CS 691 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. A *majority unit* receives binary input vectors and outputs a 1 if and only if a majority of the input components are 1; otherwise it outputs a 0. Completely describe a model of a single neuron that implements a majority unit for an arbitrary number of input components. (5 points)

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

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 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)

691

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

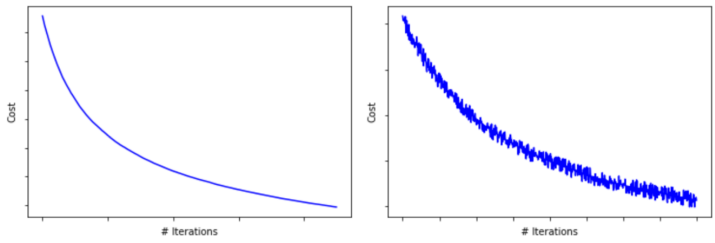
Neural Network Training (15 points)

7. Answer the following short-answer questions regarding neural network training.

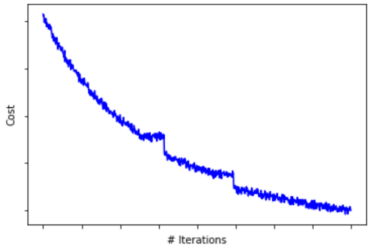
   (a) What problem(s) will result from using a learning rate that's too high? How would you detect these problems? (3 points)

   (b) What problem(s) will result from using a learning rate that's too low? How would you detect these problems? (3 points)

   (c) The figure below shows how the cost (loss) decreases (as the number of iterations increases) when two different optimization algorithms are used for training. Which of the graphs corresponds to using batch gradient descent as the optimization algorithm and which one corresponds to using stochastic gradient descent? Explain. (6 points)



   (d) The figure below shows how the cost (loss) decreases (as the number of iterations increases) during training. What could have caused the sudden drop in the cost? Explain one reason. (3 points)



691

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)

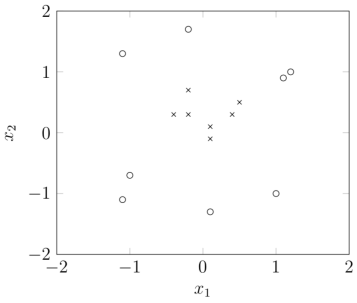
(b) Project the following sample onto the first principal component. (5 points)

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

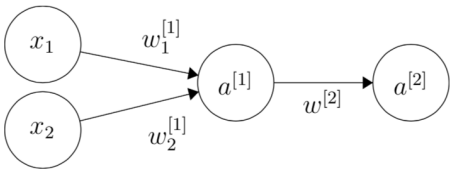
691

Back Propagation (20 points)

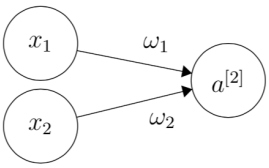
9. Consider the data below. Let's begin by modeling this problem with a simple 2 layer network: with an activation function $g^{[1]}$ and a sigmoid output unit ($\sigma(z) = \frac{1}{1+e^{-z}}$).



$$z^{[1]} = w_1^{[1]}x_1 + w_2^{[1]}x_2 + b^{[1]}$$
$$a^{[1]} = g^{[1]}(z^{[1]})$$
$$z^{[2]} = w^{[2]}a^{[1]} + b^{[2]}$$
$$a^{[2]} = \sigma(z^{[2]})$$



(a) Show that if $g^{[1]}$ is a linear activation function, $g^{[1]}(z) = \alpha z$, then the above network can be reduced to the single layer network shown below. Give the new weights and bias for this network ω_1, ω_2, β_1.



(b) Give an expression for $\nabla_{w^{[1]}} a^{[2]}$. Assume $g(z) = z^2$.

691