Learning from Data
Homework # 5

Khoi Pham

Linear Regression Error

1. We want the expected E_{in} greater than 0.008, so

$$\mathbb{E}_{\mathcal{D}}[E_{\text{in}}(\mathbf{w}_{\text{lin}})] = \sigma^2\left(1 - \frac{d+1}{N}\right) \geq 0.008$$

Substituting $\sigma = 0.1, d = 8$ gives

$$\begin{aligned} 0.1^2\left(1 - \frac{9}{N}\right) &\geq 0.008 \\ 1 - \frac{9}{N} &\geq 0.8 \\ 0.2 &\geq \frac{9}{N} \\ N &\geq 45 \end{aligned}$$

Answer: [c]

Nonlinear Transforms

2. We can consider point $(0, 0)$ at the center of the figure which is classified as $+1$. Thus, $\tilde{w}_0$ must be greater than 0. We notice that either increasing or decreasing x_1 will always increase x_1^2, so $\tilde{w}_1$ must be negative in order to make it go from the $+1$ region to the -1 region. Increasing and decreasing x_2 is equivalent to moving up and down the figure, making it more likely to go into the $+1$ region. Thus, w_2 must be positive to achieve this.
Answer: [d]

3. The dimensionality of the input space after performing the 4th order polynomial transformation becomes 15. Thus the VC dimension must be less than or equal to 15 because there might be points in the new space that are not valid transforms of points in the old space.
Answer: [c]

Gradient Descent

The code for answering questions in this section is uploaded here.

4. Differentiate $E(u, v)$ with respect to u and v gives

$$\frac{\partial E}{\partial u} = 2(ue^v - 2ve^{-u})(e^v + 2ve^{-u})$$

$$\frac{\partial E}{\partial v} = 2(ue^v - 2ve^{-u})(ue^v - 2e^{-u})$$

Answer: [e]

5. As shown by the code, it takes 10 iterations for the error to fall below 10^{-14}.
Answer: [d]

6. $u \approx 0.045, v \approx 0.024$.
Answer: [e]

7. As shown by the code, error $E(u, v)$ reaches 0.14 after 15 full iterations.
Answer: [a]

Logistic Regression

Logistic Regression code for answering question 8 and 9 is uploaded here.

8. Answer: [d]

9. Answer: [a]

PLA as SGD

10. The problem asks to use SGD along with an error function so that we can implement the perceptron learning algorithm. In each iteration of the PLA, we shall choose a misclassfied data point and try to update the weight vector according to that data point. We can imagine this selection mechanism as randomly picking a data point, then check whether that data point is misclassified in order to decide to update the weight vector or not. In other words, if the randomly picked data point is already correctly classified then we do nothing, otherwise we will update the weight vector. Thus, we ought to choose an error function that allows us do the same. The only reasonable choice would be $-\min(0, y_n\mathbf{w}^{\mathrm{T}}\mathbf{x}_n)$. This is because if the randomly picked data point in SGD is correctly classified, then y_n will have the same sign as $\mathbf{w}^{\mathrm{T}}\mathbf{x}_n$ and their product

will be positive, making the error equal to 0. On the other hand, if the data point is misclassified, then we can take the derivative of $y_n \mathbf{w}^{\mathrm{T}} \mathbf{x}_n$ to apply gradient descent.
Answer: [e]