CS 491 Midterm 2 Name Solution

Short Problems (20 points)

1. In the following statement, underline and indicate which part of this refers to recall, and which part refers to to precision. (4 points)
"I Swear To Tell The Truth, The Whole Truth and Nothing But The Truth So Help Me God"

① Recall

② Precision

2. Let's say we have a single layer neural network. Our data has N samples and D features. We have computed the gradient of our Loss function with respect to our weights. In Big-O notation, what is the cost of one gradient descent update given the gradient? (4 points)

$O(D)$

3. Suppose we take all the weights and biases in a network of perceptrons, and multiply them by a positive constant, $c > 0$. Does the behavior of the network change? Yes or no. Explain. (4 points)

Multiplying weights by positive

4. AdaBoost will eventually reach zero training error, regardless of the type of weak classifier it uses, provided enough weak classifiers have been combined. True or False. Explain. (4 points)

Adaboost is a linear combination of classifiers. If the data is not linearly separable by a combination of these classifiers then no.

5. Show for a matrix A and and eigenvector v that if $Av = \lambda v$ then $A^k v = \lambda^k v$. (4 points)

$Av = \lambda v$

$AAv = \lambda \lambda v$?

$A(\underbrace{Av}_{\lambda v}) = \lambda(\lambda v)$

$\lambda A v = \lambda \lambda v.$

true for $k = 2$

$A \underbrace{A^{k-1} v}_{\lambda^{k-1} v} = \lambda \lambda^{k-1} v$

$Av = \lambda v$

$\lambda^k v$ ✓

491

10 points

6. Consider the binary threshold neuron, $h = sign(w^T x)$ defined such that $h \in \{0, 1\}$, with no bias b or w_0. Consider the following set of four input features, x:

$$(1,0,0)^T,\ (0,1,0)^T,\ (0,0,1)^T,\ (1,1,1)^T$$

(a) Find a three-dimensional parameter vector w such that the neuron will have the output pattern $\{h\} = \{1, 1, 1, 1\}$ for the given four input features.

$w = (1, 1, 1)$

3 pts.

(b) Find a three-dimensional parameter vector w such that the neuron will have the output pattern $\{h\} = \{1, 1, 0, 0\}$ for the given four input features.

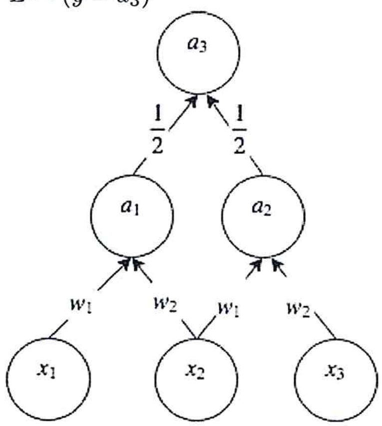
3 pts. $w = (1, 1, -3)$

(c) Find an unrealizable output pattern $\{h\}$.

4 pts. $h = \{1, 1, 1, 0\}$

20 points

7. We depict a neural network below. There is weight sharing in the first level. This means that the two edges that are labeled w_1 are required to have the same value. If we change one, we change the other by the same amount. Similarly for w_2. There are no bias terms. Between the hidden layer and the output layer there are two edges labeled $\frac{1}{2}$. This is average pooling. These weights are fixed and never change. Backprop does not affect them. We also use a regression loss. So the network can be described with the equations:
$a_1 = w_1x_1 + w_2x_2$
$a_2 = w_1x_2 + w_2x_3$
$a_3 = \frac{1}{2}a_1 + \frac{1}{2}a_2$
$L = (y - a_3)^2$



We have one training example with $x_1 = 0$, $x_2 = 2$, $x_3 = 2$, $y = 2$. We initialize the network with $w_1 = -1$, $w_2 = 1$. For this training example:

(a) What are the values of a_1, a_2, a_3, and L?

$a_1 = 2$ $\quad$ $a_3 = 1$

4pts. each

$a_2 = 0$ $\quad$ $L = 1$

(b) What is $\frac{\partial L}{\partial a_3}$?

$-2 = -2(y - a_3)$

(c) What is $\frac{\partial a_3}{\partial a_1}$?

$\frac{1}{2}$

(d) What is $\frac{\partial a_1}{\partial w_1}$?

$x_1 = 0$

(e) What is $\frac{\partial L}{\partial w_1}$?

-2

10 points

8. Suppose we perform PCA on the four points with coordinates (x,y), (x,-y), (-x,y), (-x,-y).

   (a) What is the covariance matrix for this data? (No variance scaling).

centering gives same data back.

$$Z^T = \begin{bmatrix} x & x & -x & -x \\ y & -y & y & -y \end{bmatrix} \quad Z = \begin{bmatrix} x & y \\ x & -y \\ -x & y \\ -x & -y \end{bmatrix}$$

$$Z^T Z = \begin{bmatrix} 4x^2 & 0 \\ 0 & 4y^2 \end{bmatrix}$$

   (b) For what values of x and y will the principal component be $(1, 0)$?

$$\begin{bmatrix} 4x^2 & 0 \\ 0 & 4y^2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 4x^2 \\ 0 \end{bmatrix} = 4x^2 \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

eigenvalue

other p.c.

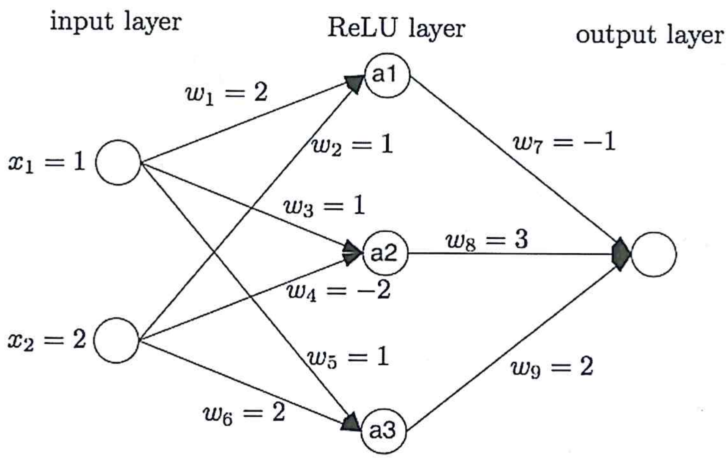
$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} = 4y^2 \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$4x^2 > 4y^2$

$|x| > |y|$ ✓

20 points

9. Consider the neural network below. $L = (y - \hat{y})^2$ Assume the label for the input sample is $y = 3$ Compute the values of all weights w_i after performing a gradient descent update with learning rate 0.1. ReLU layer indicates that every neuron in that layer applies a ReLU activation.



$$\frac{\partial L}{\partial w_7} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w_7}$$

$a_1 = 2\cdot 1 + 1\cdot 2 = 4$

$a_2 = 1\cdot 1 + -2\cdot 2 = 0$ (relu)

$a_3 = 1\cdot 1 + 2\cdot 2 = 5$

$\hat{y} = 4\cdot -1 + 3\cdot 0 + 5\cdot 2 = 6$

$$\frac{\partial L}{\partial \hat{y}} = -2(y-\hat{y}) = -2(3-6) = 6$$

$$\frac{\partial \hat{y}}{\partial w_7} = a_1 = 4$$

$$\frac{\partial \hat{y}}{\partial w_8} = a_2 = 0$$

$$\frac{\partial \hat{y}}{\partial w_9} = a_3 = 5$$

$$\frac{\partial \hat{y}}{\partial a_1} = w_7 = -1 \qquad \frac{\partial \hat{y}}{\partial a_2} = w_8 = 3 \qquad \frac{\partial \hat{y}}{\partial a_3} = w_9 = 2$$

$$\frac{\partial a_1}{\partial w_1} = x_1 = 1 \qquad \frac{\partial a_1}{\partial w_2} = x_2 = 2$$

because of relu these are both $\emptyset$

$$\frac{\partial a_2}{\partial w_3} = x_1 = 1 \qquad \frac{\partial a_2}{\partial w_4} = x_2 = 2$$

$$\frac{\partial a_3}{\partial w_5} = x_1 = 1 \qquad \frac{\partial a_3}{\partial w_6} = x_2 = 2$$

$$w_1 = w_1 - \eta \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_1} \frac{\partial a_1}{\partial w_1} = 2 - 0.1\left[6 \cdot -1 \cdot 1\right]$$

$$= 2.6$$

$$w_2 = w_2 - \eta \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_1} \frac{\partial a_1}{\partial w_2} = 1 - 0.1\left[6 \cdot -1 \cdot 2\right] = 2.2$$

$$w_3 = w_3 - \eta \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_2} \frac{\partial a_2}{\partial w_3} \qquad w_3 = 1$$

zero

$$w_4 = w_4 - \eta \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_2} \frac{\partial a_2}{\partial w_4} \qquad w_4 = -2$$

$$w_5 = w_5 - \eta \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_3} \frac{\partial a_3}{\partial w_5} \qquad w_5 = 1 - 0.1(6 \cdot 2 \cdot 1) = -0.2$$

$$w_6 = w_6 - \eta \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_3} \frac{\partial a_3}{\partial w_6} \quad w_6 = 2 - 0.1(6 \cdot 2 \cdot 2) = -0.4$$

$$w_7 = w_7 - \eta \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w_7} = -1 - .1(6 \cdot 4) = -2$$

$$w_8 = w_8 - \eta \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w_8} = 3 - 0.1(6 \cdot 0) = 3$$

$$w_9 = w_9 - \eta \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w_9} = 2 - 0.1(6 \cdot 5) = -1$$

20 points

10. In this problem, you will use Adaboost to learn a hidden function from this set of training examples. We will use two rounds of AdaBoost to learn a hypothesis for this data set. The Adaboost algorithm is provided on the back of this page for reference. Recall that in round number 1, AdaBoost chooses a weak learner that minimizes the weighted error ϵ. As weak learners, you will use axis parallel lines of the form

- if $x_1 > a$, then +1 else −1 or
- if $x_2 > b$, then +1 else −1, for some integers a, b.

(either one of these two forms, not a combination of the two).

Consider the following labeled data (x_1, x_2, y) where x_1 and x_2 are the attributes and y is the class variable:

sample	x_1	x_2	y
s_1	11	3	-1
s_2	10	1	-1
s_3	4	4	-1
s_4	12	10	+1
s_5	2	4	-1
s_6	10	5	+1
s_7	8	8	-1
s_8	6	5	+1
s_9	7	7	+1
s_{10}	7	8	+1

(a) The first step of AdaBoost is to create an initial data weight distribution D_1 (also called calculating the data weighting co-efficients). What are the initial weights given to data points s_4 and s_7 by the AdaBoost algorithm, respectively?

$\frac{1}{10}$ each.

(b) Which of the following three hypotheses minimizes the weighted error in the first round of AdaBoost, using the distribution D_1 computed in the above question? Circle one. Justify your answer.

$x_2 > 9$ $\quad$ $x_2 > 4$ $\quad$ $x_2 > 7$

Lowest error only get S_7 wrong

(c) What is the weighted error ϵ of the best classifier computed above in part (b)?

$\frac{1}{10}$

(d) Which of the following three hypotheses minimizes the weighted error in the second round of AdaBoost. Circle one. Justify your answer.

$x_2 > 9$ $\quad$ $x_1 > 5$ $\quad$ $x_2 > 7$

for details see 691 soln.

Algorithm 32 $\textsc{AdaBoost}(\mathcal{W}, \mathcal{D}, K)$

$d^{(0)} \leftarrow \langle \frac{1}{N}, \frac{1}{N}, \dots, \frac{1}{N} \rangle$ // Initialize

for $k = 1 \dots K$ **do**

  $f^{(k)} \leftarrow \mathcal{W}(\mathcal{D}, d^{(k-1)})$ //

  $\hat{y}_n \leftarrow f^{(k)}(x_n), \forall n$

  $\hat{\epsilon}^{(k)} \leftarrow \sum_n d_n^{(k-1)} [y_n \neq \hat{y}_n]$

  $\alpha^{(k)} \leftarrow \frac{1}{2} \log \left(\frac{1-\hat{\epsilon}^{(k)}}{\hat{\epsilon}^{(k)}} \right)$

  $d_n^{(k)} \leftarrow \frac{1}{Z} d_n^{(k-1)} \exp[-\alpha^{(k)} y_n \hat{y}_n], \forall n$

end for

return $f(\hat{x}) = \text{sgn} \left[\sum_k \alpha^{(k)} f^{(k)}(\hat{x}) \right]$

CS 691 Midterm 2 Name Solution

Short Problems (20 points)

1. In the following statement, underline and indicate which part of this refers to recall, and which part refers to to precision. (4 points)
"I Swear To Tell The Truth, The Whole Truth and Nothing But The Truth So Help Me God"

(1) (2)

(1) Recall

(2) Precision

2. Let's say we have a single layer neural network. Our data has N samples and D features. We have computed the gradient of our Loss function with respect to our weights. In Big-O notation, what is the cost of one gradient descent update given the gradient? (4 points)

$O(D)$

3. Suppose we take all the weights and biases in a network of perceptrons, and multiply them by a positive constant, $c > 0$. Show that the behavior of the network does not change. Use mathematical notation to show this. Do not simply reason about it. (4 points)

$net = \sum V_i h_i^{end}$ $\quad h_i^{end} = \sum w_i^{end-1} h_i^{end-1}$ $\quad h_i^{1} = \sum w_i^{1} x_i$ $\quad h_i^{1} = \sum c w_i^{1} x_i$

$h_i^{1} = Dc h_i^{1}$

$h_i^{end} = \sum c w_i^{end} (k h_i^{end})$

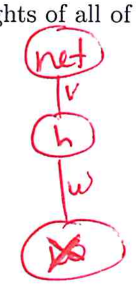
k constant. $\quad$ net = constant net.

$c > 0$ behavior does not change.

4. AdaBoost is not susceptible to outliers. True or False. If your answer is true, explain why. If your answer is false, then describe a simple way to fix Adaboost so that it is not susceptible to outliers. (4 points)

If we repeatedly misclassify a sample we eventually toss it out.

5. In a multi-layered neural network, if the activation of a hidden unit is zero, then the gradients of the weights of all of its incoming connections are zero. True or False. No need to explain. (4 points)



$net = h \cdot v \quad h = w \cdot x$

$$\frac{\partial net}{\partial w} = \frac{\partial net}{\partial h}\frac{\partial h}{\partial w} = v \cdot x$$

$$\frac{\partial net}{\partial v} = h = 0$$

only if a small change in w produces no change in h.

10 points

6. Consider the binary threshold neuron, $h = sign(w^T x)$ defined such that $h \in \{0, 1\}$, with no bias b or w_0. Consider the following set of four input features, x:

$$(1,0,0)^T,\ (0,1,0)^T,\ (0,0,1)^T,\ (1,1,1)^T$$

(a) Find a three-dimensional parameter vector w such that the neuron will have the output pattern $\{h\} = \{1, 1, 1, 1\}$ for the given four input features.

$w = (1, 1, 1)$

(b) Find a three-dimensional parameter vector w such that the neuron will have the output pattern $\{h\} = \{1, 1, 0, 0\}$ for the given four input features.

$w = (1, 1, -3)$

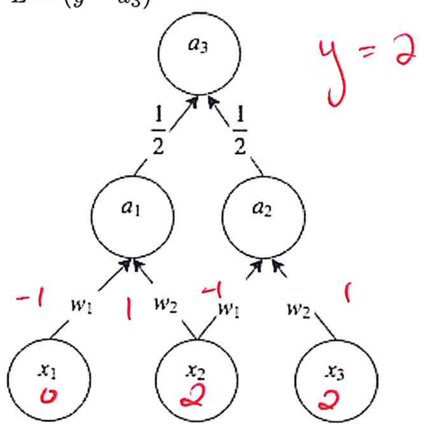
(c) Find an unrealizable output pattern $\{h\}$.

$h = \{1, 1, 1, 0\}$

or $\{0, 0, 0, 1\}$

20 points

7. We depict a neural network below. There is weight sharing in the first level. This means that the two edges that are labeled w_1 are required to have the same value. If we change one, we change the other by the same amount. Similarly for w_2. There are no bias terms. Between the hidden layer and the output layer there are two edges labeled $\frac{1}{2}$. This is average pooling. These weights are fixed and never change. Backprop does not affect them. We also use a regression loss. So the network can be described with the equations:
$a_1 = w_1x_1 + w_2x_2$
$a_2 = w_1x_2 + w_2x_3$
$a_3 = \frac{1}{2}a_1 + \frac{1}{2}a_2$
$L = (y - a_3)^2$



We have one training example with $x_1 = 0$, $x_2 = 2$, $x_3 = 2$, $y = 2$. We initialize the network with $w_1 = -1$, $w_2 = 1$. For this training example:

(a) What are the values of a_1, a_2, a_3, and L?

4pt.

$a_1 = -1\cdot 0 + 1\cdot 2 = 2$ $\quad$ $a_3 = \frac{1}{2}\cdot 2 + \frac{1}{2}\cdot 0 = 1$

$a_2 = -1\cdot 2 + 1\cdot 2 = 0$ $\quad$ $L = (2-1)^2 = 1$

(b) What is $\frac{\partial L}{\partial a_3}$?

4pt.

$$\frac{\partial L}{\partial a_3} = -2(y - a_3) = -2(2-1) = -2$$

(c) What is $\frac{\partial a_3}{\partial a_1}$?

4pt.

$$\frac{\partial a_3}{\partial a_1} = \frac{1}{2}$$

(d) What is $\frac{\partial a_1}{\partial w_1}$?

4pt.

$$\frac{\partial a_1}{\partial w_1} = x_1 = 0$$

(e) What is $\frac{\partial L}{\partial w_1}$?

4pt.

forgot this?

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial a_3}\left[\frac{\partial a_3}{\partial a_1}\frac{\partial a_1}{\partial w_1} + \overbrace{\frac{\partial a_3}{\partial a_2}\frac{\partial a_2}{\partial w_1}}^{-2}\right]$$

$$-2\left[\frac{1}{2}\cdot 0 + \frac{1}{2}\cdot 2\right] = -2$$

10 points

8. Suppose we perform PCA on a set of 2D points, in which all points have the form: $a_i u$ for a fixed vector u and for different values of a_i. Show that when we perform PCA we will get u as ~~the~~ a principal component. Do not use centering or variance scaling. Use mathematical notation to show this. Do not simply reason about it.

$$Z = \begin{bmatrix} a_1 u_1 & a_1 u_2 \\ a_2 u_1 & a_2 u_2 \\ a_3 u_1 & a_3 u_2 \\ a_4 u_1 & a_4 u_2 \\ \vdots & \end{bmatrix}$$

$$Z^T = \begin{matrix} a_1 u_1 & a_2 u_1 & a_3 u_1 & a_4 u_1 \\ a_1 u_2 & a_2 u_2 & a_3 u_2 & a_4 u_2 \cdots \end{matrix}$$

$$Z^T Z = \begin{bmatrix} \sum_N a_i^2 u_1^2 & \sum a_i^2 u_1 u_2 \\ \sum a_i^2 u_1 u_2 & \sum a_i^2 u_2^2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$Z^T Z u = \lambda u$?

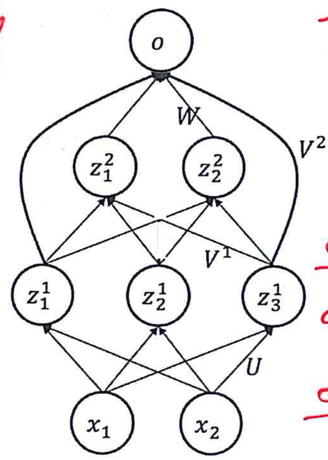
$$= \begin{bmatrix} \sum a_i^2 u_1^3 + \sum a_i^2 u_1 u_2^2 \\ \sum a_i^2 u_1^2 u_2 + \sum a_i^2 u_2^3 \end{bmatrix}$$

$$= \left(\sum a_i^2 u_1^2 + \sum a_i^2 u_2^2\right) \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

eigenvector!

20 points

9. Consider the following four layered network architecture presented. The input layer (X) is connected to the first hidden layer(z^1) through the weights U. The connection between first hidden layer(z^1) and second hidden layer(z^2) is defined using the set of weights V^1. There is also a skip connection from the first hidden layer(z^1) to the output layer(o). These skip connections are weighted by V^2. The weights of the connections between the second hidden layer(z^2) and the output layer(o) is defined by W. Derive the weight update equation for the weights of the connections from the input layer to the first hidden layer i.e., the weight updates for U. Remember each u_{ij} is going to i from j. Both the hidden layers employ sigmoid activation $\left(S(x) = \frac{1}{1+e^{-x}} \text{ with } S'(x) = S(x)(1-S(x))\right)$ and the output layer applies no nonlinearity. The network employs $L = \frac{1}{2}(y-o)^2$. Your answer should have a weight update for each u_{ij}



$$\frac{\partial L}{\partial u_{11}} = \frac{\partial L}{\partial o}\left[\frac{\partial o}{\partial z^2_1}\cdot\frac{\partial z^2_1}{\partial z^1_1}\,\frac{\partial z^1_1}{\partial u_{11}} + \frac{\partial o}{\partial z^2_2}\,\frac{\partial z^2_2}{\partial z^1_1}\,\frac{\partial z^1_1}{\partial u_{11}} + \frac{\partial o}{\partial z^1_1}\,\frac{\partial z^1_1}{\partial u_{11}}\right]$$

Same for u_{12}

Similar for u_{31} u_{32}

$$\frac{\partial L}{\partial u_{21}} = \frac{\partial L}{\partial o}\left[\frac{\partial o}{\partial z^2_1}\cdot\frac{\partial z^2_1}{\partial z^1_2}\,\frac{\partial z^1_2}{\partial u_{21}} + \frac{\partial o}{\partial z^2_2}\cdot\frac{\partial z^2_2}{\partial z^1_2}\cdot\frac{\partial z^1_2}{\partial u_{21}}\right]$$

Same for u_{21}

$$\frac{\partial L}{\partial o} = o - y$$

$$\frac{\partial o}{\partial z^2_1} = w_{11} \qquad \frac{\partial o}{\partial z^2_2} = w_{12}$$

$$\frac{\partial z^2_1}{\partial z^1_1} = S(v^1\cdot z^1)(1-S(v^1\cdot z^1))\cdot v^1_{11}$$

$$\frac{\partial z^2_1}{\partial z^1_2} = S(v^1\cdot z^1)(1-S(v^1\cdot z^1))\cdot v^1_{12}$$

$$\frac{\partial z^2_1}{\partial z^1_3} = \text{"} \quad \text{"} \cdot v^1_{13}$$

$$\frac{\partial z^2_2}{\partial z^1_1} = \text{"} \quad \text{"}\ v^1_{21}$$

$$\frac{\partial z^2_2}{\partial z^1_2} = \text{"} \quad \text{"}\ v^1_{22}$$

$$\frac{\partial z^2_2}{\partial z^1_3} = \text{"} \quad \text{"}\ v^1_{23}$$

$$\frac{\partial z^1_1}{\partial u_{1i}} = x_i \cdot S(x\cdot u)(1-S(x\cdot u))$$

$$\frac{\partial o}{\partial z^1_1} = v^2_{11} \qquad \frac{\partial o}{\partial z^1_3} = v^2_{13}$$

$$\frac{\partial z^1_2}{\partial u_{2i}} = x_i \cdot \text{"} \quad \text{"}$$

$$\frac{\partial z^1_3}{\partial u_{3i}} = x_i \cdot \text{"} \quad \text{"}$$

$$u_{11} = u_{11} - \eta \frac{\partial L}{\partial u_{11}} = u_{11} - \eta (0-y)\Big[w_{11} S(v' \cdot z')(1 - S(v' \cdot z')) v'_{11} \cdot x_1 S(x \cdot u)(1 - S(x \cdot u)) + w_{12} S(v' \cdot z')(1 - S(v' \cdot z')) v'_{21} \cdot x_2 S(x \cdot u)(1 - S(x \cdot u)) \Big]$$

and so on...

gross.

20 points

10. In this problem, you will use Adaboost to learn a hidden function from this set of training examples. We will use two rounds of AdaBoost to learn a hypothesis for this data set. The Adaboost algorithm is provided on the back of this page for reference. Recall that in round number 1, AdaBoost chooses a weak learner that minimizes the weighted error ϵ. As weak learners, you will use axis parallel lines of the form

- if $x_1 > a$, then +1 else −1 or
- if $x_2 > b$, then +1 else −1, for some integers a, b.

(either one of these two forms, not a combination of the two).

Consider the following labeled data (x_1, x_2, y) where x_1 and x_2 are the attributes and y is the class variable:

sample	x_1	x_2	y
s_1	11	3	-1
s_2	10	1	-1
s_3	4	4	-1
s_4	12	10	+1
s_5	2	4	-1
s_6	10	5	+1
s_7	8	8	-1
s_8	6	5	+1
s_9	7	7	+1
s_{10}	7	8	+1

(a) The first step of AdaBoost is to create an initial data weight distribution D_1 (also called calculating the data weighting co-efficients). What are the initial weights given to data points s_4 and s_7 by the AdaBoost algorithm, respectively?

$\frac{1}{10}$ each.

(b) Which of the following three hypotheses minimizes the weighted error in the first round of AdaBoost, using the distribution D_1 computed in the above question? Circle one. Justify your answer.

$x_2 > 9$ $\quad$ $x_2 > 4$ $\quad$ $x_2 > 7$

gives error of $\frac{1}{10}$ lowest error (S_7)

(c) What is the weighted error ϵ of the best classifier computed above in part (b)?

$\frac{1}{10}$

(d) Which of the following three hypotheses minimizes the weighted error in the second round of AdaBoost. Circle one. Justify your answer.

$x_2 > 9$ $\quad$ $x_1 > 5$ $\quad$ $x_2 > 7$

$X_2 > 9$ 4 wrong $= \frac{4}{30}$ error

$X_1 > 5$ 3 wrong $= \frac{2}{30} + \frac{3}{10}$

$X_2 > 7$ ~~4~~ 3 wrong $= \frac{2}{30} + \frac{3}{10}$

$\alpha = \frac{1}{2}\log\left(\frac{1-\frac{1}{10}}{\frac{1}{10}}\right) = \frac{1}{2}\log(9)$

$e^{-\log\sqrt{9}} = \log(3)$

$= \frac{1}{3} \cdot \frac{1}{10} = \frac{1}{30}$ for all correct.

$\frac{3}{10}$ for S_7

Algorithm 32 ADABOOST($\mathcal{W}$, $\mathcal{D}$, K)

$d^{(0)} \leftarrow \langle \frac{1}{N}, \frac{1}{N}, \dots, \frac{1}{N} \rangle$ // Initialize ı
for $k = 1 \dots K$ **do**
  $f^{(k)} \leftarrow \mathcal{W}(\mathcal{D}, d^{(k-1)})$ //
  $\hat{y}_n \leftarrow f^{(k)}(x_n), \forall n$
  $\hat{\epsilon}^{(k)} \leftarrow \sum_n d_n^{(k-1)} [y_n \neq \hat{y}_n]$
  $\alpha^{(k)} \leftarrow \frac{1}{2} \log \left(\frac{1-\hat{\epsilon}^{(k)}}{\hat{\epsilon}^{(k)}} \right)$
  $d_n^{(k)} \leftarrow \frac{1}{Z} d_n^{(k-1)} \exp[-\alpha^{(k)} y_n \hat{y}_n], \forall n$
end for
return $f(\hat{x}) = \text{sgn}\left[\sum_k \alpha^{(k)} f^{(k)}(\hat{x})\right]$