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

Department of Electrical and computer engineering, Isfahan university of technology, Isfahan, Iran

Examples of sequence data

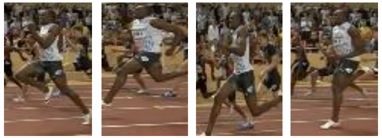
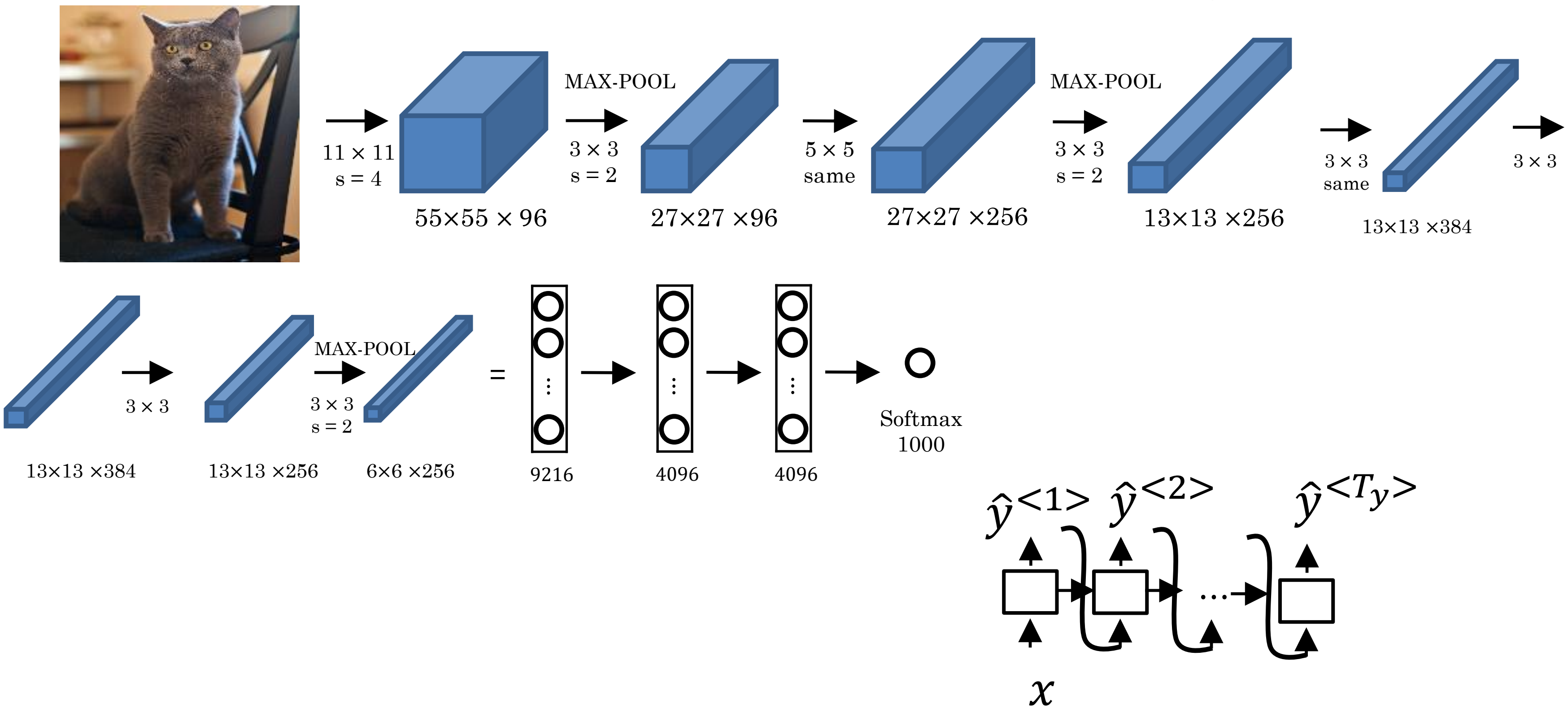
Speech recognition		→	"The quick brown fox jumped over the lazy dog."
Music generation	∅	→	
Sentiment classification	"There is nothing to like in this movie."	→	
DNA sequence analysis	AGCCCCTGTGAGGAACTAG	→	AGCCCCTGTGAGGAACTAG
Machine translation	Voulez-vous chanter avec moi?	→	Do you want to sing with 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.

$X^{\langle 1 \rangle}$ $X^{\langle 2 \rangle}$ $X^{\langle 3 \rangle}$ … … $X^{\langle t \rangle}$ … … $X^{\langle 9 \rangle}$ T_x=9

- y: 1 1 0 1 1 0 0 0 0 T_y=9

$y^{\langle 1 \rangle}$ $y^{\langle 2 \rangle}$ $y^{\langle 3 \rangle}$ … … … $y^{\langle 9 \rangle}$

- $X^{(i)\langle t \rangle}$

داده آموزشی i ام

$T_x^{(i)} = 9$

$T_y^{\langle i \rangle}$

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

Representing words

- Dictionary: 30000, 50000

$$
\text{Vocabulary} \quad
\begin{array}{r}
0 \\ 1 \\ \\ \\ \\ \\ 367 \\ \\ \\ \\ 4075 \\ \\ \\ \\ 6830 \\ \\ \\ \\ 10000
\end{array}
\begin{bmatrix}
a \\ and \\ . \\ . \\ . \\ . \\ Deep \\ . \\ . \\ . \\ hassan \\ . \\ . \\ . \\ Mobile \\ . \\ . \\ . \\ zigzag
\end{bmatrix}
\; 10000
$$

$$
\begin{bmatrix} a \\ and \\ ali \\ . \\ . \\ . \\ . \end{bmatrix}
\qquad
\begin{bmatrix} 0 \\ 0 \\ 1 \\ . \\ . \\ . \\ 0 \end{bmatrix} \; 10000
$$

One-hot vector : ali

- So, if you have chosen a dictionary of 10,000 words, what you can do is then use one-hot representations to represent each of these words.

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

$X^{\langle 1 \rangle}$ $X^{\langle 2 \rangle}$ $X^{\langle 3 \rangle}$ … … $X^{\langle t \rangle}$ … … $X^{\langle 9 \rangle}$

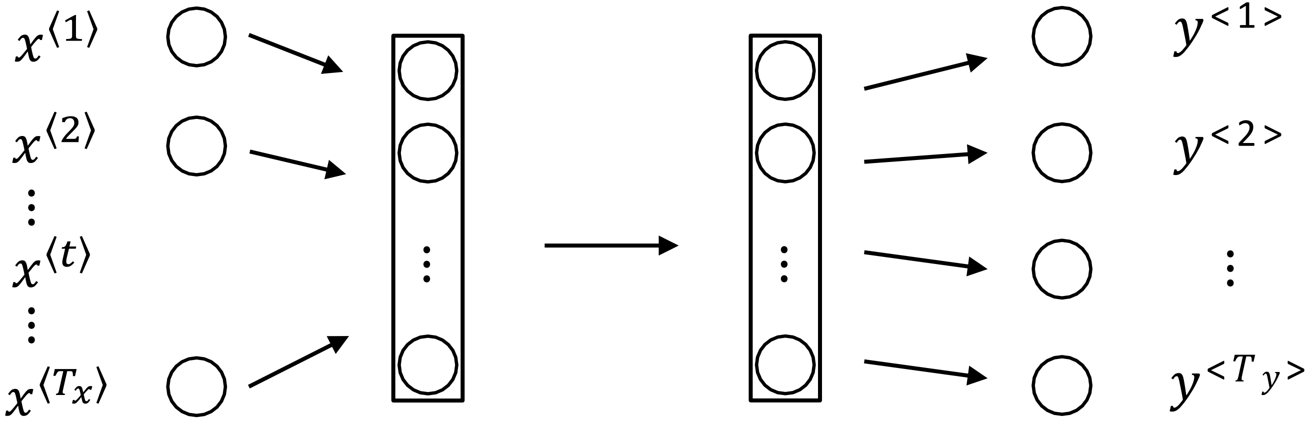
$$
\begin{array}{cc}
\begin{bmatrix} 0 \\ 0 \\ 0 \\ . \\ 1 \\ . \\ . \\ 0 \end{bmatrix}
\begin{array}{l} \\ \\ \\ \\ 4075 \\ \\ \\ \\ \end{array}
&
\begin{bmatrix} 0 \\ 0 \\ 0 \\ . \\ . \\ 1 \\ . \\ 0 \end{bmatrix}
\begin{array}{l} \\ \\ \\ \\ \\ 6830 \\ \\ \\ \end{array}
\end{array}
$$

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

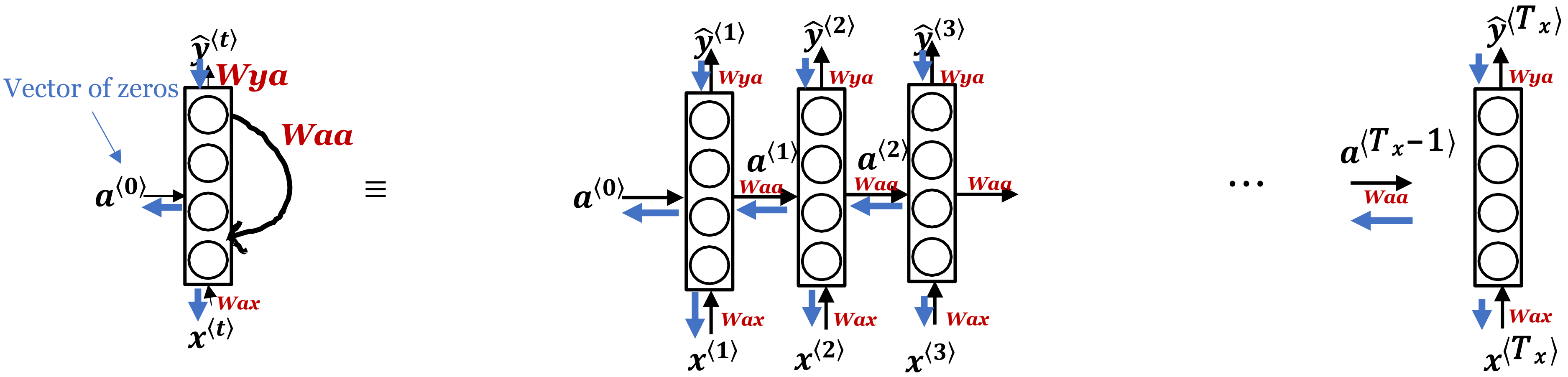
Why not a standard network?



Problems:

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

Recurrent Neural Networks



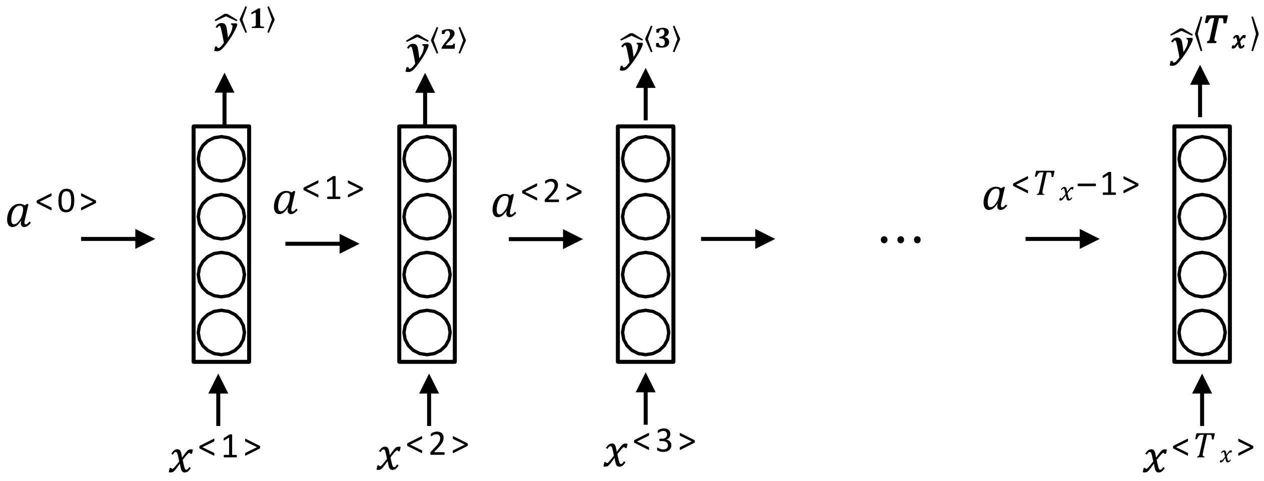
$$L^{\langle t \rangle}\left(\hat{y}^{\langle t \rangle}, y^{\langle t \rangle}\right) = -y^{\langle t \rangle}\log \hat{y}^{\langle t \rangle} - \left(1 - y^{(t)}\right)\log\left(1 - \hat{y}^{\langle t \rangle}\right)$$

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

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

Backward propagation through time

Forward Propagation



$\boldsymbol{a}^{\langle 0\rangle} = \vec{\mathbf{0}}$

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

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

$$\boxed{\begin{aligned} a^{\langle t\rangle} &= g(w_{aa}a^{\langle t-1\rangle} + w_{ax}x^{\langle t\rangle} + b_a) \\ \hat{y}^{\langle t\rangle} &= g(w_{ya}a^{\langle t\rangle} + b_y) \end{aligned}}$$

Simplified RNN notation

$$a^{\langle t\rangle} = g(\overset{(100,100)}{w_{aa}}\underset{100}{a^{\langle t-1\rangle}} + \overset{(100,10000)}{w_{ax}}\underset{10000}{x^{\langle t\rangle}} + \underset{100}{b_a}) \qquad a^{\langle t\rangle} = g(w_a[a^{\langle t-1\rangle}, x^{\langle t\rangle}] + b_a)$$

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

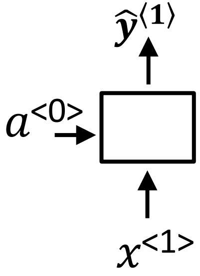
- $a^{\langle t\rangle} = g\left(w_a \underbrace{[a^{\langle t-1\rangle\prime}, x^{\langle t\rangle\prime}]'} + b_a\right)$

$$\begin{array}{cll} \begin{bmatrix} a^{\langle t-1\rangle} \\ \\ x^{\langle t\rangle} \end{bmatrix} & \begin{array}{l} 100 \\ \\ 10000 \end{array} & 10100 \end{array}$$

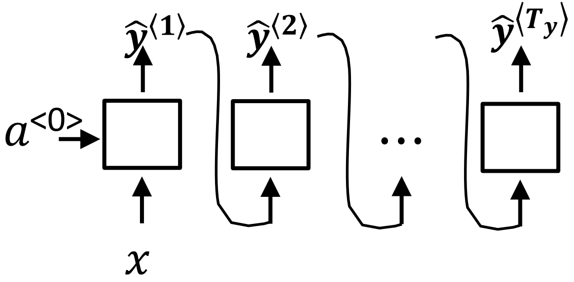
$$\underset{(100,10100)}{w_a} = \overset{100}{\updownarrow}[\overset{10000}{w_{aa}} \vdots \overset{100}{w_{ax}}]$$

$$[w_{aa} \vdots w_{ax}]\begin{bmatrix} a^{\langle t-1\rangle} \\ x^{\langle t\rangle} \end{bmatrix} = w_{aa}a^{\langle t-1\rangle} + w_{ax}x^{\langle t\rangle}$$

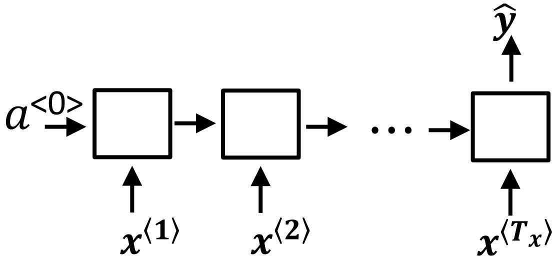
Summary of RNN types



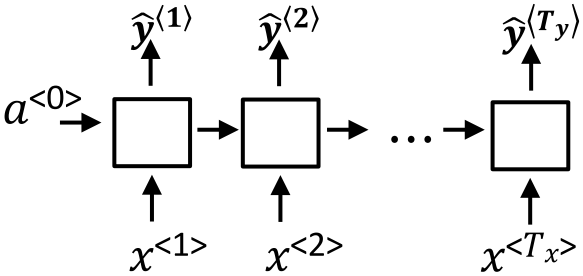
One to one



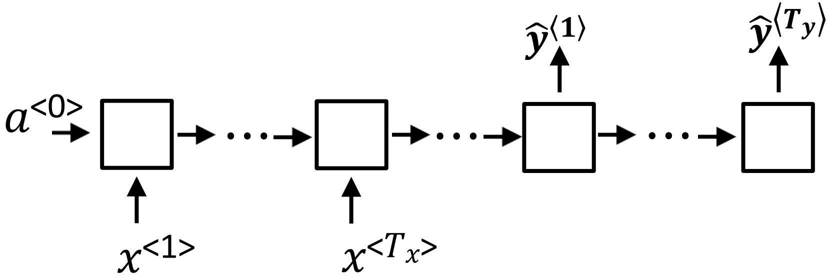
One to many



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

Language modelling with an RNN

Training set: large corpus of english text.

Tokenize

Cats average 15 hours of sleep a day. <EOS>

$y^{\langle 1 \rangle}$ $y^{\langle 2 \rangle}$ $y^{\langle 3 \rangle}$ ... $y^{\langle 9 \rangle}$

$x^{\langle t \rangle} = y^{\langle t-1 \rangle}$

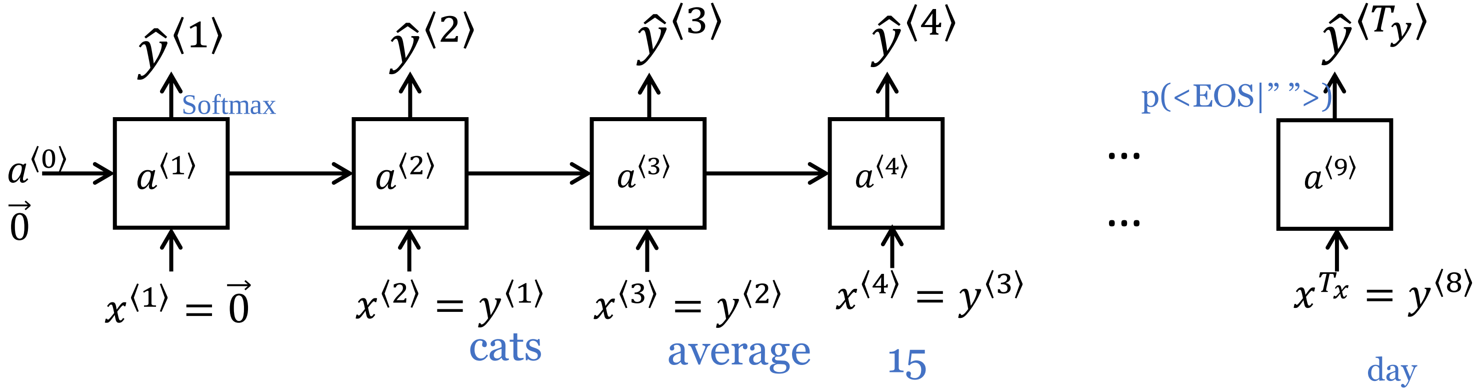
The Egyptian ~~Mau~~ is a bread of cat. <EOS>

<UNK>

RNN model

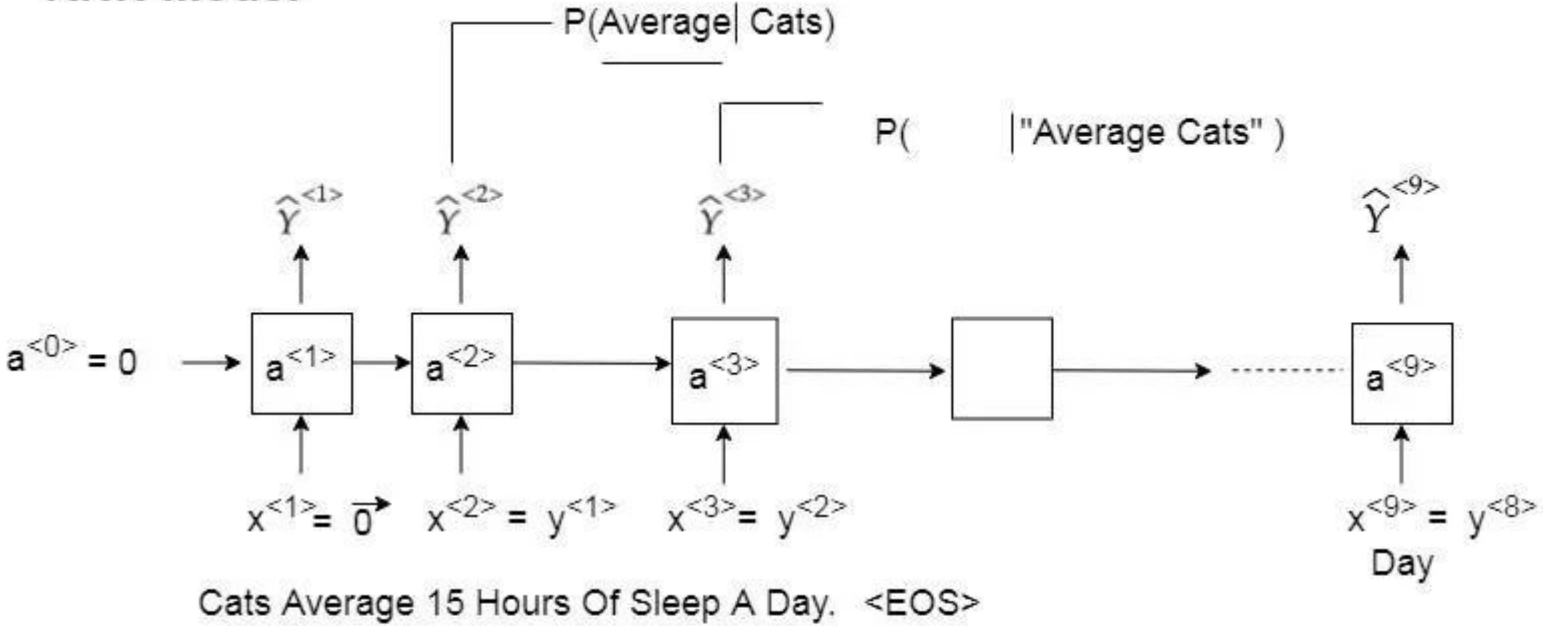
Cats average 15 hours of sleep a day.

- p(a) p(cat) p(bread) … p(Eat)
- p(a|cats) p(cat|cats) p(bread|cats) p(average|cats) …. p(Eat|cats)
- P(a|cats average) p(cats|cats average) … p(15|cats average) …. P(eat|cats average)



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

RNN Model



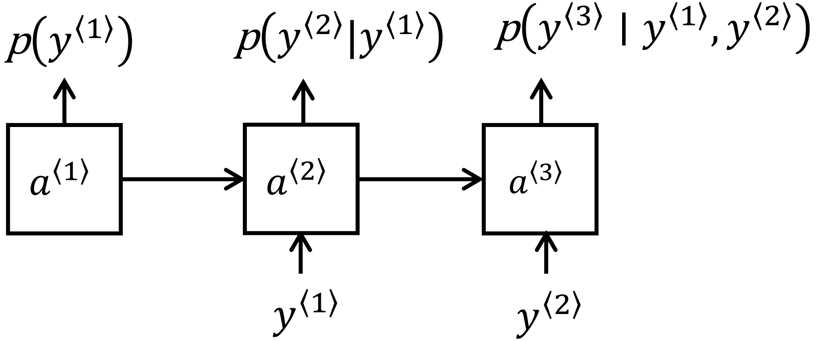
Cost Function $$L\left(\hat{Y}^{<t>}, Y^{<t>}\right) = -\sum_{i} Y_i^{<t>} \log \hat{Y}_i^{<t>}$$ SoftMax Loss Function

$$L = \sum_{t} L^{<t>}\left(\hat{Y}^{<t>}, Y^{<t>}\right)$$ Overall Loss Function

RNN model

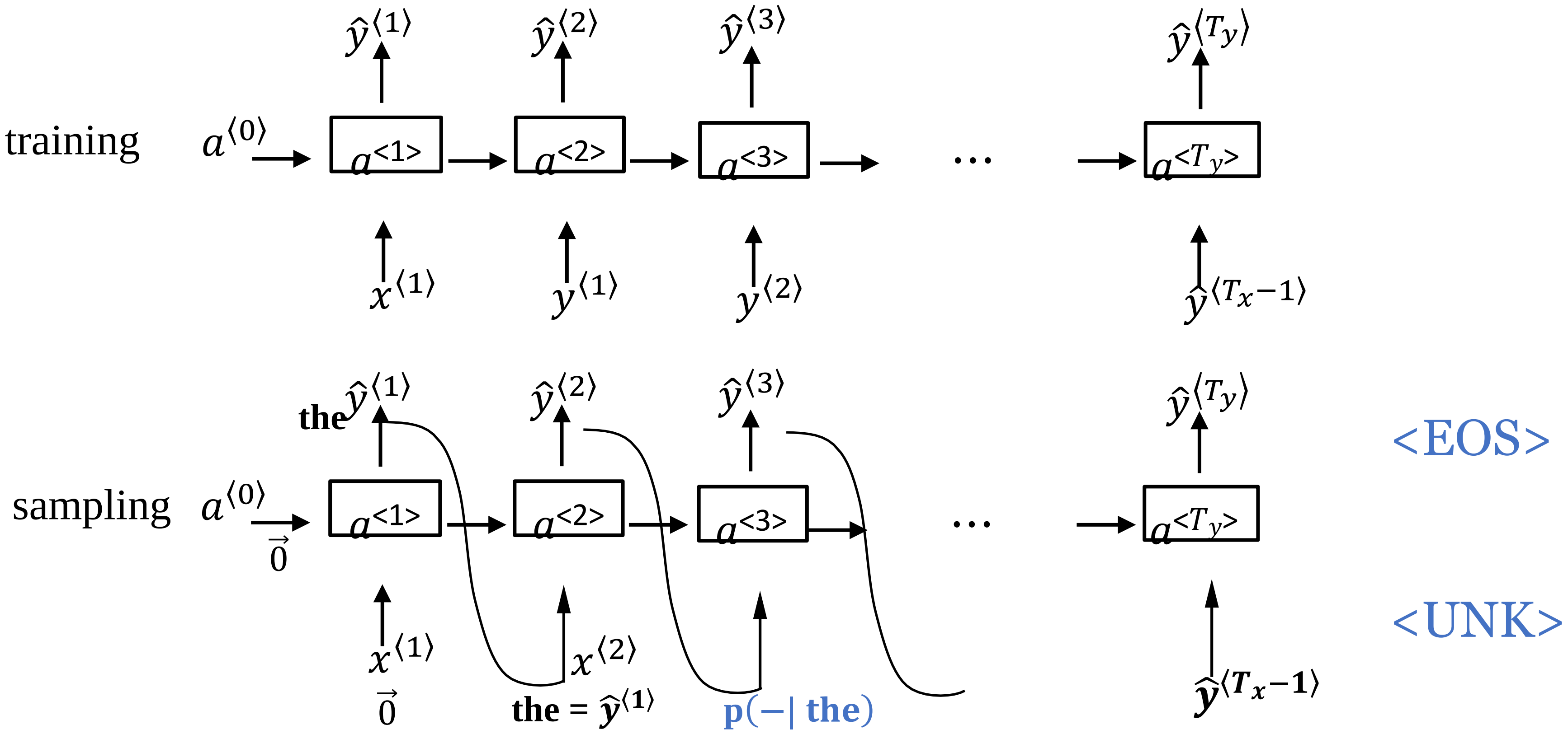
- Cats average 15 hours of sleep a day. <EOS>

- $p\left(y^{(1)}, y^{(2)}, y^{(3)}\right) = p\left(y^{(1)}\right) \cdot p\left(y^{(2)} \mid y^{(1)}\right) \cdot p\left(y^{(3)} \mid y^{(1)}, y^{(2)}\right)$



$p(- \mid$ "*cats* average 15 hours of sleep a day"$)$

Sampling a sequence from a trained RNN

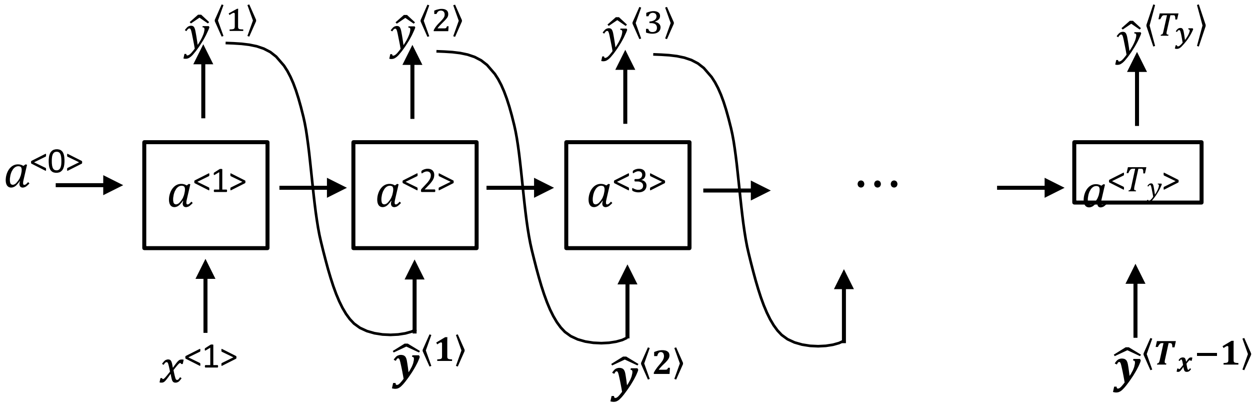


Character-level language model

- Vocabulary = [a, aaron, …, zulu, <UNK>]

- Vocabulary = [a, b, c, ... , z , ␣ , . , ..., 0, ... , 9, A, ... ,Z]

- Cat average … Mau



Sequence generation

News

President enrique peña nieto, announced sench's sulk former coming football langston paring.

"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.

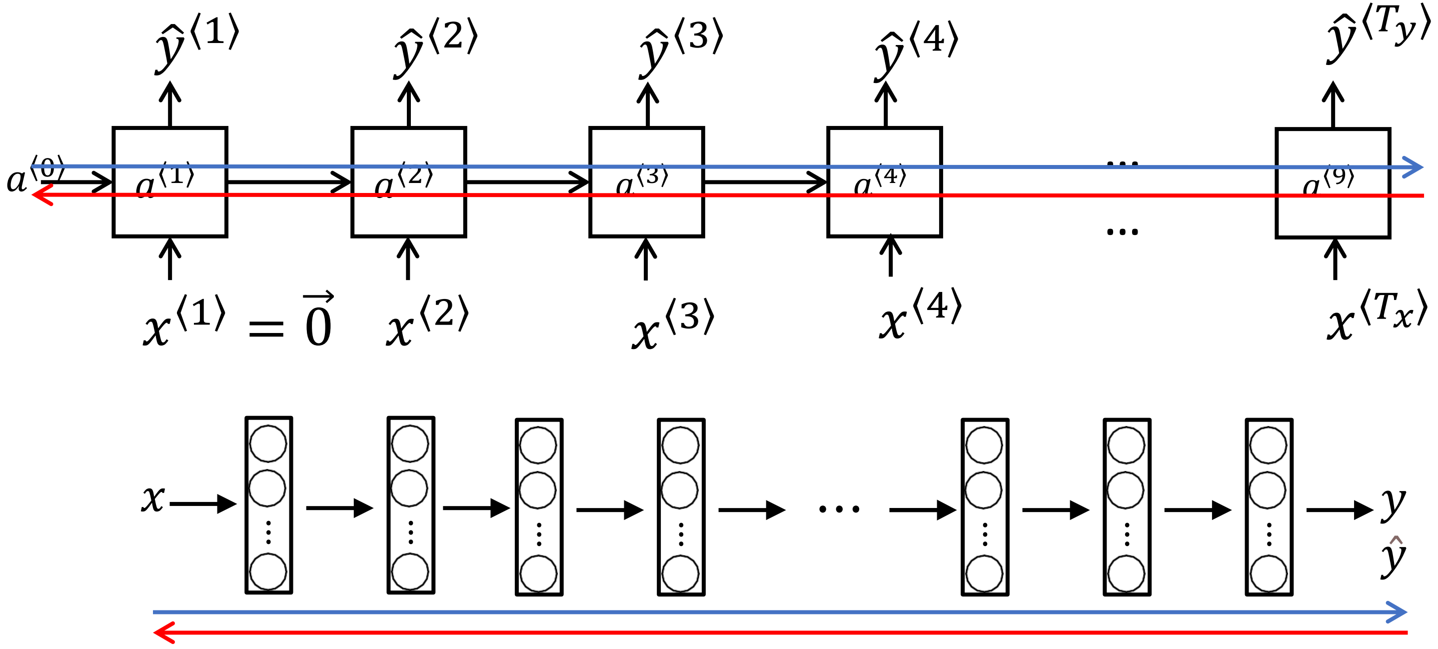
And subject of this thou art another this fold.

When besser be my love to me see sabl'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

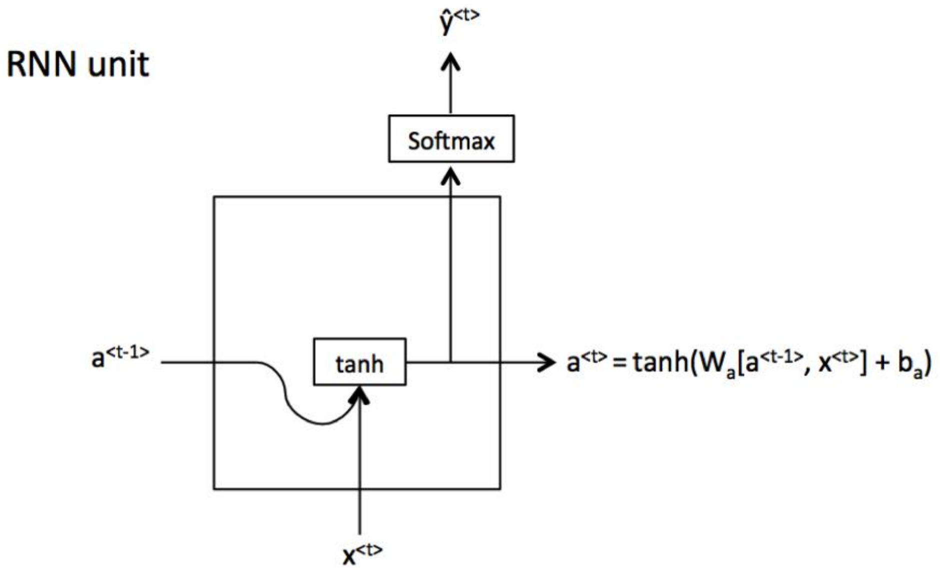


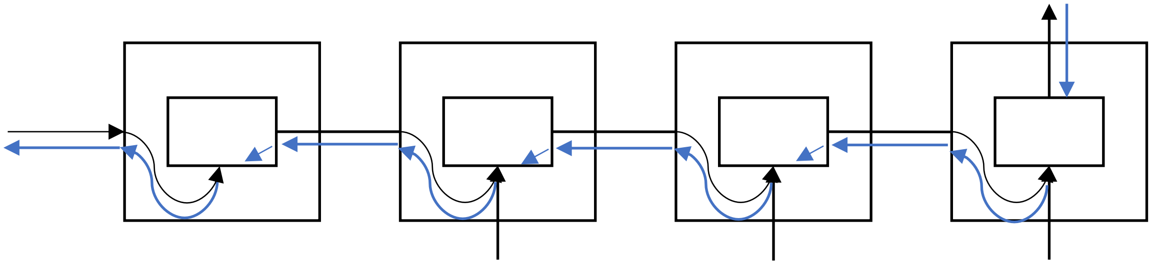
Exploding gradients. NaN gradient clipping

RNN Unit

- $a^{\langle t\rangle} = g\left(W_a\left[a^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + b_a\right)$

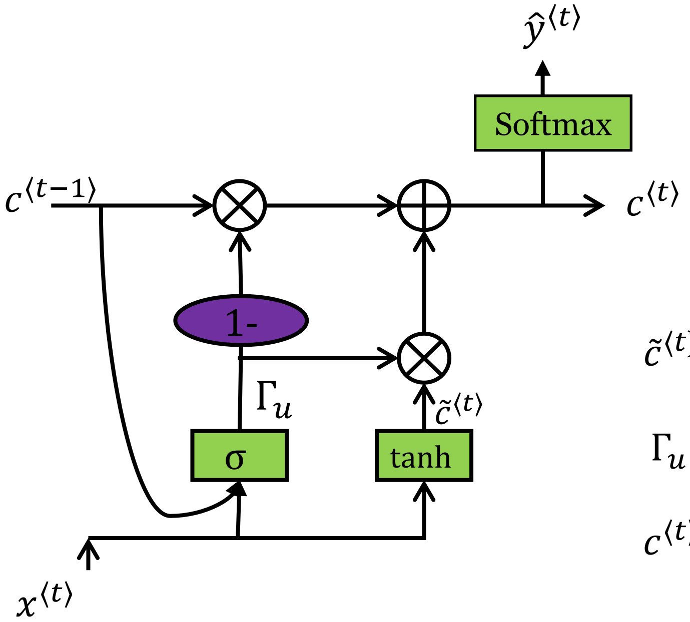
tanh





GRU (simplified)

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



C : memory cell

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

$$\Gamma_u = \sigma\left(w_{\mathrm{u}}\left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_{\mathrm{u}}\right)$$

$$c^{\langle t \rangle} = \Gamma_{\mathrm{u}} * \tilde{c}^{\langle t \rangle} + (1 - \Gamma_u) * c^{\langle t-1 \rangle}$$

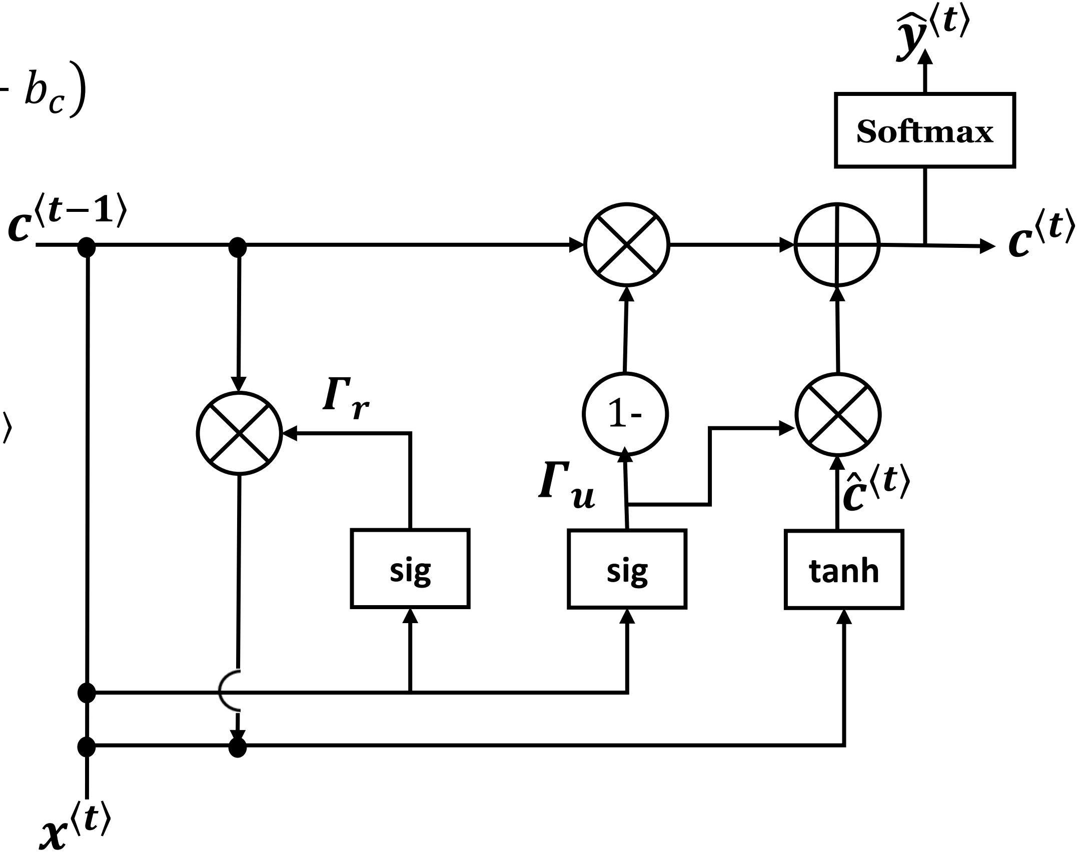
Full GRU

- $\hat{c}^{\langle t \rangle} = \tanh\left(w_c\left[\Gamma_r * c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_c\right)$

- $\Gamma_u = \sigma\left(w_{\mathrm{u}}\left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_{\mathrm{u}}\right)$

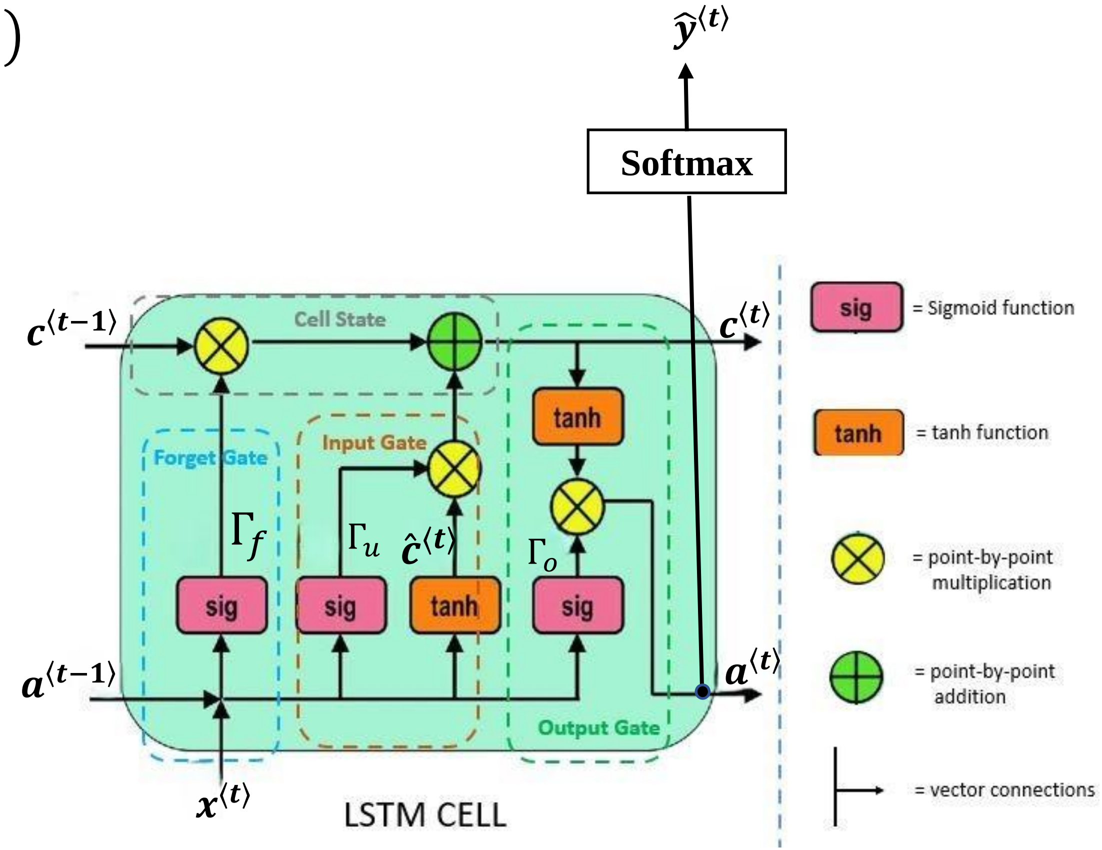
- $\Gamma_r = \sigma\left(w_{\mathrm{r}}\left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_{\mathrm{r}}\right)$

- $c^{\langle t \rangle} = \Gamma_{\mathrm{u}} * \hat{c}^{\langle t \rangle} + (1 - \Gamma_u) * c^{\langle t-1 \rangle}$

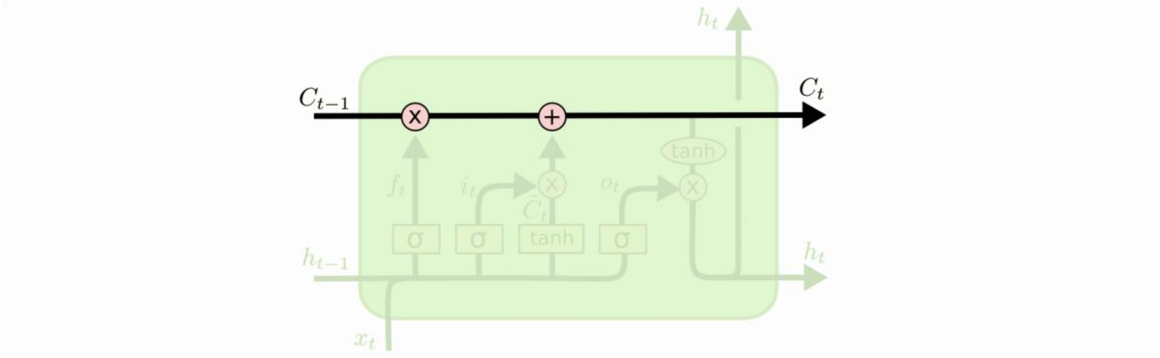


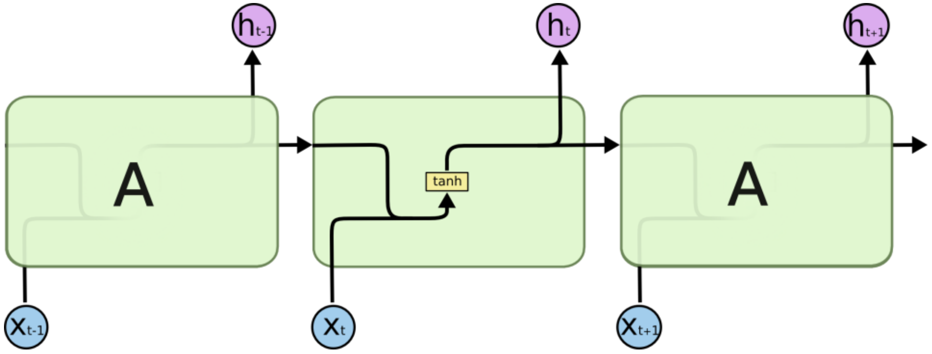
LSTM

- $\hat{c}^{\langle t \rangle} = \tanh\left(w_c\left[a^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_c\right)$
- $\Gamma_u = \sigma\left(w_{\mathrm{u}}\left[a^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_{\mathrm{u}}\right)$
- $\Gamma_f = \sigma\left(w_{\mathrm{f}}\left[a^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_{\mathrm{f}}\right)$
- $\Gamma_o = \sigma\left(w_{\mathrm{o}}\left[a^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_{\mathrm{o}}\right)$
- $c^{\langle t \rangle} = \Gamma_u * \hat{c}^{\langle t \rangle} + \Gamma_f * c^{\langle t-1 \rangle}$
- $a^{\langle t \rangle} = \Gamma_o * \tanh\left(c^{\langle t \rangle}\right)$



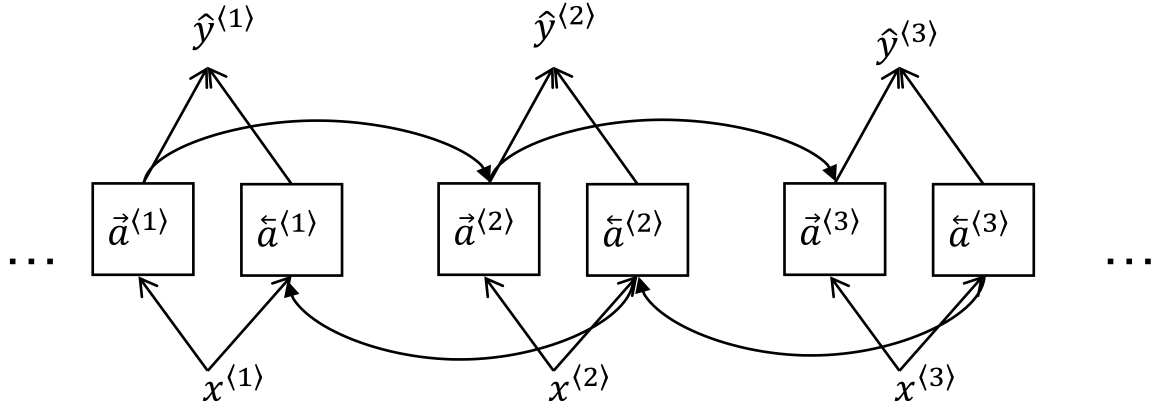
Vanishing gradient





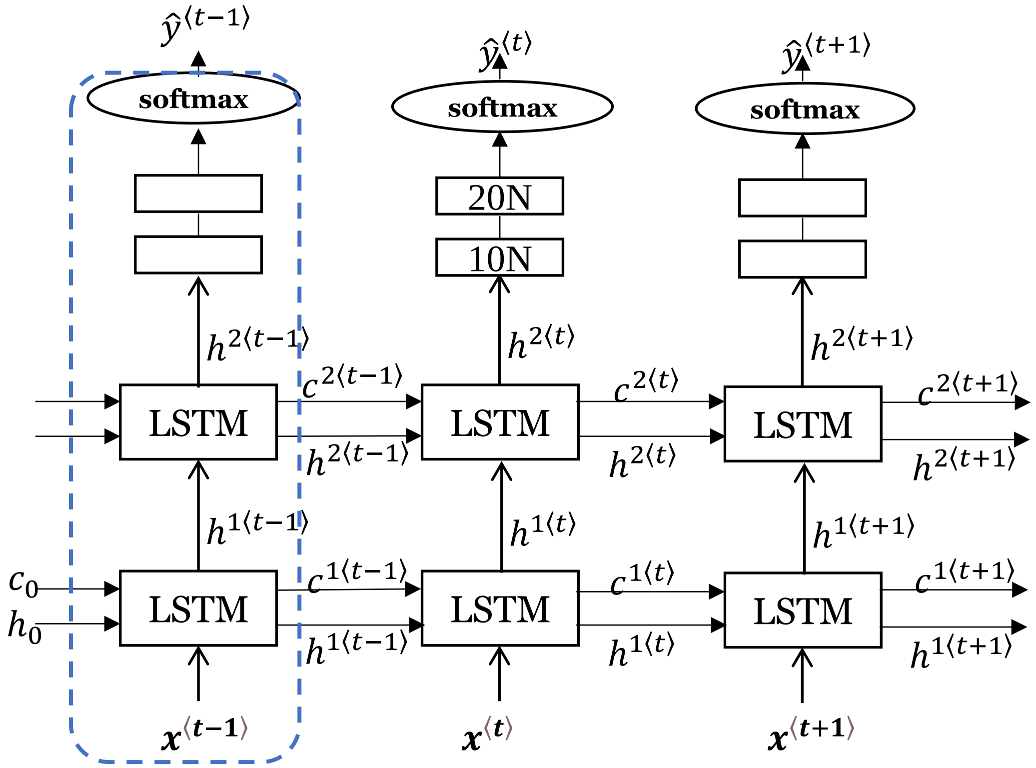
Bidirectional RNN (BRNN)

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



- $\hat{y}^{\langle t \rangle} = g\left(w_y\left[\vec{a}^{\langle t \rangle}, \overleftarrow{a}^{\langle t \rangle}\right] + by\right)$

Deep RNN (Stacked RNN)



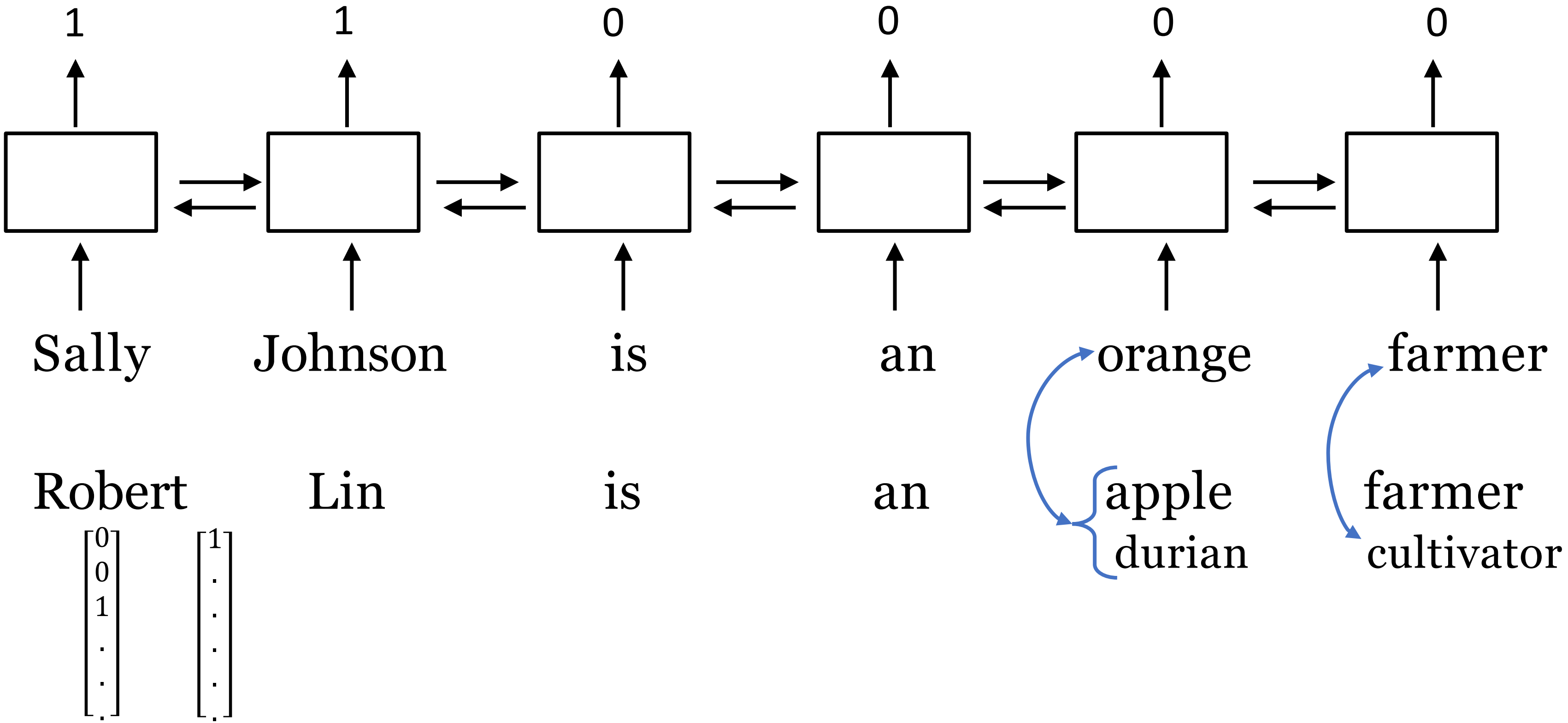
RNN-Example

```
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, ..., zulu, <UNK>]

|v| = 10000

1-hot representation

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

$$\begin{matrix} \\ \\ \\ \\ \\ 5391 \rightarrow \\ \\ \\ \\ \end{matrix}\begin{bmatrix}0\\0\\0\\0\\\vdots\\1\\\vdots\\0\\0\end{bmatrix} \quad \begin{matrix} \\ \\ \\ \\ \\ \\ 9853 \rightarrow \\ \\ \\ \end{matrix}\begin{bmatrix}0\\0\\0\\0\\0\\\vdots\\1\\\vdots\\0\end{bmatrix} \quad \begin{matrix} \\ \\ \\ \\ 4914 \rightarrow \\ \\ \\ \\ \\ \end{matrix}\begin{bmatrix}0\\0\\0\\\vdots\\1\\\vdots\\0\\0\\0\end{bmatrix} \quad \begin{matrix} \\ \\ \\ \\ \\ \\ 7157 \rightarrow \\ \\ \\ \end{matrix}\begin{bmatrix}0\\0\\0\\0\\0\\\vdots\\1\\\vdots\\0\end{bmatrix} \quad \begin{matrix} \\ \\ 456 \rightarrow \\ \\ \\ \\ \\ \\ \\ \end{matrix}\begin{bmatrix}0\\\vdots\\1\\\vdots\\0\\0\\0\\0\\0\end{bmatrix} \quad \begin{matrix} \\ \\ \\ \\ \\ \\ 6257 \rightarrow \\ \\ \\ \end{matrix}\begin{bmatrix}0\\0\\0\\0\\0\\\vdots\\1\\\vdots\\0\end{bmatrix}$$

I want a glass of orange juice .

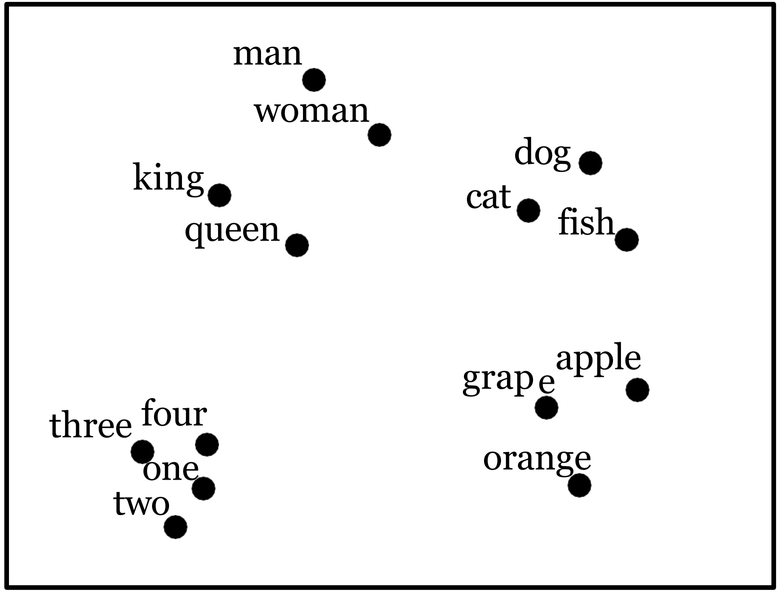
I want a glass of apple______.

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:



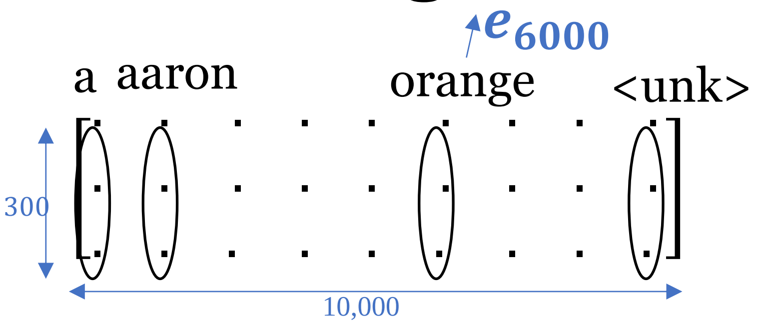
Man→ woman as King →?

$$e_{man} - e_{woman} \approx e_{king} - e_{w}?$$

$$\arg\max \ \ \mathrm{sim}\big(e_w, e_{king} - e_{man} + e_{woman}\big)$$

$$\mathbf{sim}(\boldsymbol{u}, \boldsymbol{v}) = \frac{\boldsymbol{u}^{\top}\boldsymbol{v}}{\| \boldsymbol{u} \| \| \boldsymbol{v} \|}$$

Embedding matrix



$$\begin{array}{cl} \begin{bmatrix}0\\0\\0\\ \cdot\\ \cdot\\ \cdot\\ 1\\ \cdot\\ \cdot\\ \cdot\\ 0\end{bmatrix} & \begin{array}{l} \\ \\ \\ \\ \\ \\ \leftarrow 6000 \\ \\ \\ \\ \\ \end{array} \end{array}$$

O_j

Word j^{th}

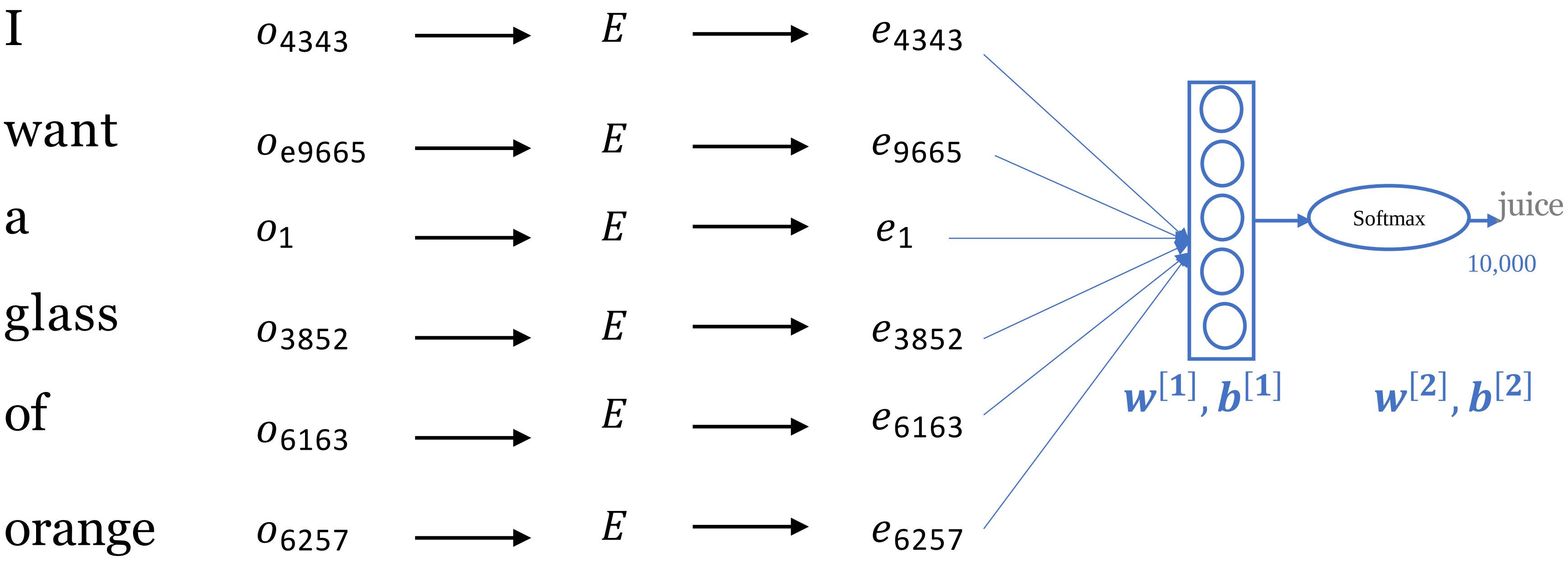
$$E \cdot O_{6000} = \begin{bmatrix} x\\ x\\ x\\ x\\ \cdot\\ \cdot\\ \cdot \end{bmatrix} \updownarrow 300 = e_{6000}$$

$$E \cdot O_j = e_j$$

Neural language model

I want a glass of orange juice.

4343 9665 1 3852 6163 6257



[Bengio et. al., 2003, A neural probabilistic language model]

Other context/target pairs

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

Context: Last 4 words.

4 words on left & right   a glass of orange _?_ to go along with

Last 1 word   orange _?_

Nearby 1 word   glass _?_

Skip gram

Skip-grams

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

Context	**target**
orange	juice
orange	glass
orange	my
.	

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

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

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

Softmax:

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

Sentiment classification problem

x

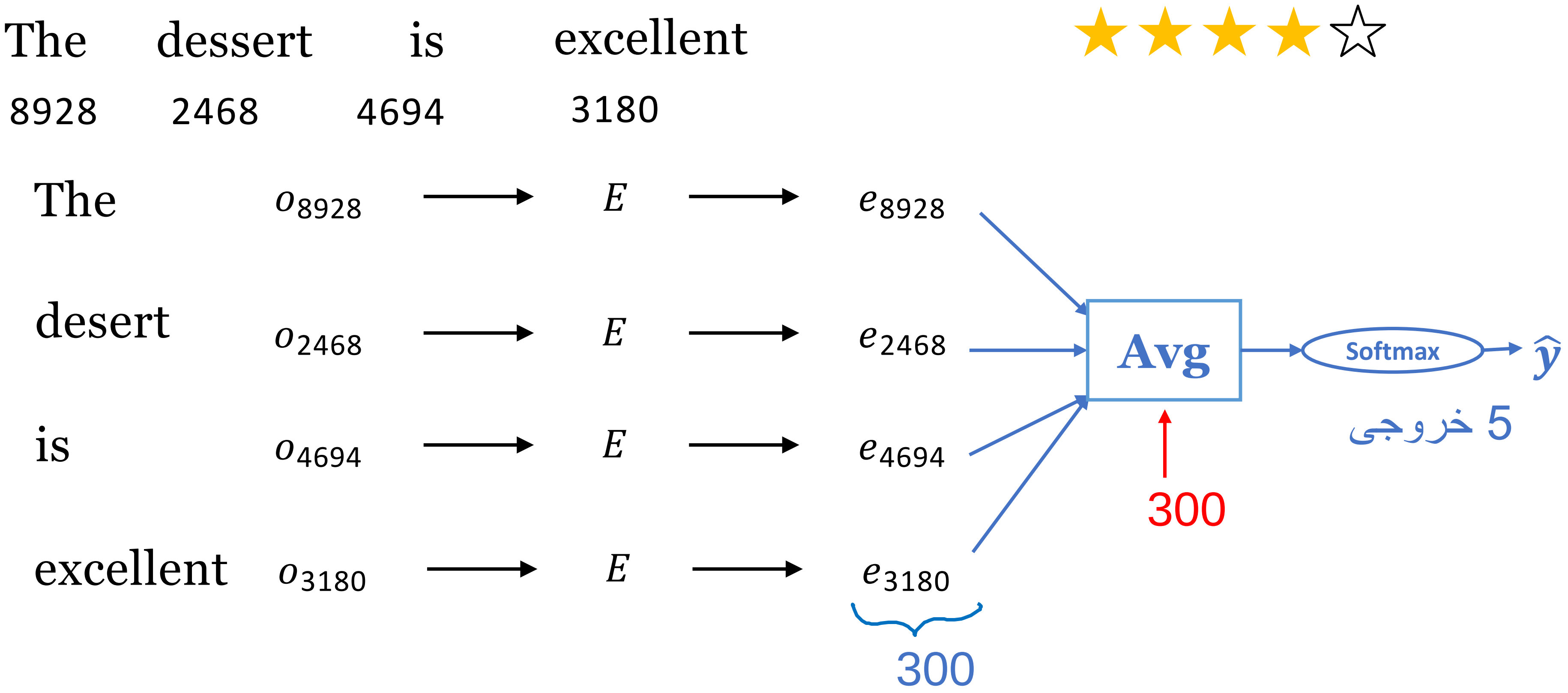
The dessert is excellent. ★★★★☆

Service was quite slow. ★★☆☆☆

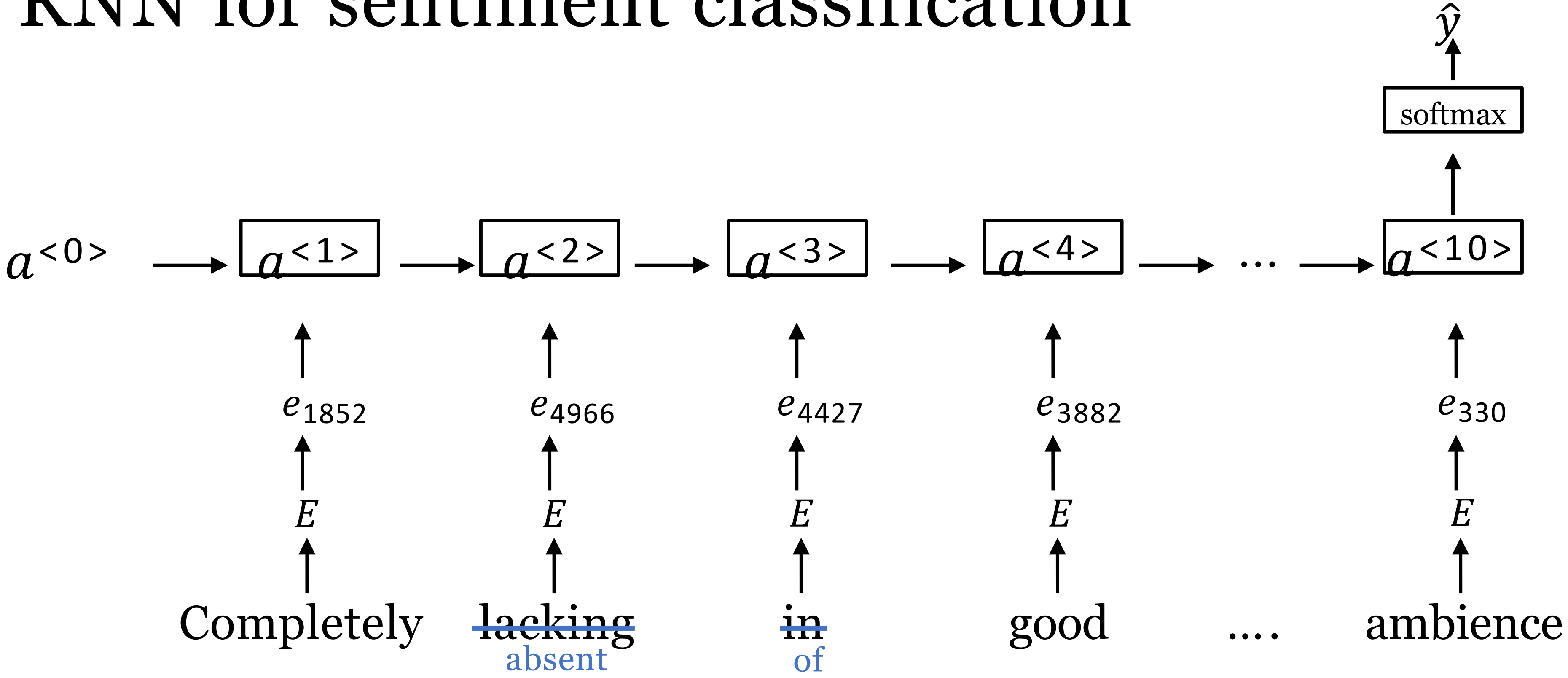
Good for a quick meal, but nothing special. ★★★☆☆

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

Simple sentiment classification model



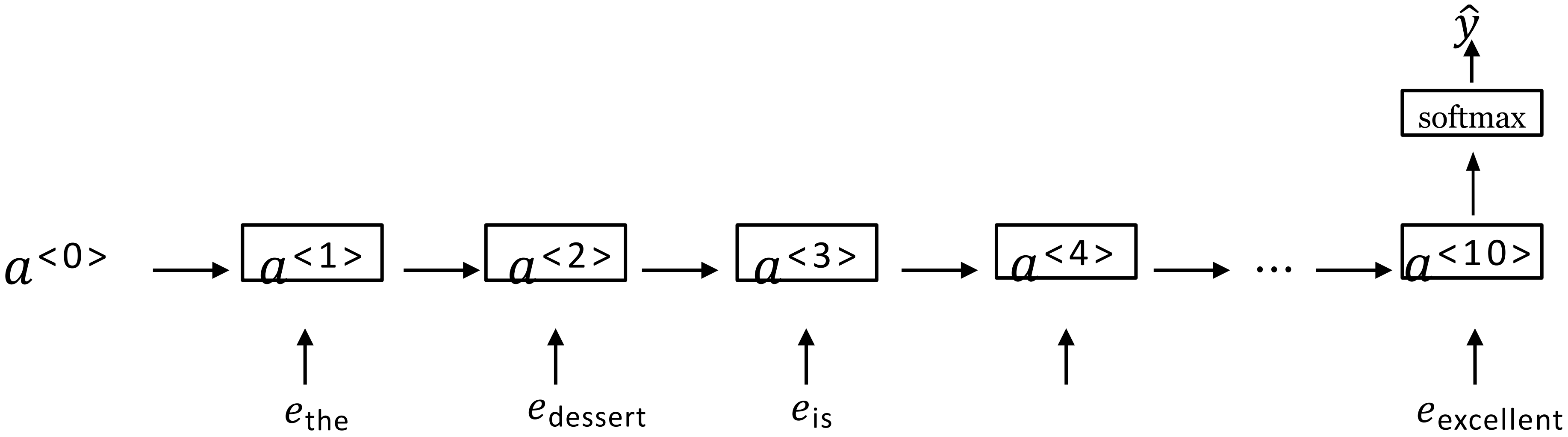
RNN for sentiment classification



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

many-to-one

RNN for sentiment classification



Sequence to sequence model

Machine translation

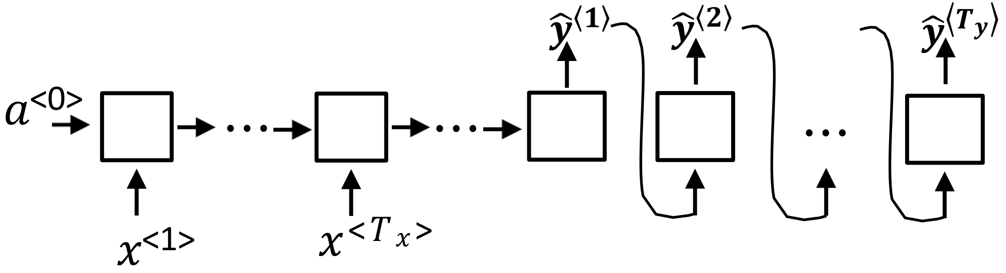
$x^{<1>}$ $x^{<2>}$ $x^{<3>}$ $x^{<4>}$ $x^{<5>}$

Jane visite l’Afrique en septembre

⟶ Jane is visiting Africa in September.

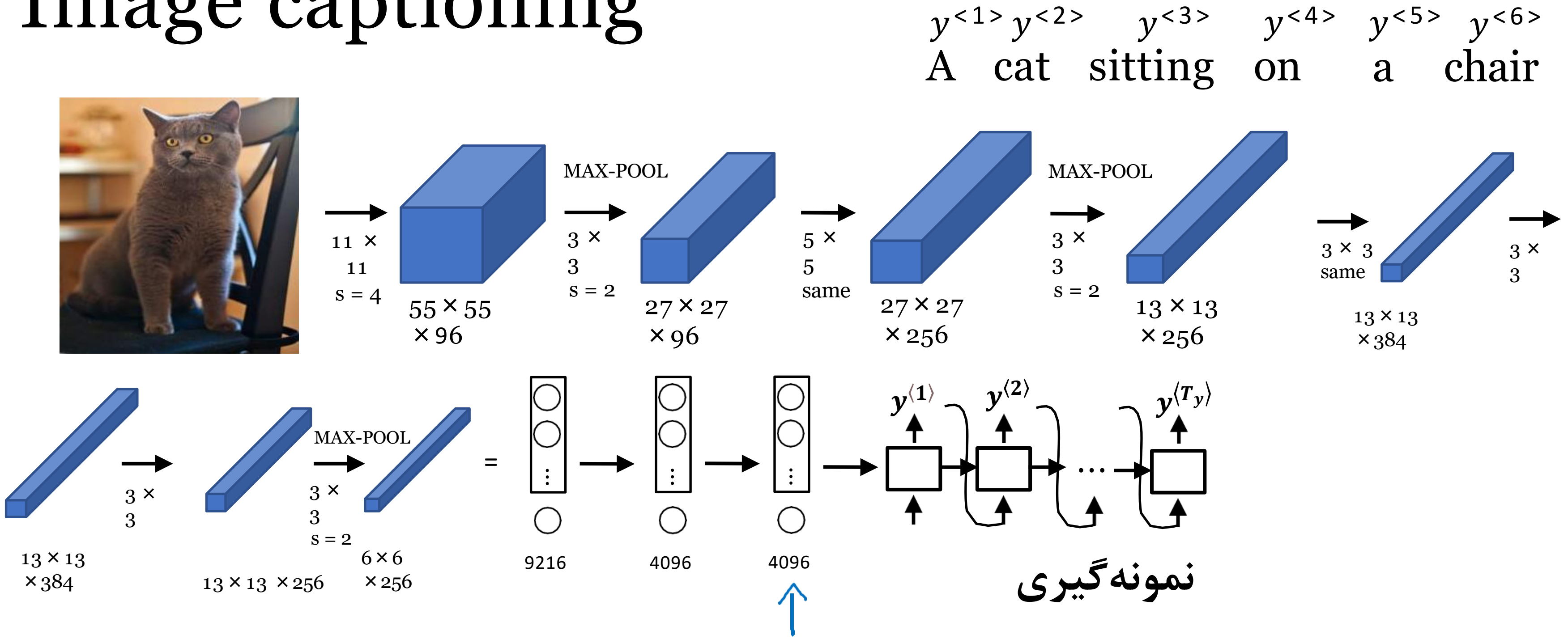
$y^{<1>}$ $y^{<2>}$ $y^{<3>}$ $y^{<4>}$ $y^{<5>}$ $y^{<6>}$

$$\boldsymbol{p}\left(\hat{\boldsymbol{y}}^{\langle 1\rangle}, \hat{\boldsymbol{y}}^{\langle 2\rangle}, \hat{\boldsymbol{y}}^{\langle 3\rangle}, \ldots, \hat{\boldsymbol{y}}^{\langle T_y\rangle} \,\middle|\, \boldsymbol{x}^{\langle 1\rangle}, \boldsymbol{x}^{\langle 2\rangle}, \ldots, \boldsymbol{x}^{\langle T_x\rangle}\right)$$



Conditional language model

Image captioning



Finding the most likely translation

Jane visite l'Afrique en septembre.

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

$$\mathbf{arg\,max\ P}(y^{\langle 1\rangle}, y^{\langle 2\rangle}, \ldots, y^{\langle T_y\rangle} | x)$$

→ Jane is visiting Africa in September.

→ Jane is going to be visiting Africa in September.

→ In September, Jane will visit Africa.

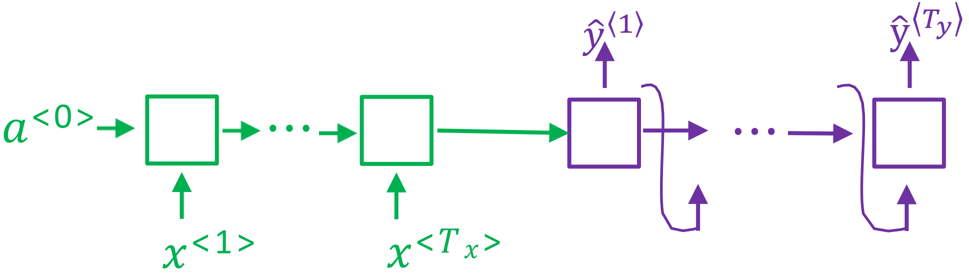
→ Her African friend welcomed Jane in September.

$$\text{arg max} \quad P(\underbrace{y^{<1>}, \ldots, y^{<T_y>}}_{\text{10,000x10,000x ...}} \mid x)$$

$y^{\langle 1\rangle}, \ldots, y^{\langle T_y\rangle}$ 10,000x10,000x ... $(10{,}000)^{T_y}$ Optimal Search

Why not a greedy search?

$P(\hat{y}^{\langle 1 \rangle}|x)$



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

Greedy Search

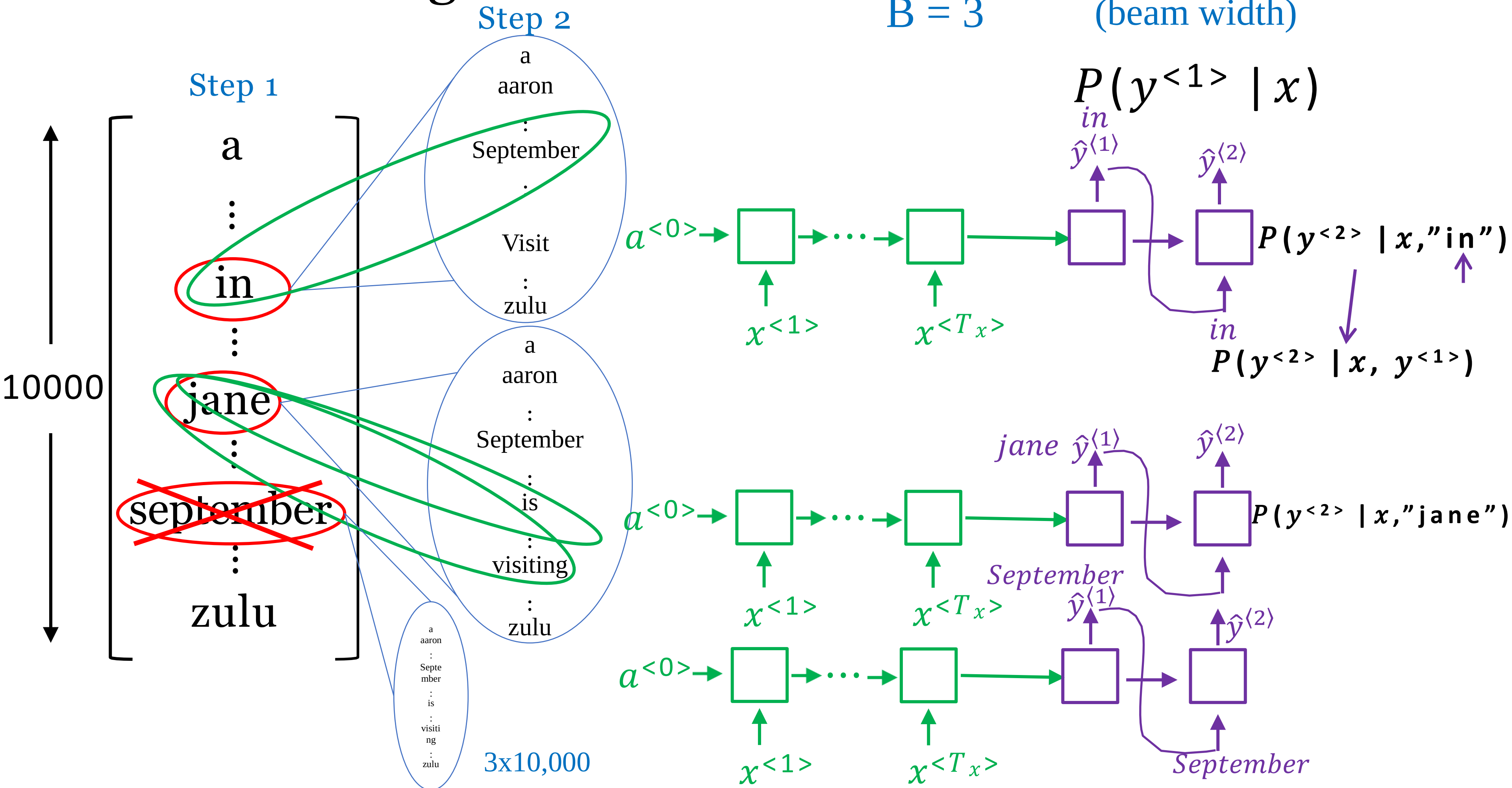
10*10,000

→ Jane is visiting Africa in September.

→ Jane is going to be visiting Africa in September.

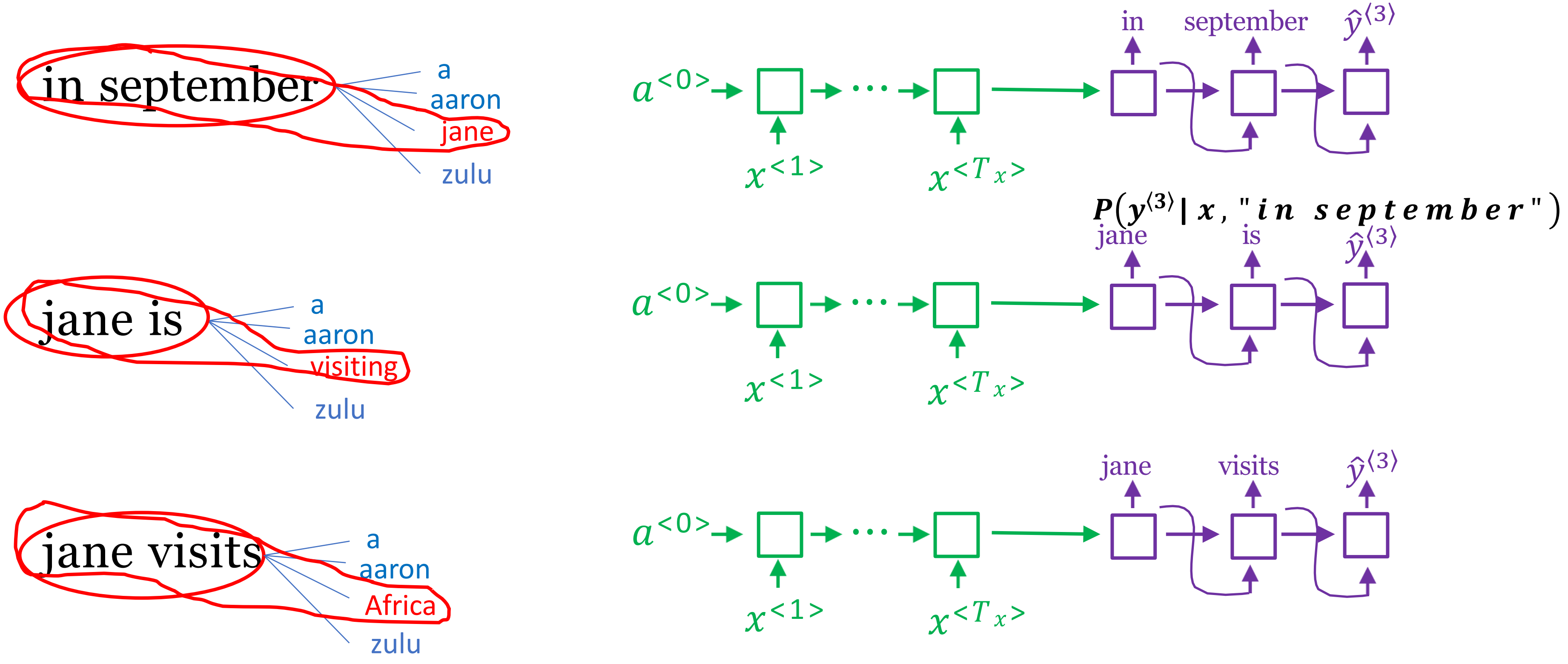
$P(\text{Jane is visiting}|x) < P(\text{Jane is going}|x)$

Beam search algorithm



Beam search ($B = 3$)

B=1 → greedy search



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

jane visits africa in september. <EOS>

Length normalization

- $\arg\max_y \sum_{t=1}^{Ty} \log P\left(y^{(t)} \mid x , y^{(1)}, \ldots y^{(t-1)}\right)$
- $\frac{1}{T_y^{\alpha}} \sum_{t=1}^{Ty} log\, P\left(y^{\langle t \rangle} \middle| x , y^{\langle 1 \rangle}, \ldots y^{\langle t-1 \rangle}\right)$

$\alpha = 0.7, 1, 0 \quad T_y = 1, 2, 3, \cdots, 30$

- **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

$\boxed{\text{B}\uparrow}$

Jane visite l'Afrique en septembre.

Human: Jane visits Africa in September. y^*

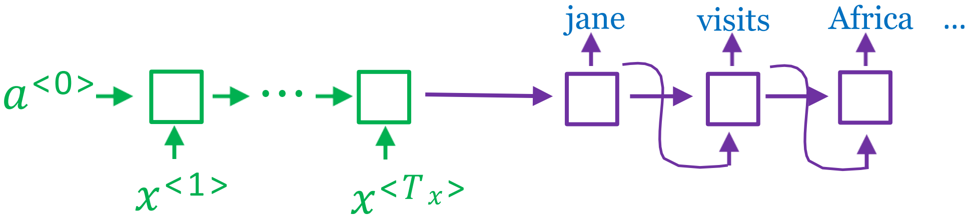
Algorithm: Jane visited Africa last September. $\hat{y}$

RNN computes:

$P(y^*|x) \ \substack{> \\ \leq} \ P(\hat{y}|x)$

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

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.	2×10^{-10}	1×10^{-10}	B
...	...	---	---	R
				B
				R
				R

Figures out what faction 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
- Ref2: there is a cat on the mat

Ref_count=2

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

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

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

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

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

- 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

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

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

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

c: length of the candidate translation
r: effective reference corpus length

Brevity Penalty

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