DEEP LEARNING CIn.AI Centro de Informática UFPE Universidade Federal de Pernambuco

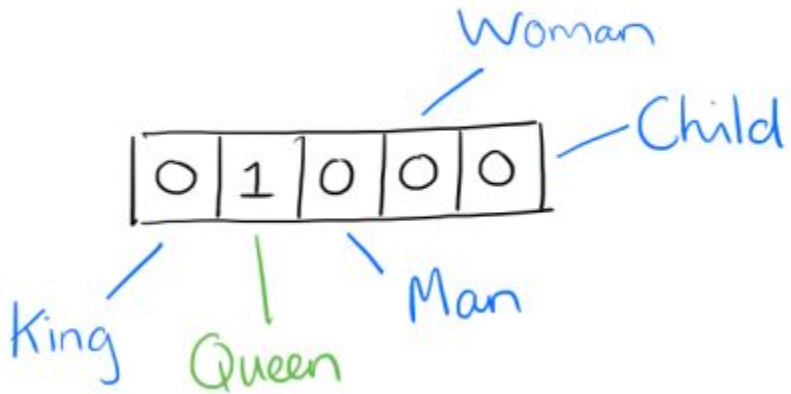
Processamento de Linguagem Natural

Representação de Características Textuais

Prof. Luciano Barbosa &
Prof. Johny Moreira
{luciano, jms5}@cin.ufpe.br

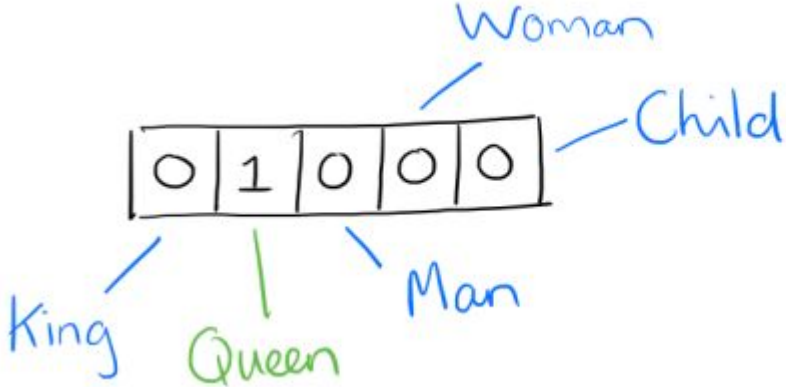
O que são Representações de Palavras e porque utilizá-las.

- Algoritmos de Machine Learning e Deep Learning não entendem textos, palavras, suas sintaxes ou significados semânticos
- É necessário transformar o texto em um formato que seja entendido por máquinas: números



Representação Categórica

One-hot encoding



One Hot Encoding

- Anteriormente fazíamos uma codificação simples dos dados categóricos (textuais) para inteiros. Porém, pode acabar sendo confuso -> indicar ordem

```
{
   0: Brasil,
   1: Japão,
   2: Chile,
   3: França, …
}
```

- Cada palavra é mapeada para uma dimensão de um vetor.
- O vetor gerado é a representação da palavra
- Abordagem simples

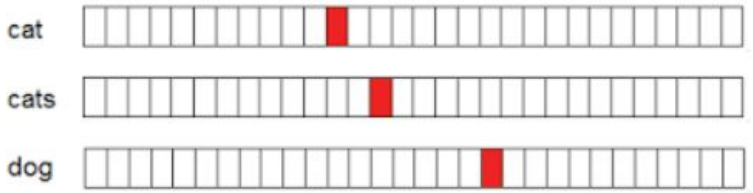
One Hot Encoding

Dimensionalidade do tamanho do vocabulário

V: {o, brasil, sediou, perdeu, a, copa}

Codificação

o:	[1,0,0,0,0,0]
brasil:	[0,1,0,0,0,0]
sediou:	[0,0,1,0,0,0]
perdeu:	[0,0,0,1,0,0]
a:	[0,0,0,0,1,0]
Copa:	[0,0,0,0,0,1]



Palavras similares podem apresentar representações muito diferentes

D_1 : "o brasil sediou a copa"
= {[1,0,0,0,0,0],[0,1,0,0,0,0], [0,0,1,0,0,0], [0,0,0,0,1,0], [0,0,0,0,0,1]}

D_2 : "o brasil perdeu a copa"
= [1,0,0,0,0,0],[0,1,0,0,0,0], [0,0,0,1,0,0], [0,0,0,0,1,0], [0,0,0,0,0,1]

Representação Baseada em Contagem

Bag-of-Words

Bag of Words

- Cada documento é representado por um vetor
- Dimensionalidade:
  - Tamanho do vocabulário
  - Cada palavra vai ter um peso (contagem)
- Simples e efetivo
- Não guarda a ordem
- Considera apenas o aspecto léxico das unidades linguísticas
- Não modela similaridade semântica

D_1 : "the cat sat on the hat"

D_2 : "the dog ate the cat and the hat"

V	[the, cat, sat, on, hat, dog, ate, and]
D_1	[2, 1, 1, 1, 1, 0, 0, 0]
D_2	[3, 1, 0, 0, 1, 1, 1, 1]

Representação Baseada em Pesos

TF-IDF

TF-IDF

- É um vetor de pesos, similar ao Bag of Words
- Medida estatística para mensurar a importância de uma palavra em um documento;
- **Term Frequency** (a frequência do termo): mede a frequência com que um termo ocorre num documento;
- **Inverse Document Frequency** (inverso da frequência nos documentos): Quantas vezes o termo aparece em outros documentos. Mede o quão importante um termo é no contexto de todos os documentos.

-> Quanto mais frequente uma palavra é em seu documento, mais importante ela tende a ser

TF-IDF

- É um vetor de pesos, similar ao Bag of Words
- Medida estatística para mensurar a importância de uma palavra em um documento;
- **Term Frequency** (a frequência do termo): mede a frequência com que um termo ocorre num documento;
- **Inverse Document Frequency** (inverso da frequência nos documentos): Quantas vezes o termo aparece em outros documentos. Mede o quão importante um termo é no contexto de todos os documentos.

-> Quanto mais frequente uma palavra é em seu documento, mais importante ela tende a ser

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

tf_{ij} = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

TF-IDF

- É um vetor de pesos, similar ao Bag of Words
- Medida estatística para mensurar a importância de uma palavra em um documento;
- **Term Frequency** (a frequência do termo): mede a frequência com que um termo ocorre num documento;
- **Inverse Document Frequency** (inverso da frequência nos documentos): Quantas vezes o termo aparece em outros documentos. Mede o quão importante um termo é no contexto de todos os documentos.

-> Quanto mais frequente uma palavra é em seu documento, mais importante ela tende a ser

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

tf_{ij} = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

```
doc1 = "Doctor Who é uma série de TV maravilhosa! Já assisti todos os episódios."
doc2 = "Doctor Who é a melhor série de TV!"
doc3 = "Doctor Who é muito boa."

corpus = [doc1, doc2, doc3]
```

TF-IDF

- É um vetor de pesos, similar ao Bag of Words
- Medida estatística para mensurar a importância de uma palavra em um documento;
- **Term Frequency** (a frequência do termo): mede a frequência com que um termo ocorre num documento;
- **Inverse Document Frequency** (inverso da frequência nos documentos): Quantas vezes o termo aparece em outros documentos. Mede o quão importante um termo é no contexto de todos os documentos.

-> Quanto mais frequente uma palavra é em seu documento, mais importante ela tende a ser

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

```
doc1 = "Doctor Who é uma série de TV maravilhosa! Já assisti todos os episódios."
doc2 = "Doctor Who é a melhor série de TV!"
doc3 = "Doctor Who é muito boa."

corpus = [doc1, doc2, doc3]
```

	assisti	boa	de	doctor	episódios	já	maravilhosa	melhor	muito	os	série	todos	tv	uma	who
0	0.325596	0.000000	0.247624	0.192302	0.325596	0.325596	0.325596	0.000000	0.000000	0.325596	0.247624	0.325596	0.247624	0.325596	0.192302
1	0.000000	0.000000	0.410475	0.318770	0.000000	0.000000	0.000000	0.539725	0.000000	0.000000	0.410475	0.000000	0.410475	0.000000	0.318770
2	0.000000	0.608845	0.000000	0.359594	0.000000	0.000000	0.000000	0.000000	0.608845	0.000000	0.000000	0.000000	0.000000	0.000000	0.359594

TF-IDF

- É um vetor de pesos, similar ao Bag of Words
- Medida estatística para mensurar a importância de uma palavra em um documento;
- **Term Frequency** (a frequência do termo): mede a frequência com que um termo ocorre num documento;
- **Inverse Document Frequency** (inverso da frequência nos documentos): Quantas vezes o termo aparece em outros documentos. Mede o quão importante um termo é no contexto de todos os documentos.

-> Quanto mais frequente uma palavra é em seu documento, mais importante ela tende a ser

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

tf_{ij} = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

```
doc1 = "Doctor Who é uma série de TV maravilhosa! Já assisti todos os episódios."
doc2 = "Doctor Who é a melhor série de TV!"
doc3 = "Doctor Who é muito boa."

corpus = [doc1, doc2, doc3]
```

	assisti	boa	de	doctor	episódios	já	maravilhosa	melhor	muito	os	série	todos	tv	uma	who
0	0.325596	0.000000	0.247624	0.192302	0.325596	0.325596	0.325596	0.000000	0.000000	0.325596	0.247624	0.325596	0.247624	0.325596	0.192302
1	0.000000	0.000000	0.410475	0.318770	0.000000	0.000000	0.000000	0.539725	0.000000	0.000000	0.410475	0.000000	0.410475	0.000000	0.318770
2	0.000000	0.608845	0.000000	0.359594	0.000000	0.000000	0.000000	0.000000	0.608845	0.000000	0.000000	0.000000	0.000000	0.000000	0.359594

TF-IDF

- É um vetor de pesos, similar ao Bag of Words
- Medida estatística para mensurar a importância de uma palavra em um documento;
- **Term Frequency** (a frequência do termo): mede a frequência com que um termo ocorre num documento;
- **Inverse Document Frequency** (inverso da frequência nos documentos): Quantas vezes o termo aparece em outros documentos. Mede o quão importante um termo é no contexto de todos os documentos.

-> Quanto mais frequente uma palavra é em seu documento, mais importante ela tende a ser

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

tf_{ij} = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

```
doc1 = "Doctor Who é uma série de TV maravilhosa! Já assisti todos os episódios."
doc2 = "Doctor Who é a melhor série de TV!"
doc3 = "Doctor Who é muito boa."

corpus = [doc1, doc2, doc3]
```

	assisti	boa	de	doctor	episódios	já	maravilhosa	melhor	muito	os	série	todos	tv	uma	who
0	0.325596	0.000000	0.247624	0.192302	0.325596	0.325596	0.325596	0.000000	0.000000	0.325596	0.247624	0.325596	0.247624	0.325596	0.192302
1	0.000000	0.000000	0.410475	0.318770	0.000000	0.000000	0.000000	0.539725	0.000000	0.000000	0.410475	0.000000	0.410475	0.000000	0.318770
2	0.000000	0.608845	0.000000	0.359594	0.000000	0.000000	0.000000	0.000000	0.608845	0.000000	0.000000	0.000000	0.000000	0.000000	0.359594

Aprendizagem de
Representação de Palavras

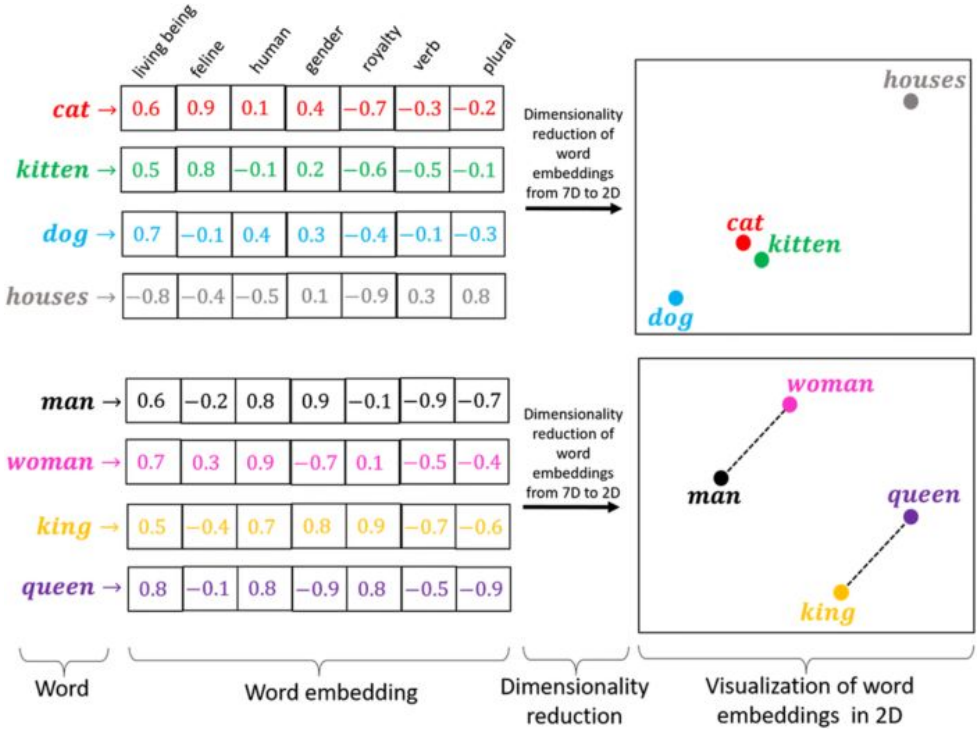
Word Embeddings

Aprendizagem de Representação de Palavras: Word Embeddings

- É um conjunto de técnicas que aprendem a melhor representação de palavras a partir de dados brutos.
- Motivado:
  - pela menor dependência na engenharia de características
  - Motivado pelo desenvolvimento dos modelos de ML e DL
  - Motivado pela busca por redução de dimensionalidade
- Representam qualquer unidade linguística como um vetor denso:
  - um caractere, uma palavra, uma sentença ou documento

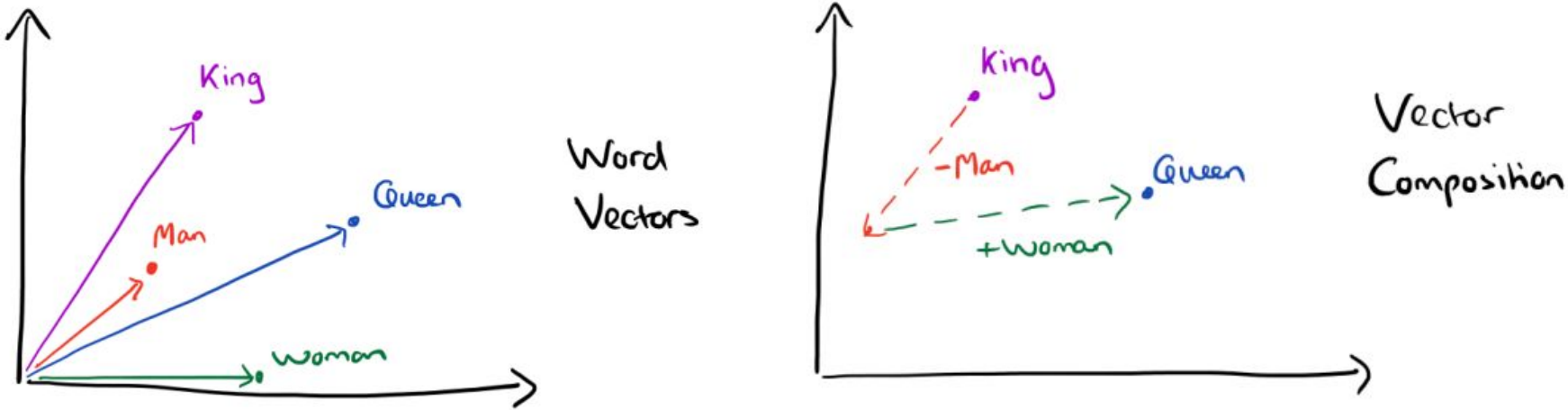
Aprendizagem de Representação de Palavras: Word Embeddings

- O contexto de uma palavra pode ser mapeado em um vetor de baixa dimensionalidade
- Os modelos mais populares são aqueles baseados em aprendizagem utilizando redes neurais profundas
- Mapeia significado semântico dessas unidades linguísticas



Fonte: https://medium.com/@hari4om/word-embedding-d816f643140

Composição de Vetores: **King – Man + Woman = ?**



FONTE: https://www.kdnuggets.com/2016/05/amazing-power-word-vectors.html

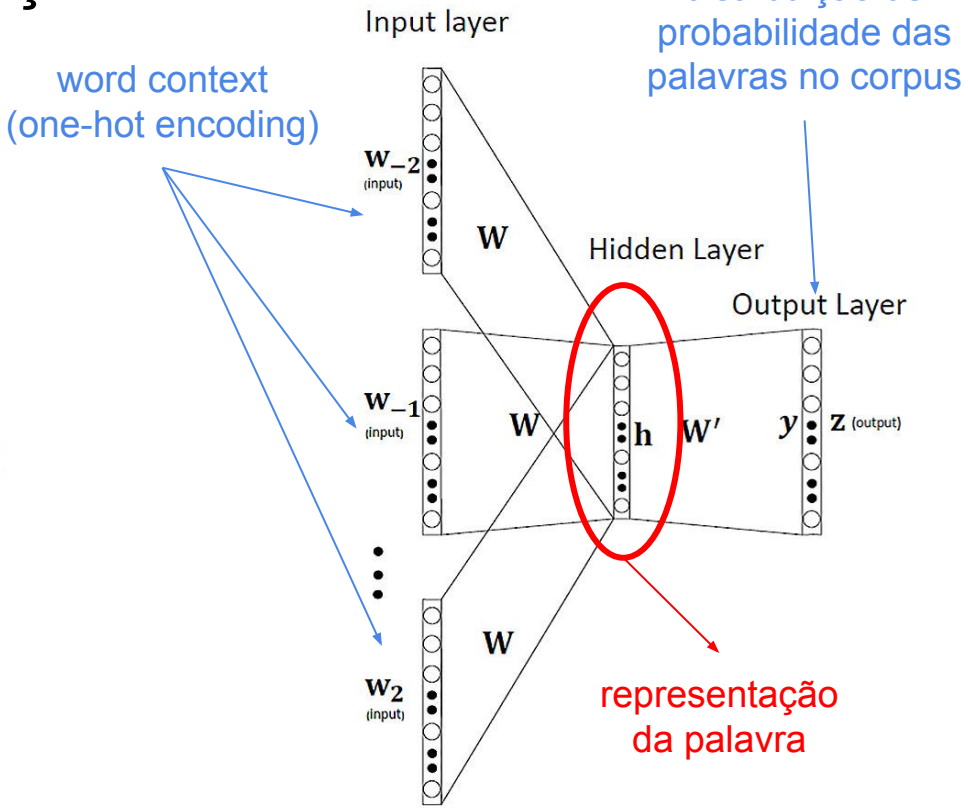
Word2Vec

CBOW e Skip-gram

Word2Vec
Modelos Baseados em Predição: CBOW

Prevê a representação de uma palavra, dado o seu contexto (palavras que aparecem antes e depois)

Fonte da Imagem
https://thinkinfi.com/continuous-bag-of-words-cbow-multi-word-model-how-it-works/

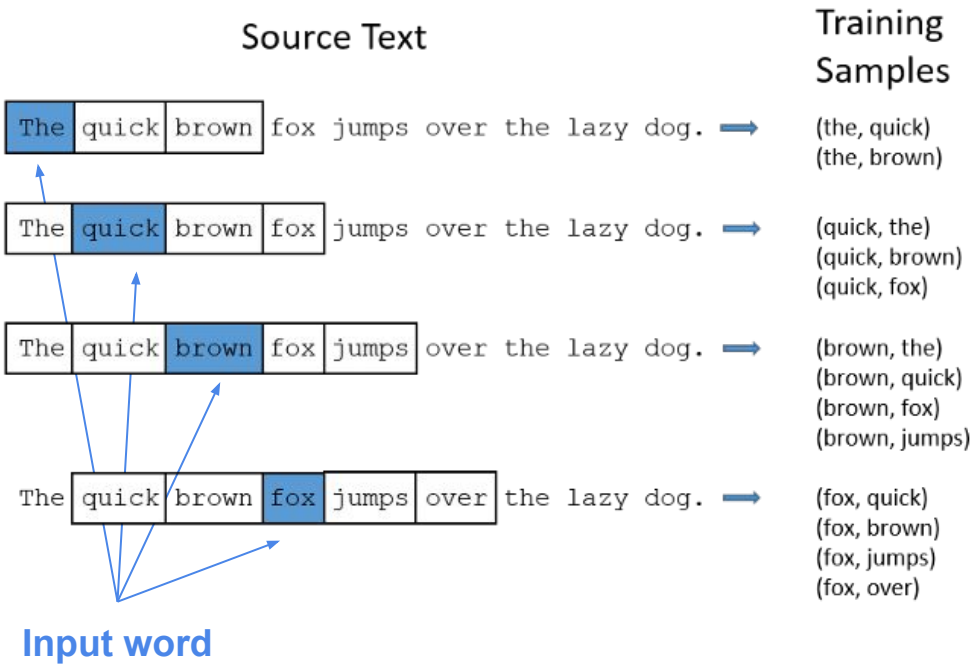


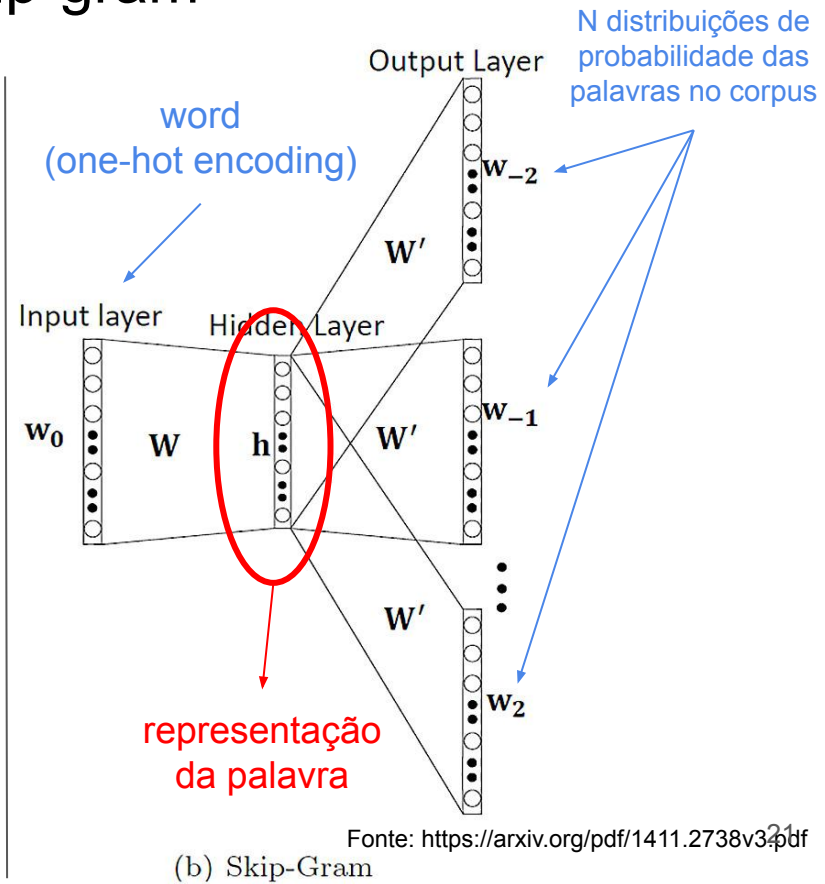
Fonte: https://arxiv.org/pdf/1411.2738v3.pdf

(a) CBOW

Word2Vec
Modelos Baseados em Predição: Skip-gram

Dada uma palavra como entrada, prevê o seu contexto.





Fonte: https://arxiv.org/pdf/1411.2738v3.pdf

Embeddings

- São amplamente utilizados
- Boas representações tendem a apresentar melhores resultados em modelos de Deep Learning
- Muitos modelos de representação pré-treinados estão disponíveis
- Prefira utilizar os modelos de representação já disponíveis

Modelos amplamente utilizados

Word2Vec
https://code.google.com/p/word2vec/

Glove
http://nlp.stanford.edu/projects/glove/

FastText
https://fasttext.cc/

Doc2Vec
https://radimrehurek.com/gensim/models/doc2vec.html

Aula Prática

Google Colab

Resumo da Aula

- Representação Textual Categórica: one-hot encoding
- Representação Textual baseada em Contagem: bag-of-words
- Representação baseada em Pesos: TF-IDF
- Representação baseada em Aprendizagem: Word2Vec
- Prática:
  - Classificação de Textos em Grupos de Notícias (fetch_20newsgroups)
  - Classificação de Sentimento - predição da contagem de estrelas de uma review de produtos (Amazon Review Corpus - Fashion)

Exercício Proposto

Google Colab

REFERÊNCIAS

JURAFSKY, Daniel; MARTIN, James H. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. https://web.stanford.edu/~jurafsky/slp3/.

MIKOLOV, Tomas et al. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

PENNINGTON, Jeffrey; SOCHER, Richard; MANNING, Christopher D. Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014. p. 1532-1543.