2.本文方法

2.1 自适应节点的硬参数网络架构

本文在硬参数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，其网络架构主要 包括 3 个部分: ①多任务底层共享参数输入层; ②自适应节点复制层；③经过单任务优化的新多任务输出层。其中硬参数网络结构如图1所示，新模型的整体架构图如图 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。

Lk

L1

共享层1

共享层2

输出层

…

…

…



W(1)

W(2)

图1

Lk

L1

共享层1

自适应节点层

输出层

…



…

…

W1)

W (2)

图2

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，输出层共有k个节点，分别记为，在最初的自适应节点层一共有j个节点，分别记，共享层1一共有m个节点，分别记为，记共享层到自适应节点层的权重矩阵为，其中到的参数记，到的参数记为，以此类推，偏置矩阵为，自适应节点层到输出层的参数矩阵为，偏置矩阵为。

我们假设模型中为两个任务的多任务模型，对于两个任务的整体损失函数为：



在反向传播过程中，对模型参数求梯度如下:



不妨设 ， 。

当时，更新参数，此时对于任务 ，损失函数减小，对于任务，损失函数增加，整体损失函数L减少。

反之，当时，此时，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E减少。

对于，它们两项单任务梯度的符号与相同。

剔除其它参数对于损失函数变动的影响，我们将节点 复制，新节点为

， 与所相对应的参数记为，我们令的初值等于。

重新更新。

此时为任务的单任务节点，为任务的单任务节点。

此时，不论的方向为正或者为为负，对于任务，若损失函数对于的梯度在一定范围内单调不变，则损失函数更大幅度减小;对于任务，损失函数依然减小，整体损失函数更大幅度减小。

本文采用Mini-batch方法训练网络,算法步骤如下:

步骤1：将数据集中的n个样本以7:3的比例划分为训练集、测试集;设置迭代次数I、每批次数据中预热训练次数阈值，连续梯度异号阈值等超参数;随机初始化共享层权重矩阵和偏置矩阵

步骤2:训练集上对进行2个任务进行联合学习，通过网络前向传播，得到输出值和损失函数值:





其中, 为输入的样本数据，共享层1到自适应节点层，以及自适应节点层到共享层2的激活函数全部设置为Sigmoid函数。接着通过误差反向传播，采用Adam算法更新共享层1和共享层2的参数、、、。

步骤3:设置T轮迭代,分别计算损失函数对共享层1到自适应节点层的权重矩阵的梯度，若损失函数对其权重的梯度方向与单任务损失函数对其权重的梯度方向相反，经过t轮迭代仍然如此，若，且总迭代次数 ，我们将自适应节点层中与其梯度相反所对应的节点复制为，将其与共享层1中所有节点相连，不妨设任务或者其中一个的损失函数中对原先的梯度＞0，整体的损失函数L对原先的梯度＞0，则将另外一个损失函数中对原先的梯度＜0的任务所连接的自适应节点层的节点的网络线拆连到所对应复制的节点，令 与所相对应的参数记为，我们令的初值等于，利用Adam算法重新更新。使得两个任务在学习到大量的共享信息之后，又进一步降低了它们的联合损失，减少了负迁移现象对模型带来的影响；

步骤4:当第一次复制完节点，更新完权重参数矩阵后，设置新的预热训练次数阈值，连续梯度异号阈值等超参数，重复步骤3，看模型是否满足复制第二次节点的条件，若满足继续重复步骤3的操作，复制第三次节点；不满足则停止迭代。

步骤5:在测试集上评估模型性能，MSE等相关评价指标。