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

	x	y
Speech recognition		"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

$y^{(i)<t>}$ $T_y^{(i)}$

Andrew Ng

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

$$\begin{bmatrix} \text{a} \\ \text{aaron} \\ \vdots \\ \text{and} \\ \vdots \\ \text{harry} \\ \vdots \\ \text{potter} \\ \vdots \\ \text{zulu} \end{bmatrix} \begin{array}{l} 1 \\ 2 \\ \vdots \\ 367 \\ \vdots \\ 4075 \\ \vdots \\ 6830 \\ \vdots \\ 10{,}000 \end{array}$$

<UNK> 10,000

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{array}{l} \\ \\ \\ \\ \leftarrow 4075 \\ \\ \\ \\ \end{array} \quad \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \begin{array}{l} \\ \\ \\ \leftarrow 6830 \\ \\ \\ \end{array} \quad \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \begin{array}{l} \\ \\ \leftarrow 367 \\ \\ \\ \end{array} \quad \updownarrow 10{,}000 \quad \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

One-hot

Representing words

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

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

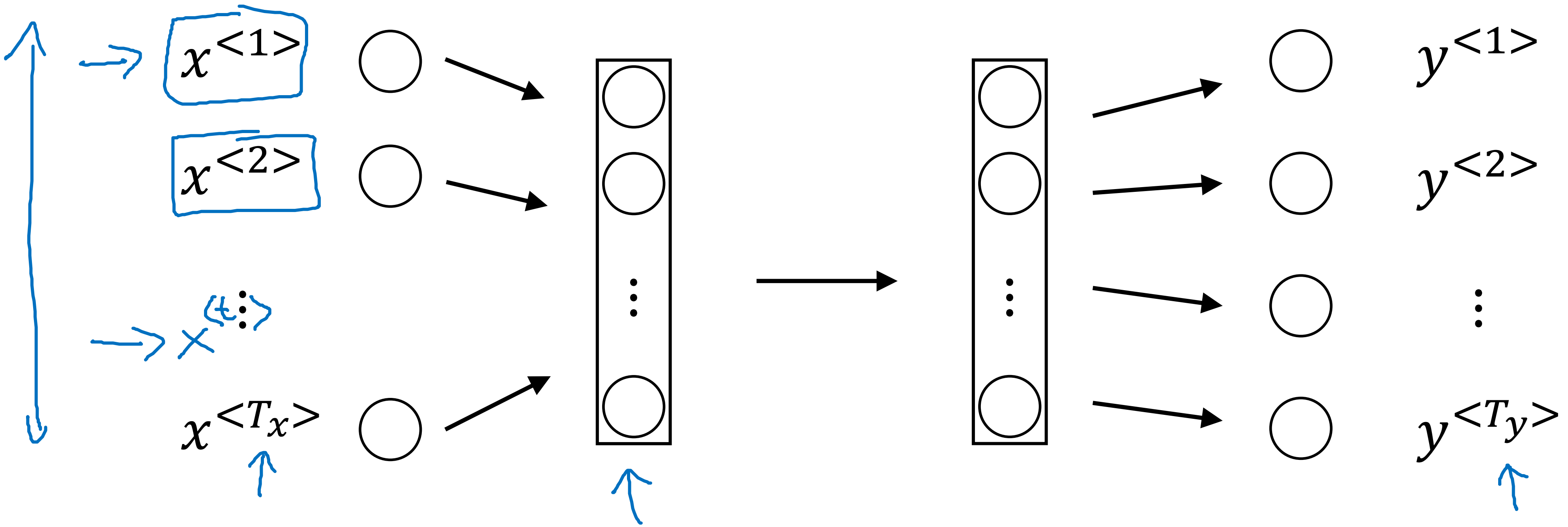
And = 367
Invented = 4700
A = 1
New = 5976
Spell = 8376
Harry = 4075
Potter = 6830
Hermione = 4200
Gran… = 4000

deeplearning.ai

Recurrent Neural Networks

Recurrent Neural Network Model

Why not a standard network?

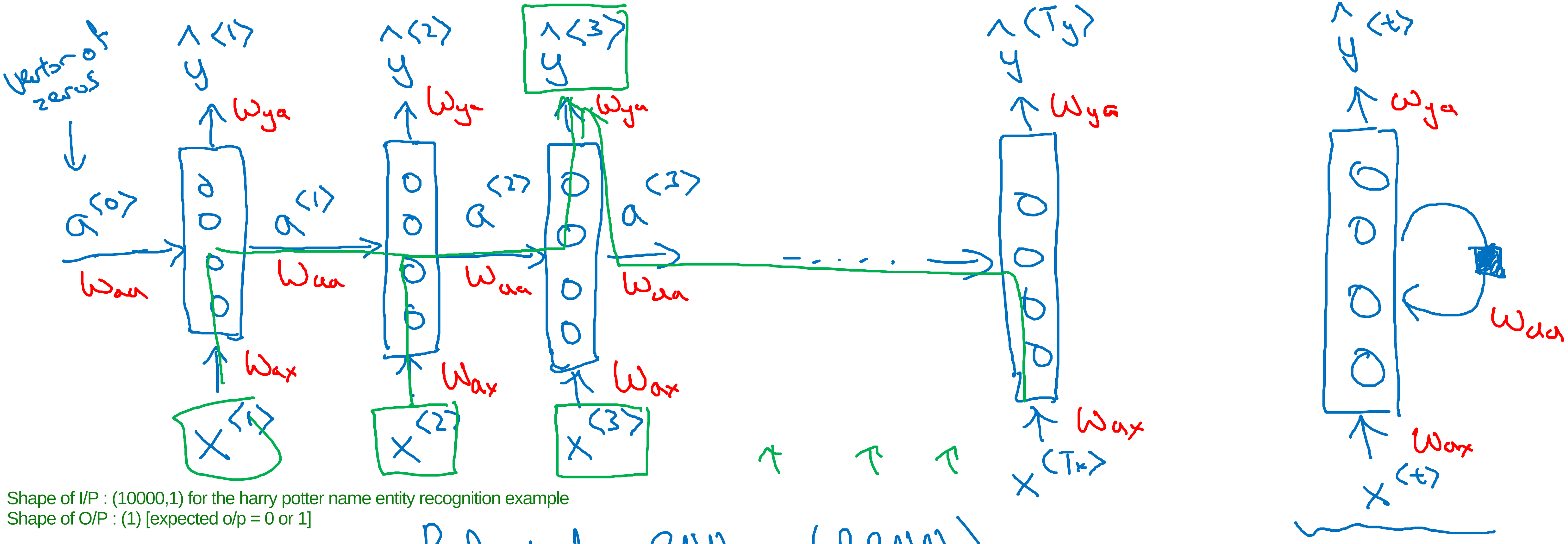


Problems:

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

Recurrent Neural Networks

$T_x = T_y$



Shape of I/P : (10000,1) for the harry potter name entity recognition example
Shape of O/P : (1) [expected o/p = 0 or 1]

[Suppose 1 hidden layer with 128 neurons]
Shape of W_ax - (128, 10000)
Shape of W_aa - (128,128) Since weight is (number of neurons in current layer, number of inputs)

Bidirectional RNN (BRNN)

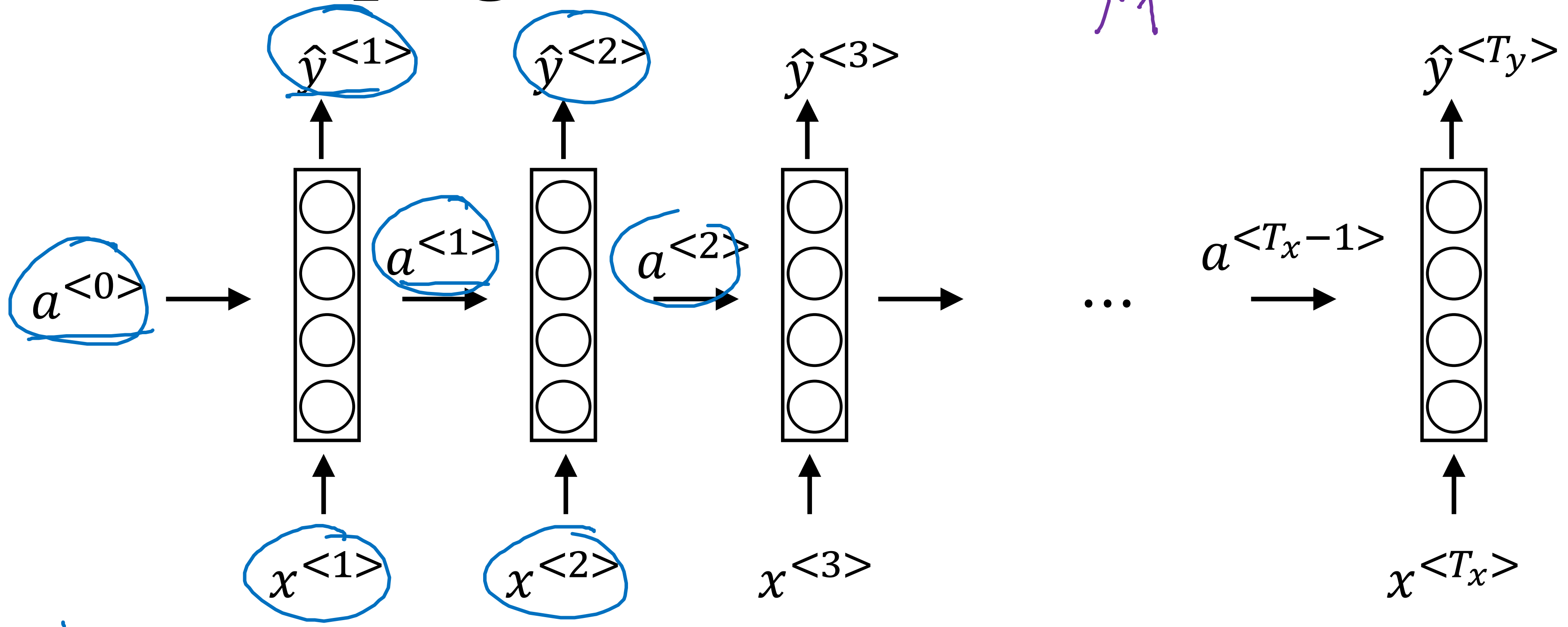
He said, "Teddy Roosevelt was a great President."

He said, "Teddy bears are on sale!"

Andrew Ng

Forward Propagation

$a \leftarrow W_{ax} x^{<1>}$



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

$a^{<1>} = g_1(W_{aa} a^{<0>} + W_{ax} x^{<1>} + b_a) \leftarrow$ tanh | ReLU

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

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

Simplified RNN notation

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

W_{aa}: (100,100) 100

W_{ax}: (100,10,000) 10,000

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

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

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

$$\underbrace{\begin{bmatrix} W_{aa} & \vdots & W_{ax} \end{bmatrix}}_{100 \qquad 10\,000} \Big\updownarrow 100 = W_a$$

W_a: (100, 10100)

$$[a^{(t-1)}, x^{(t)}] = \begin{bmatrix} a^{(t-1)} \\ \hline x^{(t)} \end{bmatrix} \begin{array}{l} \updownarrow 100 \\ \updownarrow 10000 \end{array} \Big\updownarrow 10100$$

$$[W_{aa} \vdots W_{ax}] \begin{bmatrix} a^{(t-1)} \\ x^{(t)} \end{bmatrix} = \underline{W_{aa} a^{(t-1)} + W_{ax} x^{(t)}}$$

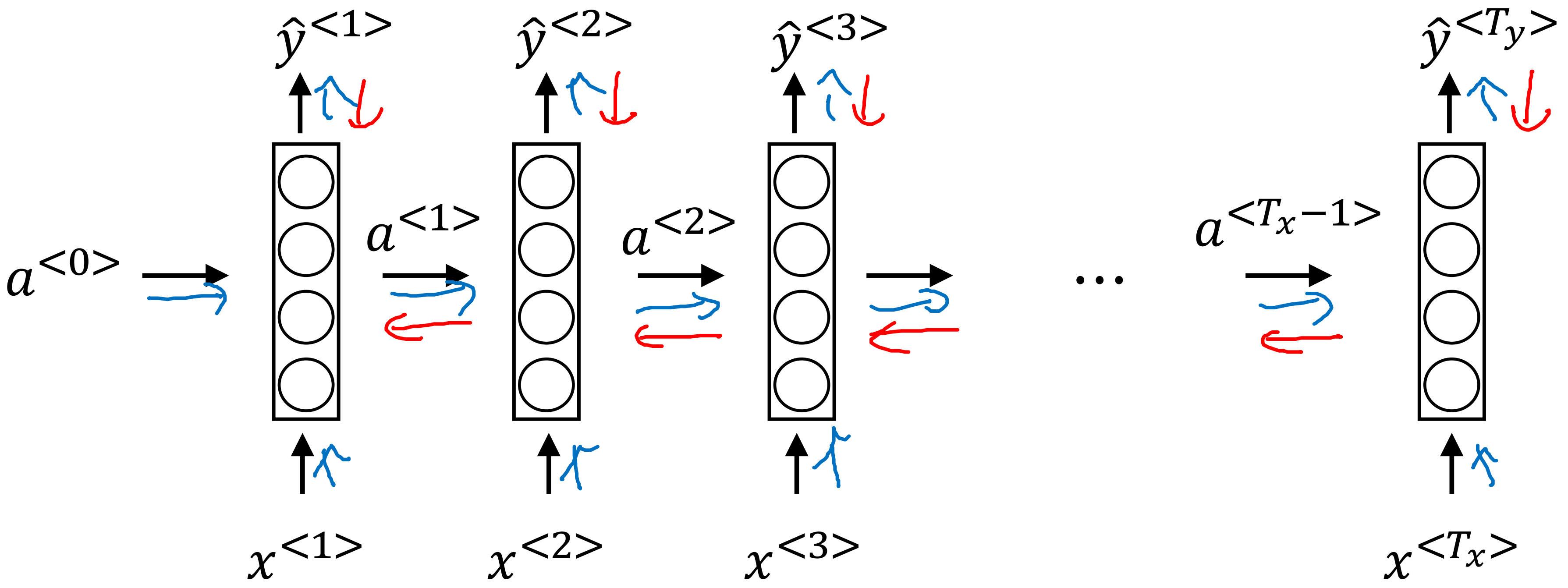
Andrew Ng

deeplearning.ai

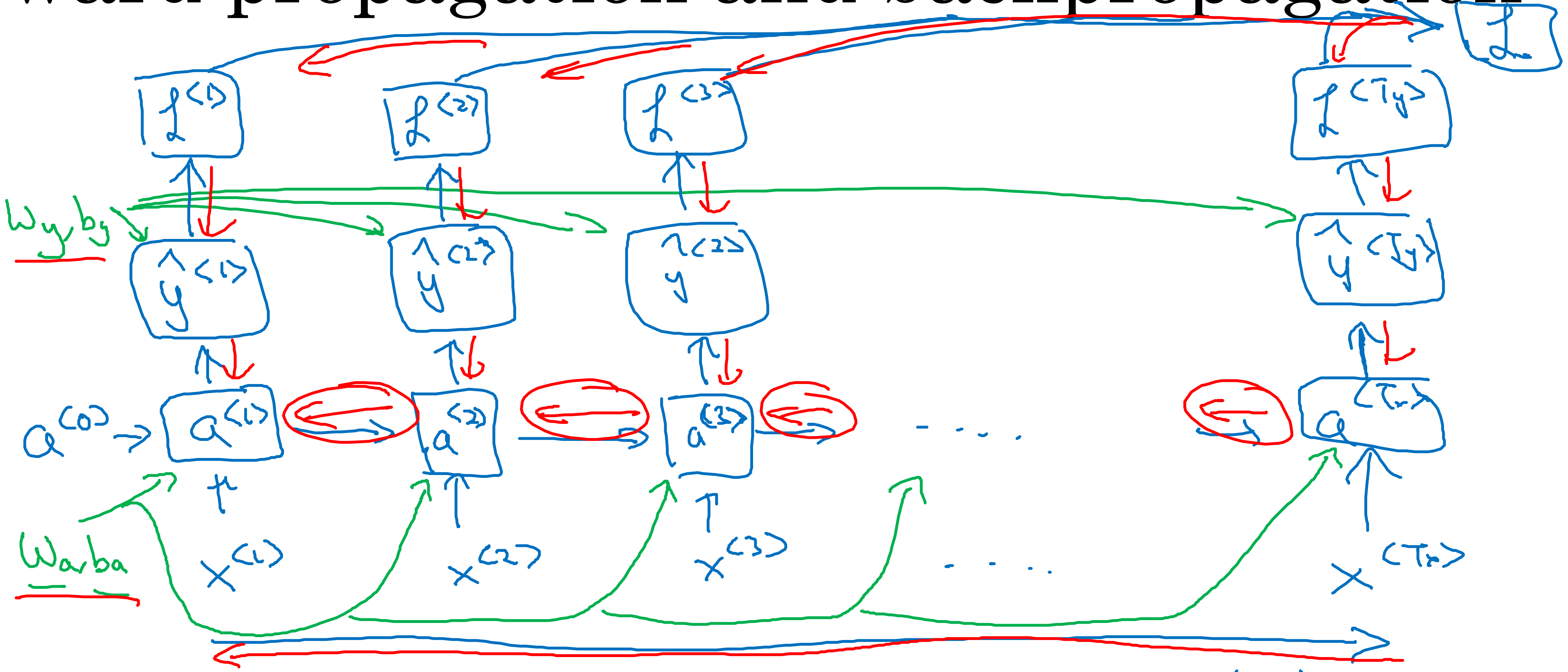
Recurrent Neural Networks

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

Andrew Ng

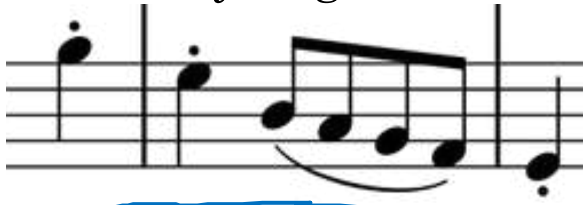
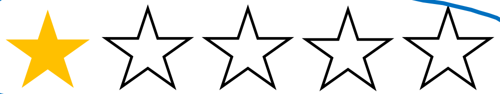
deeplearning.ai

Recurrent Neural Networks

Different types of RNNs

Examples of sequence data

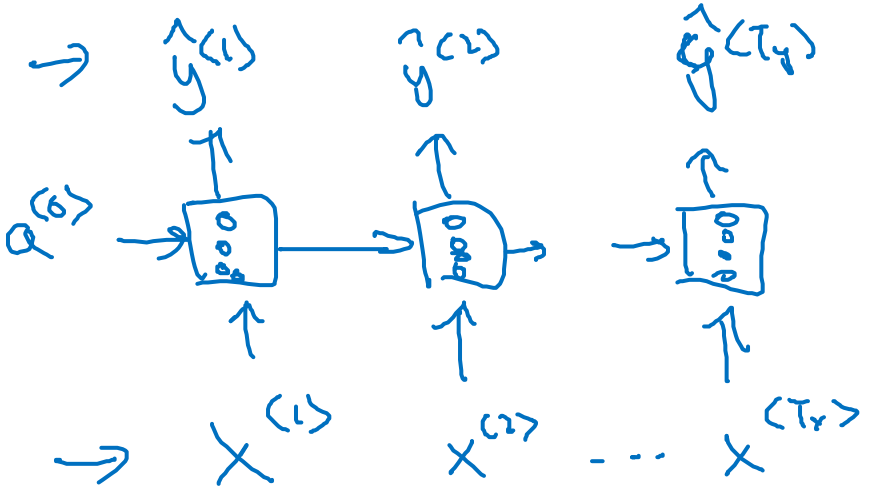
T_x T_y

	x		y
Speech recognition		→	"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

Examples of RNN architectures

$T_x = T_y$

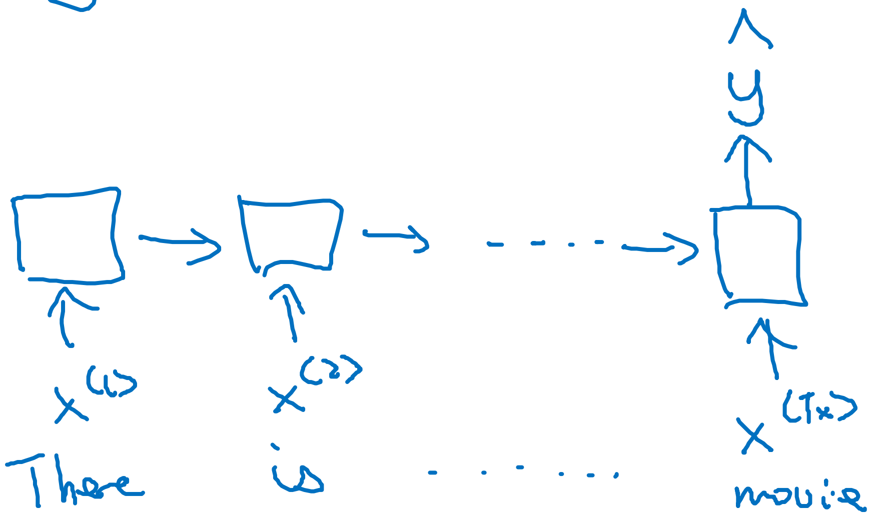


Many-to-many

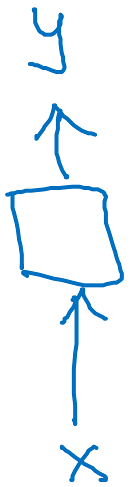
Sentiment classification

x = text

$y = 0/1 \qquad 1 \ldots 5$



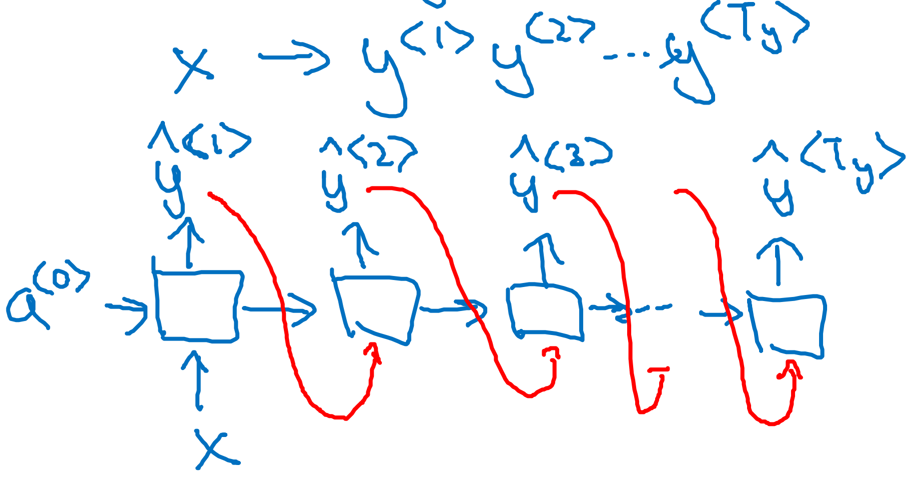
Many-to-one



one-to-one

Examples of RNN architectures

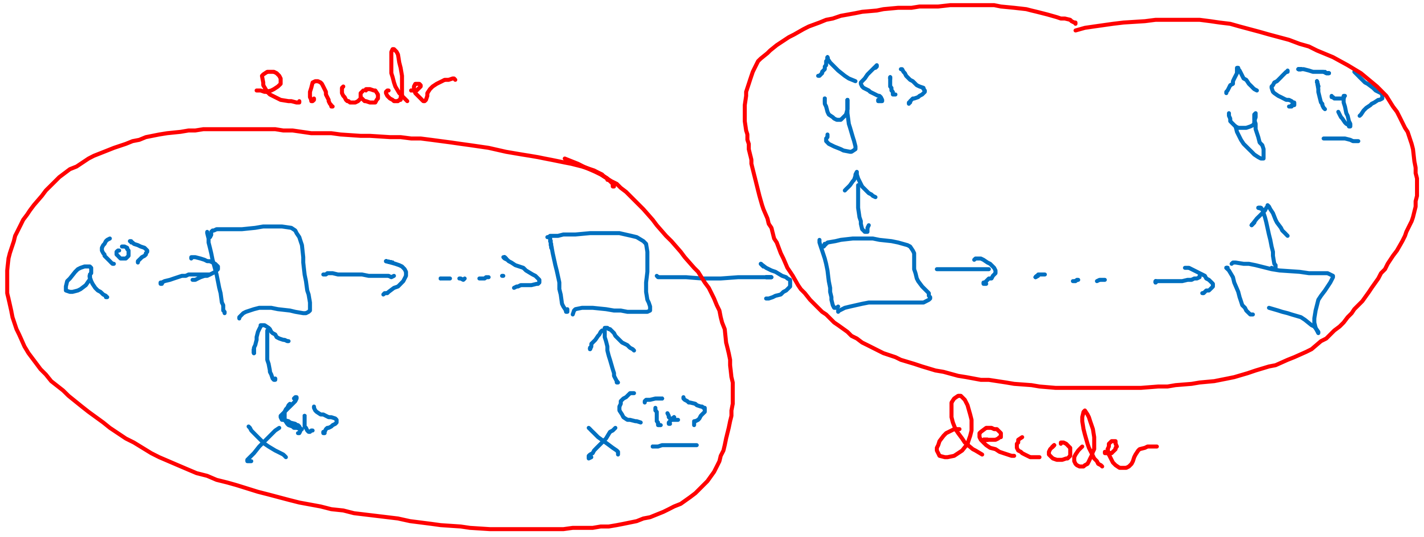
Music generation



One-to-many

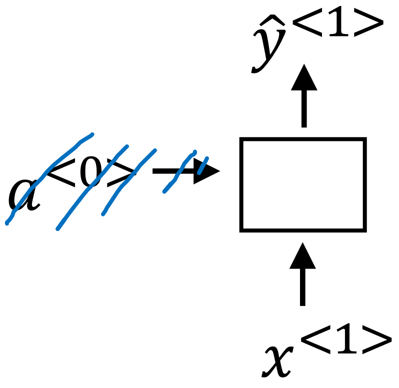
$x = \phi$

Machine translation

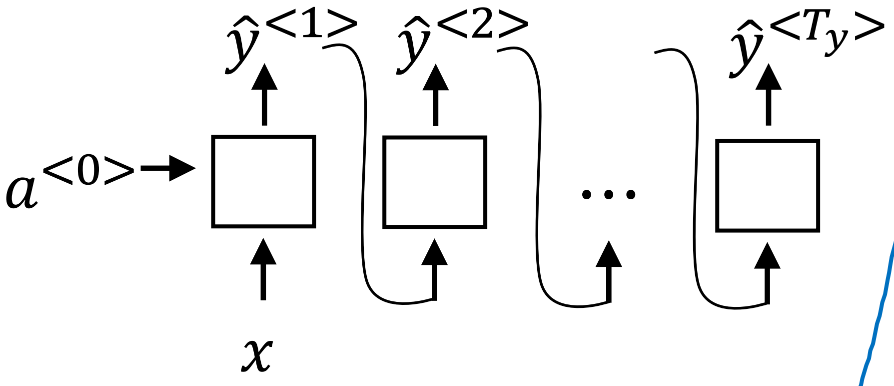


Many-to-many

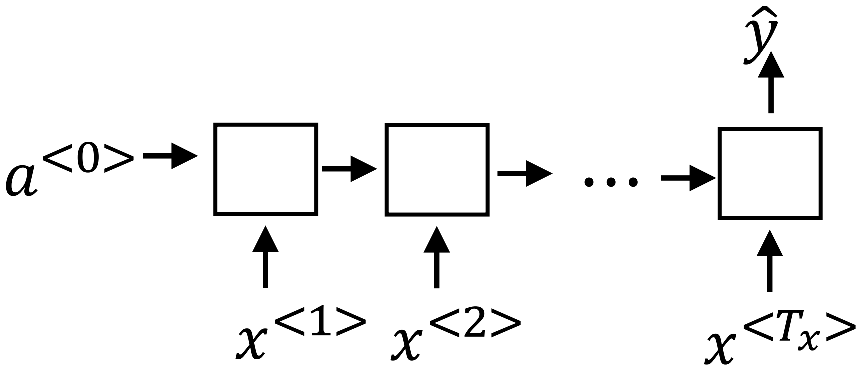
Summary of RNN types



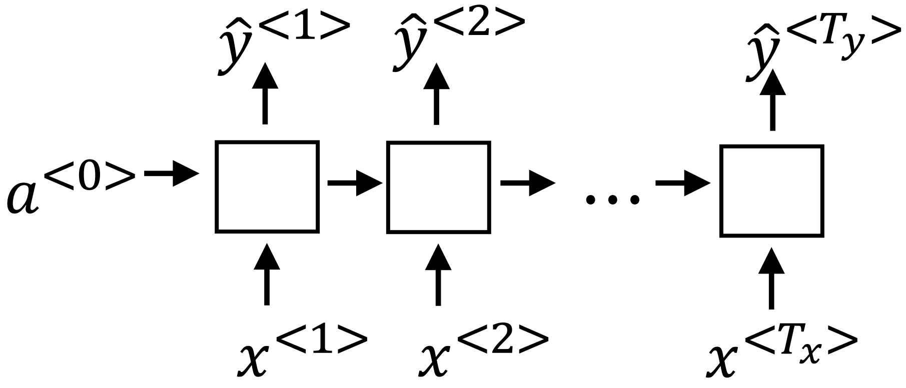
One to one



One to many

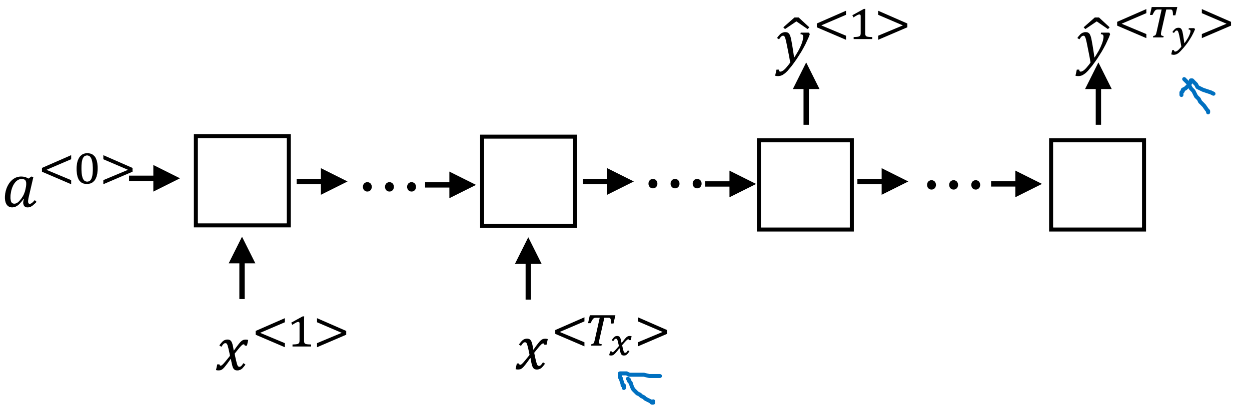


Many to one



Many to many

$T_x = T_y$



Many to many

Andrew Ng

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.

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

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

$P(\text{sentence}) = ?$ $P(y^{<1>}, y^{<2>}, \ldots, y^{<T_y>})$

Language modelling with an RNN

Training set: large corpus of english text.

Tokenize

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

$y^{<1>}$ $y^{<2>}$ $y^{<3>}$ . . . $y^{<8>}$ $y^{<9>}$

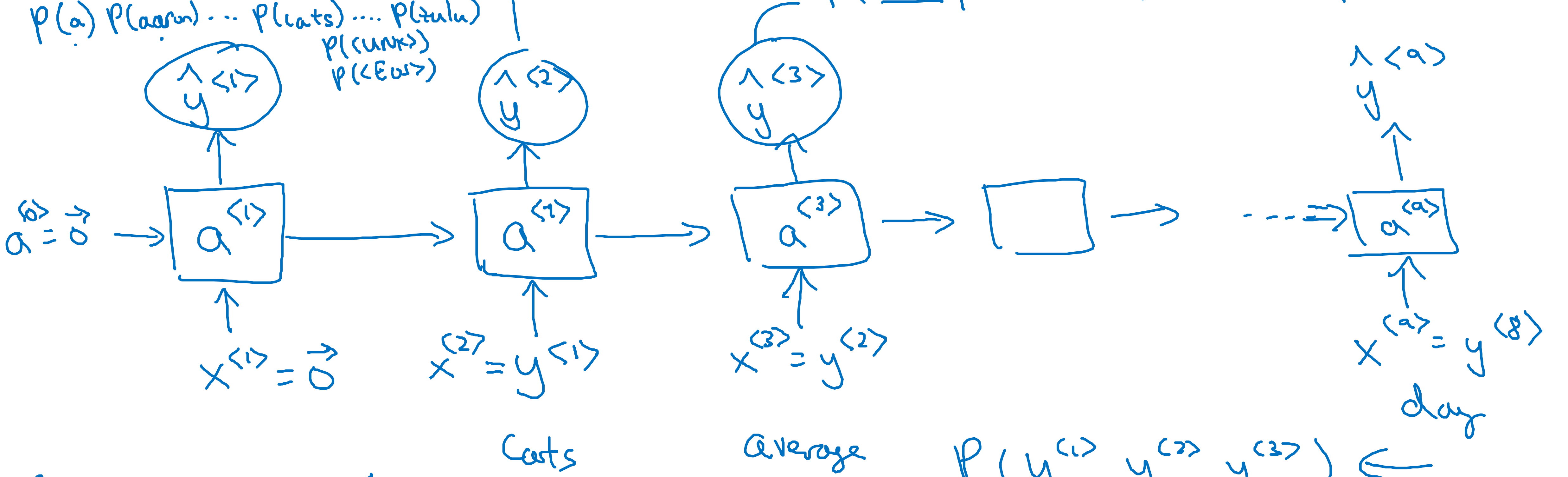
$x^{<t>} = y^{<t-1>}$

The Egyptian ~~Mau~~ is a bread of cat. <EOS>

<UNK>

10,000

RNN model



$$P(y^{<1>}, y^{<2>}, y^{<3>}) = P(y^{<1>})\,P(y^{<2>} \mid y^{<1>})\,P(y^{<3>} \mid y^{<1>}, y^{<2>})$$

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

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

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

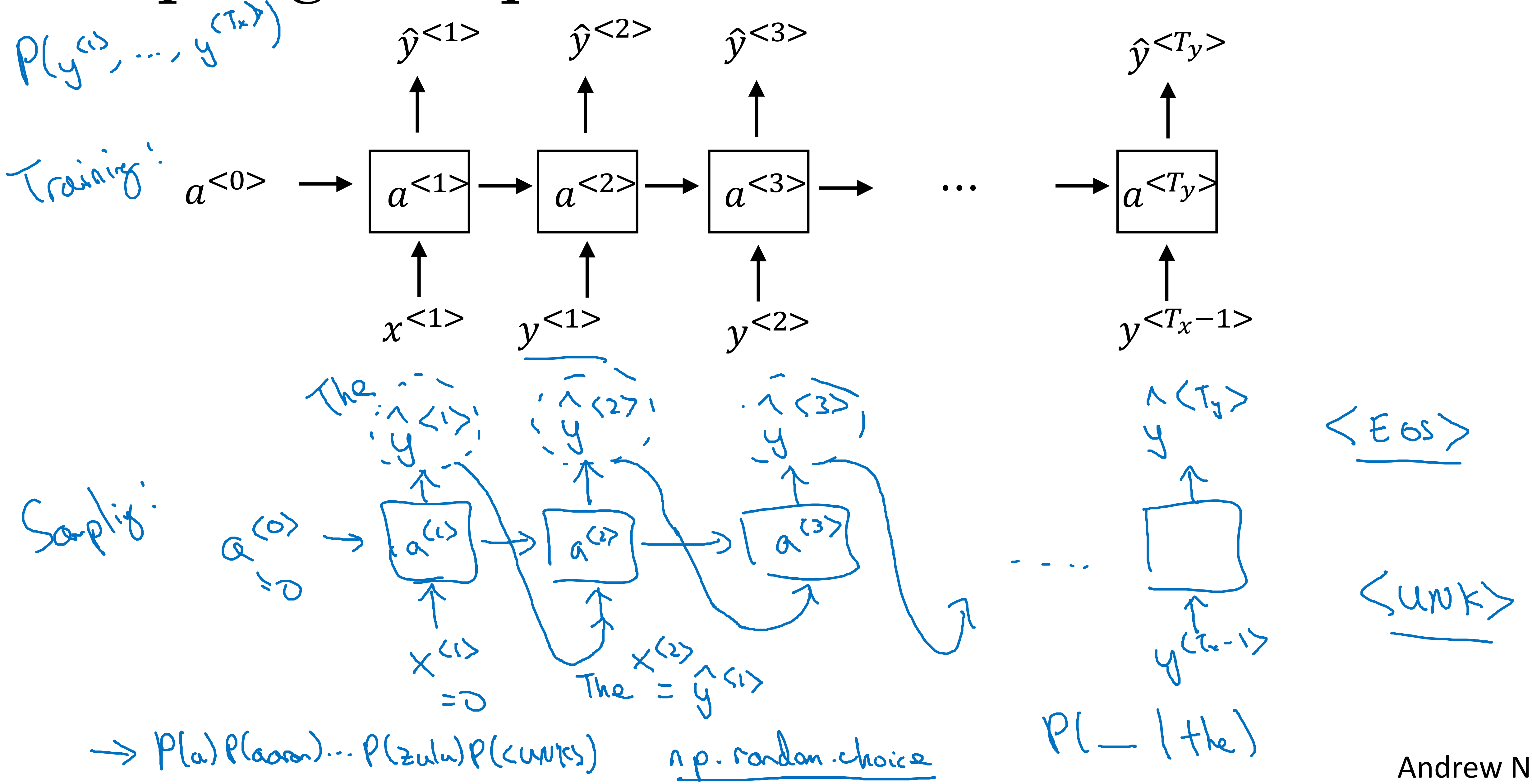
Andrew Ng

deeplearning.ai

Recurrent Neural Networks

Sampling novel sequences

Sampling a sequence from a trained RNN



Andrew Ng

Character-level language model

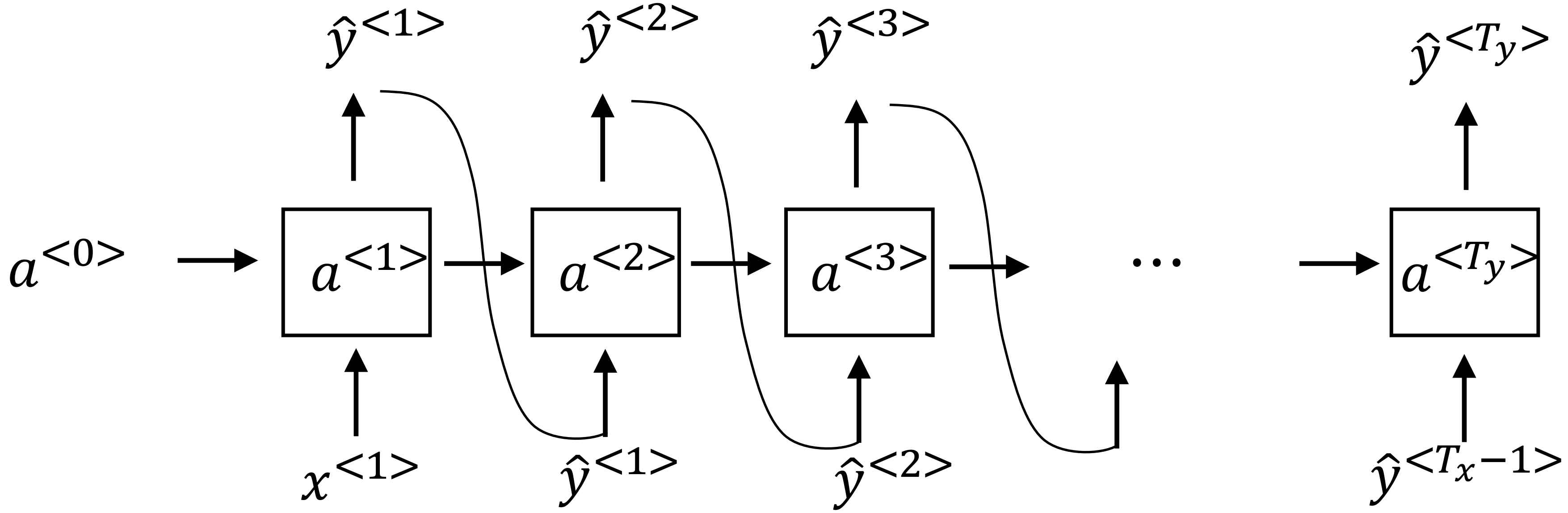
→ Vocabulary = [a, aaron, …, zulu, <UNK>] ←

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

$y^{<1>}$ $y^{<2>}$ $y^{<3>}$ $y^{<4>}$

Cat average …

Mau



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

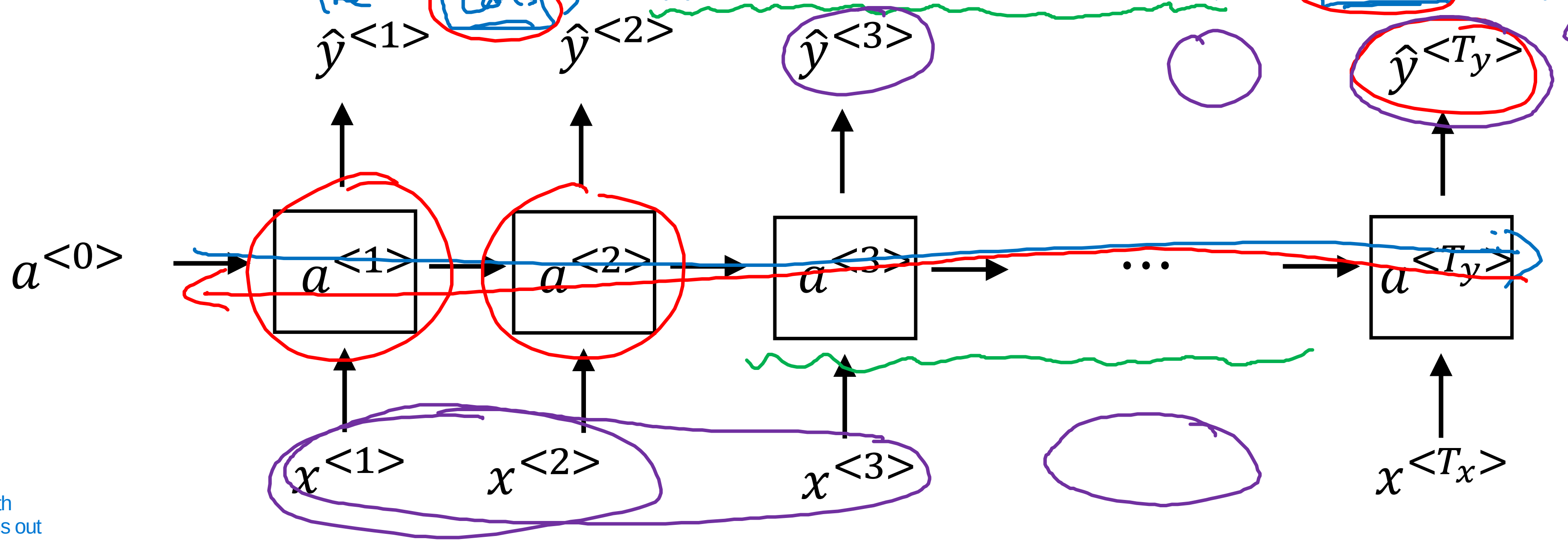
Recurrent Neural Networks

Vanishing gradients with RNNs

Vanishing gradients with RNNs

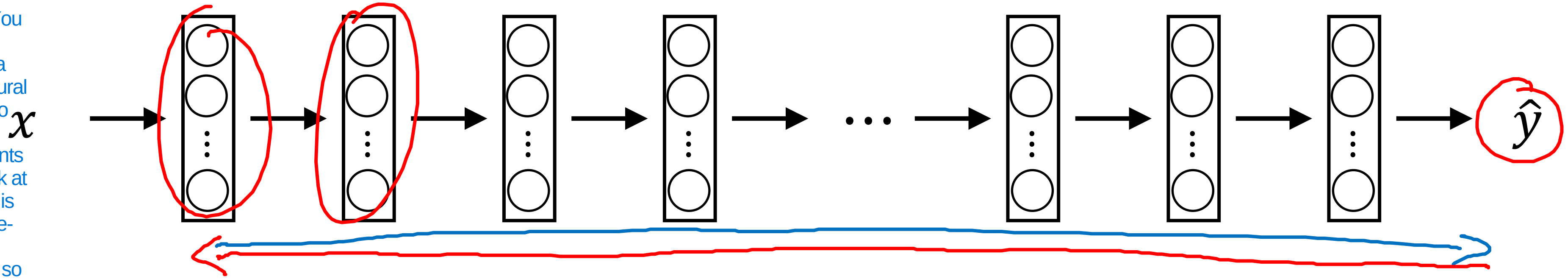
The cat, which already ate ...., was full

The cats, wh . . . ., were full



It's difficult for the outputs of the deeper layers to remember the (or get affected) by the outcome of the shallower layer as the output is greatly influenced by the layers near it rather than the ones that are far from it, hence as we saw with very deep neural networks, the problem of vanishing/exploding gradient, simple RNN architecture also faces similar problem.

Although it is easy to deal with exploding gradients as It turns out that exploding gradients are easier to spot because the parameter has just blow up. You might often see NaNs, not a numbers, meaning results of a numerical overflow in your neural network computation. If you do see exploding gradients, one solution to that is apply gradients clipping. All that means is, look at your gradient vectors, and if it is bigger than some threshold, re-scale some of your gradient vectors so that it's not too big, so that is clipped according to some maximum value



100

Exploding gradients.

NaN

Gradient clipping

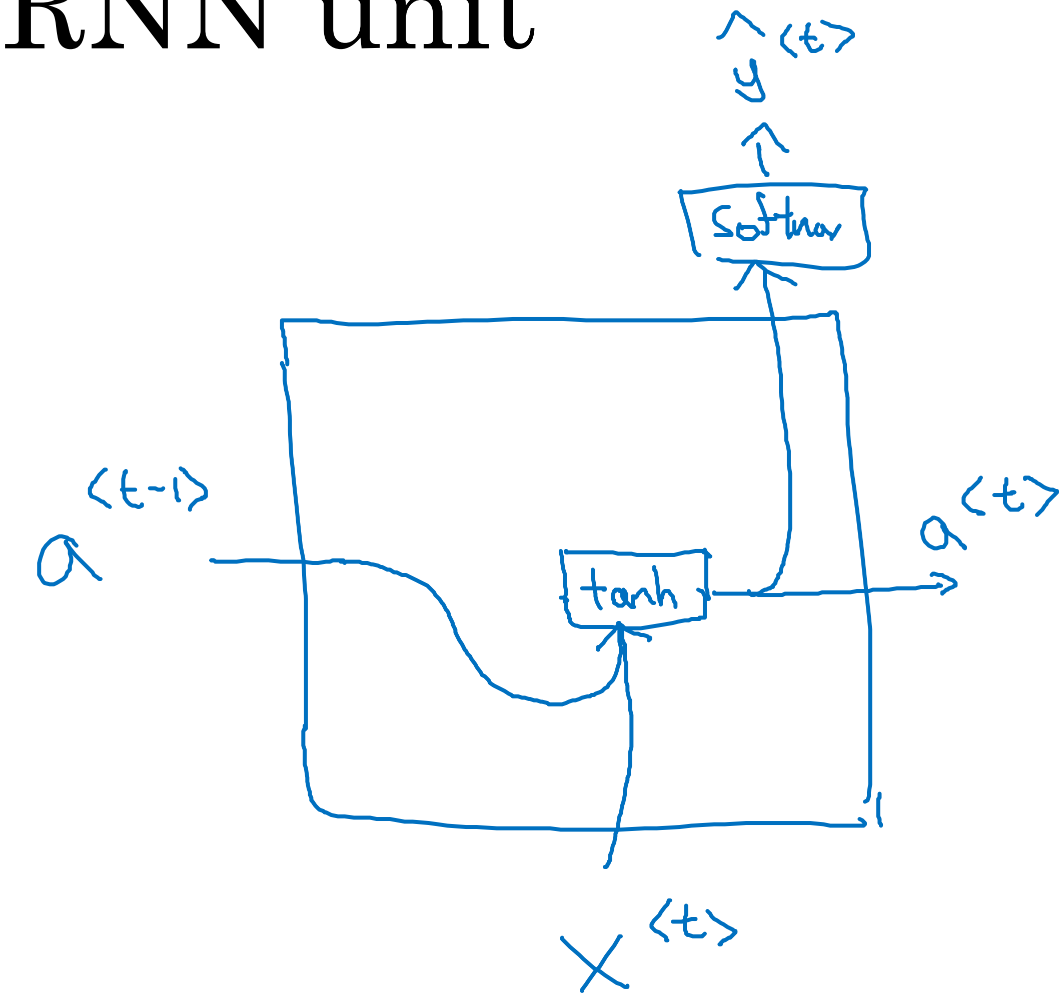
Andrew Ng

deeplearning.ai

Recurrent Neural Networks

Gated Recurrent Unit (GRU)

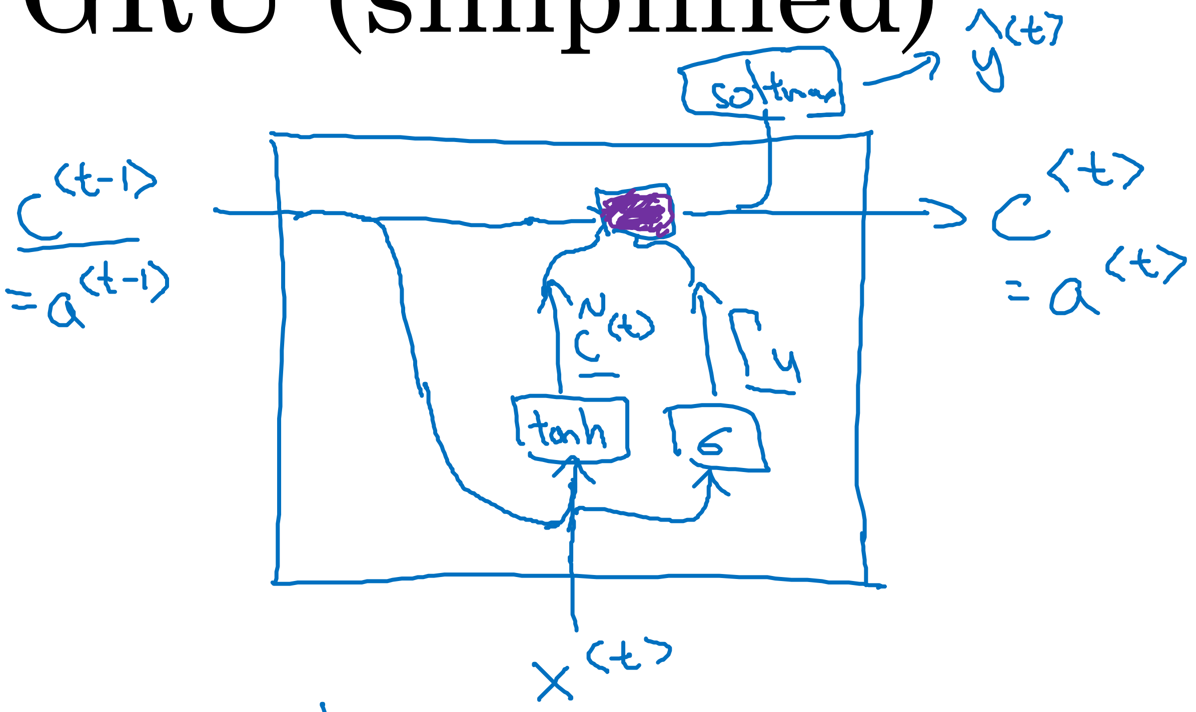
RNN unit

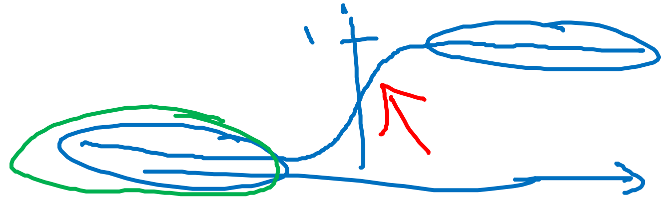


tanh

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

GRU (simplified)





c = memory cell

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

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

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

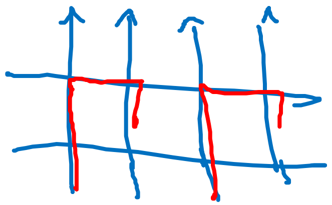
"update"

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

$\Gamma_u = 1$

element-wise

$\Gamma_u = 0.000001$



Gate

$\Gamma_u = 1$ $\Gamma_u = 0$ $\Gamma_u = 0$ $\Gamma_u = 0$ . . . . . .

$c^{<t>} = 1$ . . . . $= 1$

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]

Full GRU

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

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

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

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

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

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

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

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

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

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

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

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

[Hochreiter & Schmidhuber 1997. Long short-term memory]

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

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

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

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

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

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

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

[Hochreiter & Schmidhuber 1997. Long short-term memory]

LSTM in pictures

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

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

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

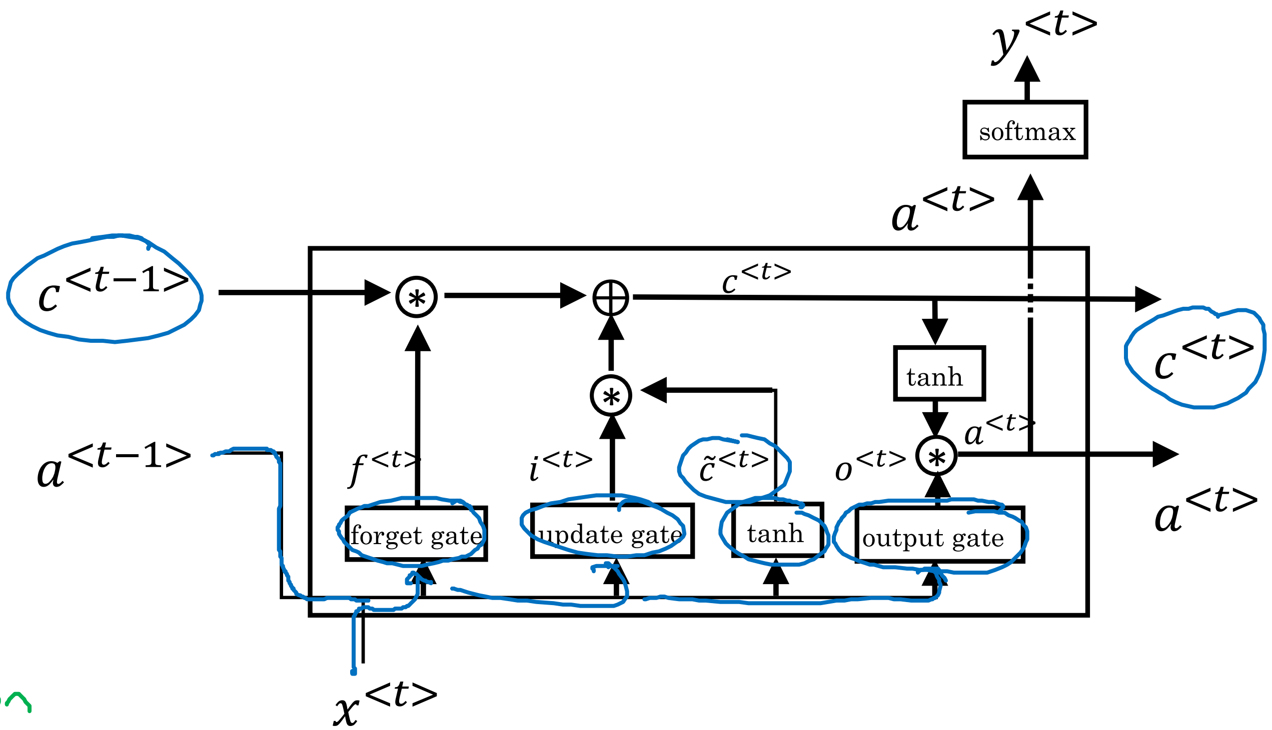
$$\Gamma_o = \sigma(W_o[\,a^{<t-1>}, x^{<t>}] + b_o)$$

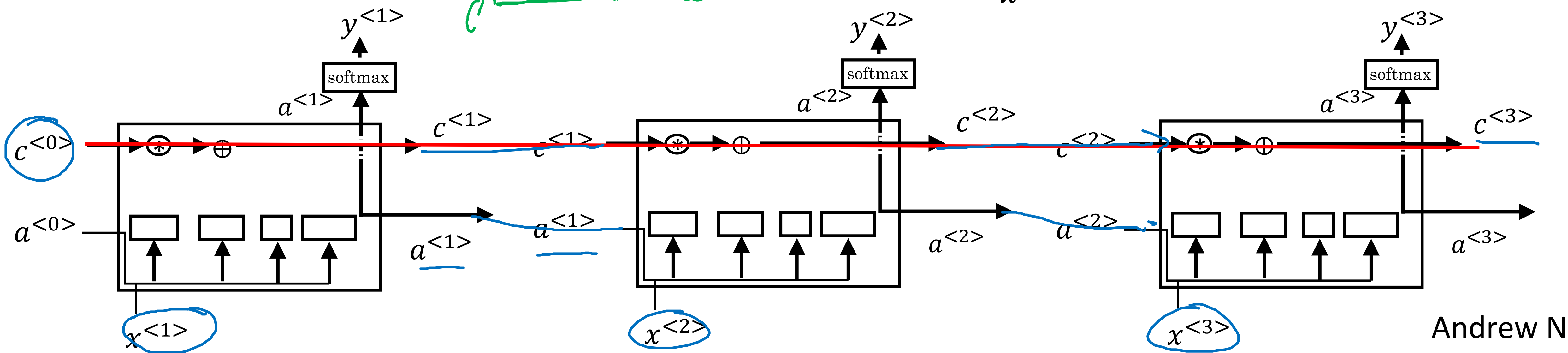
$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>}$$

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

$\boxed{c^{<t-1>}}$

peephole connection





Andrew Ng

deeplearning.ai

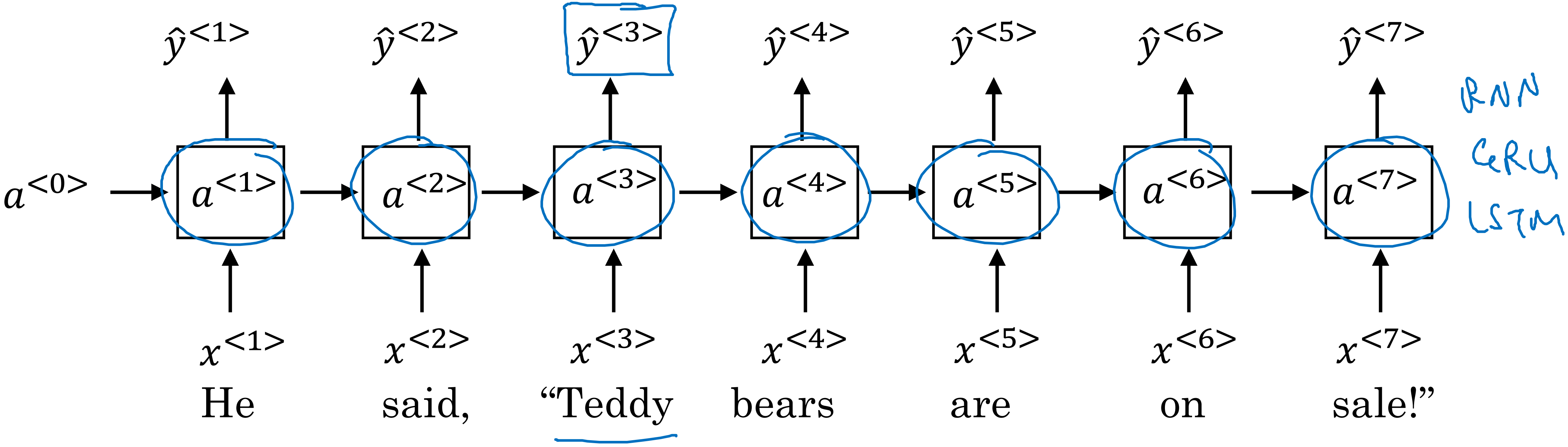
Recurrent Neural Networks

Bidirectional RNN

Getting information from the future

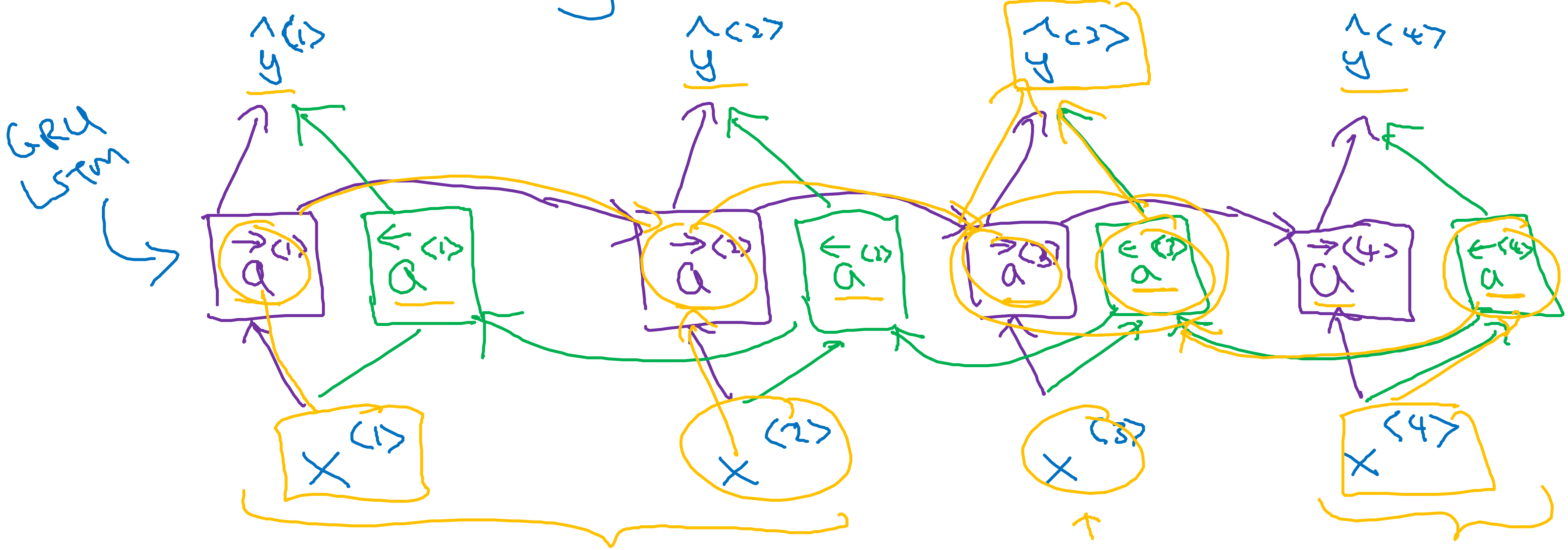
He said, “Teddy bears are on sale!”

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



Bidirectional RNN (BRNN)

$$\hat{y}^{<t>} = g(W_y[\overrightarrow{a}^{<t>}, \overleftarrow{a}^{<t>}] + b_y)$$



Acyclic graph

He said, "Teddy Roosevelt ...."

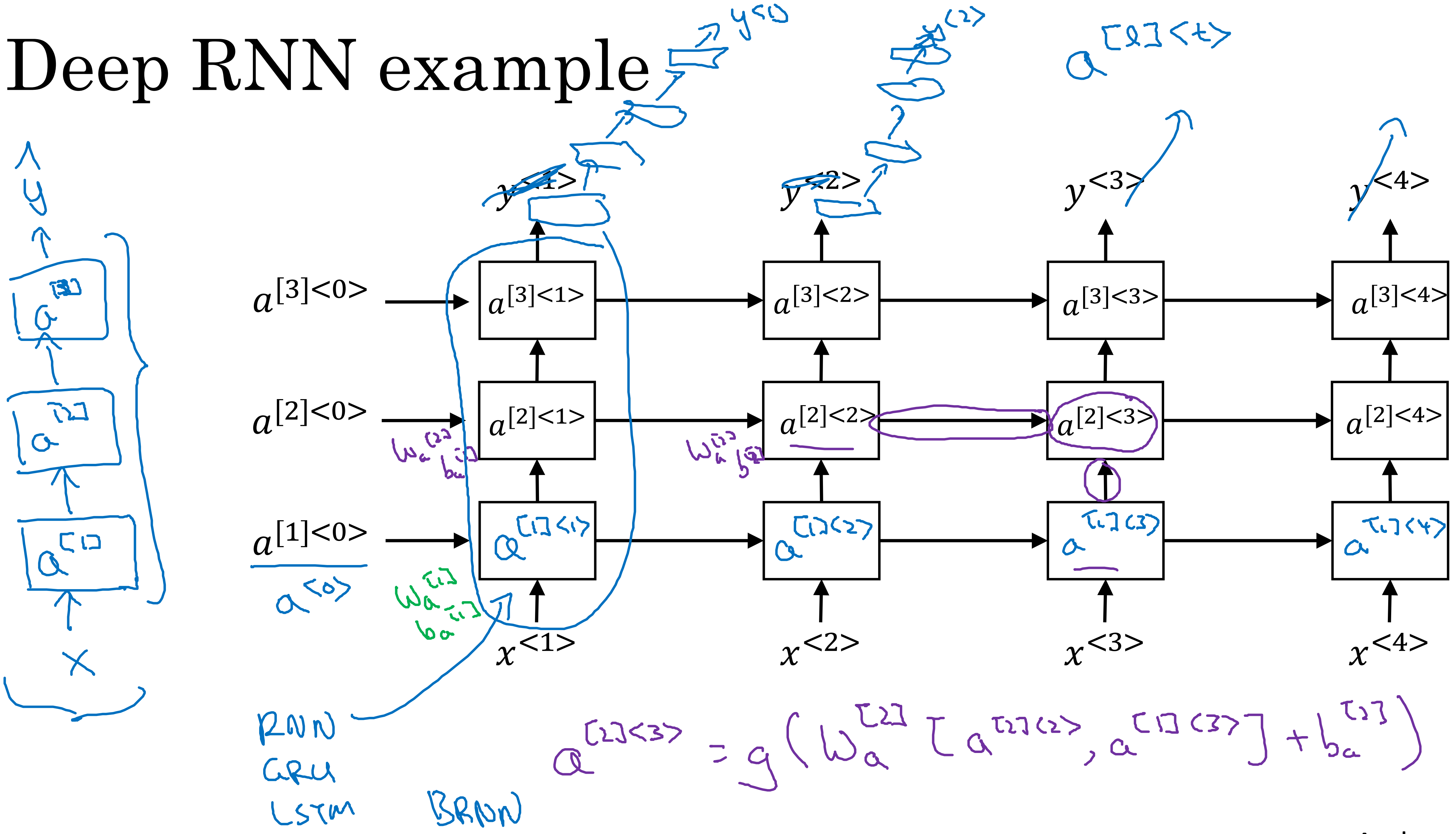
BRNN w/ LSTM

deeplearning.ai

Recurrent Neural Networks

Deep RNNs

Deep RNN example



$$a^{[2]<3>} = g\left(W_a^{[2]}\left[a^{[2]<2>}, a^{[1]<3>}\right] + b_a^{[2]}\right)$$

Andrew Ng