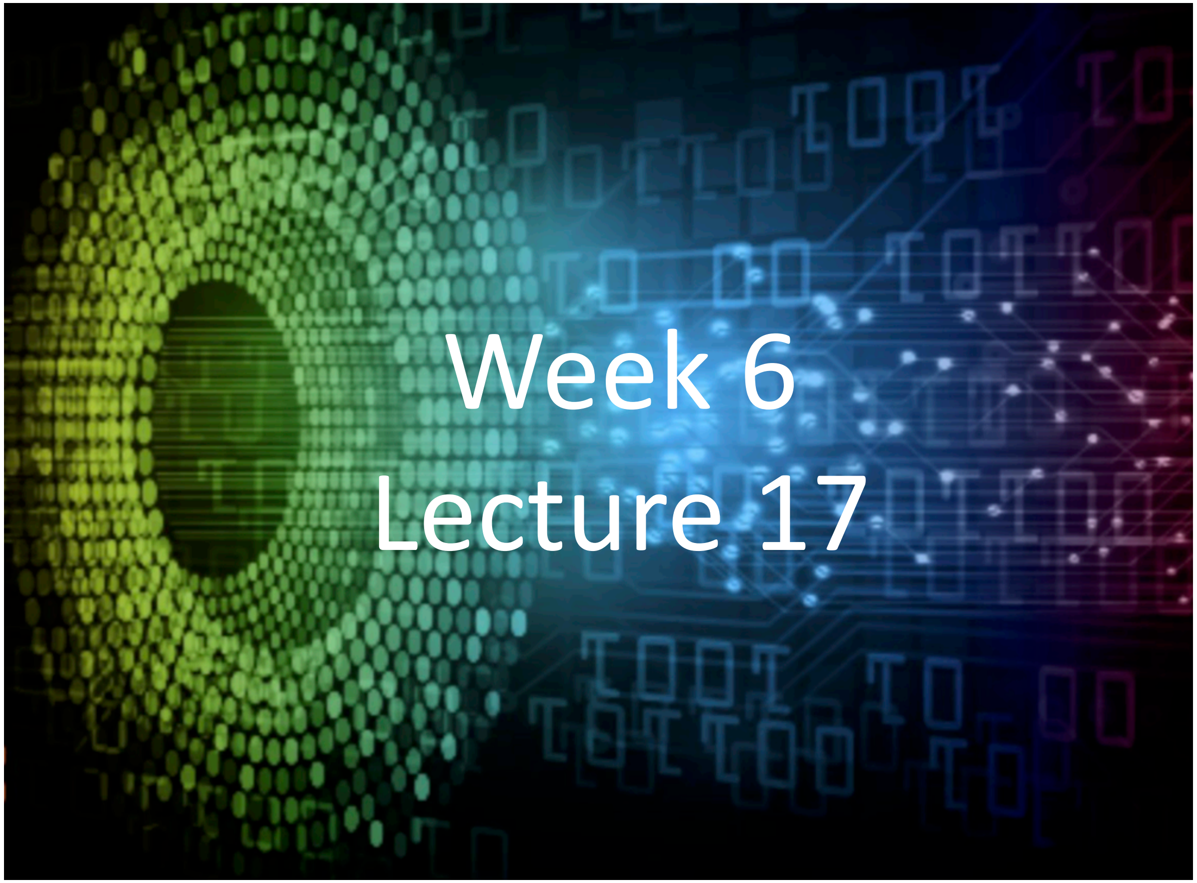
Introduction to Deep Learning Applications and Theory

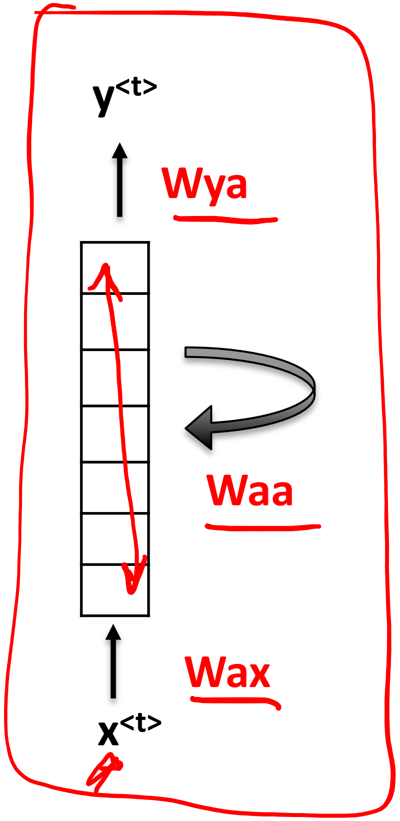


Week 6
Lecture 17

ECE 596 / AMATH 563

Previous Lecture: Recurrent Neural Networks (RNNs)

- Input-Output
- Definition
- Notation
- Backward Propagation
- Diminishing/ Exploding Gradients



RNN

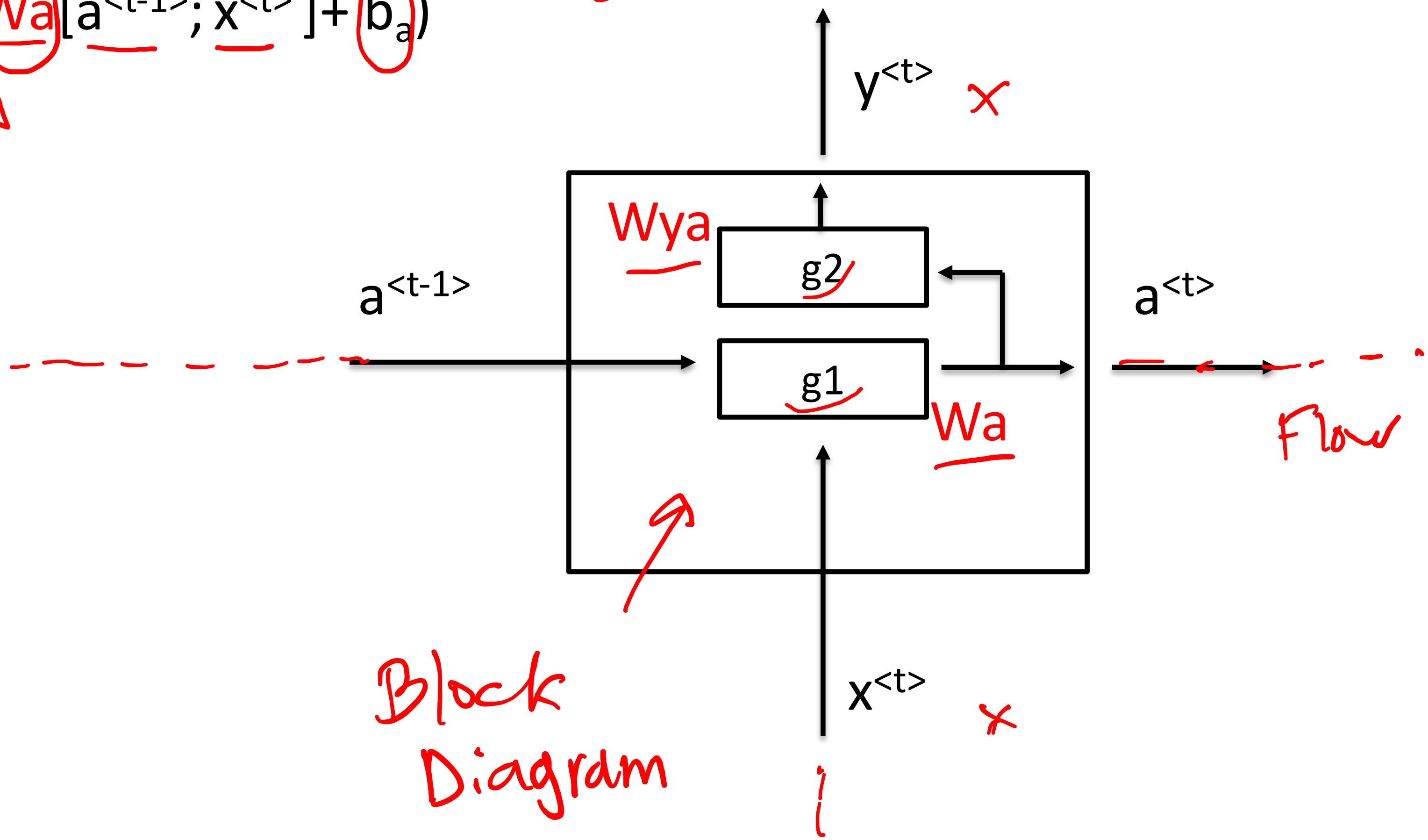
$$y^{<t>} = g2(Wya\, a^{<t>} + b_y)$$

$$a^{<t>} = g1(Wa[a^{<t-1>}; x^{<t>}] + b_a)$$

Classification

g2 ~ tanh, ···, G

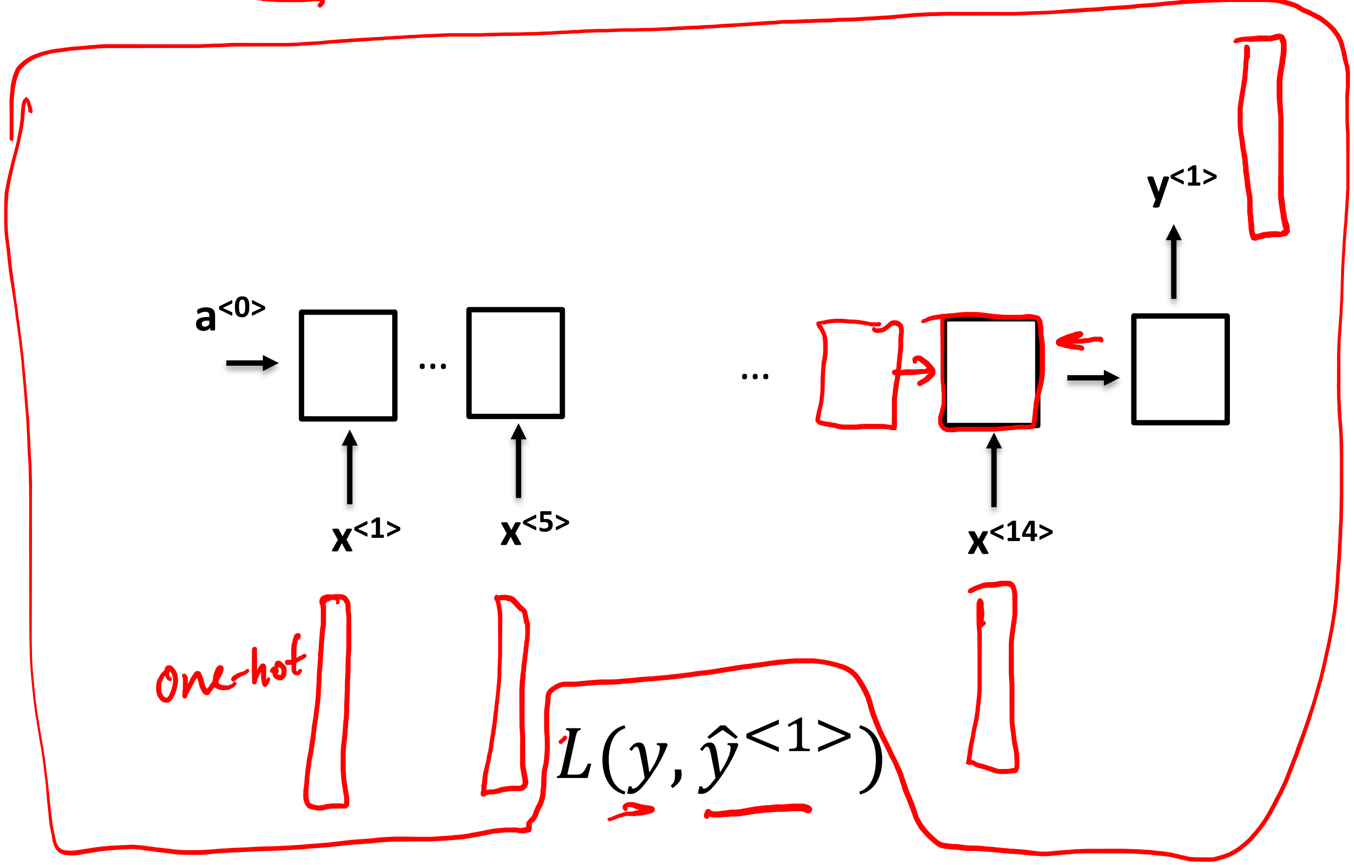
g1 ~ tanh



Block Diagram

Vanishing/Exploding Gradients

"I grew up in France and moved to the United States, therefore I speak _____"



$L(y, \hat{y}^{<1>})$

Vanishing Gradients

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

✓ ① >1 blow up

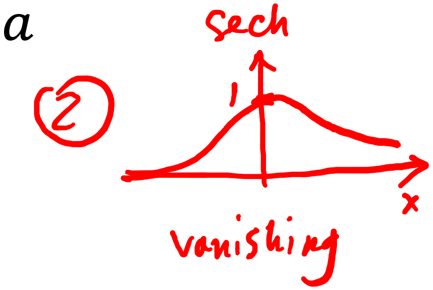
② <1 vanish

$$\underline{\frac{\partial L}{\partial W_a}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \underline{a^{<Tx>}}} \cdot \boxed{\underline{\left(\prod_{t=2}^{Tx} \frac{\partial a^{<t>}}{\partial a^{<t-1>}}\right)}} \cdot \frac{\partial a^{<1>}}{\partial W_a}$$

$\overrightarrow{\text{B.P.T}}$

$$\frac{\partial a^{<t>}}{\partial a^{<t-1>}} = \underset{\uparrow}{\underline{\text{sech}^2}}(\underset{\uparrow}{W_a} \overset{\downarrow\ ②}{\underline{a^{<t-1>}}} + W_x \overset{\downarrow}{\underline{x^{<t>}}}) \cdot \overset{①}{\underset{\uparrow}{\underline{W_a}}}$$

$W_a\,[a^{<t-1>}; x^{<t>}]$



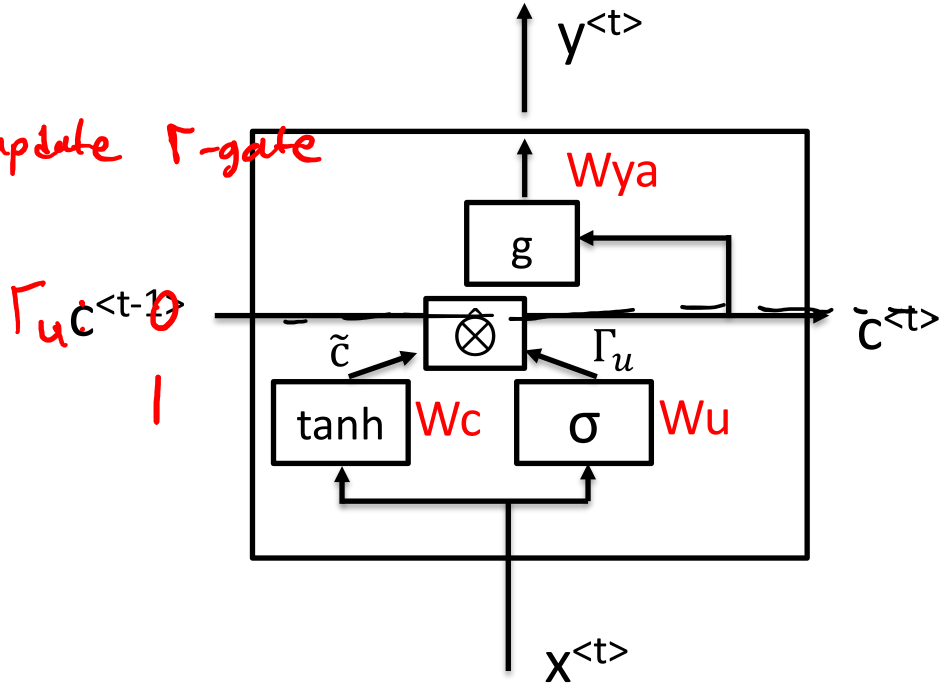
Gated Recurrent Unit (GRU)

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

$$\Gamma_u = \sigma(Wu[c^{<t-1>}; x^{<t>}] + b_u)$$ u-update Γ-gate

$$c^{<t>} = \underbrace{\Gamma_u \tilde{c}^{<t>}}_{1} + \underbrace{(1 - \Gamma_u)}_{0}c^{<t-1>}$$

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



Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014).
On the properties of neural machine translation: Encoder-decoder approaches.

Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. (2014).
Empirical evaluation of gated recurrent neural networks on sequence modeling.

Vanishing Gradients

$$\frac{\partial L}{\partial W_a} = \frac{\partial L}{\partial \hat{y}} \; \frac{\partial \hat{y}}{\partial c^{<Tx>}} \cdot \left(\underline{\prod_{t=2}^{Tx} \frac{\partial \boldsymbol{c}^{<t>}}{\partial \boldsymbol{c}^{<t-1>}}} \right) \cdot \frac{\partial c^{<1>}}{\partial W_c} \qquad a = c$$

$$\begin{aligned}
\frac{\partial \boldsymbol{c}^{<t>}}{\partial \boldsymbol{c}^{<t-1>}} = {} & \Gamma_u{}' \tanh\left(W_c \boldsymbol{c}^{<t-1>} + W_x \boldsymbol{x}^{<t>}\right) && \leftarrow \textcircled{1} \quad \Gamma' \\
& \Gamma_u \underline{\operatorname{sech}^2\left(W_c \boldsymbol{c}^{<t-1>} + W_x \boldsymbol{x}^{<t>}\right) \cdot W_c} + && \underline{\leftarrow \textcircled{2}} \\
& \Gamma_u{}' \, \boldsymbol{c}^{<t-1>} + && \leftarrow \quad \textcircled{3} \leftarrow \Gamma' \\
& \cancel{(1-\Gamma_u)} \text{ no} && \leftarrow \quad \textcircled{4}
\end{aligned}$$

Full $\underline{\text{GRU}}$

$$\tilde{c}^{<t>} = \tanh(Wc[\Gamma_r \underline{c^{<t-1>}}; x^{<t>}] + b_c)$$

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

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

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

Long Short Term Memory

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

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

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

$$\underline{\Gamma_o} = \sigma(Wo[\underline{a^{<t-1>}}; x^{<t>}] + b_0)$$

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

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

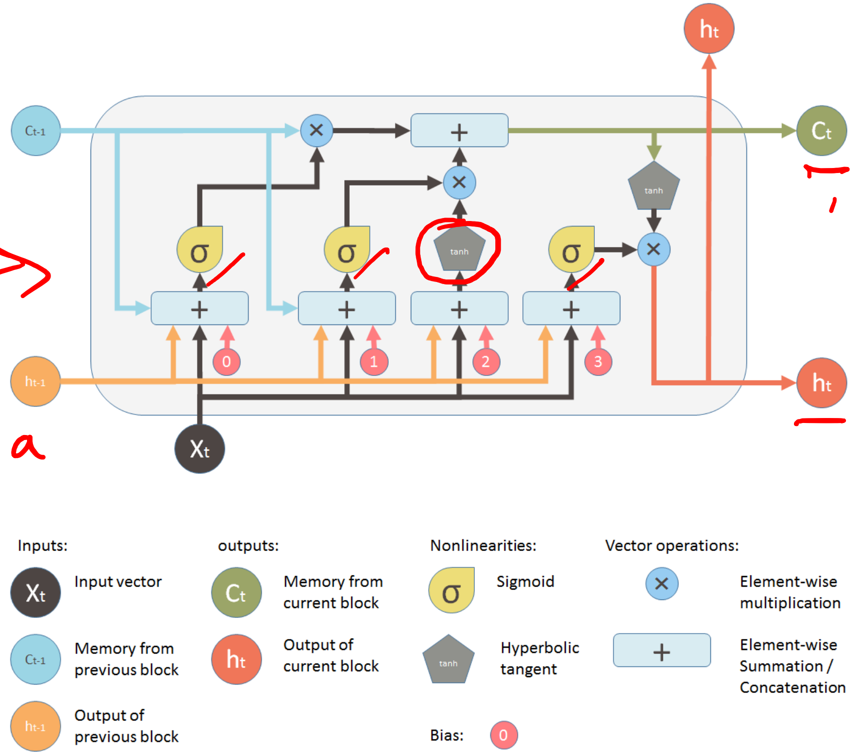
Sepp
Hochreiter

Jurgen
Schmidhuber

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

LSTM

$$\left\{\begin{aligned}
\tilde{c}^{<t>} &= \tanh(Wc[a^{<t-1>}; x^{<t>}] + b_c) \\
\Gamma_u &= \sigma(Wu[a^{<t-1>}; x^{<t>}] + b_u) \\
\Gamma_f &= \sigma(Wf[a^{<t-1>}; x^{<t>}] + b_f) \\
\Gamma_o &= \sigma(Wo[a^{<t-1>}; x^{<t>}] + b_0) \\
c^{<t>} &= \Gamma_u \tilde{c}^{<t>} + \Gamma_f c^{<t-1>} \\
a^{<t>} &= \Gamma_o \tanh c^{<t>}
\end{aligned}\right.$$

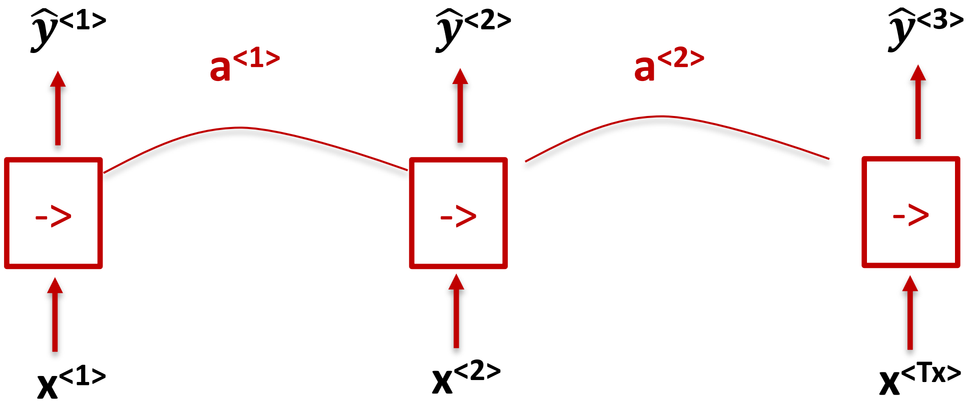


ht
Ct-1
Ct
tanh
σ
σ
tanh
σ
0
1
2
3
ht-1
ht
a
Xt
Inputs:
Xt Input vector
Ct-1 Memory from previous block
ht-1 Output of previous block
outputs:
Ct Memory from current block
ht Output of current block
Nonlinearities:
σ Sigmoid
tanh Hyperbolic tangent
Vector operations:
× Element-wise multiplication
+ Element-wise Summation / Concatenation
Bias: 0

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

Bi-Directional RNN

Forward



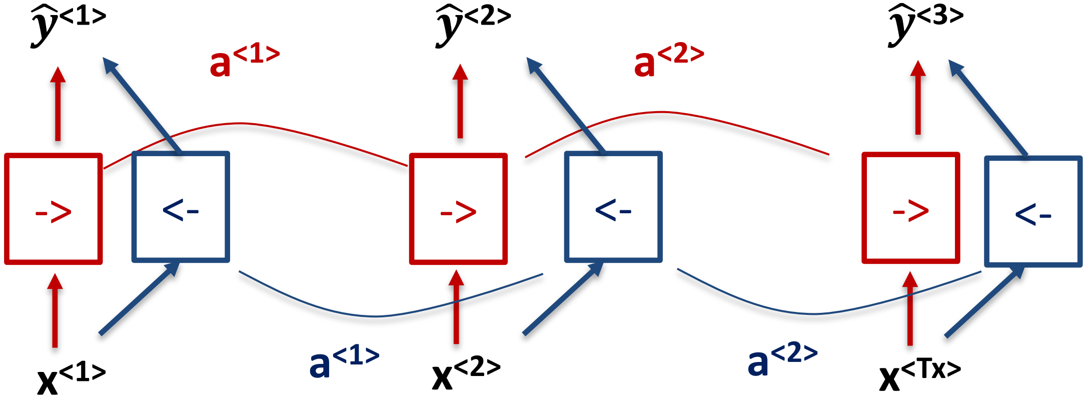
He said, Tesla is a unit of magnetic field strength

He said, Tesla is an electric automotive and sustainable energy company

Bi-Directional RNN

Forward

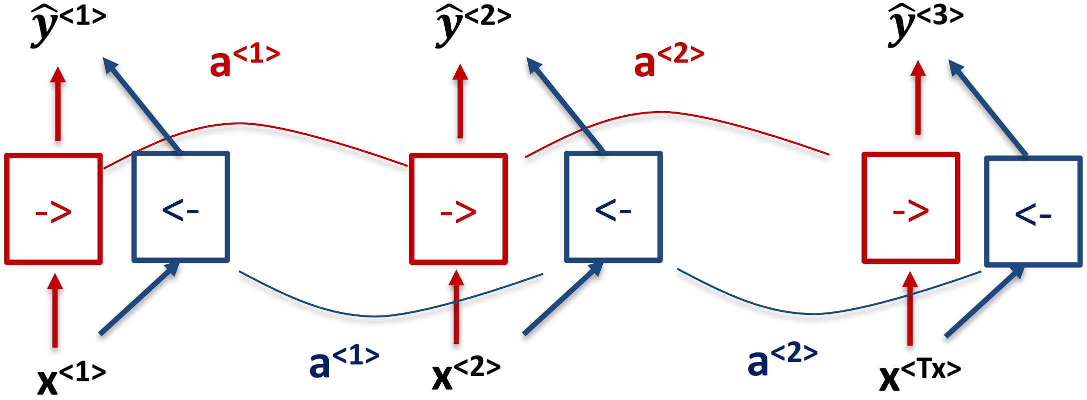
Backward



Bi-Directional RNN

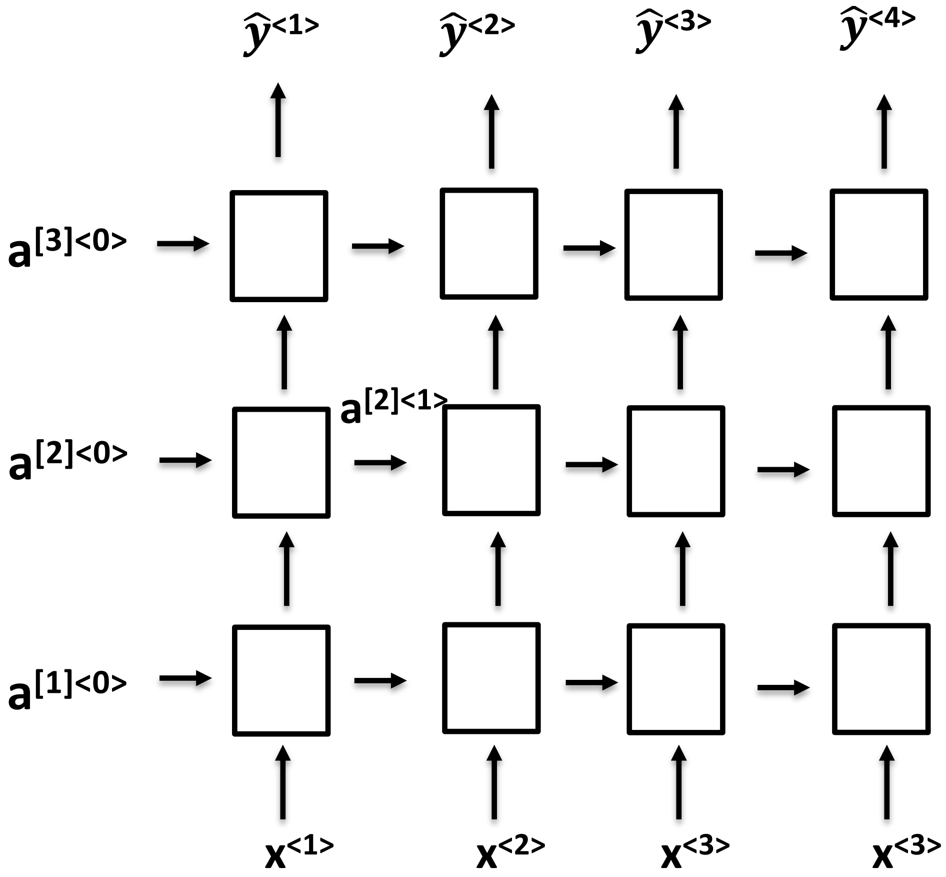
Forward

Backward

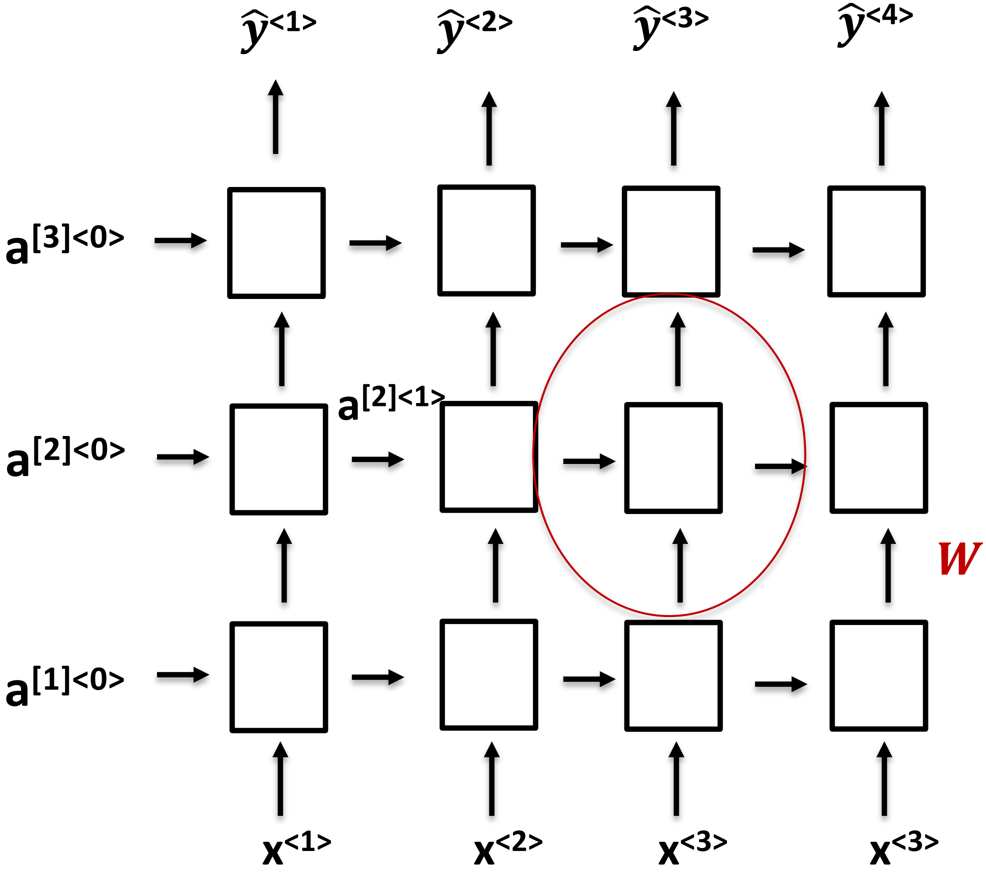


$$\hat{y}^{<t>} = g(Wy\,[a^{F<t>}; a^{B<t>}] + by)$$

Deep RNN



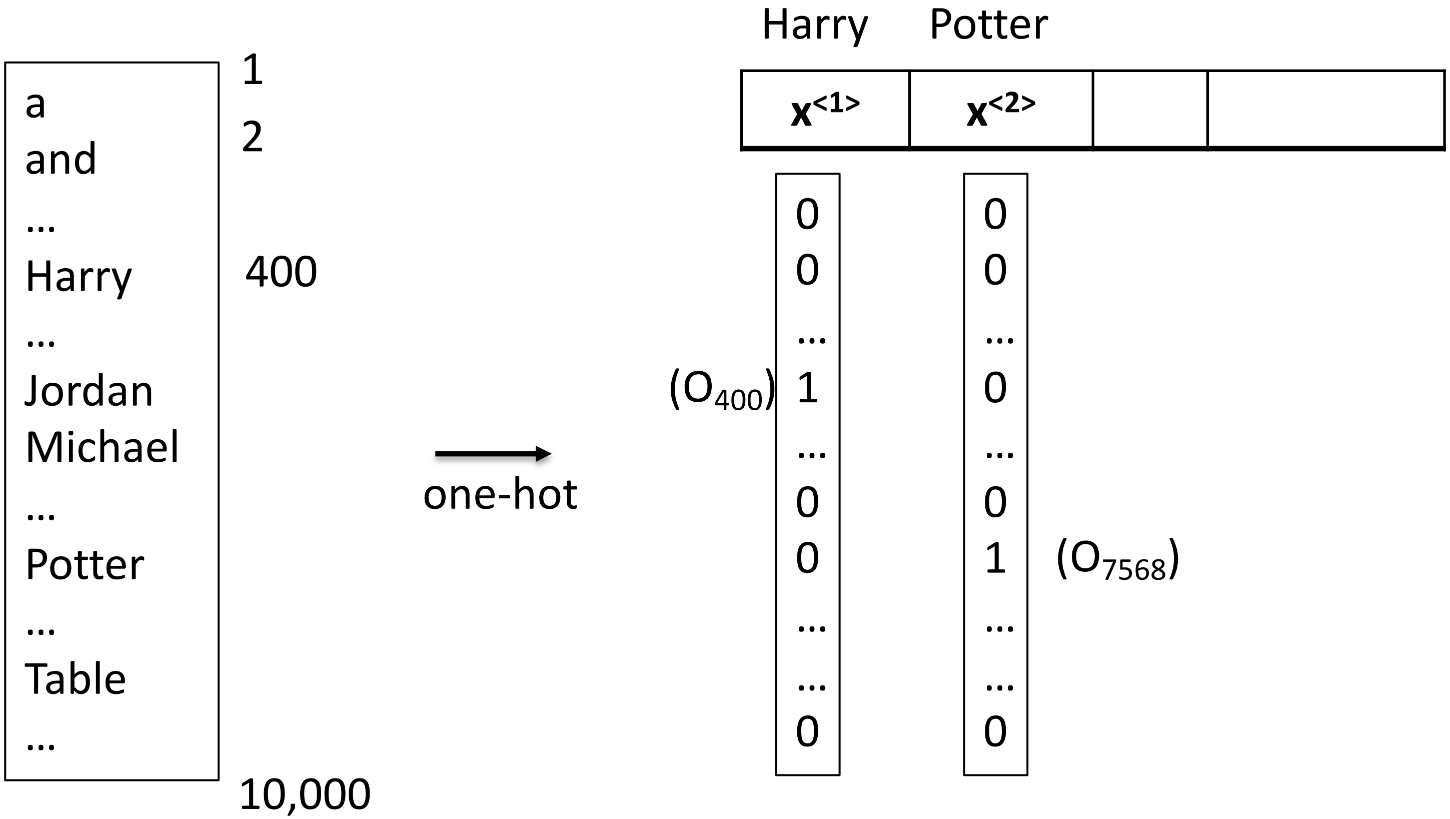
Deep RNN



$$a^{[2]<3>} = g($$
$$Wa^{[2]}\left[a^{[2]<2>}; a^{[1]<3>}\right] + b_a{}^{[2]})$$

Natural Language Processing

Vocabulary/Dictionary



a 1
and 2
…
Harry 400
…
Jordan
Michael
…
Potter
…
Table
…
10,000

one-hot

Harry Potter

$x^{<1>}$ $x^{<2>}$

0
0
…
(O_{400}) 1
…
0
0
…
…
0

0
0
…
0
…
0
1 (O_{7568})
…
…
0

Word Embeddings

I would really like to drive a Tesla ____ .

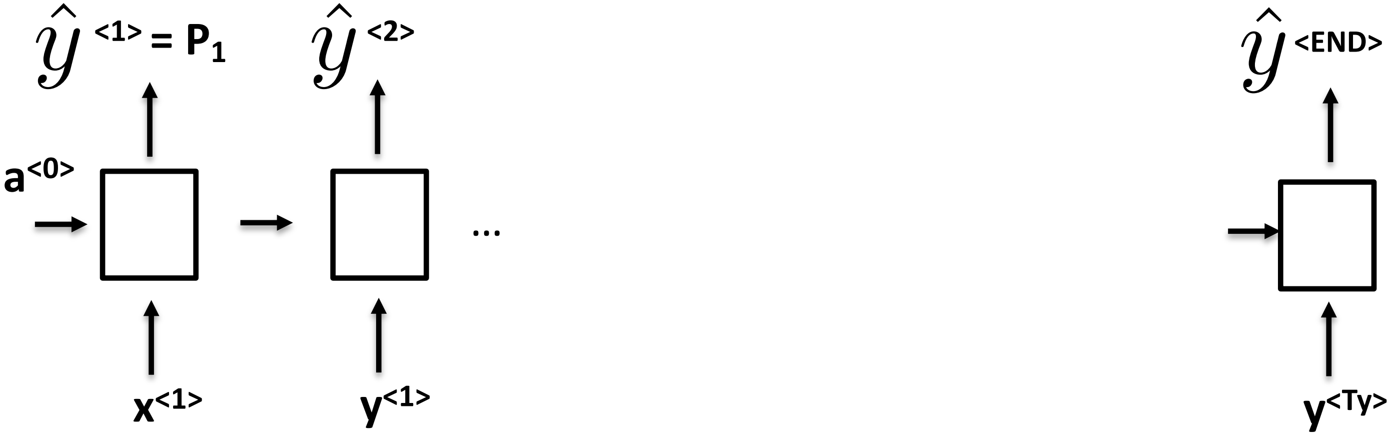
I would really like to drive a Porsche ____.

Embedding vector

	Man	Woman	King	Queen	Orange	Apple	Tesla	Porsche
Gender	-1	1	-0.95	0.97	0.01	0	-0.5	0.03
Royal	0.01	0.02	0.94	0.95	-0.02	0.04	-0.01	0.1
Food								
Size								
Engine	0.07	-0.01	0.03	0.02	0.001	0	0.95	0.98

Visualization 300 -> 2D: t-SNE

Named Entity Recognition Example



Sally Johnson is a Porsche driver.

Magic Jordan is a Tesla motorist.

Transfer Learning

- Learn word embeddings from large text corpus (1-100B words). Or download.

- Transfer to new task with smaller training set (~100k).

- Continue to tune the embeddings with the new data.

Properties of Embedding Vectors

	Man	Woman	King	Queen	Orange	Apple	Tesla	Porsche
Gender	-1	1	-0.95	0.97	0.01	0	-0.5	0.03
Royal	0.01	0.02	0.94	0.95	-0.02	0.04	-0.01	0.1
Food								
Size								
Engine	0.07	-0.01	0.03	0.02	0.001	0	0.95	0.98

Can define distances (similarity):

$e_{Man} - e_{Woman} \sim= [-2;0;0;..;0]$

$e_{King} - e_{Queen} \sim= [-2;0;0;..;0]$

$e_{Man} - e_{Woman} \sim= e_{King} - e_{?}$

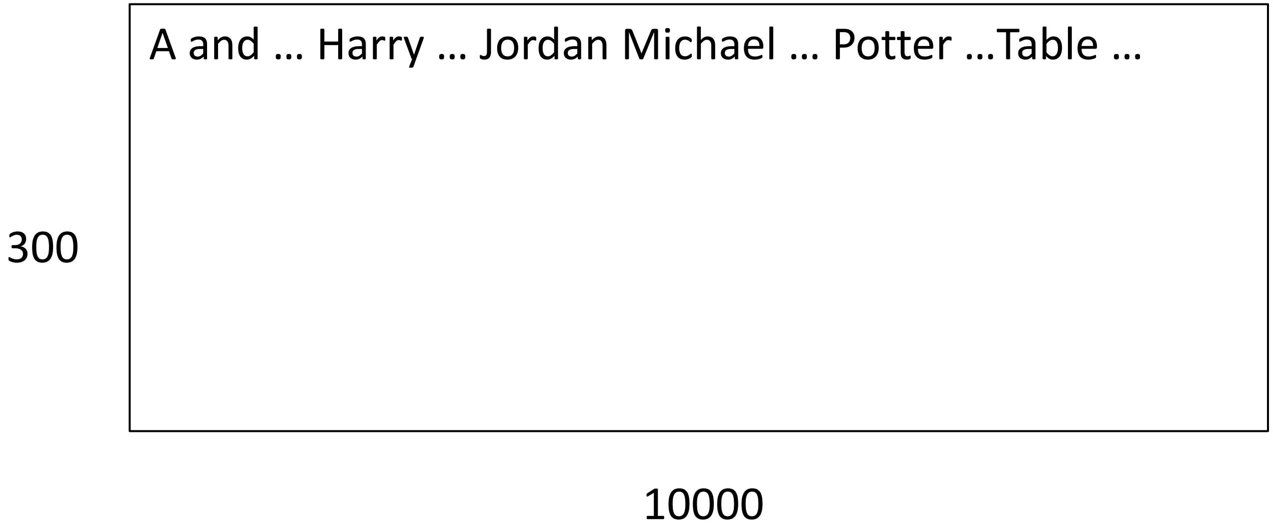
Similarity

We can define similarity in the space of embedding vectors (Full space)

$$\text{argmax}_w \ \text{sim}(e_w, v)$$

Embedding Matrix

E=



A and … Harry … Jordan Michael … Potter …Table …

300

10000

$$\overrightarrow{\boldsymbol{e_j}} = \boldsymbol{E} \cdot \overrightarrow{\boldsymbol{o_j}}$$