互联网数据下的模型探索

盖坤 - 阿里妈妈
2017/07/09

提纲

1. 互联网数据和经典模型
2. 分片线性模型和学习算法MLR
3. 大规模ID特征+MLR实践
4. 深层用户兴趣分布网络

互联网数据

- 典型问题：CTR预估
- 数据特点
  - 样本量大 百亿样本
  - 特征维度大 无损表示—id特征 原始特征轻松超十亿级
  - 稀疏数据
- 经典做法
  - 简单线性模型Logistic Regression
  - 稀疏正则 L1-Norm 特征筛选
  - 处理非线性：人工特征工程
- 问题
  - 人工能力有限，很难对非线性模式挖掘完全充分
  - 依赖人力和领域经验，方法推广到其它问题的代价大：不够智能

已有非线性模型

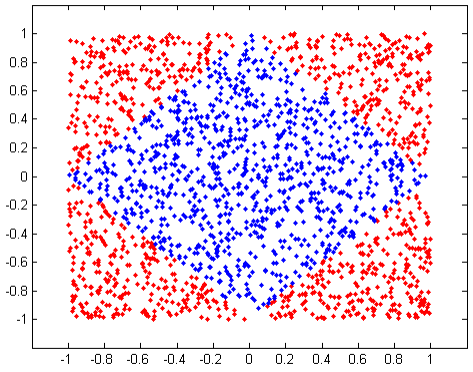
- 分析已有的非线性算法 @2011
  - Kernel方法（kernel svm）：复杂度太高
  - Tree based方法（例GBDT）
    - 大规模弱特征上表现为记忆历史行为 缺乏推广性
    - 跟到叶子路径：if(user id==useri && item id == itemj)条件判断
  - 矩阵分解(Topic Model, LDA等)
    - 适用于两种id的情况，不适合多种id输入
  - Factorization machines: $f(x) = \sum_{i,j} < v_i, v_j > x_i x_j$
    - 只拟合有限次关系(二次关系)
    - 无法拟合其它非线性关系：例如三种特征的交叉，值的高阶变换等。

- 需要的特性
  - 足够强的非线性拟合能力
  - 良好的泛化能力
  - 规模化能力

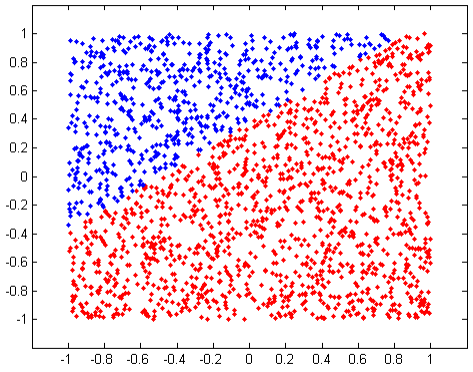
分片线性模型

- 挑战：如何从大规模数据中挖掘出推广性好的非线性模式？
- 我们提出：分片线性学习算法MLR @2011
  - 名称:Mixture of lr (MLR)
  - 任意强非线性拟合能力
  - 模型复杂度可控（分片数）
    - 平衡欠拟合和过拟合
    - 每分片对应足够量样本，并用线性规律拟合，得到好的推广性
  - 适合大规模化高维度数据，并有特征选择能力

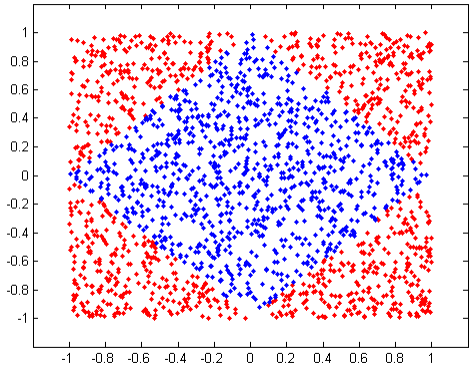
例子：分类问题



训练数据



线性模型(LR)



分片线性模型

模型形式

- 分而治之形式

聚类划分　　划分内预测

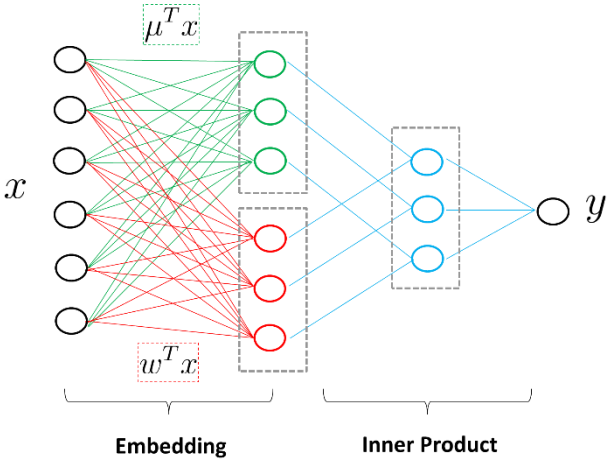
$$f(x) = g\left(\sum_i \pi_i(x,\mu)\eta_i(x,w)\right)$$

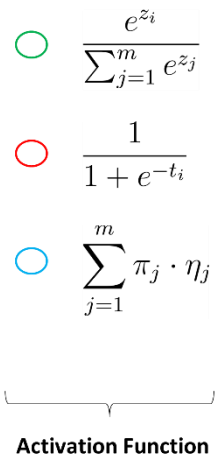
- 主要在用的形式：

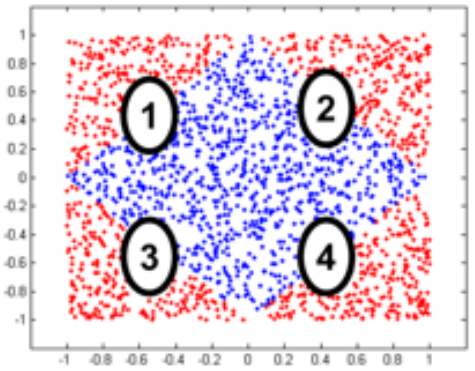
$$f(x) = \sum_{i=1}^{m} \frac{e^{\mu_i \cdot x}}{\sum_{j=1}^{m} e^{\mu_j \cdot x}} \cdot \frac{1}{1+e^{-w_i \cdot x}}$$

$$f(x) = \left(\sum_{i=1}^{m} \frac{e^{\mu_i \cdot x_u}}{\sum_{j=1}^{m} e^{\mu_j \cdot x_u}} \cdot \frac{1}{1+e^{-w_i \cdot x_a}}\right) \cdot \frac{1}{1+e^{-w \cdot x_2}}$$

  - Softmax划分，LR预测[MOE]　　MOE&LR级联
  - 神经网络视角:







如何学习参数

- 模型形式： $f(x;\theta)=\sum_{i=1}^{m}\frac{e^{\mu_i\cdot x}}{\sum_{j=1}^{m}e^{\mu_j\cdot x}}\cdot\frac{1}{1+e^{-w_i\cdot x}}$ $f(x)=\left(\sum_{i=1}^{m}\frac{e^{\mu_i\cdot x_u}}{\sum_{j=1}^{m}e^{\mu_j\cdot x_u}}\cdot\frac{1}{1+e^{-w_i\cdot x_a}}\right)\cdot\frac{1}{1+e^{-w\cdot x_2}}$
- 参数矩阵: $\theta=[w_1,...,w_m,\mu_1,...,\mu_m]$
- 分类经验损失： $l(f(x_i;\theta),y_i)=-\big(y_i\log(f(x_i;\theta))+(1-y_i)\log(1-f(x_i;\theta))\big)$
- 特征选择：同一维度对应多个权重 — 分组稀疏正则

$$\|\theta\|_{2,1}=\sum\nolimits_i\sqrt{\sum\nolimits_k\theta_{ik}^2}$$

- 目标函数:

$$\min_{\theta}\quad g(\theta)=\sum\nolimits_i l(f(x_i;\theta),y_i)+\lambda\|\theta\|_{2,1}+\beta\|\theta\|_1$$

目标函数分析

- 目标函数:

$$\min_{\theta} \quad g(\theta) = \sum_i l(f(x_i;\theta), y_i) + \lambda\|\theta\|_{2,1} + \beta\|\theta\|_1$$

- 难度和挑战：
  - 非凸
  - 非光滑（不可导，不存在次梯度）
  - 实际面对超大规模数据，高维度

- 我们提出针对非凸非光滑目标的快速优化方法
  - 证明处处方向可导
  - 寻优最优下降方向：解析解
  - 拟牛顿法加速
  - Scalability: 计算量对数据量线性

- Why not EM ?
  - EM只适用于条件概率连乘模型形式，而我们的方法对非正则部分可导的形式通用。
  - E-Step后转化为一个凸问题。
  - 参数小，无非光滑正则时这个凸问题可以用牛顿法求解（MOE算法）。
  - 但我们的情况：1. 参数维度特别大，2. 有非光滑正则。凸问题难度不比原问题小。
  - EM没有带来便利。

MLR算法

- 目标函数: $f(\Theta) = loss(\Theta) + \lambda\|\Theta\|_{2,1} + \beta\|\Theta\|_1$
- 分析：
  - 非光滑范数导致不可导
  - 可以证明：处处方向可导
  - 寻找最速下降方向？
- 最速下降方向：

$$d_{ij} = \begin{cases} s - \beta sign(\Theta_{ij}), & \Theta_{ij} \neq 0 \\ \max\{|s| - \beta, 0\} sign(s), & \Theta_{ij} = 0, \|\Theta_{i\cdot}\|_{2,1} \neq 0 \\ \frac{\max\{\|v\|_{2,1} - \lambda, 0\}}{\|v\|_{2,1}} v, & \|\Theta_{i\cdot}\|_{2,1} = 0, \end{cases}$$

  - 其中

$$s = -\nabla loss(\Theta)_{ij} - \lambda \frac{\Theta_{ij}}{\|\Theta_{i\cdot}\|_{2,1}}$$

$$v = \max\{|-\nabla loss(\Theta)_{ij}| - \beta, 0\} sign(-\nabla loss(\Theta)_{ij})$$

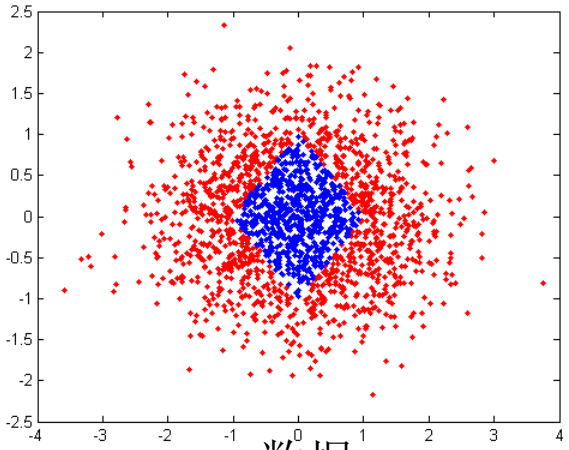
- 整体算法：
  - 拟二阶加速：基于最速下降方向（代替负梯度）的L-BFGS 做方向修正
  - 象限约束：一次更新不跨象限，变号则用0截断 [as OWL-QN]
  - Line Search确定步长
  - 一阶补足保证收敛：二阶方向无法下降时，弃用二阶，用最速下降法
  - 结束：最速下降法无法下降时

MLR特性

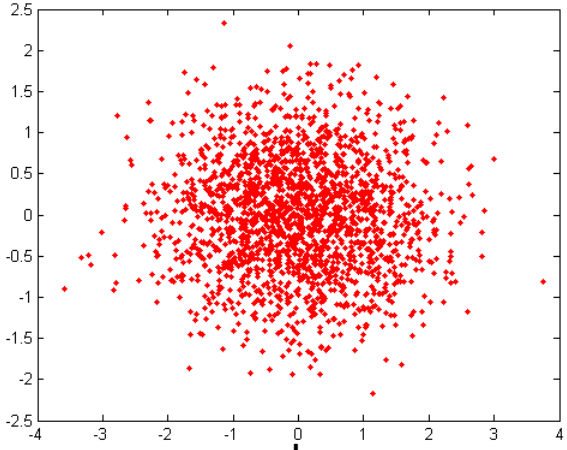
- 特点
  - 分而治之
  - 分片数足够多时，有非常强的非线性能力
  - 模型复杂度可控：有较好泛化性能
  - 具有自动特征选择作用
  - 可以适用于大规模高维度数据

- 工程实现
  - 数据并行，模型并行
  - MPI协议，支持独立部署或部署于ODPS
  - 实际运行例子：数亿特征（MLR无需人工交叉，特征膨胀不严重），数百亿样本，分片数m=12。每个job 150台机器。

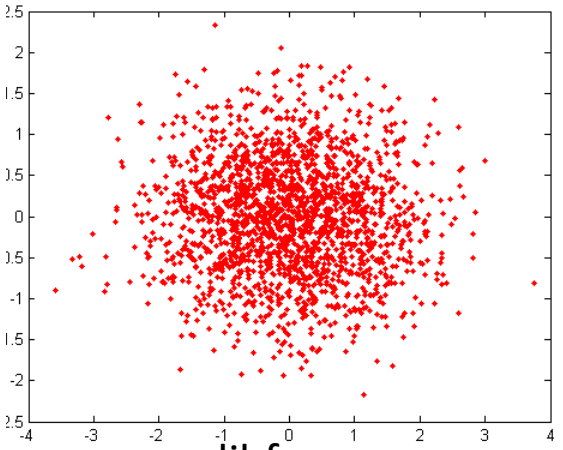
实验1：聚类和分类联动



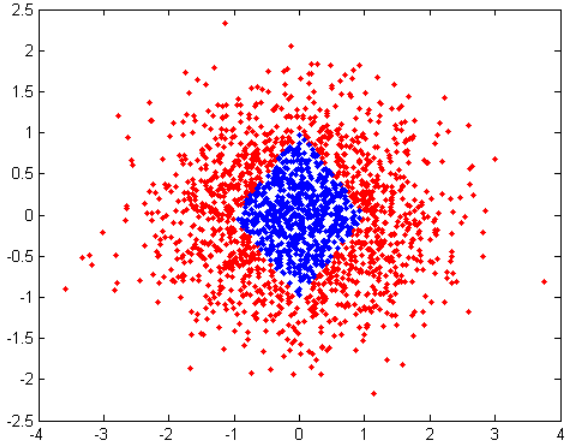
数据



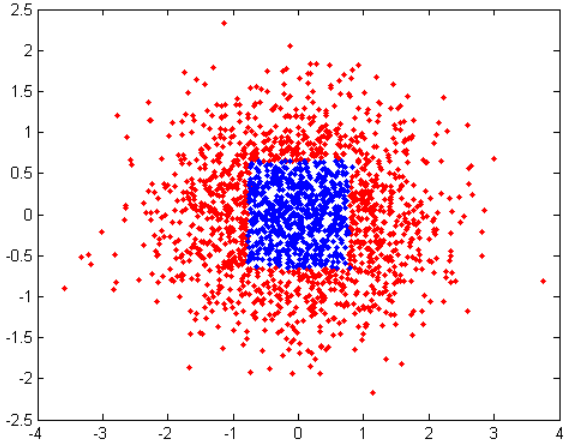
lr



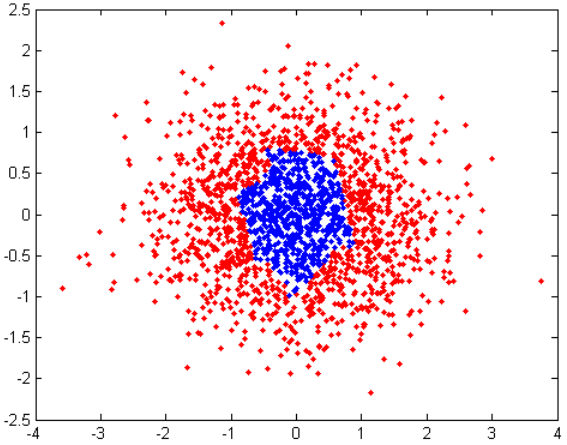
libfm



mlr(4分片)



Kmeans+lr(4 clusters)



Kmeans+lr(10 clusters)

实验2：高阶拟合

- 三种id特征交叉实验：

特征1	特征2	特征3	类别	mlr预测概率(2分片)	Libfm预测概率（topic数20）
0	0	0	1	0.999954	0.507501
1	0	0	0	0.000050	0.494845
0	1	0	0	0.000058	0.496506
0	0	1	0	0.000045	0.494050
0	1	1	1	0.999965	0.501566
1	0	1	1	0.999969	0.491462(分错)
1	1	0	1	0.999927	0.502185
1	1	1	0	0.000038	0.520136(分错)

- mlr具有更强的非线性拟合能力，不局限于二次函数
  - 可以在更少的参数下拟合更高阶非线性信息(例如多类id交叉)

MLR:模型对比

- MLR vs. LR
  - 例子: 推荐宝贝ranking数据

测试AUC	预估ctr	预估pcvr
LR	0.700112	0.748859
MLR(m=51)	**0.713173**	**0.775776**
提升值(百分点绝对值)	**+1.3061**	**+2.6917**

- MLR vs. GBDT(boosting)
  - 低维数据对比

AUC	GBDT (tree:800,depth:6)	MLR（m=50）	MLR（m=150,未收敛模型）	MLR(150，收敛模型)
训练集（前一天）	0.664416	0.660369	0.664559	0.666423
测试集（后一天）	0.661497	0.665067	0.665884	0.667163

  - 此外,GBDT不适合超高维度数据

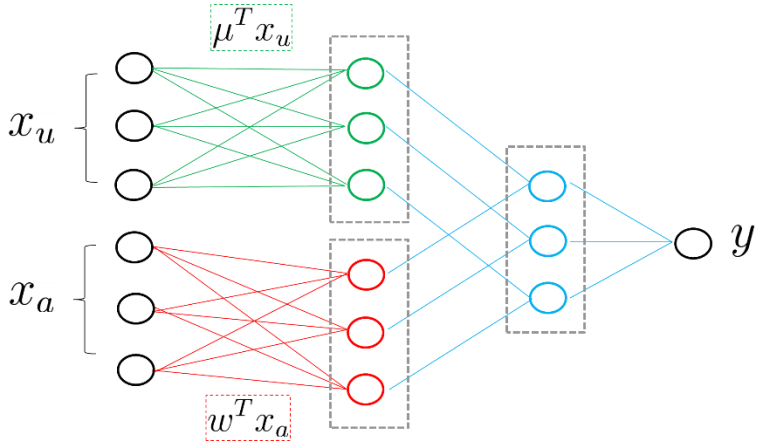
大规模id特征+MLR实践

- 任务：
  - 预估(user，item)的CTR
- 特征设置：
  - 用户行为：访问/收藏/购买过 的 shopid/categoryid
  - 用户属性：性别、年龄、地域等
  - Item特征：itemid/shopid/categoryid
- 行为id vs. 用户id
  - 用户id→用户兴趣点：用户兴趣点 拟合 训练数据中的目标item。历史记忆属性重。
  - 行为id：行为id→行为兴趣点 拟合 目标item。学习的是行为→后续兴趣的模式。更具泛化性。
  - 我们倾向行为id做特征。但是用户id在训练时可以用来做兴趣点的偏置项。
- 模型算法：MLR
  - 对比LR，AUC提升1个点以上。
  - 测试AUC对比（LS-PLM代表MLR算法）

Model	1	2	3	4	5	6	7
LS-PLM	0.6645	0.6593	0.6588	0.6620	0.6606	0.6596	0.6594
LR	0.6499	0.6445	0.6444	0.6467	0.6467	0.6463	0.6454

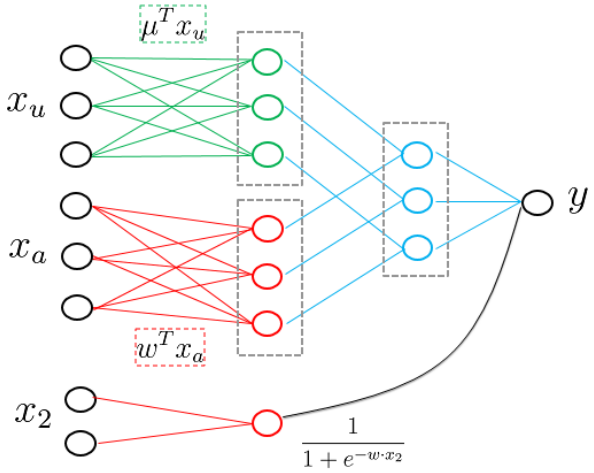
结构化先验

- 特征分组：
  - 用户特征只用来聚类
  - Item特征只用来分类
  - 实测：
    初始分组训练+全放开refine
    优于　　分组训练
    优于　　直接全放开训练



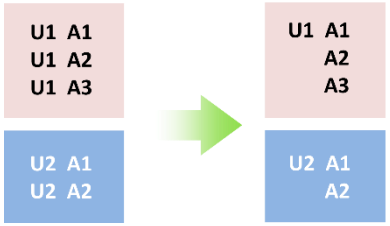
- 线性偏置
  - 位置偏差
  - 强特征（例如级联模型的输出）
  - 这些特征放在单独线性sigmoid中

$$f(x) = \left(\sum_{i=1}^{m} \frac{e^{\mu_i \cdot x_u}}{\sum_{j=1}^{m} e^{\mu_j \cdot x_u}} \cdot \frac{1}{1+e^{-w_i \cdot x_a}} \right) \cdot \frac{1}{1+e^{-w \cdot x_2}}$$



Common Feature

- 数据冗余：
  - 同一个用户的多个样本，特征中有大量重复的特征段
    - 同一个pv展示多个推荐宝贝，多个样本的用户特征完全相同。
    - 一个用户不同时刻的样本，用户特征中也有大量冗余部分
  - 平铺成样本向量时，同样数据复制多次
    - 用户部分有值id平均数量 远大于 item部分有值id平均数量 浪费严重
- 计算冗余：相同特征段的Embedding叠加的计算也是相同的
- Common Feature



- 实际对比

Dataset	Without Comm. Feat.	With Comm. Feat.
Memory used per worker	89.2 GB	3.1 GB
Total time per ite.	121s	10s

- 应用：
  - 持续迭代优化以MLR为核心的预估模型，是直通车定向、钻展广告等业务线近几年收入能力提升的主要动力之一

深度学习

- 我们如何看深度学习：
  1. 优化方法标准化，模型设计和优化方法解耦
  2. 模型设计组件化
  - 以上两个特点使得我们可以设计以往难以handle的复杂模型

- 复杂就足够了么？
  - 浅层模型：例如单隐层神经元，近邻法，Kernel方法，可以任意复杂
    - 缺点：记忆性强，泛化能力不够
  - 哪些可能影响泛化能力：
    - 深度 vs. 宽度
    - 网络结构和数据匹配度：CNN 、 LSTM 等

- 我们试图回答：
  - 互联网数据上应该有什么样的网络结构组件？

用户兴趣表示

- 目前流行的处理方法：
  - 同一类用户行为：一组行为id→一组embedding向量→Pooling/RNN→固定长度向量
    也称为用户兴趣点
  - 目标ad：adId→embedding向量
    宝贝兴趣点
  - 后面接交叉处理单元。最简单的：内积。或：多层神经网络。

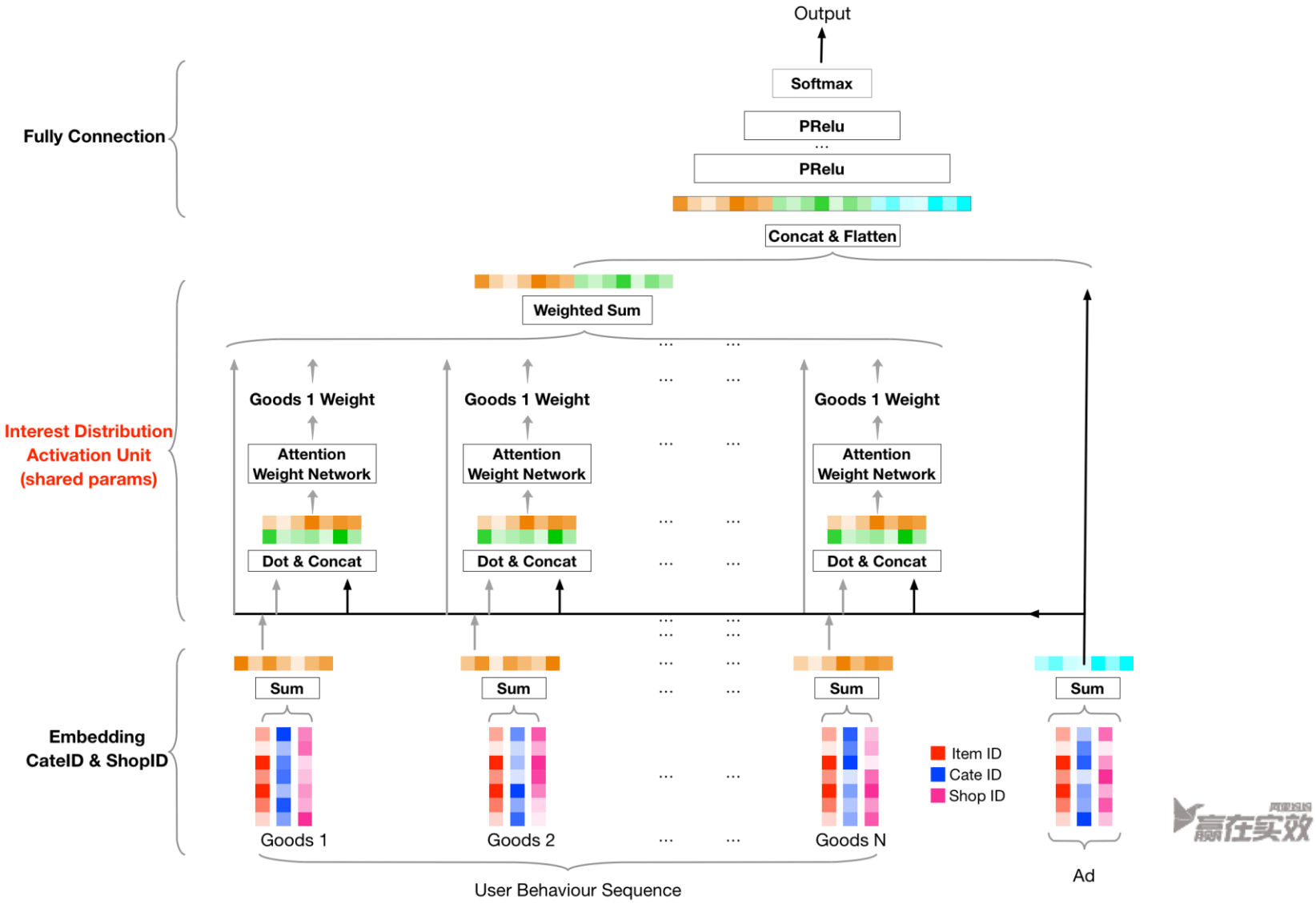
- 用户兴趣用一个向量表示：
  - k维向量最多表达k个独立兴趣　　but　　实际独立兴趣可能有很多
  - 简单办法：增大k。　　but　　极大增大计算负担，并且导致过拟合！
- 我们的动机：
  - 能否在低维兴趣空间中表达复杂的用户兴趣？
- 方法：用户兴趣不再用一个点表示，转而用一个分布
  - 分布可以是任意多峰的，可表达任意多独立兴趣
  - 宝贝仍在同一个低维空间中表达
  - 不正交也可以表达独立的兴趣，增加了低维兴趣空间的容纳能力
  - 例子：二维空间，想象12个时钟方向可以表示12个独立兴趣
    - 甚至1维空间可以表达无限多独立兴趣！

用户兴趣分布

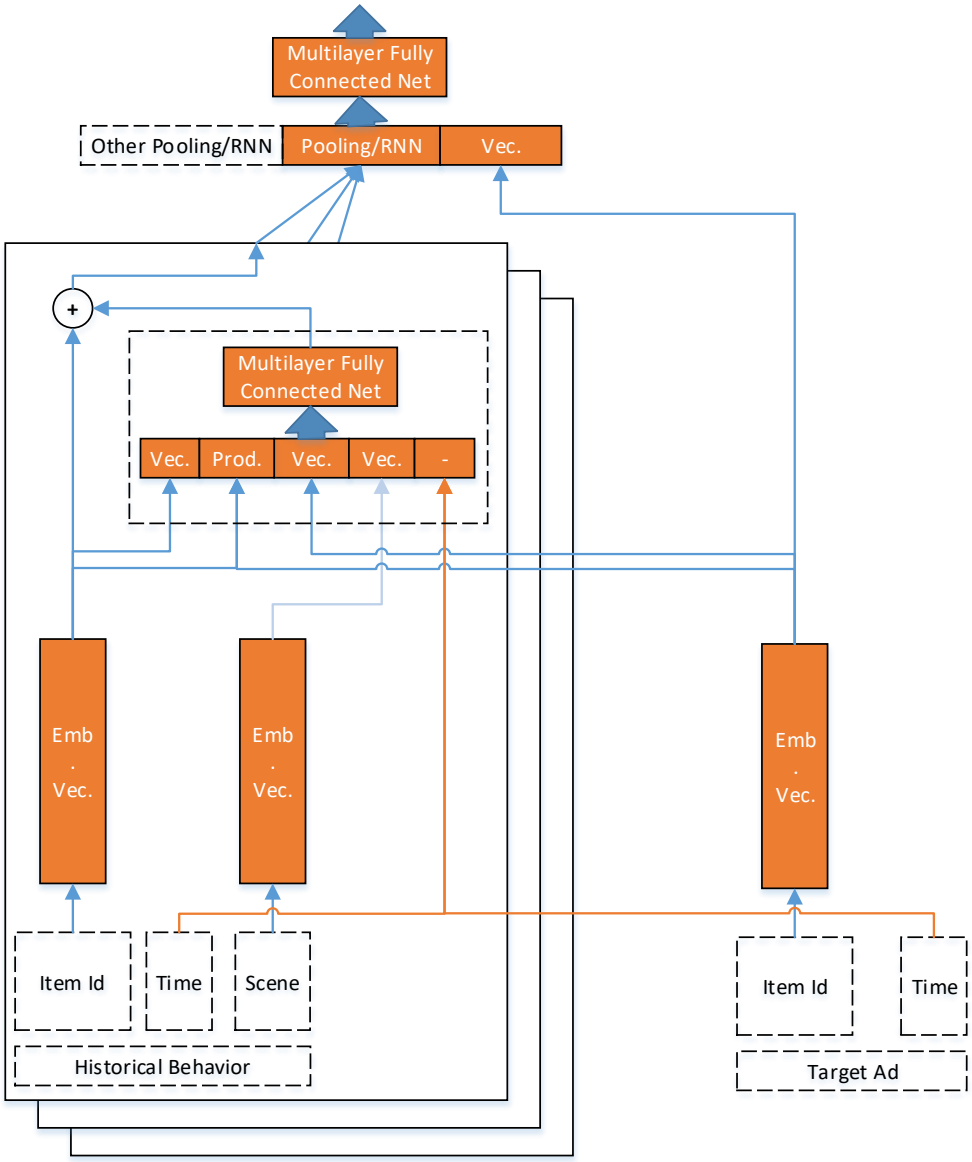
- 电商用户行为特点：
  - Diversity：多需求并发　行为序列是多个需求子序列的并集
  - Local Activation：你在具体注意到某个商品时，决定通常只其中一个或部分需求有关
- 用户兴趣分布：
  - 用户兴趣强度：$V_u(x)$，表现出的兴趣向量随测试点x不同而变
  - 预估场合里，x就是我们要去预估的item的兴趣点
- 用户兴趣点：
  - $V_u = \sum_i V_i$
  - $V_u(x) = \sum_i f(V_i, V_x) V_i$

  - 用户兴趣点是行为兴趣点的叠加
  - 非统一叠加，权重依赖正在看的商品
  - 根据目标商品，反向激活和过滤用户历史行为，只剩下相关的行为子序列
  - 如何反向激活和过滤，根据数据学习
  - 等价于Attention机制

深层用户兴趣分布网络


Output
Softmax
PRelu
...
PRelu
Concat & Flatten
Fully Connection
Weighted Sum
Goods 1 Weight
Goods 1 Weight
Goods 1 Weight
Attention Weight Network
Attention Weight Network
Attention Weight Network
Dot & Concat
Dot & Concat
Dot & Concat
Interest Distribution Activation Unit (shared params)
Sum
Sum
Sum
Sum
Embedding CateID & ShopID
Item ID
Cate ID
Shop ID
Goods 1
Goods 2
Goods N
Ad
User Behaviour Sequence

利用结构化数据


Multilayer Fully Connected Net
Other Pooling/RNN
Pooling/RNN
Vec.
+
Multilayer Fully Connected Net
Vec.
Prod.
Vec.
Vec.
-
Emb . Vec.
Emb . Vec.
Emb . Vec.
Item Id
Time
Scene
Historical Behavior
Item Id
Time
Target Ad

自适应正则

- 过拟合问题
  - 参数量极大 and 模型复杂
- 正则 vs. 稀疏
  - 0值特征理论上也有正则计算，则计算不可接受
- 我们的方法：

$$w_i \leftarrow w_i - \eta \left[\frac{1}{b} \sum_{(x_j, y_j) \in B} \frac{\partial L(f(x_j), y_j)}{\partial w_i} + \lambda \frac{1}{n_i} w_i I_i \right]$$

$$I_i = \begin{cases} 1, \exists (x_j, y_j) \in B, s.t. [x_j]_i \neq 0 \\ 0, \text{other wises} \end{cases}$$

  - 其中x_j, y_j表示第j个样本和标签，i是特征维度下标，w_i表示第i维特征对应的参数向量，n_i表示第i维特征的非零总频次。
  - 特征出现频次越高，单次正则压制约小；频次越低，单次正则压制越大

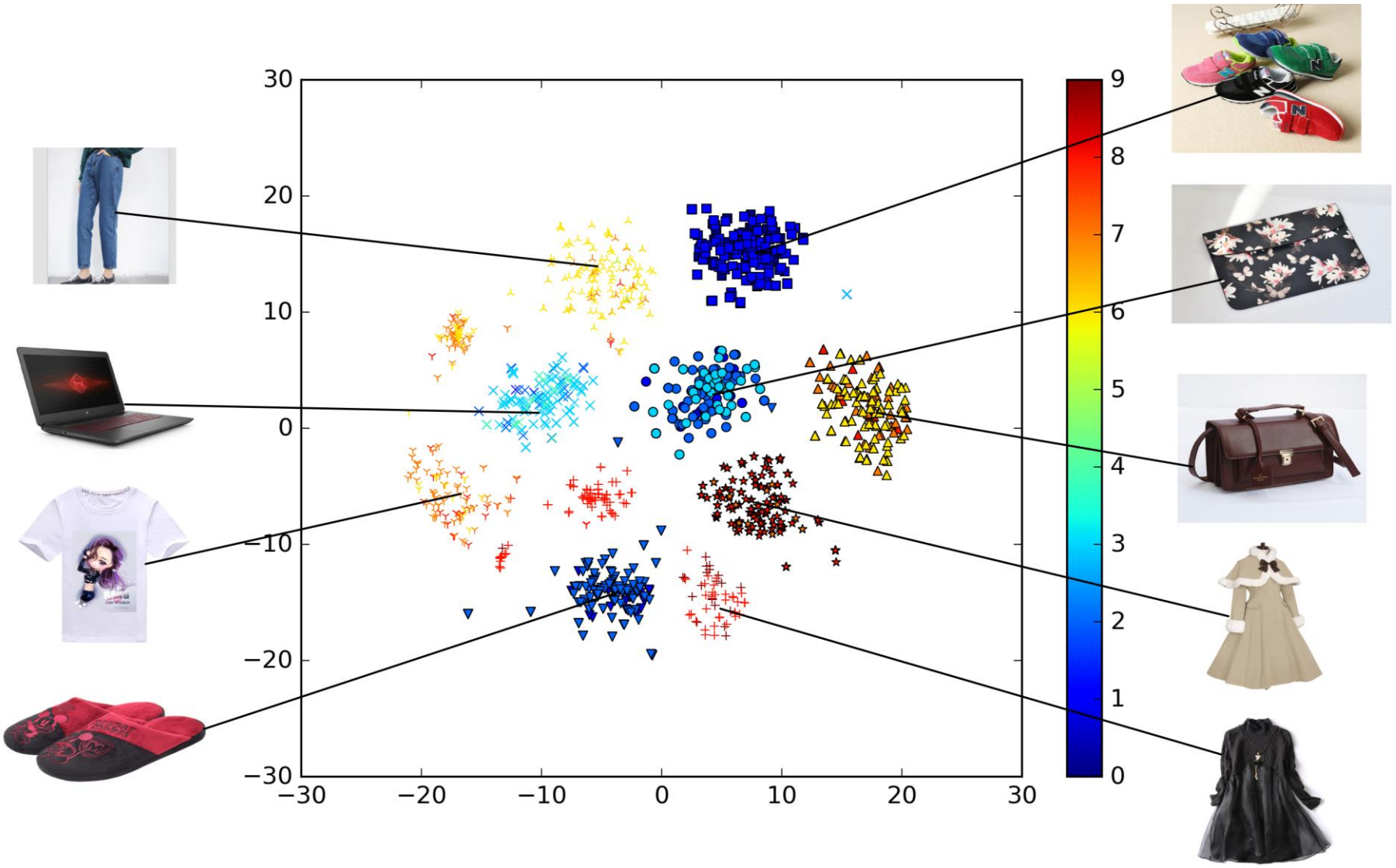
- 激活函数Dice：Prelu的改进

$$y_i = a_i(1 - p_i)y_i + p_i y_i$$

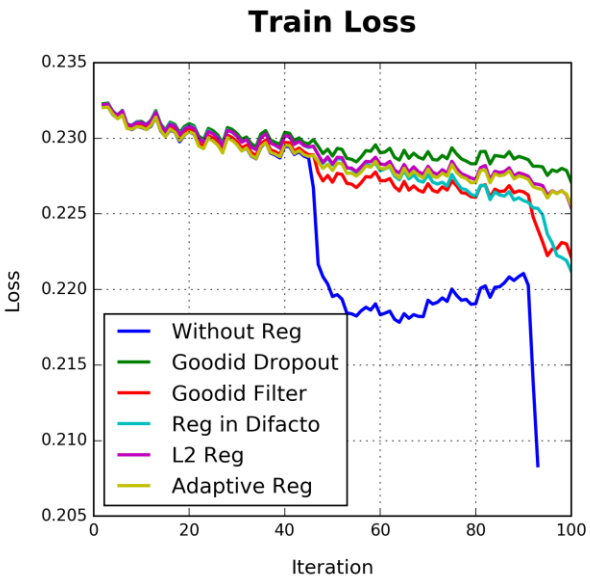
$$p_i = \frac{1}{1 + e^{-\frac{y_i - E[y_i]}{\sqrt{Var[y_i] + \epsilon}}}}$$

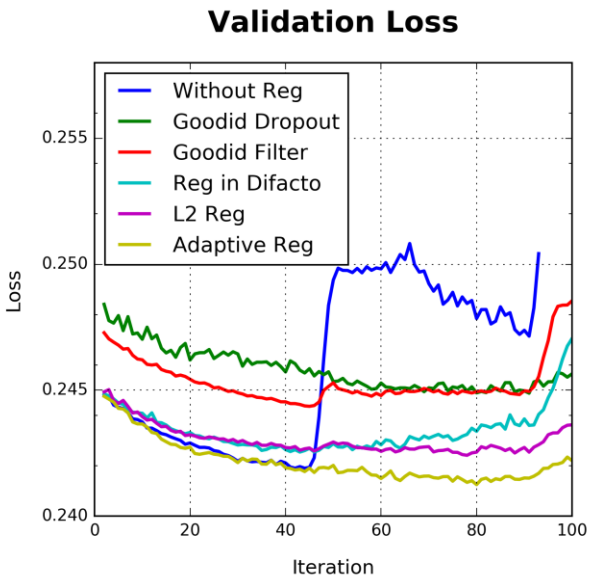
激活权重展示

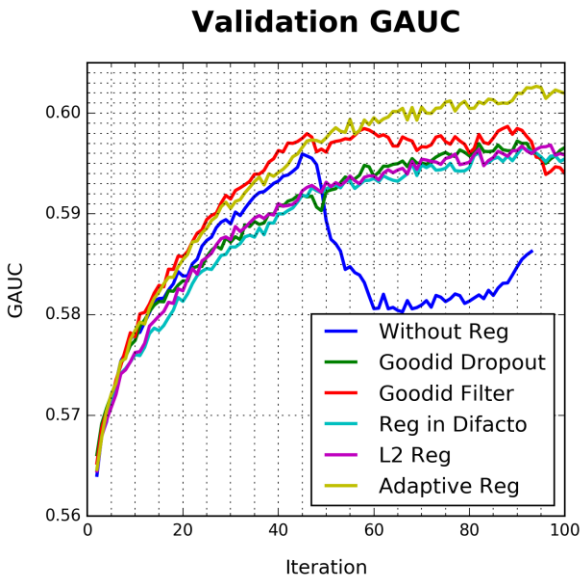
用户兴趣分布展示



正则效果



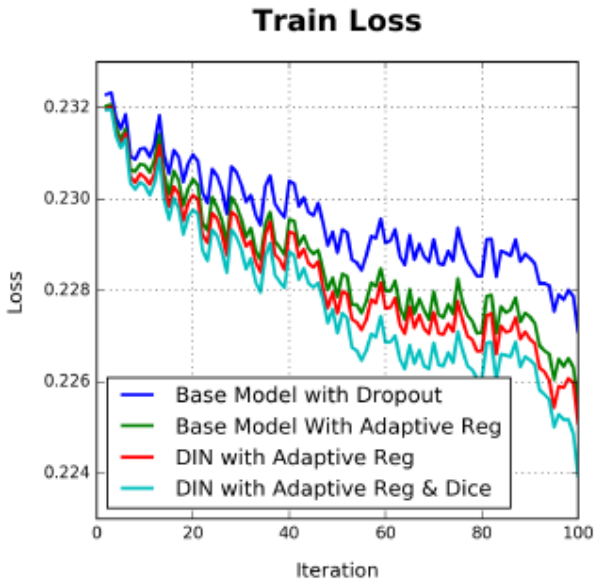


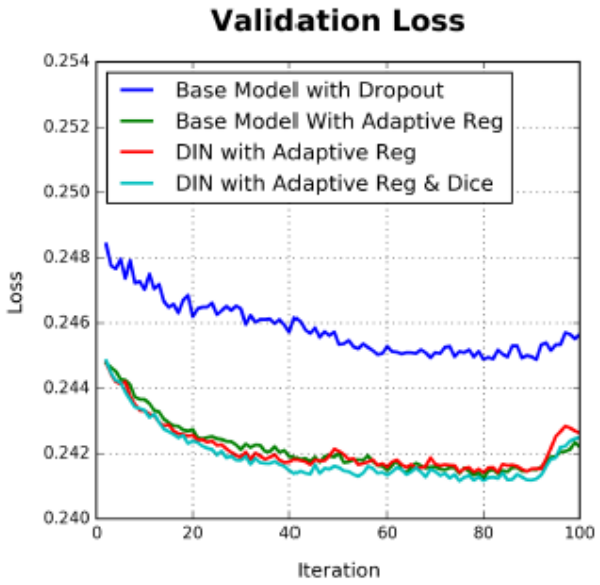


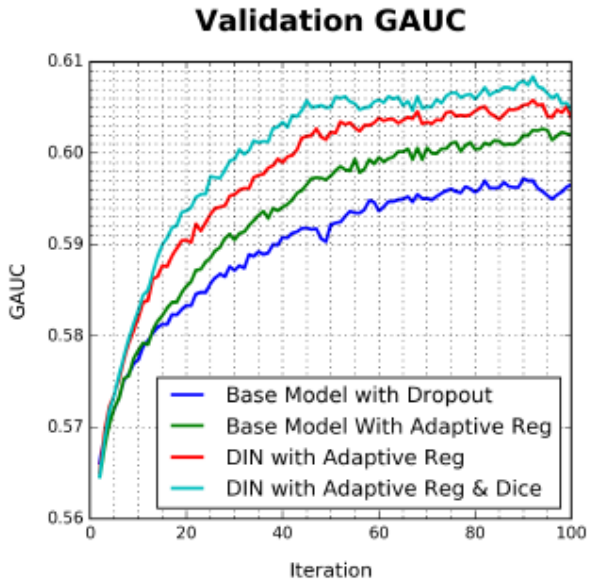
业务数据集上效果

Table 3: Comparison of model performance.

	GAUC	GAUC gain on Base
Base Model	59.59%	0.0%
Base Model with Drop out	59.70%	0.11%
Base Model with adaptive_reg	60.31%	0.72%
DIN Model with adaptive_reg	60.60%	1.01%
DIN Model with adaptive_reg and Dice	60.83%	1.24%







AI@Alibaba

- AI是阿里巴巴重要的技术方向
  - 既注重现有业务上的实用性，也注重长期对未来的储备
- 我们的团队
  - 机器学习模型算法
  - 机器学习平台
  - 视觉图像　　OCR刷新ICDAR BornDigital数据集最好成绩
  - NLP
  - 广告机制和策略　　OCPC算法 @KDD 2017
  - 客户端优化
  - 在线引擎和工程架构
- 如果你对技术有强烈热情，渴望突破和改变，欢迎加入我们！
  邮箱：jingshi.gk@taobao.com
  微博：heavenfireray
- 本文主要资料：
  - Learning Piece-wise Linear Models from Large Scale Data for Ad Click Prediction. https://arxiv.org/abs/1704.05194
  - Deep Interest Network for Click-Through Rate Prediction. https://arxiv.org/abs/1706.06978

感谢聆听

阿里妈妈愿与您一起成长！

阿里妈妈
Alimama.com

Alibaba Group
阿里巴巴集团