Distributed Representations of Words and Phrases and their Compositionality

2013b

Tomas Mikolov, Ilya Sutskever, Kai Chen,
Greg Corrado, Jeffrey Dean

Seminar “Selected Topics in Semantics and Discourse”,
presenter Yauhen Klimovich, tutor Prof. Manfred Pinkal

What is the distributed representations of words?

We talk about “Distributed Representations of Words and Phrases and their Compositionality” (Mikolov et al)

What is the distributed representations of words?

a vector

e.g. 300-d vector

(3.4, 0.45, …, 7.4, 5.63)

$(x_1, \quad x_2, \quad \ldots \quad, \quad x_{299}, \quad x_{300})$

We talk about "Distributed Representations of Words and Phrases and their Compositionality" (Mikolov et al)

What is the distributed representations of words?

a vector

word embeddings, word projections

e.g. 300-d vector

(3.4, 0.45, …, 7.4, 5.63)

$(x_1, \quad x_2, \quad \ldots \quad , \quad x_{299}, \quad x_{300})$

We talk about “Distributed Representations of Words and Phrases and their Compositionality” (Mikolov et al)

What is the distributed representations of words?

a vector

word embeddings, word projections

e.g. 300-d vector

(3.4, 0.45, …, 7.4, 5.63)

$(x_1, \quad x_2, \quad \ldots \quad , \quad x_{299}, \quad x_{300})$

Good for similarity measure

We talk about "Distributed Representations of Words and Phrases and their Compositionality" (Mikolov et al)

And phrases?

And phrases?

Meaningful word compounds

We talk about “Distributed Representations of Words and Phrases and their Compositionality” (Mikolov et al)

And phrases?

Meaningful word compounds

e.g. ‘Tesla Motors’
is neither Tesla nor Motors,
‘Toronto Maple Leafs’
is not Toronto and a maple and leafs

We talk about “Distributed Representations of Words and Phrases and their Compositionality” (Mikolov et al)

And phrases?

Meaningful word compounds

e.g. ‘Tesla Motors’
is neither Tesla nor Motors,
‘Toronto Maple Leafs’
is not Toronto and a maple and leafs

Good for accuracy

What is compositionality?

What is compositionality?

Math operations on word vectors

We talk about “Distributed Representations of Words and Phrases and their Compositionality” (Mikolov et al)

What is compositionality?

Math operations on word vectors

$$\text{vec}(\text{“Germany”}) + \text{vec}(\text{“capital”}) \approx \text{vec}(\text{“Berlin”})$$

We talk about “Distributed Representations of Words and Phrases and their Compositionality” (Mikolov et al)

What is compositionality?

Math operations on word vectors

vec("Germany") + vec("capital") ≈ vec("Berlin")

vec("Steve Ballmer") - vec("Microsoft") + vec("Google") ≈ vec("Larry Page")

We talk about "Distributed Representations of Words and Phrases and their Compositionality" (Mikolov et al)

What is compositionality?

Math operations on word vectors

vec("Germany") + vec("capital") ≈ vec("Berlin")

vec("Steve Ballmer") - vec("Microsoft") + vec("Google") ≈ vec("Larry Page")

Basic operations can give us better results

We talk about "Distributed Representations of Words and Phrases and their Compositionality" (Mikolov et al)

Agenda

- Skip-gram in details
- Improvements for skip-gram
- Phrases
- Evaluation

We talk about “Distributed Representations of Words and Phrases and their Compositionality” (Mikolov et al)

Why is this paper important?

Improvements for
vector quality and training speed

Why is this paper important?

Improvements for
vector quality and training speed

Continuous skip-gram

*“Efficient Estimation
of Word Representations
in Vector Space”*
[Mikolov et al, 2013]

Why is this paper important?

Improvements for
vector quality and training speed

Continuous skip-gram

*"Efficient Estimation
of Word Representations
in Vector Space"*
[Mikolov et al, 2013]

No dense
matrix multiplication!

Why is this paper important?

Improvements for
vector quality and training speed

Continuous skip-gram

"Efficient Estimation of Word Representations in Vector Space"
[Mikolov et al, 2013]

No dense
matrix multiplication!

Efficient!

We talk about "Distributed Representations of Words and Phrases and their Compositionality" (Mikolov et al)

Why is this paper important?

Improvements for
vector quality and training speed

Continuous skip-gram

"Efficient Estimation of Word Representations in Vector Space"
[Mikolov et al, 2013]

No dense
matrix multiplication!

Efficient!

What is new?

- Negative sampling approach
- Subsampling of frequent words

Why is this paper important?

Improvements for
vector quality and training speed

Continuous skip-gram

"Efficient Estimation of Word Representations in Vector Space"
[Mikolov et al, 2013]

No dense
matrix multiplication!

Efficient!

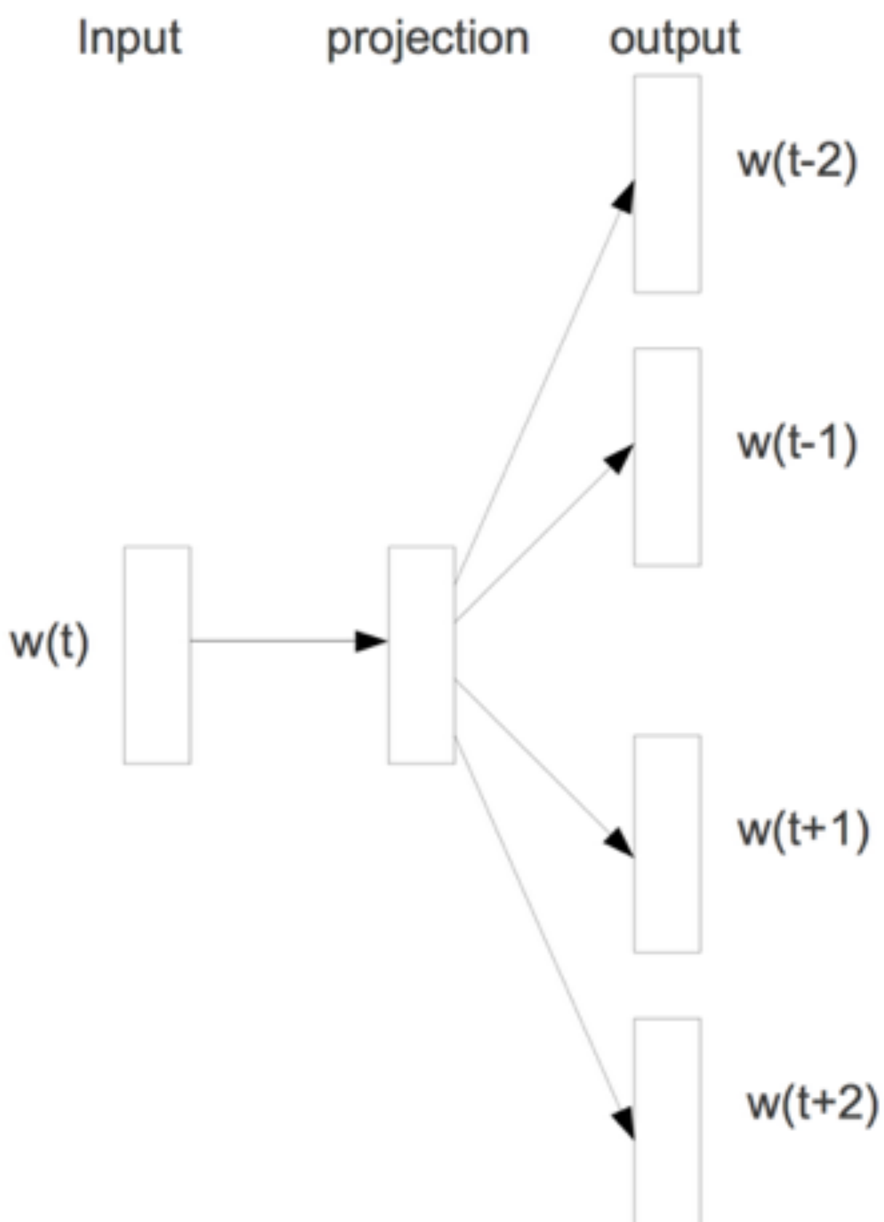
What is new?

- Negative sampling approach
- Subsampling of frequent words

- Phrases (Tesla Motors, Silicon Valley)

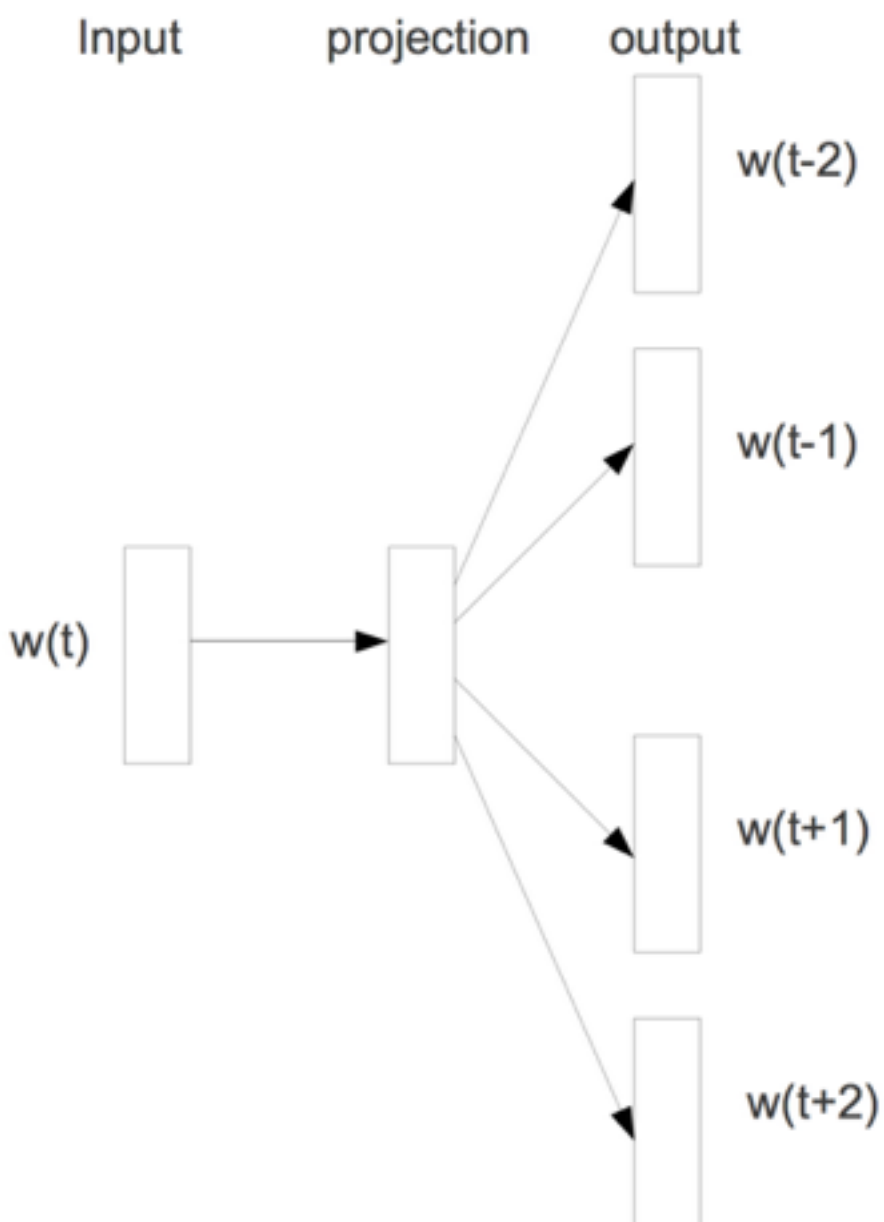
We talk about "Distributed Representations of Words and Phrases and their Compositionality" (Mikolov et al)

Skip-gram model



We talk about “Distributed Representations of Words and Phrases and their Compositionality” (Mikolov et al)

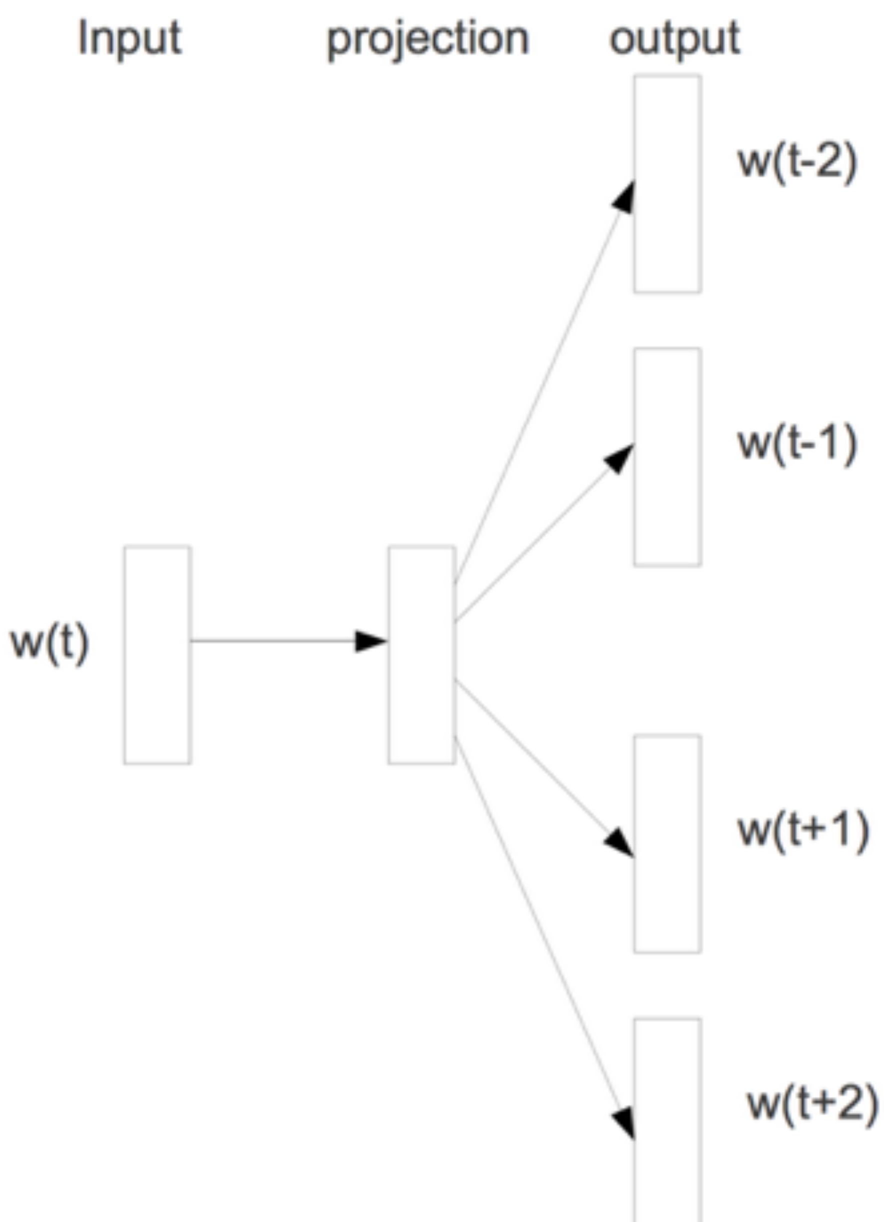
Skip-gram model



$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t)$$

We talk about "Distributed Representations of Words and Phrases and their Compositionality" (Mikolov et al)

Skip-gram model

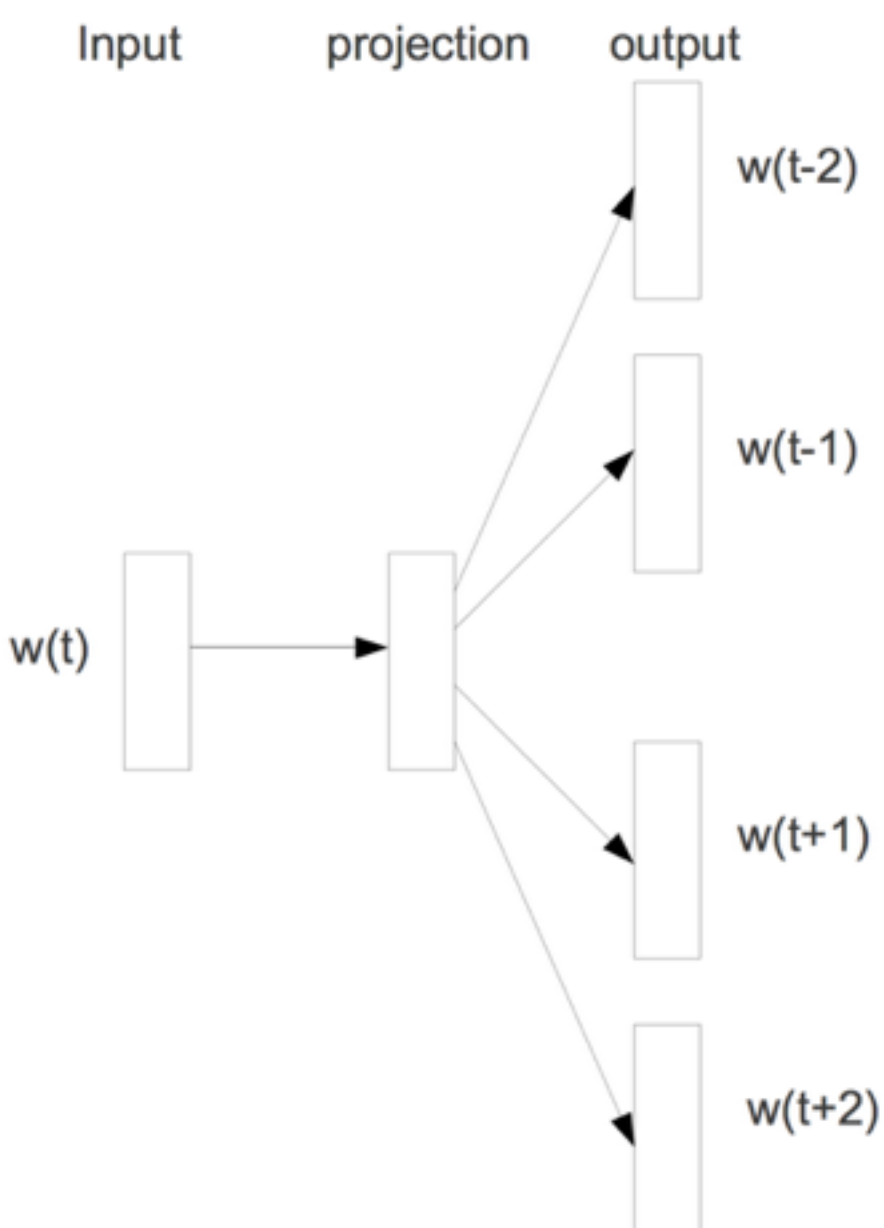


$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c,j\neq 0}\log \boxed{p(w_{t+j}|w_t)}$$

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^\top v_{w_I}\right)}{\sum_{w=1}^{W}\exp\left({v'_w}^\top v_{w_I}\right)}$$

We talk about “Distributed Representations of Words and Phrases and their Compositionality” (Mikolov et al)

Skip-gram model

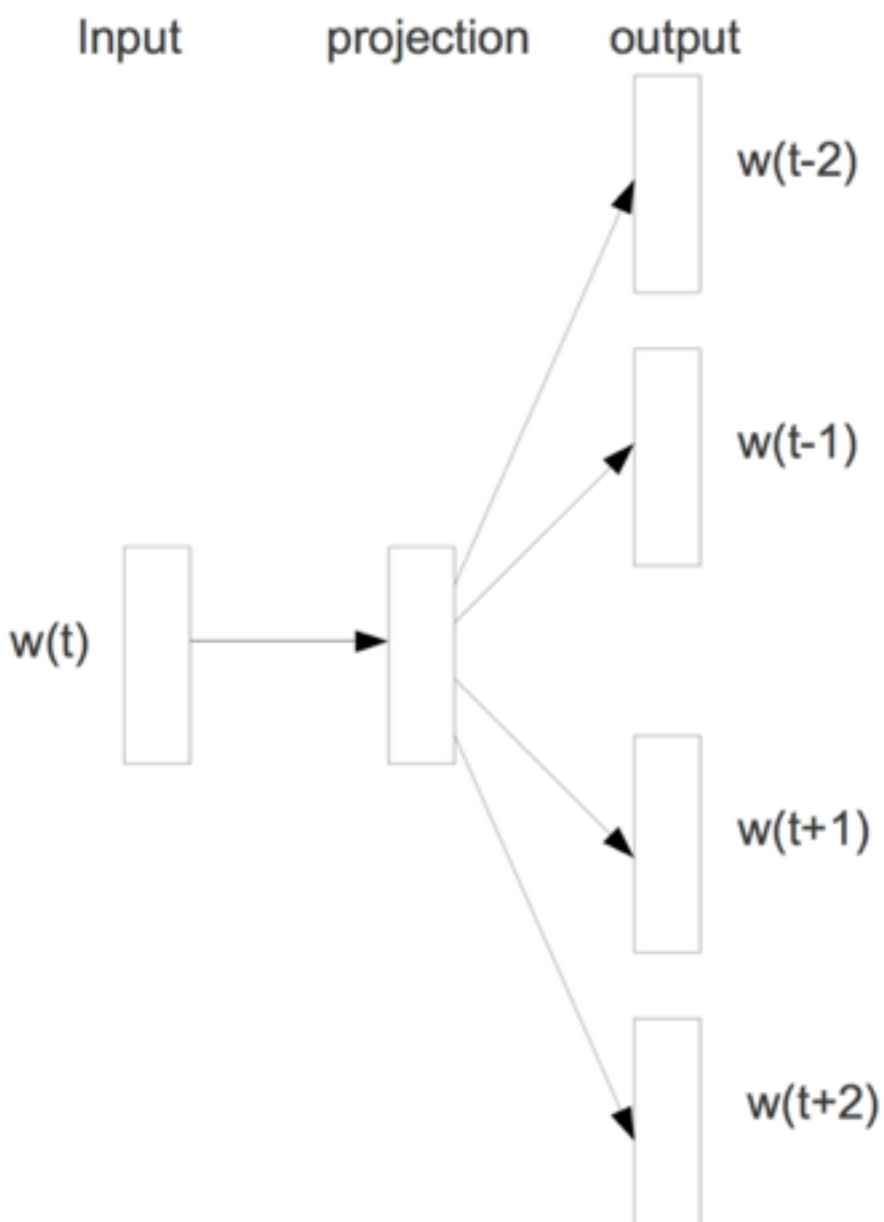


$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log \boxed{p(w_{t+j}|w_t)}$$

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^\top v_{w_I}\right)}{\boxed{\sum_{w=1}^{W}\exp\left({v'_w}^\top v_{w_I}\right)}}$$

inefficient, let's approximate

Skip-gram model



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

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^\top v_{w_I}\right)}{\boxed{\sum_{w=1}^{W} \exp\left({v'_w}^\top v_{w_I}\right)}}$$

inefficient, let's approximate

Hierarchical softmax

binary Huffmann tree

(short codes to frequent words)

We talk about "Distributed Representations of Words and Phrases and their Compositionality" (Mikolov et al)

Hierarchical softmax

Efficient way to compute softmax

Hierarchical softmax

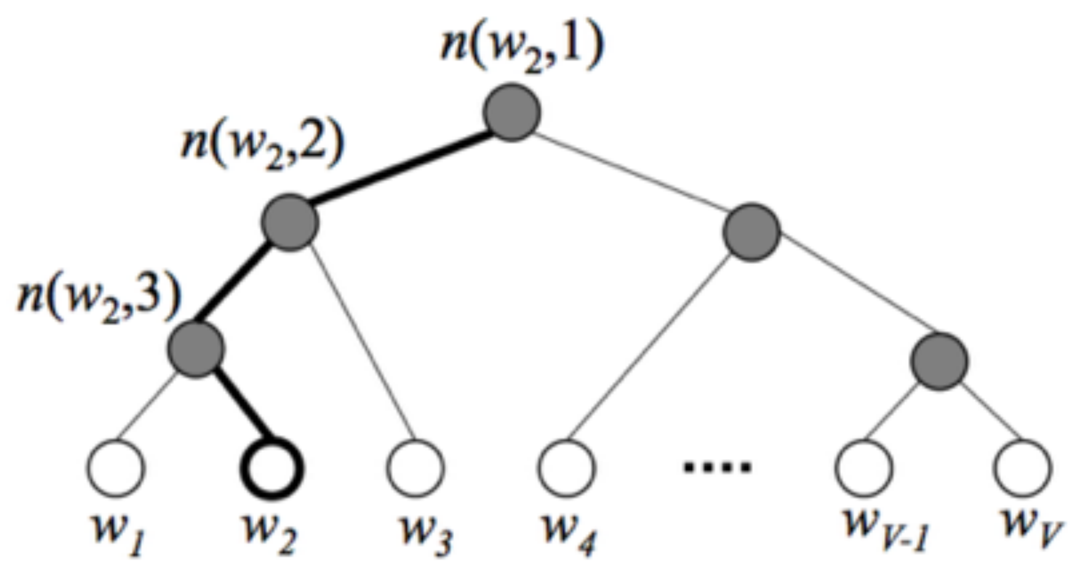
Efficient way to compute softmax

$$p(w|w_I) = \prod_{j=1}^{L(w)-1} \sigma\left([\![n(w, j+1) = \mathrm{ch}(n(w,j))]\!] \cdot {v'_{n(w,j)}}^\top v_{w_I} \right)$$

$$S(t) = \frac{1}{1+e^{-t}}.$$

for normalization

$$[\![x]\!] = \begin{cases} 1 & \text{if } x \text{ is true;} \\ -1 & \text{otherwise.} \end{cases}$$



better for infrequent words, fast training

We talk about “Distributed Representations of Words and Phrases and their Compositionality” (Mikolov et al)

Hierarchical softmax

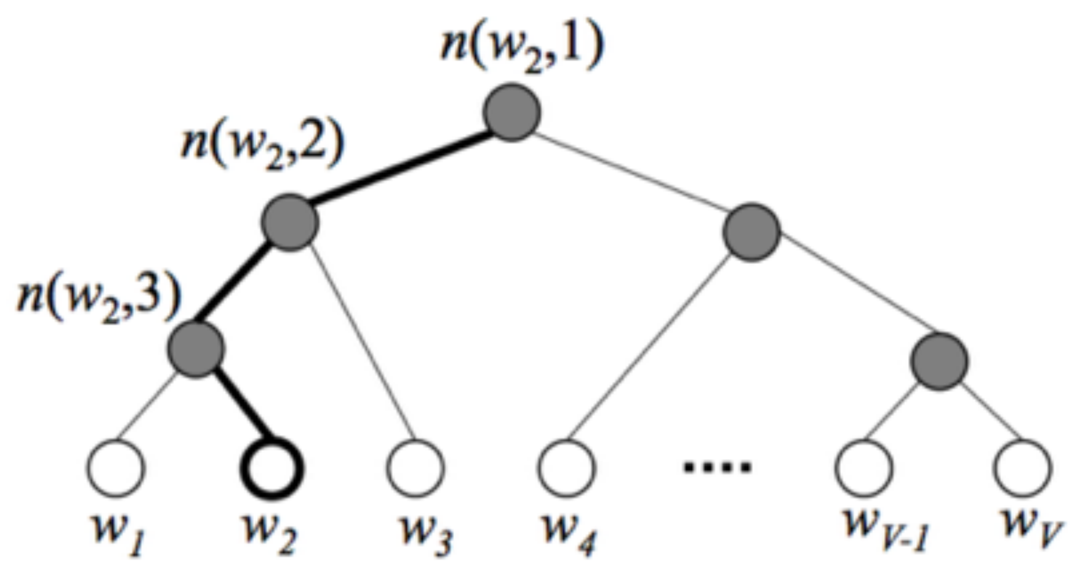
Efficient way to compute softmax

$$p(w|w_I) = \prod_{j=1}^{L(w)-1} \sigma\left([\![n(w, j+1) = \mathrm{ch}(n(w,j))]\!] \cdot {v'_{n(w,j)}}^\top v_{w_I} \right)$$

$$S(t) = \frac{1}{1+e^{-t}}.$$

for normalization

$$[\![x]\!] = \begin{cases} 1 & \text{if } x \text{ is true;} \\ -1 & \text{otherwise.} \end{cases}$$



Max logW
for each word

better for infrequent words, fast training

Hierarchical softmax

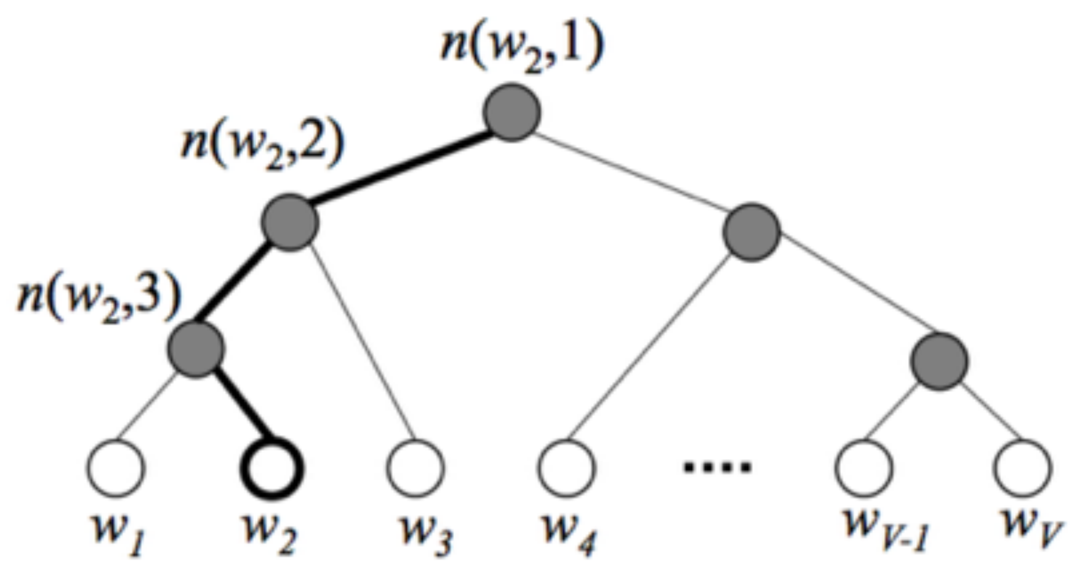
Efficient way to compute softmax

$$p(w|w_I) = \prod_{j=1}^{L(w)-1} \sigma\left([\![n(w, j+1) = \mathrm{ch}(n(w,j))]\!] \cdot {v'_{n(w,j)}}^\top v_{w_I} \right)$$

$$S(t) = \frac{1}{1+e^{-t}}.$$

for normalization

$$[\![x]\!] = \begin{cases} 1 & \text{if } x \text{ is true;} \\ -1 & \text{otherwise.} \end{cases}$$



Max logW
for each word

better for infrequent words, fast training

Structure of the tree is important

We talk about “Distributed Representations of Words and Phrases and their Compositionality” (Mikolov et al)

Negative sampling

idea is based on Noise Contrastive Estimation (NCE)

$$\log \sigma({v'_{w_O}}^\top v_{w_I}) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)} \left[\log \sigma(-{v'_{w_i}}^\top v_{w_I}) \right]$$

We talk about “Distributed Representations of Words and Phrases and their Compositionality” (Mikolov et al)

Negative sampling

idea is based on Noise Contrastive Estimation (NCE)

$$\log \sigma({v'_{w_O}}^\top v_{w_I}) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim \boxed{P_n(w)}} \left[\log \sigma(-{v'_{w_i}}^\top v_{w_I}) \right]$$

noise distribution:
best result is given by U(ω)^(¾)

trained classifier

We talk about “Distributed Representations of Words and Phrases and their Compositionality” (Mikolov et al)

Negative sampling

idea is based on Noise Contrastive Estimation (NCE)

$$\log \sigma({v'_{w_O}}^\top v_{w_I}) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim \boxed{P_n(w)}} \left[\log \sigma(-{v'_{w_i}}^\top v_{w_I}) \right]$$

noise distribution:
best result is given by U(ω)^(¾)

trained classifier

better for frequent words,
better with low dimensional vectors

Subsampling of frequent words

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

Subsampling of frequent words

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

threshold 1/10^5

Frequency of *wi*

improves the accuracy of the learned vectors of the rare words

Subsampling of frequent words

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

threshold 1/10^5

Frequency of *wi*

improves the accuracy of the learned vectors
of the rare words

cutting them off, context is larger

Evaluation

Analogical reasoning task:

- syntactic analogy
- semantic analogy

Data

- 1b words;
- cut out infrequent words(<5t),
- they got |Voc| = 692K

about 19.5k samples

- Berlin Germany Bern Switzerland
- boy girl brother sister
- amazing amazingly apparent apparently
- acceptable unacceptable certain uncertain
- cold colder great greater
- Europe euro Romania leu
- bright brightest sharp sharpest
- code coding jump jumping
- Belarus Belorussian Germany German
- flying flew enhancing enhanced
- car cars cat cats
- enhance enhances work works

Evaluation(results)

Analogical reasoning task:

- syntactic analogy
- semantic analogy

Data

- 1b words;
- cut out infrequent words(<5t),
- they got |Voc| = 692K

V

Method	Time [min]	Syntactic [%]	Semantic [%]	Total accuracy [%]
NEG-5	38	63	54	59
NEG-15	97	63	58	**61**
HS-Huffman	41	53	40	47
NCE-5	38	60	45	53
The following results use 10^{-5} subsampling				
NEG-5	14	61	58	60
NEG-15	36	61	61	**61**
HS-Huffman	21	52	59	55

Table 1: Accuracy of various Skip-gram 300-dimensional models on the analogical reasoning task as defined in [8]. NEG-k stands for Negative Sampling with k negative samples for each positive sample; NCE stands for Noise Contrastive Estimation and HS-Huffman stands for the Hierarchical Softmax with the frequency-based Huffman codes.

Evaluation(results)

Analogical reasoning task:

- syntactic analogy
- semantic analogy

Data

- 1b words;
- cut out infrequent words(<5t),
- they got |Voc| = 692K

Method	Time [min]	Syntactic [%]	Semantic [%]	Total accuracy [%]
NEG-5	38	63	54	59
NEG-15	97	63	58	**61**
HS-Huffman	41	53	40	47
NCE-5	38	60	45	53
The following results use 10^{-5} subsampling				
NEG-5	14	61	58	60
NEG-15	36	61	61	**61**
HS-Huffman	21	52	59	55

Table 1: Accuracy of various Skip-gram 300-dimensional models on the analogical reasoning task as defined in [8]. NEG-k stands for Negative Sampling with k negative samples for each positive sample; NCE stands for Noise Contrastive Estimation and HS-Huffman stands for the Hierarchical Softmax with the frequency-based Huffman codes.

Phrases

Data-driven approach to find the phrases
(words that appear frequently together and infrequently in other contexts)

$$\text{score}(w_i, w_j) = \frac{\text{count}(w_i w_j) - \delta}{\text{count}(w_i) \times \text{count}(w_j)}$$

δ is discounting coefficient

δ prevents phrases made of infrequent words

We talk about "Distributed Representations of Words and Phrases and their Compositionality" (Mikolov et al)

Phrases

Data-driven approach to find the phrases
(words that appear frequently together and infrequently in other contexts)

$$\text{score}(w_i, w_j) = \frac{\text{count}(w_i w_j) - \delta}{\text{count}(w_i) \times \text{count}(w_j)}$$

δ is discounting coefficient

δ prevents phrases made of infrequent words

Training

2-4 passes over data to form longer sequences

Demo for phrases

Evaluation of ‘Phrases’

New test set (3218, 5 categories only):

- Boston Boston_Celtics Miami Miami_Heat
- Werner_Vogels Amazon Samuel_J._Palmisano IBM
- Germany Lufthansa Spain Spanair
- Atlanta Atlanta_Thrashers Boston Boston_Bruins
- Boston Boston_Globe Seattle Seattle_Times

Evaluation of ‘Phrases’(result)

1 b, dim = 300, context= window-5

Method	Dimensionality	No subsampling [%]	10^{-5} subsampling [%]
NEG-5	300	24	27
NEG-15	300	27	42
HS-Huffman	300	19	**47**

Table 3: Accuracies of the Skip-gram models on the phrase analogy dataset. The models were trained on approximately one billion words from the news dataset.

Evaluation of ‘Phrases’(result)

1 b, dim = 300, context= window-5

Method	Dimensionality	No subsampling [%]	10^{-5} subsampling [%]
NEG-5	300	24	27
NEG-15	300	27	42
HS-Huffman	300	19	**47**

Table 3: Accuracies of the Skip-gram models on the phrase analogy dataset. The models were trained on approximately one billion words from the news dataset.

6 b, dim = 1000, context = sentence -> **accuracy 66%**

33 b, dim = 1000, context = sentence -> **accuracy 72%**

We talk about “Distributed Representations of Words and Phrases and their Compositionality” (Mikolov et al)

Evaluation of 'Phrases'(result)

1 b, dim = 300, context= window-5

Method	Dimensionality	No subsampling [%]	10^{-5} subsampling [%]
NEG-5	300	24	27
NEG-15	300	27	42
HS-Huffman	300	19	**47**

Table 3: Accuracies of the Skip-gram models on the phrase analogy dataset. The models were trained on approximately one billion words from the news dataset.

6 b, dim = 1000, context = sentence -> **accuracy 66%**

33 b, dim = 1000, context = sentence -> **accuracy 72%**

Hierarchical softmax and subsampling; amount of data is crucial

Additive compositionality

Czech + currency	Vietnam + capital	German + airlines	Russian + river	French + actress
koruna	Hanoi	airline Lufthansa	Moscow	Juliette Binoche
Check crown	Ho Chi Minh City	carrier Lufthansa	Volga River	Vanessa Paradis
Polish zolty	Viet Nam	flag carrier Lufthansa	upriver	Charlotte Gainsbourg
CTK	Vietnamese	Lufthansa	Russia	Cecile De

Table 5: Vector compositionality using element-wise addition. Four closest tokens to the sum of two vectors are shown, using the best Skip-gram model.

We talk about “Distributed Representations of Words and Phrases and their Compositionality” (Mikolov et al)

Empirical comparison with previous results

Model (training time)	Redmond	Havel	ninjutsu	graffiti	capitulate
Collobert (50d) (2 months)	conyers lubbock keene	plauen dzerzhinsky osterreich	reiki kohona karate	cheesecake gossip dioramas	abdicate accede rearm
Turian (200d) (few weeks)	McCarthy Alston Cousins	Jewell Arzu Ovitz	- - -	gunfire emotion impunity	- - -
Mnih (100d) (7 days)	Podhurst Harlang Agarwal	Pontiff Pinochet Rodionov	- - -	anaesthetics monkeys Jews	Mavericks planning hesitated
Skip-Phrase (1000d, 1 day)	Redmond Wash. Redmond Washington Microsoft	Vaclav Havel president Vaclav Havel Velvet Revolution	ninja martial arts swordsmanship	spray paint grafitti taggers	capitulation capitulated capitulating

Table 6: Examples of the closest tokens given various well known models and the Skip-gram model trained on phrases using over 30 billion training words. An empty cell means that the word was not in the vocabulary.

Conclusion

Distributed vector representation
can capture a large number of precise
syntactic and semantic word relationships

Conclusion

Distributed vector representation
can capture a large number of precise
syntactic and semantic word relationships

more regular word representations

improved training speed

new approach called Negative sampling

new approach called Subsampling

Conclusion(in details)

The **hyper-parameter choice** is crucial for **performance** (both speed and accuracy)

The main choices to make are:

architecture: skip-gram (slower, better for infrequent words) vs CBOW (fast)

the training algorithm:
hierarchical softmax (better for infrequent words)
vs
negative sampling (better for frequent words, better with low dimensional vectors)

sub-sampling of frequent words: can improve both accuracy and speed for large data sets (useful values are in range 1e-3 to 1e-5)

dimensionality of the word vectors: usually more is better, but not always

context (window) size: for skip-gram usually around 10, for CBOW around 5

https://code.google.com/p/word2vec/

What was after the paper?

What was after the paper?

A lot!

What was after the paper?

A lot!

Names to follow:
Socher, Manning, Omer Levy, Yoav Goldberg…

Why does this produce good word representations?
Good question. We don't really know (Levy, Goldberg, 2014)

Resourceful links

word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method

Richard Socher lecture and course

Hierarchical softmax in neural network language model

Linguistic Regularities in Sparse and Explicit Word Representations

Short tutorial about word2vec in Python

Distributed Representations of Sentences and Documents: Doc2vec(Paragraph2Vec)

We talk about "Distributed Representations of Words and Phrases and their Compositionality" (Mikolov et al)

Thank you