Jester Joke System

Overview

This project builds a recommendation system using a basic Multi-Layer Perceptron (MLP) model to predict user ratings of jokes from the Jester dataset. The jokes are encoded using the BERT model to convert text into numerical vectors. The project involves data preparation, model training, evaluation, and investigating the effect of different learning rates on model performance.

Firstly downloaded jester_rating and jester_items files are loaded.

Data Preparation

Extract Columns

Extract the necessary columns: joke texts, user IDs, joke IDs, and ratings from the provided datasets.

Encode Jokes

Use the BERT model from the sentence_transformers library to encode the joke texts into numerical vectors.

Prepare Dataset with Embeddings

Create a CSV file with user IDs, joke IDs, ratings, and joke embeddings. Header.

Split Data into Training and Validation Sets

Split the dataset into training and validation sets, and normalize the features.

```
#control under or overfit
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import normalize

x2 = maindata.drop(["jokeid", "rating", "usserid"], axis=1) # to predictions
y2 = maindata['rating'] # target to predict

x_train, x_val, y_train, y_val = train_test_split(x2, y2, test_size=0.3, random_state=100)
#val set to eval performance of trained model
x_train2 = normalize(x_train)#ensure each feature similar scale that influences model trai
#x_val = normalize(x_val)
```

Model Training and Evaluation

Train Basic MLP Model

Train an MLP model with default hyperparameters and evaluate its performance on the validation set.

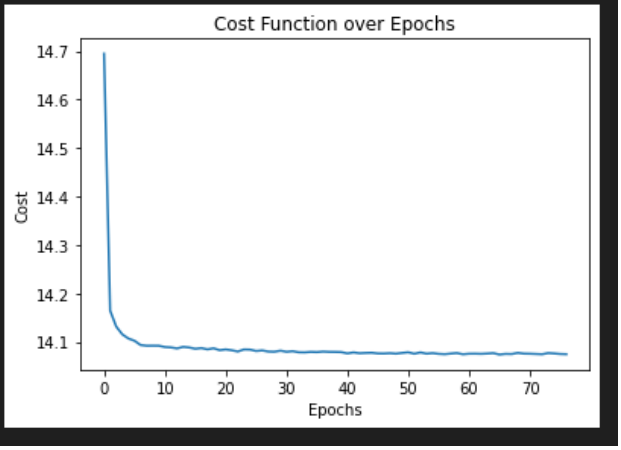
Plot Cost Function

Plot the cost function over epochs to visualize the training process.

training a basic MLP model with two hidden layers using the Jester dataset. The model is configured with default hyperparameters. After training, it predicts the validation set outcomes, evaluates performance using Mean Squared Error (MSE), and visualizes the cost function over epochs to monitor training progress and convergence.

Below is the representation:

```
Validation MSE: 28.005312488136276
```



. Investigate the effect of learning rate on performance: repeat learning for 3 different parameter values.

This section tests the effect of different learning rates (0.0001, 0.001, 0.01) on the MLP model's performance. Each model is trained with a specific learning rate, and its validation MSE is calculated. The cost functions over epochs are plotted to compare training progress across learning rates.

Below is representation of the different learning rates result with the graph.

Number of epochs through the training data is presented together with the cost(loss).