Recommendation Systems HW1 Part1
Amir Wolfensohn 300339785
Yishai Shapira 203016217
Or Sharir 201314796

Part 1 – Stochastic Gradient Descent:

1. **Write the objective of a regression model with global bias, user bias, item bias and L2 regularization.**

   A general regression model objective is to explain one variable variance using another variable and the bias is the error that is existing since we are dealing with real life problems.

   Here we are trying to predict user i (1,...,N) rating on item j (1,...,M) with global bias of general noise such as a completely new user or item for example, user bias of user noise such as a certain user on completely new item for example, item bias of item noise such as a certain item from completely new user for example.
   L2 regularisation that will improve the validation/test error.

2. **Write the update step for each parameter.**

hype parameters: $\lambda_{bu}, \lambda_{bi}, \lambda_x, \lambda_y, \eta_x, \eta_y, \eta_{bu}, \eta_{bi}$

$$\frac{\partial}{\partial x_u} loss = -2\overbrace{(r_{ui} - \hat{r}_{ui})}^{e_{ui}} y_i + 2\lambda_x x_y$$

$$x_u \leftarrow x_u + \eta_x [e_{ui} y_i - \lambda_x x_u]$$

$$y_i \leftarrow y_i + \eta_y [e_{ui} x_u - \lambda_y y_i]$$

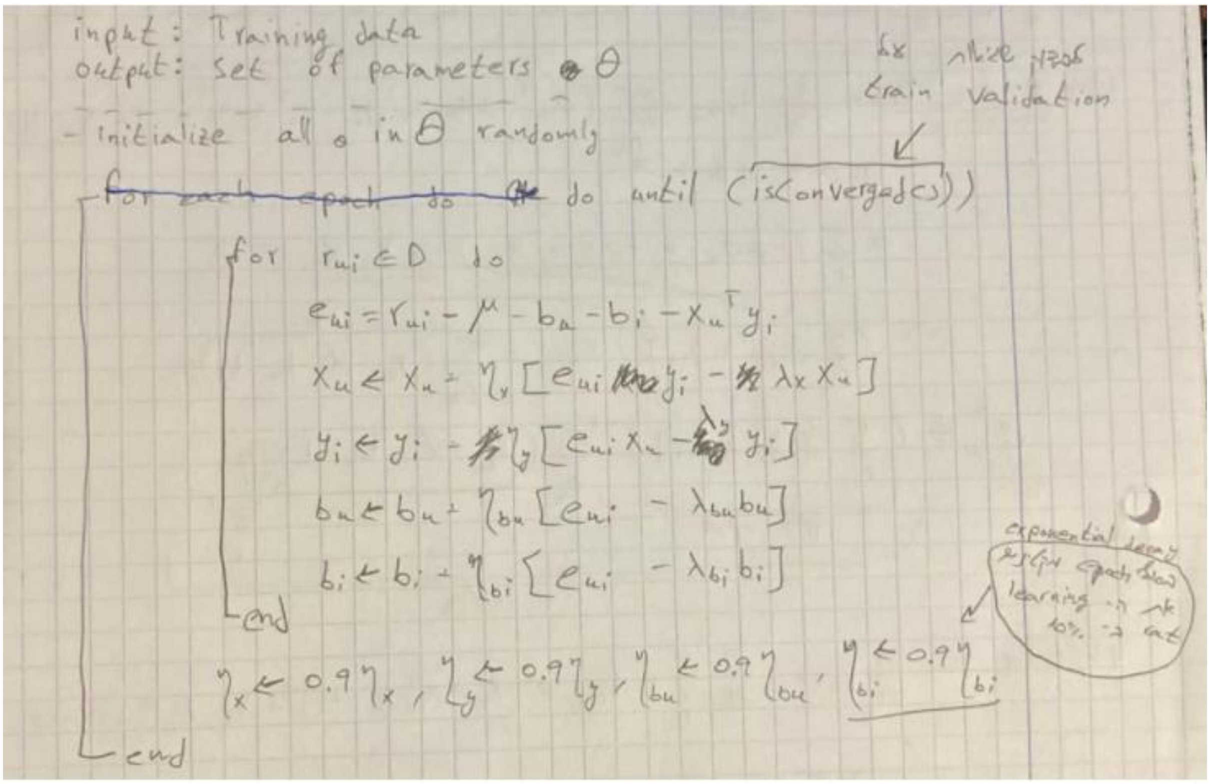
$$\frac{\partial}{\partial b_u} loss = -2e_{ui} + 2\lambda_{bu} b_u$$

$$b_u \leftarrow b_u + \eta_{bu} [e_{ui} - \lambda_{bu} b_u]$$

$$b_i \leftarrow b_i + \eta_{bi} [e_{ui} - \lambda_{bi} \cdot b_i]$$

- xu is the user latent vector (n*K).
- yi is the item latent vector(m*K).
- bu is the user bias vector(n*1)).
- bi is the item bias vector(m*1).

3. **Write a pseudo code for the algorithm**



input: Training data
output: set of parameters θ

- initialize all in θ randomly

~~for each epoch do~~ do until (isConverged())

for $r_{ui} \in D$ do

$e_{ui} = r_{ui} - \mu - b_u - b_i - x_u^T y_i$

$x_u \leftarrow x_u + \eta_x [e_{ui} y_i - \lambda_x x_u]$

$y_i \leftarrow y_i + \eta_y [e_{ui} x_u - \lambda_y y_i]$

$b_u \leftarrow b_u + \eta_{bu} [e_{ui} - \lambda_{bu} b_u]$

$b_i \leftarrow b_i + \eta_{bi} [e_{ui} - \lambda_{bi} b_i]$

end

$\eta_x \leftarrow 0.9\eta_x$, $\eta_y \leftarrow 0.9\eta_y$, $\eta_{bu} \leftarrow 0.9\eta_{bu}$, $\eta_{bi} \leftarrow 0.9\eta_{bi}$

end

train validation

exponential decay

learning

10%

4. **What hyper-parameters do you need to tune?**
    - K - the dimension of the user/item latent vectors.
    - user/item regularisation factors.
    - user/item learning rates.
    - decay rate in learning rate from one iteration to another.

5. **Explain how would you work with the validation set and how would you check for convergence?**

After every iteration with updates I'll build a prediction matrix for the validation data based on the current updates and will check the error.

I'll keep the iterations going till I'll have 2 increases in validation error (such as described above) in a row and then I'll go back to the last iteration before the first increase and take the prediction matrix based on the updates in that iteration and that is a convergence.

The increase in the validation error is caused by over fitting.

6. **How would you train the last \ best model?**

   After I'll get convergence as described above I'll have the best trained model and I can use it to predict the test set ratings.

7. **Implement a SGD solution for the model and train it using the training and validation data. Explain the main work items you had to take.**

   We initialized all the hyper parameters and general parameters randomly after we set up an 'environment' for each parameter and hyper parameter-
   K : [10, 20, 30, 40, 50, 60]
   learning rate : [0.0001 , 0.003]
   user lambda: (0.01, 0.05)
   item lambda : (0.001, 0.05)
   U,V - normal distribution
   user bias - (0.01, 0.05)
   item bias - (0.001, 0.05)

   Then we implemented the SGD process on our training data for maximum times of 100 iterations (we chose 100 for long performance reasons) but an 'early stop' check was made in each iteration so we checked in each iteration if the MSE on the validation is increasing in this iteration and the last one (2 error's increases in a row) and if so we'll take the last best iteration (the one before the first increase) to as our wanted time of iteration and if not we'll let the loop finish the 100 iterations.
   We are doing this process for 3 times in total (extra 2 times), each time we are reinitializing the parameters and hyper parameters and basically we are taking out of the 3 times the one with the minimum MSE.

   After all that we now have our best initializing and best number of iterations so we are running our SGDfor the last time with that best initializing and best number of iteration on the training + validation data.

   Finally, after we trained the model as described above, we are predicting the test data ratings.

8. **What is the RMSE, MAE, R^2 and MPR of your model based on the validation set?**

   MSE - 0.8889

RMSE - 0.94281
MAE - 0.74410
R^2 - 0.2851

```
In [15]: #priniting all the errors measures
print(msee(val_R, mu, b_u, b_i, U, V))#mse
print(np.sqrt(msee(val_R, mu, b_u, b_i, U, V)))#rmse
print(maee(val_R, mu, b_u, b_i, U, V))#mae
print(r_sq(val_R, mu, b_u, b_i, U, V))#r squares

0.8889057347002299
0.9428179753803116
0.7441010373641764
0.285133122858226
```

9. **Submit the test result file according to the following instructions:**
Was sent according to the instructions.