COMP9444
Neural Networks and Deep Learning

5b. Word Vectors

Outline

- statistical language processing
- *n*-gram models
- co-occurence matrix
- word representations
- word2vec
- word relationships
- neural machine translation
- combining images and language

Word Meaning – Synonyms and Taxonomy?

What is the meaning of meaning?

- dictionary definitions
- synonyms and antonyms
- taxonomy
  - penguin is-a bird is-a mammal is-a vertebrate

Statistical Language Processing

Synonyms for "elegant"

> stylish, graceful, tasteful, discerning, refined, sophisticated, dignified, cultivated, distinguished, classic, smart, fashionable, modish, decorous, beautiful, artistic, aesthetic, lovely; charming, polished, suave, urbane, cultured, dashing, debonair; luxurious, sumptuous, opulent, grand, plush, high-class, exquisite

Synonyms, antonyms and taxonomy require human effort, may be incomplete and require discrete choices. Nuances are lost. Words like "king", "queen" can be similar in some attributes but opposite in others.

Could we instead extract some statistical properties automatically, without human involvement?

There was a Crooked Man

There was a crooked man,
who walked a crooked mile
And found a crooked sixpence
upon a crooked stile.
He bought a crooked cat,
who caught a crooked mouse
And they all lived together
in a little crooked house.

www.kearley.co.uk/images/uploads/JohnPatiencePJ03.gif

Counting Frequencies

word	frequency
a	7
all	1
and	2
bought	1
cat	1
caught	1
crooked	7
found	1
he	1
house	1
in	1
little	1
lived	1
man	1
mile	1
mouse	1
sixpence	1
stile	1
there	1
they	1
together	1
upon	1
walked	1
was	1
who	2

- some words occur frequently in all (or most) documents
- some words occur frequently in a particular document, but not generally
- this information can be useful for document classification

Document Classification

word	doc 1	doc 2	doc X
a	.	.	7
all	.	.	1
and	.	.	2
bought	.	.	1
cat	.	.	1
caught	.	.	1
crooked	.	.	7
found	.	.	1
he	.	.	1
house	.	.	1
in	.	.	1
little	.	.	1
lived	.	.	1
man	.	.	1
mile	.	.	1
mouse	.	.	1
sixpence	.	.	1
stile	.	.	1
there	.	.	1
they	.	.	1
together	.	.	1
upon	.	.	1
walked	.	.	1
was	.	.	1
who	.	.	2

Document Classification

- each column of the matrix becomes a vector representing the corresponding document
- words like “cat”, “mouse”, “house” tend to occur in children’s books or rhymes
- other groups of words may be characteristic of legal documents, political news, sporting results, etc.
- words occurring many times in one document may skew the vector – might be better to just have a “1” or “0” indicating whether the word occurs at all

Counting Consecutive Word Pairs

word	a	all	and	bought	cat	caught	crooked	found	he	house	in	little	lived	man	mile	mouse	sixpence	stile	there	they	together	upon	walked	was	who
a							6					1													
all													1												
and								1												1					
bought	1																								
cat																									1
caught	1																								
crooked					1					1				1	1	1	1	1							
found	1																								
he				1																					
house																									
in	1																								
little							1																		
lived																					1				
man																									1
mile			1																						
mouse			1																						
sixpence																						1			
stile									1																
there																								1	
they		1																							
together											1														
upon	1																								
walked	1																								
was	1																								
who						1																	1		

Predictive 1-Gram Word Model

word	a	all	and	bought	cat	caught	crooked	found	he	house	in	little	lived	man	mile	mouse	sixpence	stile	there	they	together	upon	walked	was	who
a							$\frac{6}{7}$					$\frac{1}{7}$													
all													1												
and								$\frac{1}{2}$												$\frac{1}{2}$					
bought	1																								
cat																									1
caught	1																								
crooked					$\frac{1}{7}$					$\frac{1}{7}$				$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$							
found	1																								
he				1																					
house																									
in	1																								
little							1																		
lived																					1				
man																									1
mile			1																						
mouse			1																						
sixpence																						1			
stile									1																
there																								1	
they		1																							
together											1														
upon	1																								
walked	1																								
was	1																								
who						$\frac{1}{2}$																	$\frac{1}{2}$		

N-Gram Model

- by normalizing each row (to sum to 1) we can estimate the probability $\text{prob}(w_j|w_i)$ of word w_j occurring after w_i
- need to aggregate over a large corpus, so that unusual words like “crooked” will not dominate
- the model captures some common combinations like “there was”, “man who”, “and found”, “he bought”, “who caught”, “and they”, “they all”, “lived together”, etc.
- this unigram model can be generalized to a bi-gram, tri-gram, $\ldots$, n-gram model by considering the n preceding words
- if the vocabulary is large, we need some tricks to avoid exponential use of memory

1-Gram Text Generator

"Rashly – Good night is very liberal – it is easily said there is – gyved to a sore distraction in wrath and with my king may choose but none of shapes and editing by this , and shows a sea And what this is miching malhecho ; And gins to me a pass , Transports his wit , Hamlet , my arms against the mind impatient , by the conditions that would fain know ; which , the wicked deed to get from a deed to your tutor ."

Co-occurrence Matrix

- sometimes, we don't necessarily predict the next word, but simply a "nearby word" (e.g. a word occurring within an n-word window centered on that word)
- we can build a matrix in which each row represents a word, and each column a nearby word
- each row of this matrix could be considered as a vector representation for the corresponding word, but the number of dimensions is equal to the size of the vocabulary, which could be very large ($\sim 10^5$)
  - is there a way to reduce the dimensionality while still preserving the relationships between words?

Co-occurrence Matrix (2-word window)

word	a	all	and	bought	cat	caught	crooked	found	he	house	in	little	lived	man	mile	mouse	sixpence	stile	there	they	together	upon	walked	was	who
a				1		1	6	1			1	1										1	1	1	
all													1							1					
and								1							1	1				1					
bought	1								1																
cat							1																		1
caught	1																								1
crooked	6				1					1		1		1	1	1	1	1							
found	1		1																						
he				1														1							
house							1																		
in	1																				1				
little	1						1																		
lived		1																			1				
man							1																		1
mile			1				1																		
mouse			1				1																		
sixpence							1															1			
stile							1		1																
there																								1	
they		1	1																						
together											1		1												
upon	1																1								
walked	1																								1
was	1																		1						
who					1	1								1									1		

Co-occurrence Matrix (10-word window)

word	a	all	and	bought	cat	caught	crooked	found	he	house	in	little	lived	man	mile	mouse	sixpence	stile	there	they	together	upon	walked	was	who
a	10	2	3	2	2	2	13	3	2	1	1	1	1	2	2	1	2	2	1	2	1	2	2	1	4
all	2		1				1				1	1	1			1				1	1				
and	3	1				1	3	1			1		1		1	1	1			1	1	1	1		2
bought	2				1	1	2		1									1				1			1
cat	2			1		1	2		1							1		1							1
caught	2		1	1	1		2									1				1					1
crooked	13	1	3	2	2	2	10	2	2	1	1	1	2	2	2	1	2	3	1	1	1	2	2	1	4
found	3		1				2								1		1					1	1		
he	2			1	1		2										1	1				1			1
house	1						1				1	1									1				
in	1	1	1				1			1		1	1							1	1				
little	1	1					1			1	1		1								1				
lived	1	1	1				2				1	1				1				1	1				
man	2						2								1				1				1	1	1
mile	2		1				2	1						1			1						1		1
mouse	1	1	1		1	1	1						1							1	1				1
sixpence	2		1				2	1	1						1			1				1			
stile	2			1	1		3		1								1					1			
there	1						1							1										1	1
they	2	1	1			1	1				1		1			1					1				
together	1	1	1				1			1	1	1	1			1				1					
upon	2		1	1			2	1	1								1	1							
walked	2		1				2	1						1	1									1	1
was	1						1							1					1				1		1
who	4		2	1	1	1	4		1					1	1	1			1				1	1	

Co-occurrence Matrix

- by aggregating over many documents, pairs (or groups) of words emerge which tend to occur near each other (but not necessarily consecutively)
  - "cat", "caught", "mouse"
  - "walked", "mile"
  - "little", "house"
- common words tend to dominate the matrix
  - could we sample common words less often, in order to reveal the relationships of less common words?

Word Embeddings

"Words that are used and occur in the same contexts tend to purport similar meanings."

Z. Harris (1954)

"You shall know a word by the company it keeps."

J.R. Firth (1957)

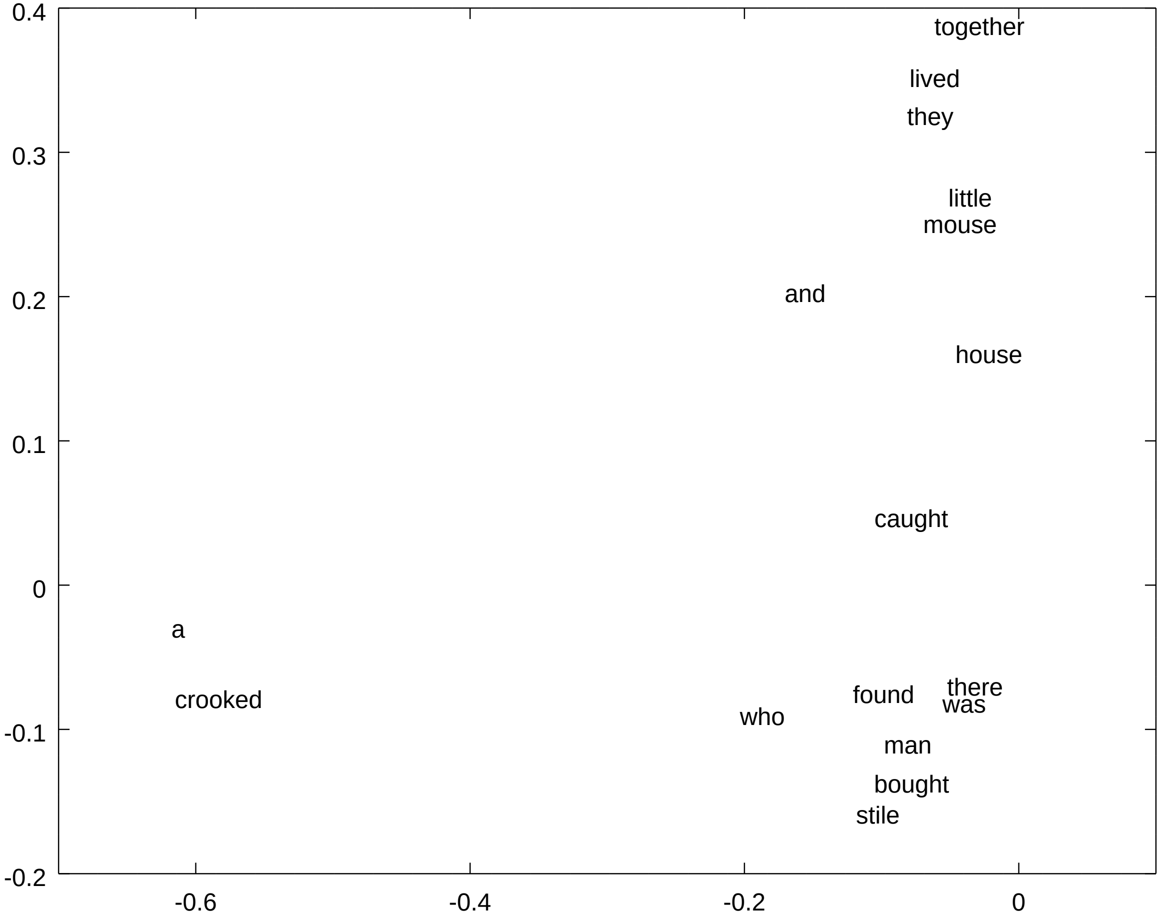
Aim of Word Embeddings:

Find a vector representation of each word, such that words with nearby representations are likely to occur in similar contexts.

History of Word Embeddings

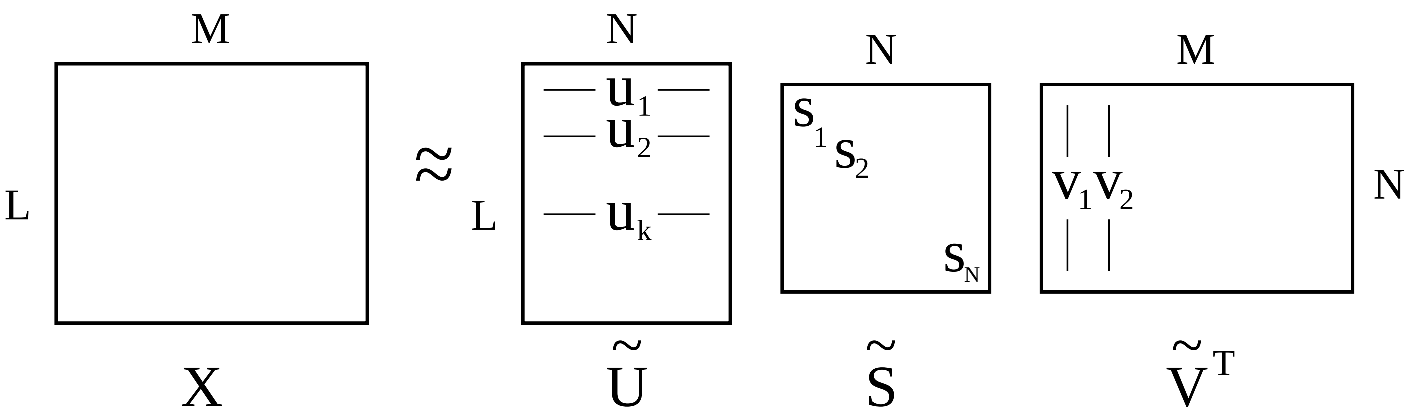
- Structuralist Linguistics (Firth, 1957)
- Recurrent Networks (Rumelhart, Hinton & Williams, 1986)
- Latent Semantic Analysis (Deerwester et al., 1990)
- Hyperspace Analogue to Language (Lund, Burgess & Atchley, 1995)
- Neural Probabilistic Language Models (Bengio, 2000)
- NLP (almost) from Scratch (Collobert et al., 2008)
- word2vec (Mikolov et al., 2013)
- GloVe (Pennington, Socher & Manning, 2014)

Word Embeddings



Singular Value Decomposition

Co-occurrence matrix $X_{(L\times M)}$ can be decomposed as $X = U\,S\,V^T$ where $U_{(L\times L)}$, $V_{(M\times M)}$ are unitary (all columns have unit length) and $S_{(L\times M)}$ is diagonal, with diagonal entries $s_1 \geq s_2 \geq \ldots \geq s_M \geq 0$



We can obtain an approximation for X of rank $N < M$ by truncating U to $\tilde{U}_{(L\times N)}$, S to $\tilde{S}_{(N\times N)}$ and V to $\tilde{V}_{(N\times M)}$. The kth row of $\tilde{U}$ then provides an N-dimensional vector representing the k^{th} word in the vocabulary.

word2vec and GloVe

Typically, L is the number of words in the vocabulary (about $60,000$) and M is either equal to L or, in the case of document classification, the number of documents in the collection. SVD is computationally expensive, proportional to $L \times M^2$ if $L \geq M$. Can we generate word vectors in a similar way but with less computation, and incrementally?

- word2vec
  - predictive model
  - maximize the probability of a word based on surrounding words
- GloVe
  - count-based model
  - reconstruct a close approximation to the co-occurrence matrix X

Eigenvalue vs. Singular Value Decomposition

Eigenvalue Decomposition:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \Omega \mathrm{D} \Omega^{-1}, \quad \text{where} \quad \Omega = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad \mathrm{D} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \Omega \mathrm{D} \Omega^{-1}, \quad \text{where} \quad \Omega = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -i & i \end{bmatrix}, \quad \mathrm{D} = \begin{bmatrix} i & 0 \\ 0 & -i \end{bmatrix}$$

Singular Value Decomposition:

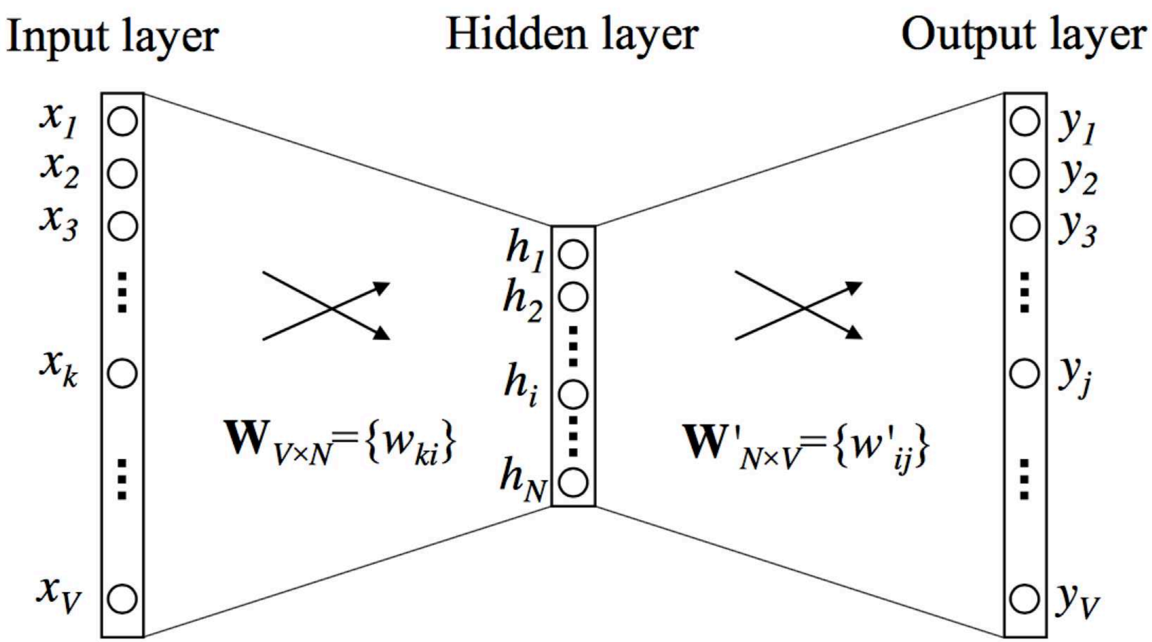
$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \mathrm{USV}^{\mathrm{T}}, \quad \mathrm{U} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathrm{S} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathrm{V} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \mathrm{USV}^{\mathrm{T}}, \quad \mathrm{U} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathrm{S} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathrm{V} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Eigenvalue vs. Singular Value Decomposition

- if X is symmetric and positive semi-definite, eigenvalue and singular value decompositions are the same.
- in general, eigenvalues can be negative or even complex, but singular values are always real and non-negative.
- even if X is a square matrix, singular value decompositon treats the source and target as two entirely different spaces.
- the word co-occurrence matrix is symmetric but not positive semi-definite; for example, if the text consisted entirely of two alternating letters ..ABABABABABABAB.. then A would be the context for B, and vice-versa.

word2vec 1-Word Context Model



The k^{th} row $\mathbf{v}_k$ of $\mathbf{W}$ is a representation of word k.

The j^{th} column $\mathbf{v}'_j$ of $\mathbf{W}'$ is an (alternative) representation of word j.

If the (1-hot) input is k, the linear sum at each output will be $u_j = {\mathbf{v}'_j}^{\mathrm{T}} \mathbf{v}_k$

Cost Function

Softmax can be used to turn these linear sums u_j into a probability distribution estimating the probability of word j occurring in the context of word k

$$\text{prob}(j|k) = \frac{\exp(u_j)}{\sum_{j'=1}^{V} \exp(u_{j'})} = \frac{\exp(\mathbf{v}'^{\,\mathrm{T}}_j \mathbf{v}_k)}{\sum_{j'=1}^{V} \exp(\mathbf{v}'^{\,\mathrm{T}}_{j'} \mathbf{v}_k)}$$

We can treat the text as a sequence of numbers $w_1, w_2, \ldots, w_T$ where $w_i = j$ means that the i^{th} word in the text is the j^{th} word in the vocabulary.

We then seek to maximize the log probability

$$\frac{1}{T} \sum_{t=1}^{T} \sum_{-c \le r \le c, r \ne 0} \log \text{prob}(w_{t+r}|w_t)$$

where c is the size of training context (which may depend on w_t)

word2vec Issues

- word2vec is a linear model in the sense that there is no activation function at the hidden nodes
- this 1-word prediction model can be extended to multi-word prediction in two different ways:
  - Continuous Bag of Words
  - Skip-Gram
- need a computationally efficient alternative to Softmax (Why?)
  - Hierarchical Softmax
  - Negative Sampling
- need to sample frequent words less often

word2vec Weight Updates

If we assume the full softmax, and the correct output is j^*, then the cost function is

$$E = -u_{j^*} + \log \sum_{j'=1}^{V} \exp(u_{j'})$$

the output differentials are

$$e_j = \frac{\partial E}{\partial u_j} = -\delta_{jj^*} + \frac{\partial}{\partial u_j} \log \sum_{j'=1}^{V} \exp(u_{j'})$$

where

$$\delta_{jj^*} = \begin{cases} 1, & \text{if} \quad j = j^*, \\ 0, & \text{otherwise.} \end{cases}$$

word2vec Weight Updates

hidden-to-output differentials

$$\frac{\partial E}{\partial w'_{ij}} = \frac{\partial E}{\partial u_j}\frac{\partial u_j}{\partial w'_{ij}} = e_j h_i$$

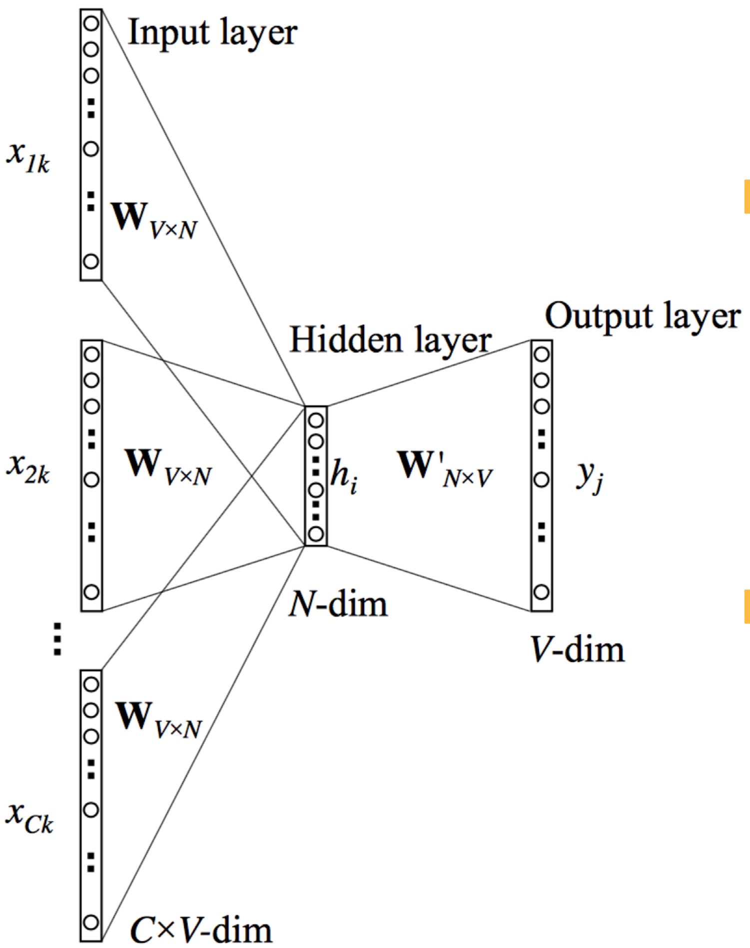
hidden unit differentials

$$\frac{\partial E}{\partial h_i} = \sum_{j=1}^{V} \frac{\partial E}{\partial u_j}\frac{\partial u_j}{\partial h_i} = \sum_{j=1}^{V} e_j w'_{ij}$$

input-to-hidden differentials

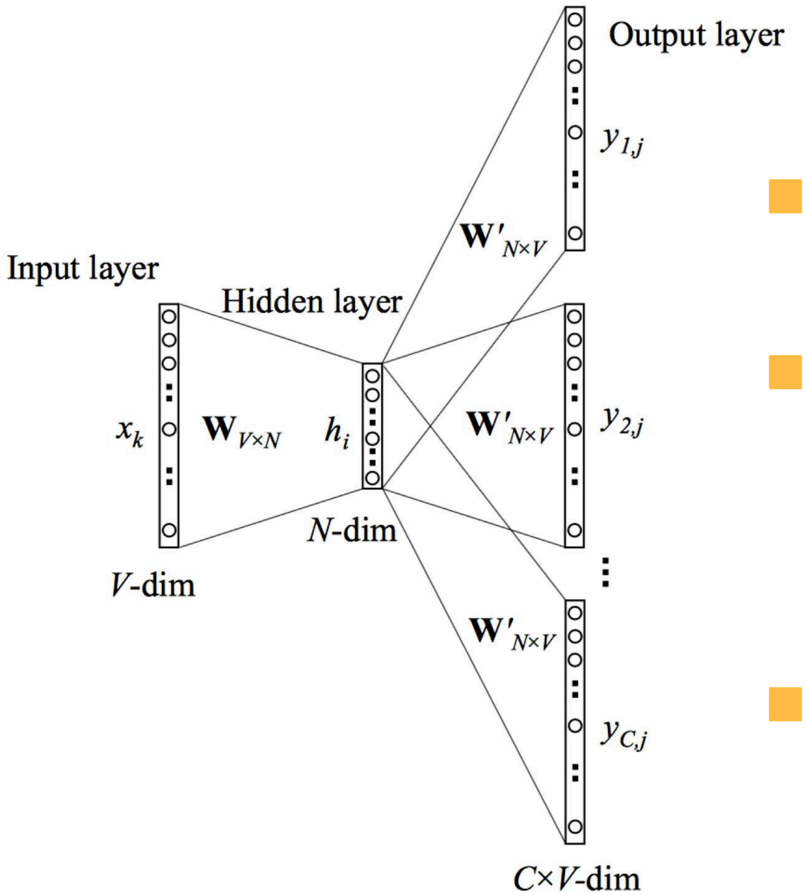
$$\frac{\partial E}{\partial w_{ki}} = \frac{\partial E}{\partial h_i}\frac{\partial h_i}{\partial w_{ki}} = \sum_{j=1}^{V} e_j w'_{ij} x_k$$

Continuous Bag Of Words



- If several context words are each used independently to predict the center word, the hidden activation becomes a sum (or average) over all the context words
- Note the difference between this and NetTalk – in word2vec (CBOW) all context words share the same input-to-hidden weights

word2vec Skip-Gram Model

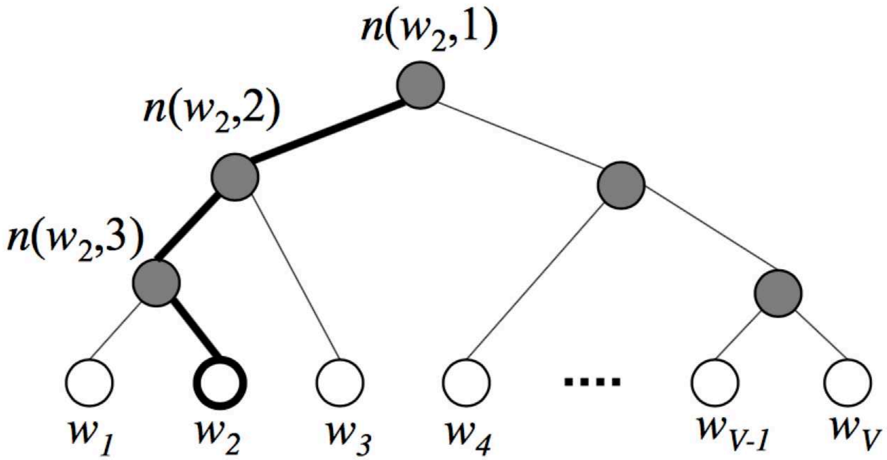


- try to predict the context words, given the center word
- this skip-gram model is similar to CBOW, except that in this case a single input word is used to predict multiple context words
- all context words share the same hidden-to-output weights

Hierarchical Softmax

- target words are organized in a Huffman-coded Binary Tree
- each output of the network corresponds to one branch point in the tree
- only those nodes that are visited along the path to the target word are evaluated (which is $\log_2(V)$ nodes on average)

Hierarchical Softmax



$$[\![n' = \text{child}(n)]\!] = \begin{cases} +1, & \text{if } n' \text{ is left child of node } n, \\ -1, & \text{otherwise.} \end{cases}$$

$$\sigma(u) = 1/(1 - \exp(-u))$$

$$\text{prob}(w = w_t) = \prod_{j=1}^{L(w)-1} \sigma([\![n(w, j+1) = \text{child}(n(w, j))]\!]\,\mathbf{v}'_{n(w,j)}{}^{\mathrm{T}}\,\mathbf{h})$$

Negative Sampling

The idea of negative sampling is that we train the network to increase its estimation of the target word j^* and reduce its estimate not of all the words in the vocabulary but just a subset of them $\mathcal{W}_{\text{neg}}$, drawn from an appropriate distribution.

$$E = -\log \sigma({\mathbf{v}'_{j^*}}^{\mathrm{T}} \mathbf{h}) - \sum_{j \in \mathcal{W}_{\text{neg}}} \log \sigma(-{\mathbf{v}'_j}^{\mathrm{T}} \mathbf{h})$$

This is a simplified version of Noise Constrastive Estimation (NCE).
It is not guaranteed to produce a well-defined probability distribution, but in practice it does produce high-quality word embeddings.

Negative Sampling

- The number of samples is 5-20 for small datasets, 2-5 for large datasets.
- Empirically, a good choice of the distribution from which to draw the negative samples is $P(w) = U(w)^{3/4}/Z$ where $U(w)$ is the unigram distribution determined by the previous word, and Z is a normalizing constant.

Subsampling of Frequent Words

In order to diminish the influence of more frequent words, each word in the corpus is discarded with probability

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}$$

where $f(w_i)$ is the frequency of word w_i and $t \sim 10^{-5}$ is an empirically determined threshold.

Sentence Completion Task

Q1. Seeing the pictures of our old home made me feel .... and nostalgic.

A. fastidious
B. indignant
C. wistful
D. conciliatory

Q2. Because the House had the votes to override a presidential veto, the President has no choice but to ....

A. object
B. abdicate
C. abstain
D. compromise

(use model to choose which word is most likely to occur in this context)

Linguistic Regularities

King + Woman - Man $\simeq$ Queen

More generally,

A is to B as C is to ??

$$d = \text{argmax}_x \frac{(v_c + v_b - v_a)^{\mathrm{T}} v_x}{||v_c + v_b - v_a||}$$

Word Analogy Task

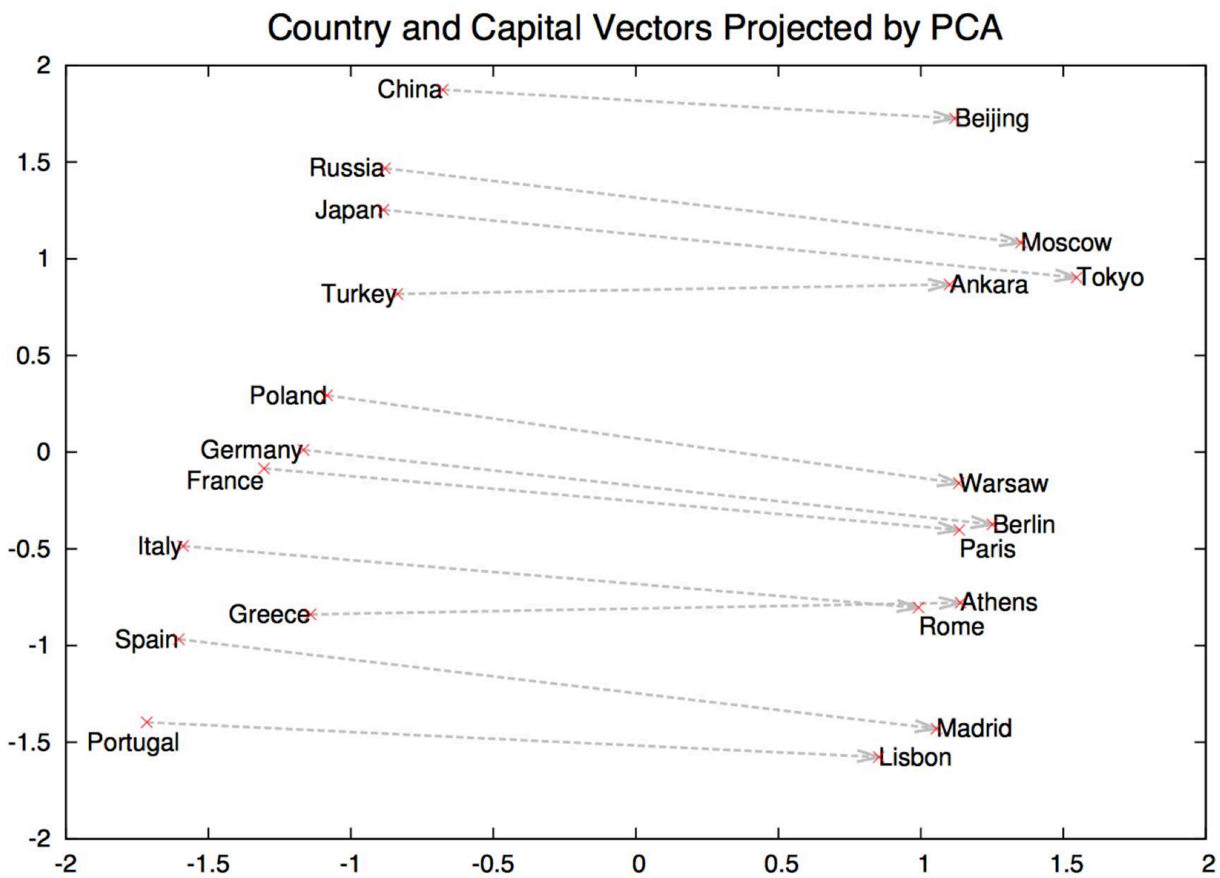
Q1. **evening** is to **morning** as **dinner** is to ....

- A. breakfast
- B. soup
- C. coffee
- D. time

Q2. **bow** is to **arrow** as .... is to **bullet**

- A. defend
- B. lead
- C. shoot
- D. gun

Capital Cities



Word Analogies

Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

Word Relationships

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

References

T. Mikolov, K. Chen, G. Corrado & J. Dean, 2013. "Efficient estimation of word representations in vector space", arXiv preprint arXiv:1301.3781.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado & J. Dean, 2013. "Distributed representations of words and phrases and their compositionality", NIPS 2013, 3111-19.

Xin Rong, 2014. "word2vec parameter learning explained.", arXiv:1411.2738.

`https://nlp.stanford.edu/projects/glove/`

`https://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-gpus-part-3/`