Word Embeddings

Neural Embeddings for Political Science Research

Fridolin Linder

November 8, 2016

Methods Comprehensive Exam Presentation

Table of contents

Introduction

Introduction

- A lot of data in political science is text

Introduction

- A lot of data in political science is text
- Every corpus is analyzed in isolation

Representing Meaning in Text Analysis

Meaning of Documents

- Three documents:

Meaning of Documents

- Three documents:
    1. *The cat jumps over the dog*

Meaning of Documents

- Three documents:
    1. *The cat jumps over the dog*
    2. *The kitten hops ovqer the hound*

Meaning of Documents

- Three documents:
    1. *The cat jumps over the dog*
    2. *The kitten hops ovqer the hound*
    3. *Die Katze springt über den Hund*

Meaning of Documents

- Three documents:
    1. *The cat jumps over the dog*
    2. *The kitten hops ovqer the hound*
    3. *Die Katze springt über den Hund*
- Take the vocabulary $\mathcal{V} = \{\text{The, cat, ...}\}$ and represent them as vectors of dimension $|\mathcal{V}|$ in a Document-Term-Matrix $\underset{|\mathcal{D}|\times|\mathcal{V}|}{\mathsf{X}}$:

	the	cat	jump	over	dog	kitten	ovqer	hound	die	katze	springt	über	den	hund
D_1	2	1	1	1	1	0	0	0	0	0	0	0	0	0
D_2	2	0	0	0	0	1	1	1	0	0	0	0	0	0
D_3	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Meaning of Documents

- Three documents:
  1. *The cat jumps over the dog*
  2. *The kitten hops ovqer the hound*
  3. *Die Katze springt über den Hund*
- Take the vocabulary $\mathcal{V} = \{\text{The, cat, ...}\}$ and represent them as vectors of dimension $|\mathcal{V}|$ in a Document-Term-Matrix $\underset{|\mathcal{D}|\times|\mathcal{V}|}{\mathrm{X}}$:

	the	cat	jump	over	dog	kitten	ovqer	hound	die	katze	springt	über	den	hund
D_1	2	1	1	1	1	0	0	0	0	0	0	0	0	0
D_2	2	0	0	0	0	1	1	1	0	0	0	0	0	0
D_3	0	0	0	0	0	0	0	0	1	1	1	1	1	1

- Each word is a dimension of meaning

Meaning of Documents

- Three documents:
  1. *The cat jumps over the dog*
  2. *The kitten hops ovqer the hound*
  3. *Die Katze springt über den Hund*
- Take the vocabulary $\mathcal{V} = \{\text{The, cat, ...}\}$ and represent them as vectors of dimension $|\mathcal{V}|$ in a Document-Term-Matrix $\underset{|\mathcal{D}|\times|\mathcal{V}|}{\mathrm{X}}$:

	the	cat	jump	over	dog	kitten	ovqer	hound	die	katze	springt	über	den	hund
D_1	2	1	1	1	1	0	0	0	0	0	0	0	0	0
D_2	2	0	0	0	0	1	1	1	0	0	0	0	0	0
D_3	0	0	0	0	0	0	0	0	1	1	1	1	1	1

- Each word is a dimension of meaning
- Similarity: Angle between these vectors ($\cos(D_i, D_j)$)

Meaning of Words

You shall know a word by the company it keeps (Firth, 1957)

- Define a context (e.g. a document)

Meaning of Words

You shall know a word by the company it keeps (Firth, 1957)

- Define a context (e.g. a document)
- How often do two words appear in the same context?

	the	cat	jump	over	dog	kitten	...
the	2	1	1	1	1	1	...
cat	1	1	1	1	1	0	...
jump	1	1	1	1	1	0	...
over	1	1	1	1	1	0	...
dog	1	1	1	1	1	0	...
kitten	1	0	0	0	0	1	...
...	...	...	...	...	...	...	

Meaning of Words

You shall know a word by the company it keeps (Firth, 1957)

- Define a context (e.g. a document)
- How often do two words appear in the same context?

	the	cat	jump	over	dog	kitten	...
the	2	1	1	1	1	1	...
cat	1	1	1	1	1	0	...
jump	1	1	1	1	1	0	...
over	1	1	1	1	1	0	...
dog	1	1	1	1	1	0	...
kitten	1	0	0	0	0	1	...
...	...	...	...	...	...	...	

- What we want: $\cos(cat, jump) > \cos(cat, kitten)$

Problem

- Many redundant dimensions

	the	cat	jump	over	dog	kitten	ovqer	hound	die	katze	springt	über	den	hund
Doc. 1	2	1	1	1	1	0	0	0	0	0	0	0	0	0
Doc. 2	2	0	0	0	0	1	1	1	0	0	0	0	0	0
Doc. 3	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Problem

- Many redundant dimensions
- Short documents have many 0's (open ended surveys, social media data)

	the	cat	jump	over	dog	kitten	ovqer	hound	die	katze	springt	über	den	hund
Doc. 1	2	1	1	1	1	0	0	0	0	0	0	0	0	0
Doc. 2	2	0	0	0	0	1	1	1	0	0	0	0	0	0
Doc. 3	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Problem

- Many redundant dimensions
- Short documents have many 0's (open ended surveys, social media data)
- Things that should be similar are not

	the	cat	jump	over	dog	kitten	ovqer	hound	die	katze	springt	über	den	hund
Doc. 1	2	1	1	1	1	0	0	0	0	0	0	0	0	0
Doc. 2	2	0	0	0	0	1	1	1	0	0	0	0	0	0
Doc. 3	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Problem

- Many redundant dimensions
- Short documents have many 0's (open ended surveys, social media data)
- Things that should be similar are not
- Computational cost: A usual corpus has 10^5 unique words sometimes more

	the	cat	jump	over	dog	kitten	ovqer	hound	die	katze	springt	über	den	hund
Doc. 1	2	1	1	1	1	0	0	0	0	0	0	0	0	0
Doc. 2	2	0	0	0	0	1	1	1	0	0	0	0	0	0
Doc. 3	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Word Embeddings

Basic Idea

- Documents / Words are ‘embedded’ in lower dimensional space

Basic Idea

- Documents / Words are ‘embedded’ in lower dimensional space
- Our example:

Basic Idea

- Documents / Words are ‘embedded’ in lower dimensional space
- Our example:
    - ‘cat’ $\sim$ {cat, kitten, Katze}

Basic Idea

- Documents / Words are ‘embedded’ in lower dimensional space
- Our example:
    - ‘cat’ $\sim$ {cat, kitten, Katze}
    - ‘dog’ $\sim$ {dog, hound, Hund}

Basic Idea

- Documents / Words are ‘embedded’ in lower dimensional space
- Our example:
  - ‘cat’ $\sim$ {cat, kitten, Katze}
  - ‘dog’ $\sim$ {dog, hound, Hund}
  - ‘article’ $\sim$ {the, The, die, den}

Basic Idea

- Documents / Words are 'embedded' in lower dimensional space
- Our example:
  - 'cat' $\sim$ {cat, kitten, Katze}
  - 'dog' $\sim$ {dog, hound, Hund}
  - 'article' $\sim$ {the, The, die, den}
- New Document Matrix:

	'cat'	'dog'	'jump'	'article'
D_1	1	1	1	2
D_2	1	1	1	2
D_3	1	1	1	2

Methods for Embeddings

- Methods to find a good low dimensional representation of a matrix

Methods for Embeddings

- Methods to find a good low dimensional representation of a matrix
  - SVD, PCA, Factor Analysis

Methods for Embeddings

- Methods to find a good low dimensional representation of a matrix
  - SVD, PCA, Factor Analysis
  - IRT Models

Methods for Embeddings

- Methods to find a good low dimensional representation of a matrix
    - SVD, PCA, Factor Analysis
    - IRT Models
    - Topic Models

Methods for Embeddings

- Methods to find a good low dimensional representation of a matrix
  - SVD, PCA, Factor Analysis
  - IRT Models
  - Topic Models
  - Neural Networks

Word Embeddings with Neural Nets

- Continuous Bag of Words (CBOW) Model (Mikolov et al., 2013; Le and Mikolov, 2014)

Word Embeddings with Neural Nets

- Continuous Bag of Words (CBOW) Model (Mikolov et al., 2013; Le and Mikolov, 2014)
- Shallow neural net

Word Embeddings with Neural Nets

- Continuous Bag of Words (CBOW) Model (Mikolov et al., 2013; Le and Mikolov, 2014)
- Shallow neural net
- Can use very large training copora

Word Embeddings with Neural Nets

- Continuous Bag of Words (CBOW) Model (Mikolov et al., 2013; Le and Mikolov, 2014)
- Shallow neural net
- Can use very large training copora
- Produces good embeddings:

$$
\begin{aligned}
\mathbf{v}_{queen} - \mathbf{v}_{woman} &\approx \mathbf{v}_{king} \\
\mathbf{v}_{paris} - \mathbf{v}_{france} &\approx \mathbf{v}_{rome}
\end{aligned}
$$

Meaning and Prediction

context .... word .... context

The cat jumps over the dog. The kitten hops over the hound.

- Intuition, let's model:

$$P(word|context)$$

For all words in $\mathcal{V}$

Meaning and Prediction

context .... word .... context

The cat *jumps* *over the dog*. *The kitten hops over the hound.*

- Intuition, let's model:

$$P(word|context)$$

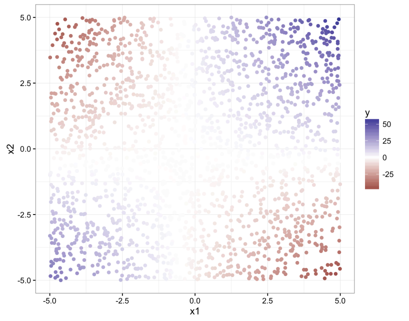
For all words in $\mathcal{V}$

- The parameters of this model will contain meaning

Neural Networks

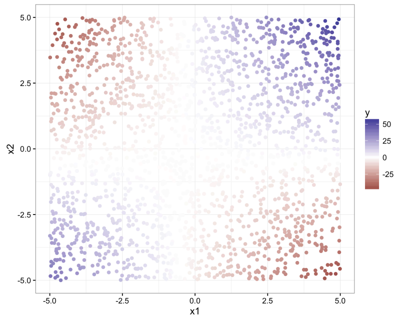
A Simple Neural Net

- Learn the function $y = f(\mathbf{x})$



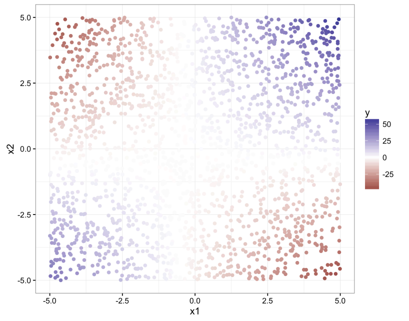
A Simple Neural Net

- Learn the function $y = f(\mathbf{x})$
- $y \in \mathbb{R}$, $\mathbf{x} \in \mathbb{R}^2$

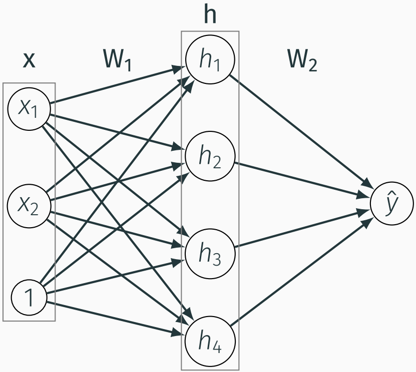


A Simple Neural Net

- Learn the function $y = f(\mathbf{x})$
- $y \in \mathbb{R}$, $\mathbf{x} \in \mathbb{R}^2$
- $y = x_1 + x_2 + 2x_1x_2 + \epsilon$

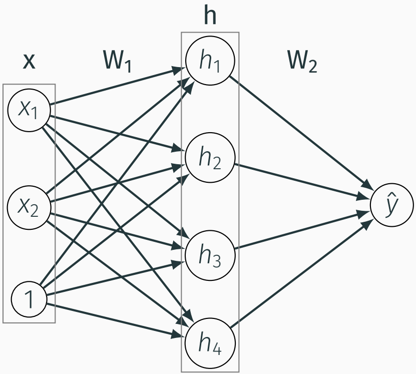


Architecture



- $p = 3$ inputs
- One hidden layer
- $k = 4$ hidden nodes
- Weight Matrices $\underset{p \times k}{\mathbf{W}_1}, \underset{1 \times k}{\mathbf{W}_2}$
- Activation function: $\sigma(x) = \frac{1}{1+e^{-x}}$

Feed Forward Prediction



Hidden Layer

$$\underset{1\times k}{\mathbf{z}} = \underset{1\times p}{\mathbf{x}} \times \underset{p\times k}{\mathbf{W_1}}$$

$$\underset{1\times k}{\mathbf{h}} = \sigma(\underset{1\times k}{\mathbf{z}})$$

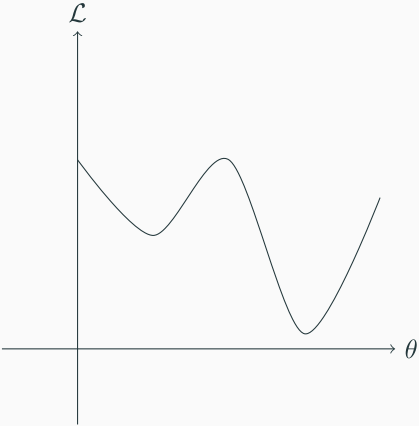
Ouput Layer

$$\underset{1\times 1}{\hat{y}} = \underset{1\times k}{\mathbf{h}} \times \underset{k\times 1}{\mathbf{W_2}}$$

Loss

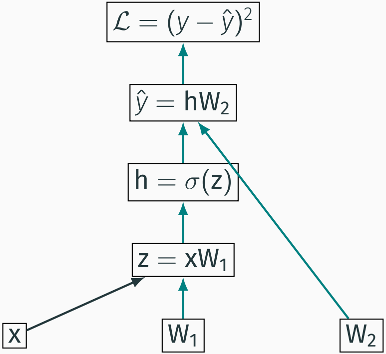
$$\mathcal{L} = (y - \hat{y})^2$$

Estimation - Error Back Propagation



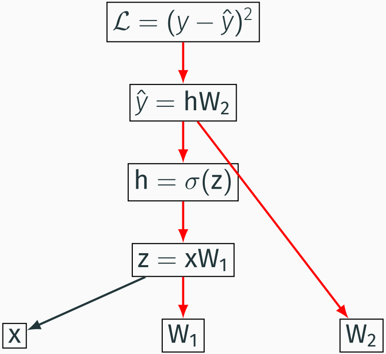
- Minimize the loss through the parameters $\theta = \{\mathsf{W}_1, \mathsf{W}_2\}$
- Need to find $\frac{\delta \mathcal{L}}{\delta \theta}$
- Then $\mathcal{L}$ can be minimized with gradient descent

Estimation - Backpropagation I



- Make a prediction given inputs and parameters

Estimation - Backpropagation II



- Propagate the errors back through the network

Training

- Feed Forward $->$ Back Propagate $->$ Adjust θ with $\frac{\delta \mathcal{L}}{\delta \theta}$ $->$ Repeat

Continuous Bag of Words Model

Overview

- Context size C
- Vocabulary $\mathcal{V} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots\}$
- $\underset{1\times|\mathcal{V}|}{\mathbf{x}_j} = [0, 0, 0, \ldots, 1, 0, \ldots, 0]$
- Focus word $\underset{1\times|V|}{\mathbf{y}} \in \mathcal{V}$
- Context words $\underset{C\times|V|}{\mathbf{X}} \subset \mathcal{V}$
- Model $P(\mathbf{y}|\mathbf{X})$

“

I’m telling you, I used to use the word incompetent. Now I just call them stupid. I went $\boxed{\text{to an Ivy}}$ $\boxed{\text{League}}$ $\boxed{\text{school. I’m very}}$ highly educated. I know words, I have the best words…but there is no better word than stupid. Right?

”

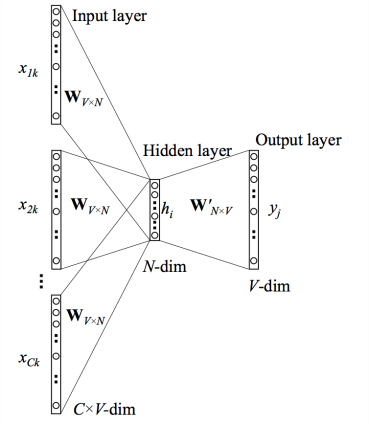


Figure 1: CBOW architecture. Figure from Rong (2014)

Hidden Layer (N units)

$$\underset{1\times N}{\mathrm{z}} = \underset{1\times|\mathcal{V}|}{(\mathrm{x}_1 + \mathrm{x}_1 + \ldots + \mathrm{x}_C)} \times \underset{|\mathcal{V}|\times N}{\mathrm{W}_1}$$

$$\underset{1\times N}{\mathrm{h}} = \frac{\mathrm{z}}{C}$$

Ouput Layer

$$\underset{1\times|V|}{\mathrm{u}} = \underset{1\times N}{\mathrm{h}} \times \underset{N\times|\mathcal{V}|}{\mathrm{W}_2}$$

$$\underset{1\times|V|}{\hat{\mathrm{y}}} = \frac{e^{\mathrm{u}}}{\sum_{i=1}^{|\mathcal{V}|} e^{u_i}}$$

Loss

$$\mathcal{L} = -logP(\mathrm{y}|\mathrm{X}) = -log(\underset{1\times|\mathcal{V}|}{\hat{\mathrm{y}}} \times \underset{|\mathcal{V}|\times 1}{\mathrm{y}})$$

Meaning of Words?

- $\underset{|\mathcal{V}|\times N}{\mathbf{W}_1}$ and $\underset{N\times|\mathcal{V}|}{\mathbf{W}_2}$ encode knowledge about $P(\mathbf{y}|\mathbf{X})$

Meaning of Words?

- $\underset{|\mathcal{V}| \times N}{\mathbf{W}_1}$ and $\underset{N \times |\mathcal{V}|}{\mathbf{W}_2}$ encode knowledge about $P(\mathbf{y}|\mathbf{X})$
- Words are represented by vectors

Meaning of Words?

- $\underset{|\mathcal{V}|\times N}{\mathbf{W}_1}$ and $\underset{N\times|\mathcal{V}|}{\mathbf{W}_2}$ encode knowledge about $P(\mathbf{y}|\mathbf{X})$
- Words are represented by vectors
- Dot product is a function of the angle (i.e. the similarity) of two vectors

Meaning of Words?

- $\underset{|\mathcal{V}|\times N}{\mathbf{W}_1}$ and $\underset{N\times|\mathcal{V}|}{\mathbf{W}_2}$ encode knowledge about $P(\mathbf{y}|\mathbf{X})$
- Words are represented by vectors
- Dot product is a function of the angle (i.e. the similarity) of two vectors
- Training objective:

Meaning of Words?

- $\underset{|\mathcal{V}|\times N}{\mathbf{W_1}}$ and $\underset{N\times|\mathcal{V}|}{\mathbf{W_2}}$ encode knowledge about $P(\mathbf{y}|\mathbf{X})$
- Words are represented by vectors
- Dot product is a function of the angle (i.e. the similarity) of two vectors
- Training objective:
  - Frequent context - word pairs: Large dot products ($\mathbf{u} = \mathbf{h}\mathbf{W}_2$)

Meaning of Words?

- $\underset{|\mathcal{V}|\times N}{\mathbf{W}_1}$ and $\underset{N\times|\mathcal{V}|}{\mathbf{W}_2}$ encode knowledge about $P(\mathbf{y}|\mathbf{X})$
- Words are represented by vectors
- Dot product is a function of the angle (i.e. the similarity) of two vectors
- Training objective:
    - Frequent context - word pairs: Large dot products ($\mathbf{u} = \mathbf{h}\mathbf{W}_2$)
    - Infrequent context - word pairs: Small dot products ($\mathbf{u} = \mathbf{h}\mathbf{W}_2$)

Remarks

- In actual training not all $\hat{y}$ are calculated (Negative Sampling, Hierarchical Softmax)
- There is much debate about why these models produce useful embeddings (Arora et al. (2015) give intuition)
- It is not clear why a lower dimensional representation does better in NLP tasks
- $\rightarrow$ Much theoretical work to be done

Conclusion

Why is this valuable for political science research?

- Short documents
- Comparative Research - Multiple languages
- Use general knowledge about data in every task

Questions?

References

Arora, S., Y. Li, Y. Liang, T. Ma, and A. Risteski (2015). Random walks on context spaces: Towards an explanation of the mysteries of semantic word embeddings. *CoRR abs/1502.03520*.

Firth, J. R. (1957). *Papers in linguistics, 1934-1951*. Oxford University Press.

Le, Q. V. and T. Mikolov (2014). Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119.

References II

Rong, X. (2014). word2vec parameter learning explained. *CoRR abs/1411.2738*.

Appendix

Estimation - Error Back Propagation

$$\boxed{\mathcal{L} = (y - \hat{y})^2}$$

$$\boxed{\hat{y} = \mathsf{h}\mathsf{W}_2}$$

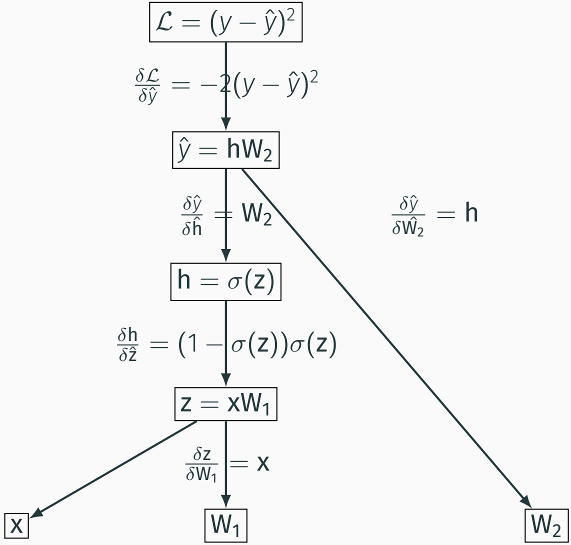
$$\boxed{\mathsf{h} = \sigma(\mathsf{z})}$$

$$\boxed{\mathsf{z} = \mathsf{x}\mathsf{W}_1}$$

$$\boxed{\mathsf{x}} \qquad \boxed{\mathsf{W}_1} \qquad \boxed{\mathsf{W}_2}$$

- How do the parameters affect $\mathcal{L}$
- Decompose $\mathcal{L}$ into a computational graph

Simple Network - Backpropagation

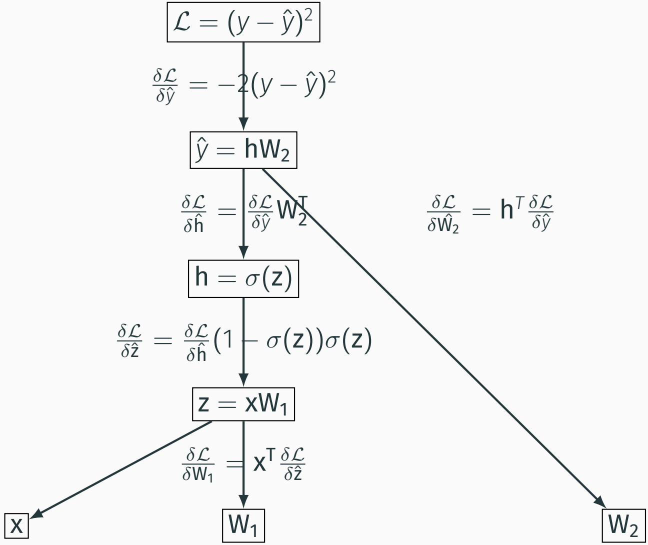


- Calculate node derivatives

Estimation - Error Back Propagation

- How do the parameters affect $\mathcal{L}$?

Simple Network - Backpropagation



- Calculate node derivatives
- Use the chain rule to propagate the errors through the network

CBOW Backprop Equations

Node Derivatives

$$
\begin{aligned}
\frac{\delta \mathcal{L}}{\delta \mathrm{u}} &= \hat{\mathrm{y}} - \mathrm{y} = \mathrm{e} \\
\frac{\delta \mathrm{u}}{\delta \mathrm{W}_2} &= \mathrm{h} \\
\frac{\delta \mathrm{u}}{\delta \mathrm{h}} &= \mathrm{W}_2 \\
\frac{\delta \mathrm{h}}{\delta \mathrm{W}_1} &= \left(\frac{\mathrm{x}_1 + \ldots + \mathrm{x}_\mathrm{C}}{C}\right)^T
\end{aligned}
$$

Chained Derivatives w.r.t. $\mathcal{L}$

$$
\begin{aligned}
\frac{\delta \mathcal{L}}{\delta \mathrm{u}} &= \hat{\mathrm{y}} - \mathrm{y} = \mathrm{e} \\
\frac{\delta \mathcal{L}}{\delta \mathrm{W}_2} &= \mathrm{h}^\mathrm{T}\mathrm{e} \\
\frac{\delta \mathcal{L}}{\delta \mathrm{h}} &= \frac{\delta \mathcal{L}}{\delta \mathrm{u}}\mathrm{W}_2^\mathrm{T} \\
\frac{\delta \mathcal{L}}{\delta \mathrm{W}_1} &= \left(\frac{\mathrm{x}_1 + \ldots + \mathrm{x}_\mathrm{C}}{C}\right)^T \frac{\delta \mathcal{L}}{\delta \mathrm{h}}
\end{aligned}
$$

Updating the Word Vectors

Intuition

- If the model is wrong for word i (i.e. $|e_i|$ is large) the corresponding column in $\mathsf{W_2}$ is adjusted
- If $\hat{y}_i > y_i$ (overestimated) $\mathsf{w_{2i}}$ is pushed away from h (the context)
- If $\hat{y}_i < y_i$ (underestimated) $\mathsf{w_{2i}}$ is drawn towards h (the context)

Updating Equations

$$
\begin{aligned}
\frac{\delta \mathcal{L}}{\delta \mathsf{u}} &= \hat{\mathsf{y}} - \mathsf{y} = \mathsf{e} \\
\frac{\delta \mathcal{L}}{\delta \mathsf{W_2}} &= \mathsf{h}^{\mathsf{T}}\mathsf{e}
\end{aligned}
$$

Example - Setup

- $V = \{\text{know, words, have}\}$
- $|V| = 3$, $N = 2$, $C = 1$
- Focus word: 'words'
- Context word: 'know'
- Initialize parameters randomly:

$$\mathsf{W}_1 = \begin{array}{c} \textit{know} \\ \textit{words} \\ \textit{have} \end{array} \begin{bmatrix} 0.3 & 0.02 \\ 0.01 & -0.03 \\ -0.4 & 0.001 \end{bmatrix} \qquad \mathsf{W}_2 = \begin{array}{ccc} \textit{know} & \textit{words} & \textit{have} \\ \end{array} \begin{bmatrix} 0.02 & -0.002 & 0.1 \\ 0.04 & 0.008 & -0.03 \end{bmatrix}$$

Example - Feed Forward

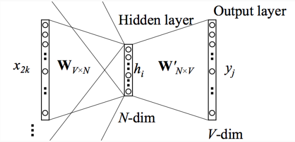


Figure 2: CBOW architecture. Figure from Rong (2014)

- $h = x_1 W_1 = [0.3, 0.02]$
- $u = hW_2 = [0.0068, -0.00044, 0.0294]$
- $\hat{y} = [0.331, 0.329, 0.339]$
- $e = \hat{y} - y = [0.331, 0.329, 0.339] - [0, 1, 0] = [0.331, -0.671, 0.339]$

Example - Update

$$
\begin{aligned}
W_2^{new} &= W_2 + \eta h^T e \\
&= W_2 + \eta \begin{array}{ccc} \textit{know} & \textit{words} & \textit{have} \\ \hline 0.010 & -0.201 & 0.101 \\ 0.006 & -0.013 & 0.006 \end{array}
\end{aligned}
$$