Copyright Notice

Recurrent Neural Networks

deeplearning.ai

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.

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

$T_x = 9$

$\rightarrow$ y: 1 1 0 1 1 0 0 0 0

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

Stylized example - where 0/1 represent name or not for each word

$T_y = 9$

$X^{(i)<t>}$ $T_x^{(i)} = 9$ 15

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

T = length

Representing words

Commercial applications may use dictionary of size 30-50k,
some internet companies may use 1mn+ dictionaries

$x^{<t>}$ (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

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

<UNK> 10,000

$$x^{<1>} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{array}{l} \\ \\ \\ \\ \leftarrow 4075 \\ \\ \\ \\ \end{array}
\quad
x^{<2>} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \begin{array}{l} \\ \\ \\ \leftarrow 6830 \\ \\ \\ \end{array}
\quad
x^{<3>} = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \begin{array}{l} \\ \\ \leftarrow 367 \\ \\ \\ \end{array}
\quad
\updownarrow 10{,}000
\quad
x^{<7>} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

One-hot

Representing words

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

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

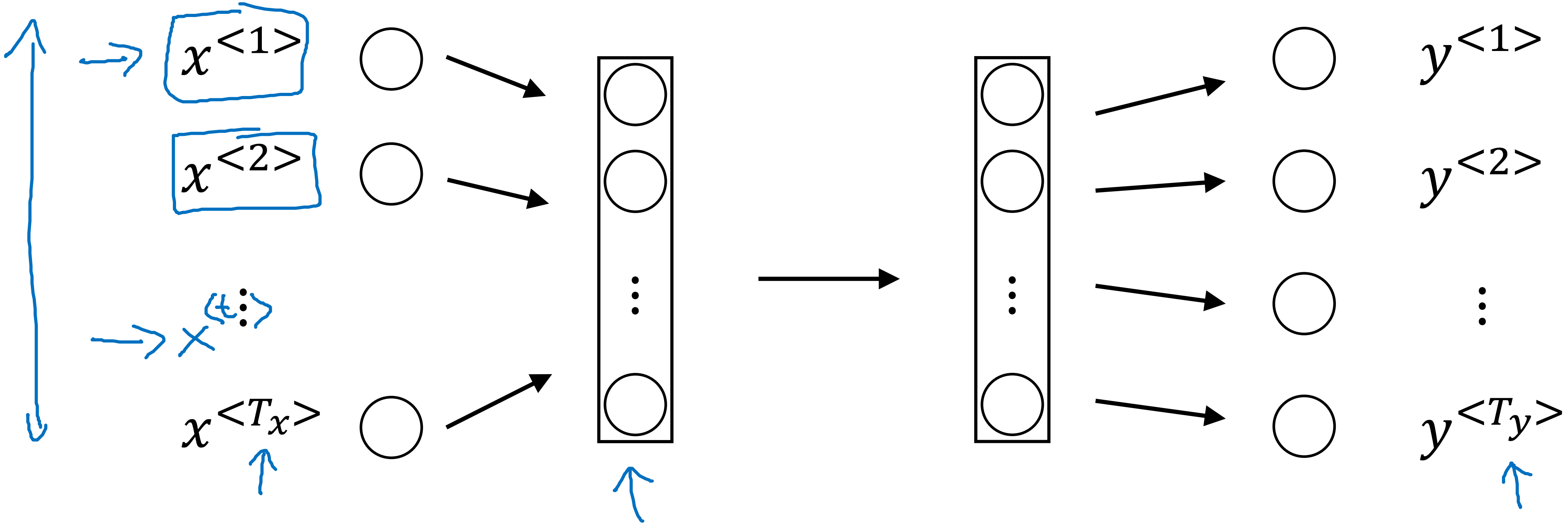
And = 367
Invented = 4700
A = 1
New = 5976
Spell = 8376
Harry = 4075
Potter = 6830
Hermione = 4200
Gran… = 4000

deeplearning.ai

Recurrent Neural Networks

Recurrent Neural Network Model

Why not a standard network?



Problems:

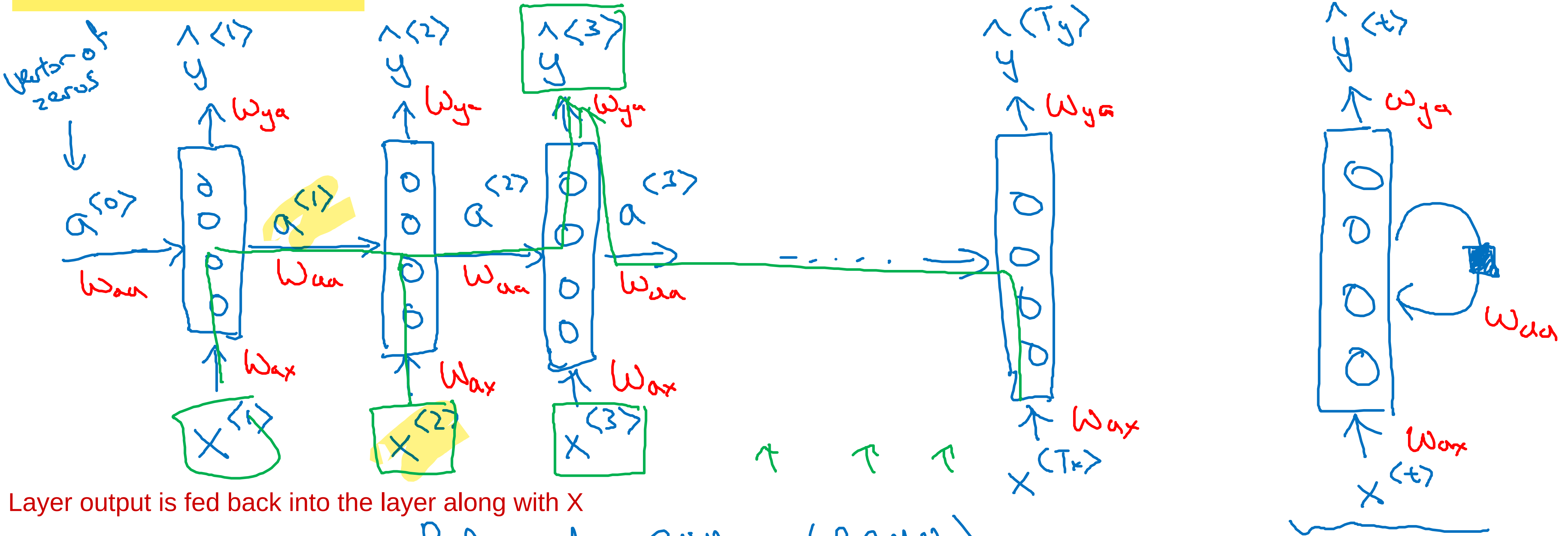
can be managed through padding

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

Andrew Ng

Recurrent Neural Networks

$T_x = T_y$



Layer output is fed back into the layer along with X

W_x and W_a parameters are shared

Bidirectional RNN (BRNN)

Weakness: only prior words can be use not following words

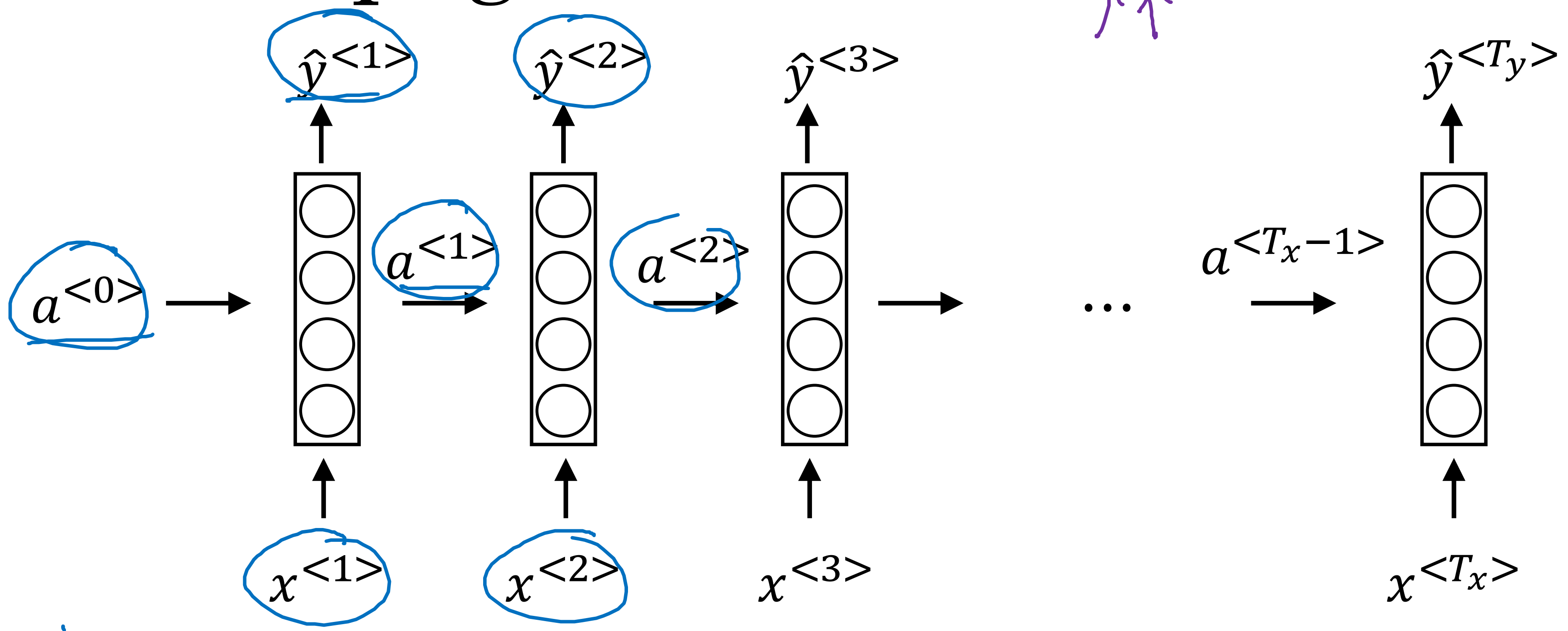
He said, "Teddy Roosevelt was a great President."

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

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

Note: 2 different activation functions are used
- usually, a and y occur in different layers of a NN
- but here same layer provides both a and y, therefore uses both activations simultaneously

Andrew Ng

Simplified RNN notation

parameters for W_aa and W_ax can be combined into one large W_a matrix

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

100 10,000

(100,100) (100,10,000)

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

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

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

$$100\ \left[\underbrace{W_{aa}}_{100} \vdots \underbrace{W_{ax}}_{10\,000}\right] = W_a$$

(100, 10100)

$$[a^{<t-1>}, x^{<t>}] = \begin{bmatrix} a^{<t-1>} \\ \hline x^{<t>} \end{bmatrix} \begin{matrix} 100 \\ 10000 \end{matrix} \quad 10100$$

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

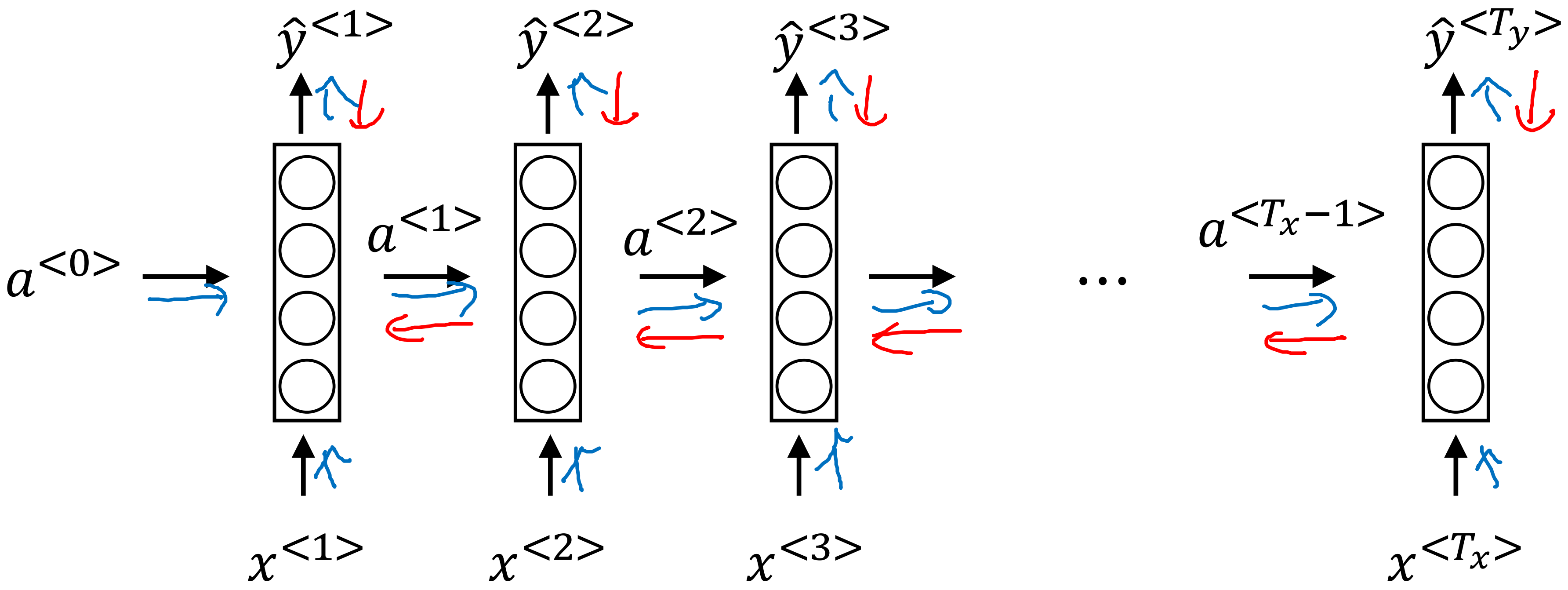
Andrew Ng

Recurrent Neural Networks

deeplearning.ai

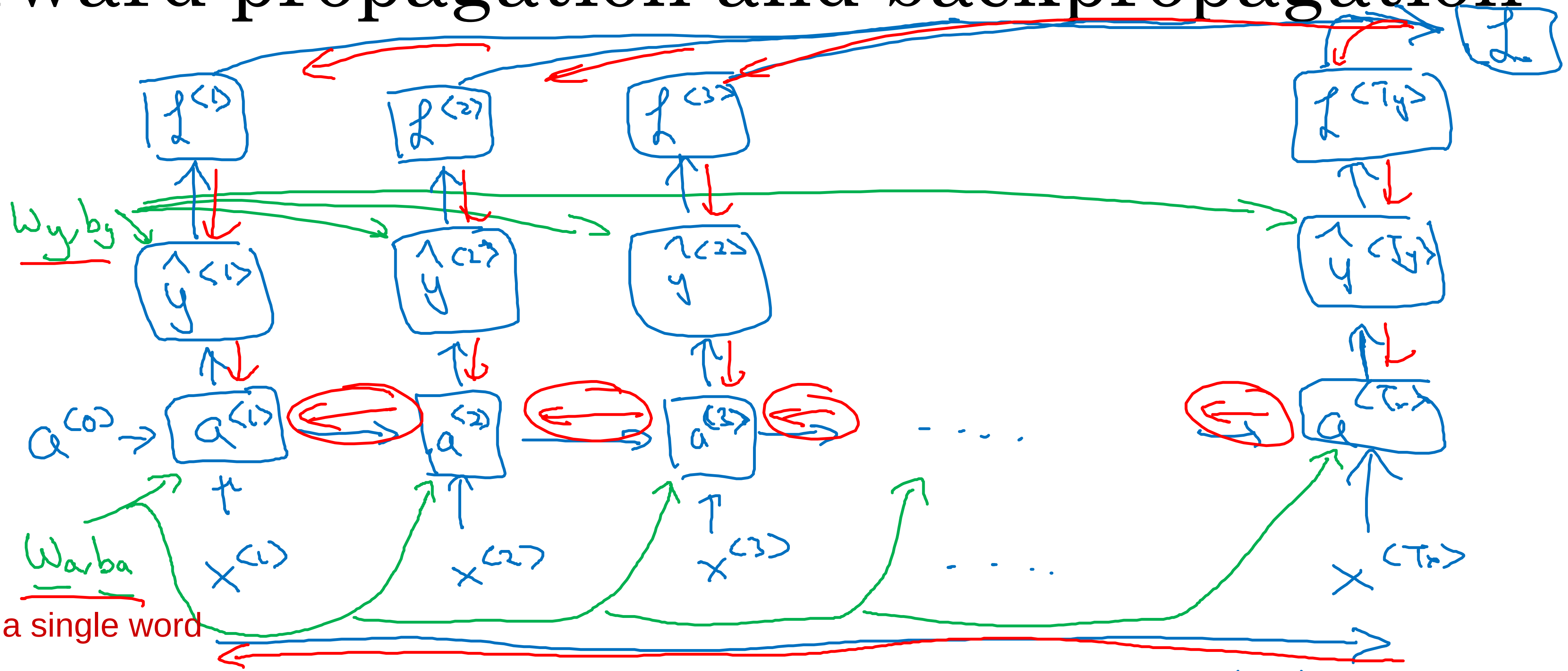
Backpropagation through time

Forward propagation and backpropagation



Andrew Ng

Forward propagation and backpropagation



Loss for a single word

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

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

Backpropagation through time

Sum of losses from each step or each word

Andrew Ng

Recurrent Neural Networks

deeplearning.ai

Different types of RNNs

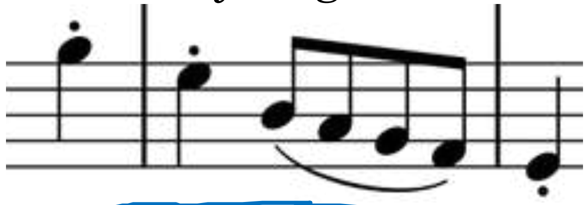
Examples of sequence data

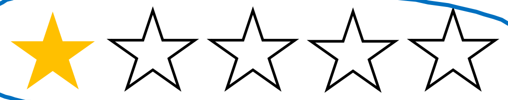
T_x T_y

x y

Speech recognition

→ "The quick brown fox jumped over the lazy dog."

Music generation
Input length = 0 or 1 Ø
→


Sentiment classification
"There is nothing to like in this movie."
→

Output length 1 integer

DNA sequence analysis
AGCCCCTGTGAGGAACTAG → AGCCCCTGTGAGGAACTAG

Machine translation
Voulez-vous chanter avec moi? → Do you want to sing with me?

Input length != output length

Video activity recognition

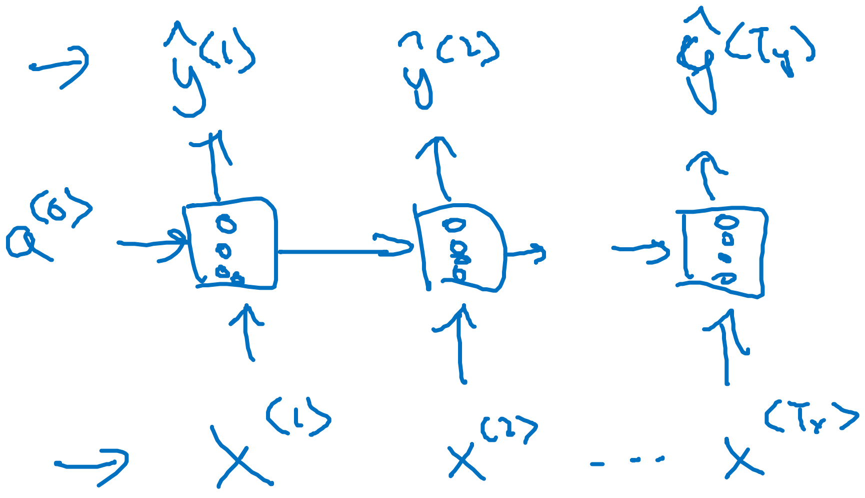
→ Running

Name entity recognition
Yesterday, Harry Potter met Hermione Granger. → Yesterday, Harry Potter met Hermione Granger.

Andrew Ng

Examples of RNN architectures

$T_x = T_y$

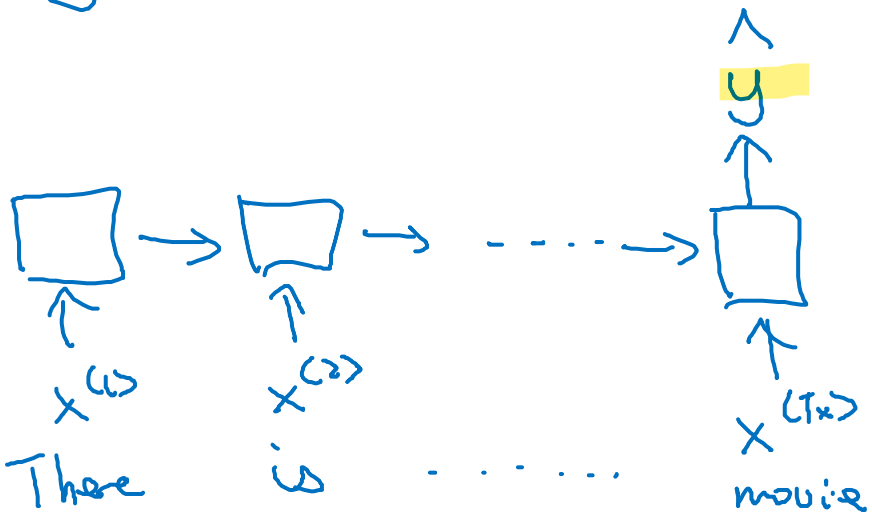


Many-to-many

Sentiment classification

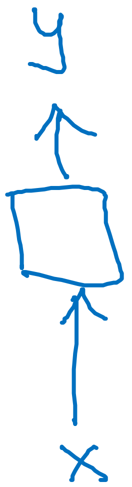
x = text

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



Many-to-one

Trivial / earlier courses

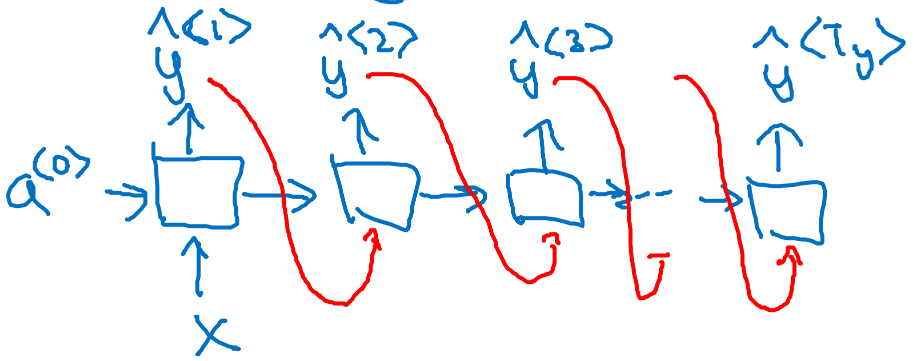


one-to-one

Examples of RNN architectures

Music generation

$x \rightarrow y^{\langle 1 \rangle}\, y^{\langle 2 \rangle} \dots y^{\langle T_y \rangle}$

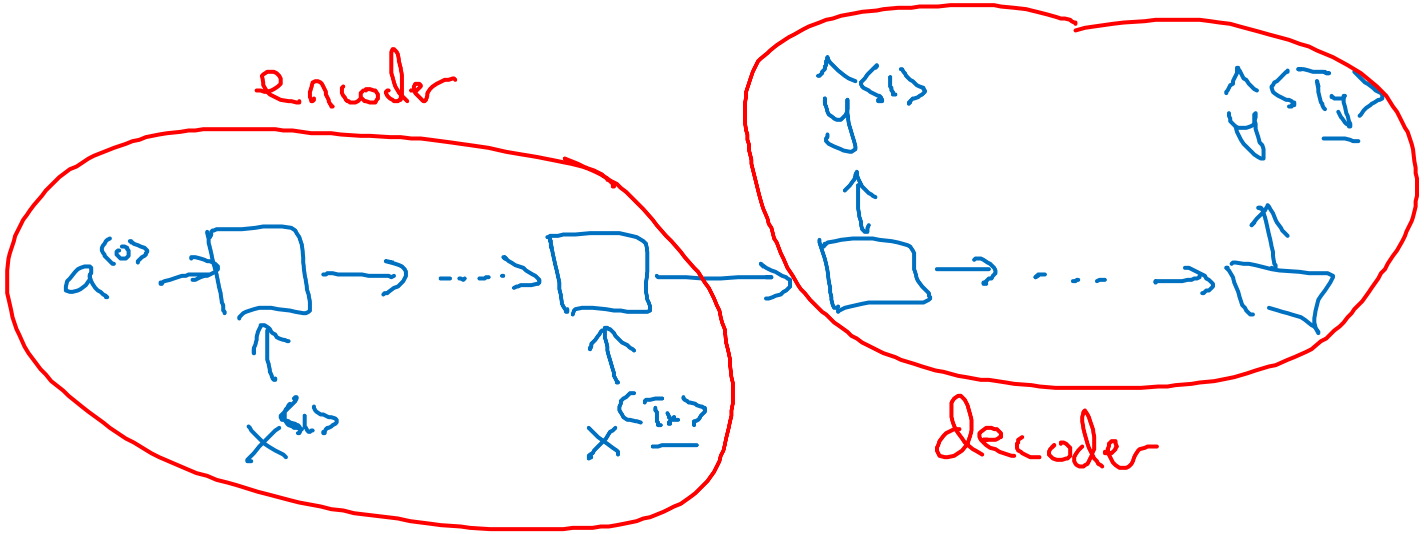


One-to-many

$x = \emptyset$

Interesting both 'a' and 'y' feed to next layer

Machine translation

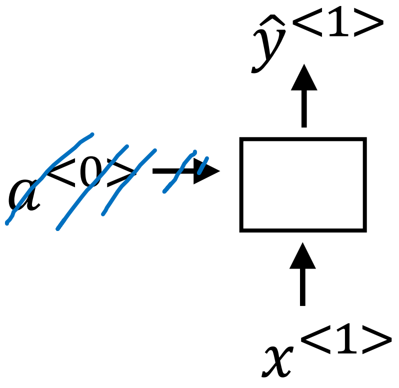


Many-to-many

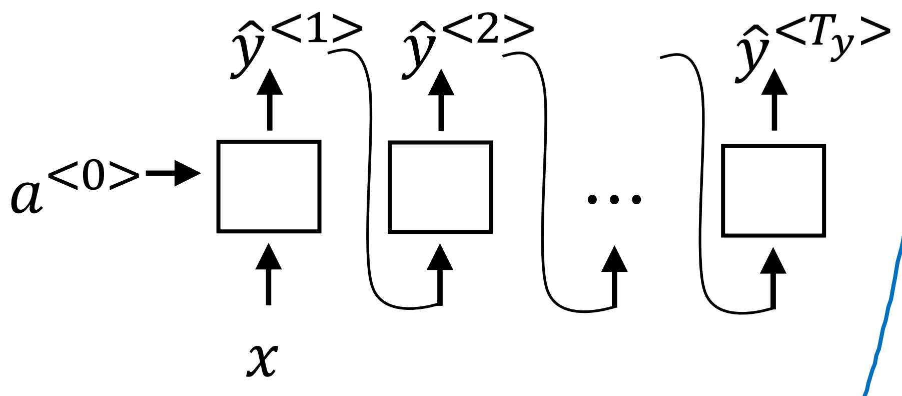
Translation

Andrew Ng

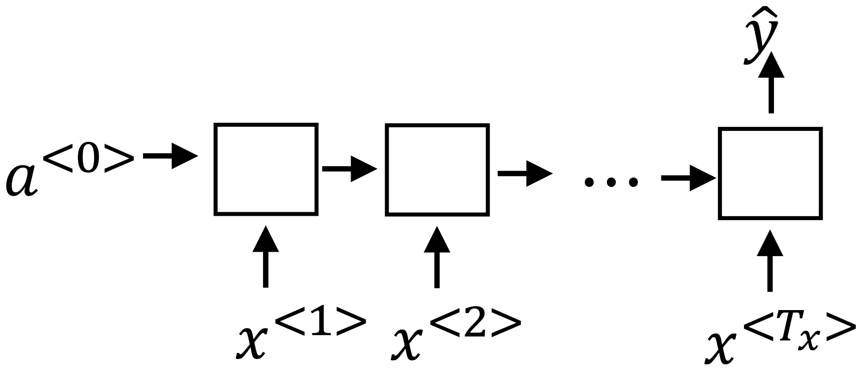
Summary of RNN types



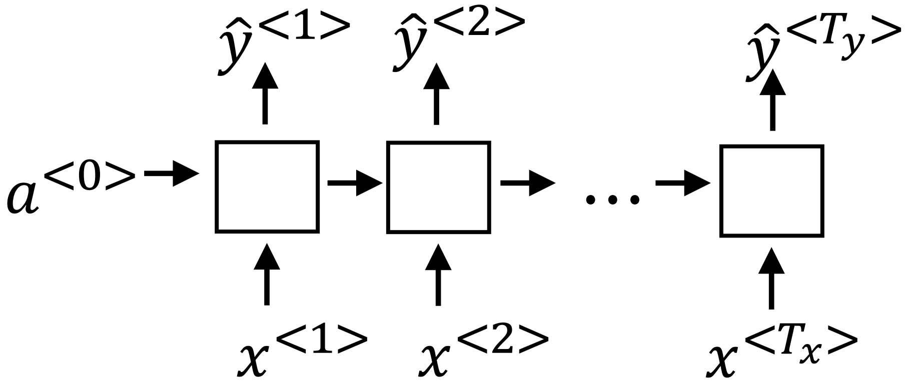
One to one



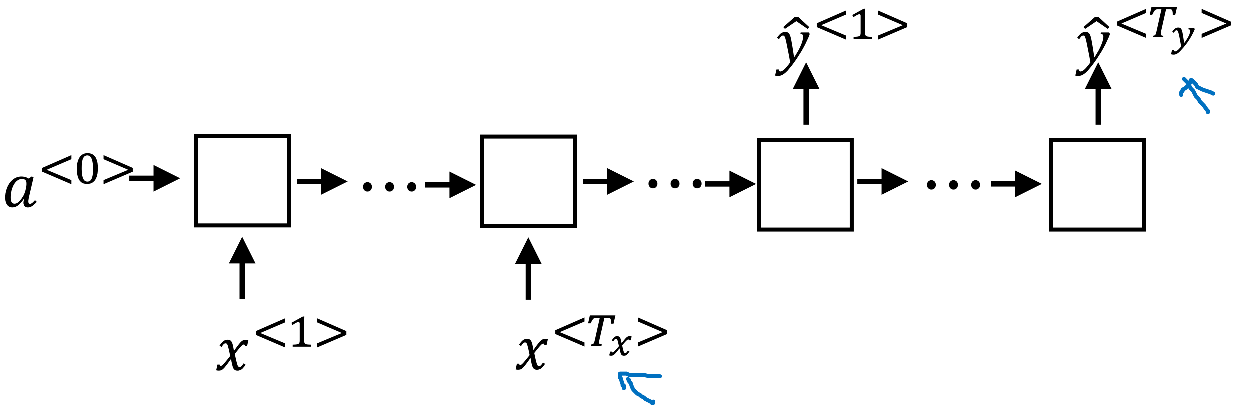
One to many



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?

Language model provides probability for a sentence

Speech recognition

The apple and pair salad.

→ The apple and pear salad.

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

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

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

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

Speech recognition system will pick the sentence with higher probability. Probability will come from Language model.

Language model assigns higher probability to second sentence. Even though both sound same.

Language modelling with an RNN

Training set: large corpus of english text.

colection

Tokenize

add an 'end of sentence' token; comes in handy

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

One-Hot vectors

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

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

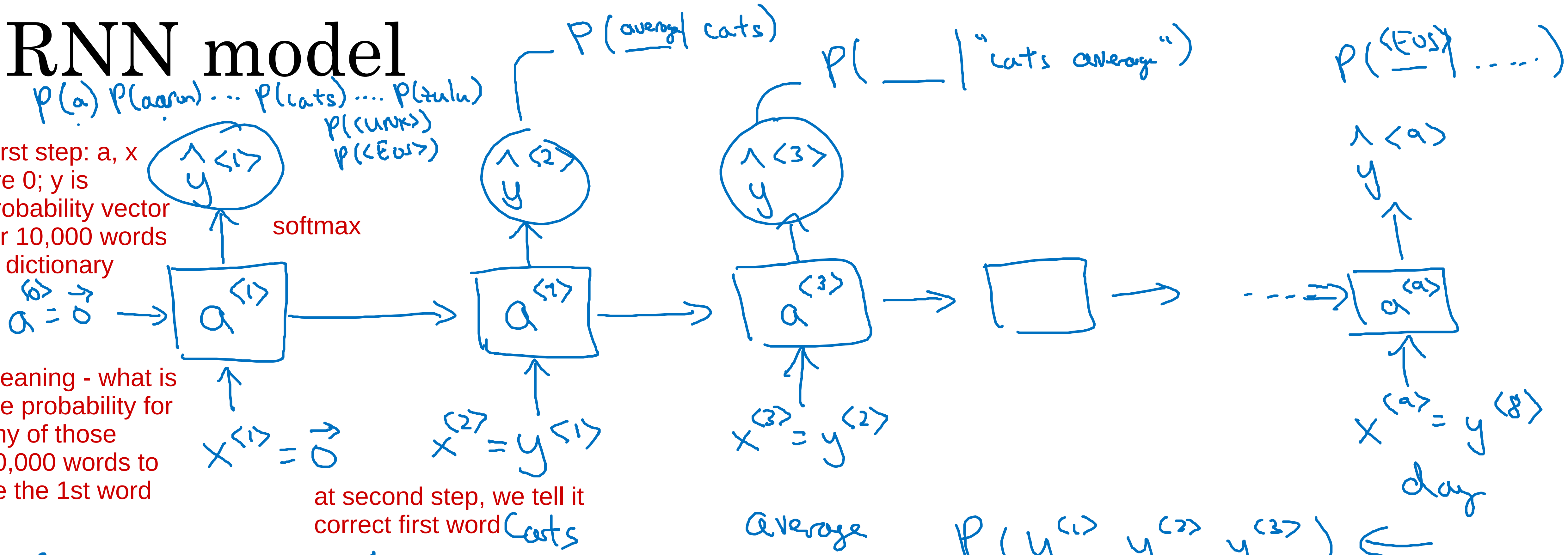
The Egyptian ~~Mau~~ is a bread of cat. <EOS>

<UNK>

10,000

Andrew Ng

RNN model



First step: a, x are 0; y is probability vector for 10,000 words in dictionary

softmax

meaning - what is the probability for any of those 10,000 words to be the 1st word

at second step, we tell it correct first word

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

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

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

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

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

trained by reducing this loss.

For speech recognition:
Prob(sentence) = multi. of conditional probabilities

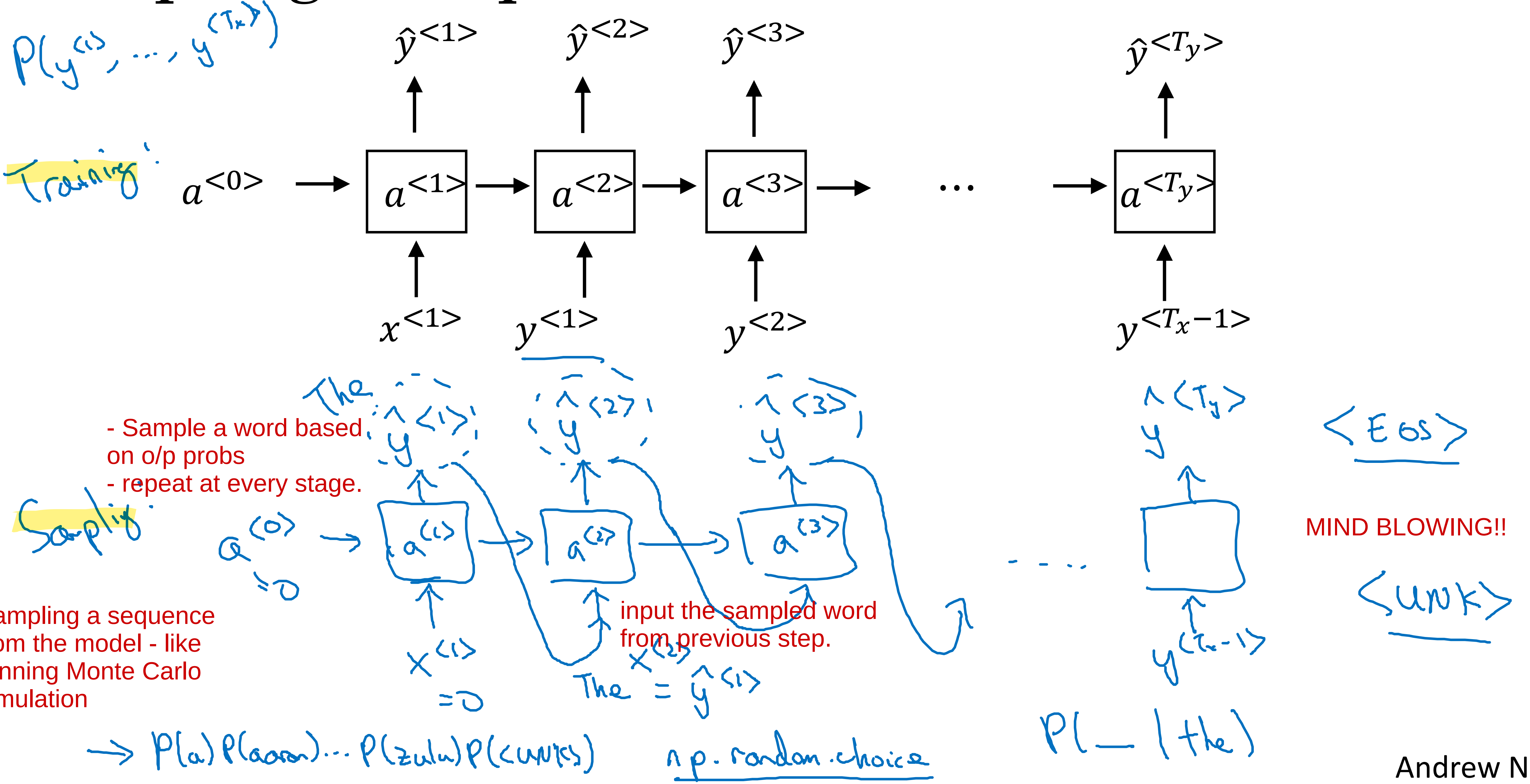
Andrew Ng

deeplearning.ai

Recurrent Neural Networks

Sampling novel sequences

Sampling a sequence from a trained RNN



- Sample a word based on o/p probs
- repeat at every stage.

input the sampled word from previous step.

MIND BLOWING!!

Sampling a sequence from the model - like running Monte Carlo simulation

Andrew Ng

Character-level language model

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

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

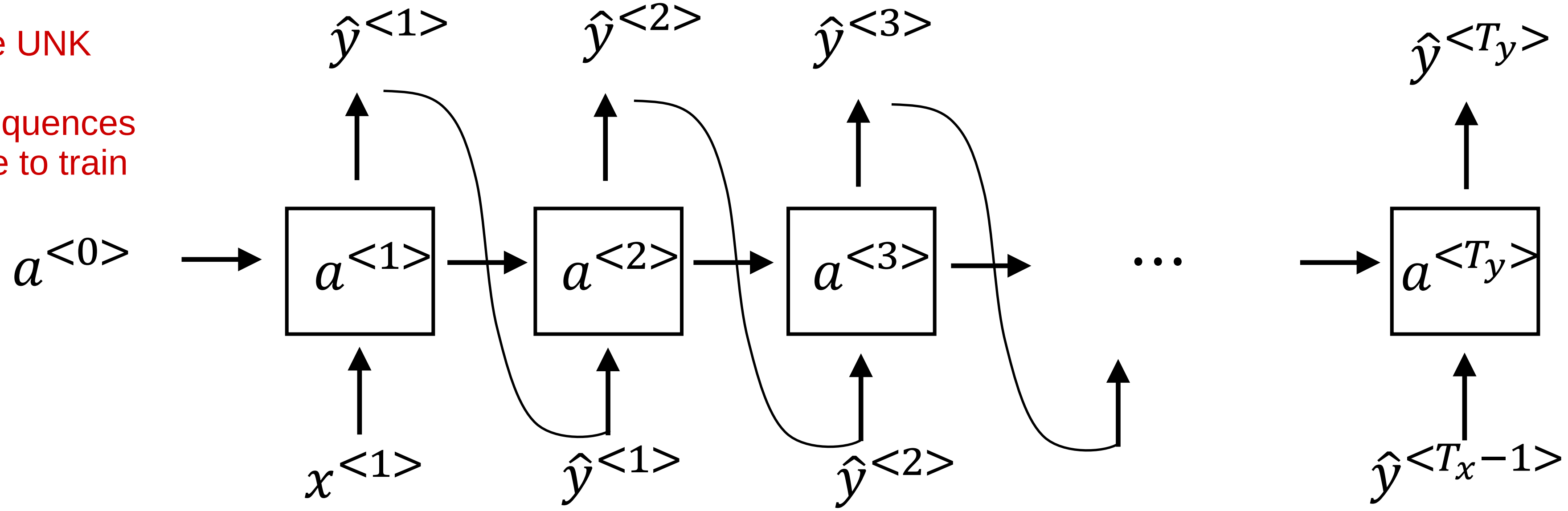
Dictionary can also be at CHARACTER level - azAZ09,.; etc.

+ never generate UNK token
- much longer sequences
- more expensive to train

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

Cat average …

Max



Andrew Ng

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

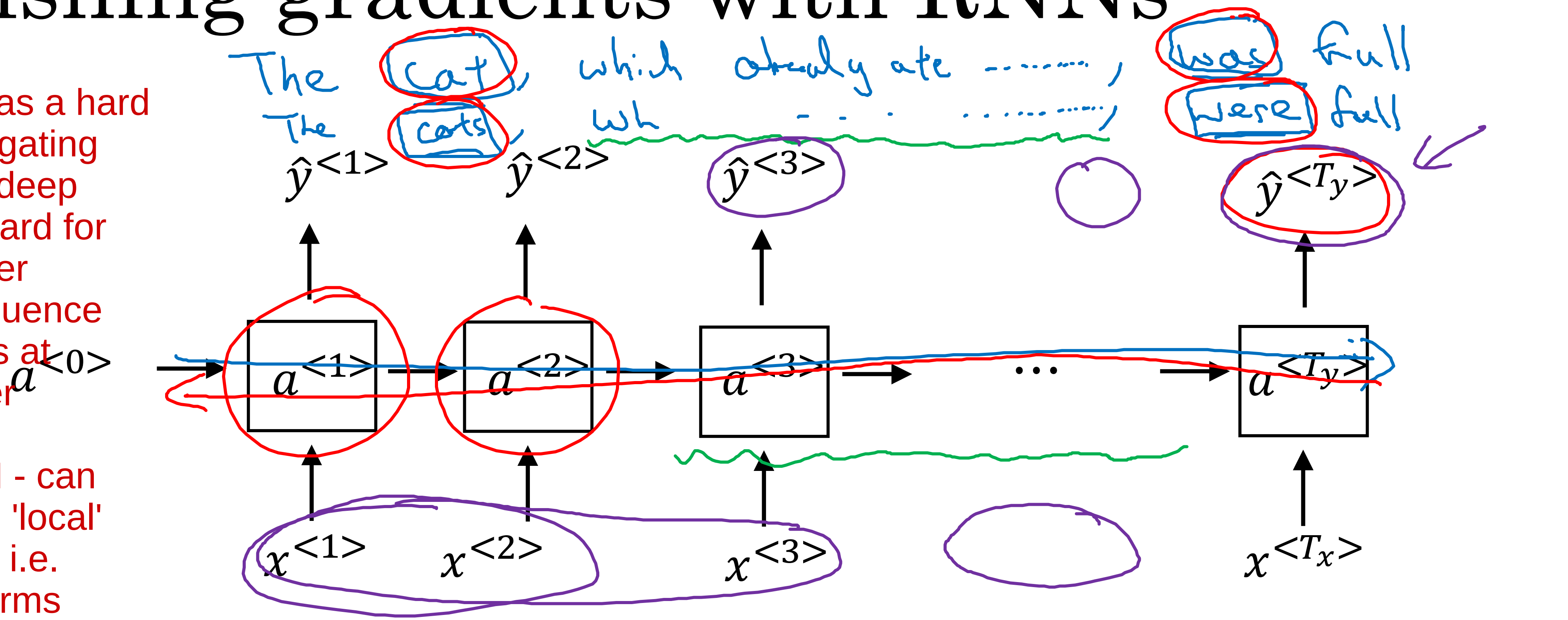
Recurrent Neural Networks

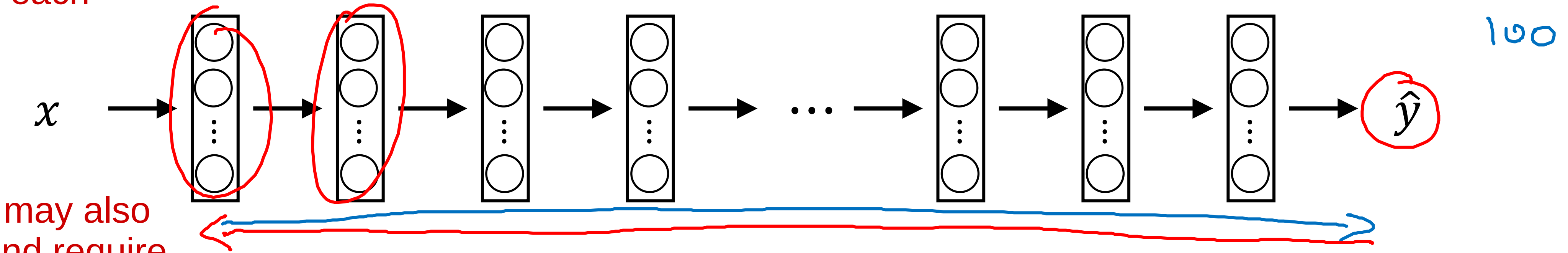
Vanishing gradients with RNNs

Vanishing gradients with RNNs

Gradient has a hard time propagating back for a deep network. Hard for o/p at a later layer to influence parameters at earlier layer

Basic RNN - can only model 'local' influences, i.e. close by terms impacting each other





gradients may also explode and require clipping

Exploding gradients.

NaN

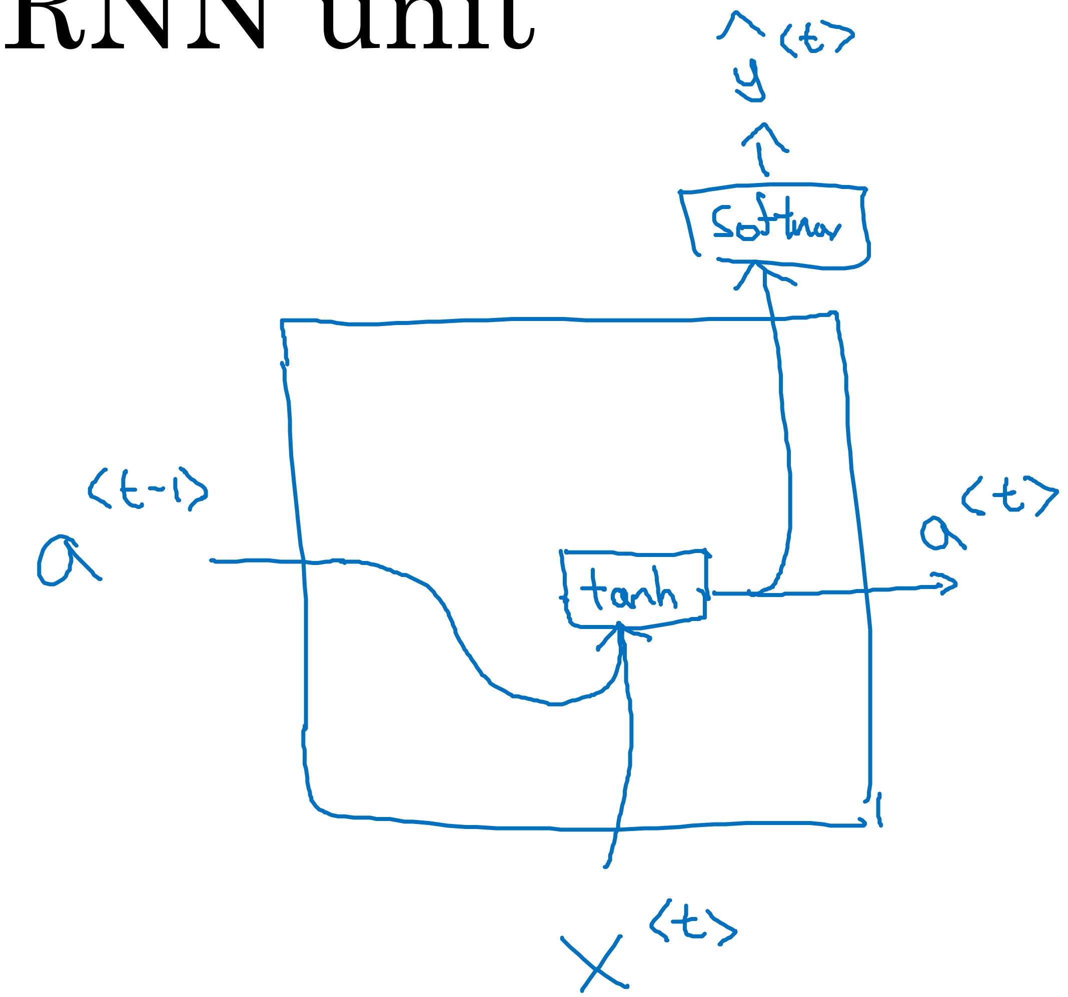
Gradient clipping

deeplearning.ai

Recurrent Neural Networks

Gated Recurrent Unit (GRU)

RNN unit



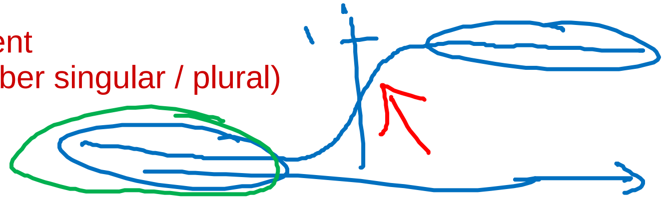
tanh

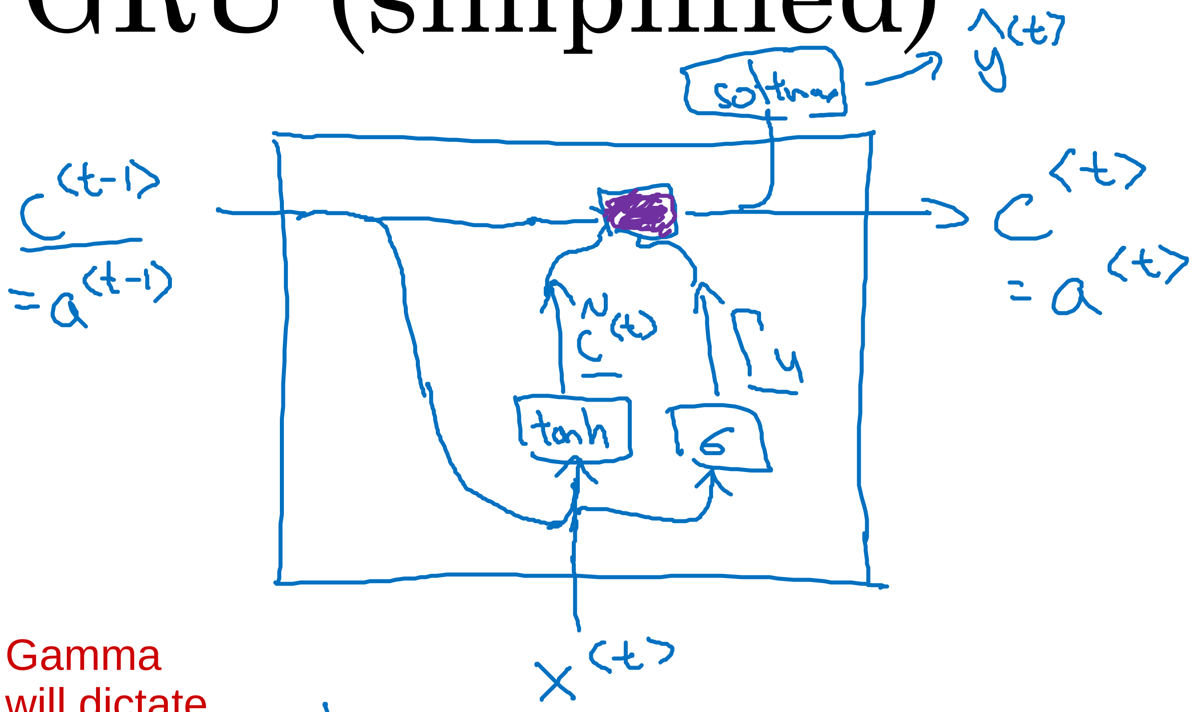
$$\underline{a^{<t>}} = \underline{g}(W_a\underline{[a^{<t-1>}, x^{<t>}]} + \underline{b_a})$$

Andrew Ng

GRU (simplified)

Key Idea: C(~t) vector is a candidate to replace C(t)
Gate (gamma-u) vector either {0,1} (sigmoid) - controls the assignment
Remember this are all vectors! (maybe one bit is assigned to remember singular / plural)





c = memory cell

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

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

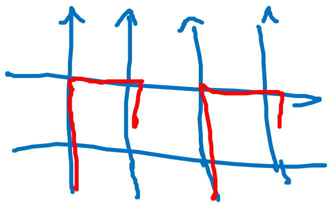
$$\Gamma_u = \sigma(W_u[c^{<t-1>}, x^{<t>}] + b_u)$$

"update"

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

$\Gamma_u = 1$

element-wise



Gate

$\Gamma_u = 0.000001$

Gamma will dictate when to update that memory cell

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

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

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

for LSTM we make one modification:
- add 'Gamma(r)' to first equation
- C(tilda) equation becomes: W_c[Gamma(r) * c(t-1), x(y)]
- Gamma(r) represents relevance - very similar to Gamma(u)

Why use Gamma(r)
- based on research ideas to ensure gradients don't vanish and long-term relevance is captured

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

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

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

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

LSTM

LSTM is another common version

KEY INSIGHT: This Gate vector can become 1 for a particular index, and replace C(tilda) with C(t-1) for that index of C(t). This takes away the impact of 'W' parameters and activation function and avoid the problem of vanishing or exploding gradients.

The cat, which ate already, was full.

Andrew Ng

Recurrent Neural Networks

deeplearning.ai

LSTM (long short term memory) unit

GRU and LSTM

a != c
two gamma gates for c(t) update - 'update' + 'forget'
one gamma gate for 'output'

GRU

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

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

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

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

Γ_f

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

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

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

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

tanh(C<t>)

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

Andrew Ng

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

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

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

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

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

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

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

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

Andrew Ng

LSTM in pictures

3 outputs from the box: c(t), a(t)
and y(t) = softmax(a(t))

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

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

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

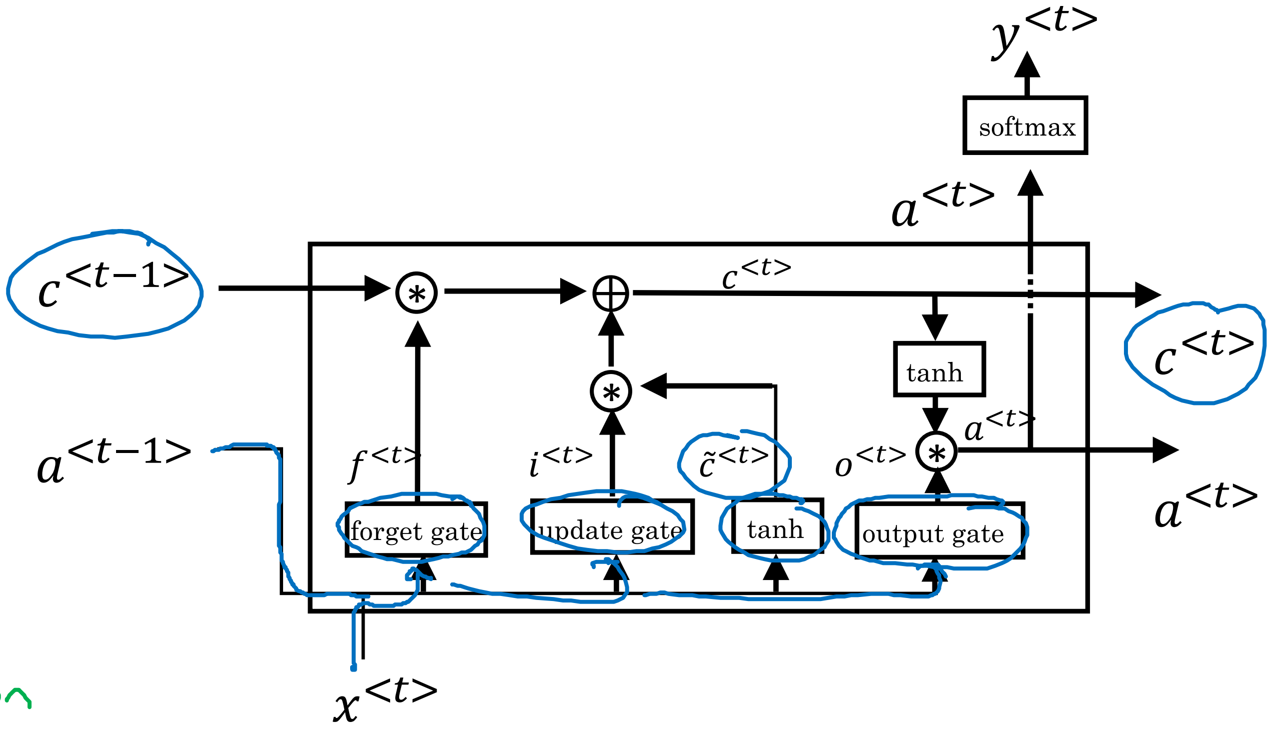
$$\Gamma_o = \sigma(W_o[\,a^{<t-1>}, x^{<t>}] + b_o)$$

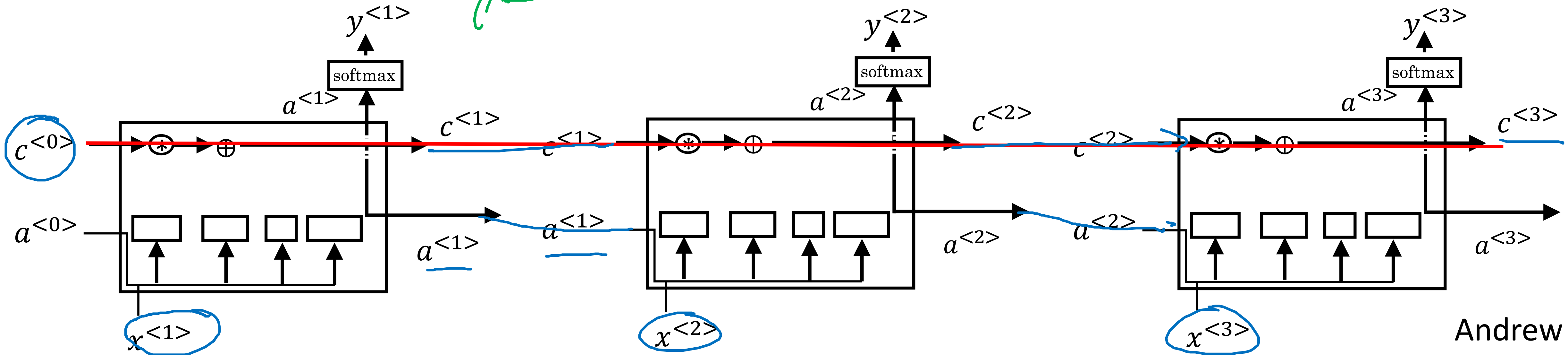
$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>}$$

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

$c^{<t-1>}$

peephole connection





Andrew Ng

deeplearning.ai

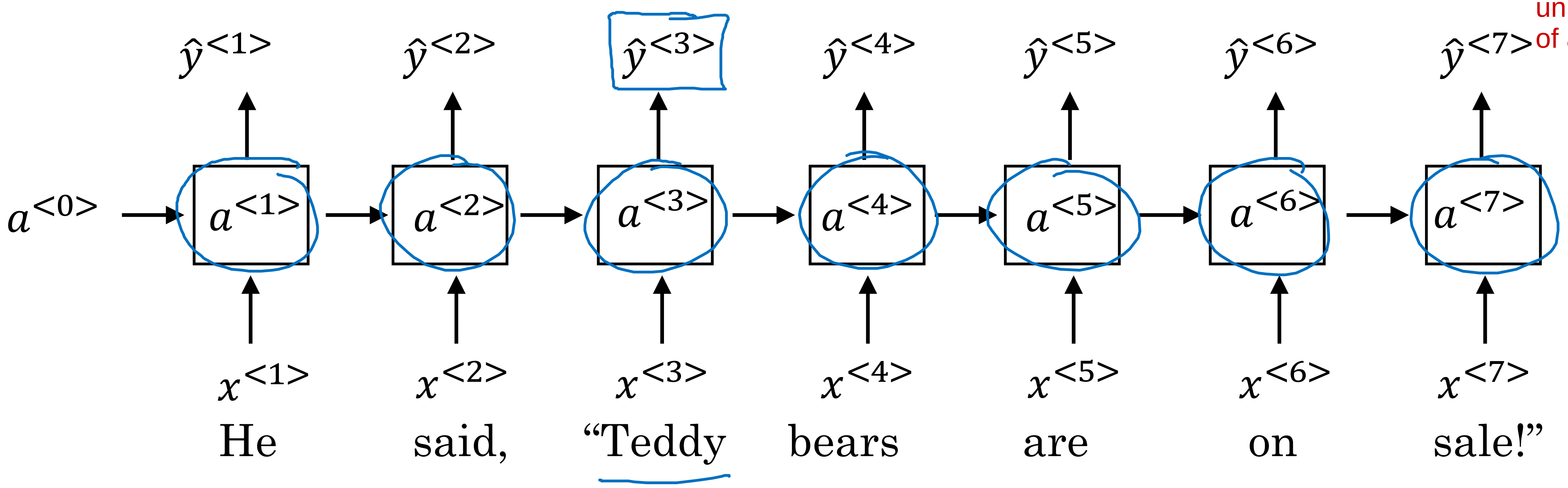
Recurrent Neural Networks

Bidirectional RNN

Getting information from the future

He said, “Teddy bears are on sale!”

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



unit can be of any type:

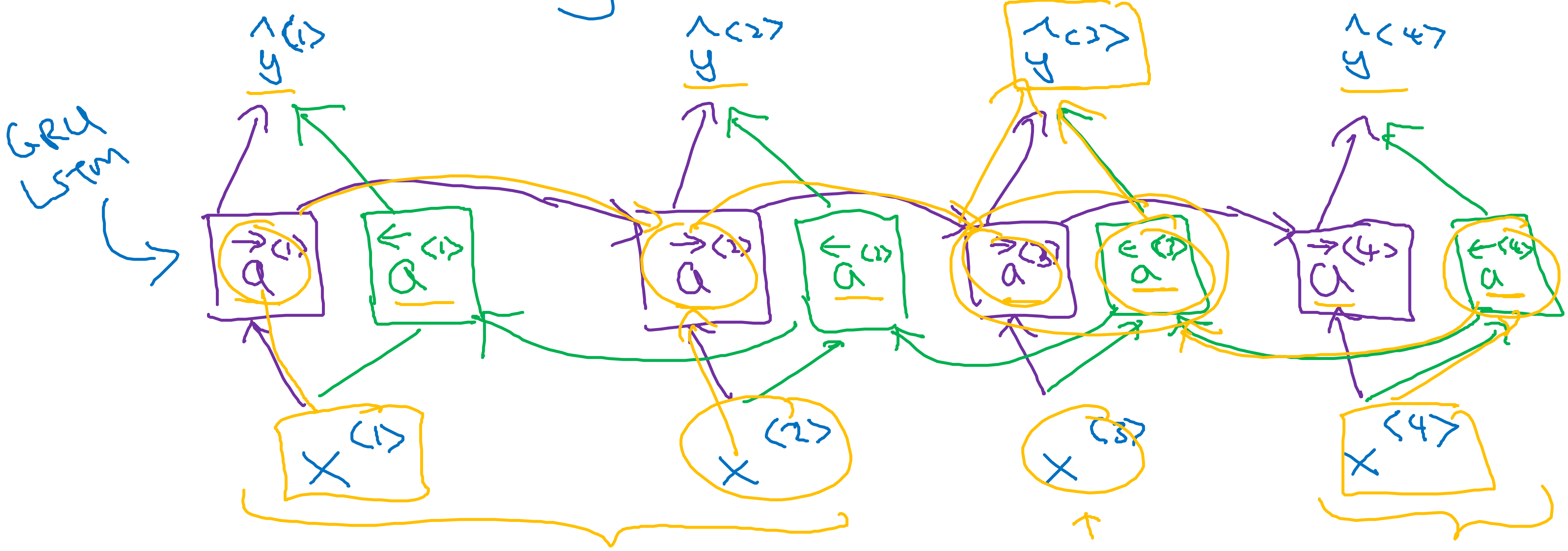
RNN

GRU

LSTM

Bidirectional RNN (BRNN)

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



Acyclic graph

He said, "Teddy Roosevelt ...."

Common choice for NLP is to use BRNN + LSTM
- need entire sequence of data to make prediction
- e.g. won't work for translation, until the person stops speaking

BRNN w/ LSTM

Andrew Ng

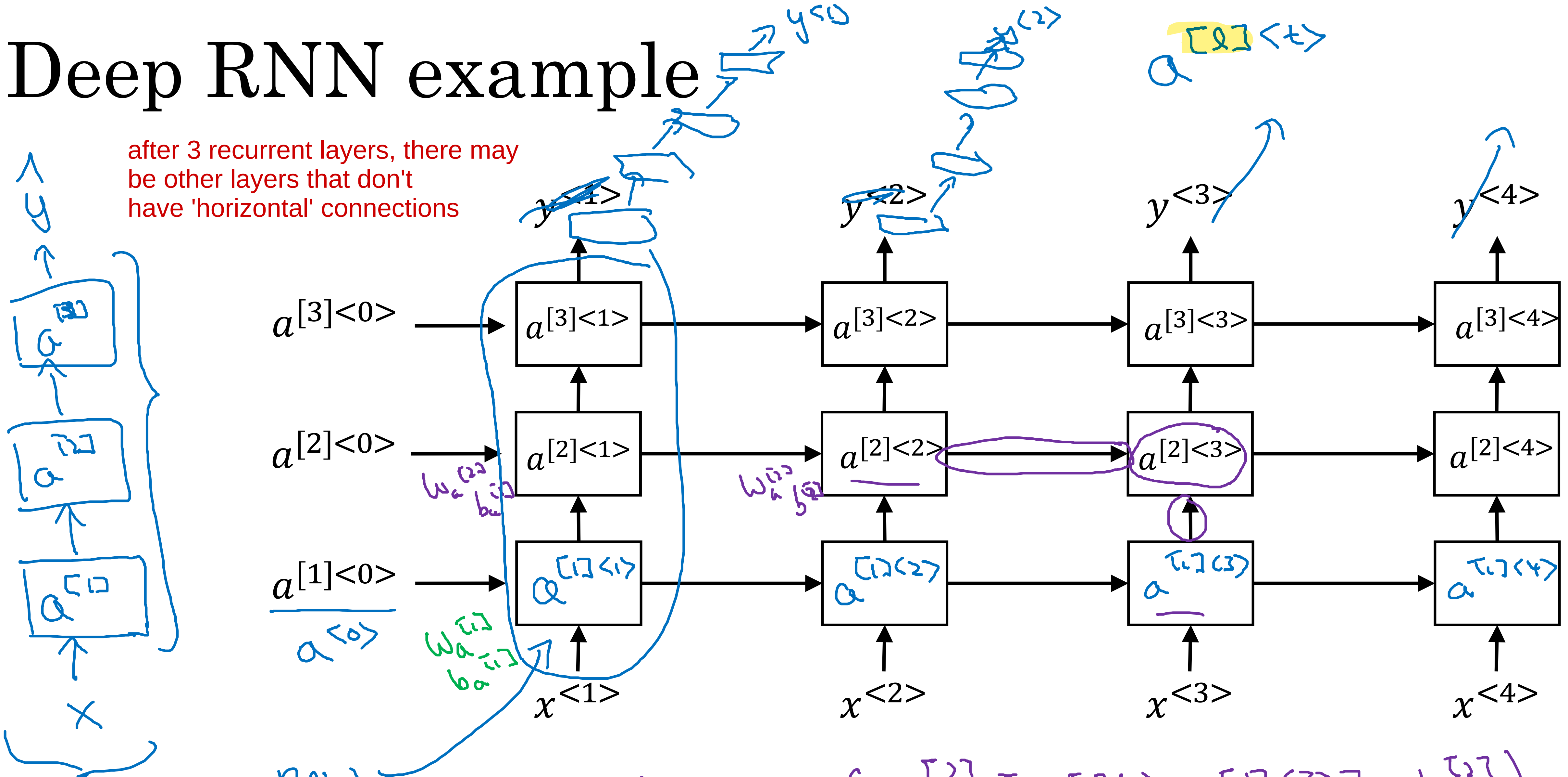
deeplearning.ai

Recurrent Neural Networks

Deep RNNs

Deep RNN example

after 3 recurrent layers, there may be other layers that don't have 'horizontal' connections



RNN
GRU
LSTM

BRNN

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

Each layer has its own W, b parameters - same as before. (Same within layer, diff. across layers)

Andrew Ng

NLP and Word Embeddings

Word representation

deeplearning.ai

Word representation

V = [a, aaron, ..., zulu, <UNK>]

$|V| = 10{,}000$

1-hot representation

$$
\begin{array}{cccccc}
\text{Man} & \text{Woman} & \text{King} & \text{Queen} & \text{Apple} & \text{Orange} \\
(5391) & (9853) & (4914) & (7157) & (456) & (6257) \\
\begin{bmatrix}0\\0\\0\\0\\\vdots\\1\\\vdots\\0\\0\end{bmatrix} &
\begin{bmatrix}0\\0\\0\\0\\0\\\vdots\\1\\\vdots\\0\end{bmatrix} &
\begin{bmatrix}0\\0\\0\\\vdots\\1\\\vdots\\0\\0\\0\end{bmatrix} &
\begin{bmatrix}0\\0\\0\\0\\0\\\vdots\\1\\\vdots\\0\end{bmatrix} &
\begin{bmatrix}0\\\vdots\\1\\\vdots\\0\\0\\0\\0\\0\end{bmatrix} &
\begin{bmatrix}0\\0\\0\\0\\0\\\vdots\\1\\\vdots\\0\end{bmatrix} \\
O_{5391} & O_{9853} & & & &
\end{array}
$$

I want a glass of $\boxed{\text{orange \_\_juice\_\_}}$.

I want a glass of $\boxed{\text{apple \_\_?\_\_}}$.

OHE doesn't make it easy generalize relationships
- because inner product of any two different OHE is 0
- even if the algorithm learns it should be 'juice' after orange, it cannot translate that to apple, and will have to relearn that for apple

Featurized representation: word embedding

Makes generalization easier

Lateral Features	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	0.7	0.69	0.03	-0.02
Food	0.04	0.01	0.02	0.01	0.95	0.97
⋮	⋮	⋮				
size						
cost						
alive						
verb						

300

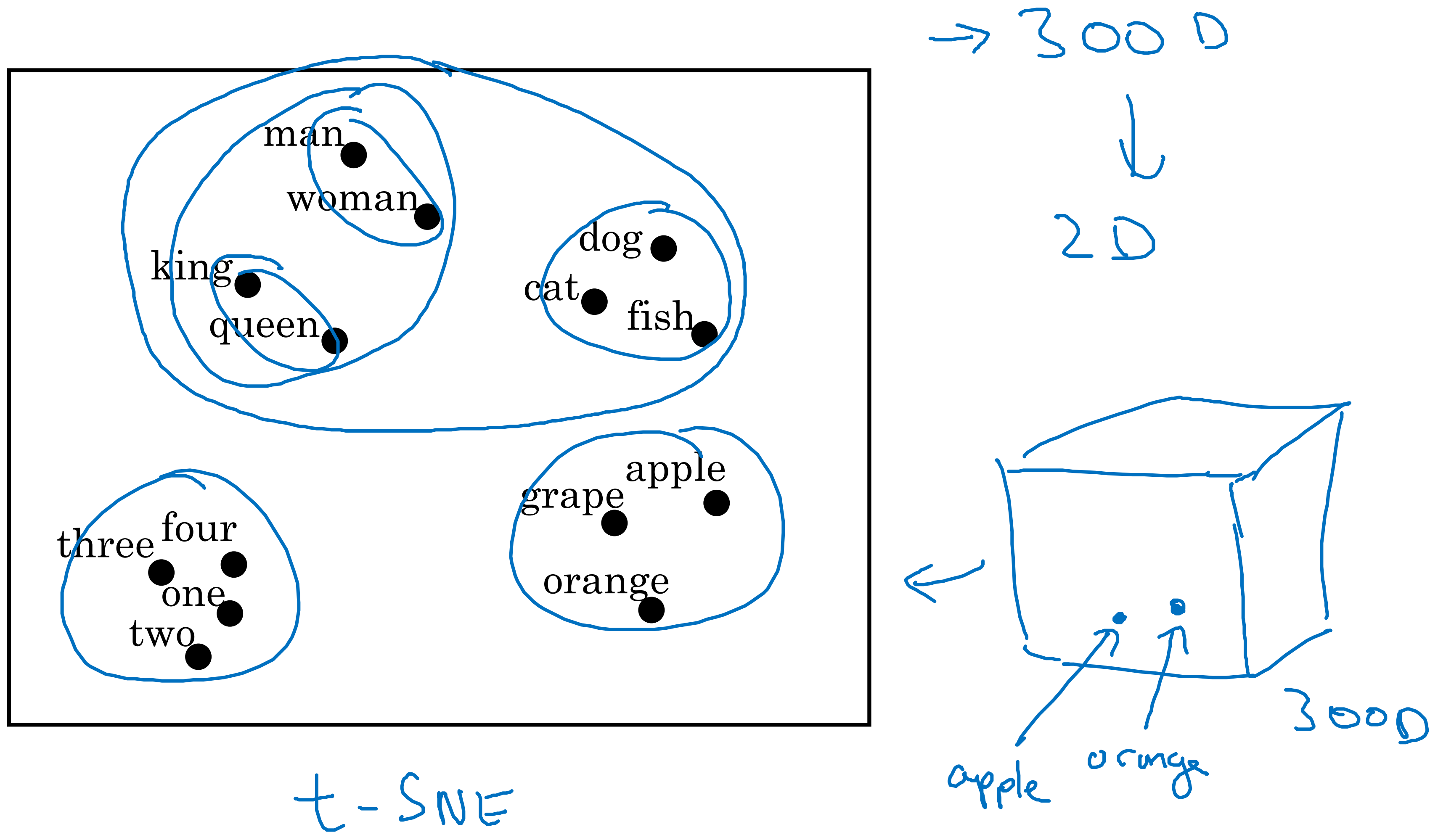
e_{5391} e_{9853}

I want a glass of orange juice.

I want a glass of apple juice.

Visualizing word embeddings

t-SNE algorithm to visualize
300D feature space in 2D graph



[van der Maaten and Hinton., 2008. Visualizing data using t-SNE]

Andrew Ng

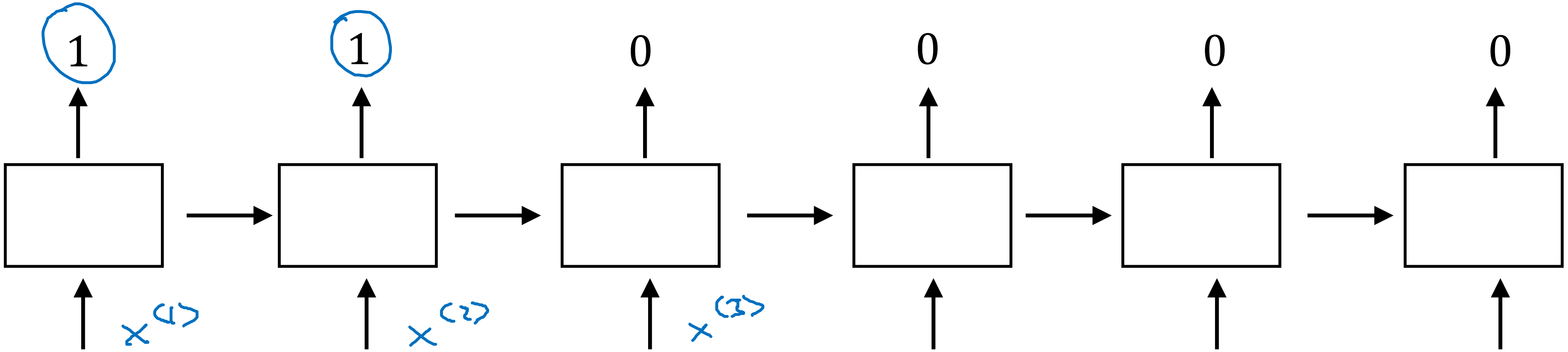
deeplearning.ai

NLP and Word Embeddings

Using word embeddings

Named entity recognition example

Word embeddings algorithm may train on large corpus of unlabeled text - to learn similarities



Sally Johnson is an orange farmer

Robert Lin is ~~an~~ ~~apple farmer~~

→ 1B words – 100B words

→ 100k words

a

durian

cultivator.

BRNN

Andrew Ng

Transfer learning and word embeddings

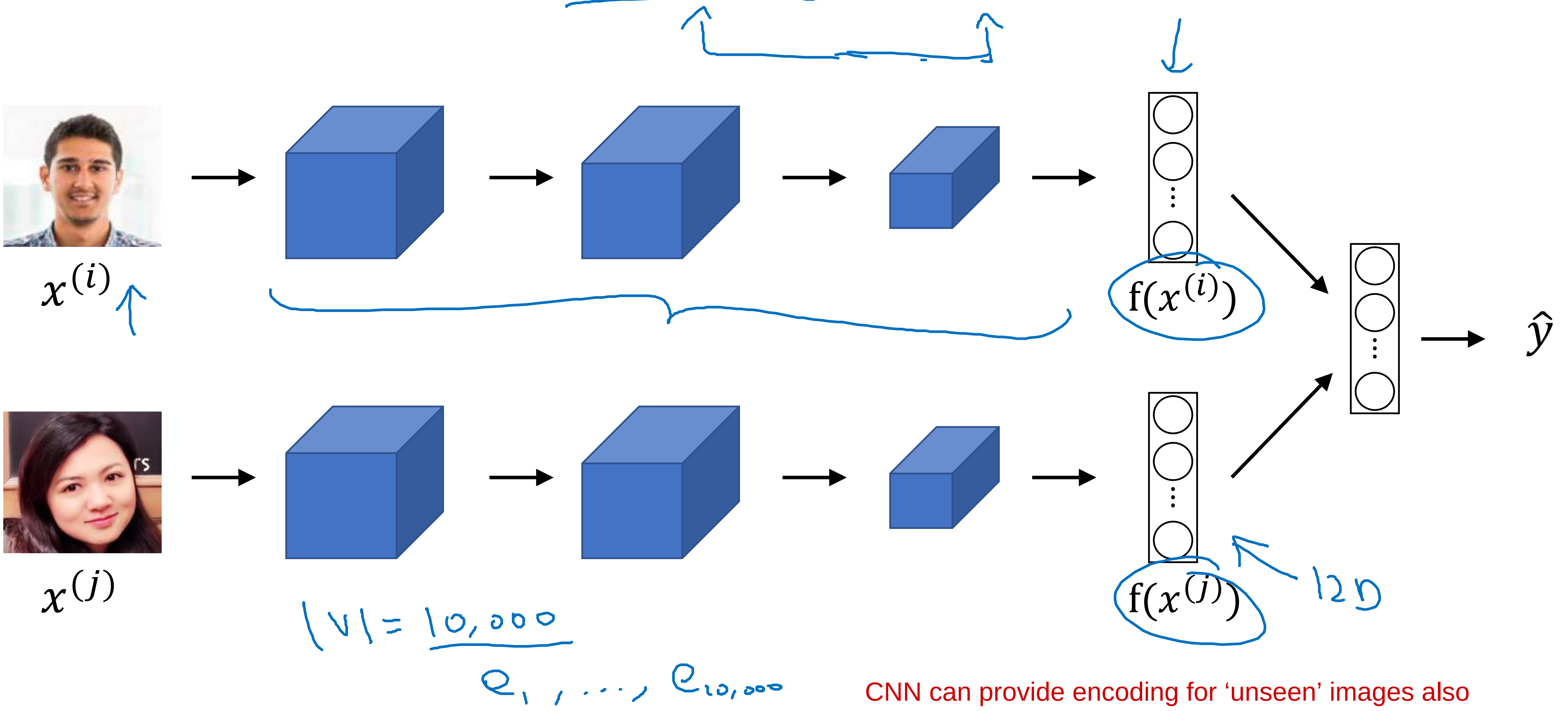
A 1. Learn word embeddings from large text corpus. (1-100B words)

(Or download pre-trained embedding online.)

B 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



CNN can provide encoding for ‘unseen’ images also
Word Embedding can only provide embedding for the seen words / fixed vocabulary

[Taigman et. al., 2014. DeepFace: Closing the gap to human level performance]

Andrew Ng

deeplearning.ai

NLP and Word Embeddings

Properties of word embeddings

Analogies

	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

e_{5391}

e_{man} e_{woman}

Man → Woman as King →? Queen

$e_{man} - e_{woman} \approx e_{king} - e_{?}$

? ← queen

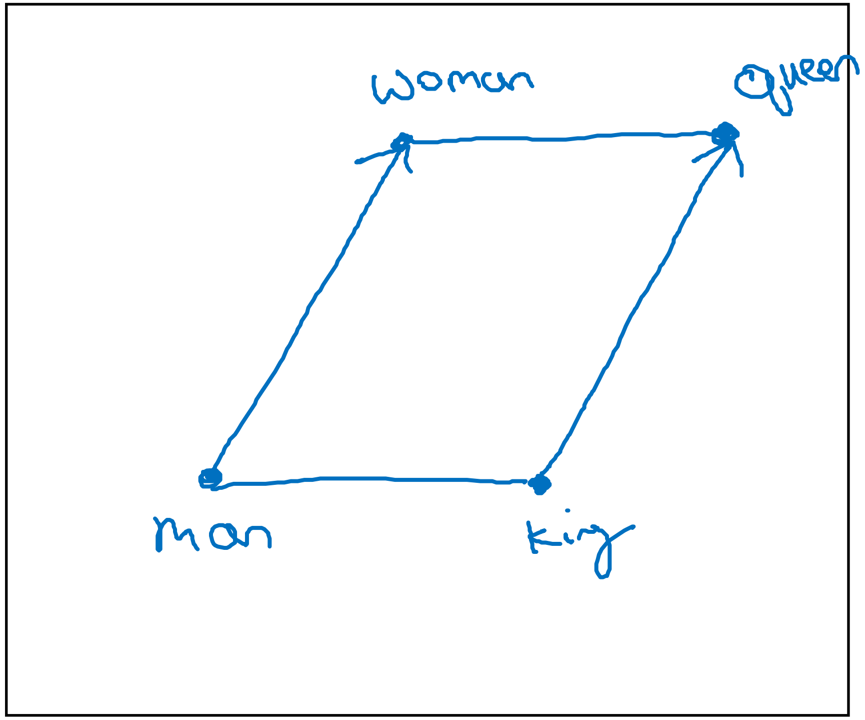
$e_{man} - e_{woman} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

$e_{king} - e_{queen} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

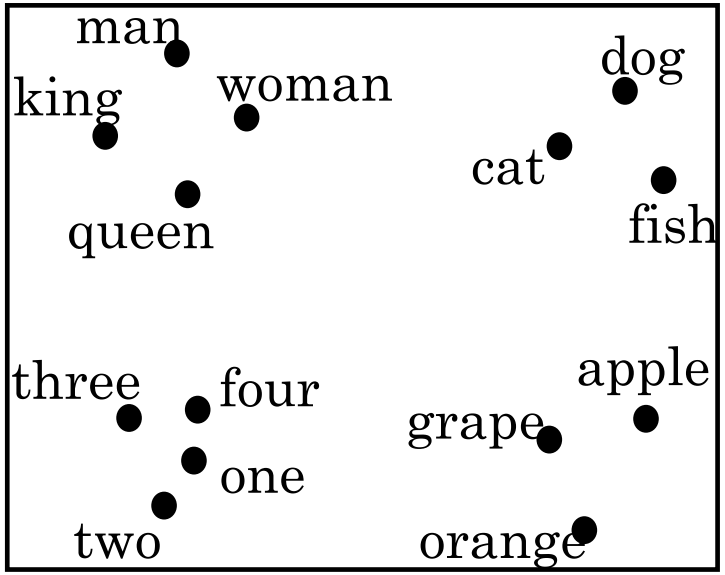
[Mikolov et. al., 2013, Linguistic regularities in continuous space word representations]

Andrew Ng

Analogies using word vectors



300 D



t-SNE

300 → 2D

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

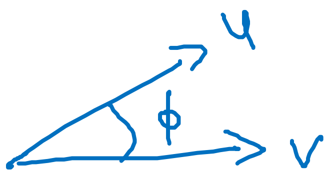
Find word w: $\arg\max_w \boxed{sim(e_w, e_{king} - e_{man} + e_{woman})}$

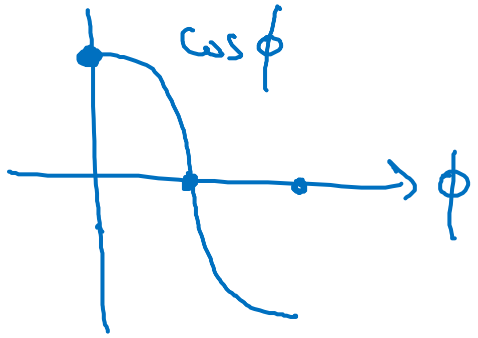
30 – 75%

Cosine similarity

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

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





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

Man:Woman as Boy:Girl

Ottawa:Canada as Nairobi:Kenya

Big:Bigger as Tall:Taller

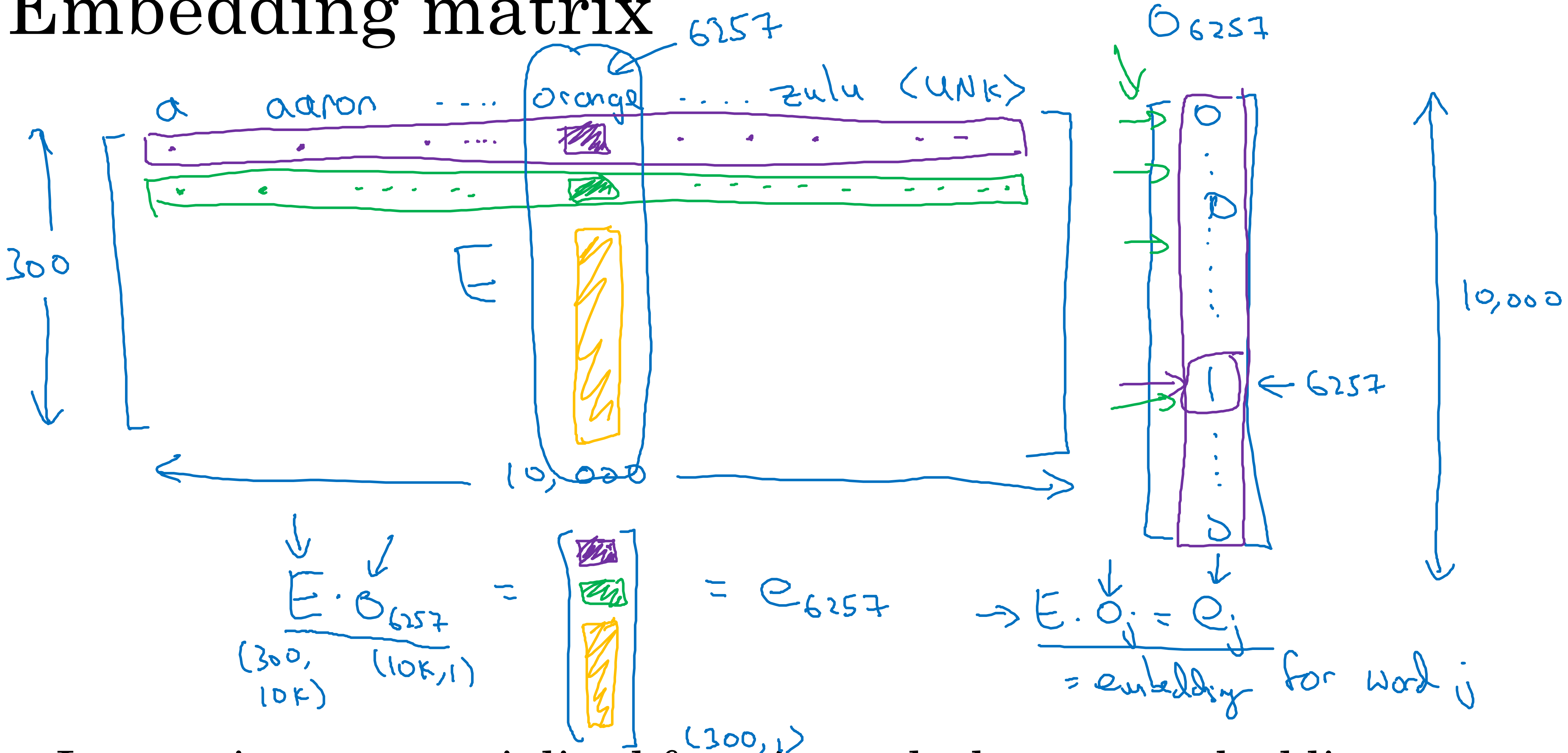
Yen:Japan as Ruble:Russia

NLP and Word Embeddings

deeplearning.ai

Embedding matrix

Embedding matrix



In practice, use specialized function to look up an embedding.

→ Embedding

deeplearning.ai

NLP and Word Embeddings

Learning word embeddings

Neural language model

An earlier approach to learn word embedding matrix E, by training a language model.

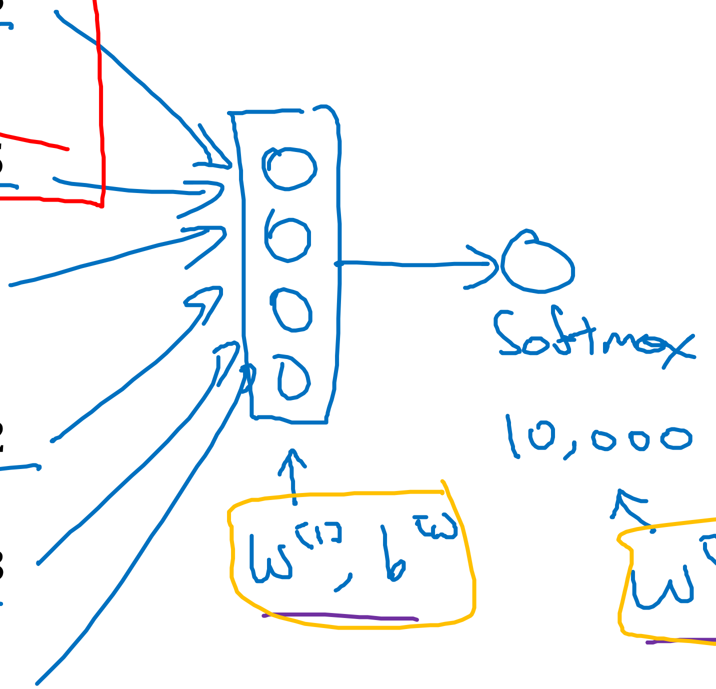
4

I	want	a	glass	of	orange	juice.
4343	9665	1	3852	6163	6257	

apple juice.

$e_{4343} = E\, o_{4343}$

I	o_{4343}	$\longrightarrow$ E $\longrightarrow$	e_{4343}
want	o_{9665}	$\longrightarrow$ E $\longrightarrow$	e_{9665}
a	o_{1}	$\longrightarrow$ E $\longrightarrow$	e_{1}
glass	o_{3852}	$\longrightarrow$ E $\longrightarrow$	e_{3852}
of	o_{6163}	$\longrightarrow$ E $\longrightarrow$	e_{6163}
orange	o_{6257}	$\longrightarrow$ E $\longrightarrow$	e_{6257}



$W^{[1]}, b^{[1]}$ $W^{[2]}, b^{[2]}$

~~1800~~ 1200

6 words, 300 length each.. or we may only select "4" word history

[Bengio et. al., 2003, A neural probabilistic language model]

Andrew Ng

Other context/target pairs

For language model, context may be few words preceding the "target'
For word embedding in general, using simpler/ different context works as well.

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

context

target

Context: Last 4 words.

4 words on left & right

a glass of orange ? to go along with

Last 1 word

orange ?

Nearby 1 word

glass ?

skip gram

skip gram model

Andrew Ng

deeplearning.ai

NLP and Word Embeddings

Word2Vec

Skip-grams

Randomly pick a context word, and then pick a target word within some range, say +/- 5 words window

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

Context	Target
orange	juice
orange	glass
orange	my

[Mikolov et. al., 2013. Efficient estimation of word representations in vector space.]

Model

One-hot vector (1,10000) x E matrix (10000, 300) => (1,300) Embedding (e_c)
=> Theta_t = [W, b] (bias can be ignored) (300,10000)
=> Softmax (1, 10000) = prob for each 'target' word given 'context' word

Vocab size = 10,000k

$x \longrightarrow \underline{y}$

Context c ("orange") 6257 $\longrightarrow$ Target $\underline{t}$ ("juice") 4834

$$O_c \longrightarrow E \longrightarrow e_c \longrightarrow O \longrightarrow \hat{y}$$

Softmax

$e_c = E\,O_c$

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

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

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

Andrew Ng

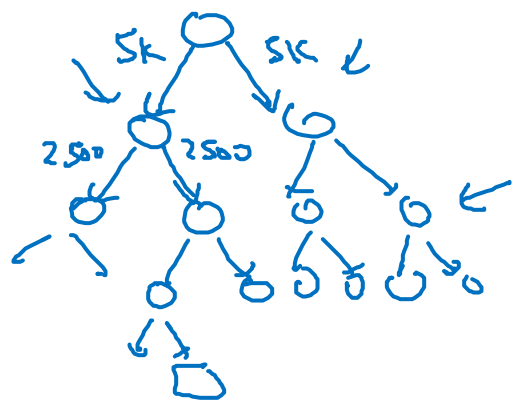
Problems with softmax classification

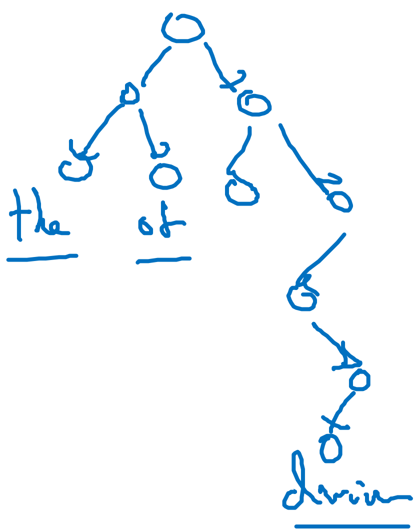
computationally expensive +
makes it harder to scale O(n)

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

Hierarchical softmax.

log |V|





making a tree of classifiers O(logn)

How to sample the context c?

t

→ the, of, a, and, to, . . . remove such common words c → t

→ orange, apple, durian P(c)

durian

NLP and Word Embeddings

Negative sampling

deeplearning.ai

Defining a new learning problem

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

Create a new supervised problem of Positive and Negative examples

Positive sample: sample context word + sample target word from a 5/10 word window

Negative sample: sample target word randomly from dictionary

x → y

Context	word	target?
orange	juice	1
orange	king	0
orange	book	0
orange	the	0
orange	of	0

k

k = 5-20 smaller datasets

k = 2-5 larger dataset

[Mikolov et. al., 2013. Distributed representation of words and phrases and their compositionality]

Model

Instead of 1 softmax, learn a bunch of binary classifiers

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

10,000-way softmax

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

context	word	target?
orange	juice	1
orange	king	0
orange	book	0
orange	the	0
orange	of	0

x: context, word (c, t); y: target?

k

instead of training all 10,000 on each iteration, we only train 'k+1' on each iteration. Scalable.

10,000 binary classification problem

k+1

Orange 6257

10000, 300

$O_{6257} \rightarrow E \rightarrow e_{6257}$

1,10000

1, 300

10,000 classifiers (300,1)

juice?

king

10,000

Andrew Ng

Selecting negative examples

$$\underbrace{\text{context} \quad \text{word}}_{x} \quad \underbrace{\text{target?}}_{y}$$

context	word	target?
orange	juice	1
orange	king	0
orange	book	0
orange	the	0
orange	of	0

t

the , of, and, ...

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

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

How to sample negative words?
(1) using natural frequency: (-ve) a, the, an, of etc.
(2) uniform sampling: (-ve) non-representative of english language
Empirical solution - pick somewhere in between, using a power of 3/4.

Andrew Ng

NLP and Word Embeddings

deeplearning.ai

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~~ j appears in context of ~~j~~ i.

$c \quad t \qquad t \qquad c$

$X_{ij} = X_{ji}$

similar if we use word window
not same if we use 'immediately preceding' as context

[Pennington et. al., 2014. GloVe: Global vectors for word representation]

Andrew Ng

Model

$$\text{minimize} \sum_{i=1}^{10,000} \sum_{j=1}^{10,000} \underbrace{f(X_{ij})}_{\text{weighting term}} \left(\theta_i^T e_j + b_i + b_j' - \log X_{ij}\right)^2$$

$t \quad c$

"$\theta_t^T e_c$"

0?

ignore cases
where Xij = 0
use 0 log(0) = 0

$f(X_{ij}) = 0$ if $X_{ij} = 0$. "$0 \log 0$" $= 0$

this, is, of, a, ...

durian

reduces weight
for frequent
words, increases
with infrequent
words

θ_i, e_j are symmetric

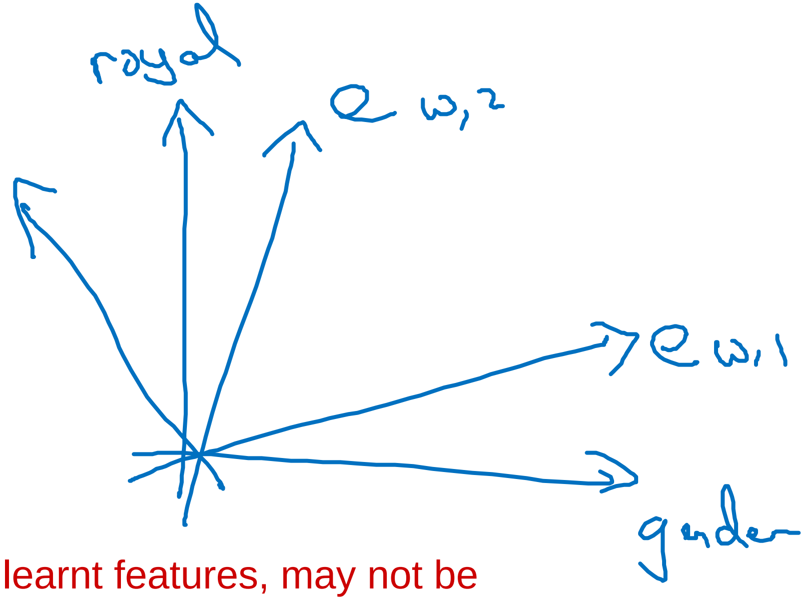
$$e_w^{(final)} = \frac{e_w + \theta_w}{2}$$

unlike before, in this formulate, theta and e pay symmetric role,
therefore after training we average them.

Andrew Ng

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



learnt features, may not be orthogonal or interpretable

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

$$(A\theta_i)^T(A^{-T}e_j) = \theta_i^T \cancel{A^T} \cancel{A^{-T}} e_j$$

Andrew Ng

deeplearning.ai

NLP and Word Embeddings

Sentiment classification

Sentiment classification problem

$x \longrightarrow y$

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

Andrew Ng

Simple sentiment classification model

The	dessert	is	excellent
8928	2468	4694	3180

★★★★☆

averaging take care of different length of sentences. ignores sequence though.

The $\quad o_{8928} \longrightarrow E \longrightarrow e_{8928}$

desert $\quad o_{2468} \longrightarrow E \longrightarrow e_{2468}$

is $\quad o_{4694} \longrightarrow E \longrightarrow e_{4694}$

excellent $\quad o_{3180} \longrightarrow E \longrightarrow \underbrace{e_{3180}}_{3000}$

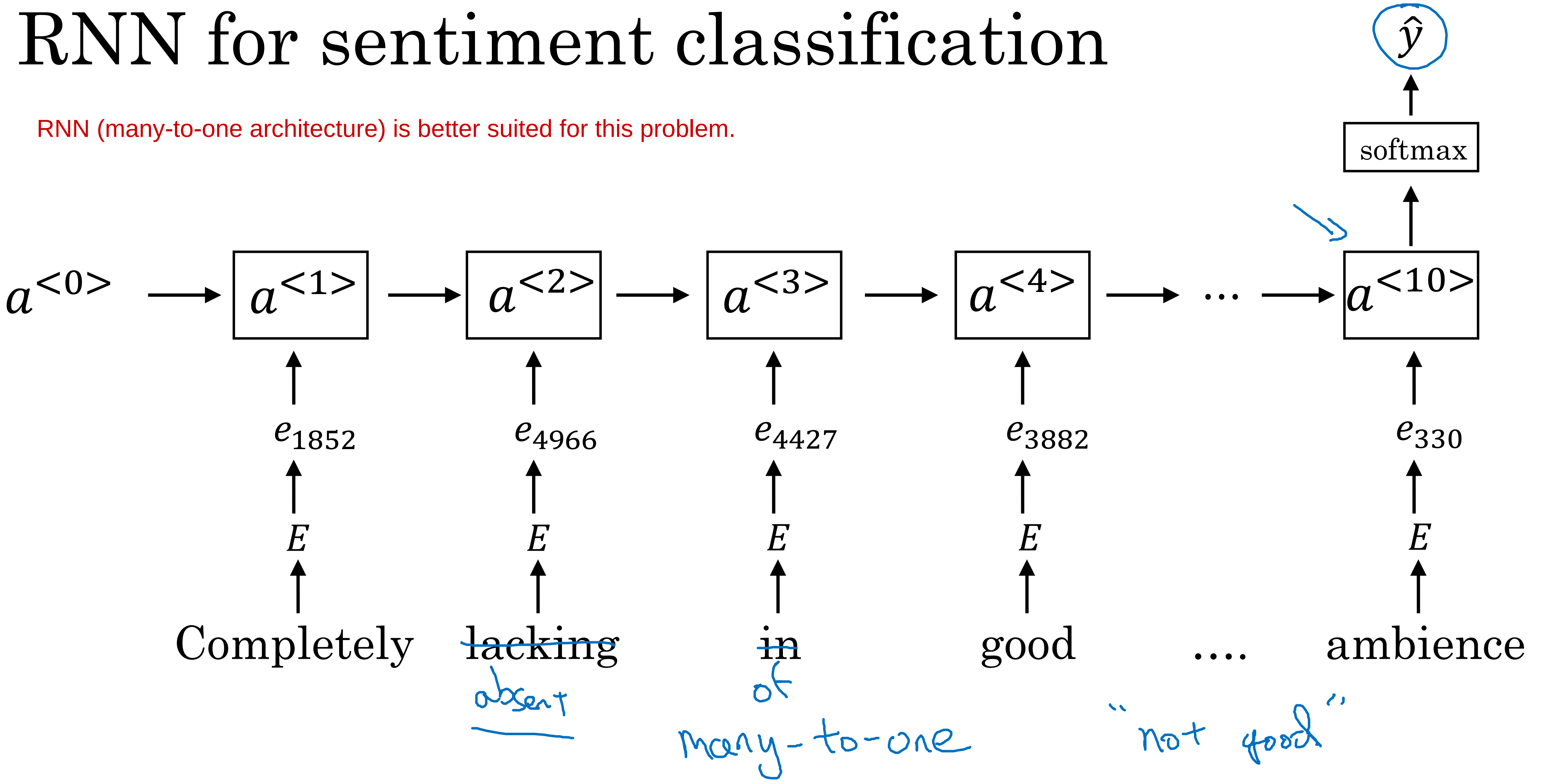
E: 100 B words

$\boxed{\text{Avg}}$ (3000) $\longrightarrow$ O (softmax, 1-5) $\longrightarrow \hat{y}$

"Completely lacking in $\boxed{\text{good}}$ taste, $\boxed{\text{good}}$ service, and $\boxed{\text{good}}$ ambience."

RNN for sentiment classification

RNN (many-to-one architecture) is better suited for this problem.



Andrew Ng

NLP and Word Embeddings

deeplearning.ai

Debiasing word embeddings

The problem of bias in word embeddings

Man:Woman as King:Queen

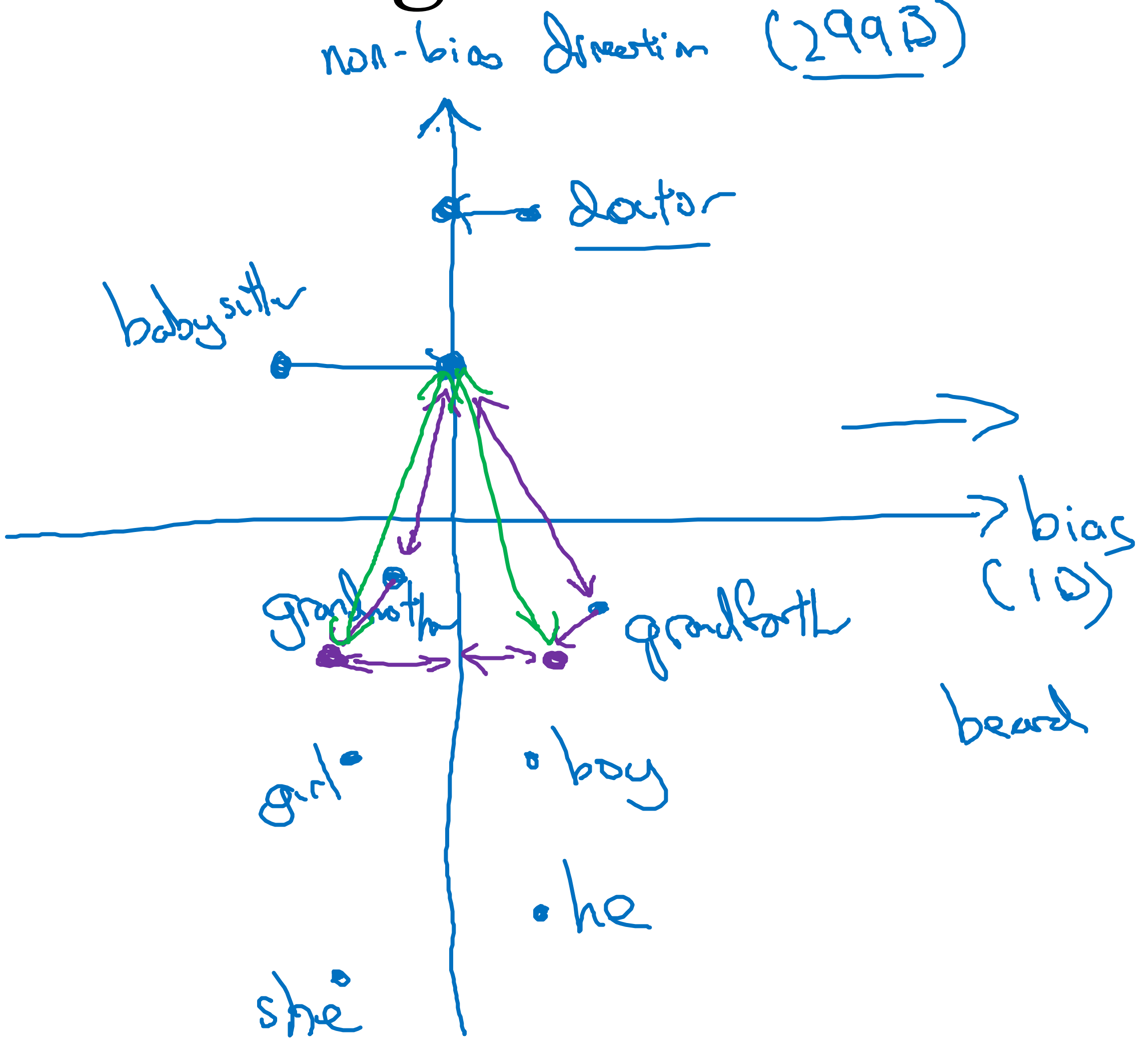
Man:Computer_Programmer as Woman: Homemaker

Father:Doctor as Mother: Nurse

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

[Bolukbasi et. al., 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings]

Addressing bias in word embeddings



1. Identify bias direction.

$e_{he} - e_{she}$

$e_{male} - e_{female}$

$\vdots$

average

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

3. Equalize pairs.

→ grandmother − grandfather

girl − boy

[Bolukbasi et. al., 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings]

Andrew Ng

Sequence to sequence models

deeplearning.ai

Basic models

Sequence to sequence model

$x^{<1>}$ $x^{<2>}$ $x^{<3>}$ $x^{<4>}$ $x^{<5>}$

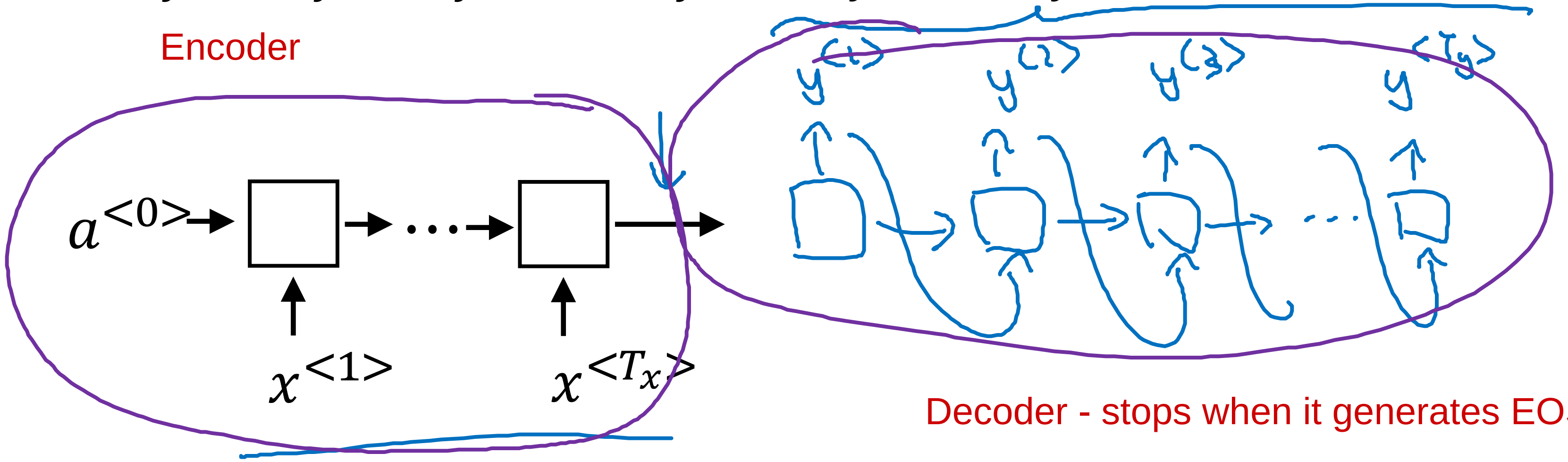
Jane visite l'Afrique en septembre

Remarkable thing is - this model works when enough training data is applied

$\longrightarrow$ Jane is visiting Africa in September.

$y^{<1>}$ $y^{<2>}$ $y^{<3>}$ $y^{<4>}$ $y^{<5>}$ $y^{<6>}$

Encoder



Decoder - stops when it generates EOS token

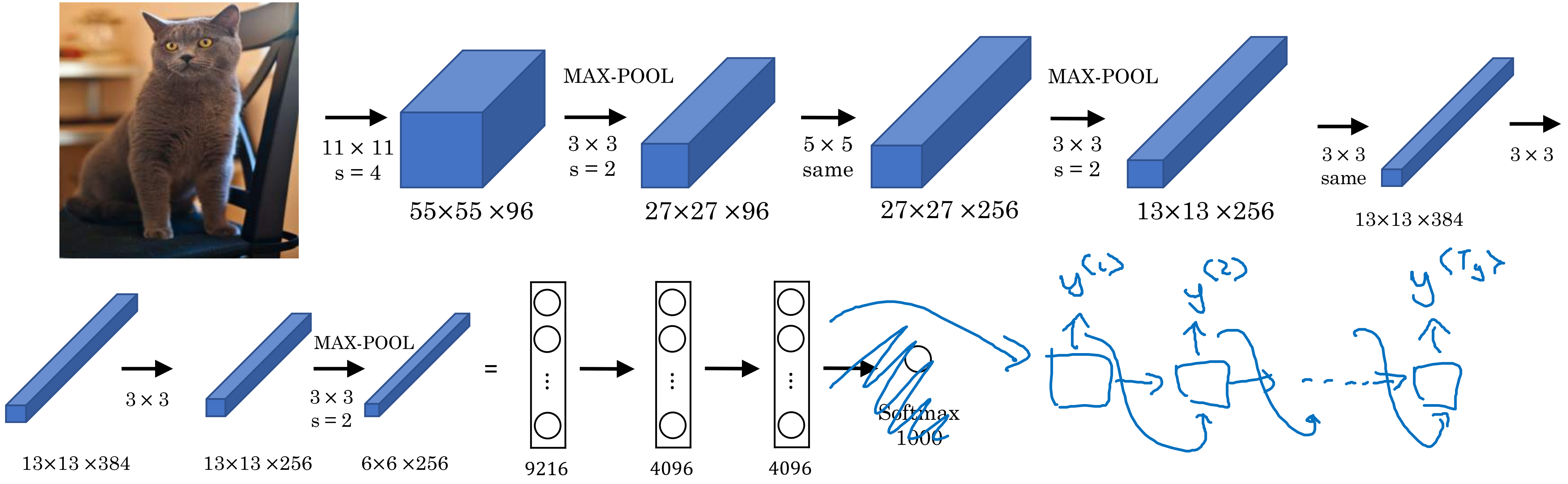
[Sutskever et al., 2014. Sequence to sequence learning with neural networks]

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

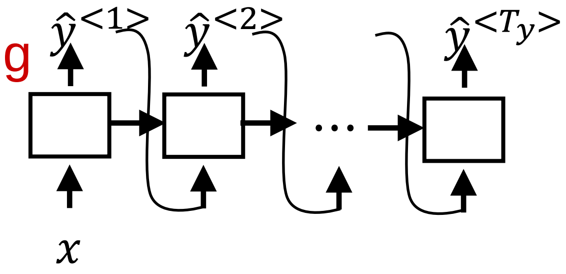
Image captioning

$y^{<1>}$ $y^{<2>}$ $y^{<3>}$ $y^{<4>}$ $y^{<5>}$ $y^{<6>}$

A cat sitting on a chair



similar architecture to last slide also works for image captioning



[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]

deeplearning.ai

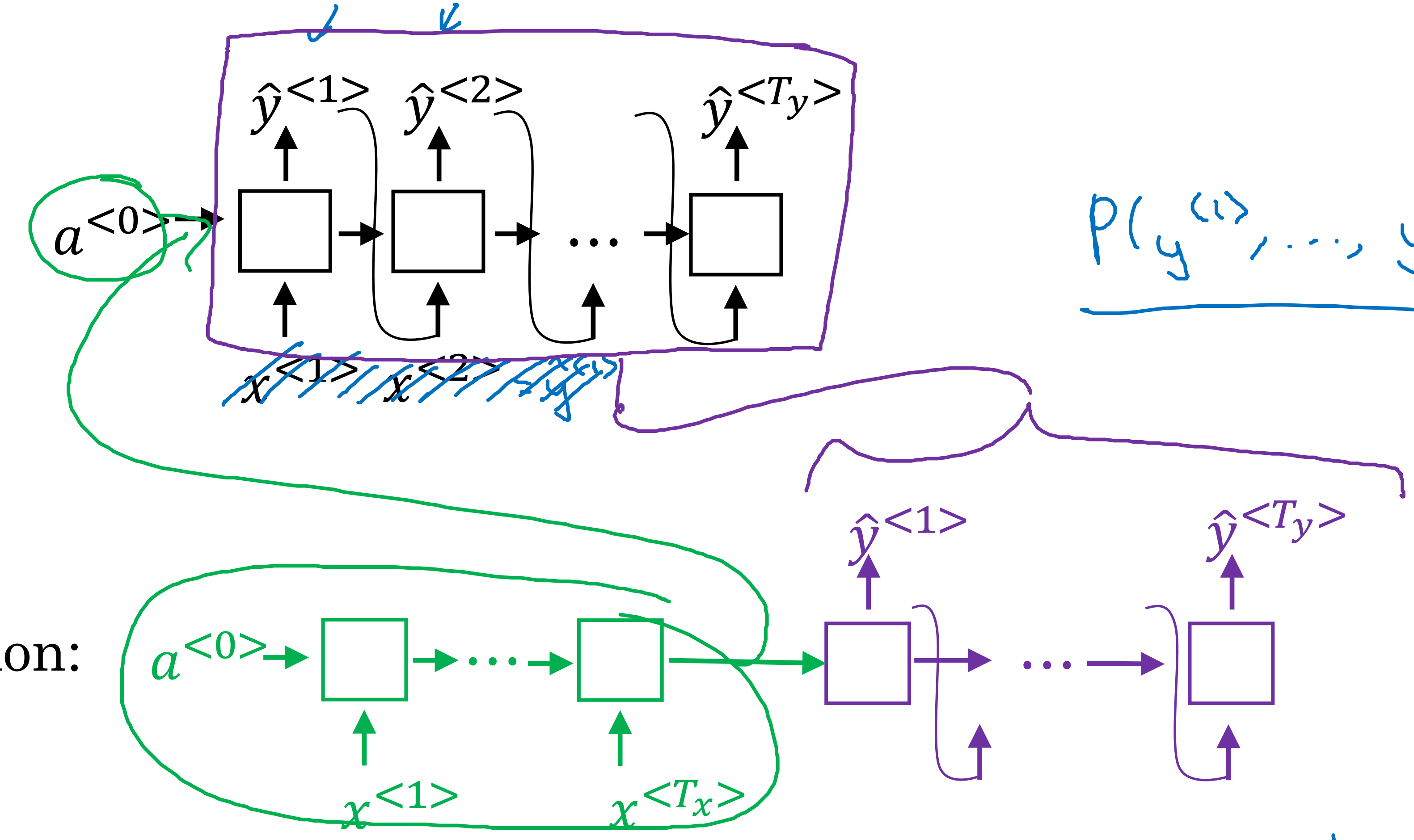
Sequence to sequence models

How to generate most likely sequence, not just a random sequence

Picking the most likely sentence

Machine translation as building a conditional language model

Language model:



$P(y^{<1>}, \ldots, y^{<T_y>})$

Machine translation:

decoder network is similar to 'conditional language model' - conditional on input French sentence

"Conditional language model"

$P(y^{<1>}, \ldots, y^{<T_y>} \mid x^{<1>}, \ldots, x^{<T_x>})$

Andrew Ng

Finding the most likely translation

Jane visite l'Afrique en septembre.

$$P(\underbrace{y^{<1>}, \ldots, y^{<T_y>}}_{\text{English}} \mid \underset{\text{French}}{x})$$

→ 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.

$$\arg\max_{y^{<1>},\ldots,y^{<T_y>}} P(y^{<1>}, \ldots, y^{<T_y>} \mid x)$$

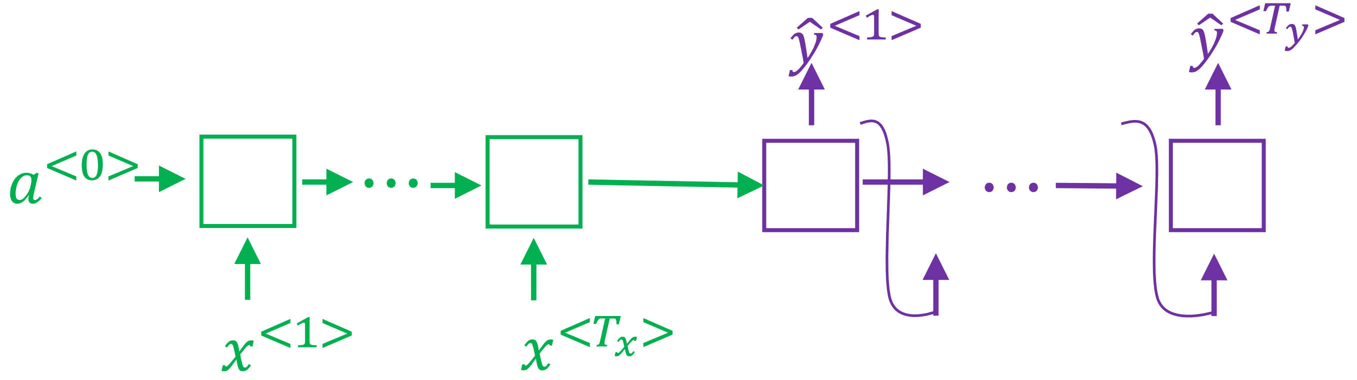
How to maximize this OVERALL prob.??

- greedy search - pick the most likely word at each stage
- beam search

Andrew Ng

Why not a greedy search?

$P(\hat{y}^{<1>} | x)$



$$\arg\max_{y} P(\underbrace{\hat{y}^{<1>}, \hat{y}^{<2>},}\ \ldots, \hat{y}^{<T_y>} | x)$$

10,000

10

$10{,}000^{10}$

$P(y|x)$

→ Jane is visiting Africa in September.

→ Jane is going to be visiting Africa in September.

P(Jane is going | x) > P(Jane is visiting | x)

Andrew Ng

deeplearning.ai

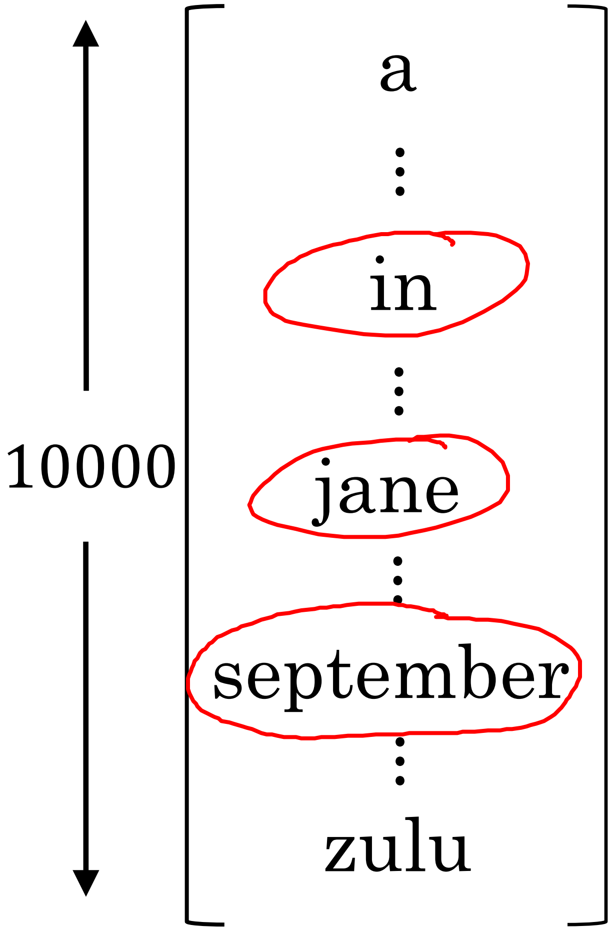
Sequence to sequence models

Beam search

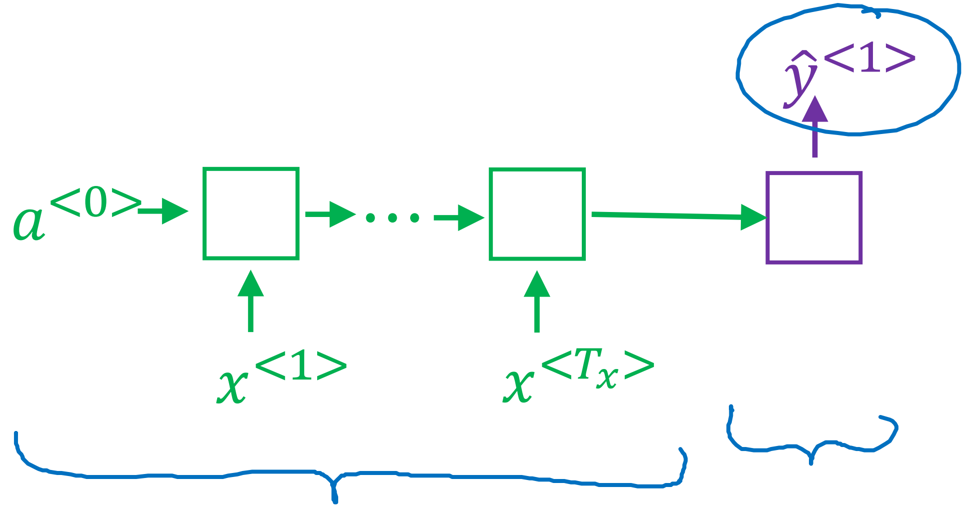
Beam search algorithm

$B = 3$ (beam width)

Step 1

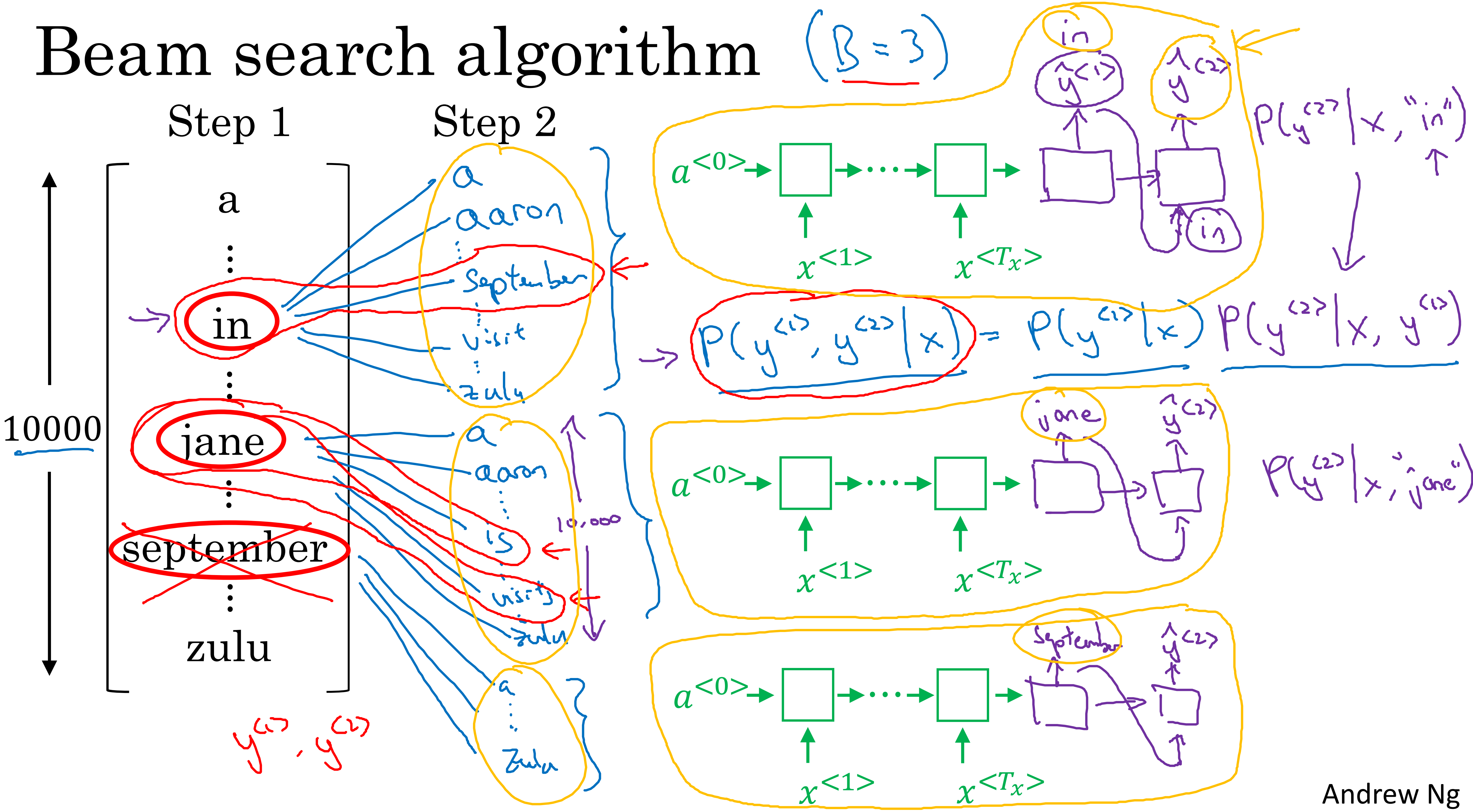


$\rightarrow P(y^{<1>} \mid x)$



- 3 best options are selected at each stage
- create 3 copies of the network at each stage

Beam search algorithm

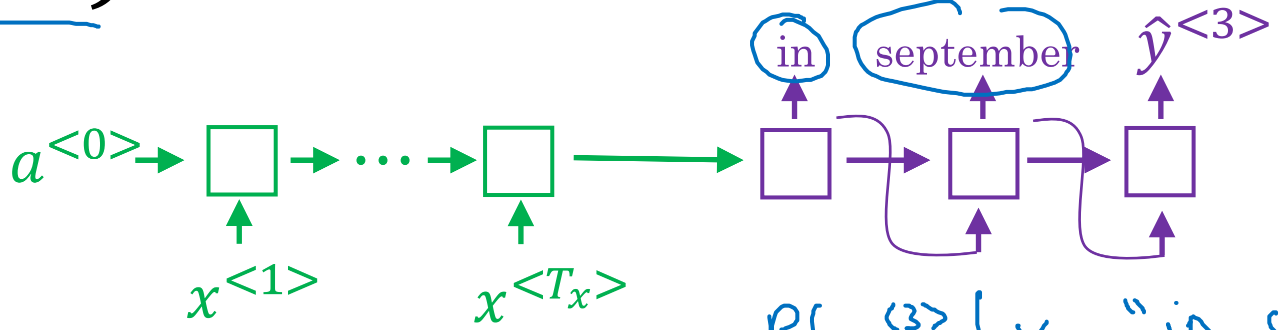


Andrew Ng

Beam search ($B = 3$)

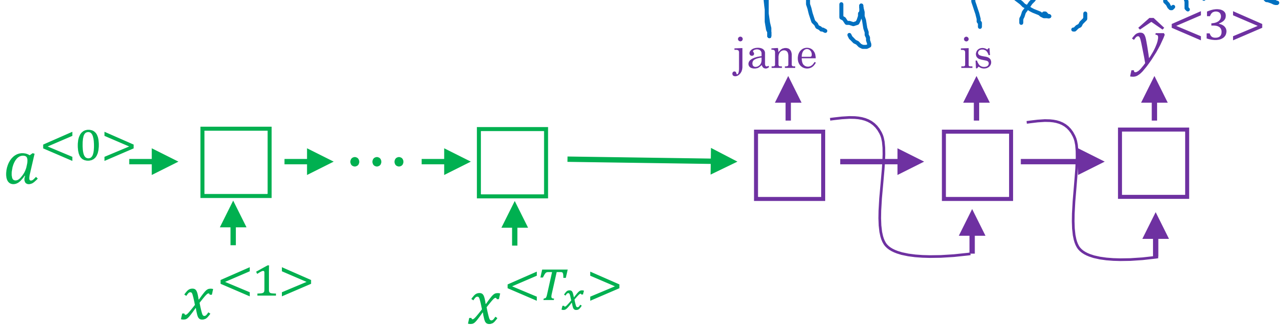
$B = 1 \rightsquigarrow$ greedy search

in september
a
aaron
⋮
jane
⋮
zulu

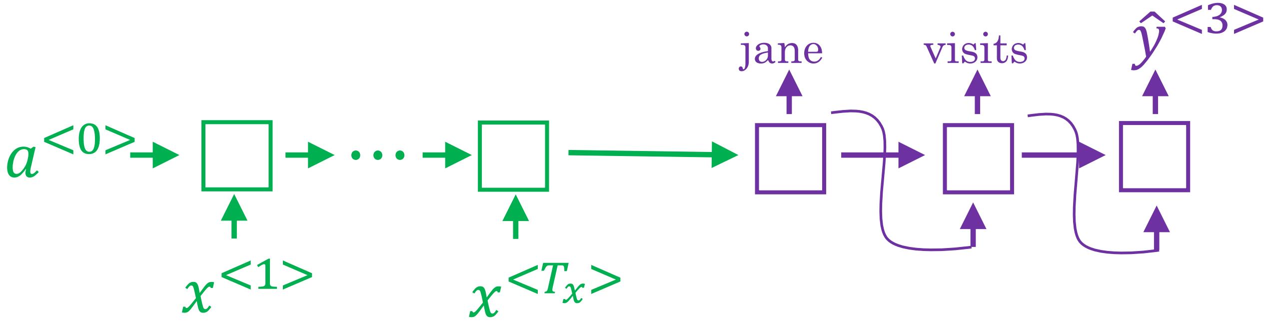


$P(y^{<3>} \mid x, \text{"in september"})$

jane is
a
⋮
visits
⋮
zulu



jane visits
a
⋮
africa
⋮
zulu



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

jane visits africa in september. <EOS>

Andrew Ng

deeplearning.ai

Sequence to sequence models

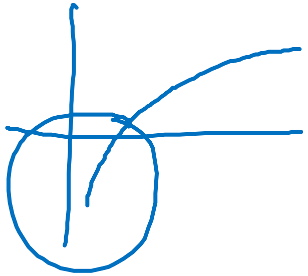
Refinements to beam search

Length normalization

$$p(y^{<1>} \dots y^{<T_y>} \mid x) = P(y^{<1>} \mid x)\, P(y^{<2>} \mid x, y^{<1>}) \dots P(y^{<T_y>} \mid x, y^{<1>}, \dots, y^{<T_y-1>})$$

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

log



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

$P(y|x) \leftarrow$

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

Changes:
probabilities are small numbers
(1) Log changes it to addition, and reduces error from numerical precision

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

joint probabilities will reduce as more words are added
(2) normalize by # of words

$\alpha = 0.7$ $\alpha = 1$ $\alpha = 0$

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

Andrew Ng

Beam search discussion

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

Beam width B?

1 → 3 → 10, 100, 1000, → 3000

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

→ RNN

→ Beam Search

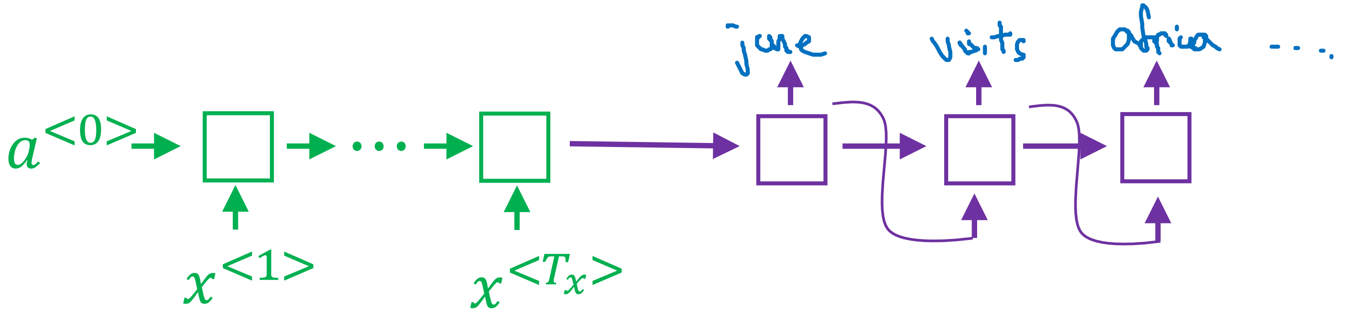
$\boxed{B\uparrow}$

Jane visite l'Afrique en septembre.

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

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

RNN computes $P(y^*|x) \; \overset{>}{\le} \; P(\hat{y}|x)$



Error analysis on beam search

$P(y^*|x)$

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

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

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

$\arg\max_{y} P(y|x)$

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

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

Conclusion: Beam search is at fault.

Same idea in ML course - calculate cost for model estimated theta and theta from another source - to see if cost function is faulty or optimization algo

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

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

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.	2×10^{-10}	1×10^{-10}	B
- - -	- - -	—	—	R
. . .	. . .	—	—	B
				R
				R
				⋮

Figures out what faction of errors are "due to" beam search vs. RNN model

Andrew Ng

deeplearning.ai

Sequence to sequence models

Bleu score (optional)

Evaluating machine translation

Bleu

bilingual evaluation understudy

French: Le chat est sur le tapis.

Reference 1: The cat is on the mat.

bilingual evaluation understudy

Reference 2: There is a cat on the mat.

MT output: the the the the the the the.

MT - machine translation

Precision: 7 / 7
each word in o/p appears in the reference

Modified precision: 2/7
it appears 2 times in first word in reference

[Papineni et. al., 2002. Bleu: A method for automatic evaluation of machine translation]

Andrew Ng

Bleu score on bigrams

instead of looking at isolated words - we look at pair of words (bigram / trigram etc.)

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
cat on	1	1
on the	1	1
the mat	1	1

$$\frac{4}{6}$$

[Papineni et. al., 2002. Bleu: A method for automatic evaluation of machine translation]

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. ($\hat{y}$)

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

every Pn will be 1, if translation is perfect

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

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

n-gram

$\sum_{n\text{-grams} \in \hat{y}} \text{Count}_{clip}(n\text{-gram})$

$\sum_{n\text{-grams} \in \hat{y}} \text{Count}(n\text{-gram})$

unigram

$\sum_{\text{Unigram} \in \hat{y}} \text{Count}_{clip}(\text{unigram})$

$\sum_{\text{Unigram} \in \hat{y}} \text{Count}(\text{unigram})$

2/7 from 2 slides back

[Papineni et. al., 2002. Bleu: A method for automatic evaluation of machine translation]

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

brevity penalty - to penalize too short of a translation (which inherently can score better, hence the penalty)

$$BP = \begin{cases} 1 & \text{if MT\_output\_length} > \text{reference\_output\_length} \\ \cancel{\exp(1 - \text{MT\_output\_length}/\text{reference\_output\_length})} & \text{otherwise} \end{cases}$$

exp(1-reference_output_length/ MT_output_length)

Bleu scored established an industry standard performance metric for these models - and allowed rapid advancements in this field.

[Papineni et. al., 2002. Bleu: A method for automatic evaluation of machine translation]

Sequence to sequence models

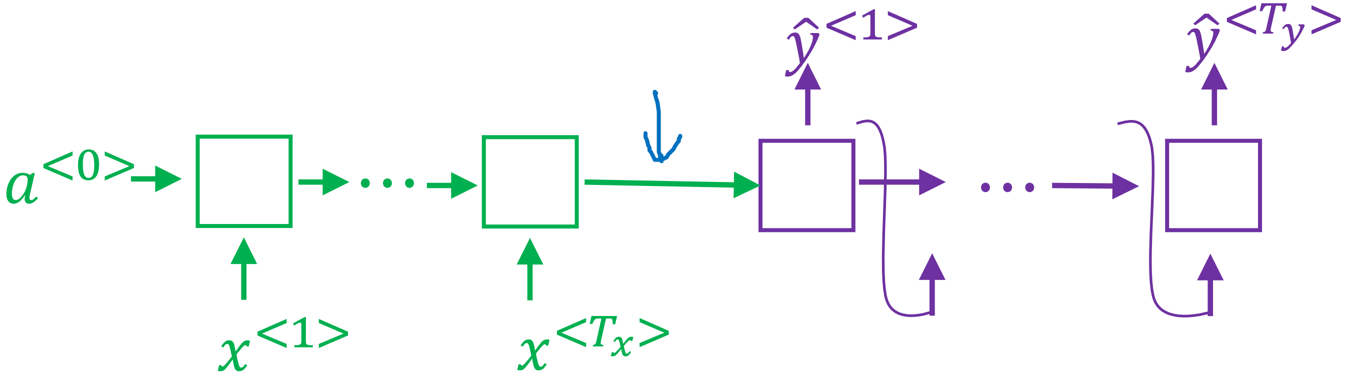
One of the most influential ideas

Attention model intuition

deeplearning.ai

The problem of long sequences

we're asking model to read in whole sentence and remember and store it in activations - so decoder can translate

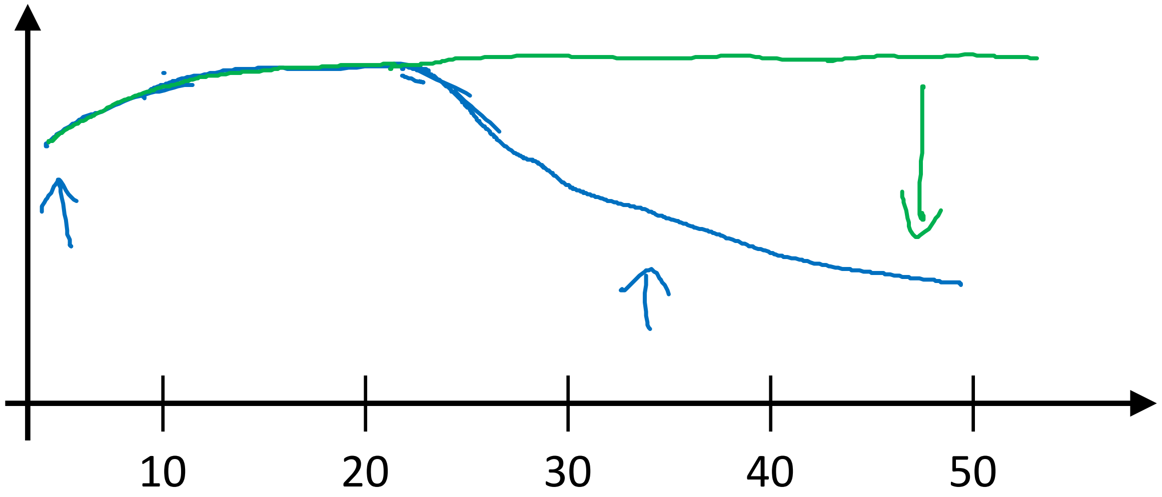


not how human does it! human would do it part by part

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.

Bleu score



Sentence length

conventional methods does poorly on longer sentences

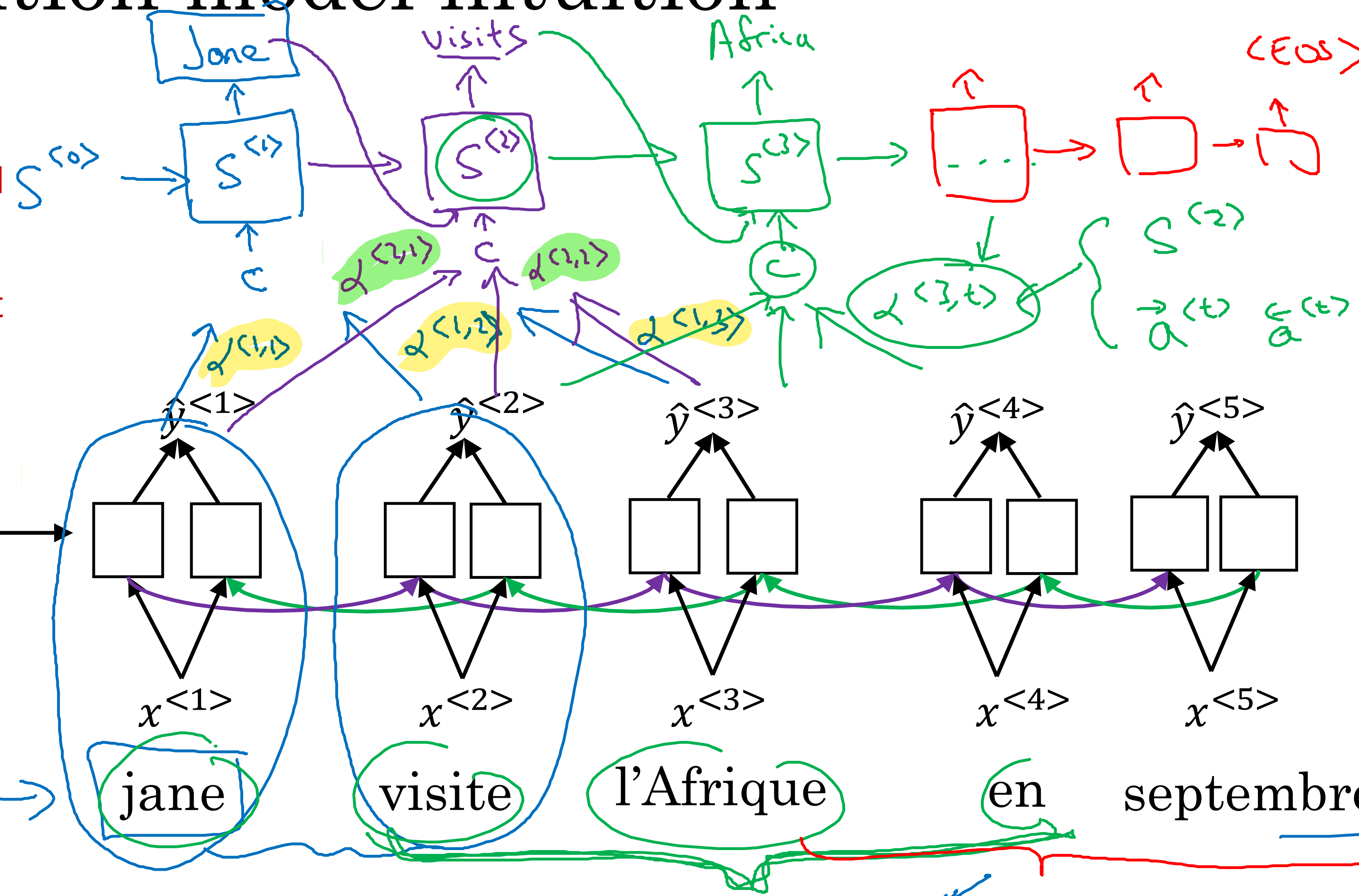
attention model can give better performance (green line)

Attention model intuition

For 1st word of translation to be Jane, we don't need the whole French sequence just the first few words.

Attention model computes set of weights (alphas)

next word will have different weights and so on.



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

attention for word 't' depends on activations at 't' + hidden state from 't-1'

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

Andrew Ng

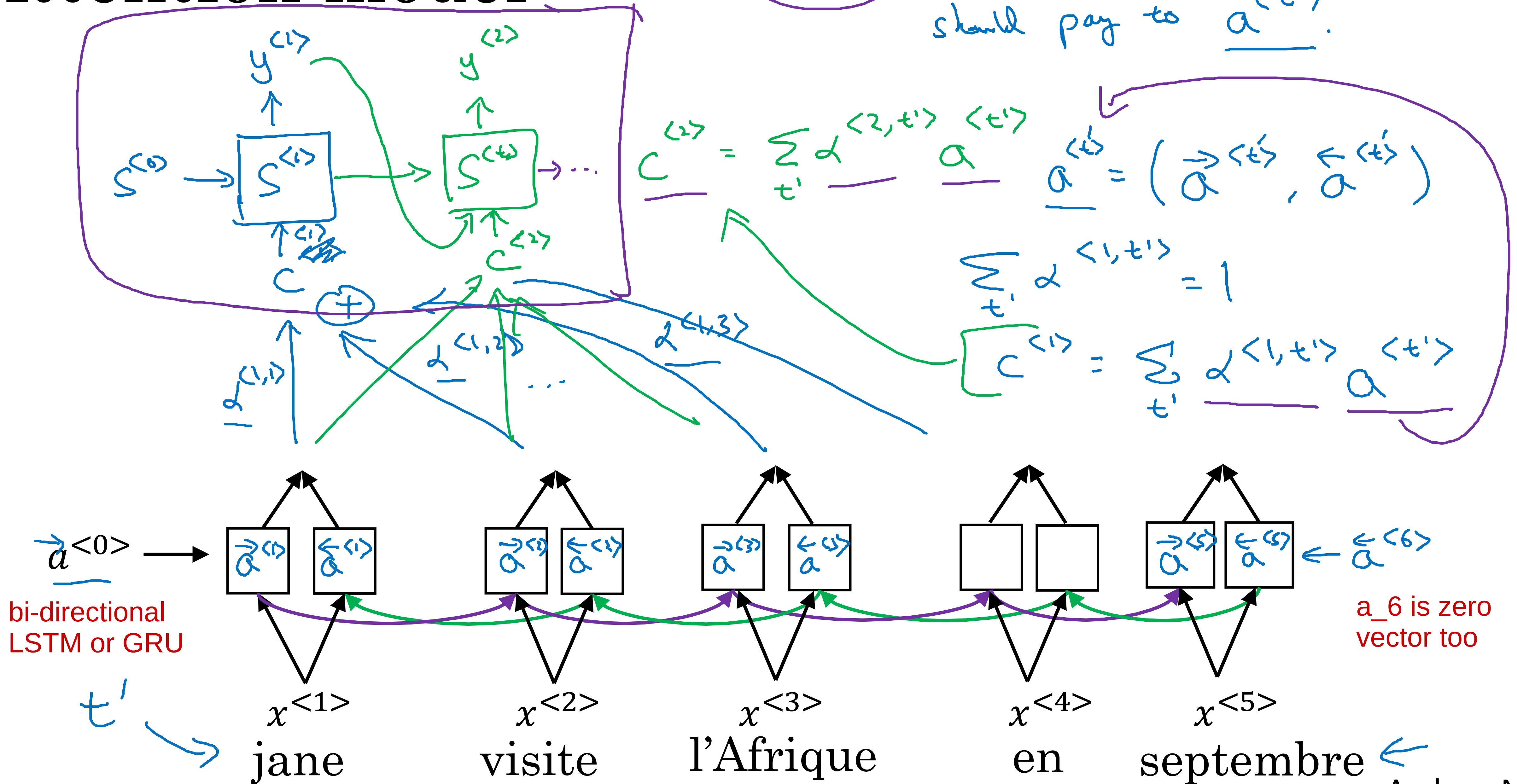
Sequence to sequence models

Attention model

deeplearning.ai

Attention model

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



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

Andrew Ng

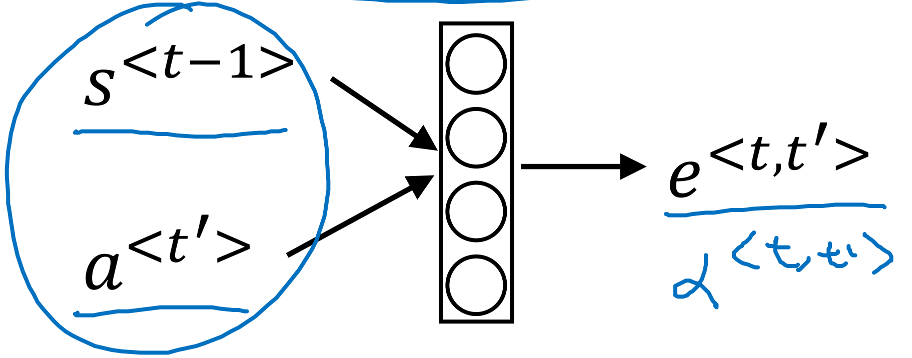
Computing attention $\alpha^{<t,t'>}$

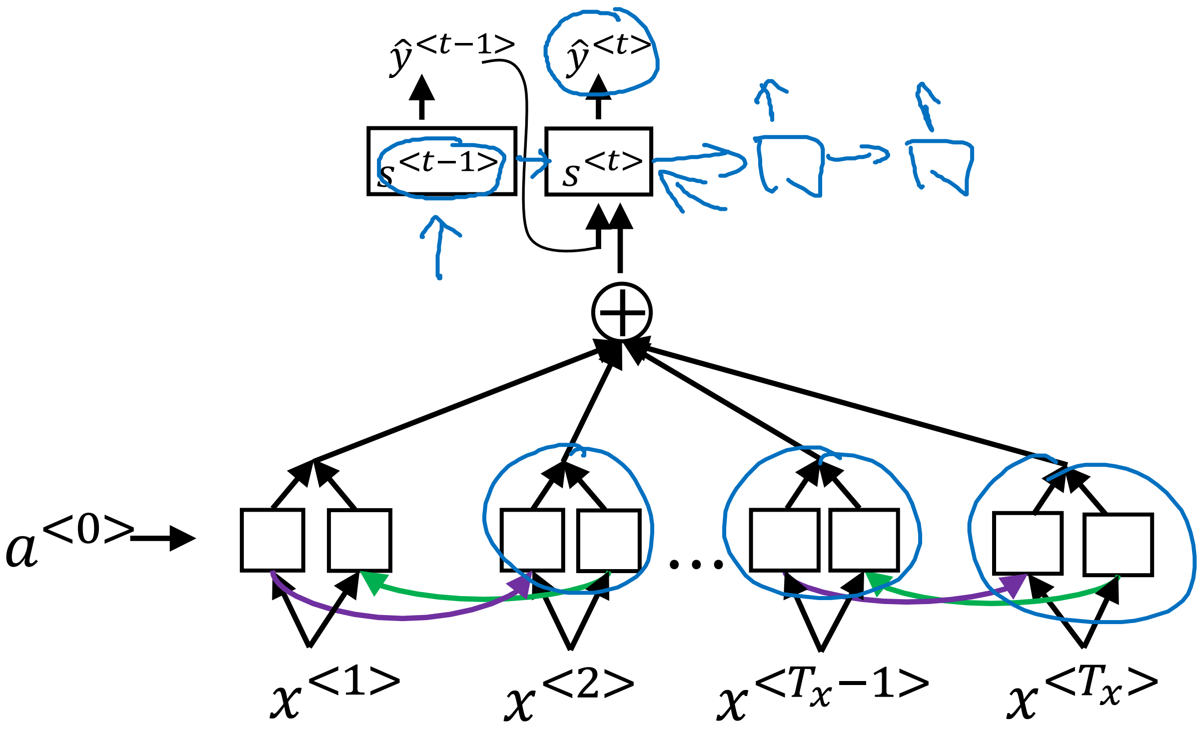
T_x T_y

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

softmax config so all weights add to 1

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





how to determine function 'e'
- we train a very small / 1-layer network for this
- train every (Tx, Ty) pair - quadratic computation time

[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]

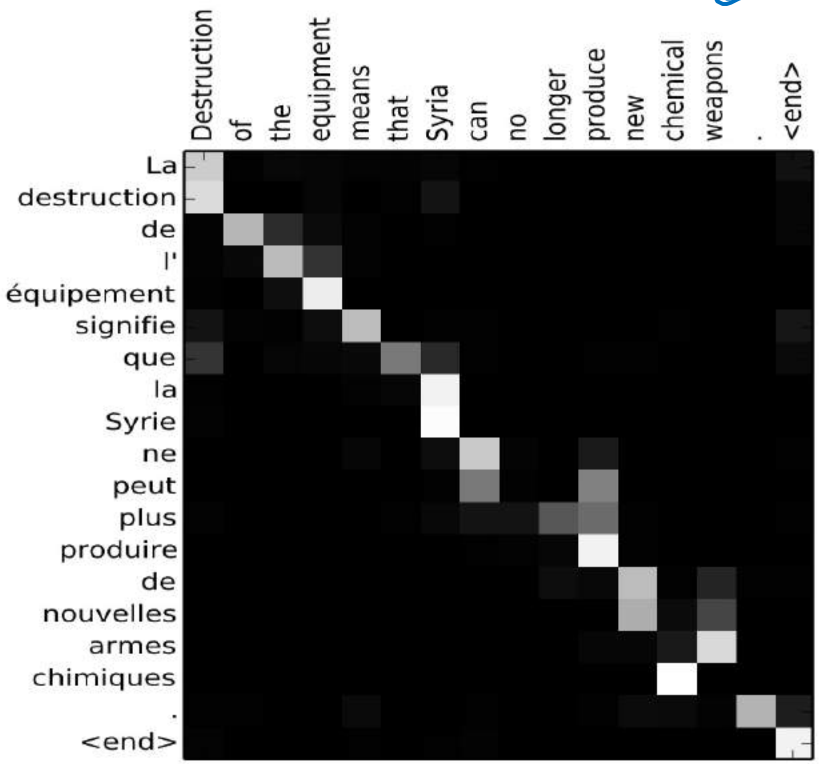
Andrew Ng

Attention examples

July 20th 1969 $\longrightarrow$ 1969 – 07 – 20

23 April, 1564 $\longrightarrow$ 1564 – 04 – 23

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



deeplearning.ai

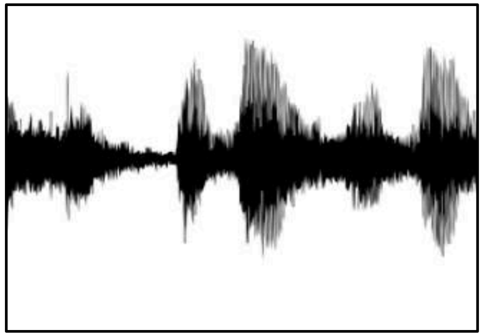
Audio data

Speech recognition

Speech recognition problem

x → y

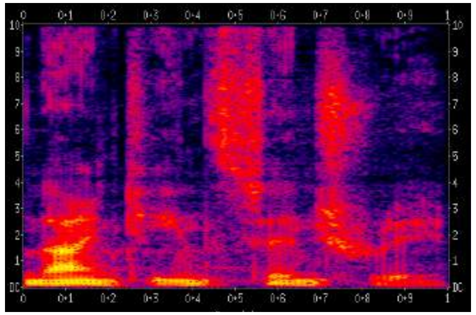
audio clip → transcript



→ "the quick brown fox"

Spectrogram - first preprocessing step

freq



time

color = how loud

previously,
modeled as
phonemes -
hand engineered

→ phonemes: de kwik braun

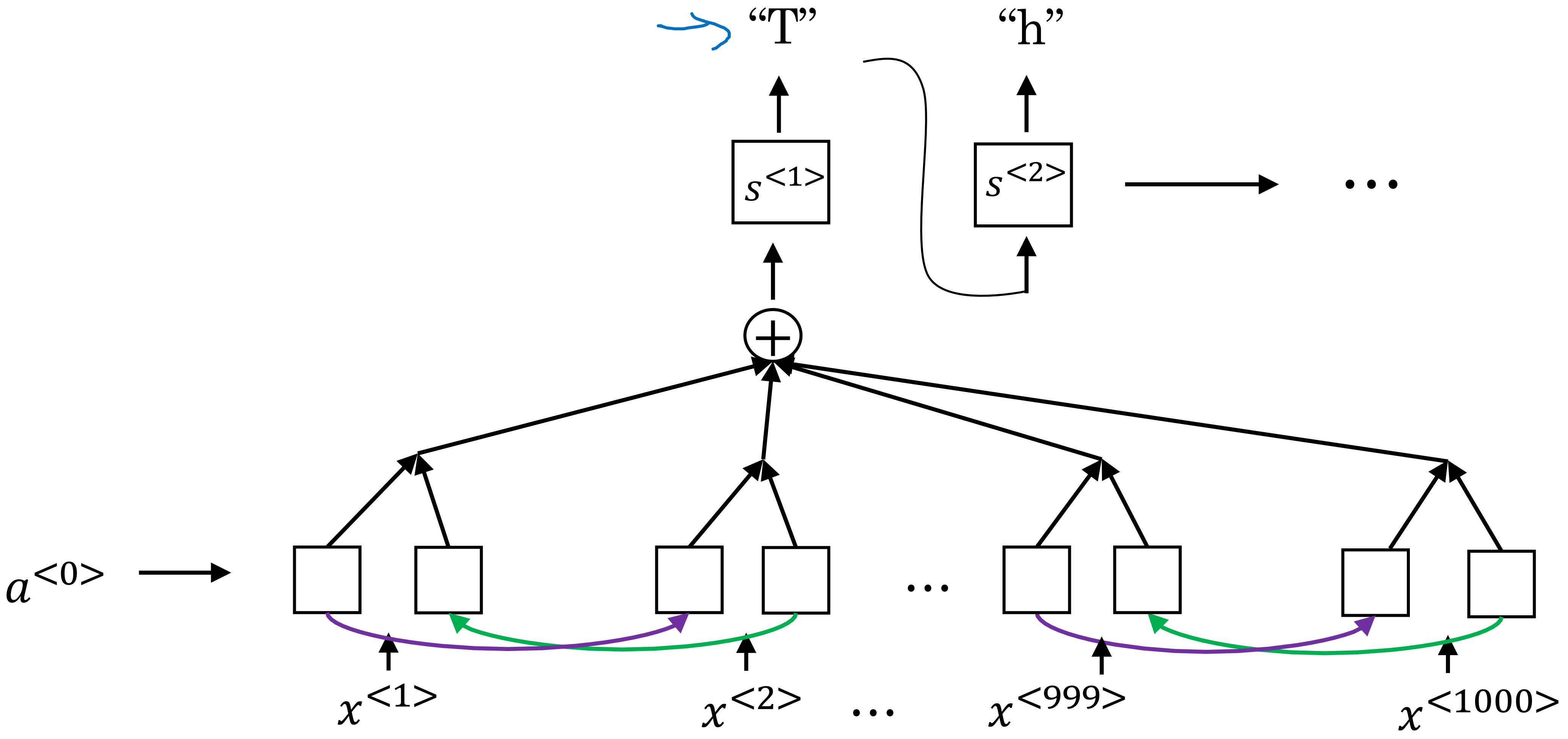
with end-to-end
deep learning
we can do it
directly w/o
hand
engineering

300h

3000h

100,000h

Attention model for speech recognition

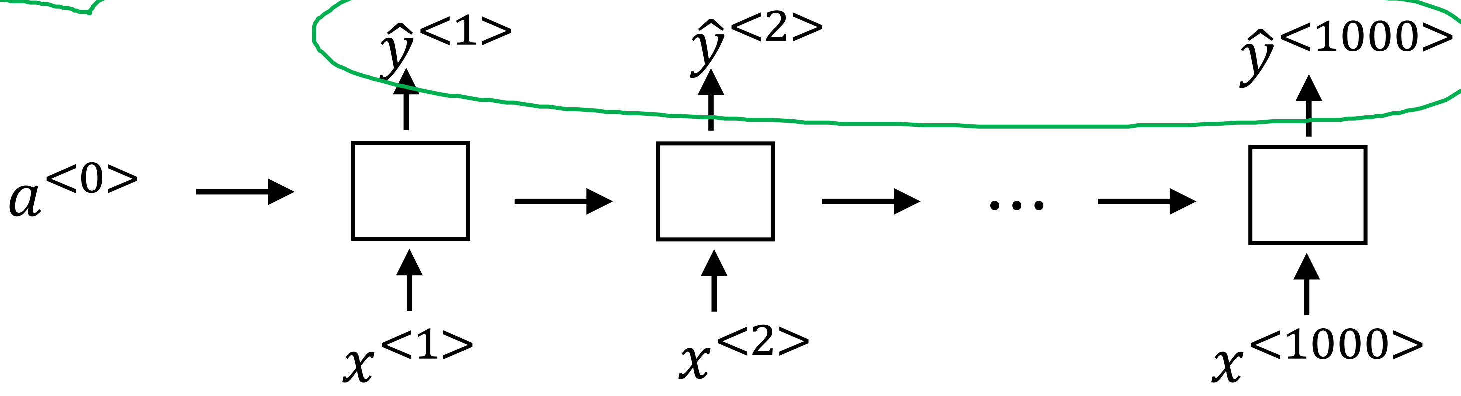


CTC cost for speech recognition

(Connectionist temporal classification)

19 characters

“the quick brown fox”



in practice it would be
bi-directional + deeper

10 second clip at 100hz
= 1,000 words

ttt_h_eee___␣___qqq__

the␣q

o/p should be much
smaller

“blank”

this is considered a correct output
from cost function point-of-view

“space”

achieved by applying
CTC rule to 1000 words

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

[Graves et al., 2006. Connectionist Temporal Classification: Labeling unsegmented sequence data with recurrent neural networks]

Andrew Ng

deeplearning.ai

Audio data

Trigger word detection

What is trigger word detection?

research on trigger word detection is still evolving and no consensus on best approach

Amazon Echo
(Alexa)

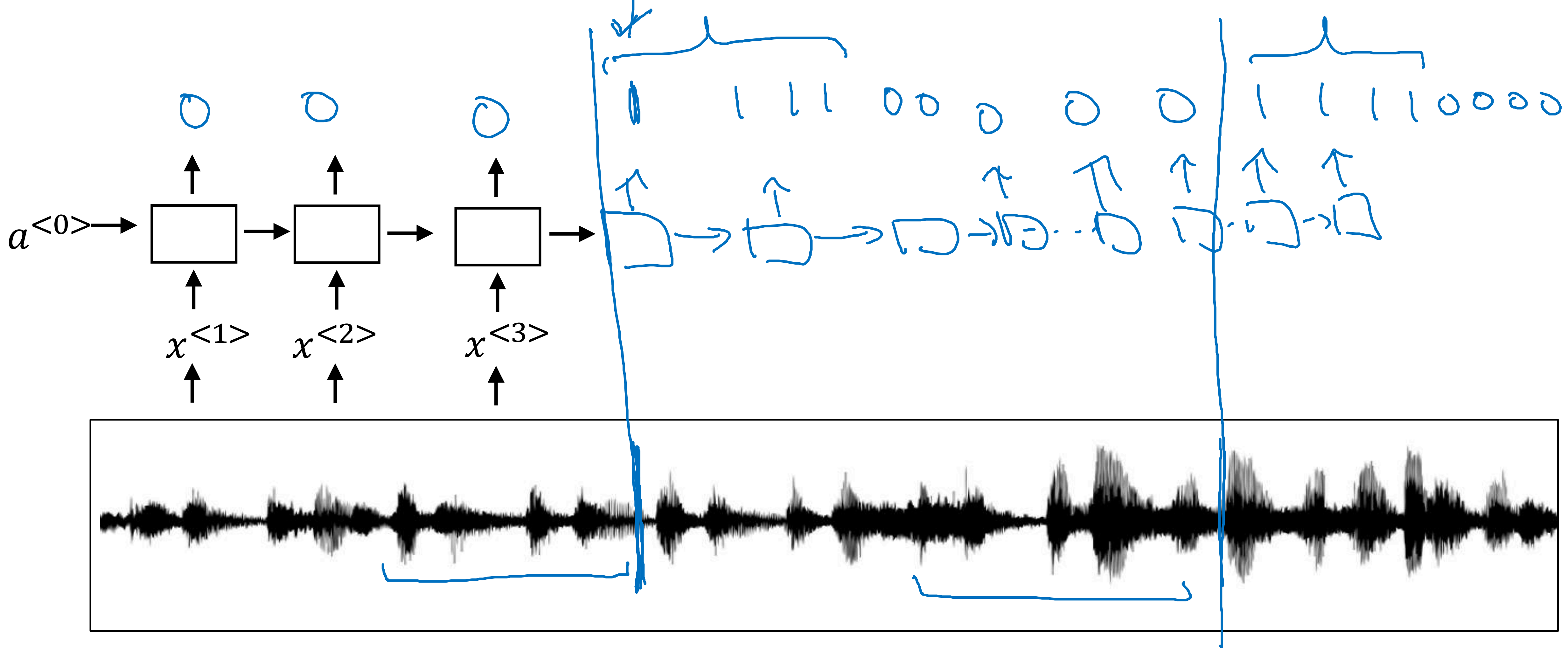
Baidu DuerOS
(xiaodunihao)

Apple Siri
(Hey Siri)

Google Home
(Okay Google)

Trigger word detection algorithm

simple approach:
use a RNN as before. training set: has 1 - right after the user finished saying the trigger word
creates an imbalanced dataset - with very few 1s. could repeat 1 for a few more following bits - to improve the imbalance



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.

Thank you.

-Andrew Ng

Copyright Notice

deeplearning.ai

Sequence to sequence models

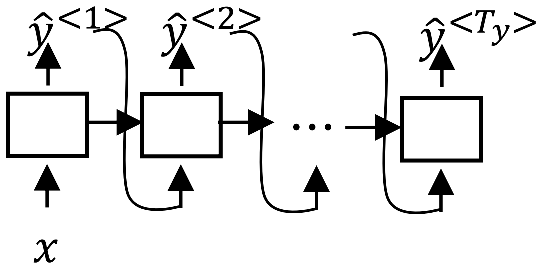
Transformers Intuition

Transformers Motivation

all these models are still 'sequential' because all units had to be read in 1-by-1

Increased complexity, sequential

RNN

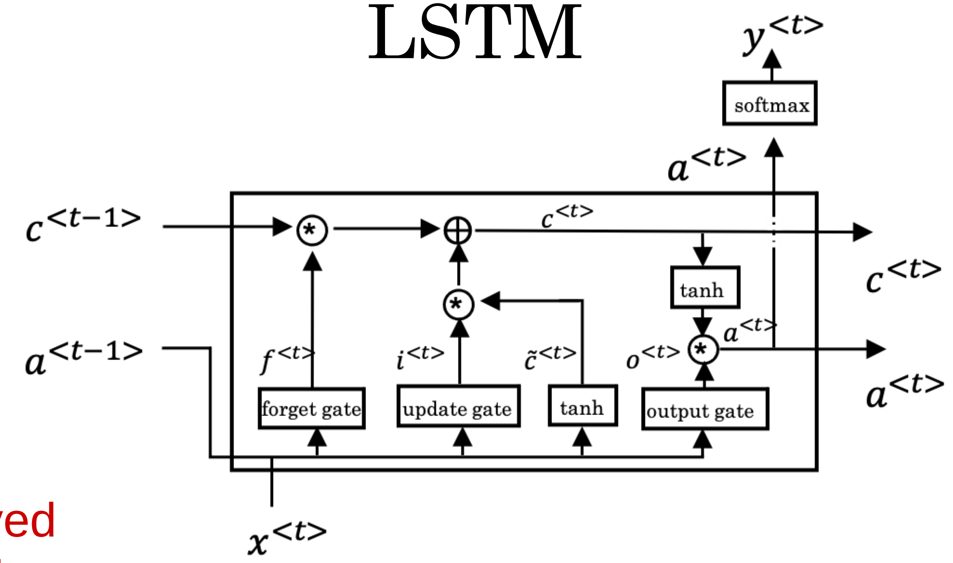


RNN had issues with training for long term dependencies due to vanishing gradients

GRU

GRU / LSTM solved this by adding gates and memory cell

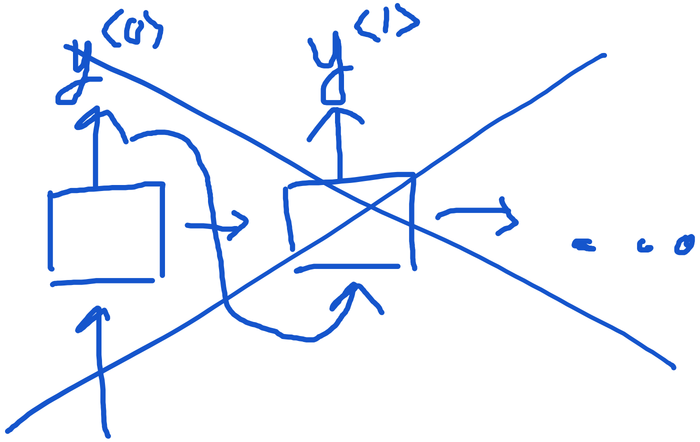
LSTM

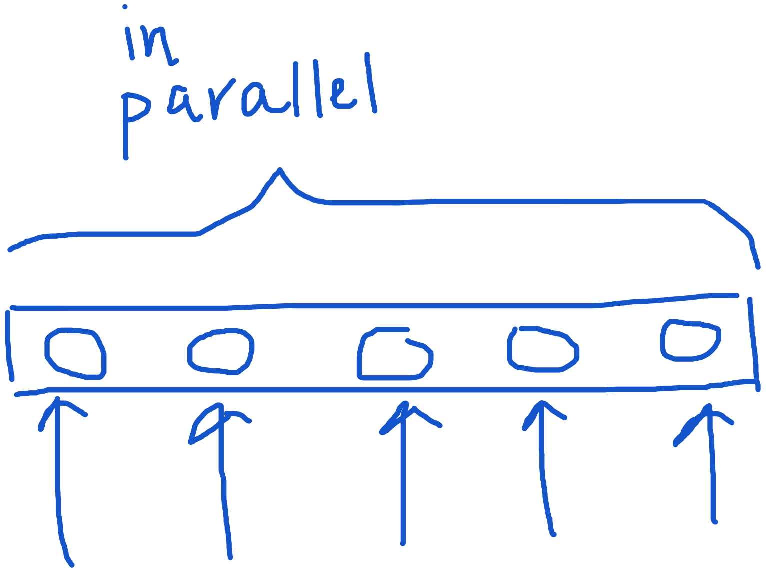


Transformers Intuition

- Attention + CNN
  - Self-Attention if our sentence has 5 words - we compute 5 representations: A<1> to A<5>
  - Multi-Head Attention ~ FOR Loop over self-attention

sequential way of processing tokens





CNN style of parallel processing

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

Q-query, K-key, V-value

RNN Attention

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

Transformers Attention

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

say we are calculating A<3> for l'Afrique
- previously we would just use a straight word embedding
- now we'll also include context - are we thinking of Africa as a holiday destination, or a place of historical significance, or second largest continent? The model will look at words around it, and use those to come up with a vector

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

[Vaswani et al. 2017, Attention Is All You Need]

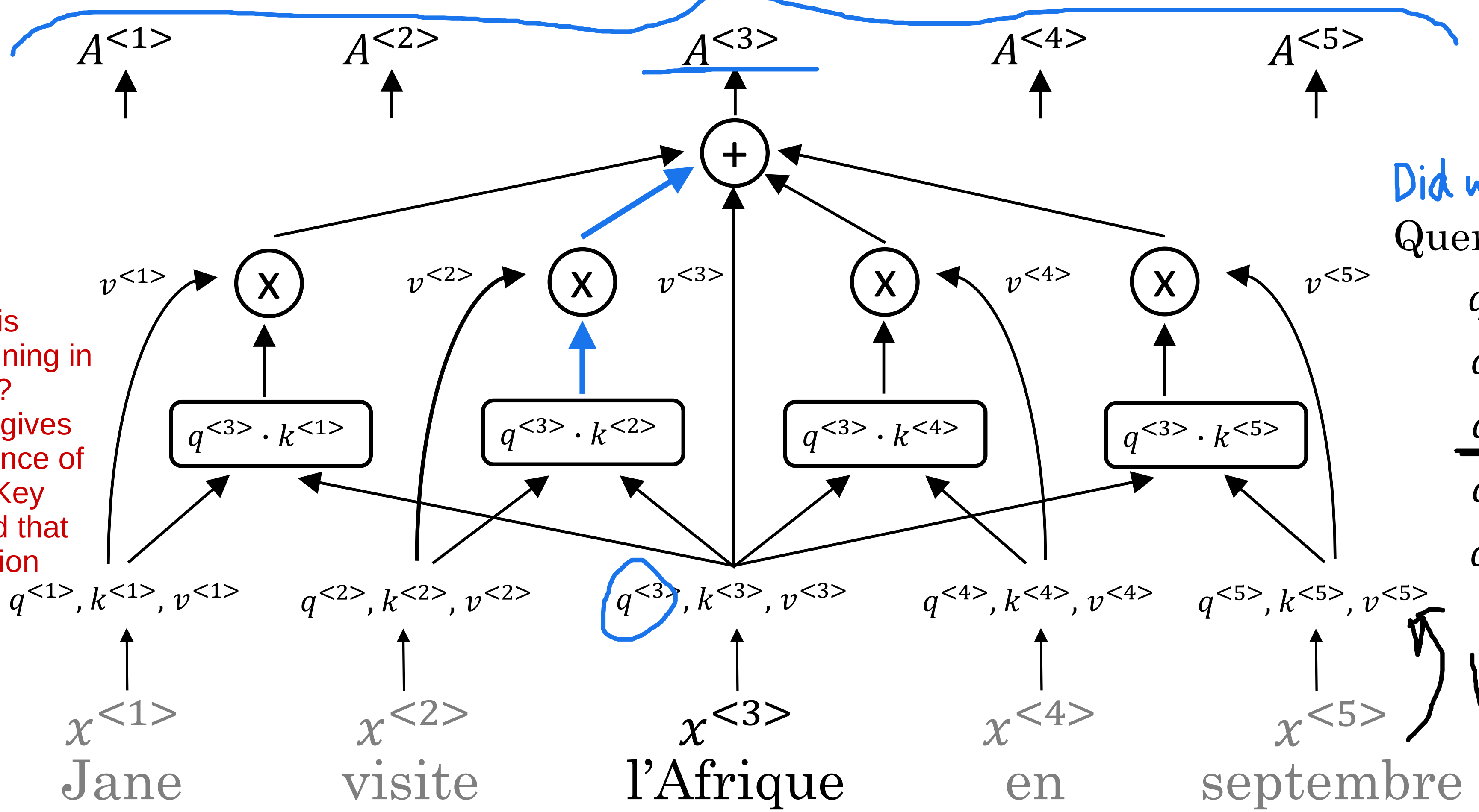
Self-Attention

softmax

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

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

vectorized representation
denominator is a scaling variable



What is happening in Africa?
Q*K - gives relevance of each Key toward that Question

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

Jane
visit

W^Q, W^K, W^V <- parameter matrices

q<3> = W_q * x<3>
k<3> = W_k * x<3>
v<3> = W_v * x<3>

[Vaswani et al. 2017, Attention Is All You Need]

Andrew Ng

deeplearning.ai

Sequence to sequence models

Multi-Head Attention

Multi-Head Attention

Each self-attention for a sequence is called 'head'
Essentially a big FOR-LOOP over the self-attention mechanism

another W0 here!

$$MultiHead(Q,K,V) = concat(head_1 head_2 \dots head_n) W_o$$

$$head_i = Attention\,(W_i^Q Q, W_i^K K, W_i^V V)$$

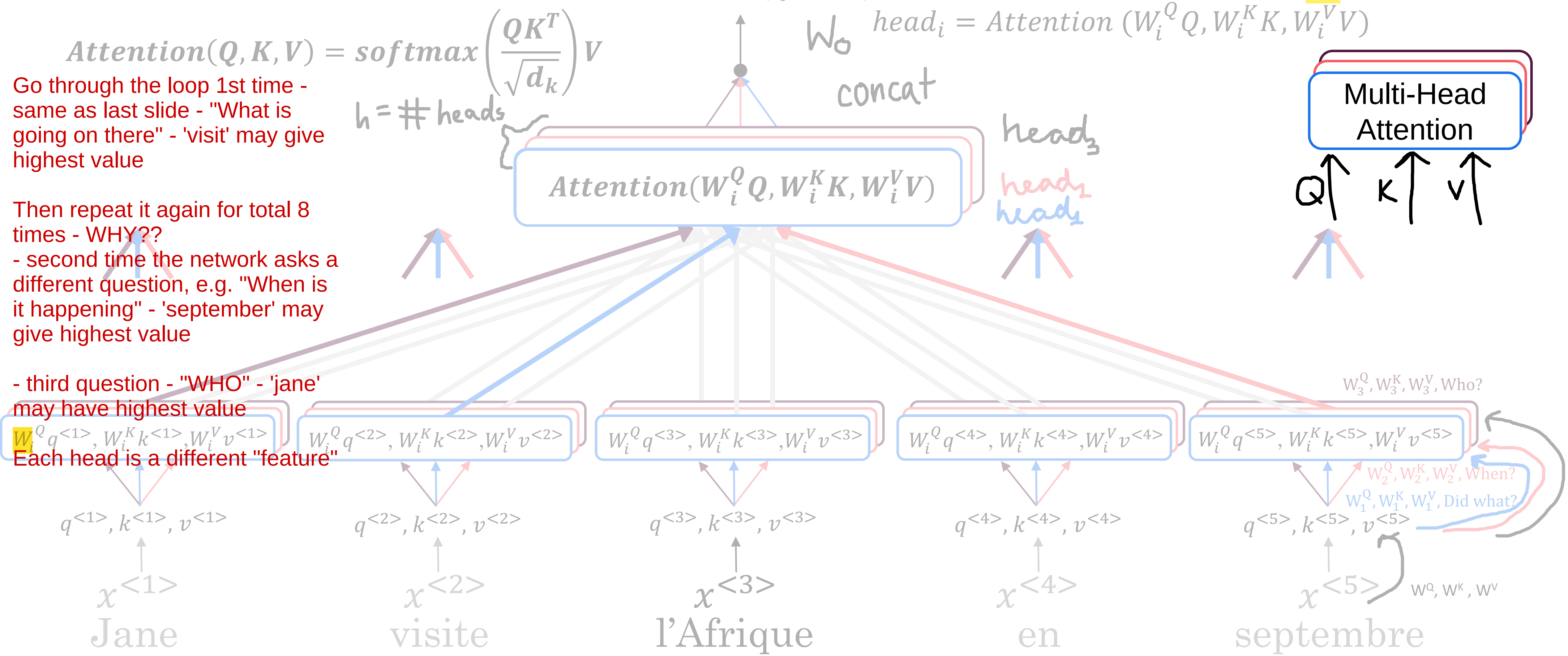
$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Go through the loop 1st time - same as last slide - "What is going on there" - 'visit' may give highest value

Then repeat it again for total 8 times - WHY??
- second time the network asks a different question, e.g. "When is it happening" - 'september' may give highest value

- third question - "WHO" - 'jane' may have highest value

Each head is a different "feature"



[Vaswani et al. 2017, Attention Is All You Need]

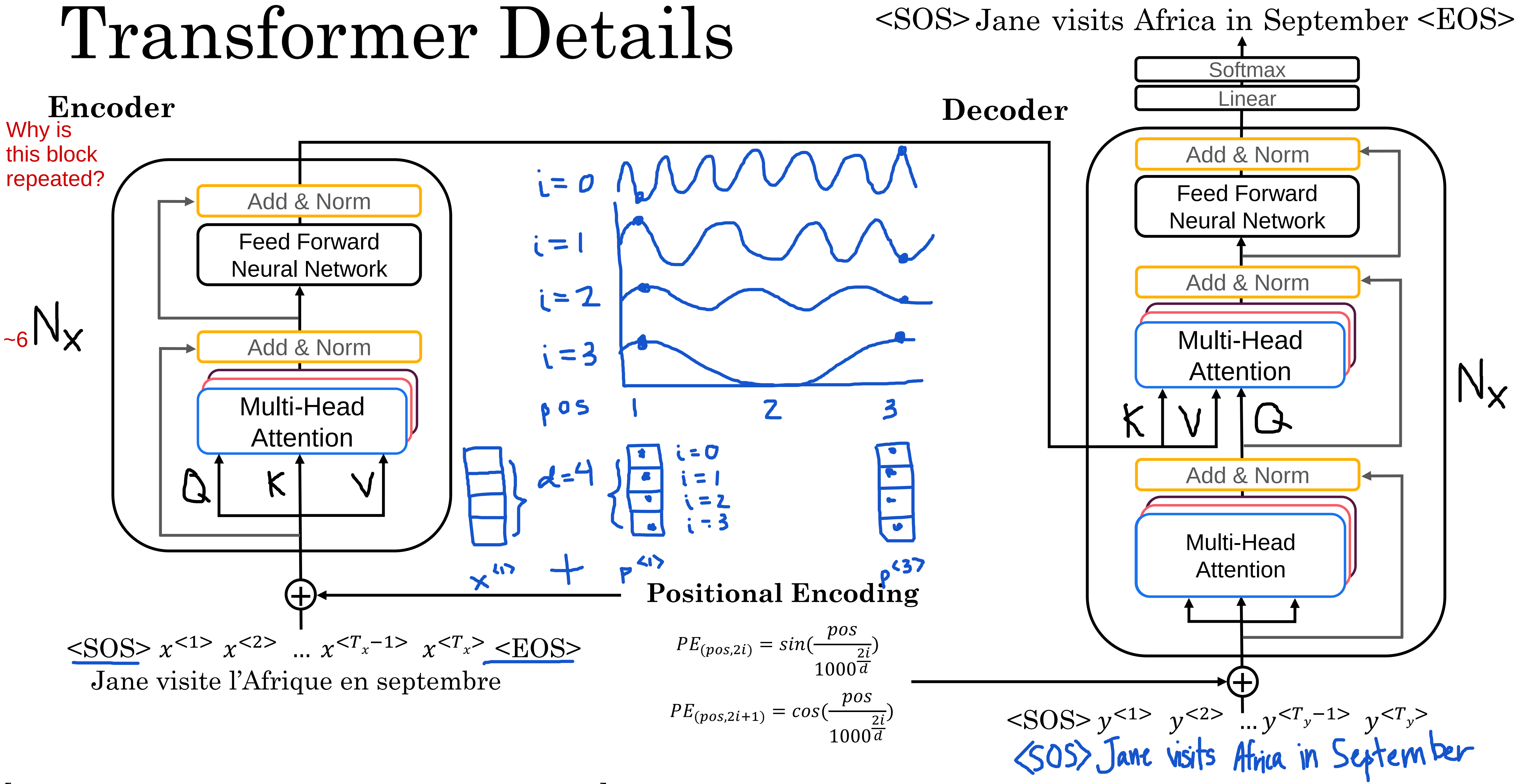
Andrew Ng

Sequence to sequence models

Transformers

deeplearning.ai

Transformer Details



Positional Encoding

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

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

[Vaswani et al. 2017, Attention Is All You Need]

Andrew Ng