**Problem 11:** Various solutions

**Problem 12:** Details for the first 3 iterations are given in Table 1. After 3 iterations all points are correctly classified and thus the misclassification error is 0. If one were to continue the AdaBoost algorithm, the split variables and cutoffs values repeat (i.e. for iteration 4 we split variable 1 at 2, for iteration 5 we split variable 1 at 7, etc.). The final classifier is displayed in Figure 1.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>$w_{1m}$</th>
<th>$w_{2m}$</th>
<th>$w_{3m}$</th>
<th>$w_{4m}$</th>
<th>$w_{5m}$</th>
<th>$w_{6m}$</th>
<th>$w_{7m}$</th>
<th>$w_{8m}$</th>
<th>$w_{9m}$</th>
<th>$w_{10m}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m = 1$</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
</tr>
<tr>
<td>$m = 2$</td>
<td>0.1000</td>
<td>0.2333</td>
<td>0.2333</td>
<td>0.2333</td>
<td>0.2333</td>
<td>0.2333</td>
<td>0.2333</td>
<td>0.2333</td>
<td>0.2333</td>
<td>0.2333</td>
</tr>
<tr>
<td>$m = 3$</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
</tr>
</tbody>
</table>

| $\alpha_m$ | 0.8472979 | 1.299230 | 1.8458267 |
| $\overline{err}_m$ | 0.3000000 | 0.2142857 | 0.1363636 |
| Split variable | 1 | 1 | 2 |
| Cutoff value | 2 | 7 | 5.25 |

Table 1: Values from the first 3 iterations of the AdaBoost algorithm

![Problem 12, final classifier](image)

Figure 1: Final classifier using AdaBoost
**Problem 13:** Let $X$ represent the input variable, $Z$ the first hidden layer variable, $W$ the second hidden layer variable, and $Y$ the output.

For hidden layer 1, $z_{0i} = 1$ and $z_{mi} = \sigma(\alpha_{0m} + \alpha'_m x_i)$, where $\alpha_m = (\alpha_{1m}, \alpha_{2m}, \alpha_{3m})'$ and $x_i = (x_{i1}, x_{i2}, x_{i3})'$, $i = 1, \ldots, N$, $m = 1, 2$. Let $z_i = (z_{1i}, z_{2i})'$ and note that $x_{i0} = 1$.

For hidden layer 2, $w_{0i} = 1$ and $w_{ki} = \sigma(\beta_{0k} + \beta'_k z_i)$, where $\beta_k = (\beta_{1k}, \beta_{2k})'$, $i = 1, \ldots, N$, $k = 1, 2$. Let $w_i = (w_{1i}, w_{2i})'$.

Finally, $f(x_i) = g(\gamma_0 + \gamma' w_i)$, where $\gamma = (\gamma_1, \gamma_2)'$.

Let $\theta$ stand for the whole set of parameters and consider the SEL fitting criterion

$$R(\theta) = \frac{1}{2} \sum_{i=1}^{N} (y_i - f(x_i))^2.$$ 

Denote $R_i = \frac{1}{2}(y_i - f(x_i))^2$. Partial derivatives with respect to the parameters are (by the chain rule)

$$\frac{\partial R_i}{\partial \gamma_k} = -(y_i - f(x_i))g'(\gamma_0 + \gamma' w_i)w_{ki}, \quad k = 0, 1, 2$$

$$= \delta_i w_{ki} \text{ for } \delta_i = -(y_i - f(x_i))g'(\gamma_0 + \gamma' w_i)$$  \hspace{1cm} (1)

$$\frac{\partial R_i}{\partial \beta_{mk}} = -(y_i - f(x_i))g'(\gamma_0 + \gamma' w_i)\gamma_k \sigma'(\beta_{0k} + \beta'_k z_i)z_{mi}, \quad m = 0, 1, 2, \quad k = 1, 2$$

$$= \delta_i \gamma_k \sigma'(\beta_{0k} + \beta'_k z_i)z_{mi}$$

$$= \eta_{ki} z_{mi} \text{ for } \eta_{ki} = \delta_i \gamma_k \sigma'(\beta_{0k} + \beta'_k z_i)$$  \hspace{1cm} (2)

$$\frac{\partial R_i}{\partial \alpha_{lm}} = -(y_i - f(x_i))g'(\gamma_0 + \gamma' w_i)\frac{\partial \sum_{k=1}^{2} \gamma_k w_{ki}}{\partial \alpha_{lm}}$$

$$= -(y_i - f(x_i))g'(\gamma_0 + \gamma' w_i)\sum_{k=1}^{2} \gamma_k \sigma'(\beta_{0k} + \beta'_k z_i)\beta_{mk}(\sigma'(\alpha_{0m} + \alpha'_m x_i)) x_{il}$$

$$= \left( \sum_{k=1}^{2} \eta_{ki} \beta_{mk} \sigma'(\alpha_{0m} + \alpha'_m x_i) \right) x_{il}$$

$$= \lambda_{mi} x_{il} \text{ for } \lambda_{mi} = \sum_{k=1}^{2} \eta_{ki} \beta_{mk} \sigma'(\alpha_{0m} + \alpha'_m x_i)$$  \hspace{1cm} (3)

An iterative search to make $R(\theta)$ small, with learning rate $\varphi_r$, proceeds by setting

$$\gamma_k^{(r+1)} = \gamma_k^{(r)} - \varphi_r \sum_{i=1}^{N} \frac{\partial R_i}{\partial \gamma_k} |_{\gamma^{(r)}} \eta_{ki}^{(r)}$$

$$= \gamma_k^{(r)} - \varphi_r \sum_{i=1}^{N} \delta_i^{(r)} w_{ki}^{(r)}$$ \hspace{1cm} (4)

$$\beta_{mk}^{(r+1)} = \beta_{mk}^{(r)} - \varphi_r \sum_{i=1}^{N} \eta_{ki}^{(r)} z_{mi}^{(r)}$$ \hspace{1cm} (5)
\[
\alpha_{lm}^{(r+1)} = \alpha_{lm}^{(r)} - \varphi_r \sum_{i=1}^{N} \lambda_{mi}^{(r)} x_{il}^{(r)}
\]  

(6)

So operationally

1. Forward pass: use the \( r \)th iterates of \( \alpha \)'s, \( \beta \)'s, and \( \gamma \)'s to compute \( f^{(r)}(x_i) \)

2. Backward pass:
   - use the \( r \)th iterates of \( f(x_i) \), \( \gamma \)'s and \( w \)'s to compute \( \delta_i^{(r)} \)
   - use the \( r \)th iterates of \( \delta \)'s, \( \gamma \)'s, \( \beta \)'s and \( z \)'s to compute \( \eta_{ki}^{(r)} \)
   - use the \( r \)th iterates of \( \eta \)'s, \( \beta \)'s, and \( \alpha \)'s to compute \( \lambda_{mi}^{(r)} \)

3. The \( \delta_i^{(r)} \), \( \eta_{ki}^{(r)} \), and \( \lambda_{mi}^{(r)} \) then provide partials using (1), (2), and (3)

4. Updates for the parameters \( \alpha \), \( \beta \), and \( \gamma \) come from (4), (5), and (6)