机器学习导论 习题三

学号, 姓名, 邮箱

2024 年 4 月 30 日

作业提交注意事项

1. 作业所需的 LaTeX 及 Python 环境配置要求请参考: [Link];
2. 请在 LaTeX 模板中第一页填写个人的学号、姓名、邮箱;
3. 本次作业需提交的文件与对应的命名方式为:
   (a) 作答后的 LaTeX 代码 — `HW3.tex`;
   (b) 由 (a) 编译得到的 PDF 文件 — `HW3.pdf`;
   (c) 第三题模型代码 — `p3_models.py`;
   (d) 第四题模型代码 — `p4_models.py`;
   (e) 第四题训练代码 — `p4_trainer.py`.

   请将以上文件**打包为 学号_姓名.zip** (例如 221300001_张三.zip) 后提交;
3. 若多次提交作业, 则在命名 .zip 文件时加上版本号, 例如 221300001_张三_v1.zip" (批改时以版本号最高的文件为准);
4. 本次作业提交截止时间为 **5 月 17 日 23:59:59**. 未按照要求提交作业, 提交作业格式不正确, **作业命名不规范**, 将会被扣除部分作业分数; 除特殊原因 (如因病缓交, 需出示医院假条) 逾期未交作业, 本次作业记 0 分; **如发现抄袭, 抄袭和被抄袭双方成绩全部取消**;
5. 学习过程中, 允许参考 ChatGPT 等生成式语言模型的生成结果, 但必须在可信的信息源处核实信息的真实性; **不允许直接使用模型的生成结果作为作业的回答内容**, 否则将视为作业非本人完成并取消成绩;
6. 本次作业提交地址为 [Link], 请大家预留时间提前上交, 以防在临近截止日期时, 因网络等原因无法按时提交作业.

1 [25pts] Principal Component Analysis

主成分分析是一种经典且常用的数据降维方法. 请仔细阅读学习《机器学习》第十章 10.3 节, 并根据图 10.5 中的算法内容, 完成对如下 6 组样本数据的主成分分析.

$$\mathbf{X} = \begin{bmatrix} 2 & 3 & 3 & 4 & 5 & 7 \\ 2 & 4 & 5 & 5 & 6 & 8 \end{bmatrix}$$

(1) **[6pts]** 试求样本数据各维的均值、标准差.

(2) **[7pts]** 试求标准化后的样本矩阵 $\mathbf{X}_{\text{std}}$, 以及 $\mathbf{X}_{\text{std}}$ 对应的协方差矩阵.
(**Hint:** 相比中心化, 标准化还需要额外除以标准差.)

(3) **[7pts]** 试求协方差矩阵对应的特征值, 以及投影矩阵 $\mathbf{W}^{\star}$.

(4) **[5pts]** 如果选择重构阈值 $t = 95\%$, 试求 PCA 后样本 $\mathbf{X}_{\text{std}}$ 在新空间的坐标矩阵.

Solution. 此处用于写解答 (中英文均可)

2 [25pts] Support Vector Machines

核函数是 SVM 中常用的工具, 其在机器学习中有着广泛的应用与研究. 请仔细阅读学习《机器学习》第六章, 并回答如下问题.

(1) **[6pts]** 试判断下图 ① 到 ⑥ 中哪些为支持向量.

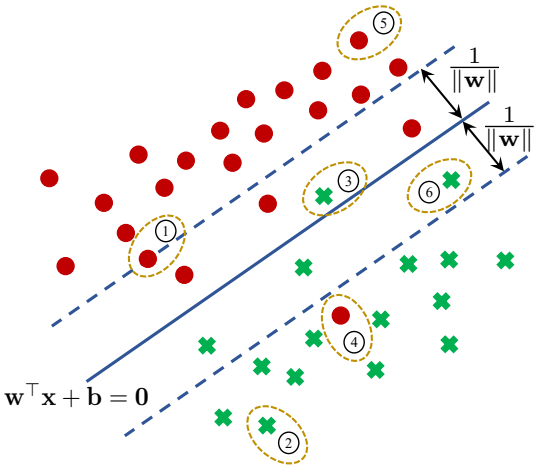

图 1: 分离超平面示意图

(2) **[5pts]** 试判断 $\kappa(\boldsymbol{x}, \mathbf{z}) = (\langle \boldsymbol{x}, \mathbf{z} \rangle + 1)^2$ 是否为核函数, 并给出证明或反例.

(3) **[5pts]** 试判断 $\kappa(\boldsymbol{x}, \mathbf{z}) = (\langle \boldsymbol{x}, \mathbf{z} \rangle - 1)^2$ 是否为核函数, 并给出证明或反例.

(4) **[9pts]** 试证明: 若 κ_1 和 κ_2 为核函数, 则两者的直积

$$\kappa_1 \otimes \kappa_2(\boldsymbol{x}, \mathbf{z}) = \kappa_1(\boldsymbol{x}, \mathbf{z}) \kappa_2(\boldsymbol{x}, \mathbf{z})$$

也是核函数. 即证明《机器学习》(6.26) 成立.

(**Hint:** 利用核函数与核矩阵的等价性.)

Solution. 此处用于写解答 (中英文均可)

3 [30pts] Basics of Neural Networks

多层前馈神经网络可以被用作分类模型. 在本题中, 我们先回顾前馈神经网络的一些基本概念, 再利用 Python 实现一个简单的前馈神经网络以进行分类任务.

[基础原理] 首先, 考虑一个多层前馈神经网络, 规定网络的输入层是第 0 层, 输入为 $\mathbf{x} \in \mathbb{R}^d$. 网络有 M 个隐层, 第 h 个隐层的神经元个数为 N_h, 输入为 $\mathbf{z}_h \in \mathbb{R}^{N_{h-1}}$, 输出为 $\mathbf{a}_h \in \mathbb{R}^{N_h}$, 权重矩阵为 $\mathbf{W}_h \in \mathbb{R}^{N_{h-1} \times N_h}$, 偏置参数为 $\mathbf{b}_h \in \mathbb{R}^{N_h}$. 网络的输出层是第 $M+1$ 层, 神经元个数为 C, 权重矩阵为 $W_{M+1} \in \mathbb{R}^{N_M \times C}$, 偏置参数为 $\mathbf{b}_{M+1} \in \mathbb{R}^C$, 输出为 $\mathbf{y} \in \mathbb{R}^C$. 网络隐层和输出层的激活函数均为 f, 网络训练时的损失函数为 $\mathcal{L}$, 且 f 与 $\mathcal{L}$ 均可微.

(1) **[5pts]** 请根据前向传播原理, 给出 $\mathbf{z}_h, \mathbf{a}_h$ $(1 \leq h \leq M)$ 及 $\mathbf{y}$ 的具体数学表示.

(2) **[5pts]** 结合 (1) 的表示形式, 谈谈为何要在神经网络中引入 (非线性) 激活函数 f?

[编程实践] 下面, 我们针对一个特征数 $d = 2$, 类别数为 2 的分类数据集, 实现一个结构为 "2-2-1" 的简单神经网络, 即: 输入层有 2 个神经元; 隐层仅一层, 包含 2 个神经元; 输出层有 1 个神经元; 所有层均使用 Sigmoid 作为激活函数. 此外, 我们使用 BP 算法进行神经网络的训练. **关于本题的细节介绍及具体要求, 请见附件: `p3_`编程题说明.** 请参考编程题说明文档与附件中的代码模板, 完成下面的任务.

(3) **[15pts]** 基于 `p3_models.py`, 补全缺失代码, 实现神经网络分类器的训练与预测功能.

(4) **[5pts]** 参考《机器学习》及第一次作业中对超参数调节流程的介绍, 为 (1) 中模型设置合适的超参数 (即: 学习率与迭代轮数). **请将选择的超参数设置为调用模型时的默认参数, 并在解答区域简要介绍你的超参数调节流程.**
(提示: 可以从数据集划分方法, 评估方法, 候选超参数生成方法等角度说明).

Solution. 此处用于写解答 (中英文均可)

4 [20(+5)pts] Neural Networks with PyTorch

> 在上一题的编程实践中, 我们使用 Python 实现了一个简单的神经网络分类器. 其中, 我们根据 BP 算法中神经网络参数梯度的数学定义, 手动实现了梯度计算及参数更新的流程. 然而, 在现实任务中, 我们往往利用深度学习框架来进行神经网络的开发及训练. 一些常用的框架例如: PyTorch, Tensorflow 或 JAX, 以及国产的 PaddlePaddle, MindSpore. 这类框架往往支持自动微分功能, 仅需定义神经网络的具体结果与前向传播过程, 即可在训练时自动计算参数的梯度, 进行参数更新. 此外, 我们可以使用由框架实现的更成熟的优化器 (如 Adam 等) 来提高模型的收敛速度, 或使用 GPU 加速以提高训练效率. 如果希望在今后的学习科研中应用神经网络, 了解至少一种框架的使用方式是极为有益的.

在本题中, 我们尝试使用 PyTorch 框架来进行神经网络的开发, 完成 FashionMNIST 数据集上的图像分类任务. 与上一题考察神经网络底层原理不同, 本题考察大家阅读文档, 搭建模型并解决实际任务的能力. **关于本题的细节介绍及具体要求, 请见附件: `p4_`编程题说明.** 请参考编程题说明文档与附件中的代码模板, 完成下面的任务.

(1) **[10pts]** 阅读文档, 配置 PyTorch 环境, 补全 `p4_models.py` 中神经网络的 `__init__` 与 `forward` 方法, 最终成功运行 `p4_main.py`. **请在解答区域附上运行 `p4_main.py` 后生成的 `plot.png`.**

(2) **[10pts]** 从 (1) 中生成的训练过程图片 `plot.png` 中可以看出: 模型明显出现了**过拟合**现象, 即训练一定轮次后, 训练集 loss 持续下降, 但测试集 loss 保持不变或转为上升. 请提出**至少两种**缓解过拟合的方法, 分别通过编程实现后, 在解答区域附上应用前后的训练过程图片, 并结合图片简要分析方法有效/无效的原因.
(提示: 可以考虑的方法包括但不限于: Dropout, 模型正则化, 数据增强等.)

(3) **[5pts]** (本题为附加题, 得分计入卷面分数, 但本次作业总得分不超过 100 分)
寻找最优的改进神经网络结构及训练方式的方法, 使模型在另一个未公开的测试集上取得尽可能高的分类准确率.
本题得分规则如下: 假设共有 N 名同学完成本题, 我们将这 N 名同学的模型测试集分类准确率由高到低排列, 对前 $K = \min(\lfloor N/10 \rfloor, 10)$ 名同学奖励附加题分数. 对于排列序号为 i 的同学 ($1 \le i \le K$), 得分为: $5 - \lfloor 5(i-1)/k \rfloor$.
(提示: 你可以自由尝试修改模型结构, 修改优化器超参数等方法.)

Solution. 此处用于写解答 (中英文均可)

Acknowledgments

允许与其他同样未完成作业的同学讨论作业的内容, 但需在此注明并加以致谢; 如在作业过程中, 参考了互联网上的资料, 且对完成作业有帮助的, 亦需注明并致谢.