Deep Learning-based Online Alternative Product Recommendations at Scale

Amodh Shenoy (171IT108)
Sagar G (171IT136)
Akhil G (171IT106)

Introduction

- Recommender systems are pervasive in ecommerce and other web systems.
- Alternative product recommendation is an important way to help customers easily find the right products and speed up their buying decision process.
- There are two main ways to obtain an alternative product list for a given product.
- First is a content-based recommendation approach.
- The second way is to leverage customer behavior to find alternative products in the style of item-to-item collaborative filtering .

Motivation

Our motivation is to enable the recommendation of alternative products based solely on the product's features, without relying on historical purchase data. By doing so, we address the so-called cold start problem, which is often found in product recommendation approaches, and that may lead to profit loss in ecommerce sites.

Issues and Challenges

- Cold Start Problem
- Scalability of the approach
- Accuracy of the prediction
- Sparse ,Missing and Malicious data
- Structured recommendations
- Privacy

Literature Review

Authors	Methodology	Advantages	Limitations
Arthur,Abdul Hussein	Uses 2 or more recommender algorithms as viewpoint Rating prediction scenario for the user who didn’t rate purchased products	Provides an enriched user item matrix to recommend products to customers	Unlabeled data were used and that is converted as labeled one. The only explicit parameter is used. Web engagement measure and temporal behaviors are not considered.
Orit Raphaeli ,Da Costa	Generating the pairwise association rule based on the implicit social graph.	Sort out the items which consume less amount of time to reach the customer.	The timestamp is not considered a major.
Timur Osadchiy, Ivan Poliakov	Worked with a recommendation procedure based upon techniques like content-based and collaborative filtering.	May help the online retailers to engage the customer in mobile at any time and anywhere.	There is no recommendation for a pair of products w.r.t regions and locations. The cost of the algorithms has not been analyzed.

Outcome of Literature Survey

One of the most crucial issues, nowadays, is to provide personalized services to each individual based on their preferences. To achieve this goal, recommender system could be utilized as a tool to help the users in decision-making process offering different items and options. They are utilized to predict and recommend relevant items to end users.

Timur Osadchiy, Ivan Poliakov started their research work with a recommendation procedure based upon techniques like content-based and collaborative filtering. The application is taken to demonstrate their work on the food recommender system.

Shaozhong Zhang worked on customer suggestions and sentiments in E-commerce systems. The key parameter they have taken as trust that has been divided into two forms such as propagation of trust and direct trust, which characterizes a trust relationship among two people.

Problem Statement

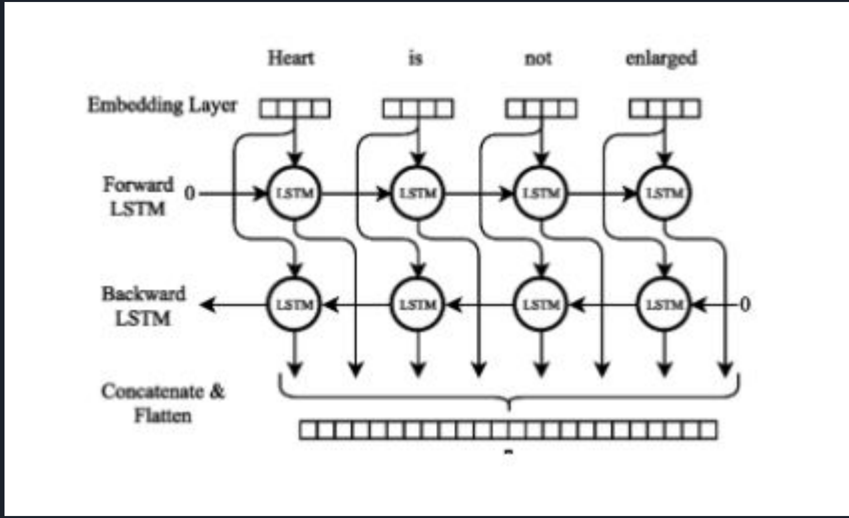
Building a recommendation system which utilises both the product details content and also the customer behavior to find alternative products to a given input product

Objectives

- Building a recommendation system using LSTMs based on Customer purchase history
- Building a recommendation system using LSTMs based on product description content
- Combining the two to give a robust system which can tackle majorly the cold start problem.

Bidirectional LSTMs

Bidirectional LSTMs are an extension of traditional LSTMs that can improve model performance on sequence classification problems.



Bidirectional LSTMs train two instead of one LSTMs on the input sequence. The first on the input sequence as-is and the second on a reversed copy of the input sequence.

Dataset

We have selected the Santander Product Recommendation Dataset for this project. Although this dataset is completely in spanish, there is a translation provided in the documentation. This dataset has 142 different products and it comes with 3 different files.

The first and the second files are that of product history. They are just split up into test and train splits of 90% and 10%. There are about 400k rows of product history in all.

The third file contains the product descriptions. This is used for the model which is supposed to retrieve information from the product details rather than product history.

Work done

- Cleanse and process the dataset from the Santander Product Recommendation Dataset
- For Product history model, we convert dataset into one-hot encoding format
- For Product description model, we tokenize the data and map words to numerals.
- Train LSTMs on the Santander product history dataset and the product description dataset.
- Check accuracy of the above models
- Combine the two models by fleecing the outputs and check improvement

Results

```
Epoch 91/100
0s - loss: 0.9267 - categorical_accuracy: 0.8617
Epoch 92/100
0s - loss: 0.9267 - categorical_accuracy: 0.8595
Epoch 93/100
0s - loss: 0.9262 - categorical_accuracy: 0.8611
Epoch 94/100
0s - loss: 0.9261 - categorical_accuracy: 0.8616
Epoch 95/100
0s - loss: 0.9258 - categorical_accuracy: 0.8613
Epoch 96/100
0s - loss: 0.9258 - categorical_accuracy: 0.8611
Epoch 97/100
0s - loss: 0.9255 - categorical_accuracy: 0.8603
Epoch 98/100
0s - loss: 0.9253 - categorical_accuracy: 0.8608
Epoch 99/100
0s - loss: 0.9251 - categorical_accuracy: 0.8595
Epoch 100/100
0s - loss: 0.9247 - categorical_accuracy: 0.8619
```

Accuracy of 86.19% achieved through product history dataset

```
In [91]: scores = model.evaluate(test_cnn_texts_mat,test_tags)

         400000/400000 [==============================] - 2655s 7ms/step

In [92]: print("Model accuracy on test data is",scores[1]*100,"percent")

         Model accuracy on test data is 94.84825000000001 percent
```

Accuracy of 94.85 % achieved with product description dataset

```
In [107]: scores = model.evaluate(test_cnn_texts_mat,test_tags)
          print("Combiend accuracy is ", scores[1]*100,"percent")

          Combiend accuracy is 92.768285000000001 percent
```

Accuracy of 92.77% achieved with merged model with fleeced output

Future work

The most important thing we notice is that the combining method of fleecing the outputs of the two models did not pay off completely in the end. So the next immediate goal is to research more into other methods of combining the outputs to get a better performance.

An improvement in the accuracy of the Model which trained on the customer history dataset can be brought about by including a more detailed feature engineering, coupling it with a heuristic algorithm like genetic algorithm and maybe utilising XGBoost as well to increase performance.

Individual contribution

Cleaning and preprocessing of dataset for the two ways of implementation: Sagar G

Training LSTMs on Santander Product History dataset : Akhil G

Training LSTMs on Santander Product Description dataset: Amodh Shenoy

THANK YOU