Lecture Notes on

Pattern Recognition

Module 09(D): Recurrent Neural Network (RNN)

Gang Li
School of Information Technology
Deakin University, VIC 3125, Australia

0

Road map

1

Recurrent Neural Network

- Why RNN?
- RNN
- RNN Example

2

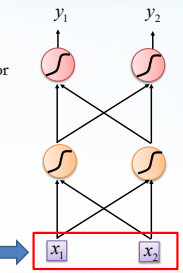
Application: Booking System

- Slot Filling

3

Application: Booking System

- Solving slot filling by FFN?
  - Input:
    - Word represented as a vector



4

One Hot Encoding

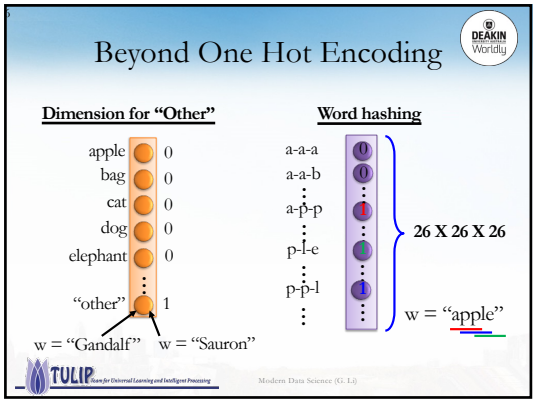
- The vector is lexicon size.
- Each dimension corresponds to a word in the lexicon
- The dimension for the word is 1, and others are 0

lexicon = {apple, bag, cat, dog, elephant}

apple = [1 0 0 0 0]
bag = [0 1 0 0 0]
cat = [0 0 1 0 0]
dog = [0 0 0 1 0]
elephant = [0 0 0 0 1]

5

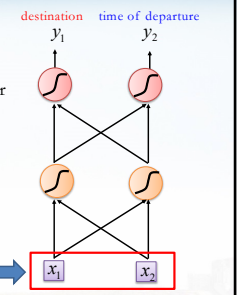
Beyond One Hot Encoding



6

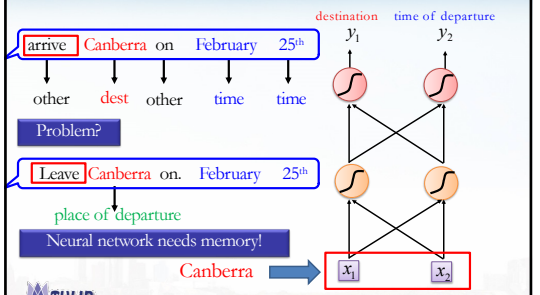
Application: Booking System

- Solving slot filling by FFN?
  - Input:
    - Word represented as a vector
  - Output:
    - Probability distribution that the input word belonging to the slots



7

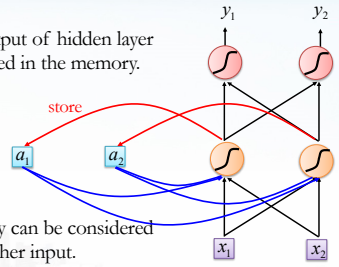
Example Application



8

Recurrent Neural Network (RNN)

The output of hidden layer are stored in the memory.



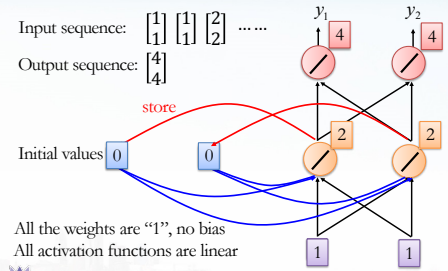
Memory can be considered as another input.

9

Example

Input sequence: $\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}2\\2\end{bmatrix}\cdots\cdots$

Output sequence: $\begin{bmatrix}4\\4\end{bmatrix}$



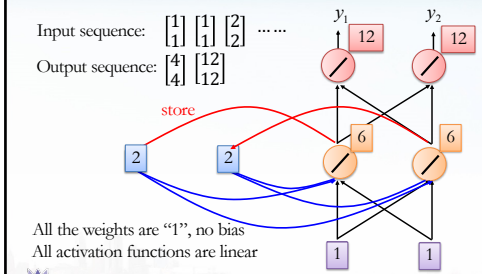
All the weights are "1", no bias
All activation functions are linear

10

Example

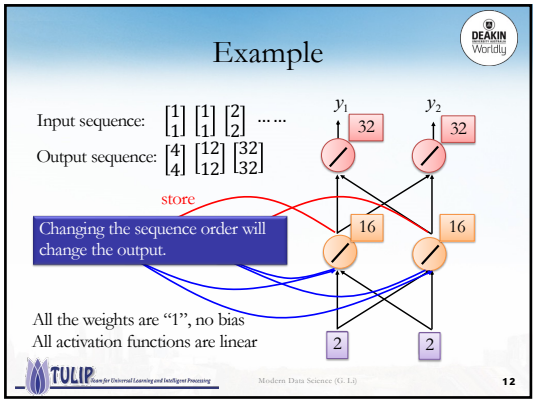
Input sequence: $\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}2\\2\end{bmatrix}\cdots\cdots$

Output sequence: $\begin{bmatrix}4\\4\end{bmatrix}\begin{bmatrix}12\\12\end{bmatrix}$

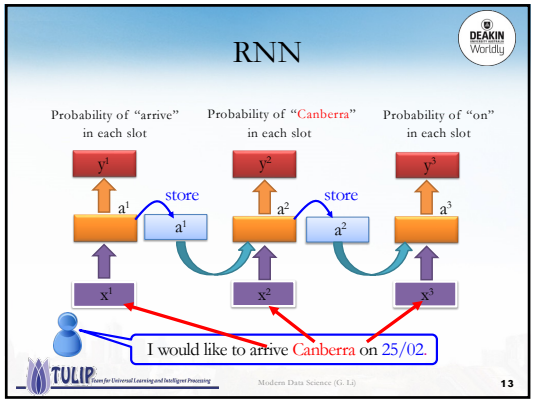


All the weights are "1", no bias
All activation functions are linear

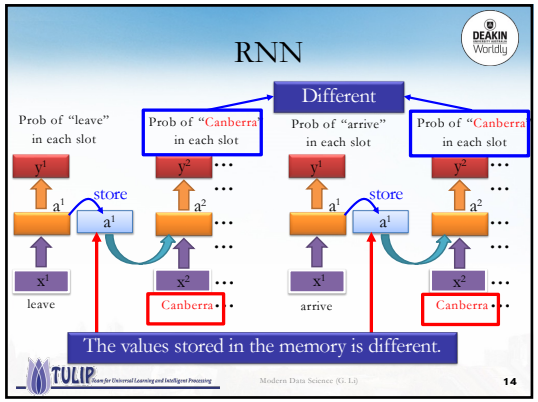
11



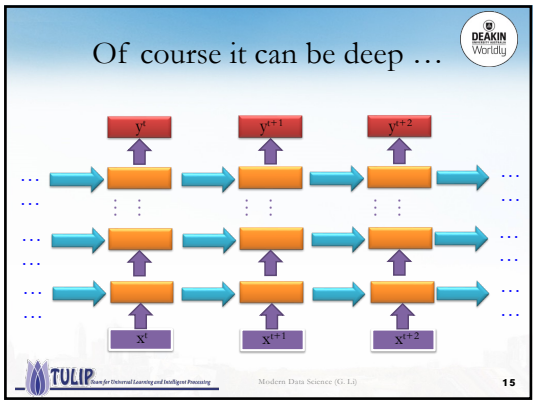
12



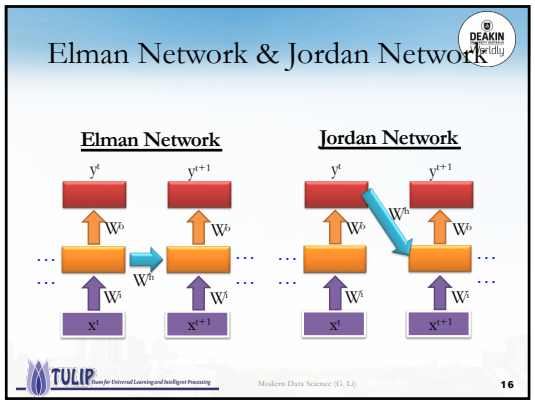
13



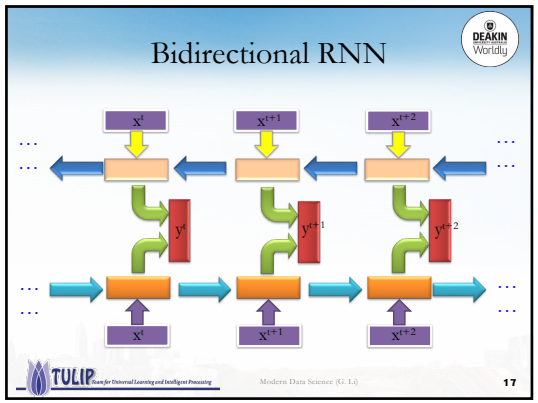
14



15



16



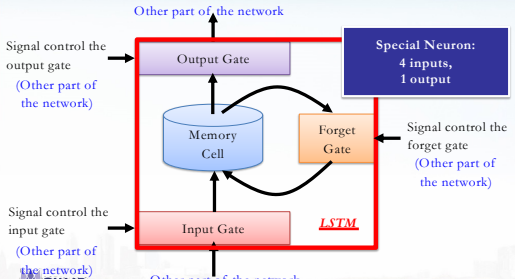
17

LSTM

- Why LSTM?
- LSTM
- LSTM and FFN

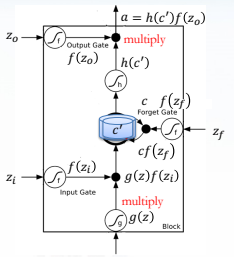
18

Long Short-Term Memory (LSTM)



19

Long Short-Term Memory (LSTM)



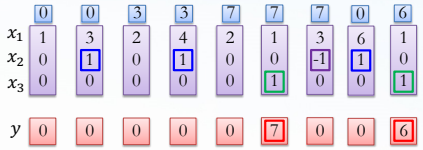
Activation function f is usually a sigmoid function

- Between 0 and 1
- Mimic open and close gate

$$c' = g(z)f(z_i) + cf(z_f)$$

20

LSTM - Example



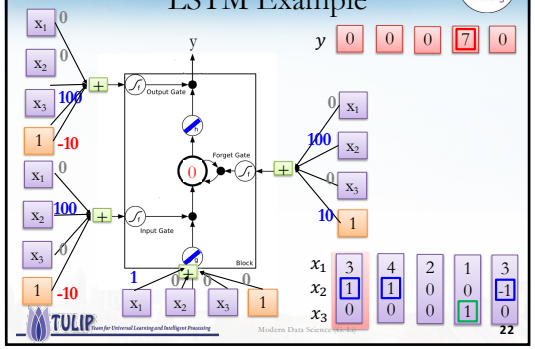
When $x_2 = 1$, add the numbers of x_1 into the memory

When $x_2 = -1$, reset the memory

When $x_3 = 1$, output the number in the memory.

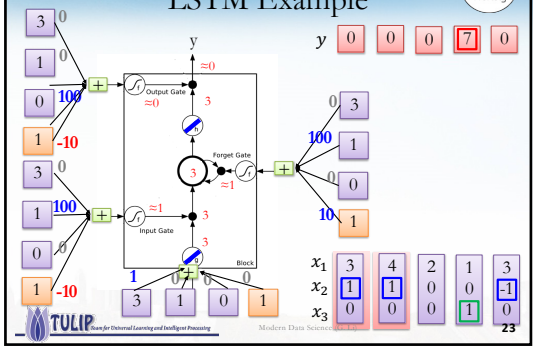
21

LSTM Example

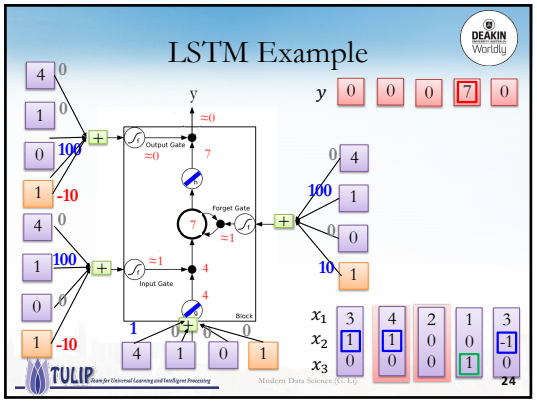


22

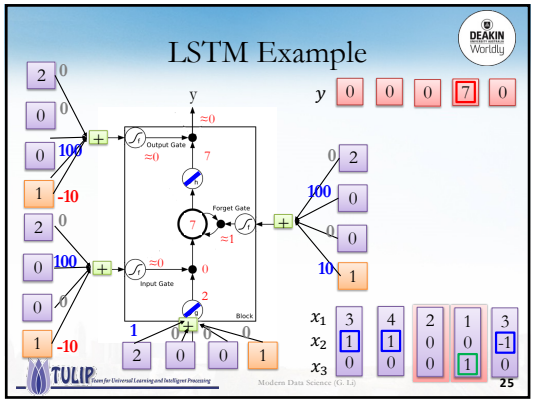
LSTM Example



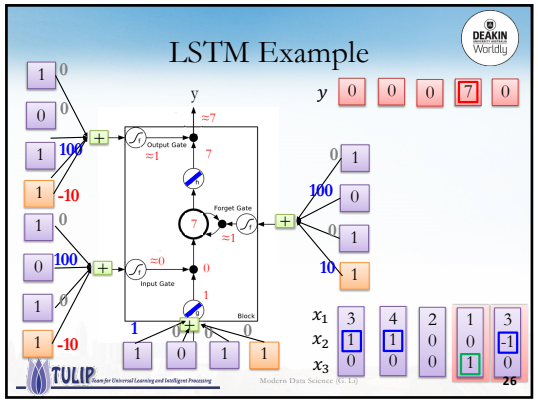
23



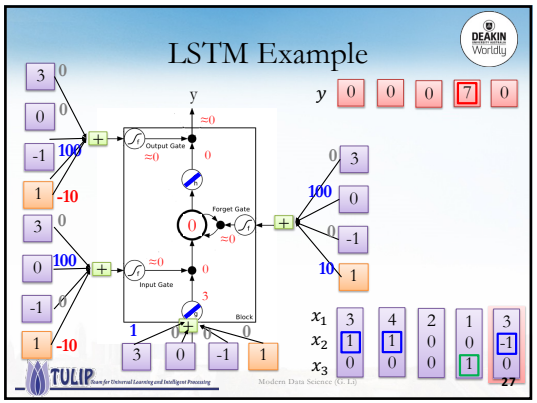
24



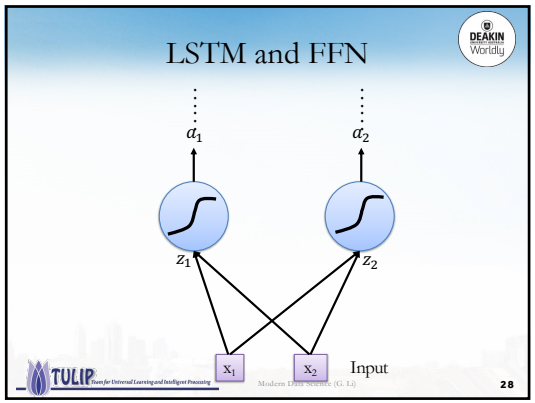
25



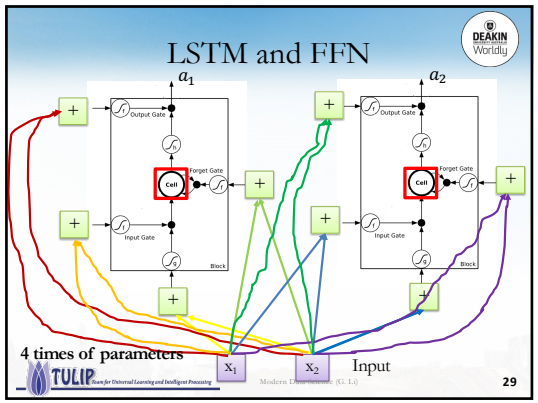
26



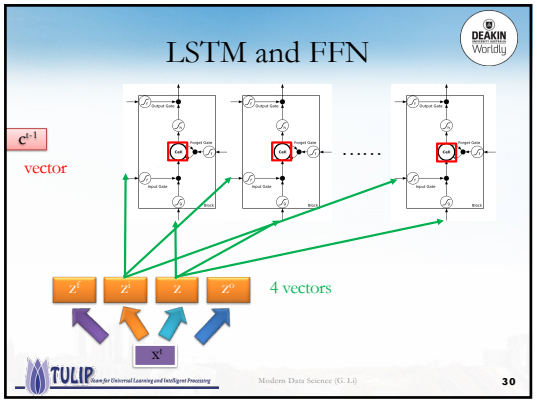
27



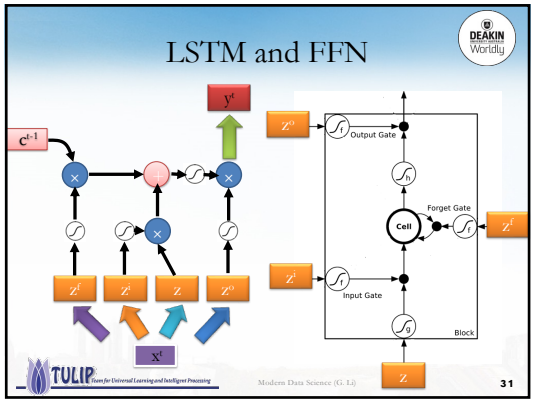
28



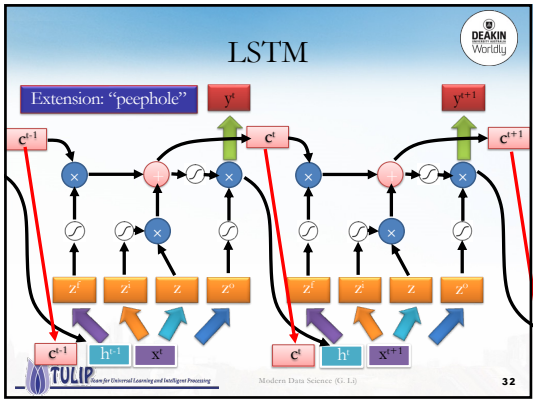
29



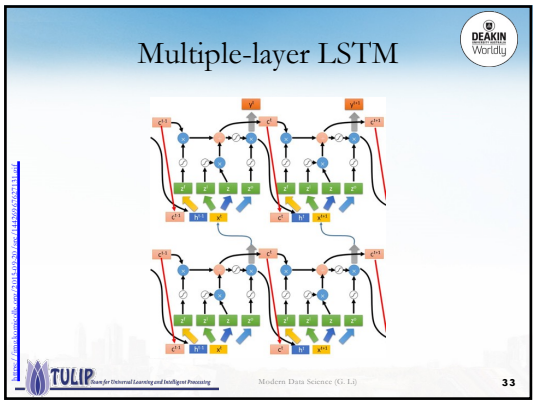
30



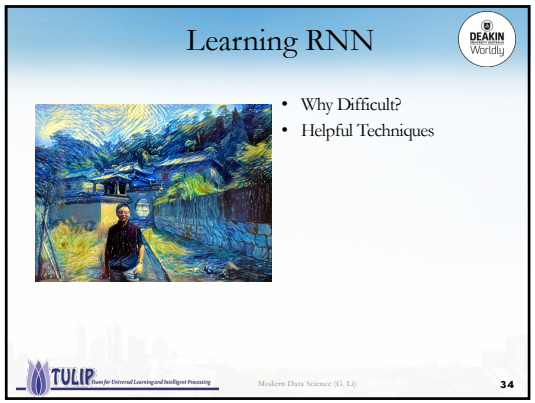
31



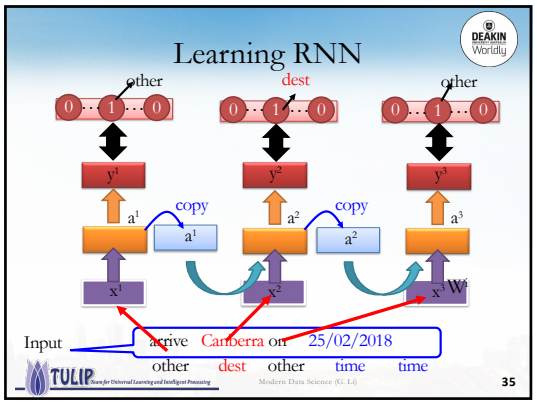
32



33

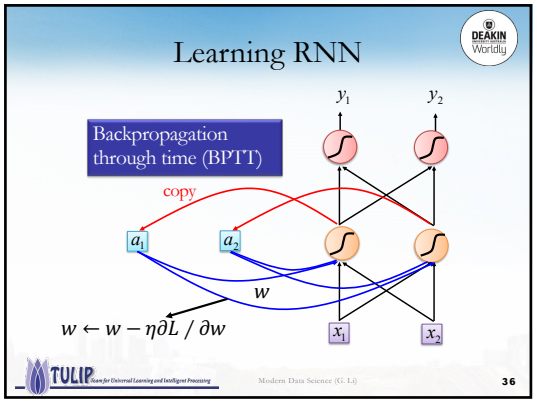


34



35

Learning RNN

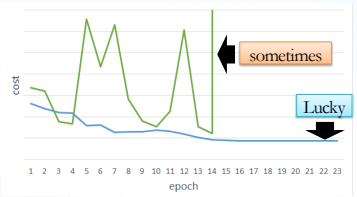


36

Learning RNN

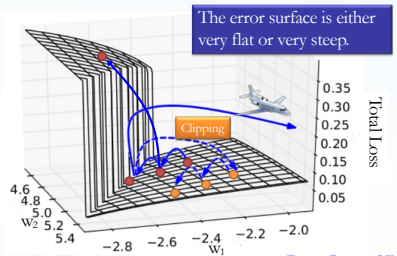
- RNN-based network is not always easy to learn

Real experiments on Language modeling



37

Learning RNN



[Razvan Pascanu, ICML'13]

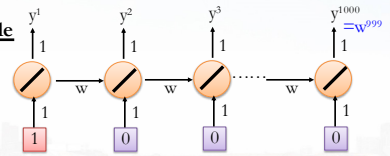
38

Why Difficult?

$w = 1 \Rightarrow y^{1000} = 1$
$w = 1.01 \Rightarrow y^{1000} \approx 20000$
Large $\partial L/\partial w$ → Small Learning rate?

$w = 0.99 \Rightarrow y^{1000} \approx 0$
$w = 0.01 \Rightarrow y^{1000} \approx 0$
small $\partial L/\partial w$ → Large Learning rate?

Toy Example

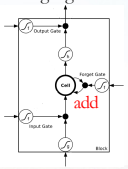


39

Helpful Techniques

- Long Short-term Memory (LSTM)
  - Can deal with gradient vanishing (not gradient explode)
  - Memory and input are **added**
    - The influence never disappears unless forget gate is closed

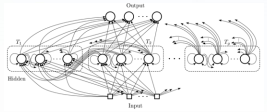
No Gradient vanishing
(If forget gate is opened.)



40

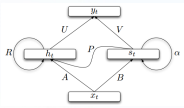
Helpful Techniques

Clockwise RNN



[Jan Koutnik, JMLR'14]

Structurally Constrained Recurrent Network (SCRN)



[Tomas Mikolov, ICLR'15]

Vanilla RNN Initialized with Identity matrix + ReLU activation function
[Quoc V. Le, arXiv'15]
- Outperform or be comparable with LSTM in 4 different tasks

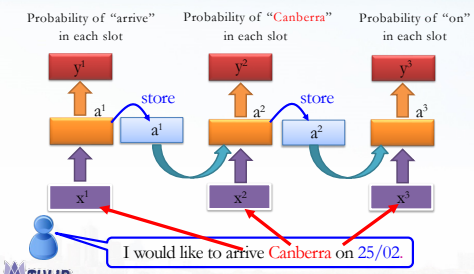
41

RNN/LSTM Applications

- Applications
- Attention-based Model
- Visual Question Answering

42

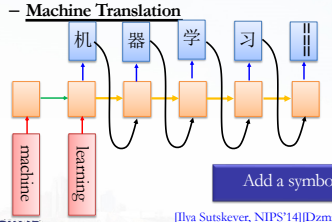
Applications: Many to Many



43

Many to Many (No Limitation)

- Both input and output are both sequences ***with different lengths.***
  → ***Sequence to sequence learning***
  – **Machine Translation**



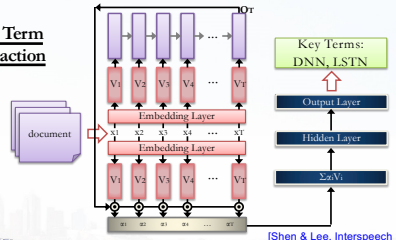
[Ilya Sutskever, NIPS'14][Dzmitry Bahdanau, arXiv'15]

44

Applications: Many to One

- Input is a vector sequence, but output is only one vector

Key Term Extraction

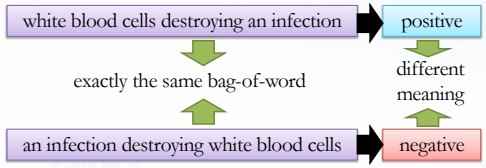


[Shen & Lee, Interspeech 16]

45

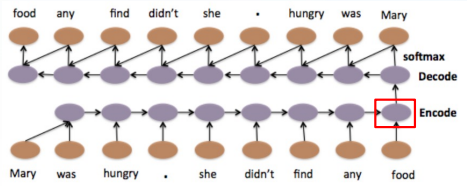
Sequence-to-sequence Auto-encoder - Text

- To understand the meaning of a word sequence, the order of the words can not be ignored.



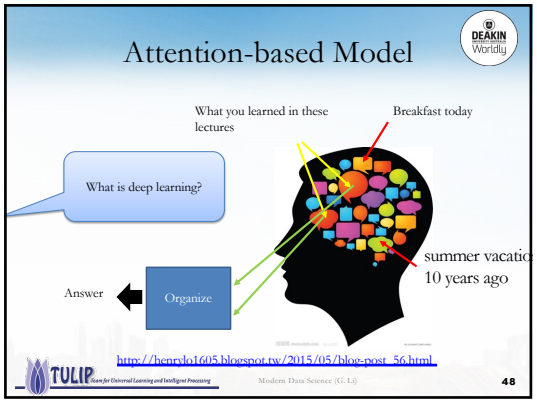
46

Sequence-to-sequence Auto-encoder - Text

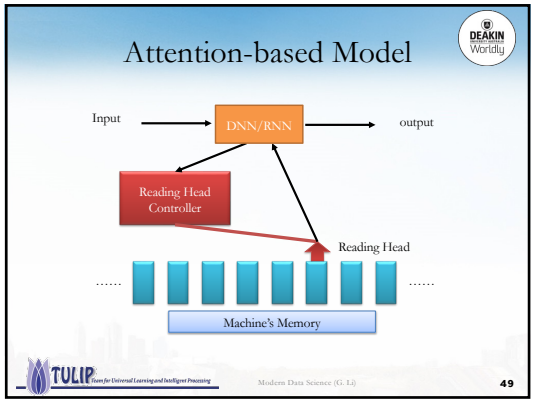


- Li, Jiwei, Minh-Thang Luong, and Dan Jurafsky. "A hierarchical neural autoencoder for paragraphs and documents." *arXiv preprint arXiv:1506.01057*(2015).

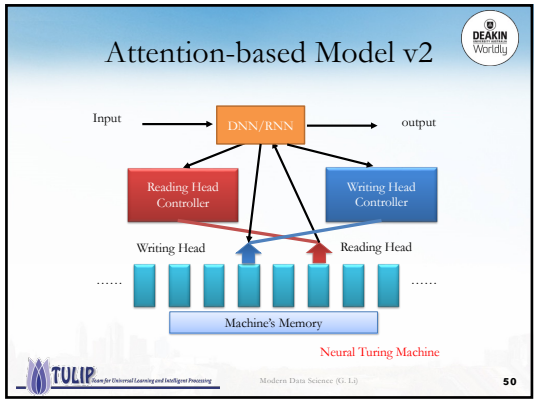
47



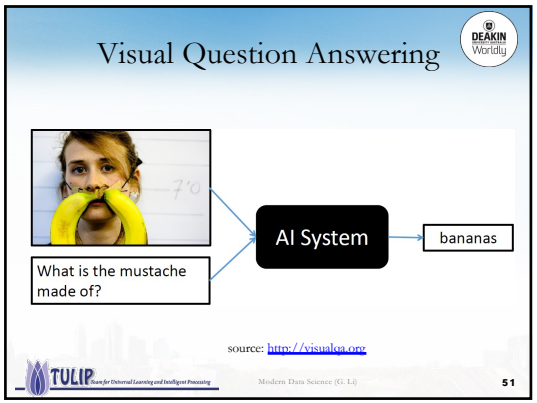
48



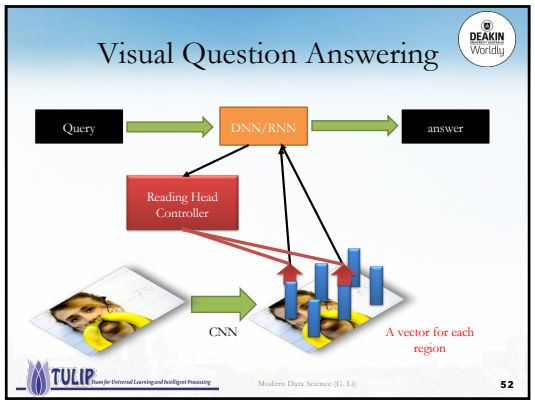
49



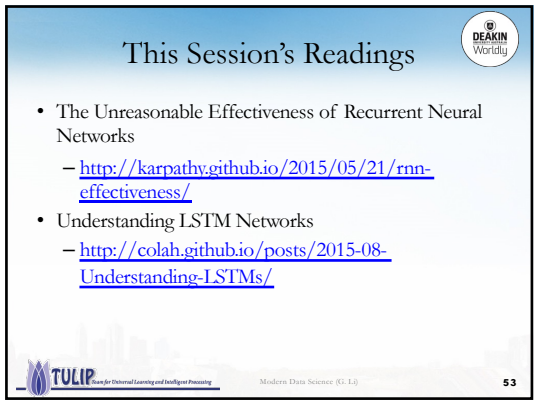
50



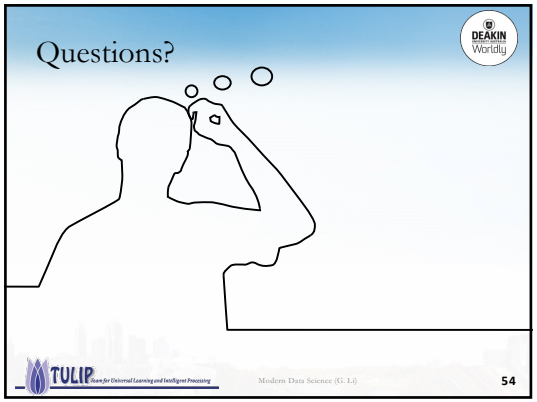
51



52



53



54