Intro to Machine Learning HomeWork2 (536-02)

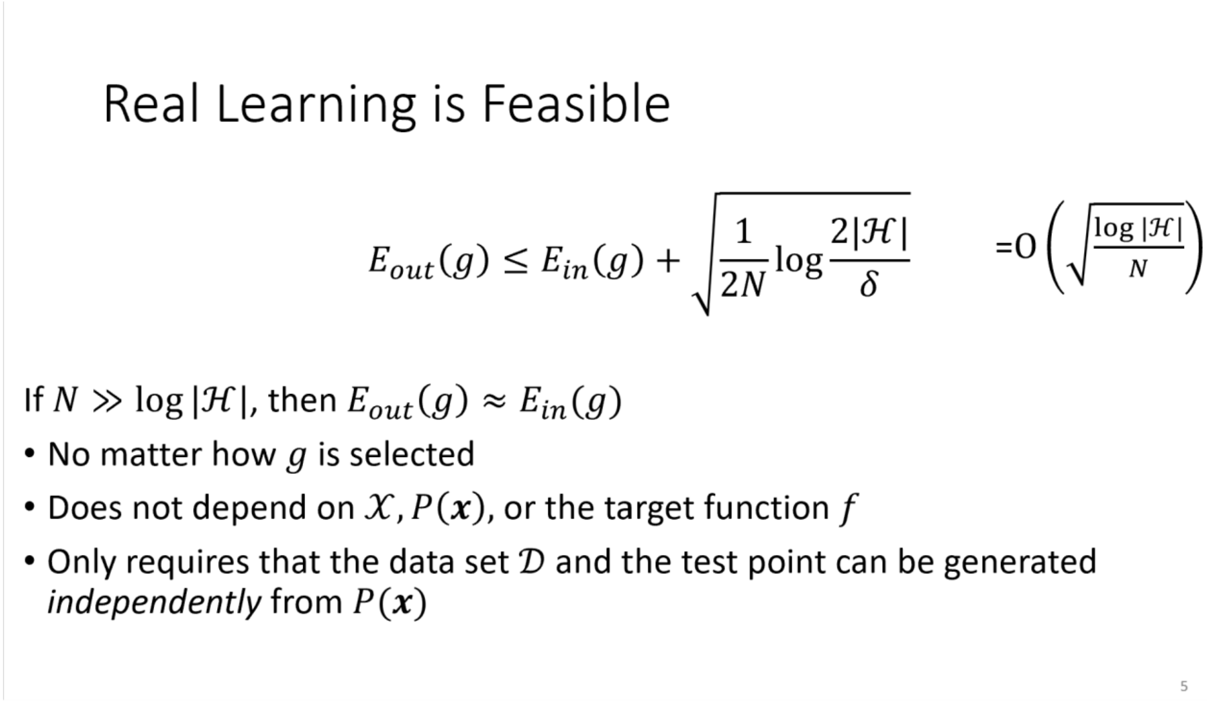
Saiprakash Nalubolu | B01037579 | snalubolu@binghamton.edu

https://colab.research.google.com/drive/1BdN9oHT8C_AiXvc1UpXxNuM5jbkUaK7p#scrollTo=9kn7s1HRPLiG

Q1: LFD Exercise 2.6:

We have a dataset of 600 examples. But we are going to train our final hypothesis g(out of 1000 hypothesis) from 400 training examples. The other 200 examples are randomly separated and are never used in the training phase.

With the help of the formula discussed in class(slides) we can estimate error bars.



Real Learning is Feasible

$$E_{out}(g) \le E_{in}(g) + \sqrt{\frac{1}{2N}\log\frac{2|\mathcal{H}|}{\delta}} \qquad = O\left(\sqrt{\frac{\log|\mathcal{H}|}{N}}\right)$$

If $N \gg \log|\mathcal{H}|$, then $E_{out}(g) \approx E_{in}(g)$

- No matter how g is selected
- Does not depend on $\mathcal{X}, P(\boldsymbol{x})$, or the target function f
- Only requires that the data set $\mathcal{D}$ and the test point can be generated *independently* from $P(\boldsymbol{x})$

5

$$E_{out}(g) \le E_{in}(g) + \underbrace{\sqrt{\frac{1}{2N}\log\frac{2|\mathcal{H}|}{\delta}}}_{\text{generalization error bar}}$$

$$\text{where } \delta = 2|\mathcal{H}|e^{-2\epsilon^2 N}$$

a. Given δ = 0.05.
   Calculating Error bar for Ein(g) : Number of training examples(N) = 400
   Number of Hypothesis(|H|) = 1000
   Error bar for Ein(g) = $\sqrt{\frac{1}{2(400)} log \frac{2(1000)}{0.05}}$ = $\sqrt{\frac{1}{800} log \frac{40000}{1}}$ = 0.115

Calculating error bar for Etest(g): N = 200

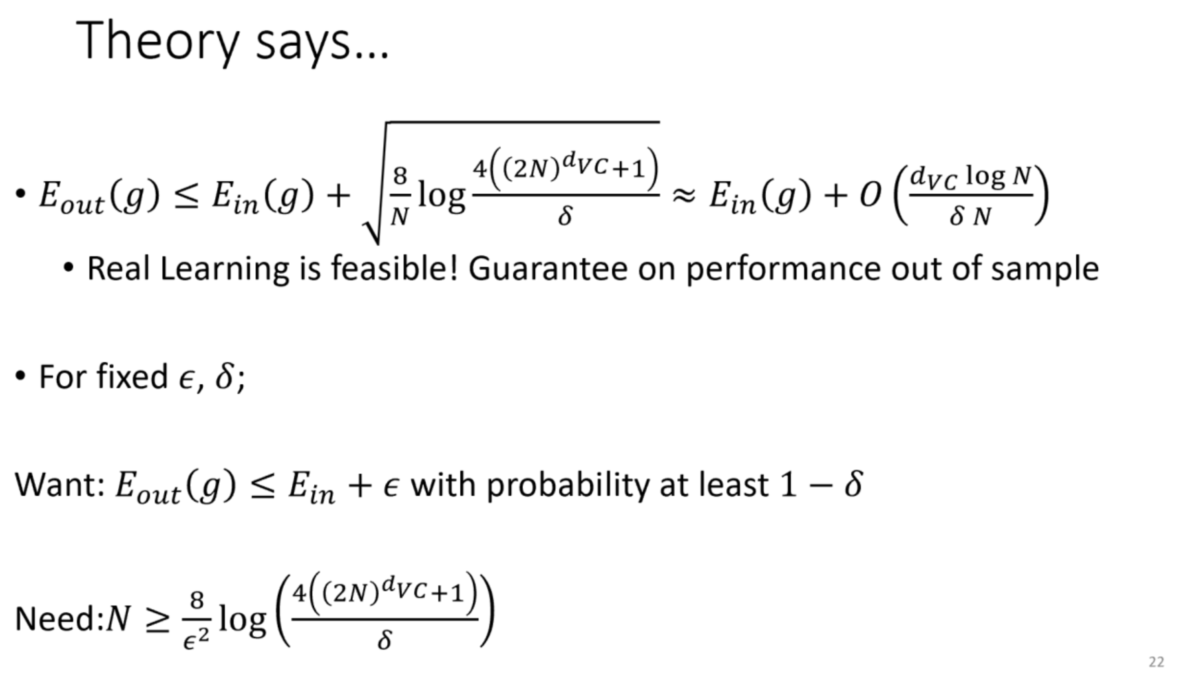
Error bar for Etest(g) = $\sqrt{\frac{1}{2(200)} log \frac{2}{0.05}}$ = $\sqrt{\frac{1}{400} log \frac{40}{1}}$ = 0.096

Out of the two estimates Ein(g), the in sample error on the 400 training examples **has the higher error bar than** Etest(g), the test error on the 200 test examples that were set aside.

b. Though we are seeing test error(Etest(g)) is less, when compared to In sample error(Ein(g)) we should not reserve more examples for testing. Because when we reserve more examples for testing the training phase is affected. We will have less data to train and we might not be able to get a good hypothesis close to our target function. So it is essential to have a good balance between number of examples reserved for testing and those used for training. Too few training examples could lead to a poorly trained model (underfitting), while too few test examples might result in an unreliable performance estimate. The goal is to ensure that we have enough data for both training the model well and accurately assessing its performance, without overly compromising on either aspect.

Q2: LFD Problem 2.12:

Based on the theory discussed in class and respective slides, the problem can be solved.



Theory says…

- $E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N} \log \frac{4\left((2N)^{d_{VC}}+1\right)}{\delta}} \approx E_{in}(g) + O\left(\frac{d_{VC} \log N}{\delta\, N}\right)$
  - Real Learning is feasible! Guarantee on performance out of sample
- For fixed ϵ, δ;

Want: $E_{out}(g) \leq E_{in} + \epsilon$ with probability at least $1 - \delta$

Need: $N \geq \frac{8}{\epsilon^2} \log\left(\frac{4\left((2N)^{d_{VC}}+1\right)}{\delta}\right)$

22

Confidence is 95% which implies 1-δ = 0.95 => δ = 0.05.

∈ = 0.05. dvc = 10. So sample size(N) I need to have is defined as N ≥ $\frac{8}{\epsilon^2}$ log($\frac{4\left((2N)^{dvc}+ 1\right)}{\delta}$)

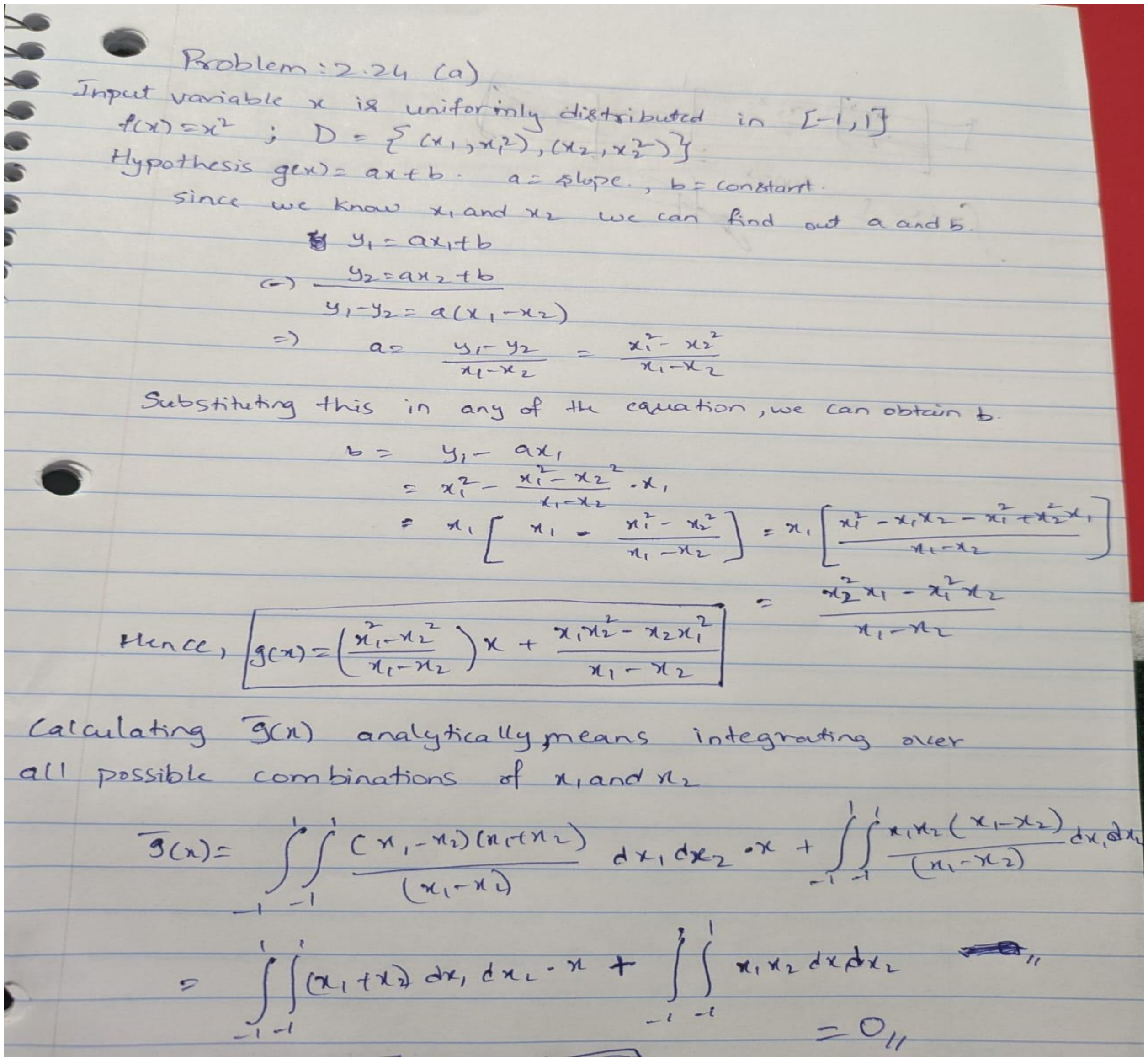
To solve this inequality I have created a python code since it involves guessing initial N values as we have N in LHS and inside Log in RHS.

If initial guess is less, then the inequality doesn't make sense, we have to take higher guess. A sample is provided in the code(available in Link provided in top of the assignment).

```
The minimum sample size N according to the generalization bound is: 452957
```

In the end, the code successfully found that with a sample size of approximately 452,957 examples, I can achieve a 95% confidence level and the generalization error will not exceed 0.05, given the VC dimension (complexity of the hypothesis set) is 10.

Q3: LFD Problem 2.24:

a. 

Problem : 2.24 (a)

Input variable x is uniformly distributed in $[-1,1]$

$f(x) = x^2$; $D = \{(x_1, x_1^2), (x_2, x_2^2)\}$

Hypothesis $g(x) = ax + b$. a = slope, b = constant.

since we know x_1 and x_2 we can find out a and b.

$$y_1 = ax_1 + b$$
$$(-)\ \underline{y_2 = ax_2 + b}$$
$$y_1 - y_2 = a(x_1 - x_2)$$
$$\Rightarrow \quad a = \frac{y_1 - y_2}{x_1 - x_2} = \frac{x_1^2 - x_2^2}{x_1 - x_2}$$

Substituting this in any of the equation, we can obtain b.

$$b = y_1 - ax_1$$
$$= x_1^2 - \frac{x_1^2 - x_2^2}{x_1 - x_2} \cdot x_1$$
$$= x_1\left[x_1 - \frac{x_1^2 - x_2^2}{x_1 - x_2}\right] = x_1\left[\frac{x_1^2 - x_1x_2 - x_1^2 + x_2^2 x_1}{x_1 - x_2}\right]$$
$$= \frac{x_2^2 x_1 - x_1^2 x_2}{x_1 - x_2}$$

Hence, $\boxed{g(x) = \left(\frac{x_1^2 - x_2^2}{x_1 - x_2}\right)x + \frac{x_1x_2^2 - x_2x_1^2}{x_1 - x_2}}$

Calculating $\bar{g}(x)$ analytically means integrating over all possible combinations of x_1 and x_2

$$\bar{g}(x) = \int_{-1}^{1}\int_{-1}^{1} \frac{(x_1 - x_2)(x_1 + x_2)}{(x_1 - x_2)}\, dx_1\, dx_2 \cdot x + \int_{-1}^{1}\int_{-1}^{1} \frac{x_1x_2(x_1 - x_2)}{(x_1 - x_2)}\, dx_1\, dx_2$$

$$= \int_{-1}^{1}\int_{-1}^{1} (x_1 + x_2)\, dx_1\, dx_2 \cdot x + \int_{-1}^{1}\int_{-1}^{1} x_1x_2\, dx\, dx_2$$

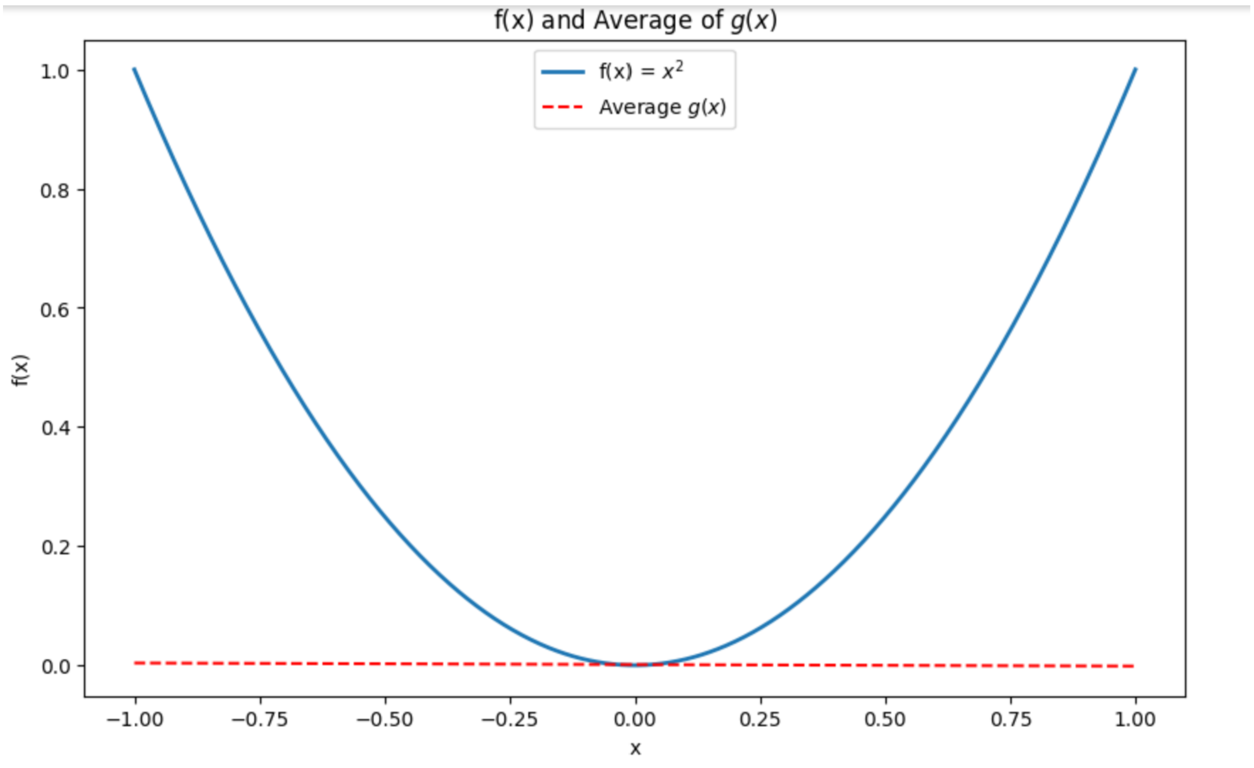
$$= 0$$

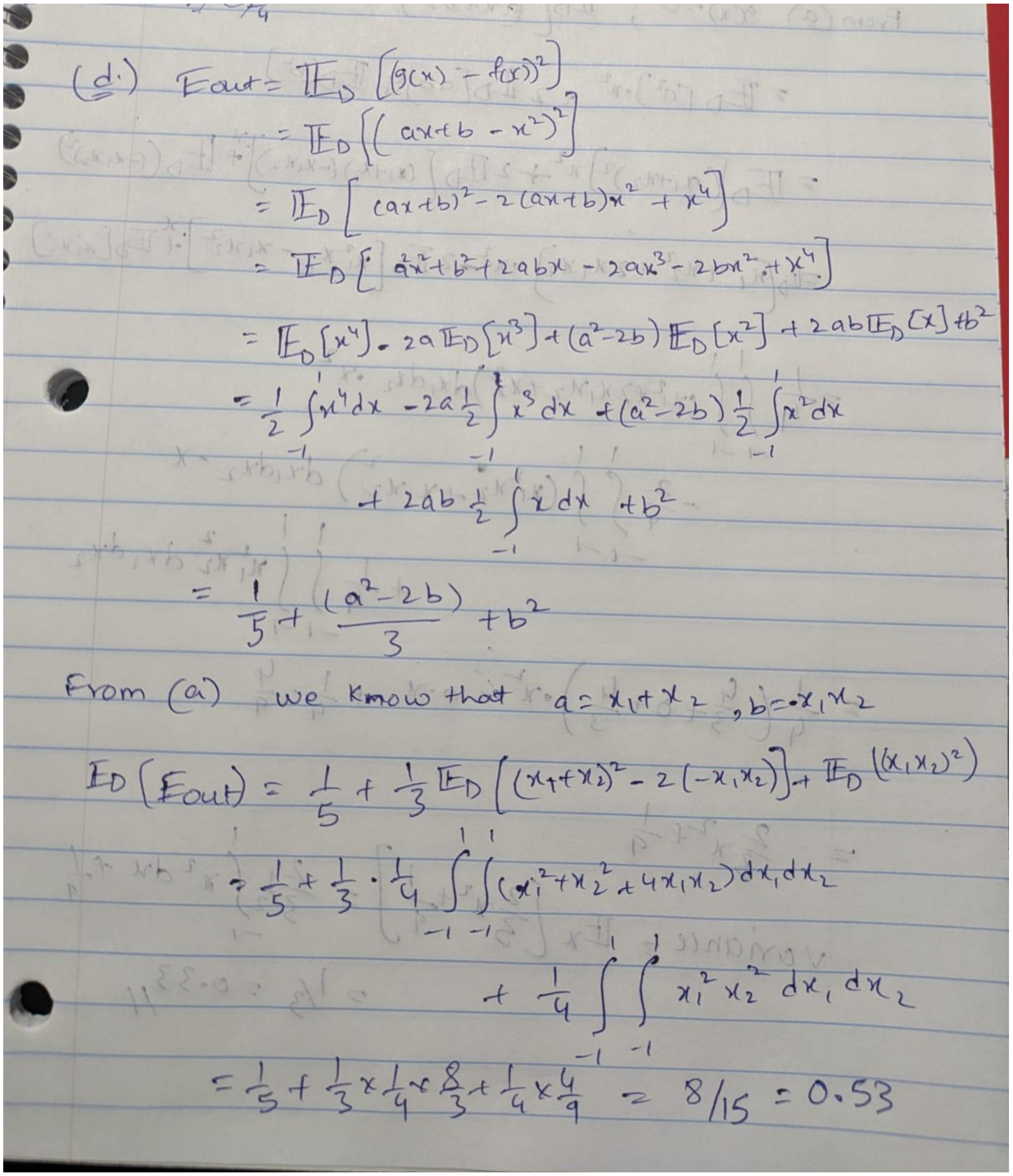
Since f(x) is odd function(f(-a)=-f(a)), both the double integrals will be 0.
Hence $\overline{g}(x)$ = 0.

b. We consider a sample model where input x is distributed uniformly in the interval [-1,1] and the target function f(x) = x^2. The learning algorithm tries to fit a linear model g(x) = ax+b to 2 points sampled from this distribution. The goal is to understand how well this linear model can approximate the quadratic target function, focusing on Eout, bias and variance.
A sample python code is made to understand the problem. Eout, Bias and variance are calculated.

```
Average g(x): -0.0026x + 0.0012
Bias: 0.20003104022624024
Variance: 0.3329088655857814
E_out: 0.5329399058120211
```

c. In the experiment I performed, Eout is 0.53. Bias is 0.2. Variance is 0.33.
Bias + Var = 0.2+0.33 = 0.53
Eout = 0.53.
⇨ Eout = Bias + Var.
A plot to show $\overline{g}(x)$ and **f(x)**





/4

(d.) $E_{out} = \mathbb{E}_D\left[(g(x) - f(x))^2\right]$

$$= \mathbb{E}_D\left[(ax+b-x^2)^2\right]$$

$$= \mathbb{E}_D\left[(ax+b)^2 - 2(ax+b)x^2 + x^4\right]$$

$$= \mathbb{E}_D\left[a^2x^2 + b^2 + 2abx - 2ax^3 - 2bx^2 + x^4\right]$$

$$= \mathbb{E}_D[x^4] - 2a\,\mathbb{E}_D[x^3] + (a^2-2b)\,\mathbb{E}_D[x^2] + 2ab\,\mathbb{E}_D[x] + b^2$$

$$= \frac{1}{2}\int_{-1}^{1} x^4\,dx - 2a\frac{1}{2}\int_{-1}^{1} x^3\,dx + (a^2-2b)\frac{1}{2}\int_{-1}^{1} x^2\,dx + 2ab\frac{1}{2}\int_{-1}^{1} x\,dx + b^2$$

$$= \frac{1}{5} + \frac{(a^2-2b)}{3} + b^2$$

From (a) we know that $a = x_1 + x_2$, $b = -x_1x_2$

$$\mathbb{E}_D(E_{out}) = \frac{1}{5} + \frac{1}{3}\mathbb{E}_D\left[(x_1+x_2)^2 - 2(-x_1x_2)\right] + \mathbb{E}_D\left((x_1x_2)^2\right)$$

$$= \frac{1}{5} + \frac{1}{3}\cdot\frac{1}{4}\int_{-1}^{1}\int_{-1}^{1}(x_1^2+x_2^2+4x_1x_2)\,dx_1\,dx_2 + \frac{1}{4}\int_{-1}^{1}\int_{-1}^{1} x_1^2x_2^2\,dx_1\,dx_2$$

$$= \frac{1}{5} + \frac{1}{3}\times\frac{1}{4}\times\frac{8}{3} + \frac{1}{4}\times\frac{4}{9} = 8/15 = 0.53$$

d.

$$\mathrm{var}(x) = \mathbb{E}_D\left[(g(x) - \bar{g}(x))^2\right]$$

From (a) $\bar{g}(x) = 0$; $\mathbb{E}_D\left[(ax+b)^2\right]$ is variance

$$= \mathbb{E}_D[a^2]\cdot x^2 + 2\,\mathbb{E}_D[ab]x + \mathbb{E}_D[b^2]$$

$$= \mathbb{E}_D\left[(x_1+x_2)^2\right]x^2 + 2\,\mathbb{E}_D\left[(x_1+x_2)(-x_1x_2)\right]\cdot x + \mathbb{E}_D\left((-x_1x_2)^2\right)$$

$$= \mathbb{E}_D\left[x_1^2 + x_2^2 + 2x_1x_2\right]\cdot x^2 - 2\,\mathbb{E}_D\left[x_1^2x_2 - x_1x_2^2\right]\cdot x + \mathbb{E}_D\left[x_1^2x_2^2\right]$$

$$= \frac{1}{4}\int_{-1}^{1}\int_{-1}^{1}(x_1^2 + 2x_1x_2 + x_2^2)\,dx_1\,dx_2 \cdot x^2$$

$$- \frac{2}{4}\int_{-1}^{1}\int_{-1}^{1}(x_1^2x_2 + x_1x_2^2)\,dx_1\,dx_2 \cdot x$$

$$+ \frac{1}{4}\int_{-1}^{1}\int_{-1}^{1}x_1^2x_2^2\,dx_1\,dx_2$$

$$= \frac{1}{4}\left(\frac{4}{3} + 0 + \frac{4}{3}\right)\cdot x^2 - 0\cdot x + \frac{1}{4}\cdot\frac{4}{9}$$

$$= \frac{2}{3}x^2 + \frac{1}{9}$$

variance $= \mathbb{E}_x\left[\frac{2}{3}x^2 + \frac{1}{9}\right] = \frac{2}{3}\cdot\frac{1}{2}\int_{-1}^{1}x^2\,dx + \frac{1}{9}$

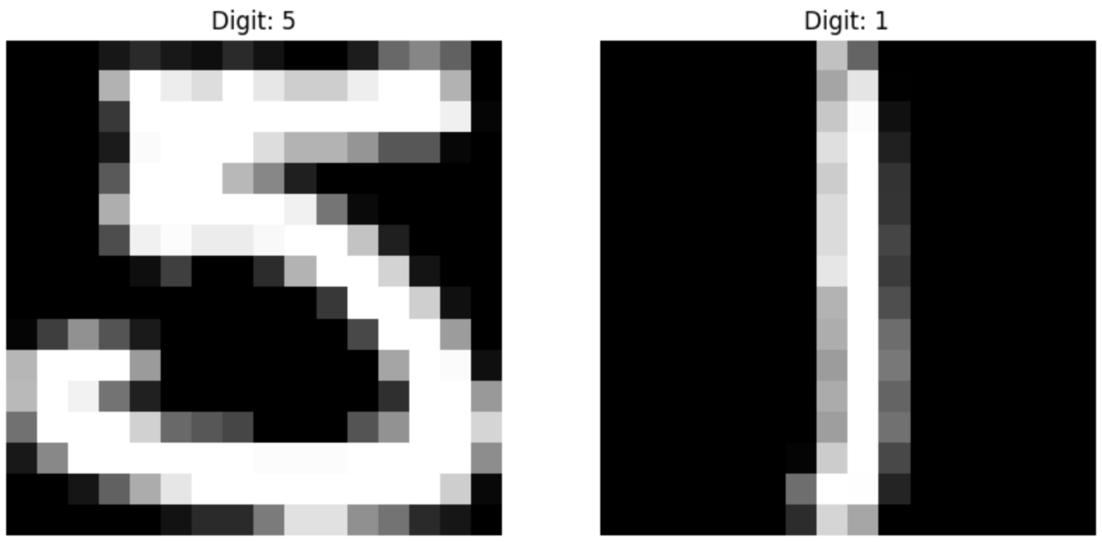
$$= 1/3 = 0.33$$

$$\mathrm{bias}(x) = (\bar{g}(x) - f(x))^2$$

From (a) $\bar{g}(x) = 0$; bias $= (f(x))^2$

$$\mathbb{E}_x\left[(x^2)^2\right] = \frac{1}{2}\int_{-1}^{1}x^4\,dx = \frac{1}{5} = 0.2$$

Q4: a.



b. I would like to use Average intensity and Symmetry features to distinguish between the digits 1 and 5.(as in example LFD 3.1)

Here is what mentioned in the example: (In reference to Learning from Data text book Example 3.1)

Let's look at two important features here: intensity and symmetry. Digit 5 usually occupies more black pixels than digit 1, and hence the average pixel intensity of digit 5 is higher. On the other hand, digit 1 is symmetric while digit 5 is not. Therefore, if we define asymmetry as the average absolute difference between an image and its flipped versions, and symmetry as the negation of asymmetry, digit 1 would result in a higher symmetry value.

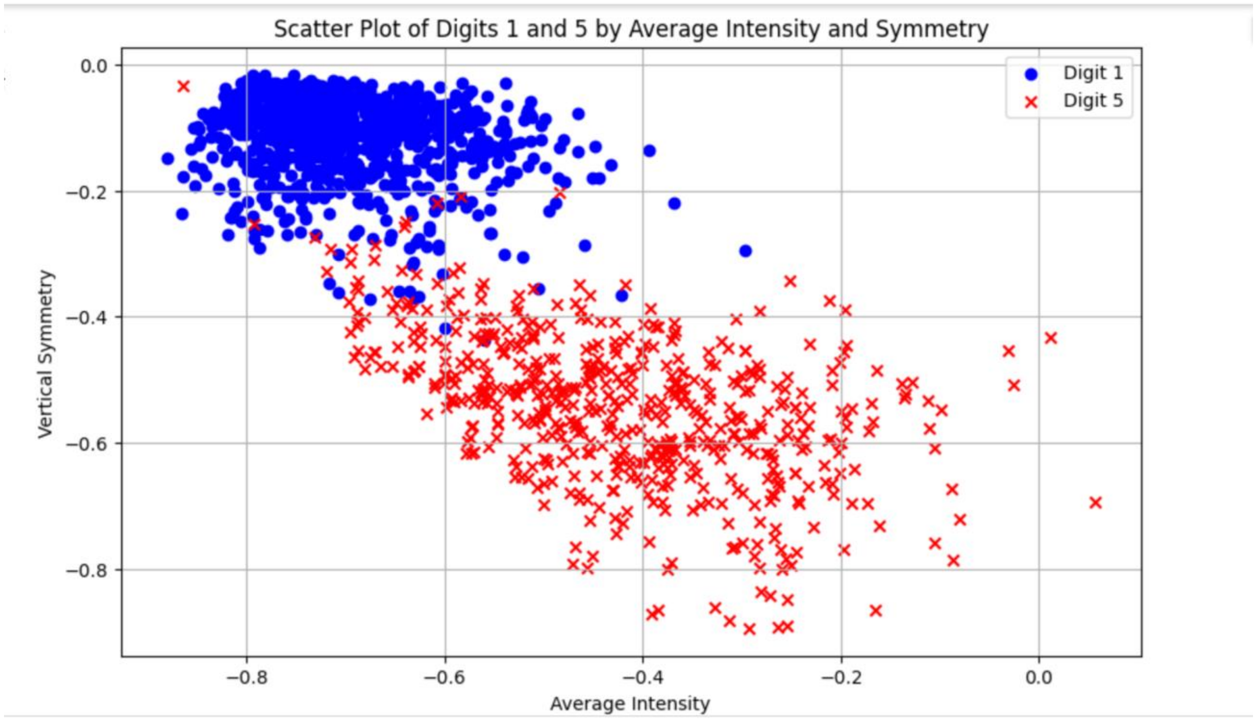
Average Intensity: The average intensity of an image can be computed as the mean of all pixel values in the image. Since we are dealing with images here lets consider it as a matrix M of size 16 X16 and average intensity

$$\text{Avg\_Intensity} = \frac{1}{256}\Sigma_{i=1}^{16}\Sigma_{j=1}^{16}M_{ij}$$

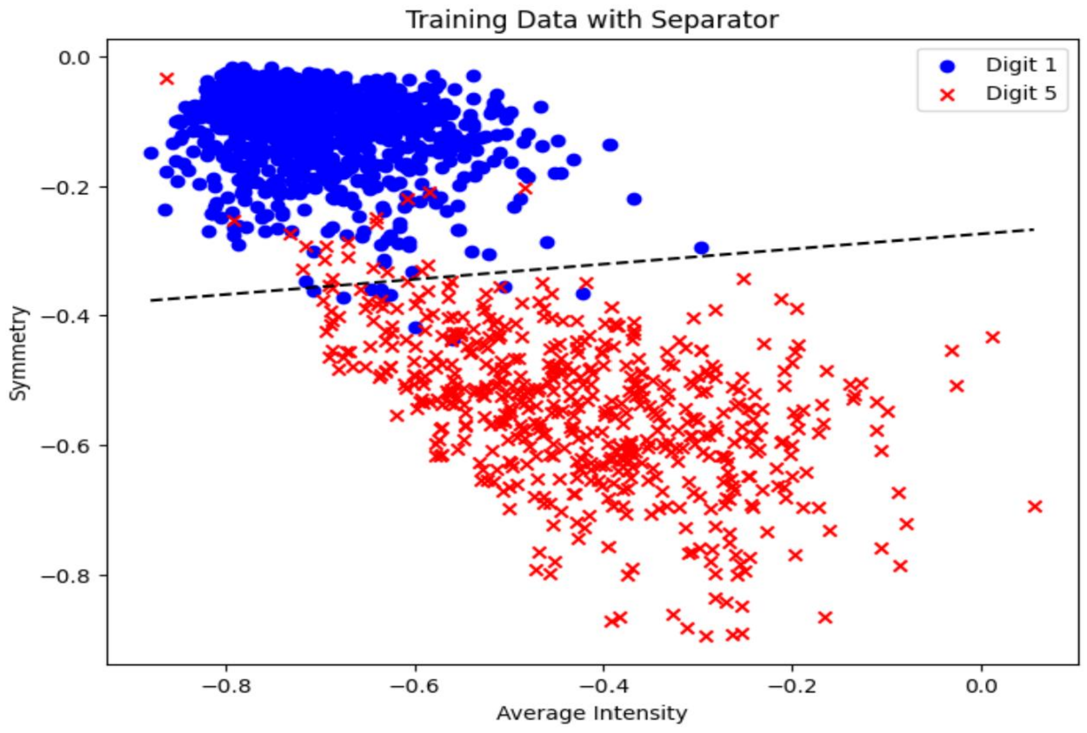
Symmetry: Symmetry is the negation of Assymetry. Where Assymetry is the average of difference between image and flipped versions. Hence Symmetry can be represented as:

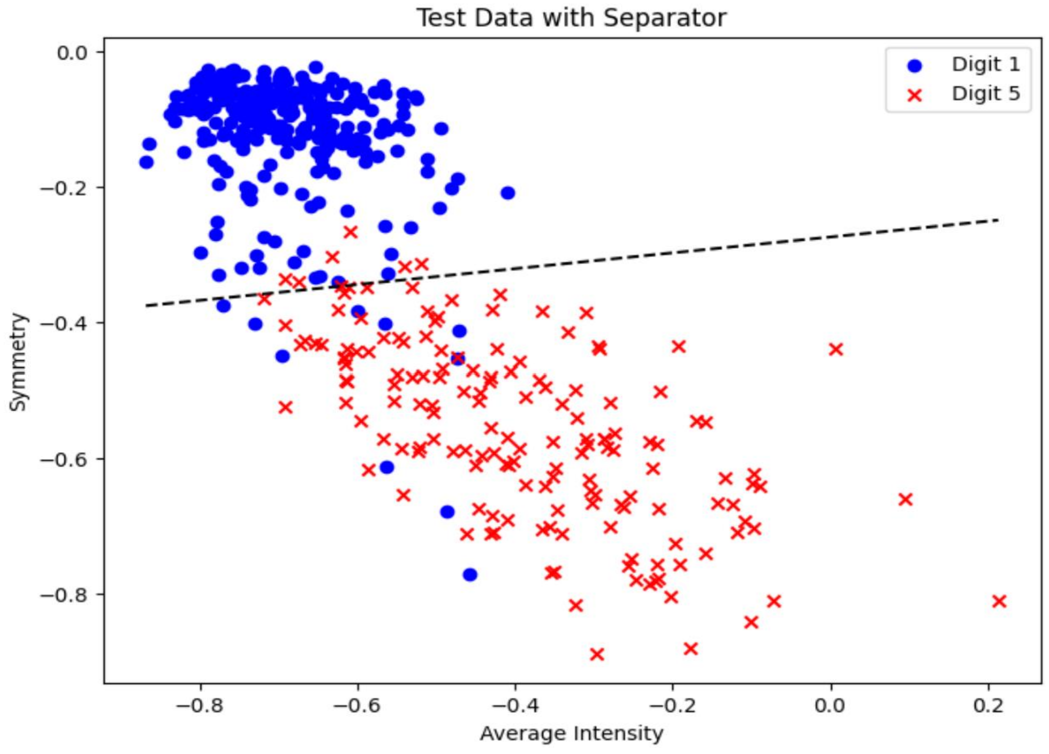
$$\text{Symmetry} = -\frac{1}{256}\Sigma_{i=1}^{16}\Sigma_{j=1}^{16}|M_{ij} - M'_{ij}| \text{ where } M'_{ij} \text{ is flipped version of } M_{ij}$$

c.

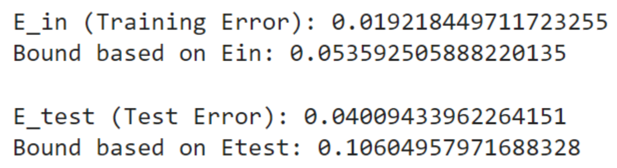

Scatter Plot of Digits 1 and 5 by Average Intensity and Symmetry
Digit 1
Digit 5
Vertical Symmetry
Average Intensity
0.0
−0.2
−0.4
−0.6
−0.8
−0.8
−0.6
−0.4
−0.2
0.0

Q5: (a)


Training Data with Separator
Digit 1
Digit 5
Symmetry
Average Intensity
0.0
−0.2
−0.4
−0.6
−0.8
−0.8
−0.6
−0.4
−0.2
0.0


Test Data with Separator
Digit 1
Digit 5
Symmetry
Average Intensity
0.0
−0.2
−0.4
−0.6
−0.8
−0.8
−0.6
−0.4
−0.2
0.0
0.2

(b) (c)



```
E_in (Training Error): 0.019218449711723255
Bound based on Ein: 0.053592505888220135

E_test (Test Error): 0.04009433962264151
Bound based on Etest: 0.10604957971688328
```

Bound based on Ein is 0.05 where as Bound based on Etest is 0.10

Therefore, the bound based on Ein is the better bound because it suggests a lower expected out-of-sample error(Eout), indicating that the model's performance on the training data is a closer estimate of how it will perform on new data, within δ = 0.05