人工智能导论第二次作业

2023年4月

1 第一题(10分)

请简要回答下列问题:

（1）在进行算法或模型选择时，如果只划分训练集和测试集，会有什么后果？引入验证集后，验证集和测试集在使用时的区别是什么？

（2）请简述K折交叉检验的流程。

（3）相比于L2正则化，L1正则化有什么特点？除这两者之外，请再列举一种正则化方法。

（4）在支持向量机(SVM)中引入核函数(Kernel function)的动机是什么？

（5）随机森林使用哪些方法增加单棵决策树的多样性？

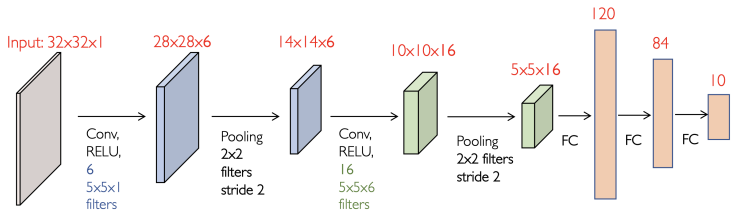
2 第二题(10分)

请简要回答下列问题:

（1）训练深度神经网络时，发现网络在训练集上的误差较小，但在测试集上的误差较大。请列举两种可以改进模型或者训练过程的措施。

（2）卷积神经网络(CNN)是针对图像类型数据特殊设计的网络结构，它的设计基于图像的哪两个基本假设？

（3）下图为LeNet网络的计算图。这个网络共有多少层？有哪些层包含可学习的参数？不考虑偏置参数(Bias)，这些层的可学习参数量分别是多少？



（4）为了解决深度神经网络难以优化的问题，ResNet在模型架构上提出了哪一改进？

（5）简述使用循环神经网络(RNN)时的训练技巧（列举两条即可）。

3 第三题(15分)

在本题中，我们将使用决策树来完成判断西瓜好坏的二分类任务。下表是关于离散属性的西瓜数据集:

色泽	根蒂	敲声	纹理	脐部	触感	好瓜
青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
青绿	稍蜷	浊响	清晰	稍凹	软粘	是
乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
青绿	硬挺	清脆	清晰	平坦	软粘	否
浅白	硬挺	清脆	模糊	平坦	硬滑	否
浅白	蜷缩	浊响	模糊	平坦	软粘	否
青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否
浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
浅白	蜷缩	浊响	模糊	平坦	硬滑	否
青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

其中，每个西瓜有色泽、根蒂、敲声、纹理、脐部、触感六个属性。我们可以计算出根节点的信息熵(Entropy)和使用“色泽”划分得到的三个子节点的信息熵（结果保留三位小数）：

$$H(D) = -\left(\frac{8}{17}\log\frac{8}{17} + \frac{9}{17}\log\frac{9}{17}\right) = 0.998$$
$$H(D_1) = -\left(\frac{3}{6}\log\frac{3}{6} + \frac{3}{6}\log\frac{3}{6}\right) = 1.000$$
$$H(D_2) = -\left(\frac{4}{6}\log\frac{4}{6} + \frac{2}{6}\log\frac{2}{6}\right) = 0.918$$
$$H(D_3) = -\left(\frac{1}{5}\log\frac{1}{5} + \frac{4}{5}\log\frac{4}{5}\right) = 0.722$$

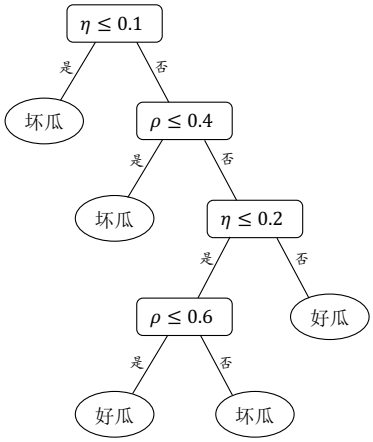
从而得到“色泽”的信息增益(Information Gain)：

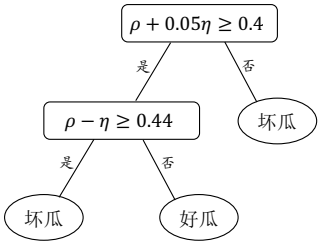
$$IG(D, \text{色泽}) = H(D) - \left(\frac{6}{17}H(D_1) + \frac{6}{17}H(D_2) + \frac{5}{17}H(D_3)\right) = 0.109$$

(1) 计算另外五个属性的信息增益。(注：代码实现可参考`entropy.ipynb`)

(2) 使用ID3算法建立决策树，每次选择信息增益最大的属性（若有多个则选择排序靠前的）进行划分。请画出ID3算法所得到的完整的决策树。

(3) 考虑以下两棵关于密度（ρ）、含糖率（η）两个连续属性的决策树：请在ρ-η 坐标系下分别画出两棵决策树的决策面。





4 第四题(25分)

在本题中，我们将推导并实现基于神经网络的多分类任务。关于标量对向量或矩阵的求导，大家可以参考：矩阵求导术。

(1) 在逻辑回归（Logistic Regression）中，我们使用伯努利分布对条件概率进行建模，对于推理结果o，相应的分布为

$$P(Y=1)=\frac{1}{1+\exp(-o)}=\sigma(o)$$

现在对于多分类问题，我们假设条件概率服从K个类上的多项分布（Multinomial Distribution），并使用Softmax回归:

$$\hat{y}_k=P(Y=k)=\frac{\exp(o_k)}{\sum_{j=1}^{k}\exp(o_j)}$$

其中$o=(o_1,o_2,...,o_K)^T\in\mathbb{R}^K$ 是模型的推理结果。对于类别k，我们将对应的标签y形式化为独热（one-hot）向量的形式，即$y=(y_1,y_2,...,y_K)^T\in\mathbb{R}^K$是一个$K$维向量，满足$y_k=1,y_j=0,\forall j\neq k$，并使用如下的交叉熵损失函数:

$$\ell=-\log\hat{y}_k=\sum_{j=1}^{K}-y_j\log\hat{y}_j$$

试求$\frac{\partial\ell}{\partial o}$，你的结果应该为一个$K$维向量。

(2) 考虑使用线性模型完成Softmax回归，即模型推理过程为$o=W^Tx$，其中$W\in\mathbb{R}^{D\times K}$为模型参数，$D$为输入列向量$x$的维度。应用矩阵求导术和链式法则求出$\frac{\partial\ell}{\partial W}$和$\frac{\partial\ell}{\partial x}$，你的结果应该与被求偏导的张量的维度相同。

(3) 进一步地，考虑使用更复杂的神经网络模型完成任务。在此我们考虑使用多层感知机（Multi-layer Perceptron, MLP）。该网络包含两层，各层参数分别为$W_1\in\mathbb{R}^{D\times H},W_2\in\mathbb{R}^{H\times K}$，其中$H$为隐藏层的维度。网络的推理过程为:

$$z=W_1^Tx,h=\sigma(z),o=W_2^Th$$

其中$\sigma(z) = \frac{1}{1+\exp(-z)}$是逐元素的Sigmoid激活函数。由(2)的结果，你已经推出了$\frac{\partial \ell}{\partial W_2}$和$\frac{\partial \ell}{\partial h}$，现在你需要进一步求出$\frac{\partial \ell}{\partial W_1}$。（提示：考虑先求出$\frac{\partial \ell}{\partial z}$）

(4) 现在我们将所推导的公式应用到实际分类问题中，并使用（不带动量的）小批量随机梯度下降（mini-batch SGD）算法进行优化。具体地，对于一个批次的样本$\{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$，目标函数为每个样本损失函数的平均值，即

$$L = \frac{1}{n}\sum_{i=1}^{n} \ell\left(\hat{y}^{(i)}, y^{(i)}\right) = \frac{1}{n}\sum_{i=1}^{n} \ell\left(\text{Softmax}\left(W_2^T \sigma\left(W_1^T x^{(i)}\right)\right), y^{(i)}\right)$$

我们的任务是MNIST手写数字数据集上的十分类问题。我们提供了代码框架，见./p4文件夹，你需要：

- 阅读并补全mlp.py，根据你前面的推导，完成神经网络的前向推理和反向传播过程，并提交代码；
- 运行mlp.py并对学习率进行适当的调参，选择合适的超参数，并汇报网络的训练集、验证集和测试集的准确率以及损失函数的训练曲线。**你的测试准确率需要达到90%。**

本题的代码基于科学计算库`numpy`完成，该库以n维数组数组`numpy.ndarray`为基础，提供了向量、矩阵等运算的快速便捷的实现以及常用的数学函数等。关于`numpy`的使用教程，以下链接可供参考：

- `https://numpy.org/`
- `https://www.runoob.com/numpy/numpy-tutorial.html`
- `https://jalammar.github.io/visual-numpy/`

(提示：在代码实现中我们通常将批次(B)作为第一维，即输入$X \in \mathbb{R}^{B\times D}$，此时批次内的每个样本将以行向量而非列向量的形式出现。因此在将公式转化为代码实现时，你需要特别注意转置的问题，例如：$Z^T = W^T X^T \Rightarrow Z = XW$)

5 第五题(40分)

编程题：在本题中，我们将基于scikit-learn等工具包解决推特文本的情感分类问题，相关材料见./p5文件夹，作业详细要求见./p5/README.md。