深度推荐模型

乔梁 2022.7.8

深度学习推荐模型

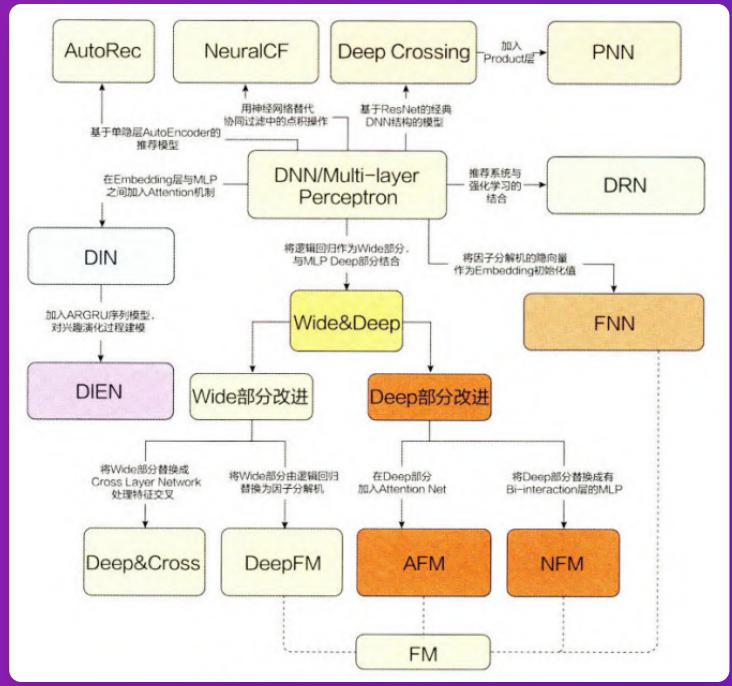
在进入深度学习时代之后，推荐模型主要在以下两方面取得了重大进展。

1. 与传统的机器学习模型相比，深度学习模型的表达能力更强，能够挖掘出更多数据中潜藏的模式。
2. 深度学习的模型结构非常灵活，能够根据业务场景和数据特点，灵活调整模型结构，使模型与应用场景完美契合。

从技术角度讲，深度学习推荐模型大量借鉴并融合了深度学习在图像、语音及自然语言处理方向的成果，在模型结构上进行了快速的演化。 接下来介绍的模型尽量遵循下面三个原则：

1. 模型在工业界和学术界影响力较大。
2. 模型已经被谷歌、阿里巴巴、微软等知名互联网公司成功应用。
3. 在深度学习推荐系统发展过程中起到重要的节点作用。

深度学习推荐模型的演化关系图



演变思路

1. 改变神经网络的复杂程度：AutoRec、Deep Crossing
2. 改变特征交叉方式：Neural CF、PNN
3. 组合模型：Wide&Deep、Deep&Cross、DeepFM
4. FM 模型的深度学习演化版本：NFM、FNN、AFM
5. 注意力机制与推荐模型的结合：AFM、DIN
6. 序列模型与推荐系统的结合：DIEN
7. 强化学习与推荐系统的结合：DRN

AutoRec——单隐层神经网络推荐模型

模型结构

2015 年由澳大利亚国立大学提出了 AutoRec 模型，AutoRec 模型是一个标准的自编码器，它的基本原理是利用协同过滤中的共现矩阵，完成物品向量或者用户向量的自编码。再利用自编码的结果得到用户对物品的预估评分，进而进行推荐排序。

重建函数

$$h(r;\theta) = f(W \cdot g(Vr + \mu) + b)$$

目标函数

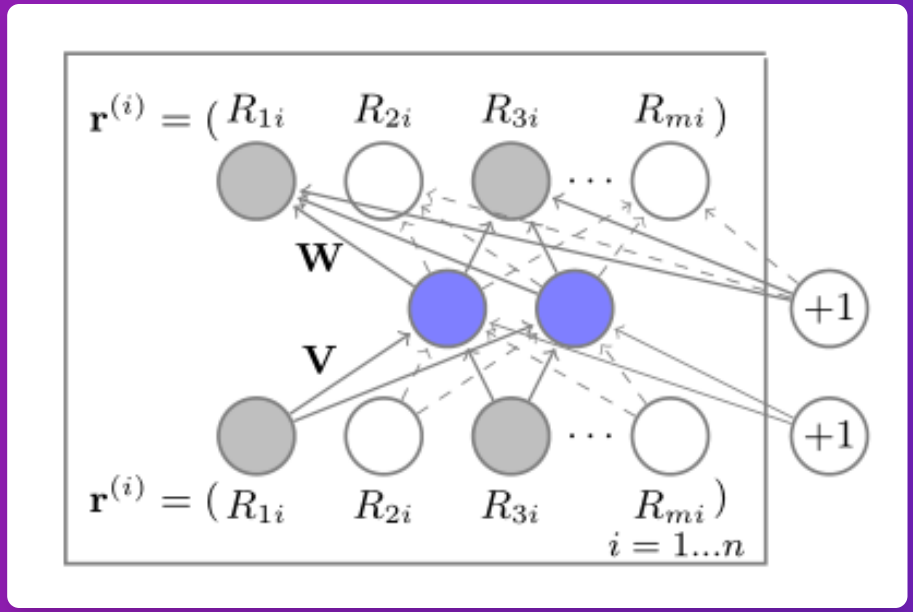
$$\min_{\theta} \sum_{i=1}^{n} \left\| \boldsymbol{r}^{(i)} - h\left(\boldsymbol{r}^{(i)};\theta\right) \right\|_{O}^{2} + \frac{\lambda}{2} \cdot \left(\|\boldsymbol{W}\|_{F}^{2} + \|\boldsymbol{V}\|_{F}^{2} \right)$$

预测评分

$$\hat{R}_{ui} = \left(h\left(\mathbf{r}^{(i)};\hat{\theta}\right) \right)_{u}$$

AutoRec——单隐层神经网络推荐模型

模型结构



AutoRec——单隐层神经网络推荐模型

AutoRec 的特点和局限性

- AutoRec 模型从神经网络的角度出发，使用一个单隐层的 AutoEncoder 泛化用户或物品评分，使模型具有一定的泛化和表达能力。由于 AutoRec 模型的结构比较简单，使其存在一定的表达能力不足的问题。
- 在模型结构上，AutoRec 模型和后来的词向量模型 Word2vec 完全一致， 但优化目标和训练方法有所不同。
- 从深度学习的角度来说，AutoRec 模型的提出，拉开了使用深度学习的思想解决推荐问题的序幕，为复杂深度学习网络的构建提供了思路。

重建函数输入也是评分，输出也是评分，有什么用呢?

输入输出都是得分向量，但是输入的时候，共现矩阵是稀疏的，一定有很多缺失的值。我们的目标就是预测这些缺失的值。而最开始输入的得分，在后面也会得到一个预测值，那么拿这个预测值和输入的真实值比较就是误差。只有让这个误差更小，我们预测那些缺失的值才会更精准。这和矩阵分解的想法很相似。

Deep Crossing——经典的深度学习架构

模型结构

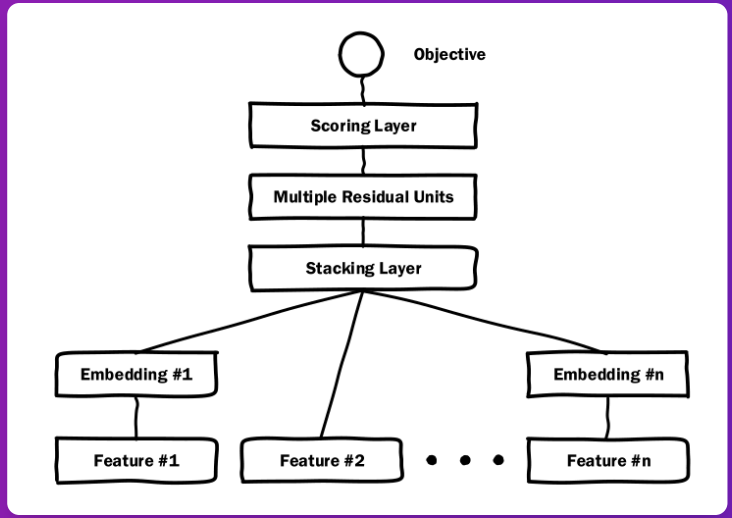
相比 AutoRec 模型过于简单的网络结构带来的一些表达能力不强的问题，Deep Crossing 模型完整地解决了从特征工程、稀疏向量稠密化、多层神经网络进行优化目标拟合等一系列深度学习在推荐系统中的应用问题，为后续的研究打下了良好的基础。微软于 2016 年提出了 Deep Crossing 模型。

应用场景

Deep Crossing模型的应用场景是微软搜索引擎 Bing中的搜索广告推荐场景。 用户在搜索引擎中输入搜索词之后，搜索引擎除了会返回相关结果，还会返回与搜索词相关的广告，这也是大多数搜索引擎的主要赢利模式。尽可能地增加搜索广告的点击率，准确地预测广告点击率，并以此作为广告排序的指标之一，是非常重要的工作，也是 Deep Crossing 模型的优化目标。

Deep Crossing——经典的深度学习架构

模型结构



Deep Crossing——经典的深度学习架构

模型结构

Embedding 层：Embedding 层的作用是将稀疏的类别型特征转换成稠密的 Embedding 向量。每一个特征（如 Feature#1，这里指的是经 one-hot 编码后的稀疏特征向量）经过 Embedding 层后，会转换成对应的 Embedding 向量（如 Embedding#1)。Feature#2 实际上代表了数值型特征，可以看到，数值型特征不需要经过 Embedding 层，直接进入了 Stacking 层。

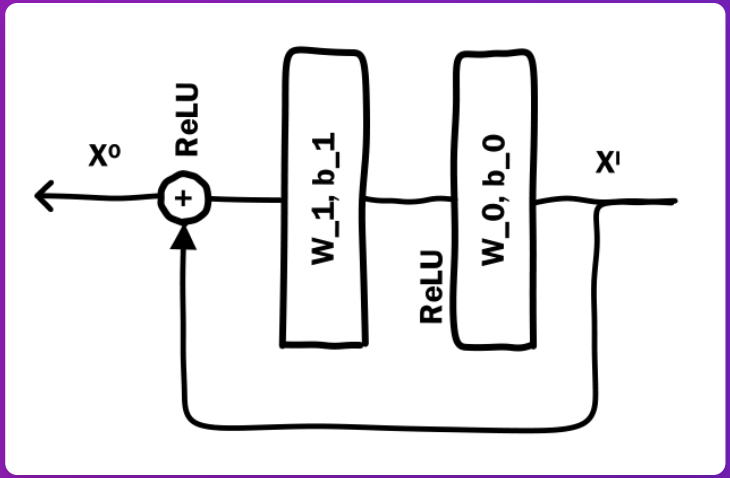
Stacking 层：Stacking 层（堆叠层）的作用比较简单，是把不同的 Embedding 特征和数值型特征拼接在一起，形成新的包含全部特征的特征向量，该层通常也被称为连接 concatenate 层。

Multiple Residual Units 层：该层的主要结构是多层感知机，相比标准的以感知机为基本单元的神经网络，Deep Crossing 模型采用了多层残差网络（Multi-Layer Residual Network）作为 MLP 的具体实现。

模型结构：Embedding+MLP+输出层

Deep Crossing——经典的深度学习架构

模型结构



Deep Crossing——经典的深度学习架构

模型结构

$$X^O = \mathcal{F}\left(X^I, \{\mathbf{W}_0, \mathbf{W}_1\}, \{\mathbf{b}_0, \mathbf{b}_1\}\right) + X^I$$

将 X^I 移项到等式左侧，可以看出函数拟合的是输入与输出之间的残差。对输入进行全连接变换之后，经过激活函数送入第二个全连接层，将输出结果与原始输入进行 element-wise add 操作，再经过 $ReLU$ 激活输出。有分析说明，残差结构能更敏感的捕获输入输出之间的信息差。

Scoring 层：Scoring 层作为输出层，就是为了拟合优化目标而存在的。对于 CTR 预估这类二分类问题，Scoring 层往往使用的是逻辑回归模型，而对于图像分类等多分类问题，Scoring 层往往采用 softmax 模型。

损失函数

$$\text{logloss} = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log\left(p_i\right) + \left(1 - y_i\right)\log\left(1 - p_i\right)\right)$$

Deep Crossing——经典的深度学习架构

对 Deep Crossing 的评价：

- 从目前的时间节点上看，Deep Crossing 模型是平淡无奇的，因为它没有引入任何诸如注意力机制 、序列模型等特殊的模型结构 ，只是采用了常规的“Embedding+多层神经网络” 的经典深度学习结构。
- 但从历史的尺度看，Deep Crossing 模型的出现是有革命意义的。Deep Crossing 模型中没有任何人工特征工程的参与，原始特征经 Embedding 后输入神经网络层，将全部特征交叉的任务交给模型。相比FM、FFM 模型只具备二阶特征交叉的能力，Deep Crossing 模型可以通过调整神经网络的深度进行特征之间的 “深度交叉”，这也是 Deep Crossing 名称的由来。

Cross 体现在哪里?

Cross的处理有两种，一种是直接两个特征 Embedding 向量乘积的显示方式，另外一种是 Embedding 喂给 MLP 的隐式方式，两种都有特征 Cross 的作用。作者本意应该是通过残差网络学习隐式的特征交叉，所以叫Deep Crossing。

NeuralCF 模型——CF 与深度学习的结合

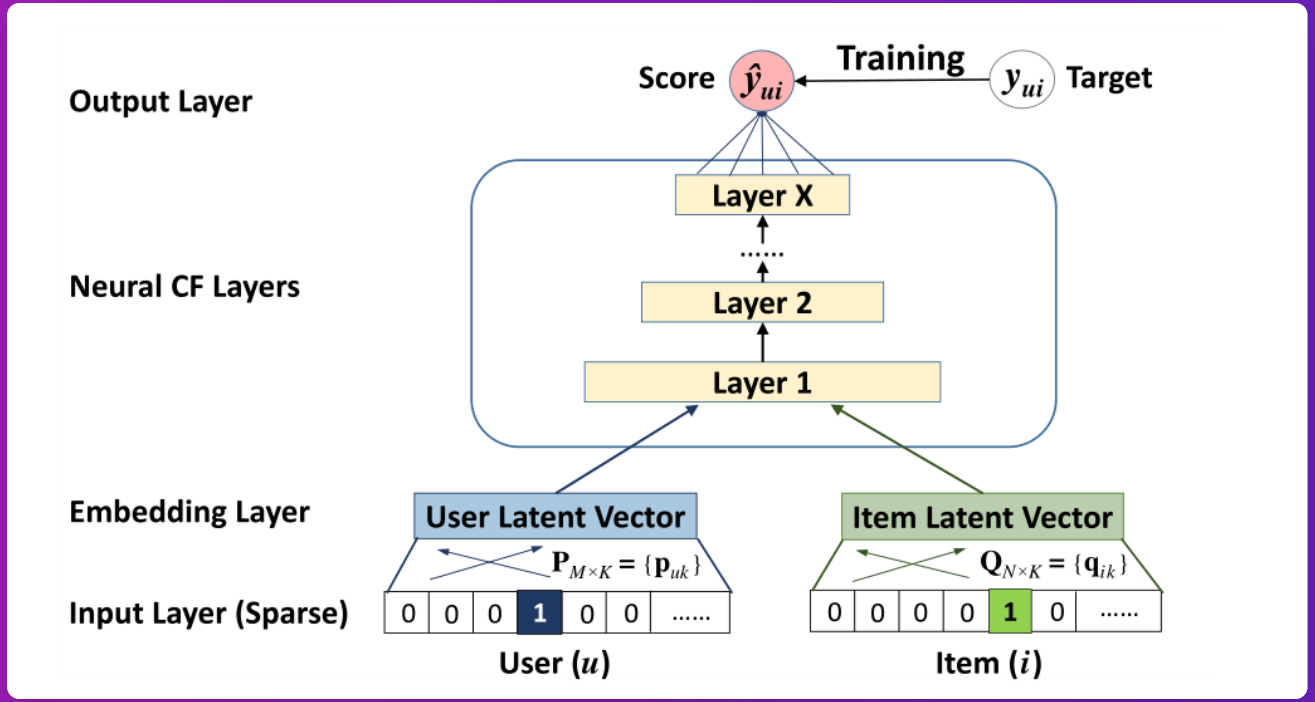
模型结构

矩阵分解是将协同过滤中的共现矩阵分解为用户向量矩阵和物品向量矩阵。其中，用户 u 隐向量和物品 i 隐向量的内积，就是用户 u 对物品 i 评分的预测。沿着矩阵分解的技术脉络，结合深度学习知识，新加坡国立大学的研究人员于 2017 年提出了基于深度学习的协同过滤模型 NeuralCF。

NeuralCF 用“多层神经网络+输出层” 的结构替代了矩阵分解模型中简单的内积操作。这样做的收益是直观的，一是让用户向量和物品向量做更充分的交叉，得到更多有价值的特征组合信息；二是引入更多的非线性特征， 让模型的表达能力更强。以此类推，事实上，用户和物品向量的互操作层可以被任意的**互操作形式**所代替，这就是所谓的 “广义矩阵分解” 模型 Generalized Matrix Factorization)。 原始的矩阵分解使用“内积”的方式让用户和物品向量进行交互，为了进一步让向量在各维度上进行充分交叉，可以通过 “元素积” element-wise product, 长度相同的两个向量的对应维相乘得到另一向量的方式进行互操作，再通过逻辑回归等输出层拟合最终预测目标。NeuralCF 中利用神经网络拟合互操作函数的做法是广义的互操作形式。

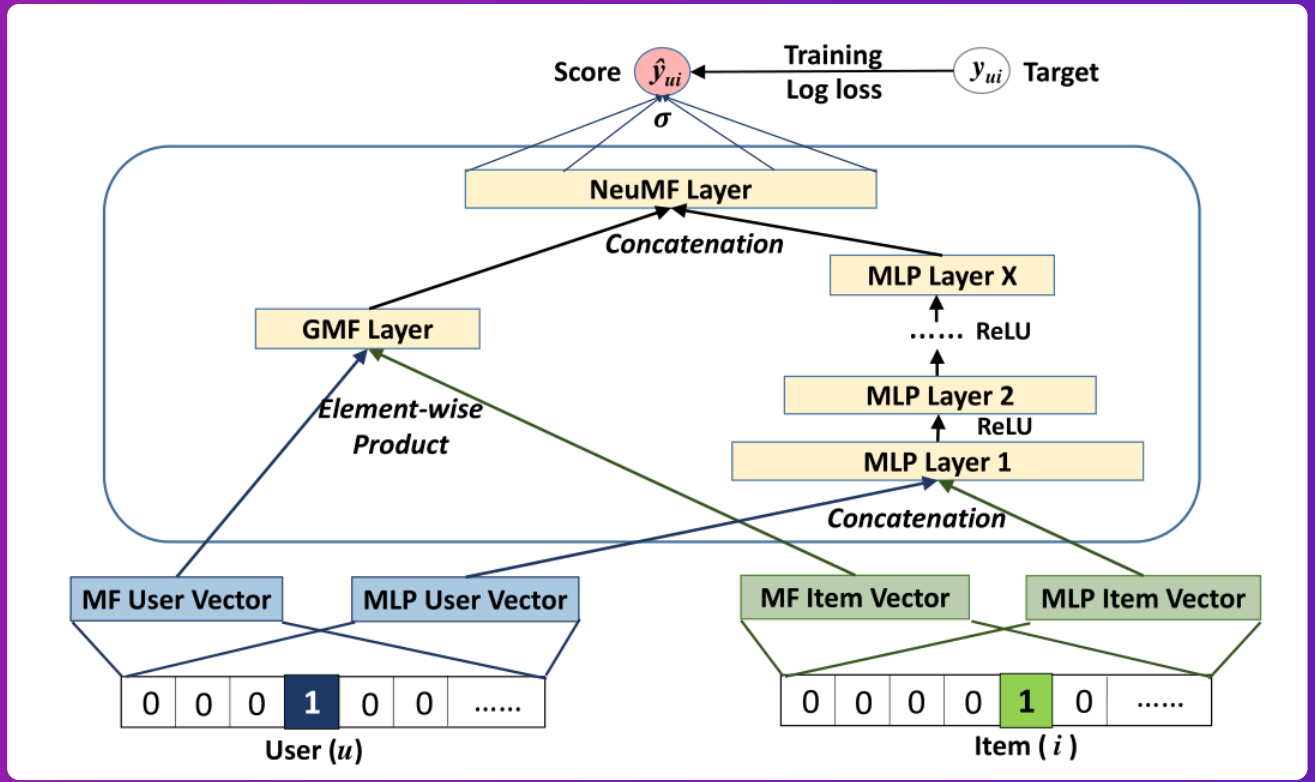
NeuralCF 模型——CF 与深度学习的结合

模型结构



NeuralCF 模型——CF 与深度学习的结合

模型结构



NeuralCF 模型——CF 与深度学习的结合

模型结构

简单方案是让GMF和MLP共享同一嵌入层，然后组合其交互函数的输出：

$$\hat{y}_{ui} = \sigma\left(\mathbf{h}^T a\left(\mathbf{p}_u \odot \mathbf{q}_i + \mathbf{W}\begin{bmatrix}\mathbf{p}_u \\ \mathbf{q}_i\end{bmatrix} + \mathbf{b}\right)\right)$$

然而，GMF和MLP的共享嵌入可能会限制融合模型的性能。例如，这意味着GMF和MLP必须使用相同大小的嵌入；对于两个模型的最佳嵌入大小变化很大的数据集，该解决方案可能无法获得最佳集成。

为了给融合模型提供更大的灵活性，我们允许GMF和MLP学习单独的嵌入，并通过连接其最后一个隐藏层来组合这两个模型。

$$\begin{aligned}
\phi^{GMF} &= \mathbf{p}_u^G \odot \mathbf{q}_i^G \\
\phi^{MLP} &= a_L\left(\mathbf{W}_L^T\left(a_{L-1}\left(\ldots a_2\left(\mathbf{W}_2^T\begin{bmatrix}\mathbf{p}_u^M \\ \mathbf{q}_i^M\end{bmatrix} + \mathbf{b}_2\right)\ldots\right)\right) + \mathbf{b}_L\right) \\
\hat{y}_{ui} &= \sigma\left(\mathbf{h}^T\begin{bmatrix}\phi^{GMF} \\ \phi^{MLP}\end{bmatrix}\right)
\end{aligned}$$

NeuralCF 模型——CF 与深度学习的结合

模型结构

首先用随机初始化训练GMF和MLP，直到收敛。然后，使用其模型参数作为 NeuralMF 参数相应部分的初始化。唯一的调整是在输出层：

$$\mathbf{h} \leftarrow \begin{bmatrix} \alpha \mathbf{h}^{GMF} \\ (1-\alpha)\mathbf{h}^{MLP} \end{bmatrix}$$

NeuralCF 模型——CF 与深度学习的结合

NeuralCF 模型的优势和局限性

- NeuralCF 模型实际上提出了一个模型框架，它基于用户向量和物品向量这两个 Embedding 层，利用不同的互操作层进行特征的交叉组合，并且可以灵活地进行不同互操作层的拼接。从这里可以看出深度学习构建推荐模型的优势——利用神经网络理论上能够拟合任意函数的能力，灵活地组合不同的特征，按需增加或减少模型的复杂度。
- 在实践中要注意：并不是模型结构越复杂、特征越多越好。一是要防止过拟合的风险，二是往往需要更多的数据和更长的训练时间才能使复杂的模型收敛， 这需要算法工程师在模型的实用性、实时性和效果之间进行权衡。
- NeuralCF 模型也存在局限性。由于是基于协同过滤的思想进行构造的，所以 NeuralCF 模型并没有引入更多其他类型的特征，这在实际应用中无疑浪费了其他有价值的信息。此外，对于模型中互操作的种类并没有做进一步的探究和说明。 这都需要后来者进行更深入的探索。

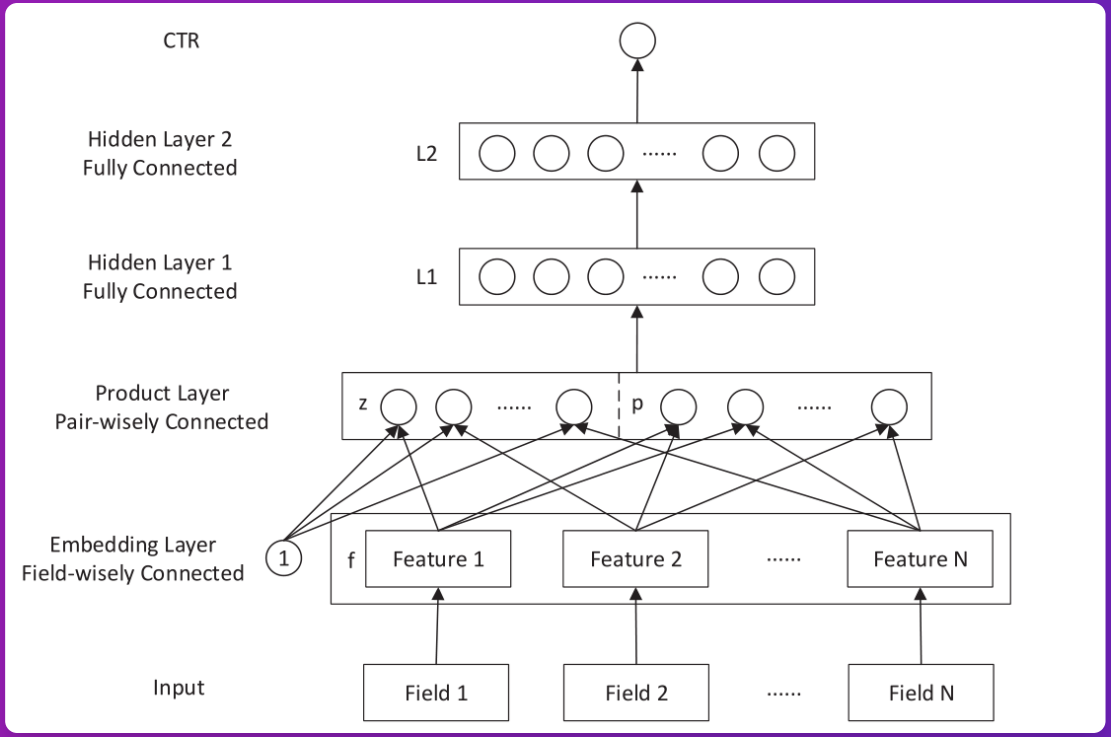
PNN 模型——加强特征交叉能力

模型结构

NeuralCF 模型的主要思想是利用多层神经网络替代经典协同过滤的点积操作，加强模型的表达能力。广义上，任何向量之间的交互计算方式都可以用来替代协同过滤的内积操作，相应的模型可称为广义的矩阵分解模型。但 NeuralCF 模型只提到了用户向量和物品向量两组特征向量，如果加入多组特征向量又该如何设计特征交互的方法呢？2016 年，上海交通大学的研究人员提出的 PNN 模型，给出了特征交互方式的几种设计思路。

PNN 模型——加强特征交叉能力

模型结构



PNN 模型——加强特征交叉能力

模型结构

相比 Deep Crossing 模型，PNN 模型在输入、Embedding 层、多层神经网络，以及最终的输出层部分并没有结构上的不同，唯一的区别在于 PNN 模型用乘积层 Product Layer 代替了 Deep Crossing 模型中的 Stacking 层。也就是说，不同特征的 Embedding 向量不再是简单的拼接，而是用 Product 操作进行两两交互，更有针对性地获取特征之间的交叉信息。

相比 NeuralCF, PNN 模型的输入不仅包括用户和物品信息，还可以有更多不同形式、不同来源的特征，通过 Embedding 层的编码生成同样长度的稠密特征 Embedding 向量。针对特征的交叉方式，PNN 模型也给出了更多具体的互操作方法。

PNN 模型对于深度学习结构的创新主要在于乘积层的引入。具体地说，PNN 模型的乘积层由线性操作部分（乘积层的 z 部分，对各特征向量进行线性拼接）和乘积操作部分（乘积层的 p 部分）组成。其中，乘积特征交叉部分又分为内积操作和外积操作，使用内积操作的 PNN 模型被称为 IPNN (Inner Product-based Neural Network), 使用外积操作的 PNN 模型被称为 OPNN (Outer Product-based Neural Network)。

PNN 模型——加强特征交叉能力

模型结构

PNN 模型在经过对特征的线性和乘积操作后，并没有把结果直接送入上层的 L_1 全连接层，而是在乘积层内部又进行了局部全连接层的转换，分别将线性部分 z 乘积部分映射成了 D_1 维的输入向量 l_z 和 l_p，再将 l_z 和 l_p 叠加，输入 L_2 隐层。

$$\hat{y} = \sigma\left(\boldsymbol{W}_3 \boldsymbol{l}_2 + b_3\right)$$

$$\boldsymbol{l}_2 = \operatorname{relu}\left(\boldsymbol{W}_2 \boldsymbol{l}_1 + \boldsymbol{b}_2\right)$$

$$\boldsymbol{l}_1 = \operatorname{relu}\left(\boldsymbol{l}_z + \boldsymbol{l}_p + \boldsymbol{b}_1\right)$$

$$\begin{aligned} \boldsymbol{l}_z &= \left(l_z^1, l_z^2, \ldots, l_z^n, \ldots, l_z^{D_1}\right), \quad l_z^n = \boldsymbol{W}_z^n \odot \boldsymbol{z} \\ \boldsymbol{l}_p &= \left(l_p^1, l_p^2, \ldots, l_p^n, \ldots, l_p^{D_1}\right), \quad l_p^n = \boldsymbol{W}_p^n \odot \boldsymbol{p} \end{aligned}$$

PNN 模型——加强特征交叉能力

PNN 模型的优点和局限性

- PNN 的结构特点在于强调了特征 Embedding 向量之间的交叉方式是多样化的，相比于简单的交由全连接层进行无差别化的处理，PNN 模型定义的内积和外积操作显然更有针对性地强调了不同特征之间的交互，从而让模型更容易捕获特征的交叉信息。
- 但 PNN 模型同样存在着一些局限性，例如在外积操作的实际应用中，为了优化训练效率进行了大量的简化操作。此外，对所有特征进行无差别的交叉，在 一定程度上忽略了原始特征向量中包含的有价值信息。如何综合原始特征及交叉特征，让特征交叉的方式更加高效，后续的 Wide&Deep 模型和基于 FM 的各类深度学习模型将给出它们的解决方案。

Wide&Deep 模型——记忆能力和泛化能力的综合

模型结构

谷歌于2016 年提出了 Wide&Deep 模型。Wide&Deep 模型的主要思路正如其名，是由单层的 Wide 部分和多层的 Deep 部分组成的混合模型。其中：

- Wide 部分的主要作用是让模型具有较强的“记忆能力” memorization ；
- Deep 部分的主要作用是让模型具有 “泛化能力” generalization。

正是这样的结构特点，使模型兼具了逻辑回归和深度神经网络的优点——能够快速处理并记忆大量历史行为特征，并且具有强大的表达能力，不仅在当时迅速成为业界争相应用的主流模型，而且衍生出了大量以 Wide&Deep 模型为基础结构的混合模型，影响力一直延续到至今。

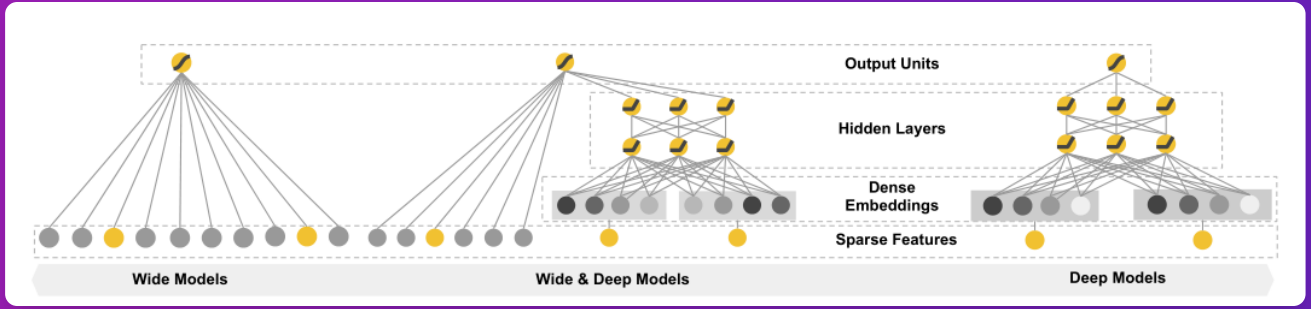
Wide&Deep 模型——记忆能力和泛化能力的综合

模型结构

- “记忆能力”可以被理解为模型直接学习并利用历史数据中物品或者特征的 “共现频率”的能力。像逻辑回归这类简单模型，如果发现这样的 “强特征”，则其相应的权重就会在模型训练过程中被调整得非常大，这样就实现了对这个特征的直接记忆。相反，对于多层神经网络来说，特征会被多层处理，不断与其他特征进行交叉，因此模型对这个强特征的记忆反而没有简单模型深刻。
- “泛化能力”可以被理解为模型传递特征的相关性，以及发掘稀疏特甚至从未出现过的稀有特征与最终标签相关的能力。矩阵分解比协同过滤的泛化能力强，因为矩阵分解引入了隐向量这样的结构，使得数据稀少的用户或者物品也能生成隐向量，从而获得有数据支撑的推荐得分，这就是非常典型的将全局数据传递到稀疏物品上，从而提高泛化能力的例子。再比如，深度神经网络通过特征的多次自动组合，可以深度发掘数据中潜在的模式，即使是非常稀疏的特征向量输入，也能得到较稳定平滑的推荐概率，这就是简单模型所缺乏的 “泛化能力”。

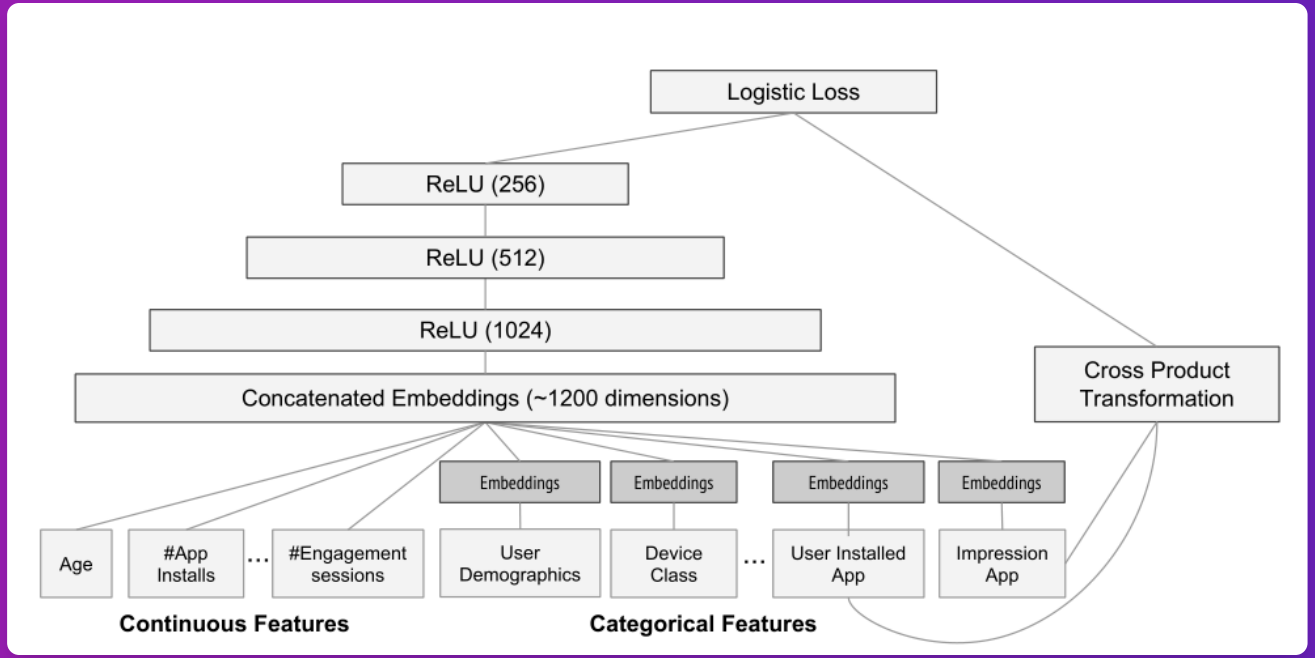
Wide&Deep 模型——记忆能力和泛化能力的综合

模型结构



Wide&Deep 模型——记忆能力和泛化能力的综合

模型结构



Wide&Deep 模型——记忆能力和泛化能力的综合

模型结构

Wide&Deep 模型把单输入层的 Wide 部分与由 Embedding 层和多隐层组成的 Deep 部分连接起来，一起输入最终的输出层。单层的 Wide 部分善于处理大量稀疏的 id 类特征；Deep 部分利用神经网络表达能力强的特点，进行深层的特征交叉，挖掘藏在特征背后的数据模式。最终，利用逻辑回归模型，输出层将 Wide 部分和 Deep 部分组合起来，形成统一的模型。

Deep 部分的输入是全量的特征向量，包括用户年龄 Age 已安装应用数量 #App Installs 、设备类型 Device Class 、已安装应用 User Installed App 、 曝光应用 Impression App 等特征。已安装应用、曝光应用等类别型特征，需要经过 Embedding 层输入连接层 Concatenated Embedding , 拼接成 1200 维的 Embedding 向量，再依次经过 3 层 ReLU 全连接层，最终输入 LogLoss 输出层。

Wide 部分的输入仅仅是已安装应用和曝光应用两类特征，其中已安装应用代表用户的历史行为，而曝光应用代表当前的待推荐应用。选择这两类特征的原因是充分发挥 Wide 部分"记忆能力"强的优势。简单模型善于记忆用户行为特征中的信息，并根据此类信息直接影响推荐结果。Wide 部分是一个广义线性模型，其形式为 $y = \mathbf{w}^T\mathbf{x} + b$ 。特征集包括原始输入特征和变换特征。最重要的转换之一是叉积转换，其定义为：

$$\phi_k(\mathbf{x}) = \prod_{i=1}^{d} x_i^{c_{ki}} \quad c_{ki} \in \{0, 1\}$$

Wide&Deep 模型——记忆能力和泛化能力的综合

对 Wide&Deep 模型的评价

Wide&Deep 模型的提出不仅综合了“记忆能力”和“泛化能力”，而且开启了不同网络结构融合的新思路。Wide&Deep 模型的影响力无疑是巨大的，不仅其本身成功应用于多家一线互联网公司，而且其后续的改进创新工作也延续至今。事实上，DeepFM、NFM 等模型都可以看成 Wide&Deep 模型的延伸。

Wide&Deep 模型能够取得成功的关键在于:

1. 抓住了业务问题的本质特点，能够融合传统模型记忆能力和深度学习模型泛化能力的优势。
2. 模型的结构并不复杂，比较容易在工程上实现、训练和上线，这加速了其在业界的推广应用。

正是从 Wide&Deep 模型之后，越来越多的模型结构被加入推荐模型中， 深度学习模型的结构开始朝着多样化、复杂化的方向发展。

Deep&Cross 模型——Wide&Deep 模型的进化

模型结构

2017年斯坦福大学和谷歌的研究人员提出了 Deep&Cross 模型。Deep&Cross 模型的主要思路是使用 Cross 网络替代原来的 Wide 部分。

Embedding 和 stacking 层

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

把 $\mathbf{x}_0$ 喂入网络。

Cross Network

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

交叉层在增加参数方面是比较"克制"的，每一层仅增加了一个维的权重向量维入向量维度，并且在每一层均保留了输入向量，因此输出与输入之间的变化不会特别明显。由多层交叉层组成的 Cross 网络在 Wide&Deep 模型中 Wide 部分的基础上进行特征的自动化交叉，避免了更多基于业务理解的人工特征组合。同 Wide&Deep 模型一样，Deep&Cross 模型的 Deep 部分相比 Cross 部分表达能力更强，使模型具备更强的非线性学习能力。

Deep&Cross 模型——Wide&Deep 模型的进化

模型结构

Deep Network

$$\mathbf{h}_{l+1} = f\left(W_l \mathbf{h}_l + \mathbf{b}_l\right)$$

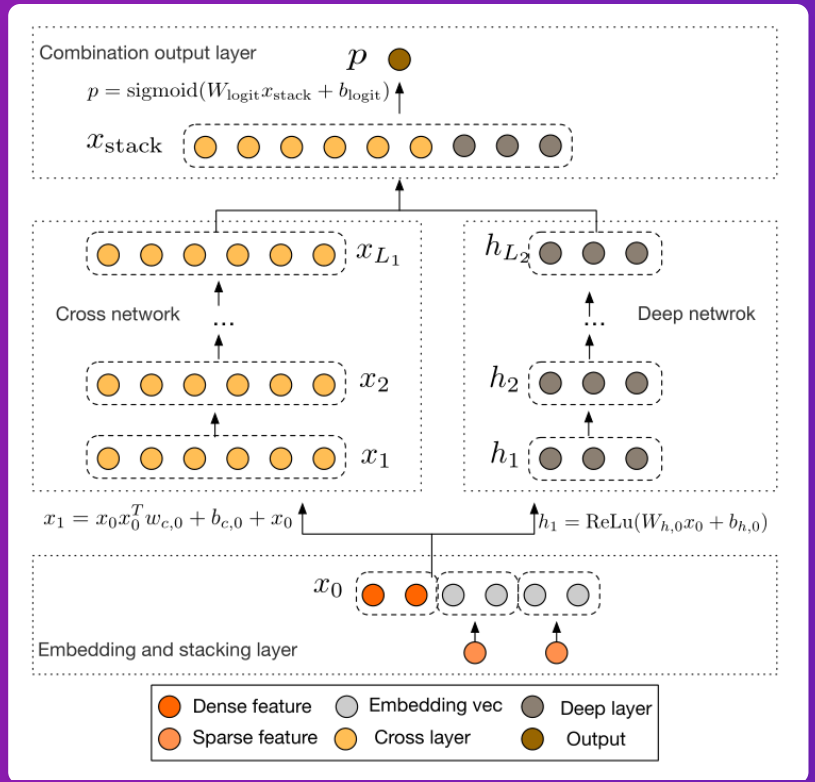
Combination 层

组合层将来自两个网络的输出串联起来，并将串联的向量馈送到标准logits层。

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

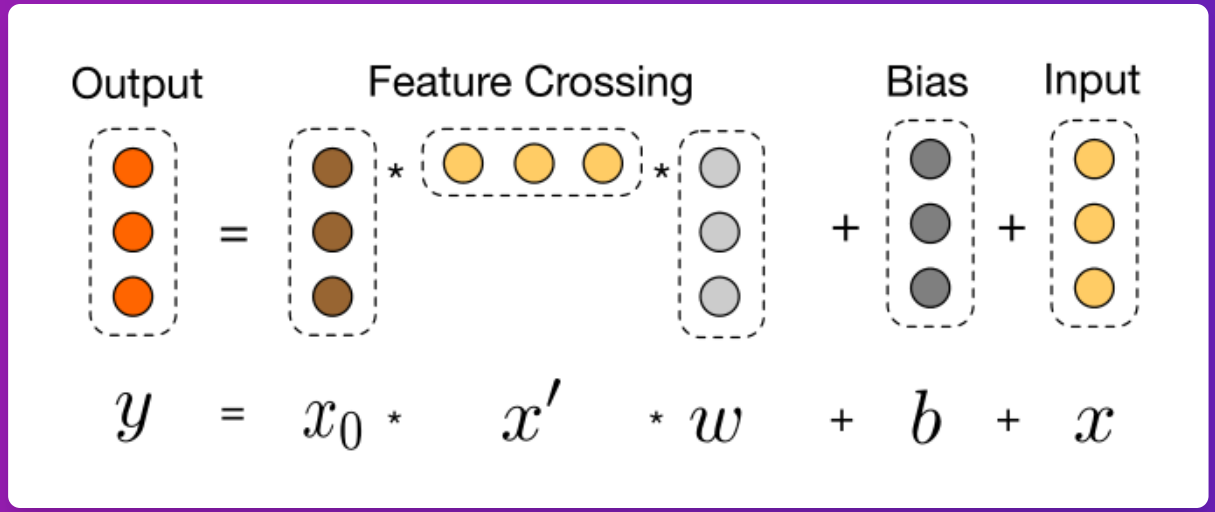
Deep&Cross 模型——Wide&Deep 模型的进化

模型结构



Deep&Cross 模型——Wide&Deep 模型的进化

模型结构



FNN——用 FM 的隐向量完成 Embedding 初始化

模型结构

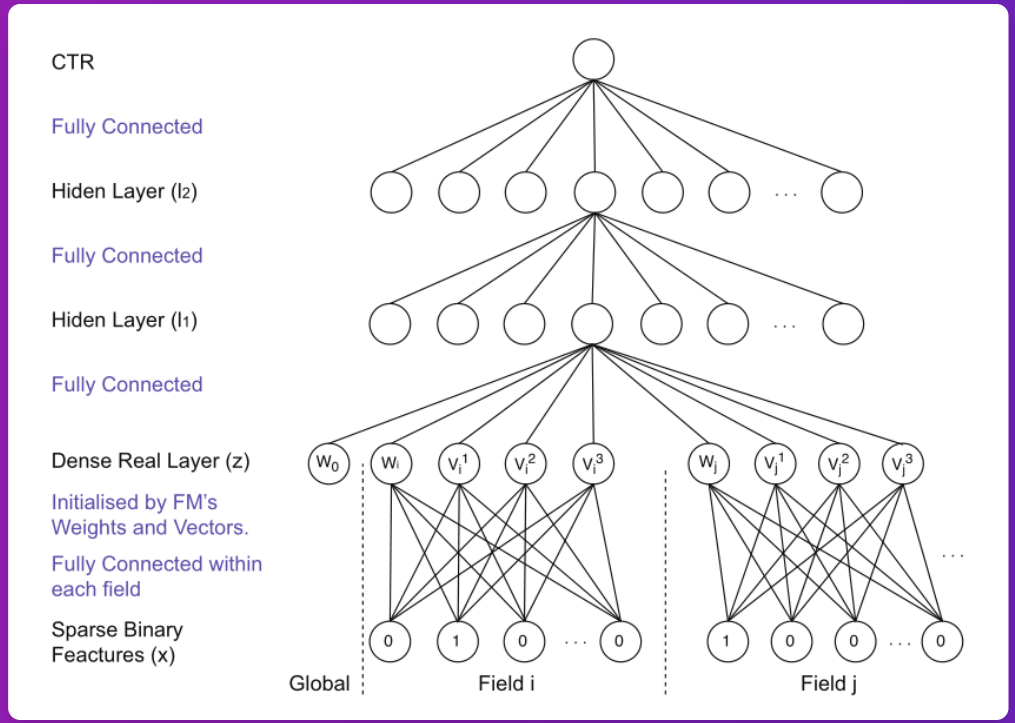
伦敦大学学院的研究人员于 2016 年提出了 FNN 模型。FNN 模型的结构初步看是一个类似 Deep Crossing 模型的经典深度神经网络，从稀疏输入向量到稠密向量的转换过程也是经典的 Embedding 层的结构。

在神经网络的参数初始化过程中， 往往采用随机初始化这种不包含任何先验信息的初始化方法。由于 Embedding 层的输入极端稀疏化，导致 Embedding 层的收敛速度非常缓慢。再加上 Embedding 层的参数数量往往占整个神经网络参数数量的大半以上，因此模型的收敛速度往往受限于 Embedding 层。

针对 Embedding 层收敛速度的难题，FNN 模型的解决思路是用 FM 模型训练好的各特征隐向量初始化 Embedding 层的参数，相当于在初始化神经网络参数时，已经引入了有价值的先验信息。也就是说，神经网络训练的起点更接近目标最优点，自然加速了整个神经网络的收敛过程。

FNN——用 FM 的隐向量完成 Embedding 初始化

模型结构



FNN——用 FM 的隐向量完成 Embedding 初始化

模型结构

$$\hat{y} = \text{sigmoid}\left(\boldsymbol{W}_3 \boldsymbol{l}_2 + b_3\right)$$

$$\boldsymbol{l}_2 = \tanh\left(\boldsymbol{W}_2 \boldsymbol{l}_1 + \boldsymbol{b}_2\right)$$

$$\boldsymbol{l}_1 = \tanh\left(\boldsymbol{W}_1 \boldsymbol{z} + \boldsymbol{b}_1\right)$$

$$\boldsymbol{z} = \left(w_0, \boldsymbol{z}_1, \boldsymbol{z}_2, \ldots \boldsymbol{z}_i, \ldots, \boldsymbol{z}_n\right)$$

$$\boldsymbol{z}_i = \boldsymbol{W}_0^i \cdot \boldsymbol{x}\left[\text{start}_i : \text{end}_i\right] = \left(w_i, v_i^1, v_i^2, \ldots, v_i^K\right)$$

z_i 作为参数矩阵的某一列。

FNN——用 FM 的隐向量完成 Embedding 初始化

对 FNN 模型的评价

FNN 模型除了可以使用 FM 参数初始化 Embedding 层权重，也为另一种 Embedding 层的处理方式——Embedding 预训练提供了借鉴思路。

DeepFM——用 FM 代替 Wide 部分

源码

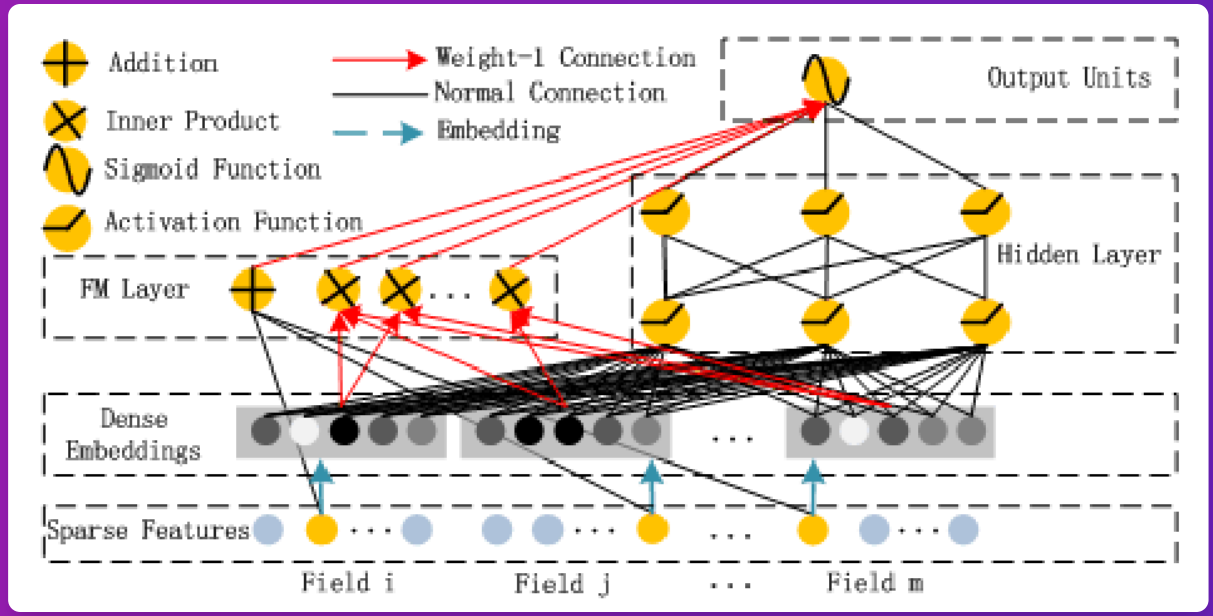
模型结构

FNN 把 FM 的训练结果作为初始化权重，并没有对神经网络的结构进行调整， 2017 年哈尔滨工业大学和华为公司联合提出了DeepFM模型。DeepFM 则将 FM 的模型结构与 Wide&Deep 模型进行了整合。

在 Wide&Deep 模型之后，诸多模型延续了双模型组合的结构，DeepFM 就是其中之一。DeepFM 对 Wide&Deep 模型的改进之处在于，它用 FM 替换了原来的 Wide 部分，加强了浅层网络部分特征组合的能力。左边的 FM 部分与右边的深度神经网络部分共享相同的 Embedding 层。左侧的FM部分对不同的特征域的 Embedding 进行了两两交叉,也就是将 Embedding 向量当作原 FM 中的特征隐向量。最后将 FM 的输出与 Deep 部分的输出一同输入最后的输出层，参与最后的目标拟合。

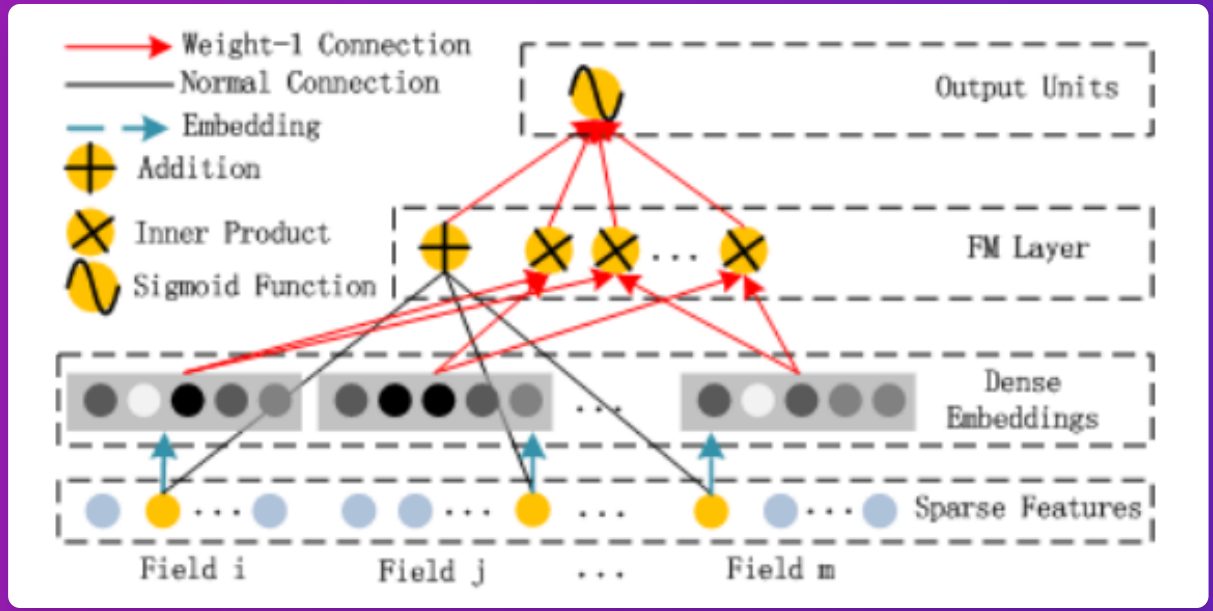
DeepFM——用 FM 代替 Wide 部分

模型结构



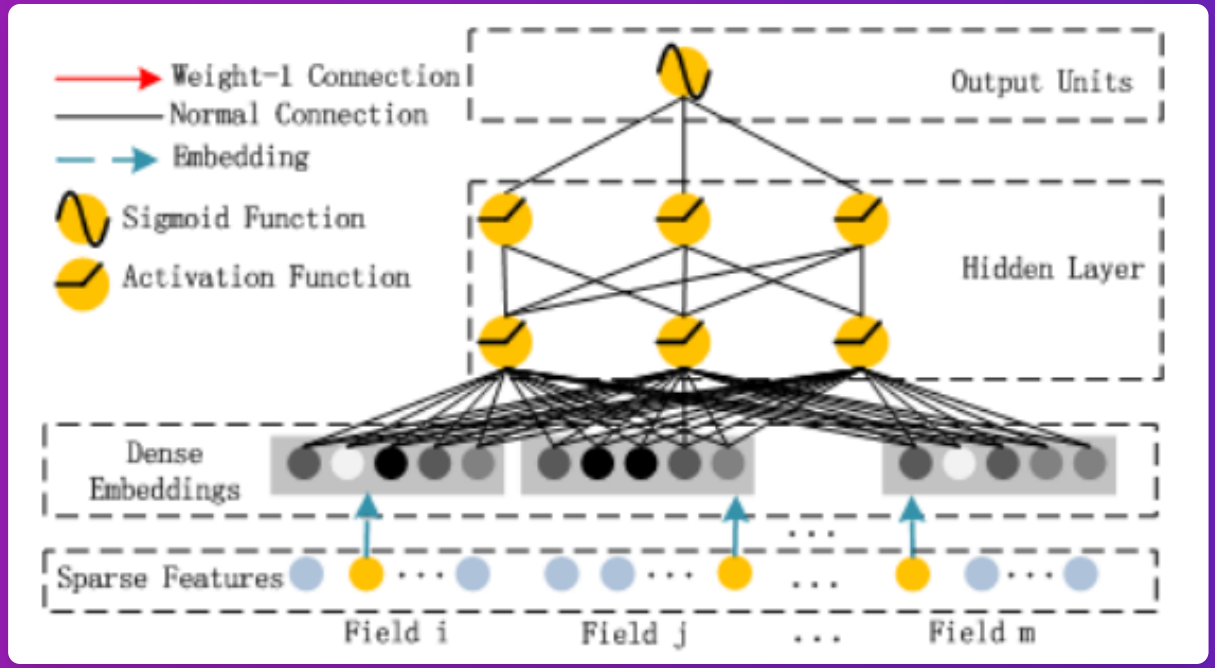
DeepFM——用 FM 代替 Wide 部分

模型结构



DeepFM——用 FM 代替 Wide 部分

模型结构



DeepFM——用 FM 代替 Wide 部分

模型结构

$$y_{FM} = \langle w, x \rangle + \sum_{j_1=1}^{d} \sum_{j_2=j_1+1}^{d} \langle V_i, V_j \rangle x_{j_1} \cdot x_{j_2}$$

输出层：

$$\hat{y} = \text{sigmoid}\left(y_{FM} + y_{DNN}\right)$$

嵌入层的输出：

$$a^{(0)} = [e_1, e_2, \ldots, e_m]$$

隐藏层：

$$a^{(l+1)} = \sigma\left(W^{(l)} a^{(l)} + b^{(l)}\right)$$

DeepFM——用 FM 代替 Wide 部分

对 DeepFM 模型的评价

与 Wide&Deep 模型相比，DeepFM 模型的改进主要是针对 Wide&Deep 模型 的 Wide 部分不具备自动的特征组合能力的缺陷进行的。这里的改进动机与 Deep&Cross 模型的完全一致，唯一的不同就在于 Deep&Cross 模型利用多层 Cross 网络进行特征组合，而 DeepFM 模型利用 FM 进行特征组合。当然，具体的应用效果还需要通过实验进行比较。

NFM——FM 的神经网络化尝试

模型结构

2017 年，新加坡国立大学的研究人员提出了 NFM 模型。

$$\hat{y}_{NFM}(\mathbf{x}) = w_0 + \sum_{i=1}^{n} w_i x_i + f(\mathbf{x})$$

第三项 $f(\mathbf{x})$ 是 NFM 的核心组件，用于建模特征交互，这是一个多层前馈神经网络。

$$f_{BI}(\mathcal{V}_x) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} x_i \mathbf{v}_i \odot x_j \mathbf{v}_j$$

$$\begin{gathered} \mathbf{z}_1 = \sigma_1\left(\mathbf{W}_1 f_{BI}(\mathcal{V}_x) + \mathbf{b}_1\right), \\ \mathbf{z}_2 = \sigma_2\left(\mathbf{W}_2 \mathbf{z}_1 + \mathbf{b}_2\right) \\ \cdots\cdots \\ \mathbf{z}_L = \sigma_L\left(\mathbf{W}_L \mathbf{z}_{L-1} + \mathbf{b}_L\right) \end{gathered}$$

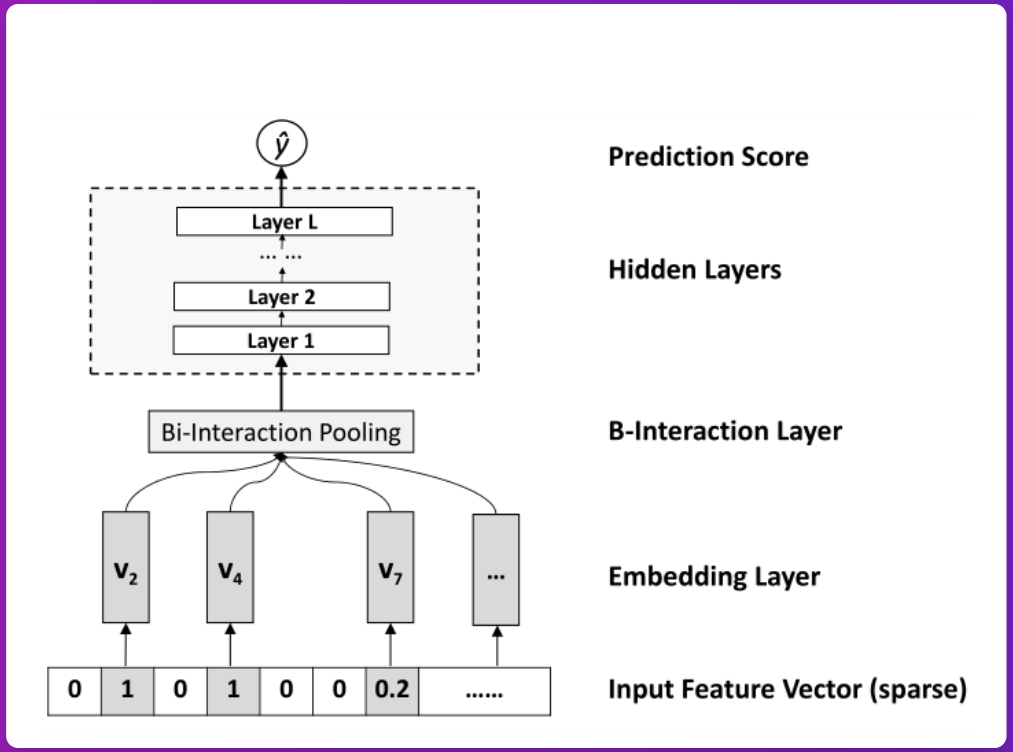
NFM——FM 的神经网络化尝试

模型结构

设 Input Feature Vector 维度为 $N \times 1$，嵌入矩阵维度为 $k \times N$，每一列对应 N 的一个位置上的嵌入向量，对于一个输入实例来说，如图所示，第 2 ，4，7 个位置上有值，就把嵌入矩阵的第 2，4，7 列向量取出来。

NFM——FM 的神经网络化尝试

模型结构



基于 FM 的深度学习模型的优点和局限性

FNN、DeepFM、NFM 三个结合 FM 思路的深度学习模型的特点都是在经典多层神经网络的基础上加入有针对性的特征交叉操作，让模型具备更强的非线性表达能力。 沿着特征工程自动化的思路，深度学习模型从 PNN 一路走来，经过了 Wide&Deep、Deep&Cross、FNN、DeepFM、NFM 等模型，进行了大量的、基于不同特征互操作思路的尝试。但特征工程的思路走到这里几乎已经穷尽了可能的尝试，模型进一步提升的空间非常小，这也是这类模型的局限性所在。

从这之后，越来越多的深度学习推荐模型开始探索更多“结构”上的尝试， 诸如注意力机制、序列模型、强化学习等在其他领域大放异彩的模型结构也逐渐进入推荐系统领域，并且在推荐模型的效果提升上成果显著。

注意力机制在推荐模型中的应用

“注意力机制”来源于人类最自然的选择性注意的习惯。最典型的例子是用户在浏览网页时，会选择性地注意页面的特定区域，忽视其他区域。在建模过程中考虑注意力机制对预测结果的影响，往往会取得不错的收益。 近年来，注意力机制广泛应用于深度学习的各个领域，无论是在自然语言处理、语音识别还是计算机视觉领域，注意力模型都取得了巨大的成功。

AFM——引入注意力机制的 FM

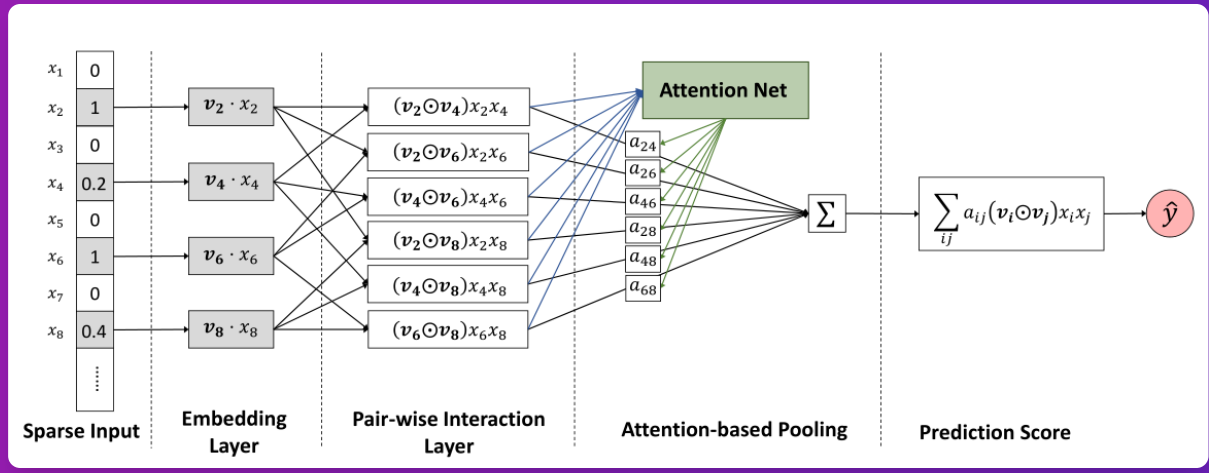
模型结构

浙江大学于2017年提出了 AFM 模型。AFM 模型可以被认为是 NFM 模型的延续。在 NFM 模型中， 不同域的特征 Embedding 向量经过特征交叉池化层的交叉，将各交叉特征向量进 行“加和”，输入最后由多层神经网络组成的输出层。问题的关键在于加和池化（Sum Pooling）操作，它相当于 “一视同仁”地对待所有交叉特征，不考虑不同特征对结果的影响程度，事实上消解了大量有价值的信息。 这里 “注意力机制” 就派上了用场，它基于假设一不同的交叉特征对于结果的影响程度不同，以更直观的业务场景为例，用户对不同交叉特征的关注程度应是不同的。举例来说，如果应用场景是预测一位男性用户是否购买一款键盘的可能性，那么“性别=男且购买历史包含鼠标”这一交叉特征，很可能比“性别= 男且用户年龄=30” 这一交叉特征更重要，模型投入了更多的 “注意力” 在前面的特征上。正因如此，将注意力机制与 NFM 模型结合就显得理所应当了。

具体地说，AFM 模型引入注意力机制是通过在特征交叉层和最终的输出层之间加入注意力网络 Attention Net 实现的。 注意力网络的作用是为每一个交叉特征提供权重，也就是注意力得分。

AFM——引入注意力机制的 FM

模型结构



AFM——引入注意力机制的 FM

模型结构

$$a'_{ij} = \mathbf{h}^T \operatorname{ReLU}\left(\mathbf{W}\left(\mathbf{v}_i \odot \mathbf{v}_j\right) x_i x_j + \mathbf{b}\right)$$

$$a_{ij} = \frac{\exp\left(a'_{ij}\right)}{\sum_{(i,j)\in\mathcal{R}_x} \exp\left(a'_{ij}\right)}$$

$$\hat{y}_{AFM}(\mathbf{x}) = w_0 + \sum_{i=1}^{n} w_i x_i + \mathbf{p}^T \sum_{i=1}^{n} \sum_{j=i+1}^{n} a_{ij}\left(\mathbf{v}_i \odot \mathbf{v}_j\right) x_i x_j$$

AFM——引入注意力机制的 FM

对 AFM 模型的评价

AFM 是研究人员从改进模型结构的角度出发进行的一次有益尝试。它与具体的应用场景无关。

DIN——引入注意力机制的深度学习网络

模型结构

相比于之前很多“学术风”的深度学习模型，阿里巴巴2018年提出的 DIN 模型显然更具业务气息。

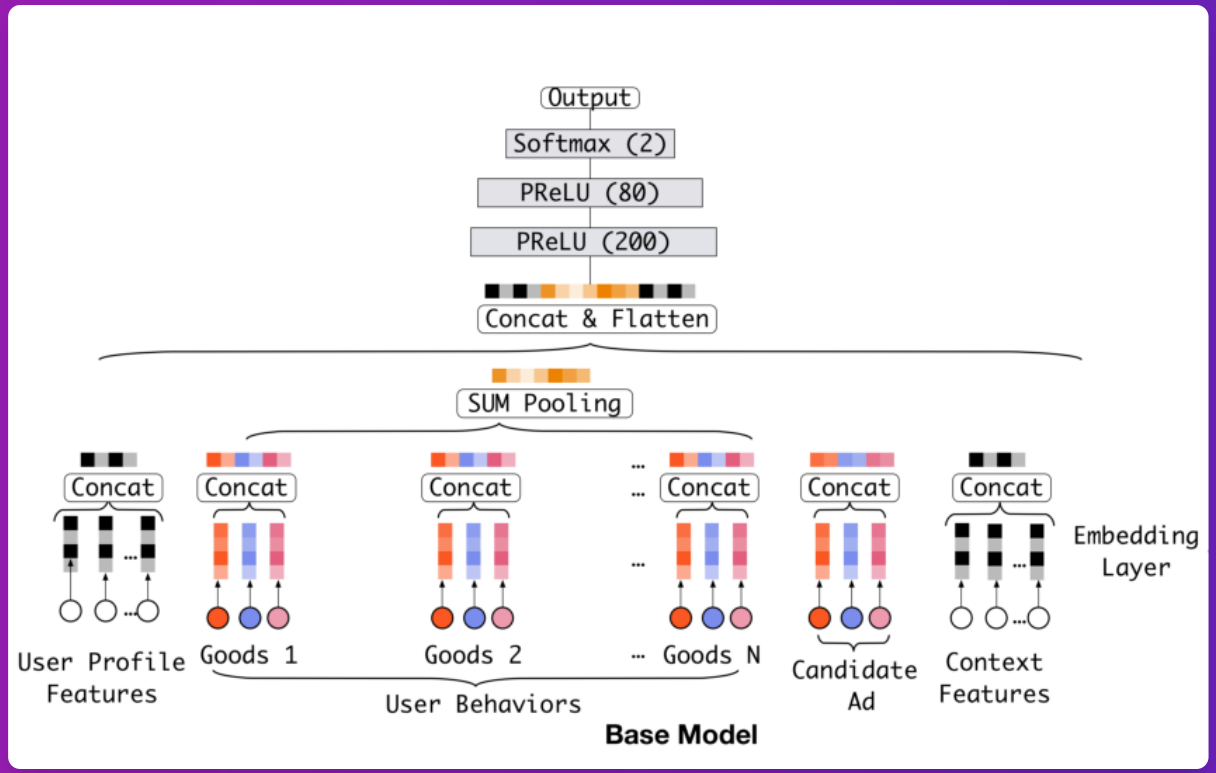
应用场景：

> 是阿里巴巴的电商广告推荐，因此在计算一个用户是否点击一个广告 a 时，模型的输入特征自然分为两大部分：一部分是用户 u 的特征组，另一部分是候选广告 a 的特征组。无论是用户还是广告，都含有两个非常重要的特征——商品 id(good_id)和商铺 id(shop_id)。用户特征里的商品 id 是一个序列， 代表用户曾经点击过的商品集合，商铺 id 同理；而广告特征里的商品 id 和商铺 id 就是广告对应的商品 id 和商铺 id (阿里巴巴平台上的广告大部分是参与推广计划的商品）。

在原来的基础模型中，用户特征组中的商品序列和商铺序列经过简单的平均池化操作后就进入上层神经网络进行下一步训练，序列中的商品既没有区分重要程度，也和广告特征中的商品 id 没有关系。 然而事实上，广告特征和用户特征的关联程度是非常强的。假设广告中的商品是键盘，用户的点击商品序列中有几个不同的商品 id 分别是鼠标、T 恤和洗面奶。从常识出发，“鼠标”这个历史 商品 id 对预测 “键盘”广告的点击率的重要程度应大于后两者。从模型的角度来说，在建模过程中投给不同特征的“注意力”理应有所不同，而且“注意力得分” 的计算理应与广告特征有相关性。

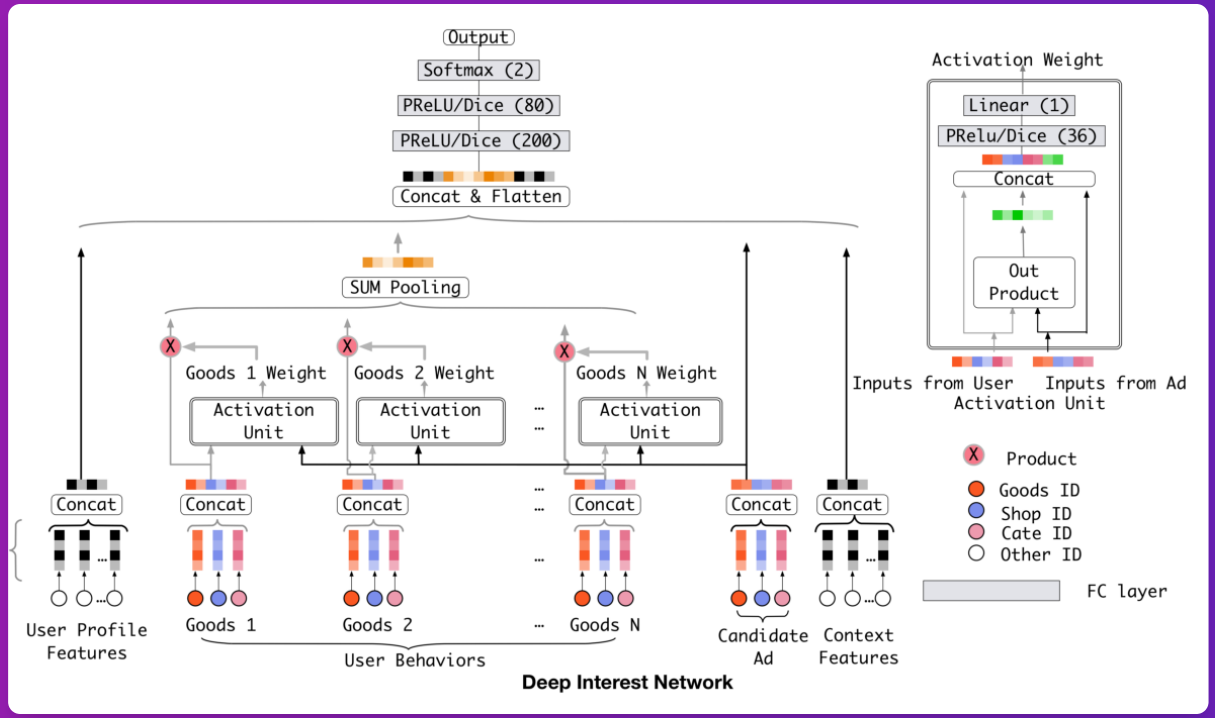
DIN——引入注意力机制的深度学习网络

模型结构



DIN--引入注意力机制的深度学习网络

模型结构



DIN——引入注意力机制的深度学习网络

模型结构

$$\boldsymbol{v}_U(A) = f\left(\boldsymbol{v}_A, \boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_H\right) = \sum_{j=1}^{H} a\left(\boldsymbol{e}_j, \boldsymbol{v}_A\right) \boldsymbol{e}_j = \sum_{j=1}^{H} \boldsymbol{w}_j \boldsymbol{e}_j$$

其中 $\left(\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_H\right)$ 是长度为 H 的用户 U 的行为的嵌入向量列表， $\boldsymbol{v}_A$ 是广告 A 的嵌入向量。这样，$\boldsymbol{v}_U(A)$ 随不同的广告而变化。$a(\cdot)$是一个前馈网络，输出作为激活权重。

除了两个输入嵌入向量外，$a(\cdot)$ 将它们的 out product 添加到后续网络中，这是一种有助于关联建模的显式知识。与传统的注意力方法不同，上式中放宽了 $\Sigma w_i = 1$ 约束，旨在保留用户兴趣的强度。也就是说，放弃了在输出上使用 softmax 进行归一化。相反， w_i 的值在某种程度上被视为激活用户兴趣强度的近似值。例如，如果一个用户的历史行为包含90%的衣服和10%的电子产品。考虑到T恤和手机两个候选广告，T恤激活了大部分属于衣服的历史行为，并且可能比手机获得更大的 $\boldsymbol{v}_U$ 值（更高的兴趣强度）。

DIN——引入注意力机制的深度学习网络

注意力机制对推荐系统的启发

注意力机制在数学形式上只是将过去的平均操作或加和操作换成了加权和或者加权平均操作。这一机制对深度学习推荐系统的启发是重大的。因为“注意力得分”的引入反映了人类天生的“注意力机制”特点。对这一机制的模拟，使得推荐系统更加接近用户真实的思考过程，从而达到提升推荐效果的目的。

从“注意力机制”开始，越来越多对深度学习模型结构的改进是基于对用户行为的深刻观察而得出的。相比学术界更加关注理论上的创新，业界的推荐工程师更需要基于对业务的理解推进推荐模型的演化。

DIEN——序列模型与推荐系统的结合

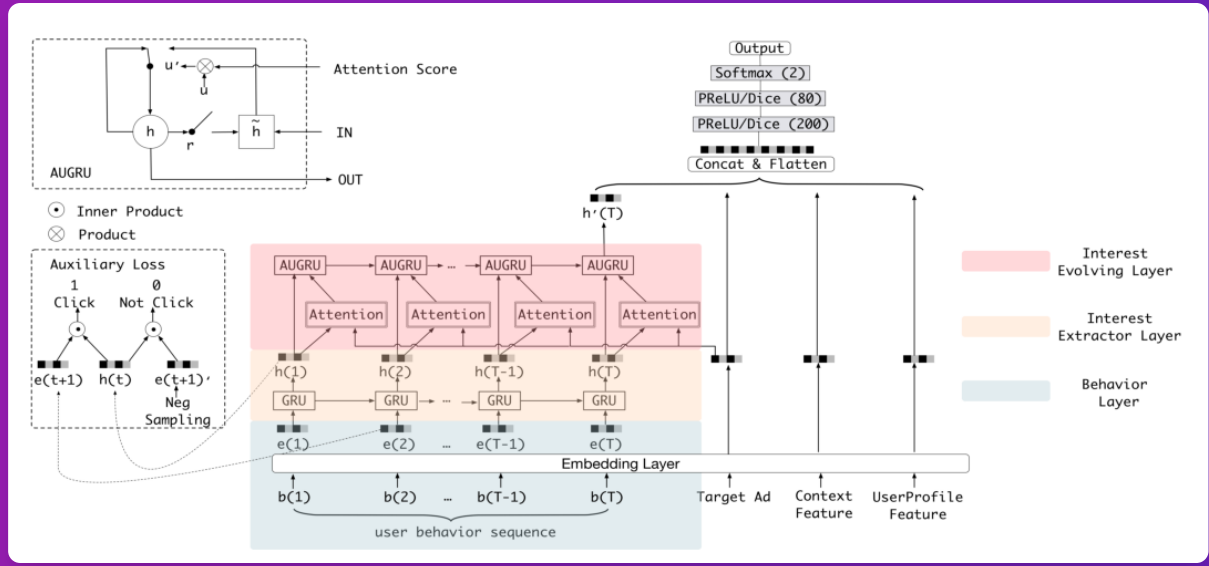
模型结构

阿里巴巴于2019年提出了 DIEN 模型。无论是电商购买行为，还是视频网站的观看行为，或是新闻应用的阅读行为，特定用户的历史行为都是一个随时间排序的序列。既然是时间相关的序列， 就一定存在或深或浅的前后依赖关系，这样的序列信息对于推荐过程无疑是有价值的。但之前介绍的所有模型，都没有利用到这层序列信息。即使是引入了注意力机制的 AFM 或 DIN 模型，也仅是对不同行为的重要性进行打分，这样的得分是时间无关的，是序列无关的。

基于引进“序列”信息的动机，阿里巴巴对 DIN 模型进行了改进，形成了 DIEN 模型的结构。模型仍是输入层+Embedding 层+连接层+多层全连接神经网络+输出层的整体架构。彩色的“兴趣进化网络”被认为是一种用户兴趣的 Embedding 方法，它最终的输出是这个用户兴趣向量。DIEN 模型的创新点在于如何构建“兴趣进化网络”。

DIEN——序列模型与推荐系统的结合

模型结构



DIEN——序列模型与推荐系统的结合

模型结构

兴趣进化网络分为三层，从下至上依次是:

1. 行为序列层（ Behavior Layer，浅绿色部分）：其主要作用是把原始的 id 类行为序列转换成 Embedding 行为序列。
2. 兴趣抽取层（ Interest Extractor Layer，米黄色部分）：其主要作用是通过模拟用户兴趣迁移过程，抽取用户兴趣。
3. 兴趣进化层（ Interest Evolving Layer，浅红色部分）：其主要作用是通过在兴趣抽取层基础上加入注意力机制，模拟与当前目标广告相关的兴趣进化过程。

在兴趣进化网络中，行为序列层的结构与普通的 Embedding 层是一致的，模拟用户兴趣进化的关键在于“兴趣抽取层” 和 “兴趣进化层”。

DIEN——序列模型与推荐系统的结合

模型结构

兴趣抽取层

兴趣抽取层的基本结构是 GRU 网络。 相比传统的序列模型 RNN 和 LSTM ，GRU 解决了 RNN 的梯度消失问题。与 LSTM 相比，GRU 的参数数量更少，训练收敛速度更快，因此成了 DIEN 序列模型的选择。

$$
\begin{aligned}
\boldsymbol{u}_t &= \sigma\left(\boldsymbol{W}^u \boldsymbol{i}_t + \boldsymbol{U}^u \boldsymbol{h}_{t-1} + \boldsymbol{b}^u\right) \\
\boldsymbol{r}_t &= \sigma\left(\boldsymbol{W}^r \boldsymbol{i}_t + \boldsymbol{U}^r \boldsymbol{h}_{t-1} + \boldsymbol{b}^r\right) \\
\widetilde{\boldsymbol{h}_t} &= \tanh\left(\boldsymbol{W}^h \boldsymbol{i}_t + \boldsymbol{r}_t \circ \boldsymbol{U}^h \boldsymbol{h}_{t-1} + \boldsymbol{b}^h\right) \\
\boldsymbol{h}_t &= \left(1 - \boldsymbol{u}_t\right) \circ \boldsymbol{h}_{t-1} + \boldsymbol{u}_t \circ \widetilde{\boldsymbol{h}_t}
\end{aligned}
$$

经过由 GRU 组成的兴趣抽取层后，用户的行为向量被进一步抽象化， 形成了兴趣状态向量。理论上，在兴趣状态向量序列的基础上，GRU 网络已经可以做出下一个兴趣状态向量的预测，但 DIEN 却进一步设置了兴趣进化层， 这是为什么呢?

DIEN——序列模型与推荐系统的结合

模型结构

兴趣进化层

DIEN 兴趣进化层相比兴趣抽取层最大的特点是加入了注意力机制。这一特点与 DIN 一脉相承。从注意力单元的连接方式可以看出，兴趣进化层注意力得分的生成过程与 DIN 完全一致,都是当前状态向量与目标广告向量进行互作用的结果。也就是说，DIEN 在模拟兴趣进化的过程中，需要考虑与目标广告的相关性。 在兴趣抽取层之上再加上兴趣进化层就是为了更有针对性地模拟与目标广告相关的兴趣进化路径。由于阿里巴巴这类综合电商的特点，用户非常有可能同时购买多品类商品，例如在购买“机械键盘”的同时 还在查看“衣服”品类下的商品，那么这时注意力机制就显得格外重要了。当目标广告是某个电子产品时，用户购买“机械键盘”相关的兴趣演化路径显然比购买 “衣服”的演化路径重要，这样的筛选功能兴趣抽取层没有。 兴趣进化层完成注意力机制的引入是通过 AUGRU（GRU with Attentional Update gate,基于注意力更新门的 GRU 结构,AUGRU 在原GRU 的更新门（update gate）的结构上加入了注意力得分。

$$\tilde{\boldsymbol{u}}_t' = a_t \cdot \boldsymbol{u}_t'$$
$$\boldsymbol{h}_t' = \left(1 - \tilde{\boldsymbol{u}}_t'\right) \circ \boldsymbol{h}_{t-1}' + \tilde{\boldsymbol{u}}_t' \circ \widetilde{\boldsymbol{h}}_t'$$

可以看出 AUGRU 在原始的 $\boldsymbol{u}_t'$ 基础上加入了注意力得分 a_t ，注意力得分的生成方式与 DIN 模型中注意力激活单元的基本一致。

DIEN——序列模型与推荐系统的结合

序列模型对推荐系统的启发

由于序列模型具备强大的时间序列的表达能力，使其非常适合预估用户经过一系列行为后的下一次动作。 事实上，不仅阿里巴巴在电商模型上成功运用了序列模型，YouTube、Netflix 等视频流媒体公司也已经成功的在其视频推荐模型中应用了序列模型，用于预测用户的下次观看行为（ next watch)。 但在工程实现上需要注意：序列模型比较高的训练复杂度，以及在线上推断过程中的串行推断，使其在模型服务过程中延迟较大，这无疑增大了其上线的难度，需要在工程上着重优化。

学习收获

1. 通过系统学习深度推荐模型，加深了对深度学习和推荐系统的理解；
2. 理解模型为什么被提出，解决了哪些问题？还存在哪些问题？
3. 教材、博客与原论文、源码相结合，学习其他人解读模型的思路。

Thank you