



# 美团实习介绍

张颖而

# 目录

- ◆ 美团联名卡获客业务 线上曝光-激活转化率提升20%
  - 项目背景
  - 单任务树模型到DNN迁移
  - 多任务模型
- ◆ 广告选择策略 线上转化率提升7%

# 美团联名卡获客业务



# 业务背景

- **目标**：帮助银行发放信用卡
- **广告投放途径**：支付完成后弹窗广告（需要与其他业务竞争该流量，因此要挑选最有可能使用信用卡的用户）
- **任务输入**：订单&用户维度特征
- **任务输出**：该订单激活信用卡的评分
- **标签**：激活-1，不激活-0

# 单任务模型

- **目标**

- 树模型—DNN；曝光—激活

- **原因**

- DNN能够实现更复杂的模型，而GBDT的模型容量比较有限。
- 新的研究成果都是基于深度学习的研究，采用深度神经网络之后可以更好的应用新的研究成果，比如后面介绍的多任务模型。

- **工作内容**

- 数据
- 模型

# 单任务模型—数据处理

**特征类型：**参考之前的业务选特征(500)，再用树模型选top权重(200)

1. 类别型 (C) :银行编号、商户号、支付通道、地域等
2. 连续型 (D) :年龄、过去180天弹窗点击次数、支付金额等

	训练集	验证集	测试集
数量	20M	3M	26M
正样本%	1%	1%	0.1%
时间顺序	前	中	后

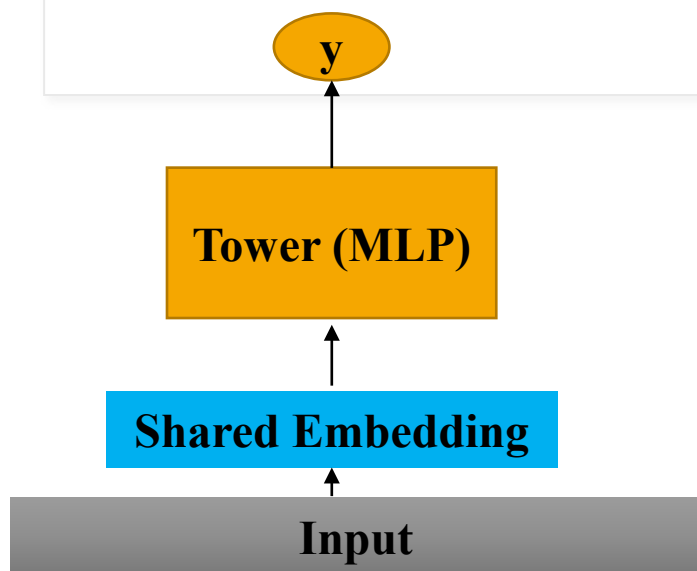
**特征处理(spark):**

1. 空值填充
2. C型特征编码后映射
3. D型特征离散化分桶

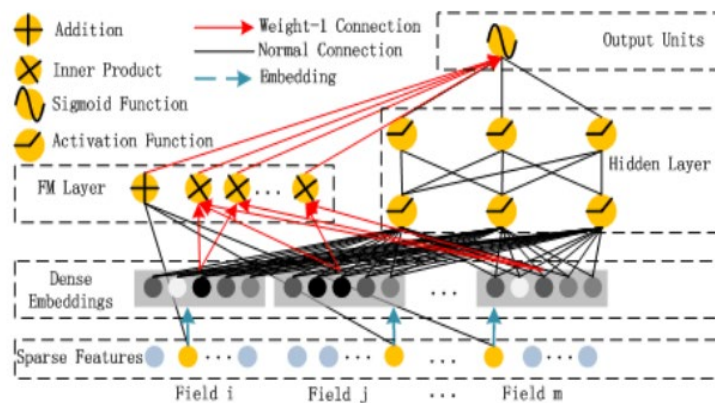
**特征构建input层:**

C类做Embedding( $d=5$ ),  
D类离散化后Embedding ;  
输出节点concat

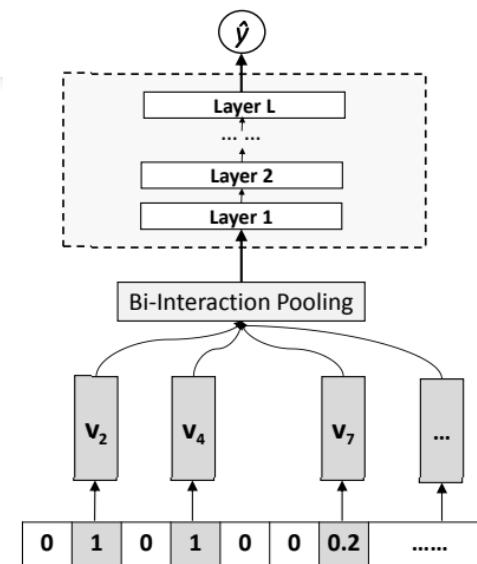
# 单任务模型——模型选择



MLP



DeepFM



NFM

Model	Industrial dataset			
	<i>approval</i> AUC	<i>activation</i> AUC	Gain	Gain
LightGBM	0.8392±0.0011	0.8536±0.0035	-	-
MLP	0.8410±0.0010	0.8602±0.0014	+0.0018	+0.0066

单任务中，DNN相比于树模型收益很小但完成了模型的迁移：数据&上线为后面多任务打基础

# 多任务模型—任务

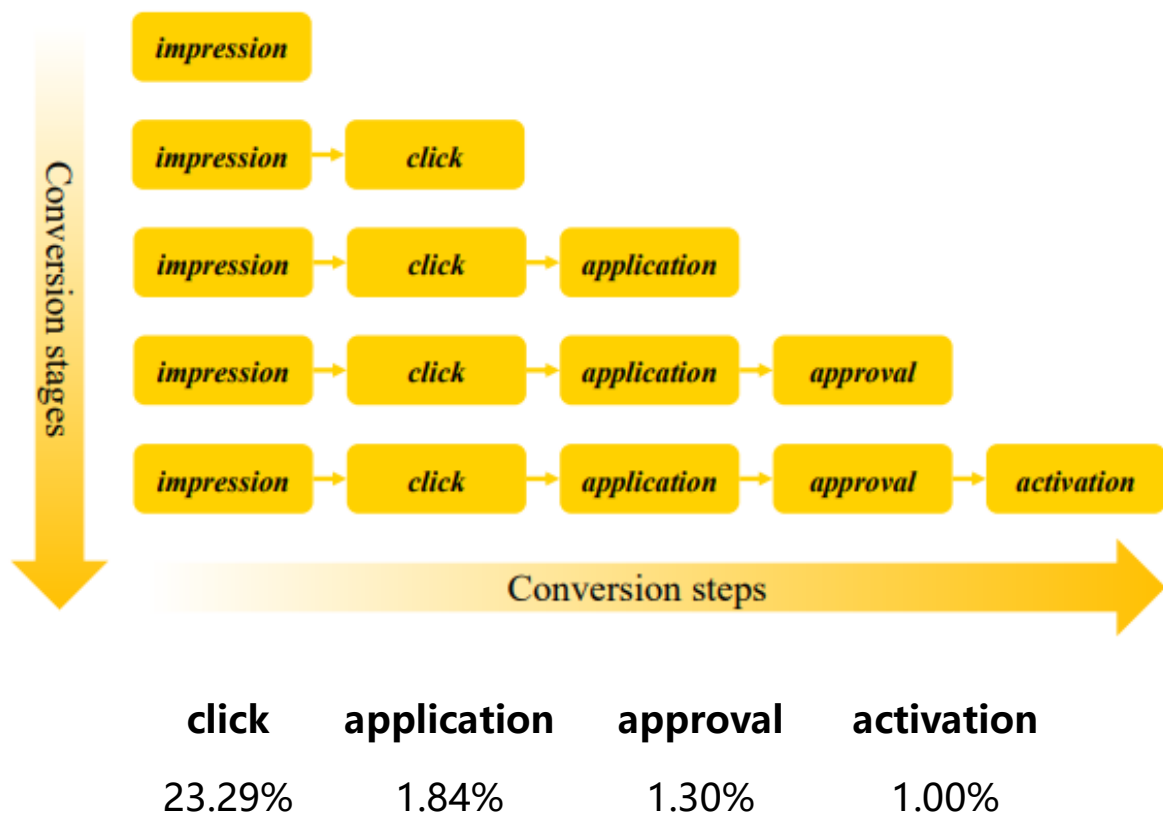
•**曝光 (Impression)**：广告被展示给了特定的用户

•**点击 (Click)**：用户如果对这个广告内容感兴趣点击这个广告，点击后，就会进入申请表格页。

•**申请 (Application)**：进入申请表格页后用户可以填写申请表格

•**核卡 (Approval)**：也称授信，意味着用户信用良好，通过申请并被授予了一定的信用卡额度。

•**激活 (Activation)**：用户在授信并且收到邮寄的信用卡之后，可以激活信用卡并使用。在这里，我们通常看用户是否会在核卡后14天内激活信用卡。

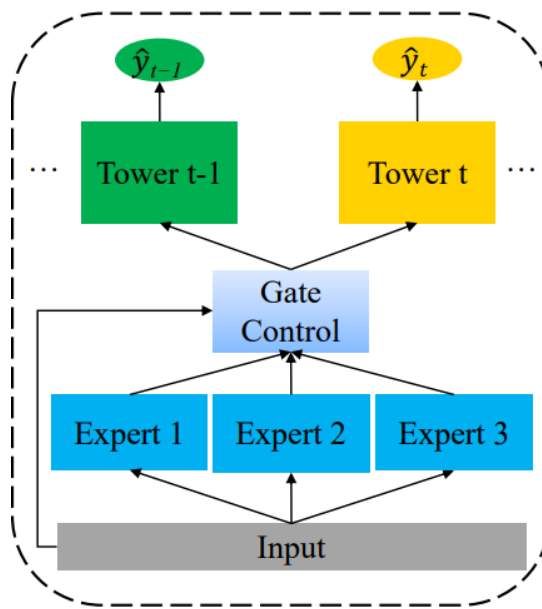


## 主要任务

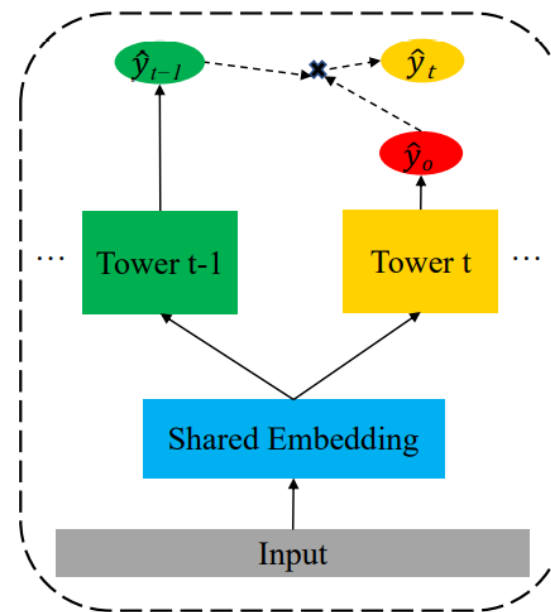


# 多任务模型—模型探索

Model	<i>approval</i> AUC	Industrial dataset <i>activation</i> AUC	Gain	
LightGBM	0.8392±0.0011	0.8536±0.0035	-	-
MLP	0.8410±0.0010	0.8602±0.0014	+0.0018	+0.0066
ESMM	0.8443±0.0028	0.8691±0.0025	+0.0051	+0.0155
MMoE	0.8444±0.0026	0.8705±0.0009	+0.0052	+0.0169

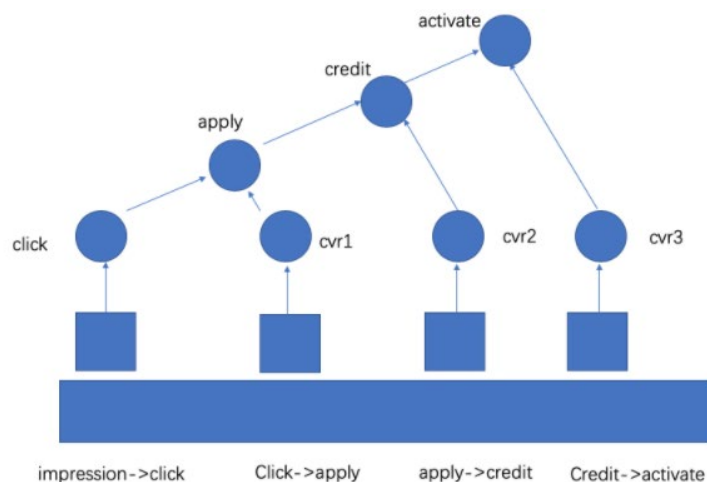


Expert-Bottom pattern  
(MMOE)

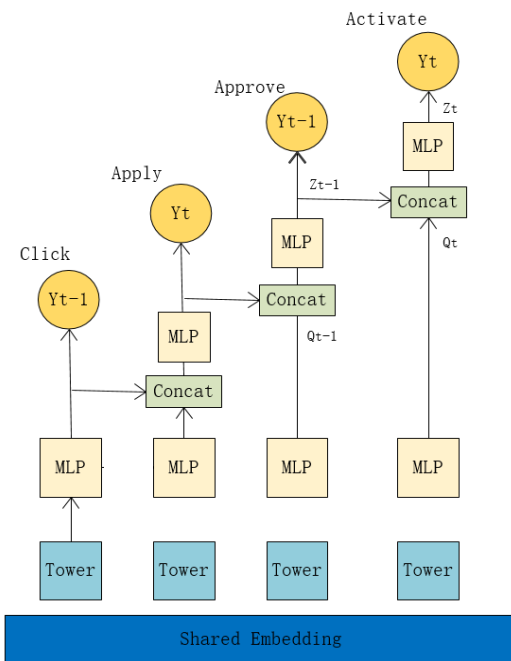


Probability-Transfer pattern  
(ESMM)

# 多任务模型—ESMM 拓展探究

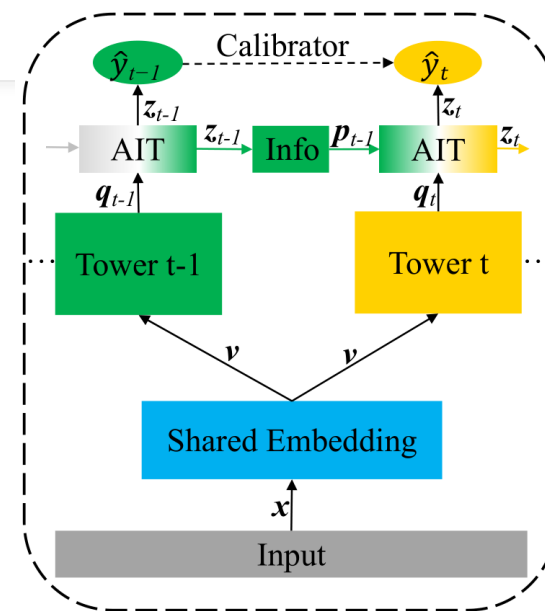


ESMM v1



ESMM v2

离线AUC比ESMM v1高 0.3%,  
线上转化率比树模型提升20%



ESMM v3  
AITM

# 多任务模型—AITM

Xi, D., Chen, Z., Yan, P., **Zhang, Y.**, Zhu, Y., Zhuang, F., & Chen, Y. Modeling the sequential dependence among audience multi-step conversions with multi-task learning in targeted display advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (pp. 3745-3755).

AIT模块利用当前任务Tower输出的向量 $q_t$ 以及前一个任务传来的信息 $p_{t-1}$ 来学习任务间如何融合信息。

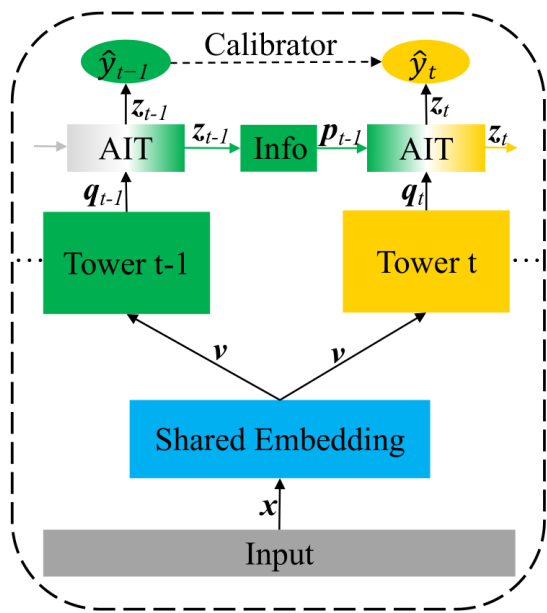
$$z_t = \text{AIT}(p_{t-1}, q_t),$$

$$\text{where } p_{t-1} = g_{t-1}(z_{t-1}),$$

具体地，AIT模块中的注意力机制如下设计：

$$z_t = \sum_{\mathbf{u} \in \{p_{t-1}, q_t\}} w_u h_1(\mathbf{u}),$$
$$w_u = \frac{\exp(\hat{w}_u)}{\sum_u \exp(\hat{w}_u)}, \quad \hat{w}_u = \frac{\langle h_2(\mathbf{u}), h_3(\mathbf{u}) \rangle}{\sqrt{k}}.$$

而迁移的信息是通过函数 $g_{t-1}$ 来学习的，这里可以是一个简单的全连接层，用来学习两个相邻的任务间应该迁移什么信息。



# 多任务模型—AITM

Model	<i>approval</i> AUC	Industrial dataset <i>activation</i> AUC	Gain	
LightGBM	0.8392±0.0011	0.8536±0.0035	-	-
MLP	0.8410±0.0010	0.8602±0.0014	+0.0018	+0.0066
ESMM	0.8443±0.0028	0.8691±0.0025	+0.0051	+0.0155
OMoE	0.8438±0.0022	0.8714±0.0009	+0.0046	+0.0178
MMoE	0.8444±0.0026	0.8705±0.0009	+0.0052	+0.0169
PLE	<u>0.8518±0.0006</u>	<u>0.8731±0.0016</u>	+0.0126	+0.0195
AITM	<b>0.8534±0.0011**</b>	<b>0.8770±0.0005*</b>	<b>+0.0142</b>	<b>+0.0234</b>

Model	Gain	
	<i>approval</i>	<i>activation</i>
MLP vs LightGBM	+16.95%	+17.55%
AITM vs MLP	+25.00%	+42.11%

# 广告选择策略



# 业务场景

- **目标：**针对联名卡的多种广告（经常更新），选择总收益最高的流量分配策略
- **旧策略：**80%流量给当前最优广告，20%流量随机剩余的广告
- **新策略：**Exploit & Explore 多臂老虎机实验仿真，个体使用何种搜索策略可以最大化其累及即时奖励。

# 多臂老虎机实验—实验设置

- 每个老虎机的摇臂数=广告数量
- 老虎机每个臂的真实概率=广告真实转化率
- 拉动试验次数=投放次数

Step1.初始化老虎机每个臂的概率

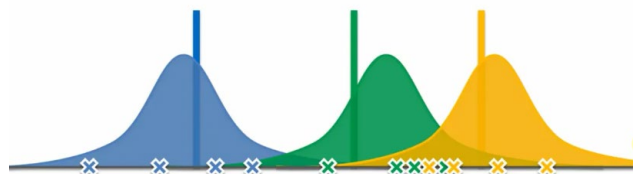
Step2.根据策略 (random, greedy, e-greedy, ucb, thompson) 选择一个臂

Step3.对该臂进行伯努利实验：发生或者不发生，更新参数

Step4.计算总共命中次数

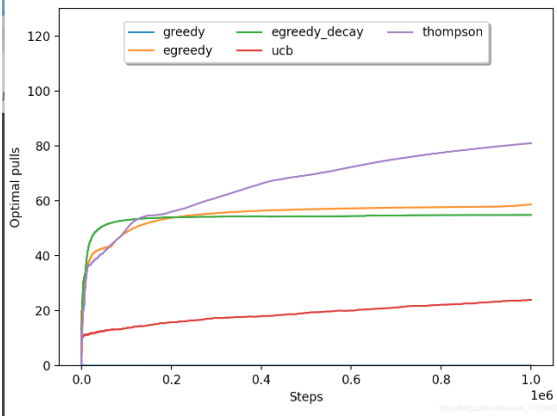
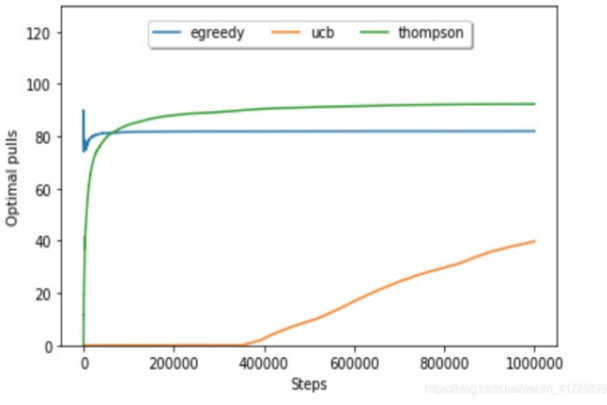
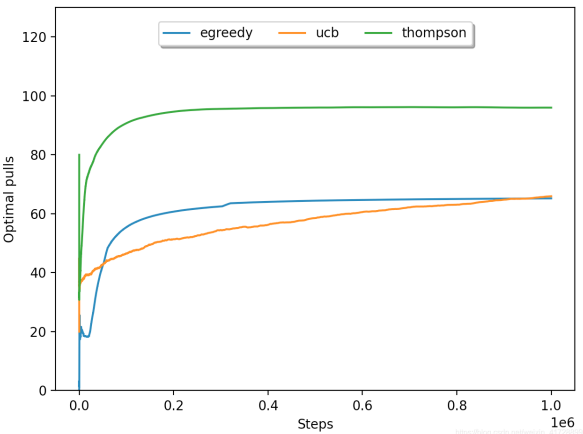
# 多臂老虎机实验—策略

策略	选择策略
Random	随机选择
Greedy	选择平均值最大的臂
E-greedy	一部分选择平均值最大的臂；另一部分随机选
Ucb	t : 总次数 Nt(a): 广告A被投放的次数 $A_t \doteq \arg \max_a \left[ Q_t(a) + c \sqrt{\frac{\ln t}{N_t(a)}} \right]$
Thompson	$\beta$ 分布(a,b) 均值=a/(a+b) a+b越大, 则分布越窄 $\beta(a_0+hits, b_0+misses)$ a推荐后用户点击的次数; b参数看成是推荐后用户未点击的次数





# 多臂老虎机实验—实验结果

<b>实验情景</b>	冷启动	线上仿真 7个旧广告+3个新广告 假设旧广告已经曝光100万次 新广告概率均小于旧广告	线上仿真 7个旧广告+3个新广告 假设旧广告已经曝光100万次 新广告概率均大于旧广告
<b>结果</b>	 <p>Line graph showing Optimal pulls vs Steps for cold start scenario. The x-axis is Steps (0 to 1.0e6) and the y-axis is Optimal pulls (0 to 120). The legend includes greedy (blue), egreedy (orange), egreedy_decay (green), ucb (red), and thompson (purple). Thompson (purple) reaches the highest value of approximately 80. Greedy (blue) and egreedy (orange) reach approximately 55. Egreedy_decay (green) reaches approximately 50. Ucb (red) reaches approximately 25.</p>	 <p>Line graph showing Optimal pulls vs Steps for online simulation with lower probabilities. The x-axis is Steps (0 to 1,000,000) and the y-axis is Optimal pulls (0 to 120). The legend includes egreedy (blue), ucb (orange), and thompson (green). Thompson (green) reaches the highest value of approximately 90. Egreedy (blue) reaches approximately 80. Ucb (orange) reaches approximately 40.</p>	 <p>Line graph showing Optimal pulls vs Steps for online simulation with higher probabilities. The x-axis is Steps (0 to 1.0e6) and the y-axis is Optimal pulls (0 to 120). The legend includes egreedy (blue), ucb (orange), and thompson (green). Thompson (green) reaches the highest value of approximately 95. Egreedy (blue) reaches approximately 65. Ucb (orange) reaches approximately 65.</p>

**Thompson 策略上线后转化率提升7%**