

# 搜广推之特征工程：基本原理和前沿研究

Dr. Bin Fu

[binfu@pku.edu.cn](mailto:binfu@pku.edu.cn)

2023-09

# 特征工程

- 数据和特征决定了效果的上限，算法和模型只是逼近这个上限的手段。
- 根据场景业务特点设计合理的特征和模型。
- AutoFE处于初级探索阶段，不够有效，依然考验经验直觉和业务知识。

# 特征工程：目录

- 特征构建
- 特征预处理
- 类别特征处理
- 稠密特征处理
- 特征交叉
- 特征选择
- 特征服务架构
- 公开数据集
- 未来探索方向

# 特征构建

## • 用户侧：（用户画像）

- 基本属性：id、人口属性（如性别、年龄、学历、职业、位置等）和注册信息（手机品牌、注册时间等）、兴趣爱好、购买力、婚育、薪资、颜值。
- 社交特征：好友、点赞、关注等。强关系和弱关系。U2U兴趣人群、同小区等。
- 行为特征：各种行为历史，如曝光、点击、播放、点赞、反对等。显式反馈和隐式反馈。
  - 不同粒度时间窗口：最近、过去1小时、过去1天、过去1周、过去1月、至今，考虑时间衰减。热度时效性等。
  - 正向/负向：转发/点赞/踩/跳过等。
  - 统计：次数/时长/金额/比率/单位价格/活跃情况。
  - 序列特征。

## • 物品侧：（物品画像）

- 基本属性：品牌、id、类目、标题、价格、产地、适用人群、评分、销量、商家信息、商圈等。
- 内容特征：基于内容理解技术打上多级分类标签或关键词topic等。知识图谱等。
  - 文本：评论、签名等。通过Ngram/TFIDF/LDA/word2vec/fasttext等挖掘。
  - 图像：通过CNN将图片解析成向量。
- 反馈信息：
  - 点击量、点击率、购买量、CTR、CVR等；月比趋势等。

## • 上下文特征：

- 地理位置（经纬度、城市、距离、IP等）、天气、社会事件、手机品牌、操作系统。
- 时间：季节、工作日、休息日、发薪日、早中晚等。
- 推荐场景特征：APP、浏览器主Feed推荐、相似推荐、当前刷次、翻页动作等；场景平均点击率转化率。搜索词query。
- 网络类型：wifi、4G、5G等。

# 特征预处理

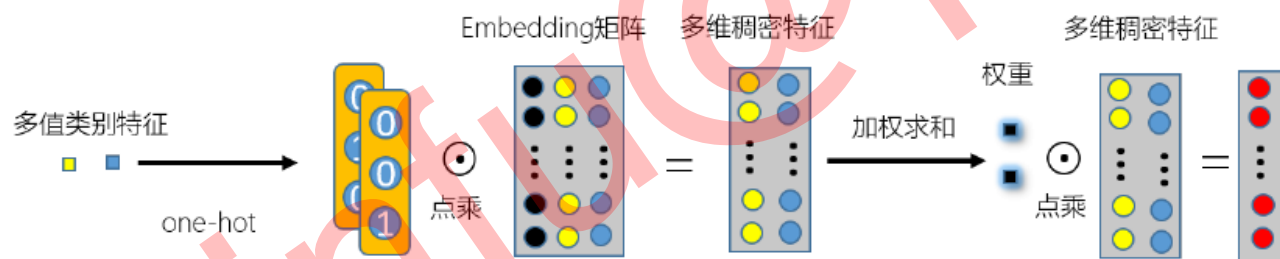
- 特征缺失值处理：
  - 固定值填充：均值/中位数/众数等。
  - 模型预测值填充：xgboost可处理缺失值。
- 统计量特征数据平滑：
  - 贝叶斯平滑：实验多次，随机事件接近其真实概率分布。如利用beta分布 $B(\alpha, \beta)$ 建模点击率先验。

$$Rate = \frac{C + \alpha}{I + \alpha + \beta}$$

- 威尔逊平滑：样本多可信，样本少不可信需要修正。
- 消偏：冷热门类目、长短视频等。相当于提权降权作用。

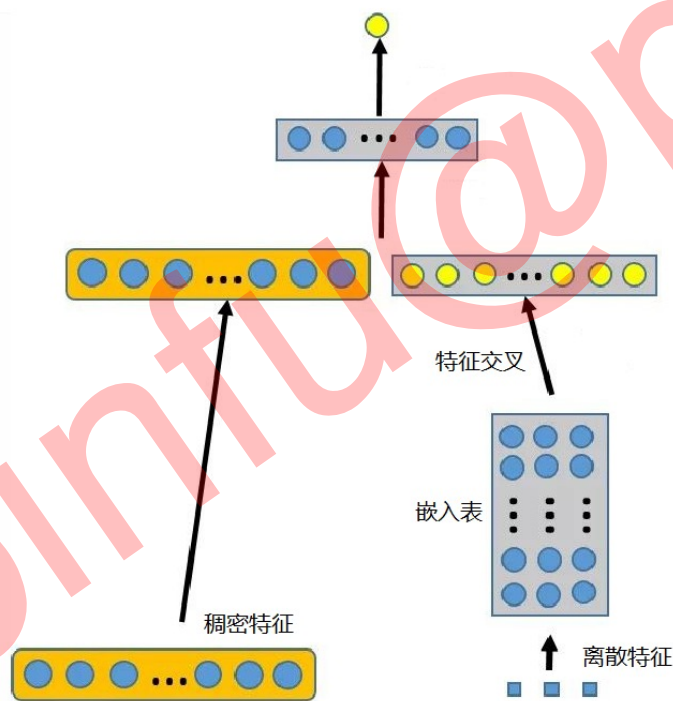
# 类别特征处理

- 类型：
  - 单值类型
  - 多值类型
    - 平均池化、最大池化、最小池化、加权池化（注意力融合，类似FiBiNET）



# 稠密特征处理

- 1. 不做处理：
  - 典型案例：Wide&Deep中作为Wide部分处理。
  - 需要归一化/标准化/非线性变换（log/sqrt/square）/缩放等。
  - 缺点：表达能力弱，无法实现与离散特征之间的交叉，数值敏感缺乏鲁棒性。



# 稠密特征处理

## • 2.先离散化再进行嵌入表征学习： $f(Dis(R^1)) \rightarrow R^{1 \times d}$

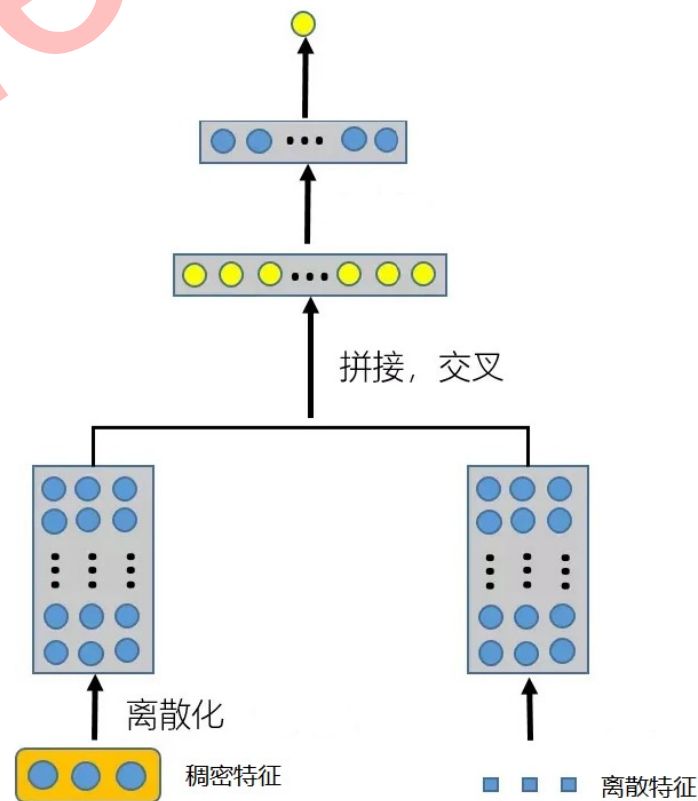
### • 硬离散化（hard）：

- 无监督分桶：等距、等频和log离散化 $\text{floor}(\log(x))$ 。
- 有监督分桶：树模型，如xgboost。

### • 优点是方便后续进行特征交叉。

### • 缺点：

- 两阶段误差扩大；
- 划分点附近相似特征被划分开；
- 同一桶内左右边界点差异大。

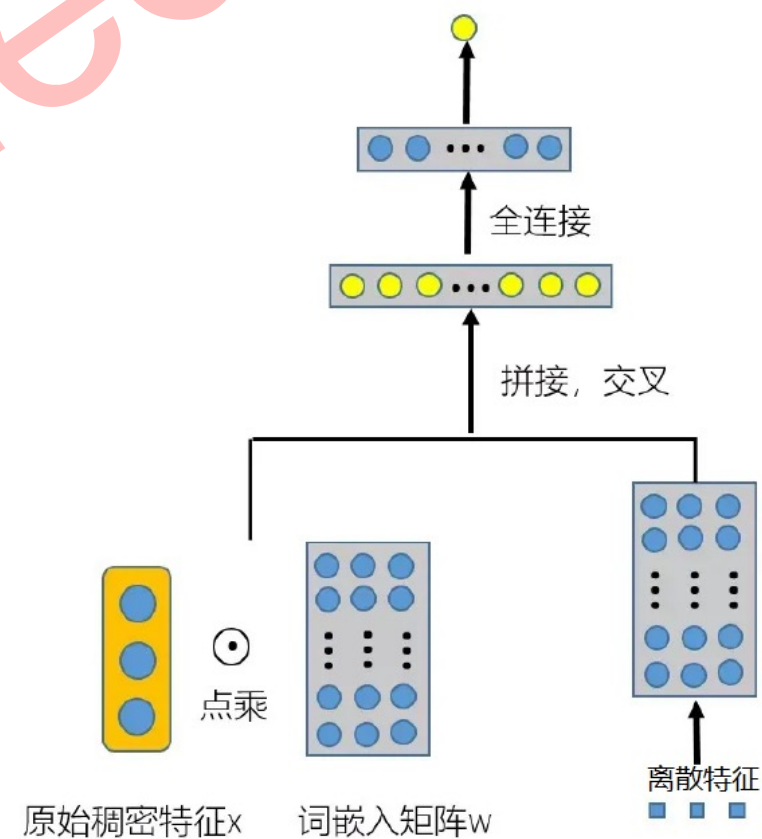
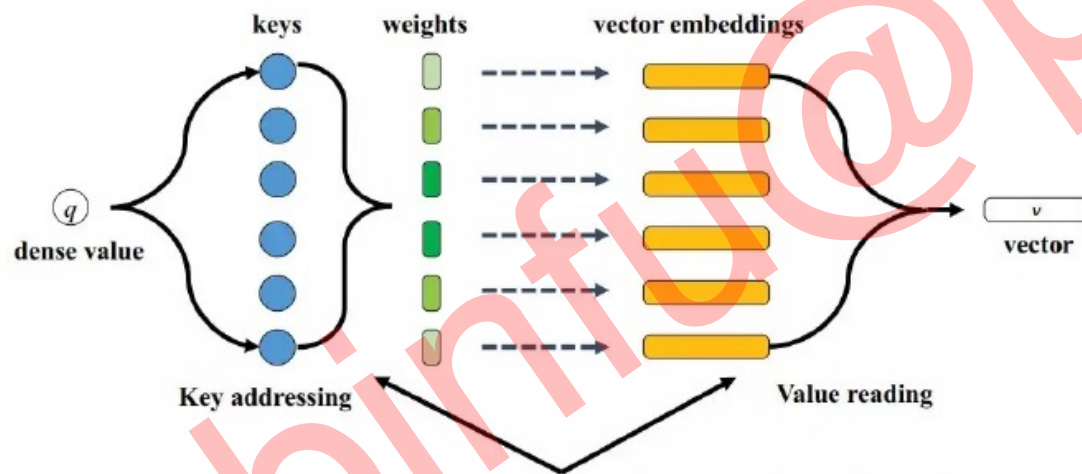




# 稠密特征处理

- 3. 端到端的离散化表征学习: (soft)
  - Key-Value Memory方法: 利用记忆网络实现一维到二维转换

$$v = \sum_{i=0}^{N-1} w_i v_i, w_i = \text{softmax}\left(\frac{1}{|q - k_i + \epsilon|}\right), k_i = \frac{2i+1}{2N}$$



# 稠密特征处理

## • 3.端到端的离散化表征学习: (soft)

- AutoDis: 注意力  $\hat{x}_j$  + 元嵌入  $ME_j$

$$\hat{x}_j = W_j h_j + \alpha h_j, h_j = \text{Leaky\_ReLU}(w_j x_j)$$

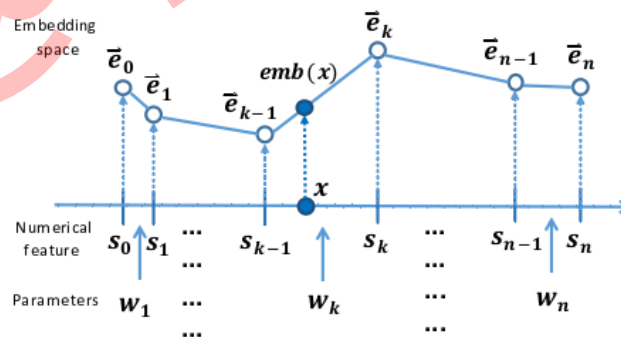
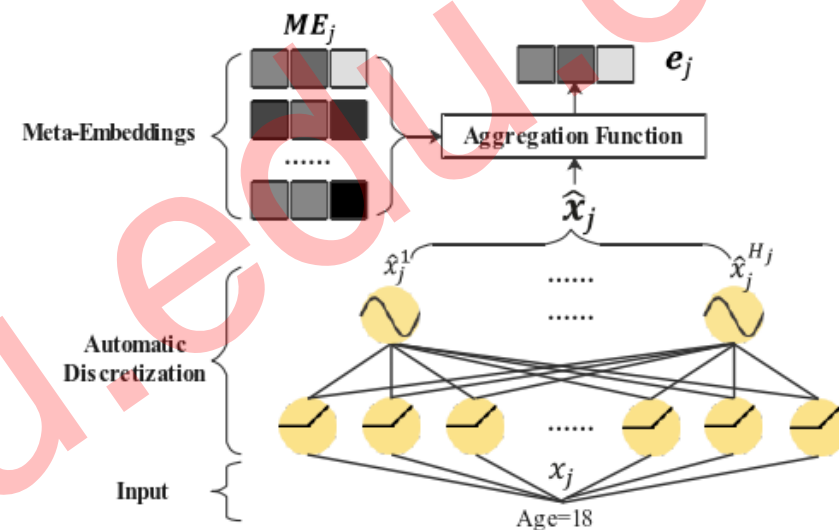
$$e_j = \text{softmax}(\hat{x}_j) \cdot ME_j, ME_j = R^{H_j \times d}$$

- 超参数: 桶数  $H_j$  和维度  $d$ 。

- DEER: 中值平滑代替注意力, 假设  $s_i - s_{i-1} \propto \exp(w_i)$ 。

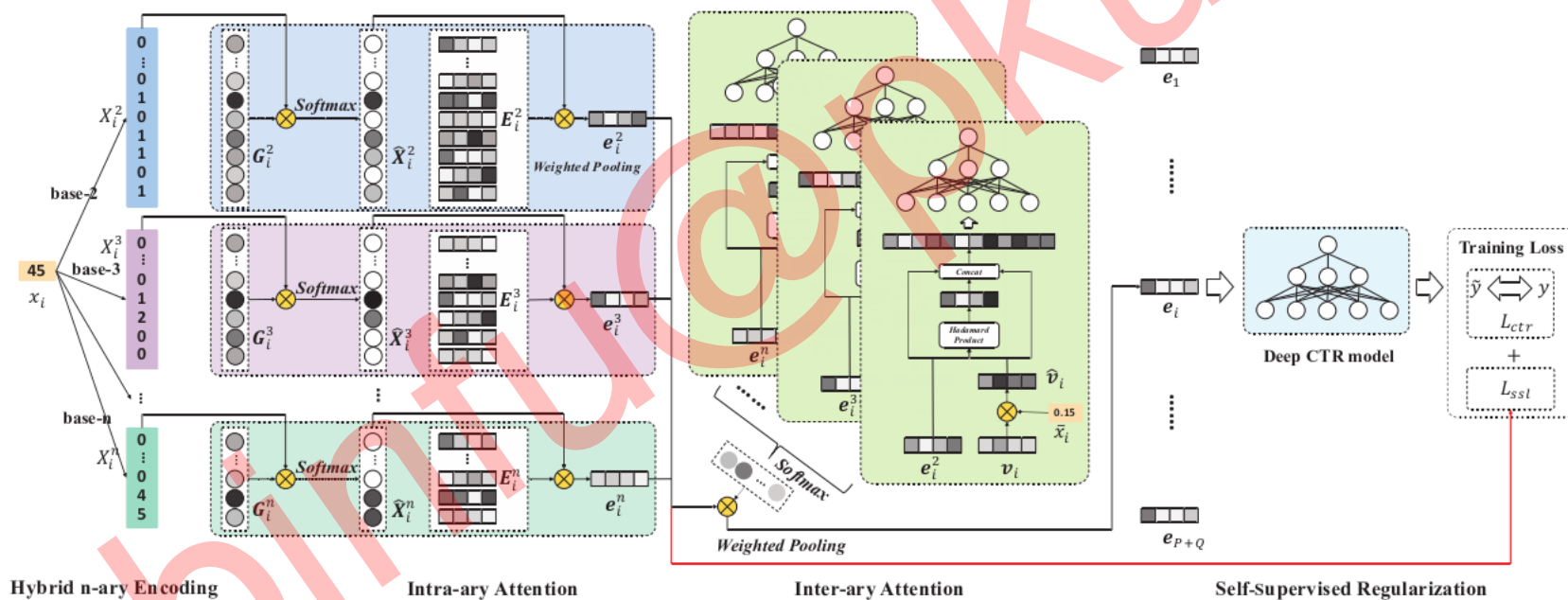
$$s_i = s_0 + \sum_{k=1}^i \frac{\exp(w_k)}{\sum_{l=1}^n \exp(w_l)} (s_n - s_0)$$

$$f(x) = \frac{s_i - s_{k-1}}{s_k - s_{k-1}} \vec{e}_{s_{k-1}} + \frac{s_k - s_i}{s_k - s_{k-1}} \vec{e}_{s_k}$$



# 稠密特征处理

- 3.端到端的离散化表征学习：（soft）
  - NaryDis: 自监督对比学习正则项。相近的相似（连续性），远离的不相似（判别性）。
    - 超参数：正则项系数 $\alpha \in [0.5, 0.9]$ 和编码空间大小 $N \in [1, 4]$ 。



# 特征交叉

- 为什么需要特征交叉？

- 特征之间存在关联模式，DNN通过隐式方式难以学习到。如年龄、性别和兴趣偏好。

- 设计角度：

- 二阶和高阶。
  - 显式和隐式：手工经验设计或自动交叉。

$$f_{ij}^{ex} = \vec{e}_i \odot \vec{e}_j, \quad f_{ij}^{im} = NN([\vec{e}_i; \vec{e}_j])$$

- 线性和非线性。
  - Bit-wise和field-wise。

# 特征交叉

- 交叉类型:

- 内积 (Inner Product) : 如 FM、FFM、AFM、DeepFM等。
- 外积 (Outer Product) : 如PNN等。
- 哈马达乘积 (Hadamard Product)
- 双线性交叉 (Bilinear Interaction)
- 注意力机制: AutoInt.

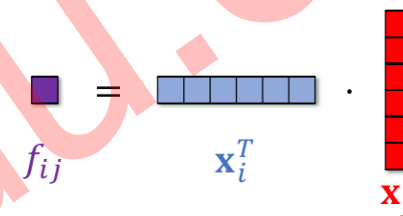


Diagram illustrating the Inner Product (dot product) operation. A single purple square representing the output  $f_{ij}$  is equal to a horizontal blue vector representing  $\mathbf{x}_i^T$  multiplied by a vertical red vector representing  $\mathbf{x}_j$ .

$$f_{ij} = \mathbf{x}_i^T \cdot \mathbf{x}_j$$

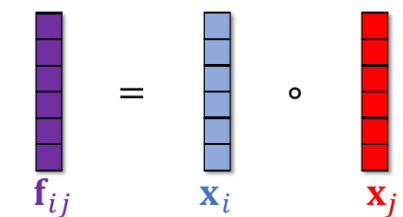


Diagram illustrating the Outer Product operation. A vertical purple vector representing  $\mathbf{f}_{ij}$  is equal to a vertical blue vector representing  $\mathbf{x}_i$  multiplied by a vertical red vector representing  $\mathbf{x}_j$ .

$$\mathbf{f}_{ij} = \mathbf{x}_i \circ \mathbf{x}_j$$

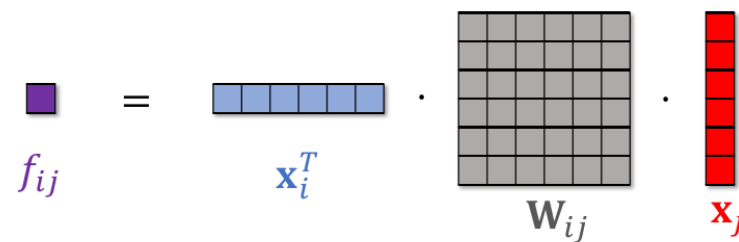


Diagram illustrating the Bilinear Interaction operation. A single purple square representing the output  $f_{ij}$  is equal to a horizontal blue vector representing  $\mathbf{x}_i^T$  multiplied by a gray grid representing the weight matrix  $\mathbf{W}_{ij}$ , which is then multiplied by a vertical red vector representing  $\mathbf{x}_j$ .

$$f_{ij} = \mathbf{x}_i^T \cdot \mathbf{W}_{ij} \cdot \mathbf{x}_j$$

# 特征交叉

- 自动特征交叉：组合优化问题。
  - AutoFeature：利用神经架构搜索（NAS）技术搜索合适的特征交叉。
    - 特征交叉视为一个子神经网络，利用朴素贝叶斯来学习这些网络有效或无效。
    - 平衡探索和利用。

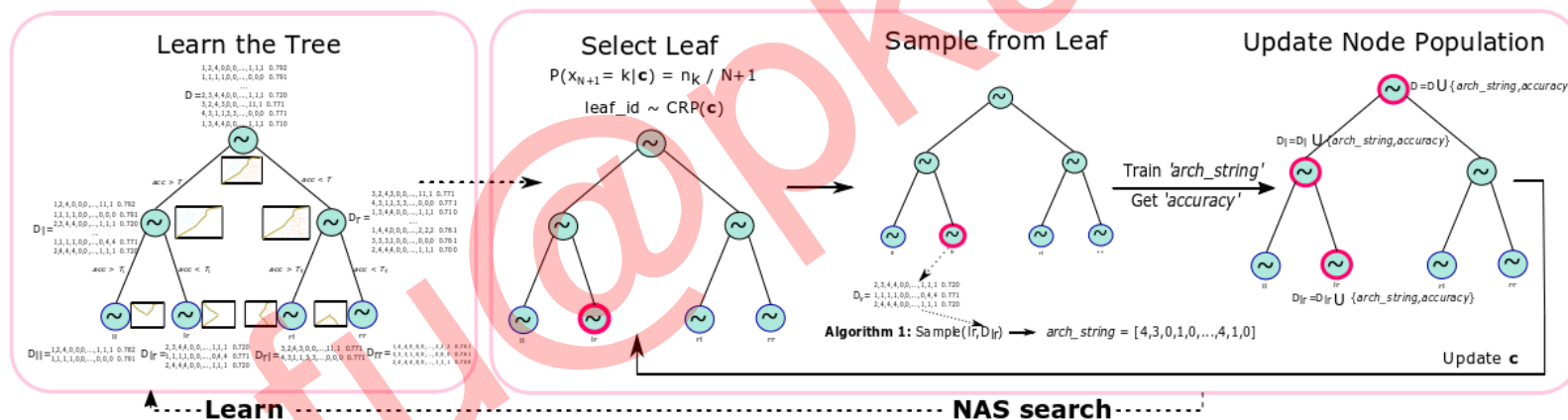


Figure 3: The overview of our proposed neural architecture search algorithm. AutoFeature includes two iterative search phases: 1) Learning the partitions of the tree; 2) Sampling and training. During phase one, we learn the subspaces based on the current architectures at each node. During phase two, we select a leaf according to CRP and randomly sample an architecture from this node. This architecture is fully trained and the result is used to update the population of each path node.

AutoFeature: Searching for Feature Interactions and Their Architectures for Click-through Rate Prediction, CIKM 2020.

Towards Automated Neural Interaction Discovery for Click-Through Rate Prediction, KDD 2020.

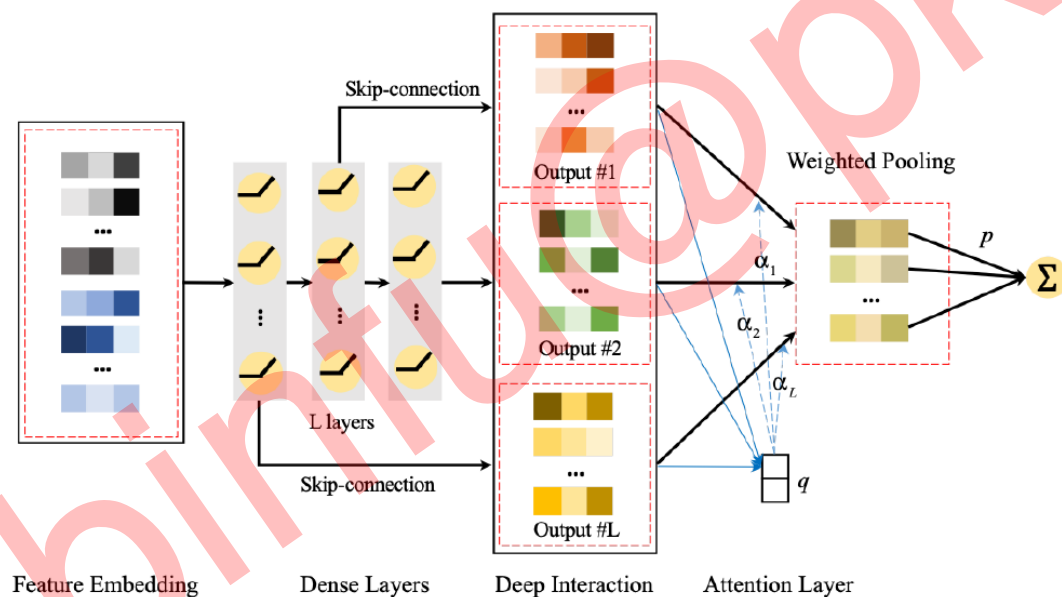
AutoGroup: Automatic Feature Grouping for Modelling Explicit High-Order Feature Interactions in CTR Prediction, SIGIR 2020.

Cognitive Evolutionary Search to Select Feature Interactions for Click-Through Rate Prediction, KDD 2023. (进化算法)

# 特征交叉

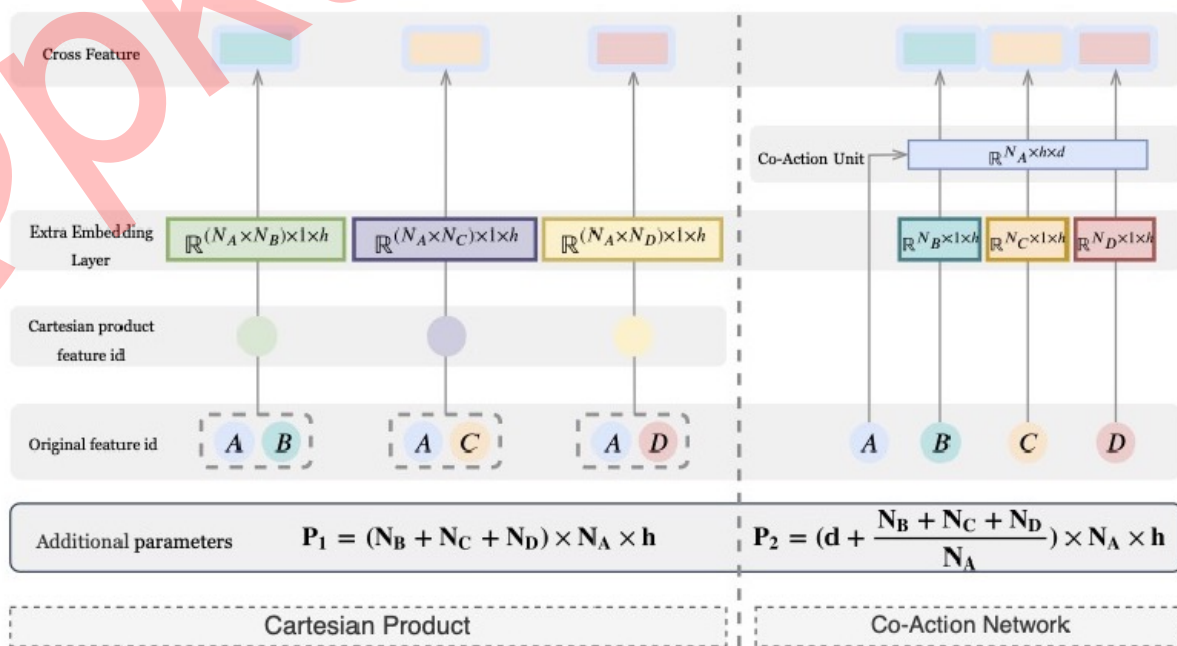
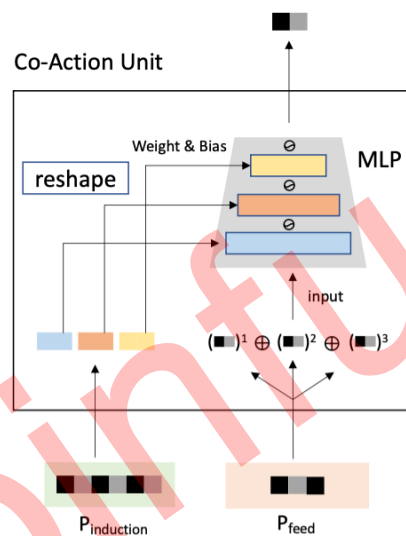
- 高阶交叉

- 典型案例：DCN、xDeepFM等。
- AdnFM深度多层 ( $\leq 3$ ) 交互：
  - 加权池化输入特征。
  - 利用残差连接和注意力获取每层的交互特征，类似DenseNet。



# 特征交叉

- 交叉形式
  - CAN (Co-Action Network) 交叉：
    - Target Item和用户行为序列进行多层次交叉。
    - 效果很好。





# 特征交叉

## 交叉形式

- EDCN层次交叉：多层反复交叉。
- FINAL利用深层网络实现高阶交叉。

$$\mathbf{h}_{l,1} = \mathbf{W}_{l,1}\mathbf{x}_{l-1} + \mathbf{b}_{l,1},$$

$$\mathbf{h}_{l,2} = \mathbf{h}_{l,1} \odot \sigma(\mathbf{W}_{l,2}\mathbf{x}_{l-1} + \mathbf{b}_{l,2}),$$

...

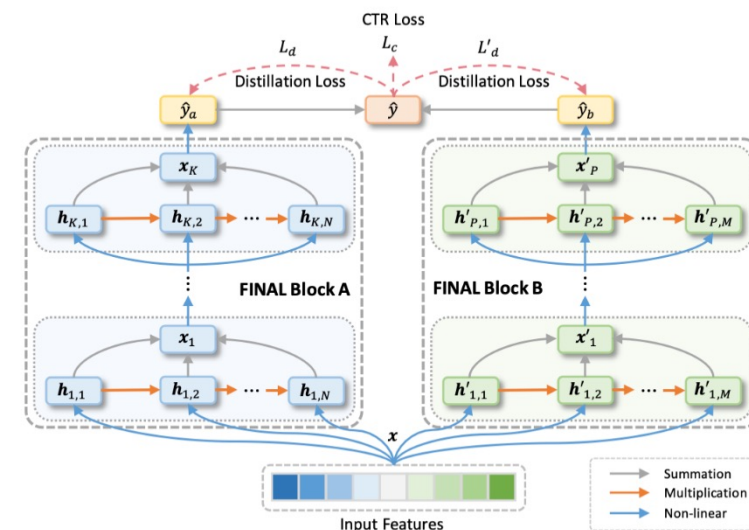
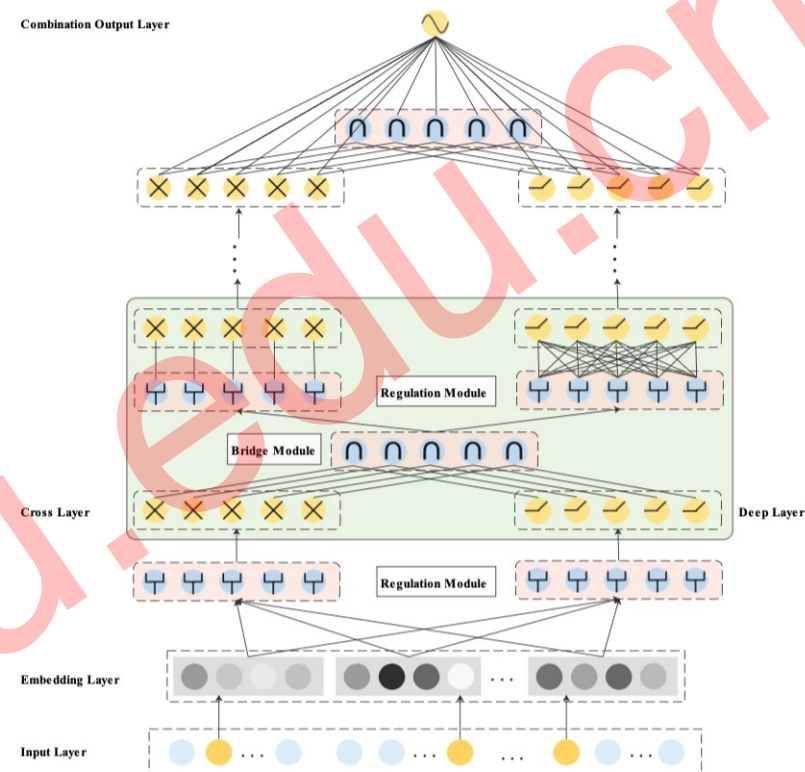
$$\mathbf{h}_{l,N} = \mathbf{h}_{l,N-1} \odot \sigma(\mathbf{W}_{l,N}\mathbf{x}_{l-1} + \mathbf{b}_{l,N}),$$

$$\mathbf{x}_l = \sum_{i=1}^N (\mathbf{h}_{l,i}),$$

$$\mathcal{L}_c = -\frac{1}{S} \sum_{i=1}^S [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)],$$

$$\mathcal{L}_d = -\frac{1}{S} \sum_{i=1}^S [\hat{y}_i \log(\hat{y}_{a,i}) + (1 - \hat{y}_i) \log(1 - \hat{y}_{a,i})],$$

$$\mathcal{L}'_d = -\frac{1}{S} \sum_{i=1}^S [\hat{y}_i \log(\hat{y}_{b,i}) + (1 - \hat{y}_i) \log(1 - \hat{y}_{b,i})],$$



# 特征选择

- 意义：

- 最大化相关、最小化冗余。
- 降低复杂性，避免过拟合，简化模型，提高泛化能力。
- 节省存储和计算时延。

- 方法：

- 过滤方法：

- 无监督：方差、覆盖率
- 有监督：互信息（类别类型之间）、皮尔逊相关系数（连续类型之间）、单特征AUC等

- 正则法：L2和L1等。

- 封装法：有监督模型。

- 深度模型方法

# 特征选择

- 深度模型方法：
  - AdaFS: 特征权重（注意力）的top-k个（超参数）。
  - 结果显示选择50%的特征其效果有竞争力。

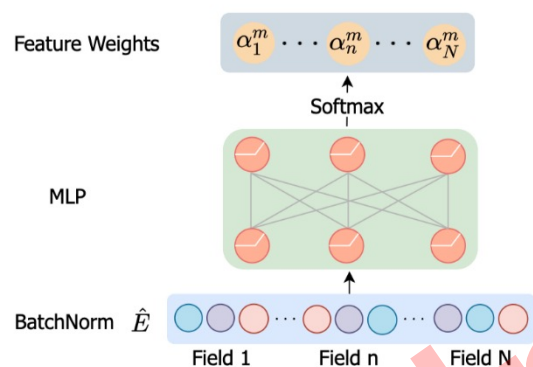
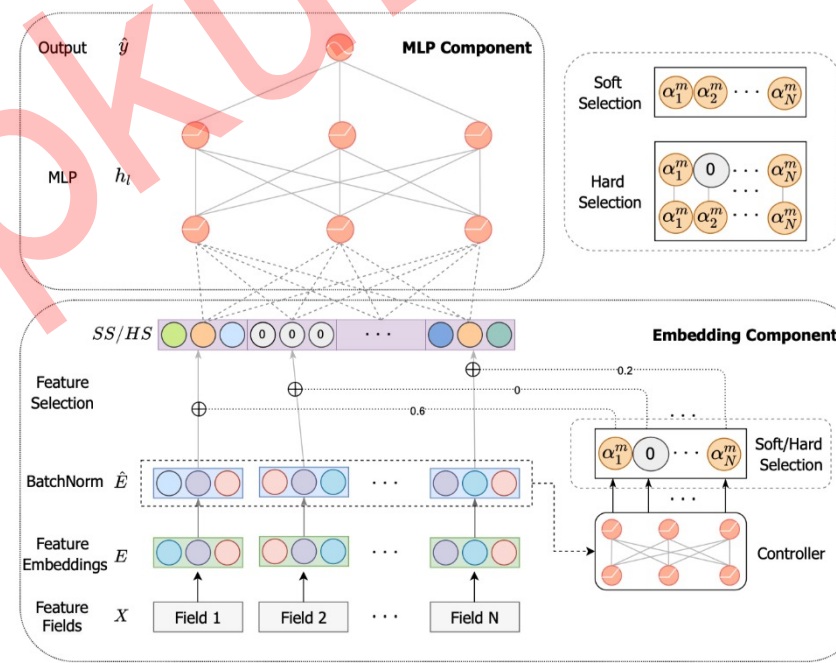


Figure 3: The process of hard selection.



# 特征选择

- 深度模型方法:

- LPFS平滑 $l^0$ 门控函数: 训练过程中不断衰减 $\epsilon$ , 使其自适应学习, 激活概率分布更加集中。

$$g_{\epsilon}(x) = \frac{x^2}{x^2 + \epsilon} = \begin{cases} = 0, & x = 0 \\ \approx 1, & x \neq 0 \end{cases}$$

$\epsilon$ 是一个极小正数。

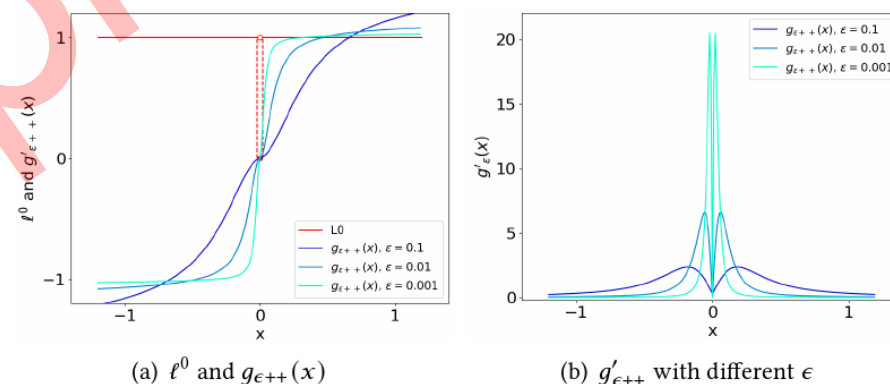
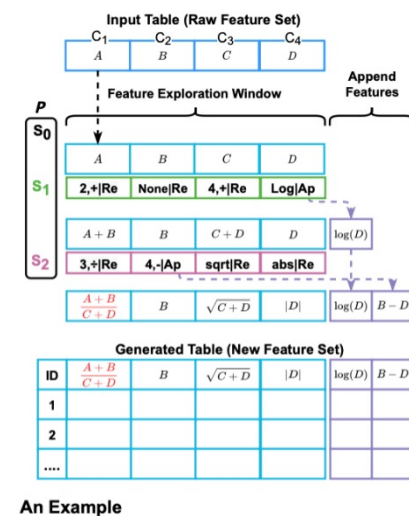
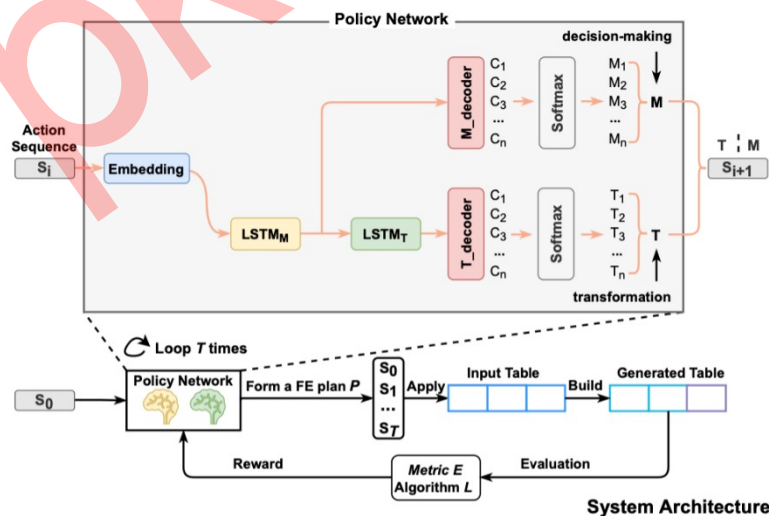
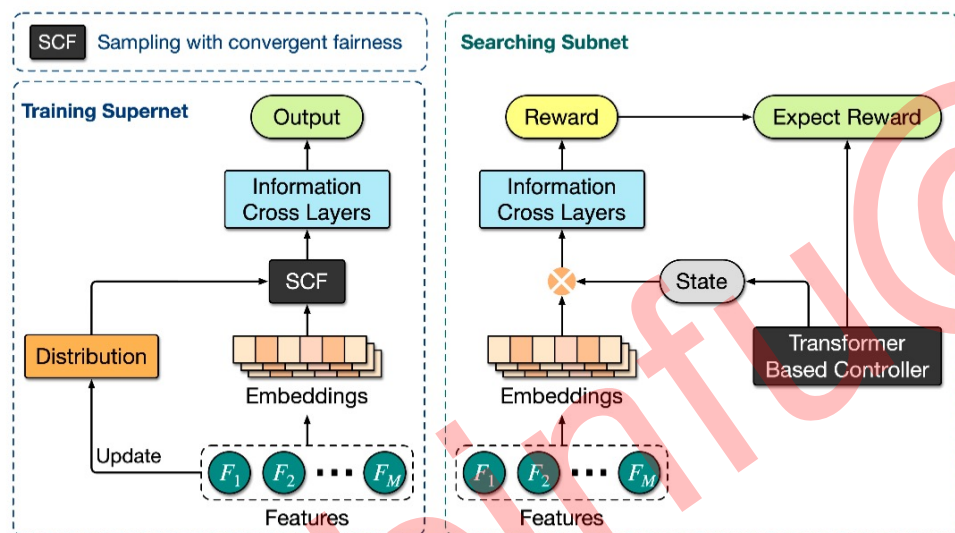


Figure 4: The graph of  $l^0$  and  $g_{\epsilon++}(x)$  with different  $\epsilon$  (left) and  $g'_{\epsilon++}$  (right). Now  $g_{\epsilon++}(x)$  is odd, not even like the smoothed  $l^0$  function. Note that the gradient at  $x = 0$  is no longer zero, but a small number decays with  $\epsilon$ .

# 特征选择

## 深度模型方法:

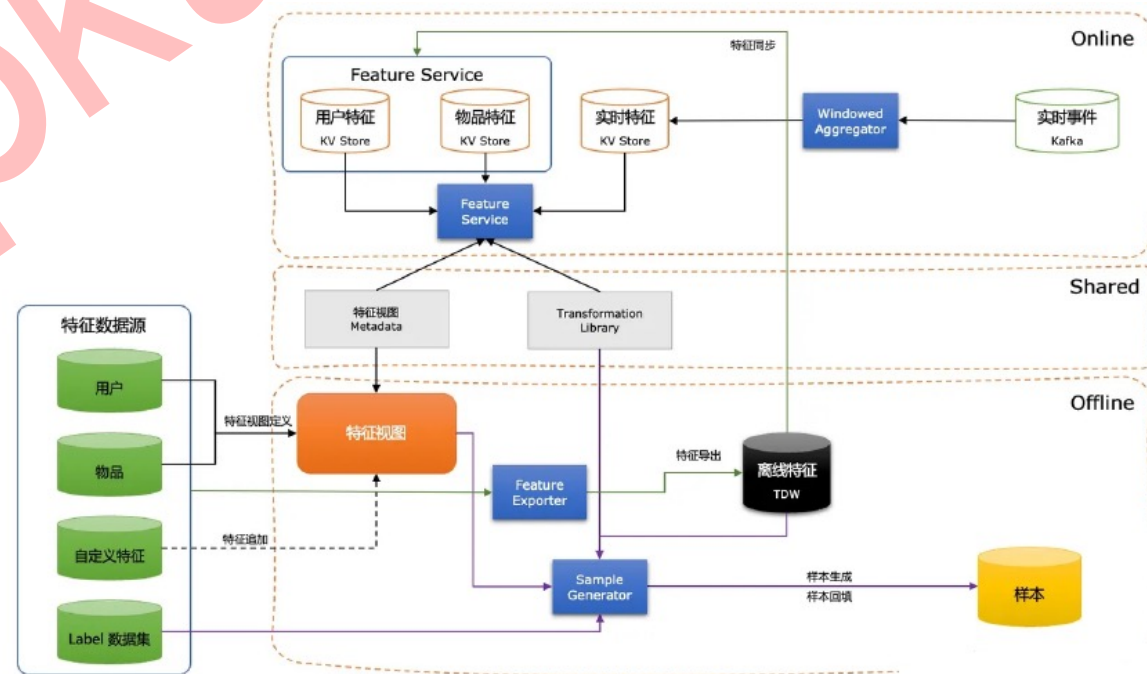
- AutoFSS: 采用神经架构搜索 (one-shot NAS) 技术 (强化学习) 搜索合适的特征子集。
- Catch: 采用强化学习来搜索合适的特征子集。





# 特征服务架构

- 离线：
  - 全量生成离线特征并存储，并用于模型训练。
  - 如采用Hive构建简单特征、Spark构建复杂逻辑的特征。
  - 模型训练框架采用Tensorflow/Pytorch等。
- 实时：
  - 根据请求（时间窗口）实时生成特征。
  - Redis线上存储KV特征。
  - 使用消息队列收集数据，如kafka/Flink。
- 离线在线特征一致性：
  - 避免数据穿越。
  - 尽量使用同一套处理逻辑。
  - 在线埋点存特征到日志处理后喂给离线模型。



# 公开数据集

- 淘宝用户购物行为数据集: <https://tianchi.aliyun.com/dataset/649>
  - 用户ID; 商品ID; 商品类目ID; 行为类型包括('pv', 'buy', 'cart', 'fav'); 时间戳
- Avazu: <https://www.kaggle.com/competitions/avazu-ctr-prediction/data>
  - id: ad identifier; click: 0/1 for non-click/click; hour: format is YYMMDDHH, so 14091123 means 23:00 on Sept. 11, 2014 UTC; C1 -- anonymized categorical variable; banner\_pos; site\_id; site\_domain; site\_category; app\_id; app\_domain; app\_category; device\_id; device\_ip; device\_model; device\_type; device\_conn\_type; C14-C21 - anonymized categorical variables
- Criteo: <http://labs.criteo.com/2014/02/kaggle-display-advertising-challenge-dataset/>
- 其他数据集:
  - <https://www.baltrunas.info/context-aware>

# 未来探索方向

- 新的特征交叉算子
- 自动搜索合适的特征交叉
- 和其他问题结合，如多任务多场景
- 与语言大模型（LLM）结合，提高特征表达能力
- 模型可解释性（如ID嵌入特征）

.....