كليـــــة العلـــــوم
والتقنيات - مراكش
FACULTE DES SCIENCES
ET TECHNIQUES - MARRAKECH

Master IAII
Pr. Sara QASSIMI
1 Decembre 2023

Tutorial :Deep neural network based Recommendation

Objective:

The primary objective of this tutorial is to guide in implementing a deep neural network-based recommender system. By using the MovieLens dataset as a starting point, the tutorial aims to provide a step-by-step walkthrough, beginning with data loading and preprocessing and concluding with the training and evaluation of a neural network model.
The tutorial further challenges is about surpassing the performance of a matrix factorization model by experimenting with hyperparameters, such as the number of layers, hidden units, hidden activation functions, and optimizers. Additionally, the tutorial encourages you to apply your acquired knowledge to a different dataset, specifically the Book-Crossing dataset, demonstrating the adaptability of the recommender system to diverse data sources.

Tasks:

Step 1: Import Libraries

```
from __future__ import print_function, division
from builtins import range, input
import pickle
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.utils import shuffle
from keras.models import Model
from keras.layers import Input, Embedding, Flatten, Dense, Concatenate
from keras.optimizers import SGD, Adam
```

Step 2: Load and Preprocess Data

```
# Load MovieLens data
df = pd.read_csv('../../movielens-20m-dataset/edited_rating.csv')

# Number of users and movies
N = df.userId.max() + 1
M = df.movie_idx.max() + 1

# Split data into train and test sets
df = shuffle(df)
cutoff = int(0.8 * len(df))
df_train = df.iloc[:cutoff]
df_test = df.iloc[cutoff:]
```

Step 3: Define Model Hyperparameters

```
K = 10  # latent dimensionality
mu = df_train.rating.mean()
epochs = 15
```

Step 4: Build Model

```
# Define input layers
u = Input(shape=(1,))
m = Input(shape=(1,))

# User and movie embeddings
u_embedding = Embedding(N, K)(u)
m_embedding = Embedding(M, K)(m)
u_embedding = Flatten()(u_embedding)
m_embedding = Flatten()(m_embedding)

# Concatenate user and movie embeddings
x = Concatenate()([u_embedding, m_embedding])

# Neural network layers
x = Dense(400)(x)
x = Activation('relu')(x)
x = Dense(1)(x)

# Build the model
model = Model(inputs=[u, m], outputs=x)

# Compile the model
model.compile(
    loss='mse',
    optimizer=SGD(lr=0.08, momentum=0.9),
    metrics=['mse'],
)
```

Step 5: Train the Model

```
r = model.fit(
    x=[df_train.userId.values, df_train.movie_idx.values],
    y=df_train.rating.values - mu,
    epochs=epochs,
    batch_size=128,
    validation_data=(
        [df_test.userId.values, df_test.movie_idx.values],
        df_test.rating.values - mu
    )
)
```

Step 6: Evaluate and Visualize Results

```
# Plot losses
plt.plot(r.history['loss'], label="train loss")
plt.plot(r.history['val_loss'], label="test loss")
plt.legend()
plt.show()

# Plot MSE
plt.plot(r.history['mean_squared_error'], label="train mse")
plt.plot(r.history['val_mean_squared_error'], label="test mse")
plt.legend()
plt.show()
```

Step 7: Hyperparameter Tuning Challenge

Modify hyperparameters such as the number of layers, hidden units, hidden activation, and optimizer to improve the model's performance. Experiment and find the combination that works best for your specific task.

كليـــــة العلـــــوم
والتقنيات - مراكش
FACULTE DES SCIENCES
ET TECHNIQUES - MARRAKECH

Master IAII
Pr. Sara QASSIMI
1 Decembre 2023

Step 8: Adapt for Book-Crossing Dataset

Replace the MovieLens dataset with the Book-Crossing dataset.

https://www.kaggle.com/datasets/ruchi798/bookcrossing-dataset

Good luck with the hyperparameter tuning and adapting the script for the Book-Crossing dataset!