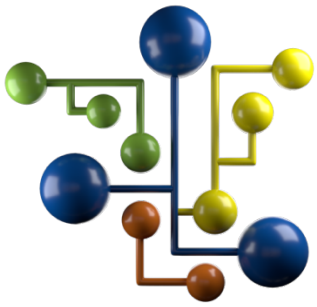


Data Science Academy

www.datascienceacademy.com.br

Machine Learning

Estudo de Caso - Tradutor de Idioma com Machine Learning e PLN

O Modelo Seq2seq usado neste Estudo de Caso é um modelo avançado e não espere aprender tudo sobre ele em um Estudo de Caso. Não é esse nosso objetivo aqui. O Modelo Seq2Seq é estudado em 3 capítulos no curso de Processamento de Linguagem Natural na DSA e nosso objetivo com este Estudo de Caso é mostrar a você uma das muitas aplicações de Inteligência Artificial na atualidade. Especialização em IA é um caminho natural para quem estuda Machine Learning.

O Seq2seq foi introduzido pela primeira vez para tradução automática, pelo Google. Antes disso, a tradução funcionava de maneira muito ingênua. Cada palavra que você costumava digitar era convertida para o idioma de destino, sem considerar a gramática e a estrutura da frase. O Seq2seq revolucionou o processo de tradução, utilizando o aprendizado profundo (Deep Learning). Ele não apenas leva em consideração a palavra / entrada atual durante a tradução, mas também sua vizinhança.

Atualmente, é usado para uma variedade de aplicações diferentes, como legendas de imagens, modelos de conversação, resumo de texto, tradução, etc.

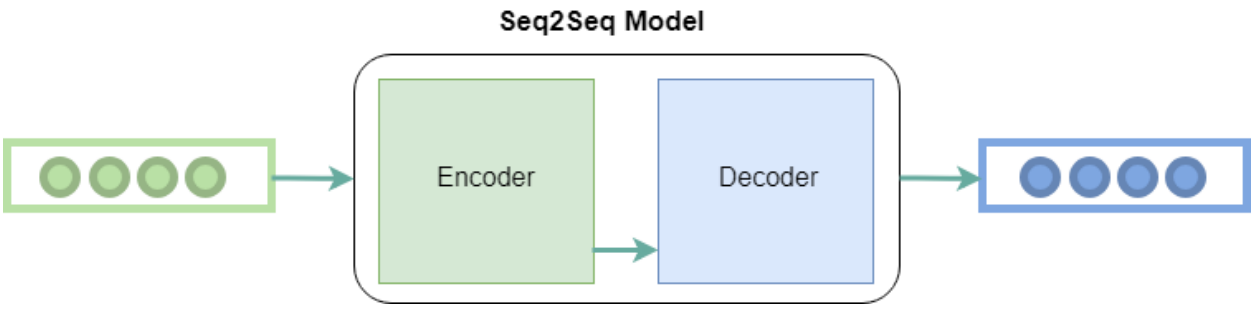
Modelo Seq2seq

Como o nome sugere, seq2seq usa como entrada uma sequência de palavras (sentença ou sentenças) e gera uma sequência de saída de palavras. Faz isso usando a rede neural recorrente (RNN), sendo comum usarmos versões avançadas da RNN, ou seja, LSTM ou GRU (estudadas no curso Deep Learning II). Isso ocorre porque a RNN sofre com o problema da dissipação do gradiente. O modelo LSTM é usado na versão proposta pelo Google. Ele desenvolve o contexto da palavra, recebendo 2 entradas em cada ponto do tempo. Um atual e outro da saída anterior, daí o nome recorrente (a saída entra como entrada).

O Seq2seq possui principalmente dois componentes: codificador e decodificador, e, portanto, às vezes é chamado de Rede Codificador-Decodificador.

Codificador: Utiliza camadas de rede neural profunda e converte as palavras de entrada em vetores ocultos correspondentes. Cada vetor representa a palavra atual e o contexto da palavra.

Decodificador: É semelhante ao codificador. Toma como entrada o vetor oculto gerado pelo codificador, seus próprios estados ocultos e a palavra atual para produzir o próximo vetor oculto e finalmente prever a próxima palavra.



Além desses dois elementos, muitas otimizações levaram a outros componentes do seq2seq:

Attention: A entrada para o decodificador é um único vetor que deve armazenar todas as informações sobre o contexto. Isso se torna um problema com grandes sequências. Portanto, o mecanismo de atenção é aplicado, permitindo que o decodificador observe a sequência de entrada seletivamente.

Beam Search: A palavra com maior probabilidade é selecionada como saída pelo decodificador. Mas isso nem sempre produz os melhores resultados, devido ao problema básico dos algoritmos gananciosos. Portanto, a pesquisa por feixe é aplicada, o que sugere possíveis traduções em cada etapa. Isso é feito criando uma árvore dos melhores resultados.

Bucketing: Sequências de comprimento variável são possíveis em um modelo seq2seq, devido ao preenchimento de 0, que é feito na entrada e na saída. No entanto, se o comprimento máximo definido por nós for 100 e a sentença tiver apenas 3 palavras, isso causará enorme desperdício de espaço. Então, usamos o conceito de **Bucketing**. Criamos variáveis de tamanhos diferentes, como (4, 8) (8, 15) e assim por diante, onde 4 é o comprimento máximo de entrada definido por nós e 8 é o comprimento máximo de saída definido.

Seq2seq é um dos modelos mais avançados para PLN e vamos usá-lo para construir um tradutor de idioma.

Isso é o que faremos neste Estudo de Caso, que você encontra no Jupyter Notebook "**09-DSA-Cap12-Seq2seq**". Leia os comentários com atenção, execute as células, modifique os exemplos e compreenda o funcionamento do Seq2seq.

Boa aula.