Applications of Neural Networks

MPhil ACS Advanced Topics in NLP

Laura Rimell
25 February 2016

NLP Neural Network Applications

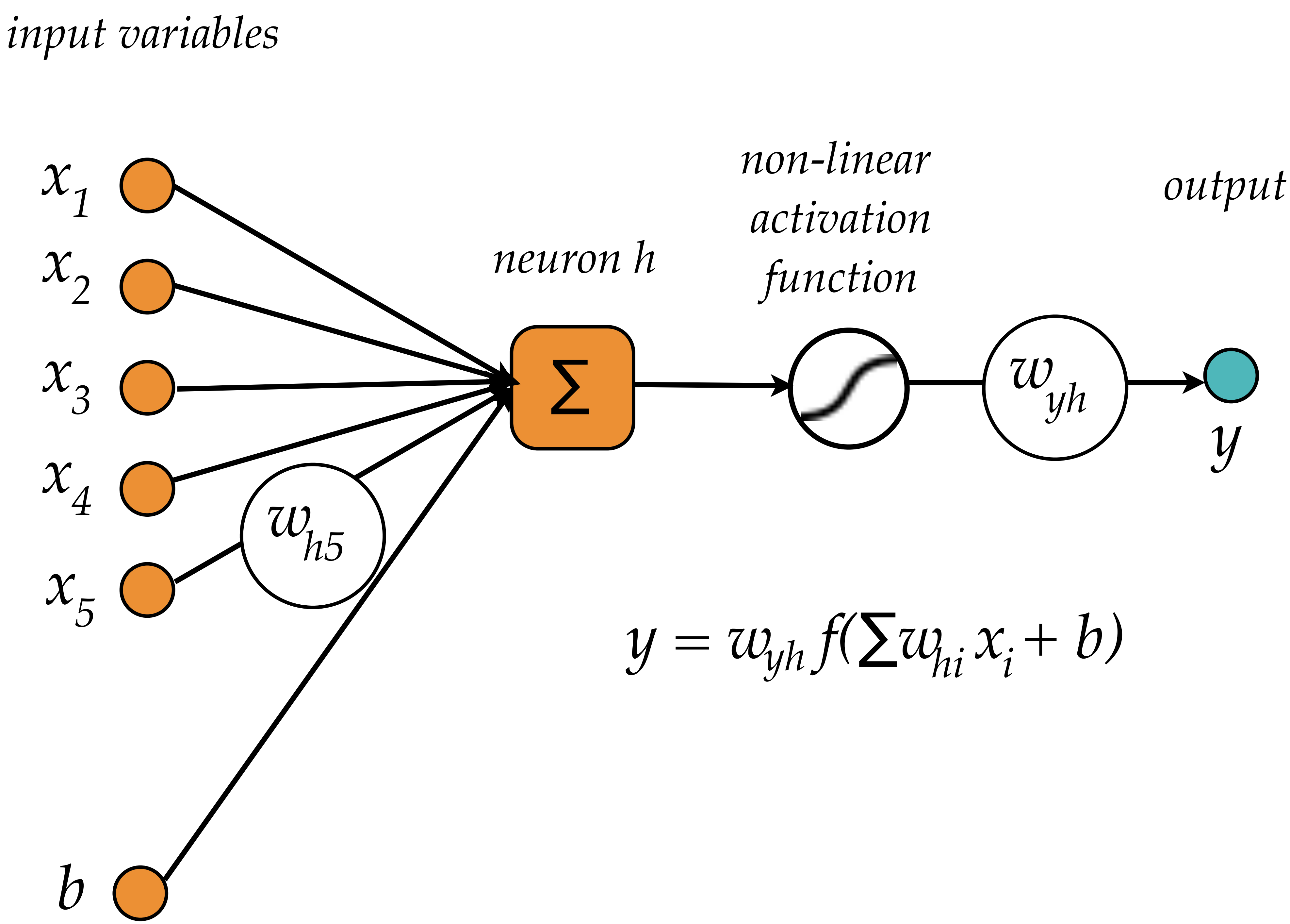
- Language Models
- Word Embeddings
- Tagging
- Parsing
- Sentiment
- Machine Translation

A neural network "dream"
(Google Research)

Outline

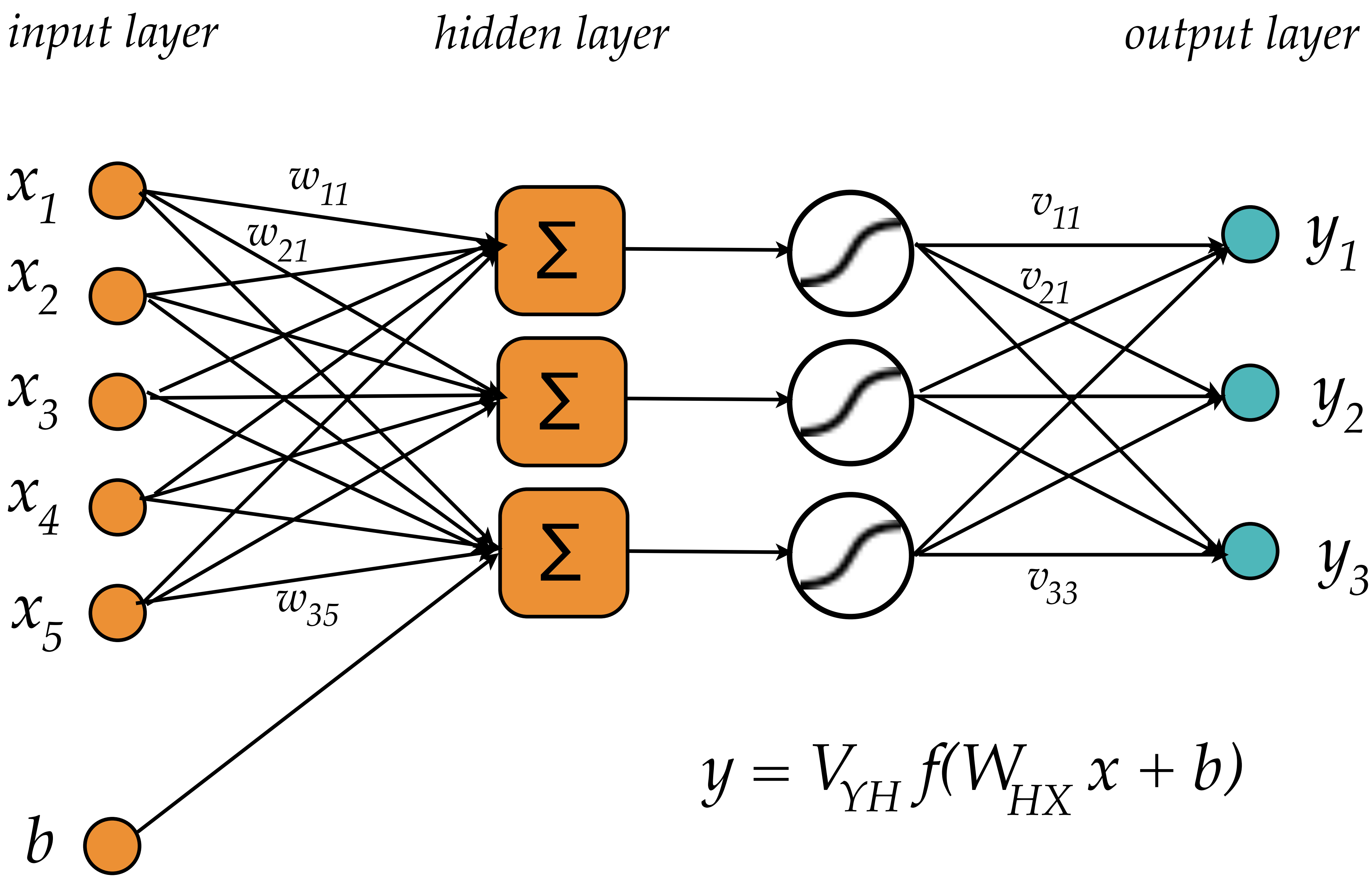
- **Neural network basics**
- NN architectures
  - Feedforward Networks and Backpropagation
  - Recursive Neural Networks
  - Recurrent Neural Networks
- Applications
  - Tagging
  - Parsing
  - Machine Translation and Encoder-Decoder Networks

Neuron



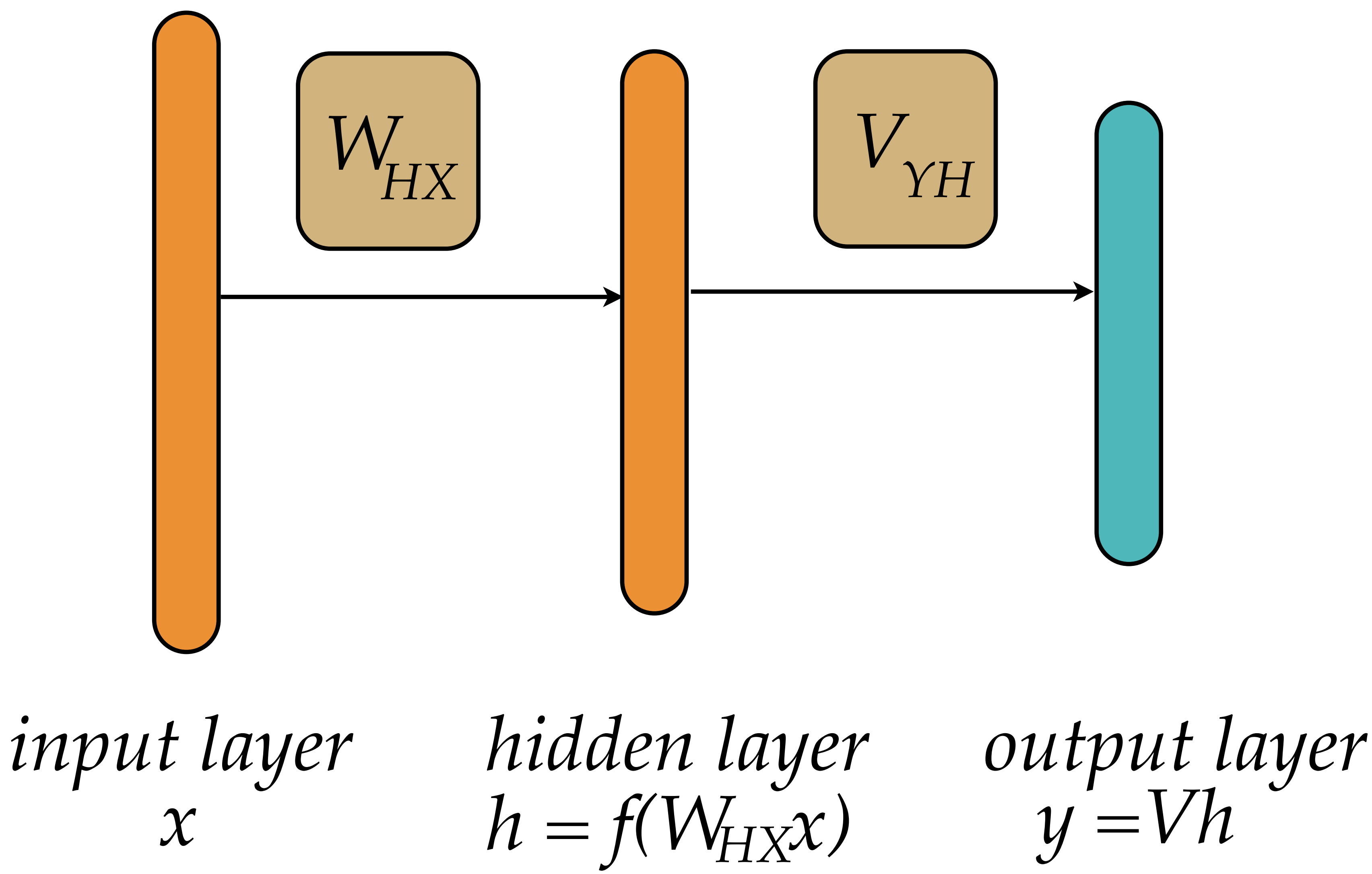
$$y = w_{yh} f(\sum w_{hi} x_i + b)$$

Neural Network



$$y = V_{YH} f(W_{HX} x + b)$$

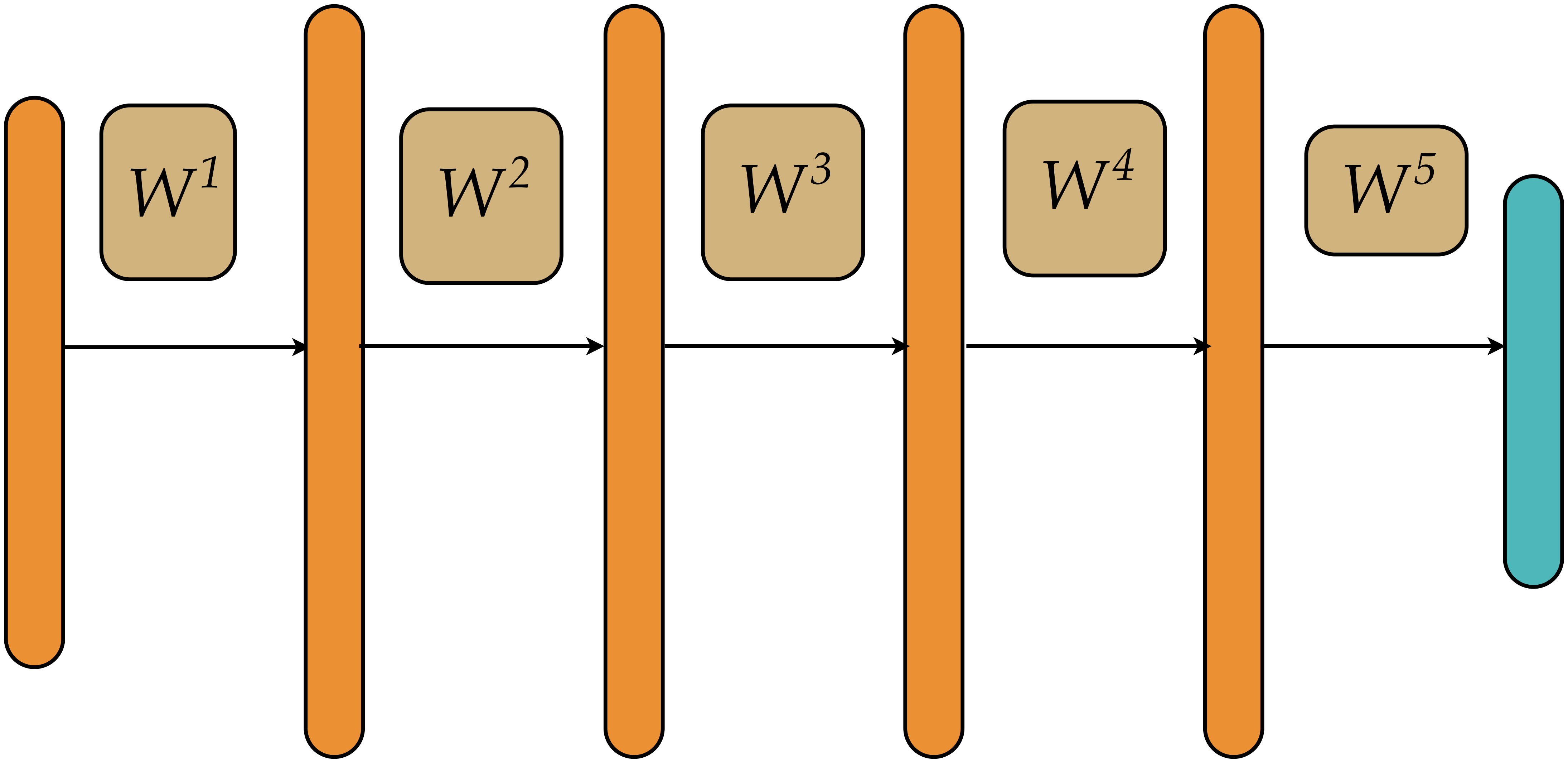
Neural Network



Outline

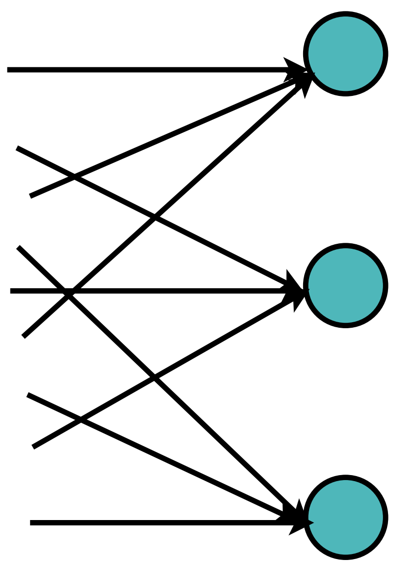
- Neural network basics
- **NN architectures**
  - **Feedforward Networks and Backpropagation**
  - Recursive Neural Networks
  - Recurrent Neural Networks
- Applications
  - Tagging
  - Parsing
  - Machine Translation and Encoder-Decoder Networks

Deep Feed-Forward Network

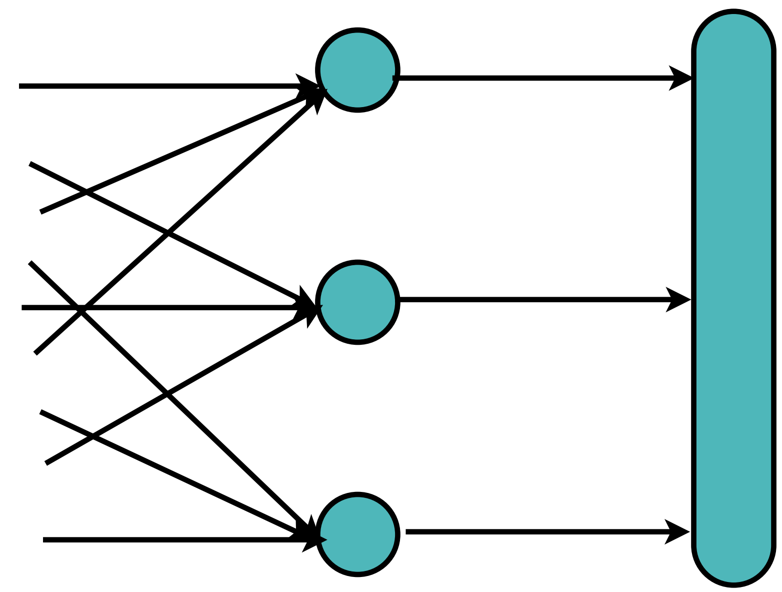


Output Layers

vector

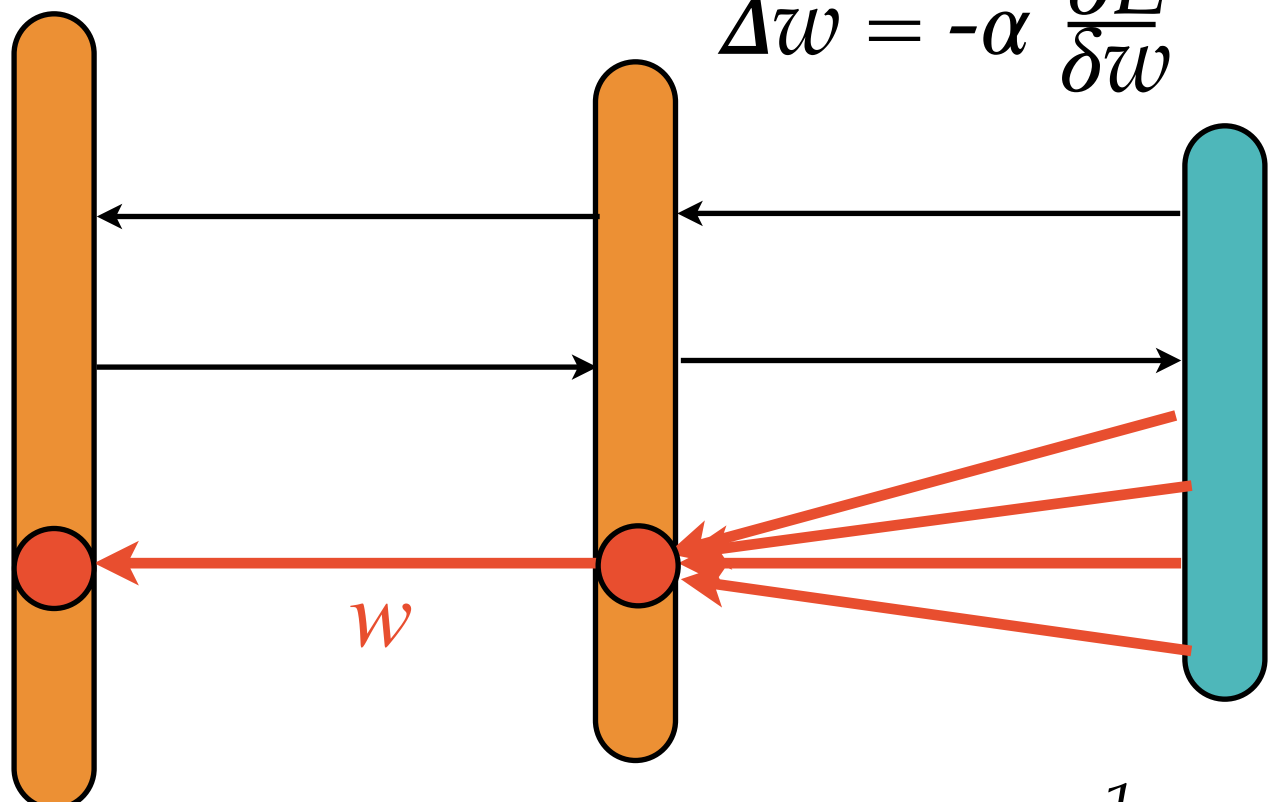


softmax



$$y_k = \frac{e^{\sum_i w_{ki} x_i}}{\sum_{l=1}^{3} e^{\sum_i w_{li} x_i}}$$

Backpropagation



$$\Delta w = -\alpha \frac{\delta E}{\delta w}$$

mean squared error

$$E = \frac{1}{n} \sum (t_k - y_k)^2$$

$$E = -\frac{1}{n} \sum t_k \ln(y_k) + (1-t_k)\ln(y_k)$$

cross-entropy

- Gradient descent
- SGD, Adagrad, Adadelta, ...

Feed-Forward Network Limitations

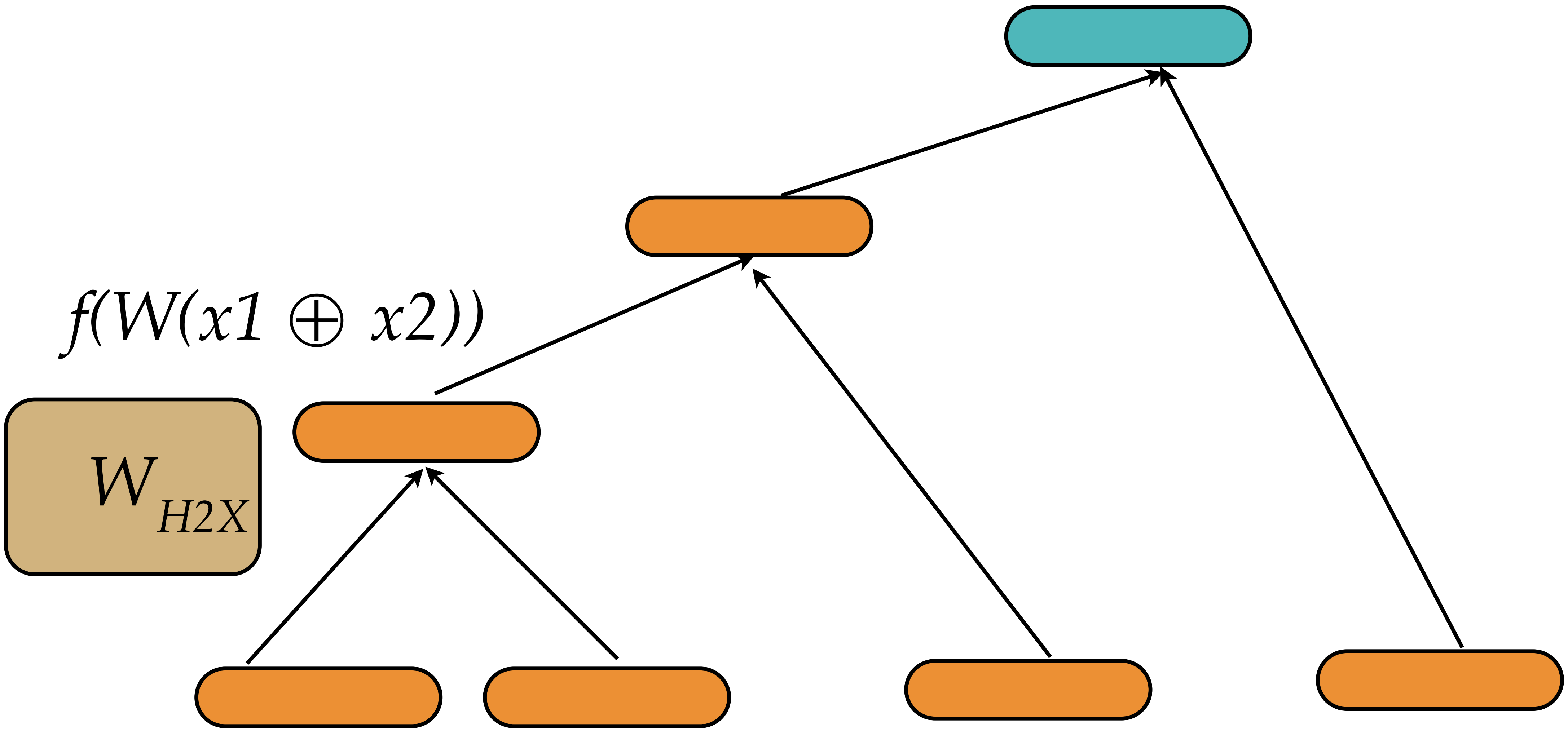
- Fixed-size input vector
- Fixed-size output vector
- Fixed number of computational steps (layers)

Outline

- Neural network basics
- **NN architectures**
  - Feedforward Networks and Backpropagation
  - **Recursive Neural Networks**
  - Recurrent Neural Networks
- Applications
  - Tagging
  - Parsing
  - Machine Translation and Encoder-Decoder Networks

Recursive Neural Network (RecNN)

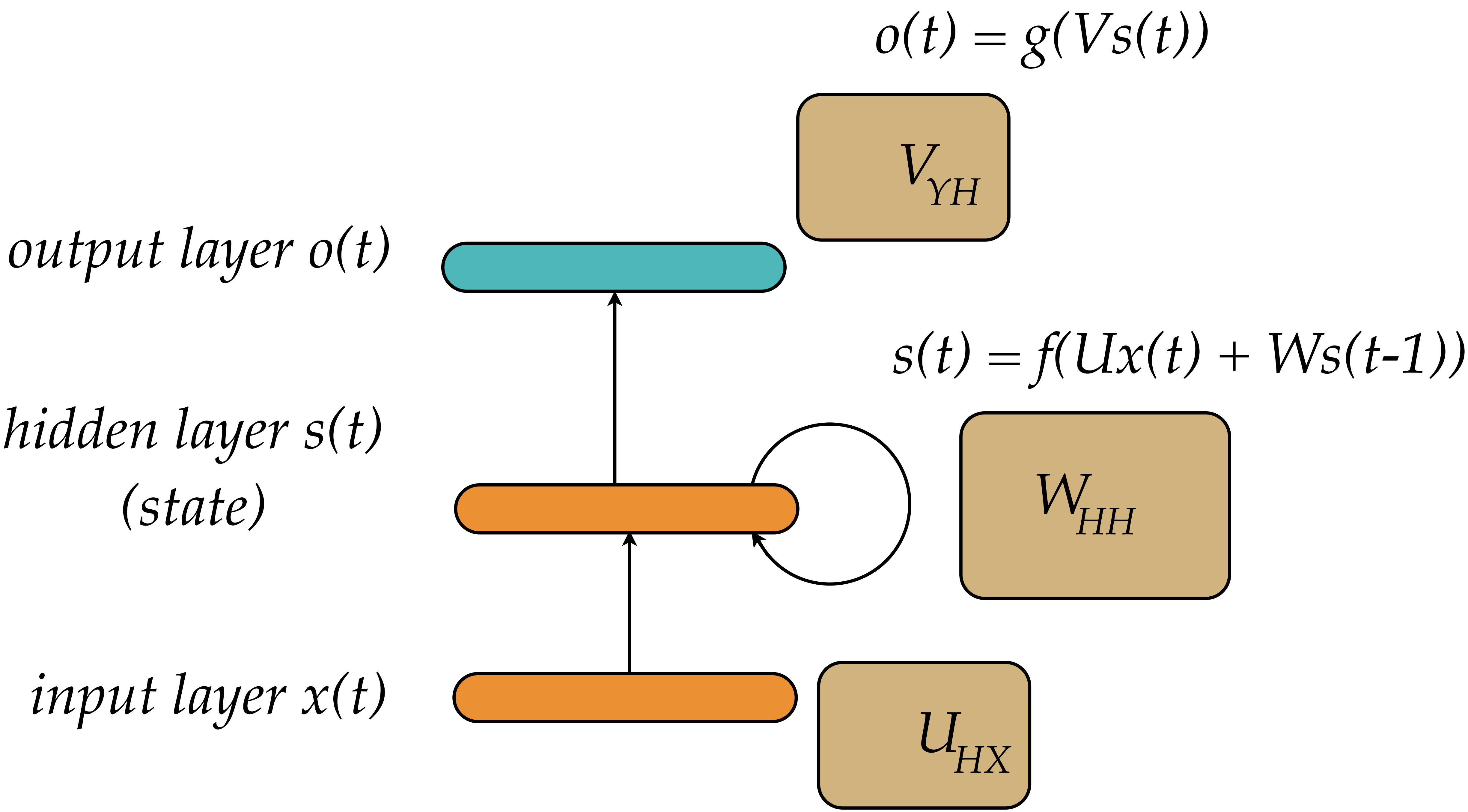
- One weight matrix (simplest version)



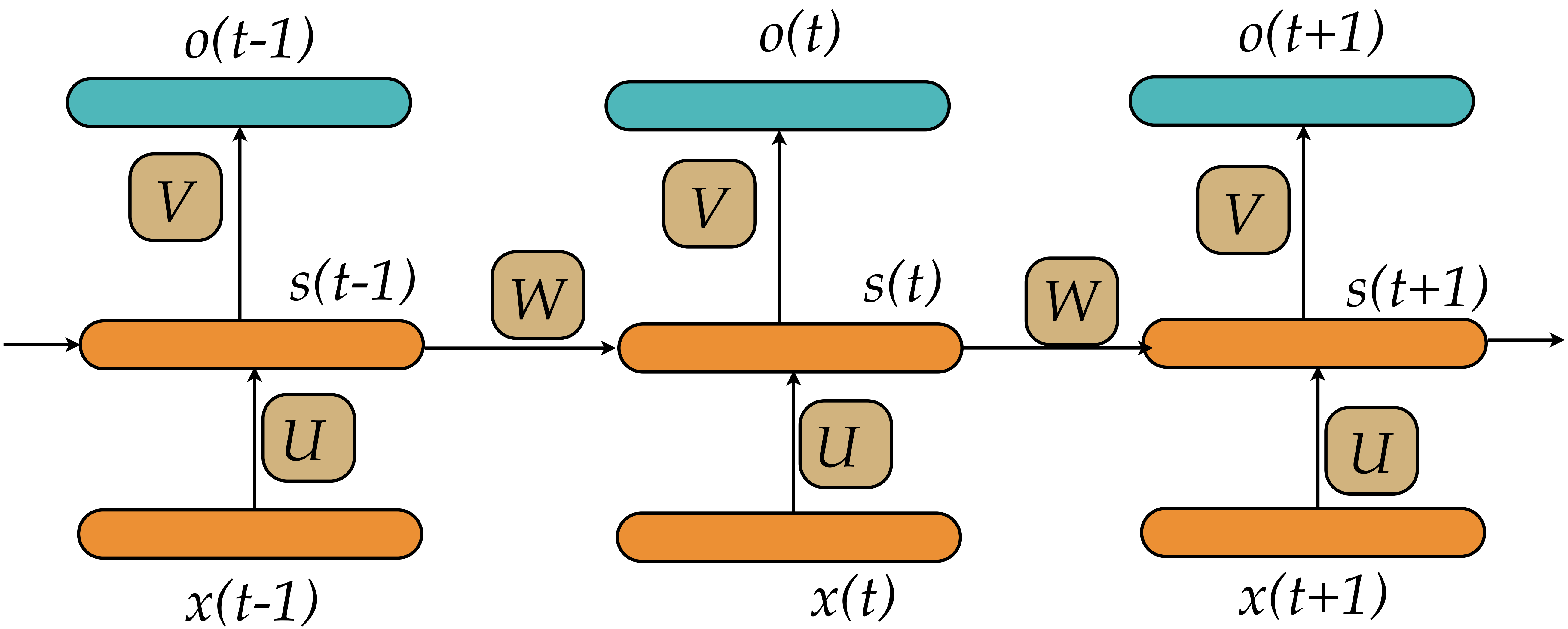
Outline

- Neural network basics
- **NN architectures**
  - Feedforward Networks and Backpropagation
  - Recursive Neural Networks
  - **Recurrent Neural Networks**
- Applications
  - Tagging
  - Parsing
  - Machine Translation and Encoder-Decoder Networks

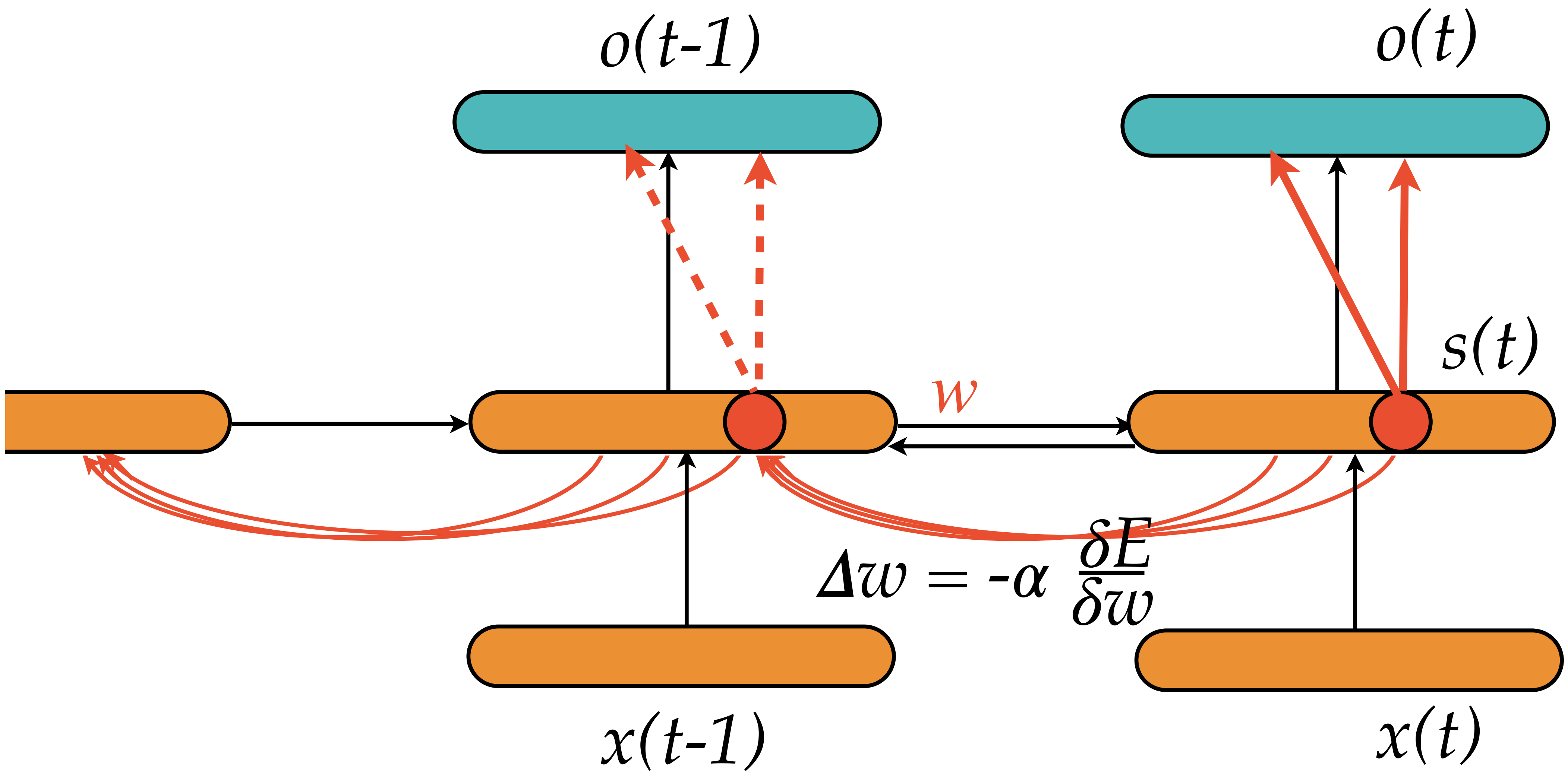
Recurrent Neural Network



Unfolded RNN



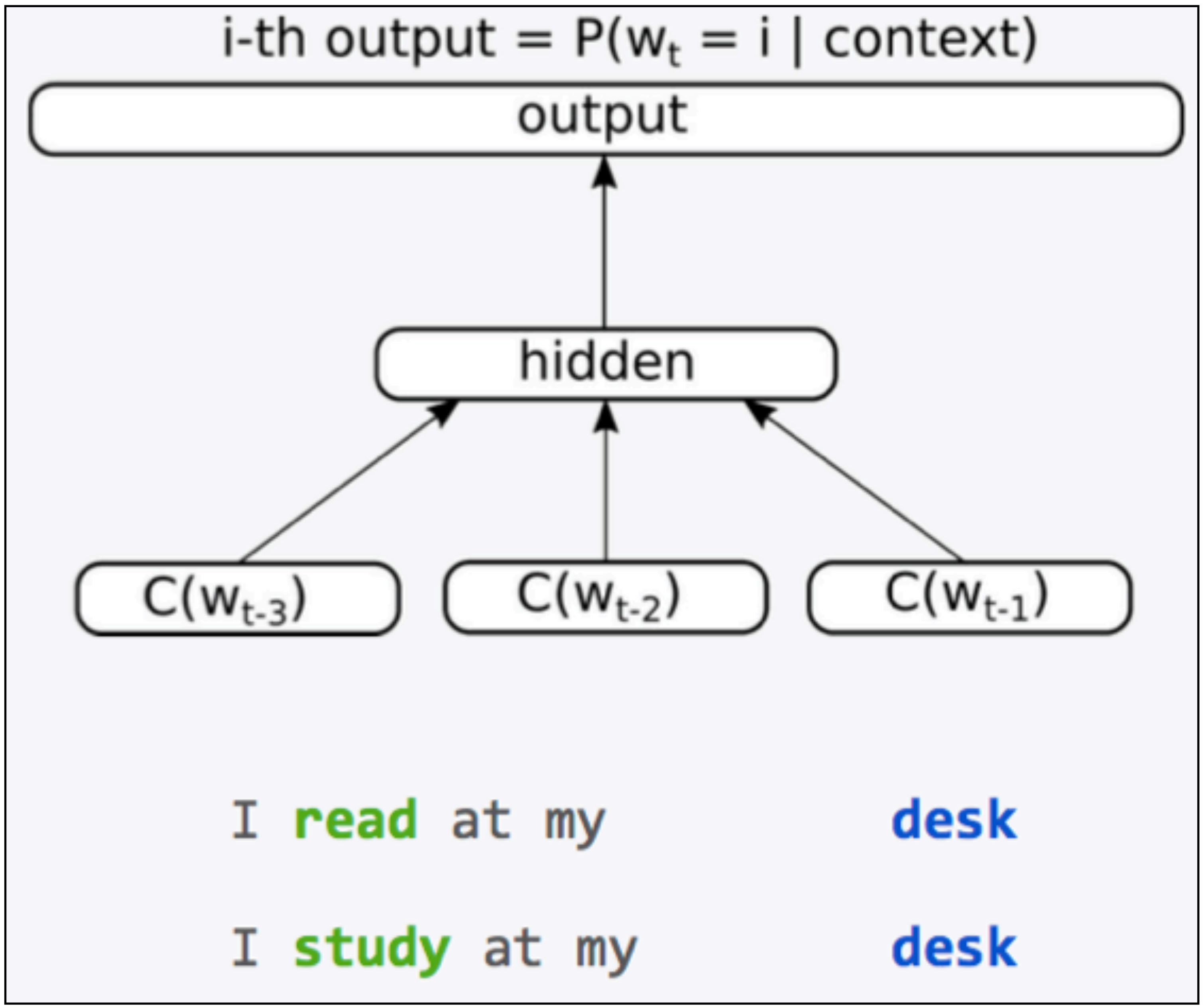
Backpropagation Through Time



- Problem: vanishing gradients

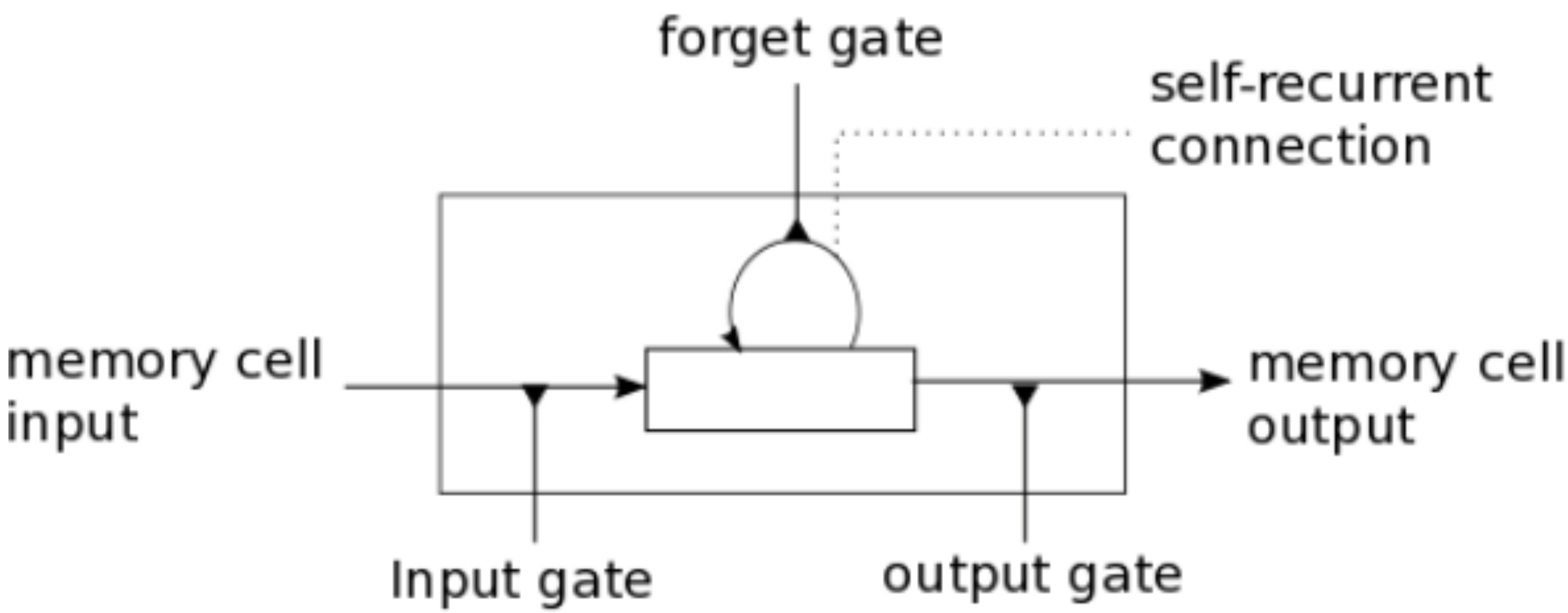
RNN Language Model

- As seen in Word Embeddings topic



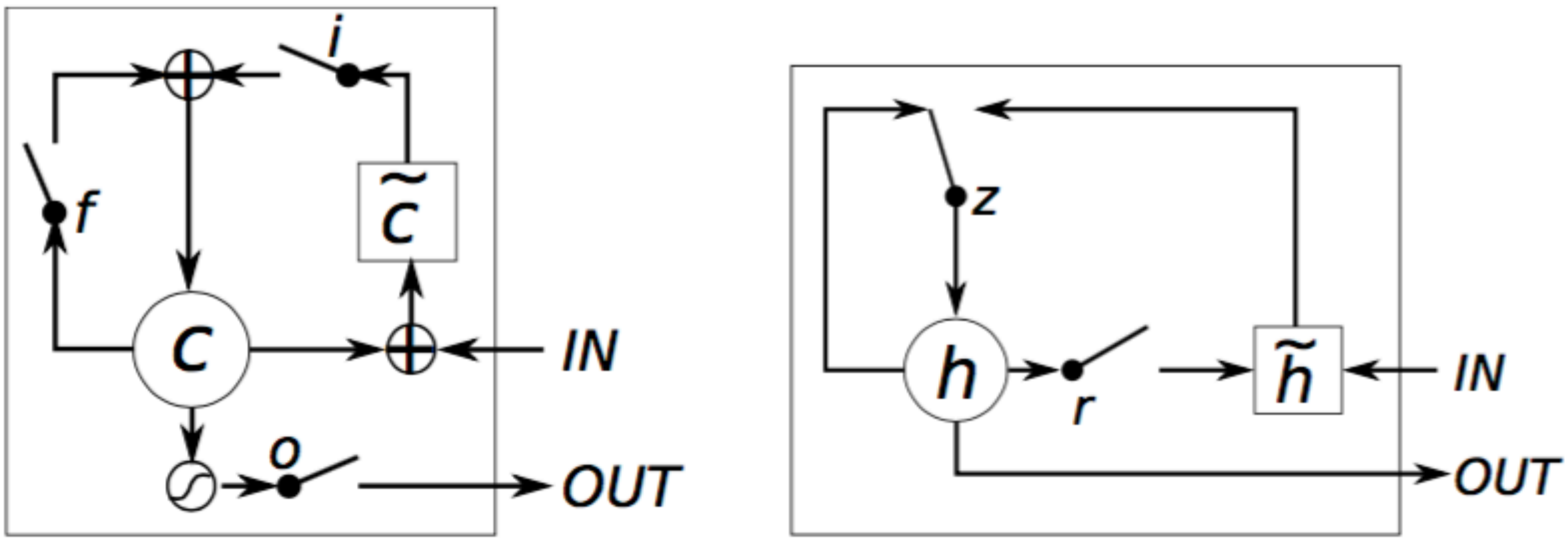
Bengio et al., 2003

Long Short Term Memory (LSTM)



- Preserves error to help with vanishing gradients
- Gate to keep or discard information
- Each cell has own set of learned weights

Gated Recurrent Unit (GRU)



(a) Long Short-Term Memory (b) Gated Recurrent Unit

Chung et al., arXiv 2014

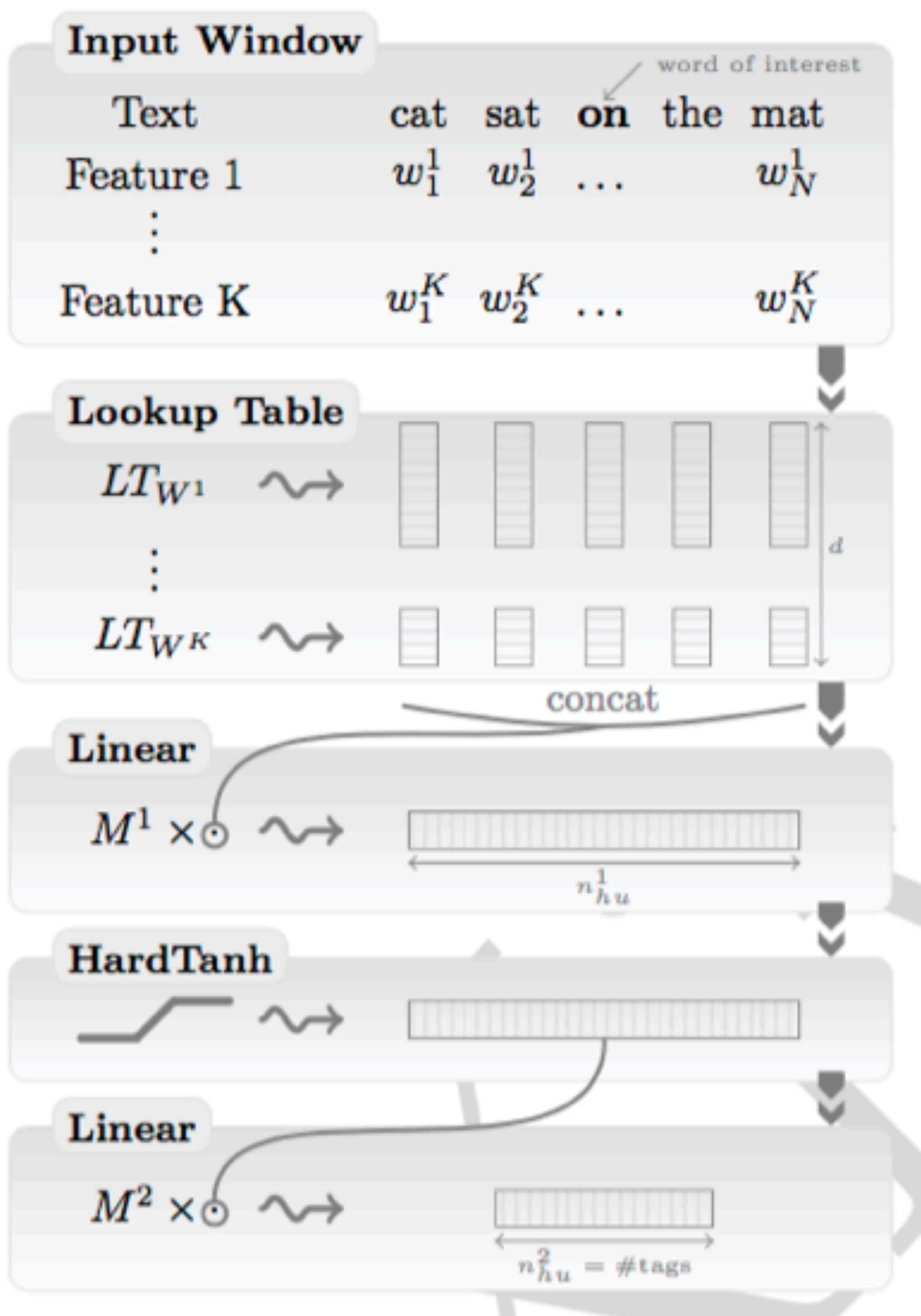
Outline

- Neural network basics
- NN architectures
  - Feedforward networks and backpropagation
  - Recursive Neural Networks
  - Recurrent Neural Networks
- **Applications**
  - **Tagging**
  - Parsing
  - Machine Translation and Encoder-Decoder Networks

Tagging

The	cat	sat	on	the	mat	.
DET	NN1	VBD	PRP	DET	NN1	PUNCT
NP/N	N	S[dcl]\NP	(S[dcl]\NP)/NP	NP/N	N	PUNCT
B_NP	I_NP	B_VP	B_PP	B_NP	I_NP	O

Tagging with Feed-Forward Network



Collobert et al., *JMLR* 2011
Lewis & Steedman, EMNLP 2014

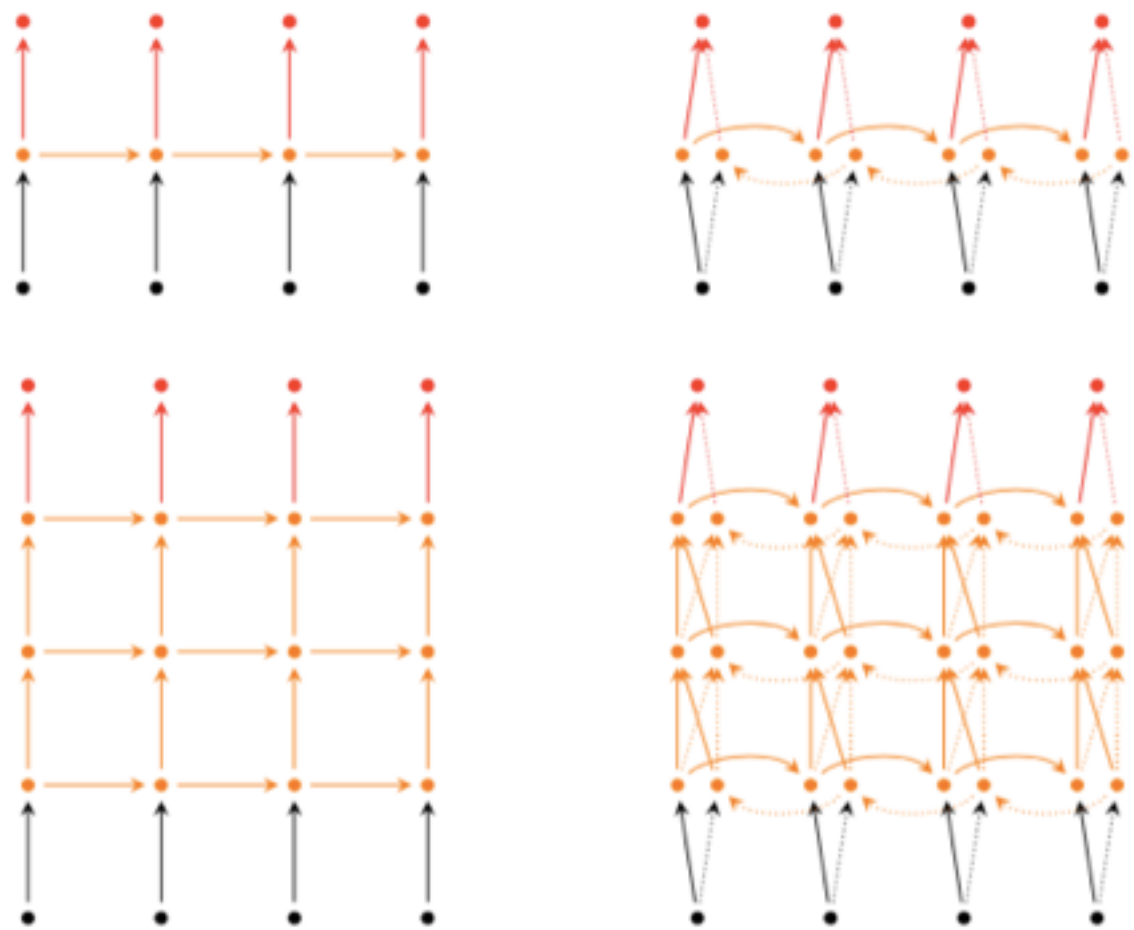
Tagging with RNN

- Task: detect and tag Direct Subjective Expressions, Expressive Subjective Expressions

The	committee	,	as	usual	,	has
O	O	O	B_ESE	I_ESE	O	B_DSE
refused	to	make	any	statements	.	
I_DSE	I_DSE	I_DSE	I_DSE	I_DSE	O	

Irsoy & Cardie, EMNLP 2014

Deep Bidirectional RNN



$$\overrightarrow{h}_t = f(\overrightarrow{W} x_t + \overrightarrow{V}\,\overrightarrow{h}_{t-1} + \overrightarrow{b})$$
$$\overleftarrow{h}_t = f(\overleftarrow{W} x_t + \overleftarrow{V}\,\overleftarrow{h}_{t+1} + \overleftarrow{b})$$
$$y_t = g(U_{\rightarrow}\overrightarrow{h}_t + U_{\leftarrow}\overleftarrow{h}_t + c)$$

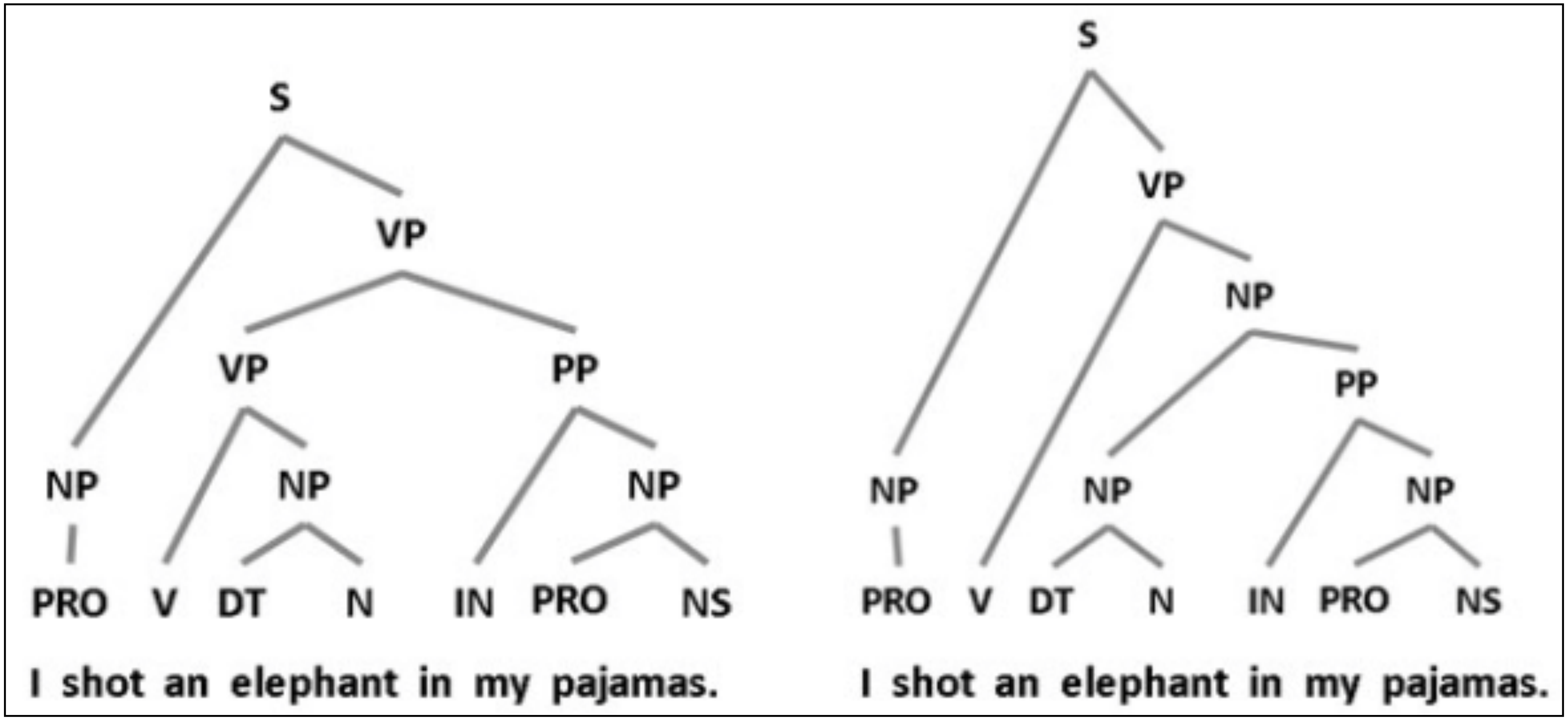
- Bidirectional RNN incorporates info from preceding and following words
- h = [$\overrightarrow{h}$; $\overleftarrow{h}$] represents past and future around a word
- Deep (stacked) RNN

Irsoy & Cardie, EMNLP 2014

Outline

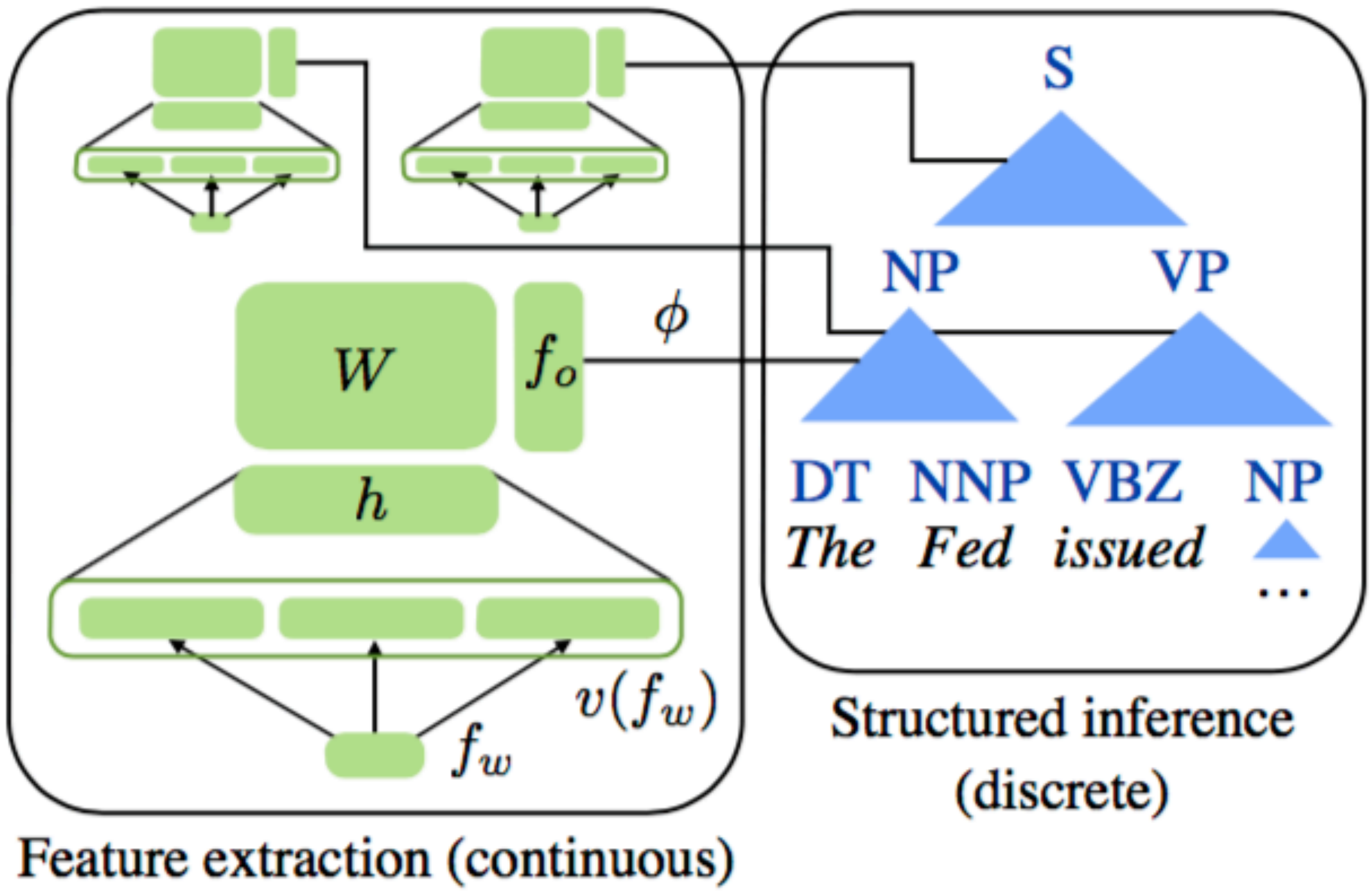
- Neural network basics
- NN architectures
  - Feedforward Networks and Backpropagation
  - Recursive Neural Networks
  - Recurrent Neural Networks
- **Applications**
  - Tagging
  - **Parsing**
  - Machine Translation
- Back to architectures for MT
  - Encoder-Decoder Networks

Constituent Parsing



Constituent Parsing with Feed-Forward NN

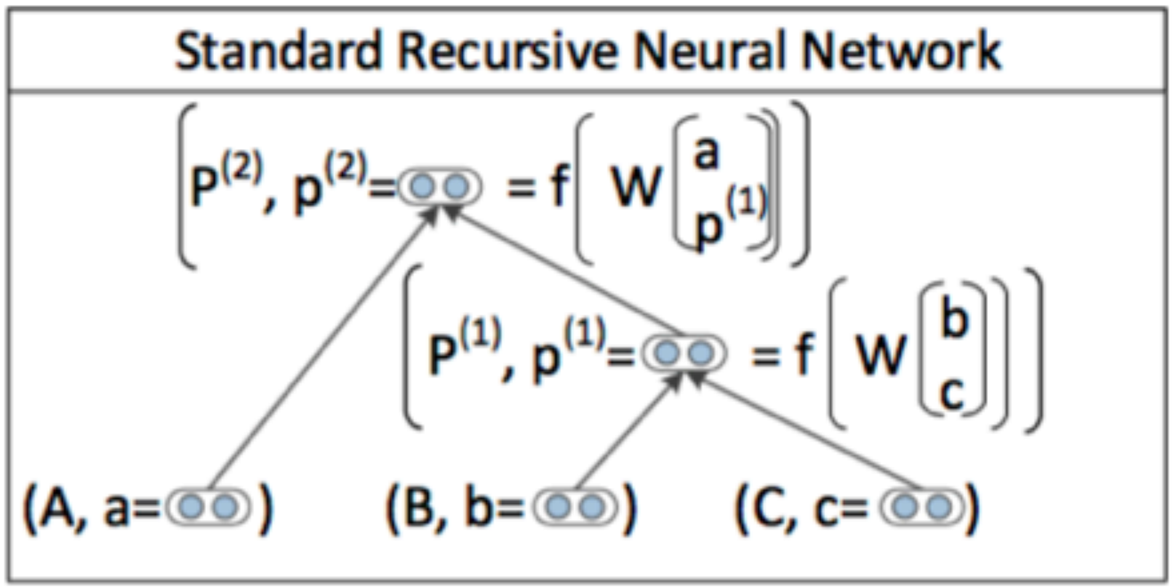
- Neural embeddings replace sparse features in CRF parser

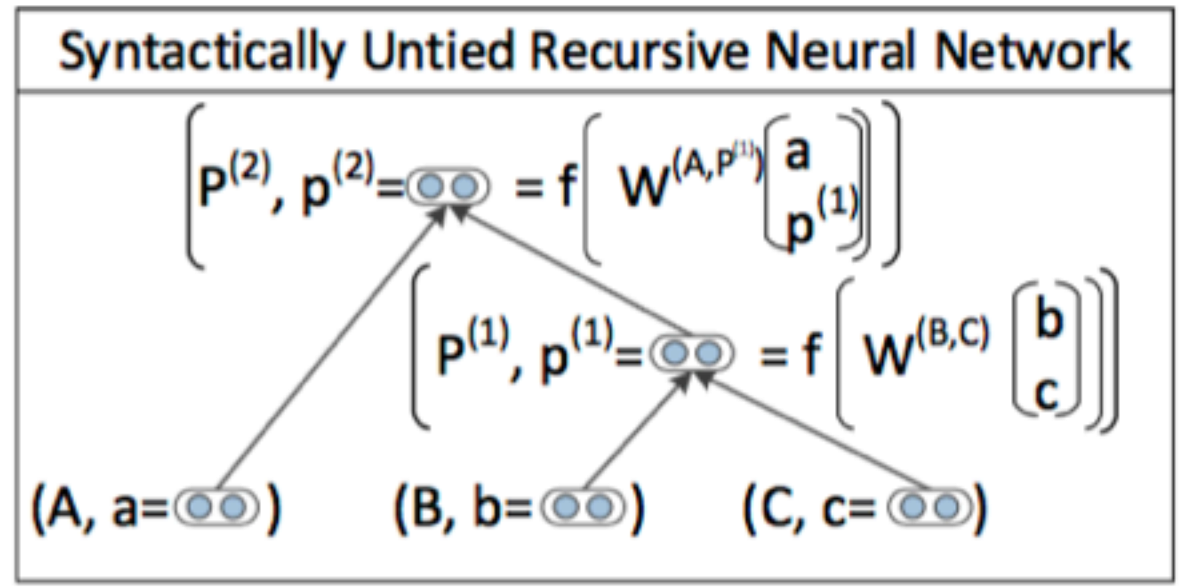


Durret & Klein, ACL 2015

Constituent Parsing with RecNN

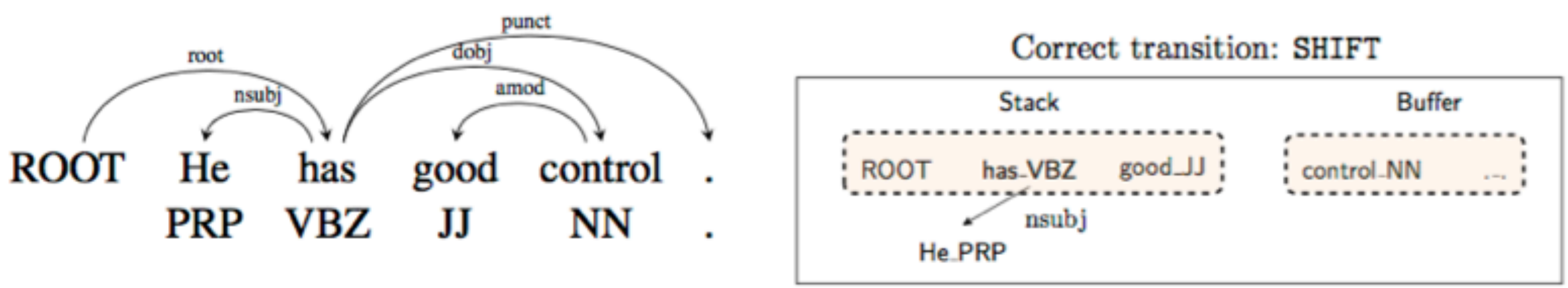
- Used for re-ranking n-best PCFG parser output





Socher et al., ACL 2013

Dependency Parsing

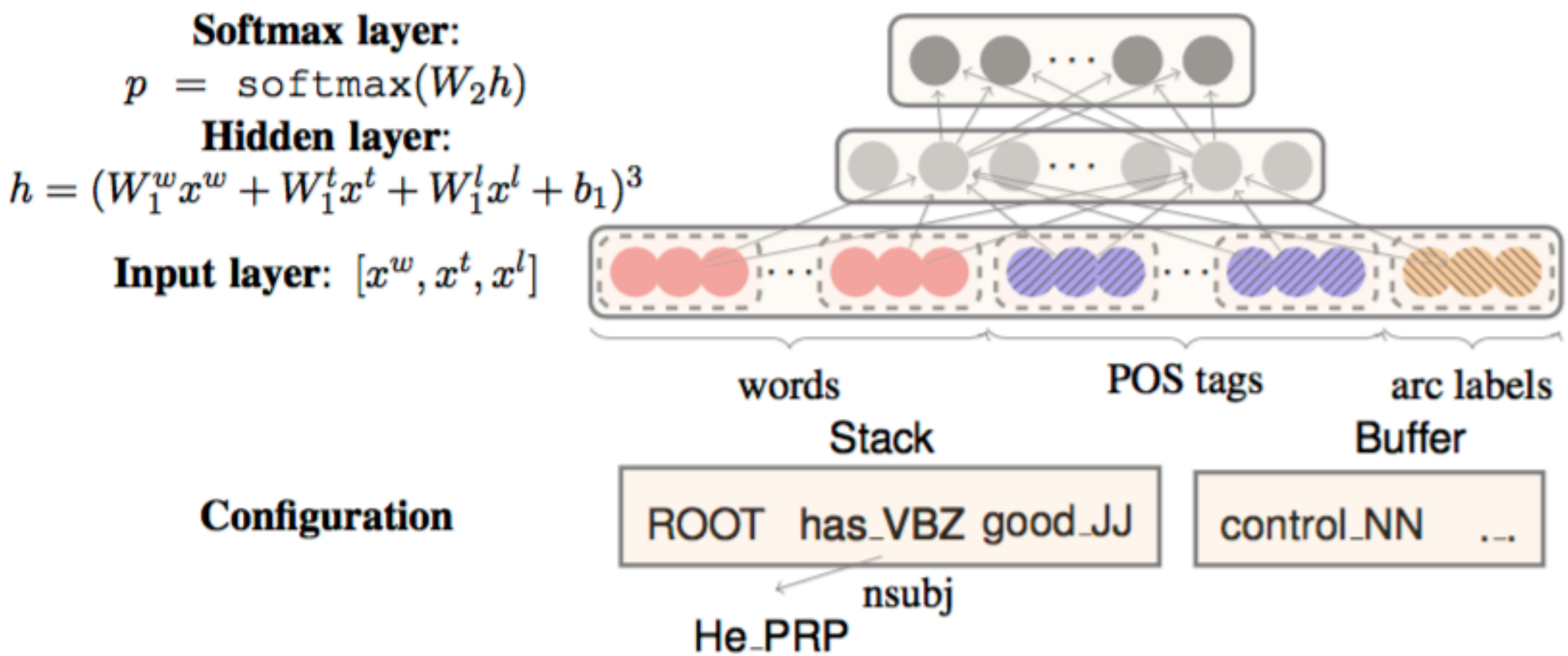


Transition	Stack	Buffer	A
	[ROOT]	[He has good control .]	$\emptyset$
SHIFT	[ROOT He]	[has good control .]	
SHIFT	[ROOT He has]	[good control .]	
LEFT-ARC(nsubj)	[ROOT has]	[good control .]	$A\cup$ nsubj(has,He)
SHIFT	[ROOT has good]	[control .]	
SHIFT	[ROOT has good control]	[.]	
LEFT-ARC(amod)	[ROOT has control]	[.]	$A\cup$amod(control,good)
RIGHT-ARC(dobj)	[ROOT has]	[.]	$A\cup$ dobj(has,control)
...	...	...	...
RIGHT-ARC(root)	[ROOT]	[]	$A\cup$ root(ROOT,has)

Figure 1: An example of transition-based dependency parsing. Above left: a desired dependency tree, above right: an intermediate configuration, bottom: a transition sequence of the arc-standard system.

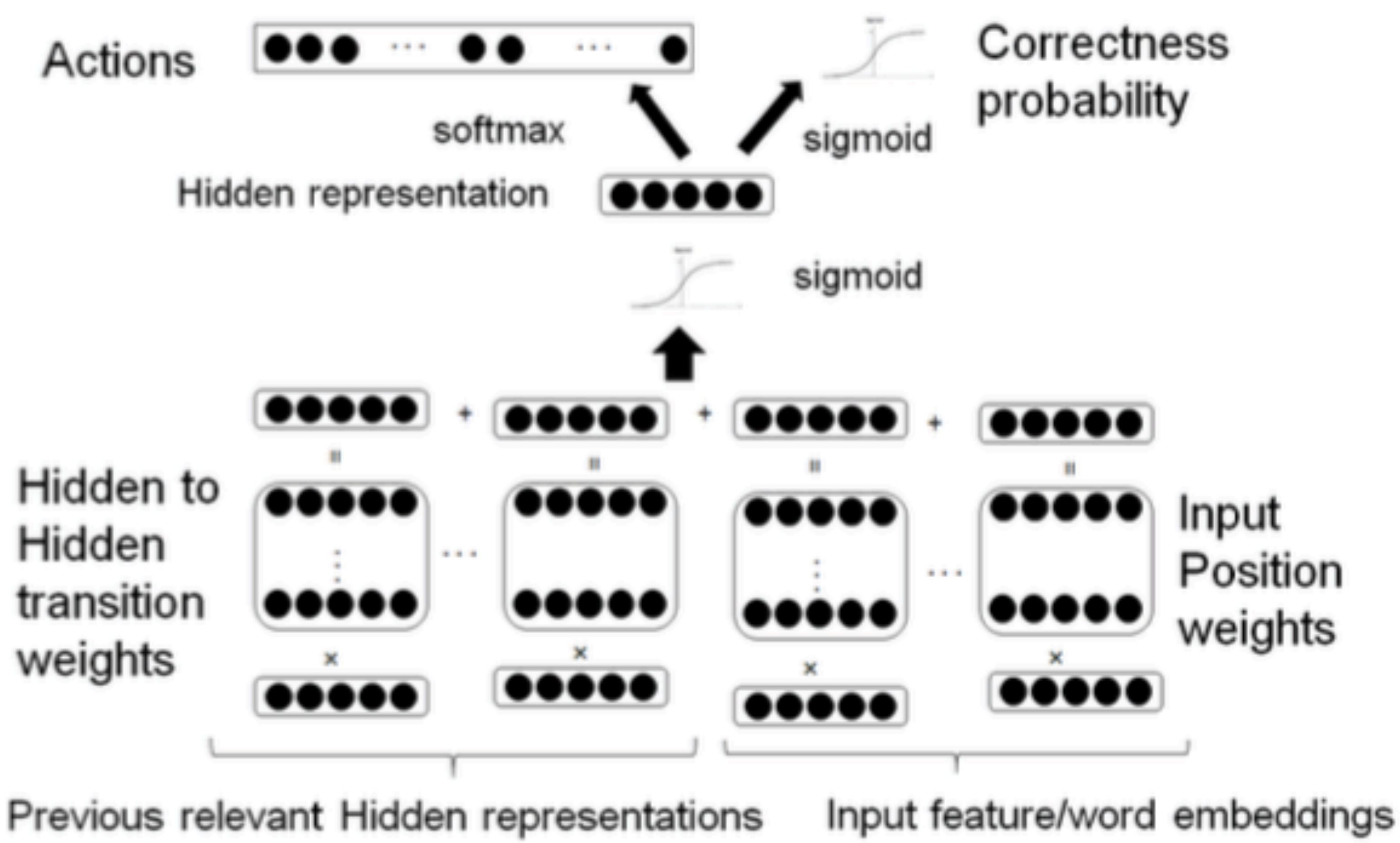
Chen & Manning, EMNLP 2014

Dependency Parsing with Feed-Forward NN



Chen & Manning, EMNLP 2014

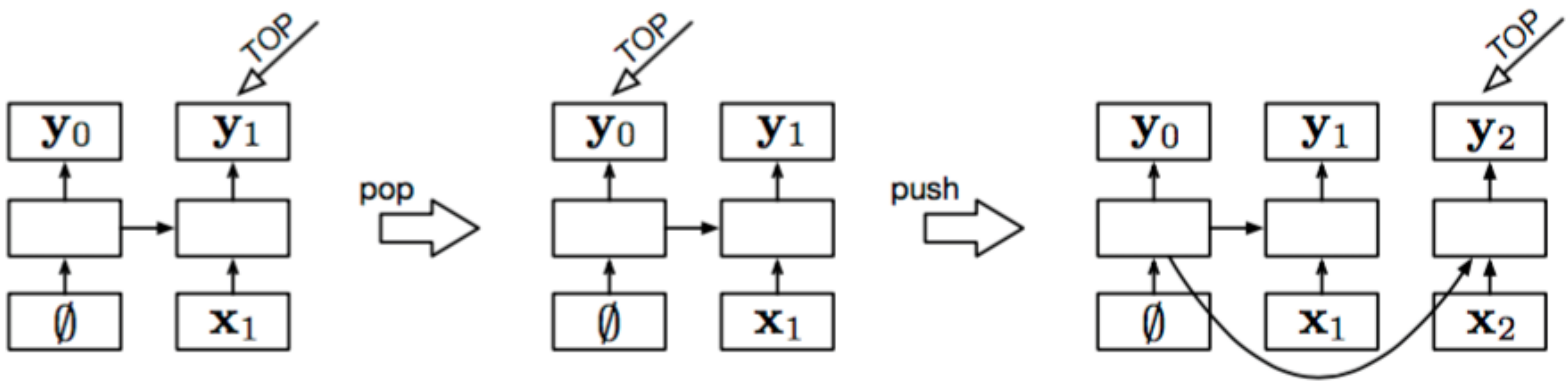
Dependency Parsing with RNN (v1)



Yazdani & Henderson, CoNLL 2015

Dependency Parsing with RNN (v2)

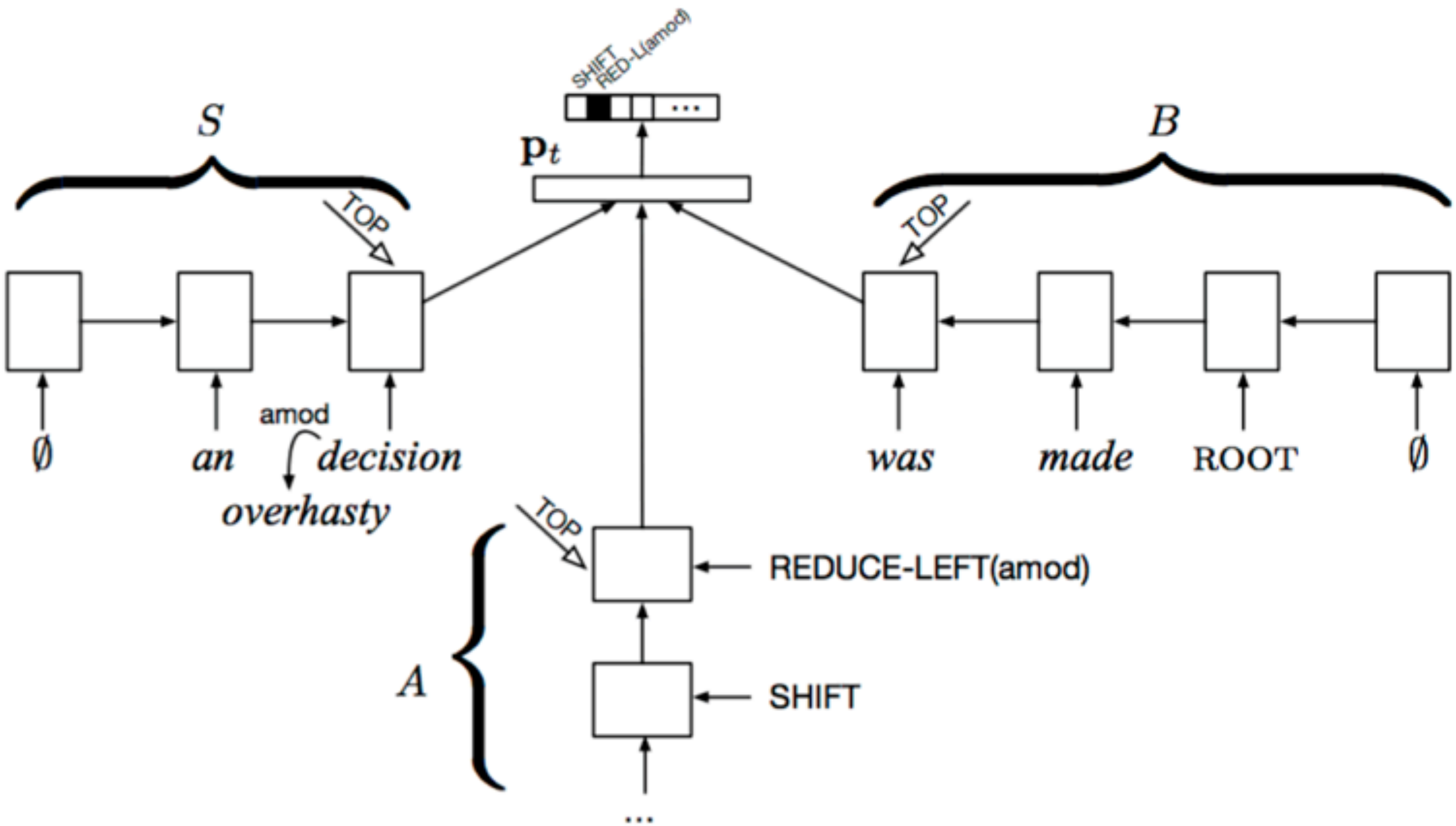
- Stack LSTM: adds stack pointer



Dyer et al., ACL 2015

Dependency Parsing with RNN (v2)

sentence: *an overhasty decision was made*

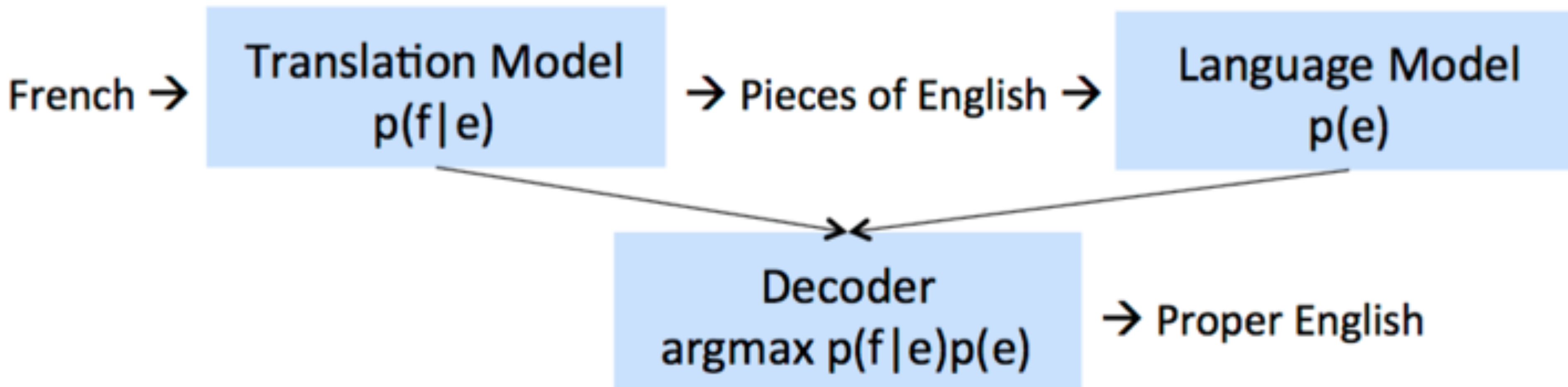


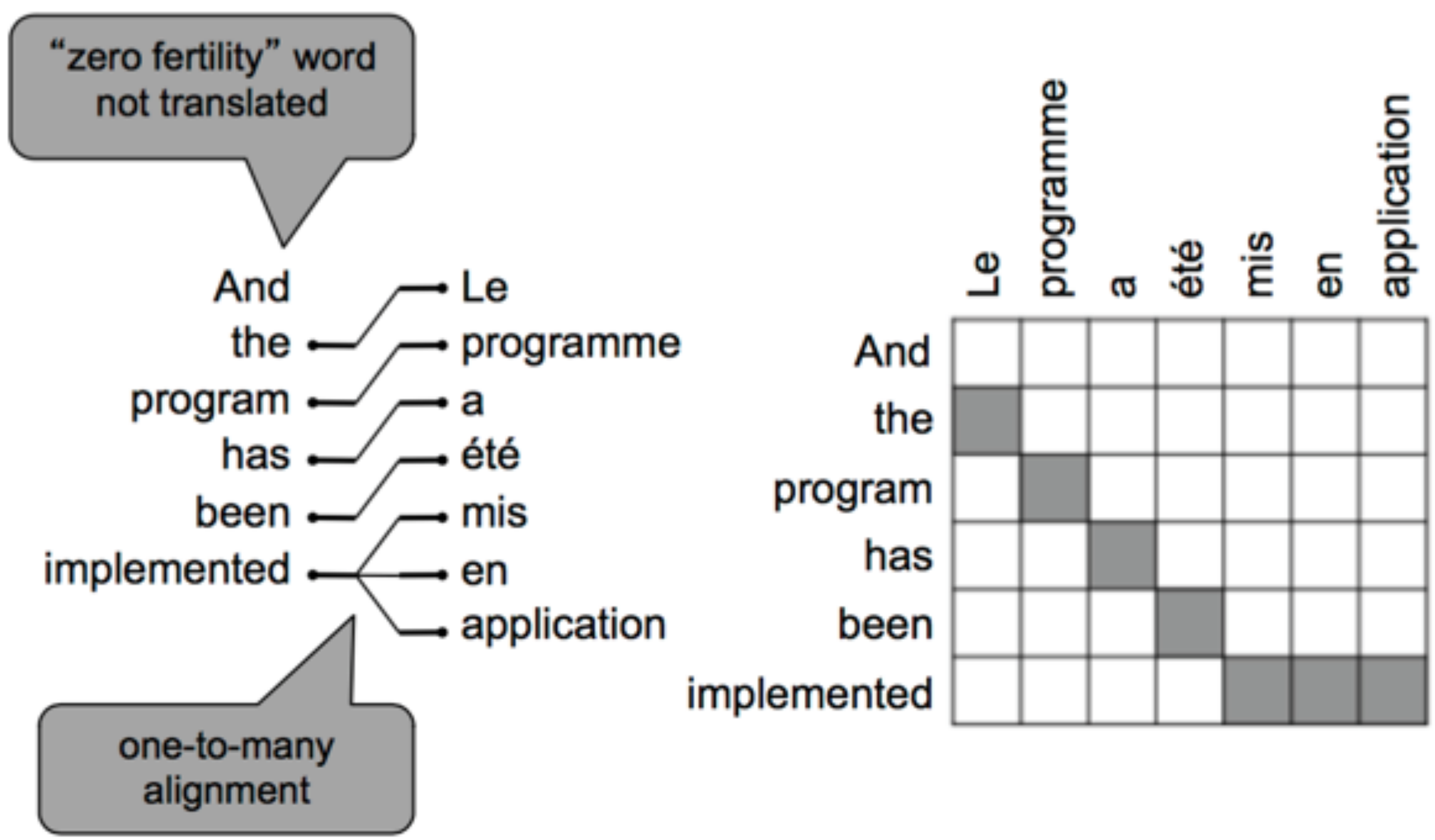
Dyer et al., ACL 2015

Outline

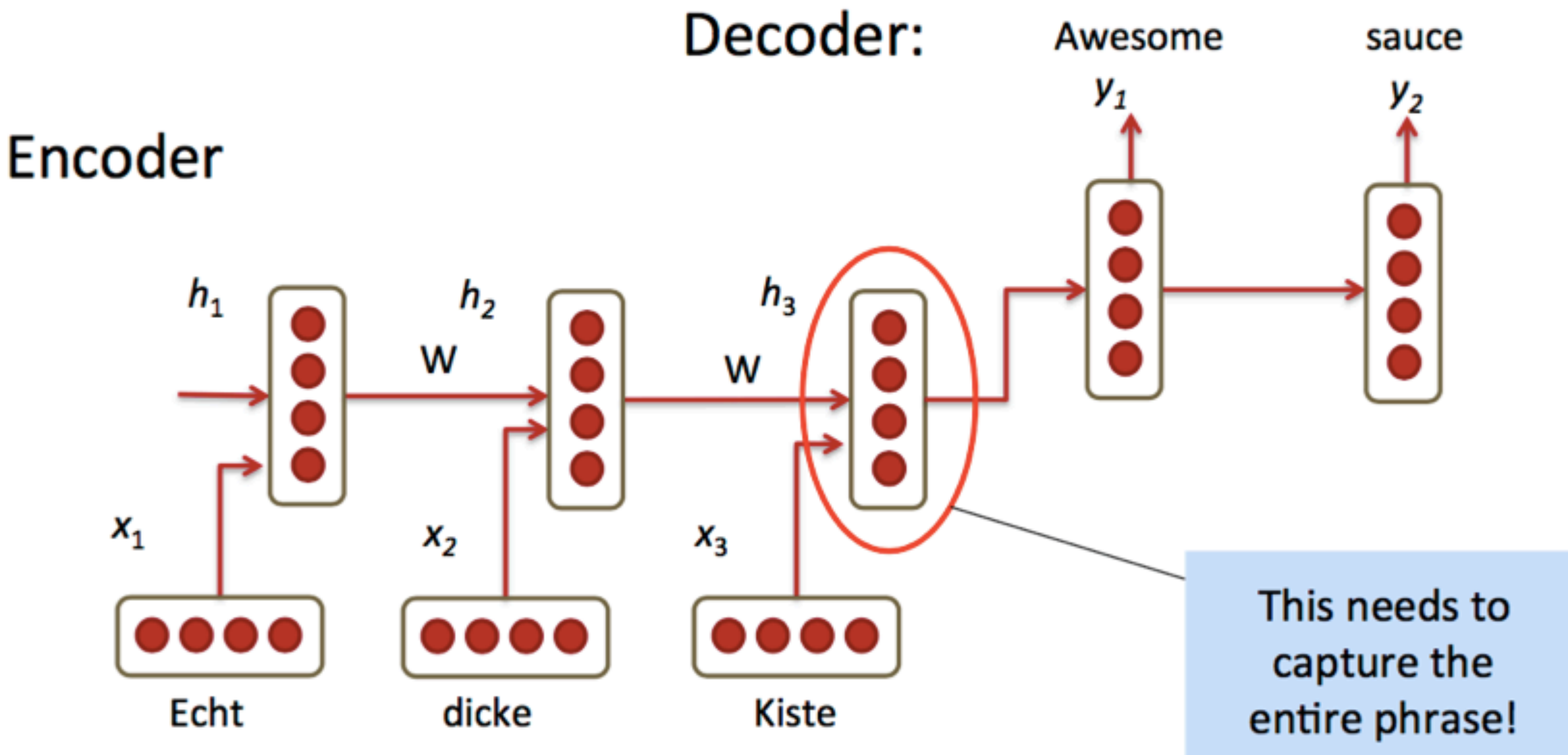
- Neural network basics
- NN architectures
  - Feedforward Networks and Backpropagation
  - Recursive Neural Networks
  - Recurrent Neural Networks
- **Applications**
  - Tagging
  - Parsing
  - **Machine Translation and Encoder-Decoder Networks**

Machine Translation





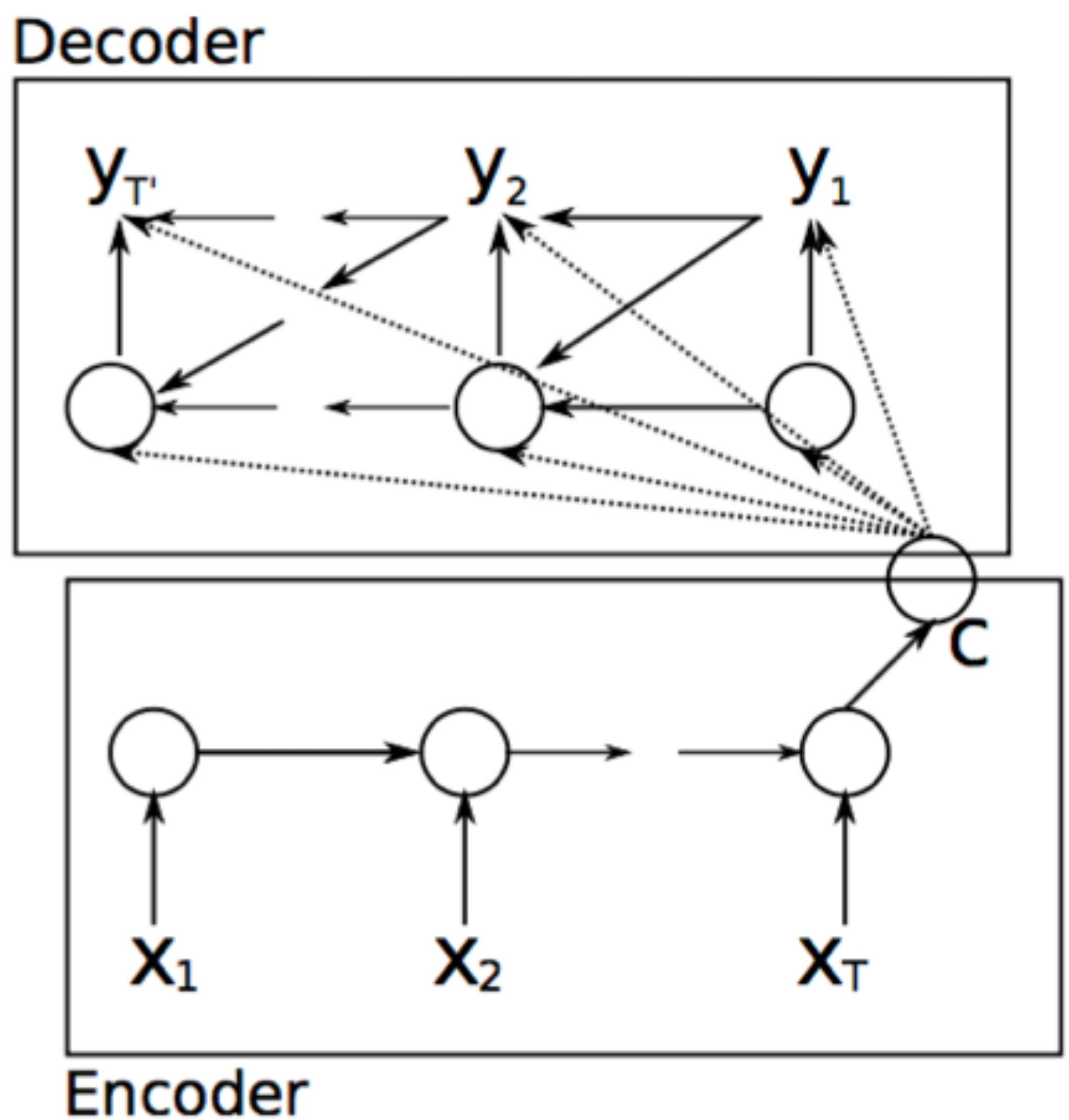
Encoder-Decoder Network (RNN) (gen'l idea)



Example from Richard Socher lecture

Encoder-Decoder Network (RNN) (v1)

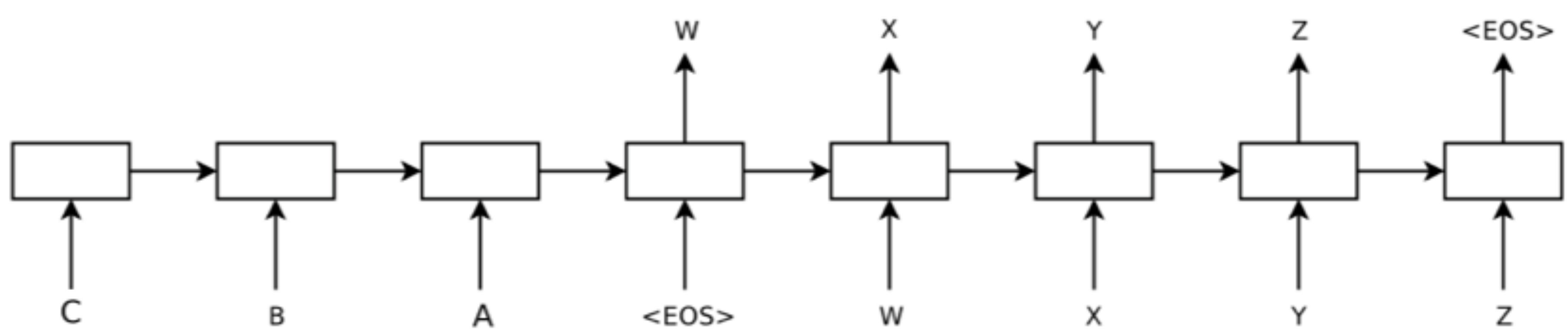
- Used for scoring phrase pairs in phrase table of standard SMT system



Cho et al., EMNLP 2014

Encoder-Decoder Network (RNN) (v2)

- Used for direct translation with beam search decoder
- 4-layer deep LSTM
- Input words in reverse order



Sutskever et al., *ANIPS* 2014

NLP Neural Network Applications