MOVIE RECOMMENDATION SYSTEM

USING

NEURAL NETWORKS

MEMBERS:

1. SOHAIB ASHRAF (20K-0488)
2. MUHAMMAD SHAHZAIB (20K-1067)
3. KHAWAJAH ABDULLAH (20K-0385)

RECOMENDATION BASED ON COLLABORATIVE FILTERING TECHNIQUE USING NEURAL NETWORKS.

INTRODUCTION:

This project is a recommendation system that recommends top 10 unwatched movies to a particular user input. The project uses the collaborative filtering technique which recommends movies based on the ratings and behavior of other users who have similar preferences.

Overview:

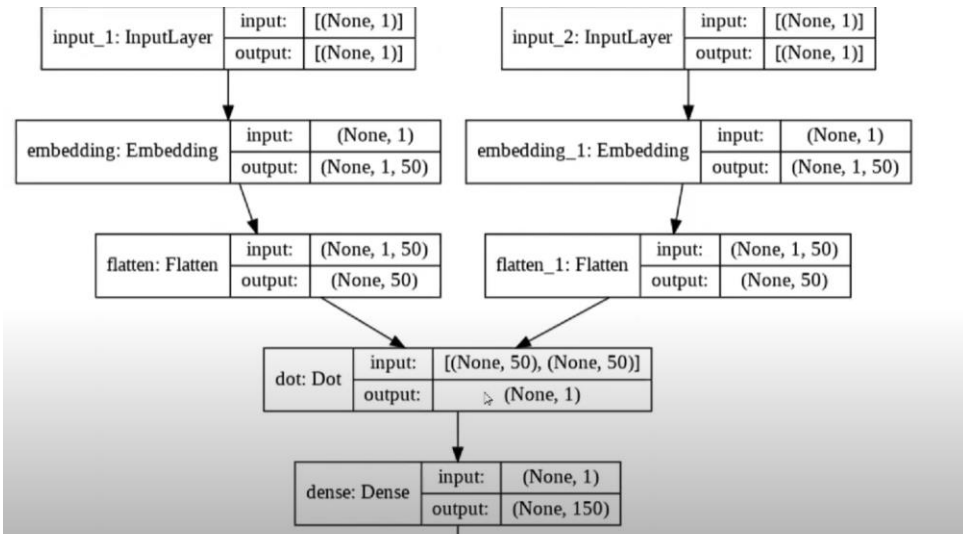
The project uses a neural network with two embedding layers - one for the user and another for the movie. The neural network takes the user and movie embeddings as input and predicts the ratings for the given user-movie pair. The predicted ratings are then used to recommend the top 10 unwatched movies to the user.

The dataset used in this project consists of two tables - movies and ratings. The tables are merged using the movie_id as the key to create a new table. A new table is then created with selected columns such as userId, movieId, rating, and timestamp. The userId and movieId are then encoded into numerical values using dictionaries.

The ratings are then normalized between 0 and 1 using the minimum and maximum ratings. The normalized ratings are split into training and validation sets. The neural network is trained on the training set and the validation set is used to evaluate the model's performance.

Finally, the user is prompted to input their user ID, and the neural network is used to recommend top 10 unwatched movies to the user. The recommended movies are printed out along with their titles and genres.

MAIN IDEA BEHIND NEURAL NETWORK EMPLOYED:



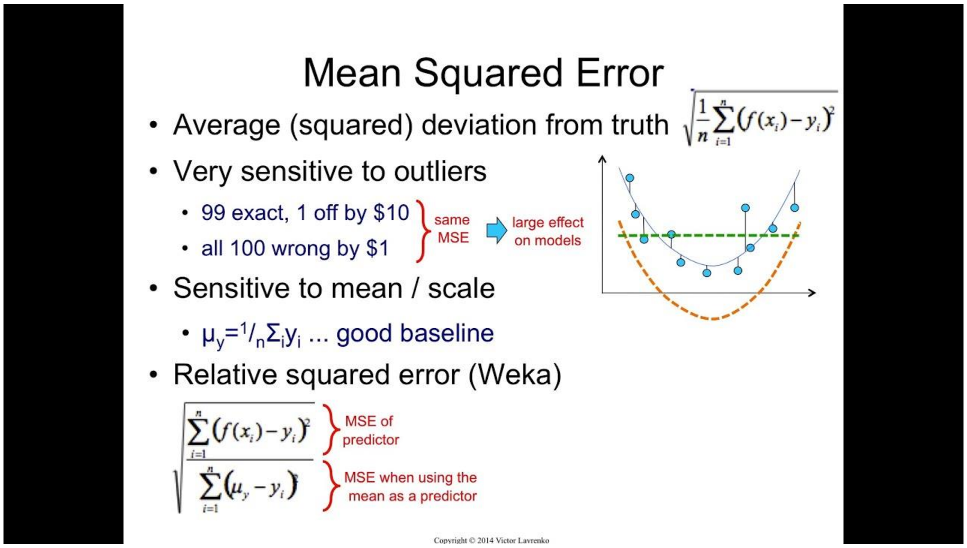
The neural collaborative filtering model is a machine learning algorithm that is designed to predict user ratings for different movies. The model is based on general matrix factorization, where each user and movie are represented by a 50-dimensional vector in a latent space of feature space. These vectors are obtained using embedding layers, which transform the user and movie input data into the latent space. The flattened embedding layers are then passed through a dot product layer, which calculates the similarity between the user and movie latent vectors.

After the dot product layer, the output is passed through a multi-layer perceptron (MLP) that creates non-linearities and increases the dimensionality of the data. The MLP is composed of two dense layers, where the first dense layer increases the dimensionality of the dot product output to 150, and the second layer reduces it back to 50. The final output of the MLP is a single value between 0 and 1, which is treated as a rating prediction for the corresponding user and movie interaction.

The neural collaborative filtering model is trained using a regression task, where the objective is to match the predicted ratings with the actual ratings from the dataset. The model takes two inputs: the user and the movie, and outputs a predicted rating. By iterating over the entire dataset, the model learns the latent space of features and can make accurate rating predictions for new user-movie interactions. Overall, the neural collaborative filtering model is a powerful tool for recommending movies to users based on their past viewing habits and preferences.

Nonlinearity is added by the 3 dense layers which take the output of the dot product, map to 150 outputs and then minimize back to one single normalized rating.

Minimization of error and optimization:



In this project, the loss function used for training the neural network is mean squared error. The goal of training the model is to minimize this loss function. During each epoch of training, the model makes predictions on the training data and calculates the mean squared error between the predicted ratings and the actual ratings. The optimizer is responsible for adjusting the weights of the neural network in order to minimize this loss function.

Algorithm 8.5 The RMSProp algorithm

Require: Global learning rate ϵ, decay rate ρ.
Require: Initial parameter $\boldsymbol{\theta}$
Require: Small constant δ, usually 10^{-6}, used to stabilize division by small numbers.
Initialize accumulation variables $\boldsymbol{r} = 0$
while stopping criterion not met **do**
Sample a minibatch of m examples from the training set $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ with corresponding targets $\boldsymbol{y}^{(i)}$.
Compute gradient: $\boldsymbol{g} \leftarrow \frac{1}{m}\nabla_{\boldsymbol{\theta}} \sum_i L(f(\boldsymbol{x}^{(i)};\boldsymbol{\theta}), \boldsymbol{y}^{(i)})$
Accumulate squared gradient: $\boldsymbol{r} \leftarrow \rho \boldsymbol{r} + (1-\rho)\boldsymbol{g} \odot \boldsymbol{g}$
Compute parameter update: $\Delta\boldsymbol{\theta} = -\frac{\epsilon}{\sqrt{\delta+\boldsymbol{r}}} \odot \boldsymbol{g}$. ($\frac{1}{\sqrt{\delta+\boldsymbol{r}}}$ applied element-wise)
Apply update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$
end while

Adam, short for Adaptive Moment Estimation, is an optimizer that combines the advantages of two other optimizers, namely RMSProp and AdaGrad. Adam computes individual adaptive learning rates for each parameter, meaning that the learning rate can vary based on the magnitude of the gradient and the past history of the gradient. This helps the optimizer to converge faster and adapt better to the changing gradients during training. Additionally, Adam also includes bias correction to correct the bias introduced in the estimates of the first and second moments of the gradients. By using Adam as the optimizer, the neural network in this project can achieve faster convergence and better generalization on the test data.

OUTPUT AND RESULTS DERIVED:

Some sample outputs:

```
1418/1418 [==============================] - 15s 10ms/step - loss: 0.0295
Epoch 5/10
1418/1418 [==============================] - 15s 10ms/step - loss: 0.0244
Epoch 6/10
1418/1418 [==============================] - 15s 10ms/step - loss: 0.0224
Epoch 7/10
1418/1418 [==============================] - 15s 10ms/step - loss: 0.0201
Epoch 8/10
1418/1418 [==============================] - 15s 11ms/step - loss: 0.0186
Epoch 9/10
1418/1418 [==============================] - 15s 10ms/step - loss: 0.0177
Epoch 10/10
1418/1418 [==============================] - 15s 10ms/step - loss: 0.0166
```

We used 10 epochs in this particular network. Using more epochs does not mean the loss in the predictions will reduce. The figure below shows the movies suggested to the user. Now, let us look at the top ten movies viewed by this particular user.

```
Rear Window (1954) : Mystery|Thriller
Gone with the Wind (1939) : Drama|Romance|War
Touch of Evil (1958) : Crime|Film-Noir|Thriller
Ben-Hur (1959) : Action|Adventure|Drama
Star Trek II: The Wrath of Khan (1982) : Action|Adventure|Sci-Fi|Thriller
Thing, The (1982) : Action|Horror|Sci-Fi|Thriller
Karate Kid, The (1984) : Drama
eXistenZ (1999) : Action|Sci-Fi|Thriller
Rashomon (Rashômon) (1950) : Crime|Drama|Mystery
Spirited Away (Sen to Chihiro no kamikakushi) (2001) : Adventure|Animation|Fantasy
PS C:\Users\User\Desktop\AI-PROJECT>
```

Now let us investigate the movies liked by the user, which are given in the figure below.

```
Godfather, The (1972) : Crime|Drama
Star Wars: Episode V - The Empire Strikes Back (1980) : Action|Adventure|Sci-Fi
Back to the Future (1985) : Adventure|Comedy|Sci-Fi
Indiana Jones and the Last Crusade (1989) : Action|Adventure
When Harry Met Sally... (1989) : Comedy|Romance
Last of the Mohicans, The (1992) : Action|Romance|War|Western
Gattaca (1997) : Drama|Sci-Fi|Thriller
Rushmore (1998) : Comedy|Drama
Cider House Rules, The (1999) : Drama
Baraka (1992) : Documentary
TOP 10 Recommended movies for the user  480
```

A close inspection of the movies proves that our model is showing positive results as the movies suggested and previously highly rated movies are in close conformity in terms of the genres. User prefers Crime, drama and sci-fi movies more often and is suggested accordingly.