Cody Hubbard 004843389

CSM146, Winter 2018

Problem Set 2

Problem 1

(a) Solution: For AND:

$\theta(x_1,x_2) = \left(\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - 1 \right)$ this can be tested by plugging in values for x_1 and x_2.

$\theta(1,1)$	1	$+$
$\theta(1,-1)$	-1	-
$\theta(-1,1)$	-1	-
$\theta(-1,-1)$	-3	-

Another valid perception is $\theta_2(x_1,x_2) = \left(\begin{bmatrix} 1 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - 1 \right)$ which can be shown to satisfy AND as well

$\theta(1,1)$	$=\frac{1}{2}$	$+$
$\theta(1,-1)$	$=-\frac{1}{2}$	-
$\theta(-1,1)$	$=-\frac{3}{2}$	-
$\theta(-1,-1)$	$=-\frac{5}{2}$	-

(b) Solution: No two input percption exists to compute the XOR function. This is becasue of the way the sets of points that map to -1 and 1 intersect make it impossible to linearly seperate them.

Problem 2

Solution: given $J(\theta) = -\sum_{n=1}^{N}[y_n log(h_\theta(x_n)) + (1-y_n) log(1-h_\theta(x_n))]$ and $h_\theta = \sigma(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$

Looking at that fucntion you can see that $\frac{\partial J}{\partial \theta}$ comes down to $log(h_\theta(x_n))'$and $log(1 - h_\theta(x_n))'$

$$log(h_\theta(x_n))' = \frac{1}{h_\theta(x_n)} \cdot h_\theta(x_n)'$$

where

$$\begin{aligned} h(x_n)' &= \left(\frac{1}{1+e^{-\theta^T x}} \right)' \\ &= \frac{xe^{-\theta^T x}}{(1+e^{-\theta^T x})^2} \end{aligned}$$

thus

$$
\begin{aligned}
log(h_\theta(x_n))' &= \frac{1}{h_\theta(x_n)} \cdot h_\theta(x_n)' \\
&= \left(1 + e^{-\theta^T x}\right) \cdot \frac{xe^{-\theta^T x}}{(1+e^{-\theta^T x})^2} \\
&= \frac{xe^{-\theta^T x}}{1+e^{-\theta^T x}} \\
&= xe^{-\theta^T x} \cdot h_\theta(x_n)
\end{aligned}
$$

and

$$
log(1 - h_\theta(x_n))' = xe^{-\theta^T x}
$$

and finally

$$
\frac{\partial J}{\partial \theta} = -\sum_{n=1}^{N} [y_n(xe^{-\theta^T x} \cdot h_\theta(x_n)) + (1 - y_n)(xe^{-\theta^T x}))]
$$

Problem 3

(a) Solution: Given $J(\theta_0, \theta_1) = \sum_{n=1}^{N} w_n(\theta_0 + \theta_1 x_{n,1} - y_n)^2$

$$
\begin{aligned}
\frac{\partial J}{\partial \theta_0} &= \sum_{n=1}^{N} 2 \cdot w_n(\theta_0 + \theta_1 x_{n,1} - y_n) \\
\frac{\partial J}{\partial \theta_1} &= \sum_{n=1}^{N} 2 \cdot x_n \cdot w_n(\theta_0 + \theta_1 x_{n,1} - y_n)
\end{aligned}
$$

(b) Solution: Take the sum so that the full verctors for X, Y, W, and θ are formed, then

$$
\begin{aligned}
J(\theta_0, \theta_1) &= (X\theta - Y)^T W(X\theta - Y) \\
&= (X\theta - Y)^T W(X\theta - Y) \\
&= (\theta^T X^T - Y^T)W(X\theta - Y) \\
&= (\theta^T X^T - Y^T)(WX\theta - WY) \\
&= (\theta^T X^T WX\theta - Y^T WX\theta - \theta^T X^T WY - Y^T WY)
\end{aligned}
$$

and now take the derivative and set equal to zero

$$
\begin{aligned}
\frac{\partial J}{\partial \theta} &= (X^T WX\theta - Y^T WX - X^T WY) \\
0 &= X^T WX\theta - Y^T WX - X^T WY \\
(X^T WX)\theta &= Y^T WX + X^T WY \\
\theta &= (X^T WX)^{-1} Y^T WX + (X^T WX)^{-1} X^T WY \\
\theta &= (X^T WX)^{-1} Y^T WX + X^{-1} Y
\end{aligned}
$$

Problem 4

(a) Solution: We are given that the data set $D = \{(x_i, y_i)\}_{i=1}^{m}$ satisfies $y_i = \begin{cases} 1 & \text{if } w^T x_i + \theta \geq 0 \\ -1 & \text{if } w^T x_i + \theta < 0 \end{cases}$

Consider the linear program

$$\begin{gathered} \min \delta \\ \text{subject to} y_i(w^T x_i + \theta) \geq 1 - \delta \\ \delta \geq 0 \end{gathered}$$

(b) Solution: Given there is an optimal solution with $\delta = 0$ then $y_i(w^T x_i + \theta) \geq 1$ is satisfied for all D and

$$\begin{gathered} \text{subject to} y_i(w^T x_i + \theta) \geq 1 \geq 0 \text{ when } y = 1 \\ \text{subject to} y_i(w^T x_i + \theta \leq -1 < 0 \text{ when } y = -1 \end{gathered}$$

which shows it to be linearly seperable.

(c) Solution: If there is a hyperplane that satifies $y_i(w^T x_i + \theta) \geq 1 - \delta$for $\forall D$ with $\delta > 0$ then the value of delta determines our linear seperability. if $\delta < 1$ then we know its seperable by (b), if $\delta > 1$ if we cannot tell if the data set is seperable, and if $\delta = 1$ we know if is not seperable.

(d) Solution: The trivial optimal solution is $w = 0$, $\theta = 0$, and $\delta = 0$. This is a solution because $y_i(w^T x_i + \theta) \geq -\delta$ is satisfied. The issue with this formulation is that the solution doesnt form a hyperplane.

(e) Solution: The possible optimal solutions are since there are only two points in our D we know the dataset is linearly seperable. Plugging into the linear program it becomes appernt we need to satisfy

$$\begin{gathered} 1 \cdot (w_1 + w_2 + w_3 + ... + w_n + \theta) \geq 1 \\ -1 \cdot (-w_1 - w_2 - w_3 - ... - w_n + \theta) \geq 1 \end{gathered}$$

which means we need $w_1 + w_2 + w_3 + ... + w_n \geq \left|1 + \theta\right|$
so the set of optimal solutions are all δ, w, θ such that $\delta = 0$, and $w_1 + w_2 + w_3 + ... + w_n \geq \left|1 + \theta\right|$.

Problem 5

(a) Solution: For the training data I see that the data has a negative sloped linear shape. I feel a negative sloped line would make a decent linear regression for predicting this data.

For the testing data I see two main groups around $(0.2, 1.5)$ and $(0.8, 0.4)$ I do not feel like a linear regression could do a good job predicting this data.

(b) Solution: I modified the code in regression.py as instructed

(c) Solution: Completed PolynomialRegression.predict as required

(d) Solution:

rate	coefficients	iterations	final value
10^{-4}	$[1.91573, -1.743589]$	10000	5.49356
10^{-3}	$[2.44638, -2.81630]$	10000	3.91257
10^{-2}	$[2.44640, -2.81635]$	1505	3.91257
0.0407	$[2.44640, -2.81635]$	381	3.91257

The coefficents all seem to converge to $[2.44, -2.81]$ which is good, means that there is consistancy so they are probably converging correctly. It seems that the larger the η the more efficent the convervegnce. $\eta = .0407$ converges much faster than $\eta = 10^{-2}$ the finals values are approximatly the same

(e) Solution: The closed form solution's coefficents are $[2.4464, -2.8163]$ which are approximately the same as those found by our gadient decent algorithm. The cost of the closed form solution was 3.9125 as well. Overall these numbers match the outputs of our gradient decent algorithm. The closed form solution algorithm runs extremely quickly compared to gradient descent as we do not need to loop 10000 times.

(f) Solution: With the learning rate of $\dfrac{1}{1+k}$ the algorithm takes 10000 interations and therefore doenst converge. The final coefficents are still close $[2.44634, -2.81623]$ and the final value is 3.91257 but with the minimal margins of differnce ive been seeing with my numbers this was far worse than the others.

(g) Solution: I updated PolynomialRegression.generate_polynomial_features(...) to create an m + 1 dimensional feature vector for each instance.

(h) Solution: RMSE is a better esitmator of error over $J(\theta)$ because it gives a much more linear error correlation between our data points and our model.

I implemented PolynomialRegression.rms_error(...).

(i) Solution: In the graph below you can see that the best degree polynomial is 5. This is becasue it has some of the lowest error on the training data and the test data. You can see evidence of overfitting on polynomials of degree 9 and 10. These two polynials have very low training error because the complex polynomial is fitting their data points more perfectly, however the error for the test data blows up becasue of the same complex polynomial is very bad at fitting new points.