

# SUTD 2021 50.007 Homework 3

James Raphael Tiovalen 1004555

## 1. Logistic Regression

### Question 1.1

- a) True.
- b) False. Despite the term “regression” being present in the name of “logistic regression”, it is actually a classification algorithm. The term “regression” is present simply because the output is a real number. However, the output is still bounded, and hence, it is solving a classification problem.
- c) True. Logistic regression is a subset and a building block of neural network. Thus, a single-layer feedforward neural network (i.e., a single neuron with only one output) using the sigmoid activation function is equivalent to logistic regression.
- d) False. The output probability value or confidence score is between 0 and 1, inclusive.

### Question 1.2

The correct statements are:

- 2) Our estimate for  $P(y = 0|x; \theta)$  is 0.72.
- 3) Our estimate for  $P(y = 1|x; \theta)$  is 0.28.

This is because  $h_\theta(x)$  is the estimated probability that  $y = 1$  on input  $x$ . Hence, by this definition, it implies that  $P(y = 1|x; \theta) = 0.28$ .

Additionally, since  $P(y = 0|x; \theta) + P(y = 1|x; \theta) = 1$ , we can substitute the value that we have previously to obtain  $P(y = 0|x; \theta) = 1 - 0.28 = 0.72$ .

### Question 1.3

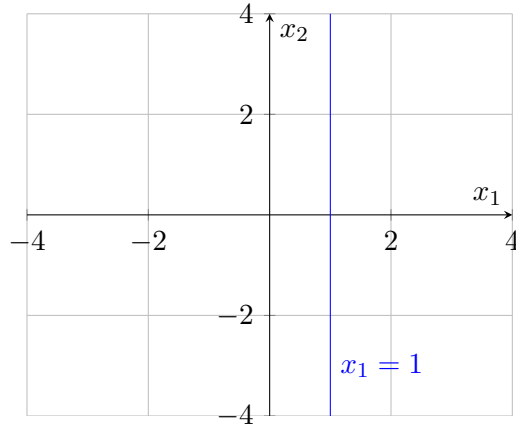
Given that  $\theta = [3 \quad -3 \quad 0]^T$ , we have:

$$y = \begin{cases} 1, & \text{if } 3 - 3x_1 \geq 0; \\ 0, & \text{if } 3 - 3x_1 < 0 \end{cases}$$

We can rearrange the inequalities to obtain our decision boundary as:

$$y = \begin{cases} 1, & \text{if } x_1 \leq 1; \\ 0, & \text{if } x_1 > 1 \end{cases}$$

Diagrammatically, we can visualize the decision boundary as such:



### Question 1.4

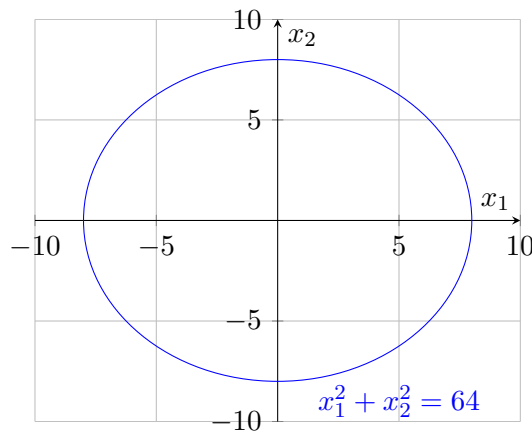
Given that  $\theta = [-64 \ 0 \ 0 \ 1 \ 1]^T$ , we have:

$$y = \begin{cases} 1, & \text{if } -64 + x_1^2 + x_2^2 \geq 0; \\ 0, & \text{if } -64 + x_1^2 + x_2^2 < 0 \end{cases}$$

We can rearrange the inequalities to obtain our decision boundary as:

$$y = \begin{cases} 1, & \text{if } x_1^2 + x_2^2 \geq 64; \\ 0, & \text{if } x_1^2 + x_2^2 < 64 \end{cases}$$

Graphically, we can visualize the decision boundary as such:



### Question 1.5

Computers use finite binary bits to represent any number. As such, when computers need to represent real numbers, they perform floating-point arithmetic, which involves some form of approximation, as well as a trade off between precision and range. By progressively multiplying many probability values together as specified by Equation 1, most of which (if not all) are not whole numbers, computer machines will quickly give us a result that is too small to be meaningfully representable in computer

memory. This is because probability values range between 0 and 1 inclusive, and as such, tend to be smaller (i.e., approach but do not equal zero) when multiplied together. This is also known as an underflow problem. In contrast, by following Equation 2, we can perform a computational sum over the terms that makes this problem less likely (although not impossible) to occur.