

机器学习人脸识别课程设计

陈赞华

1 研究问题:

仔细研究ORL数据库（Olivetti Research Laboratory in Combridge）的人脸图像ORL_32x32，对图像的特征和复杂性进行认真分析，使用机器学习中人工神经网络模型或支持向量机，设计人脸识别系统，并对系统做出评估。

2 原理

2.1 累积BP算法

```
for k =1:size(X,1)
    x=X(k,:);
    ah=Theta1(:,2:end)*x';
    b=sigmoid(ah+Theta1(:,1));
    beta=Theta2(:,2:end)*b;
    Y=sigmoid(beta+Theta2(:,1));

    g=-(y(k,:)-Y');
    e=b.*(1-b).*(g*Theta2(:,2:end))';
    %Theta2_grad(:,2:end)=g'*b';
    %Theta2_grad(:,1)=g';
    temp2=g'*[1 b'];
    temp2all=temp2all+temp2;
    %Theta2=Theta2+temp2;
    %Theta1_grad(:,2:end)=e*x;
    %Theta1_grad(:,1)=e;
    temp1=e*[1 x];
    temp1all=temp1all+temp1;
    %Theta1=Theta1+temp1;
%     Theta2=Theta2+Theta2_grad;
%     Theta1=Theta1+Theta1_grad;

end

Theta2_grad=1/m*temp2all+(lambda/m)*(Theta2);
Theta1_grad=1/m*temp1all+(lambda/m)*(Theta1);
Theta1_grad(:,1) = Theta1_grad(:,1) - ((lambda/m)*(Theta1(:,1)));
Theta2_grad(:,1) = Theta2_grad(:,1) - ((lambda/m)*(Theta2(:,1)));
```

2.2 正则化

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \sum_{k=1}^K \left[-y_k^{(i)} \log((h_{\theta}(x^{(i)}))_k) - (1 - y_k^{(i)}) \log(1 - (h_{\theta}(x^{(i)}))_k) \right] + \frac{\lambda}{2m} \left[\sum_{j=1}^{25} \sum_{k=1}^{400} (\Theta_{j,k}^{(1)})^2 + \sum_{j=1}^{10} \sum_{k=1}^{25} (\Theta_{j,k}^{(2)})^2 \right].$$

```
h = eye(num_labels);
y = h(y,:);
for k =1:size(X,1)
    x=X(k,:);
    ah=Theta1(:,2:end)*x';
    b=sigmoid(ah+Theta1(:,1));
    beta=Theta2(:,2:end)*b;
    Y=sigmoid(beta+Theta2(:,1));
    J=J +(-y(k,:)*log(Y) - (1-y(k,:)) * log(1-Y));
end
reg=lambda/(2*m)*( sum(sum(Theta1(:,2:end).^2))+...
    sum(sum(Theta2(:,2:end).^2)) );
J=1/m*J+reg;
```

3 二分类问题:

3.1 数据预处理

以是否戴眼镜作为类标签，对数据进行预处理，同时确定训练数据集、测试数据集；

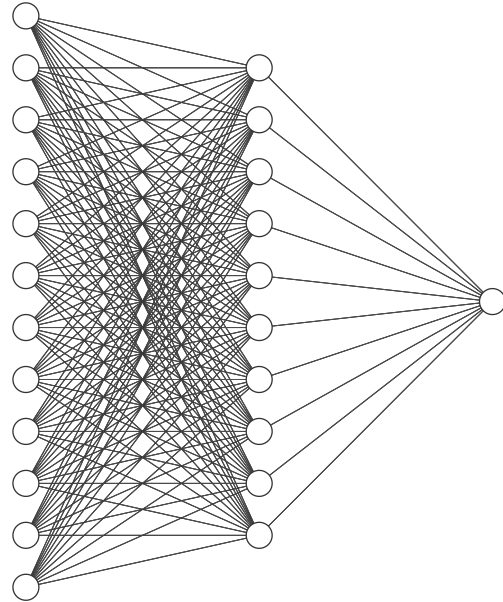
训练集测试集划分：

首先将OLR数据集打乱，然后将前300 个元素做训练集，后100个左测试集。

% 使用留出法将数据集分为训练集和测试集。

```
m = length (y);
ind = randperm(m);
X = X(ind,:);
y = y(ind);
X_tra = X(1:300,:);
X_val = X(301:400,:);
y_tra = y(1:300);
y_val = y(301:400);
```

3.2 神经网络模型建立;



Input layer : 1024 Hiden layer :64 Output layer :2

3.3 模型训练:

使用测试数据集对所建模型进行评估，（混淆矩阵，测试精度表，查准率，查全率和F1表）；

改变神经网络训练过程中学习率，神经元个数（等比较不同网络得到的测试精度，查准率，查全率和F1值。探索学习率和神经元个数对于网络性能的影响。（附不同模型得到的混淆矩阵和精度比较表、查准率比较表、查全率比较表以及F1值比较表）。

Input layer : 1024 Hiden layer :5 Output layer :2 iters = 500
 $\lambda = [2 : 2 : 40]$

lambda	真正 例	假正 例	假反 例	真反 例	查准率	查全率	F1	准确 率
2	23	7	10	60	0.766666667	0.696969697	0.73015873	0.83
4	24	6	9	61	0.8	0.727272727	0.761904762	0.85
6	0	0	33	67	NaN	0	NaN	0.67
8	22	9	11	58	0.709677419	0.666666667	0.6875	0.8
10	0	0	33	67	NaN	0	NaN	0.67
12	16	7	17	60	0.695652174	0.484848485	0.571428571	0.76
14	0	0	33	67	NaN	0	NaN	0.67
16	16	8	17	59	0.666666667	0.484848485	0.561403509	0.75
18	0	0	33	67	NaN	0	NaN	0.67
20	0	0	33	67	NaN	0	NaN	0.67
22	0	0	33	67	NaN	0	NaN	0.67
24	0	0	33	67	NaN	0	NaN	0.67
26	22	4	11	63	0.846153846	0.666666667	0.745762712	0.85
28	0	0	33	67	NaN	0	NaN	0.67
30	0	0	33	67	NaN	0	NaN	0.67
32	17	4	16	63	0.80952381	0.515151515	0.62962963	0.8
34	21	3	12	64	0.875	0.636363636	0.736842105	0.85
36	0	0	33	67	NaN	0	NaN	0.67
38	0	0	33	67	NaN	0	NaN	0.67
40	0	0	33	67	NaN	0	NaN	0.67

Input layer : 1024 Hiden layer :10 Output layer :2 iters = 500
lambda = [2 : 2 : 40]

lambda	真正 例	假正 例	假反 例	真反 例	P	R	F1	准确 率
2	18	9	8	65	0.666666667	0.692307692	0.679245283	0.83
4	21	5	5	69	0.807692308	0.807692308	0.807692308	0.9
6	22	10	4	64	0.6875	0.846153846	0.75862069	0.86
8	20	11	6	63	0.64516129	0.769230769	0.701754386	0.83
10	16	5	10	69	0.761904762	0.615384615	0.680851064	0.85
12	22	5	4	69	0.814814815	0.846153846	0.830188679	0.91
14	19	9	7	65	0.678571429	0.730769231	0.703703704	0.84
16	18	0	8	74	1	0.692307692	0.818181818	0.92
18	16	2	10	72	0.888888889	0.615384615	0.727272727	0.88
20	18	3	8	71	0.857142857	0.692307692	0.765957447	0.89
22	15	1	11	73	0.9375	0.576923077	0.714285714	0.88
24	0	0	26	74	NaN	0	NaN	0.74
26	19	4	7	70	0.826086957	0.730769231	0.775510204	0.89
28	20	3	6	71	0.869565217	0.769230769	0.816326531	0.91
30	16	3	10	71	0.842105263	0.615384615	0.711111111	0.87
32	0	0	26	74	NaN	0	NaN	0.74
34	18	1	8	73	0.947368421	0.692307692	0.8	0.91
36	0	0	26	74	NaN	0	NaN	0.74
38	18	1	8	73	0.947368421	0.692307692	0.8	0.91
40	0	0	26	74	NaN	0	NaN	0.74

Input layer : 1024 Hidden layer :20 Output layer :2 iters = 500
lambda = [2 : 2 : 40]

lambda	真正 例	假正 例	假反 例	真反 例	查准率	查全率	F1	准确 率
2	23	3	14	60	0.884615385	0.621621622	0.73015873	0.83
4	27	2	10	61	0.931034483	0.72972973	0.818181818	0.88
6	21	2	16	61	0.913043478	0.567567568	0.7	0.82
8	20	1	17	62	0.952380952	0.540540541	0.689655172	0.82
10	12	1	25	62	0.923076923	0.324324324	0.48	0.74
12	22	1	15	62	0.956521739	0.594594595	0.733333333	0.84
14	21	0	16	63	1	0.567567568	0.724137931	0.84
16	22	1	15	62	0.956521739	0.594594595	0.733333333	0.84
18	18	2	19	61	0.9	0.486486486	0.631578947	0.79
20	0	0	37	63	NaN	0	NaN	0.63
22	27	2	10	61	0.931034483	0.72972973	0.818181818	0.88
24	19	0	18	63	1	0.513513514	0.678571429	0.82
26	19	0	18	63	1	0.513513514	0.678571429	0.82
28	26	1	11	62	0.962962963	0.702702703	0.8125	0.88
30	0	0	37	63	NaN	0	NaN	0.63
32	0	0	37	63	NaN	0	NaN	0.63
34	19	2	18	61	0.904761905	0.513513514	0.655172414	0.8
36	0	0	37	63	NaN	0	NaN	0.63
38	0	0	37	63	NaN	0	NaN	0.63
40	0	0	37	63	NaN	0	NaN	0.63

Input layer : 1024 Hidden layer :30 Output layer :2 iters = 500
lambda = [2 : 2 : 40]

lambda	真正 例	假正 例	假反 例	真反 例	查准率	查全率	F1	准确 率
2	17	9	7	67	0.653846154	0.708333333	0.68	0.84
4	19	16	5	60	0.542857143	0.791666667	0.644067797	0.79
6	14	13	10	63	0.518518519	0.583333333	0.549019608	0.77
8	18	14	6	62	0.5625	0.75	0.642857143	0.8
10	17	6	7	70	0.739130435	0.708333333	0.723404255	0.87
12	14	8	10	68	0.636363636	0.583333333	0.608695652	0.82
14	18	12	6	64	0.6	0.75	0.666666667	0.82
16	17	7	7	69	0.708333333	0.708333333	0.708333333	0.86
18	15	6	9	70	0.714285714	0.625	0.666666667	0.85
20	15	6	9	70	0.714285714	0.625	0.666666667	0.85
22	12	10	12	66	0.545454545	0.5	0.52173913	0.78
24	18	6	6	70	0.75	0.75	0.75	0.88
26	16	5	8	71	0.761904762	0.666666667	0.711111111	0.87
28	18	7	6	69	0.72	0.75	0.734693878	0.87
30	17	8	7	68	0.68	0.708333333	0.693877551	0.85
32	18	8	6	68	0.692307692	0.75	0.72	0.86
34	14	6	10	70	0.7	0.583333333	0.636363636	0.84
36	13	7	11	69	0.65	0.541666667	0.590909091	0.82
38	17	4	7	72	0.80952381	0.708333333	0.755555556	0.89
40	17	6	7	70	0.739130435	0.708333333	0.723404255	0.87

Input layer : 1024
lambda = [2 : 2 : 40]

Hidden layer :40 Output layer :2 iters = 500

lambda	真正 例	假正 例	假反 例	真反 例	查准率	查全率	F1	准确 率
2	7	11	24	58	0.388888889	0.225806452	0.285714286	0.65
4	16	3	15	66	0.842105263	0.516129032	0.64	0.82
6	0	0	31	69	NaN	0	NaN	0.69
8	14	6	17	63	0.7	0.451612903	0.549019608	0.77
10	18	7	13	62	0.72	0.580645161	0.642857143	0.8
12	19	5	12	64	0.791666667	0.612903226	0.690909091	0.83
14	16	4	15	65	0.8	0.516129032	0.62745098	0.81
16	19	5	12	64	0.791666667	0.612903226	0.690909091	0.83
18	16	5	15	64	0.761904762	0.516129032	0.615384615	0.8
20	18	4	13	65	0.818181818	0.580645161	0.679245283	0.83
22	0	0	31	69	NaN	0	NaN	0.69
24	13	3	18	66	0.8125	0.419354839	0.553191489	0.79
26	13	2	18	67	0.866666667	0.419354839	0.565217391	0.8
28	0	0	31	69	NaN	0	NaN	0.69
30	8	2	23	67	0.8	0.258064516	0.390243902	0.75
32	17	6	14	63	0.739130435	0.548387097	0.62962963	0.8
34	13	2	18	67	0.866666667	0.419354839	0.565217391	0.8
36	17	5	14	64	0.772727273	0.548387097	0.641509434	0.81
38	18	5	13	64	0.782608696	0.580645161	0.666666667	0.82
40	14	2	17	67	0.875	0.451612903	0.595744681	0.81

Input layer : 1024

Hidden layer :50

Output layer :2

iters = 500

lambda = [2 : 2 : 40]

lambda	真正例	假正例	假反例	真反例	查准率	查全率	F1	准确率
2	19	6	19	56	0.76	0.5	0.603174603	0.75
4	8	4	30	58	0.666666667	0.210526316	0.32	0.66
6	13	3	25	59	0.8125	0.342105263	0.481481481	0.72
8	16	3	22	59	0.842105263	0.421052632	0.561403509	0.75
10	17	5	21	57	0.772727273	0.447368421	0.566666667	0.74
12	23	4	15	58	0.851851852	0.605263158	0.707692308	0.81
14	19	5	19	57	0.791666667	0.5	0.612903226	0.76
16	0	0	38	62	NaN	0	NaN	0.62
18	17	4	21	58	0.80952381	0.447368421	0.576271186	0.75
20	23	5	15	57	0.821428571	0.605263158	0.696969697	0.8
22	13	5	25	57	0.722222222	0.342105263	0.464285714	0.7
24	16	2	22	60	0.888888889	0.421052632	0.571428571	0.76
26	21	4	17	58	0.84	0.552631579	0.666666667	0.79
28	18	3	20	59	0.857142857	0.473684211	0.610169492	0.77
30	18	4	20	58	0.818181818	0.473684211	0.6	0.76
32	15	3	23	59	0.833333333	0.394736842	0.535714286	0.74
34	11	2	27	60	0.846153846	0.289473684	0.431372549	0.71
36	17	4	21	58	0.80952381	0.447368421	0.576271186	0.75
38	17	3	21	59	0.85	0.447368421	0.586206897	0.76
40	0	0	38	62	NaN	0	NaN	0.62

Input layer : 1024
lambda = [2 : 2 : 40]

Hidden layer :60 Output layer :2 iters = 500

lambda	真正例	假正例	假反例	真反例	查准率	查全率	F1	准确率
2	8	6	24	62	0.571428571	0.25	0.347826087	0.7
4	24	6	8	62	0.8	0.75	0.774193548	0.86
6	2	0	30	68	1	0.0625	0.117647059	0.7
8	23	6	9	62	0.793103448	0.71875	0.754098361	0.85
10	18	3	14	65	0.857142857	0.5625	0.679245283	0.83
12	24	6	8	62	0.8	0.75	0.774193548	0.86
14	21	6	11	62	0.777777778	0.65625	0.711864407	0.83
16	23	6	9	62	0.793103448	0.71875	0.754098361	0.85
18	23	5	9	63	0.821428571	0.71875	0.766666667	0.86
20	19	5	13	63	0.791666667	0.59375	0.678571429	0.82
22	21	3	11	65	0.875	0.65625	0.75	0.86
24	0	0	32	68	NaN	0	NaN	0.68
26	18	4	14	64	0.818181818	0.5625	0.666666667	0.82
28	21	5	11	63	0.807692308	0.65625	0.724137931	0.84
30	16	1	16	67	0.941176471	0.5	0.653061224	0.83
32	0	0	32	68	NaN	0	NaN	0.68
34	24	7	8	61	0.774193548	0.75	0.761904762	0.85
36	0	0	32	68	NaN	0	NaN	0.68
38	0	0	32	68	NaN	0	NaN	0.68
40	23	7	9	61	0.766666667	0.71875	0.741935484	0.84

Input layer : 1024

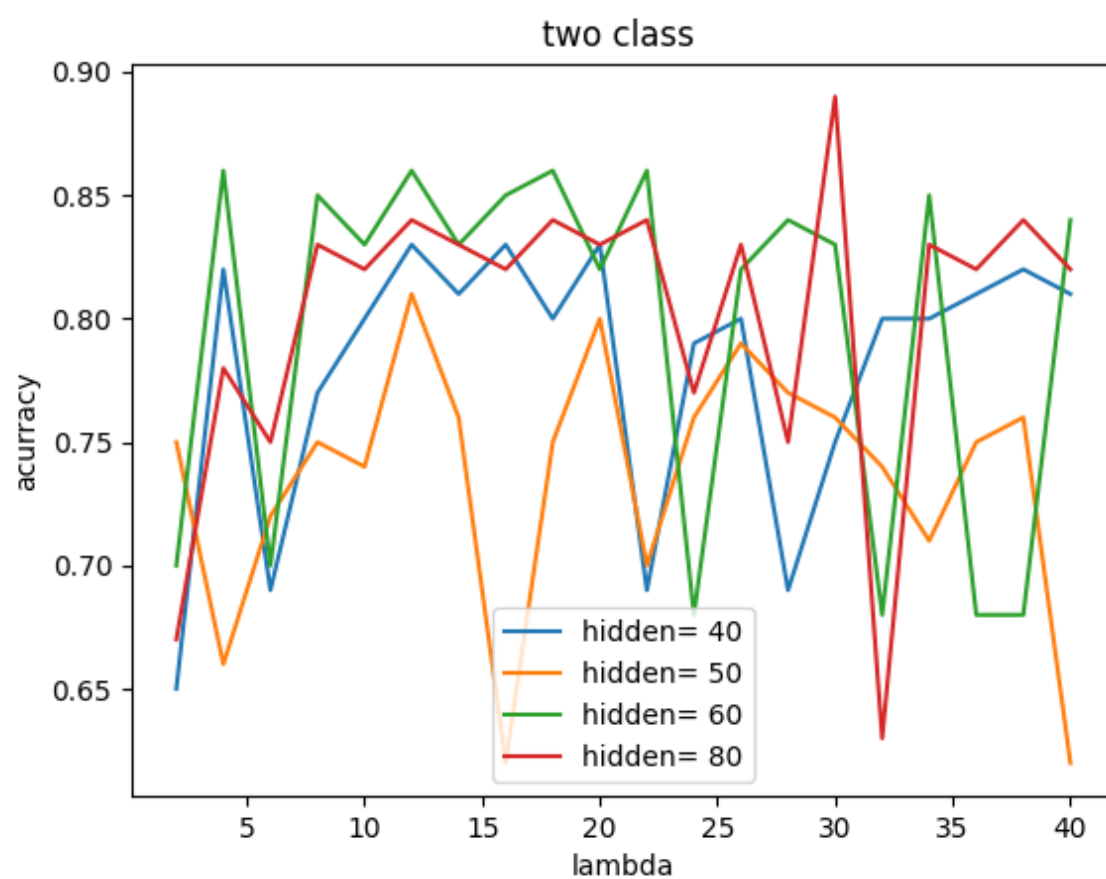
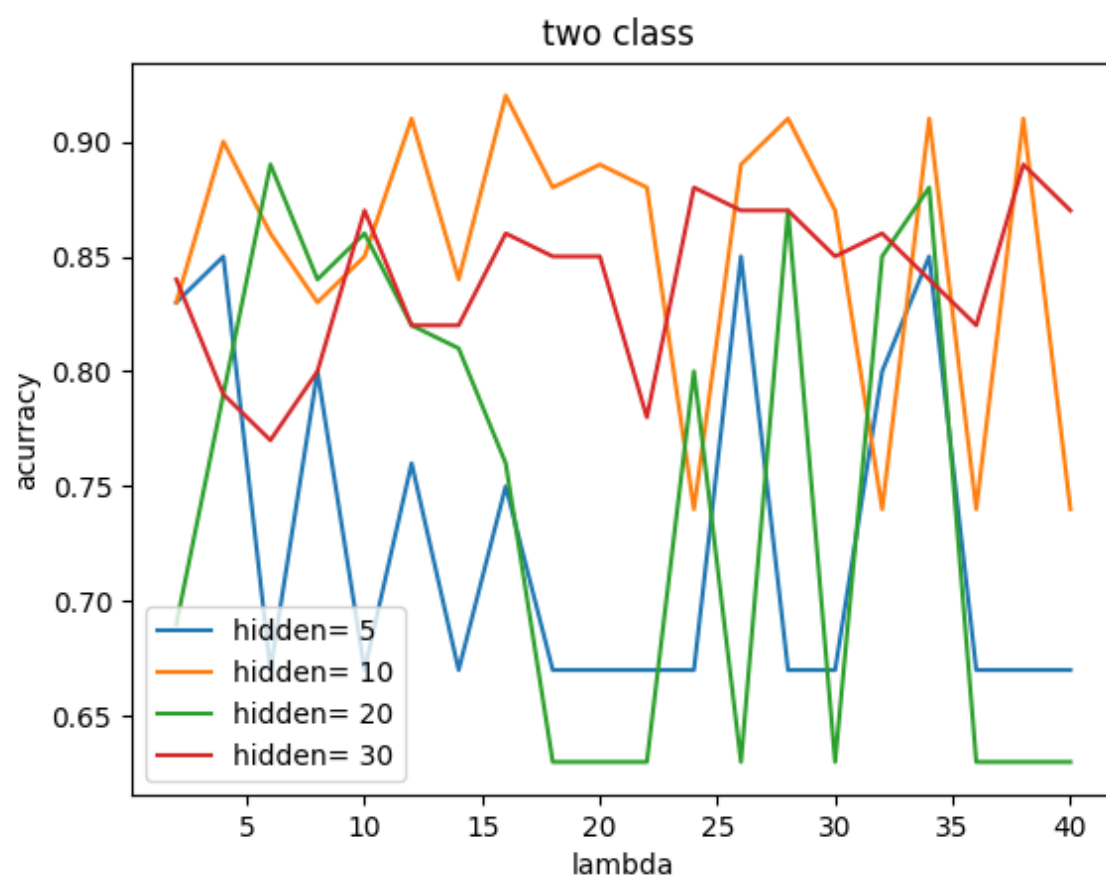
Hidden layer :80

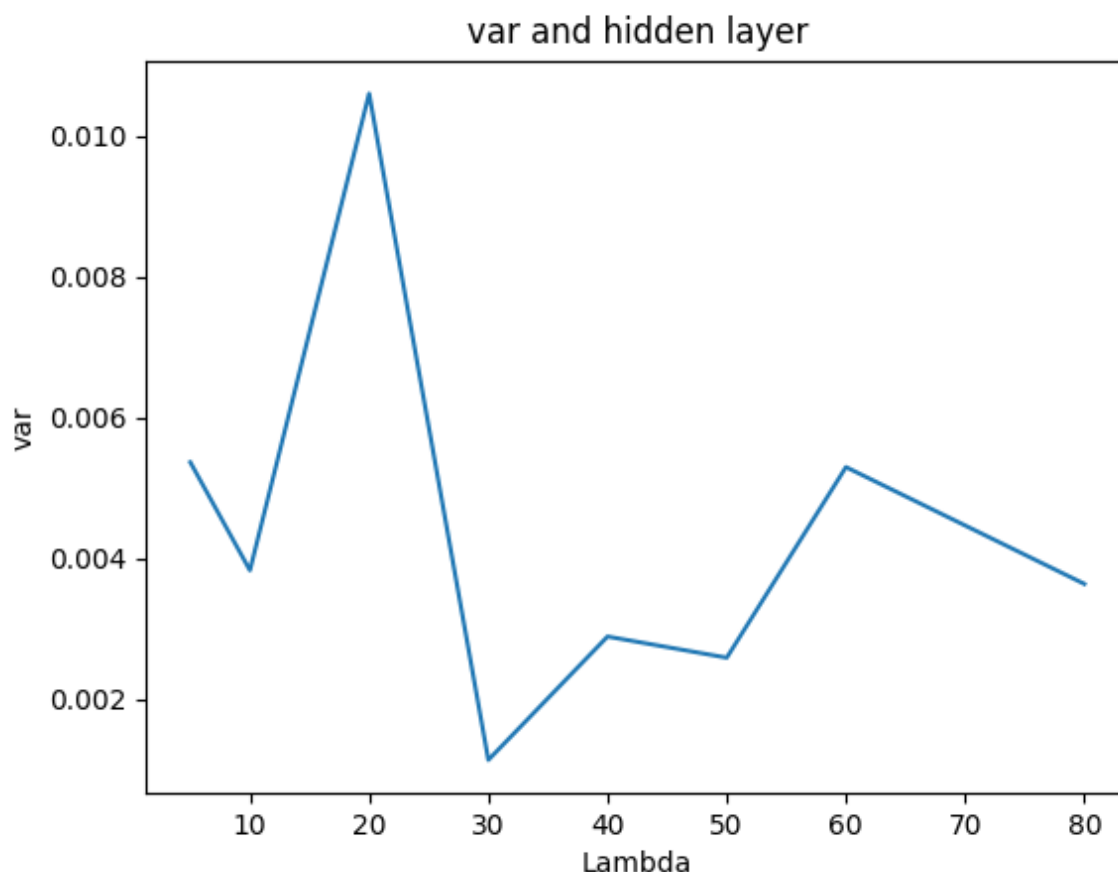
Output layer :2

iters = 500

lambda = [2 : 2 : 40]

lambda	真正 例	假正 例	假反 例	真反 例	查准率	查全率	F1	准确 率
2	12	8	25	55	0.6	0.324324324	0.421052632	0.67
4	24	9	13	54	0.727272727	0.648648649	0.685714286	0.78
6	20	8	17	55	0.714285714	0.540540541	0.615384615	0.75
8	25	5	12	58	0.833333333	0.675675676	0.746268657	0.83
10	24	5	13	58	0.827586207	0.648648649	0.727272727	0.82
12	26	5	11	58	0.838709677	0.702702703	0.764705882	0.84
14	27	7	10	56	0.794117647	0.72972973	0.76056338	0.83
16	25	6	12	57	0.806451613	0.675675676	0.735294118	0.82
18	28	7	9	56	0.8	0.756756757	0.777777778	0.84
20	26	6	11	57	0.8125	0.702702703	0.753623188	0.83
22	27	6	10	57	0.818181818	0.72972973	0.771428571	0.84
24	19	5	18	58	0.791666667	0.513513514	0.62295082	0.77
26	26	6	11	57	0.8125	0.702702703	0.753623188	0.83
28	15	3	22	60	0.833333333	0.405405405	0.545454545	0.75
30	31	5	6	58	0.861111111	0.837837838	0.849315068	0.89
32	0	0	37	63	NaN	0	NaN	0.63
34	27	7	10	56	0.794117647	0.72972973	0.76056338	0.83
36	25	6	12	57	0.806451613	0.675675676	0.735294118	0.82
38	28	7	9	56	0.8	0.756756757	0.777777778	0.84
40	26	7	11	56	0.787878788	0.702702703	0.742857143	0.82





3.4 实验结果分析：

从准确率图表，以及方差图表中，可以观察到，当**Input layer : 1024 Output layer :2 iters = 500**条件不变时，隐层从5 增加到30时，模型性能不稳定，方差很大，无法通过调整参数来实现泛化性能的提升，但是方差呈下降趋势。隐层从30 增加到80 时，模型性能不稳定，方差很大，但是方差成上升趋势。在隐层为30时，以及当前迭代次数下，模型训练程度较好，所以模型性能稳定。

3.5 理论解释：

在**Input layer : 1024 Output layer :2 iters = 500**条件不变时，隐层数量为5，10，20 时，由于隐层神经元数量较少，模型学习能力低，所以模型整体表现为极度不稳定，准确率时而很好，时而很差。

当隐层神经元数量在30附近时，网络开始变得较为复杂，模型学习能力替身，此时模型在不同的lambda出表现较为稳定，且准确率较高。

当隐层神经元进一步增加，网络越来越复杂，模型学习能力强，但是由于迭代次数限制，导致网络没有训练好，即有偏差。

3.6 实验猜想：

此时为了验证对实验数据的理论解释，我采取对当前还没有测量的数据进行预测。

1. 选取隐层神经元30左边的一组数据，对网络进行训练，并查看该隐层神经元数量下的模型的性能。

即：

预测为： 实验结果波动性大，方差比30附近的要大。

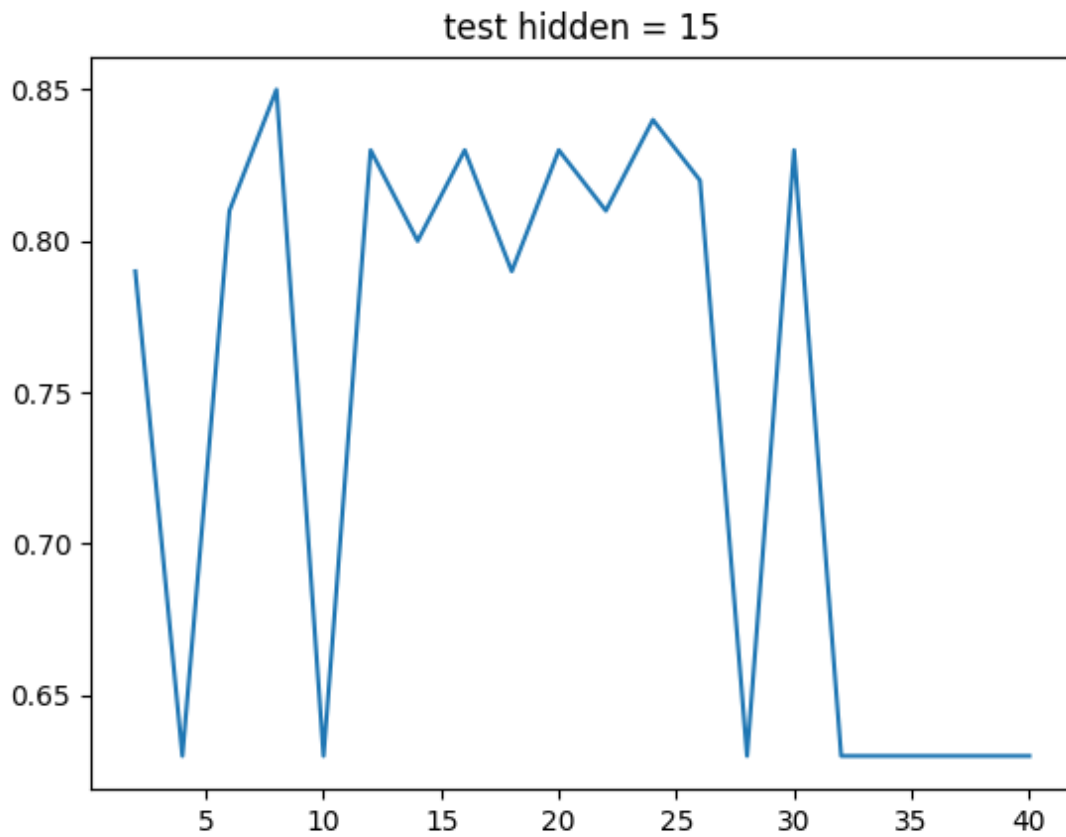
2. 选取隐层神经元为30右边的一组数据，对网络进行训练，但此时我们改变迭代次数，对波动最大的一组进行训练，看能否为高偏差而引起的模型泛化性能不好。

3.7 验证实验：

1. **Input layer : 1024** **Hidden layer :15** **Output layer :2** **iters = 500**
 $\lambda = [2 : 2 : 40]$

lambda	真正 例	假正 例	假反 例	真反 例	查准率	查全率	F1	准确 率
2	24	8	13	55	0.75	0.648648649	0.695652174	0.79
4	3	3	34	60	0.5	0.081081081	0.139534884	0.63
6	19	1	18	62	0.95	0.513513514	0.666666667	0.81
8	23	1	14	62	0.958333333	0.621621622	0.754098361	0.85
10	0	0	37	63	NaN	0	NaN	0.63
12	22	2	15	61	0.916666667	0.594594595	0.721311475	0.83
14	18	1	19	62	0.947368421	0.486486486	0.642857143	0.8
16	21	1	16	62	0.954545455	0.567567568	0.711864407	0.83
18	19	3	18	60	0.863636364	0.513513514	0.644067797	0.79
20	23	3	14	60	0.884615385	0.621621622	0.73015873	0.83
22	20	2	17	61	0.909090909	0.540540541	0.677966102	0.81
24	22	1	15	62	0.956521739	0.594594595	0.733333333	0.84
26	20	1	17	62	0.952380952	0.540540541	0.689655172	0.82
28	0	0	37	63	NaN	0	NaN	0.63
30	21	1	16	62	0.954545455	0.567567568	0.711864407	0.83
32	0	0	37	63	NaN	0	NaN	0.63
34	0	0	37	63	NaN	0	NaN	0.63
36	0	0	37	63	NaN	0	NaN	0.63
38	0	0	37	63	NaN	0	NaN	0.63
40	0	0	37	63	NaN	0	NaN	0.63

var = 0.008792749999999998



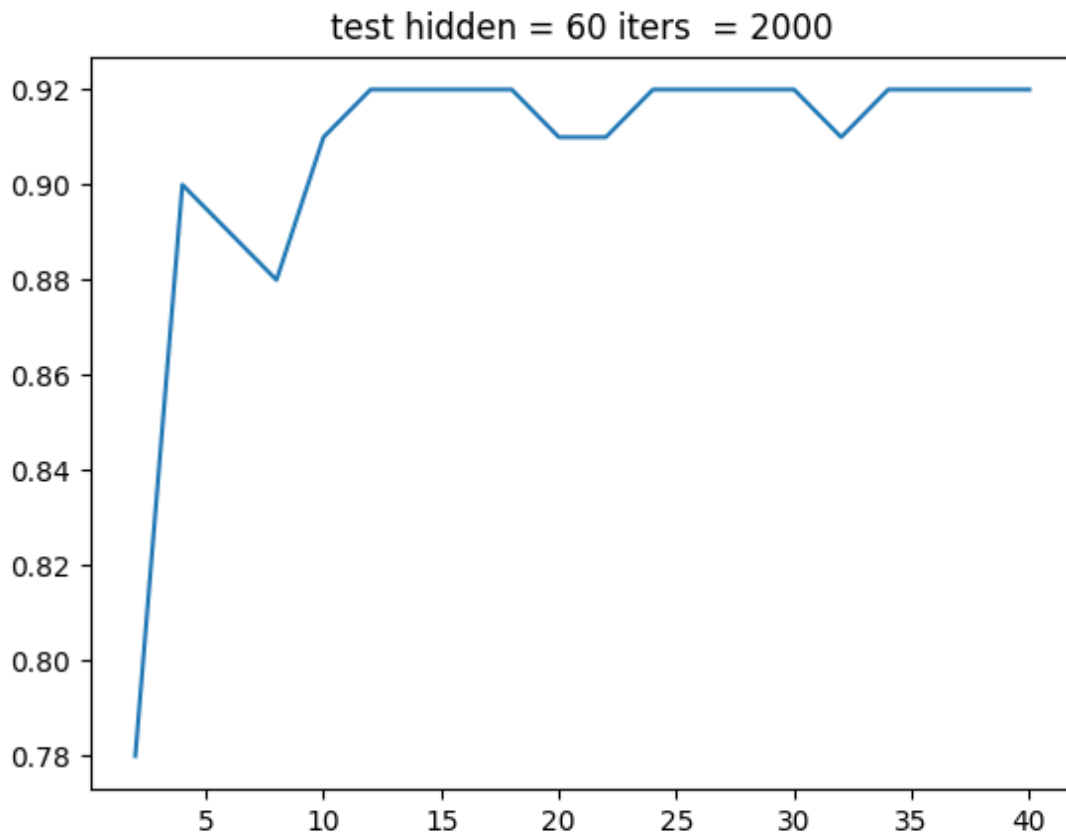
该方差明显大于隐层神经元数量为30的模型的方差。从模型在测试数据集上的表现还可以发现该模型直接将所有的样本判定为反例来，且该模型在不同lambda下数据波动剧烈。与预测结果符合，假设得到验证。

2. **Input layer : 1024**
 $\lambda = [2 : 2 : 40]$

Hidden layer :60 Output layer :2 iters = 2000

lambda	真正 例	假正 例	假反 例	真反 例	查准率	查全率	F1	准确 率
2	17	11	11	61	0.607142857	0.607142857	0.607142857	0.78
4	24	6	4	66	0.8	0.857142857	0.827586207	0.9
6	23	6	5	66	0.793103448	0.821428571	0.807017544	0.89
8	24	8	4	64	0.75	0.857142857	0.8	0.88
10	25	6	3	66	0.806451613	0.892857143	0.847457627	0.91
12	25	5	3	67	0.833333333	0.892857143	0.862068966	0.92
14	25	5	3	67	0.833333333	0.892857143	0.862068966	0.92
16	25	5	3	67	0.833333333	0.892857143	0.862068966	0.92
18	25	5	3	67	0.833333333	0.892857143	0.862068966	0.92
20	25	6	3	66	0.806451613	0.892857143	0.847457627	0.91
22	25	6	3	66	0.806451613	0.892857143	0.847457627	0.91
24	25	5	3	67	0.833333333	0.892857143	0.862068966	0.92
26	25	5	3	67	0.833333333	0.892857143	0.862068966	0.92
28	25	5	3	67	0.833333333	0.892857143	0.862068966	0.92
30	25	5	3	67	0.833333333	0.892857143	0.862068966	0.92
32	24	5	4	67	0.827586207	0.857142857	0.842105263	0.91
34	25	5	3	67	0.833333333	0.892857143	0.862068966	0.92
36	25	5	3	67	0.833333333	0.892857143	0.862068966	0.92
38	25	5	3	67	0.833333333	0.892857143	0.862068966	0.92
40	25	5	3	67	0.833333333	0.892857143	0.862068966	0.92

var : 0.00096275000000000001



该模型的方差为 $\text{var} = 0.0009627500000000001$ ，且模型性能及其稳定都很好，也验证了预测是的猜想，随着模型中隐层神经元的数量的上升，模型学习性能上升，当迭代次数为500次时模型结果变化剧烈，这是因为发生了欠拟合。

3.8 实验结论：

1. 在三层神经网络模型中，当隐层神经元数量增加时，模型的学习性能上升，同时所需要的迭代次数也更多，因为此时模型有更多的参数需要去学习。当隐层神经元过少时，模型学习能力不强，很容易导致模型偏向于样本中数量较多的一类。
2. 正则化参数的影响，在该模型中我们选择的正则化参数 $\lambda = [2 : 2 : 40]$ (数量为20个)，同时从训练程度较好的两个模型

Input layer : 1024 Hidden layer :30 Output layer :2 iters = 500 和

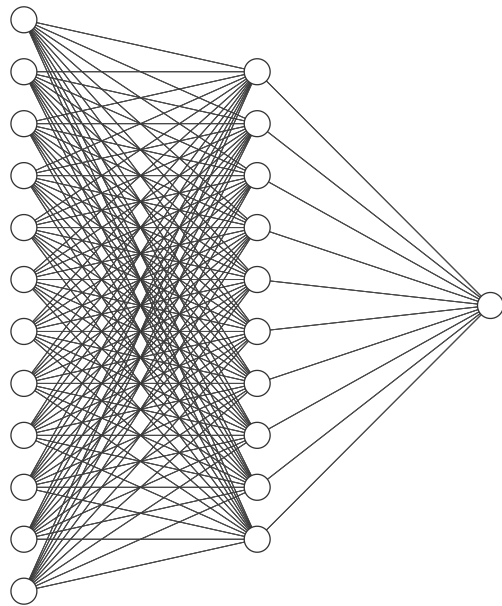
Input layer : 1024 Hidden layer :60 Output layer :2 iters = 2000 可以看到随着 λ 的增加模型泛化性能有所变化，这是我们的加入的正则项可以在模型发生高偏差和高方差发生时对模型进行调整。在没有正则项的加入，随着训练程度的变深，模型在训练集上的偏差持续变小，此时该模型在测试集上的表现往往不好。但是此时

如果我们加入正则项的话，可以增加模型的泛化性能，减少模型发生过拟合的现象。但是正则化参数不易太大，正则化参数太大会导致模型训练时欠拟合。

4 多分类问题：

4.1 建模：

Input layer : 1024 Hidden layer :60 Output layer :15 iters = 4000
 $\lambda = [2 : 2 : 40]$



4.2 数据预处理：

将集合中80%作为训练集，20%作为测试集。

```
X = fea;
y = gnd;
sample_num = 15;
%划分训练集和测试集
X_tra = zeros(sample_num*8,1024);
y_tra = zeros(sample_num*8,1);
X_val = zeros (sample_num*2,1024);
y_val = zeros(sample_num*2,1);
for i =1:sample_num
```

```

a = 1+(i-1)*8;
b = 8+(i-1)*8;
c = 1+(i-1)*10;
d = 8+(i-1)*10;
X_tra(a: b,:) = X(c:d,:);
y_tra(a:b) = y(c:d);
e = 1+(i-1)*2;
f = 2+(i-1)*2;
g = 9+(i-1)*10;
h = 10+(i-1)*10;
X_val(e:f,:) =X(g:h,:);
y_val(e:f) =y(g:h);
end

```

4.3 使用测试数据集对所建模型进行评估;

lambda	准确率
2	0.866666667
4	0.866666667
6	0.766666667
8	0.733333333
10	0.766666667
12	0.466666667
14	0.533333333
16	0.833333333
18	0.8
20	0.866666667
22	0.333333333
24	0.933333333
26	0.866666667
28	0.866666667
30	0.833333333
32	0.833333333
34	0.866666667
36	0.8
38	0.766666667
40	0.833333333

5 实验总结

1. 神经网络模型学习能力很强大，但随之而来的是模型结构变得的复杂，这就是得训练的时间变长，对计算性能的要求高，以及需要优化算法。
2. 本次实验由于自己电脑性能受限，时间不够，没办法深入地探索深度学习的其他部分，但是在接下来的学习和生活中我会继续研究学习机器学习。
3. 同时我也看到了神经网络的一些弊端，又或者说是传统机器学习算的优势，就是训练时间过长，如果那些需要快速响应的实际问题，我该如何去使用呢？