學號：b04902053 系級：資工二 姓名：鄭淵仁

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

* normalize的方法

我總共測試兩種normalize的方法，第一種是把所有的rating直接normalize，第二種是每一個UserID都各自normalize各自的rating。

* 實驗結果

我使用相同的model，並切validation set，而且save best來測試三種不同normalize的方法。實驗結果如下表：

|  |  |  |  |
| --- | --- | --- | --- |
|  | no normalize | normalize on all rating | normalize for each user |
| validation RMSE | **0.84909(勝)** | 0.86001 | 0.86832 |

從實驗結果可以發現：沒有加上normalize的結果是三者中最好的。

1. **(1%)比較不同的latent dimension的結果。**

為了方便比較，我讓user和movie的embedding的latent dimension相同。

* 實驗結果

我都使用相同的model，並切validation set，而且save best來測試。

實驗結果如下圖：

從圖片中可以發現：當latent dimension降到150的時候，RMSE是最小的。

其中，當dimension小於150的時候，RMSE會特別大；當dimension大於150之後，RMSE只會稍微增加。

* 推測原因

會有這樣的結果的原因可能是：如果latent dimension太小（<150），無法完全代表原本資料隱含的成分，結果當然很差。相反的，當latent dimension夠大並且越來越大的時候，雖然embedding能夠代表所有資料隱含的成分了，但是因為dimension太大了，gradient descent很容易會overfit trainging set，而無法提升validation set的結果。

1. **(1%)比較有無bias的結果。**

我把user和movie都個別加上bias，都使用相同的model，並切validation set，而且save best來測試。

實驗結果如下表所示：

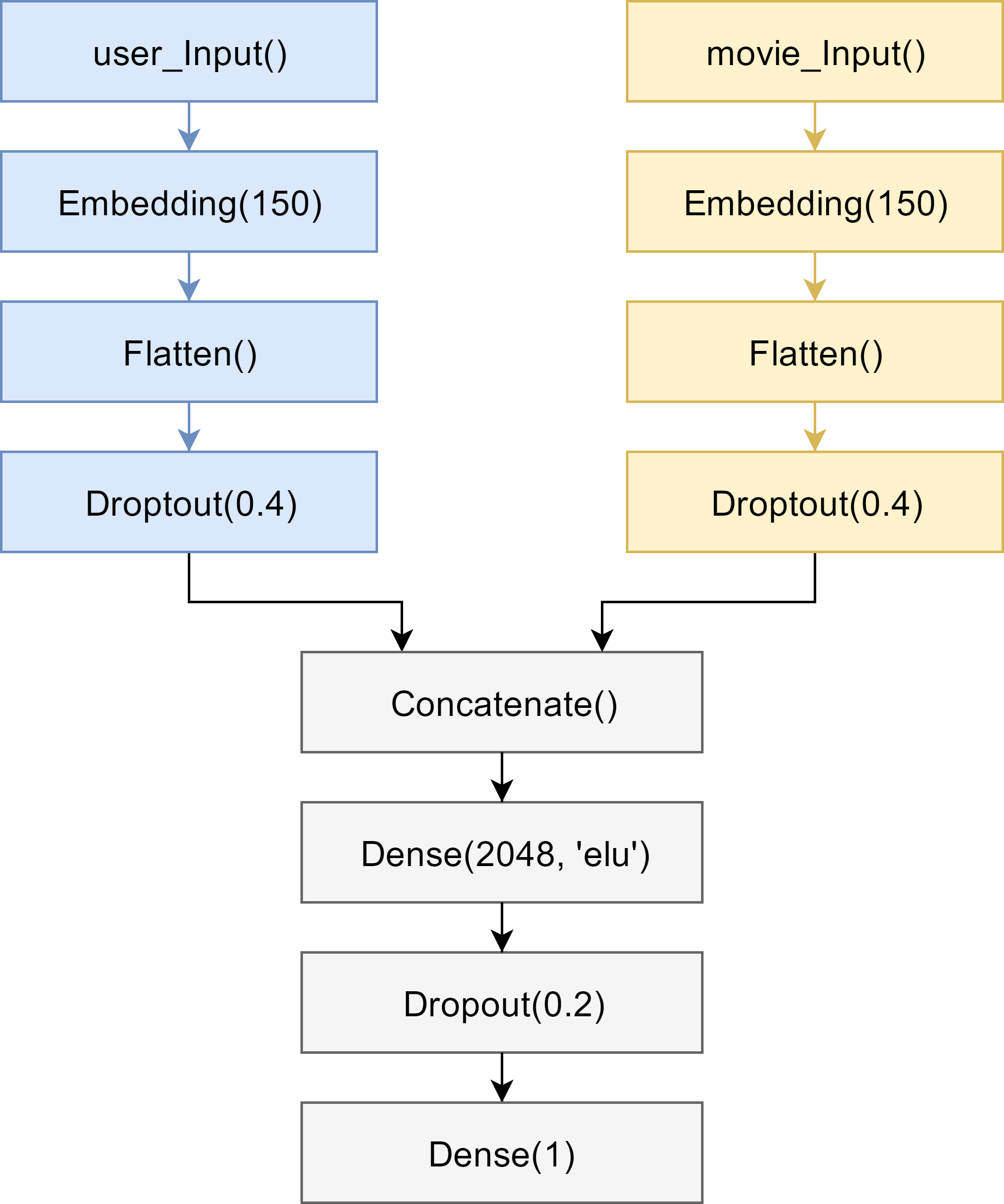
|  |  |  |
| --- | --- | --- |
|  | no bias | add bias |
| validation RMSE | **0.84909(勝)** | 0.85498 |

從實驗結果可以發現：沒有加上bias的結果是最好的。

1. **(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。**

* 實做的方法：

我把user的embedding和movie的embedding接在一起，作為DNN的input，並且設定output的結果直接是predict的rating。完整的model如下圖：



* 結果及比較：

測試兩種model的時候，我都切validation set，並且save best來測試。

實驗結果如下表：

|  |  |  |
| --- | --- | --- |
|  | MF | DNN |
| validation RMSE | 0.84909 | **0.84847(勝)** |

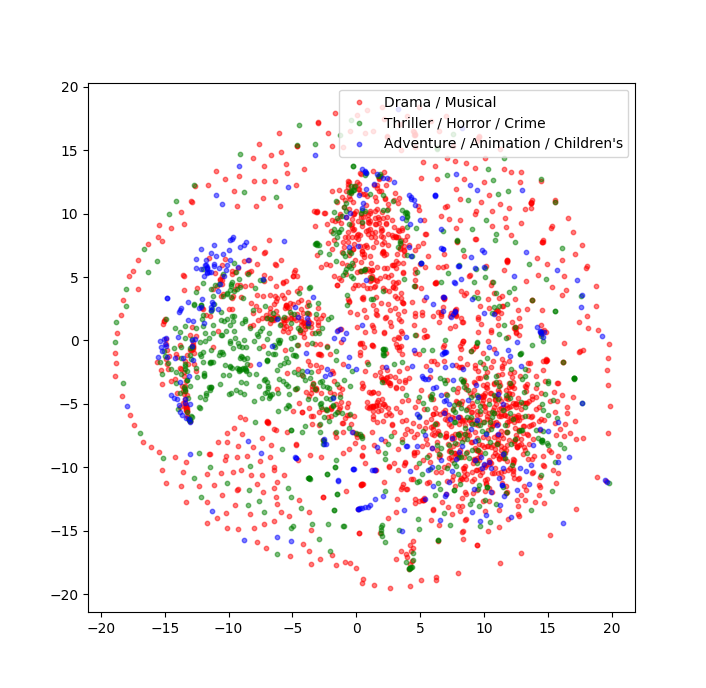
從表格中可以發現：使用DNN的結果是比較好的。

* 推測原因

會有這樣的結果的原因是因為：DNN的model其實包含了MF的model，所以DNN的結果至少會和MF一樣或更好。

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

我使用和助教的投影片一樣的分類，也就是把Drama / Musical作為一類，Thriller / Horror / Crime作為一類，Adventure / Animation / Children's作為一類。畫出來的結果如下圖：

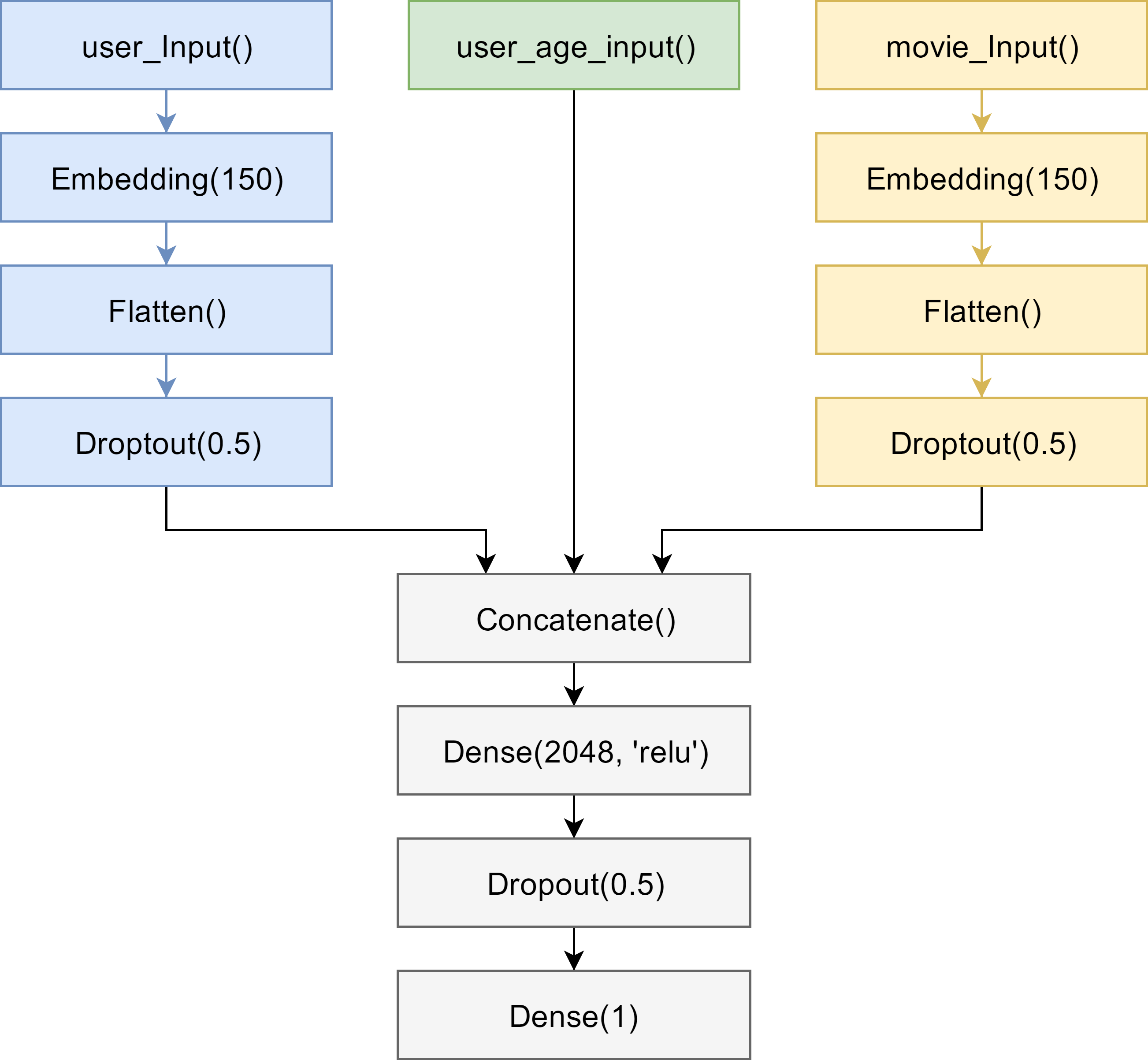


從圖中可以看出：有一部分的綠色、紅色、藍色的點會自己聚集在一起，而不會跟其他顏色混在一起。這代表我的model有一定能力可以區分不同類別。

但是三種顏色混在一起的情況也不少，這代表我的model還沒有辦法很好的處理這些情況下的資料。

1. **(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。**

我只取user的age，並Concatenate到user和movie的embedding上，再使用DNN來訓練出model。model的架構如下圖所示：



訓練出來的結果如下表：

|  |  |  |
| --- | --- | --- |
|  | only training set | training set + user age |
| validation RMSE | 0.84847 | **0.84547(勝)** |

從表格中可以發現，加入了user的age比只使用training set的資料的結果要好一點。