

Copyright Notice

These slides are distributed under the Creative Commons License.

[DeepLearning.AI](#) makes these slides available for educational purposes. You may not use or distribute these slides for commercial purposes. You may make copies of these slides and use or distribute them for educational purposes as long as you cite [DeepLearning.AI](#) as the source of the slides.

For the rest of the details of the license, see <https://creativecommons.org/licenses/by-sa/2.0/legalcode>



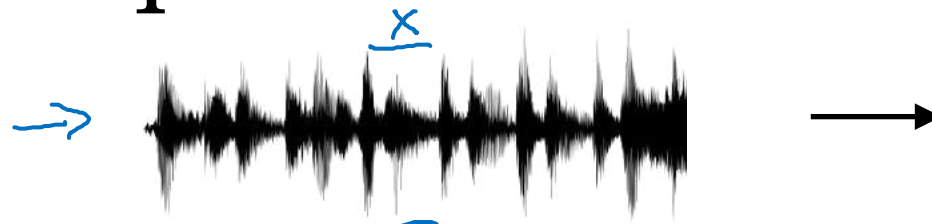
deeplearning.ai

Recurrent Neural Networks

Why sequence
models?

Examples of sequence data

Speech recognition



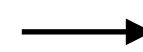
y
“The quick brown fox jumped
over the lazy dog.”

Music generation



Sentiment classification

“There is nothing to like
in this movie.”



DNA sequence analysis → AGCCCCTGTGAGGAACTAG



AGCCCCTGTGAGGAACTAG

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



deeplearning.ai

Recurrent Neural Networks

Notation

Motivating example

NLP

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

→ $x^{(1)}$ $x^{(2)}$ $x^{(3)}$... $x^{(t)}$... $x^{(9)}$

$T_x = 9$

→ $y:$

1 1 0 1 1 0 0 0 0
 $y^{(1)}$ $y^{(2)}$ $y^{(3)}$... $y^{(9)}$

$T_y = 9$

$x^{(i)(t)}$

$T_x^{(i)} = 9$

15

$y^{(i)(t)}$
 ↑

$T_y^{(i)}$

Representing words

$x^{<t>}$

(x, y)

$x \rightarrow y$

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

$x^{<1>}$

$x^{<2>}$

$x^{<3>}$

...

$x^{<7>}$

$x^{<9>}$

Vocabulary

a	1
aaron	2
...	...
and	367
...	...
harry	4075
...	...
potter	6830
...	...
zulu	10,000

<UNK> 10,000

One-hot

4075

6830

367

10,000

Representing words

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

$x^{<1>}$ $x^{<2>}$ $x^{<3>}$... $x^{<9>}$

And = 367

Invented = 4700

A = 1

New = 5976

Spell = 8376

Harry = 4075

Potter = 6830

Hermione = 4200

Gran... = 4000

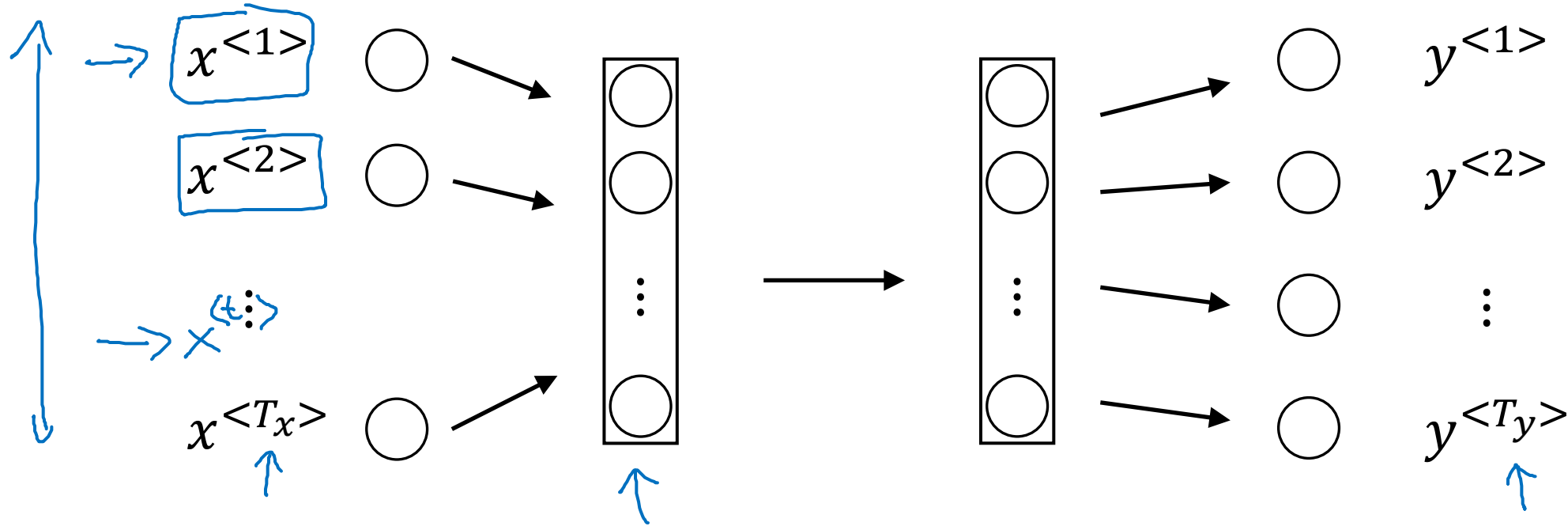


deeplearning.ai

Recurrent Neural Networks

Recurrent Neural Network Model

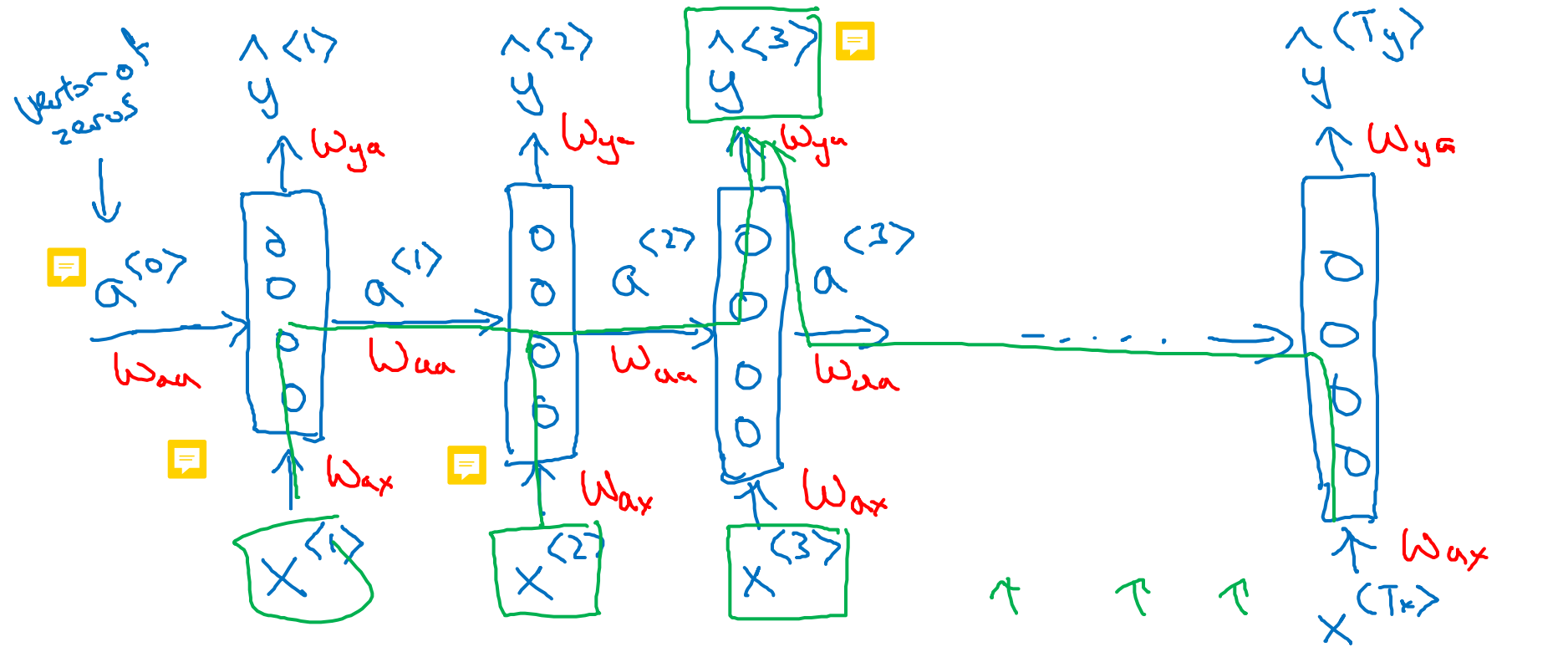
Why not a standard network?



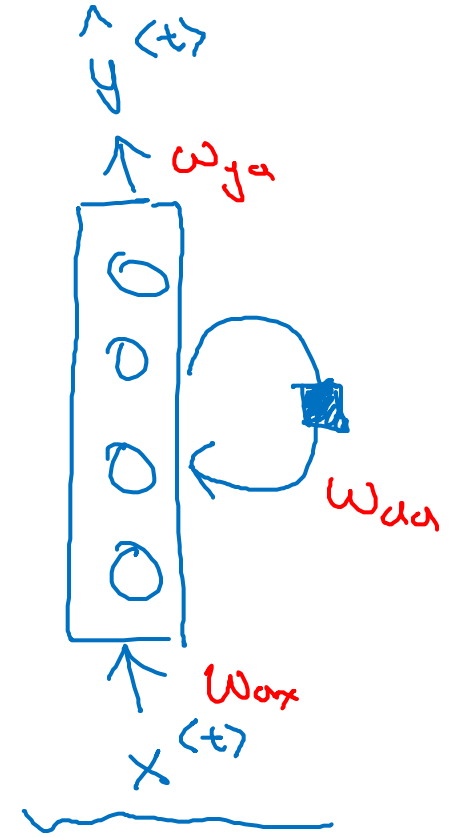
Problems:

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

Recurrent Neural Networks



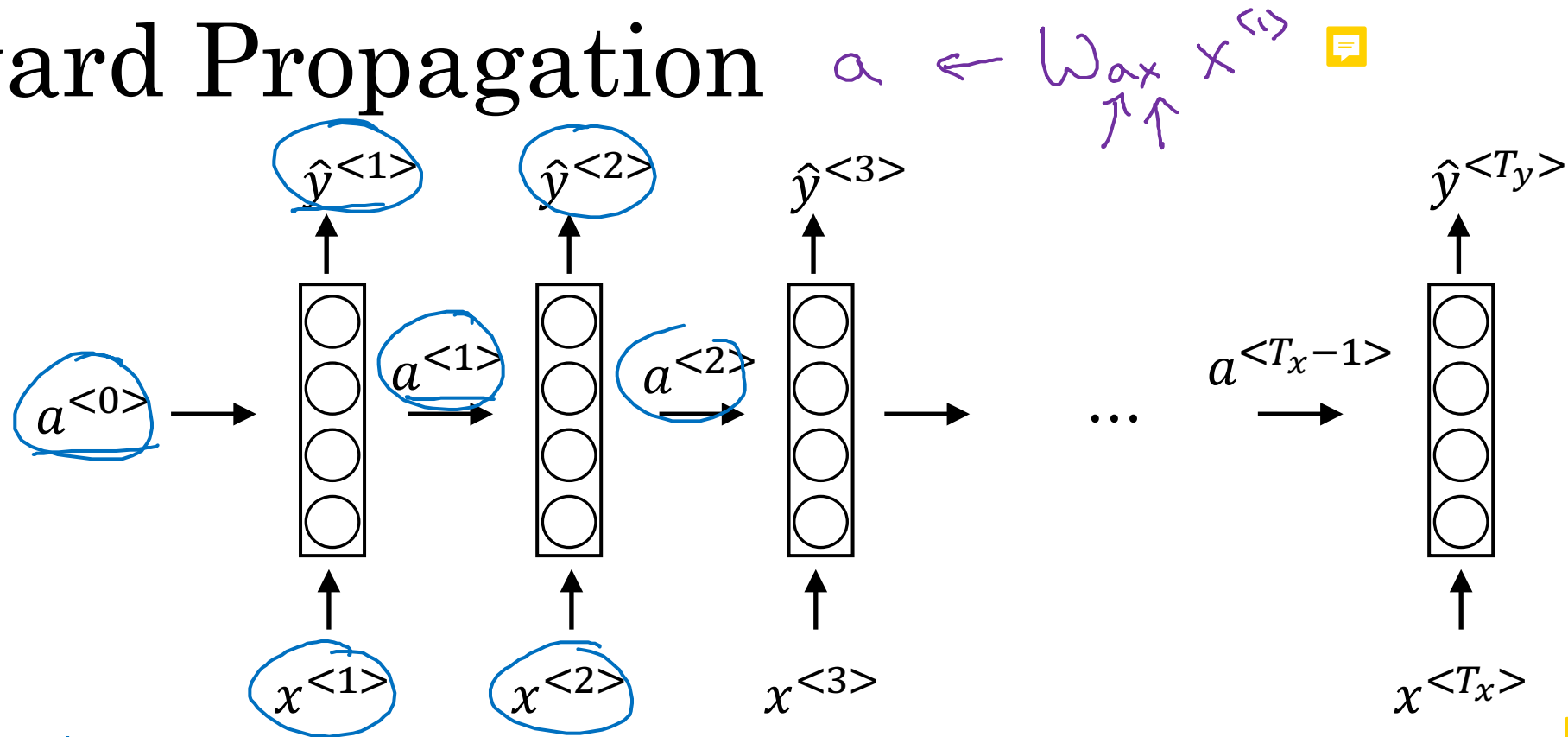
Bidirectional RNN (BRNN)



He said, "Teddy Roosevelt was a great President."

He said, "Teddy bears are on sale!"

Forward Propagation



$$a^{<0>} = \vec{0}.$$

$$\underline{a}^{<1>} = g_1(W_{aa} a^{<0>} + \underline{W_{ax}} x^{<1>} + b_a) \leftarrow \underline{\tanh / \text{Relu}}$$

$$\underline{\hat{y}}^{<1>} = g_2(\underline{W_{ya}} \underline{a}^{<1>} + b_y) \leftarrow \text{Sigmoid}$$

$$\begin{aligned} a^{<t>} &= g(W_{aa} a^{<t-1>} + W_{ax} x^{<t>} + b_a) \\ \hat{y}^{<t>} &= g(W_{ya} a^{<t>} + b_y) \end{aligned}$$

Simplified RNN notation

$$a^{<t>} = g(\underbrace{W_{aa} a^{<t-1>}}_{\substack{\uparrow \\ (100, 100)}} + \underbrace{W_{ax} x^{<t>}}_{\substack{\uparrow \\ (100, 10,000)}} + b_a)$$

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

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

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

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

(100, 10100)

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

100
10000
10100

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

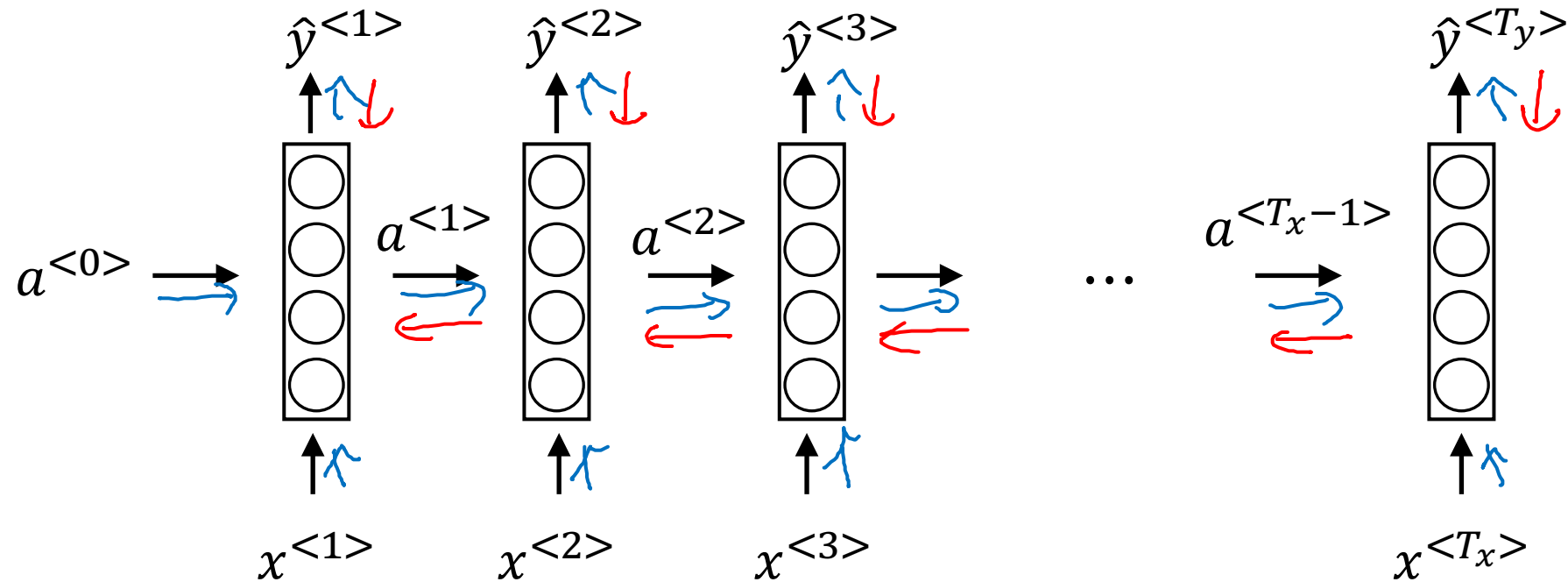


deeplearning.ai

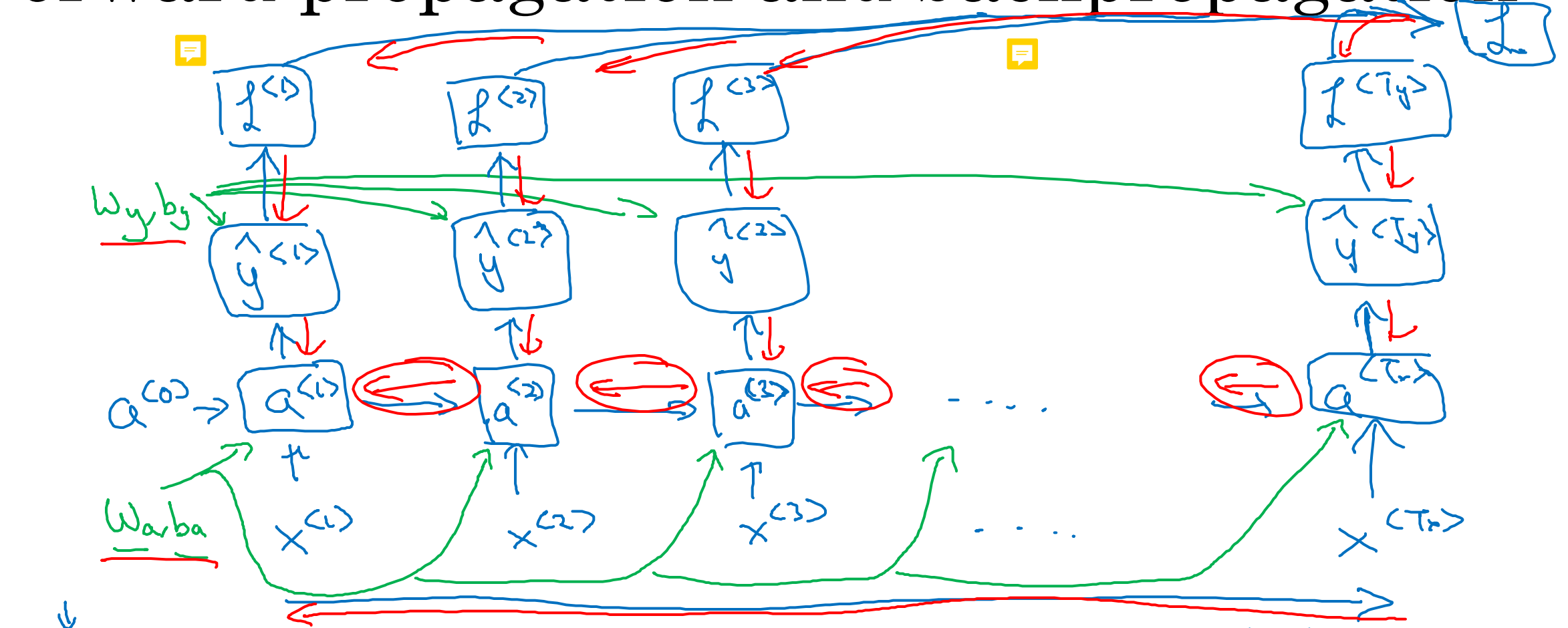
Recurrent Neural Networks

Backpropagation
through time

Forward propagation and backpropagation



Forward propagation and backpropagation



$$\mathcal{L}^{(t)}(\hat{y}^{(t)}, y^{(t)}) = -y^{(t)} \log \hat{y}^{(t)} - (1 - y^{(t)}) \log (1 - \hat{y}^{(t)})$$

$$\mathcal{L}(\hat{y}, y) = \sum_{t=1}^{T_y} \mathcal{L}^{(t)}(\hat{y}^{(t)}, y^{(t)})$$

Backpropagation through time



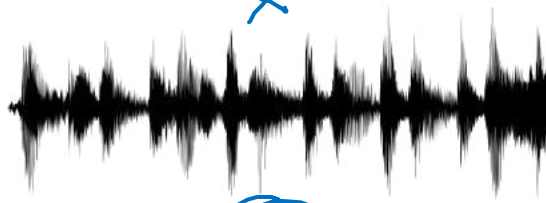
deeplearning.ai

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

AGCCCCTGTGAGGAACTAG



AG**CCCCTGTGAGGAACT**AG

Machine translation

Voulez-vous chanter avec moi?



Do you want to sing with me?

Video activity recognition



Running

Name entity recognition

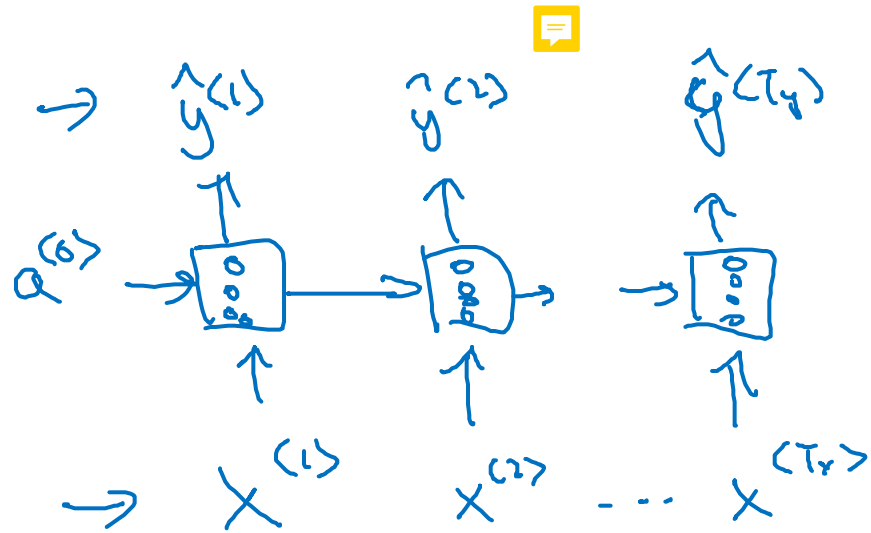
Yesterday, Harry Potter met Hermione Granger.



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

Examples of RNN architectures

$$T_x = T_y$$

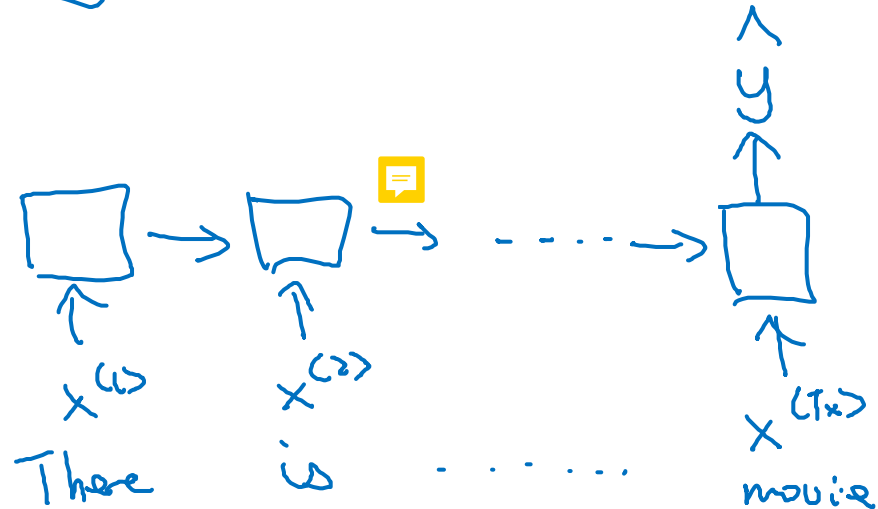


Many-to-many

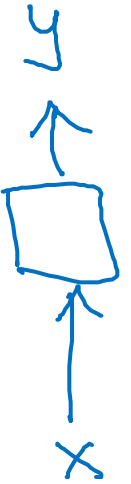
Sentiment classification-

$x = \text{text}$

$y = 0/1 \quad 1 \dots 5$

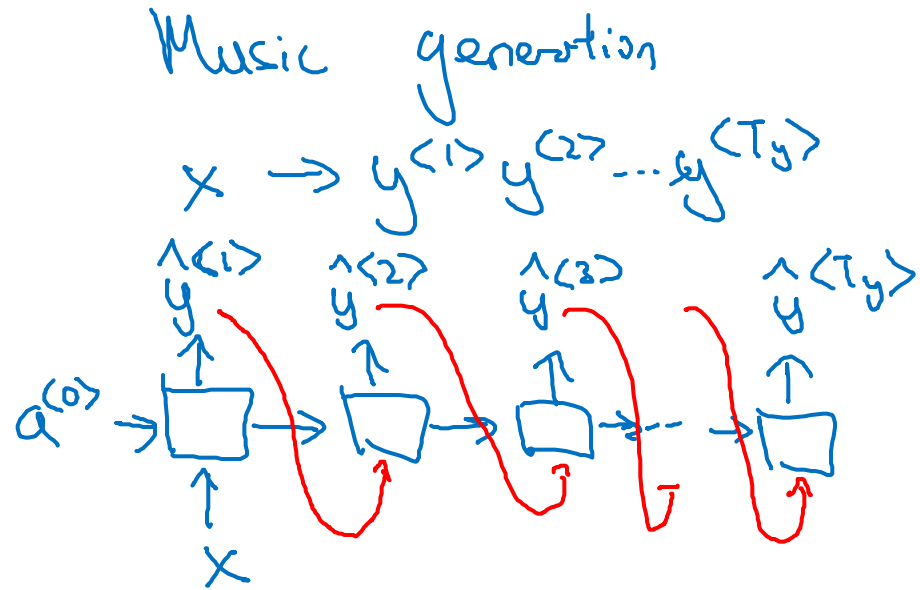


Many-to-one



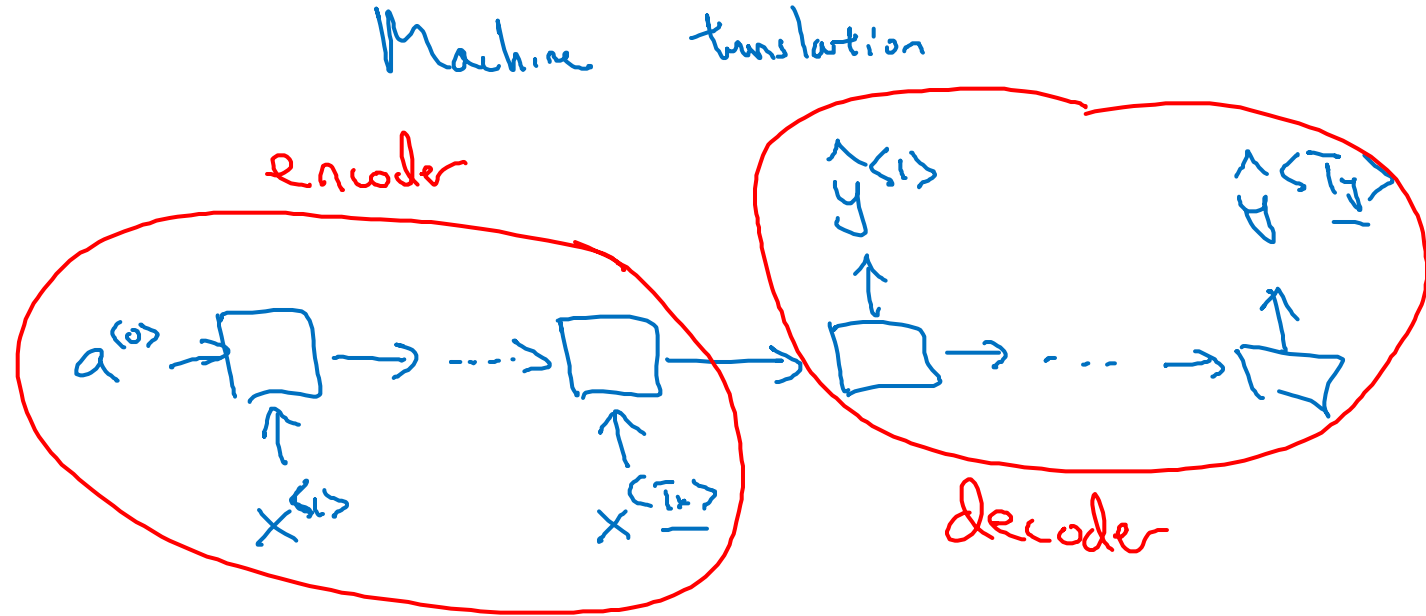
One-to-one

Examples of RNN architectures



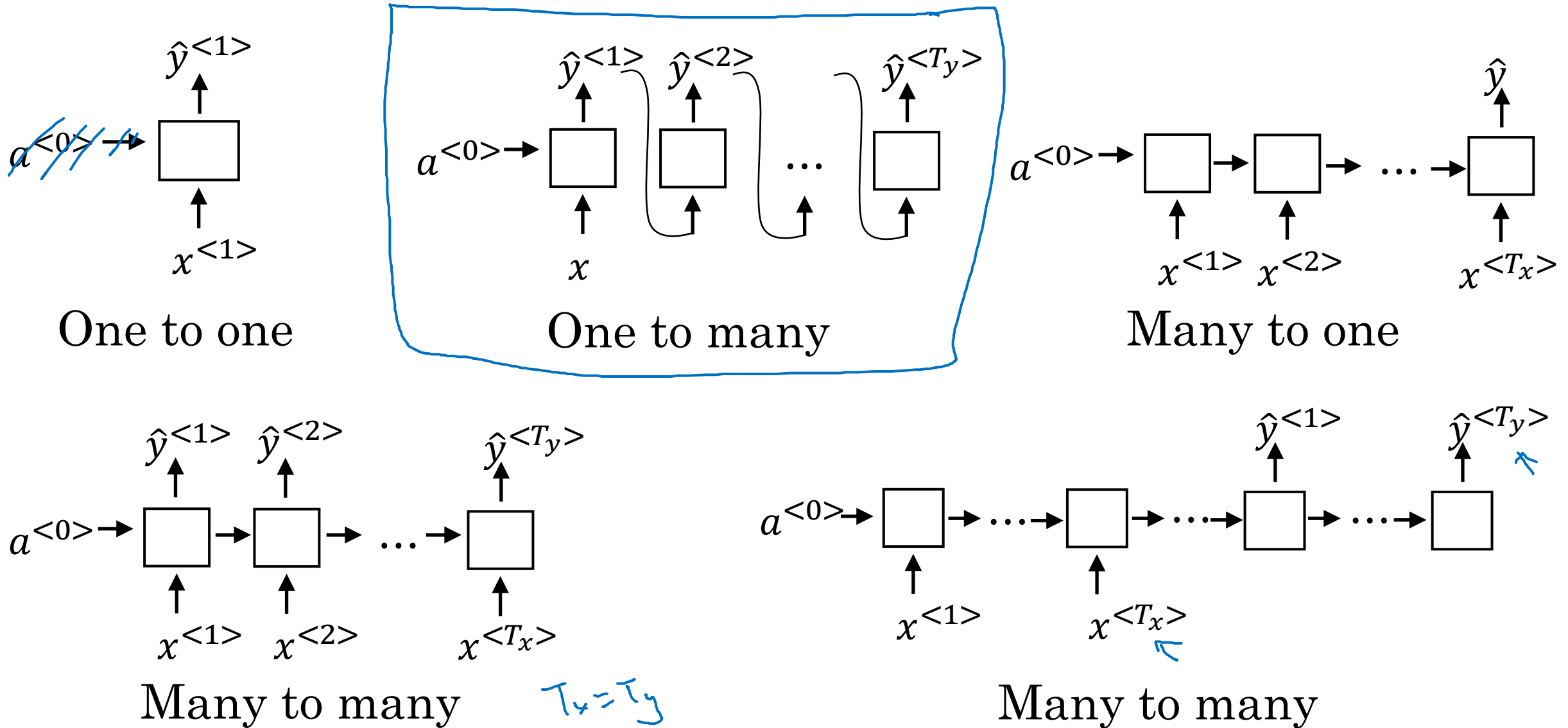
One-to-many

$$x = \phi$$



Many-to-many

Summary of RNN types





deeplearning.ai

Recurrent Neural Networks

Language model and
sequence generation

What is language modelling?

Speech recognition

The apple and pair salad.

→ The apple and pear salad.

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

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

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

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

Language modelling with an RNN

Training set: large corpus of english text.

Tokenize

Cats average 15 hours of sleep a day. \downarrow $\langle \text{EOS} \rangle$

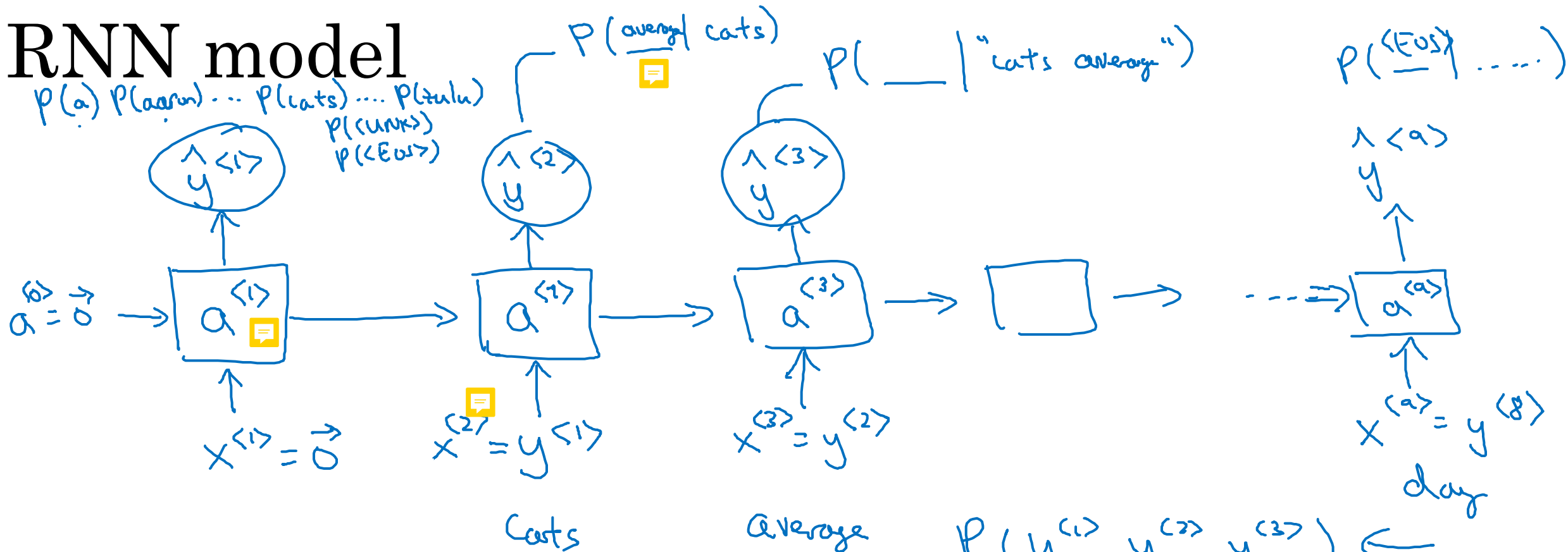
$y^{(1)}$ $y^{(2)}$ $y^{(3)}$... $y^{(8)}$ $y^{(9)}$
 $x^{(t)} = y^{(t-1)}$

The Egyptian ~~Mau~~ is a breed of cat. $\langle \text{EOS} \rangle$

$\langle \text{UNK} \rangle$

10,000

RNN model



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

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

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

$$P(y^{(1)}, y^{(2)}, y^{(3)}) \leftarrow$$

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



deeplearning.ai

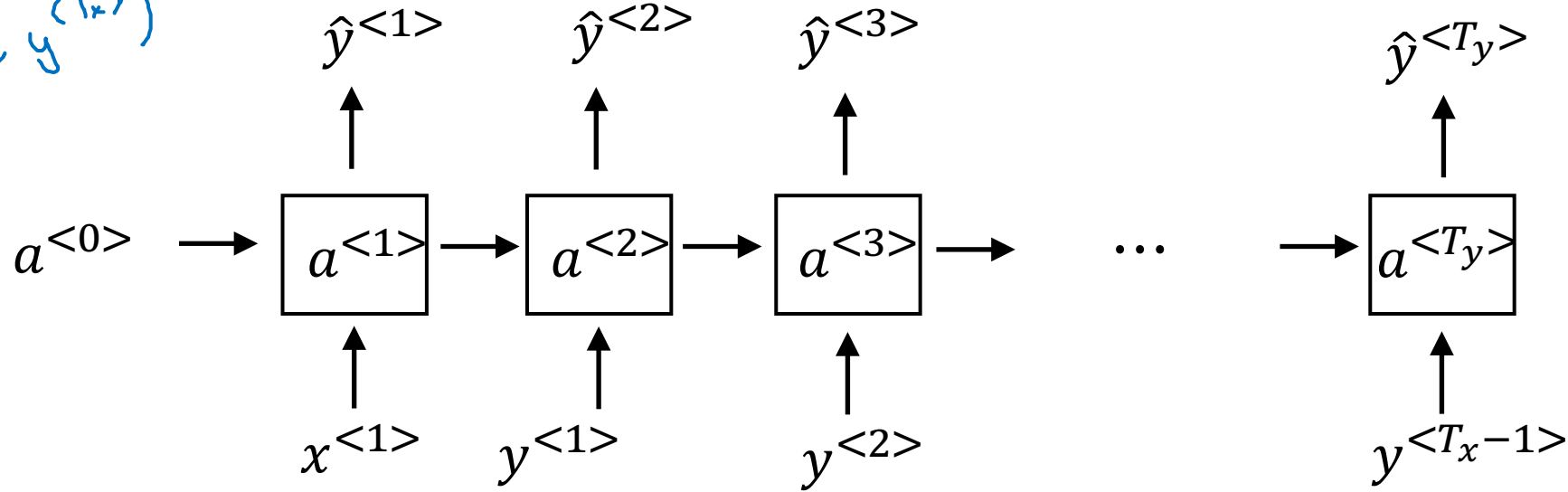
Recurrent Neural Networks

Sampling novel
sequences

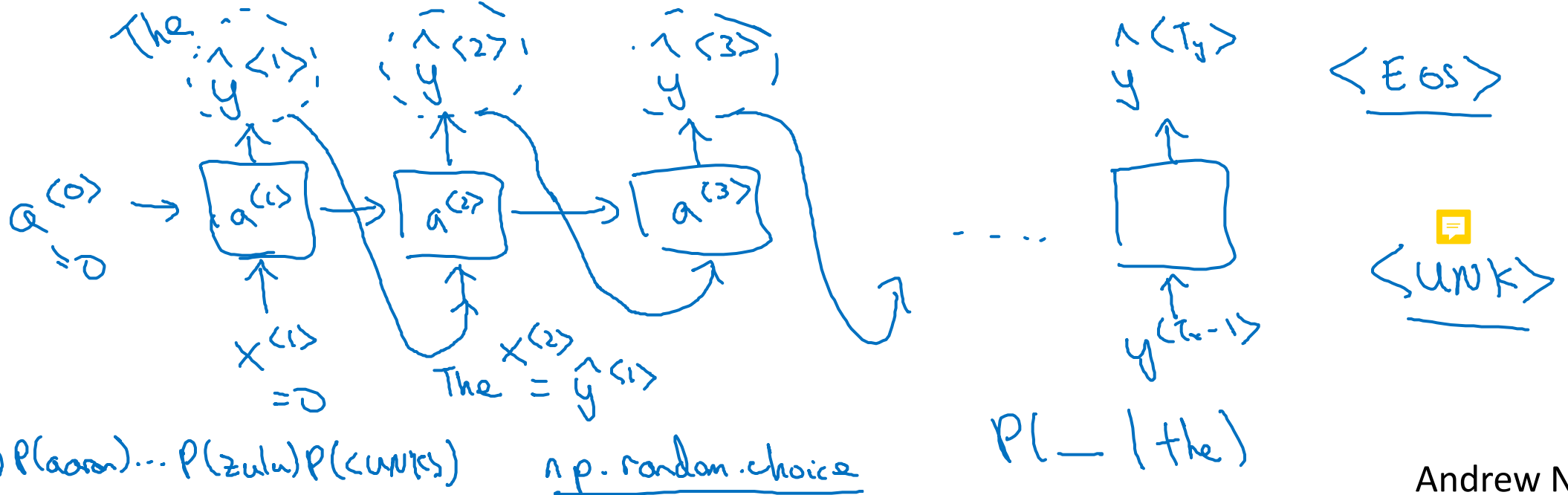
Sampling a sequence from a trained RNN

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

Training:



Sampling:



<Eos>

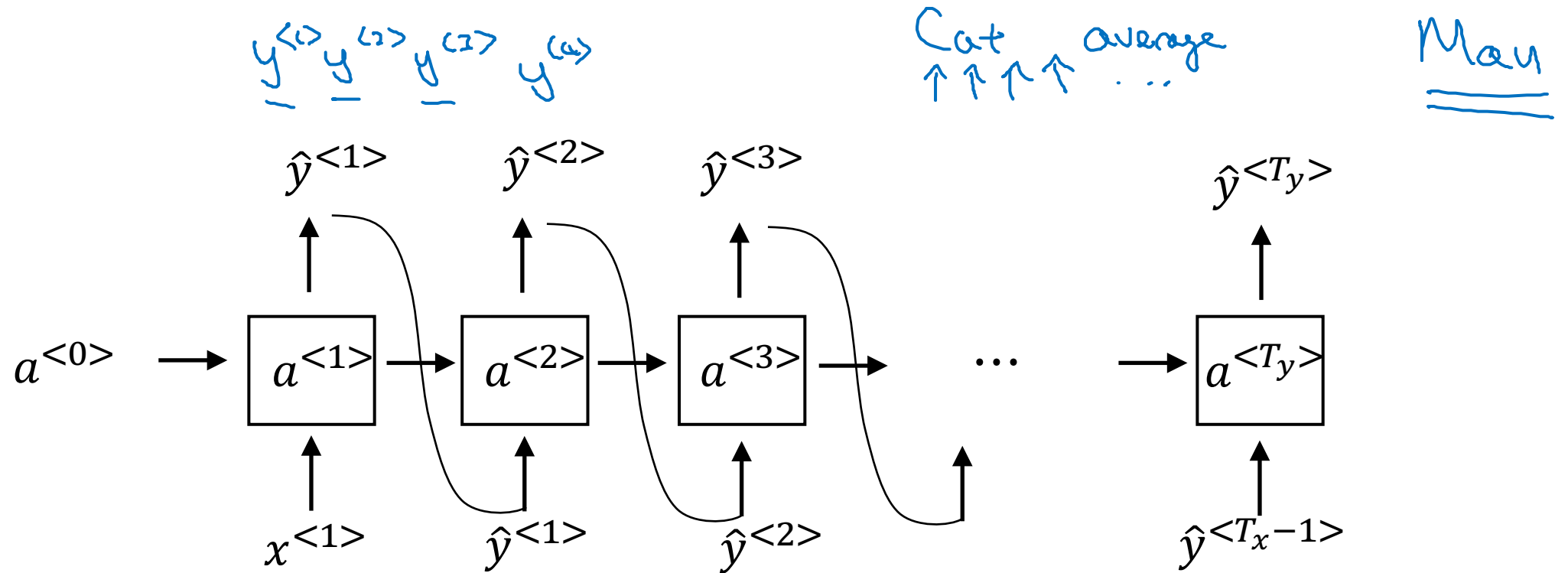
<UNK>

$P(_ | \text{the})$

Character-level language model 🗨️

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

→ Vocabulary = [a, b, c, ..., z, \backslash , ., , , ;, 0, ..., 9, A, ..., Z]



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 besser be my love to me see sabl's.

For whose are ruse of mine eyes heaves.

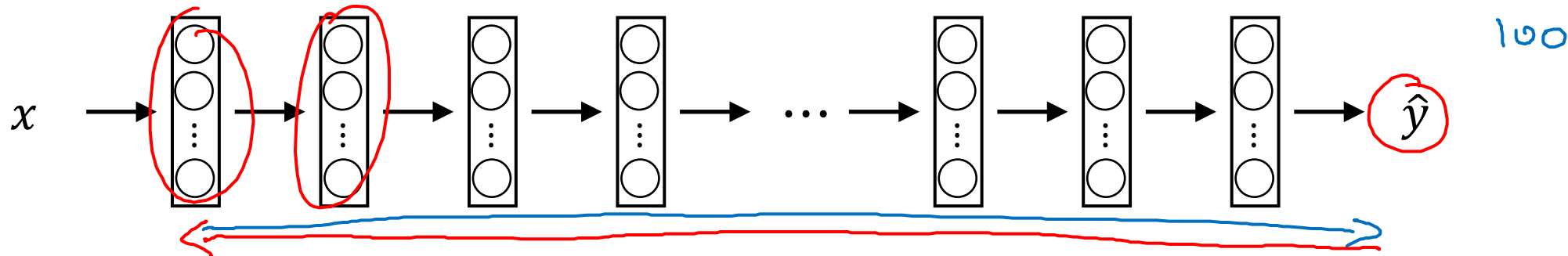
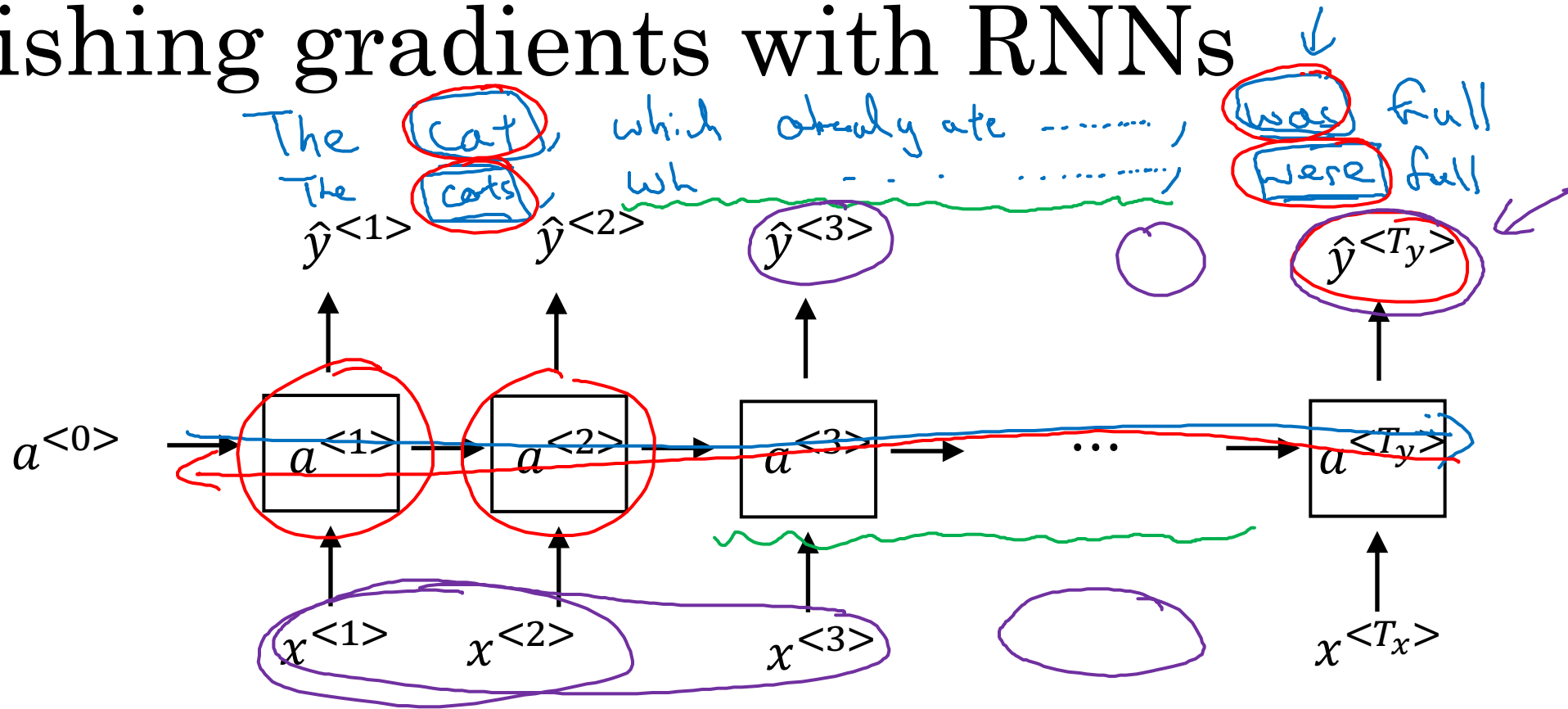


deeplearning.ai

Recurrent Neural Networks

 Vanishing gradients
with RNNs

Vanishing gradients with RNNs



Exploding gradients.

NaN

Gradient clipping

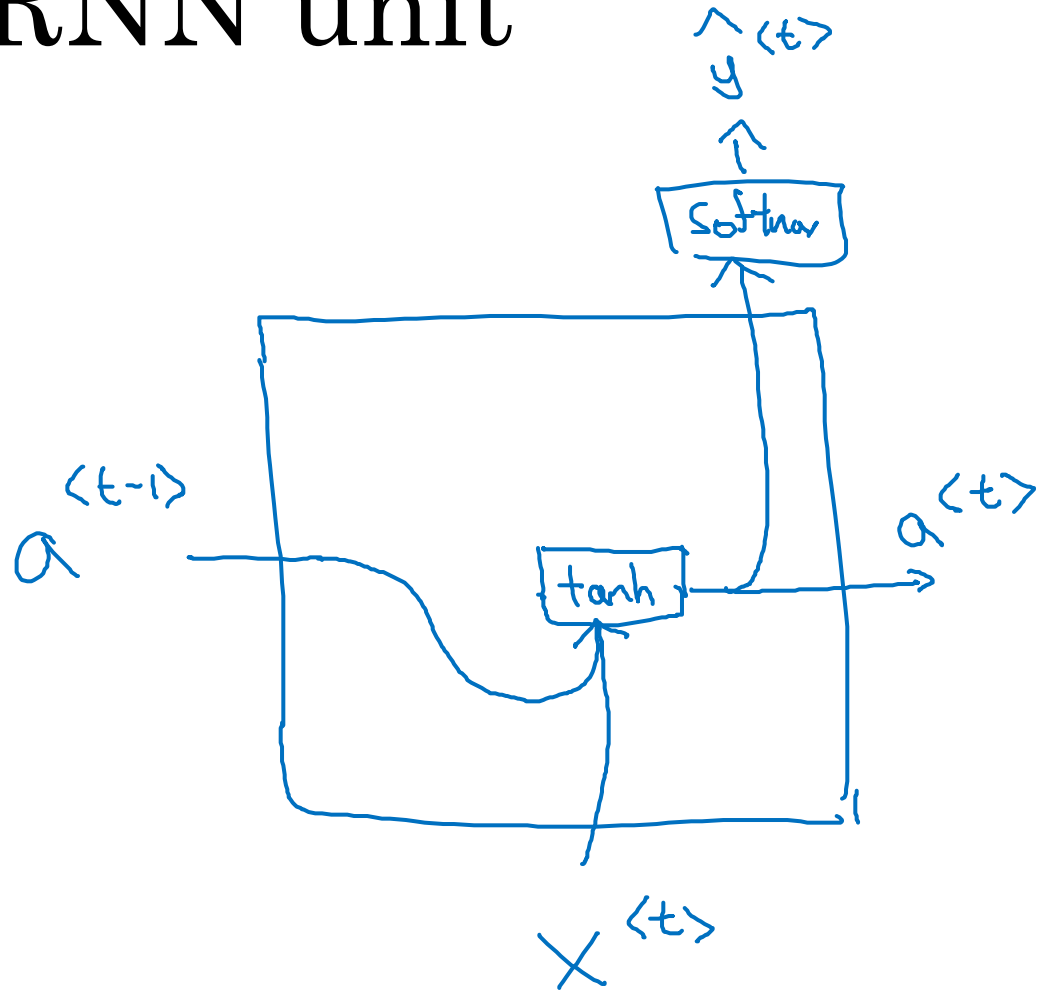


deeplearning.ai

Recurrent Neural Networks

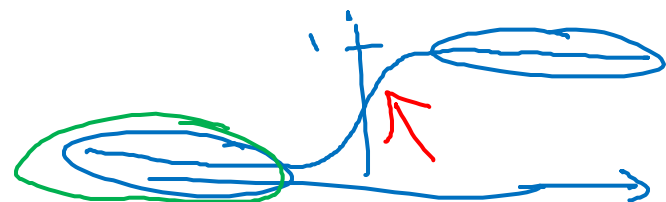
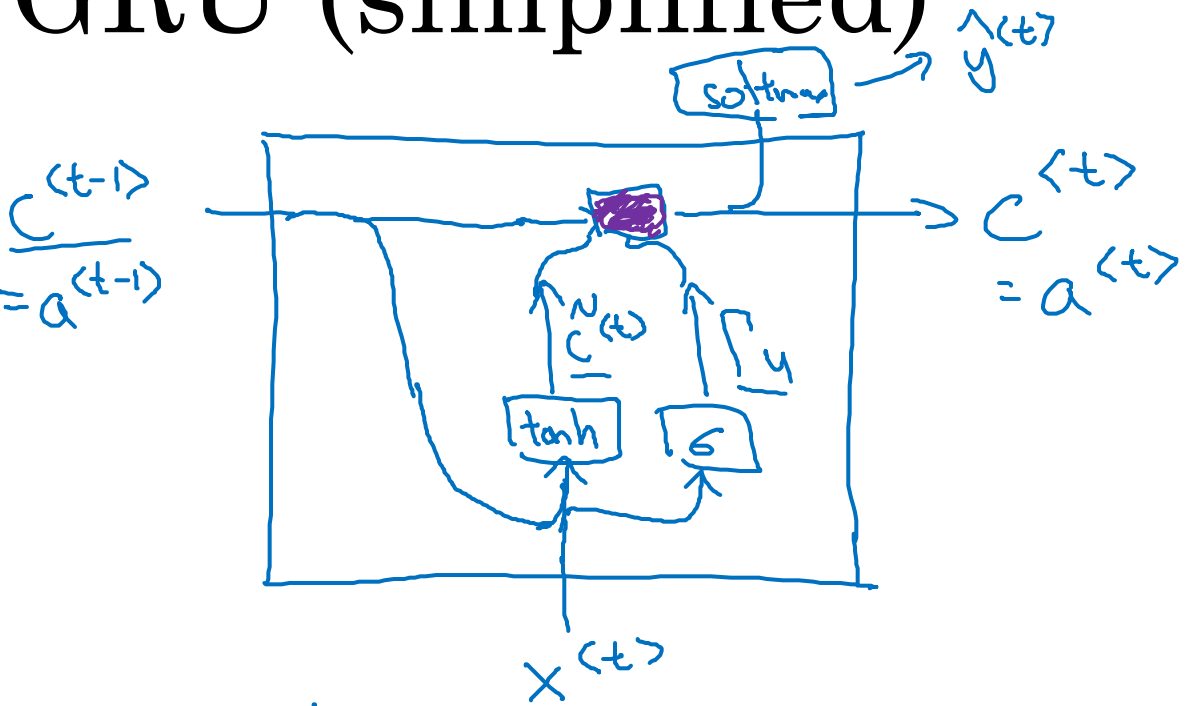
Gated Recurrent Unit (GRU)

RNN unit



$$\underline{a^{<t>}} = \overset{\substack{\text{tanh} \\ \downarrow}}{g}(\underbrace{W_a[a^{<t-1>}, x^{<t>}]}_{\uparrow} + b_a)$$

GRU (simplified)



C = memory cell

$$\rightarrow \boxed{C^{(t)}} = \underline{a}^{(t)}$$

$$\rightarrow \boxed{\tilde{C}^{(t)}} = \tanh(W_c [c^{(t-1)}, x^{(t)}] + b_c)$$

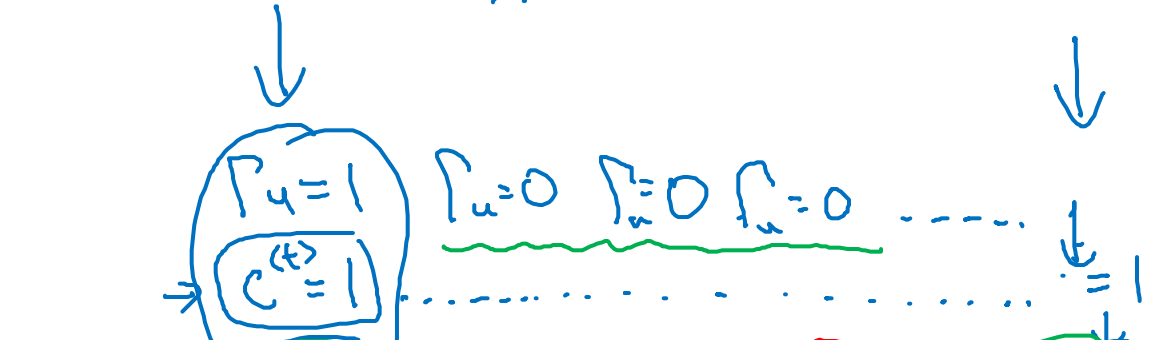
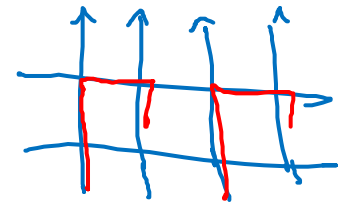
$$\rightarrow \boxed{\Gamma_u} = \sigma(W_u [c^{(t-1)}, x^{(t)}] + b_u)$$

$$\boxed{C^{(t)}} = \underbrace{\Gamma_u}_{\text{"update"}} * \tilde{C}^{(t)} + (1 - \Gamma_u) * \boxed{C^{(t-1)}}$$

element-wise

$$\Gamma_u = 0.000001$$

Gate



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]

Full GRU

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

Handwritten notes: A blue arrow points from $\tilde{c}^{<t>}$ to the input $x^{<t>}$. Above the equation, a red bracket indicates $\Gamma_r * c^{<t-1>}$ is the previous hidden state.

$$\begin{cases} \Gamma_u = \sigma(W_u[c^{<t-1>}, x^{<t>}] + b_u) \\ \Gamma_r = \sigma(W_r[c^{<t-1>}, x^{<t>}] + b_r) \end{cases}$$

LSTM

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

The cat, which ate already, was full.



deeplearning.ai

Recurrent Neural Networks

LSTM (long short
term memory) unit

GRU and LSTM

GRU

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

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

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

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

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

\uparrow
 $\underline{\Gamma}_f$

LSTM

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

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

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

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

$$\underline{c}^{<t>} = \underline{\Gamma}_u * \tilde{c}^{<t>} + \underline{\Gamma}_f * \underline{c}^{<t-1>}$$

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

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

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

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

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

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

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

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

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

$$\Gamma_o * \tanh c^{<t>}$$

LSTM in pictures



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

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

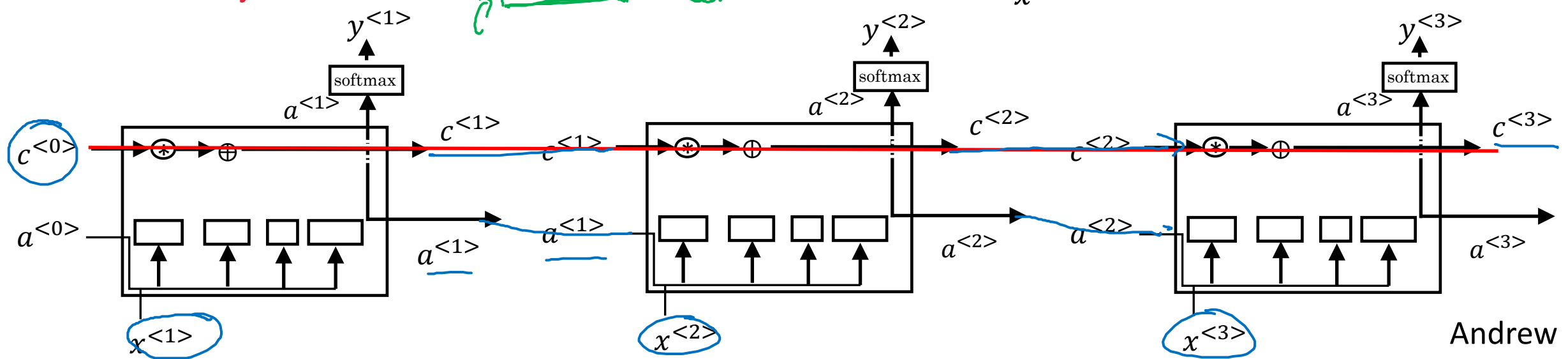
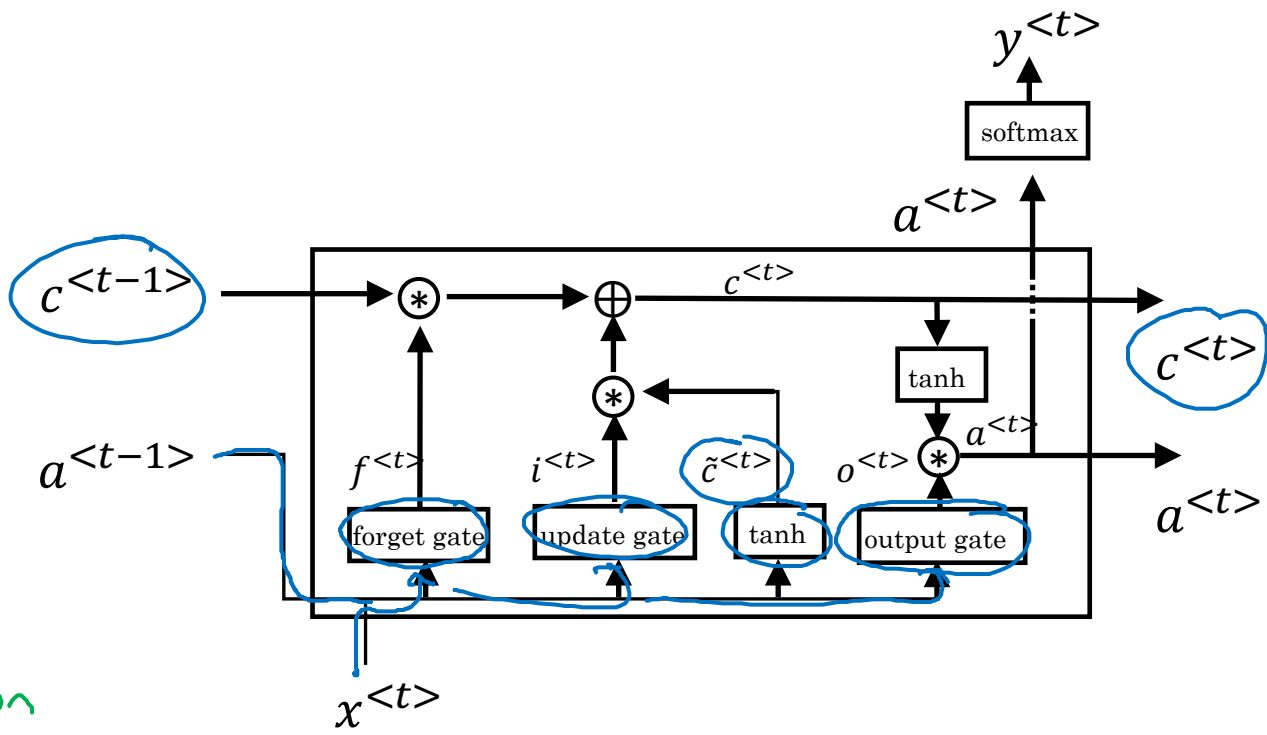
$$\Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f)$$

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

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

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

peephole connection
 $c^{<t-1>}$





deeplearning.ai

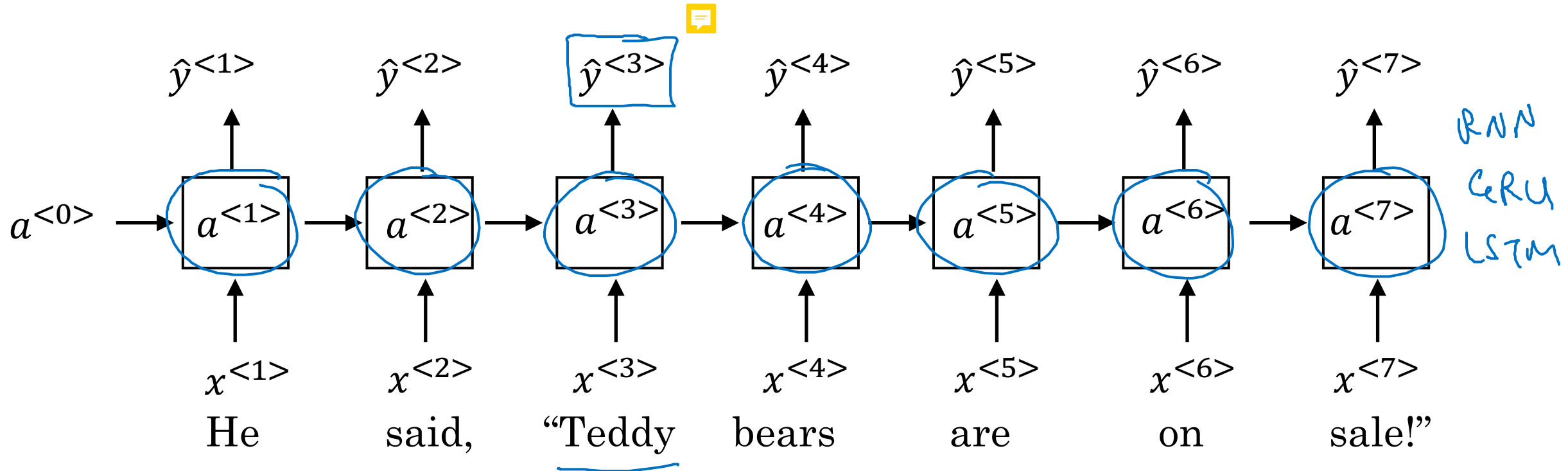
Recurrent Neural Networks

Bidirectional RNN

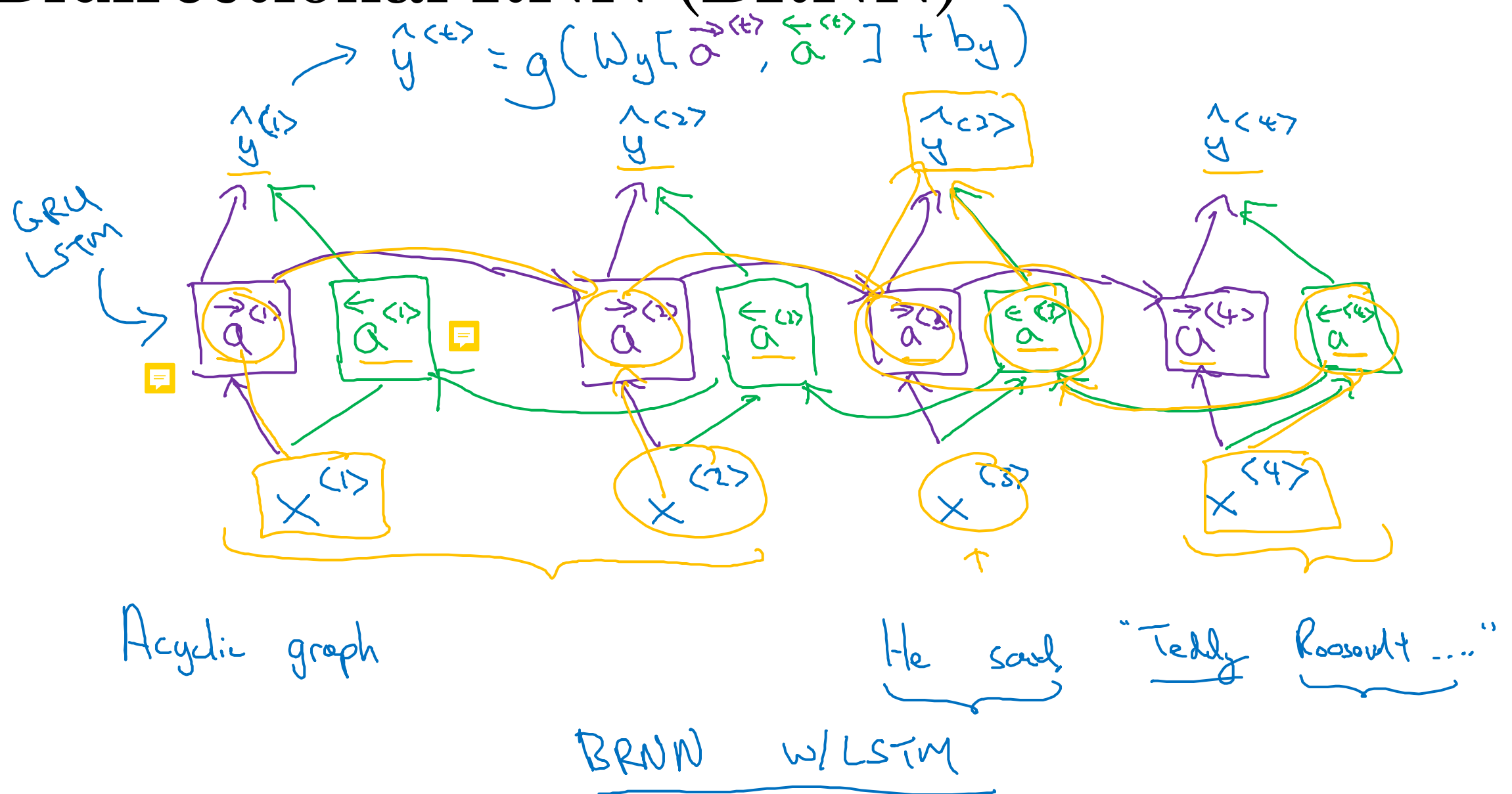
Getting information from the future

He said, “Teddy bears are on sale!”

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



Bidirectional RNN (BRNN)



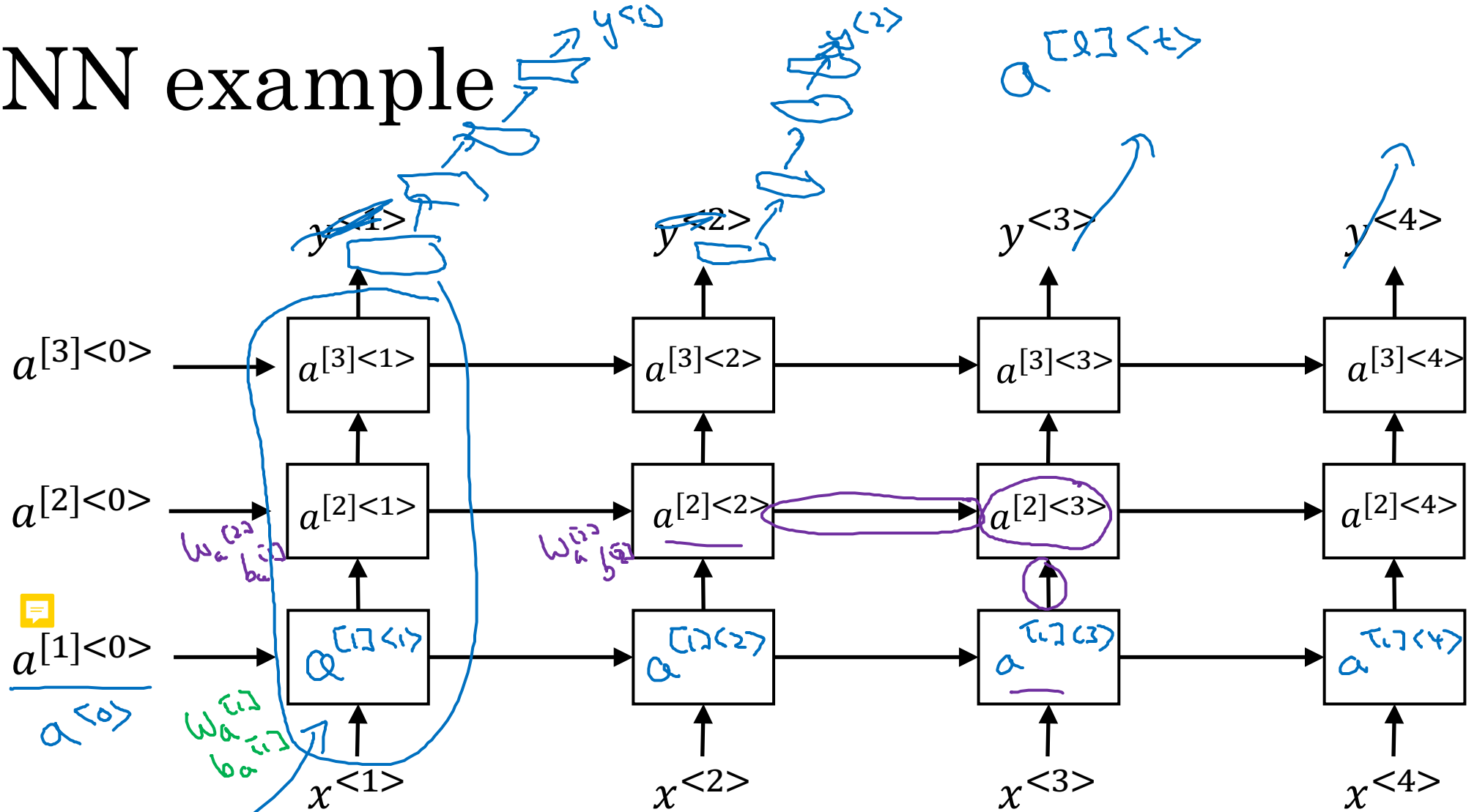
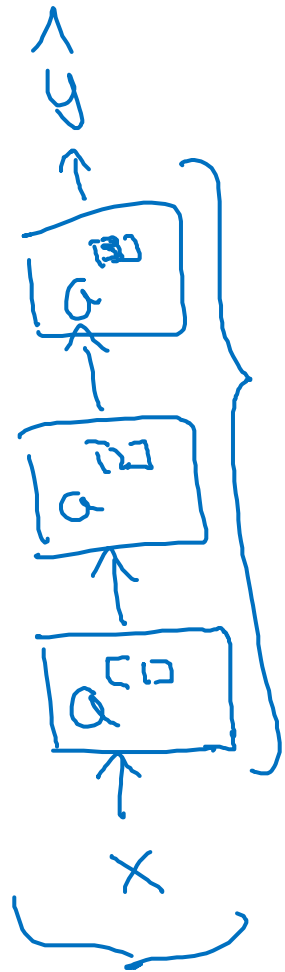


deeplearning.ai

Recurrent Neural Networks

Deep RNNs

Deep RNN example



RNN
GRU
LSTM

BRNN

$$a^{[2]} \langle 3 \rangle = g(W_a^{[2]} [a^{[2]} \langle 2 \rangle, a^{[1]} \langle 3 \rangle] + b_a^{[2]})$$