训练调参数过程

1. 要现在training data上结果是好的

在keras\_minist上的结果是0.1135

1. 修改loss 函数 改为交叉熵 三层网络训练 Accuracy: 0.8966

# mse mean\_squared\_error 均方误差损失 在分类问题结果并不好

1. train的batch\_size 改为10000 Accuracy: 0.113499999046
2. train的batch\_size 改为1 gpu无法几乎无作用
3. 改成10层时，梯度卡住了 Accuracy: 0.1047
4. 更改activation function sigmoid全部更改为relu

10000/10000 [==============================] - 0s 42us/step

Train Accuracy: 0.997300028801

10000/10000 [==============================] - 0s 37us/step

Test Accuracy: 0.952899992466

1. normal 对结果会有很大影响

# normalize 0-1之间 通常图像是0-255表示   
x\_train, x\_test = x\_train / 255, x\_test / 255

1. optimizer 由SGD 更改为adam

收敛的程度是差不多的与SGD比，但上升的速度会快

10000/10000 [==============================] - 0s 50us/step

Train Accuracy: 0.999400019646

10000/10000 [==============================] - 0s 35us/step

Test Accuracy: 0.962100028992

1. 在test上添加noise 每一个pixel加normal的noise 这样train与test是不match的

x\_test = np.random.normal(x\_test)

10000/10000 [==============================] - 0s 42us/step

Train Accuracy: 1.0

10000/10000 [==============================] - 0s 42us/step

Test Accuracy: 0.491100013256

1. 在9的基础上，dropout 解决过拟合 training上会不好，在training上不好就不要dropout

model.add(Dropout(0.7))

10000/10000 [==============================] - 0s 49us/step

Train Accuracy: 0.993300020695

10000/10000 [==============================] - 0s 49us/step

Test Accuracy: 0.58109998703

<https://blog.csdn.net/yuxeaotao/article/details/79211271>