



deeplearning.ai

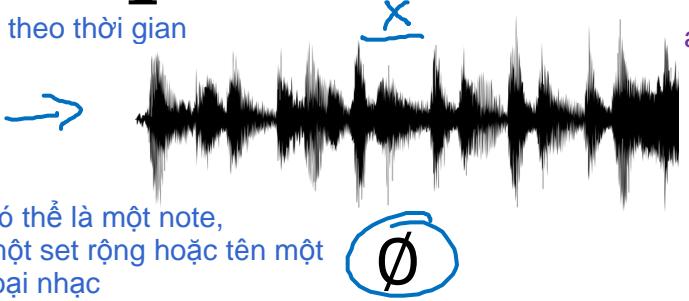
Recurrent Neural Networks

Why sequence
models?

Examples of sequence data

Các vấn đề này có thể giải quyết dạng supervised learning với X và label y. Ở đây có nhiều loại vấn đề liên quan đến sequence. X, y có thể có độ dài khác nhau, có ví dụ chỉ X hoặc Y là sequence

Speech recognition



audio clip sang đoạn text

y
“The quick brown fox jumped over the lazy dog.”

Music generation



Sentiment classification
phân loại cảm xúc, thái độ

“There is nothing to like
in this movie.” sequence



DNA sequence analysis → AGCCCCTGTGAGGAAC TAG

AG **CCCCTGTGAGGAAC** TAG
gán nhãn phần nào của chuỗi DNA tương ứng với protein

Machine translation

Voulez-vous chanter avec
moi?

Do you want to sing with
me?

Video activity recognition



Running

Name entity recognition

Nhận dạng tên người

Yesterday, Harry Potter
met Hermione Granger.

Yesterday, **Harry Potter**
met **Hermione Granger**.
Andrew Ng

Cho mỗi chuỗi, nhận dạng chữ nào tương ứng
với tên người



deeplearning.ai

Recurrent Neural Networks

Notation

Motivating example

NLP

Bài toán Name entity recognition: nhận dạng tên người trong sequence.

Có thể được ứng dụng trong search engine để tìm kiếm tên người, tên công ty, tên nước, địa chỉ (lúc mình tìm kiếm trên mạng chẳng hạn)

x:

(Harry Potter) and (Hermione Granger) invented a new spell.

superscript angle bracket

$\rightarrow \underline{x}^{<1>} \quad x^{<2>} \quad x^{<3>} \quad \dots \quad x^{<t>} \quad \dots \quad x^{<9>}$

length of the input sequence $T_x = 9$

$\rightarrow y:$

| | 0 | | 0 0 0 0
 $y^{<1>} \quad y^{<2>} \quad y^{<3>} \quad \dots \quad y^{<t>} \quad \dots \quad y^{<9>}$

length of the output sequence $T_y = 9$

Mỗi từ nhận giá trị 1 hoặc 0 tương ứng với tên người hoặc ko phải

$x^{(i)<t>} \quad y^{(i)<t>} \uparrow$

round bracket chỉ i-th training example

$T_x^{(i)} = 9 \quad 15$

$T_y^{(i)}$

Representing words

Cách biểu diễn từ trong sequence

$$\begin{array}{c} \times^{<t>} \\ \times \rightarrow \textcolor{blue}{y} \\ (\textcolor{blue}{x}, \textcolor{blue}{y}) \end{array}$$

x:

Harry Potter and Hermione Granger invented a new spell.

Có thể tạo vocabulary

Vocabulary

a	1
aaron	2
:	:
and	367
:	:
harry	4075
:	:
potter	6830
:	:
zulu	10,000
<u><UNK></u>	10,000

cho những
từ ko có
trong vocab

$x^{<1>}$

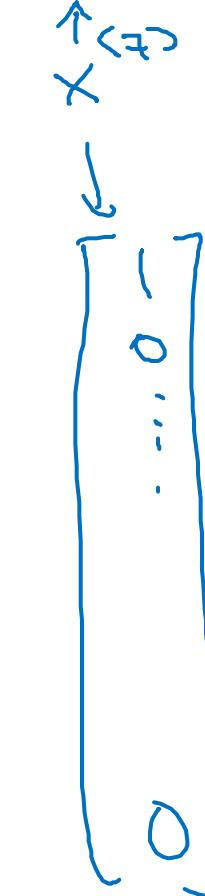
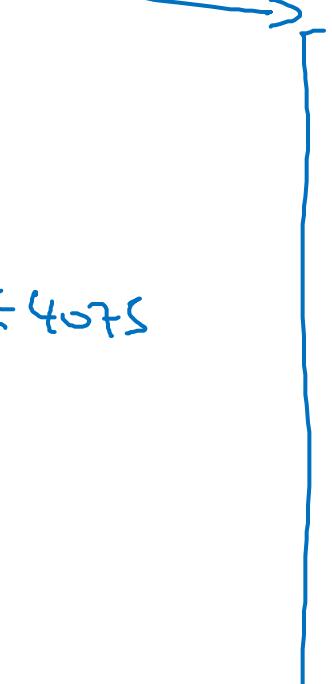
$x^{<2>}$

$x^{<3>}$

...

$x^{<\textcolor{red}{t}>}$

$x^{<9>}$



Dùng one-hot
vector để biểu
diễn các từ (tương
ứng với vị trí trong
vocabulary)

Chúng ta mới dễ dàng
xử lý bài toán dạng
supervised learning

One-hot

ví dụ có thể được lấy từ 10000 từ tiếng Anh thông dụng nhất chẳng hạn

Andrew Ng

Representing words

x: Harry Potter and Hermione Granger invented a new spell.

$x^{<1>} \quad x^{<2>} \quad x^{<3>} \quad \dots \quad x^{<9>}$

And = 367
Invented = 4700
A = 1
New = 5976
Spell = 8376
Harry = 4075
Potter = 6830
Hermione = 4200
Gran... = 4000



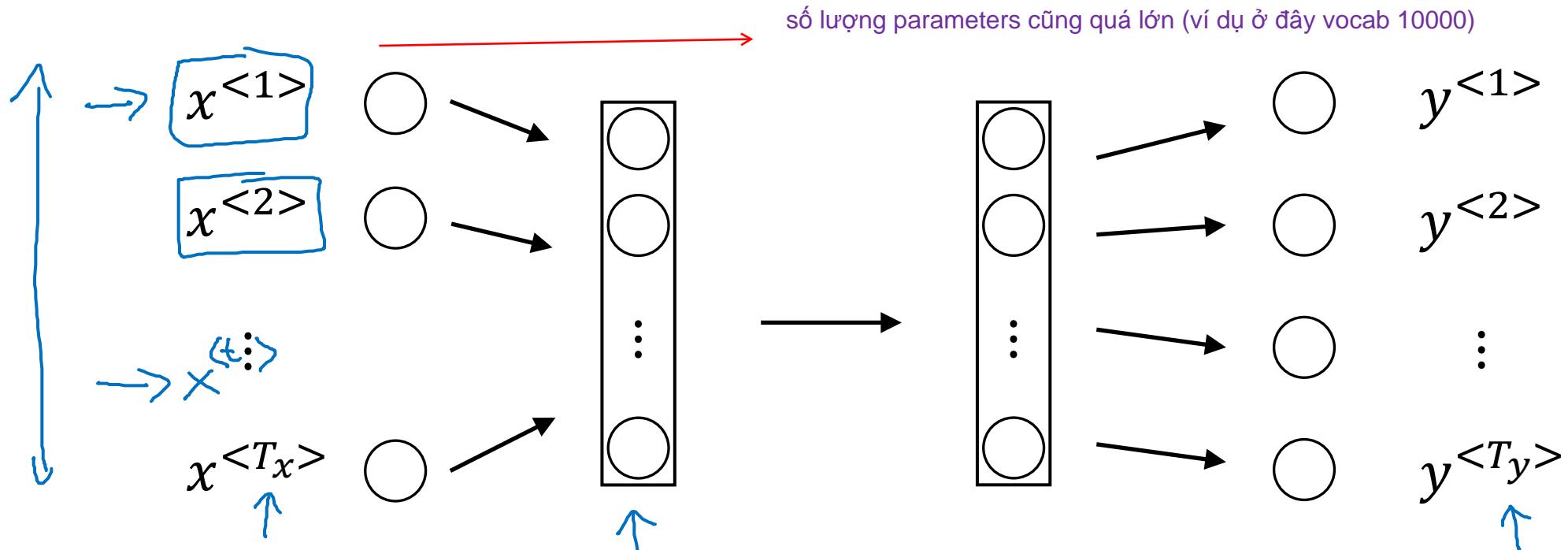
deeplearning.ai

Xây dựng NN để học mapping từ X -> y trong bài toán sequence

Recurrent Neural Networks

Recurrent Neural Network Model

Why not a standard network?



Nếu dùng mạng NN bình thường để giải quyết bài toán sequence trên (nhận diện tên người) thì hoạt động ko được tốt

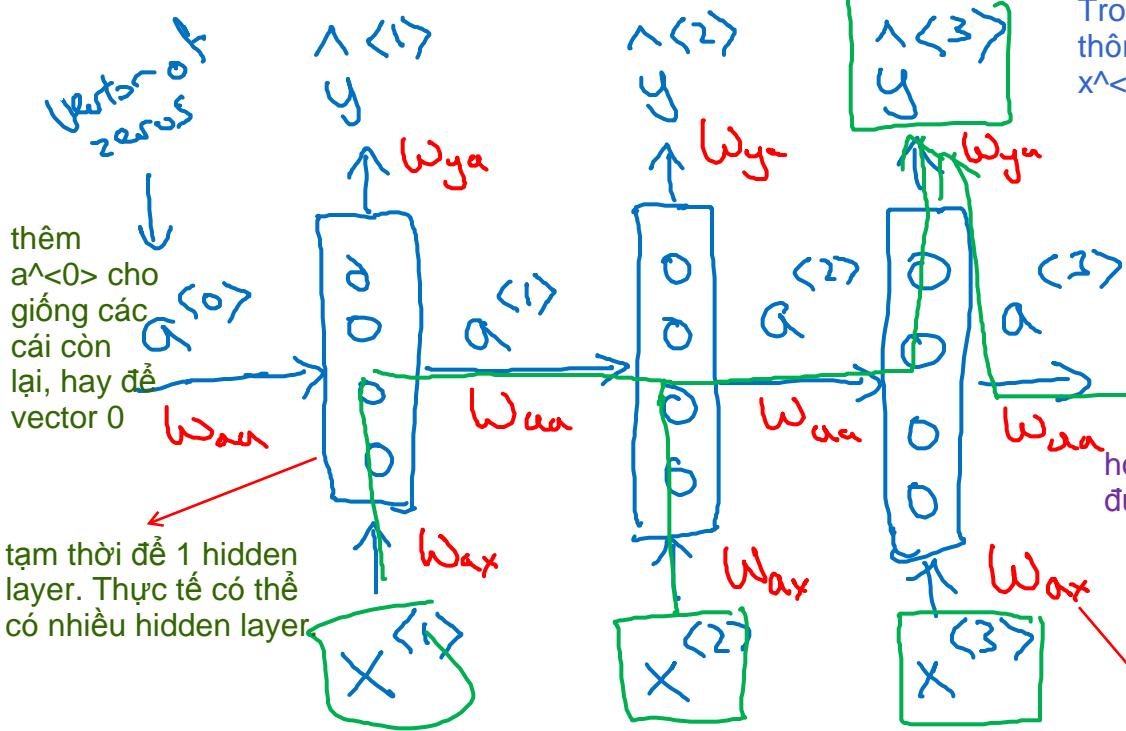
Problems:

input và output có thể có các độ dài khác nhau ở các ví dụ khác nhau nên ko tổng quát được, ngay cả khi cho input length = MAX nào đó rồi thêm pad vào cho đủ nhưng cách biểu diễn này cũng không tốt

- - Inputs, outputs can be different lengths in different examples.
- - Doesn't share features learned across different positions of text.

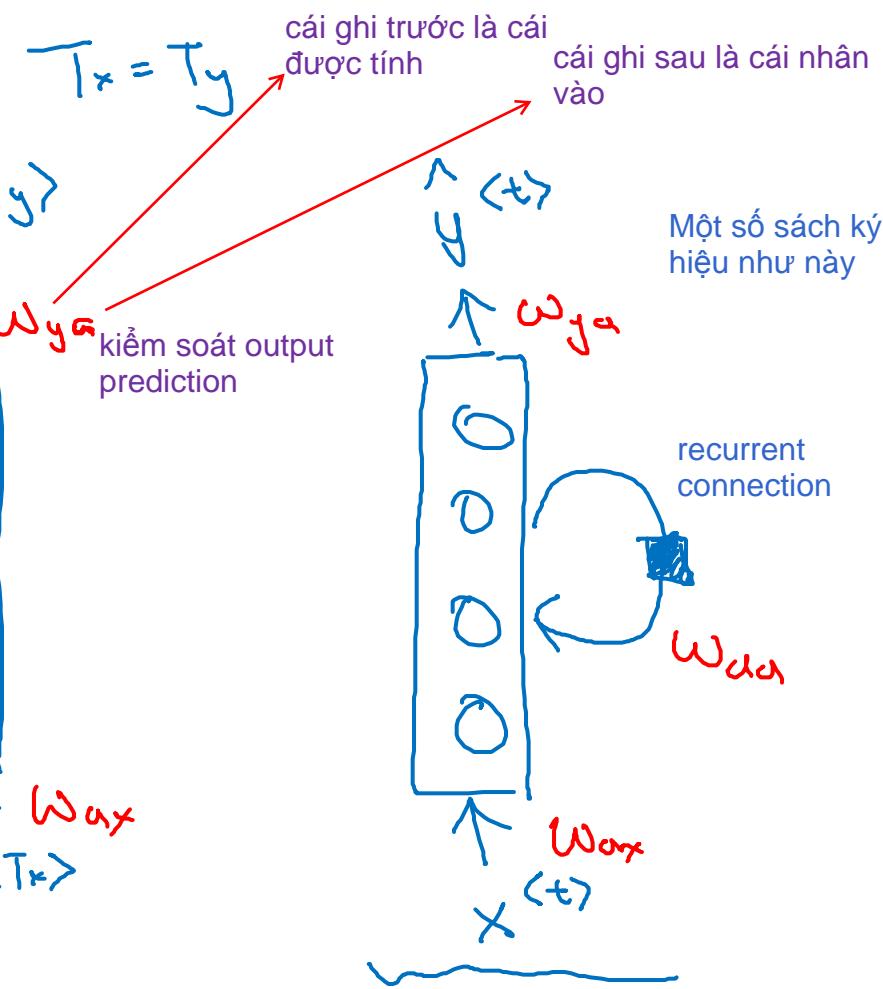
ví dụ Harry xuất hiện ở vị trí 1 có khả năng là một phần của tên người thì Harry xuất hiện chỗ khác trong câu nhiều khả năng cũng là tên người. Cái này như CNN tôngr quát hóa từ phần này đến phần khác. Mạng NN đơn giản thì không được

Recurrent Neural Networks



Trong RNN khi dự đoán $y^{<3>}$ thông tin không chỉ được lấy từ $x^{<3>}$ mà còn từ $x^{<1>}$ và $x^{<2>}$

Nhược điểm của RNN là nó ko sử dụng các từ đứng sau để dự đoán cho từ đứng trước



có từ $x^{<1>}$ sẽ đưa qua hidden layer để dự đoán $y^{<1>}$, tương tự $x^{<2>}$ đưa qua hidden layer để dự đoán $y^{<2>}$. Tuy nhiên $a^{<1>}$ activation value từ time step 1 được đưa vào time step 2.

Tương tự như vậy với các time step khác.

He said, "Teddy

Roosevelt was a great President."

Chính vì lý do này nó được gọi là Recurrent Neural Networks.

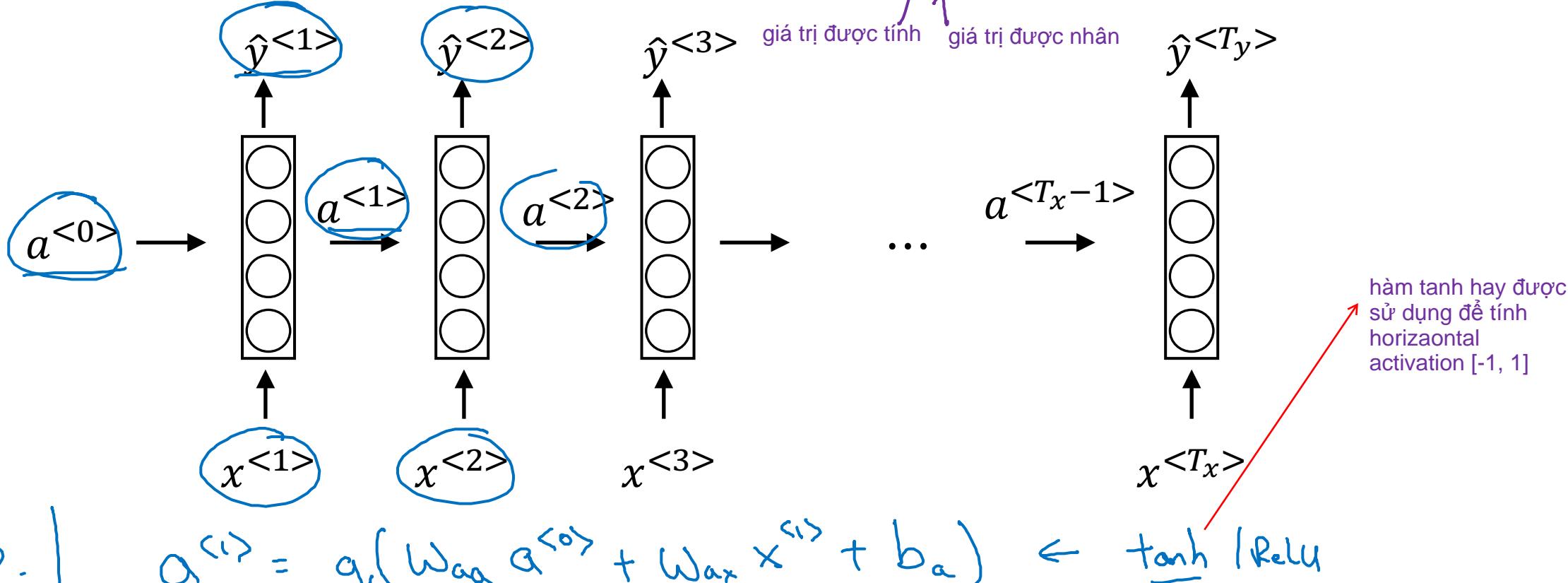
Phần này cứ dự đoán đầu ra từng từ một

He said, "Teddy

bears are on sale!"

Kiểm soát kết nối từ $x^{<t>}$ đến hidden layer W_{ax}

Forward Propagation



nằm trong [0, 1] nếu là binary classification
hoặc softmax nếu nhiều classes

dựa vào đây có thể tính được số lượng parameters

Simplified RNN notation

$$a^{<t>} = g(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$$

↑ 100 ↑ 10,000
 $(100, 100)$ $(100, 10,000)$

$$\hat{y}^{<t>} = g(W_{ya}a^{<t>} + b_y)$$

chỉ đang tính y

$$y^{<t>} = g(W_y a^{<t>} + b_y)$$

↑ ↑
cụ thể hơn là $(100, 1)$

ví dụ $a^{<t-1>}$ có chiều là 100 thì W_{aa} có chiều là $(100, 100)$

ví dụ $x^{<t>}$ có chiều là 10000 (size of vocabulary) thì W_{ax} có chiều là $(100, 10000)$

cụ thể hơn là $(10000, 1)$ - vector cột

chỉ việc đang tính a

$$a^{<t>} = g(W_a [a^{<t-1>}, x^{<t>}] + b_a)$$

stack horizontally

$$\begin{bmatrix} W_{aa} & W_{ax} \end{bmatrix} = W_a$$

↑ 100 ↓ 10000
 $(100, 100)$ $(100, 10000)$

do xếp theo chiều ngang

$$[a^{<t-1>}, x^{<t>}] = \begin{bmatrix} a^{<t-1>} \\ x^{<t>} \end{bmatrix} = W_{aa}a^{<t-1>} + W_{ax}x^{<t>}$$

↑ 100 ↓ 10000
 $(100, 10100)$ $(10100, 1)$ được $(100, 1)$

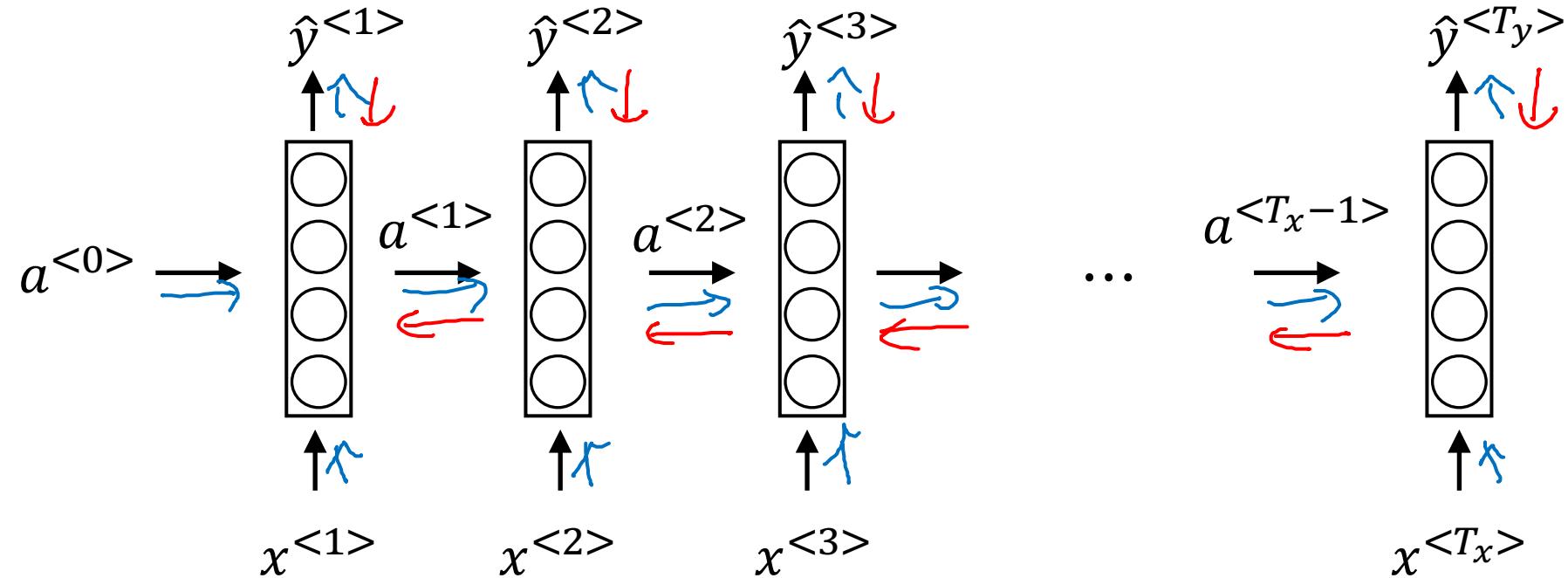


deeplearning.ai

Recurrent Neural Networks

Backpropagation through time

Forward propagation and backpropagation



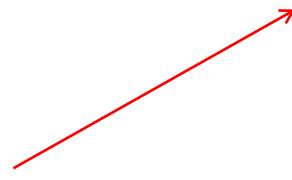
Màu xanh: forward propagation

Màu đỏ: backward propagation

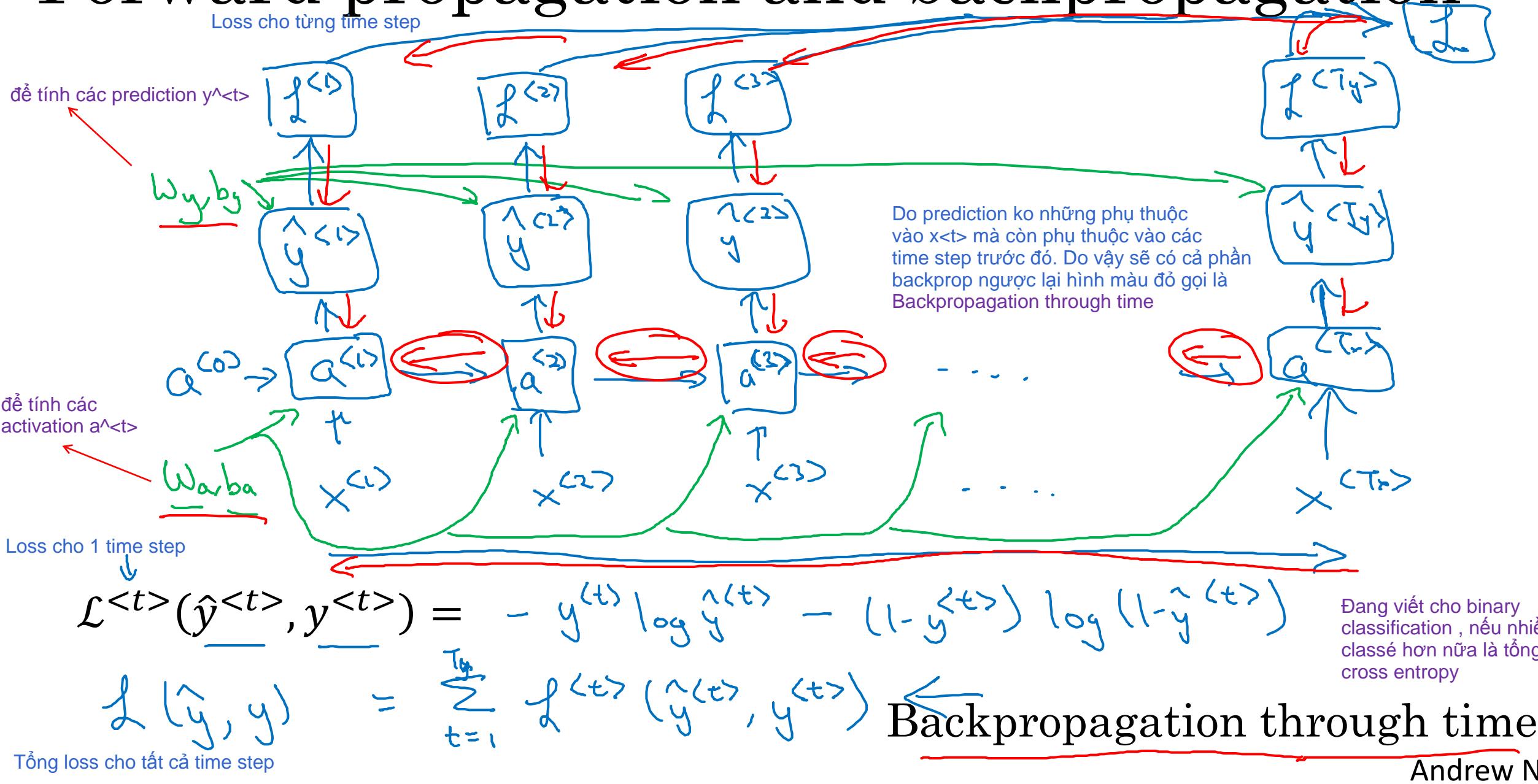
Ví dụ đạo hàm của loss theo W_y sẽ bằng tích đạo hàm của loss theo prediction * đạo hàm của prediction theo W_y (sau đó cần lấy tổng vì có nhiều time step).

Tương tự với các hệ số khác cũng vậy, ở đây phải dùng chain rule.

tất cả các activation phụ thuộc vào W_a và b_a



Forward propagation and backpropagation





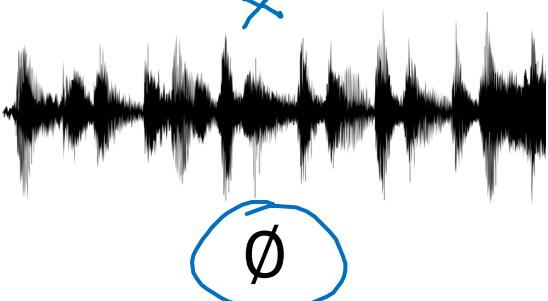
deeplearning.ai

Recurrent Neural Networks

Different types of RNNs

Examples of sequence data

Speech recognition



T_x T_y

y

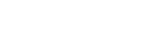
“The quick brown fox jumped
over the lazy dog.”

Music generation



Sentiment classification

“There is nothing to like
in this movie.”



DNA sequence analysis

AGCCCCTGTGAGGAAC TAG



AG~~CCCCTGTGAGGAAC~~ TAG

Machine translation

Voulez-vous chanter avec
moi?



Do you want to sing with
me?

Video activity recognition



Running

Name entity recognition

Yesterday, Harry Potter
met Hermione Granger.



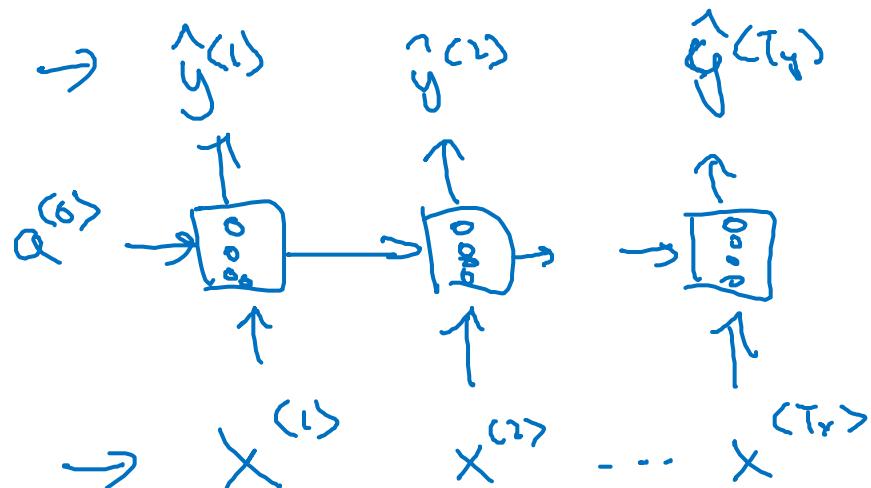
Yesterday, **Harry Potter**
met **Hermione Granger**.

Andrew Ng

Examples of RNN architectures

Độ dài của input và output có thể khác nhau

$$T_x = T_y$$



Many-to-many

bởi vì input và output sequences có nhiều inouts và outputs

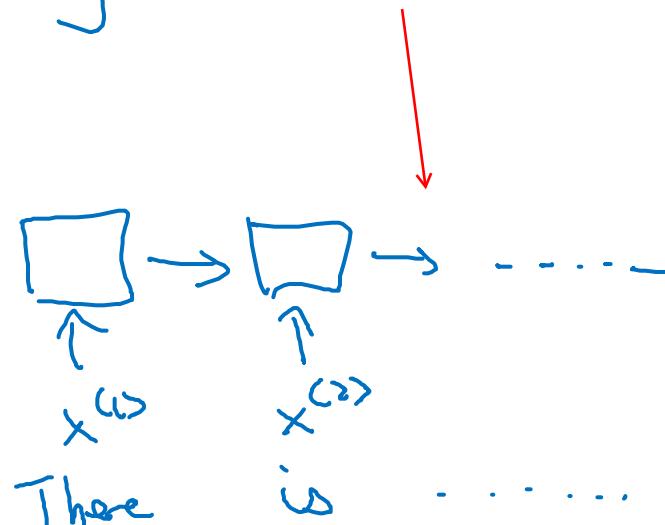
Sentiment classification

$x = \text{text}$

$y = 0/1$

1--5

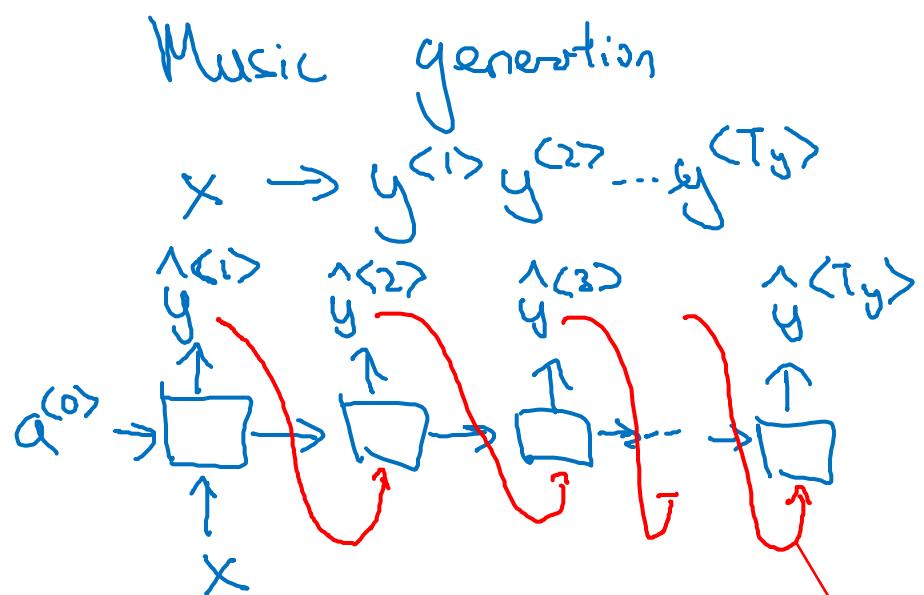
có một output, bài toán phân loại



Many - to - one

One - to - one

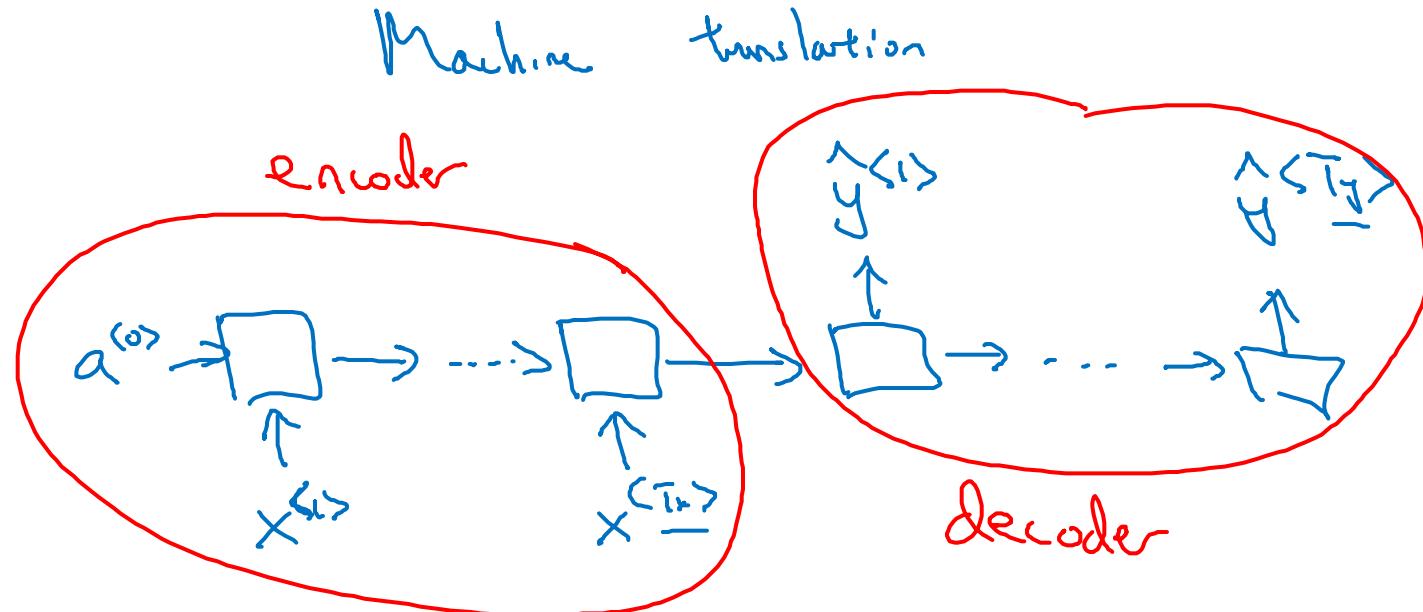
Examples of RNN architectures



One-to-many

$$x = \phi$$

có một input, sau đó lần lượt tạo ra các nốt nhạc chẳng hạn (nốt trước làm đầu vào để sinh ra note sau)

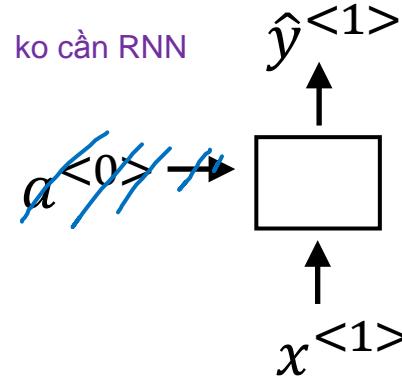


Many - to - many

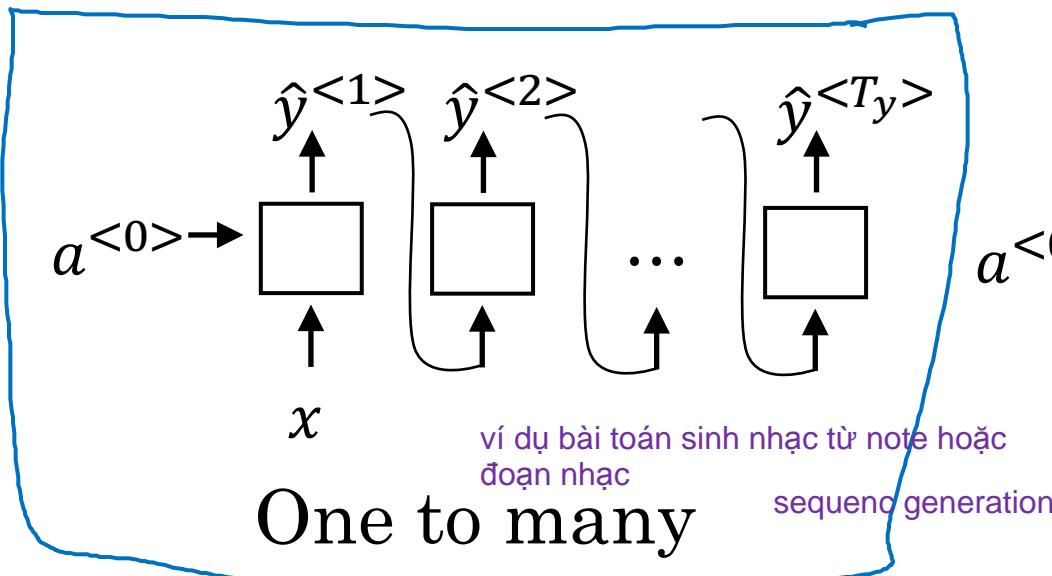
bài toán dịch

chiều dài input và output
có thể khác nhau

Summary of RNN types

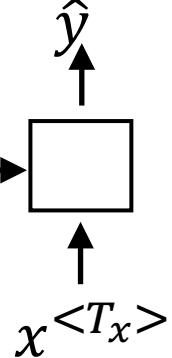


One to one

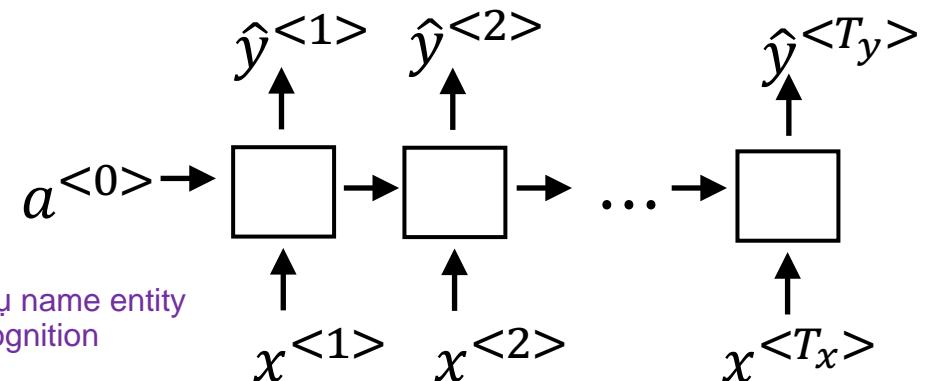


bài toán classification text chẵng hạn

phân loại text, đánh giá phim

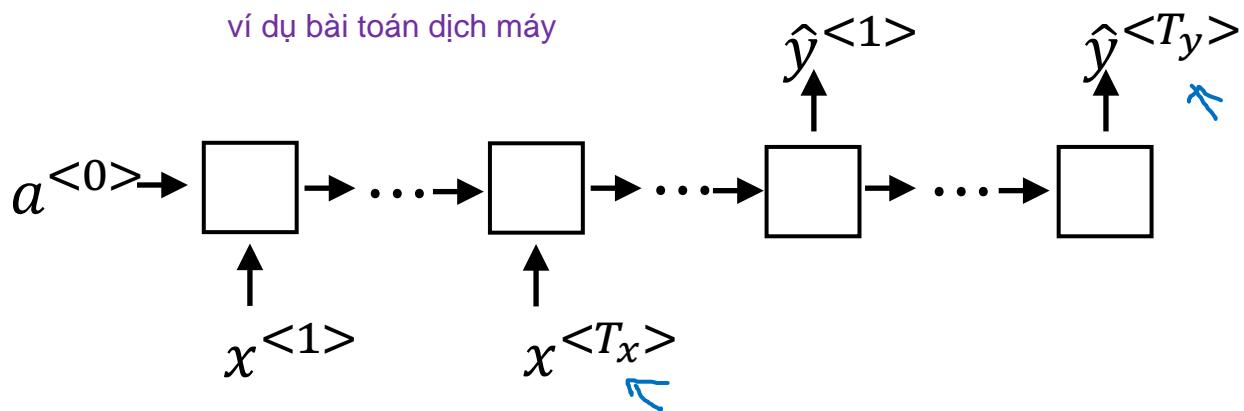


Many to one



Many to many

$$T_x = T_y$$



Many to many



deeplearning.ai

Recurrent Neural Networks

Language model and sequence generation

What is language modelling?

Speech recognition

Tùy đoạn audio trả về đoạn text

The apple and pair salad.

language model sẽ cho chúng ta probability của 2 câu này để lựa chọn câu nào có probability cao nhất

→ The apple and pear salad.

$$P(\text{The apple and pair salad}) = 3.2 \times 10^{-13}$$

$$P(\text{The apple and pear salad}) = 5.7 \times 10^{-10}$$

$$P(\text{Sentence}) = ?$$

$$P(y^{(1)}, y^{(2)}, \dots, y^{(T)})$$

Đây là nhiệm vụ của language model

là xác định xác suất của câu này

Language modelling with an RNN

NHIỆM VỤ TÍNH XÁC SUẤT CỦ TỪNG CÂU

Tokenize cho các input sentence

Đầu tiên tokenize các từ trong câu ví dụ có thể dùng one-hot vector dựa trên vocabulary

Cats average 15 hours of sleep a day. <EOS>

$$y^{(1)} \quad y^{(2)} \quad y^{(3)} \quad \dots$$

$$y^{(s)} \quad y^{(a)}$$

The Egyptian Mau is a breed of cat. <EOS>

10,000

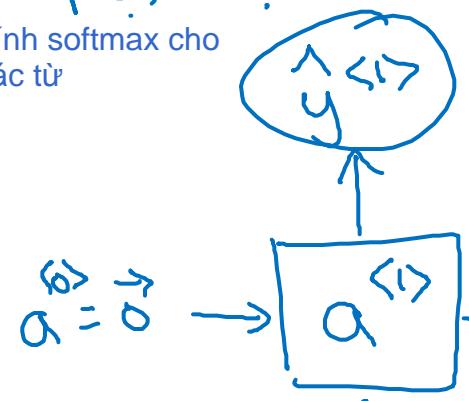
SUNK

Trong nhiều trường hợp từ không có trong vocabulary thì mình sử dụng token <UNK>

RNN model

$p(a) p(a|a) \dots p(a|cats) \dots p(a|tulu)$
 $p(c|a) p(c|cats) \dots p(c|tulu)$
 $p(y|a) p(y|cats) \dots p(y|tulu)$

Tính softmax cho các từ



Do ở đây mình đang tính xác suất của câu nên đầu vào sẽ để $x^<1>=0$

Giống như kiểu sinh ra cả câu từ empty set

\rightarrow Cats average 15 hours of sleep a day. <EOS>

tại 1 time step

$$\mathcal{L}(\hat{y}^{<t>}, y^{<t>}) = - \sum_i y_i^{<t>} \log \hat{y}_i^{<t>}$$

$$\mathcal{L} = \sum_t \mathcal{L}^{<t>}(\hat{y}^{<t>}, y^{<t>})$$

Tổng loss

biết trước rồi

cái này cũng là softmax layer

những labels (xác suất này phải được xác định trước rồi)

$p(\underline{\quad} | "cats average")$

$p(\underline{\quad} | \dots)$



X chạy từ vector 0 đến sát cuối thôi, vì chữ cuối để dự đoán (nếu đưa vào thì ko có ý nghĩa gì)

day

$$P(y^{<1>}, y^{<2>}, y^{<3>}) \leftarrow \\ = \frac{p(y^{<1>}) p(y^{<2>} | y^{<1>})}{p(y^{<3>} | y^{<1>}, y^{<2>})}$$

$P(1,2,3,4) = P(4|3,2,1) P(3|2,1) P(2|1) P(1)$

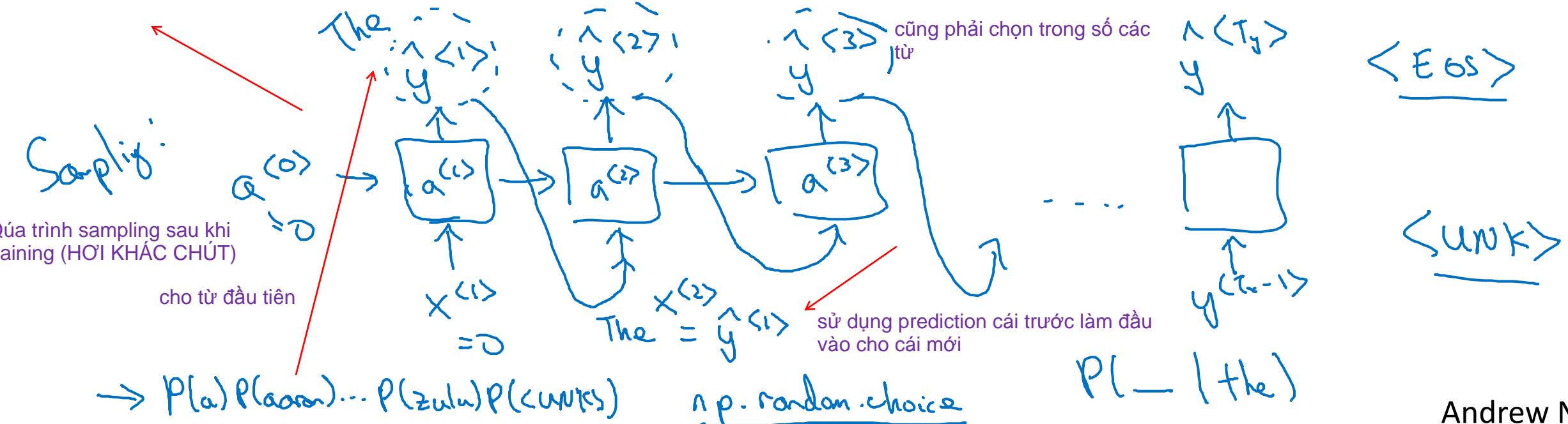
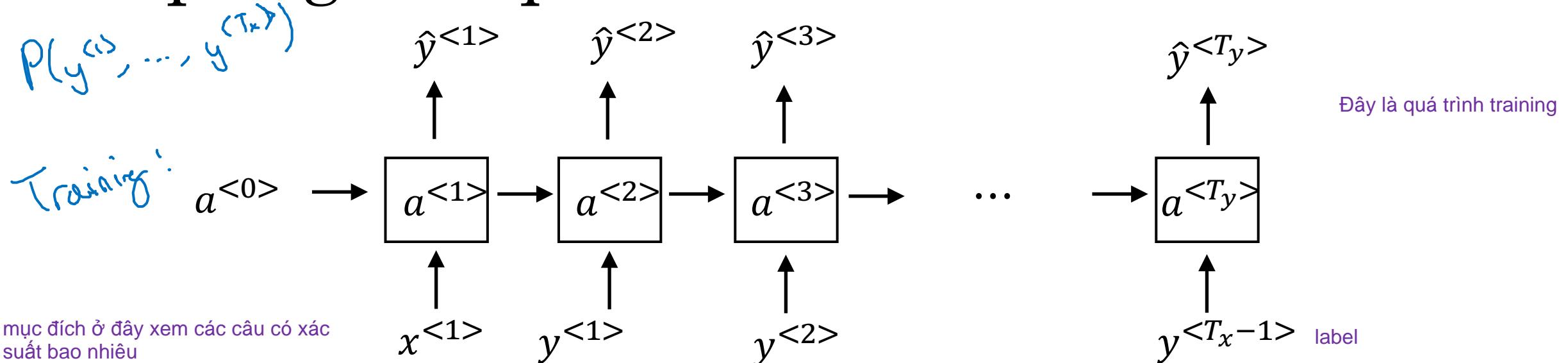


deeplearning.ai

Recurrent Neural Networks

Sampling novel
sequences

Sampling a sequence from a trained RNN



Character-level language model

có thể sử dụng character level để giảm kích thước vocabulary xuống. Tuy nhiên độ chính xác không bằng được word level

→ Vocabulary = [a, aaron, ..., zulu, <UNK>] ←

Những câu dài thì số lượng characters rất nhiều và thường không nắm bắt được sự phụ thuộc lẫn nhau.

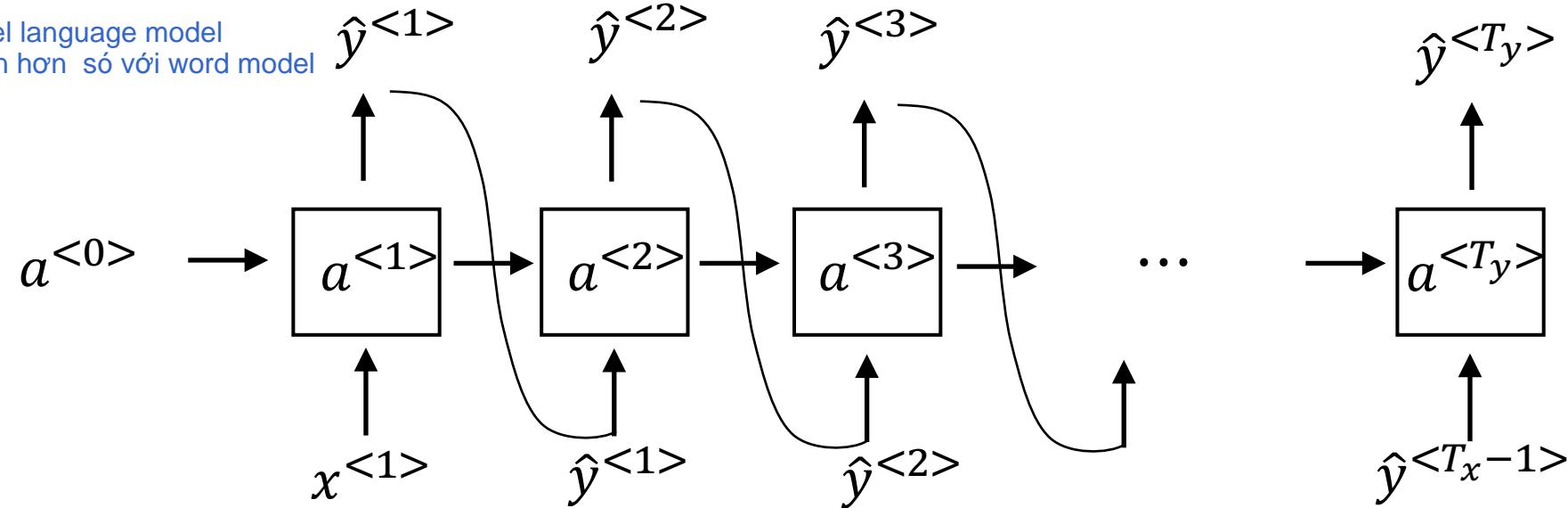
$\rightarrow \text{Vocabulary} = [a, b, c, \dots, z, \cup, \circ, \rightarrow, ;, \circ, \dots, q, A, \dots, z]$

$y_1 \leftarrow y_1 - y_2$

Cat ↑↑↑↑ average ...

May

Train character-level language model
tồn nhiều tài nguyên hơn so với word model



Sequence generation

News

President enrique peña nieto, announced
sench's sulk former coming football langston
paring.

“I was not at all surprised,” said hich langston.

“Concussion epidemic”, to be examined. ←

The gray football the told some and this has on
the uefa icon, should money as.

Shakespeare

The mortal moon hath her eclipse in love.
And subject of this thou art another this fold.
When lesser be my love to me see sabl's.
For whose are ruse of mine eyes heaves.



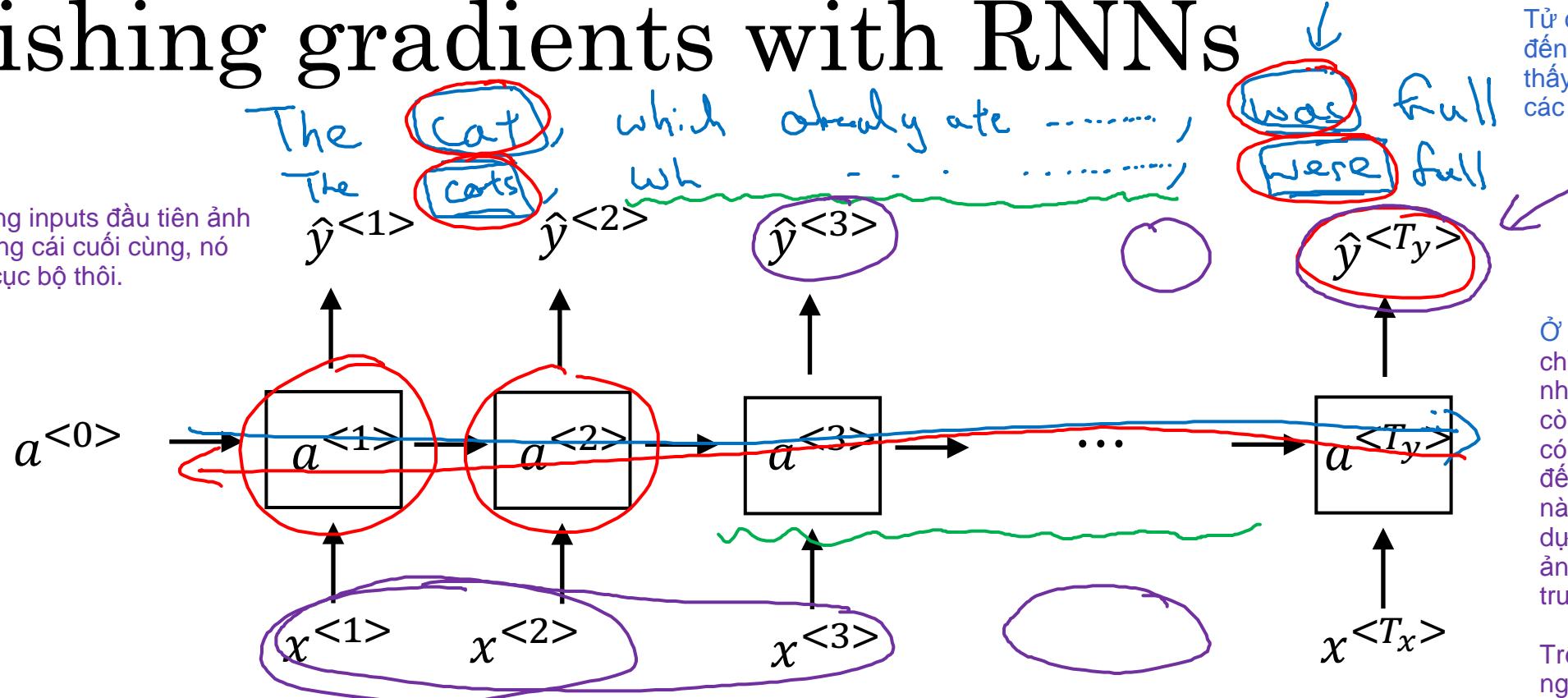
deeplearning.ai

Recurrent Neural Networks

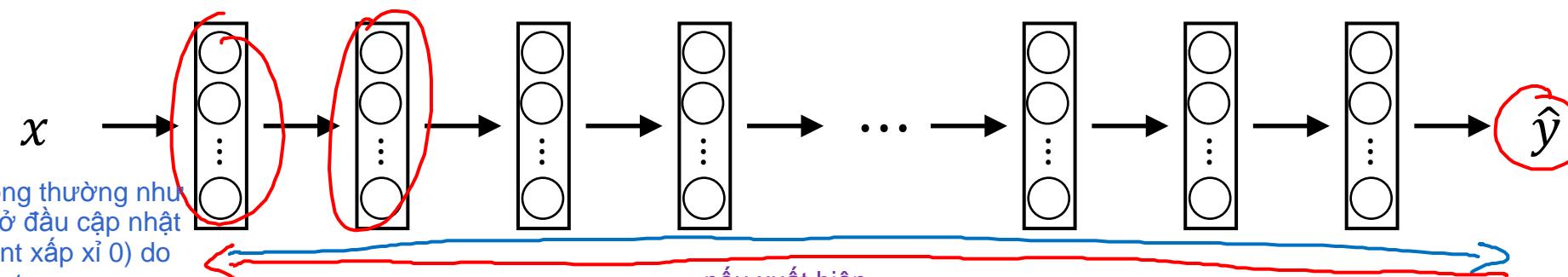
Vanishing gradients with RNNs

Vanishing gradients with RNNs

Rất khó để những inputs đầu tiên ảnh hưởng đến những cái cuối cùng, nó chỉ ảnh hưởng cục bộ thôi.



Tử ở xa có thể ảnh hưởng đến từ khác. Tuy nhiên nhận thấy RNN không nắm bắt tốt các sự phụ thuộc ở xa



những mạng thông thường như
này parameters ở đầu cập nhật
rất chậm (gradient xấp xỉ 0) do
vanishing gradient

Exploding gradients.

nếu xuất hiện
exploding thì
có thể có NaN

Gradient clipping

lớn hơn giá trị nào đó thì cho nó bằng ngưỡng,
khá là đơn giản

100

Xem thêm các tài liệu để
hiểu rõ hơn

Andrew Ng

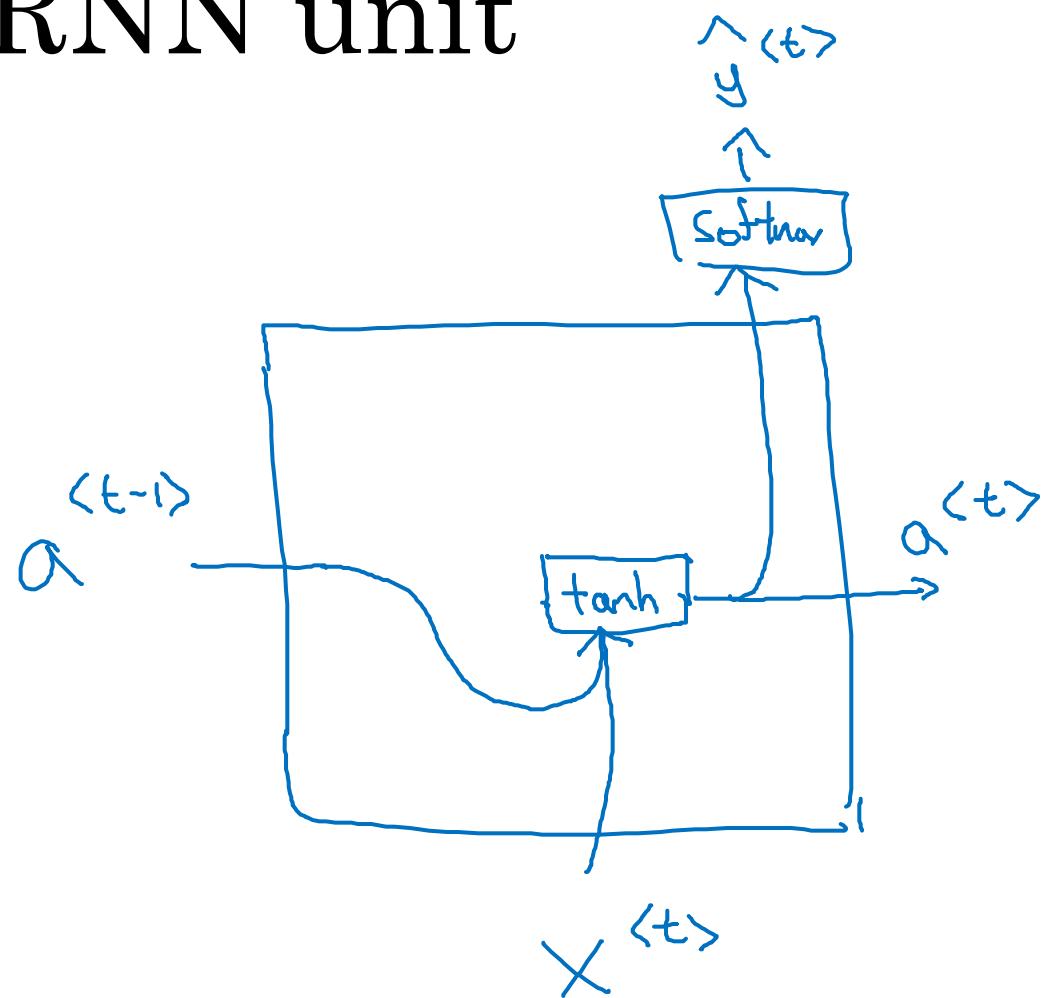


deeplearning.ai

Recurrent Neural Networks

Gated Recurrent Unit (GRU)

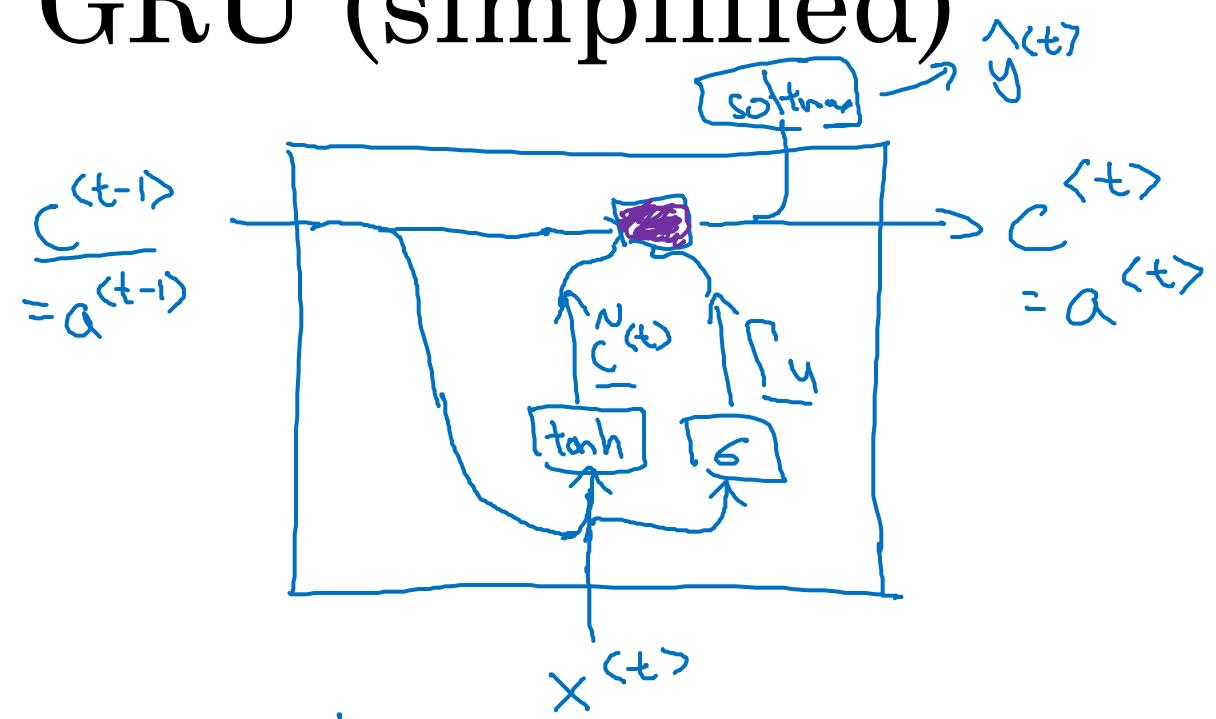
RNN unit



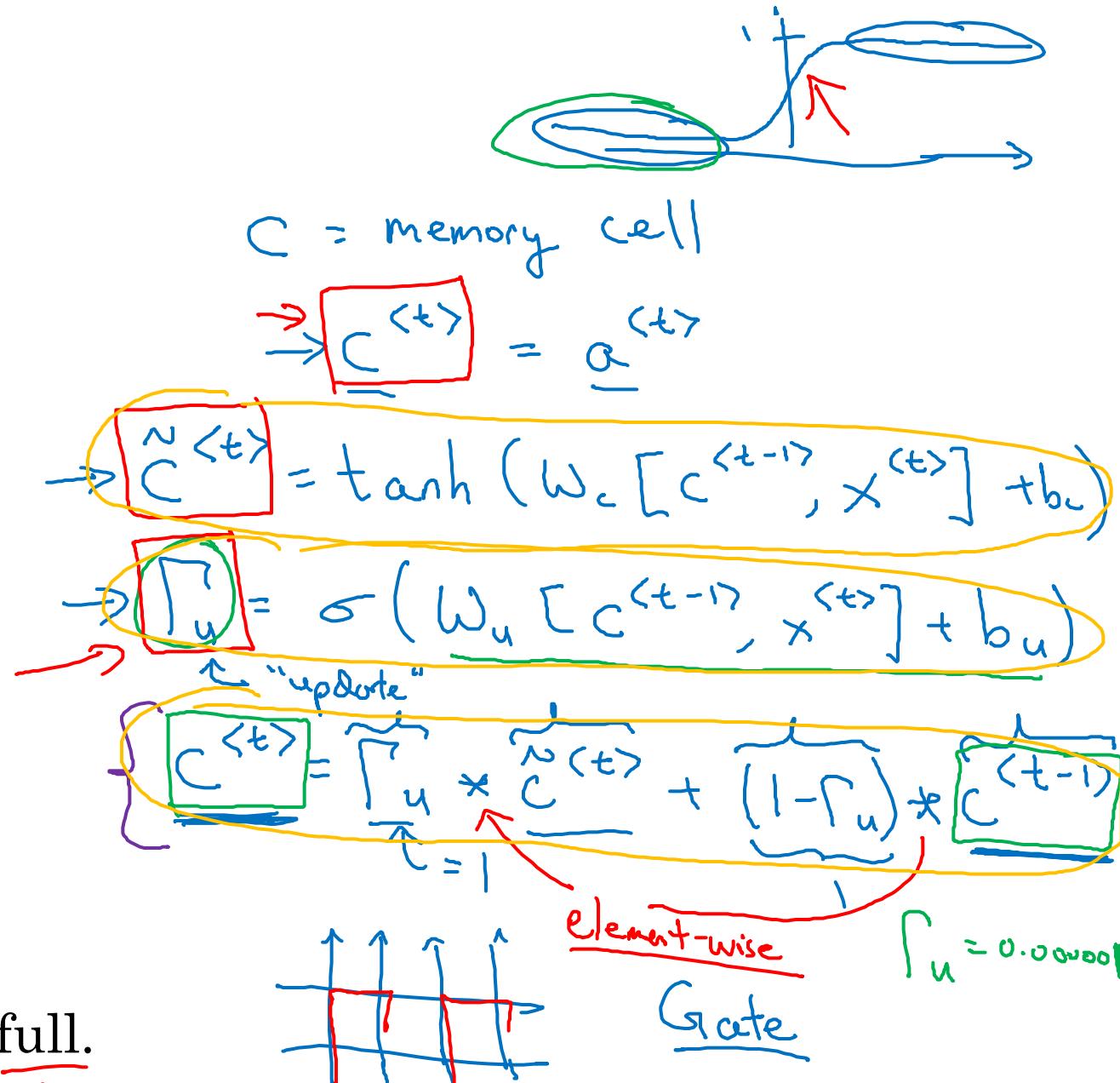
$$\underline{a}^{(t)} = g(W_a[\underline{a}^{(t-1)}, \underline{x}^{(t)}] + b_a)$$

Handwritten annotations above the equation indicate the tanh activation function applied to the input $\underline{a}^{(t-1)}$ and $\underline{x}^{(t)}$. The term W_a is enclosed in a bracket below the equation, and the bias b_a is also enclosed in a bracket.

GRU (simplified)



$\Gamma_u = 1$
 $\Gamma_u = 0 \quad \Gamma_u = 0 \quad \Gamma_u = 0 \quad \dots$
 $\Gamma_u = 1$
The cat, which already ate ..., was full.



[Cho et al., 2014. On the properties of neural machine translation: Encoder-decoder approaches]

[Chung et al., 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling]

Andrew Ng

Full GRU

$$\tilde{c}^{<t>} = \tanh(W_c[\tilde{c}_r^{<t-1>}, x^{<t>}] + b_c)$$

$$u \quad \Gamma_u = \sigma(W_u[c^{<t-1>}, x^{<t>}] + b_u)$$

$$r \quad \Gamma_r = \sigma(W_r[c^{<t-1>}, x^{<t>}] + b_r)$$

LSTM

$$h \quad c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + (1 - \Gamma_u) * c^{<t-1>}$$

The cat, which ate already, was full.



deeplearning.ai

Recurrent Neural Networks

LSTM (long short
term memory) unit

GRU and LSTM

GRU

$$\tilde{c}^{<t>} = \tanh(W_c[\Gamma_r * \underline{c}^{<t-1>}, x^{<t>}] + b_c)$$

$$\underline{\Gamma_u} = \sigma(W_u[c^{<t-1>}, x^{<t>}] + b_u)$$

$$\underline{\Gamma_r} = \sigma(W_r[c^{<t-1>}, x^{<t>}] + b_r)$$

$$c^{<t>} = \underline{\Gamma_u} * \tilde{c}^{<t>} + (1 - \underline{\Gamma_u}) * c^{<t-1>}$$

$$a^{<t>} = c^{<t>}$$



LSTM

$$\tilde{c}^{<t>} = \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c)$$

$$\underline{\Gamma_u} = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u)$$

$$\underline{\Gamma_f} = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f)$$

$$\underline{\Gamma_o} = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o)$$

$$c^{<t>} = \underline{\Gamma_u} * \underline{\tilde{c}^{<t>}} + \underline{\Gamma_f} * \underline{c^{<t-1>}}$$

$$a^{<t>} = \underline{\Gamma_o} * c^{<t>}$$

LSTM units

GRU

$$\tilde{c}^{<t>} = \tanh(W_c[\Gamma_r * c^{<t-1>}, x^{<t>}] + b_c)$$

$$\Gamma_u = \sigma(W_u[c^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_r = \sigma(W_r[c^{<t-1>}, x^{<t>}] + b_r)$$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + (1 - \Gamma_u) * c^{<t-1>}$$

$$a^{<t>} = c^{<t>}$$

LSTM

$$\tilde{c}^{<t>} = \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c)$$

$$\Gamma_u = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f)$$

$$\Gamma_o = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o)$$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>}$$

$$a^{<t>} = \Gamma_o * c^{<t>}$$

LSTM in pictures

$$\tilde{c}^{<t>} = \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c)$$

$$\Gamma_u = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u)$$

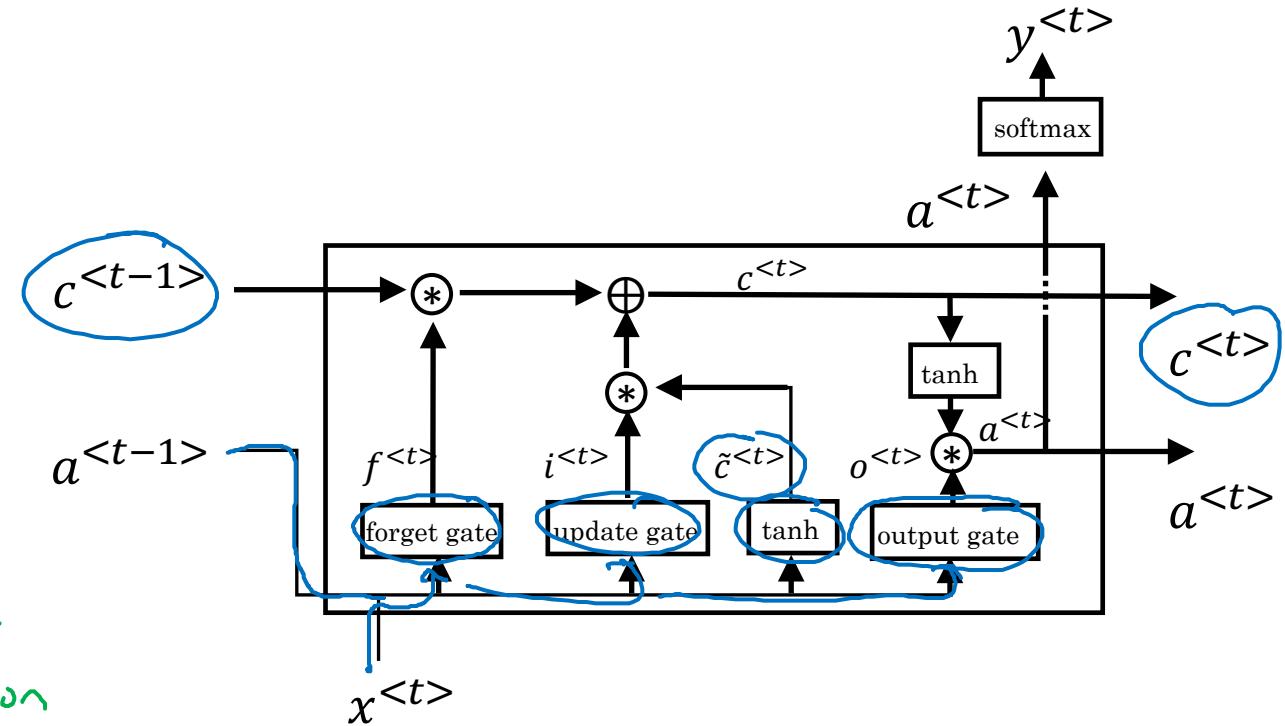
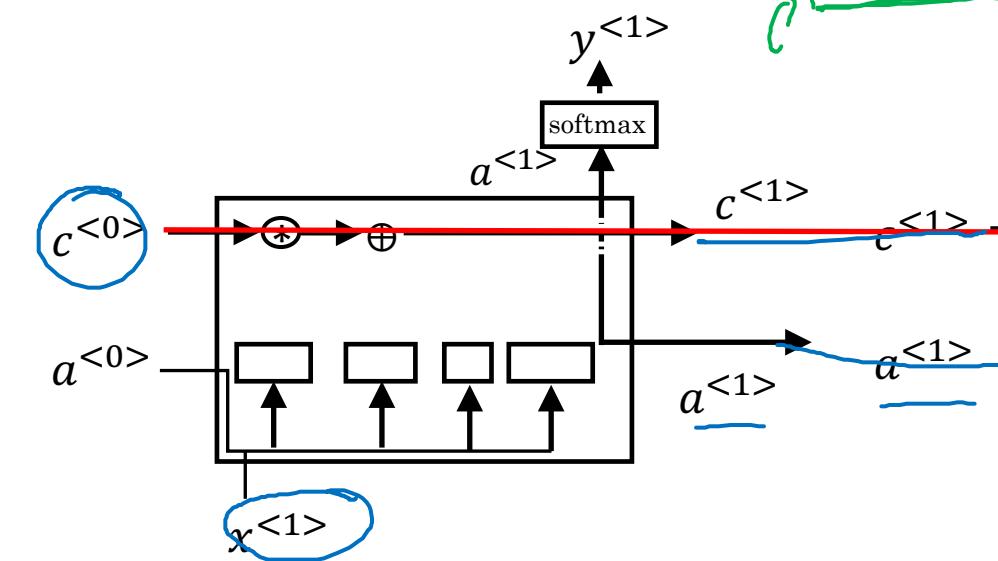
$$\Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f)$$

$$\Gamma_o = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o)$$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>}$$

$$a^{<t>} = \Gamma_o * c^{<t>}$$

peephole connection



Andrew Ng



deeplearning.ai

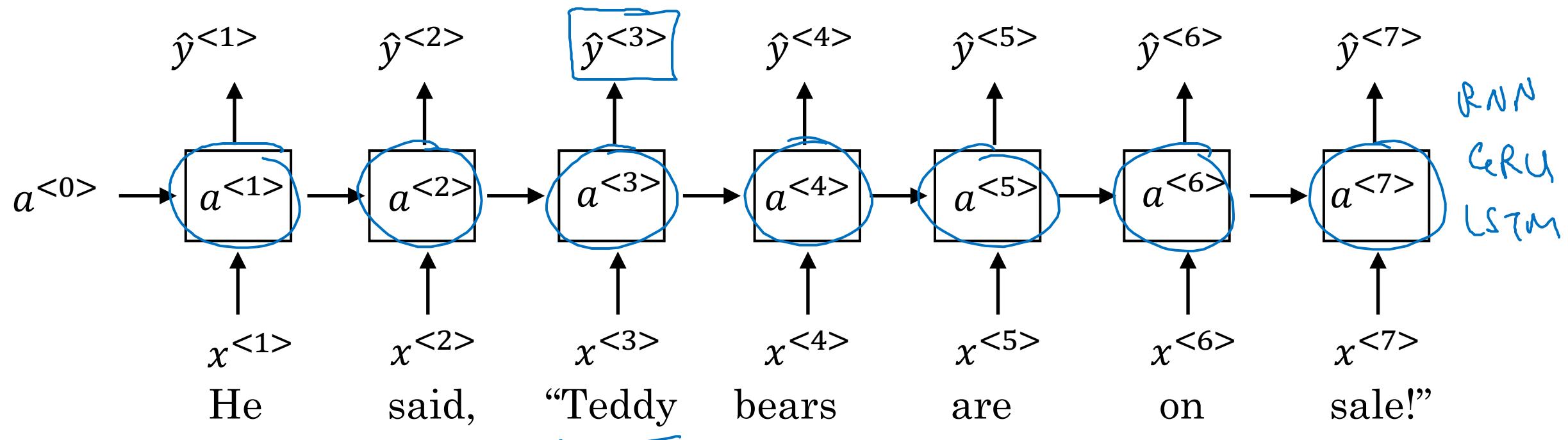
Recurrent Neural Networks

Bidirectional RNN

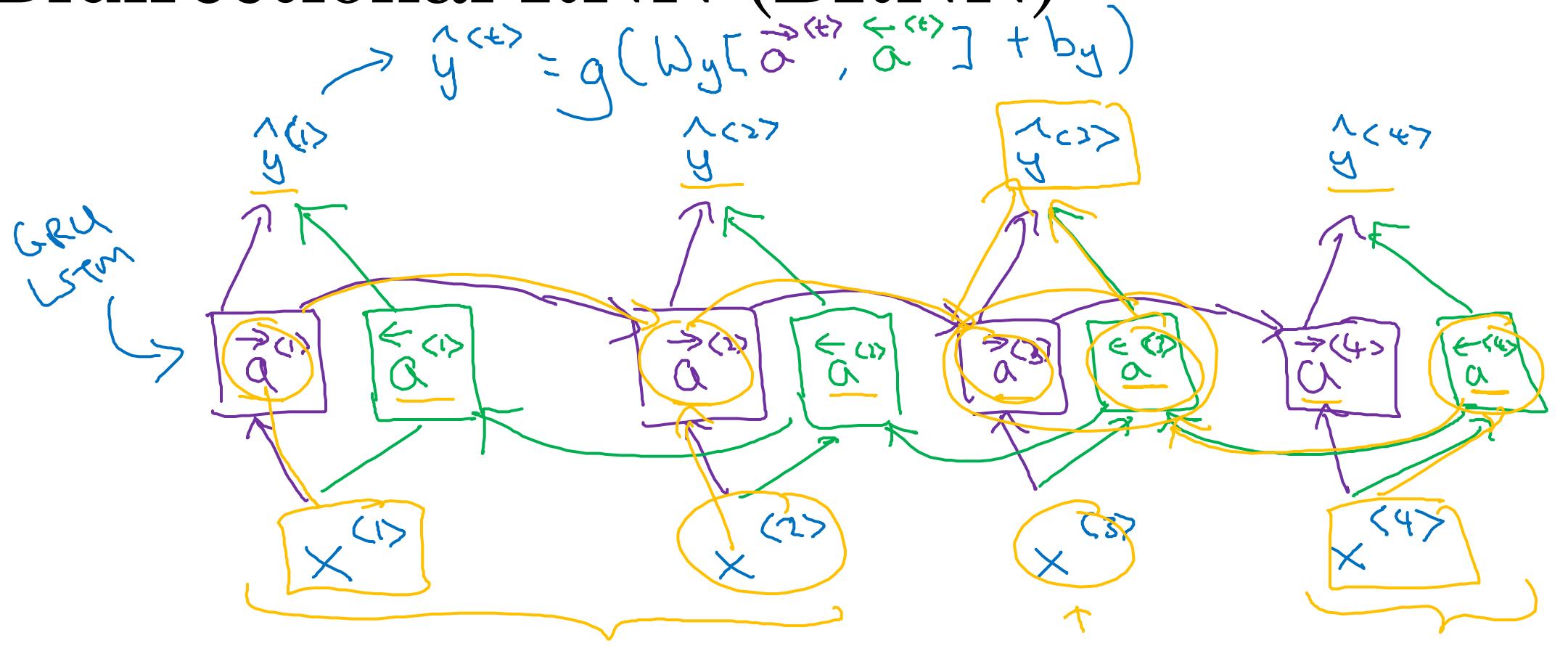
Getting information from the future

He said, “Teddy bears are on sale!”

He said, “Teddy Roosevelt was a great President!”



Bidirectional RNN (BRNN)



Acyclic graph

BRNN w/LSTM

He said,

"Teddy Roosevelt ..."



deeplearning.ai

Recurrent Neural Networks

Deep RNNs

Deep RNN example

