This handout includes space for every question that requires a written response. Please feel free to use it to handwrite your solutions (legibly, please). If you choose to typeset your solutions, the `README.md` for this assignment includes instructions to regenerate this handout with your typeset LaTeX solutions.

3.a No. Based on the train/test loss curves, parameter sharing doesn't outperform having separate models. Both the joint training loss and the evaluation loss are higher when model shares parameter.

Since $\lambda_{LF} = 0.99$, the model prioritizes reducing the loss of the factorization model. And by sharing the same parameter between factorization and regression tasks, the rate at which factorization loss decreases is slower.

This is reflected in the evaluation performance as well. The Mean Reciprocal rank - which corresponds to the performance of the factorization model - is lower for the model which shares parameter.

The regression task also suffers when using shared parameters that is primarily tuned for the factorization task. Hence we see lower MSE in both evaluation and training datasets when paramters are not shared.

3.b

No. Parameter sharing doesn't outperform having separate models in this scenario as well.

$\lambda_{LF} = \lambda_R = 0.5$. Hence, both the tasks are prioritized equally. However, the training MSE in the regression model starts off at 12 while the Factorization loss starts off at 0.5. Thus, the model tries to prioritize reducing the MSE.

As a result when the model shares parameters, reducing the training factorization loss takes a back seat and at the end of the training cycle the factorization loss and MSE converges to (0.42, 0.83), respectively.

However when the model doesn't share parameters, individual tasks and parameters can be separately optimized. Hence we can see both the training losses (factorization, MSE) decrease gradually to (0.25, 0.83).

The training impact is seen on evaluation dataset too. Since the model parameters are more tuned to the regression task when the model shares parameters, the Mean Reciprocal Rank (which reflects the performance of the Factorization task) takes a hit. And hence we see a lower rank when model shares parameter when compared to having separate models

3.c

From the training loss curve, the initial MSE = 13 and the factorization loss = 0.5. Clearly, the losses for both of these functions are not on the same scale. While the first task comes from a regression model with output ranging from 1 to 5, the second factorization task is a probability score which ranges from 0 to 1.

The model tries to reduce the joint loss = $\lambda_{LF} *$Factorization loss + λ_R*MSE.

When $\lambda_{LF} = 0.99$, the model prioritizes the factorization model and tries to learn the factorization task by reducing the factorization loss.

When $\lambda_{LF} = 0.5$, the model gives equal priority to both these tasks. However, since the loss of the regression task is an order of magnitude higher than the factorization task, the model prioritizes the regression task and tries to tune the parameters to reduce the MSE.

Thus in the Factorization task, we would expect the model with $\lambda_{LF} = 0.99$ to perform better. And that is precisely what we see. The Mean Reciprocal Rank (in the evaluation dataset) for $\lambda_{LF} = 0.99$ is higher compared to $\lambda_{LF} = 0.5$. The training Factorization loss too is lower for $\lambda_{LF} = 0.99$ compared to $\lambda_{LF} = 0.5$

Likewise, in the Regression task, we would expect the model with $\lambda_{LF} = 0.5$ to perform better. The MSE loss is lower for $\lambda_{LF} = 0.5$ on both evaluation and training dataset when compared to $\lambda_{LF} = 0.99$