Эмбеддинги в NLP

Закиева Азалия, БПМИ172

План

I. Что такое эмбеддинг
II. Примитивные способы
III. word2vector
IV. GloVe
V. FastText
VI. ELMo

Что такое эмбеддинг

Natural Language Processing (NLP)

Обработка естественного языка

A. Анализ - чтение, понимание и извлечение смысла

B. Синтез – генерация грамотного текста

Использование:

- Распознавание речи
- Анализ текста
- Информационный поиск
- Анализ тональности текста
- Вопросно-ответные системы
- Генерирование текста
- Машинный перевод

Word embedding

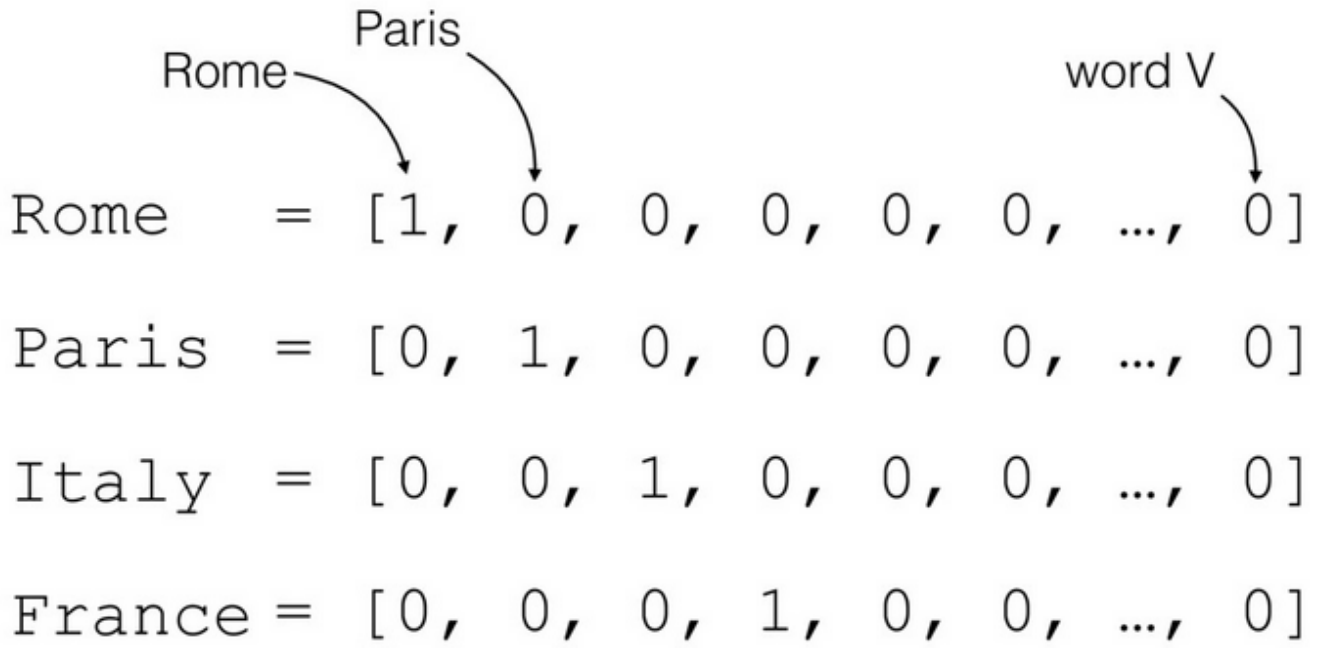
- Это преобразование языковой сущности (слова, предложения, параграфа, текста) в набор чисел (числовой вектор), для дальнейшей работы с векторами для обучения модели

- Главная задача - отобразить семантику слов

Примитивные способы

One-hot encoding: описание

Простая нумерация слов в некотором словаре и установка значения единицы в длинном векторе размерности, равной числу слов в словаре.



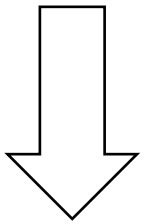
V – размер словаря

One-hot encoding: недостатки

- Проблема сходства = нет связи
- Размер словаря/длина вектора
- Вычисление: дорогая вставка, разреженные вектора

Feature vectors

Представляем слово в виде распределения его принадлежности к каждому из признаков = как вектор семантических признаков

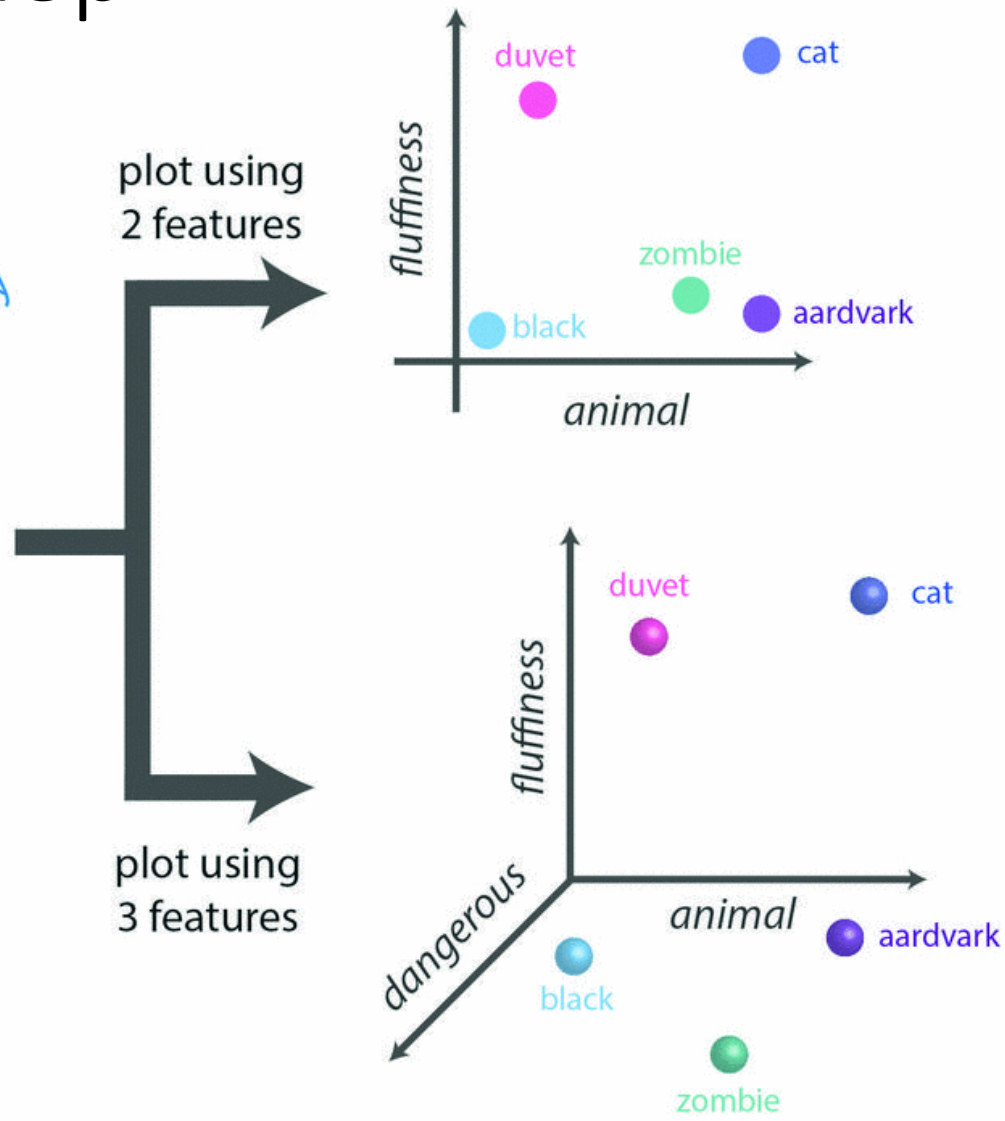


- Можем представлять слово как вектор в измерениях этих признаков (уже появляется зависимость между близкими по смыслу словами)

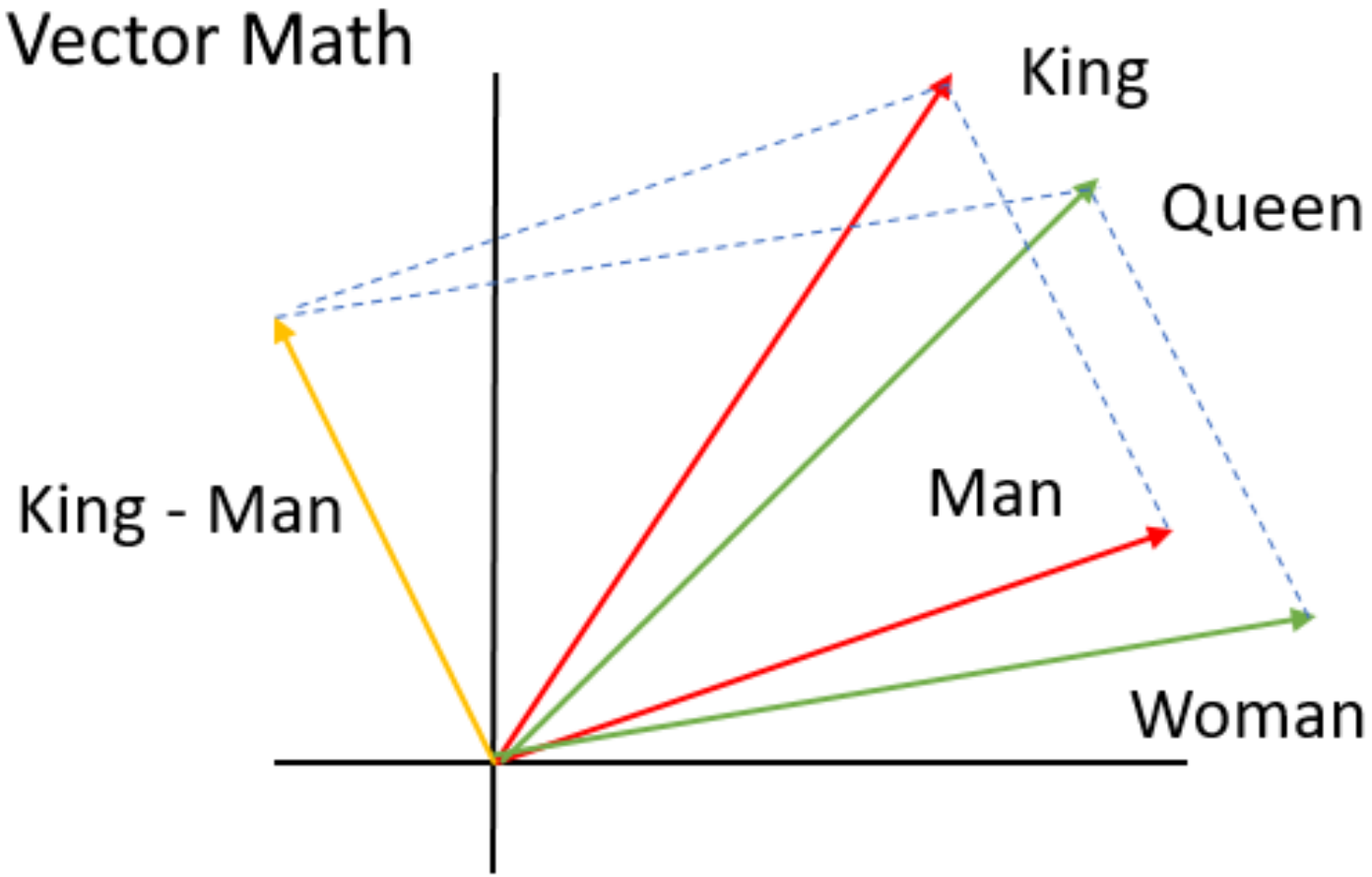
- Можем использовать математические операторы

Feature vectors: пример

	animal	fluffiness	dangerous	spooky
aardvark	0.97	0.03	0.15	0.04
black	0.07	0.01	0.20	0.95
cat	0.98	0.98	0.45	0.35
duvet	0.01	0.84	0.12	0.02
zombie	0.74	0.05	0.98	0.93



Feature vectors: пример



king – man +
woman = queen

word2vec - word2vector

word2vec

- Идея: использовать гипотезу локальности: «слова, которые встречаются в одинаковых окружениях, имеют близкие значения»

- Алгоритм:
  A. Continuous Bag of Words – предсказываем текущее слово от окружающих слов
  B. Skip-gram – предсказываем окружающие слова от текущего слова

Skip-gram

The man who passes the sentence should **swing** the sword

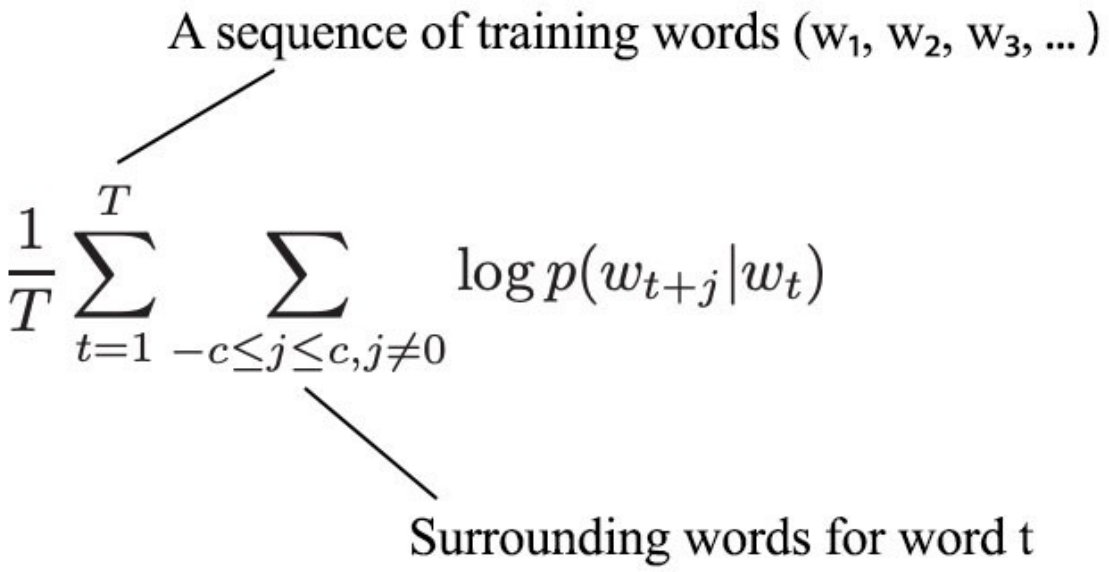
Sliding window (size = 5)	Target word	Context
[The man who]	the	man, who
[The man who passes]	man	the, who, passes
[The man who passes the]	who	the, man, passes, the
[man who passes the sentence]	passes	man, who, the, sentence
...	...	...
[sentence should swing the sword]	swing	sentence, should, the, sword
[should swing the sword]	the	should, swing, sword
[swing the sword]	sword	swing, the

target word – «swing»

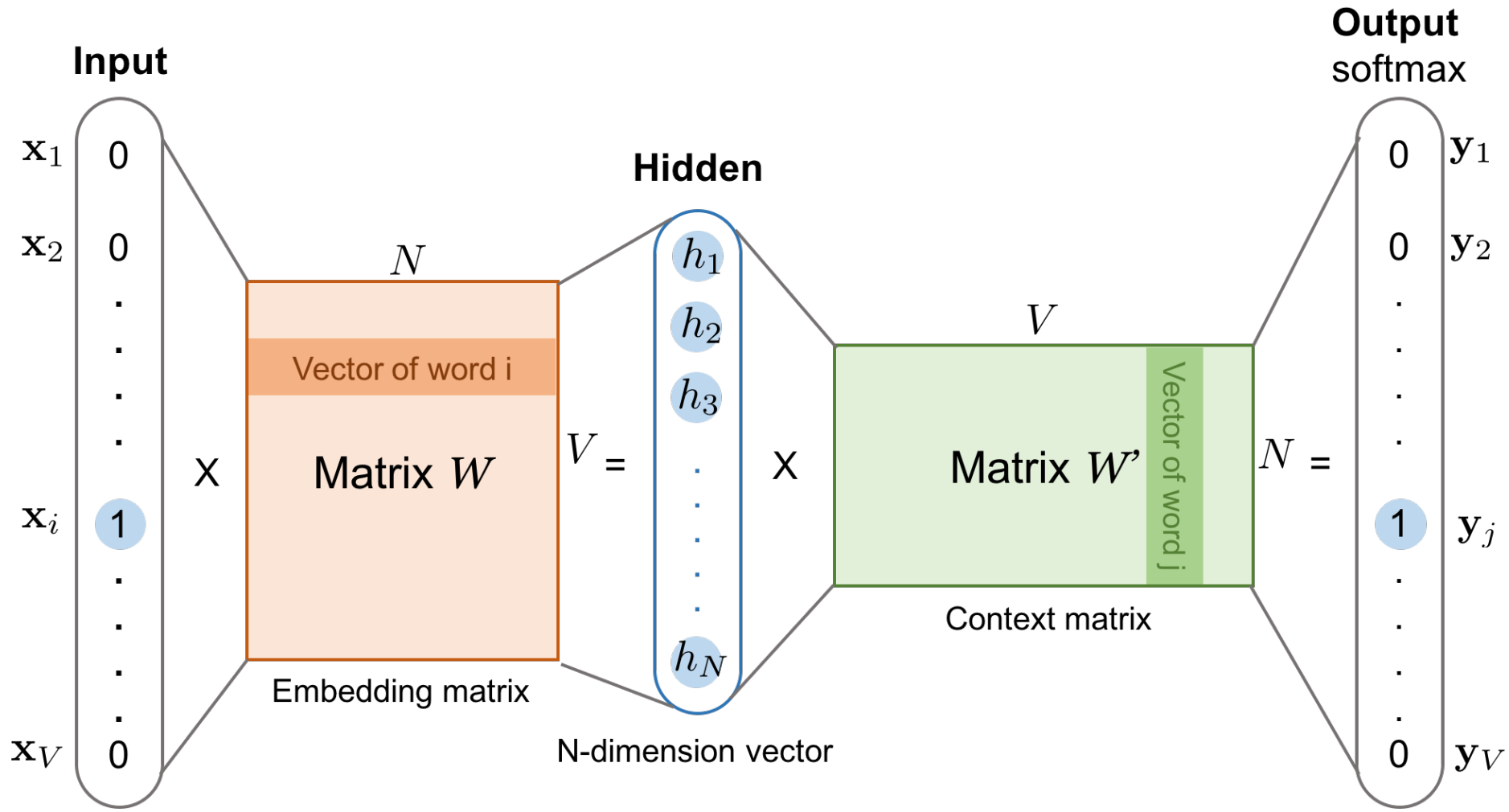
training samples:
(“swing”, “sentence”)
(“swing”, “should”)
(“swing”, “the”)
(“swing”, “sword”)

New England **Patriots** win 14th straight regular-season game at home in Gillette stadium

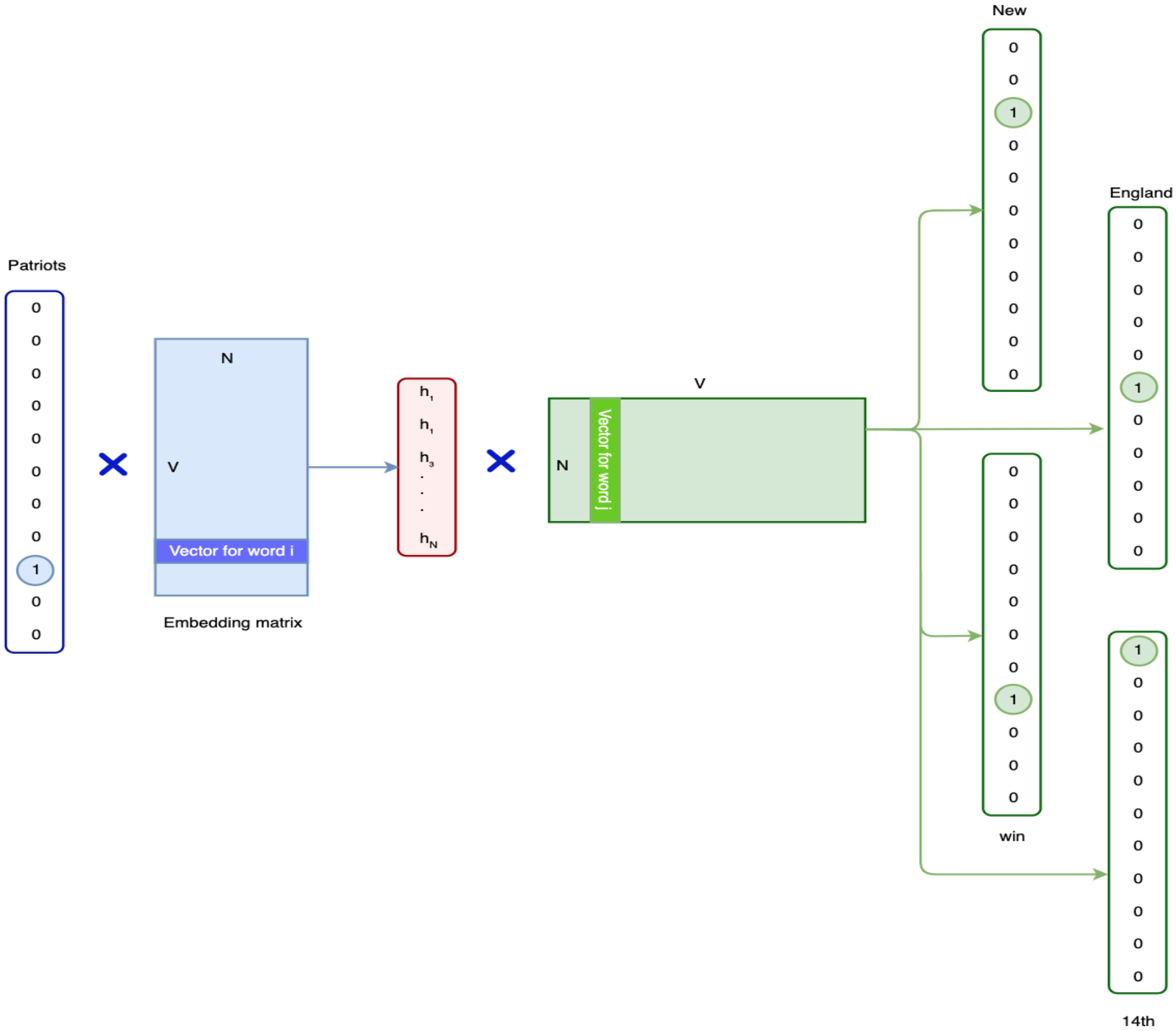
- log-likelihood для предсказанных слов с учетом целевого слова t (**Patriots**) хотим максимизировать:



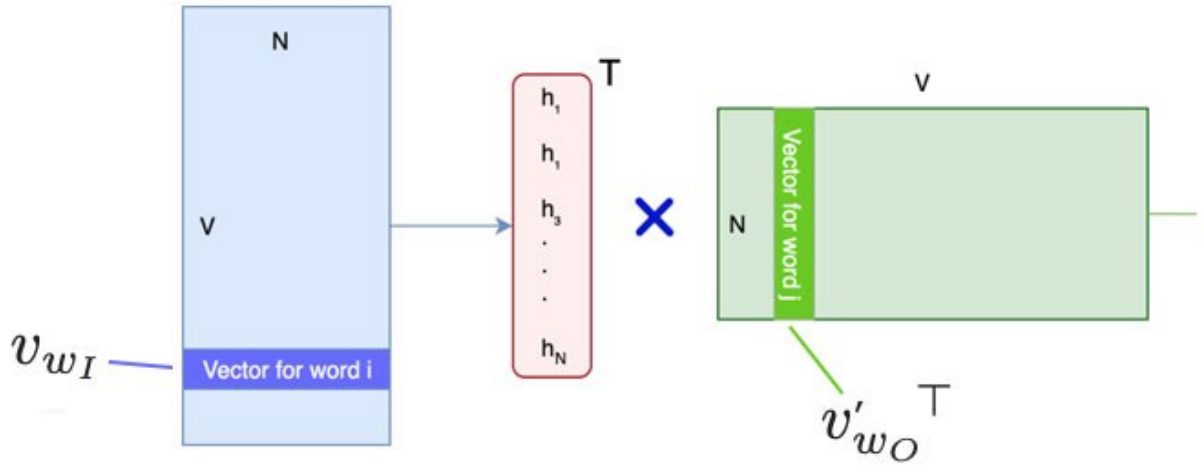
Skip-gram


Input
Output
softmax
Hidden
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_i$
$\mathbf{x}_V$
0
1
N
Vector of word i
Matrix W
Embedding matrix
V =
h_1
h_2
h_3
h_N
N-dimension vector
V
Matrix W'
Vector of word j
Context matrix
N =
$\mathbf{y}_1$
$\mathbf{y}_2$
$\mathbf{y}_j$
$\mathbf{y}_V$

Skip-gram


Patriots
0
0
0
0
0
0
0
0
1
0
0
N
V
Vector for word i
Embedding matrix
h1
h1
h3
hN
V
N
Vector for word j
New
0
0
1
0
0
0
0
0
0
0
0
England
0
0
0
0
0
1
0
0
0
0
0
win
0
0
0
0
0
0
1
0
0
0
0
1
0
0
0
0
0
0
0
0
0
0
0
14th

Хотим вычислить условную вероятность P:
мы находим соответствующие строки и столбцы, связанные с w_I и w_O в соответствующей матрице



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

sum over all words in the vocabulary (normalization)

where v_w and v'_w are the "input" and "output" vector representations of w

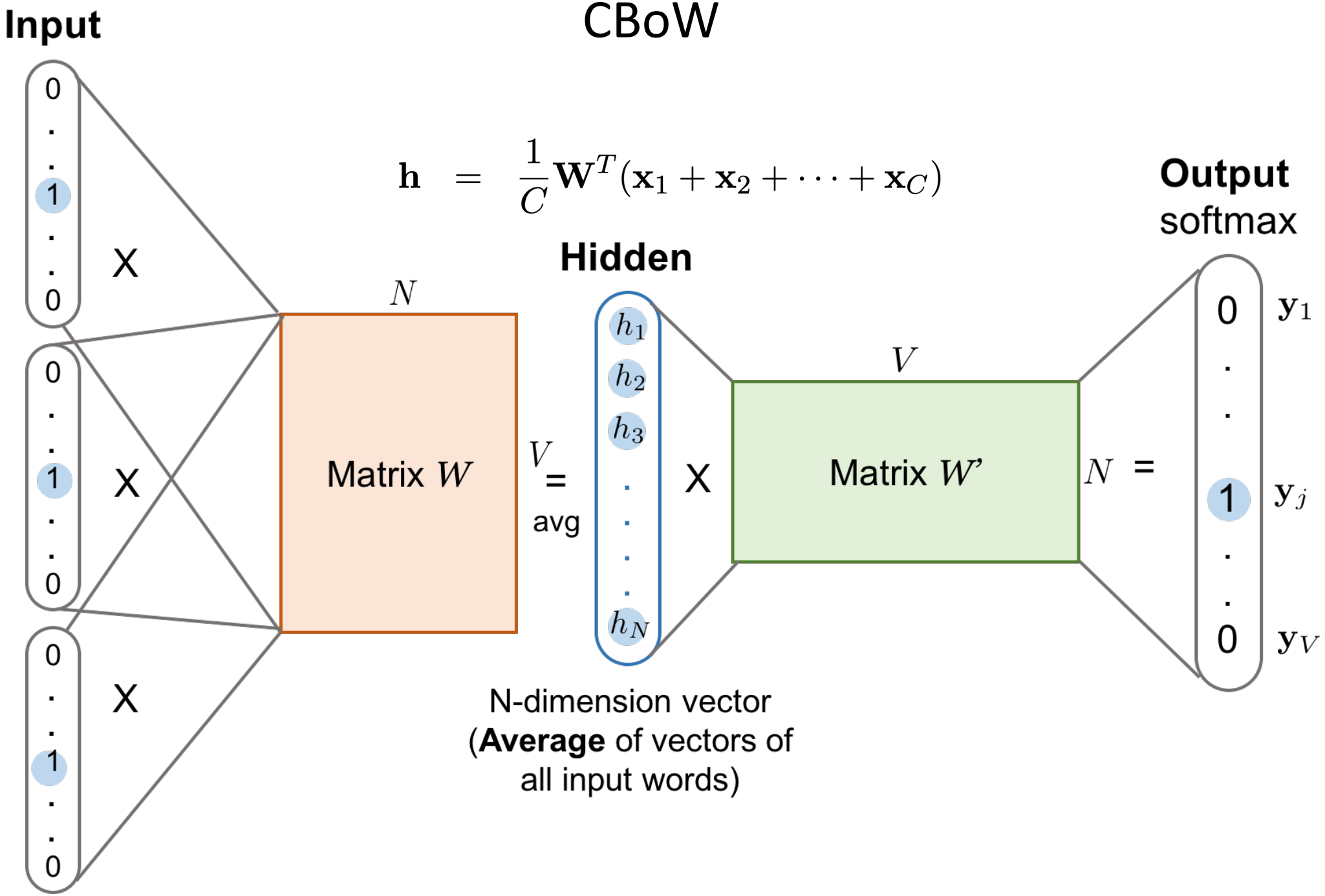
Функция потерь: Cross Entropy loss

$$\mathcal{L}_\theta = -\sum_{i=1}^{V} y_i \log p(w_i|w_I) = -\log p(w_O|w_I)$$

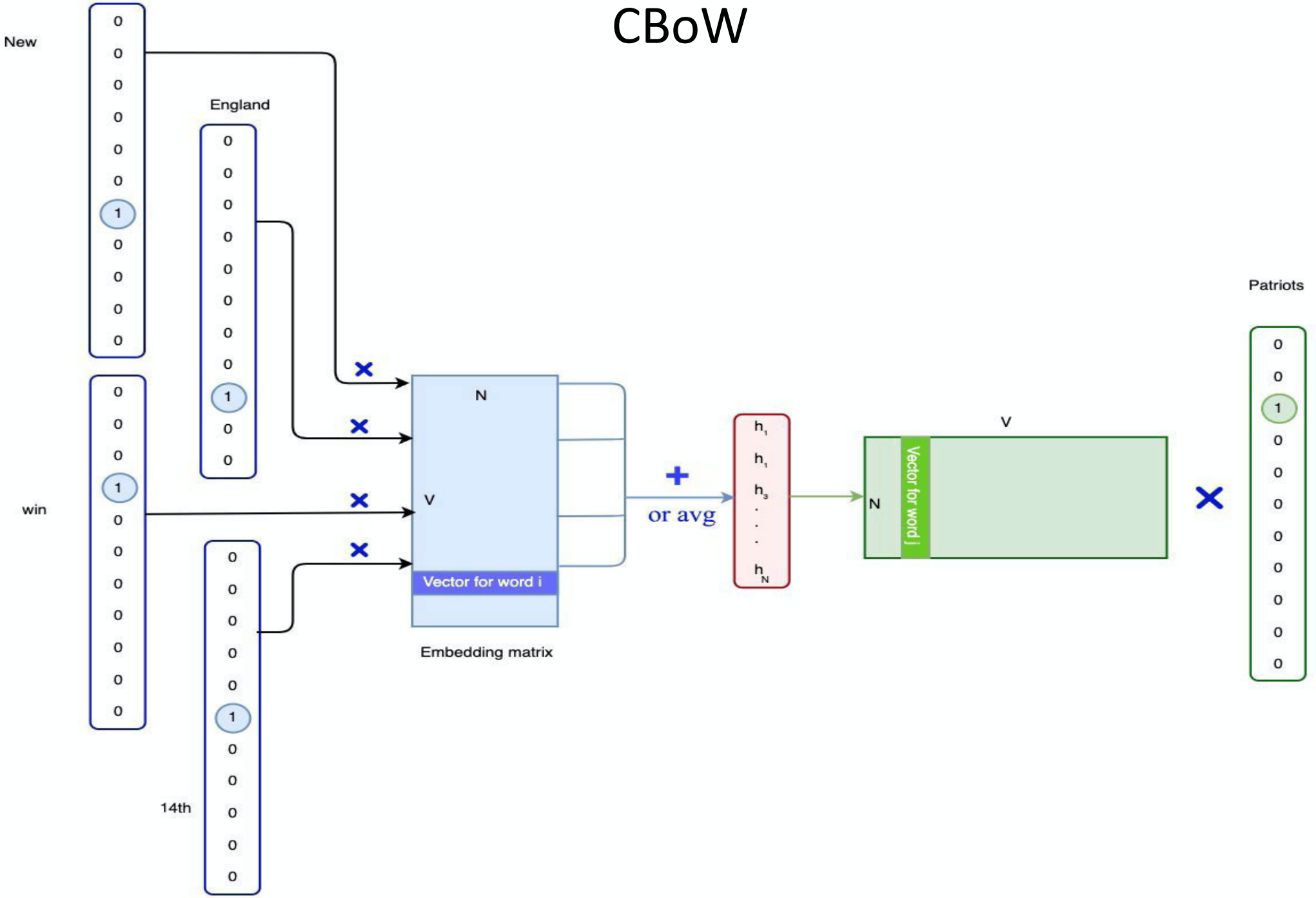
$$\mathcal{L}_\theta = -\log \frac{\exp({v'_{w_O}}^\top v_{w_I})}{\sum_{i=1}^{V} \exp({v'_{w_i}}^\top v_{w_I})} = -{v'_{w_O}}^\top v_{w_I} + \log \sum_{i=1}^{V} \exp({v'_{w_i}}^\top v_{w_I})$$

Обучение - back propagation с SGD

CBoW


Input
Hidden
Output
softmax
$\mathbf{h} = \frac{1}{C}\mathbf{W}^T(\mathbf{x}_1 + \mathbf{x}_2 + \cdots + \mathbf{x}_C)$
Matrix W
Matrix W'
N
V
V = avg
N =
h_1 h_2 h_3 h_N
$\mathbf{y}_1$ $\mathbf{y}_j$ $\mathbf{y}_V$
N-dimension vector
(**Average** of vectors of
all input words)

CBoW


New
England
win
14th
N
V
Vector for word i
Embedding matrix
or avg
V
N
Vector for word j
Patriots

Negative Sampling

- Softmax считается долго + каждая обучающая выборка будет настраивать все веса в нейронной сети – затратно считать потом градиенты
- Будем выбирать только небольшое количество отрицательных слов для обновления весов. Мы также будем по-прежнему обновлять веса для нашего положительного слова
- Вместо суммирования по всем словам, мы суммируем только несколько отрицательных слов, т. е. слов, которые ошибочны

Сравнение

- Skip-gram хорошо работает с небольшим количеством данных и хорошо представляет редкие слова.
- CBoW быстрее и имеет лучшие представления для более частых слов.

GloVe - Global Vectors

The cat sat on the mat

- Word2vec не сможет понять является ли «the» особым контекстом слов «cat» и «mat» или «the» просто шумовое слово?

- GloVe использует как глобальную статистику, так и локальную статистику корпуса текста, чтобы создавать векторы слов

The cat sat on the mat

- Семантические отношения между словами можно вывести из Матрицы совпадений:

Размер окна = 1

	the	cat	sat	on	mat
the	0	1	0	1	1
cat	1	0	1	0	0
sat	0	1	0	1	0
on	1	0	1	0	0
mat	1	0	0	0	0

X_{ij} - показывает количество раз, когда слово j встречается в контексте слова i

$X_i = \sum_k X_{ik}$ - количество раз, когда любое слово появляется в контексте слова i

GloVe

- Хотим получить метрику, которая измеряет семантическое сходство между словами

X_{ij} - показывает количество раз, когда слово j встречается в контексте слова i
$X_i = \sum_k X_{ik}$ - количество раз, когда любое слово появляется в контексте слова i
$P_{ij} = P(j|i) = X_{ij} / X_i$ - вероятность того, что слово j появится в контексте слова i

Таблица отношение вероятностей совместного возникновения

$w \in \mathbb{R}^d$ are word vectors

probe word

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$$

co-relations between the word w*i* and w*j*

co-occurrence probabilities for the word w*j* and w*k*

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

Very small or large:

solid is related to ice but not steam, or
gas is related to steam but not ice

close to 1:

water is highly related to ice and steam, or
fashion is not related to ice or steam.

1. Линейность

$w_i^T \tilde{w}_k$ relate to (high probability if they are similar)

$$F\left((w_i - w_j)^T \tilde{w}_k\right) = \frac{P_{ik}}{P_{jk}}$$

$w_j^T \tilde{w}_k$

2. Симметрия

$$F\left((w_i - w_j)^T \tilde{w}_k\right) = \frac{F(w_i^T \tilde{w}_k)}{F(w_j^T \tilde{w}_k)}$$

3. $F(x) = \exp(x)$ - поддерживаем линейность

4. $$F(w_i^T \tilde{w}_k) = P_{ik} = \frac{X_{ik}}{X_i}$$

GloVe

$$w_i^T \tilde{w}_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i)$$

$$w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log(X_{ik})$$

co-occurrence count for word w*i* and w*k*

Функционал ошибки

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

$$f(x) = \begin{cases} (x/x_{\max})^{\alpha} & \text{if } x < x_{\max} \\ 1 & \text{otherwise} \end{cases}$$

FastText – расширение word2vec

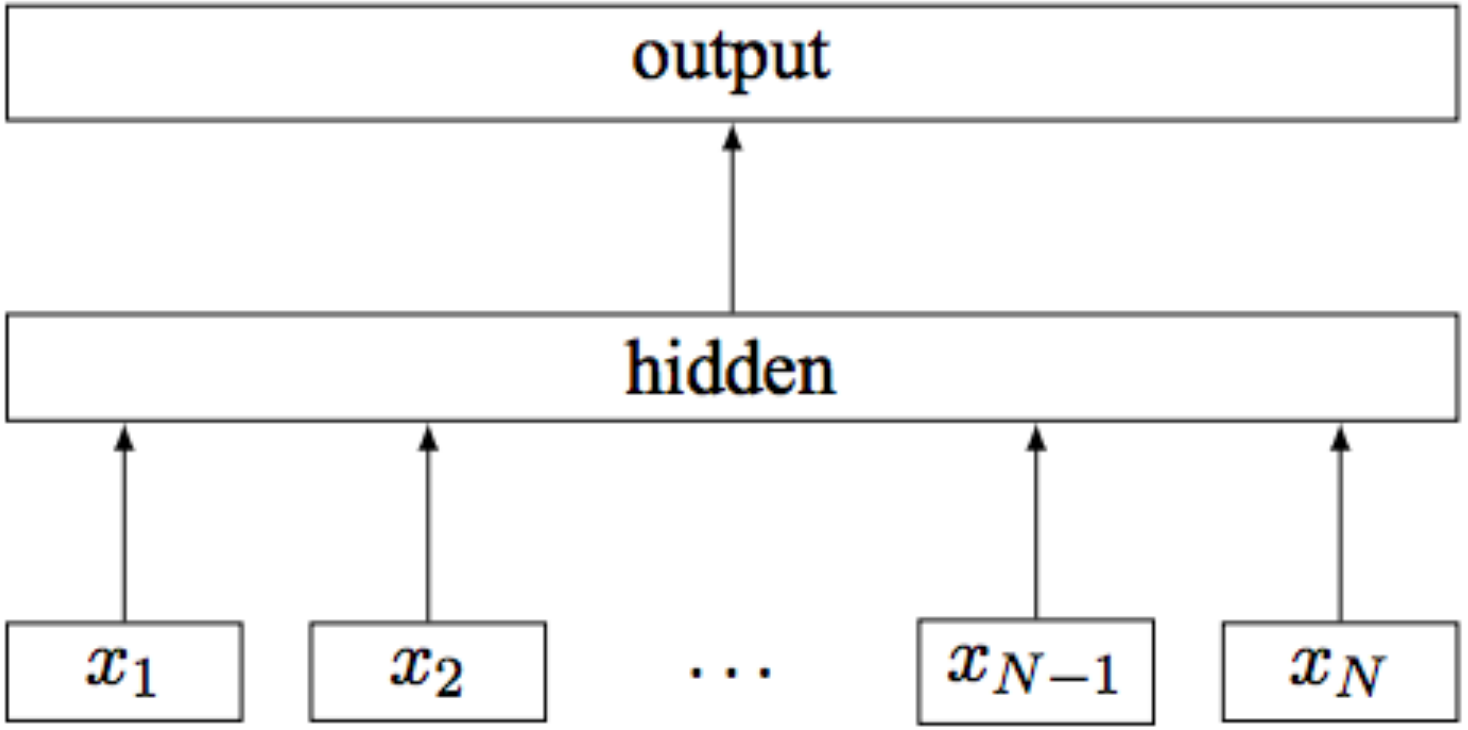
FastText

- Используем Skip-gram with Negative Sampling (SGNS) с небольшими изменениями
- N-gram метод – разделение слова на подслова
- Например, диапазон символов N-gram = 3 - 5 подслов:

Banana - ban, ana, nan, bana, anan, nana, banan, anana

- Эмбеддинг слова Banana представляется как сумма эмбеддинговых подслов

FastText



Вопросы

1. Опишите идею и алгоритм w2v (Skip-gram)
2. В чем основное отличие FastText от w2v?
3. Когда нужно использовать Elmo?

Список источников

- https://towardsdatascience.com/your-guide-to-natural-language-processing-nlp-48ea2511f6e1
- https://medium.com/@b.terryjack/nlp-everything-about-word-embeddings-9ea21f51ccfe
- https://arxiv.org/pdf/1411.2738.pdf
- https://arxiv.org/pdf/1301.3781.pdf
- https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf
- https://nlp.stanford.edu/pubs/glove.pdf
- https://arxiv.org/pdf/1607.04606.pdf
- https://arxiv.org/pdf/1607.01759.pdf