Types of Embeddings -1

- The initial types of embeddings focused on representing the words themselves
  - An embedding of a word is a representation in latent low dimension space to preserve the properties of—and the relationships between—the words
  - latent dimensions are needed to avoid sparsity and high dimensions
- Typical example: Word2Vec
  - Represent each word with a low-dimensional vector
  - Word similarity = vector similarity
  - Key idea: Using cooccurring words within a context window of a word
  - Faster and can easily incorporate a new sentence/document or add a word to the vocabulary

Context

- Context can be anything – a surrounding n-gram, a randomly sampled set of words from a fixed size window around the word

For example, assume context is defined as the word following a word.

i.e. $\boldsymbol{context(w_i) = w_{i+1}}$

Corpus : I ate the cat

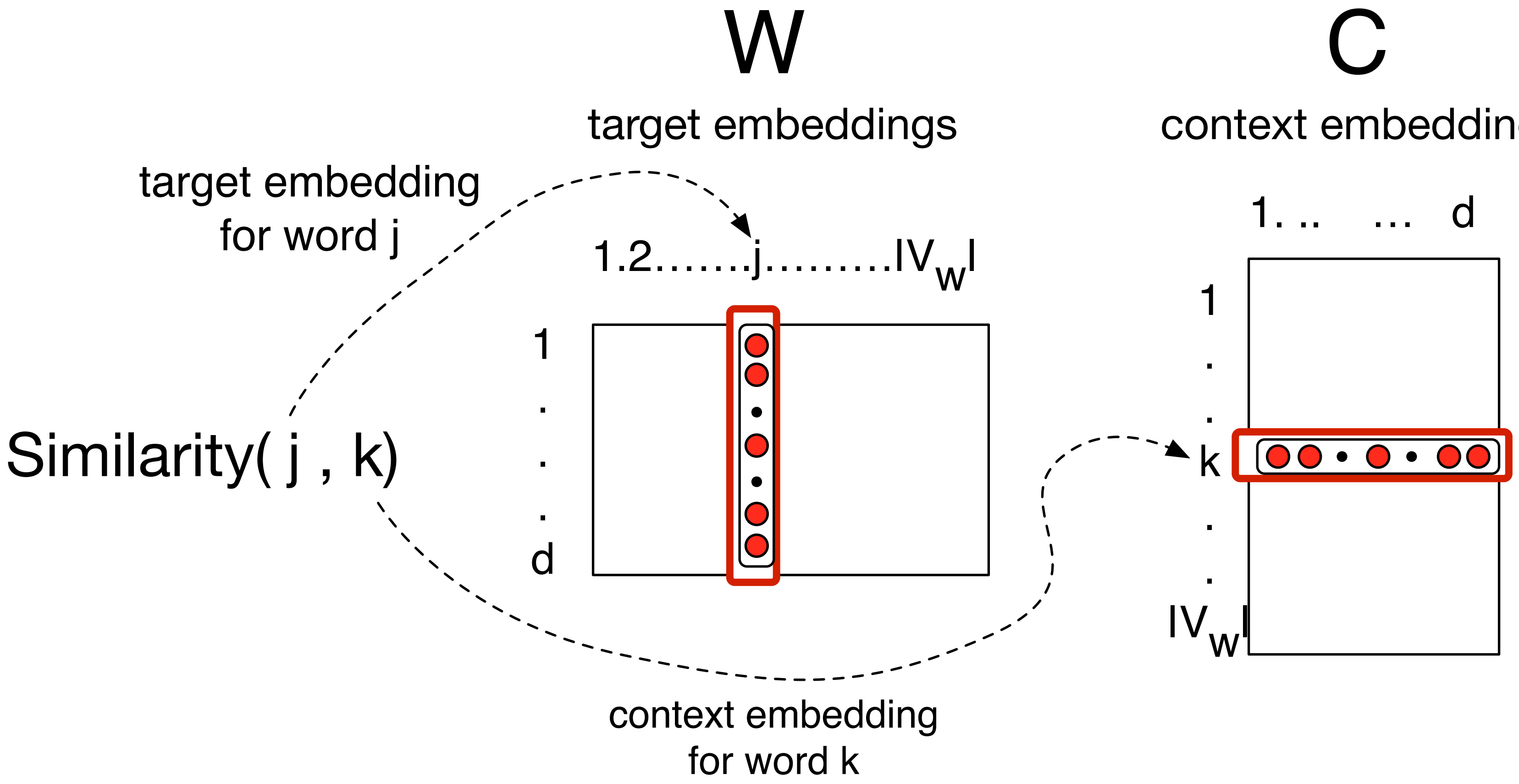
Training Set : I|ate, ate|the , the|cat, cat|.

<u>Collection of training documents :</u>

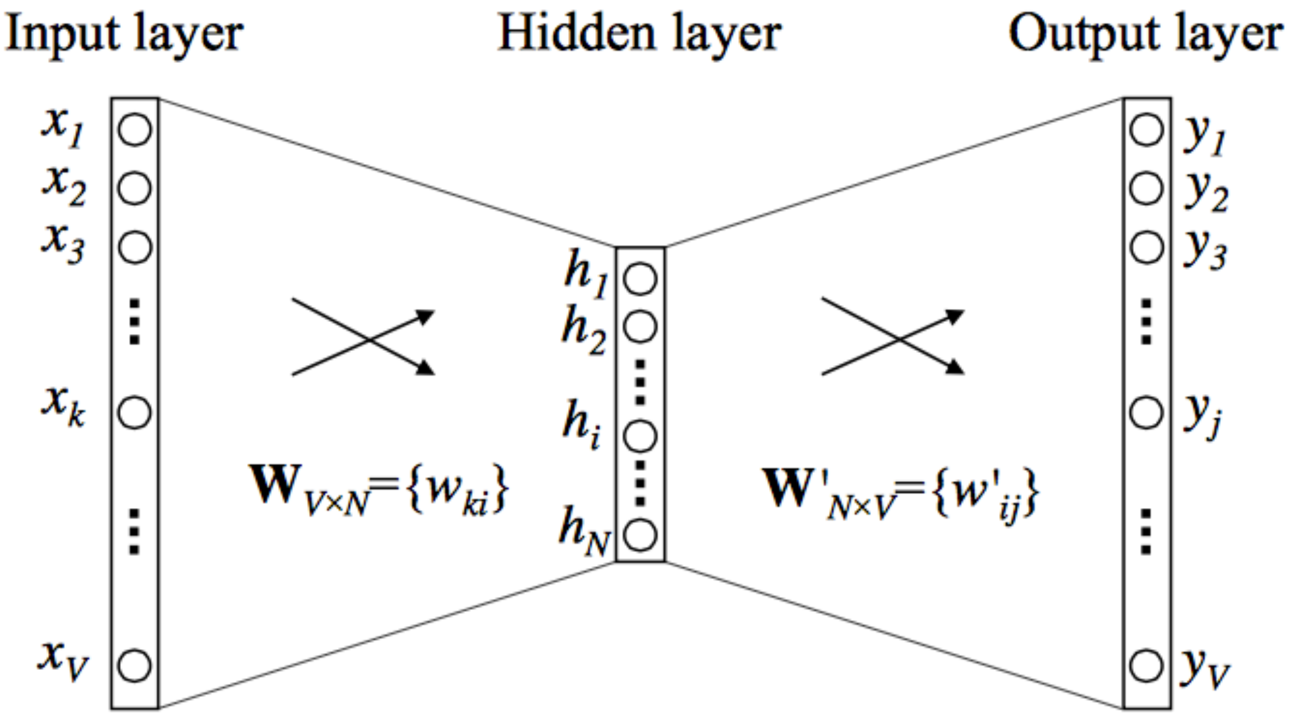
1. Drink milk and drink Juice
2. Eat apples, eat oranges and eat rice
3. Apple juice and Orange juice are juices
4. Rice milk is a actually a type of milk!

1. eat|apple
2. eat|orange
3. eat|rice
4. drink|juice
5. drink|milk
6. drink|water
7. orange|juice
8. apple|juice
9. rice|milk

Intuition 1: High Similarity between a Word Vector and its Context Vector



Intuition2: Information Bottleneck

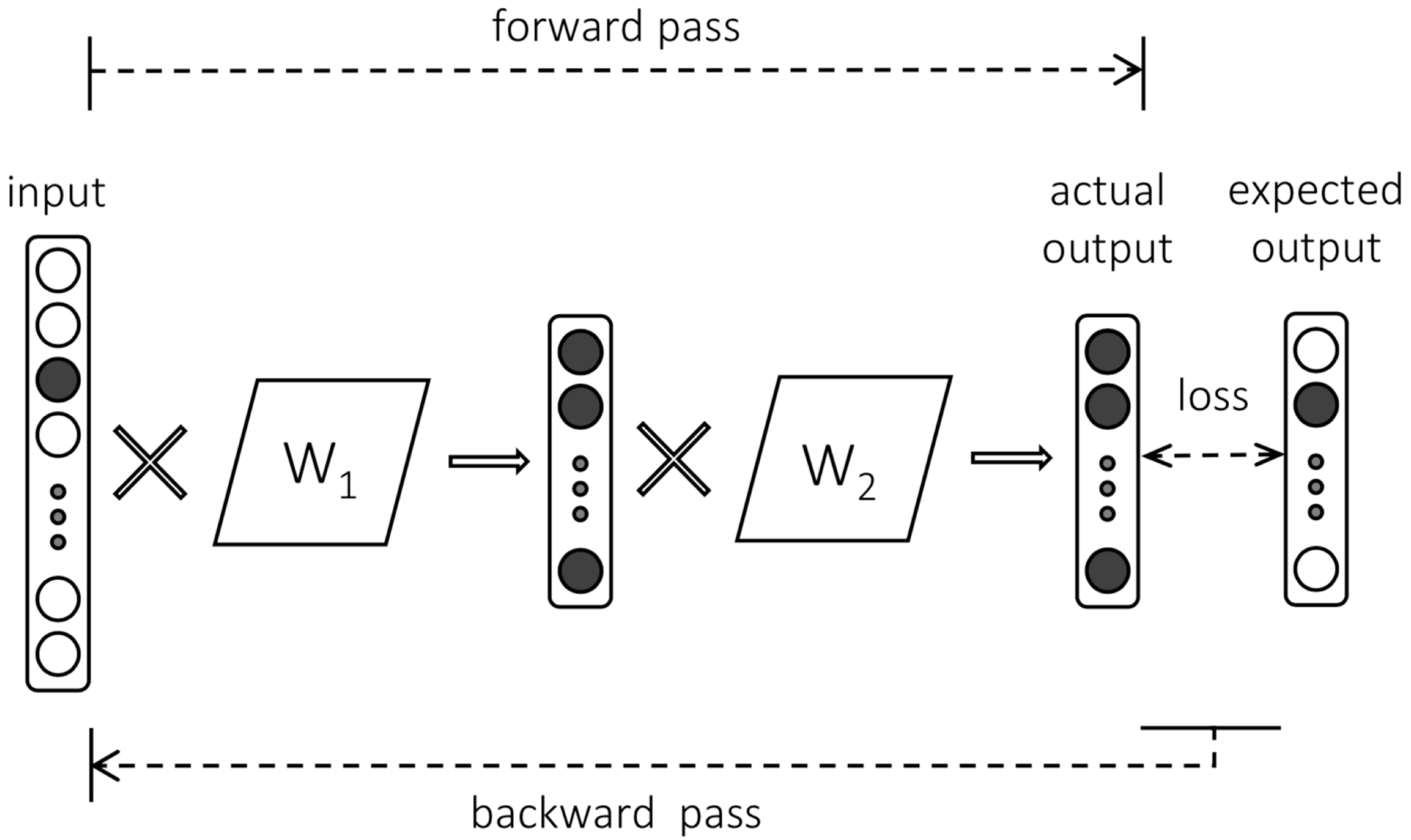


$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\ne 0}\log p(w_{t+j}|w_t)$$

$$p(w_O|w_I)=\frac{\exp\left({v'_{w_O}}^{\top}v_{w_I}\right)}{\sum_{w=1}^{W}\exp\left({v'_{w}}^{\top}v_{w_I}\right)}$$

Key idea is that N is much smaller than V

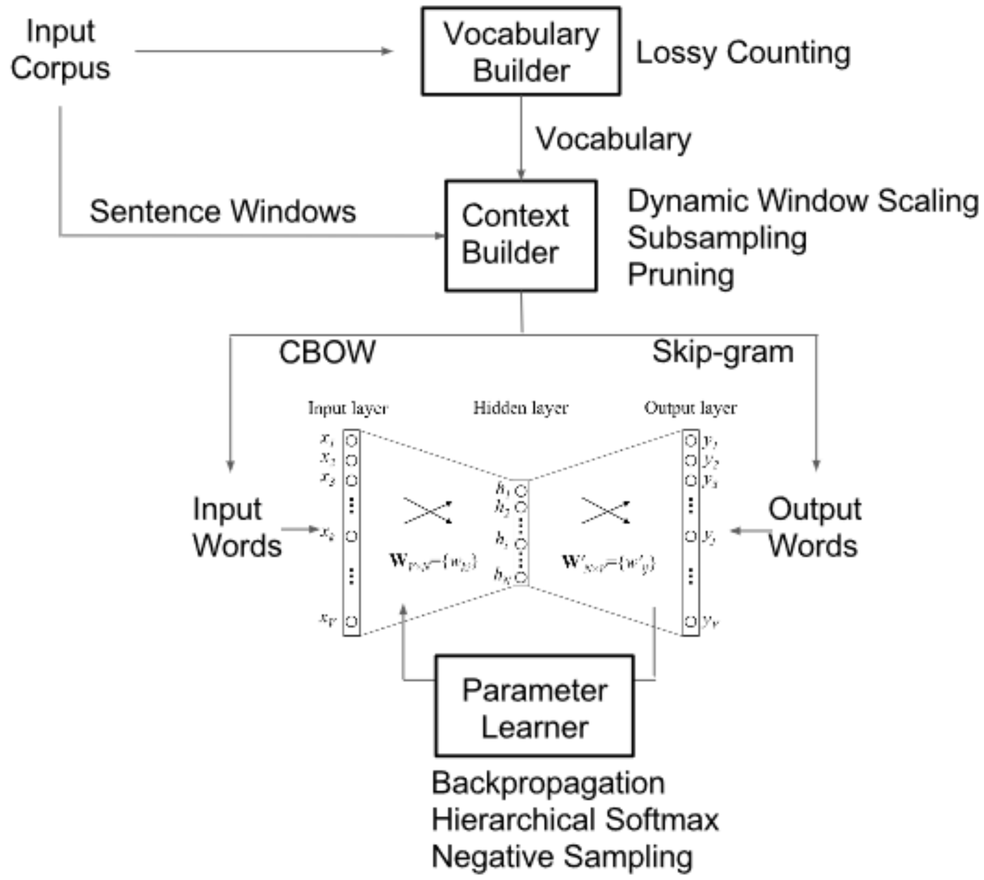
Example Neural Network



$$\vec{y} = tanh(W_2 \cdot tanh(W_1 \cdot \vec{x} + \vec{b}_1) + \vec{b}_2)$$

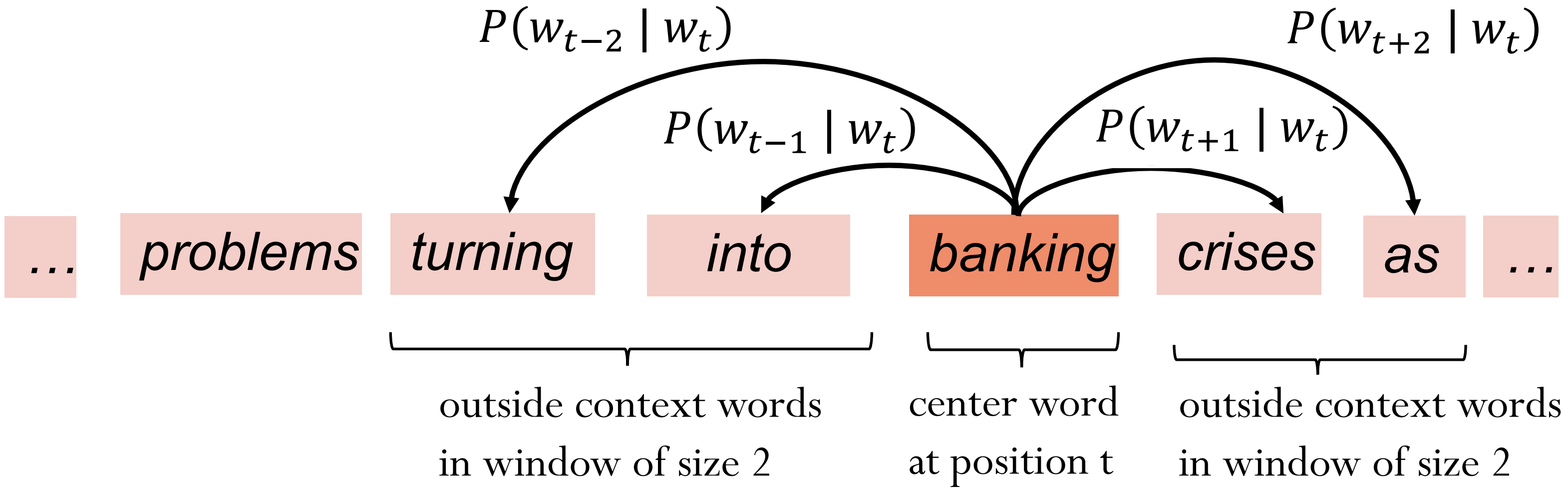
Word2Vec

- **Skip-gram** (Mikolov et al. 2013a) **CBOW** (Mikolov et al. 2013b)
- Learn embeddings as part of the process of word prediction.
- Train a neural network to predict neighboring words
  - Inspired by **neural net language models**.
  - In so doing, learn dense embeddings for the words in the training corpus.
- Advantages:
  - Fast, easy to train (much faster than SVD)
  - Available online in the `word2vec` package
  - Including sets of pretrained embeddings!



Word2Vec Idea 1: Skip-grams

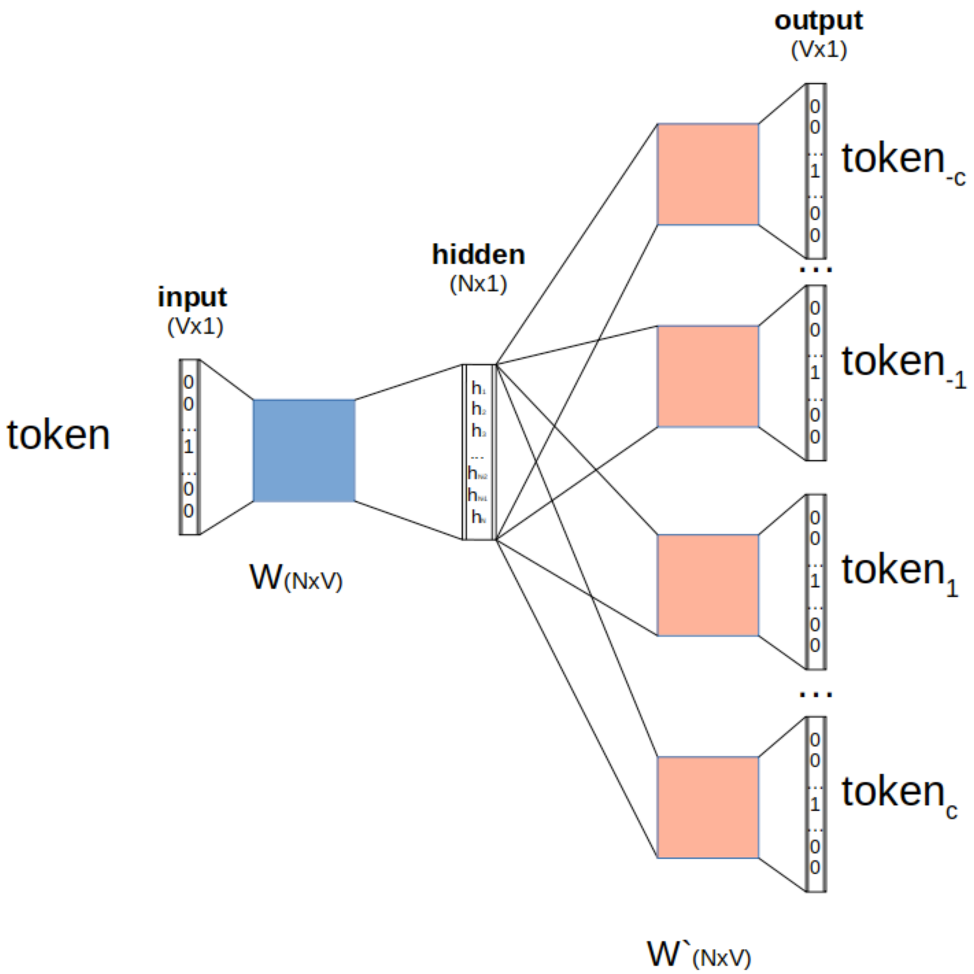
- Walking through the whole corpus, and currently pointing at word w_t, whose index in the vocabulary is t, so we'll call it w_t $(1 < t < |V|)$.
- The skip-gram model predicts each context words, whose index in the vocabulary is t+j (1<t+j<| V |). Hence our task is to compute $P(w_{t+j}|w_t)$.



Word2Vec Idea 1: Skip-grams

- Predict each neighboring word in a context window of $2C$ words from the current word.
  - So for C=2, we are given word w_t and predicting these 4 words:

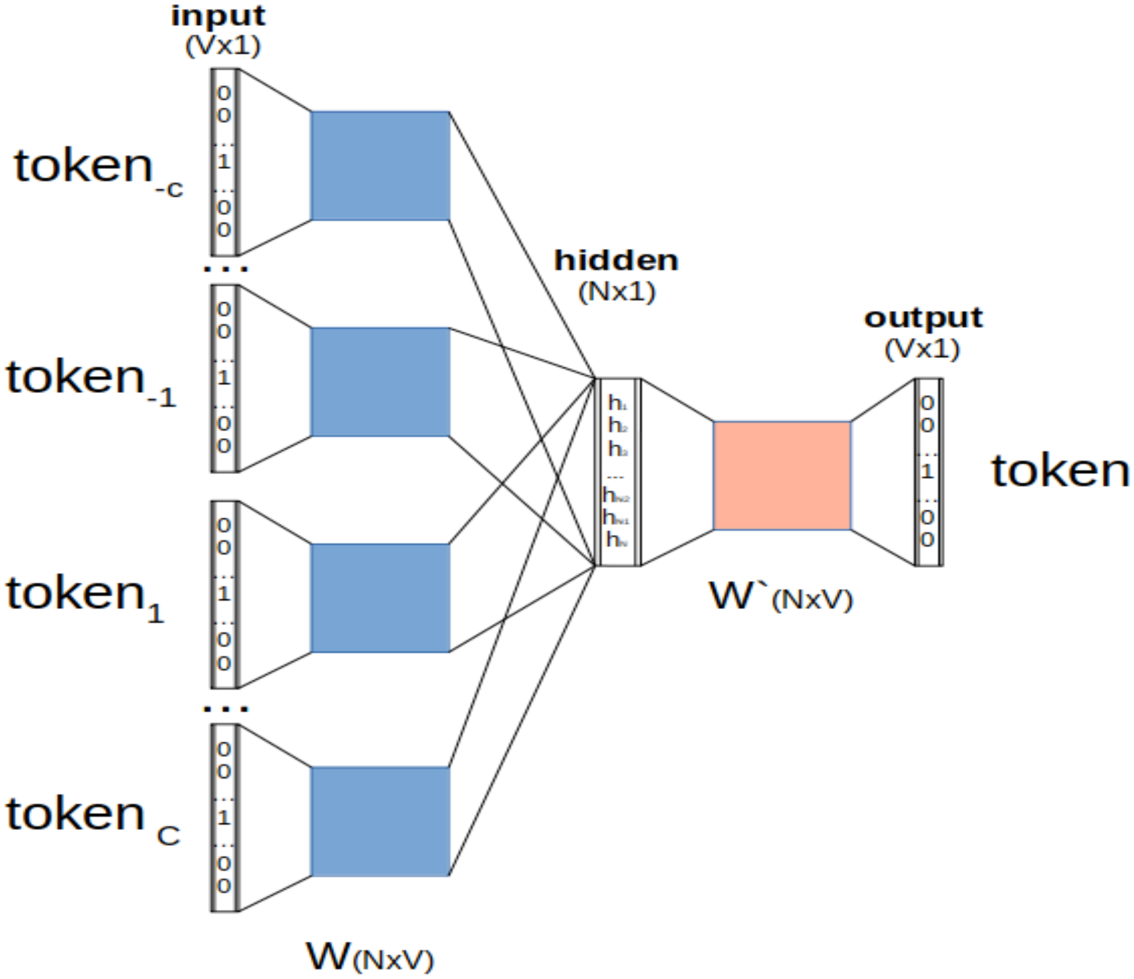
$$[w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2}]$$



Word2Vec Idea 2: CBOW

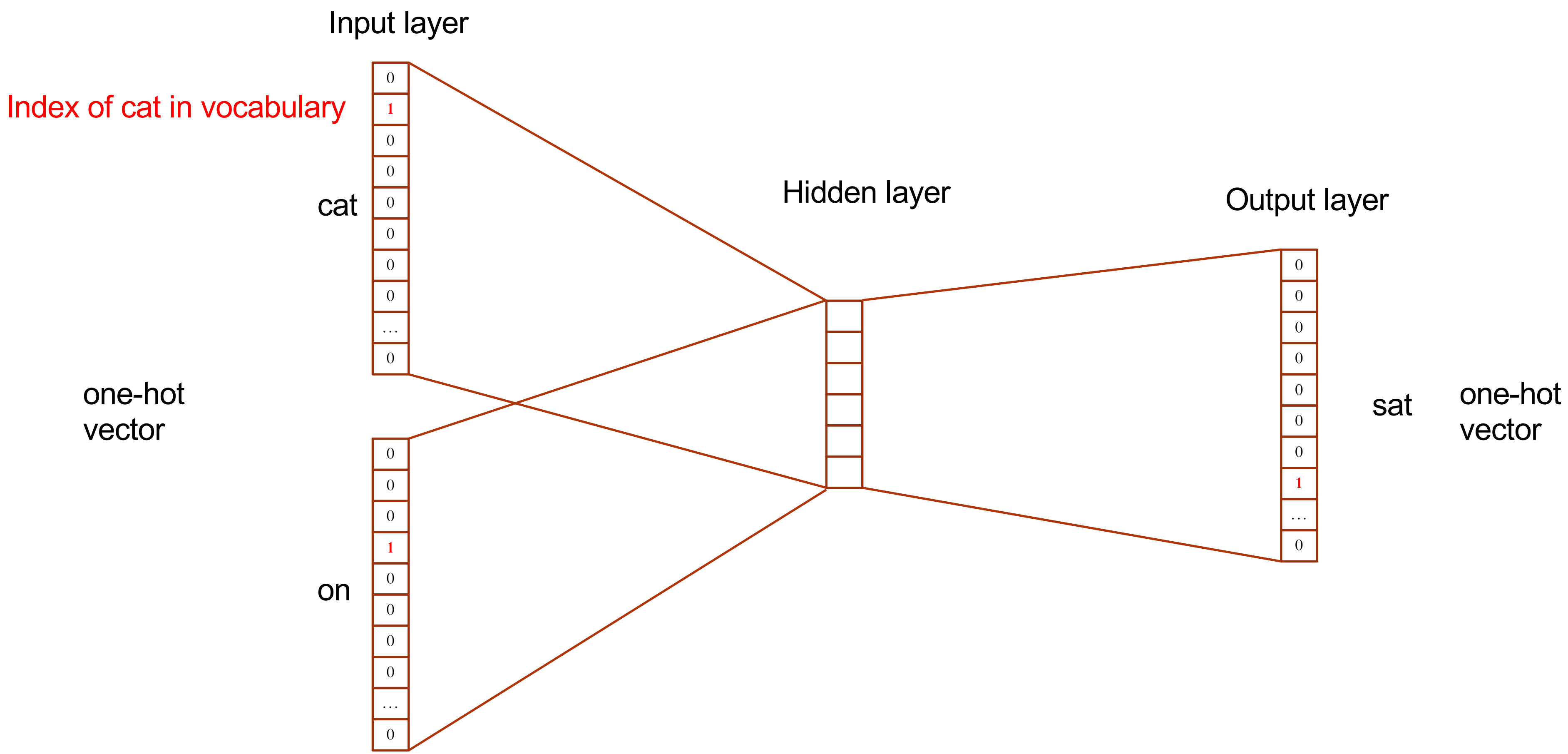
- Continuous Bag-of-Word

$$\mathcal{L}_{CBOW} = -\frac{1}{|S|}\sum_{i=1}^{|S|} log(p(t_i|t_{i-c},\ldots,t_{i-1},t_{i+1},\ldots,t_{i+c}))$$

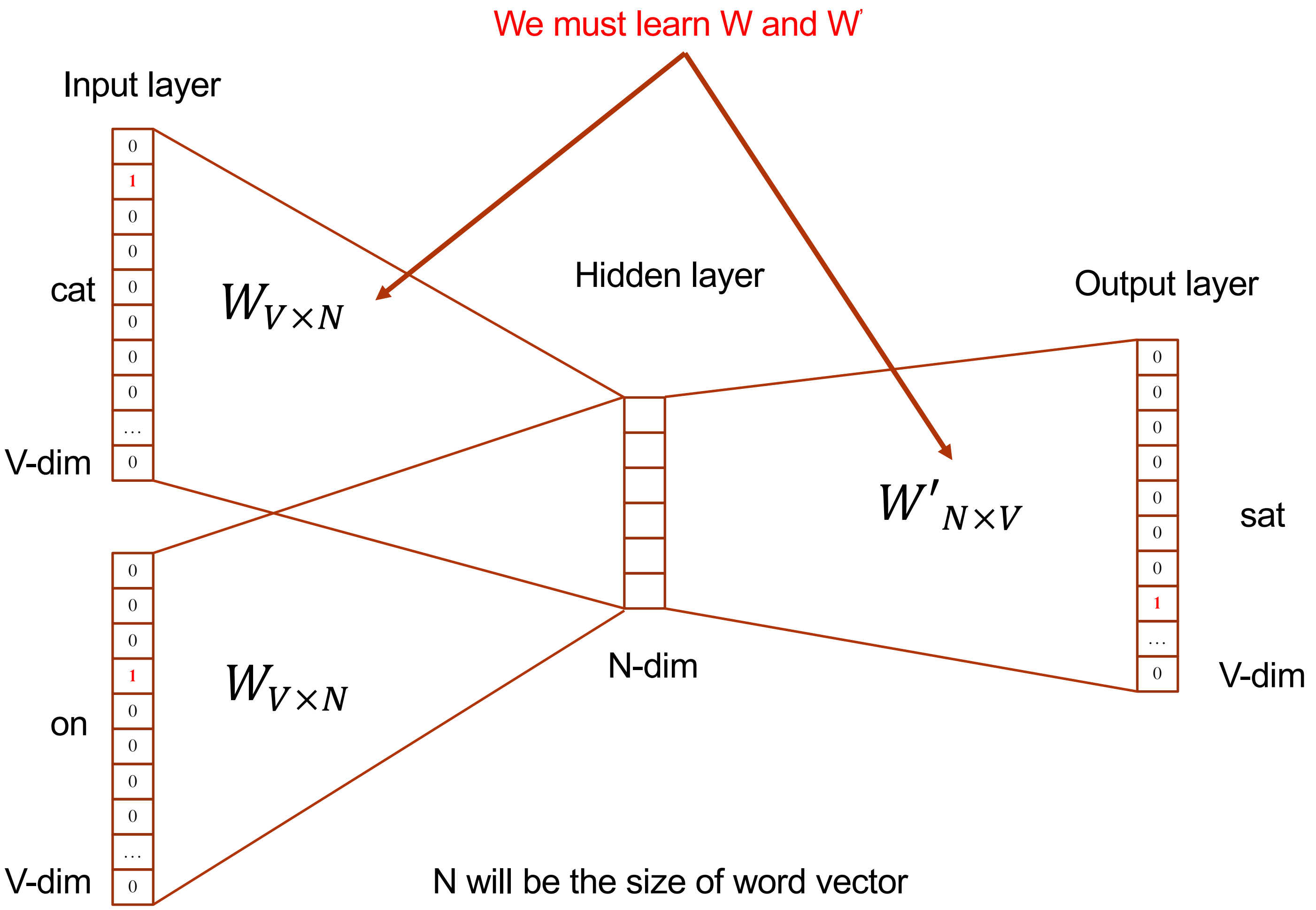


One example implementation of CBOW (self-review)

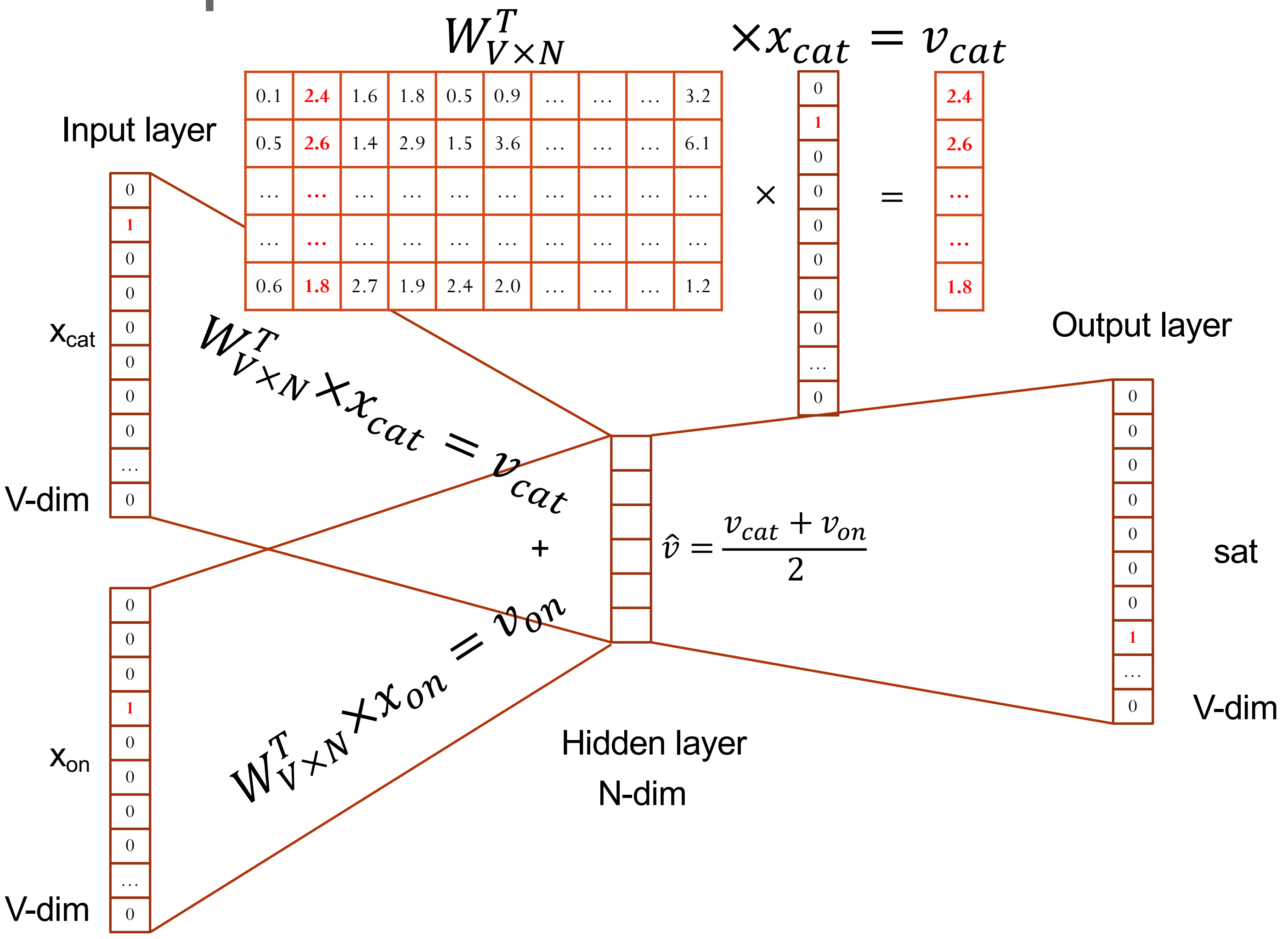
Sentence is “The cat sat on floor”



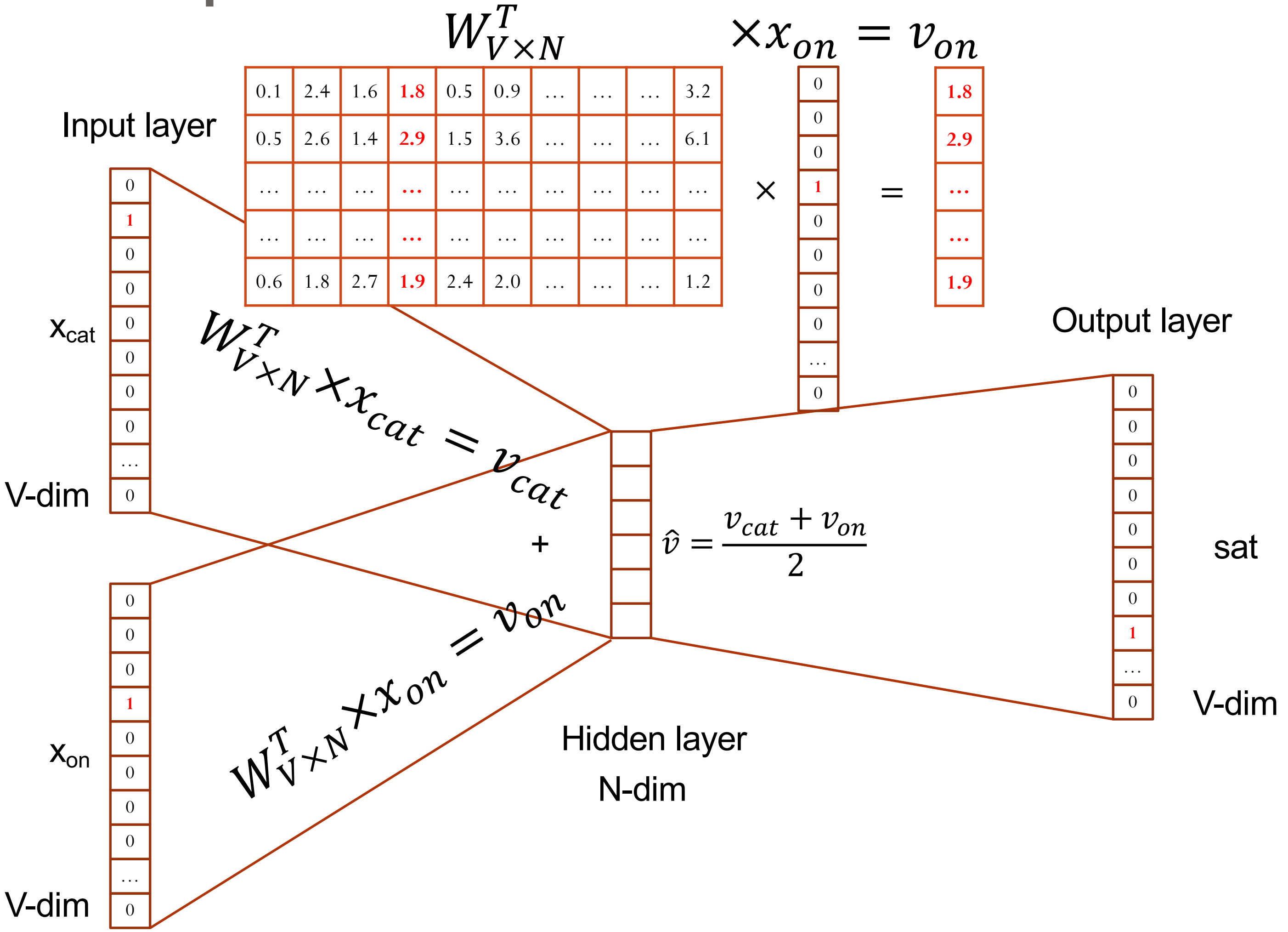
One example implementation of CBOW



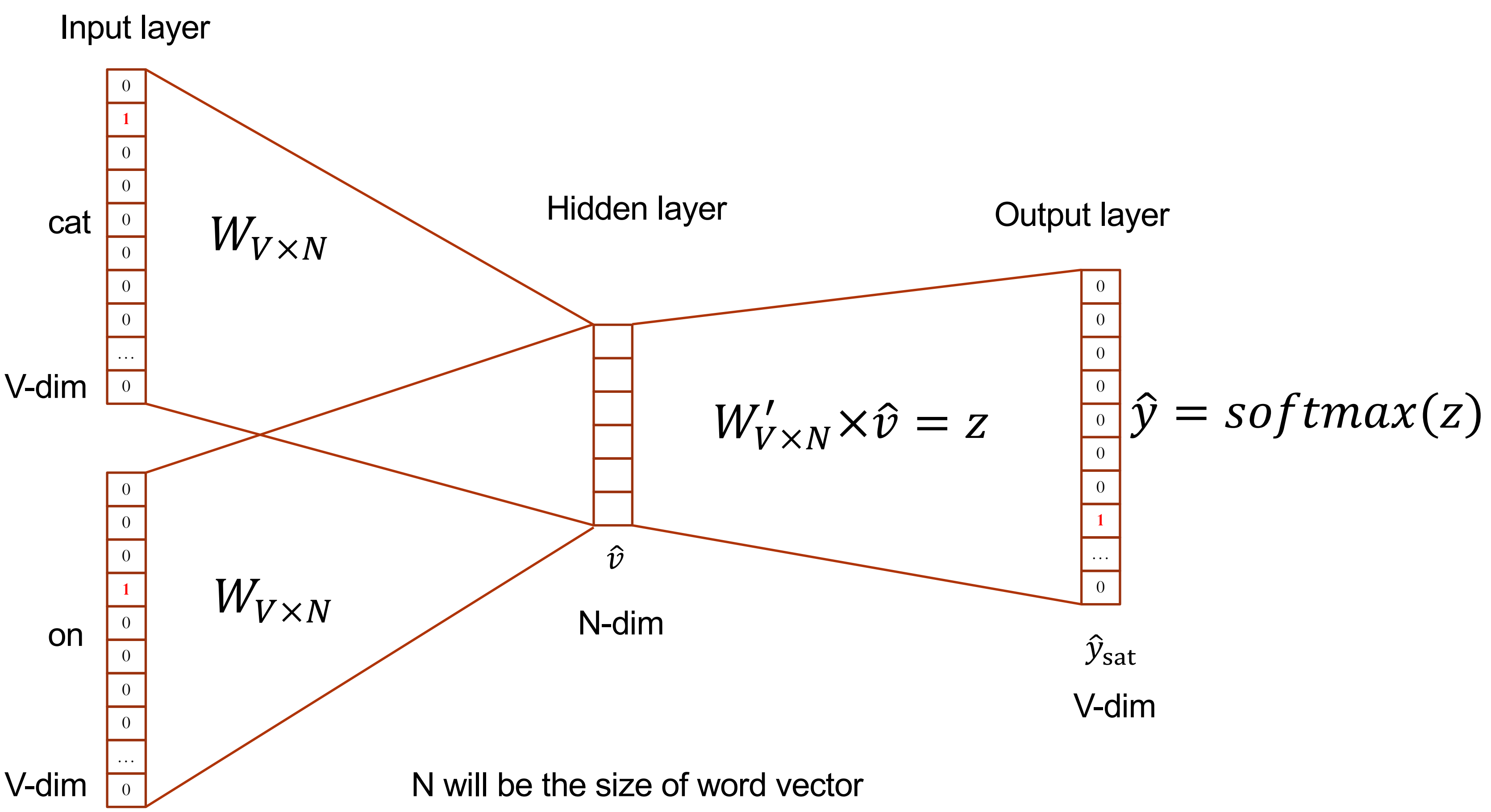
One example implementation of CBOW



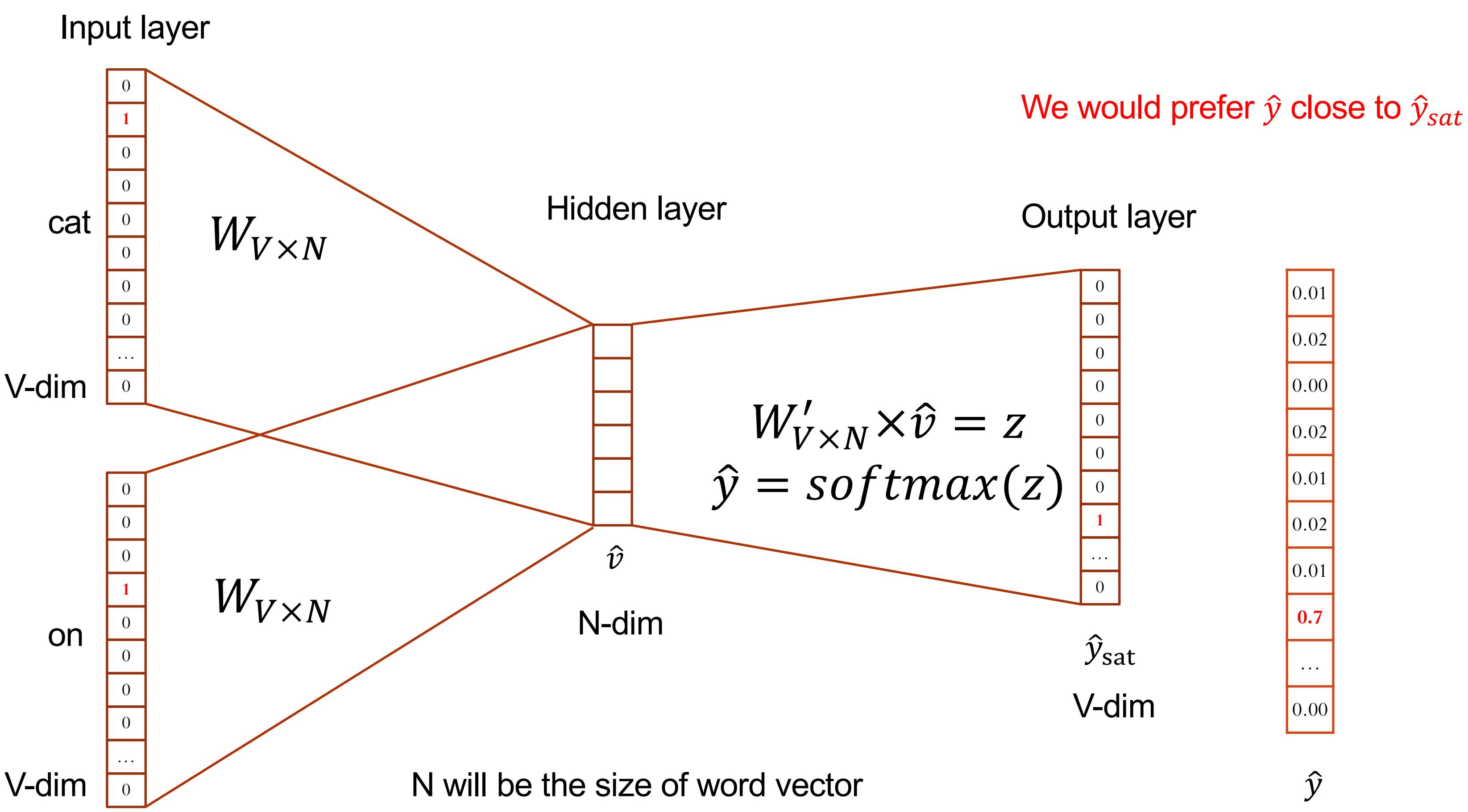
One example implementation of CBOW



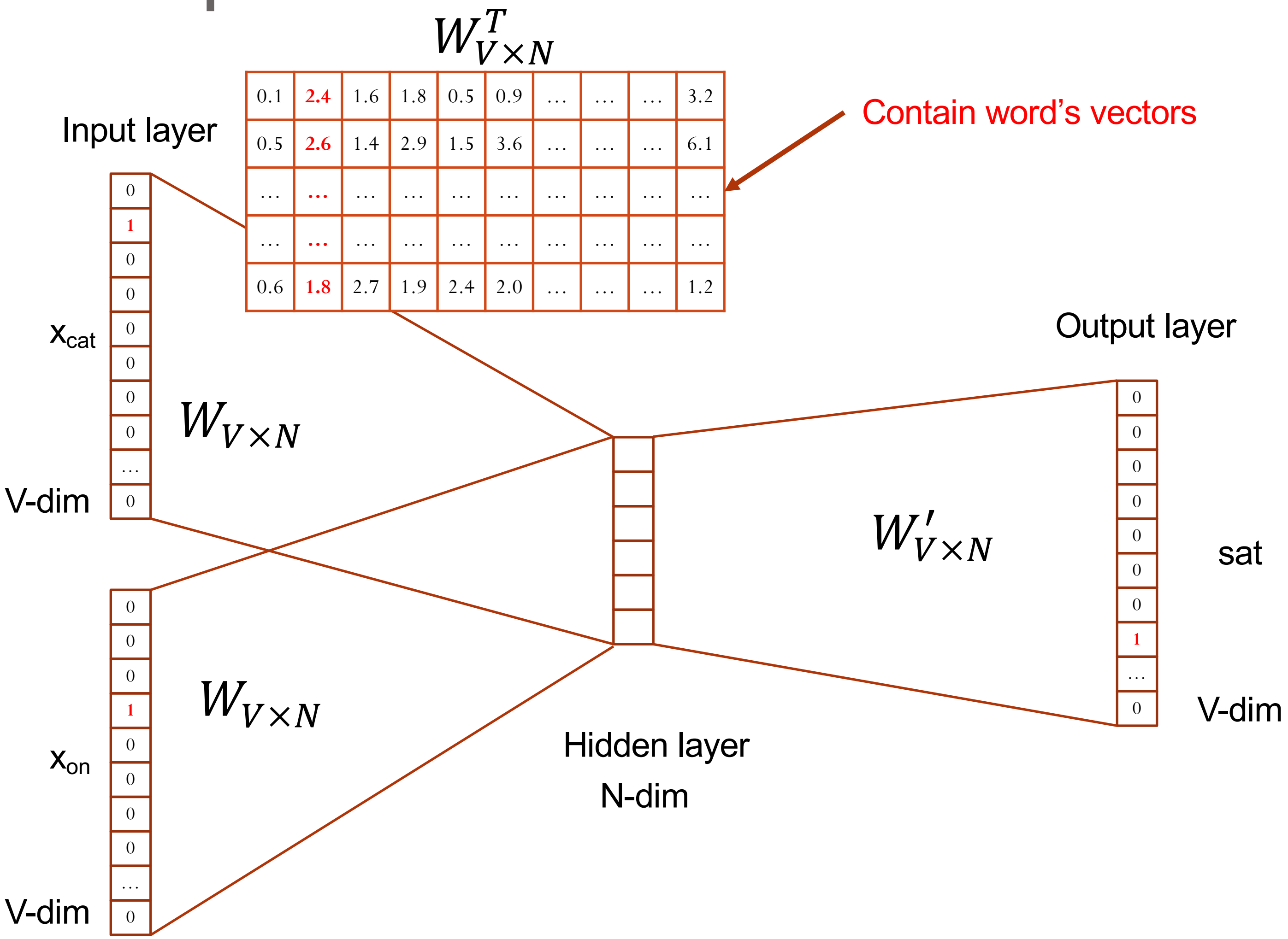
One example implementation of CBOW



One example implementation of CBOW



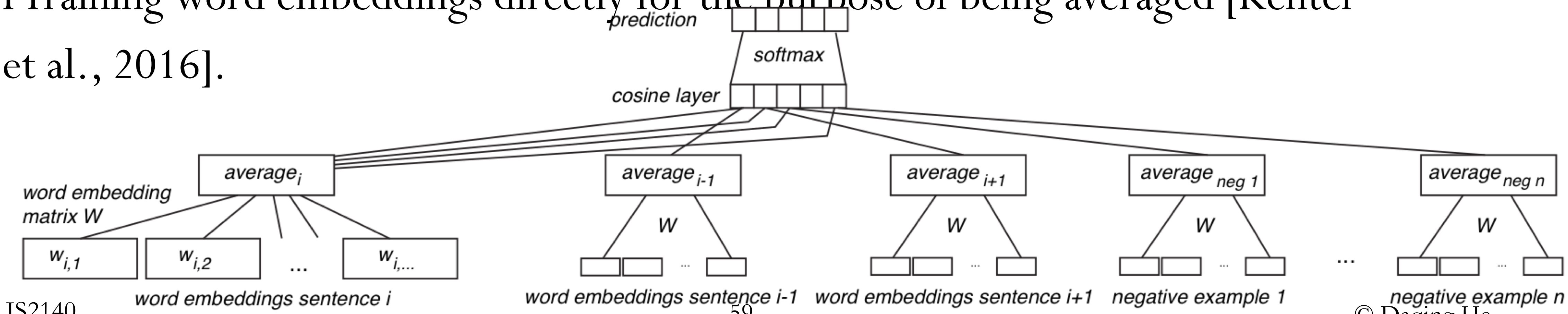
One example implementation of CBOW



We can consider either W or W' as the word's representation.
Or even take the average.

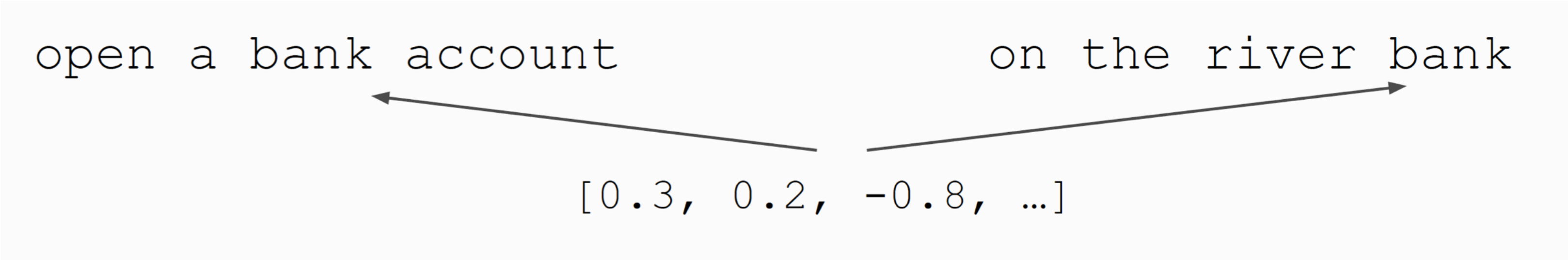
Word Embeddings to Query/Doc Embeddings

- Documents and queries are not just words, they are sequences of words
  - Obtaining embeddings of the atomic words.
  - Bag of embedded words: sum or average of word vectors.
- Averaging the word representations of query terms has been extensively explored in different settings. [Vulic and Moens, 2015, Zamani and Croft, 2016b]
- I Eective but for small units of text, e.g. query [Mitra, 2015].
- I Training word embeddings directly for the purpose of being averaged [Kenter
- et al., 2016].



Types of Embeddings -2

- Limitations of Word2Vec
  - One vector for each word
    - E.g., v(bank) = <0.3, 0.2, -0.8, …>
- Words don't appear in isolation. The word use (e.g., semantics) depends on its context.

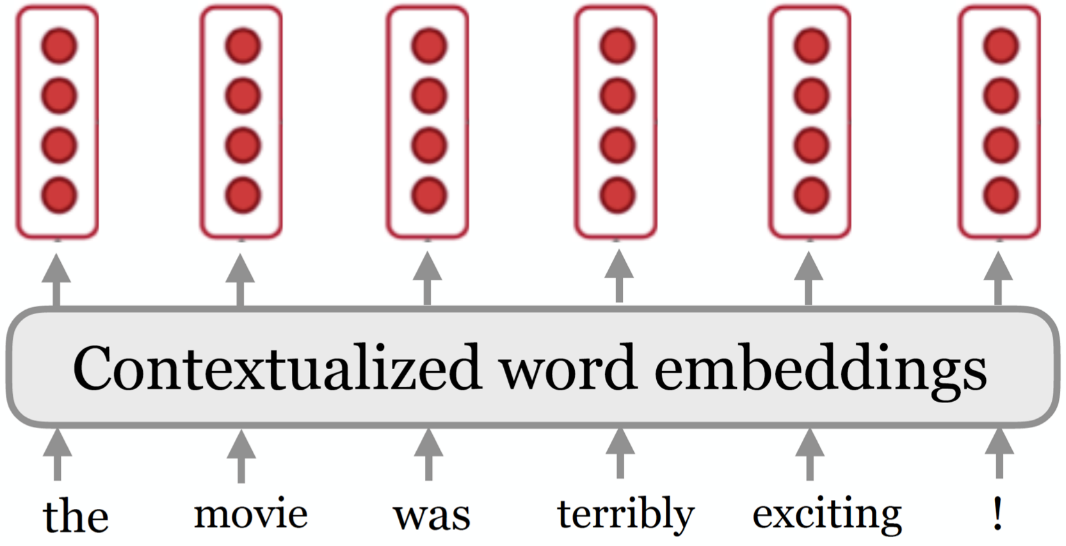


Wrong !

- Why not learn the representations for each word in its context?

Types of Embeddings -2

- Contextualized word embedding
  - Build a vector representation for each word conditioned on its **CONTEXT**!
  - i.e. representation for each word is a function of the entire input sentence



$$g : (w_1, w_2, \ldots, w_n) \longrightarrow \mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$$