Departamento de Processamento de Energia Elétrica

Disciplina de *Deep Learning*
Exercício sobre Word2Vec

Prof. Rodrigo da Silva Guerra

15 de junho de 2020

Introdução

Para este exercício, você tomará como ponto de partida o código disponível na URL shorturl.at/bmrL4. O objetivo deste exercício é que você utilize word2vec para construir um sistema de recomendação de produtos similares para um site de compras online.

Exercício

Na URL shorturl.at/bmrL4 apresento um código que lê uma base de dados com histórico de compras de um site que vende produtos pela internet. Nesta base de dados constam registros de diversas compras, com identificação anônima de usuário e com os produtos comprados em cada transação.

Tomando como ponto de partida este código, você deve treinar um modelo word2vec usando o histórico de compras de cada cliente como "frases", onde cada produto comprado representa uma "palavra". Nessa analogia, as frases são as listas de códigos dos produtos comprados em cada compra, sendo cada código de produto equivalente a uma palavra diferente. A teoria por trás deste exercício é que quando os clientes fazem uma compra, existe uma relação semântica entre os produtos selecionados naquela compra. Por exemplo, o cliente pode estar comprando material escolar, ou produtos para festa infantil, ou ainda artigos de vestuário. Desta forma o modelo

word2vec deve, em alguma medida, gerar um *embedding* que captura essas relações semânticas.

O código fornecido como exemplo já separa 90% dos clientes para treinamento e 10% para validação. Isso significa que todas as compras destes 90% dos clientes de treinamento serão usadas para treinar o *embedding* usando word2vec. Note que um mesmo cliente pode ter feito mais de uma compra na base de dados (mais de uma frase por cliente).

Seu objetivo, portanto, é:

(1) Treinar um *embedding* word2vec usando os dados dos clientes da base de dados de treinamento;

(2) Criar uma função onde, fornecendo um código de produto, a função deve:
   a. Imprimir na tela a descrição do produto;
   b. Buscar os códigos dos produtos mais similares no *embedding*;
   c. Imprimir na tela uma lista com as descrições dos produtos considerados similares, e suas respectivas taxas de similaridade.

(3) Criar outra função onde, fornecendo uma lista completa de compras (lista de produtos comprados por um cliente), a função deve:
   a. Imprimir na tela uma lista com as descrições dos produtos comprados fornecidos à função como argumento
   b. Buscar os vetores de *embedding* de cada produto fornecido na lista, e calcular um vetor médio, como sendo a média desses vetores:

$$\bar{v} = \frac{1}{N}\sum_{i=1}^{N} v_i$$

   c. Usando o vetor médio resultante da conta acima, buscar os produtos considerados similares no *embedding*. Exclua das sugestões os produtos que já constam na lista original.
   (Dica: Para realizar uma busca usando um vetor como entrada, utilize o método `model.similar_by_vector(v)` do modelo word2vec do `gensim`. Note que talvez nem todos os produtos constem no "vocabulário" treinado. Você pode testar isso tentando verificar se o id do produto consta no vocabulário com código como "`if productid in model.wv:`".)
   d. Imprimir na tela uma lista com as descrições dos produtos sugeridos como similares, e suas respectivas taxas de similaridade

(4) Finalmente, usando a função criada no item (3) acima, faça testes de sugestões tomando como exemplo compras dos clientes do grupo de validação. Faça alguns testes e demonstre os resultados. Lembre que talvez nem todos produtos comprados pelos clientes no grupo de validação constem entre as compras dos clientes no grupo de treinamento. Demonstre o funcionamento do sistema de recomendação para pelo menos 5 compras diferentes do grupo de validação.