C5 W4

Basic model:-

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

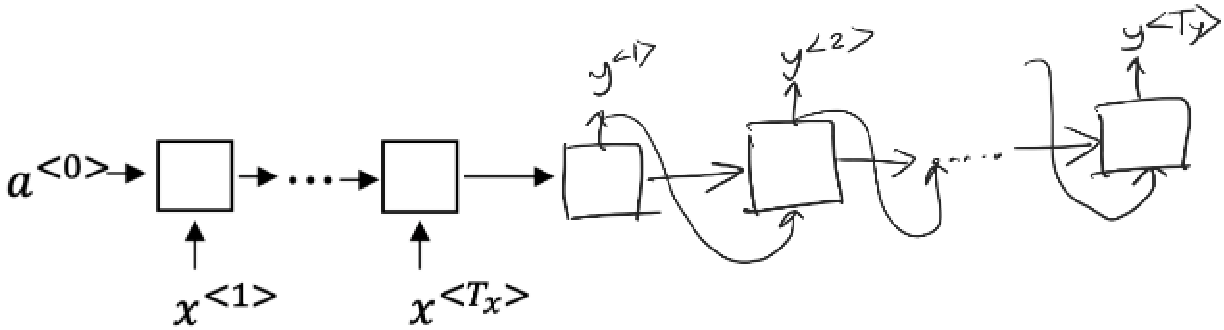
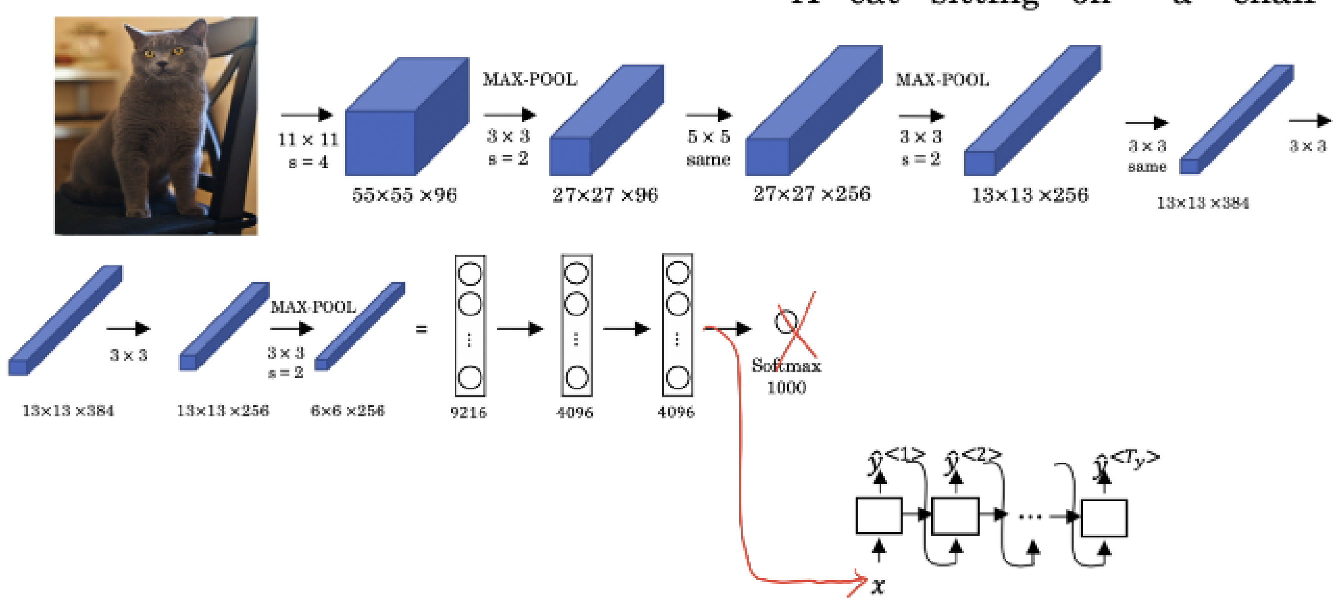


Image captioning

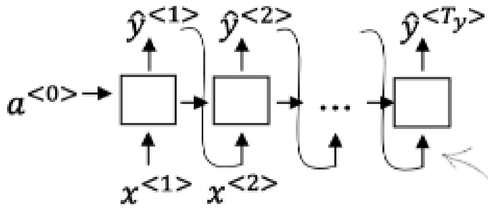
$y^{<1>}$ $y^{<2>}$ $y^{<3>}$ $y^{<4>}$ $y^{<5>}$ $y^{<6>}$
A cat sitting on a chair



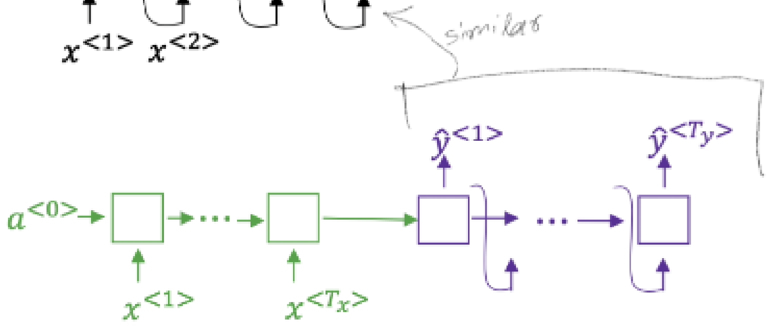
Machine translation as building a conditional language model

estimate the probability of a sentence

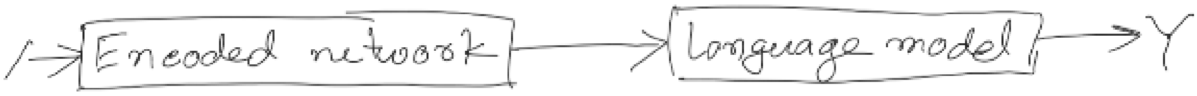
Language model:



Machine translation:



Machine translation is pretty much similar to the language model. Only difference is instead of starting the network to all zero, it has an encoded network that figure out some representation of the input sentence. And it use this representation as an input of the decoder network (which is identical to language model.)



machine translation

Machine translation provides the probability of different sentence being the translation of input sentence. This is why it is called conditional language model.

Finding the most likely translation

Jane visite l'Afrique en septembre.

English French

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

possible translation.

→ 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.

Machine translation will provide the probability of each sentence

Our goal → $\arg\max_{y^{<1>},\dots,y^{<T_y>}} P(y^{<1>},\dots,y^{<T_y>} \mid x)$ ← Done by beam search

Why not greedy search?

Greedy search select the word one at a time. For example:-

→ Jane is visiting africa in september.

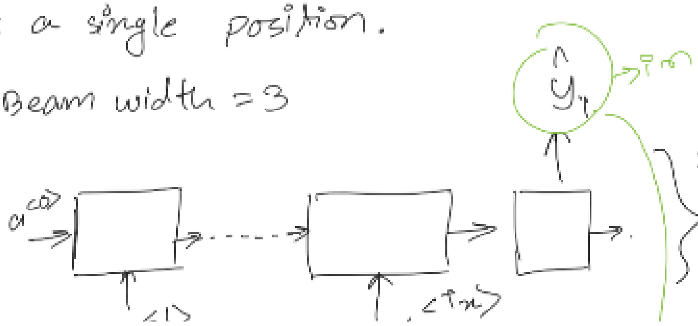
→ Jane is going to be visiting Africa in september.

"going" is more common words than visiting. so greedy search might select the second sentence instead of first one. While the first one is more appropriate.

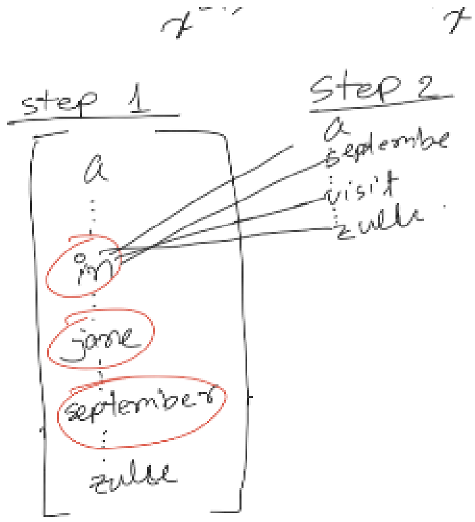
Beam search:-

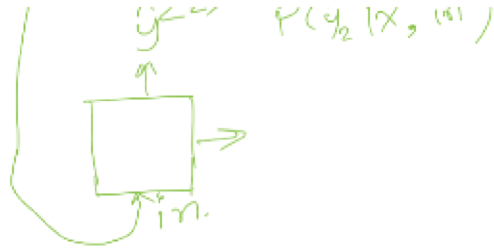
Greedy search just consider one value in each position & move on. Where Beam search consider $\underbrace{\text{multiple}}_{\text{beam width}}$ value for a single position.

Let, Beam width = 3



step 1, beam search remmember 3 most likely words, in this case: in, zulu, september





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

Repeat this for jane & september

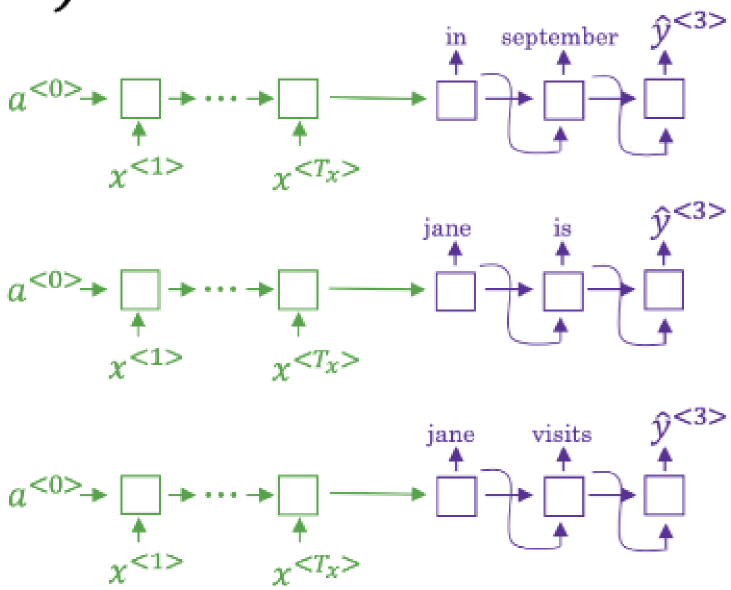
⇒ If our vocabulary size is 10,000 then in the second stage we will calculate for 30,000 possibility

Beam search ($B = 3$)

in september

jane is

jane visits



Refinements to beam search:-

Length normalization:-

$$\arg\max_{y} \prod_{t=1}^{T_y} P(y^{<t>} \mid x, y^{<1>}, \dots, y^{<t-1>}) = P(y^{<1>} \mid x)\, P(y^{<2>} \mid x, y^{<1>})\, P(y^{<3>} \mid x, y^{<1>}, y^{<2>}) \dots$$

log

some of the probability are very tiny so it will make the value very small.

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

$t=1$

this approach prefers shorter sentence than longer sentence. Even if longer sentence is more correct. Since in longer sentence we will have more probability that will be multiplied together. Hence, make the probability even less. to overcome this, we use following eqⁿ:-

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

→ normalize with length. α = hyperparameter.

$\alpha = 1$ = normalize along the length.

$\alpha = 0$ = Don't normalize at all

$\alpha = 0.7$ = works better in practice

Larger beam-width (B) means lot's of computation.

larger B :- Better result., slower, more memory

small B :- worse result, faster, less memory

$B \Rightarrow 10$ is more common in production.

$B \Rightarrow 100$ is considered huge in production.

$B \rightarrow 1000, 10{,}000 \Rightarrow$ used in academia

$B \Rightarrow 1 \rightarrow$ is greedy search.

Beam search is a heuristic search. so it doesn't always give the correct result.

Error analysis in Beam search:-

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

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

Case 1:

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

Conclusion: Beam search is at fault.

Case 2:

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

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

Andrew Ng

Bleu score

If there are multiple sentence that are good translatio

How would we chose between them?

French: Le chat est sur le tapis.

Reference 1: The cat is on the mat. →Both are good translation

Reference 2: There is a cat on the mat.

Belu looks for the human reference for this sentence

MT output: the the the the the the the.

look each of the words and see if they appeared in the references.

Precision: $\frac{7}{7}$ 7 → appeared, 7 → total 7 words (the)

Modified precision: $\frac{2}{7}$ 2 → Count$_{clip}$, 7 → count

But this is not a good translation still we are getting precision 1. So we will modified precision, where we will give credit upto the maximum number of times it appeared in the references. "The" appeared 2 times in the reference 1, and 1 times in reference-2. So we will take maximum of this number.

Bleu score on $\boxed{\text{bigrams}}$ → 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.

Bigram	Count	Count_clip
the cat	2	1
cat the	1	0
cat on	1	1
on the	1	1
the mat	1	1
	6	4

modified precision $= \frac{4}{6} = \frac{2}{3}$

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

p_n = Bleu score on n-grams only

Combined Bleu score:

BP → Bravity penalty.

BP = { 1   if MT_output_length > reference_output_length

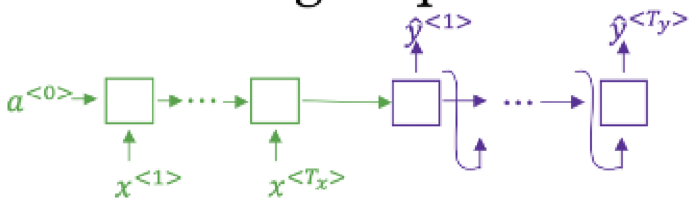
$BP = $ | exp(1 − MT_output_length/reference_output_length) otherwise

refrence_output_length/MT_output_length.

sorter sentence tends to have better precision. so we will penalize the precision if we get sorter sentence.

Attention model:-

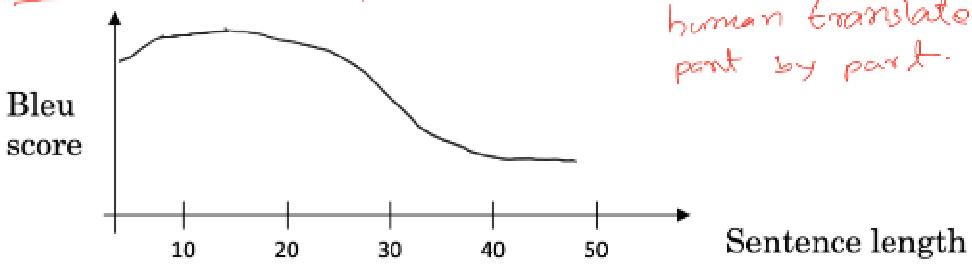
The problem of long sequences



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.

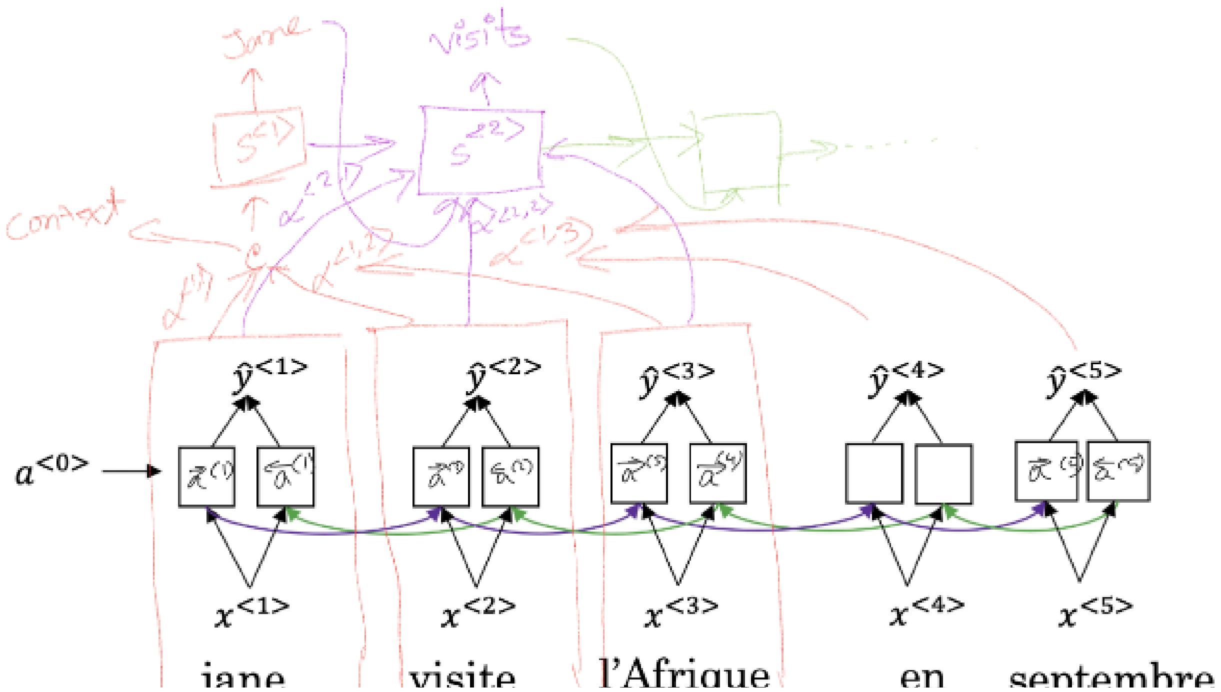
human translate long sentence part by part.



Andrew Ng

very short & very long sentences are harder to train.

which word to pay attention:-



Jane visite l'Afrique en septembre

α = how much we should pay attention to this word =

α = attention weight

$$a^{<t'>} = (\overrightarrow{a}^{<t'>}, \overleftarrow{a}^{<t'>})$$

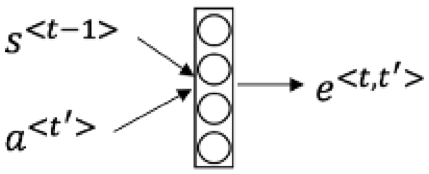
$$\sum_{t'} \alpha^{<1,t'>} = 1.$$

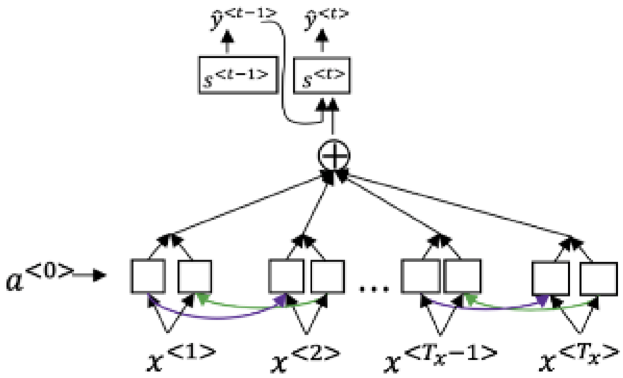
$$c^{<1>} = \sum_{t'} \alpha^{<1,t'>} a^{<t'>}$$

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

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

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





Speech recognition:-