

# 最优化方法

## 多层神经网络的训练问题

西安交通大学, 数学与统计学院, 强基数学 002

姓名  $1^a$ , 姓名  $2^b$ , 姓名  $3^c$

学号/邮箱  $1^a$ , 学号/邮箱  $2^b$ , 学号/邮箱  $3^c$

2024 年 12 月 12 日

目录

目录

# 小作业的简单标题

名称 学号 作者信息

## 1 实验目的

## 2 实验原理

## 3 实验步骤与结果分析

## 4 结论与讨论

---

```
1 # coding:UTF-8
2 import numpy as np
3 import math
4
5 y = [0]
6 h = 0.1
7 f = lambda y: 1-y
8 for i in range(1, 11):
9     k1 = h * f(y[i-1])
10    k2 = h * f(y[i-1] + k1 / 2)
11    k3 = h * f(y[i-1] + k2 / 2)
12    k4 = h * f(y[i-1] + k3)
13    tmp = y[i-1] + (k1 + 2*k2 + 2*k3 + k4) / 6
14    y.append(round(tmp * 1e3) / 1e3)
15
16 for i in range(11):
17     print('y_{{{}}} = {:.3f},\\quad '.format(i, y[i]), end='')
18     if i == 5:
19         print()
```

---

```
1 #include <iostream>
2 using namespace std;
3 int gcd(int a, int b) {return b ? gcd(b, a % b) : a;}
4 signed main() {
5     cout << gcd(32, 24) << '\n';
6     system("pause");
7     return 0;
8 }
```

---

```
1 function [H] = calc_homography(x, X)
2     shape = size(x);
3     x_camera = [x; ones(1, shape(2))];
4     x_world = [X(1:2,:); ones(1, shape(2))];
5     H = compute_homography(x_camera, x_world);
6 end
```

---