CSC 3520
Machine Learning
Florida Southern College

Assignment 3: Neural Networks
Due: Thursday, November 1, 2018

1. Boolean Algebra (20 points)

Create a neural network with one or two hidden layers to represent each of the Boolean expressions listed below (one network per expression). Try to use as few hidden neurons as possible. Manually assign weights such that the network perfectly classifies all possible combinations of the input variables. You may assume that the activation function for all hidden and output neurons is a threshold at zero (step function). **Draw each network and show all weights; weights can be given in matrix form as long as it is clear which neurons a weight belongs to (connects).**

(a) $(A \wedge B) \oplus (A \wedge C)$

(b) $(A \oplus B) \wedge (C \oplus D)$

2. Visualizing Decision Boundaries (30 points)

In this problem, you will be feeding test data through a given neural network and visualizing the results. Be sure to include all requested plots in your submission as well as explanations of your approach (e.g., equations, pseudocode) in order to earn partial credit if your answers are not exactly correct.

Consider a 2-2-1 neural network with bias. Let the activation function at the hidden and output neurons be the hyperbolic tangent function given by $\sigma(a) = \alpha \tanh \beta a$, where $\alpha = 1.716$ and $\beta = 2/3$. Suppose the matrices defining the input-to-hidden weights ($w^{(1)}$) and the hidden-to-output weights ($w^{(2)}$) are given as, respectively,

$$\begin{bmatrix} 0.5 & -0.5 \\ 0.3 & -0.4 \\ -0.1 & 1.0 \end{bmatrix} \text{ and } \begin{bmatrix} 1.0 \\ -2.0 \\ 0.5 \end{bmatrix}$$

where $w_{ij}^{(1)}$ is the weight connecting the ith input neuron to the jth hidden neuron, and $w_{jk}^{(2)}$ is the weight connecting the jth hidden neuron to the kth output neuron. The first row in all weight matrices corresponds to the bias.

(a) Sketch the network described above and label the neurons and weights accordingly.

(b) Write a Python script (prob2.py) to evaluate the network at all points in the 2-dimensional input space defined by $\{(x_1, x_2) \in \mathbb{R}^2 \mid -5 \leq (x_1, x_2) \leq 5\}$. Use a resolution of 0.2; in other words, each dimension should vary from $-5\colon 0.2\colon 5$. This yields a total of 2601 (x_1, x_2) input points.

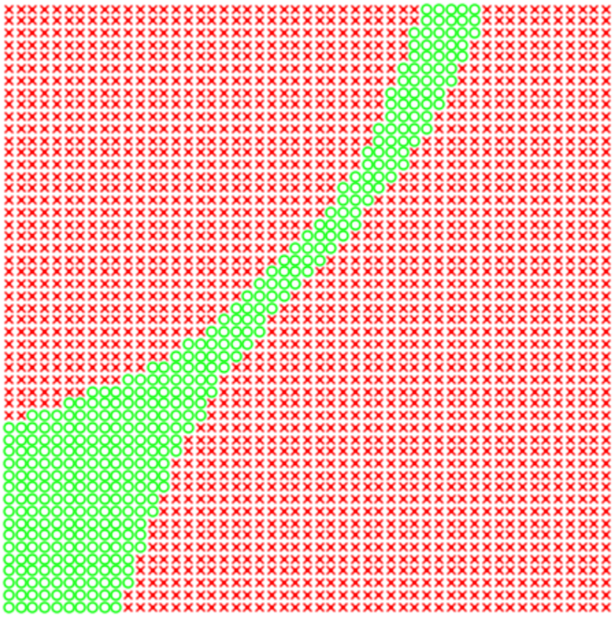
(c) Plot the results using a green ‘o’ for positive outputs and a red ‘x’ for negative outputs.

(d) What is the value of *y* (the unthresholded output of the network) at $(x_1, x_2) = (2.2, -3.2)$ and $(x_1, x_2) = (-3.2, 2.2)$?

(e) Repeat (b)-(d) for the following weight matrices:

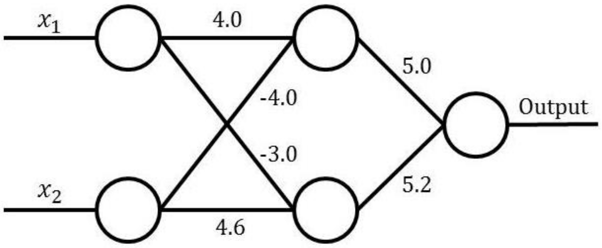
$$\begin{bmatrix} -1.0 & 1.0 \\ -0.5 & 1.5 \\ 1.5 & -0.5 \end{bmatrix} \text{ and } \begin{bmatrix} 0.5 \\ -1.0 \\ 1.0 \end{bmatrix}$$

BONUS: *(10 points max)* Considering your output plots for the two cases above, describe in a few sentences how you might design a neural network to encode the decision boundary given in the figure below. Sketch the network.



3. Backpropagation (15 points)

Consider the following 2-2-1 neural network with no bias terms:



The activation function for all hidden and output neurons in the network is the sigmoid function, $\sigma(a) = 1/(1 + e^{-x})$. Suppose you have one training example given by $(x_1, x_2) = (-0.5, 1)$.

(a) What is the output of the network for this input?

(b) If the desired output for the training example is 0.9 and the learning rate $\eta = 0.10$, what is the update to the weight (Δw) between the second hidden neuron and the output neuron (initially valued at 5.2)? Show your work.

(c) What is the update to the weight between the first input neuron and the first hidden neuron (currently valued at 4.0)? Show your work.

BONUS: *(5 points max)* Compute the updates for the remaining weights. Using the new weights, calculate the output of the network for the provided training example. Did the error decrease?

4. Handwritten Digit Recognition (35 points)

For this problem, you will use Python to train and test a neural network for recognizing handwritten digits from the popular MNIST image dataset.

To get started,

- Download the dataset (four files total) from http://yann.lecun.com/exdb/mnist/
- Extract the files to the directory of your choice (unzipped files should not have .gz extension)
- Examine the helper functions provided in mnist.py; you will use these functions to load and visualize the data

There are 60,000 samples for training and 10,000 samples for testing. Each sample contains a 28-by-28 array (image) of grayscale pixel values [0, 255]. Some example images are shown below:

Write a Python program (prob4.py) that trains and tests a neural network to predict digits (0-9) from images. You are encouraged to use Keras and TensorFlow.

(a) For your first neural network, here are the design parameters to use:
  - 784-300-10 network structure
  - inputs should be scaled to [0,1]
  - sigmoid activation function for all hidden and output neurons
  - learning rate = 0.01
  - batch size = 100
  - number of epochs for training = 1000
  - for repeatability, use a random seed of 3520

(b) How many parameters (weights) need to be learned for this network?

(c) Plot the training and testing accuracy of the network over time.

- You will need the ‘history’ of training (see model.fit)
- You will also need to include the test data as ‘validation_data’ during training (see model.fit)
- Use matplotlib.pyplot
- The x-axis is epochs and the y-axis is accuracy
- Include a legend in your plot

(d) What is the final testing accuracy and loss for this network?

(e) Print the confusion matrix for the test data. Which digit gets misclassified the most?

(f) Repeat parts (b)-(e) for a second neural network with the following design parameters:

- 784-300-300-10 network structure
- inputs should be scaled to [0,1]
- ReLU activation function for hidden layers and softmax activation function for output layer
- learning rate = 0.02
- batch size = 100
- number of epochs for training = 100
- for repeatability, use a random seed of 3520

Make sure to include your code (prob4.py) in the submitted zip file. Your answers to (b)-(e) for both networks should be included in the submitted pdf.