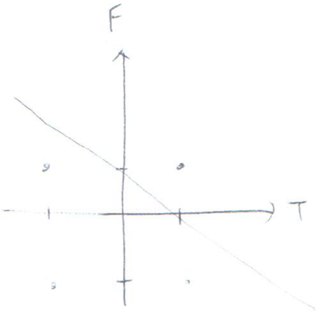
1. (a)



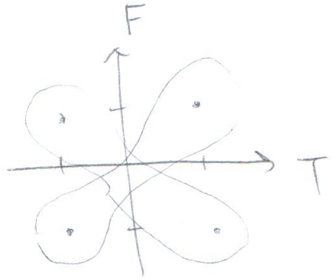
let line : $W_1 X_1 + W_2 X_2 + b = 0$

$$W_1 + W_2 + b > 0$$
$$W_1 - W_2 + b < 0$$
$$-W_1 + W_2 + b < 0$$
$$-W_1 - W_2 + b < 0$$

If we set $W_1 = 1$, $W_2 = 1$, $b = -1$ the equation can be fit. Thus the perceptron exists.

We also can find anoter line. Satisfys: $W_1 = 1$, $W_2 = 1$, $b = -0.5$

(b)



$$\begin{array}{ll} x & y \\ (1,1) \rightarrow & 0 \\ (1,0) \rightarrow & 1 \\ (0,1) \rightarrow & 1 \\ (0,0) \rightarrow & 0 \end{array} \Rightarrow \begin{array}{l} W_1 + W_2 + b > 0 \quad - \textcircled{1} \\ W_1 - W_2 + b < 0 \quad - \textcircled{2} \\ -W_1 + W_2 + b < 0 \quad - \textcircled{3} \\ -W_1 - W_2 + b > 0 \quad - \textcircled{4} \end{array}$$

$\boxed{2+3} \Rightarrow 2b < 0 \Rightarrow b < 0$

We find that ① & ④ conflict since either of $(W_1 + W_2)$ / $-(W_1 + W_2)$ should < 0, so it's impossible that both ① & ④ can be satisfied by the same W_1, W_2, b, which mean there's no solution of W_1, W_2, b for the equation. $\Rightarrow$ no perceptron exists.

2-(a)

$$\frac{\partial J(\theta)}{\partial \theta_j} = -\sum_{n=1}^{N} y_n(1-\sigma(\theta^T x))\, x_{nj} + (1-y_n)(-\sigma(\theta^T x))\, x_{nj}$$

$$= \sum_{n=1}^{N} (\sigma(\theta^T x_n) - y_n)\, x_{nj}$$

(b)

$$\frac{\partial^2 J(\theta)}{\partial \theta_j \partial \theta_k} = \frac{\partial}{\partial \theta_k}\left(\frac{\partial J(\theta)}{\partial \theta_j}\right) = \sum_{n=1}^{N} \sigma(\theta^T x)(1-\sigma(\theta^T x))\, x_{nj} \cdot x_{nk}$$

$$= \sum_{n=1}^{N} h_\theta(x_n)(1-h_\theta(x_n))\, x_{nj} \cdot x_{nk}$$

$$= \sum_{n=1}^{N} h_\theta(x_n)(1-h_\theta(x_n))\, x_n x_n^T$$

(c) Since we already know $x_n x^T$ is PSD

(Prove: $z^T x x^T z$
$= (z^T x)(x^T z)$
$= (x^T z)^T (x^T z) \geq 0$)

And two probabilities $h_\theta(x_n)$ & $(1-h_\theta(x_n))$ always ≥ 0

$$\Rightarrow z^T H z = \sum_{j,k} z_j z_k \underbrace{\sum_{n=1}^{N} h_\theta(x_n)(1-h_\theta(x_n))\, x_n x_n^T}_{\geq 0}$$

We also know that if $x_n x_n^T$ is PSD, then $c \cdot x_n x_n^T$ $(c \geq 0)$ is also PSD,

(Prove: $z^T c(x_n x_n^T) z$
$= c(z^T(x_n x_n^T) z)$
≥ 0)

so we know since $x_n x_n^T$ is PSD, H is also PSD.

3. (a)

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

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

(b)

$$\frac{\partial J}{\partial \theta_0} = 0 \quad \Rightarrow \quad \sum_{n=1}^{N} w_n(\theta_0 + \theta_1 x_{n,1} - y_n) = 0$$

$$\theta_0 \sum_{n=1}^{N} w_n + \theta_1 \sum_{n=1}^{N} w_n x_{n,1} = \sum_{n=1}^{N} w_n y_n$$

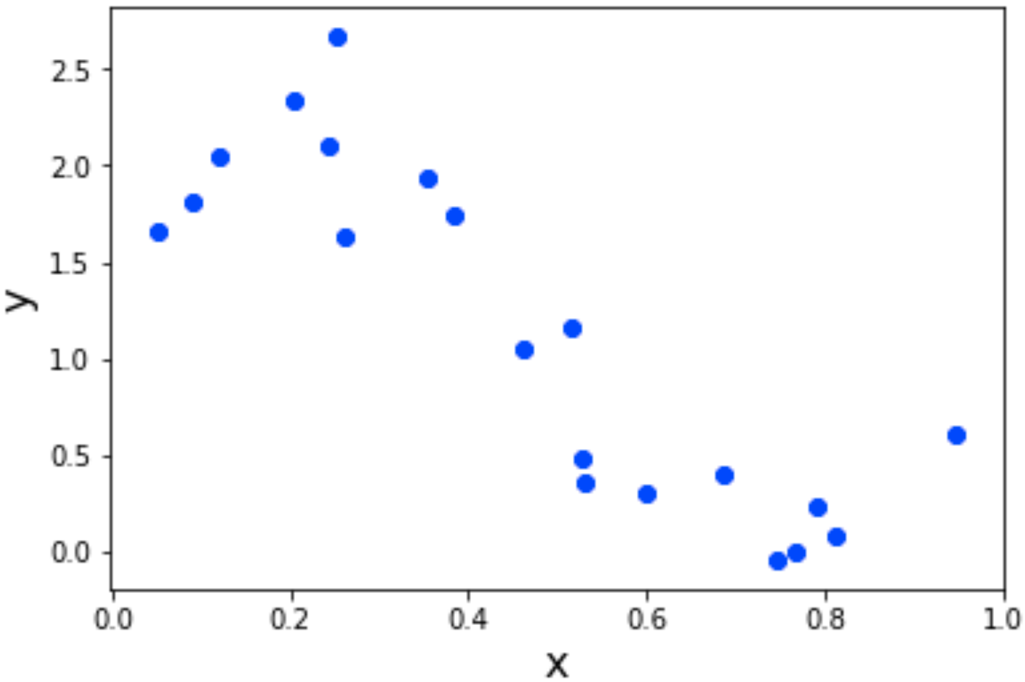
$$\theta_0 = \frac{\sum_{n=1}^{N} w_n y_n - \sum_{n=1}^{N} \theta_1 w_n x_{n,1}]}{\sum_{n=1}^{N} w_n}$$

$$\frac{\partial J}{\partial \theta_1} = 0 \quad \Rightarrow \quad \theta_0 \sum_{n=1}^{N} w_n x_{n,1} + \theta_1 \sum_{n=1}^{N} x_{n,1}^2 w_n = \sum_{n=1}^{N} w_n y_n x_{n,1}$$

$$\frac{\sum_{n=1}^{N} w_n y_n - \sum_{n=1}^{N} \theta_1 w_n x_{n,1}}{\sum_{n=1}^{N} w_n} \sum_{n=1}^{N} w_n x_{n,1} + \theta_1 \sum_{n=1}^{N} x_{n,1}^2 w_n = \sum_{n=1}^{N} w_n y_n x_{n,1}$$

$$\frac{\sum_{n=1}^{N} w_n y_n x_{n,1} \sum_{n=1}^{N} w_n - \sum_{n=1}^{N} w_n y_n \sum_{n=1}^{N} w_n x_{n,1}}{\sum_{n=1}^{N} w_n \sum_{n=1}^{N} x_{n,1}^2 w_n - \left(\sum_{n=1}^{N} w_n x_{n,1}\right)^2} = \theta_1$$

4.(a)



We can make a prediction that those data fit with a line that y decrease when x increase. However, there is some noise exist among those data. Based on these observation, I can guess that the linear regression can make some effective prediction, but probably not very accuracy because of the noise.

(d)

Step size	θ_0	θ_1	# iteration	cost	time
0.0001	1.91573585	-1.74358989	10000	5.49356558874	0.433943
0.001	2.4463815	-2.81630184	10000	3.91257640947	0.446339
0.01	2.44640696	-2.81635331	1466	3.91257640579	0.084192
0.0407	2.44640706	-2.81635353	333	3.91257640579	0.030298

From this table we can see when step size =0.0001, the coefficients are different from the other, and also the total number of iteration were used up, which indicates that probably this model hasn't converged to the global minimal point. The other three model with different step size all have similar results of coefficients and cost. When we comparing between them, we can see that the model with the largest step size = 0.047 used the fewest number of iteration and spent the least time to reach the convergence, which should be the most efficient one.

(e)

The coefficients estimated from closed-form are [2.44640709 -2.81635359].
The cost is 3.91257640579.
Time is 0.001001

We can find that the coefficients and the cost are almost the same as the answer got from gradient descent. When comparing the time, using closed-form calculation seems more efficient. In theory, the complexity of matrix inversion calculation is O(D^3) and matrix multiplication is O(ND^2), while the complexity of batch gradient descent is O(nD). But in our case, N is only 20 and D is only 2, while n could equal up to 10000. Because of the small amount of data and small number of features, using closed-form is more efficiency in our case.

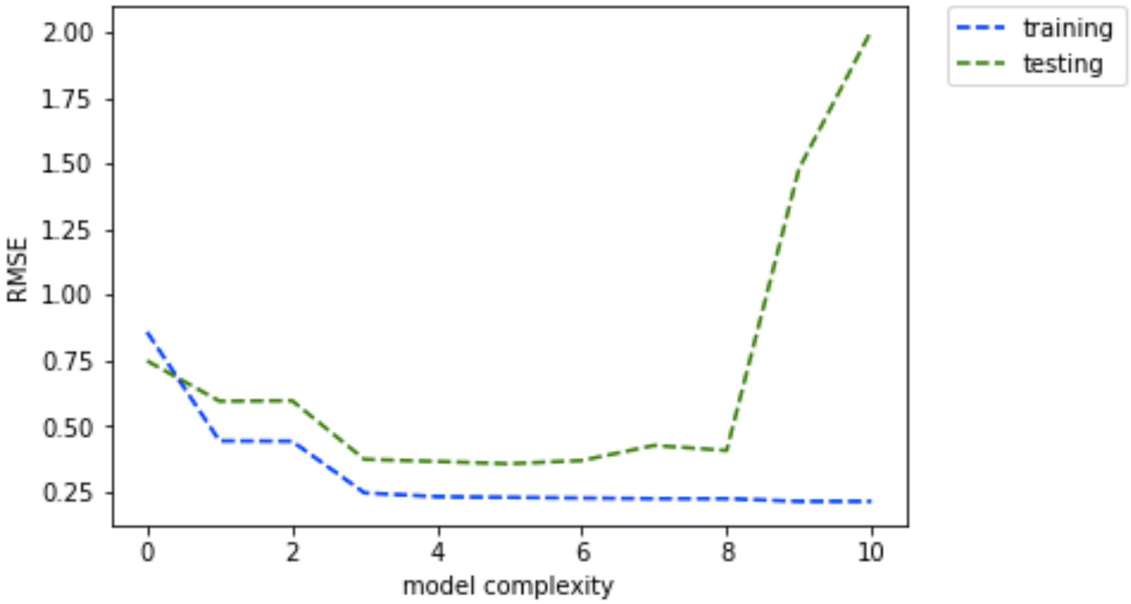
(f)

Time = 0.436044, iteration = 10000.

(h)

RMSE is a normalized form is J(θ), which can let us compare between model with different number of data set. Because of this standard form, we can also make more appropriate judgement of when does the model converge by comparing with previous model training experience.

(i)



The model has the smallest test RMSE when m = 5. We can see when m<3, the training RMSE are too high that probably caused by underfitting. When m>6, though the RMSE of training is still decreasing, the RMSE of testing is increasing, which should caused by overfitting.