

HW5: Movie Recommendation

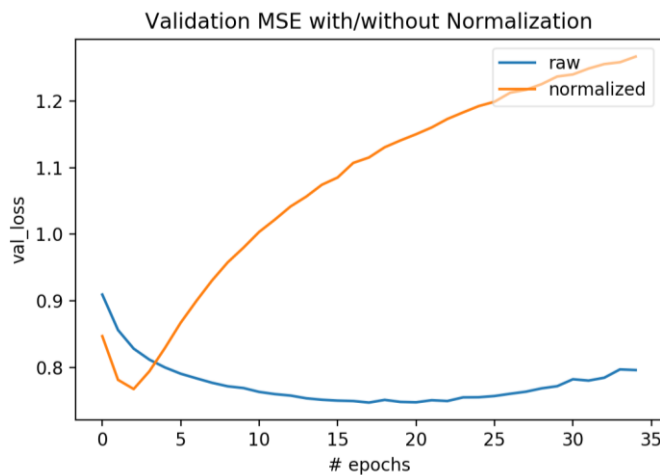
學號: R06922030 系級: 資工碩一 姓名: 傅敏桓

實驗設定

實作矩陣分解 (Matrix Factorization) 時，以 Keras 嵌入層分別將使用者編號和電影編號映射到 128 維的嵌入向量，經過向量內積運算加上 bias 後輸出。訓練時隨機取訓練資料的 10% 作為驗證集，使用 RMSProp 作為 optimizer (學習率設為 0.001)、以 MSE 作為損失函數，訓練 35 個 epoch 取在驗證集上誤差最小的模型參數。

1. (1%) 請比較有無 normalize (rating) 的差別。並說明如何 normalize.

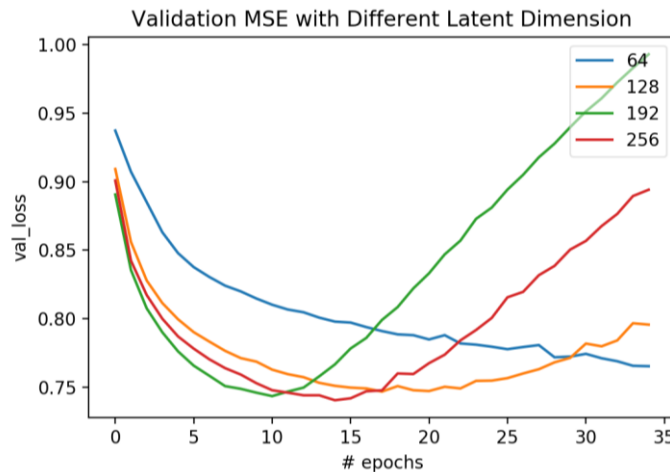
(Collaborator: 無)



以原始資料全部減去平均值 μ 、除以標準差 σ 後得到標準化的訓練資料，和沒有經過處理的資料，在相同的模型架構下訓練過程比較如上圖。由於處理後的資料和原始資料的 scale 不太一樣，直接比較 MSE 不太公平，所以有在標準化過的誤差值補乘 σ^2 回去。可以發現有經過標準化的資料在訓練時進步得比較快，但同時也較早開始發生 overfitting 的現象。

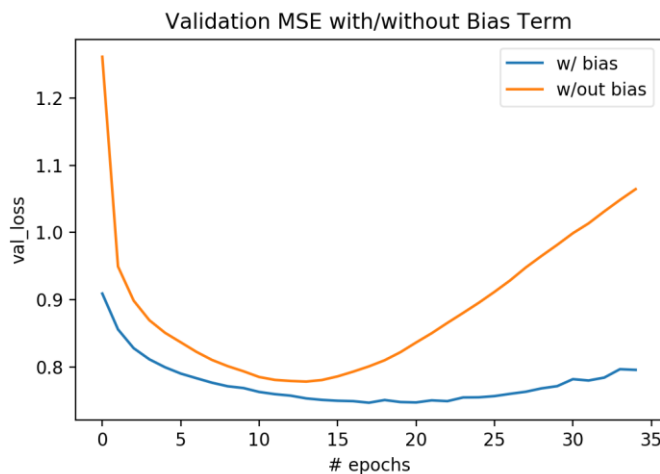
2. (1%) 比較不同的 latent dimension 的結果。(Collaborator: 無)

維持相同實驗設定分別改以 64 維、192 維及 256 維的嵌入層訓練模型，訓練過程的比較如下圖所示。可以發現維度較高的模型進步較快，也較快開始發生 overfitting 的現象，但不同模型之間誤差的最低點沒有相差太多。



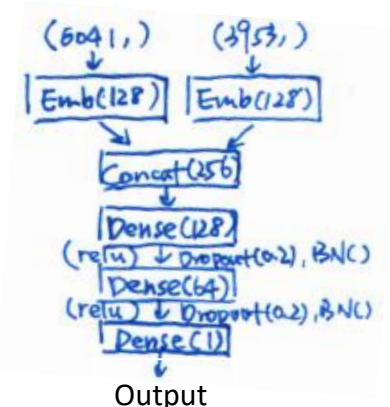
3. (1%)比較有無 bias 的結果。(Collaborator:無)

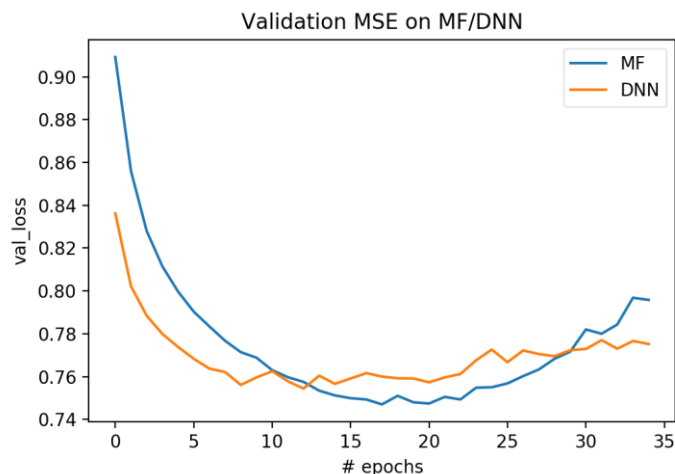
以相同的實驗設定分別訓練有加 bias 和沒加 bias 的模型，實驗結果如下圖。這裡加入的 bias 是將使用者的 bias 和電影的 bias 合在同一項。可以觀察到有加 bias 的模型整體表現都比沒加 bias 的模型好一些，也比較不容易發生 overfitting 的現象。



4. (1%)請試著用 DNN 來解決這個問題，並且說明實做的方法(方法不限)。並比較 MF 和 NN 的結果，討論結果的差異。(Collaborator:無)

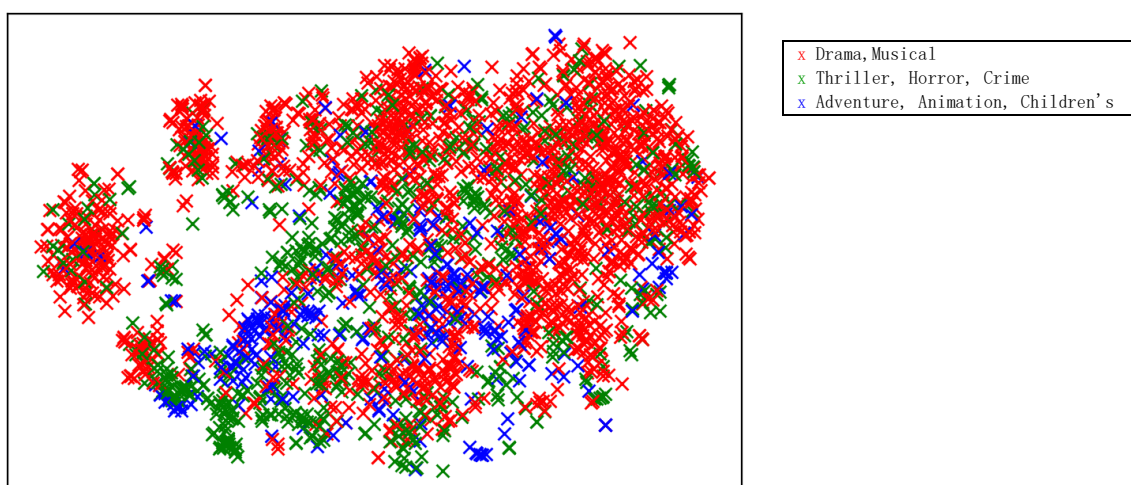
這次實作的 DNN 模型如右圖，採用和矩陣分解相同的嵌入向量層，在結合兩邊的嵌入向量時以向量連接的方式取代向量內積，再通過數層全連接層、批標準化層和 Dropout 層後輸出最後的預測值。訓練模型的方法一樣是使用 RMSProp 作為 optimizer (學習率=0.001)、以 MSE 作為損失函數 (也就是當作回歸問題來做)，訓練 35 個 epoch 取在驗證集上誤差最小的模型參數，和 MF 的訓練過程比較如下。





可以發現 DNN 在這個 task 上也有不錯的表現，訓練時較 MF 進步得稍微快一些，但整體看來並沒有顯著的進步，如果要得到更好的結果可能還需要嘗試不同的模型架構跟參數設定。

5. (1%)請試著將 movie 的 embedding 用 tsne 降維後，將 movie category 當作 label 來作圖。(Collaborator: 無)



本題比照助教的分類方法將電影按照分為三大類（有多個分類時則任選一個標籤），再透過 t-SNE 將電影的嵌入向量降維至二維空間，觀察資料的分布狀況；雖然可以看到一些資料點聚集的現象，但還沒有辦法把資料分開得很乾淨，可能在分類方法上要有所改進，或是也有可能我們的資料降到二維之後本來就不容易切割清楚。

(BONUS) (1%) 試著使用除了 rating 以外的 feature，並說明你的作法和結果，結果好壞不會影響評分。(Collaborator: 無)

我們嘗試將電影的類別資訊加入模型。先將各電影的類別表示成 18 維的向量（總共有 18 個電影類別），值為 1 就代表屬於該分類，反之亦同。將這個向量映射到一個 (0, 1) 之間的值（通過輸出只有一維的全連接層）作為電影的 bias 項，再訓練同樣的 MF 模型，過程如下圖所示。雖然整體結果沒有進步，但或許經過調整後有機會預測出更好的結果。

