Neural language modeling

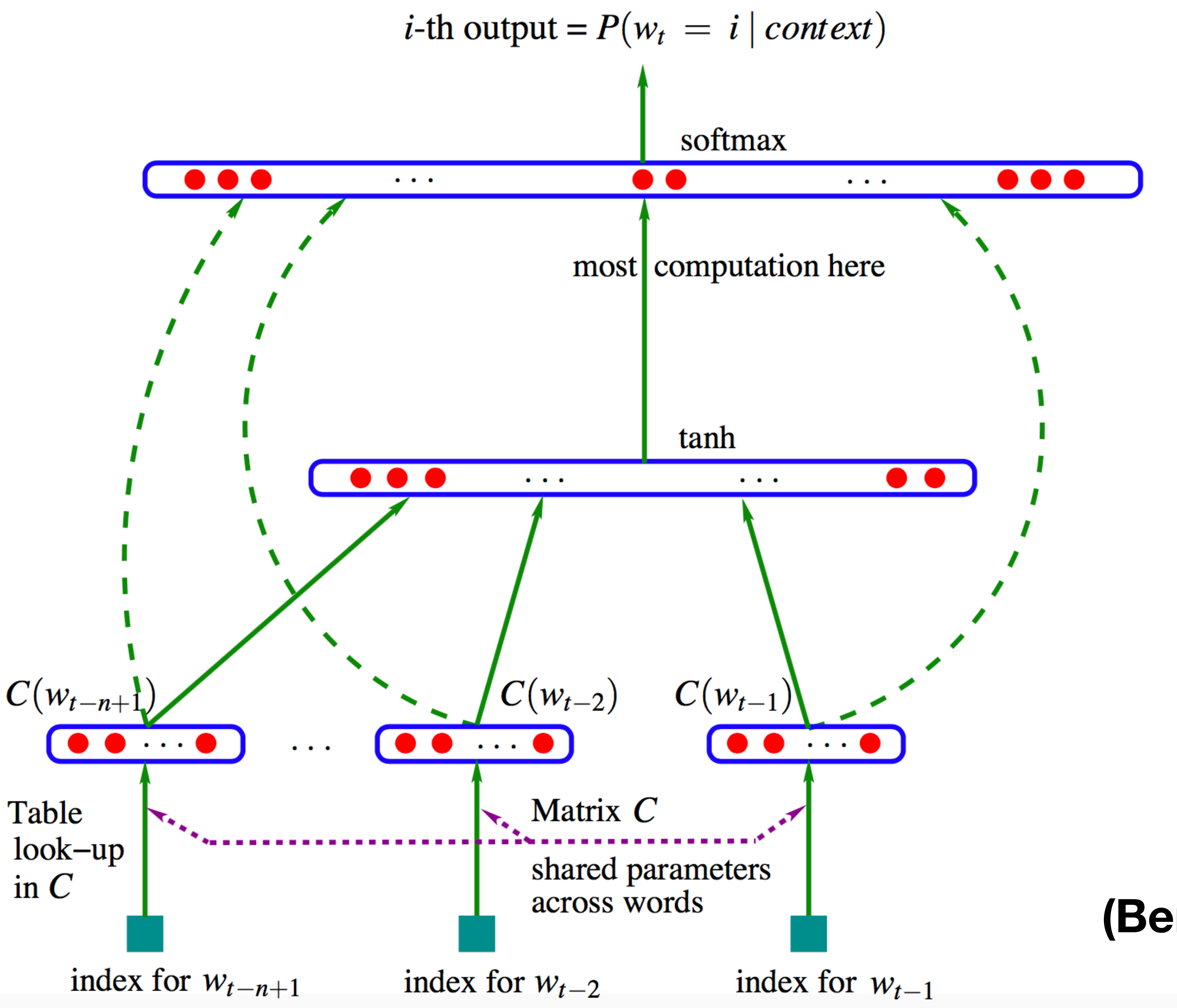
The boys went outside to _____

$$\hat{P}(w_t = w^k \,|\, w_1, \ldots, w_{t-1})$$

Objective: minimize the surprisal of the word that in fact occurred in the corpus:

$$-log\hat{P}(w)$$

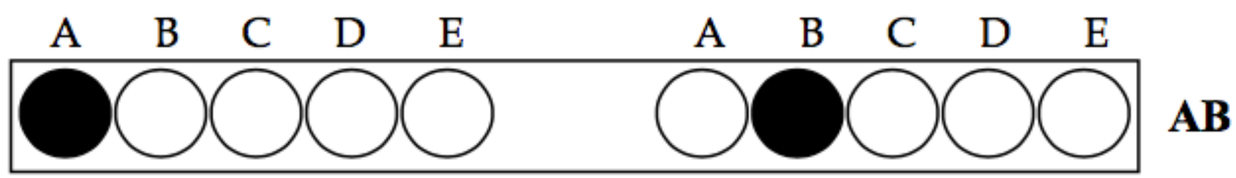
Neural language model



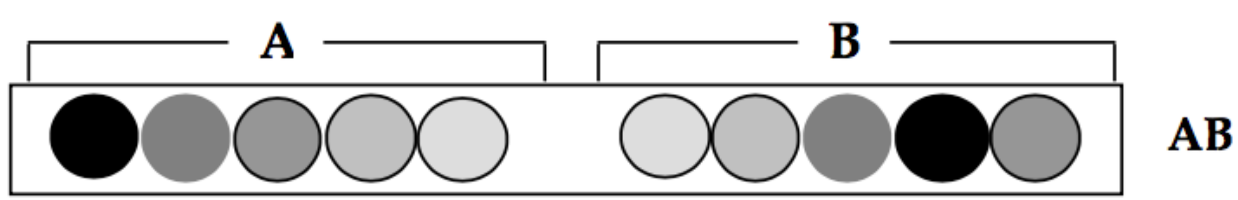
(Bengio et al., 2003)

How do we represent discrete inputs and outputs in a network?

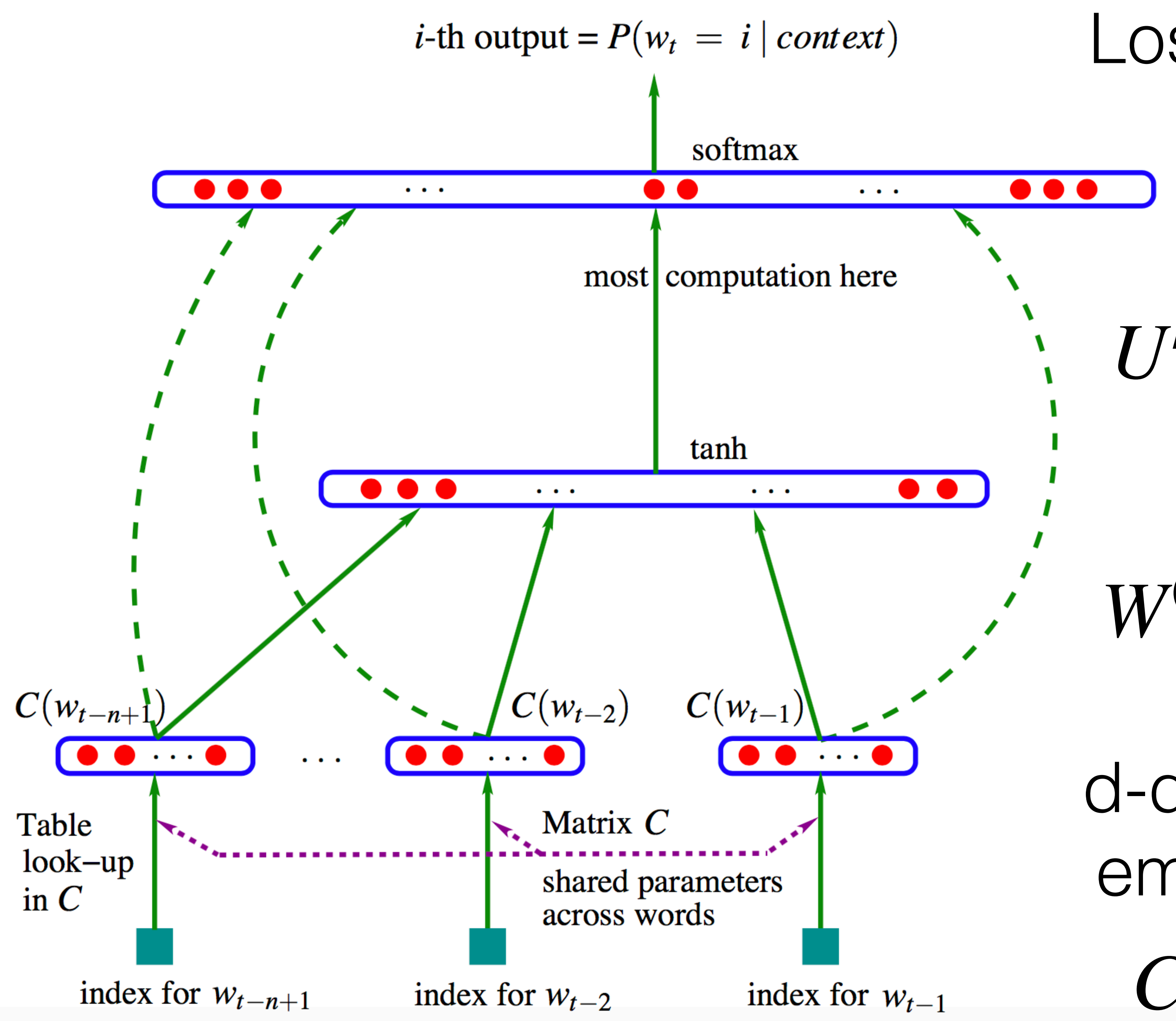
Localist ("one hot") representation: each input unit represents an item (e.g., a word)



Distributed representation: each item is represented by multiple units, and each unit participates in representing multiple items



Neural language model



Loss: $-log\hat{P}(w)$

$U^{h\times|V|}$

$W^{(n-1)d\times h}$

d-dimensional embeddings:

$C_1, \ldots, C_{|V|}$

The chain rule

$$(f(g(x)))' = f'(g(x))g'(x)$$

$$f(y) = y^2 \qquad g(x) = \sin x$$

$$f'(y) = 2y \qquad g'(x) = \cos x$$

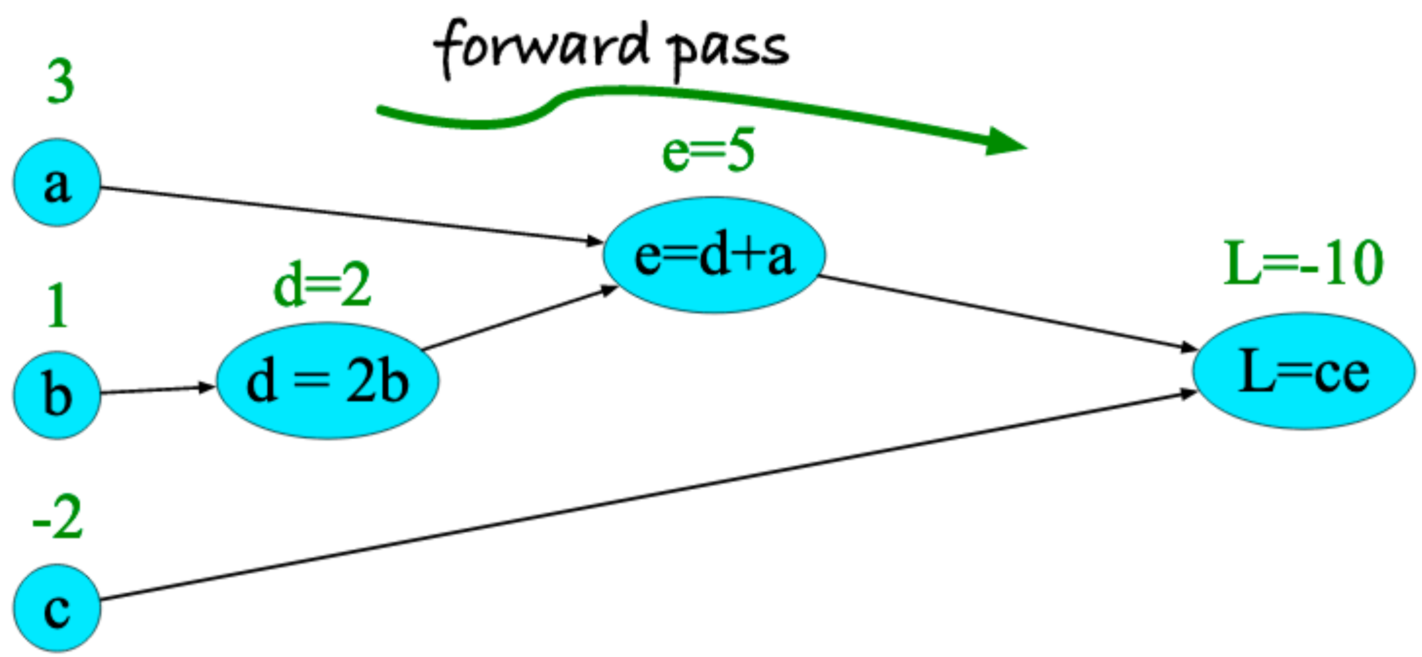
$$h(x) = f(g(x)) = (\sin x)^2$$

$$h'(x) = f'(g(x))g'(x) = 2\sin x \cos x$$

Substituting sin(x) for y in f’(y) = 2y

Computation graphs

$$L(a, b, c) = c(a + 2b)$$



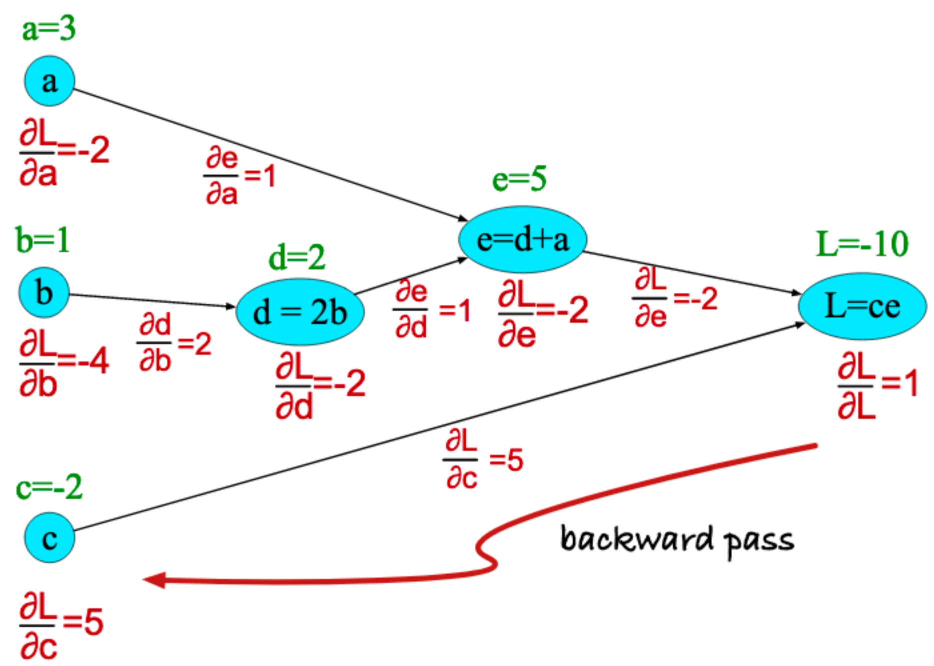
$$(f(g(x)))' = f'(g(x))g'(x)$$

Backpropagation

$$L(a, b, c) = c(a + 2b)$$

$$d = 2b \qquad e = a + d$$

$$L = ce \qquad \frac{\partial L}{\partial c} = e$$



$$\frac{\partial L}{\partial a} = \frac{\partial L}{\partial e}\frac{\partial e}{\partial a} \qquad \left.\frac{\partial L}{\partial a}\right|_{a} = \left.\frac{\partial L}{\partial e}\right|_{e(a)} \left.\frac{\partial e}{\partial a}\right|_{a}$$

$$\frac{\partial L}{\partial e} = c \qquad \frac{\partial e}{\partial a} = 1$$

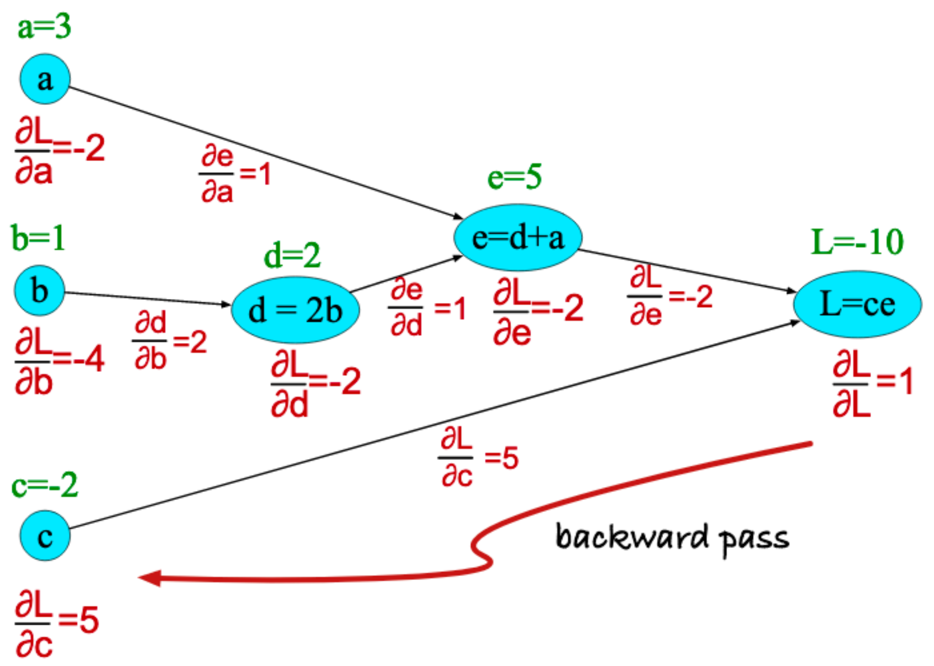
Backpropagation

$$L(a,b,c) = c(a+2b)$$

$$L = ce \qquad e = a + d \qquad d = 2b$$

$$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial e}\frac{\partial e}{\partial d}\frac{\partial d}{\partial b}$$

$$\frac{\partial L}{\partial e} = c \qquad \frac{\partial e}{\partial d} = 1 \qquad \frac{\partial d}{\partial b} = 2$$



Long-distance dependencies

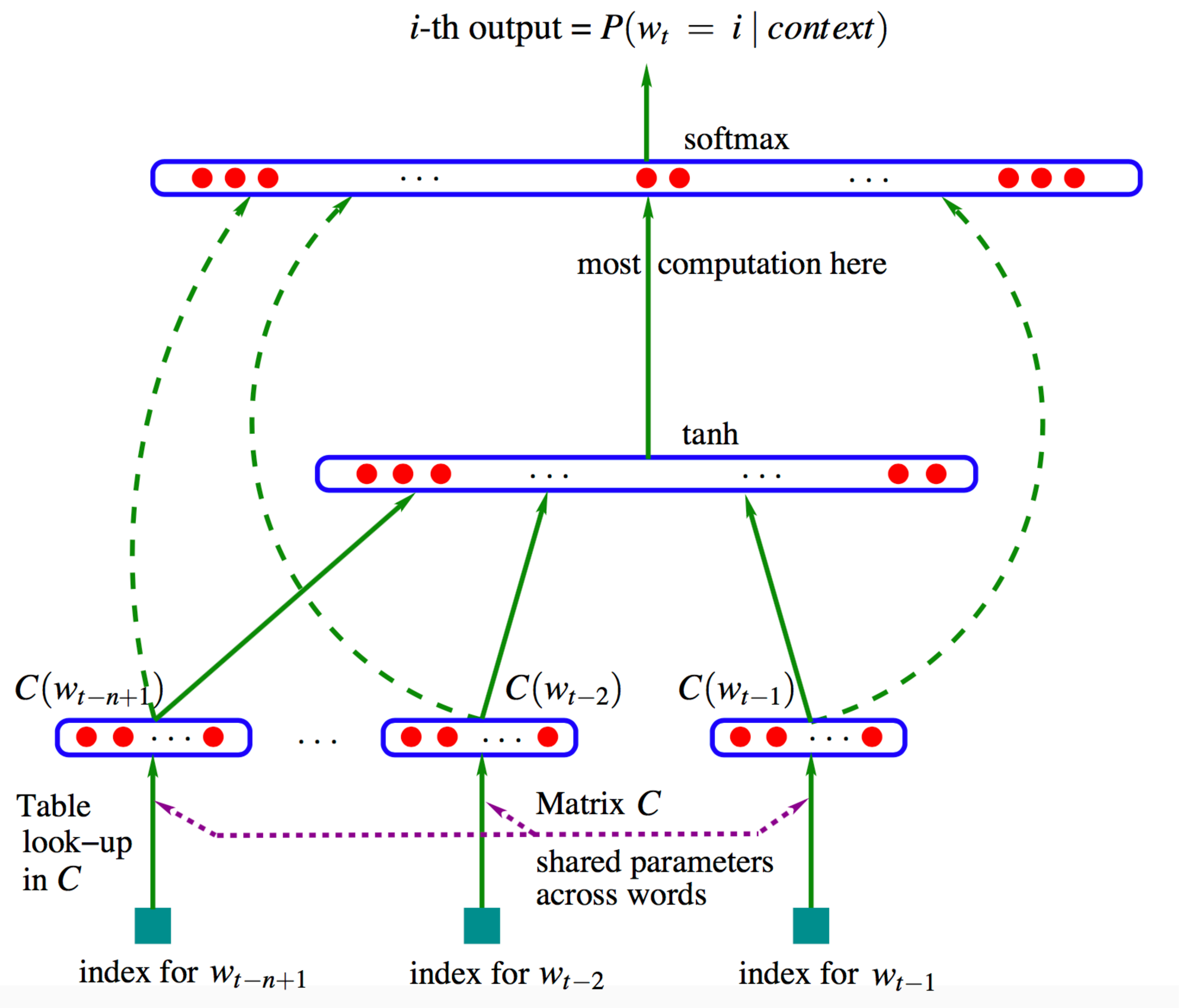
- A neural feedforward language model can generalize across n-grams (*easy money* → *easy cash*)

- But it still makes the Markov assumption, which ignores long-distance dependencies:

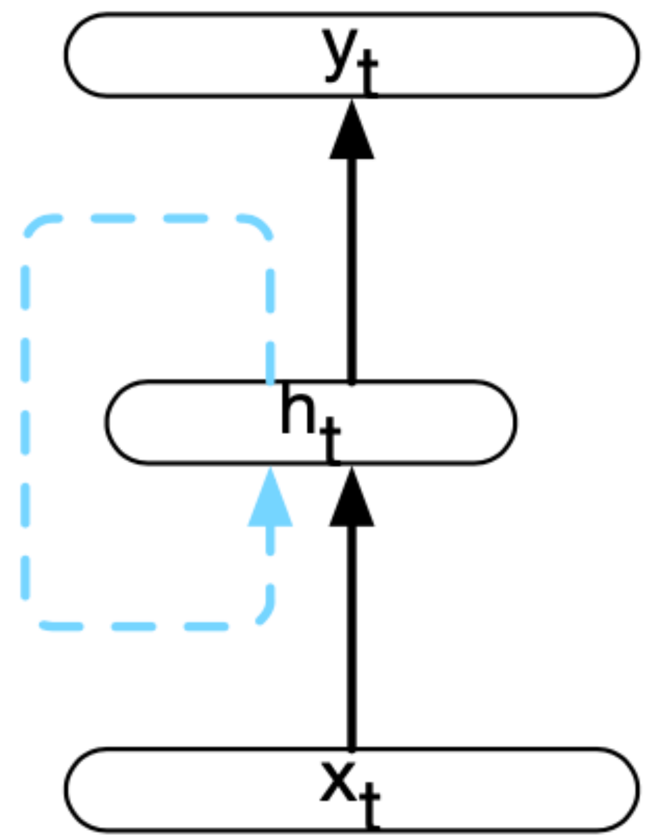
    - The **people** you saw at the grocery store last night **are** my friends.

    - I went to **Paris** but I didn't get a chance to see the rest of **France**.

Lack of temporal invariance



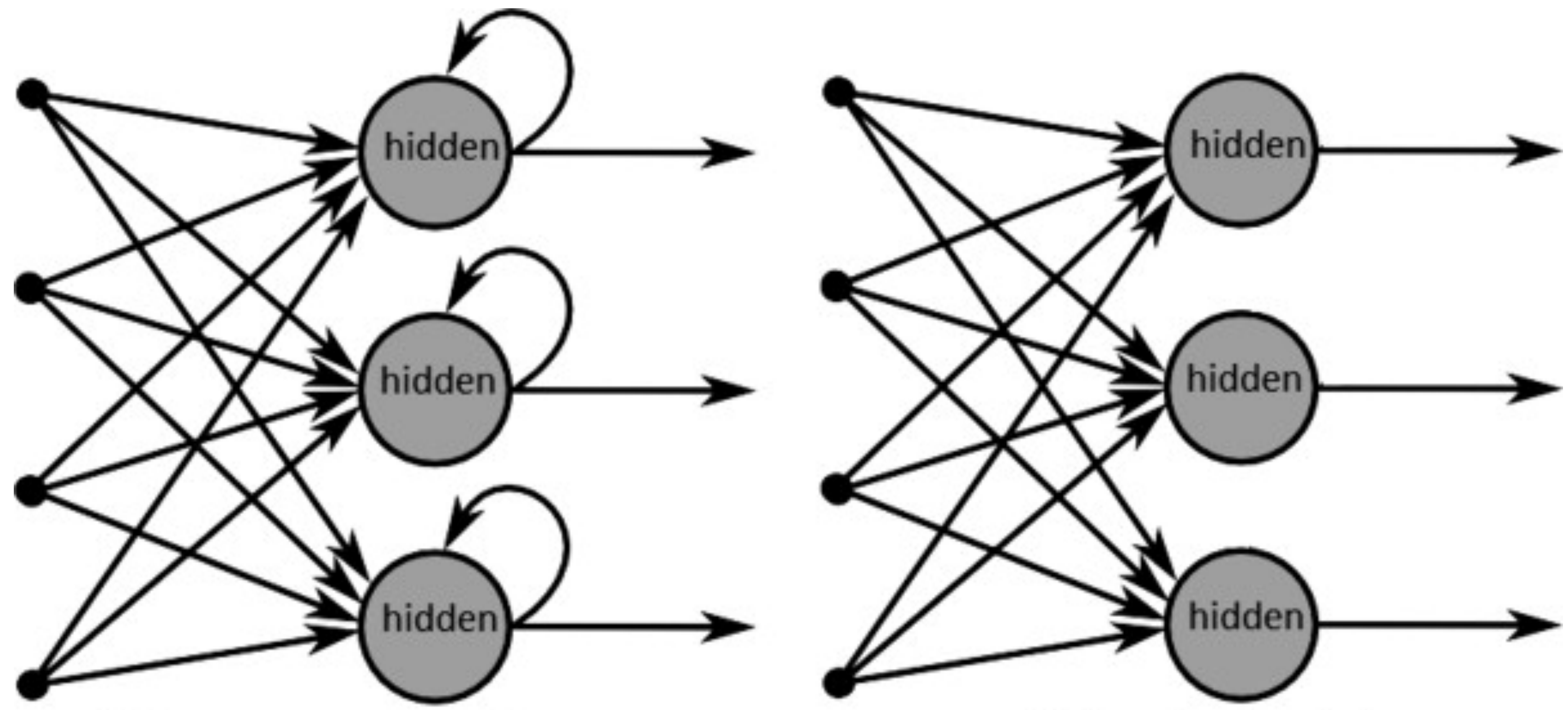
Recurrent neural network



(Elman, 1990)

Simple recurrent network

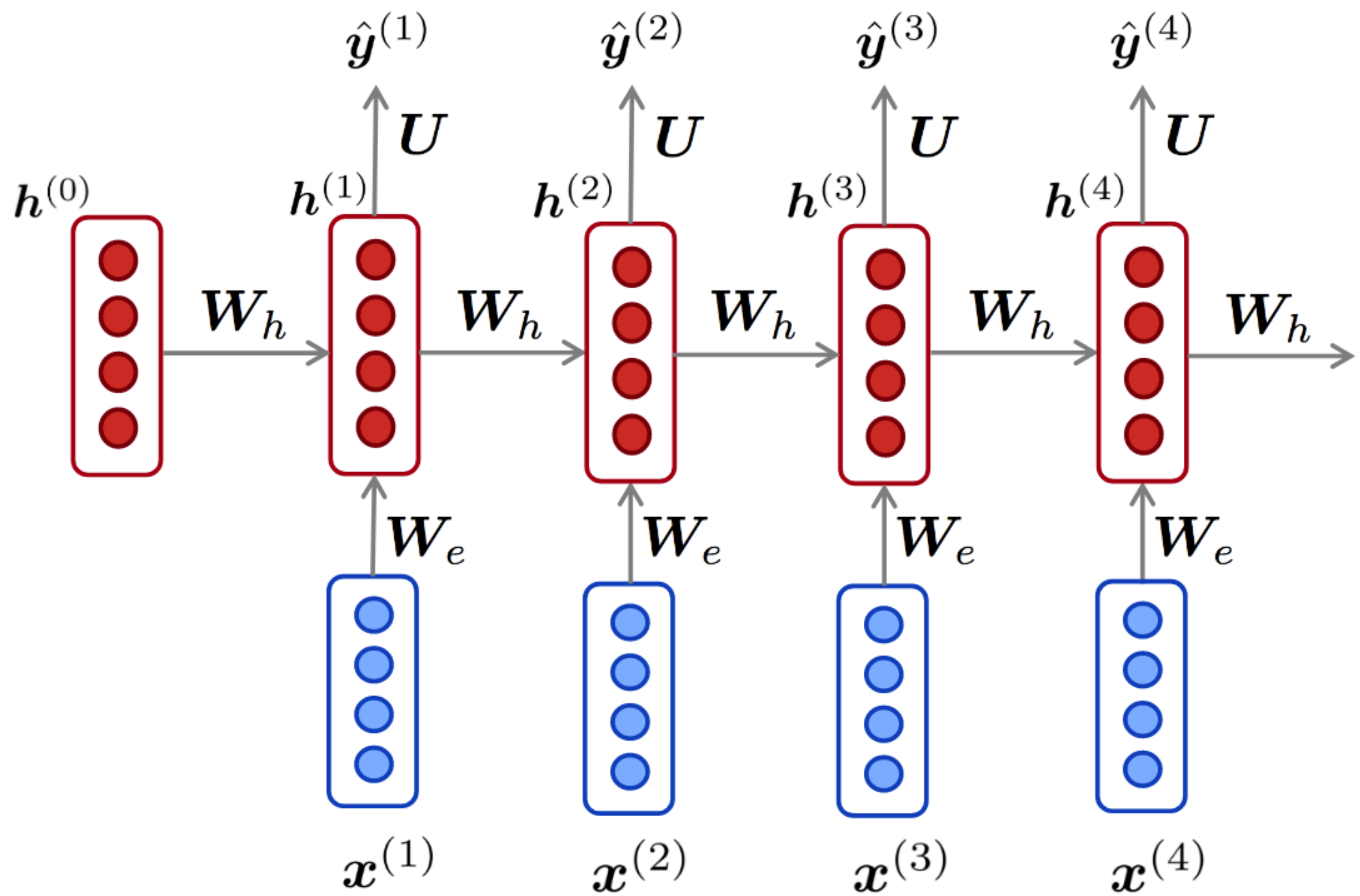
$$\overrightarrow{h}_t = U\vec{i}_t + W\overrightarrow{h}_{t-1} \qquad \overrightarrow{h}_t = U\vec{i}_t$$



(a) Recurrent neural network (b) Forward neural network

(figure from Mulder et al., 2015)

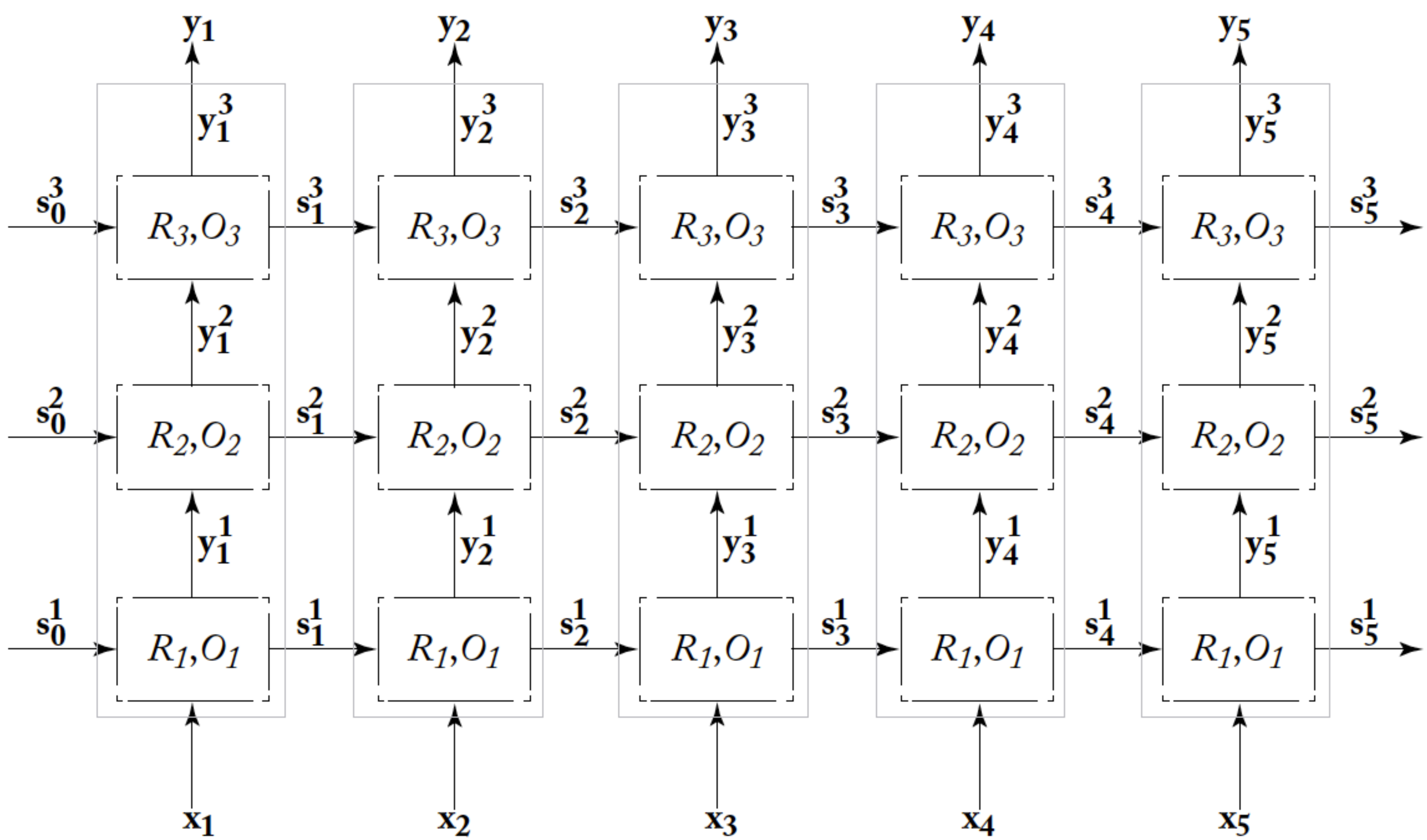
Unrolling an RNN



$$\frac{\partial L}{\partial W_h} = \sum_{t=1}^{T} \frac{\partial L_t}{\partial W_h}$$

(Figure credit: Richard Socher)

Stacked RNNs



(From Goldberg 2017)

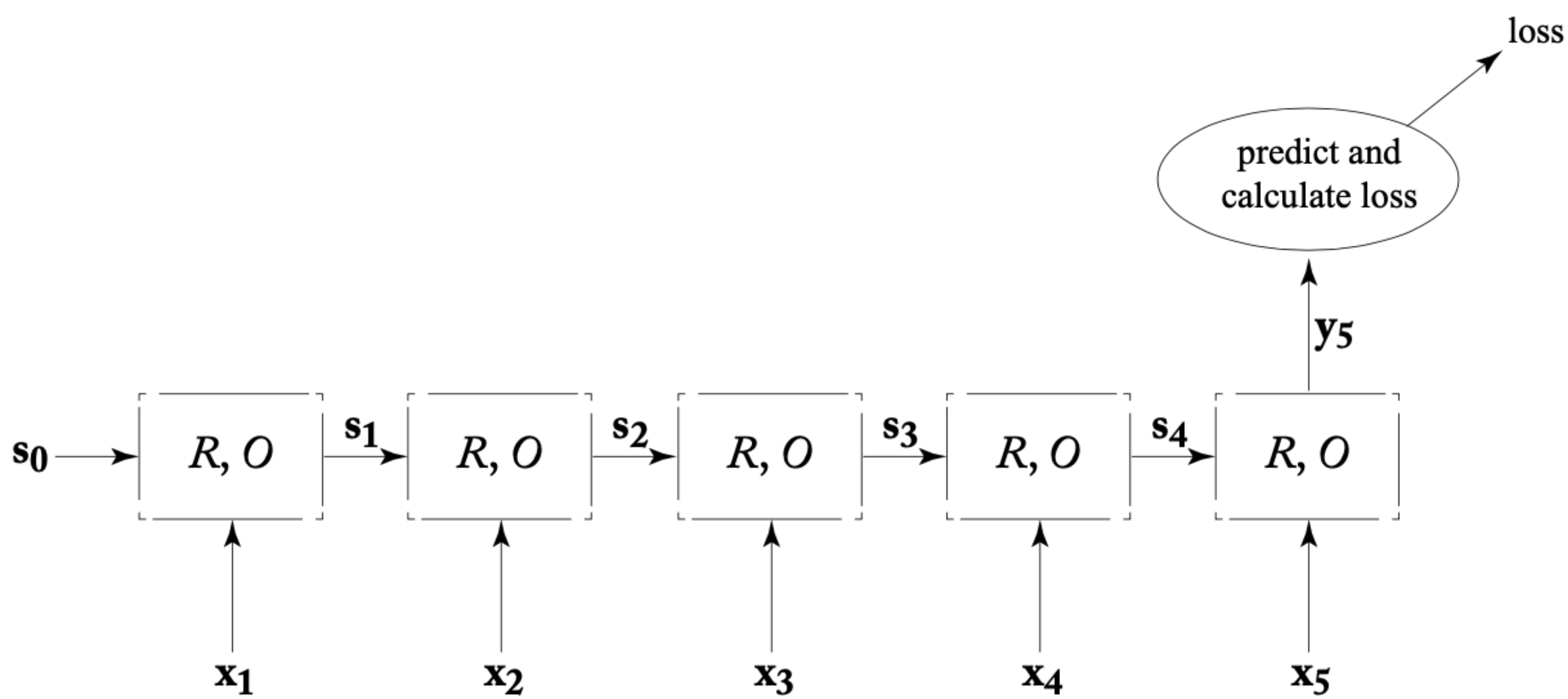
RNN as a language model

	PPL		WER	
Model	RNN	RNN+KN	RNN	RNN+KN
KN5 - baseline	-	221	-	13.5
RNN 60/20	229	186	13.2	12.6
RNN 90/10	202	173	12.8	12.2
RNN 250/5	173	155	12.3	11.7
RNN 250/2	176	156	12.0	11.9
RNN 400/10	171	152	12.5	12.1
3xRNN static	151	143	11.6	11.3
3xRNN dynamic	128	121	11.3	11.1

(Mikolov et al., 2010)

Sequence classification

Sentiment analysis, language classification, authorship identification, genre classification…

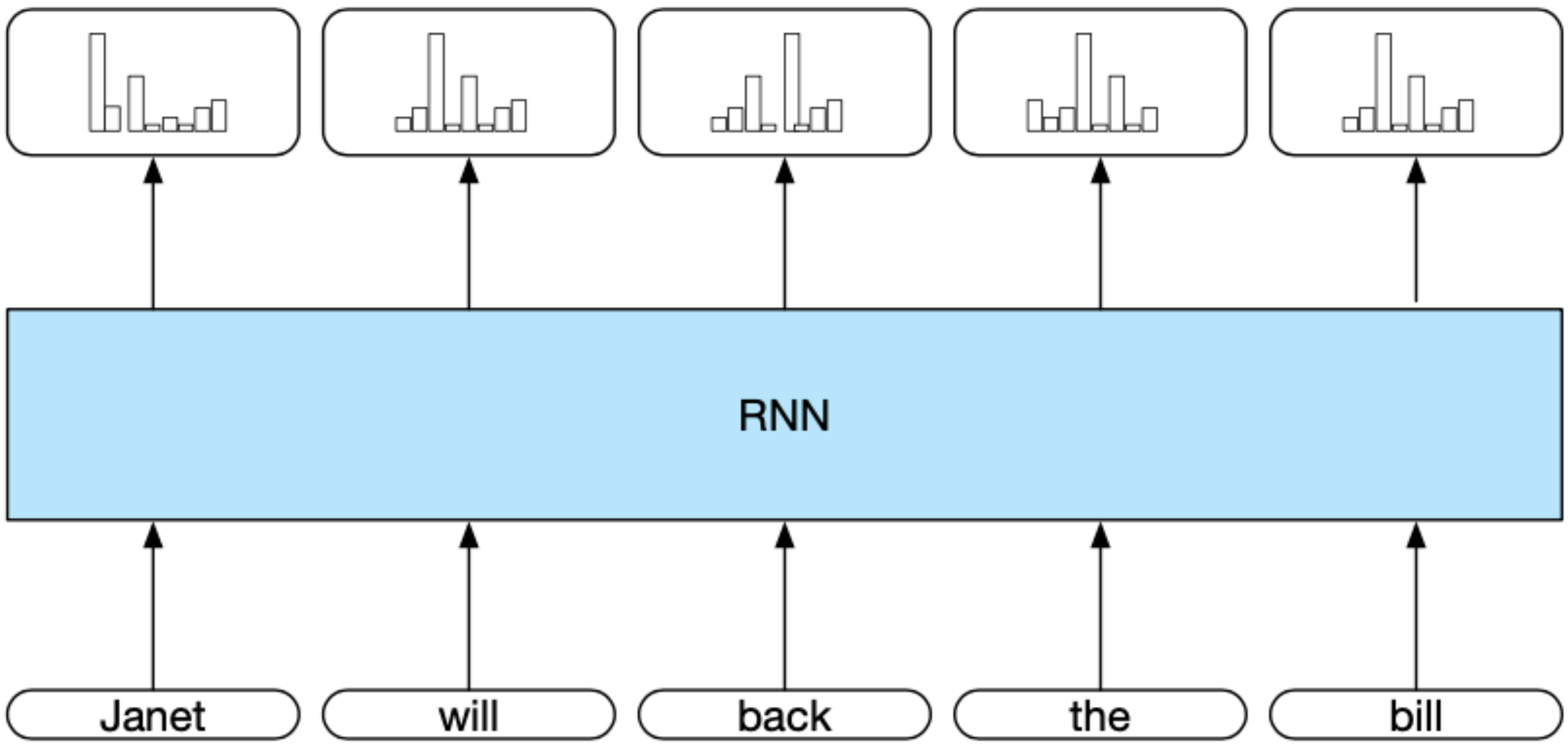


(Figure from Goldberg, 2017)

Sequence tagging

- Part-of-speech tagging:
  - The cat is about to **fall** from the tree.
  - Last **fall** I traveled to Europe.
- Named entity recognition:
  - Seattle is in **Washington**.
  - **Washington** was the first president of the United States.

Sequence tagging



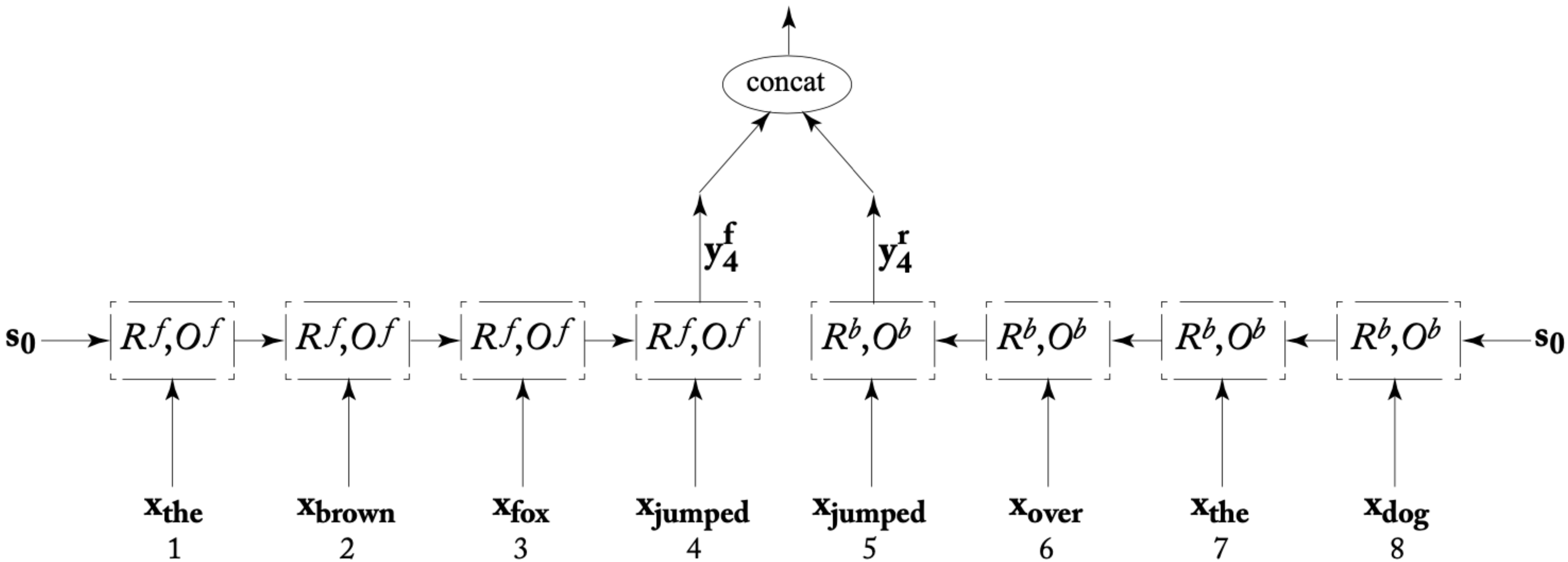
Bidirectional context can help

- **Will** gets his revenge by masquerading as Sue’s hairdresser and forcibly shaving her head bald.

- **Will** putting a patch over my eye help to get the object out of it?

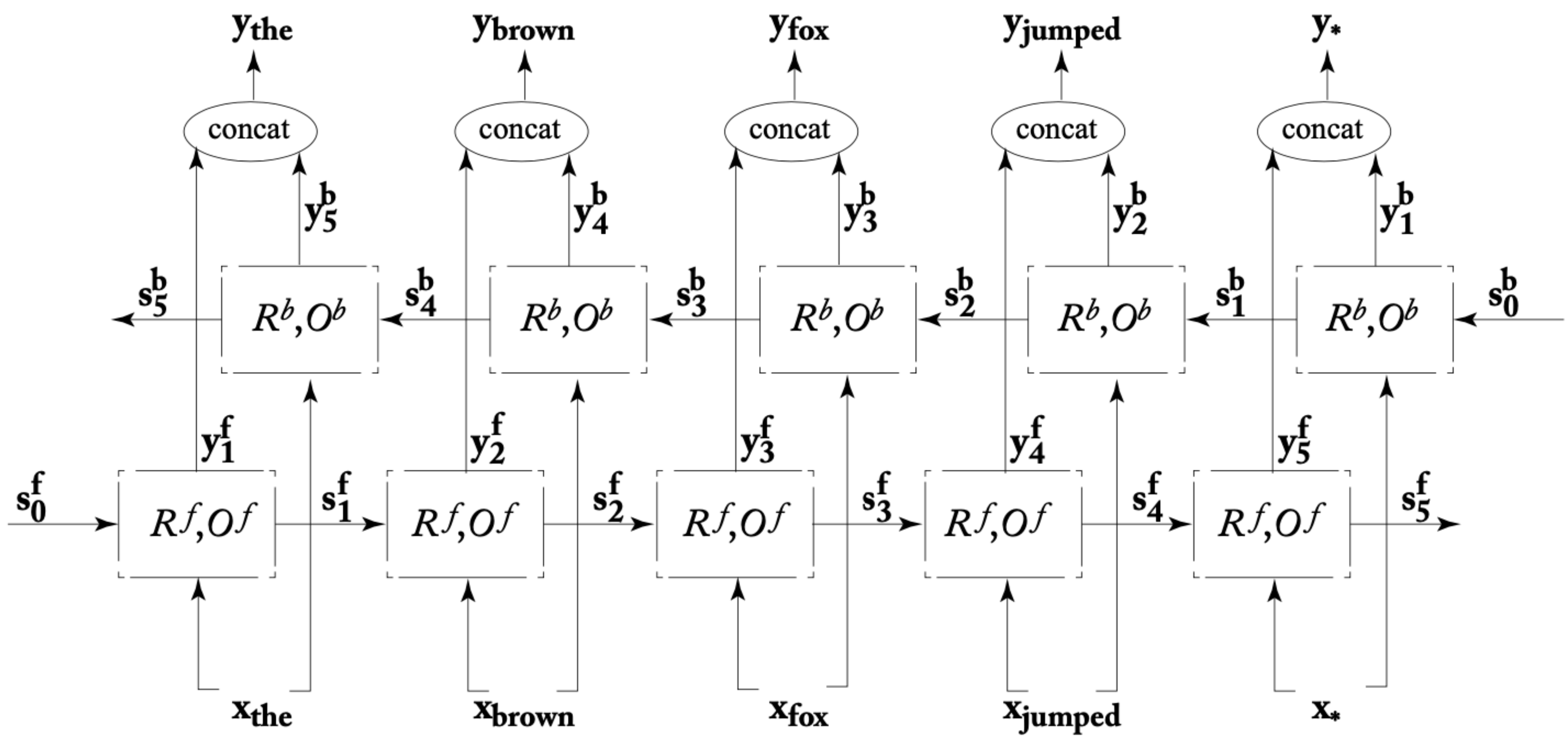
(Elkahky et al. 2018)

Bidirectional RNN



(From Goldberg 2017)

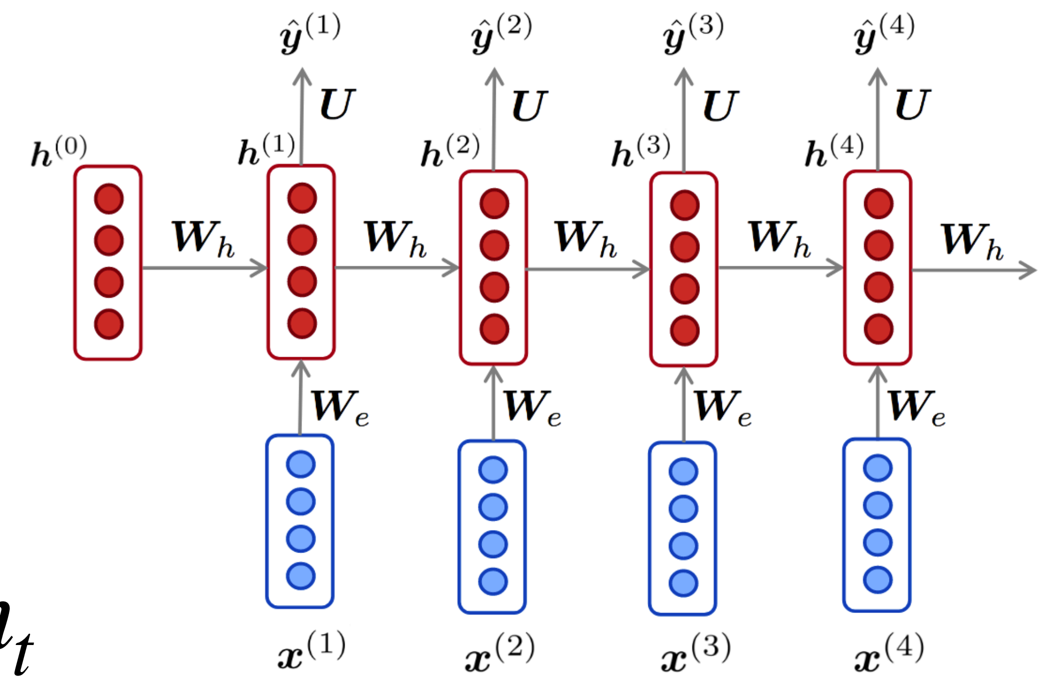
Bidirectional RNN



(From Goldberg 2017)

Vanishing gradients

$$\frac{\partial L_t}{\partial W_h} = \sum_{k=1}^{t} \frac{\partial L_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \boxed{\frac{\partial h_t}{\partial h_k}} \frac{\partial h_k}{\partial W_h}$$



$$\frac{\partial h_t}{\partial h_k} = \prod_{j=k+1}^{t} \frac{\partial h_j}{\partial h_{j-1}} = \frac{\partial h_{k+1}}{\partial h_k} \cdots \frac{\partial h_t}{\partial h_{t-1}}$$

$$\frac{\partial h_j}{\partial h_{j-1}} = W_h \, \mathbf{diag}(\sigma'(h_{j-1}))$$

(Pascanu et al., 2013)

Vanishing gradients

$$\frac{\partial h_j}{\partial h_{j-1}} = W_h \, \mathbf{diag}(\sigma'(h_{j-1}))$$

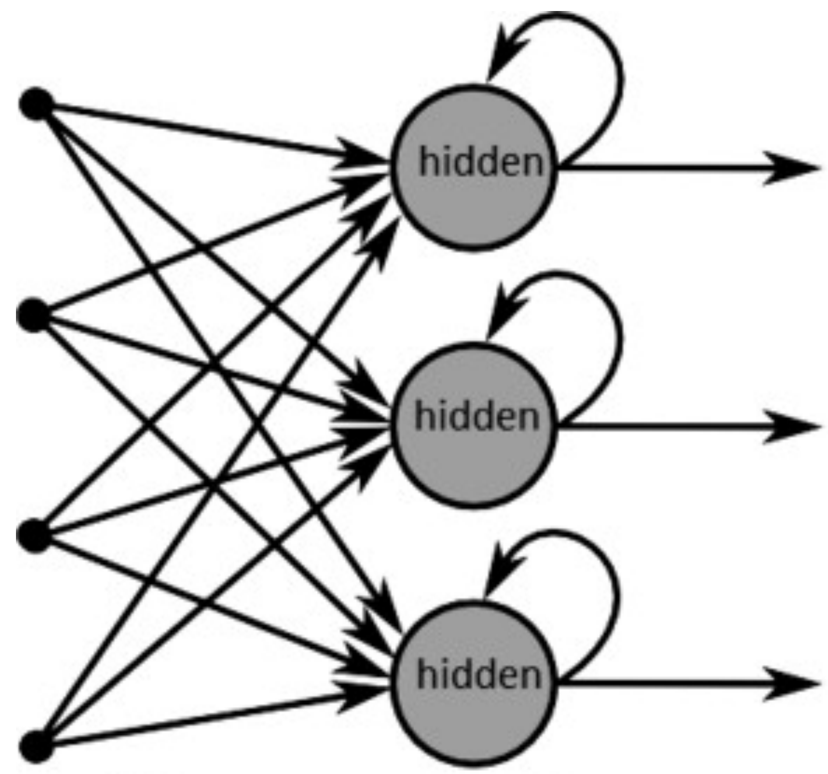
$$\left\| \frac{\partial h_j}{\partial h_{j-1}} \right\| \le \underbrace{\| W_h \|}_{\text{Fixed}} \, \underbrace{\| \mathbf{diag}(\sigma'(h_{j-1})) \|}_{\text{Bounded}} \le \| W_h \| \gamma$$

$$\left\| \frac{\partial h_t}{\partial h_k} \right\| \le (\| W_h \| \gamma)^{t-k}$$

(Pascanu et al., 2013)

Simple recurrent network

$$\overrightarrow{h}_t = U\vec{i}_t + W\overrightarrow{h}_{t-1}$$



(a) Recurrent neural network

LSTM (“long short-term memory”)

(Hochreiter & Schmidhuber 1997; figure from Ma & Hovy 2016)

$$c_t = f_t c_{t-1} + i_t g_t$$

$$z_t = \textbf{concat}(D_{t-1}, x_t)$$

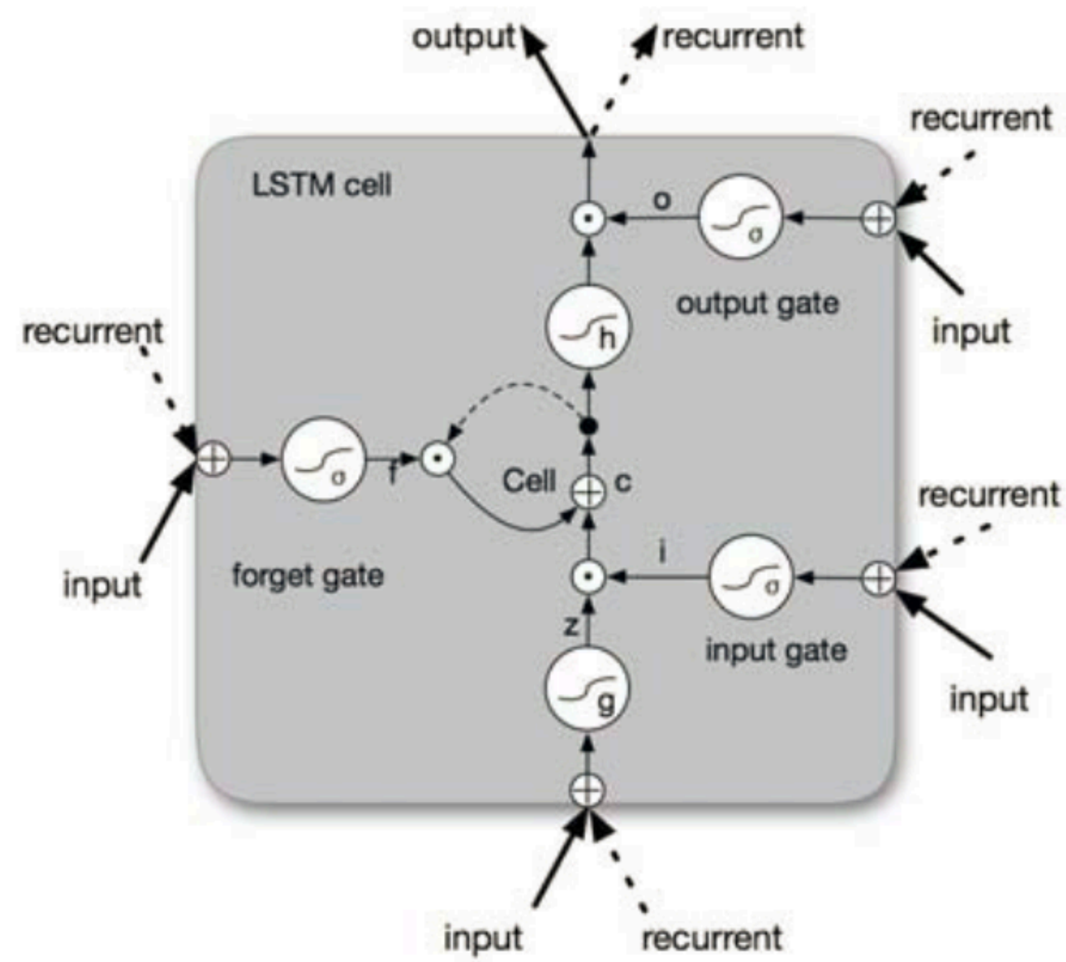
$$i_t = \sigma(W_i z_t + b_i)$$

$$f_t = \sigma(W_f z_t + b_f)$$

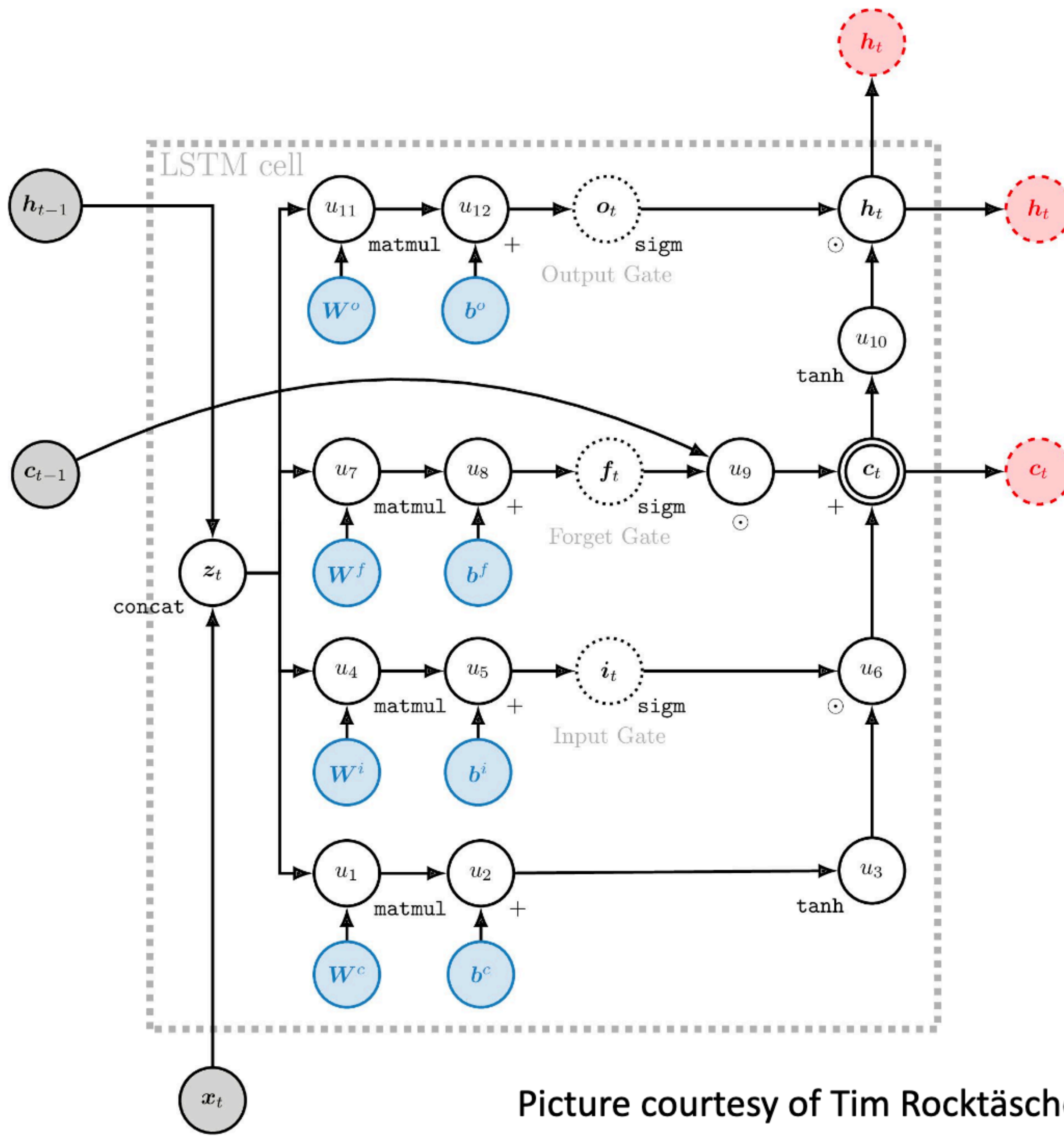
$$g_t = \tanh(W_g z_t + b_g)$$

$$D_t = o_t \tanh(c_t)$$

$$o_t = \sigma(W_o z_t + b_o)$$



LSTM computation graph



Picture courtesy of Tim Rocktäschel

LSTM language models

MODEL	TEST PERPLEXITY	NUMBER OF PARAMS [BILLIONS]
SIGMOID-RNN-2048 (JI ET AL., 2015A)	68.3	4.1
INTERPOLATED KN 5-GRAM, 1.1B N-GRAMS (CHELBA ET AL., 2013)	67.6	1.76
SPARSE NON-NEGATIVE MATRIX LM (SHAZEER ET AL., 2015)	52.9	33
RNN-1024 + MAXENT 9-GRAM FEATURES (CHELBA ET AL., 2013)	51.3	20
LSTM-512-512	54.1	0.82
LSTM-1024-512	48.2	0.82
LSTM-2048-512	43.7	0.83
LSTM-8192-2048 (NO DROPOUT)	37.9	3.3
LSTM-8192-2048 (50% DROPOUT)	32.2	3.3
2-LAYER LSTM-8192-1024 (BIG LSTM)	30.6	1.8
BIG LSTM+CNN INPUTS	**30.0**	**1.04**

(Jozefowicz et al., 2016)

Gated Recurrent Units

Reset gate: $$r_t = \sigma(U_r h_{t-1} + W_r x_t)$$

Update gate: $$z_t = \sigma(U_z h_{t-1} + W_z x_t)$$

$$\tilde{h}_t = \tanh(U(r_t h_{t-1}) + W x_t)$$

$$h_t = (1 - z_t) h_{t-1} + z_t \tilde{h}_t$$

(Cho et al. 2014)