

作業六 Movie Recommendation

學號：R06921002 系級：電機所碩一 姓名：張哲誠

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

Ans:

我分別測試了 50 個 epoch，embedding dimension 為 666，看看跑出來的結果差異

	Training mse	Validation mse
No normalized	0.7660	0.8957
Normalized	0.4581	0.6165

我做 normalized 的方式為先對 training 上的 Rating 分數取平均與標準差，並讓每筆資料都減掉平均與除以標準差，即完成 normalized。

兩個部分都出現 overfitting 的狀況，但有 normalized 的結果會比較快收斂，也有可能是因為做了 normalized，所有值都被限縮到 0~1 之間，產生整體 scale 上的變化。因此 MSE 看起來會小很多。但由於傳上 kaggle 分數相距不大，而且做 normalized 就會失去最後加 bias 的意義，因此最後決定不加 normalized。

2. (1%)比較不同的 embedding dimension 的結果

Ans:

我分別比較了四種 embedding dimension，在 epoch=50 下的結果

Embedding dimension	Training mse	Validation mse
6	0.8289	0.9150
66	0.7682	0.8952
666	0.7660	0.8957
6666	0.7651	0.89927

在比較四個 embedding dimension 後，我發現 embedding dimension 為 66 與 666 時所表現得最為優異。除了在 training mse 可以和 6666 相距不大以外，validation mse 又屬於四個中最低的，速度方面也比 6666 快非常多。最後，由於四種 embedding dimension 上傳到 kaggle 的分數，由 666 所屬的 mse 分數最低，因此我認為此次作業的 embedding dimension 可能選在十位或百位左右的大小會有比較好的表現。

3. (1%)比較有無 bias 的結果。

Ans:

我分別測試了 50 個 epoch，比較有無加 bias 之後的結果。

	Training mse	Validation mse
No bias (early stop)	8.1860	8.1803
Add bias	0.7660	0.8957

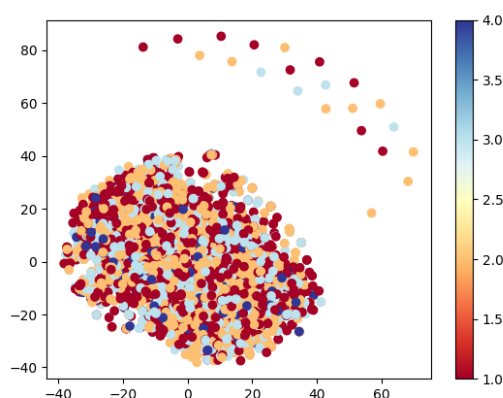
從測試的結果明顯看出，沒有加 bias 的話，會卡在 8.18 附近的 local minimum point，也因為卡在 local minimum，訓練過程在 epoch=39 時就已經停止了。我認為加了 bias 的效果和 learning rate 調整一樣，由於本次作業我的 optimizer 使用的是 Adam()，然而若是 Adam() 的 learning rate 調整不好，也會有類似的狀況。而加了 bias，使每次

的訓練資料都會一部分的 offset，因此容易跳脫出 local minimum point。

4. (1%)請試著將 movie 的 embedding 用 tsne 降維後，將 movie category 當作 label 來作圖。

Ans:

本題我將'Animation, Children's, Comedy, Adventure'歸為第一類，'Drama, Romance, Musical, Fantasy'歸為第二類，'Crime, Thriller, Horror, War, Action, Film-Noir, Mystery'歸為第三類，'Documentary, Western, Sci-fi'歸為第四類，畫出以下 tsne 示意圖。



明顯看出大多類別都集中在左下角的部分，而且也沒有明顯的顏色區分。會造成無法明顯分群的原因，我推估有以下兩點：

- (1) 原先 embedding 的維度不夠高
- (2) 原先 model 精準度就不高。

5. (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。

Ans:

我嘗試使用 Age、MovieID 當作 feature，去求 latent vector。求解的方法和上述相同，使用 matrix factorization，將 embedding dimension 設定為 666，batch_size 為 4096，epochs 為 400，下去做訓練。訓練結果發現，會在第 125 個 epoch 停止(early stop)，而且 **training loss 停在 0.9317**、**validation loss 停在 0.9761**，由此可知結果不如直接拿 UserID，用同樣的方式去做訓練來的好。

會拿 age 的原因，是因為我認為單純拿 UserID 去做 training 感覺起來較不值觀，也就是說我認為電影的推薦可能會牽扯到這個人的年紀或性別。由於時間因素，我沒有再往下繼續做，但我認為，可以用年齡將測試群分成三種等分(例：幼年、青壯年、老年等等)，並在這三個等分中使用不同的模型去做預測，亦或是使用 Gender 當作 feature，並把測試資料分成"F"與"M"兩個等分，如此下去做預測，我認為也會得到比較好的結果。

參考資料：

<https://nipunbatra.github.io/blog/2017/recommend-keras.html>