HDDA Project

Introduction

In this project, we attempt to develop a simple Recommender System. Given training data consisting of m users and i items. We would like to build a mathematical model that would enable us to predict the ratings of:

1. new users on given items
2. new items by already existing users

Notation

The matrix W is an indicator matrix indicating whether an item i was rated by a given user u. More formally: $W \in \mathbb{R}^{m,n}$ wwhere

$$W_{u,i} = \begin{cases} 1 & \text{if}(u,i) \in \Omega \\ 0 & \text{otherwise} \end{cases}$$

The ratings matrix is denoted by X. The known ratings can be mathematically expressed in matrix form as:

$$W \circ X$$

where $\circ$ is the element-wise matrix product.

Further more we define:

- S_u as the set of users in the training data
- S_i as the set of items in the training data.
- Ω as the set of pairs (u,i) where user u rated the item i

We assume that each item $i\S_i$ has been rated by at least one user, and similarly each user $u \in S_u$ rated at least one givem item.

Baseline model

The initial step is to to build numerical / mathematical representations for both users and items denoted by $U \in \mathbb{R}^{m,k}$ and $V \in \mathbb{R}^{m,k}$ respectively. The i -th row in U represents the i -th user and similarity, the j -th row in V represents the j -th item.

Modeling the rating of a the i -th user u to the i -th item as the dot product between the representations, then one to `learn` good representations from the data is to minimize the following mathematical objective:

$$\min_{U \in \mathbb{R}^{m\times r}, V \in \mathbb{R}^{n\times r}} L_1 = \frac{1}{2}\sum_{u=1}^{m}\sum_{i=1}^{n} W_{u,i} \left(X - UV^T\right)^2_{u,i} + \lambda \cdot (||U_u||^2 + ||V_i||^2) \tag{1}$$

For the k -th user and the j -th item, we define \$ \$

$$f_1(U_k, V_j) = \frac{1}{2}(X_{j,k} - U_k^T \cdot V_j)^2 + \lambda \cdot (||U_u||^2 + ||V_i||^2) \tag{2}$$

It is easy to see that

$$L_1 = \sum_{(i,j)\in\Omega} f_1(U_k, V_j) \tag{3}$$

Minimizing the optimizaton objetive

We consider 2 algorithms to solve the optimization problem (1):

- Stochastic Gradient Desent (SGD)
- Block Coordinate Descent (BCD)

SGD

Given K as the number of iterations and α as the step size.

1. Initialize V and U
2. For k to K:
    1. iterate through all pairs rated (user, item) pairs, For the j -th user and k -th item:
        - apply both updates simultaneously:
            - $V_k \leftarrow V_k - \alpha \cdot \frac{\nabla f_1(V_j, U_k)}{\nabla V_k}$
            - $U_j \leftarrow U_j - \alpha \cdot \frac{\nabla f_1(V_j, U_k)}{\nabla U_j}$

BCD

```
1. Initialize $V$ and $U$
2. For $k$ to $K$:
    1. iterate through all pairs rated (user, item) pairs, For the $j$ -th user and $k$ -th item:
        1. update items representations:
            • $V_k \leftarrow V_k - \alpha \cdot \frac{\nabla f_1(V_j, U_k)}{\nabla V_k}$
    2. iterate through all pairs rated (user, item) pairs, For the $j$ -th user and $k$ -th item:
        1. update users representations:
            • $U_j \leftarrow U_j - \alpha \cdot \frac{\nabla f_1(V_j, U_k)}{\nabla U_j}$
```

Prediction

Prediction $\hat{y}$ for a (u, i) pair is made as follows

$$\hat{y} = \begin{cases} UV^T & \text{if } (u,i) \in \Omega \\ \mu & \text{otherwise} \end{cases}$$

where μ is the average prediction computed from the training data

$$\mu = \frac{1}{\sum_{u=1}^{m} \sum_{i=1}^{n} W_{ij}} \sum_{u=1}^{m} \sum_{i=1}^{n} W_{ij} X_{ij}$$

Mathematical Derivations

The mathematical function f_1 is simple. Since the term $X_{j,k} - U_k^T \cdot V_j$, we have:

$$\frac{\nabla f(V_j, U_k)}{\nabla V_j} = -((X_{j,k} - U_k^T \cdot V_j) \cdot V_j - \lambda \cdot V_j)$$

Similarly

$$\frac{\nabla f(V_j, U_k)}{\nabla V_k} = -((X_{j,k} - U_k^T \cdot V_j) \cdot U_k - \lambda \cdot U_k)$$

By defining $e_{i,u}$ as

$$e_{i,u} = X_{i,u} - U_u^T \cdot V_i$$

The updates can be expressed as:

- $V_i \leftarrow V_i + \alpha \left(e_{u,i} \cdot U_u - \lambda V_i \right)$
- $U_u \leftarrow U_u + \alpha \left(e_{u,i} \cdot V_i - \lambda U_u \right)$

Improved Model

The main downside of the baseline model is its inability to make prediction for users or items that do not have any ratings. The baseline model is capable to predict a rating for a (u, i) pair only if such user such user $u \in \S_u$ and $i \in \S_i$.

$$\min_{U\in\mathbb{R}^{m\times r}, V\in\mathbb{R}^{n\times r}} L_2 = \frac{1}{2}\sum_{u=1}^{m}\sum_{i=1}^{n} W_{u,i}\left(X - (\mu + \beta_u + b_i + UV^T)\right)^2_{u,i} + \lambda\cdot(\beta_u + b_i + ||U_u||^2 + ||V_i||^2)$$

where μ is the average rating, β is the bias of each user, and b is the bias of each item.

$$f_2(V_k, V_k, b_k, \beta_j) = (X_{j,k} - U_k^T\cdot V_j - b_k - \beta_j)^2 + \lambda\cdot(\beta_u + b_i + ||U_u||^2 + ||V_i||^2) \quad (4)$$

It is easy to see that

$$L_2 = \frac{1}{2}\cdot\sum_{(i,j)\in\Omega} f_2(U_k, V_j, \beta_k, b_j) \quad (5)$$

Minimizing the mathematical Objective

We used the same optimizers as as for the baseline model:

SGD

```
1. Initialize $V$ and $U$
2. For $k$ to $K$:
    1. iterate through all pairs rated (user, item) pairs, For the $j$ -th user and $k$ -th item:
        • apply both updates simultaneously:
            – $V_k \leftarrow V_k - \alpha\cdot\frac{\nabla f_2(V_j,U_k,\beta_j,b_k)}{\nabla V_k}$
            – $U_j \leftarrow U_j - \alpha\cdot\frac{\nabla f_2(V_j,U_k,\beta_j,b_k)}{\nabla U_j}$
            – $b_j \leftarrow b_j - \alpha\cdot\frac{\nabla f_2(V_j,U_k,\beta_j,b_k)}{\nabla b_k}$
            – $\beta_j \leftarrow \beta_j - \alpha\cdot\frac{\nabla f_2(V_j,U_k,\beta_j,b_k)}{\nabla \beta_j}$
            – Each update for a given parameter is carried out while fixing the others.
```

BCD

```
1. Initialize $V$ and $U$
2. For $k$ to $K$:
    1. iterate through all pairs rated (user, item) pairs, For the $j$ -th user and $k$ -th item:
        1. update $V$:
            • $V_k \leftarrow V_k - \alpha \cdot \frac{\nabla f_2(V_j, U_k, \beta_j, b_k)}{\nabla V_k}$
    2. iterate through all pairs rated (user, item) pairs, For the $j$ -th user and $k$ -th item:
        1. update $U$:
            • $U_j \leftarrow U_j - \alpha \cdot \frac{\nabla f_2(V_j, U_k, \beta_j, b_k)}{\nabla U_j}$
    3. iterate through all pairs rated (user, item) pairs, For the $j$ -th user and $k$ -th item:
        1. udpate $b$:
            • $b_j \leftarrow b_j - \alpha \cdot \frac{\nabla f_2(V_j, U_k, \beta_j, b_k)}{\nabla b_k}$
    4. iterate through all pairs rated (user, item) pairs, For the $j$ -th user and $k$ -th item:
        1. update $\beta$:
            • $\beta_j \leftarrow \beta_j - \alpha \cdot \frac{\nabla f_2(V_j, U_k, \beta_j, b_k)}{\nabla \beta_j}$
```

Prediction

Prediction $\hat{y}$ for a (u, i) pair is made as follows

$$\hat{y} = \begin{cases} \mu + \beta_u + b_i + UV^T & \text{if } (u, i) \in \Omega \\ \mu + \beta_u & \text{if } u \in S_u \text{ and } i \notin S_i \\ \mu + b_i & \text{if } u \notin S_u \text{ and } i \in S_u \\ \mu & \text{otherwise} \end{cases}$$

where μ is the average prediction as is equal to

$$\mu = \frac{1}{\sum_{u=1}^{m} \sum_{i=1}^{n} W_{ij}} \sum_{u=1}^{m} \sum_{i=1}^{n} W_{ij} X_{ij}$$

Mathematical Derivations

The mathematical function f_2 is fairly similar to f_1. The gradients for f_2 with respect to U_k and V_i are the same f_1. As for the β_k and b_j,

we proceed as follows:

Since the term $X_{j,k} - U_k^T \cdot V_j - \mu - \beta_u - b_i$

$$\frac{\nabla f_2(V_j, U_k, \beta_k, b_j)}{\nabla b_j} = -((X_{j,k} - U_k^T \cdot V_j - \mu - \beta_u - b_i) - \lambda \cdot b_j)$$

$$\frac{\nabla f_2(V_j, U_k, \beta_k, b_j)}{\nabla \beta_k} = -((X_{j,k} - U_k^T \cdot V_j - \mu - \beta_u - b_i) - \lambda \cdot \beta_k)$$

The updates can be expressed as:

- $V_i \leftarrow V_i + \alpha \left(e_{u,i} \cdot U_u - \lambda V_i\right)$
- $U_u \leftarrow U_u + \alpha \left(e_{u,i} \cdot V_i - \lambda U_u\right)$

By defining $e_{u,i}$ as

$$e_{u,i} = \left(X - (\mu + \beta_u + b_i + UV^T)\right)_{u,i}$$

The updates can be expressed as:

- $b_i \leftarrow b_i + \alpha \left(e_{u,i} - \lambda b_i\right)$
- $\beta_u \leftarrow \beta_u + \alpha \left(e_{u,i} - \lambda \beta_u\right)$
- $V_i \leftarrow V_i + \alpha \left(e_{u,i} \cdot U_u - \lambda V_i\right)$
- $U_u \leftarrow U_u + \alpha \left(e_{u,i} \cdot V_i - \lambda U_u\right)$

Results

All the results were achieved using minibatch size of 128, regularization $\lambda = 0.02$ on across 20 iterations (epochs) for every model.

Final takeaways:

- Block Coordinate Descent was show to be more stable during training than SGD.
- The choice of correct step size (leaning rate) is vitally important to solve the task.
- Both models achieve comparable results of RMSE ≈ 0.86.
- Number of factors beyond 4 does not substantially decrease RMSE.

Final RMSE using SGD

Final RMSE using SGD

Delta loss

Delta MAE

Delta RMSE

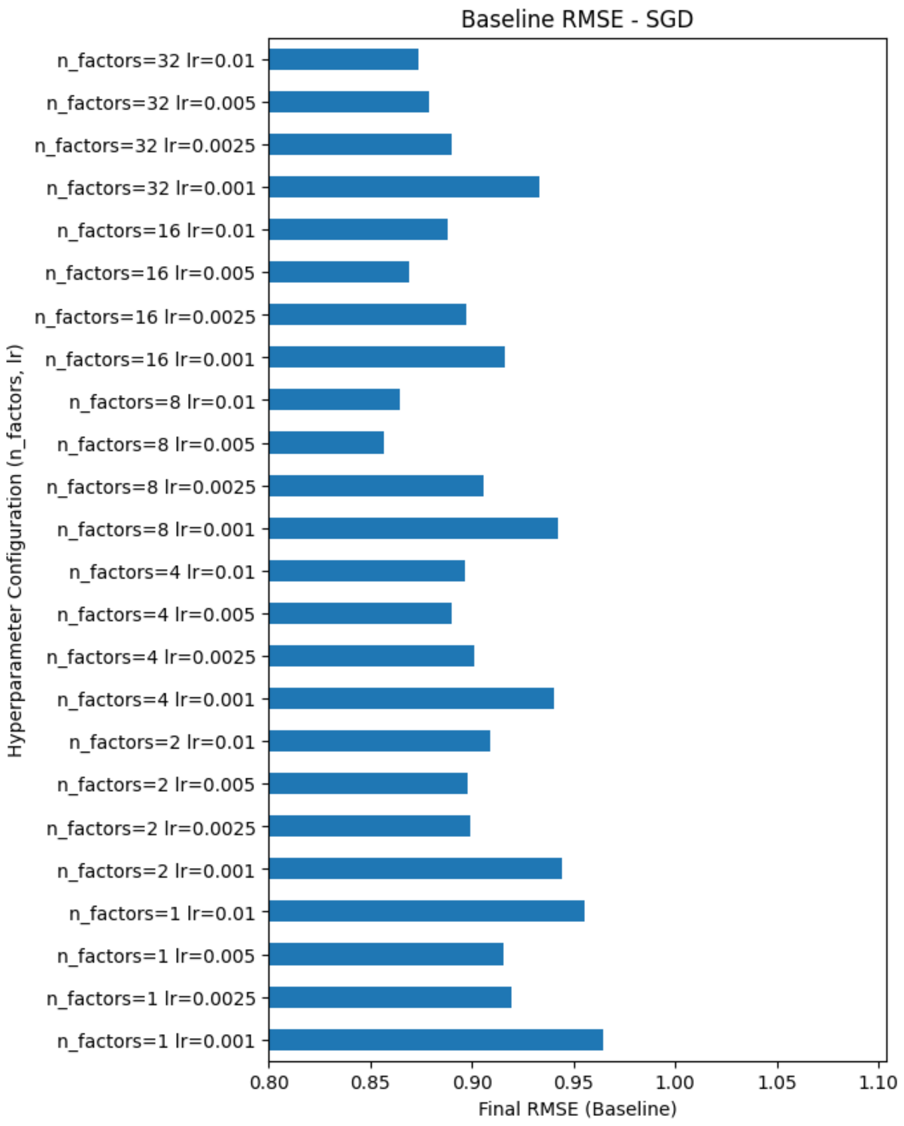

Figure 1: alt text

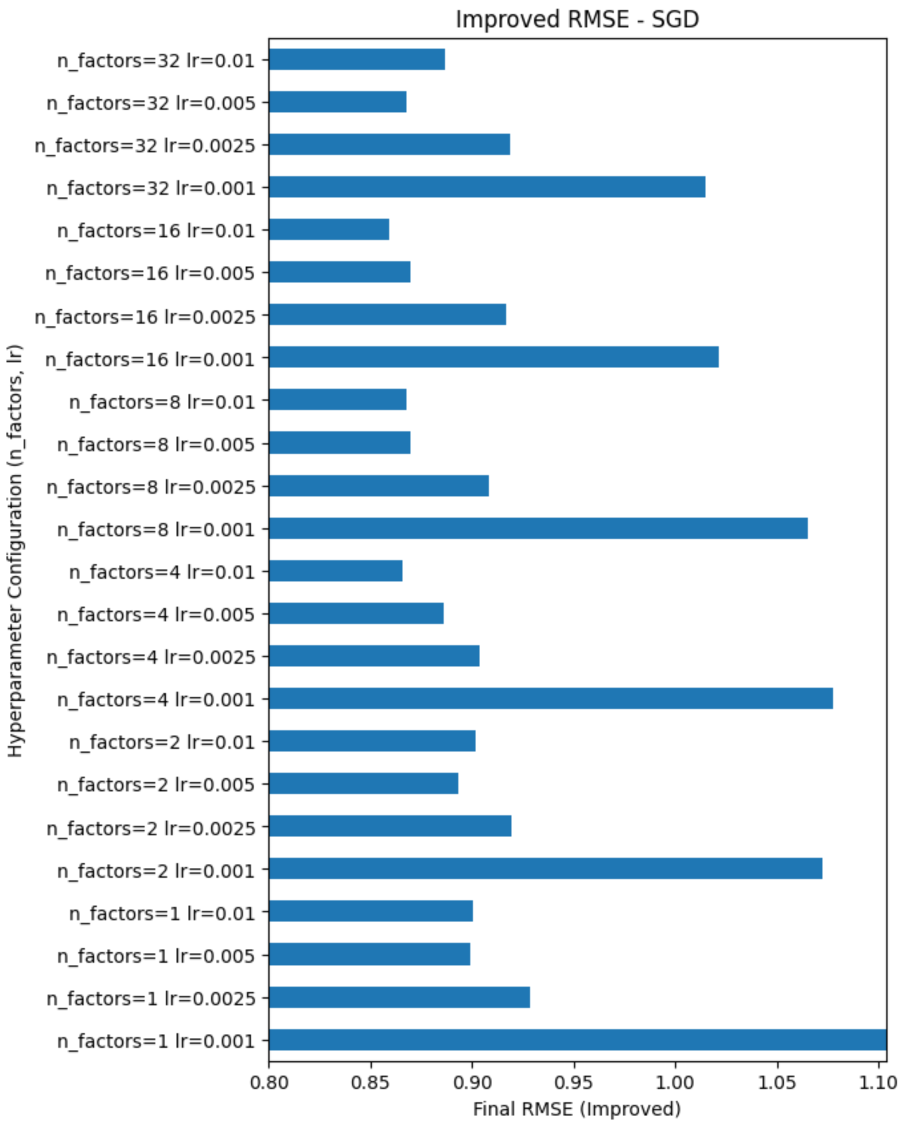

Figure 2: alt text

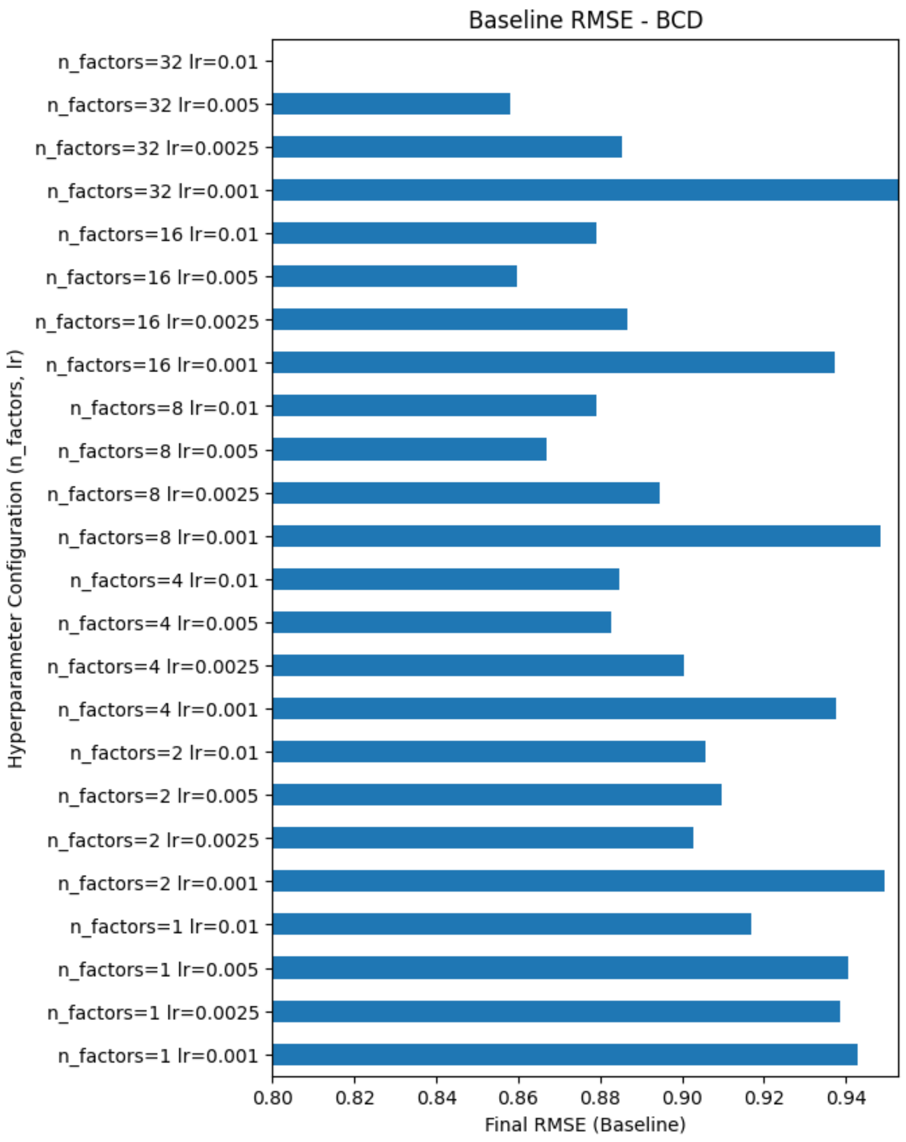

Figure 3: alt text

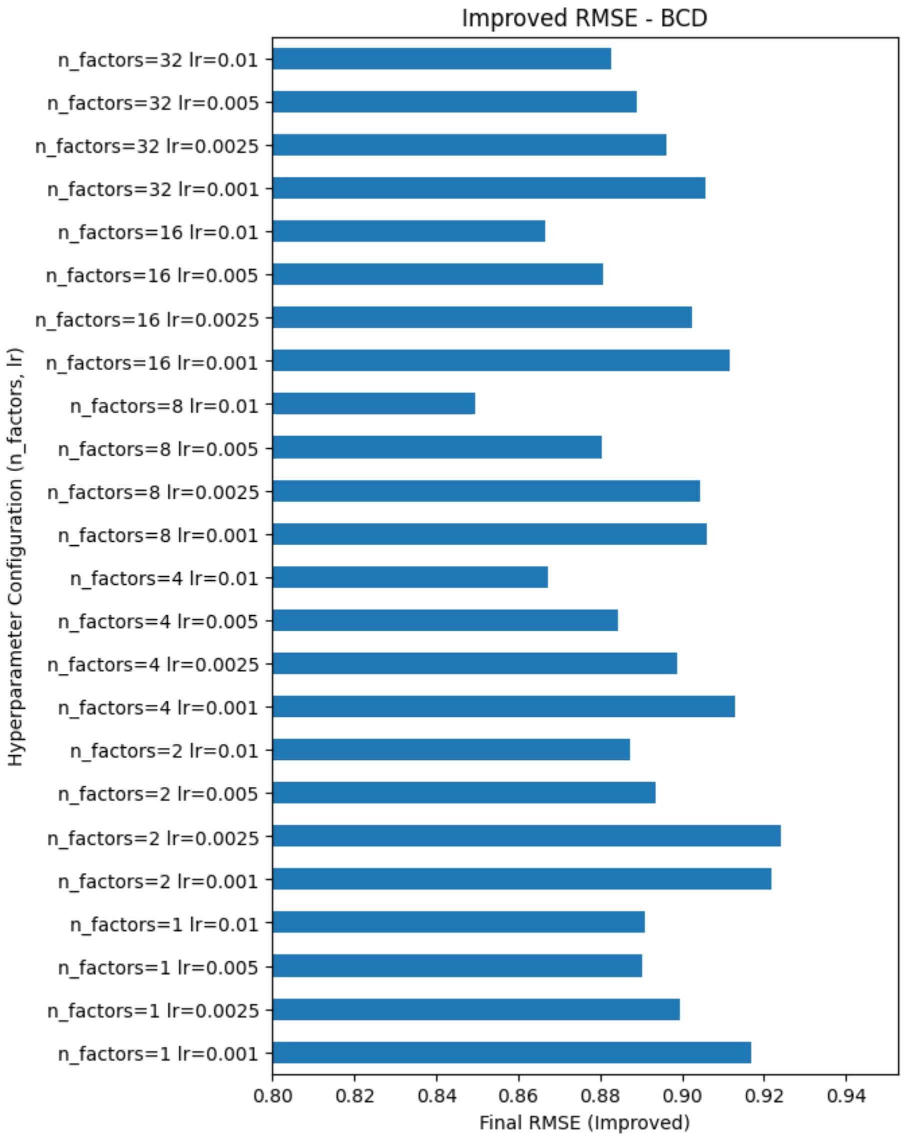

Figure 4: alt text

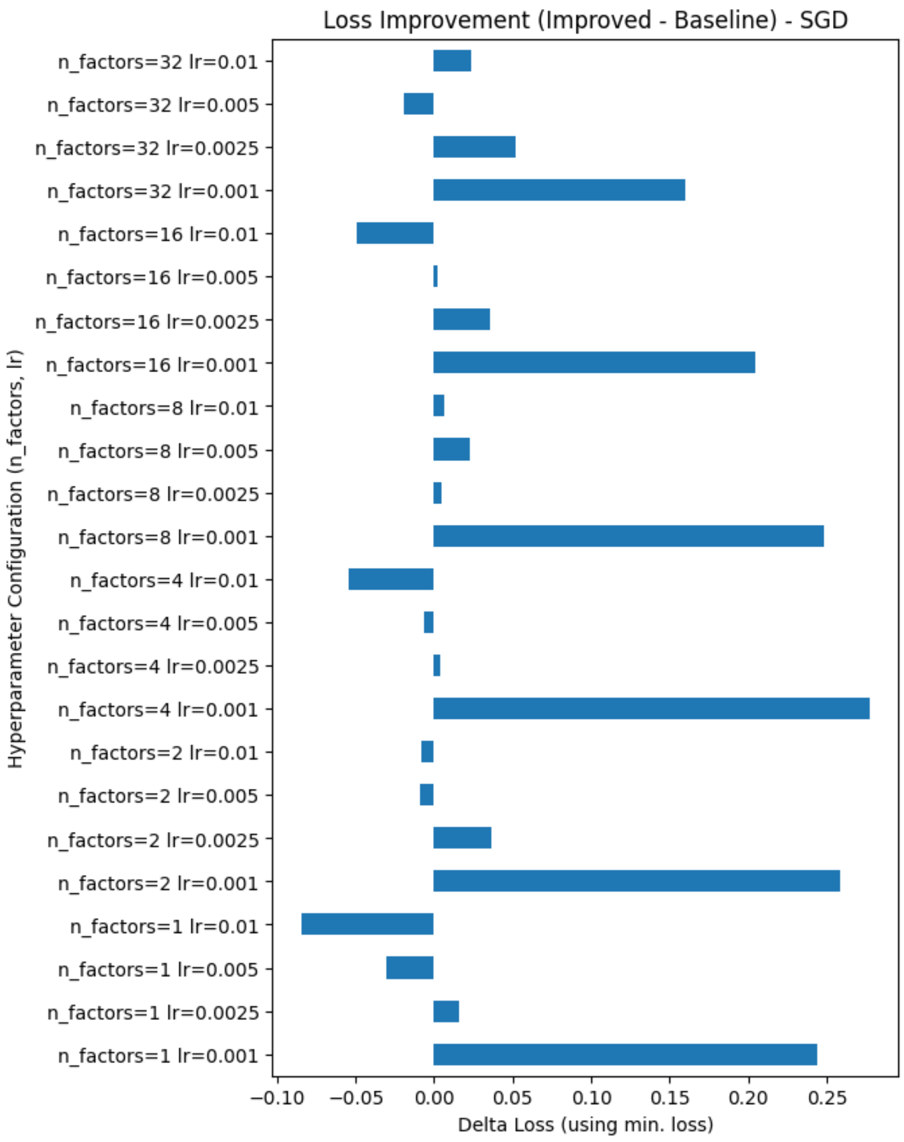

Figure 5: alt text

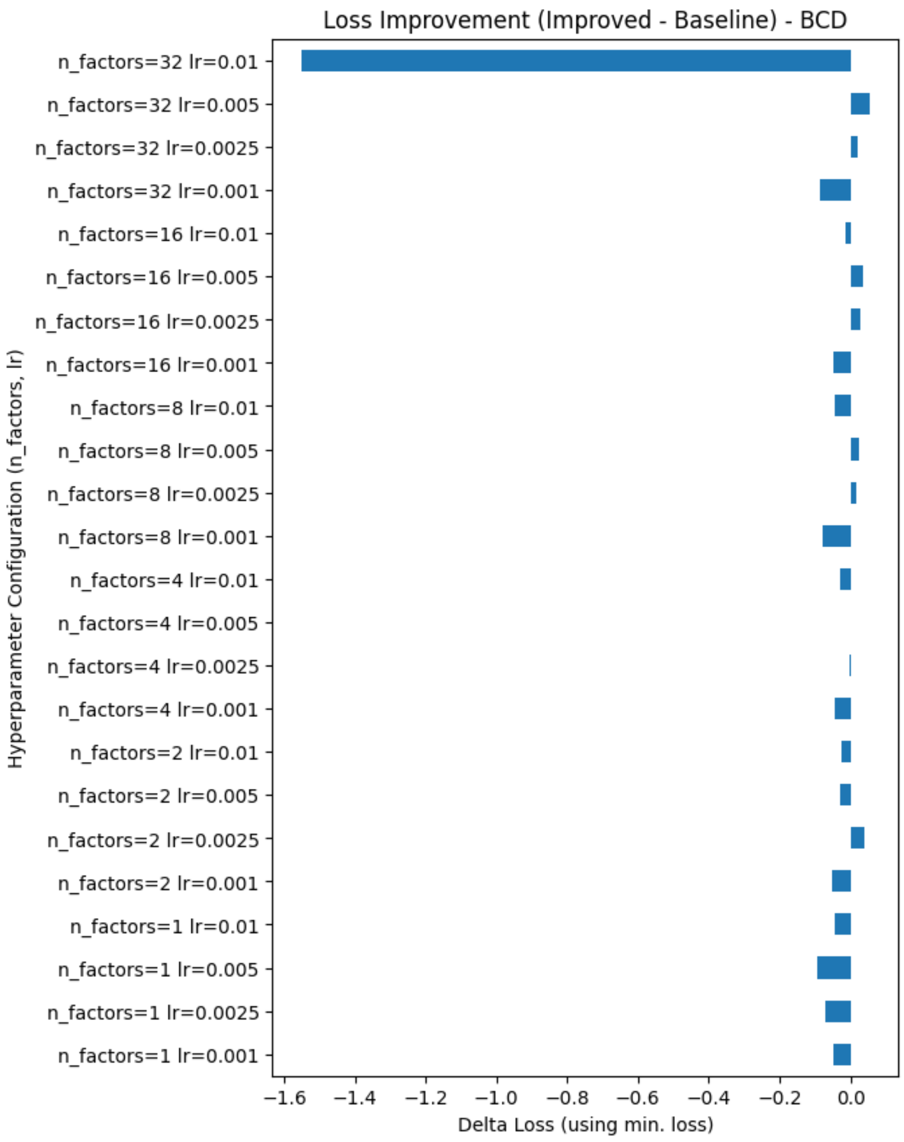

Figure 6: alt text

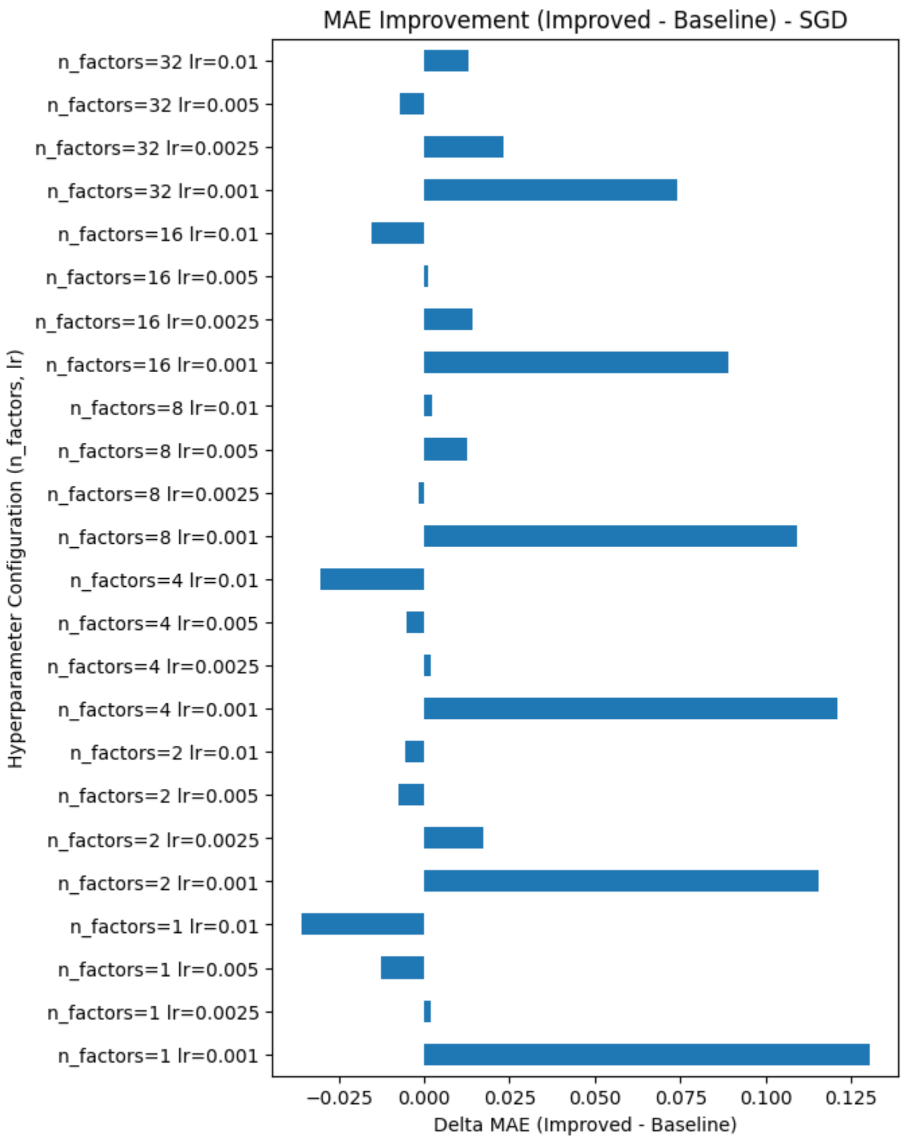

Figure 7: alt text

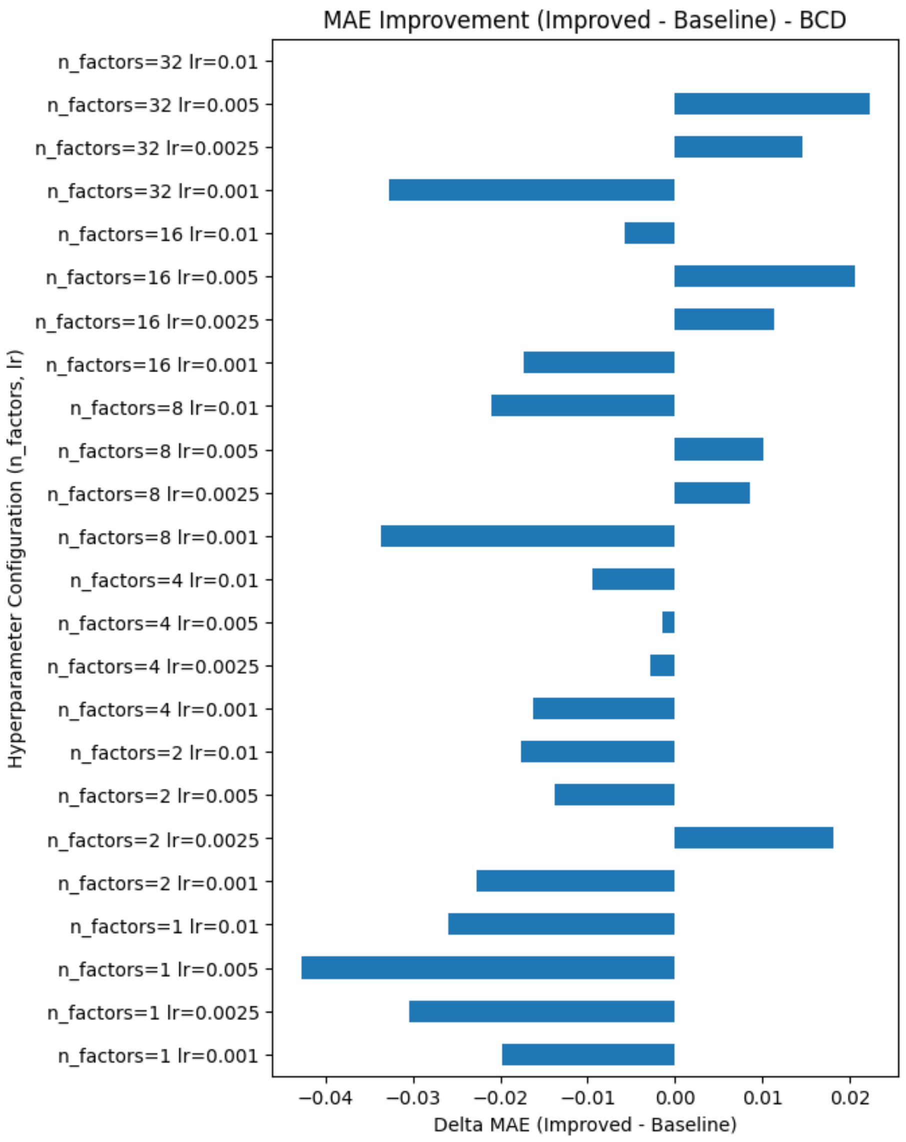

Figure 8: alt text

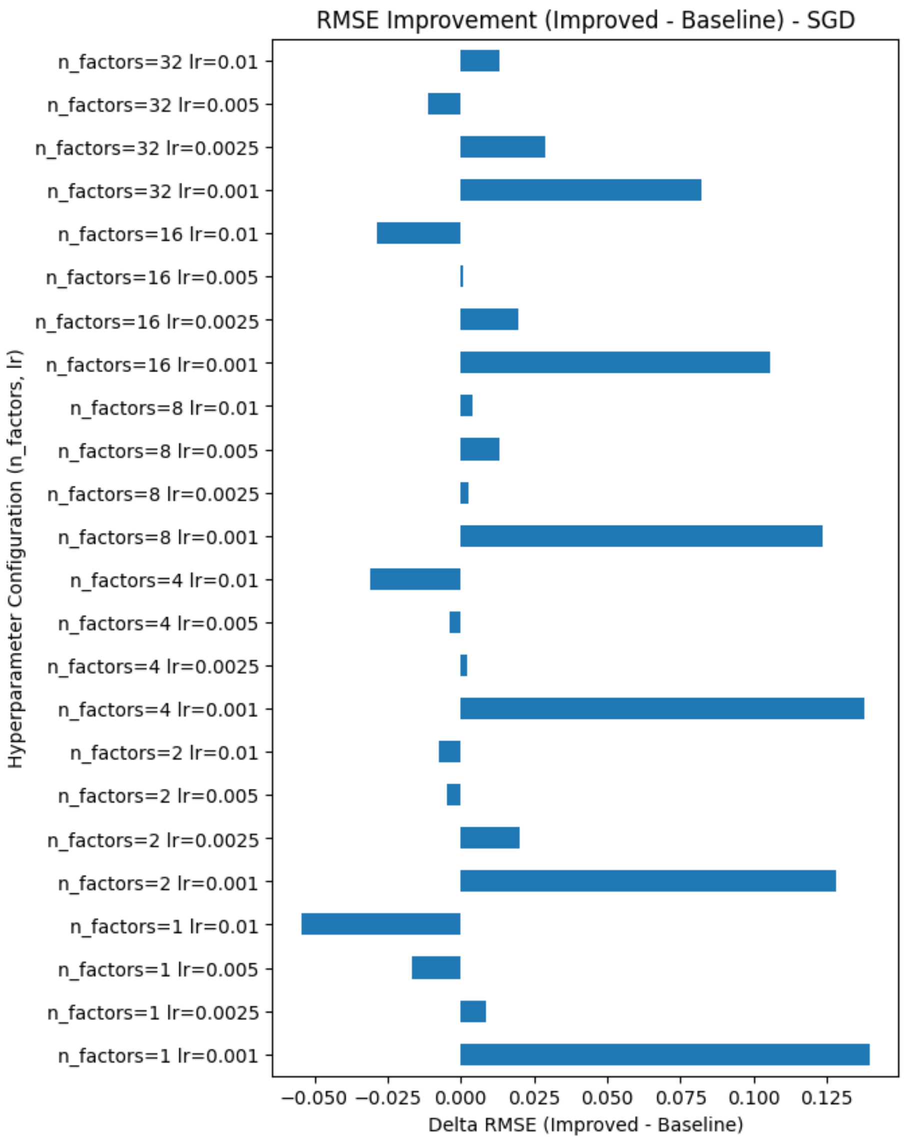

Figure 9: alt text

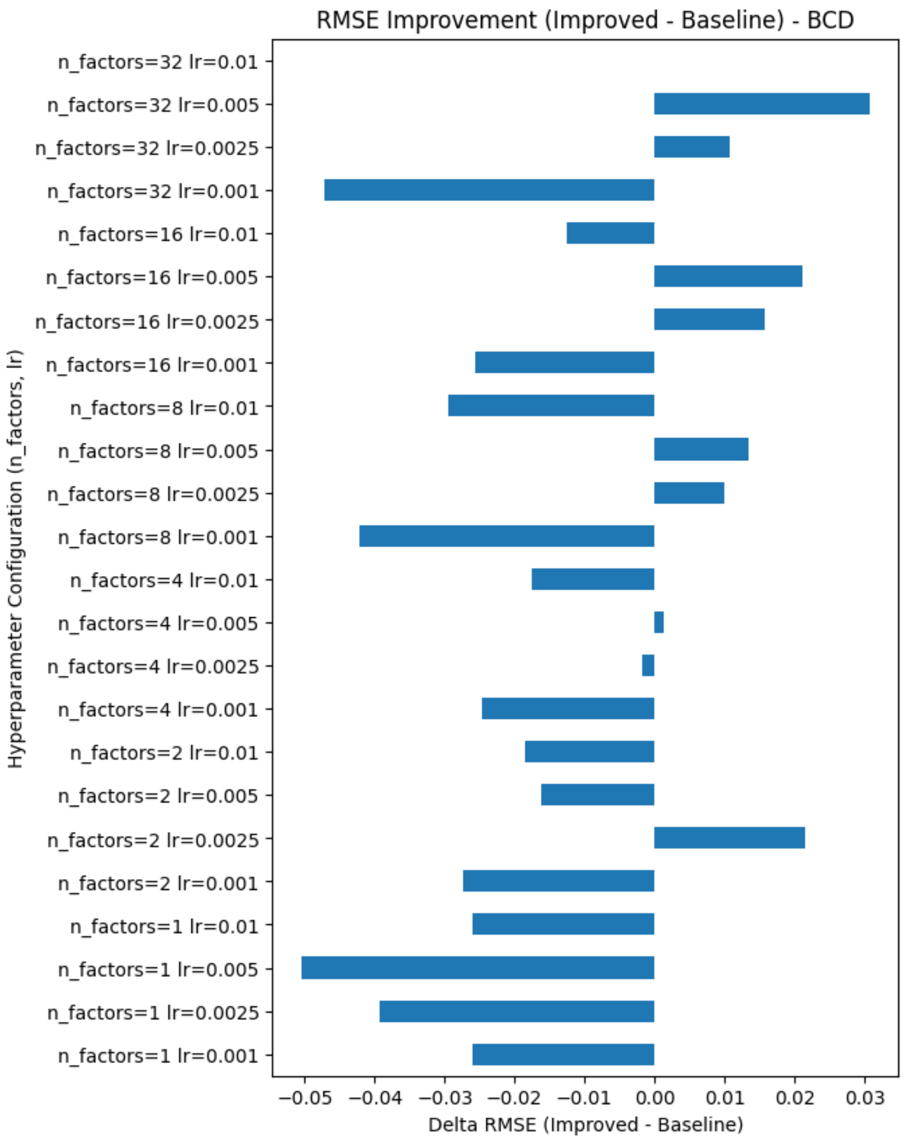

Figure 10: alt text