Homework 5

Problem 1

c is the correct answer.
Plugging in 0.008 for expected E_{in}, 8 for d, and 0.1 for σ, we get that N = 35. Thus, the smallest N that will result in an expected E_{in} greater than 0.008 is answer choice c.

Problem 2

d is the correct answer.
The output is the sign(w_0 + w_1 x_1^2 + w_2 x_2^2). When $w_1 < 0$ and $w_2 > 0$, we get positive for large values of x_2 and negative for large values of x_1 which is precisely what the hyperbolic decision boundary is displaying, assuming w_0 can be adjusted accordingly.

Problem 3

c is the correct answer.
We have that $d_{vc} \leq d + 1$ where d+1 is the number of parameters. The number of parameters in this case is 14 so $d_{vc} \leq 15$.

Problem 4

e is the correct answer.
Using simple multi variable calculus, we an take the partial derivative using chain rule to get answer choice e.

Problem 5

d is the correct answer.
See attached code. I kept track of the number of iterations before the error got less than 10^{-14} by updating the weight vector by the gradient of E(u,v) multiplied by the learning rate of 0.1. The answer came out to be 10.

Problem 6

e is the correct answer.
See attached code. The final value for my weights were (0.0447, 0.0239), closest to answer choice e.

Problem 7

a is the correct answer.
See attached code. I did as the prompt told, updating the weight vector by coordinate, one after the other. I did this 15 times (30 steps) and got 0.139 as my error after, closest to answer choice a.

Problem 8

d is the correct answer.
See attached code. Using the formula for "cross-entropy" error on slide 16 of lecture 9, I was able to get an average E_{out} of 0.097, closest to 0.1, or answer choice d.

Problem 9

a is the correct answer.
See attached code. To calculate the number of epochs, I incremented the epoch number by 1 after the weight vector was changed stochastically (point to point) for all N points. To determine if I would need another epoch, I test to see the difference between the original and modified weight vectors. If the difference ≥ 0.01, then there will be another epoch. If the difference is < 0.01, we are done. The average epoch over the 100 runs turned out to be 332.6, closest to answer choice a.

Problem 10

e is the correct answer.
Recall that for a PLA, to update the weight vector, we add $y\mathbf{x}$ if $\mathbf{x}$ is a misclassified point. For a SGD, we add -n∇E_{in}. To simulate a PLA, we want the gradient of $e_n(\mathbf{w})$ to be 0 when the point is classified correctly, and -y$\mathbf{w^T x}$ when classified incorrectly. Choice e provides this.