

Pj1

19037130069 王开阳

1. 训练

使用 **tanh** 函数作为激活函数，采用学习率下降策略，**L2** 正则化以及 **softmax** 损失函数。

```
if sum(yhat~=yvalid)/t<0.06
    stepSize=stepSize*0.5;
End
```

在结果接近收敛时，每轮迭代的学习率会进行一次下降。

```
[f,g,gb] = funObj(w,b,i);
w = w - stepSize * (g+lambda*w) ;
b = b - stepSize * gb ;
```

使用 **SGD** 对参数进行优化，加入 **L2** 正则化项，得到的迭代公式如上。

反向传播，**loss** 以及梯度的计算详见于 **MLPclassificationLoss.m**。

2. 参数查找

对学习率(**stepsize**)，隐藏层大小(**nhidden**)，正则化强度(**lambda**)进行参数查找。

对于上述三个参数，查找

$\text{stepsize} \in \{1e-2, 1e-3, 1e-4, 1e-5\}; \lambda \in \{1e-2, 1e-3, 1e-4, 1e-5\}; \text{nhidden} \in \{100, 200, 300\}$

得到部分结果如下：

200	1e-2	1e-3	1e-4	1e-5
1e-2	0.052	0.117	0.129	0.155
1e-3	0.163	0.234	0.261	0.221
1e-4	0.243	0.431	0.445	0.424
1e-5	0.852	0.834	0.805	0.791

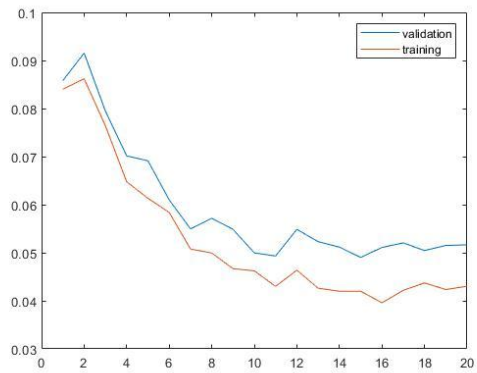
选择在 **stepsize=1e-2**，**lambda=1e-2** 附近继续进行搜索

最终选择 **stepsize=1e-2**，**lambda=5e-3**，**nhidden=200**

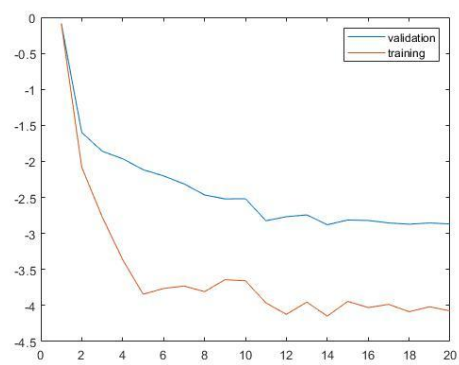
3.测试

按照上述最优参数选择，进行测试。

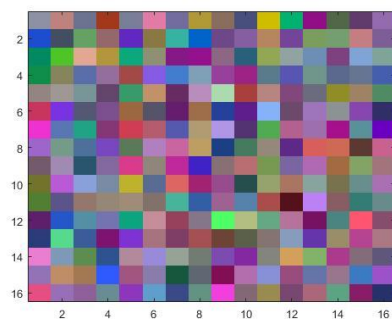
loss 曲线(横坐标为迭代轮次，每轮 5000 次):



log(error)曲线:



网络参数可视化:



把 w 参数 reshape 成 $(256, n_{hidden})$ 大小, 进行 PCA 选出前三列主成分作为图片的 RGB 值, 输出图像如上。

4.链接:

github: <https://github.com/yaozhong0930/cv-pj1>

链接: <https://pan.baidu.com/s/1kbSZ8ivtS7RdL6NWqaAelw?pwd=wkst>

提取码: wkst