



**Data Science
Academy**

www.datascienceacademy.com.br

Processamento de Linguagem Natural

Matemática do Word2vec

Vamos criar um exemplo simples, para você compreender a Matemática por trás do Word2vec. Vou pegar um conjunto de dados bem pequeno como nosso corpus. Veja as frases do nosso pequeno corpus:

- O cachorro viu um gato
- O cachorro perseguiu um gato
- O gato escalou uma árvore

As três sentenças anteriores têm oito (8) palavras únicas. Para acessá-las, nos referiremos ao índice de cada palavra. Observe a tabela a seguir:

Palavra	Índice
um	1
gato	2
perseguiu	3
escalou	4
cachorro	5
viu	6
O	7
árvore	8

Aqui, nosso valor para V é igual a 8. Em nossa rede neural, precisamos de oito neurônios de entrada e oito neurônios de saída. Agora vamos supor que teremos três (3) neurônios na camada oculta. Portanto, neste caso, nossos valores WI e WO são definidos da seguinte forma:

$$WI = [V * N] = [8 * 3]$$

$$WO = [N * V] = [3 * 8]$$

Antes do início do treinamento, essas matrizes, WI e WO, são inicializadas usando pequenos valores aleatórios, como é muito comum no treinamento de redes neurais. Apenas para fins de ilustração, vamos supor que WI e WO sejam inicializados com os seguintes valores:

WI

-0.094491	-0.443977	0.313917
-0.490796	-0.229903	0.065460
0.072921	0.172246	-0.357751
0.104514	-0.463000	0.079367
-0.226080	-0.154659	-0.038422
0.406115	-0.192794	-0.441992
0.181755	0.088268	0.277574
-0.055334	0.491792	0.263102

WO

0.023074	0.479901	0.432148	0.375480	-0.364732	-0.119840	0.266070	-0.351000
-0.368008	0.424778	-0.257104	-0.148817	0.033922	0.353874	-0.144942	0.130904
0.422434	0.364503	0.467865	-0.020302	-0.423890	-0.438777	0.268529	-0.446787

Estamos direcionando para que nossa rede neural possa aprender a relação entre as palavras gato e escalou. Então, em outras palavras, podemos explicar que a rede neural deve dar alta probabilidade para a palavra escalou quando a palavra gato é introduzida na rede neural como uma entrada. Assim, na word embedding, a palavra gato é referida como uma palavra de contexto e a palavra escalou é referida como uma palavra de destino. O vetor de entrada X que representa a palavra gato e será $[0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0]$ t. Observe que apenas o segundo componente do vetor é 1. A razão por trás disso é que a palavra de entrada gato, mantém a segunda posição em uma lista classificada do corpus. Da mesma forma, a palavra alvo é escalou e o vetor alvo será semelhante a $[0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0]$ t.

A entrada para a primeira camada é $[0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0]$ t. A saída da camada oculta H_t é calculada usando a seguinte fórmula:

$$H_t = X_t W_I = [-0,490796 \ -0,229903 \ 0,065460]$$

A partir do cálculo anterior, podemos descobrir que, aqui, a saída dos neurônios ocultos imita os pesos da segunda linha da matriz W_I por causa da representação one-hot encoding. Agora precisamos verificar um cálculo semelhante para a camada oculta e a camada de saída. O cálculo para a camada oculta e a camada de saída é definido da seguinte maneira:

$$H_t W_O = [0.100934 \ -0.309331 \ -0.122361 \ -0.151399 \ 0.143463 \ -0.051262 \ -0.079686 \ 0.112928]$$

Aqui, nosso objetivo final é obter probabilidades de palavras na camada de saída. A partir da camada de saída, estamos gerando uma probabilidade que reflete a próxima relação de palavras com a palavra de contexto na entrada. Então, a representação matemática é dada da seguinte forma:

$$\text{Probabilidade (word}_k \mid \text{wordcontext)} \text{ para } k = 1 \dots V$$

Aqui, estamos falando em termos de probabilidade, mas nossa saída está na forma de um conjunto de vetores, então precisamos converter nossa saída em probabilidade. Precisamos tomar cuidado para que a soma da saída do neurônio da camada de saída final seja adicionada a um. Em Word2vec, estamos convertendo valores de ativação dos neurônios da camada de saída em probabilidades usando a função softmax.