

**人工智能软件开发与实践**

**实验报告**

题 目 多层感知机MLP

学 院 计算机科学与技术

专 业 人工智能

学 号 2021112845

学 生 张智雄

任 课 教 师 武小荷

哈尔滨工业大学计算机科学与技术学院

2023.9

**实验一:** **多层感知机MLP**

1. **实验内容**

搭建Python和Pytorch环境，并在Iris数据集上使用多层感知机实现数据的分类，主要包括三个部分：

* 1. 下载、预处理Iris数据集
  2. 使用Pytorch实现多层感知机(包含输入输出层>3层)，多层感知机的层数、隐藏层维度，batch size和epoch按需设置
  3. 在测试集上的准确率不低于90%

1. **算法简介及其实现细节**

**2.1 神经元模型**

如图1所示，在这个模型中,神经元接收到来自个其他神经元传递过来的输入信号，这些输入信号通过带权重的连接(connection)进行传递,神经元接收到的总输入值将与神经元的阈值（偏置）进行比较,然后通过“激活函数”(activation function) 处理以产生神经元的输出。

理想中的激活函数是阶跃函数，它将输入值映射为输出值“0”或“1”，显然“1”对应于神经元兴奋，“0”对应于神经元抑制。然而，阶跃函数具有不连续、不光滑等不太好的性质，实际常用Sigmoid、ReLU等函数作为激活函数

把许多个这样的神经元按一定的层次结构连接起来,就得到了神经网络。

图示

描述已自动生成

图1 M-P神经元模型

**2.2 多层感知机模型**

多层感知机(Multilayer Perceptron, MLP)是一种前向结构的人工神经网络，映射一组输入向量到一组输出向量，MLP可以被看作是一个有向图，由多个的节点层所组成，每一层都全连接到下一层，除了输入节点，每个节点都是一个带有非线性激活函数的神经元。MLP网络结构包含输入层、输出层及多个隐藏层，其中输入层神经元接收外界输入，隐藏层与输出层神经元对信号进行加工，最终结果由输出层神经元输出。3层感知机的神经网络图如下所示：

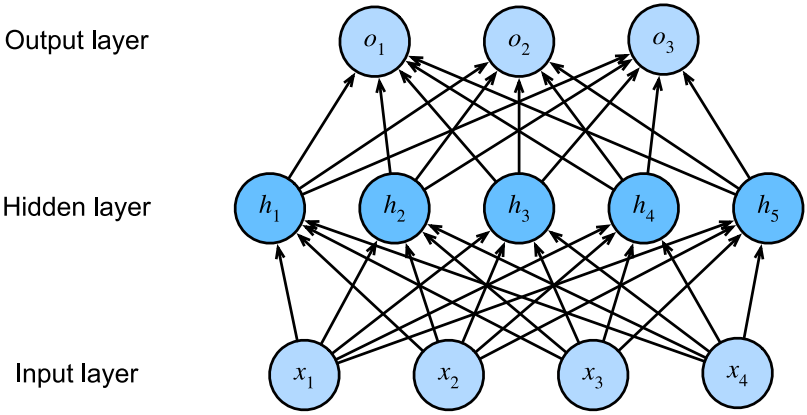


图2 多层感知机图示

一个MLP可以视为包含了许多参数的数学模型，这个模型是若干个函数相互(嵌套)代入得到的。而对于给定由个属性描述，输出为维实值向量的训练集，则隐藏层第个神经元接收到的输入为，输出层第个神经元接收到的输入为。假设神经元激活函数为。

图表, 雷达图

描述已自动生成

图3 神经网络中变量符号

模型训练主要包括前馈传播和反向传播两个步骤，前馈传播负责计算模型的预测值，而反向传播负责计算梯度并更新模型的参数，降低损失函数，以便在训练中不断改进模型的性能。

具体而言，前馈传播是神经网络中的正向计算过程，它从输入层开始，沿着网络的层级顺序将数据传递到输出层，从而计算模型的预测值，但此过程并不涉及权重和偏差的更新。

反向传播则是使用前馈传播计算模型的输出，并将其与实际目标进行比较，计算损失（误差），而后从输出层开始基于链式法则计算损失对每个权重和偏差的梯度，使用Adam、SGD等优化算法来更新网络中的权重和偏差，以减小损失函数的值。通过反复迭代前馈传播和反向传播过程，多层感知机可以逐渐调整其权重和偏差，从而提高对输入数据的表示能力和泛化能力。

1. **实验设置及结果分析（包括实验数据集）**

**3.1 Iris数据集及数据处理**

鸢尾花数据集(Iris dataset)包含了来自三个不同种类(Iris setosa, Iris versicolor, Iris virginica)的鸢尾花(Iris)的观测数据，每个种类有50个样本。每个样本由包括花萼长度(sepal length)、花萼宽度(sepal width)、花瓣长度(petal length)和花瓣宽度(petal width)四个特征描述。

在本实验中，通过np.loadtxt方法读取iris.data文件，而后通过split方法按字符’,’对特征及标签进行分割，并将标签通过字典映射为0.1.2的数字。同时将Iris数据集按0.3的划分比随机划分为训练集和验证集。

文本

描述已自动生成

图4 数据处理代码

**3.2 模型设计**

在本次实验中，实现了包含输入输出层共4层的MLP，其中激活函数选用ReLU，并通过softmax函数输出各类别的概率分布。同时使用kaiming\_uniform和xavier\_uniform方法对模型参数进行初始化，从而避免训练过程中梯度消失和梯度爆炸问题。

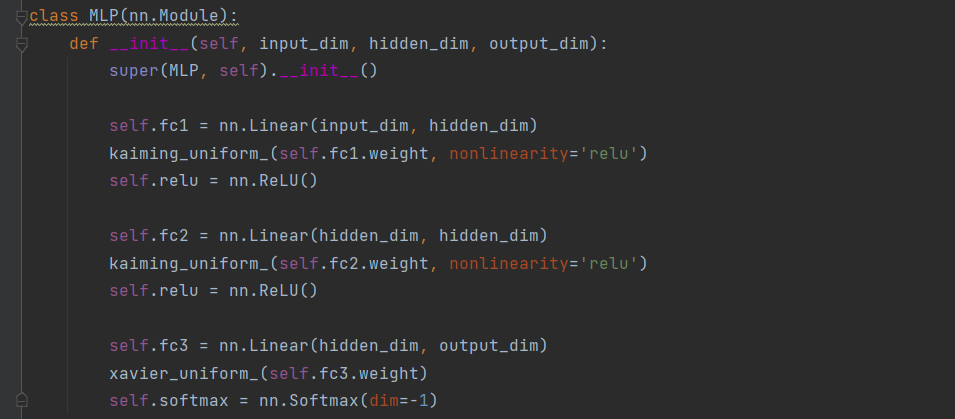


图5 模型设计代码

**3.3 实验结果**

在本次实验中，使用交叉熵损失函数和Adam优化器，将隐藏层维度设为，即MLP网络层的维度分别为，设置训练超参数为，学习率为。训练过程中损失函数的值和在测试集上的准确率变化如下图6所示。

实验发现，随训练过程的进行，损失函数不断降低，在测试集上准确率逐渐升高，最终测试正确率能够达到。测试准确率在最后阶段呈现波动态，可能原因是在局部最优点附近振荡。

同时，经过重复测试发现每次实验结果具有一定的随机性，分析可能是模型参数初始化和数据集划分的随机性导致的。

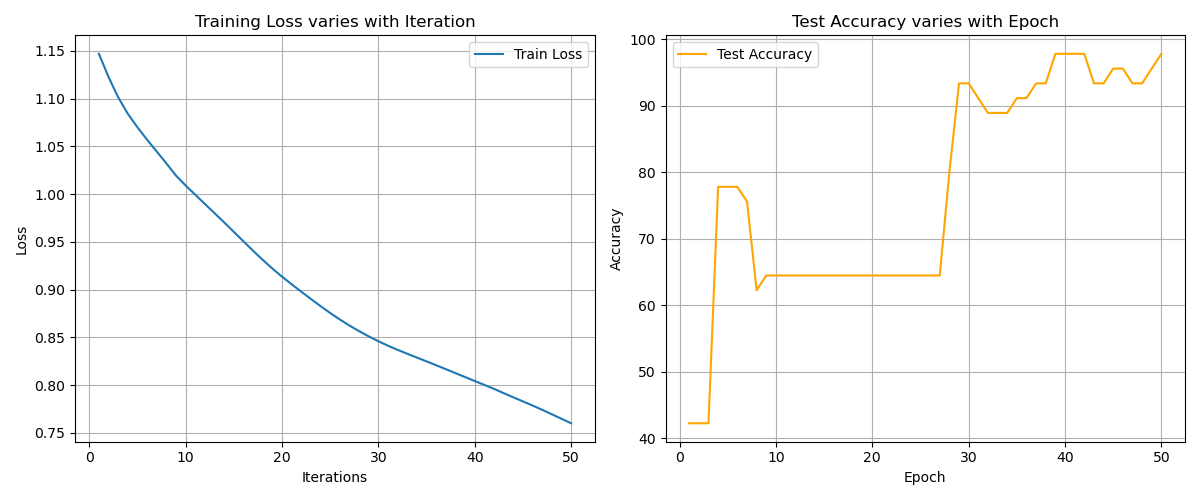


图6 实验结果（左为损失变化，右为测试集上准确率）

1. **实验结论**

多层感知机模型是矩阵与向量的乘积的非线性变换的多次重复，能够学习和捕捉复杂的非线性关系，其基本结构较为简单，其具有较强的表达能力，可适应图像分类、识别等多种人工智能任务。

1. **实验收获**

在本次实验中，通过实际使用调试了解了现有深度学习框架Pytorch，对其中常用的接口如torch.utils、torchvision.datasets、torch.nn、torch.optim等有了进一步的学习，锻炼提高了我们的代码能力。

同时在自行搭建模型框架的过程中，熟悉了MLP前向推理和反向传播的过程，熟悉了多种激活函数的使用方法。

在代码实际编写过程中，会因为不注意维度的变化和处理导致程序出错，也会因为混淆数据在CPU还是在GPU上导致赋值或比较等操作出错，也会因为学习率设置的过大或者过小导致结果不收敛，在逐步调试代码的过程中也是对我们能力的锻炼。

1. **参考文献**

[1] 李航. 统计学习方法[M]. 清华大学出版社, 2012.

[2] 周志华. 机器学习[M]. 清华大学出版社, 2016.