HW6

November 1, 2023

0.1 CS156A Homework 6

0.2 Wilson Duan

0.2.1 Problem 1.

Since H' is a subset of H, the size of H' is smaller than H, meaning H' represents a smaller set of hypotheses than H. As a result, H' is less representative of the target function f, meaning deterministic noise will increase in general. Thus, the answer is **b)**.

0.2.2 Problem 2.

[25]:
```
import numpy as np
```

[12]:
```
in_data = []
out_data = []

with open("in.dta", "r") as f:
    for line in f:
        line = line.strip().split()
        line = [float(x) for x in line]
        in_data.append(line)

with open("out.dta", "r") as f:
    for line in f:
        line = line.strip().split()
        line = [float(x) for x in line]
        out_data.append(line)

in_data = np.array(in_data)
out_data = np.array(out_data)
```

[43]:
```
def transform_data(data):
    output = np.zeros((len(data), 8))
    for i in range(len(data)):
        x1, x2 = data[i]
        output[i] = [1, x1, x2, x1**2, x2**2, x1*x2, abs(x1-x2), abs(x1+x2)]
    return output
```

[34]:
```
def linear_regression(X, y):
    inversed = np.linalg.inv(X.transpose().dot(X))
    w = inversed.dot(X.transpose()).dot(y)
    return w
```

[35]:
```
def linear_regression_decay(X, y, lmbda):
    inversed = np.linalg.inv(X.transpose().dot(X) + lmbda * np.eye(X.shape[1]))
    w = inversed.dot(X.transpose()).dot(y)
    return w
```

[45]:
```
def calculate_error(X, y, w):
    predictions = np.sign(X.dot(w))
    return np.mean(predictions != y)
```

[48]:
```
X_in = transform_data(in_data[:, :2])
y_in = in_data[:, 2]
X_out = transform_data(out_data[:, :2])
y_out = out_data[:, 2]

w = linear_regression(X_in, y_in)
print("In sample error:", calculate_error(X_in, y_in, w))
print("Out of sample error:", calculate_error(X_out, y_out, w))
```

```
In sample error: 0.02857142857142857
Out of sample error: 0.084
```

According to the code output above, the in sample and out of sample classification errors are closest to answer choice **a)** 0.03, 0.08.

0.2.3 Problem 3.

[53]:
```
k = -3
lmbda = 10 ** k

w = linear_regression_decay(X_in, y_in, lmbda)
print("In sample error:", calculate_error(X_in, y_in, w))
print("Out of sample error:", calculate_error(X_out, y_out, w))
```

```
In sample error: 0.02857142857142857
Out of sample error: 0.08
```

According to the code output above, the in sample and out of sample classification errors are closest to answer choice **d)** 0.03, 0.08.

0.2.4 Problem 4.

[54]:
```
k = 3
lmbda = 10 ** k
```

```
w = linear_regression_decay(X_in, y_in, lmbda)
print("In sample error:", calculate_error(X_in, y_in, w))
print("Out of sample error:", calculate_error(X_out, y_out, w))
```

```
In sample error: 0.37142857142857144
Out of sample error: 0.436
```

According to the code output above, the in sample and out of sample classification errors are closest to answer choice **e)** 0.4, 0.4.

0.2.5 Problem 5.

[56]:
```
k_vals = [2, 1, 0, -1, -2]

for k in k_vals:
    lmbda = 10 ** k

    w = linear_regression_decay(X_in, y_in, lmbda)
    print("Results for k = ", k)
    print("In sample error:", calculate_error(X_in, y_in, w))
    print("Out of sample error:", calculate_error(X_out, y_out, w))
    print()
```

```
Results for k =  2
In sample error: 0.2
Out of sample error: 0.228

Results for k =  1
In sample error: 0.05714285714285714
Out of sample error: 0.124

Results for k =  0
In sample error: 0.0
Out of sample error: 0.092

Results for k =  -1
In sample error: 0.02857142857142857
Out of sample error: 0.056

Results for k =  -2
In sample error: 0.02857142857142857
Out of sample error: 0.084
```

According to the code output above, the k that achieves the smallest out of sample classification error is answer choice **d)**.

0.2.6 Problem 6.

[58]:
```
k_vals = [2, 1, 0, -1, -2, -3, -4]

for k in k_vals:
    lmbda = 10 ** k

    w = linear_regression_decay(X_in, y_in, lmbda)
    print("Results for k = ", k)
    print("In sample error:", calculate_error(X_in, y_in, w))
    print("Out of sample error:", calculate_error(X_out, y_out, w))
    print()
```

```
Results for k =  2
In sample error: 0.2
Out of sample error: 0.228

Results for k =  1
In sample error: 0.05714285714285714
Out of sample error: 0.124

Results for k =  0
In sample error: 0.0
Out of sample error: 0.092

Results for k =  -1
In sample error: 0.02857142857142857
Out of sample error: 0.056

Results for k =  -2
In sample error: 0.02857142857142857
Out of sample error: 0.084

Results for k =  -3
In sample error: 0.02857142857142857
Out of sample error: 0.08

Results for k =  -4
In sample error: 0.02857142857142857
Out of sample error: 0.084
```

According to the code output above, the smallest out of sample classification error is closest to answer choice **b)** 0.06.

0.2.7 Problem 7.

In all of the answer choices, the hypothesis sets used are $H(10, 0, 3)$, $H(10, 0, 4)$, $H(10, 1, 3)$, and $H(10, 1, 4)$. We first analyze each hypothesis set.

$H(10, 0, 3)$ represents the set of all hypotheses in H_{10} such that all the weights for degree greater than or equal to 3 are zero. As a result, $H(10, 0, 3)$ only spans all polynomials with degree 2 or less. In other words, $H(10, 0, 3) = H_2$.

$H(10, 0, 4)$ represents the set of all hypotheses in H_{10} such that all the weights for degree greater than or equal to 4 are zero. As a result, $H(10, 0, 4)$ only spans all polynomials with degree 3 or less. In other words, $H(10, 0, 4) = H_3$.

$H(10, 1, 3)$ represents the set of all hypotheses in H_{10} such that all the weights for degree greater than or equal to 3 are 1. This means $H(10, 1, 3)$ contains all polynomials with degree 2 or less, as well as some polynomials with degree up to 10. In other words, $H_2 \subset H(10, 1, 3)$.

$H(10, 1, 4)$ represents the set of all hypotheses in H_{10} such that all the weights for degree greater than or equal to 4 are 1. This means $H(10, 1, 4)$ contains all polynomials with degree 3 or less, as well as some polynomials with degree up to 10. In other words, $H_3 \subset H(10, 1, 4)$.

Now we look at each answer choice.

a) $H(10, 0, 3) \cup H(10, 0, 4)$ equals the union of H_2 and H_3, which is not equal to H_4.

b) $H(10, 1, 3) \cup H(10, 1, 4)$ does not equal H_3 because both hypothesis sets contain polynomials with degree up to 10.

c) $H(10, 0, 3) \cap H(10, 0, 4) = H_2 \cap H_3 = H_2$, so this is the correct answer.

d) $H(10, 1, 3) \cap H(10, 1, 4)$ contains polynomials up to degree 10, so it does not equal H_1.

The answer is **c)**.

0.2.8 Problem 8.

Since $d^{(0)} = 5$, each of the 3 neurons in the hidden layer has 5 + 1 associated weights, which makes $6 * 3 = 18$ weights in the hidden layer. Since $d^{(1)} = 3$, the neuron in the last layer has 3 + 1 associated weights, which makes for 4 weights in the last layer. As a result, there are 22 weights in this system. Note that the +1 comes from the bias at each neuron.

In the forward pass, the operations of the form $w_{ij}^{(l)} x_i^{(l-1)}$ occur a number of times equal to the number of weights, so these operations occur 22 times.

The operations of the form $w_{ij}^{(l)} \delta_j^{(l)}$ occur in the backwards pass:

$\delta_i^{(l-1)} = (1 - (x_i^{(l-1)})^2) \sum_{j=1}^{d^{(l)}} w_{ij}^{(l)} \delta_j^{(l)}$

For the input layer, there are no operations to account for since there is no signal. For the last layer, there are also no operations to account for since δ is known. In the hidden layer, there is only one i, and since the summation goes from $j = 1$ to $d^{(1)} = 3$, there are 3 operations.

The operations of the form $x_i^{(l-1)} \delta_j^l$ occur in the update step, which occurs a number of times equal to the number of weights (since each weight is updated in the update step). Therefore, operations of this form occur 22 times.

The total number of operations is $22 + 3 + 22 = 47$. This is closest to answer choice **d)**.

0.2.9 Problem 9.

In order for each layer to be fully connected, there must be $d^{(l)} * (d^{(l+1)} - 1)$ weights for each l. In order to minimize the sum of these products, we use layers with only two units (one of them is x_0). The number of layers becomes $36/2 = 18$ hidden layers. From the input layer to the first hidden layer, there are 10 weights because the input layer has dimension 10. From a hidden layer to the next layer, there are 2 weights (x_0 and another weight). Going from a hidden layer to another hidden layer occurs 17 times, and going from the last hidden layer to the output unit occurs 1 time, so that makes for $(17 + 1) * 2 = 36$ weights. Adding the number of weights, we get $10 + 36 = 46$ weights. Therefore, the answer is **a)**.

0.2.10 Problem 10.

The maximum possible number of weights occurs when there are 2 hidden layers, as that maximizes the sum of the products $d^{(l)} * (d^{(l+1)} - 1)$. This is maximized when we have 22 units in the first layer and 14 units in the second layer, giving us a total of $10 * (22 - 1) + 22 * (14 - 1) + 14 * 1 = 510$ weights. Therefore, the answer is **e)**.