# Problem 1

Deterministic noise is the part of the target function that cannot be modeled due to a lack of hypothesis set complexity.

Deterministic noise arises from a models failure to properly fit a target function. So in general deterministic noise goes down as model complexity increases. If , in general, it is more difficult for to fit a target function properly which means that the deterministic noise will trend upwards so the answer is B

# Problem 2

The simulation resulted in and so the answer is A

# Problem 3

The simulation resulted in and so the answer is D

# Problem 4

The simulation resulting in and so the answer is E

# Problem 5

The simulation says that gives so the answer is D

# Problem 6

The simulation gives so the answer is B

# Problem 7

A

We can see that is a proper subset of of

This is because requires that all for whereas requires that all for so includes if we consider that in we **can** (although we aren’t restricted to it) have which accounts for all . This means that

We know that for so we have free for so we have and not .

B

By the same reasoning as above we know that

We can also see that since we are restricted to for which is not allowed in .

C

We know from (A) that . Thus, thus we know that for and is free for which is the same set of hypotheses as so the statement is correct and the answer is C

D

We know that if we force for as required in , we know we don’t have .

The answer is C

# Problem 8

In forward propagation we must get for and with and . So, in the first layer, we take 3 such inner products, each with 6 multiplications of some by some . So, the first layer requires 18 operations. The second layer we take 1 such inner product of depth 4 so that layer contributes 4 operations. The final layer does not require a product as we are just applying a non-linearity to an already calculated signal. So, forward propagation requires 22 operations

We don’t count computing the final delta as an operation and we also don’t calculate deltas for the constant inputs in the first layer. We also don’t compute deltas for ’s. We only compute deltas three times, once for each of the nodes in the second layer.

We also need to update in a symmetrical way compared to how we did forward propagation which requires another 22 operations bringing the total to 47 so the answer is D

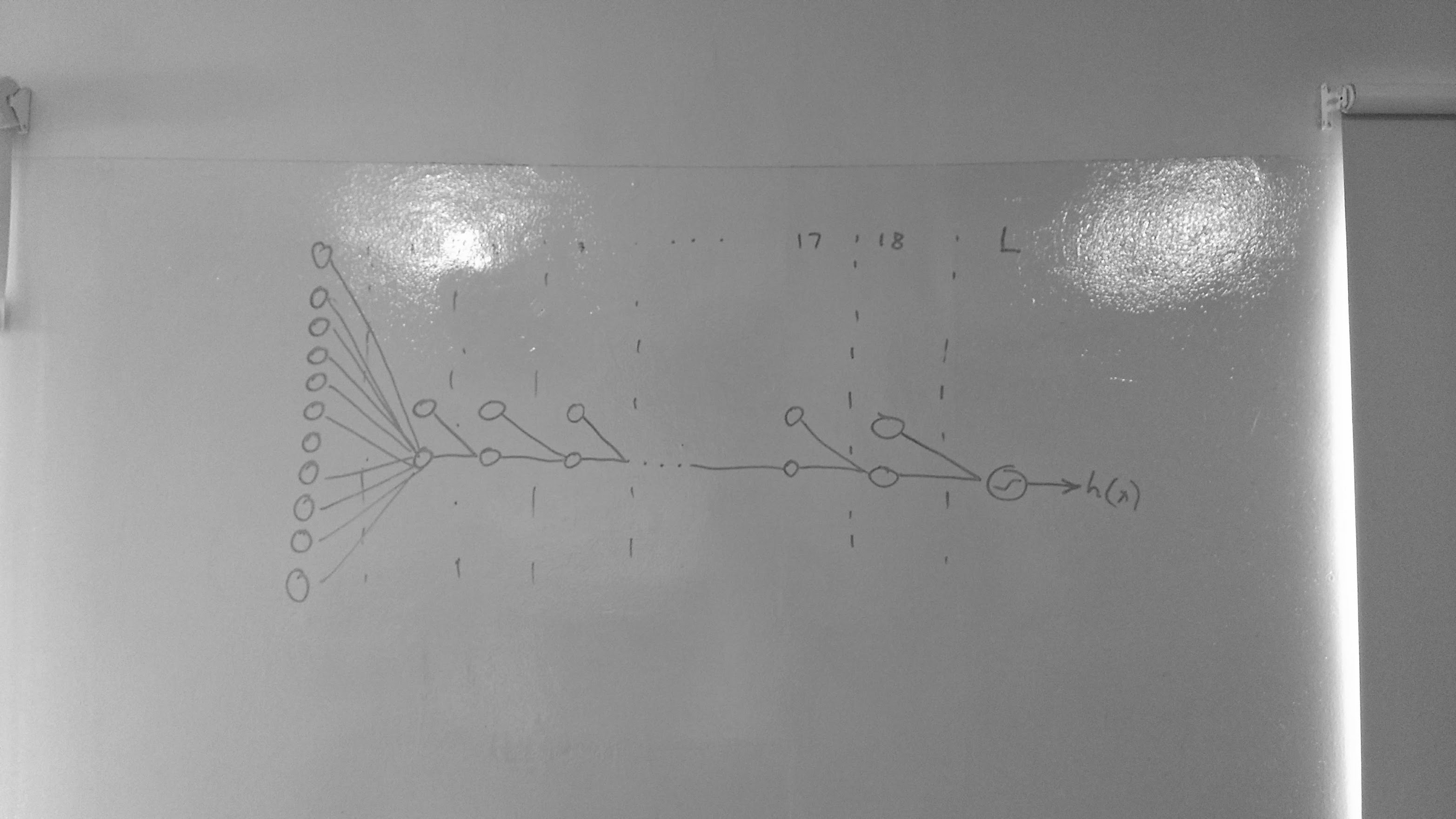
# Problem 9

We have a budget of exactly 36 units.

We know that the number of weights is

So it makes sense that the easiest way to reduce the number of weights is to keep the dimensionality of layers as low as possible.

We are given that , thus is we construct:



Which has 18 hidden layers consisting of a constant and one element . Each layer has 2 weights that influence the in the following layer

So we have: weights

So the answer is A

# Problem 10

We look to do the opposite arrangement to maximize:

Consider a network with that has 10 input units connected to second layer units which are in turn connected to third layer units. We know that .

We also know that the number of weights is:

This is a downward facing parabola so the max is easily found at the vertex.

so we have where and

Which is the largest option and the answer is E