CS 622 Exam 2 Name _____

Short Answer (10 Points)

1. Give two reasons why we might choose the ReLU activation function over a sigmoid activation function in a deep neural network? (5 points)

2. In theory, we only need 2 layers for a neural network to approximate any function. Why does this not work that well in practice, and what can we do to improve performance? (5 points)

Short Answer (10 Points)

3. If I give you one principal component in 2D can you recover the other one with no additional information? If yes, how? If no, why not? (5 points)

4. Do duplicate samples (two samples with the same features) affect the output of PCA? Briefly explain. (5 points)

Adaboost (20 Points)

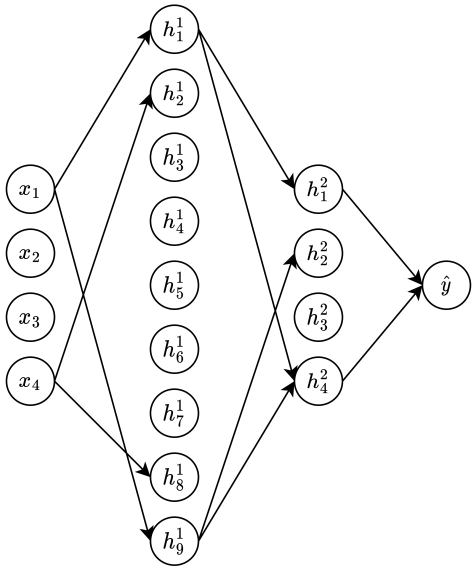
5. Assume you have four data points and you have four classifiers. The table below details which samples each classifier incorrectly classifies. Run two full iterations of adaboost, choosing the classifier with the lowest error at each iteration. Break ties by choosing h_i over h_k if $i < k$. In the table provided, indicate which classifier you chose, it's error rate and the weights for reach sample.

Classifier	Incorrect
h_1	s_1
h_2	s_1, s_2
h_3	s_2, s_3
h_4	s_4

Sample	Iter 0	Iter 1	Iter 2
s_1			
s_2			
s_3			
s_4			
h			
Error			

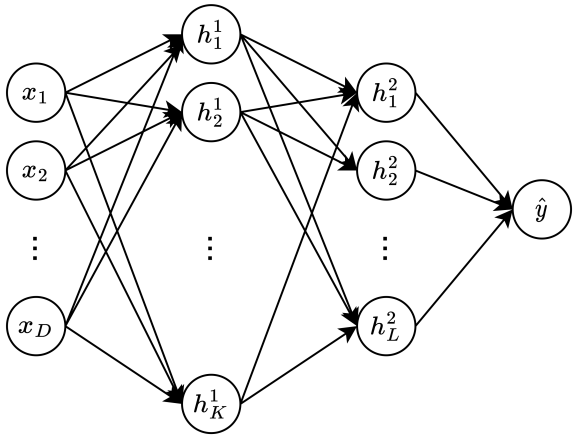
Deep Learning (10 Points)

6. Convert the following fully connected network to an equivalent convolutional neural network. That is, the networks should be equivalent in the number and application of weights to inputs, and hidden nodes. Assume the network is fully connected. I just didn't include all the connections because it takes too long to draw. Explain your choices.



Neural Networks (20 Points)

7. Given the following neural network, give the general formula for updating a weight in the first layer (w^1_{ji} the weight going from input i to hidden node h^1_j). Weights in the second layer are w^2_{kj} (the weight going from hidden node h^1_j to h^2_k) and weights in the third layer are w^3_k (the weight going from h^2_k to $\hat{y}$). Assume your non-linearity is $z = (h)^2$ (you can think of z as being the output of each hidden node, where you just square the h value) and your loss is $L = (y - \hat{y})^2$. For the formula, do not simply write $\frac{\partial L}{\partial \hat{y}} \times \ldots$, plug in the expressions for these. Please keep everything clean and clear for partial credit!



CNNs (10 Points)

8. Assume you have a CNN with one layer (input to output). The input is of size $10 \times 10 \times 5$. Assume your output is of size $4 \times 4 \times 3$ Assume you have 1 pixel of zero-padding on each side of the image. What size is your filter? How many parameters does this CNN have? (No need to include bias terms).

PCA (20 Points)

9. Use PCA to find the principal components of the following data. No need to center or standardize the data. Project the following point onto the first principal component: $(3, 3)$. Project that same point onto both of the principal components.

Sample	x_1	x_2	y
s_1	0	1	1
s_2	1	0	1
s_3	1	1	1
s_4	2	2	-1
s_5	2	1	-1
s_6	1	2	-1

Equations

Adaboost

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

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

PCA

$$Cov = (X - \bar{X})(X - \bar{X})^T$$

$$|A - \lambda I| = 0$$

$$X^* = UX$$