Intro to NLP
Word Embeddings

Gennaro S. Rodrigues, Ph.D.

Overview

1. Word Representation
2. Word Embedding
3. Continuous Bag-Of-Words (CBOW) Model
4. Training CBOW
5. Extracting Embeddings
6. Evaluating Emgeddings

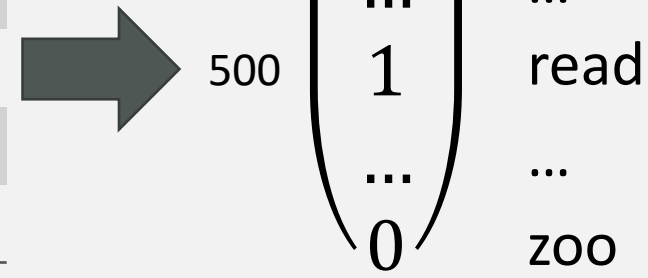
Word Representation

- Remember
- One-hot vectors
  - [0 , 1, 0, 0, …, 0]
  - Size = V
  - Each position represents a word
  - Problematic: memory

Word Representation

- Remember
- One-hot vectors
  - [0 , 1, 0, 0, …, 0]
  - Size = V
  - Each position represents a word
  - Problematic: memory

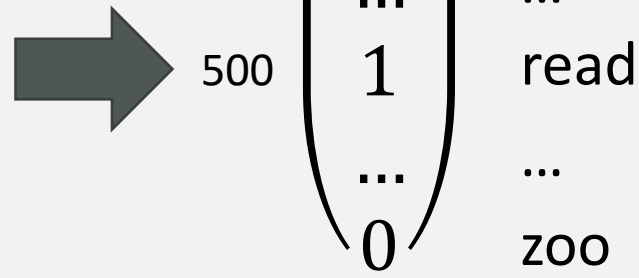
Word	Number
a	0
an	0
….	…
read	500
…	…
zoo	0



Word Representation

- Remember
- One-hot vectors
  - [0 , 1, 0, 0, …, 0]
  - Size = V
  - Each position represents a word
  - Problematic: memory
- Simple
- No implications
- Don’t hold any intrinsic information

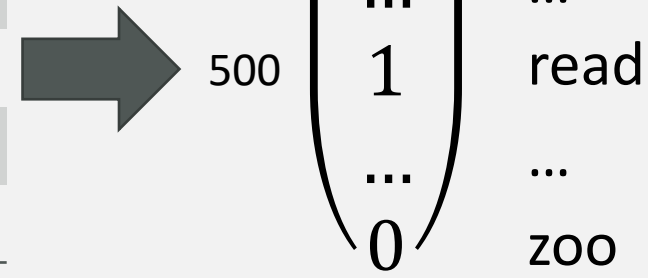
Word	Number
a	0
an	0
….	…
read	500
…	…
zoo	0



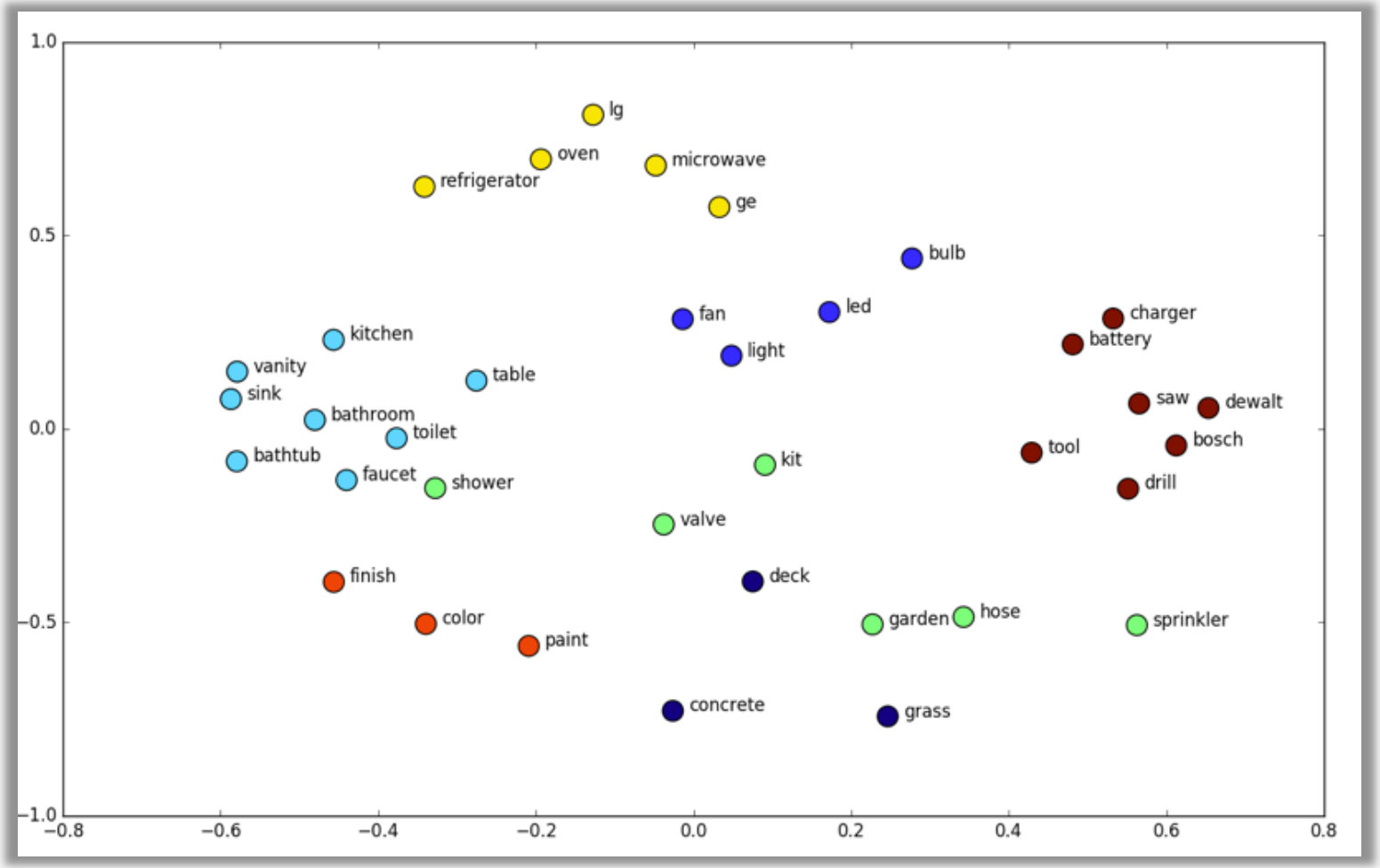
Word Representation

- Remember
- One-hot vectors
  - [0 , 1, 0, 0, …, 0]
  - Size = V
  - Each position represents a word
  - Problematic: memory
- Simple
- No implications
- Don’t hold any intrinsic information

Word	Number
a	0
an	0
….	…
read	500
…	…
zoo	0

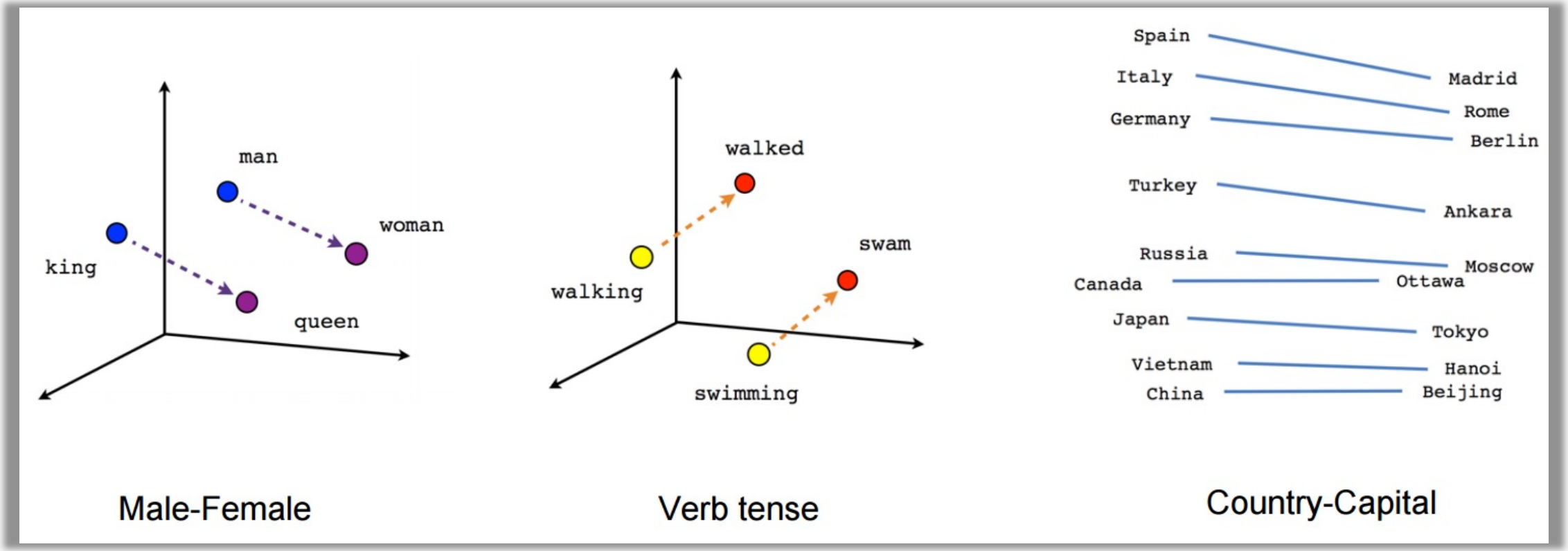


Word Representation - Vectors



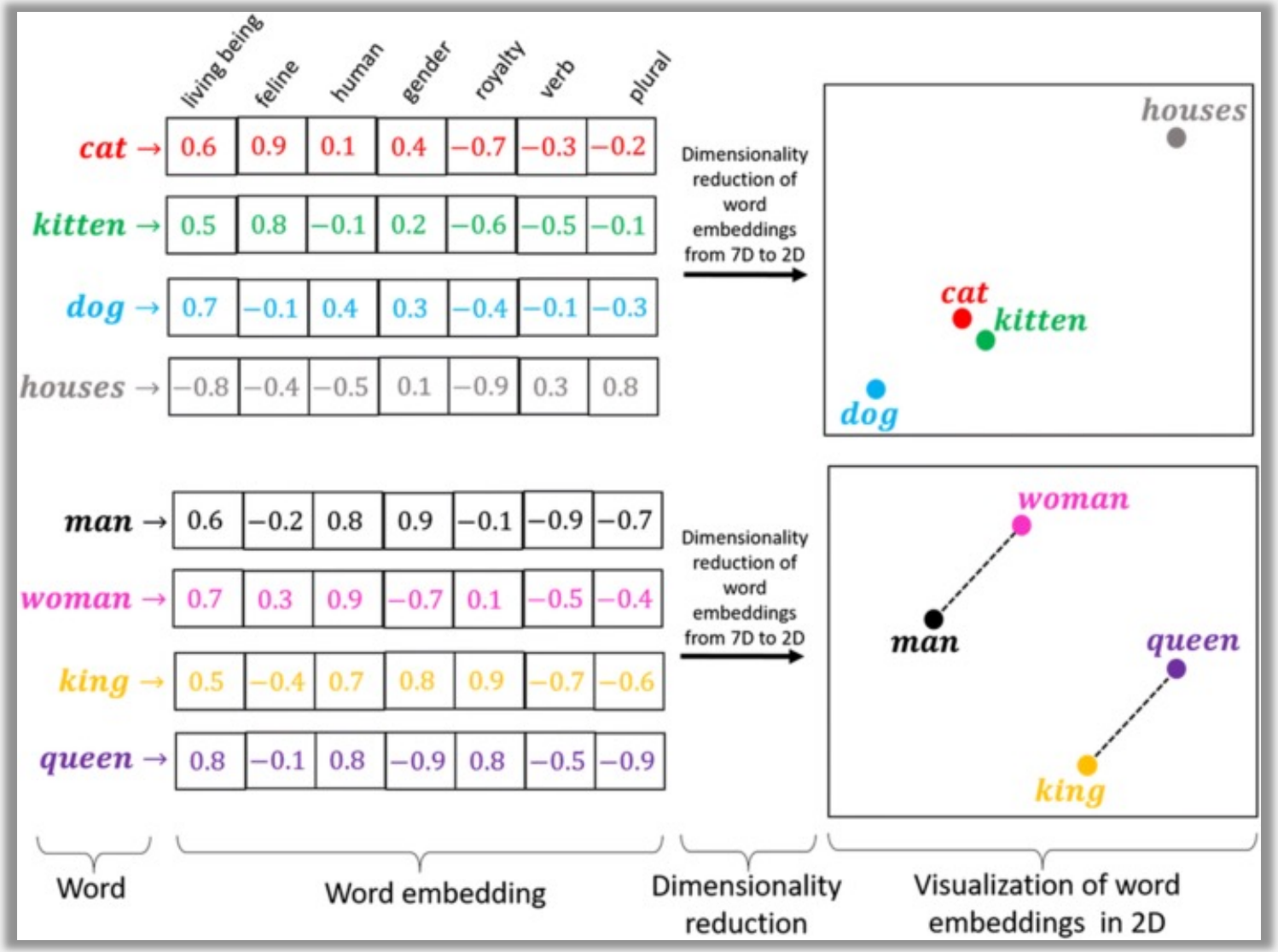
(Lynn, 2018)

Word Representation - Vectors



(Lynn, 2018)

Word Representation - Vectors



Word Representation - Vectors

- Low Dimension (is it?)
- Embedded meaning
- Semantic Distance (remember exercises?)
- Analogies
- Word Vectors == Word Embeddings

Word Embeddings - Creation

- Corpus
- Embedding Method

Word Embeddings - Creation

- Corpus
  - Words within a context
  - Example: learning to write a given author poetry: has to be a corpus from that author, not Wikipedia information or side notes.
  - Remember: the meaning of a word is related to its combination with other words.
    - Context is VERY important!
- The corpus shall take into account the context.
  - NLP for legal purposes shall learn from legal books.

Word Embeddings - Creation

- Embedding Method
  - We will focus on Machine Learning methods.
  - Learns a tasks
  - The product of the learning is the word embeddings themselves.
  - Example:

    I am [???] NLP.
  - The task is often to learn to predict the word that is missing, based on the words around it (CBOW).
    - Self-supervised: data is unlabeled, the data itself provides the context that would make up the labels.
- Hyperparameters

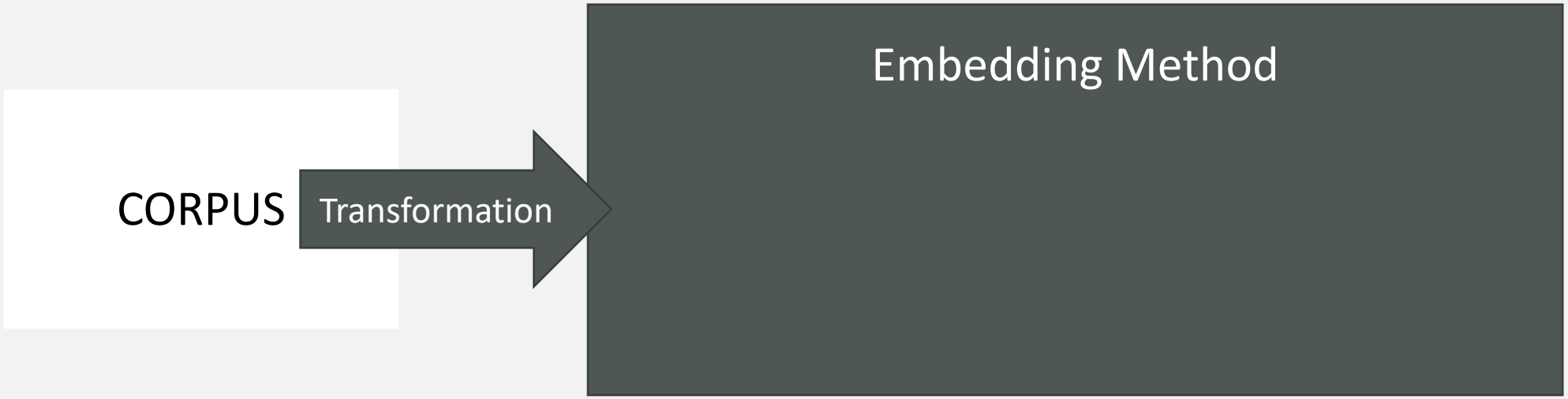
Word Embeddings - Creation

- Popular models and methods
- word2vec, by google (2013)
  - Continuous bag-of-words (CBOW)
    - Predicts a missing word given the words around it
  - Continuous skip-gram
    - Given a word, predicts the words around it
- Global Vectors (GloVe), by Stanford (2014)
  - Factorizes the logarithm of the co-occurrence matrix
- fastText, by Facebook (2016)
  - Skip-gram, words are n-grams of characters (thus supporting OOV)

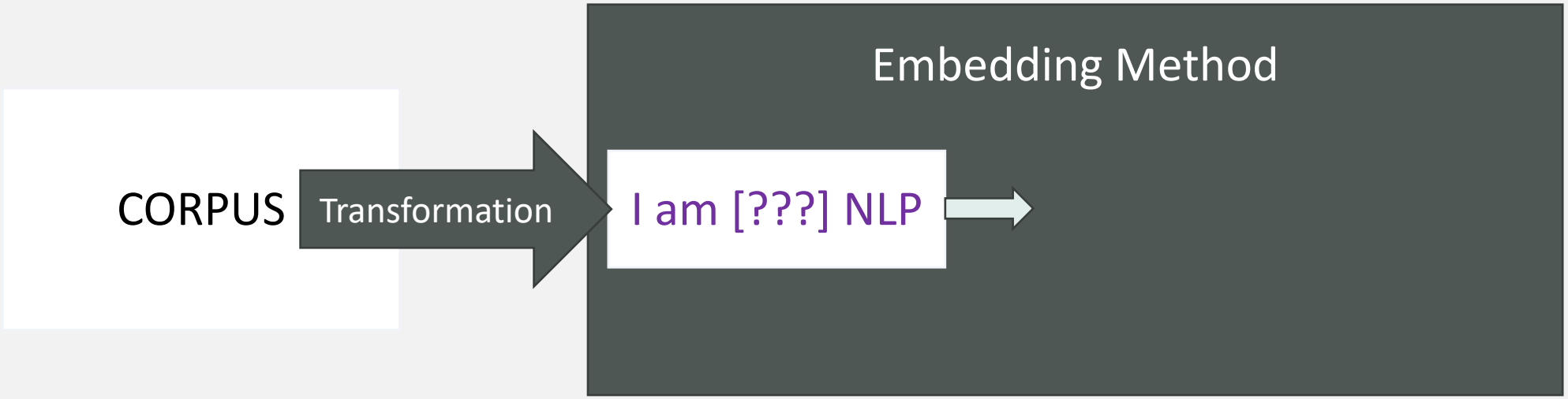
Word Embeddings - Creation

- Advanced models and methods: deep neural architectures, a word can have different vectors depending on the context (polysemy).
- BERT, by Google (2018)
- ELMo, from Allen Institute for AI (2018)
- GPT-2 and GPT-3, from OpenAI (2018 and 2020)
- Those can be downloaded pre-trained and then be tunned accordingly with your own corpus (except for GPT-3).

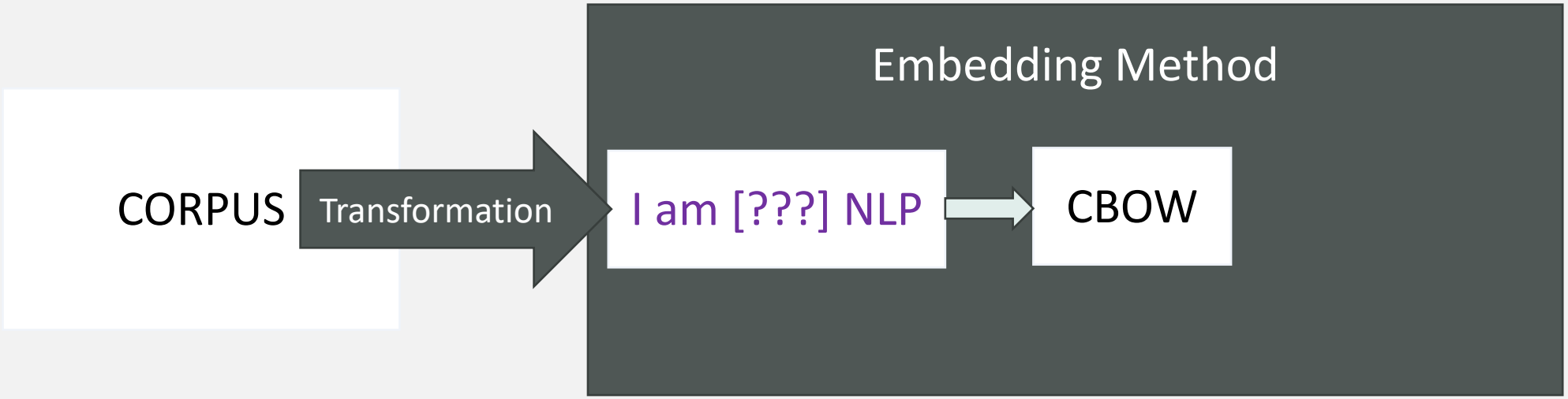
Continuous Bag-Of-Words (CBOW)



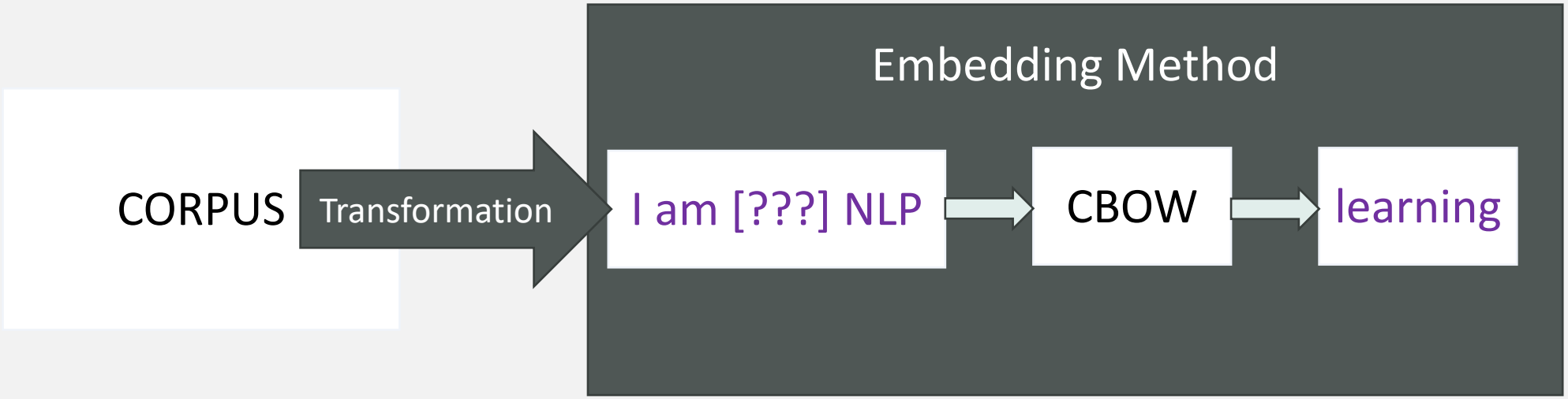
Continuous Bag-Of-Words (CBOW)



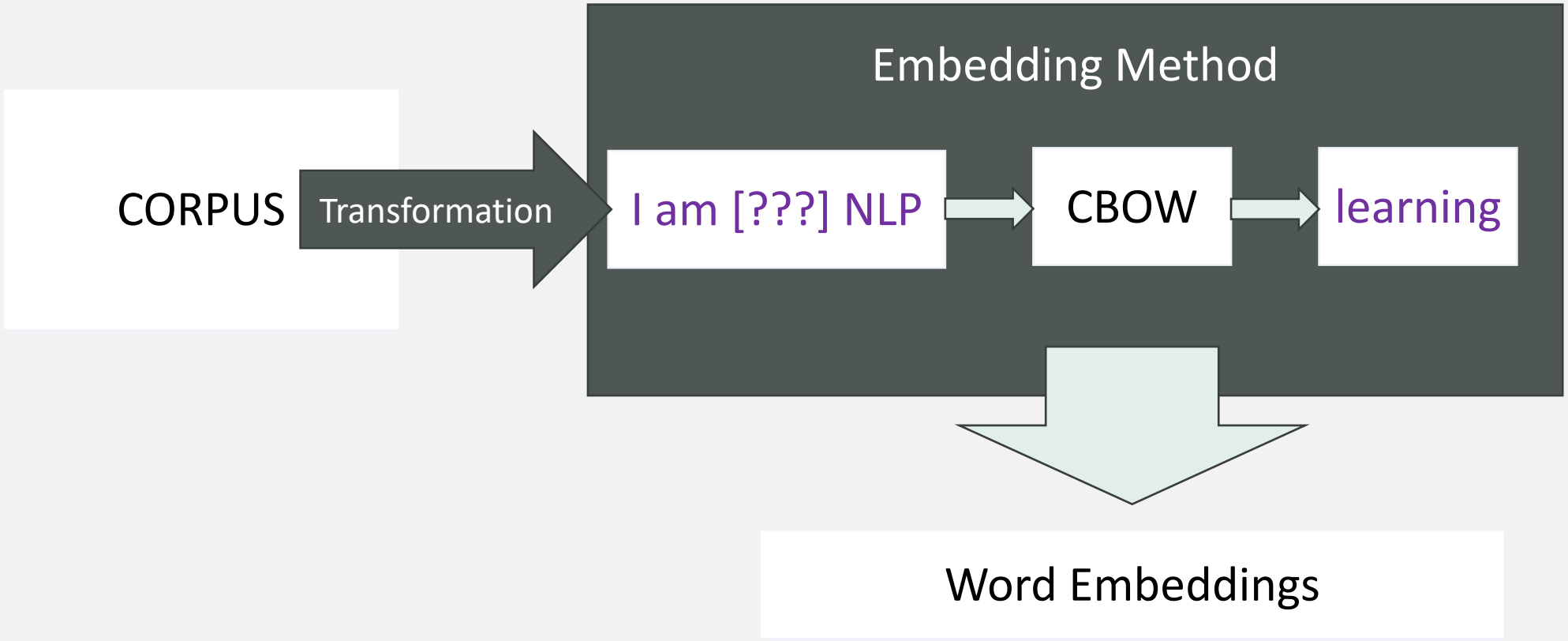
Continuous Bag-Of-Words (CBOW)



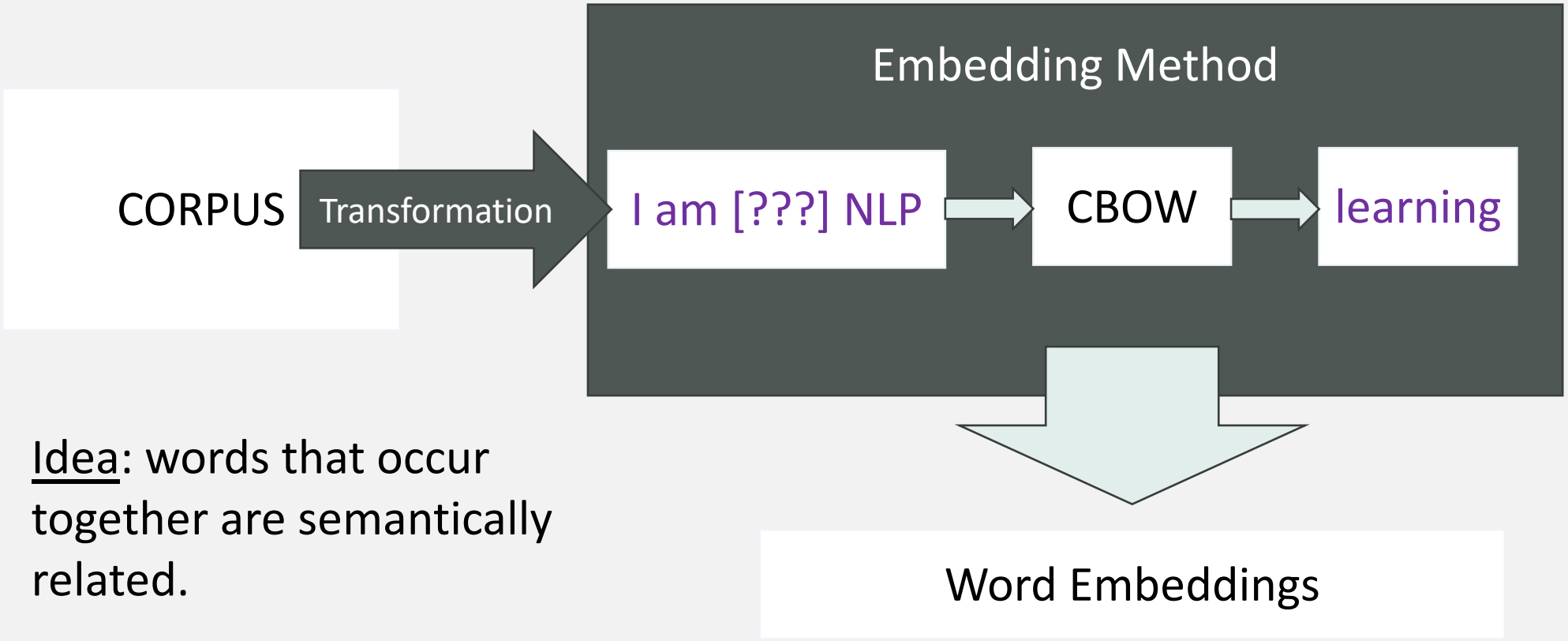
Continuous Bag-Of-Words (CBOW)



Continuous Bag-Of-Words (CBOW)



Continuous Bag-Of-Words (CBOW)



Idea: words that occur together are semantically related.

CBOW - Example

The [???] is studying NLP.

CBOW - Example

The [???] is studying NLP.

student

scientist

…

engineer

CBOW - Example

The [???] is studying NLP.

student

scientist

…

engineer

Those have related meaning.

CBOW – Creating Training Data

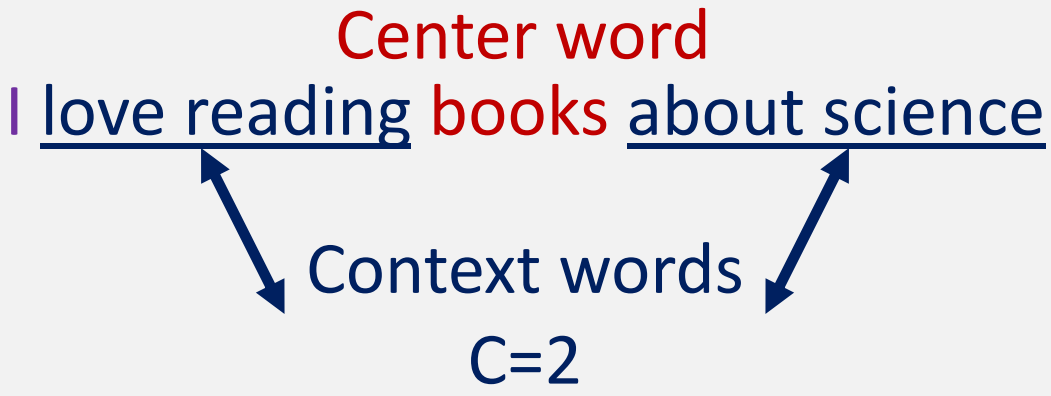
I love reading books about science

CBOW – Creating Training Data

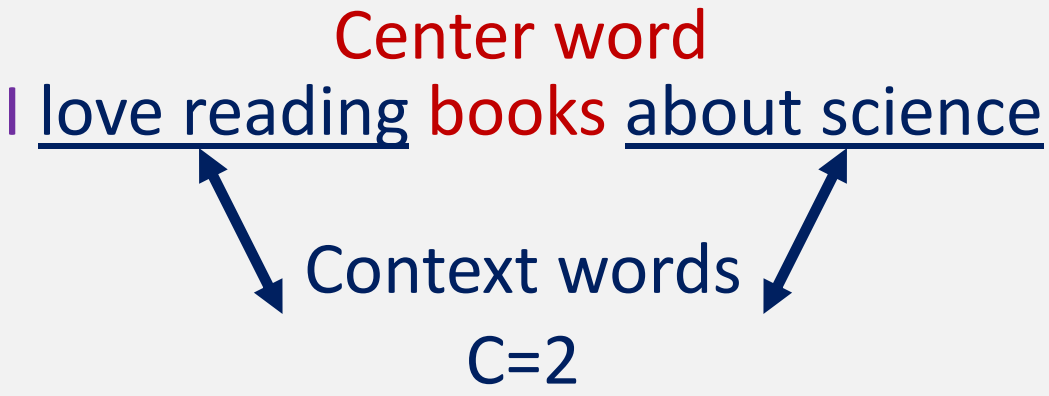
Center word

I love reading books about science

CBOW – Creating Training Data

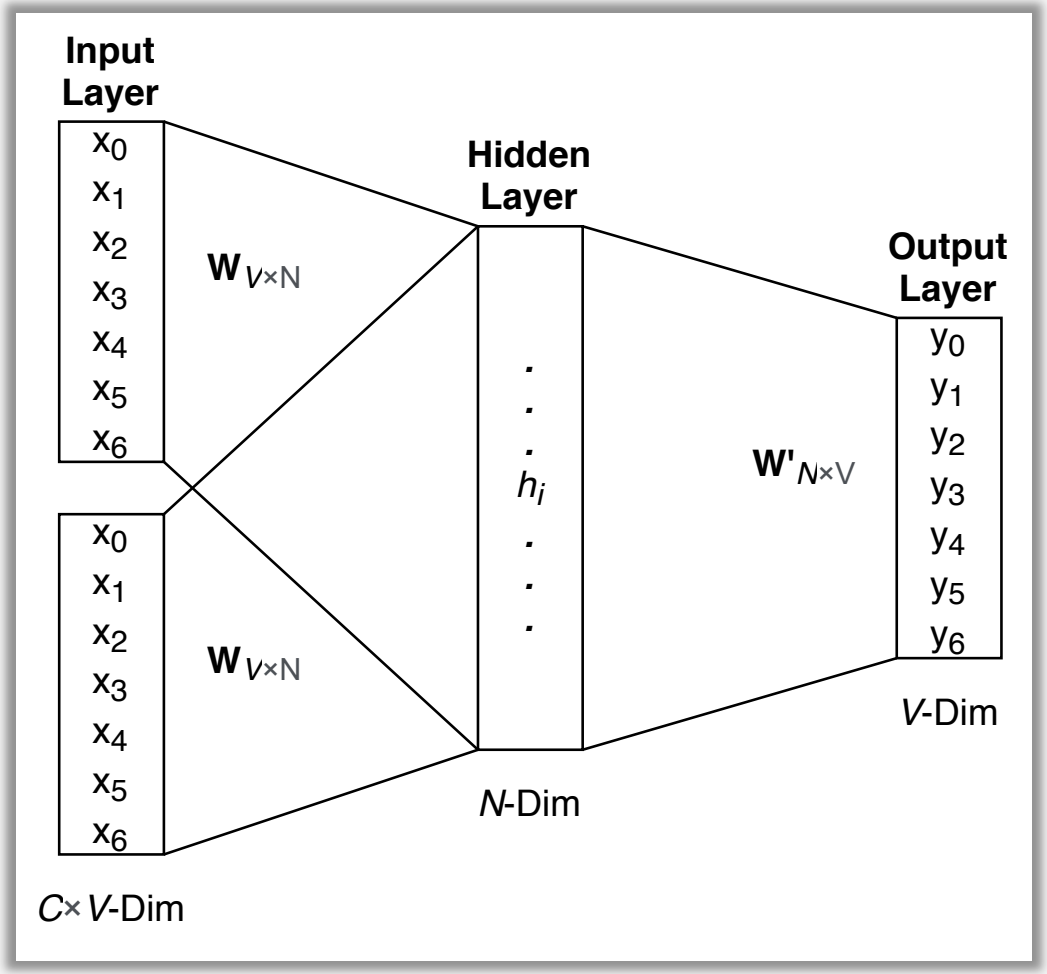


CBOW – Creating Training Data



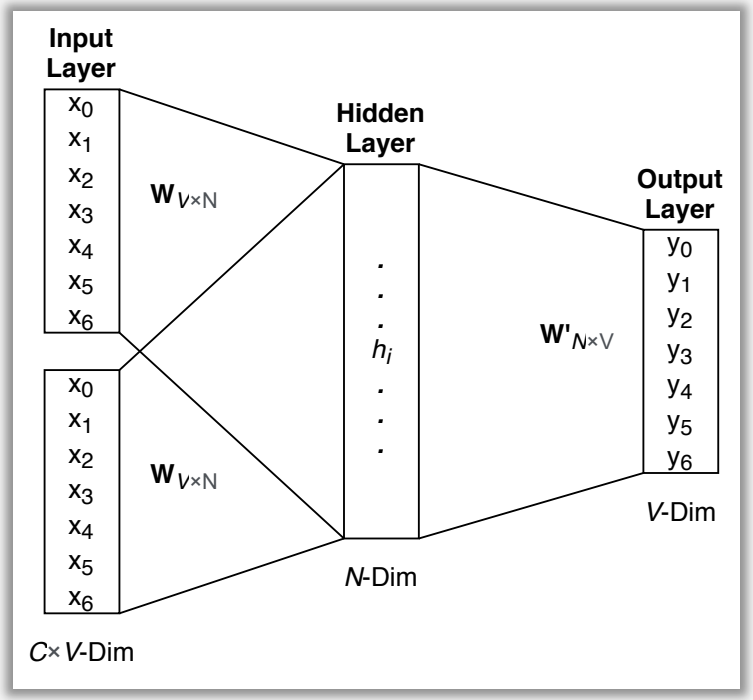
- C: half-size of the context, an hyperparameter

CBOW – Architecture



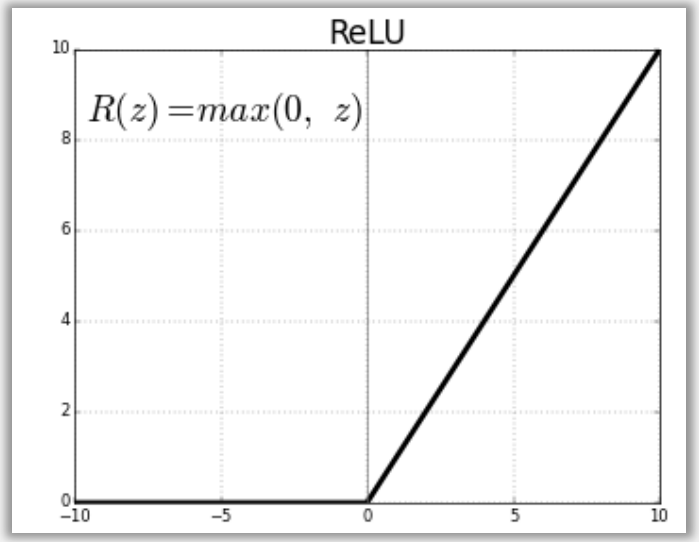
CBOW – Training

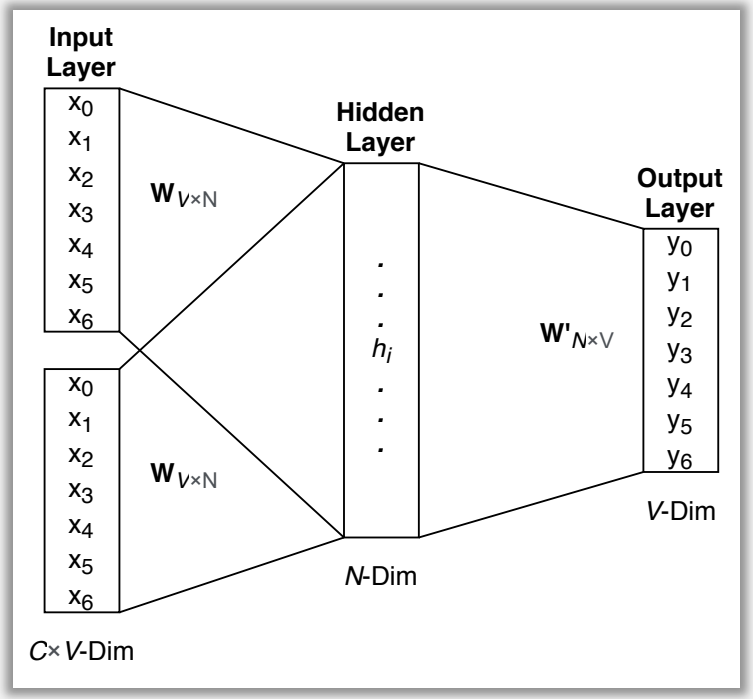
- The size of the input vectors is equal to the size of the vocabulary.
- The input and output size is equal to V.
- The size of the word embeddings is N and is an hyperparameter of the model.
- First activation: ReLU
- Last activation: softmax



CBOW – Training

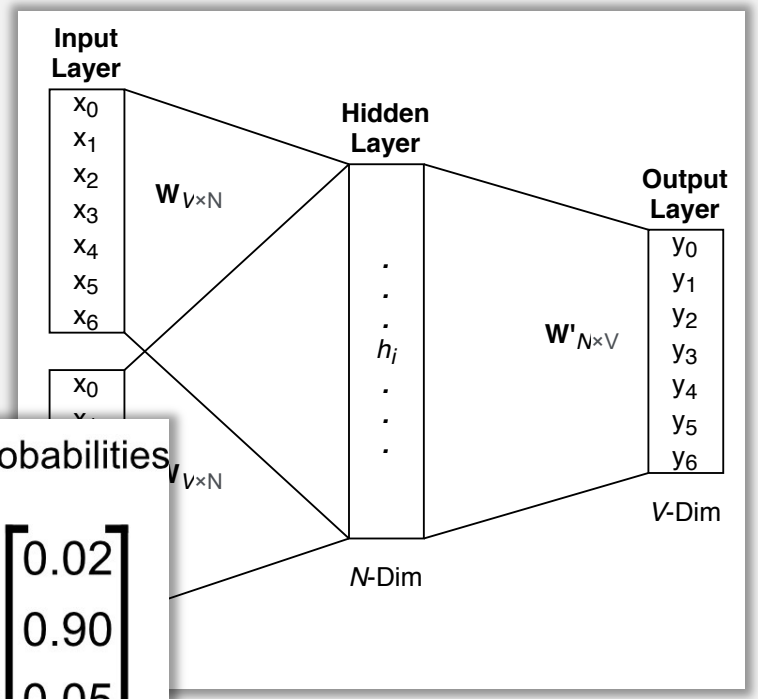
- The size of the input vectors is equal to the size of the vocabulary.
- The input and output size is equal to V.
- The size of the word embeddings is N and is an hyperparameter of the model.
- First activation: ReLU
- Last activation: softmax

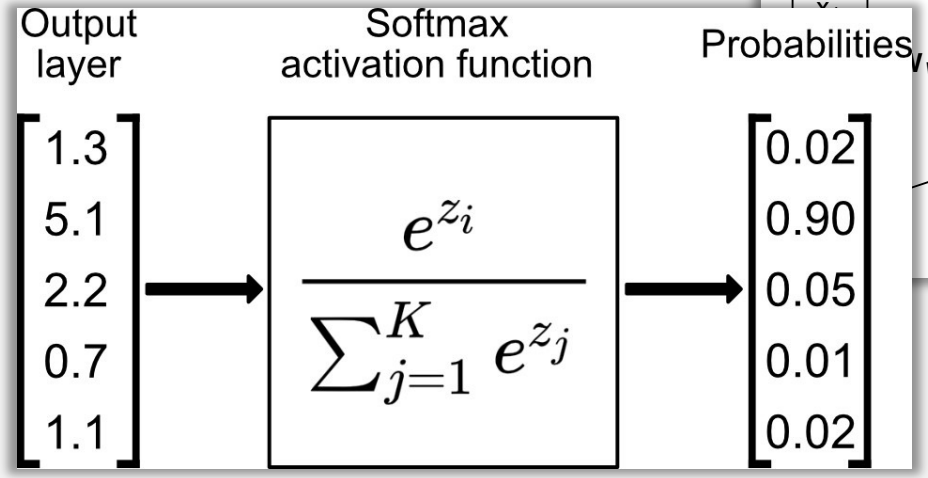




CBOW – Training

- The size of the input vectors is equal to the size of the vocabulary.
- The input and output size is equal to V.
- The size of the word embeddings is N and is an hyperparameter of the model.
- First activation: ReLU
- Last activation: softmax



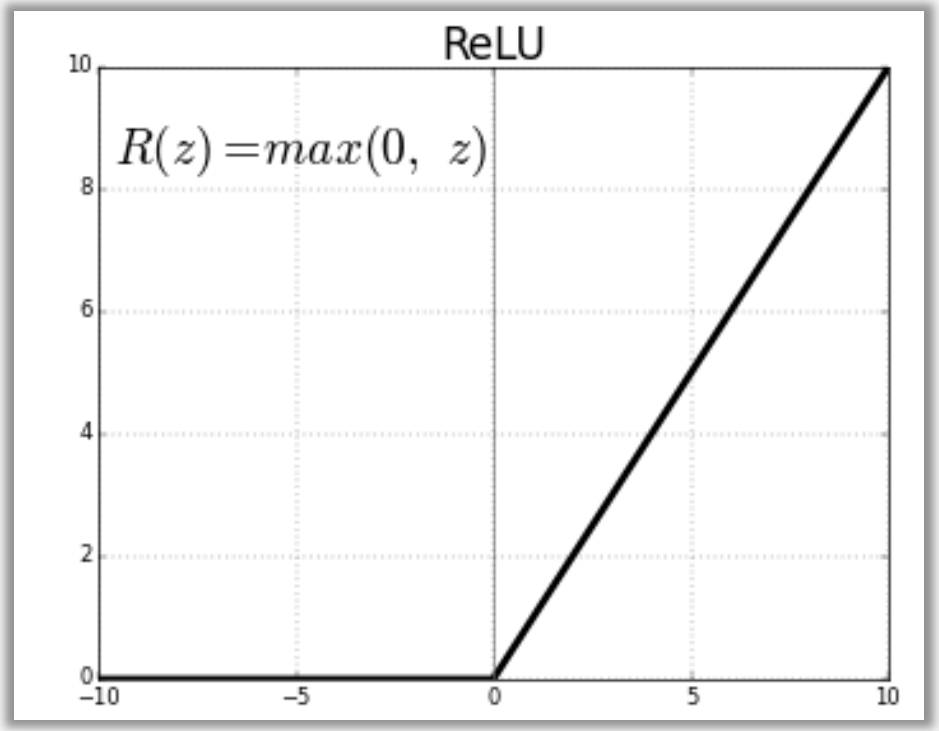


Rectified Linear Unit (ReLU)

- $z = Wx + b$
- $h = ReLU(z)$

$$\begin{pmatrix} 3.3 \\ -0.87 \\ \dots \\ 0.47 \\ \dots \\ -0.2 \end{pmatrix}$$

z

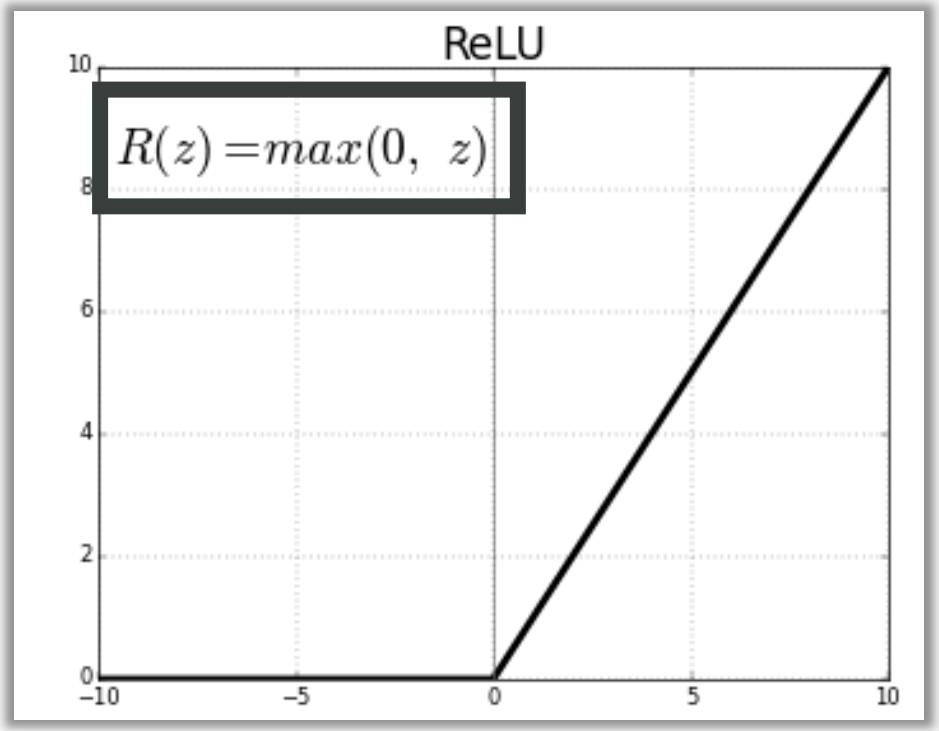


Rectified Linear Unit (ReLU)

- $z = Wx + b$
- $h = ReLU(z)$

$$\begin{pmatrix} 3.3 \\ -0.87 \\ \dots \\ 0.47 \\ \dots \\ -0.2 \end{pmatrix} \xrightarrow{\text{ReLU}} \begin{pmatrix} 3.3 \\ 0 \\ \dots \\ 0.47 \\ \dots \\ 0 \end{pmatrix}$$

z h

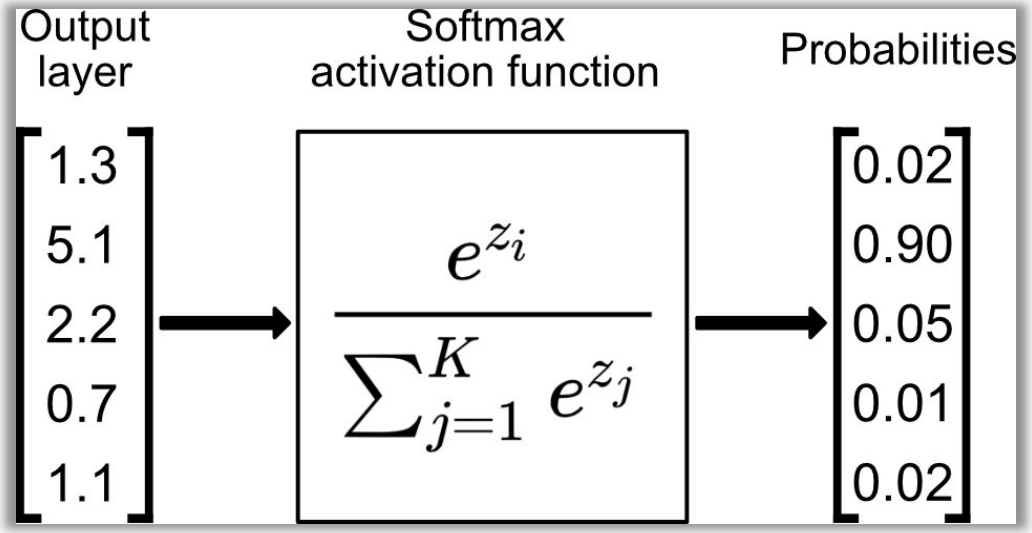


Softmax

- $z = Wx + b$
- $\hat{y} = softmax(z)$
- Returns a probability

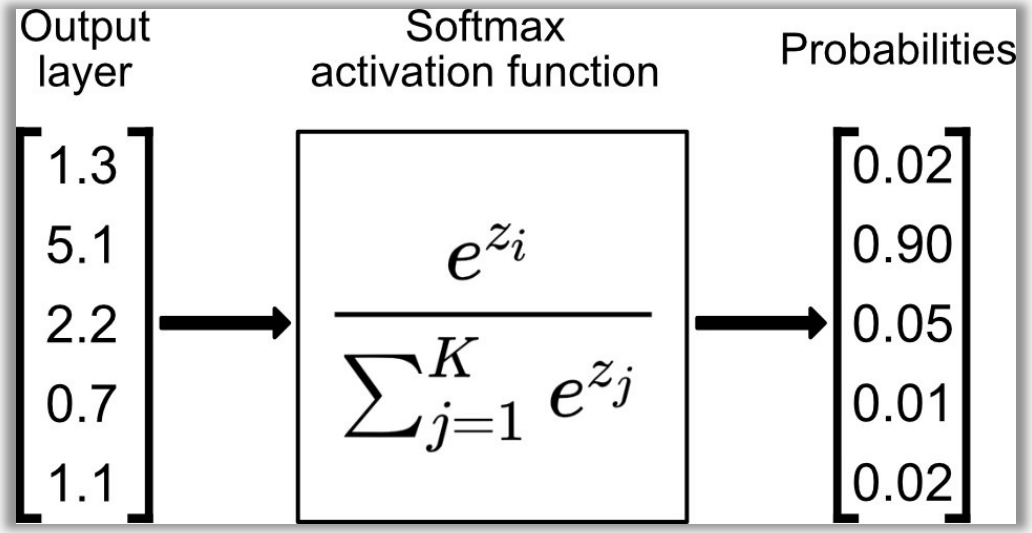
$$\begin{pmatrix} 3.3 \\ 0 \\ \dots \\ 0.47 \\ \dots \\ 0 \end{pmatrix} \xrightarrow{\text{softmax}} \begin{pmatrix} 0.02 \\ 0.67 \\ \dots \\ 0.007 \\ \dots \\ 0.126 \end{pmatrix}$$

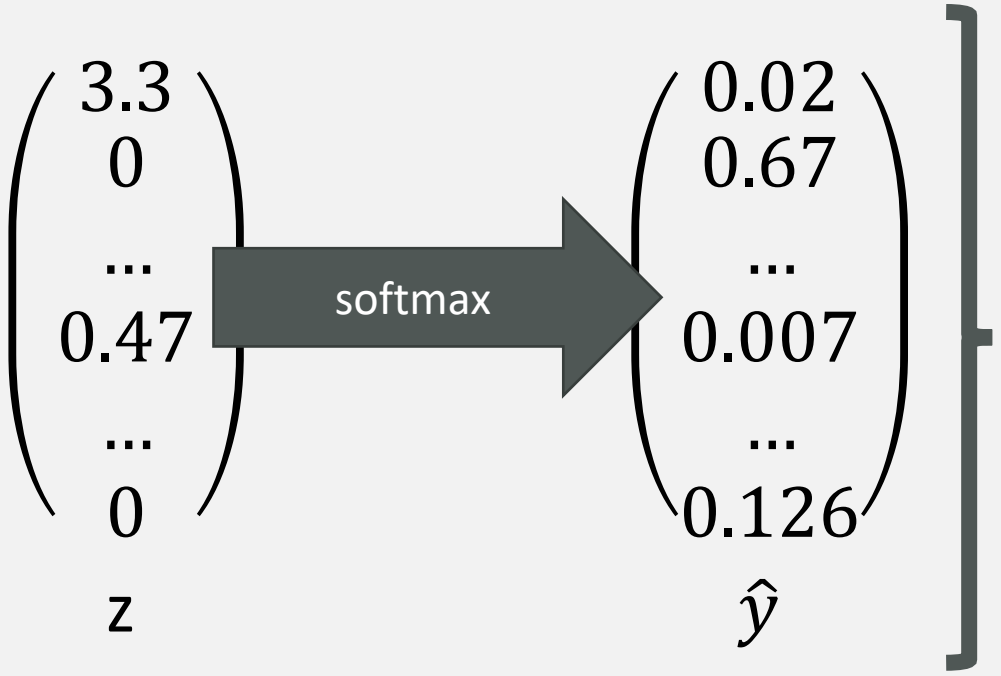
z $\qquad$ $\hat{y}$



Softmax

- $z = Wx + b$
- $\hat{y} = softmax(z)$
- Returns a probability

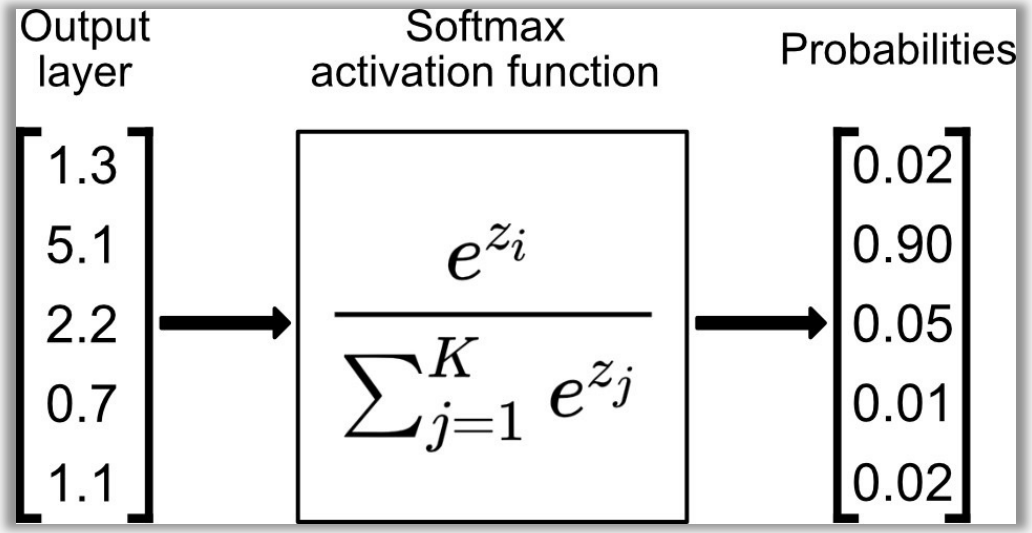


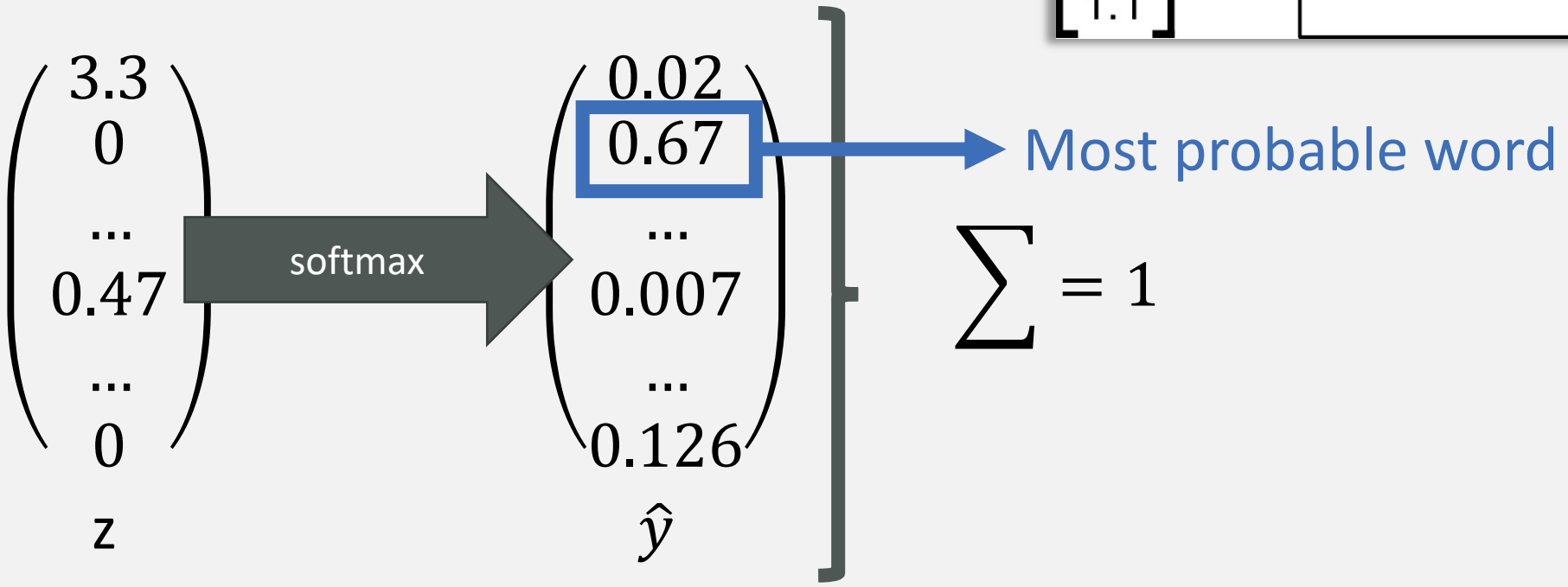


$\sum = 1$

Softmax

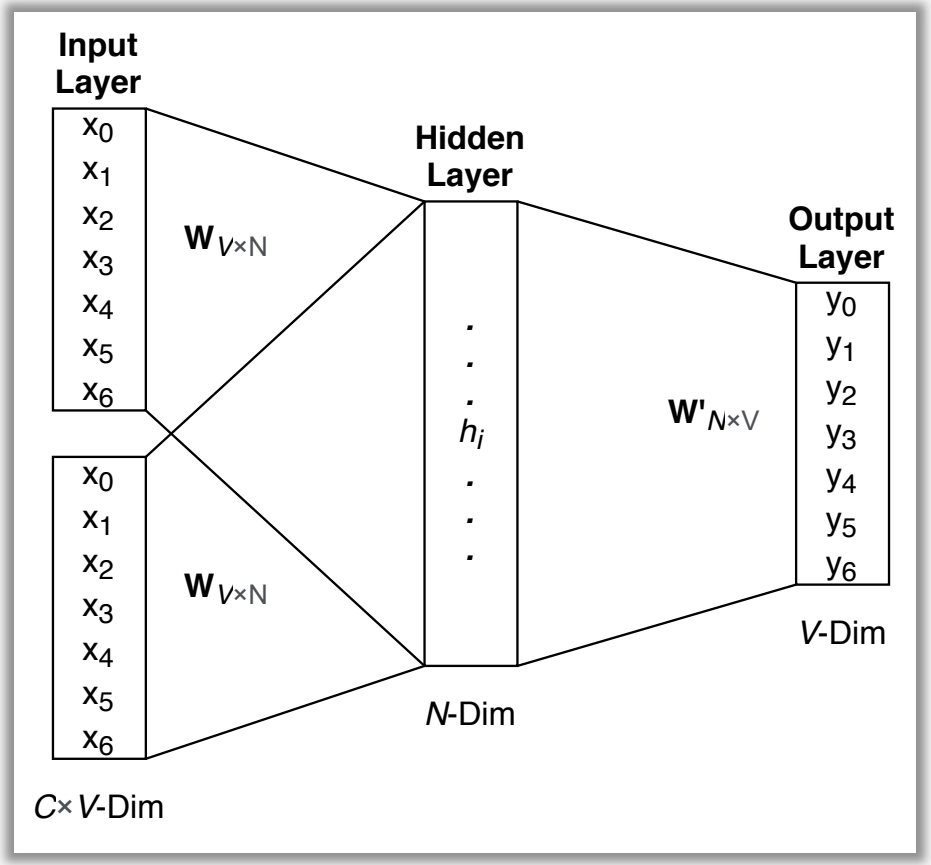
- $z = Wx + b$
- $\hat{y} = softmax(z)$
- Returns a probability





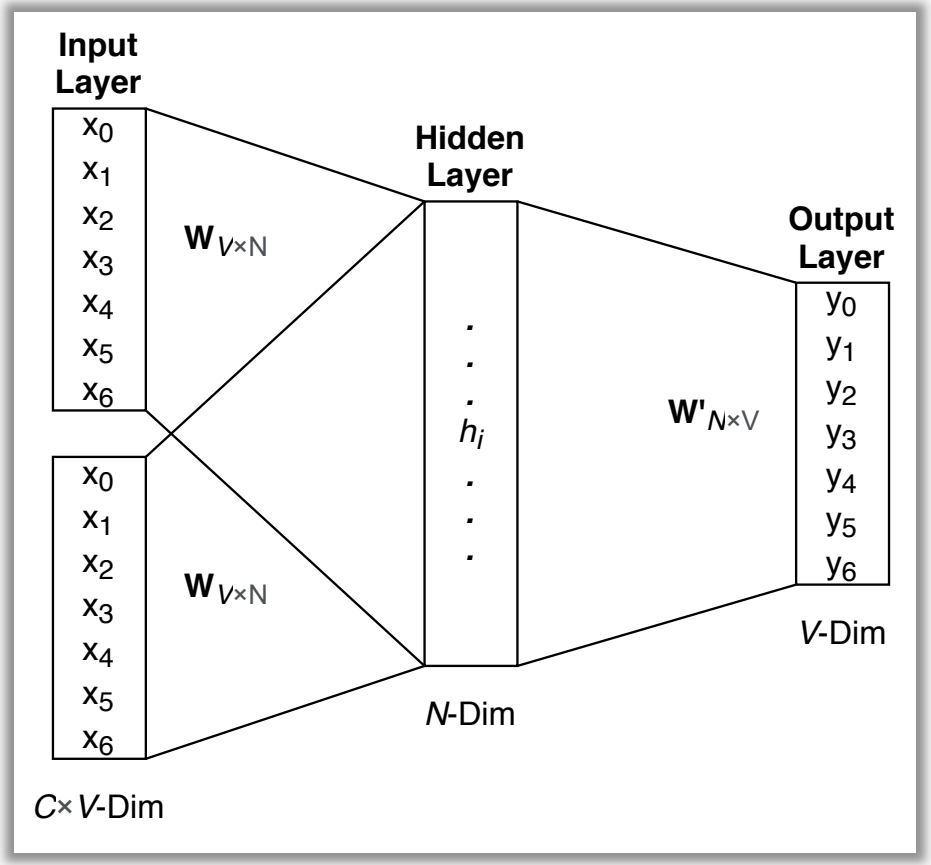
CBOW - Word Embeddings

- How do we extract word embeddings from that model?
- The output of that model is a prediction for the central word, not a word embedding…



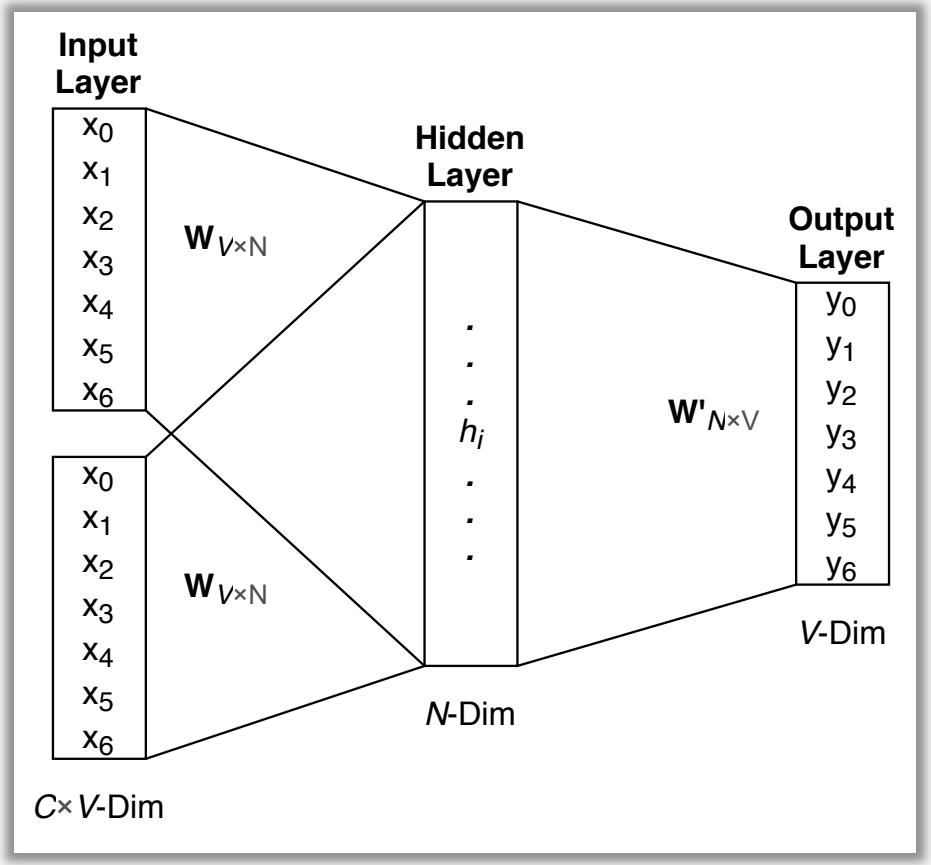
CBOW - Word Embeddings

- How do we extract word embeddings from that model?
- The output of that model is a prediction for the central word, not a word embedding…
- Remember:
  - The dimensions of the weight matrices.
  - The N value is an hyperparameter.



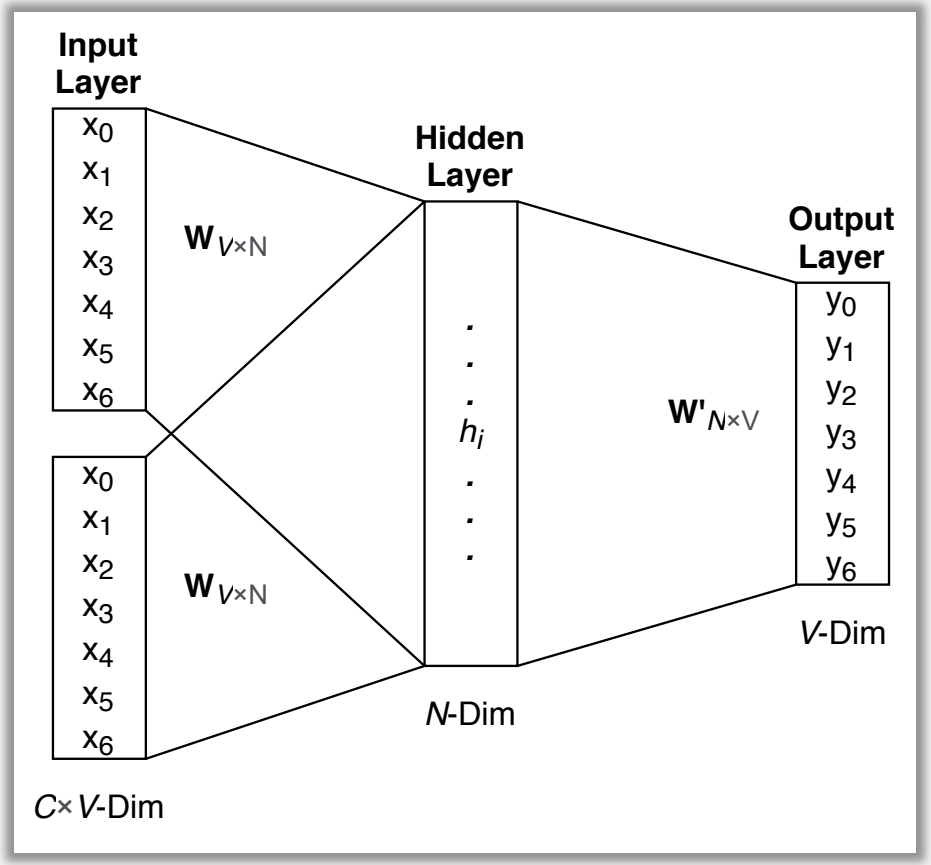
CBOW - Word Embeddings

- How do we extract word embeddings from that model?
- The output of that model is a prediction for the central word, not a word embedding…
- Remember:
  - The dimensions of the weight matrices.
  - The N value is an hyperparameter.
- The matrix W has one line per word in V.
- The matrix W’ has one column per word.



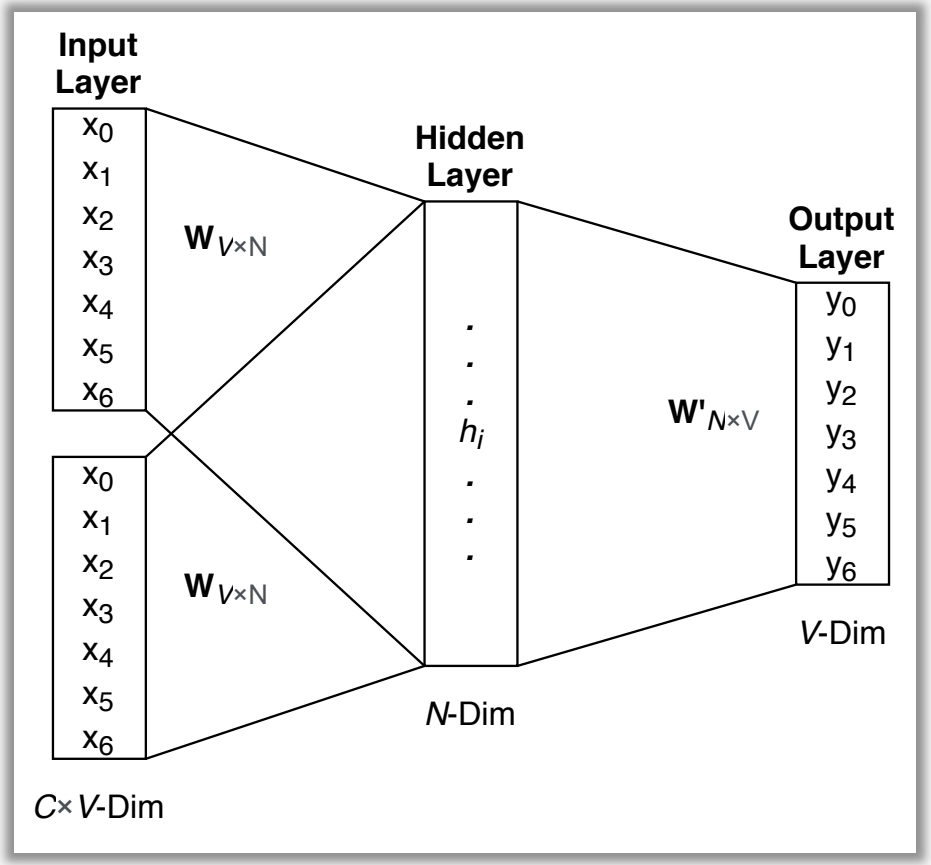
CBOW - Word Embeddings

- The matrix W has one line per word in V.
- The matrix W’ has one column per word.
- Each of those rows/columns represent a word from the vocabulary.
- Remember: each word from the vocabulary is represented by an index in the input vectors, and the index shall always be the same for a given word.



CBOW - Word Embeddings

- Option 1: take the word embeddings from the matrix **W**. Each word in its given indexed row.
- Option 2: take the word embeddings from the matrix **W'**. Each word in its given indexed column.
- Option 3: take the word embeddings from the average of the word vectors represented at each one of the matrices.



Word Embeddings – Evaluation

- Intrinsic Evaluation: tests the relations between words. How the embeddings are representing the semantic and syntactic meaning of the words and relations between words.
- Semantics: meaning.
- Syntax: grammar.
- Testing:
  - Semantic Analogies: “France” is to ”Paris”, ”Brazil” is to ???
  - Syntactic Analogies: plurals, possessives, etc. “ate” is to “eat”, “saw” is to ???
  - Attention: be aware of ambiguity. There can be multiple possible answers.
  - Clustering: graphically.

Word Embeddings – Evaluation

- Extrinsic Evaluation: tests the embeddings performing an external task, like part-of-speech tagging (POS) or named entity recognition (NER).
- This evaluates how useful the generated embeddings really are.
- Much more time-consuming.
- Relies on external methodologies.
- Harder to troubleshoot (where is the problem?).

Thank you!