



# Deep Learning

## Sequence Models

Dr. Mehran Safayani  
safayani@iut.ac.ir

safayani.iut.ac.ir



<https://www.aparat.com/mehran.safayani>



[https://github.com/safayani/deep\\_learning\\_course](https://github.com/safayani/deep_learning_course)



# Examples of sequence data

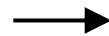
Speech recognition



“The quick brown fox jumped  
over the lazy dog.”

Music generation

∅



Sentiment classification

“There is nothing to like  
in this movie.”



DNA sequence analysis

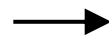
AGCCCCTGTGAGGAACTAG



AG**CCCCTGTGAGGAACT**AG

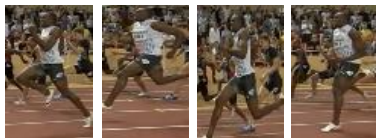
Machine translation

Voulez-vous chanter avec  
moi?



Do you want to sing with  
me?

Video activity recognition



Running

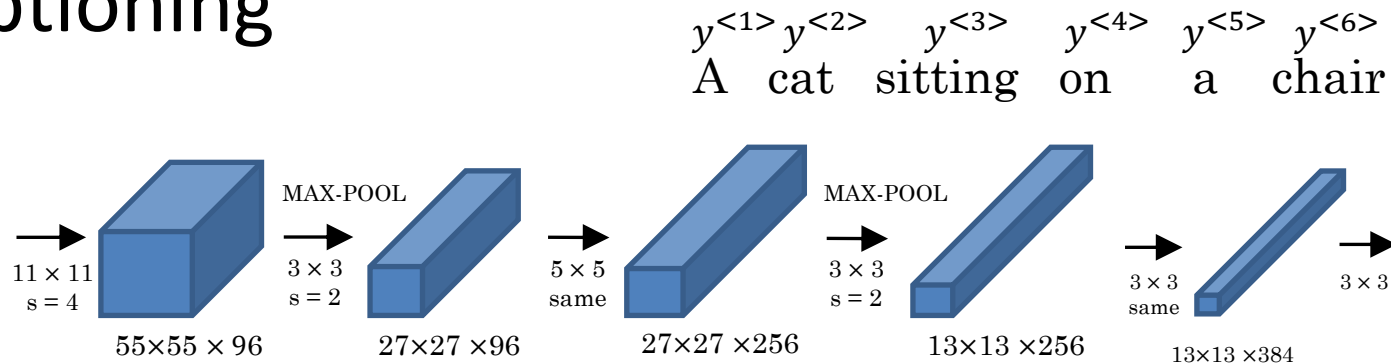
Name entity recognition

Yesterday, Harry Potter  
met Hermione Granger.

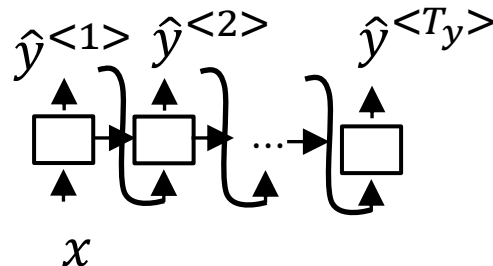
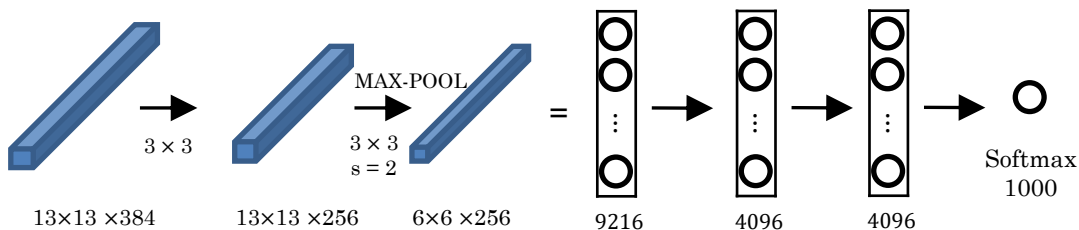


Yesterday, **Harry Potter**  
met **Hermione Granger**.

# Image captioning



$y^{<1>}$   $y^{<2>}$   $y^{<3>}$   $y^{<4>}$   $y^{<5>}$   $y^{<6>}$   
A cat sitting on a chair



[Mao et. al., 2014. Deep captioning with multimodal recurrent neural networks]

[Vinyals et. al., 2014. Show and tell: Neural image caption generator]

[Karpathy and Li, 2015. Deep visual-semantic alignments for generating image descriptions]

# Recurrent Neural Networks

- Now, given this input X let's say that you want a model to operate Y that has one outputs per input word and the target output the design Y tells you for each of the input words is that part of a person's name.

- X: (Ali Ahmadi ) and (Hassan Hamidi ) invented a new drug.

$$X^{(1)} \quad X^{(2)} \quad X^{(3)} \quad \dots \quad \dots \quad X^{(t)} \quad \dots \quad \dots \quad X^{(9)} \quad T_x=9$$

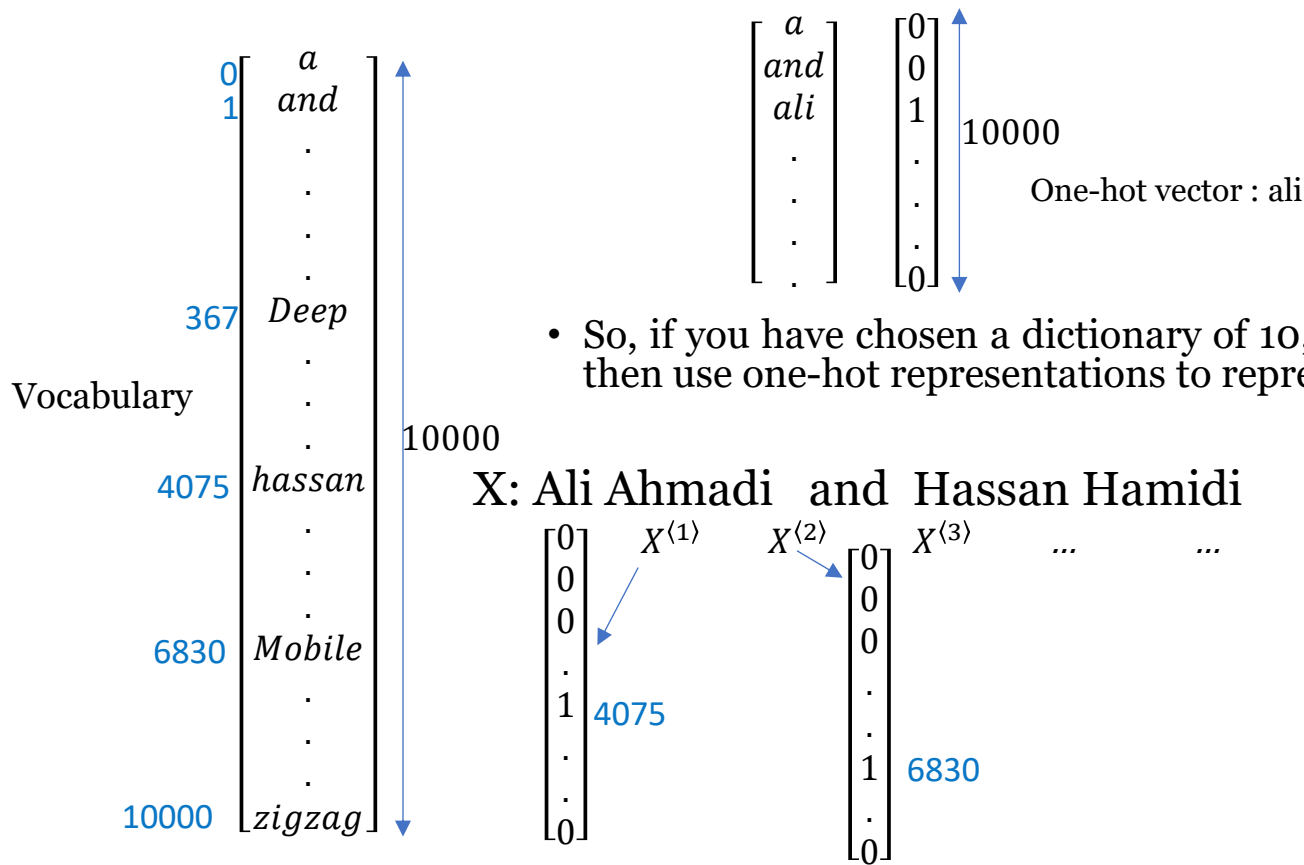
- y:  $\begin{matrix} 1 & 1 & 0 & 1 & & 1 & & 0 & 0 & 0 & 0 \\ y^{(1)} & y^{(2)} & y^{(3)} & & & & & & & & y^{(9)} \end{matrix} \quad T_y=9$

$$\begin{aligned} & \bullet X^{(i)(t)} \\ & \quad \downarrow \\ & \text{داده آموزشی } i \text{ ام} \end{aligned} \quad \begin{aligned} & T_x^{(i)} = 9 \\ & T_y^{(i)} \end{aligned}$$

- This is our first serious foray into NLP or Natural Language Processing.

# Representing words

- Dictionary: 30000, 50000

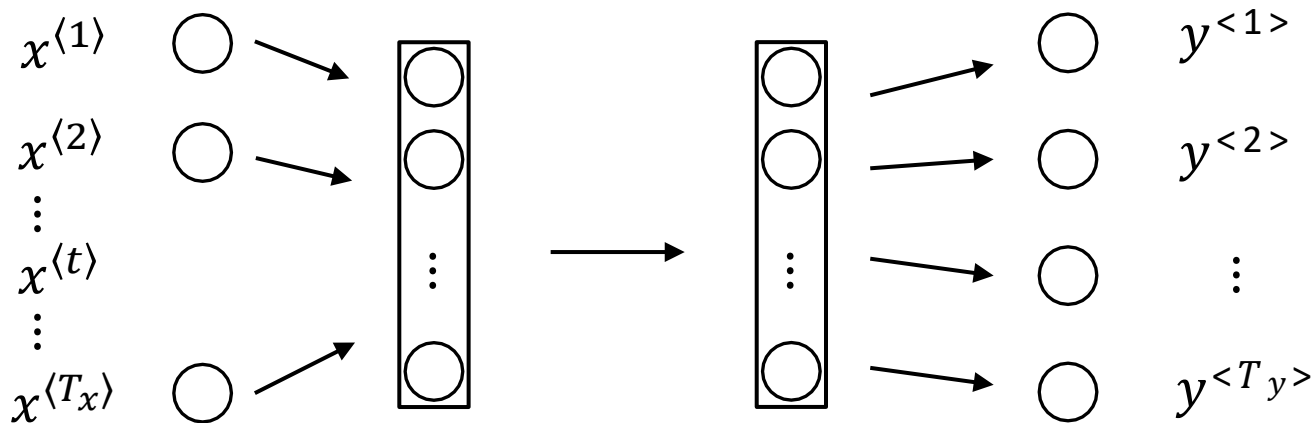


# Representing words

- Utilizing a sequence model for supervised learning to map input  $X$  to output  $Y$ .
- Introduction of an "Unknown Word" token for handling out-of-vocabulary words.
- Describing a notation for training sets in sequence data.

# Recurrent Neural Network Model

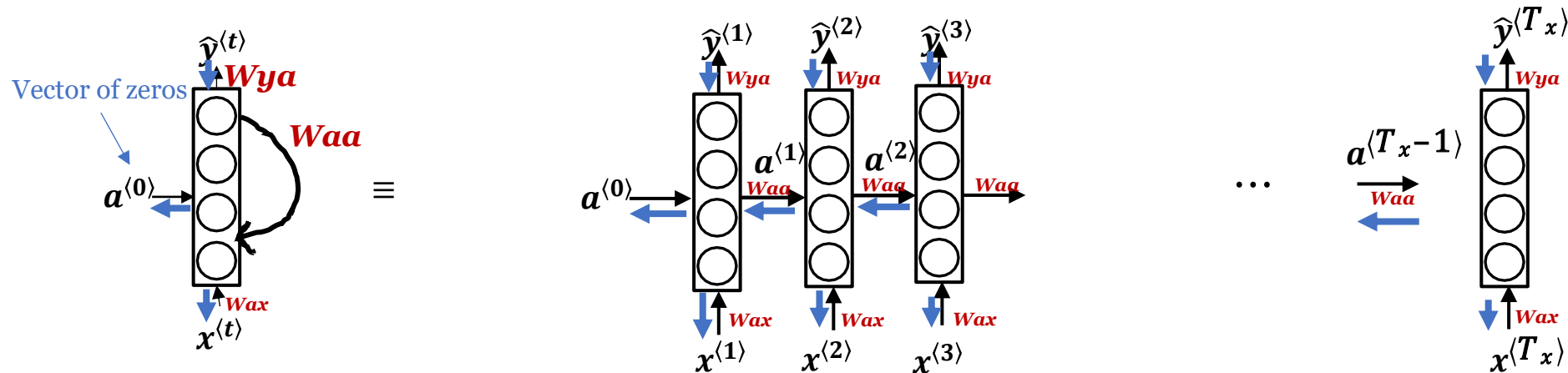
Why not a standard network?



Problems:

- Inputs, outputs can be different lengths in different examples.
- Doesn't share features learned across different positions of text.

# Recurrent Neural Networks



$$L^{(t)}(\hat{y}^{(t)}, y^{(t)}) = -y^{(t)} \log \hat{y}^{(t)} - (1 - y^{(t)}) \log(1 - \hat{y}^{(t)})$$

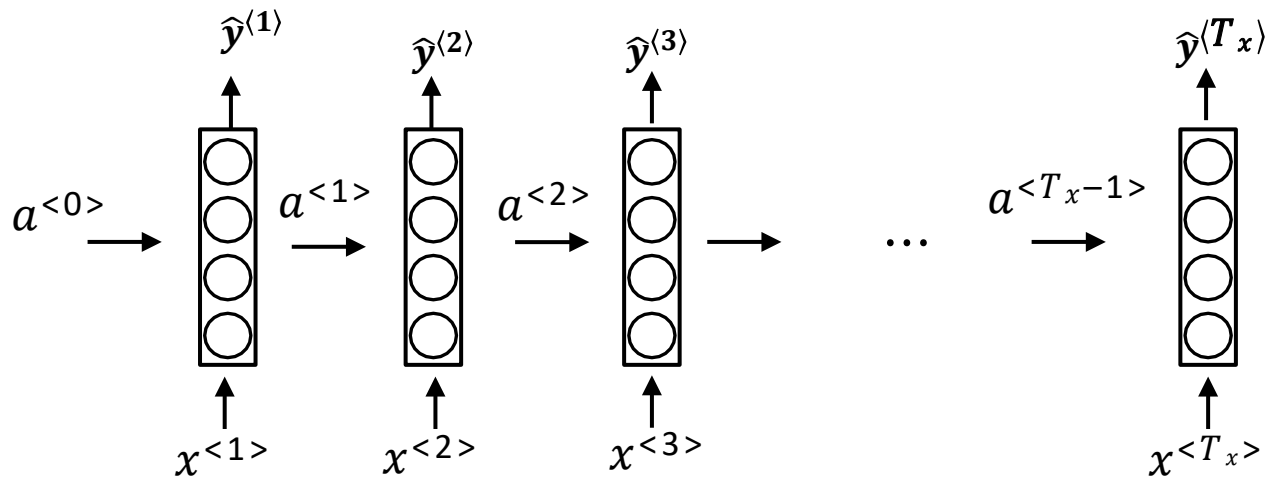
$$L(\hat{y}, y) = \sum_{t=1}^{T_x} L^{(t)}(\hat{y}^{(t)}, y^{(t)})$$

- Activation at time zero in neural networks is often initialized as a vector of zeros.
- Some researchers prefer initializing it as  $a^{(0)}$  randomly.
- There are alternative methods to initialize the activation at time zero

Backward propagation through time



# Forward Propagation



$$a^{<0>} = \vec{0}$$

$$a^{<1>} = g_1(w_{aa}a^{<0>} + w_{ax}x^{<1>} + b_a) \leftarrow \text{tanh|Relu}$$

$$\hat{y}^{<1>} = g_2(w_{ya}a^{<1>} + b_y) \leftarrow \text{sigmoid}$$

$$a^{<t>} = g(w_{aa}a^{<t-1>} + w_{ax}x^{<t>} + b_a)$$

$$\hat{y}^{<t>} = g(w_{ya}a^{<t>} + b_y)$$

# Simplified RNN notation

$$a^{(t)} = g(w_{aa}a^{(t-1)} + w_{ax}x^{(t)} + b_a)$$

Diagram annotations for the first equation:  
 -  $w_{aa}$ : dimension (100,100)  
 -  $w_{ax}$ : dimension (100,10000)  
 -  $a^{(t-1)}$ : dimension 100  
 -  $x^{(t)}$ : dimension 10000  
 -  $b_a$ : dimension 100

$$a^{(t)} = g(w_a[a^{(t-1)}, x^{(t)}] + b_a)$$

$$\hat{y}^{(t)} = g(w_{ya}a^{(t)} + b_y)$$

$$\bullet a^{(t)} = g(w_a[a^{(t-1)'}, x^{(t)'}]' + b_a)$$

$$\begin{bmatrix} a^{(t-1)} \\ x^{(t)} \end{bmatrix}$$

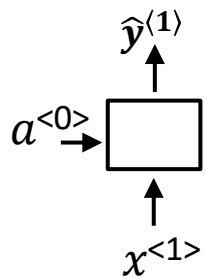
Diagram annotations for the vector equation:  
 -  $a^{(t-1)}$ : dimension 100  
 -  $x^{(t)}$ : dimension 10000  
 - Combined vector dimension: 10100

$$w_a = [w_{aa} : w_{ax}]$$

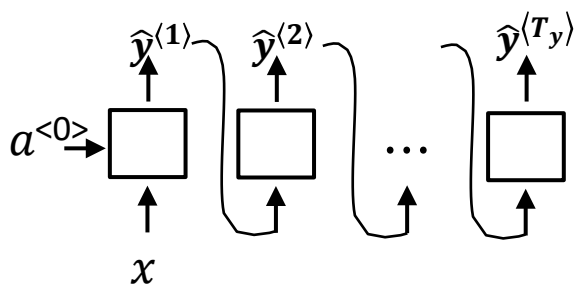
Diagram annotations for the weight matrix  $w_a$ :  
 -  $w_{aa}$ : dimension 100  
 -  $w_{ax}$ : dimension 10000  
 - Combined weight matrix dimension: (100,10100)

$$[w_{aa} : w_{ax}] \begin{bmatrix} a^{(t-1)} \\ x^{(t)} \end{bmatrix} = w_{aa}a^{(t-1)} + w_{ax}x^{(t)}$$

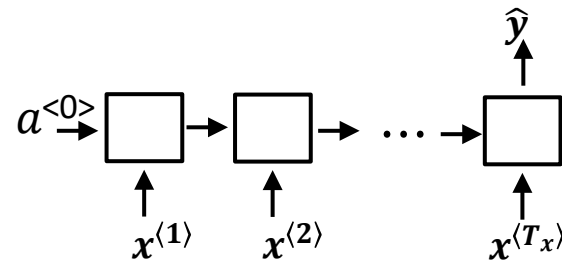
# Summary of RNN types



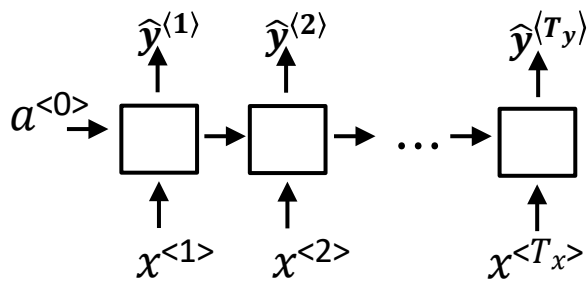
One to one



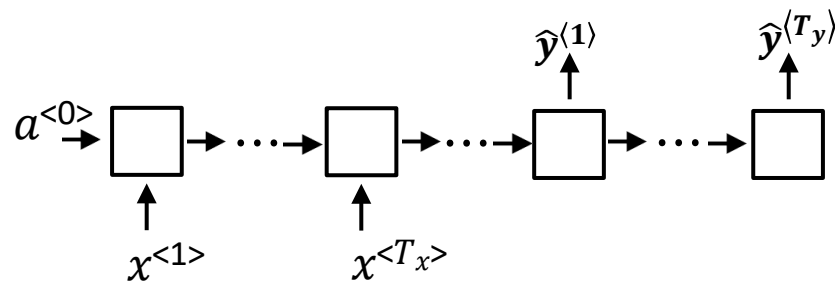
One to many



Many to one



Many to many



Many to many

# What is language modelling?

## Speech recognition

- The apple and pair salad.
- The apple and pear salad.
- $P(\text{The apple and pair salad}) = 3.2 \times 10^{-13}$
- $P(\text{The apple and pear salad}) = 5.7 \times 10^{-10}$
- $P(\text{sentence}) = P(y^{(1)}, y^{(2)}, \dots, y^{(T_y)}) = \text{احتمال وقوع جمله}$

# Language modelling with an RNN

Training set: large corpus of english text.

*Tokenize*

Cats average 15 hours of sleep a day!  $\downarrow$   $\langle \text{EOS} \rangle$   
 $y^{(1)}$        $y^{(2)}$        $y^{(3)}$       ...       $y^{(9)}$

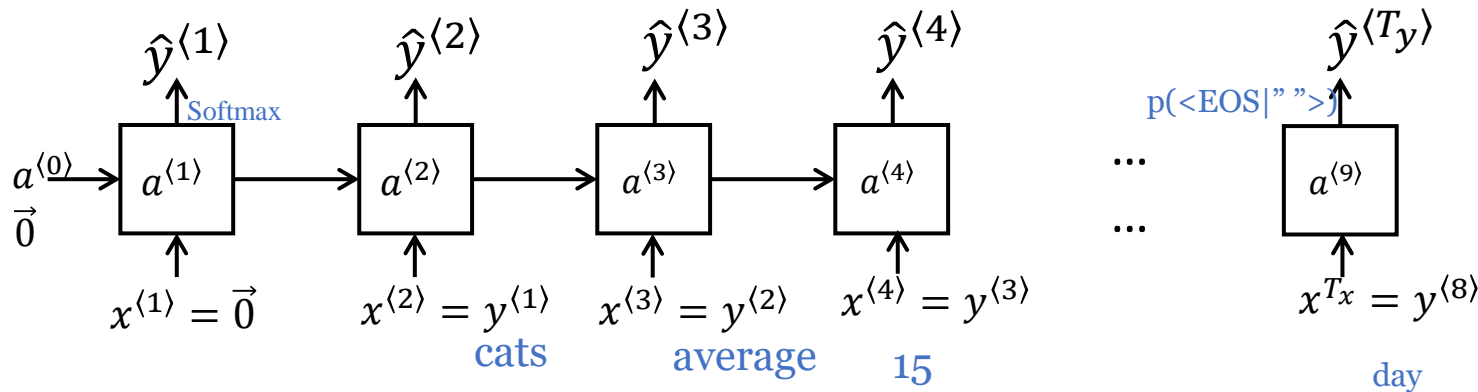
$$x^{(t)} = y^{(t-1)}$$

The Egyptian ~~Mau~~ is a breed of cat.  $\langle \text{EOS} \rangle$   
 $\langle \text{UNK} \rangle$

# RNN model

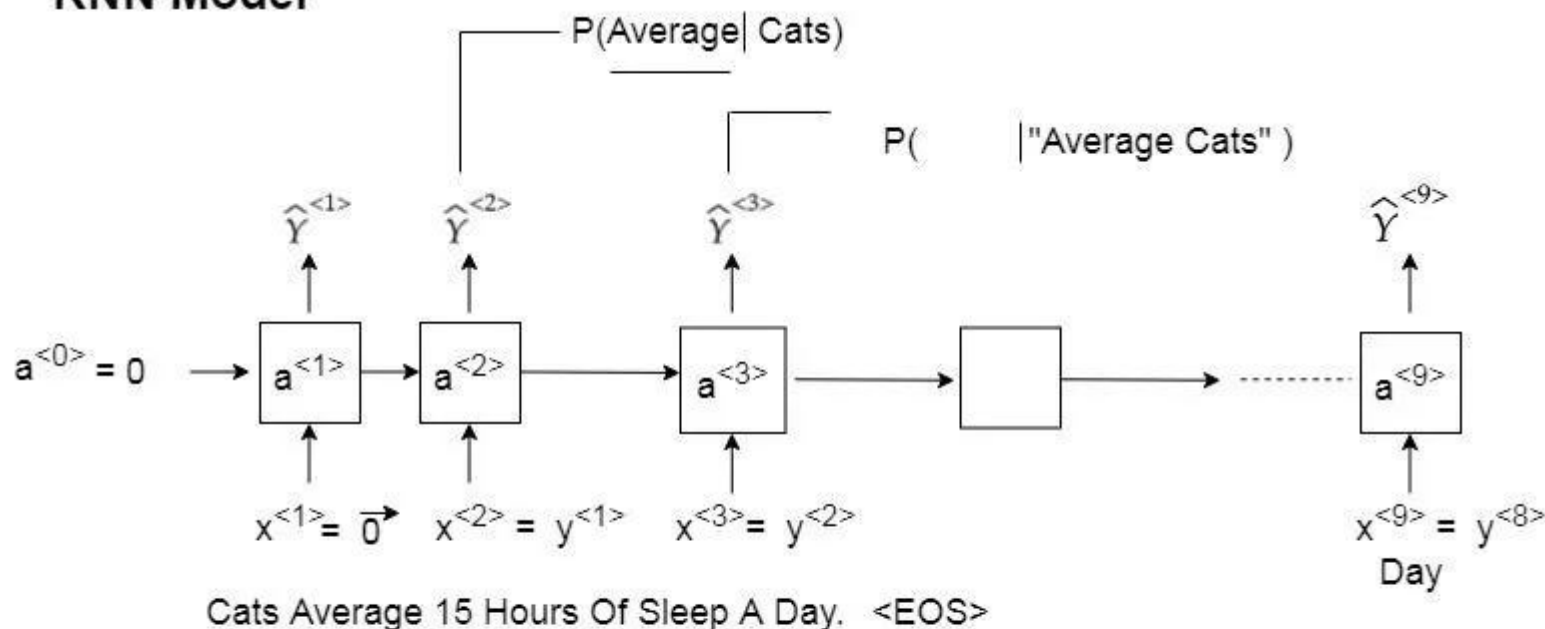
Cats average 15 hours of sleep a day.

- $p(a)$                        $p(\text{cat})$                        $p(\text{bread})$                       ...                       $p(\text{Eat})$
- $p(a|\text{cats})$                        $p(\text{cat}|\text{cats})$                        $p(\text{bread}|\text{cats})$                        $p(\text{average}|\text{cats})$                       ....  $p(\text{Eat}|\text{cats})$
- $P(a|\text{cats average})$                        $p(\text{cats}|\text{cats average})$  ...  $p(15|\text{cats average})$  ....  $P(\text{eat}|\text{cats average})$



- $L(\hat{y}(t), y(t)) = -\sum_i y_i^{(t)} \log \hat{y}_i^{(t)}$
- $L = \sum_t L^{(t)}(\hat{y}^{(t)}, y^{(t)})$

## RNN Model

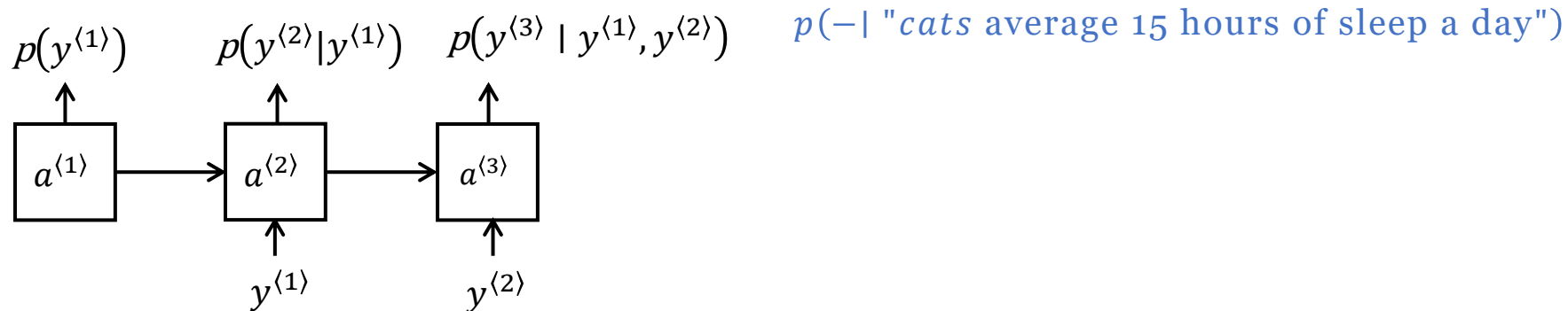


Cost Function  $L(\hat{Y}^{<t>}, Y^{<t>}) = -\sum_i Y_i^{<t>} \log \hat{Y}_i^{<t>}$  ← SoftMax Loss Function

$L = \sum_t L^{<t>}(\hat{Y}^{<t>}, Y^{<t>})$  ← Overall Loss Function

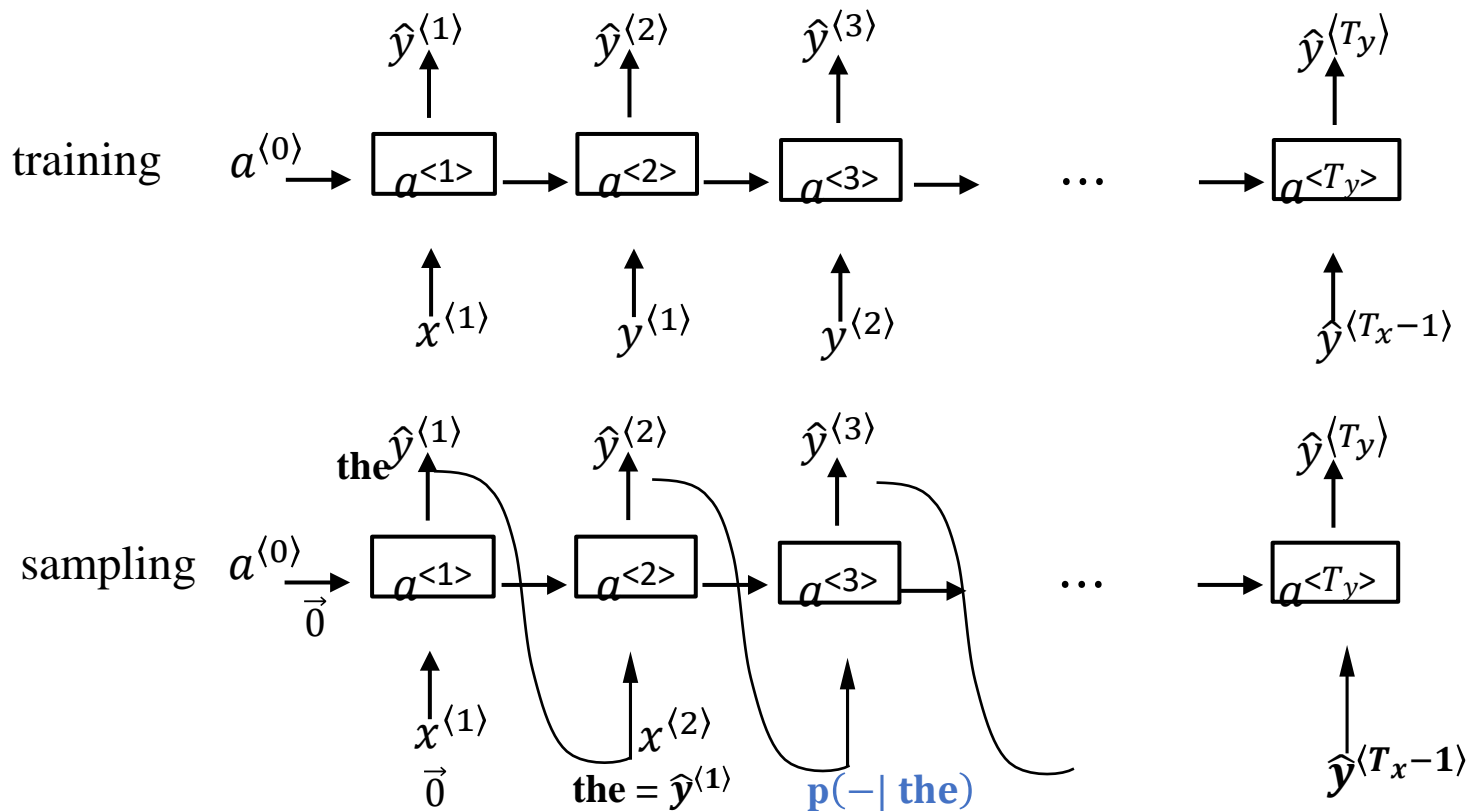
# RNN model

- Cats average 15 hours of sleep a day. <EOS>
- $p(y^{(1)}, y^{(2)}, y^{(3)}) = p(y^{(1)}) \cdot p(y^{(2)} | y^{(1)}) \cdot p(y^{(3)} | y^{(1)}, y^{(2)})$





# Sampling a sequence from a trained RNN

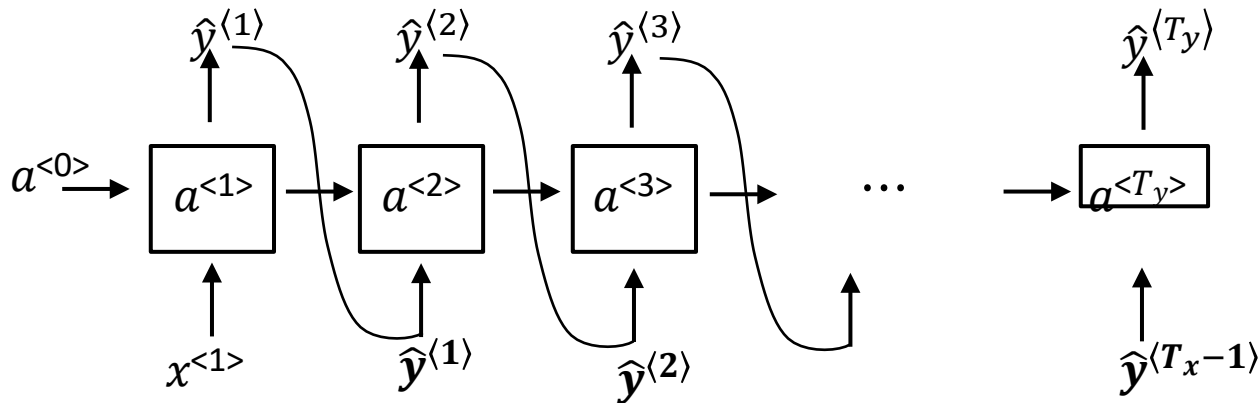


<EOS>

<UNK>

# Character-level language model

- Vocabulary = [a, aaron, ..., zulu, <UNK>]
- Vocabulary = [ a, b, c, ... , z ,  $\sqcup$  , . , ..., 0, ... , 9, A, ... ,Z]
- Cat average ... Mau  
↑↑↑



# Sequence generation

## News

President Enrique Peña Nieto, announced  
Sánchez's sulk former coming football Langston  
Parings.

"I was not at all surprised," said Hich Langston.

"Concussion epidemic", to be examined.

The gray football the told some and this has on  
the UEFA icon, should money as.

## Shakespeare

The mortal moon hath her eclipse in love.

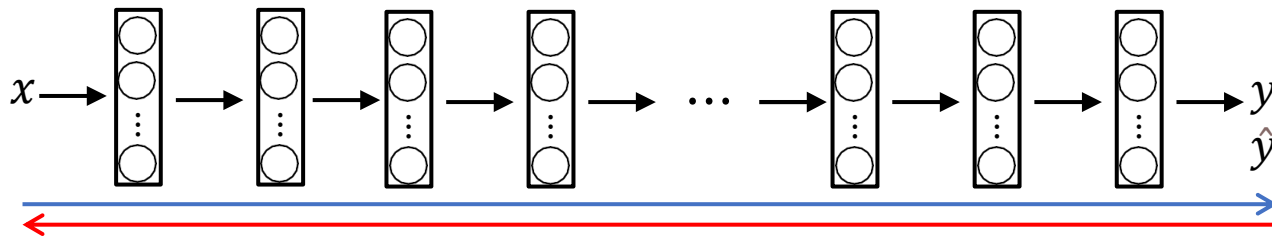
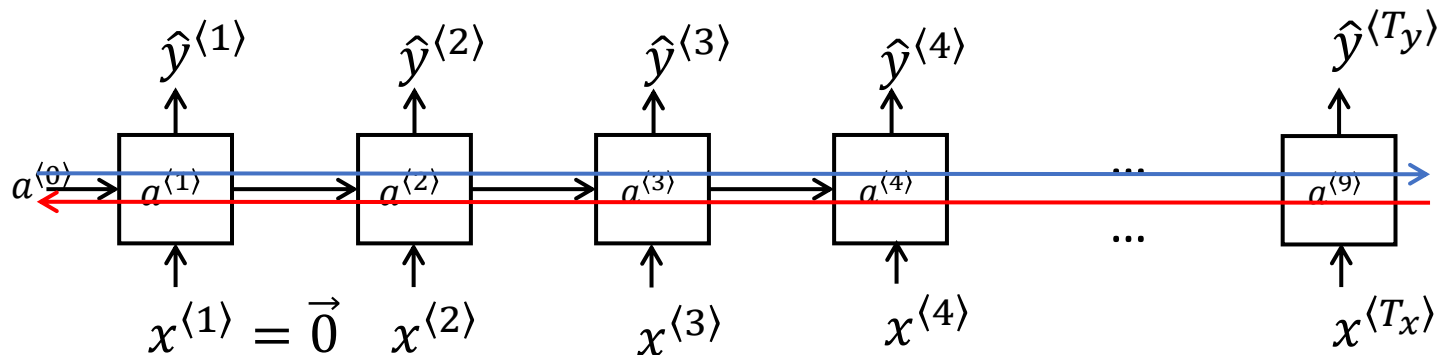
And subject of this thou art another this fold.

When better be my love to me see Sable's.

For whose are ruse of mine eyes heaves.

# Vanishing gradients with RNNs

- The cat which already ate bunch of food was full
- The cats ... were full



Exploding gradients.

NaN

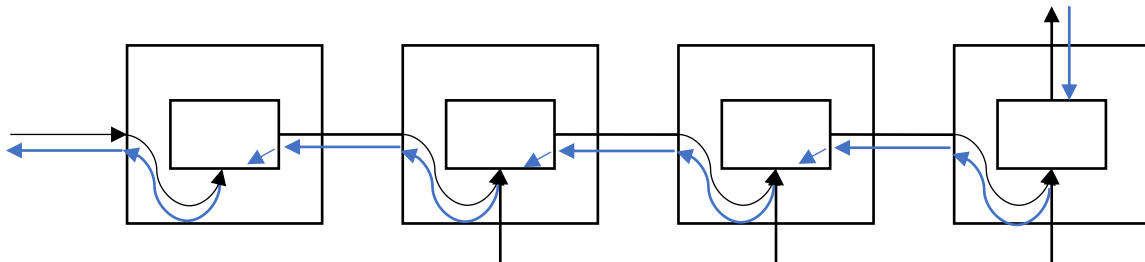
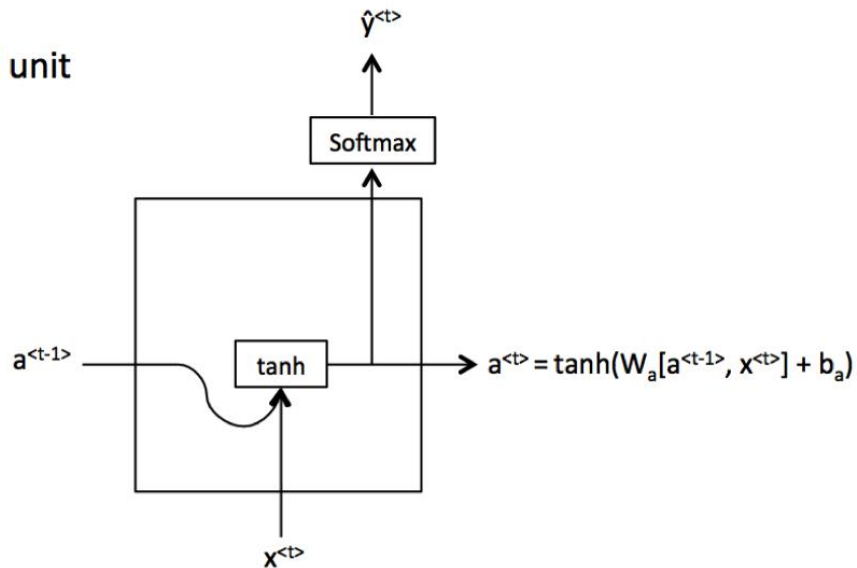
gradient clipping

# RNN Unit

- $a^{(t)} = g(W_a[a^{(t-1)}, x^{(t)}] + b_a)$

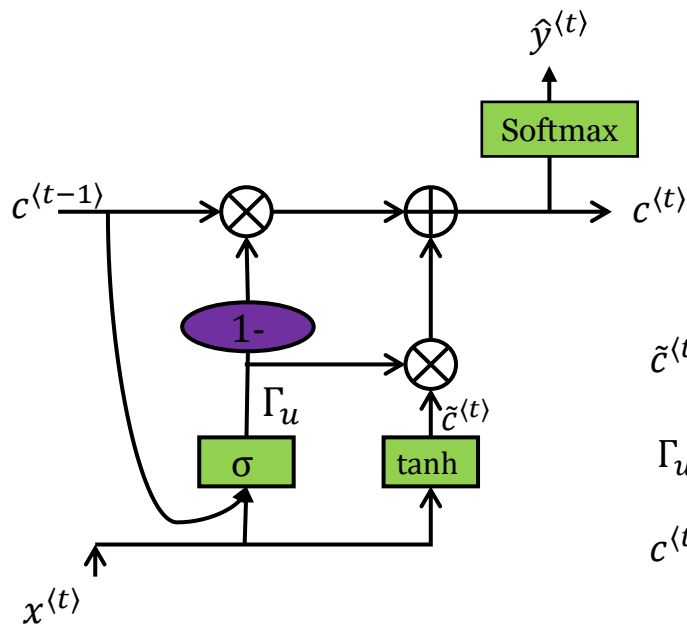
↑  
tanh

RNN unit



# GRU (simplified)

$\Gamma_u = 0$     $\Gamma_u = 1$     $\Gamma_u = 0$     $\Gamma_u = 0$     $\Gamma_u = 0$     $\dots$     $\Gamma_u = 0$   
 $c^{(t)} = 0$     $c^{(t)} = 1$     $c^{(t+1)} = 1$     $c^{(t+2)} = 1$     $c^{(t+3)} = 1$     $\dots$     $c^{(t+n)} = 1$   
 The   **cat**,   which   already   ate   ...,   **was**   full.



C : memory cell

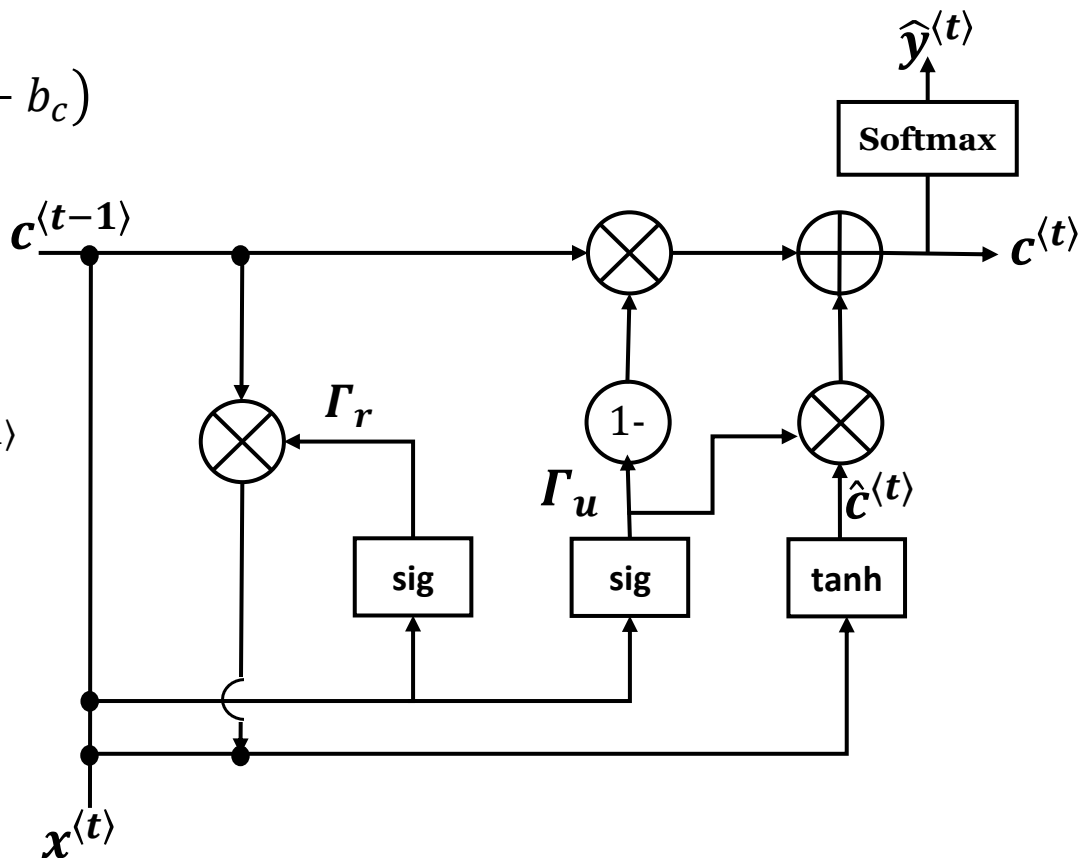
$$\tilde{c}^{(t)} = \tanh(w_c[c^{(t-1)}, x^{(t)}] + b_c)$$

$$\Gamma_u = \sigma(w_u[c^{(t-1)}, x^{(t)}] + b_u)$$

$$c^{(t)} = \Gamma_u * \tilde{c}^{(t)} + (1 - \Gamma_u) * c^{(t-1)}$$

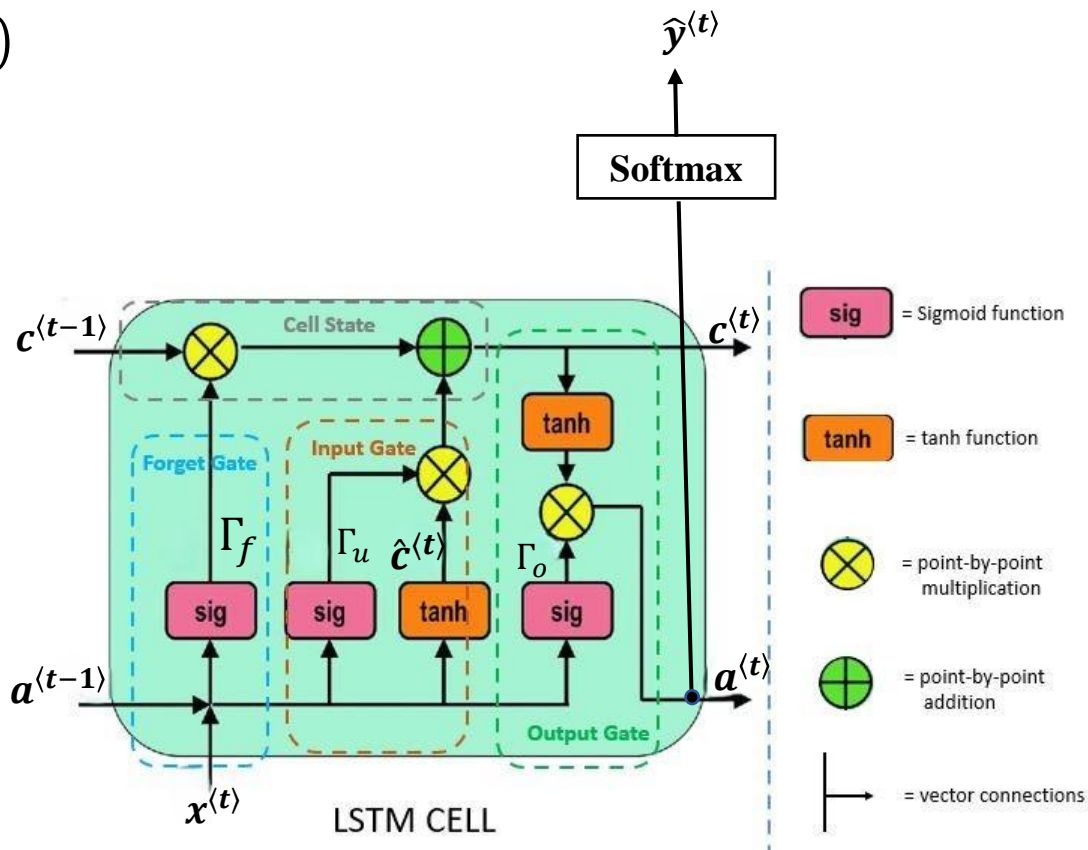
# Full GRU

- $\hat{c}^{(t)} = \tanh(w_c[\Gamma_r * c^{(t-1)}, x^{(t)}] + b_c)$
- $\Gamma_u = \sigma(w_u[c^{(t-1)}, x^{(t)}] + b_u)$
- $\Gamma_r = \sigma(w_r[c^{(t-1)}, x^{(t)}] + b_r)$
- $c^{(t)} = \Gamma_u * \hat{c}^{(t)} + (1 - \Gamma_u) * c^{(t-1)}$



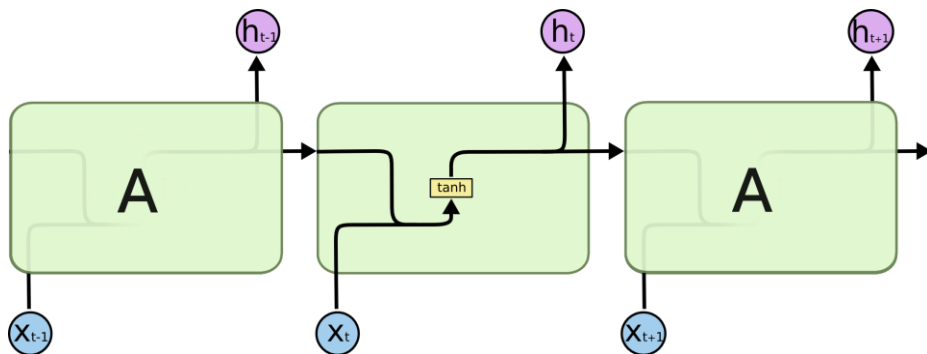
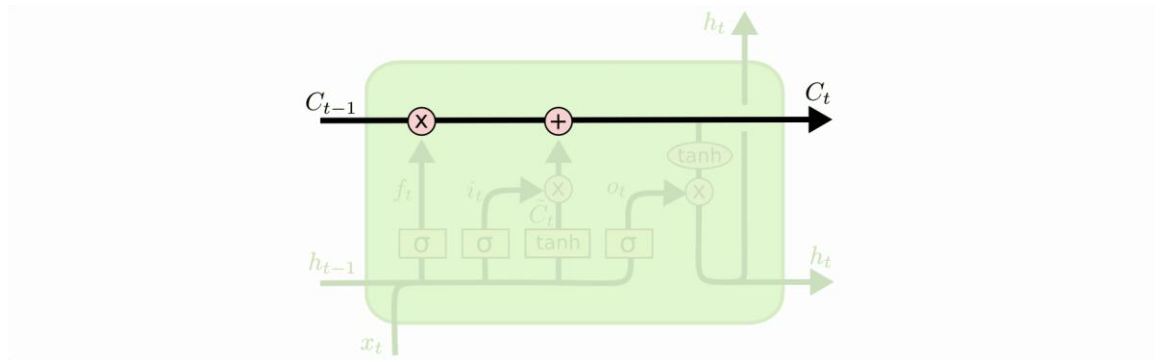
# LSTM

- $\hat{c}^{(t)} = \tanh(w_c[a^{(t-1)}, x^{(t)}] + b_c)$
- $\Gamma_u = \sigma(w_u[a^{(t-1)}, x^{(t)}] + b_u)$
- $\Gamma_f = \sigma(w_f[a^{(t-1)}, x^{(t)}] + b_f)$
- $\Gamma_o = \sigma(w_o[a^{(t-1)}, x^{(t)}] + b_o)$
- $c^{(t)} = \Gamma_u * \hat{c}^{(t)} + \Gamma_f * c^{(t-1)}$
- $a^{(t)} = \Gamma_o * \tanh(c^{(t)})$



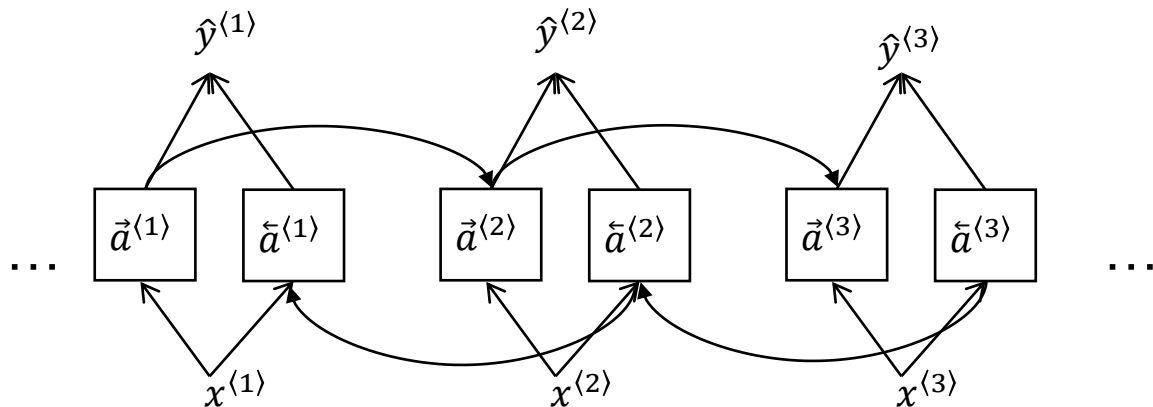


# Vanishing gradient



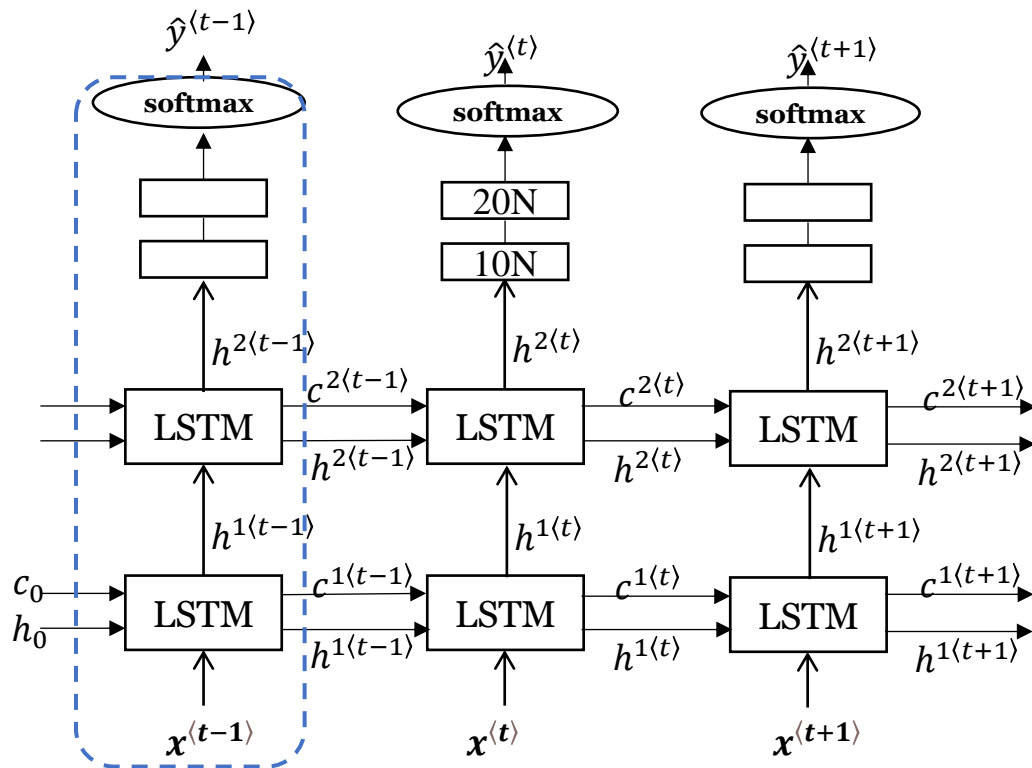
# Bidirectional RNN (BRNN)

- He said that cannon is a great compony
- He said that cannon is a very effective weapon



- $\hat{y}^{(t)} = g(w_y[\vec{a}^{(t)}, \tilde{a}^{(t)}] + by)$

# Deep RNN (Stacked RNN)



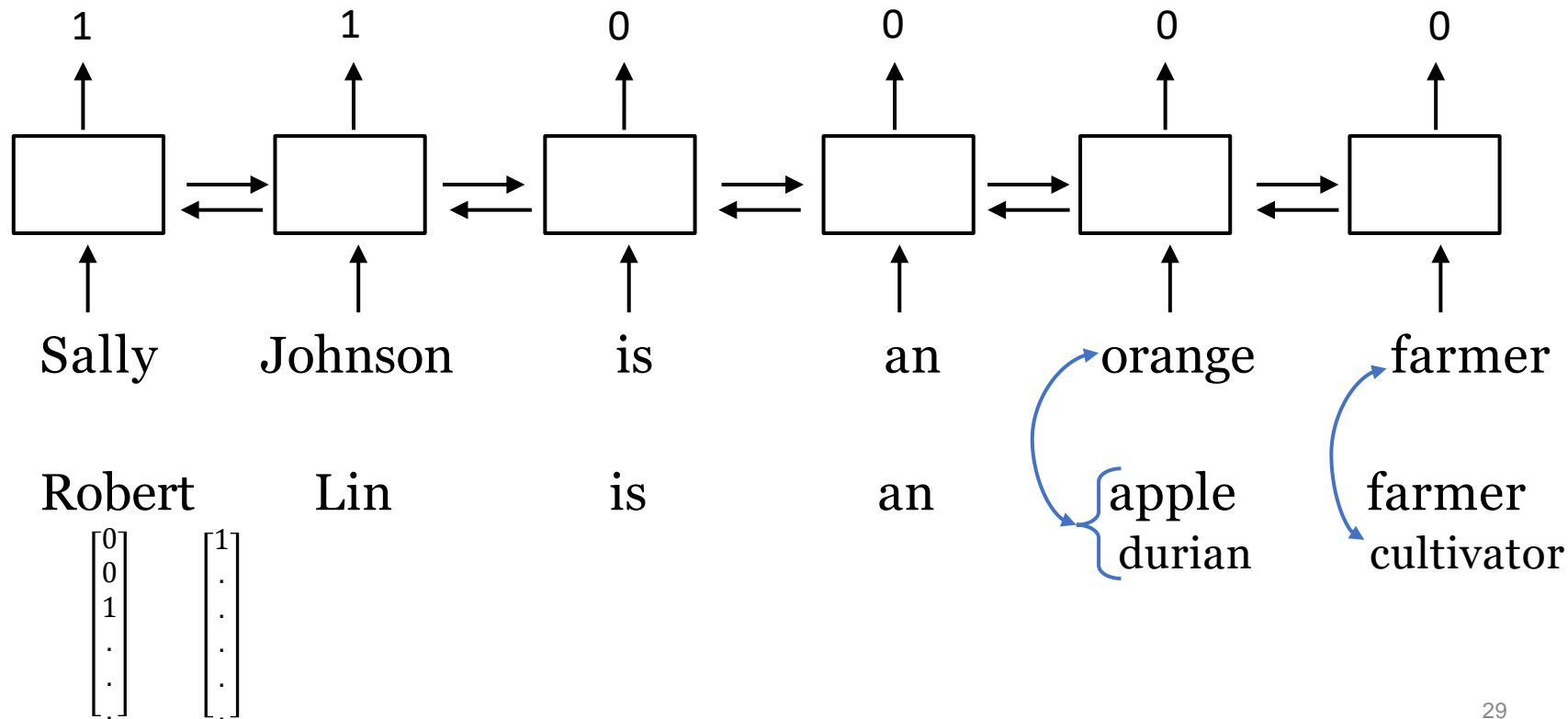
# RNN-Example

```
import torch, torch.nn as nn
rnn = nn.LSTM(input_dim=10, hidden_dim=20, num_layers=2)
input = torch.randn(seq_len=5, batch_size=3, input_dim=10)
h0 = torch.randn(num_layer=2, batch_size=3, hidden_dim=20) c0 =
torch.randn(num_layer=2, batch_size=3, hidden_dim=20)
Output, (hn, cn) = rnn(input, (h0, c0))
```

```
#output=(5,3,20), h_n=(2,3,20), c_n=(2,3,20)
```

```
lin = nn.Linear(hidden_dim, output_dim)
v1 = nn.View(seq_len*batch, hidden_size)
v2 = nn.View(seq_len, batch, output_size)
output = v2(lin(v1(out_rnn)))
```

# Named entity recognition example



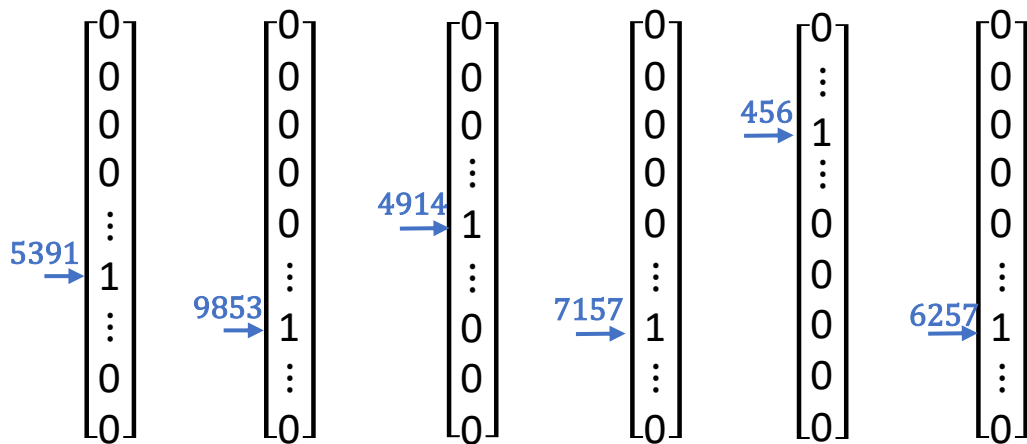
# Word representation

$V = [a, aaron, \dots, zulu, <UNK>]$

$|V| = 10000$

1-hot representation

Man	Woman	King	Queen	Apple	Orange
(5391)	(9853)	(4914)	(7157)	(456)	(6257)



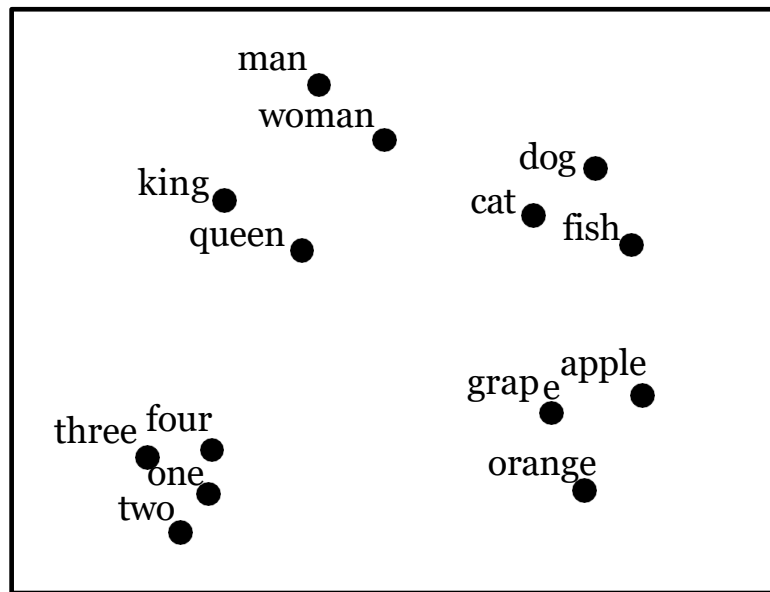
I want a glass of orange juice .  
I want a glass of apple\_\_\_\_\_.

# Feature representation: word embedding

	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	-0.02
Food	0.04	0.01	0.02	0.01	0.95	0.97
.						
.						
.						

# Visualizing word embeddings

T-SNE:

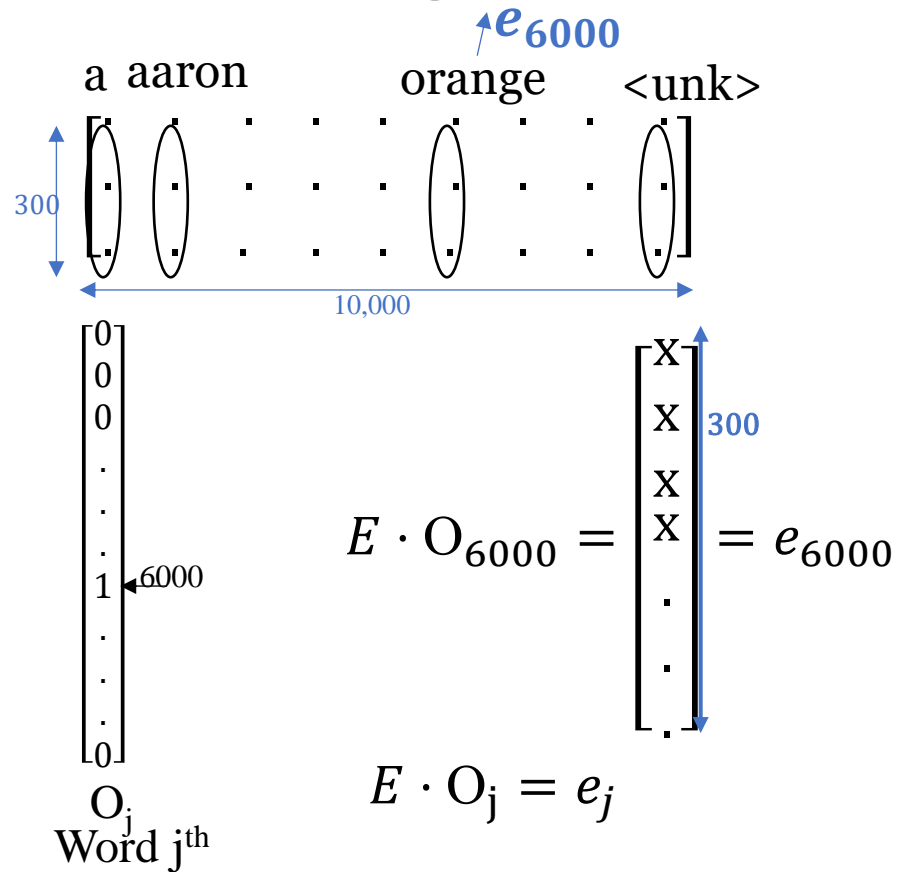


Man  $\rightarrow$  woman as King  $\rightarrow$  ?

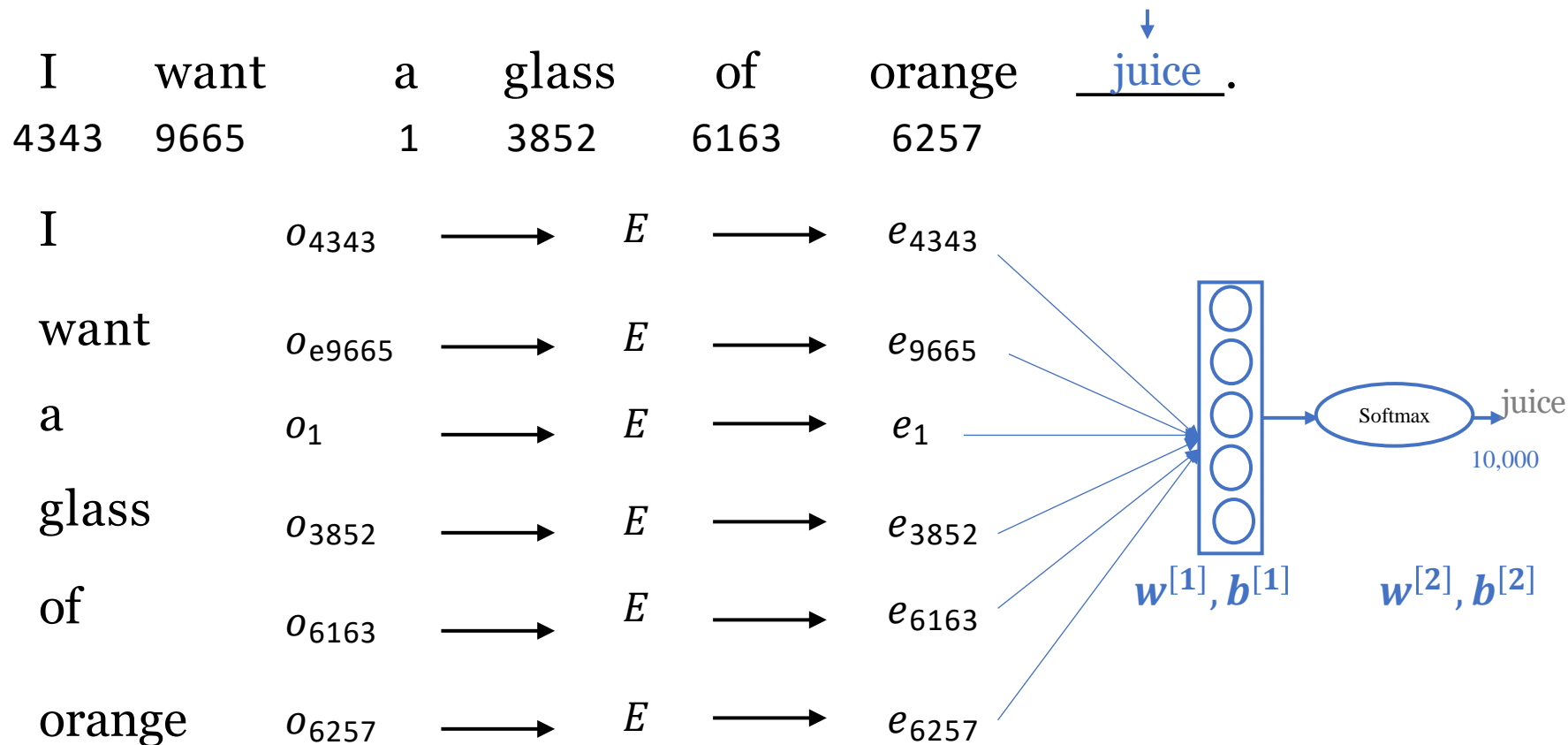
$$e_{man} - e_{woman} \approx e_{king} - e_w?$$
$$\arg \max \text{sim}(e_w, e_{king} - e_{man} + e_{woman})$$
$$\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$$



# Embedding matrix



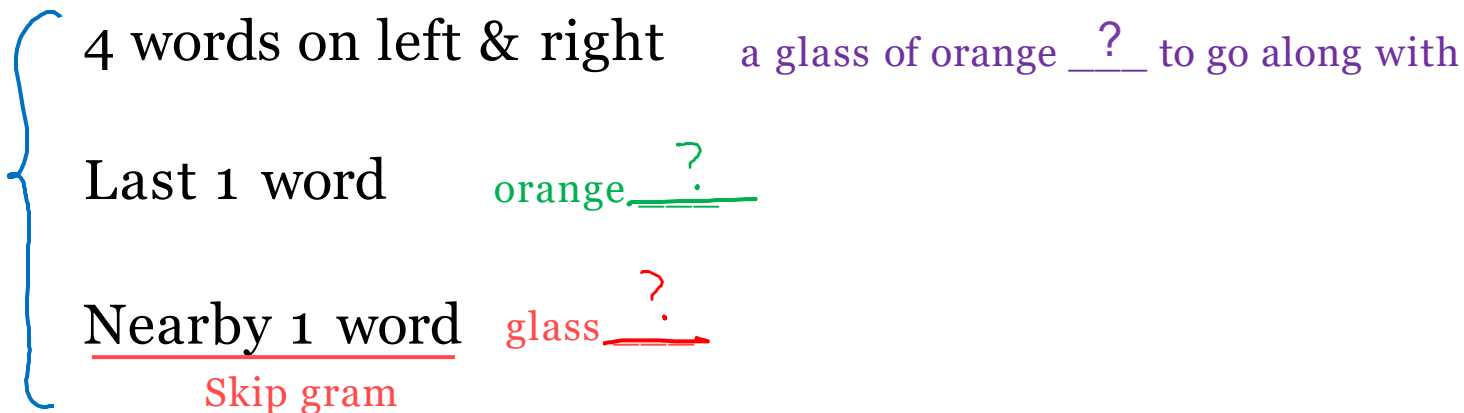
# Neural language model



# Other context/target pairs

I want a glass of orange juice to go along with my cereal.

Context: Last 4 words.



# Skip-grams

I want a glass of orange juice to go along with my cereal.

<u>Context</u>	<u>target</u>
orange	juice
orange	glass
orange	my
.	

$$O_c \rightarrow E \rightarrow e_c \rightarrow \text{Softmax} \rightarrow \hat{y}$$

**Softmax:**

$$p(t \mid c) = \frac{e^{\theta_t^\top e_c}}{\sum_{j=1}^{10,000} e^{\theta_j^\top e_c}}$$

$$L(\hat{y}, y) = -\sum_{i=1}^{10000} y_i \log \hat{y}_i$$

$$y_i = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 1 \\ \vdots \\ \vdots \\ 36 \\ 0 \end{bmatrix} \leftarrow 4500$$

# Sentiment classification problem

$x$

The dessert is excellent.



Service was quite slow.



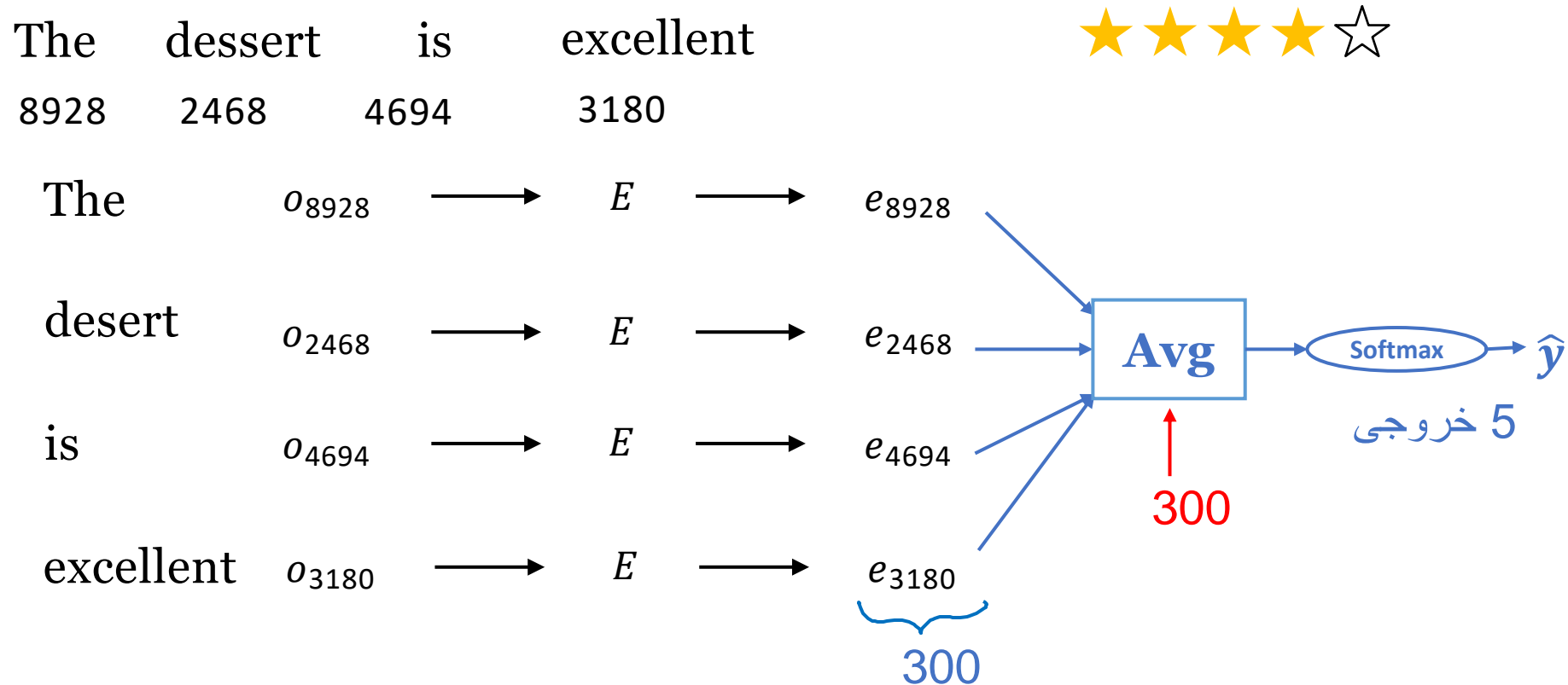
Good for a quick meal, but nothing special.



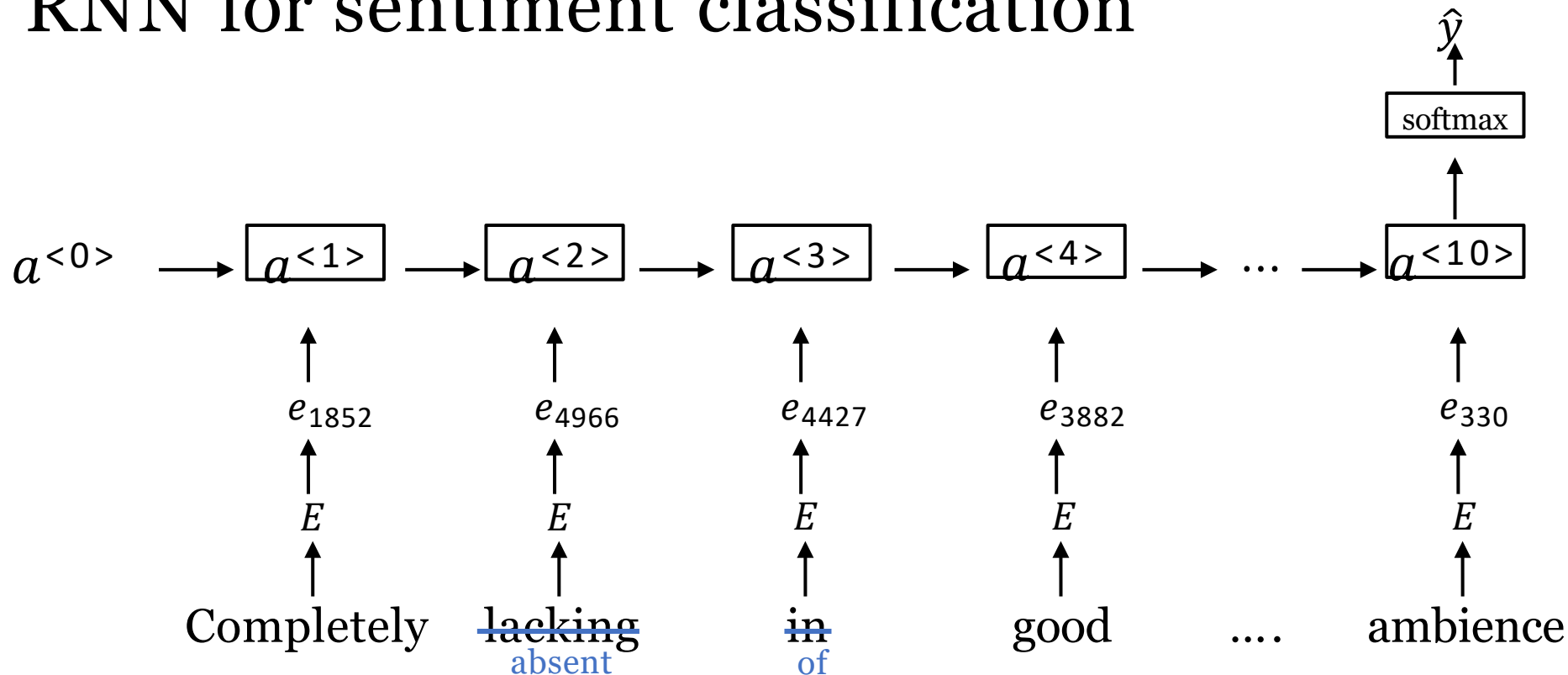
Completely lacking in good taste,  
good service, and good ambience.



# Simple sentiment classification model



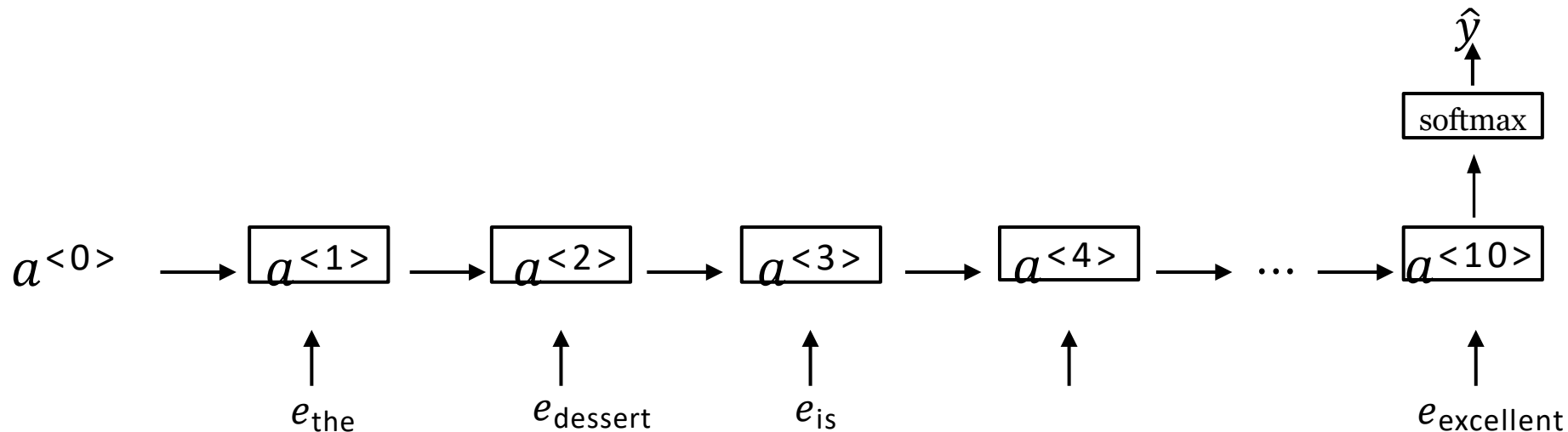
# RNN for sentiment classification



“Completely lacking in good  
taste, good service, and good  
ambience.”

many-to-one

# RNN for sentiment classification





# Sequence to sequence model

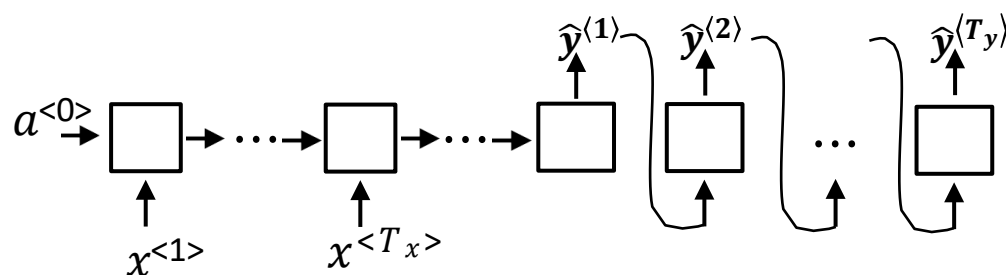
Machine translation

$x^{<1>}$   $x^{<2>}$   $x^{<3>}$   $x^{<4>}$   $x^{<5>}$

Jane visite l'Afrique en septembre

→ Jane is visiting Africa in September.

$y^{<1>}$   $y^{<2>}$   $y^{<3>}$   $y^{<4>}$   $y^{<5>}$   $y^{<6>}$



$$p\left(\hat{y}^{<1>}, \hat{y}^{<2>}, \hat{y}^{<3>}, \dots, \hat{y}^{<T_y>} \mid x^{<1>}, x^{<2>}, \dots, x^{<T_x>}\right)$$

Conditional language model

# Generation, Temperature

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

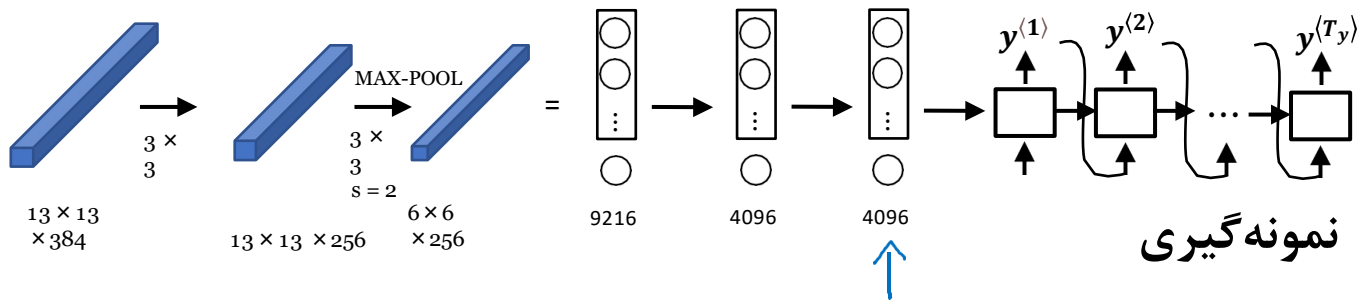
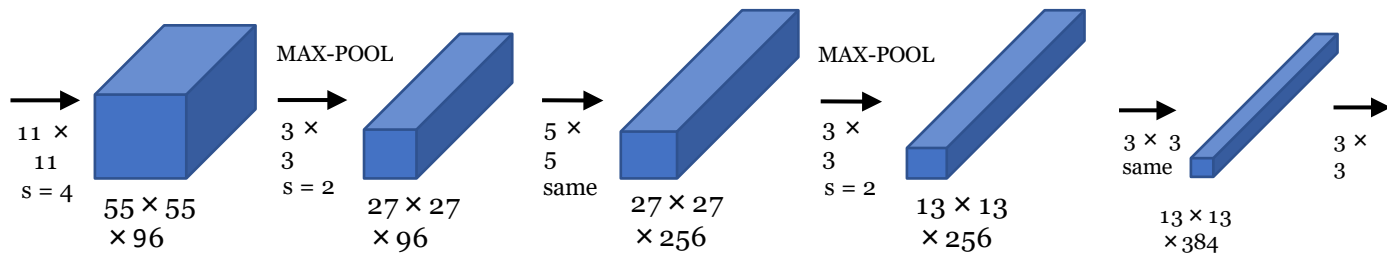
$$\text{softmax}(z_i; T) = \frac{e^{\frac{z_i}{T}}}{\sum_{j=1}^K e^{\frac{z_j}{T}}}$$

T=0; sharp distribution; T=inf uniform distribution

# Image captioning



$y^{<1>}$   $y^{<2>}$   $y^{<3>}$   $y^{<4>}$   $y^{<5>}$   $y^{<6>}$   
 A cat sitting on a chair



# Finding the most likely translation

Jane visite l'Afrique en septembre.

$$P(\overbrace{y^{<1>}, \dots, y^{<T_y>}}^{\text{English}} \mid \underbrace{x}_{\text{French}})$$

$$\arg \max P(y^{<1>}, y^{<2>}, \dots, y^{<T_y>} | x)$$

- Jane is visiting Africa in September.
- Jane is going to be visiting Africa in September.
- In September, Jane will visit Africa.
- Her African friend welcomed Jane in September.

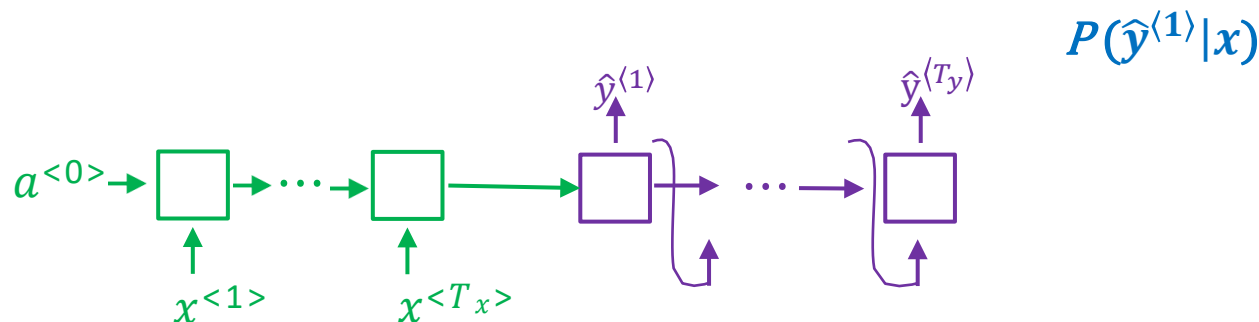
$$\arg \max P(\underbrace{y^{<1>}, \dots, y^{<T_y>}}_{10,000 \times 10,000 \times \dots} \mid x)$$

$y^{<1>}, \dots, y^{<T_y>}$

10,000x10,000x ...

$(10,000)^{T_y}$  Optimal Search

# Why not a greedy search?



$$\arg \max P(\hat{y}^{<1>}, \hat{y}^{<2>}, \dots, \hat{y}^{<T_y>}|x)$$

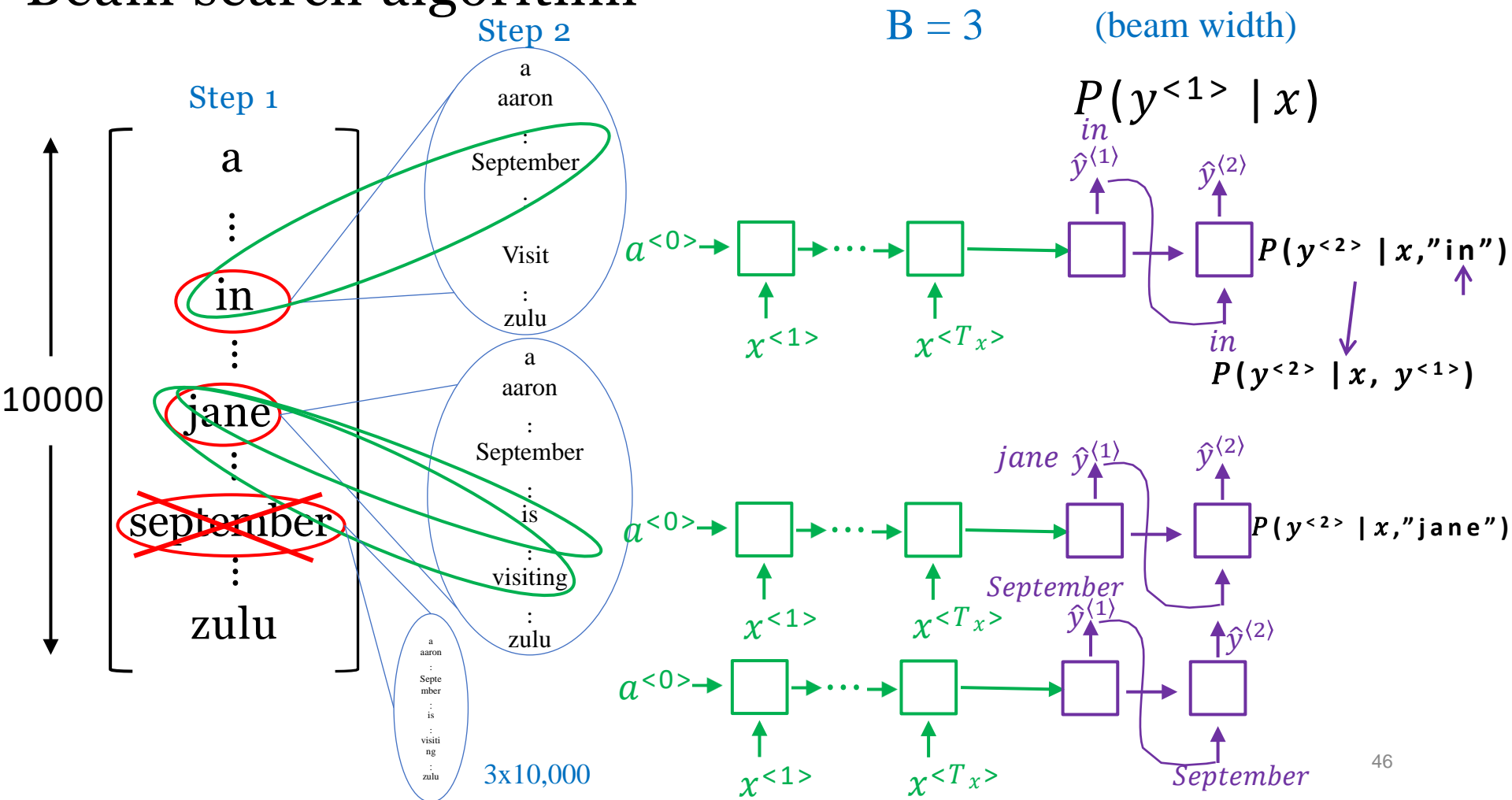
Greedy Search

10\*10,000

- Jane is visiting Africa in September.
- Jane is going to be visiting Africa in September.

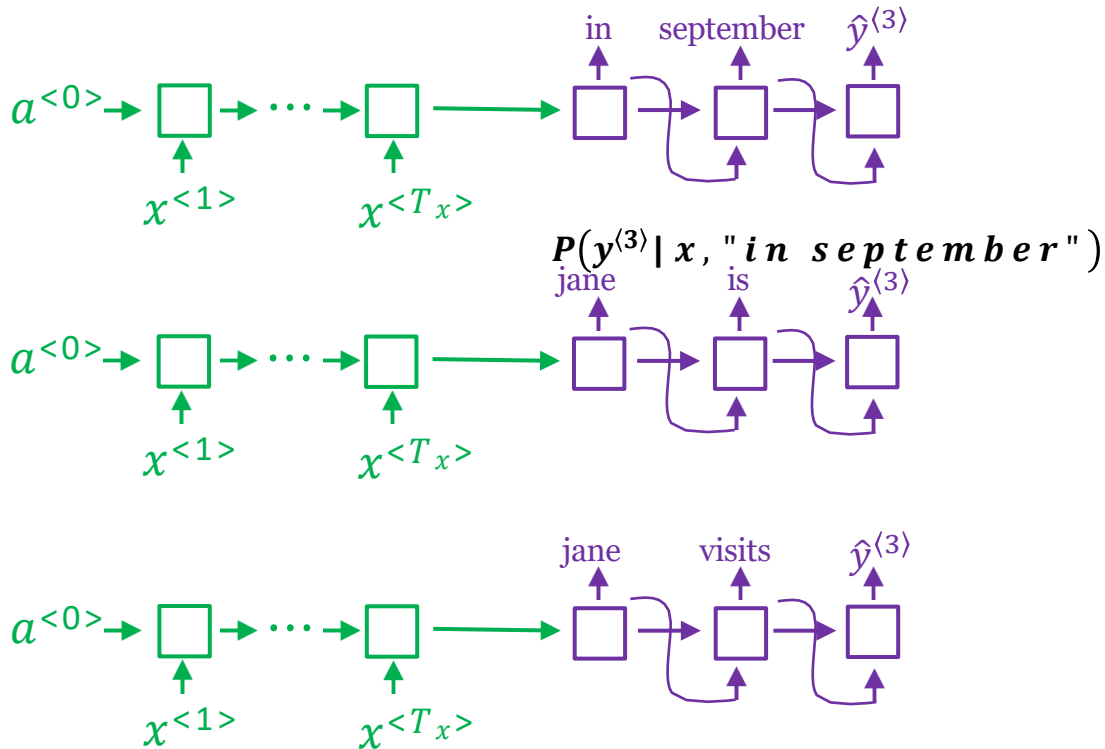
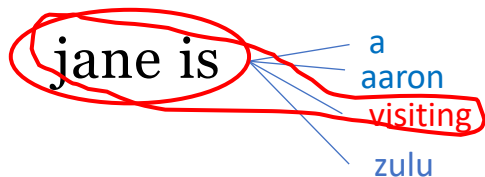
$$P(\text{Jane is visiting}|x) < P(\text{Jane is going}|x)$$

# Beam search algorithm



# Beam search ( $B = 3$ )

$B=1 \rightarrow$  greedy search



$$P(y^{<1>}, y^{<2>} | x)$$

jane visits africa in september. <EOS>

# Length normalization

- $\arg \max_y \sum_{t=1}^{T_y} \log P(y^{(t)} | x, y^{(1)}, \dots, y^{(t-1)})$
- $\frac{1}{T_y^\alpha} \sum_{t=1}^{T_y} \log P(y^{(t)} | x, y^{(1)}, \dots, y^{(t-1)})$   $\alpha = 0.7, 1, 0 \quad T_y = 1, 2, 3, \dots, 30$
- **Beam search discussion:**

Beam width B?

B: 1, 3, 10, 100

Large B: better result , slower

Small B: worse result, faster



# Example

→ RNN

→ Beam Search

B↑

Jane visite l'Afrique en septembre.

Human: Jane visits Africa in September.  $y^*$

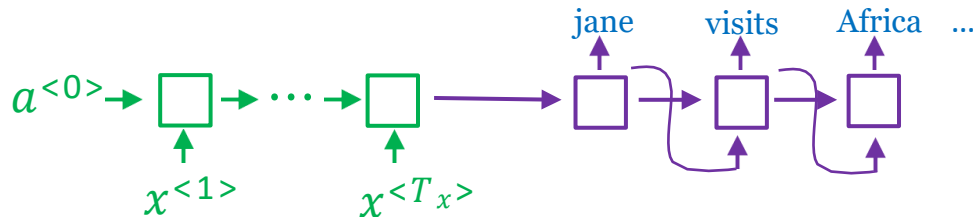
Algorithm: Jane visited Africa last September.  $\hat{y}$

RNN computes:

$$P(y^*|x) \begin{matrix} > \\ \leq \end{matrix} P(\hat{y}|x)$$

Case1:  $P(y^*|x) > P(\hat{y}|x)$   
Beam search is at fault

Case2:  $P(y^*|x) < P(\hat{y}|x)$   
RNN model is at fault



# Error analysis on beam search

$$P(y^*|x)$$

$$P(\hat{y}|x)$$

Human: Jane visits Africa in September. ( $y^*$ )

Algorithm: Jane visited Africa last September. ( $\hat{y}$ )

Case 1:  $P(y^*|x) > P(\hat{y}|x)$   $\arg \max P(y|x)$

Beam search chose  $\hat{y}$ . But  $y^*$  attains higher  $P(y|x)$ .

Conclusion: Beam search is at fault.

Case 2:  $P(y^*|x) \leq P(\hat{y}|x)$

$y^*$  is a better translation than  $\hat{y}$ . But RNN predicted  $P(y^*|x) < P(\hat{y}|x)$ .

Conclusion: RNN model is at fault.

# Error analysis process

Human	Algorithm	$P(y^* x)$	$P(\hat{y} x)$	At fault?
Jane visits Africa in September.	Jane visited Africa last September.	$2 \times 10^{-10}$	$1 \times 10^{-10}$	B
...	...	---	---	R
				B
				R
				R

Figures out what fraction of errors are “due to” beam search vs. RNN model

# Evaluating machine translation

- French: Le chat est sur le tapis

- Ref1: the cat is on the mat

Ref\_count=2

- Ref2: there is a cat on the mat

Ref\_count=1

- MT output: the the the the the the the

Count\_clip=min(max\_ref\_count,count)

- Precision= 7/7

modified precision =2/7

	Count	Max ref count	Count clip
the	7	2	2

# Unigram

- Ref1: the cat is on the mat
- Ref2: there is a cat on the mat
- MT output: the cat the cat on the mat

the	cat	On	mat
2	1	1	1
1	1	1	1

Count\_clip=min(max\_ref\_count,count)

- precision =5/7

unigram	Count	Max ref count	Count clip
the	3	2	2
cat	2	1	1
on	1	1	1
mat	1	1	1
	7		5

# Bigram

- Ref1: the cat is on the mat
- Ref2: there is a cat on the mat
- MT output: the cat the cat on the mat
- precision = 4/6

The cat	Cat the	Cat On	On the	The mat
1	0	0	1	1
0	0	1	1	1

Count\_clip=min(max\_ref\_count,count)

bigram	Count	Max ref count	Count clip
The cat	2	1	1
Cat the	1	0	0
Cat on	1	1	1
On the	1	1	1
The mat	1	1	1
	6		4

Bleu (Bilingual Evaluation Understudy) score

$$P_n = \frac{\sum_{ngram \in \hat{y}} count\_clip(ngram)}{\sum_{ngram \in \hat{y}} count(ngram)}$$

$$Blue_{score} = BP * \exp\left(\frac{1}{\#ngram} \sum_{n=1}^{\#ngram} P_n\right)$$

c: length of the candidate translation

r: effective reference corpus length

Brevity Penalty

$$BP = \begin{cases} 1, & c > r \\ \exp\left(1 - \frac{r}{c}\right), & c \leq r \end{cases}$$