

CS 330 Autumn 2021 Homework 1 Data Processing and Black-Box Meta-Learning Due Wednesday October 6, 11:59 PM PST

SUNetID: tminh
Name: Minh Tran
Collaborators: N/A

October 1, 2021

Abstract

The document contains solutions for given problems including implementation of a multi-task model which predicts user-movie interactions and the potential score an user will rate a particular movie and demonstration of experiments with different settings to evaluate the effect of embedding sharing to the training performance of multi-task model.

1 Problem 1: Data Processing for Few-Shot Classification

Implementation of multi-task more will require embedding vectors of users and items, as well as the embedding vector of biases. For sharing embedding, we will use the same embedding vectors to do two calculations in forward function, first is the interaction probability for an user and a movie, second is the score that the user would assign to a movie. For separated embedding, we need to create another set of embedding vectors for users, items, biases and use this set for either task. Details will be found in "models.py".

2 Problem 2: Memory Augmented Neural Networks (MANN)

The bug is the positions of "predictions" and "score" variables. The factorization loss function takes positive and negative predictions instead of score and negative prediction. In the other hand, the regression loss function takes observed ratings and predicted ratings instead of positive prediction. We just need

to switch the position of these two terms to get the right result. Details will be found in "multitask.py".

3 Experiments Write-up

3.1 Experiment 1: Shared representations and non-evenly divided task weights

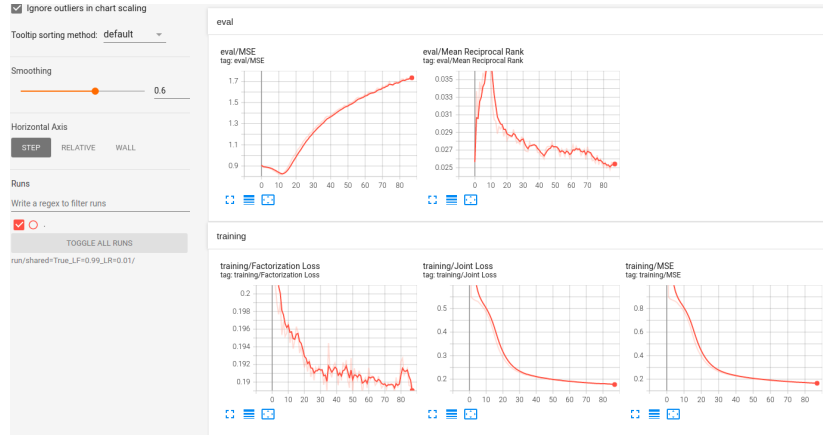


Figure 1: Shared representations and task weights $\lambda_F = 0.99, \lambda_R = 0.01$

3.2 Experiment 2: Shared representations and evenly divided task weights

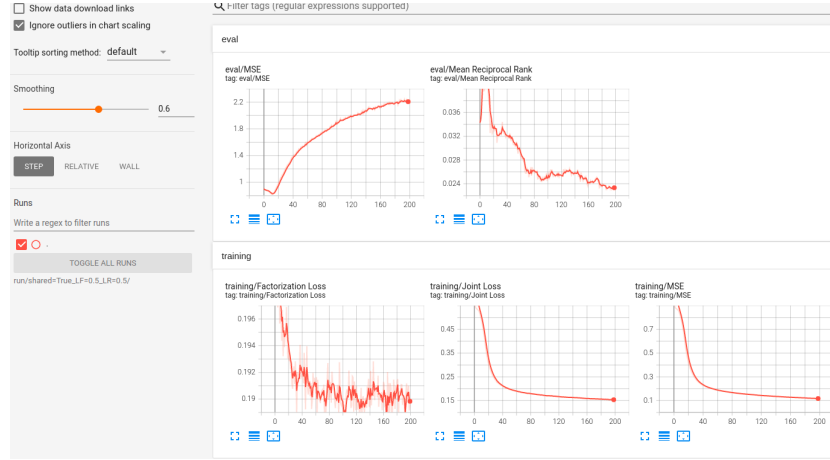


Figure 2: Shared representations and task weights $\lambda_F = 0.5, \lambda_R = 0.5$

3.3 Experiment 3: Separated representations and evenly divided task weights

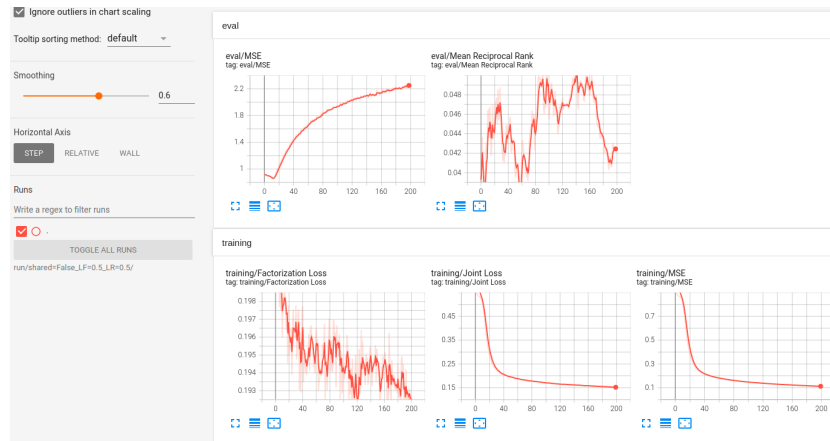


Figure 3: Separate representations and task weights $\lambda_F = 0.5, \lambda_R = 0.5$

4 Problem 3: Analysis

5 Problem 4: Experimentation

Question 1:

From the experiment 2 and 3 where we evenly divided the weights for each loss, we observed that for the training process, both factorization loss and regression loss decreased pretty well to $loss_F = 0.193$ and $loss_R = 0.1$. For testing, given that the expected MSE is below zero and the expected MRR is from 0.11 to 0.13, there could be a bug with my training process. The eval MSE could indicate overfitting but the MRR should be higher, however it fluctuated around 0.45 and peaked at only 0.5. Strangely, in experiment two which has shared embedding, MRR dropped significantly to 0.024 which is not expected. In reality with a correct setting, I would expect shared embedding model would converge slower and gave more fluctuation in joint loss, but the performance of each task will be better since shared embedding model could increase the generalization, and these two tasks are highly related.

Question 2:

From the experiment 1 and 2 where we have the weights for factorization loss overwhelm the ones of regression loss, we concluded that the way we choose the weight proportion of loss is important for the training process. For experiment 1, since the model was optimized heavily based on factorization loss, its factorization loss is smoother while the regression loss decreases slower compare to itself in experiment 2. Joint losses are pretty stable around 0.15 for both experiment. Again, the MMR in both experiments seem incorrect, so in a correct setting of experiment 1, I would expect the performance of the matrix factorization task MMR is as better as those of model 2, while MSE of regression task in experiment 1 will be worse than those of model 2 since its loss weight is too small, thus the model couldn't learn much from it.