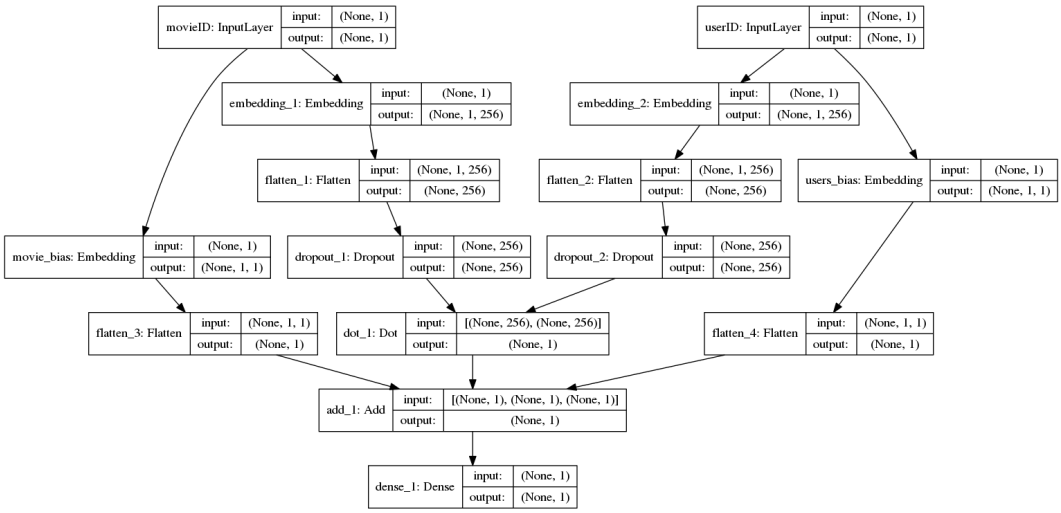
ML2017 HW6 Report

系級：生技三　　學號：b03b02014　　姓名：張皓鈞

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

在此討論的 model 架構如下圖，我使用 Kera 的 Embedding layer 將 user 和 movie ID 分別 embed 到 256 維的空間，加入 bias 之後直接通過 output layer 進行 ratings 的預測，在此我使用的 loss function 是 mean squared error，也就是將此問題視為 regression。在此模型架構下，我用相同的 batch size，隨機取 10% 的 validation data 的 MSE 作為選取模型的依據。



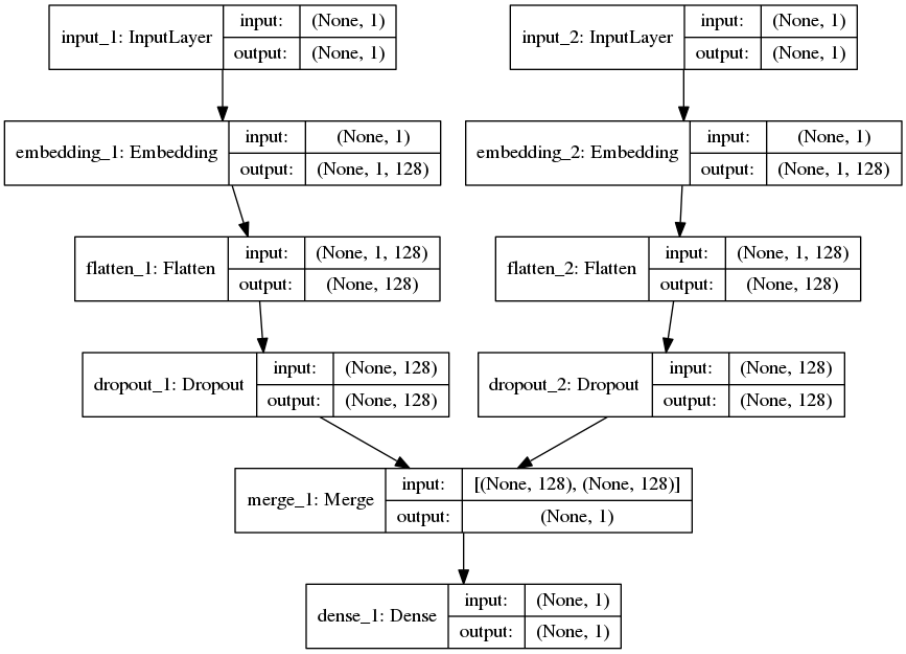
Standardization	training loss	validation loss	public score
No	0.8118	1.0224	0.86895
Yes	0.7735	0.9253	1.02972

在此標準化的過程為將 ratings 扣掉 ratings 的平均再除以標準差，最後預測結果也是乘回 training data ratings 的標準差和加回其平均。

從結果上來看，將 ratings 進行標準化，進行矩陣拆解後對預測 rating 並沒有更好的效果，而且還會使預測分數下降。

2. (1%) 比較不同的 latent dimension 的結果。

在此比較的 model 使用的模型架構如下，相同的訓練過程，ratings 並沒有經過任何 normalization，下表是比較不同的 latent dimension 對 training loss, validation loss 和 public score 的影響：



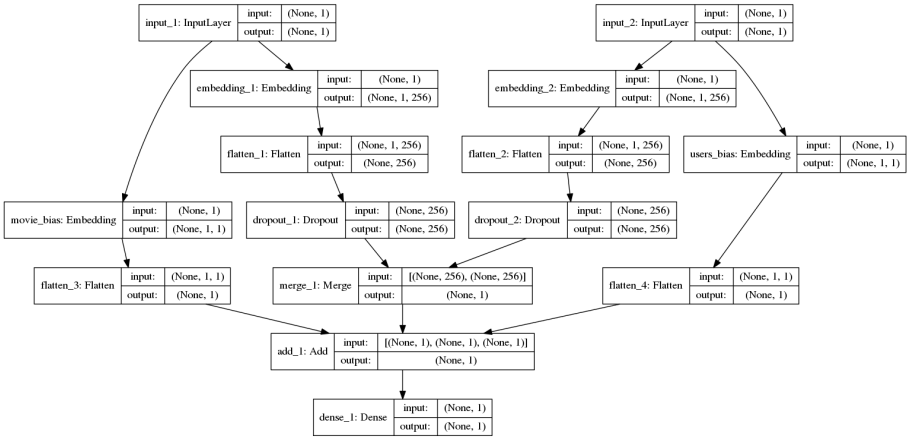
latent dims	training loss	validation loss	public score
32	0.8118	1.2697	0.90180
64	0.7304	1.2746	0.88685
128	0.6354	1.2837	0.88444
256	0.4968	1.2784	0.88295
512	0.3773	1.2854	0.88689

在此架構下，latent dimension 愈大會使 training loss 和 public score 下降，在 256 維時的 public score 最好，由此可推論 user 和 movie 之間的 latent factor 可以用比較多的維度描述，且在這幾個 latent dim 中，256 的維度數量最適合描述 latent factors。

3. (1%) 比較有無 bias 的結果。

在此比較的 model 一樣使用上述的模型架構，相同的訓練過程，ratings 並沒有經過任何 normalization，latent dimension 設為 256。

而我在 user 和 movie 的 embedding layer 之前先加入 bias term，並且用 uniform 初始化 bias，加入 bias 之後的模型架構如圖：



下表是有無 bias 的訓練結果：

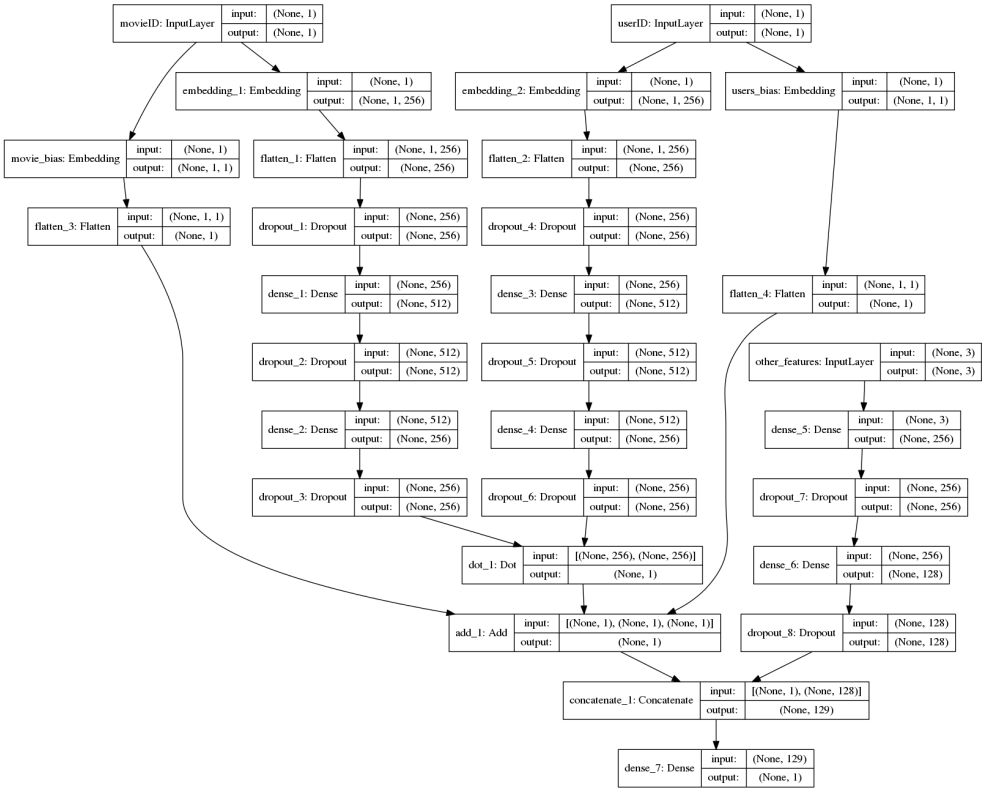
Bias	training loss	validation loss	public score
Yes	0.4607	1.0395	0.86821
No	0.4968	1.2784	0.88295

我發現在訓練過程中，training loss 下降的速度比 val loss 還要快速許多。從訓練結果上來看，有加 bias 的效果更好，顯然有些 latent factor 是需要用 bias 來描述的。

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。

我使用的 DNN 作法是在第 3 題所提的模型架構下，latent factor 改成 256 維。另外再加入 user 的其他 feature，一樣進行 mean squared error 的最小化，輸出的數值是 rating 的正數。

model 架構細節如附圖。



訓練結果如下：

Method	training loss	validation loss	public score
DNN	0.8328	0.9763	0.90789
MF no bias	0.4968	1.2784	0.88295
MF add bias	0.4607	1.0395	0.86821

在此 DNN 的結果並沒有 MF 的好，但是 validation loss 比較能代表 public score 的表現。但之所以沒有比 MF 還好，可能是因為 DNN 的參數還沒辦法讓 feature 轉換的夠好。

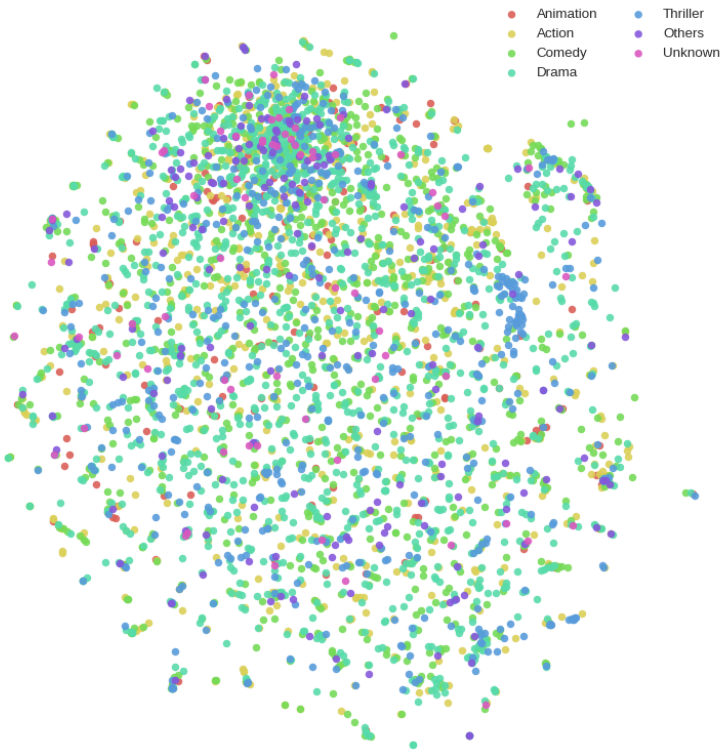
5. (1%) 請試著將 movie 的 embedding 用 tsne 降維後，將 movie category 當作 label 來作圖。

此題討論的 model 架構是第 3 題的加入 bias 後的 model。而類別的處理是先將每個電影都只保留第一個類別，之後按照下表將一些細項的電影類別歸類到其他類別，並且新增 Others 的類別代表其他電影類別，Unknown 代表在 movie.csv 中沒有該 movieID 的電影資料。

New Genre	Old Genre
Drama	Musical, Romance
Thriller	Horror, Crime, Mystery
Action	Sci-Fi, Adventure, Fantasy
Childrenś	Animation

我先使用 PCA 將 256 維的 embedding vector 降維到 45 維，再使用 tsne 壓縮到 2 維。最終各類別的資料點個數：

Genre	Animation	Action	Comedy	Drama	Thriller	Others	Unknown
Number of points	179	706	1024	1251	530	193	69

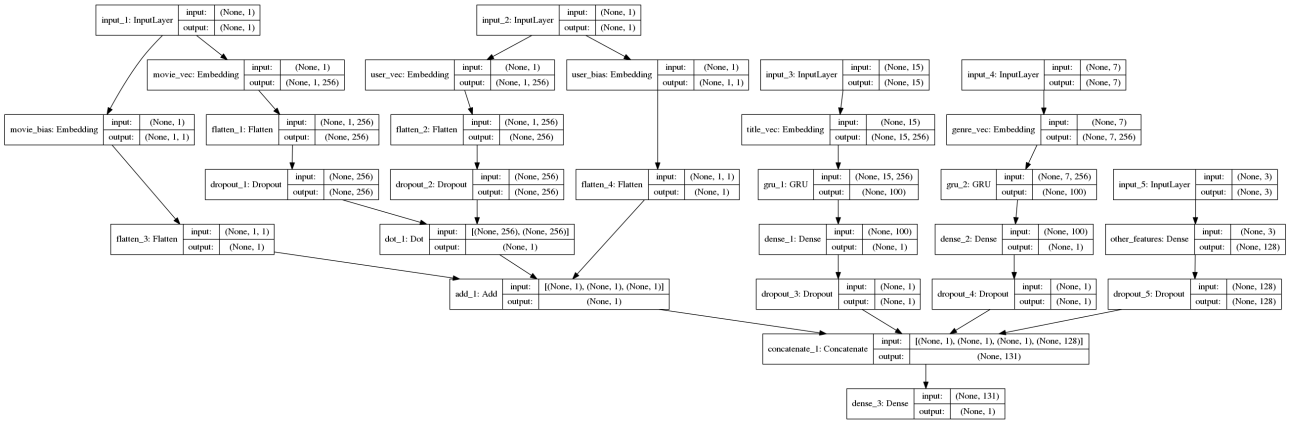


從圖中可以看到有多數的點聚集在左上角，Others 和 Unknown 多聚集在左上角，只有少數有在外圍。另外在圖中，右手邊有一群明顯的 Thriller 聚集在一起，右上角也有一小條細長的資料點聚集。由於每個電影都屬於多個類別，我只取第一個類別作為主要類別，因此有些不同類別的電影點重疊在一起或是很靠近表示這兩部電影可能有重複的類別存在。

6. (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。

我嘗試加入 movie title, genre 和 user age, gender, occupation 的資訊作為 feature。首先 title 和 genre 分別經過不同的 tokenization，並且將其 index pad 成相同長度後訓練一層 GRU layer。

另外 user 的資訊我另外用成一個 3 維的 vector，經過 Dense layer 的轉換後一起併入最後的 tensor。訓練結果和模型架構如下圖：



Method	training loss	validation loss	public score
Add extra info.	0.4388	1.1285	0.87230
Add bias	0.4607	1.0395	0.86821

原本預期中，加入更多關於 user 和 movie 的資訊應該會有更好的效果，然而實際結果卻是比較差的，原因可能是模型架構不夠複雜，導致 validation loss 沒辦法一直下降，造成 early stopping; 也有可能是因為加入太多沒有什麼幫助的 feature，像是 movie genres 的類別分類太詳細，反而影響到 model 的預測能力。