[Blog Post Link](https://roma-coffin.medium.com/predicting-movie-recommendations-by-leveraging-deep-learning-and-movielens-data-17661ba4c3b1)

[Github Link](https://github.com/annieptba/DATA2040_Final_Project-YARD)

MovieLens Recommendation System

1. Original Project

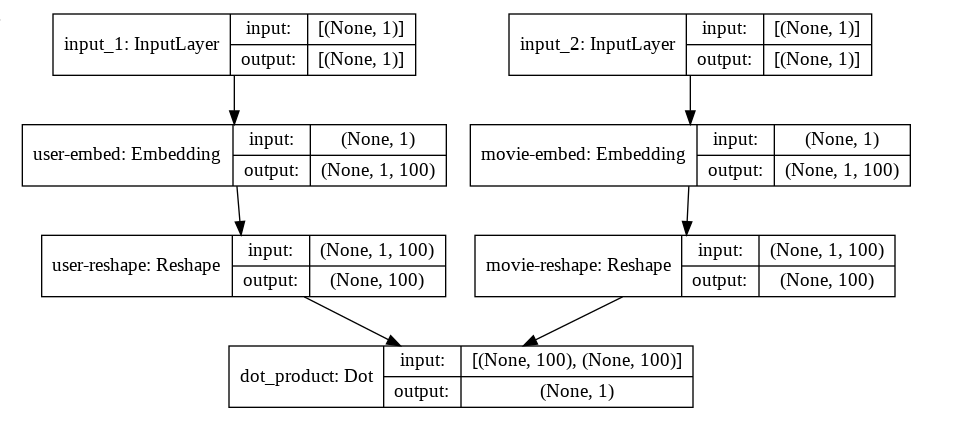
* Replicate project from James Le, a data scientist on MovieLens 1M Dataset
* Jame’s Github Repo: <https://github.com/khanhnamle1994/movielens>
  + Implemented 4 methods: content-based, collaborative filtering, SVD, deep learning
  + Can focus on deep learning, ignore the other 3 methods (may revisit if we have time)
* Jame’s Medium blog post: <https://le-james94.medium.com/the-4-recommendation-engines-that-can-predict-your-movie-tastes-bbec857b8223>
  + Best RMSE value of *0.8616* on epoch 17 (from original code)
* Replicated Implementations of Jame’s work from other people: https://github.com/bradleypallen/keras-movielens-cf

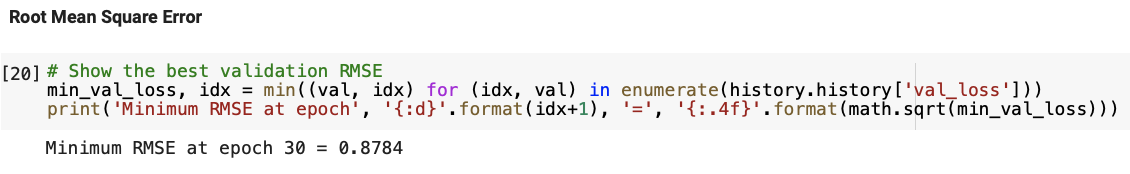
1. Final Project Requirements and Outline

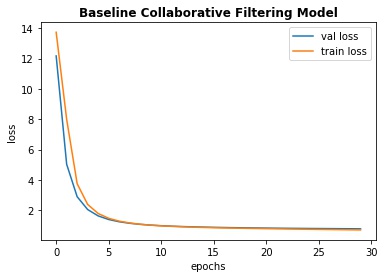
* Algorithm. Pick a problem and develop or improve an algorithm for solving it. Your starting baseline should be a correctly implemented state of the art model.
  + Suggestions: adding more Dense Layers, using Transfer Learning, adding regularization, changing the learning rate schedules, changing the optimizer, etc (Boqing)
  + Includes code to re-implement baseline from keras 2 to keras 2.4.4: <https://medium.com/@jdwittenauer/deep-learning-with-keras-recommender-systems-e7b99cb29929>
  + <https://www.onceupondata.com/2019/02/10/nn-collaborative-filtering/>
* Application. pick a Deep Learning application and explore how best to solve it using different models, or focusing on improving one model
  + Suggestions the author currently implements a neural network, we need to do a different type of DL model, such as transformer, RNN, CNN, etc (Boqing)
  + Variational autoencoder: <https://github.com/noveens/svae_cf> (Cang)
  + Transformer: Kera’s transformer implementation on the MovieLens dataset: <https://keras.io/examples/structured_data/movielens_recommendations_transformers/> (Cang, Sayan, Boqing)
    - Test Score: Mean Absolute Error (MAE) ~ 0.7
    - Transformers run faster on TPU
* Baseline Model implementation translate from Jame’s code to Keras 2.4.3 code, class implementation
  + <https://keras.io/examples/structured_data/collaborative_filtering_movielens/>
  + <https://colab.research.google.com/github/keras-team/keras-io/blob/master/examples/structured_data/ipynb/collaborative_filtering_movielens.ipynb>

1. Baseline Deep Learning Model

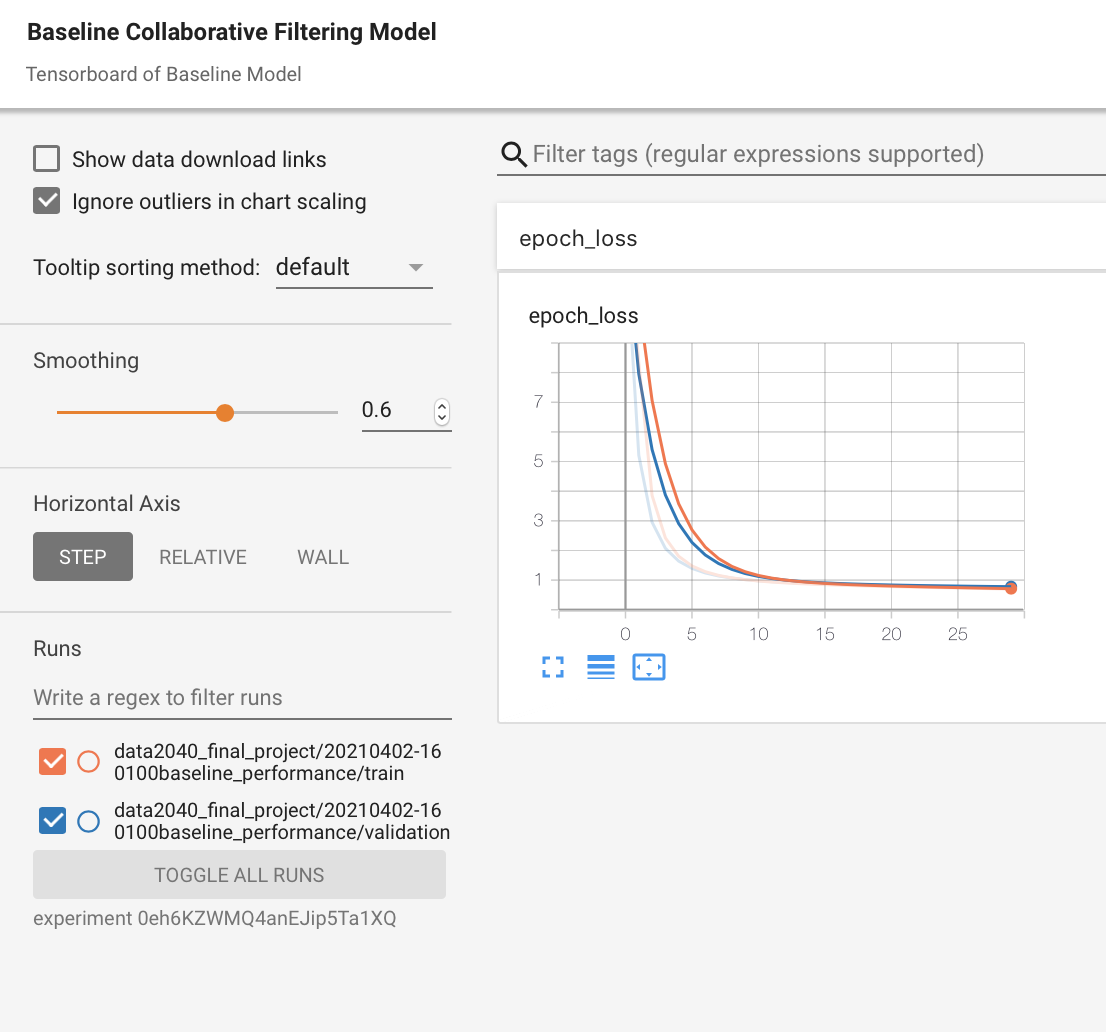
* Model Structure



* RMSE: 
* Plt plot:



* Tensorboard:
  + <https://tensorboard.dev/experiment/0eh6KZWMQ4anEJip5Ta1XQ/>



1. Ratings Predictions and Movies Recommendation Using Trained Baseline Model

* Movie Ratings Prediction of User with ID = 2000



* Recommended Movies for User with ID = 2000 based on Predicted Ratings



1. Questions to ask TAs/ Instructors

* Customize paths for Tensorboard