

# 实验一实验报告

---

## 1.实验介绍：

本实验使用softmax回归实现MNIST的手写数字识别功能，主要工作是完成softmax分类器的代码实现。

## 2.代码分析：

softmax分类器函数代码如下：

```
def softmax_classifier(w, input, label, lamda):
    t=np.dot(input,w)
    p=np.exp(t)

    for i in range(p.shape[0]):
        p[i]=p[i]/np.sum(p[i])

    prediction=np.argmax(p, axis=1)
    loss=np.sum(np.log(p)*label)/-p.shape[0]+lamda/2*np.sum(W*W)
    gradient=lamda*W+ np.dot(np.transpose(input), (p-label))/p.shape[0]

    return loss, gradient, prediction
```

可以看到，该函数的输入为W(参数矩阵784\*10)，input(输入图片,为100 \* 784),label(标签，one-hot编码 100 \* 10),lamda(正则项系数，取0.2)

在该函数中，首先计算了input \* W,然后进行指数化操作，最后对于每行归一化处理，得到每一项的概率值，预测值就是对于当前概率最大对应的一项

然后根据公式计算loss和gradient，loss为Cross-Entropy注意这里的loss是增加了一项正则项的结果，因此计算出的gradient也需要增加相应项。

最后将计算出的loss,gradient,prediction返回。

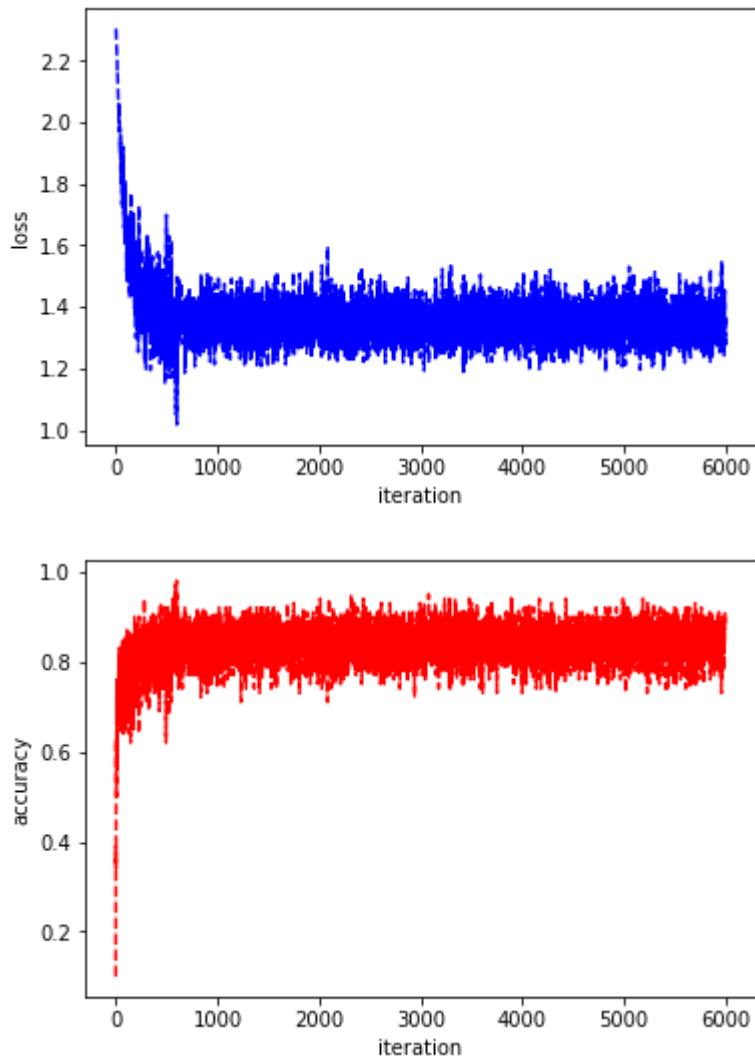
## 3.结果分析：

本实验采用了SGD的参数更新方法，对于超参数的选择如下：

batch size=100, epoch数=10, learning rate=0.01, lamda=0.2

test的准确率为：0.8488 与模型在训练集上的准确率相比差距不大，未出现明显的过拟合现象。

loss和accuracy的变化曲线如下：



可以看见，当迭代次数较少时准确率快速上升，当迭代约500次时基本达到收敛，剩余迭代次数内准确率基本在0.7到0.9之间震荡，可以看到，一层的softmax分类器效果并不特别理想。

此后，我在softmax层之前加入了两个卷积层和池化层，再次测试时发现模型准确率达到了97%以上，取得了较为理想的结果。可以看到，相对于简单的线性分类器，加入了非线性的卷积神经网络的分类器效果有了明显提升。由于以上内容不在此作业范围之内，且使用了tensorflow库内容，具体不多加叙述。

此外，为了改进softmax分类器的性能，我认为还可以多加一项额外的偏执量bias以应对输入的一些噪声与干扰量。