

實作心得報告

一、論文要解決的問題

- 預測 user 下一個進行的 interaction
- 預測 user state change

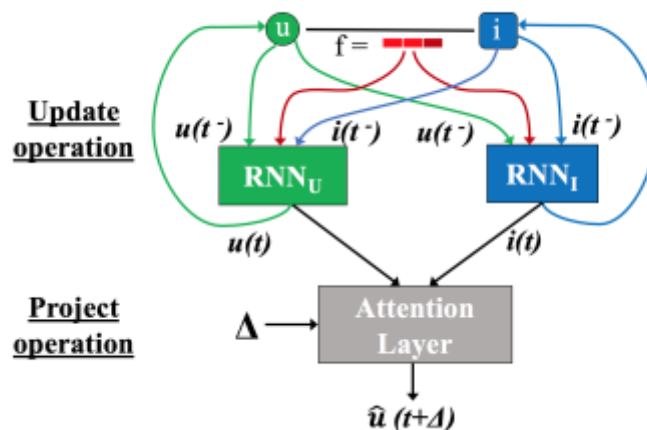
二、論文方法設計

1. Update operation:

使用兩個 RNN 生成 user 和 item 在 t 時間的 dynamic embedding

2. Project operation:

使用 attention 機制來預測 user 在 $t+\Delta$ 時間的 user embedding



3. Training to predict next item embedding:

1) Predicted item embedding:

$$\tilde{j}(t + \Delta) = W_1 \hat{u}(t + \Delta) + W_2 \bar{u} + W_3 i(t + \Delta^-) + W_4 \bar{i} + B$$

2) Loss function:

$$\begin{aligned} Loss = & \sum_{(u, j, t, f) \in S} ||\tilde{j}(t) - [j, j(t^-)]||_2 \\ & + \lambda_U ||u(t) - u(t^-)||_2 + \lambda_I ||j(t) - j(t^-)||_2 \end{aligned}$$

4. t-batch:

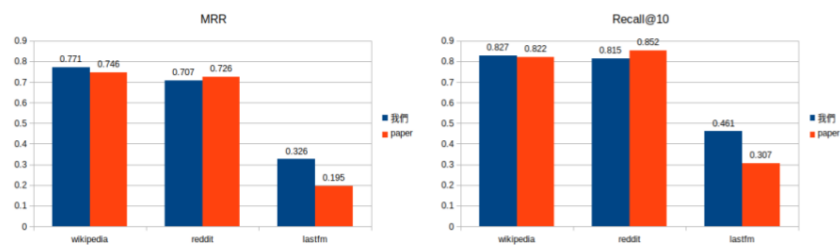
- 1) Select step : a new batch is created by selecting the maximal edge set
- 2) Reduce step : the selected edges are removed from the network

三、實驗結果及討論

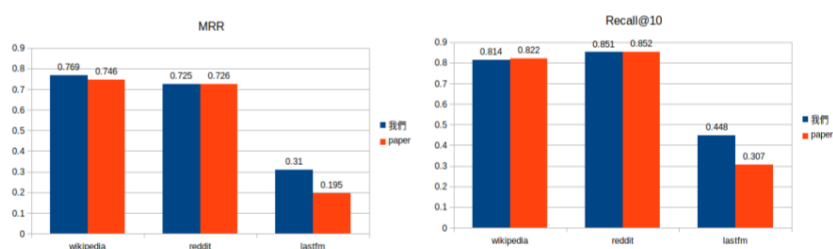
1) 與論文結果比較及分析

interaction

Validation



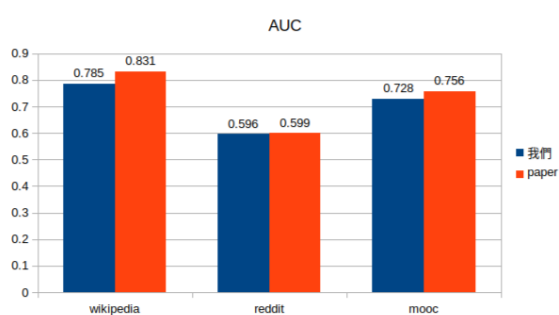
Test



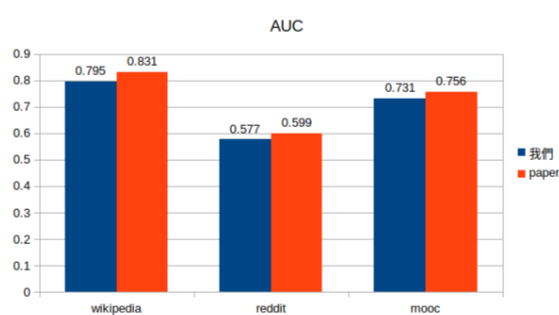
討論: lastfm 以外之資料集之結果與 paper 無明顯差異。由於 lastfm 資料集, 使用者進行同樣 interaction 的比例僅 8.6%, 此為與 paper 結果差異之可能原因。

user state

Validation



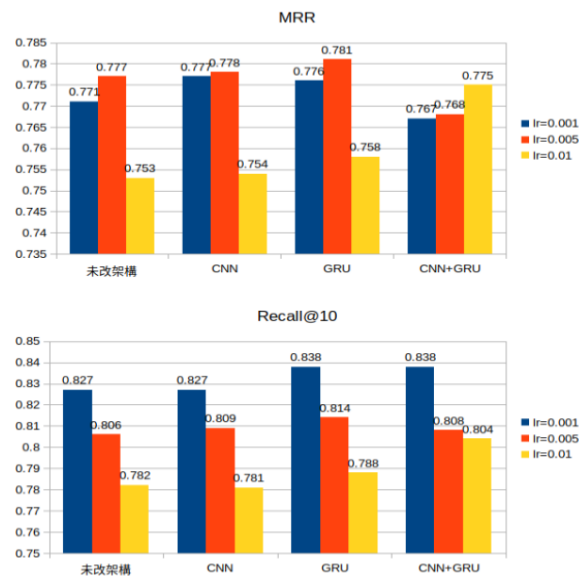
Test



討論: 結果與paper無明顯差異

2) 改動結果比較及分析

interaction(validation)



討論：

A. MRR:

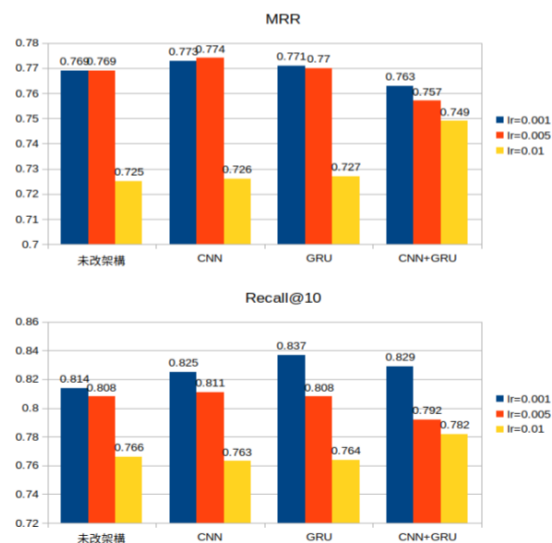
CNN+GRU 以外之 model 在 learning rate=0.005 有最佳結果, learning rate=0.001 時其次, learning rate=0.01 時最佳。CNN+GRU 則是隨 learning rate 增加而遞減。

B. Recall@10:

所有 model 結果都隨 learning rate 增加而遞減, 因為 learning rate 太大導致無法走到 local minimum 的點。

C. GRU 結果稍微比其他 model 好

interaction(test)



討論：

A. MRR:

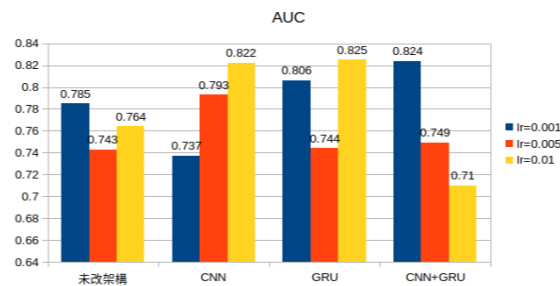
CNN+GRU 以外之 model 在 learning rate=0.005 有最佳結果, learning rate=0.001 時其次, learning rate=0.01 時最佳。CNN+GRU 則是隨 learning rate 增加而遞減。

B. Recall@10:

所有 model 結果都隨 learning rate 增加而遞減, 因為 learning rate 太大導致無法走到 local minimum 的點。

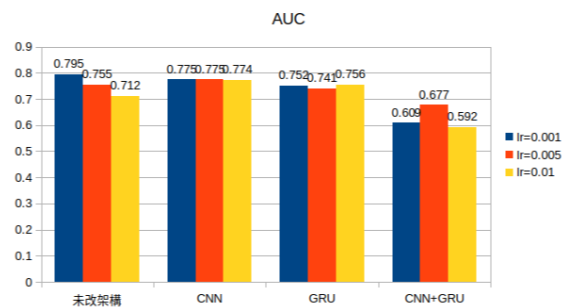
C. GRU 結果稍微比其他 model 好

user state(validation)



討論: 四個 model 隨著 learning rate 上升, 結果呈現不同趨勢。

user state(test)



討論:原本架構之結果隨著 learning rate 上升而變差。CNN 和 GRU 之結果則不受 learning rate 大小所影響。CNN+GRU 則在 learning rate=0.005 時有最佳結果。全部當中最最好的結果為原本架構 learning rate=0.001 之情況下, AUC=0.795