

W1L1: Recurrent Neural Networks: why sequence models?

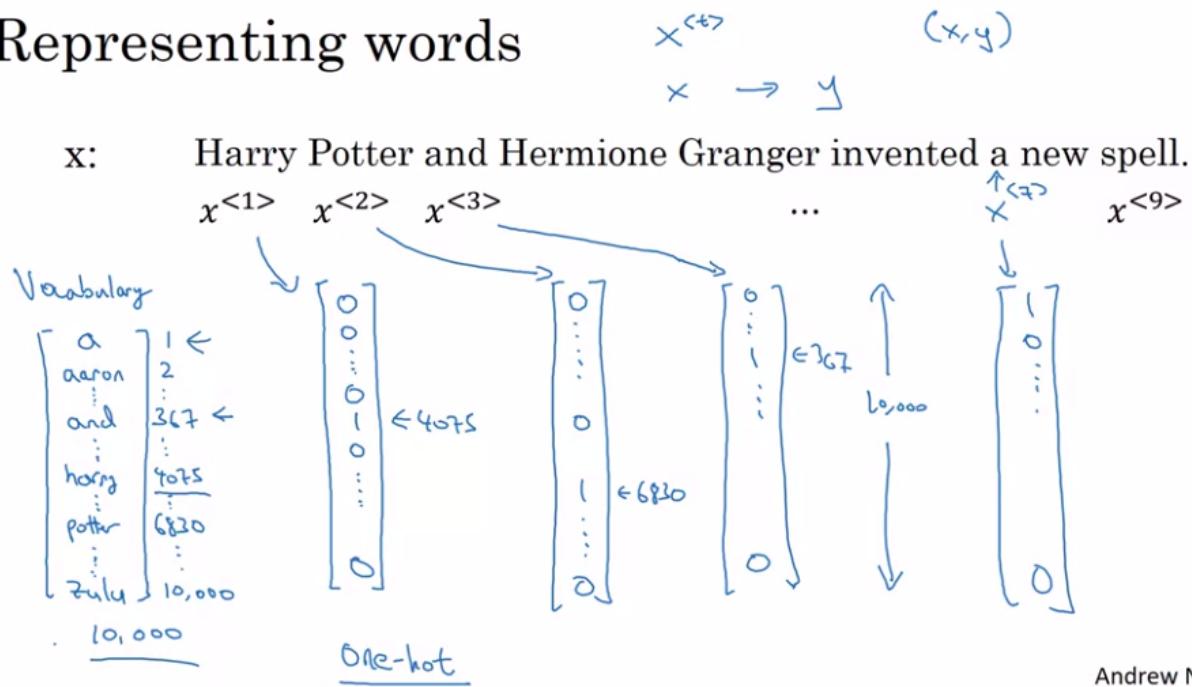
- Examples:

1. Speech Recognition: Speech to Text
2. Music Generation: Empty set or music genre as an input to generate music
3. Sentiment Analysis: textual review to the rating(number of stars out of 5)
4. DNA sequence analysis: finding the particular type of DNA from the given DNA sequence
5. Machine translation: linguistic translations
6. Video activity recognition
7. Name entity recognition: finding people in the sentences.

W1L2: Notations

- $X^{(i)\leftrightarrow}$: i^{th} training sample with the t^{th} sequence element
- $Tx^{(i)}$: input sequence length of the i^{th} training sample. (length can be different for the training examples)
- Similarly for y
- Vocabulary or dictionary in case of words
- One hot representation

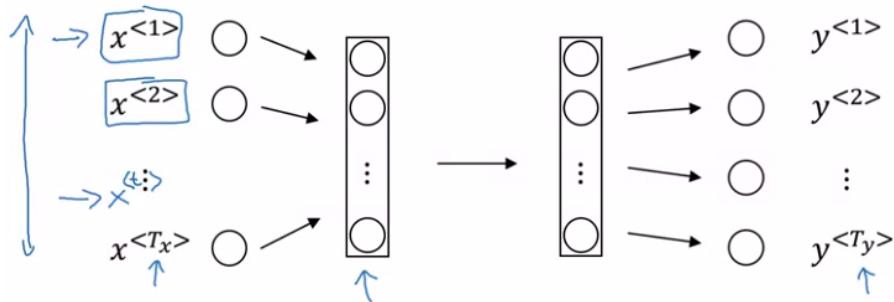
Representing words



Andrew Ng

W1L3: RNN

- Why standard NN would not work in case of sequence models:

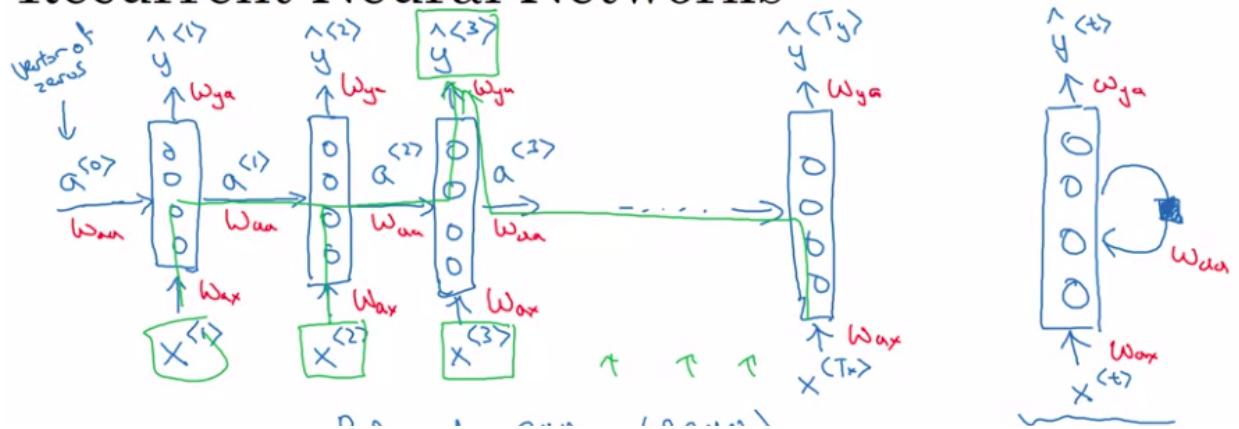


Problems:

- - Inputs, outputs can be different lengths in different examples.
- - Doesn't share features learned across different positions of text.

Andrew Ng

Recurrent Neural Networks



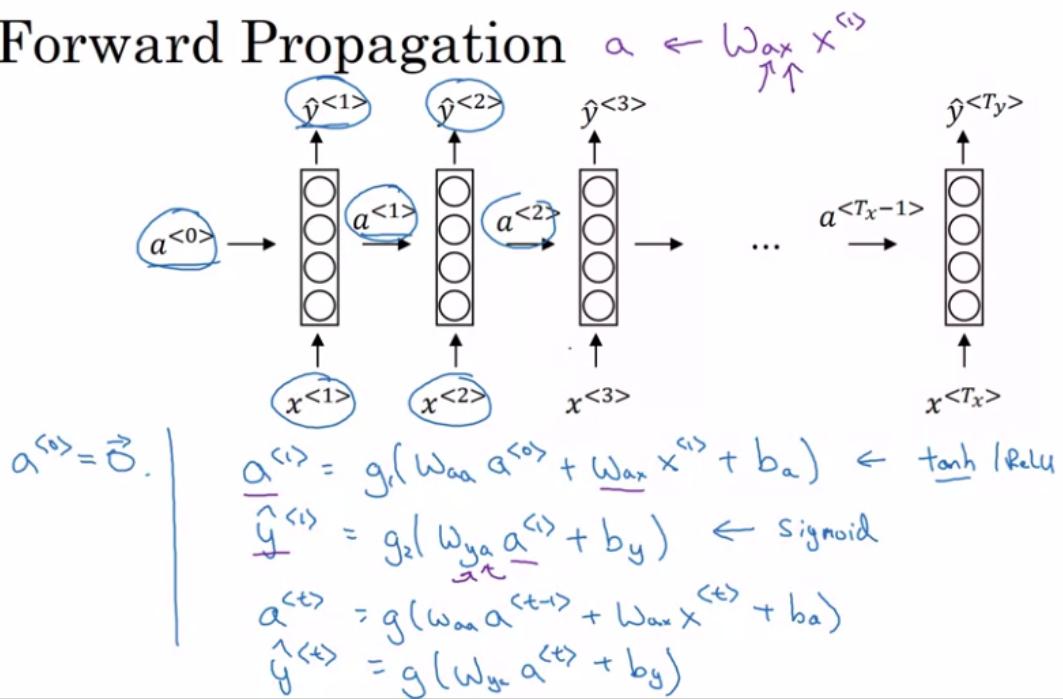
Limitation: only used previous words for the predictions. Due to this, in the second sentence, Teddy will be identified as the name of a person while it's not.

He said, “Teddy Roosevelt was a great President.”

He said, “Teddy bears are on sale!”

- Forward Propagation:

Forward Propagation



- Simplified RNN notations

$$a^{<t>} = g(W_{aa} a^{<t-1>} + W_{ax} x^{<t>} + b_a)$$

$$(100, 100) \quad (100, 10, 000) \quad (10, 000)$$

$$\hat{y}^{<t>} = g(W_{ya} a^{<t>} + b_y)$$

$$\hat{y}^{<t>} = g(W_y a^{<t>} + b_y)$$

$$\begin{bmatrix} W_{aa} & W_{ax} \end{bmatrix} = W_a$$

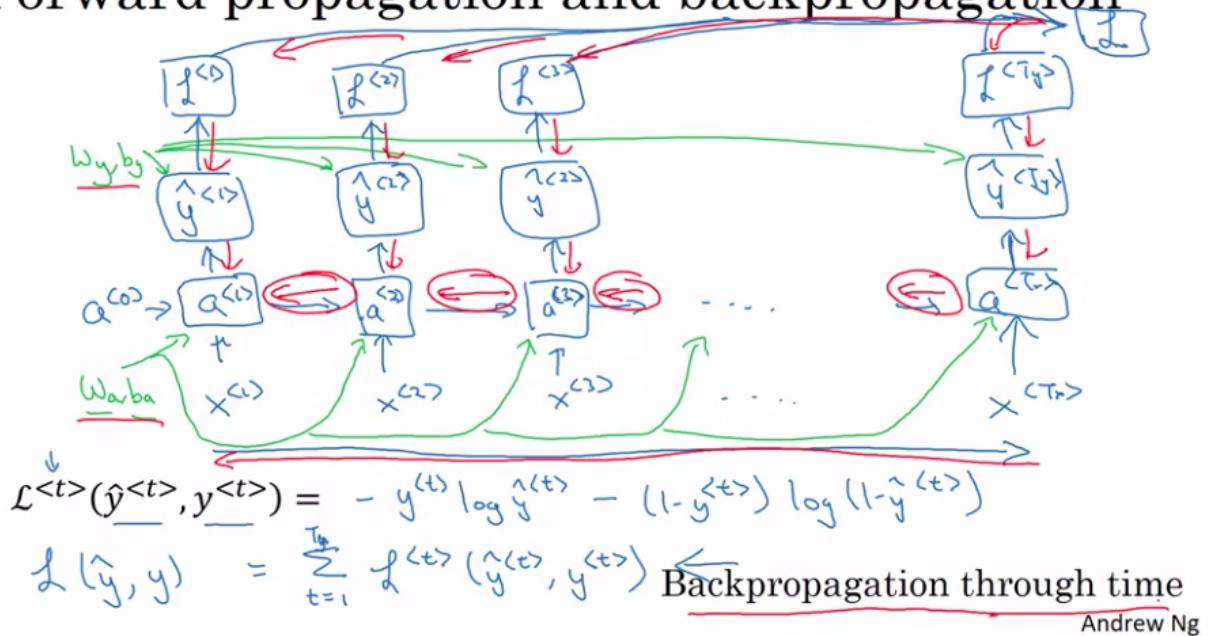
$$(100, 10100)$$

$$[a^{<t-1>}, x^{<t>}] = \begin{bmatrix} a^{<t-1>} \\ x^{<t>} \end{bmatrix}$$

$$\begin{bmatrix} W_{aa} & W_{ax} \end{bmatrix} \begin{bmatrix} a^{<t-1>} \\ x^{<t>} \end{bmatrix} = W_{aa} a^{<t-1>} + W_{ax} x^{<t>}$$

W1L4: Backpropagation: similar to Backprop in NN

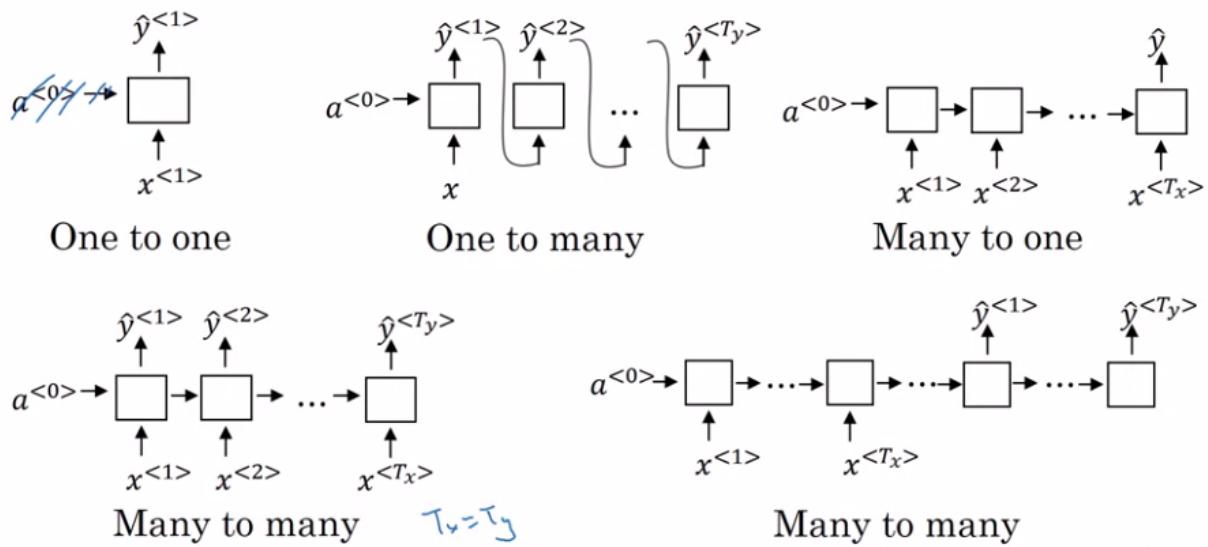
Forward propagation and backpropagation



W1L5: types of RNNs

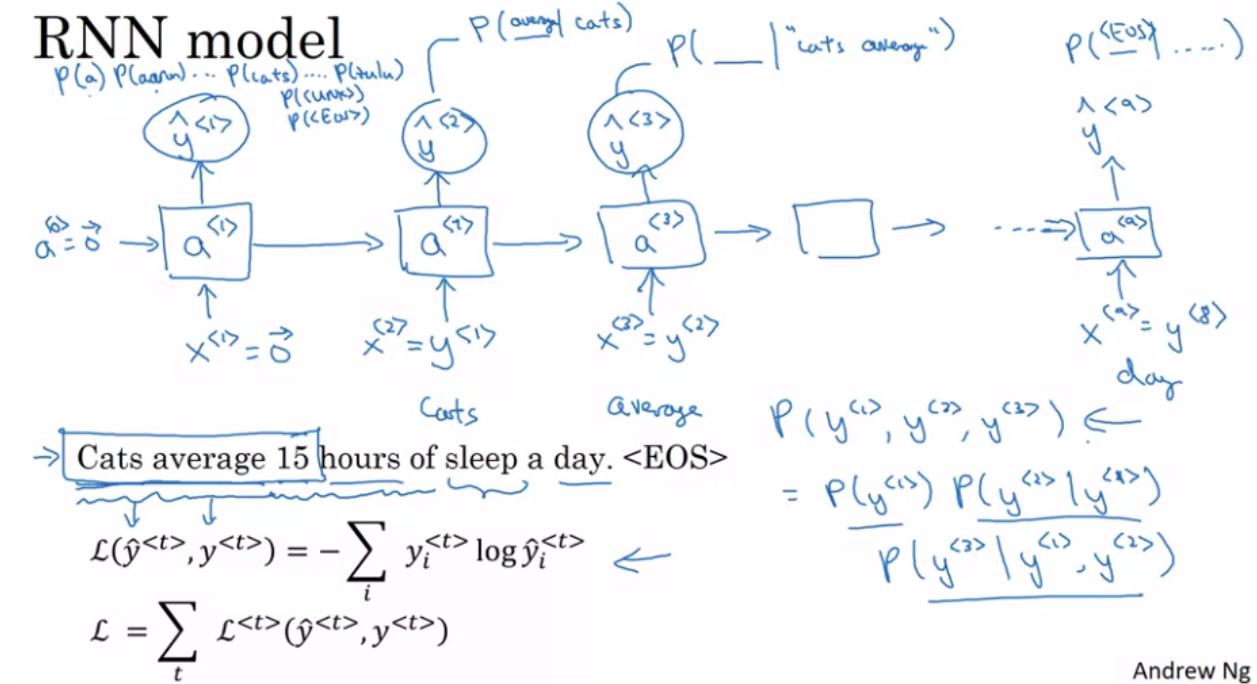
- One to one: $T_x = T_y$
- One to many: music generation (T_x is the empty set)
- Many to many: machine translation (T_x not equal to T_y)
- Many to one: movie review/ sentiment classification

Summary of RNN types



W1L6: Language models and sequence generation

- Predicting the probability of occurrence of the next words in the sentence.



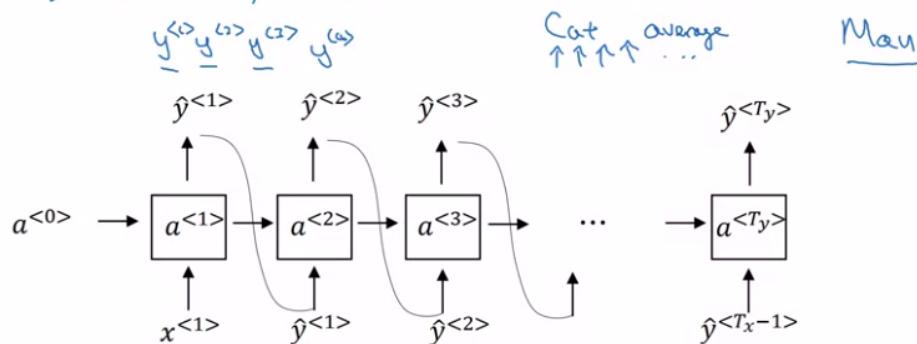
W1L7: Sampling Novel Sequences

- Character level: computationally expensive but accurate
 - Word level models are simple and most widely used.

Character-level language model

→ Vocabulary = [a, aaron, ..., zulu, <UNK>] ↵

$\rightarrow \text{Vocabulary} = [a, b, c, \dots, z, \cup, ., , ;, 0, \dots, 9, A, \dots, Z]$



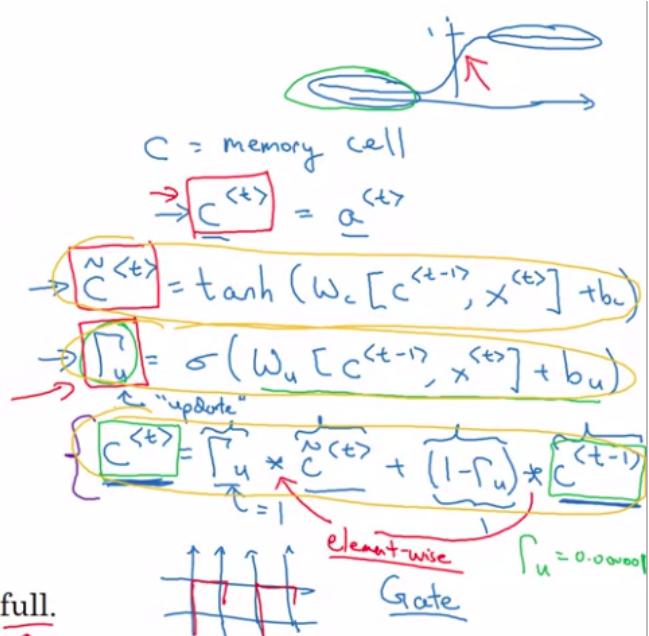
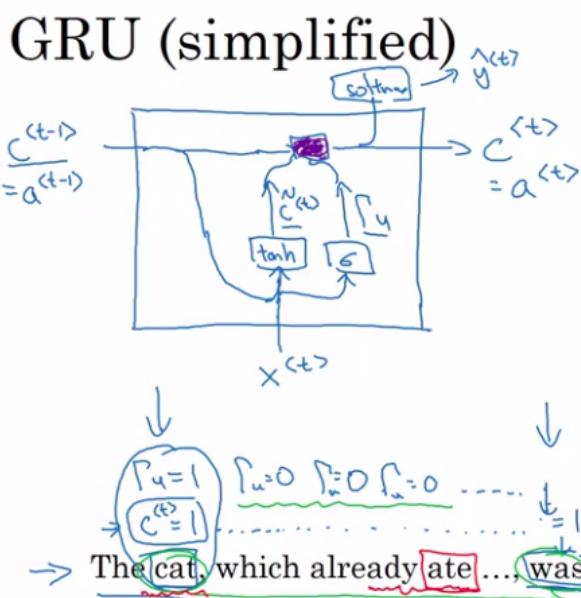
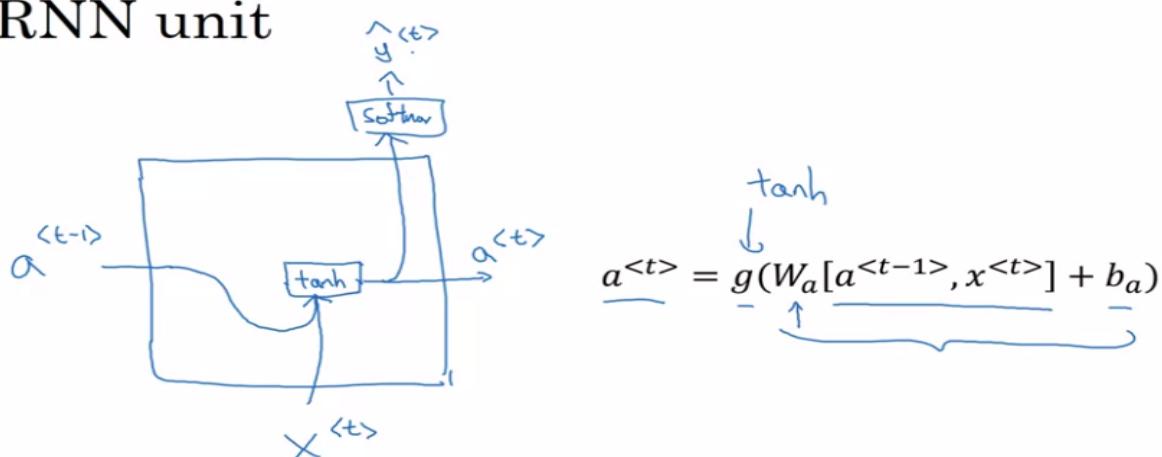
Andrew Ng

W1L8: Vanishing gradients with RNN

- For an RNN, the output is dependent on the inputs close to it, and not much on the earlier inputs, i.e. it fails to backpropagate all the way to start for computing the output towards the end. So, RNNs are not very good at capturing long-range dependencies. This is an issue with the RNNs.
- The sentences might be long and last words could be dependent on the starting words, this can't be captured by the RNNs.
- To take care of this: GRUs.

W1L9: Gated Recurrent Unit

RNN unit



[Cho et al., 2014. On the properties of neural machine translation: Encoder-decoder approaches] [Chung et al., 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling]

Andrew Ng

Full GRU

$$\tilde{c}^{<t>} = \tanh(W_c[\Gamma_r * c^{<t-1>}, x^{<t>}] + b_c)$$

$$\Gamma_u = \sigma(W_u[c^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_r = \sigma(W_r[c^{<t-1>}, x^{<t>}] + b_r)$$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + (1 - \Gamma_u) * c^{<t-1>}$$

Correction in "Gated Recurrent Unit (GRU)" (14:04):

The last line should use an element-wise multiplication "*" instead of a plus sign "+".

W1L10: LSTM

GRU and LSTM

GRU	LSTM
$\tilde{c}^{<t>} = \tanh(W_c[\Gamma_r * c^{<t-1>}, x^{<t>}] + b_c)$	$c^{<t>} = \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c)$
$\Gamma_u = \sigma(W_u[c^{<t-1>}, x^{<t>}] + b_u)$	$\Gamma_u = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u)$
$\Gamma_r = \sigma(W_r[c^{<t-1>}, x^{<t>}] + b_r)$	$\Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f)$
$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + (1 - \Gamma_u) * c^{<t-1>} \quad \text{(output)}$	$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>}$
$a^{<t>} = c^{<t>}$	$a^{<t>} = \Gamma_o * c^{<t>}$

LSTM in pictures

$$\tilde{c}^{<t>} = \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c)$$

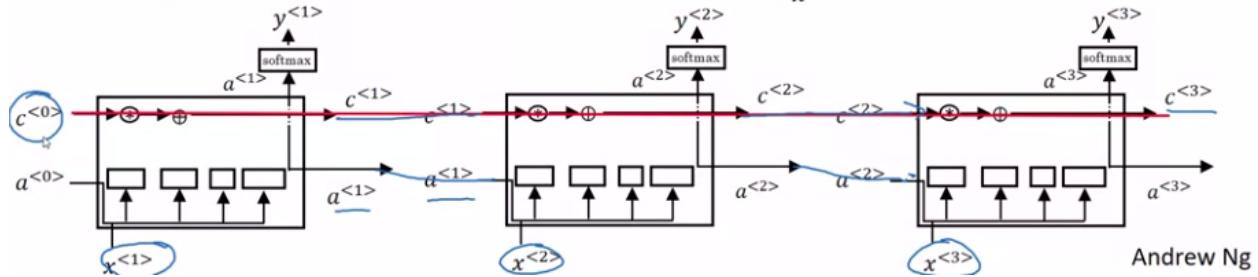
$$\Gamma_u = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f)$$

$$\Gamma_o = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o)$$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>}$$

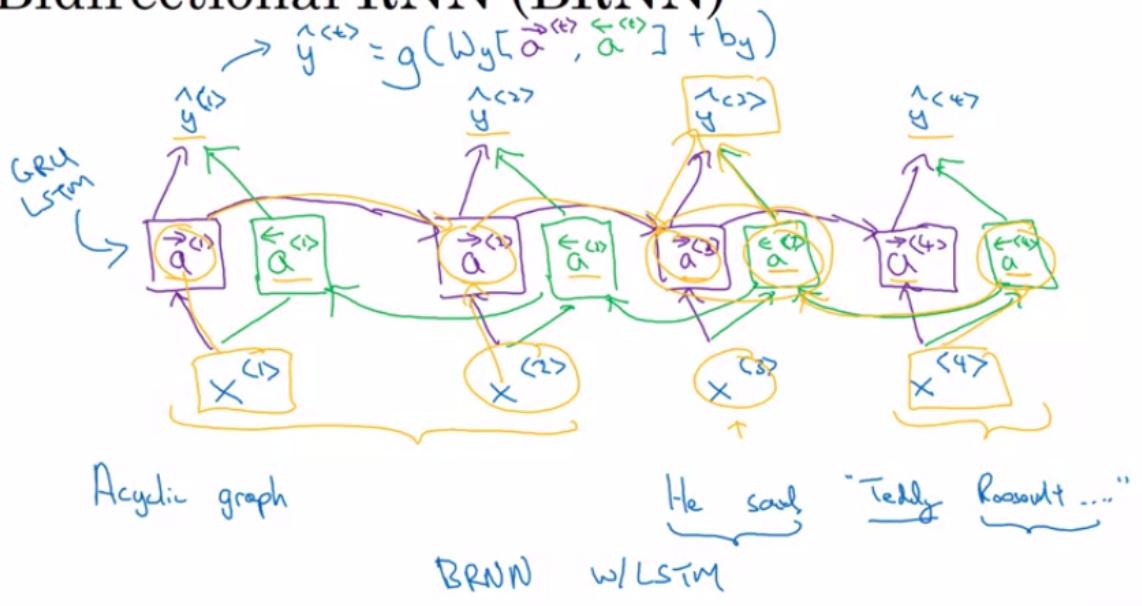
$$a^{<t>} = \Gamma_o * \tanh c^{<t>}$$



W1L11: Bidirectional RNN

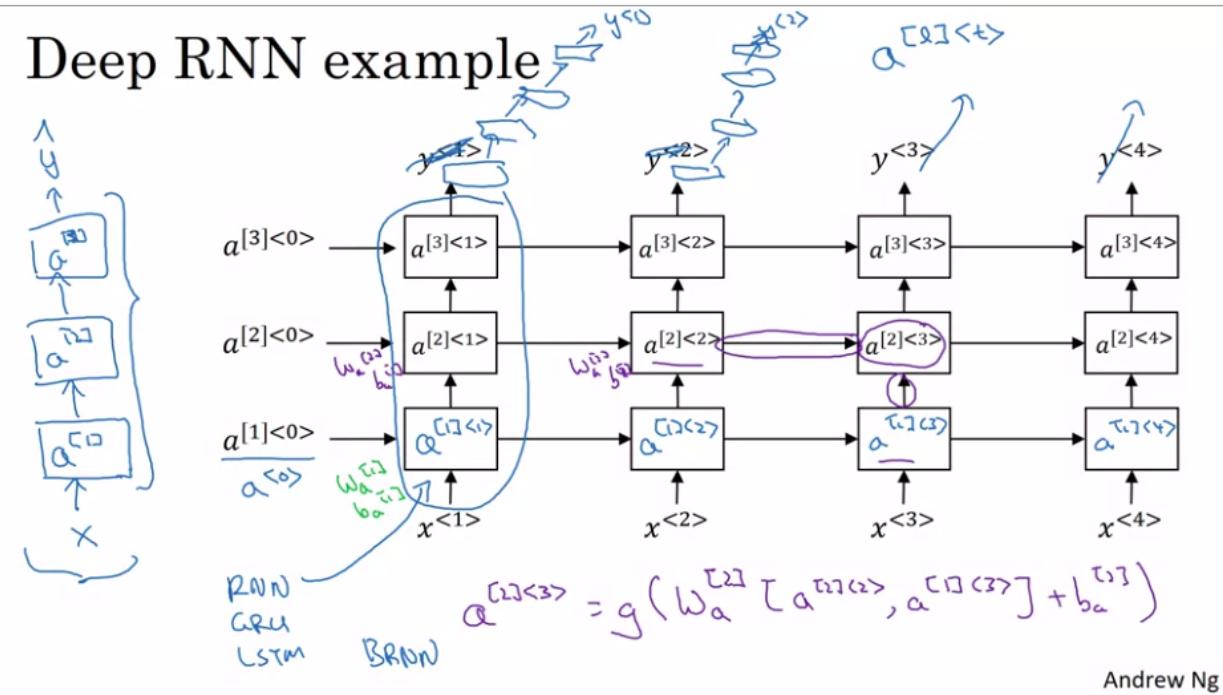
- BRNN with LSTM is a popular model.

Bidirectional RNN (BRNN)



Andrew Ng

W1L12: Deep RNNs



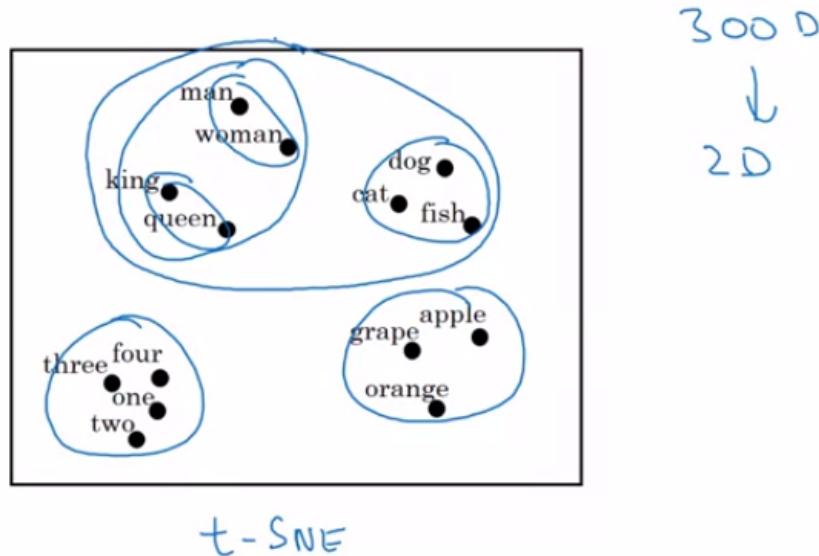
Andrew Ng

W2L1: Word representation

Featurized representation: word embedding

	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	-1	1	-0.95	0.97	0.00	0.01
300 Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	-0.02
Food	0.04	0.01	0.02	0.01	0.95	0.97
size cost alt+ verb	⋮	⋮	⋮	⋮	I want a glass of orange juice.	
	e_{5391}	e_{9853}			I want a glass of apple juice. Andrew Ng	

Visualizing word embeddings



- t-SNE algorithm for the visualization

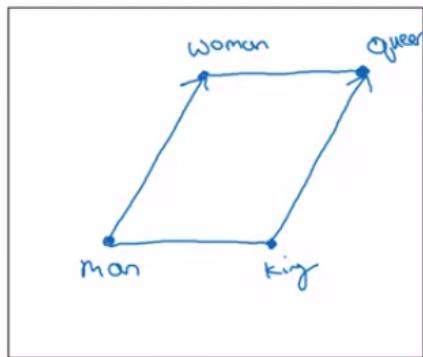
W2L2:

Transfer learning and word embeddings

- A [1. Learn word embeddings from large text corpus. (1-100B words)
(Or download pre-trained embedding online.)
- B [2. Transfer embedding to new task with smaller training set.
(say, 100k words) $\rightarrow 10,000 \rightarrow 300$
3. Optional: Continue to finetune the word embeddings with new data.

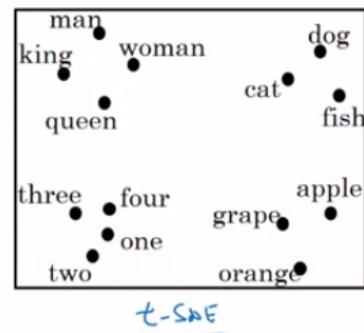
W2L3: Properties of Word Embeddings: similarity

Analogies using word vectors



Find word w: $\arg \max_w$

$3000 \rightarrow 20$



$$e_{\text{man}} - e_{\text{woman}} \approx e_{\text{king}} - e_{\text{queen}}$$

$$\text{Sim}(e_w, e_{\text{king}} - e_{\text{man}} + e_{\text{woman}})$$

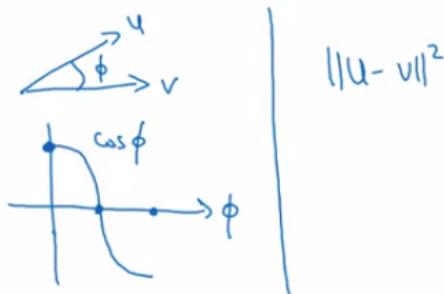
30 - 75%

Andrew Ng

Cosine similarity

$$\rightarrow \boxed{\text{sim}(e_w, e_{king} - e_{man} + e_{woman})}$$

$$\text{sim}(u, v) = \frac{u^T v}{\|u\|_2 \|v\|_2}$$



Man:Woman as Boy:Girl

Ottawa:Canada as Nairobi:Kenya

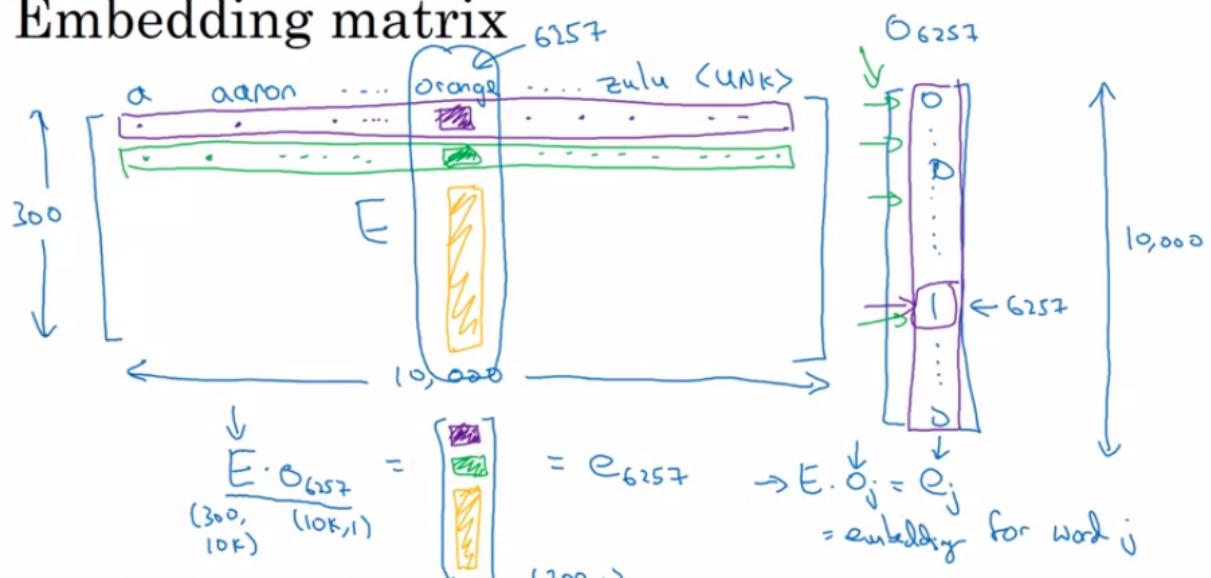
Big:Bigger as Tall:Taller

Yen:Japan as Ruble:Russia

Andrew Ng

W2L4: Embedding Matrix

Embedding matrix

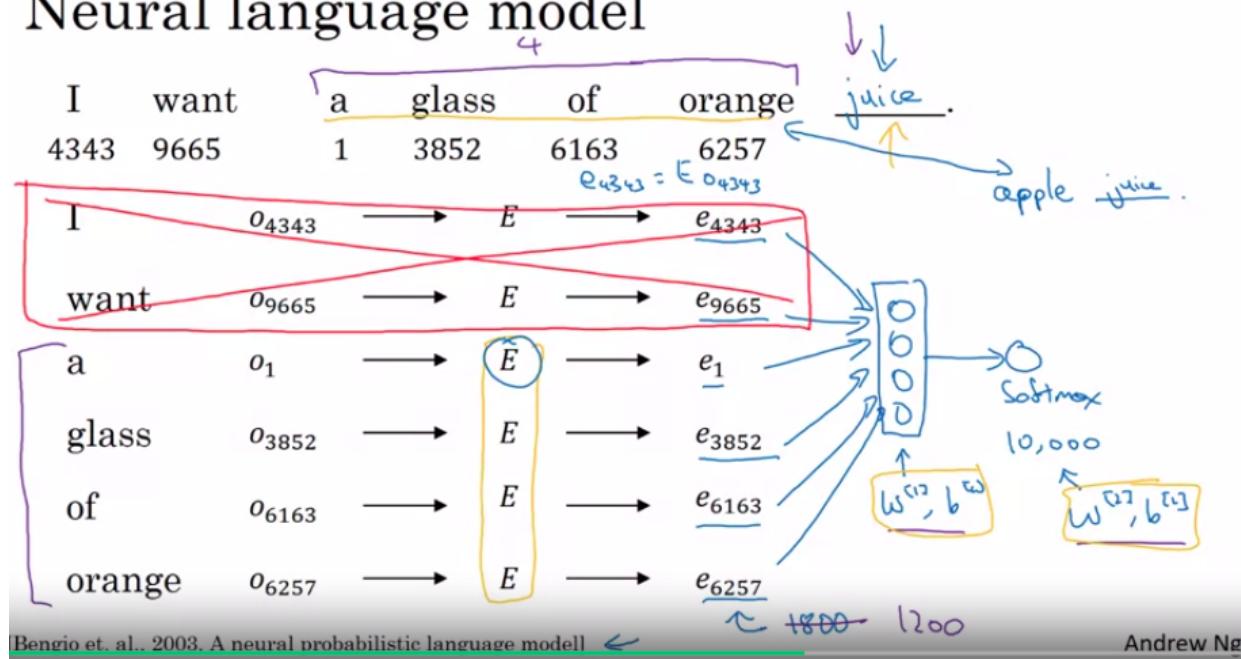


In practice, use specialized function to look up an embedding.

Andrew Ng

W2L5: Learning word embeddings

Neural language model

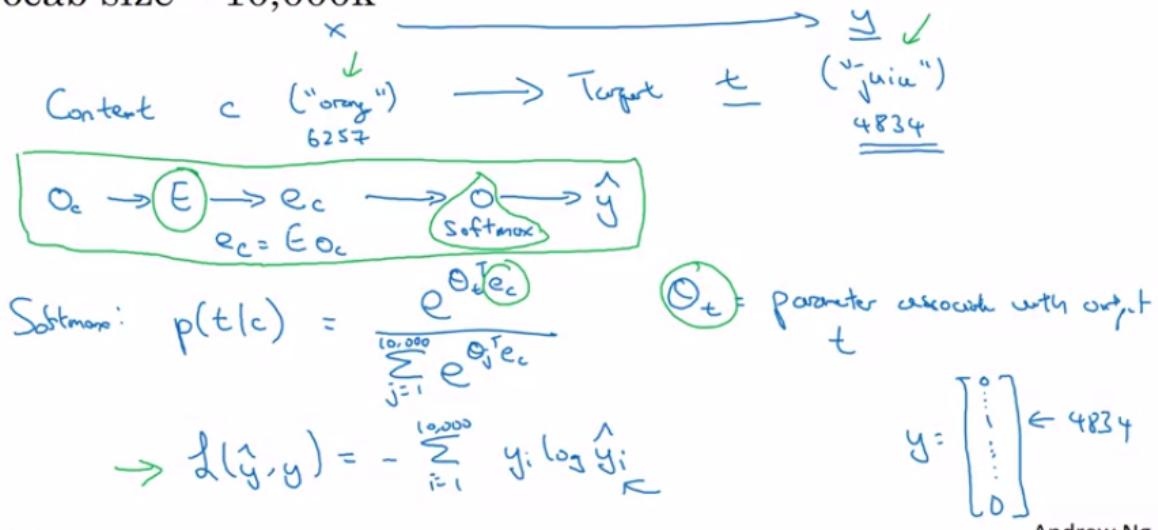


- The window of context can be variable and be also decided to use left and right 4 words or just one previous word and so on.

W2L6: Word2Vec

Model

Vocab size = 10,000k



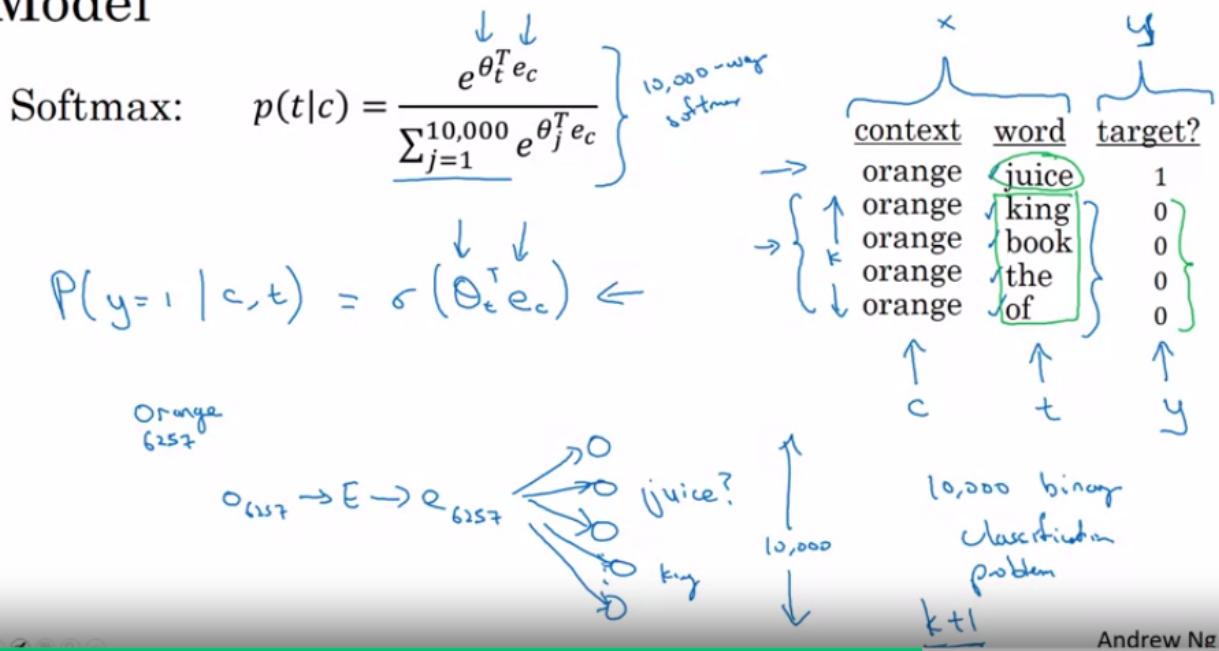
- To sample context c : uniformly random probability is used.
- The issue with this model: The softmax calculation is computationally expensive.

- Solution:

1. Hierarchical softmax: using the tree data structure. Generally, the trees are unbalanced as the most commonly used words are in upper nodes.
2. Negative sampling

W2L7: Negative Sampling

Model



- Choosing negative samples:

$$P(w_i) = \frac{f(w_i)^{3/4}}{\sum_{j=1}^{10,000} f(w_j)^{3/4}}$$

W2L8: Global vectors for word representations(GloVe):

- X_{ij} : number of times the target word(t) has occurred with context(c)

Model

Minimize

$$\sum_{i=1}^{10,000} \sum_{j=1}^{10,000} f(x_{ij}) (\Theta_t^T e_j + b_i + b_j) - \log x_{ij}^2$$

weighting term

$f(x_{ij}) = 0$ if $x_{ij} = 0$. "0 log 0" = 0

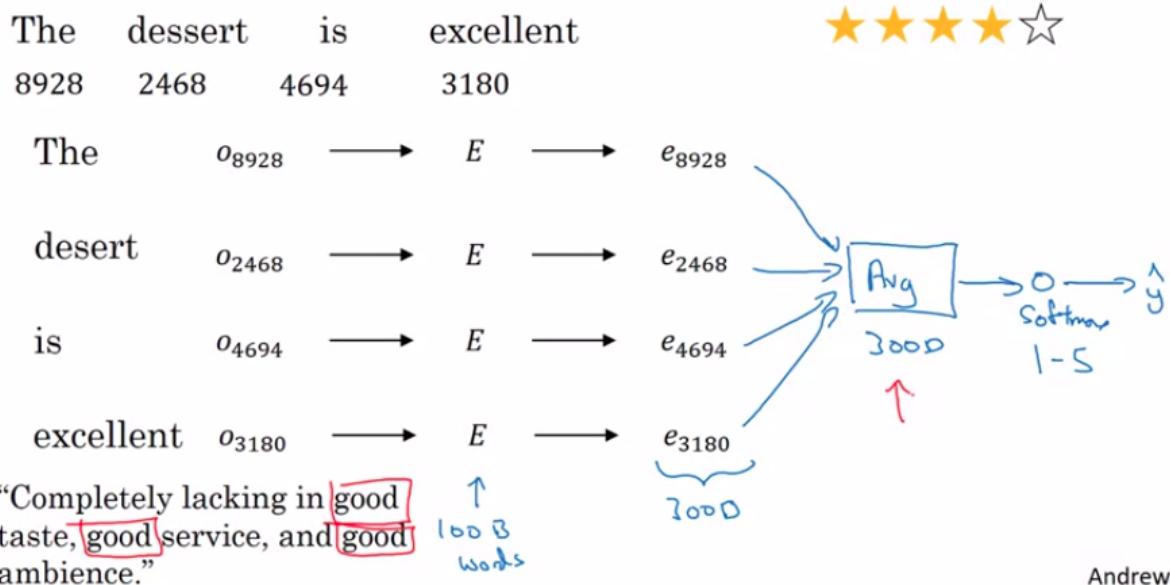
this is at a ... $\Theta_t^T e_c$ are symmetric

$\theta_w^{(final)} = \frac{\theta_w + \theta_u}{2}$

Andrew Ng

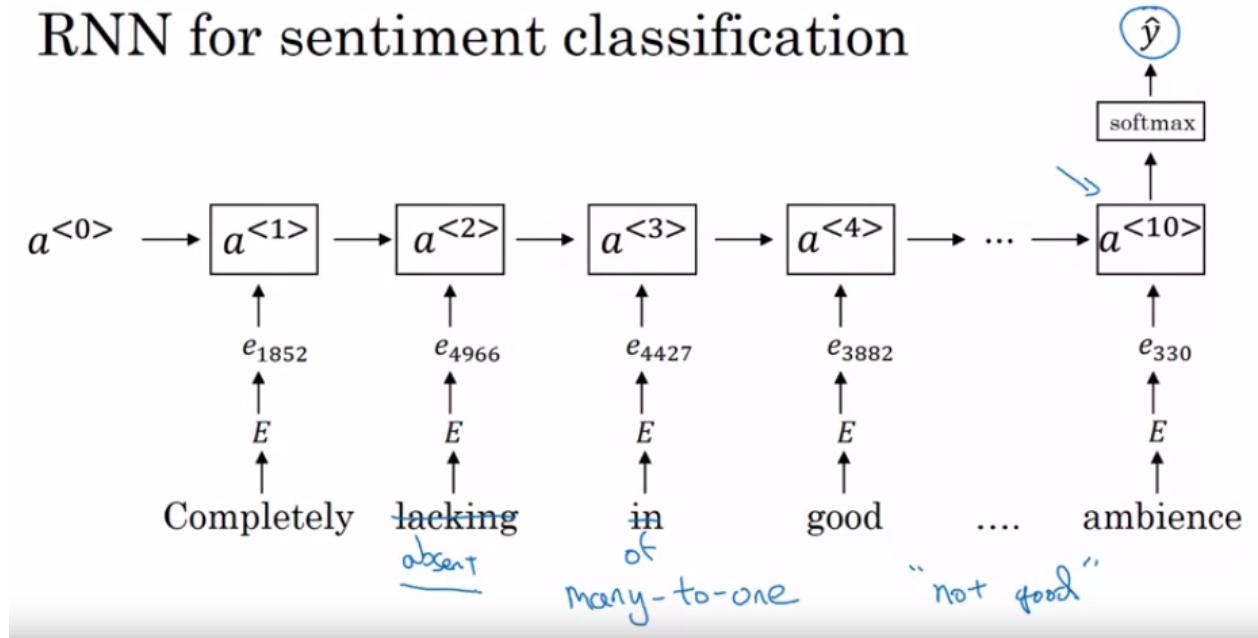
W2L9: Application: Sentiment Classification

Simple sentiment classification model



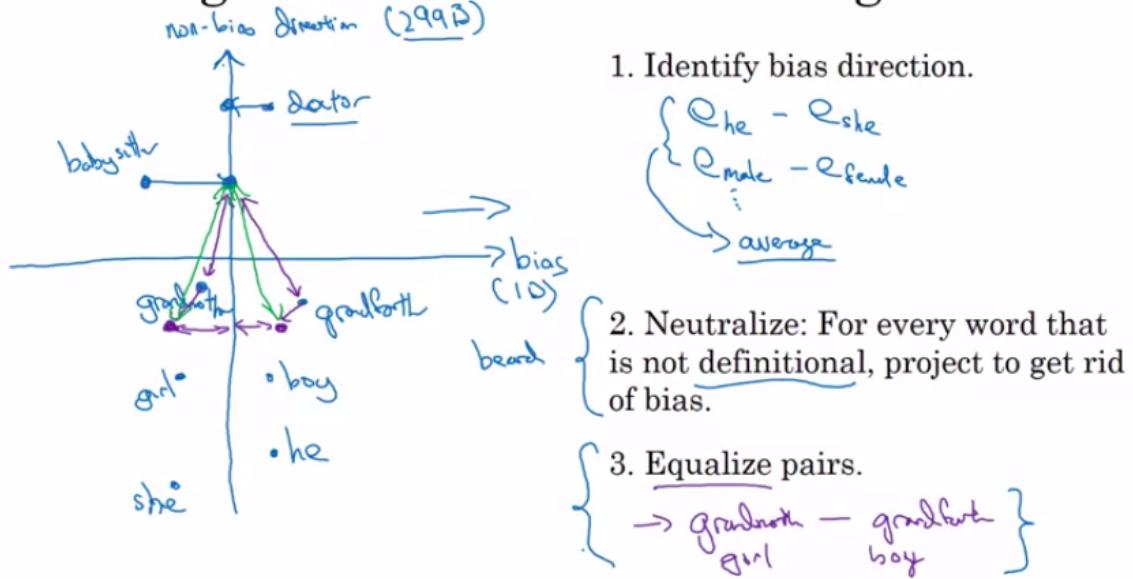
Andrew Ng

RNN for sentiment classification



W2L10: Debiasing

Addressing bias in word embeddings



Bolukbasi et. al., 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings] ↵

Andrew Ng

Week 3: Sequence to sequence architecture

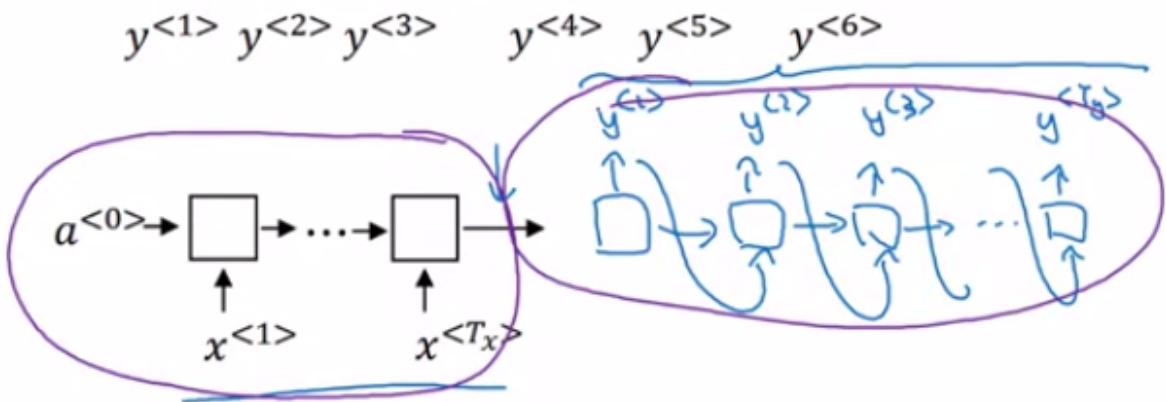
W3L1: Basic models

- Machine translation model:

Sequence to sequence model

$x^{<1>} \quad x^{<2>} \quad x^{<3>} \quad x^{<4>} \quad x^{<5>}$
 Jane visite l'Afrique en septembre

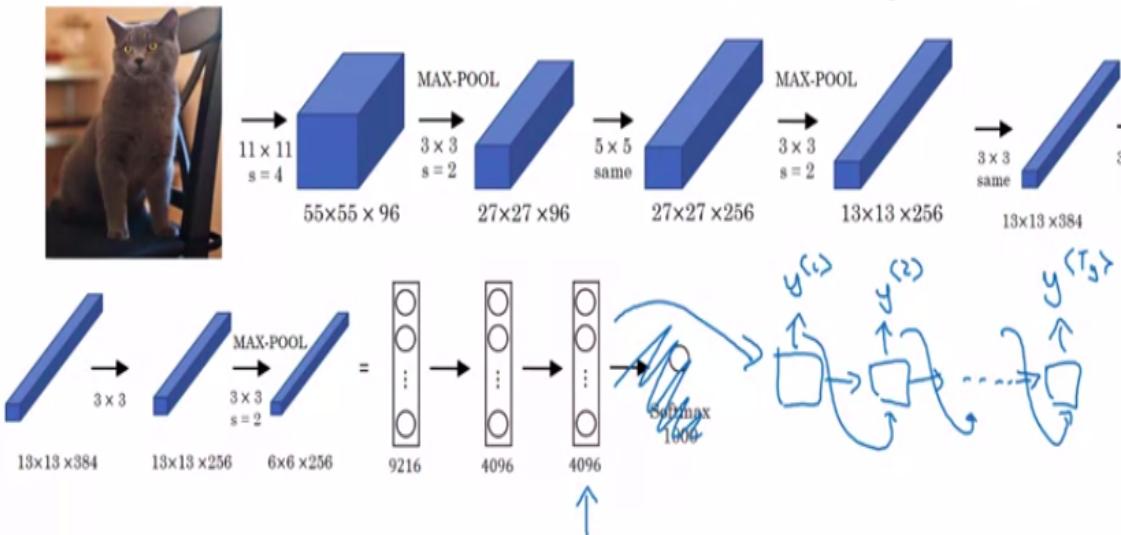
→ Jane is visiting Africa in September.



- Image captioning:

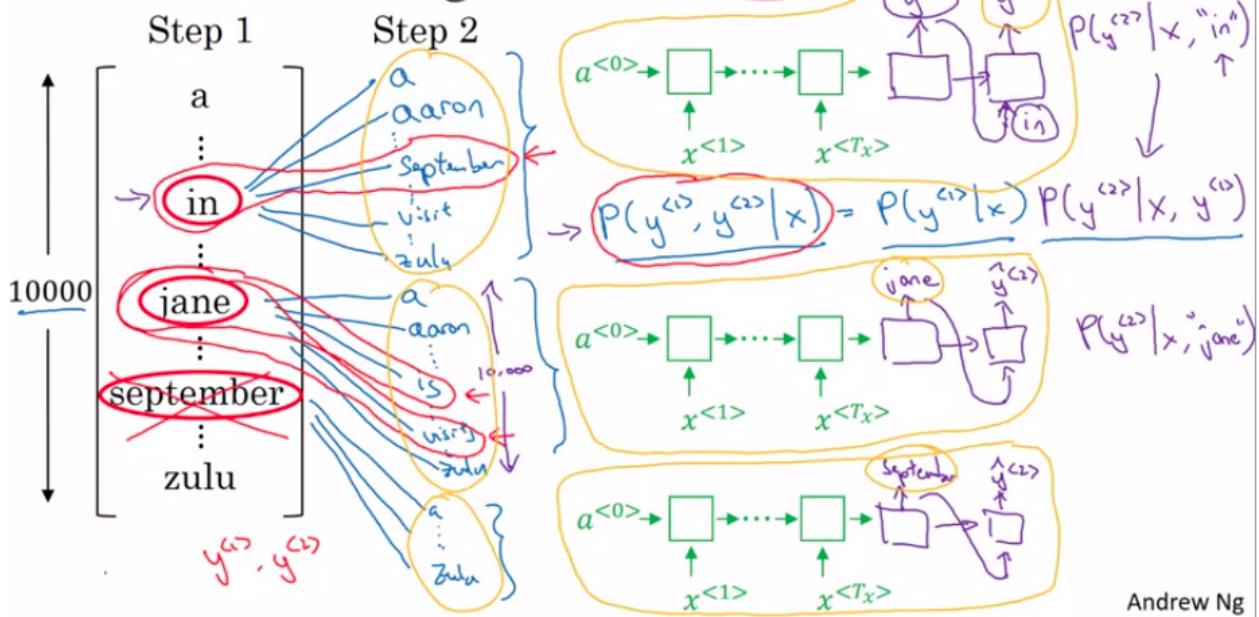
Image captioning

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

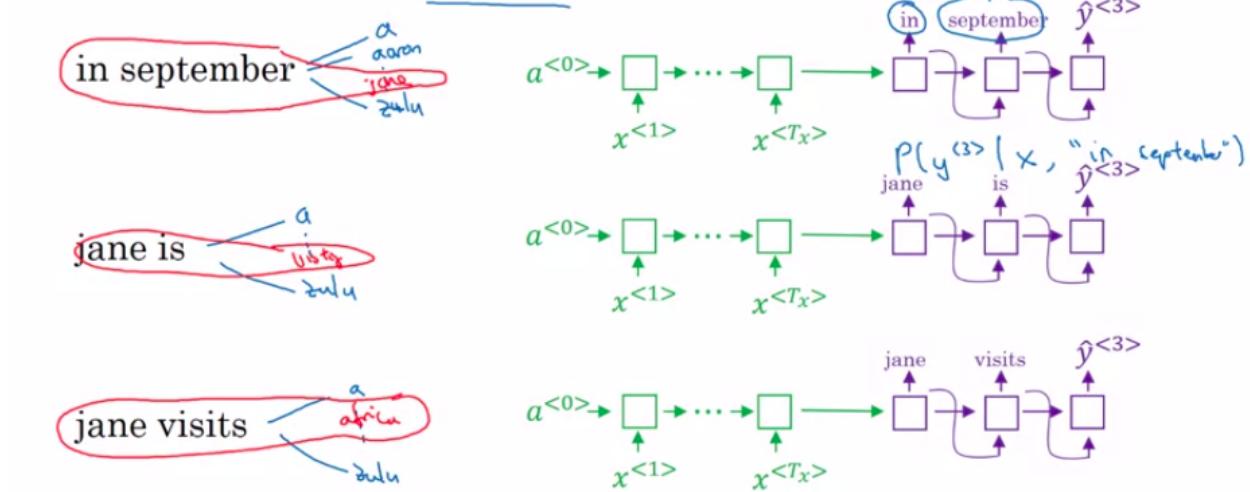


W3L3: Beam Search

Beam search algorithm ($B=3$)



Beam search ($B = 3$)



W3L4: Refinements in the beam search

Length normalization

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

Andrew Ng

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

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

$P(y|x) \leftarrow$

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

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

$\rightarrow \frac{1}{T_y^\alpha} \sum_{t=1}^{T_y} \log P(y^{<t>} | x, y^{<1>}, \dots, y^{<t-1>}) \quad \alpha = 0.7 \quad \alpha = 1 \quad \alpha = 0$

- Choosing beam width(B) large: better results but slower
- Beam width small: worst results but faster
- In production: 10-100 is mostly used.

W3L5: Error analysis in beam search

Error analysis on beam search

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

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

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

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.

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
...	...	—	—	R R R ...
...	...	—	—	

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

Andrew Ng

- Modify the one causing maximum errors.
-

W3L5: Bleu Score

- There might be more than 1 correct translations of any sentence. In that case, the bleu score evaluates the correctness of the machine-translated output.

Evaluating machine translation

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.

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

Bleu
bilingual evaluation under study
Modified precision: $\frac{2}{7}$

Count_{clip} ("the")
Count ("the")

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	Countclip	
the cat	2 ←	1 ←	
cat the	1 ←	0	
cat on	1 ←	1 ←	
on the	1 ←	1 ←	
the mat	1 ←	1 ←	

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

Andrew Ng

Bleu score on unigrams

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

$$P_1, P_2, \dots = 1.0$$

Reference 2: There is a cat on the mat.

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

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

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

Bleu details

p_n = Bleu score on n-grams only

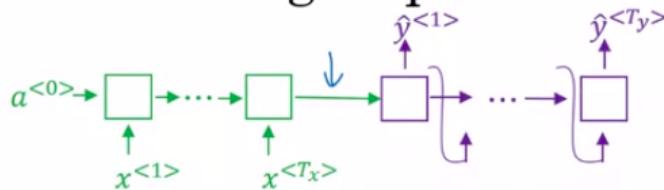
$$P_1, P_2, P_3, P_4$$

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

BP = bleu penalty

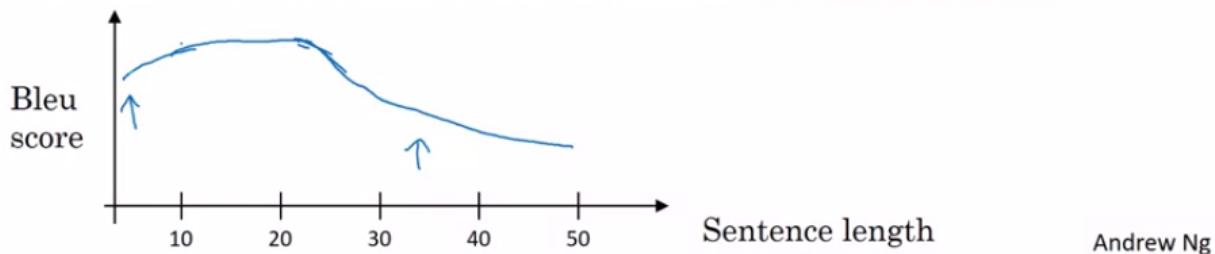
$$\text{BP} = \begin{cases} 1 & \text{if } \text{MT_output_length} > \text{reference_output_length} \\ \exp(1 - \text{MT_output_length}/\text{reference_output_length}) & \text{otherwise} \\ \exp(1 - \text{reference_output_length}/\text{MT_output_length}) & \end{cases}$$

The problem of long sequences



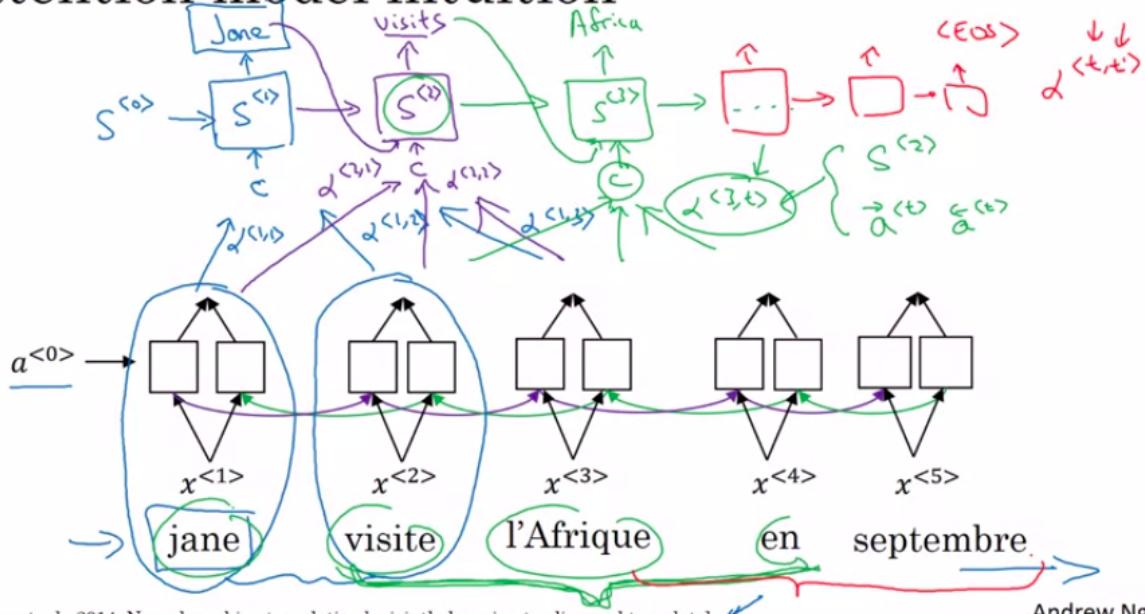
Jane s'est rendue en Afrique en septembre dernier, a apprécié la culture et a rencontré beaucoup de gens merveilleux; elle est revenue en parlant comment son voyage était merveilleux, et elle me tente d'y aller aussi.

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.



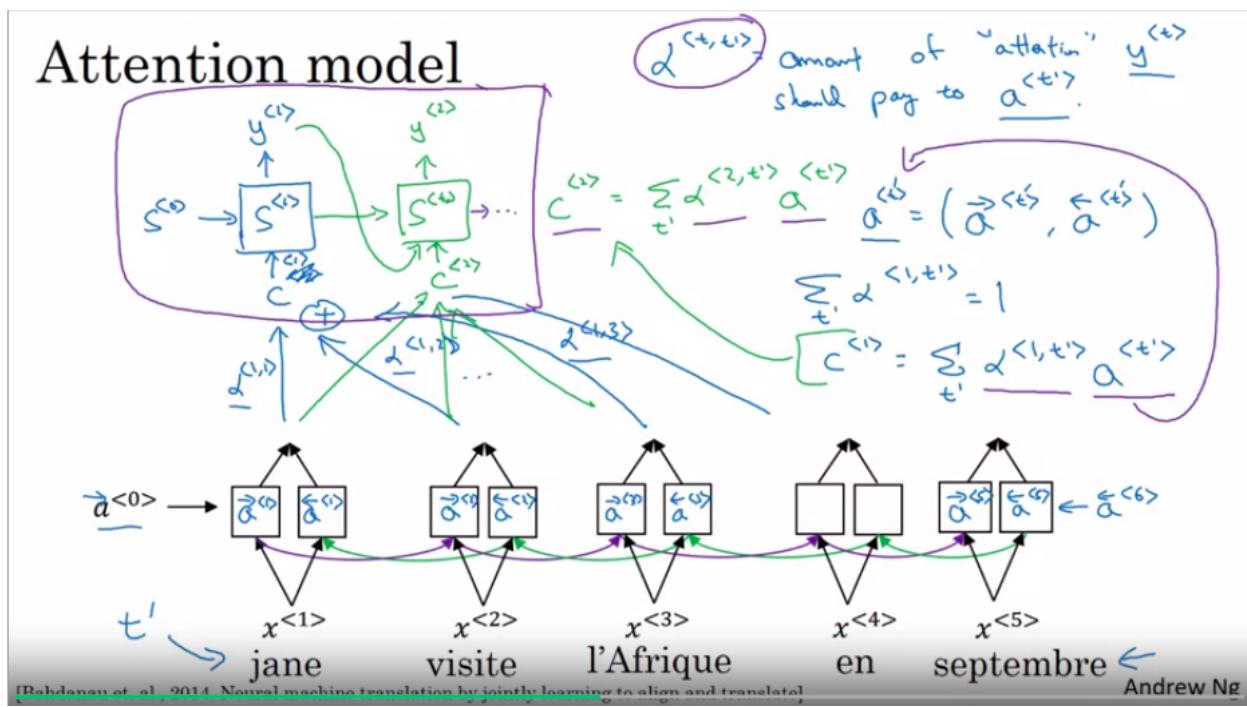
- To avoid this, ATTENTION MODEL

Attention model intuition

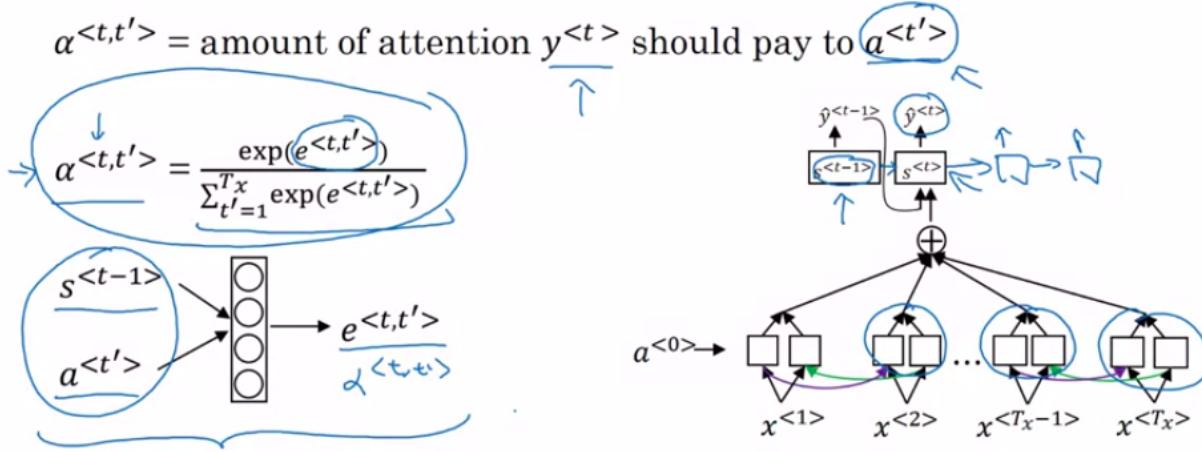


- $\alpha(m,n)$: the amount of attention paid to n-th word to predict m-th word

Attention model



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



Trigger word detection algorithm

