GLIDER - 发现重要交叉特征

Global Interaction Detection and Encoding for Recommendation(GLIDER) 是一种发现神经网络学习到的任意阶的**交叉特征**的方法。主要包含以下两篇论文：

1. Detecting Statistical Interactions from Neural Network Weights (ICLR'18)
2. Feature Interaction Interpretability: A Case for Explaining Ad-Recommendation Systems via Neural Interaction Detection (ICLR'19)

第一篇文章提出一种**交叉特征检测的方法 (NID)**，主要是发现 MLP 学习到的比较重要的交叉特征。第二篇文章把 NID 方法用到推荐模型上，去发现推荐模型学习到的交叉特征。之后，再把发现的重要后验交叉特征加到原始的模型上，然后重新训练模型提升模型效果。

1. Neural Interaction Detection (NID) -- 发现MLP的重要交叉特征

如果把整个网络看成一个有向无环图，输入层每个特征和中间隐层的神经元看成图的节点，连接节点之间的权重看成边。对任何一个交叉特征集合 $\mathcal{I}$ ，都一定存在一个节点 $V_{\mathcal{I}}$ 是他们共同的子孙。基于这个想法，第一篇文章把MLP第一个隐层的所有节点看作是我们需要找的“共同子孙” $V_{\mathcal{I}}$ 。那么在第一层的第 i 个神经元上，交叉特征 $\mathcal{I}$ 的强度(interaction strength)记为$\omega_i(\mathcal{I})$。整个模型交叉特征 $\mathcal{I}$ 的组合强度则是把 $\omega_i(\mathcal{I})$ 累加起来，记为 $\omega(\mathcal{I})$

$$\omega(\mathcal{I}) = \sum_{i=1}^{p_1} \omega_i(\mathcal{I})$$

$$\omega_i(\mathcal{I}) = z_i^{(1)} \cdot \mu(|\mathbf{w}_{i,\mathcal{I}}^{(1)}|)$$

从上面式子可以看出，交叉特征的强度是神经元 i 前面部分($\mu(|\mathbf{w}_{i,\mathcal{I}}^{(1)}|)$)和后面部分($z_i^{(1)}$)的乘积，

$\mu(|\mathbf{w}_{i,\mathcal{I}}^{(1)}|)$是交叉特征 $\mathcal{I}$ 与神经元 i 连接的权重$\mathbf{w}_{\mathcal{I}}$ 的某种均值函数（实际取的是min）,后面部分$z_i^{(1)}$ 是神经元 i 对最终预测 y 的影响，或者说是神经元 i 的重要度（参考图1 示意图）。

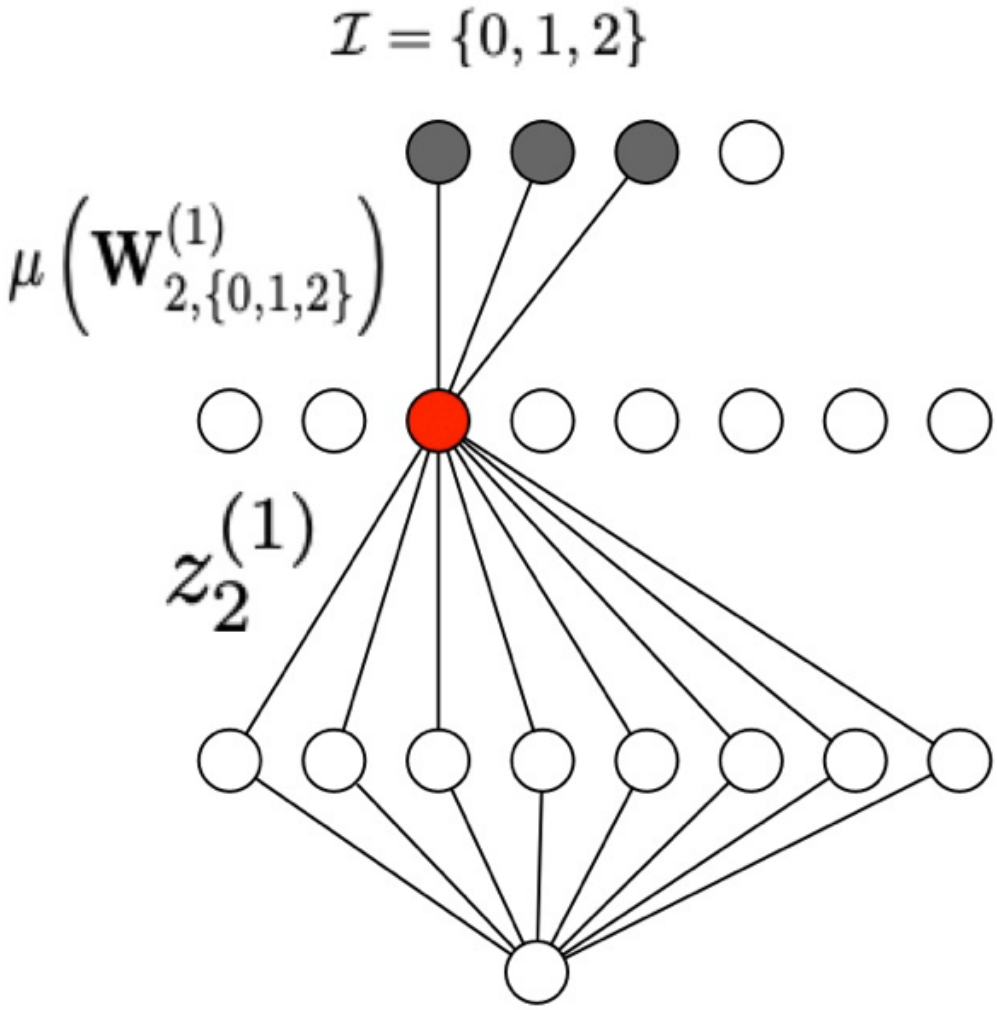


图1

第一篇文章用了**权重矩阵连乘**来作为神经元 i 的重要度 $z_i^{(1)}$，它是模型输出 y 对神经元 i 输出梯度绝对值的上界。这个不等式的证明参考原文的附录C. 为啥要证明它是梯度绝对值的上界，是因为梯度绝对值是一种常见的重要度度量方案，这里NID**用权重矩阵连乘来近似。**

$$z^{(1)} = |\mathbf{w}^y|^T \cdot \prod_{l=L}^{2} |\mathbf{w}^{(l)}|$$

$$z_i^{(1)} \geq \left|\frac{\partial y}{\partial h_i^{(1)}}\right|$$

Greedy Ranking

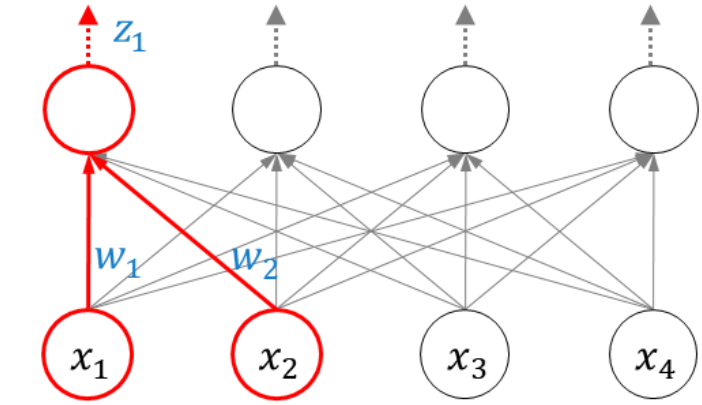
greedy ranking，对每个神经元不进行全量的$2^p - 2$个特征组合遍历，而是每次取 top n 的特征组合，那么特征组合的数量由$O(2^p) \rightarrow O(p)$，这个过程可以参考图3的动图。

Algorithm 1 NID Greedy Ranking Algorithm

Input: input-to-first hidden layer weights $\mathbf{W}^{(1)}$, aggregated weights $\mathbf{z}^{(1)}$
Output: ranked list of interaction candidates $\{\mathcal{I}_i\}_{i=1}^{m}$

$d \leftarrow$ initialize an empty dictionary mapping interaction candidate to interaction strength
for each row $\mathbf{w}'$ of $\mathbf{W}^{(1)}$ indexed by r **do**
    for $j = 2$ to p **do**
        $\mathcal{I} \leftarrow$ sorted indices of top j weights in $\mathbf{w}'$
        $d[\mathcal{I}] \leftarrow d[\mathcal{I}] + z_r^{(1)} \mu\left(|\mathbf{w}'_{\mathcal{I}}|\right)$
$\{\mathcal{I}_i\}_{i=1}^{m} \leftarrow$ interaction candidates in d sorted by their strengths in descending order

注意中间是两层循环，每次都去只更新前top p个重要交叉特征的强度：



Interactions	Strengths
{1,2}	$z_1 \min(\|w_1\|, \|w_2\|)$

$$|w_1| > |w_2| > |w_3| > |w_4|$$

2. GLIDER

NID只能发现 **MLP** 的交叉特征，那么我们的推荐模型如果不是MLP怎么样呢？第二篇文章结合 **LIME** 方法学习一个**局部**代理模型(MLP)，然后再使用 NID 去发现这个代理模型的交叉特征。

这里简单说下 LIME 扰动数据的思路：给一个样本$\boldsymbol{x}$，我们可以随机改变它的某一维特征值，对于实值类型则置为默认值（例如0），这样就能得到一个新样本$\boldsymbol{x'}$，重复 n 次就能根据一个样本生成 n 个样本。这些样本都是分布在原始样本的附近，那么可以用分类模型 $\boldsymbol{f_{rec}}$（MLP）对这些样本进行预测，这样就能构成一个新的数据集

$$\mathcal{D}_p = \{< x^{'}, y^{'} = f_{rec}(x^{'}) > | x^{'} \in \text{perturbate}(x)\}$$

然后在这个新的数据集上的训练一个MLP，并用NID去检测这个 MLP 的交叉特征。到目前为止得到的交叉特征可以看作是 $\boldsymbol{f_{rec}}$ 的**局部代理模型**（毕竟 MLP 是在 x 附近的点上训练得到的）。第二篇文章中提出针对推荐模型的全局特征检测方法：随机选取$\boldsymbol{N}$个点，重复上面用代理模型做交叉特征检测的操作，然后把这 N 次结果得到的特征组合累计。这就是“Global”的由来。

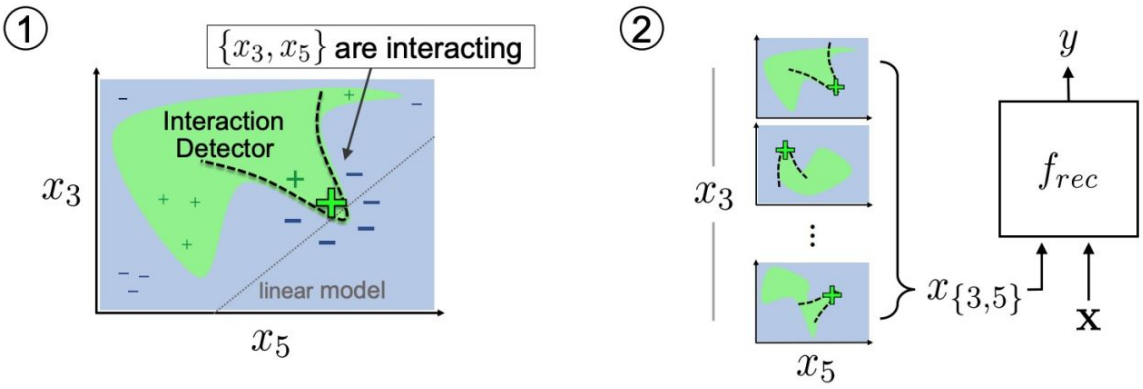


Figure 1: A simplified overview of GLIDER. ① GLIDER utilizes Neural Interaction Detection and LIME together to interpret feature interactions learned by a source black-box model at a data instance, denoted by the large green plus sign. ② GLIDER identifies interactions that consistently appear over multiple data samples, then explicitly encodes these interactions in a target black-box recommender model f_{rec}.

最后，把发现的重要交叉特征再次加入模型中。

2. Ante-hoc可解释性

使用注意力机制，例如FiBiNET, InterHAt, AutoInt