

推荐系统入门系列(三)-Wide&Deep算法理论与实践

何无涯 何无涯的技术小屋 5月9日

点击蓝字，带你发现更大的世界

不积硅步，无以至千里，不积小流，无以成江海！

—— 荀子

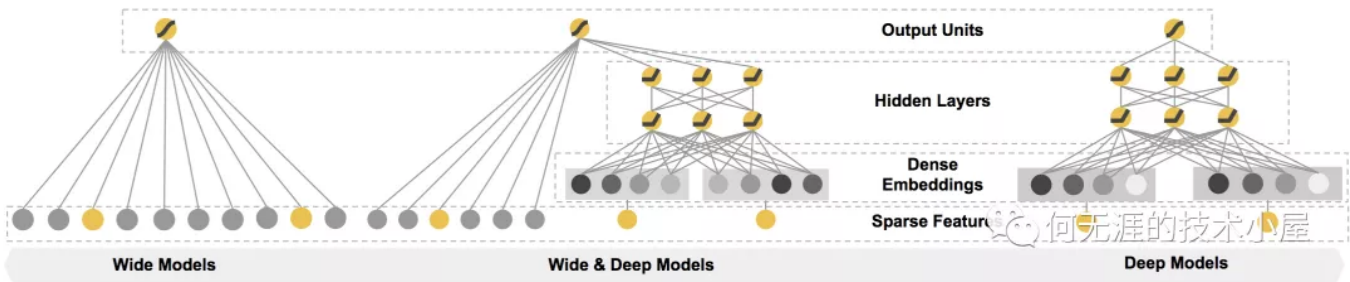


一、Wide&Deep算法背景

在推荐系统中，线性模型（包括逻辑回归）通常使用one-hot编码训练二值化的稀疏特征，模型最简单，并且可解释性强，因此被广泛用在工业环境中的大规模在线推荐和排名系统中；而分解机（FM）或者深度神经网络（DNN），通过为每个特征学习到低维密集嵌入向量，进一步推广到以前从未见到的一些特征组合，然而由于特征的稀疏性，很多特征组合其实是并没有交互，然而低维密集嵌入向量将导致所有特征组合的非零预测，因此可能会产生一些不相关的推荐。

二、Wide&Deep算法思想

Wide&Deep本质上是融合了Wide和Deep的思想，以期望结合二者的优点。线性模型，我们称之为Wide，擅于 memorization，即从历史数据中学习特征的共性和相关性；深度神经网络模型，我们称之为Deep，擅于Generalization，即善于发现历史数据中很少或者没有出现过的新的特征组合，从而提高推荐物品的多样性。模型结构图如下图所示。



Wide部分使用的是广义的线性模型，Deep部分使用的是深度前馈神经网络，将两部分进行联合训练。

三、Wide&Deep算法实战

下面使用PyTorch实现Wide&Deep算法，核心代码如下：

```
1 class WideAndDeepModel(nn.Module):
2     """
3     A pytorch implementation of wide and deep learning.
4     Reference:
5     HT Cheng, et al. Wide & Deep Learning for Recommender Systems, 2016.
```

```
6      """
7      def __init__(self, filed_dims, embed_dim, mlp_dims, dropout):
8          super.__init__()
9          self.linear = FeaturesLinear(filed_dims)
10         self.embedding = FeaturesEmbedding(filed_dims, embed_dim)
11         self.embed_output_dim = len(filed_dims) * embed_dim
12         self.mlp = MultilayerPerception(self.embed_output_dim, mlp_dims, dropout)
13
14     def forward(self, x):
15         """
16         :param x : Long tensor of size ``(batch_size, num_fields)``
17         """
18         embed_x = self.embedding(x)
19         # Wide part + Deep part
20         x = self.linear(x) + self.mlp(embed_x.view(-1, self.embed_output_dim))
21         return torch.sigmoid(x.squeeze(1))
```

详细的代码请参考我的github：
<https://github.com/yyHaker/RecommendationSystem>

小结：Wide&Deep是为了融合Wide线性模型和Deep深度前馈神经网络模型的优点，不过个人认为Wide&Deep模型仍然存在LR的一些问题，捕获组合特征的能力并不是很强。

参考文章：

【1】Cheng, Heng-Tze, et al. "Wide & deep learning for recommender systems." Proceedings of the 1st workshop on deep learning for recommender systems. 2016.



何无涯的技术小屋

微信号码：leyanyuanyu

机器学习 | 深度学习 | 推荐算法 | NLP | 投资