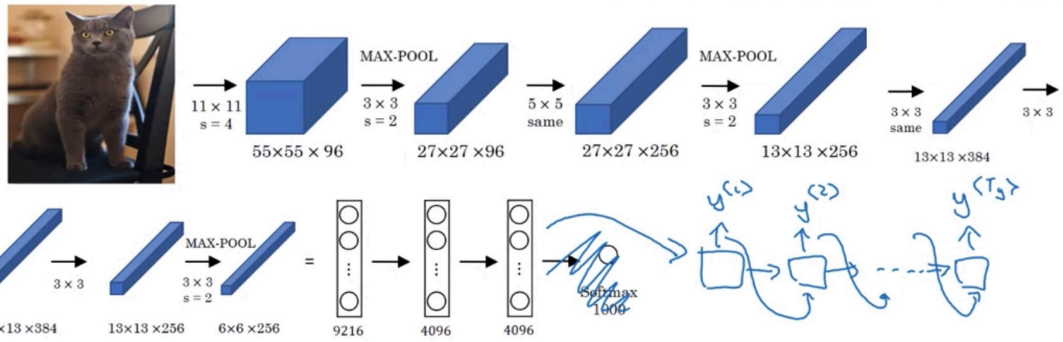
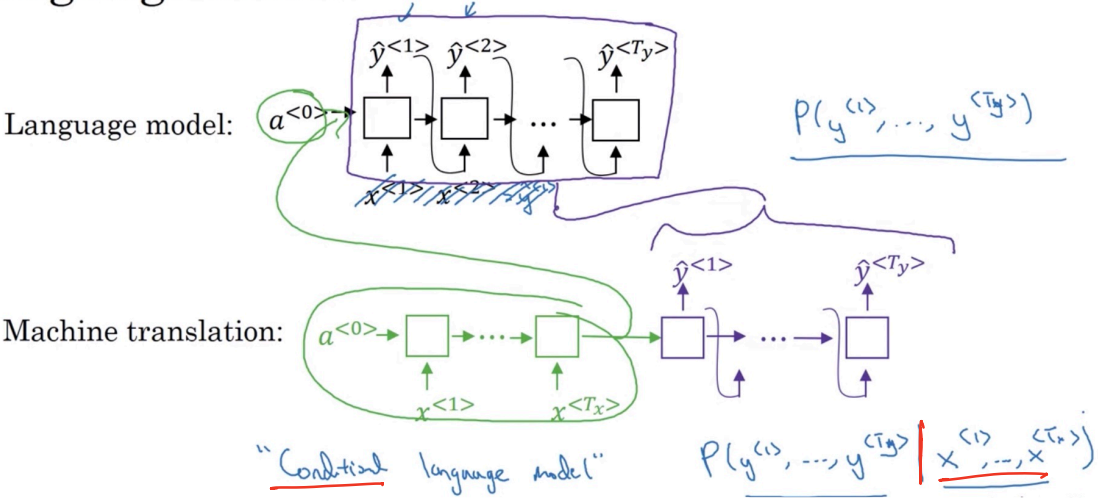
Image captioning

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



Machine translation as building a conditional language model



Why not greedy search? i.e. pick one optimal word at a time.

Consider: → Jane is visiting Africa in Sept.
→ Jane is going to be visiting Africa in Sept.

more comun word, more likely chosen.

first sentence is optimal as it's less verbose.

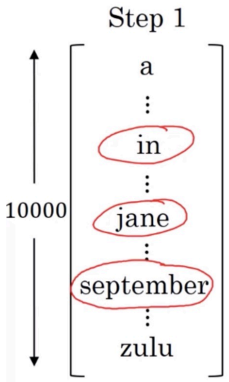
but $P(\text{Jane is going} | x) > P(\text{Jane is visiting} | x)$

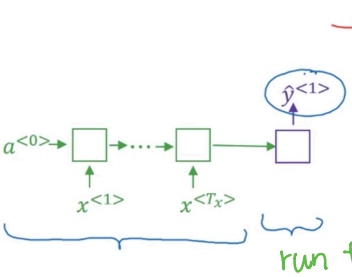
therefore picked 2nd sentence.

Brutal Search : 10000^{10} possibility

Solution : Approximate Search Algorithm.

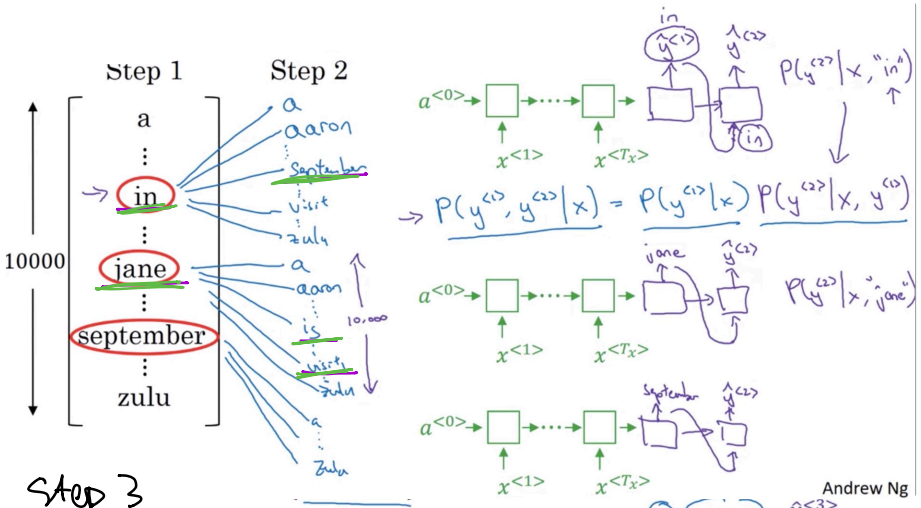
Beam Search



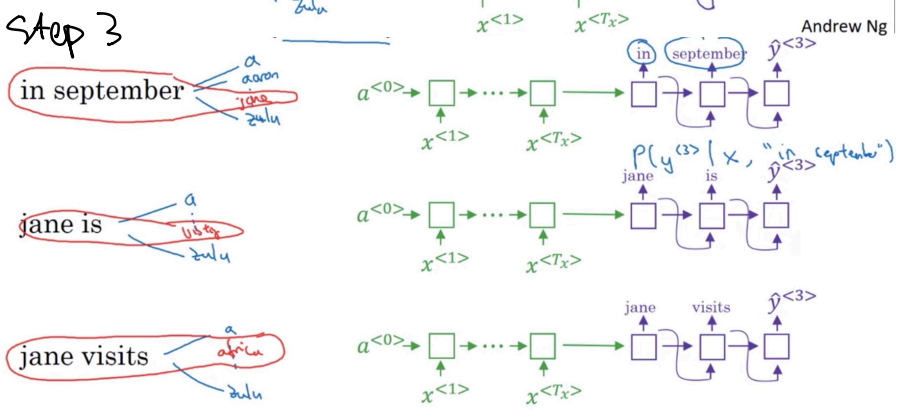


Beam Width = 3

run through 1st network → generate probability for 10,000 → keep top 3



3 copies of the same network



Andrew Ng

$P(y^{<1>}, y^{<2>} \mid x)$ jane visits africa in september. <EOS>

Length normalization

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

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

log

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

$\log P(y|x)$ ←

$P(y|x)$ ←

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

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

compute the score of searches

The larger the beam width B, the more computationally expensive.

Beam Search does not guarantee optimum like a BFS / DFS.

Error Analysis.

eg: ✓ y^* Human: Jane visited Africa in Sept.

× $\hat{y}$ machine: Jane visited Africa last Sept.

→ RNN ? ⇒ $P(y|x)$ [test if $P(y^*|x) \geq P(\hat{y}|x)$]

→ Beam ? ⇒ $\arg\max_y P(y|x)$

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

Beam choose $\hat{y}$. but y^* attains higher $P(y|x)$

→ Beam Search at fault.

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

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

→ RNN 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

Given a French sentence, there could be a few version that is equally as good. How do we evaluate?

Bleu : Bilingual evaluate understanding

Evaluating machine translation (unigram)

French: Le chat est sur le tapis.

→ Reference 1: The cat is on the mat. ← 2 appearance

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

→ MT output: the the the the the the the.

Precision: $\frac{7}{7}$

Modified precision: $\frac{2}{7}$ ← Count_{clip}("the") / Count("the")

Bleu
bilingual evaluation understudy

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

unigram

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

Andrew Ng

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

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

4 / 6

Bleu details

p_n = Bleu score on n-grams only $\quad p_1, p_2, p_3, p_4$

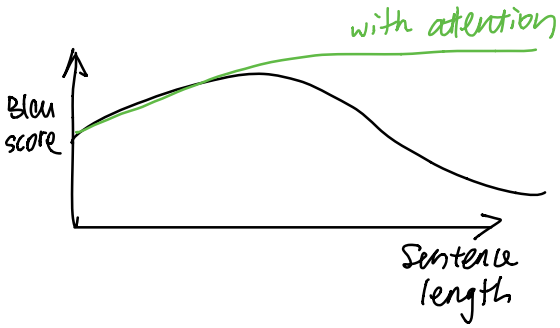
Combined Bleu score: $BP \exp\left(\frac{1}{4}\sum_{n=1}^{4} p_n\right)$

BP = brevity penalty

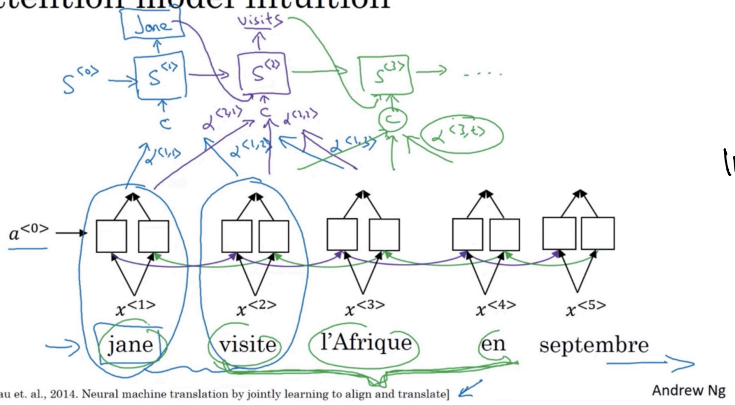
$$BP = \begin{cases} 1 & \text{if MT\_output\_length} > \text{reference\_output\_length} \\ \exp(1 - \text{MT\_output\_length}/\text{reference\_output\_length}) & \text{otherwise} \end{cases}$$

Attention Model

The problem of long sequence:



Attention model intuition



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

Andrew Ng

Intuition

S - hidden states of another RNN (for English translation)

α - attention weights

Attention Model

$a^{<t>} = (\vec{a}^{<t>}, \overleftarrow{a}^{<t>})$ - concat forward / backward RNN

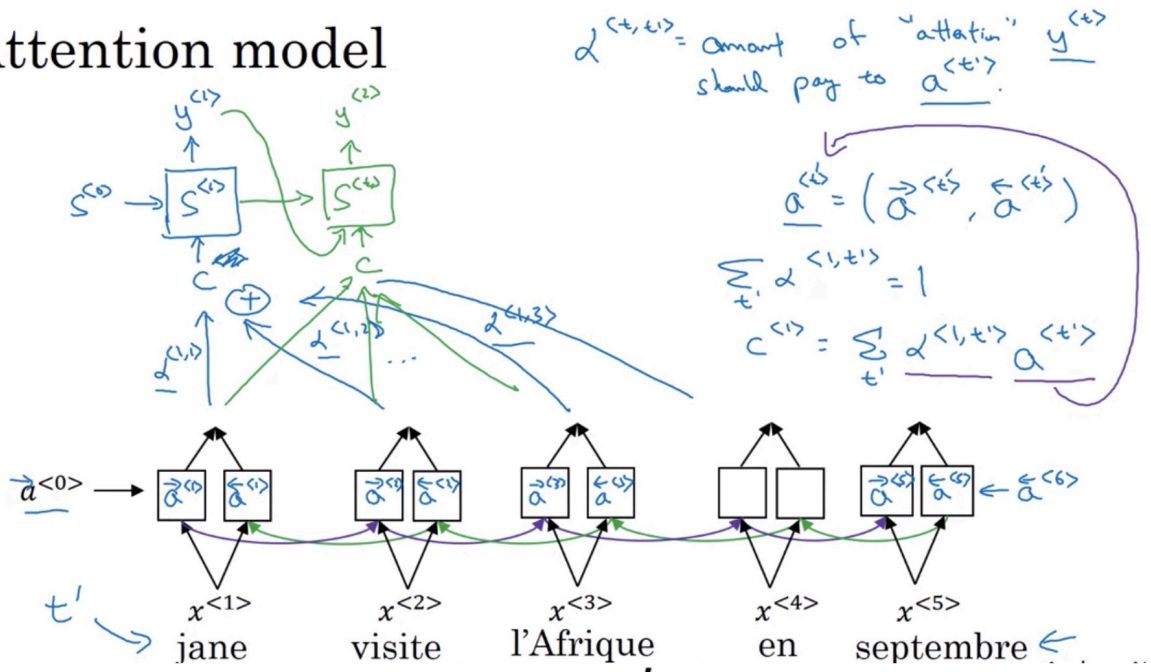
$\alpha^{<1,2>}$ how much the 1st word translated depends on the context 2nd word in original sentence

amount of attention

C = weighted sum of activations

context α $a^{<t>}$

Attention model



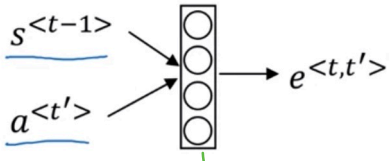
Computing attention $\alpha^{<t,t'>}$

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

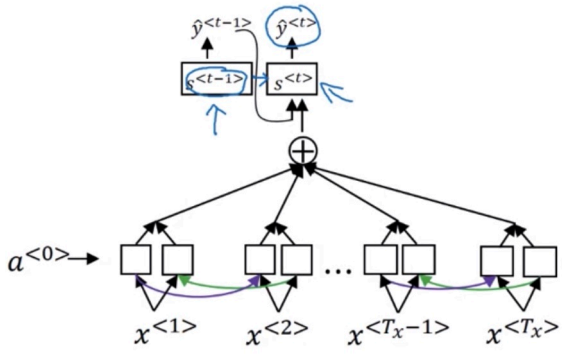
sum to 1

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

softmax



usually simple neural network to combine the two



=> Because we don't know how much should we depend on prev translation/context

Downside : Runs quadratic times $\alpha^{<t,t'>} = O(T_x \times T_y)$

input output

Attention can apply to image captions as well.

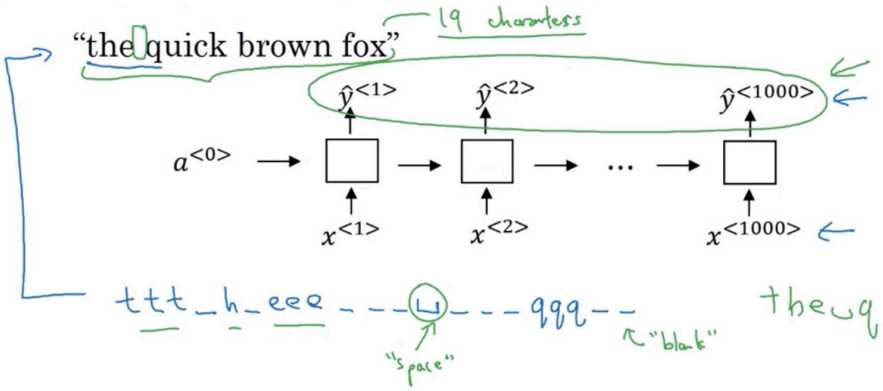
Speech Recognition

x ⟶ y.

audio clip transcript

Attention

CTC cost (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] An

Trigger Words