[DCN]Deep & Cross Network for Ad Click Predictions

原创 飞翔的死胖子 高山仰止z 4月11日

特征工程是很多预测任务成功与否的关键，然而其过程耗时耗力。DNN可以自动地学习特征交互，但是很难学到所有的交互特征。本文提出Deep & Cross Network (以下简称DCN) 模型，在保留了DNN的优势的基础上加入了cross network，使其能够更高效地学习特征交互。DCN在每一层中都加入了特征交叉，无需人工特征工程。另外，相比DNN，DCN的复杂度提升微不足道。

本文的主要贡献包括：

- 我们提出了一种新的交叉网络，在每个层上明确地应用特征交叉，有效地学习有界度的预测交叉特征，并且不需要手工特征工程或穷举搜索。
- 跨网络简单而有效。通过设计，各层的多项式级数最高，并由层深度决定。网络由所有的交叉项组成，它们的系数各不相同。
- 跨网络内存高效，易于实现。
- 我们的实验结果表明，交叉网络（DCN）在LogLoss上与DNN相比少了近一个量级的参数量。

1.模型结构：

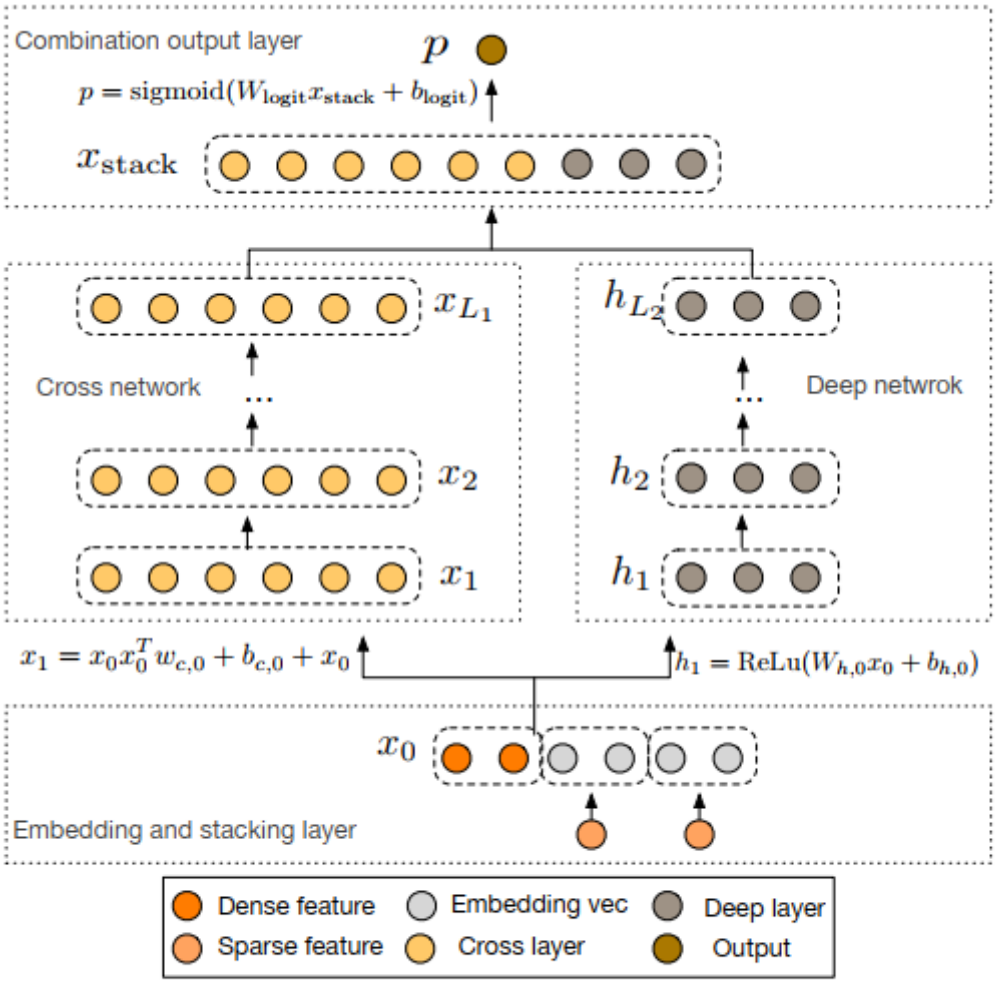

Figure 1: The Deep & Cross Network

一个DCN模型从嵌入层和堆积层开始，接着是一个cross network和一个与之平行的deep network，之后是最后的组合层，它结合了两个网络的输出。完整的网络模型如图1所示。

2.Embedding and Stacking Layer:

我们的输入数据有稀疏和稠密特征。在互联网推荐系统的CTR预估任务中，输入数据大多数为类别型特征，比如：国籍-中国，这些特征通常会被处理成onehot向量，这会导致特征的维度很高。为了减小维度，我们用embedding来把它们转化为稠密向量：

$$\mathbf{x}_{\text{embed},i} = W_{\text{embed},i}\mathbf{x}_i, \qquad (1)$$

Xembed,i为embedding向量，Xi为第i个类别型特征onehot后的输入向量

$$W_{\text{embed},i} \in \mathbb{R}^{n_e \times n_v}$$

W为embedding矩阵，会在网络中和其他参数一起优化。ne,nv分别为embedding尺寸和vocabulary 尺寸。

最后我们把所有embedding向量堆叠起来，形成一个向量：

$$\mathbf{x}_0 = \left[\mathbf{x}_{\text{embed},1}^T, \ldots, \mathbf{x}_{\text{embed},k}^T, \mathbf{x}_{\text{dense}}^T\right], \qquad (2)$$

再将X0输入神经网络。

3.cross network：

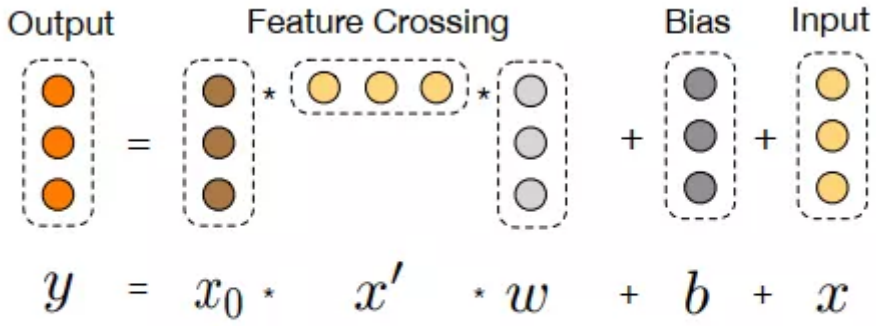

Figure 2: Visualization of a cross layer.

我们的cross network的核心是高效进行特征交叉。cross network由cross layer组成，每一层的公式如下：

$$\mathbf{x}_{l+1} = \mathbf{x}_0 \mathbf{x}_l^T \mathbf{w}_l + \mathbf{b}_l + \mathbf{x}_l = f(\mathbf{x}_l, \mathbf{w}_l, \mathbf{b}_l) + \mathbf{x}_l, \qquad (3)$$

其中，Xl和Xl+1分别为第l层和第l+1层的输出，Wl和bl分别为第l层的weight和bias。每层的输入和原始输入X0进行交互，最后还要加上本层的输入。特殊的cross network结构导致特征的交互随着层数增加而更深入。

复杂度分析：假设Lc表示交互层层数，d表示输入维度。则整个网络的参数数量为：

$$d \times L_c \times 2.$$

所以时间复杂度和空间复杂度依然是线性的。

cross network的参数数量很少，这限制了模型的性能， 因此我们还引入了一个deep network来捕获非线性特征交互。

4.Deep Network：

deep network是一个全连接前向传播神经网络，公式如下：

$$\mathbf{h}_{l+1} = f(W_l \mathbf{h}_l + \mathbf{b}_l), \qquad (4)$$

5.Combination Layer：

连接层先拼接两个网络的输出，然后将拼接后的向量输入给一个标准的逻辑回归模型。公式如下：

$$p = \sigma\left([\mathbf{x}_{L_1}^T, \mathbf{h}_{L_2}^T]\mathbf{w}_{\text{logits}}\right), \qquad (5)$$

其中，X和h分别为cross network和deep network的输出。w为连接层的权重。σ(x) 是一个sigmoid函数。

损失函数是一个带有正则项的交叉熵损失函数：

$$\text{loss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(p_i) + (1 - y_i)\log(1 - p_i) + \lambda \sum_{l} \|\mathbf{w}_l\|^2, \quad (6)$$

其中，pi是公式（5）的输出，yi是真实的label，N为input的样本数量。