

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: hyperparameters tuning		学号: 201600181073
日期: 2019. 3. 23	班级: 智能 16	姓名: 唐超
Email:		
<p>实验目的:</p> <ul style="list-style-type: none">• Understand the basic Image Classification pipeline and the data-driven approach;• Get a basic understanding of performance improvements from using higher-level representations than raw pixels (e.g. color histograms, Histogram of Gradient (HOG) features)• Master basic neural network adjustment skills and try to improve deep neural networks		
<p>实验软件和硬件环境:</p> <p>Spyder&python3.6</p>		
<p>实验步骤: (不要求罗列完整源代码)</p> <ol style="list-style-type: none">1. 填写 initialization, regularization, optimization method, gradient checking 等 notebook 教程, 在填写完每处代码后将 run 的结果与答案比对验证代码的正确性。学习进一步提升神经网络的相关方法, 包括 break symmetry 的初始化方法 random initialization, he initialization, 避免过拟合的 L-2 regularization, dropout 方法, 优化方法 momentum, adam, 以及 gradient checking。2. 尝试用以上方法提升该神经网络并进一步调整超参, 提升图片分类的准确率。由于原来 neural_net.py 中已采用了 L-2 regularization, 这里采用了 he initialization 和 momentum 的方法以期在上次实验的基础上进一步加快收敛速度和提升分类准确率。 <p>He initialization:</p>		

```
def __init__(self, layers_dims):

    np.random.seed(3)
    self.params = {}
    L = len(layers_dims) - 1 # integer representing the number of layers

    for l in range(1, L + 1):
        self.params['W' + str(l)] = np.random.randn(layers_dims[l-1],
                                                    layers_dims[l])*np.sqrt(2./layers_dims[l])
        self.params['b' + str(l)] = np.zeros((1, layers_dims[l]))
```

Momentum 更新参数：

```
L = 3 # number of layers in the neural networks
# Momentum update for each parameter
beta = 0.9
learning_rate=0.01

for l in range(L):
    ### START CODE HERE ### (approx. 4 lines)
    # compute velocities
    v['dW' + str(l+1)] = beta * v['dW' + str(l+1)] - (1-beta)*grads['dW' + str(l+1)]
    v["db" + str(l+1)] = beta * v["db" + str(l+1)] - (1-beta)*grads["db" + str(l+1)]
    # update parameters
    p["W" + str(l+1)] = p["W" + str(l+1)] + learning_rate * v["dW" + str(l+1)]
    p["b" + str(l+1)] = p["b" + str(l+1)] + learning_rate * v["db" + str(l+1)]
```

结论分析与体会：

在上一次实验得到的“最优超参”附近进一步调整得到更高的准确率：

```
Training params: batch size: 300, hidden size: 1024, learning rate: 0.001,
reg strength: 0.0001
2 train_accuracy:0.47925678493561937 val_accuracy:0.48

Test accuracy 0.477
```

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：

在用 he initialization 时，直接在原来 initialization.py 的代码上改，W, b 的形状与 layer size 的关系没有重新调整，导致在矩阵运算时出现了很多问题，后来进行了调整。