《模式识别》作业四

姓名：谷绍伟　　学号：202418020428007

1　计算与证明

1. Consider a three-layer network for classification with n_H nodes in hidden layer, and c nodes in output layer. The patterns (also say samples) are in d dimensional space. The activation function (or transfer function) for the nodes in the hidden layer is the sigmoid function. Differently, the nodes in the output layer will employ the following softmax operation as their activation function:

$$z_j = \frac{e^{net_j}}{\sum_{m=1}^{c} e^{net_m}}, \quad j = 1, 2, \ldots, c$$

where net_j stands for the weighted sum at the j-th node in the output layer.

Please derive the learning rule under the back propagation framework if the criterion function for each sample is the sum of the squared errors, that is （即分析每一层权重的更新方法）:

$$J(\mathbf{w}) = \frac{1}{2}\sum_{j=1}^{c}(t_j - z_j)^2$$

Where t_j is the known target value for the sample at the j-th node in the output layer.

注意：本题只需要推导出单个样本对权重更新的贡献即可（因为多个样本只是简单地相加）

答：计算损失的梯度：

$$\frac{\partial J(\mathbf{w})}{\partial z_j} = \ z_j - t_j \ z_j'$$

Softmax 的函数为 $s_i = \frac{e^{x_i}}{\sum_{t=1}^{k} e^{x_k}}$，对 softmax 函数求导，得到的结果为：

$$\frac{\partial s_i}{\partial x_j} = \begin{cases} s_i - s_i^2, & i = j \\ -s_i s_j, & i \neq j \end{cases}$$

隐藏层的激活函数为 Sigmoid，$o_{hj} = \frac{1}{1+e^{-x}}$，求导结果为：

$$\frac{\partial o_{hj}}{\partial x} = o_{hj}(1 - o_{hj})$$

从隐含层到输出层权重为 w_{hj}，其中 h 指隐含层节点，j 指输出层节点，其权重更新为 $w_{hj} = w_{hj} + \Delta w_{hj}$，其中 $\Delta w_{hj} = -\eta \frac{\partial J}{\partial w_{hj}}$，根据链式法则：

$$\begin{aligned}\frac{\partial J}{\partial w_{hj}} &= \frac{\partial J}{\partial z_j}\frac{\partial z_j}{\partial net_j}\frac{\partial net_j}{\partial w_{hj}} \\ &= (z_j - t_j)y_h\frac{\partial z_j}{\partial net_j} \\ &= (z_j - t_j)y_h(z_j - z_j^2)\end{aligned}$$

上式中 y_h 为隐含层节点 h 的激活后输出。

从输入层到隐含层的权重为 w_{ih}，其权重更新量为 $-\eta\Delta w_{ih}$，记 $hid_h = \sum_{h=0}^{n_H} w_{ih}x_i$ 根据链式法则：

$$\begin{aligned}\Delta w_{ih} &= \frac{\partial J}{\partial w_{ih}} \\ &= \frac{\partial J}{\partial y_h}\frac{\partial y_h}{\partial hid_h}\frac{\partial hid_h}{\partial w_{ih}} \\ &= x_i y_h(1 - y_h)\frac{\partial J}{\partial y_h} \\ &= x_i y_h(1 - y_h)\sum_{j=1}^{c}\frac{\partial J}{\partial z_j}\frac{\partial z_j}{\partial net_j}\frac{\partial net_j}{\partial y_h} \\ &= x_i y_h(1 - y_h)\sum_{j=1}^{c}(z_j - t_j)(z_j - z_j^2)w_{hj}\end{aligned}$$

2. 请对反向传播算法的训练步骤进行总结；结合三层网络给出不超过三个有关权重更新的公式，并用文字描述所述公式的含义；指出哪些因素会对网络的性能产生影响。

答：反向传播算法的训练步骤：

初始化。确定网络的结构，随机初始化神经网络中各层之间连接的权重和偏置项，设定合适的学习率等超参数。

前向传播。进行一次前向传播，计算出相应的输出值。

计算损失。根据输出层的预测输出和对应的真实标签使用合适的损失函数，计算当前预测结果与真实结果之间的差异程度。

反向传播。从输出层开始，依据计算出的损失值，按照链式法则依次反向计算各层权重和偏置对损失的梯度。

权重更新。根据计算得到的梯度，利用优化算法来更新网络中各层之间连接的权重以及偏置项。

不断重复上述步骤, 经过多轮训练, 直到网络收敛。

三层网络中权重更新的公式及含义：

输出层权重更新公式：$\Delta w_{ij}^{out} = -\eta \frac{\partial L}{\partial w_{ij}^{out}}$。其中 Δw_{ij}^{out} 表示输出层第 i 个神经元与上一层（隐藏层）第 j 个神经元之间连接权重的更新量，η 是学习率，$\frac{\partial L}{\partial w_{ij}^{out}}$ 是损失函数 L 关于该权重的偏导数，即梯度。

隐藏层权重更新公式：$\Delta w_{jk}^{hid} = -\eta \frac{\partial L}{\partial w_{jk}^{hid}}$，$\Delta w_{jk}^{hid}$ 代表隐藏层第 j 个神经元与输入层第 k 个神经元之间连接权重的更新量，同样 η 为学习率，$\frac{\partial L}{\partial w_{jk}^{hid}}$ 是损失函数关于该隐藏层权重的偏导数。

考虑偏置项的权重更新，以输出层的偏置为例，$\Delta b_i^{out} = -\eta \frac{\partial L}{\partial b_i^{out}}$，$\Delta b_i^{out}$ 表示输出层第 i 个神经元的偏置更新量，$\frac{\partial L}{\partial b_i^{out}}$ 是损失函数关于这个偏置的偏导数，η 依旧是学习率。

影响网络性能的因素：

层数和神经元数量。如果网络层数过少或者每层神经元数量过少，可能无法学习到复杂的数据模式，导致欠拟合，即对训练数据和测试数据的拟合效果都不好；相反，若网络层数过多、神经元数量过大，容易出现过拟合现象。

学习率。学习率过大可能导致权重更新幅度过大，无法收敛甚至使损失值越来越大；学习率过小则会使权重更新太慢，训练时间过长，可能陷入局部最优解而无法达到全局最优解，从而影响网络最终的性能表现。

数据质量。数据量过少难以让网络学习到足够多的模式，数据质量差（如有大量错误标注、噪声等）会干扰网络学习正确的规律，都会使网络性能不佳。

2 计算机编程

本题使用的数据如下：

第一类 10 个样本（三维空间）：

[1.58, 2.32, -5.8]	[0.67, 1.58, -4.78]	[1.04, 1.01, -3.63]	[-1.49, 2.18, -3.39]
[-0.41, 1.21, -4.73]	[1.39, 3.16, 2.87]	[1.20, 1.40, -1.89]	[-0.92, 1.44, -3.22]
[0.45, 1.33, -4.38]	[-0.76, 0.84, -1.96]		

第二类 10 个样本（三维空间）：

[0.21, 0.03, -2.21]	[0.37, 0.28, -1.8]	[0.18, 1.22, 0.16]	[-0.24, 0.93, -1.01]
[-1.18, 0.39, -0.39]	[0.74, 0.96, -1.16]	[-0.38, 1.94, -0.48]	[0.02, 0.72, -0.17]
[0.44, 1.31, -0.14]	[0.46, 1.49, 0.68]		

第三类 10 个样本（三维空间）：

[-1.54, 1.17, 0.64]	[5.41, 3.45, -1.33]	[1.55, 0.99, 2.69]	[1.86, 3.19, 1.51]
[1.68, 1.79, -0.87]	[3.51, -0.22, -1.39]	[1.40, -0.44, -0.92]	[0.44, 0.83, 1.97]
[0.25, 0.68, -0.99]	[0.66, -0.45, 0.08]		

1. 请编写两个通用的三层前向神经网络反向传播算法程序，一个采用批量方式更新权重，另一个采用单样本方式更新权重。其中，隐含层结点的激励函数采用双曲正切函数，输出层的激励函数采用 sigmoid 函数。目标函数采用平方误差准则函数。

2. 请利用上面的数据验证你写的程序，分析如下几点：

(a) 隐含层不同结点数目对训练精度的影响；

(b) 观察不同的梯度更新步长对训练的影响，并给出一些描述或解释；

(c) 在网络结构固定的情况下，绘制出目标函数随着迭代步数增加的变化曲线。

答：编写的代码见 code/文件夹，在代码中设置了两种训练方式，一种为单样本更新，一种为批样本更新，通过指定训练的 mode 可以选择不同的训练方式，其中，批量大小设置为 10。

选择中间节点数量为 10 个，学习率为 0.1，将所有数据都用于训练，共训练 400 轮次，得到不同的更新方式下的目标函数曲线如图 1。

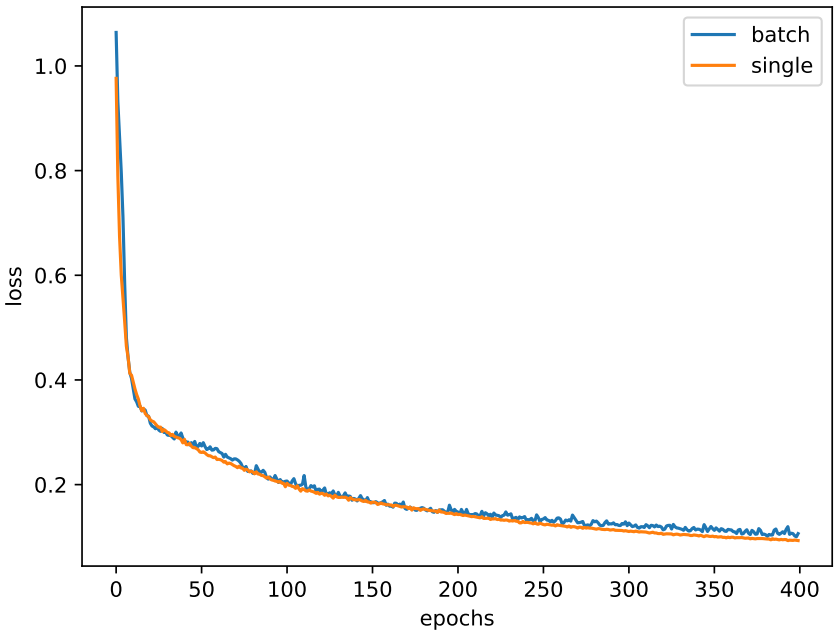

Figure 1: 两种权重更新方式下的训练损失

可见，两种不同的更新方式最终得到的训练损失相差不大，但相同参数下，批量更新方式下的曲线是震荡相对更大，单个样本更新更加稳定。

选择更新方式为批量更新，学习率为 0.1，设置中间节点个数分别为 3 个、6 个、9 个、12 个、15 个，训练 400 轮次，得到的目标函数曲线如图 2

可见，当迭代次数足够多且保持不变的情况下，隐含层结点越多，最终的训练误差越小，训练精度越高. 而当迭代次数较少时，隐含层结点数量的增多导致网络训练参数的增多，较难训练，因此结点数越多精度越低。

选择更新方式为批量更新，隐藏层节点数量为 12 个，设置学习率为 1，0.1，0.01，0.001，0.0001。为了公平比较，设置训练轮次为 800 轮，得到的目标函数曲线如图 3。

可以看到，梯度更新步长的增加可以加快误差的下降，在迭代次数相同且较小的情况下，梯度更新步长一定程度上的增大，可以提升训练精度。但当梯度更新步长过大时，

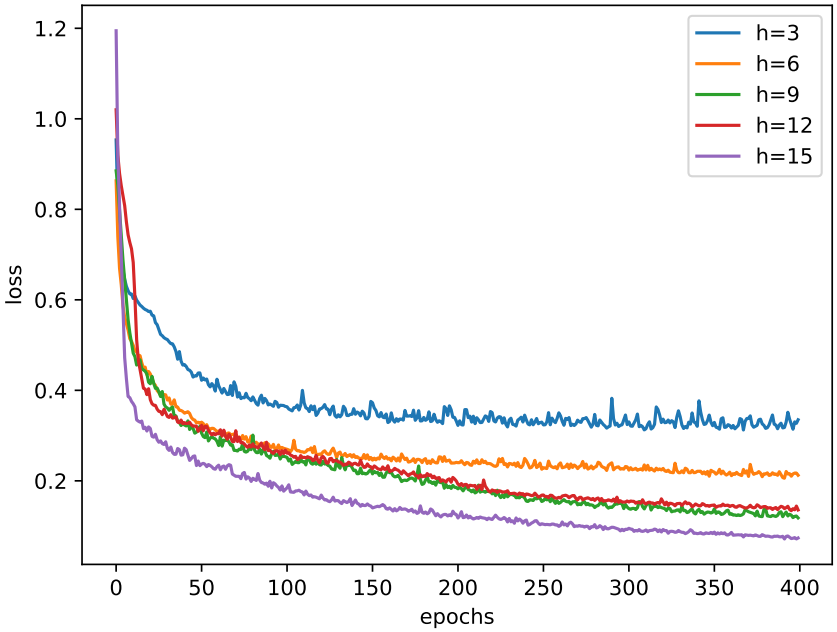

Figure 2: 不同隐藏层节点数量训练损失

网络训练变得不稳定, 会出现震荡的现象，这是因为步长过大，更新时可能越过了极小值点。因此需要根据实际情况选择合适的学习率。

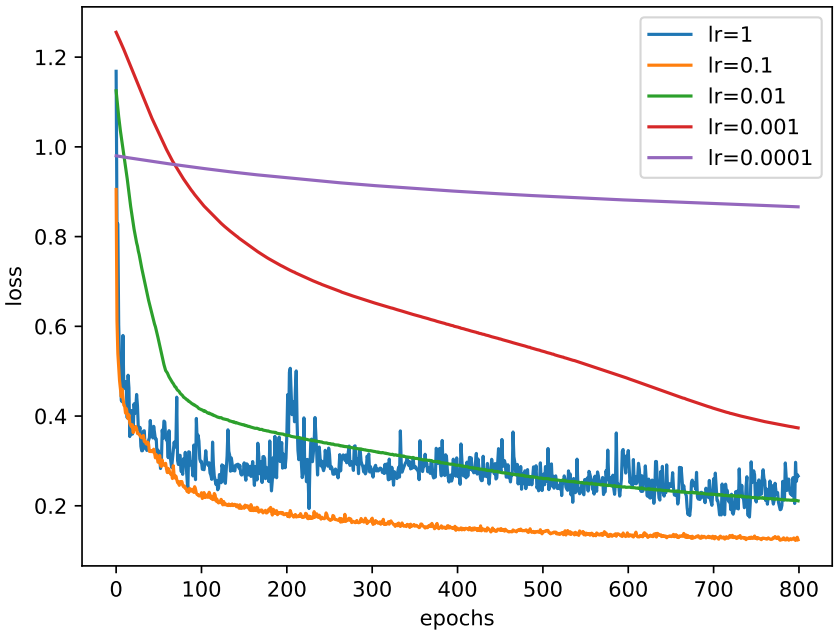

Figure 3: 不同学习率下训练损失