E9 253: Neural Networks and Learning Systems - I	**Fall 2021**
Homework #2	
Course Instructor: Prof. Shayan Srinivasa Garani	*Scribes: Kamal Bhatt*

Question 1: A student decided to use the multilayer perceptron (MLP) to solve the linear differential equation $\sum_{i=0}^{N-1} a_i \frac{d^i x}{dt^i} = y(t)$, $a_i \in \mathbf{R}$. You may assume that all the initial conditions are provided. How would you approach to set up and configure the MLP to solve this problem? Show all the details explicitly along with any assumptions made.

Solution: It must be noted that since we are following a computational approach, some assumptions must be made that $x(t)$ is evaluated over a domain i.e., $t \in [a, b]$. We also need to assume that we are given some initial conditions i.e., $x^{(i)} = x_i(0)$. For different values of t, we need to first get x as a function of t. This is where neural networks using a multilayer perceptron (MLP) with a single layer of hidden neurons can help since they can do universal approximation. We shall write $x = MLP(t)$, where an MLP is used to approximate x. Through the neural network parameters, all the functional signals can be computed to get an estimate of $x(t)$. We can also compute $\frac{d^i x}{dt^i}$ as a function of the MLP network.

We need to set up the optimization problem to obtain the neural network parameters that has the constraints specified by the linear differential equation along with the given initial conditions. The optimization problem can be formulated as:

$$\min_{MLP-wts.} \left\{ \left(\sum_{i=0}^{N-1} a_i \frac{d^i \hat{x}}{dt^i} - y(t) \right)^2 + \sum_{i=0}^{N-1} (\hat{x}_i - x_i)^2 \right\}.$$

With this framework, we are now set to solve the problem through techniques learnt in the class.

Question 2: Consider a 3-variable extension of the XOR problem done in the class. The output $y = x_1 + x_2 \mod (3)$, where x_1 and x_2 take values 0, 1 and 2. This relation is the usual addition modulo(3) over ternary values. Sketch the points along with their outputs. Derive analytically a classifier based on the MLP using a nonlinear activation function of your choice. Your final neural network architecture must have the least number of neurons. Show the explicit decision boundaries analytically and sketch them.

Solution:

Writing the input-output table for addition over $\mod (3)$.

x_1	x_2	$y = (x_1 + x_2) \mod (3)$
0	0	0
0	1	1
0	2	2
1	0	1
1	1	2
1	2	0
2	0	2
2	1	0
2	2	1

Now, consider the following activation function:

$$\varphi(x) = \begin{cases} 0, & x < 0 \\ 1, & x > 0 \\ 2, & x = 0 \end{cases}$$

Now, similar to the XOR problem solved in the class, consider the following network by extending an additional alphabet:

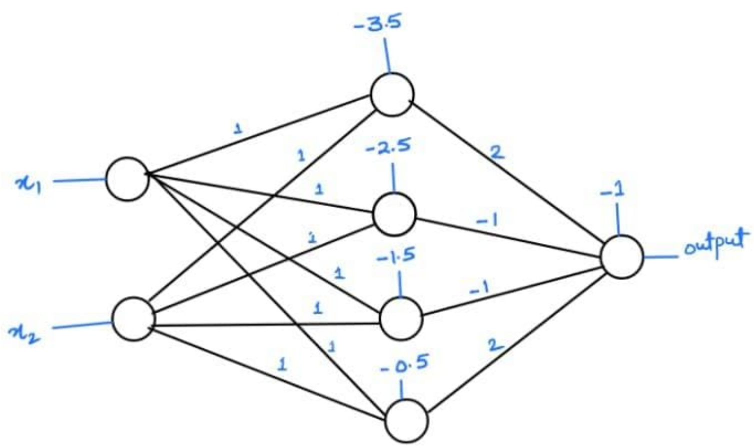

Figure 2.1: Neural Network for Question-2.

All neurons have activation function φ.
Now, let us check if the above network gives the correct output.
Input/output table:

x_1	x_2	y_1 (b=-3.5)	$\varphi(y_1)$	y_2 (b=-2.5)	$\varphi(y_2)$	y_3 (b=-1.5)	$\varphi(y_3)$	y_4 (b=-0.5)	$\varphi(y_4)$
0	0	-3.5	0	-2.5	0	-1.5	0	-0.5	0
0	1	-2.5	0	-1.5	0	-0.5	0	0.5	1
0	2	-1.5	0	-0.5	0	0.5	1	1.5	1
1	0	-2.5	0	-1.5	0	-0.5	0	0.5	1
1	1	-1.5	0	-0.5	0	0.5	1	1.5	1
1	2	-0.5	0	0.5	1	1.5	1	2.5	1
2	0	-1.5	0	-0.5	0	0.5	1	1.5	1
2	1	-0.5	0	0.5	1	1.5	1	2.5	1
2	2	0.5	1	1.5	1	2.5	1	3.5	1

$\varphi(y_1)$	$\varphi(y_2)$	$\varphi(y_3)$	$\varphi(y_4)$	out=$2\varphi(y_1) - \varphi(y_2) - \varphi(y_3) + 2\varphi(y_4) - 1$	$\varphi(out)$
0	0	0	0	-1	0
0	0	0	1	1	1
0	0	1	1	0	2
0	0	0	1	1	1
0	0	1	1	0	2
0	1	1	1	-1	0
0	0	1	1	0	2
0	1	1	1	-1	0
1	1	1	1	1	1

It can be verified that the above MLP satisfies the given operator $y = x_1 + x_2 \mod (3)$. It is to be noted that the hidden layer has only 4 neurons and 1 neuron at output layer, and this is the *least* number of neurons such that the network satisfies the given relation.
Following the outputs from the neurons, the decision boundaries are evaluated as:

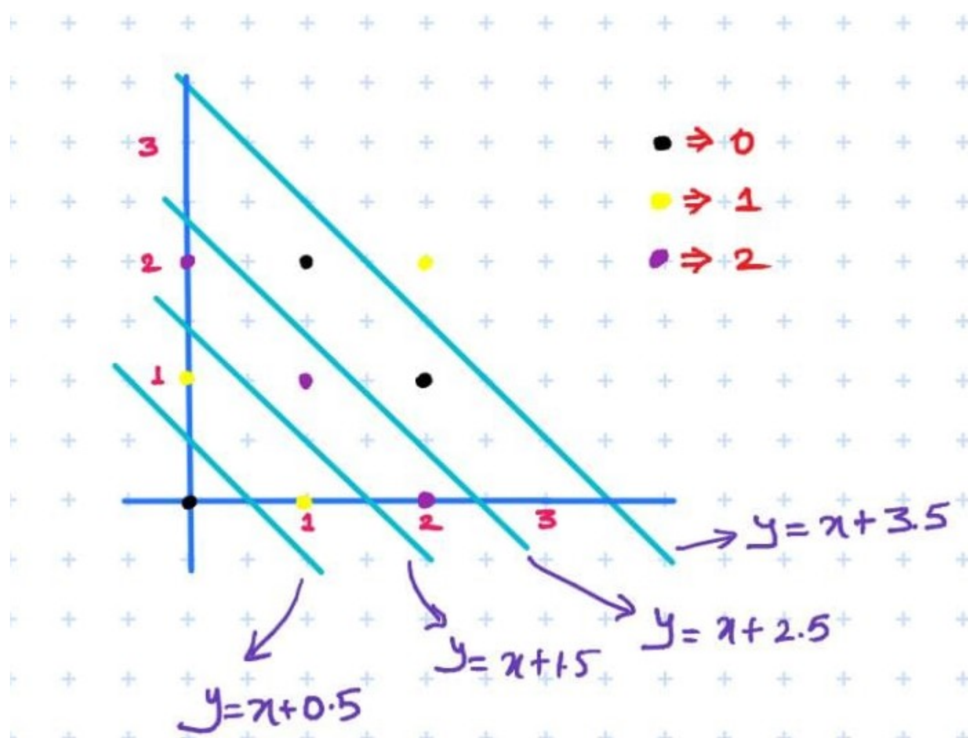

Figure 2.2: Decision boundaries for Question-2.

It can be seen clearly that the separate the points into their respective classes. Just like in XOR problem, we constructed a hidden layer to represent the decision boundaries.