UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Efficient Estimation of Word Representations in Vector Space

by Tomos Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean

Published on 7 Sep 2013

ECE 594 Paper Presentation

Hanyin Shao

Feb 10, 2022

Background

Language modeling (LM) is the use of various statistical and probabilistic techniques (e.g. word representations) to determine the probability of a given sequence of words occurring in a sentence.

These techniques are used in various NLP applications such as machine translation, question answering, sentiment analysis, etc.

Example
Where are we ___?

History

~1948 – Birth of **N-Grams**

1986 – the first ideas of **representing words as vectors** by Hinton et al.

Recurrent Neural Network (RNN) by Rumelhart et al.

1997 – **Long Short-Term Memory networks (LSTM)** by Hochreiter et al.

2003 – the first **neural network language model (NNLM)** by Bengio et al.

2013 – Birth of Widespread Pretrained Word Embeddings (**Word2Vec**) by Mikolov et al.

2014 – **GloVe: Global Vectors for Word Representation** by Pennington et al.

2017 – **BERT: Pre-training of Deep Bidirectional Transformers** by Vaswani et al.

......

Limitations of previous work

Distributional Representations

- Treat words as atomic units – there is no notion of similarity between words (n-grams)
- **Latent Semantic Analysis (LSA):** not good at preserving linear regularities

  Vectorized form of words should follow linear additive properties
  e.g. vec(apparent) - vec(apparently) + vec(rapid) => vec(rapidly)

- **Latent Dirichlet Allocation (LDA):** computationally very expensive on large data sets

Distributed Representations

- **Feedfoward Neural Net Language Model (NNLM) and Recurrent Neural Net Language Model (RNNLM):** unable to be trained on more than a few hundred of millions of words (computationally expensive).

Goal of this paper

- Learn **high-quality** word vectors from **huge data sets** (billions of words and millions of words in the vocabulary)

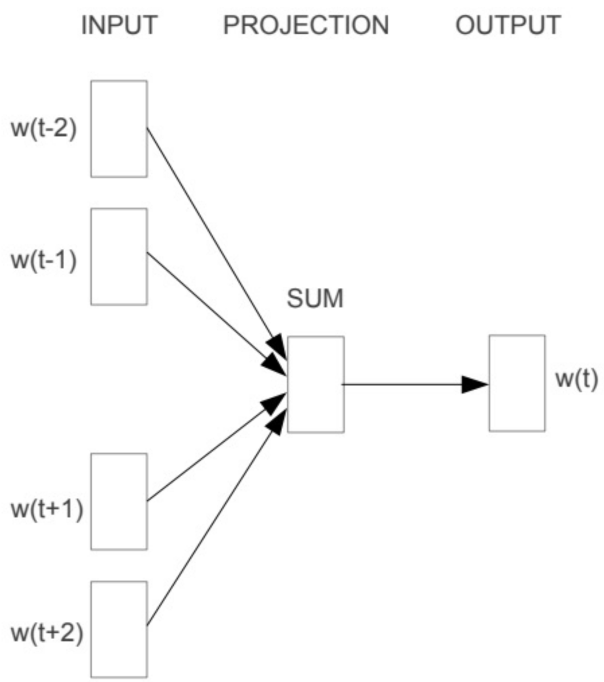
- Similar words should tend to be close to each other and words can have **multiple degrees of similarity**

  "car" and "bus" are semantically similar
  "walked" and "swam" are syntactically similar

- Maximize accuracy of vector operations by developing new model architectures that preserve the linear regularities

  vector("King") - vector("Man") + vector("Woman") closest to vector("Queen")

Continuous Bag-of-Words Model (CBOW)



CBOW

Predict the current word based on the context

Input: word vectors of context words

Output: probabilities of all words in the vocabulary appearing at the current position

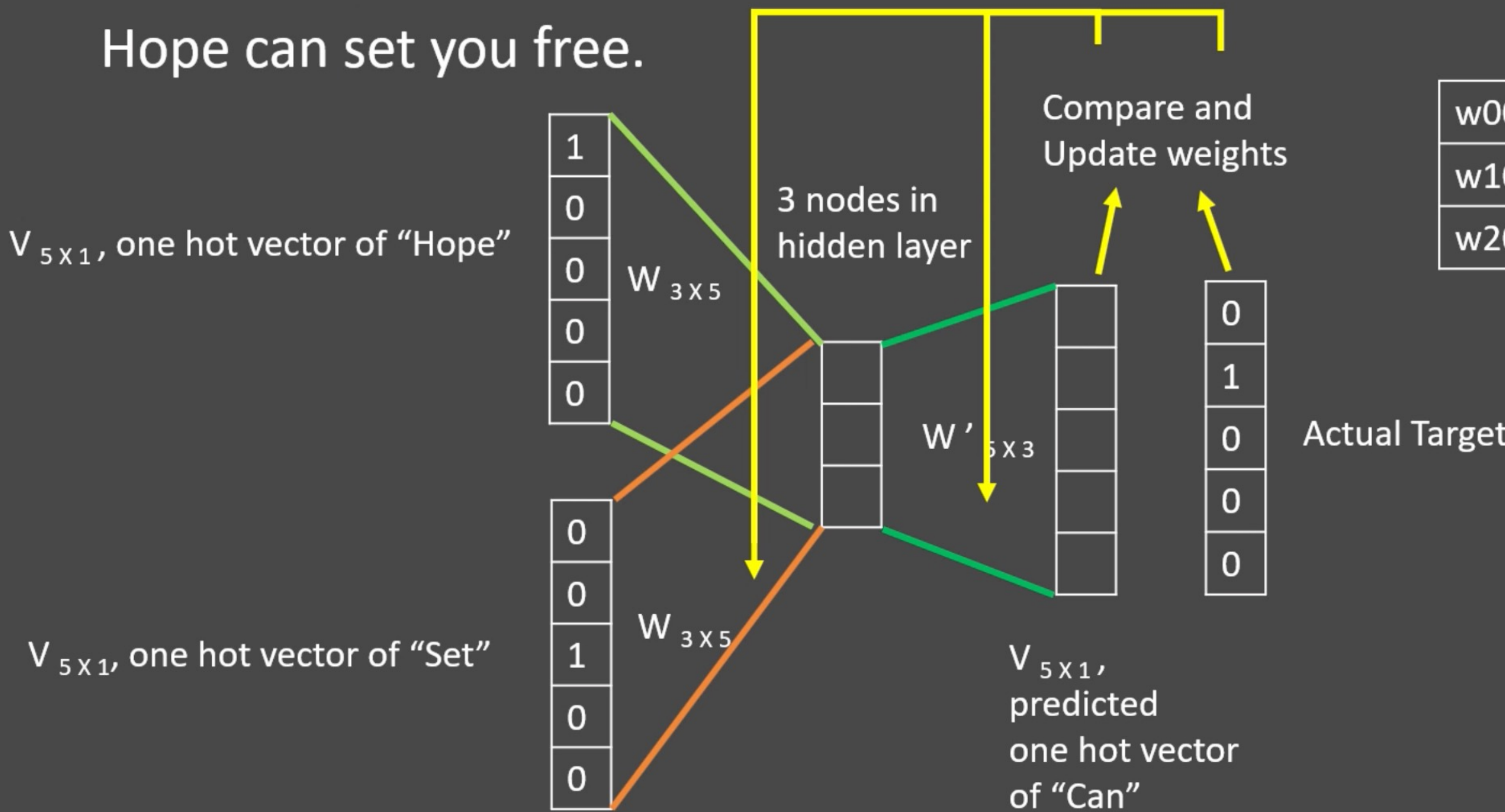
Objective: maximize the probabilities of the word in the training set appearing at this position

Example

... two novel model architectures for computing continuous vector representations of words ...

CBOW - Working

Hope can set you free.

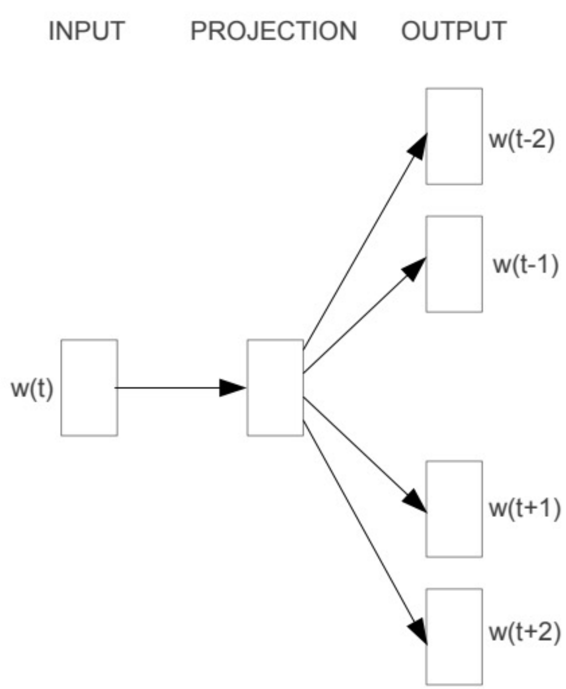


w00	w01	w02	w03	w04
w10	w11	w12	w13	w14
w20	w21	w22	w23	w24

$W_{3 \times 5}$

www.youtube.com/thesemicolon

Continuous Skip-gram Model (Skip-gram)



Skip-gram

Predict surrounding words given the current word

Input: word vectors of the current word

Output: probabilities of all words in the vocabulary appearing at the surrounding positions

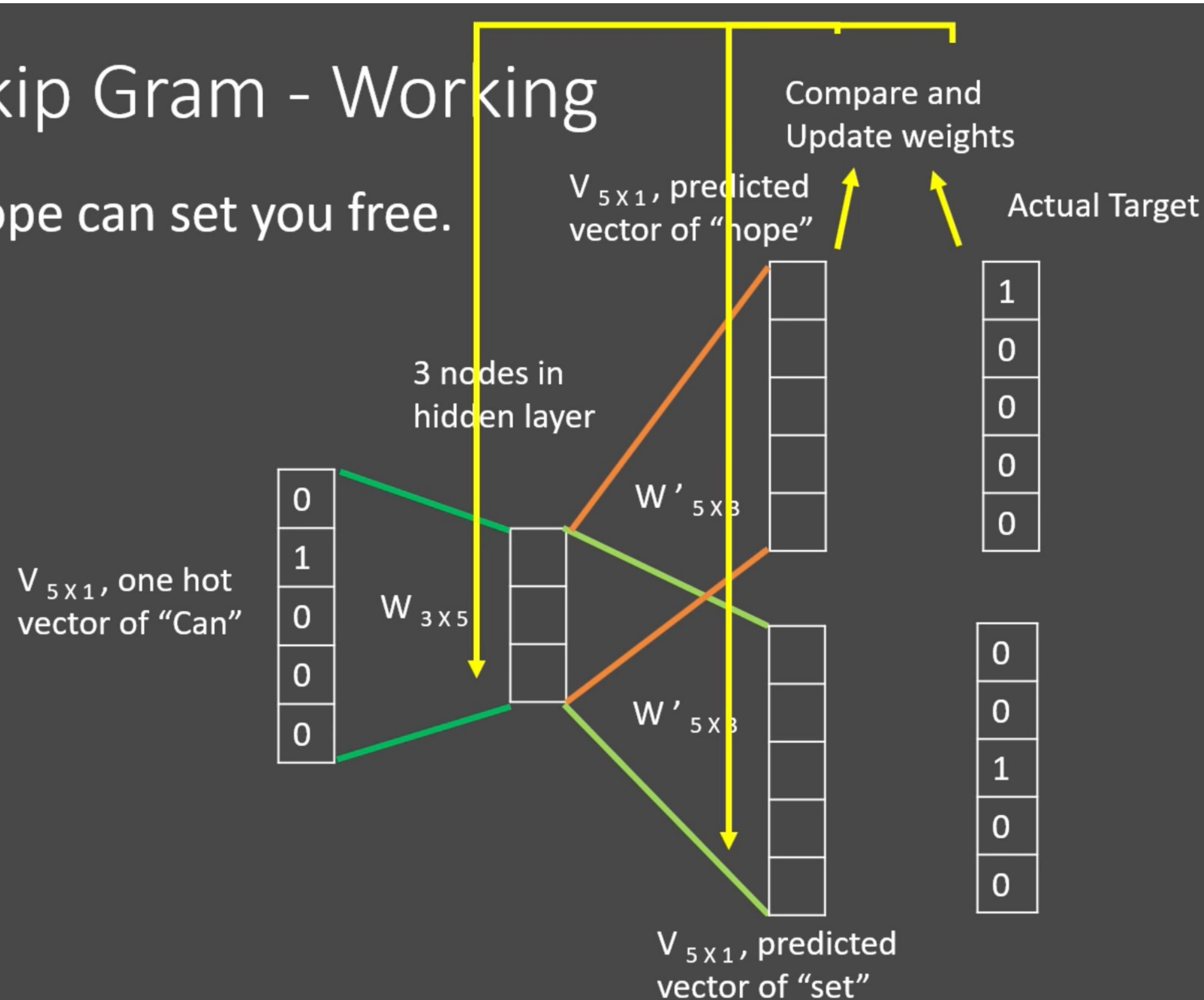
Objective: maximize the probabilities of words in the training set appearing in the contexts

Example

... two novel model architectures for computing continuous vector representations of words ...

Skip Gram - Working

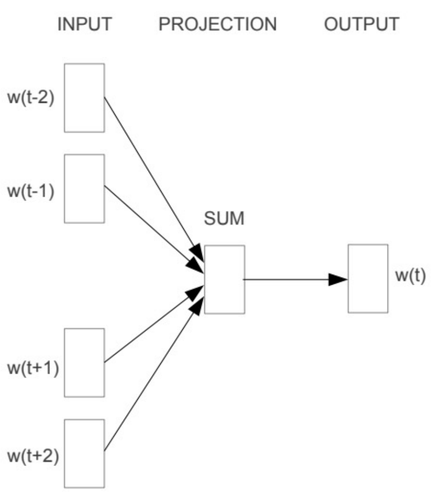
Hope can set you free.



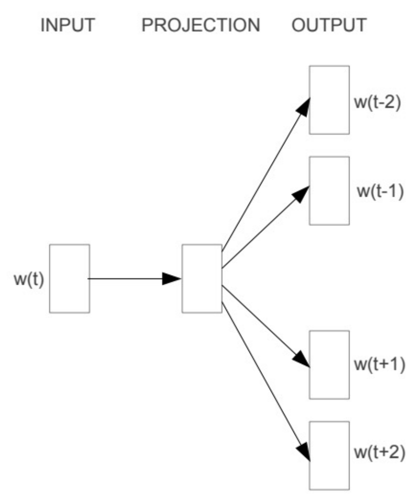
www.youtube.com/thesemicolon

Prediction

Predict from model



CBOW



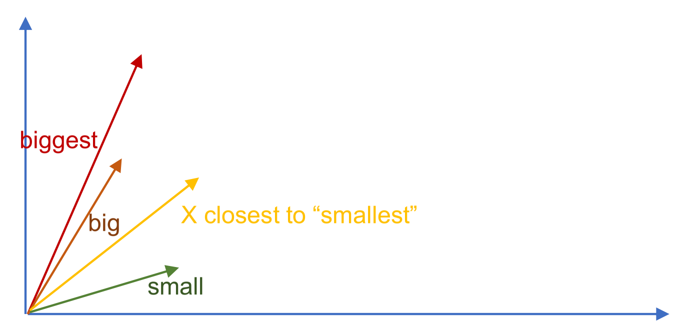
Skip-gram

Predict from word vectors

X : small :: biggest : big
X = ?

Vector Computation:
X = vec("biggest") – vec("big") + vec("small").

Then search in the vector space for the word closest to X measured by cosine distance.



Results – Task Description

5 types of semantic questions
9 types of syntactic questions

Example:
Chicago : Illinois :: Stockton : X
Chicago : X :: Stockton : California
…

Predict X

Accuracy:
Question is assumed to be correctly answered only if the closest word to the vector computed is exactly the same as the correct word in the question.

Table 1: *Examples of five types of semantic and nine types of syntactic questions in the Semantic-Syntactic Word Relationship test set.*

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

Results – Maximization of Accuracy

Corpus: Google News

Training epochs: 3

Stochastic gradient descent and backpropagation

Learning rate: 0.025 and decreased linearly till zero

Table 2: *Accuracy on subset of the Semantic-Syntactic Word Relationship test set, using word vectors from the CBOW architecture with limited vocabulary. Only questions containing words from the most frequent 30k words are used.*

Dimensionality / Training words	24M	49M	98M	196M	391M	783M
50	13.4	15.7	18.6	19.1	22.5	23.2
100	19.4	23.1	27.8	28.7	33.4	32.2
300	23.2	29.2	35.3	38.6	43.7	45.9
600	24.0	30.1	36.5	40.8	46.6	50.4

Observation:

After some point, adding more dimensions or adding more training data provides diminishing improvements.

So, we have to increase both vector dimensionality and the amount of the training data together.

Results – Comparison of Models

Table 3: *Comparison of architectures using models trained on the same da word vectors. The accuracies are reported on our Semantic-Syntactic W and on the syntactic relationship test set of [20]*

Model Architecture	Semantic-Syntactic Word Relationship test set		M
	Semantic Accuracy [%]	Syntactic Accuracy [%]	
RNNLM	9	36	
NNLM	23	53	
CBOW	24	64	
Skip-gram	55	59	

Table 4: *Comparison of publicly available word vectors on the Semantic-Syntactic Word Relationship test set, and word vectors from our models. Full vocabularies are used.*

Model	Vector Dimensionality	Training words	Accuracy [%]		
			Semantic	Syntactic	Total
Collobert-Weston NNLM	50	660M	9.3	12.3	11.0
Turian NNLM	50	37M	1.4	2.6	2.1
Turian NNLM	200	37M	1.4	2.2	1.8
Mnih NNLM	50	37M	1.8	9.1	5.8
Mnih NNLM	100	37M	3.3	13.2	8.8
Mikolov RNNLM	80	320M	4.9	18.4	12.7
Mikolov RNNLM	640	320M	8.6	36.5	24.6
Huang NNLM	50	990M	13.3	11.6	12.3
Our NNLM	20	6B	12.9	26.4	20.3
Our NNLM	50	6B	27.9	55.8	43.2
Our NNLM	100	6B	34.2	**64.5**	50.8
CBOW	300	783M	15.5	53.1	36.1
Skip-gram	300	783M	**50.0**	55.9	**53.3**

Observation

Semantic Tasks: Skip-gram > CBOW >= NNLM > RNNLM

Syntactic Tasks: CBOW > Skip-gram > NNLM > RNNLM

Results – Comparison of Models

Table 5: *Comparison of models trained for three epochs on the same data and models trained for one epoch. Accuracy is reported on the full Semantic-Syntactic data set.*

Model	Vector Dimensionality	Training words	Accuracy [%]			Training time [days]
			Semantic	Syntactic	Total	
3 epoch CBOW	300	783M	15.5	53.1	36.1	1
3 epoch Skip-gram	300	783M	50.0	55.9	53.3	3
1 epoch CBOW	300	783M	13.8	49.9	33.6	0.3
1 epoch CBOW	300	1.6B	16.1	52.6	36.1	0.6
1 epoch CBOW	600	783M	15.4	53.3	36.2	0.7
1 epoch Skip-gram	300	783M	45.6	52.2	49.2	1
1 epoch Skip-gram	300	1.6B	52.2	55.1	53.8	2
1 epoch Skip-gram	600	783M	56.7	54.5	55.5	2.5

Observation

Training a model on twice as much data using one epoch gives comparable or better results than iterating over the same data for three epochs and provides additional small speedup.

Results – Comparison of Models

Table 6: *Comparison of models trained using the DistBelief distributed framework. Note that training of NNLM with 1000-dimensional vectors would take too long to complete.*

Model	Vector Dimensionality	Training words	Accuracy [%]			Training time [days x CPU cores]
			Semantic	Syntactic	Total	
NNLM	100	6B	34.2	64.5	50.8	14 x 180
CBOW	1000	6B	57.3	68.9	63.7	2 x 140
Skip-gram	1000	6B	66.1	65.1	65.6	2.5 x 125

Observation

Computational Complexity: NNLM >> Skip-gram > CBOW

Results – Learned Relationships

Table 8: *Examples of the word pair relationships, using the best word vectors from Table 4 (Skip-gram model trained on 783M words with 300 dimensionality).*

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

Summary

Two novel model architectures (CBOW and Skip-gram) for computing word vectors/embeddings

Highlight
- High-quality word vectors which perform well on both syntactic and semantic questions.
- Low computational complexity.
- CBOW performs better on syntactic tasks. Skip-gram performs better on semantic tasks and has better overall accuracy.

Limitation
- Cannot handle out-of-vocabulary words.
- Learned static embeddings for each word, i.e. the same word under two different contexts will have the same embeddings.
- Ordering of words within a text is not considered in the CBOW model.
- The evaluation task cannot prove the word embeddings can be helpful to other NLP tasks.
- Learned relationships can have bias.
- …

Discussion