1. use $m_H(N) \le N^{d_{vc}}$

$d_{vc} = 10$ want generalization error ≤ 0.05 w/95% confindence

find sample size that VC generalization bound predicts

VC inequality (from slide 23): $\varepsilon \le \sqrt{\frac{8}{N} \cdot \ln\left(\frac{4 m_H(2N)}{\delta}\right)}$

we want $0.95 \ge 1 - \delta \Rightarrow \delta = 0.05$

$\varepsilon \le 0.05$ $\quad m_H(2N) = (2N)^{d_{vc}} = (2N)^{10}$

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

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

$N = 452957 \Rightarrow$ [d]

2. $d_{vc} = 50 \quad \delta = 0.05 \quad N = 10{,}000$

$m_H(N) = N^{50}$

a: $$\varepsilon \leq \sqrt{\frac{8}{N} \ln\left(\frac{4 m_H(2N)}{\delta}\right)}$$

$$\varepsilon \leq \sqrt{\frac{8}{10000} \ln\left(\frac{4 \cdot (2 \cdot 1000)^{50}}{0.05}\right)}$$

$$\varepsilon \leq 0.632$$

b: $$\varepsilon \leq \sqrt{\frac{2 \ln(2N m_H(N))}{N}} + \sqrt{\frac{2}{N} \ln \frac{1}{\delta}} + \frac{1}{N}$$

$$\varepsilon \leq \sqrt{\frac{2 \cdot \ln(20000 \cdot 10000^{50})}{10000}} + \sqrt{\frac{2}{10000} \cdot \ln \frac{1}{0.05}} + \frac{1}{10000}$$

$$\varepsilon \leq 0.331$$

c: $$\varepsilon \leq \sqrt{\frac{1}{10000} \left(2\varepsilon + \ln \frac{6 \cdot (2 \cdot 10000)^{50}}{0.05}\right.}$$

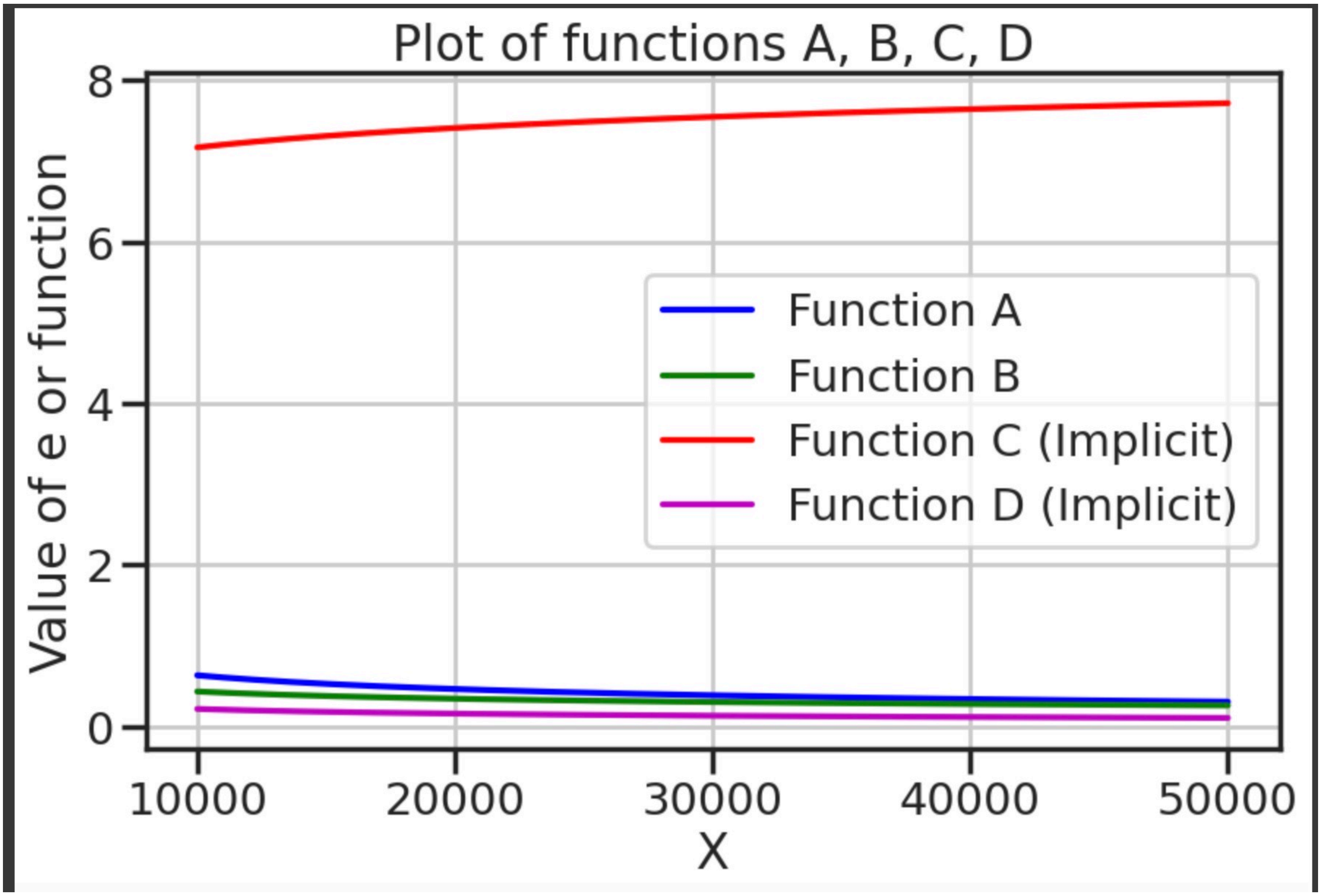
$$\varepsilon \leq \sqrt{0.05 + 0.0002\varepsilon}$$

$$\varepsilon \leq 0.224$$

$$d: \quad \varepsilon \leq \sqrt{\frac{1}{20000}\left(4\varepsilon + 4\varepsilon^2 + \ln \frac{4 \cdot (10000^2)^{50}}{0.05}\right)}$$

$$\varepsilon \leq \sqrt{0.0002\varepsilon + 0.0002\varepsilon^2 + 0.0463}$$

$$\varepsilon \leq 0.215$$



Based on my calculations and the graph, the Devroye bound is the smallest for very large N => $\boxed{[d]}$

3. $d_{vc} = 50 \quad \delta = 0.05 \quad N = 5 \quad m_H(N) = N^{50}$

a: $\varepsilon \leq \sqrt{\frac{8}{5} \cdot \ln\left(\frac{4 \cdot (10)^{50}}{0.05}\right)}$

$\varepsilon \leq 13.8$

b: $\varepsilon \leq \sqrt{\frac{2\ln(10 \cdot 5^{50})}{5}} + \sqrt{\frac{2}{5}\ln\frac{1}{0.05}} + \frac{1}{5}$

$\varepsilon \leq 7.05$

c: $\varepsilon \leq \sqrt{\frac{1}{5}\left(2\varepsilon + \ln\frac{6 \cdot (10)^{50}}{0.05}\right)}$

$\varepsilon \leq \sqrt{0.4\varepsilon + 23.98}$

$\varepsilon \leq 5.10$

d: $\varepsilon \leq \sqrt{\frac{1}{10}\left(4\varepsilon + 4\varepsilon^2 + \ln\left(\frac{4 \cdot 25^{50}}{0.05}\right)\right)}$

$\varepsilon \leq \sqrt{0.4\varepsilon + 0.4\varepsilon^2 + 16.5}$

$\varepsilon \leq 5.59$

Based on the calculations, the Parrondo and van den Broek bound is smallest $\Rightarrow$ [c]

4. *derivation of coefficient calculation in question 7

```
import math
import numpy as np

def target(x):
  return math.sin(math.pi*x)


def generate_training():
  X = np.random.uniform(-1.0, 1.0, 2)
  Y = np.array([target(point) for point in X])
  slope = ((X[0] * Y[0]) + (X[1] * Y[1])) / (X[0]**2 + X[1]**2)
  return slope

total = 0
N = 1000
for _ in range(N):
  total += generate_training()

print(total/N)
```

```
1.47169985148112
```

Based on the code above, the average slope over 1000 trials is 1.47, which does not exactly match any answer choices => [e]

5.

```
slope = total / N
bias = 0
for _ in range(N):
  x = np.random.uniform(-1.0, 1.0)
  y = target(x)
  bias += (slope * x - y)**2

print(bias/N)
```

```
0.27068804972349436
```

Based on the code above, the average bias across 1000 trials is 0.271, closest to 0.3 => [b]

6.

Problem 6:

```
variance = 0
for hypothesis in slopes:
    x = np.random.uniform(-1.0, 1.0)
    variance += (hypothesis*x - slope*x)**2

print(variance / N)
```

```
0.2371159516726178
```

Based on the code above, the average variance across 1000 trials is 0.237, closest to 0.2 => [a]

7. For each type of hypothesis, we want to calculate them in such a way that minimizes MSE.

[a] $h(x)=b$ $\quad MSE = \frac{1}{2}\left((b-y_1)^2+(b-y_2)^2\right)$

differentiate w.r.t. b: $\frac{1}{2}\left(2(b-y_1)+2(b-y_2)\right)=0$

$4b-2(y_1+y_2)=0$ $\quad \boxed{b=\frac{y_1+y_2}{2}}$

[b] $h(x)=ax$ $\quad MSE=\frac{1}{2}\left((ax_1-y_1)^2+(ax_2-y_2)^2\right)$

differentiate w.r.t. a: $\frac{1}{2}\left(2x_1(ax_1-y_1)+2x_2(ax_2-y_2)\right)$

$\cancel{2}x_1^2a-\cancel{2}x_1y_1+\cancel{2}x_2^2a-\cancel{2}x_2y_2=0$

$$\boxed{a=\frac{x_1y_1+x_2y_2}{x_1^2+x_2^2}}$$

[c] $h(x)=ax+b$ $\quad mse=\frac{1}{2}\left((ax_1+b-y_1)^2+(ax_2+b-y_2)^2\right)$

differentiate w.r.t. a: $\frac{1}{2}\left(2x_1(ax_1+b-y_1)+2x_2(ax_2+b-y_2)\right)$

$\cancel{2x_1^2a}+\cancel{2x_1b}-\cancel{2}x_1y_1+\cancel{2x_2^2a}+\cancel{2x_2b}-\cancel{2}x_2y_2=0$

$a(x_1^2+x_2^2)+b(x_1+x_2)=x_1y_1+x_2y_2$

differentiate w.r.t. b: $ax_1+b-y_1+ax_2+b-y_2=0$

$a(x_1+x_2)+2b=y_1+y_2$ $\quad b=\frac{y_1+y_2-a(x_1+x_2)}{2}$

$$a(x_1^2+x_2^2)+\frac{(x_1+x_2)(y_1+y_2)-a(x_1+x_2)^2}{2}=x_1y_1+x_2y_2$$

$$2a(x_1^2+x_2^2)+x_1y_1+x_1y_2+x_2y_1+x_2y_2-a(x_1+x_2)^2=2x_1y_1+2x_2y_2$$

$$ax_1^2+ax_2^2+x_2y_1+x_1y_2-2ax_1x_2=x_1y_1+x_2y_2$$

$$a(x_1-x_2)^2=x_1y_1-x_1y_2+x_2y_2-x_2y_1$$

$$a(x_1-x_2)^2=x_1(y_1-y_2)-x_2(y_1-y_2)$$

$$a=\frac{(x_1-x_2)(y_1-y_2)}{(x_1-x_2)^2}=\frac{y_1-y_2}{x_1-x_2}=a$$

$$b=\frac{1}{2}\left(y_1+y_2-\frac{(x_1+x_2)(y_1-y_2)}{x_1-x_2}\right)=x_1y_2-x_2y_1=b$$

$$(y_1+y_2)(x_1-x_2)=\cancel{x_1y_1}-x_2y_1+x_1y_2-\cancel{x_2y_2}$$

$$(x_1+x_2)(y_1-y_2)=\cancel{x_1y_1}-x_1y_2+x_2y_1-\cancel{x_2y_2}$$

[d] $h(x)=ax^2$ $\quad MSE=\frac{1}{2}\left((ax_1^2-y_1)^2+(ax_2^2-y_2)^2\right)$

differentiate w.r.t. a: $\frac{1}{2}\left(2x_1^2(ax_1^2-y_1)+2x_2^2(ax_2^2-y_2)\right)$

$$x_1^2(ax_1^2-y_1)+x_2^2(ax_2^2-y_2)=0$$

$$a(x_1^4+x_2^4)=x_1^2y_1+x_2^2y_2$$

$$a=\frac{x_1^2y_1+x_2^2y_2}{x_1^4+x_2^4}$$

[e] $h(x) = ax^2+b$ $\quad MSE = \frac{1}{2}\left((ax_1^2+b-y_1)^2+(ax_2^2+b-y_2)^2\right)$

differentiate w.r.t. a: $\frac{1}{2}\left(2x_1^2(ax_1^2+b-y_1)+2x_2^2(ax_2^2+b-y_2)\right)$

$$x_1^2(ax_1^2+b-y_1)+x_2^2(ax_2^2+b-y_2)=0$$

$$a(x_1^4+x_2^4)+b(x_1^2+x_2^2) = x_1^2y_1+x_2^2y_2$$

differentiate w.r.t. b: $ax_1^2+b-y_1+ax_2^2+b-y_2=0$

$$a(x_1^2+x_2^2)+2b=y_1+y_2$$

$$b=\frac{y_1+y_2-a(x_1^2+x_2^2)}{2}$$

$$2a(x_1^4+x_2^4)+(x_1^2+x_2^2)(y_1+y_2-a(x_1^2+x_2^2))=2x_1^2y_1+2x_2^2y_2$$

$$\cancel{2a(x_1^4+x_2^4)}+\cancel{x_1^2y_1}+x_1^2y_2+x_2^2y_1+\cancel{x_2^2y_2}-a(\cancel{x_1^4}+2x_1^2x_2^2+\cancel{x_2^4})$$
$$=2x_1^2y_1+2x_2^2y_2$$

$$a(x_1^4+x_2^4)+x_1^2y_2+x_2^2y_1-2ax_1^2x_2^2=x_1^2y_1+x_2^2y_2$$

$$a(x_1-x_2)^2 = \underbrace{x_1^2y_1-x_1^2y_2+x_2^2y_2-x_2^2y_1}_{x_1^2(y_1-y_2)-x_2^2(y_1-y_2)}$$

$$a=\frac{(x_1^2-x_2^2)(y_1-y_2)}{(x_1-x_2)^2}=\frac{(x_1+x_2)(y_1-y_2)}{(x_1-x_2)}=a$$

$$a=\frac{y_1}{x_1^2-x_2^2}-\frac{y_2}{x_1^2-x_2^2} \qquad b=\frac{x_1^2y_2-x_2^2y_1}{x_1^2-x_2^2}$$

Problem 7:

```python
#for each hypothesis, calculate the values of a and/or b for 1000 sets of 2 randomly generated points
#then for another randomly generated 1000 points, calculate average variance and bias

first_b = 0
option_a = []

second_a = 0
option_b = []

third_a = 0
third_b = 0
option_c = []

fourth_a = 0
option_d = []

fifth_a = 0
fifth_b = 0
option_e = []
```

```python
for _ in range(N):
  X = np.random.uniform(-1.0, 1.0, 2)
  Y = np.array([target(point) for point in X])
  slope1 = (Y[0] + Y[1])/2
  first_b += slope1
  option_a.append(slope1)
  slope2 = ((X[0] * Y[0]) + (X[1] * Y[1])) / (X[0]**2 + X[1]**2)
  second_a += slope2
  option_b.append(slope2)
  slope3a = (Y[0] - Y[1]) / (X[0] - X[1])
  third_a += slope3a
  slope3b = X[0]*Y[1] - X[1]*Y[0]
  third_b += slope3b
  option_c.append([slope3a, slope3b])
  slope4 = (Y[0]*X[0]**2 + Y[1]*X[1]**2) / (X[0]**4 + X[1]**4)
  fourth_a += slope4
  option_d.append(slope4)
  slope5a = (Y[0] - Y[1]) / (X[0]**2 - X[1]**2)
  fifth_a += slope5a
  slope5b = (Y[1]*X[0]**2 - Y[0]*X[1]**2) / (X[0]**2 - X[1]**2)
  fifth_b += slope
  option_e.append([slope5a, slope5b])

#now we have all the values for a  and b and we want to calculate error
first_b /= N
second_a /= N
third_a /= N
third_b /= N
fourth_a /= N
fifth_a /= N
fifth_b /= N
```

```python
a_error = 0
b_error = 0
c_error = 0
d_error = 0
e_error = 0

for i in range(N):
  x = np.random.uniform(-1.0, 1.0)
  y = target(x)
  a_error += (y - option_a[i])**2 + (first_b - option_a[i])**2
  b_error += (y - option_b[i]*x)**2 + (second_a*x - option_b[i]*x)**2
  c_error += (y - option_c[i][0]*x - option_c[i][1])**2 + (third_a*x + third_b - (option_c[i][0]*x + option_c[i][1]))**2
  d_error += (y - option_d[i]*x**2)**2 + (fourth_a*x**2 - option_d[i]*x**2)**2
  e_error += (y - option_e[i][0]*x**2 - option_e[i][1])**2 + (fifth_a*x**2 + fifth_b - (option_e[i][0]*x**2 + option_e[i][1]))**2

print("A error: ", a_error / N)
print("B error: ", b_error / N)
print("C error: ", c_error / N)
print("D error: ", d_error / N)
print("E error: ", e_error / N)
```

```
A error:  1.070606457960489
B error:  0.6968755932433993
C error:  2.0514767813221
D error:  21.202840836703643
E error:  14113.000301250535
```

Based on my calculations and code above, the hypothesis of the form $h(x) = ax$ has the least expected value of out-of-sample error (bias + variance) over 1000 trials. $\Rightarrow$ $\boxed{[b]}$

8. $m_H(N+1) = 2m_H(N) - \binom{N}{q}$

$\hookrightarrow$ VC-dimension of hypothesis set w/ this growth function $\Rightarrow$ largest value of N for which $m_H(N) = 2^N$

largest # of points that N can shatter

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

$$2m_H(N) = 2\underbrace{m_H(N-1)} - \binom{N-1}{q}$$

$$2m_H(N-2) - \binom{N-2}{q}$$

$$m_H(N+1) = 4m_H(N-1) - 2\binom{N-1}{q} - \binom{N}{q}$$

$$= 8m_H(N-2) - 4\binom{N-2}{q} - 2\binom{N-1}{q} - \binom{N}{q}$$

$$\ldots = 2^{N-q-1} m_H(q) - 2^{N-q-2}\binom{q}{q} - 2^{N-q-3}\binom{q+1}{q} + \ldots$$

$m_H(1) = 2 \quad m_H(2) = 2 \cdot 2 - \binom{2}{q}$

$m_H(3) = 8 - 2\binom{2}{q} - \binom{3}{q}$

$m_H(4) = \underbrace{16}_{2^N} - 4\binom{2}{q} - 2\binom{3}{q} - \binom{4}{q}$

if $N < q$, the $m_H(N) = 2^N$

when $N \geq q$, $m_H(N) = 2^N - \underbrace{\text{binomial terms}}$

Binomial terms are nonzero when $N \geq q$,

so q is the vc dimension because

it's the largest value of N for which

$m_H(N) = 2^N$. $\Rightarrow$ $\boxed{[c]}$

9. $H_1, H_2, H_3, \ldots H_K$ each have $d_{vc}(H_n)$

$d_{vc}(\bigcap_{k=1}^{K} H_k)$ = VC-dimension of intersection of all K hypothesis sets (hypotheses that are common to all hypothesis sets

d_{vc} = largest number of points that can be shattered by hypothesis set

[a] True - because the number of points shattered by the intersection of the hypothesis sets cannot be larger than the total number of points shattered by the individual hypothesis sets. The intersection of all the hypothesis sets will be smaller than or the same size as all the hypotheses.

[b] True - because the intersection of all the hypothesis sets will be smaller than or the same size as the hypothesis set that has the smallest VC dimension. If hypothesis set A is smaller than hypothesis set B, the VC dimension of A cannot be larger than that of B because there are less

ways (i.e. 'options' of hypotheses) to shatter the same number of points.

[c] True - Using similar reasoning as above, the intersection of all the hypothesis sets will only be as big if not smaller than the hypothesis set that results in the largest vc dimension. This means that the vc dimension of the intersection cannot be larger than the maximum vc dimension of all the hypothesis sets.

[d] and [e] are <u>false</u> because the lower bound must be 0. In the case where there are no common hypotheses between all the sets, the intersection will be 0 so the vc-dimension will be 0. Moreover, we know that all the vc dimensions of the hypothesis sets are positive, so the inequality would not hold true.

Between a, b, and c, b is the tightest bound because all the vc dimensions of the hypothesis sets are positive, so

$$\min \{d_{vc}(H_k)\}_{k=1}^{K} \leq \max \{d_{vc}(H_k)\}_{k=1}^{K} < \sum_{k=1}^{K} d_{vc}(H_k)$$

Final answer: [b]

10. bound for <u>union</u> of sets

Since the union of sets contains all the hypotheses in all of the hypothesis sets, we know that the lower bound of $d_{vc}(\bigcup_{k=1}^{K} H_k)$ will be at least the maximum d_{vc} of the original set of hypotheses. The union will contain at least those hypotheses in the set that achieved the maximum d_{vc}, so $\max\{d_{vc}(H_k)\}_{k=1}^{K}$ is the tighest lower bound and options a, b, and c are not correct. We have options [d] and [e] left, so we want to determine whether $\sum_{k=1}^{K} d_{vc}(H_k)$ is an appropriate upper bound, since it is tighter. $\sum_{k=1}^{K} d_{vc}(H_k)$ is only the sum of the individual d_{vc} for each of the hypothesis sets, but it does not account for the interaction between hypothesis sets in terms of flexibility: d_{vc} correlates to how 'flexible' a hypothesis set is, so the upper bound being $\sum_{k=1}^{K} d_{vc}(H_k)$ assumes that there is no interaction between hypothesis sets. So,

$K-1+\sum_{k=1}^{K} d_{vc}(H_k)$ is a more appropriate and accurate upper bound for $d_{vc}(\bigcup_{k=1}^{K} H_k)$. $\Rightarrow$ [e]