Advanced Machine Learning
Lecture 9: Recurrent neural networks

Sandjai Bhulai
Vrije Universiteit Amsterdam

s.bhulai@vu.nl
03 October 2023

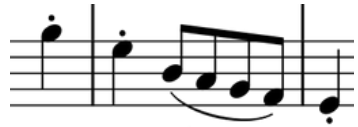
Sequence data

- Speech recognition

→ [illegible]

- Music generation

∅ →



- Sentiment classification

[illegible] →

- DNA sequence analysis

[illegible] → [illegible]

- Machine translation

[illegible] → [illegible]

- Video activity recognition

→ [illegible]

- Name entity recognition

[illegible] → [illegible]

VU

Name entity recognition

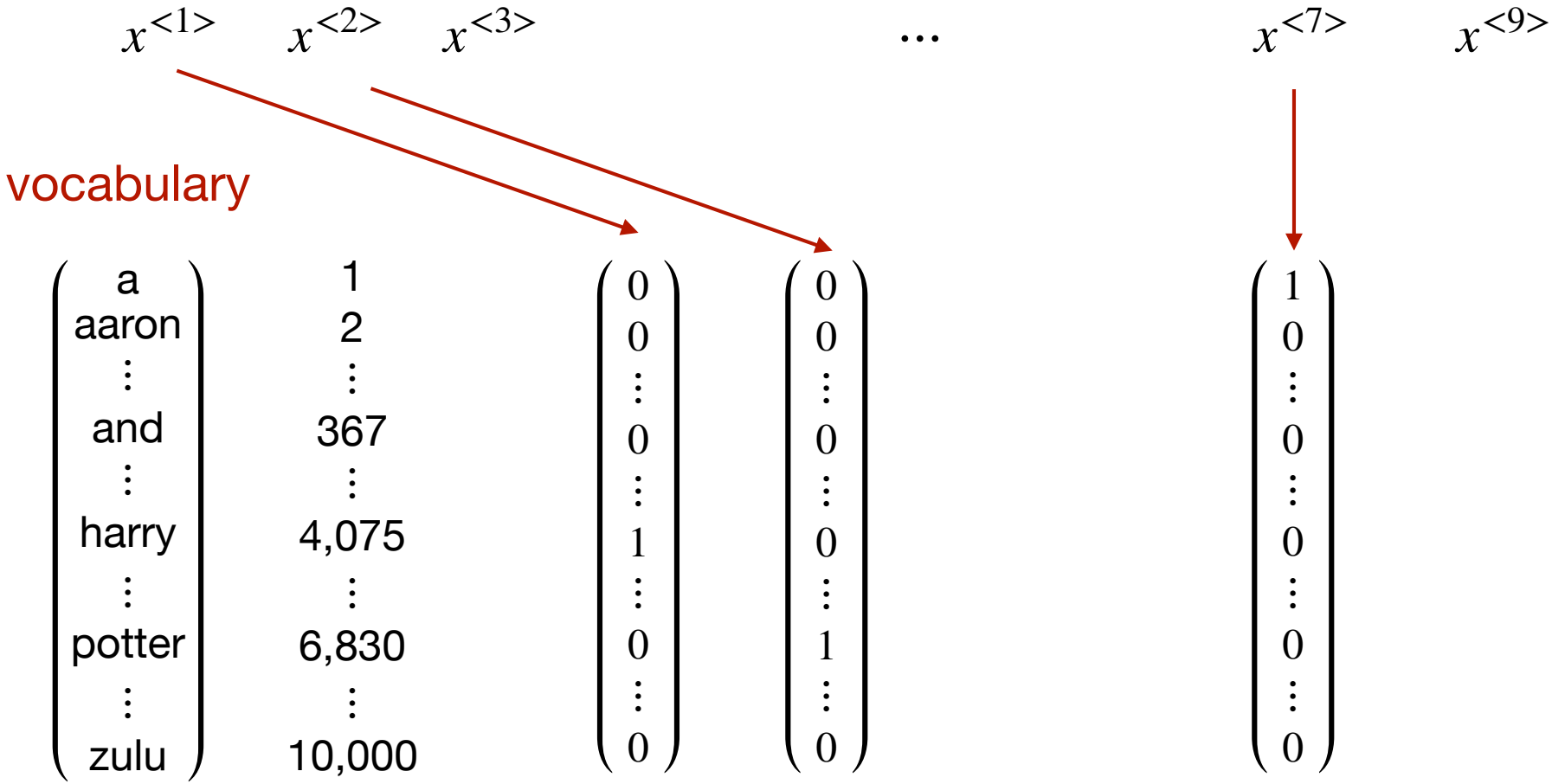
- x: Harry Potter and Hermione Granger invented a new spell.

$x^{<1>}$	$x^{<2>}$	$x^{<3>}$		…				$x^{<9>}$
1	1	0	1	1	0	0	0	0
$y^{<1>}$	$y^{<2>}$	$y^{<3>}$		…				$y^{<9>}$

- Length $T_x = 9$
- Length $T_y = 9$

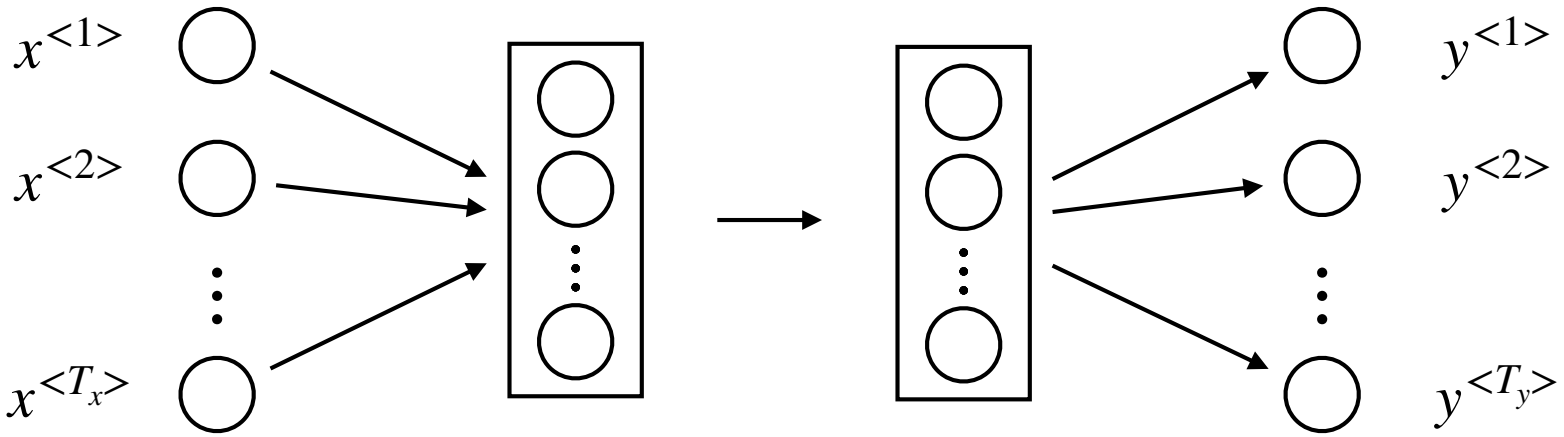
Name entity recognition

- x: Harry Potter and Hermione Granger invented a new spell.

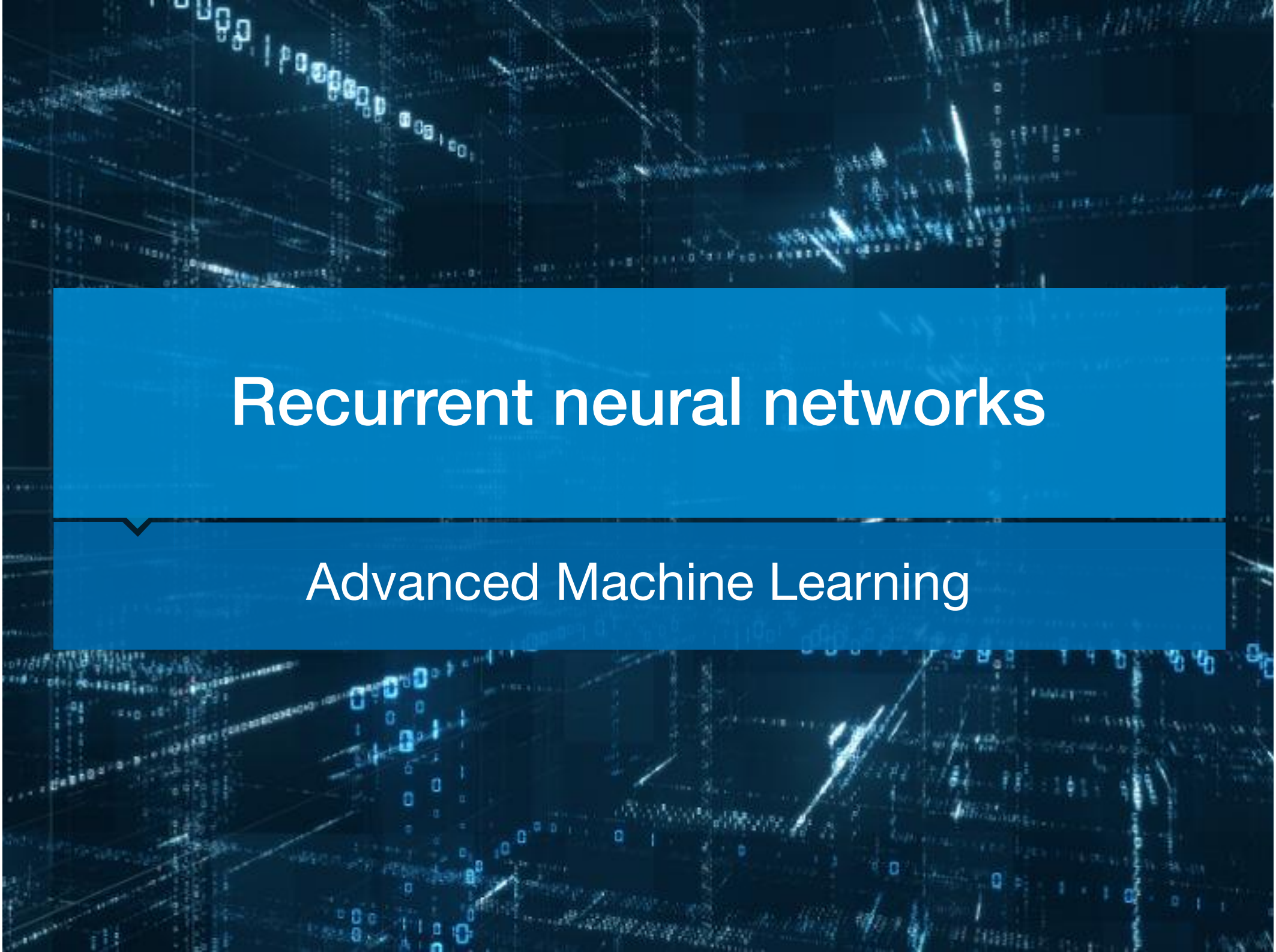


Name entity recognition

- Use a standard neural network



- Inputs and outputs can be of different lengths in different examples
- Does not share features learned across different positions of text

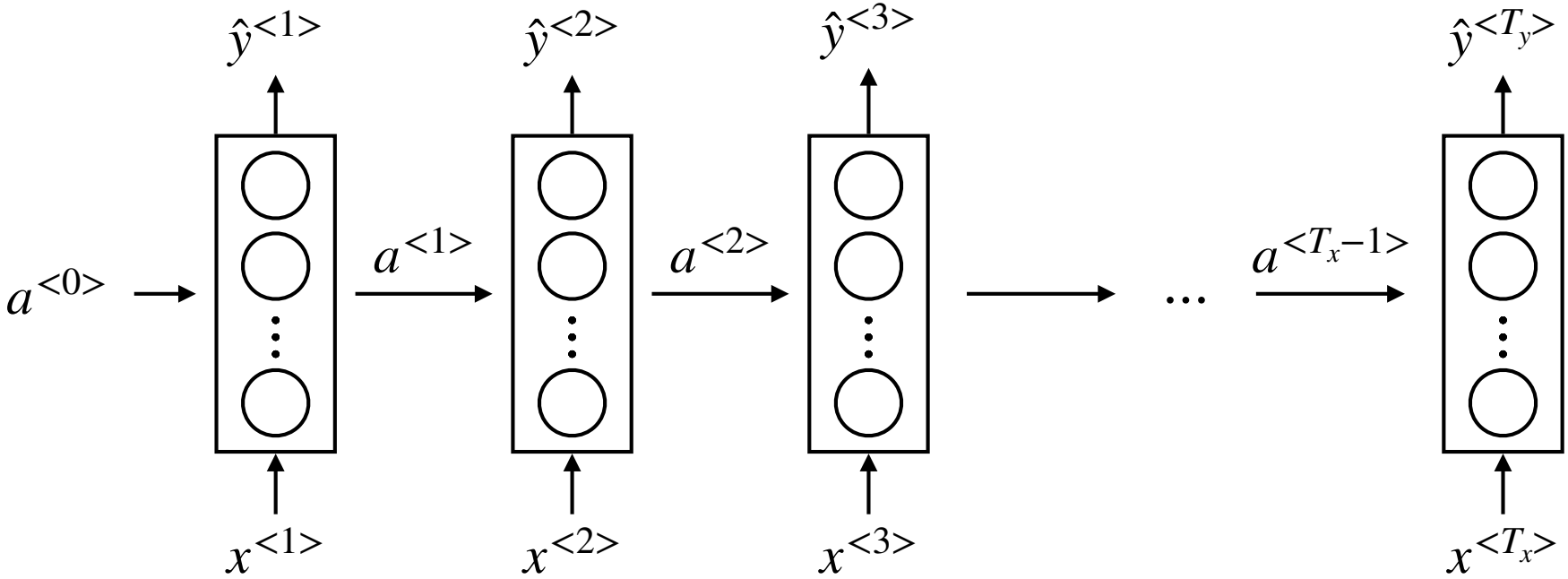


Recurrent neural networks

Advanced Machine Learning

Recurrent neural networks

- Use a recurrent neural network



- Forward propagation

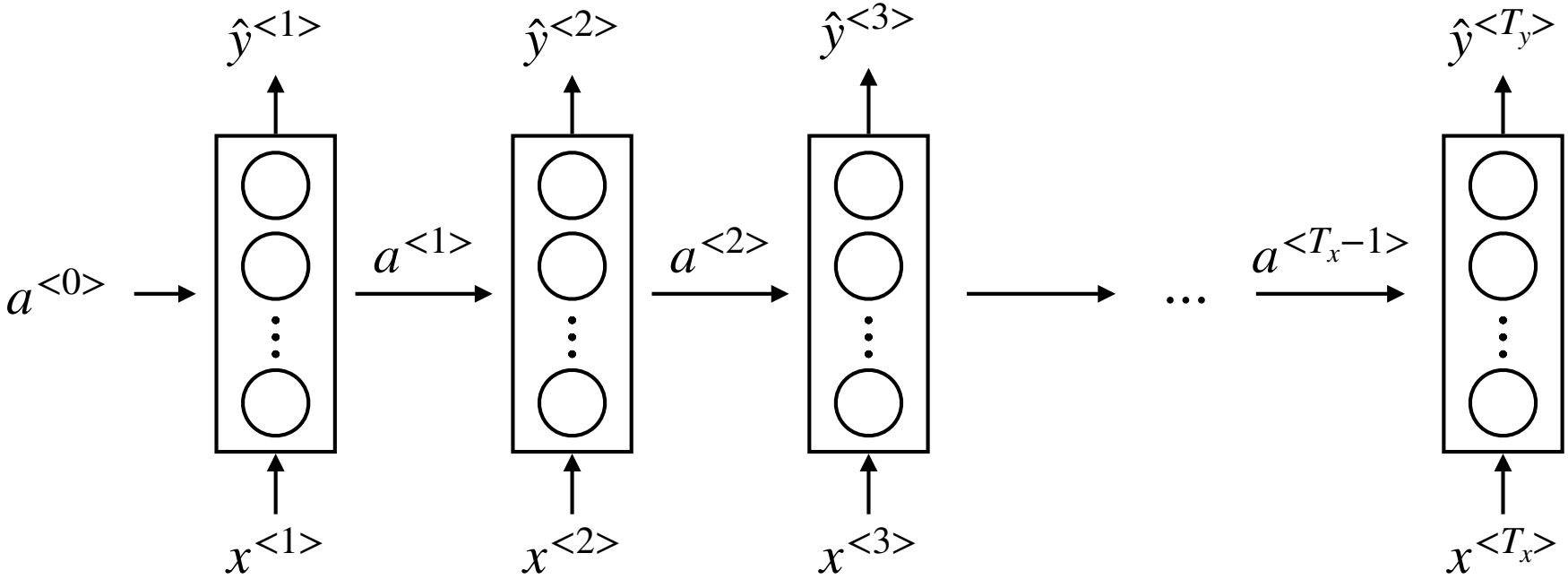
$$a^{<t>} = g(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$$

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

VU

Recurrent neural networks

- Use a recurrent neural network



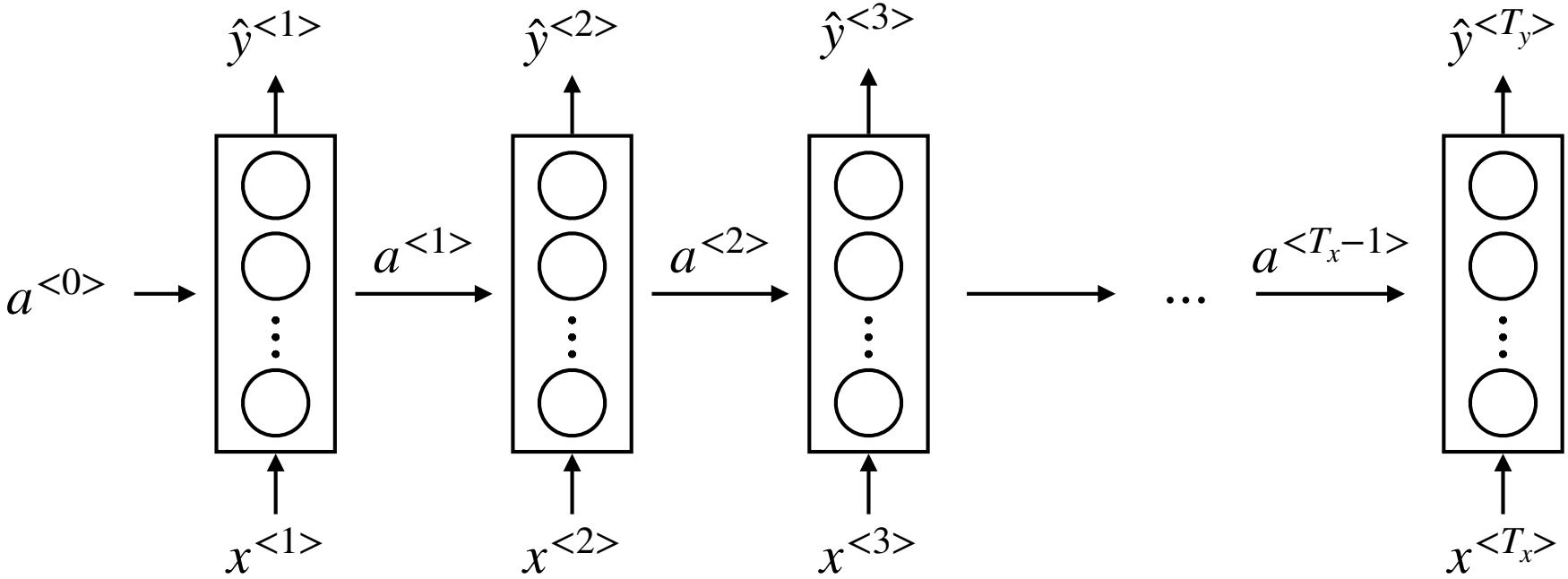
- Forward propagation

$$a^{<t>} = g([W_{aa} \;\; W_{ax}][a^{<t-1>}, x^{<t>}] + b_a)$$

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

Recurrent neural networks

- Use a recurrent neural network



- Forward propagation

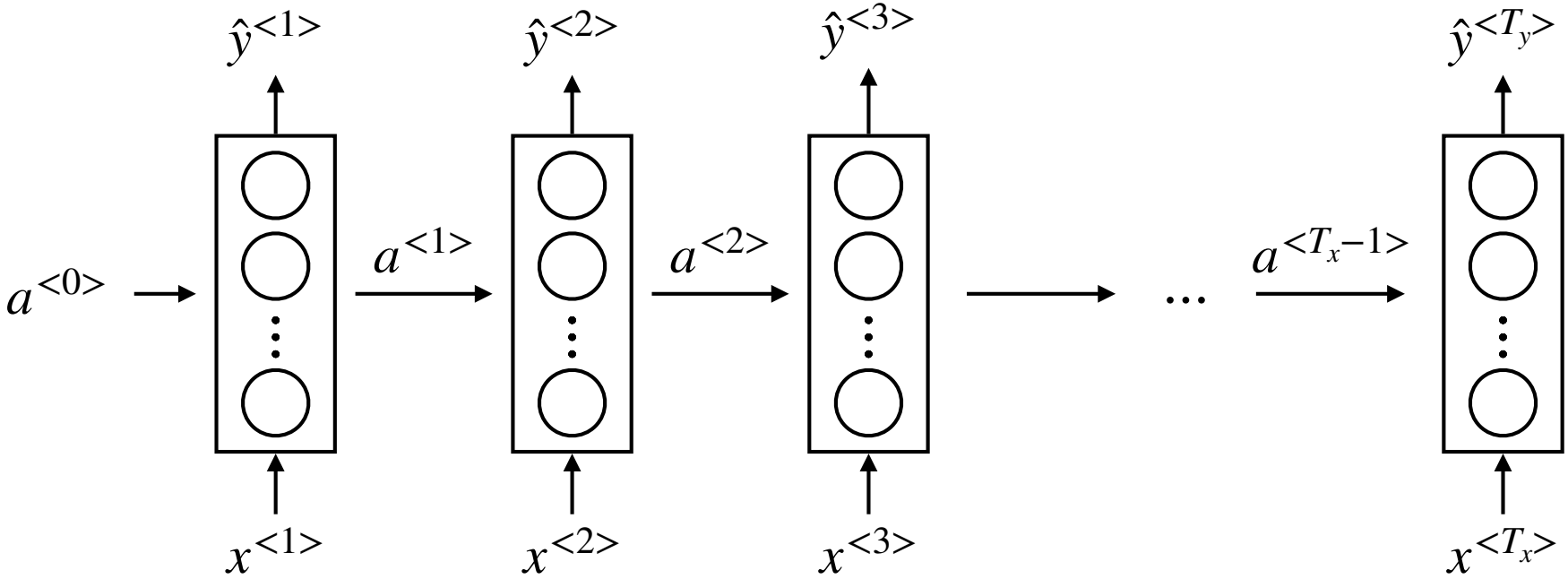
$$a^{<t>} = g(W_a[a^{<t-1>}, x^{<t>}] + b_a)$$

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

VU

Recurrent neural networks

- Use a recurrent neural network



- Loss function: $\mathcal{L}^{<t>}(\hat{y}^{<t>}, y^{<t>}) = -y^{<t>}\log\hat{y}^{<t>} - (1-y^{<t>})\log(1-\hat{y}^{<t>})$

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

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

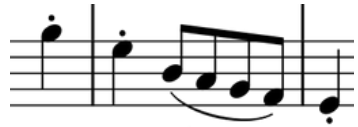
Recurrent neural networks

- 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

AVPPRWHPAQPPVAVT

→ AVPPRWHPAQPPVAVT

- Machine translation

"Voulez-vous chanter avec moi?"

→ "Do you want to sing with me?"

- Video activity recognition

→ Running

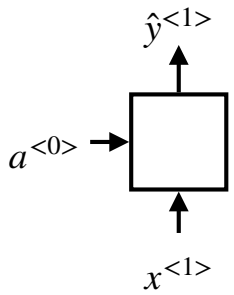
- Name entity recognition

Yesterday, Harry Potter met Hermione Granger.

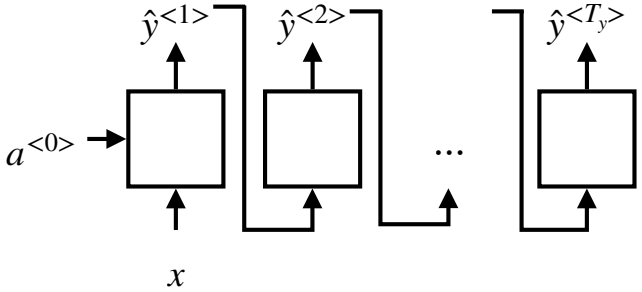
→ Yesterday, Harry Potter met Hermione Granger.

VU

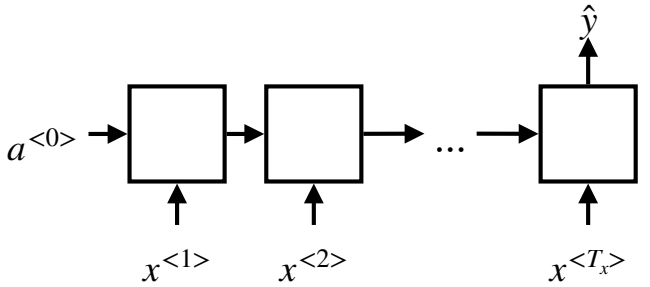
Recurrent neural networks



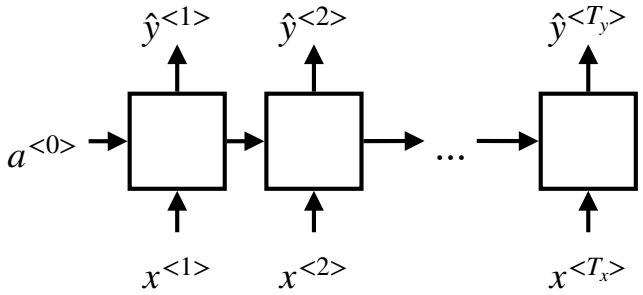
one to one



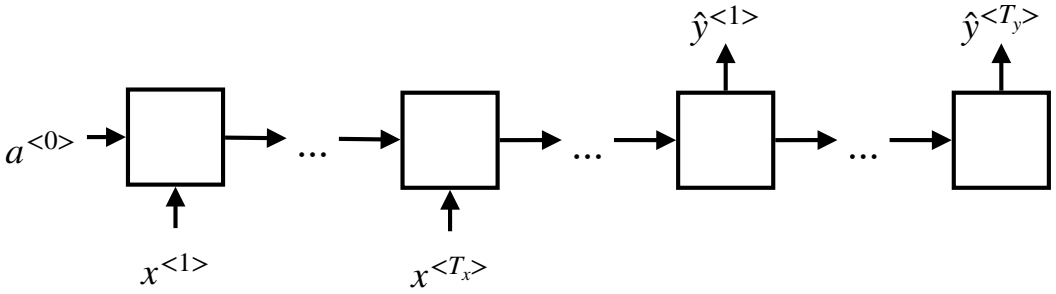
one to many



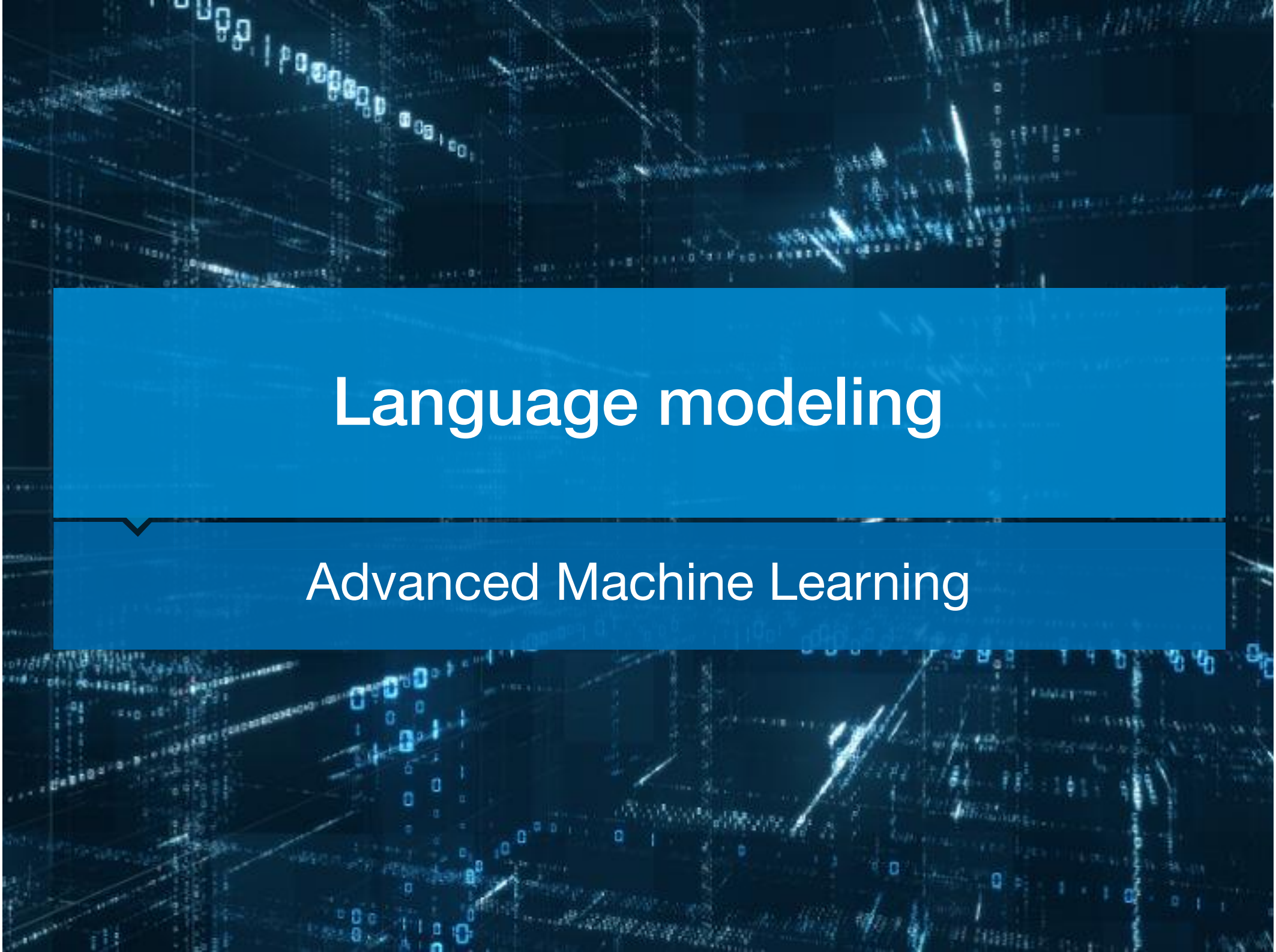
many to one



many to many



many to many



Language modeling

Advanced Machine Learning

Language modeling

- Speech recognition
  - The apple and pair salad
  - The apple and pear salad

- Language model
  - $\mathbb{P}$(The apple and pair salad) = 3.2×10^{-13}
  - $\mathbb{P}$(The apple and pear salad) = $5.7 = 10^{-10}$

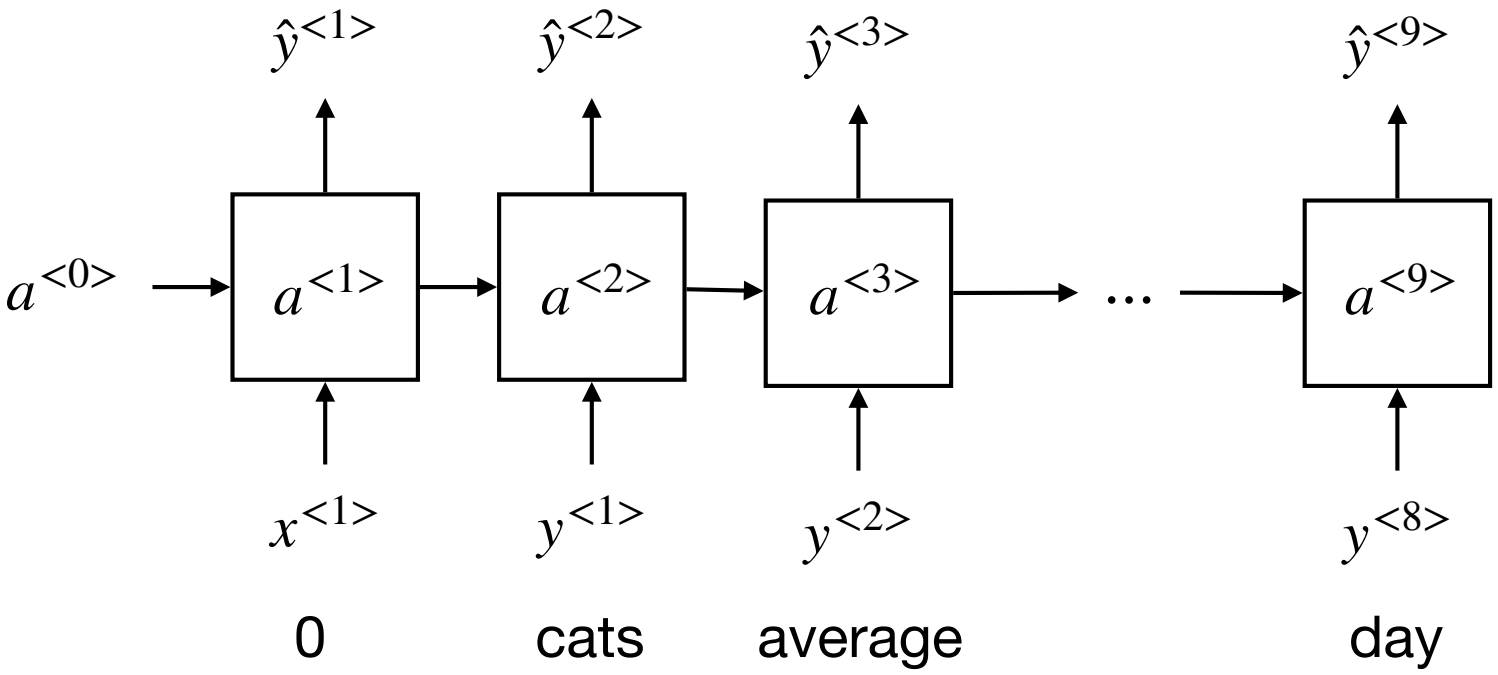
Language modeling

- Training set: large corpus of English text

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

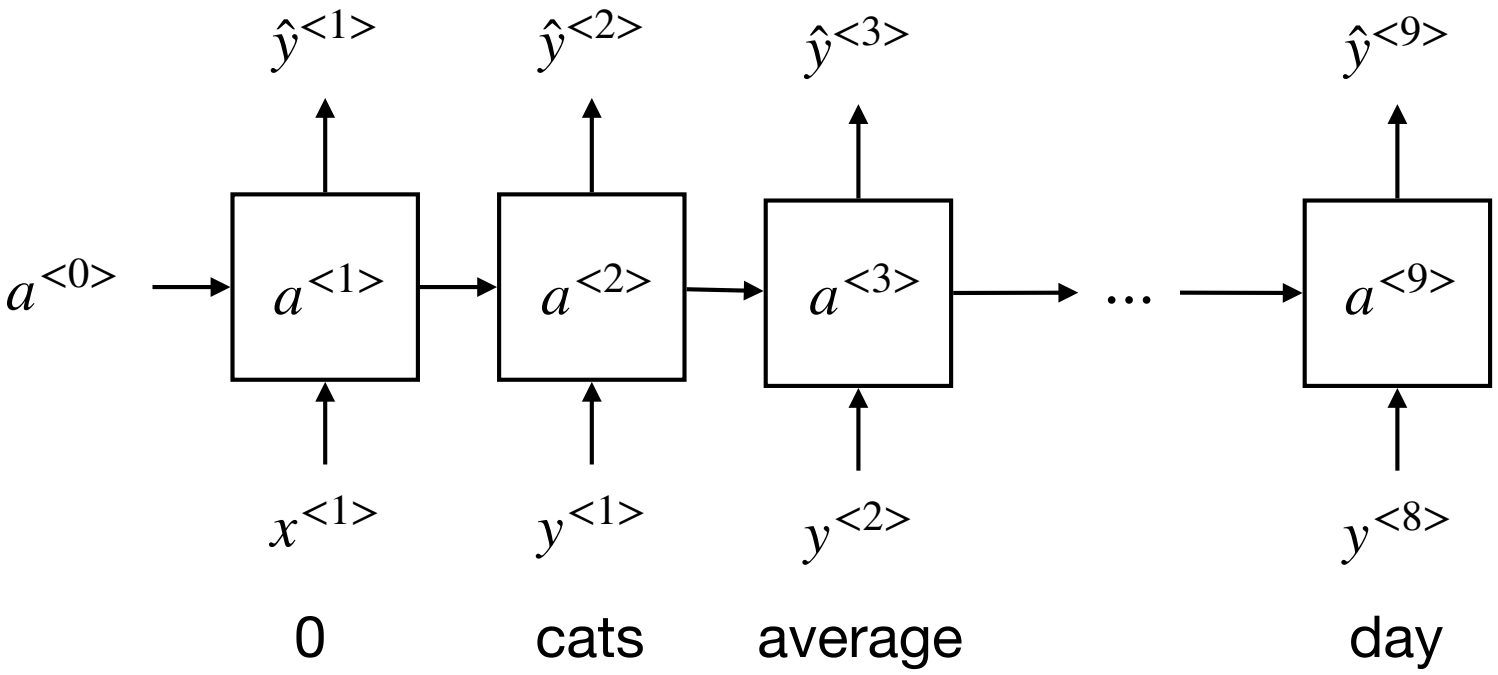
- The Egyptian Mau is a breed of cat. <EOS>
- The Egyptian <UNK> is a breed of cat. <EOS>

Language modeling



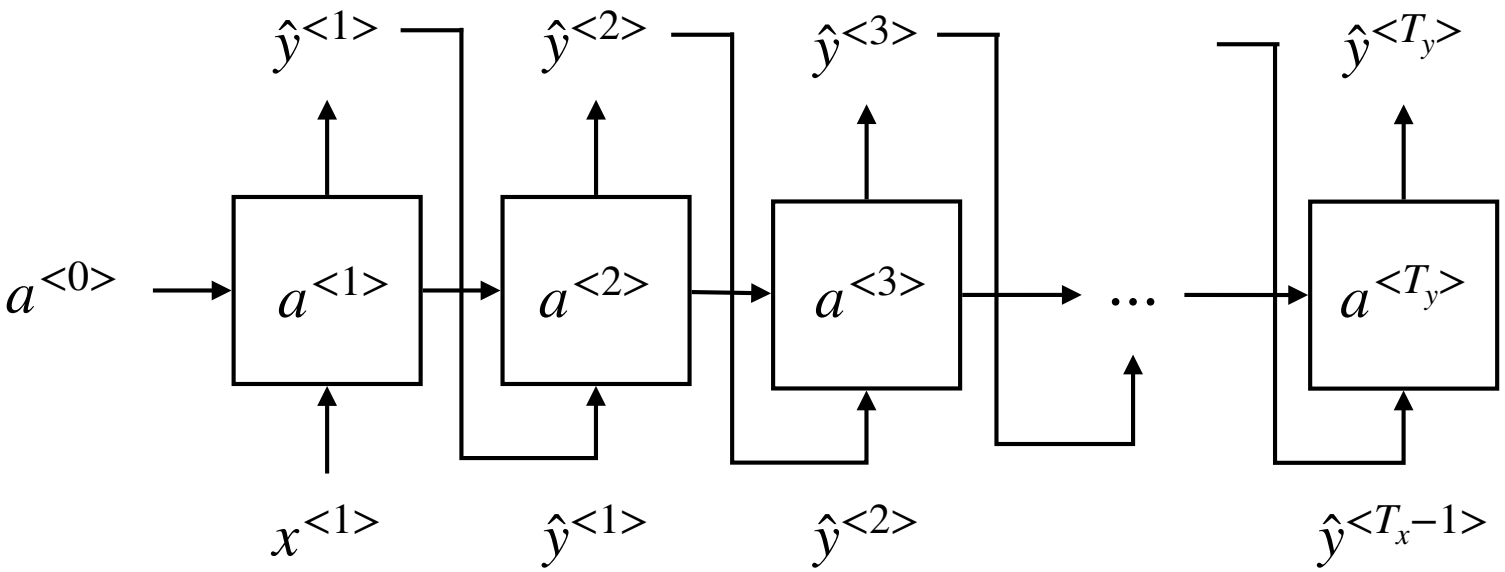
- Cats average 15 hours of sleep a day. <EOS>

Language modeling



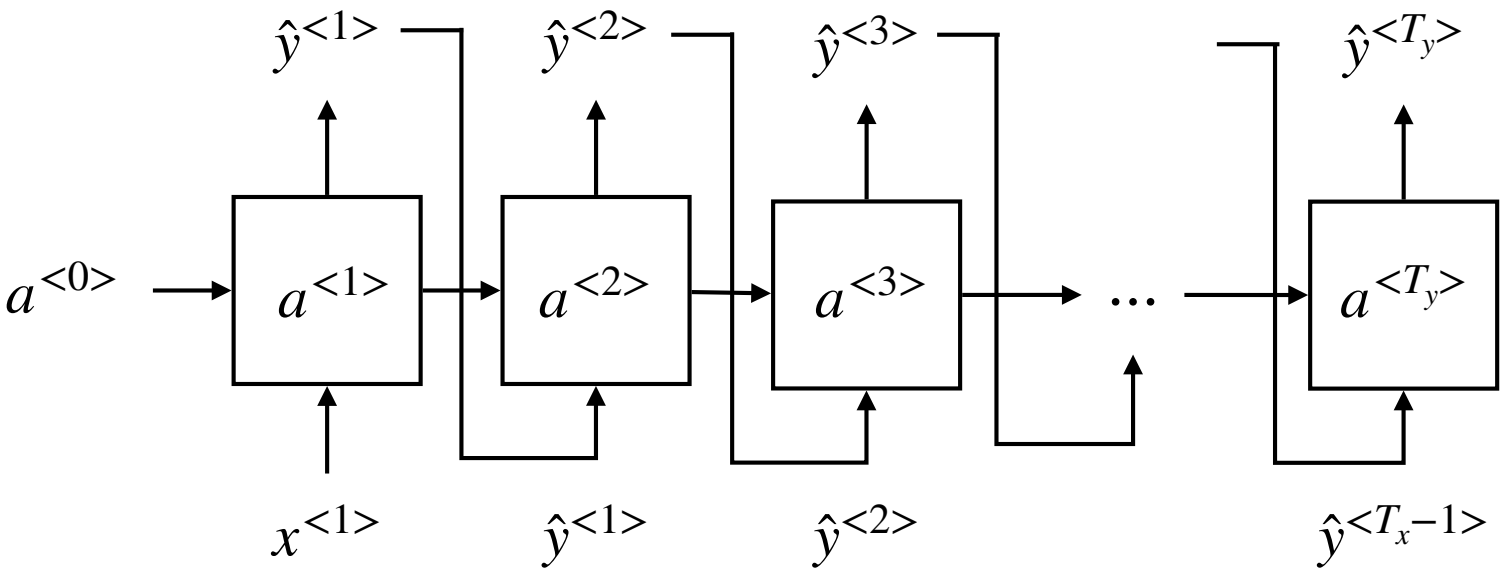
- Cats average 15 hours of sleep a day. <EOS>
- $\mathbb{P}(y^{<1>}, y^{<2>}, y^{<3>}) = \mathbb{P}(y^{<1>})\mathbb{P}(y^{<2>} \mid y^{<1>})\mathbb{P}(y^{<3>} \mid y^{<1>}, y^{<2>})$

Language modeling



- Sampling a sentence
- Remove <UNK> terms

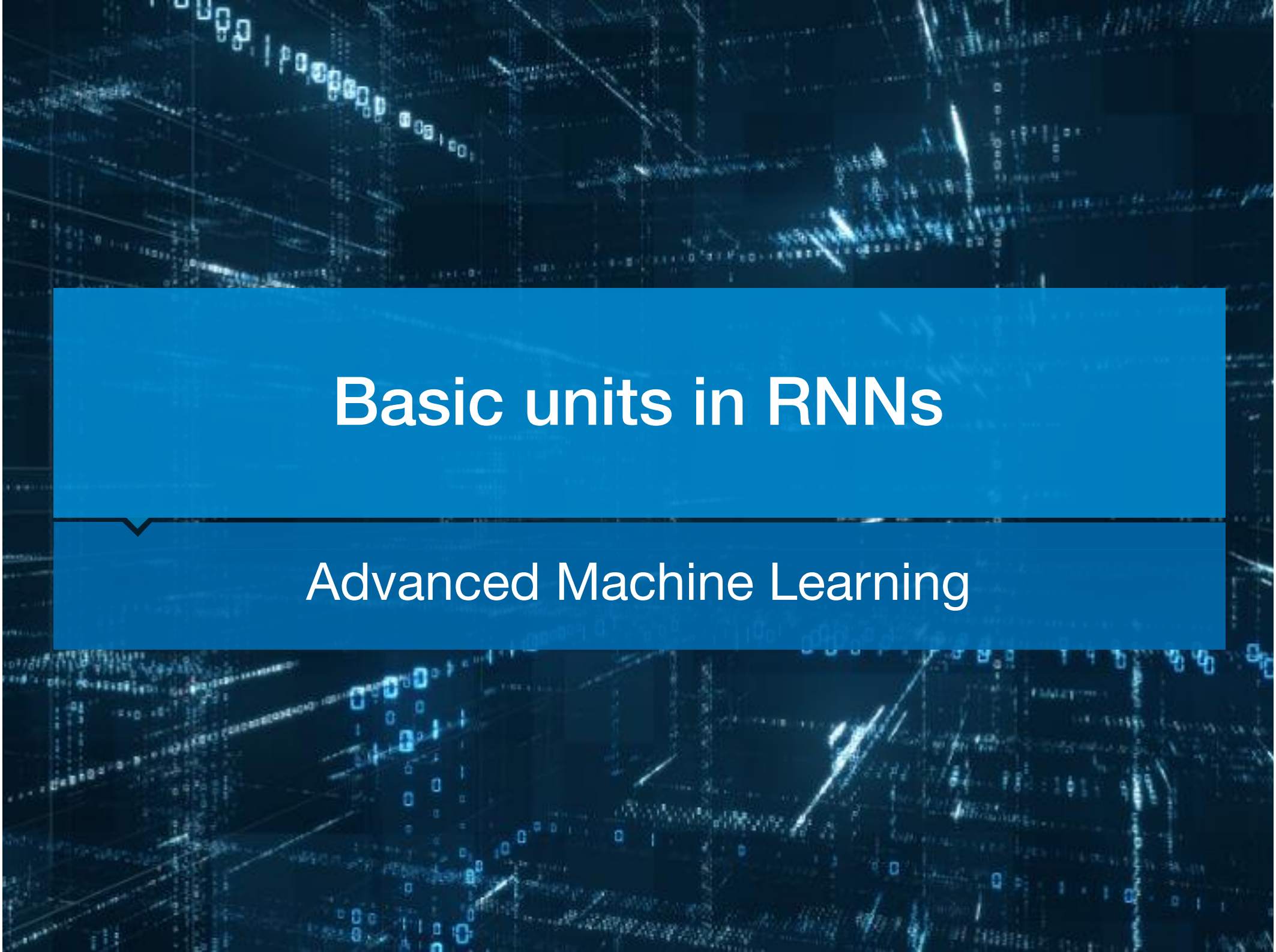
Language modeling



- Sentence model vocabulary:
  [a, aaron, …, zulu, <UNK>]
- Character model vocabulary:
  [a, b, c, …, z, , ., ,, ;, 0, …, 9, A, …, Z]

Language modeling

News	Shakespeare
President enrique peña nieto, announced sench's sulk former coming football langston paring.	The mortal moon hath her eclipse in love.
"I was not at all surprised," said hich langston.	And subject of this thou art another this fold.
"Concussion epidemic", to be examined.	When besser be my love to me see sabl's.
The gray football the told some and this has on the uefa icon, should money as.	For whose are ruse of mine eyes heaves.

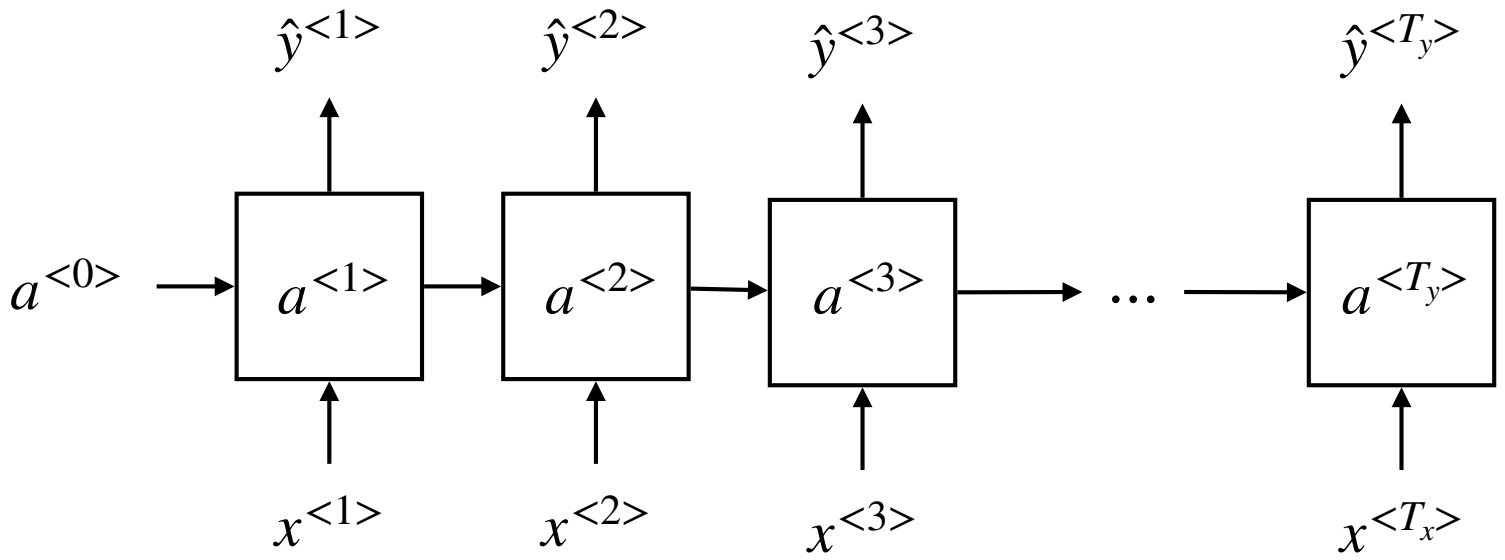


Basic units in RNNs

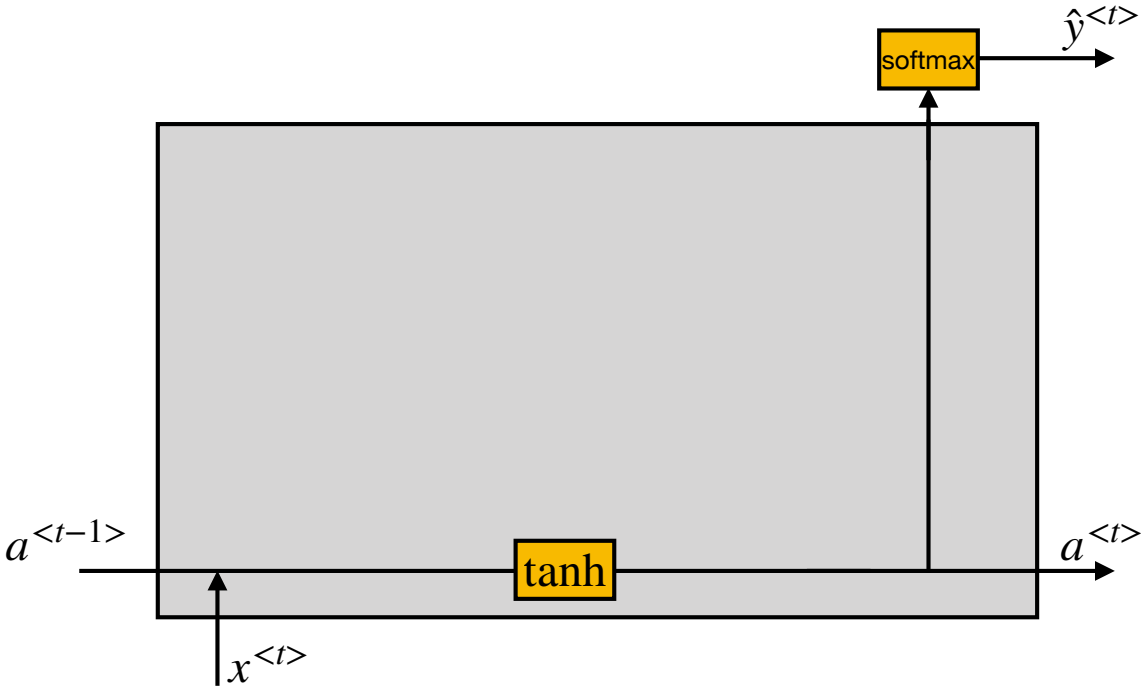
Advanced Machine Learning

Vanishing gradients

- The cat, which already ate …, was full
- The cats, which already ate …, were full



RNN unit



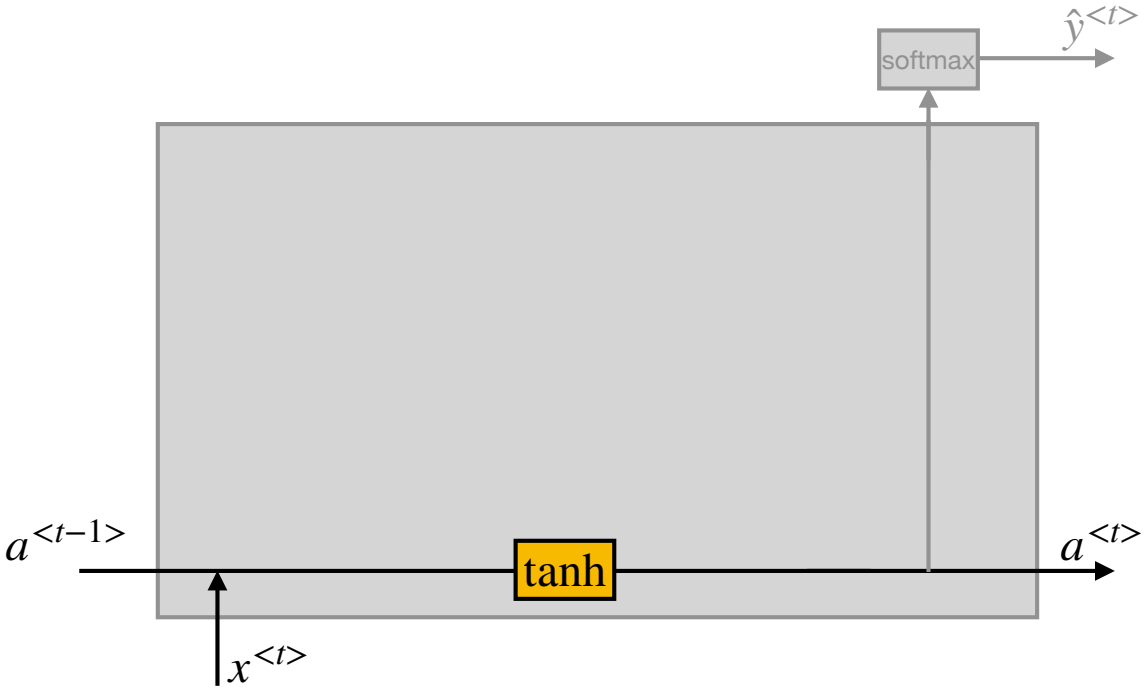
$$a^{<t>} = \tanh(W_a[a^{<t-1>}, x^{<t>}] + b_a)$$

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

RNN unit

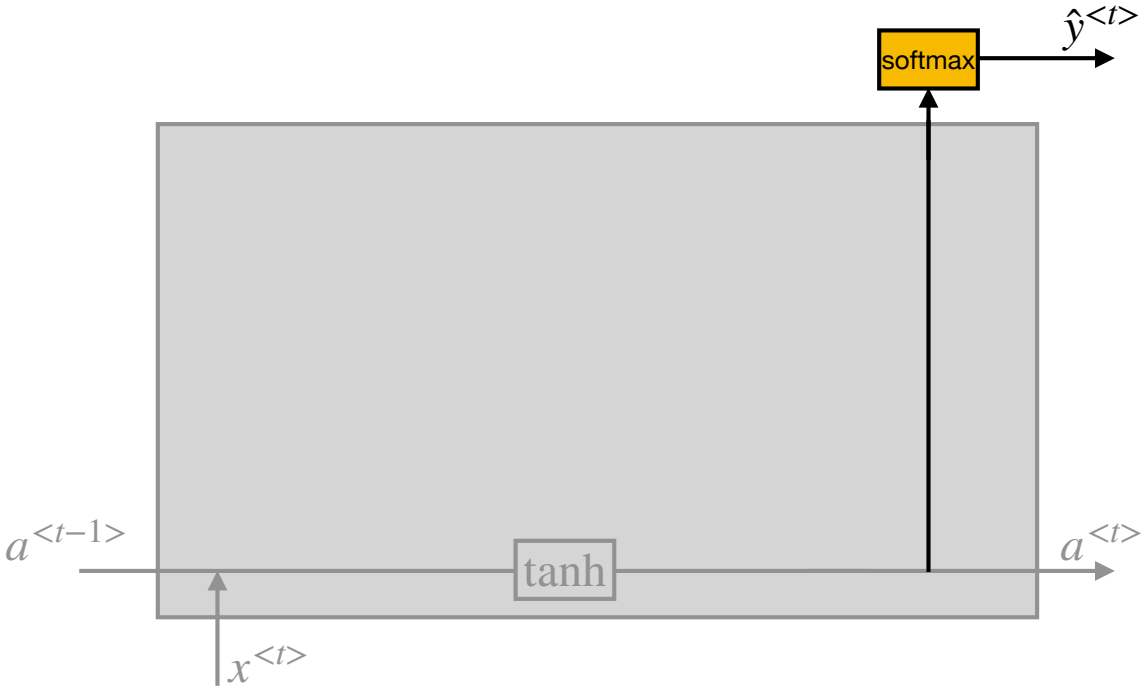
$$a^{<t>} = \tanh(W_a[a^{<t-1>}, x^{<t>}] + b_a)$$

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



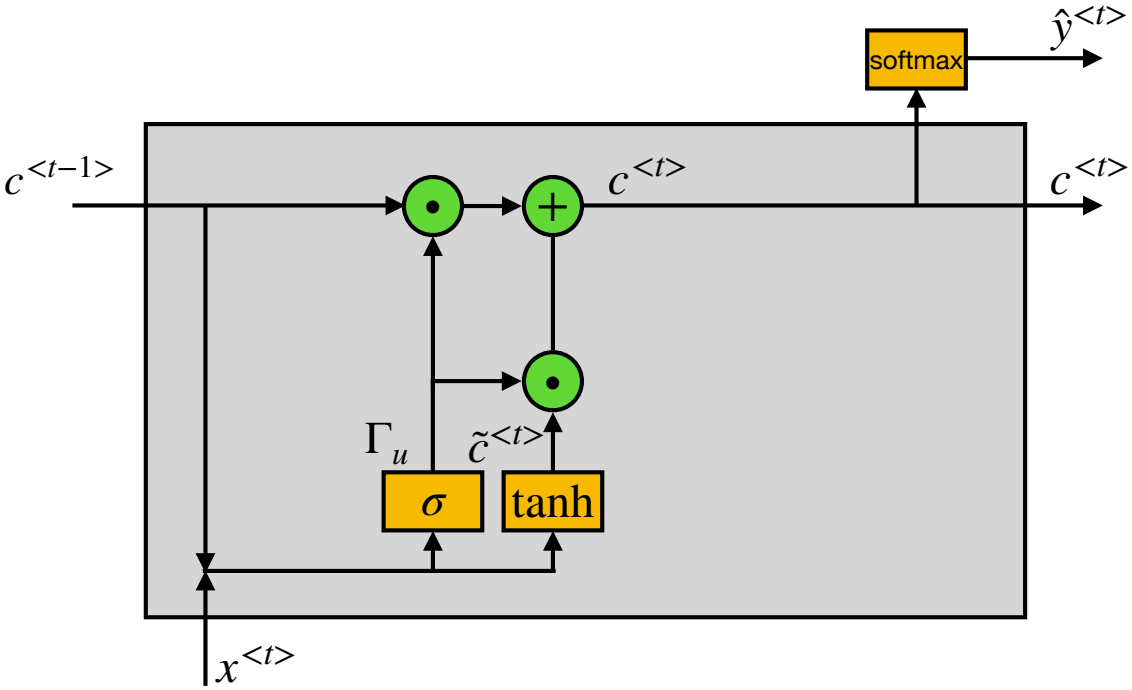
RNN unit

$$a^{<t>} = \tanh(W_a[a^{<t-1>}, x^{<t>}] + b_a)$$



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

Gated recurrent unit (GRU)



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

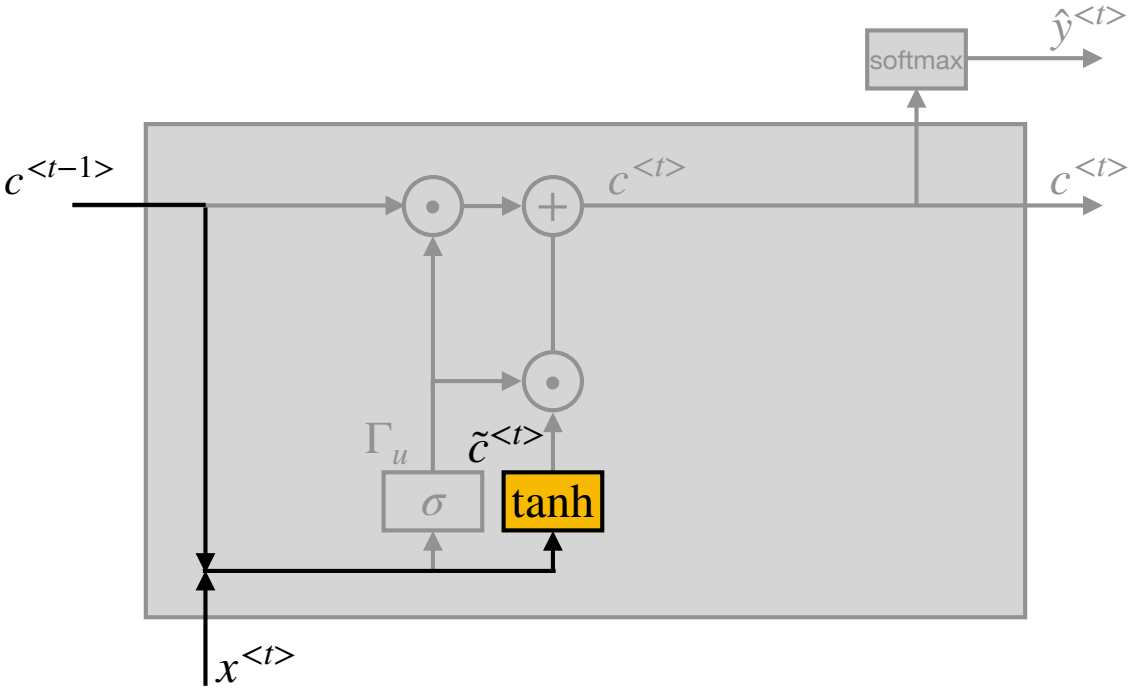
$$\Gamma_u = \sigma(W_u[c^{<t-1>}, x^{<t>}] + b_u)$$

$$c^{<t>} = \Gamma_u \cdot \tilde{c}^{<t>} + (1 - \Gamma_u) \cdot c^{<t-1>}$$

$$\hat{y}^{<t>} = g(W_y c^{<t>} + b_y)$$

The cat, which already ate …, was full

Gated recurrent unit (GRU)



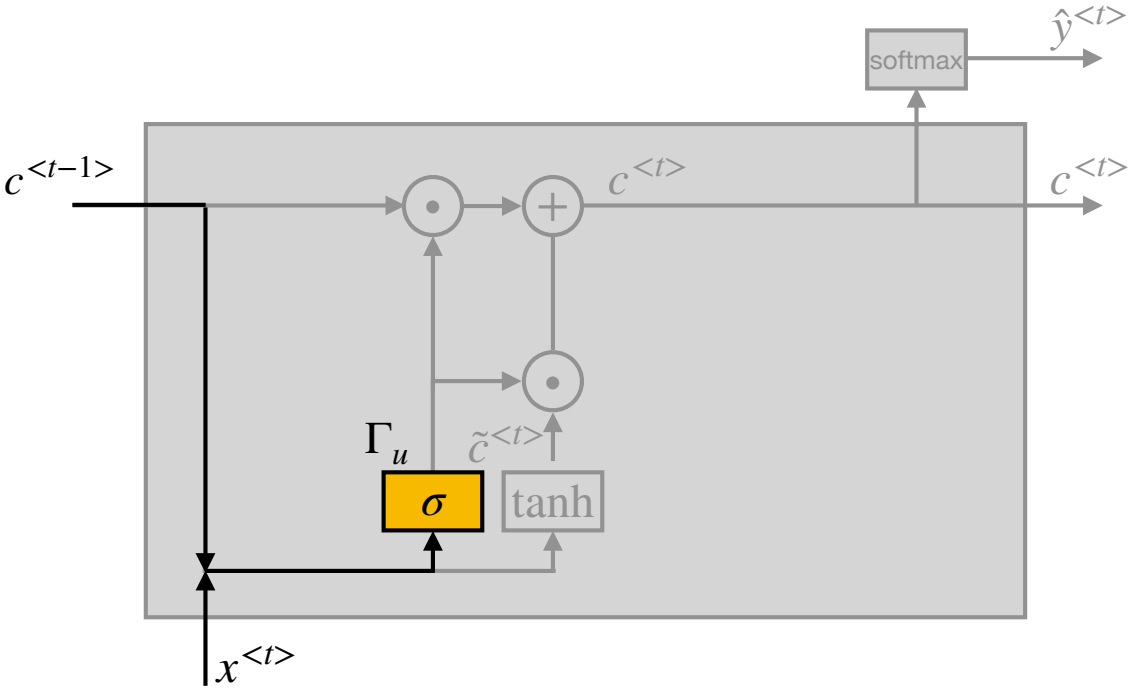
$$\tilde{c}^{<t>} = \tanh(W_c[c^{<t-1>}, x^{<t>}] + b_c)$$

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

$$c^{<t>} = \Gamma_u \cdot \tilde{c}^{<t>} + (1 - \Gamma_u) \cdot c^{<t-1>}$$

$$\hat{y}^{<t>} = g(W_y c^{<t>} + b_y)$$

Gated recurrent unit (GRU)



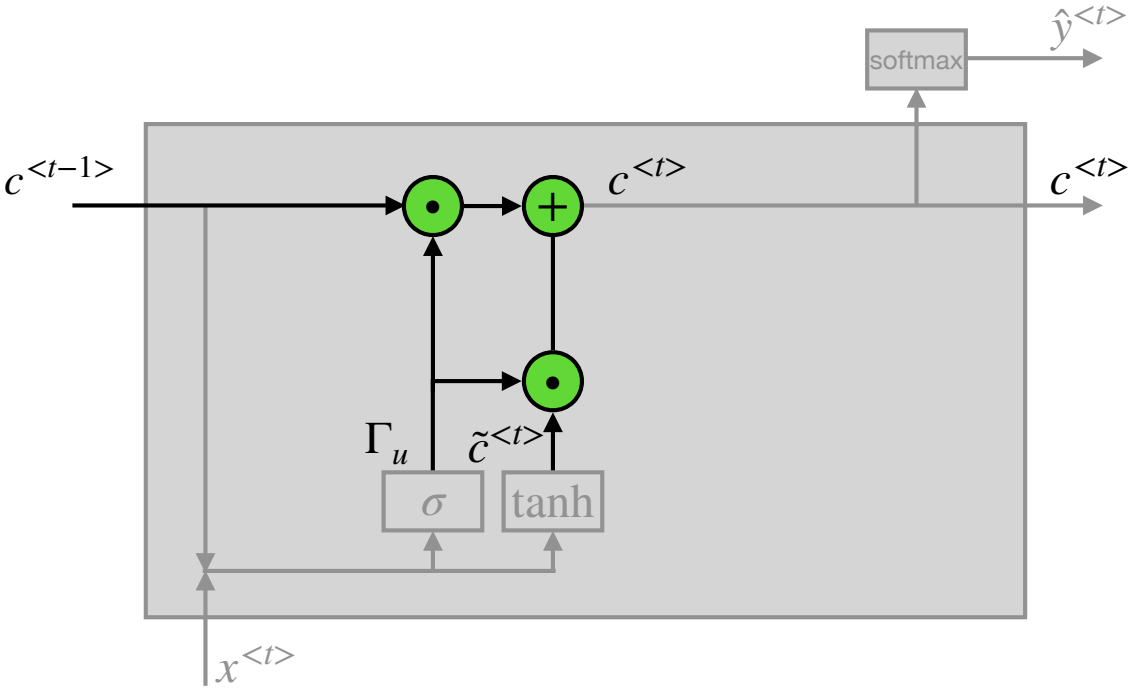
$$\tilde{c}^{<t>} = \tanh(W_c[c^{<t-1>}, x^{<t>}] + b_c)$$

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

$$c^{<t>} = \Gamma_u \cdot \tilde{c}^{<t>} + (1 - \Gamma_u) \cdot c^{<t-1>}$$

$$\hat{y}^{<t>} = g(W_y c^{<t>} + b_y)$$

Gated recurrent unit (GRU)



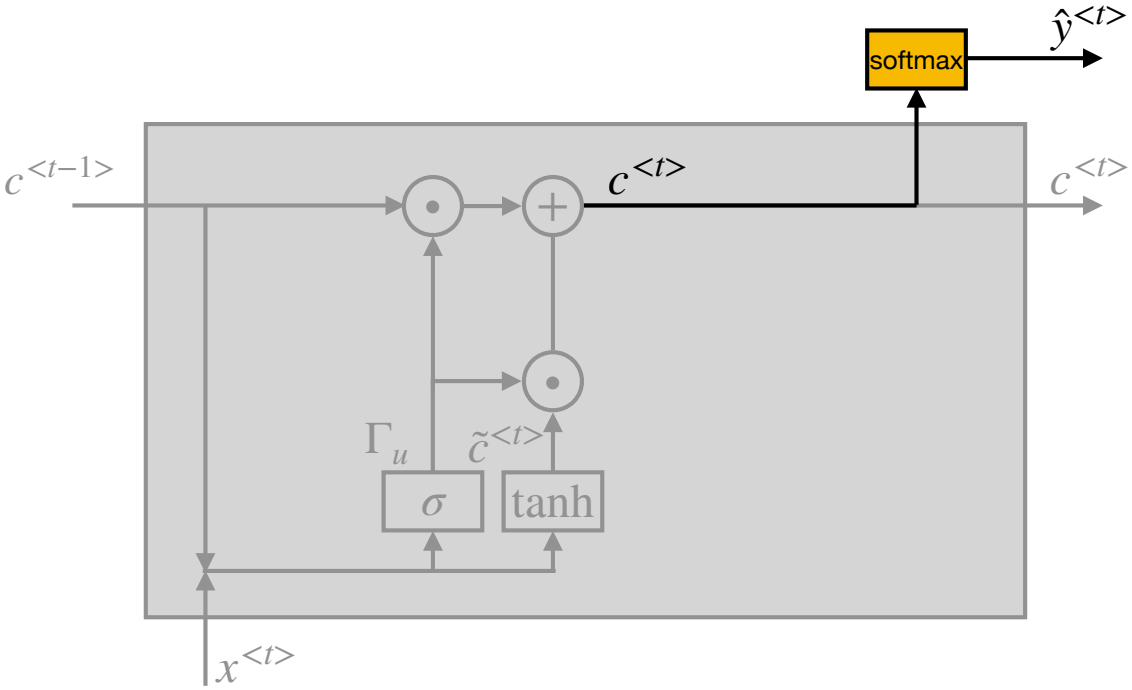
$$\tilde{c}^{<t>} = \tanh(W_c[c^{<t-1>}, x^{<t>}] + b_c)$$

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

$$c^{<t>} = \Gamma_u \cdot \tilde{c}^{<t>} + (1 - \Gamma_u) \cdot c^{<t-1>}$$

$$\hat{y}^{<t>} = g(W_y c^{<t>} + b_y)$$

Gated recurrent unit (GRU)



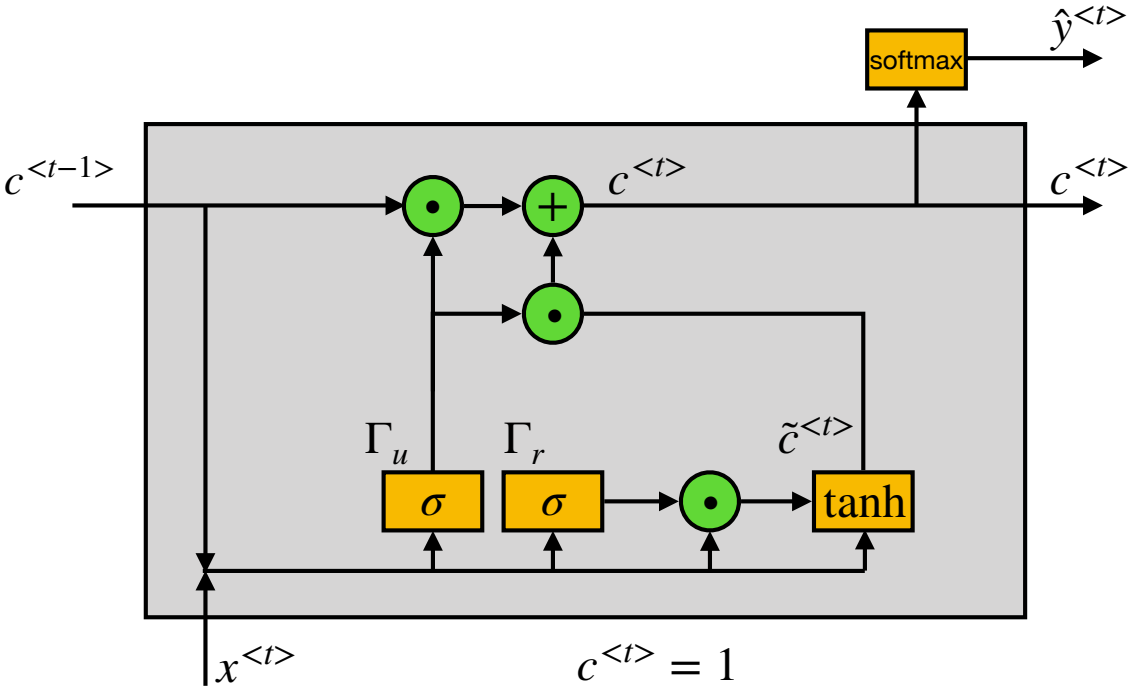
$$\tilde{c}^{<t>} = \tanh(W_c[c^{<t-1>}, x^{<t>}] + b_c)$$

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

$$c^{<t>} = \Gamma_u \cdot \tilde{c}^{<t>} + (1 - \Gamma_u) \cdot c^{<t-1>}$$

$$\hat{y}^{<t>} = g(W_y c^{<t>} + b_y)$$

Fully-gated recurrent unit



$$\tilde{c}^{<t>} = \tanh(W_c[\Gamma_r \cdot c^{<t-1>}, x^{<t>}] + b_c)$$

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

$$\Gamma_r = \sigma(W_r[c^{<t-1>}, x^{<t>}] + b_r)$$

$$c^{<t>} = \Gamma_u \cdot \tilde{c}^{<t>} + (1 - \Gamma_u) \cdot c^{<t-1>}$$

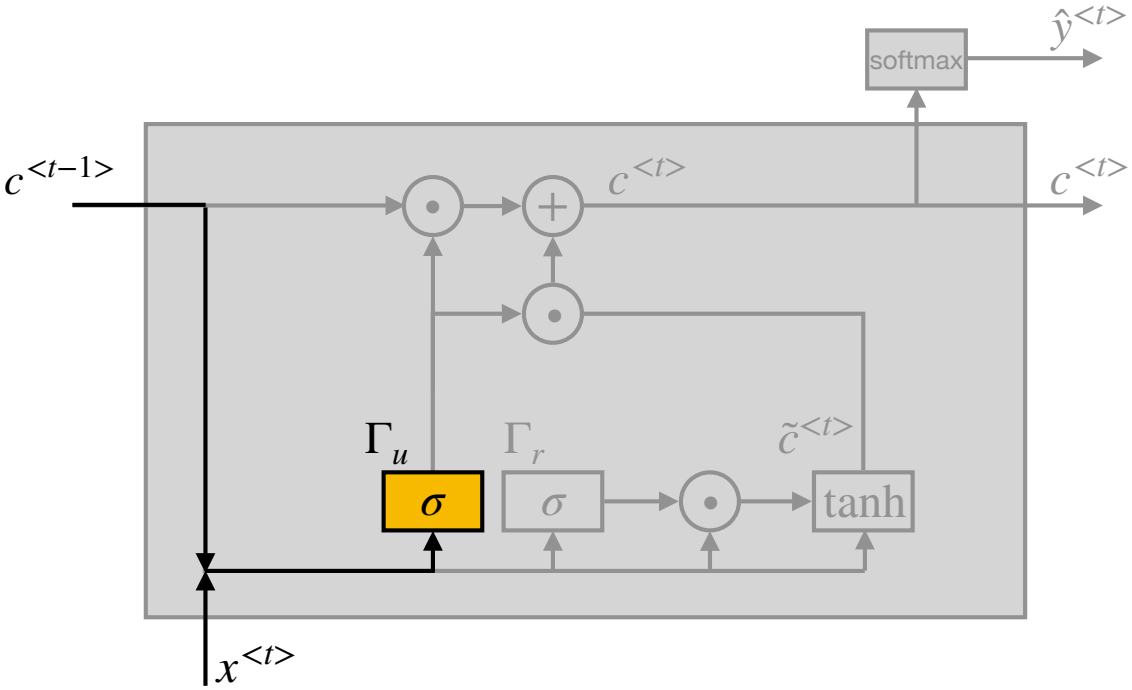
$$\hat{y}^{<t>} = g(W_y c^{<t>} + b_y)$$

$c^{<t>} = 1$

$\Gamma_u = 1 \quad \Gamma_u = 0 \quad \Gamma_u = 0 \quad \Gamma_u = 0 \quad \Gamma_u = 1$

The cat, which already ate …, was full

Fully-gated recurrent unit



$$\tilde{c}^{<t>} = \tanh(W_c[\Gamma_r \cdot c^{<t-1>}, x^{<t>}] + b_c)$$

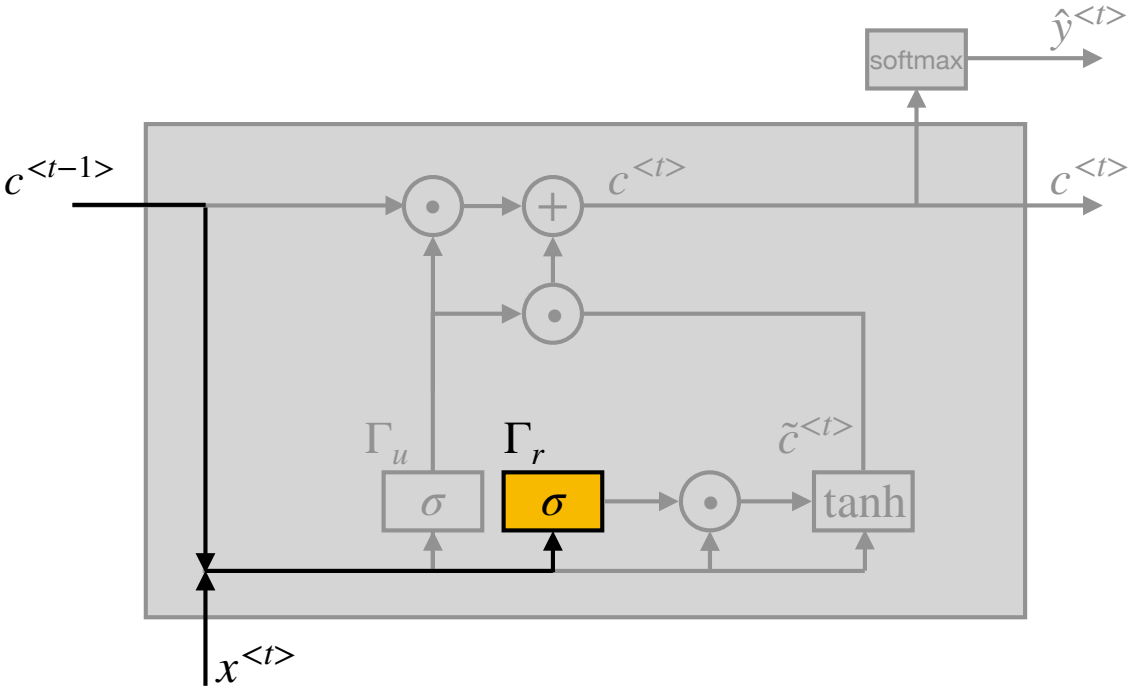
$$\Gamma_u = \sigma(W_u[c^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_r = \sigma(W_r[c^{<t-1>}, x^{<t>}] + b_r)$$

$$c^{<t>} = \Gamma_u \cdot \tilde{c}^{<t>} + (1 - \Gamma_u) \cdot c^{<t-1>}$$

$$\hat{y}^{<t>} = g(W_y c^{<t>} + b_y)$$

Fully-gated recurrent unit



$$\tilde{c}^{<t>} = \tanh(W_c[\Gamma_r \cdot c^{<t-1>}, x^{<t>}] + b_c)$$

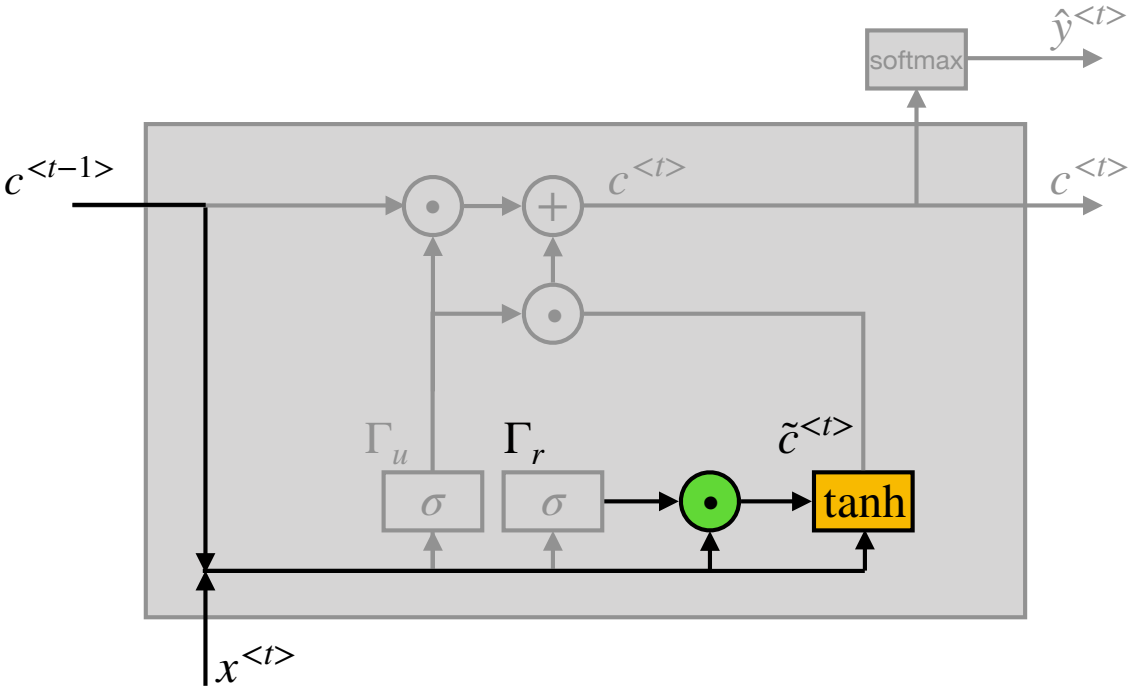
$$\Gamma_u = \sigma(W_u[c^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_r = \sigma(W_r[c^{<t-1>}, x^{<t>}] + b_r)$$

$$c^{<t>} = \Gamma_u \cdot \tilde{c}^{<t>} + (1 - \Gamma_u) \cdot c^{<t-1>}$$

$$\hat{y}^{<t>} = g(W_y c^{<t>} + b_y)$$

Fully-gated recurrent unit



$$\tilde{c}^{<t>} = \tanh(W_c[\Gamma_r \cdot c^{<t-1>}, x^{<t>}] + b_c)$$

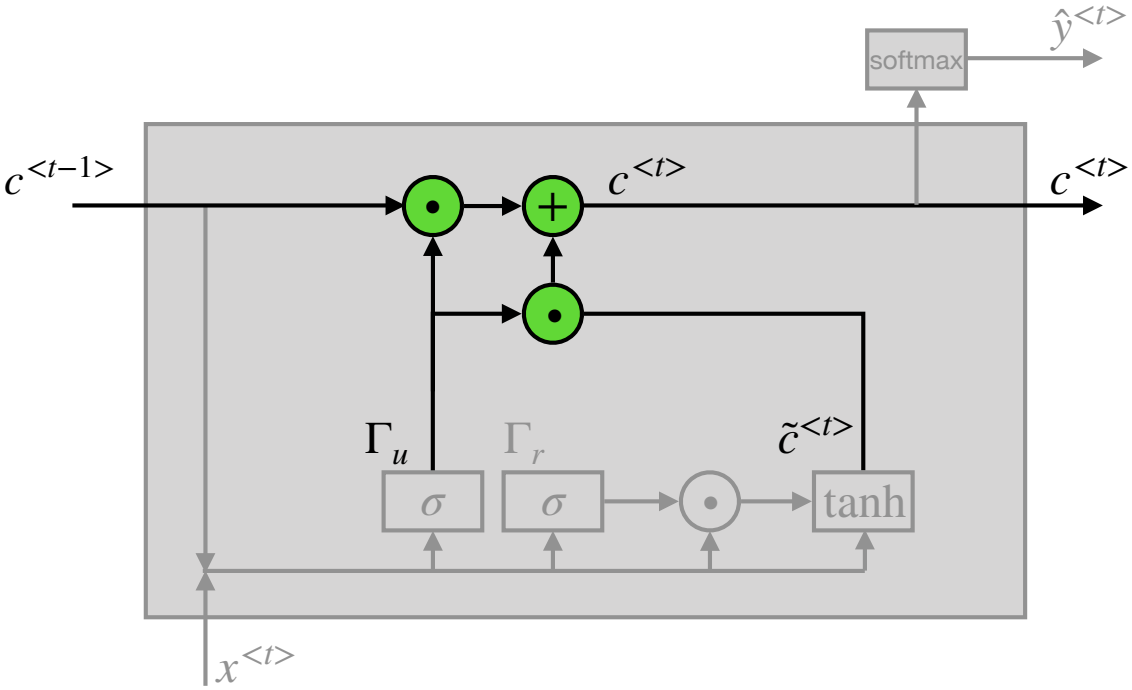
$$\Gamma_u = \sigma(W_u[c^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_r = \sigma(W_r[c^{<t-1>}, x^{<t>}] + b_r)$$

$$c^{<t>} = \Gamma_u \cdot \tilde{c}^{<t>} + (1 - \Gamma_u) \cdot c^{<t-1>}$$

$$\hat{y}^{<t>} = g(W_y c^{<t>} + b_y)$$

Fully-gated recurrent unit



$$\tilde{c}^{<t>} = \tanh(W_c[\Gamma_r \cdot c^{<t-1>}, x^{<t>}] + b_c)$$

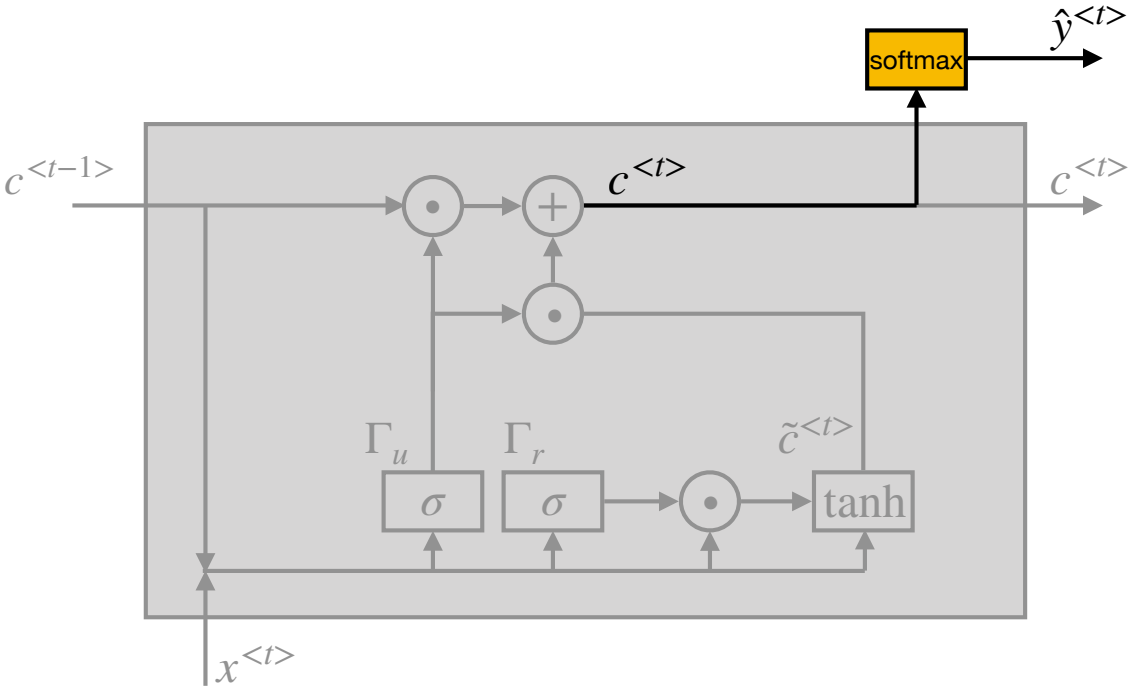
$$\Gamma_u = \sigma(W_u[c^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_r = \sigma(W_r[c^{<t-1>}, x^{<t>}] + b_r)$$

$$c^{<t>} = \Gamma_u \cdot \tilde{c}^{<t>} + (1 - \Gamma_u) \cdot c^{<t-1>}$$

$$\hat{y}^{<t>} = g(W_y c^{<t>} + b_y)$$

Fully-gated recurrent unit



$$\tilde{c}^{<t>} = \tanh(W_c[\Gamma_r \cdot c^{<t-1>}, x^{<t>}] + b_c)$$

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

$$\Gamma_r = \sigma(W_r[c^{<t-1>}, x^{<t>}] + b_r)$$

$$c^{<t>} = \Gamma_u \cdot \tilde{c}^{<t>} + (1 - \Gamma_u) \cdot c^{<t-1>}$$

$$\hat{y}^{<t>} = g(W_y c^{<t>} + b_y)$$

Long Short Term Memory (LSTM) unit

$$\tilde{c}^{<t>} = \tanh(W_c[\Gamma_r \cdot c^{<t-1>}, x^{<t>}] + b_c)$$

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

$$\Gamma_r = \sigma(W_r[c^{<t-1>}, x^{<t>}] + b_r)$$

$$c^{<t>} = \Gamma_u \cdot \tilde{c}^{<t>} + (1 - \Gamma_u) \cdot c^{<t-1>}$$

$$\hat{y}^{<t>} = g(W_y c^{<t>} + b_y)$$

$\longrightarrow$

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

$$\Gamma_u = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f)$$

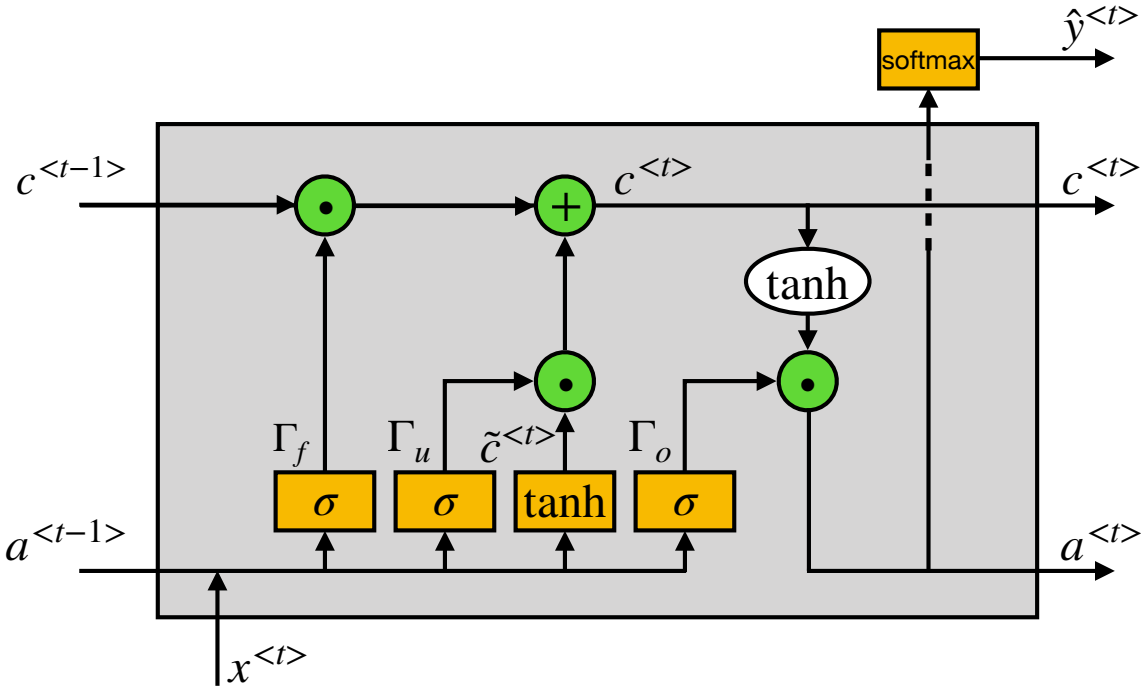
$$\Gamma_o = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o)$$

$$c^{<t>} = \Gamma_u \cdot \tilde{c}^{<t>} + \Gamma_f \cdot c^{<t-1>}$$

$$a^{<t>} = \Gamma_o \cdot \tanh(c^{<t>})$$

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

Long Short Term Memory (LSTM) unit



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

$$\Gamma_u = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f)$$

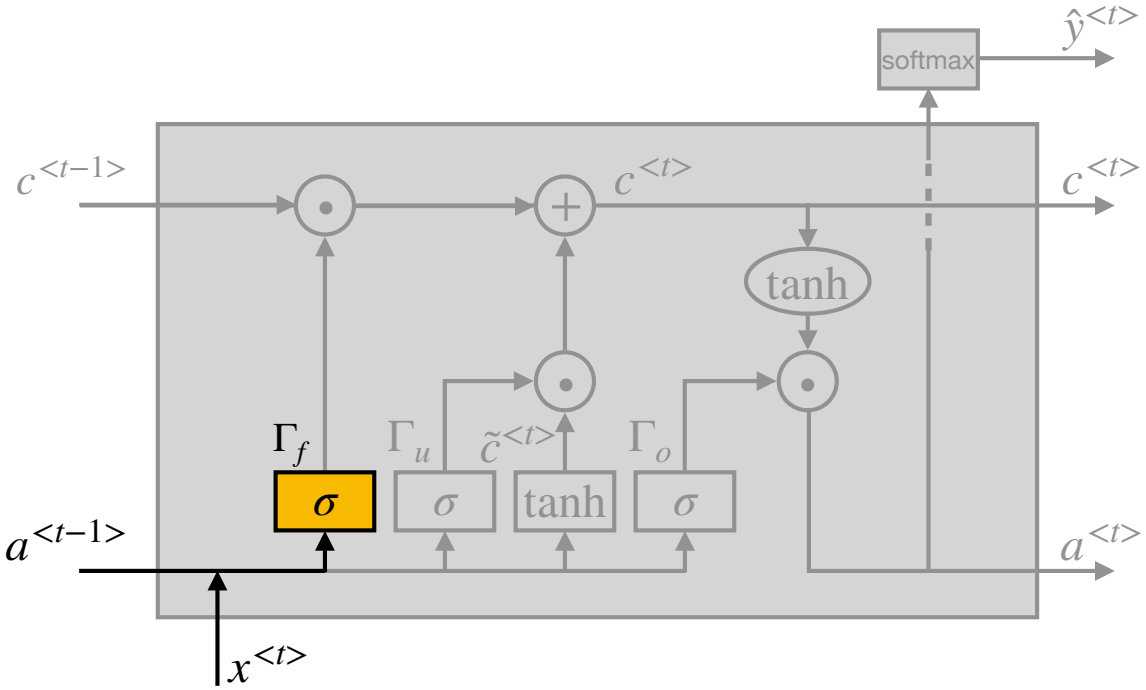
$$\Gamma_o = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o)$$

$$c^{<t>} = \Gamma_u \cdot \tilde{c}^{<t>} + \Gamma_f \cdot c^{<t-1>}$$

$$a^{<t>} = \Gamma_o \cdot \tanh(c^{<t>})$$

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

Long Short Term Memory (LSTM) unit



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

$$\Gamma_u = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f)$$

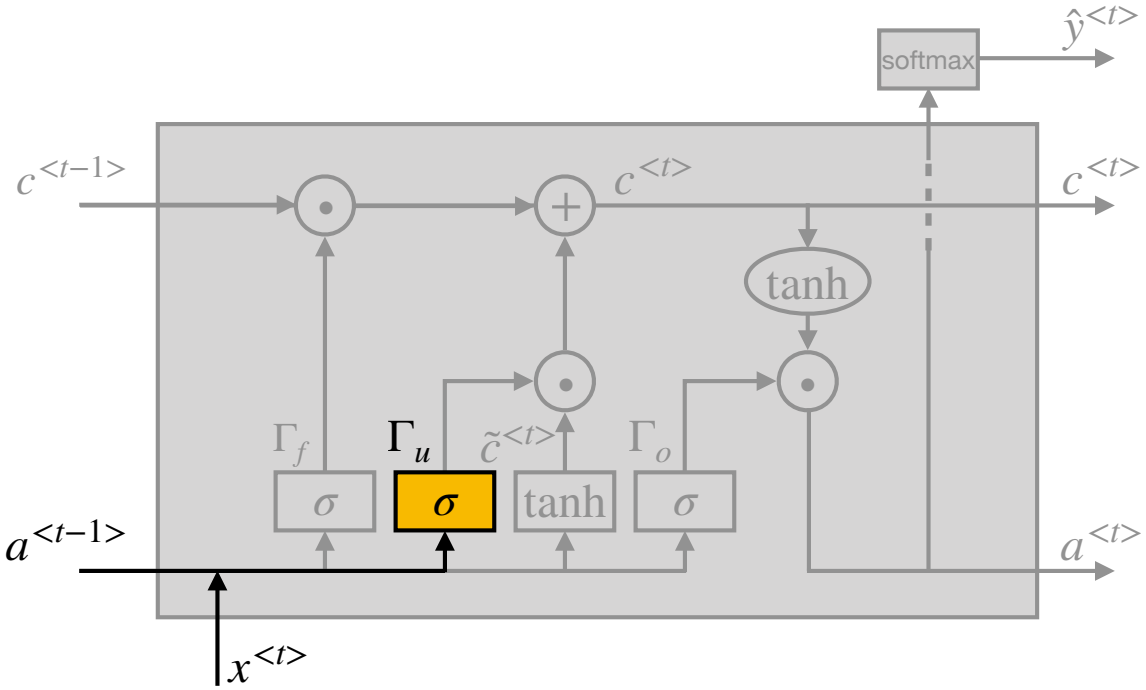
$$\Gamma_o = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o)$$

$$c^{<t>} = \Gamma_u \cdot \tilde{c}^{<t>} + \Gamma_f \cdot c^{<t-1>}$$

$$a^{<t>} = \Gamma_o \cdot \tanh(c^{<t>})$$

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

Long Short Term Memory (LSTM) unit



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

$$\Gamma_u = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f)$$

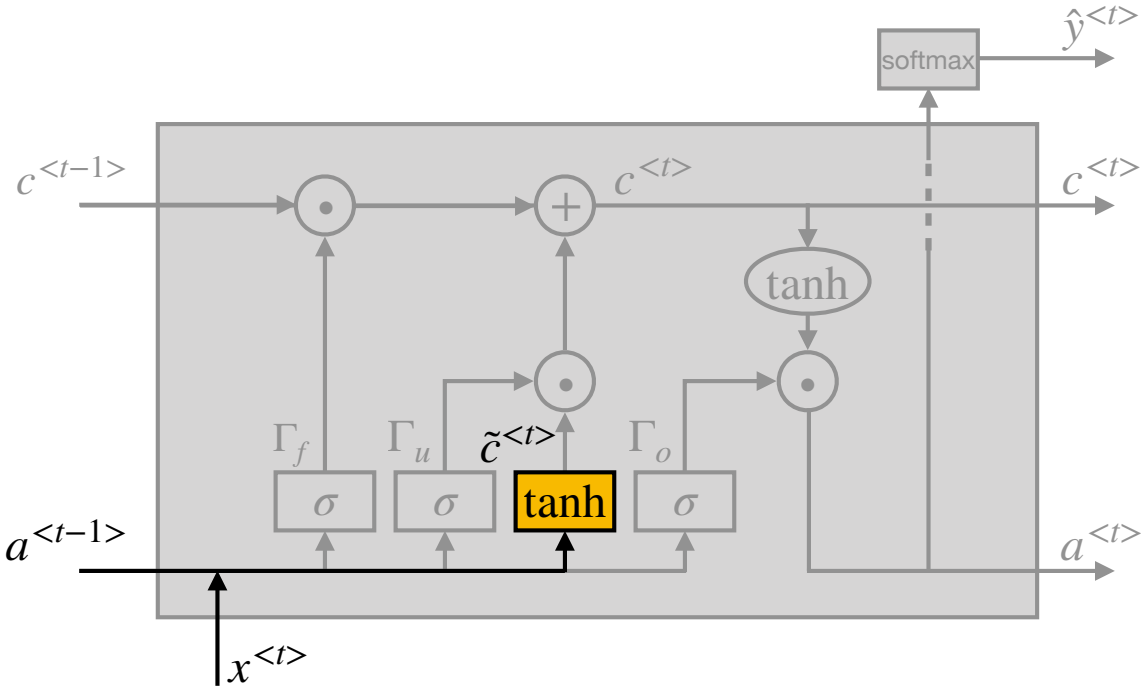
$$\Gamma_o = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o)$$

$$c^{<t>} = \Gamma_u \cdot \tilde{c}^{<t>} + \Gamma_f \cdot c^{<t-1>}$$

$$a^{<t>} = \Gamma_o \cdot \tanh(c^{<t>})$$

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

Long Short Term Memory (LSTM) unit



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

$$\Gamma_u = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f)$$

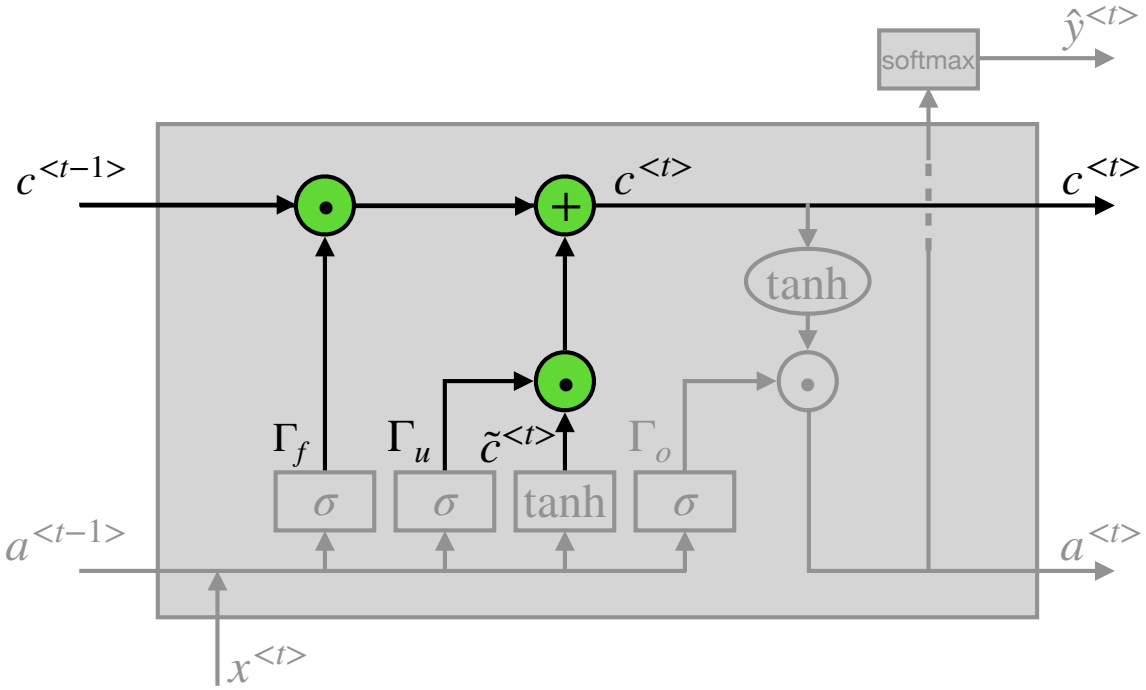
$$\Gamma_o = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o)$$

$$c^{<t>} = \Gamma_u \cdot \tilde{c}^{<t>} + \Gamma_f \cdot c^{<t-1>}$$

$$a^{<t>} = \Gamma_o \cdot \tanh(c^{<t>})$$

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

Long Short Term Memory (LSTM) unit



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

$$\Gamma_u = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f)$$

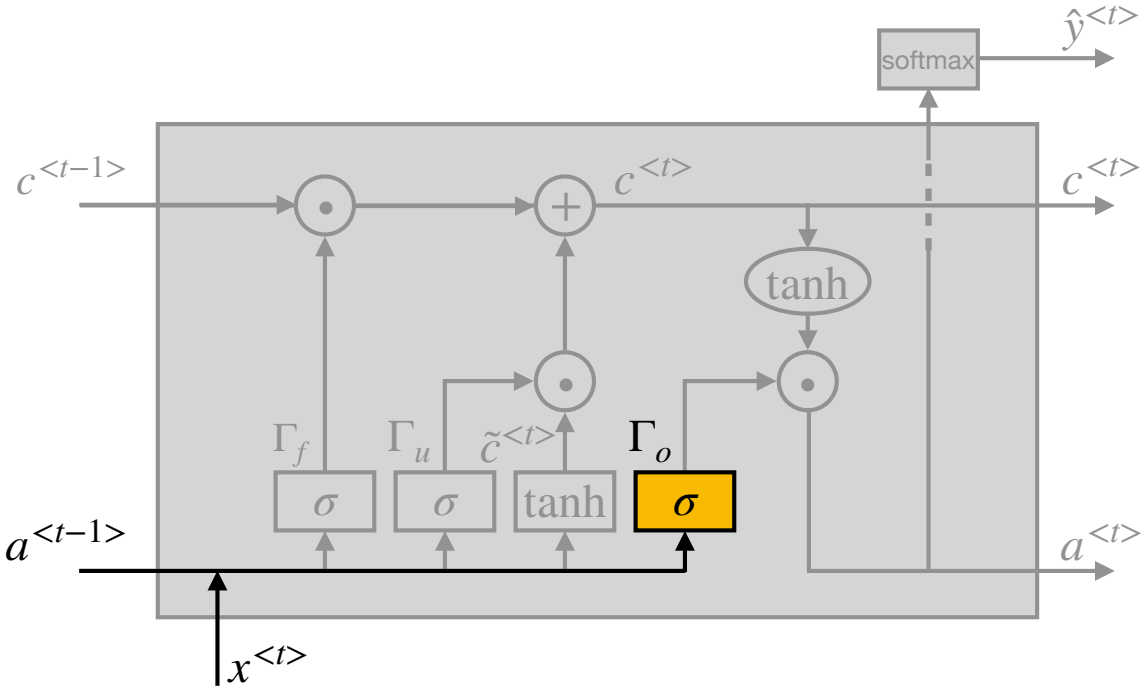
$$\Gamma_o = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o)$$

$$c^{<t>} = \Gamma_u \cdot \tilde{c}^{<t>} + \Gamma_f \cdot c^{<t-1>}$$

$$a^{<t>} = \Gamma_o \cdot \tanh(c^{<t>})$$

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

Long Short Term Memory (LSTM) unit



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

$$\Gamma_u = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f)$$

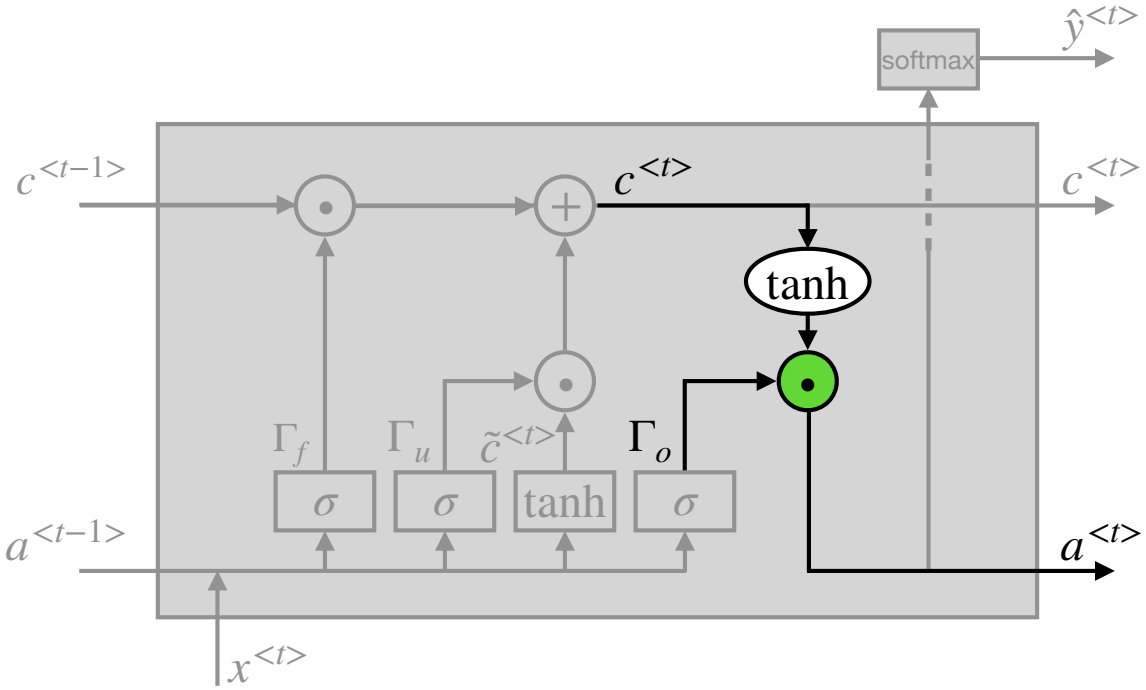
$$\Gamma_o = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o)$$

$$c^{<t>} = \Gamma_u \cdot \tilde{c}^{<t>} + \Gamma_f \cdot c^{<t-1>}$$

$$a^{<t>} = \Gamma_o \cdot \tanh(c^{<t>})$$

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

Long Short Term Memory (LSTM) unit



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

$$\Gamma_u = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f)$$

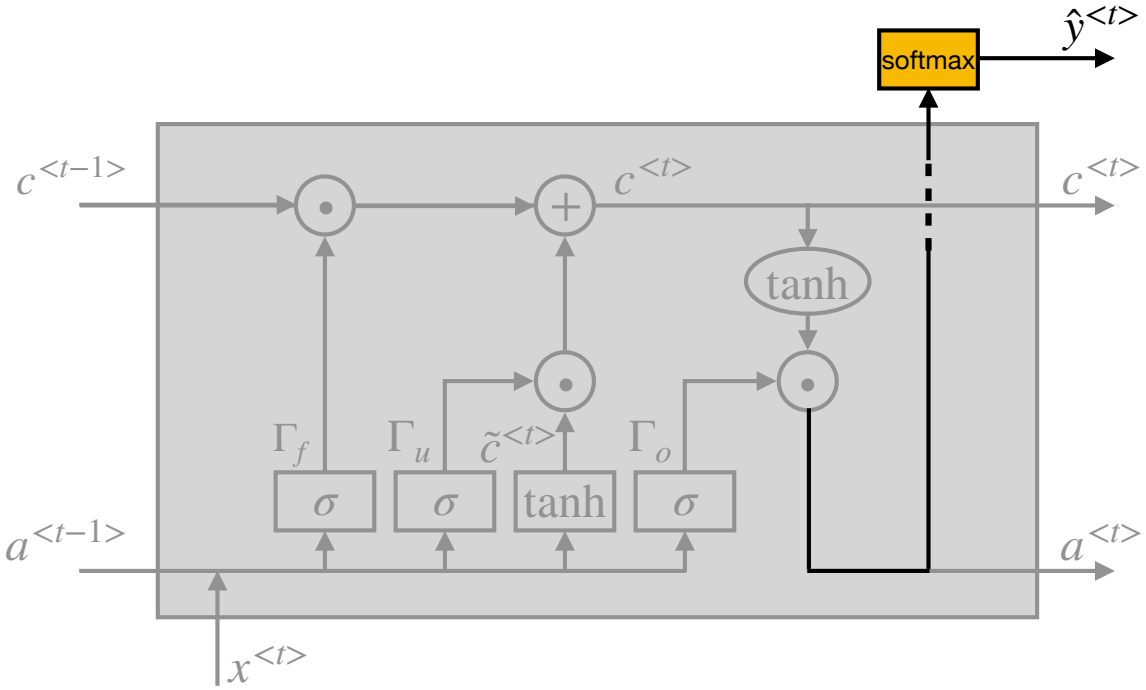
$$\Gamma_o = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o)$$

$$c^{<t>} = \Gamma_u \cdot \tilde{c}^{<t>} + \Gamma_f \cdot c^{<t-1>}$$

$$a^{<t>} = \Gamma_o \cdot \tanh(c^{<t>})$$

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

Long Short Term Memory (LSTM) unit



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

$$\Gamma_u = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f)$$

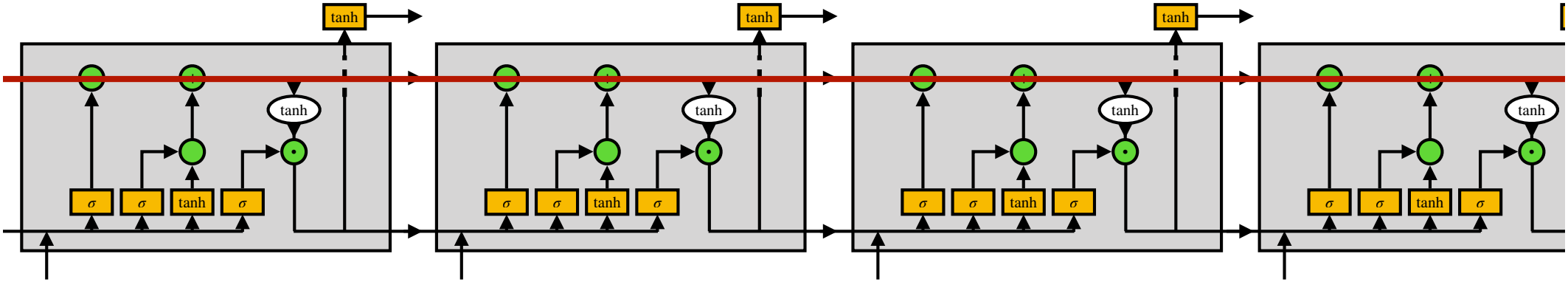
$$\Gamma_o = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o)$$

$$c^{<t>} = \Gamma_u \cdot \tilde{c}^{<t>} + \Gamma_f \cdot c^{<t-1>}$$

$$a^{<t>} = \Gamma_o \cdot \tanh(c^{<t>})$$

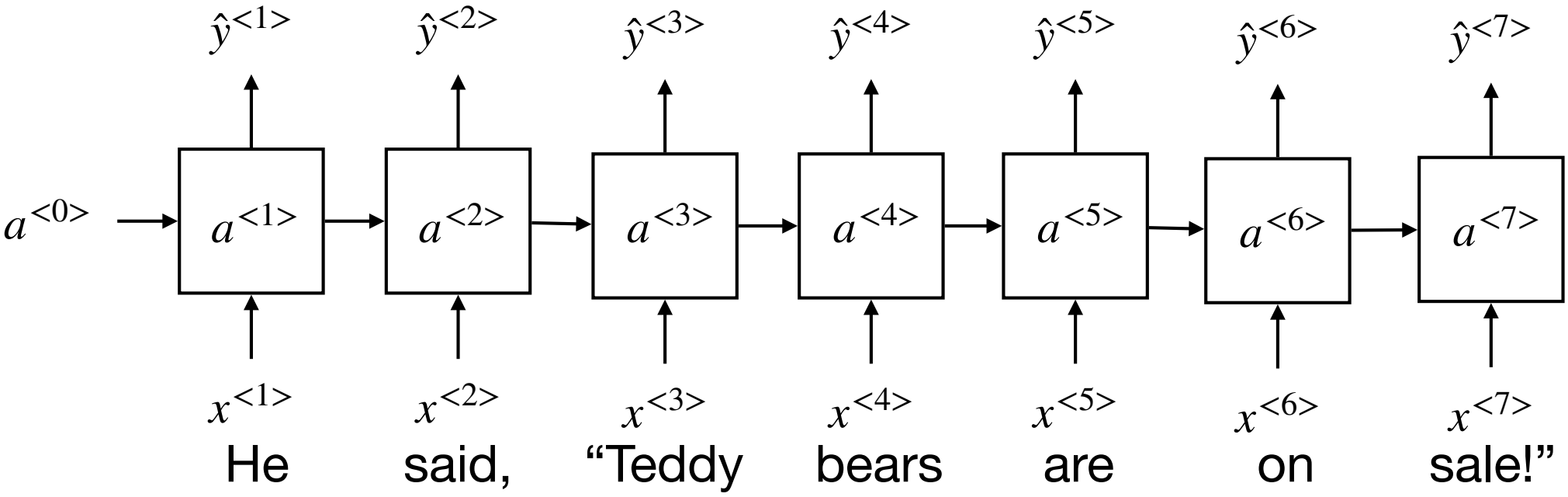
$$\hat{y}^{<t>} = g(W_y a^{<t>} + b_y)$$

Long Short Term Memory (LSTM) unit



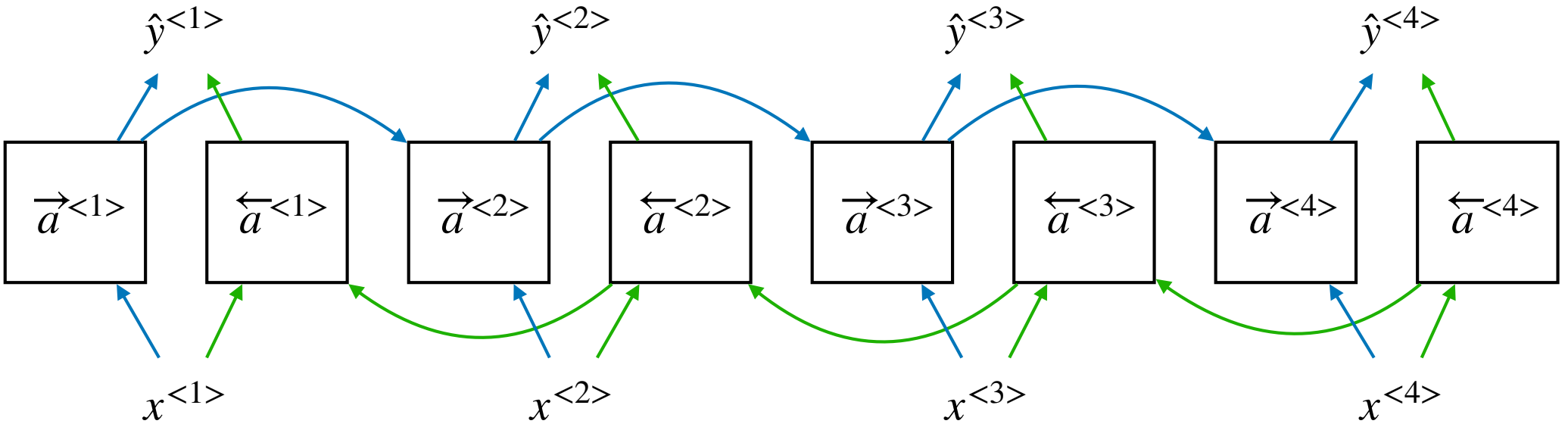
Bidirectional RNNs

- He said, “Teddy bears are on sale!”
- He said, “Teddy Roosevelt was a great president!”



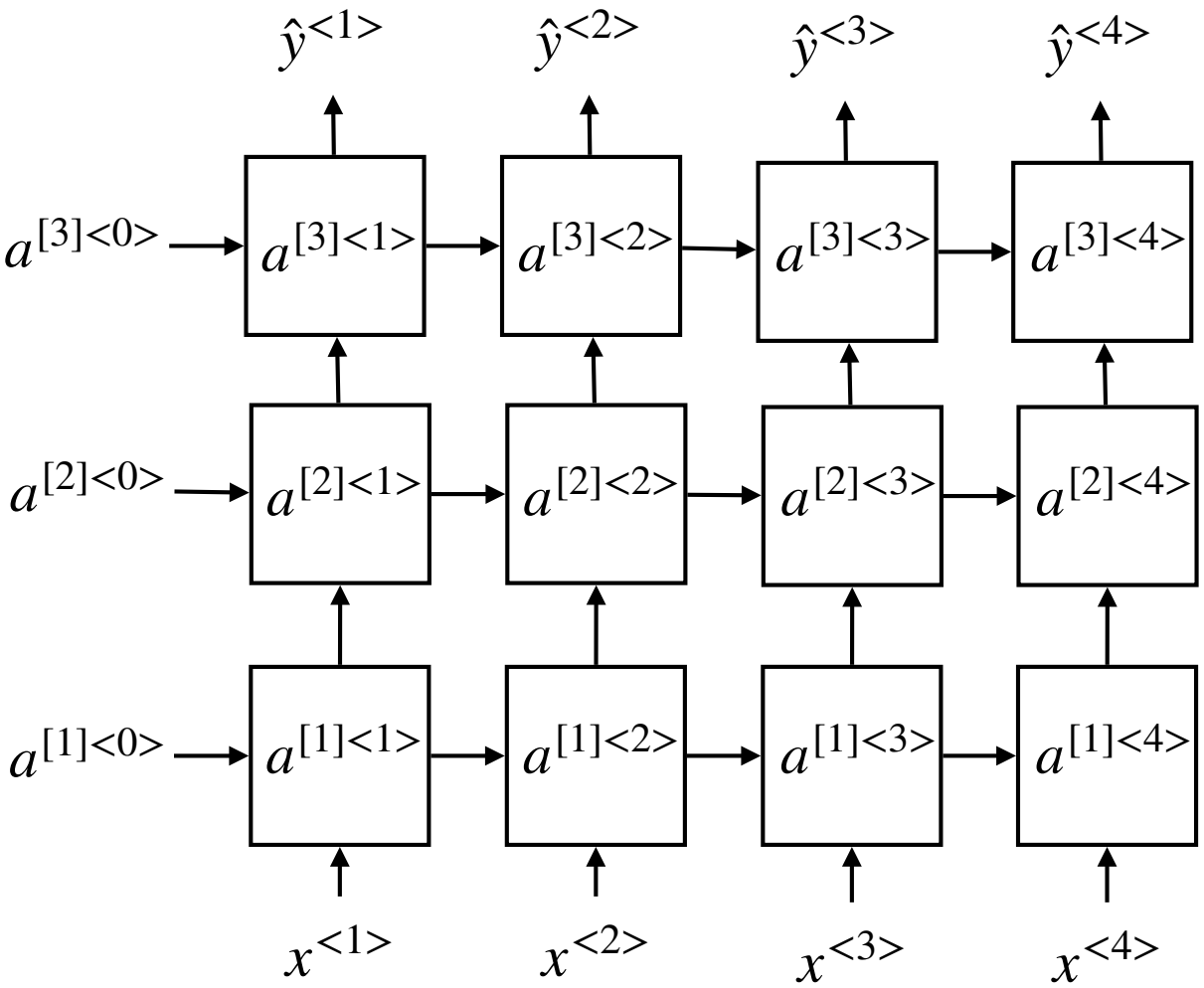
Bidirectional RNNs

- He said, “Teddy bears are on sale!”
- He said, “Teddy Roosevelt was a great president!”

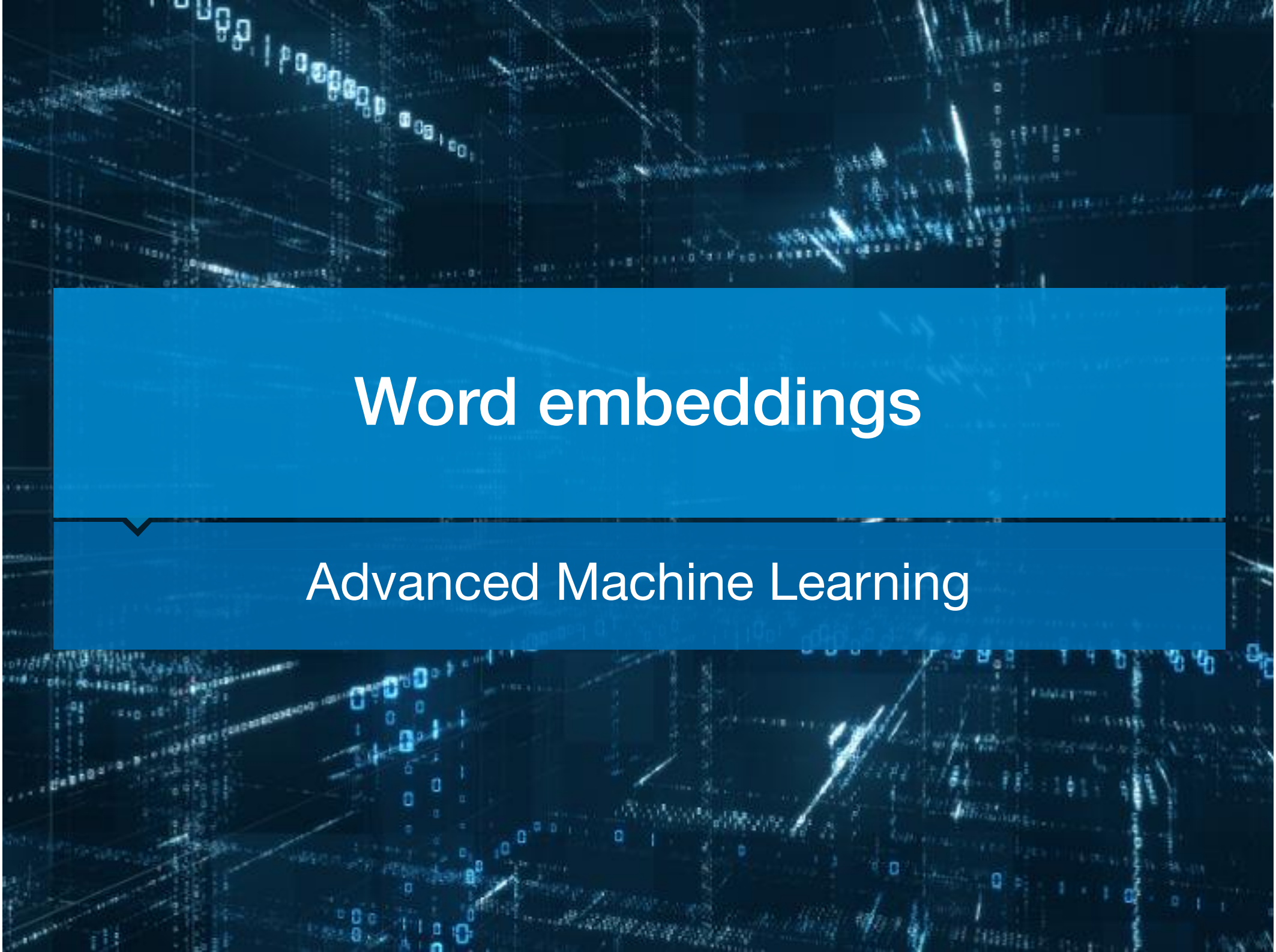


$$\hat{y}^{<t>} = g(W_y[\overrightarrow{a}^{<t>}, \overleftarrow{a}^{<t>}] + b_y)$$

Deep RNNs



$$a^{[2]<3>} = g(W_a^{[2]}[a^{[2]<2>}, a^{[1]<3>}] + b_a^{[2]}$$



Word embeddings

Advanced Machine Learning

Word representation

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

man	woman	king	queen	apple	orange
(5391)	(9853)	(4914)	(7157)	(456)	(6257)

$$\begin{pmatrix}0\\0\\0\\0\\\vdots\\1\\\vdots\\0\\0\end{pmatrix} \quad \begin{pmatrix}0\\0\\0\\0\\\vdots\\1\\\vdots\\0\end{pmatrix} \quad \begin{pmatrix}0\\0\\0\\\vdots\\1\\\vdots\\0\\0\\0\end{pmatrix} \quad \begin{pmatrix}0\\0\\0\\0\\\vdots\\1\\\vdots\\0\end{pmatrix} \quad \begin{pmatrix}0\\\vdots\\1\\\vdots\\0\\0\\0\\0\end{pmatrix} \quad \begin{pmatrix}0\\0\\0\\0\\\vdots\\1\\\vdots\\0\end{pmatrix}$$

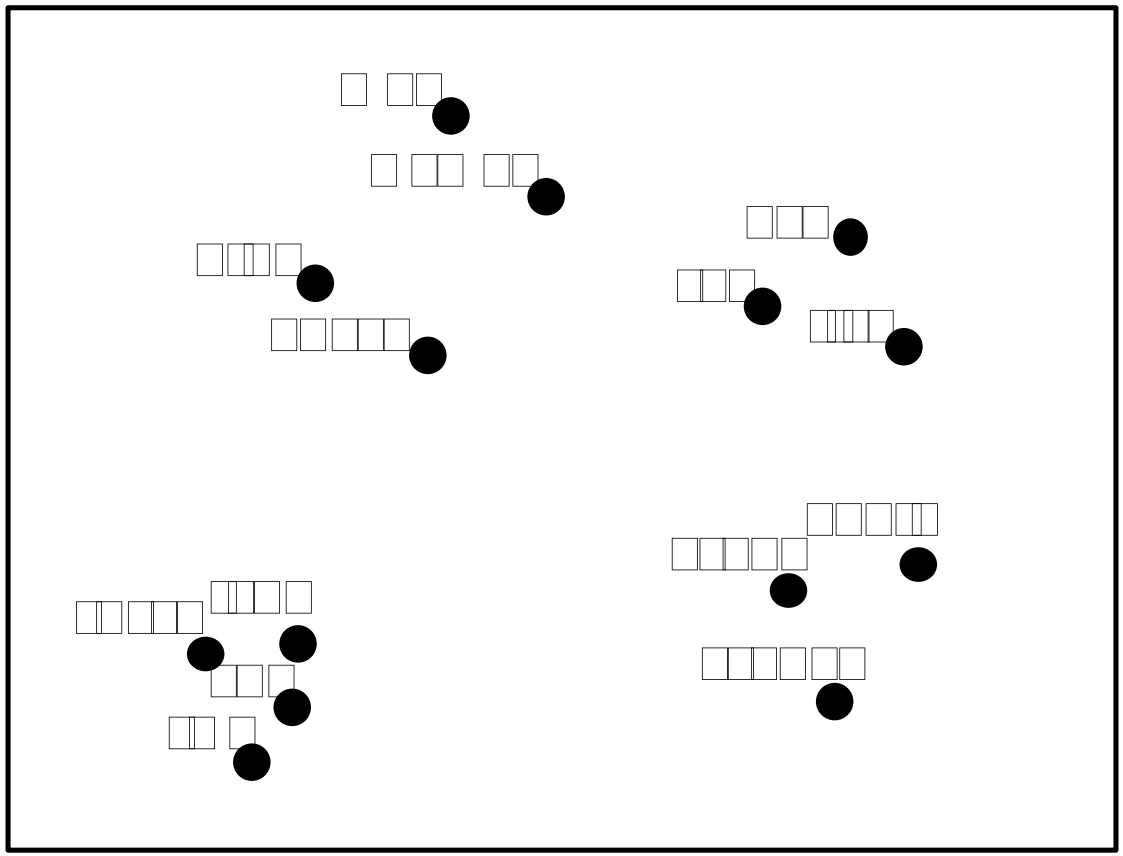
I want a glass of orange _____.

I want a glass of apple _____.

Word embedding

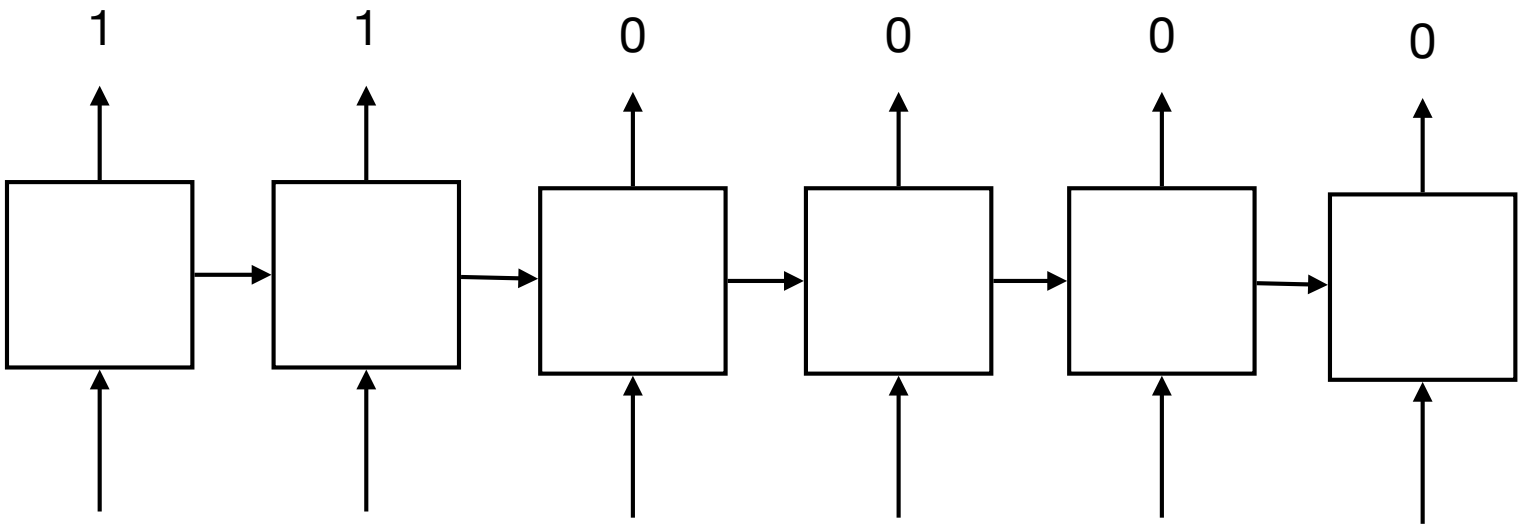
	man (5391)	woman (9853)	king (4914)	queen (7157)	apple (456)	orange (6257)
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.70	0.69	0.03	-0.02
Food	0.09	0.01	0.02	0.01	0.95	0.97

Word embedding



Visualization with t-SNE

Word embedding



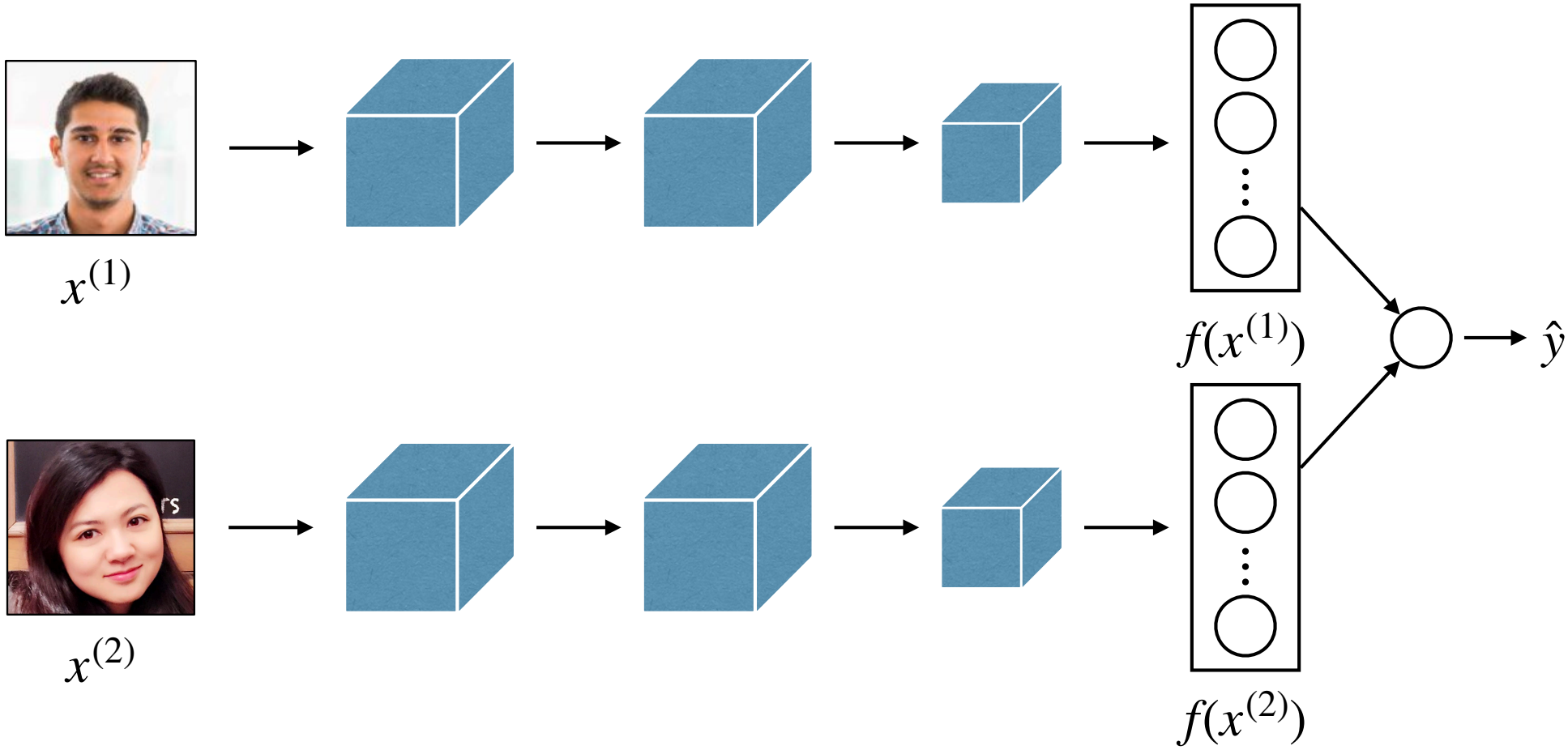
- Sally Johnson is an orange farmer
- Robert Lin is an apple farmer
- Jack Jones is a durian cultivator

Word embedding

- Learn word embeddings from large text corpus (1-100B words) or download pre-trained embedding online
- Transfer embedding to new task with smaller training set (say, 100K words)
- Optional: continue to finetune the word embeddings with new data

Word embedding

- Relation to face encoding



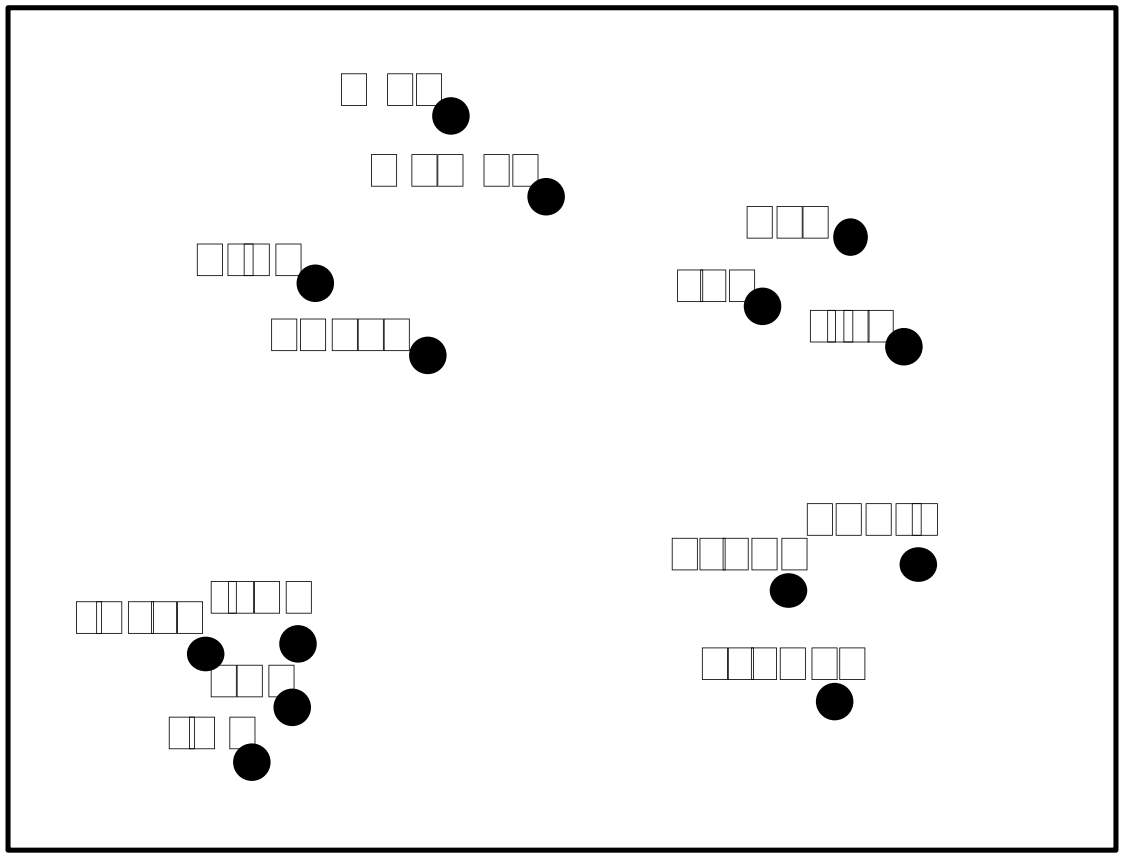
Word embedding

- Analogy to 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	0.70	0.69	0.03	-0.02
Food	0.09	0.01	0.02	0.01	0.95	0.97

- man → woman as king → ??

Word embedding



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

Word embedding

- Similarity function: $\text{sim}(u, v) = \dfrac{u^\top v}{\|u\|_2 \, \|v\|_2}$

- Examples:
  - Man:Woman as Boy:Girl
  - Ottawa:Canada as Nairobi:Kenya
  - Big:Bigger as Tall:Taller
  - Yen:Japan as Ruble:Russia

Learning word embeddings

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

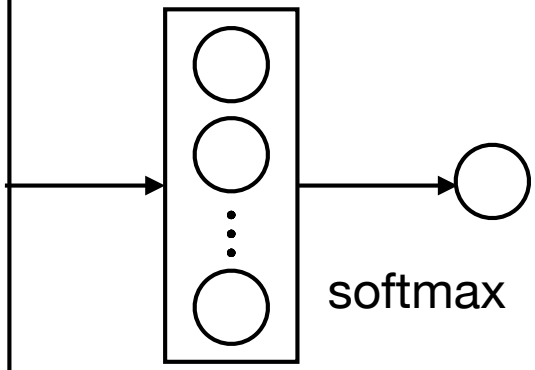
- Say that vocabulary has 10,000 words
- We want an embedding of 300 words
- How to learn an embedding matrix E ?
  size of matrix = $300 \times 10{,}000$

Neural language model

- I want a glass of orange _____.

4343 9665 1 3852 6163 6257

I	o_{4343}	$\rightarrow$	E	$\rightarrow$	e_{4343}
want	o_{9665}	$\rightarrow$	E	$\rightarrow$	e_{9665}
a	o_{1}	$\rightarrow$	E	$\rightarrow$	e_{1}
glass	o_{3852}	$\rightarrow$	E	$\rightarrow$	e_{3852}
of	o_{6163}	$\rightarrow$	E	$\rightarrow$	e_{6163}
orange	o_{6257}	$\rightarrow$	E	$\rightarrow$	e_{6257}



Neural language model

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

- Other contexts:
  - 4 words on left and right
    “a glass of orange” … “to go along with”
  - Last 1 word
    “orange” …
  - Nearby 1 word (skip gram)
    “glass” …

Neural language model

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

context	word
orange	juice
orange	glass
orange	my

- Vocabulary: 10,000 words
- Model: $O_c \rightarrow E \rightarrow e_c \rightarrow$ softmax $\rightarrow \hat{y}$
- Parameter associated with output t is θ_t

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

Word2vec - skip gram model

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

- Computationally expensive; you need complete vocabulary
  - > Hierarchical softmax ($\log V$)

- How to sample the context c?
  - > "the", "of", "a", "and", "to", …
  - > "orange", "apple", "durian", …

Word2vec - negative sampling

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

context	word	target?
orange	juice	1
orange	king	0
orange	book	0
orange	the	0
orange	of	0

- Vocabulary: 10,000 words
- Model: $O_c \rightarrow E \rightarrow e_c \rightarrow$ 10,000 log.res. problems
- $\mathbb{P}(y = 1 \mid c, t) = \sigma(\theta_t^\top e_c)$

Word2vec - negative sampling

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

context	word	target?
orange	juice	1
orange	king	0
orange	book	0
orange	the	0
orange	of	0

- How to select negative samples?
  - > Uniform
  - > Inverse frequency

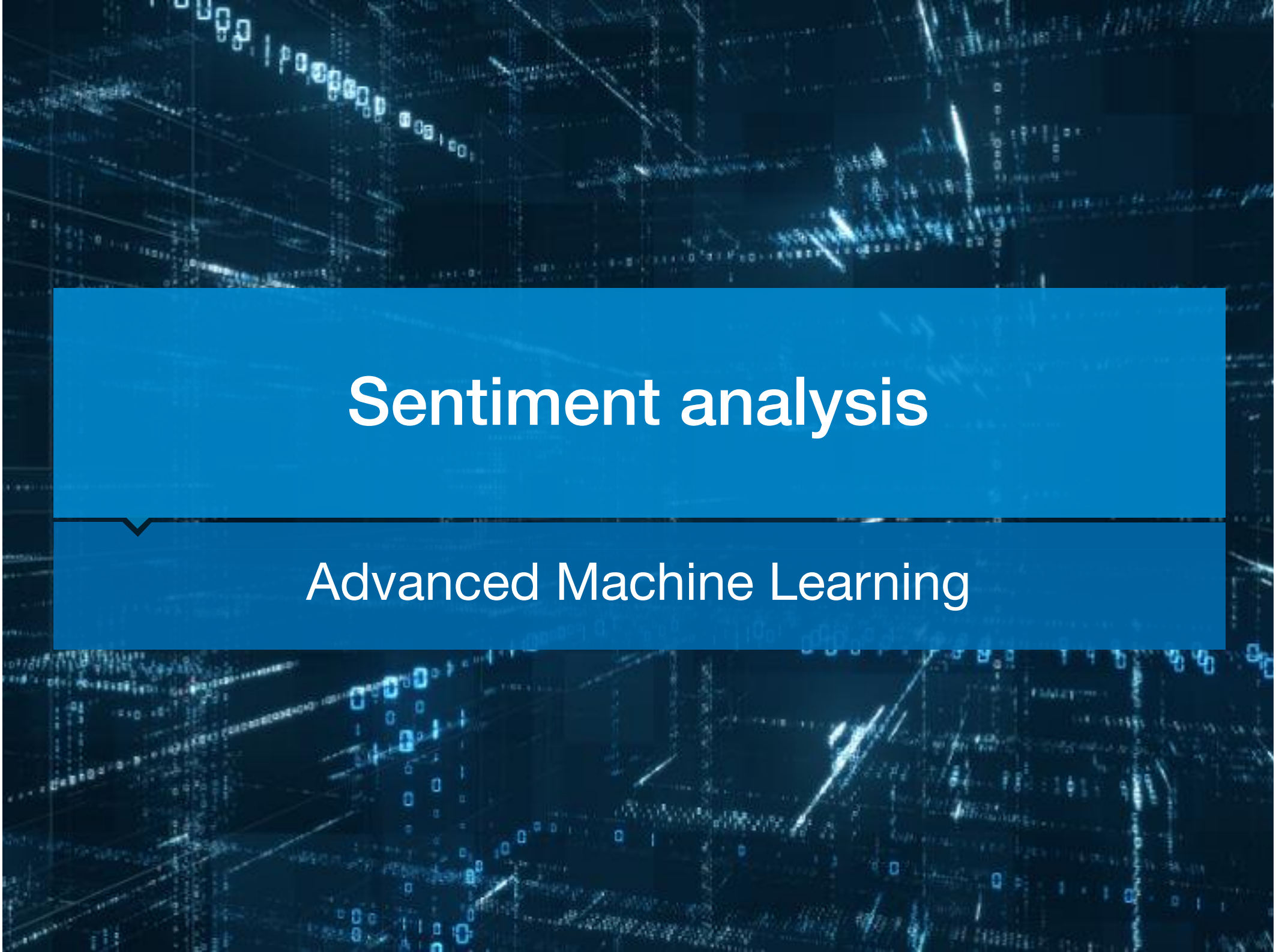
$$p(w_i) = \frac{f(w_i)^{\frac{3}{4}}}{\sum_{j=1}^{10,000} f(w_j)^{\frac{3}{4}}}$$

Word2vec - global vectors (GloVe)

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

- Context c
- Target t

- X_{ij} = # times word j appears in context of word i
- Usually, $x_{ij} = x_{ji}$
- Minimize $\sum_{i=1}^{10,000} \sum_{j=1}^{10,000} f(X_{ij})(e_i^\top e_j + b_i + b_j - \log X_{ij})^2$



Sentiment analysis

Advanced Machine Learning

Sentiment analysis

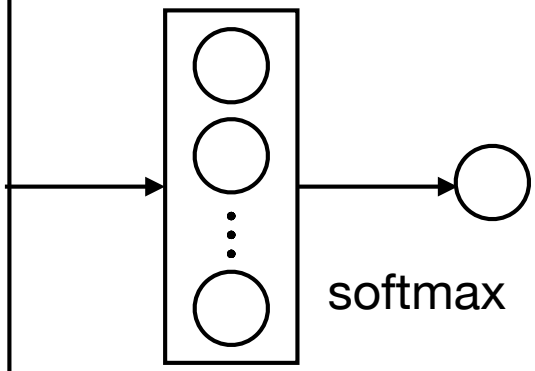
x	y
The dessert is excellent.	
Service was quite slow.	
Good for a quick meal, but nothing special.	
Completely lacking in good taste, good service, and good ambience.	

Sentiment analysis

- The dessert is excellent

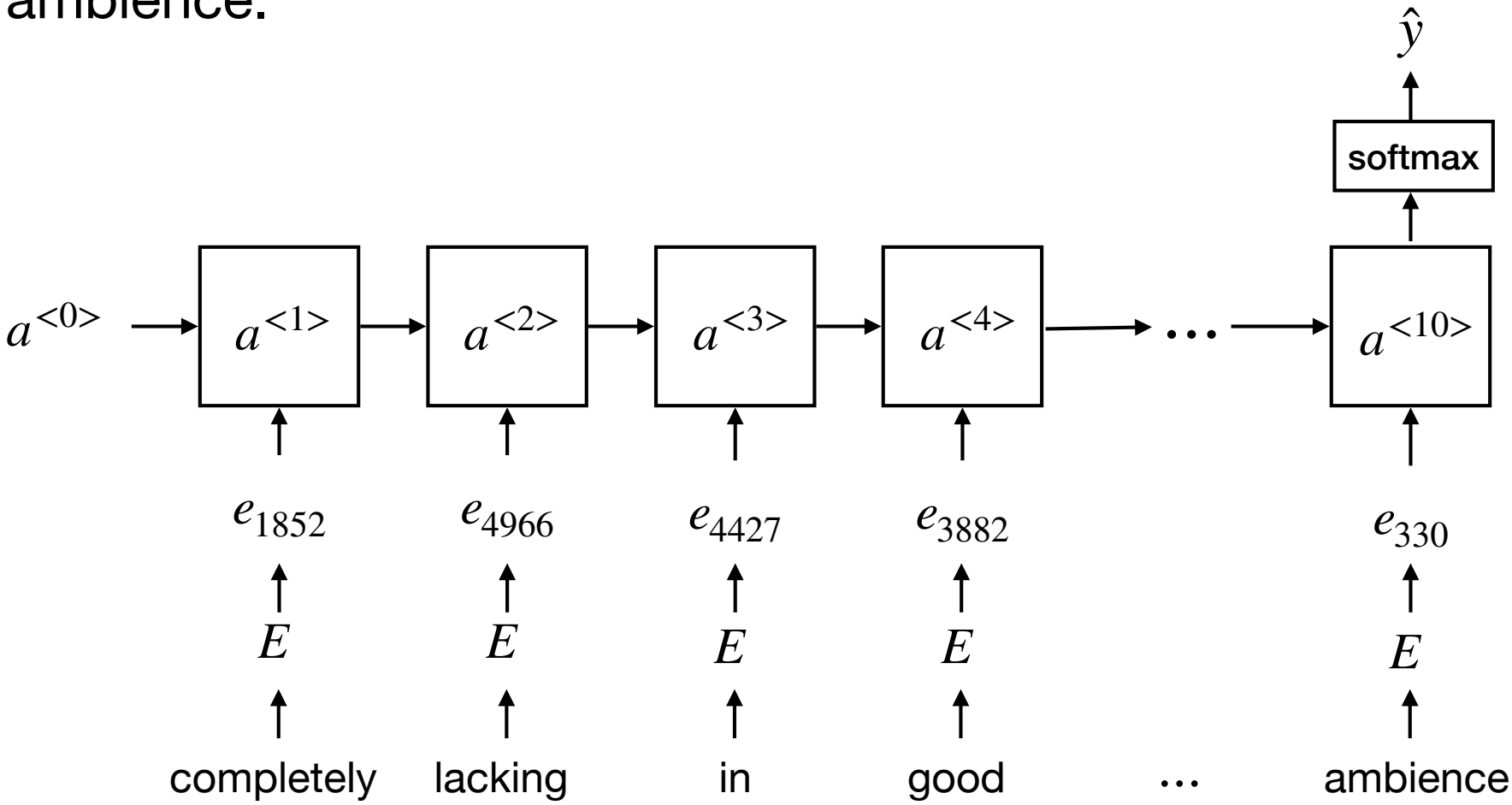
8928 2468 4694 3180

The	o_{8928}	$\rightarrow$	E	$\rightarrow$	e_{8928}
dessert	o_{2468}	$\rightarrow$	E	$\rightarrow$	e_{2468}
is	o_{4694}	$\rightarrow$	E	$\rightarrow$	e_{4694}
excellent	o_{3180}	$\rightarrow$	E	$\rightarrow$	e_{3180}



Sentiment analysis

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



Bias in word embeddings

- Bias in word embeddings

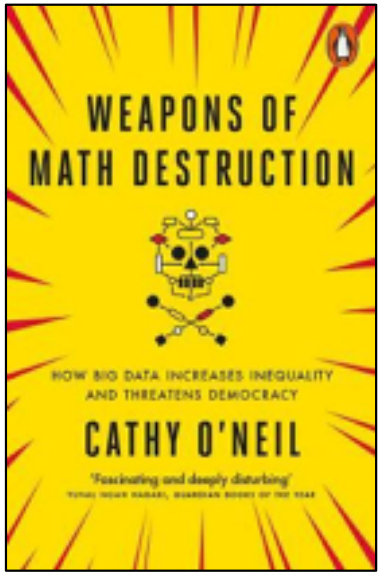
- Man:Woman as King:??
- Man:Computer_Programmer as Woman:??
- Father:Doctor as Mother:??

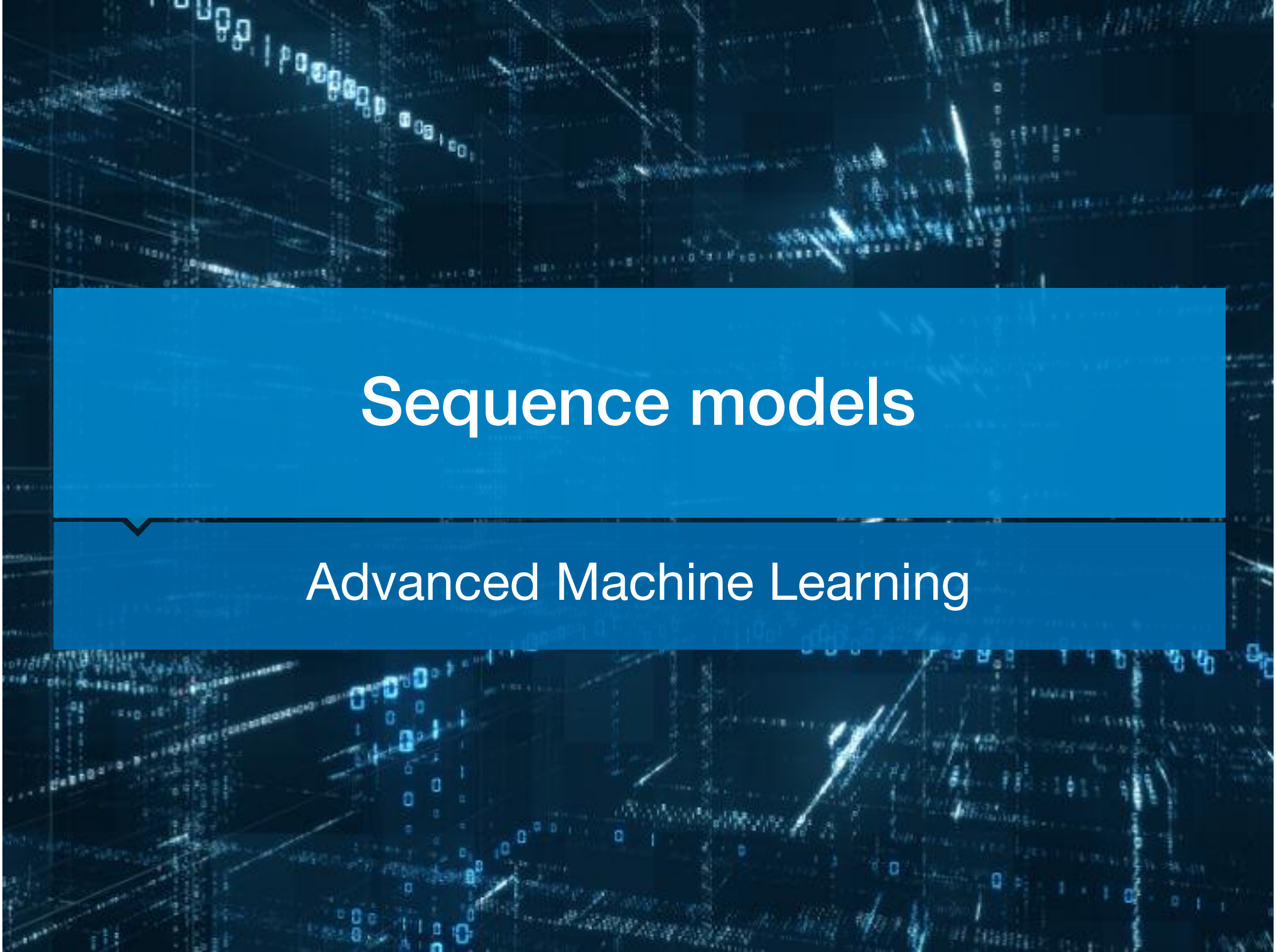
Bias in word embeddings

- Bias in word embeddings

- Man:Woman as King:Queen
- Man:Computer_Programmer as Woman:Homemaker
- Father:Doctor as Mother:Nurse

- Word embeddings can reflect gender, ethnicity age, sexual orientation, and other biases of the text used to train the model

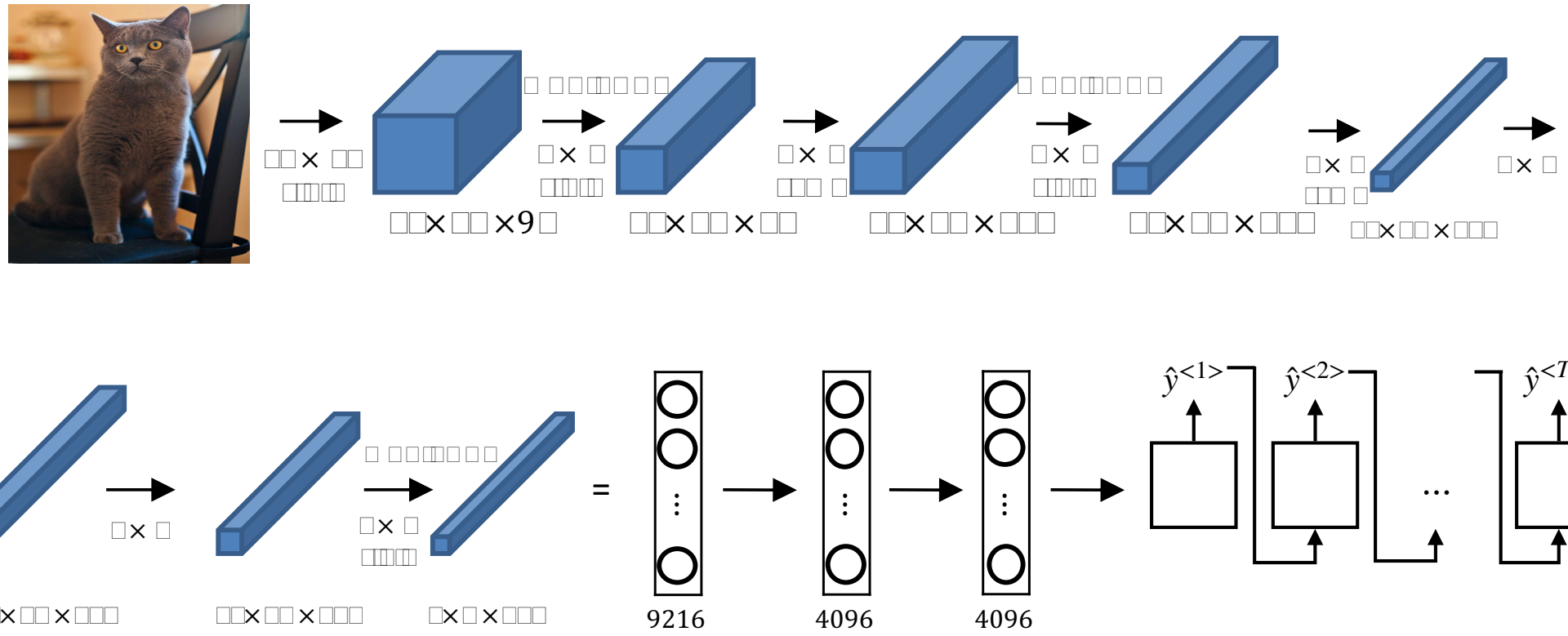




Sequence models

Advanced Machine Learning

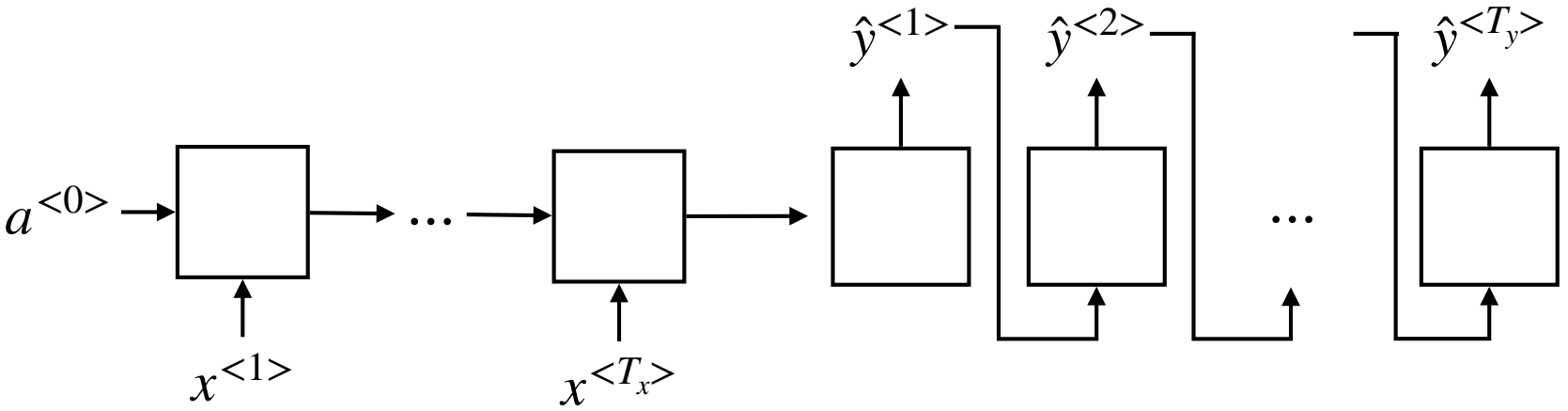
Image captioning



Machine translation

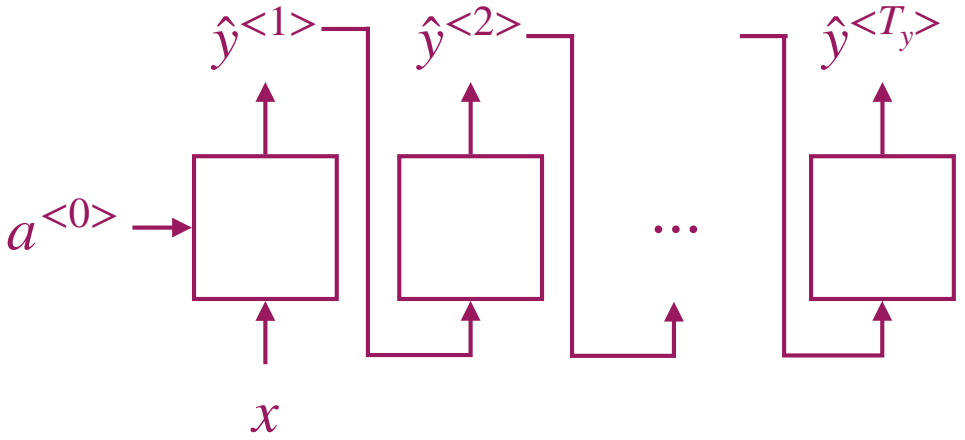
- Jane visite l'Afrique en Septembre.
  $x^{<1>}$ $x^{<2>}$ $x^{<3>}$ $x^{<4>}$ $x^{<5>}$

- Jane is visiting Africa in September.
  $y^{<1>}$ $y^{<2>}$ $y^{<3>}$ $y^{<4>}$ $y^{<5>}$ $y^{<6>}$

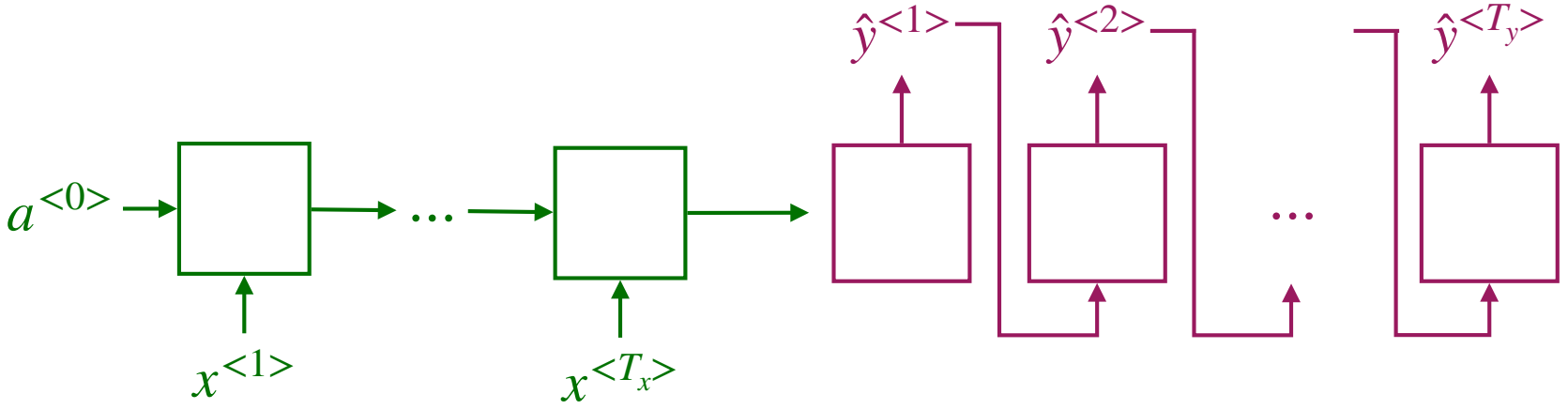


Machine translation

- Language model



- Machine translation



Machine translation

- Jane visite l'Afrique en Septembre.
  - > 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.

$$\mathbb{P}(y^{<1>}, \ldots, y^{<T_y>} \quad x)$$

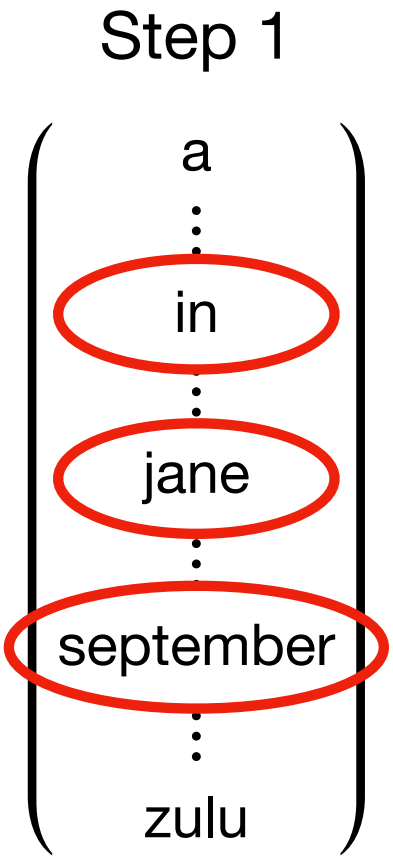
Machine translation

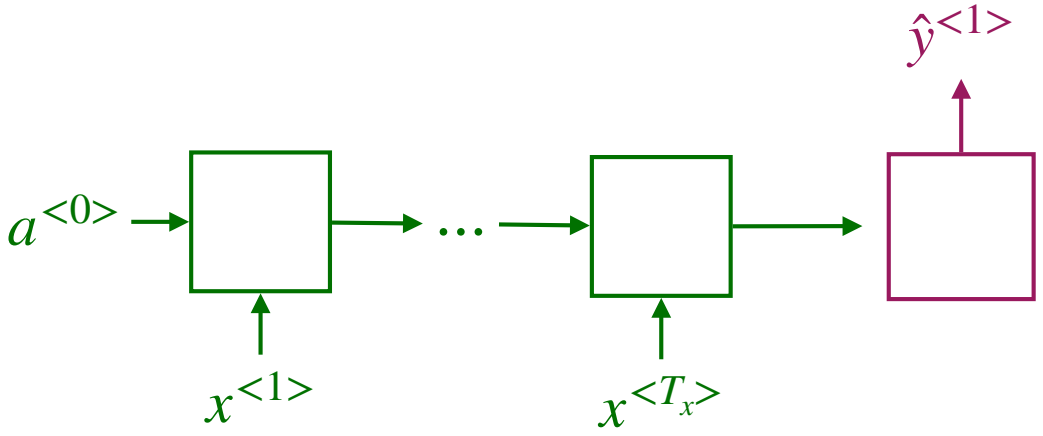
$$\arg \max_{y^{<1>},\ldots,y^{<T_y>}} \mathbb{P}(y^{<1>}, \ldots, y^{<T_y>} \mid x)$$

- Why not greedy search?
  - Jane is visiting Africa in September.
  - Jane is going to be visiting Africa in September.

$$\mathbb{P}(\text{Jane is going} \mid x) > \mathbb{P}(\text{Jane is visiting} \mid x)$$

Beam search

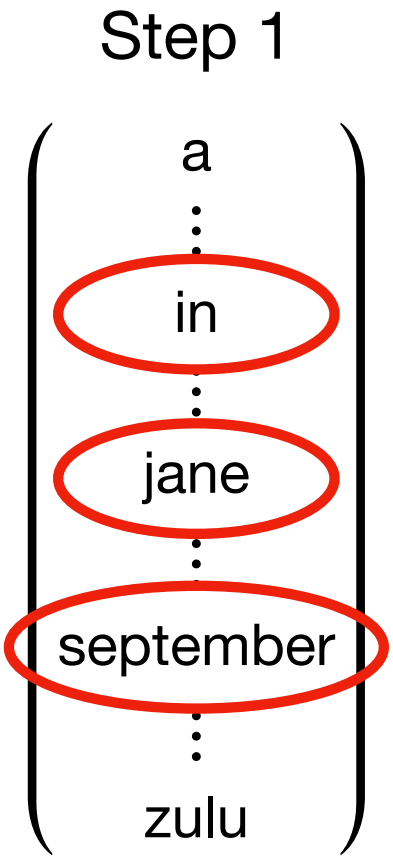


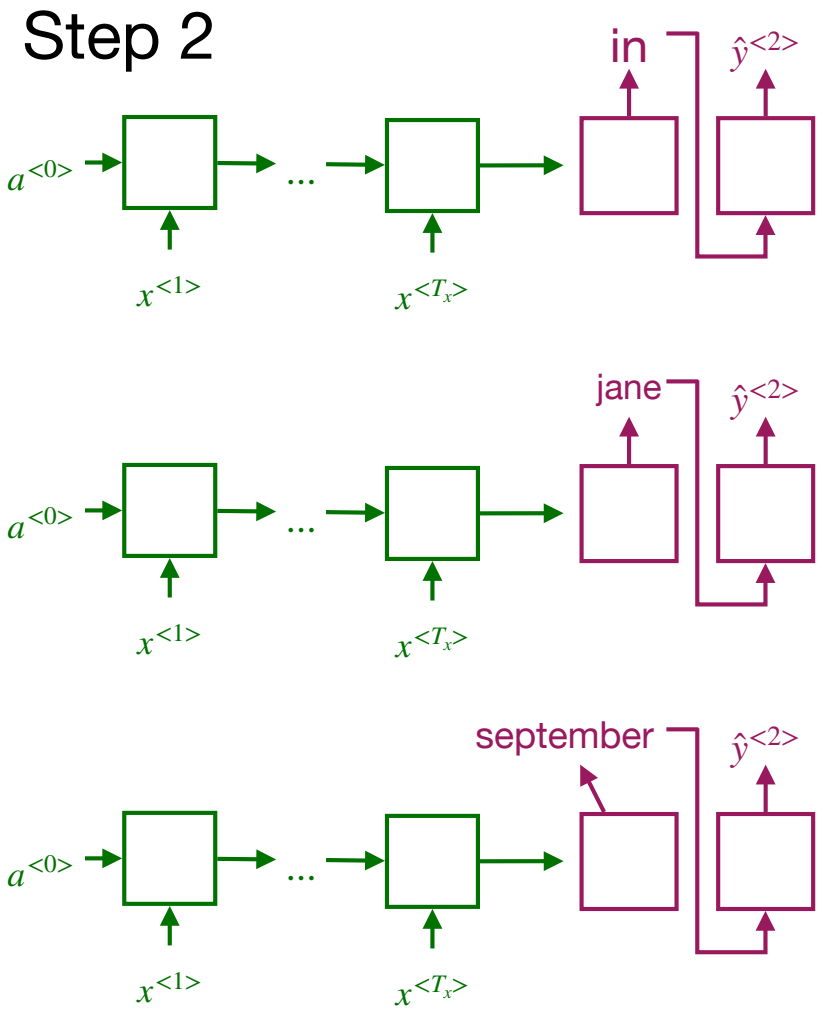


$$\mathbb{P}(y^{<1>} \quad x)$$

Beam width = 3

Beam search





$\mathbb{P}(y^{<2>} \quad x, \text{"in"})$

$\mathbb{P}(y^{<2>} \quad x, \text{"jane"})$

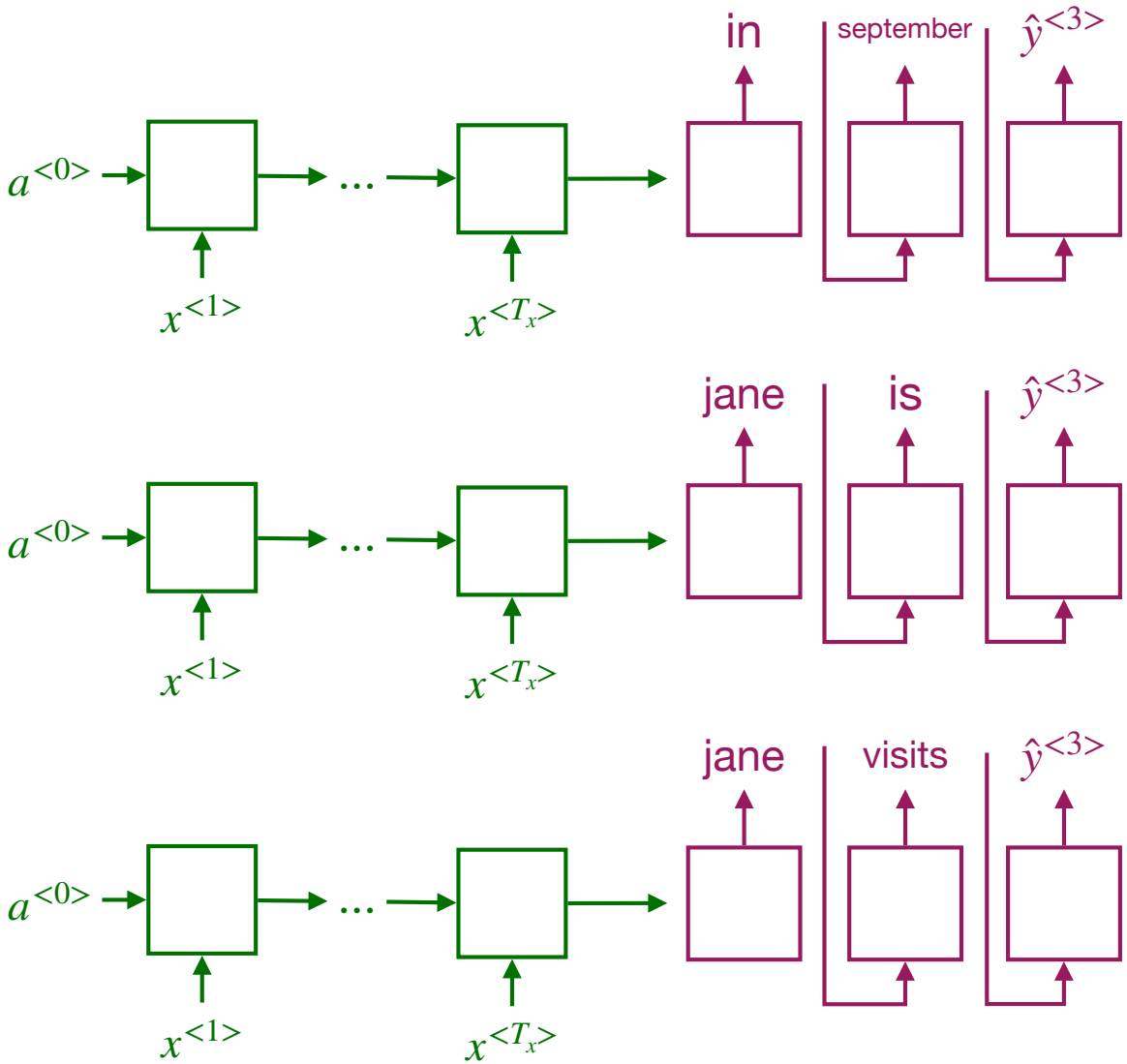
$\mathbb{P}(y^{<2>} \quad x, \text{"september"})$

Beam search

in september

jane in

jane visits



Practical issues

- Length normalization

$$\arg \max_{y^{<1>},\ldots,y^{<T_y>}} \mathbb{P}(y^{<1>},\ldots,y^{<T_y>} \mid x) = \arg\max_{y} \prod_{t=1}^{T_y} \mathbb{P}(y^{<t>} \mid x, y^{<1>},\ldots,y^{<y-1>})$$

$$\downarrow$$

$$\arg\max_{y} \sum_{t=1}^{T_y} \log \mathbb{P}(y^{<t>} \mid x, y^{<1>},\ldots,y^{<t-1>})$$

$$\downarrow$$

$$\frac{1}{T_y^{\alpha}} \sum_{t=1}^{T_y} \log \mathbb{P}(y^{<t>} \mid x, y^{<1>},\ldots,y^{<t-1>})$$

Practical issues

- Error analysis
  - > Jane visite l'Afrique en Septembre
  - > Human: Jane visits Africa in September.
  - > Algorithm: Jane visited Africa last September.

Practical issues

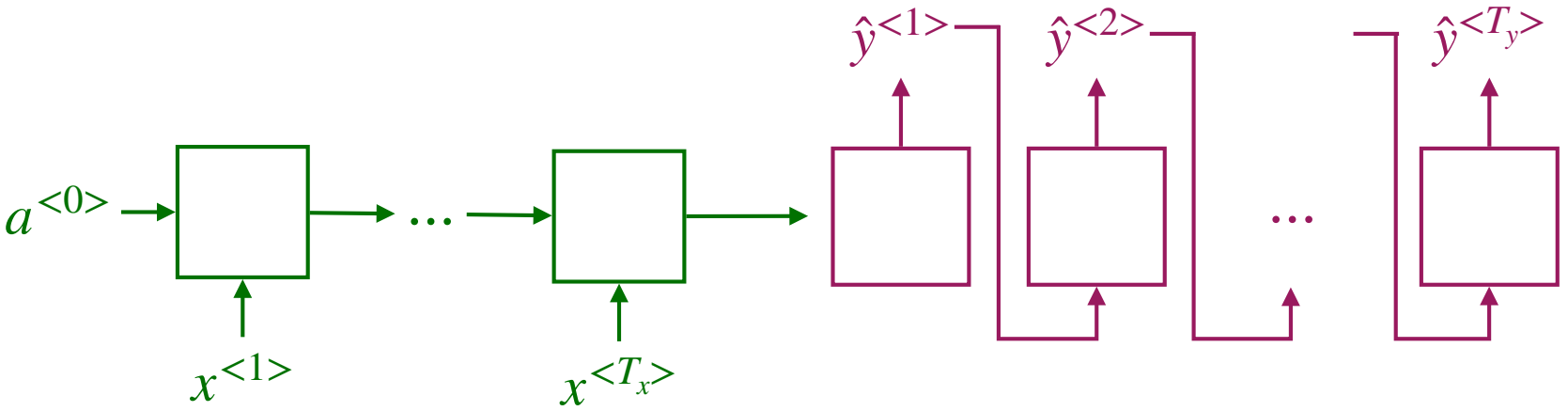
- Error analysis
  - Human: Jane visits Africa in September. (y^*)
  - Algorithm: Jane visited Africa last September. $(\hat{y})$
- Case 1: $\mathbb{P}(y^* \mid x) > \mathbb{P}(\hat{y} \mid x)$
  - Beam search chose $\hat{y}$. But, y^* attains higher $\mathbb{P}(y \mid x)$.
  - Conclusion: Beam search is at fault.
- Case 2: $\mathbb{P}(y^* \mid x) \leq \mathbb{P}(\hat{y} \mid x)$
  - y^* is a better translation than $\hat{y}$. But RNN predicted $\mathbb{P}(y^* \mid x) < \mathbb{P}(\hat{y} \mid x)$
  - Conclusion: RNN model is at fault.


Recent developments
Advanced Machine Learning

Attention mechanism

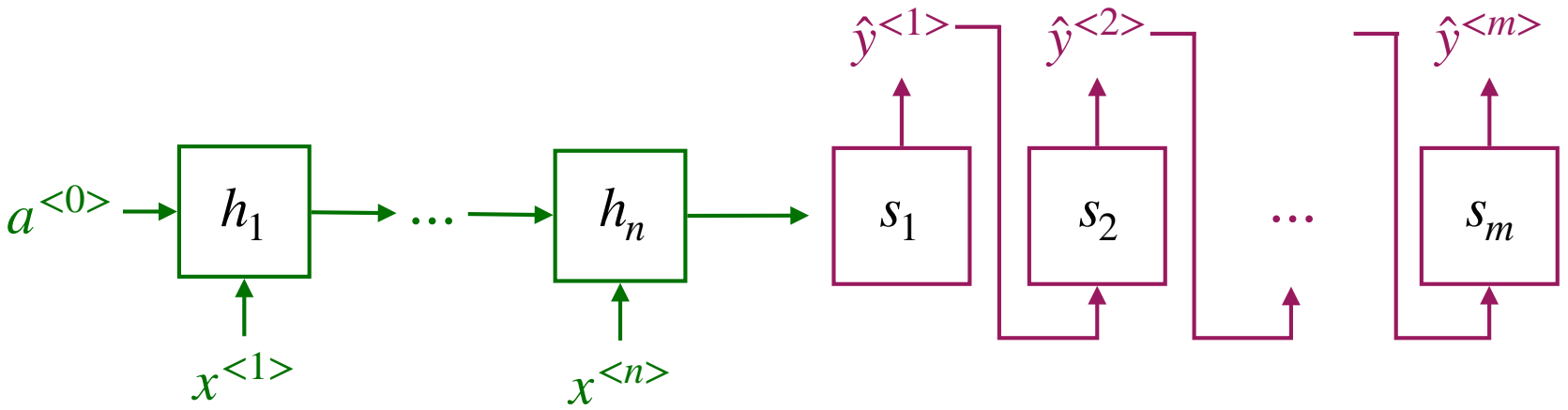
- Seq2Seq model (2014) → encoder-decoder representation

- Major drawback:
  - > Fixed length representation of LSTM cell, loss of memory
  - > Very hard to train



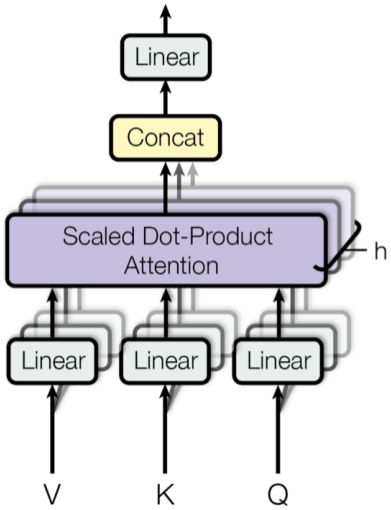
Attention mechanism

- Given an input $\mathbf{x} = [x_1, \ldots, x_n]$ and output $\mathbf{y} = [y_1, \ldots, y_m]$
- Define the context as $c_i = \sum_{i=1}^{n} \alpha_{t,i} h_i$
- The weights $\alpha_{t,i} = \text{align}(y_t, x_i) = \frac{\exp(\text{score}(s_{t-1}, h_i))}{\sum_{j=1}^{n} \exp(\text{score}(s_{t-1}, h_j))}$



Attention mechanism

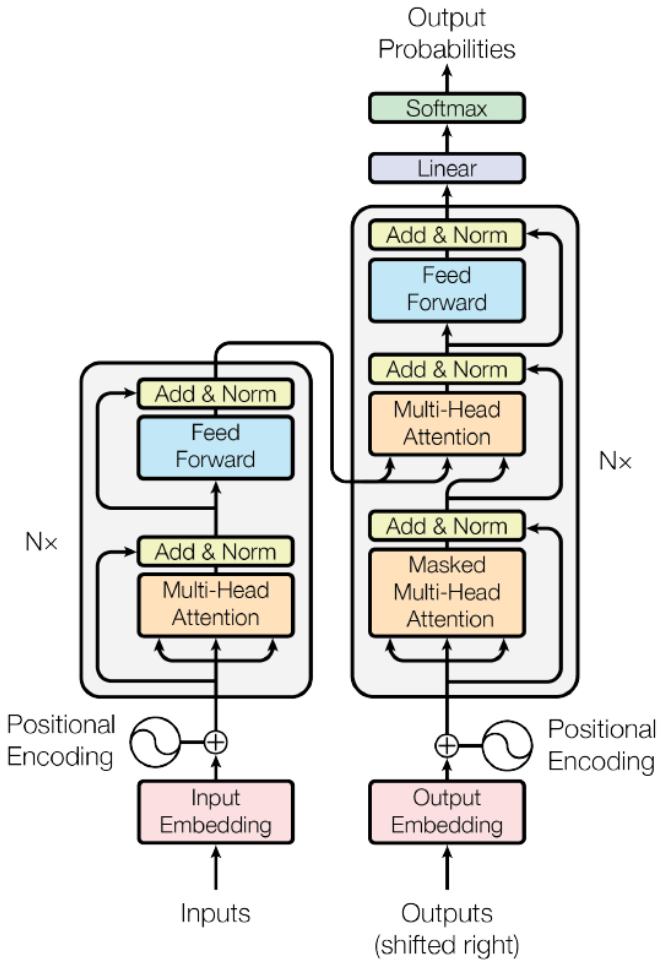
- Multi-head self attention (2017)
- Q(uery), K(ey) and V(alue) operators
- Attention is given by $\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{n}}\right)V$



Transformer (2017)

- Paper “Attention is all you need”
- Transformer is an architecture for transforming one sequence into another one
- Without any Recurrent Networks!

- Positional encoding takes over part of the RNN functionality



BERT (Google, 2018)

- Bidirectional Encoder Representations from Transformers
- 340 million parameters

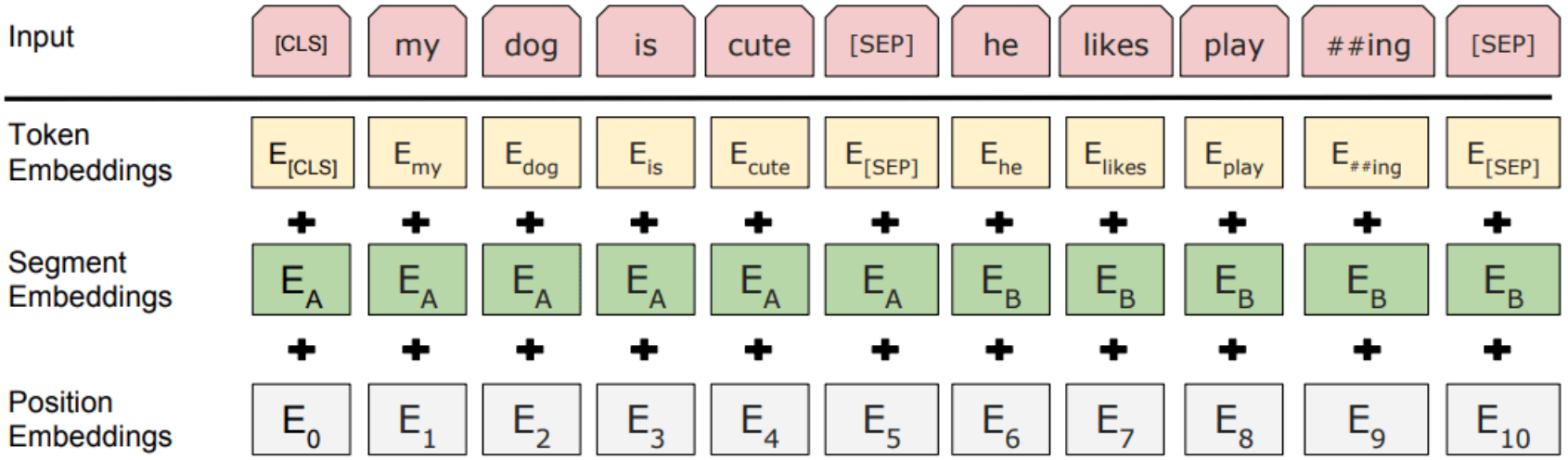
- Traditional models:
  - I want a glass of orange _____.
  - Trained with one-directional models
  - Word embeddings are context-free

- Innovation:
  - Bi-directional training (actually, non-directional)
  - Masked Language Models (Masked LM)

BERT (Google, 2018)

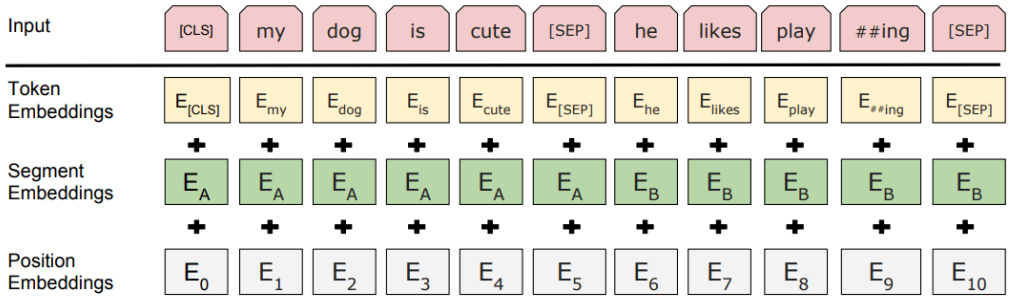
Input processing

- Token embeddings
- Segment embeddings
- Positional embeddings



BERT (Google, 2018)

- During training the model is fed with two input sentences at a time such that
  - 50% of the time, sentence comes after the first one
  - 50% of the time, random sentence from the full corpus.
- Masked LM (MLM): Randomly mask out 15% of the words in the input
- Next Sentence Prediction



GPT-3 (OpenAI, 2020)

- Generative Pretrained Transformer version 3
- GPT-3 uses 175 billion parameters

- The model can perform “few-shot,” “one-shot,” or “zero-shot” learning, i.e., no fine-tuning is necessary
- OpenAI has not disclosed details about it yet
- GPT-3 175B model required 3.14E23 FLOPS of computing for training
- training: at theoretical 28 TFLOPS for V100, this will take 355 GPU-years and cost $4.6M