CS410 Tech Review - Embedding

YUL10@

Abstract

In this article, we briefly discussed the embedding technique in the field of NLP. It included the application of the idea of embedding in different topics. We will start from the general procedure of dealing with the sequence problem, word embedding, item embedding, and follow by entity embedding, graph embedding, and end with a specific application, which is using wording embedding to get the abstract of the text.

Summarization on Embedding

Section 1 The framework to deal with sequence problem

It is common to deal with sequence problems in the industry, such as text processing, web browser history, time series, and so on. In a sequence problem, the general framework includes the following steps: 1) raw processing 2) index 3) embedding (static/ dynamic) 4) following tasks. Among those steps, embedding is the most important one, which is critical to the model performance. There are many ways to do embedding, including word2vec, transformer, BERT, and ALBERT. In section 2, we will focus on the most foundational method – word2vec.

Section 2 Word Embedding - Word2Vec

The token cannot be used directly as the model input, so vectorizing the token is always a fundamental question. In the beginning, one-hot encoding is used to present tokens, but it is high-dimensional and sparse. Later, people came up with word embedding, with the method of word2vec. Google proposed word2vec first [1] and make the idea of word embedding popular in the industry. This encoding method is relatively low dimensional and dense.

There are two ways to generate the word embedding vector, the CBOW model and the skip-gram model. CBOW model tries to predict the target by its context, in other words, its objection function is p(w|context(w)); while the skip-gram model tries to predict the context by

the target, in other words, its objection function is p(context(w)|w). Note that the bottleneck of those algorithms is to update the look-up table W is the backprop. There are two methods to optimize the calculation, including Hierarchical Softmax and Negative Sampling.

Section 3 Item Embedding

According to the discussion around word2vec in section 2, it is easy to notice we can generalize this idea to more areas, which has the sequence data. Microsoft proposed the idea of item2vec in its paper: Item2Vec: Neural Item Embedding for Collaborative Filtering [2]. In this paper, they generalized the idea of Skip-Gram and Negative Sampling to Item-based collaborative filtering. Use the concurrence among items to replace the context in NLP.

Similarly, Airbnb also proposed an embedding method to capture the user's short/long-term interest. In the paper, real-time personalization using embedding for search ranking at Airbnb [3], they utilized the user's click session and booking session to obtain the embedding of the user and list. Then used the embedding info to boost their ranking model.

Section 4 Entity Embedding

People always need to deal with the categorical variable in traditional machine learning problems. Processing those categorical variables is also one of the most important tasks in feature engineering. There are usually two kinds of categorical variables, ordinal and nominal. The traditional method is using one-hot embedding. However, the one-hot embedding might be high-dimensional and sparse, moreover, it cannot reflect the underlying relationship between those entities. Here, we can use the embedding to compress the vectors to lower dimensions and concatenate the embedding vector with other dense features to improve the model performance. The embedding learned by one model can be used in another model as well. It can boost the model performance in structured data.

Section 5 Graph Embedding

The embedding can also be used on the graphs. The graph represents a two-dimensional relationship, while the sequence represents a one-dimensional relationship. so the general idea in graph embedding is to convert the graph to sequence first, and then convert the sequence to embedding.

There are two methods to do this conversion from a graph to a sequence. The first one is the deepwalk [4]: first using random walk to sample in the graph-based DFS and generate the

sequence, then using the skip-gram method to generate embedding. Note the Deepwalk is only used in the undirected unweighted graph. The second one is the LINE [5], which cannot be used in a directed/ undirected, weighted/ unweighted graph. This method can get a more balanced smoothing in the node embedding. Besides the two methods above, Stanford publish the node2vec in 2016 [6], which emphasizes both homophily and structure equivalence by BFS and DFS.

Conclusion

As this article indicated, the idea of embedding is quite flexible. It can not only be applied to word embedding, but also to items, entities, and graphs. The key idea is to find the information of an object from its context, utilizing the structure information in the corpus to enrich the vector of the object. This is a very useful technique and its application is far beyond the NLP area.

Reference

[1] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).

[2] Barkan, Oren, and Noam Koenigstein. "Item2vec: neural item embedding for collaborative filtering." *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2016.

[3] Grbovic, Mihajlo, and Haibin Cheng. "Real-time personalization using embeddings for search ranking at airbnb." *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018.

[4] Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations." *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014.

[5] Tang, Jian, et al. "Line: Large-scale information network embedding." *Proceedings of the 24th international conference on world wide web*. 2015.

[6] Grover, Aditya, and Jure Leskovec. "node2vec: Scalable feature learning for networks." *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 2016.