Course Project Report: Advanced Math Analysis with Matlab

KeZheng Xiong

22920202204622

2021 年 12 月 21 日

摘要

The report for the end-of-term project of Advanced Math Analysis with MATLAB fall 2021 course. All the source code is open-sourced on the Github repository `https://github.com/SmartPolarBear/matlab-math-analysis-csxmu-2021` under **GPLv3** license

目录

1 Problem 1

1.1 Problem Description

Given function $F(x,y) = 0.2x^2 + 0.1y^2 + sin(x+y)$, please work out its gradient. Based on the gradient, please find out the local extreme of function F(x,y) when both x and y are in the range of $[-2*\pi, 2*\pi]$. The 2D and 3D views of the function is given in Fig. 1.

1.2 Solution

1.2.1 The gradient of the function

I get the gradient of the function using the following code

Listing 1: Gradient Calculation

```
syms x y;
f=0.2*x^2+0.1*y^2+sin(x+y);
diff(f,x)
diff(f,y)
```

Based on the result, the gradient is

$$\nabla \cdot f(x,y) = (\frac{2*x}{5} + cos(x+y), \frac{y}{5} + cos(x+y)) \tag{1}$$

1.2.2 Find the extreme values

To find the extreme values of $F(x,y)$ with gradient decent method, we walk little steps towards the direction of the gradient. To formalize this idea, the algorithm is shown as follows.

Algorithm 1 Gradient Descent

Input: Initial point x_0, a constant α, $k=0$

while termiation condition does not hold **do**

$k = k+1$

$x_{k+1} = x_k - \alpha\nabla \cdot f(x_k)$

Various problems occurs if this brute-force algorithm is implemented directly. The speed of convergence is annoyingly slow if parameters are not chosen right. In fact, I never succeeded finding a set of parameters that works. A well-known optimization is called Stochastic gradient descent, or SGD, improve it significantly.

The given parameter α in the brute-force algorithm, which is referred as learning rate, will change each round according to the situation. To be more exact, SGD tries to find a learning rate m, so that it minimize the function

$$h(x,y,m) = \mathbf{x_0} + \nabla \cdot f(x,y) \tag{2}$$

so the following equation is solved each round in the while-loop

$$\frac{\partial h}{\partial m} = 0 \tag{3}$$

The implementation is shown in Code 2

Listing 2: SGD Gradient Descent

```
        syms x y;
        f=0.2*x^2+0.1*y^2+sin(x+y);
        diff(f,x)
        diff(f,y)
```

1.3 Acknowledgment

Thanks to (TODO)