**基于多任务深度学习的自适应节点网络**

**摘要**

多任务学习通过享受任务之间的相关信息来提升各个任务之间的性能。现阶段主流的深度多任务学习模型大多数基于硬参数共享机制，它可以降低模型过拟合的风险；但在联合训练多个任务时,可能会产生知识负迁移现象,阻碍了各个任务性能的提升。基于此，我们提出了自适应节点的深度多任务学习方法。在硬参数共享网络架构的基础上，通过参数关于损失函数的关系，设置连续梯度异号阈值以及预热训练次数阈值，动态更新网络中的节点数量；使得各个任务在充分享受共享信息后，利用自适应节点进一步优化每一个任务，减少了负迁移现象带来的影响。模拟和实证都验证了该方法的有效性，其性能优于主流的硬参数共享多深度任务学习方法。

关键词:多任务学习；连续梯度异号阈值；预热训练次数阈值；共享信息;自适应节点

# Introduction

随着物联网和大数据技术的在环境领域的深入应用，累积大量空气质量监测数据，基于人工智能的机器学习模型不断被提出，其中以神经网络为主的网络模型在空气质量预测中得到了广泛的应用(Gao et al., 2018; Nieto et al., 2018; Park et al., 2018), 尤其是基于深度学习的预测模型(LeCun et al., 2015; Schmidhuber et al., 2015)可以免除传统机器学习的特征工程过程，更好的揭示空气质量复杂非线性机理，成为当前空气质量预测研究的热点。目前空气治理的主要任务是强化多污染物协同控制和区域协同治理，加强细颗粒物和臭氧（O3）协同控制。

作为机器学习中一个很有前途的领域，多任务学习（MTL）旨在通过利用它们之间的有用信息来提高多个相关学习任务的性能Zhang and Yang (2018)。在深度学习时代，MTL 转化为设计能够从多任务监督信号中学习共享表示的网络, 如果相关任务共享互补信息或充当彼此的正则化器，它们就有可能提高性能(Vandenhende et al.,,2020);对于给定的模型，利用多个任务之间的相关性，对多个任务进行联合训练，来更好地概况我们的原始任务，通过共享表示从而提高模型在多个任务上的学习能力。基于深度学习的多任务学习最早可追溯到1997年，Caruana, (1993)提出了神经网络中硬参数共享的结构，同时基于实证研究提出了多任务学习起作用的几个可能的机制。与多任务学习相结合的深度学习框架主要有下面两种方法：（1）基于硬约束的多任务深度学习方法。硬共享是目前应用最为广泛的共享机制，硬参数共享大大降低了过度拟合的风险Ruder S，(2017);它把多个任务的数据表示嵌入到同一个语义空间中，再为每个任务使用任务特定层提取任务特定表示。大多数硬参数共享网络通过共享隐层来学习特征的共享表示，在输出端为各任务设计单独的分类器预测输出层,例如Collobert and Weston（2008）将词性标注、词块分割、命名实体识别及词语相似度任务统一到一个语言模型中, 利用其他任务中自动学习的特征 来 提 升 语 义 角 色 标 注 任 务 的 性 能。Collobert and Weston (2008); Subramanian et al. (2018); Liu et al. (2019)使用硬共享方法将特定于任务的层堆叠在共享层的顶部。（2）基于软约束的多任务深度学习方法。参数的软共享机制，可以不用考虑任务的相关性，不需要对任务相关性做任何假设，但由于为每个任务分配一个网络，常常需要增加很多参数，每个任务都有自己的模型和参数。然后对模型参数之间的距离进行正则化，以鼓励参数相似，例如Long et al., (2015)使用L2距离进行正则化，而Yang et al., (2017)使用轨迹范数,同时MTL正则化约束也给了软参数共享很大的启发。

深度学习已成为许多领域的主要方法，这种方法很强大，但它很容易受到噪声和异常任务的影响，这些任务会显着降低性能。我们认为设计灵活且健壮的深度多任务模型是可取的。Zhang and Yang (2021).目前，在深度多任务学习中，还存在一定的不足，在联合训练多个任务时, 提高模型在一项任务上的性能会损害具有不同需求的任务的性能，这种现象称为负迁移或破坏性干扰(Crawshaw,2020)。最小化负迁移是 MTL 方法的一个关键目标。基于此，我们从多任务的联合损失出发，提出一种自适应节点的多任务学习方法，其创新点是:在多个任务联合训练达到一定的条件时，将符合条件的节点复制，将多任务训练与单任务训练相结合，在充分享受了多任务联合训练收益的同时，动态地更新网络结构，进一步优化每一个任务，极大地减少了负迁移所带来的影响，使得整体预测效果更好，代价仅仅是少量的时间。

本文结构如下：第 2 章将具体介绍本文的新模型框架，理论价值以及其具体的算法。第3章是模拟, 实例在第 4 章, 第 5 章是理论证明, 最后第 6 章是总结。

# 2. Models and Algorithms

## 2.1 自适应节点网络架构

本文在硬参数共享网络结构的基础上进行改进，研究了一种自适应节点的多任务深度学习方法，主旨思想是将神经网络最后一个共享层转化为自适应节点层，在自适应节点层中可以实现节点的复制以达到更好的预测效果。硬参数共享网络架构如图1所示，其网络架构主要包括3个部分: ①输入层，②共享层，③输出层;自适应节点网络架构如图2所示:网络结构一共包括了4层，分别为输入层，共享层，自适应节点层以及输出层。

图示, 示意图

描述已自动生成

图1 硬参数共享网络架构

图示, 示意图

描述已自动生成

## 图2 自适应节点网络架构

## 2.2 自适应节点算法

假设空间足够大，可以同时包含多个任务的解；假设有k个任务，则输出层共有k个节点，分别记为，共享层i(i=1,…,n-1)一共有j(i)个节点，记为，第n层为自适应节点层，在复制节点前一共有j(n)个节点，记，记共享层i-1到共享层i的权重矩阵为，其中,（p=1,…,j(i)）,偏置矩阵为;记共享层n-1到自适应节点层的权重矩阵为。

为了简单起见，我们以两层共享层，两个任务的硬参数共享网络作为改进模型为例，来阐述我们的思想，对于两个任务的整体损失函数为：



图示, 示意图

描述已自动生成

图3 两个任务两个共享层的硬参数共享网络架构

图示, 示意图

描述已自动生成

## 图4 两个任务两个共享层的自适应节点网络架构

图3和图4是两个任务，两层共享层的自适应节点网络架构和硬参数共享网络架构。

在反向传播过程中，根据前文定义，我们将最后一层共享层作为自适应节点网络的自适应节点层，该层中节点（p=1,…,j(1)）所连接的上一层的权重向量为，对于总体损失函数的梯度如下:

，

当，，，时

本文用Adam算法更新参数(Kingma and Ba，2015),更新参数







以此类推，其中代表学习率，t表示更新的步长，表示过往梯度与当前梯度的均值, 表示过往梯度的平滑与当前梯度平方的均值,代表经过矫正后过往梯度与当前梯度的均值，代表经过矫正后过往梯度与当前梯度平方的均值，是为了维持数值稳定性而添加的常数，和代表的是指数衰减率。

根据Adam算法，此时对于任务 ，由于需要优化的梯度方向与整体多任务需要优化的梯度方向相同，损失函数减小；对于任务，由于需要优化的梯度方向与整体多任务需要优化的梯度方向相反，损失函数增加；对于整体多任务，整体多任务损失函数减少。

反之，当时，同理，对于任务，损失函数增加；对于任务，损失函数减少；整体损失函数减少。其它相似情况同理。针对这种情况我们定义为梯度异号，梯度异号是指自适应节点层中某节点，与上层共享层中连接的权重，在反向传播过程中，在选用值域包含于 或 的激活函数，对于有sign()，sign()，…，sign()不全相等，这种情况称之为梯度异号。

Remark：对于值域包含于 或 的激活函数，则有（i=1，…,k），这将在第五章中给出证明。

本文中首先介绍单任务节点和多任务节点的概念，单任务节点是指该自适应节点层中仅仅只与输出层的单个节点相连的节点，多任务节点是指该自适应节点层中与输出层中多个节点相连的节点。

为了减少多任务学习中负迁移现象带来的影响，我们将节点 复制，新节点为,我们令的初值等于， 原本为多任务节点，它与输出层的连接权重为和，经过复制后，由多任务节点转为单任务节点，它与输出层的连接权重为， 也是单任务节点，与输出层的连接权重为，对于新节点连接的相关参数其更新方式仍然采用adam优化算法。

此时，假设自适应节点网络的整体损失函数，对于任务，的损失函数对的梯度为大于原模型中整体多任务损失函数对梯度，则损失函数更大幅度减小;对于任务， 根据Adam优化算法，损失函数减小。则整体损失函数在新一轮迭代中将会比原模型更大幅度减小。该部分相关证明在第五章给出。

图表, 折线图

描述已自动生成

图5 自适应节点网络的损失函数关于参数的变化情况图

图中两条曲线的最低点分别表示两个任务损失函数的极小值点，绿色虚线代表两个任务迭代的最初值，红色虚线代表开始进行节点复制后损失函数关于参数的变化情况。

算法Algorithm1详细描述了构建自适应节点网络架构的过程，设置了梯度异号阈值 ,预热迭代次数阈值

。

Algorithm1 自适应节点网络学习算法

Require:连续梯度异号阈值  ；预热迭代次数阈值 ；loss function (L1,L2)，自适应节点层H= ,自适应节点计数向量V= =

1：For t = 1…T do

2： For i = 1…j(n) do

3： Calculate gradient  and 

4： If ,

let ,Otherwise  = 0

end

5： If  and  ,

复制节点 and restart for，return to step 1.

end

其中设置预热训练次数阈值是为了防止模型过早的进行节点复制，使得各任务之间可以充分利用相互有用的信息进行学习，发挥多任务学习的优势；设置连续梯度异号阈值是为了防止模型参数为局部最优解，因为当连续梯度异号情况发生时，模型参数关于损失函数长期单调不变，若异号次数不小于阈值并且模型预热训练次数满足条件时进行节点复制，迭代终点为全局最优解的概率更大。

自适应节点层所有节点均可复制，本章主要针对两个任务的情况进行阐述，对于k(k>2)个任务依旧适用，在进行节点复制时，应当根据具体实际情况或需求来给定梯度异号复制节点的条件以及方式，每个自适应层的节点至少连接一个任务，最多可以增加至j(n)\*k个。

## 2.3 模型损失函数

在本文的多任务深度神经网络模型回归问题中，采用均方误差(root mean square error,MSE) 作为评价指标，其计算公式为:



# 3.Simulation

为了简单起见，本节采用如下模型：



情形1:输入的为服从最大值为8，最小值为0的均匀分布的n\*5维随机数，为服从最大值为4，最小值为0的均匀分布的5\*4维随机数， 为服从最大值为4，最小值为0的均匀分布的4\*2维随机数，为服从标准正态分布的4\*1维随机数，为服从标准正态分布的2\*1维随机数，噪音为服从均值为0，方差为1的正态分布的n\*2维随机数，为n\*2维矩阵。

情形2:输入的为服从标准正态分布的n\*5维随机数，为服从标准正态分布的5\*4维随机数，为服从标准正态分布的4\*2维随机数，为服从标准正态分布的4\*1维随机数，为服从标准正态分布的2\*1维随机数，噪音为服从均值为0，方差为0.2的正态分布的n\*2维随机数，输出为n\*2维矩阵。

情形3:输入的为服从参数为10的指数分布的n\*5维随机数，为服从标准正态分布的5\*4维随机数，为服从标准正态分布的4\*2维随机数，为服从标准正态分布的4\*1维随机数，为服从标准正态分布的2\*1维随机数，噪音为服从标准正态分布的n\*2维随机数，输出为n\*2维矩阵。

本文构建的算法程序包括: CPU为r5-5600H，GPU为RTX2060，操作系统为 64 位 windows，开发语言为Python;所有方法的实现均基于**pytorch**(Gonçalves等,2016)框架实现的。设置输入层的神经元数为5，共享层的神经元数为4，最初的自适应节点层神经元数为3，输出层的神经元数为2。Adam 优化算法学习率=0.001，指数衰减率=0.9 ，=0.9999 。

对于情形1，情形2，情形3，经过调参可以得到最优的超参数如下:

表1 模拟数据的超参数表

|  |  |  |  |
| --- | --- | --- | --- |
| 超参数 | 均匀分布 | 标准正态分布 | 指数分布 |
|  | 6 | 6 | 6 |
|  | 6 | 7 | 5 |
|  | 4 | 8 | 8 |
|  | 50 | 50 | 50 |
|  | 20 | 20 | 25 |
|  | 10 | 20 | 15 |

其中每个超参数的含义如下:

-硬参数共享网络进行第一次复制节点的连续梯度异号阈值。

-复制一次节点的自适应节点网络进行第二次复制节点的连续梯度异号阈值。

-复制二次节点的自适应节点网络进行第三次复制节点的连续梯度异号阈值。

-硬参数共享网络进行第一次复制节点的预热训练次数阈值

-复制一次节点的自适应节点网络进行第二次复制节点的预热训练次数阈值。

-复制二次节点的自适应节点网络进行第三次复制节点的预热训练次数阈值。

对于上述情形分别设置与硬参数共享网络的对比实验，为保证对比的有效性，原网络的总迭代次数与新网络复制节点前和每次复制节点后的累加总迭代次数相等。

对于上面三种模拟情形，分别取样本量n为1000，2000和3000。整体数据集按7:3的比例分别随机划分为训练集和测试集，将训练集中一部分数据作为验证集，用于选择出最优的模型。测试集用于评估模型性能和稳健性，并采用100轮结果的均值作为实验结果。

本文采用均方误差(mean square error,MSE) 作为评价指标，

性能提升值计算公式如下:



表2 两个网络100次模拟MSE的均值()及性能提升值(%)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | n=1000 | | | n=2000 | | | n=3000 | | |
|  | 原网络 | 新网络 | 提升值 | 原网络 | 新网络 | 提升值 | 原网络 | 新网络 | 提升值 |
| 均匀分布 | 0.314 | 0.224 | 28.7 | 0.284 | 0.184 | 35.2 | 0.258 | 0.142 | 44.9 |
| 标准正态分布 | 1.604 | 1.382 | 13.8 | 1.230 | 1.038 | 15.6 | 0.894 | 0.764 | 14.5 |
| 指数分布 | 0.627 | 0.483 | 22.9 | 0.554 | 0.436 | 21.3 | 0.526 | 0.418 | 20.5 |

表2分别为情形1，情形2，情形3在自适应节点网络和硬参数共享网络上的对比实验结果。随着样本量的增大，两个网络的MSE 值越来越小。从提升值可以看出，新网络相比于原网络提升值都超过13%。

表3 两个网络1000个样本100次模拟的平均耗时(s)

|  |  |  |
| --- | --- | --- |
|  | 原网络 | 新网络 |
| 均匀分布 | 85 | 92 |
| 标准正态分布 | 97 | 102 |
| 指数分布 | 89 | 93 |

表3表示模拟数据的耗时表，在耗时方面，就1000个样本而言，自适应节点网络在3个模拟数据上的平均每轮模拟耗时与硬参数共享网络相比在7s之内，不难看出，自适应节点网络的整体性能优于硬参数共享网络，说明了新网络相较于原网络具有良好的预测和估计能力，代价仅仅是少量的时间。

下面是样本量为1000，模拟数据分别为均匀分布、标准正态分布和指数分布的自适应节点网络与硬参数共享网络的MSE变化对比图，由于在复制节点之前新网络和原网络的MSE的十分接近，在模拟中不展示复制节点前两者MSE的变化过程，从自适应节点网络第一次进行节点复制开始展示。其中图6-8中的前三个标记点依次代表自适应节点网络第一次、第二次和第三次节点复制。最后两个标记点代表两个网络MSE的最低点。由图6-8中可以看出在迭代过程中新网络的MSE一直低于原网络，说明了自适应节点网络良好的性能。

图6 输入数据为均分分布的新网络与原网络的MSE变化对比图

图7 输入数据为标准正态分布的新网络与原网络的MSE变化对比图

图8 输入数据为指数分布的新网络与原网络的MSE变化对比图

# 4. Example

本节考虑南京市空气污染数据，该数据来源于中国气象数据网和真气环境大数据中心。数据包含了南京市2013年12月2日至2022年8月28日每一天的PM2.5浓度()、PM10浓度()、CO浓度()、SO2浓度()、NO2浓度()、臭氧8小时O3\_8h()这个7变量，其中PM2.5浓度和臭氧8小时 O3\_8h()分别为输出变量和。PM10浓度、CO浓度、SO2浓度和NO2浓度分别为输入变量,,和，样本量n为3191个。具体考虑输入变量,,和来估计未来两个任务和的输出值。

空气污染预测模型： 

其中输入是3191\*4维矩阵，为4\*4维矩阵， 为4\*3维矩阵，为3\*2维矩阵，输出是3191\*2维矩阵，为4\*1维矩阵，为3\*1维矩阵，为2\*1维矩阵。

对于此预测模型，经过调参可以得到最优的超参数如下:

表4 实例的超参数

|  |  |
| --- | --- |
| 超参数 | 数值 |
|  | 15 |
|  | 8 |
|  | 7 |
|  | 70 |
|  | 20 |
|  | 20 |

将南京市的空气污染数据分别放入自适应节点网络和硬参数共享网络，且估计效率的衡量仍采用MSE。由表5可知，自适应节点网络比硬参数共享网络的效果提升了9.1个百分点。

表6是南京市的空气污染数据放入不同模型的耗时表，自适应节点网络的时间为321s，硬参数共享网络的时间为310s，新网络仅仅比原网络多了11s。

表5 两个网络的MSE值(10-2) 、耗时(s)及性能提升值(%)

|  |  |  |  |
| --- | --- | --- | --- |
|  | MSE | 时间 | 提升值 |
| 原网络 | 1.3829 | 321s | 9.1 |
| 新网络 | 1.2584 | 310s |

图9 南京市空气污染数据新模型与原模型的MSE变化对比图

图9是南京市空气污染数据自适应节点网络与硬参数共享网络的MSE变化对比图，图9中的标记点模拟数据时的标记点含义一样，不同的是，展示了在第一次复制节点前两个模型的MSE变化趋势，实质上是大致相同。从图9中可以看出对于南京市空气污染数据，在迭代过程中新网络的MSE几乎一直低于原网络，证实了新模型的的良好性能。

# 5. Proof

## 5.1梯度同号证明

在硬参数共享网络中，各层的激活函数选择sigmoid函数。试证明任意隐藏层中任意节点所连接的上一层各个权重，在反向传播更新参数时对于各个任务损失函数的梯度方向相同。

设总层数为n+1的硬参数共享网络中，共k个任务， 表示第i层中j(i)个节点的输出向量, 表示第i-1层中连接第i层第P个节点的权重向量。

在网络正向传播过程中



损失函数：



由于sigmoid函数值域为（0，1），故对于任意和有，

在反向传播过程中，

根据分母布局链式求导法则，第d层第p个节点 所连接的上层权重为（j(d-1)为第d-1层网络的节点数）对于任务i损失函数Li的梯度有：



不妨令



因此,，得证。

Remark：该证明仅仅只针对sigmoid激活函数，对于值域包含于 或 的激活函数同样适用，如Relu、Softplus等。

## 5.2自适应节点网络拥有更低的损失函数

本文以两个任务为例，其中第n+1层为输出层，第n层为自适应节点层，假设损失函数关于参数在一定范围内单调不变。

对于硬参数共享网络整体损失函数。

表示第n-1层连接到第n层的权重矩阵，其中表示第n-1层中连接第n层中第i个节点的权重向量，第n-1层的节点数为，自适应节点层的节点向量记为 。

记整体两个任务的损失函数对n-1层中连接自适应节点层中第i（其中）个节点的参数向量的梯度为

 (1)

记任务的损失函数对参数向量的梯度为

 (2)

记任务的损失函数对参数的梯度为

 (3)

当(1)＞0,(2)＞0,(3)＜0时.

更新关于整体两个任务的参数

 (4)

因(1)＞0，向负梯度方向迭代，值减少，根据梯度下降算法原理整体损失函数L下降。

对于任务，因(2)＞0，该任务损失函数L1对参数的更新方向与整体损失函数L对参数的更新方向相同，故L1下降。

对于任务y2，因(3)＜0，该任务损失函数L2对参数的更新方向与整体损失函数L对参数的更新方向相反，故L2上升。

因，不妨设在此轮迭代中,下降了个单位，上升了个单位，则下降了个单位。

经过复制节点，新节点为仅与任务y2相关的单任务节点，原节点为只与任务y1相关的单任务节点。且使得==，对于自适应节点网络的整体损失函数。

记为仅与任务相关的参数向量，的损失函数对的梯度为

 (5)

记为仅与任务相关的参数向量，的损失函数对的梯度为

 (6)

更新关于任务的参数

 (7)

原网络中损失函数对参数向量的梯度为



将(1)与(5)对比,其中



对于（2）中



故(1)中的第一项与(5)的值相等,又因（3）<0，故,下降（）个单位。

更新关于任务的参数

 (8)

因，故的值增加，根据梯度下降算法原理，下降。

不妨设下降了个单位。

因，故下降了 个单位。

故下降的单位多于下降的单位。

当(1)＞0,(2)＜0,(3)＞0;(1)＜0,(2)＜0,(3)＞0;(1)＜0,(2)＞0,(3)＜0。的时候同理可证得下降的单位多于下降的单位。

Remark:此结论使用的优化算法也可以是其它优化算法（例如:adam算法等），不仅仅局限于梯度下降法。

# 6.结语

本文基于多任务深度学习方法，结合硬参数共享网络，给出一种自适应节点的多任务学习方法，设置连续梯度异号阈值和预热训练次数阈值，结合bp算法和adam优化算法，动态地更新网络结构，进一步减少了多任务学习中的负迁移现象。数值模拟的结果显示，本文给出的新的自适应节点网络比硬参数共享网络拥有更好的估计表现。然后通过一个实际例子验证了自适应节点网络架构的有效性和可行性，代价仅仅是少量的时间。最后从理论上证明了梯度同号现象和自适应节点网络拥有更好的估计效果。此外，本文只实现了两个任务的自适应节点网络的情形，在未来的研究中可将此多任务学习方法实现到相似的优化算法和更多任务的网络架构中。

# 7.References

M.Gao,L.Yin,J.Ning,Artifificial neural network model for ozone concentration estimation and Monte Carlo analysis,Atmospheric Environment 184(2018) 129-139.

P.J.G.Nieto,G.-G.E,A.B.S，anchez,A.A.R.Miranda,Air quality modeling using the PSO SVM-based approach,MLP neural network,and M5 model tree in the metropolitan area of Oviedo(Northern Spain),Environmental Modeling and Assessment 23(3)(2018)229-247.

S.Park,M.Kim,M.KIM,H.-G.Namgung,K-T.Kim,K.H.Cho,S.-B.Kwon,Predicting PM10 concentration in Seoul metropolitan subway stations using artifificial neural network(ANN),Journal of Hazardous Materials 341(2018)75-82.

Y.LeCun,Y.Bengio,G.Hinton,Deeplearning,Nature 521(7553)(2015)436-444.

J.Schmidhuber,Deep learning in neural networks:an overview,Neural Networks 61(2015)85-117.

Caruana R A. Multitask Learning: A Knowledge-Based Source of Inductive Bias[J]. Machine Learning Proceedings, 1993, 10(1):41-48.

Long D, Cohn T, Bird S, et al. Low Resource Dependency Parsing: Cross-lingual Parameter Sharing in a Neural Network Parser[C]// Meeting of the Association for Computational Linguistics and the, International Joint Conference on Natural Language Processing. 2015:845-850.

Yang, Y. et. al. 2017. Trace Norm Regularized Deep Multi-Task Learning. ICLR2017 workshop.

Collobert R,Weston J.A unified architecture for natural language processing:Deepneural networks with multitask learning[C]Proceedings of the 25th international conference on Machine learning-lCML,08,New York:ACM Press,2008:160-167.

Liu, X.; He, P.; Chen, W.; and Gao, J. 2019. Multi-taskdeep neural networks for natural language understanding. In ACL, 4487–4496.

Subramanian, S.; Trischler, A.; Bengio, Y.; and Pal, C. J.2018. Learning general purpose distributed sentence representations via large scale multi-task learning. In ICLR.

Kingma D P, Ba J. Adam: A Method for Stochastic Optimization[A]. International Conference on Learning Representations (ICLR), 2015: 1‒15．

Ruder S. An overview of multi-task learning in deep neural networks[J]. arXiv preprint arXiv:1706.05098, 2017.

Zhang Y, Yang Q. An overview of multi-task learning[J]. National Science Review, 2018, 5(1): 30-43.

Zhang Y, Yang Q. A survey on multi-task learning[J]. IEEE Transactions on Knowledge and Data Engineering, 2021

Crawshaw M. Multi-task learning with deep neural networks: A survey[J]. arXiv preprint arXiv:2009.09796, 2020..

Vandenhende S, Georgoulis S, Proesmans M, et al. Revisiting multi-task learning in the deep learning era[J]. arXiv preprint arXiv:2004.13379, 2020, 2(3).