山东大学 计算机科学与技术 学院

机器学习与模式识别 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202300130183 | 姓名： 宋浩宇 | | 班级：23级人工智能班 |
| 实验题目：Regularization | | | |
| 实验学时：2 | | 实验日期：2025/3/25 | |
| 实验环境：  软件环境：  系统：Windows 11 家庭中文版23H2 22631.4317  计算软件：MATLAB 版本: 9.8.0.1323502 (R2020a)  Java 版本: Java 1.8.0\_202-b08 with Oracle Corporation Java HotSpot(TM) 64-Bit Server VM mixed mode  硬件环境：  CPU：13th Gen Intel(R) Core(TM) i9-13980HX 2.20 GHz  内存：32.0 GB (31.6 GB 可用)  磁盘驱动器：NVMe WD\_BLACKSN850X2000GB  显示适配器：NVIDIA GeForce RTX 4080 Laptop GPU | | | |
| 1. 实验内容   In this exercise, you will implement regularized linear regression and regularized  logistic regression.   1. 实验步骤   第一部分的实验：   1. 加载数据      1. 构造模型 2. 使用matlab计算参数 3. 完成结果可视化 4. 更改lambda值，重新计算并绘制曲线   第二部分实验：   1. 加载数据      1. 构造模型 2. 使用matlab完成参数计算 3. 完成结果的可视化 4. 更改lambda的值，重新计算绘制等高线图。 5. 测试结果   关于第一部分实验：    （此时lambda为0）    （此时lambda为1）    （此时lambda为10）  简单来说，lambda越大，模型的预测值对于点的距离就越远。  关于第二部分实验：    （此时lambda为0）    （此时lambda为1）    （此时lambda为10）  复杂地来说，lambda会影响模型收敛的速度和位置，适当的lambda的值对于避免过拟合问题起着重要作用，而lambda过小有可能效果不明显，lambda过大则有可能会导致模型错误。   1. 附录：实现源代码  |  | | --- | | %% 清空工作区  clc;clear;close all;  %% 加载数据  x = load('ex5Data/ex5Logx.dat')  y = load('ex5Data/ex5Logy.dat')  figure  pos = find(y)  neg = find(y==0)  plot(x(pos,1),x(pos,2),'+')  hold on  plot(x(neg,1),x(neg,2),'o')  % hold off  new\_x = map\_feature(x(:,1),x(:,2))  theta = ones(size(new\_x,2),1)  lambda = 10  % disp(train(new\_x,y,lambda))  % dis = dJ(new\_x,y,theta,lambda)  % dis = H(theta,new\_x,y,lambda)  % disp(inv(dis))  theta = train(new\_x,y,lambda)  u = linspace(-1,1.5,200)  v = linspace(-1,1.5,200)  z = zeros(length(u),length(v))  for i = 1:length(u)      for j = 1:length(v)          z(i,j) = map\_feature([u(i)],[v(j)])\*theta;      end  end  contour(u,v,z',[0,0],'LineWidth',2);  %% 函数部分  % h(x)  function hx = h(theta,x)      % 计算sigmoid函数值      hx = 1./(1+exp(-(x\*theta)));      % disp("hx=")      % disp(hx)  end  % J(x)  function J = J(theta,x,y,lambda)      m = size(x,1);      sum = 0;      for i = 1:m          sum = sum - y(i)\*log(h(theta,x(i,:))) - (1-y(i))\*log(1-h(theta,x(i,:)));      end      J = sum / m + (lambda/(2\*m))\*(theta'\*theta);  end  % dJ(x)  function dJ = dJ(x,y,theta,lambda)      m = size(theta,1);      n = size(x,1);      sum = 0;      for k = 1:size(x,1)          sum = sum + (h(theta,x(k,:)) - y(k)).\*x(k,1);      end      dJ = sum / n;      sum = 0;      for j = 2:size(x,2)          for k = 1:size(x,1)              sum = sum + (h(theta,x(k,:)) - y(k)).\*x(k,j);          end          sum = sum / n          sum = sum + (lambda/n)\*theta(j);          dJ = [dJ;sum];          sum = 0      end      % disp("dJ")      % disp(dJ)  end  % has(x)  function H = H(theta,x,y,lambda)      m = size(x,1);      sum = 0;      for i = 1:m          sum = sum + h(theta,x(i,:))\*(1-h(theta,x(i,:)))\*(x(i,:)')\*x(i,:)          % disp('h(theta,x(i,:))\*(1-h(theta,x(i,:)))')          % disp(h(theta,x(i,:))\*(1-h(theta,x(i,:))))      end      sum = sum / m      tem = eye(size(theta,1))      tem(1,1) = 0;      H = sum + (lambda/m)\*tem  end  function result = train(x,y,lambda)      theta = zeros(size(x,2),1)      % disp("theta0")      % disp(theta)      J\_his = []      for i = 1:15          J\_his = [J\_his,J(theta,x,y,lambda)];          theta = theta - inv(H(theta,x,y,lambda))\*dJ(x,y,theta,lambda);      end      result = theta      disp(J\_his)  end | | | | |