Deep Learning

Xiaoqing Zheng

zhengxq@fudan.edu.cn

Deep Learning

- ***Deep learning*** algorithms have been proposed in recent years to move ***machine learning*** systems towards the discovery of ***multiple levels of representation***.
- Companies like ***Google***, ***Microsoft***, ***Apple***, ***IBM*** and ***Baidu*** are investing in deep learning, with the first widely distributed products being used by consumers aimed at ***speech recognition***.
- The ***New York Times*** covered the subject twice in 2012, with front-page articles. Another series of articles (including a third New York Times article) covered a more recent event showing off the application of deep learning in amajor Kaggle competition for drug discovery (for example see "Deep Learning - The Biggest Data Science Breakthrough of the Decade"
- ***Google*** bought out ("acqui-hired") a company (***DNNresearch***) created by University of Toronto professor ***Geoffrey Hinton*** (the founder and leading researcher of deep learning) and two of his PhD students, ***Ilya Sutskever*** and ***Alex Krizhevsky***, with the press writing titles such as "***Google Hires Brains that Helped Supercharge Machine Learning***"

Challenge

Will a ***computer program*** ever be able to convert a piece of ***human language text*** into a programmer friendly ***data structure*** that describes the ***meaning*** of the natural language text?

Unfortunately, no consensus has emerged about the form or the existence of such a data structure.

Chinese word segmentation

- 组合歧义

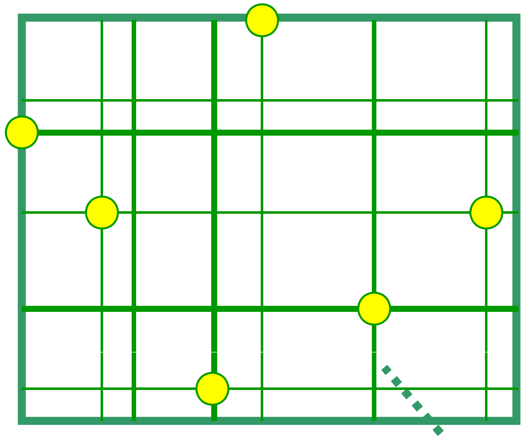
以 / 我 / 个人 / 的名义
他 /一 / 个 / 人 / 在家

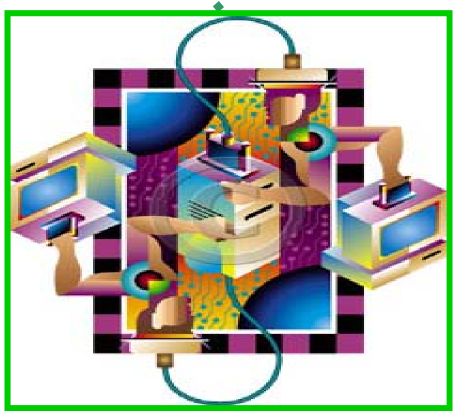
- 交叉歧义

从 / 小学 / 到 / 中学
从小 / 学 / 计算机

- 真歧义

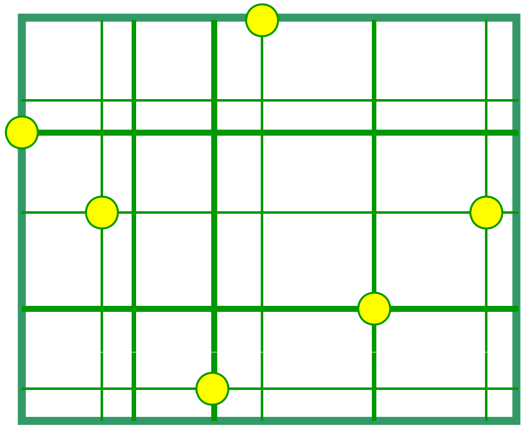
美国 / 会 / 采取 / 行动
美 / 国会 / 采取行动





Chinese word segmentation

他从小学画画
他从小学开始画画
他从小学着画画



他 / 从小 / 学 / 画画
他 / 从 / 小学 / 开始 / 画画
他 / 从小 / 学着 / 画画

数量、时间，如： “2012年5月22日” 、 “一市斤” 、 “356克”
人名、机构名、地名，如： “李维汉” 、 “阿凡提” 、 “上海浦东张江镇”
外文翻译及缩写，如： “阿诺德·施瓦辛格” 、 “萨默维尔” 、 “FDA”
专用名称的缩写，如： “复旦” 、 “中航” 、 “央行”
新词，如： “安家费” 、 “三通” 、 “菜鸟” 、 “八卦”

Chinese word

- **重叠词**

  AA式：爸爸、宝宝、星星
  AABB式：大大咧咧、形形色色、漂漂亮亮

- **派生词**

  前缀 + 词根：阿姨、老虎、老婆
  词根 + 后缀：椅子、木头、鸟儿
  词根 + 中缀 + 词根：对得起、来得及

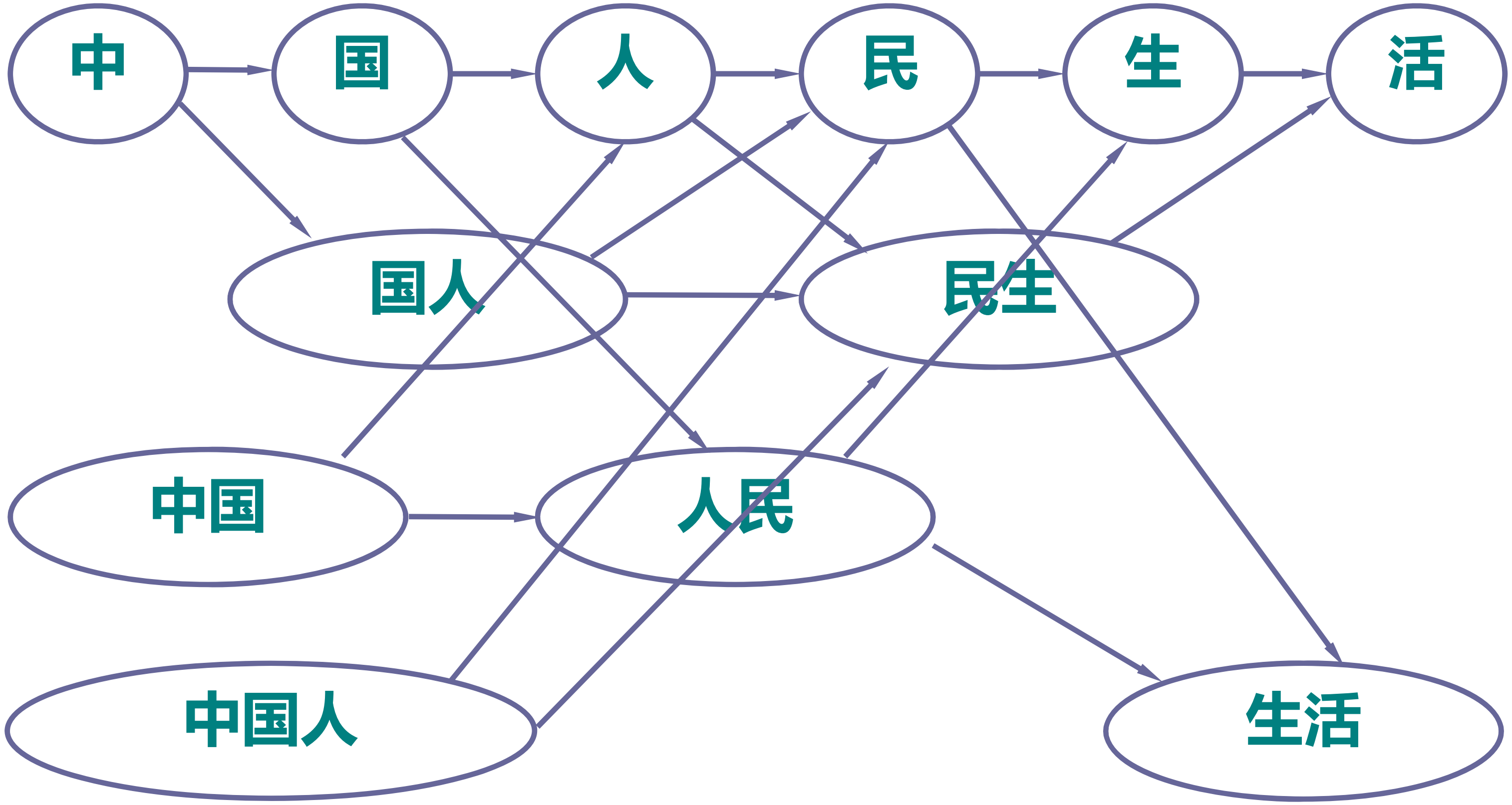
- **词转化为短语**

  动宾结构式：鞠躬 → 鞠个躬 → 鞠个九十度的躬
  偏正或联合结构：同学 → 同过三年学
  补充结构式：达到 → 达得/不到

- **短语转化为词**

  ABCD → AC式：土地改革 → 土改、地下铁道 → 地铁
  ABCD → AD式：空中小姐 → 空姐、高等院校 → 高校
  截段简缩：中国南极长城站 → 长城站、复旦大学 → 复旦
  综合简缩：联合国安全理事会 → 安全理事会 → 安理会

Chinese word segmentation



Chinese word segmentation

$$\underbrace{C_{-2}\quad C_{-1}\quad C_{0}\quad C_{1}\quad C_{2}}_{\textit{\textbf{Windows}}}$$

$$\Big\downarrow\ \textit{\textbf{Predict}}$$

The tag of C_0

Example

Corpus

S	*S*	*S*
他	恨	她

Feature template

C_0

Features

$f(S_0 = S, C_0 = \text{他}) = 1$

$f(S_0 = S, C_0 = \text{恨}) = 1$

$f(S_0 = S, C_0 = \text{她}) = 1$

Test

S	*S*	*S*
他	爱	她

	他	爱	她
	C_0	C_0	C_0
B	**0**	**1/*S***	**1/*B***
E	**0**	**1/*S***	**1/*B***
I	**0**	**1/*S***	**1/*B***
S	**1**	**1/*S***	**2/*B***
	S	***B***	***S***

Example

Corpus

S S S

他 恨 她

Feature template

C_0 $S_{-1}S_0$

Features

$f(S_0 = S, C_0 = \text{他}) = 1$

$f(S_0 = S, C_0 = \text{恨}) = 1$

$f(S_0 = S, C_0 = \text{她}) = 1$

$f(S_{-1} = S, S_0 = S) = 2$

$f(S_{-1} = NIL, S_0 = S) = 1$

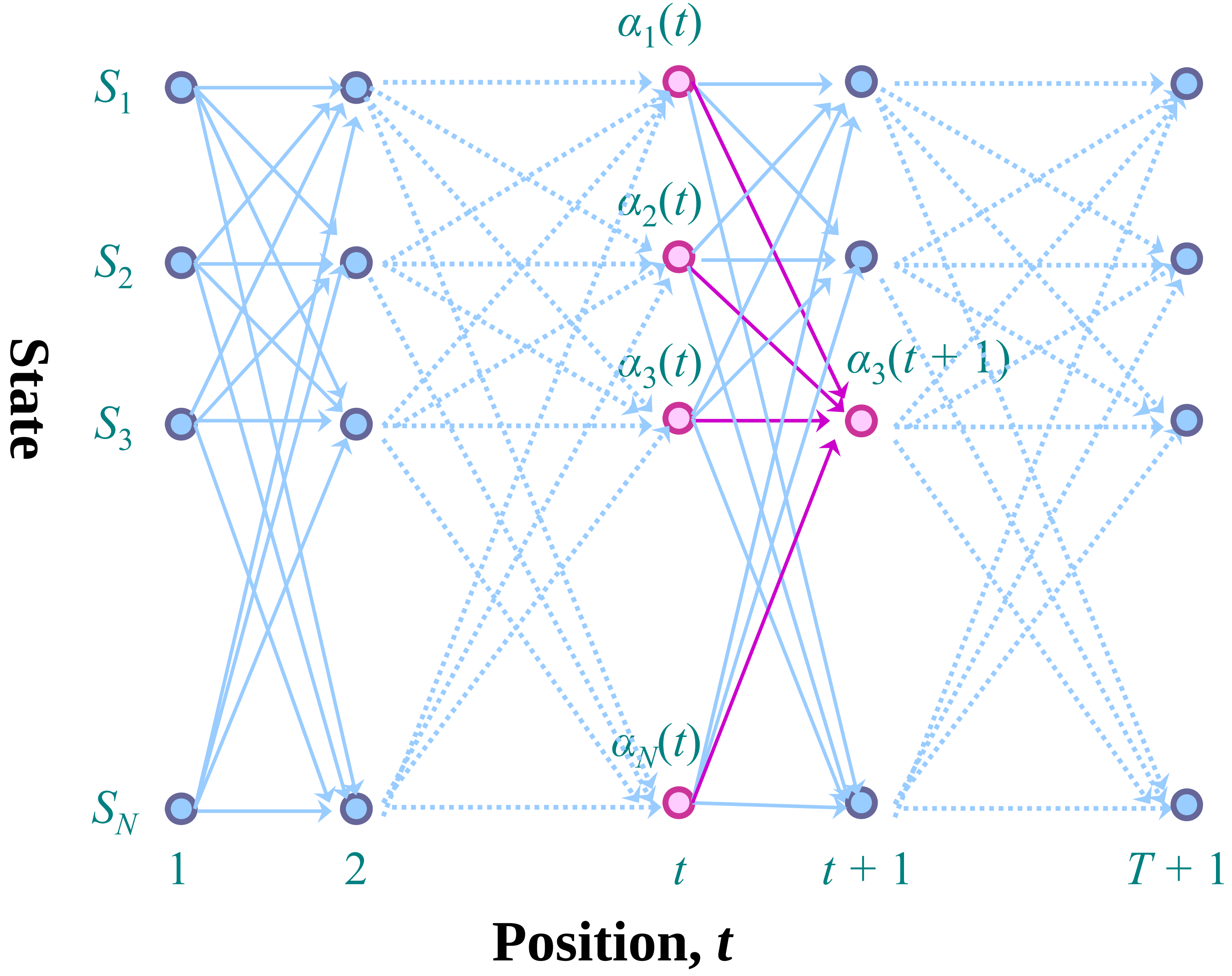
Test

S S S

他 爱 她

	他	**爱**	**她**
	C_0	C_0	C_0
B	**0**	**2/*S***	**4/*S***
E	**0**	**2/*S***	**4/*S***
I	**0**	**2/*S***	**4/*S***
S	**2**	**4/*S***	**7/*S***
	S	***S***	***S***

Viterbi algorithm



Conditional random fields

S	**B**	**E**	**S**	**B**	**E**	**S**	**S**	**B**	**E**	**S**
他	**从**	**小**	**就**	**开**	**始**	**学**	**画**	**动**	**物**	**。**

$$f(S_0 = S, O_0 = \text{就})$$

$$f(S_0 = S, O_{-1} = \text{小})$$

$$f(S_0 = S, O_{-2} = \text{从})$$

$$f(S_0 = S, O_1 = \text{开})$$

$$f(S_0 = S, O_2 = \text{始})$$

$$f(S_0 = S, O_{-1} = \text{小}, O_0 = \text{就})$$

$$f(S_0 = S, O_{-2} = \text{从}, O_0 = \text{就})$$

$$f(S_0 = S, O_0 = \text{就}, O_1 = \text{开})$$

$$f(S_0 = S, O_0 = \text{就}, O_2 = \text{始})$$

$$f(S_0 = S, O_{-1} = \text{小}, O_0 = \text{就}, O_1 = \text{开})$$

Conditional random fields

S	**B**	**E**	**S**	**B**	**E**	**S**	**S**	**B**	**E**	**S**
他	**从**	**小**	**就**	**开**	**始**	**学**	**画**	**动**	**物**	**。**

$$f(S_0 = S, S_{-1} = E, O_0 = \text{就})$$

$$f(S_0 = S, S_1 = B, O_0 = \text{就})$$

$$f(S_0 = S, S_{-2} = B, O_0 = \text{就})$$

$$f(S_0 = S, S_2 = E, O_0 = \text{就})$$

$$f(S_0 = S, S_{-1} = E, O_0 = \text{就}, O_{-1} = \text{小})$$

CRF training

S	**B**	**E**	**I**	**B**	**E**	**S**	**S**	**B**	**E**	**S**
S	**B**	**E**	**S**	**B**	**E**	**S**	**S**	**B**	**E**	**S**
他	**从**	**小**	**就**	**开**	**始**	**学**	**画**	**动**	**物**	**。**

$f(S_0, O_0)$ $\alpha(S_0 = S, O_0 = \text{就}) + +$ $\alpha(S_0 = I, O_0 = \text{就}) - -$

$f(S_0, O_{-1}, O_0)$ $\alpha(S_0 = S, O_{-1} = \text{小}, O_0 = \text{就}) + +$ $\alpha(S_0 = I, O_{-1} = \text{小}, O_0 = \text{就}) - -$

$f(S_0, O_0, O_1)$ $\alpha(S_0 = S, O_0 = \text{就}, O_1 = \text{开}) + +$ $\alpha(S_0 = I, O_0 = \text{就}, O_1 = \text{开}) - -$

$f(S_0, O_{-1}, O_1)$ $\alpha(S_0 = S, O_{-1} = \text{小}, O_1 = \text{开}) + +$ $\alpha(S_0 = I, O_{-1} = \text{小}, O_1 = \text{开}) - -$

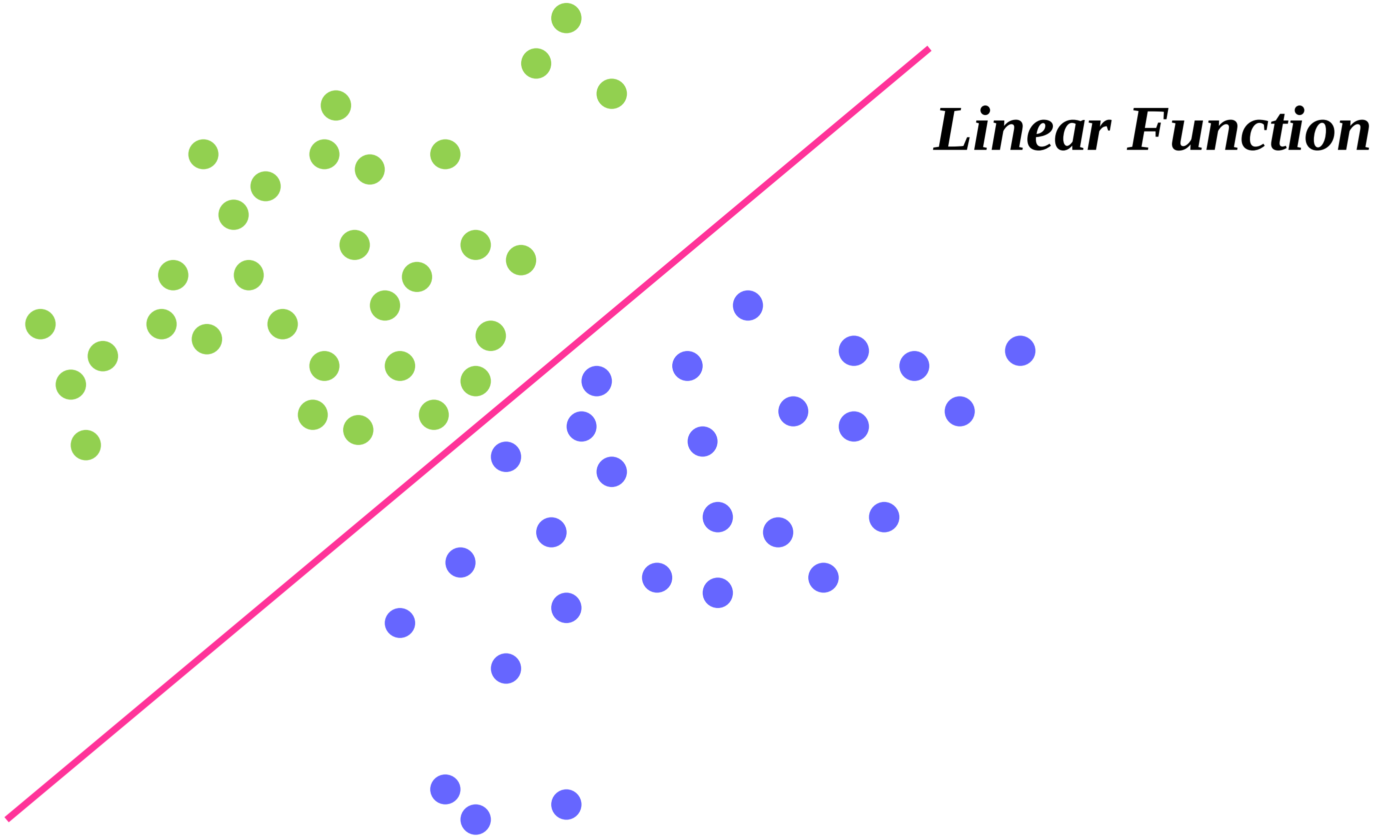
$f(S_{-1}, S_0)$ $\alpha(S_{-1} = E,\ S_0 = S) + +$ $\alpha(S_{-1} = E,\ S_0 = I) - -$

$\alpha(S_{-1} = S,\ S_0 = B) + +$ $\alpha(S_{-1} = I,\ S_0 = B) - -$

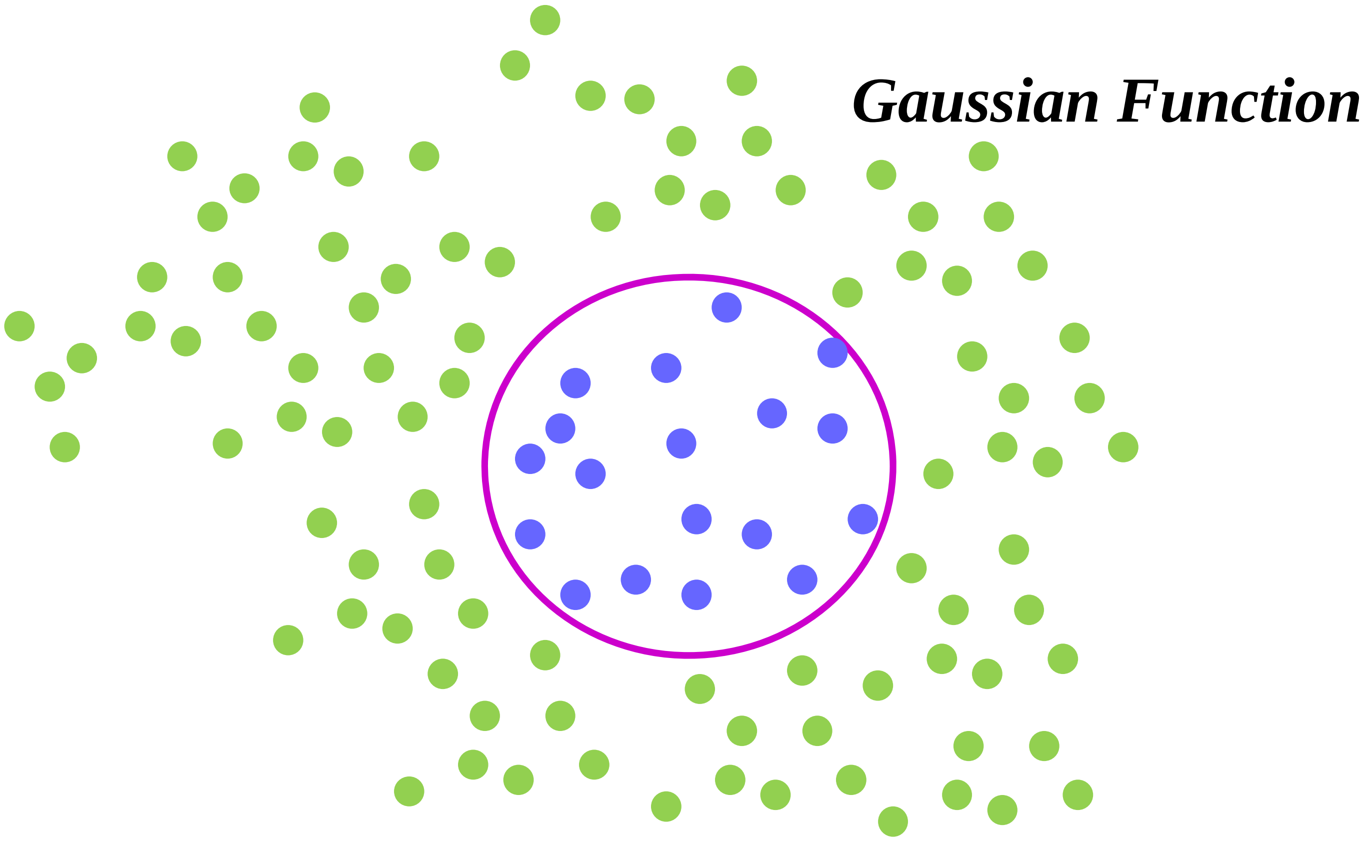
Question

- ***Why we need Deep Learning***
- ***Why we use Neural Network to implement Deep Learning***

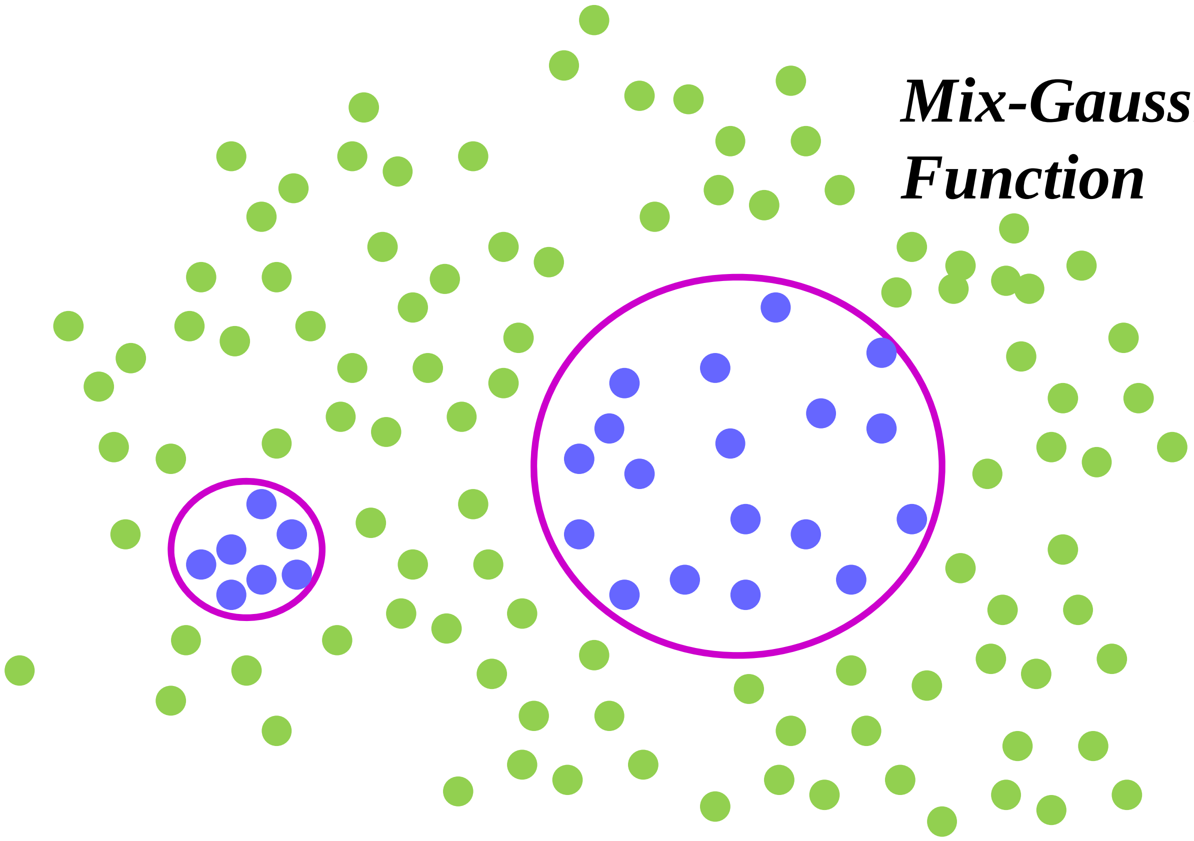
Classification



Classification

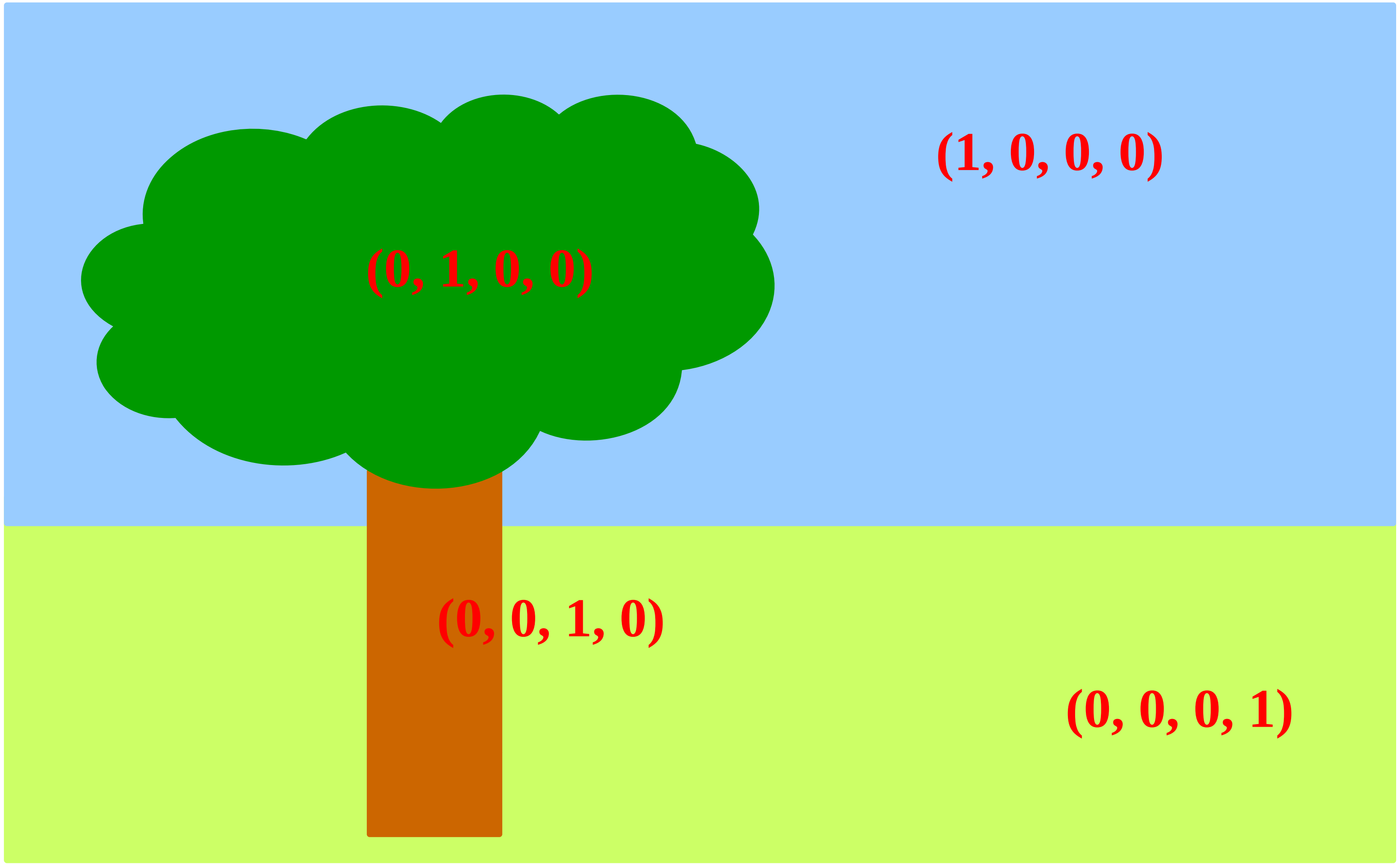


Classification

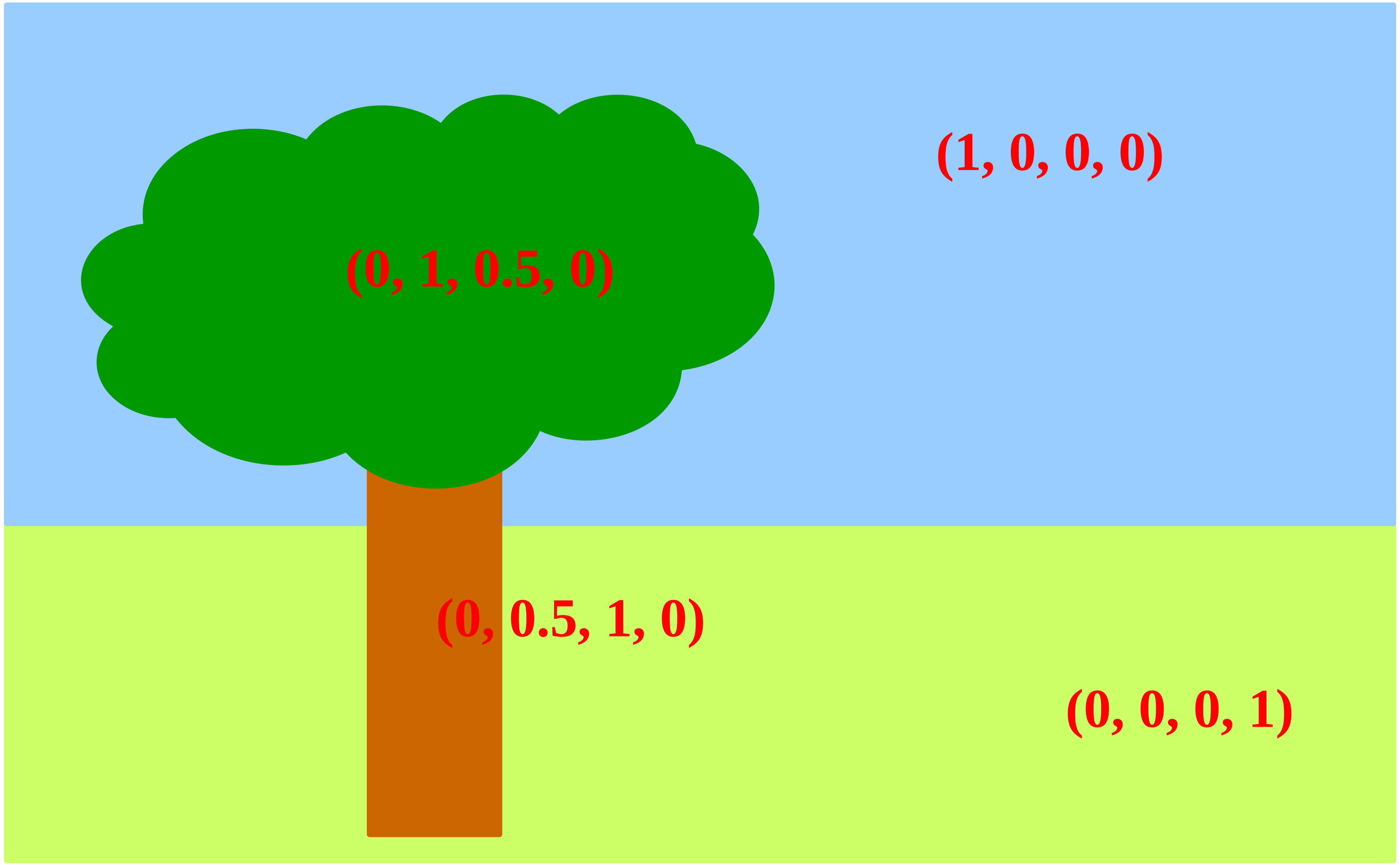


Mix-Gaussian Function

Recognition



Recognition



Architecture

$$f_\theta(\cdot) = f_\theta^L(f_\theta^{L-1}(\ldots f_\theta^1(\cdot)\ldots))$$

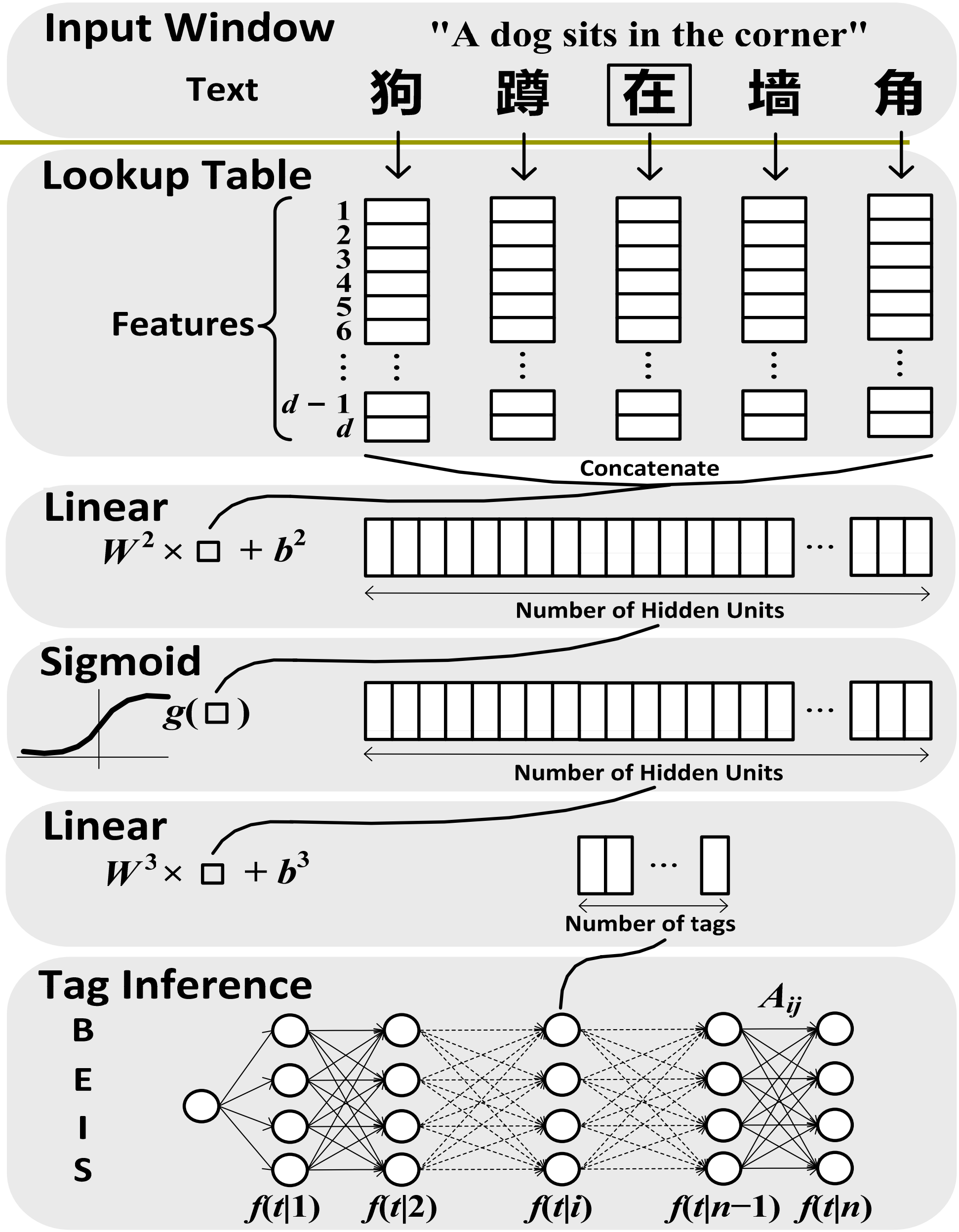
$$f_\theta^1(c_i) = \begin{pmatrix} \mathcal{Z}_\mathcal{D}(c_{i-w/2}) \\ \vdots \\ \mathcal{Z}_\mathcal{D}(c_i) \\ \vdots \\ \mathcal{Z}_\mathcal{D}(c_{i+w/2}) \end{pmatrix}$$

□

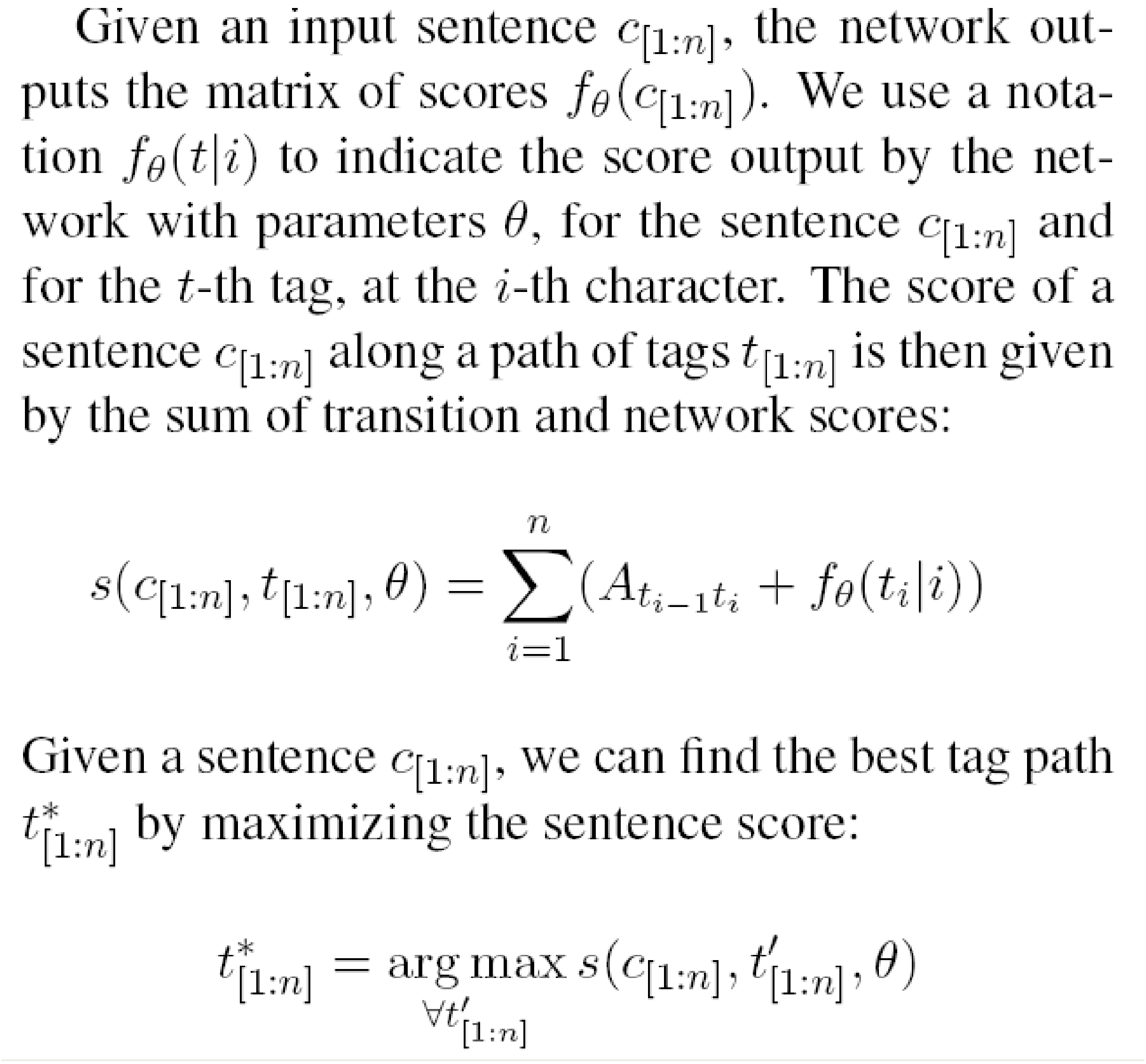
○

$$f_\theta(c_i) = f_\theta^3(g(f_\theta^2(f_\theta^1(c_i))))$$
$$= W^3 g(W^2 f_\theta^1(c_i) + b^2) + b^3$$

$$g(x) = 1/(1+e^{-x})$$



Tag inference



Given an input sentence $c_{[1:n]}$, the network outputs the matrix of scores $f_\theta(c_{[1:n]})$. We use a notation $f_\theta(t|i)$ to indicate the score output by the network with parameters θ, for the sentence $c_{[1:n]}$ and for the t-th tag, at the i-th character. The score of a sentence $c_{[1:n]}$ along a path of tags $t_{[1:n]}$ is then given by the sum of transition and network scores:

$$s(c_{[1:n]}, t_{[1:n]}, \theta) = \sum_{i=1}^{n} (A_{t_{i-1}t_i} + f_\theta(t_i|i))$$

Given a sentence $c_{[1:n]}$, we can find the best tag path $t^*_{[1:n]}$ by maximizing the sentence score:

$$t^*_{[1:n]} = \arg\max_{\forall t'_{[1:n]}} s(c_{[1:n]}, t'_{[1:n]}, \theta)$$

Training

$$\theta = (\mathcal{M}, W^2, b^2, W^3, b^3, A)$$

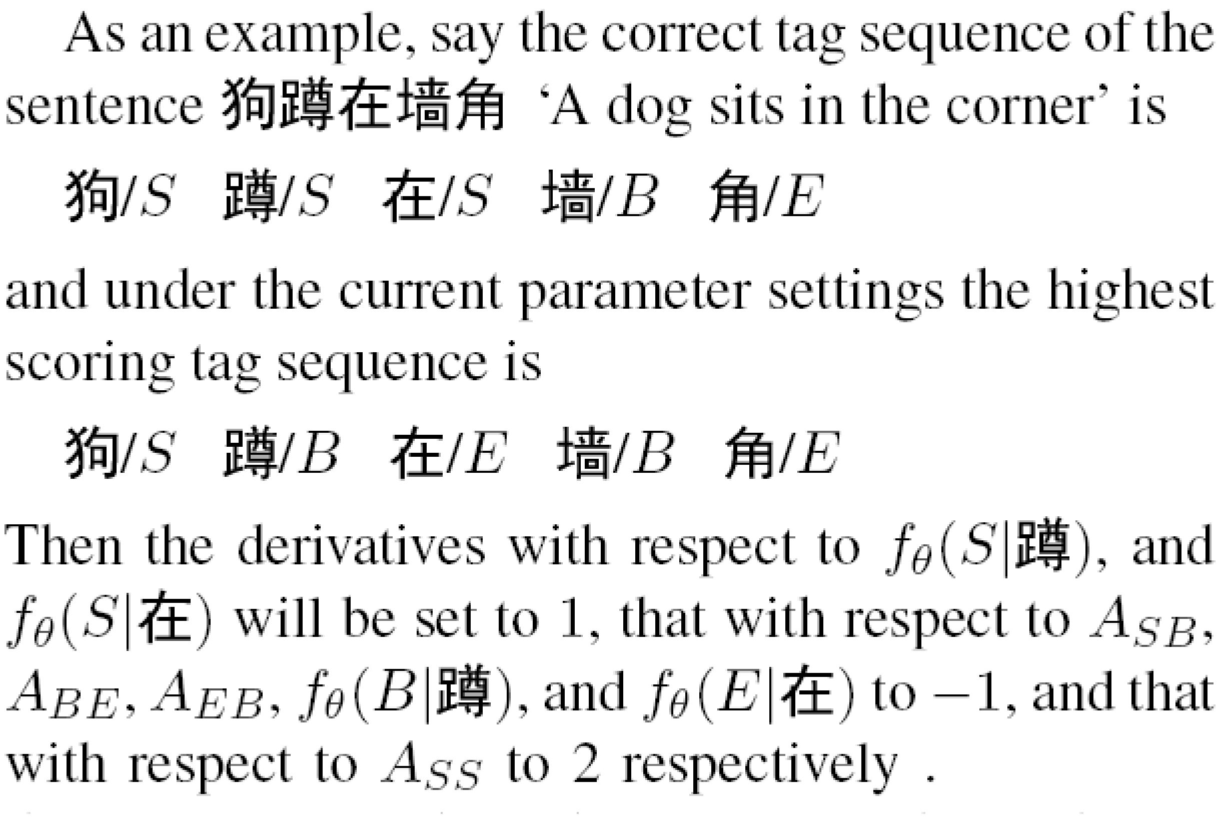
$$\theta \mapsto \sum_{\forall (c,t) \in \mathcal{R}} \log p(t|c,\theta)$$

$$\theta \leftarrow \theta + \lambda \frac{\partial \log p(t|c,\theta)}{\partial \theta}$$

$$\log p(t|c,\theta) = s(c,t,\theta) - \log \sum_{\forall t'} \exp\{s(c,t',\theta)\}$$

Perceptron-style algorithm

$$\frac{\partial L_\theta(t,t'|c)}{\partial f_\theta(t_i|i)}{++}, \quad \frac{\partial L_\theta(t,t'|c)}{\partial f_\theta(t'_i|i)}{--} \qquad \frac{\partial L_\theta(t,t'|c)}{\partial A_{t_{i-1}t_i}}{++}, \quad \frac{\partial L_\theta(t,t'|c)}{\partial A_{t'_{i-1}t'_i}}{--}$$



As an example, say the correct tag sequence of the sentence 狗蹲在墙角 ‘A dog sits in the corner’ is

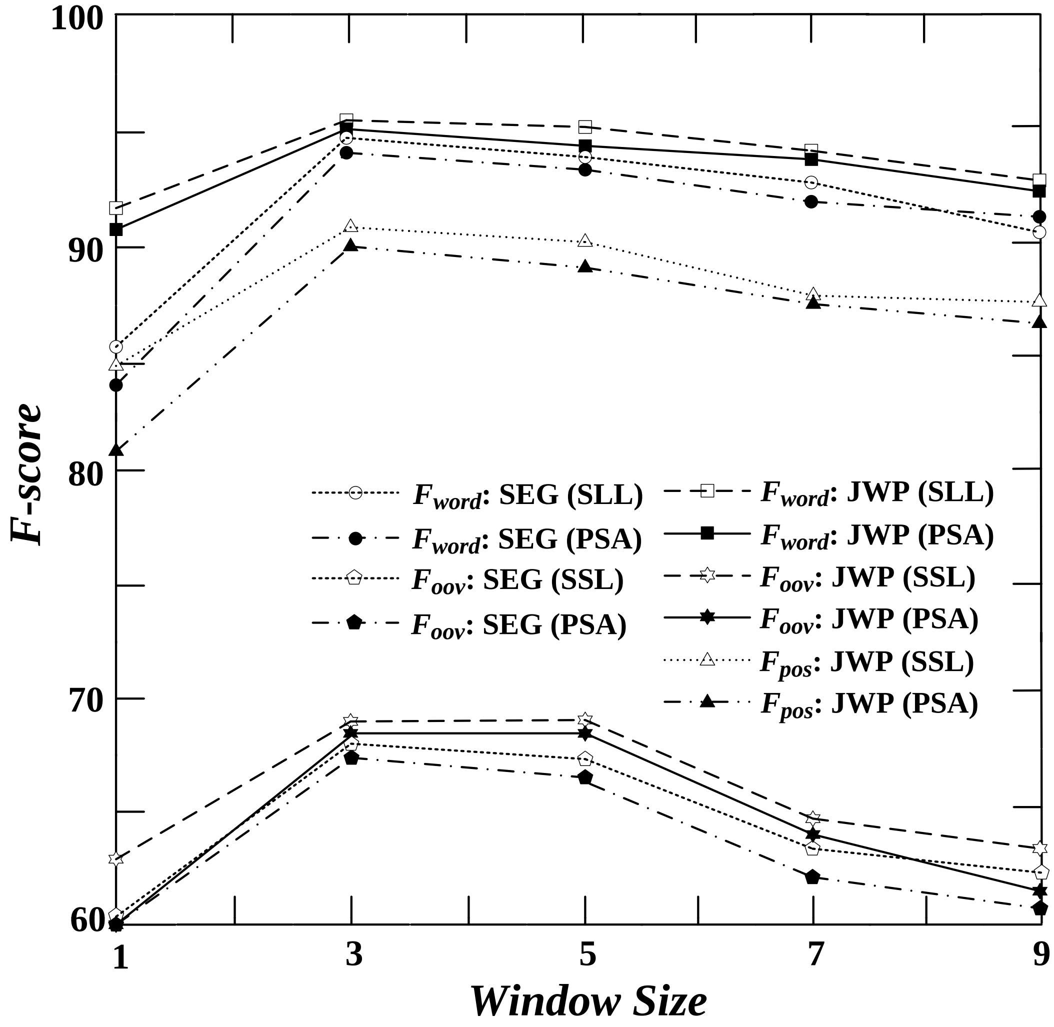
狗/S 蹲/S 在/S 墙/B 角/E

and under the current parameter settings the highest scoring tag sequence is

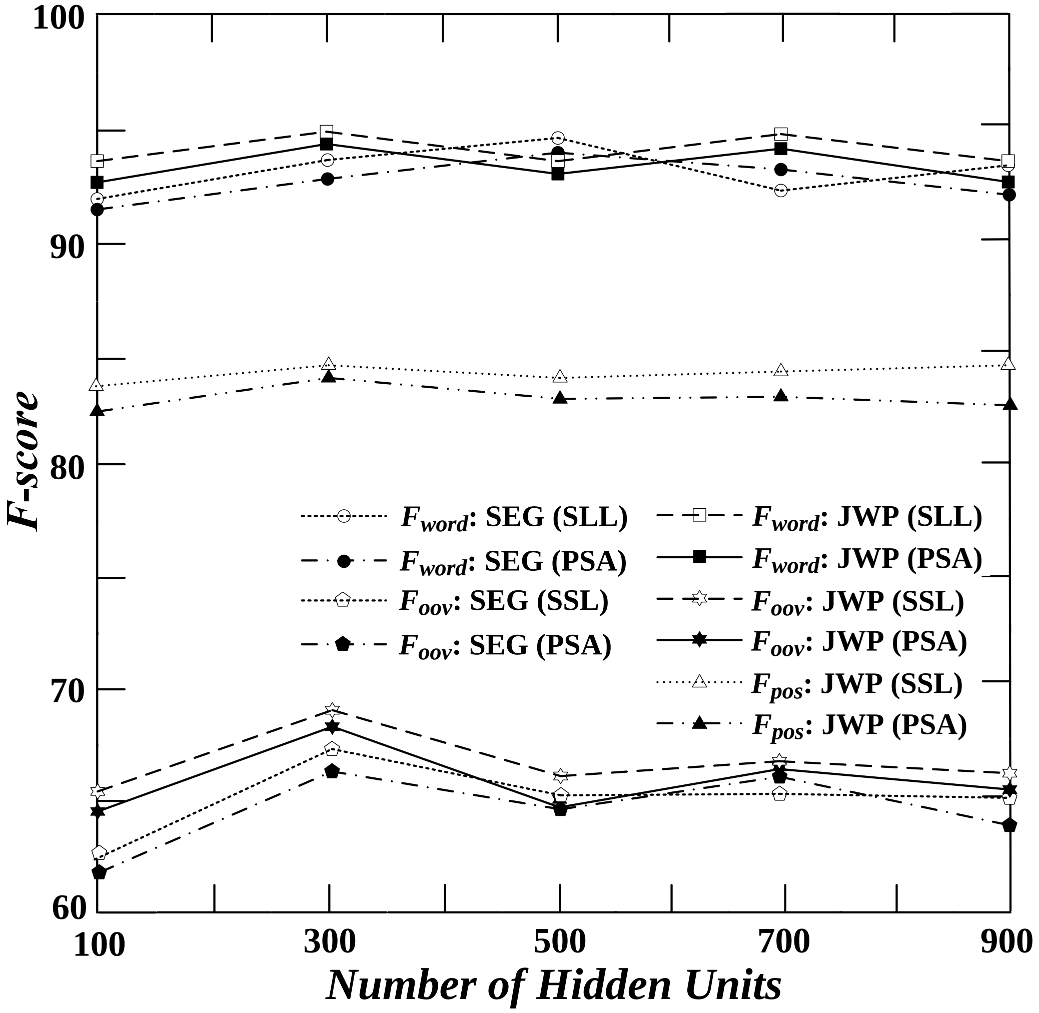
狗/S 蹲/B 在/E 墙/B 角/E

Then the derivatives with respect to $f_\theta(S|$蹲$)$, and $f_\theta(S|$在$)$ will be set to 1, that with respect to A_{SB}, A_{BE}, A_{EB}, $f_\theta(B|$蹲$)$, and $f_\theta(E|$在$)$ to -1, and that with respect to A_{SS} to 2 respectively .

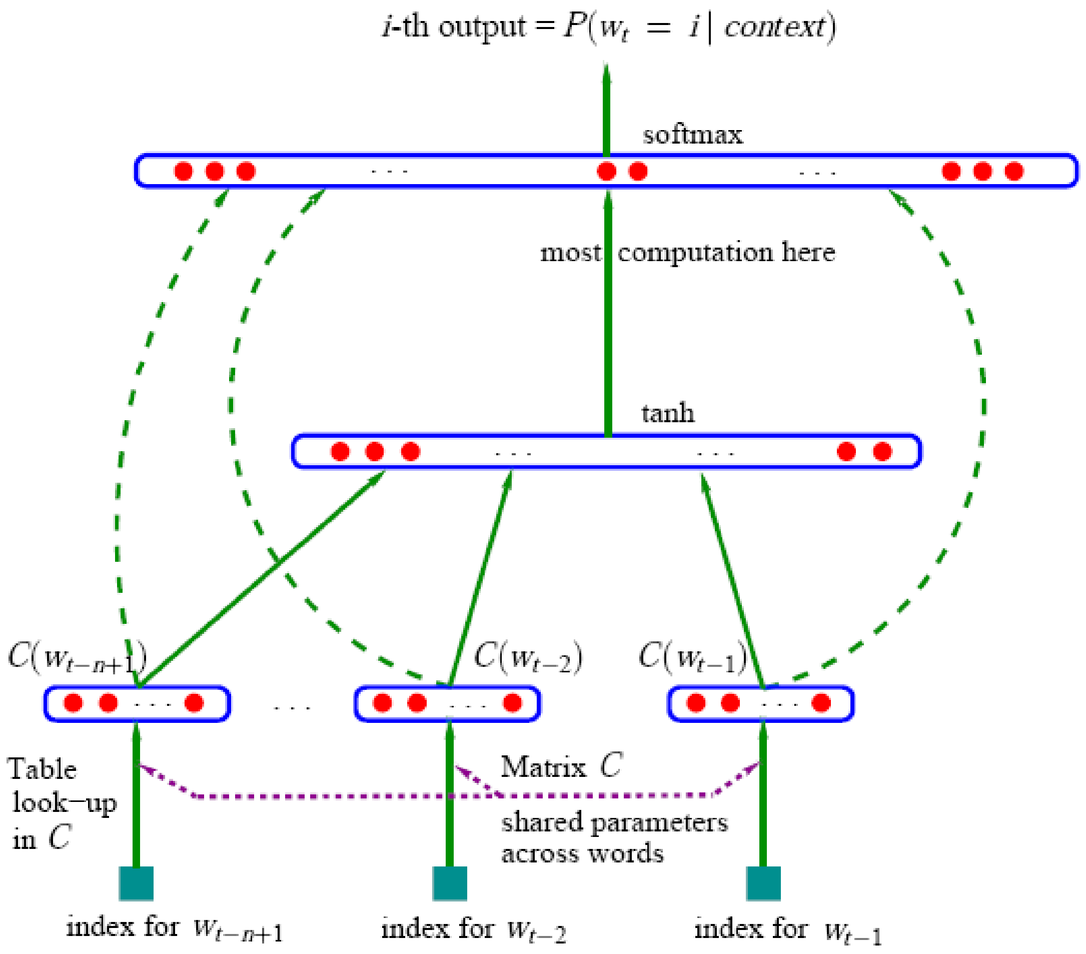
Window size


100
90
80
70
60
1
3
5
7
9
F-score
Window Size
F_word: SEG (SLL)
F_word: SEG (PSA)
F_oov: SEG (SSL)
F_oov: SEG (PSA)
F_word: JWP (SLL)
F_word: JWP (PSA)
F_oov: JWP (SSL)
F_oov: JWP (PSA)
F_pos: JWP (SSL)
F_pos: JWP (PSA)

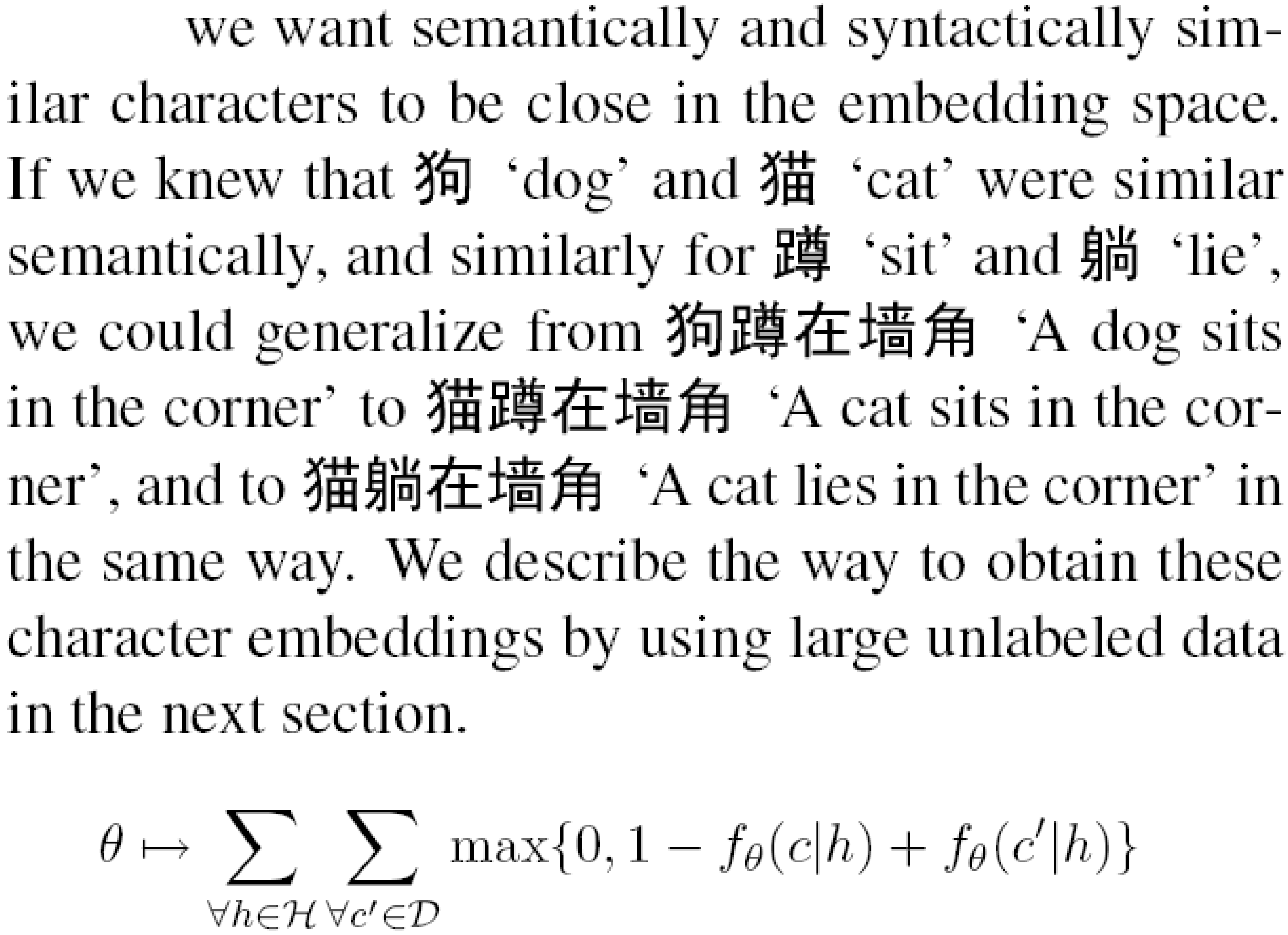
Number of hidden units



Bengio 2003 (Université de Montréal)



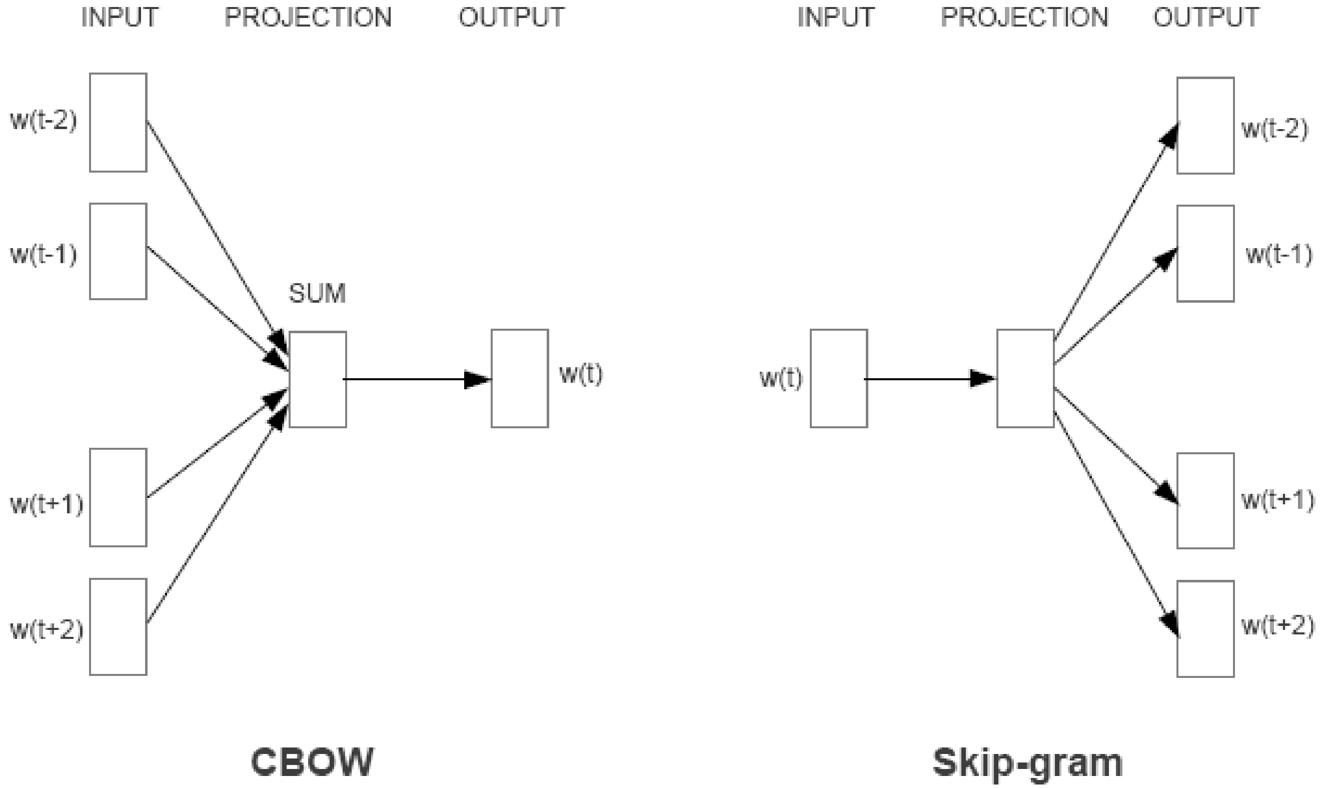
Collobert 2008 (NEC lab)



we want semantically and syntactically similar characters to be close in the embedding space. If we knew that 狗 ‘dog’ and 猫 ‘cat’ were similar semantically, and similarly for 蹲 ‘sit’ and 躺 ‘lie’, we could generalize from 狗蹲在墙角 ‘A dog sits in the corner’ to 猫蹲在墙角 ‘A cat sits in the corner’, and to 猫躺在墙角 ‘A cat lies in the corner’ in the same way. We describe the way to obtain these character embeddings by using large unlabeled data in the next section.

$$\theta \mapsto \sum_{\forall h \in \mathcal{H}} \sum_{\forall c' \in \mathcal{D}} \max\{0, 1 - f_\theta(c|h) + f_\theta(c'|h)\}$$

Mikolov 2013 (Google)



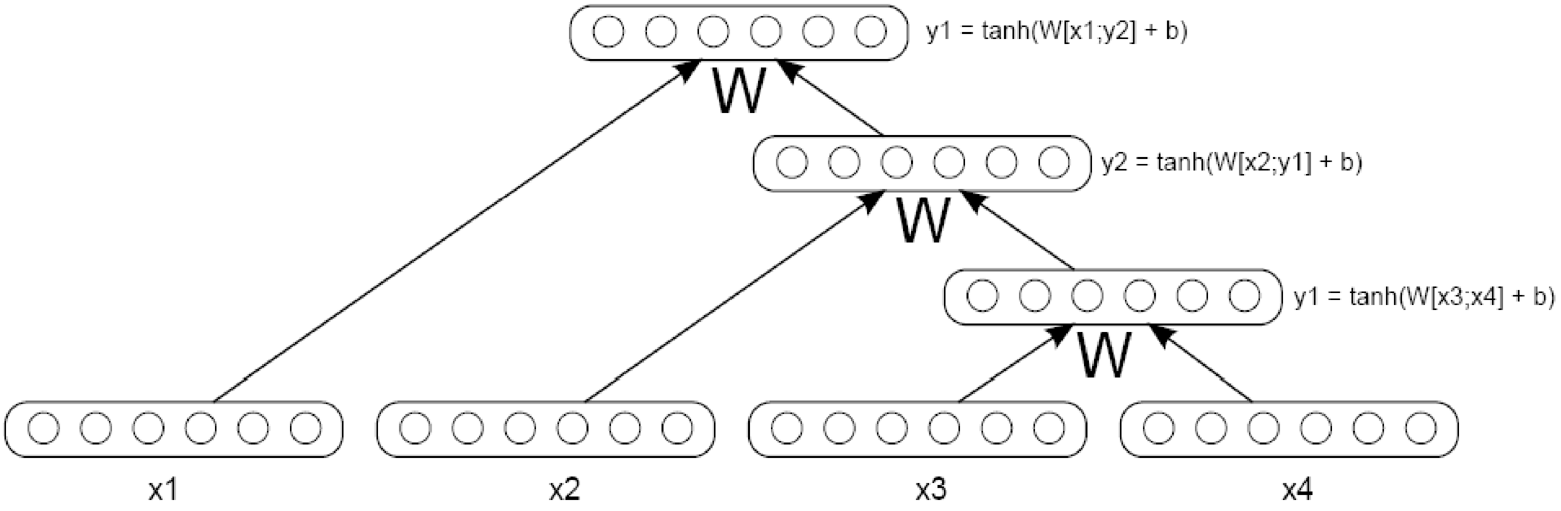
Word2Vec tool

CTB-3 dataset

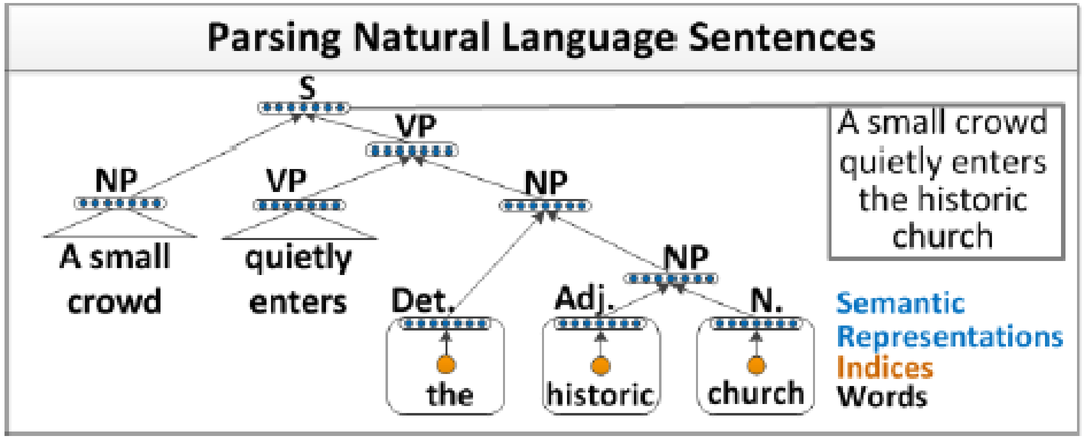
Approach		F_{word}	R_{oov}	F_{pos}
SEG	(Zhao et al., 2006)	93.30	70.70	–
	(Wang et al., 2006)	93.00	68.30	–
	(Zhu et al., 2006)	92.70	63.40	–
	(Zhang et al., 2006)	92.60	61.70	–
	(Feng et al., 2006)	91.70	68.00	–
	PSA	92.59	64.24	–
	PSA + LM	94.57	70.12	–
JWP	(Ng and Lou, 2004)	95.20	–	–
	(Zhang and Clark, 2008)	95.90	–	91.34
	(Jiang et al., 2008)	97.30	–	92.50
	(Kruengkrai et al., 2009)	96.11	–	90.85
	PSA	93.83	68.21	90.79
	PSA + LM	95.23	72.38	91.82

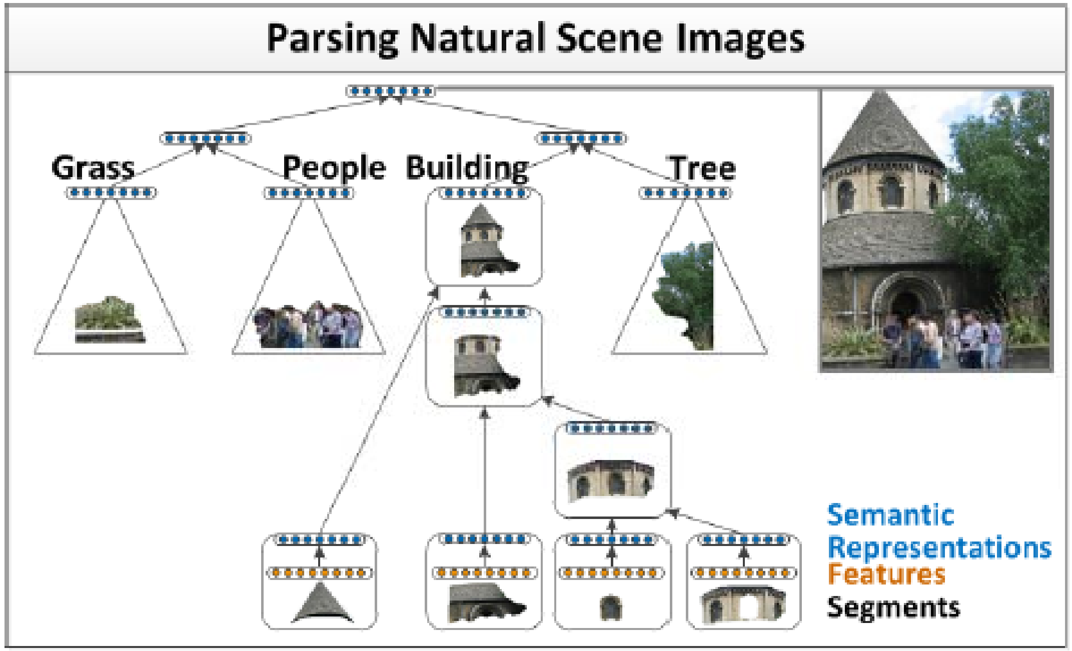
System	**Number of parameters**	**Time (s)**
(Tsai et al., 2006)	3.1×10^6	1669
(Zhao et al., 2006)	3.8×10^6	2382
Neural network	4.7×10^5	138

Recursive Neural Networks

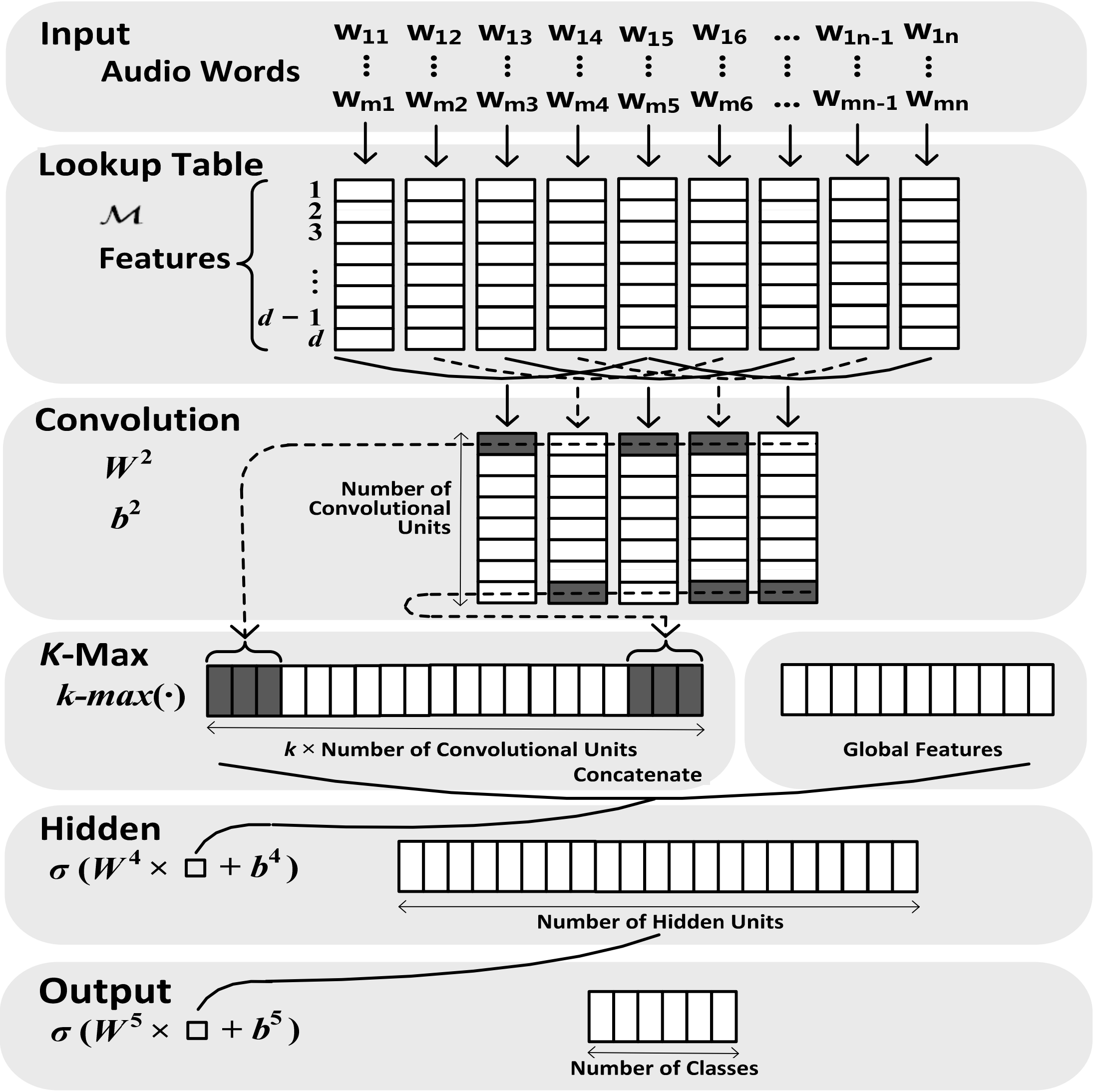


RNN


Parsing Natural Language Sentences
S
VP
NP
VP
NP
A small crowd
quietly enters
NP
Det.
Adj.
N.
the
historic
church
A small crowd quietly enters the historic church
Semantic Representations
Indices
Words


Parsing Natural Scene Images
Grass
People
Building
Tree
Semantic Representations
Features
Segments

Convolutional networks


Input
Audio Words
w_{11} w_{12} w_{13} w_{14} w_{15} w_{16} … w_{1n-1} w_{1n}
w_{m1} w_{m2} w_{m3} w_{m4} w_{m5} w_{m6} … w_{mn-1} w_{mn}
Lookup Table
$\mathcal{M}$
Features
1
2
3
$d-1$
d
Convolution
W^2
b^2
Number of Convolutional Units
K-Max
$k\text{-}max(\cdot)$
k × Number of Convolutional Units
Concatenate
Global Features
Hidden
$\sigma(W^4 \times \square + b^4)$
Number of Hidden Units
Output
$\sigma(W^5 \times \square + b^5)$
Number of Classes

Any question?

Xiaoqing Zheng

Fudan University