

Machine Learning

Machine Learning

Estudo de Caso - Inteligência Artificial Para Previsão de Sentenças em Embargos de Declaração

Neste estudo de caso vamos trabalhar com mais um modelo de PLN usando o PyTorch como nosso framework. Vamos receber como entrada um Embargo de Declaração (texto), preparar os dados, construir e treinar um modelo de rede neural artificial profunda (Deep Learning) e fazer previsões de sentenças com o modelo treinado. [O Jupyter Notebook completo você encontra ao final do capítulo.](#) Mais a frente no curso teremos um capítulo sobre Deep Learning e o objetivo aqui é mostrar a você exemplos de aplicações de PLN. De qualquer modo, o Deep Learning Book trata de diversos conceitos que podem ajudar na compreensão do Estudo de Caso, se considerar necessário: <http://www.deeplearningbook.com.br>

O que são embargos de declaração?

Os embargos de declaração são um instrumento jurídico por meio do qual uma das partes pode pedir esclarecimentos ao juiz ou tribunal sobre a decisão judicial proferida. Também conhecidos como embargos declaratórios, por meio deles é possível resolver dúvidas causadas por contradições ou obscuridades. Do mesmo modo, pode-se suprir omissões ou, ainda, apontar erros materiais.

Entretanto, merece esclarecimento uma das principais dúvidas sobre os embargos de declaração. Afinal, eles são ou não uma forma de recurso? Segundo o Novo Código de Processo Civil, sim, uma vez que estão incluídos no rol de recursos no Novo CPC, em seu art. 994. Apesar disso, esta não é uma inovação. De fato, os embargos de declaração já estavam incluídos entre os recursos desde o Código anterior, em seu art. 496, inciso IV. Aqui você encontra mais detalhes:

<https://blog.sajadv.com.br/embargos-de-declaracao-novo-cpc/>

E como vamos criar nossa solução de IA? Usando um modelo CBoW. Vamos compreender alguns conceitos e definir o que é um modelo CBoW (é uma rede neural artificial). Acompanhe a leitura do texto abaixo.

Word Embedding

A Word Embedding é uma das representações mais populares do vocabulário de um documento. É capaz de capturar o contexto de uma palavra em um documento, semelhança semântica e sintática, relação com outras palavras, etc.

Em termos gerais, as Word Embeddings são representações vetoriais de uma palavra específica. Word Embedding é apenas uma maneira elegante de dizer “representação numérica de palavras”. Uma boa analogia seria como usamos a representação RGB para cores.

Por que precisamos de Word Embeddings?

Como humano, intuitivamente falando, não faz muito sentido querer representar palavras ou qualquer outro objeto no universo usando números porque os números são usados para quantificação e por que seria necessário quantificar palavras?

Quando na ciência, dizemos que a velocidade do meu carro é 50 km/h, percebemos o quão rápido / lento estamos dirigindo. Se dissermos que meu amigo está dirigindo a 70 km/h, podemos comparar qual de nós está indo mais rápido. Além disso, podemos calcular onde estaremos em um determinado momento, quando chegaremos ao nosso destino, pois sabemos a distância de nossa jornada, etc.

Da mesma forma, fora da ciência, usamos números para quantificar uma qualidade; quando citamos o preço de um objeto, tentamos quantificar seu valor, o tamanho de uma peça de roupa, tentamos quantificar as proporções corporais que se ajustam melhor.

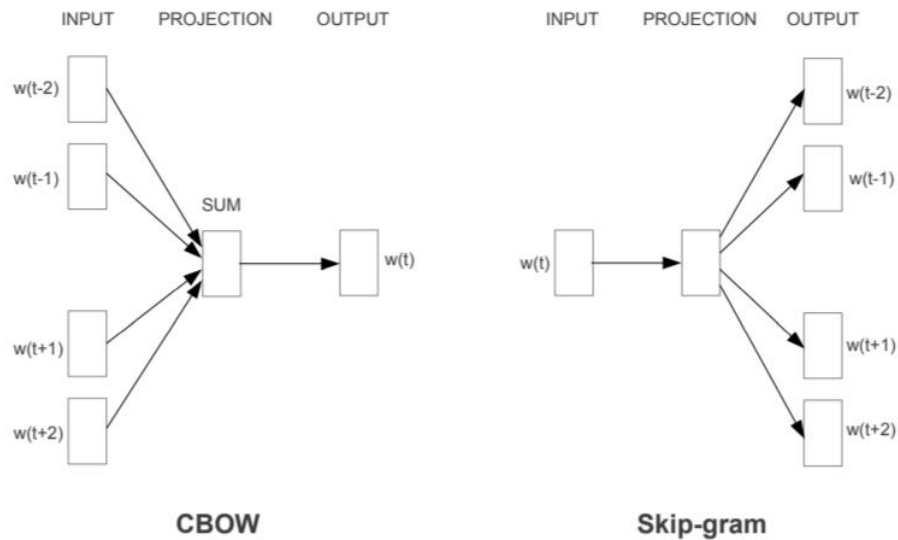
Todas essas representações fazem sentido porque, usando números, facilitamos muito a análise e a comparação com base nessas qualidades. Quanto vale um sapato ou uma bolsa? Bem, por mais diferentes que sejam esses dois objetos, uma maneira de responder é comparar os preços. Além do aspecto da quantificação, não há mais nada a ganhar com essa representação.

Agora que sabemos que a representação numérica de objetos ajuda na análise ao quantificar uma certa qualidade, a questão é que qualidade de palavras queremos quantificar?

A resposta para isso é que queremos quantificar a semântica. Queremos representar as palavras de tal maneira que capte seu significado da maneira que os humanos fazem. Não é o significado exato da palavra, mas contextual.

Continuous Bag of Words Model (CBOW) e Skip-gram

Ambas são arquiteturas para aprender as representações de palavras usando redes neurais, atualmente uma das principais técnicas de IA.



No modelo CBOW, as representações distribuídas de contexto (ou palavras circundantes) são combinadas para prever a palavra no meio. Enquanto no modelo Skip-gram, a representação distribuída da palavra de entrada é usada para prever o contexto.

Um pré-requisito para qualquer rede neural ou qualquer técnica de treinamento supervisionado é ter dados de treinamento rotulados. Como você treina uma rede neural para prever a Embedding de palavras quando não possui dados rotulados, como palavras e a Embedding de palavras correspondente? É onde entra o modelo Skip-gram.

Faremos isso criando uma tarefa "falsa" para a rede neural treinar. Não estaremos interessados nas entradas e saídas dessa rede, mas o objetivo é apenas aprender os pesos da camada oculta que são os "vetores de palavras" que estamos tentando aprender.

A tarefa "falsa" do modelo Skip-gram seria dada uma palavra, tentaremos prever as palavras vizinhas. Definiremos uma palavra vizinha pelo tamanho da janela - um hiperparâmetro. Essa imagem abaixo demonstra como poderíamos pré-processar os dados de texto para treinar o modelo:



Source Text	Training Samples generated from source text			
I will have orange juice and eggs for breakfast	(will, I)	(will, have)	(will, orange)	
I will have orange juice and eggs for breakfast	(have, I)	(have, will)	(have, orange)	(have, juice)
I will have orange juice and eggs for breakfast	(orange, will)	(orange, have)	(orange, juice)	(orange, and)
I will have orange juice and eggs for breakfast	(juice, have)	(juice, orange)	(juice, and)	(juice, eggs)
I will have orange juice and eggs for breakfast	(and, orange)	(and, juice)	(and, eggs)	(and, for)
I will have orange juice and eggs for breakfast	(eggs, juice)	(eggs, and)	(eggs, for)	(eggs, breakfast)
I will have orange juice and eggs for breakfast	(for, and)	(for, eggs)	(for, breakfast)	

As dimensões do vetor de entrada serão $1 \times V$ - onde V é o número de palavras no vocabulário - ou seja, uma representação One-Hot Encoding da palavra. A única camada oculta terá a dimensão $V \times E$, onde E é o tamanho da word embedding e é um hiperparâmetro. A saída da camada oculta seria de dimensão $1 \times E$, que alimentaremos em uma camada softmax. As dimensões da camada de saída serão de $1 \times V$, onde cada valor no vetor será a pontuação de probabilidade da palavra-alvo nessa posição. O aprendizado é feito com o algoritmo Backpropagation.

Esse é um tema avançado e por isso é estudado em detalhes na Formação Inteligência Artificial e não na Formação Cientista de Dados.

Mas no Jupyter Notebook “**06-DSA-Cap12-CBOW.ipynb**” você encontra um exemplo completo, comentado linha a linha.

Execute o Jupyter Notebook, leia os comentários e tenha uma visão geral de como funciona esse modelo de PLN.

Boa aula.