanh hay sigmoid) của ta sẽ cho kết quả đầu ra nằm trong đoạn [−1,1] (với sigmoid là [0,1]) nên đạo hàm của nó sẽ bị đóng trong khoảng [0,1] (với sigmoid là [0, 0.25]).

Diagram

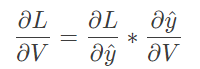
Description automatically generated

Ví dụ 1 mạng RNN

Cách tính toán đạo hàm bằng backpropagation through time in RNN: để cập nhập hệ số W.

Ta sẽ kí hiệu W, U, V là 3 hệ số trong RNN mà chúng ta cần cập nhập trong RNN.

Tính đạo hàm V:



Nhìn đây ta thấy V khá đơn giản:

Tính đạo hàm của U, W:

Text

Description automatically generated with low confidence

Do s(30)  = f(W \* s(29) + V\* x(30)) nên s(29) phụ thuộc vào W nên áp dụng công thức ta có:

Text

Description automatically generated

Suy ra:

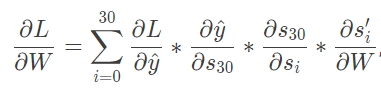
Box and whisker chart

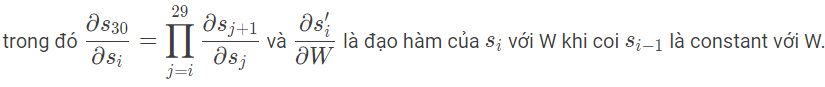
Description automatically generated with low confidence

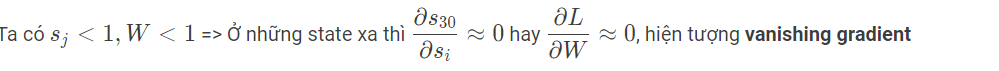
Trong đây Text

Description automatically generatedlà đạo hàm của s(30) với W khi coi s(29) với hằng số W:

Áp dụng công thức chain rule suy ra:







Đây là hiện tượng vanishing gradient khi đạo hàm sấp sỉ bằng 0 và không thể cập nhập hệ số,

Và để giảm thiểu hiện tượng này một mô hình mới xuất hiện: **LSTM**

=========================================================

LSTM

Mô hình LSTM (Long Short-Term Memory) là một loại mạng RNN được giới thiệu bởi Hochreiter và Schmidhuber vào năm 1997. Mô hình LSTM được thiết kế để giải quyết vấn đề vanishing gradient bằng cách sử dụng các cơ chế gọi là "cổng" để kiểm soát thông tin đi qua các layer. Mỗi cổng được điều khiển bởi một hàm sigmoid, cho phép mô hình quyết định xem liệu nội dung thông tin nên được giữ lại hay loại bỏ.

Mô hình LSTM bao gồm ba cổng chính: cổng quên (forget gate), cổng đầu vào (input gate) và cổng đầu ra (output gate). Các cổng này cho phép mô hình quyết định xem liệu thông tin cần được giữ lại hay bị loại bỏ. Trong quá trình lan truyền thuận, mô hình LSTM tính toán các giá trị của các cổng và lượng thông tin mới cần được thêm vào bằng cách kết hợp với thông tin từ các layer trước đó. Điều này giúp giữ lại thông tin quan trọng và loại bỏ thông tin không quan trọng.

Diagram

Description automatically generated

Hình thể hiện cơ chế của LSTM

Diagram

Description automatically generated

Cơ chế LSTM

Output: *ct*​,*ht*​, ta gọi ct là cell state, ht là hidden state.

Input: *ct*−1​,*ht*−1​,*xt*​. Trong đó *xt*​ là input ở state thứ t của model. *ct*−1​,*ht*−1​ là output của layer trước. **h** đóng vai trò khá giống như **s**ở RNN, trong khi **c**là điểm mới của LSTM.

Cell state là 1 bộ nhớ lưu trữ dài hạn được tích luỹ trong thời gian từ input data (lưu trữ thông tin quan trọng) nó sẽ update sau mỗi lần đi vào 3 cổng mới (input gate, forget gate, output gate)

Forget gate: đóng vai trò là quyết định xem lấy bao nhiêu từ cell state

Input gate: lấy bao nhiêu từ của hidden state trước

Output gate: lấy bao nhiêu từ cell state để trở thành output của hidden state

Cell gate: quyết định bao nhiêu từ được them vào cell state dựa vào current input và output trước đó

Ngoài ra ht còn đóng vai trò là output yt để dự đoán kết quả từ.

Tại sao LSTM lại khắc phục được vanishing gradient. Nghĩa là gradient giảm dần và biến mất khi lan truyền ngươck trong quá trình huấn luyện. Khi gradient biến bất các trọng số không được cập nhập đúng mức, làm giảm độ chính xác của mô hình.

Một trong lý do LSTM khắc phục được vấn đề bởi vì cấu trúc đặc biệt của nó. Nó dùng các gates để điều chỉnh thông tin được truyền qua mạng, giúp mô hình quyết định những thông tin nào được giữ lại và những thông tin nào bị bỏ qua. Các cổng này gồm input, output, forget, cell.

Các cổng này giúp LSTM có khả năng học được các phụ thuộc dài hạn trong dữ liệu chuỗi. Khi dữ liệu được truyền qua mạng LSTM, các cổng này quyết định lượng thông tin nào được giữ lại trong cell state (bộ nhớ dài hạn của mạng) và hidden state (bộ nhớ ngắn hạn của mạng). Do đó, giá trị gradient không giảm dần như trong RNN mà vẫn được giữ ở mức ổn định, giúp mô hình học tốt hơn và cải thiện độ chính xác.

Tóm lại, các cổng trong LSTM giúp giải quyết vấn đề vanishing gradient bằng cách cho phép mô hình lưu trữ thông tin quan trọng và quyết định những thông tin nào được giữ lại. Vì vậy, LSTM hiệu quả hơn trong việc học các phụ thuộc dài hạn trong dữ liệu chuỗi so với RNN.

Về thuật toán backprogation:

**LSTM Chống vanishing gradient**

Diagram

Description automatically generated

Thành phần chính gây ra hiện tượng vanishing gradient RNN là Logo, company name

Description automatically generated

Tương tự ta quan tâm tới A picture containing text, clock, watch

Description automatically generateddo 0 < ft < 1 nên cơ bản LSTM vẫn bị vanishing gradient nhưng ít hơn RNN. Và khi mang thông tin trên cell state ít khi phải quên giá trị cell cũ nên ft 1 => tránh được vanishing gradient.