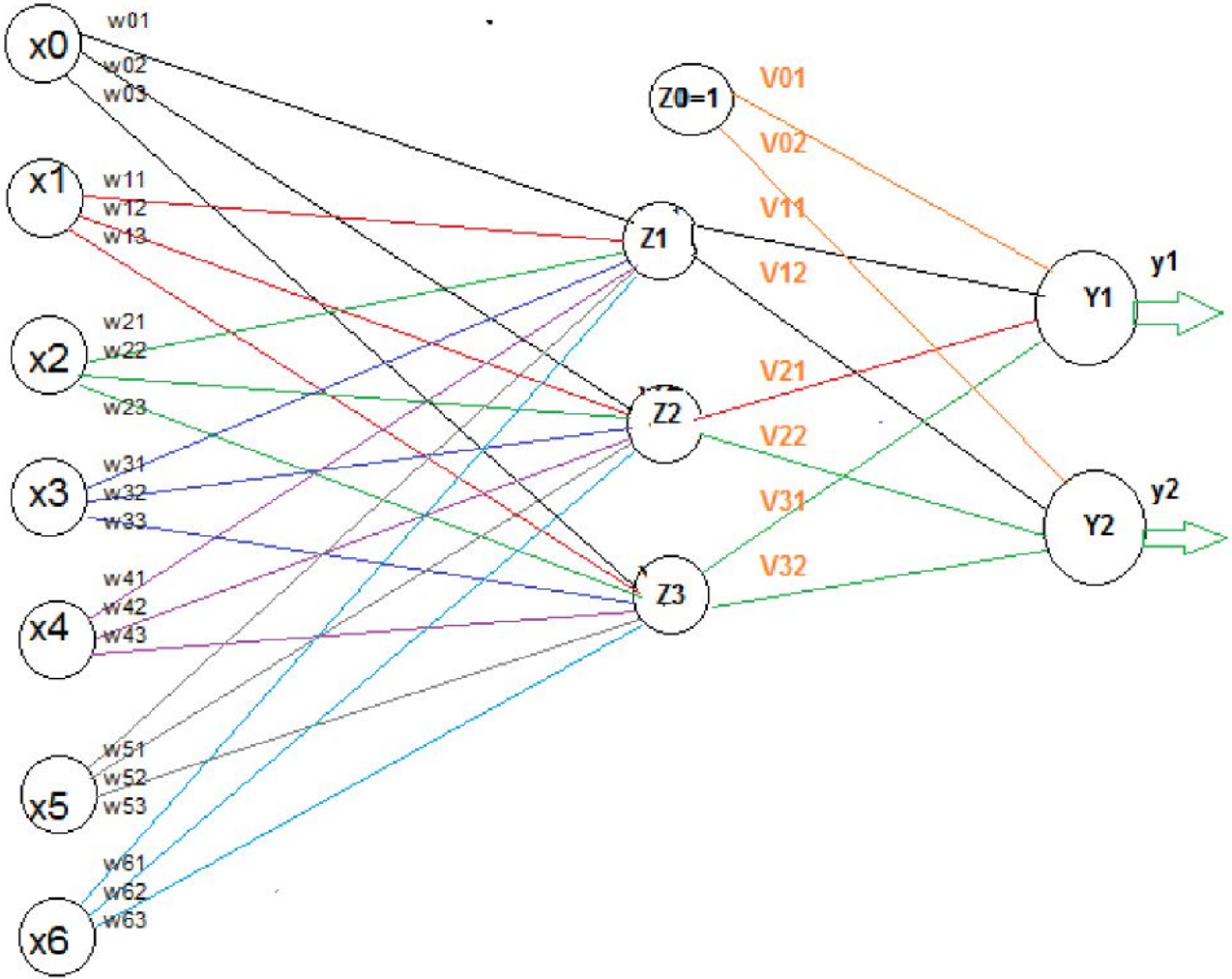
Question1 (5)

Given that we have a training data containing 6 features {x1, x2….x6} and two outputs {y1,y2} for certain classification problem. This problem is not linearly separable, however if we transform data into 3D (using some function) it becomes linearly separable.

Draw an Artificial Neural Network architecture for above problem.


x0
x1
x2
x3
x4
x5
x6
w01
w02
w03
w11
w12
w13
w21
w22
w23
w31
w32
w33
w41
w42
w43
w51
w52
w53
w61
w62
w63
Z0=1
Z1
Z2
Z3
V01
V02
V11
V12
V21
V22
V31
V32
Y1
Y2
y1
y2

Question1 (5)

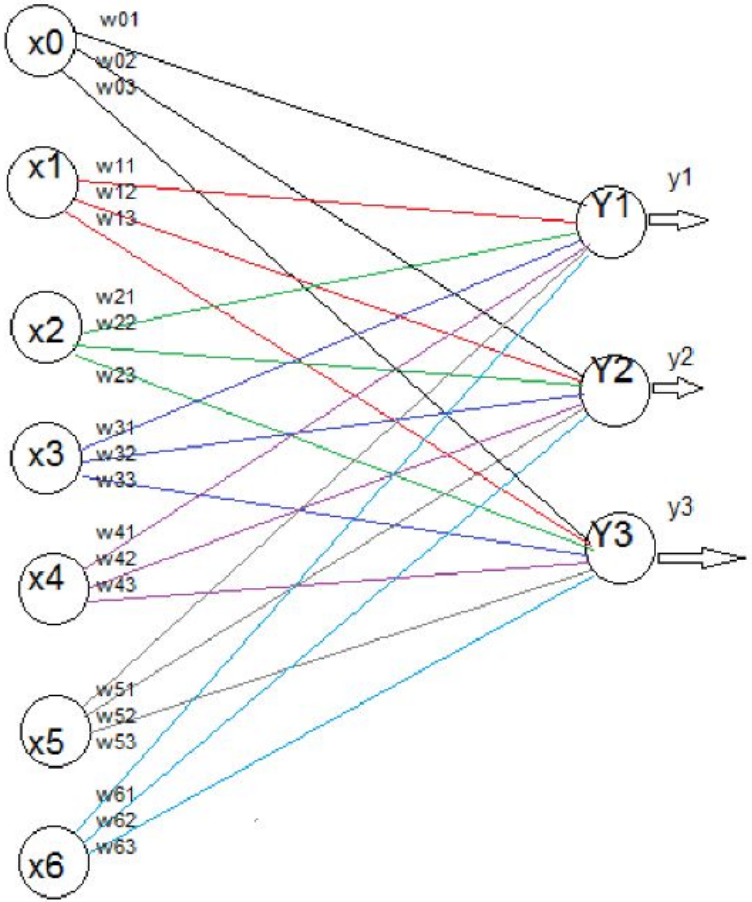
We are given a training data containing 6 features {x1, x2….x6} and one response {y} for certain multiclass classification problem. y can be amongst one of the following classes {c1, c2, c3, c4, c5}. The problem is linearly separable.

Draw a minimal Artificial Neural Network Architecture for the given problem. Also show the mapping of your ANN to the given classes.

Table 1Mapping of output to classes

Table 1Mapping of output to classes

y1	y2	y3	Class
0	0	0	C1
0	0	1	C2
0	1	0	C3
0	1	1	C4
1	0	0	C5


x0
x1
x2
x3
x4
x5
x6
w01
w02
w03
w11
w12
w13
w21
w22
w23
w31
w32
w33
w41
w42
w43
w51
w52
w53
w61
w62
w63
Y1
Y2
Y3
y1
y2
y3

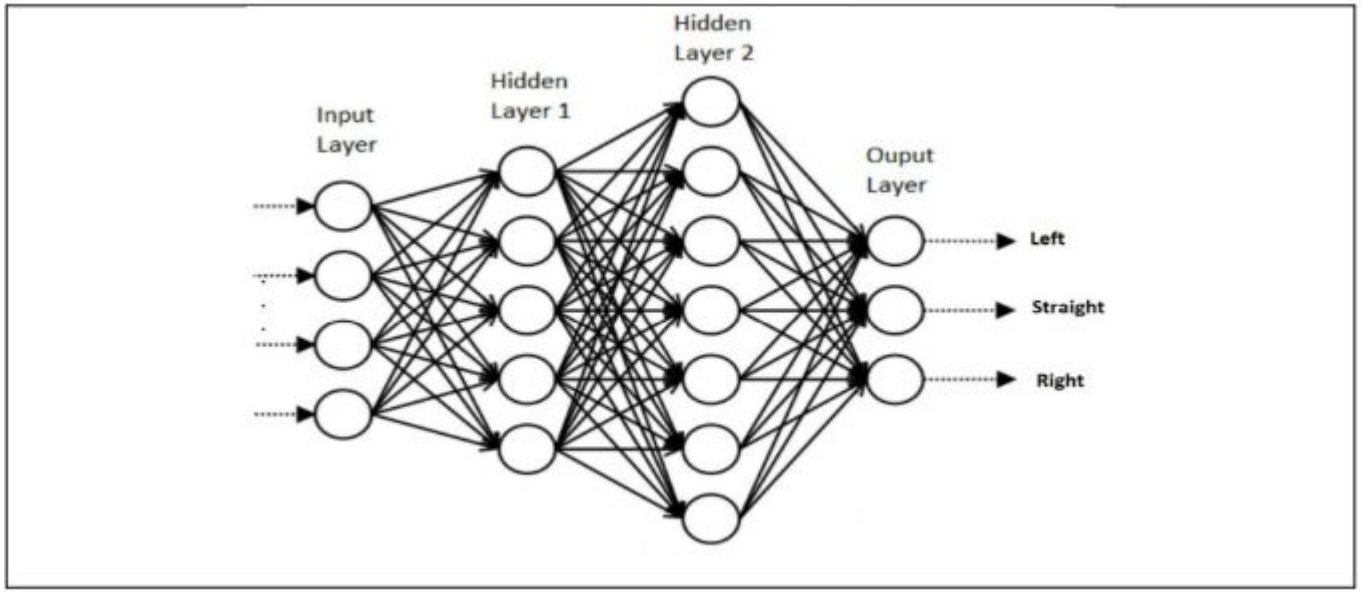
Q No 5. [Neural Networks]

Inspired by the movie WALL-E a teacher at FAST-LHR is designing a line following robot. You might search/google Internet to see various line following robots but don't spend much time surfing the Internet as this is a limited time Exam.

Being aware of the representation and learning power of a feed-forward artificial neural network, he decided to use such a neural network with two hidden layers of neurons for creating the main controller/brain of the robot. The initial prototype of the robot will only follow smooth lines i.e. lines without any sharp turns.

Input to the controller/brain of this robot will be a grey-scale camera image of dimensions **N x N**. The number of neurons in the first hidden layer will be **H1** and the number of neurons in the second hidden layer will be **H2** whereas there will be **3** neurons in the output layer with each output being **between 0 and 1** specifying the direction in which the robot need to turn. The robot will be turned slightly towards left, right or will go straight depending upon the largest value of the output neuron **Left, Straight or Right** respectively.

Such a neural network is depicted in the figure below



In this question we will assume that **N is equal to the last four digit number in you university ID whereas H1** is the **number consisting of the year/batch number** in your ID and **H2** is the **most significant two digits of N**. For example if your id is L17-**4526** then N is 4526, H1 is 17 and H2 is 45.

Part a) [Neural Network Weights] **[2 Points]**

Specify the number of weights needed to create such a neural network. Assume that each neuron in this network has a bias term as well. Further if we assume that each weight is a double precision number represented using 8-bytes, approximately how many kilo bytes will be needed to store this neural network on some storage.

Solution for Roll No: 17L-4526: 2720628.57

Part b) [Neural Network Computations] **[3 Points]**

Assuming that the camera is generating 30 frames/Images per second and the neural network will be used to make a decision for each one of the frames, how many floating point multiplications and additions will be computed per second if this network is used for making decision.

Solution for Roll No: 17L-4526: multiplications: 10447211760 additions: 10447213710

In this question we will assume that **N** is 4526, **H1** is 17 and **H2** is 45.

Part a) [Neural Network Weights] **[2 Points]**

Specify the number of weights needed to create such a neural network. Assume that each neuron in this network has a bias term as well. Further if we assume that each weight is a double precision number represented using 8-bytes, approximately how many kilo bytes will be needed to store this neural network on some storage.

Total weights = 348240457

Storage = 2720628.57KB or 2785923656 BYTES

Part b) [Neural Network Computations] **[3 Points]**

Assuming that the camera is generating 30 frames/Images per second and the neural network will be used to make a decision for each one of the frames, how many floating point multiplications and additions will be computed per second if this network is used for making decision.

Total weights excluding bias = 348240392 **total weights = 348240457**
Total multiplications = 348240392 * 30 = 10447211760 ---------> 1 mark

Total weights = 348240457
Total Additions = 348240457 * 30 = 10447213710 ---------> 1 mark

Total computations = 20894425470 ---------> 1 mark

Part c) [Neural Network Coding] **[5 Points]**

Write a C++ function that can be used to compute output of a single network layer. Assume that the activation function is **sigmoid** for each neuron of the layer and that the weights of neurons of that layer are stored in a global array of weights **W** having dimensions **H x N** i.e. **H** neurons each has **N** weights. Your function will have only one parameter i.e. a vector/array X of inputs and will return a vector/array **O** of computed outputs.

A nested loop computing product of matrix and vector and once the sum has been accumulated the activation function must be present

ARRAY VERSION (Variables W[][], H and N are assumed to be globally defined

```
void computeLayerO(double X[], double O[]){
        for(int neuron = 0; neuron< H; neuron++){
                O[neuron] = 0;
                for(int feature = 0; feature< N; feature++)
                                O[neuron]+= W[neuron][feature]*X[feature];
                O[neuron] = 1.0/ (1 + exp(O[neuron]));
        }
        return;
}
```

A vector version will have a similar structure with the variable O being defined as a vector and returned at the end and the parameter X also replaced with a vector …

Part d) [Preparing Training Data] **[2 + 3 Points]**

We know that weights of a neural network are typically learned using back-propagation learning algorithm which is a supervised learning algorithm.

i) What do we mean by supervised learning? Give names of three supervised learning algorithms for classification.

When we learn model parameters using LABELED TRAINING DATA.

i. Neural Networks
ii. Decision Trees
iii. SVM
iv. Perceptron Learning
v. KNN

ii) Briefly describe some method to prepare training data for learning weights of the network described in the previous parts.

SUBJECTIVE: ANY VALID WAY OF PREPARING TRAINING DATA CONSISTING OF IMEGES AND CORRESPONDING LABELS IN THE FORM OF BINARY VECTOR

For example a video in which a human is controlling the bot can be used with each frame used as an image and the drivers action used as a label.

Reward (up to 7 marks added into your mid1 exam)

You have to implement the ANN's Feed Forward Network and train it using GA. You will use the Irsi/Fisher's Flower data set. This data set has three classes. You can assign a class label of 1, 2 and 3 to these classes.

The ANN will have the following configuration:

1. Input layer: 4 nodes
2. Hidden layer: 10 nodes
3. Output layer: same as number of classes

You will use sigmod as activation functions. The population size is 100.

CONSTANTS

Typically, the chromosome size would be (4 * 10 + 10 + 10* 3 + 3) but you can experiment with different chromosomes.

Your goal is to achieve the best accuracy (in comparison with other students of AI-C and AI-D) without changing the ANN's architecture.

You can play around with different schemes and try out different things to get an improved accuracy. You can try different fitness functions. You can also have a more targeted approach where you only target/mutate/crossover a specific **part of the chromosome** because it is either very fit or very unhealthy etc.

Question 1:

a) Suppose you are given with a 1024*1024 image and 5 layers of convolution+pooling are applied before fully stacked neural network. Compute weight and biases at each level and dimensions of resultant matrix. Consider stride size is 2 throughout. Pooling window size is also 2 by 2 (except last cell, it uses its own pooling window size and stride). You can use this formula to compute after pooling dimensions: **12 Marks**

(Input image width – filter image width)*(1/stride)+1

Weights are number of elements in each filter while for each filter there is only 1 bias. For example for 4 3*3 filters weights will be 3*3*4 = 36 and biases will be 4 (1 for each filter).

Input image dim	Filter dims	No. of filters	Weights,biases	Resultant image	After pooling
1024*1024*1	3*3	4	3*3*4= 36, 4	Floor(511.5)= 511*511*1*4	Floor(255.5)= 255*255*4
255*255*4	2*2	2			
	2*2	1			
	2*2	4			
	1*1	2			Pooling filter= 1*1 and stride=1

b) After this a fully connected Neural network is implemented which uses extracted features of above layers. 2 hidden layers and 1 output layer is attached. Hidden layer 1 has 32 neurons while hidden layer 2 has 15 neurons and model is used to perform multi classification between 10 classes. Softmax is applied at output layer. What will be the dimensions of weight matrices (excluding biases) between all layers? 3 Marks