计算机科学与技术学院神经网络与深度学习课程实验报告

实验题目：The Optimization of Neural Networks		学号：201600181058
日期：2019-3-28	班级：16 人工智能班	姓名： 张多
Email：976539567@qq.com		

实验目的：

Adjust the parameters of the three-layer-neural networks and optimize the output result. Complete the code separately and test the effect of different regularization coefficients, different methods of gradient descent, different methods of initialization of the parameters and the gradient checking process.

实验软件和硬件环境：

Python 3.6

ThinkPad X1Carbon 8G+256G

Experiment Principles and methods:

1. Gradient checking

   Gradient checking aims at checking the correctness of the analytical solutions of gradients calculated by the matrix calculus by comparing the analytical solutions with the numerical gradients calculated with the method of center difference.

2. Initialization of the Parameters

   The initialization of the parameters has three prevalent methods: zero-initialization, random initialization, He initialization.

   - Zero-initialization

     Allocate zeros to all parameters. However, this kind of initialization will lead to plenty of problems and sometimes the parameters are not changing at all.

   - Random initialization

     Allocate random numbers to all weights and zeros to all biases. The scale and magnitude of the random numbers are critical factors which greatly influence the result of the gradient descent process. If the parameters are not set properly, sometimes the loss will exceed or overflow the calculation limits and become ***NAN*** in Python.

   - He initialization

     We all know that the scale of the random numbers matters. So He's method gives a clear approach which is to set the coefficients of the random numbers to

     $$\sqrt{\frac{2}{\text{dimensions of last layer}}}$$

     which gives a better result.

3. Optimization methods
   The common method to optimize the loss function of a neural network is gradient descent. There are some kinds of gradients descent methods:
   - Batch Gradient Descent
     Use all samples to calculate the loss function and implement the gradient descent to the loss function. The method is slow and need expensive calculations.
   - Stochastic Gradient Descent
     Use only one sample to calculate the loss value and implement the gradient descent to this sample. SGD converges very slowly and is unstable, but faster than batch gradient descent.
   - Mini-Batch Gradient Descent
     Use a group of samples to implement the gradient descent, not all samples or just one sample. The convergence efficiency is much higher and not so computationally expensive and more stable.
   - Adam Gradient Descent
     Add a momentum term to the gradient descent to keep the descending direction near the original direction, and boost and speed up the convergence.
4. Regularization
   Regularization is a great way to avoid the overfitting problem. This experiment we implement the L2 norm regularization. The bigger regularization coefficients are, the overfitting is more unlikely to happen.

实验步骤：（不要求罗列完整源代码）

1. Complete the code snippets.
2. Run the code and test all the optimization methods.
3. Use all the methods to optimize the neural network in the last experiment.
4. Compare the results and get the best parameters.

结论分析与体会：

1. The gradient checking passed the test successfully. There are two mistakes in the gradient calculating part. One is dW2 is multiplied by 2 and db2 is multiplied by 4.

```
J = 8
dtheta = 2
The gradient is correct!
difference = 2.919335883291695e-10
Your backward propagation works perfectly fine! difference = 1.1890913023330276e-07
```

2. In the initializing part, the zero-initialization failed because all the parameters remain zero.
   The random initialization has a better performance.
   The training accuracy is 0.83, while the test accuracy is 0.86.
   Here's the loss function decay:

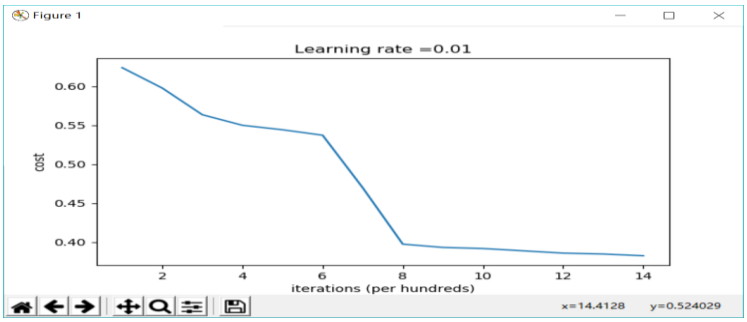

Figure 1. the loss decadency

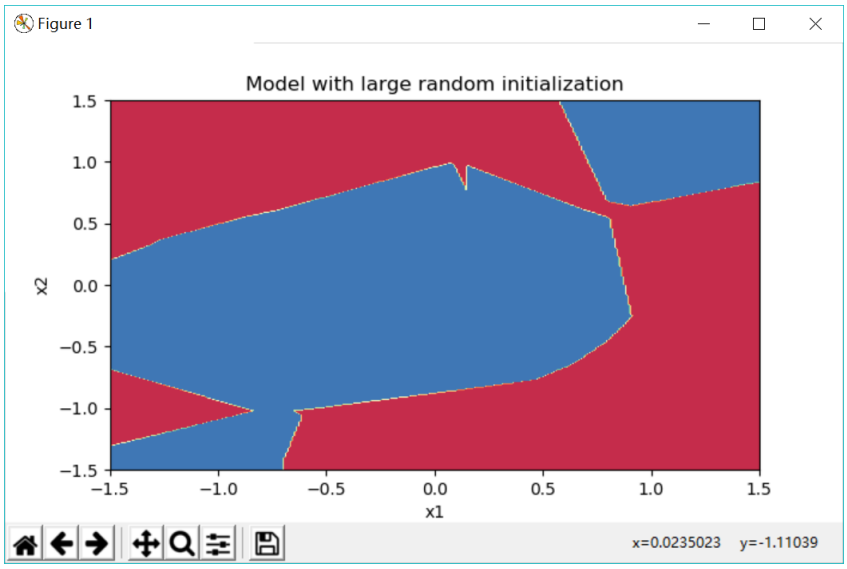

Figure 2. the classification result

The He initialization has an accuracy on the train set of 0.993 and the test accuracy is 0.93. The He initialization obviously has a better performance:

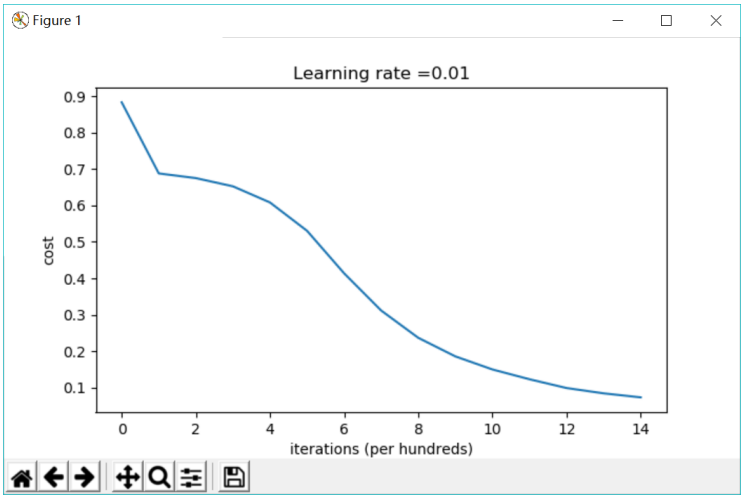

Figure 3. the loss decay of the he initialization

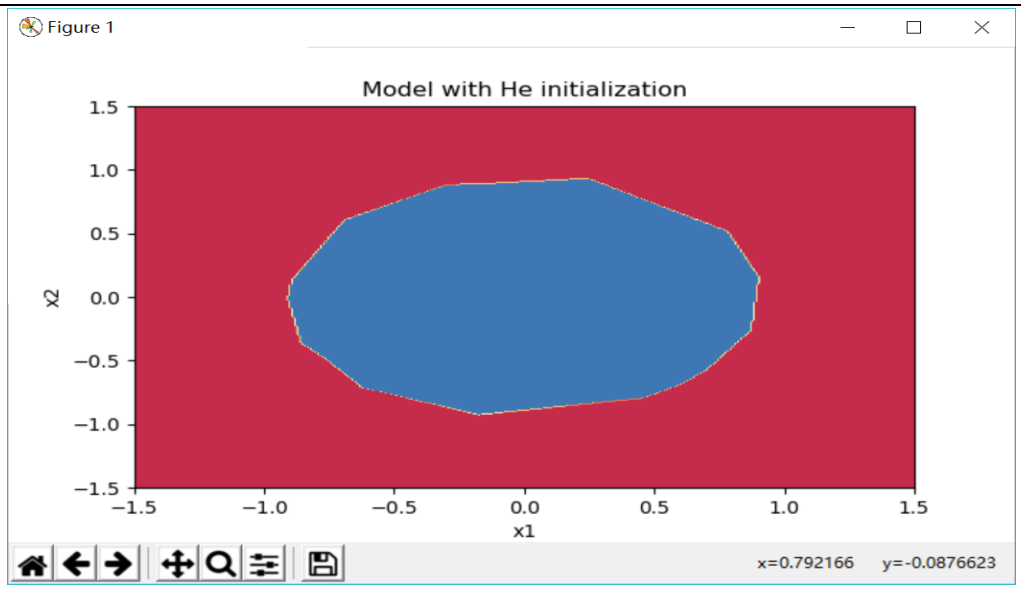

Figure 4. the performance of He initialization

3. The optimization of gradient descent:

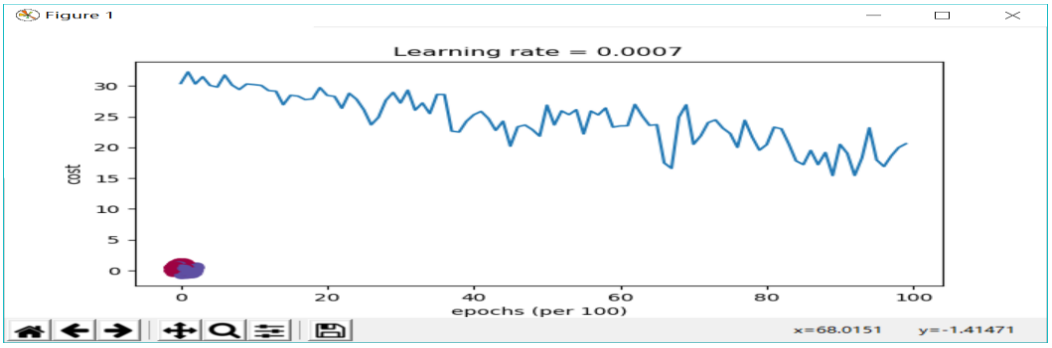

Figure 5. the regular GD

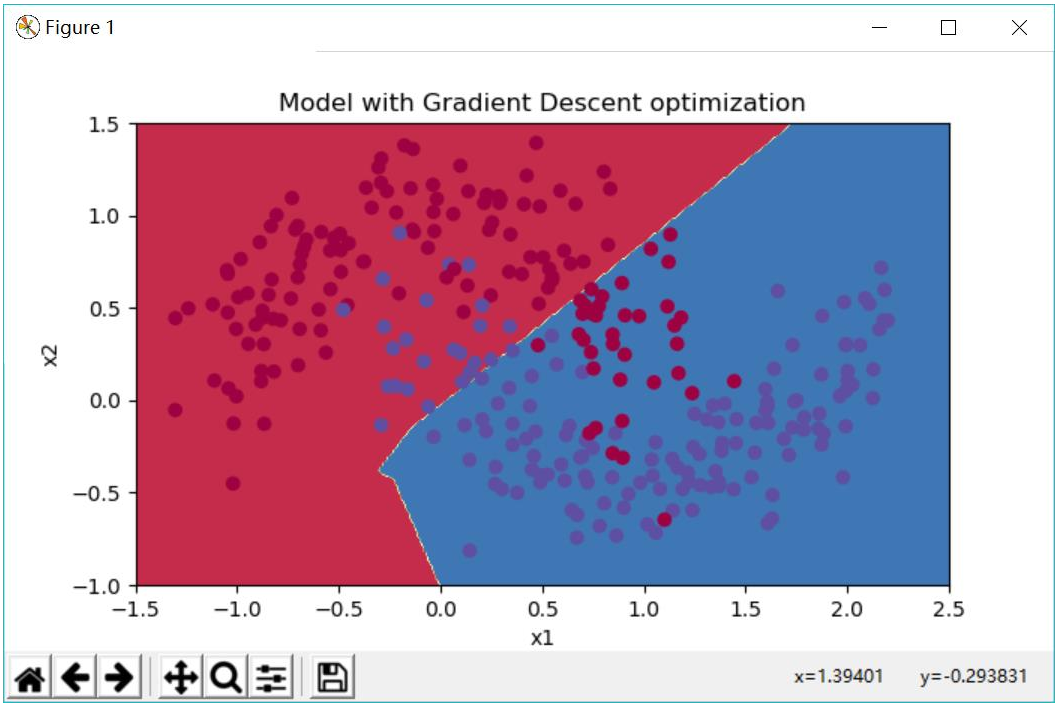

Figure 6. the classification result of regular GD

The accuracy is 0.796.

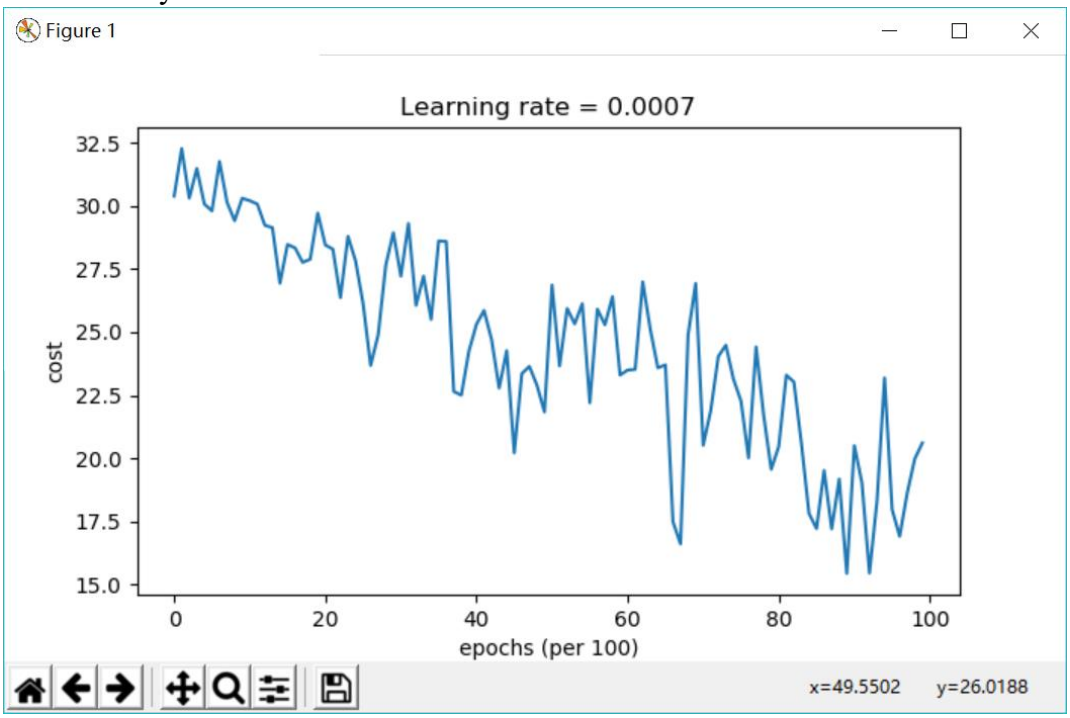

Figure 7. the loss decadency of momentum method

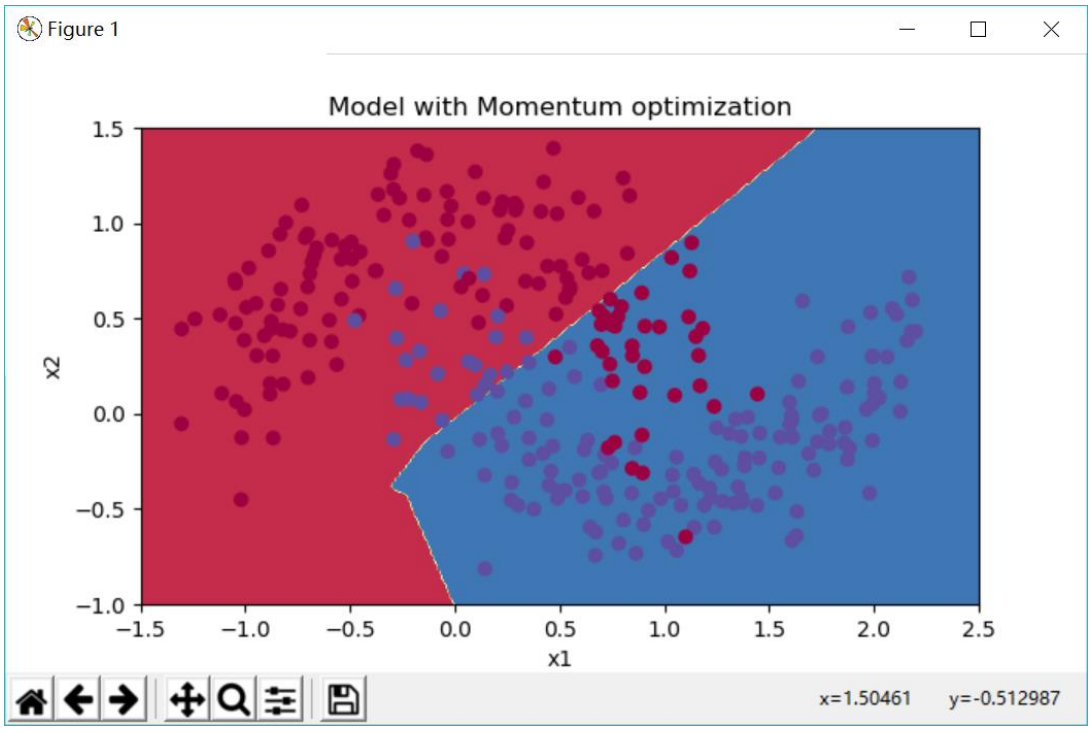

Figure 8 the classification result of momentum method

The accuracy is still 0.796

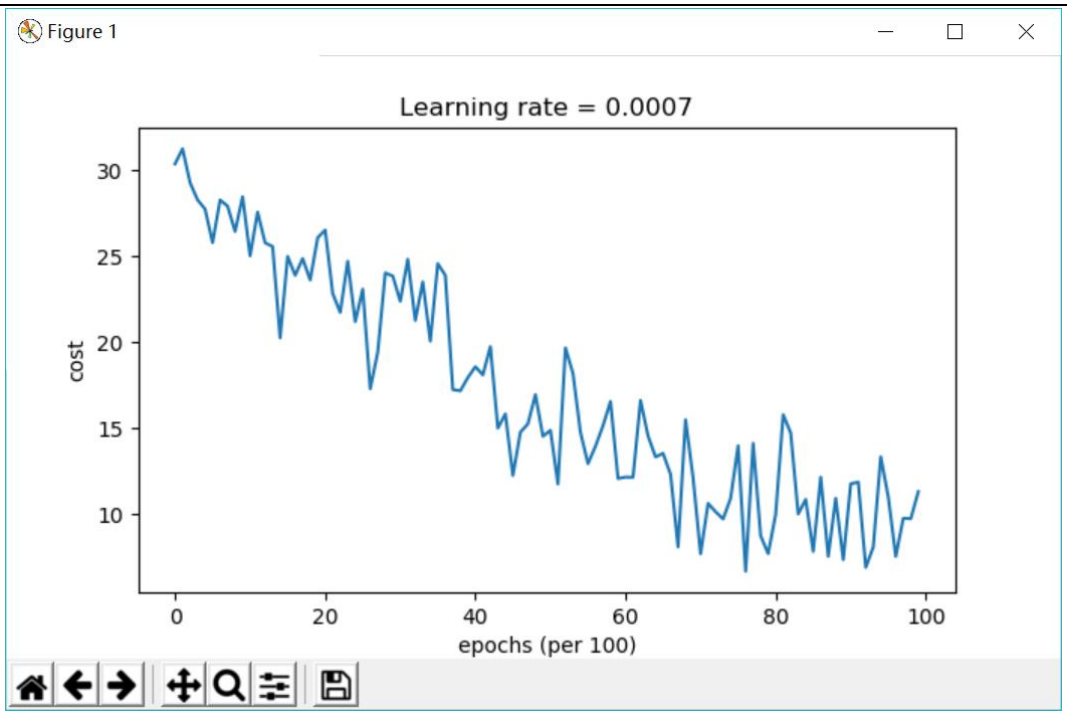

Figure 9. the loss decay of the Adam method

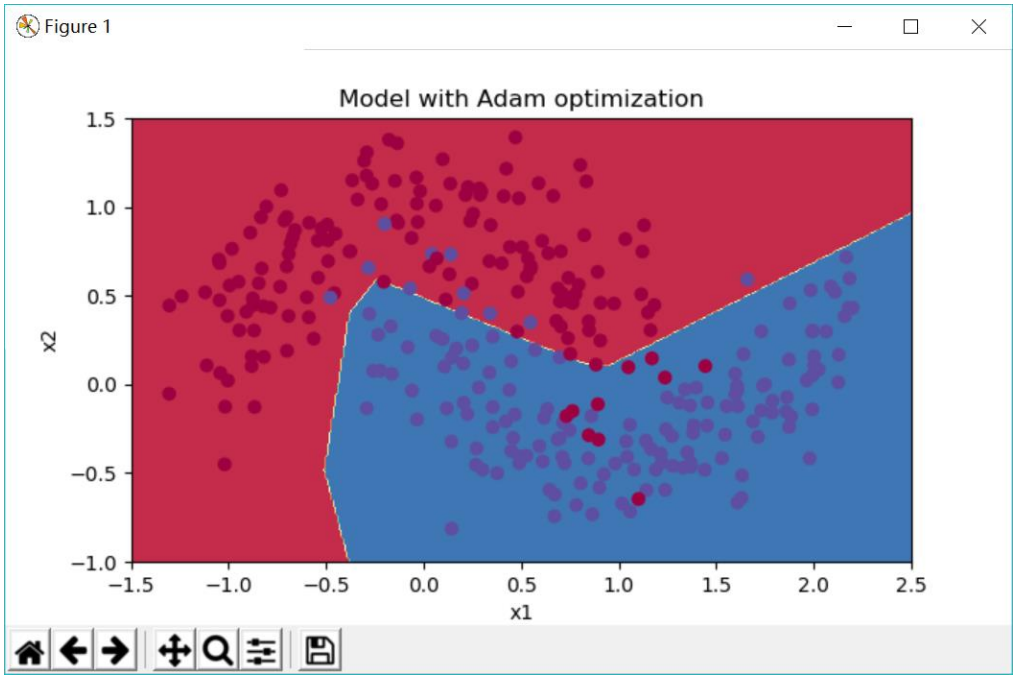

Figure 10. the classification result of Adam method

The accuracy is 0.93 and obviously the Adam method has a better convergence.

4. The regularization

The coefficient of regularization also matters and the greater regularized, the neural networks are more difficult to overfit, but also higher training error.

The common regularization method are: L1 norm regularization, L2 norm regularization, dropout, etc.L1 norm creates a sparse weights matrix. L2 norm generate the weights decadency and dropout method only train a part of neurons in one iteration and increases the robustness.

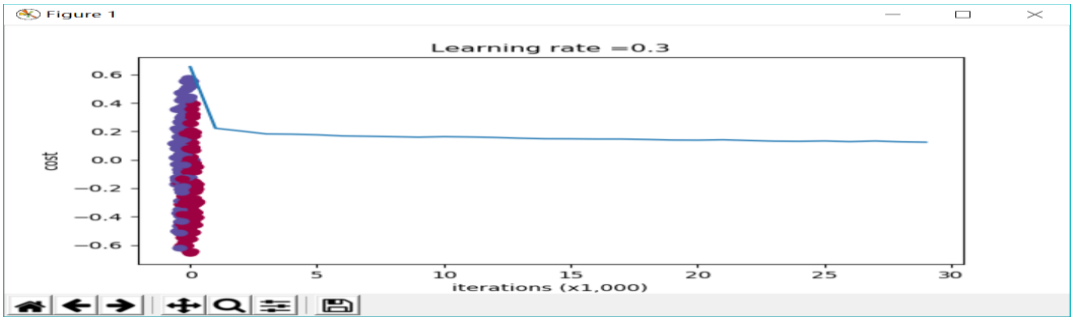

Figure 11. the loss decadency of unregularized model

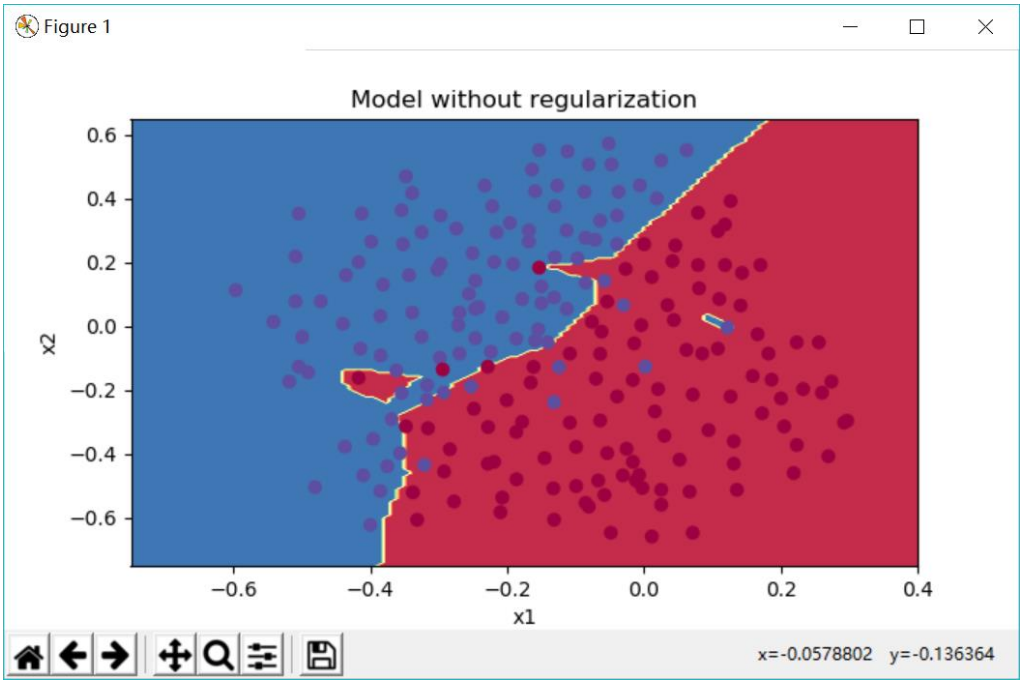

Figure 12. the result of best check ball point model without regularization

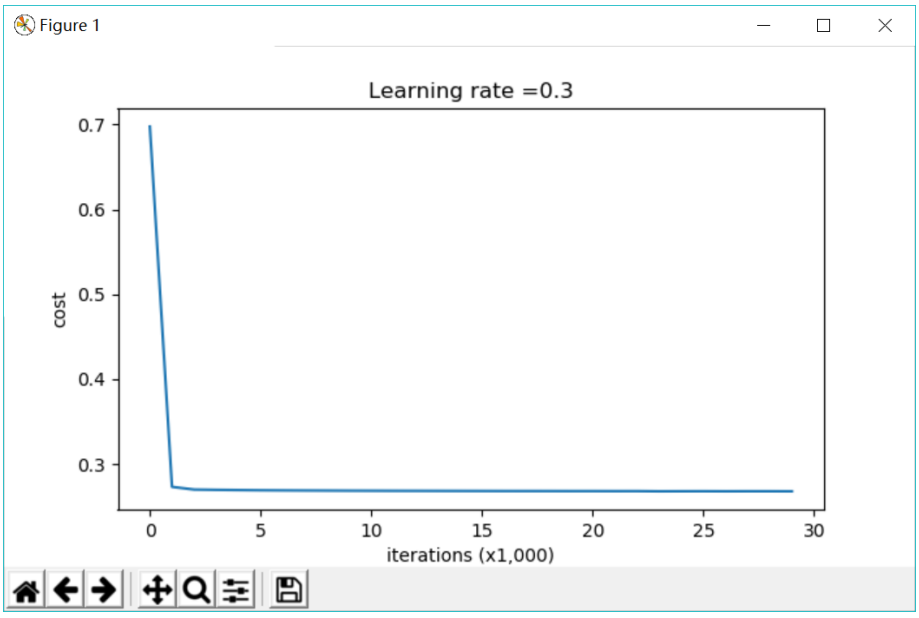

Figure 13. the loss decadency of L2 regularization

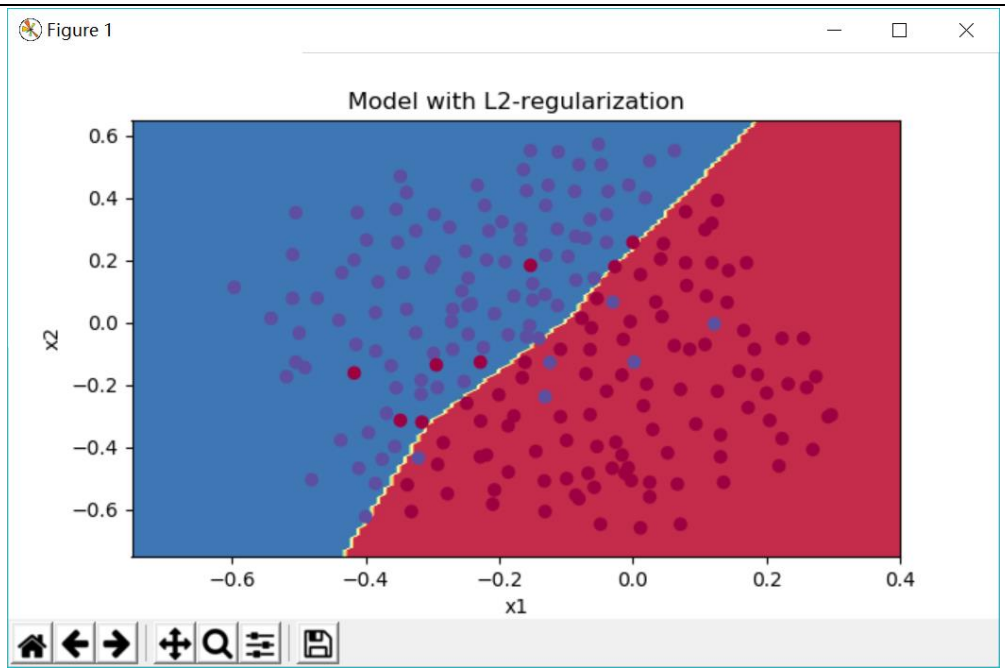

Figure 13. the result of best check ball point model with L2 regularization

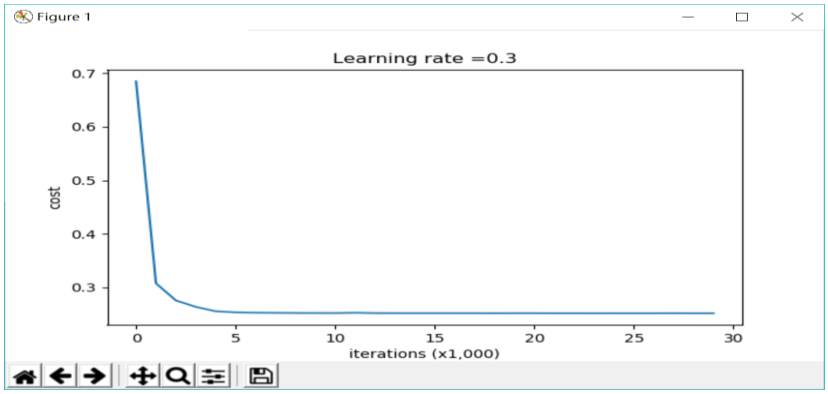

Figure 14. the loss decadency of dropout

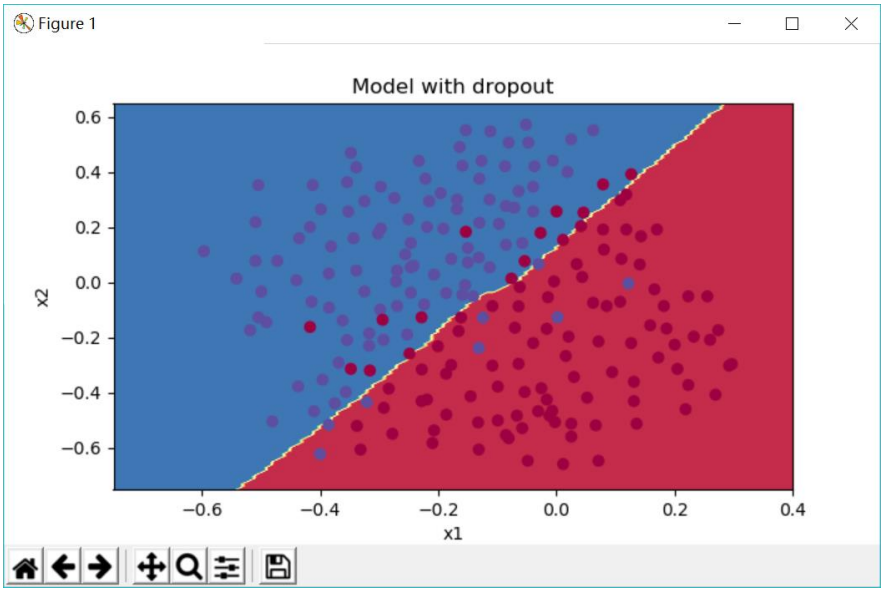

Figure 15. the result of dropout method

5. I used 192 types of combination of parameters to test their effects. The best parameters on the neural network are:

```
lr=0.0001, lrdecay=0.99, batch_size=256, reg=0.25
```

After 3000 iterations, the validation accuracy is 0.546, and the test accuracy is 0.539.

就实验过程中遇到和出现的问题，你是如何解决和处理的，自拟 1－3 道问答题：

The biggest problem is the calculation speed of my laptop and the convergence speed of the gradient descent. I use the Adam gradient descent to speed up the convergence and reduce the iterations. Then the total efficiency took a big stride