Word embedding

(Some slides come from Geoffrey Hinton’s lectures)

Outline

- Motivation
- What is word embedding?
- A semantic task
- Bias in word embedding
- Neural LM

N-gram LM

- Given a sentence $w_1\, w_2 \ldots w_n$, how to estimate $P(w_1 \ldots w_n)$?

- The Markov independence assumption:
  $P(w_n \mid w_1, \ldots, w_{n-1})$ depends only on the previous k words.

- $P(w_1 \ldots w_n)$
  $= P(w_1) * P(w_2|w_1) * \ldots P(w_n \mid w_1, \ldots, w_{n-1})$
  $\approx P(w_1) * P(w_2|w_1) * \ldots P(w_n \mid w_{n-k+1}, \ldots, w_{n-1})$

- 0^{th} order Markov model: unigram model
- 1^{st} order Markov model: bigram model
- 2^{nd} order Markov model: trigram model
- …

Limitation of n-gram LM

- It does not understand the similarities between words:
  - Ex: cat/dog, garden/yard, Friday/Monday, Seattle/LA, king/queen

  ➔ Represent each word as a feature vector

- We cannot use a bigger context (i.e., large n) because there are too many parameters to store and most ngrams will be unseen.

  ➔ Represent the context as a vector

- Sentences have structures.

- …

What is word embedding?

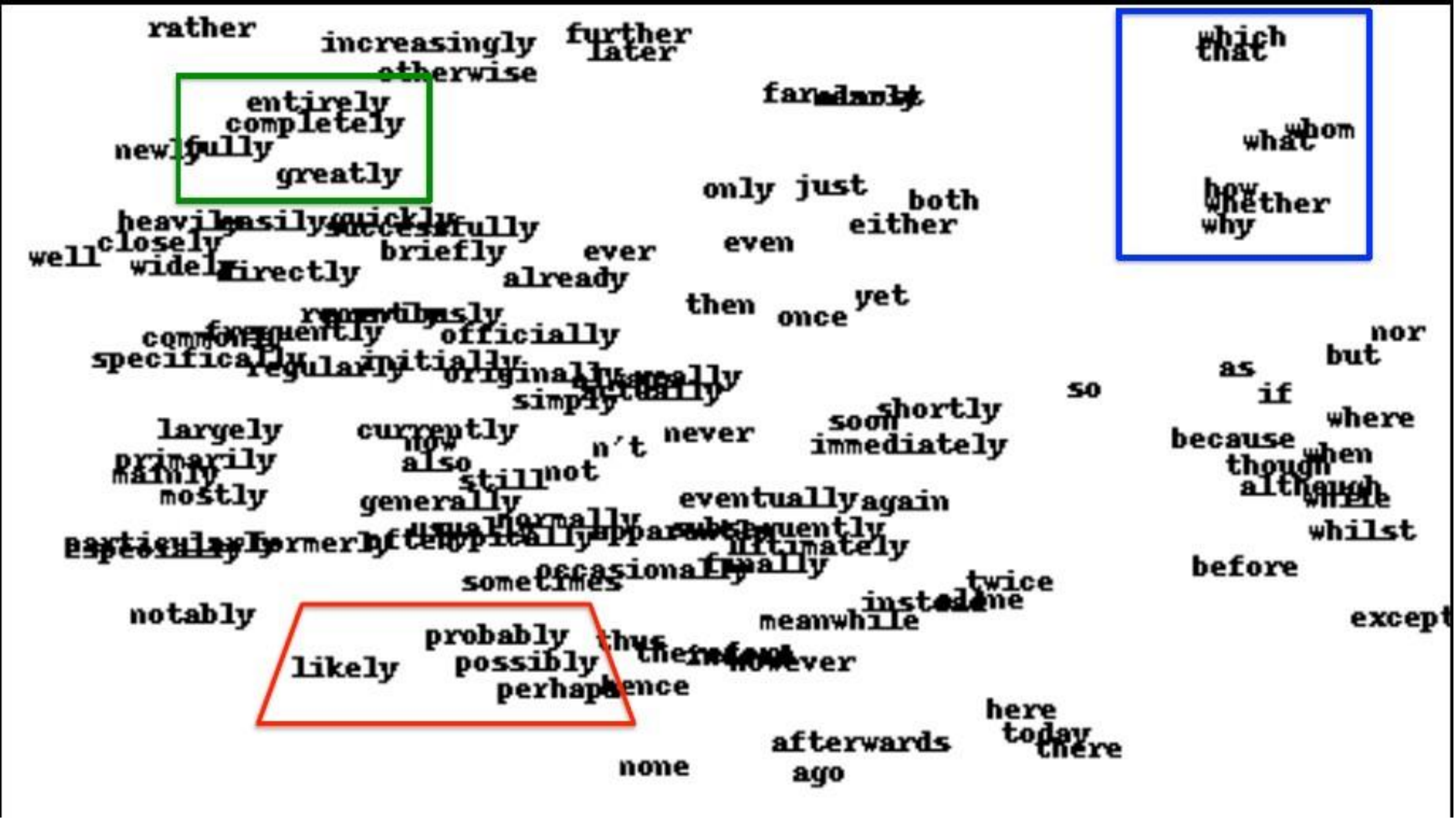
- Represent a word as a d-dimensional vector:
  - Vectors created by a system such as word2vec

- Similar words should have similar embeddings.

- Recent work on embedding of phrases, sentences, documents, etc.

Part of a 2-D map of the 2500 most common words


virginia
columbia
missouri
maryland
colorado
tennessee
wisconsin
oregon
carolina
houston
pennsylvania
georgia
hollywood
toronto
massachusetts
sydney
montreal
manchester
london
victoria
quebec
paris
moscow
mexico
scotland
england
ireland
canada
britain
sweden
singapore
america
europe
austria
asia
africa
russia
india
japan
korea
rome
egypt
israel



rather
increasingly
further
later
otherwise
which
that
entirely
completely
newly
fully
greatly
whom
what
only
just
both
how
whether
why
either
even
ever
closely
well
briefly
already
then
once
yet
officially
specifically
regularly
simply
so
as
if
but
nor
where
because
though
when
whilst
shortly
soon
immediately
never
n't
largely
also
primarily
mostly
not
generally
eventually
again
ultimately
occasionally
sometimes
twice
before
notably
meanwhile
except
probably
possibly
likely
however
here
afterwards
none
ago

Many studies on neural network

- Early studies: (Hinton 1986), (Pollack 1990), (Elman 1991), etc.
- Feed-forward networks: (Bengio et al., 2003; 2006)
- Recurrent neural networks: (Mikolov et al., 2010; 2011; 2013)
- Now: tons of papers in 2014-now

Where is the word embedding used?

- Answer semantic questions: e.g., A:B is like C:D
- LM
- Classification
- Sequence labeling: POS tagging, chunking, NER, etc.
- Structure prediction: parsing
- Question answering
- …

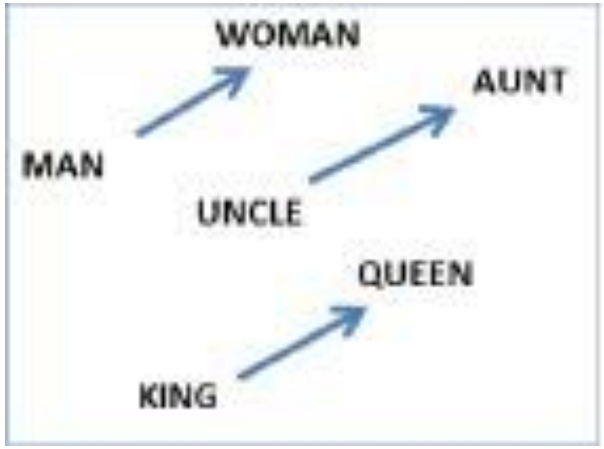
Outline

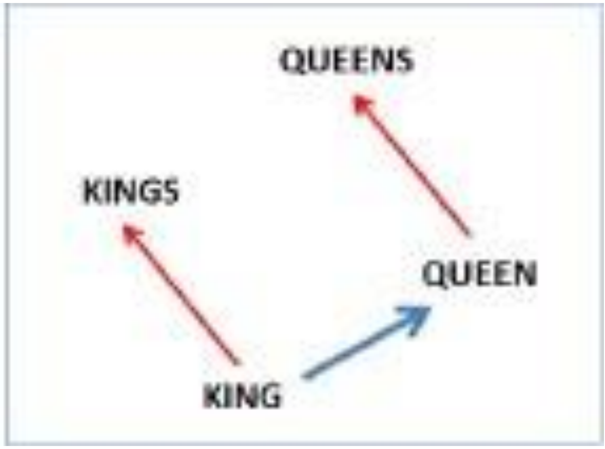
- Motivation
- What is word embedding?
- A semantic task
- Bias in word embedding
- Neural LM

A semantic task (Mikolov et al., 2013)

- Task: “A:B C:D”
  - Training: given a corpus of text, learn word embedding
  - Test: given A, B, and C, find D
  - Evaluation: percentage of examples with the correct D.

- Examples:
  - Good:better   rough:____
  - Year:years   law:____
  - See:sees   return:____
  - come:go   borrow:____





Algorithm

- A:B is like C:D

  ➔ $x_b - x_a = x_d - x_c$

  ➔ $x_b - x_a + x_c = x_d$

- Represent each word w as a word vector x_w
- Compute $y = x_b - x_a + x_c$
- Find $w^* = \arg\max_w \mathrm{sim}(x_w, y)$

Results

Method	Adjectives	Nouns	Verbs	All
LSA-80	9.2	11.1	17.4	12.8
LSA-320	11.3	18.1	20.7	16.5
LSA-640	9.6	10.1	13.8	11.3
RNN-80	9.3	5.2	30.4	16.2
RNN-320	18.2	19.0	45.0	28.5
RNN-640	21.0	25.2	54.8	34.7
RNN-1600	**23.9**	**29.2**	**62.2**	**39.6**

Bias in word embedding

- (Bolukbasi et al., 2016): Man is to computer programmer as woman is to homemaker? Debiasing word embeddings

- Ex: man : woman = king : queen
- But man : woman = programmer : homemaker

Extreme *she* occupations	Extreme *he* occupations
1. homemaker	1. maestro
nurse	2. skipper
receptionist	
librarian	philosopher
socialite	captain
hairdresser	architect
8. nanny	8. financier
9. bookkeeper	9. warrior
stylist	broadcaster
housekeeper	magician

Gender stereotype *she-he* analogies

sewing-carpentry	registered nurse-physician	housewife-shopkeeper
nurse-surgeon	interior designer-architect	softball-baseball
		cosmetics-pharmaceuticals
giggle-chuckle	vocalist-guitarist	petite-lanky
sassy-snappy	diva-superstar	charming-affable
volleyball-football		lovely-brilliant

Gender appropriate *she-he* analogies

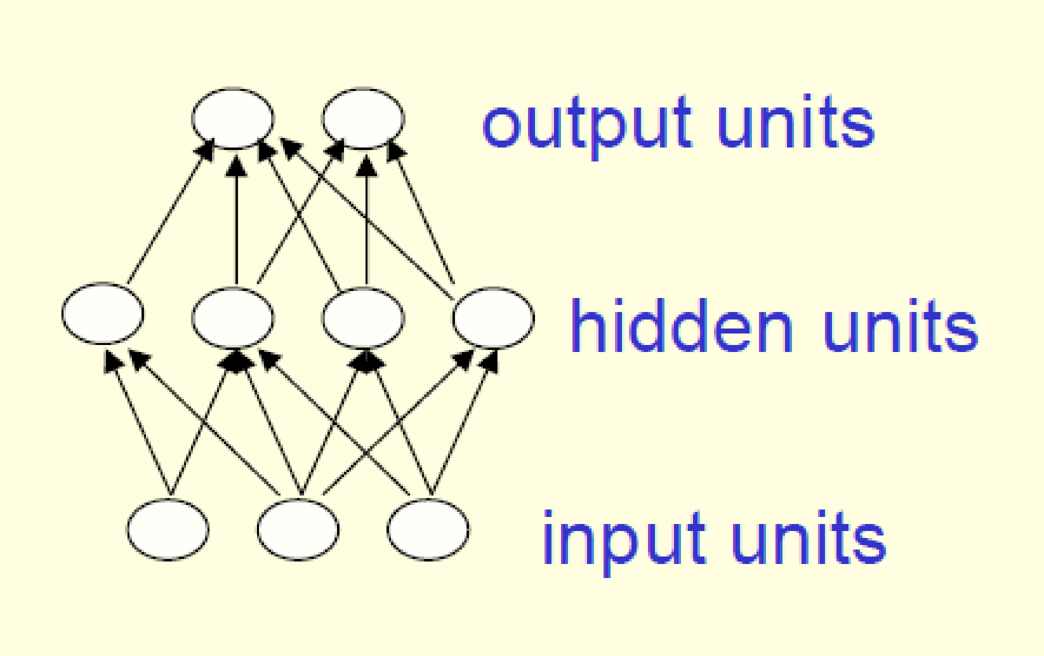
queen-king	sister-brother	mother-father
waitress-waiter	ovarian cancer-prostate cancer	convent-monastery

Outline

- Motivation
- What is word embedding?
- A semantic task
- Bias in word embedding
- Neural LM

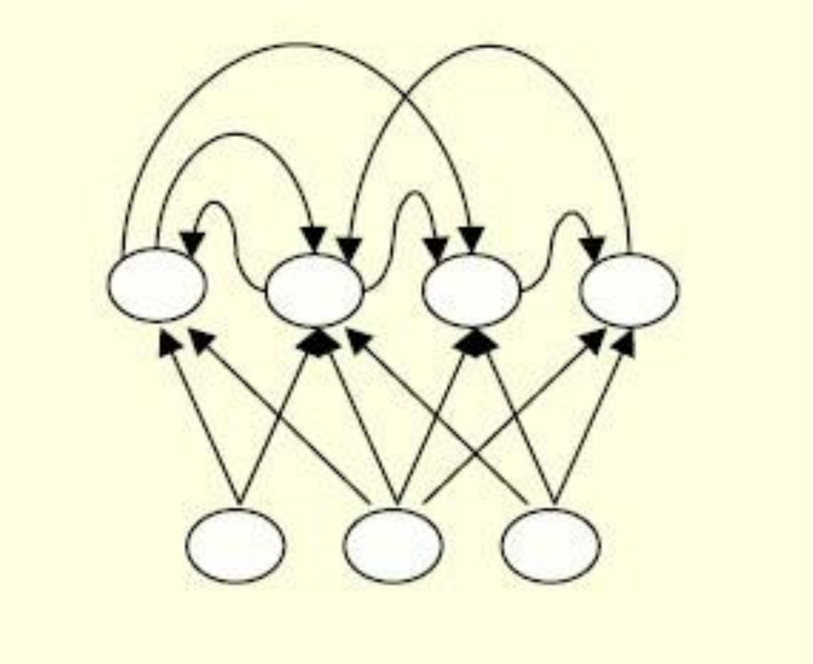
Feed-forward neural network

- This is the simplest type of NN:
  - The first layer is the input and the last layer is the output
  - If there is more than one hidden layer, we call them deep NN
- Training: learn the weights on the arcs, using backpropagation.



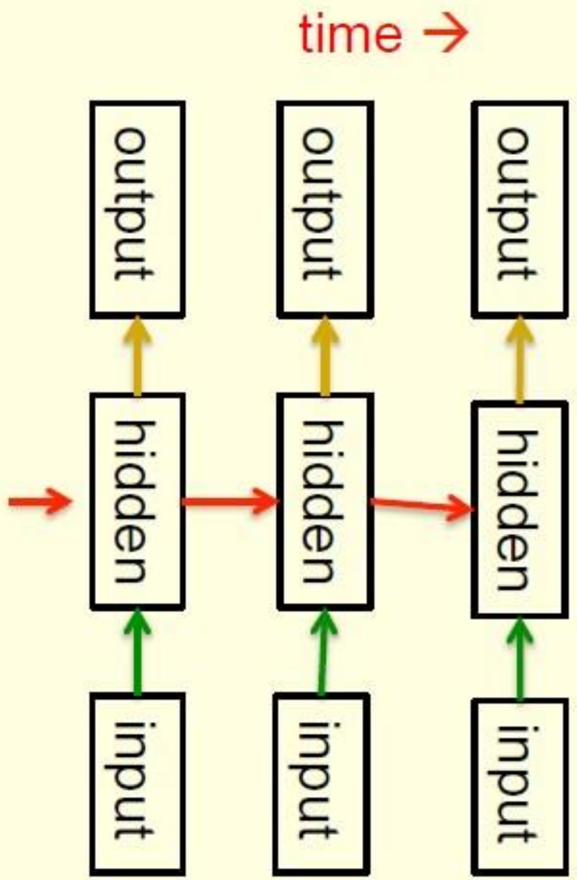
Recurrent neural network (RNN)

- There are directed cycles in the network.
- The training becomes more difficult.
- One of the most commonly used NN types in the NLP field right now.

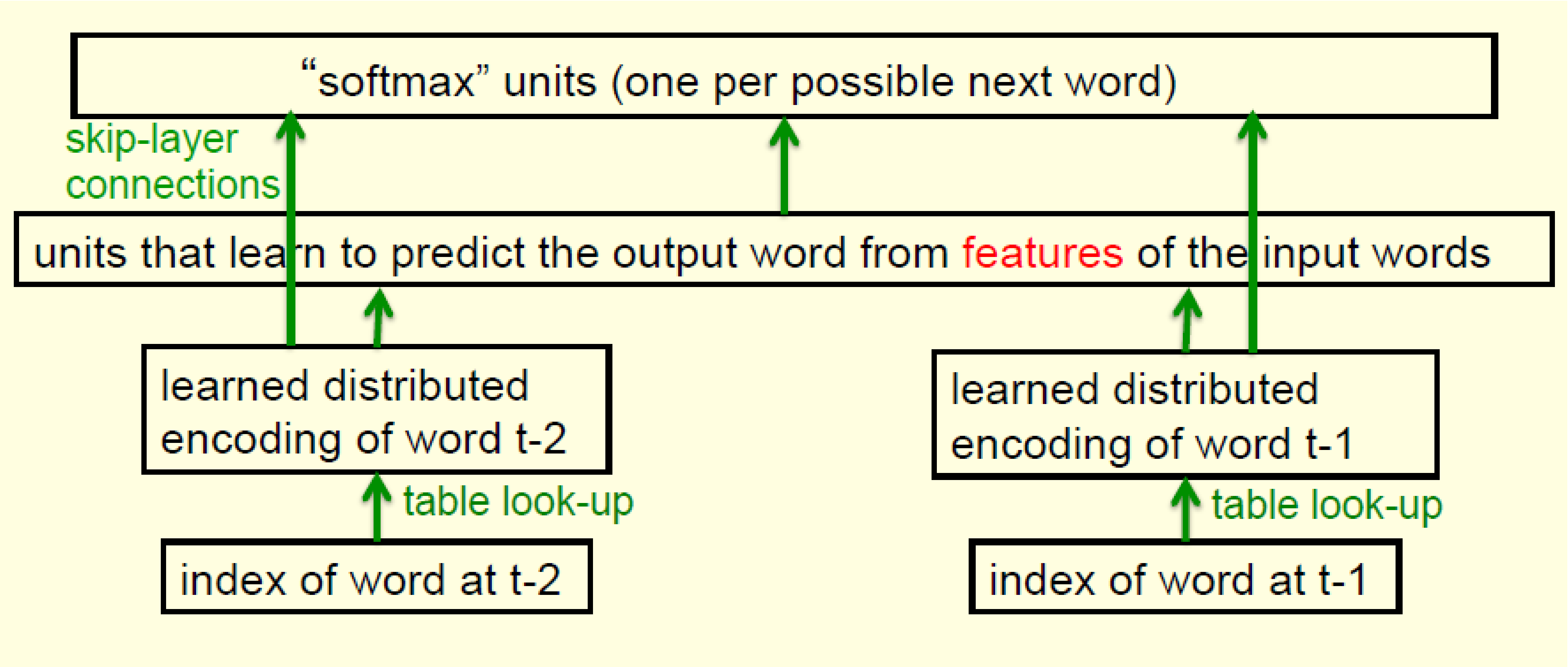


RNNs for modeling sequences

- Recurrent neural networks are a very natural way to model sequential data:
  - They are equivalent to very deep nets with one hidden layer per time slice.
  - Except that they use the same weights at every time slice and they get input at every time slice.
- They have the ability to remember information in their hidden state for a long time.
  - But its very hard to train them to use this potential.



Bengio’s neural LM

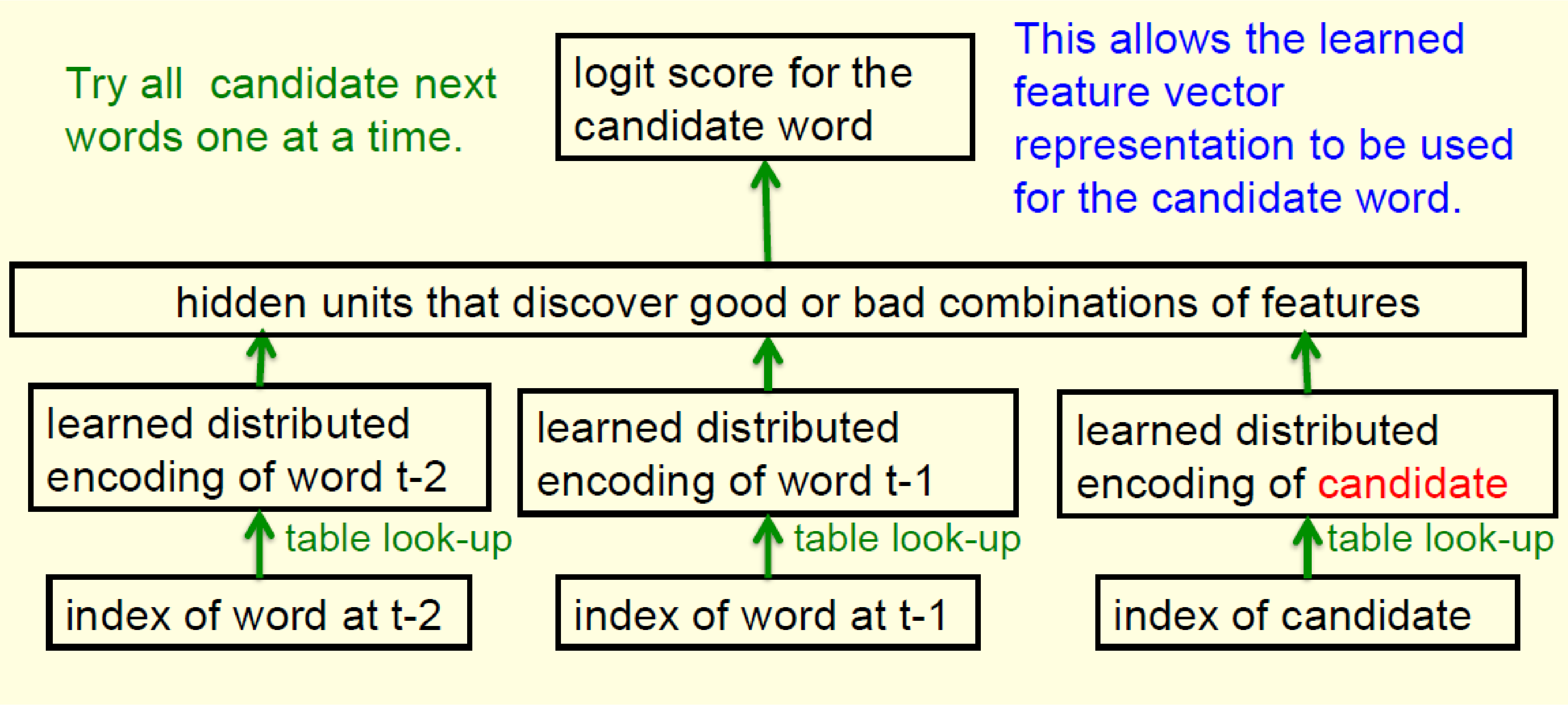


A problem with having 100K output words

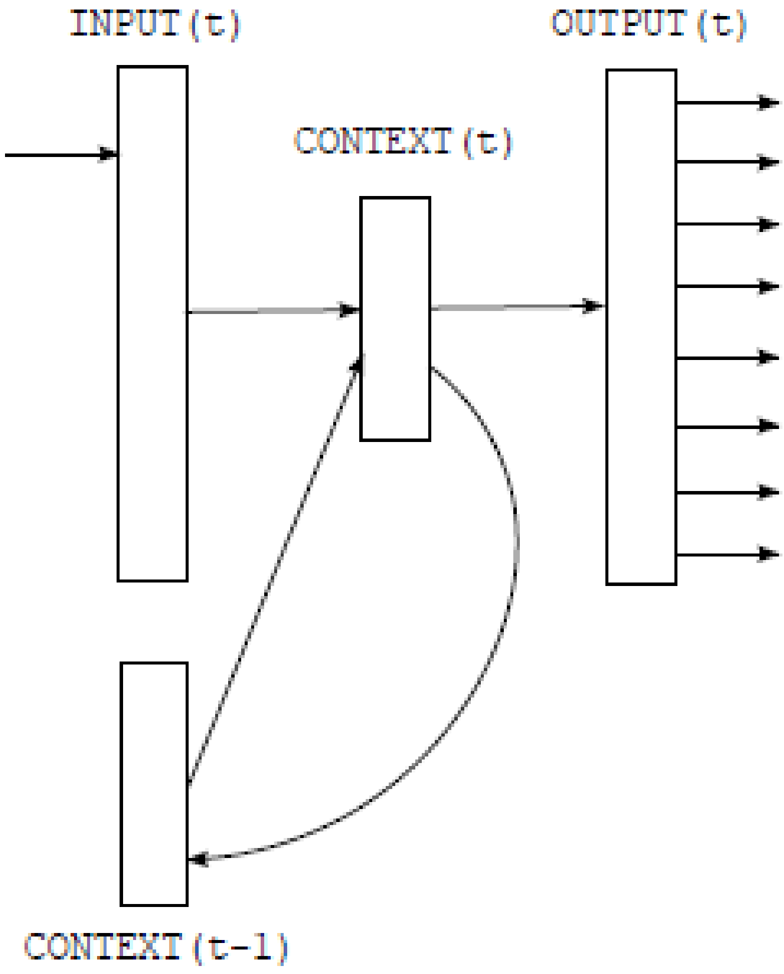
- Each unit in the last hidden layer has 100K outgoing weights:
  - If the number of units in the hidden layer is big, we need a huge number of training instances.
  - If the number of units in the hidden layer is small, it's hard to get the 100K probabilities right.

- Is there a better way to deal with a large number of outputs?

A serial architecture



Recurrent Neural Network LM (Mikolov et al., 2010)



Input layer is w(t) and s(t-1)
Hidden layer represents context s(t)
Output layer is the prob distribution of w(t+1)

W(t) uses 1-to-N coding.

Training is slow
➔ Cannot use large amount of data

Results

Table 1: *Performance of models on WSJ DEV set when increasing size of training data.*

Model	# words	PPL	WER
KN5 LM	200K	336	16.4
KN5 LM + RNN 90/2	200K	271	15.4
KN5 LM	1M	287	15.1
KN5 LM + RNN 90/2	1M	225	14.0
KN5 LM	6.4M	221	13.5
KN5 LM + RNN 250/5	6.4M	156	11.7

Summary

- Word embedding is to represent a word as a vector, and it is often used in the input layer of neural networks.

- Neural networks have been used in many NLP tasks. As a case study, we look at three neural LMs.

- There are tons of recent studies on neural network. To find out more, go to ACL anthology, and look at recent proceedings (e.g. ACL 2017).