宋琪语-10235501465第六周作业

2024年10月14日　　18:19

2. PageRank 的设计思想是什么？
3. 贝叶斯定理的内容是什么？它又有哪些重要应用？
4. 试阐述蒙特卡罗方法的基本原理。
5. 梯度下降法的主要思想是什么？你能用通俗的语言解释出来吗？

复习题：

第二题：PageRank的设计思想一个网页如果被许多其他重要的网页链接到，那么这个网页本身也应该是重要的。即利用互联网自身的超链接结构来给网页打分

第三题：

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

在B事件下A事件发生的概率。在计算患者患有某种疾病的概率，特征数据预测样本的类别和根据测试结果和缺陷率预测产品的质量等方面有应用

第四题：蒙特卡罗方法的基本原理是通过大量随机抽样来近似计算数学和物理问题的概率和统计特性。需要概率模型描述了系统的状态空间和系统状态的概率分布，再从其中随机抽样，计算出一个与问题相关的统计量或函数值，重复多次，取该统计值的平均值。

第五题：梯度下降法用于寻找目标函数（通常是损失函数或代价函数）的局部最小值。其核心思想是在每次迭代过程中，沿着目标函数在当前参数位置的负梯度方向调整参数值，因为负梯度方向指示了函数值下降最快的方向。通过不断调整参数，使函数值逐渐减小，直到达到一个足够小的变化量或者达到预定的迭代次数为止。

践习题：

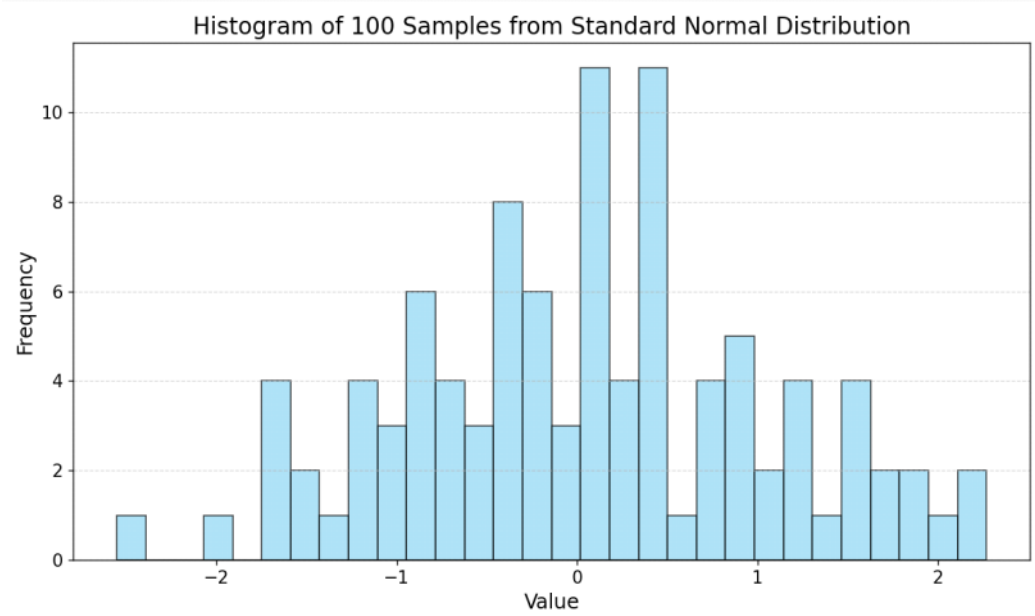
第一题：

```
import numpy as np
np.random.seed(0)
samples = np.random.normal(loc=0.0, scale=1.0, size=100)
print(samples)
```

```
-1.70627019  1.9507754  -0.50965218 -0.4380743  -1.25279536  0.77749036
-1.61389785 -0.21274028 -0.89546656  0.3869025  -0.51080514 -1.18063218
-0.02818223  0.42833187  0.06651722  0.3024719  -0.63432209 -0.36274117
-0.67246045 -0.35955316 -0.81314628 -1.7262826   0.17742614 -0.40178094
-1.63019835  0.46278226 -0.90729836  0.0519454   0.72909056  0.12898291
 1.13940068 -1.23482582  0.40234164 -0.68481009 -0.87079715 -0.57884966
-0.31155253  0.05616534 -1.16514984  0.90082649  0.46566244 -1.53624369
 1.48825219  1.89588918  1.17877957 -0.17992484 -1.07075262  1.05445173
-0.40317695  1.22244507  0.20827498  0.97663904  0.3563664   0.70657317
 0.01050002  1.78587049  0.12691209  0.40198936]
```

第二题：

```
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(0)
samples = np.random.normal(loc=0.0, scale=1.0, size=100)
plt.figure(figsize=(10, 6))
plt.hist(samples, bins=30, color='skyblue', edgecolor='black', alpha=0.7)
plt.title('Histogram of 100 Samples from Standard Normal Distribution', fontsize=16)
plt.xlabel('Value', fontsize=14)
plt.ylabel('Frequency', fontsize=14)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.grid(axis='y', linestyle='--', linewidth=0.7, alpha=0.7)
plt.tight_layout()
plt.show()
```



第三题：

```
import numpy as np
# 定义一个2x2的矩阵
A = np.array([[2, 1],
              [4, 5]])
# 计算矩阵的特征值和特征向量
eigenvalues, eigenvectors = np.linalg.eig(A)
print("矩阵的特征值为:")
print(eigenvalues)
print("对应每个特征值的特征向量为:")
for i in range(len(eigenvalues)):
    print(f"特征值 {eigenvalues[i]}: {eigenvectors[:, i]}")
```

```
矩阵的特征值为:
[1. 6.]
对应每个特征值的特征向量为:
特征值 1.0: [-0.70710678  0.70710678]
特征值 6.0: [-0.24253563 -0.9701425 ]
```

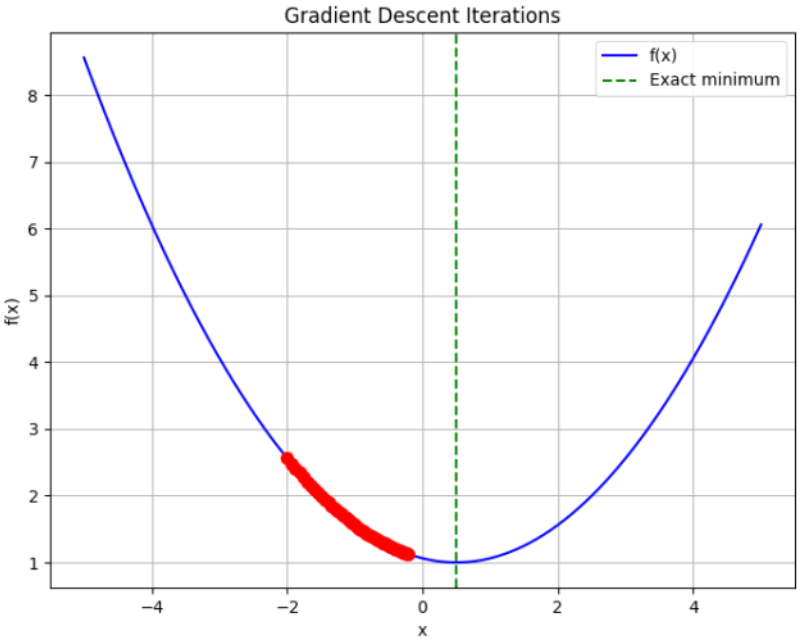
第五题：

```
import numpy as np
B = np.array([
    [1, 2, 3],
    [1, -1, 4],
    [1, 3, -1]
]).T
cov_matrix = np.cov(B, rowvar=False)
print(cov_matrix)
```

```
[[ 1.         1.5        -1.        ]
 [ 1.5        6.33333333 -5.        ]
 [-1.        -5.         4.        ]]
```

附加题：

```
import numpy as np
import matplotlib.pyplot as plt
def f(x):
    return 0.25 * (x - 0.5)**2 + 1
def df(x):
    return 0.5 * (x - 0.5)
learning_rate = 0.05
num_iterations = 50
x_initial = -2
x_values = [x_initial]
f_values = [f(x_initial)]
x = x_initial
for i in range(num_iterations):
    x -= learning_rate * df(x)
    x_values.append(x)
    f_values.append(f(x))
x_range = np.linspace(-5, 5, 400)
plt.figure(figsize=(8, 6))
plt.plot(x_range, f(x_range), label='f(x)', color='blue')
plt.scatter(x_values, f_values, color='red', s=50, zorder=5)
plt.plot(x_values, f_values, color='red', linestyle='--', zorder=1)
plt.axvline(x=0.5, color='green', linestyle='--', label='Exact minimum')
plt.xlabel('x')
plt.ylabel('f(x)')
plt.title('Gradient Descent Iterations')
plt.legend()
plt.grid(True)
plt.show()
print("局部极小值点近似为：", x)
print("对应的函数值为：", f(x))
```


Gradient Descent Iterations
f(x)
Exact minimum
f(x)
x
8
7
6
5
4
3
2
1
−4
−2
0
2
4