

ML2017 HW6 Report

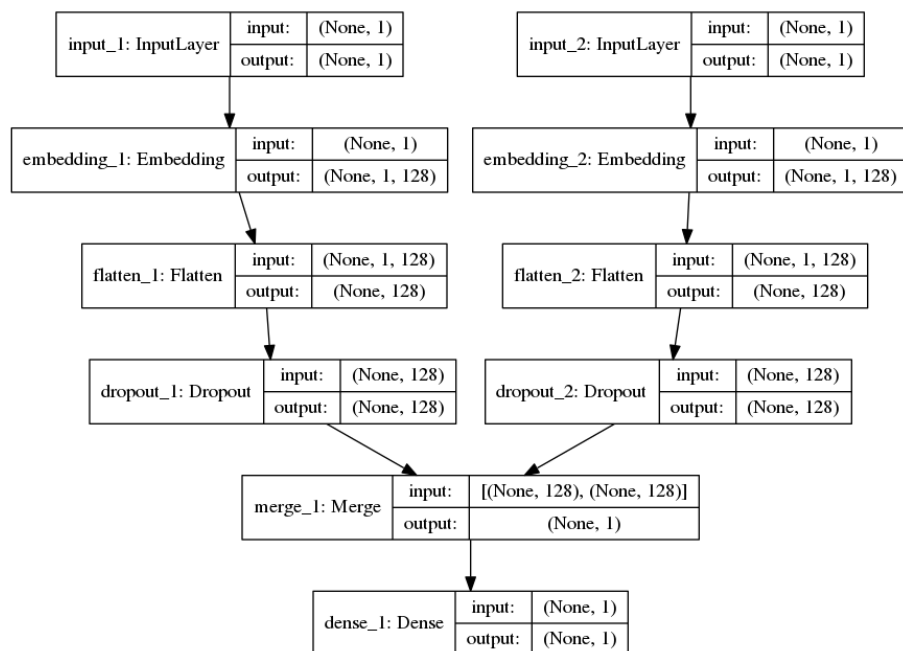
系級：生技三

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 128 維的空間，並直接通過 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。



在沒有對 ratings 做任何處理時，預測出的結果在 public kaggle score 是 0.88444; 而當我對 ratings 進行 max-scaling 後，結果在 public kaggle score 是 0.88495。

從結果上來看，將 ratings 除以最大的可能值，也就是 5，進行矩陣拆解後再經過 DNN 預測 rating 並沒有顯著的影響。

2. (1%) 比較不同的 latent dimension 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，下表是比較不同的 latent dimension 對 training loss, validation loss 和 public score 的影響：

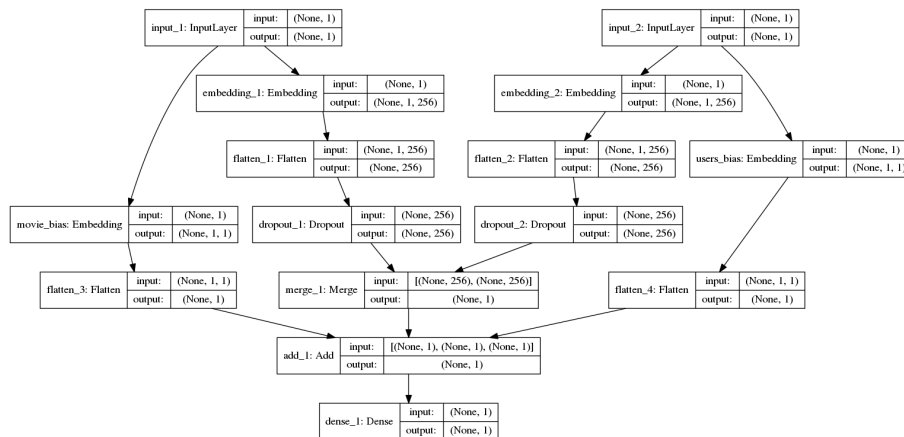
latent dims	training loss	validation loss	public score
32	0.8118	1.2697	0.90180
64	0.7304	1.2746	0.88685
128	0.6354	1.2837	0.88444
256	0.4968	1.2784	0.88295
512	0.3773	1.2854	0.88689

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。

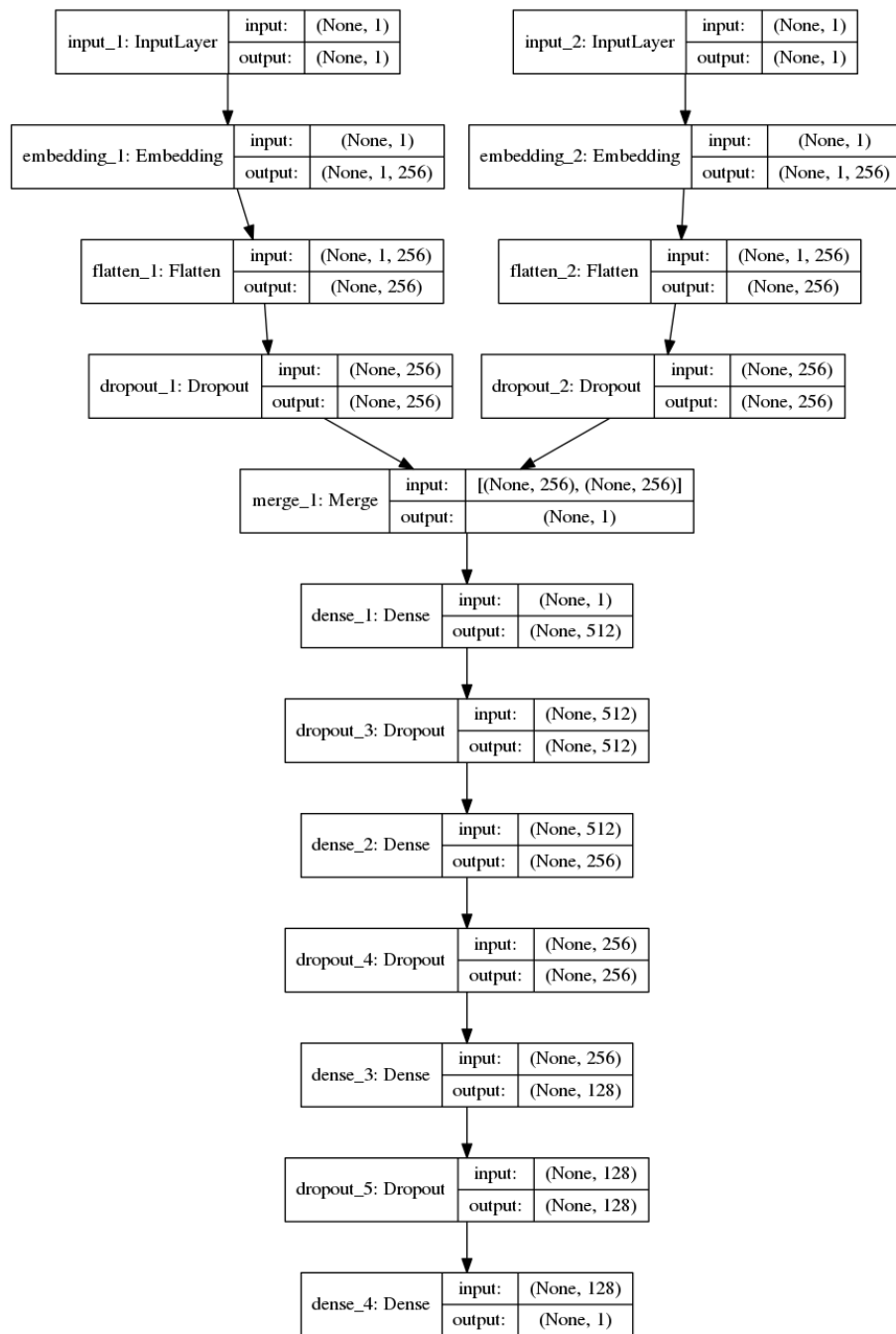
4. (1%) 請試著用 DNN 來解決這個問題，並且說明實做的方法 (方法不限)。並比較 MF 和 NN 的結果，討論結果的差異。

我使用的 DNN 作法是在第 1 題所提的模型架構下，latent factor 改成 256 維，並將 merge 起來之後的部份多加數層 hidden layer，並且一樣進行 mean squared error 的最小化，輸出的數值是 rating 的正數。多加的 hidden layer 架構細節如附圖。

訓練結果如下：

Method	training loss	validation loss	public score
DNN	1.5849	1.2960	1.11271
MF	0.4968	1.2784	0.88295

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



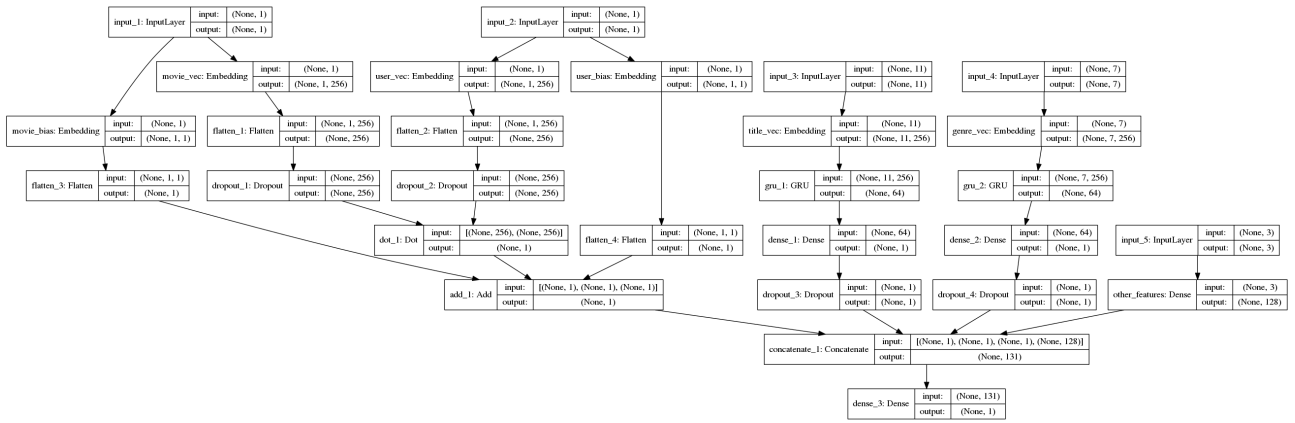
6. (BONUS)(1%) 試著使用除了 rating 以外的 feature, 並說明你的作法和結果, 結果好壞不會影響評分。

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.	0.6897	1.1167	0.89683
Add bias	0.4607	1.0395	0.86821
Problem 1	0.4968	1.2784	0.88295

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。