CS 491 Midterm 2 Name ____________________________

Short Problems (20 points)

1. Adaboost accounts for outliers by lowering the weights of training points that are repeatedly misclassified. True or False? Circle one and explain. (5 points)

2. The more hidden-layer units a deep neural network has, the better it can predict desired outputs for *new* inputs that it was *not* trained with. True or False? Circle one and explain. (5 points)

3. With a supervised learning algorithm, we can specify target output values, but we may never get close to those targets at the end of learning. Give one reason why this might happen. (5 points)

4. What problem(s) will result from using a learning rate that's too high? How would you detect these problems? (5 points)

491

Neural Network Parameters (10 points)

5. Consider a convolution layer. The input consists of 6 feature maps, each of size 20×20. The output consists of 8 feature maps, and the filters are of size 5×5. The convolution is done with a stride of 2 and zero padding, so the output feature maps are of size 10×10. For both parts, you can leave your expression as a product of integers; you do not need to actually compute the product. You do not need to show your work, but doing so can help you receive partial credit. *Don't forget about the bias terms!*

   (a) Determine the number of weights in this convolution layer. (5 points)

   (b) Now suppose we made this a fully connected layer, but where the number of input and output units are kept the same as in the network described above. Determine the number of weights in this layer. (5 points)

491

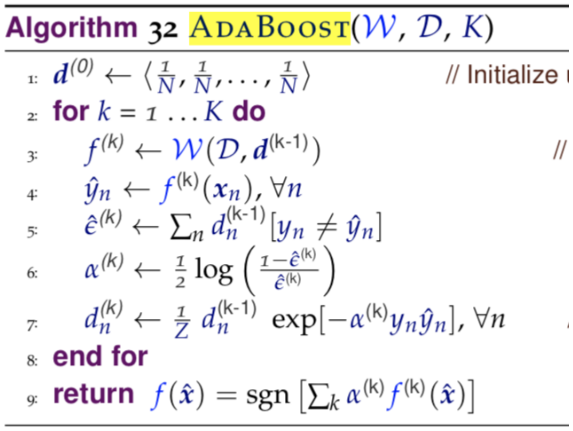
Adaboost (20 points)

6. You have six training points (A, B, C, D, E, F) and five classifiers (h1, h2, h3, h4, h5) which make the following misclassifications:

Classifier	Misclassified training points (A, B, C, D, E, F)				
h_1	A			D	F
h_2				D	
h_3		B	C		
h_4	A	B			F
h_5		B	C	D	

Perform two rounds of boosting with these classifiers and training data. In each round, pick the classifier with the **lowest error rate**. Break ties by picking the classifier that comes first in this list: h_1, h_2, h_3, h_4, h_5. Space for scratch work is provided on the back of the page.

	Round 1	Round 2
weight_A	1/6	
weight_B		
weight_C		
weight_D		
weight_E		
weight_F		
Error rate of h_1		
Error rate of h_2		
Error rate of h_3		
Error rate of h_4		
Error rate of h_5		
weak classifier (h)		
classifier error (ε)		



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 $f(\hat{\boldsymbol{x}}) = \mathrm{sgn}\left[\sum_k \alpha^{(k)} f^{(k)}(\hat{\boldsymbol{x}})\right]$
```

Neural Network Training (15 points)

7. Write down the back propagation algorithm. For full credit, your description must be clear enough for someone who knows what a multilayer perceptron is, to implement the algorithm. The steps of your algorithm should be clearly listed. If you have more than 10 steps, you have too many steps.

491

PCA (15 points)

8. Use the covariance matrix below to answer the following questions.

$$Z^T Z = \begin{bmatrix} 5 & 1 \\ 4 & 5 \end{bmatrix}$$

(a) Find the first principal component of the data. (10 points)

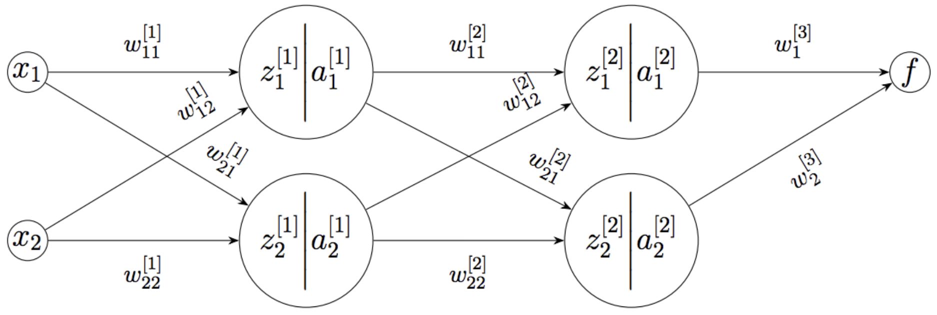
(b) Project the following sample onto the first principal component. (5 points)

$$X = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

491

Back Propagation (20 points)

9. Consider the three layer network below. Given that $f = w_1^{[3]} a_1^{[2]} + w_2^{[3]} a_2^{[2]}$, and $\sigma(z) = z^2$, compute $\frac{\partial f(x)}{\partial w_{11}^{[1]}}$. USE THE CHAIN RULE for partial credit. Keep this clean by expressing the derivatives in terms of the chain rule and then defining each derivative separately. You do not have to use matrix notation.



$$Z^{[1]} = \begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \end{bmatrix} = \begin{bmatrix} w_{11}^{[1]} & w_{12}^{[1]} \\ w_{21}^{[1]} & w_{22}^{[1]} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad , \quad A^{[1]} = \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \end{bmatrix} = \begin{bmatrix} \sigma(z_1^{[1]}) \\ \sigma(z_2^{[1]}) \end{bmatrix}$$

$$Z^{[2]} = \begin{bmatrix} z_1^{[2]} \\ z_2^{[2]} \end{bmatrix} = \begin{bmatrix} w_{11}^{[2]} & w_{12}^{[2]} \\ w_{21}^{[2]} & w_{22}^{[2]} \end{bmatrix} \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \end{bmatrix} \quad , \quad A^{[2]} = \begin{bmatrix} a_1^{[2]} \\ a_2^{[2]} \end{bmatrix} = \begin{bmatrix} \sigma(z_1^{[2]}) \\ \sigma(z_2^{[2]}) \end{bmatrix}$$

491