Applications: machine translation, image captioning (sequence generation from image).

deeplearning.ai

Sequence to sequence models

Basic models

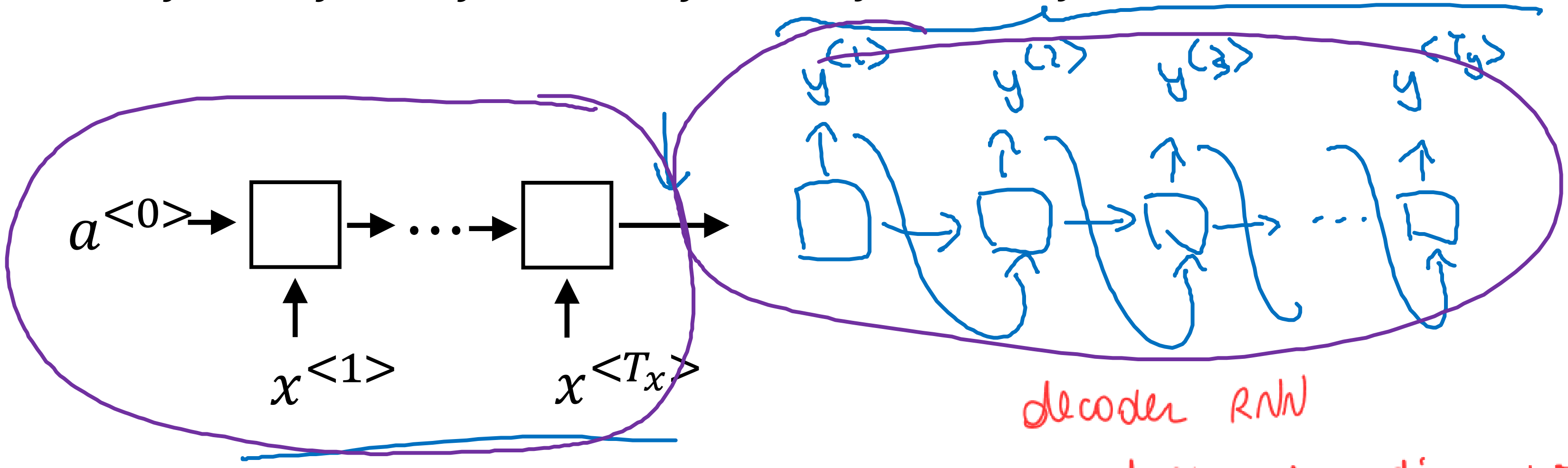
Sequence to sequence model

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

Jane visite l'Afrique en septembre

$\longrightarrow$ Jane is visiting Africa in September.

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



decoder RNN

produces decoding into different language

encoder RNN

encodes original message

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

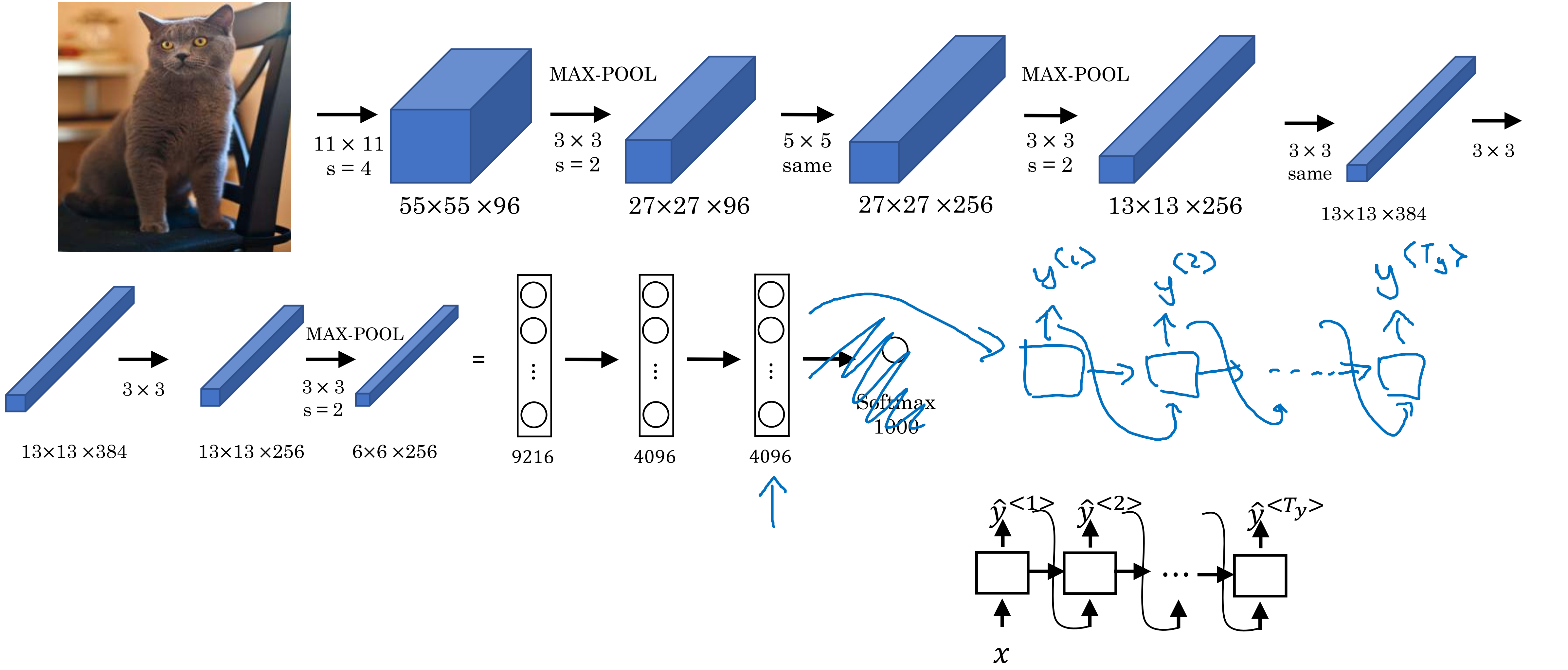
[Cho et al., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation]

Andrew Ng

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]

deeplearning.ai

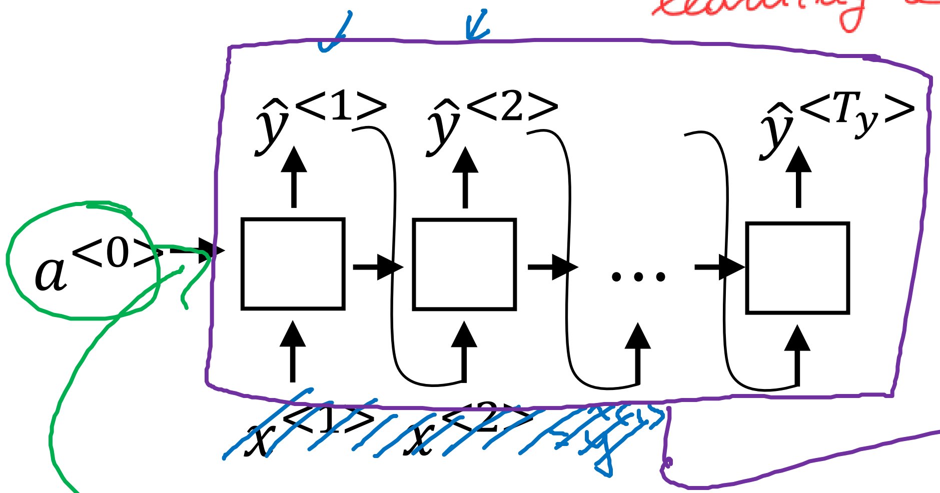
Sequence to sequence models

Picking the most likely sentence

Machine translation as building a conditional language model

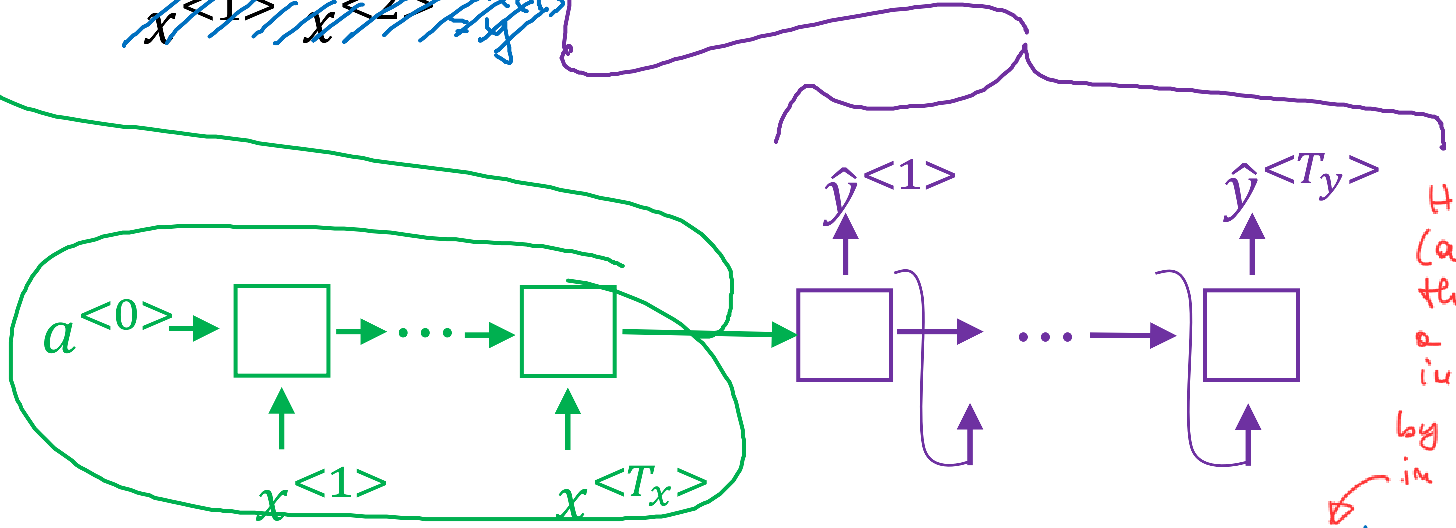
Ng sees machine translation as the task of learning a CONDITIONAL LANGUAGE MODEL. While in language modeling you learn

Language model:



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

Machine translation:



Here you learn (approximate) the probability of a series of words in english conditioned by a series of words in french.

"Conditional language model"

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

Andrew Ng

Finding the most likely translation

Jane visite l'Afrique en septembre.

English sentence

French sentence

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

While decoding you basically sample words from the learned probability distribution, one word at a time. But this could represent a suboptimal translation and the words sequence $y^{<1>}, y^{<2>}, \dots, y^{<T_y>}$ could change every time.

→ 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.

=> Use a search algorithm to find the sequence $y^{<1>}, \dots, y^{<T_y>}$ that maximizes the prob.

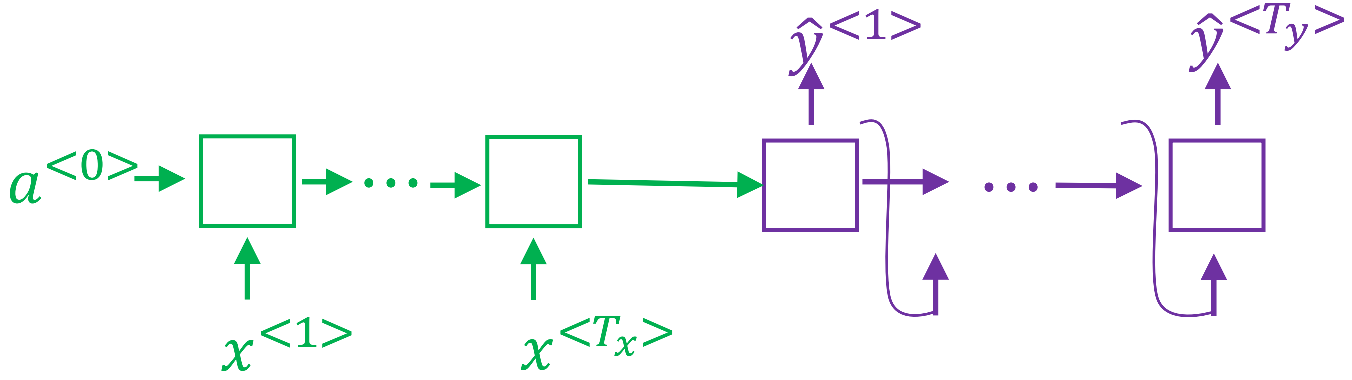
$$\arg\max_{y^{<1>},\dots,y^{<T_y>}} P(y^{<1>}, \dots, y^{<T_y>} \mid x)$$

usually BEAM search is used instead of greedy approaches.

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

$\underline{10{,}000^{10}}$

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

→ 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)

Beam search is an extension of greedy search where the B greediest options are selected at each step.
- At step 1 the B=3 most likely words are selected (st. highest $P(y^{<1>}|x) = \hat{y}^{<1>1}$, 2nd highest $P(y^{<1>}|x) = \hat{y}^{<1>2}$, 3rd highest $P(y^{<1>}|x) = \hat{y}^{<1>3}$)
- At step 2, for each of those 3 words, the set of possible following words is computed and the 3 most likely sequences $y^{<1>}, y^{<2>}$ are determined.

Sequence to sequence models

- At step 3, repeat step 2 using the three sequences of words.

Each of the B "branches" is closed when the sequence generates EOL.

This algorithm reduces the greediness of pure greedy search.

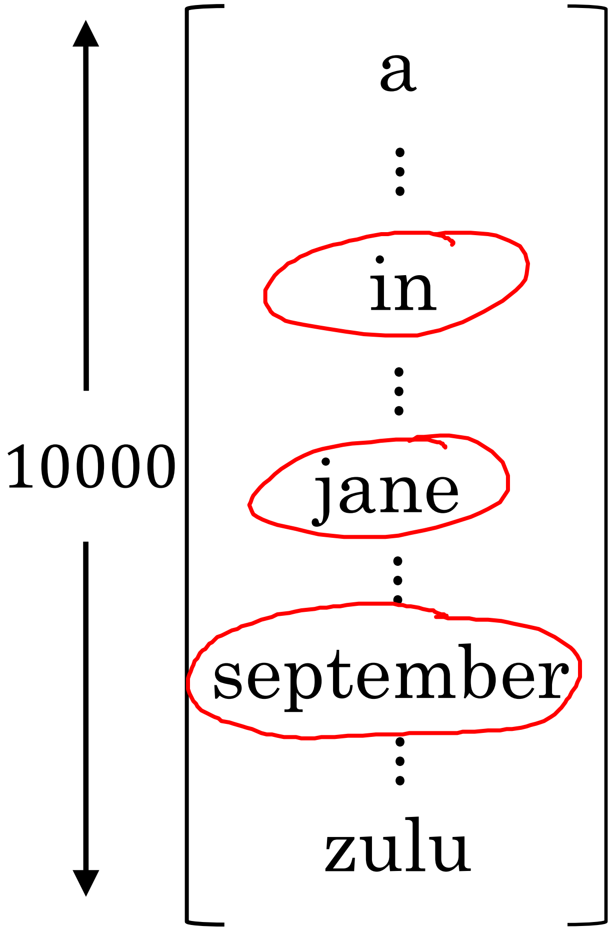
Beam search

deeplearning.ai

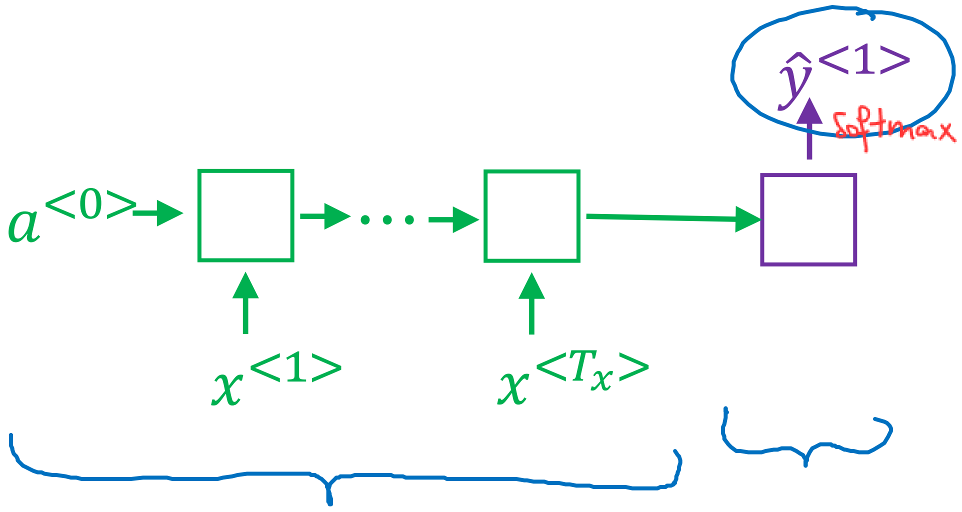
Beam search algorithm

$B = 3$ (beam width)

Step 1

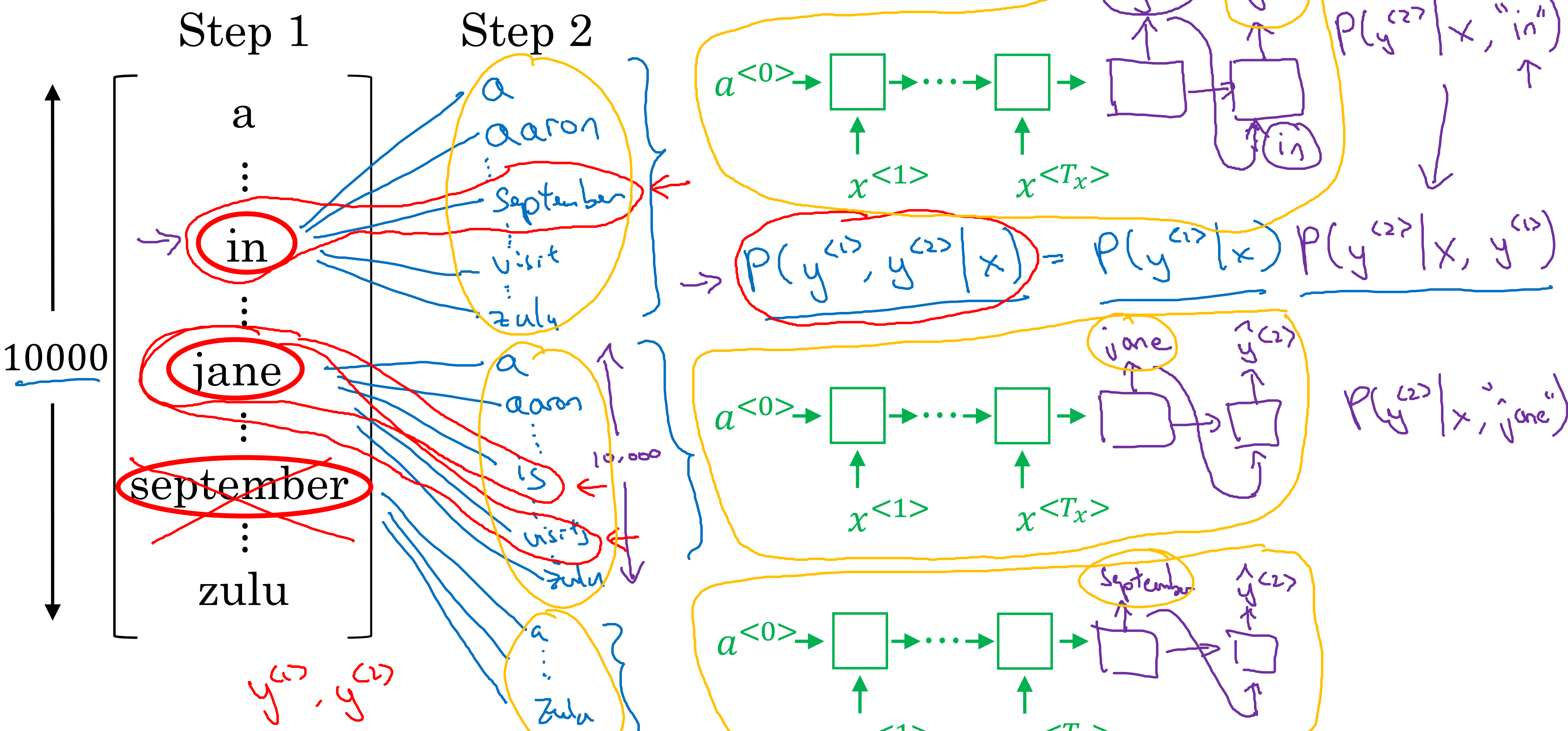


$$\rightarrow P(y^{<1>} \mid x)$$



Beam search algorithm

$(B = 3)$



Andrew Ng

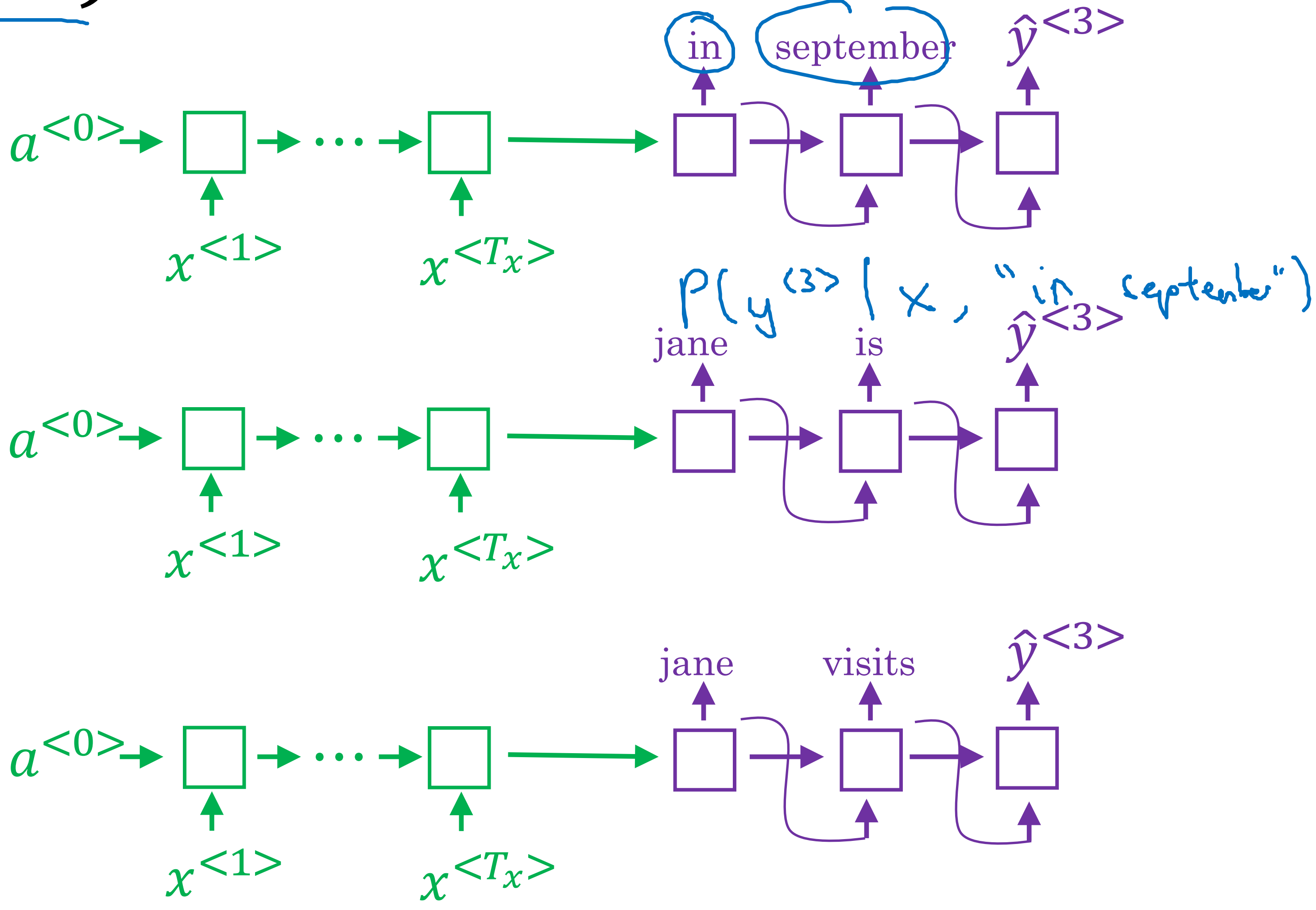
Beam search ($B = 3$)

$B = 1 \rightsquigarrow$ greedy search

in september — a, aaron, jane, zulu

jane is — a, visits, zulu

jane visits — a, africa, zulu



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

jane visits africa in september. <EOS>

Andrew Ng

The B sequences returned by beam search may have different length because sequences are terminated when they are among the B most likely and terminate with a <EOL>.
For example at the end of beam search you may have: "Jane likes apples <EOL>", "Jane likes eating apples <EOL>", "Jane eats the apple <EOL>".

Sequence to sequence models

Longer sentences are more penalized by the log loss.

Therefore one can normalize the loss by the length of the sequence.

Refinements to beam search

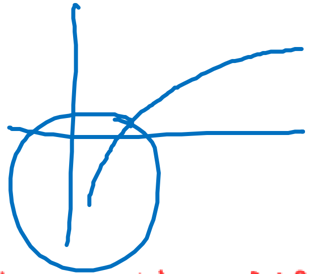
deeplearning.ai

Length normalization

what we do in sequence generation is $P(y^{<1>} \dots y^{<T_y>} | x) = P(y^{<1>}|x)\, P(y^{<2>}|x, y^{<1>}) \dots P(y^{<T_y>}|x, y^{<1>}, \dots, y^{<T_y-1>})$

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

which is essentially equivalent to log



$\log P(y|x)$

$P(y|x)$

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

(which is more numerically stable (avoid rounding errors due to very small values arising from many multiplications).

This value is inversely proportional to the length of the sequence. (= Sentence becomes less likely the more long it is).

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

To remove this dependency we normalize this formula by the length of the sentence

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

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

α regulates the amount of normalization

BUT HOW DOES THIS INFLUENCE SEQUENCE GENERATION IN PRACTICE?

Andrew Ng

Beam search discussion

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

Beam width B?

1 → 3 → 10, 100, 1000, → 3000

production system

research applications

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

Mispredictions in a sequence generation problem can be due to RNN errors or the beam search process. Error analysis can be performed on the mispredicted sentences (of the training or development set depending on whether to investigate sources for high bias or variance) to determine who's responsible for the error. For the mispredicted sample yhat collect the human ground truth y*. Compute the probabilities of yhat and t* by plugging them into the RNN. If P(y*|x) > P(y-hat|x), beam search is at fault.If P(y*|x) ≤ P(y-hat|x), the RNN model is at fault. Beam search can be improved increasing the beam size B.

Sequence to sequence models

deeplearning.ai

Error analysis on beam search

Example

→ RNN

→ Beam Search

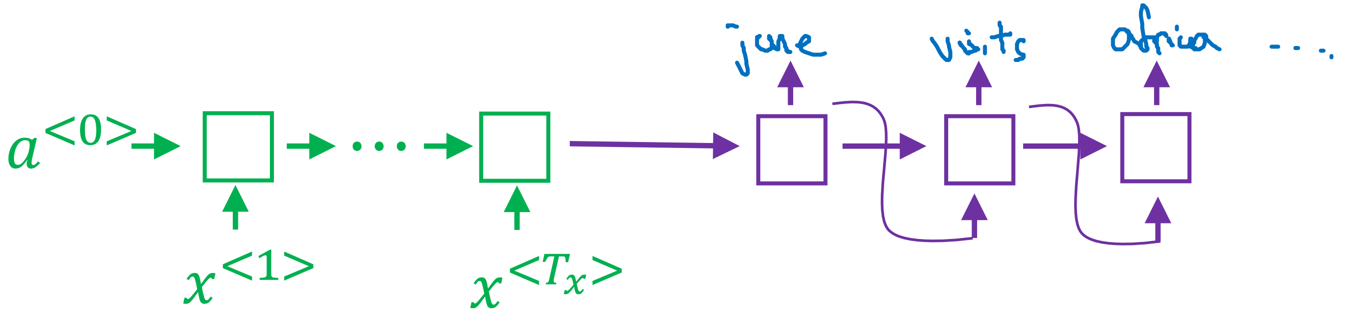
$\boxed{\text{BT}}$

Jane visite l'Afrique en septembre.

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

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

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



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. $(\hat{y})$

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

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.

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

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:

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

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
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. $(\hat{y})$

$p_1, p_2, = \underline{1.0}$

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

unigram

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

n-gram

$$p_n = \frac{\sum_{n\text{-grams} \in \hat{y}} count_{clip}(n\text{-gram})}{\sum_{n\text{-grams} \in \hat{y}} count(n\text{-gram})}$$

[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

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

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

The Attention Model is a modification to the Encoder-Decoder architecture for machine translation that allows it to perform better, especially on long sentences. Instead of memorizing the entire input sentence before translating, the Attention Model focuses on parts of the input while generating the output. Attention weights are used to determine how much focus should be given to each input word while generating a certain output.

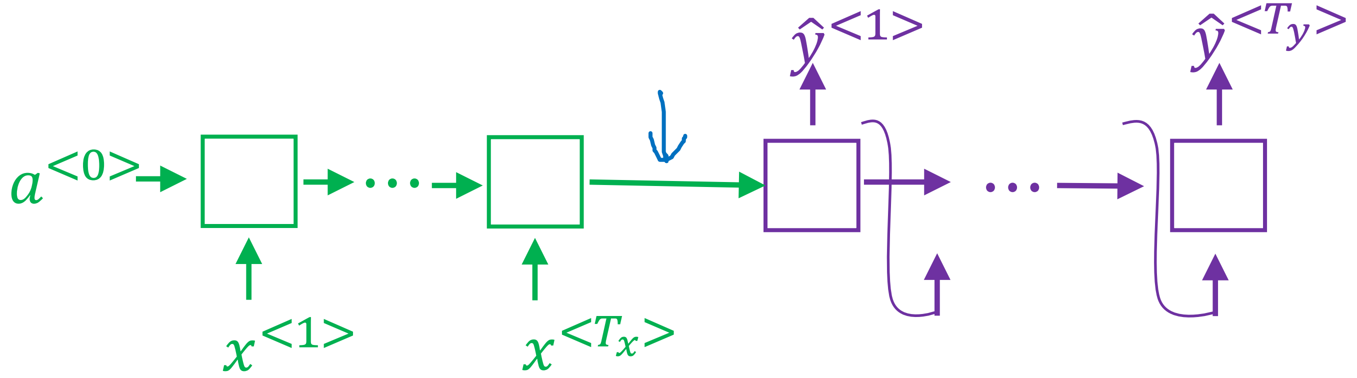
Sequence to sequence models

deeplearning.ai

Attention model intuition

Specifically, the decoder at the step t receives yhat^<t-1> as input, along with a set of attention weights that tell which input words serve as context for the t-th prediction. The weights had been computed by the encoder prior to that. Note that if Tx and Ty are the lengths of the input and ouput sentences there will be Ty sets of attention weights and each set contains Tx weights.

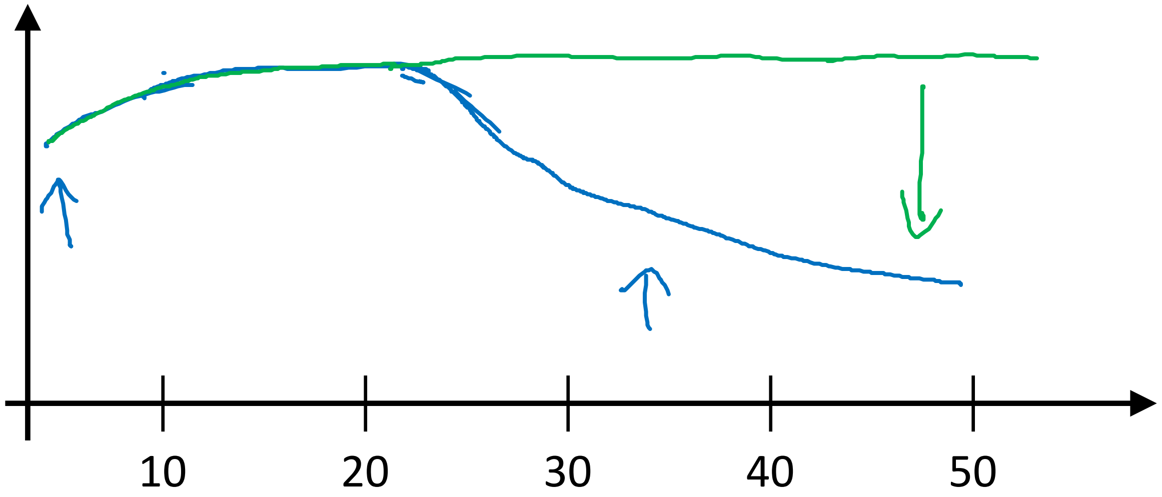
The problem of long sequences



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.

Bleu score

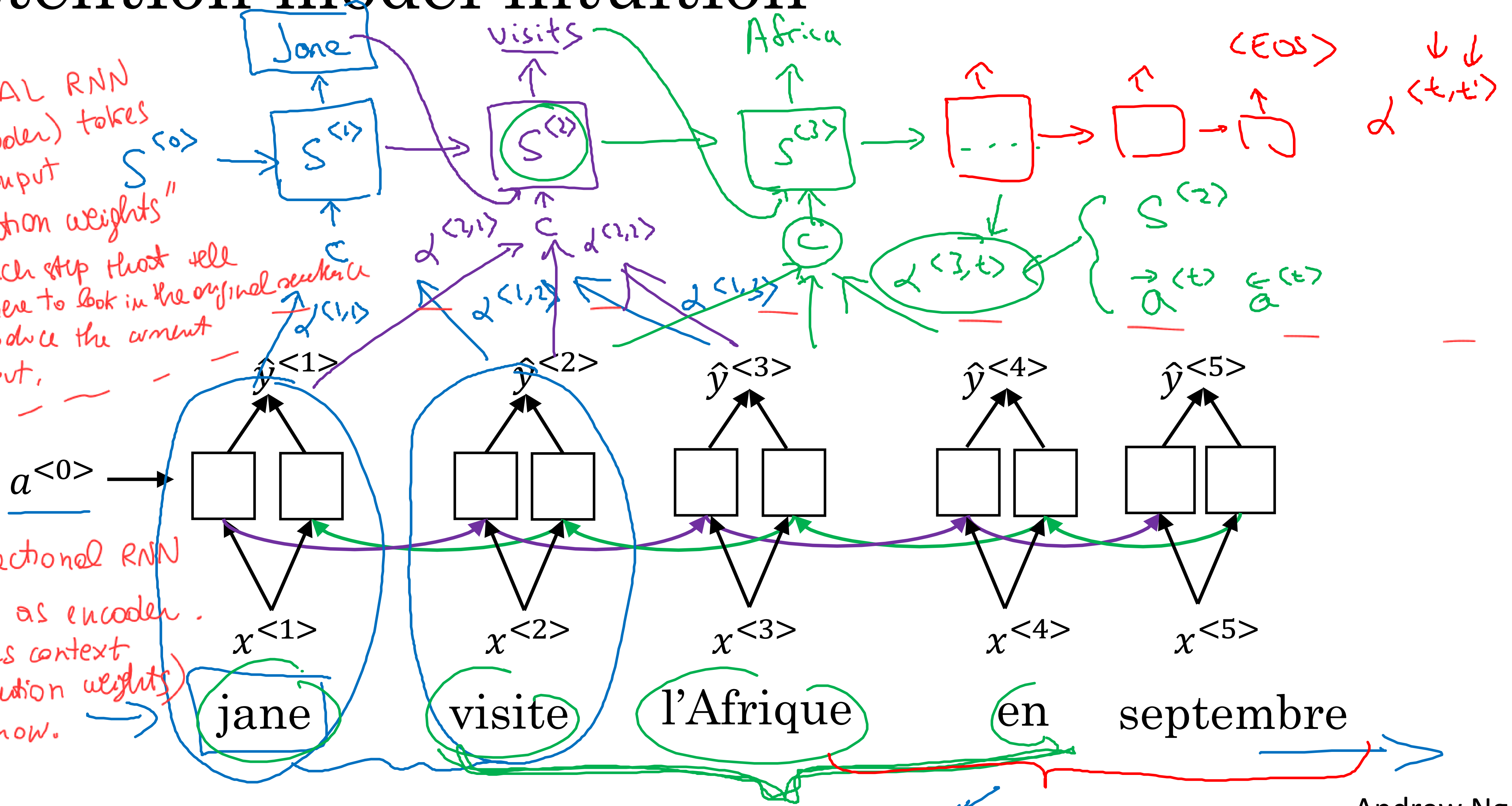


Sentence length

Attention model intuition

② USUAL RNN (decoder) takes $S^{<0>}$ as input "attention weights" at each step that tells it where to look in the original sentence to produce the current output.

① Bidirectional RNN used as encoder. (learns context = attention weights somehow.



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

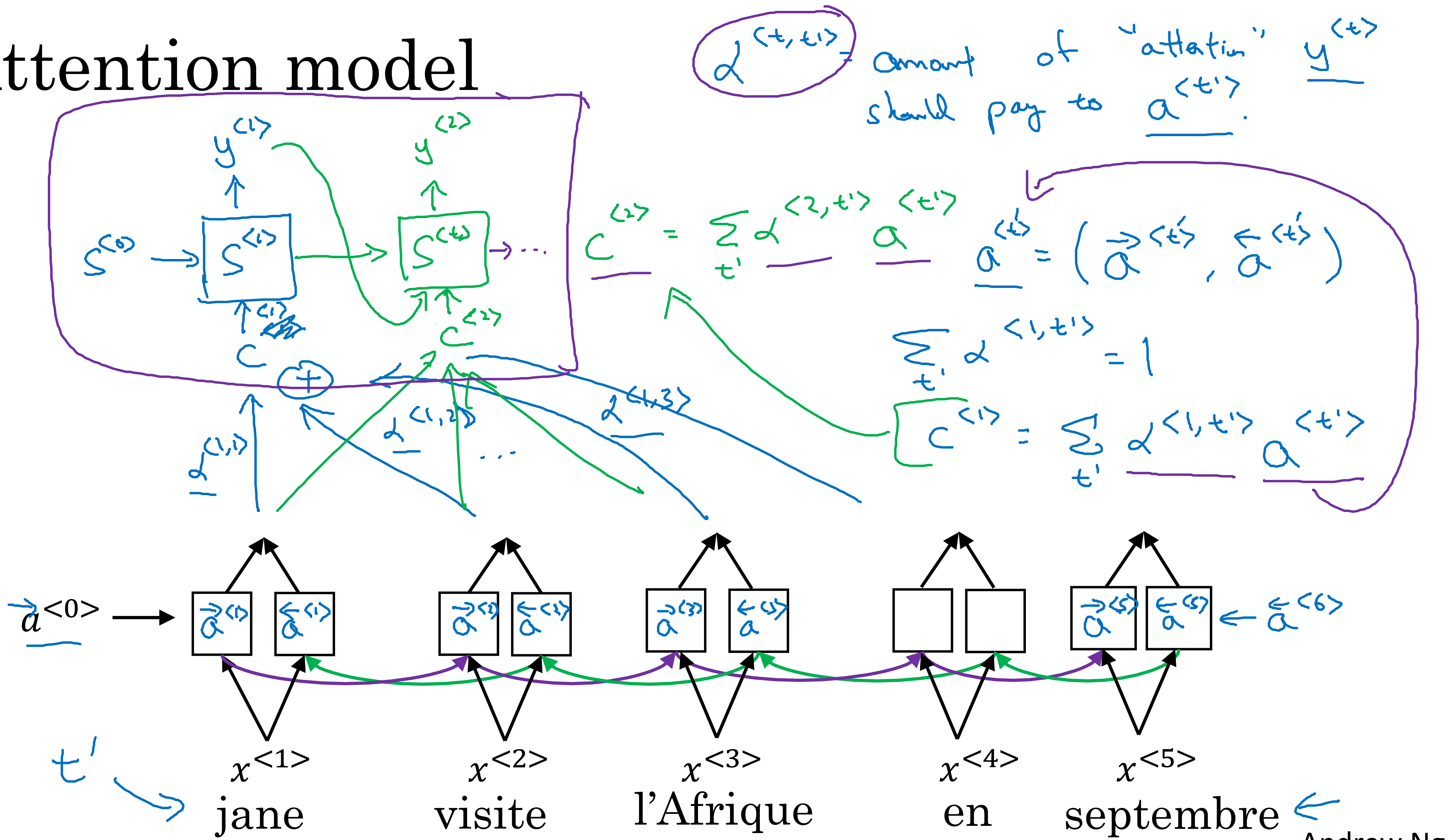
Andrew Ng

deeplearning.ai

Sequence to sequence models

Attention model

Attention model



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

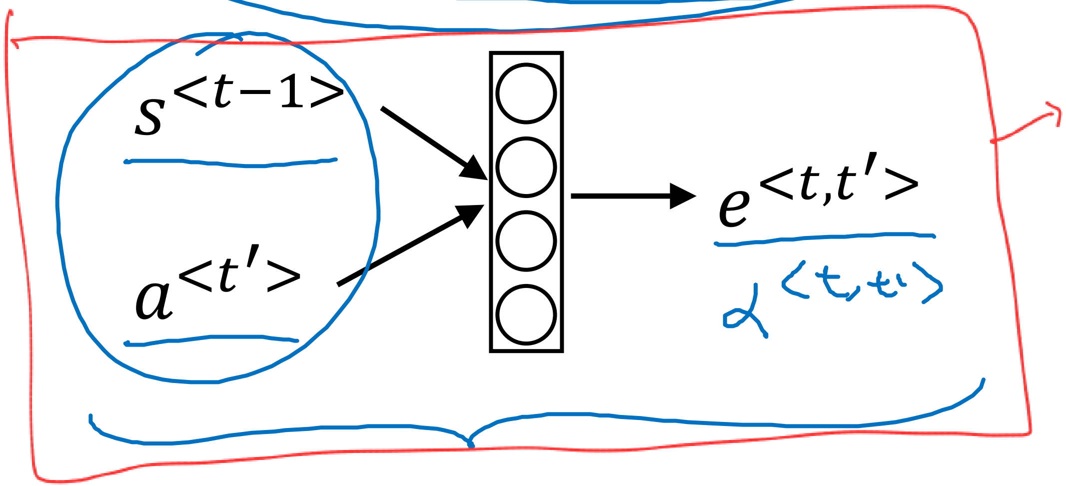
Andrew Ng

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

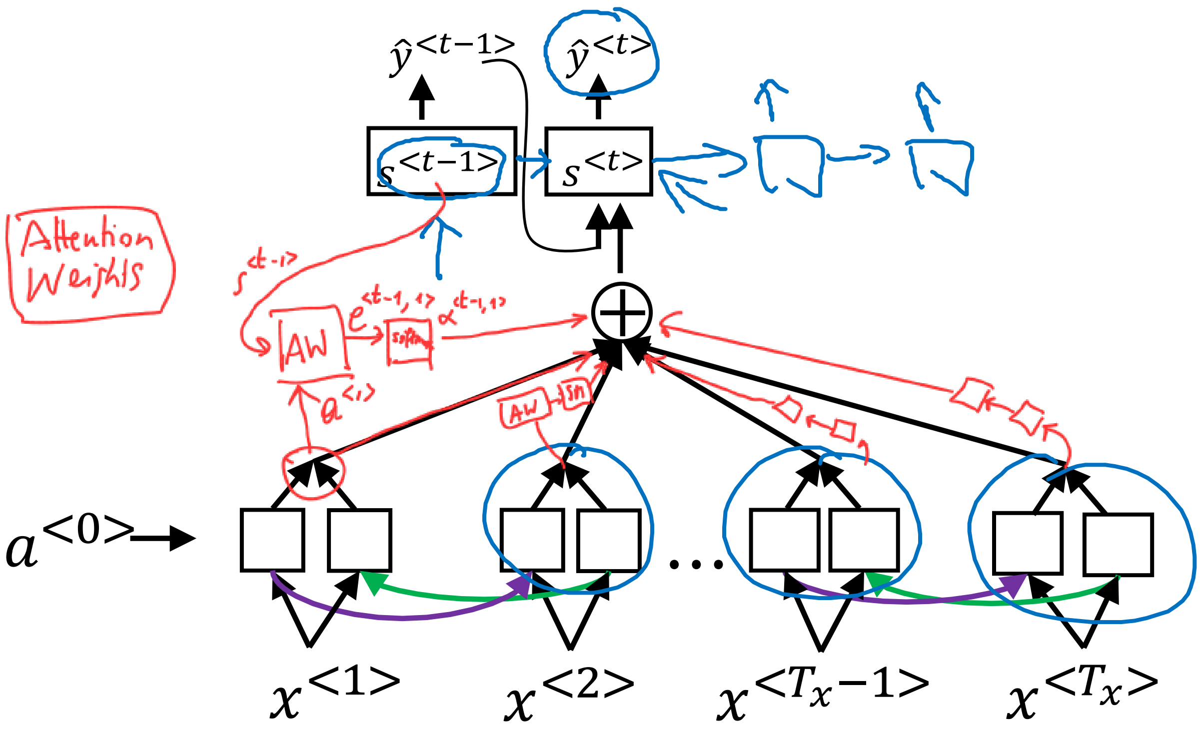
T_x T_y

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

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



This node is plugged into the whole ENC-DEC architecture



[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]

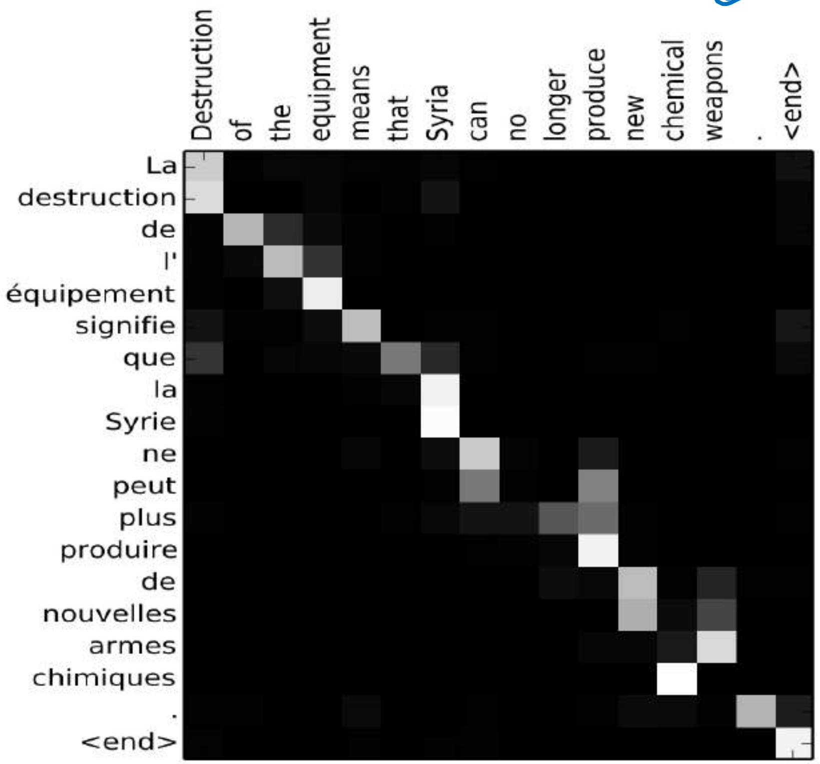
Andrew Ng

Attention examples

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

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

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



deeplearning.ai

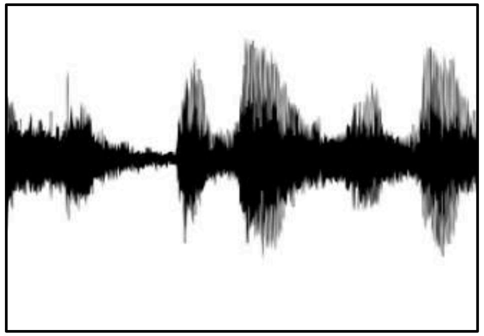
Audio data

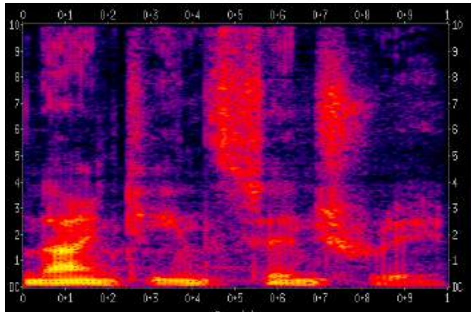
Speech recognition

Speech recognition problem

x → y

audio clip → transcript





"the quick brown fox"

phonemes: de kwik braun

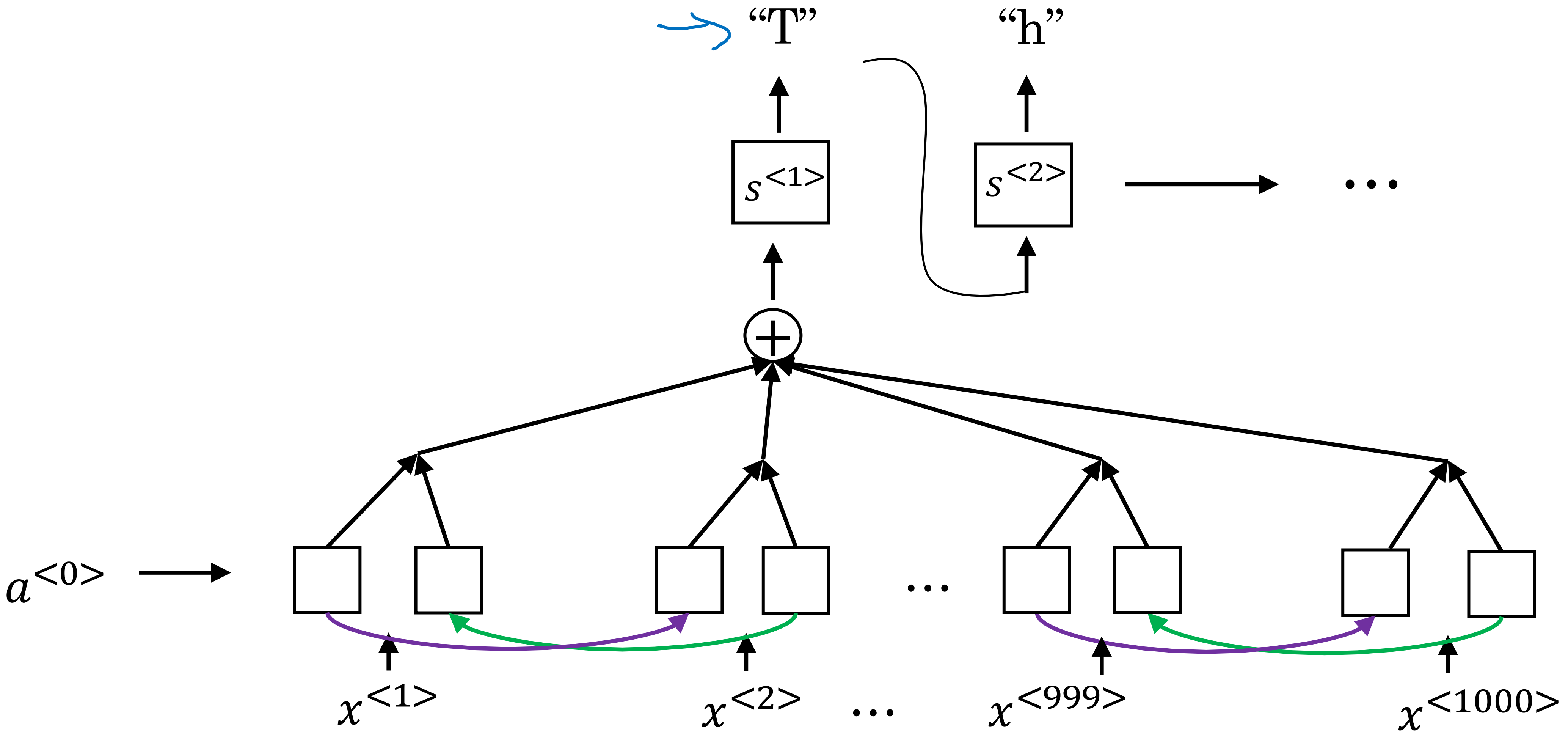
300h

3000h

100,000h

Andrew Ng

Attention model for speech recognition



Andrew Ng

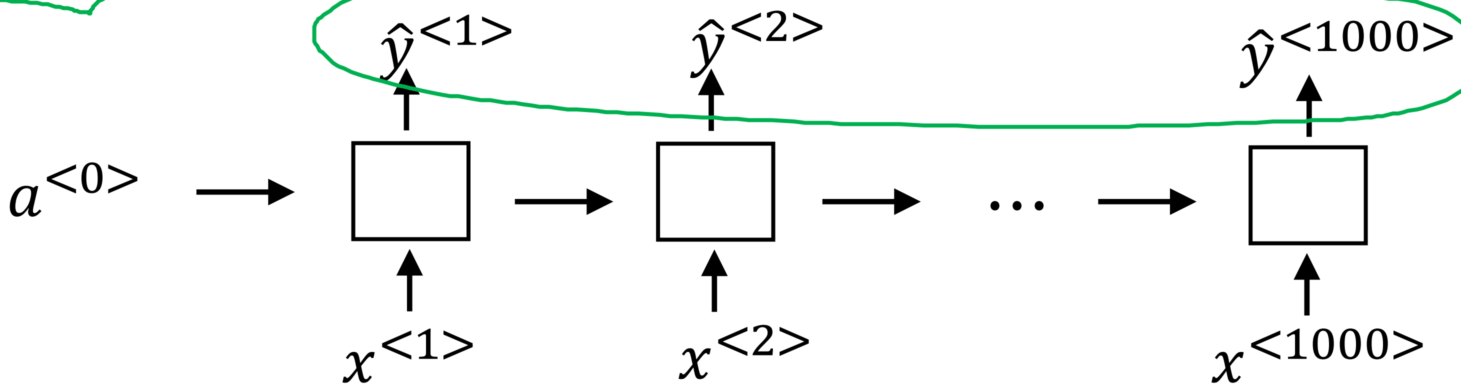
CTC cost for speech recognition

(Connectionist temporal classification)

- Use same number of inputs and outputs
-

"the quick brown fox"

19 characters



ttt_h_eee___⊔___qqq__

"space"

"blank"

the⊔q

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?

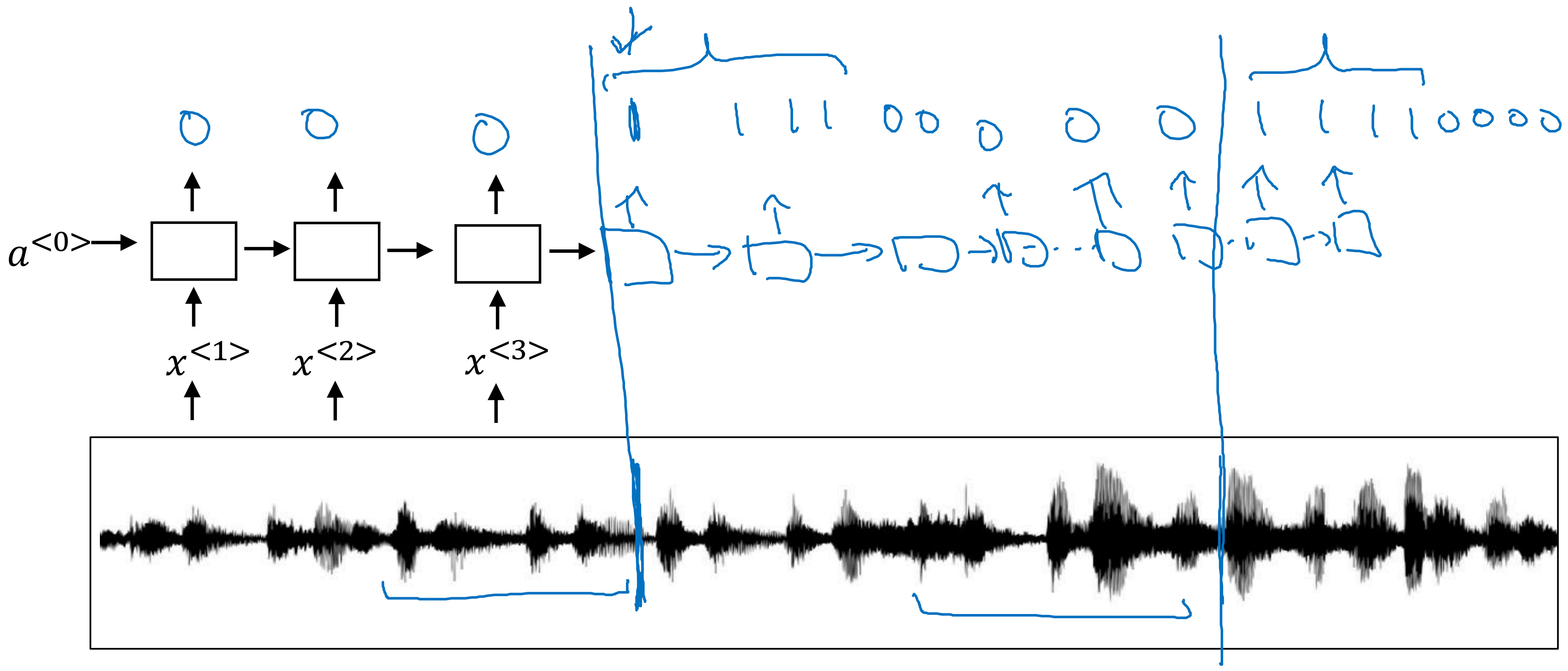
Amazon Echo
(Alexa)

Baidu DuerOS
(xiaodunihao)

Apple Siri
(Hey Siri)

Google Home
(Okay Google)

Trigger word detection algorithm



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