Lab 5-8

Dumitrana Mihnea

Model Development - code

1. Introduction to Recommendation System Modeling

In the realm of personalized content recommendations, the goal is to tailor suggestions based on users' preferences, behaviors, or past interactions. These systems are widely used in domains such as online retail (recommending products), streaming platforms (recommending movies or shows), and social media (recommending posts or friends). The challenge lies in predicting what content will be most relevant to an individual user from a large set of possible options.

For this research, we focus on using Neural Collaborative Filtering (NCF) as a recommendation model. NCF allows us to effectively model user-item interactions by learning complex patterns in the data through a neural network-based approach. Unlike traditional collaborative filtering methods, NCF provides better flexibility and accuracy, especially when data is sparse or when non-linear relationships exist between users and items.

2. Model Selection

Collaborative Filtering

Collaborative Filtering (CF) is one of the most popular techniques used in recommendation systems. It works by finding patterns in user behavior or item interactions, assuming that users who have agreed in the past will continue to agree in the future. CF can be:

- User-based: Recommending items based on what similar users liked.
- Item-based: Recommending items that are similar to those a user has previously liked.

While these techniques are effective, they have limitations, especially in handling sparse data and capturing non-linear relationships between users and items.

Content-Based Filtering

Content-based filtering approaches, in contrast, use features of the items themselves to make recommendations. For instance, in movie recommendations, content-based systems might recommend movies with similar genres, directors, or actors. However, these systems are often limited by the features available and may not capture the deeper relationships between users and items.

Neural Collaborative Filtering (NCF)

Neural Collaborative Filtering (NCF) combines the best of both worlds by leveraging the strengths of collaborative filtering while also capturing complex interactions through deep learning. NCF is particularly well-suited for recommendation tasks where user-item interactions are rich but complex. The model learns user and item embeddings and combines them through neural network layers, enabling it to capture non-linear patterns and interactions that simpler models might miss.

Given its flexibility and effectiveness in handling high-dimensional sparse data, NCF was selected as the primary model for this research.

3. Data Preprocessing

Data preprocessing is a critical step before training a neural network model. The following steps were performed:

1. Data Cleaning: The dataset was cleaned by removing missing or corrupted entries. Any rows with missing userId, movieId, or rating values were discarded.

2. Feature Encoding: Categorical features such as userId and movieId were encoded as integer values using Label Encoding. This is important for deep learning models, as neural networks require numerical inputs. The encoding was performed such that each unique user and movie received a unique integer ID.

3. Data Splitting: The dataset was split into training, validation, and test sets. An 80-20 split was used for training and validation, and a separate test set was used for evaluation.

4. Neural Network Architecture

The architecture of the Neural Collaborative Filtering (NCF) model includes several key components:

- User and Item Embedding Layers: User and item IDs are mapped to dense embedding vectors. These vectors are learned during the training process to capture latent features that represent users and items in a lower-dimensional space.

- Merging Layer: The embeddings for both the user and the item are concatenated to form a combined vector. This combined representation is then passed through additional neural network layers to learn the complex interactions between users and items.

- Fully Connected Layers: These layers (with non-linear activation functions like ReLU) learn higher-order interactions between the user-item embeddings. These layers are designed to capture non-linear patterns in the interactions.

- Output Layer: The output of the network is a sigmoid activation, which produces a probability between 0 and 1. This can be interpreted as the likelihood that the user will like the item (in the case of binary classification for ratings >= 3).

- Loss Function: The model uses binary cross-entropy as the loss function, which is suitable for the binary classification problem (ratings >= 3 or < 3).

5. Model Training

The model was trained using the following parameters:

- Optimizer: Adam optimizer was used, with a learning rate of 0.001. Adam is an adaptive optimizer that adjusts the learning rate based on the gradient, helping with convergence in complex models.
- Epochs: The model was trained for 10 epochs. The choice of 10 epochs was made based on experimentation and the fact that the loss seemed to converge well within this range.
- Batch Size: A batch size of 64 was used, balancing between efficient training and memory constraints.
- Regularization: To prevent overfitting, dropout was used in the hidden layers, and L2 regularization was applied to the embedding layers.

6. Model Evaluation

The model's performance was evaluated using the following metrics:

- Training and Validation Loss: The loss values for both training and validation sets were tracked during training. The model's performance was compared on both datasets to check for overfitting.
- Precision: Measures the proportion of positive predictions that were actually correct. A high precision indicates that most of the positive predictions were accurate.
- Recall: Measures the proportion of actual positives that were correctly predicted. High recall indicates that the model successfully identified most of the positive examples.
- F1 Score: The harmonic mean of precision and recall, providing a balance between the two metrics. A high F1 score means the model is performing well in both identifying positives and minimizing false positives.
- Confusion Matrix: A confusion matrix was generated to analyze the true positive, false positive, true negative, and false negative predictions.

7. Hyperparameter Tuning

Several hyperparameters were tuned during model development:

- Learning Rate: The learning rate was tuned using trial and error. Starting with 0.001, the learning rate was adjusted based on the model's performance.
- Batch Size: Various batch sizes were tested, and 64 was found to be optimal for training speed and memory usage.

- Dropout Rate: Dropout was applied at various rates (e.g., 0.2, 0.3), and the final model used a dropout rate of 0.3 for regularization.

8. Results and Insights

Training Loss and Validation Loss: The model showed a steady decrease in both training and validation loss, with validation loss slightly higher than training loss, indicating some overfitting.

Confusion Matrix: The confusion matrix showed that the model correctly predicted the majority of positive and negative cases. However, some false positives and false negatives were still present, which is typical in recommendation systems where thresholds for "liking" items can vary across users.

Performance Metrics: The precision, recall, and F1 scores confirmed that the model was able to correctly identify a substantial portion of positive recommendations while minimizing false positives.

9. Conclusion

The Neural Collaborative Filtering model performed well in predicting user preferences for movies, with a balanced precision, recall, and F1 score. While the model showed promising results, further improvements could be made by incorporating more data, tuning additional hyperparameters, or experimenting with more advanced architectures, such as incorporating temporal dynamics or multi-task learning.

Future work should also focus on scaling the model to larger datasets and investigating the potential of incorporating additional features into the model (such as genre, director, or actor preferences).