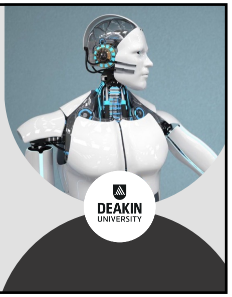
## SIT330-770: Natural Language Processing

Week 6 - Neural Networks for NLP

Dr. Mohamed Reda Bouadjenek

School of Information Technology, Faculty of Sci Eng & Built Env

reda.bouadjenek@deakin.edu.au



1

Andrew Ng

Neural Networks and Deep Learning

(Optional)

2

## SIT330-770: Natural Language Processing

Week 6.11 - Applying feedforward networks to NLP tasks

Dr. Mohamed Reda Bouadjenek

School of Information Technology, Faculty of Sci Eng & Built Env

3

### Use cases for feedforward networks

- Let's consider 2 (simplified) sample tasks:

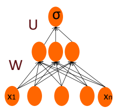
1. Text classification
2. Language modeling

- State-of-the-art systems use more powerful neural architectures, but simple models are useful to consider!

4

### Classification: Sentiment Analysis

- We could do exactly what we did with logistic regression
- Input layer are binary features as before
- Output layer is 0 or 1



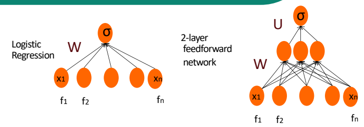

5

### Sentiment Features

| Var | Definition |
|---|---|
| $x_1$ | count(positive lexicon) $\in$ doc) |
| $x_2$ | count(negative lexicon) $\in$ doc) |
| $x_3$ | $\begin{cases} 1 & \text{if "no"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$ |
| $x_4$ | count(1st and 2nd pronouns $\in$ doc) |
| $x_5$ | $\begin{cases} 1 & \text{if "!"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$ |
| $x_6$ | log(word count of doc) |

6

## Feedforward nets for simple classification



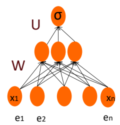

- Just adding a hidden layer to logistic regression
  - allows the network to use non-linear interactions between features
  - which may (or may not) improve performance.



7

## Even better: representation learning

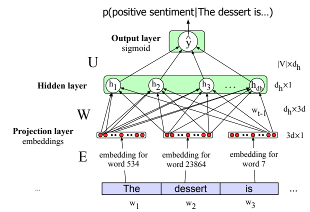
- The real power of deep learning comes from the ability to **learn** features from the data
- Instead of using hand-built human-engineered features for classification
- Use learned representations like embeddings!






8

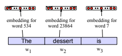
## Neural Net Classification with embeddings as input features!






9

## Issue: texts come in different sizes




- This assumes a fixed size length (3)!
- Kind of unrealistic.
- Some simple solutions (more sophisticated solutions later)

1. Make the input the length of the longest review
   - If shorter then pad with zero embeddings
   - Truncate if you get longer reviews at test time
2. Create a single "sentence embedding" (the same dimensionality as a word) to represent all the words
   - Take the mean of all the word embeddings
   - Take the element-wise max of all the word embeddings
     - For each dimension, pick the max value from all words

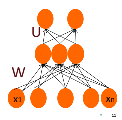


10

## Reminder: Multiclass Outputs

- What if you have more than two output classes?
  - Add more output units (one for each class)
  - And use a "softmax layer"

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \quad 1 \le i \le D$$






11

## Neural Language Models (LMs)

- **Language Modeling**: Calculating the probability of the next word in a sequence given some history.
  - We've seen N-gram based LMs
  - But neural network LMs far outperform n-gram language models
- State-of-the-art neural LMs are based on more powerful neural network technology like Transformers
- But **simple feedforward LMs** can do almost as well!



12

## Simple feedforward Neural Language Models

**Task**: predict next word $w_t$

given prior words $w_{t-1}, w_{t-2}, w_{t-3}, \ldots$

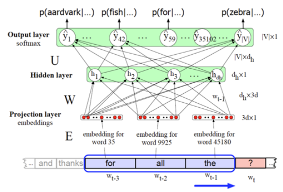
**Problem**: Now we're dealing with sequences of arbitrary length.

**Solution**: Sliding windows (of fixed length)

$$P(w_t|w_1^{t-1}) \approx P(w_t|w_{t-N+1}^{t-1})$$

13

## Neural Language Model




14

## Why Neural LMs work better than N-gram LMs

- **Training data:**
  - We've seen: I have to make sure that the cat gets fed.
  - Never seen: dog gets fed
- **Test data:**
  - I forgot to make sure that the dog gets ___
- N-gram LM can't predict "fed"!
- Neural LM can use similarity of "cat" and "dog" embeddings to generalize and predict "fed" after dog

15

# SIT330-770: Natural Language Processing

## Week 6 - Neural Networks and Neural LMs

Dr. Mohamed Reda Bouadjenek

School of Information Technology, Faculty of Sci Eng & Built Env

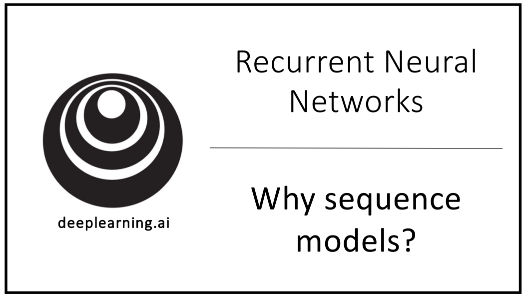
reda.bouadjenek@deakin.edu.au

16

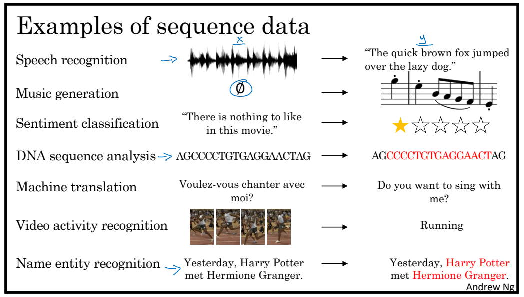
Andrew Ng

Neural Networks and Deep Learning

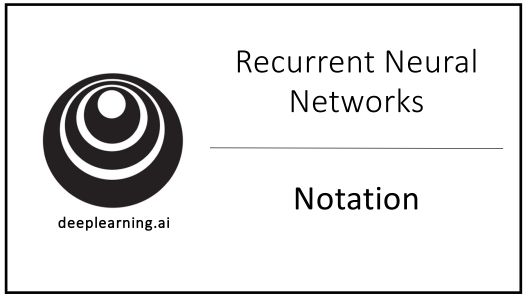
17


deeplearning.ai
Recurrent Neural Networks
Why sequence models?

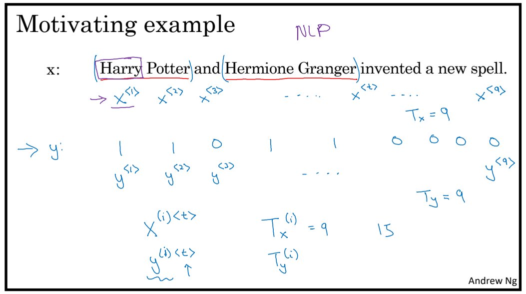
1


Examples of sequence data
Speech recognition
"The quick brown fox jumped over the lazy dog."
Music generation
Sentiment classification
"There is nothing to like in this movie."
DNA sequence analysis
AGCCCCTGTGAGGAACTAG
AGCCCCTGTGAGGAACTAG
Machine translation
Voulez-vous chanter avec moi?
Do you want to sing with me?
Video activity recognition
Running
Name entity recognition
Yesterday, Harry Potter met Hermione Granger.
Yesterday, Harry Potter met Hermione Granger.
Andrew Ng

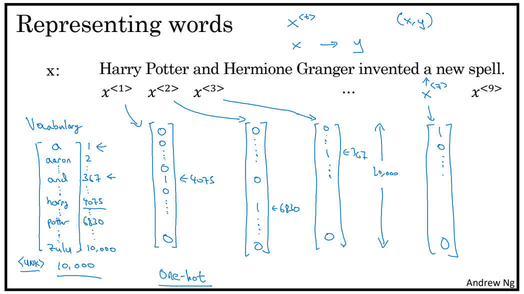
2


deeplearning.ai
Recurrent Neural Networks
Notation

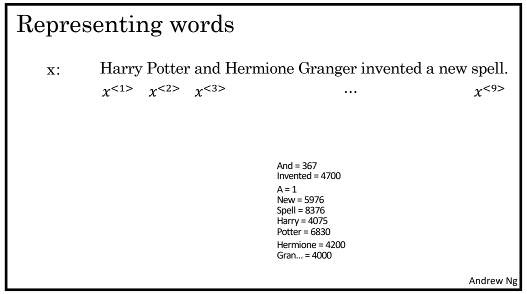
3


Motivating example
NLP
x: Harry Potter and Hermione Granger invented a new spell.
Tx = 9
Ty = 9
Andrew Ng

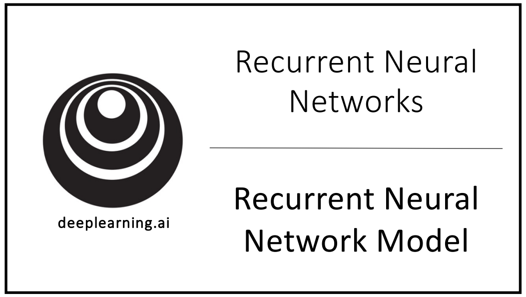
4


Representing words
x: Harry Potter and Hermione Granger invented a new spell.
Vocabulary
a
aaron
and
harry
potter
zulu
10,000
One-hot
Andrew Ng

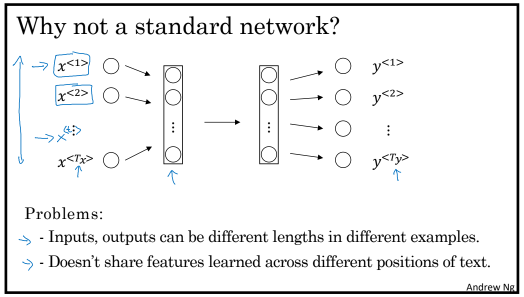
5


Representing words
x: Harry Potter and Hermione Granger invented a new spell.
And = 367
Invented = 4700
A = 1
New = 5976
Spell = 8376
Harry = 4075
Potter = 6830
Hermione = 4200
Gran… = 4000
Andrew Ng

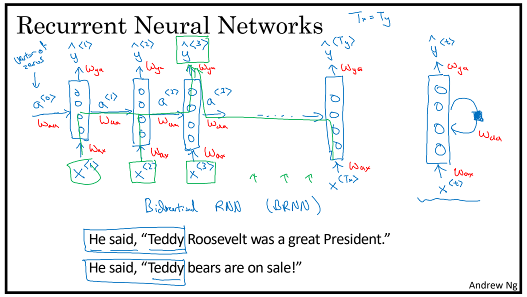
6


Recurrent Neural Networks
Recurrent Neural Network Model
deeplearning.ai

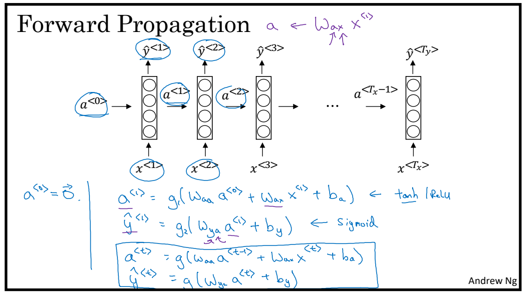
7


Why not a standard network?
$x^{<1>}$
$x^{<2>}$
$x^{<T_x>}$
$y^{<1>}$
$y^{<2>}$
$y^{<T_y>}$
Problems:
- Inputs, outputs can be different lengths in different examples.
- Doesn't share features learned across different positions of text.
Andrew Ng

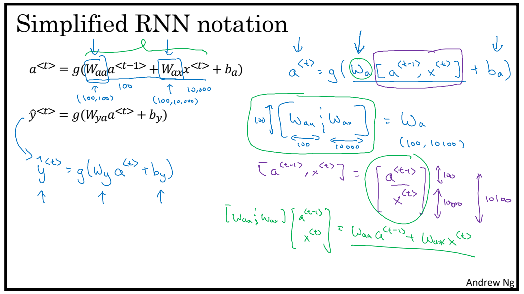
8


Recurrent Neural Networks
$T_x = T_y$
Bidirectional RNN (BRNN)
He said, "Teddy Roosevelt was a great President."
He said, "Teddy bears are on sale!"
Andrew Ng

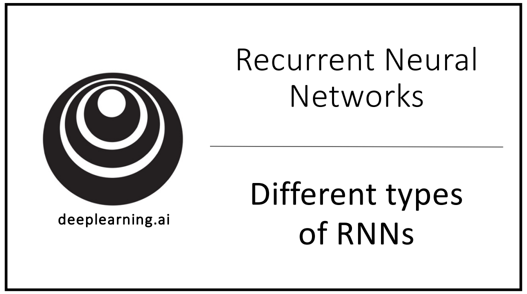
9


Forward Propagation
$a^{<0>}$
$\hat{y}^{<1>}$
$\hat{y}^{<2>}$
$\hat{y}^{<3>}$
$\hat{y}^{<T_y>}$
$x^{<1>}$
$x^{<2>}$
$x^{<3>}$
$x^{<T_x>}$
$a^{<T_x-1>}$
$a^{<0>} = \vec{0}$
$a^{<1>} = g_1(W_{aa} a^{<0>} + W_{ax} x^{<1>} + b_a)$ ← tanh / ReLU
$\hat{y}^{<1>} = g_2(W_{ya} a^{<1>} + b_y)$ ← sigmoid
$a^{<t>} = g(W_{aa} a^{<t-1>} + W_{ax} x^{<t>} + b_a)$
$\hat{y}^{<t>} = g(W_{ya} a^{<t>} + b_y)$
Andrew Ng

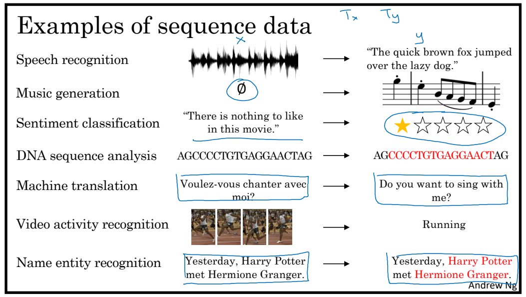
10


Simplified RNN notation
$a^{<t>} = g(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$
$\hat{y}^{<t>} = g(W_{ya}a^{<t>} + b_y)$
$a^{<t>} = g(W_a [a^{<t-1>}, x^{<t>}] + b_a)$
$[W_{aa} ; W_{ax}] = W_a$
$\hat{y}^{<t>} = g(W_y a^{<t>} + b_y)$
Andrew Ng

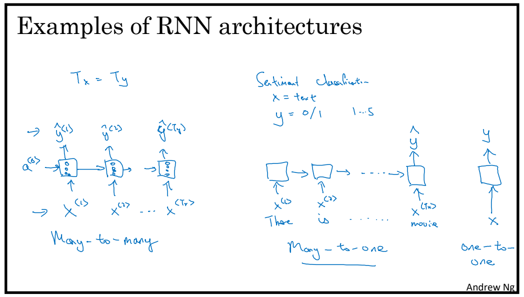
11


Recurrent Neural Networks
Different types of RNNs
deeplearning.ai

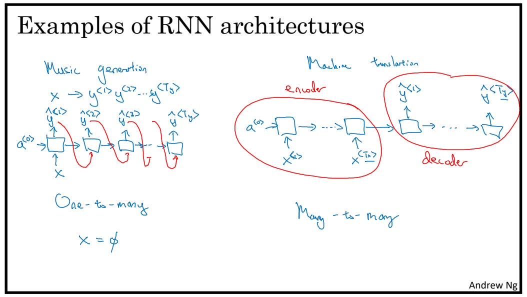
12



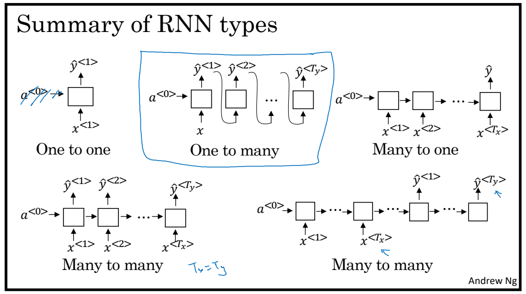

13



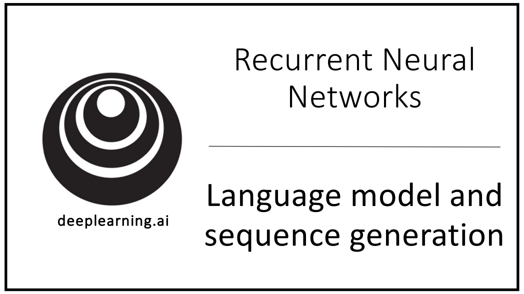

14



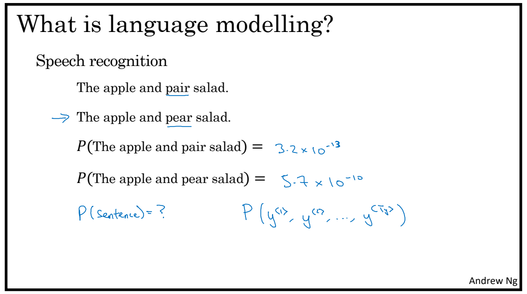

15



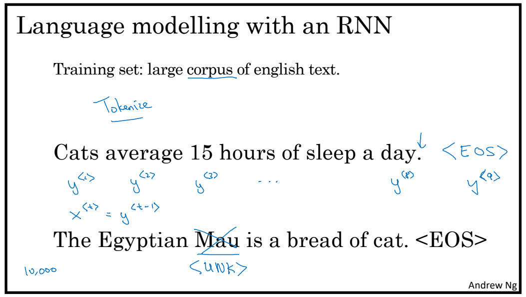

16



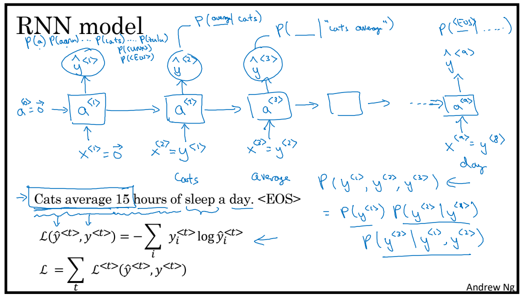

17



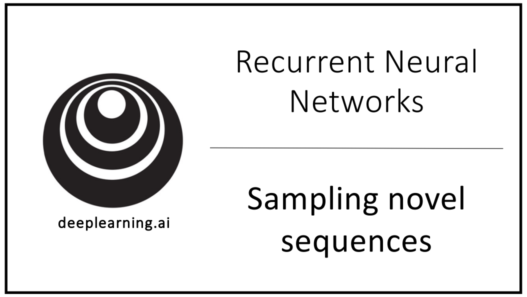

18



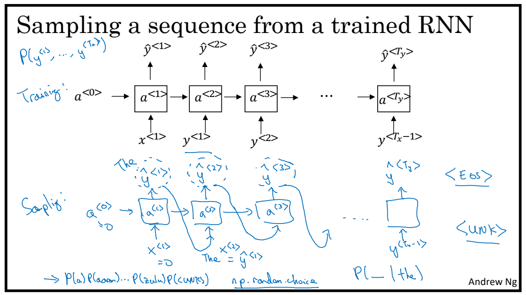

19



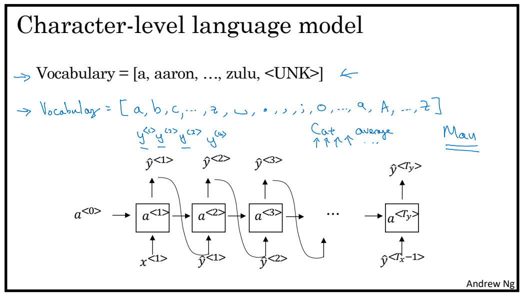

20



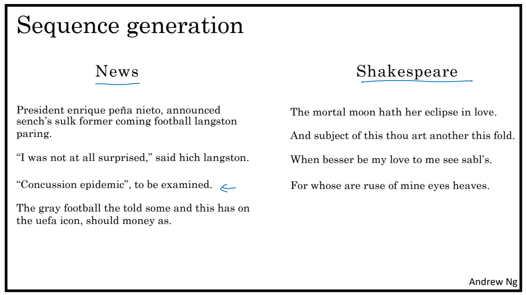

21




22




23



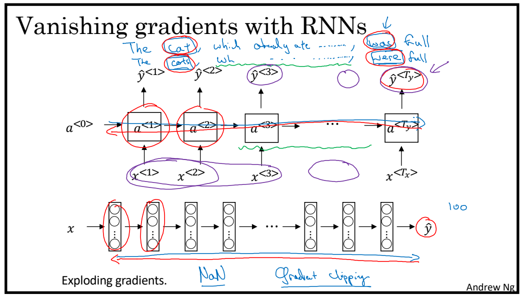

24

deeplearning.ai

Recurrent Neural Networks

Vanishing gradients with RNNs

25



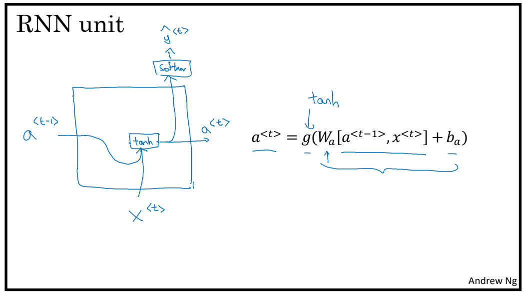

26

deeplearning.ai

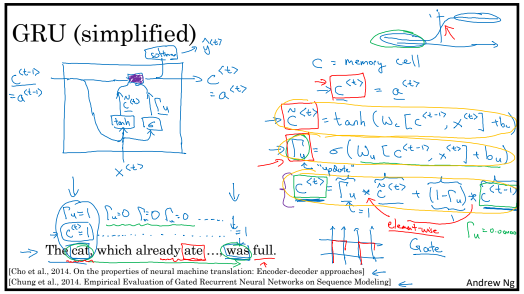
Recurrent Neural Networks

Gated Recurrent Unit (GRU)

27




28




29

## Full GRU

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

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

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

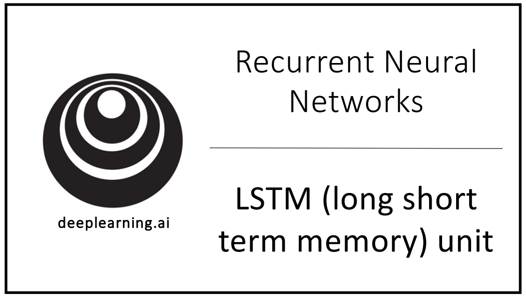
h $c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + (1 - \Gamma_u) + c^{<t-1>}$

LSTM

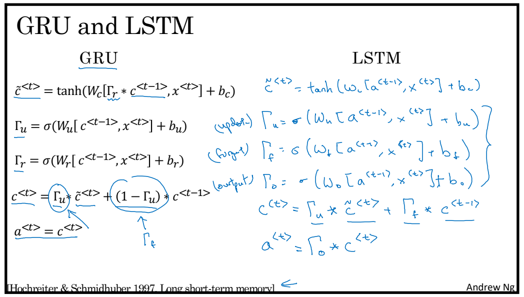
The cat, which ate already, was full.

Andrew Ng

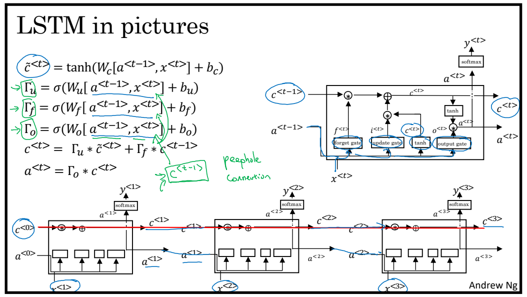
30



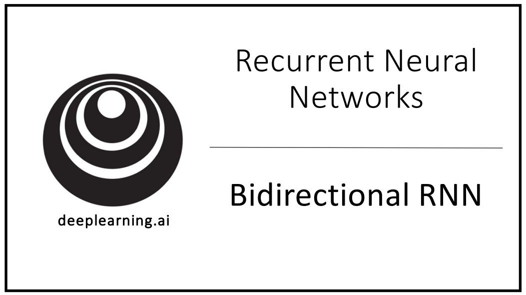

31



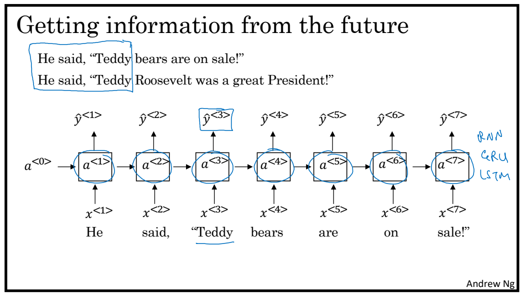

32



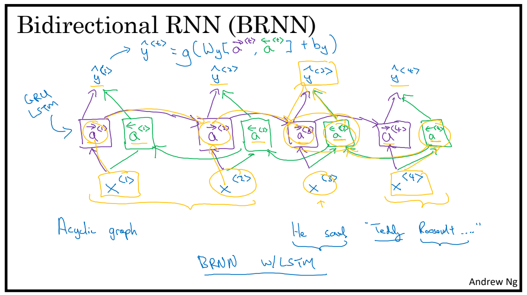

34



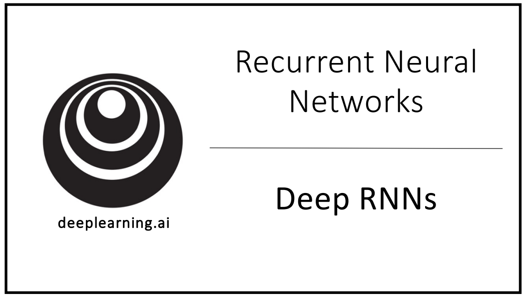

35



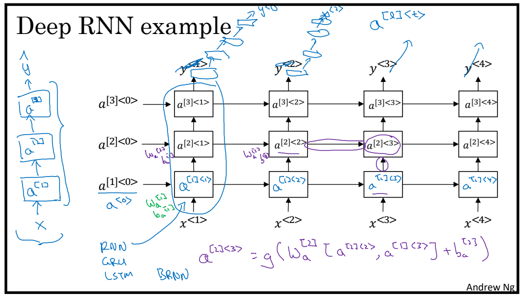

36



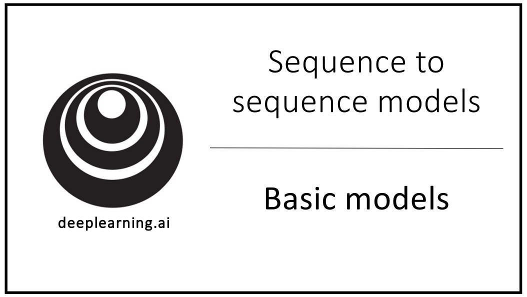

37



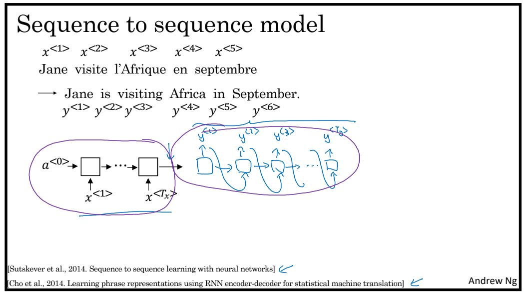

38



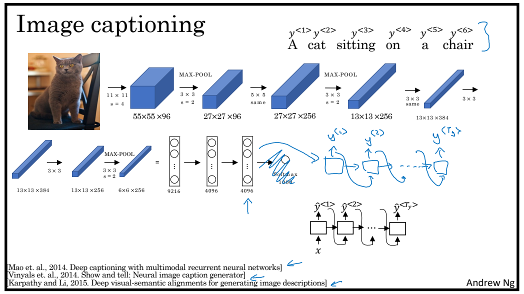

39



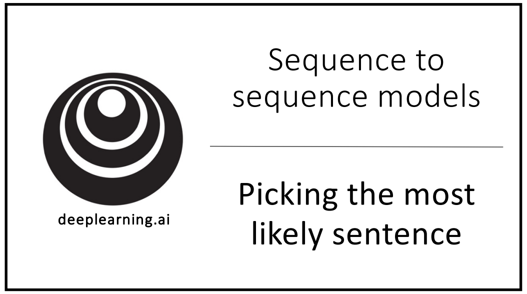

40



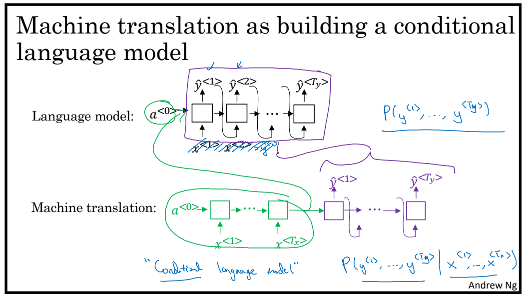

41



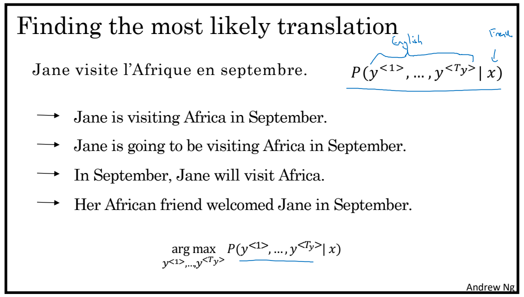

42



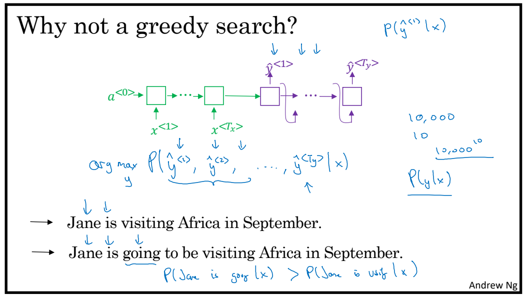

43



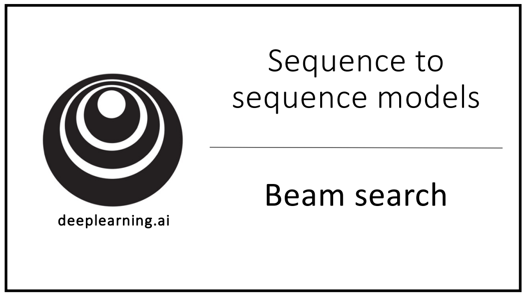

44



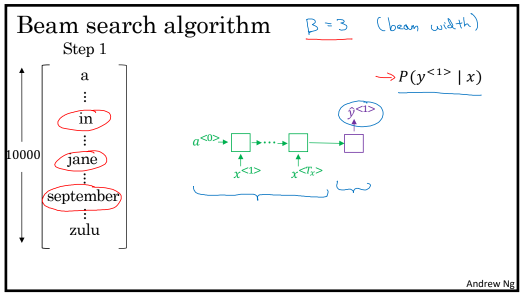

45



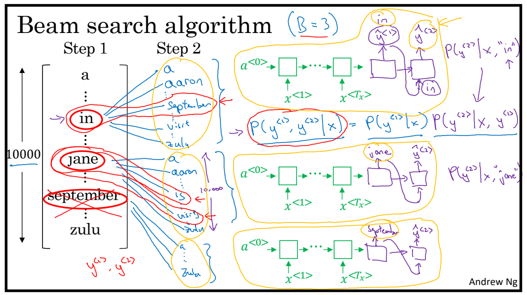

46



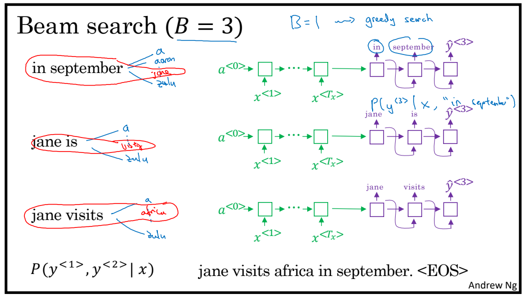

47




48




49



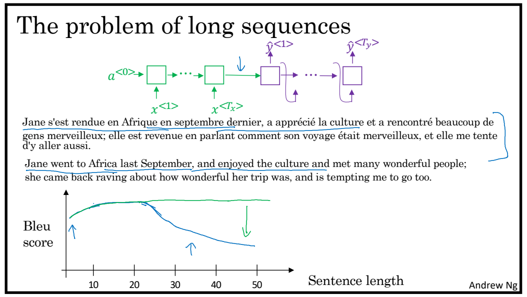

50

Sequence to sequence models

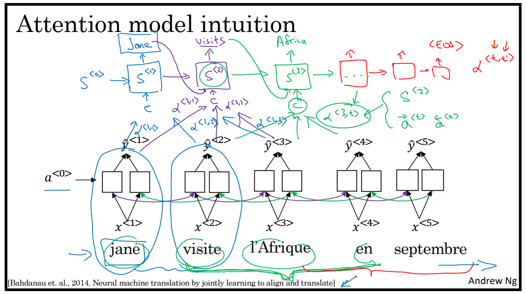
Attention model intuition

deeplearning.ai

51




52



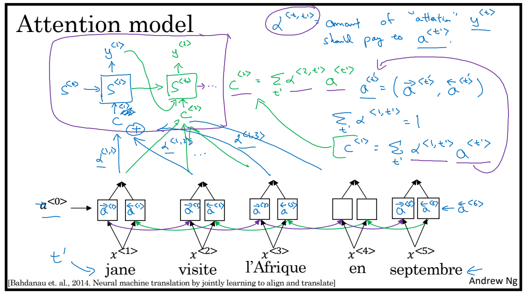

53

Sequence to sequence models

Attention model

deeplearning.ai

54



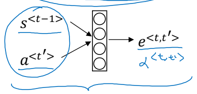

55

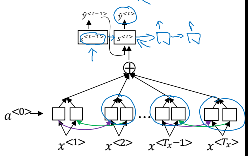
## Computing attention $\alpha^{<t,t'>}$

$T_x$ $T_y$

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

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




[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

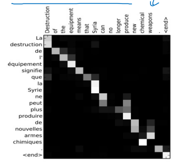
56

## Attention examples

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

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

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




Andrew Ng

57