Solutions to Assignment 5

1. Which of the following statements are true? Check all that apply.
   A) The cost function $J(\theta)$ for logistic regression trained with $m \geq 1$ examples is always greater than or equal to zero
   B) Linear regression always works well for classification if you classify by using a threshold on the prediction made by linear regression
   C) For logistic regression, sometimes gradient descent will converge to a local minimum (and fail to find the global minimum).
   D) The one-vs-all technique allows you to use logistic regression for problems in which each $y^{(i)}$ comes from a fixed, discrete set of values.

Answer: A, B, C, D

Suppose you have the following training set, and fit a logistic regression classifier $h_{\mathbf{w}}(x) = g(w_0 + w_1 x_1 + w_2 x_2)$.

x_1	x_2	y
0.5	1	1
1.5	1	1
1	2	0
1	3	1

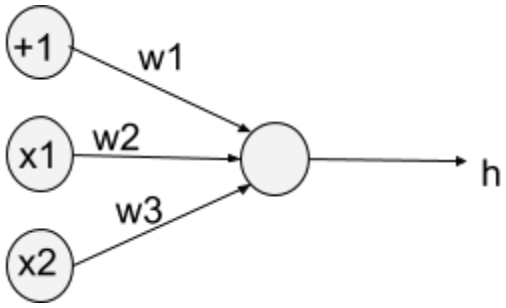
Which of the following are true? Check all that apply.

   A) At the optimal value of $\mathbf{w}$, we will have $J(\mathbf{w}) \geq 0$.
   B) If we train gradient descent for enough iterations, for some examples $x^{(i)}$ in the training set it is possible to obtain $h_{\mathbf{w}}(x^{(i)}) > 1$.
   C) Adding polynomial features (e.g., instead using $h_{\mathbf{w}}(x) = g(w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_1 x_2 + w_5 x_2^2)$) could increase how well we can fit the training data.
   D) Because the positive and negative examples cannot be separated using a straight line, linear regression will perform as well as logistic regression on this data.

Answer: A,B,C,D

The logistic sigmoid always outputs a non-negative real number less greater than or equal to zero and approaches 1 for large values of the argument. It would be instructive to plot the data points provided to see if the classes can be separated by a straight line.

2. Choose the correct combination of network weights, for it to work as AND gate.



A) w1= 30, w2= -20 ,w3= -20
B) w1= -10, w2= 20, w3= 20
C) w1= -30, w2= 20, w3= 20
D) w1= 20, w2= 10, w3= -30

Answer: C

MATLAB Code

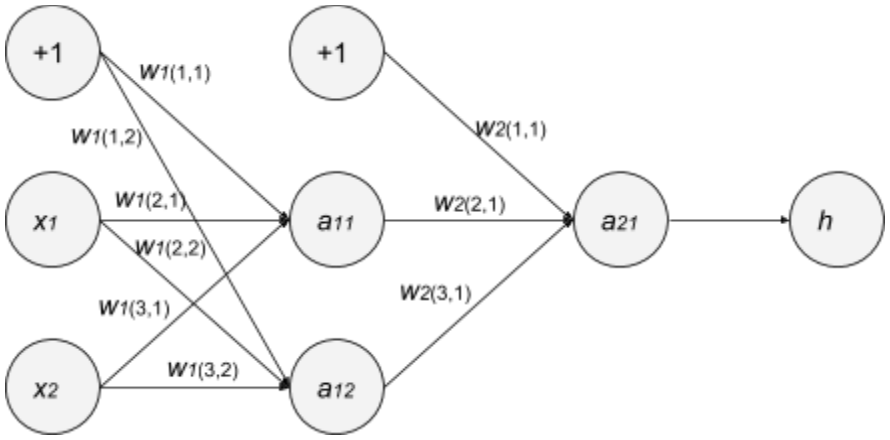
```
clc
clear
x = [0,0;0,1;1,0;1,1];   % defining x vector
m = size(x,1);           % defining m, number of examples
X = [ones(m,1),x];       % appending ones (bias) to the input vector
w = zeros(size(X,2),1);          % defining the weight matrix
%%%% case (A) %%%%
w(1) = -11; w(2) = 6; w(3) = 6;          % initializing the weights for case (A)
h = X*w;         % finding the hypothesis value
fprintf('%f\n',h)
fprintf('\n')
%%% Case (B)  %%%
w(1) = -11; w(2) = 20; w(3) = 20;       % initializing the weights for case (B)
h = X*w;                         % finding the hypothesis value
fprintf('%f\n',h)
fprintf('\n')
%%%% Case (C)  %%%
w(1) = -35; w(2) = 20; w(3) = 20;       % initializing the weights for case (B)

h = X*w;                         % finding the hypothesis value
```

```
fprintf('%f\n',h)
fprintf('\n')
%%% Case (D)  %%%
w(1) = 25; w(2) = 10; w(3) = -30;                % initializing the weights for case (B)
h = X*w;                                         % finding the hypothesis value
fprintf('%f\n',h)
fprintf('\n')
```

3. Consider the data provided below.

x1	x2	y
0	0	0
0	1	1
1	0	1
1	1	0

Given the Neural Network with one hidden layer consisting of two neurons. Also assume that activation function and cost function to be sigmoid and binary cross entropy, respectively.



What is the total number of weights (including bias) connecting the input to first hidden layer ?

A) 3
B) 4
C) 5
D) 6

Answer: D

The total number of weights connecting inputs to the first hidden layer are 6

4. What is the total number of weights (including bias) connecting the first hidden layer to the output layer
   A) 3
   B) 4
   C) 5
   D) 6

   Answer: A

   Total number of weights connecting first hidden layer to the output layer are 3

5. Initialize the weights and biases to 1. For the 3rd example (**x** = [1 0]), the corresponding output is
   A) 0.84
   B) 0.94
   C) 0.99
   D) 0.89

   Answer: B

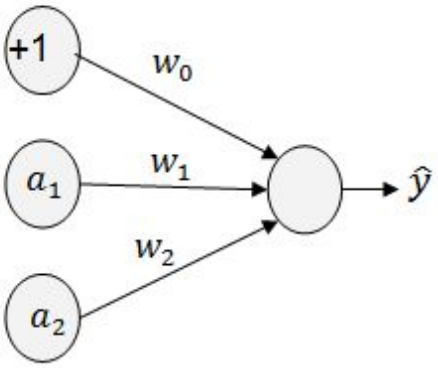
6. Initialize the weights and biases to 1. For which of the above examples will the outputs be identical?
   A) 1st and 2nd
   B) 2nd and 3rd
   C) 3rd and 4th
   D) 4th and 1st

   Answer: B

   Identical outputs are 2nd and 3rd. And the values are 0.9406 and 0.9406

7.

Consider the final set of weights of the network (that is the weights from the hidden to the output). We have shown below this layer with simplified notation for the activations and the weights.



Assume all the weights and biases (connecting all the layers) are initialized to 1. Consider further, just the 4th data point (That is, **x** = [1 1]). Answer the following questions

What is $\frac{\partial J}{\partial w_0}$?

A) 0.8413
B) 0.9458
C) 0.9991
D) 0.8997

Answer: Derivative of dJ/dw0 = 0.9481

8. What is $\frac{\partial J}{\partial w_1}$?

A) 2.8459
B) 2.9973
C) 2.9319
D) 2.8997

Answer: Derivative of dJ/dw1 = 0.9031

9. What is $\frac{\partial J}{\partial w_2}$?

A) 2.8459
B) 2.9973
C) 2.9319
D) 2.8997

Answer: Derivative of dJ/dw2 = 0.9031

MATLAB Code for Questions 4 to 10

```
clc
clear
x = [0,0;0,1;1,0;1,1];   % defining input matrix
y = [0;1;1;0];           % defining output vector
m = size(x,1);           % defining m, number of examples
hidden_1 = 2;            % defining number of hidden neurons
X = [ones(m,1),x];       % appending ones (bias) to the input matrix
w1 = zeros(size(X,2),hidden_1);% defining weight matrix connecting input to hidden
layer
w2 = zeros(hidden_1+1,1);  % defining weight matrix connecting hidden to output layer
fprintf('total number of weights connecting inputs to the first hidden layer are
\t%d\n',size(w1,1)*size(w1,2))
fprintf('total number of weights connecting first hidden layer to the  output layer are
\t%d\n',size(w2,1)*size(w2,2))
w1 = ones(size(X,2),hidden_1);       % initializing first weight matrix with ones
w2 = ones(hidden_1+1,1);             % initializing second weight matrix with ones
z1 = X*w1;                           % first hypothesis layer
a1 = 1./(1+exp(-z1));                % first activation layer
a1 = [ones(m,1),a1];                 % appending ones to first activation layer
z2 = a1*w2;                          % second hypothesis layer
a2 = 1./(1+exp(-z2));                % second activation layer
fprintf('for 3rd example, output is \t%0.4f\n',a2(3))
fprintf('identical outputs are 2nd and 3rd. values are \t %0.4f \t and %0.4f\n',a2(2),a2(3))
dJ_w = (a2(4)-y(4)).*(a1(4,:));% derivative of cost with respect to second weights
fprintf('derivative of dJ/dw0 = %0.4f\n',dJ_w(1))
fprintf('derivative of dJ/dw1 = %0.4f\n',dJ_w(2))
fprintf('derivative of dJ/dw2 = %0.4f\n',dJ_w(3))
```