Natural Language Processing (5)

Collocations, Language Models, and Recurrent Neural Networks

Daisuke Kawahara

Department of Communications and Computer Engineering,
Waseda University

Lecture Plan

1. Overview of Natural Language Processing
2. Formal Language Theory
3. Word Senses and Embeddings
4. Topic Models
5. Collocations, Language Models, and Recurrent Neural Networks
6. Sequence Labeling and Morphological Analysis
7. Parsing (1)
8. Parsing (2)
9. Transfer Learning
10. Knowledge Acquisition
11. Information Retrieval, Question Answering, and Machine Translation
12. Guest Talk (1)
13. Guest Talk (2)
14. Project: Survey or Programming
15. Project Presentation

Collocations

Collocations

- Non-compositionality (in meaning)
  - cannot derive the meaning from parts
  - e.g., kick the bucket, white wine/hair/woman
- Non-substitutability (in context)
  - e.g., white wine, strong tea
- Non-modifiability
  - e.g., get a frog in one's throat (声がガラガラ)

Associations and Co-occurrences

- Do not fall under “collocations”, but:
- Interesting just because it does often appear together or in the similar context
  - (doctor, nurse)
  - (plane, airport)
  - (gas, fuel)

Subclasses of Collocations

- Light verbs
  - make, do, take, …
    - make a decision, do a favor
- Verb particles (phrasal verbs)
  - tell off (叱る)
  - take off
- Proper nouns (proper names)
  - Washington, D.C.
- Terminological expressions
  - hydraulic oil filter

How to Find Collocations?

- Frequency
  - plain
  - filtered
- Mean and variance
- Hypothesis testing
  - t test
  - χ^2 test
- Likelihood ratio
- Pointwise mutual information

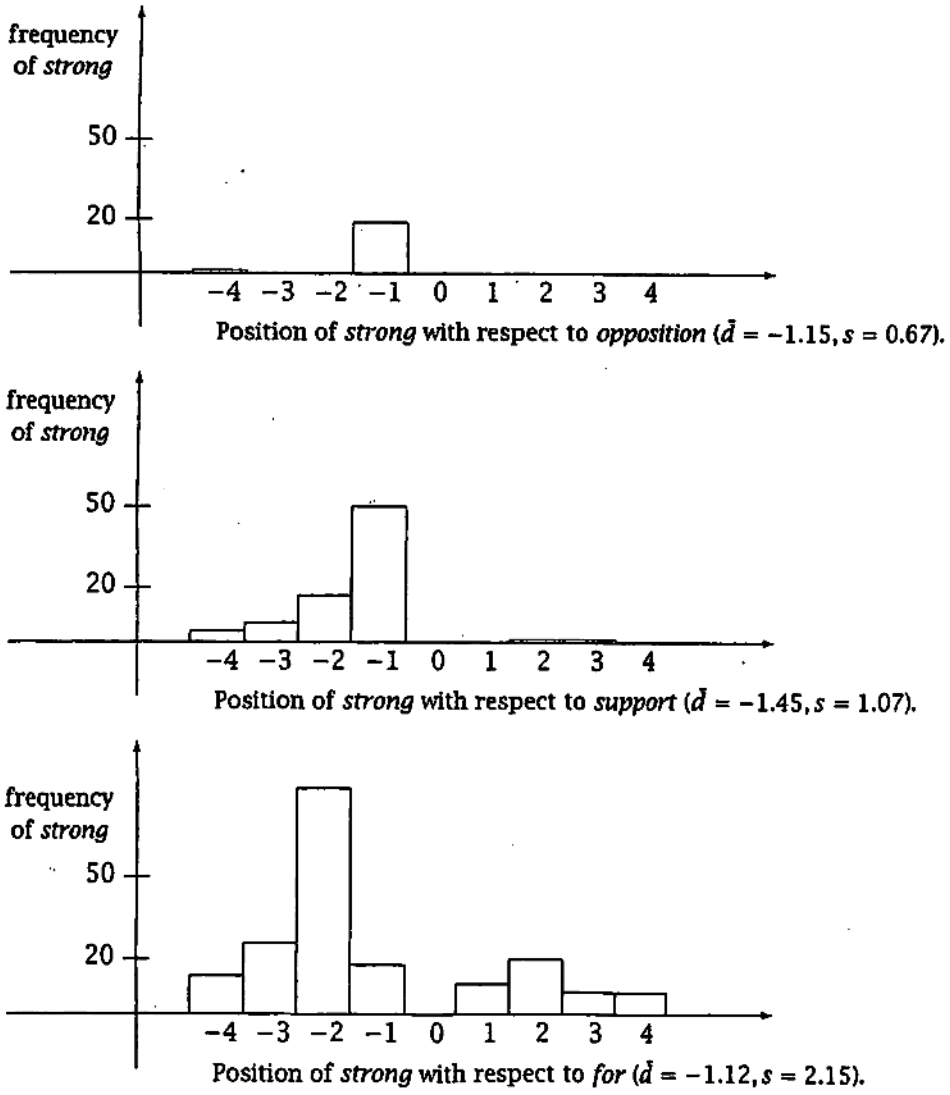
Frequency

- Plain frequency
  - Count n-grams; high frequency n-grams are candidates
    - of the 80871
    - in the 58841
    - to the 26430
    - …
- Filtered frequency (using parts of speech)
  - Adj+Noun, Noun+Noun, Noun+Prep+Noun, …
    - New York 11487
    - United States 7261
    - Los Angeles 5412
    - last year 3301
    - …

Mean and Variance

- Many collocations consist of two words that stand in a more flexible relationship to one another
- Examples (*knocked door*):
  - she knocked on his door
  - they knocked at the door
  - 100 woman knocked on Donaldson's door
  - a man knocked on the metal front door
- Compute the mean and variance of the distances between the two words

$$d = \frac{1}{4}(3+3+5+5) = 4.0 \qquad s = 1.15$$



s	$\bar{d}$	Count	Word 1	Word 2
0.43	0.97	11657	New	York
0.48	1.83	24	previous	games
0.15	2.98	46	minus	points
0.49	3.87	131	hundreds	dollars
4.03	0.44	36	editorial	Atlanta
4.03	0.00	78	ring	New
3.96	0.19	119	point	hundredth
3.96	0.29	106	subscribers	by
1.07	1.45	80	strong	support
1.13	2.57	7	powerful	organizations
1.01	2.00	112	Richard	Nixon
1.05	0.00	10	Garrison	said

Histograms of the position of *strong* relative to three words (*opposition*, *support*, and *for*)

[Manning & Schütze 1999]

Hypothesis Testing (t test)

- t statistic: $$t = \frac{\bar{x} - \mu}{\sqrt{\frac{s^2}{N}}}$$
- Example
  - Null hypothesis: the mean height of a population of men is 158cm ($\mu = 158$)
  - We are given a sample of 200 men ($N = 200$) with:
    - $\bar{x} = 169$ and $s^2 = 2600$

$$t = \frac{169 - 158}{\sqrt{\frac{2600}{200}}} \approx 3.05 > 2.576$$

(confidence level: $\alpha = 0.005$)

t test for Finding Collocations

- Let us compute the t value for *new companies*
- Null hypothesis: occurrences of *new* and *companies* are independent

	w1=new	w1≠new
w2=companies	8 (new companies)	4667 (e.g., old companies)
w2≠companies	15820 (e.g., new machines)	14287173

$$P(new\ companies) = P(new)P(companies)$$

$$= \frac{15828}{14307668} \times \frac{4675}{14307668} \approx 3.615 \times 10^{-7} \quad = \mu$$

$$s^2 = p(1-p) \approx p = \bar{x} \qquad \bar{x} = \frac{8}{14307668} \approx 5.591 \times 10^{-7} \qquad t \approx 0.999932 < 2.576$$

t	$C(w^1)$	$C(w^2)$	$C(w^1\ w^2)$	w^1	w^2
4.4721	42	20	20	Ayatollah	Ruhollah
4.4721	41	27	20	Bette	Midler
4.4720	30	117	20	Agatha	Christie
4.4720	77	59	20	videocassette	recorder
4.4720	24	320	20	unsalted	butter
2.3714	14907	9017	20	first	made
2.2446	13484	10570	20	over	many
1.3685	14734	13478	20	into	them
1.2176	14093	14776	20	like	people
0.8036	15019	15629	20	time	last

t scores applied to 10 bigrams that occur with frequency 20

[Manning & Schütze 1999]

Pearson's Chi-square test

- χ^2 test:

$$X^2 = \sum_{i,j} \frac{\left(O_{ij} - E_{ij}\right)^2}{E_{ij}}$$

O_{ij} : Observed count of events (i,j)

E_{ij} : Expected count of events (i,j)

- Example

	w1=new	w1≠new
w2=companies	8 (new companies)	4667 (e.g., old companies)
w2≠companies	15820 (e.g., new machines)	14287173

$$E_{new,companies} = N \times P(new)P(companies)$$

$$= 14307668 \times \frac{15828}{14307668} \times \frac{4675}{14307668} \approx 5.2$$

$$X^2 \approx 1.55 < 3.841$$

(confidence level: α = 0.05)

Likelihood Ratio

- H_1: independent

$$P(w_2 \mid w_1) = P(w_2 \mid \neg w_1) = p$$

- H_2: dependent

$$P(w_2 \mid w_1) = p_1 \quad \neq \quad P(w_2 \mid \neg w_1) = p_2$$

$$p = \frac{C_2}{N}$$

$$p_1 = \frac{C_{12}}{C_1}$$

$$p_2 = \frac{C_2 - C_{12}}{N - C_1}$$

$$L(H_1) = \begin{pmatrix} C_1 \\ C_{12} \end{pmatrix} p^{C_{12}} (1-p)^{C_1 - C_{12}} \times \begin{pmatrix} N - C_1 \\ C_2 - C_{12} \end{pmatrix} p^{C_2 - C_{12}} (1-p)^{N - C_1 - C_2 + C_{12}}$$

$$L(H_2) = \begin{pmatrix} C_1 \\ C_{12} \end{pmatrix} p_1^{C_{12}} (1-p_1)^{C_1 - C_{12}} \times \begin{pmatrix} N - C_1 \\ C_2 - C_{12} \end{pmatrix} p_2^{C_2 - C_{12}} (1-p_2)^{N - C_1 - C_2 + C_{12}}$$

$$\log \lambda = \log \frac{L(H_1)}{L(H_2)}$$

$-2 \log \lambda$	$C(w^1)$	$C(w^2)$	$C(w^1 w^2)$	w^1	w^2
1291.42	12593	932	150	most	powerful
99.31	379	932	10	politically	powerful
82.96	932	934	10	powerful	computers
80.39	932	3424	13	powerful	force
57.27	932	291	6	powerful	symbol
51.66	932	40	4	powerful	lobbies
51.52	171	932	5	economically	powerful
51.05	932	43	4	powerful	magnet
50.83	4458	932	10	less	powerful
50.75	6252	932	11	very	powerful
49.36	932	2064	8	powerful	position
48.78	932	591	6	powerful	machines
47.42	932	2339	8	powerful	computer
43.23	932	16	3	powerful	magnets
43.10	932	396	5	powerful	chip
40.45	932	3694	8	powerful	men
36.36	932	47	3	powerful	486
36.15	932	268	4	powerful	neighbor
35.24	932	5245	8	powerful	political
34.15	932	3	2	powerful	cudgels

Bigrams of *powerful* with the highest likelihood ratio

[Manning & Schütze 1999]

Pointwise Mutual Information (PMI)

- This is NOT the MI as defined in information theory, which is the average of the following

$$I(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

- Example

$$I(new, companies) = \log_2 \frac{\frac{8}{14307668}}{\frac{15828}{14307668} \times \frac{4675}{14307668}} \approx 0.63$$

$I(w^1, w^2)$	$C(w^1)$	$C(w^2)$	$C(w^1\ w^2)$	w^1	w^2
18.38	42	20	20	Ayatollah	Ruhollah
17.98	41	27	20	Bette	Midler
16.31	30	117	20	Agatha	Christie
15.94	77	59	20	videocassette	recorder
15.19	24	320	20	unsalted	butter
1.09	14907	9017	20	first	made
1.01	13484	10570	20	over	many
0.53	14734	13478	20	into	them
0.46	14093	14776	20	like	people
0.29	15019	15629	20	time	last

10 bigrams that occur with frequency 20, ranked according to PMI

[Manning & Schütze 1999]

N-gram Language Models

Noisy Channel Model

- Input
- Output (observation)

- Predict the input from the output

$$\arg\max_i P(i \mid o) = \arg\max_i \frac{P(o \mid i)P(i)}{P(o)}$$

- Applications
  - Speech recognition
  - Machine translation

N-gram Language Model

$$P(w_n \mid w_1, \cdots, w_{n-1})$$

- n-gram = (n-1)th order Markov assumption

  Sue swallowed the large green ____
  (飲み込む)

  pill
  frog
  tree
  car
  mountain

- parameters
  - bigram: 20000^2 = 400 million
  - trigram: 20000^3 = 8 trillion

Google N-gram

- 1～5-grams are extracted from a Web corpus of 1 trillion words
  - Number of words: 1,024,908,267,229
  - Number of sentences: 95,119,665,584
  - Number of unigrams: 13,588,391
  - Number of fivegrams: 1,176,470,663

Examples of Google n-grams

3-gram

ceramics collectables collectibles	55
ceramics collectables fine	130
ceramics collected by	52
ceramics collectible pottery	50
ceramics collectibles cooking	45
ceramics collection ,	144
ceramics collection .	247
ceramics collection	120
ceramics collection and	43
ceramics collection at	52
ceramics collection is	68
ceramics collection of	76
ceramics collection |	59
ceramics collections ,	66
ceramics collections .	60
ceramics combined with	46
ceramics come from	69
ceramics comes from	660
ceramics community ,	109
ceramics community .	212

4-gram

serve as the incoming	92
serve as the incubator	99
serve as the independent	794
serve as the index	223
serve as the indication	72
serve as the indicator	120
serve as the indicators	45
serve as the indispensable	111
serve as the indispensible	40
serve as the individual	234
serve as the industrial	52
serve as the industry	607
serve as the info	42
serve as the informal	102
serve as the information	838
serve as the informational	41
serve as the infrastructure	500
serve as the initial	5331
serve as the initiating	125
serve as the initiation	63
serve as the initiator	81
serve as the injector	56
serve as the inlet	41
serve as the inner	87
serve as the input	1323

Maximum Likelihood Estimation

- Simple estimation: using relative frequency
- Example
  - comes across as 8 $P(as \mid comes\ across) = 0.8$
  - comes across more 1 $P(more \mid comes\ across) = 0.1$
  - comes across a 1 $P(a \mid comes\ across) = 0.1$
  - $P(X \mid comes\ across) = 0$

$$P_{MLE}\left(w_1 \cdots w_n\right) = \frac{C\left(w_1 \cdots w_n\right)}{N}$$

$$P_{MLE}\left(w_n \mid w_1 \cdots w_{n-1}\right) = \frac{C\left(w_1 \cdots w_n\right)}{C\left(w_1 \cdots w_{n-1}\right)}$$

In person	*she*		*was*		*inferior*		*to*		*both*		*sisters*	
1-gram	$P(\cdot)$		$P(\cdot)$		$P(\cdot)$		$P(\cdot)$		$P(\cdot)$		$P(\cdot)$	
1	the	0.034	the	0.034	the	0.034	the	0.034	the	0.034	the	0.034
2	to	0.032	to	0.032	to	0.032	**to**	**0.032**	to	0.032	to	0.032
3	and	0.030	and	0.030	and	0.030			and	0.030	and	0.030
4	of	0.029	of	0.029	of	0.029			of	0.029	of	0.029
...												
8	was	0.015	**was**	**0.015**	was	0.015			was	0.015	was	0.015
...												
13	**she**	**0.011**			she	0.011			she	0.011	she	0.011
...												
254					both	0.0005			**both**	**0.0005**	both	0.0005
...												
435					sisters	0.0003					**sisters**	**0.0003**
...												
1701					**inferior**	**0.00005**						
2-gram	$P(\cdot \mid person)$		$P(\cdot \mid she)$		$P(\cdot \mid was)$		$P(\cdot \mid inferior)$		$P(\cdot \mid to)$		$P(\cdot \mid both)$	
1	and	0.099	had	0.141	not	0.065	**to**	**0.212**	be	0.111	of	0.066
2	who	0.099	**was**	**0.122**	a	0.052			the	0.057	to	0.041
3	to	0.076			the	0.033			her	0.048	in	0.038
4	in	0.045			to	0.031			have	0.027	and	0.025
...												
23	**she**	**0.009**							Mrs	0.006	she	0.009
...												
41									what	0.004	**sisters**	**0.006**
...												
293									**both**	**0.0004**		
...												
∞					**inferior**	**0**						
3-gram	$P(\cdot \mid In,person)$		$P(\cdot \mid person,she)$		$P(\cdot \mid she,was)$		$P(\cdot \mid was,inf.)$		$P(\cdot \mid inferior,to)$		$P(\cdot \mid to,both)$	
1	UNSEEN		did	0.5	not	0.057	UNSEEN		the	0.286	to	0.222
2			**was**	**0.5**	very	0.038			Maria	0.143	Chapter	0.111
3					in	0.030			cherries	0.143	Hour	0.111
4					to	0.026			her	0.143	Twice	0.111
...												
∞					**inferior**	**0**			**both**	**0**	**sisters**	**0**
4-gram	$P(\cdot \mid u,I,p)$		$P(\cdot \mid I,p,s)$		$P(\cdot \mid p,s,w)$		$P(\cdot \mid s,w,i)$		$P(\cdot \mid w,i,t)$		$P(\cdot \mid i,t,b)$	
1	UNSEEN		UNSEEN		in	1.0	UNSEEN		UNSEEN		UNSEEN	
...												
∞					**inferior**	**0**						

Coping with Zero Frequency Problem

- Discounting
- Interpolation
- Backing-off

Laplace's Law

- Adding one

$$P_{LAP}\left(w_1 \cdots w_n\right) = \frac{C\left(w_1 \cdots w_n\right)+1}{N+V^n}$$

- Problem: overestimation of unseen n-grams
  - N = 44 million (4.4×10^7)
  - V = 0.4 million
    - n=2: 1.6×10^{11}

Lidstone's Law

- Add not one, but some smaller possible value

$$P_{LID}\left(w_1 \cdots w_n\right) = \frac{C\left(w_1 \cdots w_n\right) + \lambda}{N + V^n \lambda}$$

$$= \mu \frac{C\left(w_1 \cdots w_n\right)}{N} + \left(1 - \mu\right)\frac{1}{V^n} \qquad \left(\mu = \frac{N}{N + V^n \lambda}\right)$$

- Interpolation between the MLE estimate and a uniform prior
- $\lambda = ½$ is most widely used, which is called
  - Jeffreys-Perks Law
  - Expected Likelihood Estimation (ELE)

Good-Turing Estimation

$$C(w_1 \cdots w_n) = r > 0$$

$$P_{GT}(w_1 \cdots w_n) = \frac{r^*}{N} = \frac{1}{N} \cdot (r+1) \frac{N_{r+1}}{N_r}$$

Bigrams

r	N_r	r	N_r
1	138741	28	90
2	25413	29	120
3	10531	30	86
4	5997	31	98
5	3565	32	99
6	2486		...
7	1754	1264	1
8	1342	1366	1
9	1106	1917	1
10	896	2233	1
	...	2507	1

r=2

$$r^* = 3 \cdot \frac{N_3}{N_2} = 3 \cdot \frac{10531}{25413} \approx 1.228$$

N_r: the number of n-grams that occur r times

$N = 617091$

$V = 14585$

Good-Turing Estimation

$$C(w_1 \cdots w_n) = r > 0$$

$$P_{GT}(w_1 \cdots w_n) = \frac{r^*}{N} = \frac{1}{N} \cdot (r+1) \frac{N_{r+1}}{N_r}$$

$$C(w_1 \cdots w_n) = 0$$

$$\sum_{r \geq 1} N_r \frac{r^*}{N} = ?$$

$$P_{GT}(w_1 \cdots w_n) = \frac{1 - \sum_{r \geq 1} N_r \frac{r^*}{N}}{N_0} = \frac{N_1}{N_0 N}$$

Good-Turing Estimation

- r=2

$$r^* = 3 \cdot \frac{N_3}{N_2} = 3 \cdot \frac{10531}{25413} \approx 1.228$$

- r=0

$$P = \frac{N_1}{N_0 N} = \frac{1}{N_0} \cdot \frac{N_1}{N} = \frac{1}{14585^2 - 199252} \cdot \frac{138741}{617091}$$

$$= 1.058 \times 10^{-9}$$

$$\because N_0 = V^2 - \sum_{r \geq 1} N_r$$

Bigrams

r	N_r	r	N_r
1	138741	28	90
2	25413	29	120
3	10531	30	86
4	5997	31	98
5	3565	32	99
6	2486		...
7	1754	1264	1
8	1342	1366	1
9	1106	1917	1
10	896	2233	1
	...	2507	1

$N = 617091$

$V = 14585$

r	r^*	$P_{\mathrm{GT}}(\cdot)$
0	0.0007	1.058×10^{-9}
1	0.3663	5.982×10^{-7}
2	1.228	2.004×10^{-6}
3	2.122	3.465×10^{-6}
4	3.058	4.993×10^{-6}
5	4.015	6.555×10^{-6}
6	4.984	8.138×10^{-6}
7	5.96	9.733×10^{-6}
8	6.942	1.134×10^{-5}
9	7.928	1.294×10^{-5}
10	8.916	1.456×10^{-5}
...		
28	26.84	4.383×10^{-5}
29	27.84	4.546×10^{-5}
30	28.84	4.709×10^{-5}
31	29.84	4.872×10^{-5}
32	30.84	5.035×10^{-5}
...		
1264	1263	0.002062
1366	1365	0.002228
1917	1916	0.003128
2233	2232	0.003644
2507	2506	0.004092

[Manning & Schütze 1999]

Combining Estimators (1)

- Linear interpolation
  - also called a mixture model or deleted interpolation
- In case of trigram:

$$P_{li}\left(w_n \mid w_{n-2}, w_{n-1}\right) = \lambda_1 P_1\left(w_n\right) + \lambda_2 P_2\left(w_n \mid w_{n-1}\right) + \lambda_3 P_3\left(w_n \mid w_{n-2}, w_{n-1}\right)$$

$$0 \le \lambda_i \le 1 \quad \sum_i \lambda_i = 1$$

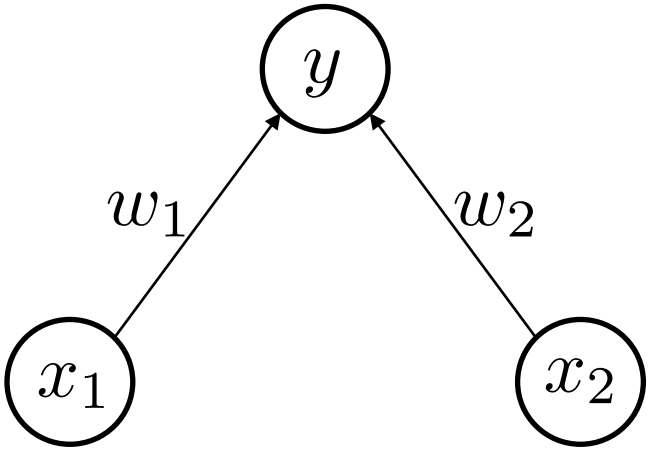
Combining Estimators (2)

- Katz's backing-off

$$P_{katz}\left(w_i \mid w_{i-n+1}\cdots w_{i-1}\right)=\begin{cases} P^*\left(w_i \mid w_{i-n+1}\cdots w_{i-1}\right) & \text{if } C\left(w_{i-n+1}\cdots w_i\right)>k \\ \alpha_{w_{i-n+1}\cdots w_{i-1}} P_{katz}\left(w_i \mid w_{i-n+2}\cdots w_{i-1}\right) & \text{otherwise} \end{cases}$$

Recurrent Neural Networks

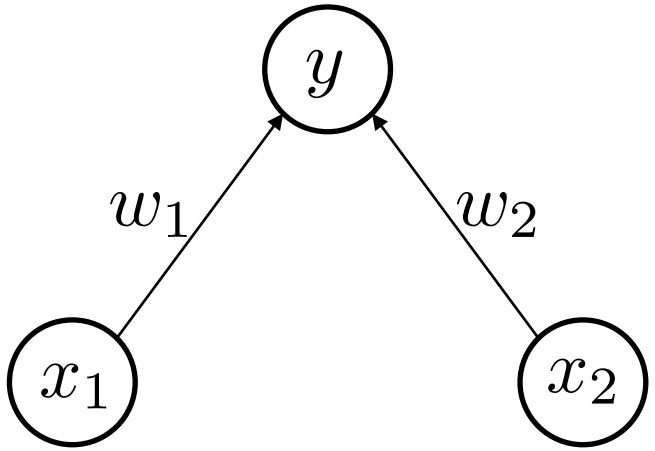
Most Basic Neural Network



$$y = \begin{cases} 0 & (w_1 x_1 + w_2 x_2 \leq \theta) \\ 1 & (w_1 x_1 + w_2 x_2 > \theta) \end{cases}$$

AND Gate

x_1	x_2	y
0	0	0
1	0	0
0	1	0
1	1	1

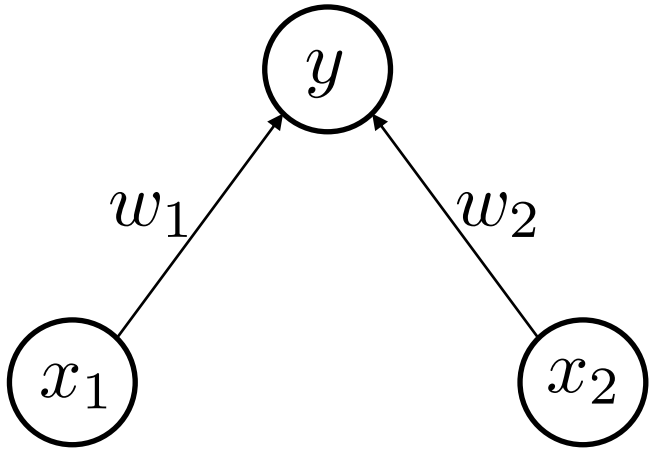


$$y = \begin{cases} 0 & (w_1 x_1 + w_2 x_2 \leq \theta) \\ 1 & (w_1 x_1 + w_2 x_2 > \theta) \end{cases}$$

$$y = \begin{cases} 0 & (0.5 x_1 + 0.5 x_2 \leq 0.7) \\ 1 & (0.5 x_1 + 0.5 x_2 > 0.7) \end{cases}$$

NAND Gate

x_1	x_2	y
0	0	1
1	0	1
0	1	1
1	1	0

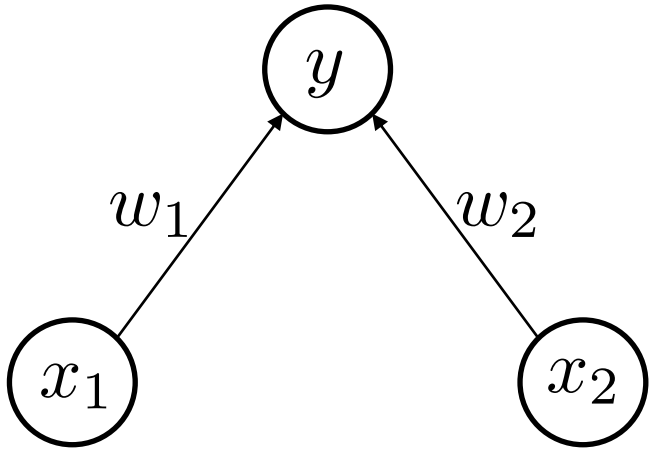


$$y = \begin{cases} 0 & (w_1 x_1 + w_2 x_2 \leq \theta) \\ 1 & (w_1 x_1 + w_2 x_2 > \theta) \end{cases}$$

$$y = \begin{cases} 0 & (-0.5 x_1 - 0.5 x_2 \leq -0.7) \\ 1 & (-0.5 x_1 - 0.5 x_2 > -0.7) \end{cases}$$

OR Gate

x_1	x_2	y
0	0	0
1	0	1
0	1	1
1	1	1

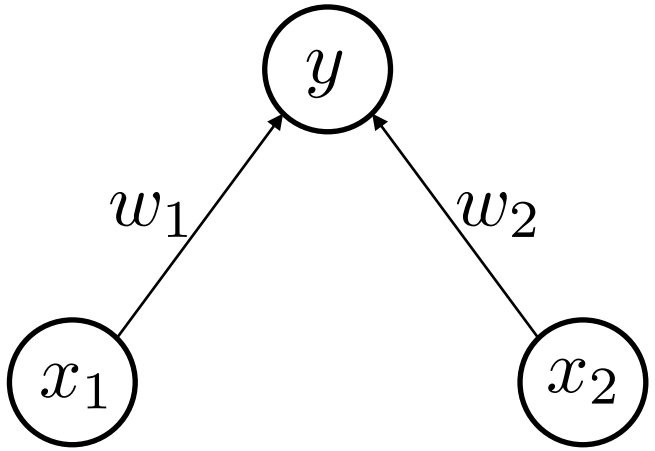


$$y = \begin{cases} 0 & (w_1 x_1 + w_2 x_2 \leq \theta) \\ 1 & (w_1 x_1 + w_2 x_2 > \theta) \end{cases}$$

$$y = \begin{cases} 0 & (0.7 x_1 + 0.7 x_2 \leq 0.5) \\ 1 & (0.7 x_1 + 0.7 x_2 > 0.5) \end{cases}$$

XOR Gate?

x_1	x_2	y
0	0	0
1	0	1
0	1	1
1	1	0

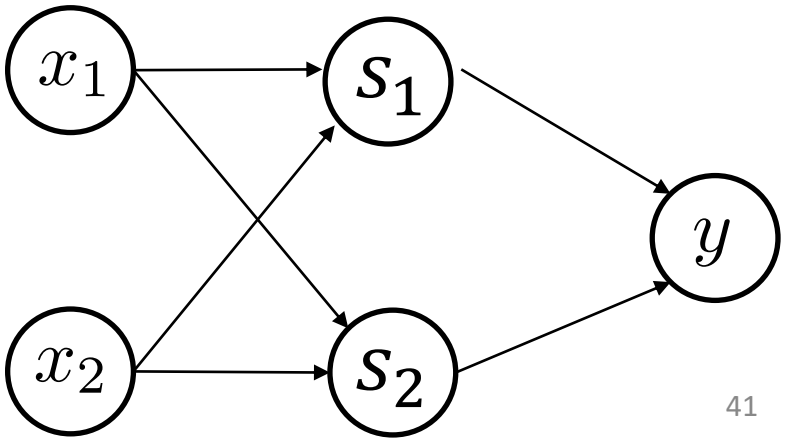


$$y = \begin{cases} 0 & (w_1 x_1 + w_2 x_2 \leq \theta) \\ 1 & (w_1 x_1 + w_2 x_2 > \theta) \end{cases}$$

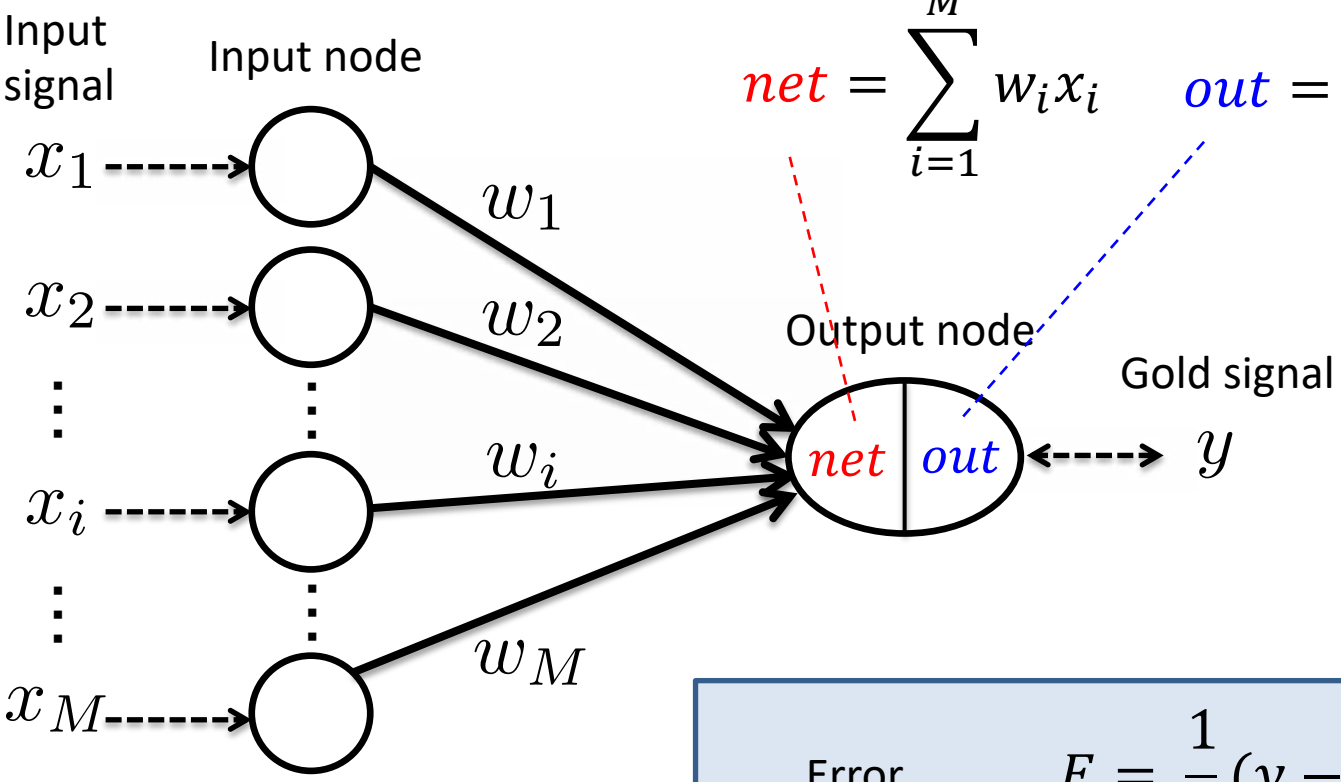
?

XOR Gate?

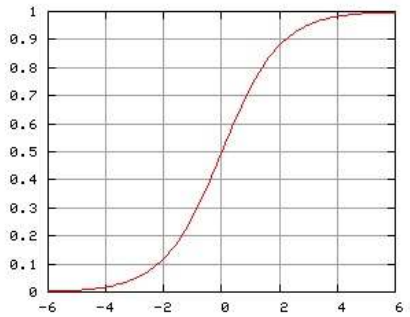
x_1	x_2	NAND	OR	NAND ∧ OR	y
0	0	1	0	0	0
1	0	1	1	1	1
0	1	1	1	1	1
1	1	0	1	0	0



Neural Network



$$net = \sum_{i=1}^{M} w_i x_i \qquad out = \frac{1}{1 + \exp(-net)}$$

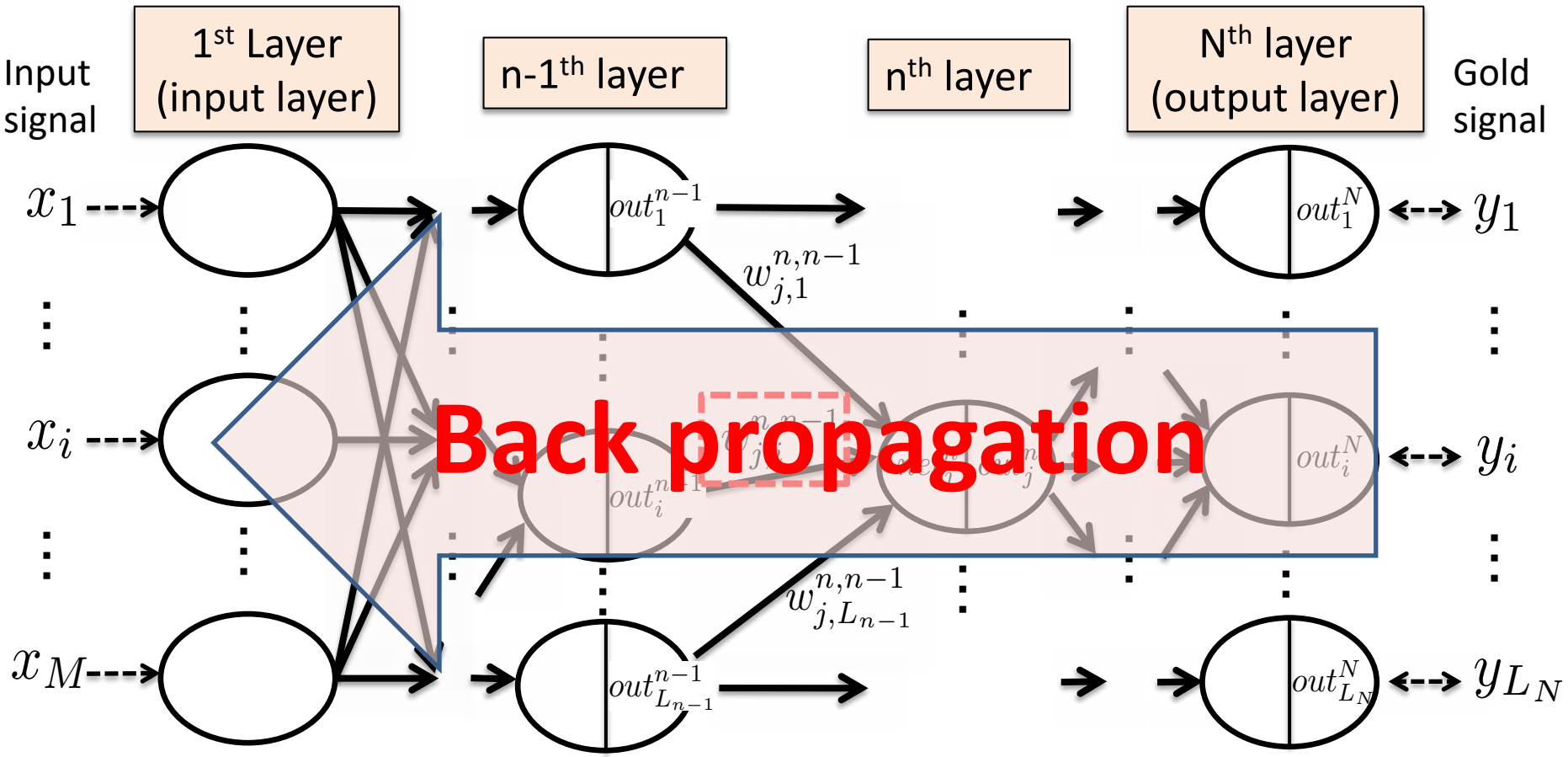


Sigmoid function

Error $$E = \frac{1}{2}(y - out)^2$$

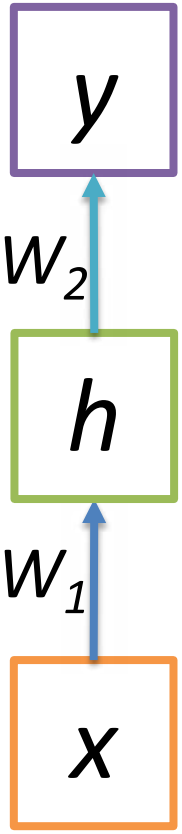
Weight update $$w_i^{(new)} = w_i^{(old)} - \eta \frac{\partial E}{\partial w_i}$$

Feedforward Neural Network

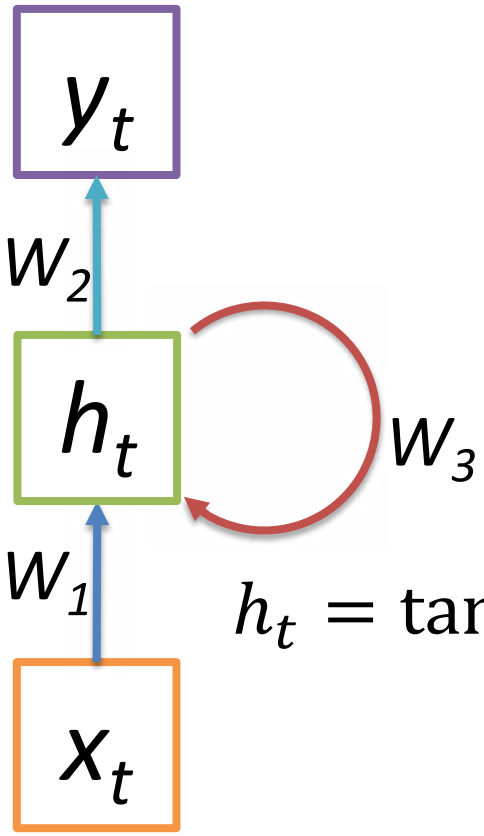


$$E = \frac{1}{2}\sum_{i=1}^{L_N}(y_i - out_i^N)^2 \qquad w_{j,i}^{n,n-1(new)} = w_{j,i}^{n,n-1(old)} - \eta \frac{\partial E}{\partial w_{j,i}^{n,n-1}}$$

Feedforward Neural Network

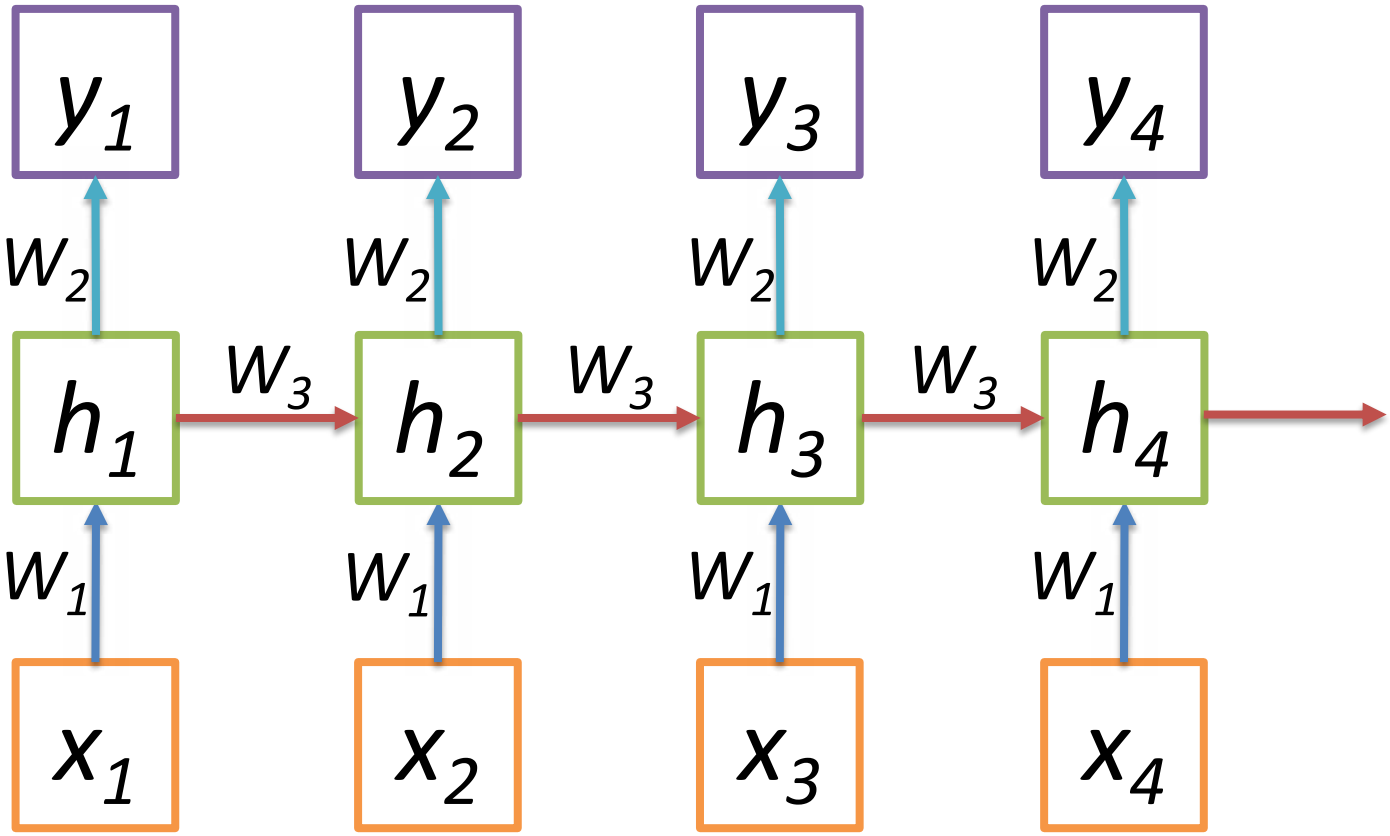


Recurrent Neural Network (RNN)

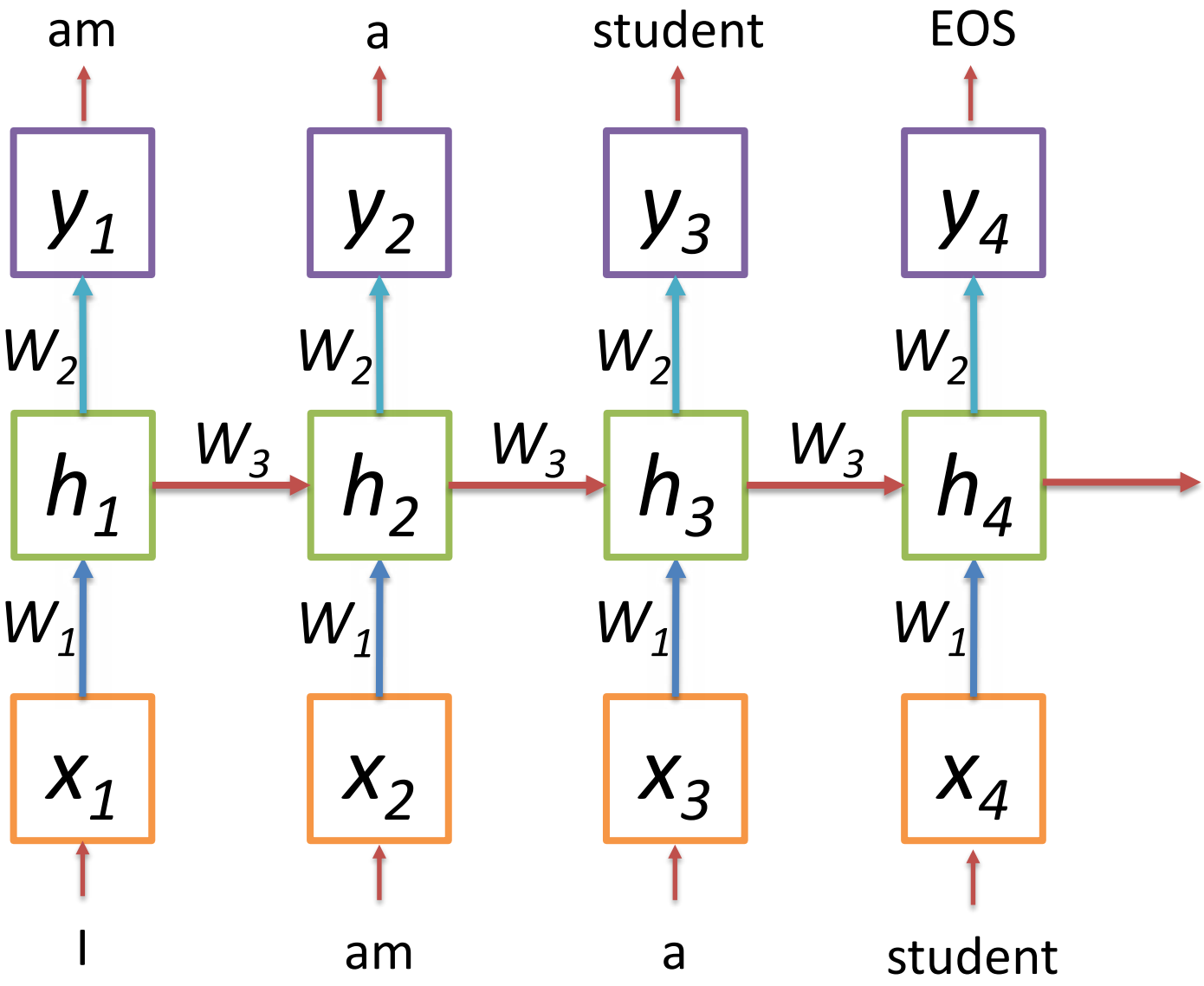


$$h_t = \tanh(W_3 h_{t-1} + W_1 x_t)$$

Recurrent Neural Network (RNN)



RNN Language Model



Limitations of (Ordinary) RNNs

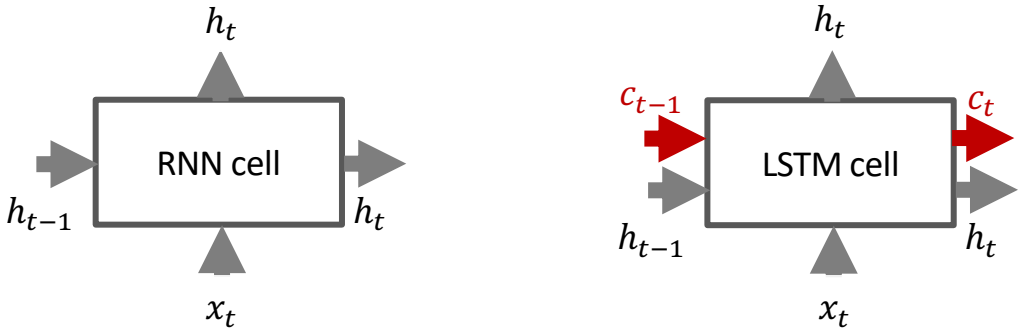
- Difficult to capture long-distance dependencies
  - Because the hidden vector is updated by $h_t = \tanh(W_3 h_{t-1} + W_1 x_t)$ every time, old information tends to disappear
- Training is not stable
  - Vanishing/exploding gradient problems

    During back propagation, the gradient will be vanishing to zero or exploding to infinite

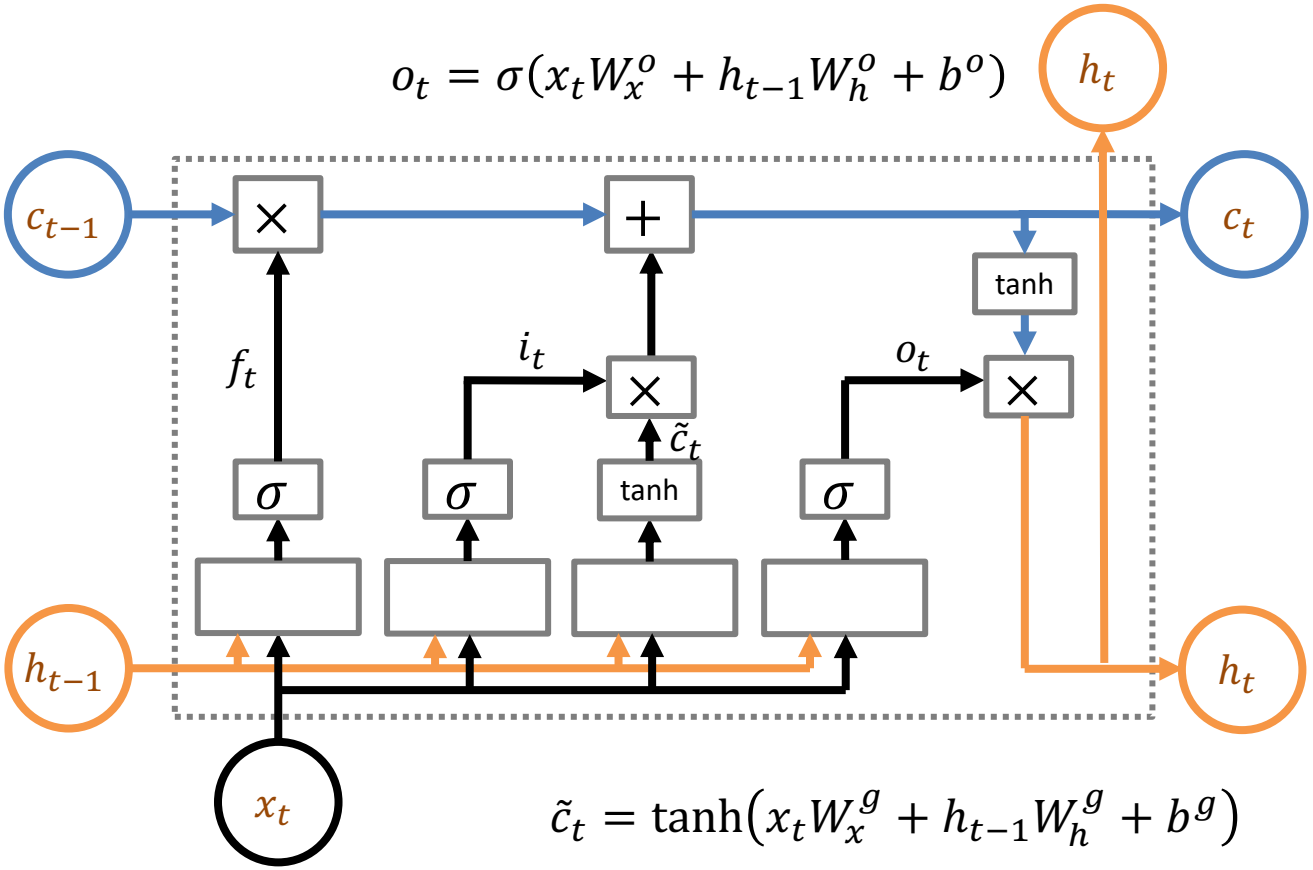
LSTM (Long Short-Term Memory)

[Hochreiter+ 1997]

- To keep information for a long time, a memory cell c_t is added
- At time t, do forgetting, reading, and writing of the memory cell
  - Each of them is controlled by the gate that is dynamically determined by h_{t-1} and x_t



LSTM (Long Short-Term Memory)



Summary

- Collocations
  - Frequency
  - Mean and variance
  - Hypothesis testing
  - Likelihood ratio
  - Pointwise mutual information
- N-gram language models
  - Discounting
    - Laplace's law
    - Lidstone's law
    - Good-Turing estimation
  - Backing-off
- Recurrent neural networks