COMP90042

Workshop Week 07

- Homework 3
  - Due: Wednesday, April 18

Syllabus

1	Introduction and Preprocessing	Text classification
2	Lexical semantics	Distributional semantics
3	Part of Speech Tagging	Hidden Markov Models
4	Unsupervised Hidden Markov Models	Context-Free Grammars
5	Probabilistic Parsing	Dependency parsing
	Easter holiday break	
6	N-gram language models	Neural language models
7	Information Extraction	Question Answering
8	Topic Models	*ANZAC day holiday*
9	Information Retrieval -- Boolean search and the vector space model	Indexing and querying in the vector space model, evaluation
10	Index and vocabulary compression	Efficient query processing
11	The Web as a Graph: Page-rank & HITS	Machine Translation (word based)
12	Machine translation (phrase based) and neural encoder-decoder	Subject review

Outline

- N-gram language models
- Neural language models
- Smoothing

Language models (LMs)

- LMs can tell us the probability of a sequence of words (usually a sentence)
  - Calculate $P(This\ is\ an\ apple)$
- LMs can generate sentences
  - The model samples the next word based on probability distribution until ‘</s>’ is generated
  - Because the model defines $P(current\ word|previous\ words)$

Two types of language models

- Note: Others probably don't call them Type I & II…
- Type I
  - $\sum_{all\ sents} P(sent) \neq 1$, but $\sum_{len(sent)=L} P(sent) = 1$ for any L
  - Example: most commonly used unigram LMs
- Type II
  - $\sum_{all\ sents} P(sent) = 1$
  - Strictly following the principle (probabilities sum up to 1)

Type I example

- sent1 = a b c a b c
- sent2 = c b a b c
- sent3 = b a b a

- Unigram counts

- {a: 5, b: 6, c: 4}

- Bigram counts

		current		
		a	b	c
previous	<s>	1	1	1
	a	0	4	0
	b	3	0	3
	c	1	1	0

Type I models

- sent1 = a b c a b c
- sent2 = c b a b c
- sent3 = b a b a

- Unigram counts

- {a: 5, b: 6, c: 4}

- Bigram counts

		current		
		a	b	c
previous	<s>	1	1	1
	a	0	4	0
	b	3	0	3
	c	1	1	0

- Unigram model (type I)
- $P(a) = \frac{5}{15} = \frac{1}{3}$
- $P(b) = \frac{6}{15} = \frac{2}{5}$
- $P(c) = \frac{4}{15}$

- Bigram model (type I)

		current		
		a	b	c
previous	<s>	1/3	1/3	1/3
	a	0	1	0
	b	1/2	0	1/2
	c	1/2	1/2	0

Type I models

- Unigram model
- $P(a, b, c) = P(a)P(b)P(c)$
- Bigram model
- $P(a, b, c) = P(a|<s>)P(b|a)P(c|b)$

- Unigram model (type I)
- $P(a) = \frac{5}{15} = \frac{1}{3}$
- $P(b) = \frac{6}{15} = \frac{2}{5}$
- $P(c) = \frac{4}{15}$
- Bigram model (type I)

		current		
		a	b	c
previous	<s>	1/3	1/3	1/3
	a	0	1	0
	b	1/2	0	1/2
	c	1/2	1/2	0

Type II example

- sent1 = a b c a b c </s>
- sent2 = c b a b c </s>
- sent3 = b a b a </s>

- Unigram counts

- {a: 5, b: 6, c: 4, </s>: 3}

- Bigram counts

		current			
		a	b	c	</s>
previous	<s>	1	1	1	0
	a	0	4	0	1
	b	3	0	3	0
	c	1	1	0	2

Type II models

- sent1 = a b c a b c </s>
- sent2 = c b a b c </s>
- sent3 = b a b a </s>
- Unigram counts
- {a: 5, b: 6, c: 4, </s>: 3}
- Bigram counts

		current			
		a	b	c	</s>
previous	<s>	1	1	1	0
	a	0	4	0	1
	b	3	0	3	0
	c	1	1	0	2

- Unigram model (type II)
- $P(a) = \frac{5}{18}, P(b) = \frac{6}{18} = \frac{1}{3}$
- $P(c) = \frac{4}{18} = \frac{2}{9}$
- $P(</s>) = \frac{3}{18} = \frac{1}{6}$
- Bigram model (type II)

		current			
		a	b	c	</s>
previous	<s>	1/3	1/3	1/3	0
	a	0	4/5	0	1/5
	b	1/2	0	1/2	0
	c	1/4	1/4	0	1/2

Type II models

- Unigram model

$$P(a,b,c) = P(a)P(b)P(c)P(</s>)$$

- Bigram model

$$P(a,b,c) = P(a|<s>)\,P(b|a)\,P(c|b)\,P(</s>|c)$$

- Unigram model (type II)
- $P(a) = \frac{5}{18}$, $P(b) = \frac{6}{18} = \frac{1}{3}$
- $P(c) = \frac{4}{18} = \frac{2}{9}$
- $P(</s>) = \frac{3}{18} = \frac{1}{6}$

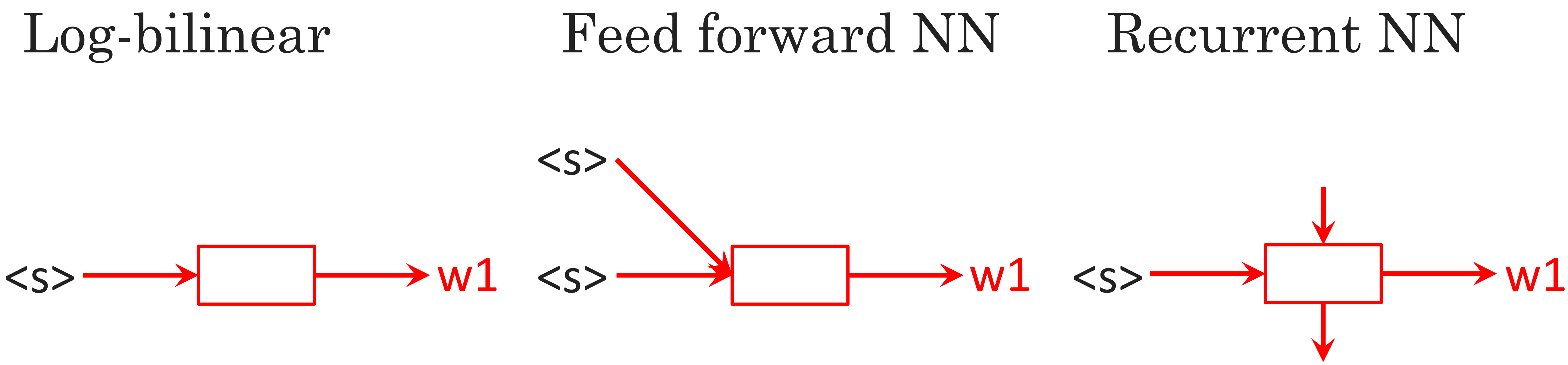
- Bigram model (type II)

		current			
		a	b	c	</s>
previous	<s>	1/3	1/3	1/3	0
	a	0	4/5	0	1/5
	b	1/2	0	1/2	0
	c	1/4	1/4	0	1/2

Outline

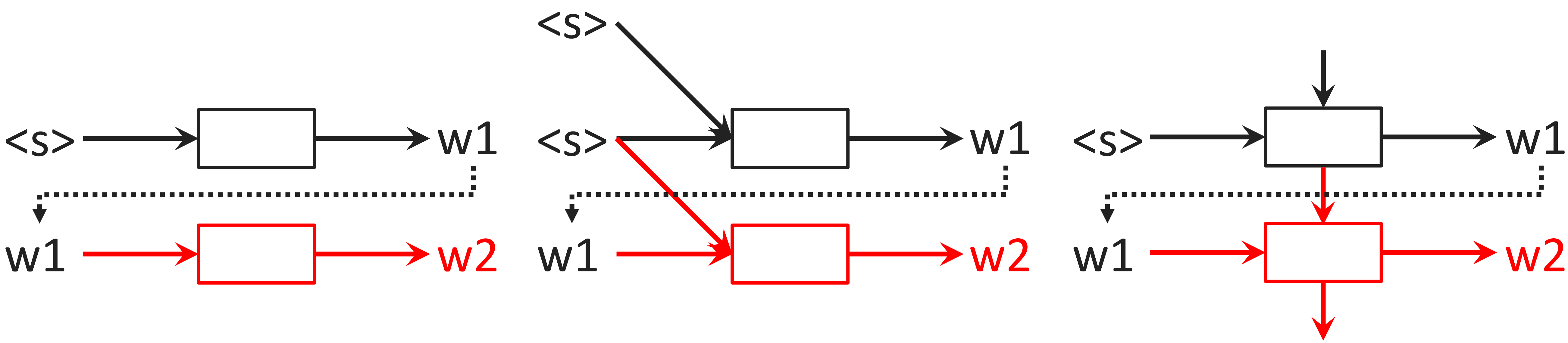
- N-gram language models
- Neural language models
- Smoothing

To generate a sentence



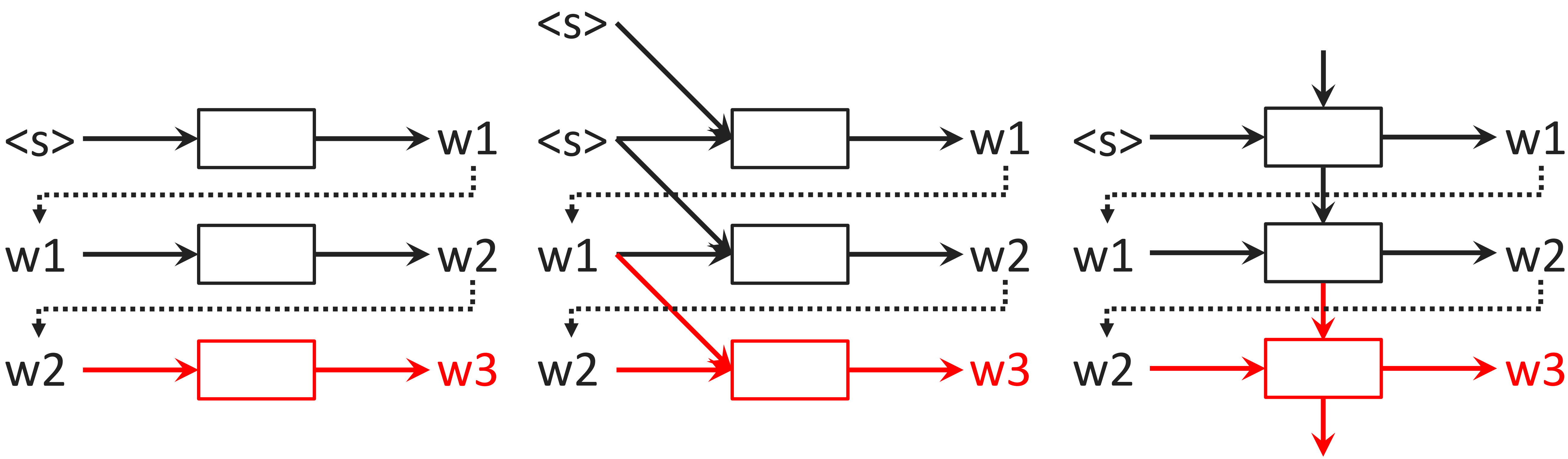
To generate a sentence

Log-bilinear | Feed forward NN | Recurrent NN



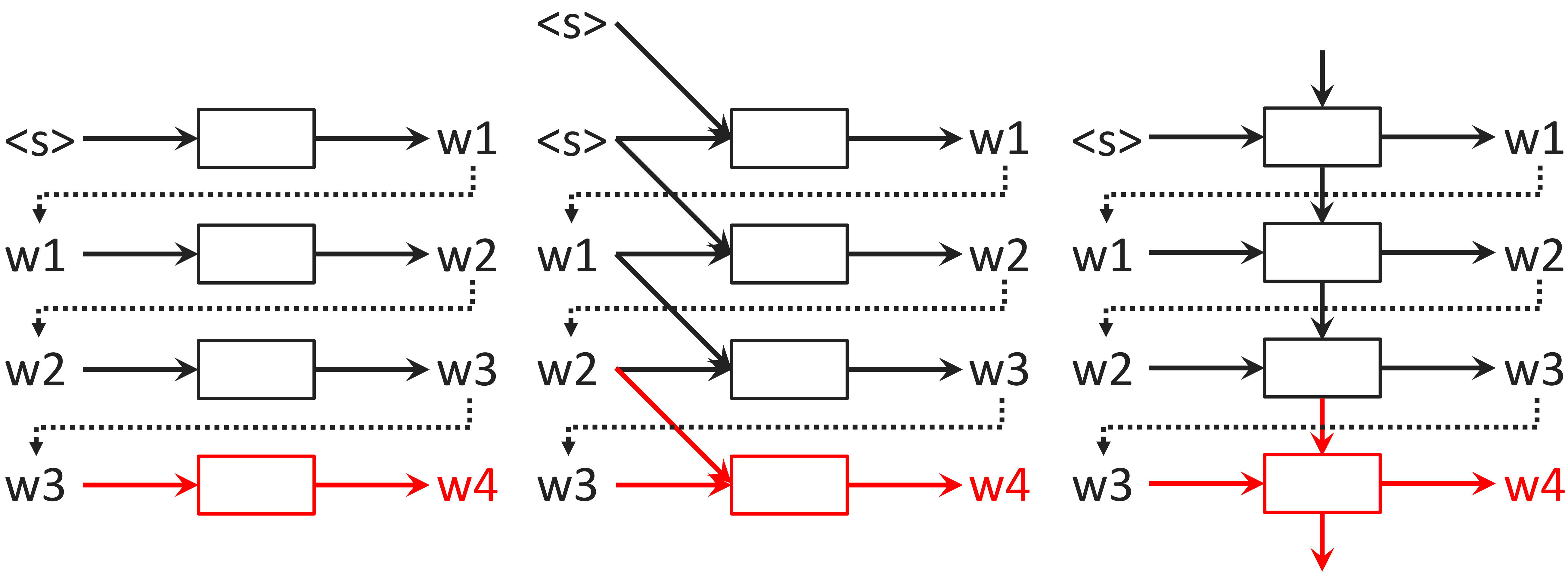
To generate a sentence

Log-bilinear | Feed forward NN | Recurrent NN



To generate a sentence

Log-bilinear | Feed forward NN | Recurrent NN



Softmax

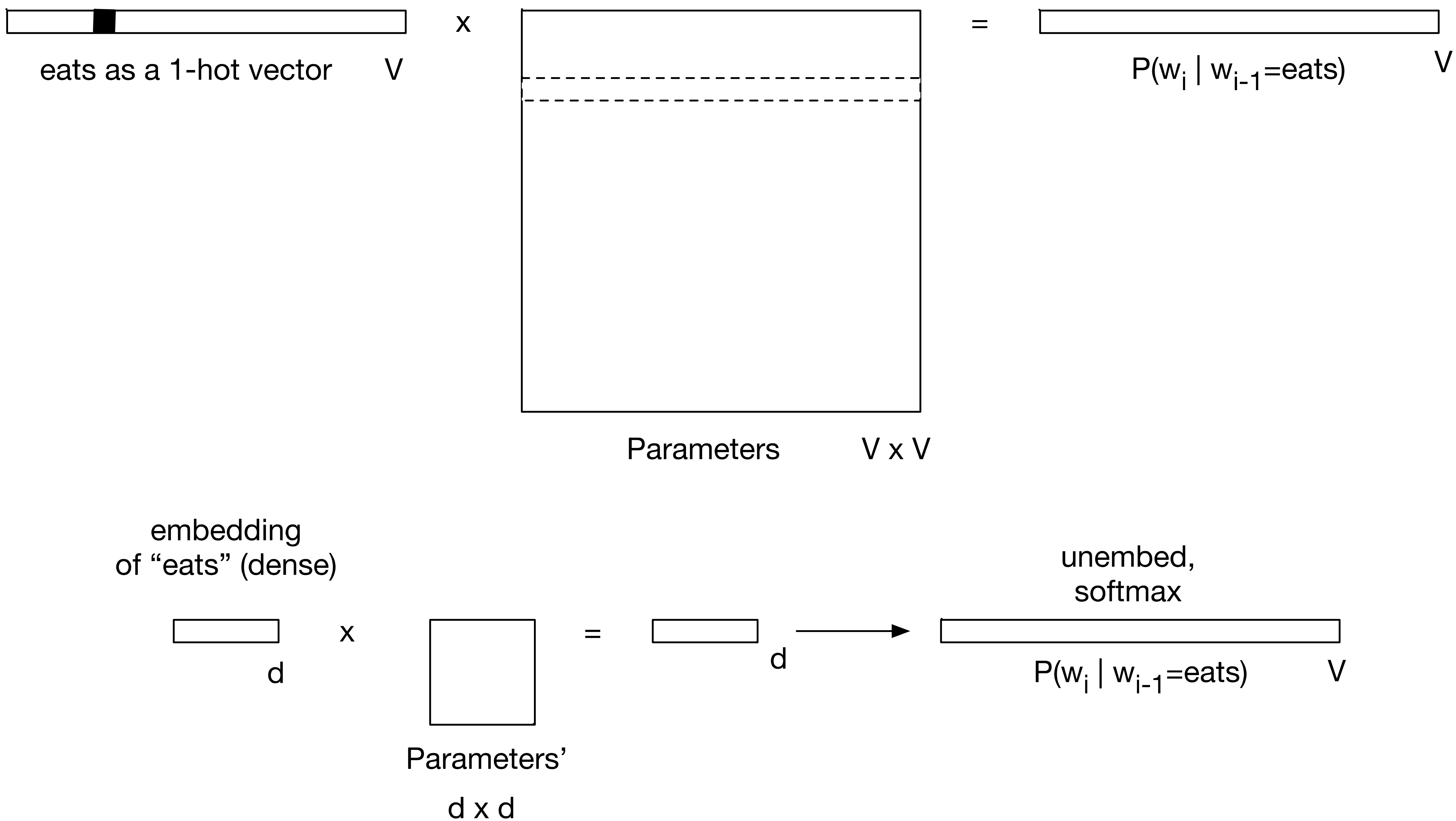
- Convert an array to a valid probability distribution
  - Values are all positive and sum up to 1

- $a = [2, 3, 5]$
- $\text{softmax}(a) = \left[\frac{e^2}{e^2+e^3+e^5}, \frac{e^3}{e^2+e^3+e^5}, \frac{e^5}{e^2+e^3+e^5}\right] \approx [0.04, 0.11, 0.84]$

Log-Bilinear LM

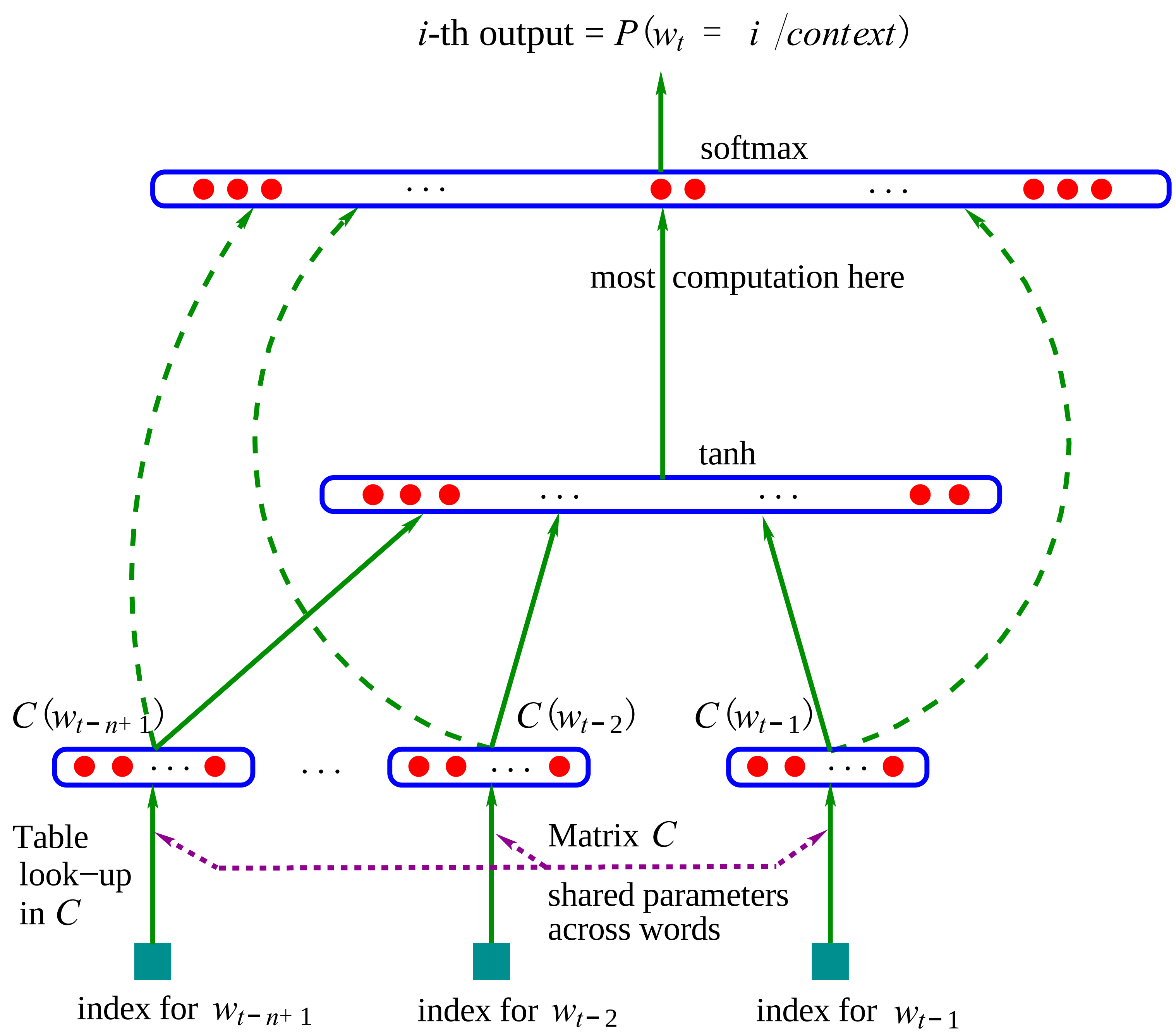
- Parameters:
  - embedding matrix (or 2 matrices, for input & output) of size V x d
  - weights now d x d
- If $d \ll V$ then considerable saving vs V^2
- d chosen as parameter typically in [100, 1000]
- Model is called the *log-bilinear LM*, and is closely related to *word2vec*

Can’t we use word embeddings?



FFNNLM

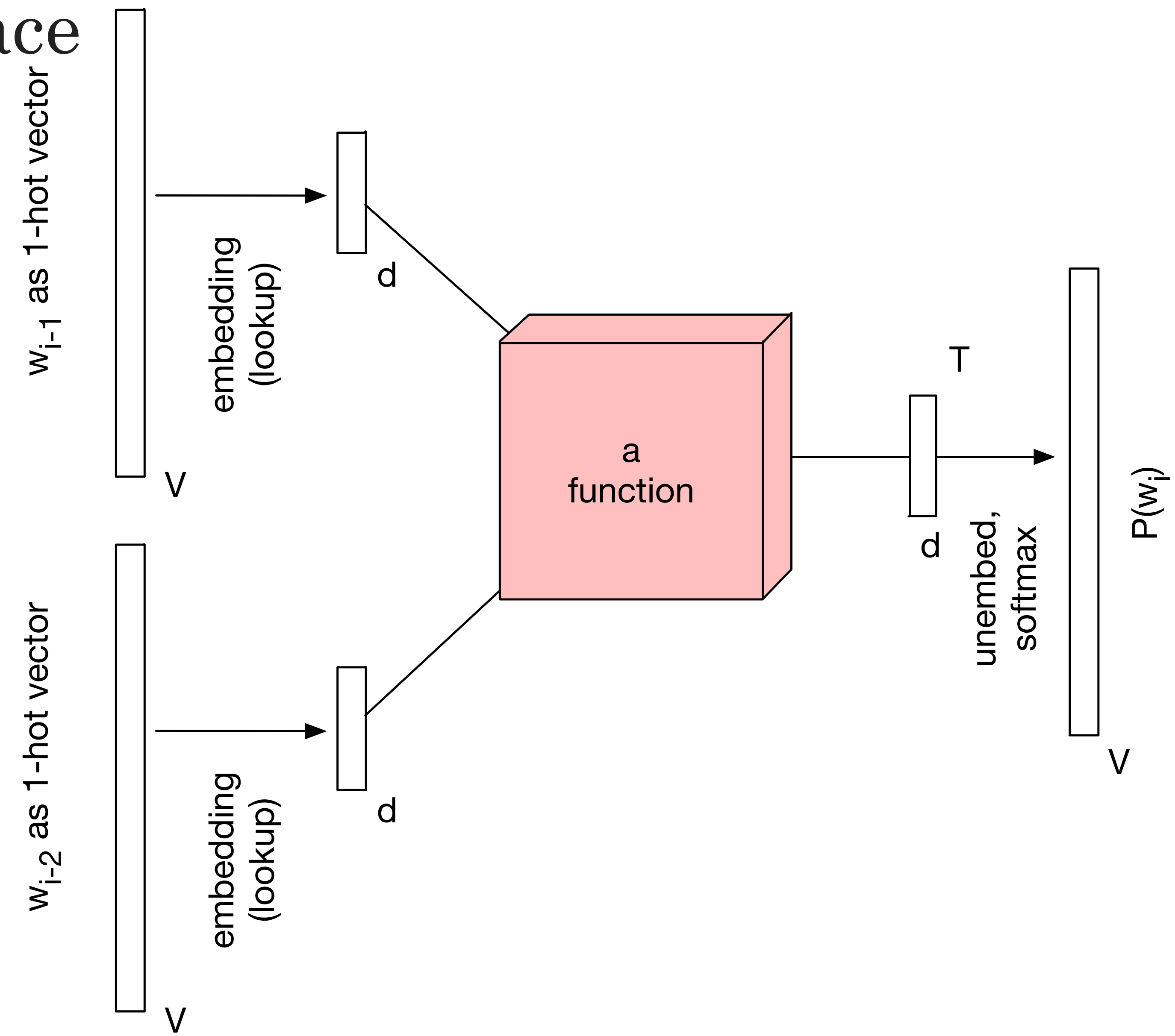
- Application to language modelling



Bengio et al, 2003

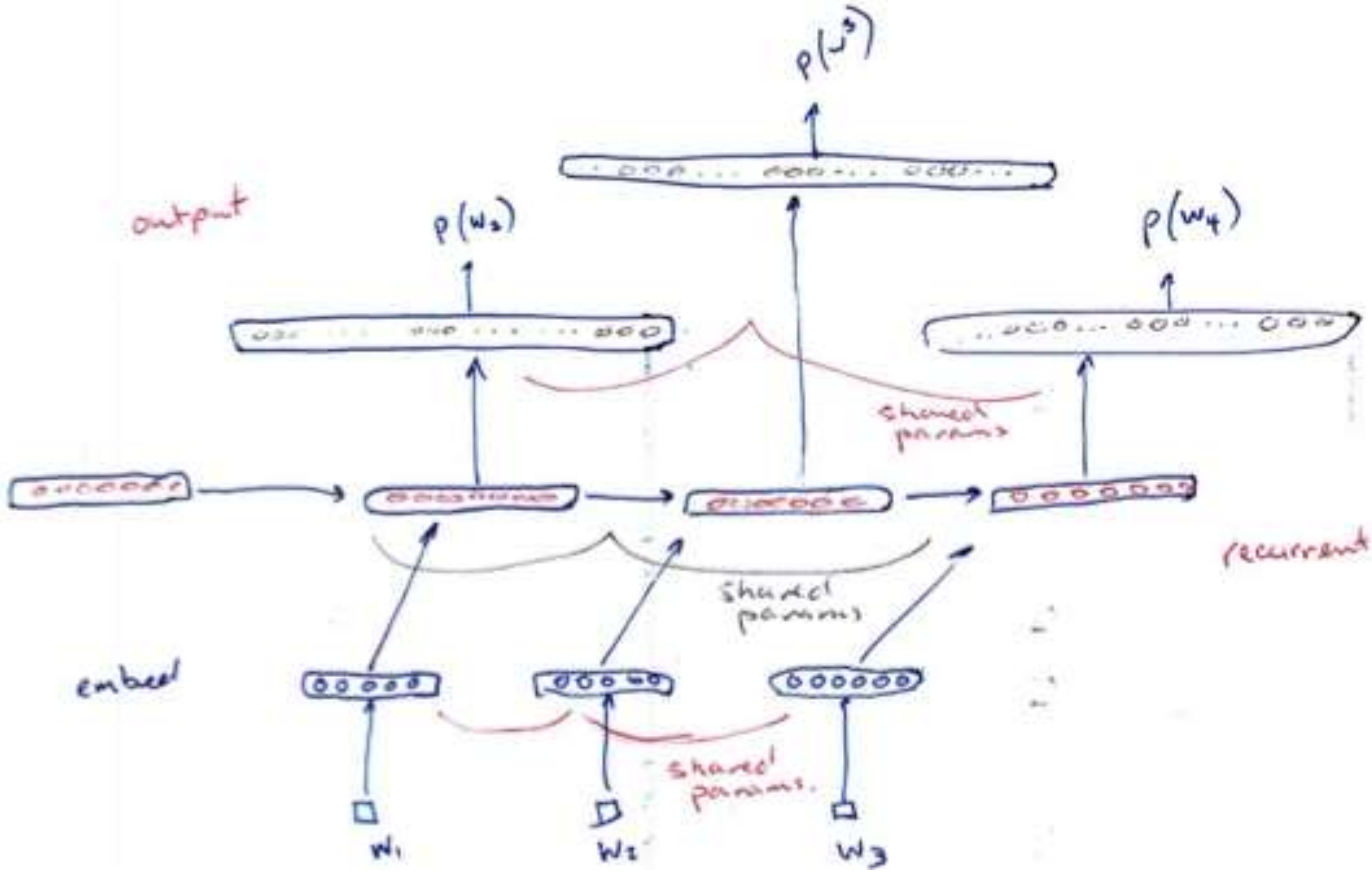
Feed forward neural net LMs

- Neural networks more general approach, based on same principle
  - embed input context words
  - transform in “hidden” space
  - un-embed to prob over full vocab
- Neural network used to define transformations
  - e.g., feed forward LM (FFLM)

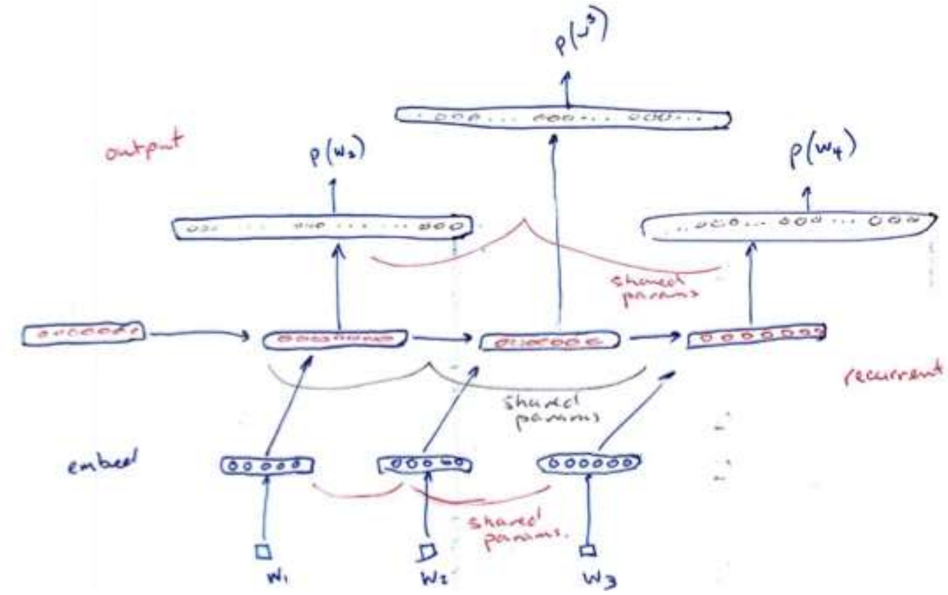


Recurrent NNLMS

- What if we structure the network differently, e.g., according to sequence with Recurrent Neural Networks (RNNs)



Recurrent NNLMS



- Start with
  - initial hidden state $\boldsymbol{h}_0$
- For each word, w_i, in order $i=1..m$
  - embed word to produce vector, $\boldsymbol{e}_i$
  - compute hidden $\boldsymbol{h}_i = \tanh(W\,\boldsymbol{e}_i + V\,\boldsymbol{h}_{i-1} + \boldsymbol{b})$
  - compute output $\mathrm{P}(\boldsymbol{w}_{i+1}) = \mathrm{softmax}(U\,\boldsymbol{h}_i + \boldsymbol{c})$
- Train such to minimise $\sum_i -\log \mathrm{P}(\boldsymbol{w}_i)$
  - to learn parameters W, V, U, $\boldsymbol{b}$, $\boldsymbol{c}$, $\boldsymbol{h}_0$

Outline

- N-gram language models
- Neural language models
- Smoothing

Laplacian (Add-one) smoothing

- Simple idea: pretend we've seen each *n*-gram once more than we did.

For unigram models (**V**= the vocabulary),

$$P_{add1}(w_i) = \frac{C(w_i) + 1}{M + |\boldsymbol{V}|}$$

For bigram models,

$$P_{add1}(w_i|w_{i-1}) = \frac{C(w_{i-1}, w_i) + 1}{C(w_{i-1}) + |\boldsymbol{V}|}$$

For trigram models generally,

$$P_{add1}(w_i|w_{i-1}, w_{i-2}) = \frac{C(w_{i-2}, w_{i-1}, w_i) + 1}{C(w_{i-2}, w_{i-1}) + |\boldsymbol{V}|}$$

Laplacian smoothing

- Bigram counts & probs

		current			
		a	b	c	</s>
previous	<s>	1	1	1	0
	a	$0+\epsilon$	$4+\epsilon$	$0+\epsilon$	$1+\epsilon$
	b	3	0	3	0
	c	1	1	0	2

		current			
		a	b	c	</s>
previous	<s>	1/3	1/3	1/3	0
	a	$\frac{0+\epsilon}{5+4\epsilon}$	$\frac{4+\epsilon}{5+4\epsilon}$	$\frac{0+\epsilon}{5+4\epsilon}$	$\frac{1+\epsilon}{5+4\epsilon}$
	b	1/2	0	1/2	0
	c	1/4	1/4	0	1/2

- ϵ is usually very small

- Could use different ϵ's to different rows

Backoff and Interpolation

- Smooth using lower-order probabilities (less context)
- Backoff: fall back to n-1-gram counts only when n-gram counts are zero

$$P_{BO}(w_i|w_{i-2},w_{i-1}) =$$

$$P^*(w_i|w_{i-2},w_{i-1}) \quad if\ C(w_{i-2},w_{i-1},w_i) > 0$$

$$\alpha(w_{i-2},w_{i-1}) * P_{BO}(w_i|w_{i-1}) \quad otherwise$$

P* and α must preserve "sum to 1" property.

Backoff

- Bigram probabilities

P(a|a)	P(b|a)	P(c|a)	P(</s>|a)
0	4/5	0	1/5

- Unigram probabilities

P(a)	P(b)	P(c)	P(</s>)
5/18	1/3	2/9	1/6

- Going to use p(a) and p(c) for p(a|a) and p(c|a)

- But need to sum up to 1

- Unigram model (type II)

- $P(a) = \frac{5}{18}, P(b) = \frac{6}{18} = \frac{1}{3}$

- $P(c) = \frac{4}{18} = \frac{2}{9}$

- $P(</s>) = \frac{3}{18} = \frac{1}{6}$

- Bigram model (type II)

		current			
		a	b	c	</s>
previous	<s>	1/3	1/3	1/3	0
	a	0	4/5	0	1/5
	b	1/2	0	1/2	0
	c	1/4	1/4	0	1/2

Backoff

- Bigram probabilities

$P(a \mid a)$	$P(b \mid a)$	$P(c \mid a)$	$P(</s> \mid a)$
0	4/5	0	1/5

- Unigram probabilities

P(a)	P(b)	P(c)	P(</s>)
5/18	1/3	2/9	1/6

- Backoff probabilities

$P(a \mid a)$	$P(b \mid a)$	$P(c \mid a)$	$P(</s> \mid a)$
$\mu 5/18$	$\lambda 4/5$	$\mu 2/9$	$\lambda 1/5$

- Choose μ and λ so that

$$\mu \frac{5}{18} + \lambda \frac{4}{5} + \mu \frac{2}{9} + \lambda \frac{1}{5} = 1$$

- Unigram model (type II)
- $P(a) = \frac{5}{18}$, $P(b) = \frac{6}{18} = \frac{1}{3}$
- $P(c) = \frac{4}{18} = \frac{2}{9}$
- $P(</s>) = \frac{3}{18} = \frac{1}{6}$
- Bigram model (type II)

		current			
		a	b	c	</s>
previous	<s>	1/3	1/3	1/3	0
	a	0	4/5	0	1/5
	b	1/2	0	1/2	0
	c	1/4	1/4	0	1/2

Backoff and Interpolation

- Interpolation involves taking a linear combination of all relevant probabilities
- Defined recursively:

$$P_{interp}(w_i|w_{i-2},w_{i-1}) = \lambda(w_{i-2},w_{i-1})\,P(w_i|w_{i-2},w_{i-1}) + (1 - \lambda(w_{i-2},w_{i-1}))P_{interp}(w_i|w_{i-1})$$

- Interpolation of probabilities preserves "sum to 1" property
- λs can be constant across all contexts
  - But better if sensitive to n-grams
- Parameters need to be trained on held out data

Interpolation

- Bigram probabilities

P(a|a)	P(b|a)	P(c|a)	P(</s>|a)
0	4/5	0	1/5

- Unigram probabilities

P(a)	P(b)	P(c)	P(</s>)
5/18	1/3	2/9	1/6

- Interpolation probabilities

P(a|a)	P(b|a)	P(c|a)	P(</s>|a)
λ 0 + (1-λ)5/18	λ 4/5 + (1-λ)1/3	λ 0 + (1-λ)2/9	λ 1/5 + (1-λ)1/6