



昵称：山竹小果
园龄：5年1个月
粉丝：188
关注：0

关注成功

| | | | | | | | |
|----|---------|----|----|----|----|----|---|
| < | 2024年1月 | | | | | | > |
| 日 | 一 | 二 | 三 | 四 | 五 | 六 | |
| 31 | 1 | 2 | 3 | 4 | 5 | 6 | |
| 7 | 8 | 9 | 10 | 11 | 12 | 13 | |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 | |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | |
| 28 | 29 | 30 | 31 | 1 | 2 | 3 | |
| 4 | 5 | 6 | 7 | 8 | 9 | 10 | |

◇ 搜索

找找看

◇ 随笔分类

Algorithm(22)

C++(15)

Computer Vision(4)

Deep Learning(48)

Interview(11)

Machine Learning(23)

NLP(76)

Notes(28)

Papers(30)

Python(32)

Reinforcement Learning(4)

论文阅读 | A Survey on Multi-Task Learning

摘要

多任务学习(Multi-Task Learning, MTL)是机器学习中的一种学习范式，其目的是利用包含在多个相关任务中的有用信息来帮助提高所有任务的泛化性能。

首先，我们将不同的MTL算法分为**特征学习法**、**低秩方法**、**任务聚类方法**、**任务关系学习方法和分解方法**，然后讨论每种方法的特点。为了进一步提高学习任务的性能，MTL可以与其他学习范式相结合，包括半监督学习、主动学习、非监督学习、强化学习、多视图学习和图形模型。当任务数量较大或数据维数较高时，批量MTL模型难以处理，本文对在线、并行和分布式MTL模型以及降维和特征哈希进行了综述，揭示了它们在计算和存储方面的优势。许多实际的应用程序使用MTL来提高它们的性能。最后，我们提出了理论分析，并讨论了未来的几个方向。

1 介绍

数据集难以收集时，MTL是个好的解决方案。

在MTL中，有多个学习任务，每个学习任务都可以是一个通用的学习任务，如监督任务(如分类或回归问题)、非监督任务(如聚类问题)。半监督任务，强化学习任务。多视图学习任务或图形模型。在这些学习任务中，**所有这些任务或至少其中一部分被认为是相互关联的**。在这种情况下，我们发现联合学习这些任务可以比单独学习更有效地提高性能。这一观察结果导致了MTL的诞生。因此，当多个任务相关时，MTL旨在提高它们的泛化性能。

MTL的设置与迁移学习[2]相似，但也有显著性差异。在MTL中，不同任务之间没有区别，目标是提高所有任务的性能。但是迁移学习是在在源任务的帮助下提高目标任务的性能，目标任务比源任务起着更重要的作用。

因此，MTL对所有的任务都一视同仁，但在转移学习中，目标任务是所有任务中最受关注的。在[3], [4], [5]中，研究了一种新的MTL设置，称为**非对称多任务学习**，该设置考虑了一个不同的场景，**当多个任务通过某种MTL方法联合学习时，又有了新任务**。一个简单的解决方案是从头开始学习旧的和新的任务，但是这需要大量的计算。而非对称多任务学习仅仅是在旧任务的帮助下学习新任务，因此核心问题是如何将旧任务中包含的知识转移到新任务中。从这个意义上说，这种设置更类似于转移学习而不是MTL。

在这篇论文中，我们对MTL进行了研究。在给出了MTL的定义之后，我们将不同的MTL算法分为几个类别：**特征学习方法**，它可以分为**特征变换和特征选择方法**、**低秩方法**、**任务聚类方法**、**任务重组学习方法和分解方法**。我们讨论每种方法的特征。MTL可以与其他学习范式相结合，进一步提高学习任务的性能，因此我们讨论了MTL与其他学习范式的结合，包括半监督学习、主动学习、非监督学习、强化学习、多视图学习和图形模型。当任务数量较大时，所有任务中的训练数据量都可能非常大，这就需要进行MTL模型的在线并行计算。在这种情况下，不同任务的训练数据可以放在不同的机器上，因此分布式的MTL模型是一个很好的解决方案。在MTL中，维数重构和特征哈希是降低高维数据维数的重要工具。因

◇ 随笔档案

2021年1月(3)

2020年10月(4)

2020年9月(1)

2020年8月(3)

2020年7月(1)

2020年6月(3)

2020年5月(7)

2020年4月(4)

2020年3月(19)

2020年2月(8)

2020年1月(21)

2019年12月(19)

2019年11月(23)

2019年10月(3)

2019年9月(16)

更多

◇ 阅读排行榜

1. 用BERT做语义相似度匹配

任务：计算相似度的方式

(23170)

2. NLP 语义相似度计算 整理

总结(19114)

3. python 输出对齐(14782)

此，我们回顾了多个任务中处理大数据时有用的技术。作为一种通用的学习范式，MTL在各个领域都有广泛的应用，本文简要回顾了它在计算机视觉、自然语言

[首页](#) [新随笔](#) [联系](#) [订阅](#) [管理](#)

息学、健康信息学、语音、自然语言处理、网络应用和普适计算等领域的应用。

2 MTL模型

MTL：假设 m 个学习任务 $T_i(i=1\sim m)$ ，其中所有任务或其中一个子集是相关的，多任务学习的目的是通过使用所有或部分 m 个任务中包含的知识来帮助改进 T_i 模型的学习。

在此，我们需要区分同构MTL和异构MTL。[7],异构MTL被认为是由不同类型的监督任务,包括分类和回归问题,这里我们推广到更一般的设定任务的异构MTL由不同类型包括监督学习、无监督学习,semisupervised学习,强化学习,多视点学习和图形模型。与异构MTL相对的是同构MTL，它只包含一种类型的任务。综上所述，同质类和异质类MTL在学习任务类型上存在差异，而同质类特征MTL与异质类特征MTL在原始特征表示上存在差异。类似地，没有特殊的解释，默认的MTL设置是同类的MTL。

为了描述MTL定义中的关联性，需要解决三个问题：何时共享、共享什么以及如何共享。

“何时共享”问题是针对一个多任务问题在单任务请求和多任务模型之间做出选择。目前，这样的决定是由人类专家做出的，很少有人学习研究它的方法。一个简单的计算解决方案是将这样的决策表示为模型选择问题，然后使用模型选择技术(例如交叉验证)来进行决策，但是这种解决方案通常计算量很大，并且可能需要更多的训练数据。另一种解决方案是使用多任务模型，该模型可以退化为单任务模型，例如problem(34)，当不同任务的学习变为（加和符号）时可以解耦。在这种情况下，我们可以让训练数据来决定（加和符号）的形式，从而做出隐含的选择。

要共享什么内容，需要确定所有任务之间可以通过哪些形式进行知识共享。通常，共享内容有三种形式，包括特性、实例和参数。基于特征的MTL旨在学习不同任务之间的共同特征，从而共享知识。基于实例的MTL希望在任务中为其他任务标识有用的数据实例，然后通过标识的实例共享知识。基于参数的MTL在任务中使用模型参数(例如，线性模型中的系数)，以某种方式帮助学习其他任务中的模型参数，例如正则化。现有的MTL研究主要集中在基于特征和基于参数的方法上，属于基于实例方法的研究较少。代表基于实例的方法是多任务分布匹配方法[8],密度比率首次估计概率之间的每个实例以及它的标签属于自己的任务和混合所有的任务,然后使用加权训练数据的所有任务基于学习模型参数的估计的密度比为每个任务。由于基于实例的MTL研究较少，我们主要综述基于特征和基于参数的MTL模型。

在确定了“共享什么”之后，“如何共享”指定了在任务之间共享知识的具体方法。在基于特征的MTL中，有一种主要的方法:特征学习方法。特征学习方法侧重于学习基于浅层或深层模型的多任务的公共特征表示，其中学习的公共特征表示可以是原始特征表示的子集或转换。基于参数MTL。主要有四种方法:低秩方法、任务聚类方法、任务关系学习方法和分解方法。低秩方法将多个任务之间的关联解释为这些任务的参数矩阵的低秩性。任务集群方法假设所有任务都形成几个集群，其中集群中的任务彼此相关。任务关系学习方法旨在从数据中自动学习任务之间的定量关系。分解方法将所有任务的模型参数分解为两个或多个分量，由不同的正则化器进行惩罚。总之，基于特征和基于参数的MTL主要有五种方法。在接下来

数 (附python实现)(14209)

5. C++ | priority_queue的用法

(含自定义排序方式) (13

785)

◇ 评论排行榜

1. 百度架构师手把手教深度学习 | 一些心得体会(6)

2. Notes | 学术前沿趋势分析

(2)

3. 论文阅读 | Adversarial Train

ing for Large Neural Langu

age Models(2)

4. Feature map transfer in Tin

yBert/MobileBert(2)

5. Python | numpy | np.split()

与np.array_split()函数(2)

◇ 推荐排行榜

1. Pearson、Spearman秩相关

系数、kendall等级相关系

数 (附python实现)(6)

2. C++ | priority_queue的用法

(含自定义排序方式) (5)

3. 用BERT做语义相似度匹配

任务：计算相似度的方式

(5)

4. 特征分解(5)

特征转移方法

输入所有样本 经过Hidden layers → 输出该样本属于哪个任务。

这里隐藏层的输出可以被视为m的共同特征表示学习任务 and 转换从原始表示学到一个取决于连接输入和隐藏层的权重以及激活函数采用隐藏的单位。因此，如果隐层中的激活函数是线性的，那么变换就是线性的，反之就是非线性的。与用于单任务学习的多层前馈神经网络相比，网络结构的不同之处在于输出层，在单任务学习中只有一个输出单元，而在MTL中有m个输出单元。在[9]中，只有一个隐含层的径向基函数网络通过贪婪地确定隐含层的结构扩展到MTL。与这些神经网络模型不同的是，Silver等人提出了一个上下文敏感的多任务神经网络，它只有一个由不同任务共享的输出单元，但有一个特定于任务的上下文作为额外的输入。

与连接主义模型的多层前馈神经网络不同，多任务特征学习(MTFL)方法在正则化框架下，以目标函数为

分类: [Deep Learning](#)

关注成功

山竹小果
粉丝 - 188 关注 - 0

推荐 1

反对 0

支持成功 [撤回](#)[升级成为会员](#)« 上一篇: [论文阅读 | ERNIE: Enhanced Representation through Knowledge Integration](#)» 下一篇: [面试总结 | 百度 NLP 实习生](#)

posted @ 2019-11-10 18:26 山竹小果 阅读(1822) 评论(0) 编辑 收藏 举报

发表评论

[升级成为园子VIP会员](#)[B](#) [🔗](#) [</>](#) [“](#) [🖼](#)

支持 Markdown

提交评论

[退出](#) [订阅评论](#) [我的博客](#)[🔗 自动补全](#)

【推荐】通义灵码，灵动指间，快码加编，你的智能编码助手

【推荐】编程路上的催化剂：大道至简，给所有人看的编程书

【推荐】会员力量，点亮园子希望，期待您升级成为园子会员

【推荐】阿里云云市场联合博客园推出开发者商店，欢迎关注



编辑推荐:

- 深度解析 C# 数组对象池 ArrayPool< T > 底层原理
- C# AsyncLocal 是如何实现 Thread 间传值
- 一个例子形象地理解同步与异步
- C# 线程本地存储 为什么线程间值不一样
- [动画进阶] 神奇的 3D 卡片反光闪烁动效

阅读排行:

- 全球 IPv4 耗尽, 下个月开始收费!
- Nginx被它打败了?
- C# AsyncLocal 是如何实现 Thread 间传值
- 《HelloGitHub》第 94 期
- 纯c#运行开源本地大模型Mixtral-8x7B