Recommendation report

Hokimiyon Muhammadjon m.hokimiyon@innopolis.university

December 2023

1 Introduction

This report describe solution of Recommendation system movies.

2 Data analysis

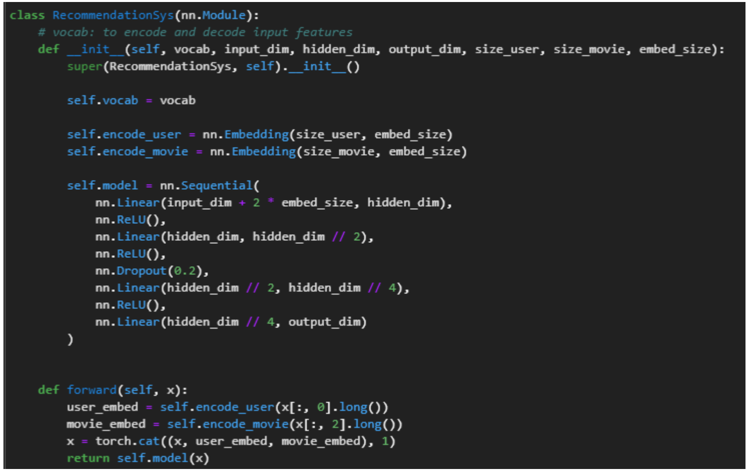
MovieLens 100K Dataset used to solve this problem.

The main table have information about user_id, movie_id and rating that user assign. The information about users and movies are assigned into different tables.

For our purpose, I put movie and user information into main table. Problem with string features solved by one hot encoding.

3 Model Implementation

To solve this problem, I implement Deep neural network. user_id and movie_id goes to embedding layer to achieve more information with their initial own features, then goes to 4 full connected layers.



```
class RecommendationSys(nn.Module):
    # vocab: to encode and decode input features
    def __init__(self, vocab, input_dim, hidden_dim, output_dim, size_user, size_movie, embed_size):
        super(RecommendationSys, self).__init__()

        self.vocab = vocab

        self.encode_user = nn.Embedding(size_user, embed_size)
        self.encode_movie = nn.Embedding(size_movie, embed_size)

        self.model = nn.Sequential(
            nn.Linear(input_dim + 2 * embed_size, hidden_dim),
            nn.ReLU(),
            nn.Linear(hidden_dim, hidden_dim // 2),
            nn.ReLU(),
            nn.Dropout(0.2),
            nn.Linear(hidden_dim // 2, hidden_dim // 4),
            nn.ReLU(),
            nn.Linear(hidden_dim // 4, output_dim)
        )


    def forward(self, x):
        user_embed = self.encode_user(x[:, 0].long())
        movie_embed = self.encode_movie(x[:, 2].long())
        x = torch.cat((x, user_embed, movie_embed), 1)
        return self.model(x)
```

4 Model Advantages and Disadvantages

Advantage :

- For movie i and user j, model works good if there enough people who ranked this movie.

Disadvantages :

- Model works bad on movie i if not enough users rank this movie.
- Need to train again if new movie or user appear.

5 Training Process

Training process of Recommendation system consist of several parts, first define loss function and optimizer.

Loss function: MSELoss

Optimization Algorithm: Adam

The training process for a Recommendation system involves loading and batching preprocessed data, initializing model parameters, defining a loss function, selecting an optimization algorithm, and iterating over the dataset for a set number of epochs. In each training iteration, the model computes forward and backward passes, calculates gradients using the loss function, and updates parameters via the chosen optimization algorithm. Validation on a separate dataset is performed periodically to monitor performance, and save the model with best performance.

Trained on 20 epochs with parameters:

- batch_size = 64
- input_dim = 45
- hidden_dim = 256
- output_dim = 1
- vocab_size = 23
- embed_size = 128
- size_user = 944
- size_movie = 1683
- learning_rate = 1e-3

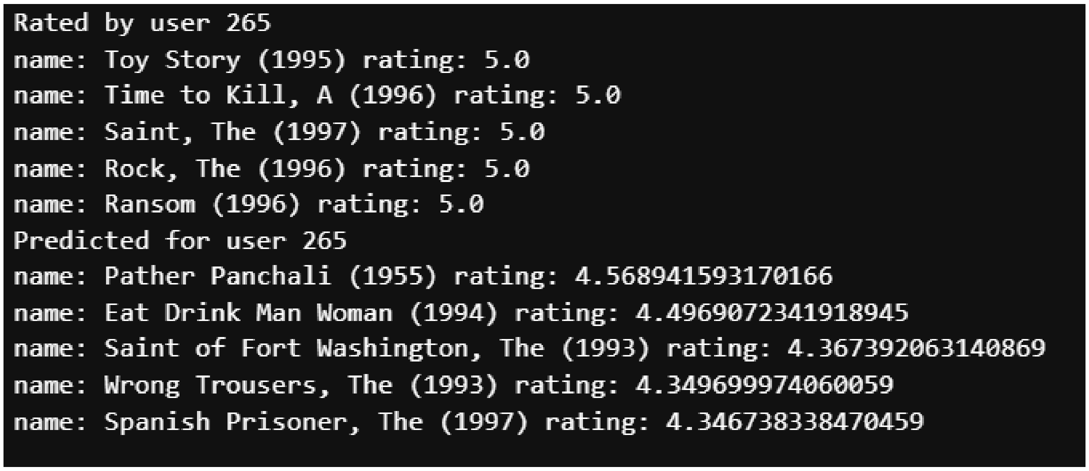
6 Evaluation

To evaluate model, MSE were used.

MSE: 0.8289415148454666

7 Results

Let's look at example. User_id 265 chosen by random, and that what model predict.



```
Rated by user 265
name: Toy Story (1995) rating: 5.0
name: Time to Kill, A (1996) rating: 5.0
name: Saint, The (1997) rating: 5.0
name: Rock, The (1996) rating: 5.0
name: Ransom (1996) rating: 5.0
Predicted for user 265
name: Pather Panchali (1955) rating: 4.568941593170166
name: Eat Drink Man Woman (1994) rating: 4.4969072341918945
name: Saint of Fort Washington, The (1993) rating: 4.367392063140869
name: Wrong Trousers, The (1993) rating: 4.349699974060059
name: Spanish Prisoner, The (1997) rating: 4.346738338470459
```