Efficient Estimation of Word Representations in Vector Space

Authors: Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean
Presenter: Feng Lin & Aidan Groves

Outline

1. Introduction
2. Existing Models
3. Proposed Models
4. Results

Limitation of traditional system

1. view words as atomic units without inter-relationship
2. data is scarce or not diverse enough [Quantity]
3. the availability of high-quality transcribed speech data [Quality]

Simple Solution - Scaling by increase N in n-grams

- the number of possible sequences grows exponentially [Computation]
- many combinations occur infrequently or not at all in the training data [Sparse]
- keep increasing N will not always be helpful [diminishing returns]
- overfitting - cannot be generalized …

Novel Methods needed

Goal of the paper

- develop model architectures for learning high-quality word vectors from large datasets
- **Multi-Degrees of Similarity**
    - dog & puppy; cat & kitten … [meaning/semantic]
    - noun+ing; noun+s … [syntactic]
    - King - Man + Woman = Queen [“algebraic” operations]
- remove non-linear hidden layer
    - keep “linear relationship”
    - reduce complexity

Computational complexity

$$O = E \times T \times Q$$

where

E = number of training epochs (3-50)

T = number of the words in the training set

Q = complexity per training sample (defined later)

Train on **Stochastic Gradient Descent and Backpropagation**

Feedforward Neural Net Language Model (NNLM)

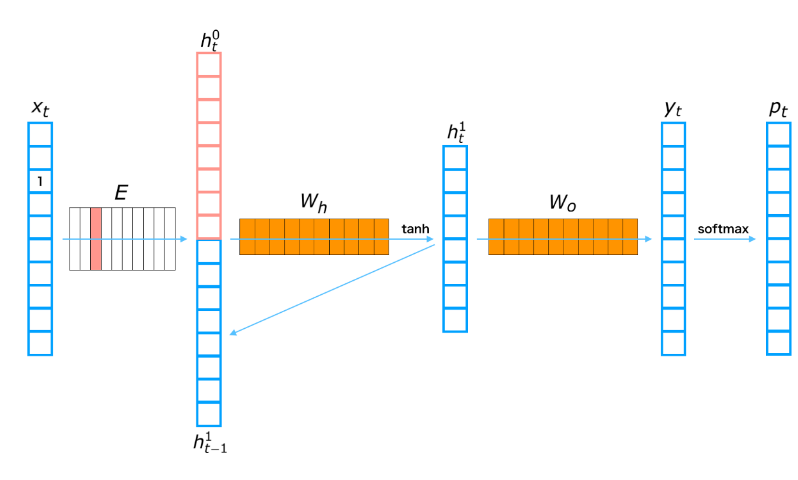
- we have V words in vocabulary
- encoding input N words using a "1-of-V" coding method
    - each word is represented by a unique vector where only one element is 1, and all others are 0
    - high dimensional sparse dataset
- project into a projection layer P with dimensionality N×D
- Hidden Layer with size H
- Output Layer: The output is a probability distribution over the vocabulary, computed from the hidden layer, thus having a dimensionality of V

$$Q = N * D + N * D * H + H * V$$

Recurrent Neural Net Language Model (RNNLM)

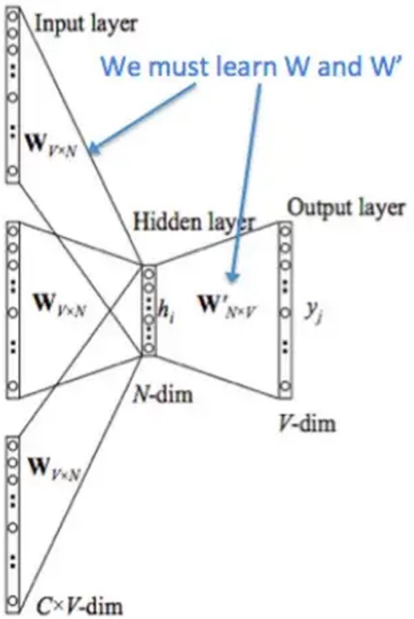
- does not have projection layer
- short term memory - time delayed connection
    - utilize both input layer x_t and hidden layer output h_{t-1} from last time step to get h_{t}

$$h_t = \theta(W_{hh} * h_{t-1} + W_{dh} * x_t + b_t)$$



Proposed Model: Continuous Bag-of-Words Model

- remove non-linear hidden layer
- the projection layer is shared for all words - order of words are not considered
- utilize future and history to classify the current



$$Q = N * D + D * log_2(V)$$

An example of CBOW Model

Corpus = { I drink coffee everyday }

$W^{I} = [1,0,0,0]$

$W^{drink} = [0,1,0,0]$

target $\boxed{W^{coffee} = [0,0,1,0]}$

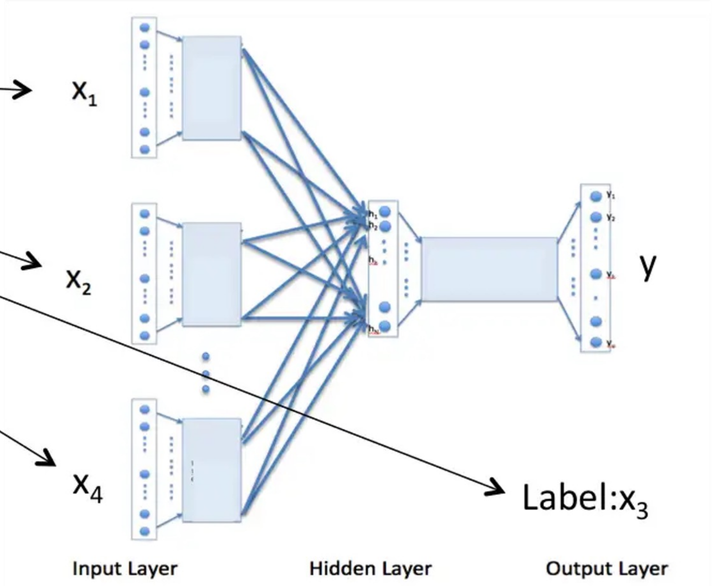
$W^{everyday} = [0,0,0,1]$

Set:

Window size: 2

Target word: coffee

Context word: I, drink, everyday



An example of CBOW Model

Corpus = { I drink coffee everyday }

Initialize:

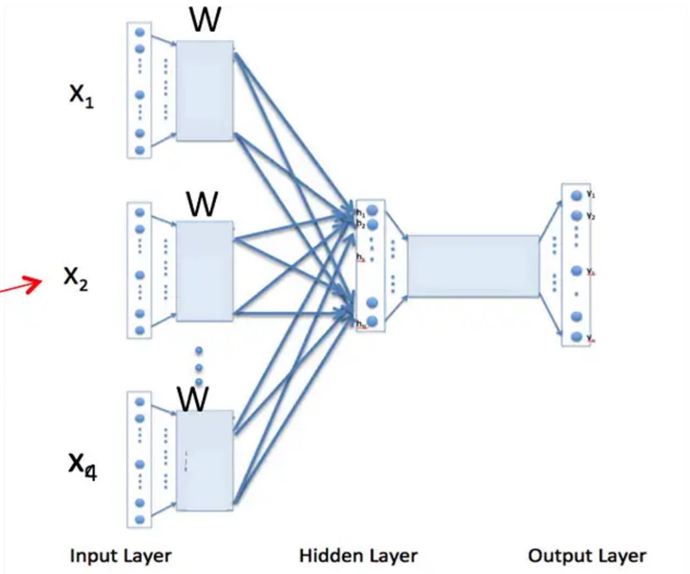
$$W = \begin{bmatrix} 1 & 2 & 3 & 0 \\ 1 & 2 & 1 & 2 \\ -1 & 1 & 1 & 1 \end{bmatrix}$$

Ex:

W^{drink} = [0,1,0,0]

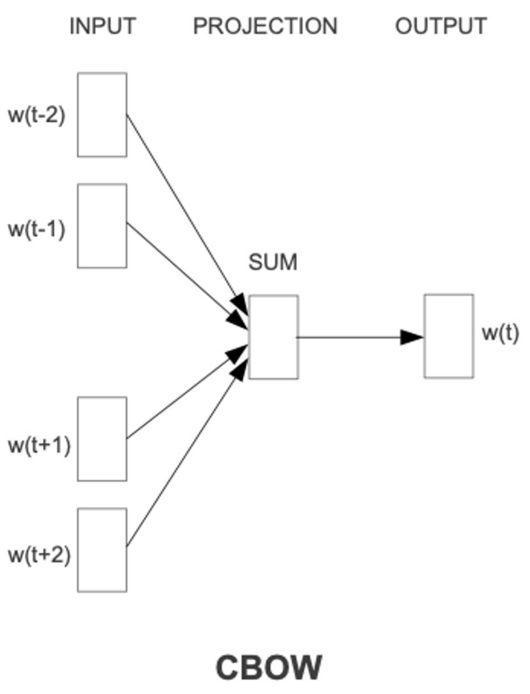
$$Wx_2 = v_2$$

$$\begin{bmatrix} 1 & \boxed{2} & 3 & 0 \\ 1 & \boxed{2} & 1 & 2 \\ -1 & \boxed{1} & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix}$$



Continuous bag-of-words (Mikolov et al., 2013)

Architecture



Why better

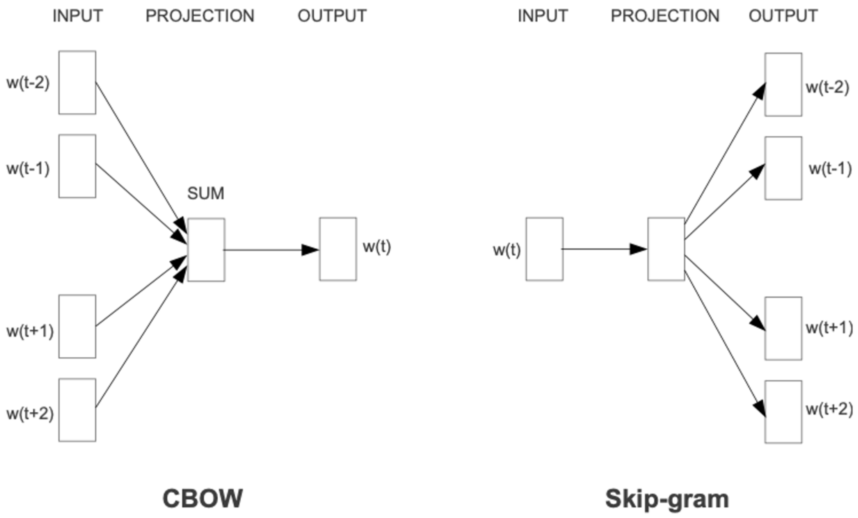
1. simpler architecture
2. contextual awareness - classifying the middle word
3. shared projection matrix - a common representation for all words (X order)
4. continuous distributed representation of context

Proposed Model - Continuous Skip-gram Model

- based on a word, predict words around it in the same sentence

CBOW: The cat ate _____. Fill in the blank, in this case, it's "food".

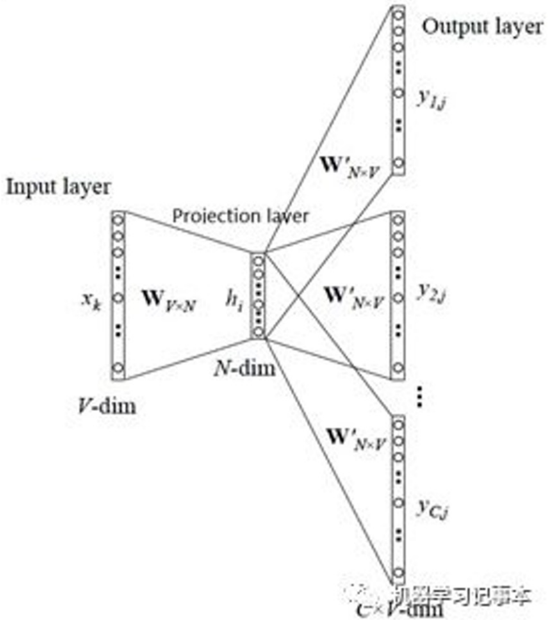
Skip-gram: ___ ___ ___ food. Complete the word's context. In this case, it's "The cat ate"



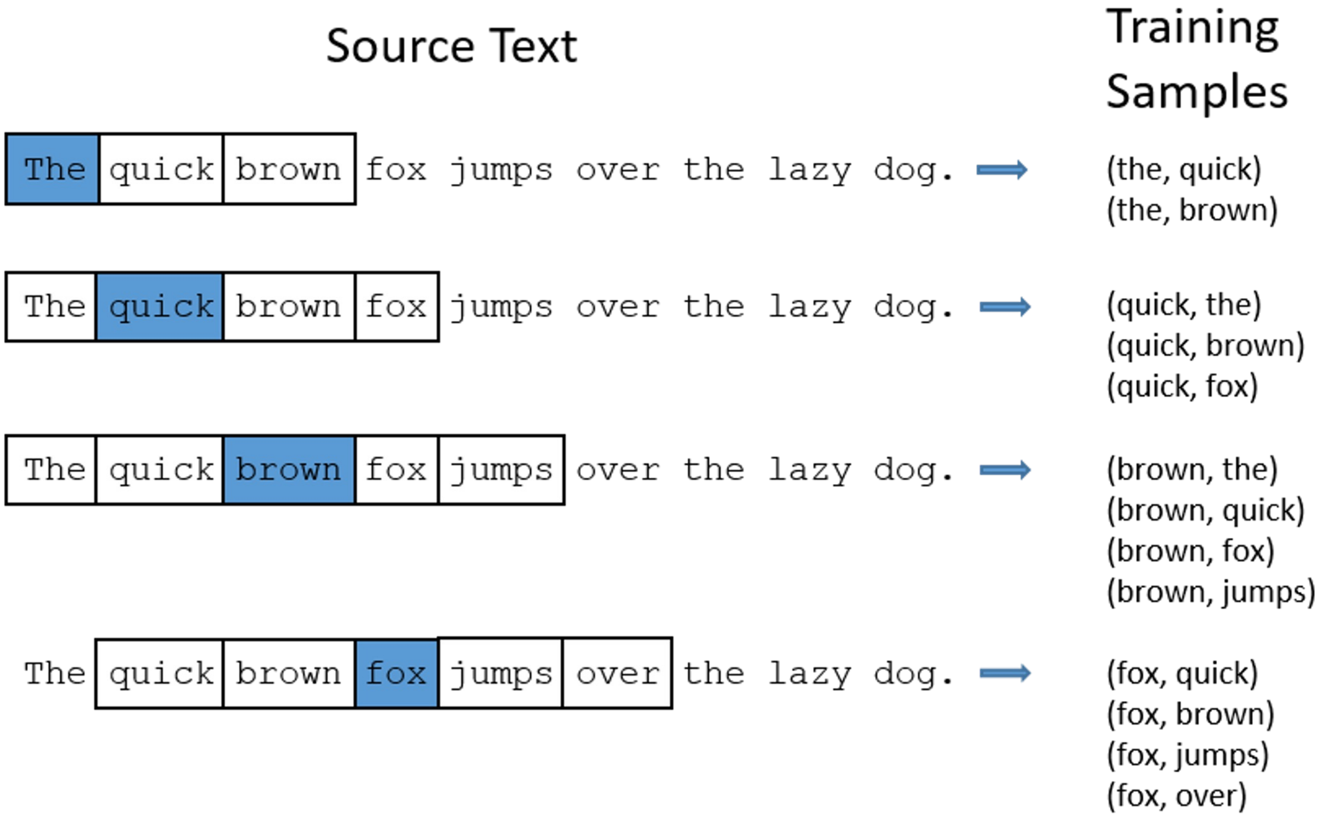
Proposed Model - Continuous Skip-gram Model

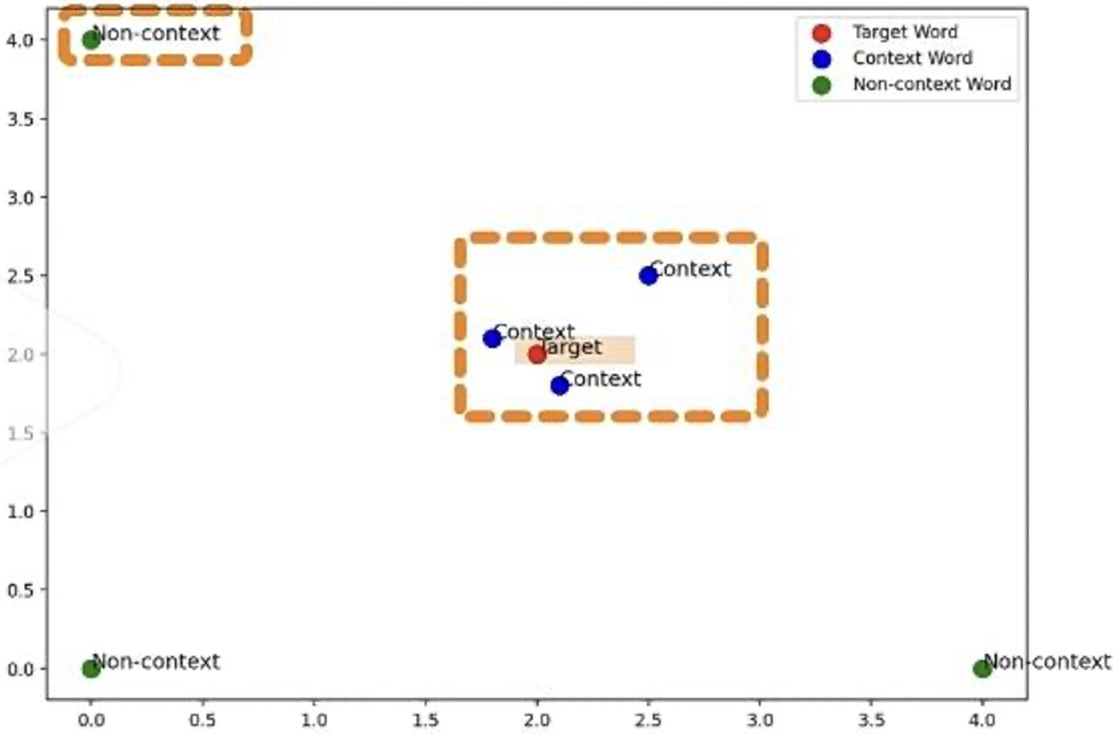
- C as maximum distance of the words
- R is a random number in the interval [1, C]
- therefore, we have to do 2*R word predictions for each sample

$$Q = C * (D + D * log_2(V))$$



- R is random so distant words are sampling less
- Positional Embedding
- Adjust the loss function
- Adjust the softmax function


Source Text
Training Samples
The quick brown fox jumps over the lazy dog.
(the, quick)
(the, brown)
The quick brown fox jumps over the lazy dog.
(quick, the)
(quick, brown)
(quick, fox)
The quick brown fox jumps over the lazy dog.
(brown, the)
(brown, quick)
(brown, fox)
(brown, jumps)
The quick brown fox jumps over the lazy dog.
(fox, quick)
(fox, brown)
(fox, jumps)
(fox, over)


Target Word
Context Word
Non-context Word
Non-context
Non-context
Non-context
Context
Context
Target
Context
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Result

- a comprehensive test set contains the following types of relationship in Question forms
- metrics - accuracy for all question types and for each type separately
- synonyms are counted as mistakes

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

Maximizing Accuracy

- Diminishing Marginal Return - increase D or increase training data
- E = 3, learning rate = 0.025 and decrease it linearly

Model Architecture	Semantic-Syntactic Word Relationship test set		MSR Word Relatedness Test Set [20]
	Semantic Accuracy [%]	Syntactic Accuracy [%]	
RNNLM	9	36	35
NNLM	23	53	47
CBOW	24	64	61
Skip-gram	55	59	56

- NNLM perform better than RNNLM
  - word representation vector directly connect to non-linear hidden layer without projection layer
- CBOW better than NNLM
- Skip-gram slightly worse on syntactic but better on semantic compare to CBOW

Maximize Accuracy

- Real-World Applicability and Validation
  - compare the models trained on a single CPU against publicly available word vectors.

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**

- train a model on twice as much data using one epoch gives better results than iterating over the same data for three epochs and provides additional speedup

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

use Ada-grad - adaptive learning rate for each dimension

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

Microsoft Sentence Completion Challenge

$$model = \beta * Skip - gram + (1 - \beta) * RNNLMs$$

Table 7: *Comparison and combination of models on the Microsoft Sentence Completion Challenge.*

Architecture	Accuracy [%]
4-gram [32]	39
Average LSA similarity [32]	49
Log-bilinear model [24]	54.8
RNNLMs [19]	55.4
Skip-gram	48.0
Skip-gram + RNNLMs	**58.9**

Problems

- Bias in dataset
  - King - Man + Woman = Queen (from earlier slide)
  - Doctor - man + woman = Nurse

Thank You!