學研實習計畫
階段性成果報告

洪睿甫、彭彥霖、陳弘軒

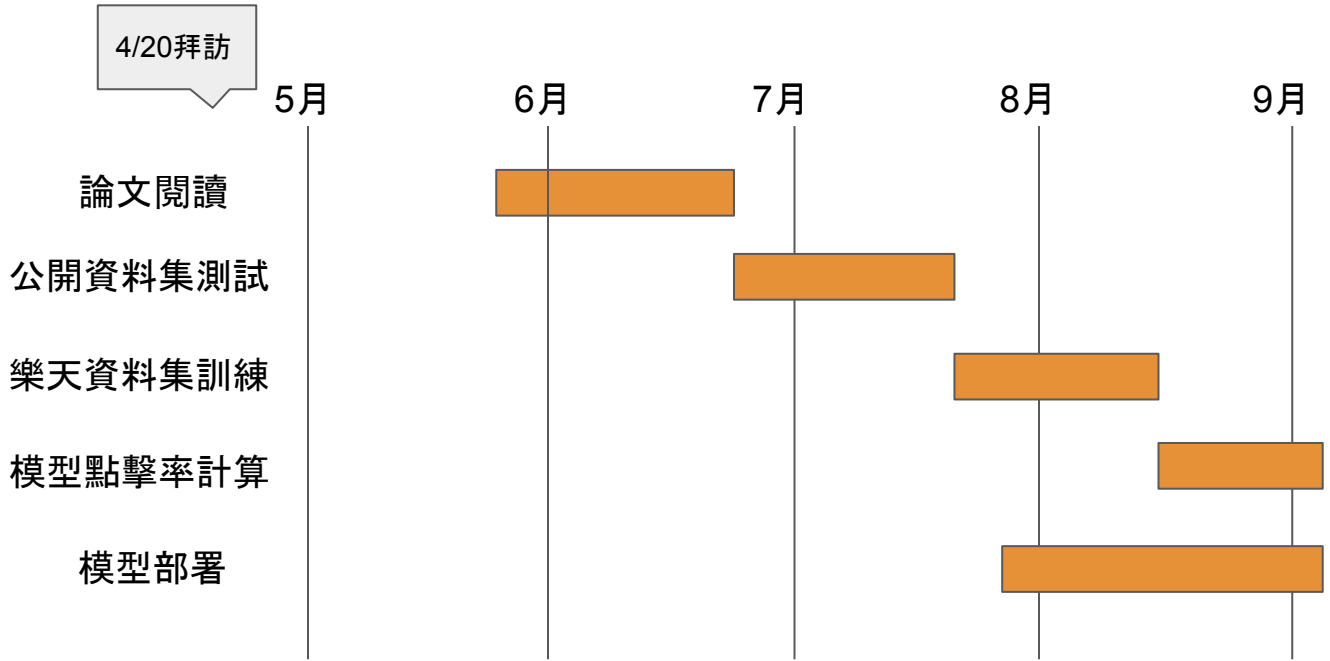
2021-09-08

大綱

- 緣由
- 推薦方法概敘
  - Query-based recommendation vs. model-based recommendation
  - Accelerating the inference of model-based recommendation
    - Models: DIN, W&D, PNN, DeepFM
- Offline evaluation
  - 使用公開資料集的結果
  - 使用樂天資料集的結果
- 使用 online recommendation model 的經驗
  - 模型訓練方式
  - 模型部署方式
- 未來方向

緣由

- 樂天專案成效提升遇瓶頸
- VenRaaS垂直電商推薦需求各不同，希望發展「有自學能力」的推薦演算法
- 嘗試透過學研合作計畫，發展較大階差的技術突破
- 4/21拜訪弘軒與兩位同學，討論此合作的想法
- 合作方式：
  - 由於樂天資料機密性因素，讓 彥霖、睿甫進工研院，以實習身份參與合作
  - 一方面也因同學們需要一些時間學習，累積實作經驗
  - 請弘軒老師指導，有成果後，可再談 產學合作計畫


4/20拜訪
5月
6月
7月
8月
9月
論文閱讀
公開資料集測試
樂天資料集訓練
模型點擊率計算
模型部署

推薦方法概敘

Query-based recommendation

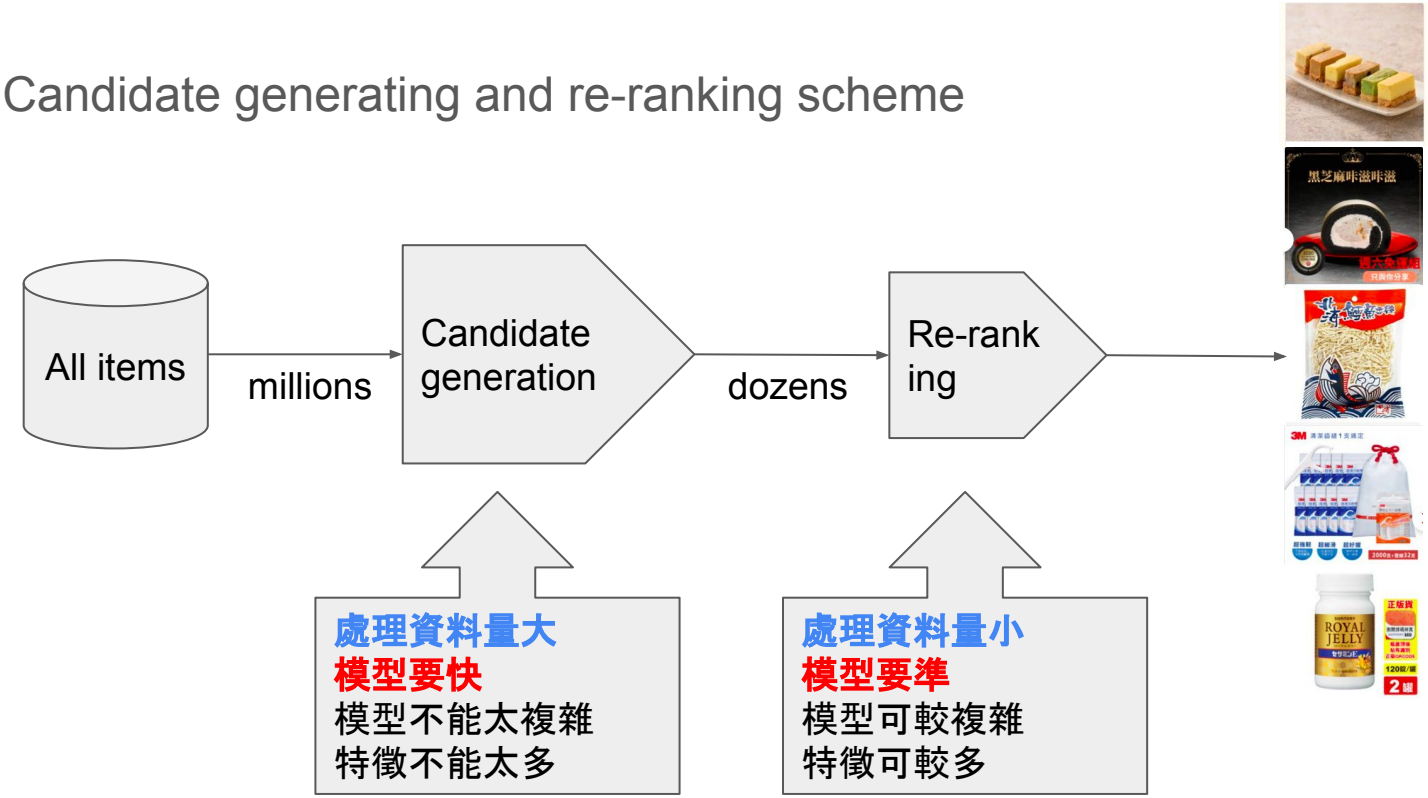
- 離線計算好各種推薦策略，將推薦策略存入資料庫
- 上線後查詢資料庫，決定推薦清單
- E.g.,
  - 剛看完唇膏，應推薦什麼？
  - 百萬商品，故每個商品的推薦清單可事先計算好
- 優點
  - 線上的計算量極低，可快速回應
- 缺點
  - 當推薦情境較多時，無法將所有可能情境全部事先計算

Model-based recommendation

- 離線計算好推薦模型，儲存模型 (而非推薦策略)
- 上線後模型依當時情境計算推薦清單
- E.g.,
  - 假日、晴天、女性使用者、剛看過唇膏、化粧水，應推薦什麼？
  - 情境極多，難以事先列舉所有可能性
- 優點
  - 可應付更多的情境
- 缺點
  - 上線後需要依情境做各種計算，故線上的計算量較高

加速 model-based recommendation 的推論時間

- Candidate generating and re-ranking scheme



加速 model-based recommendation 的推論時間

- Candidate generating
  - 使用目前 ITRI 目前線上模型的推薦清單當作candidate items
    - 線下計算各個商品的candidate recommendation items
    - 線上僅需查表 (快！)
- Re-ranking
  - 重排 candidate item，讓最可能被點擊(或購買) 的商品放在前面
  - 測試演算法
    - Wide & Deep (W&D)
      - Proceedings of the 1st workshop on deep learning for recommender systems 2016
      - Citations: 1481
    - Product-based Neural Networks (PNN)
      - ICDM 2016
      - Citations: 262
    - DeepFM
      - IJCAI 2017
      - Citations: 755
    - Deep interest network (DIN)
      - SIGKDD 2018
      - Citation: 446

An overview of W&D, PNN, DeepFM, and DIN

- W&D, PNN, DeepFM：使用當下資訊預測下一步
  - 使用 “目前” 商品的 feature、一個 “candidate 商品” 的 feature、及其他 feature (e.g., 使用者的資訊、或其他 contextual feature) 共同預測該 candidate item 是下一個被點擊的商品的機率
  - 舉例
    - 正在觀看商品 *a*, candidate items 包括：*c*1, *c*2, *c*3, 推薦演算法 *f*
    - 將 *c*1, *c*2, *c*3 依 *f*(*a*, *c*1), *f*(*a*, *c*2), *f*(*a*, *c*3) 的結果由高至低排列
- DIN：使用過去及當下資訊共同預測下一步
  - 使用 “目前” 商品的 feature、過去點擊過的商品的 feature、一個 “candidate 商品” 的 feature、及其他 feature (e.g., 使用者的資訊、或其他 contextual feature) 共同預測該 candidate item 是下一個被點擊的商品的機率
  - 舉例
    - 一個目前為止的 clickstream: [*a*1, *a*2, *a*3, *a*4], , candidate items 包括：*c*1, *c*2, *c*3, 推薦演算法 *f*
    - 將 *c*1, *c*2, *c*3 依 *f*([*a*1, *a*2, *a*3, *a*4], *c*1), *f*([*a*1, *a*2, *a*3, *a*4], *c*2), *f*([*a*1, *a*2, *a*3, *a*4], *c*3) 的結果由高至低排列

User Behavior History

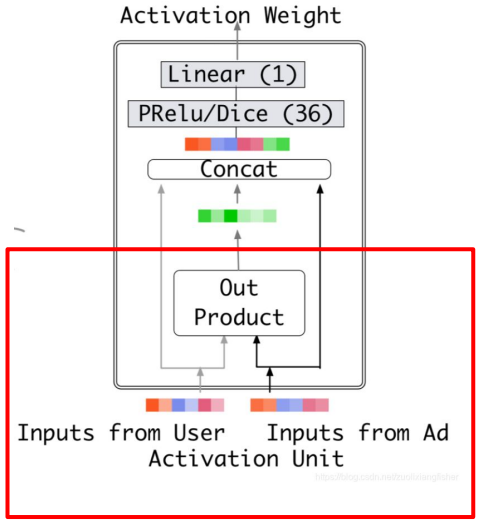
- Diverse:當使用者逛店商網站時會對很多商品都感興趣
- Local Activation:使用者是否會點擊推薦給他的廣告，只取決於部分的行為紀錄

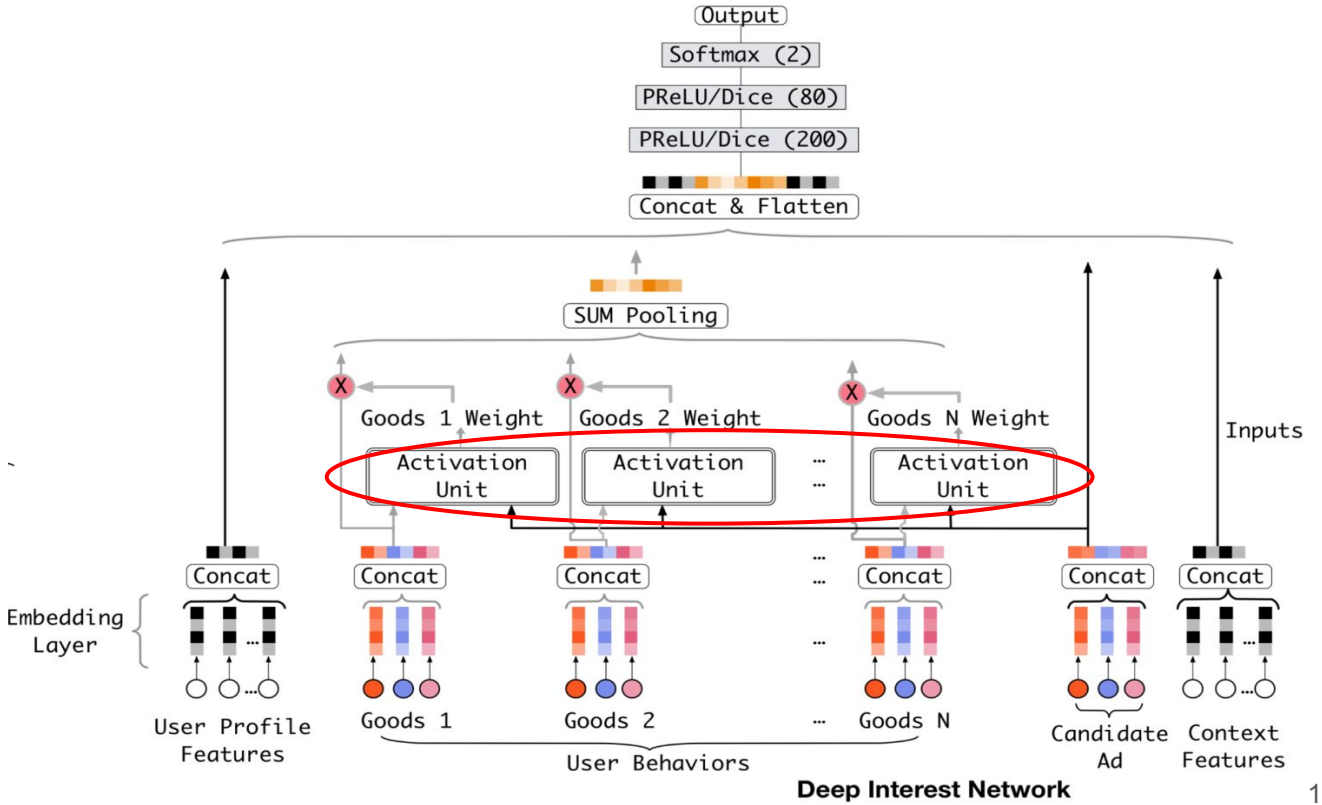
Table 1: Examples of user behavior history from online product.

User	Behavior History	Candidate Ad
Young Mother	woolen coat, T-shirts, earrings,children's coat leather handbag, miniskirt, sports underwear	long sleeved jacket
Swimmer	bathing suit, kickboard, swimming cap, travel book tent, potato chips, nuts, potato chips, ice cream	goggle

DIN

- **DIN model**





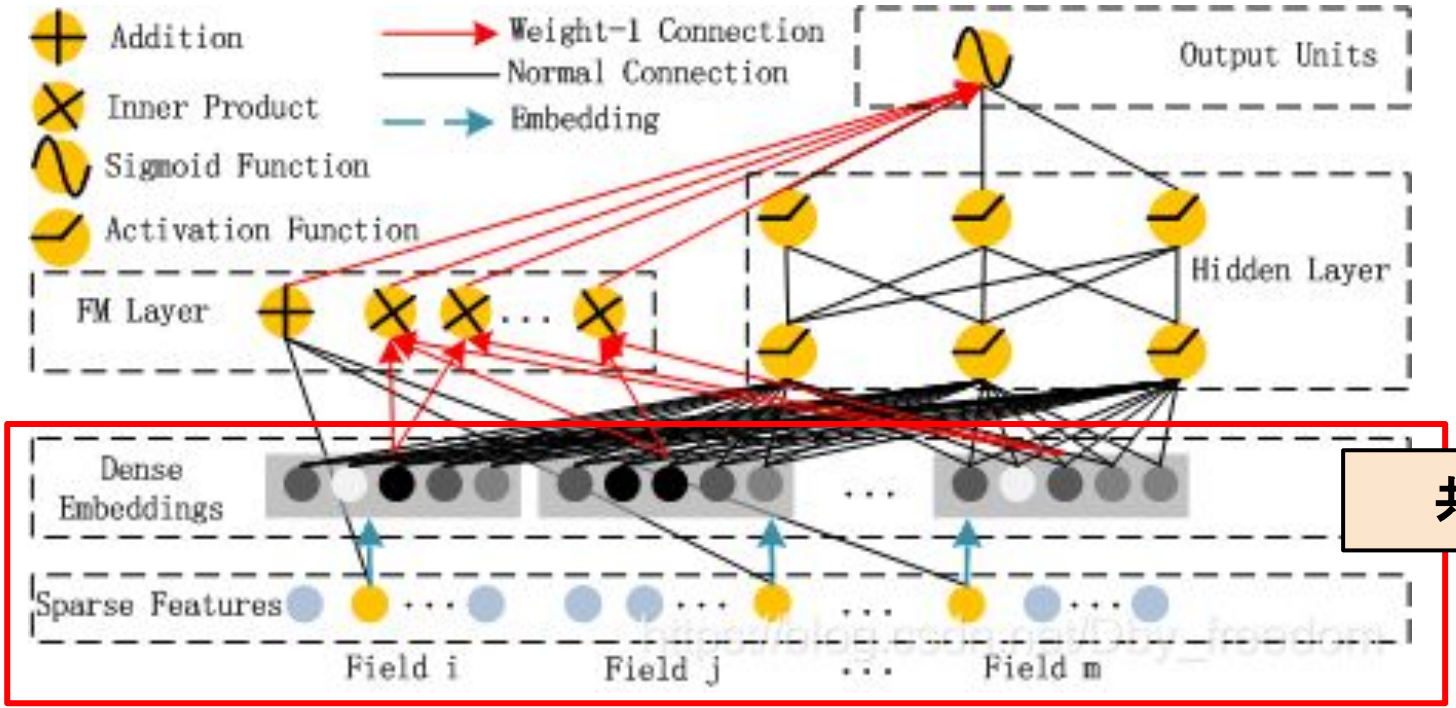
Deep Interest Network

DeepFM

- **Linear模型:**
  泛化能力弱，沒辦法學習未出現過或是較高階的商品組合
- **Deep模型:**
  學習潛在規則，推導出從未見過的商品組合

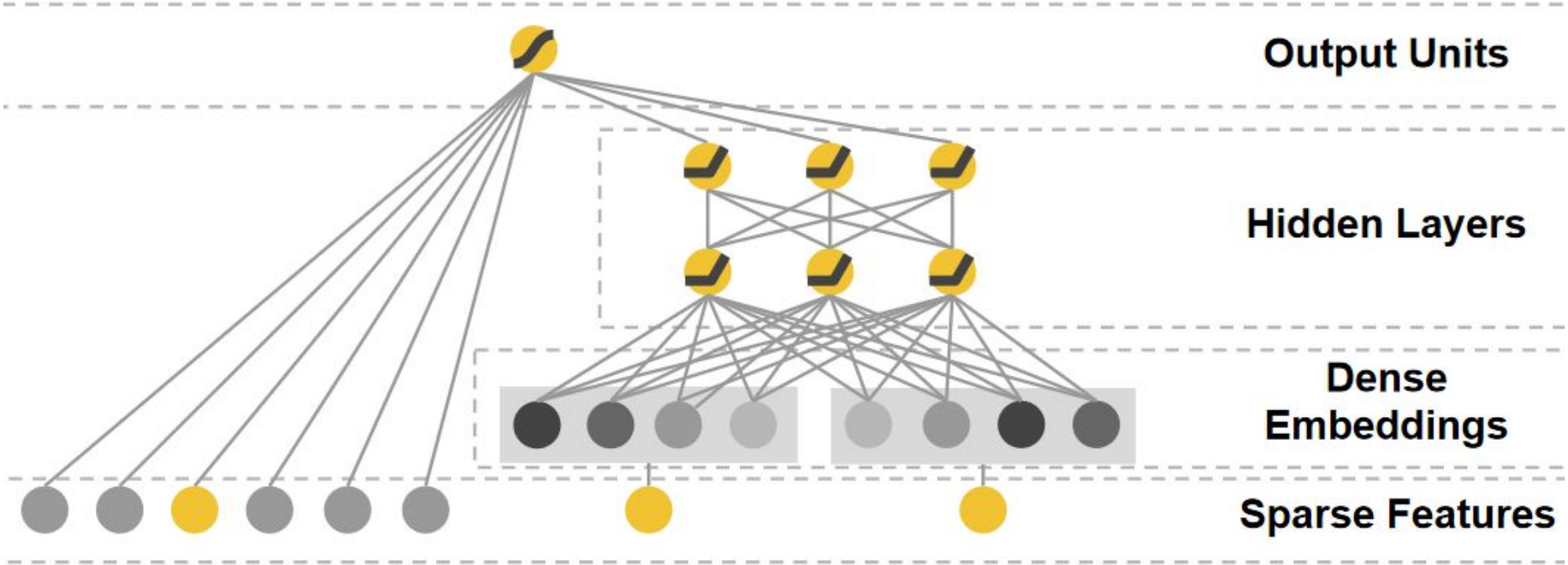
- **Factorization Machine:**
  善於處理高度稀疏的資料，比起線性模型更省時間

DeepFM Model



共用input

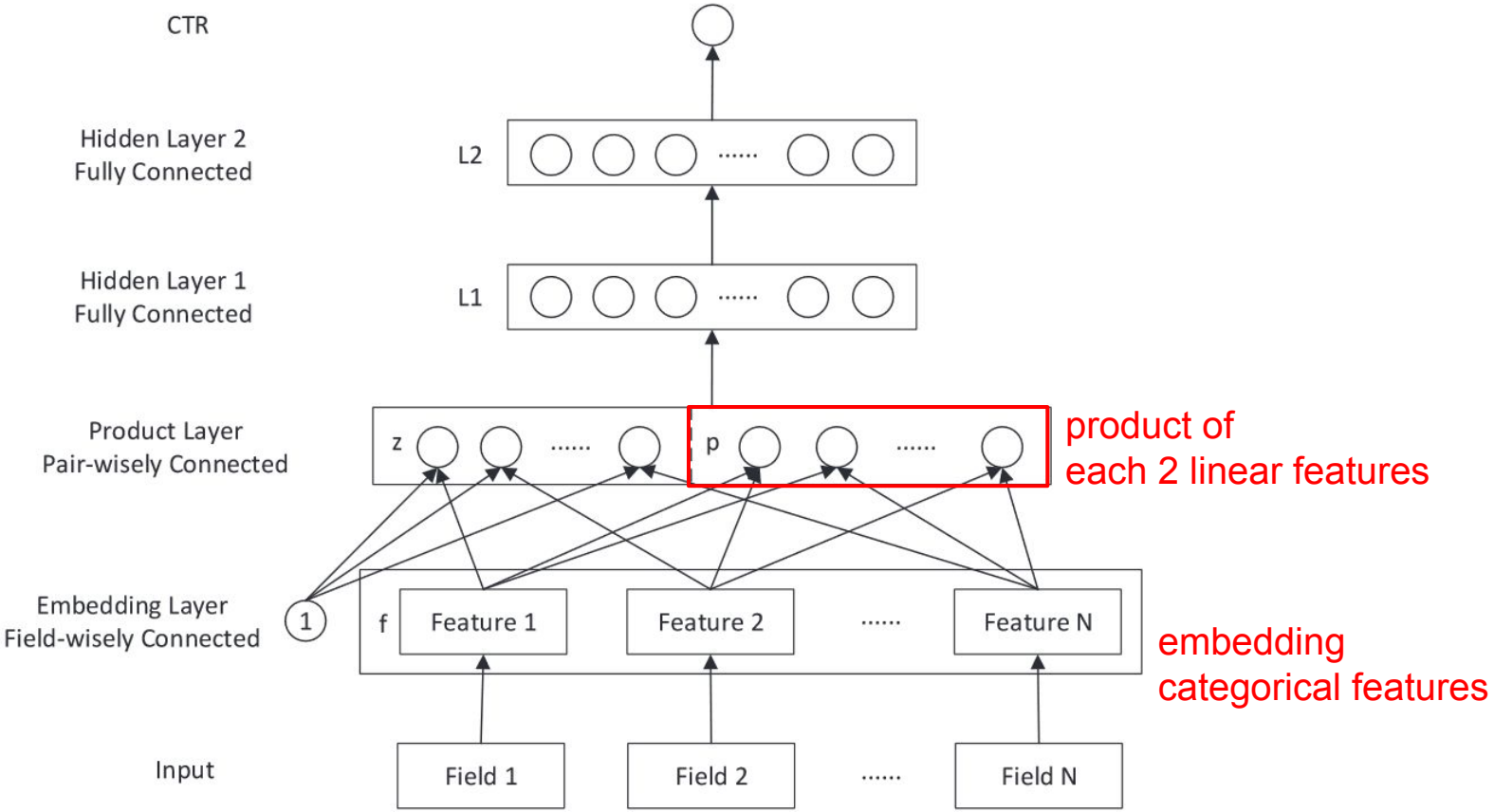
Wide & Deep


Output Units
Hidden Layers
Dense Embeddings
Sparse Features
Wide linear model
Deep neural networks

Wide & Deep

- The Wide Part
  - for memorization
    - 擅長記住training data中的特例
  - Input features:
    - categorical features
    - transformed features (cross-product of categorical features)
- The Deep Part
  - for generalization
    - 學習潛在規則，對未知input做預測
  - Input features:
    - continuous features
    - categorical features (convert to embedding vectors)

Product-based Neural Networks



Product-based Neural Networks

- 由內積/外積運算產生複雜signal來學習潛在規則
- Input features:
  - categorical features (convert to embedding vectors)
- Product layer
  - linear signal
    - 原始signal
  - product signal (二次)
    - 由linear signal互相內積/外積產生
- (linear signal + product signal)當作下層hidden layer輸入

Offline evaluation

1. Evaluation 1: 在 open datasets 使用新方法做 re-ranking
2. Evaluation 2: 在樂天資料集使用新方法做 re-ranking
3. Evaluation 3: 在樂天資料集使用新方法+ITRI目前的推薦方法做 re-ranking

Offline evaluation 1: 公開資料集

- criteo_small (2.7MB): 8,000 + 2,000 (train/eval data)
- criteo_medium (261MB): 800,000 + 200,000 (train/eval data)
- criteo_large (2.3GB): 8,000,000 + 2,000,000 (train/eval data)
- ipinyou_2997 (155MB): 312,438 + 156,064 (train/eval data)
- ipinyou_3386 (912MB): 747,647 + 300,929 (train/eval data)
- Amazon-electron_review(1G): 1,689,188
- MovieLens1m(23.4 MB): 1,000,000

Offline evaluation 1: 公開資料集

使用公開資料集的原因:

1. 取得樂天資料前先做測試
2. 熟悉各模型的使用方式
3. 確認模型可得到接近論文回報的 AUC 數值

AUC score	Criteo (medium)	Criteo (large)	iPinyou (medium)	iPinyou (all)	Amazon	Movielens
DIN	----------	----------	----------	0.5791	0.6242	**0.7365**
PNN	0.6932	**0.7913**	**0.6366**	0.5717	**0.6332**	0.6747
DeepFM	0.7018	0.7905	0.5905	**0.6026**	0.62	0.6832
W&D	**0.7033**	0.7896	0.6	0.5793	0.62	0.6755

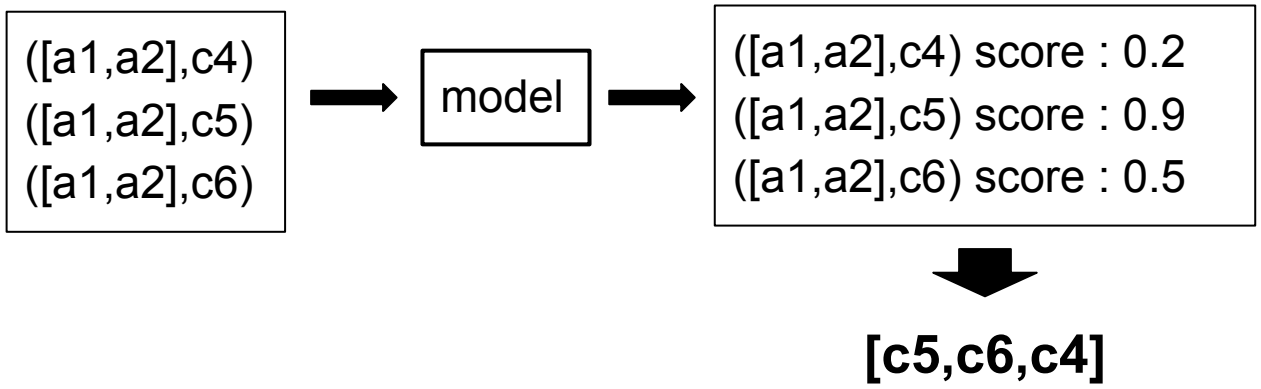
Offline evaluation 2: 樂天資料集 - DIN

- **validation set**

uid	gid	next_gid	clickstream	candidate_item
1	a1	a2	[a1]	[c1,c2,c3]
1	a2	a3	[a1,a2]	[c4,c5,c6]
1	a3	a4	[a1,a2,a3]	[c7,c8,c9]

- **weblog**

uid	gid
1	a1
1	a2
1	a3
1	a4



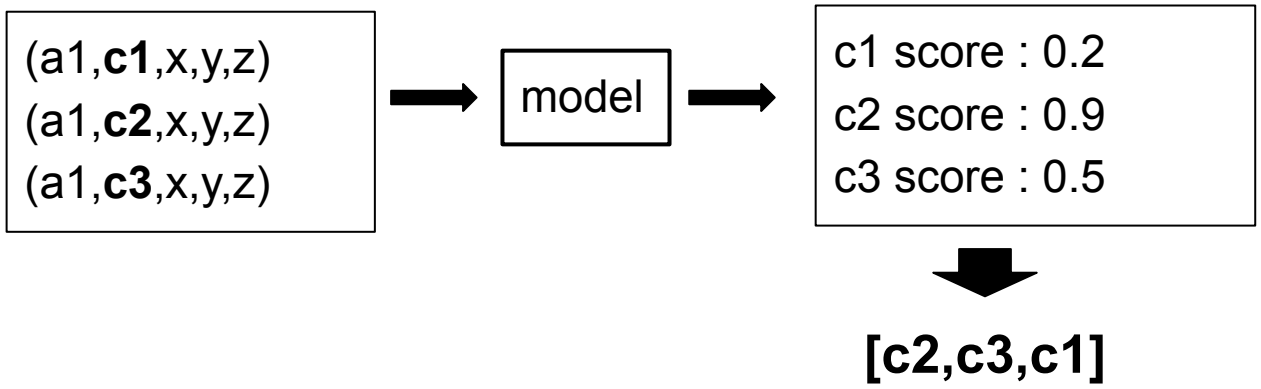
Offline evaluation 2: 樂天資料集 - PNN / DeepFM / W&D

- **validation set**

uid	gid	next_gid	candidate_item	categ_code	agent	device
1	a1	a2	[c1,c2,c3]	x	y	z
1	a2	a3	[c4,c5,c6]	...	...	...
1	a3	a4	[c7,c8,c9]	...	...	...

- **weblog**

uid	gid
1	a1
1	a2
1	a3
1	a4



Offline evaluation 2: 樂天資料集

- 訓練：4/1 - 4/30，測試：5/1 (top-*k* accuracy)

	Candidate	DIN	PNN	DeepFM	W&D
top5	0.2351	0.1179	**0.142**	0.1396	0.1404
top10	0.3239	0.2132	**0.2351**	0.2339	0.2336
top15	0.3722	0.2913	**0.3083**	0.3082	0.3066

- 訓練：4/1 - 5/1，測試：5/2 (top-*k* accuracy)

	Candidate	DIN	PNN	DeepFM	W&D
top5	0.2387	**0.138**	0.1226	0.1209	0.123
top10	0.329	**0.2251**	0.1944	0.1921	0.195
top15	0.3798	**0.2971**	0.2774	0.2717	0.2758

Offline evaluation 2: 樂天資料集 (cont')

- 訓練：4/1 - 5/2，測試：5/3 (top-*k* accuracy)

	Candidate	DIN	PNN	DeepFM	W&D
top5	0.2255	**0.1251**	0.098	0.098	0.0982
top10	0.313	**0.21**	0.1549	0.1565	0.1551
top15	0.3618	**0.2787**	0.2254	0.2225	0.2266

- 訓練：4/1 - 5/3，測試：5/4 (top-*k* accuracy)

	Candidate	DIN	PNN	DeepFM	W&D
top5	0.2305	**0.1297**	0.0985	0.1024	0.0986
top10	0.3203	**0.214**	0.156	0.1631	0.1575
top15	0.3681	**0.2836**	0.2266	0.2329	0.2247

Offline evaluation 2: 樂天資料集 (cont')

- 訓練：4/1 - 5/4，測試：5/5 (top-*k* accuracy)

	Candidate	DIN	PNN	DeepFM	W&D
top5	0.2347	**0.1356**	0.1013	0.0992	0.1006
top10	0.322	**0.2213**	0.1594	0.157	0.1585
top15	0.3687	**0.289**	0.2248	0.2268	0.2271

Offline evaluation 3: 樂天資料集 ensemble

- 對原本舊方法的candidate排序和新方法的預測分數作權重平均
- 舊方法的candidate list產生分數
  - 排序從最開始到結束，分數由 1開始遞減
  - 遞減的分數為 1/n，n為candidate list的gid數

Offline evaluation 3: 樂天資料集 ensemble

- 訓練：4/1 - 4/30，測試：5/1 (top-*k* accuracy)

	Candidate	DIN	PNN	DeepFM	W&D
top5	0.2351	**0.252 (+7.2%)**	0.236	0.236	0.2358
top10	0.3239	**0.334 (+3.1%)**	0.3265	0.3269	0.3269
top15	0.3722	**0.379 (+1.8%)**	0.3756	0.3752	0.3755

- 訓練：4/1 - 5/1，測試：5/2 (top-*k* accuracy)

	Candidate	DIN	PNN	DeepFM	W&D
top5	0.2351	0.2376	**0.2423 (+3.1%)**	0.2415	0.2422
top10	0.3239	0.331	0.3341	0.3337	**0.3356 (+3.6%)**
top15	0.3722	0.380	0.3832	0.3826	**0.3834 (+3.0%)**

Offline evaluation 3: 樂天資料集 ensemble

- 訓練：4/1 - 5/2，測試：5/3 (top-*k* accuracy)

	Candidate	DIN	PNN	DeepFM	W&D
top5	0.2255	0.224	0.2283	0.2276	**0.2296 (+1.8%)**
top10	0.313	0.315	0.3177	0.3159	**0.319 (+1.9%)**
top15	0.3618	0.362	0.3642	0.3634	**0.3646 (+0.8%)**

- 訓練：4/1 - 5/3，測試：5/4 (top-*k* accuracy)

	Candidate	DIN	PNN	DeepFM	W&D
top5	0.2305	0.231	0.2334	**0.2338 (+1.4%)**	0.2332
top10	0.3203	0.321	0.3248	**0.3251 (+1.5%)**	0.3246
top15	0.3681	0.3684	0.3686	**0.3697 (+0.4%)**	0.3688

Offline evaluation 3: 樂天資料集 ensemble

- 訓練：4/1 - 5/4，測試：5/5 (top-*k* accuracy)

	Candidate	DIN	PNN	DeepFM	W&D
top5	0.2255	0.233	**0.2375 (+5.3%)**	0.237	**0.2375 (+5.3%)**
top10	0.313	0.323	0.3264	0.3255	**0.3268 (+4.4%)**
top15	0.3618	0.370	0.3702	0.3692	**0.3711 (+2.6%)**

使用 online recommendation：經驗1 -- 訓練方式

- 直接訓練 4/1 - 4/30 所有的 weblog 所花的時間極長
- 使用 incremental training
  - 載入到前一天為止的模型
  - 只用本日的 weblog 繼續訓練模型
  - 訓練完後存下模型 (供明日訓練時繼續載入使用)
- 實際只用一天的 weblog 做 incremental training 的訓練時間：
  - DIN : 10個epoch，每個epoch約在660-800 sec
  - PNN：10個epoch，每個epoch約在78-96 sec
  - DeepFM 10個epoch，每個epoch約在78-99 sec
  - W&D 10個epoch，每個epoch約在80-100 sec

使用 online recommendation：經驗2 -- 部署方式

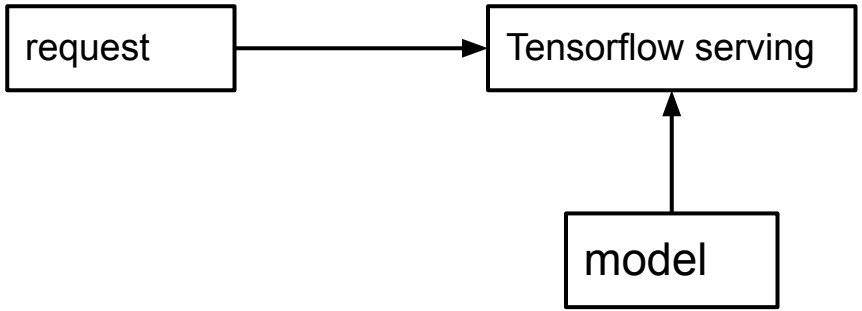
- TF Serving vs Flask

	說明	優點
TF Serving	Tensorflow 官方支援的模型部署方式	高吞吐量 (throughput)
Flask	Web 框架	對輸入/輸出做靈活的前處理/後處理

使用 online recommendation：經驗2 -- 部署方式

- TF serving :
  - 針對 model 部署成 api 的部分做過優化
  - 輸入輸出格式固定，沒有彈性，資料需要在前端做處理，

    ex：模型輸出為每個click stream pair的預測分數，排序要到前端做。

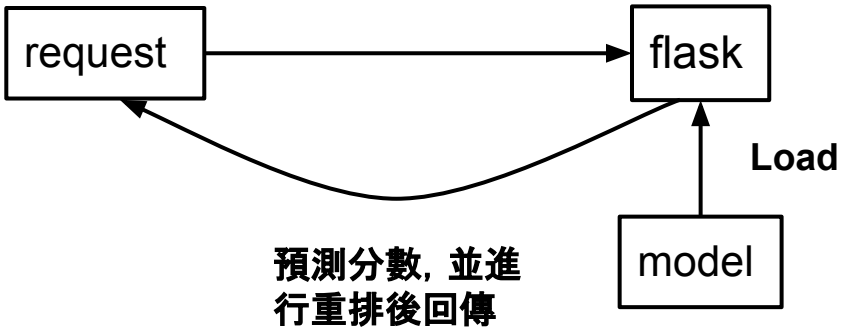


使用 online recommendation：經驗2 -- 部署方式

- Flask :
  - 輸入輸出格式有彈性，可以透過 flask調整成模型輸入的格式，

    再輸入到模型做預測，預測完的分數也是在 flask處理，

    得到重排完的推薦清單再把結果回傳到前端。
- 花費時間：2000筆request 花費185.78sec，平均0.092sec/筆

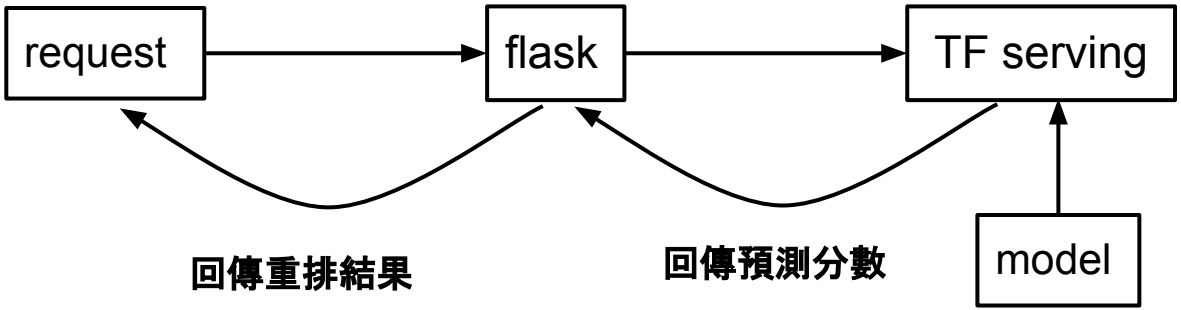


Future work -- 提高模型的 CTR

- 模型面
  - 改變 deep learning 模型 (e.g, 模型層數)
  - 調整超參數
  - 調整 ensemble 的策略
- 資料面
  - 加長訓練資料的天數 (目前只用大約一個月的訓練資料)
  - Feature engineering
- 觀察
  - 推薦成功/失敗的實例有什麼特性？

Future work -- 嘗試不同的線上部署方式

- TF serving + Flask :
  - model使用TF serving部署，輸入前先透過 flask調整成模型輸入的格式，再跟 TF serving做request，預測完的分數也是在 flask處理，得到重排完的推薦清單再把結果輸出回前端。



-END-