Neural Word Embeddings as Matrix Factorization

Master’s Thesis Mathematics

Presented by:
Theresa Klumpp

Supervisors:
Prof. P. Pfaffelhuber
Prof. H. Bast

January 15, 2020
**Goal:** word vectors that reflect similarities and dissimilarities
Problem

Goal: word vectors that reflect similarities and dissimilarities

Underlying hypothesis: words in similar contexts have similar meanings
Goal: word vectors that reflect similarities and dissimilarities

Underlying hypothesis: words in similar contexts have similar meanings
  - I get to work faster when I take the ***.

This model has amazing acceleration for a *** of its size.
I would never drive my *** into Paris if I could get there by train.
Goal: word vectors that reflect similarities and dissimilarities

Underlying hypothesis: words in similar contexts have similar meanings
  - I get to work faster when I take the ***.
  - This model has amazing acceleration for a *** of its size.
Goal: word vectors that reflect similarities and dissimilarities

Underlying hypothesis: words in similar contexts have similar meanings

- I get to work faster when I take the ***.
- This model has amazing acceleration for a *** of its size.
- I would never drive my *** into Paris if I could get there by train.
**Goal:** word vectors that reflect similarities and dissimilarities

**Underlying hypothesis:** words in similar contexts have similar meanings

- I get to work faster when I take the ***.
- This model has amazing acceleration for a *** of its size.
- I would never drive my *** into Paris if I could get there by train.

**Demo**
Contributions

- Gaining an understanding of the objective functions of skip-gram (with and without negative sampling) and the statistical models behind them.
- Finding a maximum for skip-gram’s objective.
- Showing the connection between the neural networks and Singular Value Decomposition (SVD).
- Comparing different metrics on the sphere.
- Finding a formula for the expectation of the distance of the closest vector.
- An implementation of the SGNS neural network and the SVD variant for both skip-gram and SGNS.
- Evaluation of the models on word similarity and analogy tasks.
Contributions

- Gaining an understanding of the objective functions of skip-gram (with and without negative sampling) and the statistical models behind them.
- Finding a maximum for skip-gram’s objective.
- Showing the connection between the neural networks and Singular Value Decomposition (SVD).
- Comparing different metrics on the sphere.
- Finding a formula for the expectation of the distance of the closest vector.
- An implementation of the SGNS neural network and the SVD variant for both skip-gram and SGNS.
- Evaluation of the models on word similarity and analogy tasks.
Questions?
Table of Contents

1 Problem

2 Solution

3 Evaluation
### Definition: Context

<table>
<thead>
<tr>
<th>Text</th>
<th>Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>I get to work faster when I take the car. ⇒ (l, get) (l, to) (get, l) (get, to) (get, work)</td>
<td></td>
</tr>
<tr>
<td>I get to work faster when I take the car. ⇒ (to, l) (to, get) (to, work) (to, faster)</td>
<td></td>
</tr>
<tr>
<td>I get to work faster when I take the car. ⇒ (work, get) (work, to) (work, faster) (work, when)</td>
<td></td>
</tr>
</tbody>
</table>

...
Notation

- \( V_W \) and \( V_C \): word and context vocabulary (we have \( V_W = V_C \))
- \( D \): observed word context pairs
- \#(w, c): number of times the pair \((w, c)\) appears in \( D \)
- \#(w) = \( \sum_{c' \in V_C} \#(w, c') \) and \#(c) = \( \sum_{w' \in V_W} \#(w', c) \)
Mathematical Goal

Find embeddings such that $\vec{w} \cdot \vec{c}$ is

- high for pairs with large $\#(w, c)$ and
- small for pairs with low $\#(w, c)$
Mathematical Goal

Find embeddings such that $\vec{w} \cdot \vec{c}$ is
- high for pairs with large $\#(w, c)$ and
- small for pairs with low $\#(w, c)$

Why does this yield good embeddings?
Mathematical Goal

Find embeddings such that $\vec{w} \cdot \vec{c}$ is

- high for pairs with large $\# (w, c)$ and
- small for pairs with low $\# (w, c)$

Why does this yield good embeddings?

<table>
<thead>
<tr>
<th></th>
<th>$c_1 = \text{drive}$</th>
<th>$c_2 = \text{road}$</th>
<th>$c_3 = \text{space}$</th>
<th>$c_4 = \text{bottle}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_1 = \text{car}$</td>
<td>0.9</td>
<td>0.8</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>$w_2 = \text{truck}$</td>
<td>0.8</td>
<td>0.7</td>
<td>0.2</td>
<td>0.2</td>
</tr>
</tbody>
</table>
Mathematical Goal

Find embeddings such that $\vec{w} \cdot \vec{c}$ is

- high for pairs with large $\# (w, c)$ and
- small for pairs with low $\# (w, c)$

$$W = \begin{pmatrix} \vec{w}_1 \\ \vdots \\ \vec{w}_{|V_W|} \end{pmatrix} \text{ and } C = \begin{pmatrix} \vec{c}_1 \\ \vdots \\ \vec{c}_{|V_C|} \end{pmatrix}$$
Mathematical Goal

Find embeddings such that $\vec{w} \cdot \vec{c}$ is

- high for pairs with large $\# (w, c)$ and
- small for pairs with low $\# (w, c)$

$$W = \begin{pmatrix} \vec{w}_1 \\ \vdots \\ \vec{w}_{|V_W|} \end{pmatrix} \quad \text{and} \quad C = \begin{pmatrix} \vec{c}_1 \\ \vdots \\ \vec{c}_{|V_C|} \end{pmatrix}$$

⇒ Find a function $\ell (W, C)$ that is maximized when the properties above hold.
Skip-Gram: Objective functions

\[ \ell_{SG}(W, C) = \sum_{(w,c) \in D} \left( \vec{w} \cdot \vec{c} - \log \left( \sum_{c' \in V_c} \exp \left( \vec{w} \cdot \vec{c}' \right) \right) \right) \]
Skip-Gram: Objective functions

\[ \ell_{SG}(W, C) = \sum_{(w,c) \in D} \left( \vec{w} \cdot \vec{c} - \log \left( \sum_{c' \in V_C} \exp \left( \vec{w} \cdot \vec{c}' \right) \right) \right) \]

\[ \ell_{SGNS}(W, C) = \sum_{(w,c) \in D} \left( \log \sigma (\vec{w} \cdot \vec{c}) + \sum_{j=1}^{k} \log \sigma (-\vec{w} \cdot \vec{c}_j) \right) \]

[Graph showing the sigmoid function \( \sigma(x) = \frac{1}{1 + e^{-x}} \)]
Optimal value for the dot products

- $\ell_{SGNS}(W, C)$ is maximized for

$$\left(\vec{w} \cdot \vec{c}\right)^{OPT} = \log\left(\frac{\#(w, c) \cdot |D|}{\#(w) \cdot \#(c)}\right) - \log k$$
Optimal value for the dot products

• $\ell_{SGNS}(W, C)$ is maximized for

\[ (\vec{w} \cdot \vec{c})^{OPT} = \log \left( \frac{\# (w, c) \cdot |D|}{\# (w) \cdot \# (c)} \right) - \log k \]

• Note that

\[ (W \cdot C^T)_{ij} = \vec{w}_i \cdot \vec{c}_j \]
Optimal value for the dot products

- $\ell_{\text{SGNS}}(W, C)$ is maximized for

$$\text{(OPT)} = \log \left( \frac{\# (w, c) \cdot |D|}{\# (w) \cdot \# (c)} \right) - \log k$$

- Note that

$$(W \cdot C^T)_{ij} = \vec{w}_i \cdot \vec{c}_j$$

- Let $M_{\text{OPT}}$ be the matrix containing the optimal dot products, that is

$$M_{ij}^{\text{OPT}} = (\vec{w}_i \cdot \vec{c}_j)^{\text{OPT}}$$
Singular Value Decomposition (SVD)

\[(W \cdot C^T)_{ij} = \vec{w}_i \cdot \vec{c}_j \quad \text{and} \quad M_{ij}^{\text{OPT}} = (\vec{w}_i \cdot \vec{c}_j)^{\text{OPT}}\]
Singular Value Decomposition (SVD)

\[
(W \cdot C^T)_{ij} = \vec{w}_i \cdot \vec{c}_j \quad \text{and} \quad M_{ij}^{\text{OPT}} = (\vec{w}_i \cdot \vec{c}_j)^{\text{OPT}}
\]

Skip-gram with negative sampling is trying to find \(W\) and \(C\) such that

\[
W \cdot C^T = M^{\text{OPT}}
\]
Singular Value Decomposition (SVD)

- \((W \cdot C^T)_{ij} = \vec{w}_i \cdot \vec{c}_j\) and \(M_{ij}^{OPT} = (\vec{w}_i \cdot \vec{c}_j)^{OPT}\)

- Skip-gram with negative sampling is trying to find \(W\) and \(C\) such that \(W \cdot C^T = M^{OPT}\)

- Truncated SVD gives us a factorization of the best rank \(d\) approximation of \(M^{OPT}\):
  \[W_{SVD} \cdot C_{SVD}^T = \arg \min_{M|\text{rk}(M)=d} \|M - M^{OPT}\|_F\]
Skip-Gram (without negative sampling)

Recall from previous slide:

\[
\ell_{SG}(W, C) = \sum_{(w, c) \in D} \left( w \cdot c - \log \left( \sum_{c' \in V_C} \exp (w \cdot c') \right) \right)
\]

Computations for the skip-gram model (without negative sampling) yield a maximum for

\[(w \cdot c)^{OPT} = \log \# (w, c)\]
Problems with SVD

\[ M_{ij}^{\text{OPT}} = \log \left( \frac{\# (w_i, c_j) \cdot |D|}{\# (w_i) \cdot \# (c_j)} \right) - \log k \]
Problems with SVD

\[ M_{ij}^{\text{OPT}} = \log \left( \frac{\# (w_i, c_j) \cdot |D|}{\# (w_i) \cdot \# (c_j)} \right) - \log k \]

1. What about pairs with \( \# (w_i, c_j) = 0? \) (This is the case for more than 98% of our pairs!)
2. \( M^{\text{OPT}} \) is dense.
Problems with SVD

\[ M_{ij}^{\text{OPT}} = \log \left( \frac{\# (w_i, c_j) \cdot |D|}{\# (w_i) \cdot \# (c_j)} \right) - \log k \]

1. What about pairs with \( \# (w_i, c_j) = 0? \) (This is the case for more than 98% of our pairs!)
2. \( M^{\text{OPT}} \) is dense.

**Solution:** Factorize

\[ M_{ij}^+ = \max \left( \log \left( \frac{\# (w_i, c_j) \cdot |D|}{\# (w_i) \cdot \# (c_j)} \right) - \log k, 0 \right) \]
Questions?
Table of Contents

1 Problem

2 Solution

3 Evaluation
Experiment Setup

data: \sim 4.6 \text{ million English Wikipedia articles}
vocabulary size: \sim 160,000  
            (words that appeared at least 300 times)
window size: 2
word-context samples: \sim 9.7 \text{ billion}
embedding dimension: 200
3 Evaluation
- Optimizing the objective
- Word Similarity Tasks
- Analogy Tasks
### Optimizing the Objective

The following table shows the percentage of deviation from the optimal value, that is

\[
\frac{\ell - \ell^\text{OPT}}{\ell^\text{OPT}}.
\]

<table>
<thead>
<tr>
<th>$k$</th>
<th>$\ell^\text{OPT}$</th>
<th>$\ell^+$</th>
<th>SVD</th>
<th>NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0%</td>
<td>5.7%</td>
<td>25.1%</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>0%</td>
<td>29.3%</td>
<td>38.8%</td>
<td>22.7%</td>
</tr>
<tr>
<td>5</td>
<td>0%</td>
<td>120.9%</td>
<td>124.7%</td>
<td>9.5%</td>
</tr>
<tr>
<td>15</td>
<td>0%</td>
<td>309.0%</td>
<td>310.4%</td>
<td>8.9%</td>
</tr>
</tbody>
</table>

**Table:** Percentage of deviation from the optimal objective value.
Table of Contents

3 Evaluation
- Optimizing the objective
- Word Similarity Tasks
- Analogy Tasks
Word Similarity Tasks

Models were tested to two datasets:
- WordSim353: 353 word pairs
- MEN: 3000 word pairs

<table>
<thead>
<tr>
<th>word pairs</th>
<th>human assigned similarity scores</th>
</tr>
</thead>
<tbody>
<tr>
<td>stock market</td>
<td>8.08</td>
</tr>
<tr>
<td>physics chemistry</td>
<td>7.35</td>
</tr>
<tr>
<td>game round</td>
<td>5.97</td>
</tr>
<tr>
<td>experience music</td>
<td>3.47</td>
</tr>
<tr>
<td>stock jaguar</td>
<td>0.92</td>
</tr>
</tbody>
</table>

Table: Examples from the WordSim353 dataset
# Word Similarity Tasks

<table>
<thead>
<tr>
<th>k</th>
<th>WordSim353</th>
<th>MEN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NN</td>
<td>SVD</td>
</tr>
<tr>
<td>0</td>
<td>-</td>
<td>0.601</td>
</tr>
<tr>
<td>1</td>
<td>0.524</td>
<td>0.613</td>
</tr>
<tr>
<td>5</td>
<td>0.658</td>
<td>0.536</td>
</tr>
<tr>
<td>15</td>
<td>0.644</td>
<td>0.400</td>
</tr>
</tbody>
</table>

**Table:** Spearman’s correlation between dataset similarity scores and similarity scores that different the models returned.

**Note:** Spearman’s correlation \( \rho_S \in [-1, 1] \), where negative (positive) numbers indicate negative (positive) correlation and zero indicates no correlation.

[More about Spearman’s correlation](#)
Table of Contents

3 Evaluation
- Optimizing the objective
- Word Similarity Tasks
- Analogy Tasks
Analogy Tasks

Berlin is to Germany as Paris is to France.
**Analogy Tasks**

*Berlin* is to *Germany* as *Paris* is to *France*.
Analogy Tasks

Berlin is to Germany as Paris is to France.

⇒ \text{vec (Germany)} - \text{vec (Berlin)} = \text{vec (France)} - \text{vec (Paris)}
Analogy Tasks

Berlin is to Germany as Paris is to France.

\[ \text{vec} (\text{Germany}) - \text{vec} (\text{Berlin}) = \text{vec} (\text{France}) - \text{vec} (\text{Paris}) \]

in other words:

\[ \text{vec} (\text{France}) = \text{vec} (\text{Germany}) - \text{vec} (\text{Berlin}) + \text{vec} (\text{Paris}) \]
## Analogy Tasks

<table>
<thead>
<tr>
<th>k</th>
<th>Mixed dataset</th>
<th></th>
<th>Syntactic dataset</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>19.500 analogies</td>
<td></td>
<td>8.000 analogies</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>-</td>
<td>26.8%</td>
<td>-</td>
<td>28.7%</td>
</tr>
<tr>
<td>1</td>
<td>27.3%</td>
<td>30.6%</td>
<td>32.3%</td>
<td>19.6%</td>
</tr>
<tr>
<td>5</td>
<td>51.0%</td>
<td>12.0%</td>
<td>51.0%</td>
<td>5.7%</td>
</tr>
<tr>
<td>15</td>
<td>53.2%</td>
<td>5.9%</td>
<td>47.9%</td>
<td>1.4%</td>
</tr>
</tbody>
</table>

**Table:** Percentage of correct answers on two word analogy datasets.
Questions?
Expectation of the closest vector

**Figure:** Expectation of the cosine distance to the nearest vector for 159,862 vectors depending on the embedding dimension.
Figure: The expectation of the distance to the closest word depending on the embedding dimension and the number of words.
Skip-Gram

Input

Hidden Layer

Dot Products (Prob. Distr.)

Output

Label

$\in \mathbb{R}^{V_W}$

$\in \mathbb{R}^d$

$\in \mathbb{R}^{V_C}$
Objective function \( \ell_{SG} \):

\[
\ell_{SG}(W, C) = \sum_{(w, c) \in D} \log \frac{\exp(\vec{w} \cdot \vec{c})}{\sum_{c' \in V_C} \exp(\vec{w} \cdot \vec{c'})}
\]

\[
= \sum_{(w, c) \in D} \left( \vec{w} \cdot \vec{c} - \log \left( \sum_{c' \in V_C} \exp(\vec{w} \cdot \vec{c'}) \right) \right)
\]
Skip-Gram with negative sampling

Input Layer

Hidden Layer

Dot Products

Output Layer

Label

\[ i \]

\[ W \]

\[ C^T \]

\[ j_1 \]

\[ j_2 \]

\[ j \]

\[ j_k \]

\[ \in \mathbb{R}^{V_w} \]

\[ \in \mathbb{R}^d \]

\[ \in \mathbb{R}^{V_c} \]

\[ \in \mathbb{R}^{k+1} \]
Objective function SGNS

\[ \ell_{\text{SGNS}} (W, C) = \sum_{(w_i, c_j) \in D} \left( \log \sigma (\vec{w}_i \cdot \vec{c}_j) + \sum_{l=1}^{k} \log (1 - \sigma (\vec{w}_i \cdot \vec{c}_{j_l})) \right) \]

\[ = \sum_{(w_i, c_j) \in D} \left( \log \sigma (\vec{w}_i \cdot \vec{c}_j) + \sum_{l=1}^{k} \log \sigma (-\vec{w}_i \cdot \vec{c}_{j_l}) \right) \]

\[ \approx \sum_{(w, c) \in D} \left( \log \sigma (\vec{w} \cdot \vec{c}) + k \cdot \mathbb{E}_{c_{\sim} \sim P_D} [\log \sigma (-\vec{w} \cdot \vec{c}_{\sim})] \right) \]
Truncated SVD

\[ M = U \Sigma V^T \]

\[ M_d = U_d \Sigma_d V_d^T \]
Spearman correlation

Let $X_i$ be the human-assigned scores and $Y_i$ be the cosine similarity of the vectors. Then, the Spearman correlation is defined as

$$\rho_S = \frac{\text{cov}(\text{rg}(X), \text{rg}(Y))}{\sigma(\text{rg}(X)) \sigma(\text{rg}(Y))} \in [-1, 1].$$

Figure: Datasets with different Spearman correlation

(a) positive  (b) negative  (c) around zero
Analogy Tasks

Figure: Examples of various relations between words