CS 691 Midterm 2 Name _____

## 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"

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)

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)

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)

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)

691

## 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.

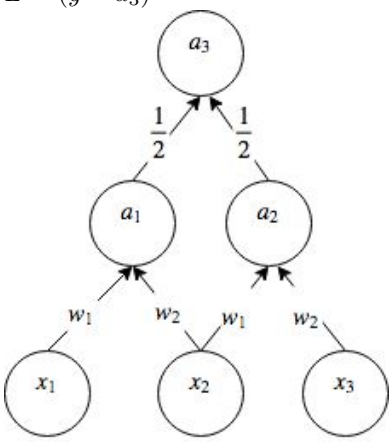
   (b) Find a three-dimensional parameter vector $w$ such that the neuron will have the output pattern $\{h\} = \{1, 1, 0, 0\}$ for the given four input features.

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

691

## 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:
$a_1 = w_1 x_1 + w_2 x_2$
$a_2 = w_1 x_2 + w_2 x_3$
$a_3 = \frac{1}{2} a_1 + \frac{1}{2} a_2$
$L = (y - a_3)^2$




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 $L$?

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

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

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

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

691

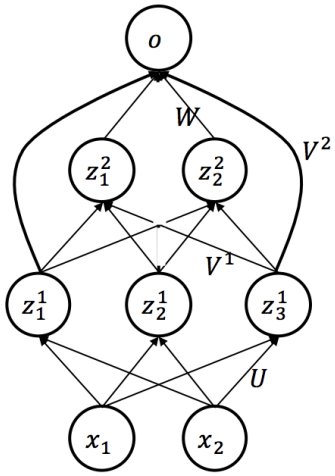
## 10 points

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.

691

## 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 $u_{ij}$



691

## 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 $\epsilon$. 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?

    (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.
    $x_2 > 9$ $x_2 > 4$ $x_2 > 7$

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

    (d) Which of the following three hypotheses minimizes the weighted error in the second round of AdaBoost. Circle one. Justify your answer.
    $x_2 > 9$ $x_1 > 5$ $x_2 > 7$

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

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