

DCN-M: Google提出改进版DCN, 用于大规模排序系统的特征交叉学习(附代码)

原创 深度匹配树 浅梦的学习笔记 今天

收录于话题

#推荐&广告算法技术原理与实践 49 #推荐算法 6 #推荐系统 9

“ 本文结合DeepCTR-Torch中的代码实现, 介绍了DCN的改进版——DCN-M。该模型能更有效地学习特征交叉, 并通过低秩矩阵分解对参数矩阵进行降维, 降低计算成本。受MOE结构启发, 作者还在多个子空间中建模特征交叉。实验表明, 传统的基于ReLU的神经网络在学习高阶特征交叉时效率较低; DCN-M能够在保证效率较高的同时, 取得优于SOTA方法的效果。”

本文介绍的论文是《DCN-M: Improved Deep & Cross Network for Feature Cross Learning in Web-scale Learning to Rank Systems》 论文地址:
<https://arxiv.org/abs/2008.13535>

代码实现: DeepCTR-Torch(<https://github.com/shenweichen/DeepCTR-Torch>)
中 DCN-M 和 DCN-Mix (点击文末阅读原文可访问)

摘要

在大规模(几十亿样本)场景下, DCN^[1]中cross网络的表达能力有限, 无法学到更加有用的特征交叉。尽管学术界做出了大量进展, 但工业界很多深度模型还是依赖于传统的DNN来低效地学习特征交叉。

基于DCN的优缺点以及现有的特征交叉方法, 作者提出了改进版的DCN-M^[2]来使模型更容易在大规模工业场景下落地。大量实验结果表明, DCN-M在学习特征交叉时的表达能力更强且效率较高, 在主流数据集上能够超过SOTA方法。在引入混合低秩矩阵后效果更好。DCN-M结构简单, 容易作为building blocks, 并在许多大规模L2R系统中取得了显著的线下和线上指标提升。

贡献

- 提出了一种新的DCN-M模型来有效地学习显式和隐式特征交叉, 模型高效、简单的同时, 表达能力更强。

- 基于DCN-M中学习出的低秩矩阵，利用低秩方法来在子空间中进行近似特征交叉，在模型效果和时延上达到了更好的权衡。受MOE结构启发，将矩阵分解至多个子空间，随后通过门控机制来对这些子空间进行融合。
- 使用人造数据集进行了研究，结果表明传统的基于ReLU的神经网络在学习高阶特征交叉时效率较低。
- 在Criteo和ml-1m数据上的大量实验表明，DCN-M模型能够显著胜过SOTA方法。

模型

DCN回顾

首先回顾一下DCN的模型结构：

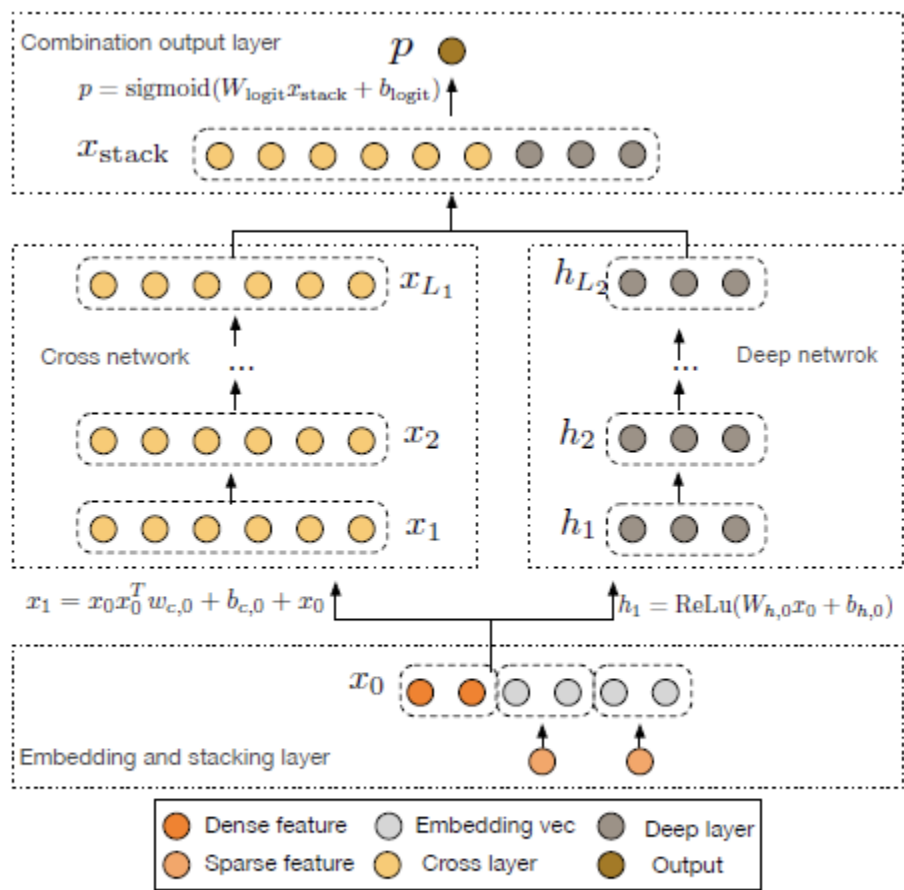


Figure 1: The Deep & Cross Network

特征经过embedding、stack后，分别输入cross network和DNN，两路的输出stack后经过单层nn映射为一维的预测分数。

embedding/stack/DNN不必赘述，主要看cross network。cross network的核心思想是更高效地实现显式特征交叉，每一层的计算如下：

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

其中 $x_l, x_{l+1}, w_l, b_l \in \mathbb{R}^d$

图示：

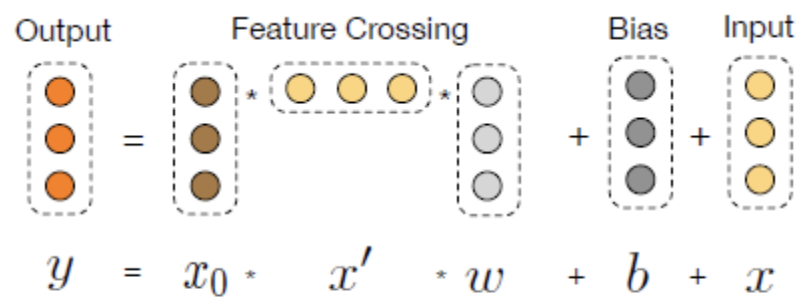


Figure 2: Visualization of a cross layer.

x_0, x_l, w_l, b_l 都是d维的列向量，形状是(d,1)。 $x_0 x_l^T w_l$ 的形状是(d,1) * (1,d) * (d,1) = (d,1)，与 b_l, x_l 一致。cross网络每一层仅增加2d个参数（ w_l 和 b_l ），整体参数量为 $2l_c d$ （ l_c 为网络层数），参数量相比DNN是少得多的。

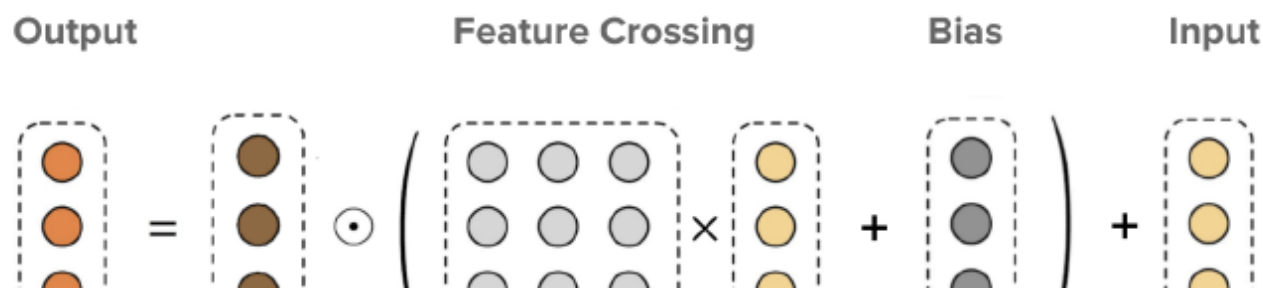
cross网络的改进


DCN中cross网络的参数是向量，DCN-M中换成了矩阵来提高表达能力、方便落地。DCN-M是指“DCN-matrix”，原来的DCN在这里称为DCN-V（“DCN-vector”）。

$$\mathbf{x}_{l+1} = \mathbf{x}_0 \odot (W_l \mathbf{x}_l + \mathbf{b}_l) + \mathbf{x}_l$$

其中 $x_l, x_{l+1}, b_l \in \mathbb{R}^d, W_l \in \mathbb{R}^{d \times d}$

图示：





$$x_{i+1} = x_0 \odot (W \times x_i + b) + x_i$$

Figure 2: Visualization of a cross layer.

目前最新版的DeepCTR-Torch^[3]中已实现了DCN和DCN-M, 只需调整 `parameterization` 参数即可切换模型。其中CrossNet的核心代码如下:

```
if self.parameterization == 'vector':
    x1_w = torch.tensordot(x_1, self.kernels[i], dims=([1], [0]))
    dot_ = torch.matmul(x_0, x1_w)
    x_1 = dot_ + self.bias[i]
elif self.parameterization == 'matrix':
    dot_ = torch.matmul(self.kernels[i], x_1) # W * xi (bs, in_features,
    dot_ = dot_ + self.bias[i] # W * xi + b
    dot_ = x_0 * dot_ # x0 . (W * xi + b) Hadamard-product
    x_1 = dot_ + x_1
```

完整代码地址: https://github.com/shenweichen/DeepCTR-Torch/blob/bc881dcd417fec64f840b0cacce124bc86b3687c/deepctr_torch/layers/interaction.py#L406-L461

Deep和cross的结合方式

结合方式分为堆叠（串行）和并行两种:

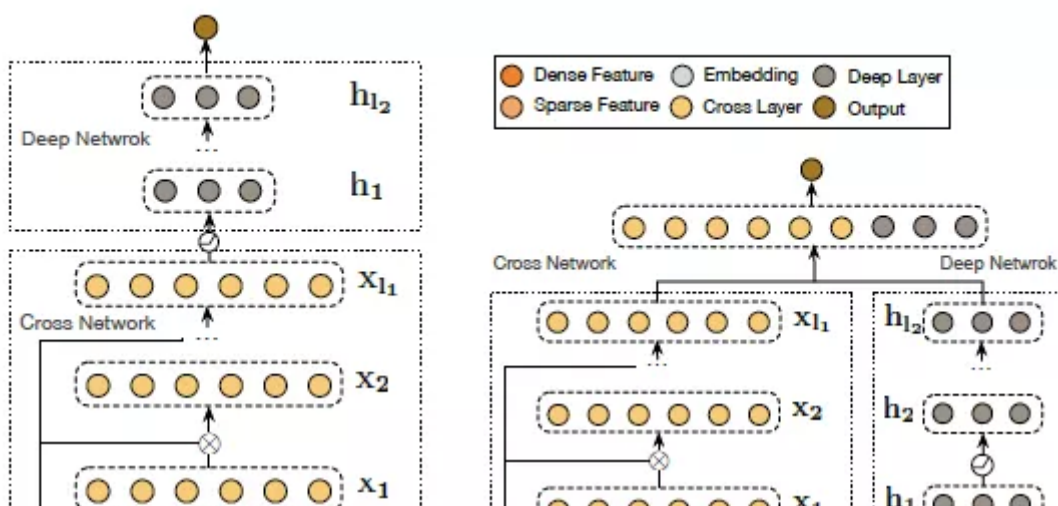




Figure 1: Visualization of DCN-M. \otimes represents the cross operation in Eq. (1), i.e., $x_{l+1} = x_0 \odot (W_l x_l + b_l) + x_l$.

这两种结合方式下的DCN-M效果都优于基准算法。但这两种结构之间的优劣不能一概而论，与数据集有关。串行结构在criteo数据集上更好，而并行结构在Movielen-1M上效果更好。

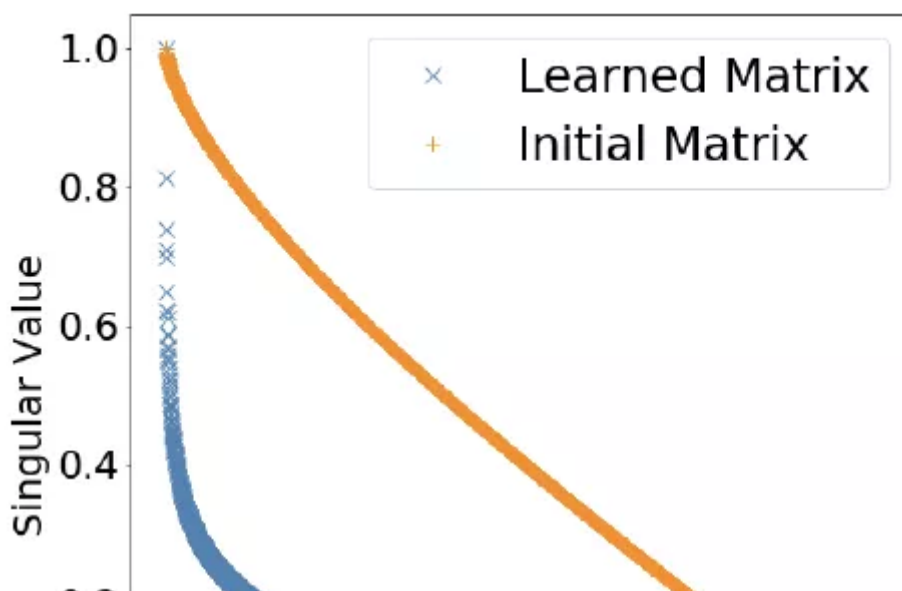
损失函数

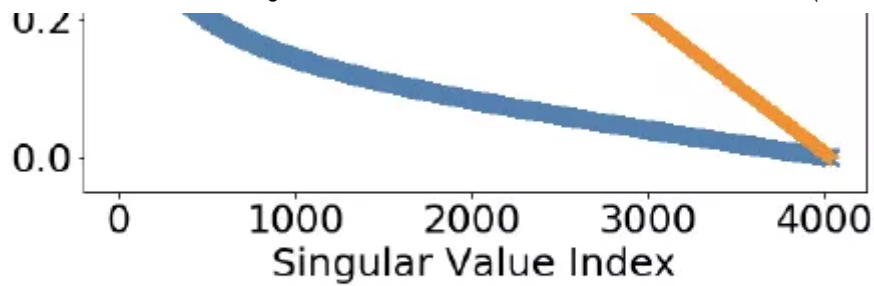
损失函数为带L2正则化的log loss:

$$\text{loss} = -\frac{1}{N} \sum_{i=1}^N y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) + \lambda \sum_l \|W_l\|_2^2$$

混合低秩矩阵

工业界模型往往受计算资源和响应时间限制，需要在保证效果的同时降低计算成本。低秩方法被广泛用于降低计算成本——将一个稠密矩阵近似分解为两个“高瘦”的低秩矩阵。而且，当原矩阵的奇异值差异较大或快速衰减时，低秩分解的方法会更加有效。作者发现，DCN-M中学到的参数矩阵是低秩的（所以比较适合做矩阵分解）。下图展示了DCN-M中学到的参数矩阵的奇异值衰减趋势，比初始化的矩阵衰减更快：





(a) Singular Values

因此, 作者将参数矩阵 $W_l \in \mathbb{R}^{d \times d}$ 分解为了两个低秩矩阵 $U_l, V_l \in \mathbb{R}^{d \times r}$:

$$\mathbf{x}_{l+1} = \mathbf{x}_0 \odot (U_l(V_l^\top \mathbf{x}_i) + \mathbf{b}_l) + \mathbf{x}_i$$

这个公式有两种解释:

(1) 在子空间中学习特征交叉

(2) 将输入特征 \mathbf{x} 映射到低维空间 \mathbb{R}^r 中, 然后再映射回到 \mathbb{R}^d

这两种解释分别激发了作者随后的两处改进:

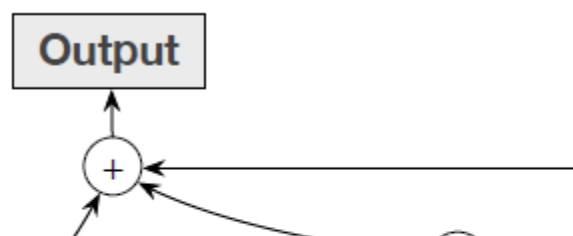
(1) 激发了作者使用 Mixture-of-Experts (MoE) 的思想, 在多个子空间中学习, 然后再进行融合。MOE 方法包含两部分: 专家网络 E (即上个公式中使用低秩矩阵分解的 cross 网络) 和门控单元 G (一个关于输入 x_l 的函数), 通过门控单元来聚合 K 个专家网络的输出结果:

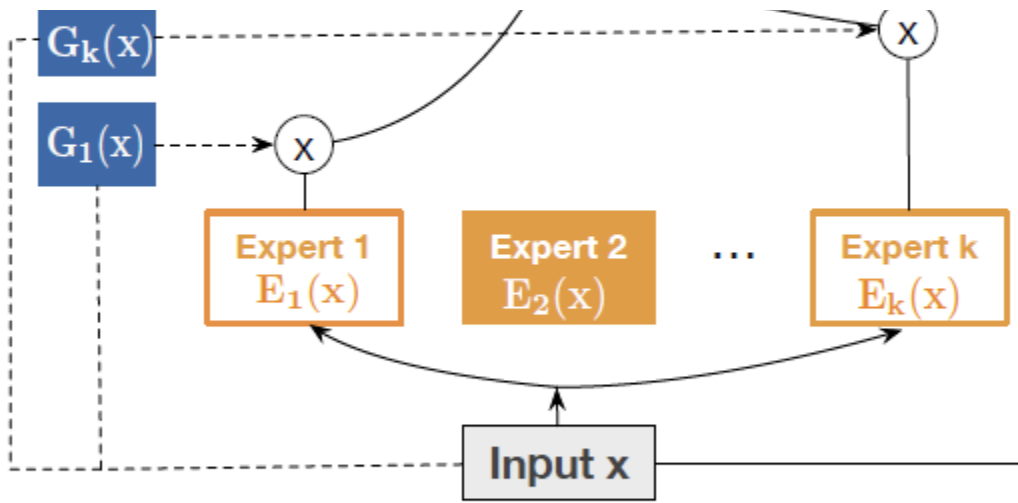
$$\mathbf{x}_{l+1} = \sum_{i=1}^K G_i(\mathbf{x}_l) E_i(\mathbf{x}_l) + \mathbf{x}_l$$

$$E_i(\mathbf{x}_l) = \mathbf{x}_0 \odot (U_l^i (V_l^{i\top} \mathbf{x}_l) + \mathbf{b}_l)$$

图示:

Gatings





(b) Mixture of Low-rank Experts

(2) 激发了作者利用映射空间的低秩性。在映射回原有空间之前，施加了非线性变换来提炼特征：

$$E_i(\mathbf{x}_l) = \mathbf{x}_0 \odot (U_l^i \cdot g(C_l^i \cdot g(V_l^{i\top} \mathbf{x}_l)) + \mathbf{b}_l)$$

此公式的代码实现：（低秩空间中的非线性函数目前采用tanh）

```
# E(x_l)
# project the input x_l to  $\mathbb{R}^r$ 
v_x = torch.matmul(self.V_list[i][expert_id].T, x_l) # (bs, low_rank, 1)

# nonlinear activation in low rank space
v_x = torch.tanh(v_x)
v_x = torch.matmul(self.C_list[i][expert_id], v_x)
v_x = torch.tanh(v_x)

# project back to  $\mathbb{R}^d$ 
uv_x = torch.matmul(self.U_list[i][expert_id], v_x) # (bs, in_features,

dot_ = uv_x + self.bias[i]
dot_ = x_0 * dot_ # Hadamard-product
```

完整代码：[https://github.com/shenweichen/DeepCTR-](https://github.com/shenweichen/DeepCTR-Torch/blob/bc881dcd417fec64f840b0cacce124bc86b3687c/deepctr_torch/la)

[Torch/blob/bc881dcd417fec64f840b0cacce124bc86b3687c/deepctr_torch/la](https://github.com/shenweichen/DeepCTR-Torch/blob/bc881dcd417fec64f840b0cacce124bc86b3687c/deepctr_torch/la)

yers/interaction.py#L464-L537

复杂度

DCN-M中的cross网络的时空复杂度是 $O(d^2 L_c)$, 采用混合低秩矩阵后 (称作DCN-Mix) 的时空复杂度是 $O(2drKL_c)$, 当 $rK \ll d$ 时会更加高效。

实验

「RQ1: 在什么情况下, 显式学习特征交叉的模型能比基于ReLU的DNN更有效? 」

很多CTR的工作都在针对显式特征交叉进行建模 (传统神经网络无法高效地学习到), 但很多工作都只在公开数据集上进行研究, 这些公开数据集上特征交叉的模式是未知的, 且包含许多噪声数据。因此, 作者通过特定的特征交叉模式来生成数据集, 验证各模型的效果。

首先考虑「2阶特征交叉」。按照难度由易到难的顺序指定特征交叉的模式:

$$\begin{aligned} f_1(\mathbf{x}) &= x_1^2 + x_1x_2 + x_3x_1 + x_4x_1 \\ f_2(\mathbf{x}) &= x_1^2 + 0.1x_1x_2 + x_2x_3 + 0.1x_3^2 \\ f_3(\mathbf{x}) &= \sum_{(i,j) \in S} w_{ij}x_ix_j, \quad \mathbf{x} \in \mathbb{R}^{100}, |S| = 100 \end{aligned}$$

f_3 中的集合 S 和权重 w_{ij} 是随机指定的。下面我们看看各模型能否有效的学习到这些特征交叉 (CN是指单独的Cross Network):

Table 1: Polynomial Fitting of Increasing Difficulty.

Model	CN-V-1Layer	CN-M-1Layer	DNN-1Layer	DNN-large
f_1	8.9E-13	5.1E-13	2.7E-02	4.7E-03
f_2	1.0E-01	4.5E-15	3.0E-02	1.4E-03
f_3	3.6E+00	3.0E-07	3.8E-01	1.5E+00

Entries are RMSE values. The smaller the better.

从RMSE上来看模型拟合的效果：CN-V和CN-M效果较好。当交叉的模式变得复杂时（ f_3 ），所有方法的效果都有所下降，但CN-M仍然是很准确的。DNN的效果较差，即使是使用更宽、更深的DNN（DNN-large），效果仍然较差。

「1-4阶特征交叉」（与实际情况较为接近）：

$$f(\mathbf{x}) = \mathbf{x}^\top \mathbf{w} + \sum_{\alpha \in S} w_\alpha x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_d^{\alpha_d} + 0.1 \sin(2\mathbf{x}^\top \mathbf{w}_s + 0.1) + 0.01\epsilon$$

CN-M和DNN的效果如下表所示：

Table 2: Combined-order (1 - 4) Polynomial Fitting.					
#Layers	1	2	3	4	5
CN-M	1.43E-01	2.89E-02	9.82E-03	9.87E-03	9.92E-03
DNN	1.32E-01	1.03E-01	1.03E-01	1.09E-01	1.05E-01

当增大层数时，CN-M能够捕捉数据中更高阶的特征交叉、达到更好的效果。由于CN-M中的残差项和偏置项，即使模型超过3层（引入了多余的特征交叉），效果也没有变差。

「RQ2：去掉DNN后，baselines中的特征交叉部分表现如何？」

数据集：Criteo

Table 5: LogLoss (test) of feature interaction component of each model (no DNN). Only categorical features were used. In the ‘Setting’ column, l stands for number of layers.

	Model	LogLoss	Best Setting
2nd	PNN [34]	$0.4715 \pm 4.430\text{e-}04$	OPNN, kernel=matrix
	FM	$0.4736 \pm 3.04\text{E-}04$	–
>2	CIN [25]	$0.4719 \pm 9.41\text{E-}04$	$l=3$, cinLayerSize=100
	AutoInt [45]	$0.4711 \pm 1.62\text{E-}04$	$l=2$, head=3, attEmbed=40
	DNN	$0.4704 \pm 1.57\text{E-}04$	$l=2$, size=1024
	CrossNet	$0.4702 \pm 3.80\text{E-}04$	$l=2$
	CrossNet-Mix	$0.4694 \pm 4.35\text{E-}04$	$l=5$, expert=4, gate= $\frac{1}{1+e^{-x}}$

1. 更高阶的模型会比2阶的模型效果更好，说明在Criteo数据集上更高阶的交叉也是有意义的。
2. 在高阶模型中，Cross Network取得了最好的效果

「RQ3 DCN-M的效果与baselines相比如何？能否在准确性和计算成本上取得更好的权衡？」

数据集：Criteo、ml-1m

	Model	Criteo					MovieLens-1M		
		Logloss	#Params	FLOPS	Best Setting		Logloss	#Params	FLOPS
Baselines	PNN	0.4421 ± 5.75E-04	3.06E+06	6.11E+06	(3, 1024)	OPNN	0.3182 ± 1.4E-03	5.4E+04	1.1E+05
	DeepFm	0.4420 ± 1.39E-04	1.38E+06	2.78E+06	(2, 768)	-	0.3202 ± 1.0E-03	4.6E+04	9.3E+04
	DLRM	0.4427 ± 3.09E-04	1.06E+06	2.15E+06	(2, 768)	[512,256,64]	0.3245 ± 1.1E-03	7.7E+03	1.6E+04
	xDeepFm	0.4421 ± 1.56E-04	3.67E+06	3.19E+07	(3, 1024)	$l=2, n=100$	0.3251 ± 4.3E-03	1.6E+05	9.9E+05
	AutoInt+	0.4420 ± 5.71E-05	4.22E+06	8.67E+06	(4, 1024)	$l=2, h=2, e=40$	0.3204 ± 4.4E-04	2.6E+05	5.0E+05
	DCN-V	0.4420 ± 1.60E-04	2.10E+06	4.20E+06	(2, 1024)	$l=4$	0.3197 ± 1.9E-04	1.1E+05	2.2E+05
	DNN	0.4421 ± 6.49E-05	3.15E+06	6.30E+06	(3, 1024)	-	0.3201 ± 4.1E-04	4.6E+04	9.2E+04
Ours	DCN-M	0.4406 ± 6.15E-05	3.45E+06	6.98E+06	(2, 768)	$l=2$	0.3170 ± 3.6E-04	1.1E+05	2.2E+05
	DCN-Mix	0.4408 ± 1.02E-04	2.38E+06	4.76E+06	(2, 512)	$l=3, K=4, r=258$	0.3160 ± 4.9E-04	1.1E+05	2.1E+05
	CrossNet	0.4413 ± 2.45E-04	2.12E+06	4.24E+06	-	$l=4, K=4, r=258$	0.3185 ± 3.0E-04	6.5E+04	1.3E+05

FLOPS是模型运行时间的近似估计。大部分模型的运行时间大约是参数量#Params的2倍，但xDeepFM却高出了一个数量级，难以落地。DCN-M效果最好，而且相对来说效率比较高；DCN-Mix进一步降低了计算成本，在准确性和计算成本上实现了更好的权衡。

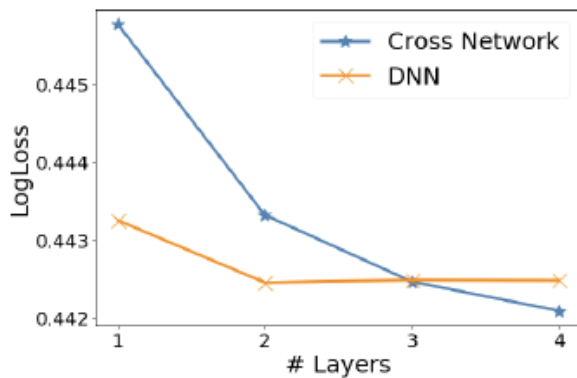
「RQ4 cross网络能否替代ReLU层？」

Table 7: Logloss (test) with a fixed memory budget.

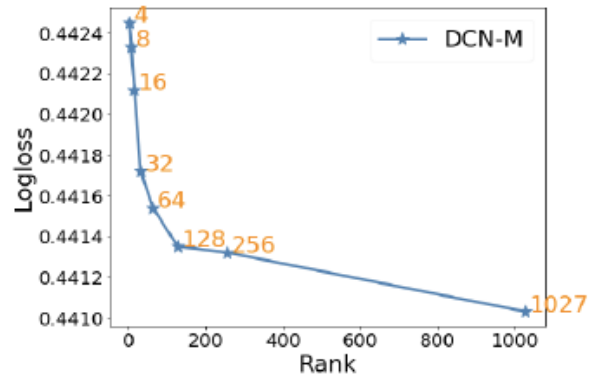
#Params	7.9E+05	1.3E+06	2.1E+06	2.6E+06
CrossNet	0.4424	0.4417	0.4416	0.4415
DNN	0.4427	0.4426	0.4423	0.4423

作者进一步对比了DNN和CrossNet的效果。由于实际生产环境中资源有限，往往需要限制模型大小。因此作者限制了模型的内存占用（即参数量）。结果显示，在相同的参数量限制下，CrossNet的效果更好。那是不是说CrossNet就能替代ReLU层？作者表示：还需要更多实验和分析...

「RQ5 DCN-M中的各项参数是如何影响模型效果的？」



(a) Layer depth



(b) Matrix Rank

Figure 5: Logloss (test) v.s. depth & matrix rank.

1. 网络层数：

当cross网络层数增加时，效果会稳定提升，说明能够捕捉更有用的交叉。但提升的速度越来越慢，说明高阶特征交叉的作用是低于低阶交叉的。作者也对比了一个相同规模的DNN，层数 ≤ 2 时DNN效果比cross网络更好，但层数更多时，差距会减小甚至出现反超。

2. 矩阵的秩：

当秩小于64时，logloss几乎是呈线性下降；大于64时下降速度放缓。这说明最重要的特征能够被最大的64个奇异值所捕捉。

Table 8: Logloss (test) of varying # low-rank experts.

#Experts	1	4	8	16	32
LogLoss	0.4418	0.4416	0.4416	0.4422	0.4420

3. 专家网络的数量：

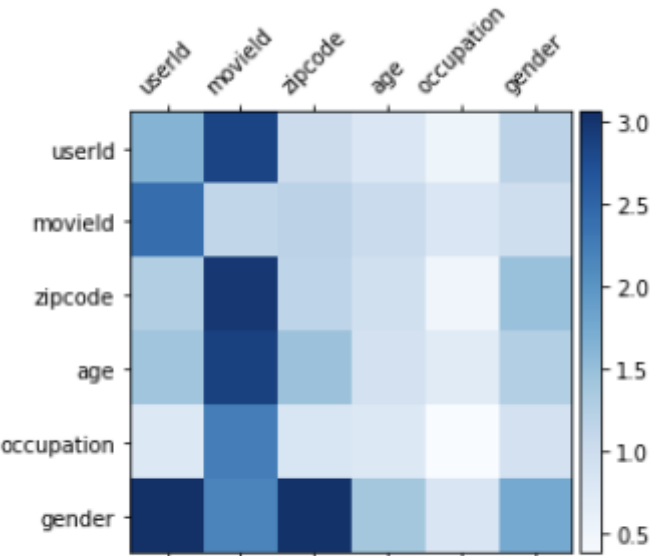
当其他参数设置为最优时，使用更多的专家网络并没有明显的提升，这可能是由于门控机制和优化方法比较朴素。作者认为，如果采用更精细化的门控机制和优化方法，会从MOE结构中取得更大收益。

「RQ6 DCN-M能否捕捉重要的特征交叉？」

DCN-M中的权重矩阵 W 能够反映不同交叉特征的重要程度：

$$\mathbf{x} \odot W \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} \odot \begin{bmatrix} W_{1,1} & W_{1,2} & \cdots & W_{1,k} \\ W_{2,1} & W_{2,2} & \cdots & W_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ W_{k,1} & W_{k,2} & \cdots & W_{k,k} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix}$$

根据 W 绘制出不同交叉特征的权重图谱：



(b) Movielen-1M

可以看到，模型能够学到一些强特征，例如 $\text{gender} \times \text{userid}$ ， $\text{movielid} \times \text{userid}$ 。

总结

DCN-M模型能够简单且有效地建模显式特征交叉，并通过混合低秩矩阵在模型效果和时延上实现了更好的权衡。DCN-M已成功应用于多个大型L2R系统，取得了显著的线下及线上收益。实验结果表明DCN-M的效果超过了现有SOTA方法。

参考资料

[1] Wang R, Fu B, Fu G, et al. Deep & cross network for ad click predictions[M]//Proceedings of the ADKDD'17. 2017: 1-7.

[2] Wang R, Shivanna R, Cheng D Z, et al. DCN-M: Improved Deep & Cross Network for Feature Cross Learning in Web-scale Learning to Rank Systems[J]. arXiv preprint arXiv:2008.13535, 2020.