

# CS 156a Set 6

Cason Shepard

January 18, 2023

## problem 1

In general, deterministic noise will increase. This is because When  $H' \subset H$ , we won't have the full scope of the  $H$  that is used to approximate  $f$ , thus, noise in our approximation will generally increase.

The answer is [b]

The following Linear Regression algorithm was used/manipulated to answer questions 2-6.

```
import numpy
import random

N = len(input_data)
in_points = [] * N
classification_list = [0] * N
class_check = [0] * N

for i in range(0,N):
    x1 = input_data[i][0]
    x2 = input_data[i][1]
    y = input_data[i][2]
    in_points.append([1, x1, x2, x1*x1, x2*x2, x1*x2, abs(x1-x2), abs(x1+x2)])
    classification_list[i] = y

k = -2
lamb = 10**k
X_points = numpy.array(in_points)
y = numpy.array(classification_list)
first_half = numpy.linalg.pinv(numpy.add(numpy.matmul(X_points.T, X_points), lamb*numpy.identity(8)))
w = numpy.matmul(first_half, numpy.matmul(X_points.T, y))

for idx in range(0, N):
    if numpy.dot(in_points[idx], w) > 0:
        class_check[idx] = 1
    else:
        class_check[idx] = -1

diff = count_diff(classification_list, class_check)
E_in = diff/N

N_out = len(out_data)
out_transform = []
class_out = [0]*N_out
out_points = []
classification_check = [0]*N_out

for i in range(0, N_out):
    x1 = out_data[i][0]
    x2 = out_data[i][1]
    y = out_data[i][2]
    out_points.append([1, x1, x2, x1*x1, x2*x2, x1*x2, abs(x1-x2), abs(x1+x2)])
    class_out[i] = y

for idx in range(0, N_out):
    if numpy.dot(out_points[idx], w) > 0:
        classification_check[idx] = 1
    else:
        classification_check[idx] = -1
```

```

diff = count_diff(class_out, classification_check)
E_out = diff/N_out
E_list.append(E_out)

print(E_in)
print(E_out)

```

## problem 2

After running Linear Regression on the data provided, I got an  $E_{in}$  of 0.0286 and an  $E_{out}$  of 0.084, which is closest to a.

The answer is [a]

## problem 3

After running Linear Regression with the weight decay on the data provided (with  $k = -3$ ), I got an  $E_{in}$  of 0.0286 and an  $E_{out}$  of 0.08, which is closest to d.

The answer is [d]

## problem 4

After running Linear Regression with the weight decay on the data provided (with  $k = 3$ ), I got an  $E_{in}$  of 0.37 and an  $E_{out}$  of 0.436, which is closest to e.

The answer is [e]

## problem 5

After running Linear Regression with the weight decay on the data provided, whilst varying  $k$ , I got the  $E_{in}$  and  $E_{out}$  as reported below.

- (a)  $k = 2, E_{in} = 0.2, E_{out} = 0.228$
- (b)  $k = 1, E_{in} = 0.057, E_{out} = 0.124$
- (c)  $k = 0, E_{in} = 0.0, E_{out} = 0.092$
- (d)  $k = -1, E_{in} = 0.029, E_{out} = 0.056$
- (e)  $k = -2, E_{in} = 0.29, E_{out} = 0.084$

The smallest  $E_{out}$  was found when  $k = -1$ .

The answer is [d]

## problem 6

When  $k$  was varied from  $[-20, 20]$ , the smallest value for  $E_{out}$  was 0.056. This is closest to 0.06.

The answer is [b]

## problem 7

- (a)  $\mathcal{H}(10, 0, 3) \cup \mathcal{H}(10, 0, 4) = \mathcal{H}_4$   
In this option,  $\mathcal{H}(10, 0, 3)$  represents a hypothesis set containing polynomials of degree 2 or less.  $\mathcal{H}(10, 0, 4)$  represents a hypothesis set containing polynomials of degree 3 or less. This option is not true as the union of these two sets does not represent  $\mathcal{H}_4$ .
- (b)  $\mathcal{H}(10, 1, 3) \cup \mathcal{H}(10, 1, 4) = \mathcal{H}_3$   
In this option,  $\mathcal{H}(10, 1, 3)$  represents a hypothesis set containing polynomials up to degree 9.  $\mathcal{H}(10, 1, 4)$  also represents a hypothesis set containing polynomials up to degree 9. This option is not true as the union of these two sets does not represent only polynomials up to degree 3 as denoted by  $\mathcal{H}_3$ .
- (c)  $\mathcal{H}(10, 0, 3) \cap \mathcal{H}(10, 0, 4) = \mathcal{H}_2$   
In this option,  $\mathcal{H}(10, 0, 3)$  represents a hypothesis set containing polynomials of degree 2 or less.  $\mathcal{H}(10, 0, 4)$  represents a hypothesis set containing polynomials of degree 3 or less. **This option is true as the intersection of these two sets represents a hypothesis set with polynomial of degree 2 or less as denoted by  $\mathcal{H}_2$ .**

(d)  $\mathcal{H}(10, 1, 3) \cap \mathcal{H}(10, 1, 4) = \mathcal{H}_1$

In this option,  $\mathcal{H}(10, 1, 3)$  represents a hypothesis set containing polynomials up to degree 9.  $\mathcal{H}(10, 1, 4)$  also represents a hypothesis set containing polynomials up to degree 9. This option is not true as the intersection of these two sets does not represent only polynomials up to degree 0 as denoted by  $\mathcal{H}_1$ . The intersection of these two sets will contain polynomials of degree up to 9.

The answer is [c]

## problem 8

First to find the number of weights,  $|W| = 6 * 3 + 4 * 1 = 22$ . Now, we know that the forward operations and weight updates must require 22 updates. Such gives us:

$$w_{ij}^{(l)} x_i^{(l-1)} = x_i^{(l-1)} \delta_j^{(l)} = 22$$

The only thing left to compute is now the backward propagation as described by  $w_{ij}^{(l)} \delta_j^{(l)}$ . In this case, we only have 3 layers:  $d^{(0)}, d^{(1)}, d^{(2)}$ . We need only to calculate the back ward propagation for the middle of these layers, as denoted by  $0 < l < L - 1$ . Thus, the number of operations for this step is:

$$\begin{aligned} w_{ij}^{(l)} \delta_j^{(l)} &= \sum_{i=1}^{d^{(1)}} d^{(2)} \\ w_{ij}^{(l)} \delta_j^{(l)} &= \sum_{i=1}^3 1 \\ w_{ij}^{(l)} \delta_j^{(l)} &= 3 \end{aligned}$$

The total of these is  $22 + 22 + 3 = 47$ , which is closest to 45.

The answer is [d]

## problem 9

For the first 10 input units, we know we must have an equal amount of weights. Thus, we must have at least 10 weights. Now we must find the minimum amount of nodes for the rest of the hidden layers. We know that hidden layers can be arranged into layers of minimum 2 units, so the max number of hidden layers can be  $\frac{36}{2} = 18$ . Since our weights only act between the connections of these units, we can have a max of 17 connection spaces between these 18 units. This allows us to minimize the number of weights, as we can assume each of these 2-unit spaces requires 2 weights. Thus the number of weights for the hidden layers in this scenario is  $17 * 2 = 34$ . Between the final hidden layer and the output, we will have one last transformation, requiring 2 more weights. This brings the total weights of the minimum configuration to be:

$$10 + 34 + 2 = 46$$

The answer is [a]

## problem 10

For this network, we will construct it in a way that will maximize the weights required. Consider the 10 input units, followed by 2 hidden layer containing 18 units. In this situation, the number of weights required for each the input→hidden, hidden→hidden, and hidden→output levels will be as follows:

$$\begin{aligned} w_{input \rightarrow hidden} &= 10(18 - 1) = 170 \\ w_{hidden \rightarrow hidden} &= 18(18 - 1) = 306 \\ w_{hidden \rightarrow output} &= 18 \\ w_{total} &= 170 + 306 + 18 = 494 \end{aligned}$$

If our network contained 1 hidden layer or 3 hidden layers, our  $w_{total}$  would equal 386, which is less than 494. Thus, 494 is our max number of weights for this scenario.

The answer is [c]