UNIVERSITY OF SAN FRANCISCO

The Rating Seekers

By Kyril Panfilov and Surbhi Prasad

Techniques Applied

Matrix Factorization (Pytorch)

- Creating and training user and item embeddings and multiplying the embedding matrices U*V to estimate predictions.

Embedding Vectorization

- Creating user and item embeddings again, but also adding context and item features by concatenating the features(with or without context and item feature) into a concatenated x vector with the label 1 or 0 as the target column.Instead of a matrix multiplication model, this technique transforms the problem into a tabular data set.
- Tried NeuMF where combined MF and Embedding Vectorization using MLP.

Ensemble Modeling

- We created separate model for users of all combinations: new users-old items,old users-old items,new users-new items,old users-new items and ensembled the probabilities for each to predict on test set.
- We selected models with similar performance on the data set using different features and hyperparameters, then averaged the test probabilities of each model to create an ensemble model.

Features Created

Negative Sampling

- First, we applied 1:4 ,1:2 sampling, where for every positive sample, we included four zero samples to be able to calculate probabilities.
- Later, we chose a 1:1 negative sampling technique which gave us better results.
- We also tried item category freq based probabilistic sampling(weighted sampling based on items)
- We also implemented Negative Sampling batch wise so that our negative samples change in every epoch.

Unknown User Sampling

- To help our model account for unknown users in validation and testing data, we sampled a fraction of our test data, and replaced the users with an unknown label.
- This allowed us to keep the real training data, while also using actual data to estimate a random user. We experimented with sampling different portions of data from 0.01 to 0.3 as a hyperparameter.

User and Item Frequency

- Created a column that measured how many times the item and user appeared in the training data. The idea was to help the model identify popular items and power users.

Experimental Results

Hyperparameters Tested

- Unknowns sampled as a proportion of training data: [0.05, 0.3]
- Negative sampling ratios of ones to zeros: [1:1, 1:4]
- Hidden layers: [10, 20, 30, 40, 50, 100, 200]
- Embedding size for users, items, item features, and context features: [10, 20, 30, 40, 50, 100, 200]
- Model dropout: [0, 0.001, 0.01, 0.1]
- DataLoader batch size: [1000, 2000, 3000, len(dataset)/2, len(dataset)/5, len(dataset)]
- Training epoch: [3, 5, 10, 20, 50]
- Learning rate: [0.001, 0.001, 0.01, 0.03, 0.1, 0.3, 1]
- Weight decay: [1e-5, 0.0001, 0.001, 0.01, 0.1]

Experimental Results

Model 1: Matrix Factorization

- Best hyperparameters:
  - Embedding size: 100
  - Epochs: 25
  - Learning rate: 0.1,0.01,0.001
  - Weight decay: 1e-5

Results:

- Train loss: 0.513
- Valid loss: 0.487
- Test loss: 0.49

Model 2: Vectorization Sequential NN

- Best hyperparameters:
  - Embedding size: 100
  - Epochs: 15
  - Learning rate: 0.01,0.03
  - Weight decay: 0
  - Dropout:0.01

Results:

- Train loss: 0.355
- Valid loss: 0.375
- Test loss: 0.43

Experimental Results

Model 3: NeuMF (MF+MLP)

- Best hyperparameters:
  - Embedding size: 64
  - Hidden Layers=3
  - Epochs: 43
  - Learning rate: 0.1,0.01,0.001
  - Weight decay: 1e-3, 1e-5
  - Dropout: p=0.2

Results:

- Train loss: 0.423
- Valid loss: 0.455
- Test loss: 0.446

Model 2: Ensemble of 4 models (User New, Old user, New Item, Old Item)

- Best hyperparameters:
  - Embedding size: 50,100
  - Epochs: 32
  - Learning rate: 0.03
  - Weight decay: 0

Results:

- Train loss: 0.437
- Valid loss: 0.492
- Test loss: 0.452

Experimental Results

Final Model: Ensemble Model

Average of Predictions of top 2 models (Vectorization + NeuMF) : Test Loss: 0.418

Model 2: Vectorization Sequential NN

- Best hyperparameters:
  - Embedding size: 100
  - Epochs: 15
  - Learning rate: 0.01,0.03
  - Weight decay: 0
  - Dropout:0.01

Results:

- Train loss: 0.355
- Valid loss: 0.375
- Test loss: 0.43

Model 3: NeuMF (MF+MLP)

- Best hyperparameters:
  - Embedding size: 64
  - Hidden Layers=30
  - Epochs: 43
  - Learning rate: 0.1,0.01,0.001
  - Weight decay: 1e-3, 1e-5
  - Dropout: p=0.2

Results:

- Train loss: 0.423
- Valid loss: 0.455
- Test loss: 0.446

Lessons Learned

- Some features are less important and can create more random noise that affects the model
- Implicit ratings are difficult because we have to make assumptions to negative sample the data. **This makes negative sampling the most critical part.**
- **Hyperparameter testing** was one of the most important parts of the problem that improved our validation loss from 0.49 to 0.43.
- GPU processing can greatly improve hyperparameter testing time cost
- There are different ways to implement recommender systems other than matrix factorization which allow us to use more features e.g. **3 layered NN and NeuMF**.
- Unknown users is an important problem to solve based on content based models since there were many new users in the testing data
- It's difficult to train users with only a few data points, so a **larger batch size** seemed to improve results
- **Ensembling** different approach models can improve accuracy