

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>



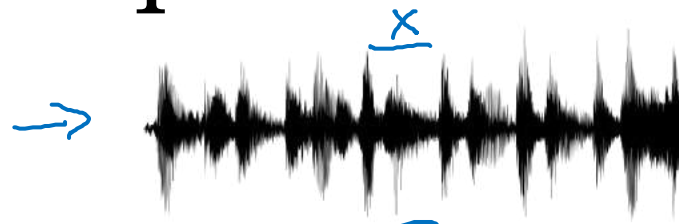
deeplearning.ai

Recurrent Neural Networks

Why sequence
models?

Examples of sequence data

Speech recognition



“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**CCCCTGTGAGGAACTAG**

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.

$\rightarrow x^{(1)} \quad x^{(2)} \quad x^{(3)} \quad \dots \quad x^{(t)} \quad \dots \quad x^{(9)}$

$$T_x = 9$$

$\rightarrow y:$

$y^{(1)} \quad y^{(2)} \quad y^{(3)} \quad \dots \quad y^{(9)}$

$$T_y = 9$$

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

$$T_x^{(i)} = 9$$

15

$y^{(i)(t)}$
 \uparrow

$$T_y^{(i)}$$

(x, y)

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

↑ $\langle \tau \rangle$
X



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

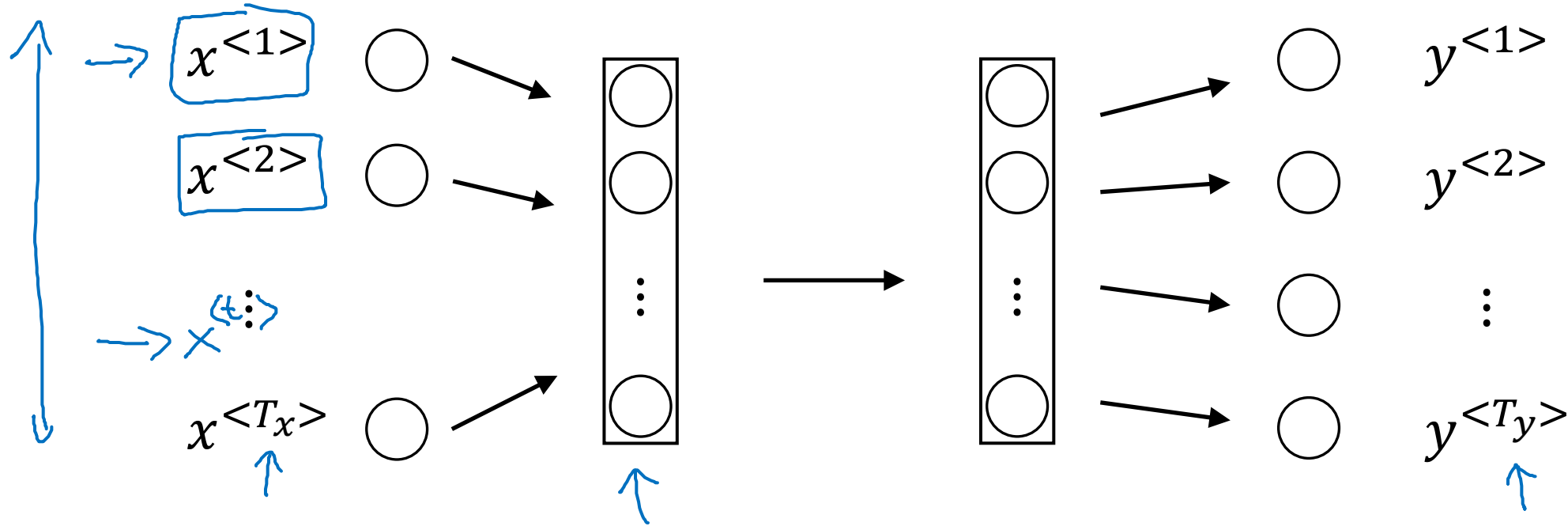


deeplearning.ai

Recurrent Neural Networks

Recurrent Neural Network Model

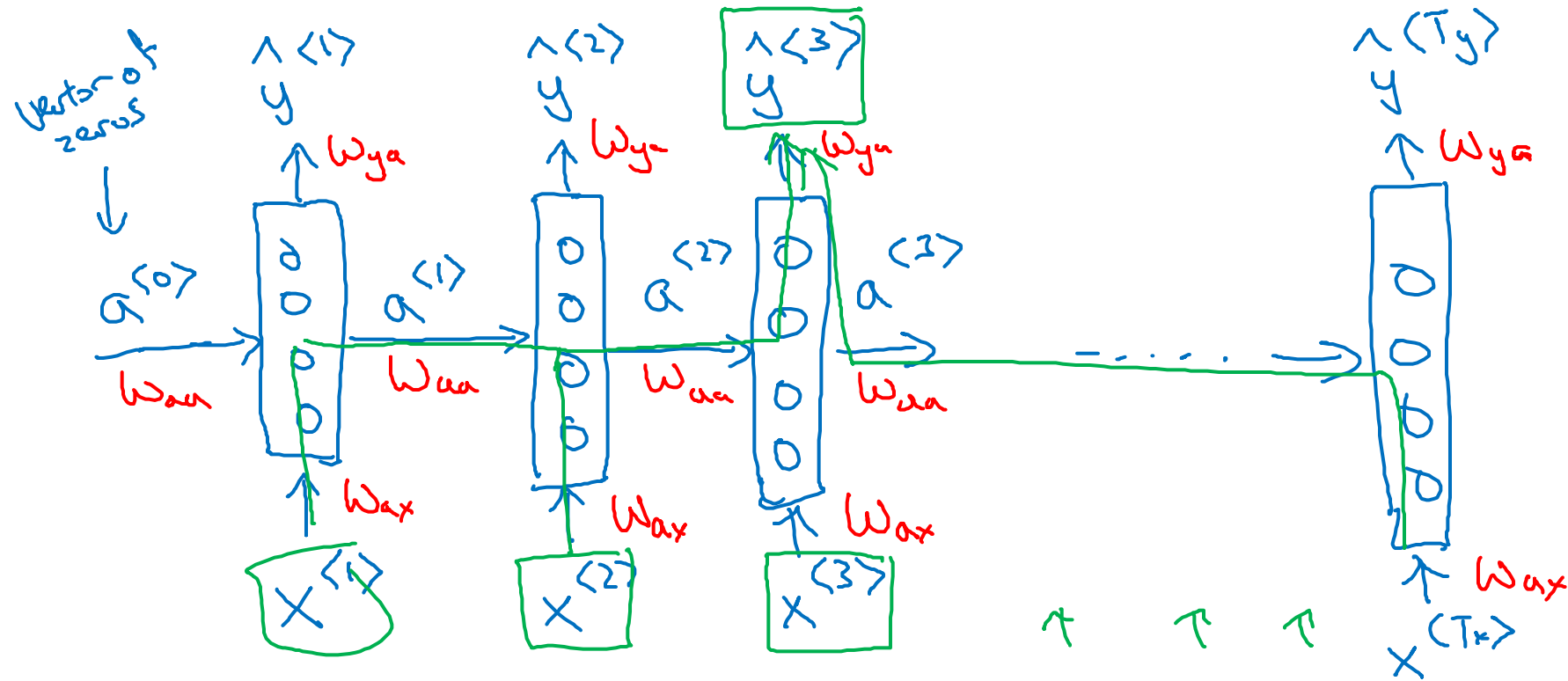
Why not a standard network?



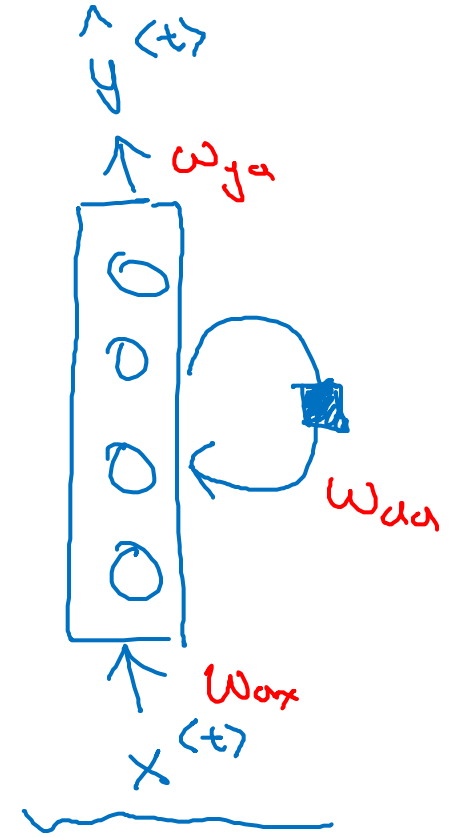
Problems:

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

Recurrent Neural Networks



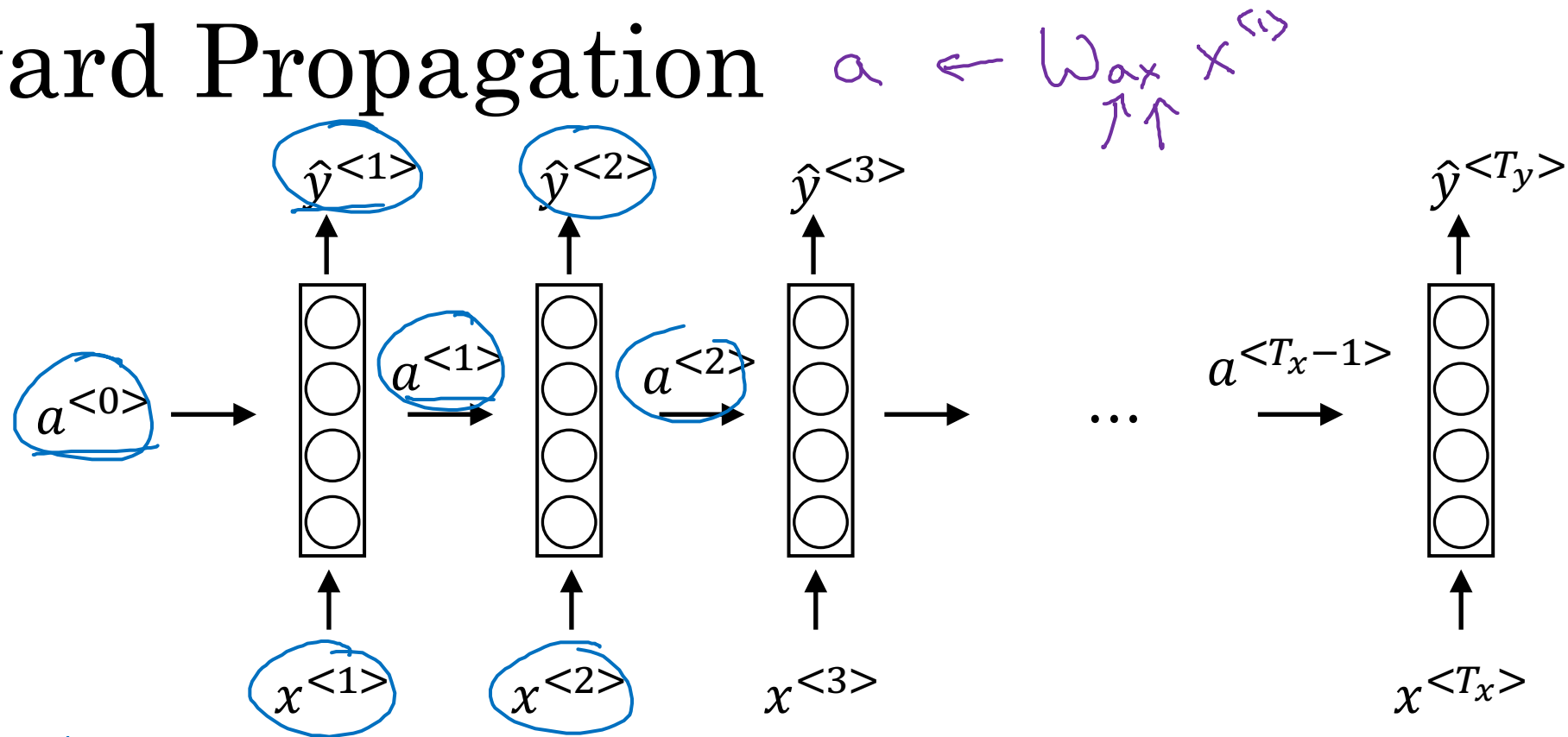
Bidirectional RNN (BRNN)



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

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

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

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

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

$$\begin{matrix} \uparrow 100 \\ \left[W_{aa} \mid W_{ax} \right] \\ \leftarrow 100 \quad \leftarrow 10,000 \end{matrix} = W_a \quad (100, 10,000)$$

$$[a^{<t-1>}, x^{<t>}] = \begin{bmatrix} a^{<t-1>} \\ x^{<t>} \end{bmatrix} \quad \begin{matrix} \updownarrow 100 \\ \updownarrow 10,000 \\ \updownarrow 10,100 \end{matrix}$$

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

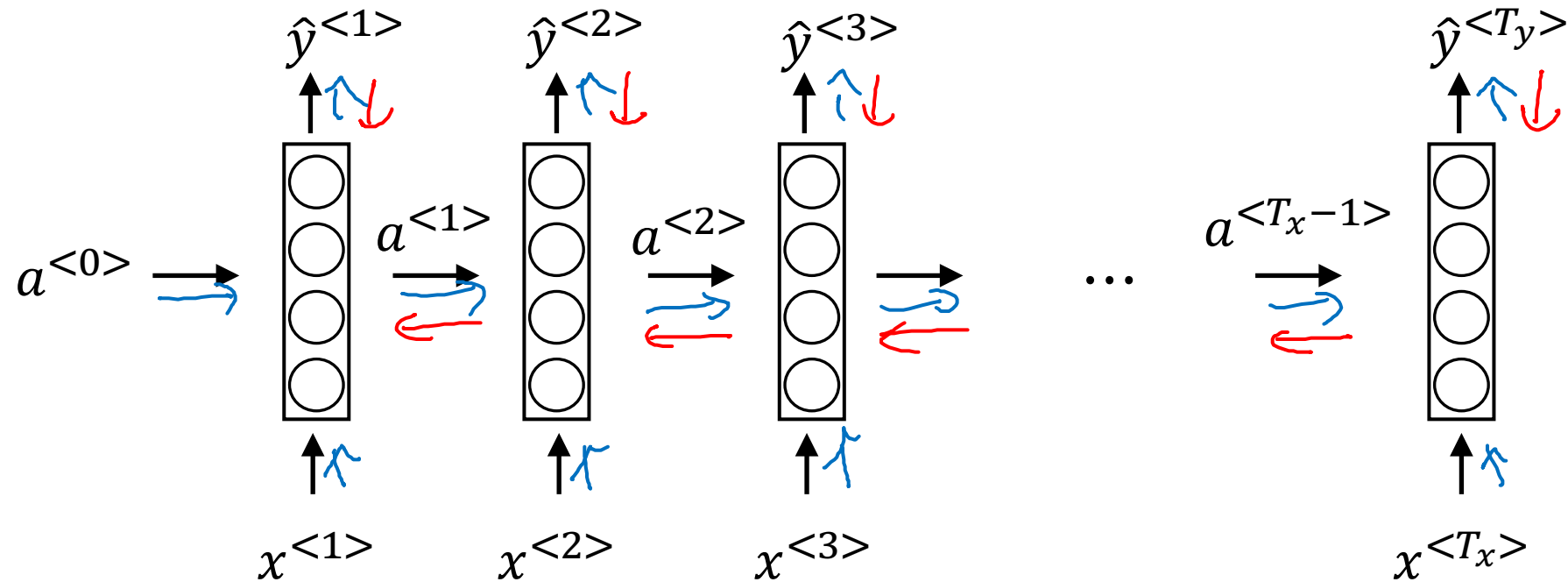


deeplearning.ai

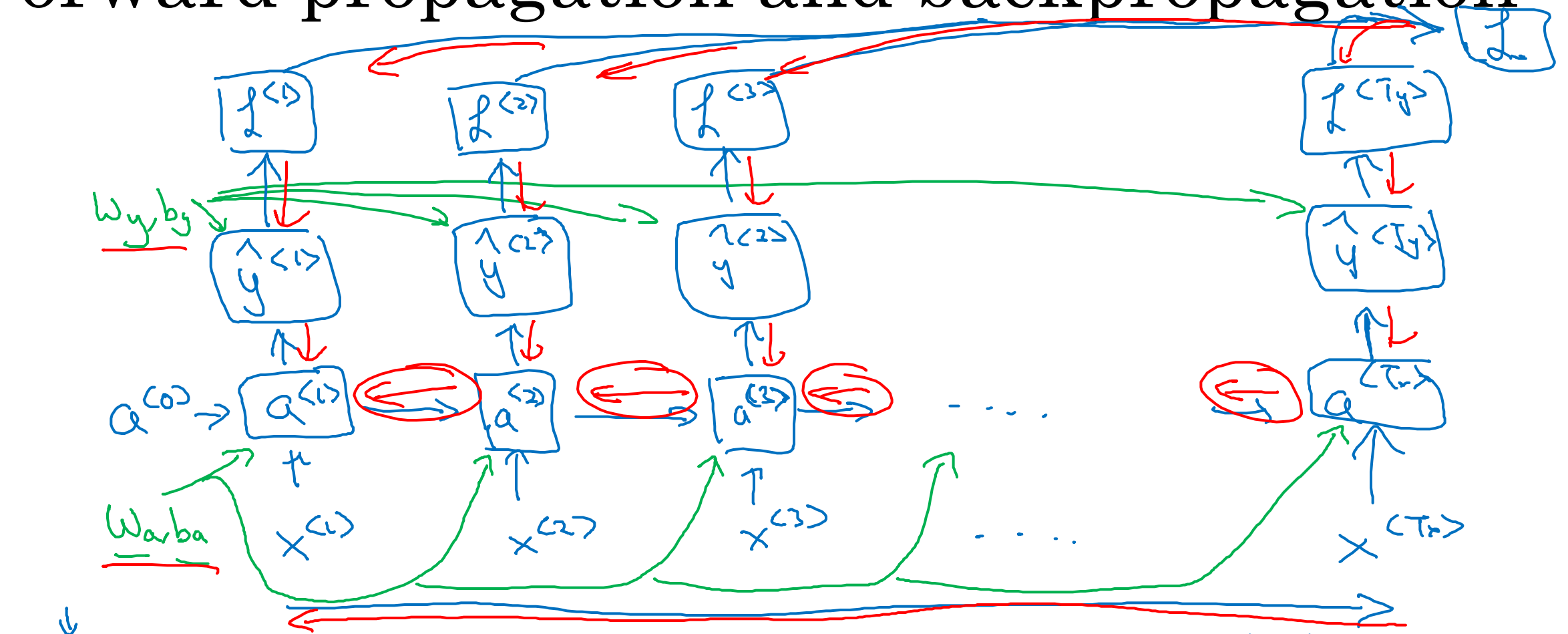
Recurrent Neural Networks

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



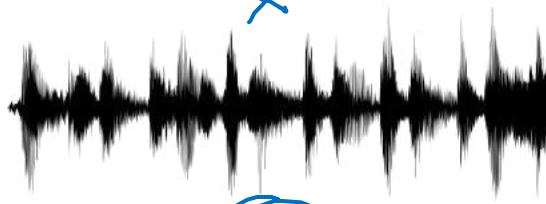
deeplearning.ai

Recurrent Neural Networks

Different types of RNNs

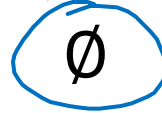
Examples of sequence data

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

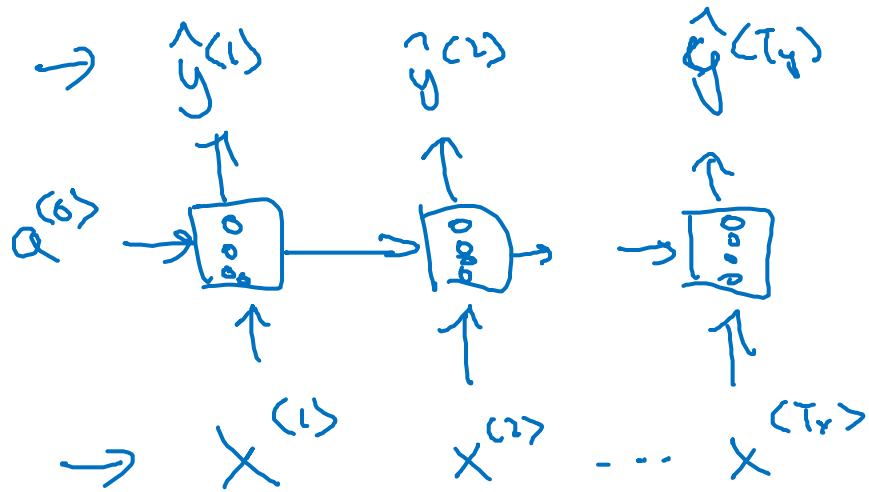
Yesterday, Harry Potter met Hermione Granger.



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

Examples of RNN architectures

$$T_x = T_y$$

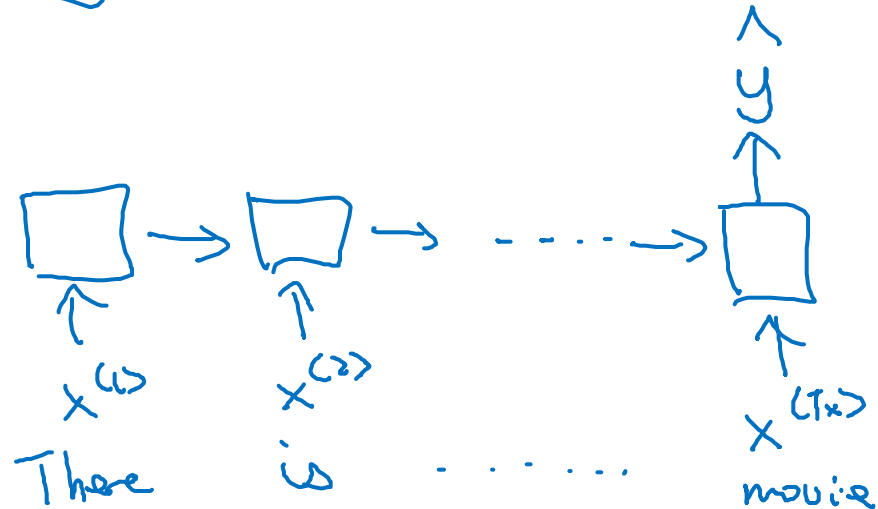


Many-to-many

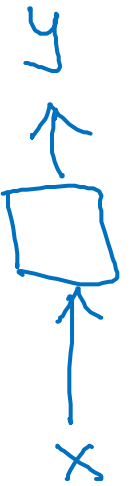
Sentiment classification-

$x = \text{text}$

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

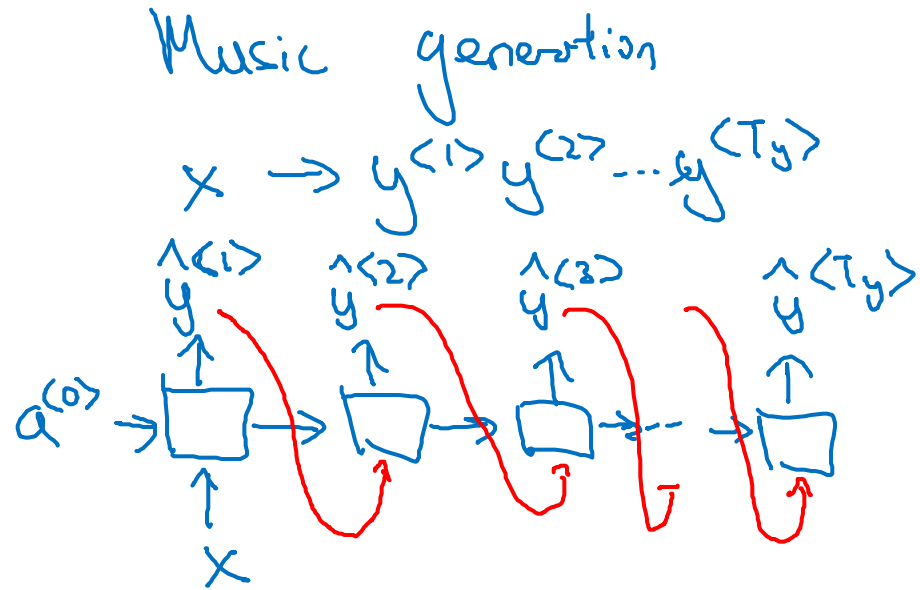


Many-to-one



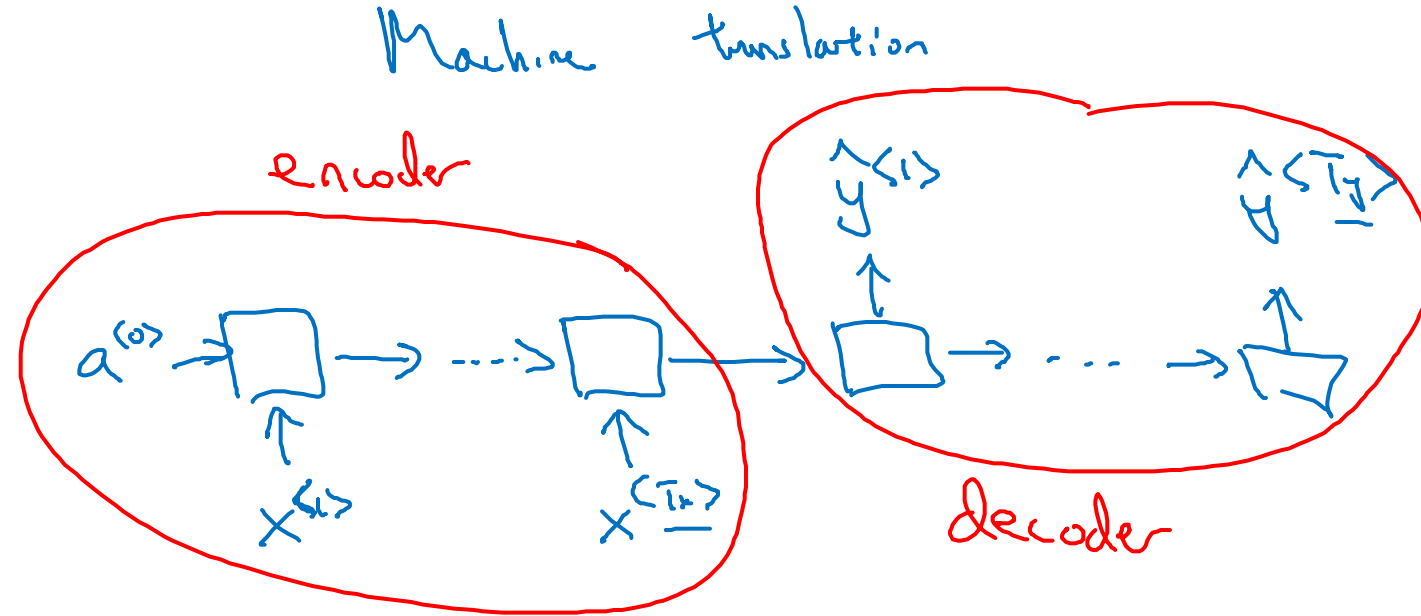
One-to-one

Examples of RNN architectures



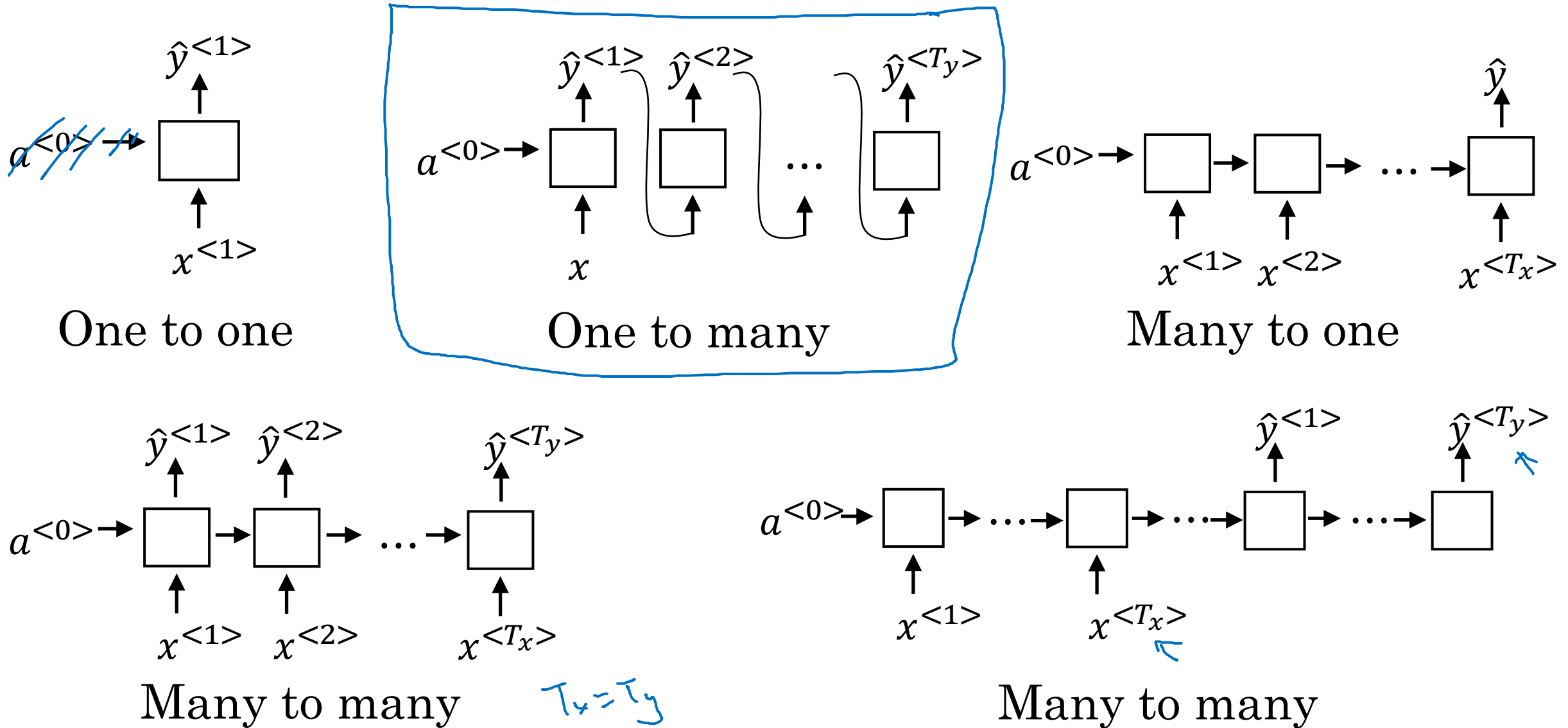
One-to-many

$$x = \phi$$



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.

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

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

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

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

Language modelling with an RNN

Training set: large corpus of english text.

Tokenize

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

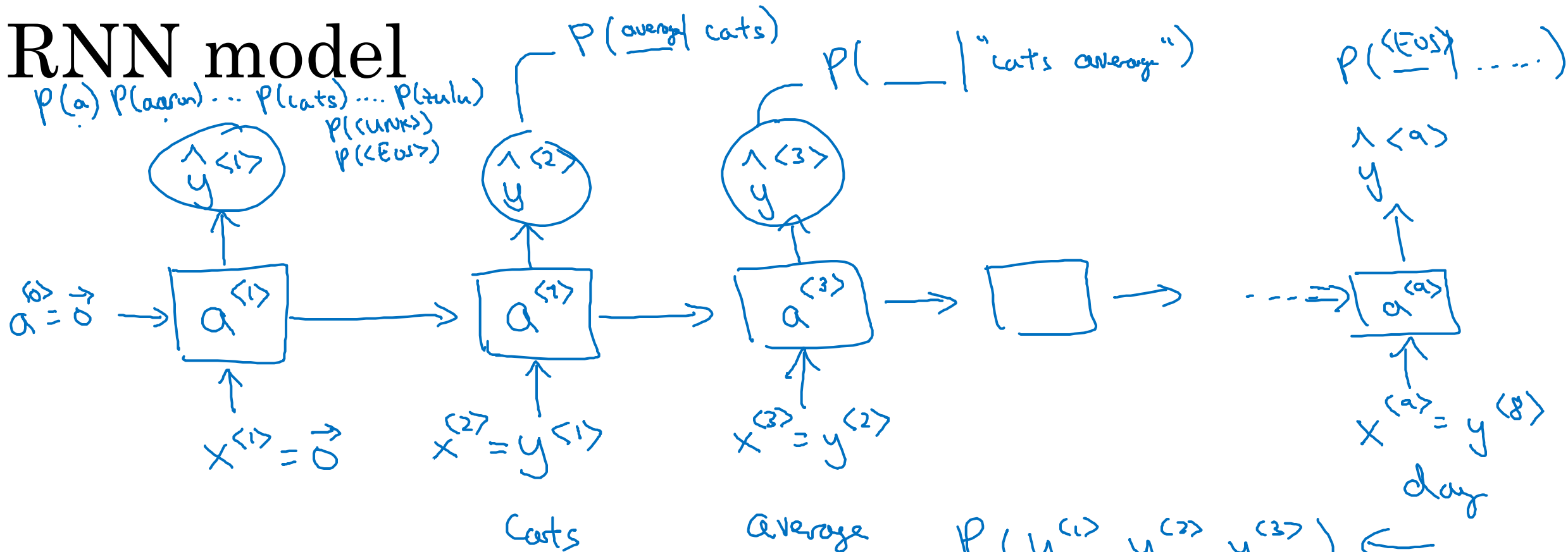
$y^{(1)}$ $y^{(2)}$ $y^{(3)}$... $y^{(8)}$ $y^{(9)}$
 $x^{(t)} = y^{(t-1)}$

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

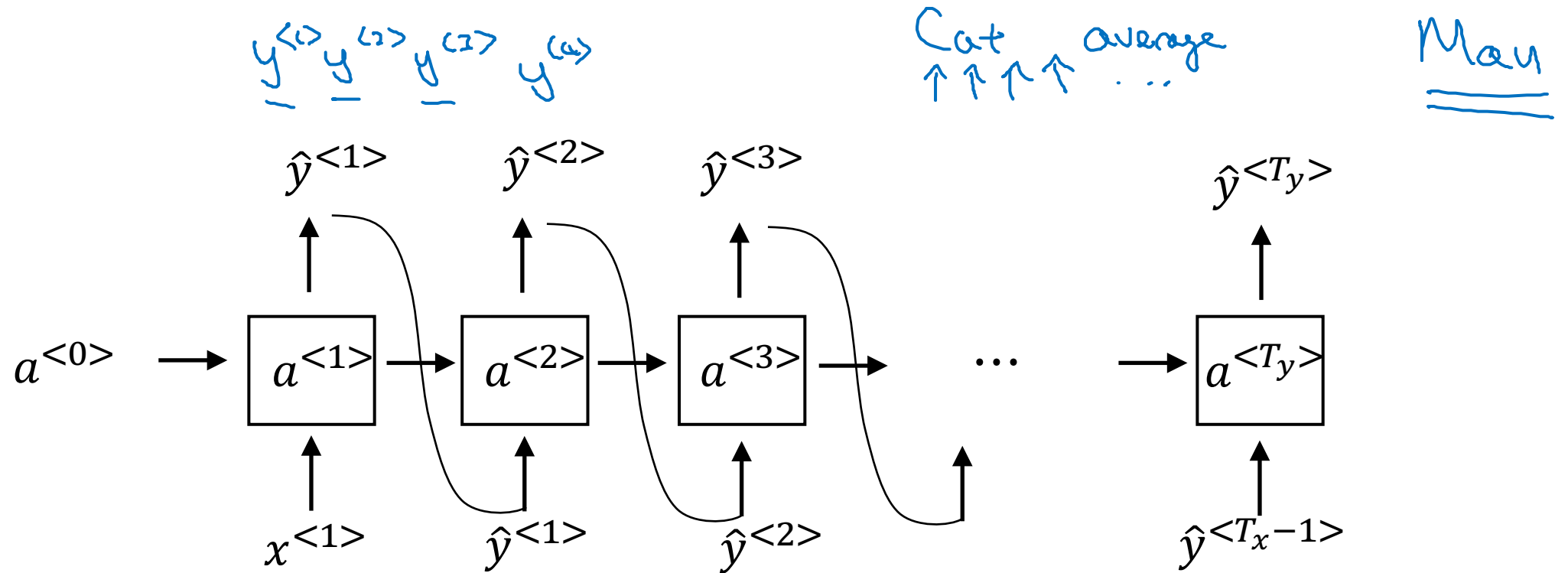
Sampling novel
sequences

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


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.

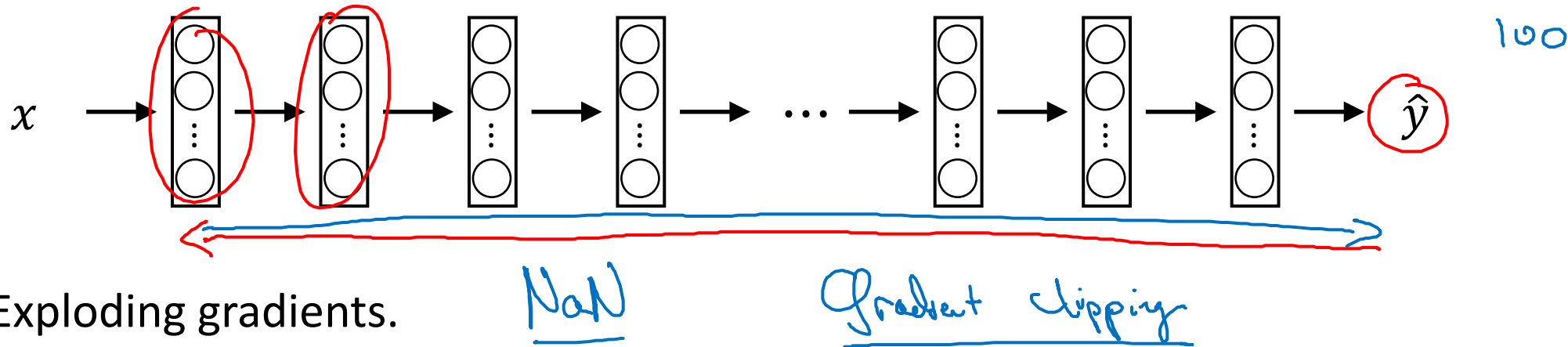
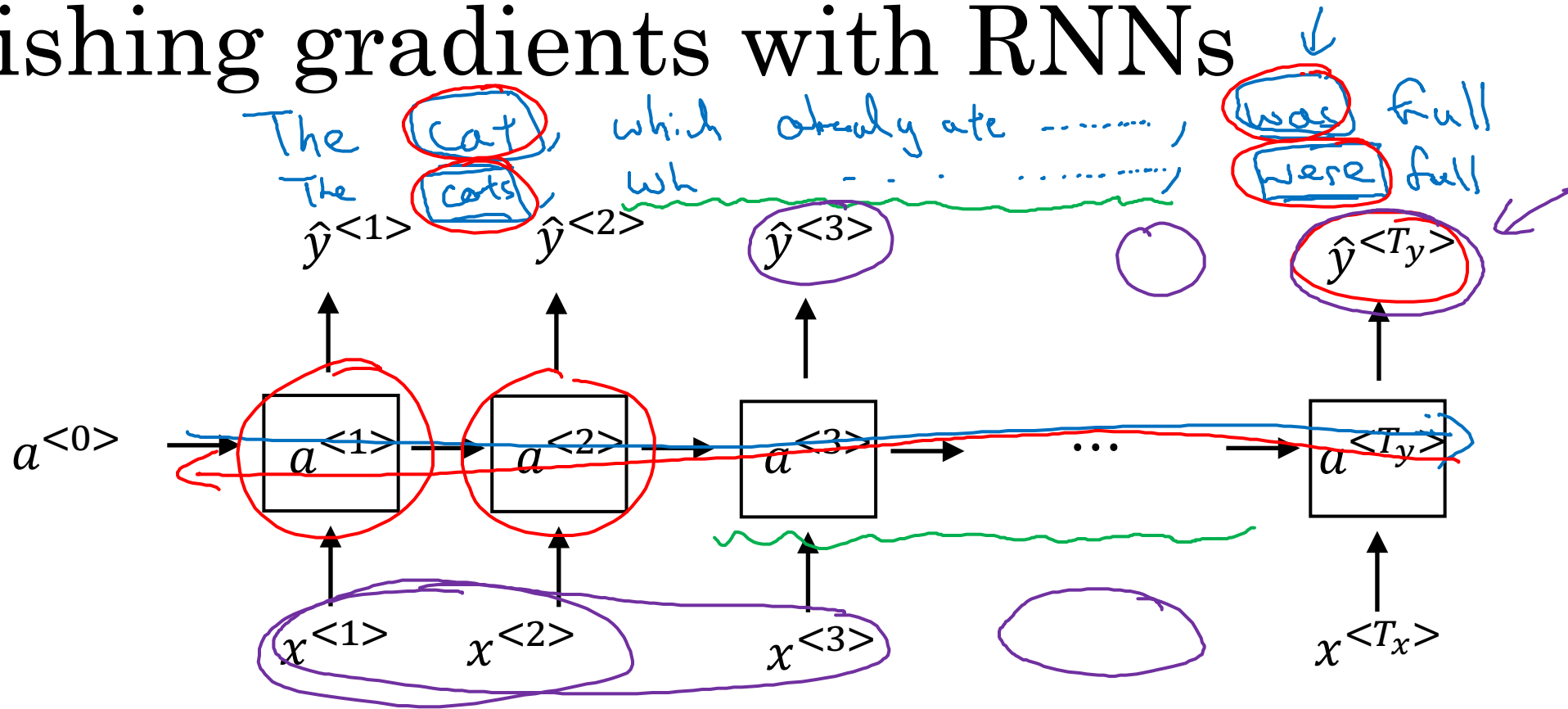


deeplearning.ai

Recurrent Neural Networks

Vanishing gradients with RNNs

Vanishing gradients with RNNs



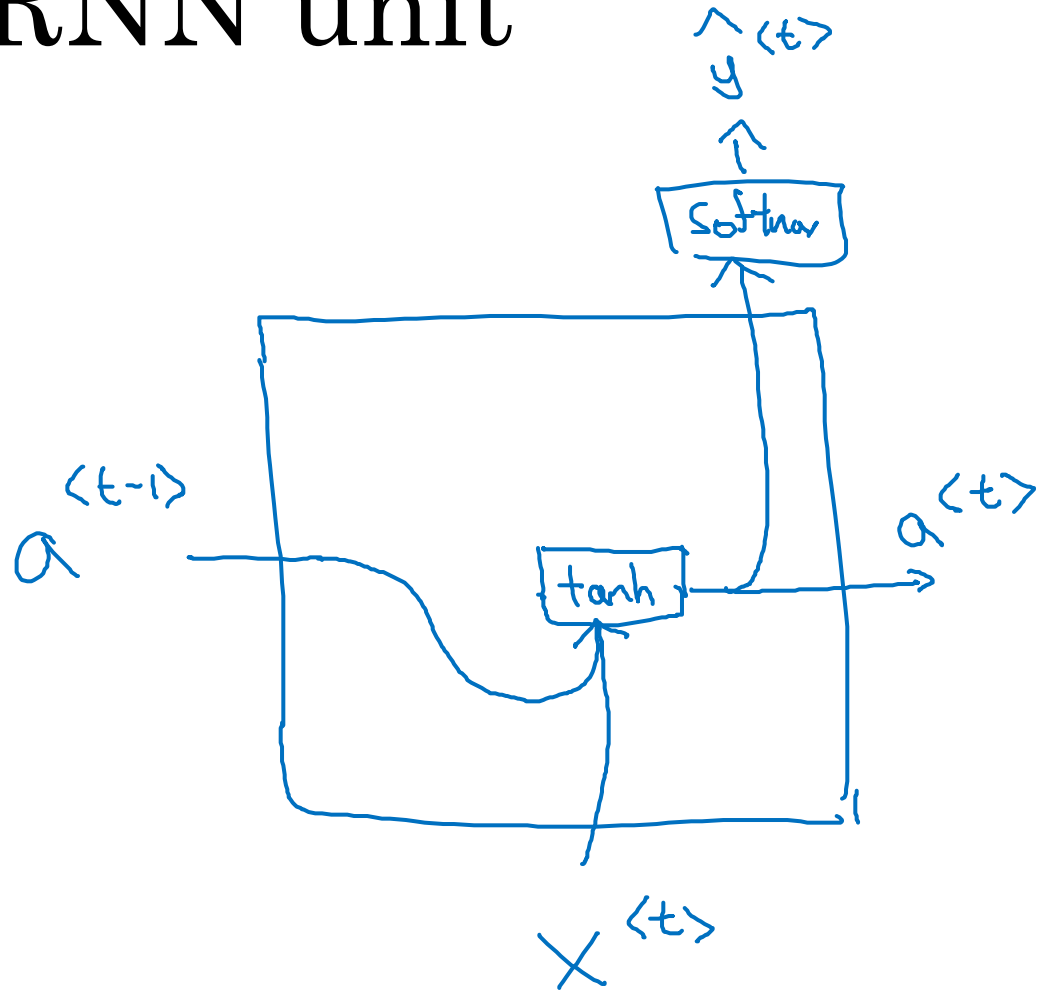


deeplearning.ai

Recurrent Neural Networks

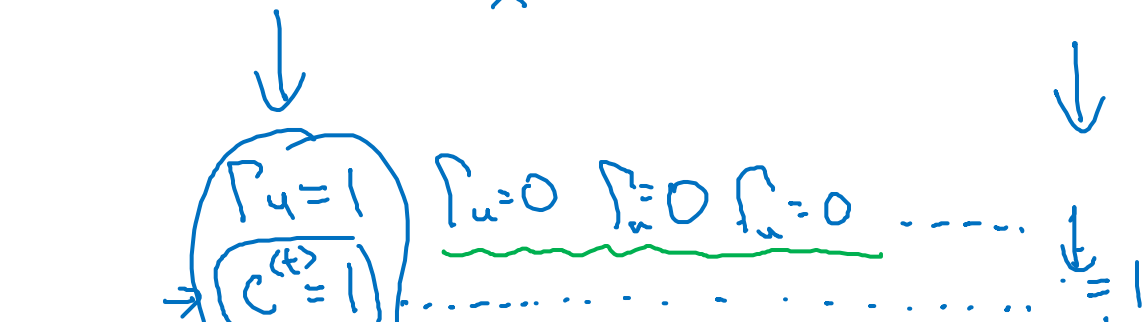
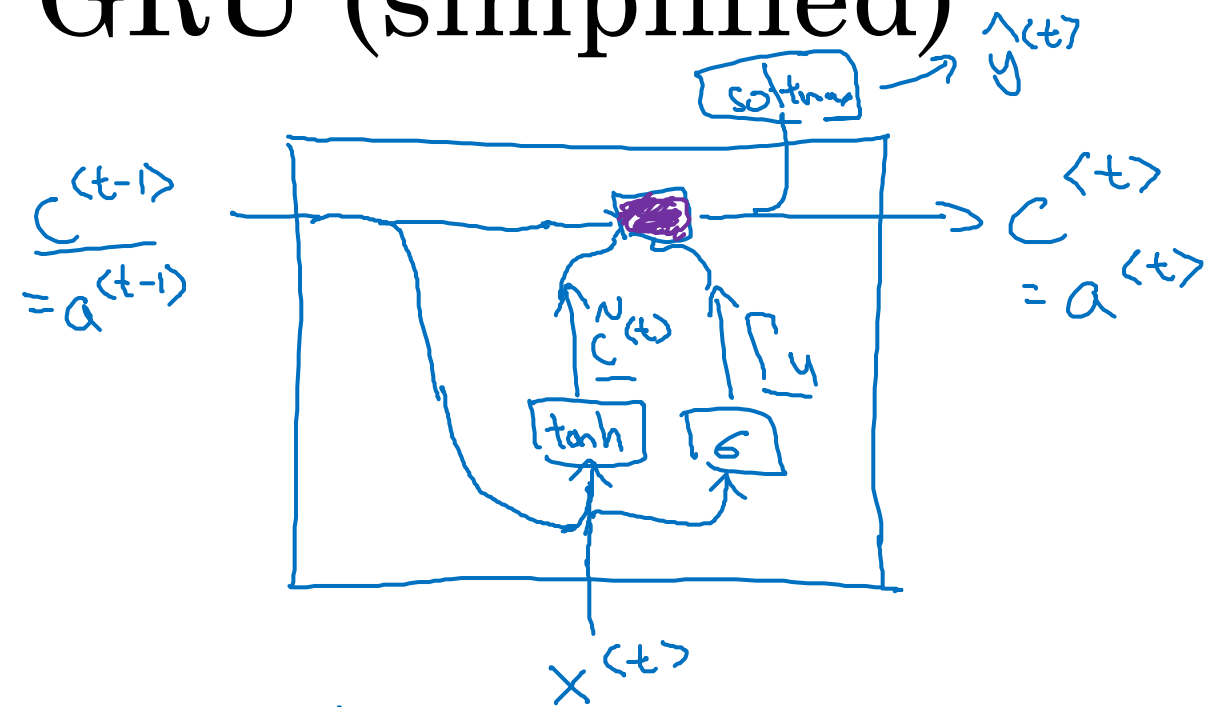
Gated Recurrent Unit (GRU)

RNN unit

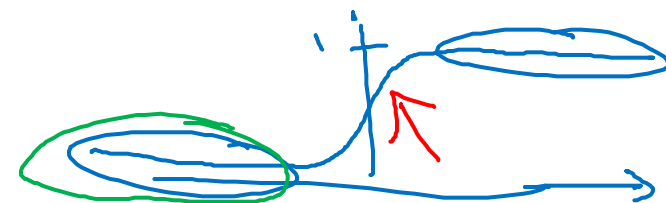


$$\underline{a^{<t>}} = \overset{\substack{\text{tanh} \\ \downarrow}}{g}(\underbrace{W_a[a^{<t-1>}, x^{<t>}]}_{\uparrow} + b_a)$$

GRU (simplified)



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



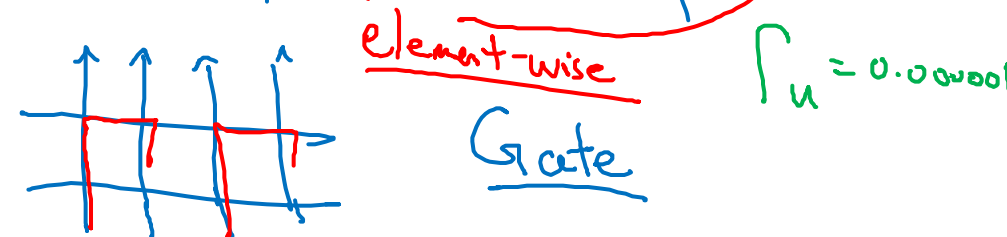
C = memory cell

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

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

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

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



Full GRU

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

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

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

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

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

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

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

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

\uparrow
 Γ_f

LSTM

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

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

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

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

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

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

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

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

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

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

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

LSTM in pictures

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

$$\rightarrow \Gamma_u = \sigma(W_u[\underbrace{a^{<t-1>}, x^{<t>}}_{\text{input}}, b_u])$$

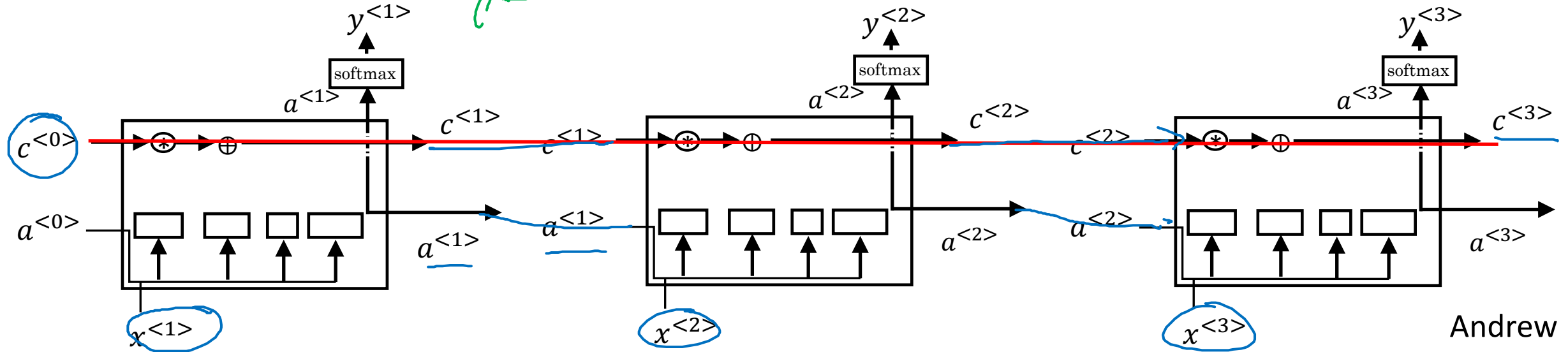
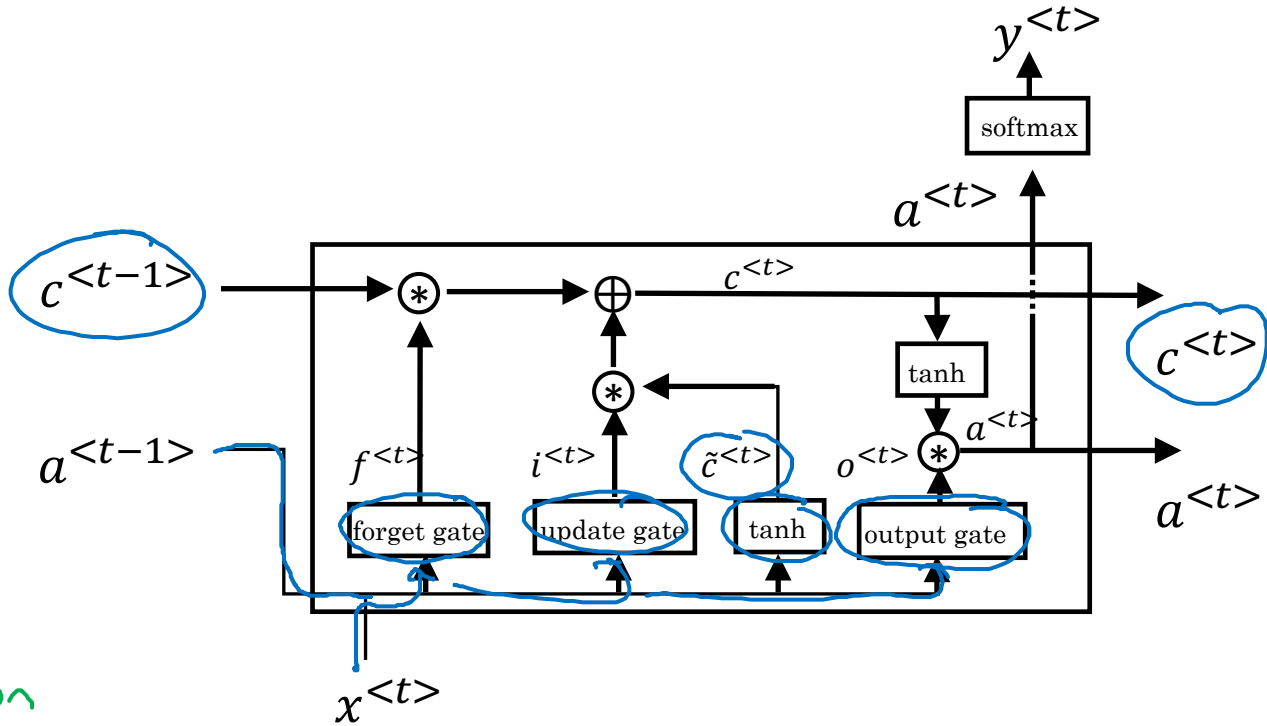
$$\rightarrow \Gamma_f = \sigma(W_f[\underline{a^{<t-1>}, x^{<t>}}] + b_f)$$

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

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

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

peephole
connection





deeplearning.ai

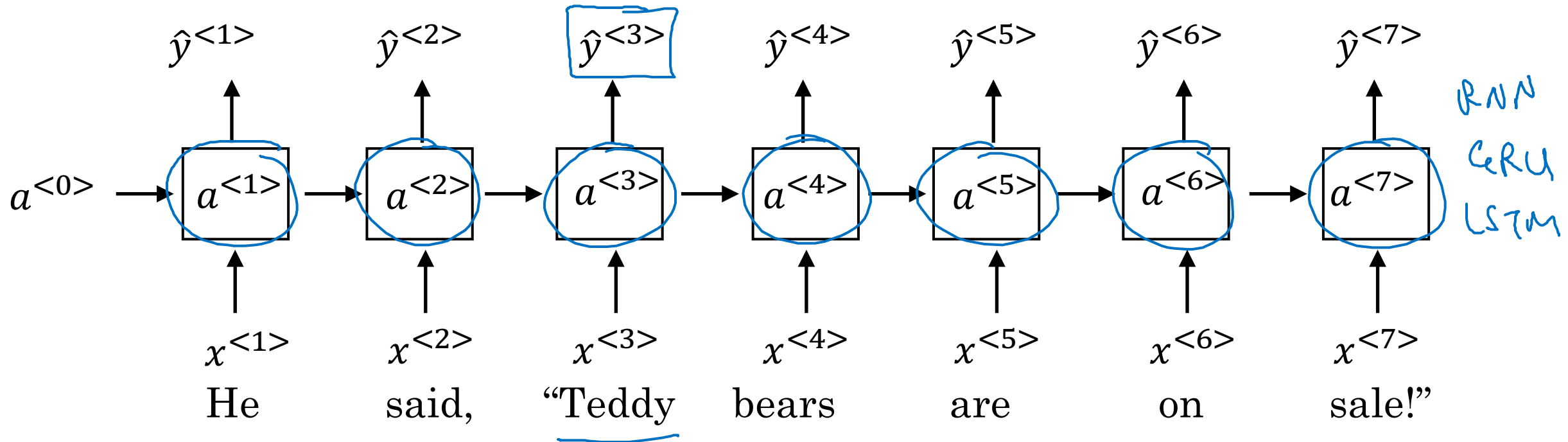
Recurrent Neural Networks

Bidirectional RNN

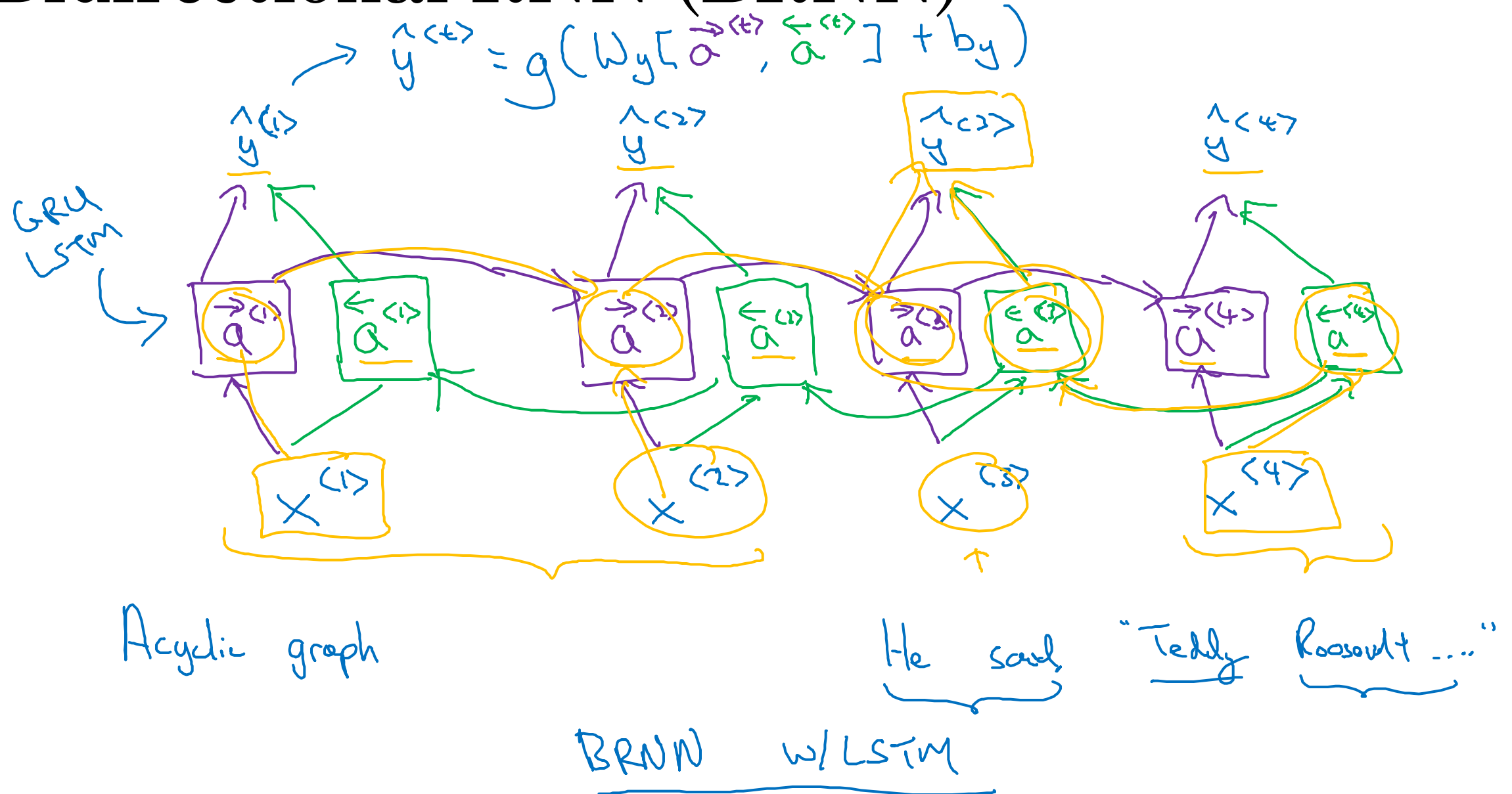
Getting information from the future

He said, “Teddy bears are on sale!”

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



Bidirectional RNN (BRNN)



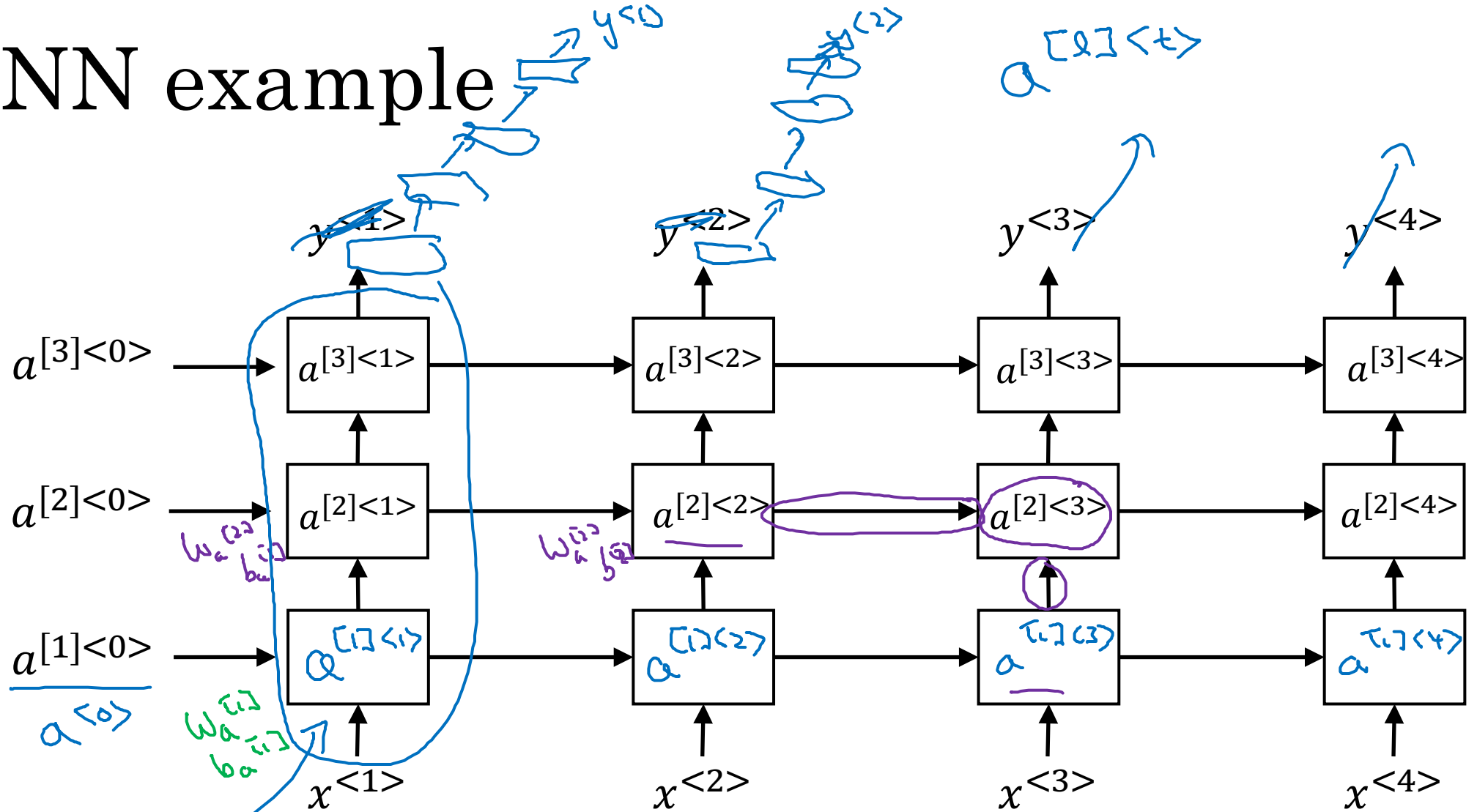
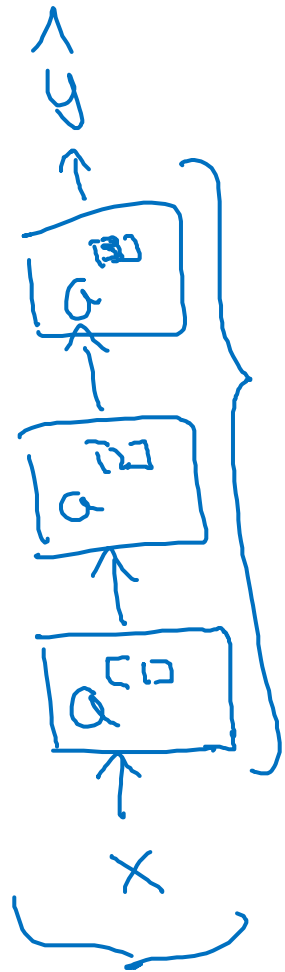


deeplearning.ai

Recurrent Neural Networks

Deep RNNs

Deep RNN example



RNN
GRU
LSTM

BRNN

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

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

NLP and Word Embeddings

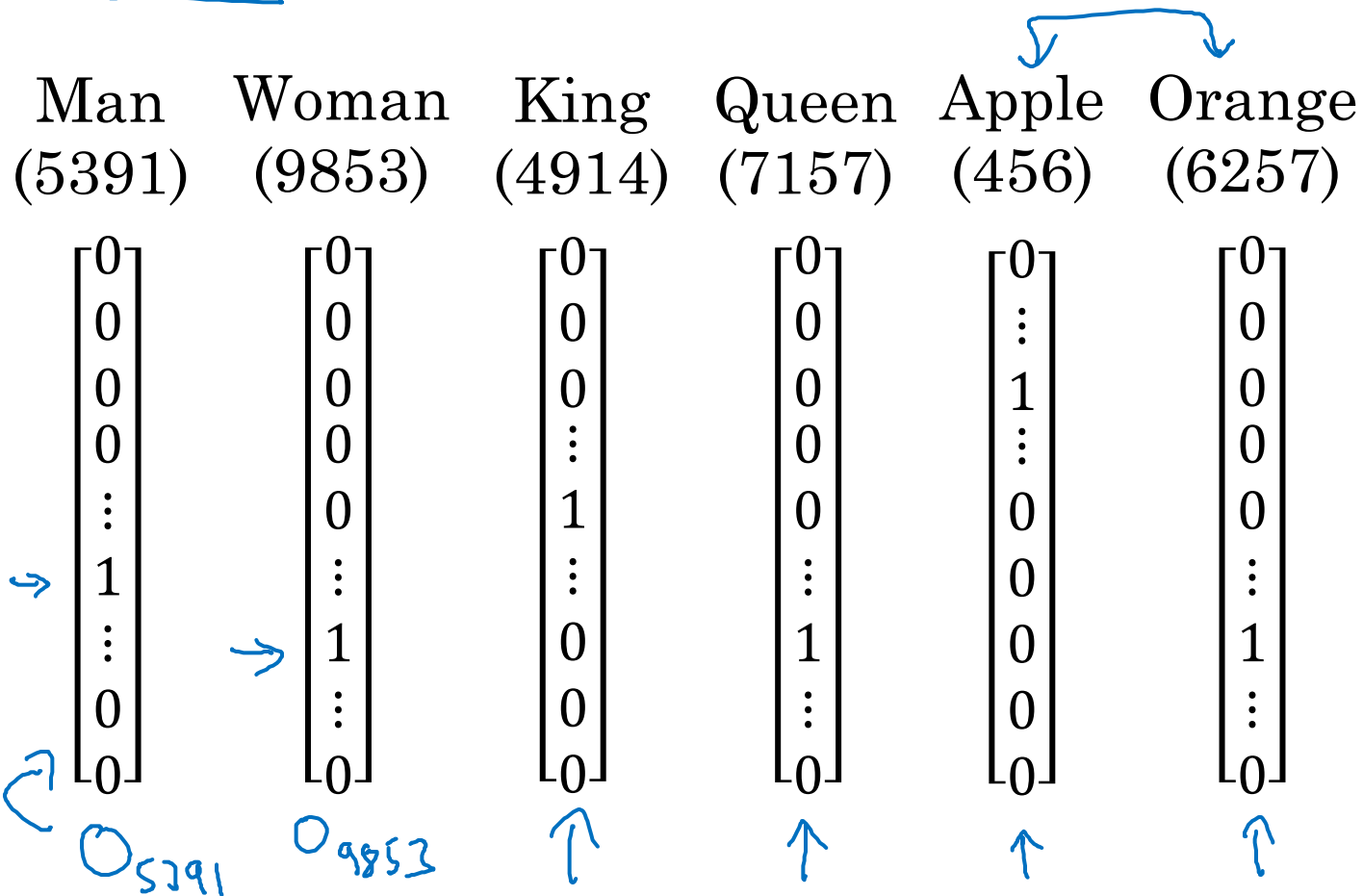
Word representation

Word representation

$V = [a, aaron, \dots, zulu, <UNK>]$

$|V| = 10,000$

1-hot representation



I want a glass of orange juice.

I want a glass of apple ?.

Featurized representation: word embedding

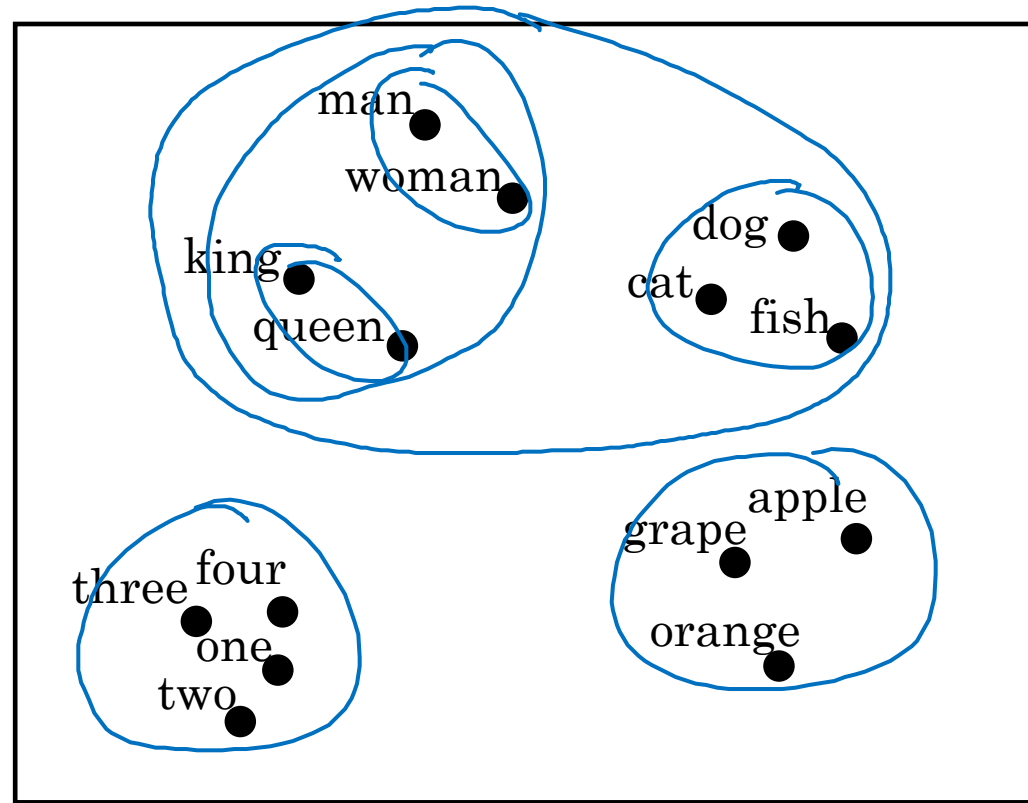
	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	<u>0.93</u>	<u>0.95</u>	-0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	-0.02
Food	0.04	0.01	0.02	0.01	0.95	0.97
...				
size						
cost						
alive						
verb						

I want a glass of orange juice.

I want a glass of apple juice.

Andrew Ng

Visualizing word embeddings

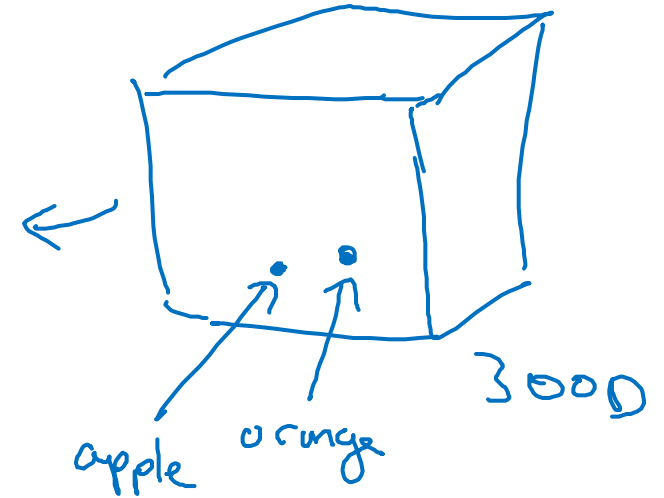


t-SNE

→ 300D



2D



300D

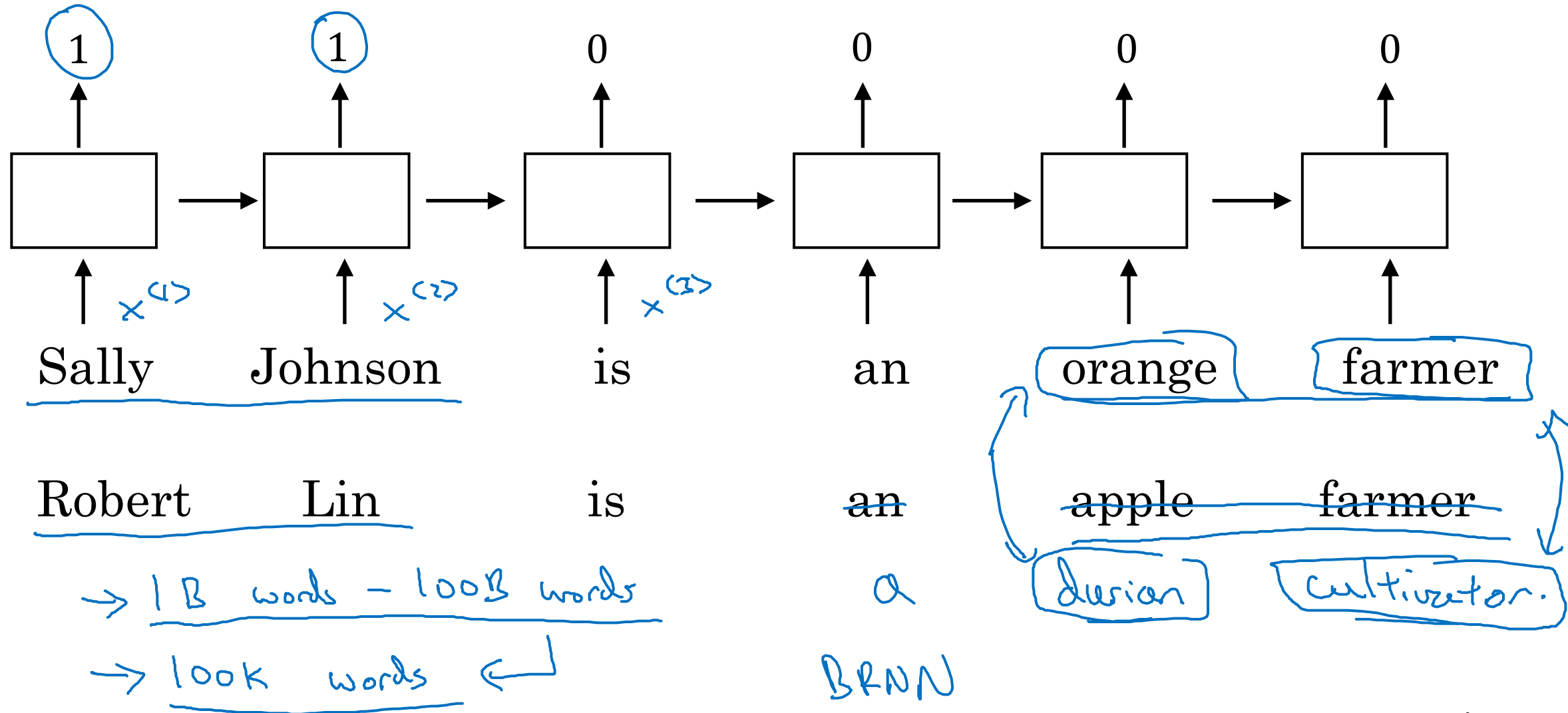


deeplearning.ai


NLP and Word Embeddings

Using word
embeddings

Named entity recognition example



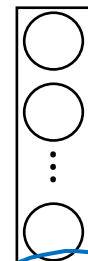
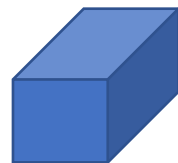
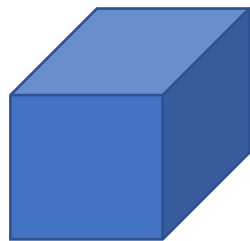
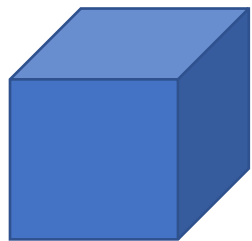
Transfer learning and word embeddings

- 
1. Learn word embeddings from large text corpus. (1-100B words)
(Or download pre-trained embedding online.)
 2. Transfer embedding to new task with smaller training set.
(say, 100k words) → 10,000 → 300
 3. Optional: Continue to finetune the word embeddings with new data.

Relation to face encoding (embedding) 128D



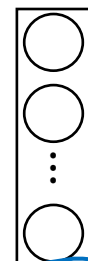
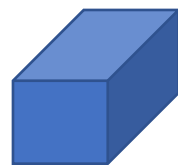
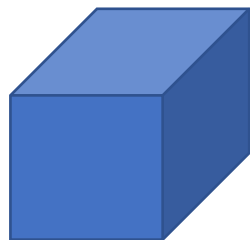
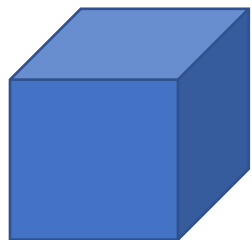
$x^{(i)}$



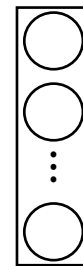
$f(x^{(i)})$



$x^{(j)}$



$f(x^{(j)})$



\hat{y}

$|V| = 10,000$

$e_1, \dots, e_{10,000}$



deeplearning.ai

NLP and Word Embeddings

Properties of word embeddings

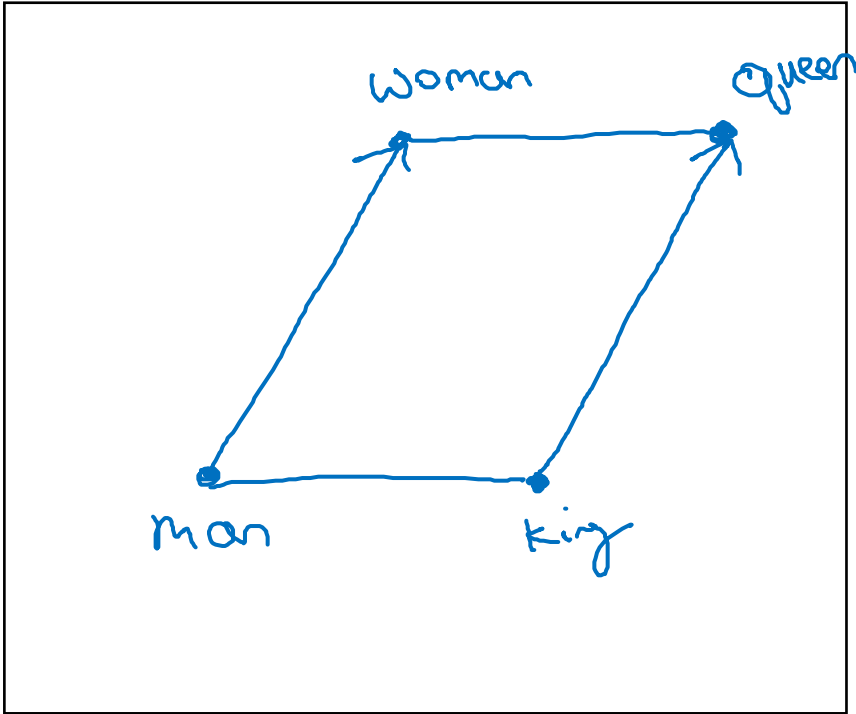
Analogy

	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.70	0.69	0.03	-0.02
Food	0.09	0.01	0.02	0.01	0.95	0.97

$\underbrace{e_{5391}}_{e_{\text{man}}} \rightarrow \underbrace{e_{9853}}_{e_{\text{woman}}} \quad \Leftrightarrow \quad \underbrace{e_{4914}}_{e_{\text{king}}} \rightarrow ? \quad \underbrace{e_{7157}}_{e_{\text{queen}}}$
 $e_{\text{man}} - e_{\text{woman}} \approx e_{\text{king}} - e_{\text{?}}$

$\underline{e_{\text{man}}} - \underline{e_{\text{woman}}} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$
 $\underline{e_{\text{king}}} - \underline{e_{\text{queen}}} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

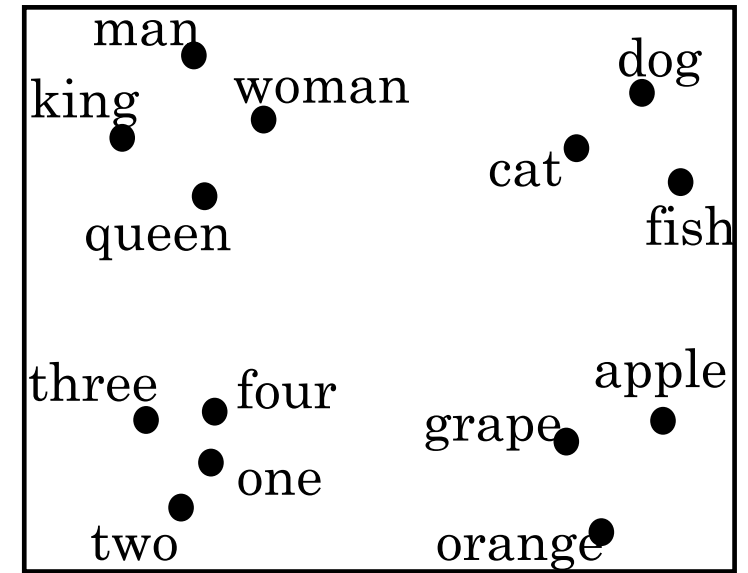
Analogies using word vectors



300 D

Find word w : $\arg \max_w$

3000 \rightarrow 20
↑



t-SNE

$$e_{man} - e_{woman} \approx e_{king} - \cancel{e_w} \quad \underline{e_w}$$

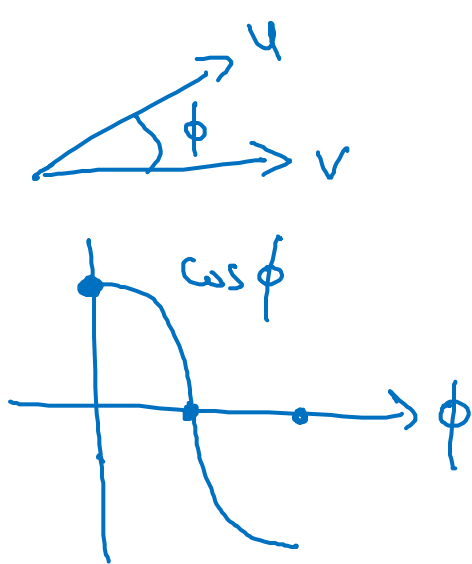
$$\text{Sim}(\underline{e_w}, \underline{e_{king} - e_{man} + e_{woman}})$$

30 - 75%

Cosine similarity

$$\rightarrow \text{sim}(e_w, e_{king} - e_{man} + e_{woman})$$

$$\text{sim}(u, v) = \frac{u^T v}{\|u\|_2 \|v\|_2}$$



$$\|u - v\|^2$$

Man:Woman as Boy:Girl

Ottawa:Canada as Nairobi:Kenya

Big:Bigger as Tall:Taller

Yen:Japan as Ruble:Russia

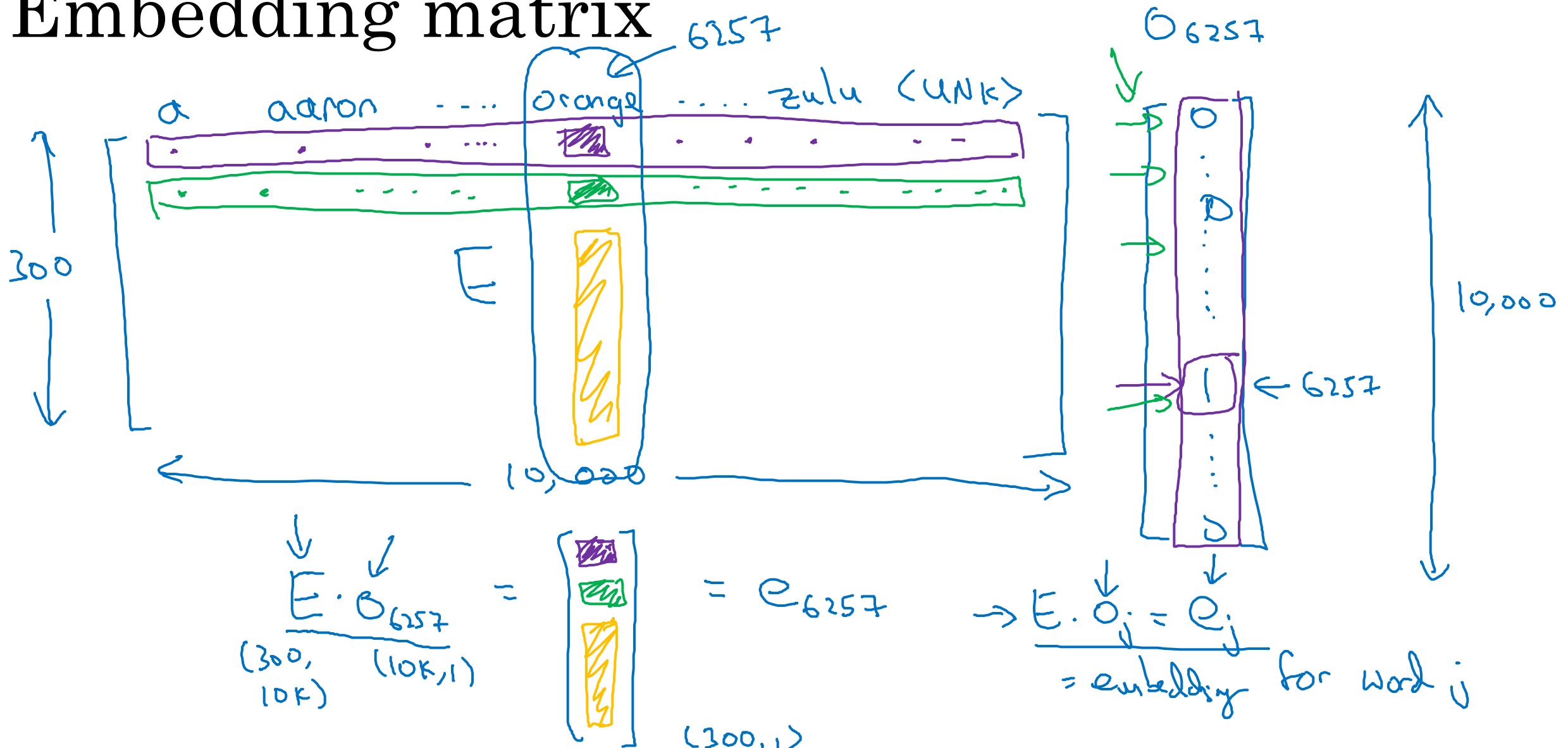


deeplearning.ai

NLP and Word Embeddings

Embedding matrix

Embedding matrix



In practice, use specialized function to look up an embedding.
 $\rightarrow \text{Embedding}$



deeplearning.ai

NLP and Word Embeddings

Learning word embeddings

4



~~1800~~ 1200

Other context/target pairs

I want a glass of orange juice to go along with my cereal.

The diagram illustrates the context and target for the word 'juice'. A purple bracket under 'a glass of orange' is labeled 'context'. A blue bracket under 'juice' is labeled 'target'. A green arrow points from the 'orange' box to the 'juice' target. A blue arrow points from the 'juice' target to the 'context' bracket.

Context: Last 4 words.

- 4 words on left & right
- Last 1 word
- Nearby 1 word

a glass of orange ? to go along with

orange ?

glass ?

skip gram



deeplearning.ai

NLP and Word Embeddings

Word2Vec

Skip-grams

I want a glass of orange juice to go along with my cereal.



Context

orange

orange

orange



Target

juice

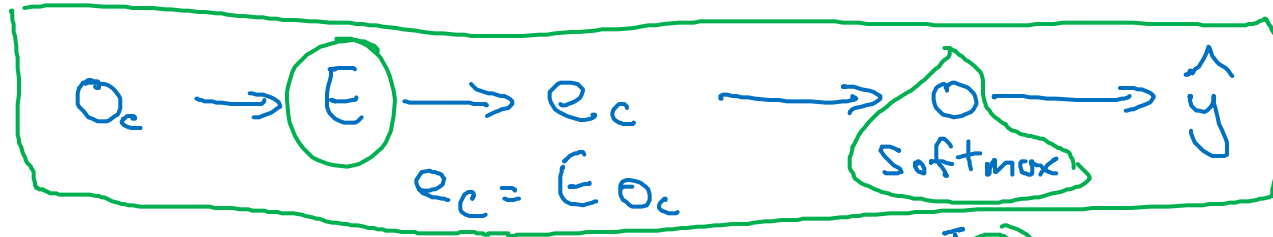
glass

my



Model

Vocab size = 10,000k



Softmax:
$$p(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10,000} e^{\theta_j^T e_c}}$$

θ_t = parameter associated with output t

Loss function (Cross-Entropy Loss):

$$\mathcal{L}(\hat{y}, y) = - \sum_{i=1}^{10,000} y_i \log \hat{y}_i$$

Output vector y (one-hot encoding):

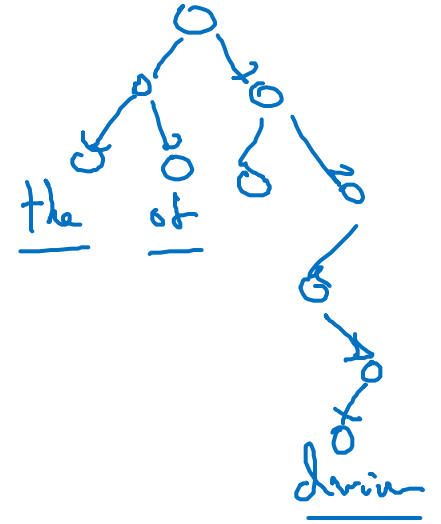
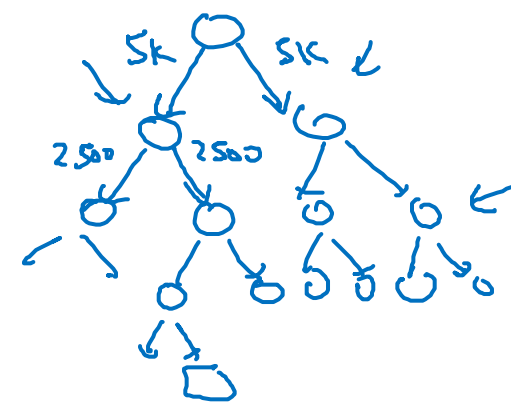
$$y = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \leftarrow 4834$$

Problems with softmax classification

$$\underline{p(t|c)} = \frac{e^{\theta_t^T \underline{e_c}}}{\sum_{j=1}^{10,000} e^{\theta_j^T e_c}}$$

Hierarchical softmax.

$\log |V|$



How to sample the context c ?

→ the, of, a, and, to, ...

→ orange, apple, durian

P_{durian}

t

$c \rightarrow t$

$P(c)$



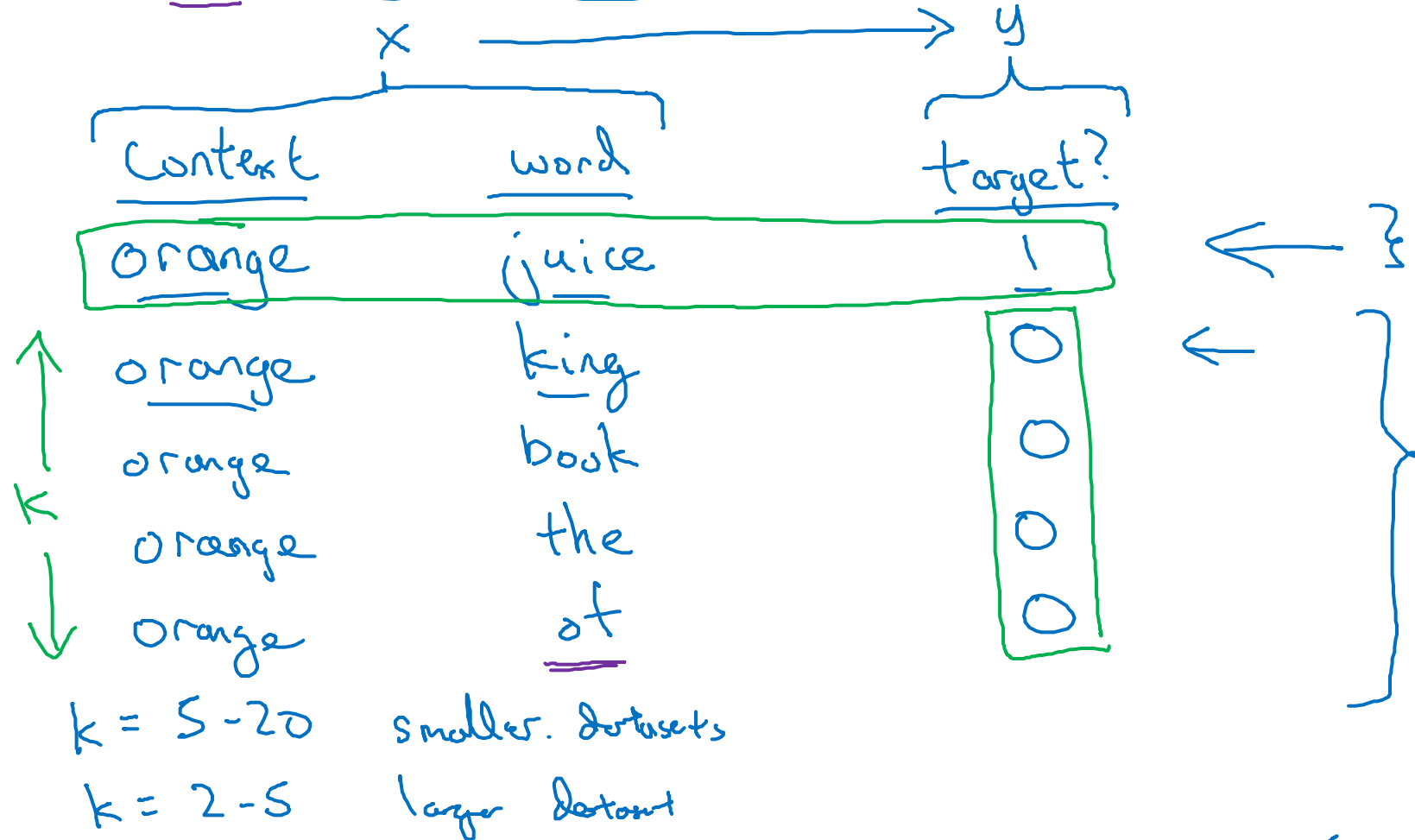
deeplearning.ai

NLP and Word Embeddings

Negative sampling

Defining a new learning problem

I want a glass of orange juice to go along with my cereal.



Model

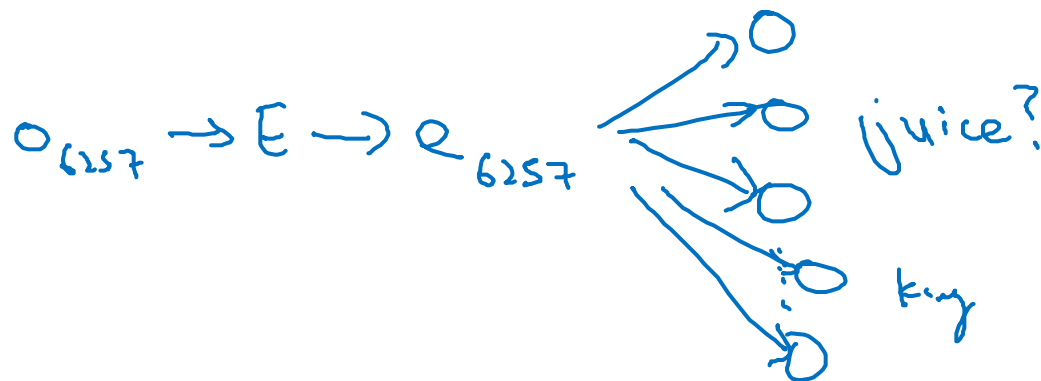
Softmax:
$$p(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10,000} e^{\theta_j^T e_c}}$$

10,000-way softmax

$$P(y=1 | c, t) = \sigma(\theta_t^T e_c) \leftarrow$$

x		y
<u>context</u>	<u>word</u>	<u>target?</u>
orange	juice	1
orange	king	0
orange	book	0
orange	the	0
orange	of	0
\uparrow c	\uparrow t	\uparrow y

Orange
6257



10,000 binary
classification
problem

$k+1$

Selecting negative examples

<u>context</u>	<u>word</u>	<u>target?</u>
orange	juice	1
orange	king	0
orange	book	0
orange	the	0
orange	of	0

↑
t

the, of, and, ...

$$P(w_i) = \frac{f(w_i)^{3/4}}{\sum_{j=1}^{10,000} f(w_j)^{3/4}}$$

$$\frac{1}{|V|}$$

↑



deeplearning.ai

NLP and Word Embeddings

GloVe word vectors

GloVe (global vectors for word representation)

I want a glass of orange juice to go along with my cereal.

c, t

X_{ij} = # times i appears in context of j .

$\begin{matrix} \uparrow & \uparrow \\ c & t \end{matrix}$ $\begin{matrix} \uparrow \\ t \end{matrix}$ $\begin{matrix} \uparrow \\ c \end{matrix}$

$X_{ij} = X_{ji}$ ←

Model

minimize

$$\sum_{i=1}^{10,000} \sum_{j=1}^{10,000} f(x_{ij}) \left(\underbrace{\Theta_i^T e_j}_{\substack{t \quad c \\ \text{"}\Theta_t^T e_c\text{"}}} + b_i + b_j' - \log x_{ij} \right)^2$$

←

0?

weighting
term

$$f(x_{ij}) = 0 \text{ at } x_{ij} = 0.$$

$$0 \log 0 = 0$$

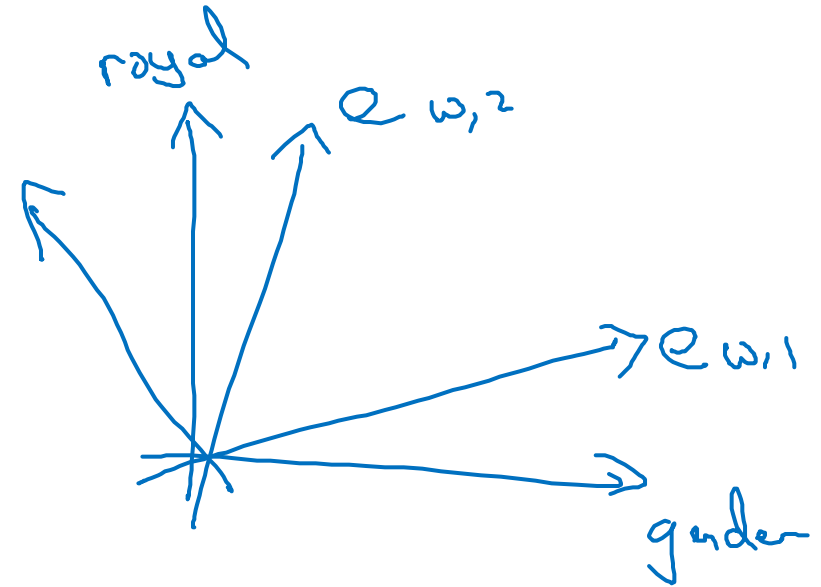
→ this, is, of, a, ...
→ derivation

Θ_i, e_j are symmetric

$$e_w^{(final)} = \frac{e_w + \Theta_w}{2}$$

A note on the featurization view of word embeddings

	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	
Gender	-1	1	-0.95	0.97	←
Royal	0.01	0.02	0.93	0.95	←
Age	0.03	0.02	0.70	0.69	←
Food	0.09	0.01	0.02	0.01	←



$$\text{minimize } \sum_{i=1}^{10,000} \sum_{j=1}^{10,000} f(X_{ij}) (\underbrace{\theta_i^T e_j}_{\text{handwritten}} + b_i - b'_j - \log X_{ij})^2$$

$$\text{handwritten: } (A\theta_i)^T (A^T e_j) = \theta_i^T A^T A e_j$$



deeplearning.ai

NLP and Word Embeddings

Sentiment classification

Sentiment classification problem



The dessert is excellent.



Service was quite slow.



Good for a quick meal, but nothing special.



Completely lacking in good taste,
good service, and good ambience.



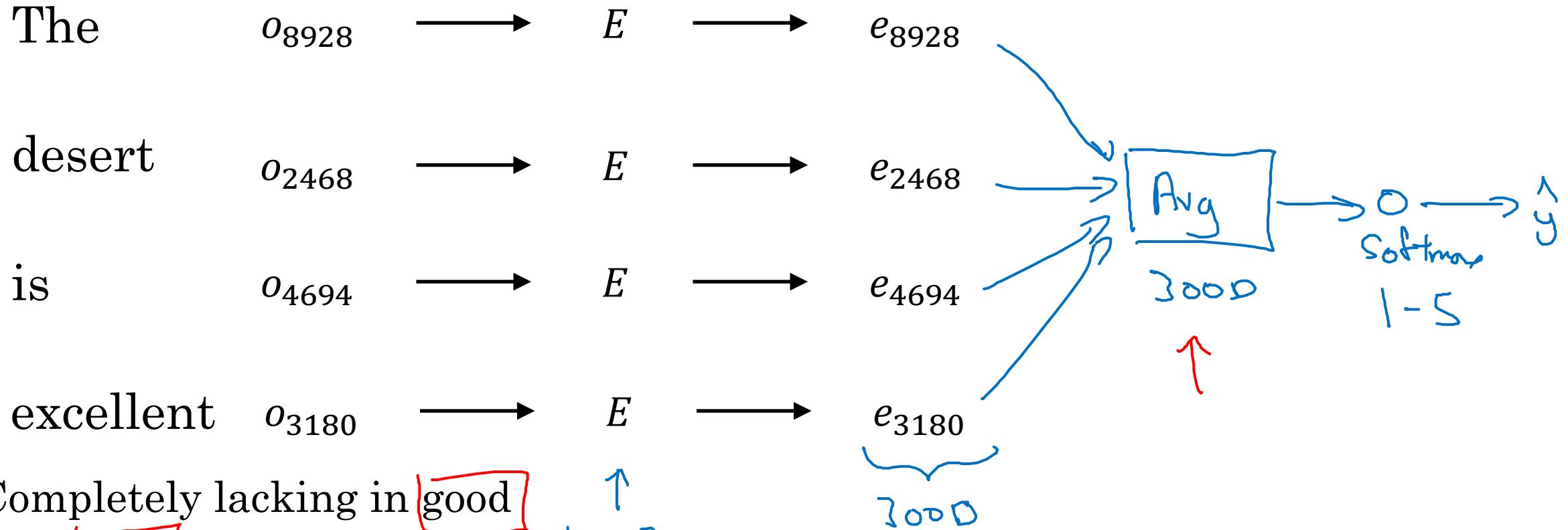
10,000  100,000 words

Simple sentiment classification model

The dessert is excellent



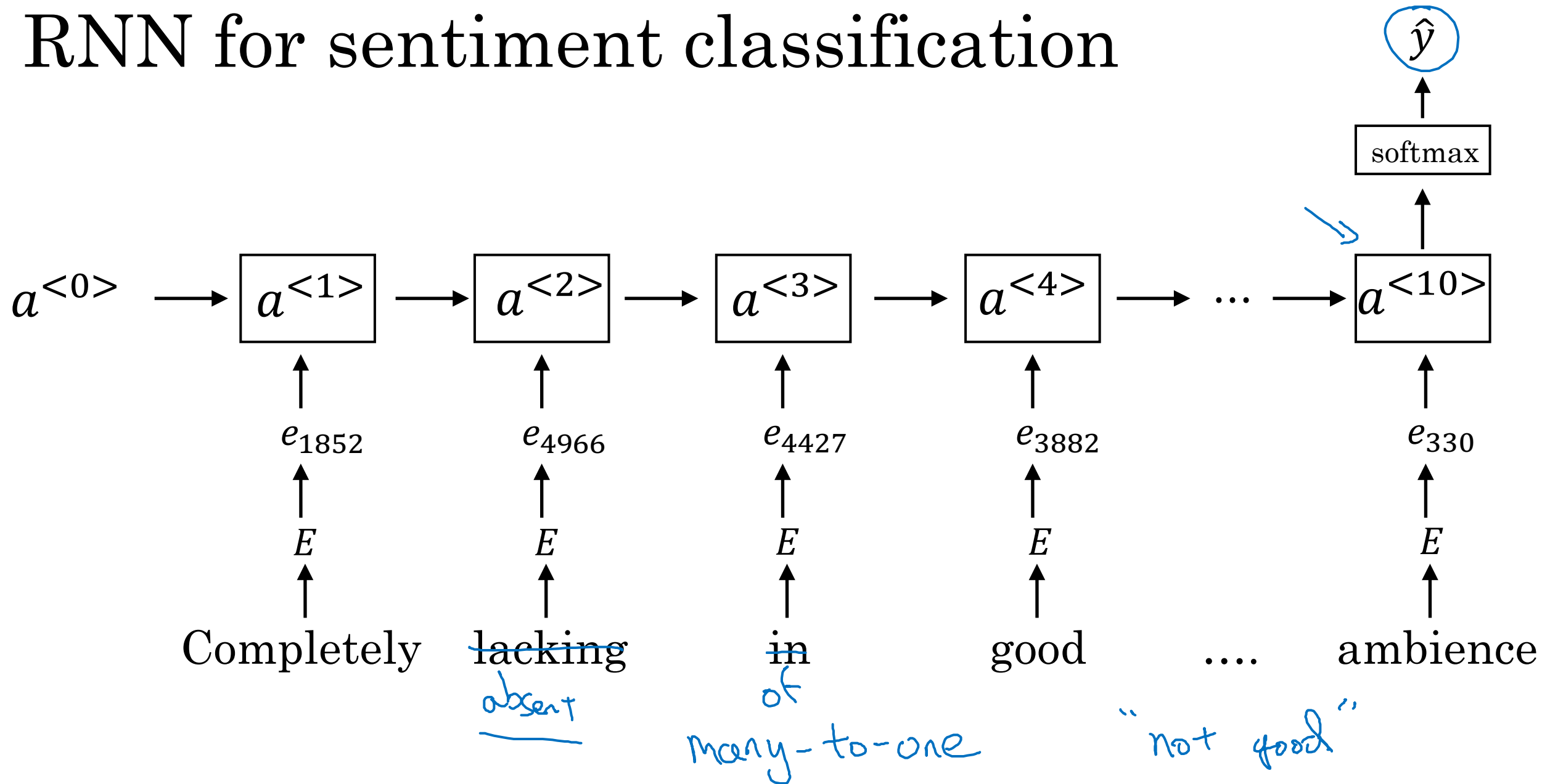
8928 2468 4694 3180



“Completely lacking in good taste, good service, and good ambience.”

↑
100 B
words

RNN for sentiment classification





deeplearning.ai

NLP and Word Embeddings

Debiasing word embeddings

The problem of bias in word embeddings

Man:Woman as King:Queen

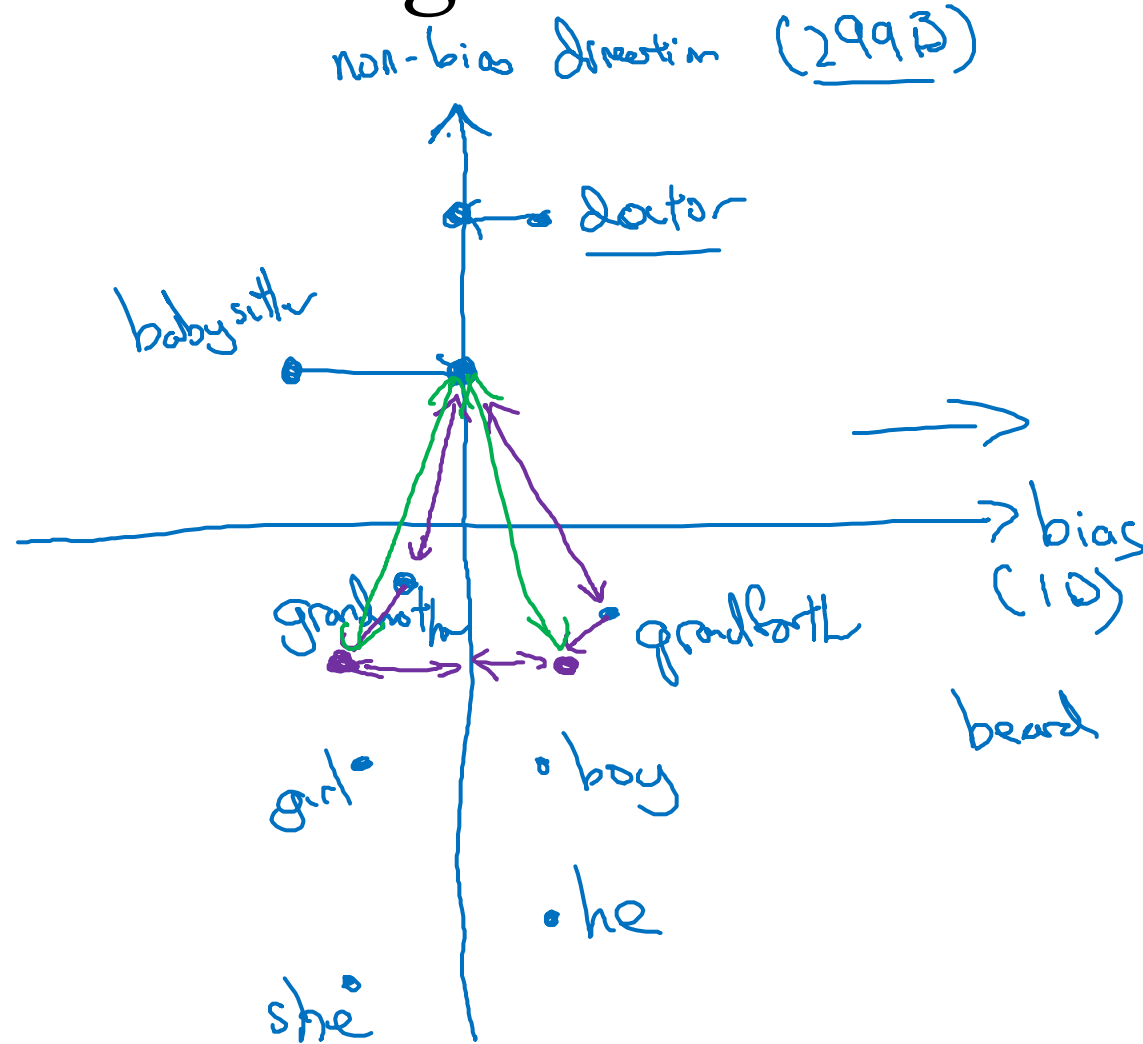
Man:Computer_Programmer as Woman:Homemaker X

Father:Doctor as Mother:Nurse X

Word embeddings can reflect gender, ethnicity, age, sexual orientation, and other biases of the text used to train the model.



Addressing bias in word embeddings



1. Identify bias direction.

$$\begin{cases} e_{he} - e_{she} \\ e_{male} - e_{female} \\ \vdots \end{cases} \rightarrow \text{average}$$

2. Neutralize: For every word that is not definitional, project to get rid of bias.

3. Equalize pairs.

$$\rightarrow \begin{cases} \text{grandmother} - \text{grandfather} \\ \text{girl} - \text{boy} \end{cases}$$

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

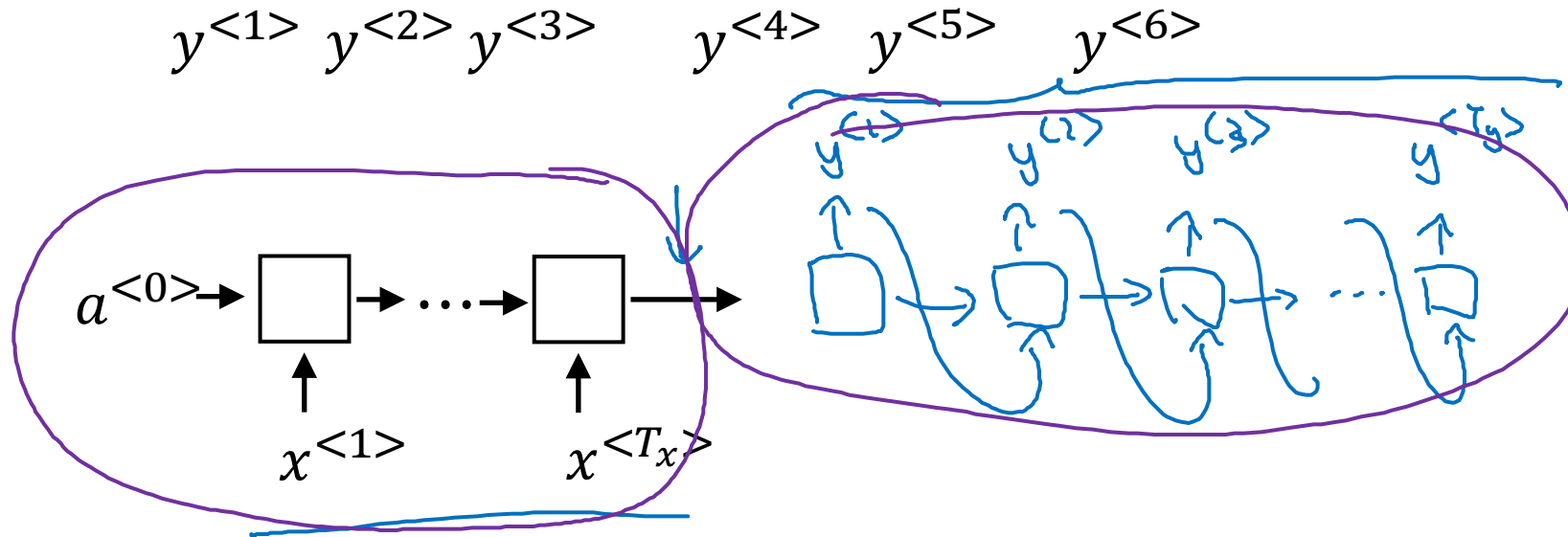
Sequence to sequence models

Basic models

Sequence to sequence model

$x^{<1>}$ $x^{<2>}$ $x^{<3>}$ $x^{<4>}$ $x^{<5>}$
Jane visite l'Afrique en septembre

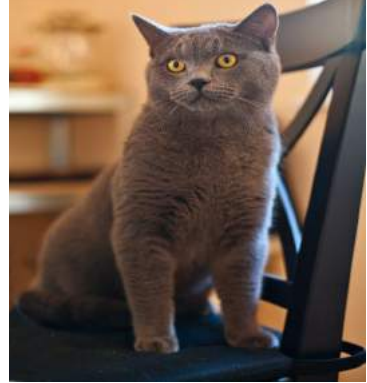
→ Jane is visiting Africa in September.



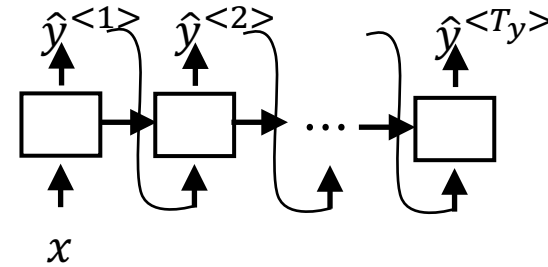
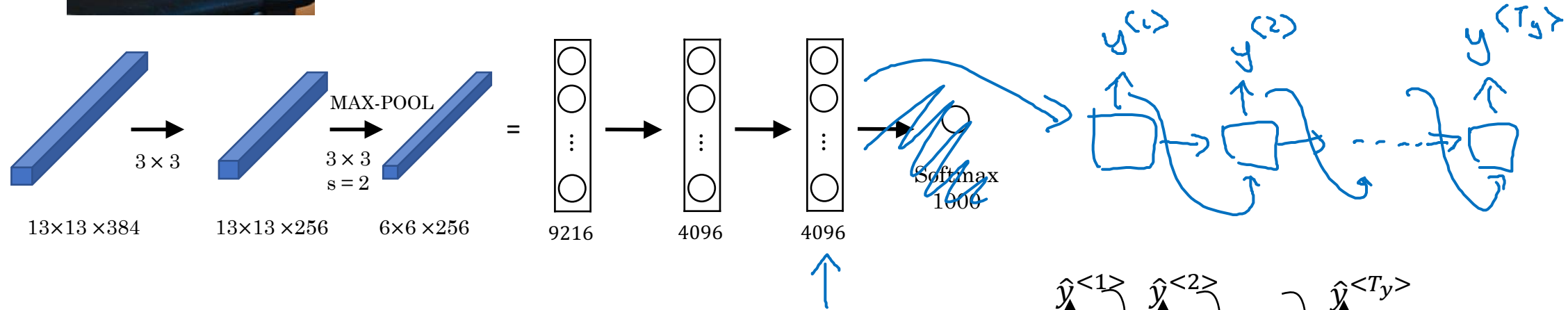
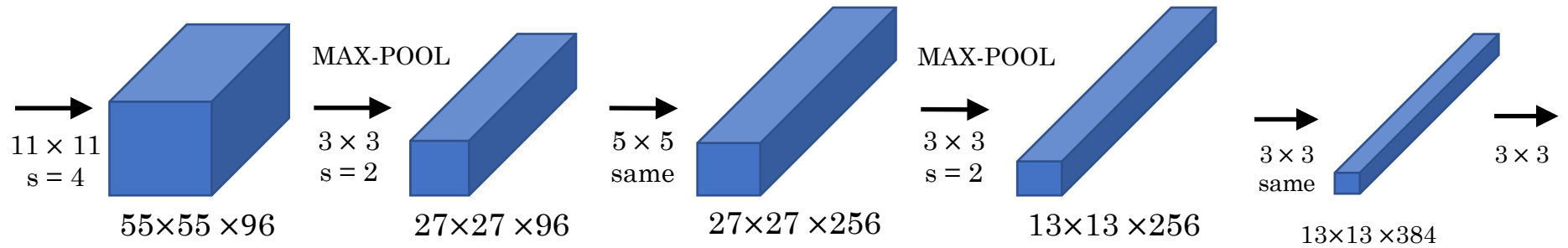
[Sutskever et al., 2014. Sequence to sequence learning with neural networks] ↩

[Cho et al., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation] ↩

Image captioning



$y^{<1>}$ $y^{<2>}$ $y^{<3>}$ $y^{<4>}$ $y^{<5>}$ $y^{<6>}$ }
 A cat sitting on a chair



[Mao et. al., 2014. Deep captioning with multimodal recurrent neural networks]

[Vinyals et. al., 2014. Show and tell: Neural image caption generator]

[Karpathy and Li, 2015. Deep visual-semantic alignments for generating image descriptions]

Andrew Ng



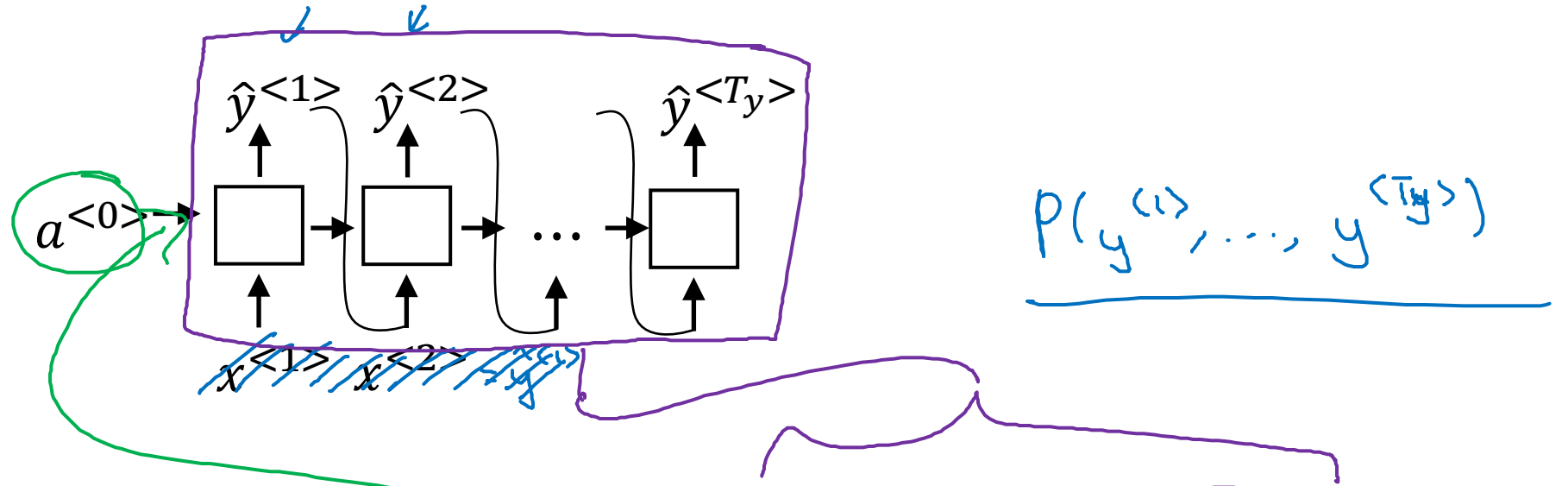
deeplearning.ai

Sequence to sequence models

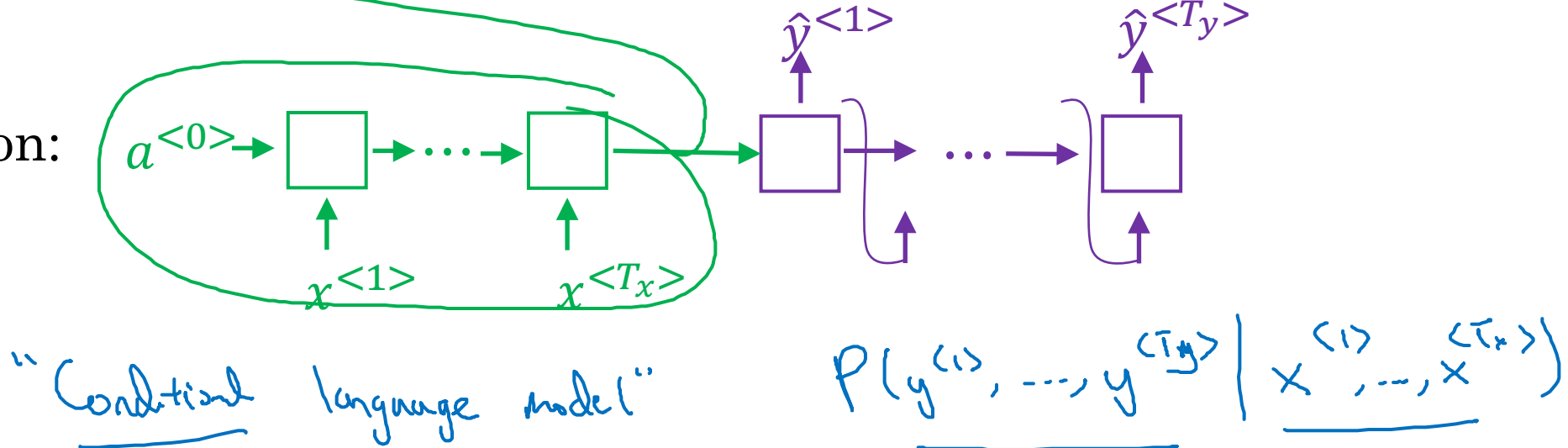
Picking the most likely sentence

Machine translation as building a conditional language model

Language model:



Machine translation:



Finding the most likely translation

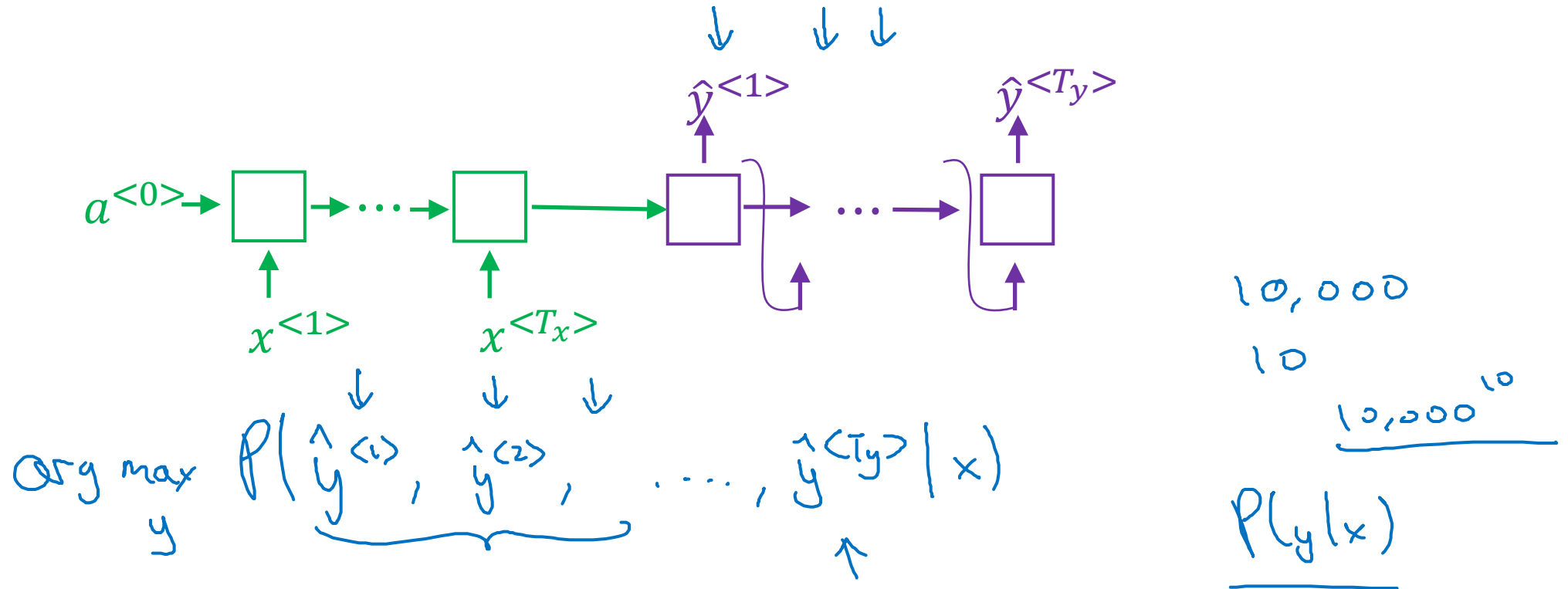
Jane visite l'Afrique en septembre.

$$P(y^{<1>}, \dots, y^{<T_y>} | x)$$

- Jane is visiting Africa in September.
- Jane is going to be visiting Africa in September.
- In September, Jane will visit Africa.
- Her African friend welcomed Jane in September.

$$\arg \max_{y^{<1>}, \dots, y^{<T_y>}} \underline{P(y^{<1>}, \dots, y^{<T_y>} | x)}$$

Why not a greedy search?



→ Jane is visiting Africa in September.

→ Jane is going to be visiting Africa in September.

$$P(\text{Jane is going} | x) > P(\text{Jane is visiting} | x)$$



deeplearning.ai

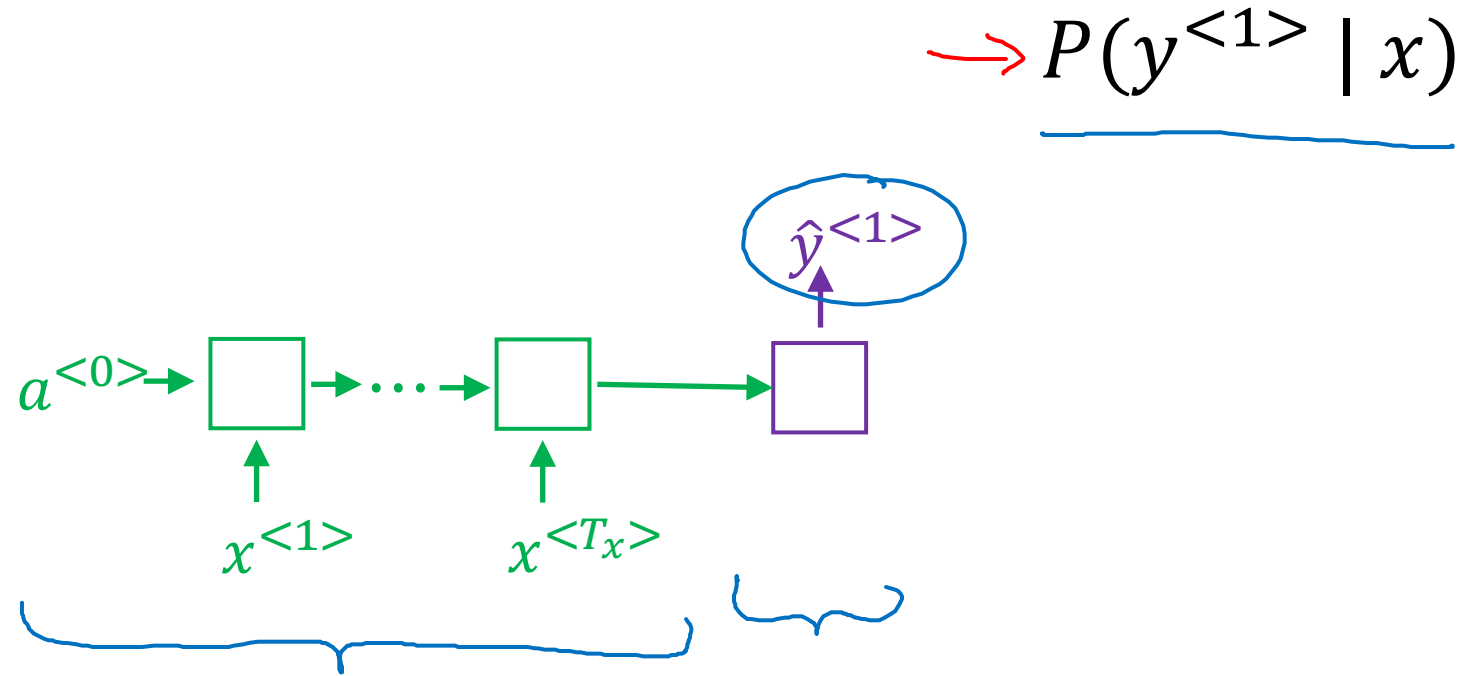
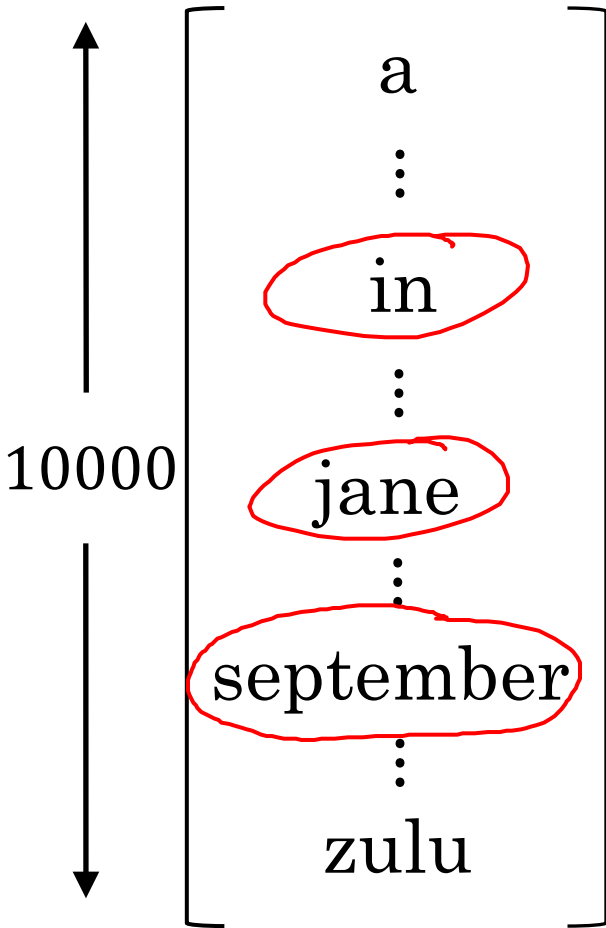
Sequence to sequence models

Beam search

Beam search algorithm

$B = 3$ (beam width)

Step 1

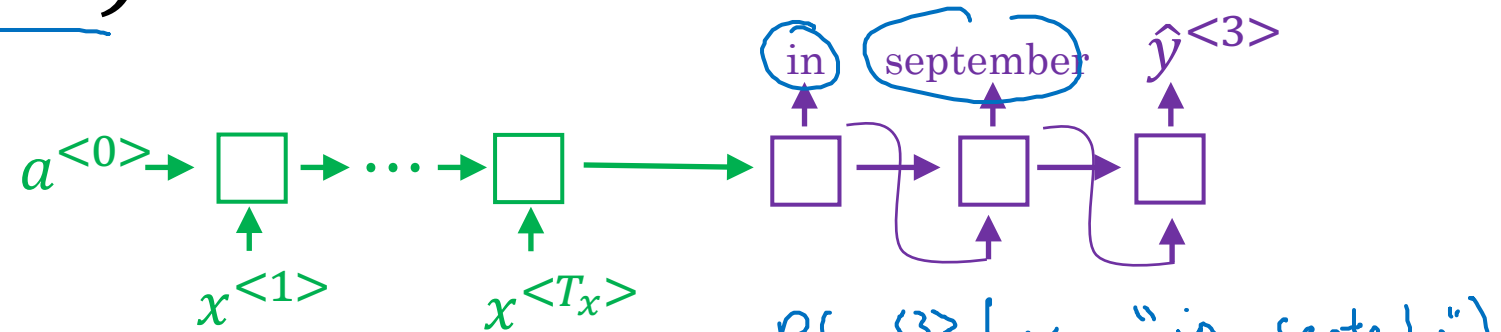


Beam search ($B = 3$)

$B=1 \rightsquigarrow$ greedy search

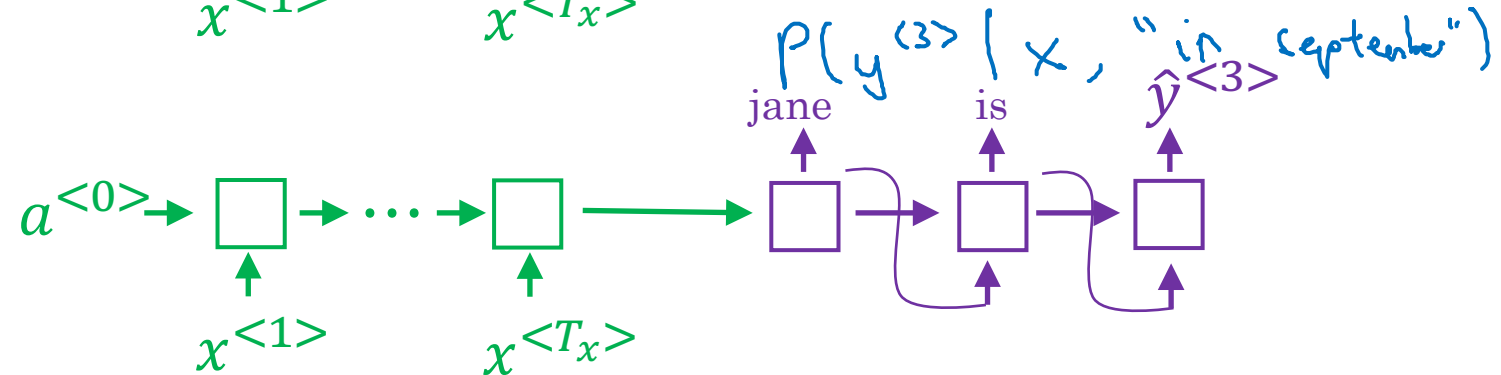
in september

a
aaron
jane
zulu



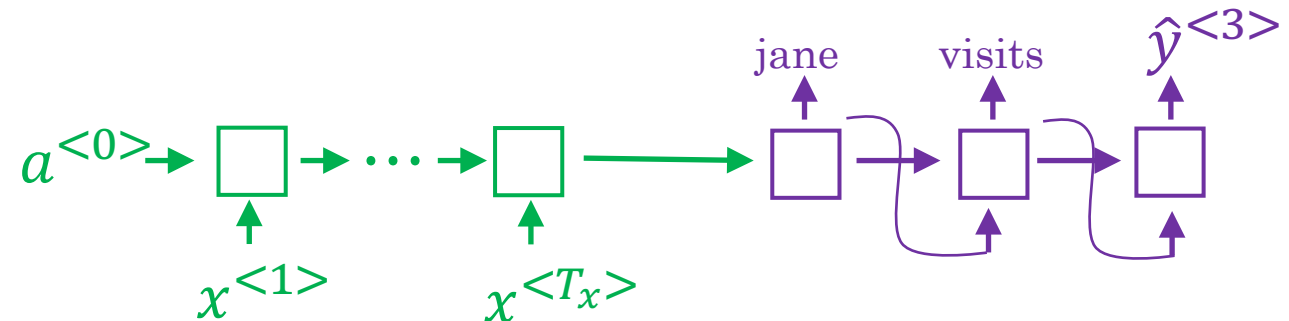
jane is

a
visits
zulu



jane visits

a
africa
zulu



$$P(y^{<1>}, y^{<2>} | x)$$

jane visits africa in september. <EOS>



deeplearning.ai

Sequence to sequence models

Refinements to beam search

Length normalization

$$P(y^{(1)} \dots y^{(T_y)} | x) = \frac{P(y^{(1)} | x) P(y^{(2)} | x, y^{(1)}) \dots}{P(y^{(T_y)} | x, y^{(1)}, \dots, y^{(T_y-1)})}$$

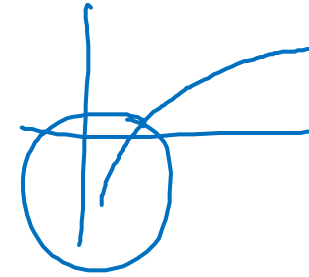
$$\arg \max_y \prod_{t=1}^{T_y} P(y^{(t)} | x, y^{(1)}, \dots, y^{(t-1)})$$

log

$$\arg \max_y \sum_{t=1}^{T_y} \log P(y^{(t)} | x, y^{(1)}, \dots, y^{(t-1)}) \leftarrow$$

$T_y = 1, 2, 3, \dots, 30.$

$$\rightarrow \frac{1}{T_y^\alpha} \sum_{t=1}^{T_y} \log P(y^{(t)} | x, y^{(1)}, \dots, y^{(t-1)})$$



$$\log P(y|x) \leftarrow$$

$$P(y|x) \leftarrow$$

$$\underline{\alpha = 0.7}$$

$$\underline{\alpha = 1}$$

$$\underline{\alpha = 0}$$

Beam search discussion

Beam width B?

$1 \rightarrow 3 \rightarrow 10, \quad 100, \quad 1000 \rightarrow 3000$

large B: better result, slower
small B: worse result, faster

Unlike exact search algorithms like BFS (Breadth First Search) or DFS (Depth First Search), Beam Search runs faster but is not guaranteed to find exact maximum for $\arg \max_y P(y|x)$.



deeplearning.ai

Sequence to sequence models

Error analysis on beam search

Example

Jane visite l'Afrique en septembre.

→ RNN

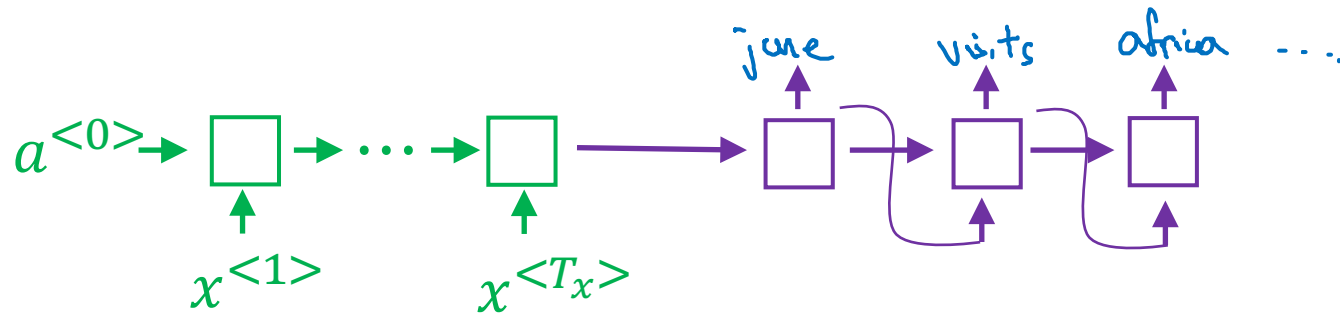
→ Beam Search

BT

Human: Jane visits Africa in September. (y^*)

Algorithm: Jane visited Africa last September. (\hat{y}) ←

RNN computes $P(y^*|x) \geq P(\hat{y}|x)$



Error analysis on beam search

Human: Jane visits Africa in September. (y^*)

$$P(y^*|x)$$

$$P(\hat{y}|x)$$

Algorithm: Jane visited Africa last September. (\hat{y})

Case 1: $P(y^*|x) > P(\hat{y}|x)$ \leftarrow

$$\arg \max_y P(y|x)$$

Beam search chose \hat{y} . But y^* attains higher $P(y|x)$.

Conclusion: Beam search is at fault.

Case 2: $P(y^*|x) \leq P(\hat{y}|x)$ \leftarrow

y^* is a better translation than \hat{y} . But RNN predicted $P(y^*|x) < P(\hat{y}|x)$.

Conclusion: RNN model is at fault.

Error analysis process

Human	Algorithm	$P(y^* x)$	$P(\hat{y} x)$	At fault?
Jane visits Africa in September. - - - ...	Jane visited Africa last September. - - - ...	$\frac{2 \times 10^{-10}}{\text{---}}$ ---	$\frac{1 \times 10^{-10}}{\text{---}}$ ---	<div>B</div> <div>R</div> <div>R</div> <div>R</div> <div>R</div> <div>...</div>

Figures out what fraction of errors are “due to” beam search vs. RNN model



deeplearning.ai

Sequence to sequence models

Bleu score (optional)

Evaluating machine translation

French: Le chat est sur le tapis.

Bleu
bilingual evaluation understudy

Reference 1: The cat is on the mat. ←

Reference 2: There is a cat on the mat. ←

MT output: the the the the the the the.

Precision:

Modified precision:

Bleu score on bigrams

Example: Reference 1: The cat is on the mat. ←

Reference 2: There is a cat on the mat. ←

MT output: The cat the cat on the mat. ←

	Count	Count _{clip}	
the cat	2 ←	1 ←	
cat the	1 ←	0	4
cat on	1 ←	1 ←	—
on the	1 ←	1 ←	6
the mat	1 ←	1 ←	
	↑		

Bleu score on unigrams

Example: Reference 1: The cat is on the mat.

Reference 2: There is a cat on the mat.

→ MT output: The cat the cat on the mat. (\hat{y})

$$P_1, P_2 = \underline{1.0}$$

$$p_1 = \frac{\sum_{unigram \in \hat{y}} \text{count}_{clip}(unigram)}{\sum_{unigram \in \hat{y}} \text{count}(unigram)}$$

Handwritten notes: \hat{y} is written above the summation indices. "unigram" is written below the denominator summation. "count(unigram)" is written below the denominator summation.

$$p_n = \frac{\sum_{ngram \in \hat{y}} \text{count}_{clip}(ngram)}{\sum_{ngram \in \hat{y}} \text{count}(ngram)}$$

Handwritten notes: "n-gram" is written above the summation indices. "count(n-gram)" is written below the denominator summation.

Bleu details

p_n = Bleu score on n-grams only

p_1, p_2, p_3, p_4

Combined Bleu score: $BP \exp\left(\frac{1}{4} \sum_{n=1}^4 p_n\right)$

BP = brevity penalty

$$BP = \begin{cases} 1 & \text{if } \underline{MT_output_length} > \underline{reference_output_length} \\ \exp(1 - MT_output_length/reference_output_length) & \text{otherwise} \end{cases}$$

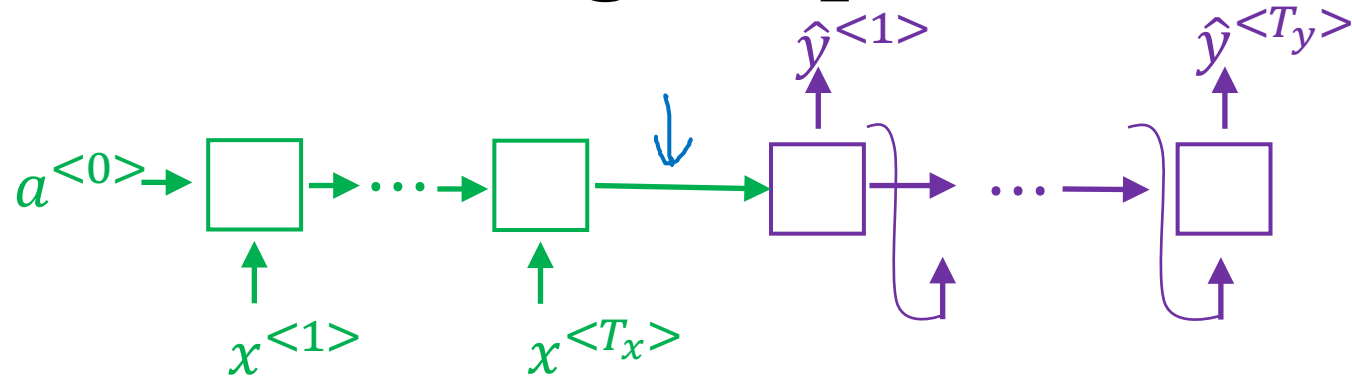


deeplearning.ai

Sequence to sequence models

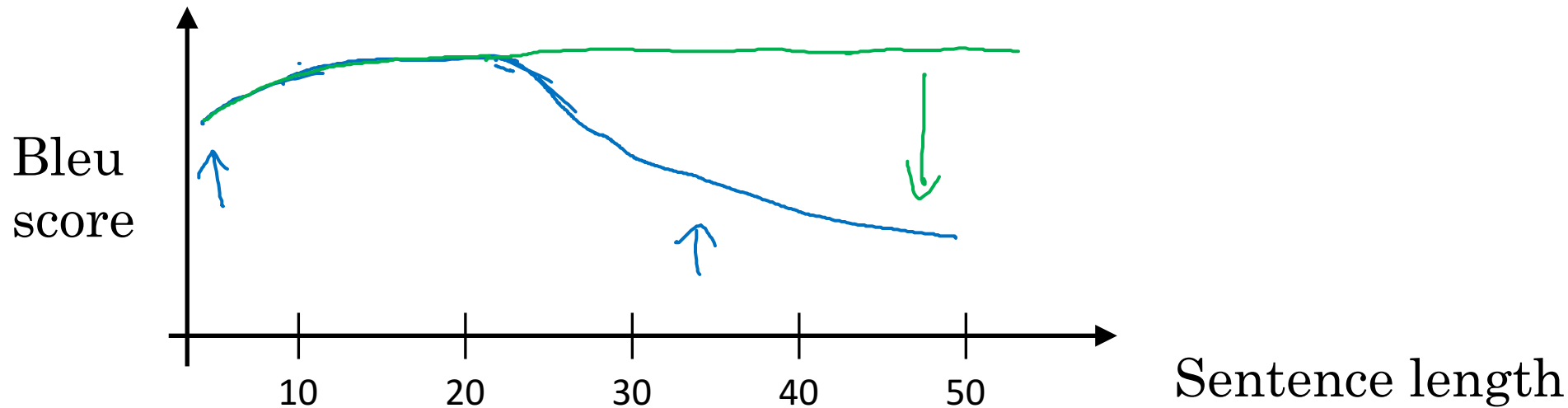
Attention model intuition

The problem of long sequences

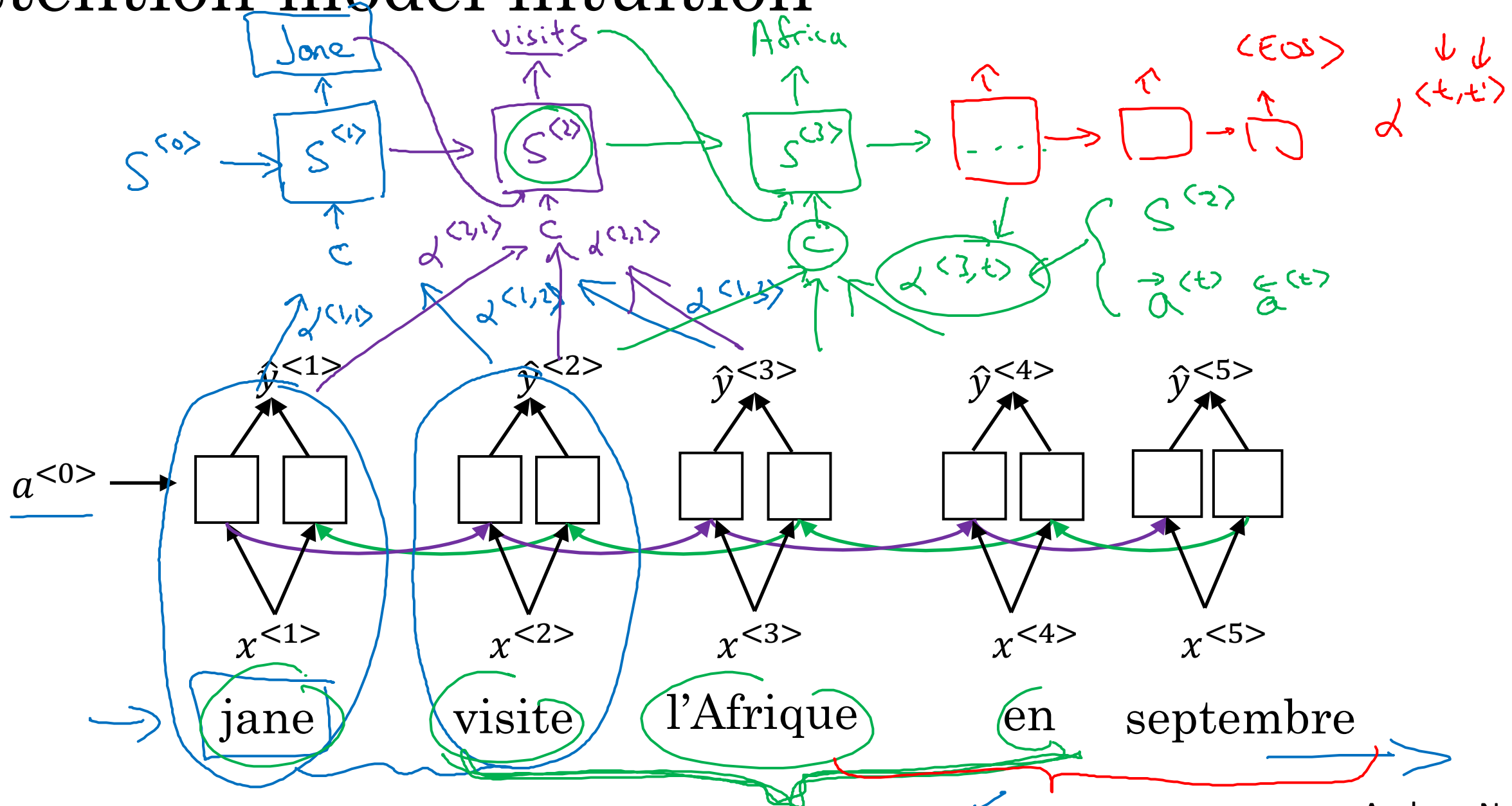


Jane s'est rendue en Afrique en septembre dernier, a apprécié la culture et a rencontré beaucoup de gens merveilleux; elle est revenue en parlant comment son voyage était merveilleux, et elle me tente d'y aller aussi.

Jane went to Africa last September, and enjoyed the culture and met many wonderful people; she came back raving about how wonderful her trip was, and is tempting me to go too.



Attention model intuition





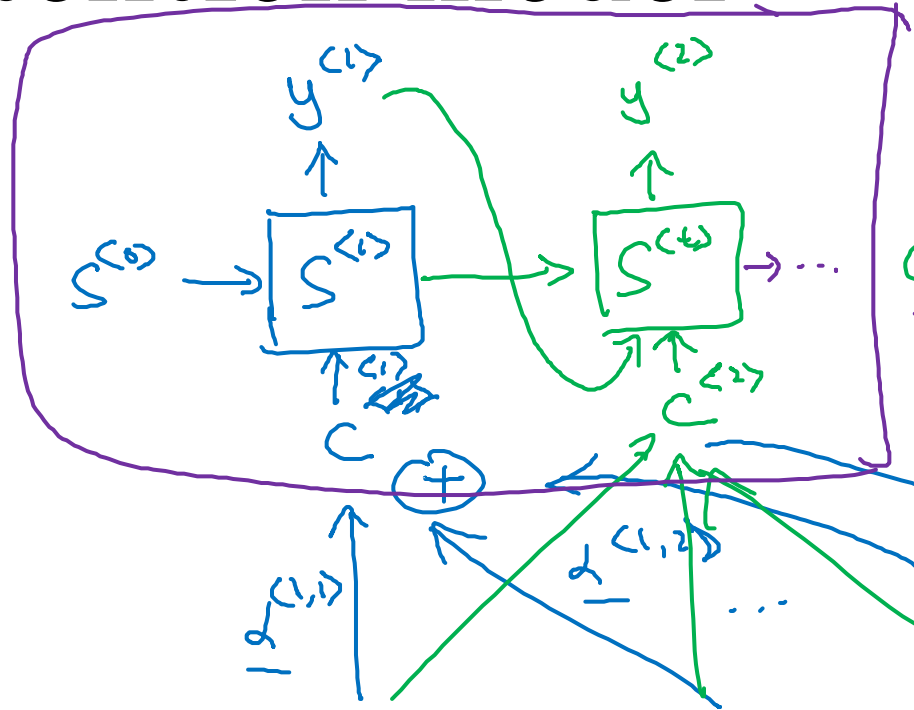
deeplearning.ai

Sequence to sequence models

Attention model

Attention model

$\alpha^{(t,t')}$ = amount of 'attention' $y^{(t)}$ should pay to $a^{(t')}$.

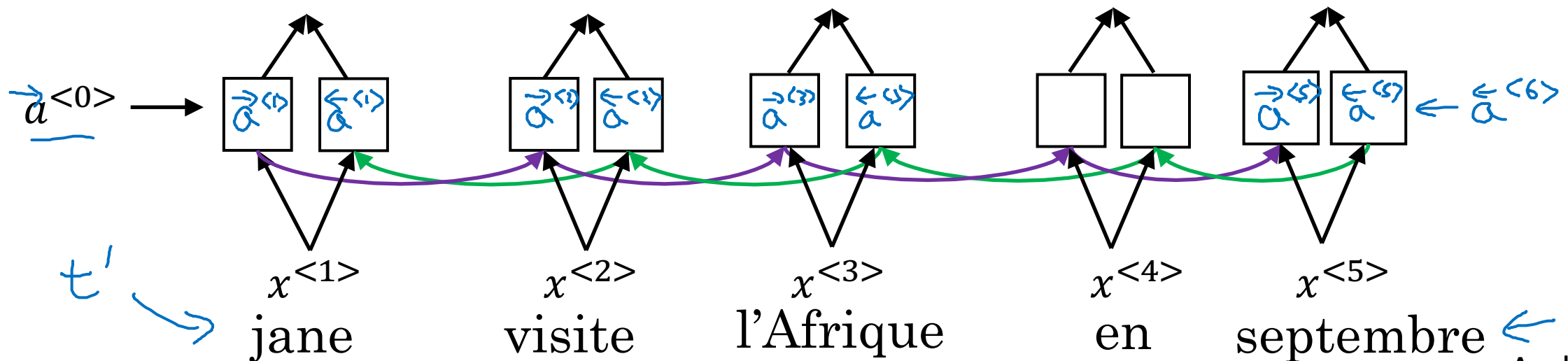


$$c^{(2)} = \sum_{t'} \alpha^{(2,t')} s^{(t')}$$

$$a^{(t')} = (\vec{a}^{(t')}, \leftarrow a^{(t')})$$

$$\sum_{t'} \alpha^{(1,t')} = 1$$

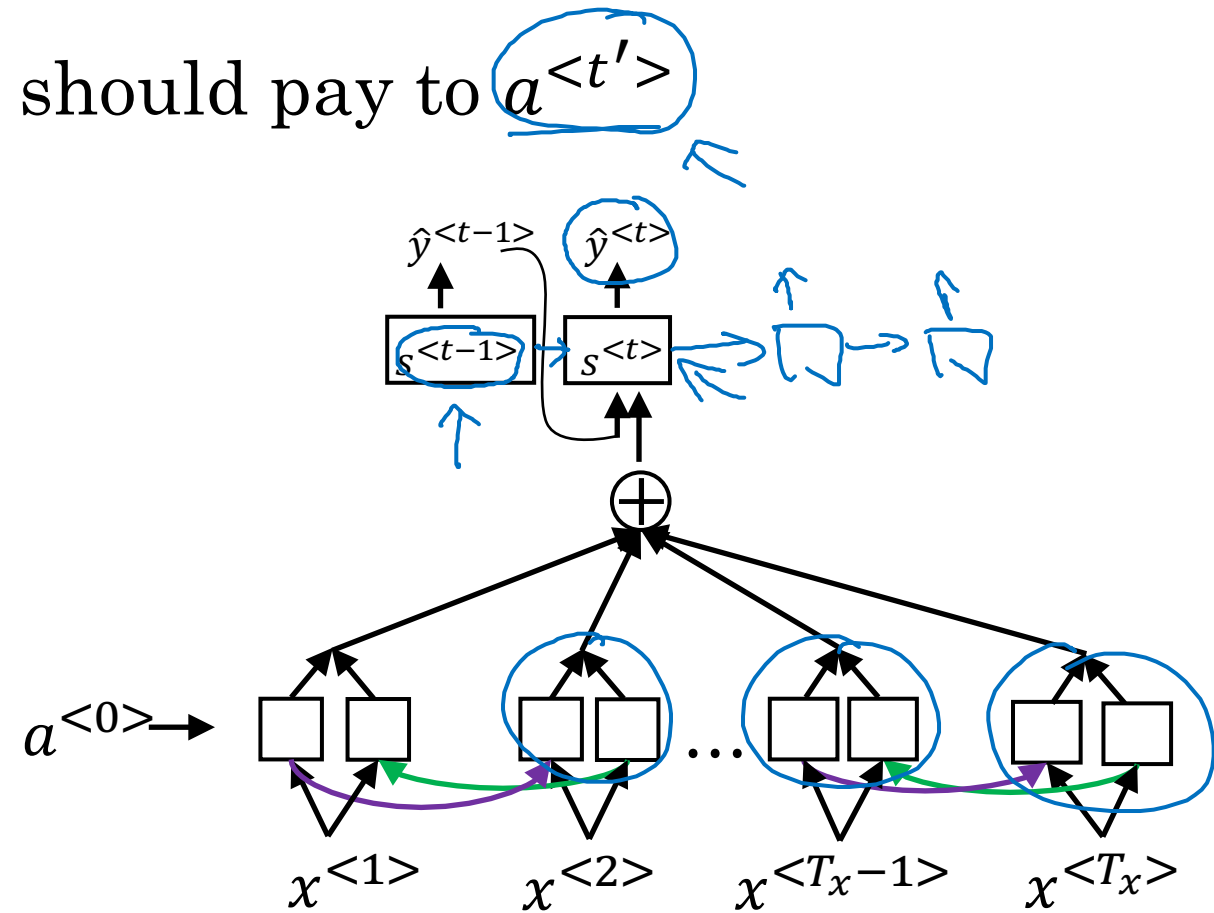
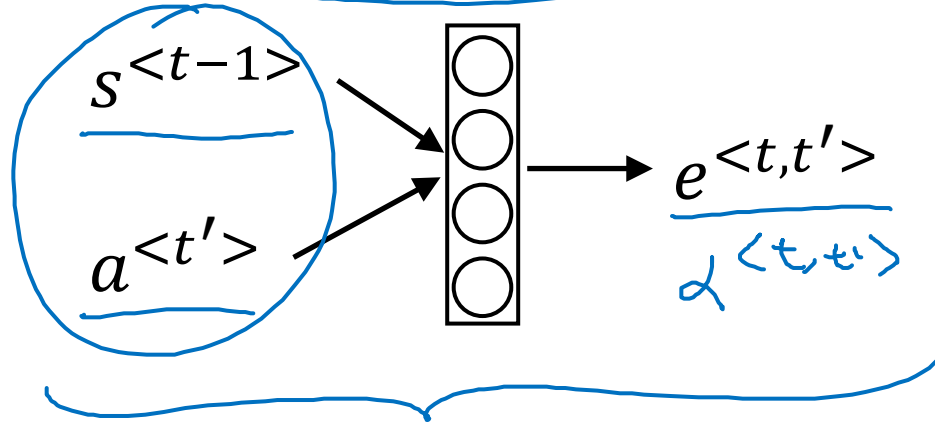
$$c^{(1)} = \sum_{t'} \alpha^{(1,t')} a^{(t')}$$



Computing attention $\alpha^{<t,t'>}$

$\alpha^{<t,t'>}$ = amount of attention $y^{<t>}$ should pay to $a^{<t'>}$

$$\alpha^{<t,t'>} = \frac{\exp(e^{<t,t'>})}{\sum_{t'=1}^{T_x} \exp(e^{<t,t'>})}$$



[Bahdanau et. al., 2014. Neural machine translation by jointly learning to align and translate]

[Xu et. al., 2015. Show, attend and tell: Neural image caption generation with visual attention]

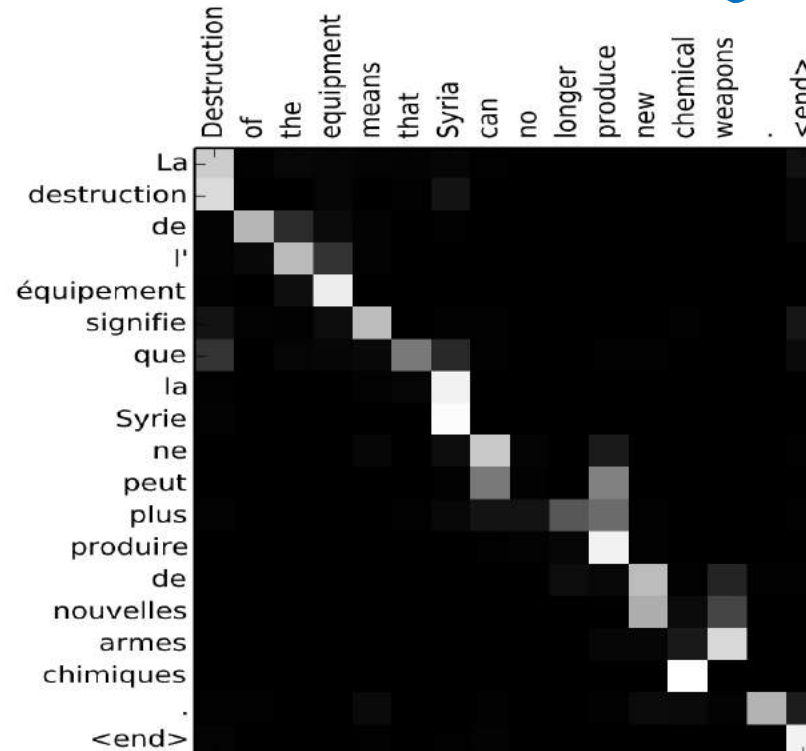
Andrew Ng

Attention examples

July 20th 1969 → 1969 – 07 – 20

23 April, 1564 → 1564 – 04 – 23

Visualization of $\alpha^{<t,t'>}$:





deeplearning.ai

Audio data

Speech recognition

Speech recognition problem

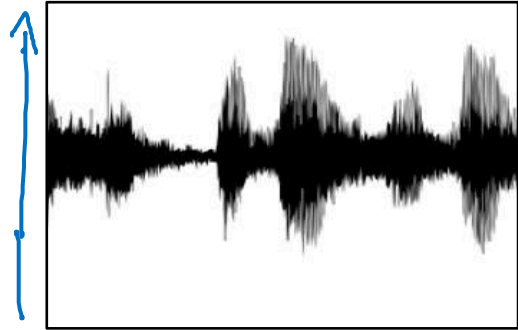
x

audio clip



y

transcript



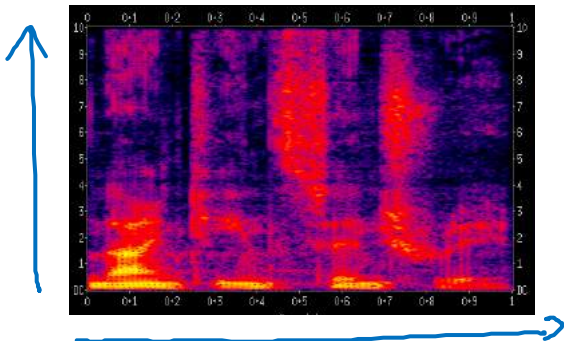
“the quick brown fox”

→ phonemes: de kwik brawn

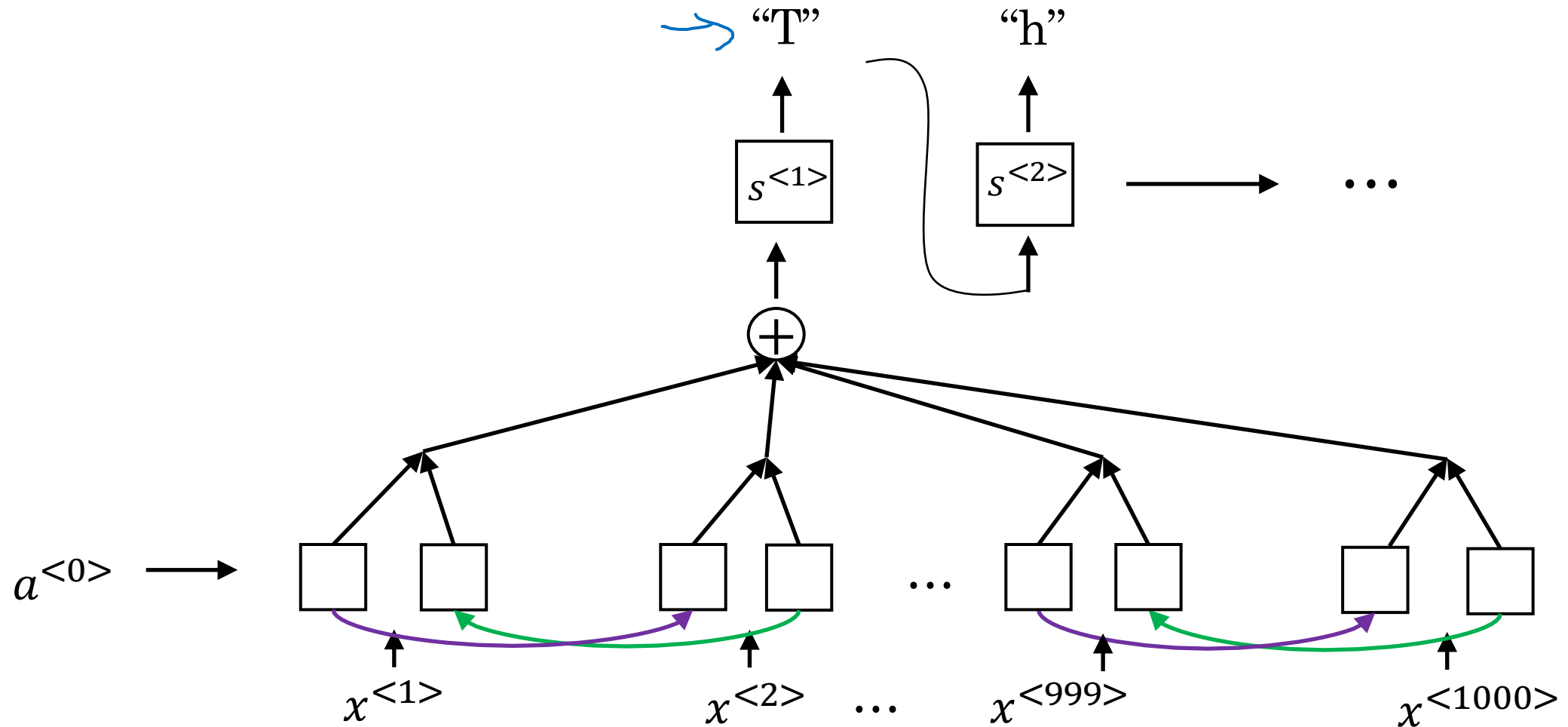
300h

3000h

100,000h

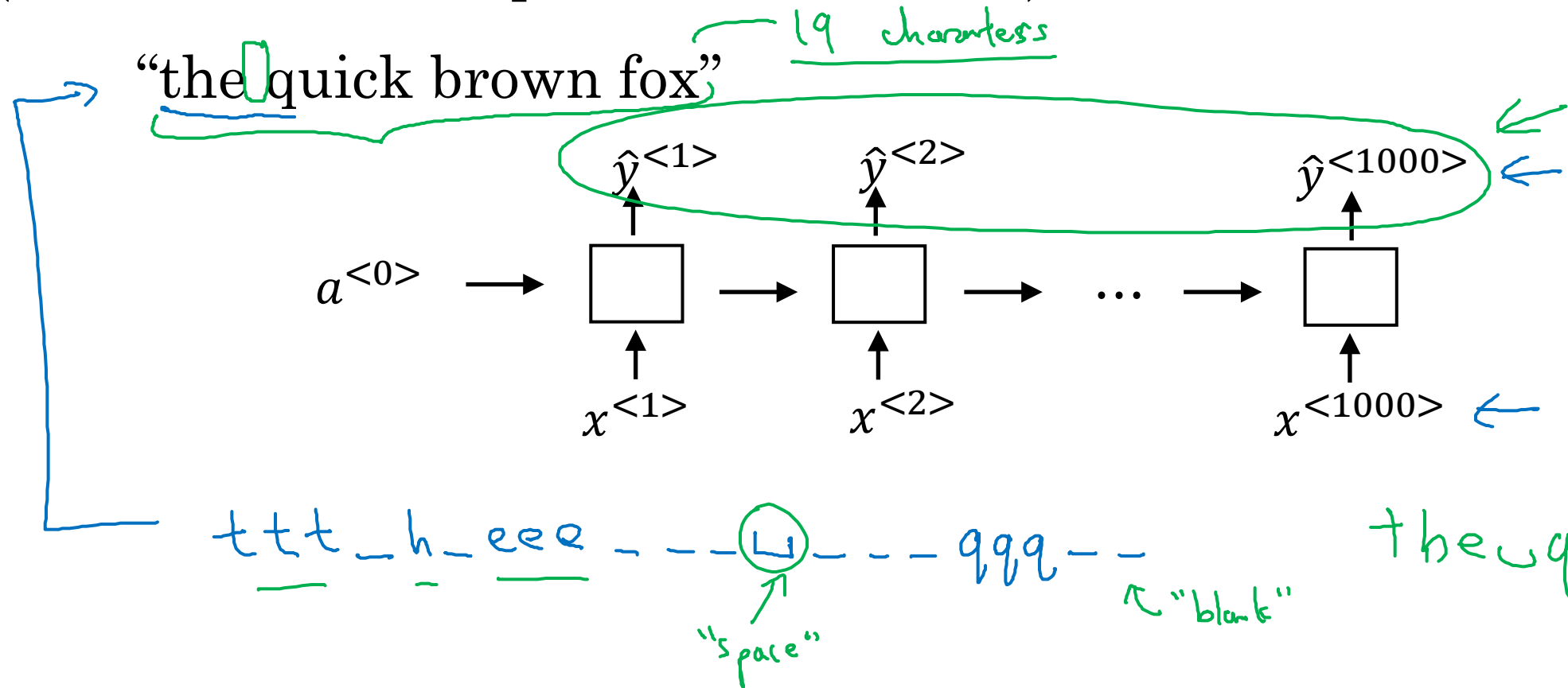


Attention model for speech recognition



CTC cost for speech recognition

(Connectionist temporal classification)



Basic rule: collapse repeated characters not separated by “blank”

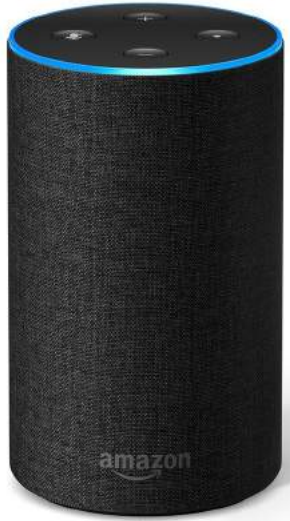


deeplearning.ai

Audio data

Trigger word
detection

What is trigger word detection?



Amazon Echo
(Alexa)



Baidu DuerOS
(xiaodunihao)

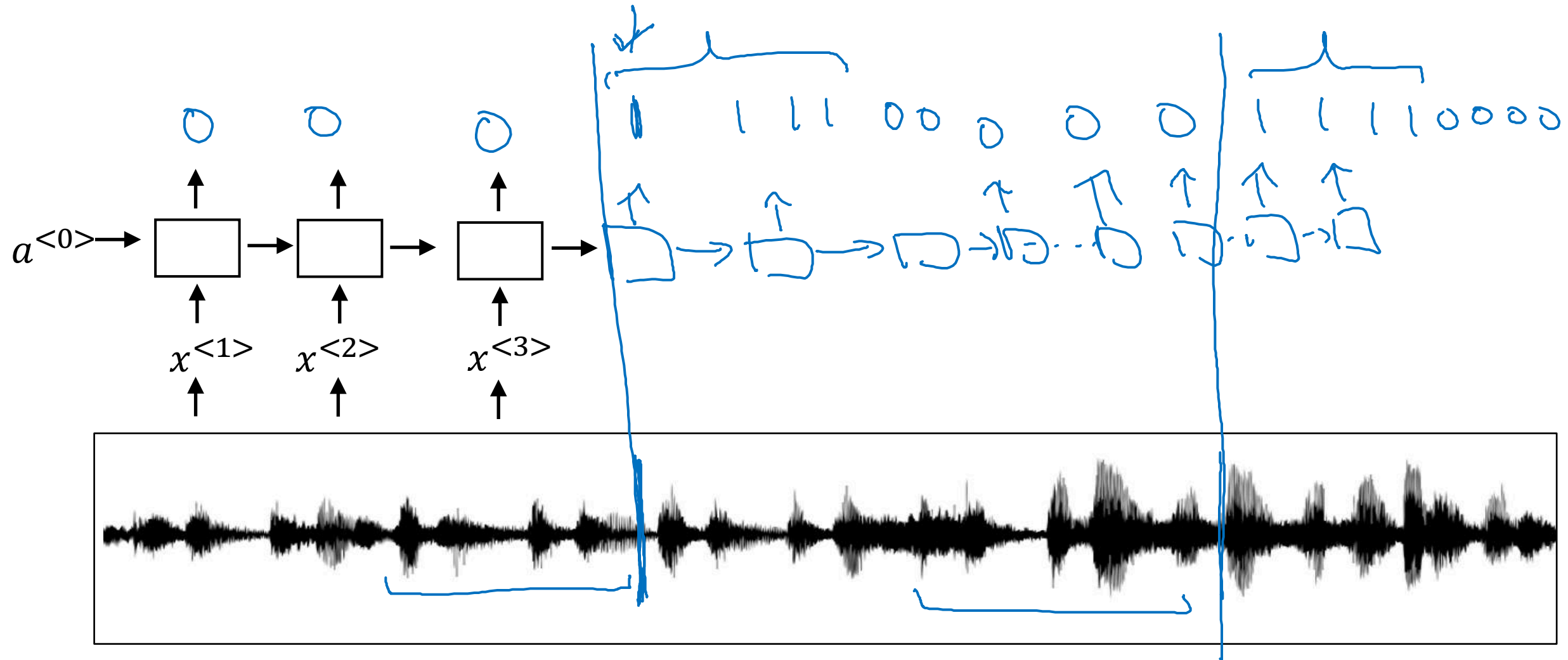


Apple Siri
(Hey Siri)



Google Home
(Okay Google)

Trigger word detection algorithm





deeplearning.ai

Conclusion

Summary and thank you

Specialization outline

1. Neural Networks and Deep Learning
2. Improving Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization
3. Structuring Machine Learning Projects
4. Convolutional Neural Networks
5. Sequence Models

Deep learning is a super power

Please buy this
from shutterstock
and replace in
final video.



www.shutterstock.com · 331201091

Thank you.

- Andrew Ng

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

Sequence to sequence models

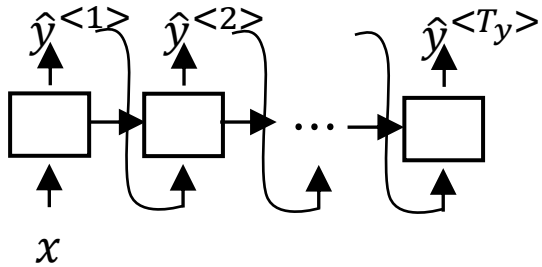
Transformers Intuition

Transformers Motivation

Increased complexity,
sequential

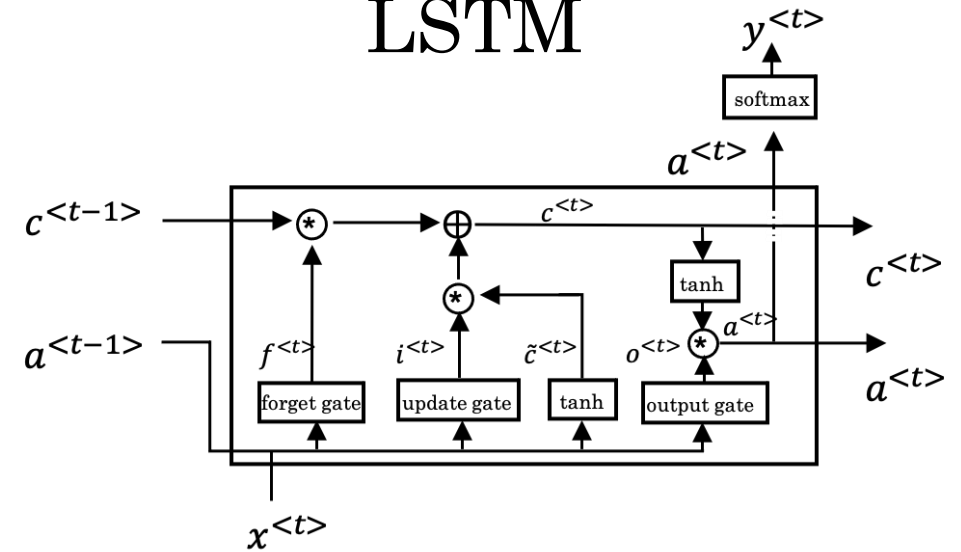


RNN



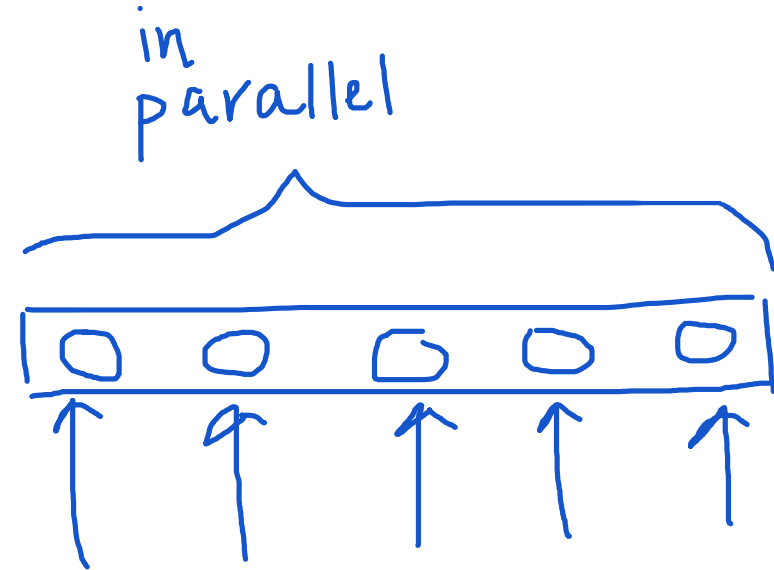
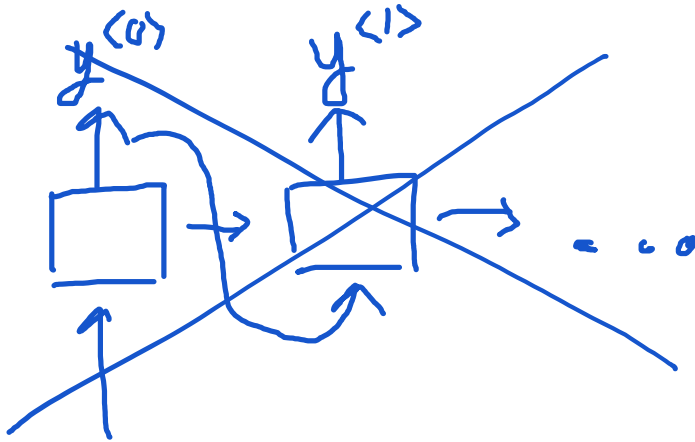
GRU

LSTM



Transformers Intuition

- Attention + CNN
 - Self-Attention
 - Multi-Head Attention





deeplearning.ai

Sequence to sequence models

Self-Attention

Self-Attention Intuition

$A(q, K, V)$ = attention-based vector representation of a word
→ calculate for each word

RNN Attention

$$\alpha^{<\cancel{t}, t'>} = \frac{\exp(e^{<t, t'>})}{\sum_{t'=1}^{T^x} \exp(e^{<t, t'>})}$$

$x^{<1>}$
Jane

$x^{<2>}$
visite

$x^{<3>}$
l'Afrique

$x^{<4>}$
en

$x^{<5>}$
septembre

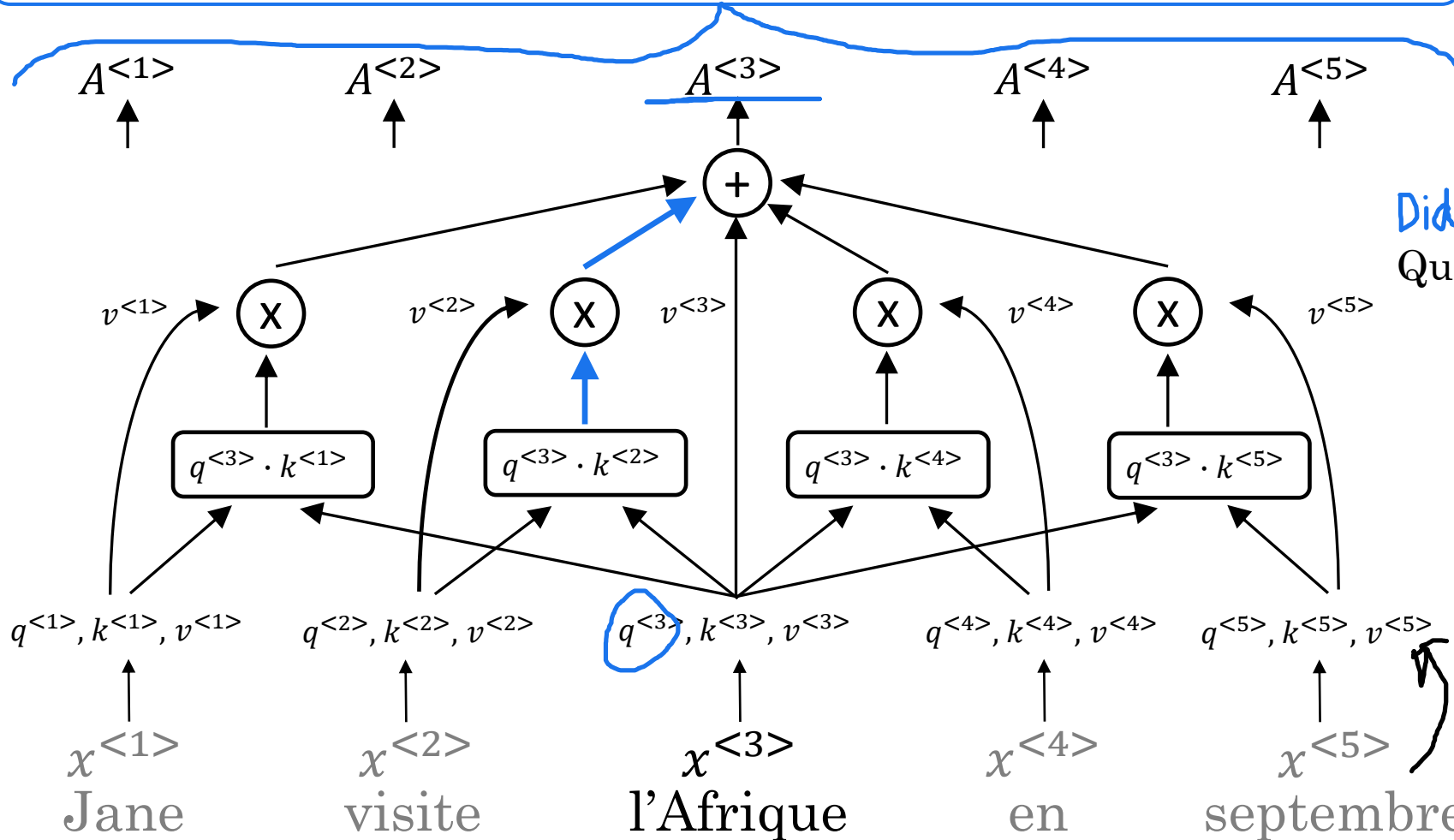
Transformers Attention

$$A(q, K, V) = \sum_i \frac{\exp(q \cdot k^{<i>})}{\sum_j \exp(q \cdot k^{<j>})} v^{<i>}$$

Self-Attention

$$A(q, K, V) = \sum_i \frac{\text{softmax}(\exp(e^{q \cdot k^{<i>}}))}{\sum_j \exp(e^{q \cdot k^{<j>}})} v^{<i>}$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



Did what?

Query (Q)

Key (K)

Value (V)

$q^{<1>}$

$k^{<1>}$

$v^{<1>}$

$q^{<2>}$

$k^{<2>}$

$v^{<2>}$

$q^{<3>}$

$k^{<3>}$

$v^{<3>}$

$q^{<4>}$

$k^{<4>}$

$v^{<4>}$

$q^{<5>}$

$k^{<5>}$

$v^{<5>}$

What's happening there?

person
action
Janu
visit

W^Q, W^K, W^V

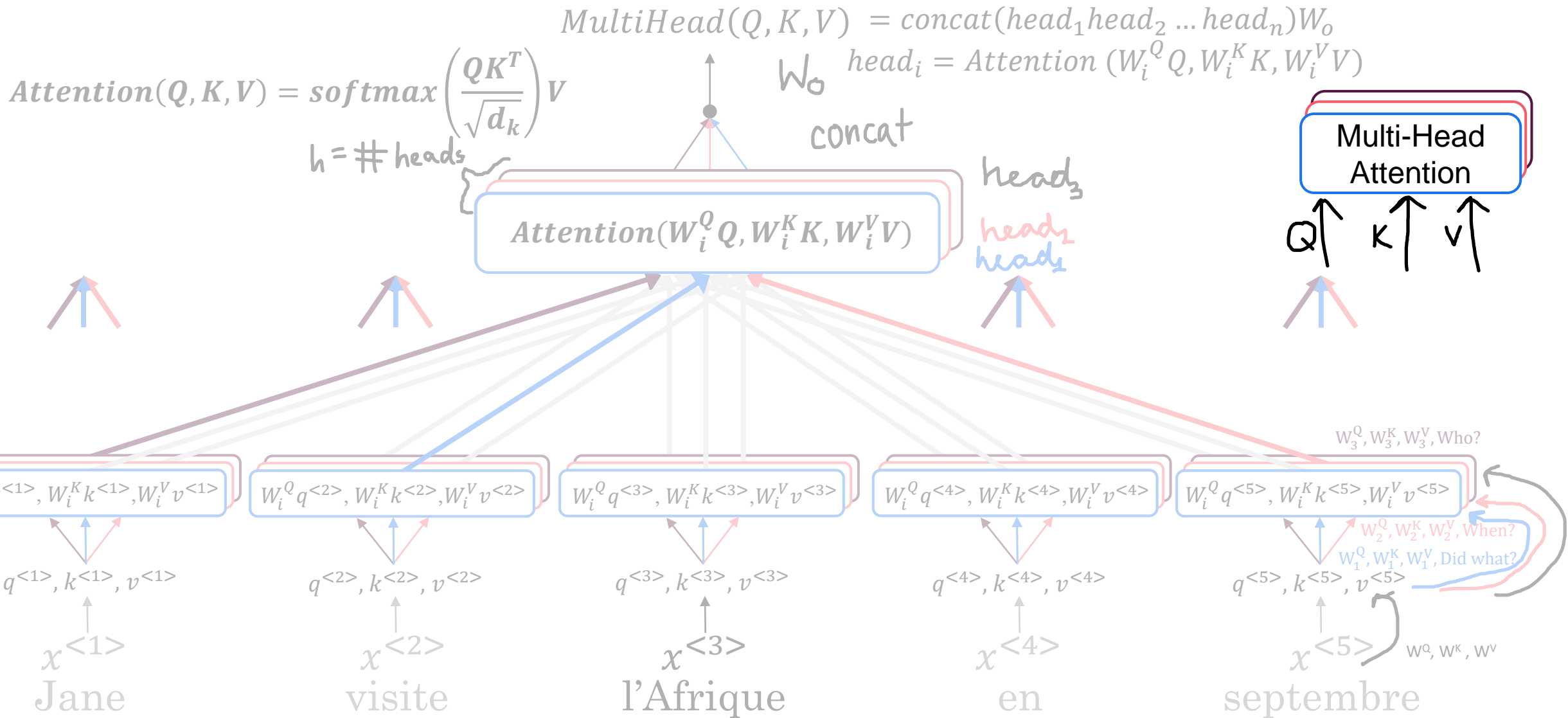


deeplearning.ai

Sequence to sequence models

Multi-Head Attention

Multi-Head Attention





deeplearning.ai

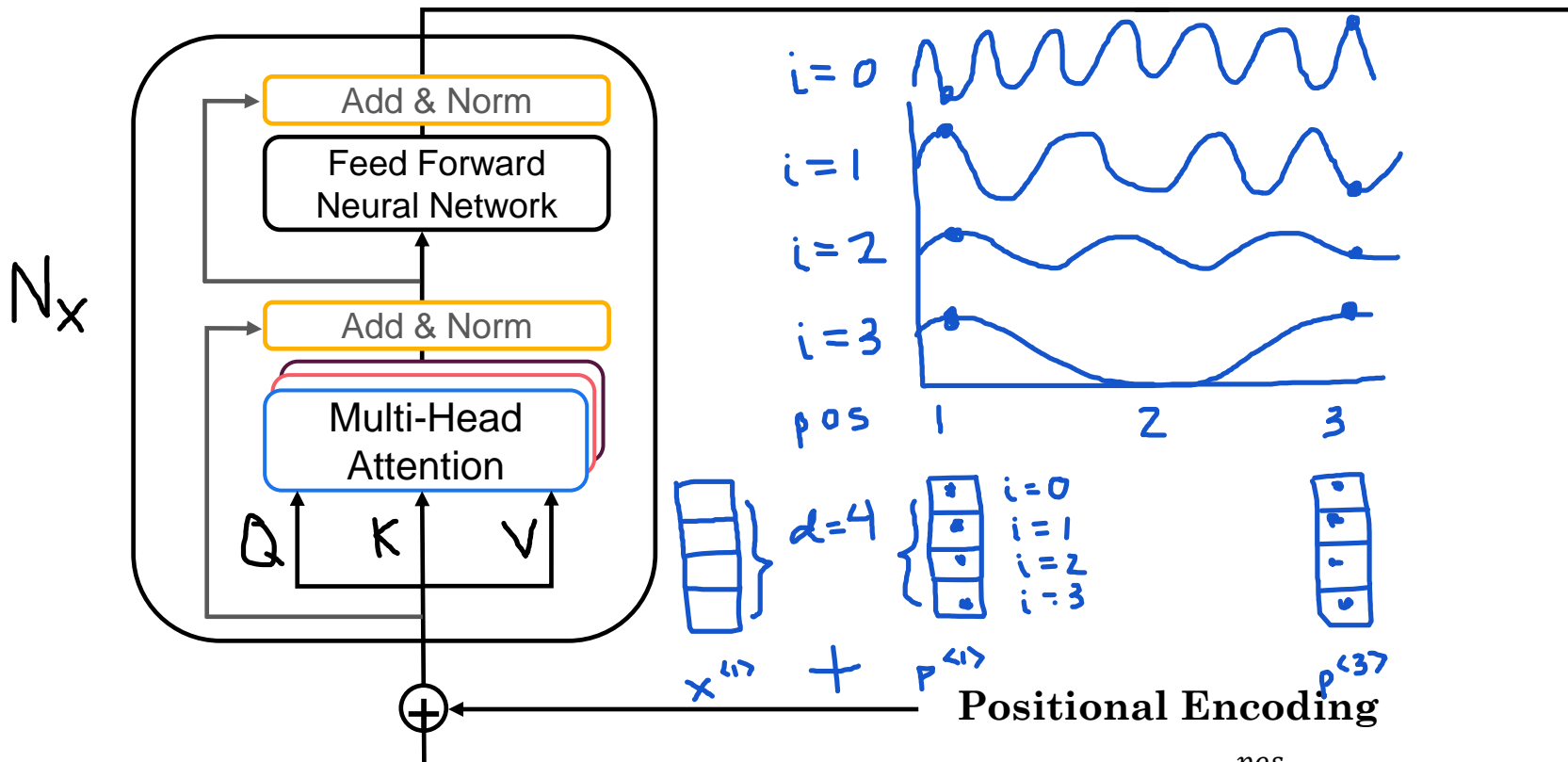
Sequence to sequence models

Transformers

Transformer Details

<SOS> Jane visits Africa in September <EOS>

Encoder

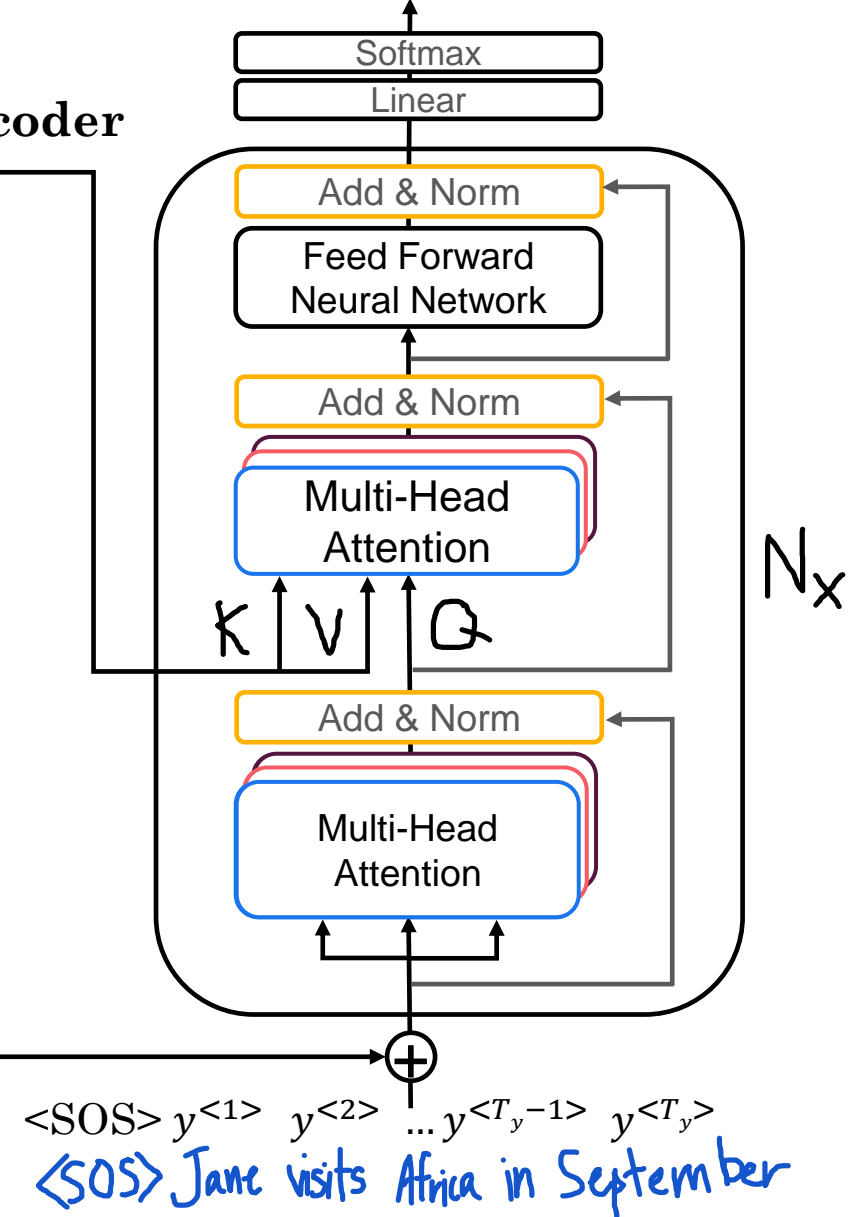


$\langle \text{SOS} \rangle x^{<1>} x^{<2>} \dots x^{<T_x-1>} x^{<T_x>} \langle \text{EOS} \rangle$
Jane visite l'Afrique en septembre

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{1000^{\frac{2i}{d}}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{1000^{\frac{2i}{d}}}\right)$$

Decoder



$\langle \text{SOS} \rangle y^{<1>} y^{<2>} \dots y^{<T_y-1>} y^{<T_y>}$
 $\langle \text{SOS} \rangle$ Jane visits Africa in September