



deeplearning.ai

# Sequence to sequence models

---

## Basic models

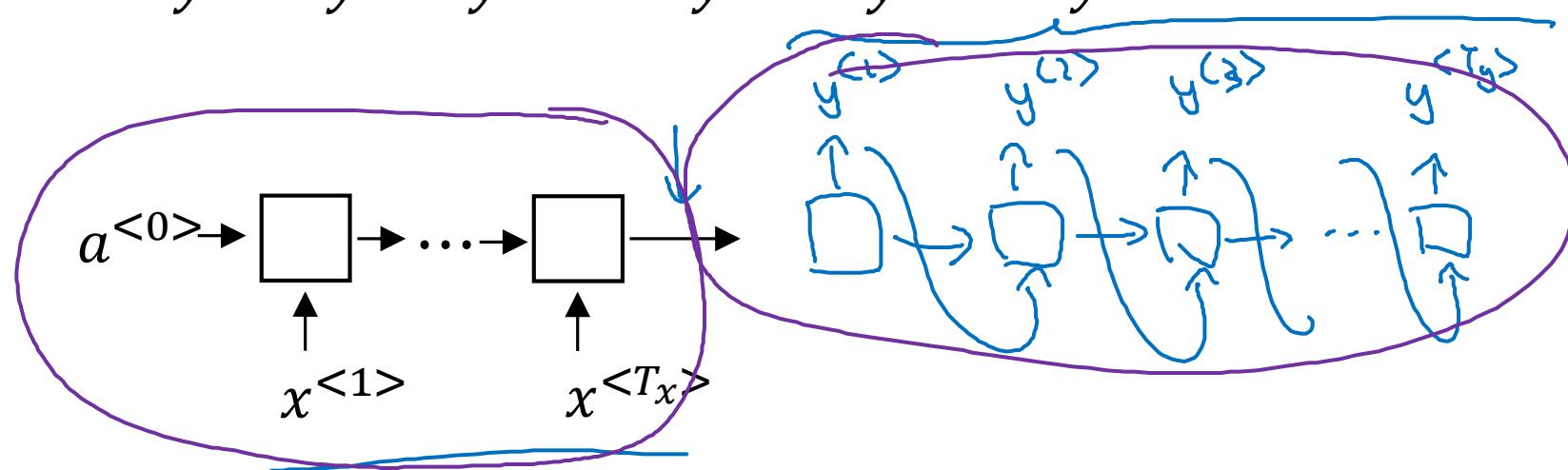
# Sequence to sequence model

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

Jane visite l'Afrique en septembre

→ Jane is visiting Africa in September.

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

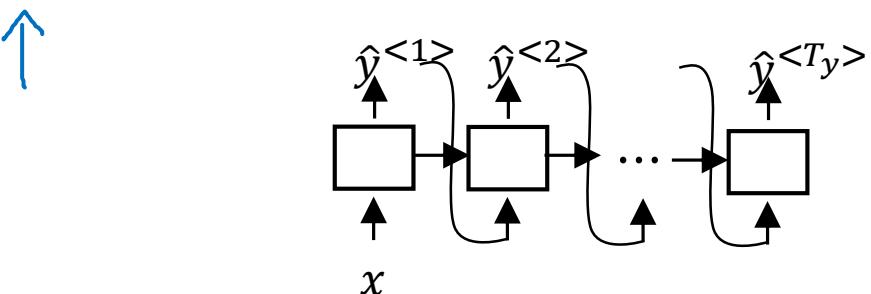
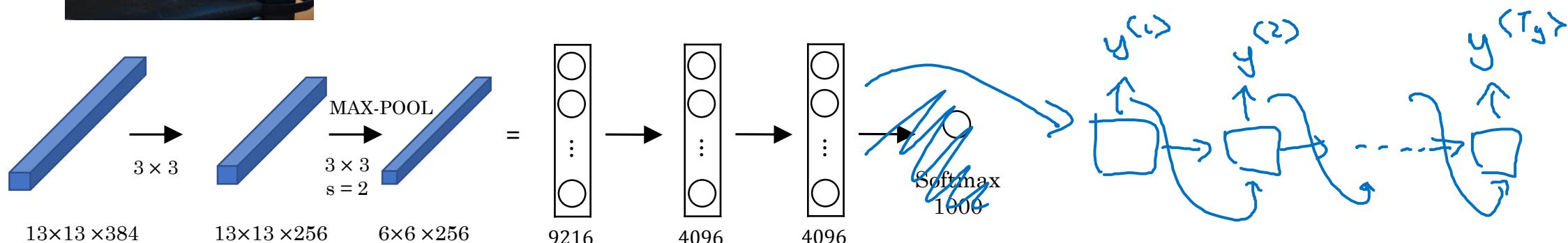
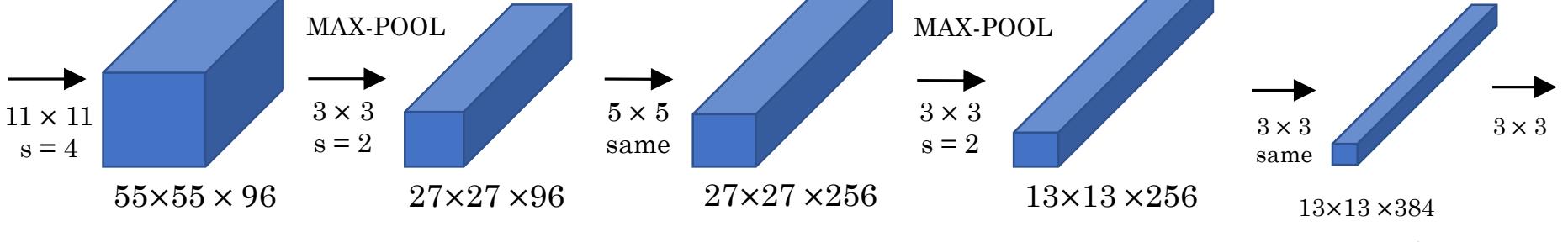
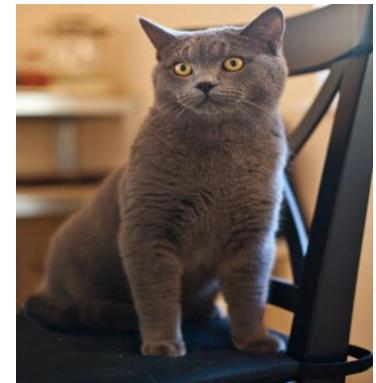


[Sutskever et al., 2014. Sequence to sequence learning with neural networks] ↪

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

Andrew Ng

# 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]

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



deeplearning.ai

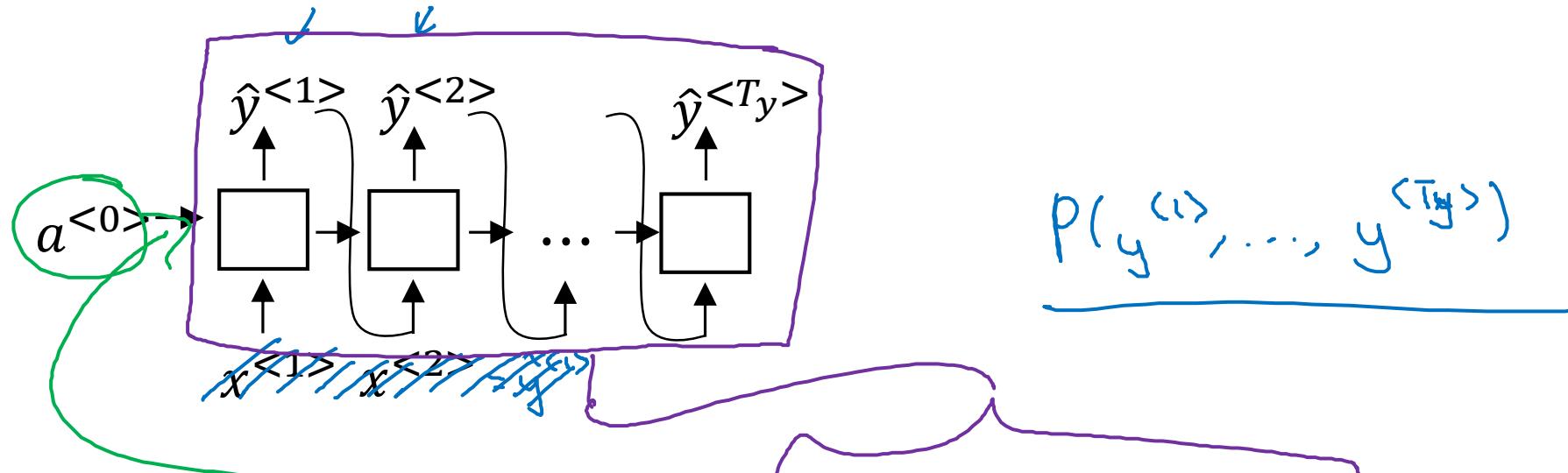
# Sequence to sequence models

---

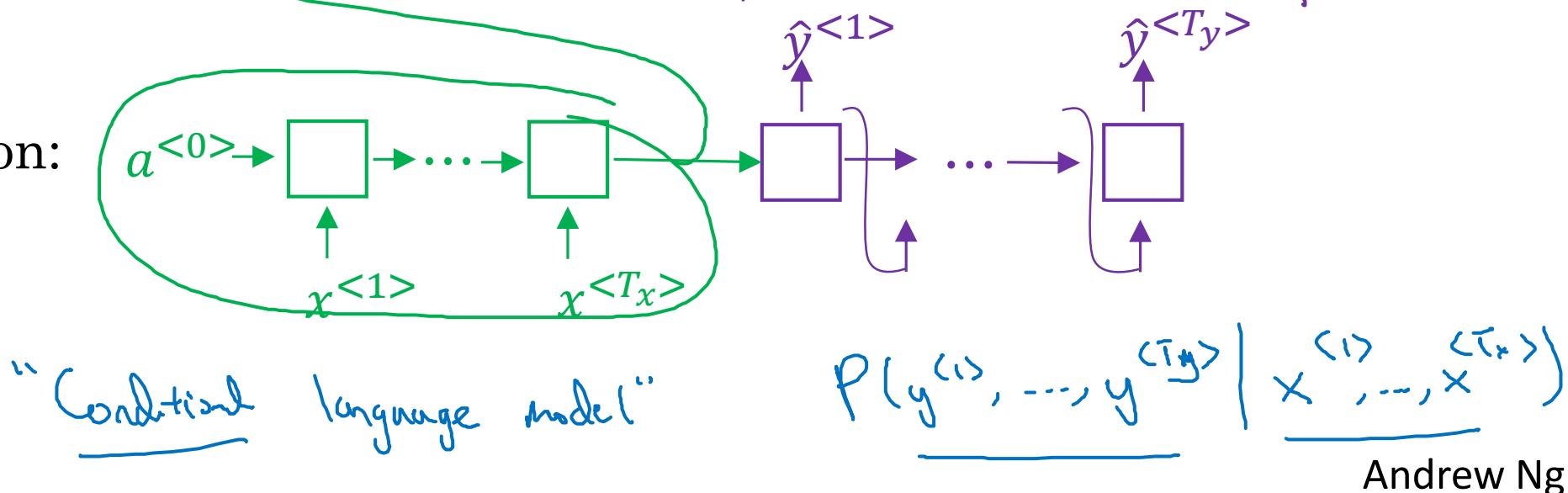
Picking the most  
likely sentence

# Machine translation as building a conditional language model

Language model:



Machine translation:



# Finding the most likely translation

Jane visite l'Afrique en septembre.

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

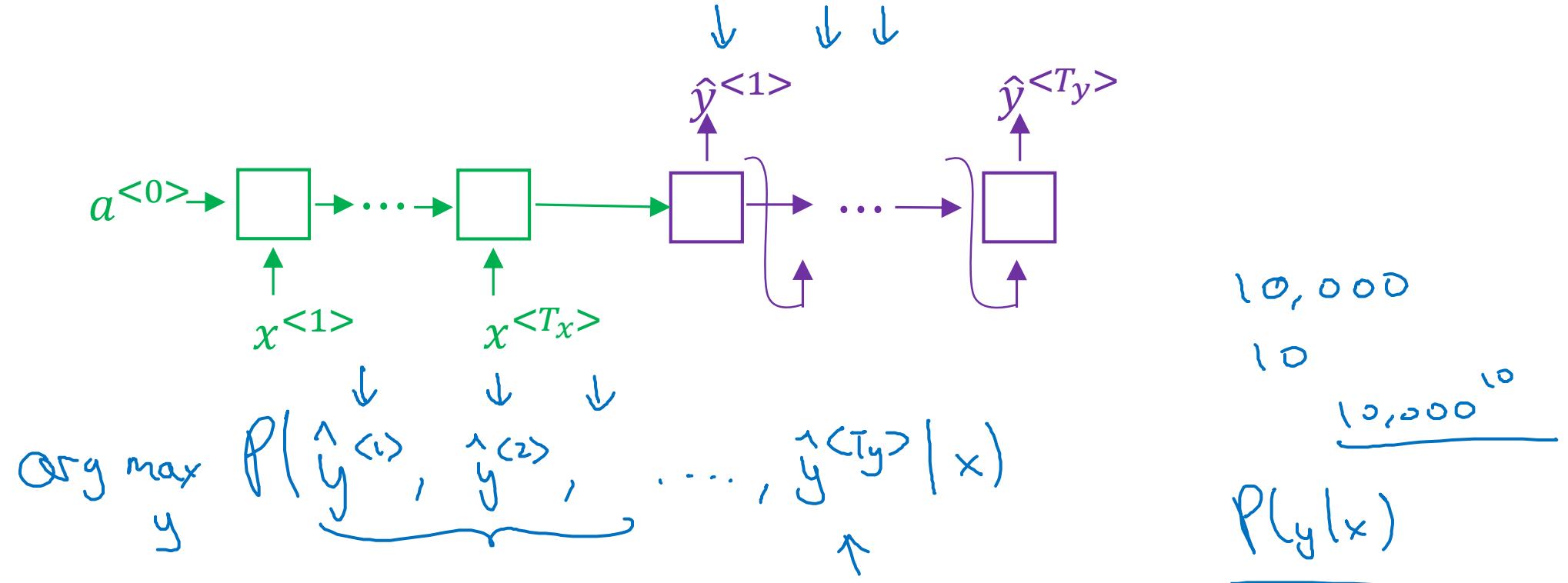
French  
↓  
English

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

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

# Why not a greedy search?

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



→ Jane is visiting Africa in September.

→ Jane is going to be visiting Africa in September.

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



deeplearning.ai

# Sequence to sequence models

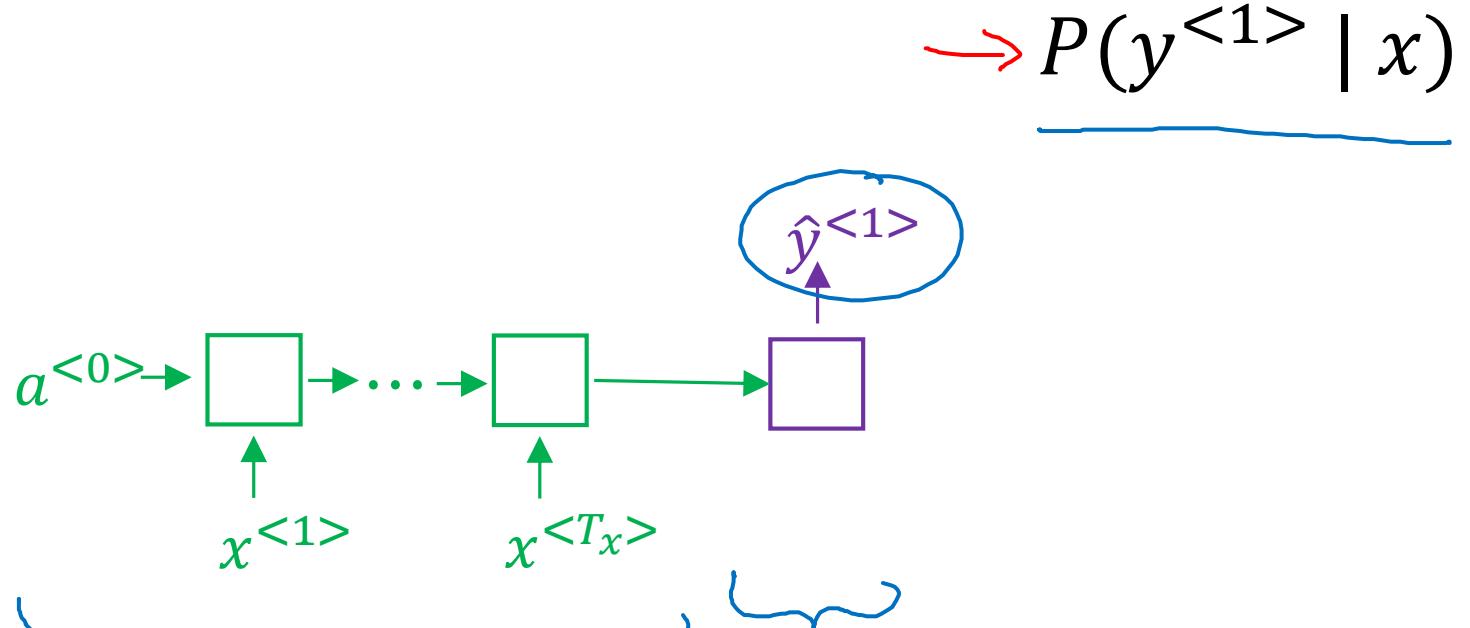
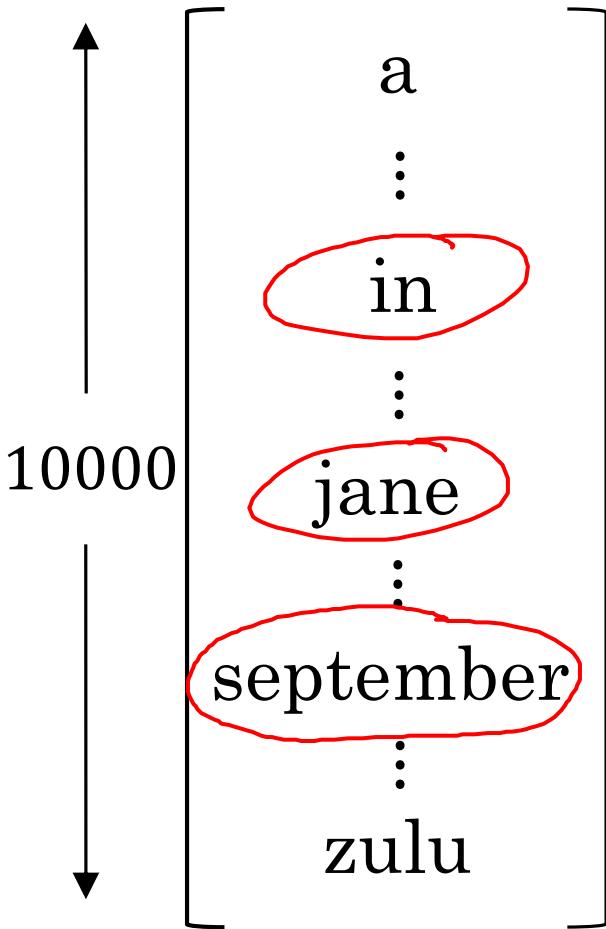
---

## Beam search

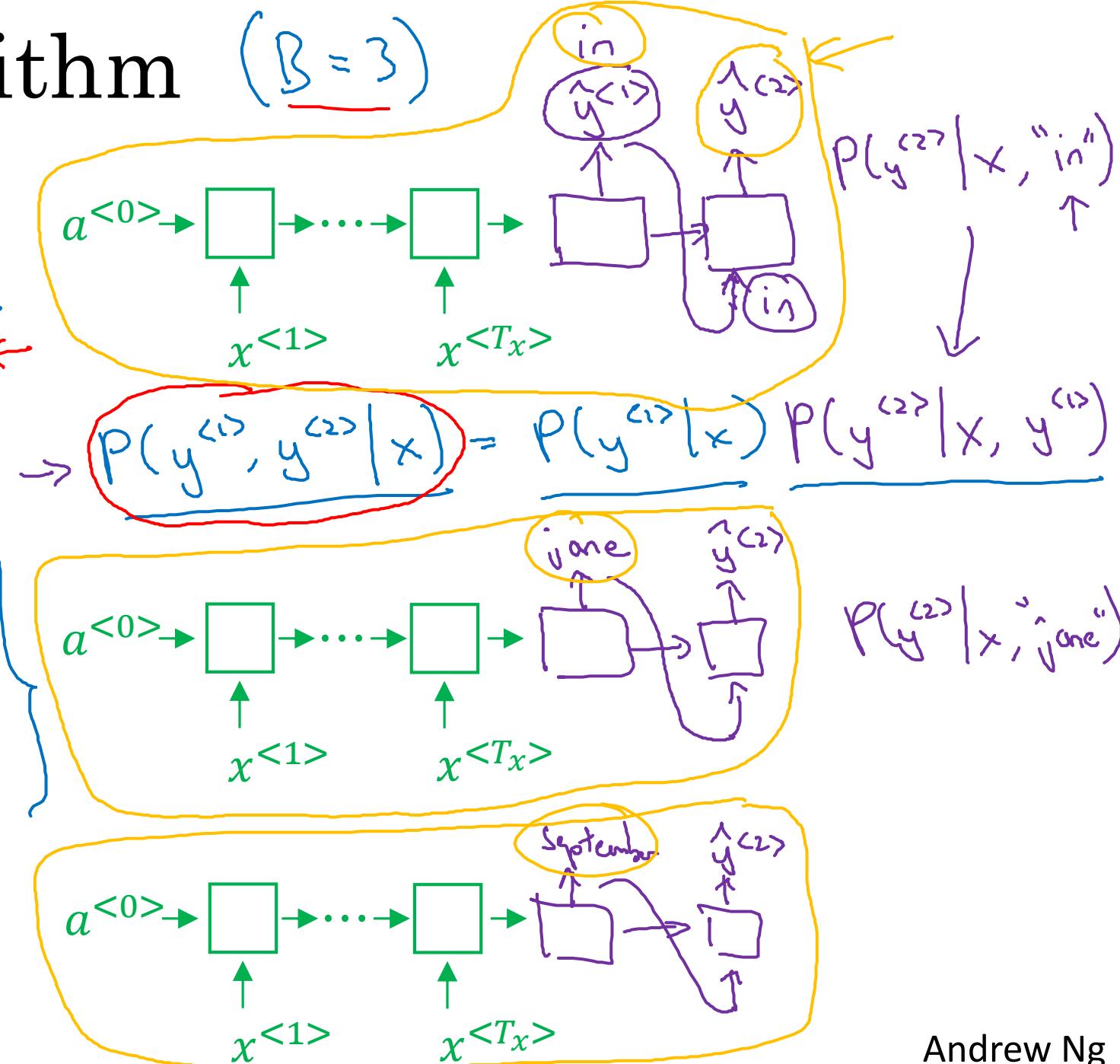
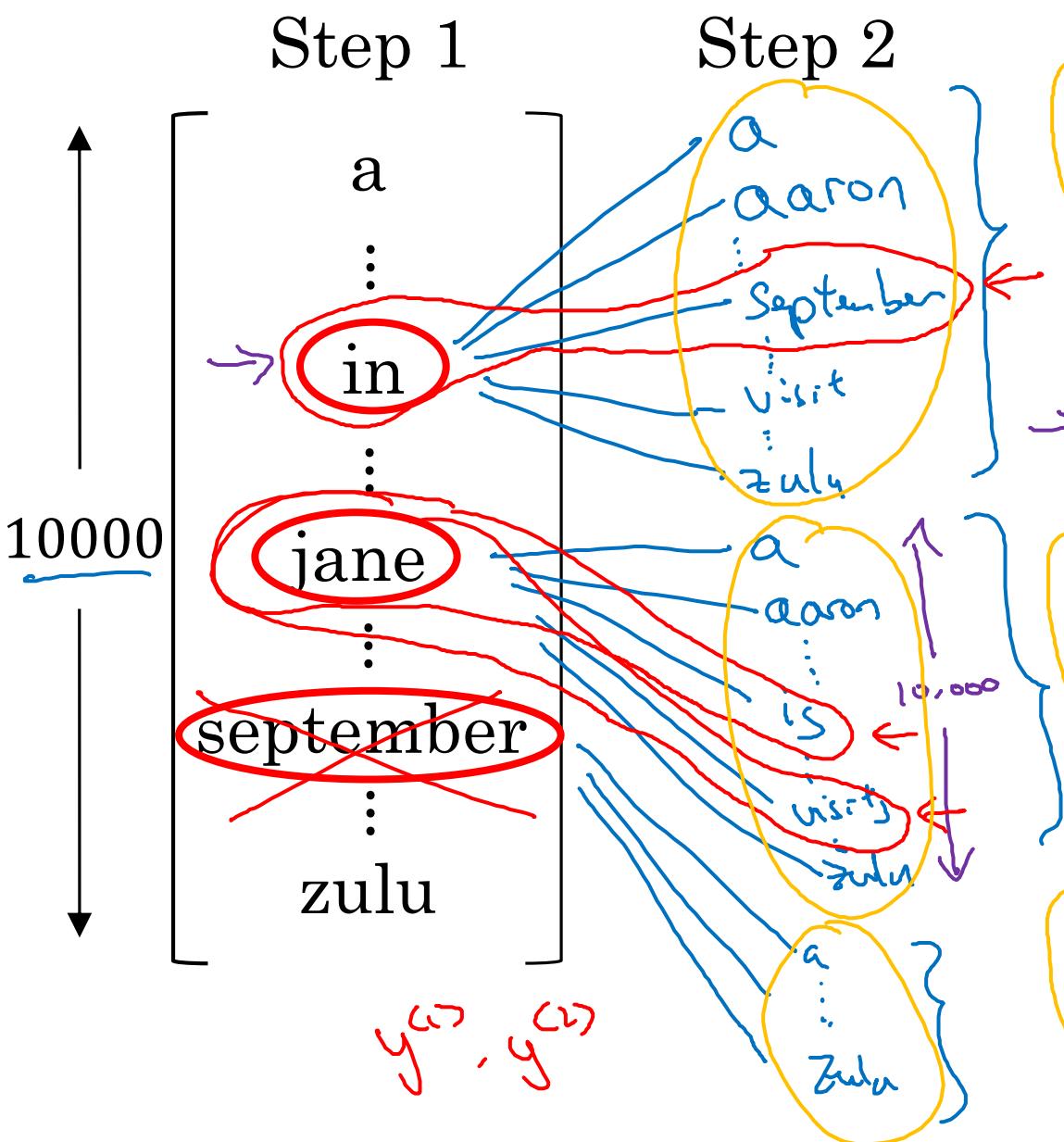
# Beam search algorithm

B = 3 (beam width)

Step 1



# Beam search algorithm



# Beam search ( $B = 3$ )

in september

$a^{<0>} \rightarrow \square \rightarrow \dots \rightarrow \square \rightarrow \hat{y}^{<3>}$

jane is

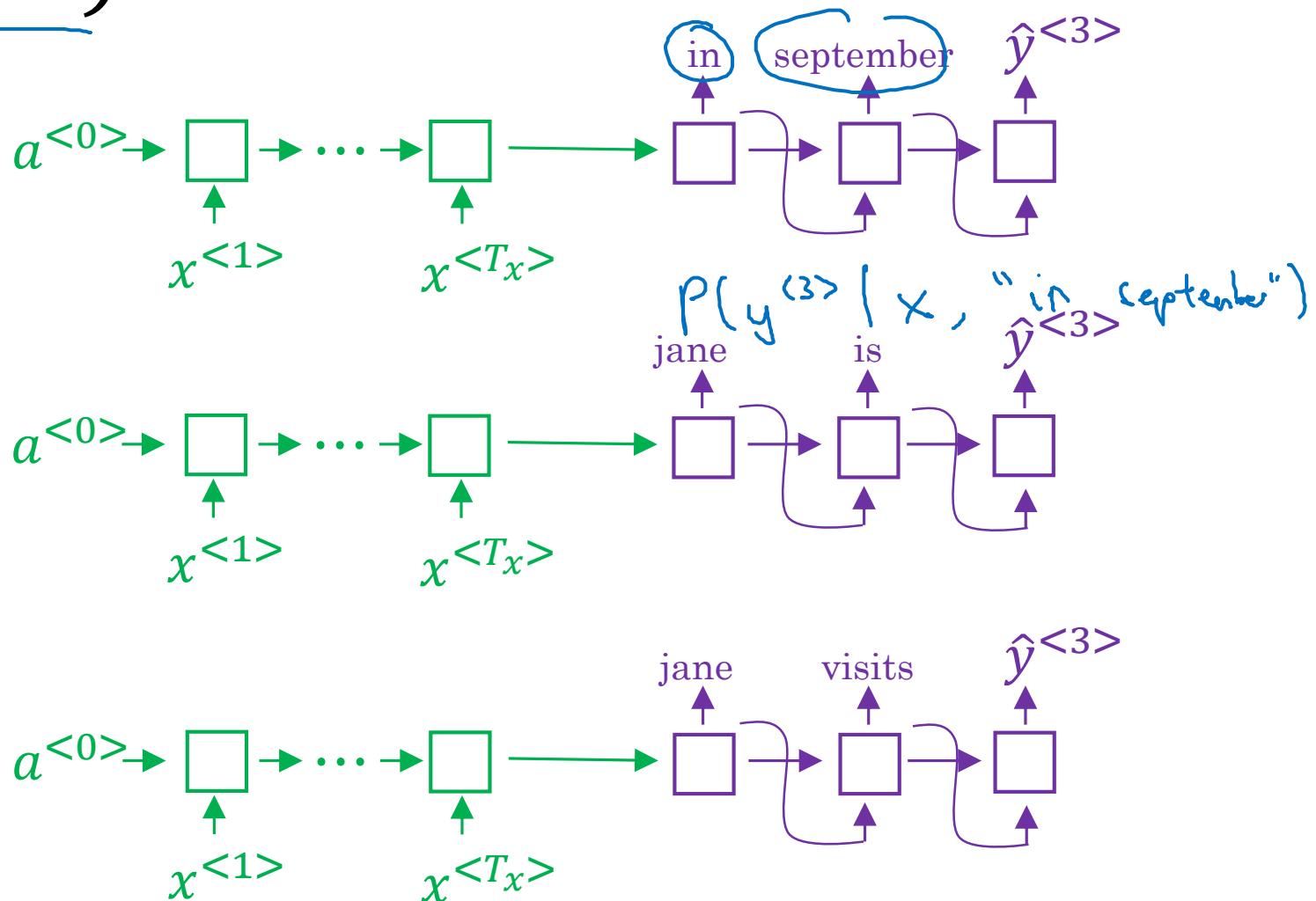
$a^{<0>} \rightarrow \square \rightarrow \dots \rightarrow \square \rightarrow \hat{y}^{<3>}$

jane visits

$a^{<0>} \rightarrow \square \rightarrow \dots \rightarrow \square \rightarrow \hat{y}^{<3>}$

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

$B=1 \rightsquigarrow$  greedy search



jane visits africa in september. <EOS>



deeplearning.ai

# Sequence to sequence models

---

## Refinements to beam search

# Length normalization

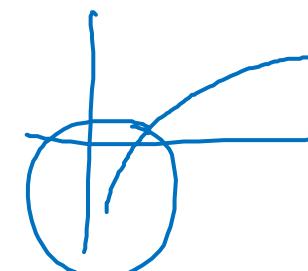
$$\arg \max_y \prod_{t=1}^{T_y} P(y^{<t>} | x, y^{<1>}, \dots, y^{<t-1>})$$

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

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

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

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



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

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

$$\alpha = 0.7$$

$$\begin{cases} d = 1 \\ d = 0 \end{cases}$$

# Beam search discussion

Beam width B?

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

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

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

$y$



deeplearning.ai

# Sequence to sequence models

---

## Error analysis on beam search

# Example

Jane visite l'Afrique en septembre.

→ RNN

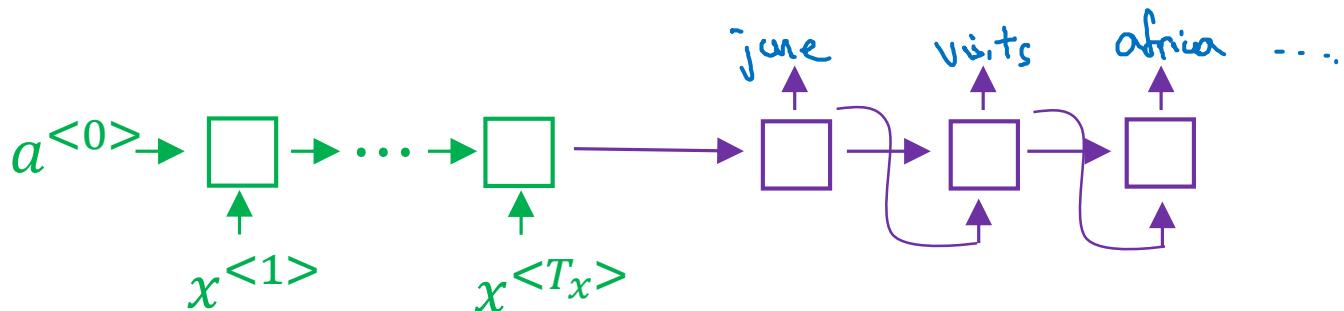
→ Beam Search

B↑

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

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

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



# Error analysis on beam search

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

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

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

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

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

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

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

Conclusion: Beam search is at fault.

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

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

Conclusion: RNN model is at fault.

# Error analysis process

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

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



deeplearning.ai

# Sequence to sequence models

---

Bleu score  
(optional)

# Evaluating machine translation

French: Le chat est sur le tapis.

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.

Precision:

Modified precision:

Bleu  
bilingual evaluation understudy

# 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 <sub>clip</sub>	
the cat	2 ←	1 ←	
cat the	1 ←	0	
cat on	1 ←	1 ←	
on the	1 ←	1 ←	
the mat	1 ←	1 ←	

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

# Bleu score on unigrams

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

$$P_1, P_2, = 1.0$$

Reference 2: There is a cat on the mat.

→ MT output: The cat the cat on the mat. (↑)

$$P_1 = \frac{\sum_{\text{unigrams} \in \hat{y}} \text{count}_{clip}(\text{unigram})}{\sum_{\text{unigrams} \in \hat{y}} \text{count}(\text{unigram})}$$

↑  
Unigram

Count (unigram) Count (unigram)

$\sum_{\text{unigrams} \in \hat{y}}$   $\text{count}_{clip}(\text{unigram})$

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

↑  
 $n\text{-gram}$

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

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

# 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

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





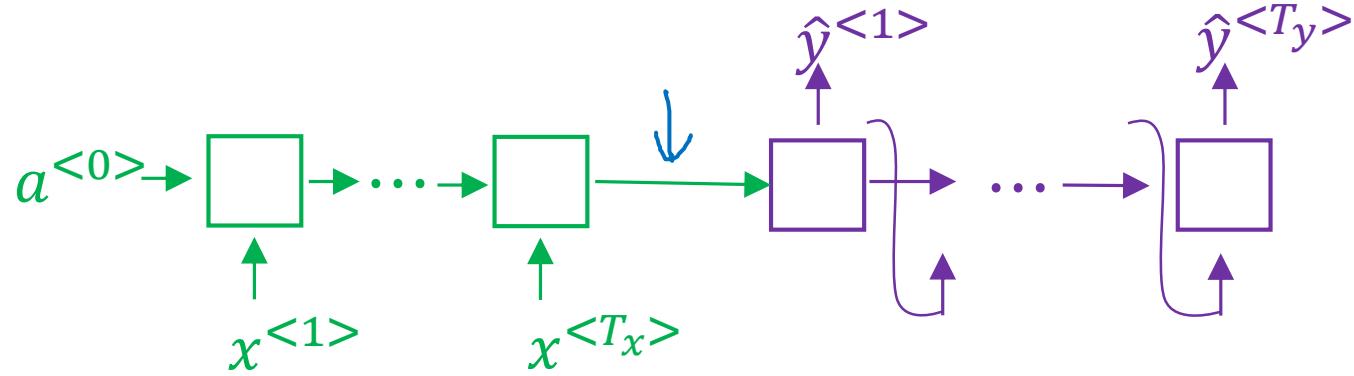
deeplearning.ai

# Sequence to sequence models

---

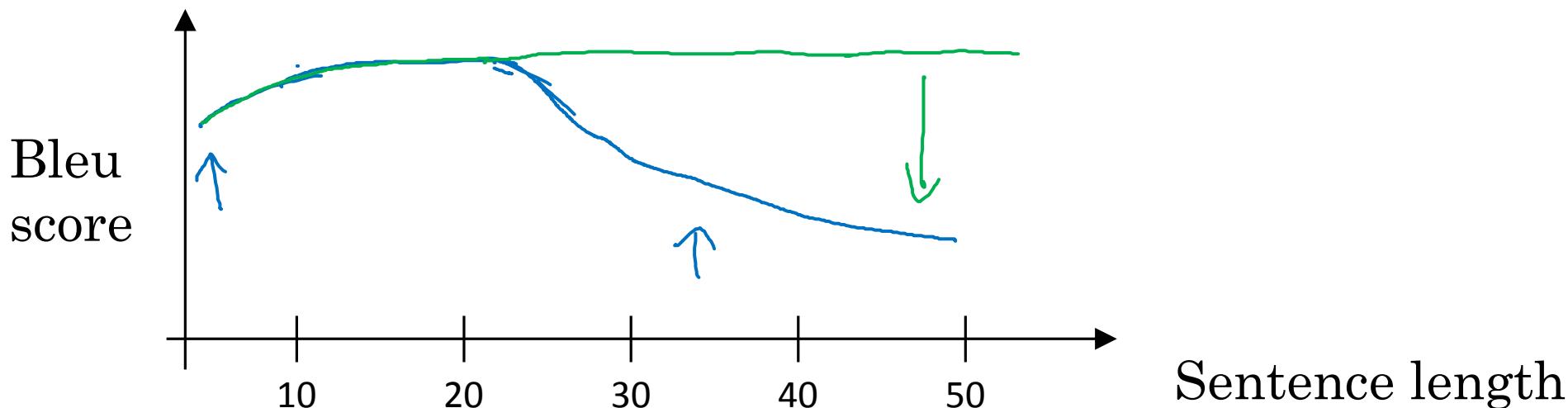
## Attention model intuition

# The problem of long sequences

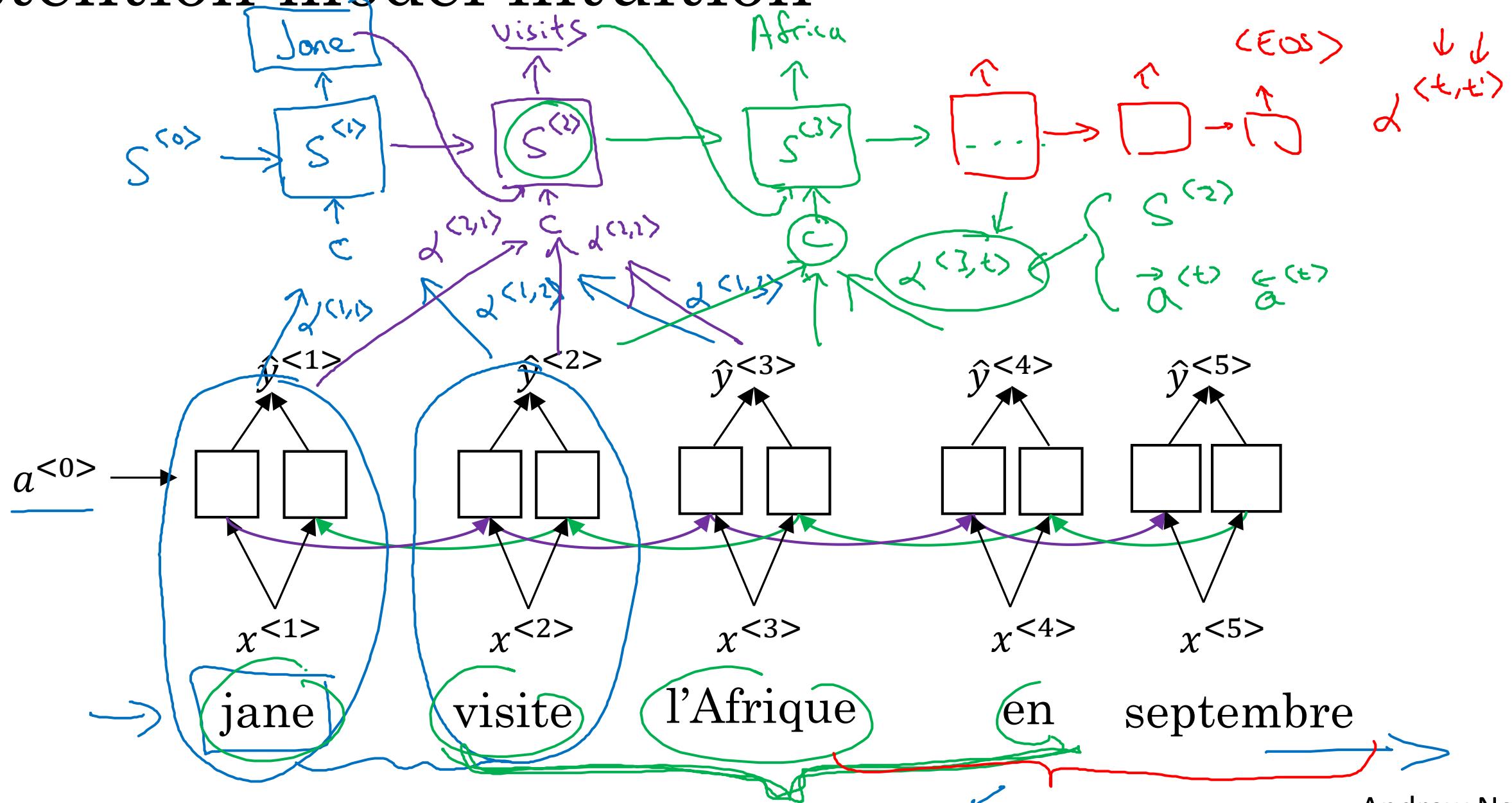


Jane s'est rendue en Afrique en septembre dernier, a apprécié la culture et a rencontré beaucoup de gens merveilleux; elle est revenue en parlant comment son voyage était merveilleux, et elle me tente d'y aller aussi.

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



# Attention model intuition





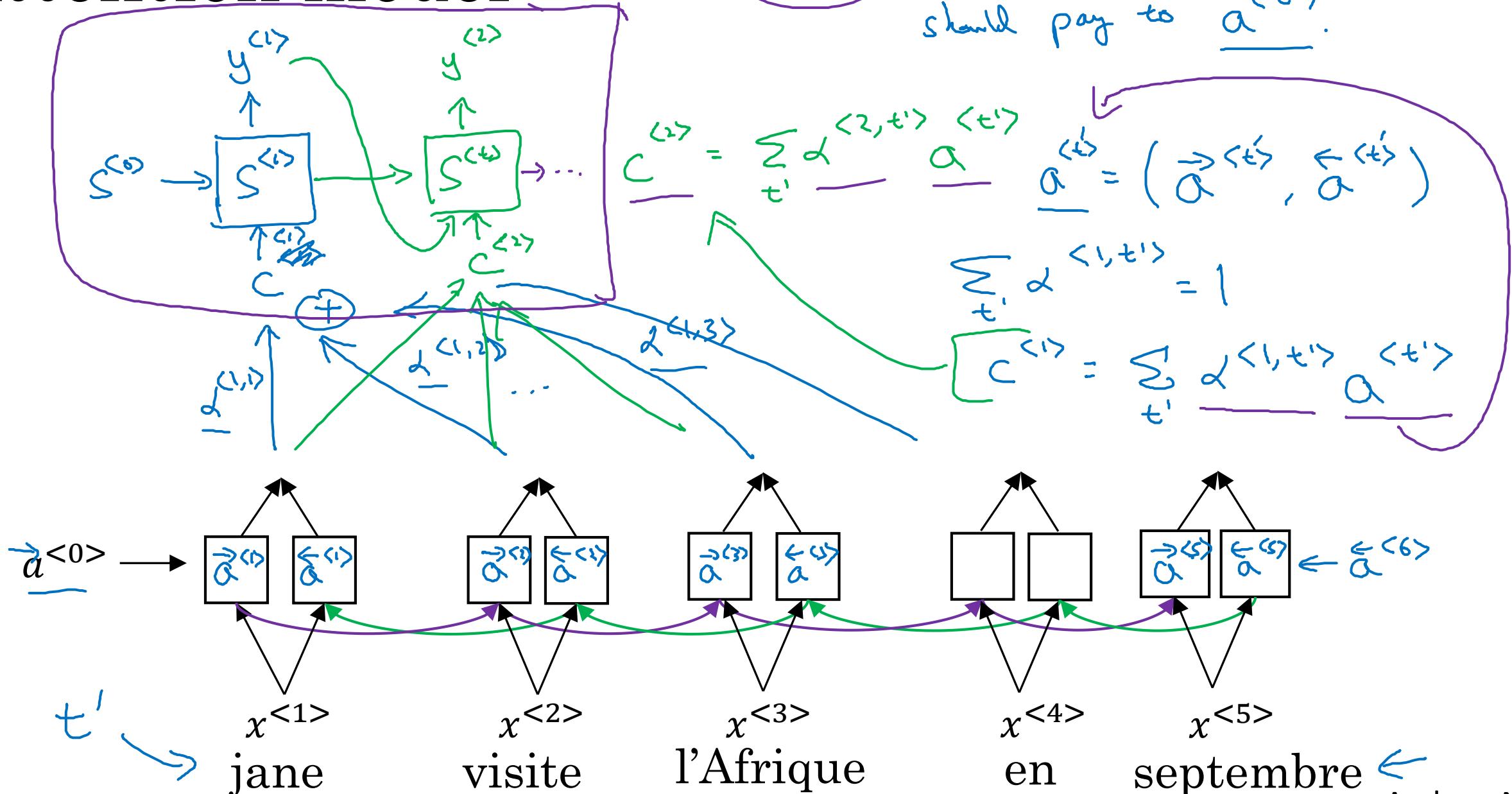
deeplearning.ai

# Sequence to sequence models

---

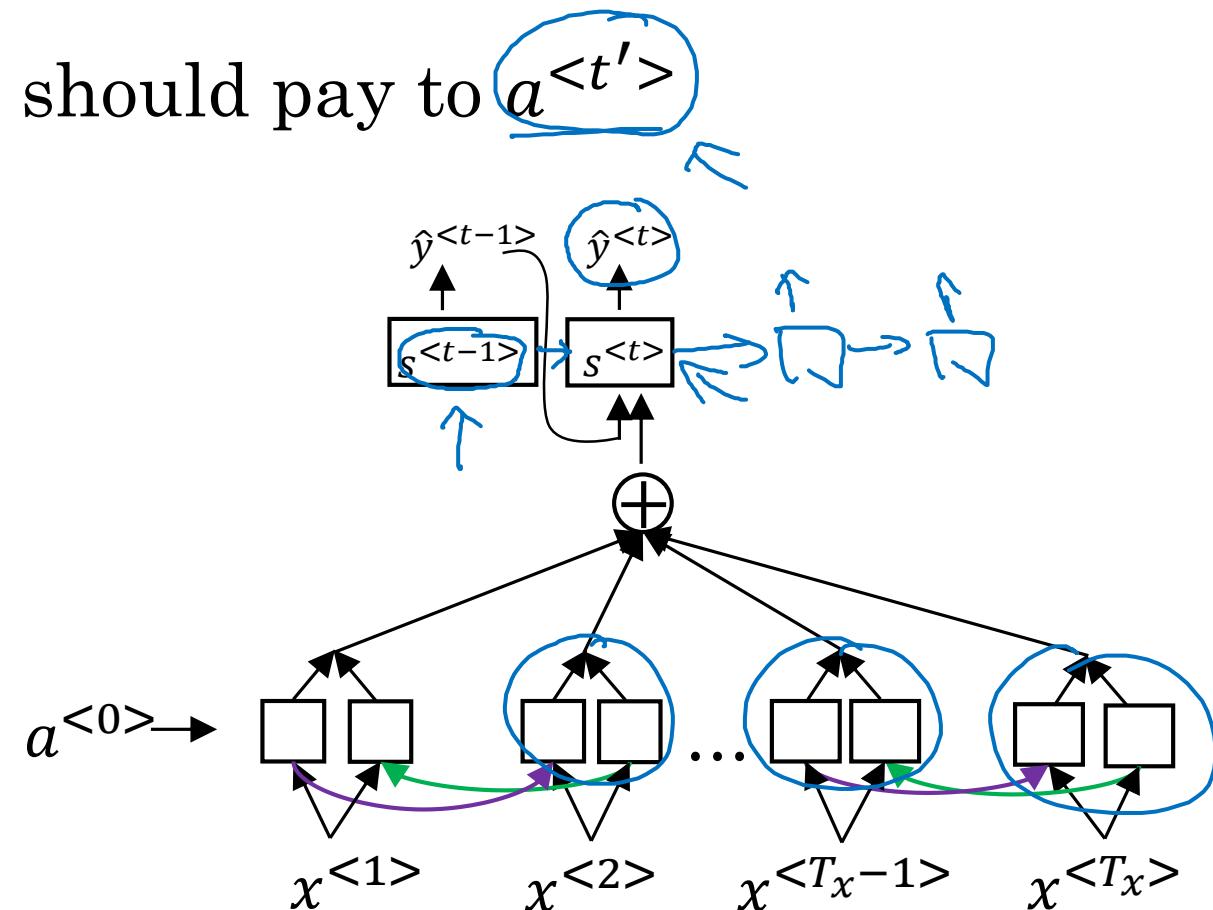
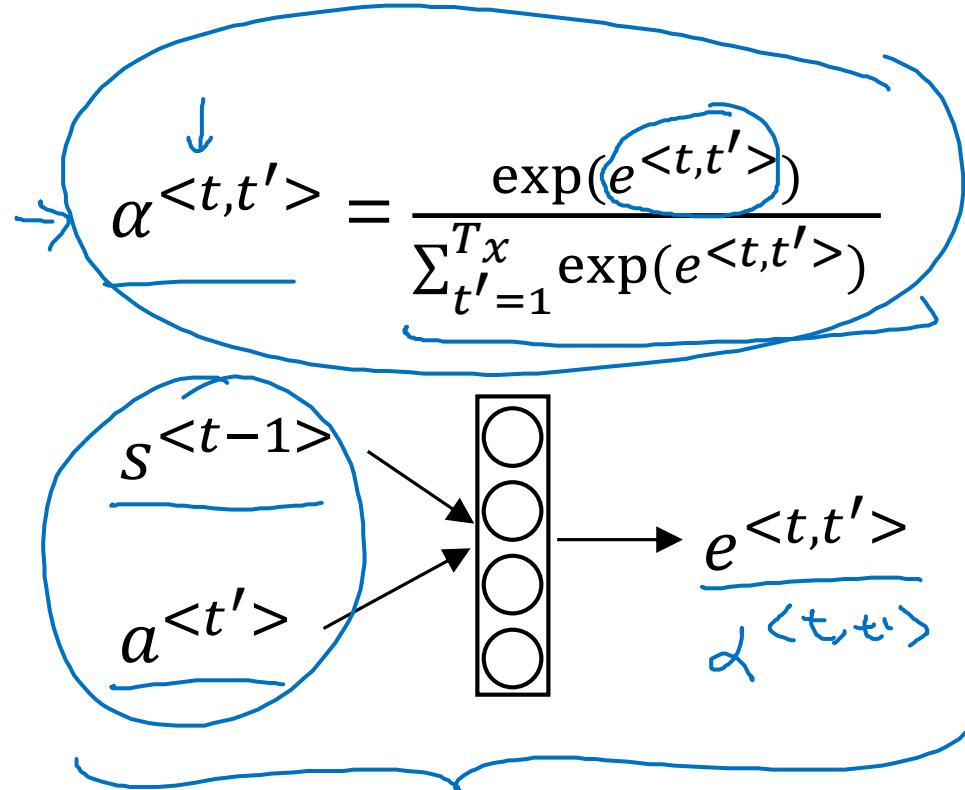
## Attention model

# Attention model



# Computing attention $\underline{\alpha^{'}}$

$\alpha^{'}$  = amount of attention  $y^{'}$  should pay to  $a^{'}$



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

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

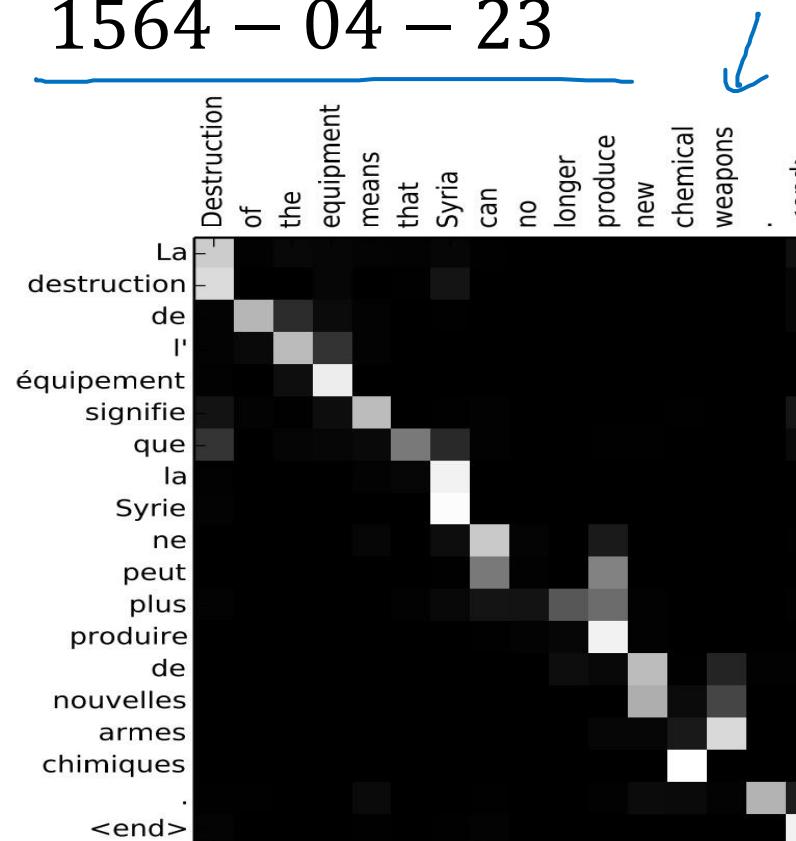
Andrew Ng

# Attention examples

July 20th 1969 → 1969 – 07 – 20

23 April, 1564 → 1564 – 04 – 23

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





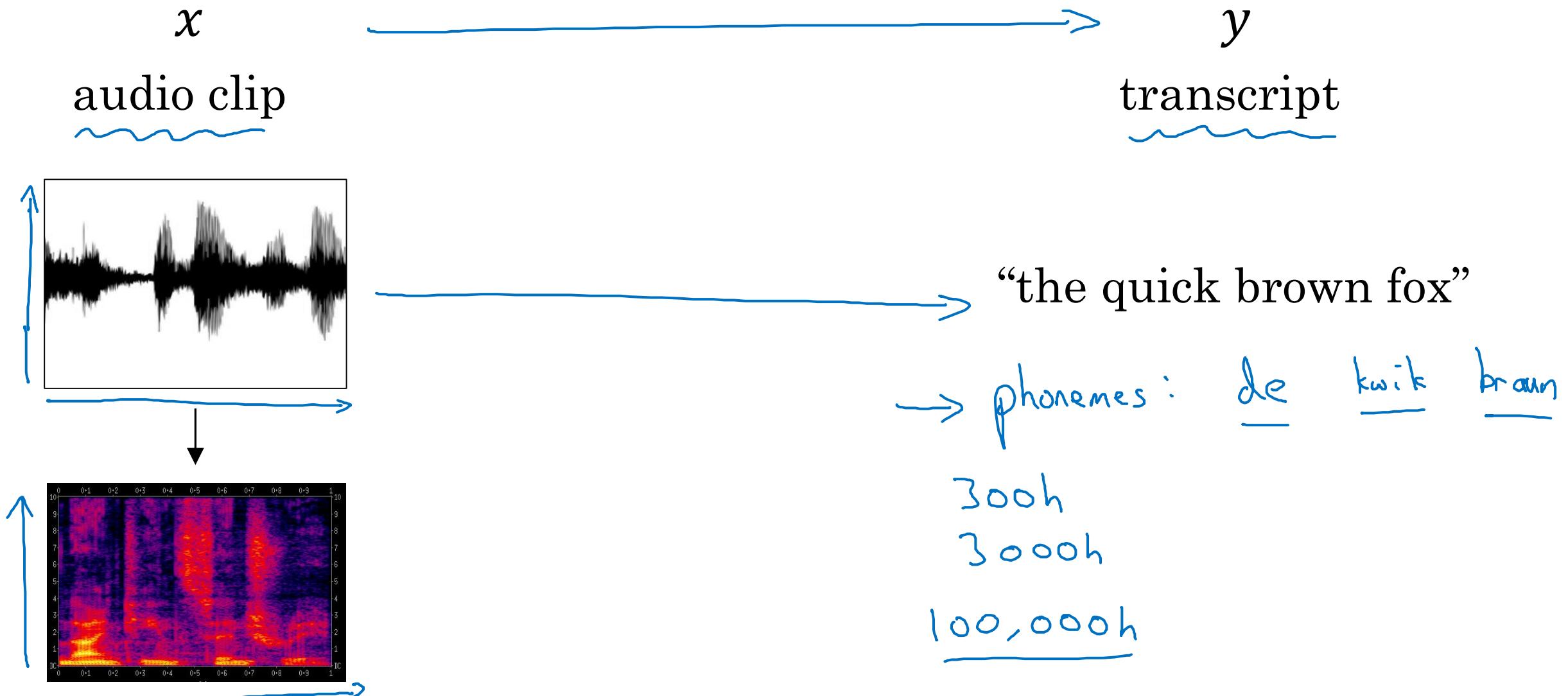
deeplearning.ai

Audio data

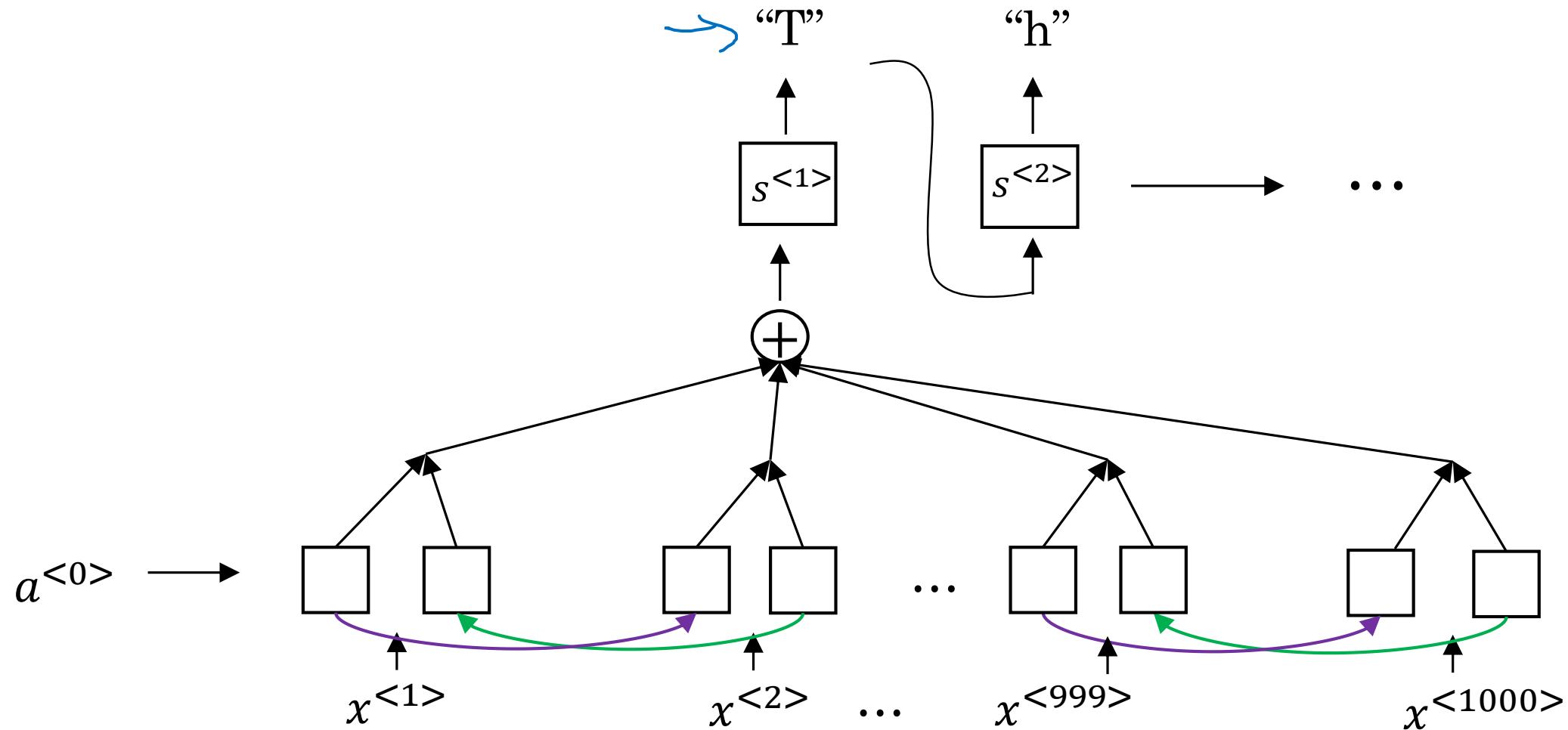
---

Speech recognition

# Speech recognition problem

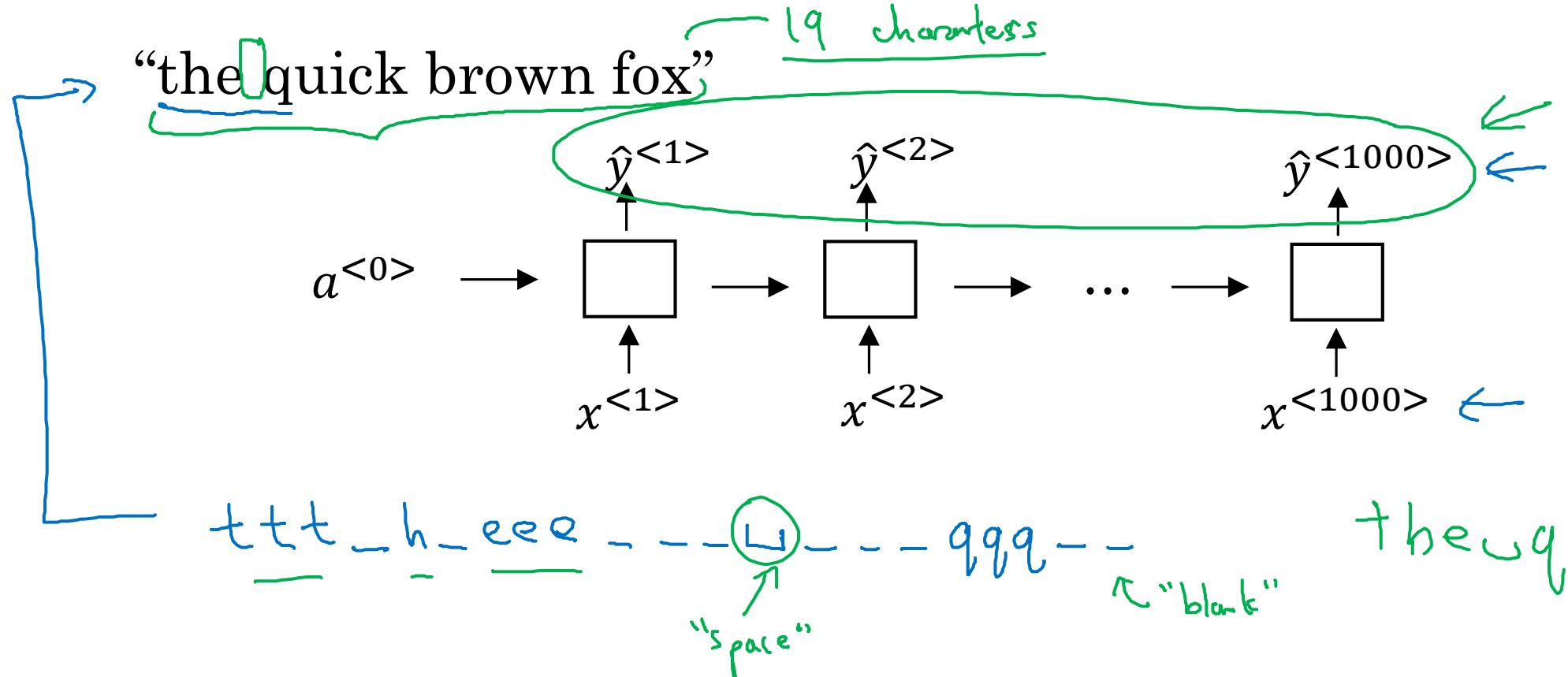


# Attention model for speech recognition



# CTC cost for speech recognition

(Connectionist temporal classification)



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



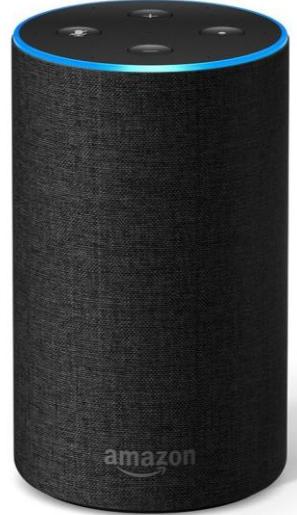
deeplearning.ai

Audio data

---

Trigger word  
detection

# What is trigger word detection?



Amazon Echo  
(Alexa)



Baidu DuerOS  
(xiaodunihao)

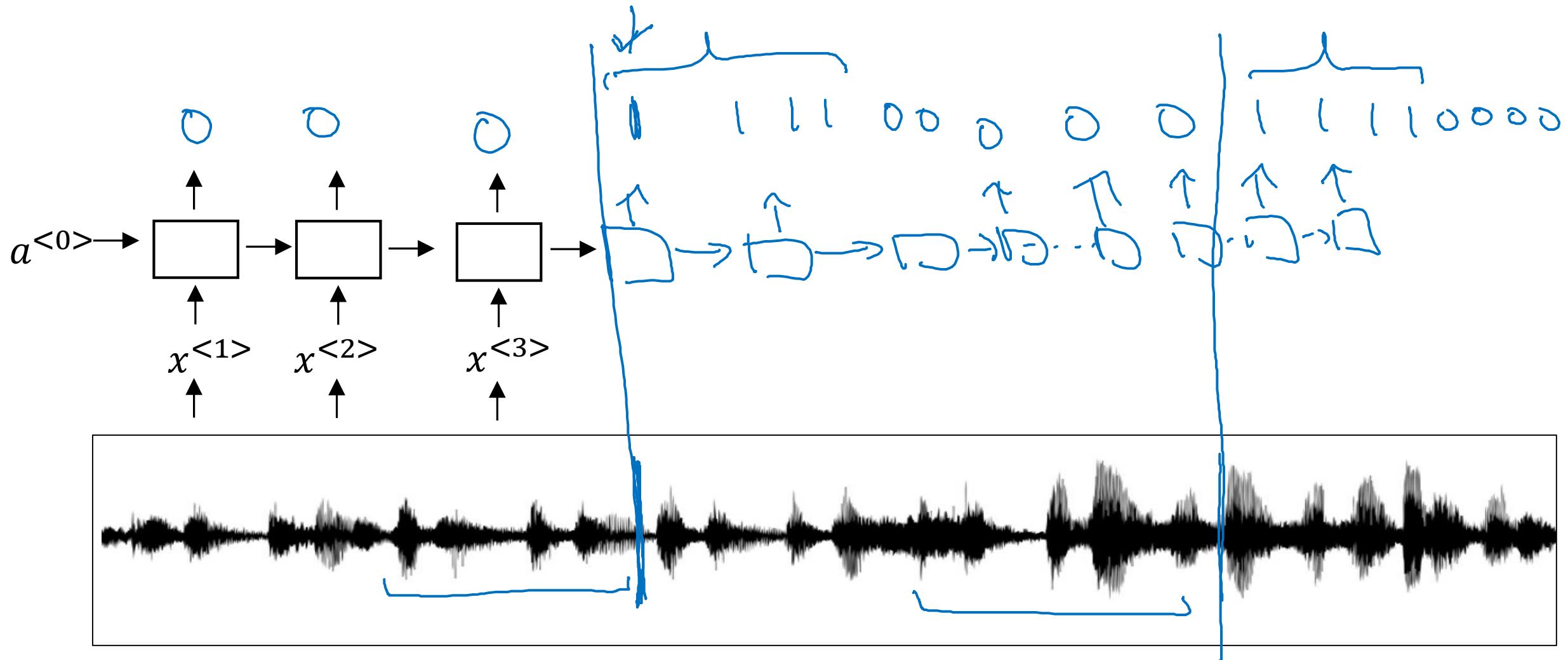


Apple Siri  
(Hey Siri)



Google Home  
(Okay Google)

# Trigger word detection algorithm





deeplearning.ai

# Conclusion

---

## Summary and thank you

# Specialization outline

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

# Deep learning is a super power

Please buy this  
from shutterstock  
and replace in  
final video.



Thank you.

- Andrew Ng