deeplearning.ai

Sequence to sequence models

Basic models

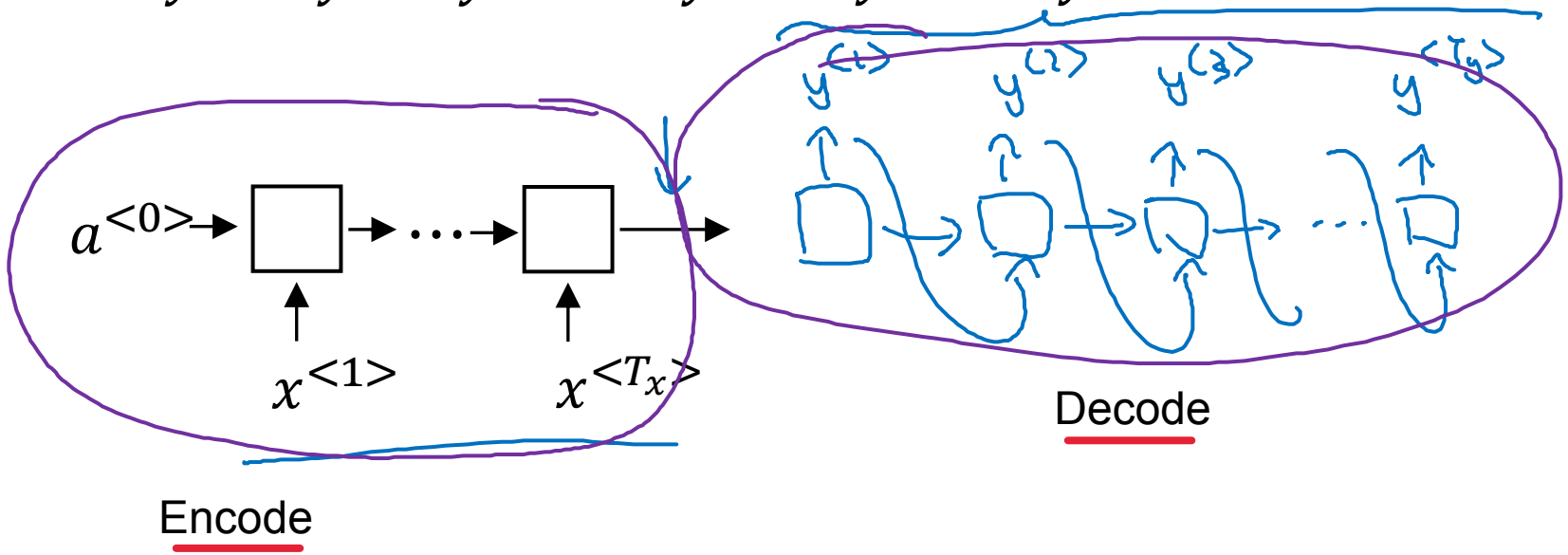
Sequence to sequence model

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

Jane visite l'Afrique en septembre

⟶ Jane is visiting Africa in September.

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



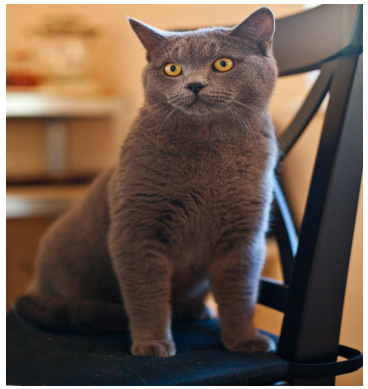
[Sutskever et al., 2014. Sequence to sequence learning with neural networks]

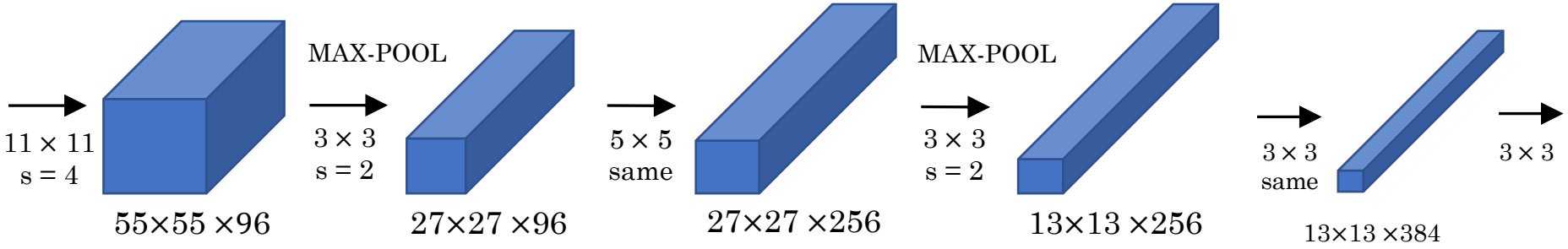
[Cho et al., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation]

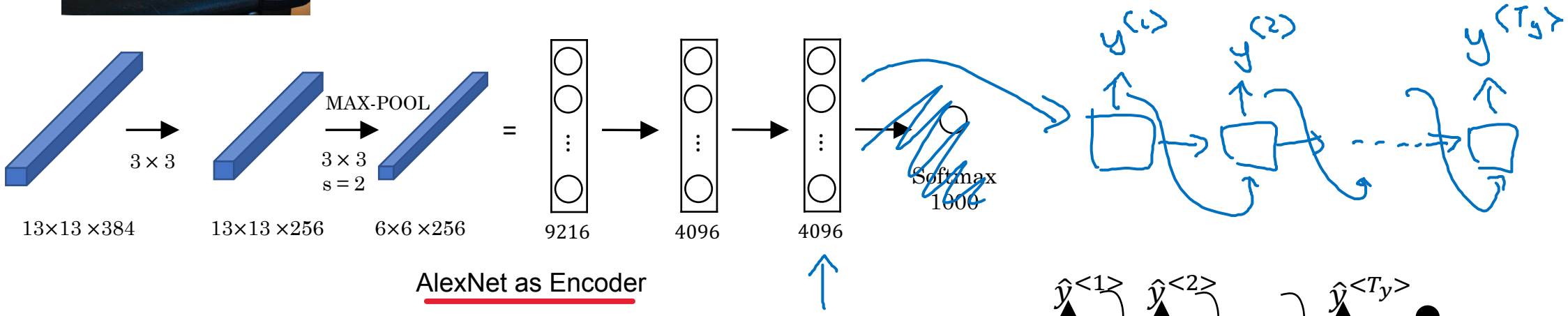
Image captioning

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

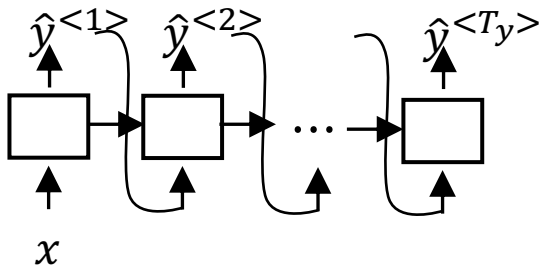
A cat sitting on a chair







AlexNet as Encoder



- such approach works well with similar short sentences

[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]

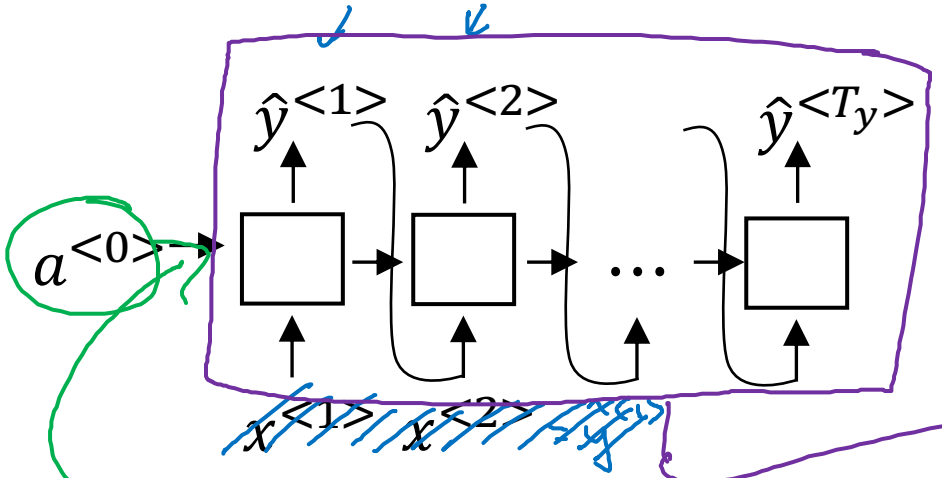
deeplearning.ai

Sequence to sequence models

Picking the most likely sentence

Machine translation as building a conditional language model

Language model:

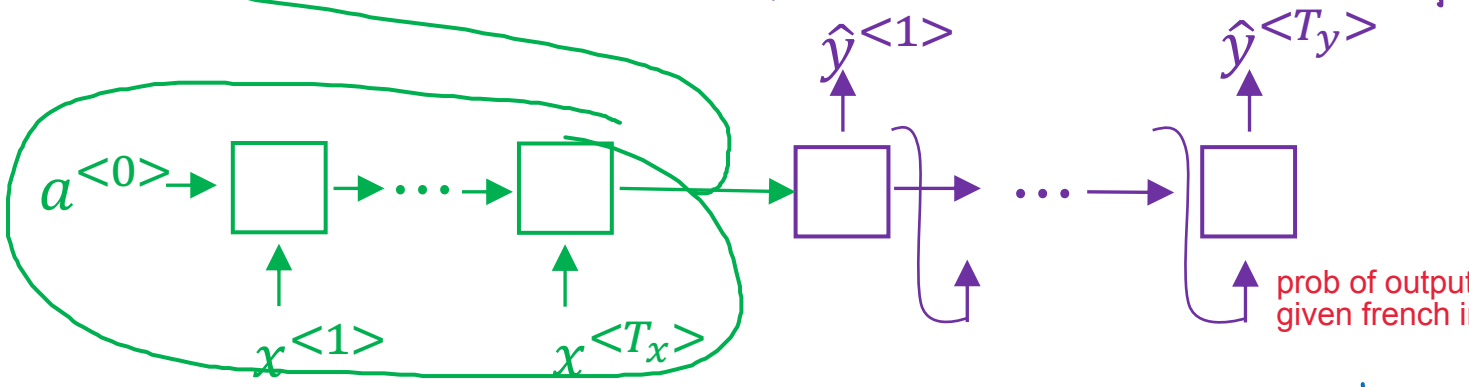


prob of any sentence

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

Machine translation:

encoded network figures out some representations of input sentence



prob of output english translation given french input sentence

"Conditional language model"

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

Finding the most likely translation

Jane visite l'Afrique en septembre.

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

English

French

previously, we generated sentences at random
we want to avoid such random sampling here
we shall focus on the most likely sentence instead

→ Jane is visiting Africa in September.

→ Jane is going to be visiting Africa in September.

→ In September, Jane will visit Africa.

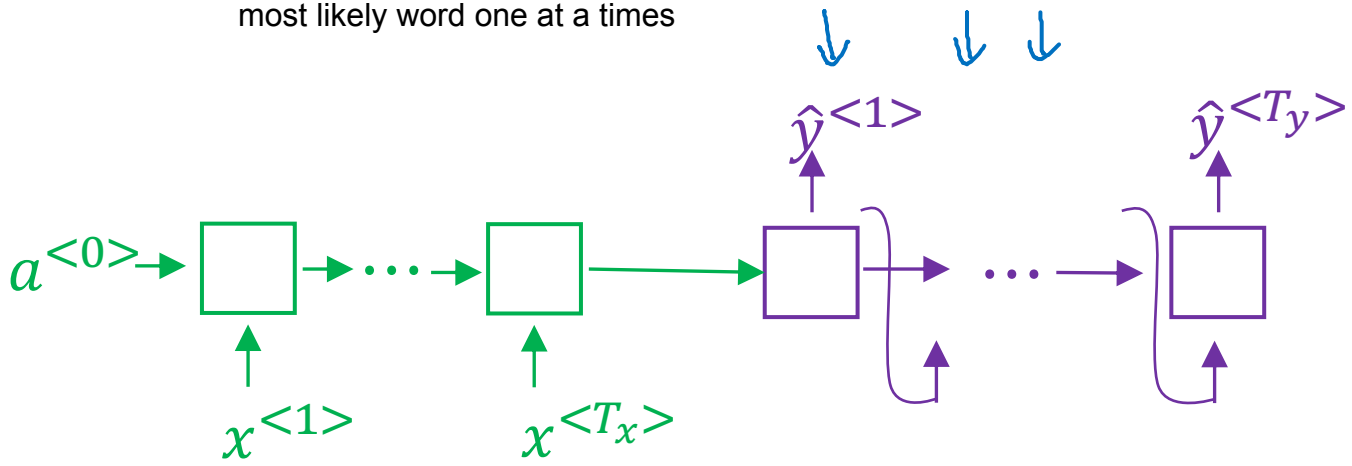
→ Her African friend welcomed Jane in September.

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

Andrew Ng

Why not a greedy search?

most likely word one at a times



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

we try to maximize
most probable sequence
of words i.e sentences

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

words in vocab
$10,000$
words long
10
$10,000^{10}$
total possible sentences that are 10-word long

$P(y|x)$

→ Jane is visiting Africa in September.

→ Jane is going to be visiting Africa in September.

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

Andrew Ng

deeplearning.ai

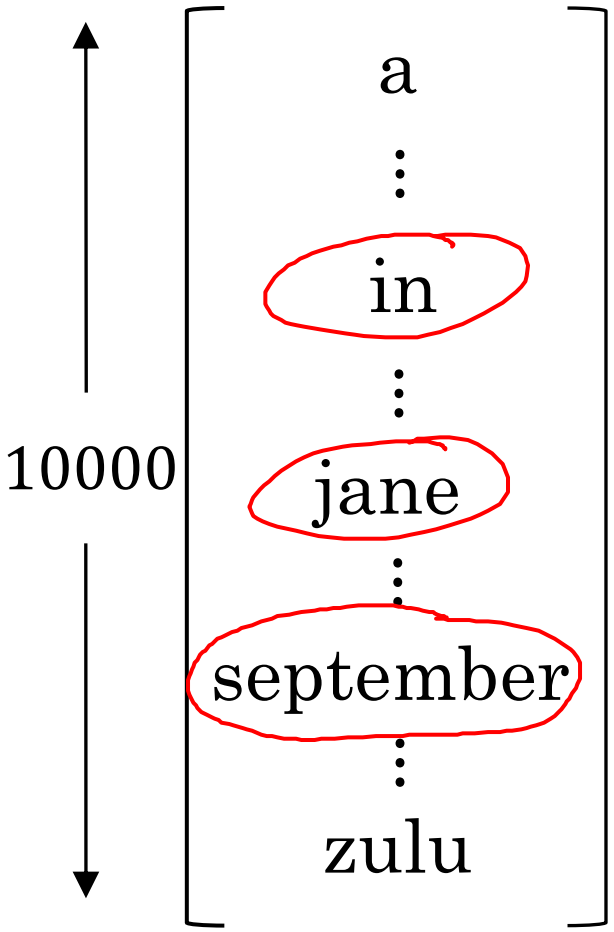
Sequence to sequence models

Beam search

Beam search algorithm

$B = 3$ (beam width)

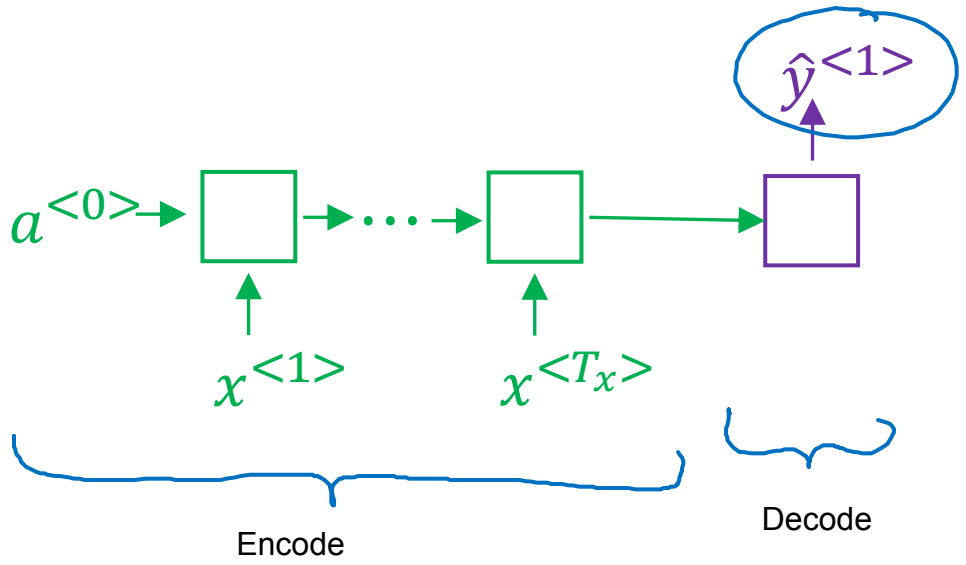
Step 1



list of all possiible words in vocab

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

softmax output all 10,000 possibilities

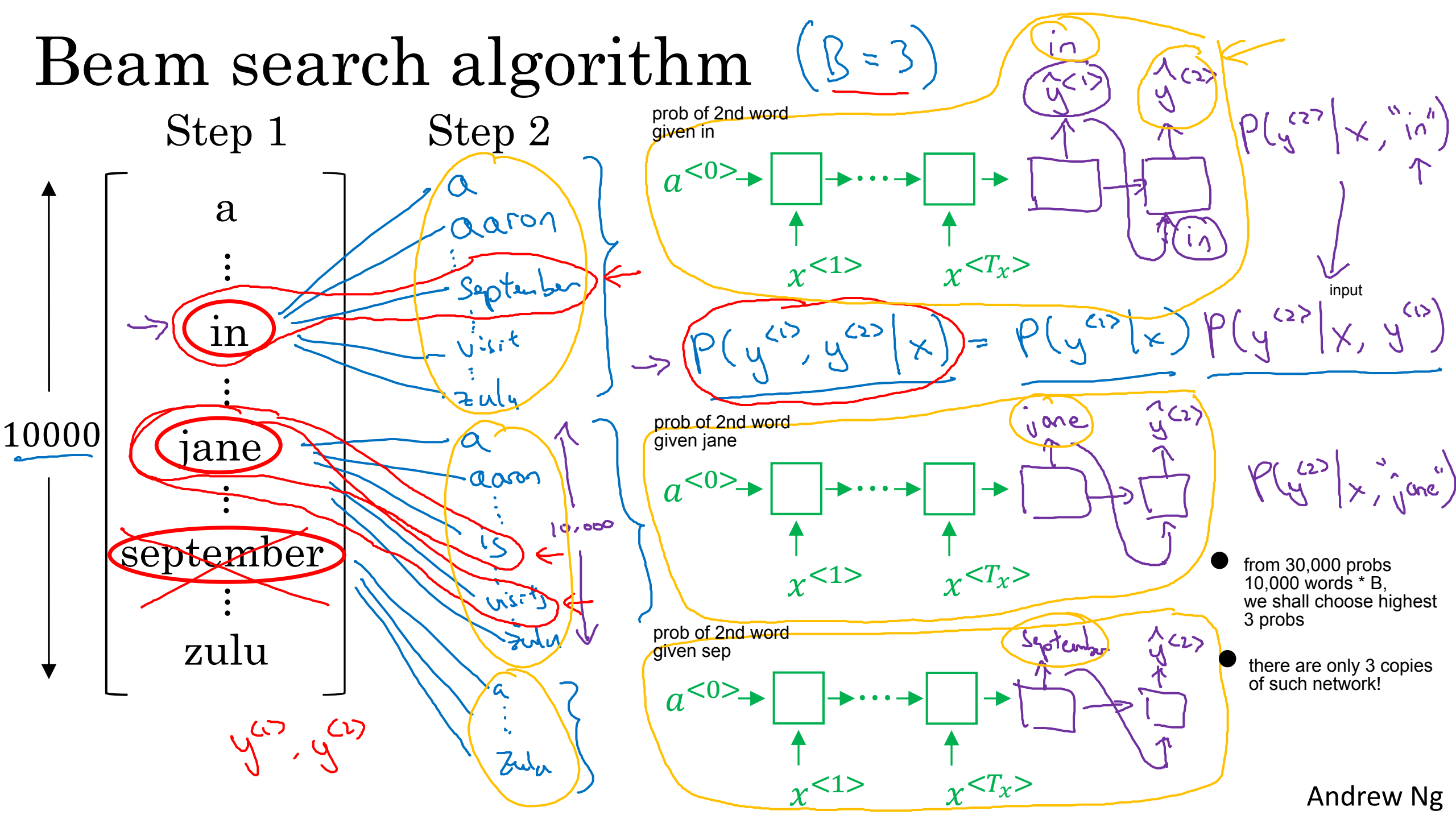


unlike Greedy which choose the most likely word and move on to the next.

Beam Search shall consider # beam_width words at a time, keeping track of them.

Andrew Ng

Beam search algorithm

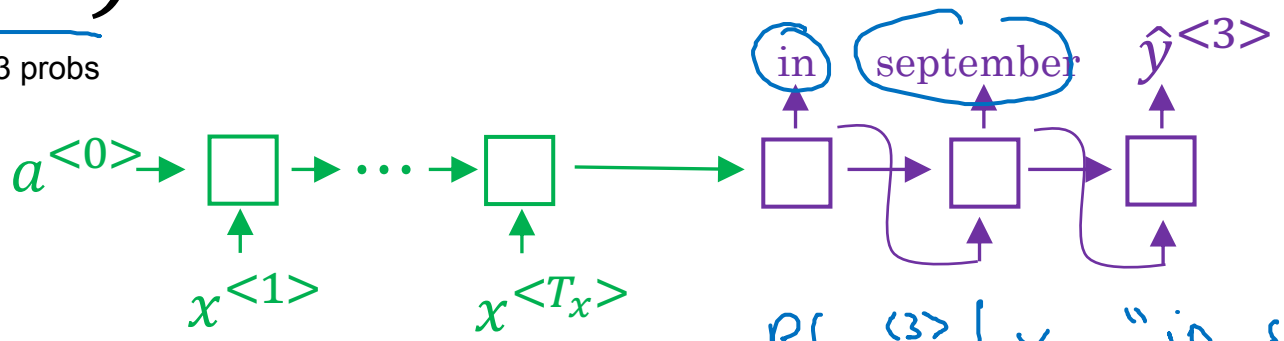


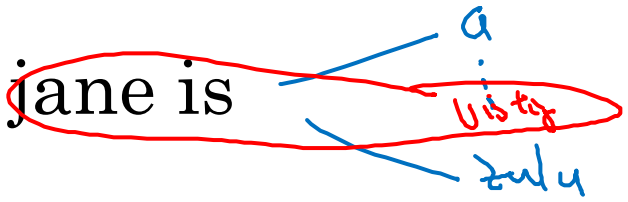
Beam search ($B = 3$)

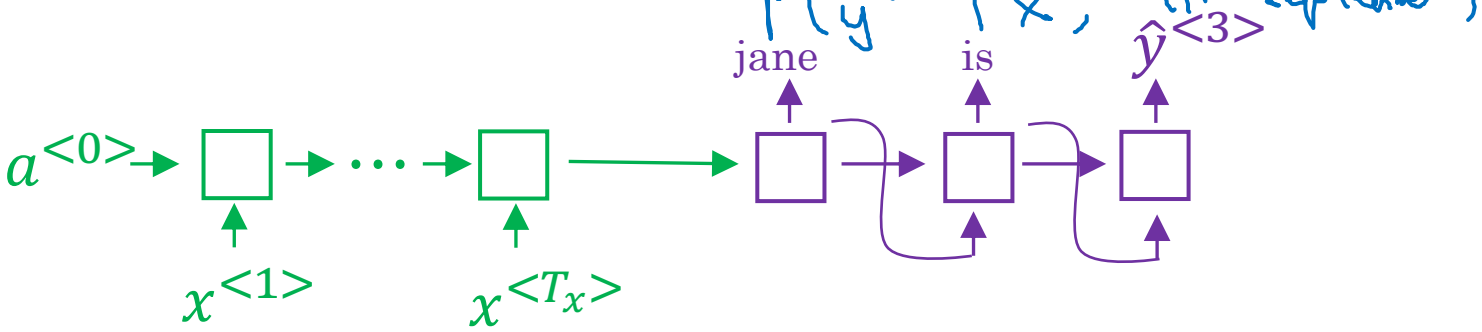
i.e top 3 probs

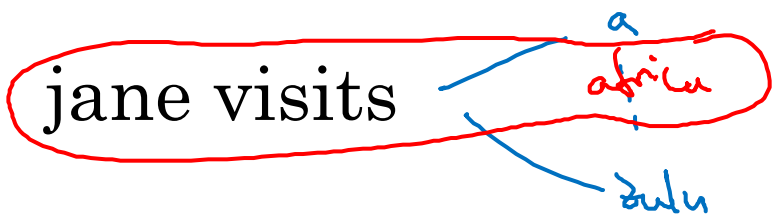
$B = 1$ ⇝ greedy search

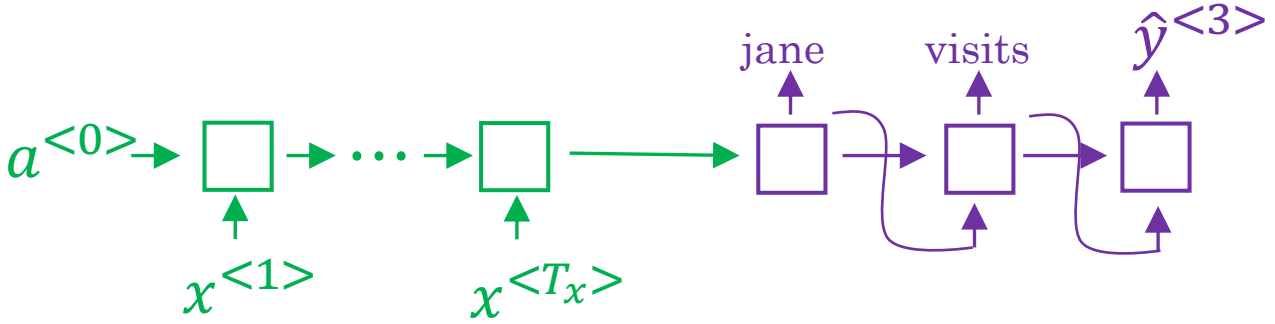
we keep it going!











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

jane visits africa in september. <EOS>

Andrew Ng

deeplearning.ai

Sequence to sequence models

Refinements to beam search

Length normalization

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

output would be too small

log

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

more numerically stable, less prone to round errors

length of sentence

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

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

Normalized Log-Likelihood Objective

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

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

$\log P(y|x)$ ←

same result as

$P(y|x)$ ←

- CON: such equation prefers shorter sentences. thus result becomes more -ve, when multiplying many probs.

  that's why we normalize it.

Andrew Ng

Beam search discussion

is a heuristic approach

yet computationally expensive

large B: better result, slower

Small B: worse result, faster

Beam width B?

1 -> 3 -> 10, 100, 1000, -> 3000

notice for diminishing returns with larger values of B

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

Analyzing which shall be tuned → RNN

→ Beam Search

BT

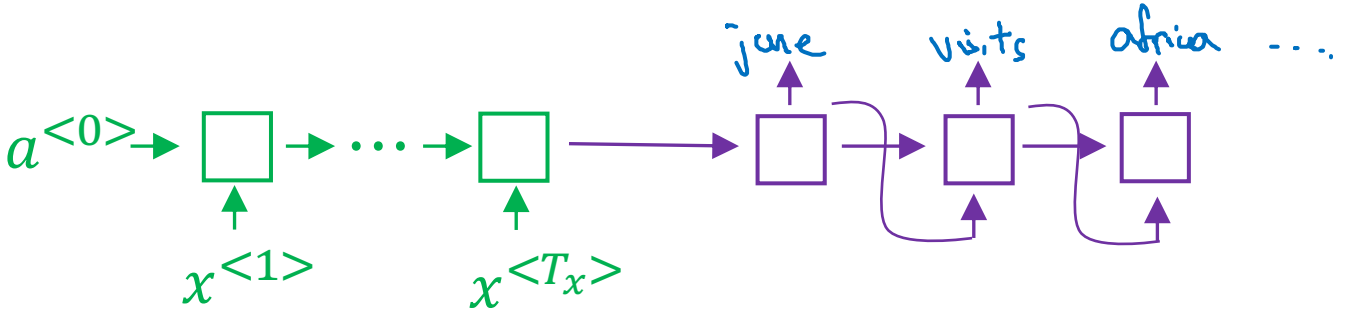
Jane visite l'Afrique en septembre.

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

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

Idea: deploying RNN with both probs

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



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. ($\hat{y}$)

Case 1: $P(y^*|x) > P(\hat{y}|x)$ ←

this is the main job of beam search $\arg\max_y P(y|x)$

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

Conclusion: Beam search is at fault.

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

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

Conclusion: RNN model is at fault.

Andrew Ng

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

we are trying to evaluate degrees by which MT output is similar to or overlaps with multiple referencess

Bleu
bilingual evaluation understudy

French: Le chat est sur le tapis.

Reference 1: The cat is on the mat.

Reference 2: There is a cat on the mat.

Machine Translation

MT output: the the the the the the the.

Precision: 7 / 7

each word in MT output appears at least in one of the refs

each word in MT output will be given credit only up to the max occurances in the ref sentences

Modified precision: 2 / 7

in ref 1, the appeared twice,
in ref 2, the appeared once

taking 2 over 1, we get 2 / 7

Here, we look at words in isolation 'unigrams'

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

Bleu score on bigrams

pairs of words appearing next to each other

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.

Bigrams	Count (in MT output)	$Count_{clip}$ (in refs)
the cat	2	1
cat the	1	0
cat on	1	1
on the	1	1
the mat	1	1

$$\frac{\text{\# count clips}}{\text{\# counts}} = \frac{4}{6}$$

[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.

$p_1, p_2, \ldots = 1.0$

MT output is exactly equal to one of the refs

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

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

as basic units of sound

[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

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

adjusting factor; penalizing too short outputs

Bleu is a single real num evaluatioin metric, used only in this scenario of evaluating a generated text to another ref one.
It shall no be used in speech recongition; since there's only one ground truth you're evaluating against.

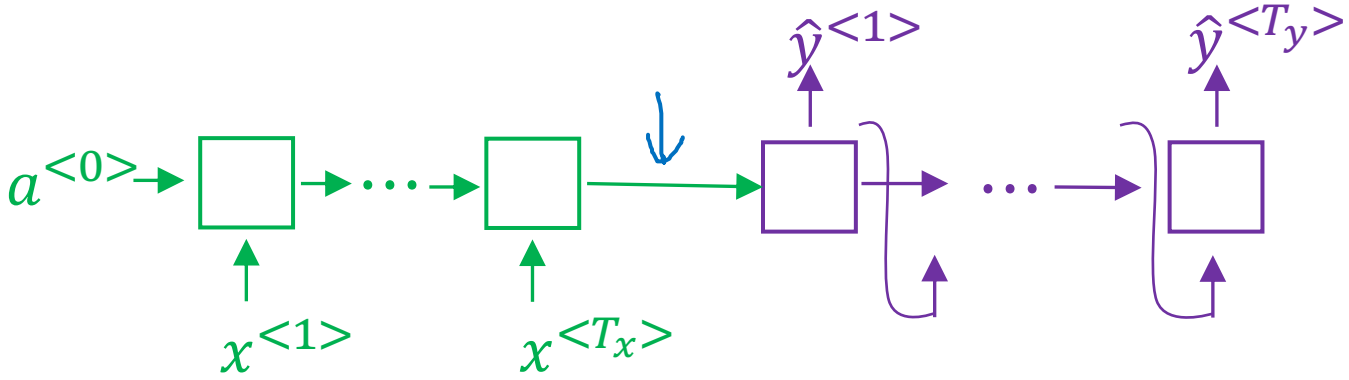
[Papineni et. al., 2002. Bleu: A method for automatic evaluation of machine translation]

deeplearning.ai

Sequence to sequence models

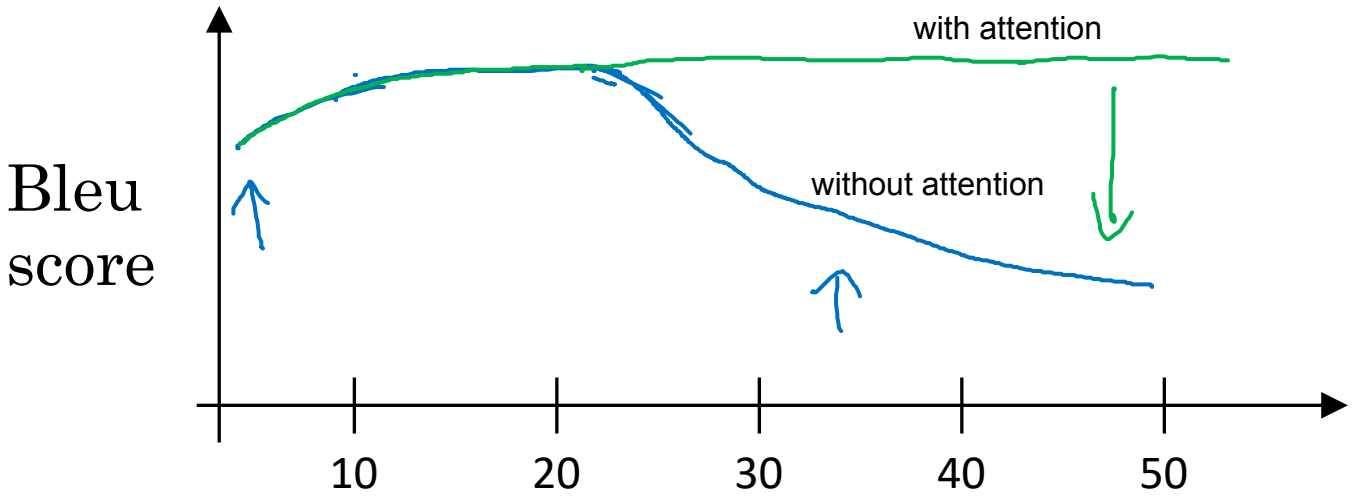
Attention model intuition

The problem of long sequences



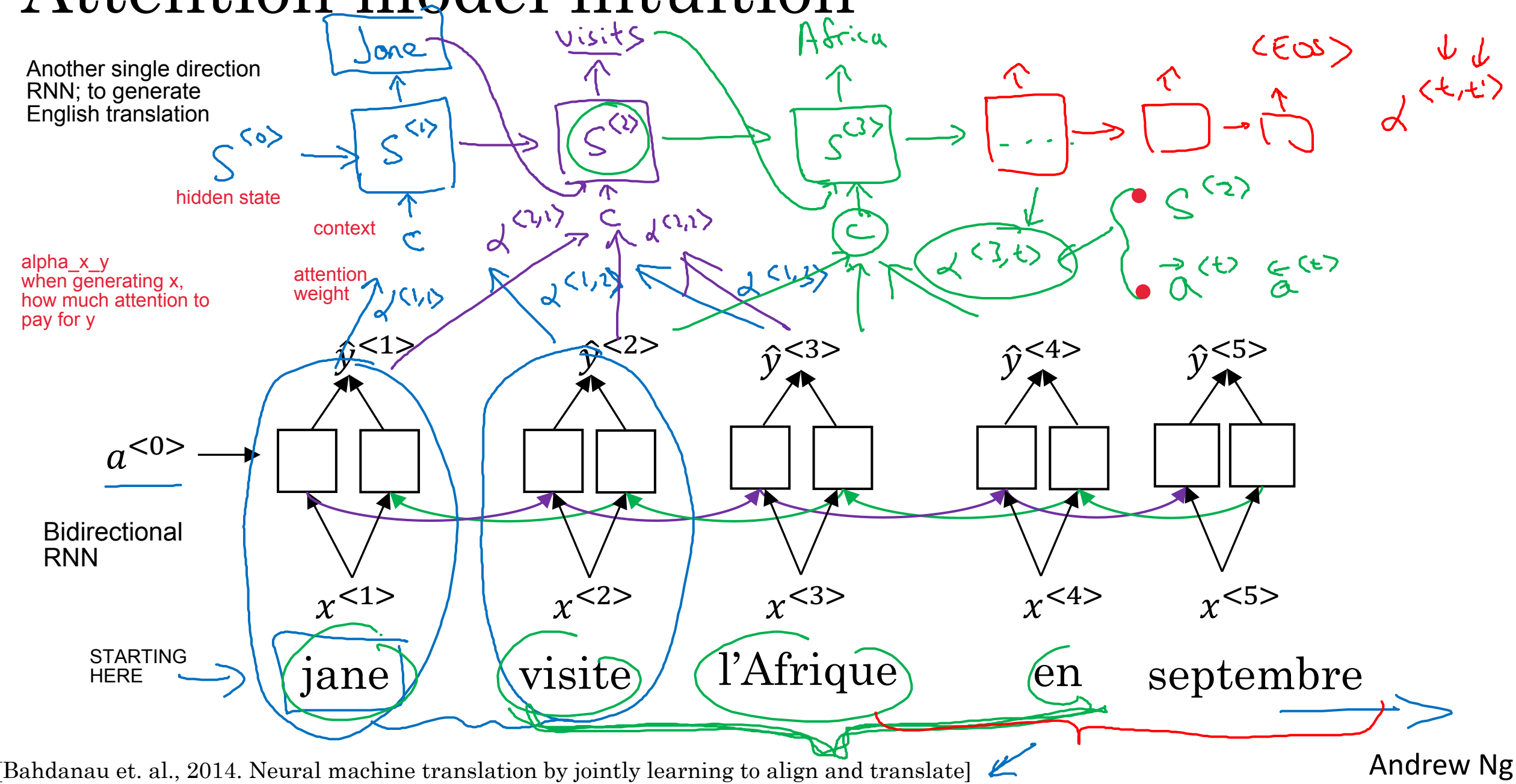
Jane s'est rendue en Afrique en septembre dernier, a apprécié la culture et a rencontré beaucoup de gens merveilleux; elle est revenue en parlant comment son voyage était merveilleux, et elle me tente d'y aller aussi.

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

using a set of attention weights



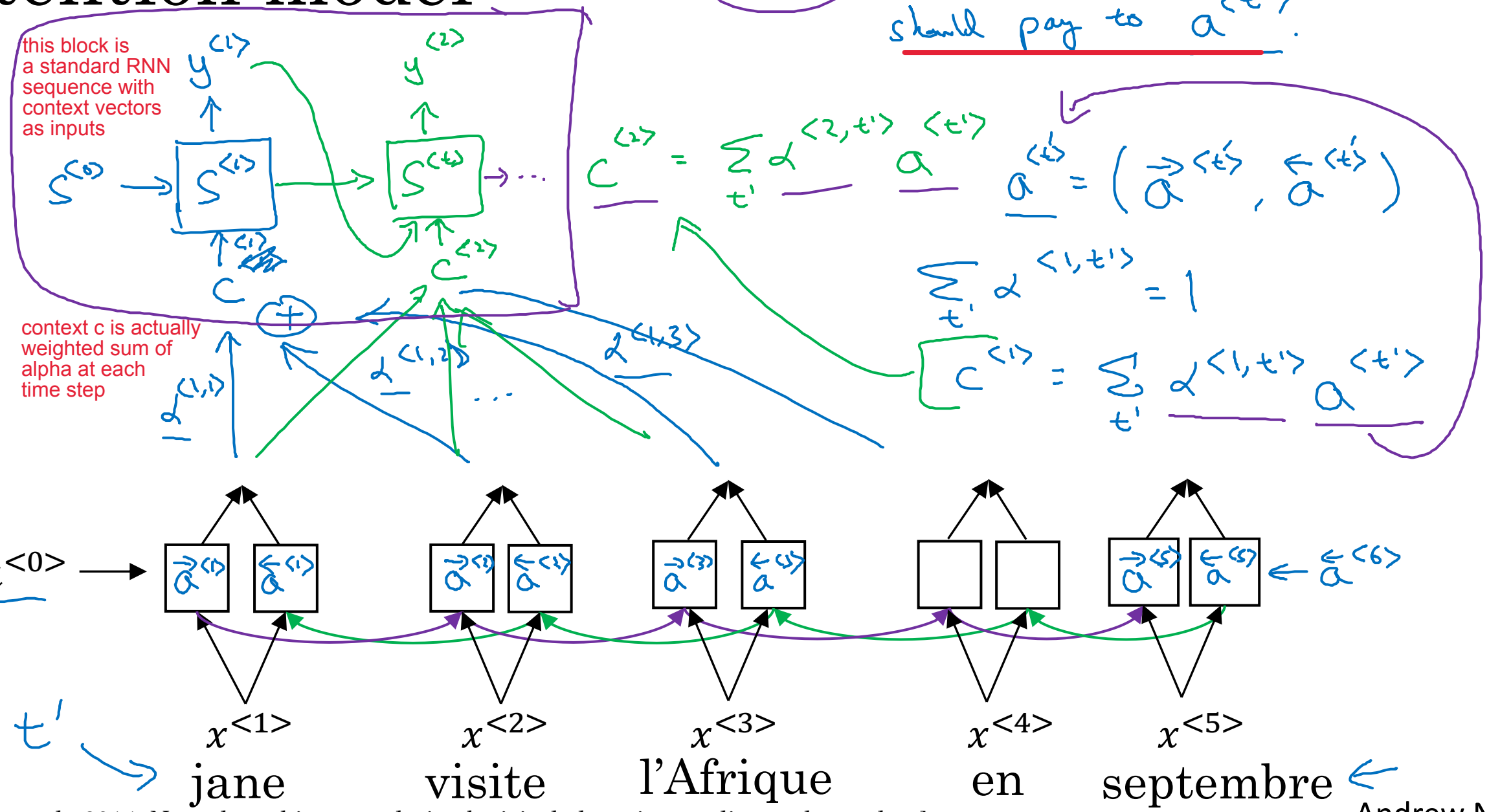
[Bahdanau et. al., 2014. Neural machine translation by jointly learning to align and translate]

deeplearning.ai

Sequence to sequence models

Attention model

Attention model



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

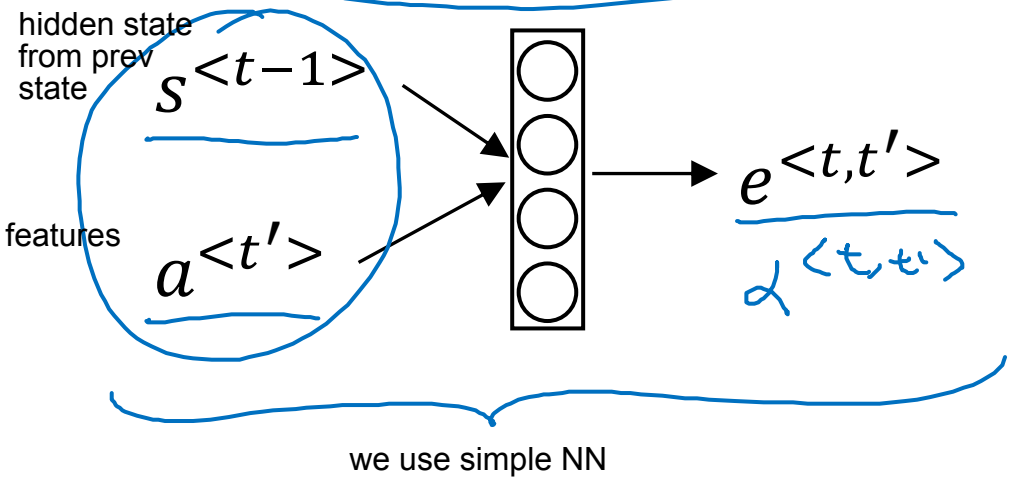
Andrew Ng

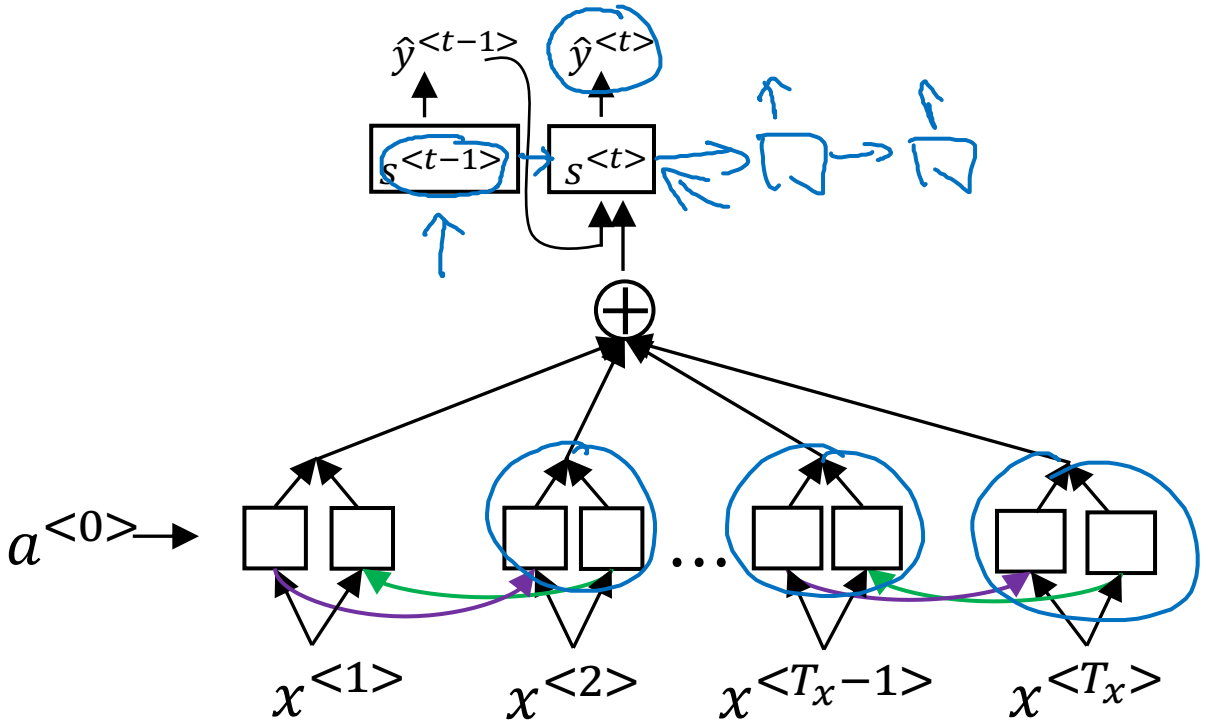
Computing attention $\alpha^{<t,t'>}$

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

CON: the cost is quadratic

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





[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]

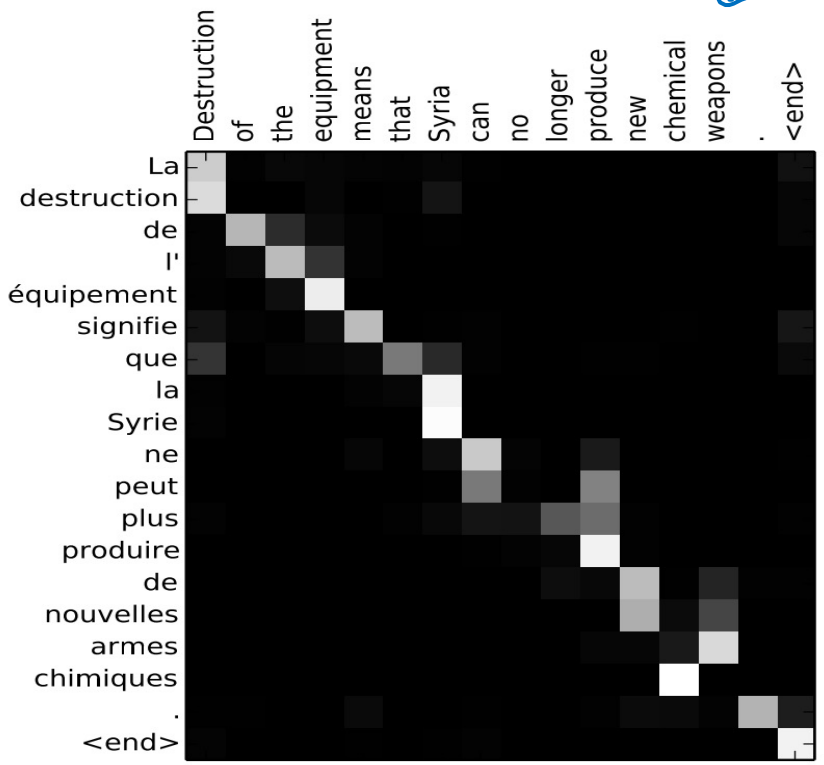
Andrew Ng

Attention examples

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

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

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



Andrew Ng

deeplearning.ai

Audio data

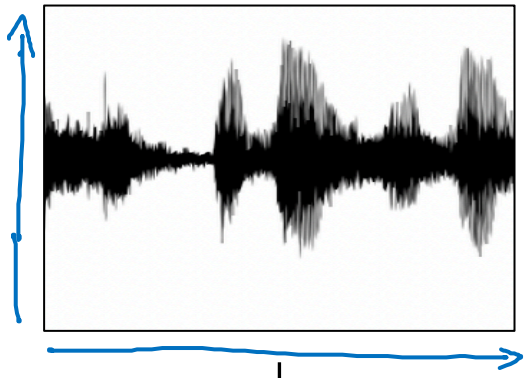
Speech recognition

Speech recognition problem

x → y

audio clip → transcript

detecting air pressures



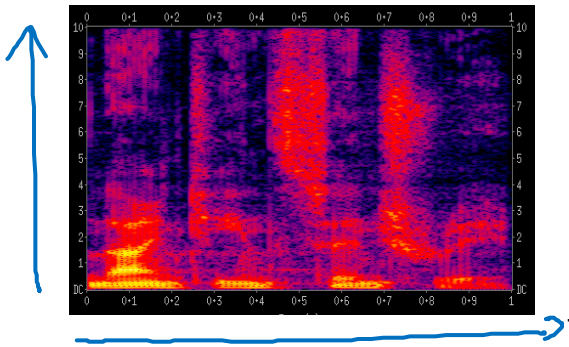
"the quick brown fox"

phonemes: de kwik braun

as basic units of sound

Spectrogram

intensity



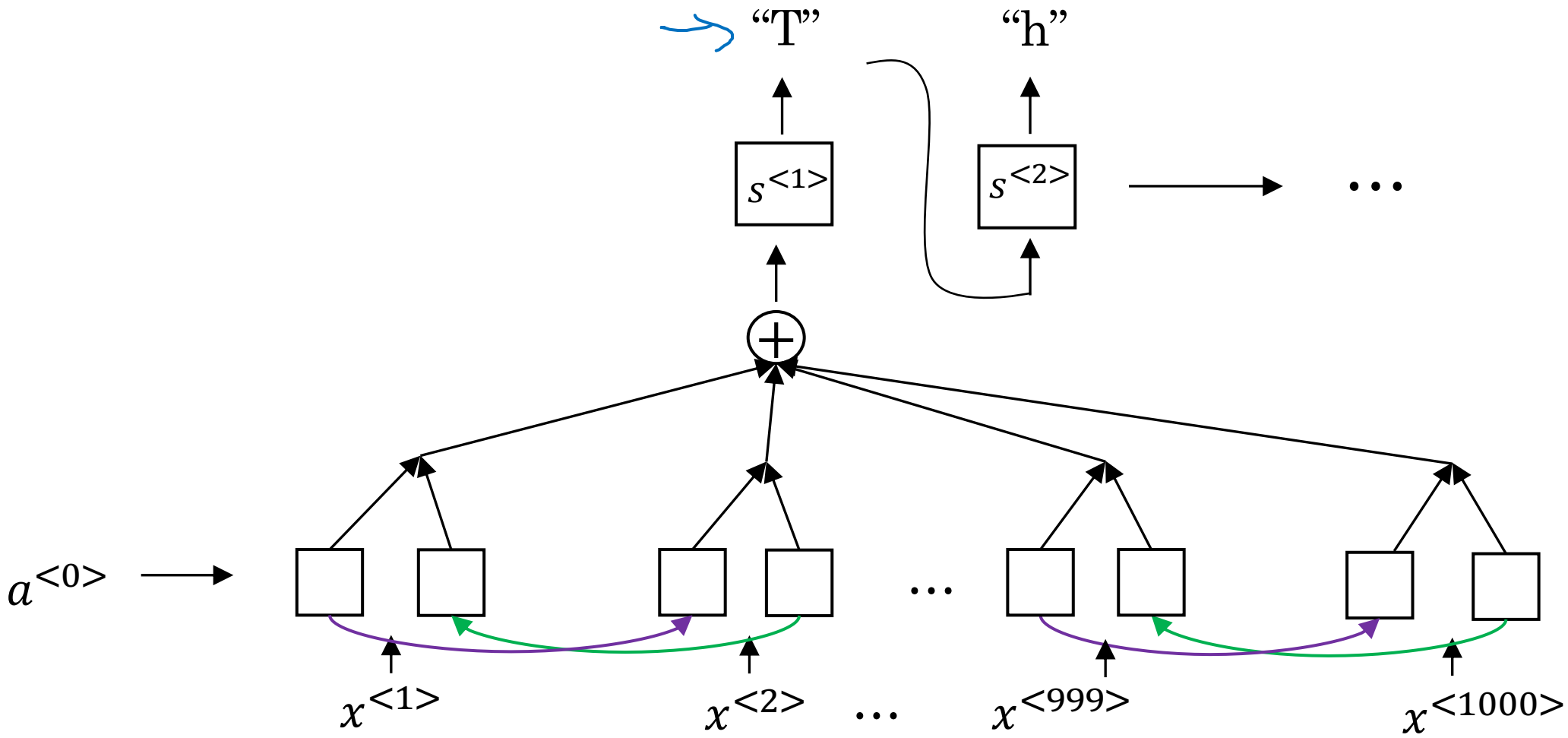
time

previously trained on 300h 3000h

currently 100,000h

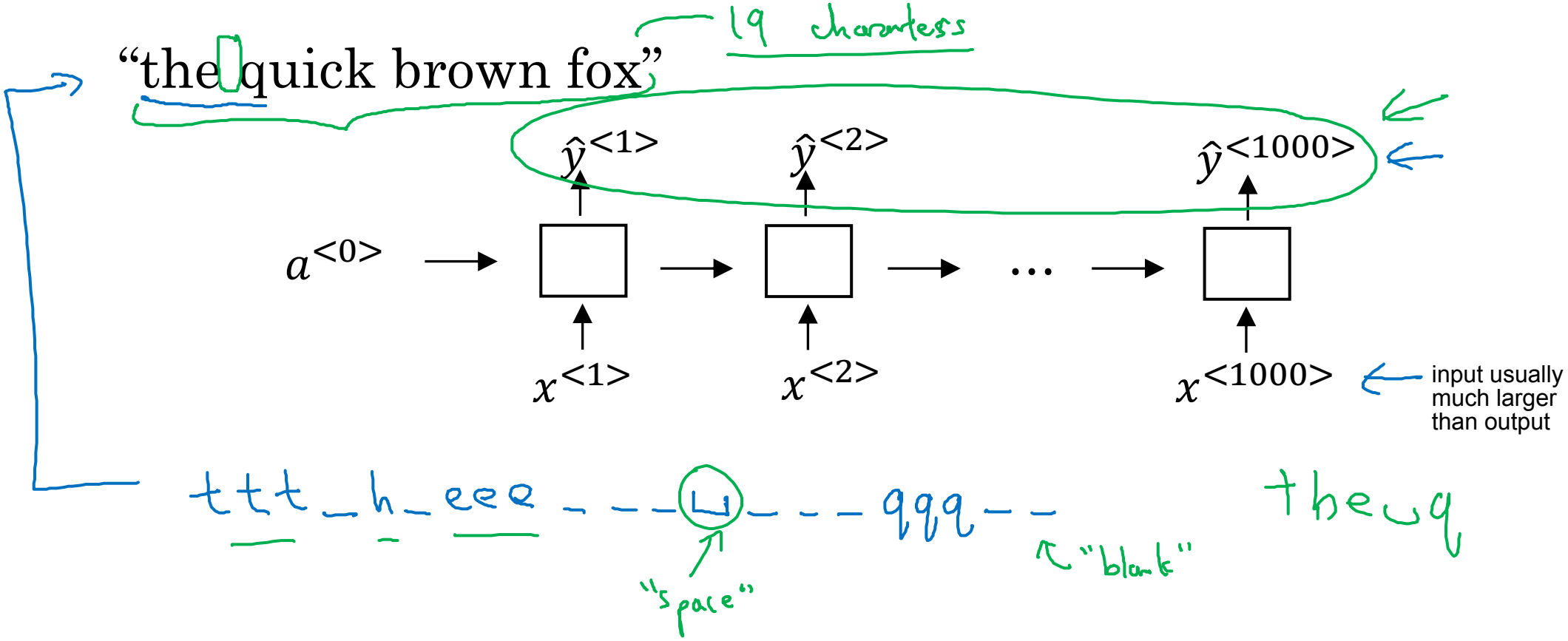
Andrew Ng

Attention model for speech recognition



CTC cost for speech recognition

(Connectionist temporal classification)



Basic rule: collapse repeated characters not separated by "blank"

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

deeplearning.ai

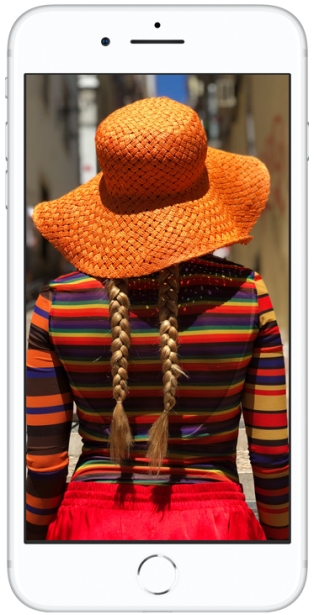
Audio data

Trigger word detection

What is trigger word detection?

Amazon Echo
(Alexa)

Baidu DuerOS
(xiaodunihao)

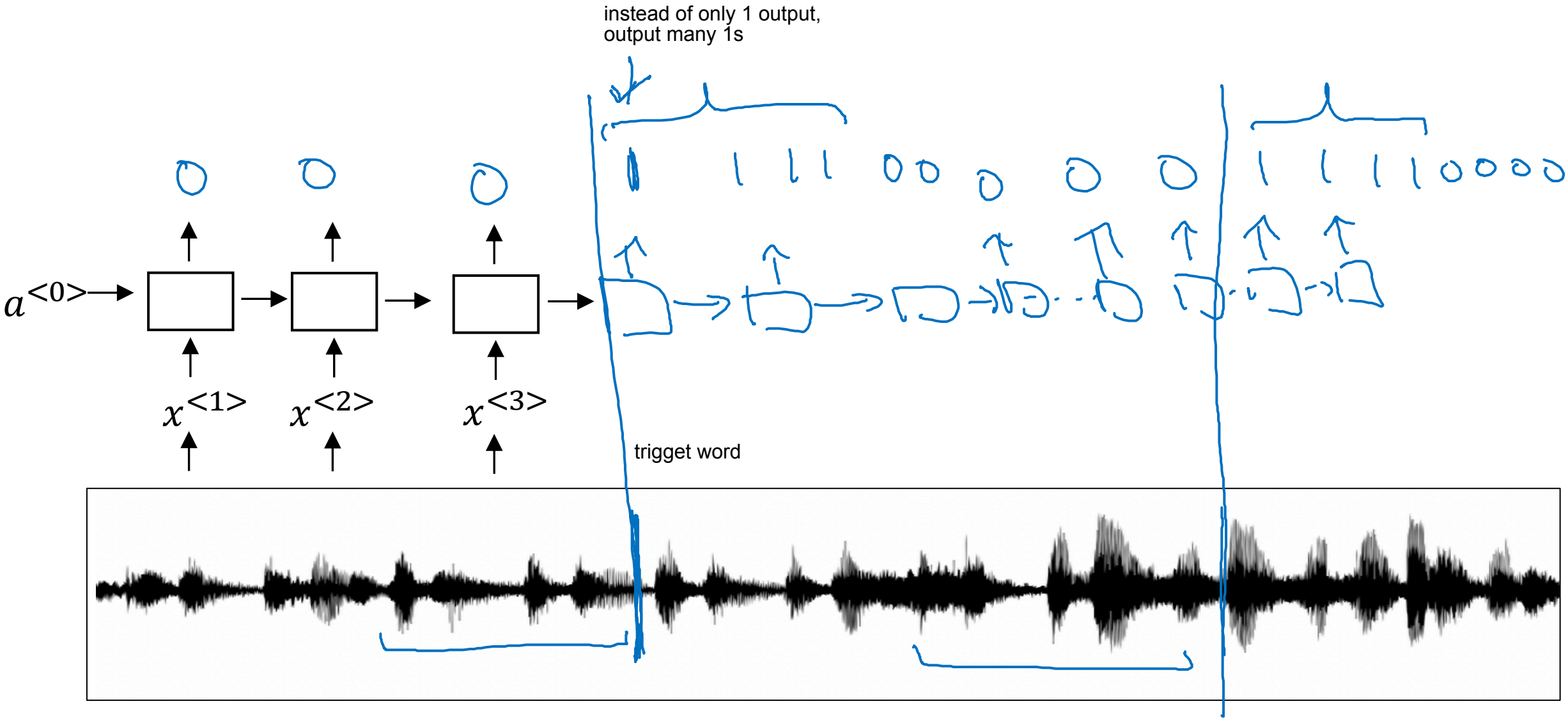


Apple Siri
(Hey Siri)

Google Home
(Okay Google)

Trigger word detection algorithm

no consensus yet on which alg works best



Andrew Ng

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.

Andrew Ng

Thank you.

-Andrew Ng