



**Data Science  
Academy**

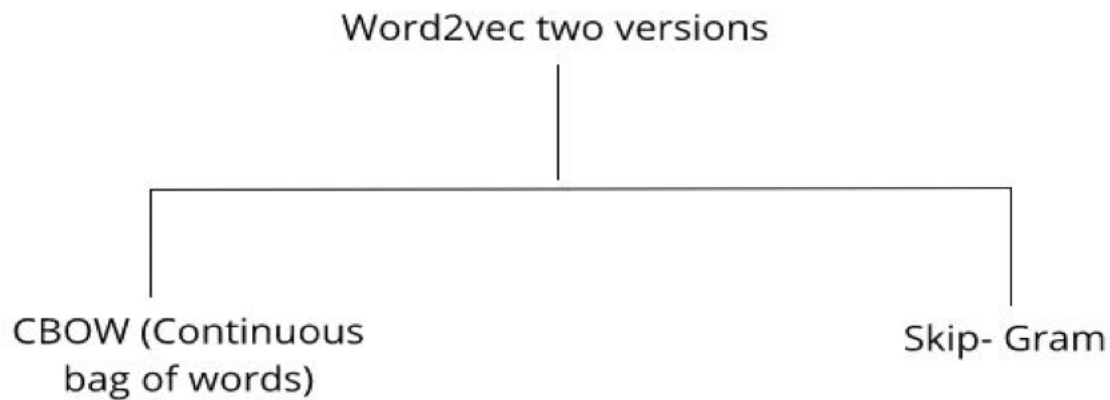
[www.datascienceacademy.com.br](http://www.datascienceacademy.com.br)

Processamento de Linguagem Natural

Revisando a Arquitetura Word2vec

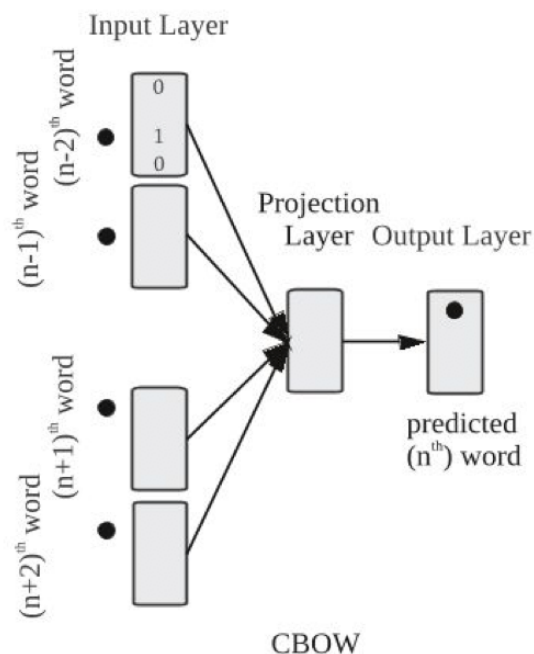


O modelo Word2vec é formado por 2 algoritmos, conforme imagem abaixo:



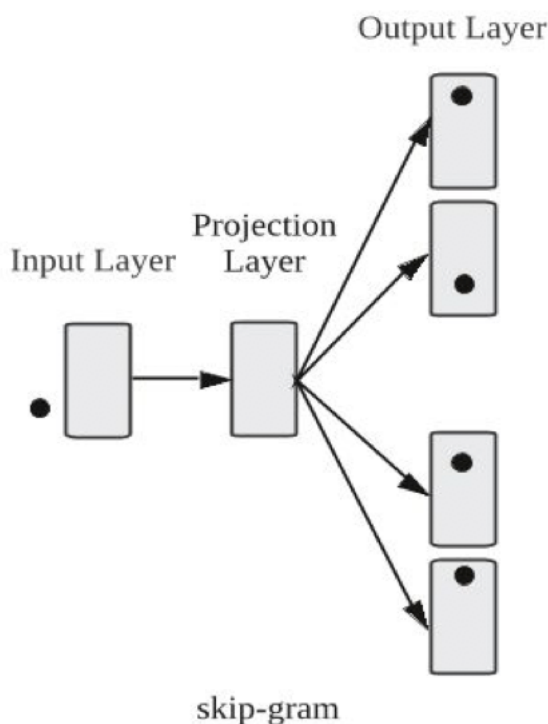
## CBOW (Continuous Bag of Words)

No algoritmo CBOW, o contexto é representado por várias palavras para determinadas palavras-alvo. Basta relembrar o nosso exemplo nas aulas anteriores, em que nossa palavra de contexto era *gato* e nossa palavra alvo era *escalou*. Por exemplo, podemos usar *gato* e *árvore* como palavras de contexto para prever a palavra *escalou* como a palavra alvo. Neste caso, precisamos mudar a arquitetura da rede neural, especialmente a camada de entrada. Se você aumentar as palavras de contexto, precisará colocar uma camada de entrada adicional para representar cada uma das palavras, e todas essas camadas de entrada serão conectadas à camada oculta, conforme imagem abaixo:



## Skip-gram

O modelo Skip-gram (SG) inverte o uso de palavras de destino e palavras de contexto. Aqui, a palavra de destino é dada como entrada para a camada de entrada na forma de um vetor one-hot encoded. A camada oculta permanece a mesma. A camada de saída da rede neural é repetida várias vezes para gerar o número escolhido de palavras de contexto. Vamos dar um exemplo das palavras gato e árvore como palavras de contexto e a palavra escalou como palavra alvo. O vetor de entrada no modelo Skip-gram será o vetor one-hot encoded da palavra escalou  $[0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]$  e nossos vetores de saída devem ser vetores para a palavra gato e árvore. Portanto, o vetor de saída deve ser  $[0\ 1\ 0\ 0\ 0\ 0\ 0\ 0]$  para gato e  $[0\ 0\ 0\ 0\ 0\ 0\ 0\ 1]$  para árvore. O esquema é representado na imagem abaixo:



Desta vez, a saída não será um único vetor de probabilidade, mas dois vetores diferentes de probabilidade, já que temos duas palavras como a palavra de contexto. A pequena mudança no Skip-gram é que os vetores de erro de todas as camadas de saída são somados para ajustar os pesos via backpropagation.

Referências:

<https://www.semanticscholar.org/>