Homework #3

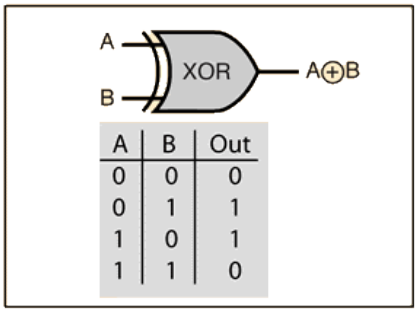
Assigned: 02/23/2021
Due: 03/09/2021 (11:59 PM on CANVAS)

A. Theory Problems

1. **Loss Functions for Logistic Regression:** As discussed in class, it is useful for a cost function to be convex for a given type of model in order to find the optimum model parameters. Assume we want to build a binary classifier using logistic regression for a single feature x and output classes $y \in \{0,1\}$. The basic logistic regression function is given by $h(\theta) = \frac{1}{1+e^{-\theta x}}$ and the estimate is given by $\hat{y} = 0$ if $h(\theta) < \frac{1}{2}$ and $\hat{y} = 1$ otherwise. Assume we have N measurements of the input and output where the n-th measurement is given by x^n and y^n with corresponding estimate $\hat{y}^n$. Note we are not including the constant (bias) term for simplicity.
   a) Show that the mean square error cost function $J_{MSE}(\theta) = \frac{1}{N}\sum_{n=1}^{N}\left(\hat{y}^{(n)} - y^{(n)}\right)^2$ is NOT convex.
   b) Show that the log-loss cost function, $J_L(\theta) = \frac{1}{N}\sum_{n=1}^{N} -y^{(n)} log(\hat{y}^{(n)}) - (1 - y^{(n)}) log(1 - \hat{y}^{(n)})$, is convex.

   **Hint: A function is convex if and only if its second derivative is always greater than zero.*

2. **Basic Neural Network for XOR:** In class we showed that the XNOR gate can be approximated by a 3-layer MLP neural network with one hidden layer.

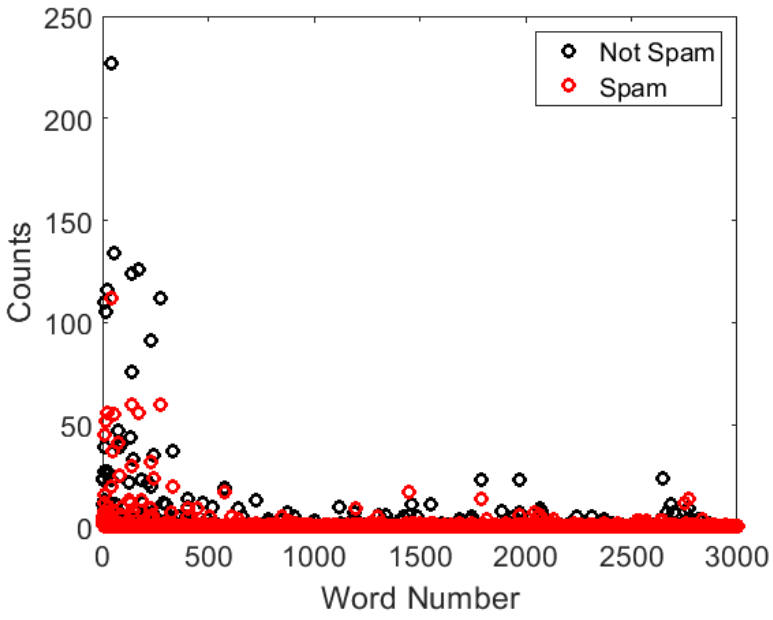


A	B	Out
0	0	0
0	1	1
1	0	1
1	1	0

   a) Design a neural network that implements the XOR gate (see figure above), explicitly show the weight and bias for each connection. **Hint: You can utilize the pre-made neurons shown in Lecture 11, Slide 31.*
   b) Show that the truth table using your neural network does in fact match the XOR truth table.

B. Coding Problems

3. **Spam Email Classifier:** For this problem you will design classifiers to determine if an email is spam or not. In the file, *spam_data.csv*¸ you can find the processed data for 5172 emails. For each email (row) the word count is shown for 3000 different possible words (columns). The last column contains the class ("1" for spam and "0" for not spam). To visualize this, the figure below shows the word count vs word number for a spam and a non-spam email.



Split the data using an 80-20 split of training to testing data. Using the training data, build a classifier model using:

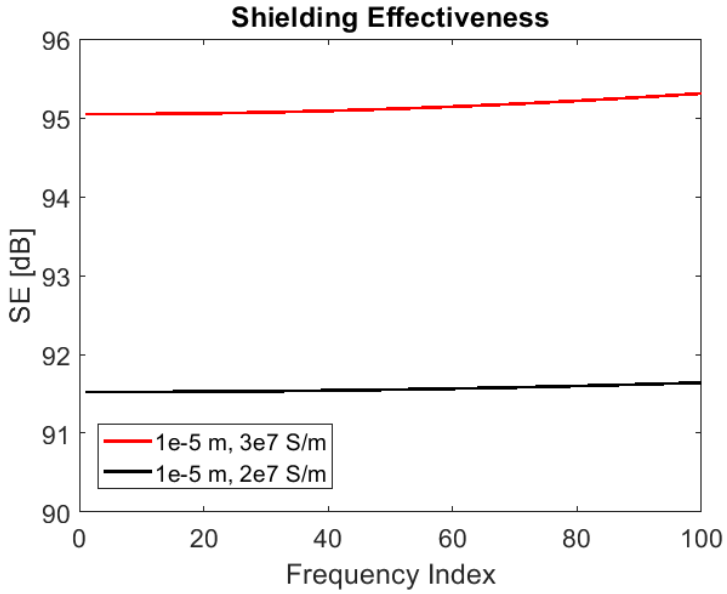
a) Logistic Regression.

b) Multi-layer perceptron with one hidden layer with **N** neurons using the sigmoid activation function. *Note*: The number **N** is an engineering parameter that will balance accuracy and computational cost. You can determine what this number is for your model.*

Run of each of the above models on the testing data and show a confusion matrix for each. ***Which model is the better option for this dataset?***

4. **Machine Learned Model of Electromagnetic Shielding:** In electromagnetic capability engineering, circuit systems must often be shielding from external radio interference. A common way to do this is by shielding the circuit behind a thin layer of highly conductive materials (like metal). The shielding effectiveness is usually estimated using theory and simulations, however, this can often be time consuming. In this problem you will design models to predict the shielding effectiveness (in dB) of a conductive sheet using pre-computed simulation data.

   The files 'Shield_inputs_train.csv' contains the conductivity and thickness of the shield for 100,000 cases. The file 'Shield_output_train.csv' contains the shielding effectiveness for 100 different frequencies (columns) for each of the 100,000 cases. A visualization for example cases are shown below.



   Using the training data, build a model using:

   a) Linear regression.

   b) Polynomial regression (degree 3).

   c) Multi-layer perceptron (MLP) neural network model (with one hidden layer containing 30 neurons) using the ReLu activation function.

   d) For the models above plot the actual and predicted shielding effectiveness as a function of frequency for the 2nd, 510th and, 4096th test cases in 'Shield_input_test.csv' and 'Shield_output_test.csv'. **Note: The actual*

frequencies are not given to you so you can simply call the x-axis "frequency index".

e) For the models from (a) and (b) make a scatterplot of the predict output vs the actual output for all the testing data. *Note*: Since the output is 100 element vector, simply treat each element of the vector as a different output and include them all on the same scatterplot.*

f) Comment on the accuracy and performance of each model. Explicitly describe the metrics you use to determine accuracy. Which model is the better option?

Hint: The units of the inputs are vastly different, make sure to scale your input features before training.*