



**Data Science  
Academy**

[www.datascienceacademy.com.br](http://www.datascienceacademy.com.br)

Processamento de Linguagem Natural

Função Softmax

Nesta aula, veremos a função softmax. A função softmax é usada para converter vetores de saída em uma forma de probabilidade. Estamos usando a função softmax porque queremos converter a saída da última camada em termos de probabilidade e a função softmax pode facilmente converter valores vetoriais em valores de probabilidade. Aqui, a saída do k-neurônio será calculada pela seguinte equação, onde a ativação (n) representa o valor de ativação do enésimo neurônio da camada de saída:

$$y_k = \Pr(\text{word}_k | \text{word}_{\text{context}}) = \frac{\exp(\text{activation}(k))}{\sum_{n=1}^V \exp(\text{activation}(n))}$$

Usando esta equação, podemos calcular as probabilidades para oito palavras no corpus definido na aula anterior e os valores de probabilidade são dados como segue:

[0.143073 0.094925 0.114441 0.111166 0.149289 0.122874 0.119431 0.144800]

Como você pode ver, a probabilidade 0.111166 é para a palavra-alvo escolhida nesse caso escalou (as probabilidades são calculadas em seu código Python com o uso de funções para Softmax). Como sabemos, o vetor alvo é [0 0 0 1 0 0 0 0] t, então podemos calcular o erro da predição. Para gerar um erro de previsão ou vetor de erro, precisamos subtrair o vetor de probabilidade do vetor de destino e, uma vez que sabemos o vetor de erro ou os valores de erro, podemos ajustar o peso de acordo com isso. Aqui, precisamos ajustar os valores de peso das matrizes WI e WO.

A técnica de propagação de erros na rede e reajuste dos valores de peso de WI e WO é chamada de retropropagação (Backpropagation). Assim, o treinamento pode continuar tomando diferentes pares de palavras de contexto-alvo do corpus. É assim que o Word2vec aprende as relações entre as palavras para desenvolver uma representação vetorial das palavras no corpus.

Simples, não? 😊