1.

（1）

class	-	-	-	-	+	+	-	-	+	+
	0.08	0.15	0.35	0.44	0.45	0.47	0.55	0.67	0.69	0.73
TP	2									
FP	2									
TN	4									
FN	2									
accuracy	0.6									
precision	0.5									
TPR(recall)	0.5									
FPR	1/3									
F-measure	0.5									

（2）

class	+	+	-	-	+	-	-	-	+	-
	0.01	0.03	0.04	0.05	0.09	0.31	0.38	0.45	0.61	0.68
TP	1									
FP	1									
TN	5									
FN	3									
accuracy	0.6									
precision	0.5									
TPR(recall)	0.25									
FPR	1/6									
F-measure	1/3									

TPR,M1>M2,分类模型 M1 在这个测试集上表现得更好。

（3）

class	-	-	-	-	+	+	-	-	+	+
	0.08	0.15	0.35	0.44	0.45	0.47	0.55	0.67	0.69	0.73
TP	4									
FP	4									
TN	2									
FN	0									
accuracy	0.6									
precision	0.5									
TPR(recall)	1									
FPR	2/3									
F-measure	2/3									

TPR=1,阈值为 0.2 时结果更好。

（4）

对于模型 M1，

class	-	-	-	-	+	+	-	-	+	+	
Threshold>=	0.08	0.15	0.35	0.44	0.45	0.47	0.55	0.67	0.69	0.73	1.0
TP	4	4	4	4	4	3	2	2	2	1	0
FP	6	5	4	3	2	2	2	1	0	0	0
TN	0	1	2	3	4	4	4	5	6	6	6
FN	0	0	0	0	0	1	2	2	2	3	4
accuracy	0.4	0.5	0.6	0.7	0.8	0.7	0.6	0.7	0.8	0.7	0.6
precision											
TPR	1	1	1	1	1	0.75	0.5	0.5	0.5	0.25	0
FPR	1	5/6	2/3	0.5	1/3	1/3	1/3	1/6	0	0	0
F-measure											

阈值取 0.45 时最优，此时 accuracy = 0.8, TPR=1， FPR=1/3.

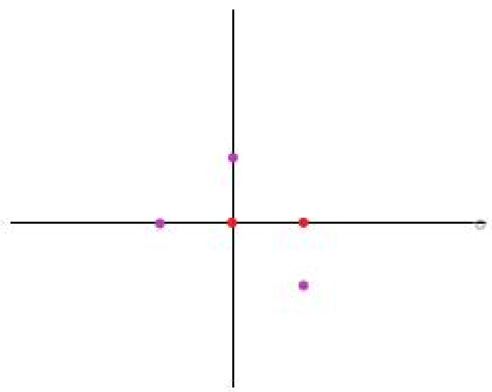
对于模型 M2，

class	+	+	-	-	+	-	-	-	+	-
Threshold>=	0.01	0.03	0.04	0.05	0.09	0.31	0.38	0.45	0.61	0.68
TP	4	3	2	2	2	1	1	1	1	0
FP	6	6	6	5	4	4	3	2	1	1
TN	0	0	0	1	2	2	3	4	5	5
FN	0	1	2	2	2	3	3	3	3	4
accuracy	0.4	0.3	0.2	0.3	0.4	0.3	0.4	0.5	0.6	0.5
precision										
TPR(recall)	1	0.75	0.5	0.5	0.5	0.25	0.25	0.25	0.25	0
FPR	1	1	1	5/6	2/3	2/3	0.5	1/3	1/6	1/6
F-measure										

阈值取 0.61 时最优，此时 accuracy = 0.6, TPR = 0.25, FPR = 1/6.

2.

（1）

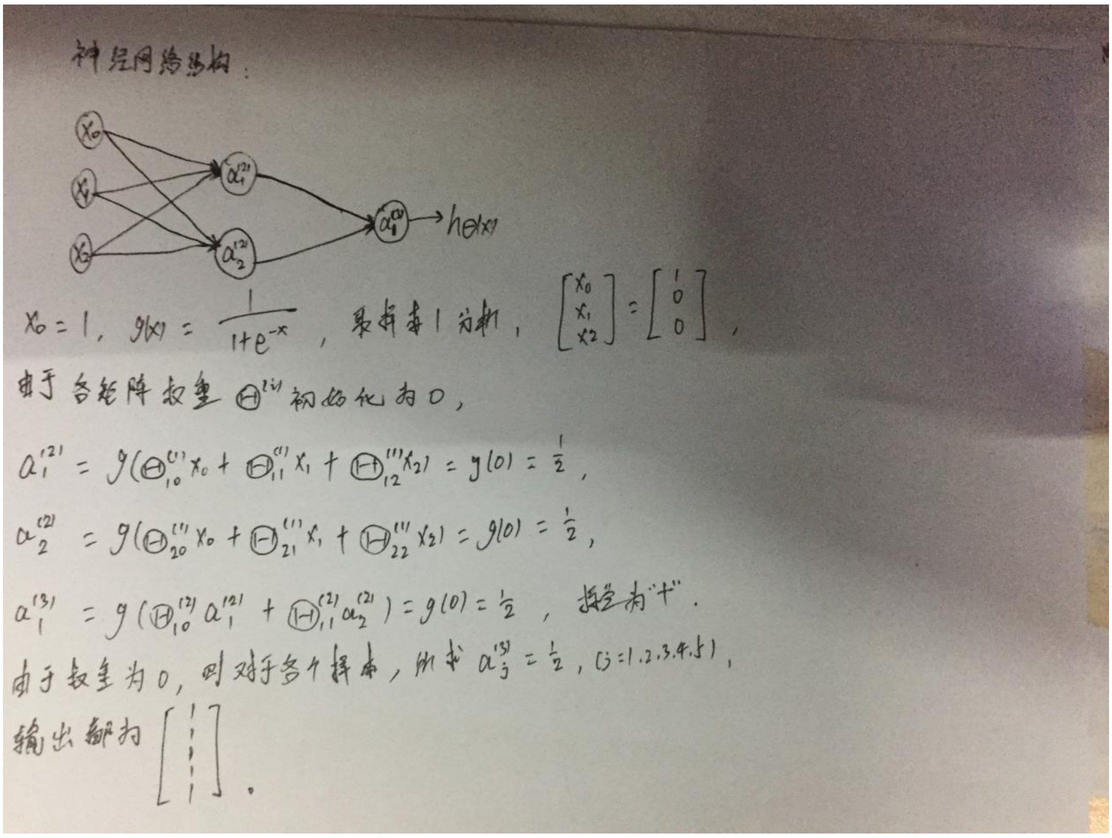


由图可以看出，不存在任意一条直线可以将样本正确分类，所以此训练样本线性不可分。

（2）

使用 Sigmoid 激活函数，给神经元引入了非线性因素，使得神经网络可以任意逼近任何非线性函数，这样神经网络就可以应用到众多的非线性模型中。而如果将 Sigmoid 函数换成线性函数，则隐藏层就失去了意义，每一层输出都是上层输入的线性函数，无论神经网络有多少层，输出都是输入的线性组合。

（3）



神经网络结构：

$x_0=1$, $g(x)=\frac{1}{1+e^{-x}}$，取样本1分析，$\begin{bmatrix}x_0\\x_1\\x_2\end{bmatrix}=\begin{bmatrix}1\\0\\0\end{bmatrix}$，

由于各矩阵权重 $\Theta^{(1)}$ 初始化为0，

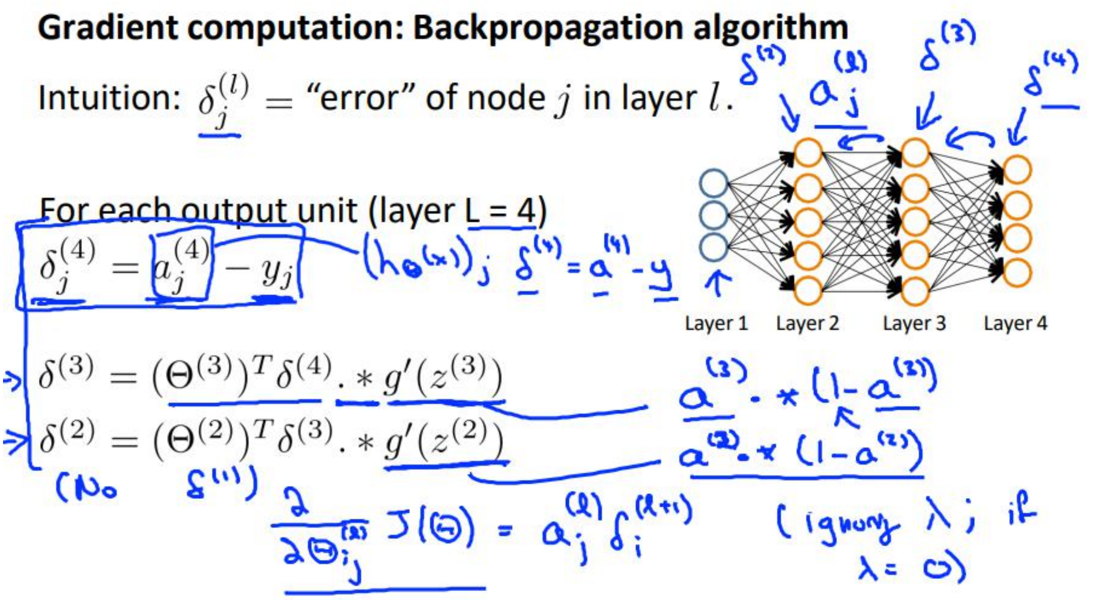
$a_1^{(2)}=g(\Theta_{10}^{(1)}x_0+\Theta_{11}^{(1)}x_1+\Theta_{12}^{(1)}x_2)=g(0)=\frac{1}{2}$，

$a_2^{(2)}=g(\Theta_{20}^{(1)}x_0+\Theta_{21}^{(1)}x_1+\Theta_{22}^{(1)}x_2)=g(0)=\frac{1}{2}$，

$a_1^{(3)}=g(\Theta_{10}^{(2)}a_1^{(2)}+\Theta_{11}^{(2)}a_2^{(2)})=g(0)=\frac{1}{2}$，预测为“1”.

由于权重为0，则对于各个样本，所求 $a_j^{(3)}=\frac{1}{2}$，$(j=1,2,3,4,5)$，

输出都为 $\begin{bmatrix}1\\1\\1\\1\\1\end{bmatrix}$.



Backpropagation algorithm

Training set $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$

Set $\triangle_{ij}^{(l)} = 0$ (for all l, i, j). (use to compute $\frac{\partial}{\partial \Theta_{ij}^{(l)}} J(\Theta)$)

For $i = 1$ to m ← $(x^{(i)}, y^{(i)})$

  Set $a^{(1)} = x^{(i)}$

  Perform forward propagation to compute $a^{(l)}$ for $l = 2, 3, \ldots, L$

  Using $y^{(i)}$, compute $\delta^{(L)} = a^{(L)} - y^{(i)}$

  Compute $\delta^{(L-1)}, \delta^{(L-2)}, \ldots, \delta^{(2)}$

  $\triangle_{ij}^{(l)} := \triangle_{ij}^{(l)} + a_j^{(l)} \delta_i^{(l+1)}$ $:= \Delta^{(l)} + \delta^{(l+1)} (a^{(l)})^T$

$D_{ij}^{(l)} := \frac{1}{m} \triangle_{ij}^{(l)} + \lambda \Theta_{ij}^{(l)}$ if $j \neq 0$

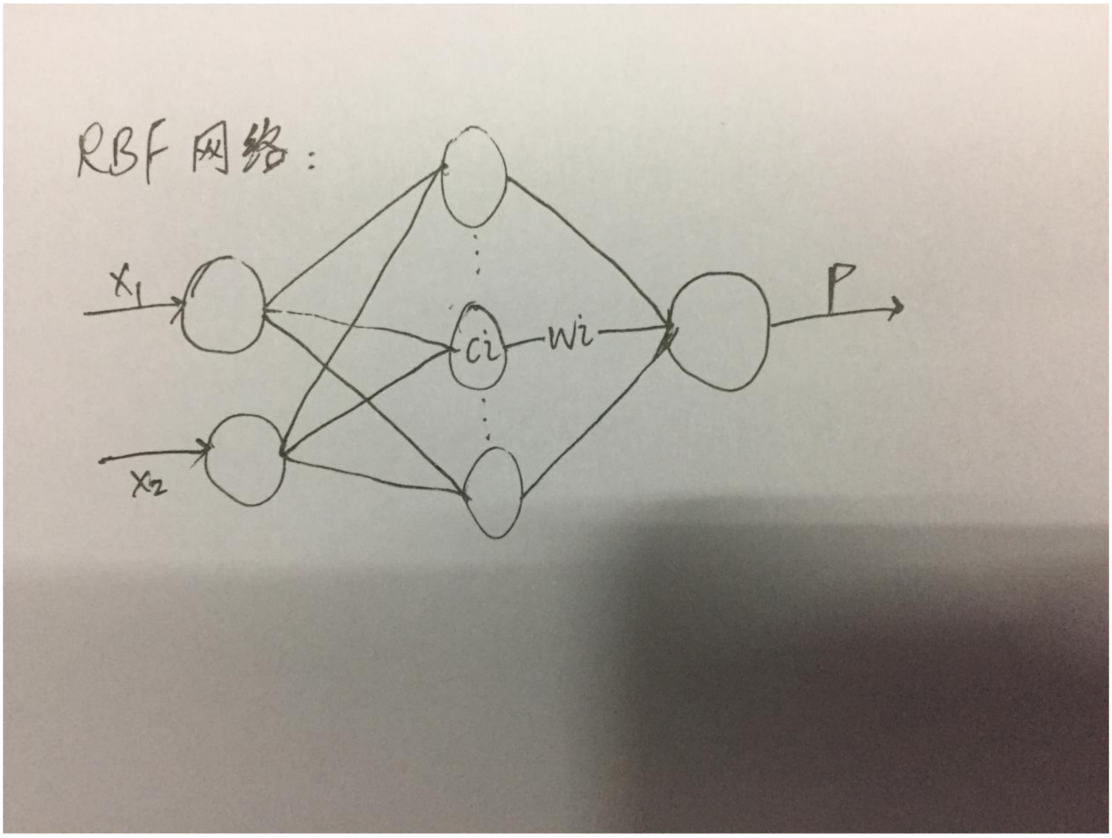
$D_{ij}^{(l)} := \frac{1}{m} \triangle_{ij}^{(l)}$ if $j = 0$

$$\frac{\partial}{\partial \Theta_{ij}^{(l)}} J(\Theta) = D_{ij}^{(l)}$$

在前馈算法中，矩阵权重初始化为 0 导致计算的结果都为‘+’，分类不正确，神经网络没起到任何作用；在 BP 算法中，权重初始为 0，将导致最终结果都为 0，反向传播也失去了意义。

（4）

设计 RBF 神经网络如图，



激活函数：

$$f(x) = \sum_{i=1}^{q} w_i \rho(x, c_i),$$

高斯径向基函数：

$$\rho(x, c_i) = e^{-\beta_i ||x - c_i||^2}$$

通常采用两步过程来训练 RBF 网络，第一步：确定神经元中心 Ci，常用的方式包括随机取样，聚类等；第二步，利用 BP 算法等来确定参数 wi 和βi。

经过代码测试，取隐藏层数为 10，学习率为 0.5，结果如图：

命令行窗口

```
输出结果
    1.0001
    0.9949
    0.0040
   -0.0460
    0.0549

参数wi
    0.4528    0.2573   -0.2403    0.6879   -0.1175   -0.2976   -0.3193    0.2277    0.6545    0.1640

参数betai
    1.7908    1.3807    0.5279    1.5383    1.0207    1.0877    0.9553    1.4465    1.5477    1.1109

参数ci
    0.5975    0.1249
    0.3353    0.0244
    0.2992    0.2902
    0.4526    0.3175
    0.4226    0.6537
    0.3596    0.9569
    0.5583    0.9357
    0.7425    0.4579
    0.4243    0.2405
fx  0.4294    0.7639
```


3.

（1）

根节点信息熵 Ent(root) = - $\sum_{k=1}^{2} p_k \log_2 p_k$ = -(4/10) $\log_2(4/10)$ -(6/10) $\log_2(6/10)$=0.97

以属性 A 划分的信息熵 Ent(D^1) = 0.985, Ent(D^2) = 0;

信息熵增益 GAIN = Ent(D)- $\sum_{k=1}^{2} \frac{|D^k|}{|D|}$ Ent(D^k) = 0.28;

以属性 B 划分的信息熵 Ent(D^1) = 0.81, Ent(D^2) = 0.65;

信息熵增益 GAIN = Ent(D)- $\sum_{k=1}^{2} \frac{|D^k|}{|D|}$ Ent(D^k) = 0.26;

以 A 划分的信息熵增益更大，所以选择 A 划分。

（2）

根节点 Gini 系数 Gini(root) = 0.48;

以属性 A 划分 Gini(T) = 0.49, Gini(F) = 0;Gini 增益为 0.48-0.34 = 0.12;

以属性 B 划分 Gini(T) = 0.38,Gini(F) = 0.28,Gini 增益为 0.48-0.32 = 0.16;

以属性 B 划分的 Gini 增益更大，所以选择 B 划分。

（3）

根节点的分类误差 Error(root) = 0.4;

以属性 A 划分 Error(T) = 3/7,Error(F) = 0,Error 增益为 0.1;

以属性 B 划分 Error(T) = 0.25,Error(F) = 1/6,Error 增益为 0.2;

以属性 B 划分的分类误差增益更大，所以选择 B 划分。
（4）
信息熵增益偏好于属性 A，Gini 增益，分类误差增益更偏好于属性 B。