# Recap

Word2Vec

- Method to represent words in vector space
- Train a neural network on a certain task, extract word weights
- Tasks: Skip-gram and continuous bag of words

# Word2Vec (Missing Details)

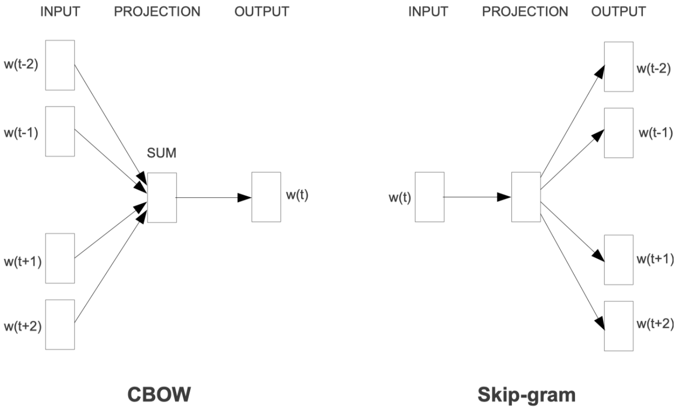
## Sprachverarbeitung (VL + Ü)

Nils Reiter

June 27, 2023

INSTITUT FÜR
DIGITAL HUMANITIES
UNIVERSITÄT ZU KÖLN

# Two tasks




**Continuous Bag of Words (CBOW)**

Context words used to predict a single word

**Skip-Gram**

One word used to predict its context

# Skip-gram

- Intuition: »a word is likely to occur near the target if its embedding is similar to the target embedding« Jurafsky/Martin (JM20, 112)

# Skip-gram

- Intuition: »a word is likely to occur near the target if its embedding is similar to the target embedding« Jurafsky/Martin (JM20, 112)
- Classifier:
  - Predict for $(t, c)$ wether $c$ are *really* context words for $t$
  - Probability of $\vec{t}$ and $\vec{c}$ being positive examples: $p(+|\vec{t}, \vec{c})$
  - Classifier training requires a loss function (as in logistic regression)

# Loss Function

- Maximize $p(+|t, c)$ (positive samples)
- Minimize $p(+|t, c_n)$ (negative samples) $\Rightarrow$ Max. $p(-|t, c_n)$

# Loss Function

- Maximize $p(+|t, c)$ (positive samples)
- Minimize $p(+|t, c_n)$ (negative samples)

$$J(\theta) = \sum_{(t,c)} \log p(+|t, c) + \sum_{(t,c_n)} \log p(-|t, c_n)$$

($\theta$: Concatenation of all $\vec{t}$, $\vec{c}$, $\vec{c}_n$)

# Loss Function

- Maximize $p(+|t, c)$ (positive samples)
- Minimize $p(+|t, c_n)$ (negative samples)

$$J(\theta) = \sum_{(t,c)} \log p(+|t, c) + \sum_{(t,c_n)} \log p(-|t, c_n)$$

($\theta$: Concatenation of all $\vec{t}$, $\vec{c}$, $\vec{c}_n$)

- How to calculate $p(+|t, c)$ and $p(-|t, c_n)$?
- Where to we get negative samples?

# How to Calculate $p(+|t, c)$ and $p(-|t, c_n)$?

- Metric that takes two vectors and returns a similarity score
- Linear algebra: dot product (»Skalarprodukt«)

# How to Calculate $p(+|t, c)$ and $p(-|t, c_n)$?

- Metric that takes two vectors and returns a similarity score
- Linear algebra: dot product (»Skalarprodukt«)

$$\vec{a} \cdot \vec{b} \;=\; \sum_{i=1}^{N} a_i b_i$$

# How to Calculate $p(+|t, c)$ and $p(-|t, c_n)$?

## Dot product

$$\begin{aligned} \vec{a} &= [0, 1, 1] \\ \vec{b} &= [1, 1, 0.5] \\ \vec{a} \cdot \vec{b} &= \end{aligned}$$

$(0 \cdot 1) + (1 \cdot 1) + (1 \cdot 0{,}5)$

$= 0 + 1 + 0{,}5$

$= 1{,}5$

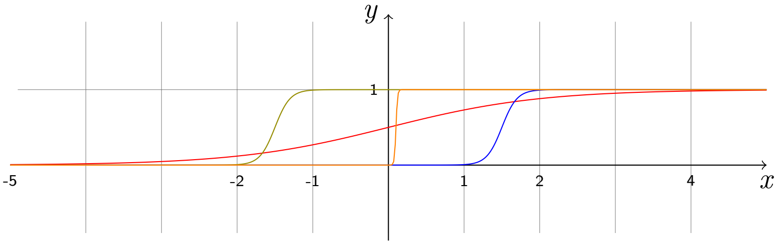
# How to Calculate $p(+|t, c)$ and $p(-|t, c_n)$?

## Dot product

$$
\begin{aligned}
\vec{a} &= [0, 1, 1] \\
\vec{b} &= [1, 1, 0.5] \\
\vec{a} \cdot \vec{b} &= 1.5
\end{aligned}
$$

# How to Calculate $p(+|t, c)$ and $p(-|t, c_n)$?

## The Logistic Function




$$y = \frac{1}{1+e^{-x}} = \frac{1}{1+e^{-(ax+b)}} = \frac{1}{1+e^{-(1*x+0)}}$$

$$y = \frac{1}{1+e^{-(10*x-15)}}$$

$$y = \frac{1}{1+e^{-(10*x+15)}}$$

$$y = \frac{1}{1+e^{-(100*x-10)}}$$

# How to Calculate $p(+|t, c)$ and $p(-|t, c_n)$?

logistic function

dot product

$$
\begin{aligned}
p(+|t, c) &= \frac{1}{1 + e^{-\vec{t}\cdot\vec{c}}} \\
p(-|t, c) &= 1 - p(+|t, c) = 1 - \frac{1}{1 + e^{-\vec{t}\cdot\vec{c}}} = \frac{e^{-\vec{t}\cdot\vec{c}}}{1 + e^{-\vec{t}\cdot\vec{c}}}
\end{aligned}
$$

## How to Calculate $p(+|t, c)$ and $p(-|t, c_n)$?

$$
\begin{aligned}
p(+|t, c) &= \frac{1}{1 + e^{-\vec{t}\cdot\vec{c}}} \\
p(-|t, c) &= 1 - p(+|t, c) = 1 - \frac{1}{1 + e^{-\vec{t}\cdot\vec{c}}} = \frac{e^{-\vec{t}\cdot\vec{c}}}{1 + e^{-\vec{t}\cdot\vec{c}}}
\end{aligned}
$$

### More than one context word

Assumption: They are independent, allowing multiplication

$$
p(+|t, c_{1:k}) = \prod_{i=1}^{k} \frac{1}{1 + e^{-\vec{t}\cdot\vec{c}_i}}
$$

# Where to we get negative samples?

- Negative examples
  - Training a classifier needs negative examples, i.e., words that are not in the context of each other

# Where to we get negative samples?

- Negative examples
  - Training a classifier needs negative examples, i.e., words that are not in the context of each other
- Negative sampling
  - For every positive tuple $(t, c)$, we add $k$ negative tuples
  - Negative tuple $(t, c_n)$, with $c_n$ randomly selected (and $t \neq c_n$)

# Where to we get negative samples?

- Negative examples
  - Training a classifier needs negative examples, i.e., words that are not in the context of each other
- Negative sampling
  - For every positive tuple $(t, c)$, we add $k$ negative tuples
  - Negative tuple $(t, c_n)$, with $c_n$ randomly selected (and $t \neq c_n$)
  - Select noise words according to their weighted frequency
  - $p_\alpha(w) = \frac{count(w)^\alpha}{\sum_{w'} count(w')^\alpha}$
    - This leads to rare words being more frequently selected, frequent words less

# Where to we get negative samples?

- Negative examples
  - Training a classifier needs negative examples, i.e., words that are not in the context of each other
- Negative sampling
  - For every positive tuple $(t, c)$, we add $k$ negative tuples
  - Negative tuple $(t, c_n)$, with $c_n$ randomly selected (and $t \neq c_n$)
  - Select noise words according to their weighted frequency
  - $p_\alpha(w) = \frac{count(w)^\alpha}{\sum_{w'} count(w')^\alpha}$
    - This leads to rare words being more frequently selected, frequent words less
- Two new 'parameters' on this slide: $k$ and $\alpha$
  - They have a different status than $\theta$ (the parameters we want to learn)
  - Therefore: Hyperparameters

# Remarks and observations

- Each word is used twice, with different roles
  - As target word (for predicting its context)
  - As context word (to be predicted from another target word)
  - Different options: Only use one embedding, combine them by addition or concatenation

# Remarks and observations

- Each word is used twice, with different roles
  - As target word (for predicting its context)
  - As context word (to be predicted from another target word)
  - Different options: Only use one embedding, combine them by addition or concatenation
- Matrices
  - Conceptually, it is not hugely important how the embeddings are stored in detail
  - But for the implementation because of efficiency
  - All target vectors are stored in matrix $W$ (word matrix)
  - All context vectors are stored in matrix $C$ (context matrix)
  - $\theta = (W, C)$

# Section 1

## Bias in Embeddings

# Bias in Embeddings

- Important discussion: How *biased* are embeddings?
- And related: How can we measure it?

# Bias in Embeddings

- Important discussion: How *biased* are embeddings?
- And related: How can we measure it?
- WEAT: Word-Embedding Association Test Caliskan et al. (2017)
  - Inspired by Implicit Association Test, used in pychology/psycho linguistics Greenwald et al. (1998)
- Measures association between word groups

# WEAT

- Two sets of target words $(X,\ Y)$
  - E.g., programmer/scientist/engineer vs. nurse/teacher/librarian
- Two sets of attribute words $(A,\ B)$
  - E.g., man/male vs. woman/female

# WEAT

- Two sets of target words $(X,\ Y)$
  - E.g., programmer/scientist/engineer vs. nurse/teacher/librarian
- Two sets of attribute words $(A,\ B)$
  - E.g., man/male vs. woman/female
- Null hypothesis: Target word sets are equally similar to both attribute words

# WEAT

- Two sets of target words $(X,\ Y)$
  - E.g., programmer/scientist/engineer vs. nurse/teacher/librarian
- Two sets of attribute words $(A,\ B)$
  - E.g., man/male vs. woman/female
- Null hypothesis: Target word sets are equally similar to both attribute words

$$s(X, Y, A, B) = \sum_{x \in X} s(x, A, B) - \sum_{y \in Y} s(y, A, B)$$

$$s(w, A, B) = \frac{1}{|A|} \sum_{a \in A} \cos(\vec{w}, \vec{a}) - \frac{1}{|B|} \sum_{b \in B} \cos(\vec{w}, \vec{b})$$

# WEAT

$$
\begin{aligned}
s(X, Y, A, B) &= \sum_{x \in X} s(x, A, B) - \sum_{y \in Y} s(y, A, B) \\
s(w, A, B) &= \frac{1}{|A|} \sum_{a \in A} \cos(\vec{w}, \vec{a}) - \frac{1}{|B|} \sum_{b \in B} \cos(\vec{w}, \vec{b})
\end{aligned}
$$

*In other words, $s(w, A, B)$ measures the association of $w$ with the attribute, and $s(X, Y, A, B)$ measures the differential association of the two sets of target words with the attribute.* *(Caliskan et al., 2017, 184)*

# Example

1. Expected, inoffensive bias
   - Flowers (aster, clover, ...) vs. Insects (ant, caterpillar, ...)
   - Pleasant (caress, freedom, ...) vs. Unpleasant (abuse, crash, ...)
2. Offensive bias
   - Science (science, technology, ...) vs. Arts (poetry, art, ...)
   - Male (brother, father, ...) vs. Female (sister, mother, ...)

# Exercise

The word sets used by Caliskan et al., 2017 can be found here:
`https://www.science.org/action/downloadSupplement?doi=10.1126%2Fscience.aal4230&file=caliskan-sm.pdf`,
two files are stored in `/teaching/summer-2023/sprachverarbeitung/data/weat1.txt` resp. `weat8.txt`.

- Identify (small) sets of words for which you expect bias in the embeddings you've trained last week. Verify that the words actually are in the embeddings.
- Perform a word embeddings association test.