CLASS. VISION

AI & DEEP LEARNING COURSES

حامیان دوره

همتک

تنا

صندوق نوآوری و شکوفایی

شبکه‌های بازگشتی
و پیاده سازی در Tensorflow2 و Keras

CLASS.VISION
AI & DEEP LEARNING COURSES

Alireza AkhavanPour

Akhavanpour.ir
CLASS.VISION

November 2019

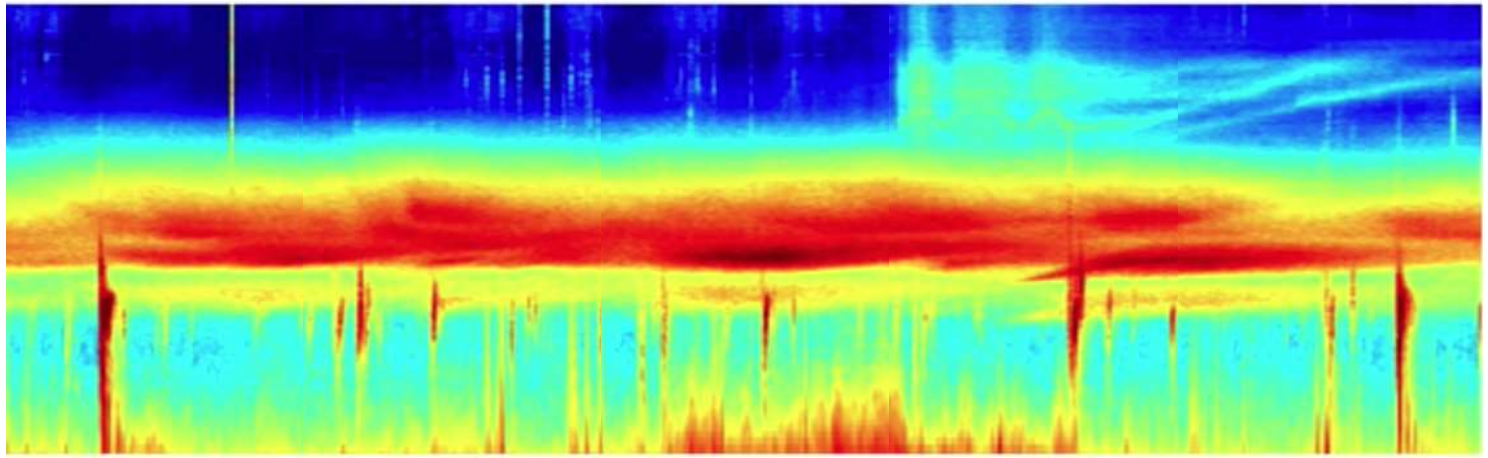
شبکه‌های بازگشتی – RNN:

- So far, all the DNNs that we have explored process training data with the assumption that **there is no relationship** between any two **training samples**.

مسیر توپ کدام سمت است؟

کاربردهای RNN:

صوت

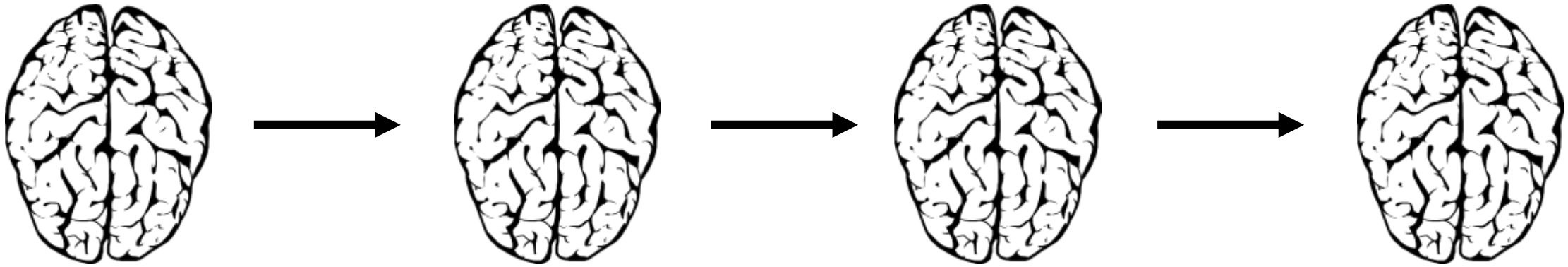


کاربردهای RNN:

م ی ت و ا ن ی م

می‌توانیم به متن مثل دنباله‌ای از داده‌ها نگاه کنیم

حافظه ترتیبی(Sequential Memory)



حافظه ترتیبی(Sequential Memory)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

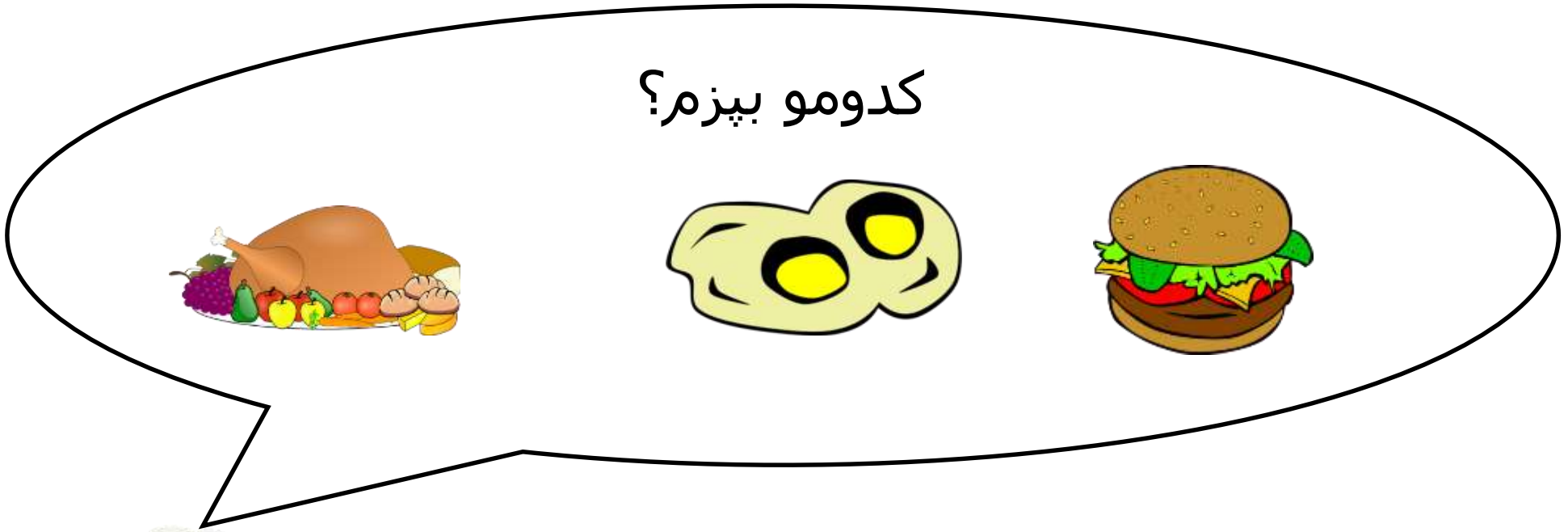
حافظه ترتیبی(Sequential Memory)

Z Y X W V U T S R Q P O N M L K J I H G F E D C B A

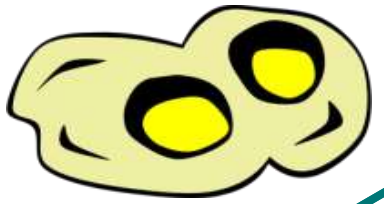
حافظه ترتیبی(Sequential Memory)

F G H I J K L M N O P Q R S T U V W X Y Z

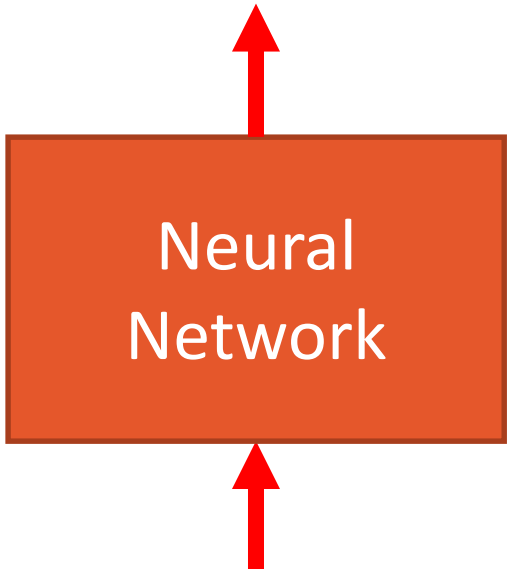
شبکه‌های بازگشتی – مقدمات:



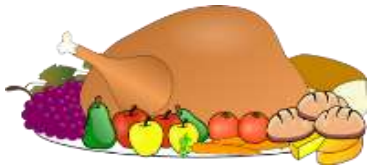
شبکه‌های بازگشتی – مقدمات:

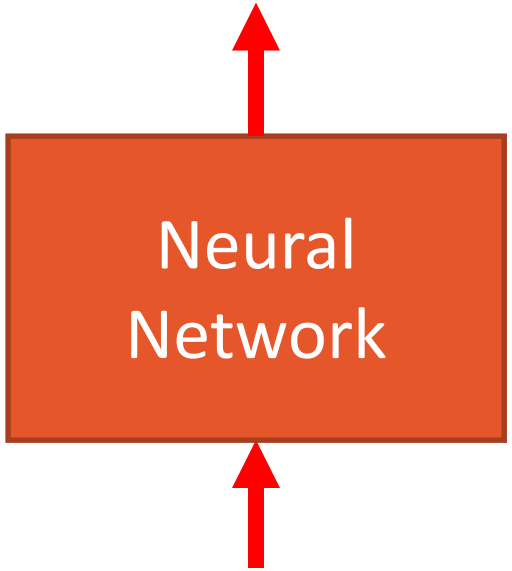


شبکه‌های بازگشتی – مقدمات:

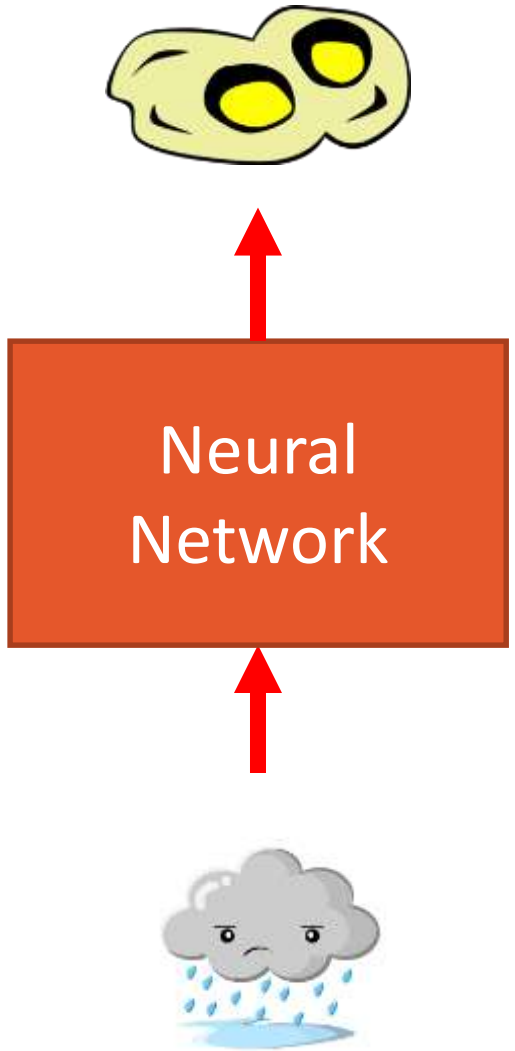


شبکه‌های بازگشتی – مقدمات:



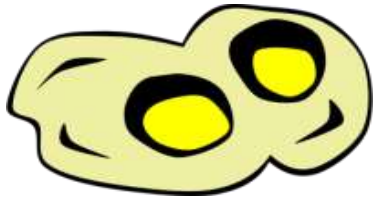


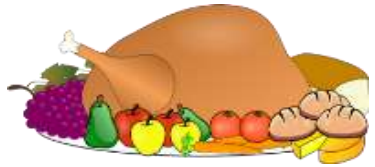
شبکه‌های بازگشتی – مقدمات:



شبکه‌های بازگشتی – مقدمات:

$\begin{bmatrix}0\\0\\1\end{bmatrix}$ 

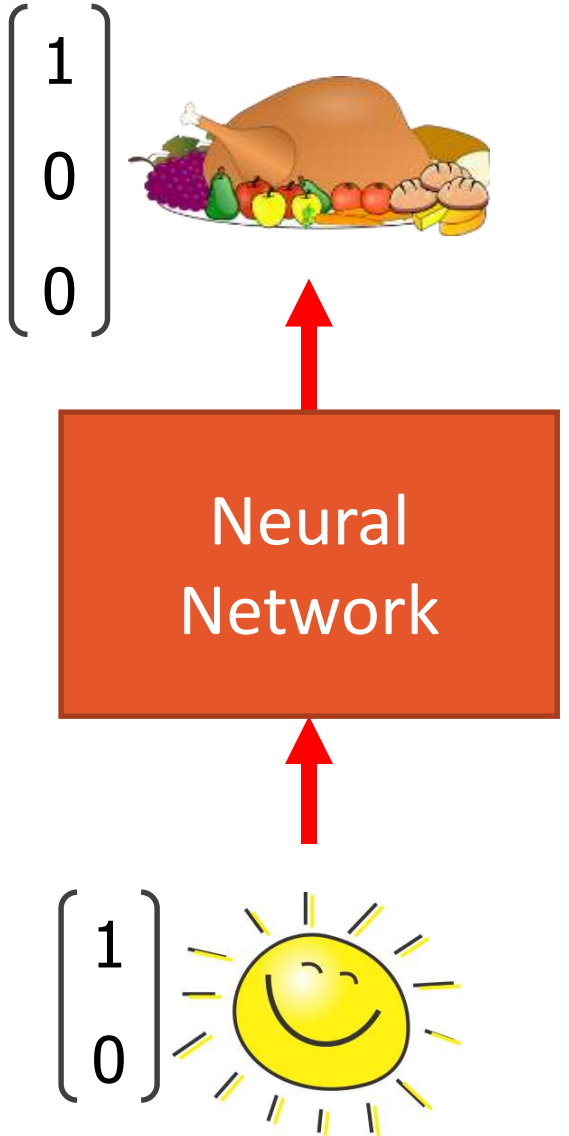
$\begin{bmatrix}0\\1\\0\end{bmatrix}$ 

$\begin{bmatrix}1\\0\\0\end{bmatrix}$ 

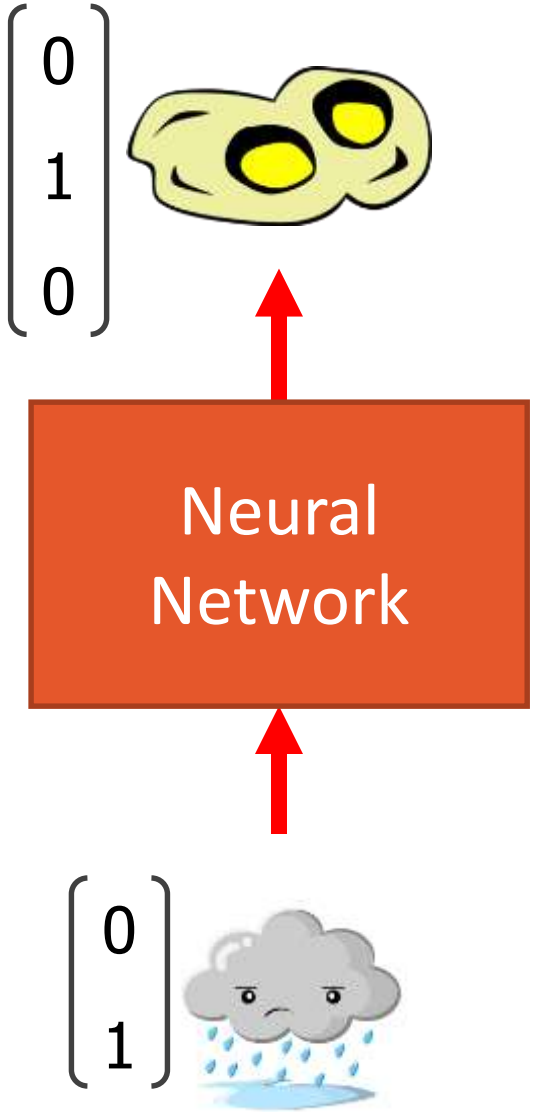
$\begin{bmatrix}0\\1\end{bmatrix}$ 

$\begin{bmatrix}1\\0\end{bmatrix}$ 

شبکه‌های بازگشتی – مقدمات:



شبکه‌های بازگشتی – مقدمات:



شبکه‌های بازگشتی – مقدمات:

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

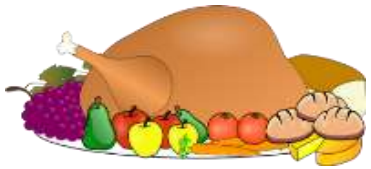
$$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

شبکه‌های بازگشتی – مقدمات:

$$\begin{bmatrix}1\\0\end{bmatrix}$$ 

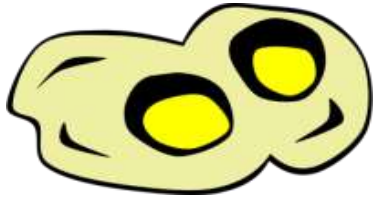
$$\begin{bmatrix}0\\1\end{bmatrix}$$ 

$$\begin{bmatrix}1 & 0\\0 & 1\\0 & 0\end{bmatrix} \begin{bmatrix}1\\0\end{bmatrix}$$  $$=$$ $$\begin{bmatrix}1\\0\\0\end{bmatrix}$$ 

شبکه‌های بازگشتی – مقدمات:

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$ 

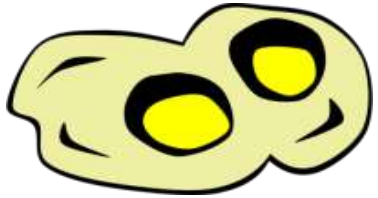
$$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$ 

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$  $=$ $$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$ 

شبکه‌های بازگشتی – مقدمات:

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$ 

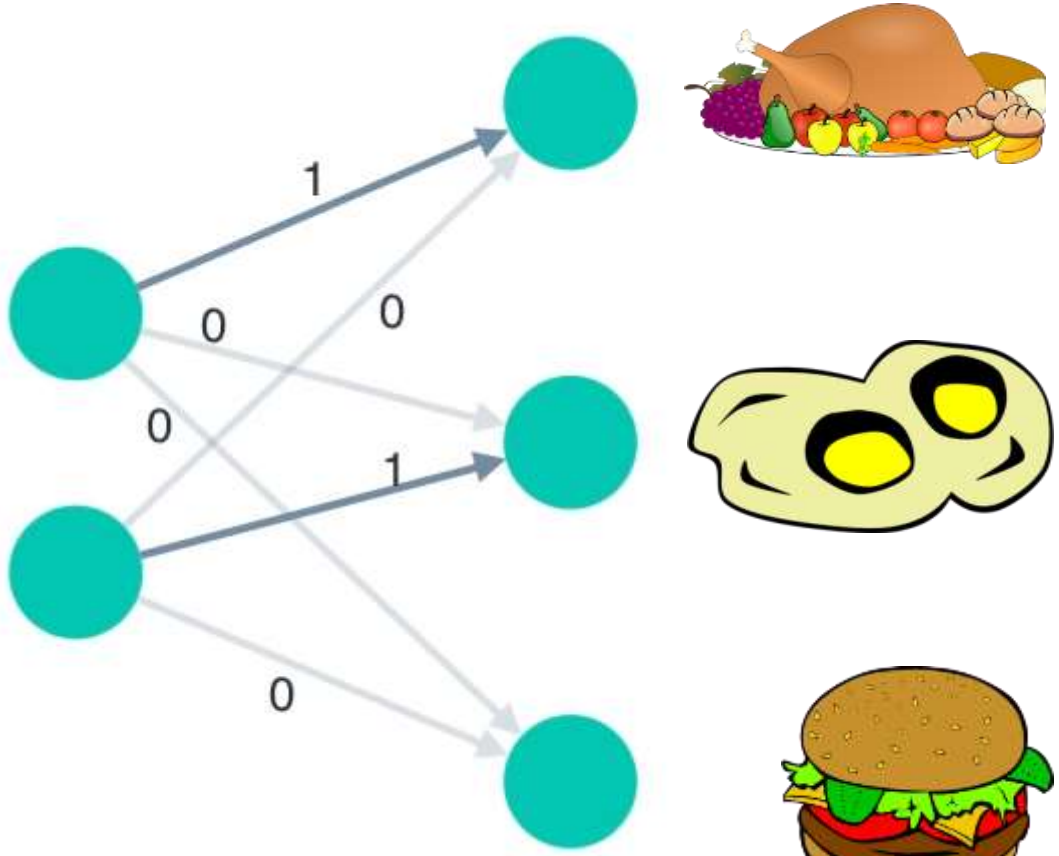
$$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$ 

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$  $=$ $$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$ 

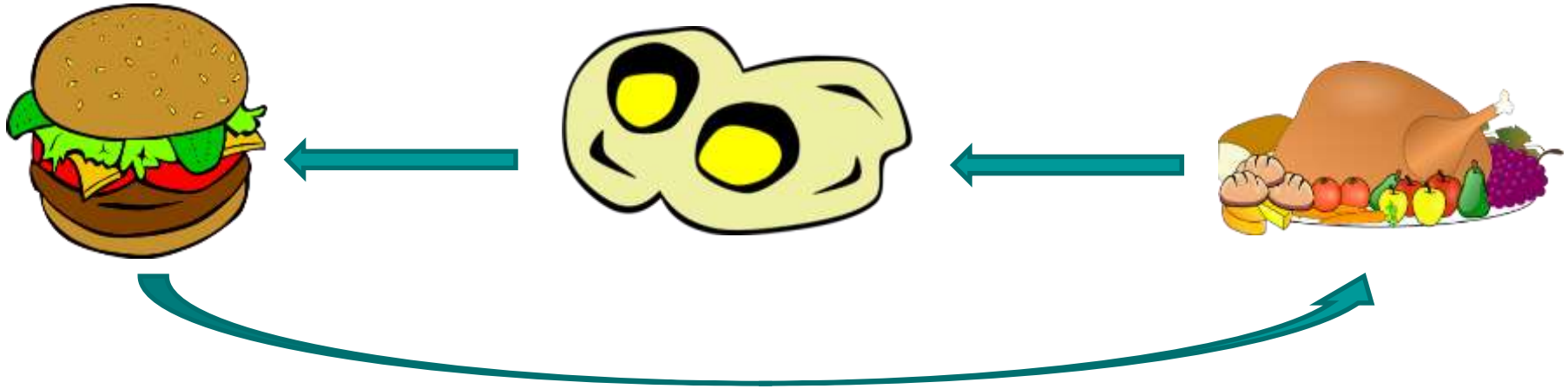
شبکه‌های بازگشتی – مقدمات:

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$



شبکه‌های بازگشتی – مقدمات:



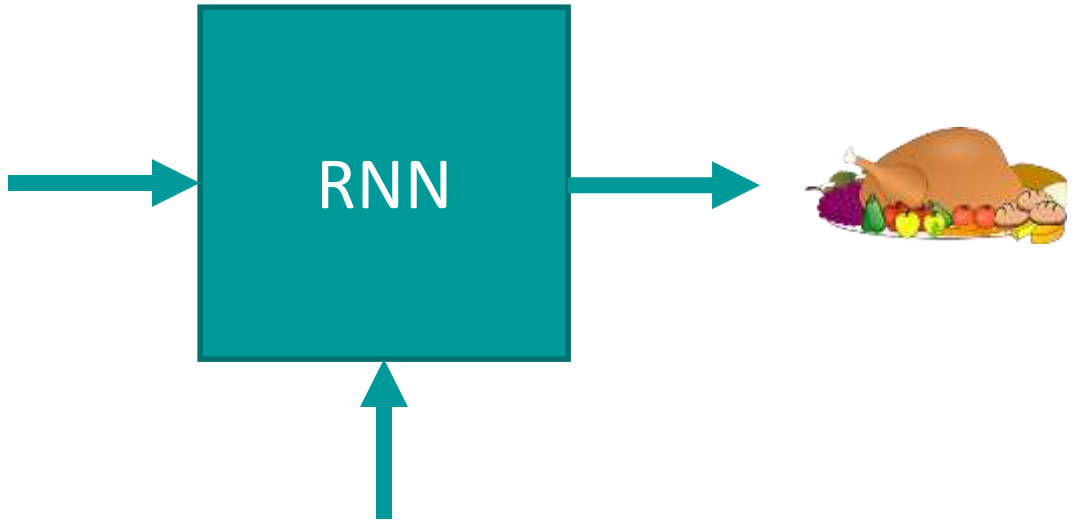
شبکه‌های بازگشتی – مقدمات:

شنبه یکشنبه دوشنبه سه‌شنبه چهارشنبه پنج شنبه

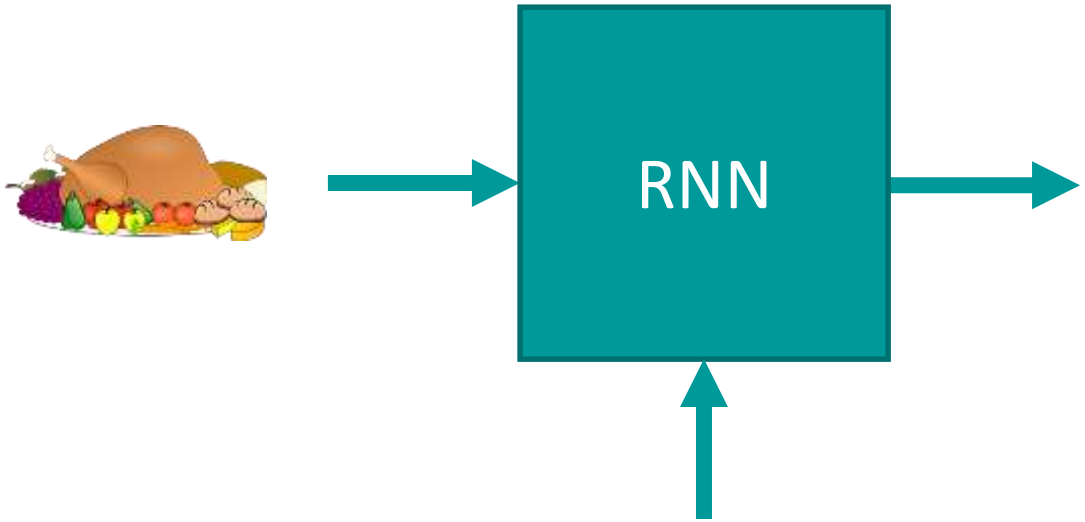




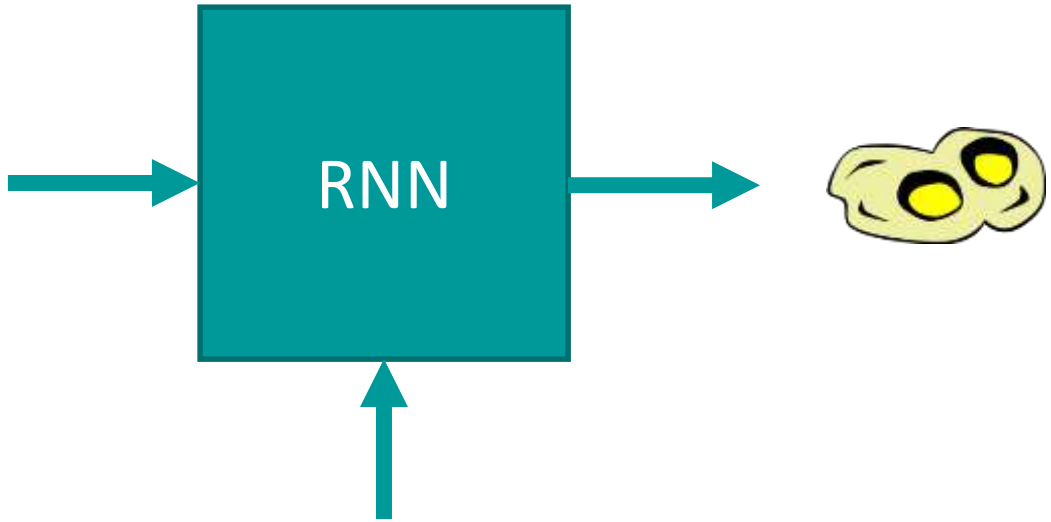

شبکه‌های بازگشتی – مقدمات:



شبکه‌های بازگشتی – مقدمات:



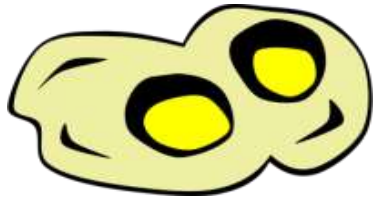
شبکه‌های بازگشتی – مقدمات:



شبکه‌های بازگشتی – مقدمات:

شبکه‌های بازگشتی – مقدمات:

$$\begin{bmatrix}1\\0\\0\end{bmatrix}$$ 

$$\begin{bmatrix}0\\1\\0\end{bmatrix}$$ 

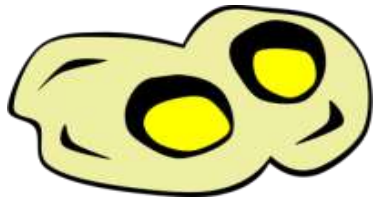
$$\begin{bmatrix}0\\0\\1\end{bmatrix}$$ 

$$=$$

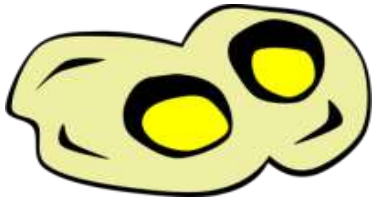
$$\begin{bmatrix}0 & 0 & 1\\1 & 0 & 0\\0 & 1 & 0\end{bmatrix}$$

شبکه‌های بازگشتی – مقدمات:

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

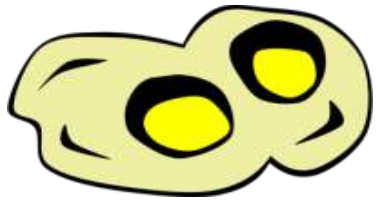
$$\begin{pmatrix}0\\1\\0\end{pmatrix}$$ 

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

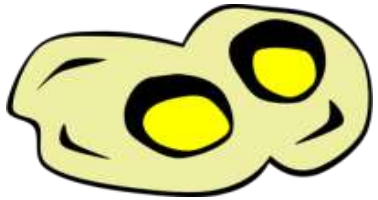
$$\begin{pmatrix}0&0&1\\1&0&0\\0&1&0\end{pmatrix}\begin{pmatrix}1\\0\\0\end{pmatrix}$$  $$=$$ $$\begin{pmatrix}0\\1\\0\end{pmatrix}$$ 

شبکه‌های بازگشتی – مقدمات:

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

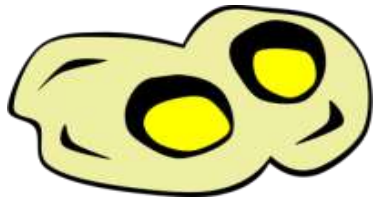
$$\begin{pmatrix}0\\1\\0\end{pmatrix}$$ 

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

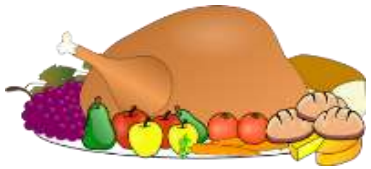
$$\begin{pmatrix}0&0&1\\1&0&0\\0&1&0\end{pmatrix}\begin{pmatrix}0\\1\\0\end{pmatrix}$$  $$=$$ $$\begin{pmatrix}0\\0\\1\end{pmatrix}$$ 

شبکه‌های بازگشتی – مقدمات:

$$\begin{bmatrix}1\\0\\0\end{bmatrix}$$ 

$$\begin{bmatrix}0\\1\\0\end{bmatrix}$$ 

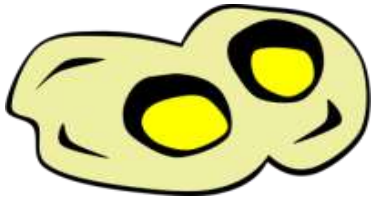
$$\begin{bmatrix}0\\0\\1\end{bmatrix}$$ 

$$\begin{bmatrix}0 & 0 & 1\\1 & 0 & 0\\0 & 1 & 0\end{bmatrix} \begin{bmatrix}0\\0\\1\end{bmatrix}$$  $$=$$ $$\begin{bmatrix}1\\0\\0\end{bmatrix}$$ 

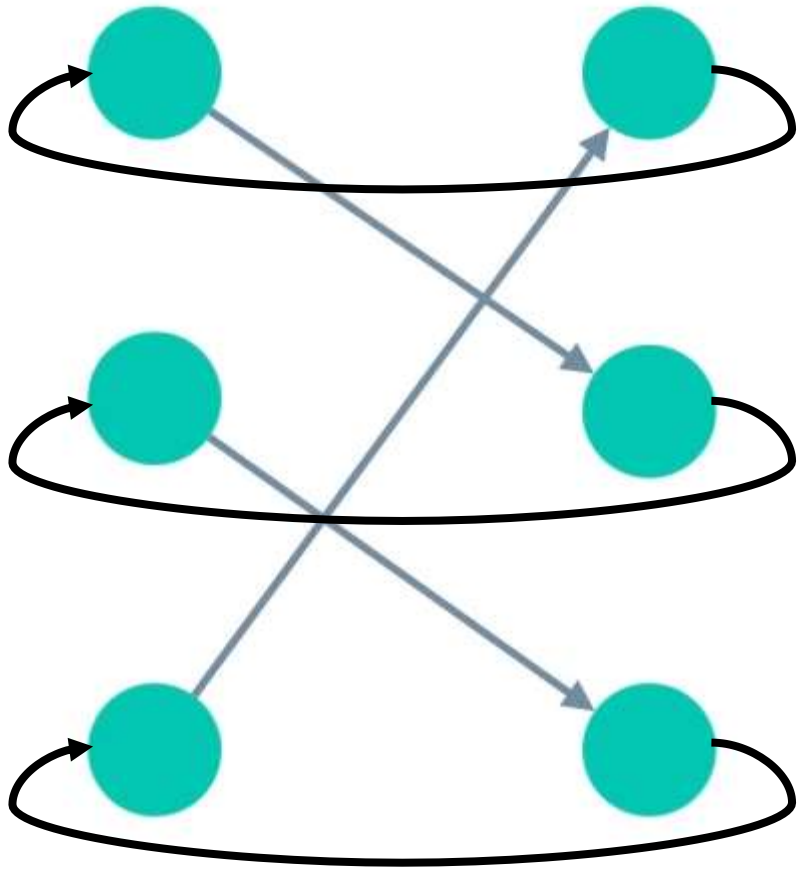
شبکه‌های بازگشتی – مقدمات:

$$\begin{bmatrix}1\\0\\0\end{bmatrix}$$

$$\begin{bmatrix}0\\1\\0\end{bmatrix}$$



$$\begin{bmatrix}0\\0\\1\end{bmatrix}$$

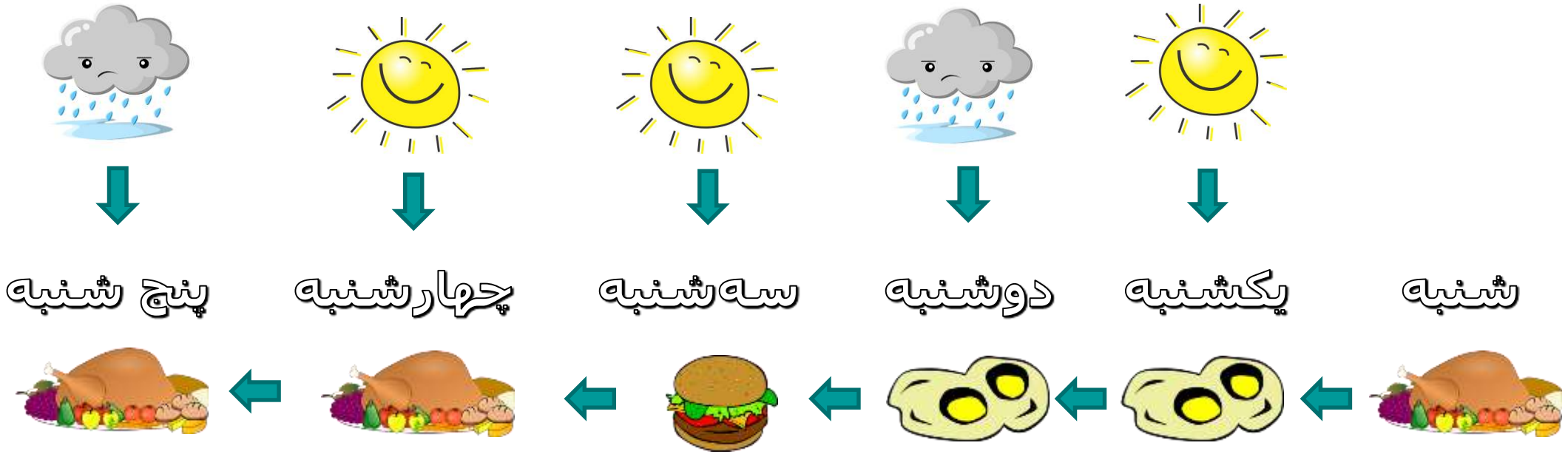


شبکه‌های بازگشتی – مقدمات:

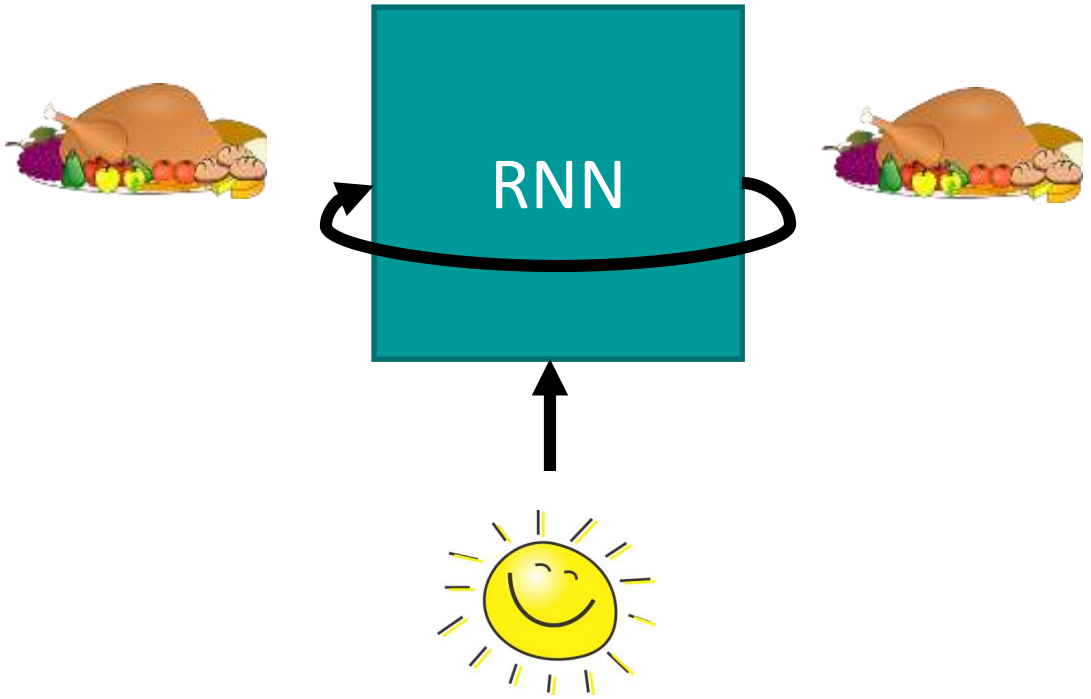
نمی‌تونم برم بیرون ☹
همون دیروزی را بخوریم

طبق لیست غذای بعدی را می‌پزم ☺

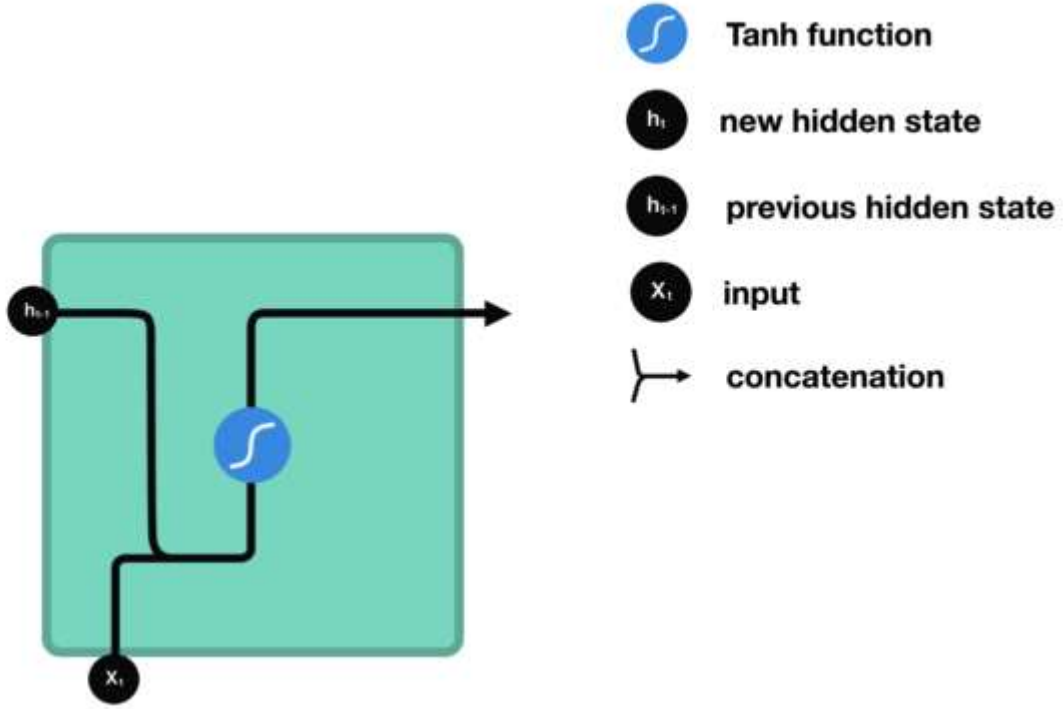
شبکه‌های بازگشتی – مقدمات:



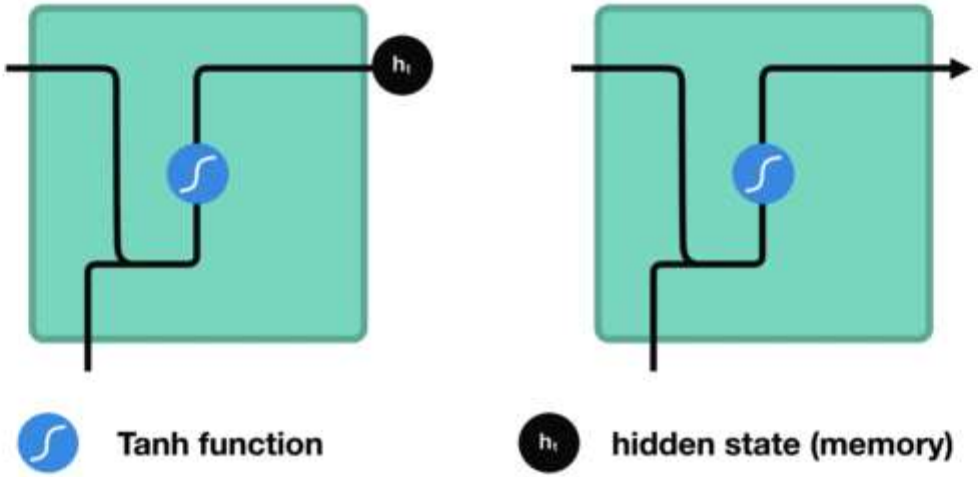
شبکه‌های بازگشتی – مقدمات:



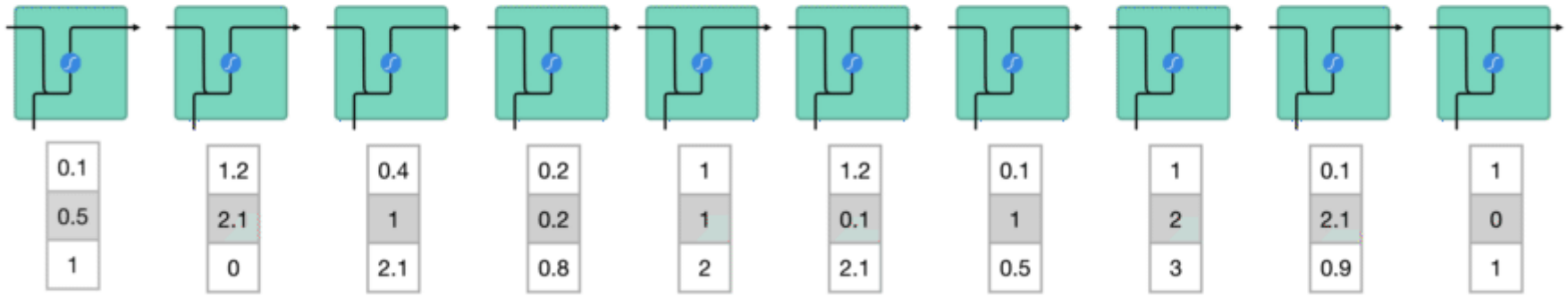
شبکه‌های بازگشتی – مقدمات:



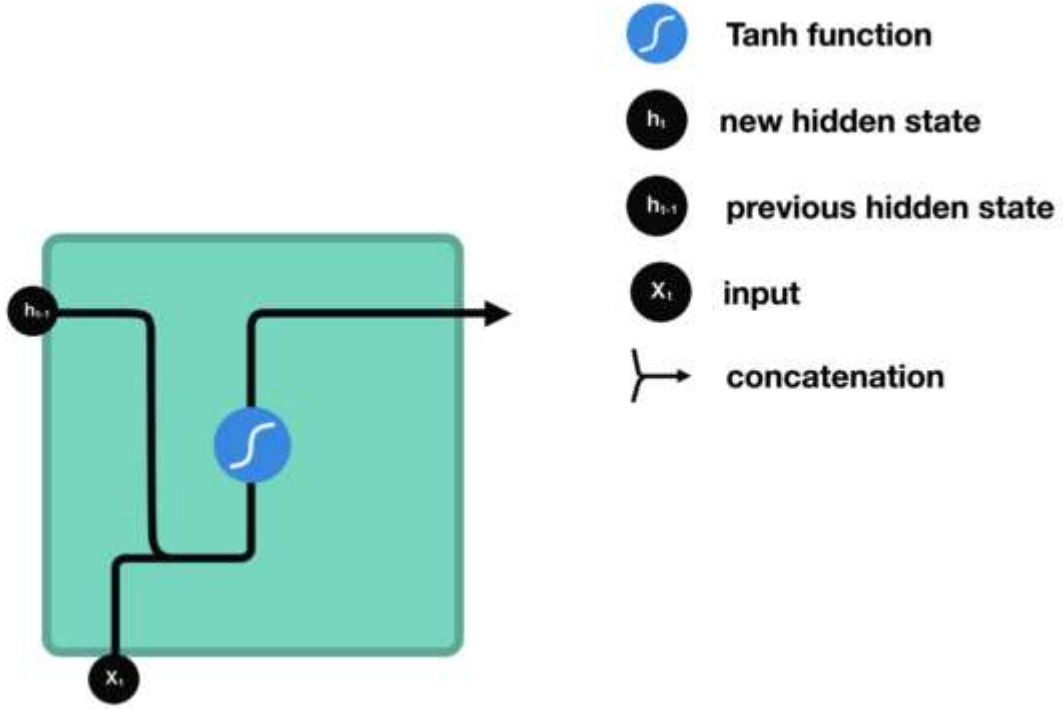
شبکه‌های بازگشتی – مقدمات:



شبکه‌های بازگشتی – مقدمات:

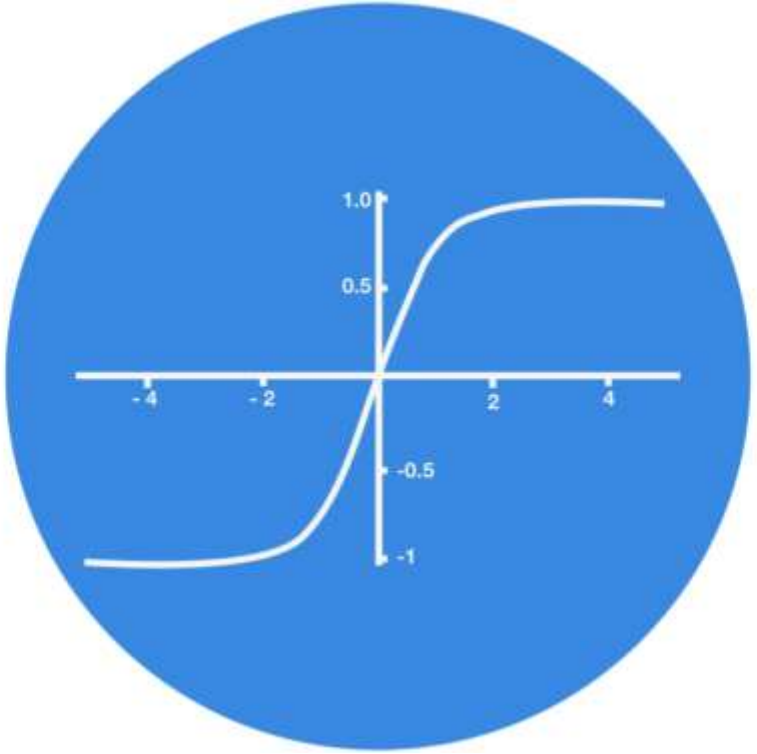


شبکه‌های بازگشتی – مقدمات:



شبکه‌های بازگشتی – مقدمات:

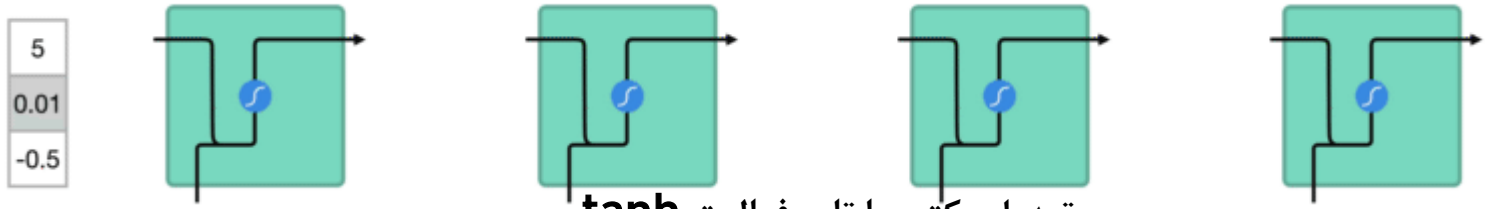
5
0.1
-0.5



شبکه‌های بازگشتی – مقدمات:

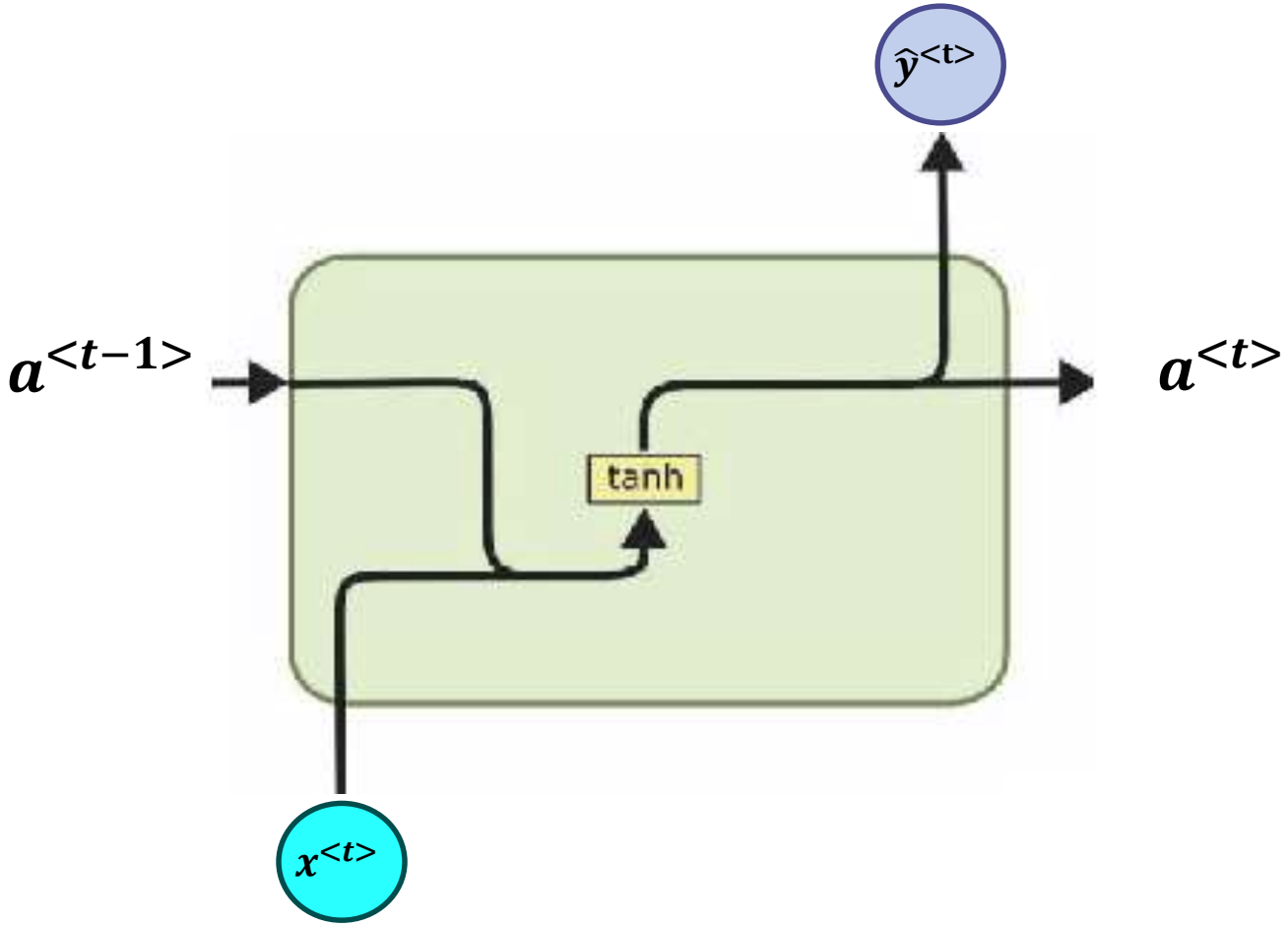
- فرض کنید که از تابع‌فعالیت **tanh** استفاده نمی‌کردیم!
- فرض کنید هر بار مقادیر در ۳ ضرب می‌شدند. مشکلی نبود؟!

تبدیل وکتور بدون تابع فعالیت

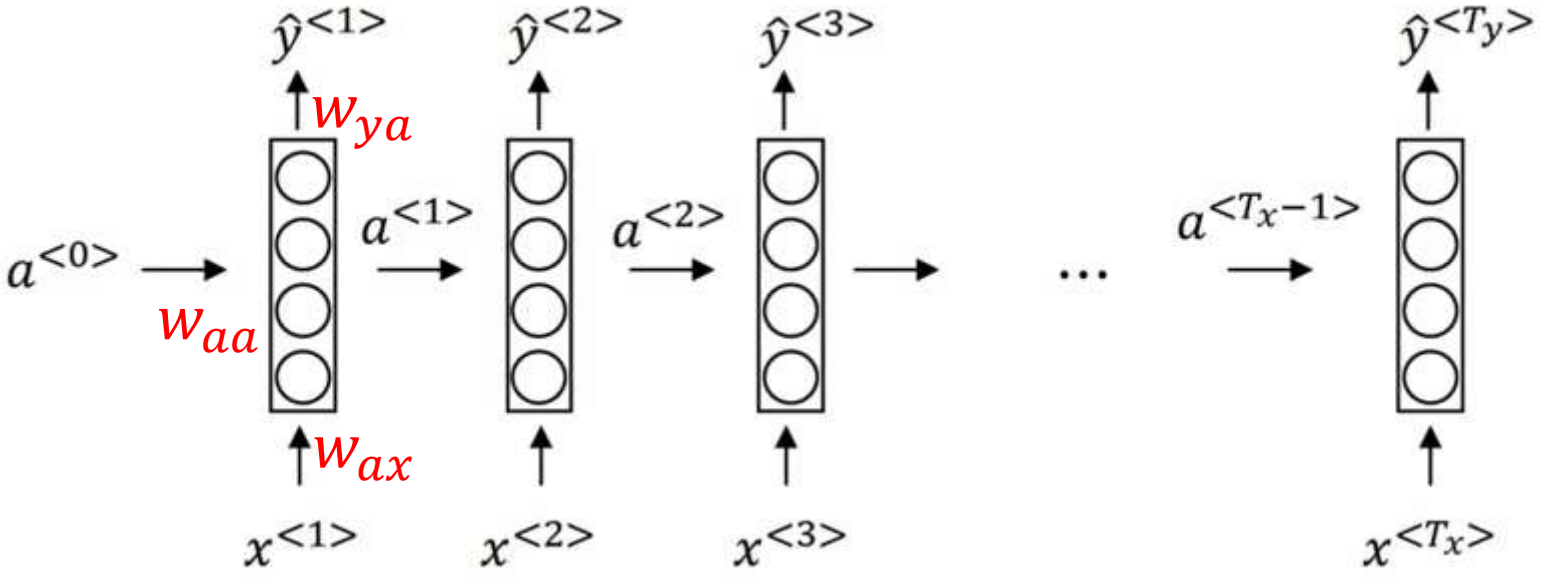


تبدیل وکتور با تابع فعالیت **tanh**

شبکه‌های بازگشتی – مقدمات:



Forward propagation



$$a^{<t>} = g_1(w_{aa}a^{<t-1>} + w_{ax}x^{<t>} + b_a)$$

$$\hat{y}^{<t>} = g_2(w_{ya}a^{<t>} + b_y)$$

Simplified RNN notation

$$a^{<t>} = g_1(w_{aa}a^{<t-1>} + w_{ax}x^{<t>} + b_a)$$

100 — 10,000

(100, 100) — (100, 10,000)

$$\hat{y}^{<t>} = g_2(w_{ya}a^{<t>} + b_y)$$

بازنویسی به صورت ساده تر:

$$a^{<t>} = g_1(w_a\,[a^{<t-1>}, x^{<t>}] + b_a)$$

$$\hat{y}^{<t>} = g_2(w_y a^{<t>} + b_y)$$

$$\begin{matrix} 100 \\ \downarrow \end{matrix} \begin{pmatrix} w_{aa} & w_{ax} \end{pmatrix} = w_a$$

100 — 10,000

(100, 10,100)

$$[a^{<t-1>}, x^{<t>}] = \begin{pmatrix} a^{<t-1>} \\ x^{<t>} \end{pmatrix}$$

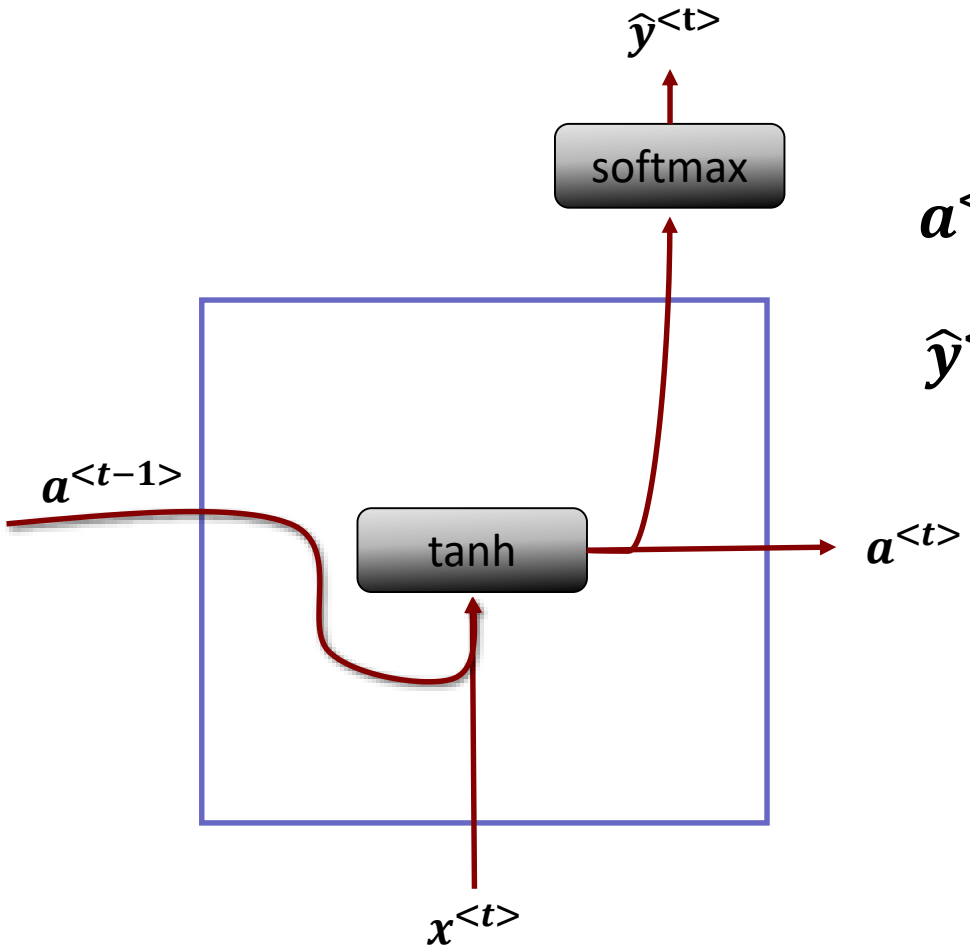
100 — 10,000 — 10,100

Simplified RNN notation

$$a^{<t>} = g_1(w_a \left[a^{<t-1>}, x^{<t>}\right] + b_a)$$
$$\hat{y}^{<t>} = g_2(w_y a^{<t>} + b_y)$$

- w_a is w_{aa} and w_{ax} stacked horizontally.
- $[a^{<t-1>}, x^{<t>}]$ is $a^{<t-1>}$ and $x^{<t>}$ stacked vertically.
- w_a shape: (NoOfHiddenNeurons, NoOfHiddenNeurons + n_x)
- $[a^{<t-1>}, x^{<t>}]$ shape: (NoOfHiddenNeurons + n_x, 1)

یک واحد RNN ساده



$$a^{<t>} = g_1(w_a\left[a^{<t-1>}, x^{<t>}\right] + b_a)$$

$$\hat{y}^{<t>} = g_2(w_y a^{<t>} + b_y)$$

تخمین تابع با RNN ساده

01-simple-RNN.ipynb

مزایای RNN ساده

- نسبت به مدل‌های پیچیده تر نظیر **LSTM** و **GRU** و ... تعداد پارامتر بسیار کمتر دارد.
- در جای مناسب خود (مثل سری‌های کوتاه) دقت خوبی دارد!

معایب RNN ساده

- شبکه‌های **RNN** ساده از حافظه کوتاه‌مدت رنج می‌برند!

- اگر یک دنباله به اندازه کافی طولانی باشد، آنها نمی‌توانند اطلاعات زیادی را از مراحل زمانی خیلی قبل‌تر انتقال دهند.

معایب RNN ساده – محو شدگی گرادیان‌ها

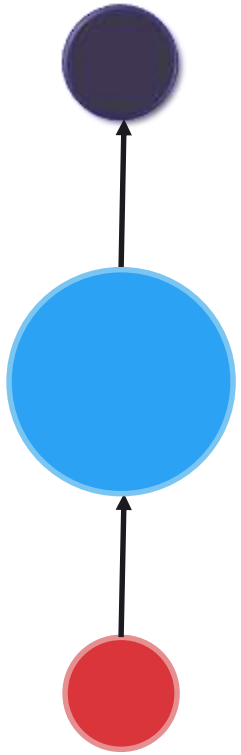
new weight = weight - learning rate*gradient

2.0999 = 2.1 - 0.001

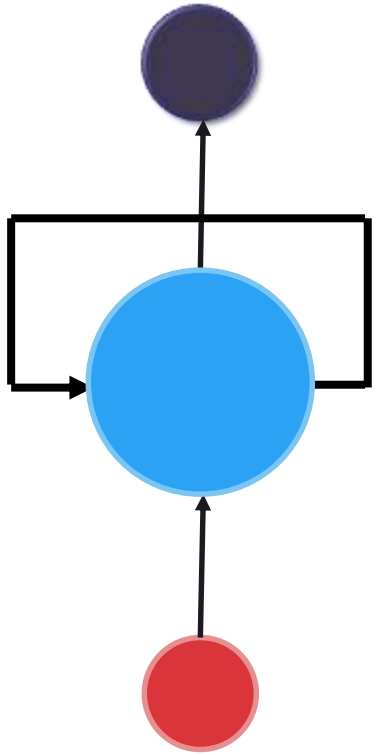
Not much of a difference

update value

معایب RNN ساده – محو شدگی گرادیان‌ها



Feed Forward Neural Network

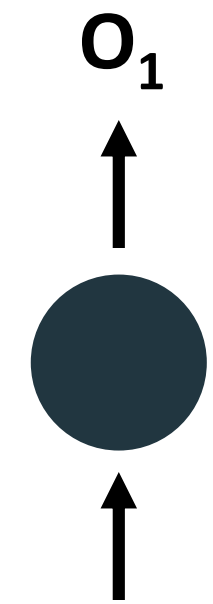


Recurrent Neural Network

معایب RNN ساده – محو شدگی گرادیان‌ها

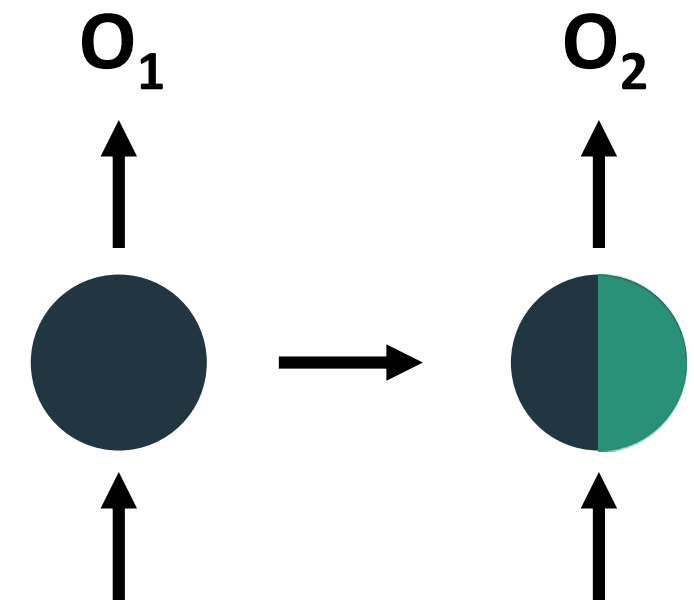
What time is it ?

معایب RNN ساده – محو شدگی گرادیان‌ها



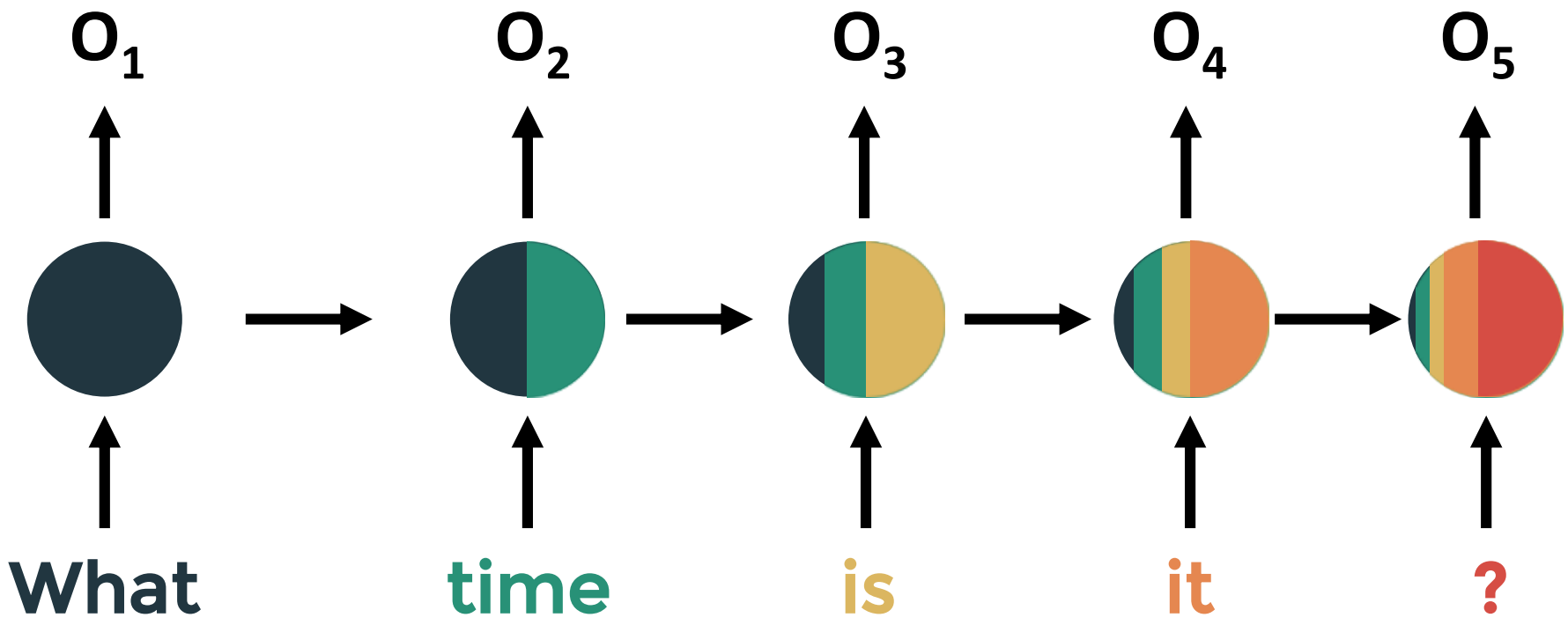
What time is it ?

معایب RNN ساده – محو شدگی گرادیان‌ها

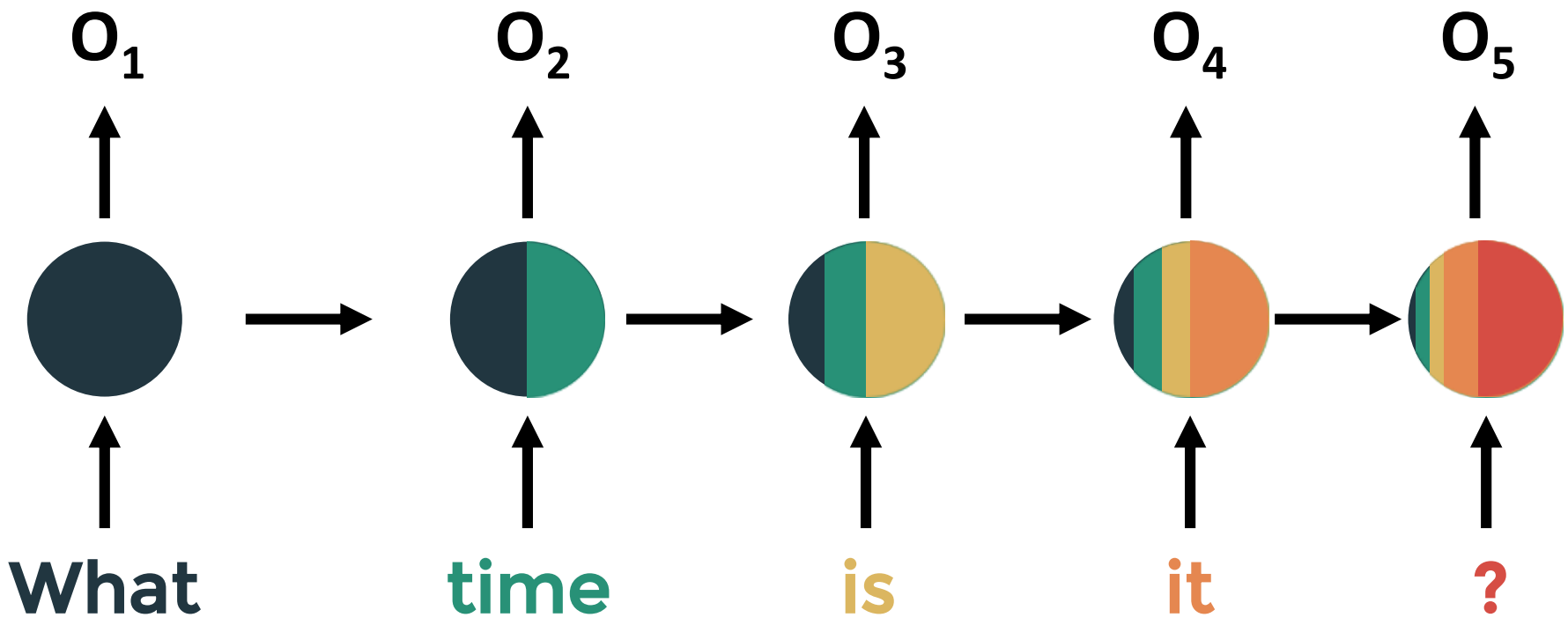


What time is it ?

معایب RNN ساده – محو شدگی گرادیان‌ها



معایب RNN ساده – محو شدگی گرادیان‌ها



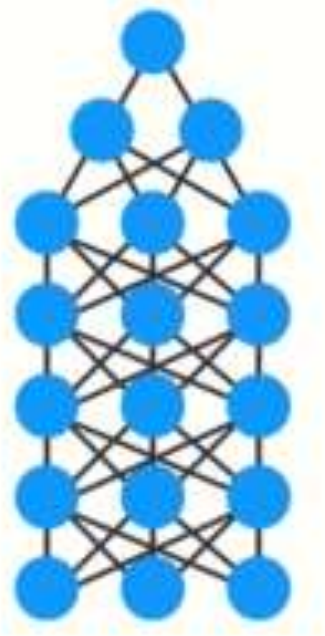
معایب RNN ساده – محو شدگی گرادیان‌ها

- محو شدگی گرادیان (Vanishing Gradient)

معایب RNN ساده – محو شدگی گرادیان‌ها

سه گام آموزش یک شبکه عصبی

1. پیش‌بینی در مسیر forward pass
2. مقایسه پیش‌بینی با ground truth با تابع loss
3. استفاده از مقدار خطا برای انجام back propagation و محاسبه گرادیان‌های هر گره در شبکه

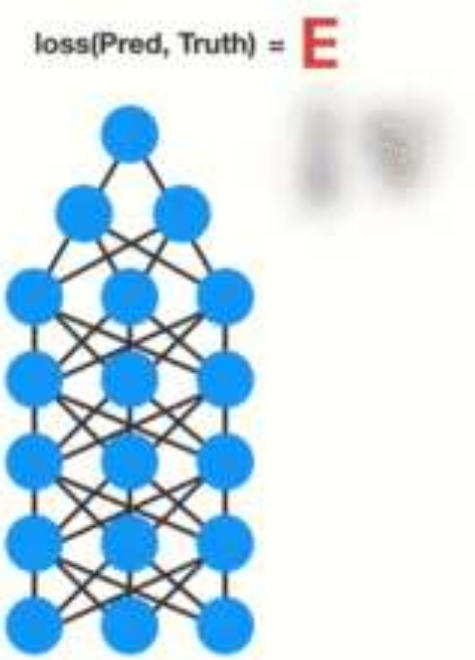


معایب RNN ساده – محو شدگی گرادیان‌ها

شیب یا گرادیان

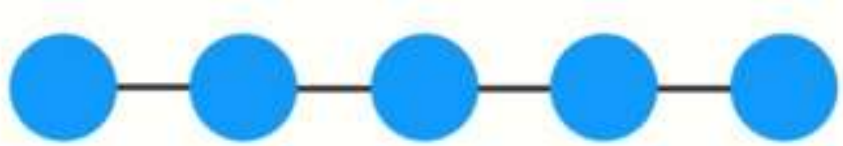
- ❖ برای تنظیم وزن‌های شبکه استفاده می‌شود
- ❖ و به شبکه امکان یادگیری می دهد

هرچه شیب بزرگتر باشد ، تغییرات وزن‌ها نیز بیشتر است...



و اما مشکل؟!

معایب RNN ساده – محو شدگی گرادیان‌ها



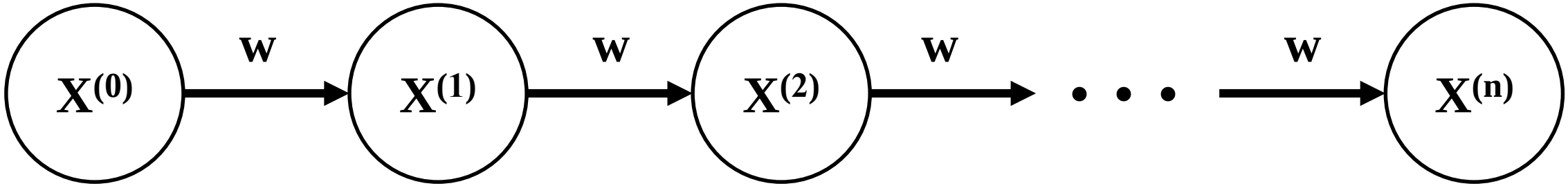
به **ایران** سفر کرده بودم،
مردمان مهمان نوازی داشت،
دوستانی پیدا کردم و آن‌ها اماکن مختلف تاریخی کشورشان را به من نشان دادند،
...
در انتهای سفر زبان ____?____ را به خوبی یاد گرفته بودم.

عربی
فارسی
انگلیسی

معایب RNN ساده – محو شدگی گرادیان‌ها

چه قدر طول دنباله یا Sequence می‌تواند طولانی باشد؟

از لحاظ تئوری؟! بی نهایت!



$$x^{(n)} = W^n x^{(0)}$$

$$x^{(i)}, W \in \mathbb{R} \qquad i \in [0, n]$$

$$x^{(i)} \epsilon \mathbb{R}^D \qquad W \epsilon \mathbb{R}^{D \times D}$$

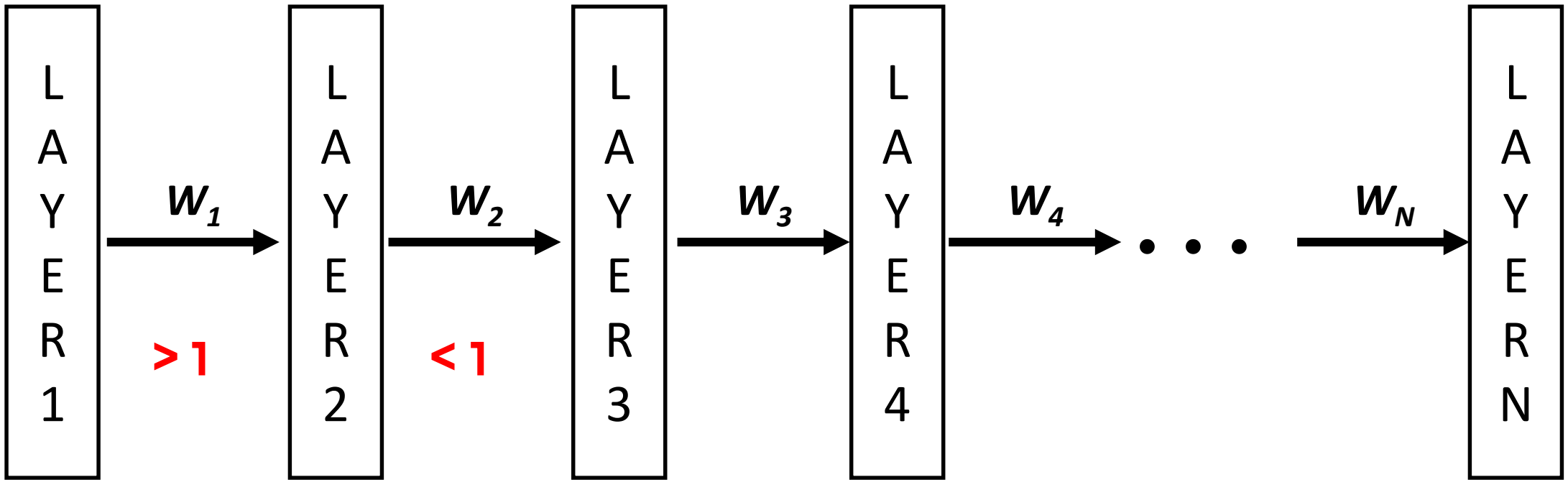
$$W^n x^{(0)} \rightarrow \begin{cases} \infty; W > 1 \\ 0; W < 1 \end{cases}$$

$$\frac{\delta W^n x^{(0)}}{\delta W} \rightarrow \begin{cases} \infty; W > 1 \\ 0; W < 1 \end{cases}$$

معایب RNN ساده – محو شدگی گرادیان‌ها

در DNN معمولی مگر مشکل (Vanishing / Exploding) Gradient نیست؟!

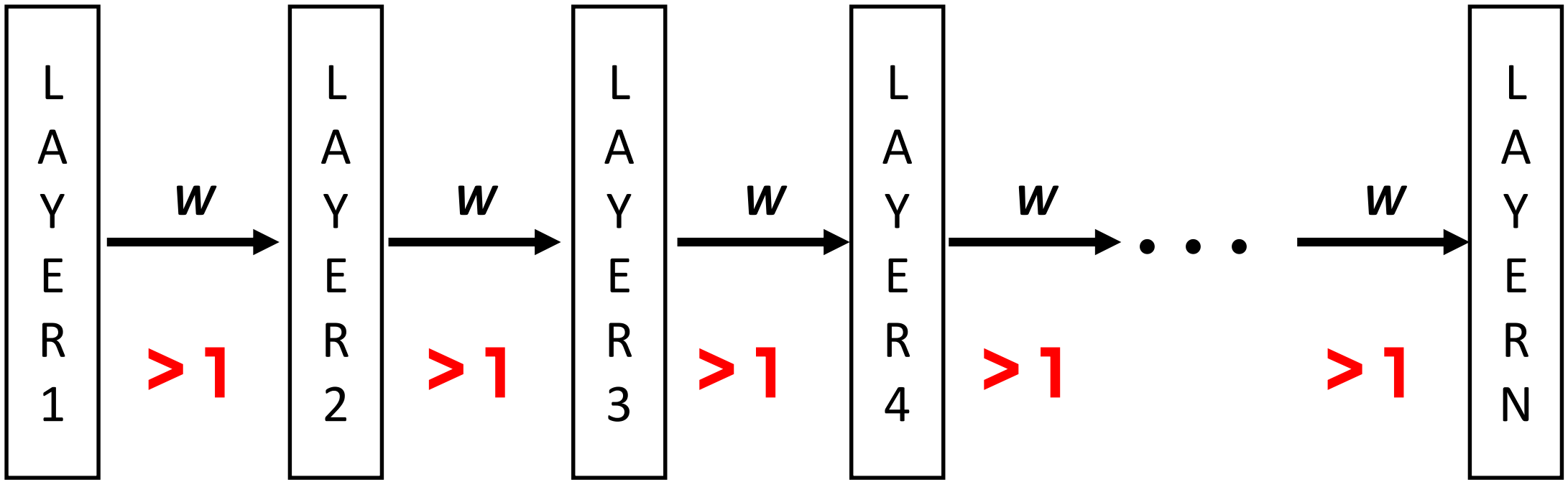
در RNNها به مراتب اوضاع بدتر از Deep NN های ساده است!



معایب RNN ساده – محو شدگی گرادیان‌ها

در DNN معمولی مگر مشکل (Vanishing / Exploding) Gradient نیست؟!

در RNNها به مراتب اوضاع بدتر از Deep NN های ساده است!



LONG SHORT-TERM MEMORY

Neural Computation 9(8):1735–1780, 1997

Sepp Hochreiter
Fakultät für Informatik
Technische Universität München
80290 München, Germany
hochreit@informatik.tu-muenchen.de
http://www7.informatik.tu-muenchen.de/~hochreit

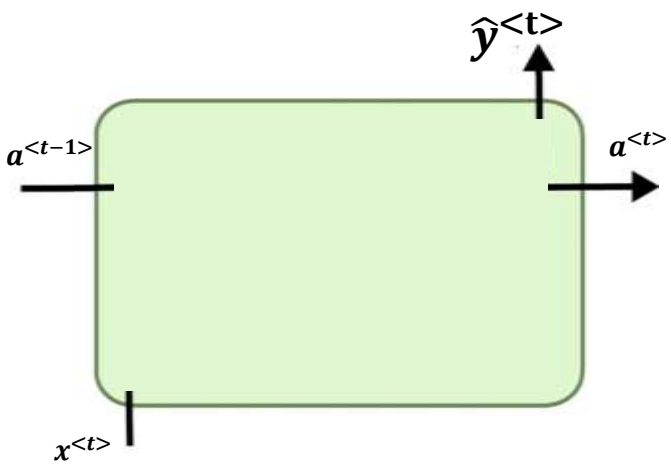
Jürgen Schmidhuber
IDSIA
Corso Elvezia 36
6900 Lugano, Switzerland
juergen@idsia.ch
http://www.idsia.ch/~juergen

Abstract

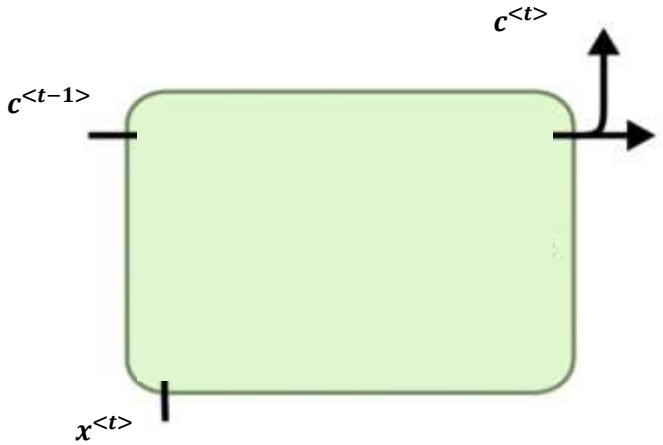
Learning to store information over extended time intervals via recurrent backpropagation takes a very long time, mostly due to insufficient, decaying error back flow. We briefly review Hochreiter's 1991 analysis of this problem, then address it by introducing a novel, efficient, gradient-based method called "Long Short-Term Memory" (LSTM). Truncating the gradient where this does not do harm, LSTM can learn to bridge minimal time lags in excess of 1000 discrete time steps by enforcing *constant* error flow through "constant error carrousels" within special units. Multiplicative gate units learn to open and close access to the constant error flow. LSTM is local in space and time; its computational complexity per time step and weight is $O(1)$. Our experiments with artificial data involve local, distributed, real-valued, and noisy pattern representations. In comparisons with RTRL, BPTT, Recurrent Cascade-Correlation, Elman nets, and Neural Sequence Chunking, LSTM leads to many more successful runs, and learns much faster. LSTM also solves complex, artificial long time lag tasks that have never

انواع واحدها/سلول‌های RNN

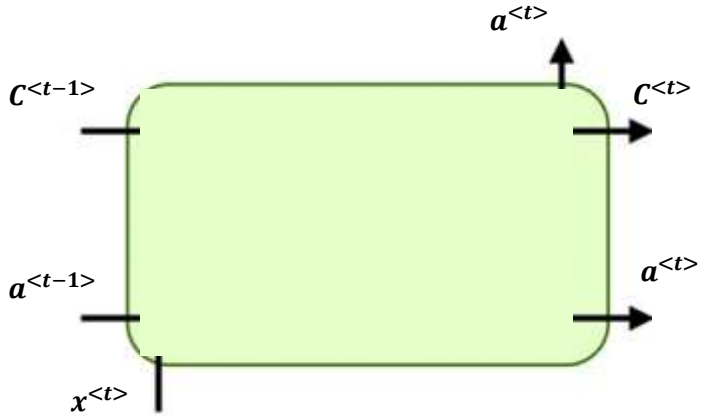
Simple RNN cell



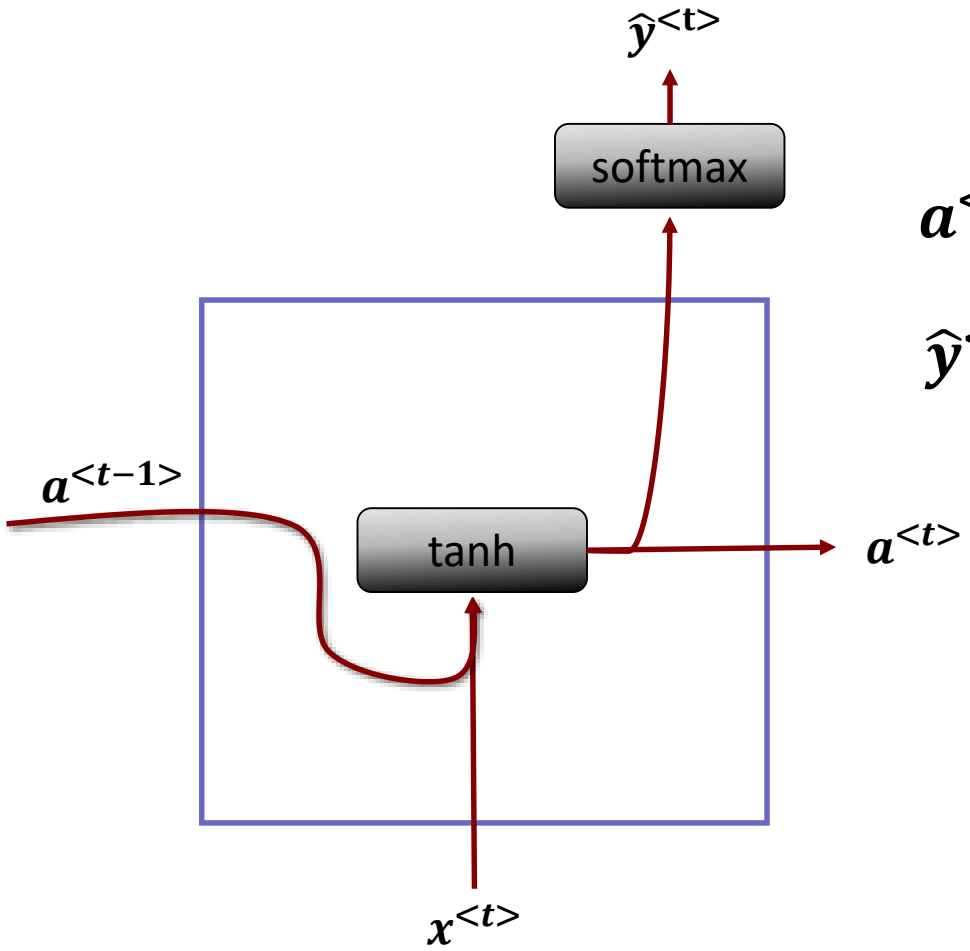
GRU



LSTM



یک واحد RNN ساده



$$a^{<t>} = g_1(w_a\left[a^{<t-1>}, x^{<t>}\right] + b_a)$$

$$\hat{y}^{<t>} = g_2(w_y a^{<t>} + b_y)$$

Gated Recurrent Unit یک واحد GRU (ساده شده)

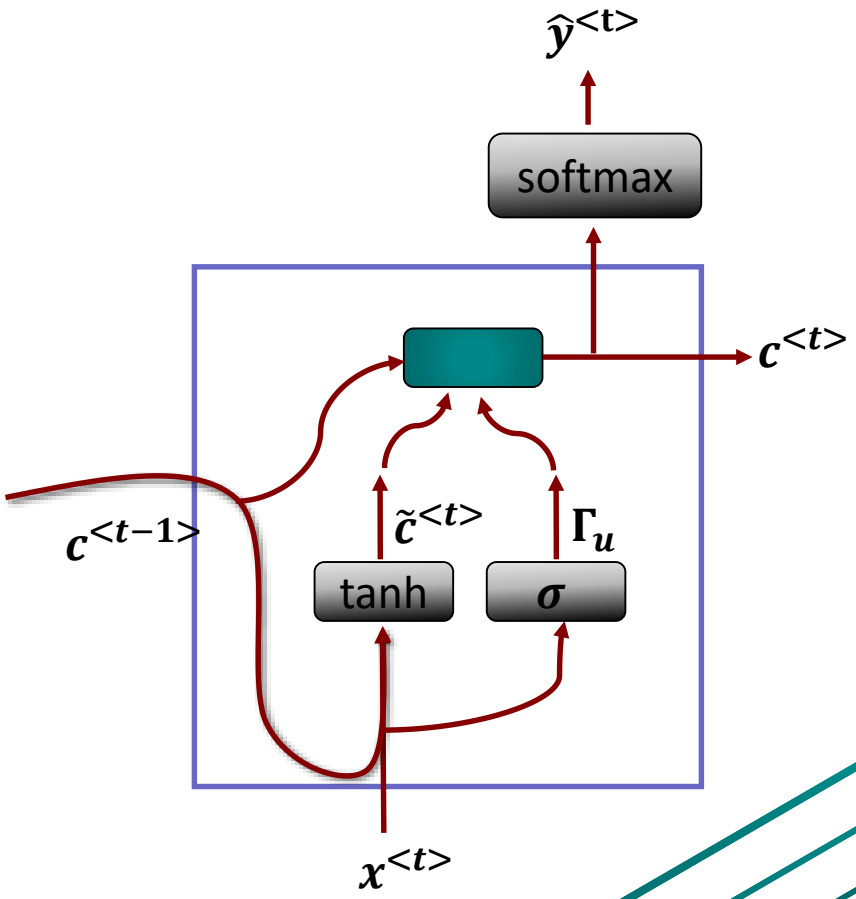
c = memory cell

$$c^{<t>} = a^{<t>}$$

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

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

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



Γ_u =1 Γ_u= 0 Γ_u= 0 Γ_u= 0 ...

The cat, which already ate ..., was full.

$c^{<t>}$ = 1

1

[Cho et al., 2014. On the properties of neural machine translation: Encoder-decoder approaches]
[Chung et al., 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling]

یک واحد GRU (Full GRU)

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

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

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

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

یک واحد LSTM یا Long Short-term Memory

GRU

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

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

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

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

$$a^{<t>} = c^{<t>}$$

LSTM

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

[Hochreiter & Schmidhuber 1997. Long short-term memory]

یک واحد LSTM یا Long Short-term Memory

GRU

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

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

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

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

$$a^{<t>} = c^{<t>}$$

LSTM

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

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

[Hochreiter & Schmidhuber 1997. Long short-term memory]

یک واحد LSTM یا Long Short-term Memory

GRU

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

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

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

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

$$a^{<t>} = c^{<t>}$$

Forget
(فراموشی)

LSTM

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

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

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

[Hochreiter & Schmidhuber 1997. Long short-term memory]

یک واحد LSTM یا Long Short-term Memory

GRU

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

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

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

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

$$a^{<t>} = c^{<t>}$$

LSTM

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

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

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

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

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

[Hochreiter & Schmidhuber 1997. Long short-term memory]

یک واحد LSTM یا Long Short-term Memory

GRU

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

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

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

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

$$a^{<t>} = c^{<t>}$$

LSTM

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

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

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

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

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

$$a^{<t>} = \Gamma_o \circledast c^{<t>}$$

Element-Wise

[Hochreiter & Schmidhuber 1997. Long short-term memory]

LSTM in picture

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

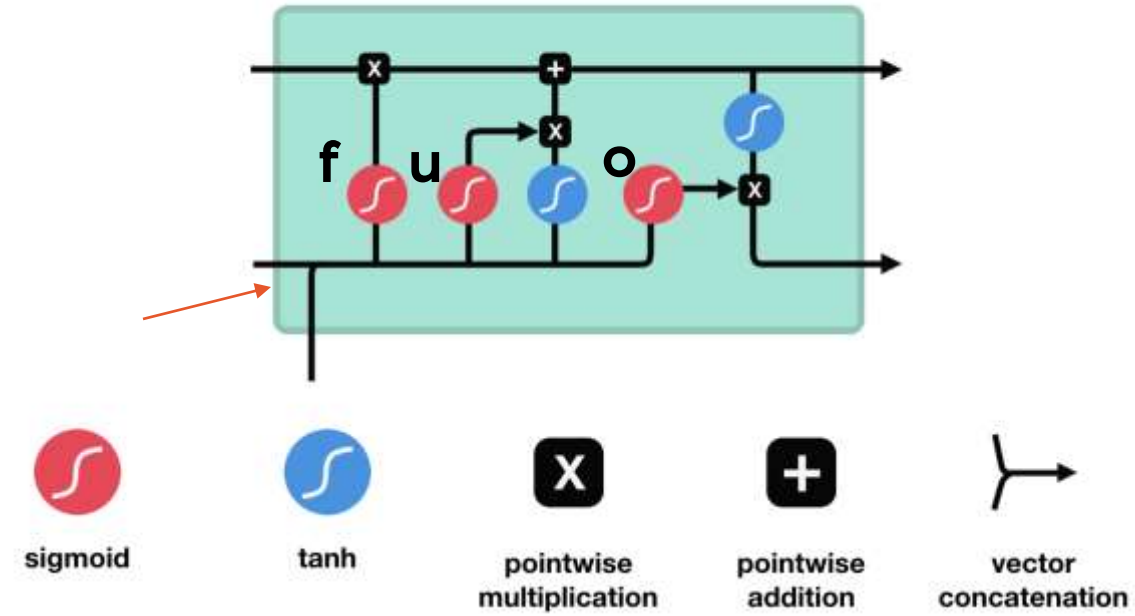
$$\Gamma_u = \sigma(w_u\,[a^{<t-1>}, x^{<t>}] + b_u)$$

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

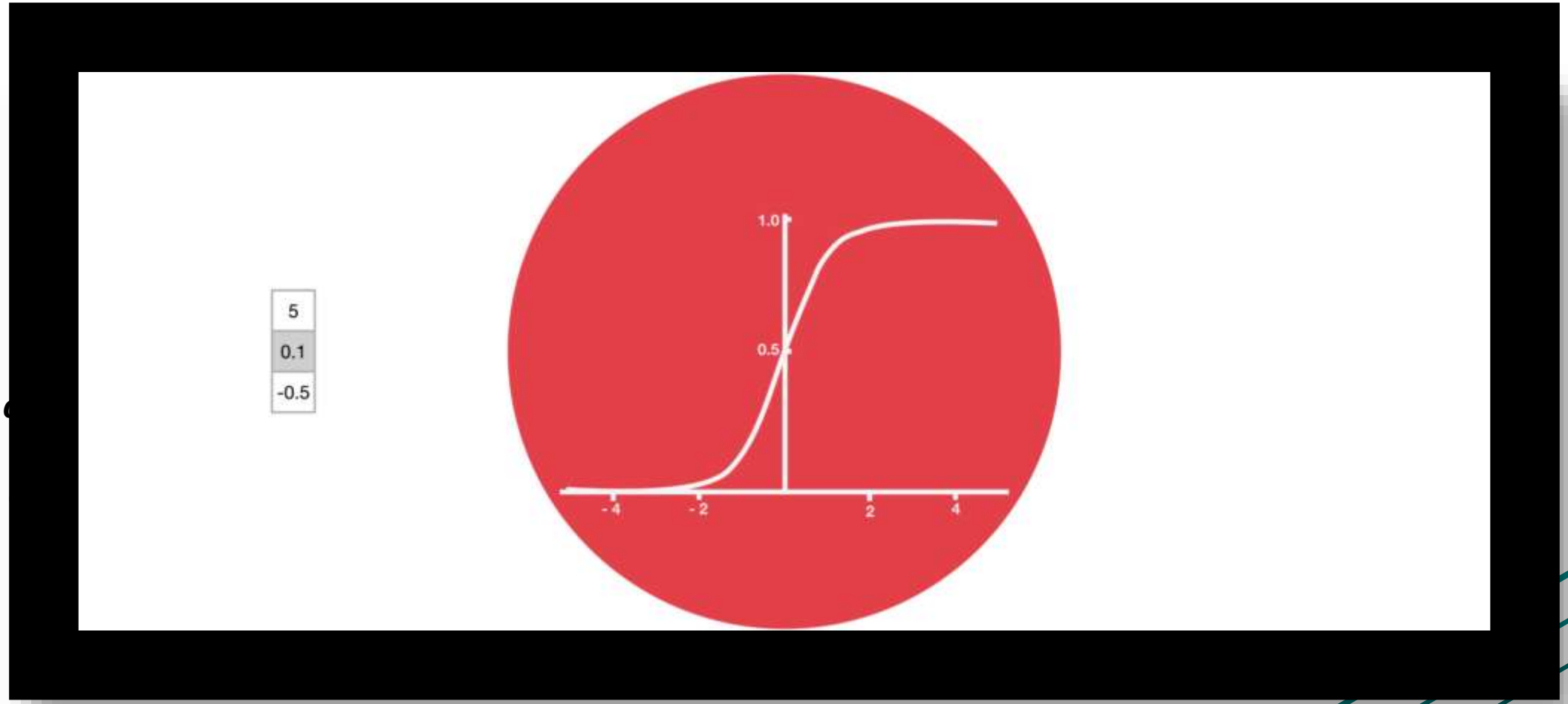
$$\Gamma_o = \sigma(w_o\,[a^{<t-1>}, x^{<t>}] + b_o)$$

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

$$a^{<t>} = \Gamma_o * c^{<t>}$$



LSTM in picture



LSTM in picture

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

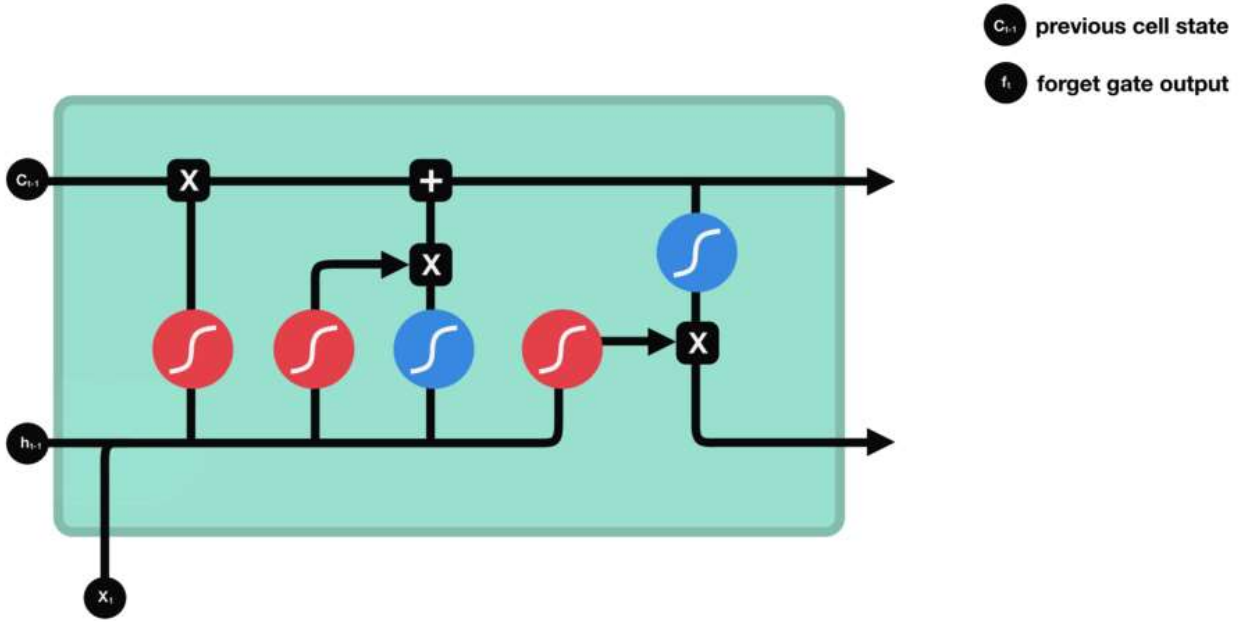
$$\Gamma_u = \sigma(w_u\,[a^{<t-1>}, x^{<t>}] + b_u)$$

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

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

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

$$a^{<t>} = \Gamma_o * c^{<t>}$$



LSTM in picture

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

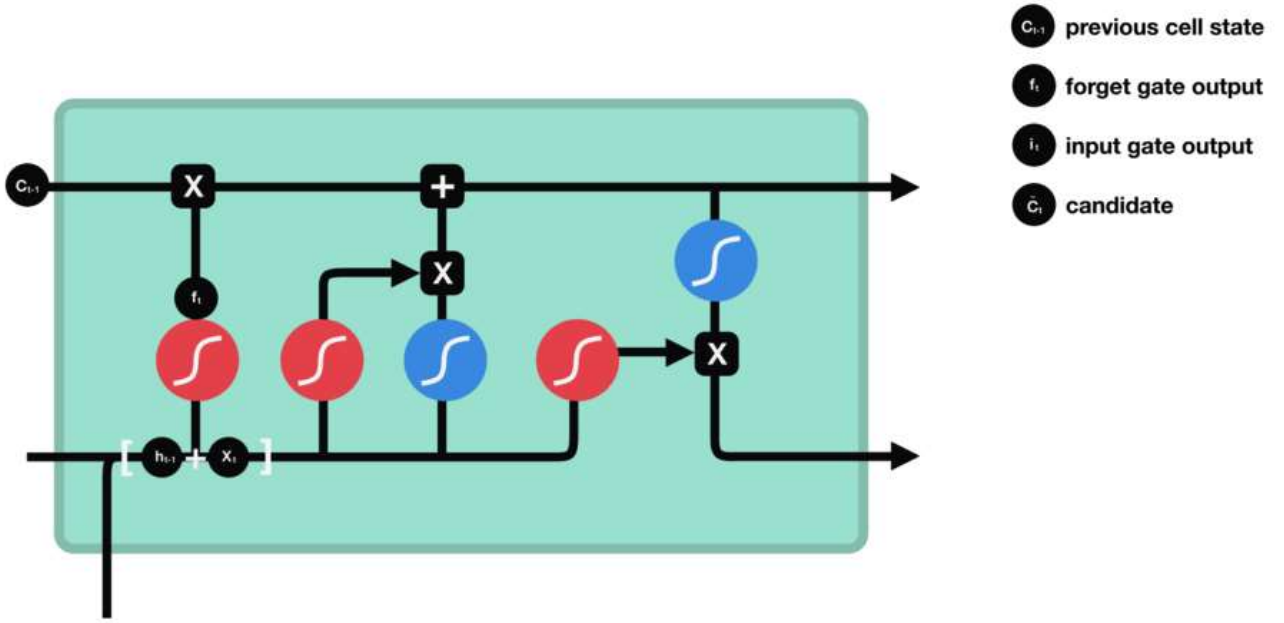
$$\Gamma_u = \sigma(w_u\,[a^{<t-1>}, x^{<t>}] + b_u)$$

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

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

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

$$a^{<t>} = \Gamma_o * c^{<t>}$$



LSTM in picture

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

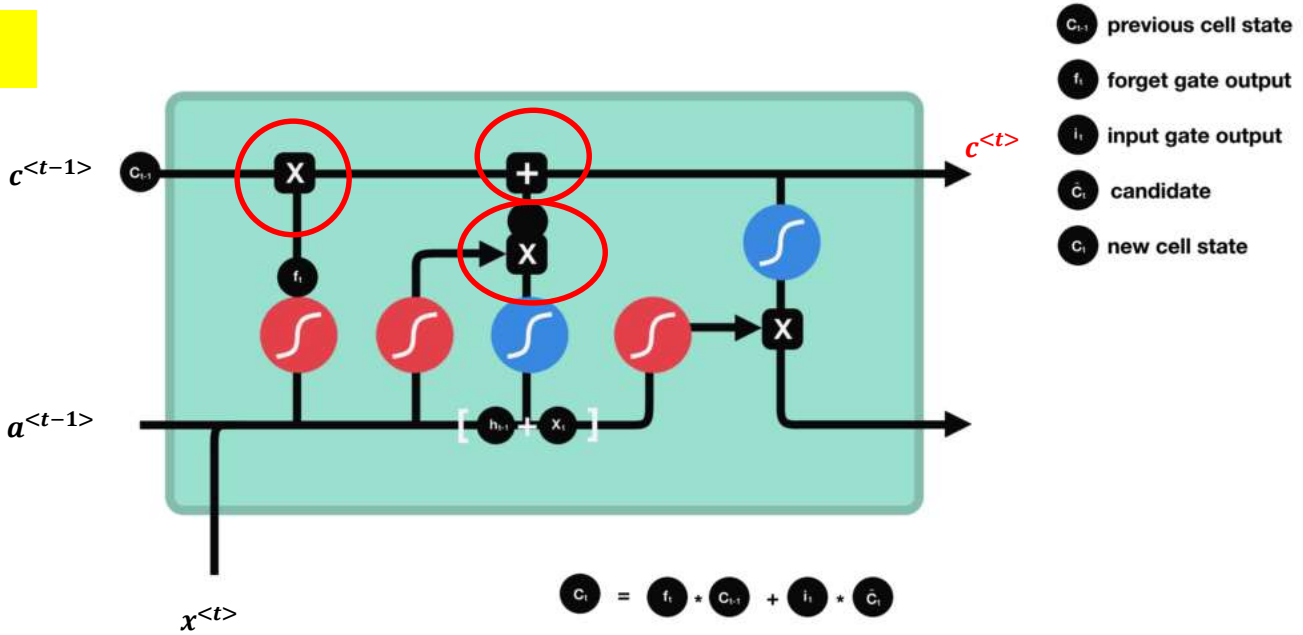
$$\Gamma_u = \sigma(w_u\,[a^{<t-1>}, x^{<t>}] + b_u)$$

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

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

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

$$a^{<t>} = \Gamma_o * c^{<t>}$$



LSTM in picture

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

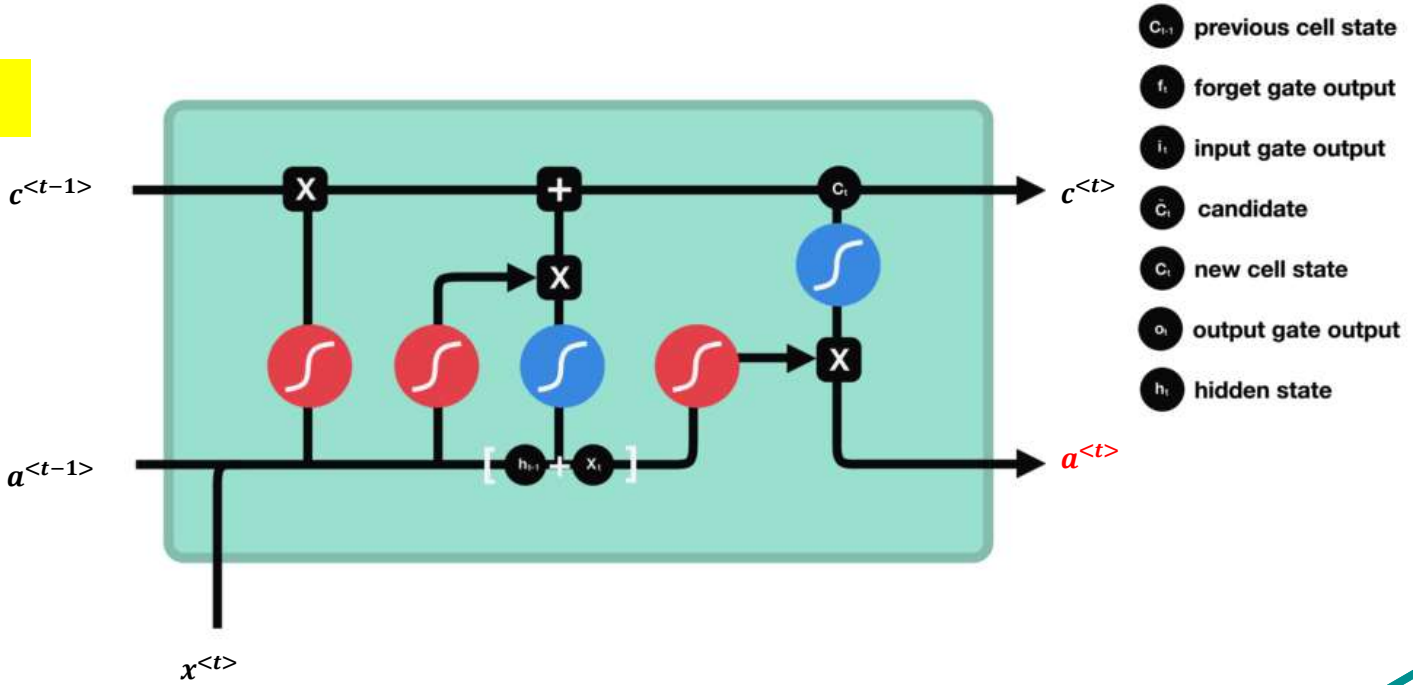
$$\Gamma_u = \sigma(w_u\,[a^{<t-1>}, x^{<t>}] + b_u)$$

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

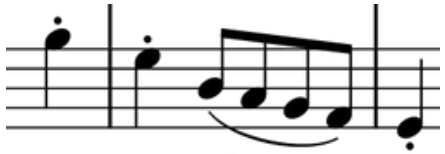
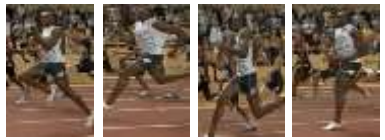
$$\Gamma_o = \sigma(w_o\,[a^{<t-1>}, x^{<t>}] + b_o)$$

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

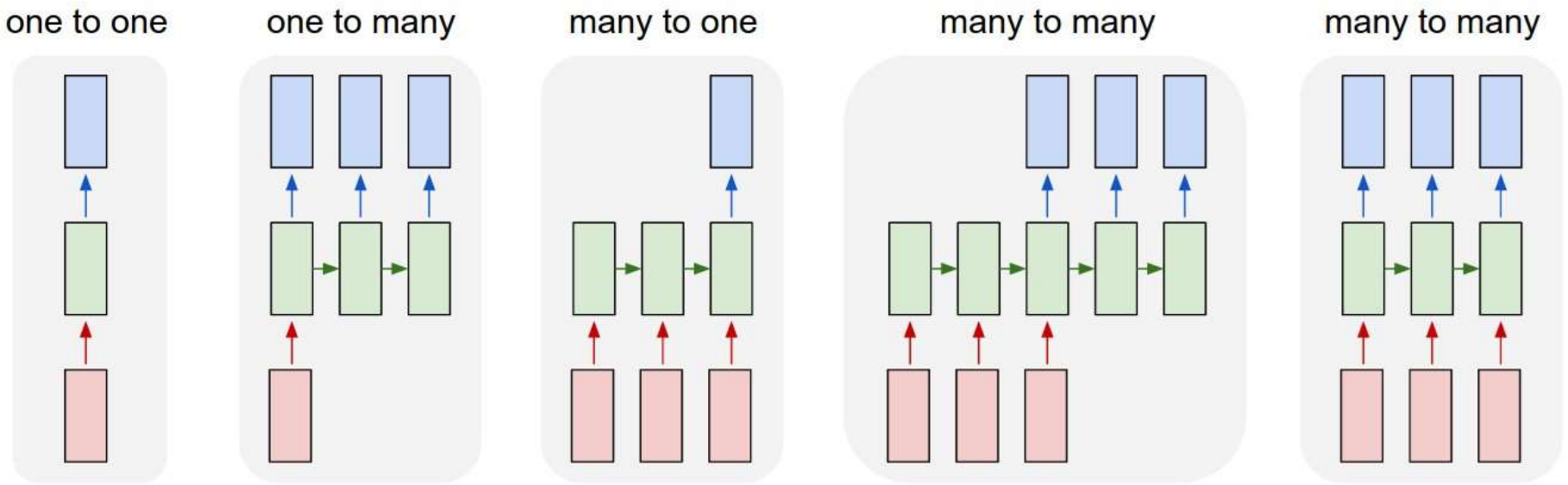
$$a^{<t>} = \Gamma_o * c^{<t>}$$



مثال‌هایی از داده های توالی و ترتیبی (sequence data)

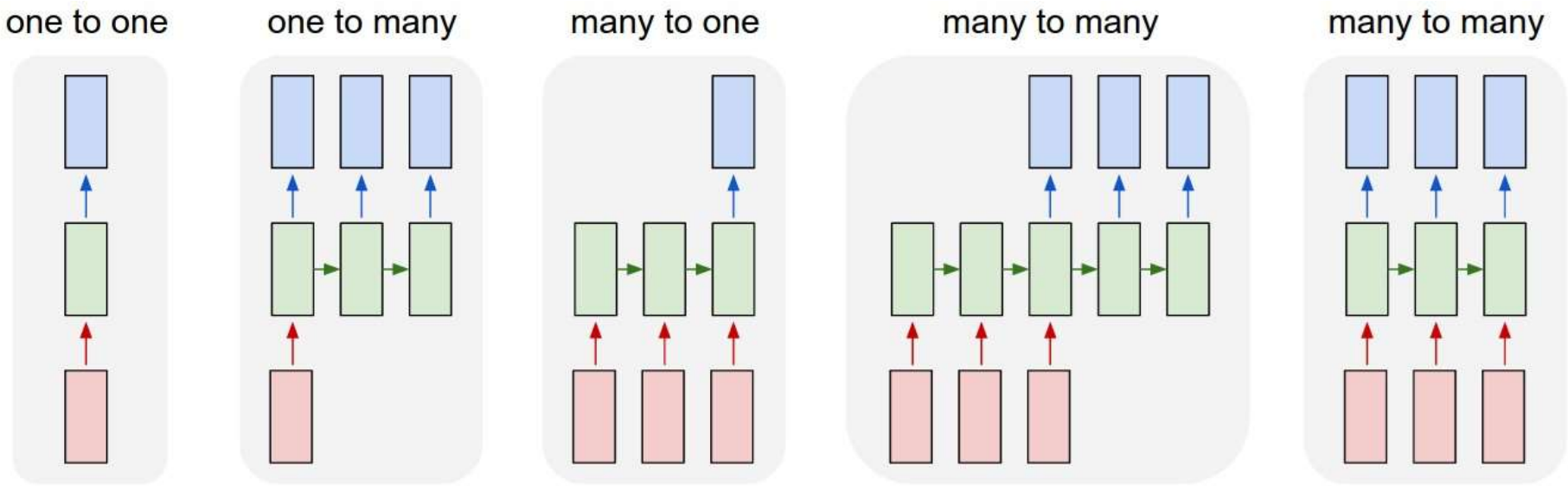
Music generation	∅	
Sentiment classification	“There is nothing to like in this movie.”	★
DNA sequence analysis	AGCCCCTGTGAGGAACTAG	AGCCCCTGTGAGGAACTAG
Machine translation	حالت چه طوره؟	How are you?
Video activity recognition		Running
Name entity recognition	Yesterday, Harry Potter met Hermione Granger.	Yesterday, Harry Potter met Hermione Granger.
Speech recognition		“The quick brown fox jumped over the lazy dog.”

انواع شبکه‌های ترتیبی



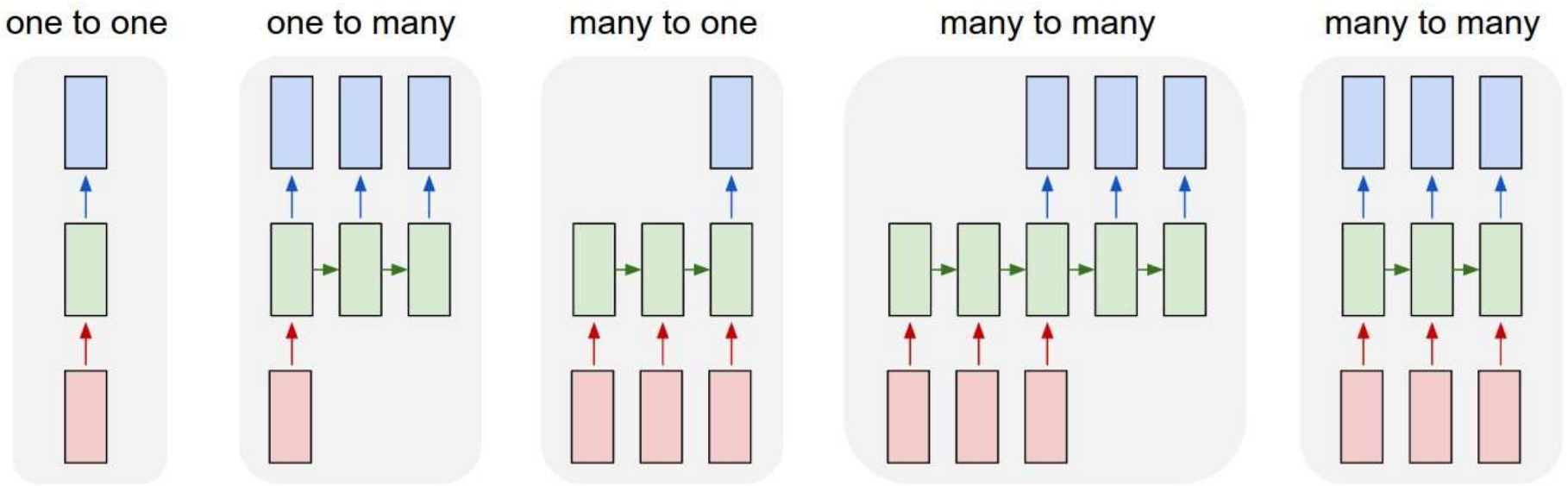
Vanilla Neural Networks

انواع شبکه‌های ترتیبی



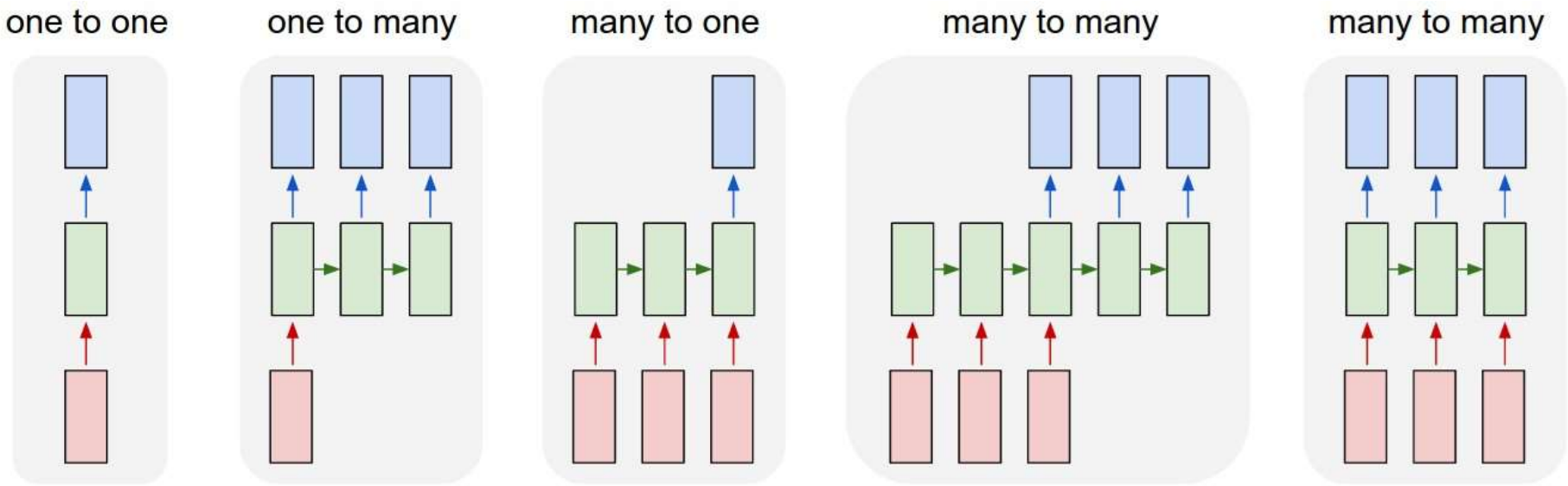
e.g. **Image Captioning**
image -> sequence of words

انواع شبکه‌های ترتیبی



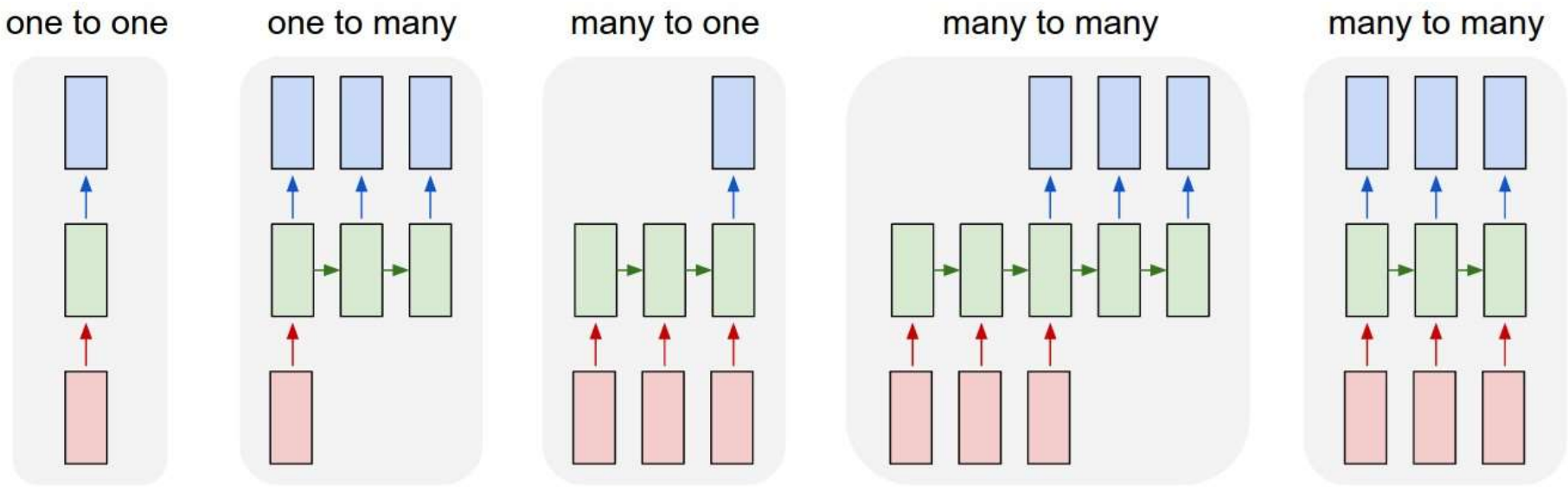
e.g. **Sentiment Classification**
sequence of words -> sentiment

انواع شبکه‌های ترتیبی



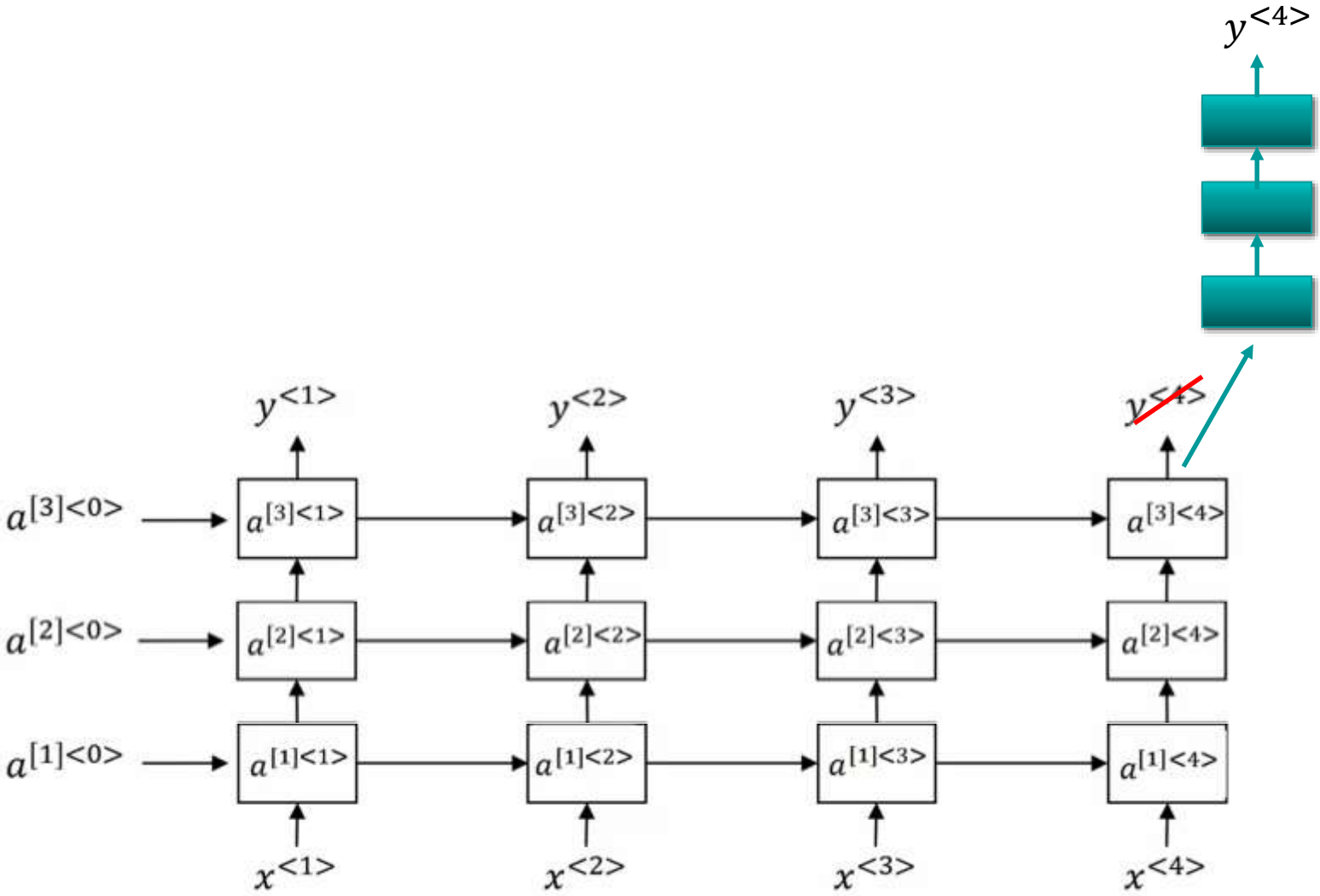
e.g. **Machine Translation**
seq of words -> seq of words

انواع شبکه‌های ترتیبی



e.g. **Video classification on frame level**

شبکه‌های بازگشتی عمیق



تخمین قیمت ارزهای دیجیتال

03_1_Cryptocurrency-predicting.ipynb

مسیر توپ!

04_simple-CNN-LSTM.ipynb

طبقه بندی ویدیو

05_1_video_action_recognition

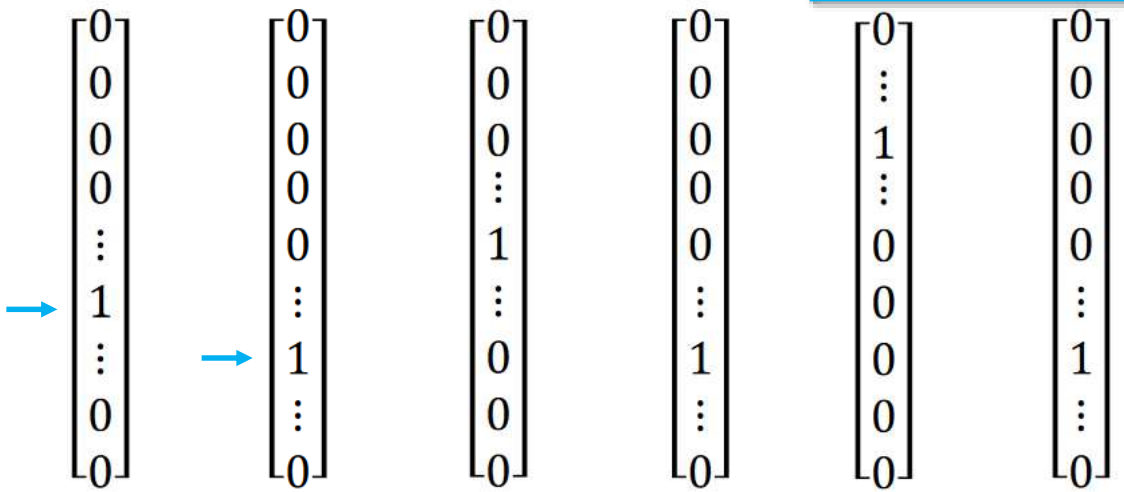
بردار کلمات و بازنمایی کلمات — Word Embeddings

V = [a, aaron, ..., zulu, <UNK>]

$|V| = 10,000$

1-hot representation

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



O_{5391} O_{9853}

I want a glass of orange _____ .

juice

I want a glass of apple _____ .

?

مشکل؟

فاصله اقلیدسی تمام بردارها یکسان است.

از روی کلماتی که در آموزش دیده است نمیتواند generalize کند.

بردار کلمات و بازنمایی کلمات — Word Embeddings

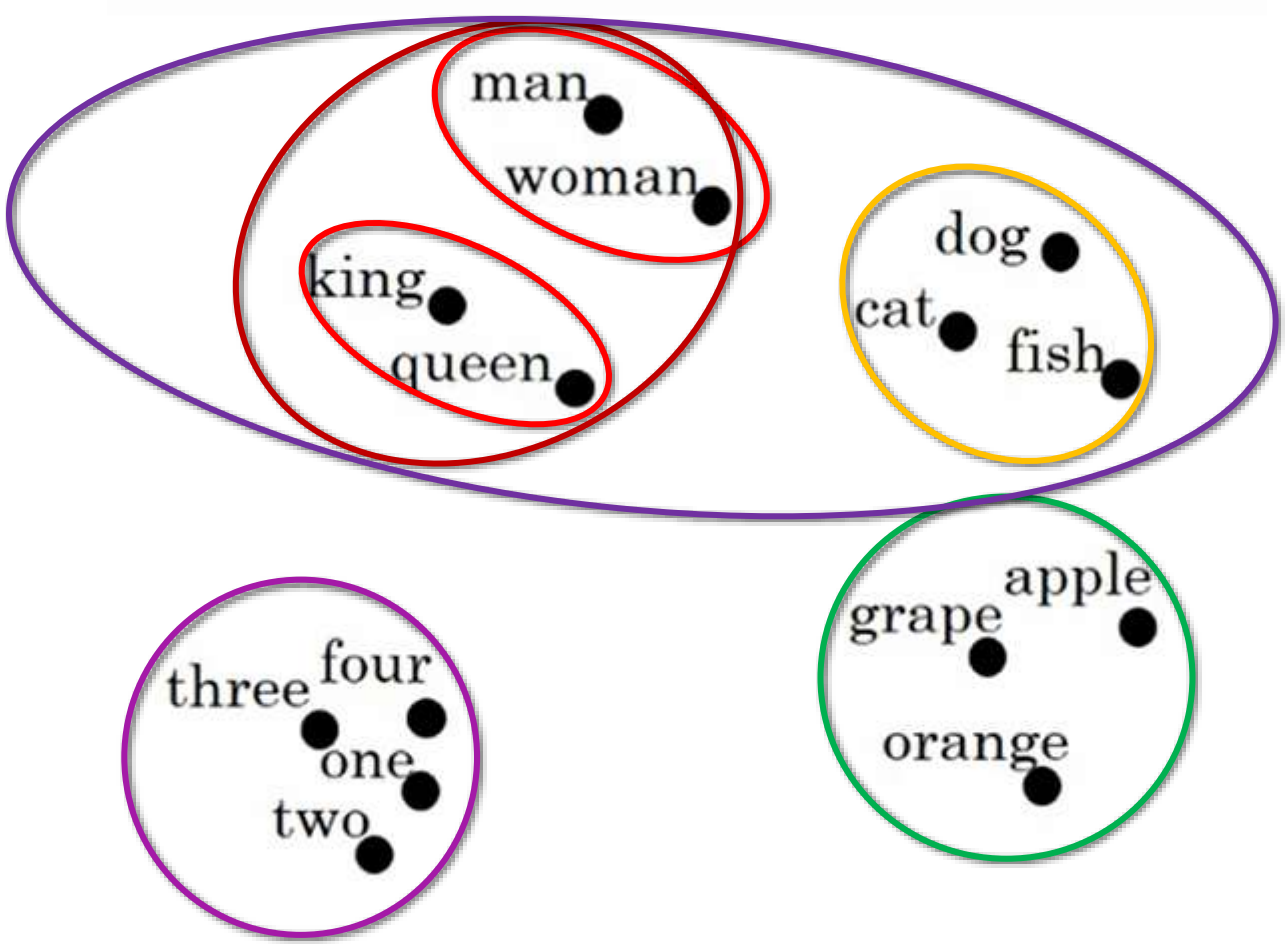
Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)	
-1	1	-0.95	0.97	0.00	0.01	جنسیت
0.01	0.02	0.94	0.93	-0.01	0.00	سلطنتی
0.03	0.02	0.71	0.69	0.03	-0.02	سن
0.01	-0.01	0.02	0.00	0.96	-0.97	خوراکی
				⋮	⋮	سایز زنده قیمت فعل ....
				e_{456}	e_{6257}	

300

I want a glass of orange _____ *juice* .

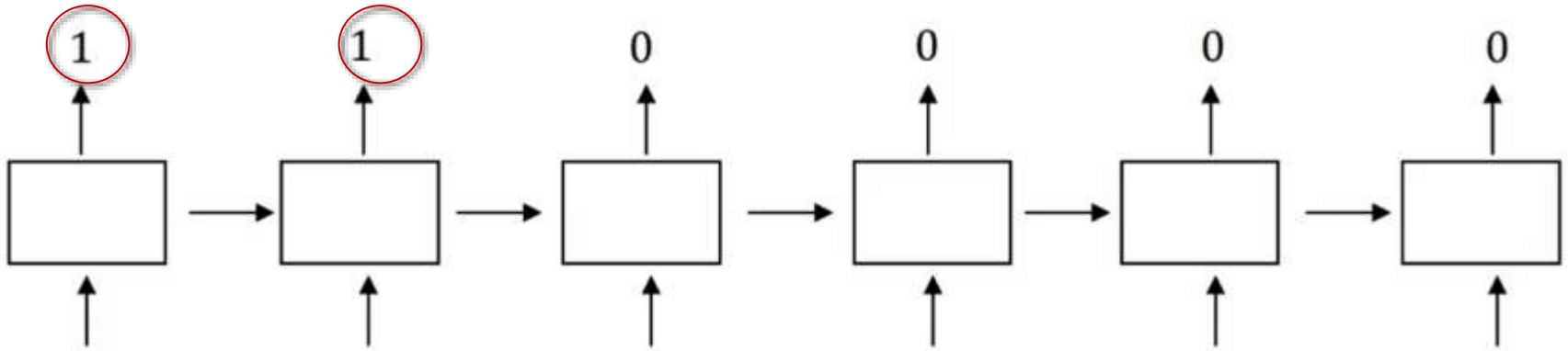
I want a glass of apple _____ *juice* .

Visualizing word embeddings – بصری‌سازی بردار



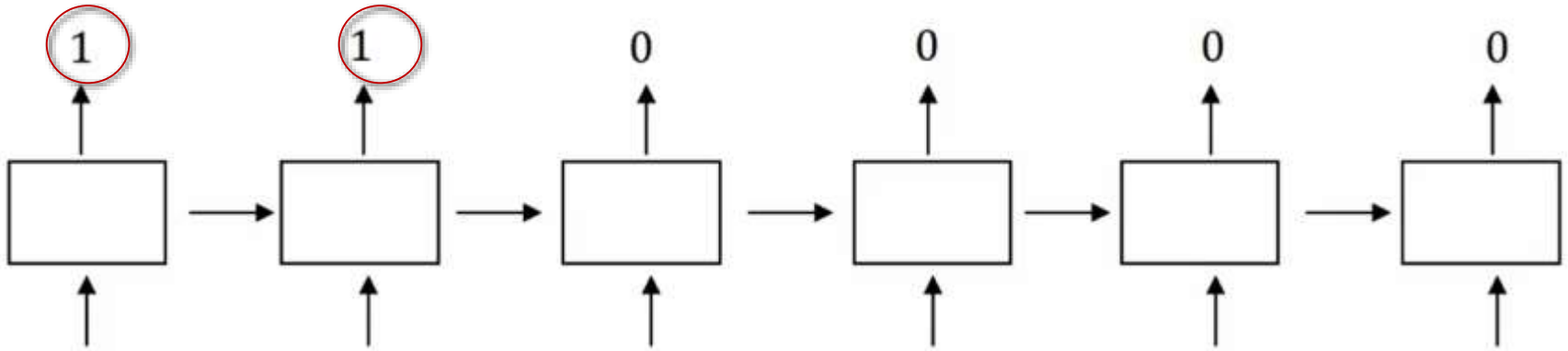
[van der Maaten and Hinton., 2008.Visualizing data using t-SNE]

استفاده از بردار کلمات برای Named entity recognition



Sally	Johnson	is	an	orange	farmer	Train
Robert	**Lin**	is	an	apple	farmer	Test
Robert	**Lin**	is	a	durian	cultivator	Test

استفاده از بردار کلمات برای Named entity recognition



Sally	Johnson	is	an	orange	farmer	Train
Robert	**Lin**	is	an	apple	farmer	Test
Robert	**Lin**	is	a	durian	cultivator	Test

یادگیری انتقالی و word embedding

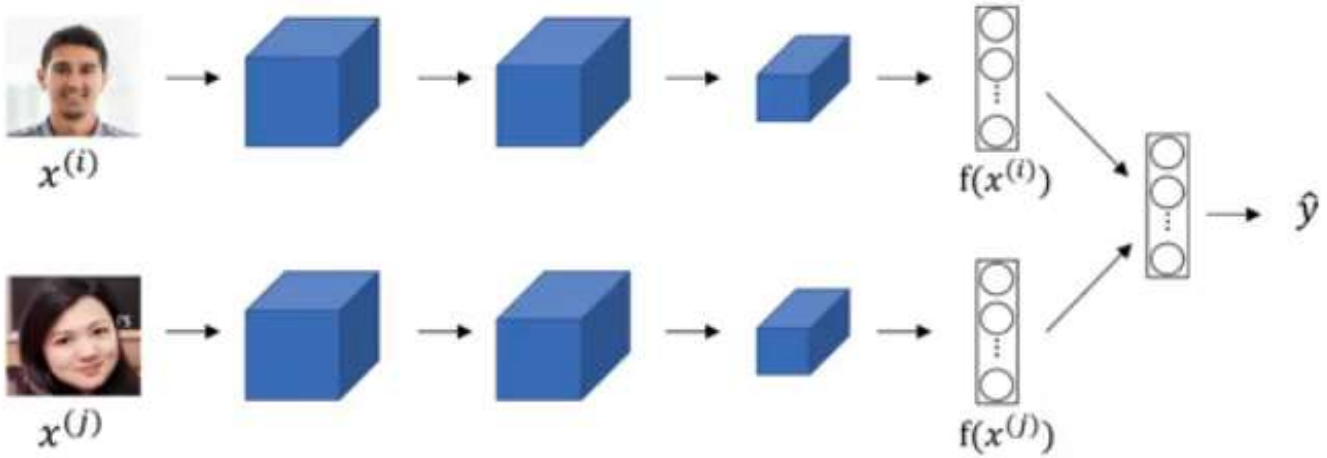
I. Learn word embeddings from large text corpus (1-100 billion of words).
   - Or download pre-trained embedding online.

II. Transfer embedding to new task with the smaller training set (say, 100k words).

III. Optional: continue to finetune the word embeddings with new data.
   - You bother doing this if your smaller training set (from step 2) is big enough.

رابطه با encoding یا Embeddingهای چهره

- **Word embeddings have an interesting relationship to the face recognition task:**
  - In this problem, we encode each face into a vector and then check how similar are these vectors.
  - Words encoding and embeddings have a similar meaning here.
- In the word embeddings task, we are learning a representation for each word in our vocabulary (unlike in image encoding where we have to map each new image to some n-dimensional vector).



[Taigman et. al., 2014. DeepFace: Closing the gap to human level performance]

ویژگی‌های word embedding

- Analogies

	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
	e_{Man}	e_{Woman}	e_{King}	e_{Queen}		

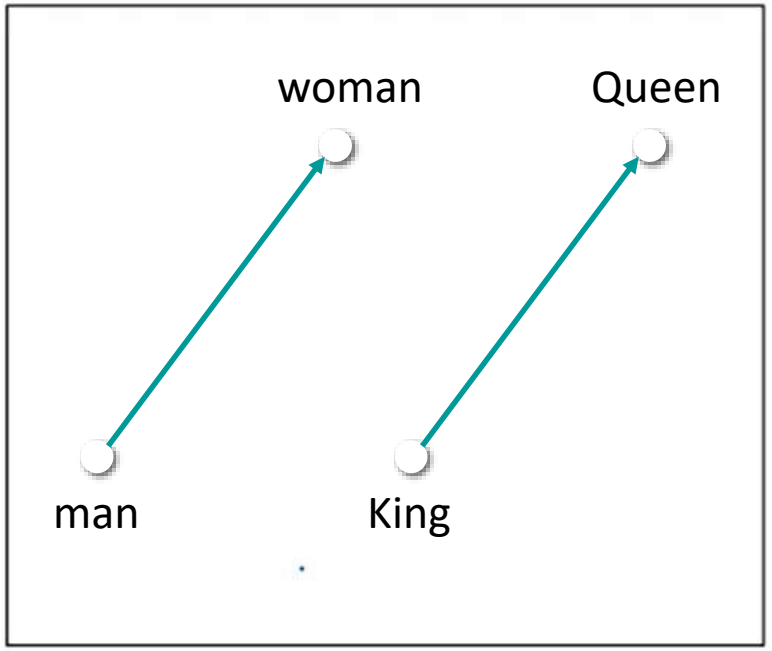
❑ **Can we conclude this relation:**

- Man ==> Woman
- King ==> ??

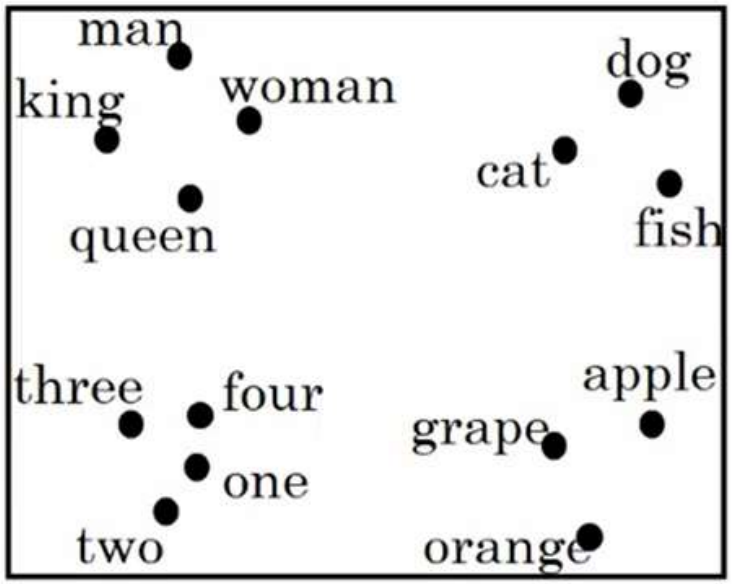
$$e_{Man} - e_{Woman} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad e_{King} - e_{Queen} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

[Mikolov et. al., 2013, Linguistic regularities in continuous space word representations]

Analogies using word vectors



300 D

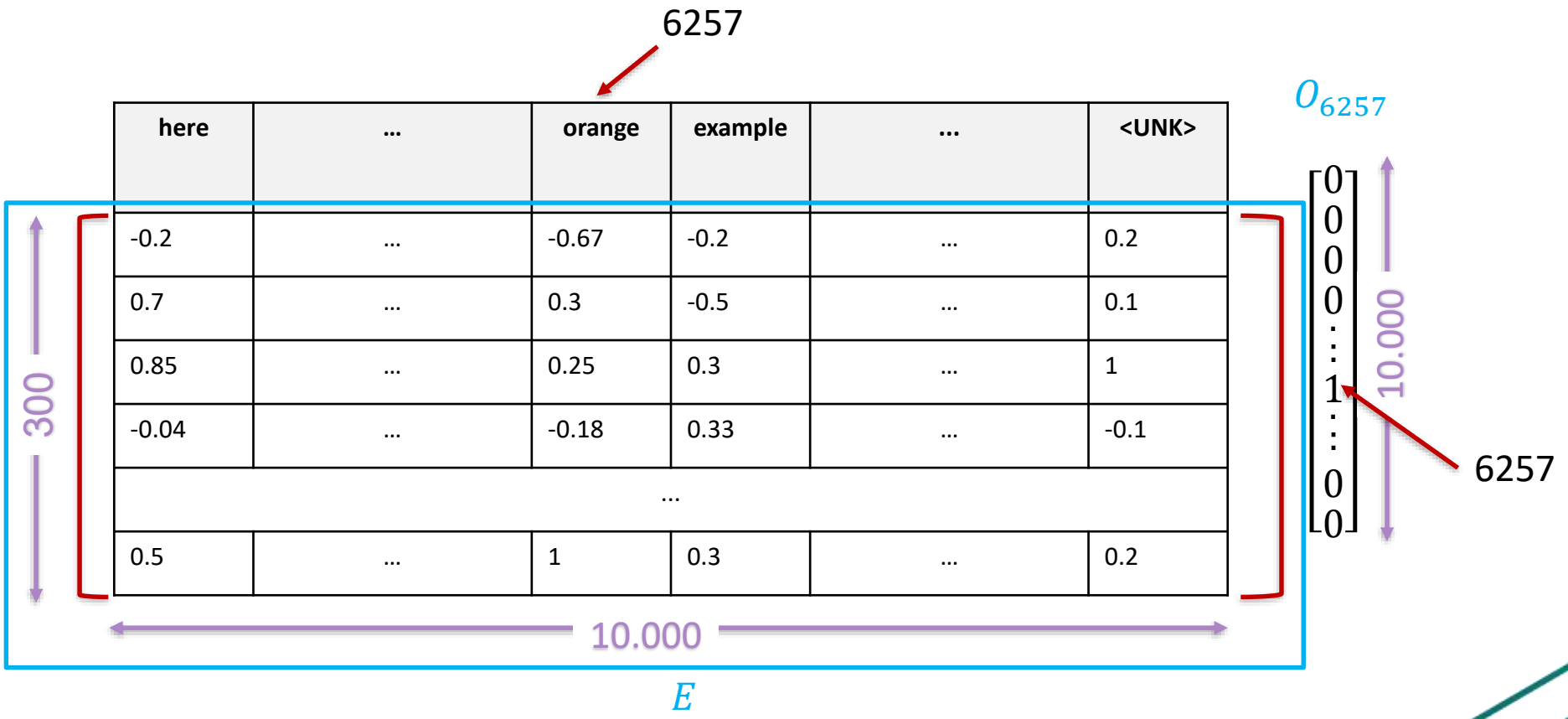


t-SNE

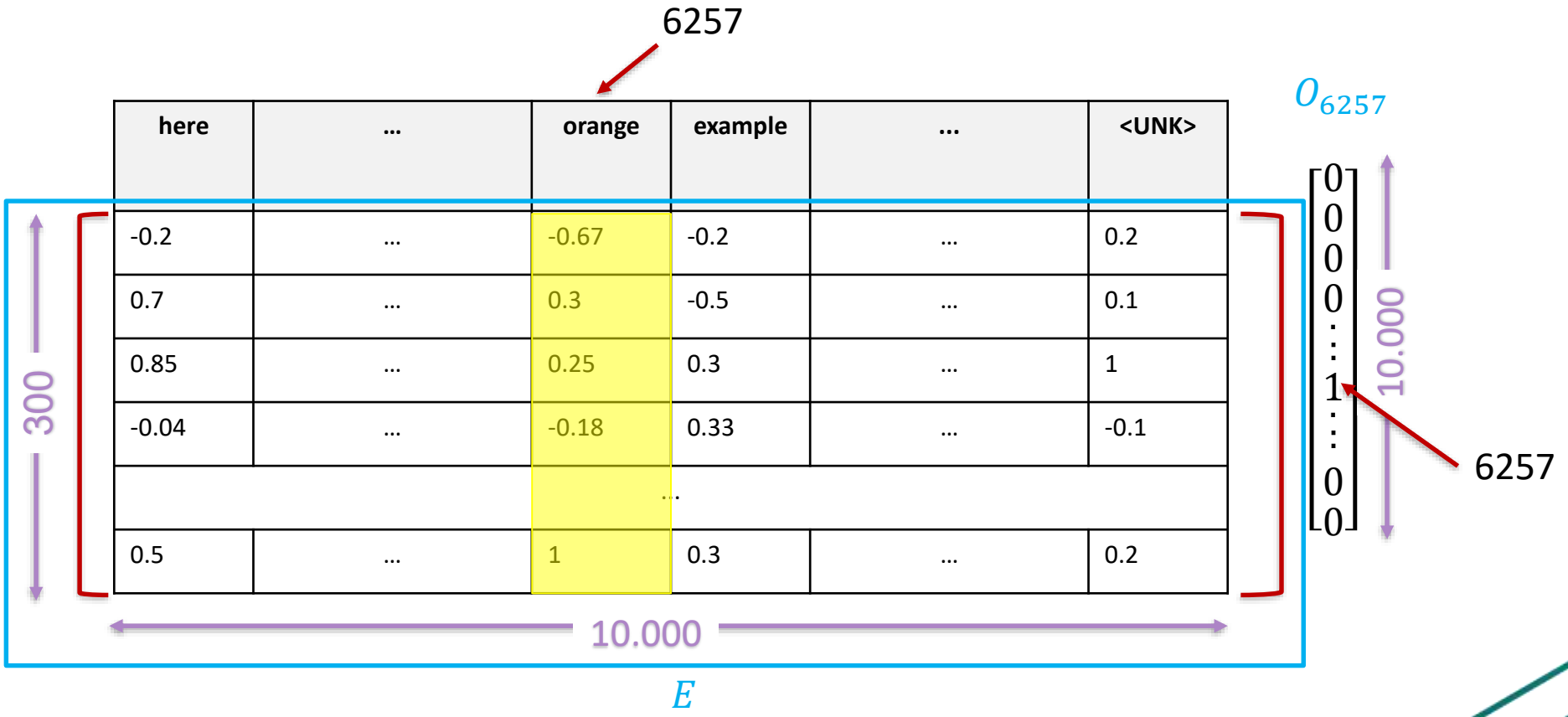
$$e_{man} - e_{woman} \approx e_{king} - \cancel{e_?}\ e_w$$

$$\textbf{Find word } w\text{: } arg\max_w sim(e_w, e_{king} - e_{man} + e_{woman})$$

ماتریس Embedding



ماتریس Embedding



$$E \cdot O_{6257} = e_{6257}$$

ماتریس Embedding

here	...	orange	example	...	<UNK>
-0.2	...	-0.67	-0.2	...	0.2
0.7	...	0.3	-0.5	...	0.1
0.85	...	0.25	0.3	...	1
-0.04	...	-0.18	0.33	...	-0.1
		...			
0.5	...	1	0.3	...	0.2

$$\begin{bmatrix}0\\0\\0\\0\\\vdots\\1\\\vdots\\0\\0\end{bmatrix}$$

$$E \cdot O_{6257} = e_{6257}$$

300x10k

10k x 1

300 x 1

- If O_{6257} is the one hot encoding of the word **orange** of shape (10000, 1), then $np.dot(E,O_{6257}) = e_{6257}$ which shape is (300, 1).
- Generally $np.dot(E, O_j) = e_j$ ***(embedding for word j)***

ماتریس Embedding

https://keras.io/layers/embeddings/

The **Embedding layer** is best understood **as a dictionary mapping integer indices** (which stand for specific words) to dense vectors.
It takes as input integers, it looks up these integers into an internal dictionary, and it returns the associated vectors. **It's effectively a dictionary lookup.**

```
from keras.layers import Embedding

# The Embedding layer takes at least two arguments:
# the number of possible tokens, here 1000 (1 + maximum word index),
# and the dimensionality of the embeddings, here 64.
embedding_layer = Embedding(1000, 64)
```

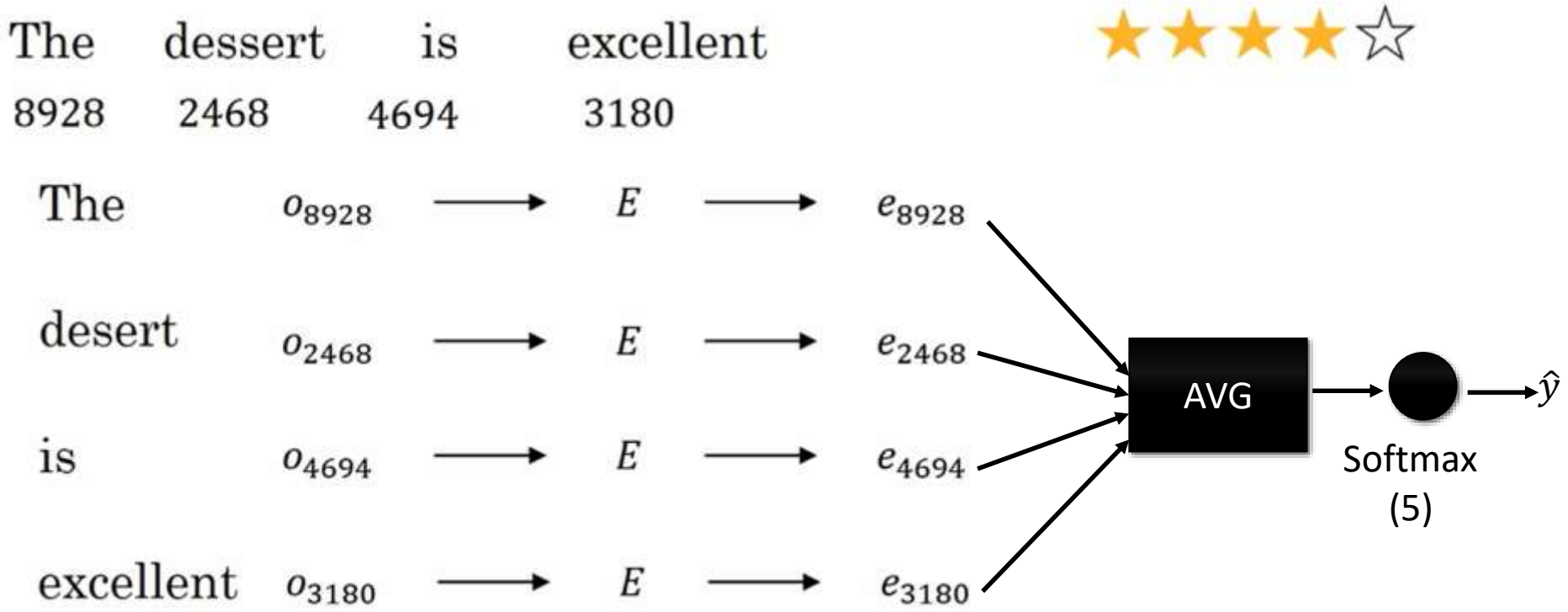
آنالوژی!

06_analogy-using-embeddings.ipynb

Sentiment classification problem

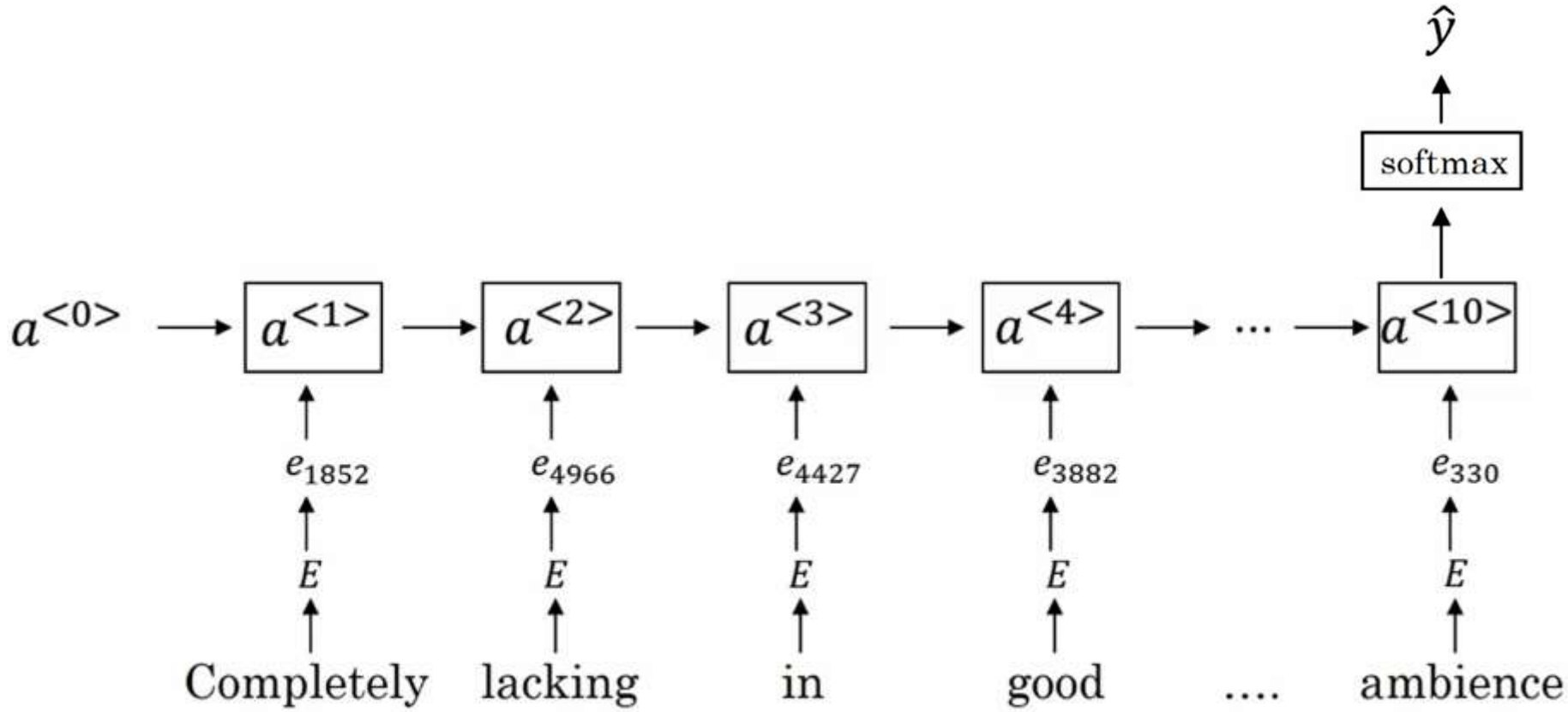
x	y
The dessert is excellent.	★★★★☆
Service was quite slow.	★★☆☆☆
Good for a quick meal, but nothing special.	★★★☆☆
Completely lacking in good taste, good service, and good ambience.	★☆☆☆☆

Simple sentiment classification model



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

RNN for sentiment classification

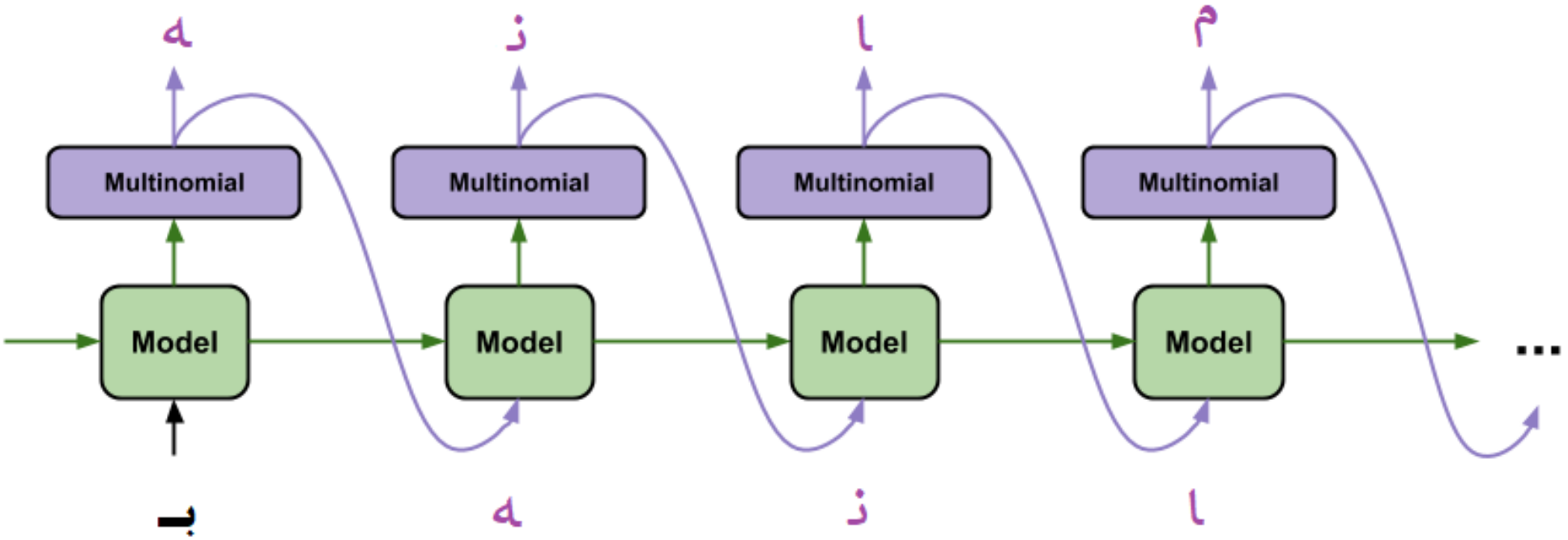


Many-to-one

طبقه بندی متن

07-text-classification-Emojify.ipynb

مدل زبانی – در سطح کاراکتر

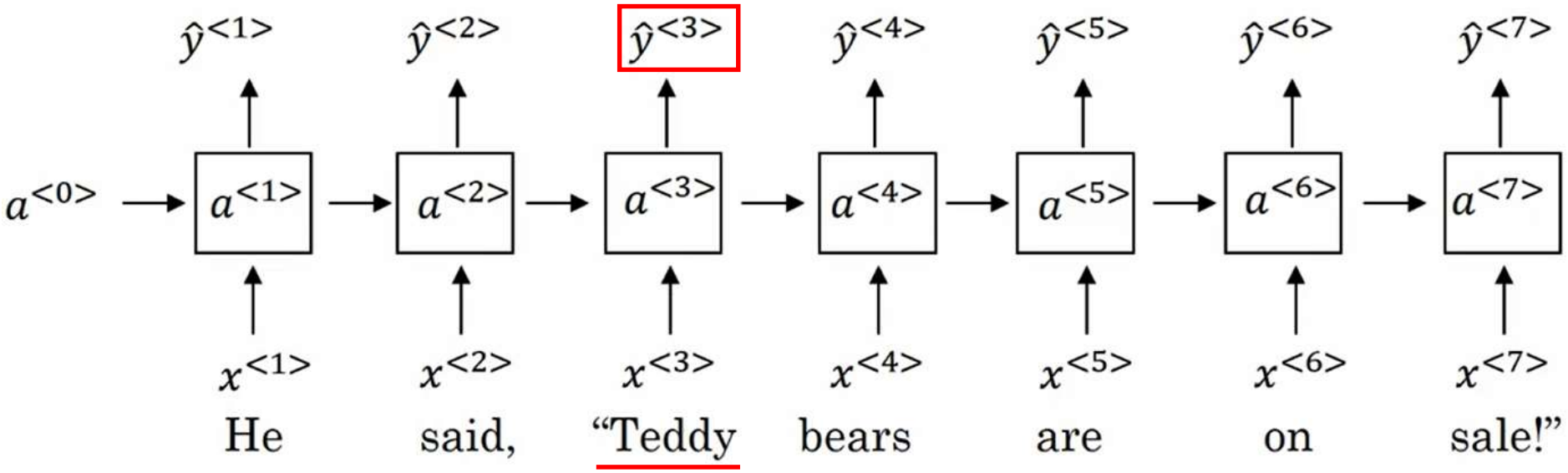


مدل زبانی

08_text_generation.ipynb

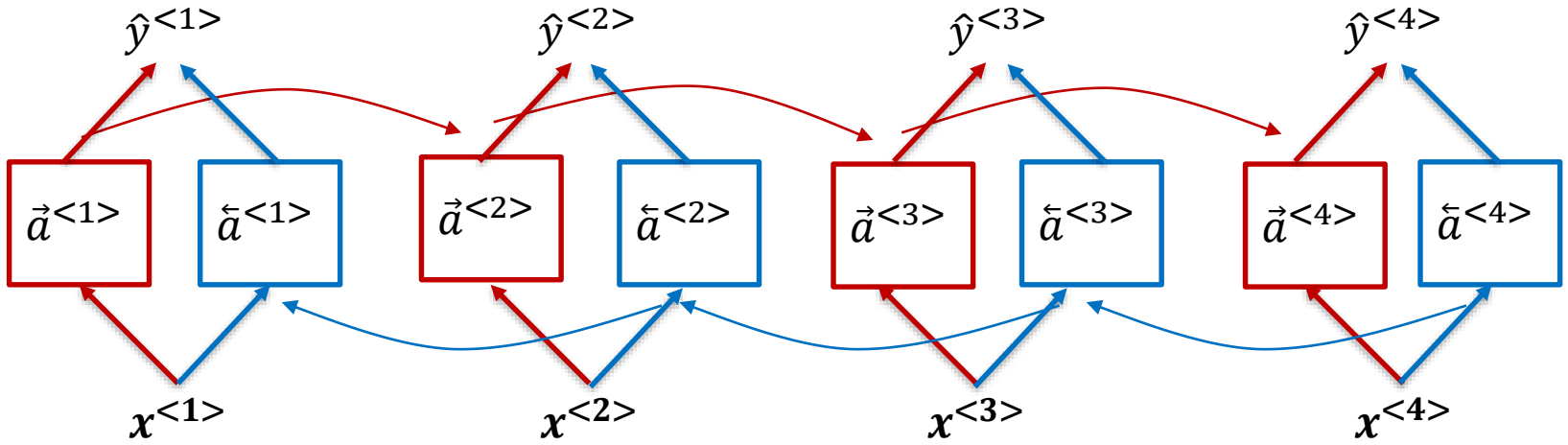
استفاده از اطلاعات آینده با Bidirectional RNN !

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



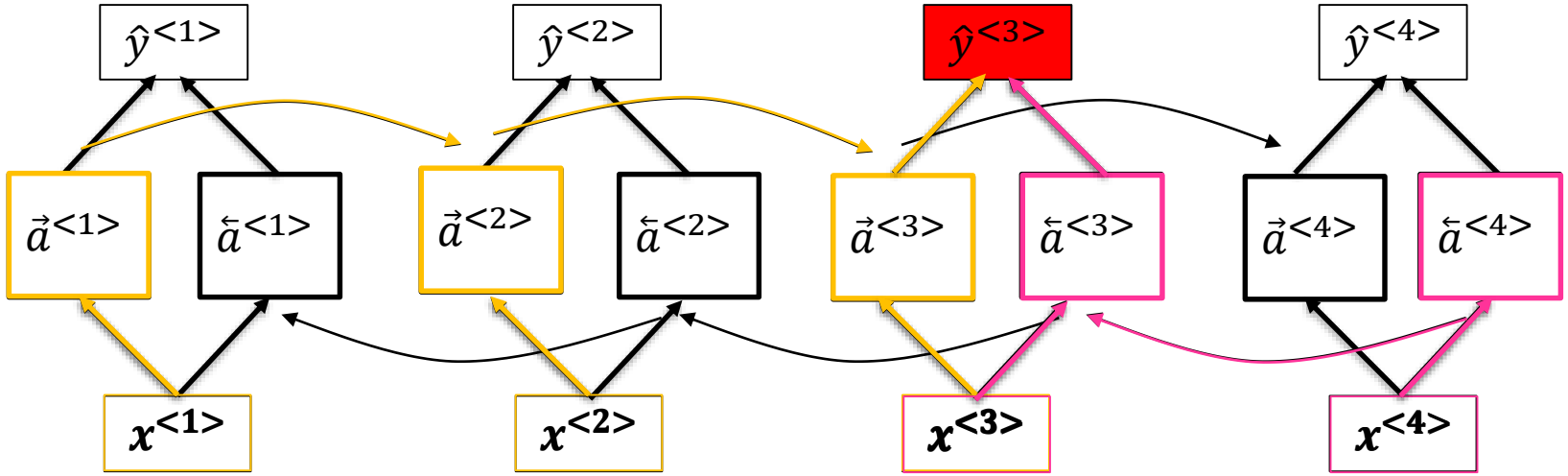
استفاده از اطلاعات آینده با Bidirectional RNN !

$$\hat{y}^{<t>} = g(w_y[\vec{a}^{<1>}, \overleftarrow{a}^{<1>}] + b_y)$$



استفاده از اطلاعات آینده با Bidirectional RNN !

$$\hat{y}^{<t>} = g(w_y[\vec{a}^{<1>}, \overleftarrow{a}^{<1>}] + b_y)$$



برای مثال اگر بخواهیم $\hat{y}^{<3>}$ را حساب کنیم

استفاده از اطلاعات آینده با Bidirectional RNN !

from keras.layers import Bidirectional

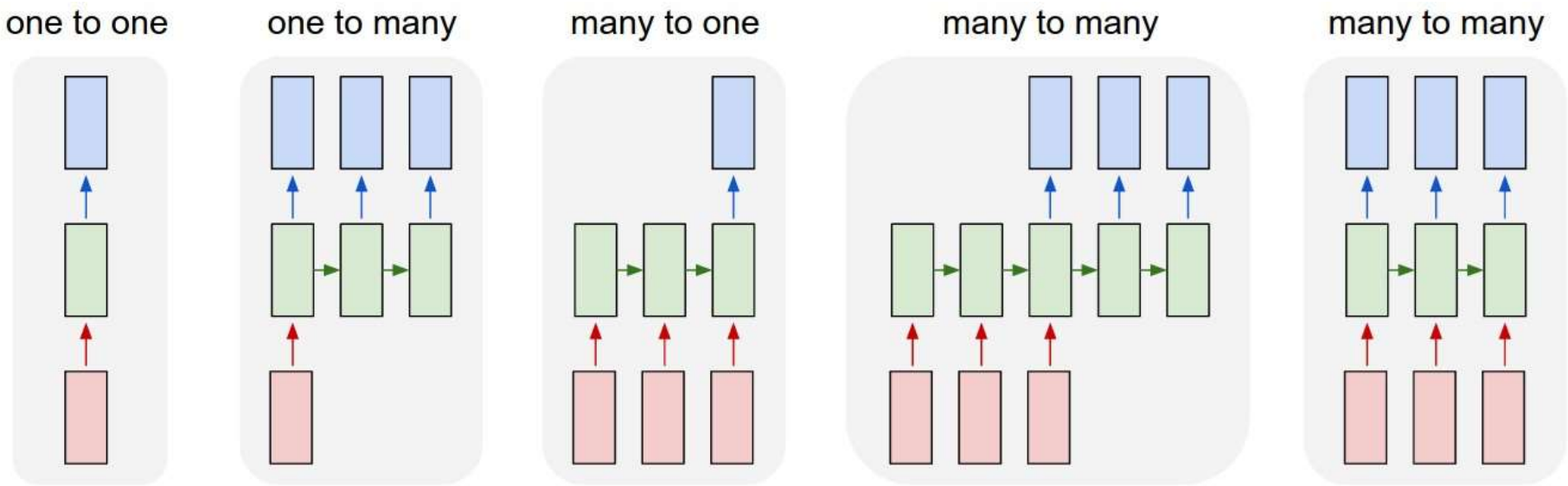
Examples

```
model = Sequential()
model.add(Bidirectional(LSTM(10, return_sequences=True),
                        input_shape=(5, 10)))
model.add(Bidirectional(LSTM(10)))
model.add(Dense(5))
model.add(Activation('softmax'))
model.compile(loss='categorical_crossentropy', optimizer='rmsprop'
```

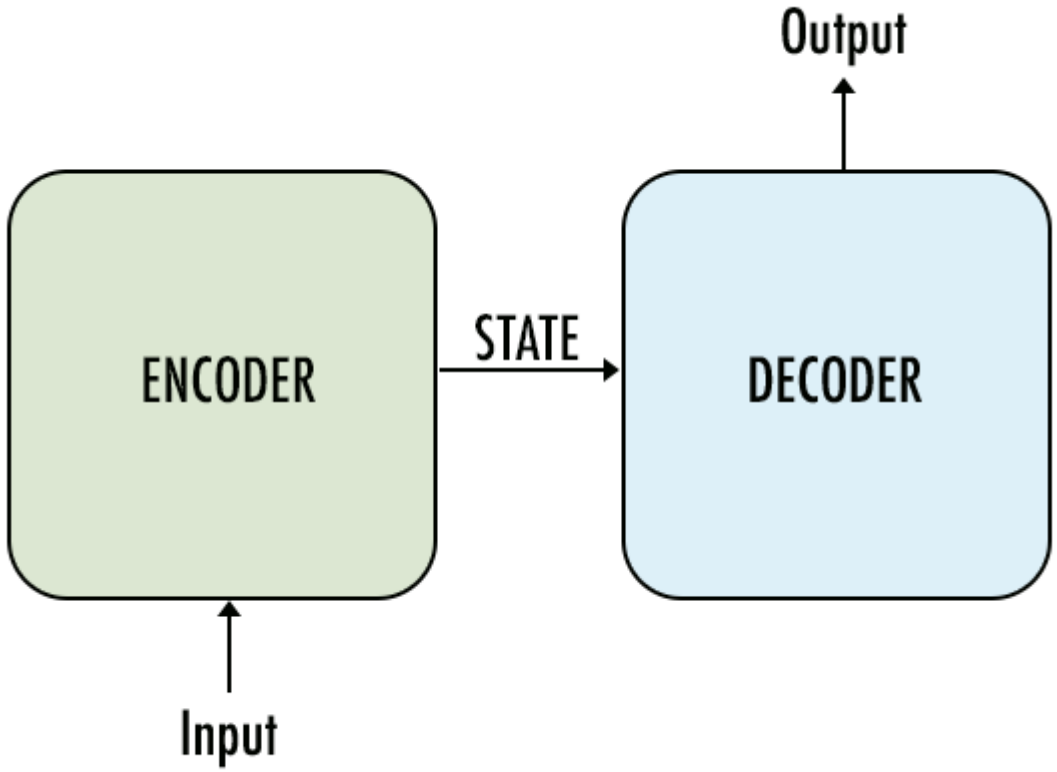
https://keras.io/layers/wrappers/#bidirectional

Seq2Seq
(Encoder-Decoder)

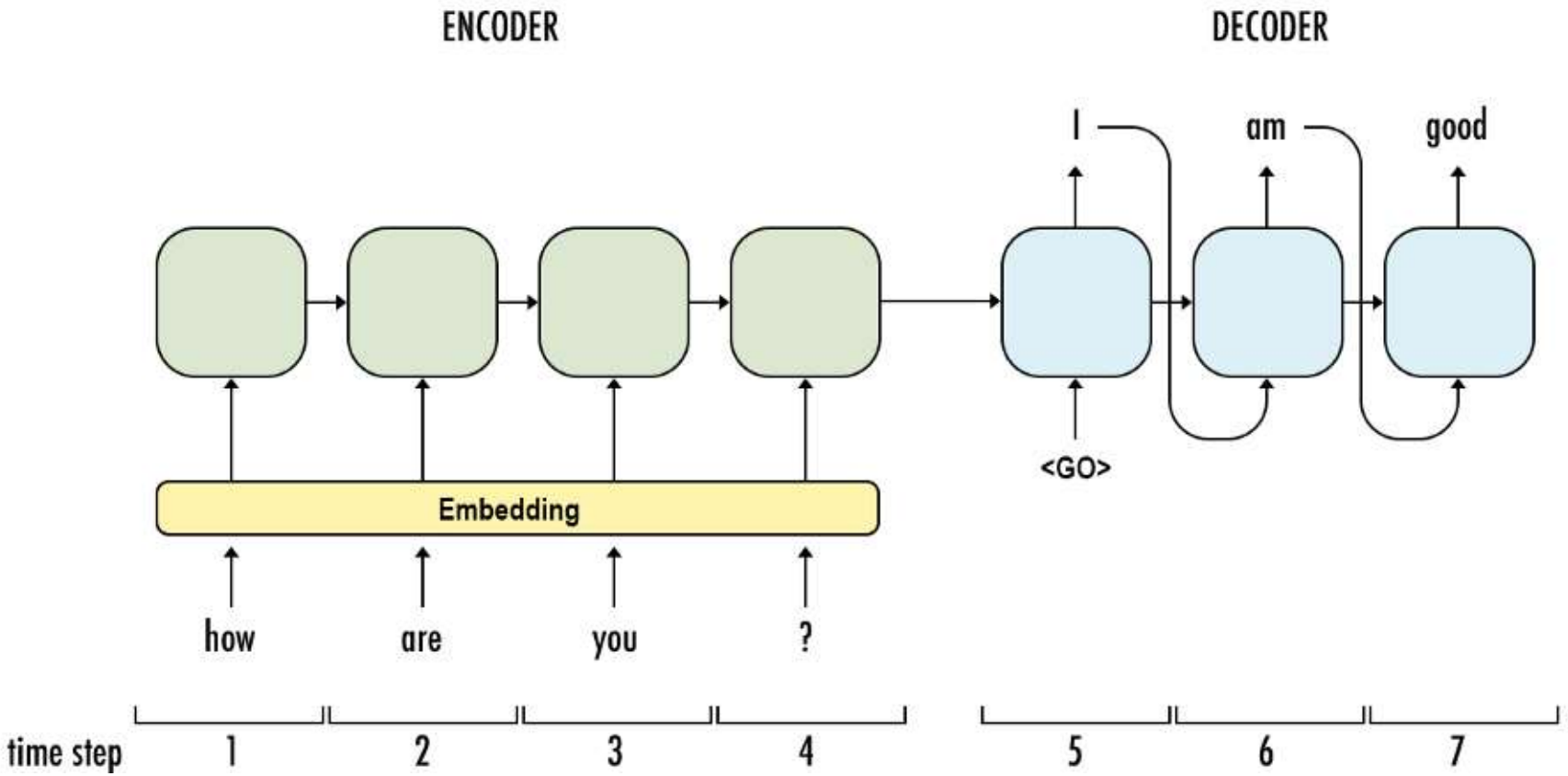
انواع شبکه‌های ترتیبی (یادآوری)



مدل‌های Seq2Seq



مدل‌های Seq2Seq



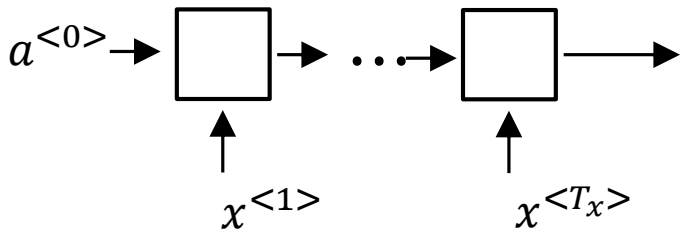
مدل‌های Seq2Seq

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

Jane visite l'Afrique en septembre

Jane is visiting Africa in September.

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



[Sutskever et al., 2014. Sequence to sequence learning with neural networks]
[Cho et al., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation]

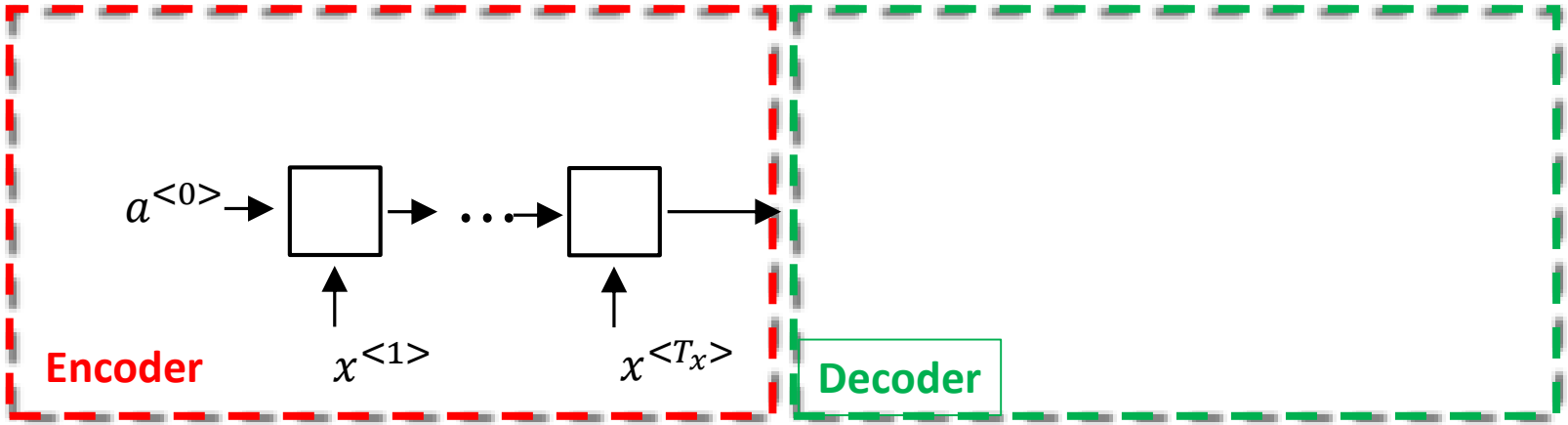
مدل‌های Seq2Seq

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

Jane visite l'Afrique en septembre

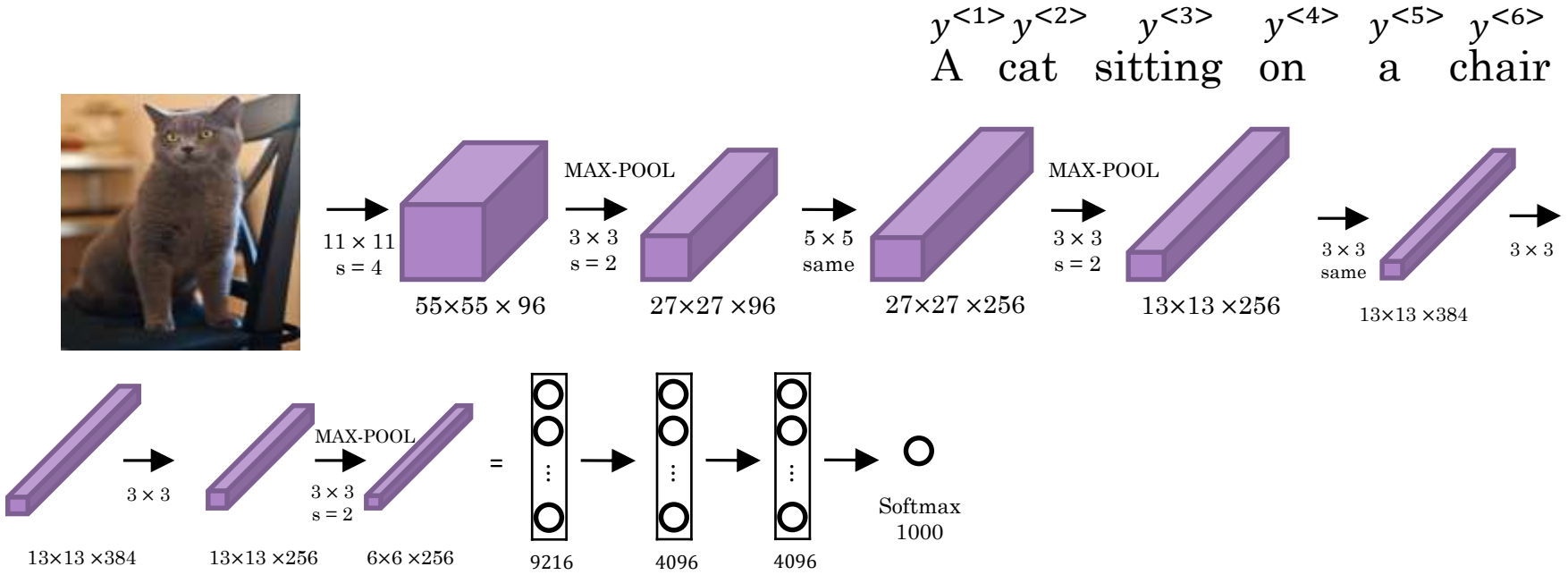
Jane is visiting Africa in September.

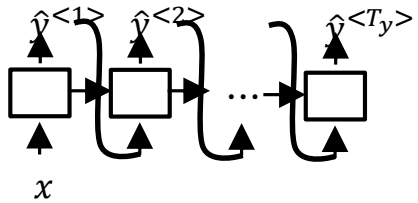
$y^{<1>}$ $y^{<2>}$ $y^{<3>}$ $y^{<4>}$ $y^{<5>}$ $y^{<6>}$



[Sutskever et al., 2014. Sequence to sequence learning with neural networks]
[Cho et al., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation]

Image captioning

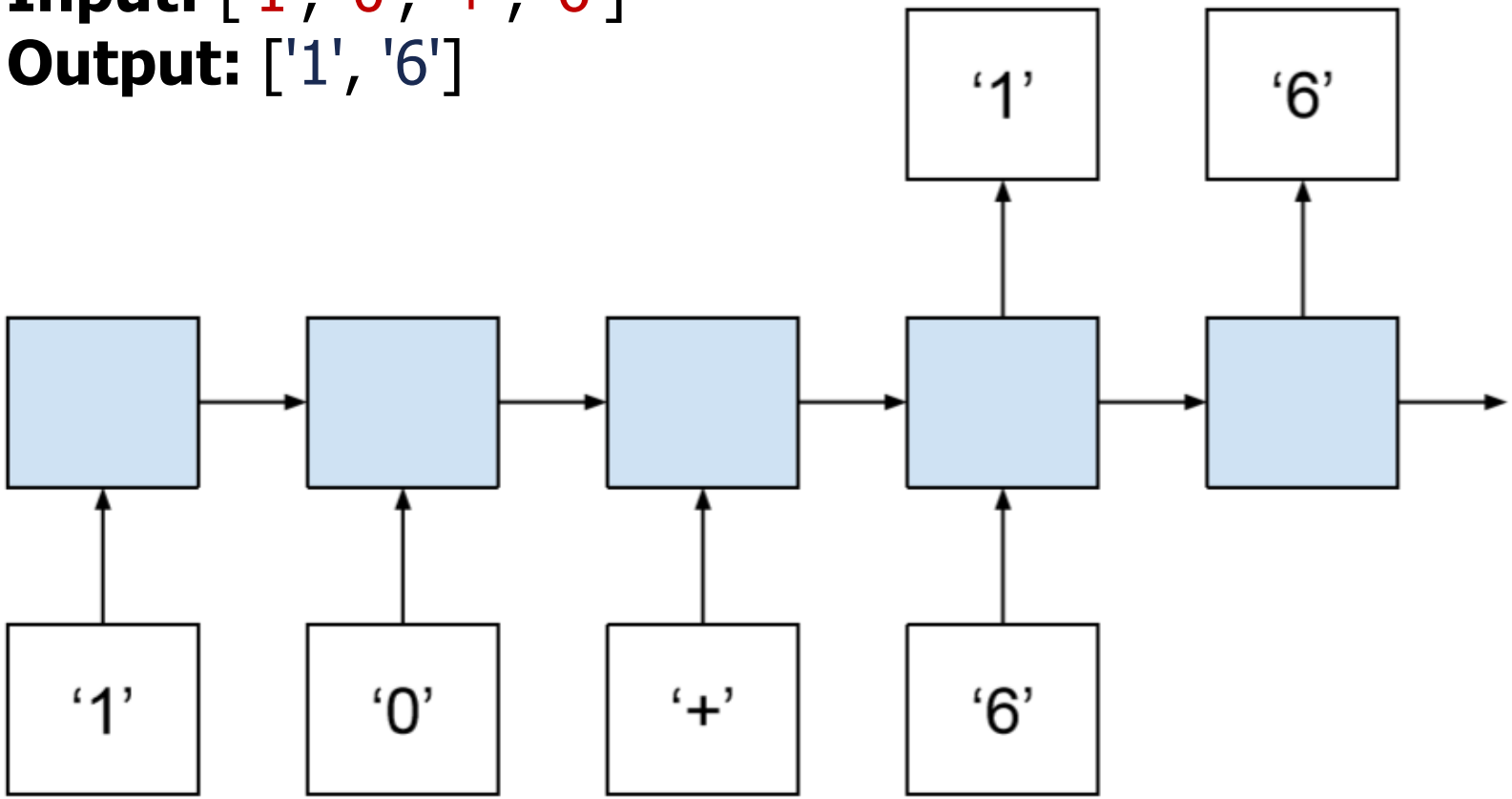




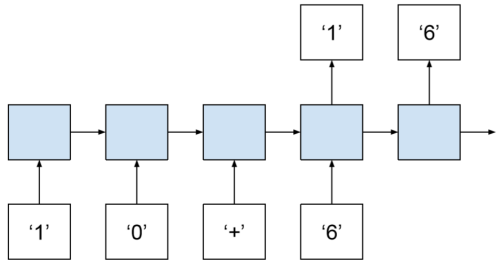
[Mao et. al., 2014. Deep captioning with multimodal recurrent neural networks]
[Vinyals et. al., 2014. Show and tell: Neural image caption generator]
[Karpathy and Li, 2015. Deep visual-semantic alignments for generating image descriptions]

مدل‌های Seq2Seq

Input: ['1', '0', '+', '6']
Output: ['1', '6']



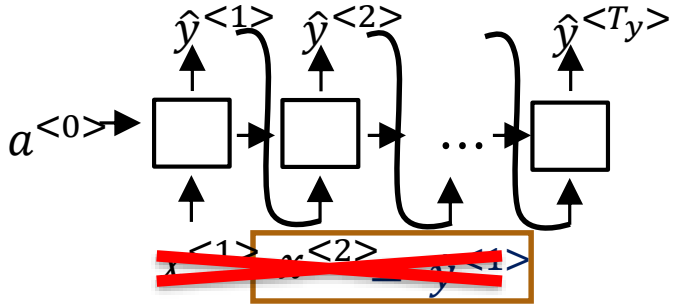
فهم رشته ای از اعمال ریاضی با seq2seq



09_dd_numbers_with_seq2seq

ترجمه ماشینی
به عنوان یک مدل زبان شرطی!

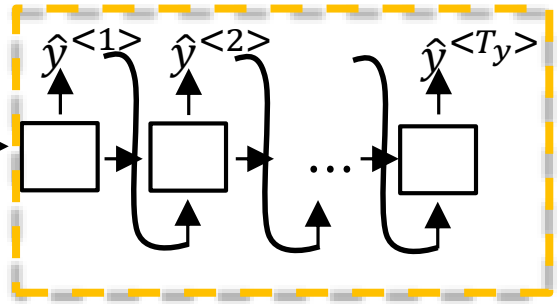
Language model:



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

ترجمه ماشینی
به عنوان یک مدل زبان شرطی!

Language model:

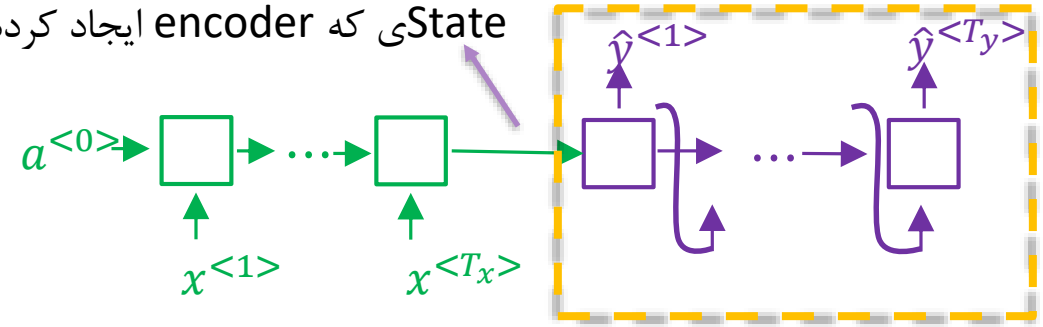


$$p(y^{<1>}, \dots, y^{<T_y>})$$

وکتور ۰

State ی که encoder ایجاد کرده

Machine translation:



$$p(y^{<1>}, \dots, y^{<T_y>} \mid x^{<1>}, \dots, x^{<T_x>})$$

Conditional language model

پیدا کردن محتمل‌ترین ترجمه!

Jane visite l’Afrique en septembre.

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

English

French

→ Jane is visiting Africa in September.

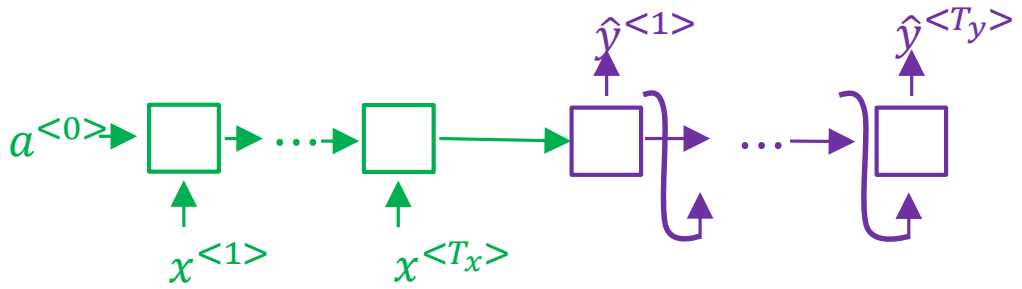
→ Jane is going to be visiting Africa in September.

→ In September, Jane will visit Africa.

→ Her African friend welcomed Jane in September.

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

چرا جست‌وجوی حریصانه (greedy search) مناسب نیست؟!



$$\arg\max_{y} P(\hat{y}^{<1>}, \hat{y}^{<2>}, \ldots, \hat{y}^{<T_y>} \mid x)$$

→ Jane is visiting Africa in September.

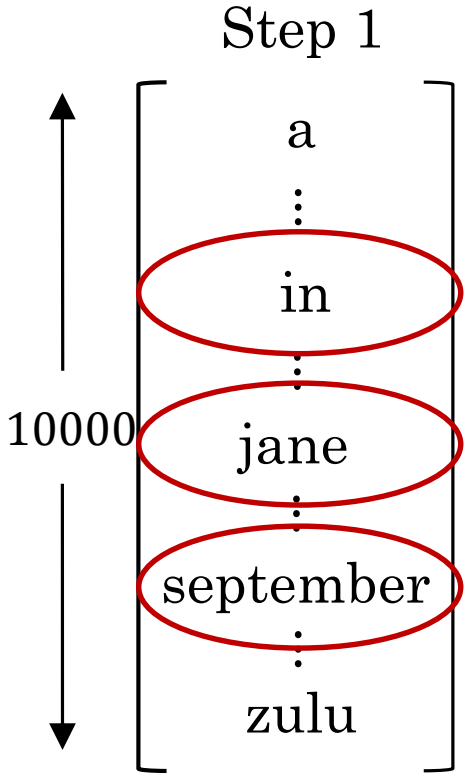
→ Jane is going to be visiting Africa in September.

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

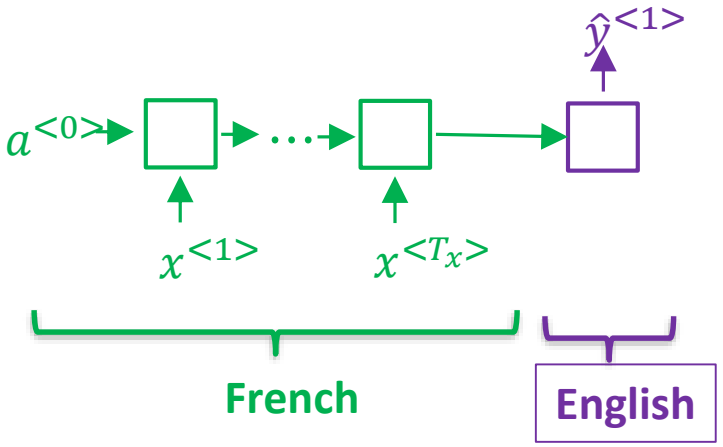
Beam search

Beam search الگوریتم

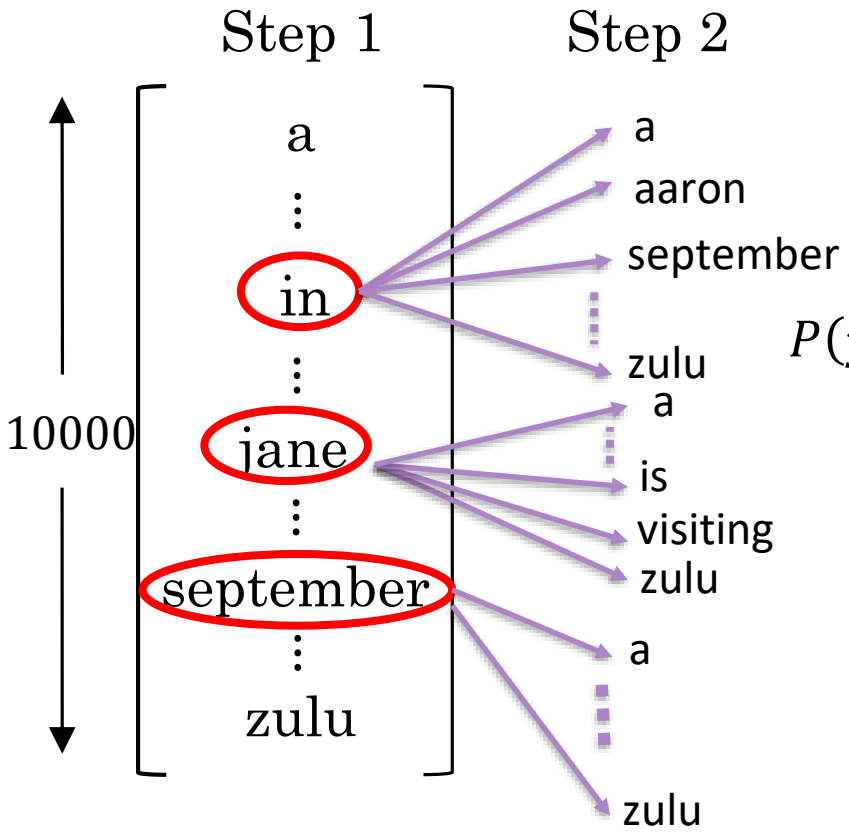
B = 3 (Beam width)



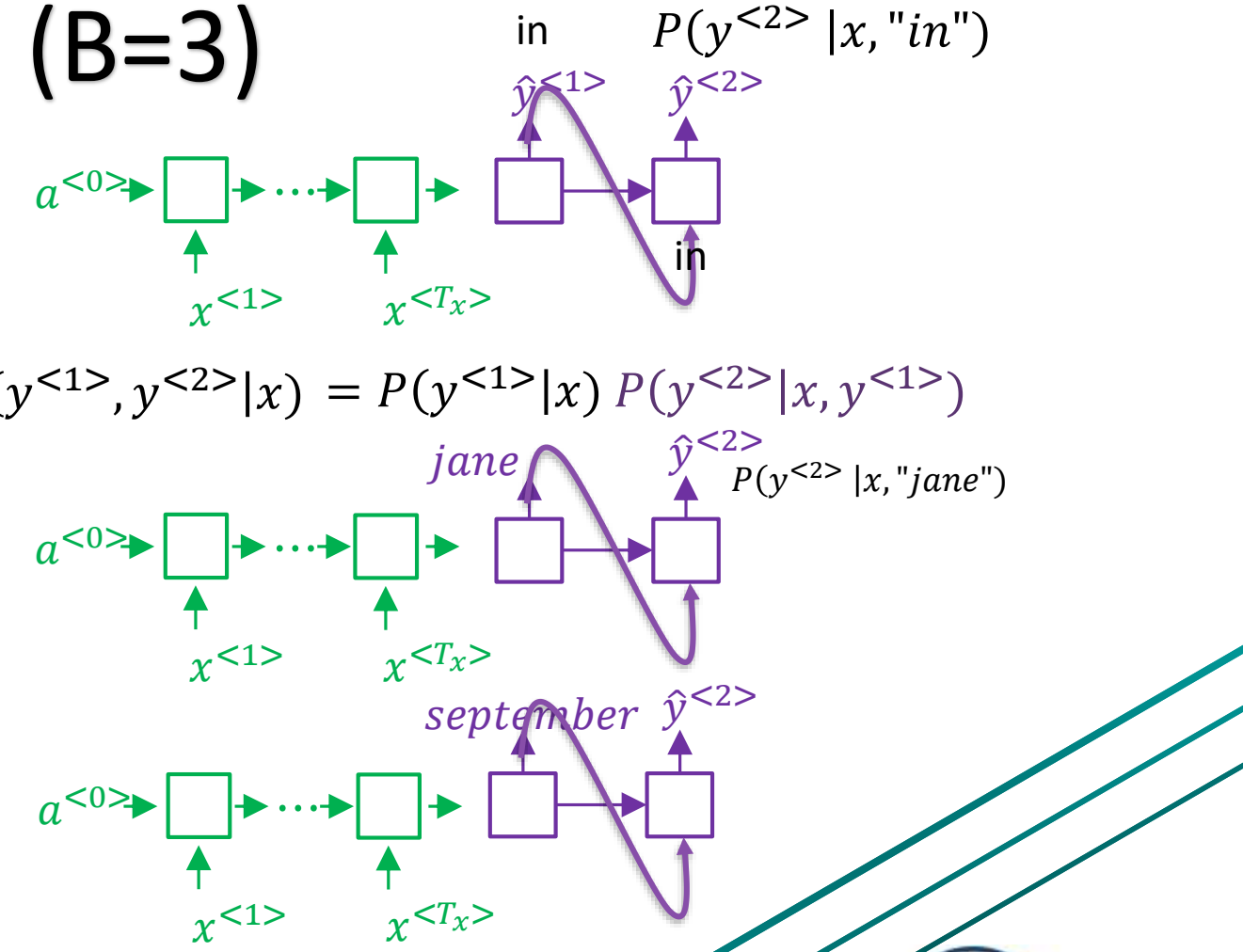
$P(y^{<1>} \mid x)$



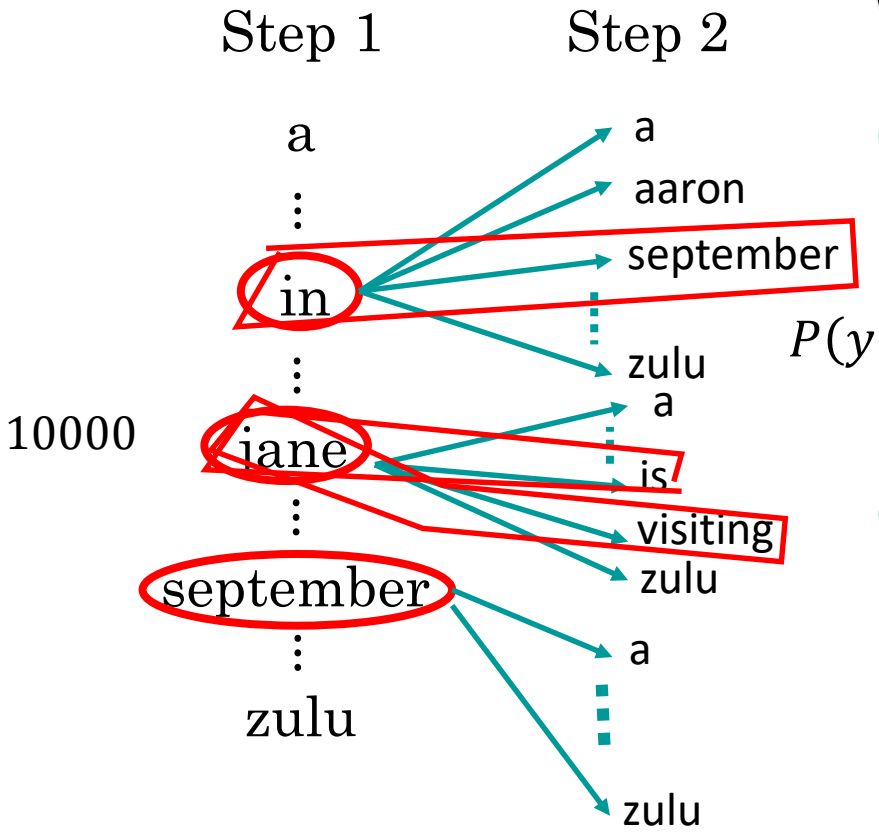
Beam search الگوریتم



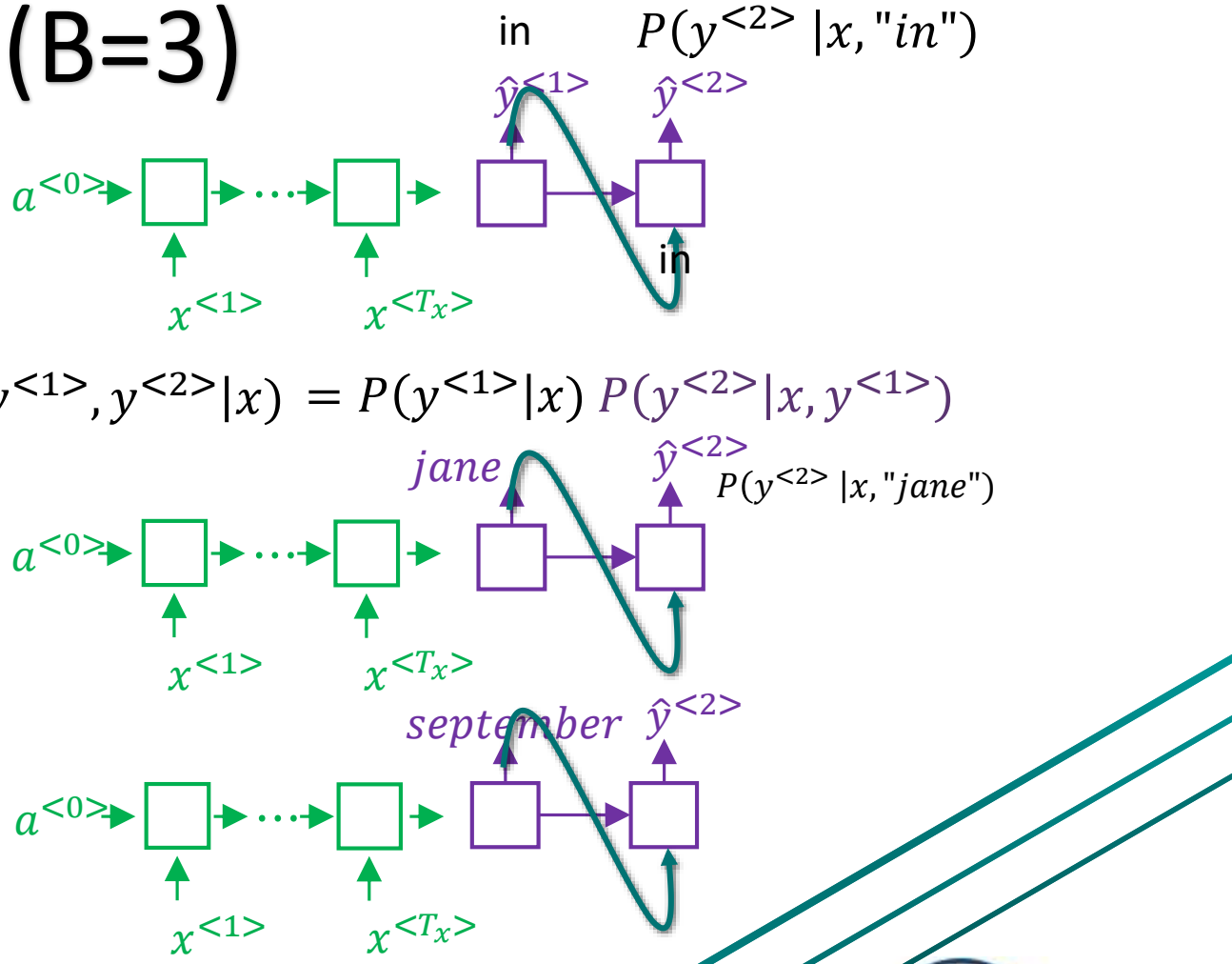
(B=3)



Beam search الگوریتم

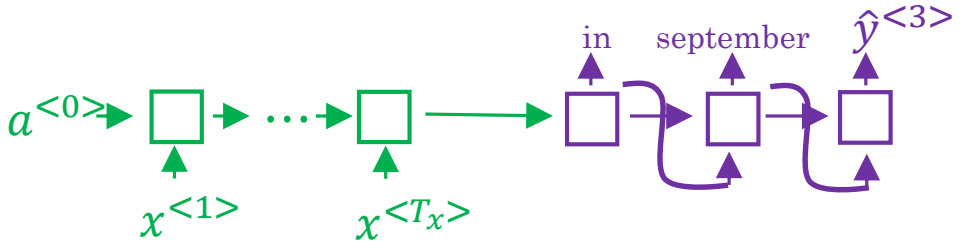


(B=3)

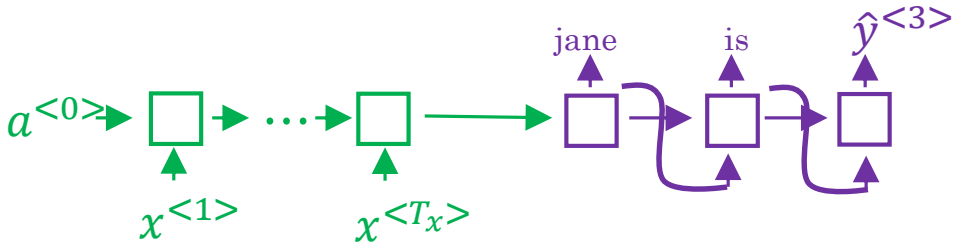


Beam search الگوریتم

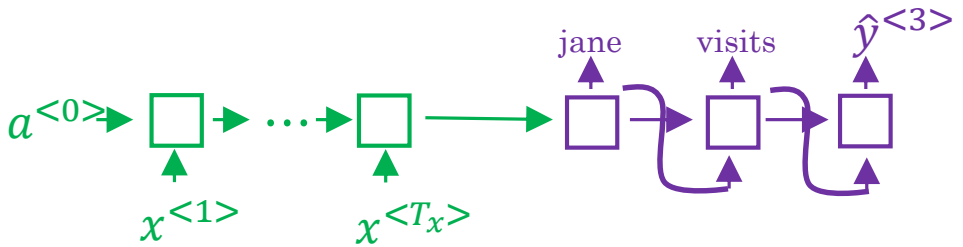
in september



jane is



jane visits



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

jane visits africa in september. <EOS>

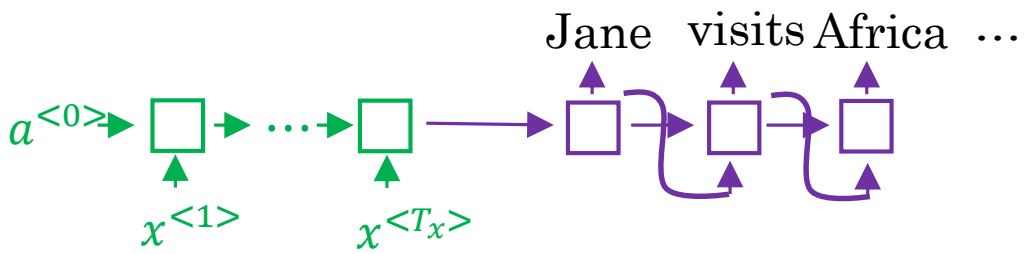
Error analysis on beam search

تحلیل خطا!

Jane visite l’Afrique en septembre.

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

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



- ❑ RNN
- ❑ Beam search

تحلیل خطا!

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

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

Case 1: $(P(y^* \mid X) > P(\hat{y} \mid X))$

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

Conclusion: Beam search is at fault.

Case 2: $(P(y^* \mid X) <= P(\hat{y} \mid X))$

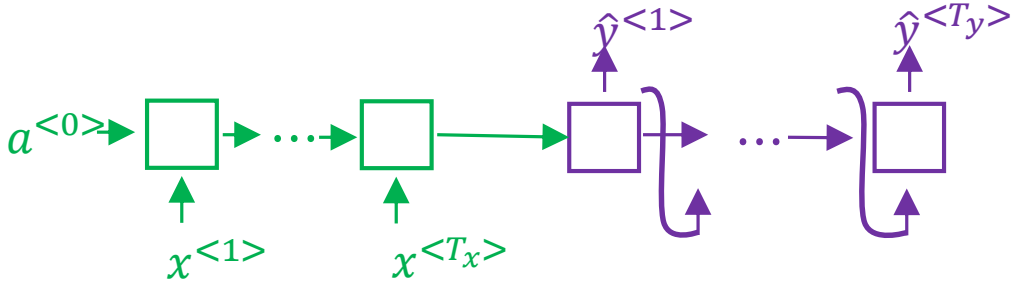
y^* is a better translation than $\hat{y}$. But RNN predicted $P(y^*|x) < P(\hat{y}|x)$.

Conclusion: RNN model is at fault.

Attention and Memory

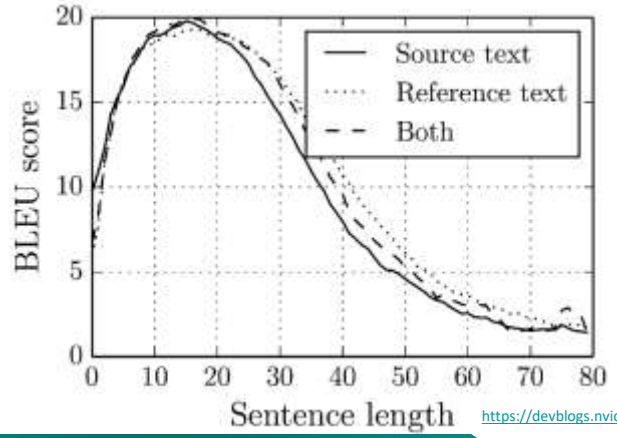
مشکل Sequence های طولانی!

یک مترجم تا انتهای کتاب نمیخونه سپس ترجمه کنه!



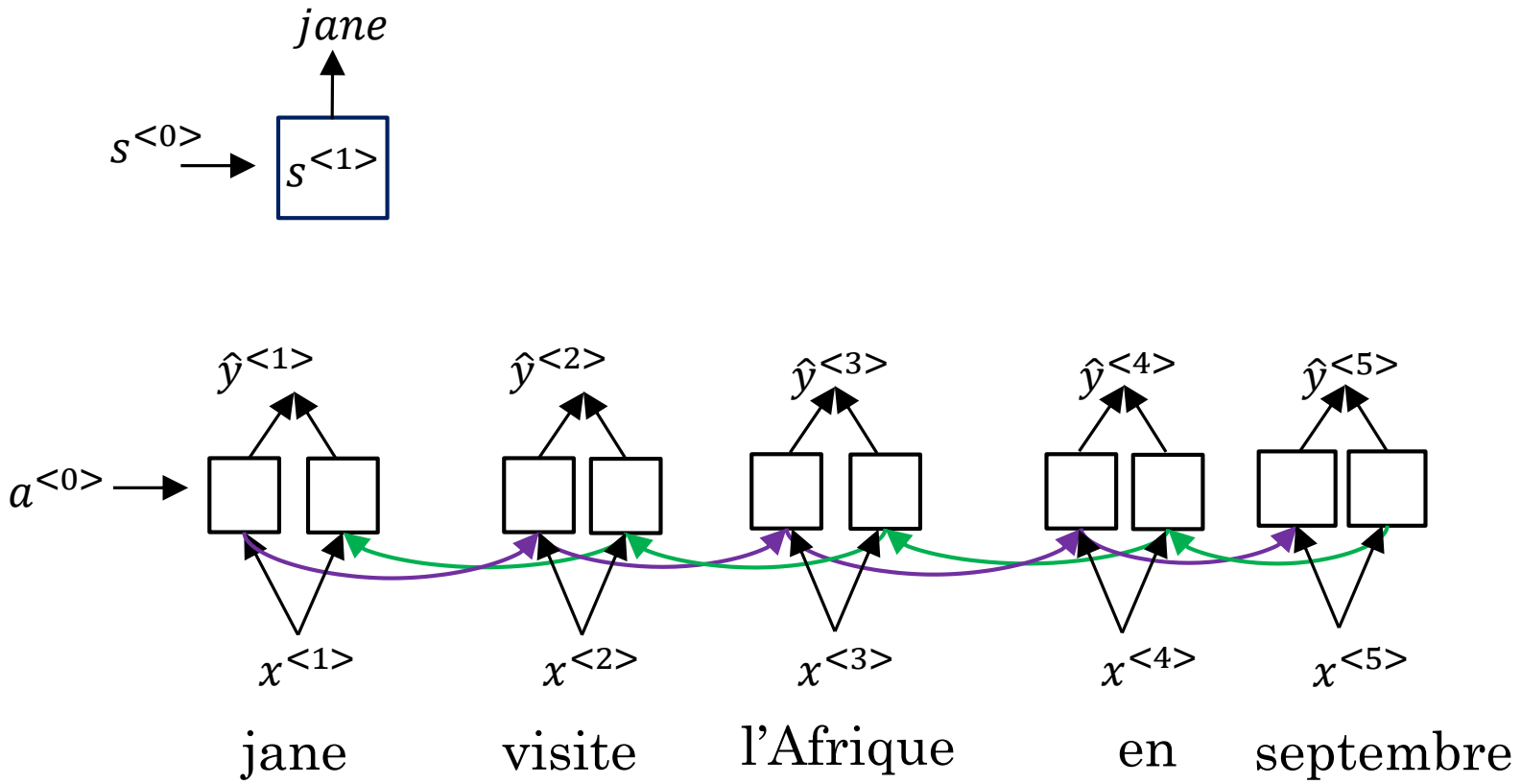
Jane s'est rendue en Afrique en septembre dernier, a apprécié la culture et a rencontré beaucoup de gens merveilleux; elle est revenue en parlant comment son voyage était merveilleux, et elle me tente d'y aller aussi.

Jane went to Africa last September, and enjoyed the culture and met many wonderful people; she came back raving about how wonderful her trip was, and is tempting me to go too.



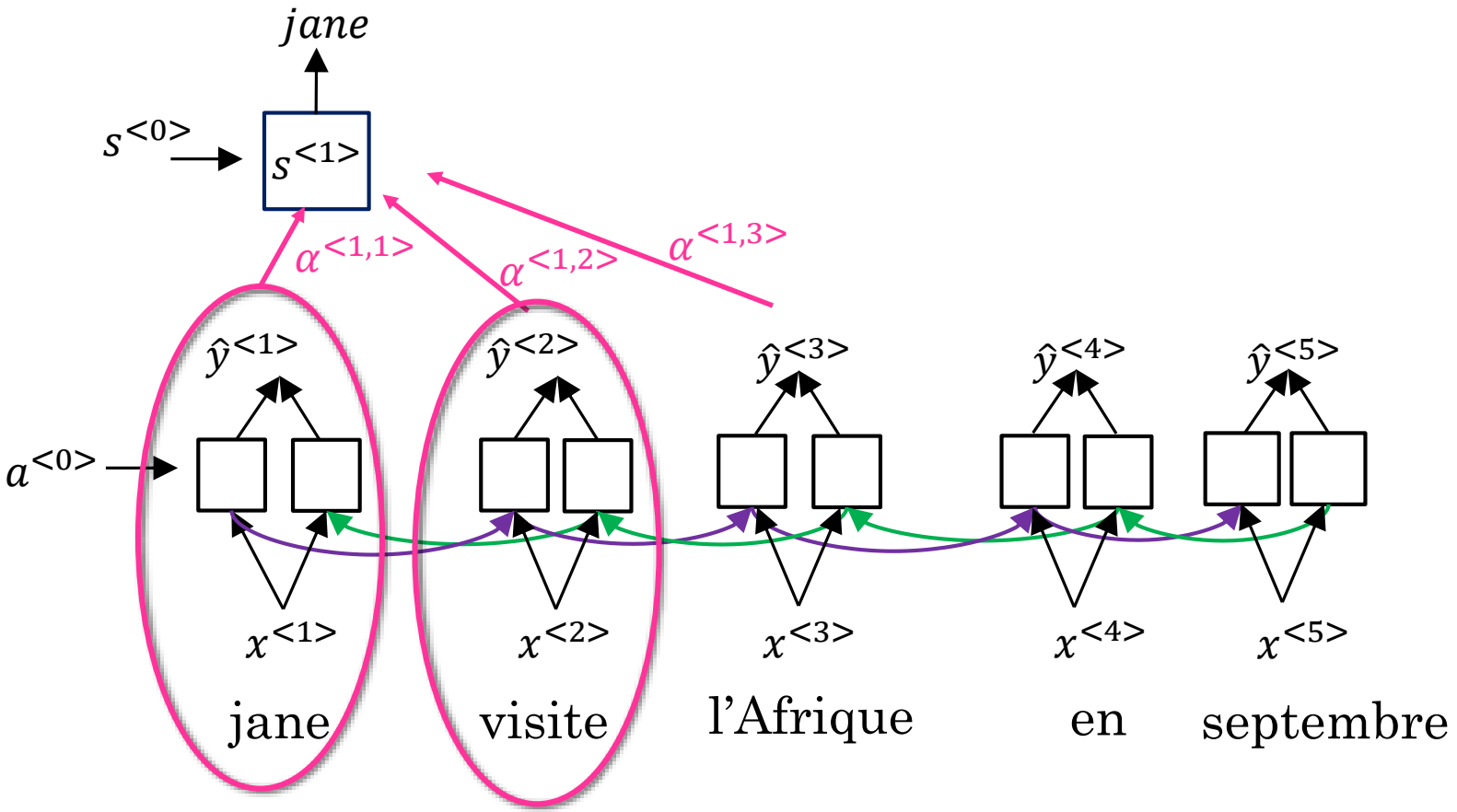
https://devblogs.nvidia.com/introduction-neural-machine-translation-gpus-part-3/figure1_bleuscore_vs_sentencelength/

توجه (Attention)



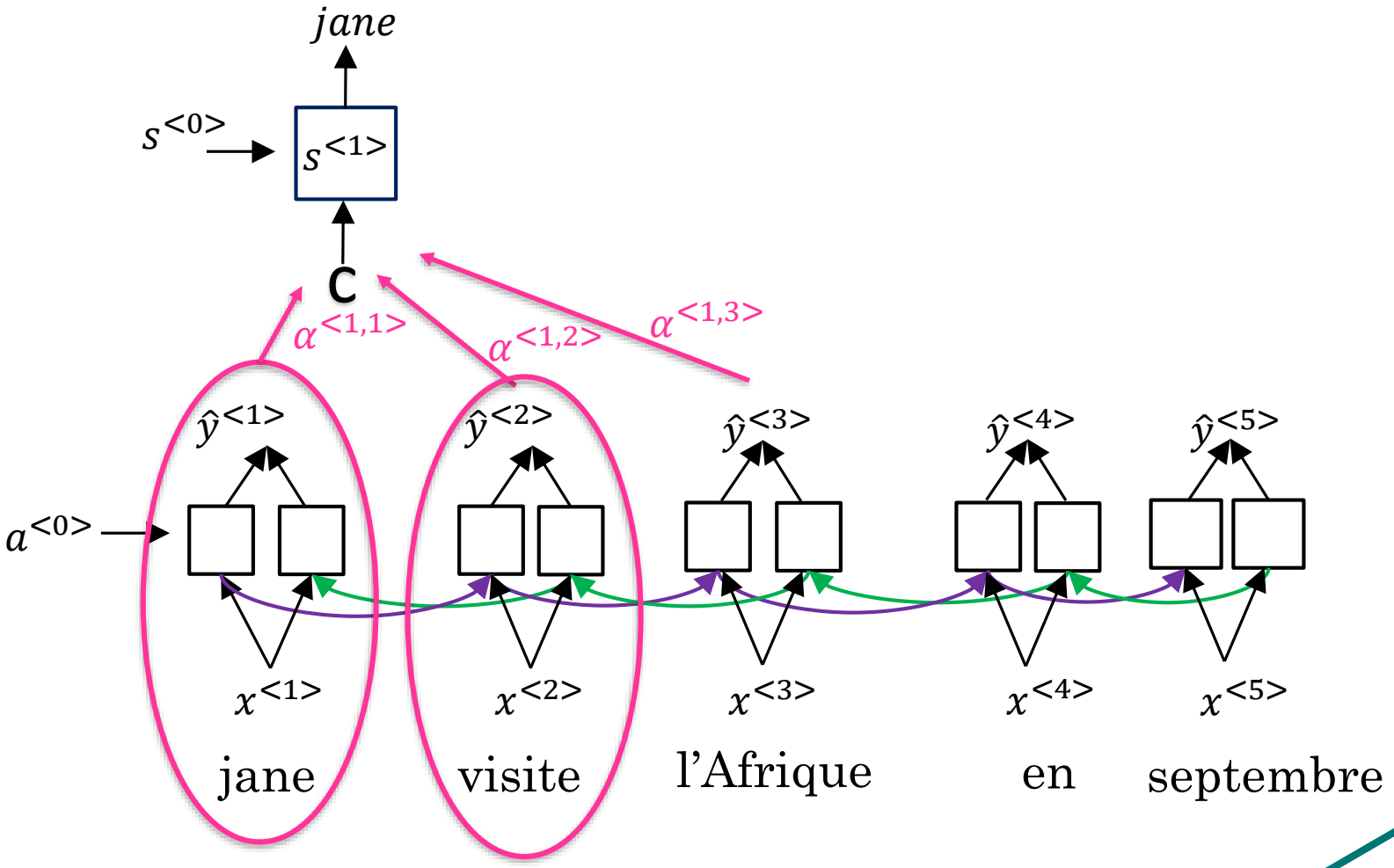
[Bahdanau et. al., 2014. Neural machine translation by jointly learning to align and translate]

توجه (Attention)



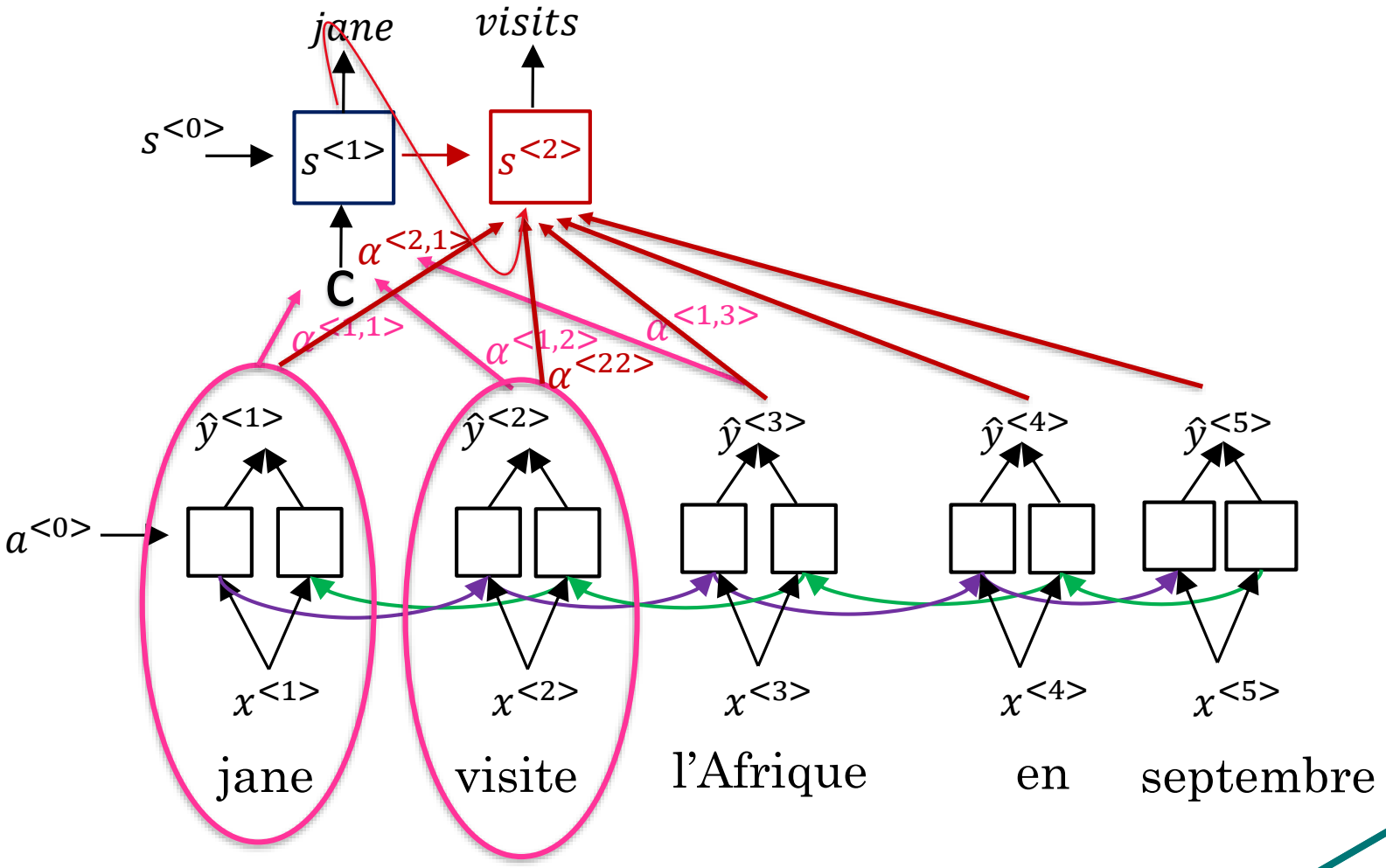
[Bahdanau et. al., 2014. Neural machine translation by jointly learning to align and translate]

توجه (Attention)



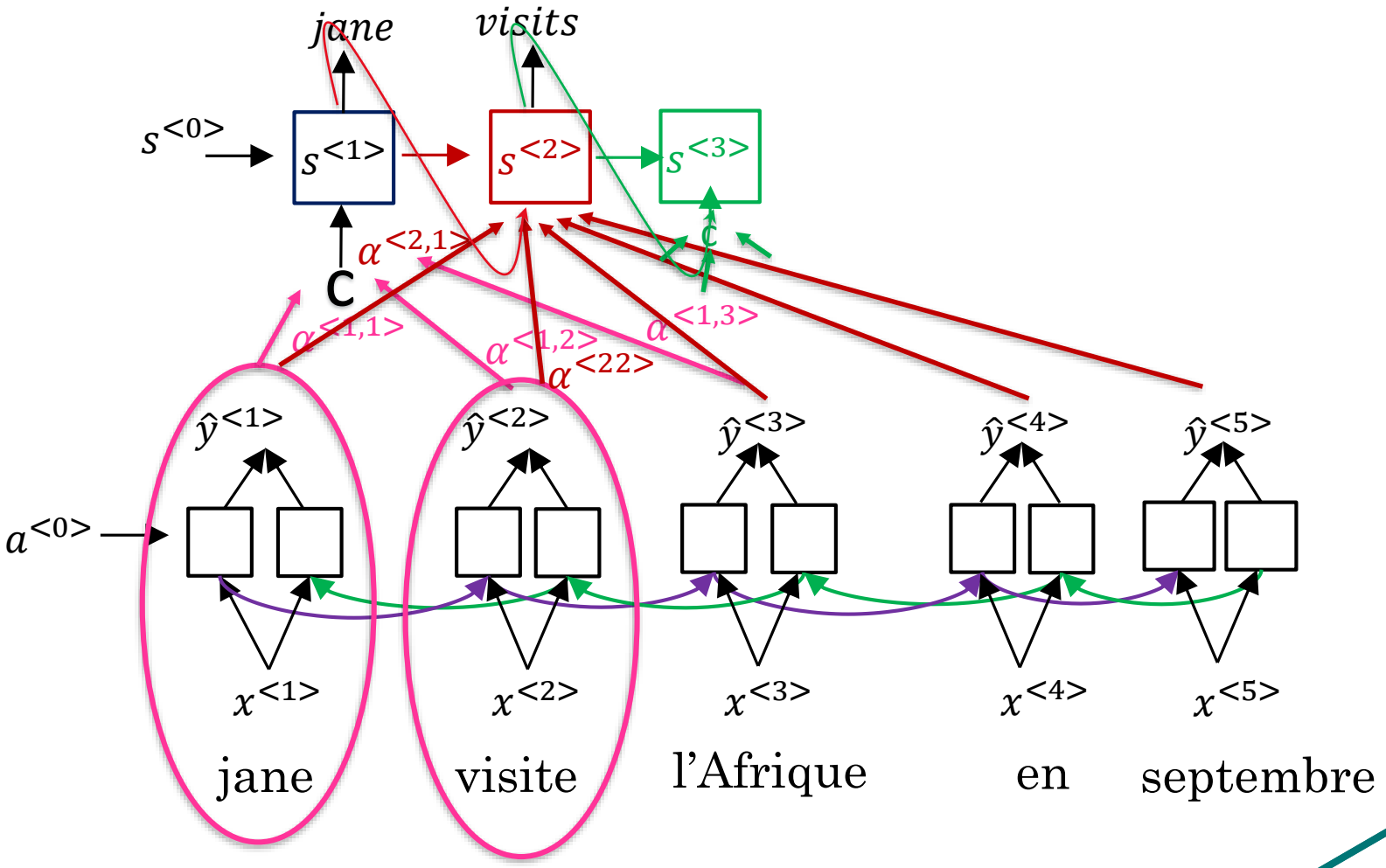
[Bahdanau et. al., 2014. Neural machine translation by jointly learning to align and translate]

توجه (Attention)



[Bahdanau et. al., 2014. Neural machine translation by jointly learning to align and translate]

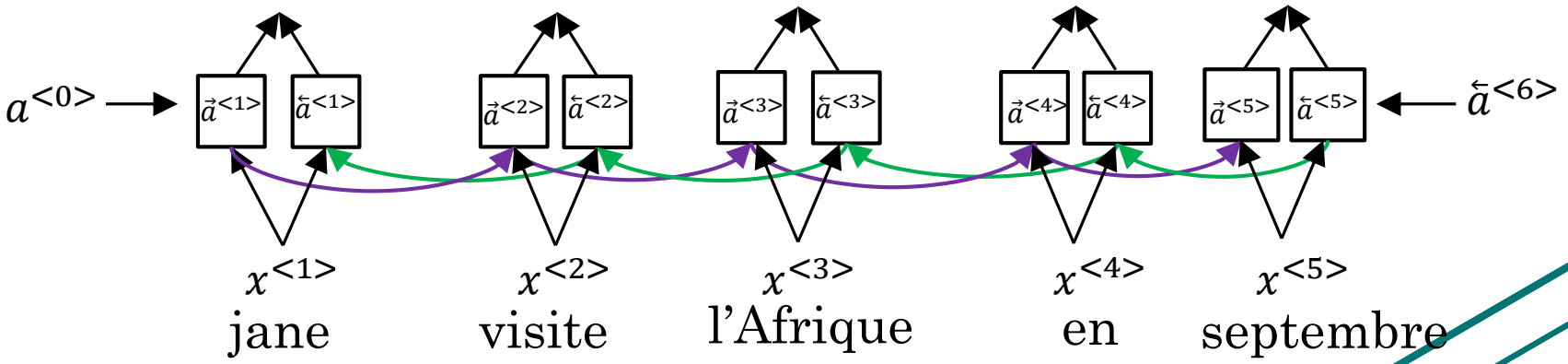
توجه (Attention)



[Bahdanau et. al., 2014. Neural machine translation by jointly learning to align and translate]

توجه (Attention)

$$a^{<t>} = (\vec{a}^{<t>}, \overleftarrow{a}^{<t>})$$



[Bahdanau et. al., 2014. Neural machine translation by jointly learning to align and translate]

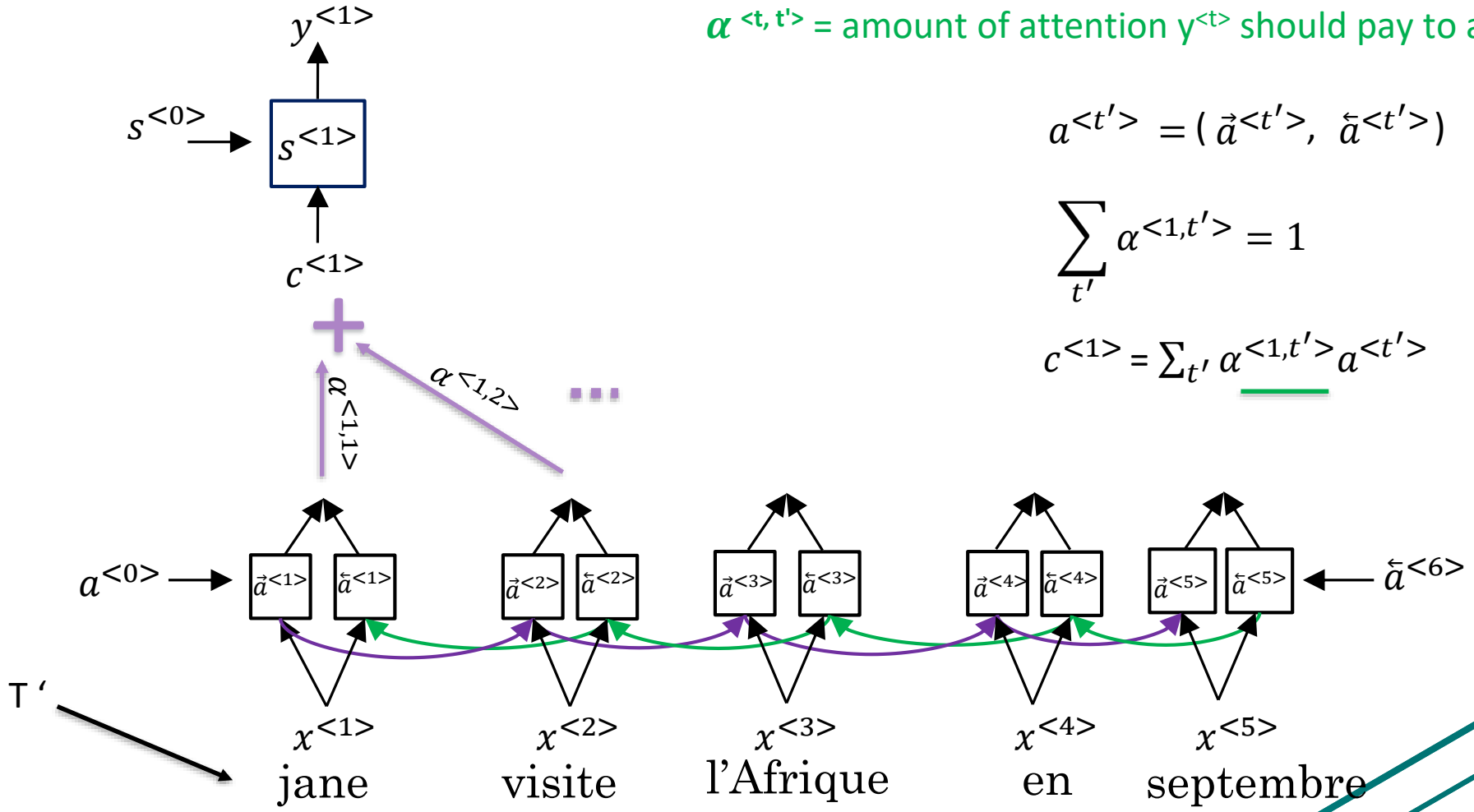
توجه (Attention)

$\alpha^{<t,t'>}$ = amount of attention $y^{<t>}$ should pay to $a^{<t'>}$

$$a^{<t'>} = (\vec{a}^{<t'>}, \overleftarrow{a}^{<t'>})$$

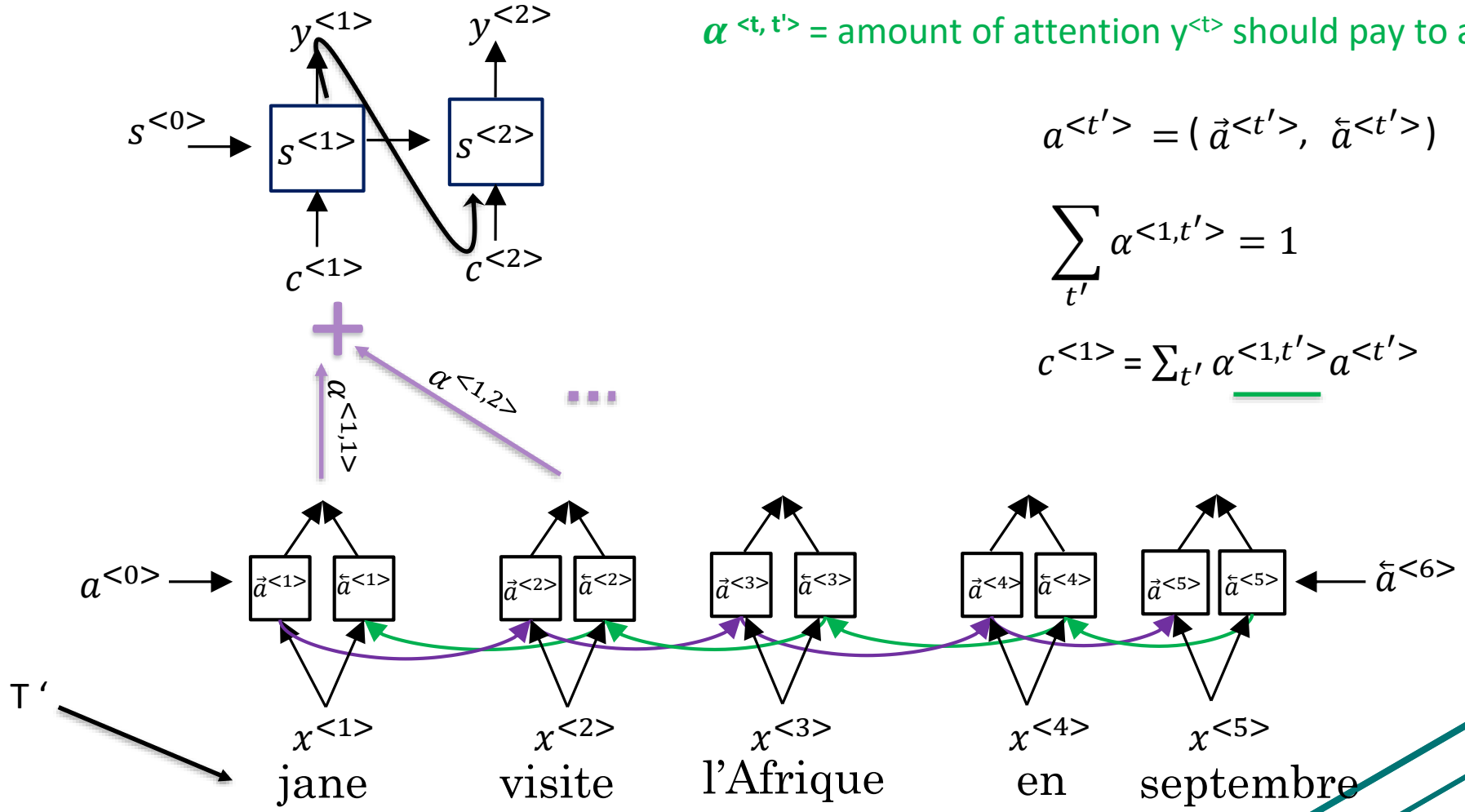
$$\sum_{t'} \alpha^{<1,t'>} = 1$$

$$c^{<1>} = \sum_{t'} \alpha^{<1,t'>} a^{<t'>}$$



[Bahdanau et. al., 2014. Neural machine translation by jointly learning to align and translate]

توجه (Attention)



$\alpha^{<t,t'>}$ = amount of attention $y^{<t>}$ should pay to $a^{<t'>}$

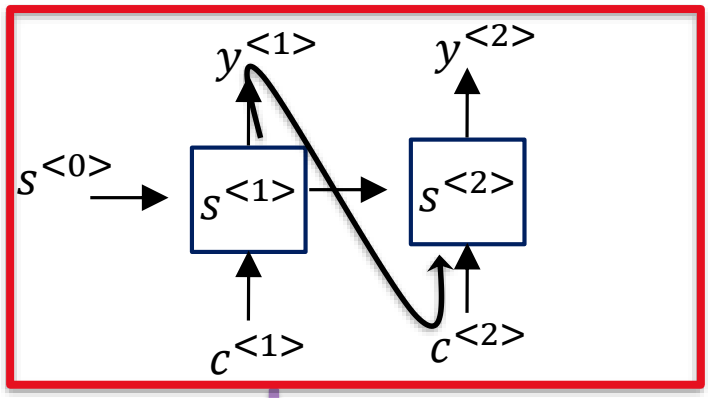
$$a^{<t'>} = (\vec{a}^{<t'>}, \overleftarrow{a}^{<t'>})$$

$$\sum_{t'} \alpha^{<1,t'>} = 1$$

$$c^{<1>} = \sum_{t'} \alpha^{<1,t'>} a^{<t'>}$$

[Bahdanau et. al., 2014. Neural machine translation by jointly learning to align and translate]

توجه (Attention)



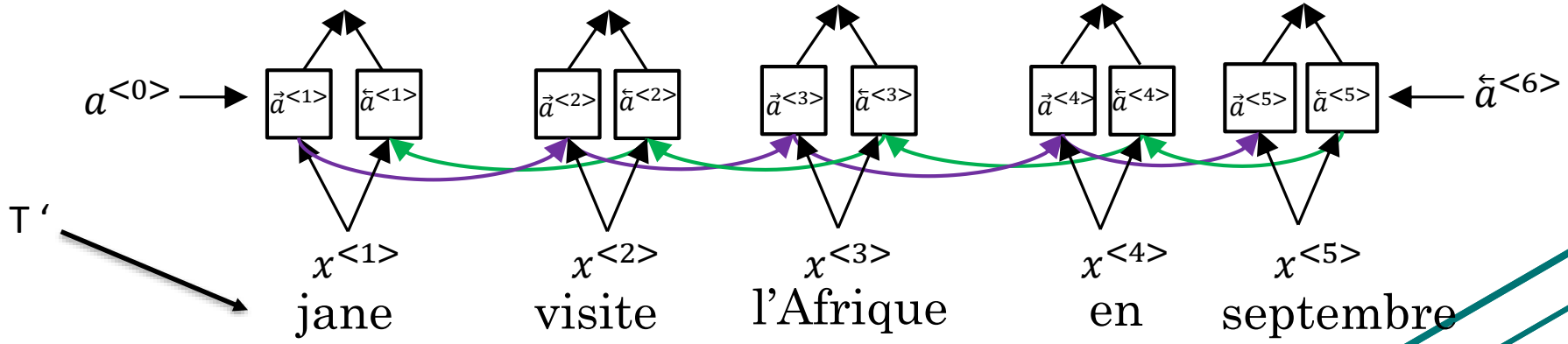
یک شبکه بازگشتی ساده

$\alpha^{<t,t'>}$ = amount of attention $y^{<t>}$ should pay to $a^{<t'>}$

$$a^{<t'>} = (\vec{a}^{<t'>}, \overleftarrow{a}^{<t'>})$$

$$\sum_{t'} \alpha^{<1,t'>} = 1$$

$$c^{<1>} = \sum_{t'} \alpha^{<1,t'>} a^{<t'>}$$

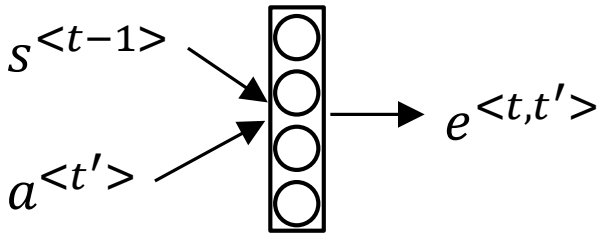


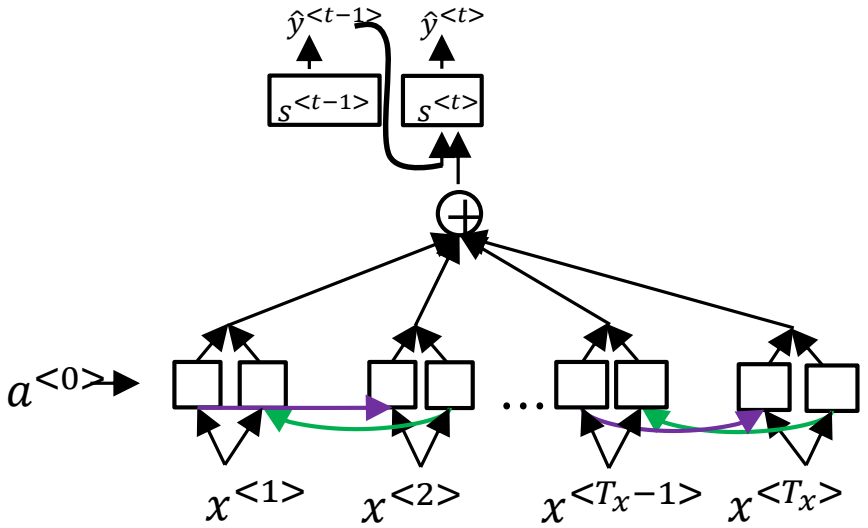
[Bahdanau et. al., 2014. Neural machine translation by jointly learning to align and translate]

$\alpha^{<t,t'>}$ - محاسبه توجه (Attention)

$\alpha^{<t,t'>}$ = amount of attention $y^{<t>}$ should pay to $a^{<t'>}$

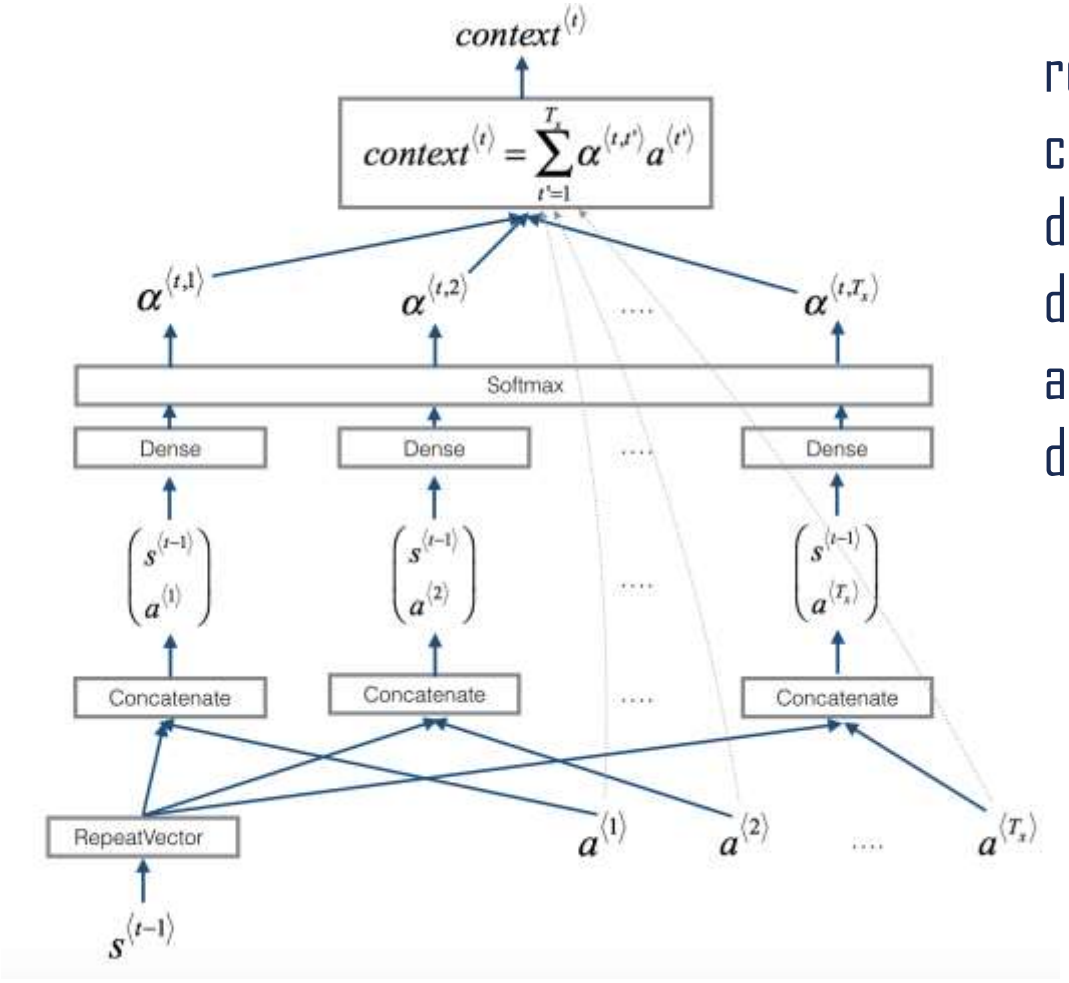
$$\alpha^{<t,t'>} = \frac{\exp(e^{<t,t'>})}{\sum_{t'=1}^{T_x} \exp(e^{<t,t'>})}$$





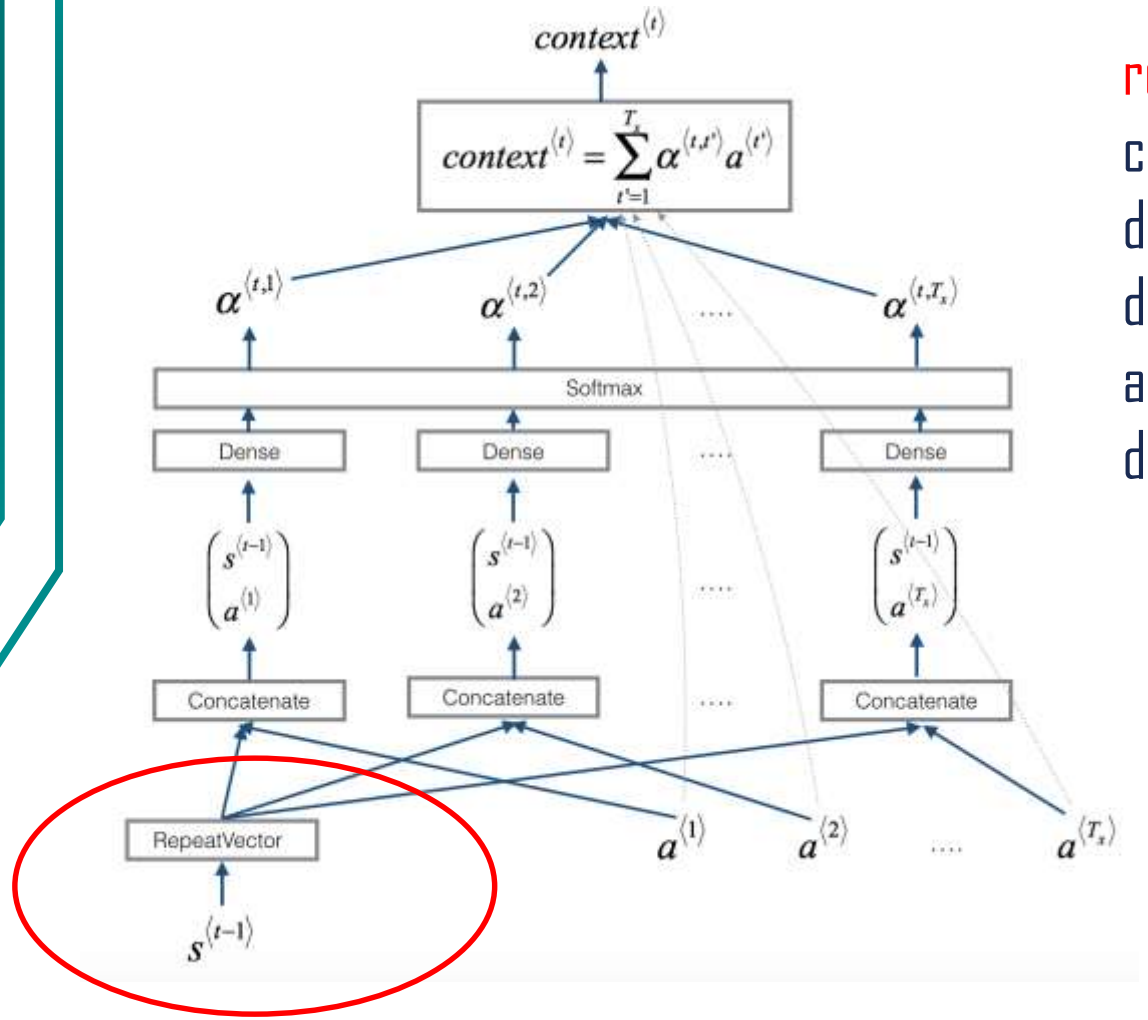
[Bahdanau et. al., 2014. Neural machine translation by jointly learning to align and translate]
[Xu et. al., 2015. Show, attend and tell: Neural image caption generation with visual attention]

$\alpha^{<t,t'>}$ - محاسبه توجه (Attention)



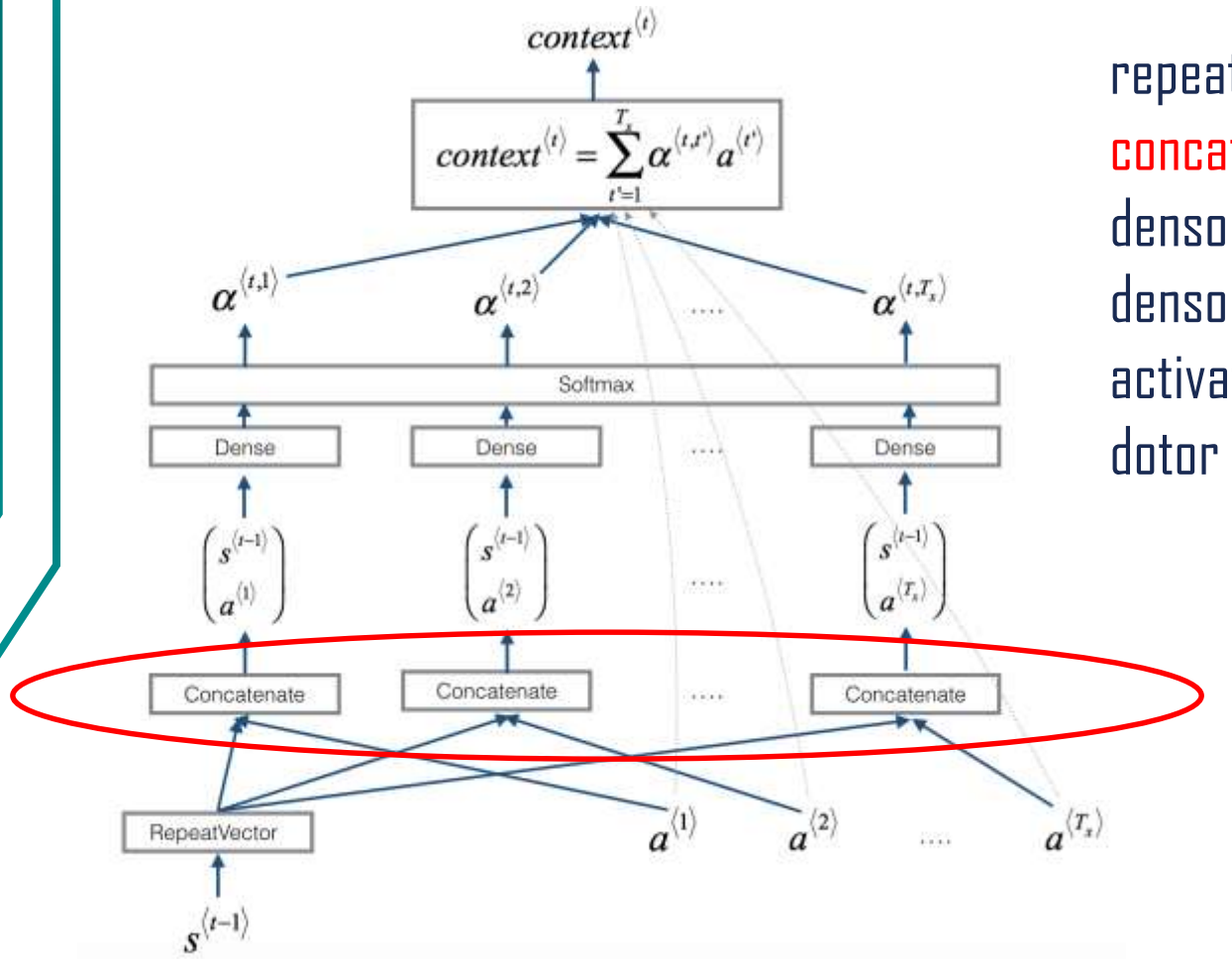

```
repeator = RepeatVector(Tx)
concatenator = Concatenate(axis=-1)
densor1 = Dense(10, activation = "tanh")
densor2 = Dense(1, activation = "relu")
activator = Activation(softmax, name='attention_weights')
dotor = Dot(axes = 1)
```

$\alpha^{<t,t'>}$ - محاسبه توجه (Attention)



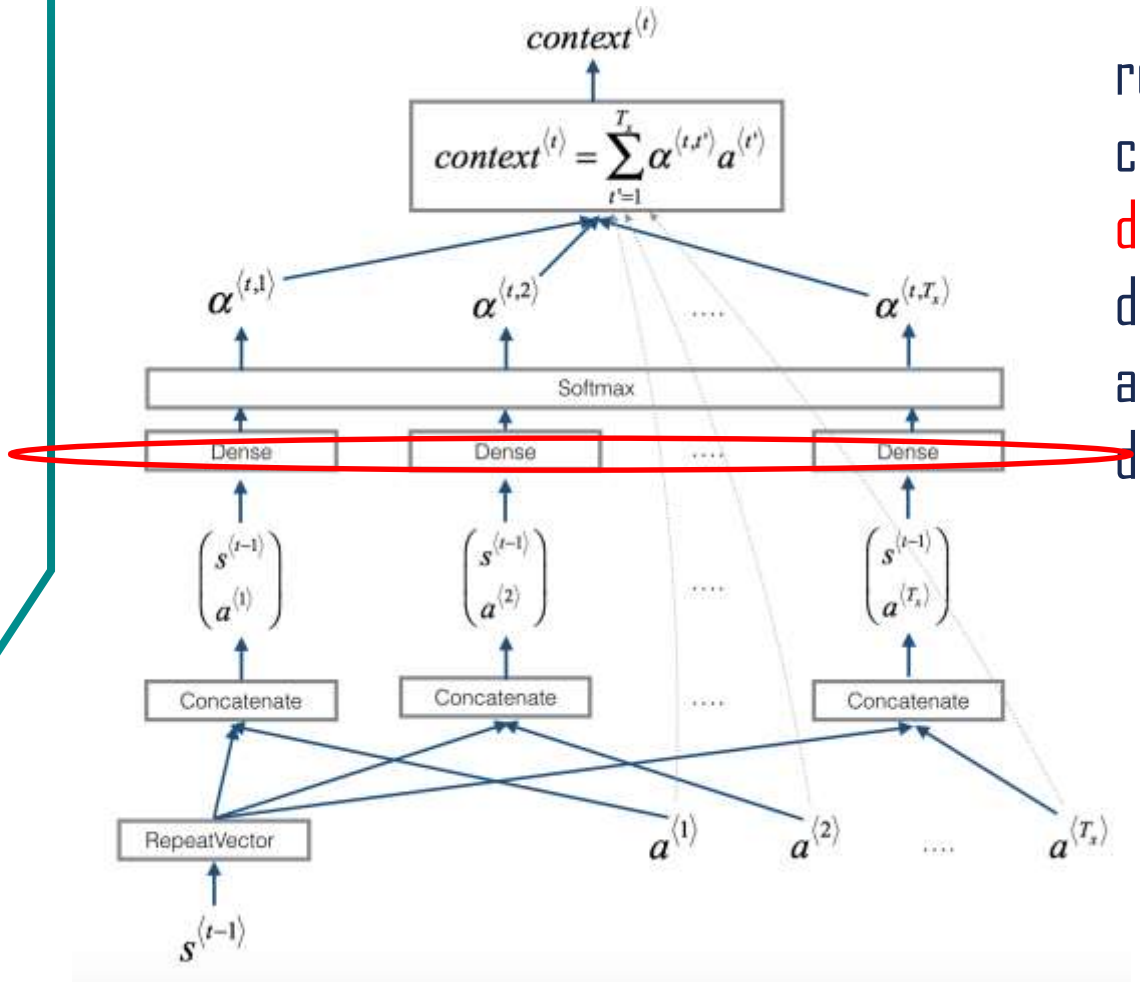

```
repeator = RepeatVector(Tx)
concatenator = Concatenate(axis=-1)
densor1 = Dense(10, activation = "tanh")
densor2 = Dense(1, activation = "relu")
activator = Activation(softmax, name='attention_weights')
dotor = Dot(axes = 1)
```

$\alpha^{<t,t'>}$ - محاسبه توجه (Attention)



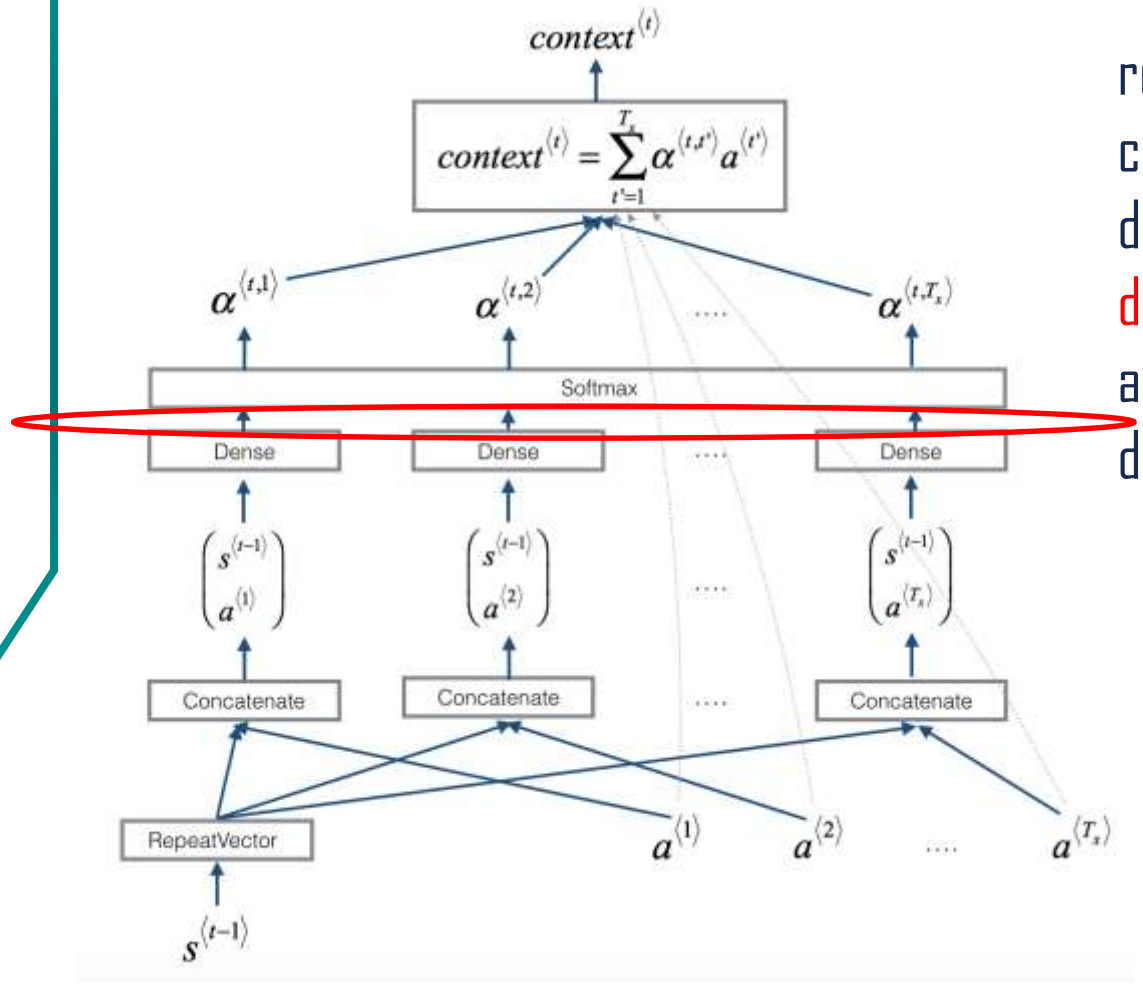

```
repeator = RepeatVector(Tx)
concatenator = Concatenate(axis=-1)
densor1 = Dense(10, activation = "tanh")
densor2 = Dense(1, activation = "relu")
activator = Activation(softmax, name='attention_weights')
dotor = Dot(axes = 1)
```

محاسبه توجه (Attention) - $\alpha^{<t,t'>}$



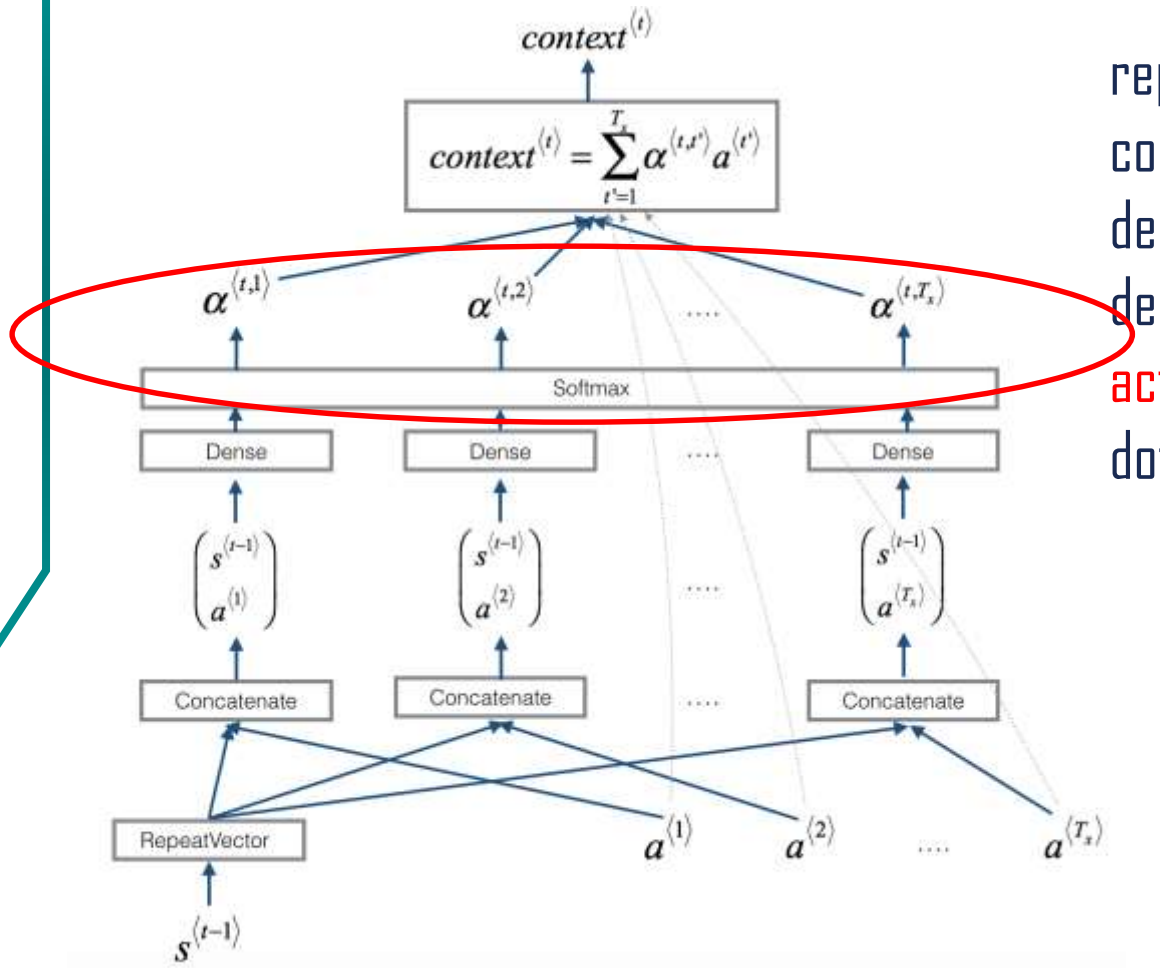

```
repeator = RepeatVector(Tx)
concatenator = Concatenate(axis=-1)
densor1 = Dense(10, activation = "tanh")
densor2 = Dense(1, activation = "relu")
activator = Activation(softmax, name='attention_weights')
dotor = Dot(axes = 1)
```

$\alpha^{<t,t'>}$ - محاسبه توجه (Attention)



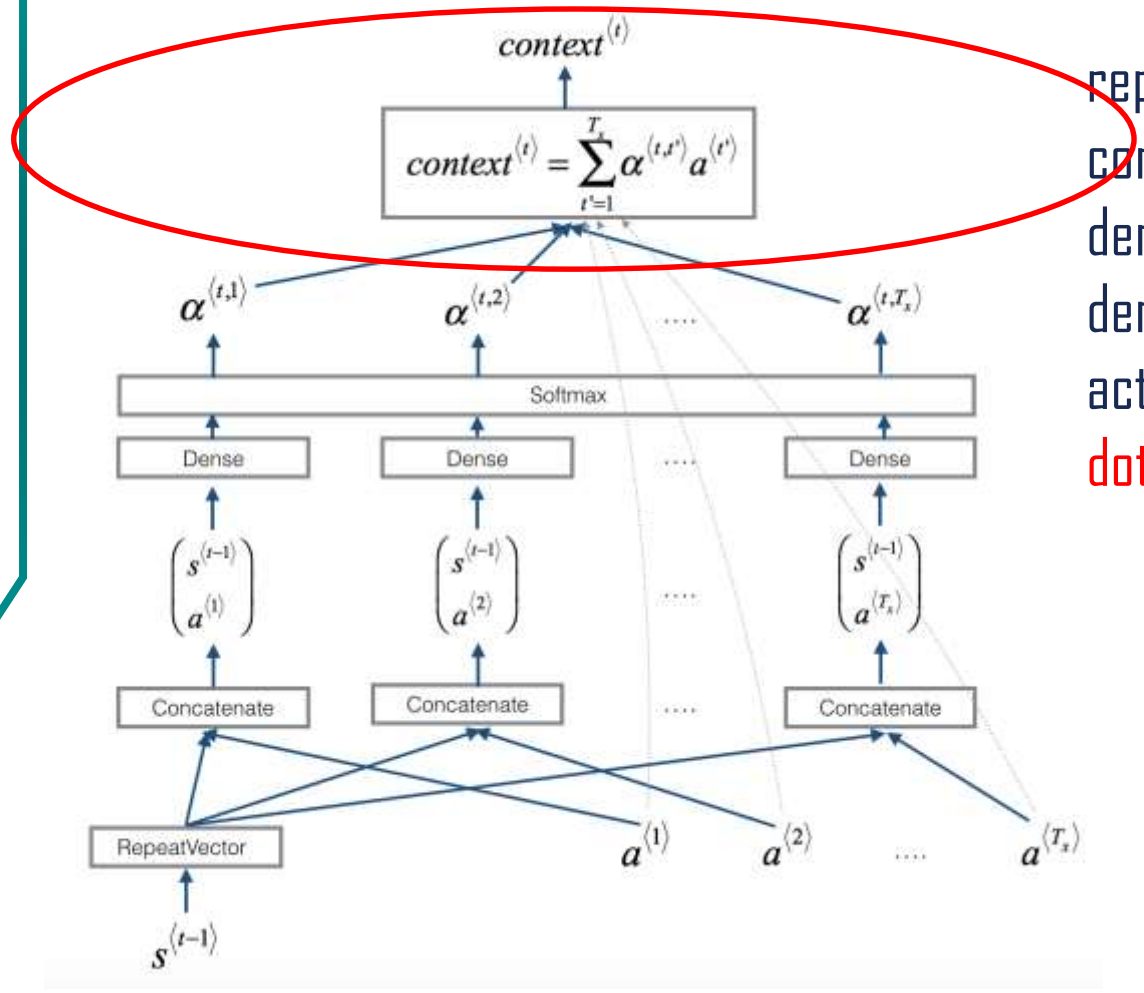

```
repeator = RepeatVector(Tx)
concatenator = Concatenate(axis=-1)
densor1 = Dense(10, activation = "tanh")
densor2 = Dense(1, activation = "relu")
activator = Activation(softmax, name='attention_weights')
dotor = Dot(axes = 1)
```

$\alpha^{<t,t'>}$ - محاسبه توجه (Attention)



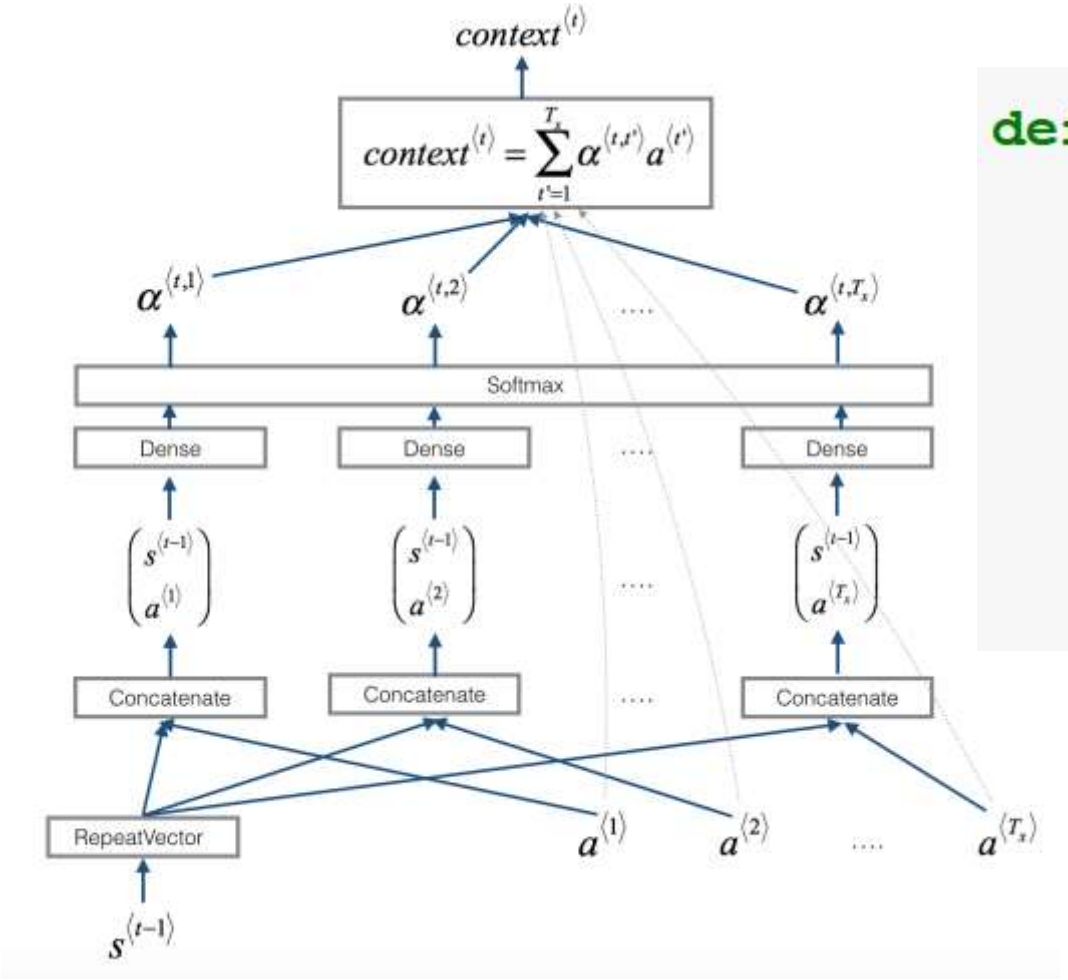

```
repeator = RepeatVector(Tx)
concatenator = Concatenate(axis=-1)
densor1 = Dense(10, activation = "tanh")
densor2 = Dense(1, activation = "relu")
activator = Activation(softmax, name='attention_weights')
dotor = Dot(axes = 1)
```

محاسبه توجه (Attention) - $\alpha^{<t,t'>}$

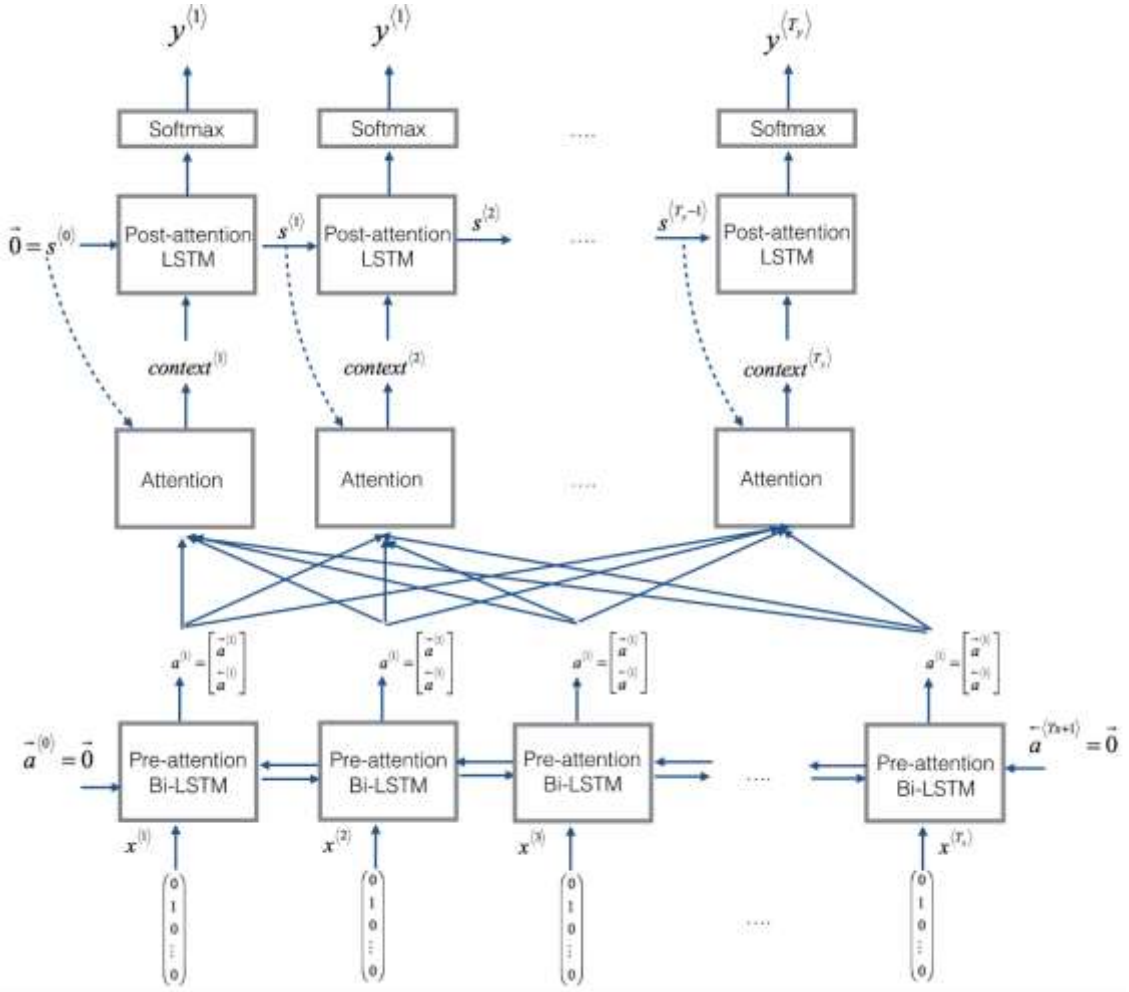



```
repeator = RepeatVector(Tx)
concatenator = Concatenate(axis=-1)
densor1 = Dense(10, activation = "tanh")
densor2 = Dense(1, activation = "relu")
activator = Activation(softmax, name='attention_weights')
dotor = Dot(axes = 1)
```

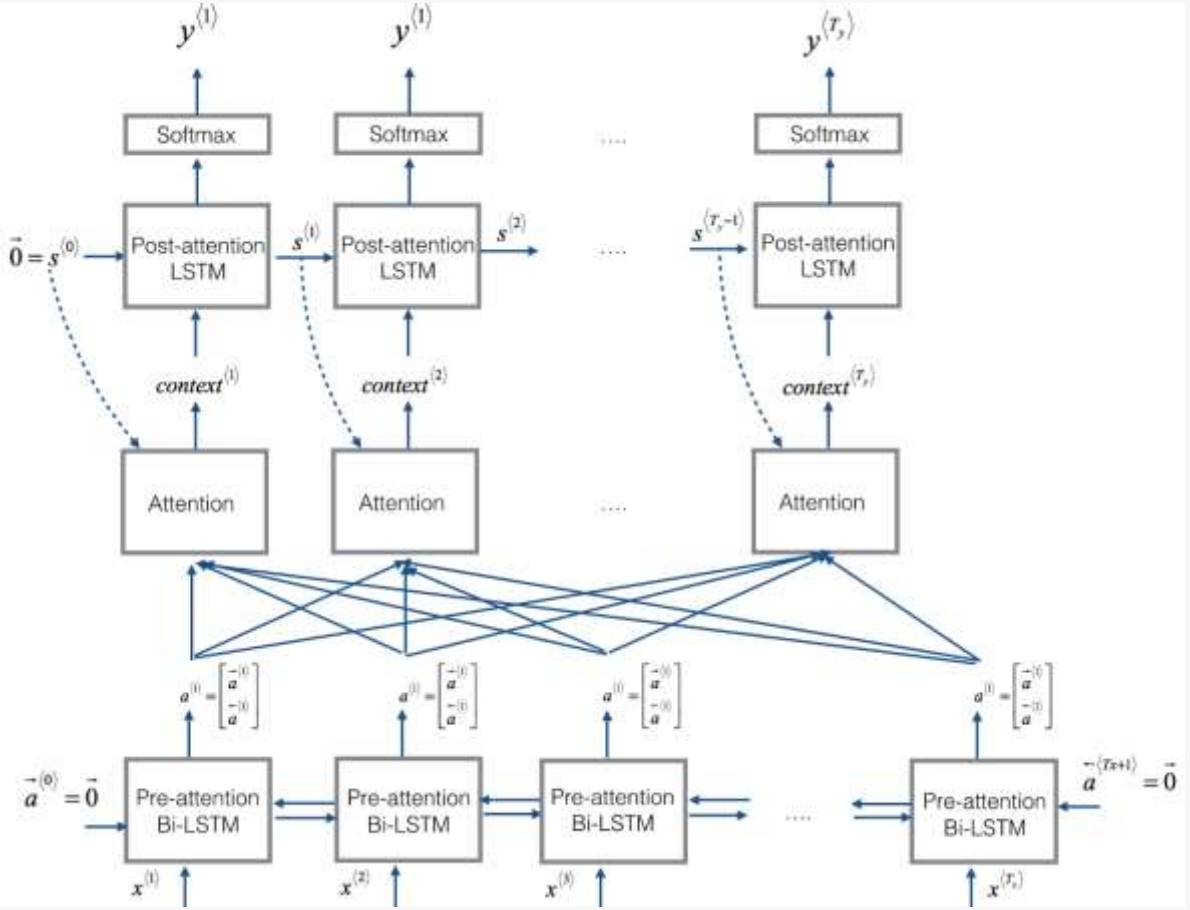
محاسبه توجه (Attention) - $\alpha^{<t,t'>}$



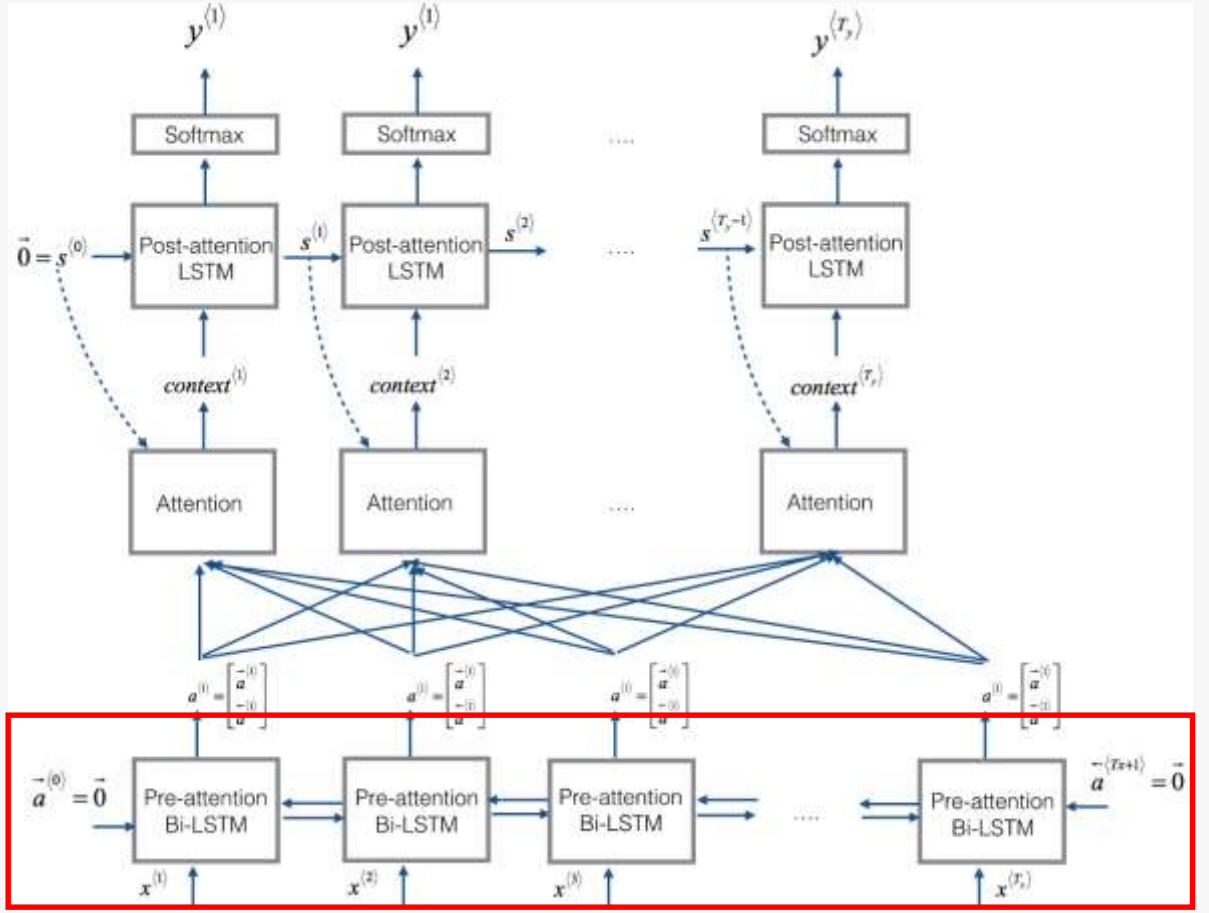

```python
def one_step_attention(a, s_prev):
    s_prev = repeator(s_prev)
    concat = concatenator([a, s_prev])
    e = densor1(concat)
    energies = densor2(e)
    alphas = activator(energies)
    context = dotor([alphas, a])
    return context
```



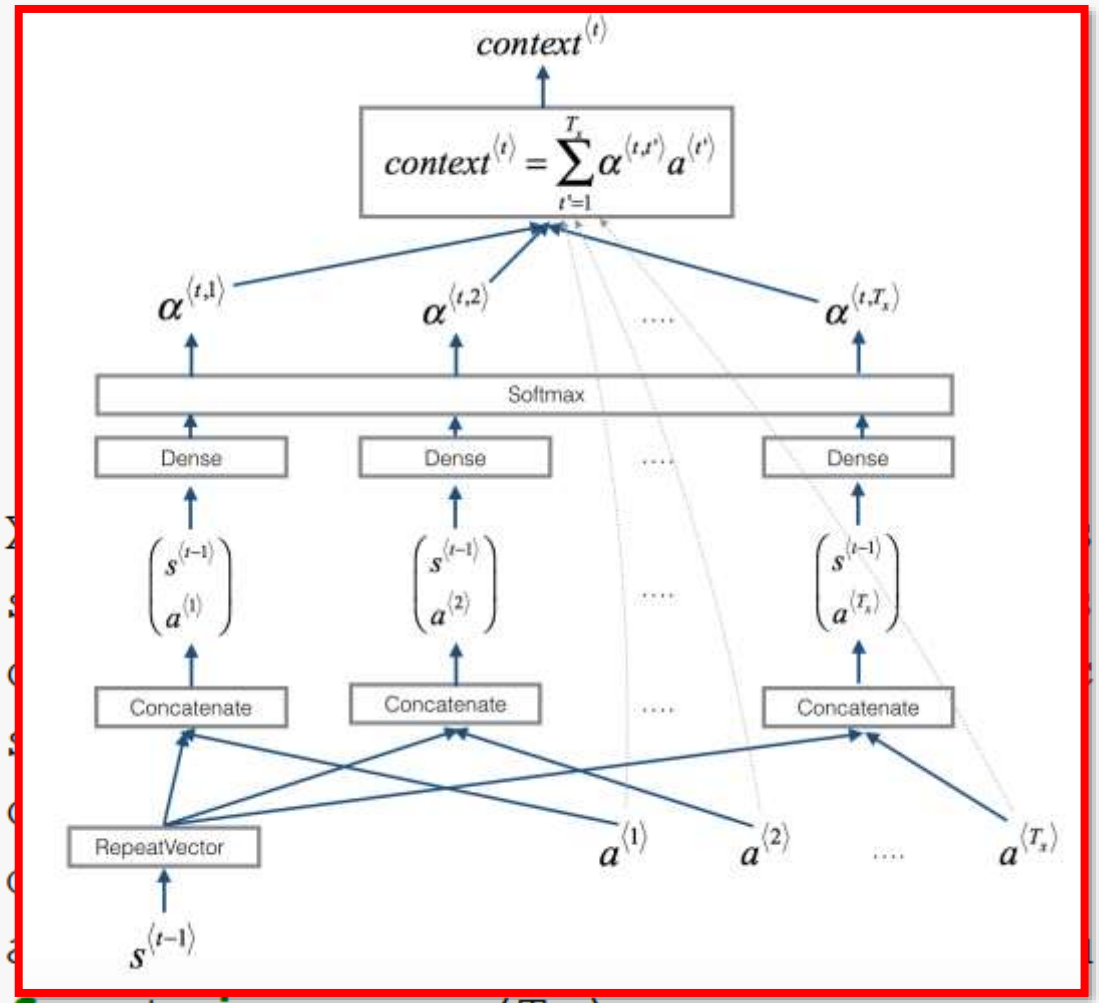

```
post_activation_LSTM_cell = LSTM(n_s, return_state = True)
output_layer = Dense(len(dest_vocab), activation=softmax)
```

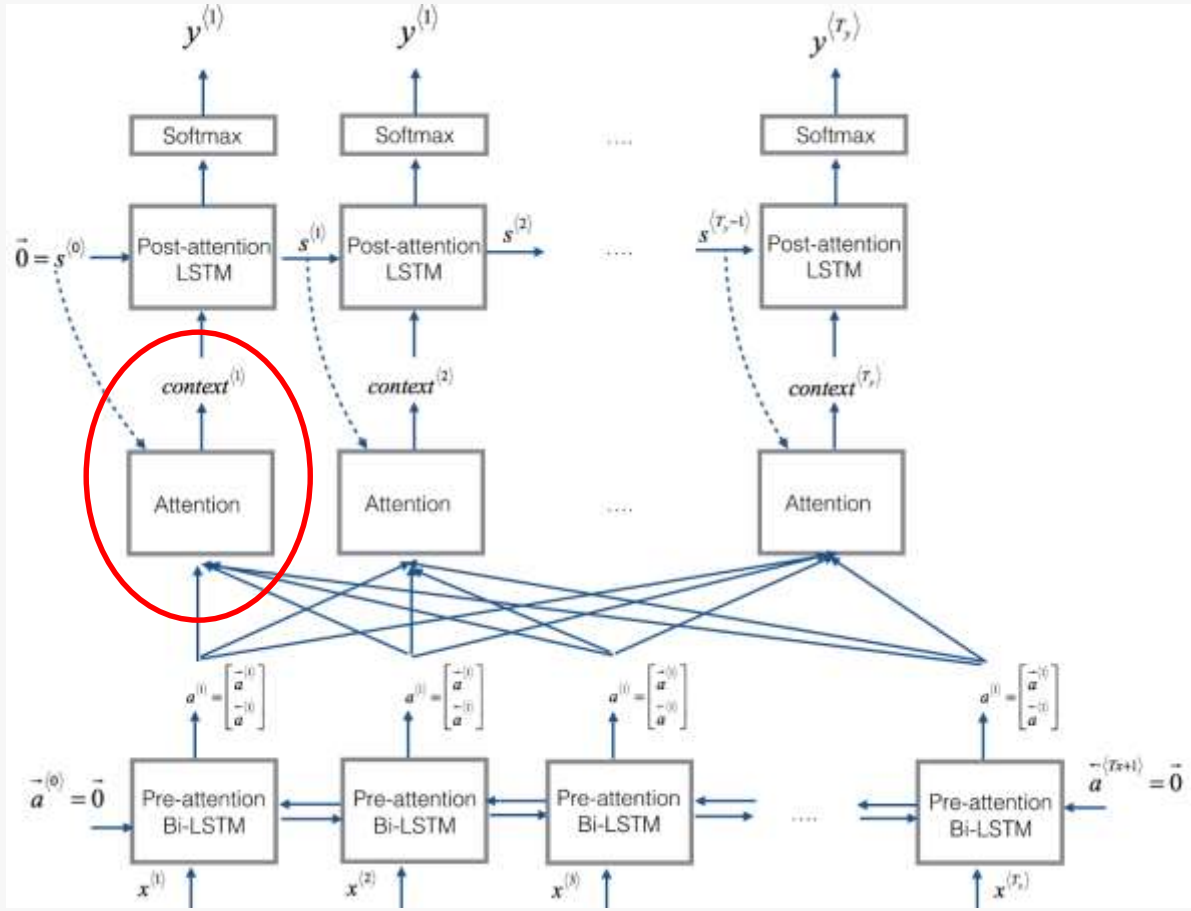



```
X = Input(shape=(Tx, src_vocab_size))
s0 = Input(shape=(n_s,), name='s0')
c0 = Input(shape=(n_s,), name='c0')
s = s0
c = c0
outputs = []
a = Bidirectional(LSTM(n_a, return_sequences=True))(X)
for t in range(Ty):
    context = one_step_attention(a, s)
    s, _, c = post_activation_LSTM_cell(context, initial_state = [s, c])
    out = Dense(len(dest_vocab), activation=softmax)(s)
    outputs.append(out)
model = Model(inputs=[X, s0, c0], outputs=outputs)
```

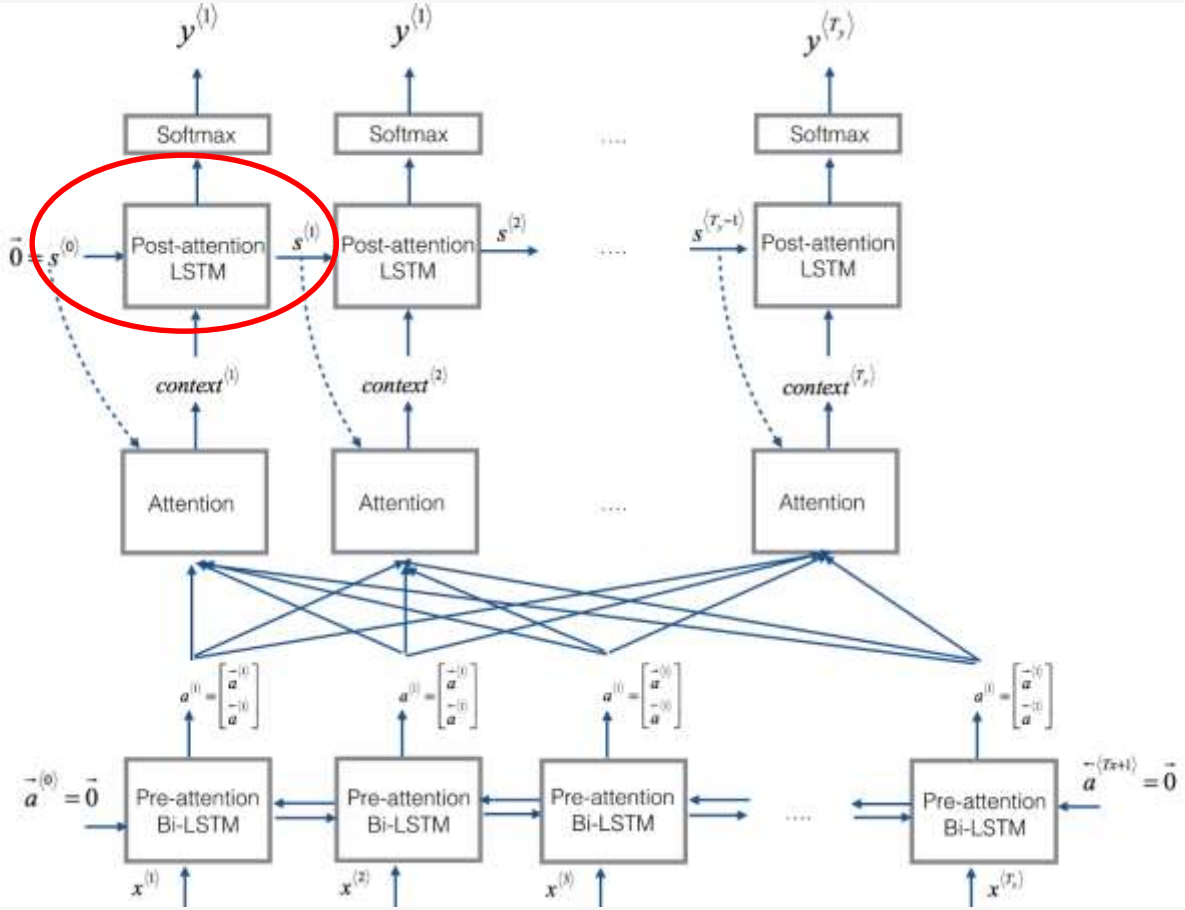



```
X = Input(shape=(Tx, src_vocab_size))
s0 = Input(shape=(n_s,), name='s0')
c0 = Input(shape=(n_s,), name='c0')
s = s0
c = c0
outputs = []
a = Bidirectional(LSTM(n_a, return_sequences=True))(X)
for t in range(Ty):
    context = one_step_attention(a, s)
    s, _, c = post_activation_LSTM_cell(context, initial_state = [s, c])
    out = Dense(len(dest_vocab), activation=softmax)(s)
    outputs.append(out)
model = Model(inputs=[X, s0, c0], outputs=outputs)
```

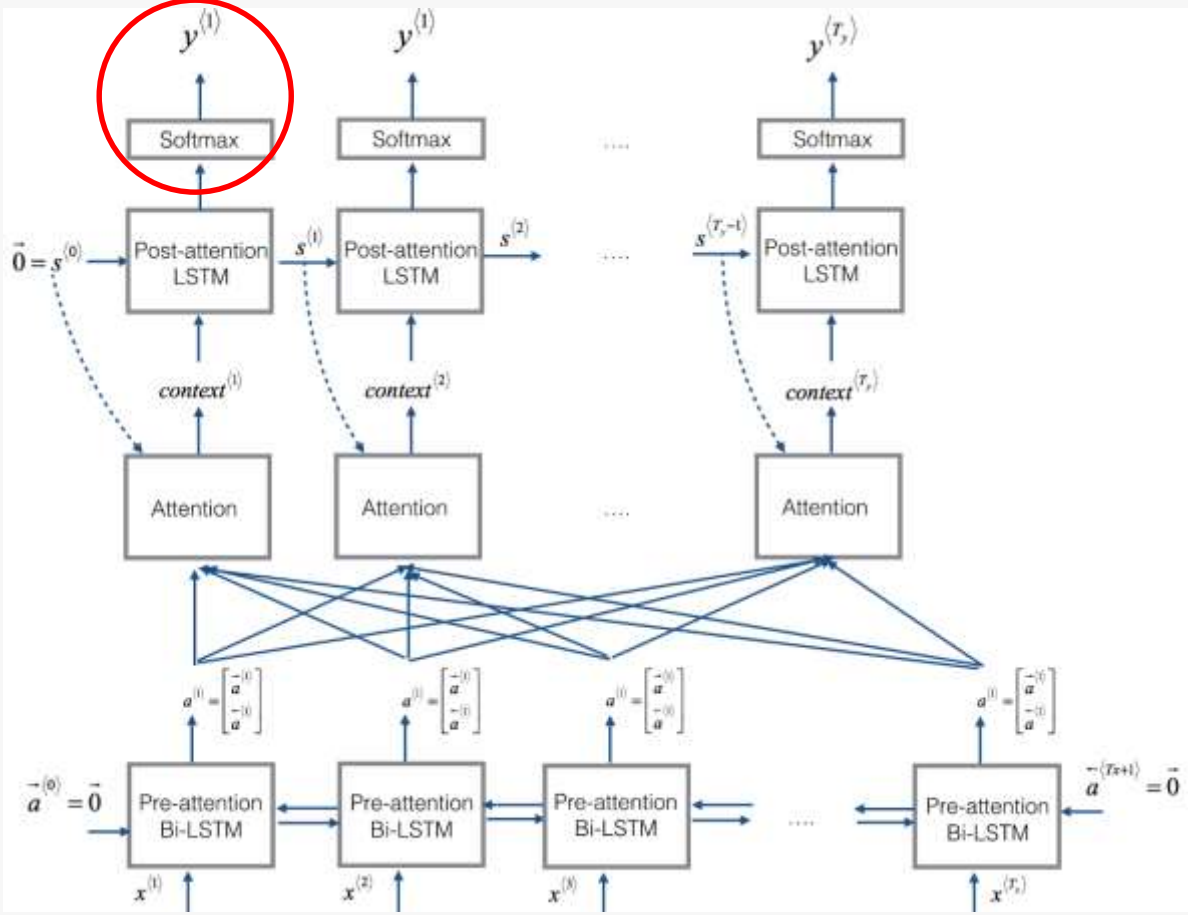





```
ize))
0')
0')
rn_sequences=True))(X)
for t in range(Ty):
    context = one_step_attention(a, s)
    s, _, c = post_activation_LSTM_cell(context, initial_state = [s, c])
    out = Dense(len(dest_vocab), activation=softmax)(s)
    outputs.append(out)
model = Model(inputs=[X, s0, c0], outputs=outputs)
```




```
X = Input(shape=(Tx, src_vocab_size))
s0 = Input(shape=(n_s,), name='s0')
c0 = Input(shape=(n_s,), name='c0')
s = s0
c = c0
outputs = []
a = Bidirectional(LSTM(n_a, return_sequences=True))(X)
for t in range(Ty):
    context = one_step_attention(a, s)
    s, _, c = post_activation_LSTM_cell(context, initial_state = [s, c])
    out = Dense(len(dest_vocab), activation=softmax)(s)
    outputs.append(out)
model = Model(inputs=[X, s0, c0], outputs=outputs)
```



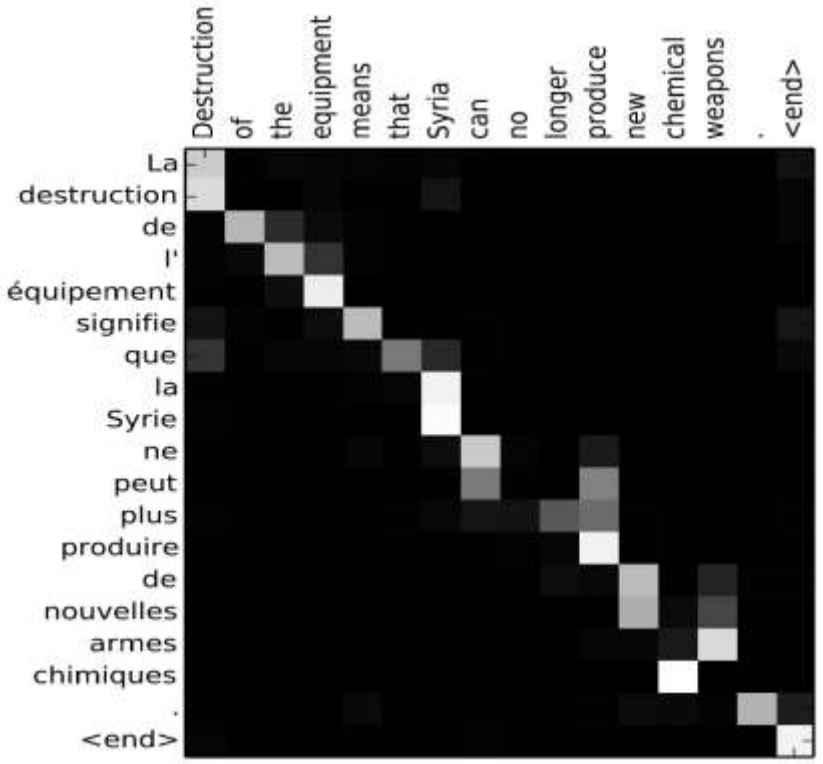

```
X = Input(shape=(Tx, src_vocab_size))
s0 = Input(shape=(n_s,), name='s0')
c0 = Input(shape=(n_s,), name='c0')
s = s0
c = c0
outputs = []
a = Bidirectional(LSTM(n_a, return_sequences=True))(X)
for t in range(Ty):
    context = one_step_attention(a, s)
    s, _, c = post_activation_LSTM_cell(context, initial_state = [s, c])
    out = Dense(len(dest_vocab), activation=softmax)(s)
    outputs.append(out)
model = Model(inputs=[X, s0, c0], outputs=outputs)
```

توجه (Attention)

July 20th 1969 $\longrightarrow$ 1969 – 07 – 20

23 April, 1564 $\longrightarrow$ 1564 – 04 – 23

Visualization of $\alpha^{<t,t'>}$:



ترجمه ماشینی – تبدیل تاریخ

09_add_numbers_with_seq2seq

What is Teacher Forcing?

[Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neuralnetworks.Neural computation,1(2), 270–280]

What is Teacher Forcing?

تاحالا در مدرسه با سوالات قسمت الف، ب، ج، د که پاسخ هر قسمت برای قسمت دوم نیاز است برخورد داشته اید؟

دسته ای از سوالات بود که پاسخ قسمت الف برای محاسبه قسمت ب و پاسخ قسمت ب برای محاسبه ج و ... ضروری بود! ☹

اگر نمره ی قسمت الف را نمی‌گرفتیم چه می‌شد؟!

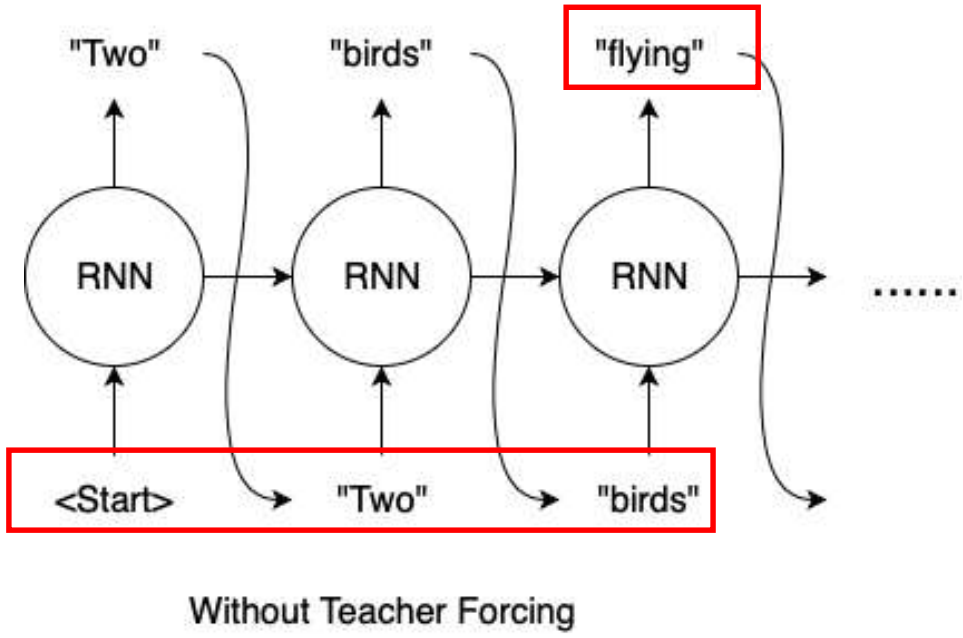
بهتر نبود معلم بعد از هر قسمت نمره ی ما را برای آن قسمت ثبت می‌کرد و بعد از گفتن جواب صحیح بهمون فرصت می‌داد قسمت بعدی را حل کنیم؟! ☺

روش Teacher Forcing برای آموزش RNNها

خروجی مورد نتظار: Two people reading a book

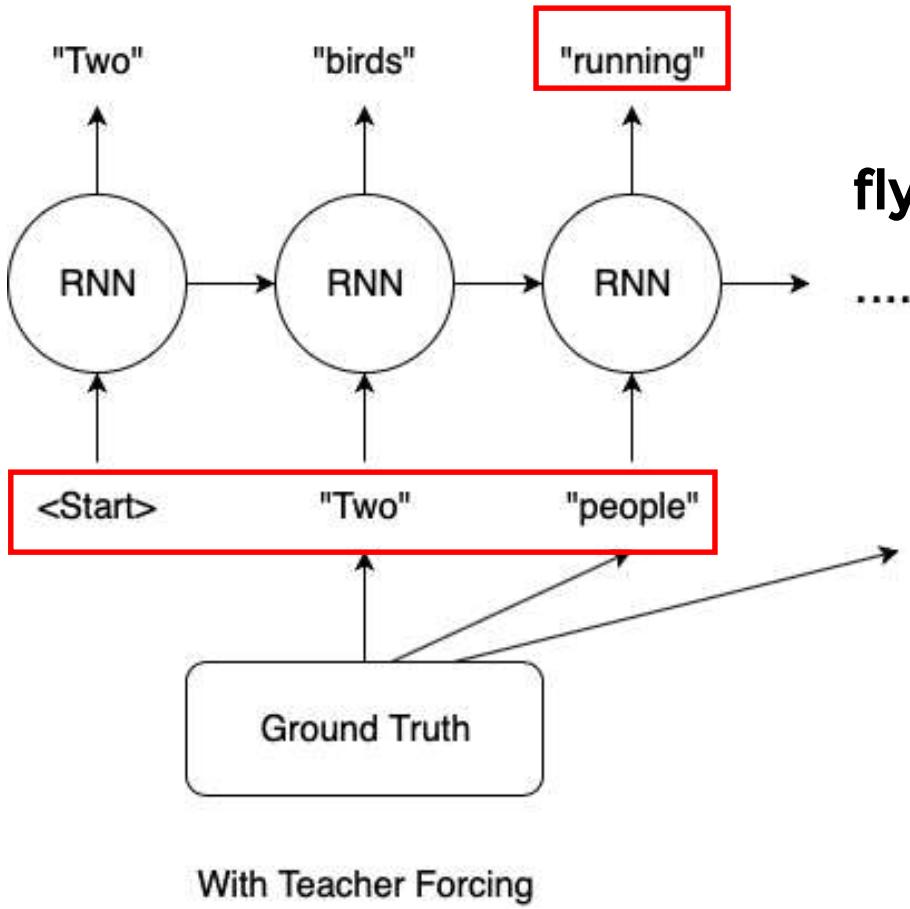
خروجی مدل برای ۲ گام زمانی نخست: Two birds

خروجی برای گام زمانی سوم چه خواهد بود؟! Reading یا flying



Without Teacher Forcing

روش Teacher Forcing برای آموزش RNNها



With Teacher Forcing

خروجی مورد نتظار: **Two people reading a book**

خروجی مدل برای ۲ گام زمانی نخست: **Two birds**

خروجی برای گام زمانی سوم چه خواهد بود؟! **Reading** یا **flying**

.....

روش Teacher Forcing برای آموزش RNNها

مزایا

1. در ابتدای آموزش همگرایی سریع‌تر خواهد بود.

Training with *Teacher Forcing* converges faster. At the early stages of training, the predictions of the model are very bad. **If we do not use *Teacher Forcing*, the hidden states of the model will be updated by a sequence of wrong predictions, errors will accumulate**, and it is difficult for the model to learn from that.

روش Teacher Forcing برای آموزش RNNها

معایب

1. تفاوت در آموزش و inference
2. کاهش دقت در زمان inference

During inference, since there is usually no ground truth available, the RNN model will need to feed its own previous prediction back to itself for the next prediction. Therefore there is a discrepancy between training and inference, and **this might lead to poor model performance and instability**. This is known as Exposure Bias in literature.

روش Teacher Forcing برای آموزش RNNها

راه حل؟!

Unfortunately, this procedure can result in problems in generation as small prediction error compound in the conditioning context. This can lead to poor prediction performance as the RNN's conditioning context (the sequence of previously generated samples) diverge from sequences seen during training.

Professor Forcing: A New Algorithm for Training Recurrent Networks

مطالعه بیشتر و سورس‌کدهای ترجمه ماشینی

- https://www.tensorflow.org/tutorials/text/nmt_with_attention
- http://opennmt.net/
- https://github.com/OpenNMT/OpenNMT-tf

توجه (Attention) در Image Captioning

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

Show, Attend and Tell: Neural Image Caption Generation with Visual Attention [2016]

منابع

- https://www.coursera.org/specializations/deep-learning
- https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9
- https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21
- https://www.datatechnotes.com/2018/12/rnn-example-with-keras-simplernn-in.html
- https://www.youtube.com/watch?v=QciIcRxJvsM
- Long Short-Term Memory Networks With Python , Develop Sequence Prediction Models With Deep Learning. By Jason Brownlee
- *https://towardsdatascience.com/sequence-to-sequence-model-introduction-and-concepts-44d9b41cd42d*
- *https://towardsdatascience.com/what-is-teacher-forcing-3da6217fed1c*