

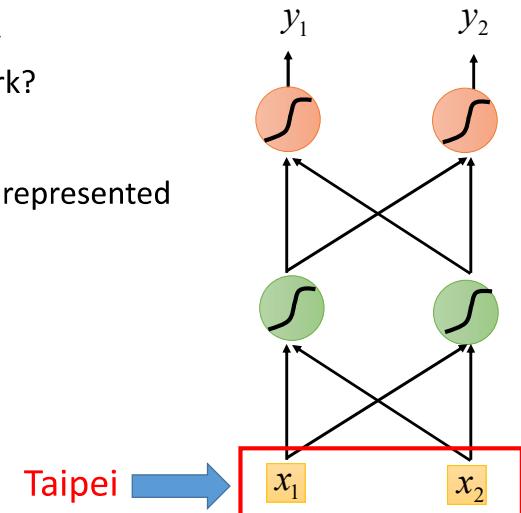
Recurrent Neural Network (RNN)

Example Application

Solving slot filling by
Feedforward network?

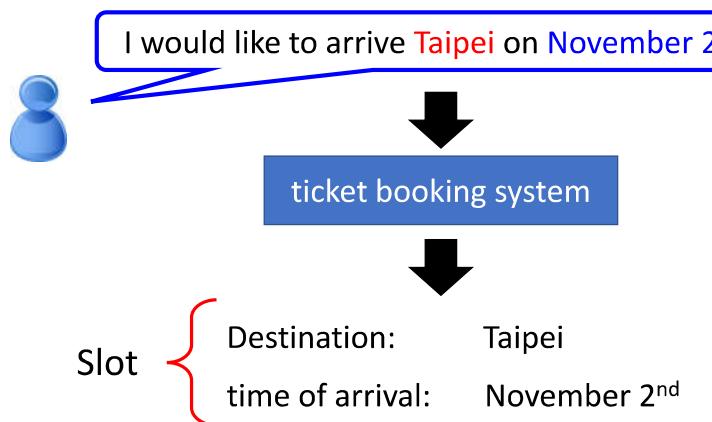
Input: a word

(Each word is represented
as a vector)



Example Application

- Slot Filling



1-of-N encoding

How to represent each word as a vector?

1-of-N Encoding lexicon = {apple, bag, cat, dog, elephant}

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

apple = [1 0 0 0 0]

bag = [0 1 0 0 0]

cat = [0 0 1 0 0]

dog = [0 0 0 1 0]

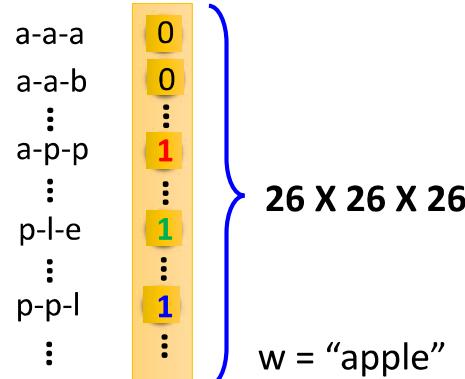
elephant = [0 0 0 0 1]

Beyond 1-of-N encoding

Dimension for “Other”

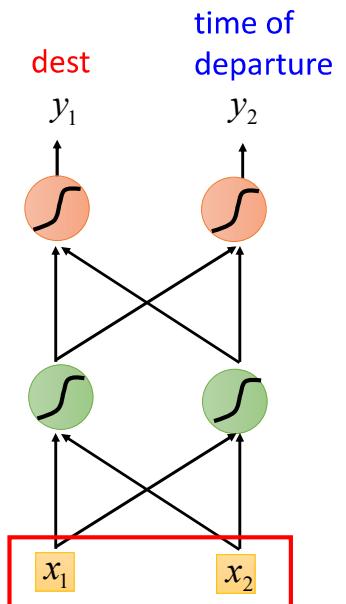
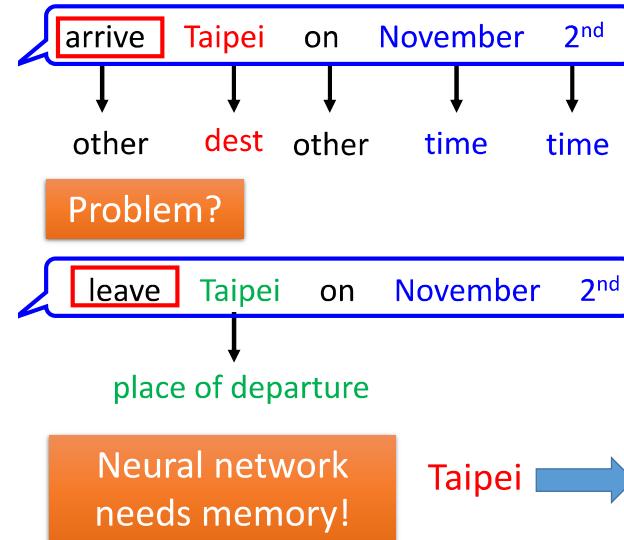
apple	0
bag	0
cat	0
dog	0
elephant	0
“other”	1
w = “Gandalf”	w = “Sauron”

Word hashing



5

Example Application



Example Application

Solving slot filling by Feedforward network?

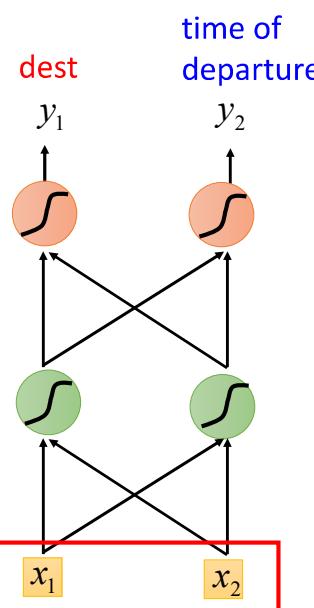
Input: a word

(Each word is represented as a vector)

Output:

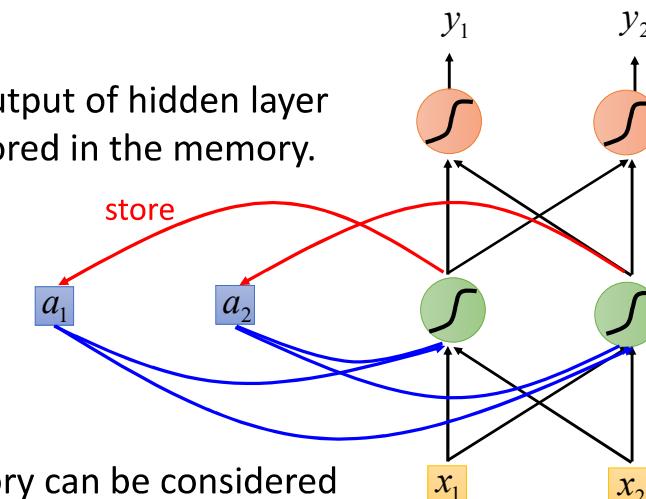
Probability distribution that the input word belonging to the slots

Taipei



Recurrent Neural Network (RNN)

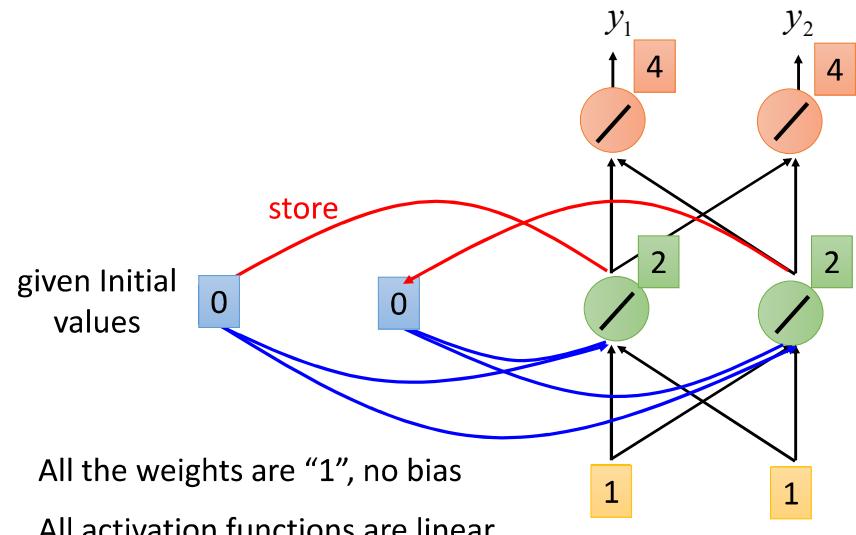
The output of hidden layer are stored in the memory.



Memory can be considered as another input.

Example

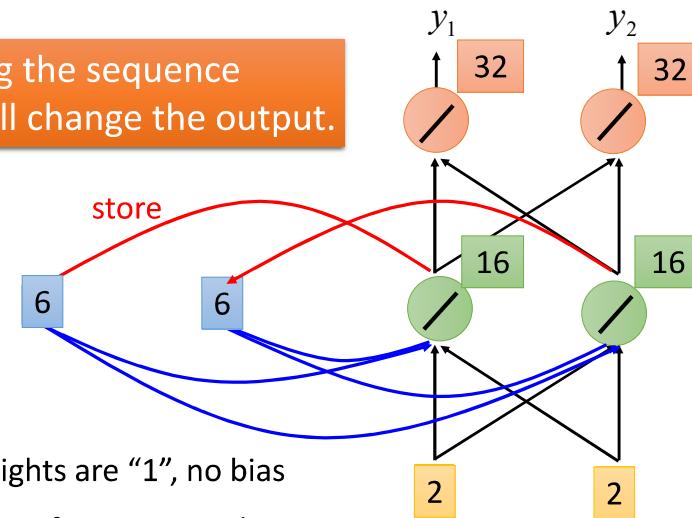
Input sequence: $\begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} \dots \dots$
 output sequence: $\begin{bmatrix} 4 \\ 4 \end{bmatrix}$



Example

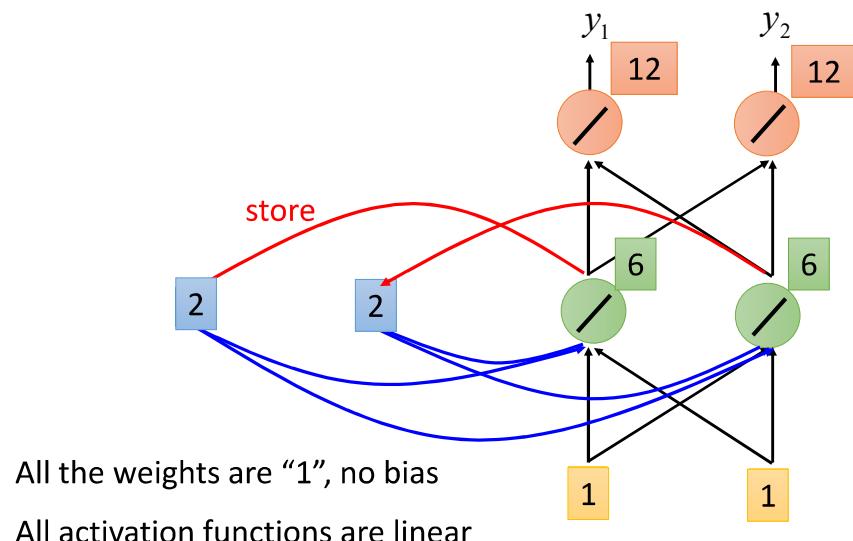
Input sequence: $\begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} \dots \dots$
 output sequence: $\begin{bmatrix} 4 \\ 4 \end{bmatrix} \begin{bmatrix} 12 \\ 12 \end{bmatrix} \begin{bmatrix} 32 \\ 32 \end{bmatrix}$

Changing the sequence order will change the output.



Example

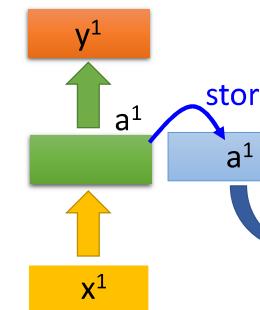
Input sequence: $\begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} \dots \dots$
 output sequence: $\begin{bmatrix} 4 \\ 4 \end{bmatrix} \begin{bmatrix} 12 \\ 12 \end{bmatrix}$



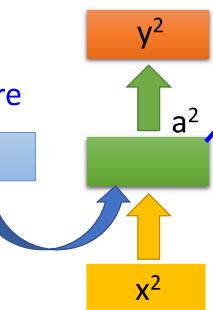
RNN

The same network is used again and again.

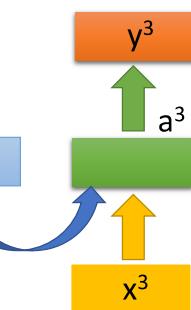
Probability of “arrive” in each slot



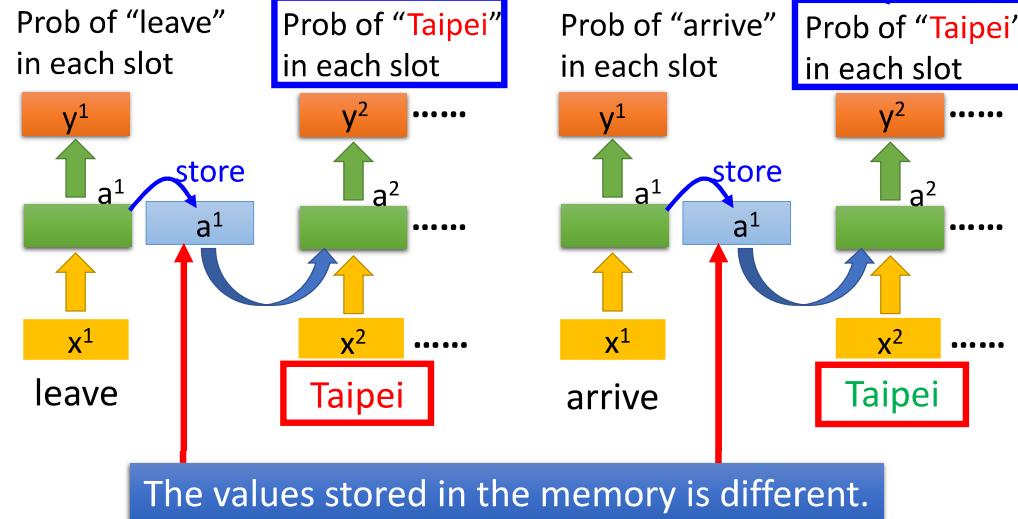
Probability of “Taipei” in each slot



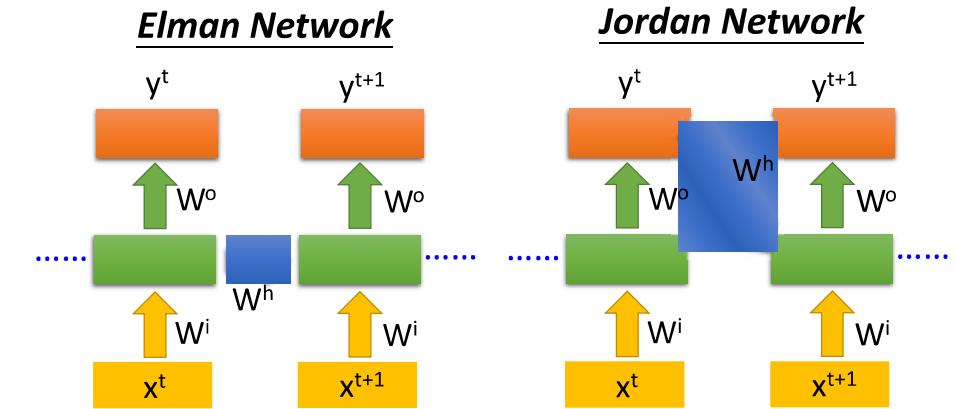
Probability of “on” in each slot



RNN

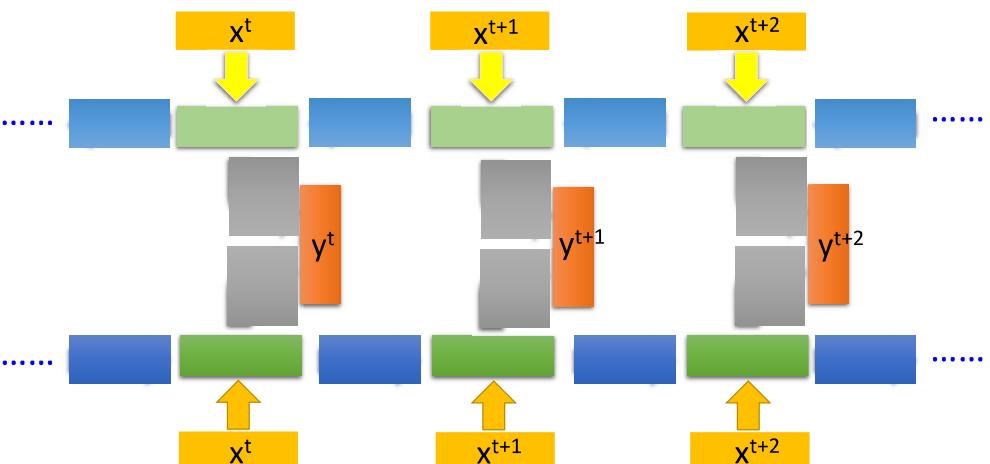
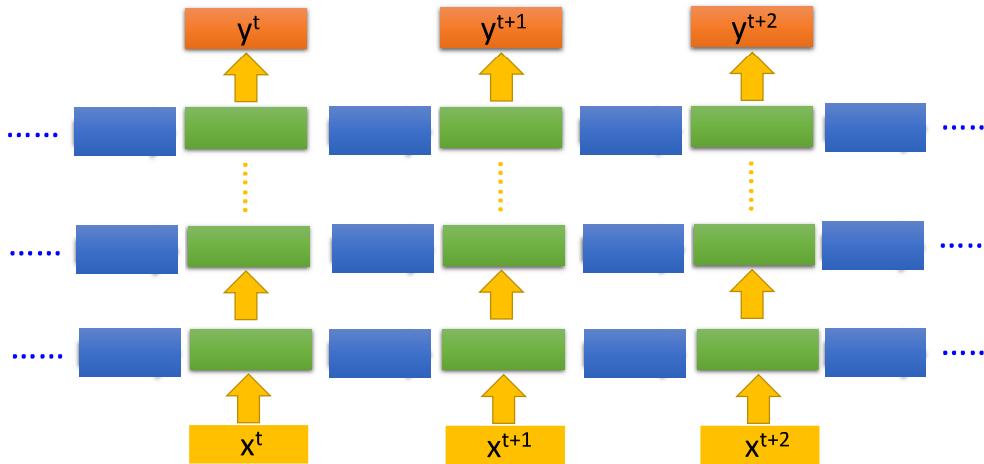


Elman Network & Jordan Network

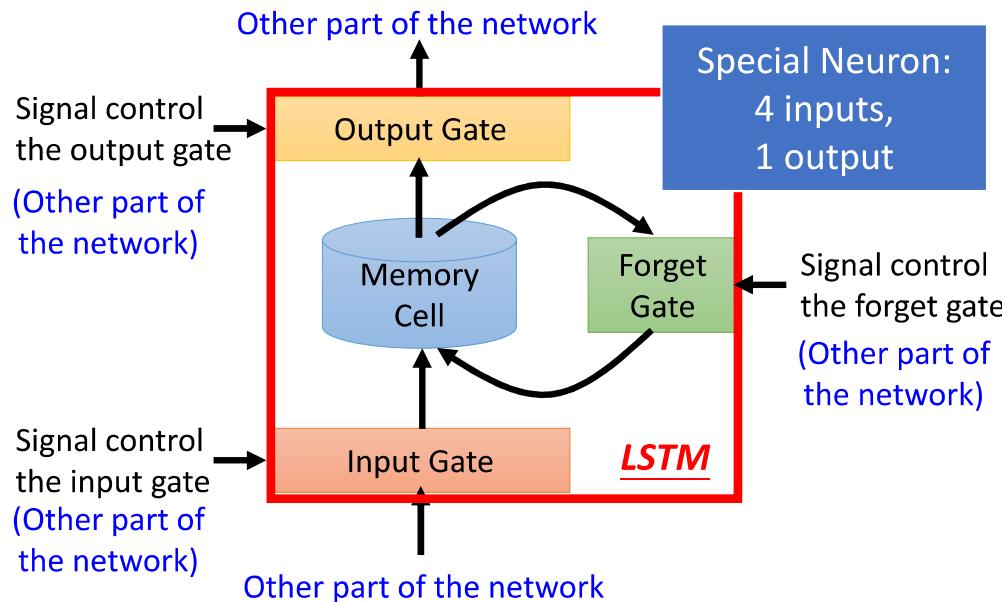


Of course it can be deep ...

Bidirectional RNN



Long Short-term Memory (LSTM)



LSTM - Example

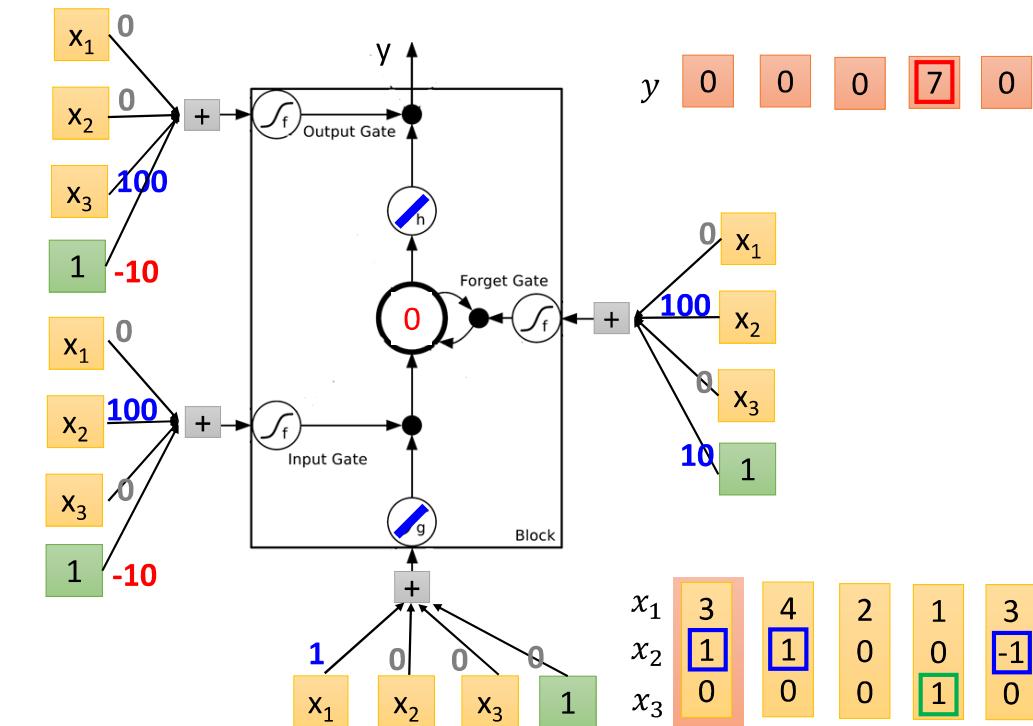
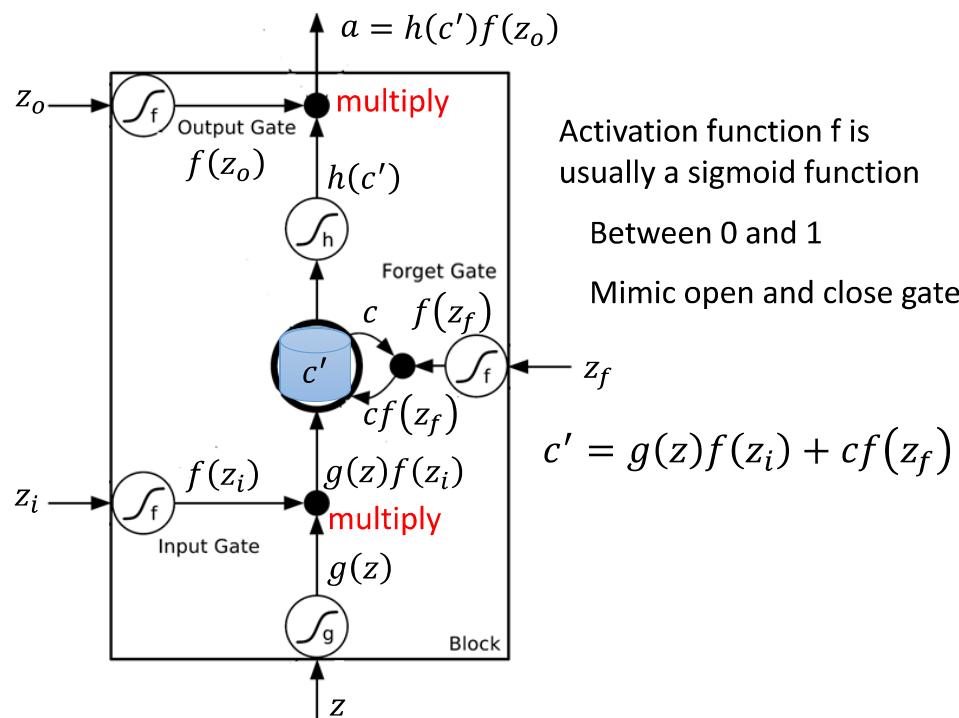
x_1	0	0	3	2	3	7	7	7	0	6
x_2	1	3	1	0	1	0	1	-1	1	0
x_3	0	0	0	0	0	0	1	0	1	1

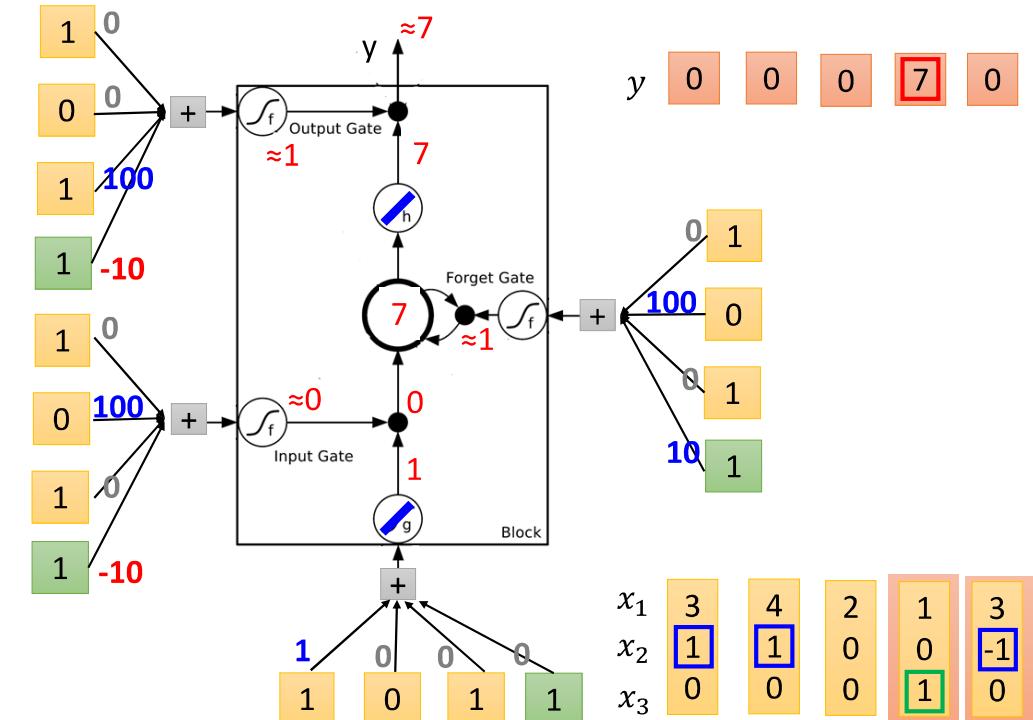
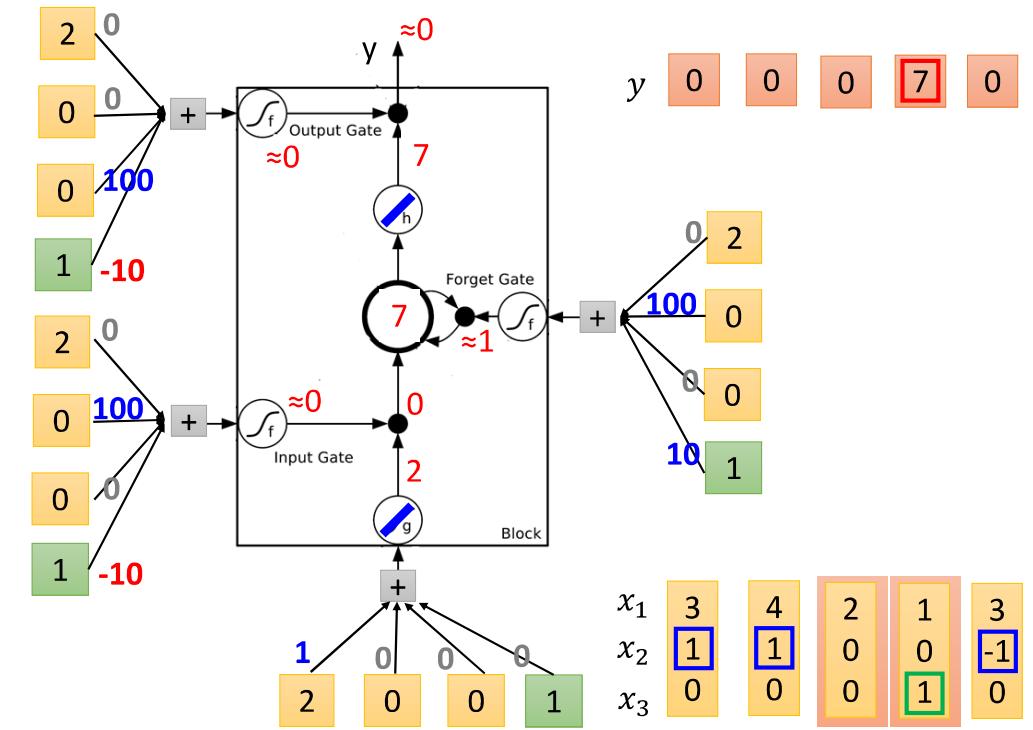
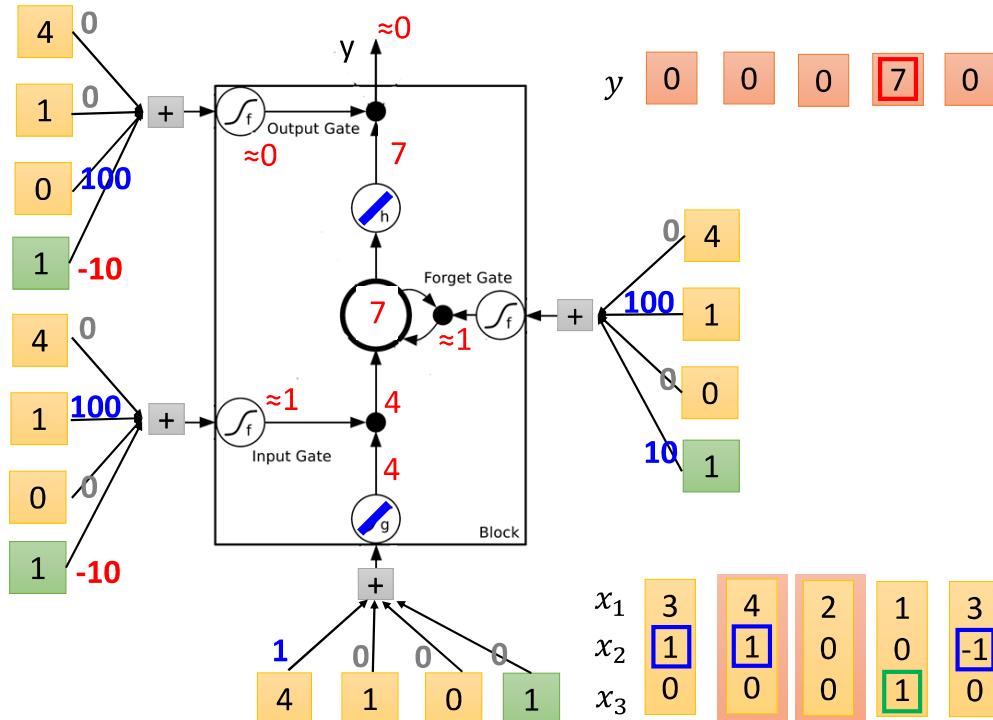
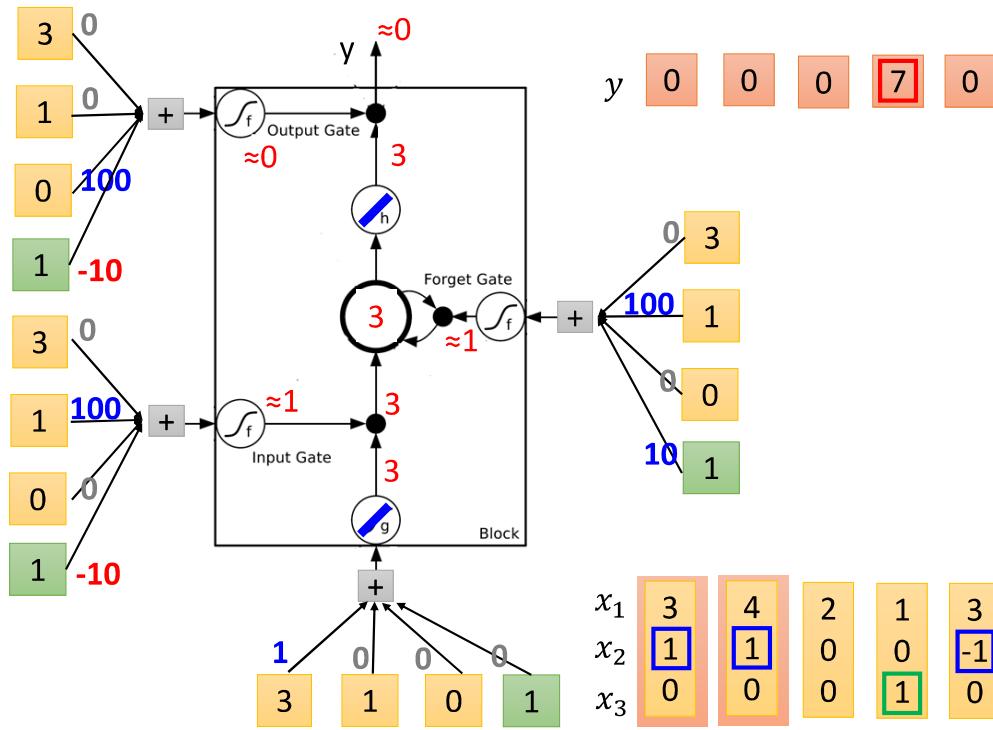
y	0	0	0	0	0	7	0	0	6
-----	---	---	---	---	---	---	---	---	---

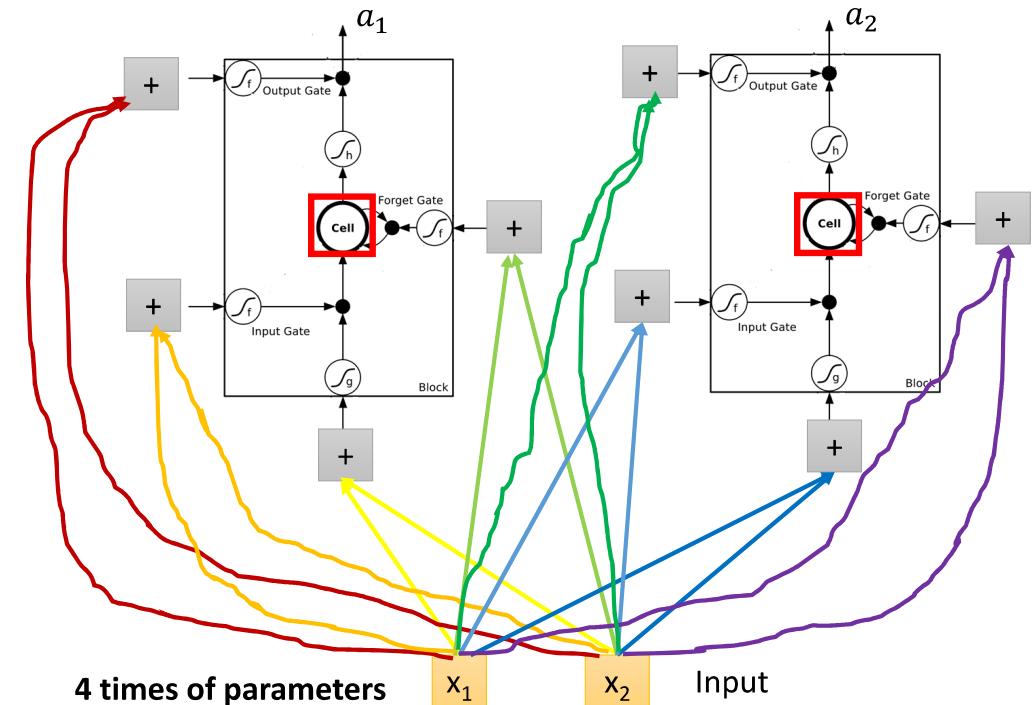
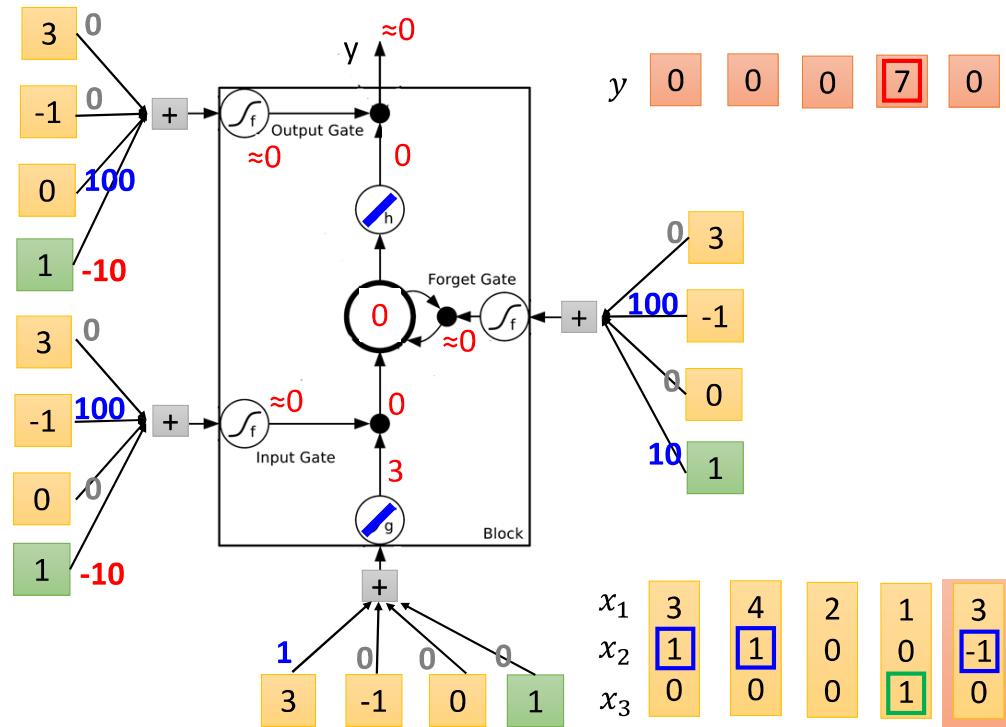
When $x_2 = 1$, add the numbers of x_1 into the memory

When $x_2 = -1$, reset the memory

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

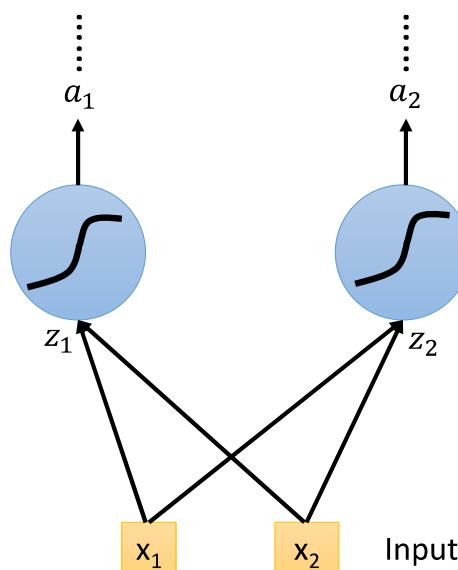




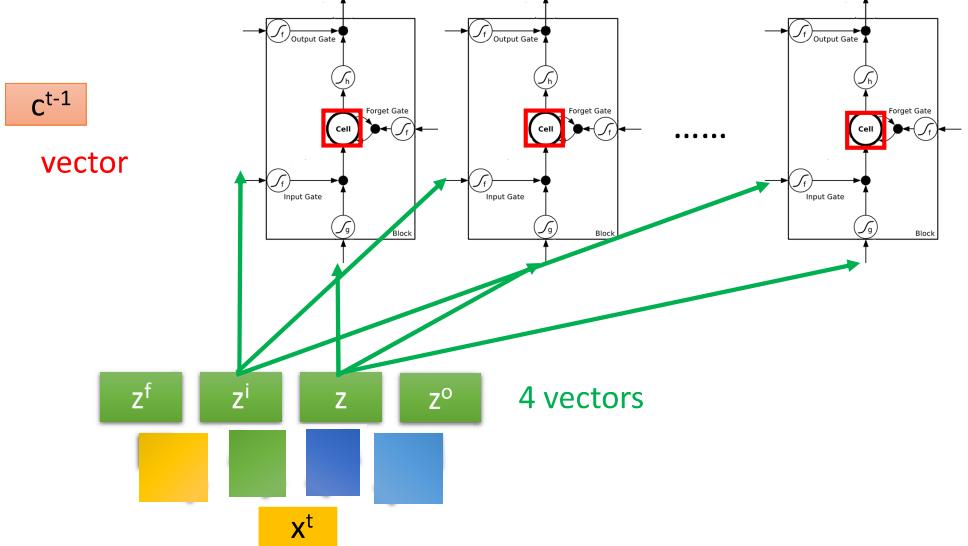


Original Network:

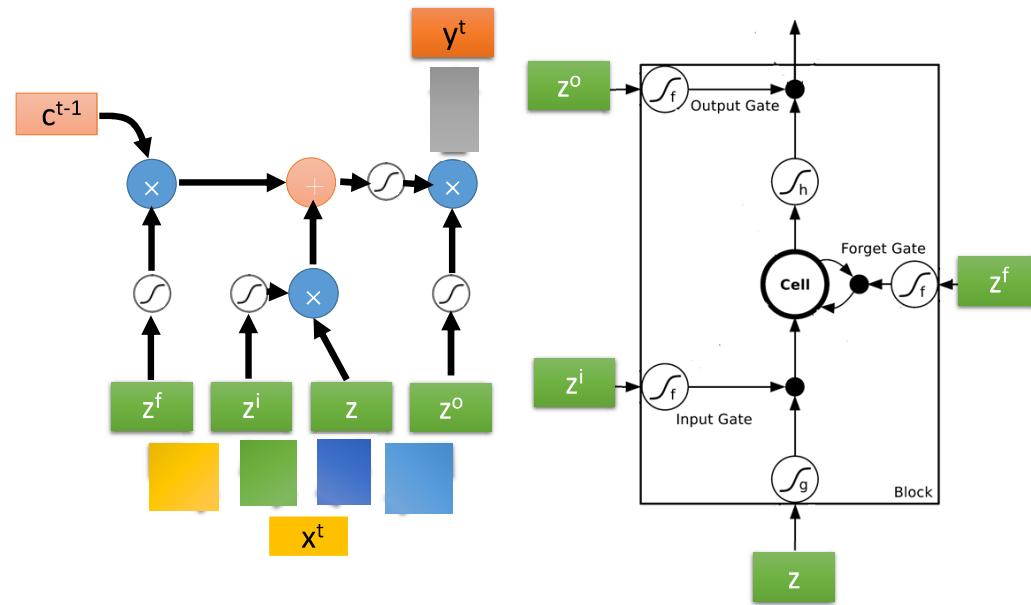
➤ Simply replace the neurons with LSTM



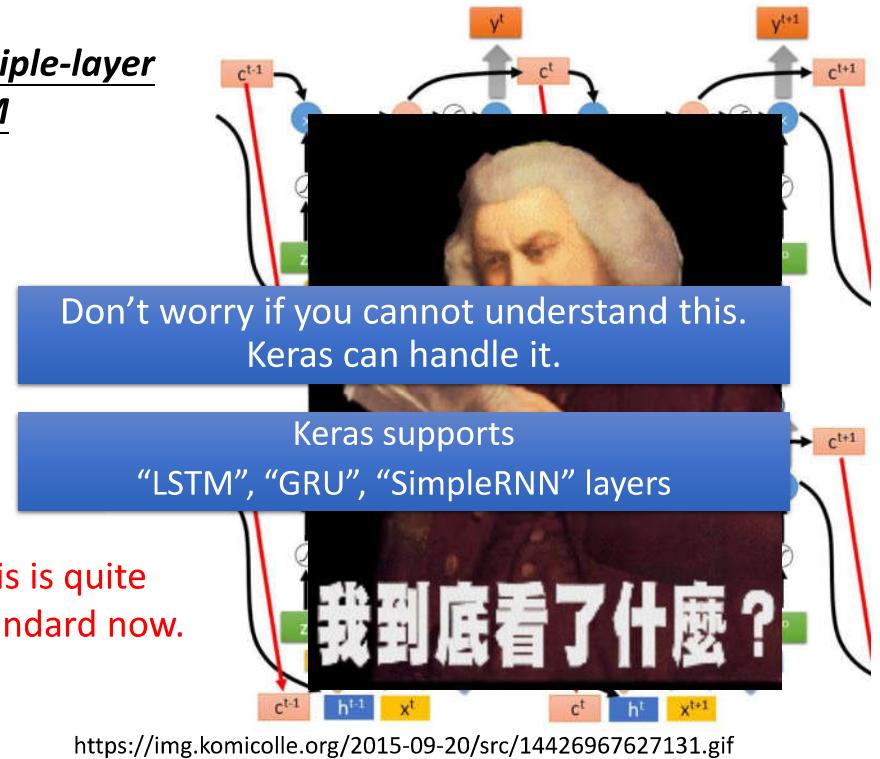
LSTM



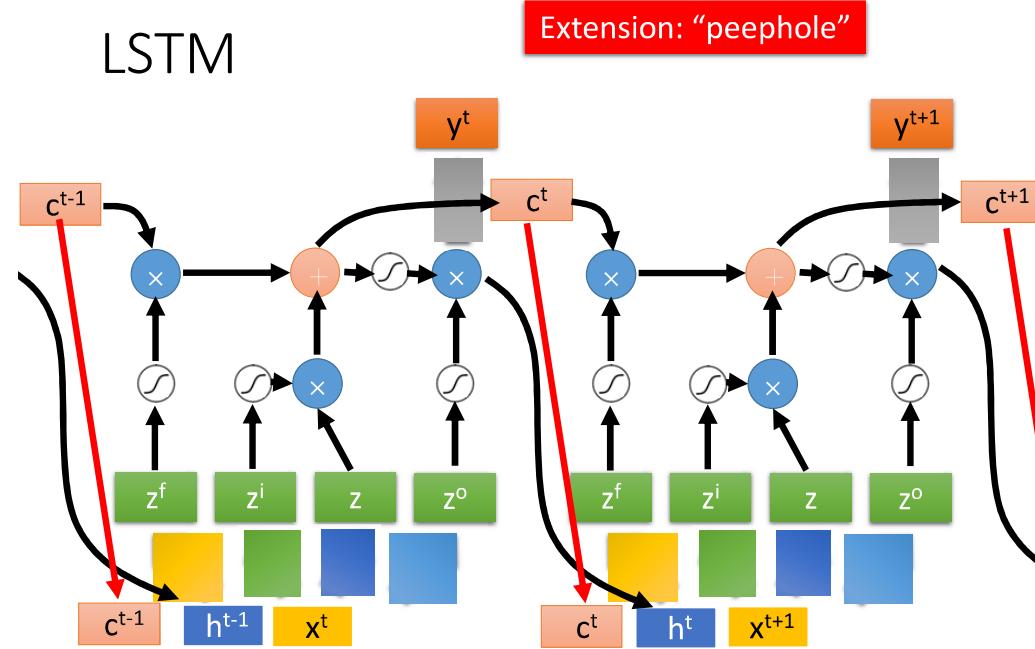
LSTM



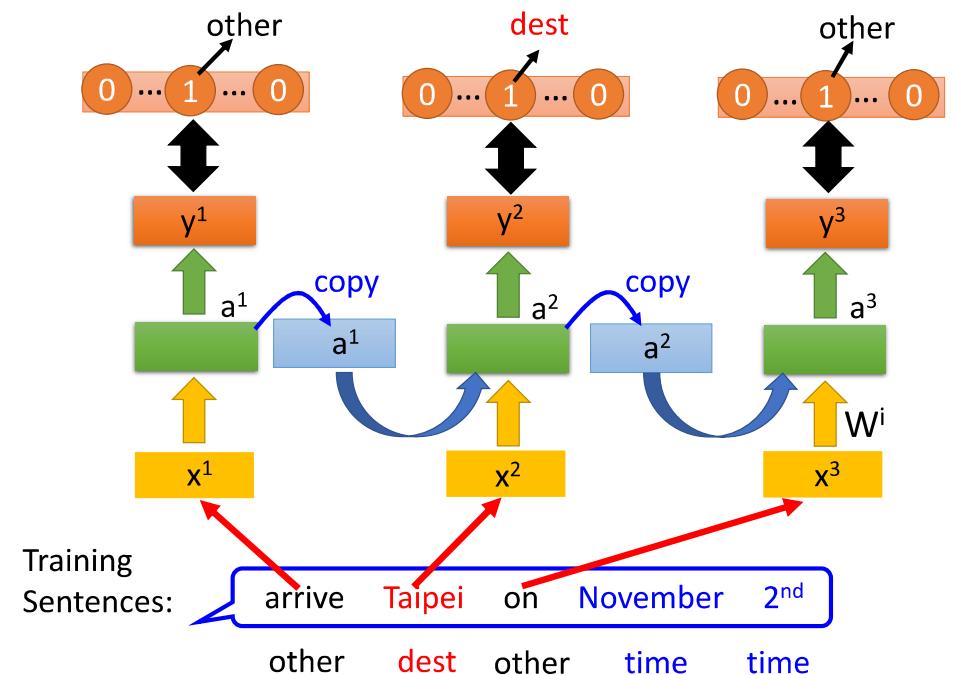
Multiple-layer LSTM



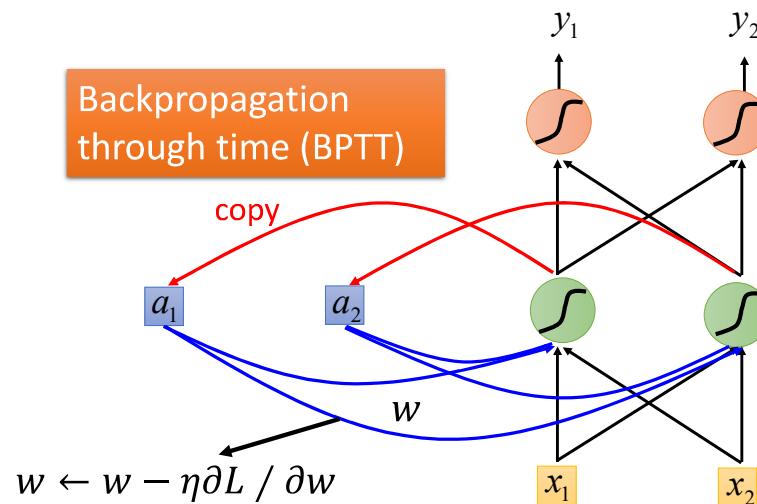
LSTM



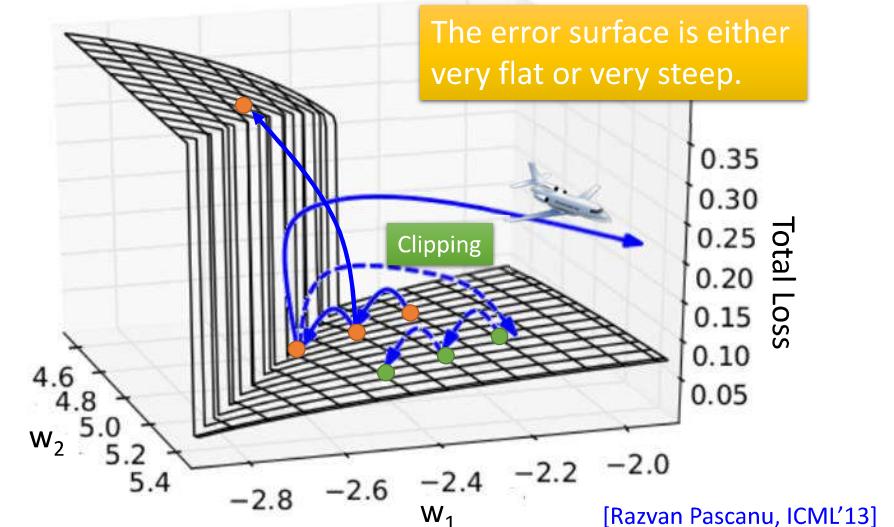
Learning Target



Learning



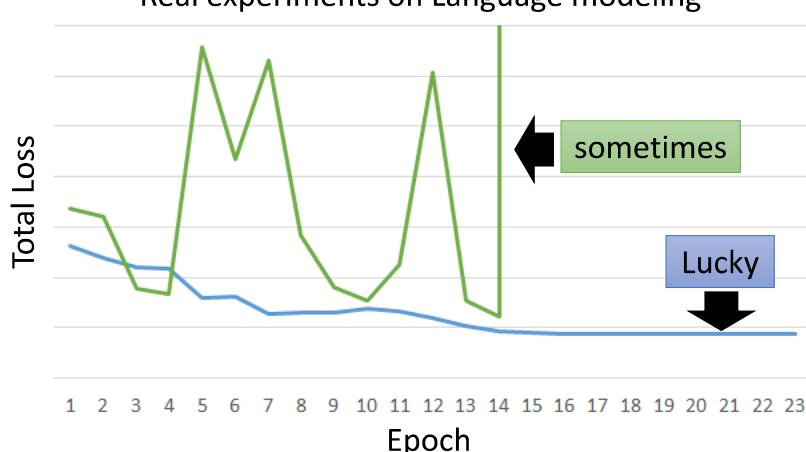
The error surface is rough.



Unfortunately

- RNN-based network is not always easy to learn

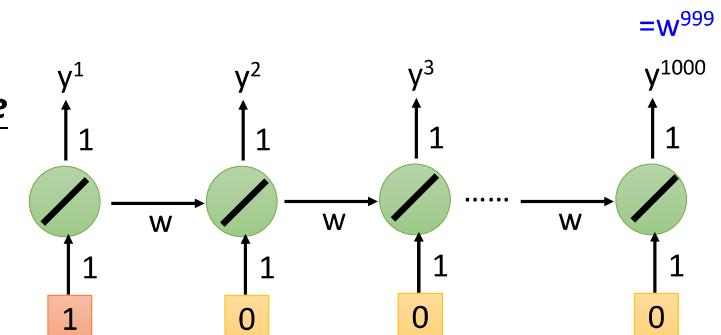
Real experiments on Language modeling



Why?

$w = 1$	$y^{1000} = 1$	Large $\partial L / \partial w$	Small Learning rate?
$w = 1.01$	$y^{1000} \approx 20000$		
$w = 0.99$	$y^{1000} \approx 0$	small $\partial L / \partial w$	Large Learning rate?
$w = 0.01$	$y^{1000} \approx 0$		

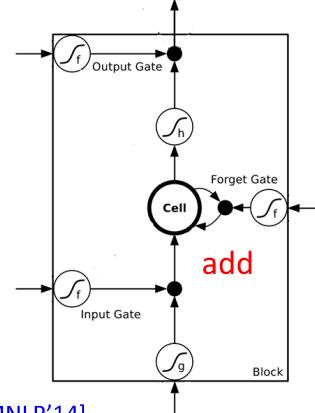
Toy Example



Helpful Techniques

• Long Short-term Memory (LSTM)

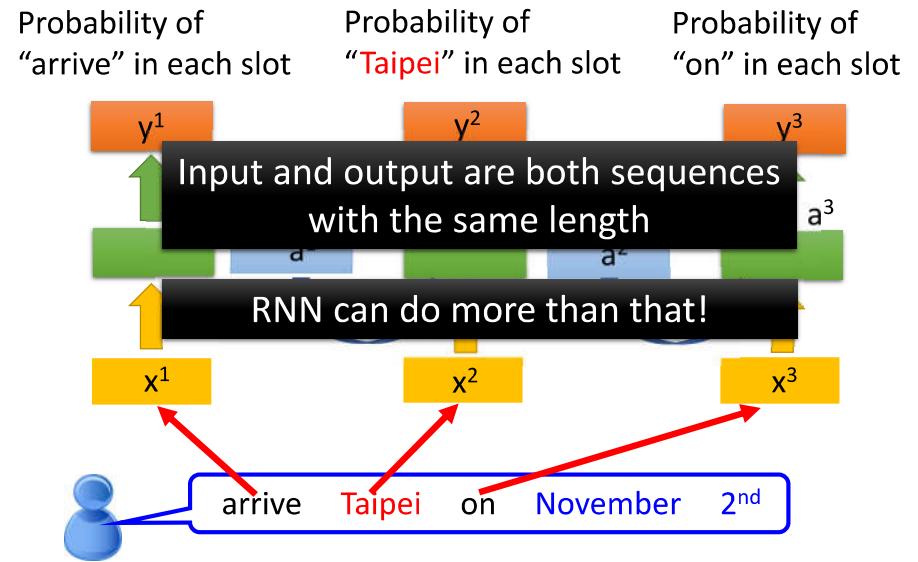
- Can deal with gradient vanishing (not gradient explode)
- Memory and input are added
- The influence never disappears unless forget gate is closed
- ➡ No Gradient vanishing (If forget gate is opened.)



Gated Recurrent Unit (GRU):
simpler than LSTM

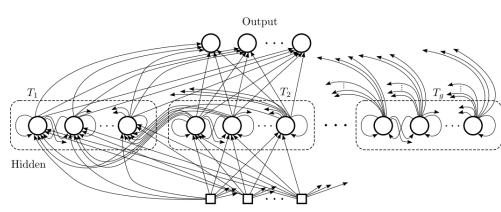
[Cho, EMNLP'14]

More Applications



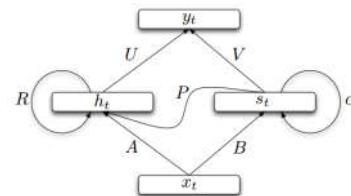
Helpful Techniques

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

Many to one

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

Sentiment Analysis

看了這部電影覺得很開心

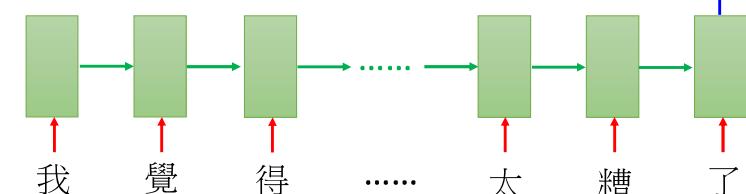
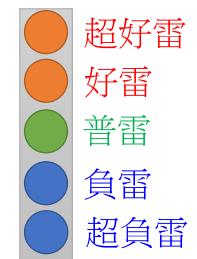
Positive (正雷)

這部電影太糟了

Negative (負雷)

這部電影很棒

Positive (正雷)

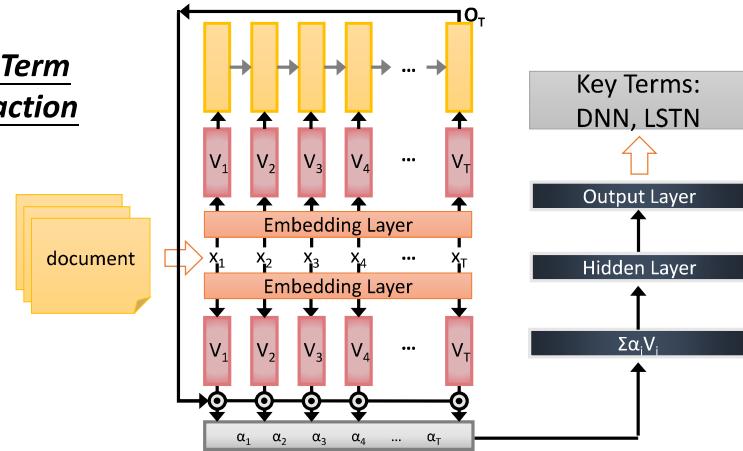


Many to one

[Shen & Lee, Interspeech 16]

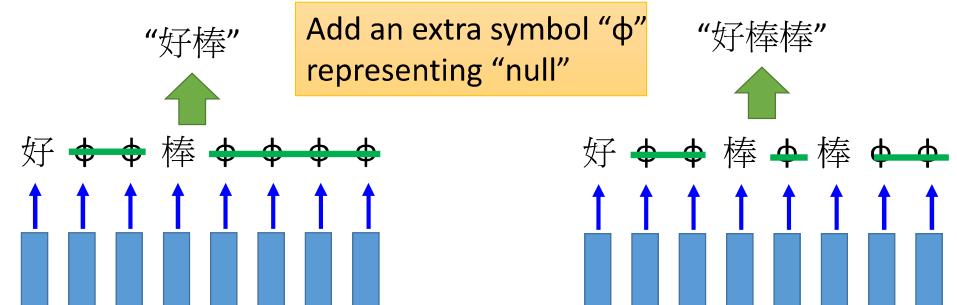
- Input is a vector sequence, but output is only one vector

Key Term Extraction



Many to Many (Output is shorter)

- Both input and output are both sequences, but the output is shorter.
- Connectionist Temporal Classification (CTC) [Alex Graves, ICML'06][Alex Graves, ICML'14][Hasim Sak, Interspeech'15][Jie Li, Interspeech'15][Andrew Senior, ASRU'15]



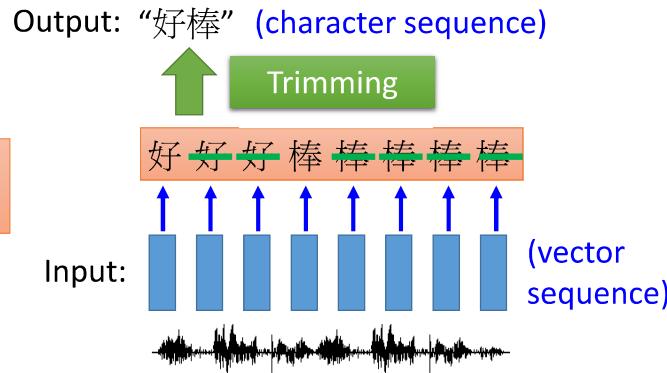
Many to Many (Output is shorter)

- Both input and output are both sequences, but the output is shorter.

- E.g. Speech Recognition

Problem?

Why can't it be
“好棒棒”



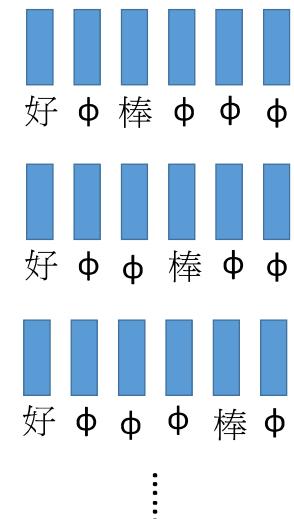
Many to Many (Output is shorter)

- CTC: Training

Acoustic Features:

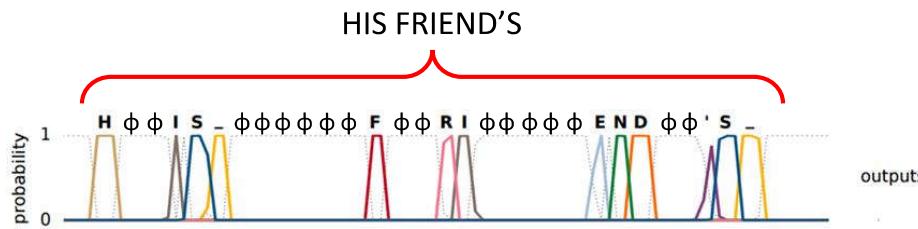
Label: 好 棒

All possible alignments are considered as correct.



Many to Many (Output is shorter)

- CTC: example

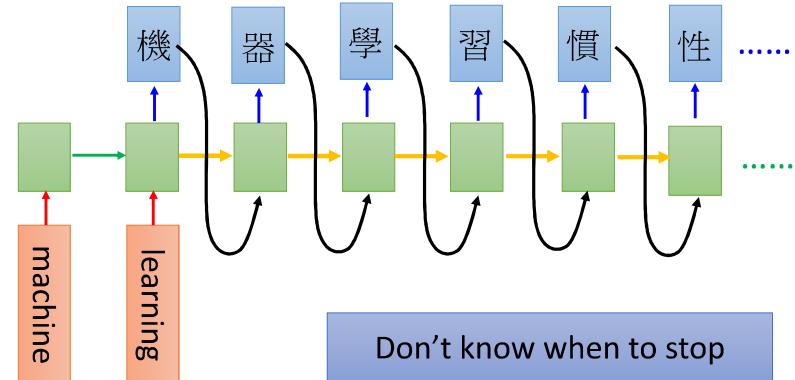


Graves, Alex, and Navdeep Jaitly. "Towards end-to-end speech recognition with recurrent neural networks." *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. 2014.

Many to Many (No Limitation)

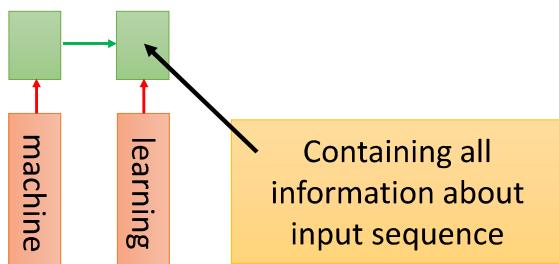
- Both input and output are both sequences with different lengths. → Sequence to sequence learning

- E.g. Machine Translation (machine learning→機器學習)



Many to Many (No Limitation)

- Both input and output are both sequences with different lengths. → Sequence to sequence learning
- E.g. Machine Translation (machine learning→機器學習)



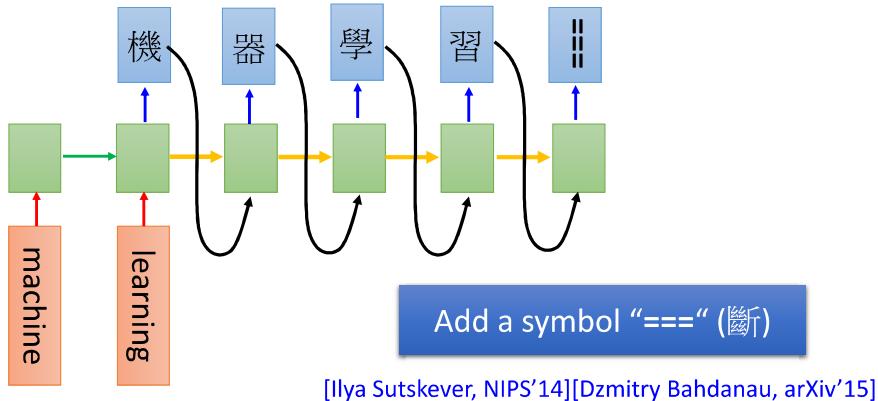
Many to Many (No Limitation)

推	:	超	06/12 10:39
推	n:	人	06/12 10:40
推	tion:	正	06/12 10:41
→	host:	大	06/12 10:47
推	:	中	06/12 10:59
推	403:	天	06/12 11:11
推	推	外	06/12 11:13
推	527:	飛	06/12 11:17
→	990b:	仙	06/12 11:32
→	512:	草	06/12 12:15
推 tlkagk: =====斷=====			

接龍推文是ptt在推文中的一種趣味玩法，與推齊有些類似但又有所不同，是指在推文中接續上一樓的字句，而推出連續的意思。該類玩法確切起源已不可知(鄉民百科)

Many to Many (No Limitation)

- Both input and output are both sequences with different lengths. → Sequence to sequence learning
 - E.g. Machine Translation (machine learning→機器學習)



<https://arxiv.org/pdf/1612.01744v1.pdf>

Many to Many (No Limitation)

- Both input and output are both sequences with different lengths. → Sequence to sequence learning
 - E.g. Machine Translation (machine learning→機器學習)

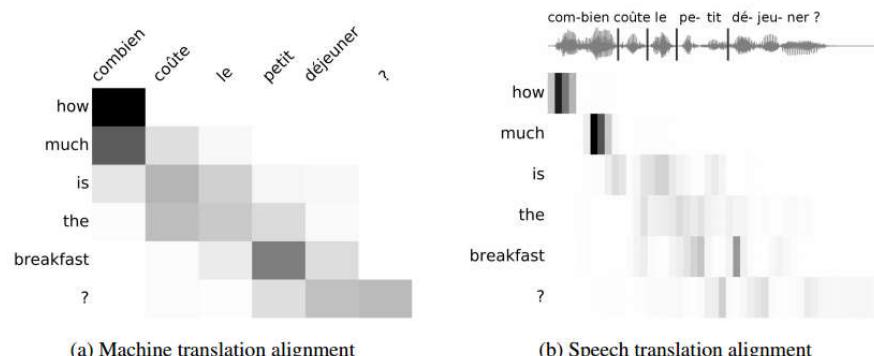
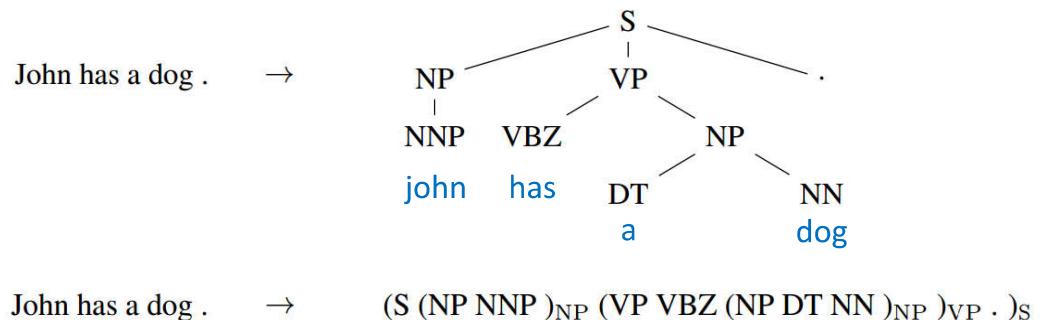


Figure 1: Alignments performed by the attention model during training

Beyond Sequence

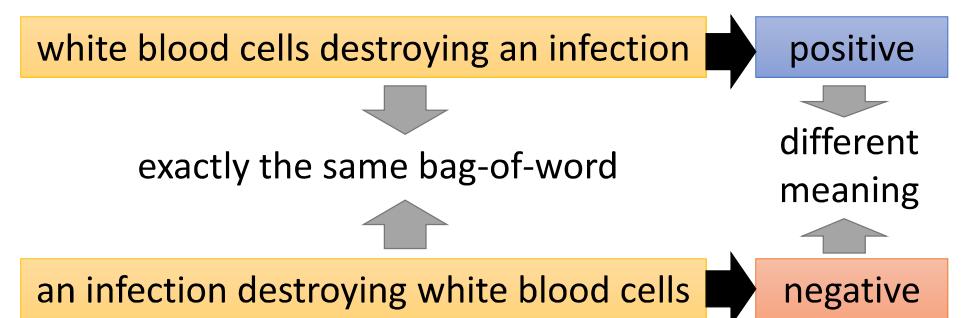
- Syntactic parsing



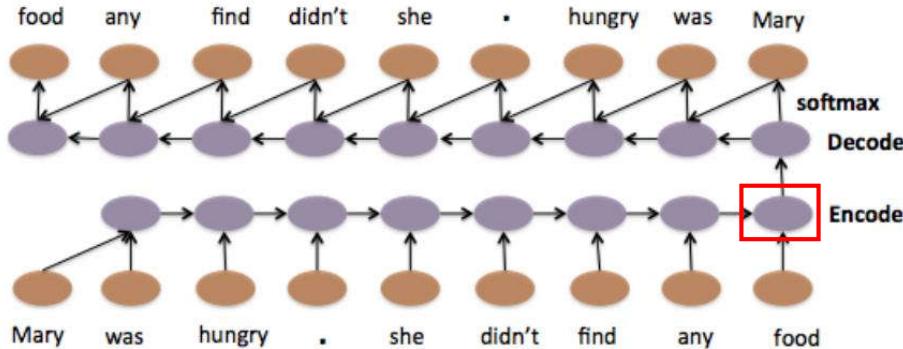
Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, Geoffrey Hinton, Grammar as a Foreign Language, NIPS 2015

Sequence-to-sequence Auto-encoder - Text

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

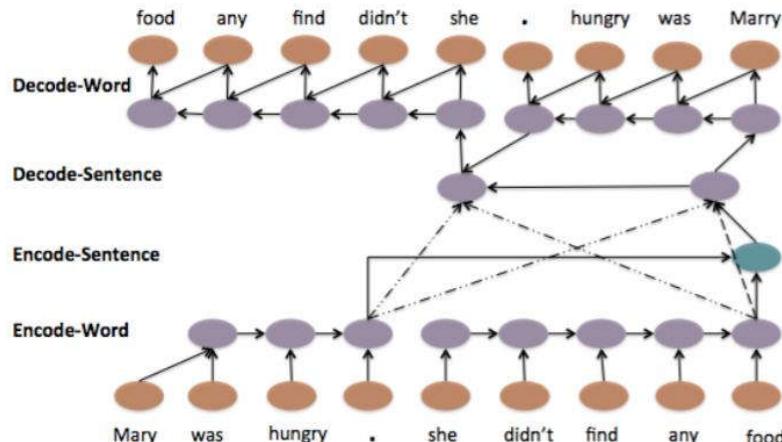


Sequence-to-sequence Auto-encoder - Text



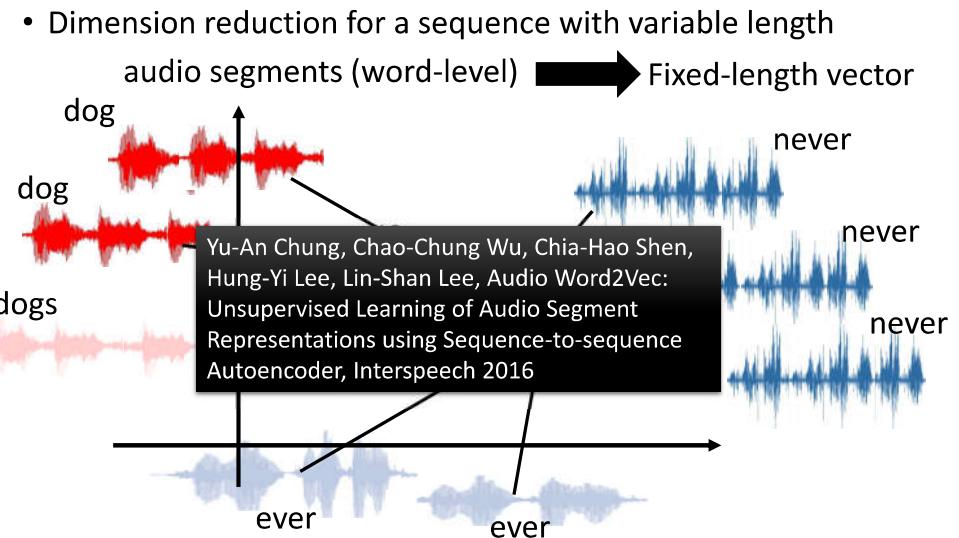
Li, Jiwei, Minh-Thang Luong, and Dan Jurafsky. "A hierarchical neural autoencoder for paragraphs and documents." *arXiv preprint arXiv:1506.01057*(2015).

Sequence-to-sequence Auto-encoder - Text

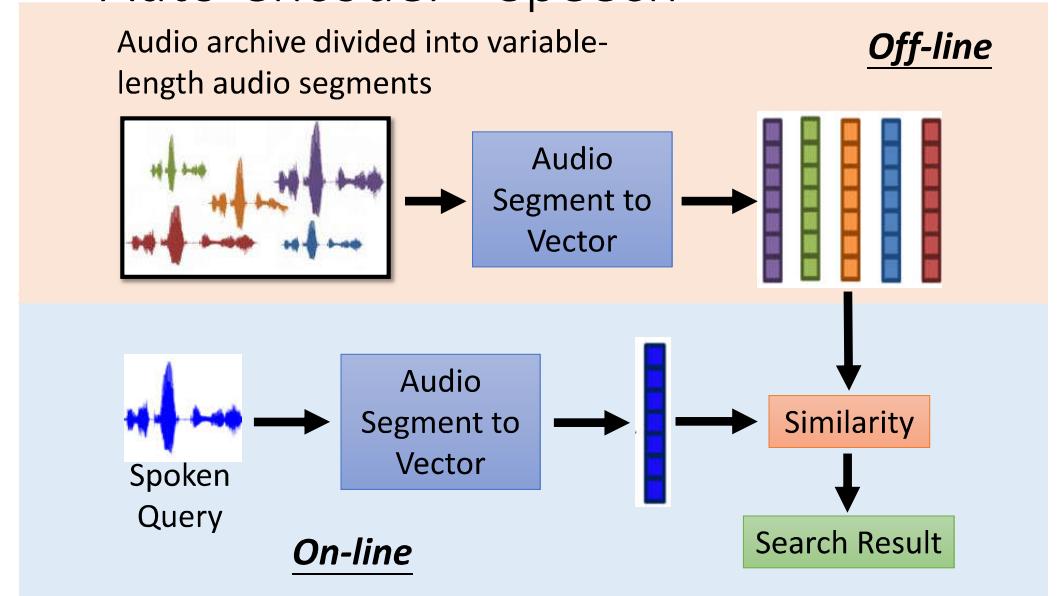


Li, Jiwei, Minh-Thang Luong, and Dan Jurafsky. "A hierarchical neural autoencoder for paragraphs and documents." *arXiv preprint arXiv:1506.01057*(2015).

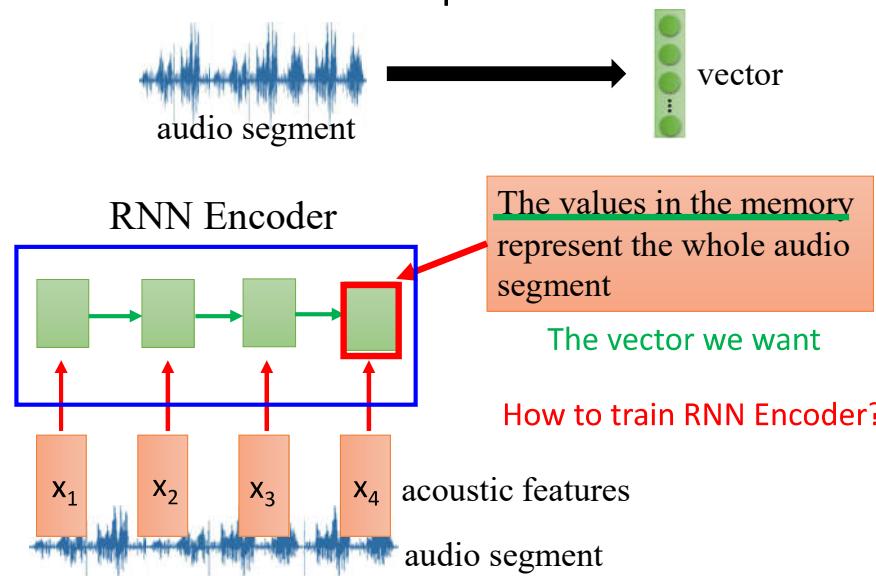
Sequence-to-sequence Auto-encoder - Speech



Sequence-to-sequence Auto-encoder - Speech

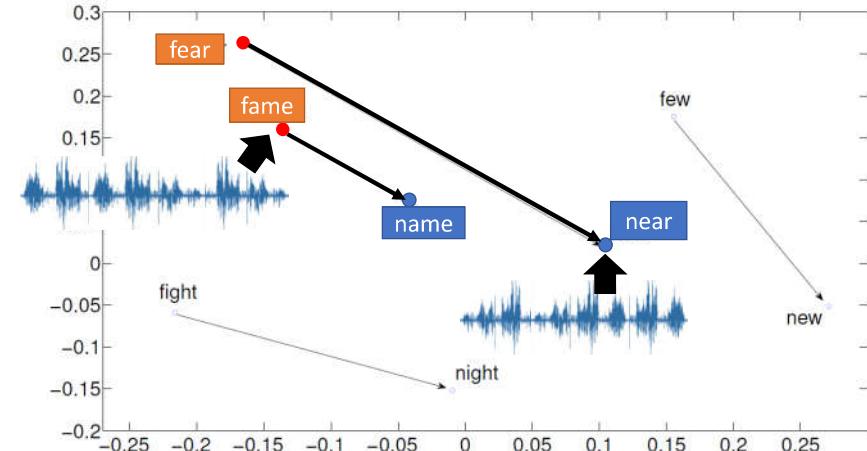


Sequence-to-sequence Auto-encoder - Speech



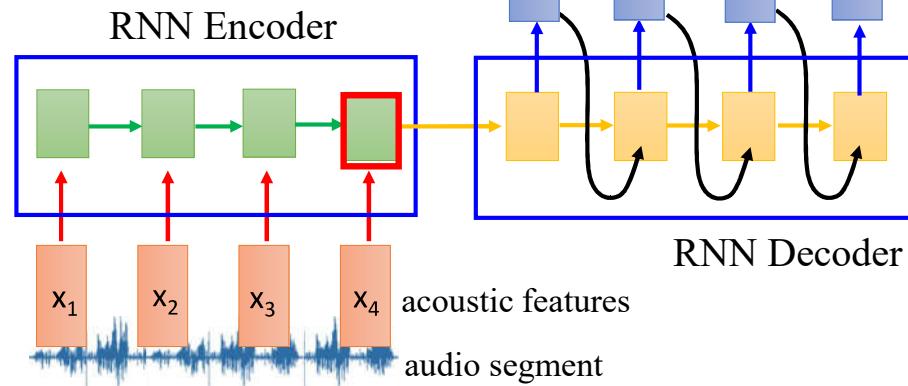
Sequence-to-sequence Auto-encoder - Speech

- Visualizing embedding vectors of the words

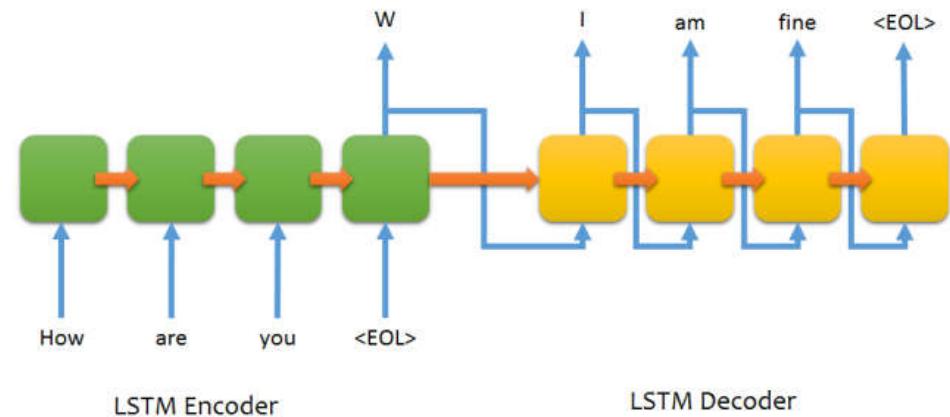


Sequence-to-sequence Auto-encoder

The RNN encoder and decoder are jointly trained.



Demo: Chat-bot



電視影集 (~40,000 sentences)、美國總統大選辯論

Demo: Chat-bot

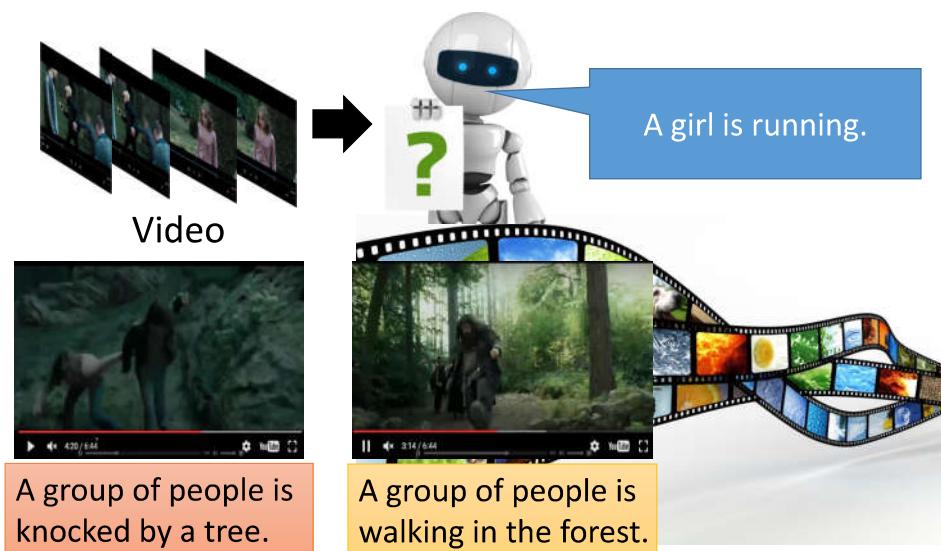
- Develop Team
 - Interface design: Prof. Lin-Lin Chen & Arron Lu
 - Web programming: Shi-Yun Huang
 - Data collection: Chao-Chuang Shih
 - System implementation: Kevin Wu, Derek Chuang, & Zhi-Wei Lee (李致緯), Roy Lu (盧柏儒)
 - System design: Richard Tsai & Hung-Yi Lee

Demo: Video Caption Generation

- Can machine describe what it see from video?
- Demo: 台大語音處理實驗室 曾柏翔、吳柏瑜、盧宏宗
- Video: 莊舜博、楊棋宇、黃邦齊、萬家宏



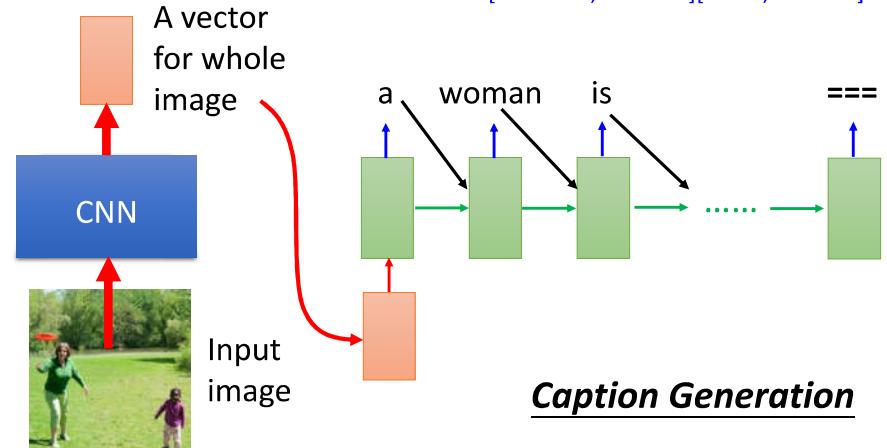
Demo: Video Caption Generation



Demo: Image Caption Generation

- Input an image, but output a sequence of words

[Kelvin Xu, arXiv'15][Li Yao, ICCV'15]



Demo: Image Caption Generation

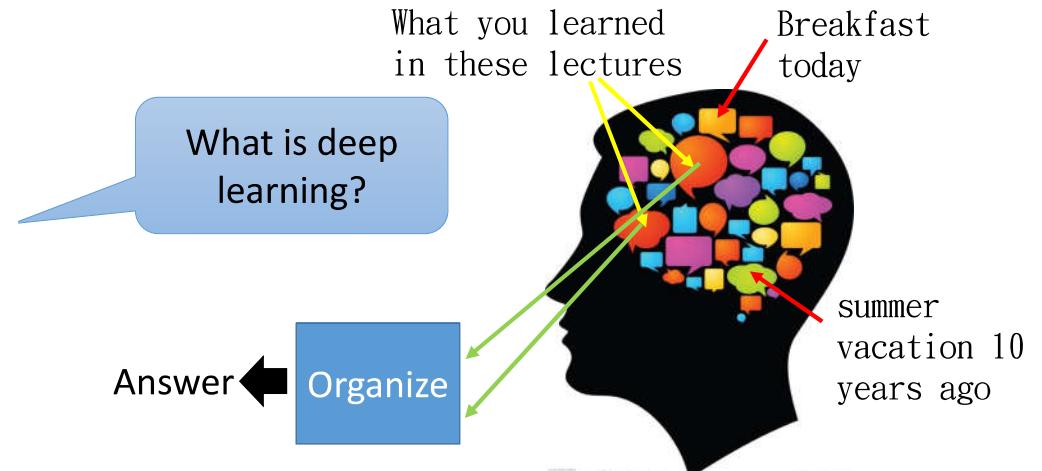
- Can machine describe what it see from image?
- Demo: 台大電機系 大四 蘇子睿、林奕辰、徐翊祥、陳奕安

MTK 產學大聯盟



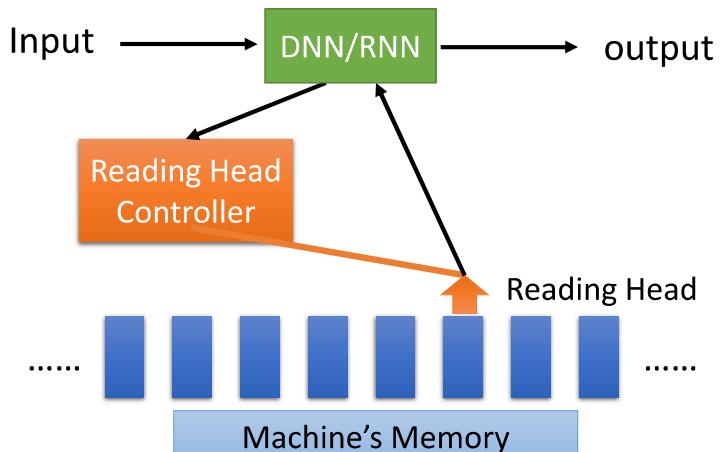
http://news.ltn.com.tw/photo/politics/breakingnews/975542_1

Attention-based Model



http://henrylo1605.blogspot.tw/2015/05/blog-post_56.html

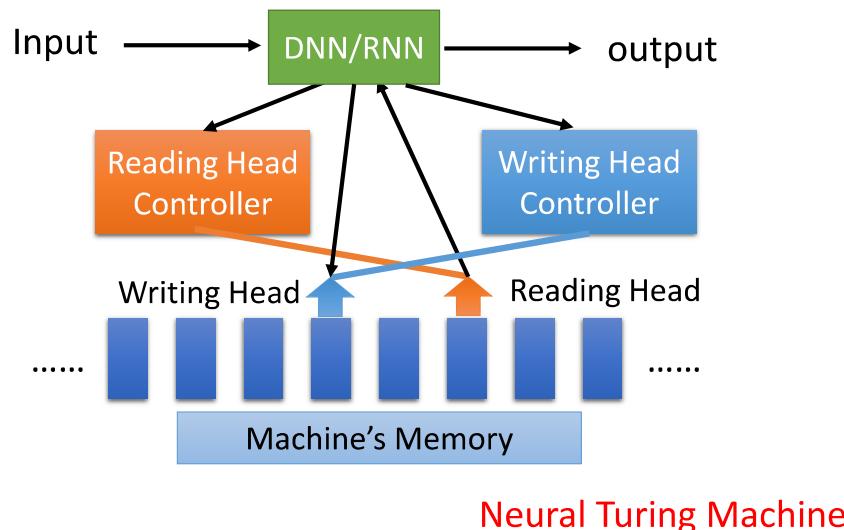
Attention-based Model



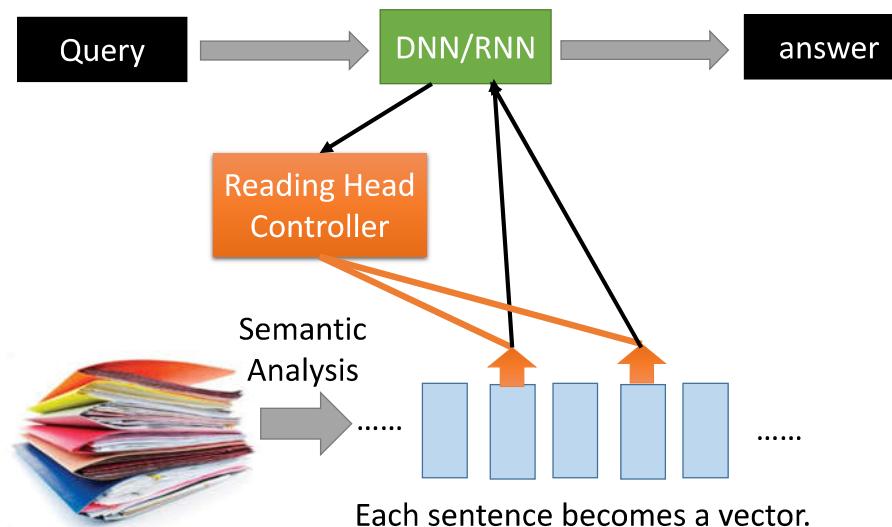
Ref:

[http://speech.ee.ntu.edu.tw/~tlkagk/courses/MLDS_2015_2/Lecture/Attain%20\(v3\).ecm.mp4/index.html](http://speech.ee.ntu.edu.tw/~tlkagk/courses/MLDS_2015_2/Lecture/Attain%20(v3).ecm.mp4/index.html)

Attention-based Model v2



Reading Comprehension



Reading Comprehension

- End-To-End Memory Networks. S. Sukhbaatar, A. Szlam, J. Weston, R. Fergus. NIPS, 2015.

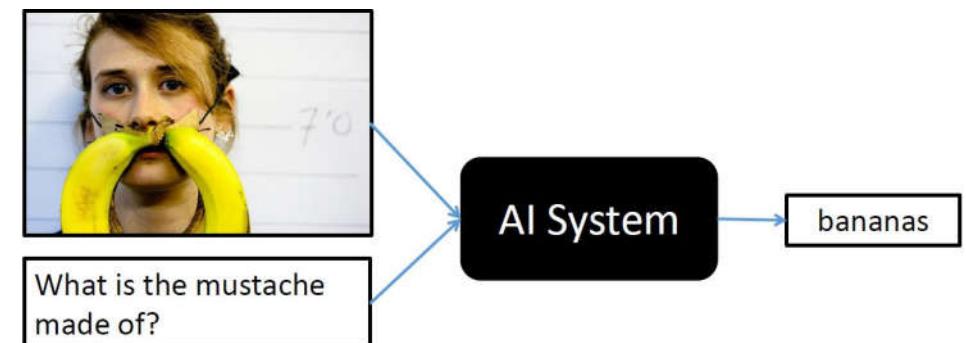
The position of reading head:

Story (16: basic induction)	Support	Hop 1	Hop 2	Hop 3
Brian is a frog.	yes	0.00	0.98	0.00
Lily is gray.		0.07	0.00	0.00
Brian is yellow.	yes	0.07	0.00	1.00
Julius is green.		0.06	0.00	0.00
Greg is a frog.	yes	0.76	0.02	0.00
What color is Greg?	Answer: yellow	Prediction: yellow		

Keras has example:

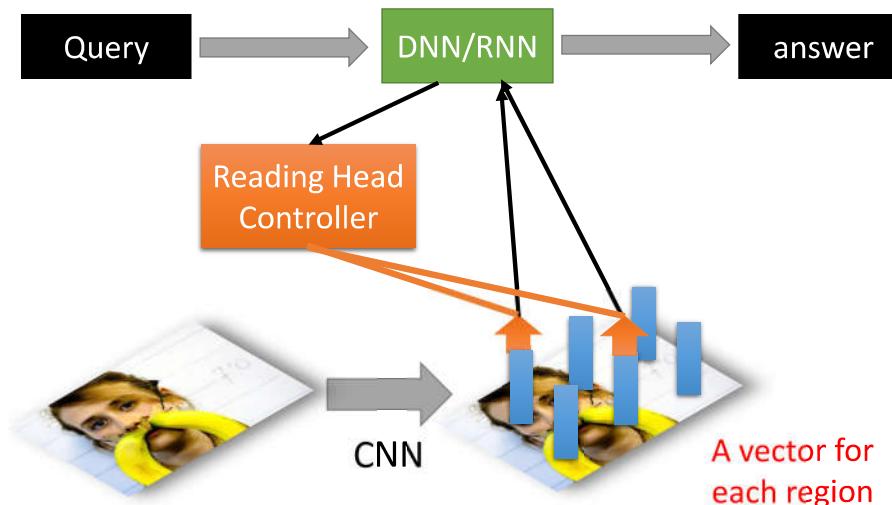
https://github.com/fchollet/keras/blob/master/examples/babi_memnn.py

Visual Question Answering

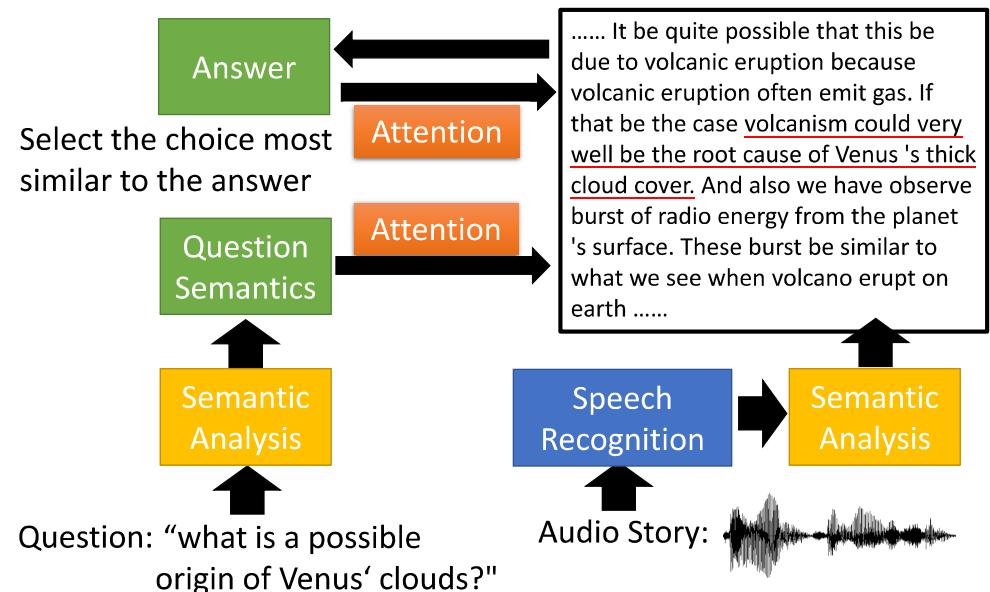


source: <http://visualqa.org/>

Visual Question Answering



Model Architecture



Everything is learned from training examples

Speech Question Answering

• TOEFL Listening Comprehension Test by Machine

- Example:

Audio Story: (The original story is 5 min long.)

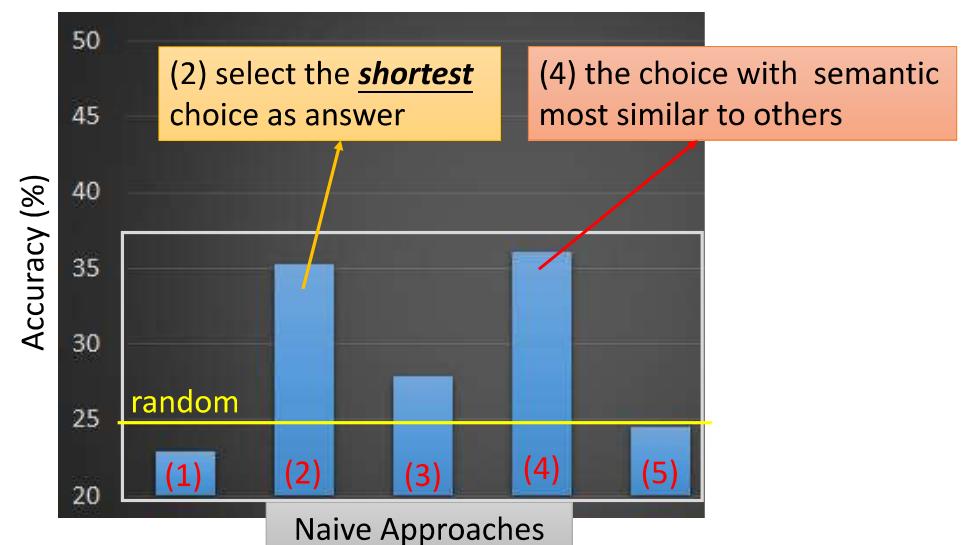
Question: "What is a possible origin of Venus' clouds?"

Choices:

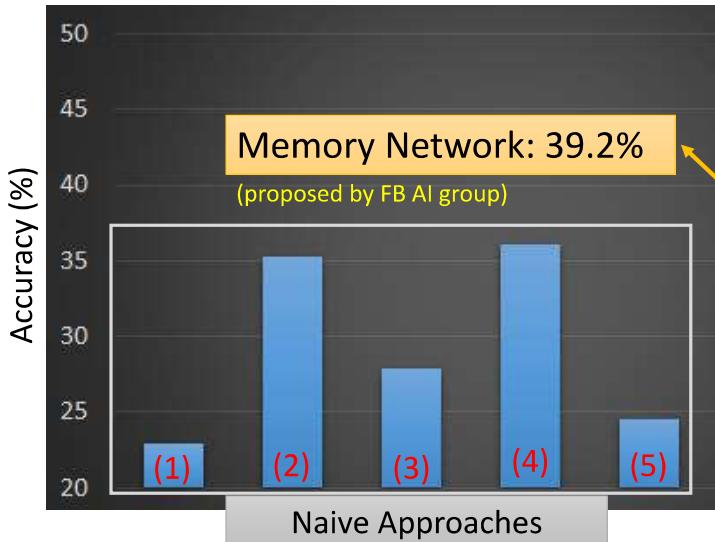
- (A) gases released as a result of volcanic activity
- (B) chemical reactions caused by high surface temperatures
- (C) bursts of radio energy from the plane's surface
- (D) strong winds that blow dust into the atmosphere

Simple Baselines

Experimental setup:
717 for training,
124 for validation, 122 for testing



Memory Network

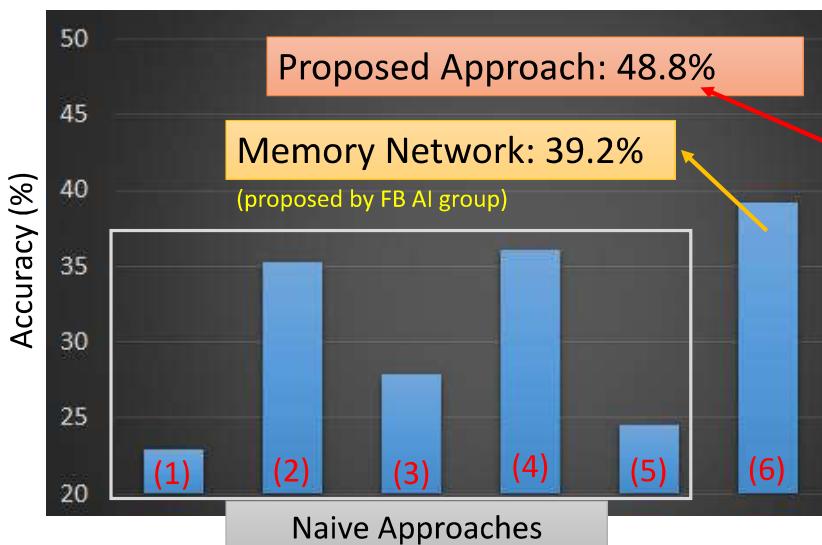


To Learn More

- The Unreasonable Effectiveness of Recurrent Neural Networks
 - <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>
- Understanding LSTM Networks
 - <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Proposed Approach

[Tseng & Lee, Interspeech 16]
[Fang & Hsu & Lee, SLT 16]



Deep & Structured

RNN v.s. Structured Learning

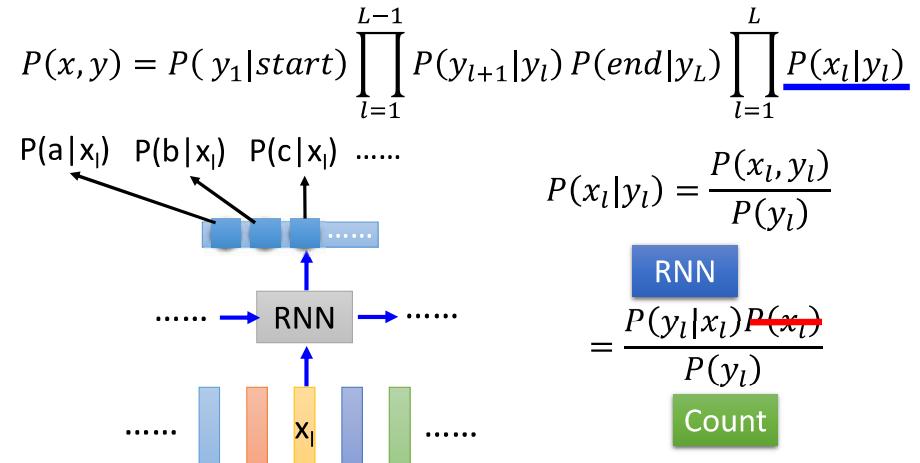
- RNN, LSTM
 - Unidirectional RNN does NOT consider the whole sequence
 - Cost and error not always related
 - Deep 



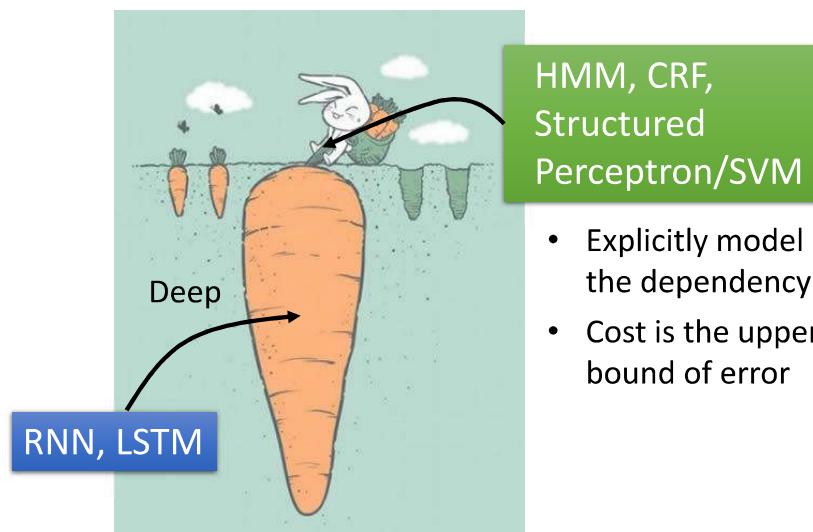
- HMM, CRF, Structured Perceptron/SVM
 - Using Viterbi, so consider the whole sequence 
 - How about Bidirectional RNN?
 - Can explicitly consider the label dependency 
 - Cost is the upper bound of error 

Integrated together

- Speech Recognition: CNN/LSTM/DNN + HMM



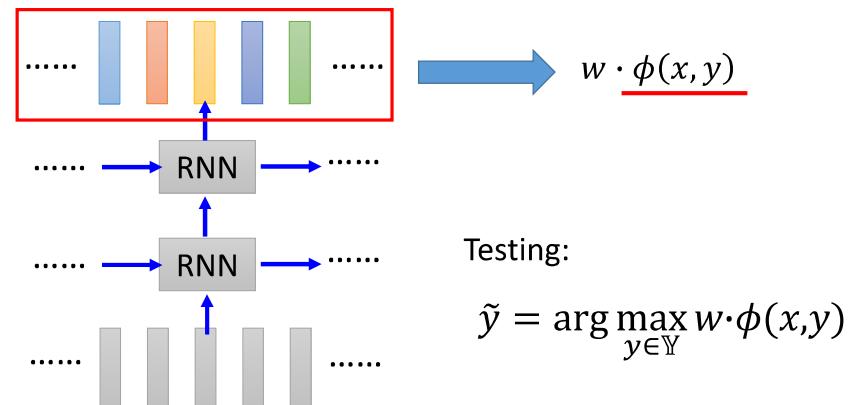
Integrated Together



<http://photo30.bababian.com/upload1/20100415/42E9331A6580A46A5F89E98638B8FD76.jpg>

Integrated together

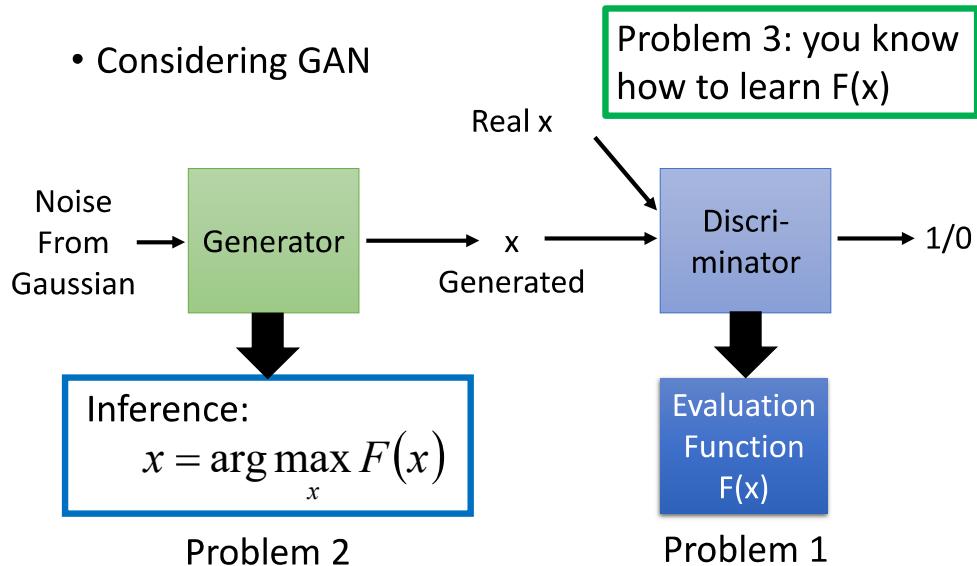
- Semantic Tagging: Bi-directional LSTM + CRF/Structured SVM



Deep and Structured
will be the future.

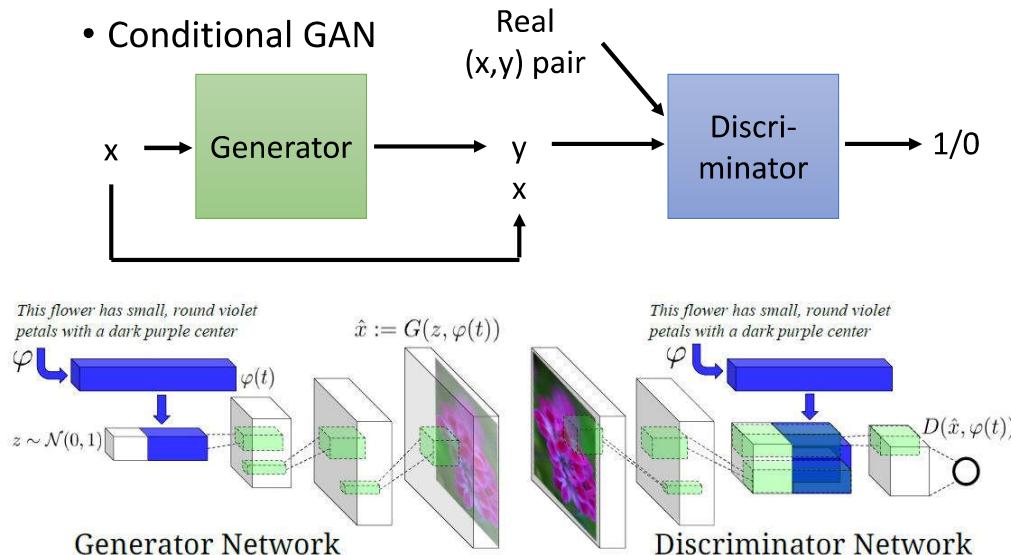
Is structured learning practical?

- Considering GAN



Is structured learning practical?

- Conditional GAN



Sounds crazy?
People do think in this way ...

- Connect Energy-based model with GAN:
 - A Connection Between Generative Adversarial Networks, Inverse Reinforcement Learning, and Energy-Based Models
 - Deep Directed Generative Models with Energy-Based Probability Estimation
 - ENERGY-BASED GENERATIVE ADVERSARIAL NETWORKS
- Deep learning model for inference
 - Deep Unfolding: Model-Based Inspiration of Novel Deep Architectures
 - Conditional Random Fields as Recurrent Neural Networks

Machine learning and having it deep and structured (MLDS)

- 和 ML 的不同

- 在這學期 ML 中有提過的內容 (DNN, CNN ...)，在 MLDS 中不再重複，只做必要的復習

- 教科書：“Deep Learning”
(<http://www.deeplearningbook.org/>)

- Part II 是講 deep learning 、Part III 就是講 structured learning

- Part II: Modern Practical Deep Networks

- 6 Deep Feedforward Networks
- 7 Regularization for Deep Learning
- 8 Optimization for Training Deep Models
- 9 Convolutional Networks
- 10 Sequence Modeling: Recurrent and Recurrent Models
- 11 Practical Methodology
- 12 Applications

- Part III: Deep Learning Research

- 13 Linear Factor Models
- 14 Autoencoders
- 15 Representation Learning
- 16 Structured Probabilistic Models for Deep Learning
- 17 Monte Carlo Methods
- 18 Confronting the Partition Function
- 19 Approximate Inference
- 20 Deep Generative Models

Machine learning and having it deep and structured (MLDS)

- 所有作業都 2~4 人一組，可以先組好隊後一起來修
- MLDS 的作業和之前不同
 - RNN (把之前 MLDS 的三個作業合為一個)、Attention-based model 、Deep Reinforcement Learning 、Deep Generative Model 、Sequence-to-sequence learning
- MLDS 初選不開放加簽，以組為單位加簽，作業0的內容是做一個 DNN (可用現成套件)