**10M Data Results Summary**

**Data**

10M MovieLens dataset here: <https://files.grouplens.org/datasets/movielens/ml-10m-README.html>

Unlike 1M or 100K, no user demographic information is included (just users, movies, and ratings).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Model (10M Dataset)** | **FC Layer Units** | **Dropout Rate** | **Attention Heads** | **Transf. Blocks** | **Learning Rate** | **Batch Size** | **RMSE** |
| Baseline BST | [256, 128] | 0.1 | 3 | 1 | 0.01 | 256 | 0.876 |
| Tuned BST1 | [128] | 0.1 | 8 | 2 | 0.01 | 512 | 0.905 |
| Tuned BST2 | [128] | 0.5 | 8 | 2 | 0.02 | 512 | 0.918 |
| Tuned BST3 | [256, 128] | 0.4 | 10 | 3 | 0.02 | 512 | 0.889 |
| Tuned BST4 | [256, 128] | 0.4 | 3 | 2 | 0.02 | 256 | 0.879 |
| Tuned BST5\* | [256, 128] | 0.3 | 8 | 2 | 0.02 | 256 | 0.863 |
| Tuned BST6\*\* | [256, 128] | 0.3 | 8 | 2 | 0.02 | 256 | 0.860 |

*\*Tuned BST 5 Also extended sequence length from 4 to 6*

*\*\*Tuned BST 6 Kept sequence length 6 and added userid and movie feature input embeddings*

**Overview**

Baseline BST: 0.876 RMSE

Tuned BST 1: 0.905 RMSE

Tuned BST 2: 0.918 RMSE

Tuned BST 3: 0.889 RMSE

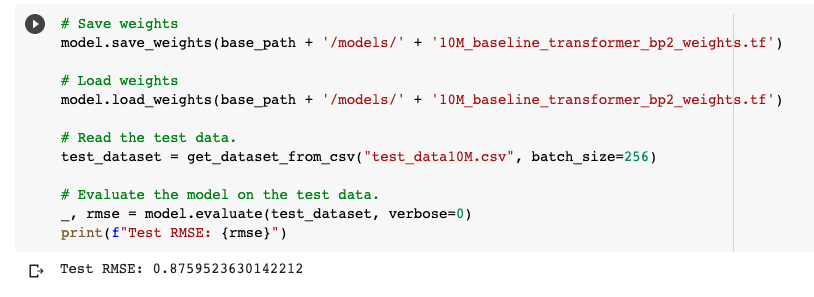
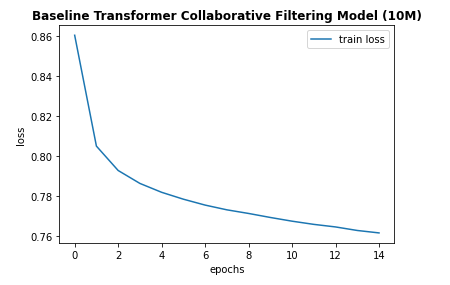
Tuned BST 4: 0.879 RMSE

Tuned BST 5: 0.863 RMSE (best model, structure plot included)

Tuned BST 6: 0.860 RMSE\* (new best model, same as 5 + user ID and movie features embedding)

For each, I include the training plot, screenshot of results, and a short description of parameters.

**Baseline BST:**



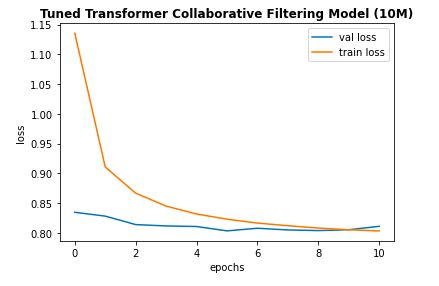
Training our baseline transformer model on the 10M data (~900’000 left in training given 90-10 train-test split) resulted in a 0.876 RMSE evaluated on the test set. This was a significant improvement from our 1M results, even without the user demographic information.

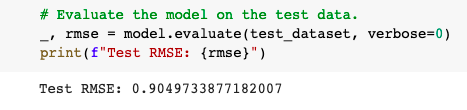
As before, our baseline transformer-based model has 1 transformer block, 3 attention heads, a dropout rate of 0.1, and 2 FC layers with 256 and 128 neurons respectively.

**Tuned BST:**

**1.**

Applying our previously tuned transformer-based model (with 2 transformer blocks and 8 heads, and only 1 FC layer) to the 10M dataset gave a worse test error (RMSE of 0.905) compared to the baseline model. This means that the predicted ratings were farther away from the actual ratings given by users.

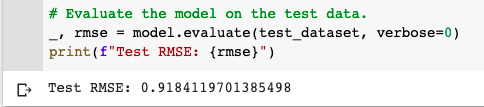
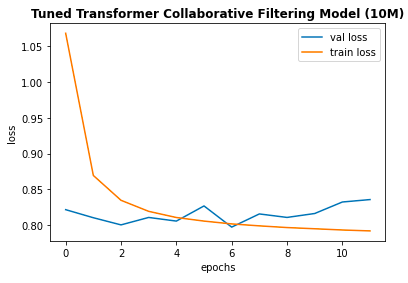




For this model, we used a learning rate of 0.01 with Adagrad, and a batch size of 512.

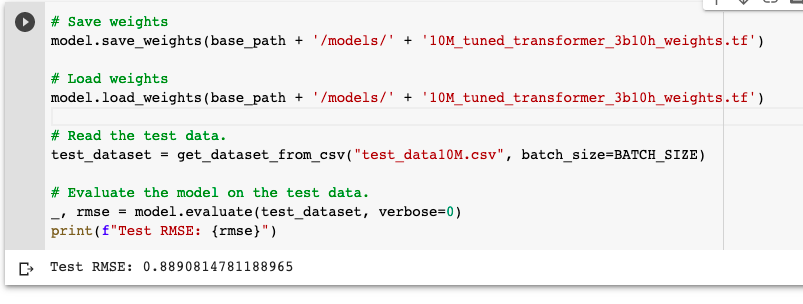
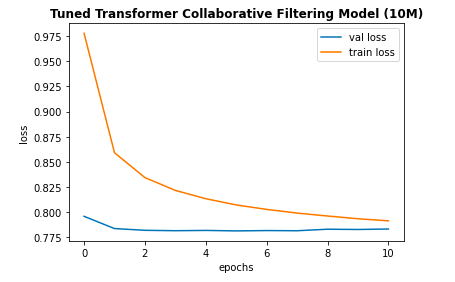
**2.**

Increasing the learning rate to 0.02 and the drop-out rate to 0.5 gave less smooth training and a worse RMSE of 0.918:



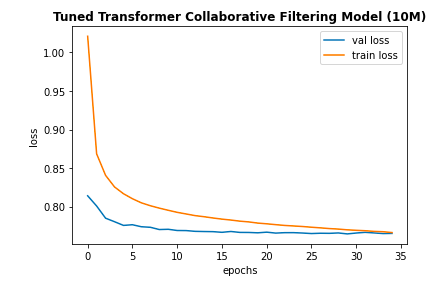
**3.**

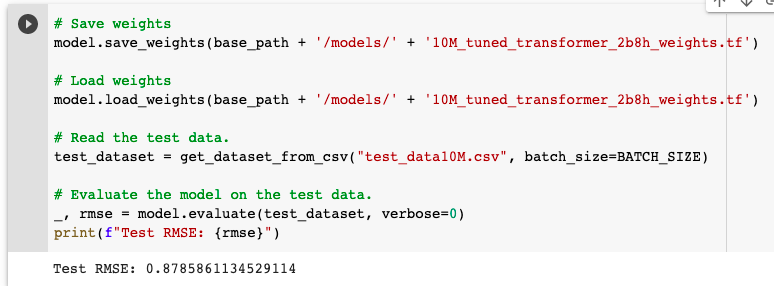
Restoring the 2nd FC layer and Increasing the number of blocks to 3 and heads to 10 improved on the tuned transformer model, resulting in a test RMSE of 0.889. However, the baseline BST model still outperformed this tuned one.



**4.**

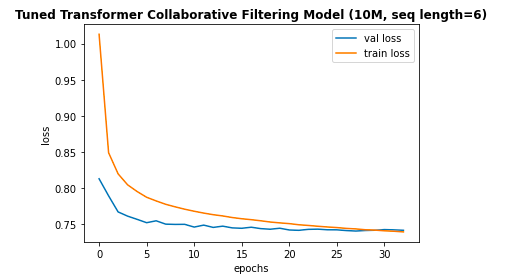
Reducing the number of blocks to 2 and heads to 3, while keeping 2 fully-connected layers and increasing dropout to 0.4 improved the test RMSE to 0.879, slightly above the baseline performance.

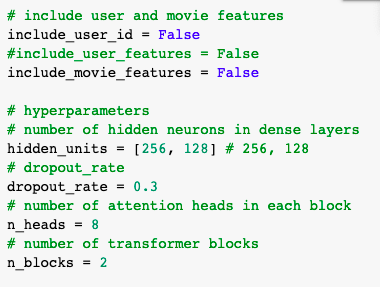




**5. 2nd best model**

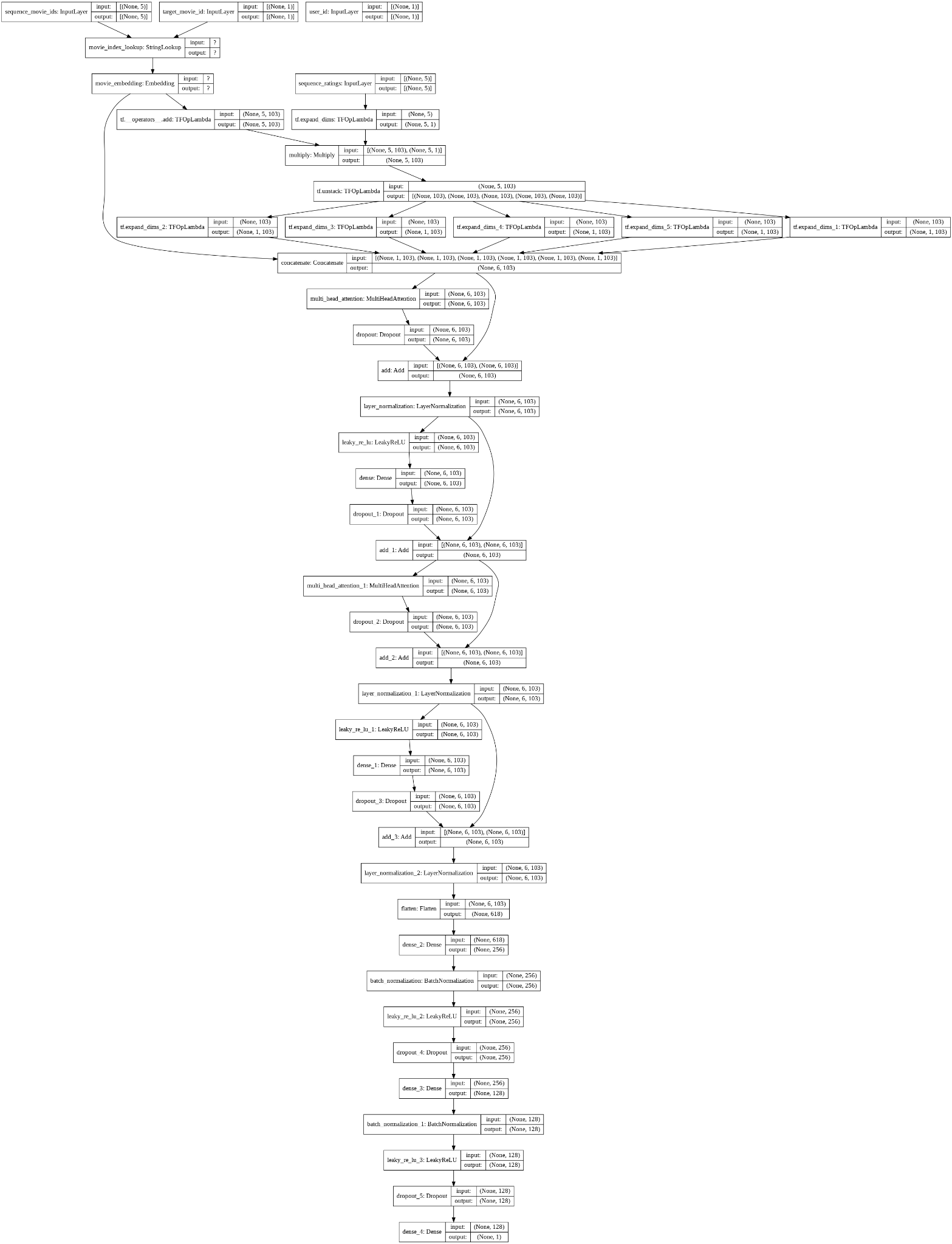
However, by extending the length of the movie and rating sequences from 4 to 6, we boosted our model predictions, resulting in our lowest test RMSE, of 0.863.

****

****

This tuned transformer-based model did not encode movie features or user IDs into embeddings. The model had 2 fully-connected layers with 256 and 128 neurons, a dropout rate of 0.3, 2 transformer blocks and 8 heads. It was trained with a batch size of 256. Thus, it appears that using slightly longer sequences improves upon our model’s ability to predict ratings, given our positional embedding to capture sequence order and the ability of multi-headed attention to capture both short and longer-term dependencies.

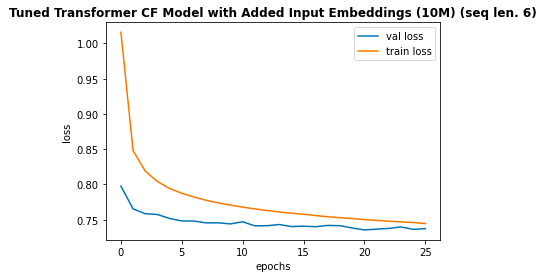
Best tuned transformer-based model structure:

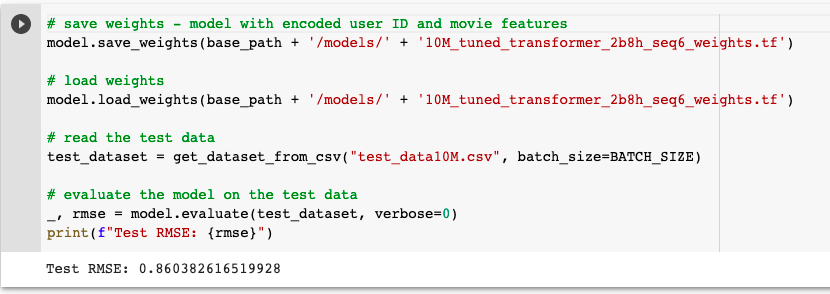


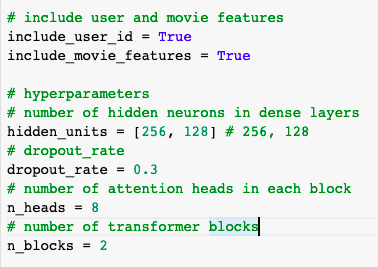
**6. Best model - with user ID and movie feature embeddings**

Same as model 5 but with added input embeddings

* Include\_user\_id = True
* Include\_movie\_features = True

****





Adding the user ID and movie features into the input encoder only slightly improved the RMSE, from 0.863 to 0.0860. Thus, the additional embeddings did not improve the predictions significantly.

