Recurrent Neural Network (RNN)

1-of-N encoding

How to represent each word as a vector?

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

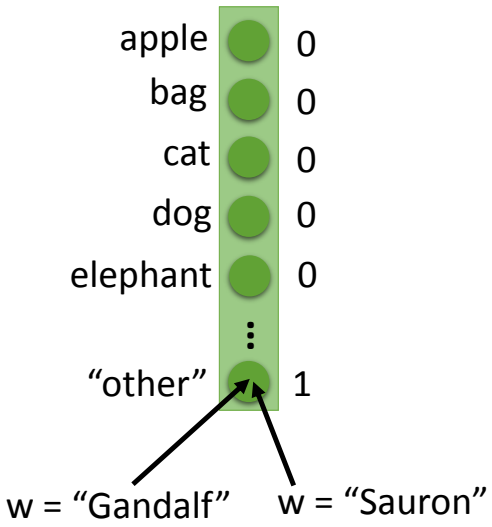
The vector is lexicon size.

Each dimension corresponds to a word in the lexicon

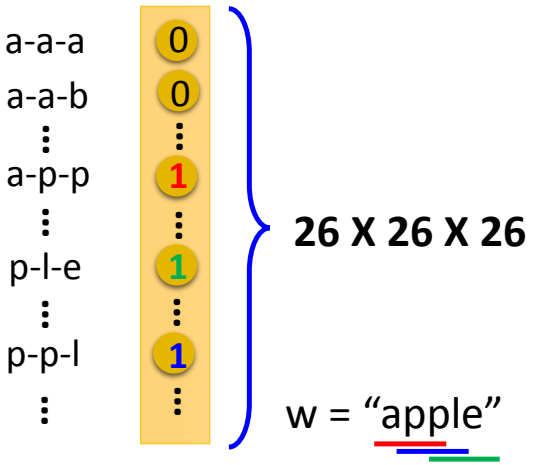
The dimension for the word is 1, and others are 0

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]

Dimension for "Other"

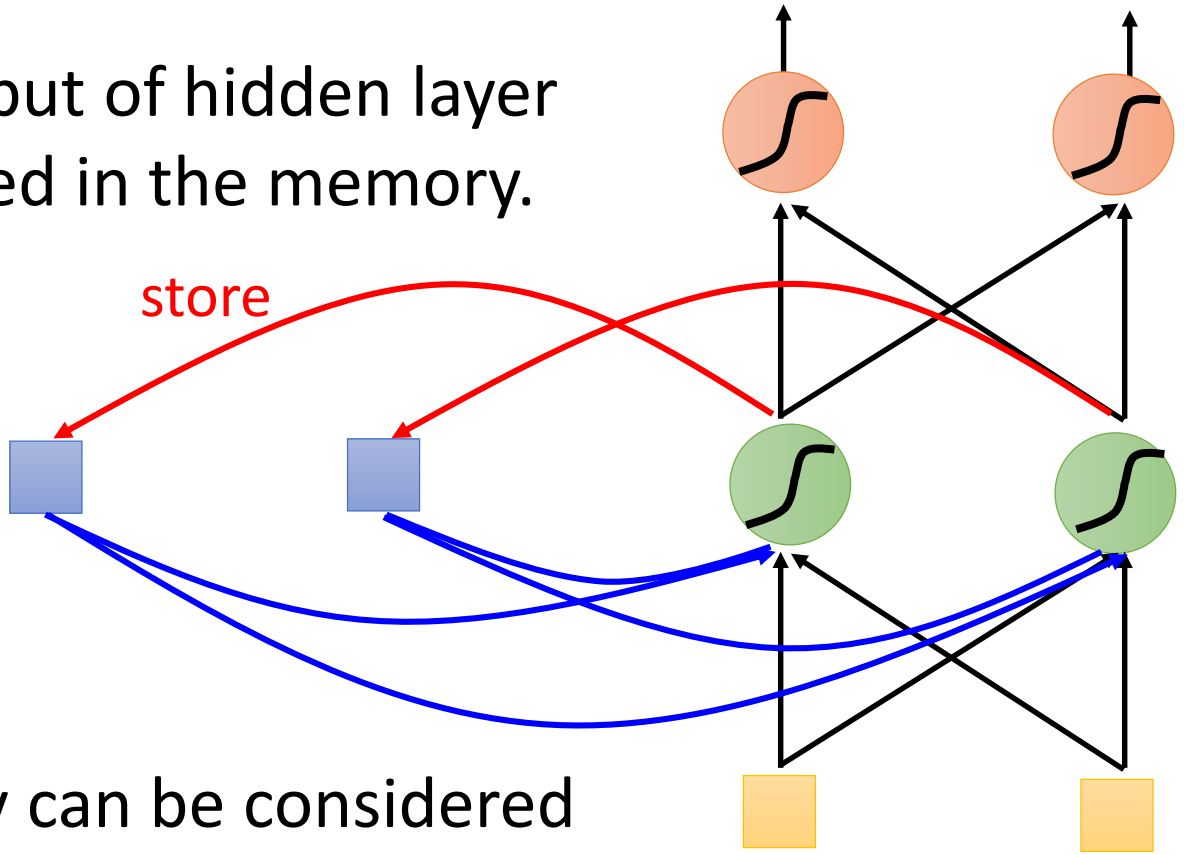


Word hashing



Recurrent Neural Network (RNN)

The output of hidden layer are stored in the memory.



Memory can be considered as another input.

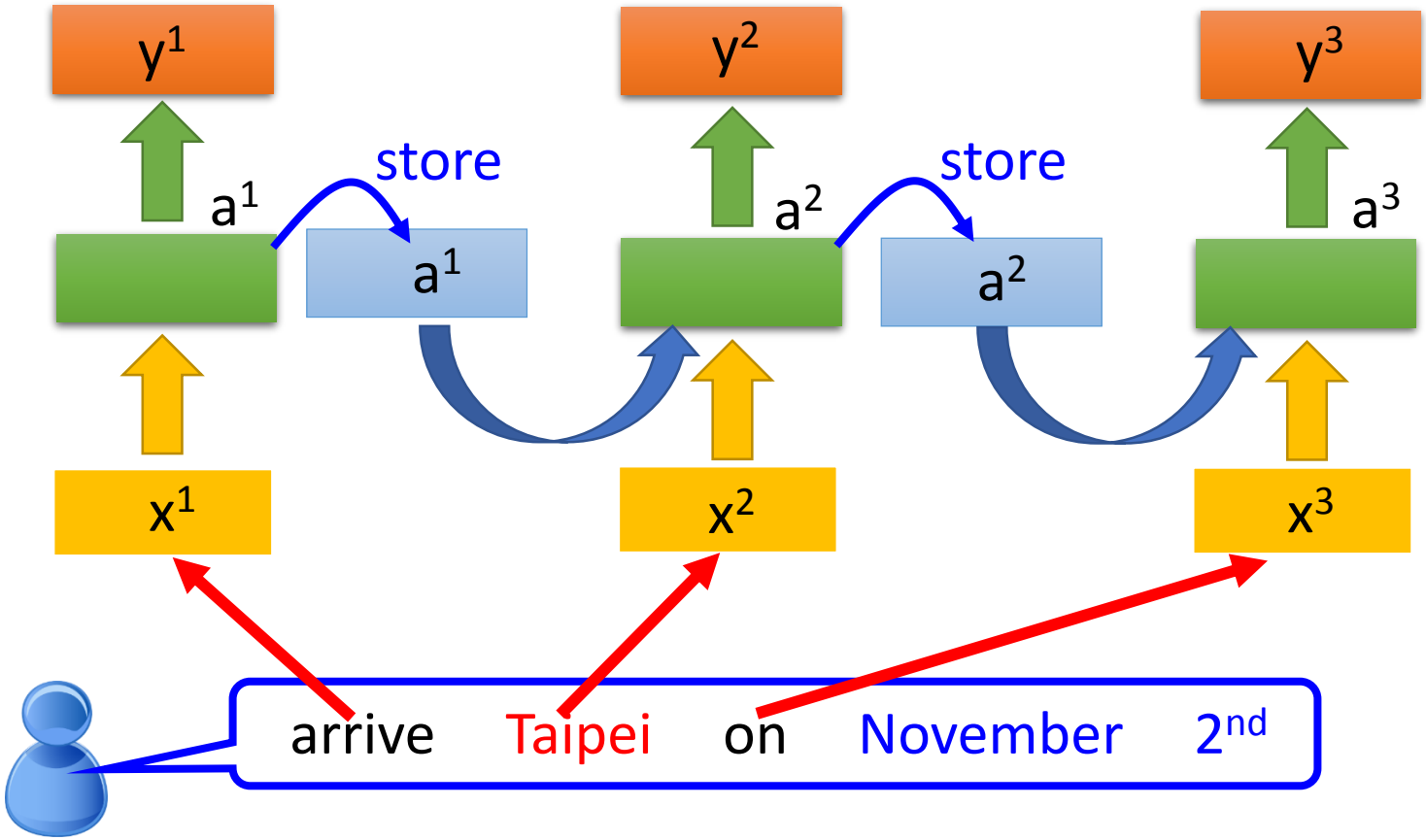
RNN

The same network is used again and again.

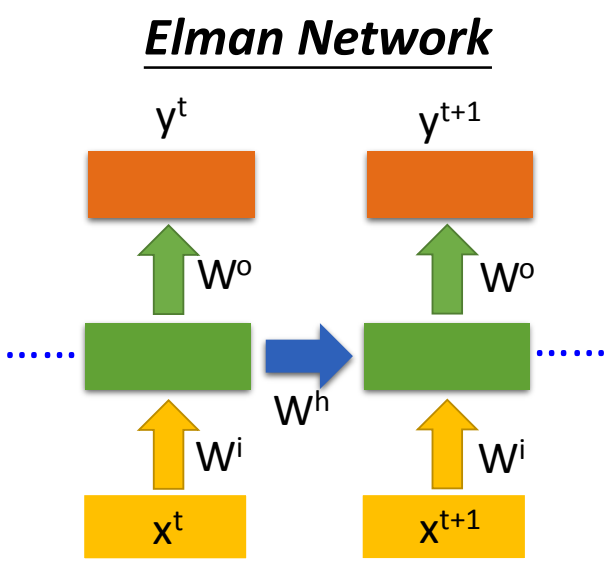
Probability of "arrive" in each slot

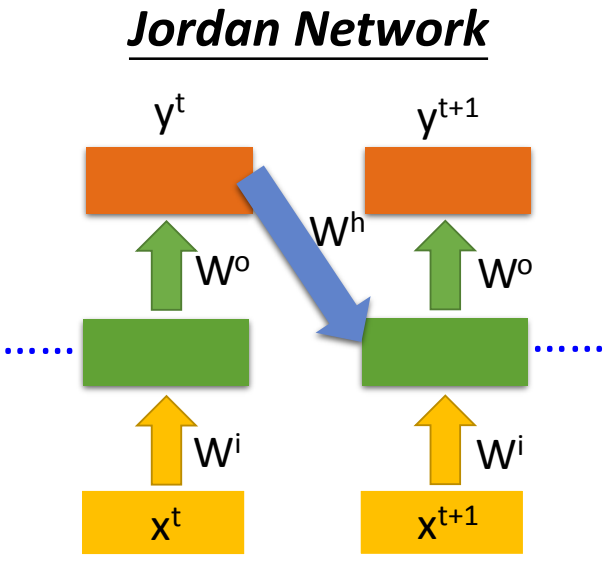
Probability of "Taipei" in each slot

Probability of "on" in each slot

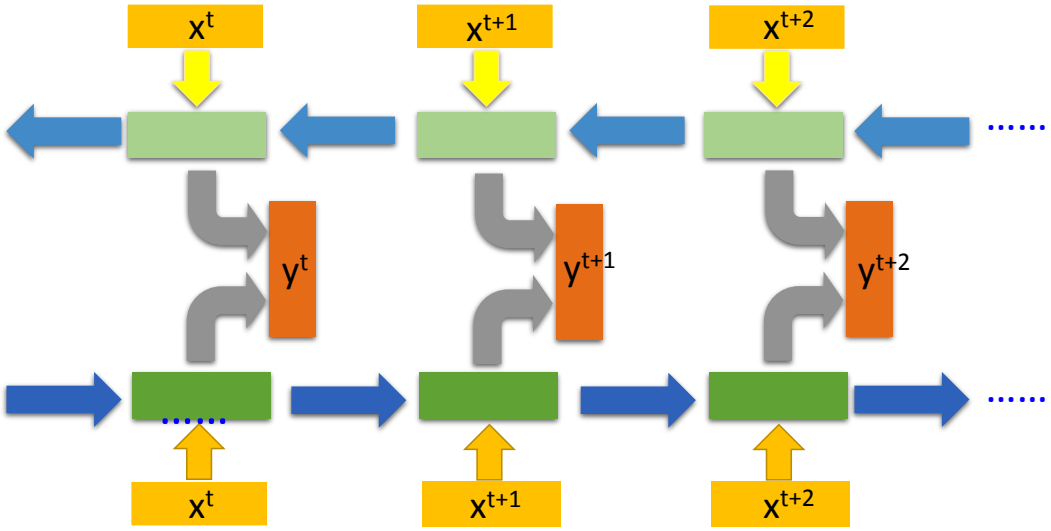


Elman Network & Jordan Network

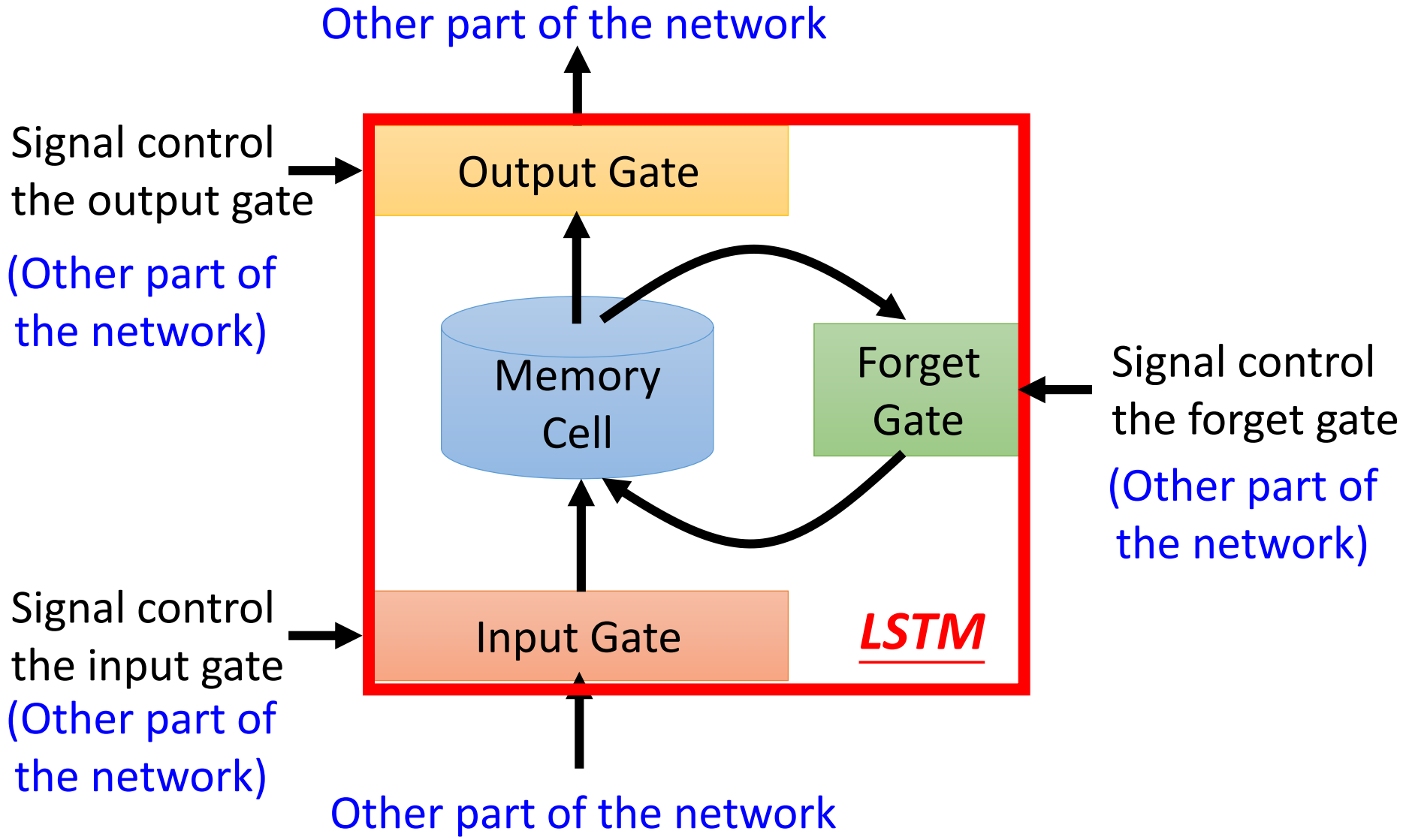


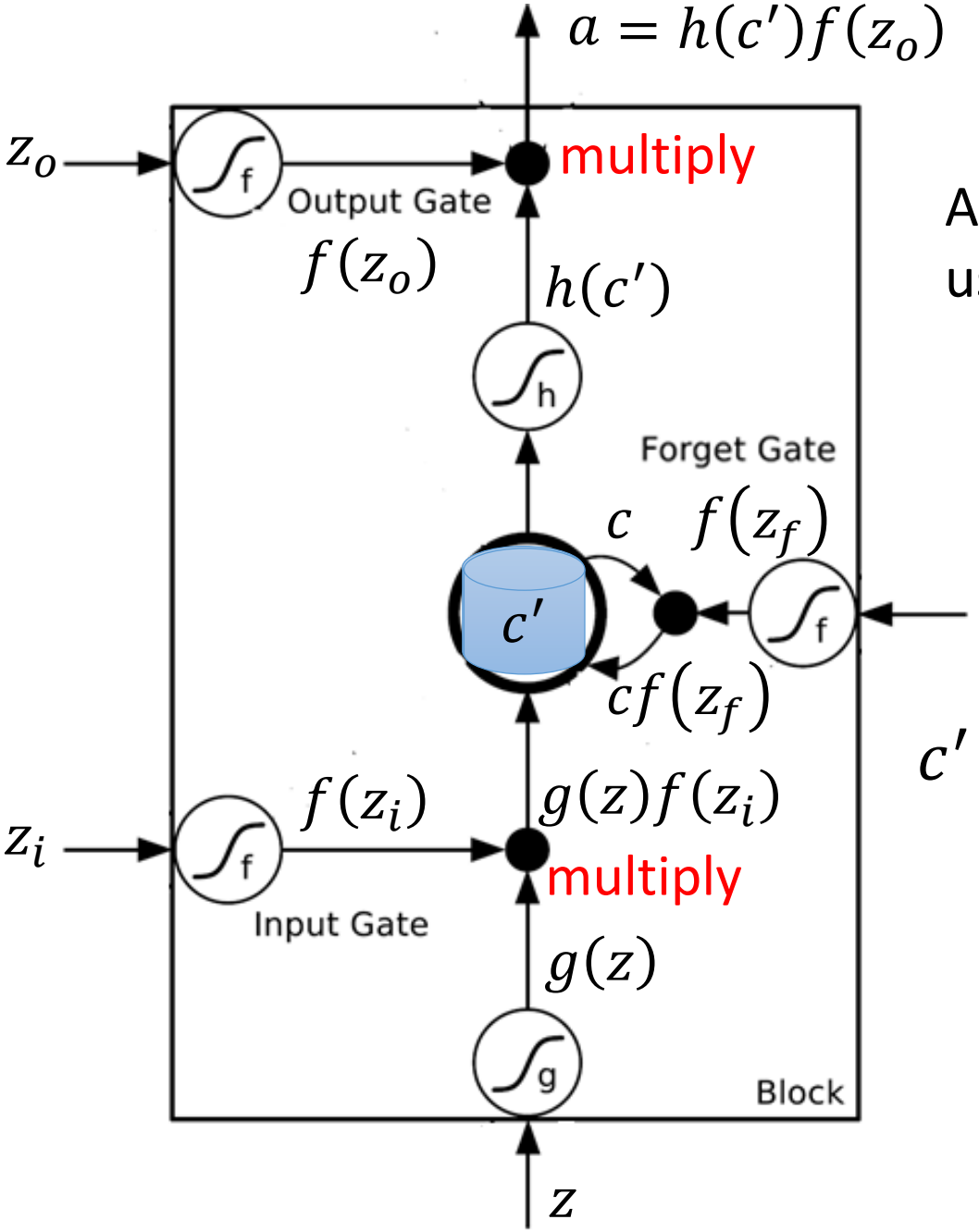


Bidirectional RNN



Long Short-term Memory (LSTM)





Activation function f is usually a sigmoid function

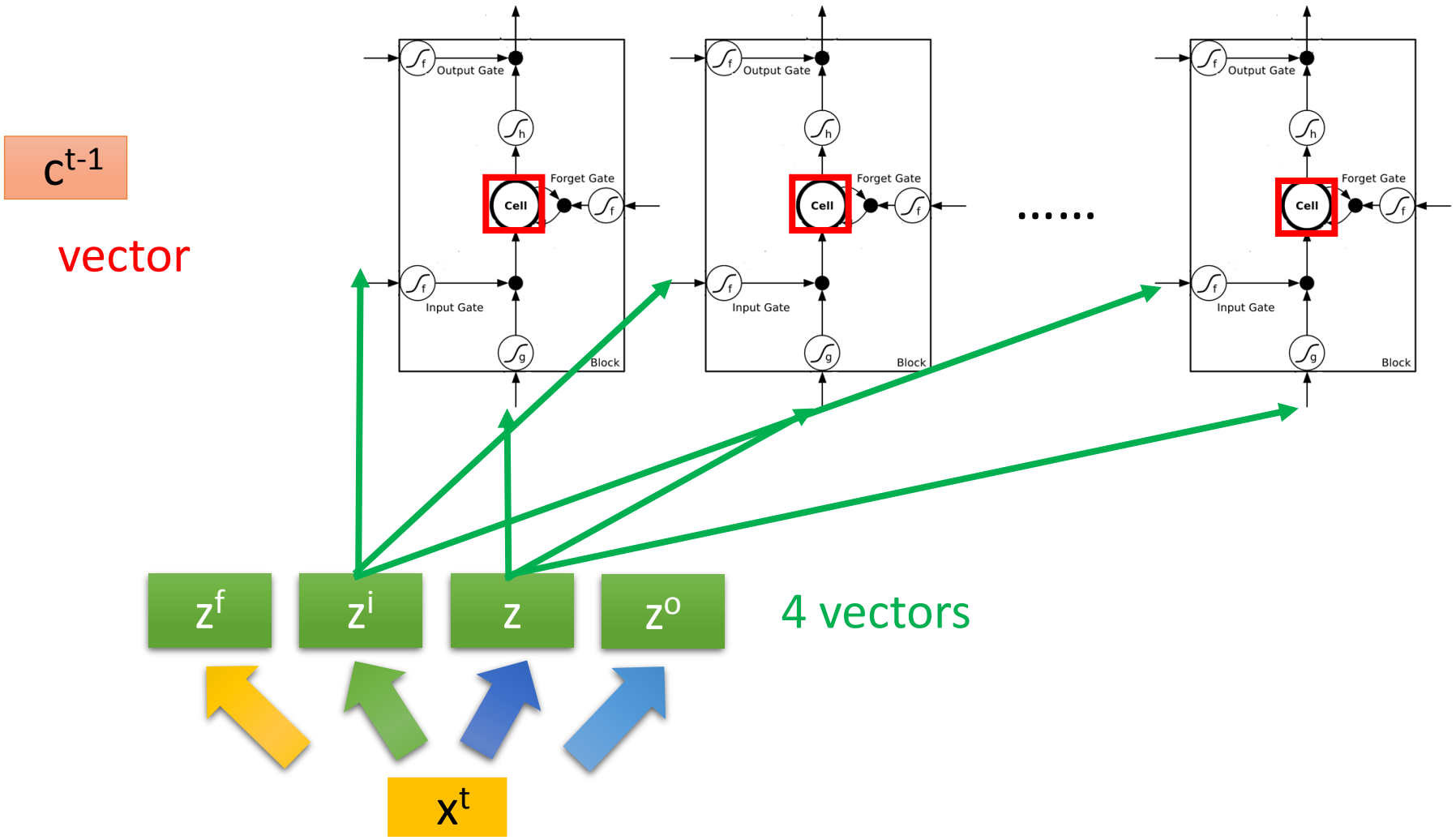
Between 0 and 1

Mimic open and close gate

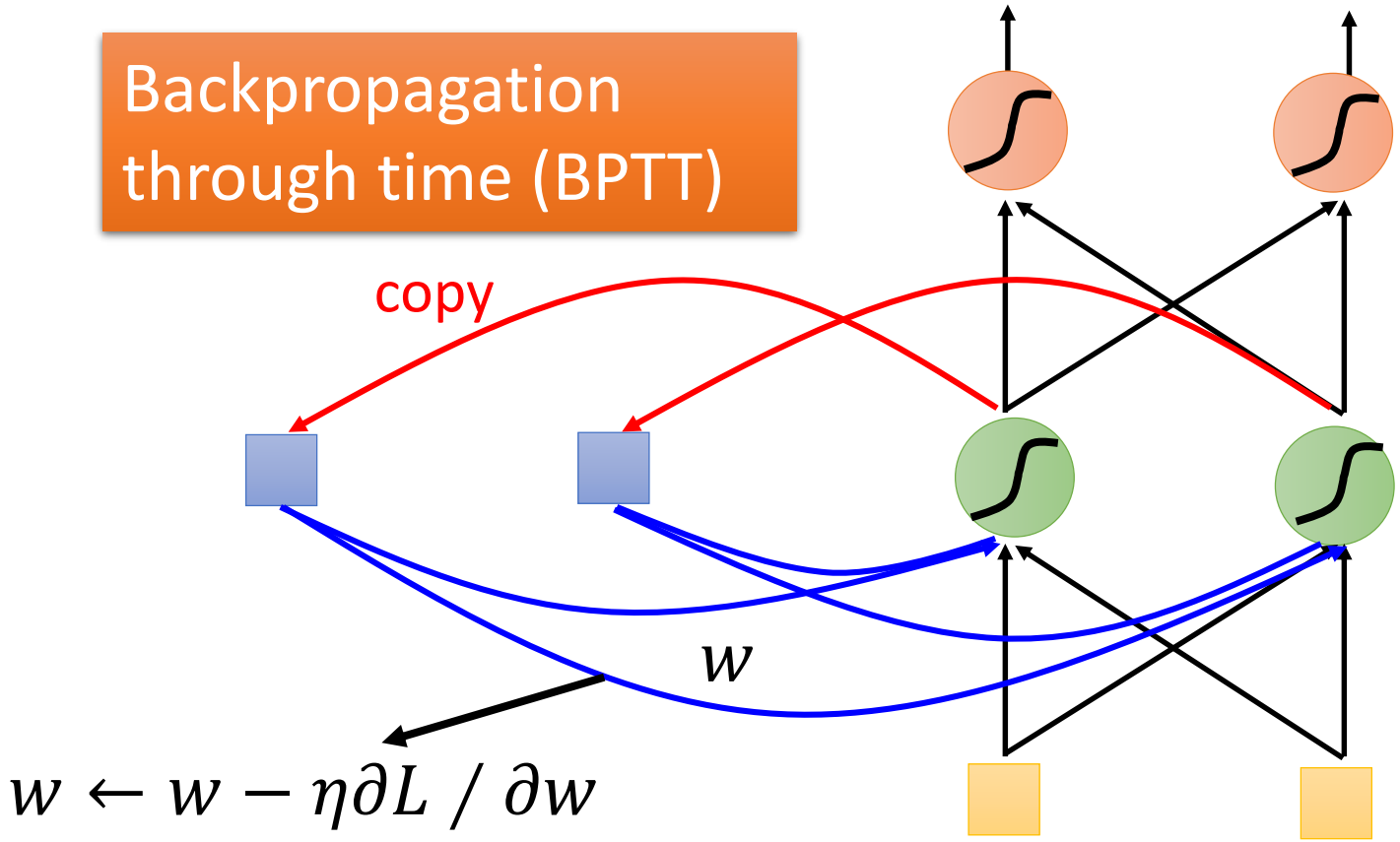
$$c' = g(z)f(z_i) + cf(z_f)$$

LSTM

4 times of parameters



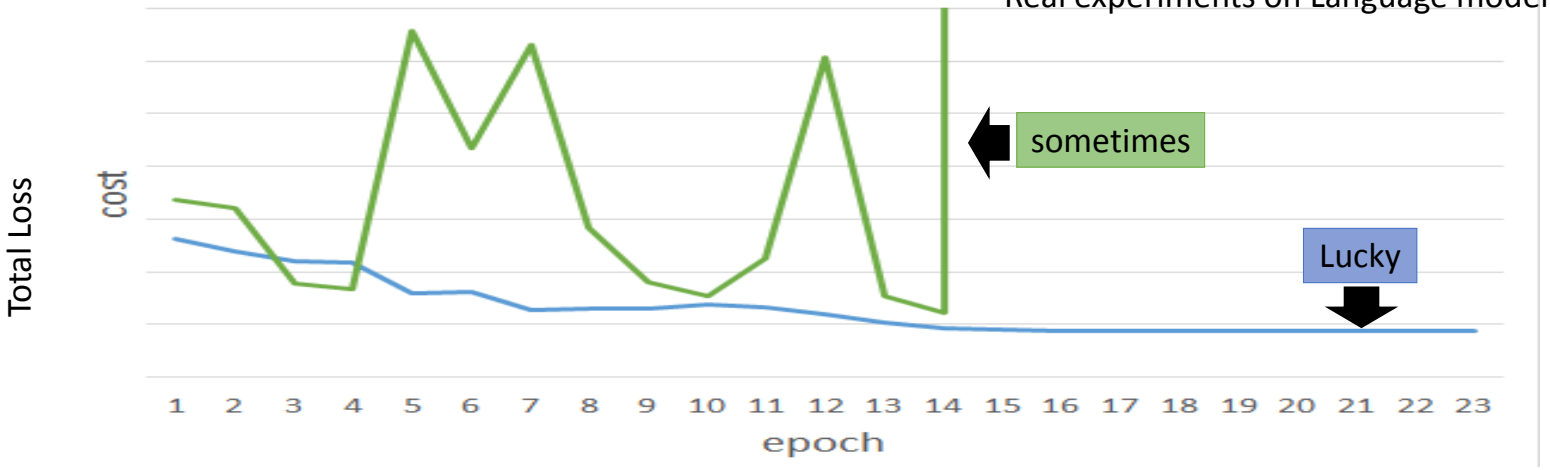
Learning


Backpropagation through time (BPTT)
copy
w
$w \leftarrow w - \eta \partial L / \partial w$

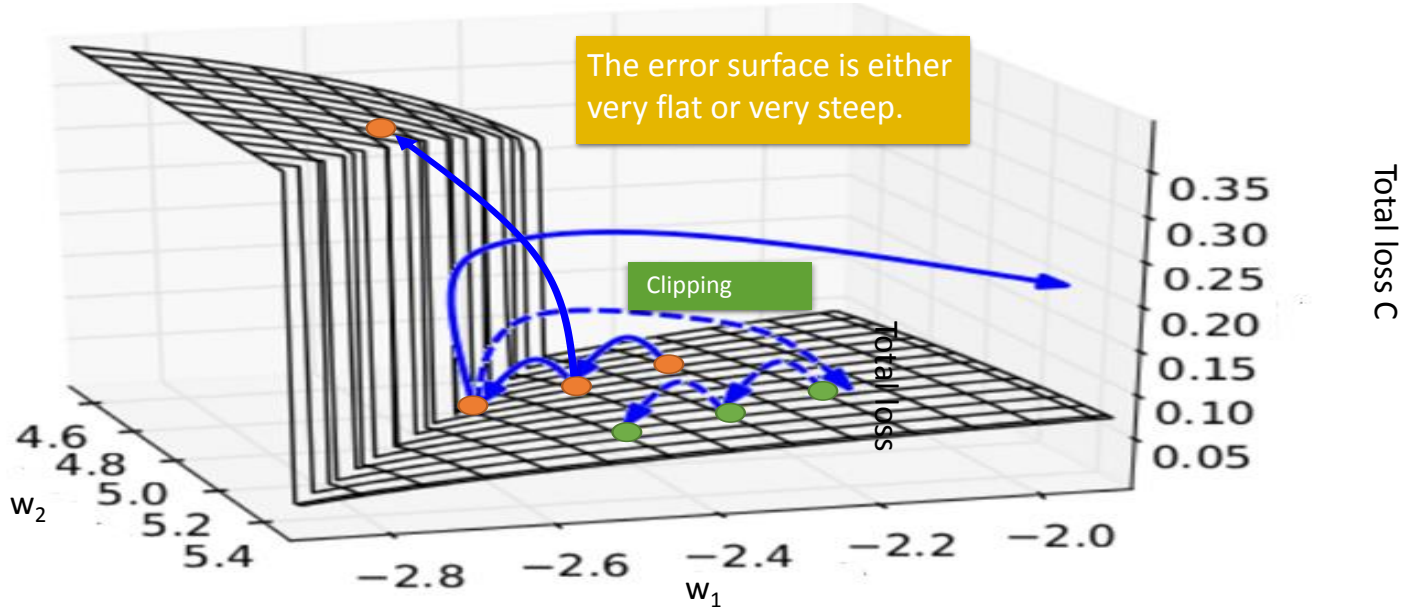
Unfortunately ……

- RNN-based network is not always easy to learn

Real experiments on Language modeling

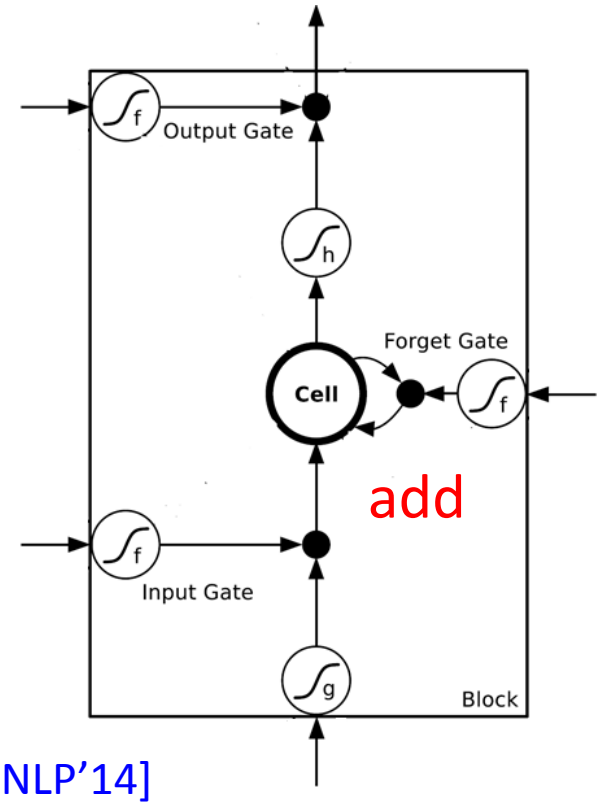


The error surface is rough.



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



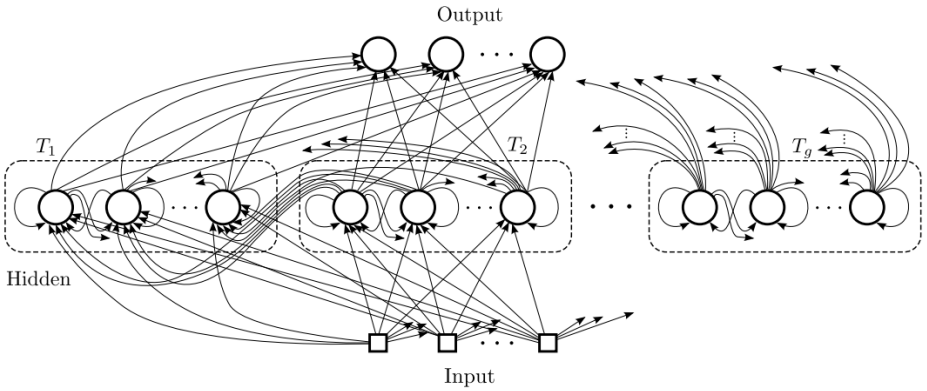
No Gradient vanishing
(If forget gate is opened.)

Gated Recurrent Unit (GRU): simpler than LSTM

[Cho, EMNLP'14]

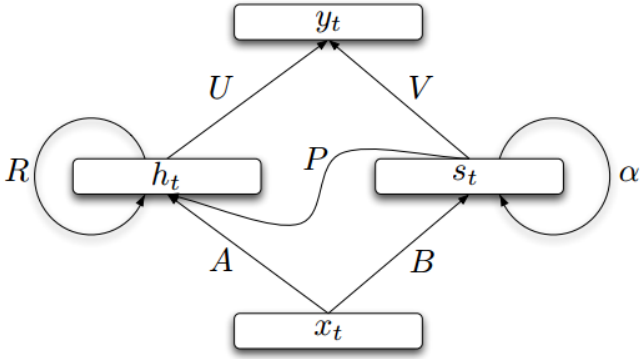
Helpful Techniques

Clockwise RNN



[Jan Koutnik, JMLR'14]

Structurally Constrained Recurrent Network (SCRN)



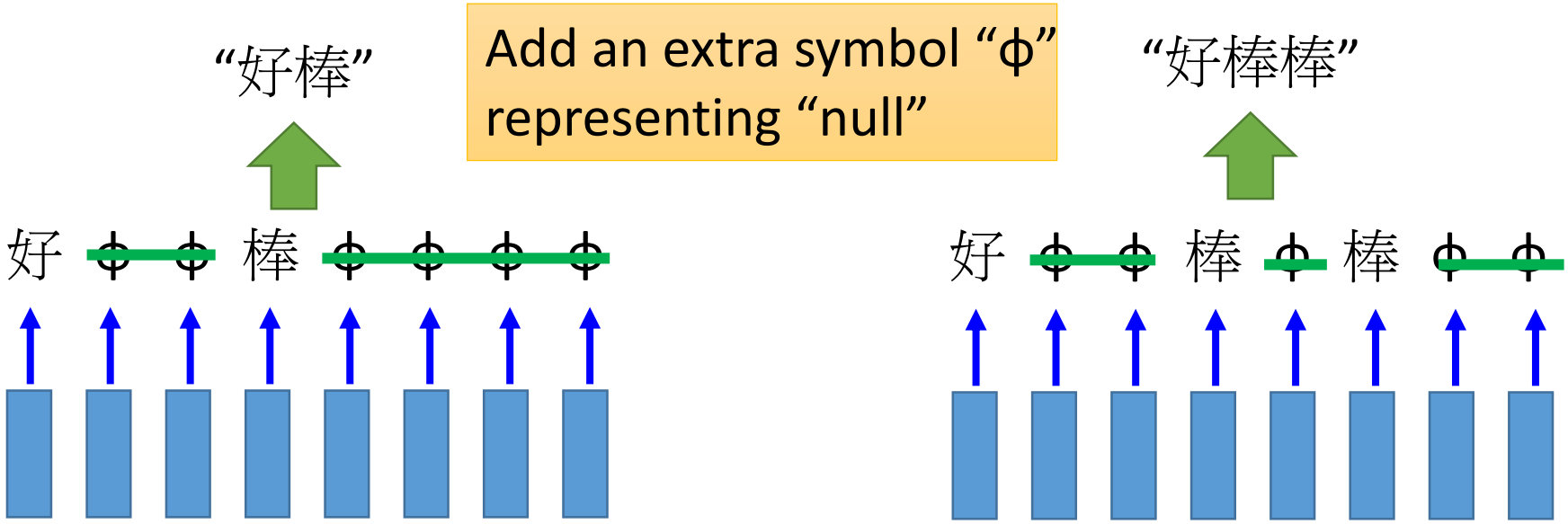
[Tomas Mikolov, ICLR'15]

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 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][Haşim Sak, Interspeech'15][Jie Li, Interspeech'15][Andrew Senior, ASRU'15]



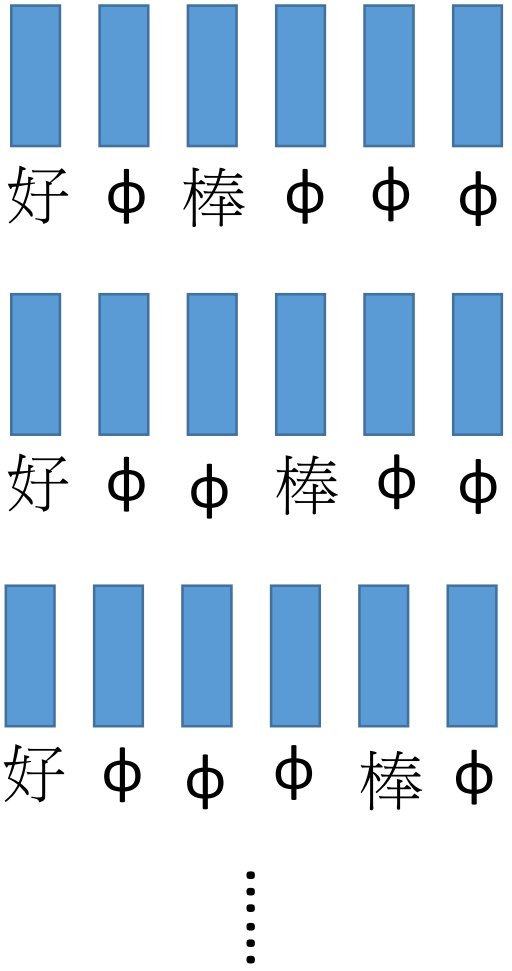
Many to Many (Output is shorter)

- CTC: Training

Acoustic Features:

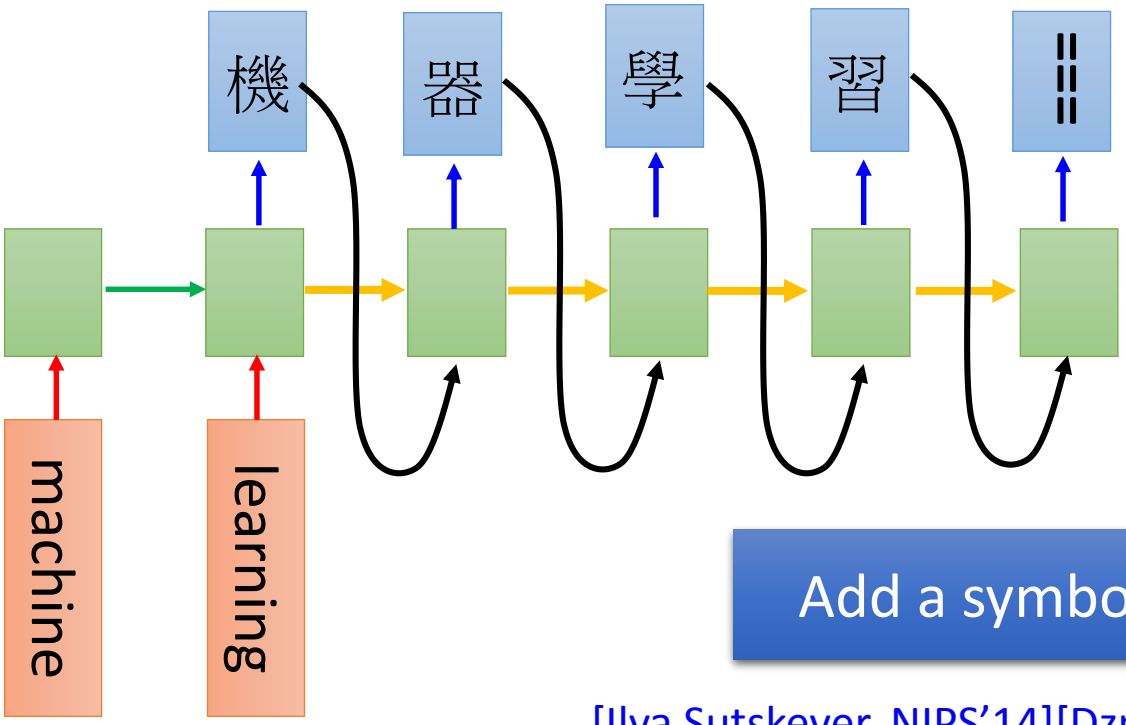
Label: 好 棒

All possible alignments are considered as correct.



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)

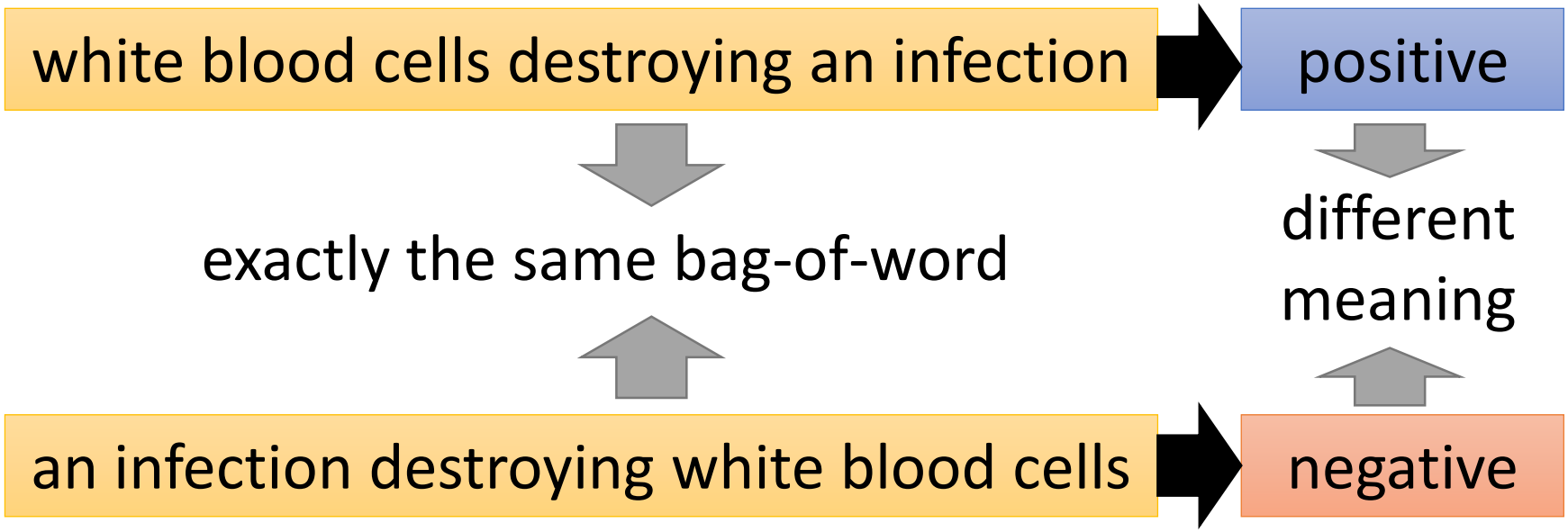


Add a symbol "===" (斷)

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

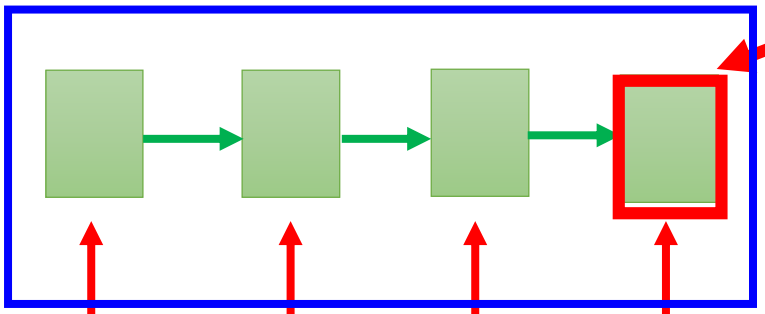
Sequence-to-sequence Auto-encoder - Text

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



Sequence-to-sequence Auto-encoder - Speech

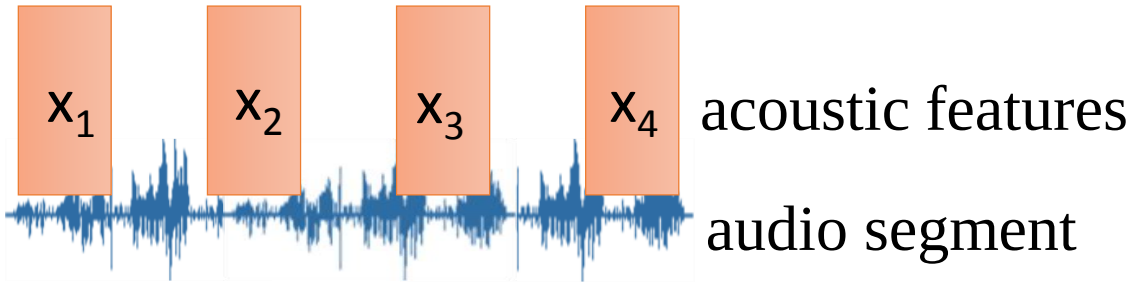
RNN Encoder



The values in the memory represent the whole audio segment

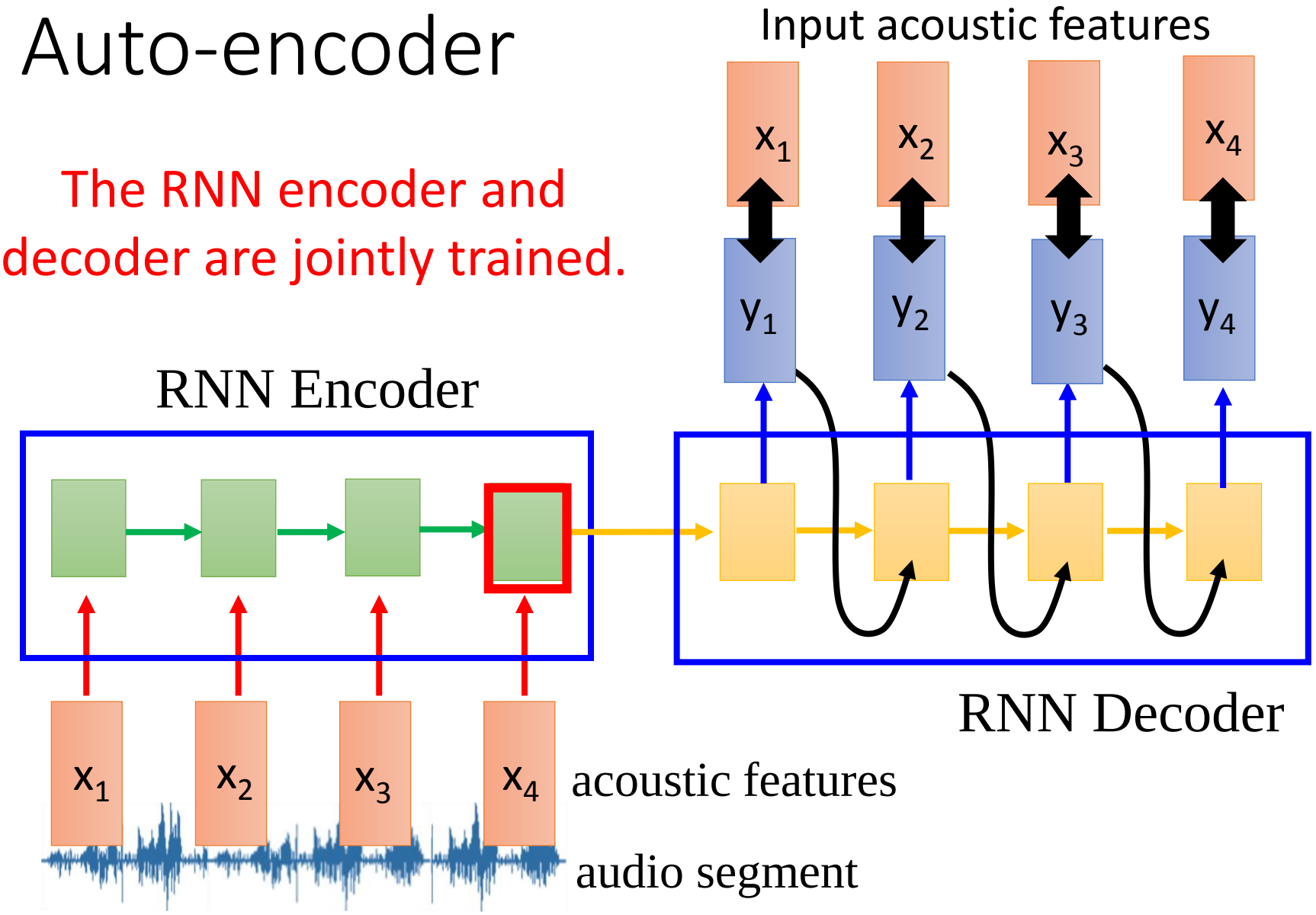
The vector we want

How to train RNN Encoder?

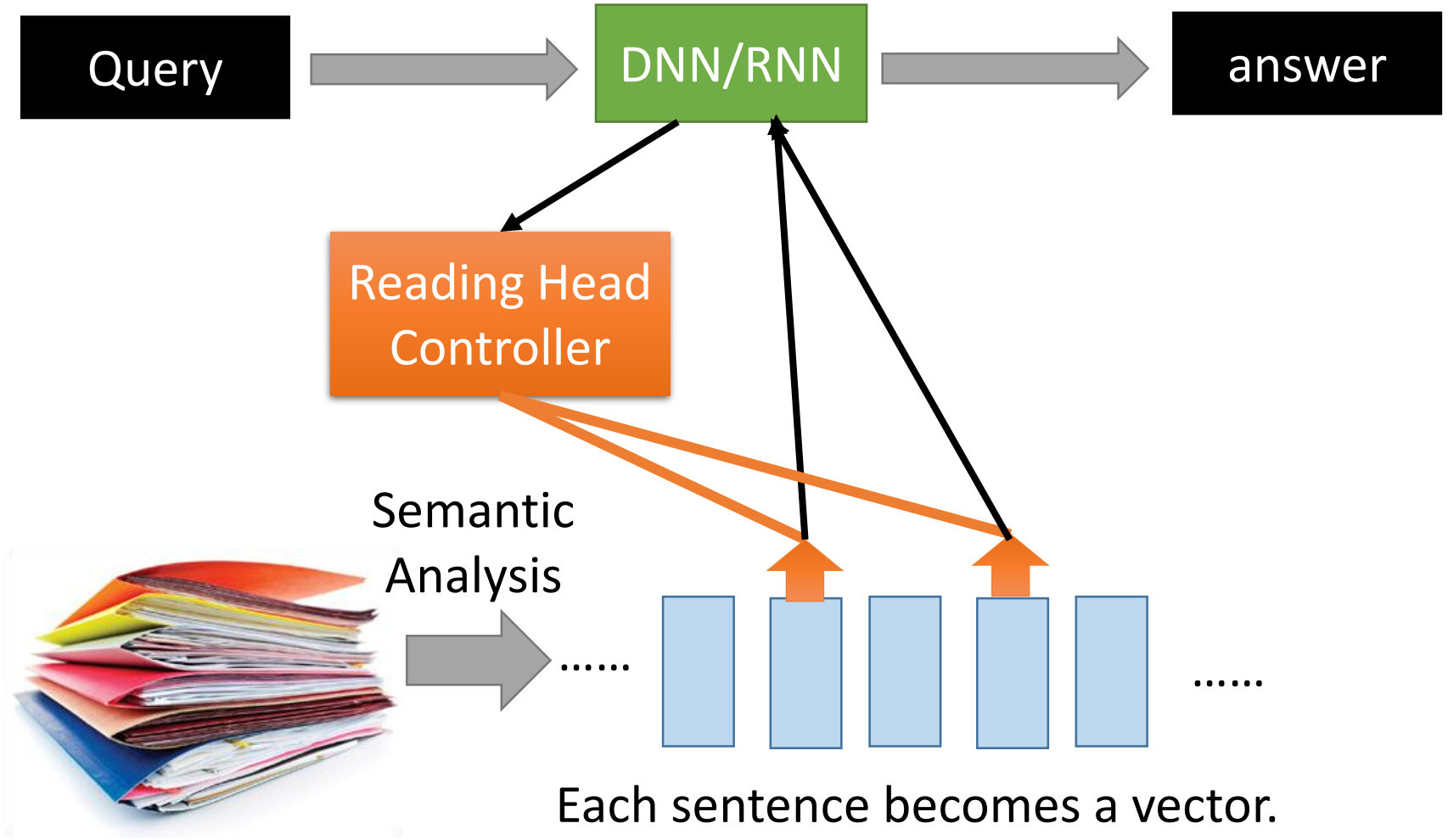


Sequence-to-sequence Auto-encoder

The RNN encoder and decoder are jointly trained.



Reading Comprehension



Deep & Structured

RNN v.s. Structured Learning

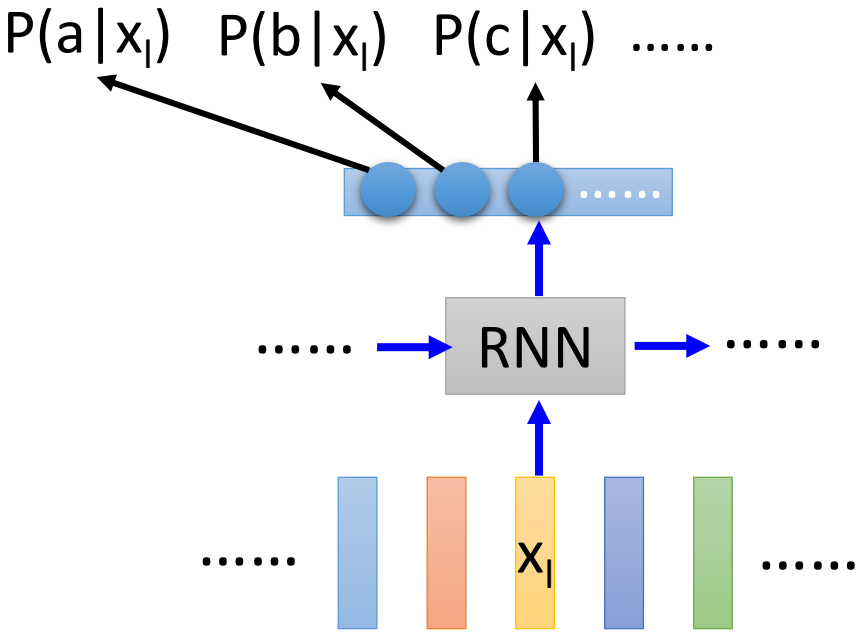
- RNN, LSTM
  - Unidirectional RNN does NOT consider the whole sequence
  - Cost and error not always related
  - Deep 勝

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

Integrated together

- Speech Recognition: CNN/LSTM/DNN + HMM

$$P(x,y) = P(y_1|start)\prod_{l=1}^{L-1} P(y_{l+1}|y_l)\, P(end|y_L)\prod_{l=1}^{L} P(x_l|y_l)$$



$$P(x_l|y_l) = \frac{P(x_l,y_l)}{P(y_l)}$$

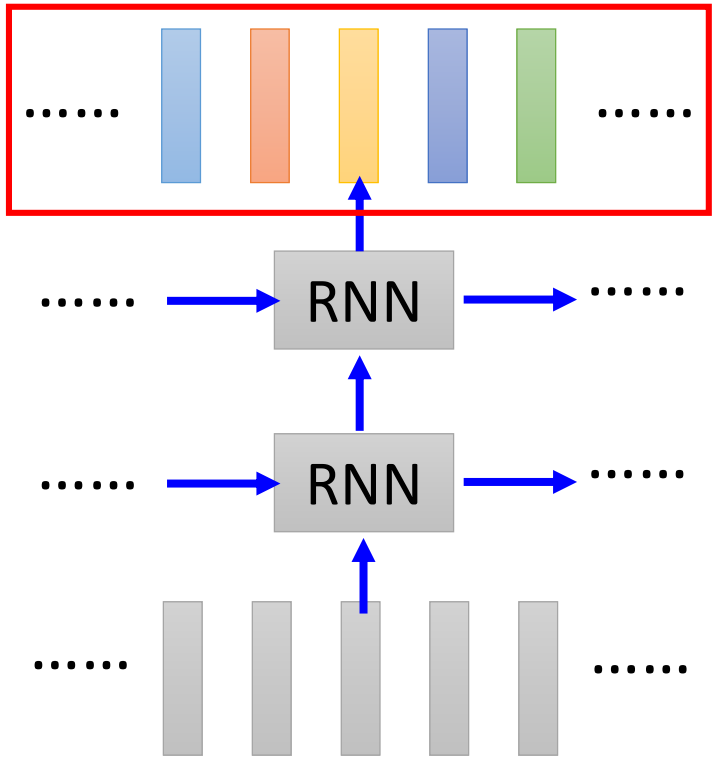
RNN

$$= \frac{P(y_l|x_l)\cancel{P(x_l)}}{P(y_l)}$$

Count

Integrated together

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



$w \cdot \phi(x, y)$

Testing:

$$\tilde{y} = \arg\max_{y \in \mathbb{Y}} w \cdot \phi(x, y)$$

Is structured learning practical?

- Conditional GAN