Word Representation

Word Representation - One Hot Vector

Word Representation - One Hot Vector

- Represent every word as an $\mathbb{R}^{|V| \times 1}$ vector with all 0s and one 1 at the index of that word.($|V|$ is the size of our vocabulary)

Word Representation - One Hot Vector

- Represent every word as an $\mathbb{R}^{|V|\times 1}$ vector with all 0s and one 1 at the index of that word.($|V|$ is the size of our vocabulary)

Example:
High, Low , Hotel, Resort

$$w^{High} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, w^{Low} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, W^{Hotel} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, w^{Resort} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Word Representation - One Hot Vector

Problem with One Hot Vector

Word Representation - One Hot Vector

Problem with One Hot Vector

- Vector size very big.(Consider vocab size of 13M words)

Word Representation - One Hot Vector

Problem with One Hot Vector

- Vector size very big.(Consider vocab size of 13M words)
- Sparse Vector

Word Representation - One Hot Vector

Problem with One Hot Vector

- Vector size very big.(Consider vocab size of 13M words)
- Sparse Vector
- Cannot capture semantic similarity.

Word Representation - One Hot Vector

Problem with One Hot Vector

- Vector size very big.(Consider vocab size of 13M words)
- Sparse Vector
- Cannot capture semantic similarity.
  Example:

Word Representation - One Hot Vector

Problem with One Hot Vector

- Vector size very big.(Consider vocab size of 13M words)
- Sparse Vector
- Cannot capture semantic similarity.
  Example:
  Hotel and Resort are synonyms but vector are not similar

Word Representation - One Hot Vector

Problem with One Hot Vector

- Vector size very big.(Consider vocab size of 13M words)
- Sparse Vector
- Cannot capture semantic similarity.
  Example:
  Hotel and Resort are synonyms but vector are not similar
  Hotel = [0 0 1 0]
  Resort = [0 0 0 1]

Learn encode similarity in the vectors themselves

Word Vector - Word2Vec

- How we learn to encode similarity in the vectors themselves?

Word Vector - Word2Vec

- How we learn to encode similarity in the vectors themselves?
  - Represent words by their context.

Word Vector - Word2Vec

- How we learn to encode similarity in the vectors themselves?
  - Represent words by their context.
- **Distributional semantics:**

Word Vector - Word2Vec

- How we learn to encode similarity in the vectors themselves?
  - Represent words by their context.
- **Distributional semantics:**

  A word's meaning is given by the words that frequently appear close-by

Word Vector - Word2Vec

- How we learn to encode similarity in the vectors themselves?
  - Represent words by their context.
- **Distributional semantics:**

  A word's meaning is given by the words that frequently appear close-by
  - "You shall know a word by the company it keeps"
    (J. R. Firth 1957: 11)

Word Vector - Word2Vec

- How we learn to encode similarity in the vectors themselves?
  - Represent words by their context.
- **Distributional semantics:**

  A word's meaning is given by the words that frequently appear close-by
  - "You shall know a word by the company it keeps"
    (J. R. Firth 1957: 11)
- Context of word w in text is a set of words (within a fixed-size window) that appear nearby.

Word Vector - Word2Vec

- How we learn to encode similarity in the vectors themselves?
  - Represent words by their context.
- **Distributional semantics:**

  A word's meaning is given by the words that frequently appear close-by
  - "You shall know a word by the company it keeps"
    (J. R. Firth 1957: 11)
- Context of word w in text is a set of words (within a fixed-size window) that appear nearby.
- Contexts are used to construct a representation of w

Word Representation - Word2Vec - Context

. . . government debt problems turning into **banking** crises as happened in 2009. . .

. . . saying that Europe needs unified **banking** regulation to replace the hodgepodge. . .

. . . India has just given its **banking** system a shot in the arm. . .

These context words will represent **banking**

Word Representation - Word2Vec - Context

...government debt problems turning into **banking** crises as happened in 2009...

...saying that Europe needs unified **banking** regulation to replace the hodgepodge...

...India has just given its **banking** system a shot in the arm...

These context words will represent **banking**

- Construct a dense vector for each word to be similar to word vectors that appear in similar contexts.

Word Representation - Word2Vec - Context

...government debt problems turning into **banking** crises as happened in 2009...

...saying that Europe needs unified **banking** regulation to replace the hodgepodge...

...India has just given its **banking** system a shot in the arm...

These context words will represent **banking**

- Construct a dense vector for each word to be similar to word vectors that appear in similar contexts.

Word Representation - Word2Vec - Window Size

Window size $= 2$

Word Representation - Word2Vec - Window Size

Window size $= 2$

Center

India has just given its **banking** system a shot in the arm

Word Representation - Word2Vec - Window Size

Window size $= 2$

Center

India has just $\boxed{\text{given its}}$ **banking** $\boxed{\text{system a}}$ shot in the arm

Context

Word Representation - Word2Vec - Window Size

Window size $= 2$

Center

India has just $\boxed{\text{given its}}$ **banking** $\boxed{\text{system a}}$ shot in the arm

Context

India has just given $\boxed{\text{its banking}}$ **system** $\boxed{\text{a shot}}$ in the arm

Word Representation - Word2Vec - Window Size

Window size = 2

Center

India has just given its **banking** system a shot in the arm

Context

India has just given its banking **system** a shot in the arm

India has just given its banking system **a** shot in the arm

Word Representation - Word2Vec - Window Size

Window size = 2

Center

India has just $\boxed{\text{given its}}$ **banking** $\boxed{\text{system a}}$ shot in the arm

Context

India has just given $\boxed{\text{its banking}}$ **system** $\boxed{\text{a shot}}$ in the arm

India has just given its $\boxed{\text{banking system}}$ **a** $\boxed{\text{shot in}}$ the arm

Word Representation - Word2Vec

Word2vec (Mikolov et al. 2013) [1] is a framework for learning word vectors.

[1] Tomas Mikolov et al.,Distributed Representations of Words and Phrases and Their Compositionality, NIPS, Volume 2, 2013, 3111–3119

Word Representation - Word2Vec

Word2vec (Mikolov et al. 2013) [1] is a framework for learning word vectors.

Idea:

[1] Tomas Mikolov et al.,Distributed Representations of Words and Phrases and Their Compositionality, NIPS, Volume 2, 2013, 3111–3119

Word Representation - Word2Vec

Word2vec (Mikolov et al. 2013) [1] is a framework for learning word vectors.

Idea:

- We have a large corpus of text

[1] Tomas Mikolov et al.,Distributed Representations of Words and Phrases and Their Compositionality, NIPS, Volume 2, 2013, 3111–3119

Word Representation - Word2Vec

Word2vec (Mikolov et al. 2013) [1] is a framework for learning word vectors.

Idea:

- We have a large corpus of text
- Every word in a fixed vocabulary is represented by a vector.

[1] Tomas Mikolov et al.,Distributed Representations of Words and Phrases and Their Compositionality, NIPS, Volume 2, 2013, 3111–3119

Word Representation - Word2Vec

Word2vec (Mikolov et al. 2013) [1] is a framework for learning word vectors.

Idea:

- We have a large corpus of text
- Every word in a fixed vocabulary is represented by a vector.
- Go through each position t in the text, which has a center word c and context ("outside") words o.

[1] Tomas Mikolov et al.,Distributed Representations of Words and Phrases and Their Compositionality, NIPS, Volume 2, 2013, 3111–3119

Word Representation - Word2Vec

Word2vec (Mikolov et al. 2013) [1] is a framework for learning word vectors.

Idea:

- We have a large corpus of text
- Every word in a fixed vocabulary is represented by a vector.
- Go through each position t in the text, which has a center word c and context ("outside") words o.
- Use the similarity of the word vectors for c and o to calculate the probability of o given c (or vice versa).

[1] Tomas Mikolov et al.,Distributed Representations of Words and Phrases and Their Compositionality, NIPS, Volume 2, 2013, 3111–3119

Word Representation - Word2Vec

Word2vec (Mikolov et al. 2013) [1] is a framework for learning word vectors.

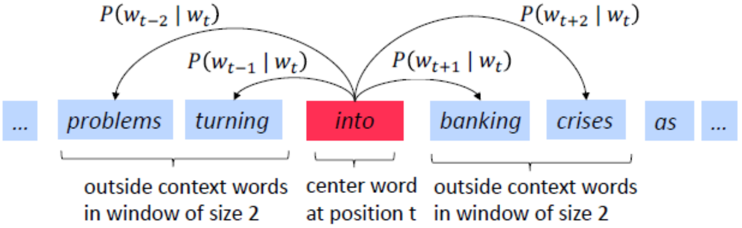
Idea:

- We have a large corpus of text
- Every word in a fixed vocabulary is represented by a vector.
- Go through each position t in the text, which has a center word c and context ("outside") words o.
- Use the similarity of the word vectors for c and o to calculate the probability of o given c (or vice versa).
- Keep adjusting the word vectors to maximize this probability.

[1] Tomas Mikolov et al.,Distributed Representations of Words and Phrases and Their Compositionality, NIPS, Volume 2, 2013, 3111–3119

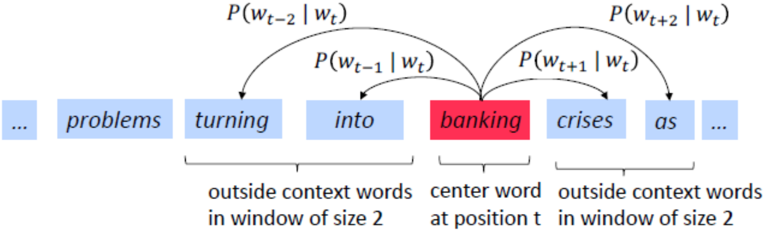
Word Representation - Word2Vec

Example windows and process for computing $P(w_{t+1}|w_t)$



http://web.stanford.edu/class/cs224n/slides/cs224n-2019-lecture01-wordvecs1.pdf

Example windows and process for computing $P(w_{t+1}|w_t)$



http://web.stanford.edu/class/cs224n/slides/cs224n-2019-lecture01-wordvecs1.pdf

Word Representation - Word2Vec

- Two Algorithm

Word Representation - Word2Vec

- Two Algorithm
  1. Continuous bag-of-words (CBOW)

Word Representation - Word2Vec

- Two Algorithm
  1. Continuous bag-of-words (CBOW)
     - Predict a center word from the surrounding context in terms of word vectors.

Word Representation - Word2Vec

- Two Algorithm
  1. Continuous bag-of-words (CBOW)
     - Predict a center word from the surrounding context in terms of word vectors.
  2. Skip-gram

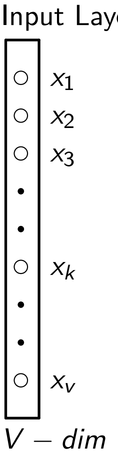
Word Representation - Word2Vec

- Two Algorithm
  1. Continuous bag-of-words (CBOW)
     - Predict a center word from the surrounding context in terms of word vectors.
  2. Skip-gram
     - Predicts the probability of context words from a center word.

Word Representation - Word2Vec

- Two Algorithm
  1. Continuous bag-of-words (CBOW)
     - Predict a center word from the surrounding context in terms of word vectors.
  2. Skip-gram
     - Predicts the probability of context words from a center word.

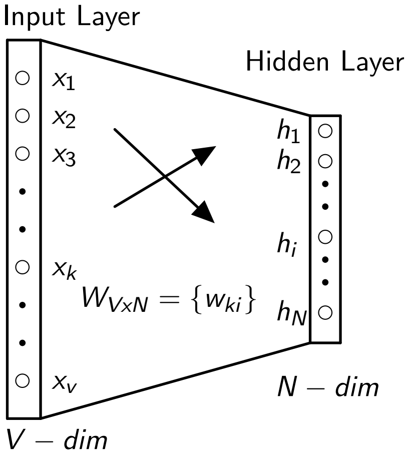
Word Representation - Word2Vec - CBOW



A simple CBOW model with only one word in the context

$V =$ Vocabulary size $N =$ Word embedding vector size

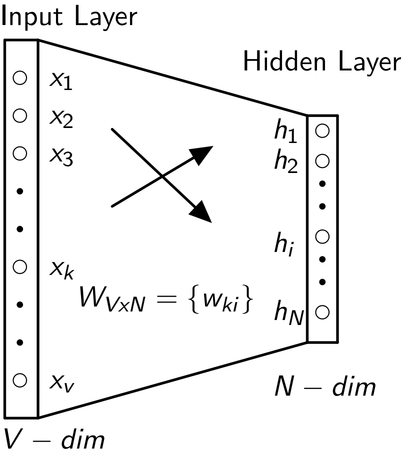
Word Representation - Word2Vec - CBOW



A simple CBOW model with only one word in the context

V = Vocabulary size N = Word embedding vector size

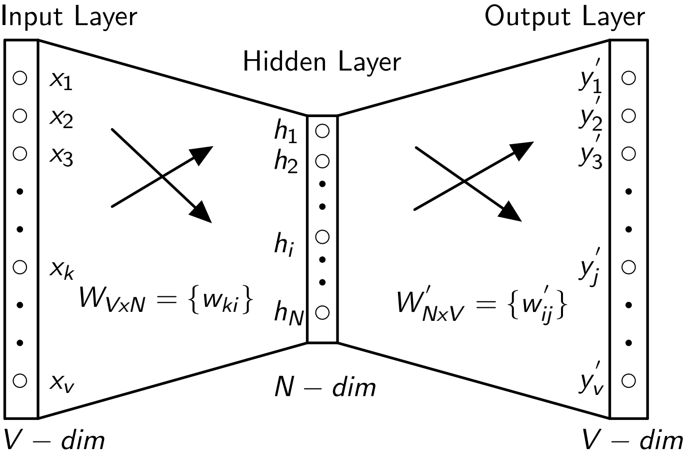
Word Representation - Word2Vec - CBOW



A simple CBOW model with only one word in the context

V = Vocabulary size N = Word embedding vector size

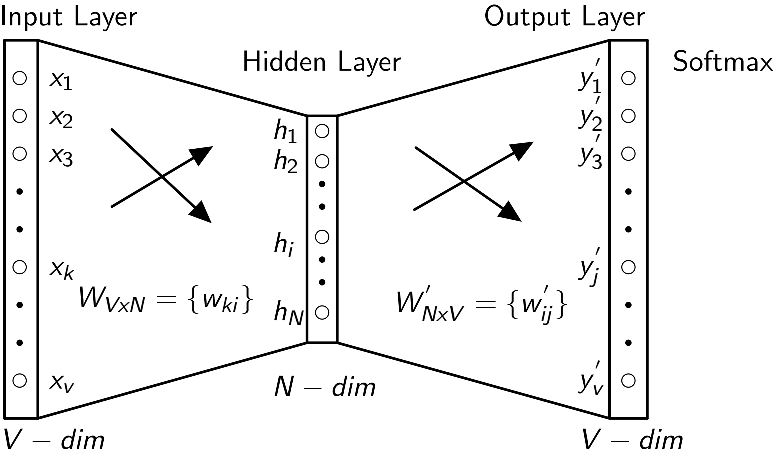
Word Representation - Word2Vec - CBOW



A simple CBOW model with only one word in the context

V = Vocabulary size N = Word embedding vector size

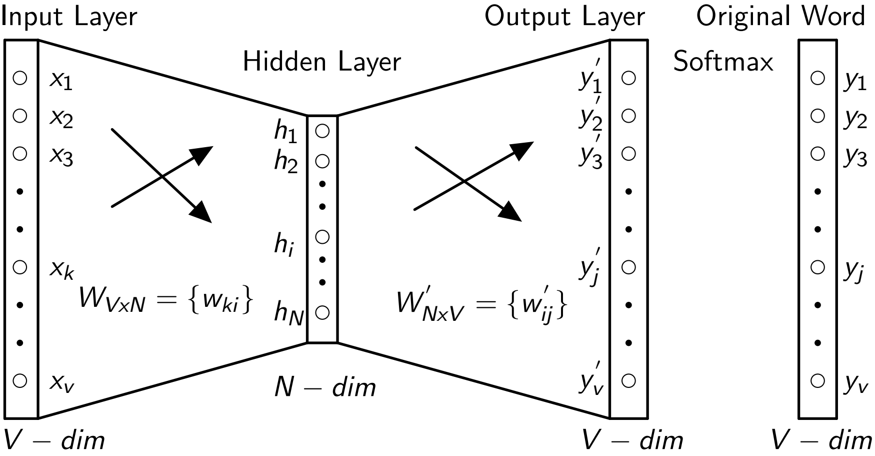
Word Representation - Word2Vec - CBOW



A simple CBOW model with only one word in the context

V = Vocabulary size N = Word embedding vector size

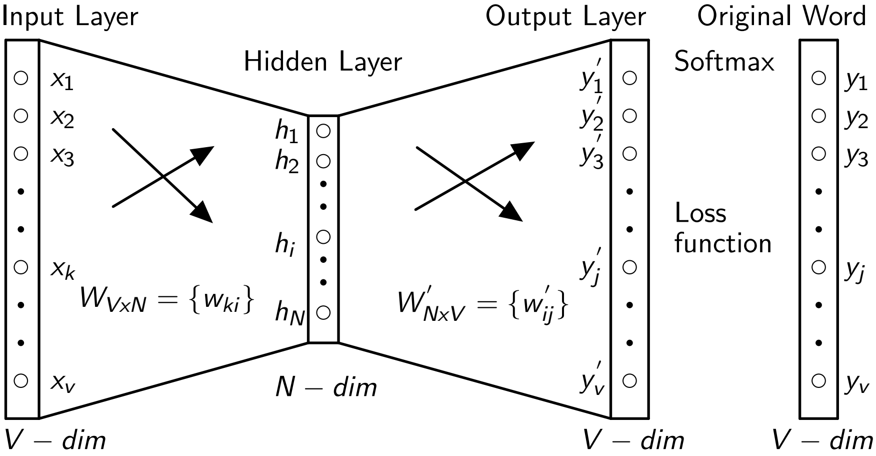
Word Representation - Word2Vec - CBOW



A simple CBOW model with only one word in the context

V = Vocabulary size N = Word embedding vector size

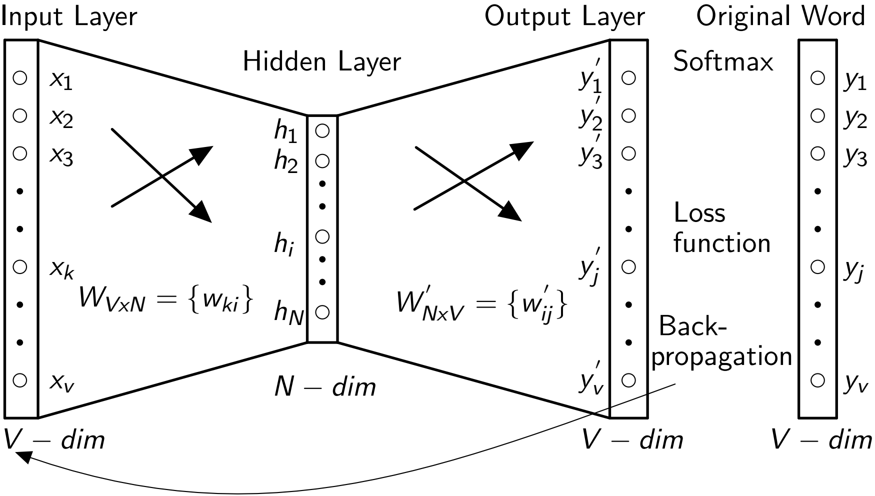
Word Representation - Word2Vec - CBOW



A simple CBOW model with only one word in the context

V = Vocabulary size N = Word embedding vector size

Word Representation - Word2Vec - CBOW



A simple CBOW model with only one word in the context

V = Vocabulary size N = Word embedding vector size

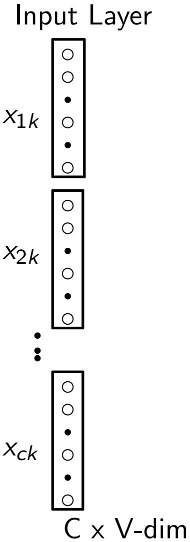
Word Vector - Word2Vec - CBOW

Example: How we take word embedding/ word vector from Weight $W'_{N \times V}$

$$[0 \quad 0 \quad 0 \quad \boxed{1} \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ \boxed{10} & \boxed{12} & \boxed{19} \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

https://israelg99.github.io/2017-03-23-Word2Vec-Explained/

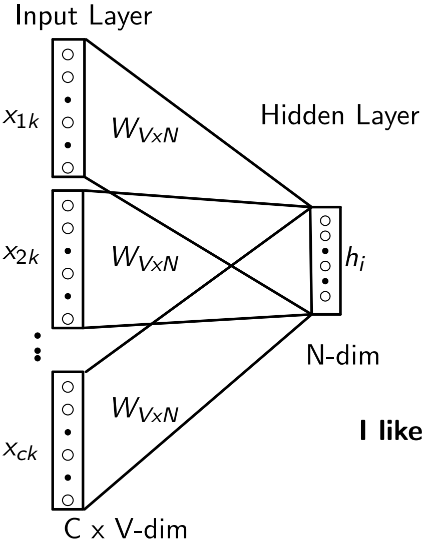
Word Vector - Word2Vec - CBOW



I like **playing** **football with** friend

CBOW Architecture

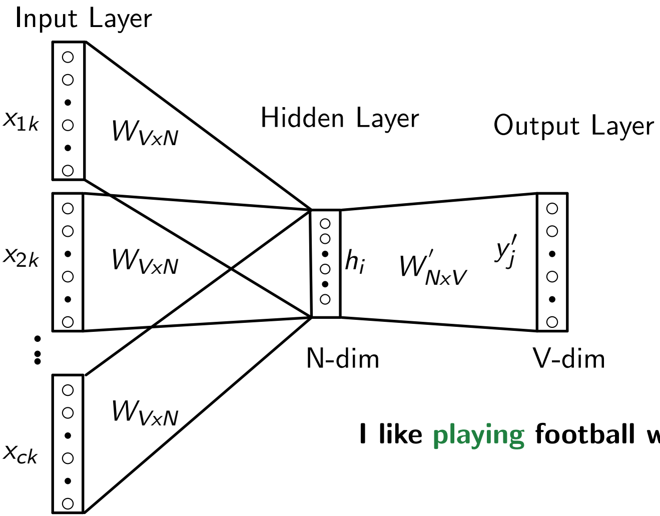
Word Vector - Word2Vec - CBOW



I like **playing** **football with** friend

CBOW Architecture

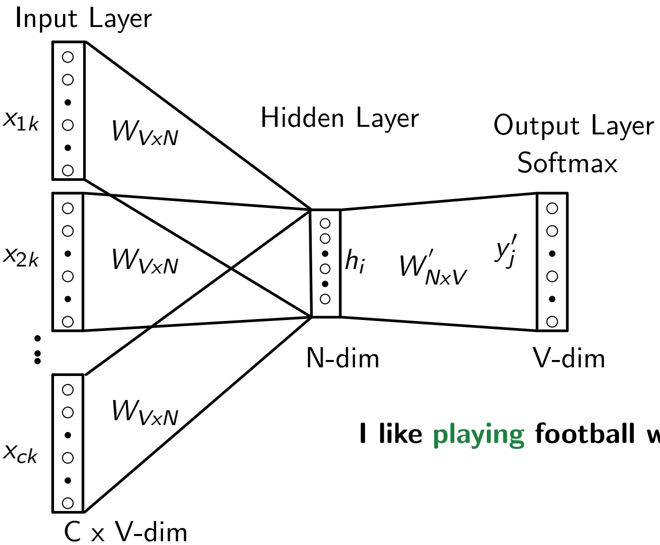
Word Vector - Word2Vec - CBOW



I like **playing** **football with** friend

CBOW Architecture

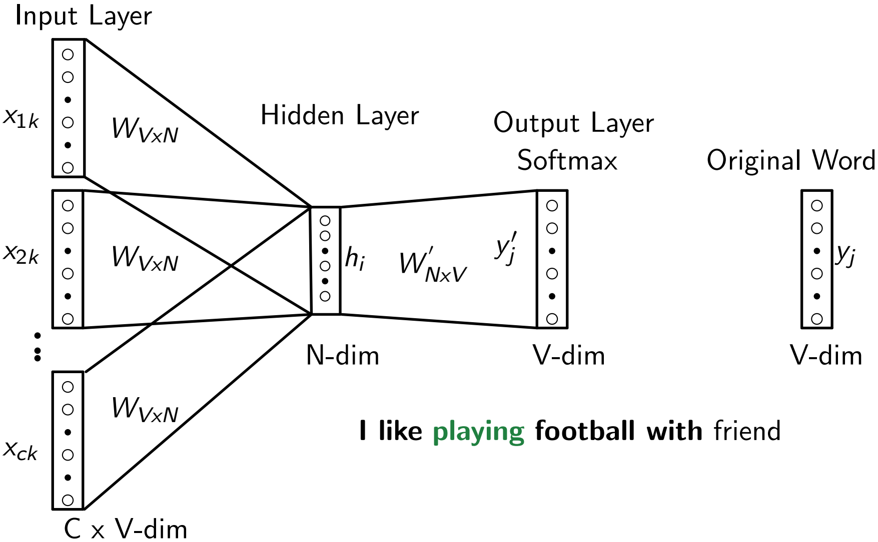
Word Vector - Word2Vec - CBOW



I like **playing** **football with** friend

CBOW Architecture

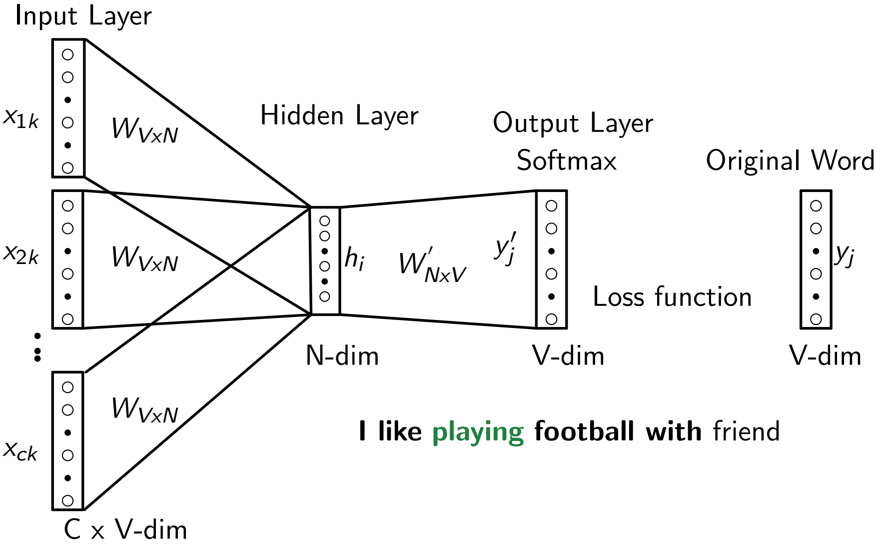
Word Vector - Word2Vec - CBOW



I like **playing** **football with** friend

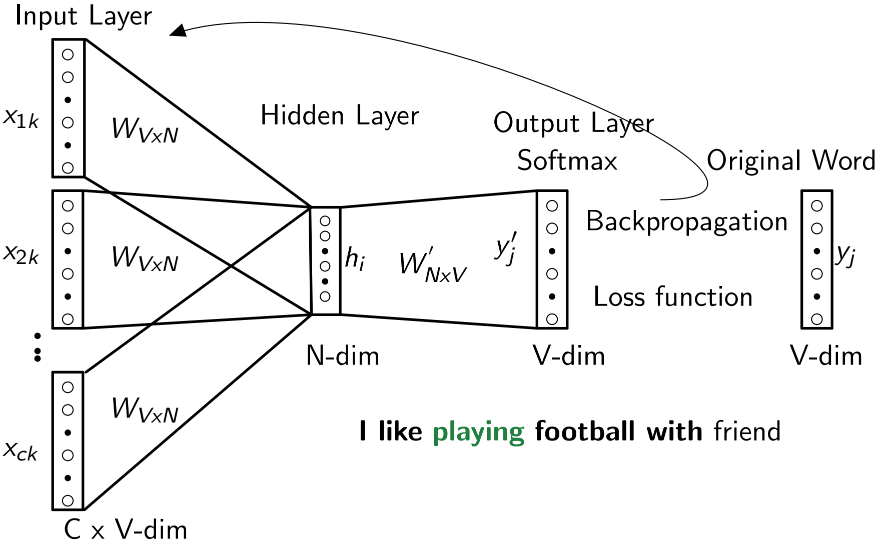
CBOW Architecture

Word Vector - Word2Vec - CBOW



CBOW Architecture

Word Vector - Word2Vec - CBOW



I like playing **football with** friend

CBOW Architecture

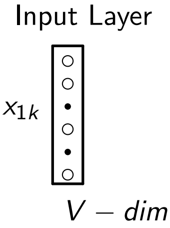
Word Representation - Word2Vec - CBOW

- Input layer takes the one hot encoded vectors of the context words as input.
- Two sets of weights
  1. between input layer and hidden layer (size - VxN)
  2. between input layer and hidden layer (size - NxV)
- Input is multiplied with input-hidden weights and hidden input is multiplied with hidden-output weights to produce a output vector.
- Output vector is then passed through a softmax function which gives the probability of target word w.r.t each context word.
- Target word is the one hot encoding representation of the word
- Error between output and target is calculated and then update weights.

Word Representation - Word2Vec - CBOW - Softmax

- Simply calculates probability of occurrence of the target word w.r.t. the context word
- Consider that target word is denoted by vector u_c and context words are denoted by vectors $\hat{v}$.
- The probability sums to 1

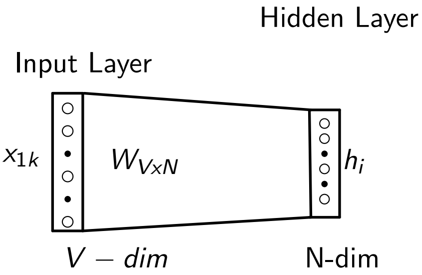
$$P(u_c|\hat{v}) = \frac{exp(u_c^T \hat{v})}{\sum_{j=1}^{|V|} exp(u_j^T \hat{v})}$$



I like **playing** **football with** friend

Skip-gram Architecture

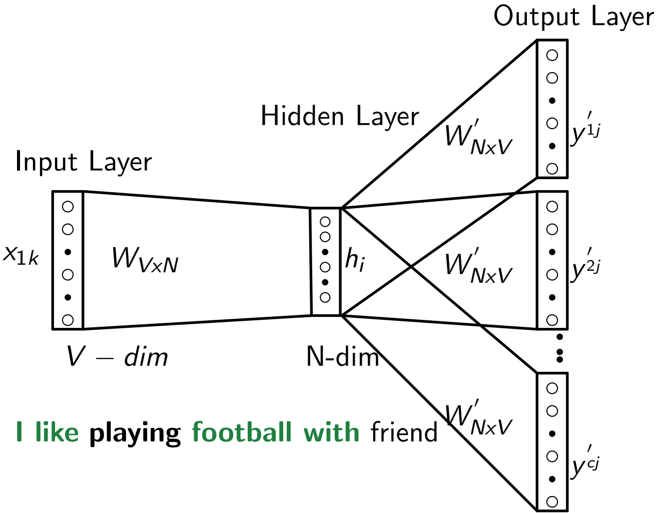
Word Representation - Word2Vec - Skip gram



I like **playing** **football with** friend

Skip-gram Architecture

Word Representation - Word2Vec - Skip gram

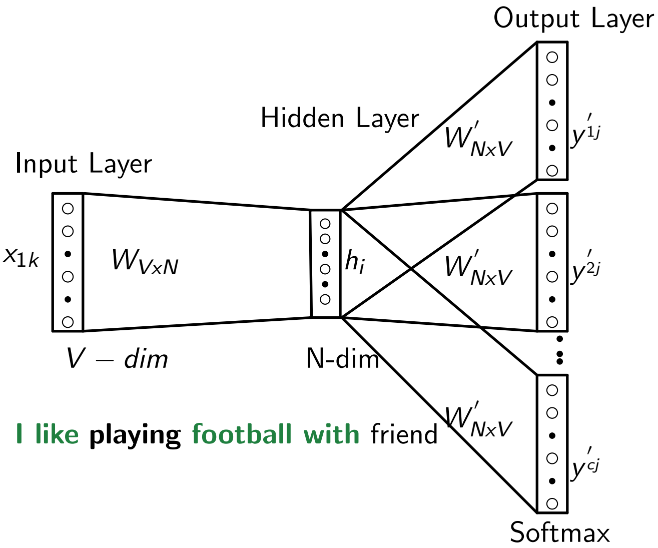


I like **playing** **football with** friend

Skip-gram Architecture

$CxV-dim$

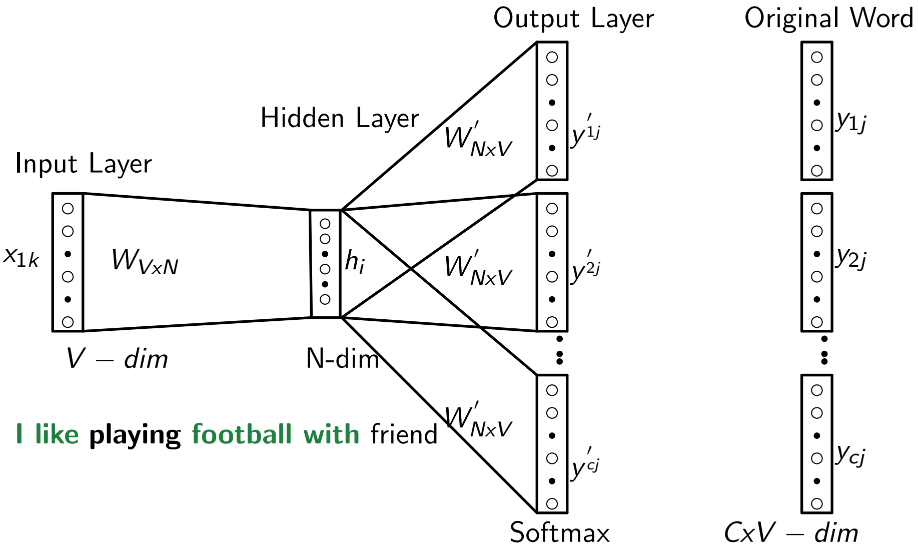
Word Representation - Word2Vec - Skip gram



Skip-gram Architecture

$CxV - dim$

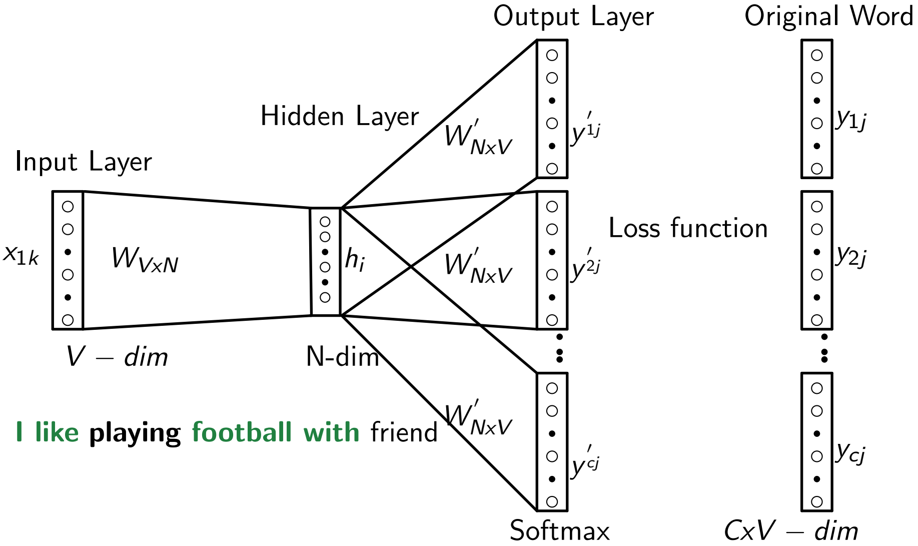
Word Representation - Word2Vec - Skip gram



Skip-gram Architecture

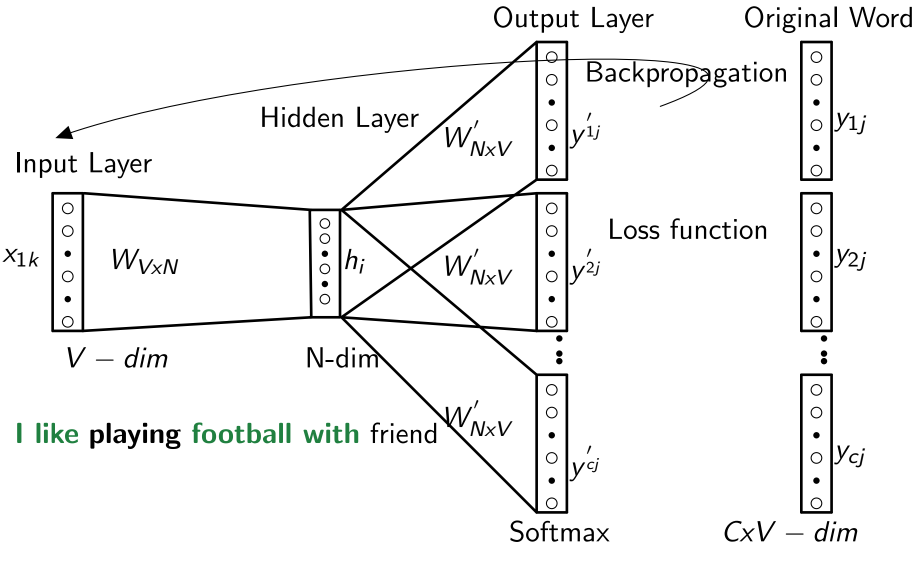
$CxV - dim$

Word Representation - Word2Vec - Skip gram



Skip-gram Architecture $CxV - dim$

Word Representation - Word2Vec - Skip gram



Skip-gram Architecture

Word Representation - Word2Vec - Skip gram

- One hot encoded vector of target word is taken as input and multiplied with VxN input hidden matrix.
- Hidden layer is multiplied with hidden output matrix to give the output vectors.
- Take softmax function over the output vectors and then calculate the error between the actual vector and the predicted vectors and backpropagate to update the weights.

Word Representation - Word2Vec - Objective Function

- For each position $t = 1, \ldots, T$, predict context words within a window of fixed size m, given center word w_j.

$$\text{Likelihood} \ = \ L(\theta) \ = \prod_{t=1}^{T} \prod_{-m \leq j \leq m, \, j \neq 0} P(w_{t+j} | w_t; \theta)$$

Word Representation - Word2Vec - Objective Function

- For each position $t = 1, \ldots, T$, predict context words within a window of fixed size m, given center word w_j.

$$\text{Likelihood} \; = \; L(\theta) \; = \prod_{t=1}^{T} \prod_{-m \leq j \leq m,\, j \neq 0} P(w_{t+j} | w_t; \theta)$$

- The cost function $J(\theta)$ is the (average) negative log likelihood:

$$J(\theta) \; = -\frac{1}{T} \log L(\theta) \; = -\frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m,\, j \neq 0} P(w_{t+j} | w_t; \theta)$$

Word Representation - Word2Vec - Objective Function

- For each position $t = 1, \ldots, T$, predict context words within a window of fixed size m, given center word w_j.

$$\text{Likelihood} = L(\theta) = \prod_{t=1}^{T} \prod_{-m \leq j \leq m,\, j \neq 0} P(w_{t+j} | w_t; \theta)$$

- The cost function $J(\theta)$ is the (average) negative log likelihood:

$$J(\theta) = -\frac{1}{T} \log L(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m,\, j \neq 0} P(w_{t+j} | w_t; \theta)$$

Word Representation - Word2Vec - Limitation

Word Representation - Word2Vec - Limitation

- Relies only on local information of a corpus.

Word Representation - Word2Vec - Limitation

- Relies only on local information of a corpus.
- The semantics learnt for a given word, is only affected by the surrounding words.

Word Representation - Word2Vec - Limitation

- Relies only on local information of a corpus.
- The semantics learnt for a given word, is only affected by the surrounding words.
- Require to look local information and global information of a corpus.