1.

已知训练集$D=\{((0,0)^T,0),((0,1)^T,1),((1,0)^T,1),((1,1)^T,0)\}$

设隐层两个突触的权值、偏置值、阈值分别为(v_{11}, v_{21}, k_1, γ_1)、(v_{12}, v_{22}, k_1, γ_2)

设输出层突触的权值、偏置值、阈值为(ω_{11}, ω_{21}, k_2, θ_1)

由于需要得到对于四个输入的无差错感知机参数，样本数少且重要，因而不妨采用较大的学习率，故设学习率$\eta=1$

由于反向传播算法对于任意参数v的更新估计式为

$$v \leftarrow v+\Delta v$$

因而对于偏置值，有

$$\Delta k_1=-\eta\frac{\partial E_k}{\partial k_1}=-\eta\frac{\partial E_k}{\partial \hat{y}_j^k}\cdot\frac{\partial \hat{y}_j^k}{\partial \alpha_j}\cdot\frac{\partial \alpha_j}{\partial b_h}\cdot\frac{\partial b_h}{\partial \beta_h}\cdot\frac{\partial \beta_h}{\partial k_1}$$

由于

$$\begin{aligned} f'(x)&=\left(\frac{1}{1+e^{-x}}\right)' \\ &=\frac{e^{-x}}{(1+e^{-x})^2} \\ &=\frac{1}{1+e^{-x}}\cdot\frac{1+e^{-x}-1}{1+e^{-x}} \\ &=\frac{1}{1+e^{-x}}\cdot(1-\frac{1}{1+e^{-x}}) \\ &=f(x)(1-f(x)) \end{aligned}$$

记$g_j=\hat{y}_j^k(1-\hat{y}_j^k)(y_j^k-\hat{y}_j^k)$，故有

$$\Delta k_1=\eta g_j\omega_{hj}b_h(1-b_h)$$

同理可得

$$\Delta k_2=-\eta\frac{\partial E_k}{\partial k_2}=-\eta\frac{\partial E_k}{\partial \hat{y}_j^k}\cdot\frac{\partial \hat{y}_j^k}{\partial \alpha_j}\cdot\frac{\partial \alpha_j}{\partial k_2}=\eta g_j$$

对其余参数，同理可得

$$\begin{gathered} \Delta\omega_{hj}=\eta b_h g_j \\ \Delta\theta_j=-\eta g_j \\ \Delta v_{ih}=\eta e_h x_i \\ \Delta\gamma_h=-\eta e_h \end{gathered}$$

由于

$$A\oplus B=\overline{A}B+A\overline{B}$$

故不妨设初始值

$$\begin{gathered} v_{11}=1,v_{12}=-1,v_{21}=-1,v_{22}=1,\omega_{11}=1,\omega_{21}=1 \\ k_1=0,k_2=0 \\ \theta_1=0.5,\gamma_1=0.5,\gamma_2=0.5 \end{gathered}$$

以下演示一组计算过程，对于该感知机，有

$$d=2,q=2,l=1$$

代入第一组训练样本$((x_1,x_2)^T,y_1)=((0,0)^T,0)$

由

$$b_h=f(\beta_h-\gamma_h)$$
$$\beta_h=\sum_{i=1}^{d}v_{ih}x_i+k_1$$

得

$$\beta_1=v_{11}x_1+v_{21}x_2+k_1=0$$
$$\beta_2=v_{12}x_1+v_{22}x_2+k2=0$$
$$b_1=f(\beta_1-\gamma_1)\approx 0.3775406687981454$$
$$b_2=f(\beta_2-\gamma_2)\approx 0.3775406687981454$$

由

$$\hat{y}_j^k=f(\alpha_j-\theta_j)$$

$$\alpha_h=\sum_{h=1}^{q}\omega_{hj}b_h+k_2$$

得

$$\alpha_1=\omega_{11}b_1+\omega_{21}b_2+k_2\approx 0.7550813375962908$$
$$\hat{y}_1^1=f(\alpha_1-\theta_1)\approx 0.5634267935467275$$

由

$$g_j=\hat{y}_j^k(1-\hat{y}_j^k)(y_j^k-\hat{y}_j^k)$$

得

$$g_1=\hat{y}_1^1(1-\hat{y}_1^1)(y_1^1-\hat{y}_1^1)\approx -0.13859005598150353$$

由

$$e_h=b_h(1-b_h)\sum_{j=1}^{h}\omega_{hj}g_j$$

得

$$e_1=b_1(1-b_1)\omega_{11}g_1\approx -0.032569177629880125$$
$$e_2=b_2(1-b_2)\omega_{21}g_1\approx -0.032569177629880125$$

综上，又由

$$\Delta\omega_{hj}=\eta b_h g_j$$
$$\Delta\theta_j=-\eta g_j$$
$$\Delta v_{ih}=\eta e_h x_i$$
$$\Delta\gamma_h=-\eta e_h$$

得

$$\Delta k_1=-0.06513835525976025,\Delta k_2=-0.13859005598150353$$
$$\Delta\omega_{11}=-0.05232338242402926,\Delta\omega_{21}=-0.05232338242402926$$
$$\Delta\theta_1=0.13859005598150353$$
$$\Delta v_{11}=0,\Delta v_{12}=0,\Delta v_{21}=0,\Delta v_{22}=0,$$
$$\Delta\gamma_1=0.032569177629880125,\Delta\gamma_2=0.032569177629880125$$

故更新后的参数为

$$k_1\leftarrow k_1+\Delta k_1=-0.06513835525976025$$
$$k_2\leftarrow k_2+\Delta k_2=-0.13859005598150353$$
$$\omega_{11}\leftarrow\omega_{11}+\Delta\omega_{11}=0.9476766175759708$$
$$\omega_{21}\leftarrow\omega_{21}+\Delta\omega_{21}=0.9476766175759708$$

$$\gamma_1 \leftarrow \gamma_1 + \Delta\gamma_1 = 0.5325691776298801$$
$$\gamma_2 \leftarrow \gamma_2 + \Delta\gamma_2 = 0.5325691776298801$$
如此循环代入四个样本，具体程序代码见 1.py，最终得到的参数为
$$v_{11} = 0.9757206400721918, v_{12} = -1.056984190439433$$
$$v_{21} = -1.053977868161747, v_{22} = 0.9863261213588721$$
$$\omega_{11} = 0.9616023439200964, \omega_{21} = 0.9836969702898878$$
$$k_1 = -0.12568051150158813, k_2 = -0.08490324360998451$$
$$\theta_1 = 0.5849032436099846$$
$$\gamma_1 = 0.5646080033260386, \gamma_2 = 0.5558520057149792$$
此时若代入四组数据，得到的神经网络输出值为

$$(x_1, x_2)^T = (0,0)^T\text{时，}\ \hat{y}_1^1 = 0.4954455330215097$$

$$(x_1, x_2)^T = (0,1)^T\text{时，}\ \hat{y}_1^1 = 0.509869352923375$$

$$(x_1, x_2)^T = (1,0)^T\text{时，}\ \hat{y}_1^1 = 0.5065533105295847$$

$$(x_1, x_2)^T = (1,1)^T\text{时，}\ \hat{y}_1^1 = 0.48748879703791137$$

使用阈值函数

$$\sigma(t) = \begin{cases} 0, & t \le 0.5 \\ 1, & t > 0.5 \end{cases}$$

即可得到符合预期的输出。

2.

(1).

指出 Fisher 线性判别中，w 的比例因子对 Fisher 判别结果无影响的原因：

$Fisher$线性判别的思想在于，让同类样例投影点的协方差尽可能小，即令类内散度矩阵$S_\omega = \sum_0 + \sum_1$尽可能小。同时，让不同类中心之间的距离尽可能大，即类间散度矩阵$S_b = (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T$尽可能大。

但该问题难以直接求解，因而引入了正交投影矩阵ω，将该问题降维，借助广义瑞利商的概念，将问题转化为最大化$J(\omega) = \dfrac{\omega^T S_b \omega}{\omega^T S_\omega \omega}$

因而$Fisher$线性判别的结果，也即对于最大化$J(\omega)$的问题而言，ω作为一个正交投影矩阵，结果只与正交投影矩阵的投影方向的因子有关，而与正交投影矩阵的比例因子无关。

分析 J(w)可用 Lagrange 乘子法求解的条件：

将等式约束$\omega^T S_\omega \omega = 1$代入，得
$$\mathcal{L}(\omega, \lambda) = C(\omega) = -S_b\omega + \lambda(\omega^T S_\omega \omega - 1)$$
假设使$J(\omega)$取得极值的解为(ω^*, λ^*)，则由 KKT 条件可知
$$\frac{\partial}{\partial \omega}\mathcal{L}(\omega^*, \lambda^*) = 0$$
$$\frac{\partial}{\partial \lambda}\mathcal{L}(\omega^*, \lambda^*) = 0$$

由于$\lambda \in \mathcal{R}$可导，因而只需确定$\mathcal{L}(\omega, \lambda)$对$\omega$可导。
若样本的维数为n，则$\mathcal{L}(\omega, \lambda)$是$n \times n$的矩阵，矩阵对矩阵求导后有$|S_b| = n$、$|S_b \omega| = n-1$。从几何意义上而言，也即对原有的样本进行一次降维的投影变换。
故求解条件为——ω是一个投影矩阵。

(2).
随机变量x的分布律为

$$P(x = x_i) = p(x|\theta) = \frac{1}{\theta}$$

故似然函数为

$$L(\theta) = \prod_{i=1}^{N} P(x = x_i) = \prod_{i=1}^{N} p(x|\theta) = \frac{1}{\theta^N}$$

而

$$lnL(\theta) = ln\frac{1}{\theta^N} = -Nln\theta$$

$$\frac{d}{d\theta}lnL(\theta) = -\frac{N}{\theta} < 0$$

故

$$\hat{\theta} = \max_k x_k$$

3.
鸢尾花数据集D中$data$域包含四个属性（不妨命名为$data1, data2, data3, data4$），在$target$域中将其划分为了三种花，按照信息熵的定义，有

$$Ent(target) = -\sum_{k=1}^{3} p_k log_2 p_k$$

由于数据集为连续值，为了应用最大信息增益算法，假定每个属性值只产生 2 个分支结点，即$v = 2$，分别代表大于、小于划分点值的两种情况

$$Gain(D, a) = Ent(D) - \sum_{v=1}^{2} \frac{|D^v|}{|D|} Ent(D^v)$$

因此，首先对四个属性值分别进行二分离散化，方法为：
对于连续属性a在D中的n个不同取值，先将这些值从小到大排序，记为$\{a^1, a^2, ..., a^n\}$，基于划分点t可将D分为子集D_t^-和D_t^+，其中D_t^-表示在属性a上不大于t的样本，D_t^+反之。由此，我们可考察包含$n-1$个元素的候选划分点集合

$$T_a = \{\frac{a^i + a^{i+1}}{2} | 1 \le i \le n-1\}$$

由此，对于属性a的信息增益，可取为

$$Gain(D, a) = \max_{t \in T_a} Gain(D, a, t) = \max_{t \in T_a} \left(Ent(D) - \sum_{\lambda \in \{-,+\}} \frac{|D_t^\lambda|}{|D|} Ent(D_t^\lambda) \right)$$

以下依次为程序运行后，基于训练集的决策树、经过测试集预剪枝后的决策树：