

The skip-gram model

Marco Kuhlmann

Department of Computer and Information Science

The skip-gram model

Eisenstein § 14.5.2

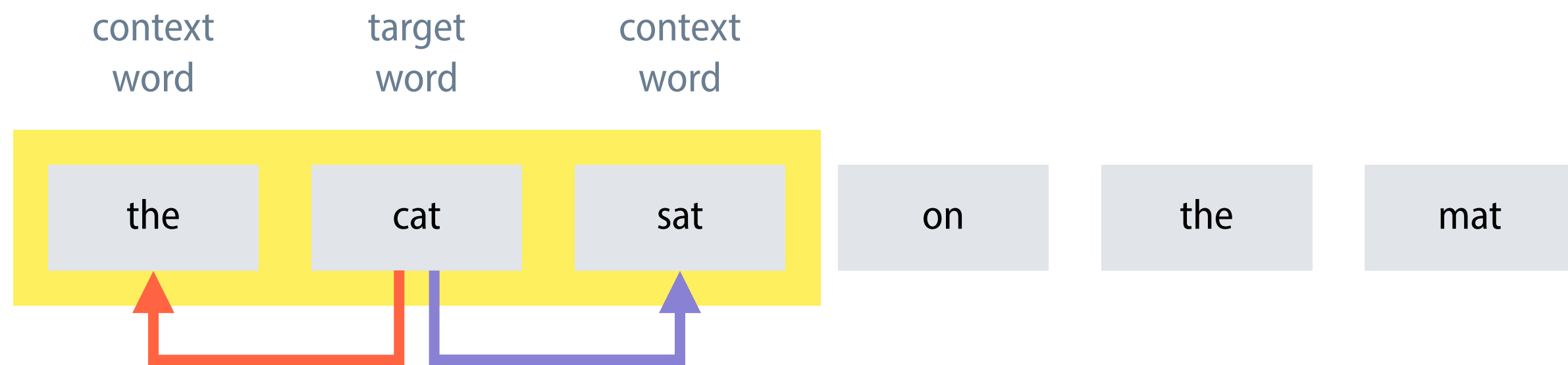
- The **skip-gram model** is one of two word embedding models implemented in Google's word2vec software.
- In the context of this model, a **skip-gram** is a pair of words from a text that are separated by at most k other words.
- The word embeddings are obtained as a by-product of the task to predict one word in the skip-gram from the other word.

[Mikolov et al. \(2013\)](#)

Training the skip-gram model

- Start with random word vectors.
- Move a small, symmetric window over the words in a text. Each window contains a target word w and context words c .
- For each window, use the similarity of the current word vectors for w and c to define a conditional probability $P(c|w)$.
- Tweak the word vectors to maximize this probability.

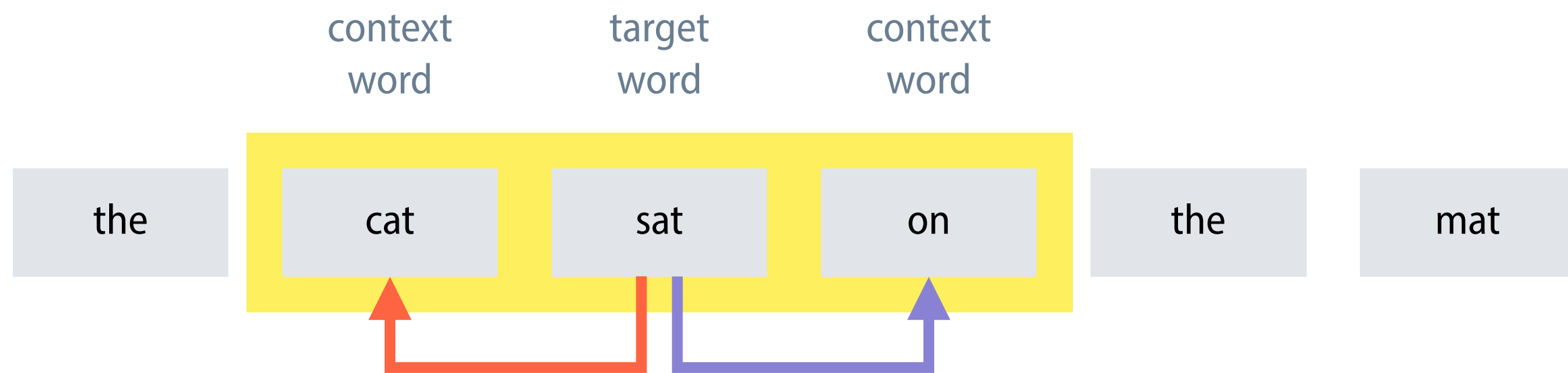
Training the skip-gram model



$$P(\text{the} \mid \text{cat}) \propto \mathbf{v}'_{\text{the}}{}^T \mathbf{v}_{\text{cat}} \quad P(\text{sat} \mid \text{cat}) \propto \mathbf{v}'_{\text{sat}}{}^T \mathbf{v}_{\text{cat}}$$

target word vectors	\mathbf{v}_{cat}	\mathbf{v}_{mat}	\mathbf{v}_{on}	\mathbf{v}_{sat}	\mathbf{v}_{the}
context word vectors	\mathbf{v}'_{cat}	\mathbf{v}'_{mat}	\mathbf{v}'_{on}	\mathbf{v}'_{sat}	\mathbf{v}'_{the}

Training the skip-gram model



$$P(\text{cat} \mid \text{sat}) \propto \mathbf{v}'_{\text{cat}}{}^T \mathbf{v}_{\text{sat}} \quad P(\text{on} \mid \text{sat}) \propto \mathbf{v}'_{\text{on}}{}^T \mathbf{v}_{\text{sat}}$$

target word vectors	\mathbf{v}_{cat}	\mathbf{v}_{mat}	\mathbf{v}_{on}	\mathbf{v}_{sat}	\mathbf{v}_{the}
context word vectors	\mathbf{v}'_{cat}	\mathbf{v}'_{mat}	\mathbf{v}'_{on}	\mathbf{v}'_{sat}	\mathbf{v}'_{the}

The skip-gram model in detail (1)

- We maintain two separate vector representations: one for target words and one for context words. Initially, they are random.
- The probability of a context word c given a target word w is defined using the softmax function:

The diagram shows the softmax function for the skip-gram model. The equation is
$$P(c | w; \theta) = \frac{\exp(\mathbf{v}'_c \top \mathbf{v}_w)}{\sum_{x \in V} \exp(\mathbf{v}'_x \top \mathbf{v}_w)}$$
 Annotations include: 'vector representation for context words' pointing to \mathbf{v}'_c , 'vector representation for target words' pointing to \mathbf{v}_w , and 'all parameters of the model' pointing to θ .

vector representation for context words

vector representation for target words

$$P(c | w; \theta) = \frac{\exp(\mathbf{v}'_c \top \mathbf{v}_w)}{\sum_{x \in V} \exp(\mathbf{v}'_x \top \mathbf{v}_w)}$$

all parameters of the model

The skip-gram model in detail (2)

To *maximize* the conditional probabilities, we *minimize* the cross-entropy loss on the training data:

$$J(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{i=1}^N \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \log P(w_{i+j} | w_i; \boldsymbol{\theta})$$

all parameters of the model

length of the text

size of each window

Computational complexity

Eisenstein § 14.5.3

Computing the softmax is expensive: For each position in the text, we need to sum over the complete vocabulary.

- **Solution 1:** Decompose the standard softmax computation into a tree-like structure of simpler computations.

hierarchical softmax

- **Solution 2:** Instead of maximizing the conditional probabilities directly, maximize simpler quantities that approximate them.

negative sampling

Skip-gram with negative sampling

Eisenstein § 14.5.3

- Maximize the probability of observed word–context pairs, while minimizing the probability of randomly drawn samples.

$$J(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{i=1}^N \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \log \sigma(\mathbf{v}'_{w_{i+j}} \top \mathbf{v}_{w_i}) + \sum_{c \sim D} \log \sigma(-\mathbf{v}'_c \top \mathbf{v}_{w_i})$$

length of the text

logistic function

negative samples

- The negative samples are drawn from $D(c) \propto \#(c)^\alpha$, where α is a hyperparameter (default value: 0.75).

Skip-gram with negative sampling in detail

- Subsampling: To reduce the influence of very frequent words (and speed up learning), discard a token w with probability

$$P(w) = \max\left(0, 1 - \sqrt{tN / \#(w)}\right)$$

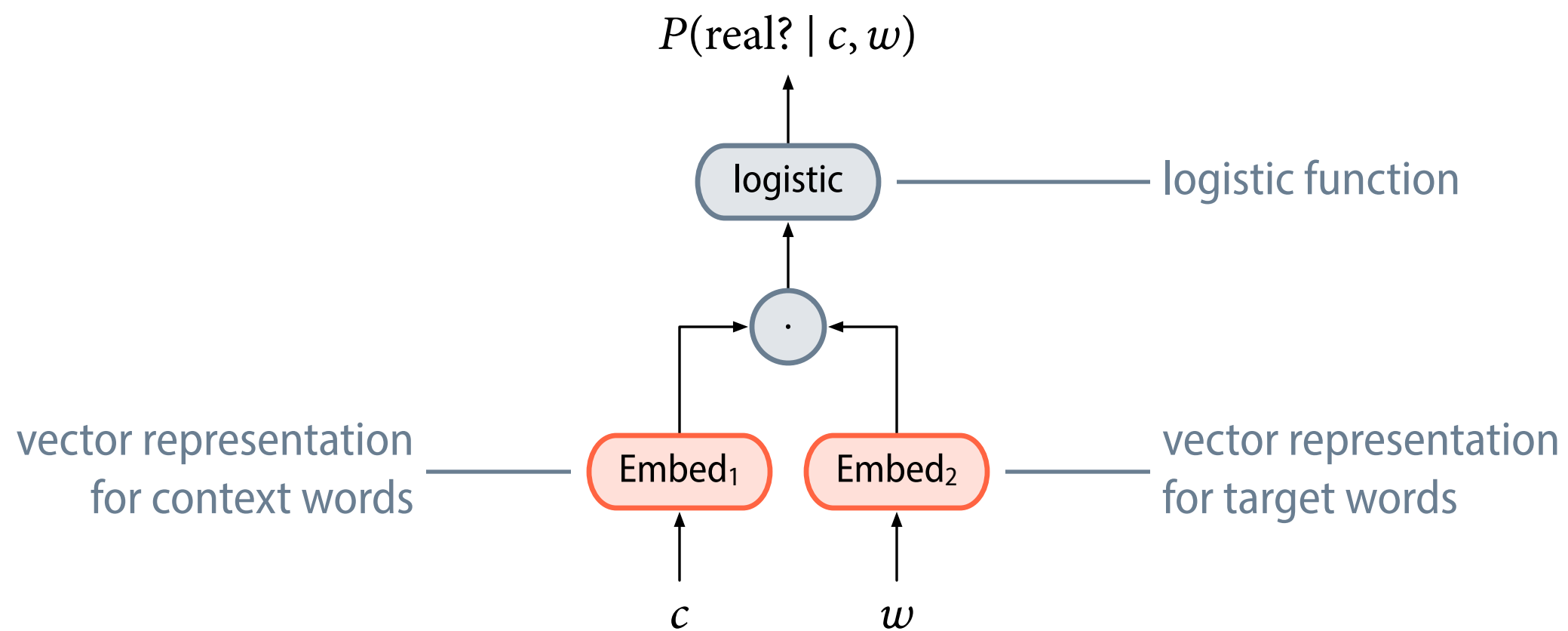
count of the
word w

where t is a chosen threshold (default value: 0.001).

- Do not use a constant window size; instead, sample window sizes up to the maximum size m with uniform probability.

As a consequence, far-away context words will get less influence.

The SGNS model as a neural network



Connecting the two worlds

- Count-based methods and prediction-based methods take two seemingly very different approaches to word embeddings.
- However, a mathematical analysis reveals that the skip-gram model is implicitly factorizing a PMI co-occurrence matrix.
Each entry is shifted by $-\log k$, where k is the number of negative samples.