Word Embeddings

Introduction

- Why vector for Natural language ?
- Convensional representations for words and documents
- Methods of Dimensionality reduction

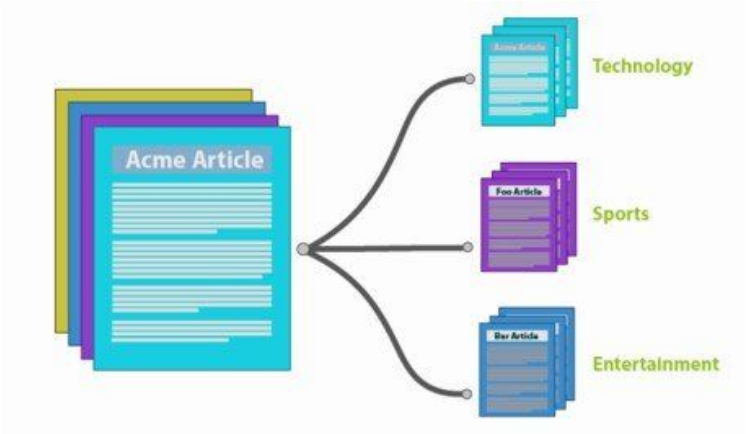
Deep learning models:

- Continuous Bag of words model
- Other Models (SKip Gram Model, GloVe)
- Evaluation of Word Vectors
- Readings and references

Introduction: Why Vectors

Document Classification or Clustering :

- Documents composed of words
- Similar documents will contain similar words
- Machine Learning love vectors
- A Machine Learning algorithm shall know which words are significant which category



Bag of Words Model

"Represent each document which the bag of words it contains"

d1 : Mary loves Movies, Cinema and Art

d2 : John went to the Football game

d3 : Robert went for the Movie Delicatessen

Class 1 : Arts

Class 2 : Sports

Class : Arts

	Mary	Loves	Movies	Cinema	Art	John	Went	to	the	Delicatessen	Robert	Football	Game	and	for
d1	1	1	1	1	1									1	
d2						1	1	1	1			1	1		
d3			1				1		1		1				1

Bag of Words Model

Can a Machine learning algorithm know that “the” and “for” are un important words ?

- Yes : But will need lots of training labeled data

What to do ?

- Use hand crafted features (weighting features for words)
- Make lots of them
- Keep doing this for 50 years
- Regret later .. cry hard

Bag of Words Model + Weghting eiFeatures

Weighting features example TF-IDF

- TF-IDF ~= Term Frequency / Document frequency
- Motivation : Words appearing in large number of documents are not significant

	Mary	Loves	Movies	Cinema	Art	John	Went	to	the	Delicatessen	Robert	Football	Game	and	for
d1	0.3779	0.3779	0.3779	0.3779	0.3779									0.0001	
d2							0.4402	0.001	0.02			0.4558	0.458		
d3			0.001				0.01		0.01		0.458				0.0001

Word Vector Representations

Document can be represented by words, But how to represent words themselves ?

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

John Rupert Firth

Word Vector Representations

Use a sliding window over a big corpus of text and count word co-occurences in between.

1. I enjoy flying.

2. I like NLP.

3. I like deep learning.

$$X = \begin{array}{c} \\ I \\ like \\ enjoy \\ deep \\ learning \\ NLP \\ flying \\ . \end{array} \begin{array}{c} \begin{array}{cccccccc} I & like & enjoy & deep & learning & NLP & flying & . \end{array} \\ \begin{bmatrix} 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Bag of words Representations: Drawbacks

- High dimensionality and Very sparse !!!!!
- Unable to capture word order
  - “ good but expensive” “expensive but good” will have same representation.
- Unable to capture semantic similarities (mostly because of sparsity)
  - “boy”, “girl” and “car”
  - “Human”, “Person” and “Giraffe”

Dimensionality Reduction using Matrix factorization

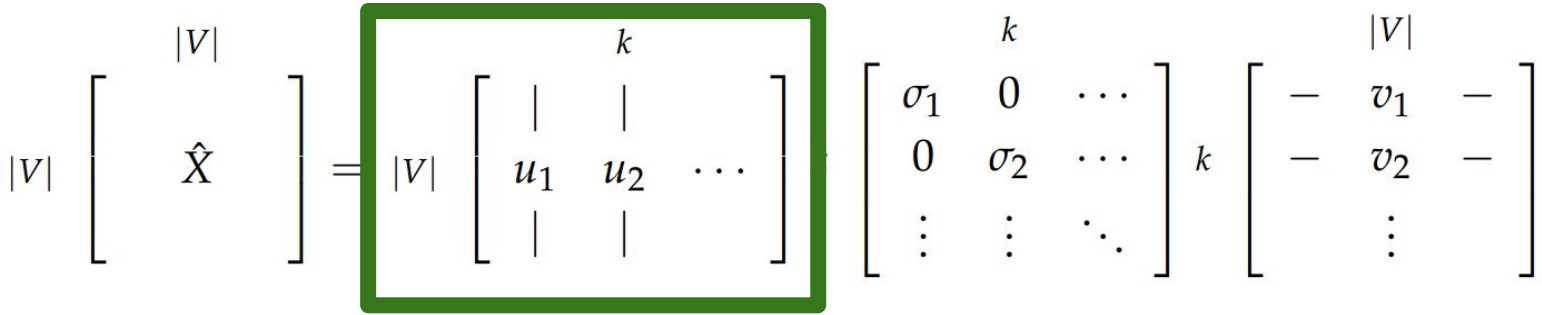
Singular value decomposition

$$\underset{|V|}{\underset{|V|}{X}} = \underset{|V|}{\underset{|V|}{\begin{bmatrix} | & | & \\ u_1 & u_2 & \cdots \\ | & | & \end{bmatrix}}} \underset{|V|}{\underset{|V|}{\begin{bmatrix} \sigma_1 & 0 & \cdots \\ 0 & \sigma_2 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}}} \underset{|V|}{\underset{|V|}{\begin{bmatrix} - & v_1 & - \\ - & v_2 & - \\ & \vdots & \end{bmatrix}}}$$

$$\underset{|V|}{\underset{|V|}{\hat{X}}} = \underset{|V|}{\underset{k}{\begin{bmatrix} | & | & \\ u_1 & u_2 & \cdots \\ | & | & \end{bmatrix}}} \underset{k}{\underset{k}{\begin{bmatrix} \sigma_1 & 0 & \cdots \\ 0 & \sigma_2 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}}} \underset{k}{\underset{|V|}{\begin{bmatrix} - & v_1 & - \\ - & v_2 & - \\ & \vdots & \end{bmatrix}}}$$

where : $\sigma_1 > \sigma_2 \; .. \; > \sigma_n > 0$

Singular value decomposition



- Lower dimensionality K << |V|
- taking the most significant projection of your vectors space

Latent semantic Indexing / Analysis (1994)

$$
\begin{array}{ccccccc}
 & X & & U & & \Sigma & V^T \\
 & (\mathbf{d}_j) & & & & & (\hat{\mathbf{d}}_j) \\
 & \downarrow & & & & & \downarrow \\
(\mathbf{t}_i^T) \rightarrow & \begin{bmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,n} \end{bmatrix} & = & (\hat{\mathbf{t}}_i^T) \rightarrow & \begin{bmatrix} \begin{bmatrix} \\ \mathbf{u}_1 \\ \\ \end{bmatrix} \dots \begin{bmatrix} \\ \mathbf{u}_l \\ \\ \end{bmatrix} \end{bmatrix} & \cdot \begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_l \end{bmatrix} \cdot \begin{bmatrix} [& \mathbf{v}_1 &] \\ & \vdots & \\ [& \mathbf{v}_l &] \end{bmatrix}
\end{array}
$$

∪ : are dense word vector representations

V : are dense Document vector representations

LSA / LSI , HAL methods made huge advancements in document retrieval and semantic similarity

Deep learning Word Embeddings (2003)

"A Neural Probabilistic Language Model" Bengio et al. 2003

Original task "Language Modeling" :

- Prediction of next word given sequence of previous words.
- Useful in Speech Recognition, Autcompletion, Machine translation.

"The Cat Chills on a *mat* " , *Calculate : P(mat | the, cat, chills, on, a)*

Deep learning Word Embeddings (2003)

"A Neural Probabilistic Language Model" Bengio et al. 2003

Quoting from the paper:

"This is intrinsically difficult because of the **curse of dimensionality**: a word sequence on which the model will be tested is likely to be different from all the word sequences seen during training."

"We propose to fight the curse of dimensionality by **learning a distributed representation for words**"

Continuous Bag of Words model (CBOW)
Tomas Mikolov et al. (2013)

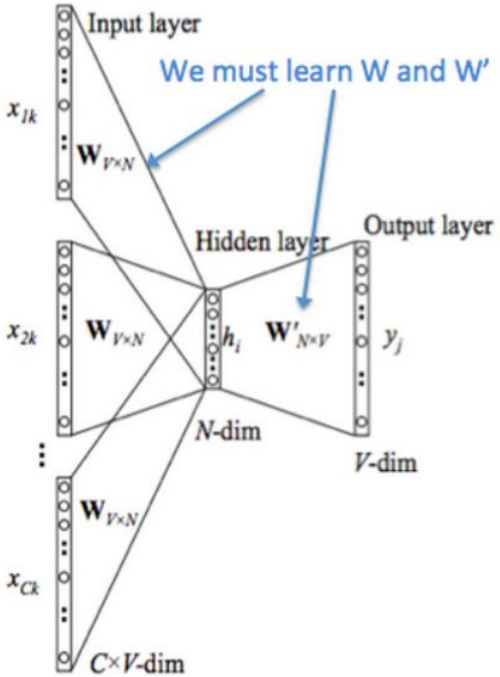
The model Predicts the current word given the context

scan text in large corpus with a window

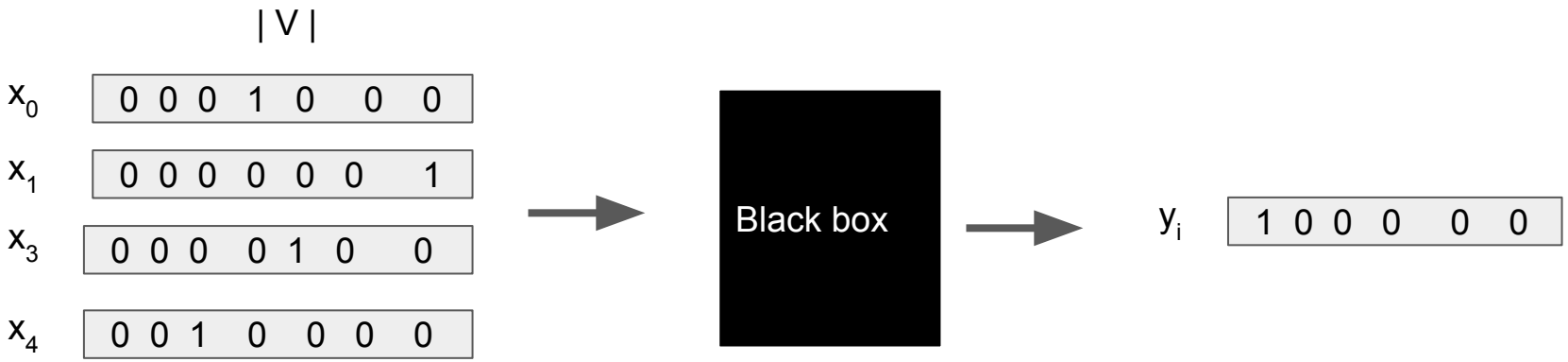
Input : x_0 , x_1 , x_3 , x_4 output : x_2

“ **The Cat Chills on a mat** ”

x_0 x_1 x_2 x_3 x_4 x_5



Continuous Bag of Words model (CBOW)(2013)

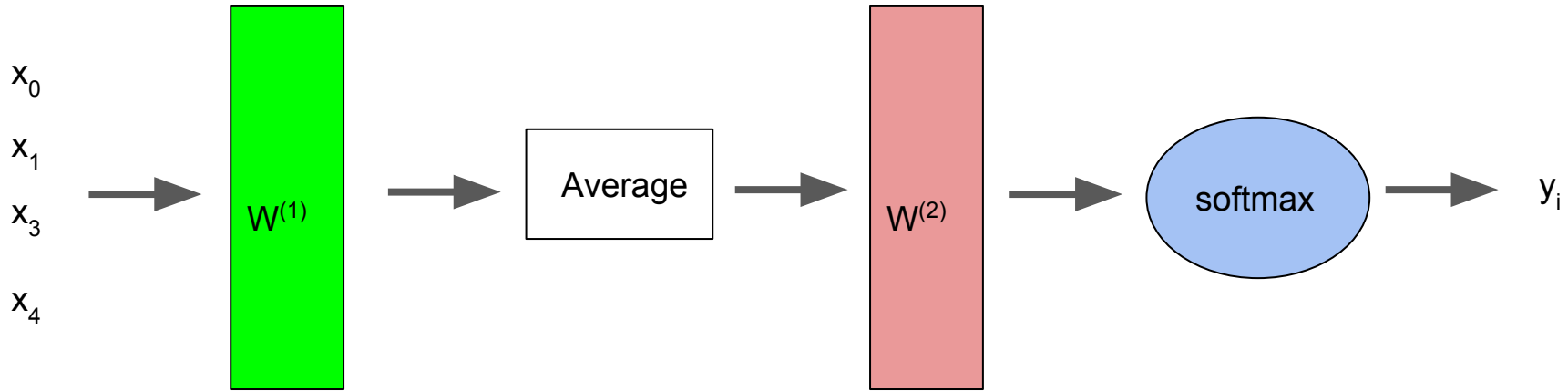


$|V|$ vocabulary size

$X_i \in R^{1 \times |V|}$ 1 hot vector representation of each word

$y_i \in R^{|V| \times 1}$ one hot representation of the correct middle word (expected output)

Continuous Bag of Words model (CBOW)(2013)

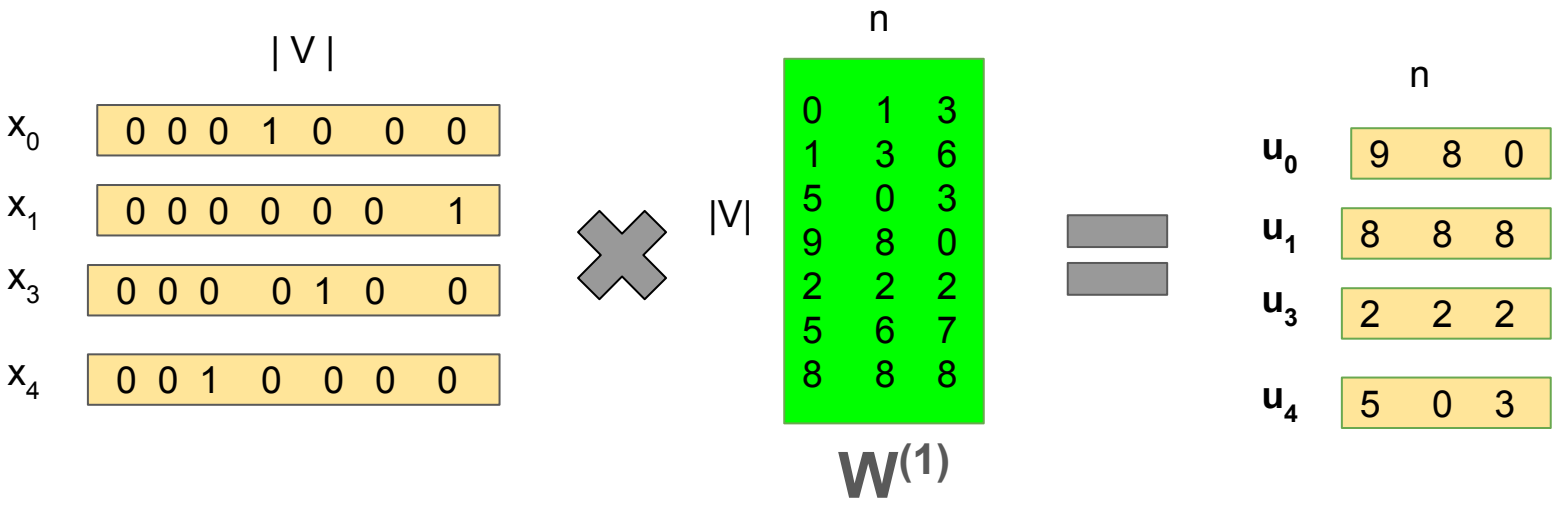


$|V|$ vocabulary size

$X_i \in R^{1 \times |V|}$ 1 hot vector representation of each word

$y_i \in R^{|V| \times 1}$ one hot representation of the correct middle word (expected output)

Continuous Bag of Words model (CBOW)(2013)

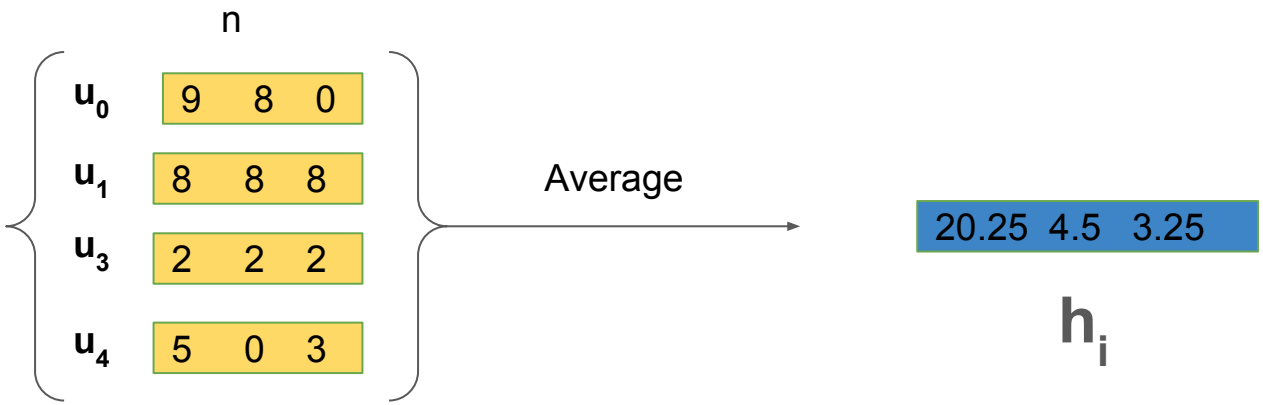


n arbitary length of our word embeddings

$W^{(1)} \in R^{n \times |V|}$ Input word vector

$u_i \in R^{n \times 1}$ Representation of X_i After multiplication with input matrix

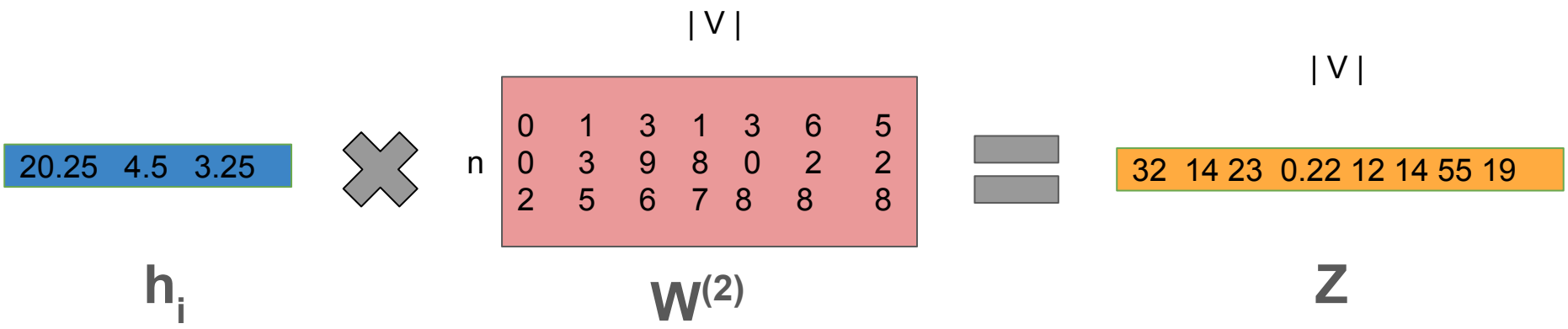
Continuous Bag of Words model (CBOW)(2013)



$h_i \in R^{n \times 1}$

h_i = Average of $u_0\, u_1\, u_3 u_4$

Continuous Bag of Words model (CBOW)(2013)

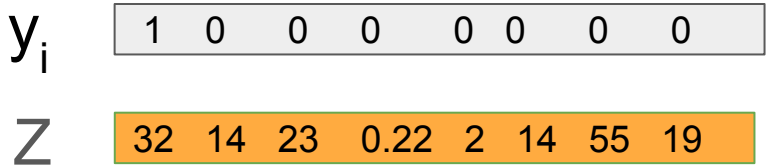


$W^{(2)} \in R^{n \times |V|}$ Output word vector

$Z \in R^{|V| \times 1}$ Output vector representation of X_i

$Z = h_i W^{(2)}$

Continuous Bag of Words model (CBOW)(2013)



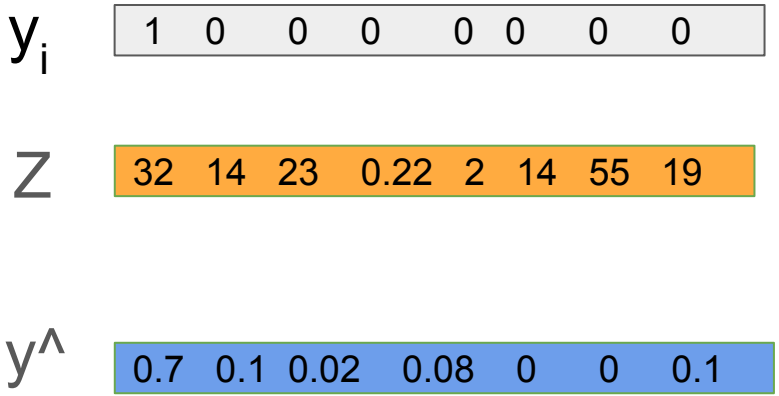
How to compare Z to y_i ?

Largest value corresponds to the correct class ? … no Softmax

Softmax: squashes a K-dimensional vector of arbitrary real values to a K-dimensional vector of real values in the range (0, 1)

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

Continuous Bag of Words model (CBOW)(2013)



y^ = softmax (Z)

$y_i \in R^{|V| \times 1}$ one hot representation of the correct middle word

Continuous Bag of Words model (CBOW)(2013)

- We need estimated words y^ to be closest to the original answer
- One common error function is the cross entropy H(yˆ, y) (why ?).

$$H(\hat{y}, y) = -\sum_{j=1}^{|V|} y_j \log(\hat{y}_j)$$

Since y is one hot vector

$$H(\hat{y}, y) = -y_i \log(\hat{y}_i)$$

Continuous Bag of Words model (CBOW)(2013)

- We need estimated words y^ to be closest to the original answer
- One common error function is the cross entropy error H(yˆ, y) (why ?).

$$H(\hat{y}, y) = -\sum_{j=1}^{|V|} y_j \log(\hat{y}_j)$$

Since y is one hot vector

$$H(\hat{y}, y) = -y_i \log(\hat{y}_i)$$

$$H(\hat{y}, y) = -\log(\hat{y}_i)$$

Continuous Bag of Words model (CBOW)(2013)

Perfect language model will expect the propability of the correct word $\hat{y}_i = 1$

So loss will be 0

$$H(\hat{y}, y) = -\log(\hat{y}_i)$$

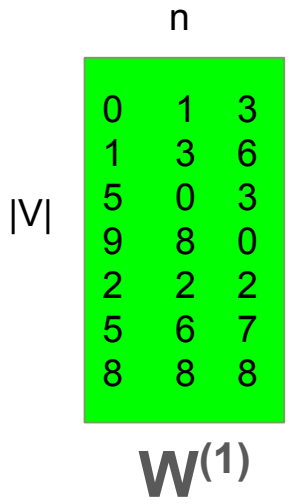
Optimization task :

- Learn $W^{(1)}$ and $W^{(2)}$ to minimize the cost function over all the dataset.
- using back propagation, update weights in $W^{(1)}$ and $W^{(2)}$

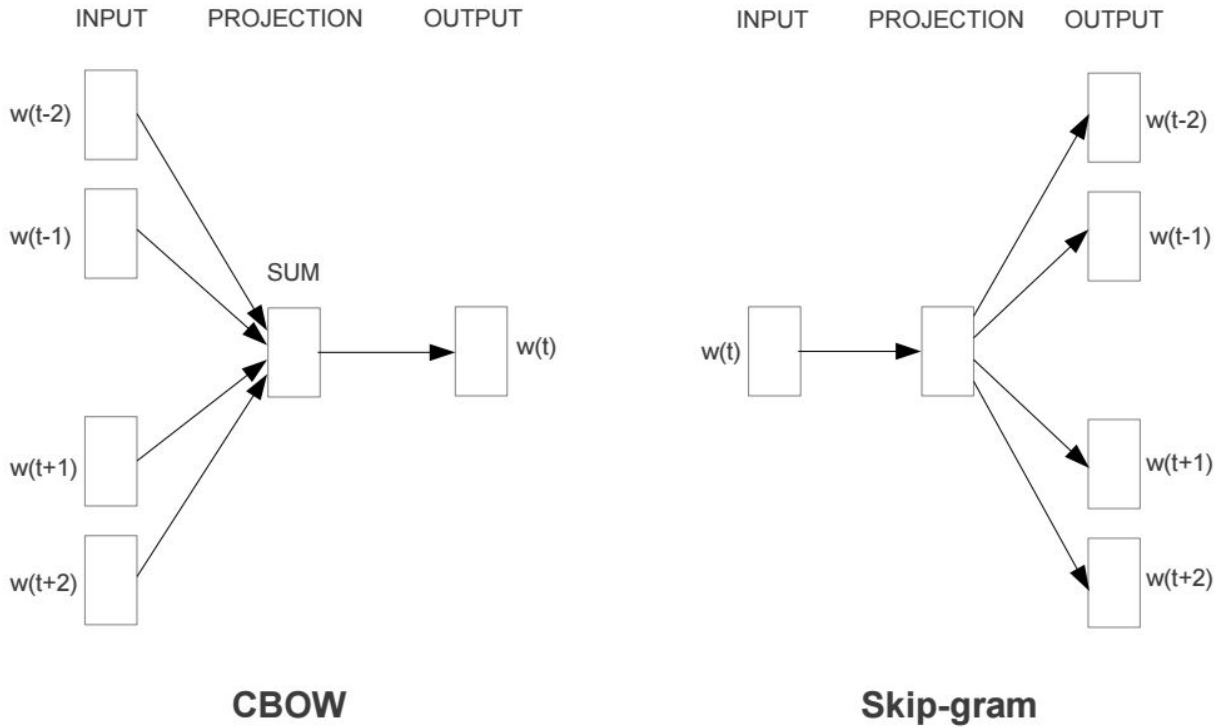
Continuous Bag of Words model (CBOW)(2013)

$W^{(1)}$:

- After training over a large corpus
- Each row represents a dense vector for each word in the vocabulary
- These word vectors contains better semantic and syntactic representation than other dense vectors (will be proven later)
- These word vectors performs better for all NLP tasks (will be proven later)



Skip Gram model (2013)



GloVe: Global Vectors for Word Representation, Pennington et al. (2014)

Motivation:

ice - steam = (solid, gas, water, fashion) ?

- A distributional model should capture words that appears with “ice” but not “steam”.
- Hence, doing well in semantic analogy task (explained later)

GloVe: Global Vectors for Word Representation, Pennington et al. (2014)

Starts from a co-oocurrrence matrix

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

$p(solid \mid ice) = X_{solid,ice} / X_{ice}$

GloVe: Global Vectors for Word Representation, Pennington et al. (2014)

Optimize the Objective function:

$$F(w_i - w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}.$$

W_i word vector of word i

P_{ik} probability of word k to occurs in context of word i

Ok, But are word vectors really good ?!

Evaluation of word vectors :

1. **Intrinsic evaluation :** make sure it encodes semantic information
2. **Extrinsic evaluation :** make sure it's useful for other NLP tasks **(the hype)**

Intrinsic Evaluation of Word Vectors

Word similarity task

Word 1	Word 2	Human (mean)
tiger	cat	7.35
tiger	tiger	10.00
book	paper	7.46
computer	internet	7.58
plane	car	5.77
professor	doctor	6.62
stock	phone	1.62
stock	CD	1.31
stock	jaguar	0.92

Intrinsic Evaluation of Word Vectors

Word similarity task

Model	Size	WS353	MC	RG	SCWS	RW
SVD	6B	35.3	35.1	42.5	38.3	25.6
SVD-S	6B	56.5	71.5	71.0	53.6	34.7
SVD-L	6B	65.7	72.7	75.1	56.5	37.0
CBOW†	6B	57.2	65.6	68.2	57.0	32.5
SG†	6B	62.8	65.2	69.7	58.1	37.2
GloVe	6B	65.8	72.7	77.8	53.9	38.1
SVD-L	42B	74.0	76.4	74.1	58.3	39.9
GloVe	42B	**75.9**	**83.6**	**82.9**	**59.6**	**47.8**
CBOW*	100B	68.4	79.6	75.4	59.4	45.5

Results from : GloVe: Global Vectors for Word Representation, Pennington et al 2014.
Word similarity dataset “WS353”: http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/

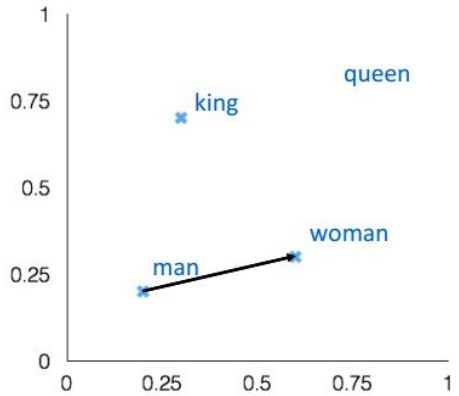
Intrinsic Evaluation of Word Vectors

Word Analogy task

$$x_{apple} - x_{apples} \approx x_{car} - x_{cars} \approx x_{family} - x_{families}$$

man:woman :: king:?

+	king	[0.30 0.70]
-	man	[0.20 0.20]
+	woman	[0.60 0.30]
	queen	[0.70 0.80]



Intrinsic Evaluation of Word Vectors

Word Analogy task

: capital-world
Abuja Nigeria Accra Ghana

: gram3-comparative
bad worse big bigger

: gram2-opposite
acceptable unacceptable aware unaware

: gram1-adjective-to-adverb
amazing amazingly apparent apparently

Evaluation data : https://word2vec.googlecode.com/svn/trunk/questions-words.txt

Intrinsic Evaluation of Word Vectors

Word Analogy task

Model	Dim.	Size	Sem.	Syn.	Tot.
ivLBL	100	1.5B	55.9	50.1	53.2
HPCA	100	1.6B	4.2	16.4	10.8
GloVe	100	1.6B	67.5	54.3	60.3
SG	300	1B	61	61	61
CBOW	300	1.6B	16.1	52.6	36.1
vLBL	300	1.5B	54.2	64.8	60.0
ivLBL	300	1.5B	65.2	63.0	64.0
GloVe	300	1.6B	80.8	61.5	70.3
SVD	300	6B	6.3	8.1	7.3
SVD-S	300	6B	36.7	46.6	42.1
SVD-L	300	6B	56.6	63.0	60.1
CBOW†	300	6B	63.6	67.4	65.7
SG†	300	6B	73.0	66.0	69.1
GloVe	300	6B	77.4	67.0	71.7
CBOW	1000	6B	57.3	68.9	63.7
SG	1000	6B	66.1	65.1	65.6
SVD-L	300	42B	38.4	58.2	49.2
GloVe	300	42B	**81.9**	**69.3**	**75.0**

Pretrained Word vectors ready for use

word2vec :

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

GloVe :
http://nlp.stanford.edu/projects/glove/

Dependency based :
https://levyomer.wordpress.com/...