

Lab1Part1

一、代码基本架构

回归：拟合函数部分代码在/src/Fitting/Fitting.py 文件中，其中定义了 sigmoid 方法（即激活函数）、d_sigmoid 方法（即激活函数的导数）、compute_loss 方法（即计算平均误差）、dataloader 方法（即加载数据集）和 BPNetwork 类。类似地，分类：对 12 个手写汉字进行分类部分代码在/src/Classifier/Classifier.py 等四个文件中，其中 DataLoader.py 包含了加载数据集的方法、GradientDescent.py 包含了 BP 网络的类与方法、Classifier.py 包含了训练网络的相关代码，AccuracyTest.py 包含了测试准确率的相关代码。此外，/src/Classifier/model 文件用以存储最近一次训练网络时的参数信息。

二、不同网络结构、网络参数的实验比较

以回归：拟合函数部分为例（均为 20000 次迭代）：

网络结构	学习率	平均误差	
3 层，中间层神经元个数为 25	0.1	0.001347	[19999] The loss is 0.001347
3 层，中间层神经元个数为 25	0.05	0.009105	[19999] The loss is 0.009105
3 层，中间层神经元个数为 25	0.2	0.000532	[19999] The loss is 0.000532
3 层，中间层神经元个数为 25	0.4	0.188161	[19999] The loss is 0.188161
3 层，中间层神经元个数为 50	0.1	0.005656	[19999] The loss is 0.005656
3 层，中间层神经元个数为 100	0.1	0.000697	[19999] The loss is 0.000697
3 层，中间层神经元个数为 10	0.1	0.001605	[19999] The loss is 0.001605
3 层，中间层神经元个数为 5	0.1	0.017215	[19999] The loss is 0.017215
4 层，中间层神经元个数分别为 10、10	0.1	0.004727	[19999] The loss is 0.004727

三、对反向传播算法的理解

反向传播算法的基本思想是，学习过程由信号的正向传播和误差的反向传播两个过程组成，输入从输入层输入，经隐层处理以后，传向输出层。如果输出层的实际输出和期望输出不符合，就进入误差的反向传播阶段。误差反向传播是将输出误差以某种形式通过隐层向输

入层反向传播，并将误差分摊给各层的所有单元，从而获得各层单元的误差信号，这个误差信号就作为修正各个单元权值的依据，直到输出的误差满足一定条件或者迭代次数达到一定次数。反向传播近似最速下降算法，该方法对网络中所有权重计算损失函数的梯度。这个梯度会反馈给最优化方法，用来更新权值以最小化损失函数。