HW 4:

(1)

$$\epsilon = \sqrt{\frac{8}{N}\ln\left(\frac{4m_{\mathcal{H}}(2N)}{\Delta}\right)}$$

$\Delta \geq 1 - .95$; $\Delta \geq 0.05$;

Maximum power of $m_{\mathcal{H}}(2N) = (2N)^{d_{vc}} = (2N)^{10}$

hence:

$$\epsilon = 0.05 = \sqrt{\frac{8}{N}\ln\left(\frac{4(2N)^{10}}{0.05}\right)};$$

$$\frac{N(0.05)^2}{8} = \ln\left(\frac{4 \cdot 2^{10} \cdot N^{10}}{0.05}\right)$$

$N = \pm 0.32, 452957$;

hence the answer is d bcz it is the closest to $N = 452957$;

(2) $d_{vc} = 50$; $\delta = 0.05$; $N = 10{,}000$; $m_{\mathcal{H}}(N) = N^{d_{vc}}$

See plots on page 2

[a]
$$\epsilon \leq \sqrt{\frac{8}{10{,}000}\ln\left(\frac{4(2(10{,}000))^{50}}{0.05}\right)};$$

$\epsilon \leq 0.632$;

[b]
$$\epsilon \leq \sqrt{\frac{2\ln(2(10{,}000)^{51})}{10000}} + \sqrt{\frac{2}{10{,}000}\ln\left(\frac{1}{0.05}\right)} + \frac{1}{10{,}000}$$

$\epsilon \leq 0.331$;

[c]
$$\left(\epsilon \leq \sqrt{\frac{1}{10000}\left(2\epsilon + \ln\left(\frac{6(2N)^{50}}{0.05}\right)\right)}\right)\sqrt{10{,}000}$$

$$100\epsilon \leq \sqrt{2\epsilon + 444.96}$$

$$100^2\epsilon^2 - 2\epsilon - 444.96 \leq 0;$$

$\epsilon \leq 0.2237$

[d]
$$\epsilon \leq \sqrt{\frac{1}{2(10{,}000)}\left((4\epsilon + 4\epsilon^2) + \ln\left(\frac{4(10{,}000)^{100}}{0.05}\right)\right)}$$

$$\epsilon^2 - \frac{1}{2(10{,}000)}\left((4\epsilon + 4\epsilon^2) + \ln\left(\frac{4(10{,}000)^{100}}{0.05}\right)\right) \leq 0;$$

$\epsilon \leq 0.215$

[e] Wrong bcz not all equal;

Hence the correct answer is d bcz it is the smallest ϵ;

(3) $N = 5$; $d_{vc} = 50$; $\delta = 0.05$; $m_{\mathcal{H}}(N) = N^{d_{vc}}$ (max) plugin:

[a]
$$\epsilon \leq \sqrt{\frac{8}{5}\ln\left(\frac{4(2 \cdot 5)^{50}}{0.05}\right)}$$

$\epsilon \leq 15.828$

[b]
$$\epsilon \leq \sqrt{\frac{2\ln(10(5)^{50})}{5}} + \sqrt{\frac{2}{5}\ln\left(\frac{1}{0.05}\right)} + \frac{1}{5}$$

$\epsilon \leq 7.049$

[c]
$$\epsilon \leq \sqrt{\frac{1}{5}\left(2\epsilon + \ln\left(\frac{6(10)^{50}}{0.05}\right)\right)}$$

$$\epsilon^2 - \frac{1}{5}2\epsilon - \frac{1}{5}\ln\left(\frac{6(2.5)^{50}}{0.05}\right) \leq 0;$$

$\epsilon \leq 5.101$

[d]
$$\epsilon \leq \sqrt{\frac{1}{10}\left((4\epsilon + 4\epsilon^2) + \ln\left(\frac{4(25)^{50}}{0.05}\right)\right)}$$

$$\epsilon^2 - \frac{1}{10}\left((4\epsilon + 4\epsilon^2) + \ln\left(\frac{4(25)^{50}}{0.05}\right)\right) \leq 0;$$

$\epsilon \leq 5.593$

[e] Wrong bcz they are not all equal.

Hence by the above analysis, the correct answer is C as it has the smallest ϵ given $N = 5$;

(4)

$$\left((ax_1 - \sin(\pi x))^2 + (ax_2 - \sin(\pi x))^2\right) = \epsilon$$

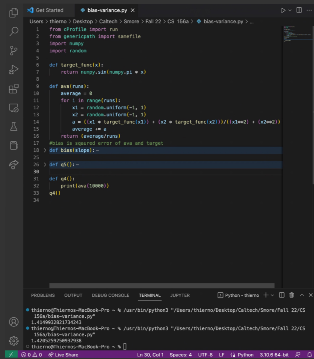
lowest when ϵ is smallest thus find derivative with respect to a.

$$2x_1(ax_1 - \sin(\pi x_1)) + 2x_2(ax_2 - \sin(\pi x_2)) = 0$$

$$2ax_1^2 - 2x_1\sin(\pi x_1) + 2ax_2^2 - 2x_2\sin(\pi x_2) = 0$$

$$a(2x_1^2 + 2x_2^2) = 2x_1\sin(\pi x_1) + 2x_2\sin(\pi x_2)$$

$$a = \frac{x_1\sin(\pi x_1) + x_2\sin(\pi x_2)}{x_1^2 + x_2^2}$$



```
Users > thierno > Desktop > Caltech > Smore > Fall 22 > CS 156a > bias-variance.py > ...
from cProfile import run
from genericpath import samefile
import numpy
import random

def target_func(x):
    return numpy.sin(numpy.pi * x)

def ava(runs):
    average = 0
    for i in range(runs):
        x1 = random.uniform(-1, 1)
        x2 = random.uniform(-1, 1)
        a = ((x1 * target_func(x1)) + (x2 * target_func(x2)))/((x1**2) + (x2**2))
        average += a
    return (average/runs)
#bias is sqaured error of ava and target
def bias(slope):⋯

def q5():⋯

def q4():
    print(ava(10000))
q4()
```

```
thierno@Thiernos-MacBook-Pro ~ % /usr/bin/python3 "/Users/thierno/Desktop/Caltech/Smore/Fall 22/CS 156a/bias-variance.py"
1.4149932821734243
thierno@Thiernos-MacBook-Pro ~ % /usr/bin/python3 "/Users/thierno/Desktop/Caltech/Smore/Fall 22/CS 156a/bias-variance.py"
1.4205259250932938
thierno@Thiernos-MacBook-Pro ~ %
```

hence according to the following code → the correct answer is E because after 10,000 runs the average $\hat{a}$ was 1.42 which does not match a–d;

(7) [a] $h(x) = b$ has a bias $= 0.50$ & a variance $= 0.25$ according to lecture 7 slide 15. Thus $E_{out} = Var + bias = 0.50 + 0.25 = 0.75$.

[b] $h(x) = ax$ has bias $= 0.27$ & Variance $= 0.20$ as shown in question 5 & 6 code hence $E_{out} = 0.27 + 0.20 = 0.47$;

[c] $h(x) = ax + b$ has a variance $= 1.69$ & bias $= 0.21$ as proven in lecture 7 slide 15 hence $E_{out} = 1.69 + 0.21 = 1.90$;

[d] $h(x) = ax^2$

$$a = \frac{x_1^2\sin(\pi x_1^2) + x_2^2\sin(\pi x_2^2)}{x_1^4 + x_2^4}$$

as we are taking derivative with respect to a hence similar.

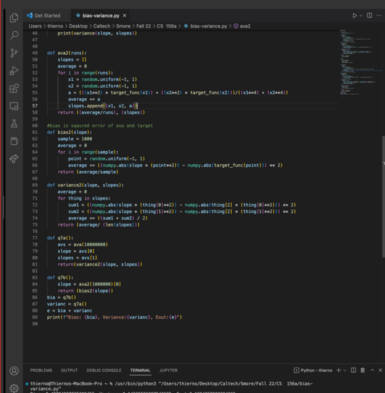
Find $\vec{g}$ using this as done in the code.

According to the below code the bias ≈ 0.50 & Variance $= 0.14$ with $E_{out} = 0.64$.

[e] $h(x) = ax^2 + b$ is similar to $h(x) = ax^2$ as $h(x) = ax + b$ is similar to $h(x) = ax$ hence $h(x) = ax^2 + b$ would have a higher E_{out} as seen in the relationship between choices b & c.

Thus e cannot have the least expected E_{out}.

- The correct answer is b bcz it has the lowest E_{out} as seen in the analysis above.



```
Users > thierno > Desktop > Caltech > Smore > Fall 22 > CS 156a > bias-variance.py > ava2
    print(variance(slope, slopes))


def ava2(runs):
    slopes = []
    average = 0
    for i in range(runs):
        x1 = random.uniform(-1, 1)
        x2 = random.uniform(-1, 1)
        a = (((x1**2) * target_func(x1)) + ((x2**2) * target_func(x2)))/((x1**4) + (x2**4))
        average += a
        slopes.append((x1, x2, a))
    return ((average/runs), (slopes))

#bias is sqaured error of ava and target
def bias2(slope):
    sample = 1000
    average = 0
    for i in range(sample):
        point = random.uniform(-1, 1)
        average += ((numpy.abs(slope * (point**2)) - numpy.abs(target_func(point))) ** 2)
    return (average/sample)

def variance2(slope, slopes):
    average = 0
    for thing in slopes:
        sum1 = ((numpy.abs(slope * (thing[0]**2)) - numpy.abs(thing[2] * (thing[0]**2))) ** 2)
        sum2 = ((numpy.abs(slope * (thing[1]**2)) - numpy.abs(thing[2] * (thing[1]**2))) ** 2)
        average += ((sum1 + sum2) / 2)
    return (average/ (len(slopes)))

def q7a():
    avs = ava(10000000)
    slope = avs[0]
    slopes = avs[1]
    return(variance2(slope, slopes))

def q7b():
    slope = ava2(1000000)[0]
    return (bias2(slope))
bia = q7b()
varianc = q7a()
e = bia + varianc
print(f"Bias: {bia}, Variance:{varianc}, Eout:{e}")
```

```
thierno@Thiernos-MacBook-Pro ~ % /usr/bin/python3 "/Users/thierno/Desktop/Caltech/Smore/Fall 22/CS 156a/bias-variance.py"
Bias: 0.49704997025399456, Variance:0.14235591997543195, Eout:0.6394058902294265
```

(8) All growth functions are in the form 2^N thus

$$m_{\mathcal{H}}(N+1) = 2m_{\mathcal{H}}(N) - \binom{N}{q}$$

$$2^{N+1} = 2(2^N) - \binom{N}{q}$$

$1 = 1 - \binom{N}{q}$ hence for this to be true $\binom{N}{q}$ must equal zero hence $q > N$ therefore the breakpoint of the growth function is $q+1$ as;

$$2^{q+1} \neq 2^{q+1} + \binom{q}{q}$$

$$\neq 2^{q+1} + 1$$

given q breakpoint of $q+1$ hence

d_{vc} = breakpoint − 1 hence giving a VC dimension of 9.

(9) We know that the smallest bound should be zero for when we have no intersecting sets hence eleminating choices d & e. Next the sum of the VC dimensions does not make sense for a tight bound because the intersection of sets should be less than the largest VC dimension hence summing them would result in a high VC dimension thus we can eleminate choice a as

$$d_v(H) \leq \sum_{k=1}^{K} d_v(H_k);$$

Now we know that for the case that we have entirely overlapping will be the upperbound with;

$\min\{d_{vc}(H_k)\}_{k=1}^{K}$ being a tighter bound than $\max\{d_{vc}(H_k)\}_{k=1}^{K}$ because the intersection of sets that we have should have the same VC dimension as the hypothesis set with the smallest VC dimension as the minimum d_{vc} of a hypothesis set can't have greater VC dimensions than those it intersects with hence making the tightest bound $\min\{d_{vc}(H_k)\}_{k=1}^{K}$. Hence the correct answer is b because it is the only one that matches the above description.

(5)

```
Users > thierno > Desktop > Caltech > Smore > Fall 22 > CS 156b > bias-variance.py > ...
from cProfile import run
from genericpath import samefile
import numpy
import random

def target_func(x):
    return numpy.sin(numpy.pi * x)

def ava(runs):
    average = 0
    for i in range(runs):
        x1 = random.uniform(-1, 1)
        x2 = random.uniform(-1, 1)
        a = ((x1 * target_func(x1)) + (x2 * target_func(x2)))/((x1**2) + (x2**2))
        average += a
    return (average/runs)
#bias is squared error of ava and target
def bias(slope):
    sample = 1000
    average = 0
    for i in range(sample):
        point = random.uniform(-1, 1)
        average += ((numpy.abs(slope * point) - numpy.abs(target_func(point))) ** 2)
    return (average/sample)

def q5():
    slope = ava(10000000)
    print (bias(slope))

def q4():

q5()
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER
thiernoThiernos-MacBook-Pro ~ % /usr/bin/python3 "/Users/thierno/Desktop/Caltech/Smore/Fall 22/CS 156b/bias-variance.py"
0.2661526289721333
thiernoThiernos-MacBook-Pro ~ % /usr/bin/python3 "/Users/thierno/Desktop/Caltech/Smore/Fall 22/CS 156b/bias-variance.py"
0.2710806928264555
thiernoThiernos-MacBook-Pro ~ %
```

hence the bias ≈ 0.27 according to the code on the left hence the correct answer is b as it is the closest.

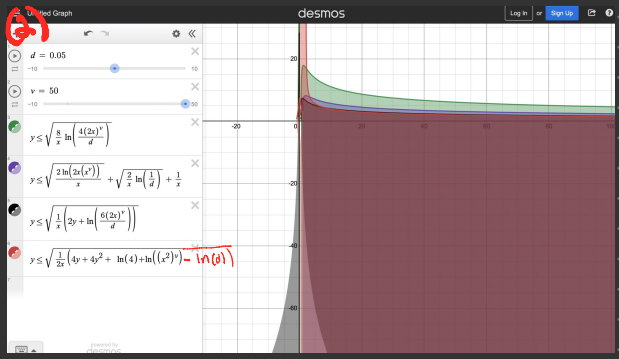
(6)

```
Users > thierno > Desktop > Caltech > Smore > Fall 22 > CS 156a > bias-variance.py > ...
import numpy
import random

def target_func(x):
    return numpy.sin(numpy.pi * x)

def ava(runs):
    slopes = []
    average = 0
    for i in range(runs):
        x1 = random.uniform(-1, 1)
        x2 = random.uniform(-1, 1)
        a = ((x1 * target_func(x1)) + (x2 * target_func(x2)))/((x1**2) + (x2**2))
        average += a
        slopes.append((x1, x2, a))
    return ((average/runs), (slopes))

#bias is squared error of ava and target
def bias(slope):

def variance(slope, slopes):
    average = 0
    for thing in slopes:
        sum1 = ((numpy.abs(slope * thing[0]) - numpy.abs(thing[2] * thing[0])) ** 2)
        sum2 = ((numpy.abs(slope * thing[1]) - numpy.abs(thing[2] * thing[1])) ** 2)
        average += ((sum1 + sum2) / 2)
    return (average/ (len(slopes)))

def q5():

def q4():

def q6():
    avs = ava(10000000)
    slope = avs[0]
    slopes = avs[1]
    print(variance(slope, slopes))
q6()
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER
thiernoThiernos-MacBook-Pro ~ % /usr/bin/python3 "/Users/thierno/Desktop/Caltech/Smore/Fall 22/CS 156a/bias-variance.py"
0.2826312425892293
thiernoThiernos-MacBook-Pro ~ % /usr/bin/python3 "/Users/thierno/Desktop/Caltech/Smore/Fall 22/CS 156a/bias-variance.py"
0.2825884785739614
```

Hence the variance ≈ 0.20 according to the code above hence the correct answer is a as it is exactly that value.

(7)



(10). We know that the lower bound of a union is the case where we have all intersecting sets hence we cannot have zero as a lower bound. Furthermore, as d_{VC} is positive for all of our hypothesis sets thus the union must be atleast 1 for the case where they all have $\min(d_{VC})$ and the same values.

Now looking at the upper bounds $\sum_{k=1}^{K} d_{VC}(H_k)$ is not a valid upper bound. This is because consider H_1 & H_2 for the following input space $[\frac{1}{2}, \frac{1}{2}]$. Then $d_{VC}=1$ for H_1 & $d_{VC}=1$ for H_2 as $[-1-]$ & $[+1+]$ both cannot be shattered. Now looking at $\sum_{k=1}^{2} d_{VC}(H_k)$ we would get their union as $1+1=2$ which is false as their union should be 3 as they can shatter 3 points for when both are on the classification line. Hence making $\sum_{k=1}^{K} d_{VC}(H_k)$ an incorrect Union upper bound hence making $k-1+\sum_{k=1}^{K} d_{VC}(H_k)$ the valid upper bound as $(2-1)+1+1=3$ V.

[a] wrong because incorrect lower and upper bound as discussed above.

[b] wrong because incorrect lower bound as discussed above.

[c] wrong because incorrect upper bound as discussed above.

[d] wrong because incorrect upper bound as discussed above.

[e] Correct because valid upper and lower bounds as discussed above.