INDEX

Name : Vishal Pravin Jotti

Std : FYMCA Div : IInd Sem Roll No : 26

Sub : Soft Computing (Assingment)

School / College : GECCS

Sr. No.	Date	Title	Remark	Pg. No.

Q1. What is pattern? What are different approaches in pattern recognition? How data is classified using pattern classification? Explain

Pattern -
Pattern is everything around in this digital world. A pattern can be either seen physically or it can be observed mathematically by applying algorithms.

Different approaches in pattern recognition include:

1) Statistical Pattern Recognition -
This approach involves modeling patterns using statistical techniques such as probability distributions.

2) Neural Networks -
Utilizing ANNs to recognize patterns by learning from example.

3) Machine learning -
Employing various machine learning algorithm to identify pattern and make predictions based on training data.

4) Template Matching -
Comparing input data with predefined templates or prototypes.

5) clustering -

Grouping input similar data points together based on certain features or characteristics.

Process of data classification using pattern

i) Feature Extraction -
Identifying relevant features in the input data that distinguish patterns

ii) Training -
Using a labeled dataset to teach the classifier how to recognize different patterns.

iii) Testing -
Evaluating the classifier's performance on new unseen data to assess it's ability to generalize.

iv) Classification -
Assigning input data to one or more predefined classes based on the learned patterns.

Q. 2. What is neuron function? Explain the neuron signal function with suitable example.

Neurons are the basic building blocks of the nervous system function of neurons is the transit signals in the form of electrical impulses A neuron receives signals through it's dendrites processes this information in the cell body and the transmits the signal along its axon to communicate with other neurons or effector cells.

i) Receive information
ii) Integrate incoming signals
iii) communicate signals to target cells

The signal transmission in a neuron can be explained through the following step

1) Reception -
Dendrites receive from other neurons or sensory receptors. These signals are typically in the form of neurotransmitters.

2) Integration -
The cell body integrates the received signals. If the combined signals reach a certain threshold an action potential is generated. This is a rapid, electrical impulse that travels along the axon of the neuron.

3) Propogation -
The action potential travels down the axon

by ion channels and the movement of ions across the cell membrane.

4) Axon Terminal -
The action potential reaches the axon terminal prompting the release of neurotransmitter into the synapse.

5) Synaptic Transmission -
Neurotransmitters cross the synapse and bind to receptors on the dendrites of the neighboring neuron, transmitting the signal to continue the process.
e.g.
Consider a sensory neuron responsible for conveying the sensative of heat. When you touch a hot surface, specilized receptors in your skin generate electrical signals. These signals are then transmitted as action potential the sensory neuron. The neurons ultimately leading to the perception of heat in your brain

Q.3 What are linear and non-linearly seperable problems? How it is classified using various algorithms using ANNs.

A) Linear seperable -
In a linearly seperable problems a straight line can completely seperate the classes.
e.g. Imagine a 2D space where points from two classes can be perfectly seperated by a straight line.

B) Non-linear seperability -
In a non-linearly seperable problems, a simple straight line cannot effectively divide the classes and a more complex boundary is required.
e.g. Points from two classes may be intertwinded in a way that cannot be accurately seperated by a straight line.

Classification using Artificial Neural Networks.
i) Linear Separability in ANN -
For linearly separable problems, a single layer perceptron (SLP) can be sufficient. SLP's have one layer i/p nodes and a single layer of o/p nodes with a linear activation function. They can be learn and classify linearly seperable patterns.

ii) Non-linearly separability in ANN -
For non-linearly separatable problems multi-

layer perceptrons (MLP's) with hidden layers and non-linear activation function are often used. The addition layers and non-linear activation function allows ANNs models and learn complex non-linear relationships in the data.

iii) Activation Function -

Common non-linear activation functions include:

- Sigmoid - sequashes o/p between 0 and 1
- Hyperbolic Tangent (tanh) - sequenshes o/p between -1 and 1.
- Rectified liner unit (Relu) - Allows positive values to pass through while negative values are set to zero.

iv) Hidden layers -

The introduction of hidden layers in ANNs provides the capacity to learn intricate patterns making them suitable for non-linearly seperatble problems. The back propogation algorithm is common used to train ANNs by adjusting weights based on the error in prediction.

Q 4. What is linear separable problem? How it is solved using logic gates? Give the example of few gates with sample data

Linearly seperable problem is one where the classes or categories can be separated by a straight line. Essentially. It means that there exists a linear decision boundray that can accurately distingwish between different classes in the dataset.

Solving linearly separate problems using logic gates -

Logic gates can be used to represent simple decision boundries. Each logic gate correspond to a specific operation. and combinations of these gates can create more complex decision rules. Here are a few examples of logic gates.

1) AND Gate -

operation - output is true(1) only if both inputs are true(1)

e.g.	a	b	output
	0	0	0
	0	1	0
	1	0	0
	1	1	1

2) OR Gate -

operation - output is true(1) if an element at least one input is true(1)

e.g.

a	b	output
0	0	0
0	1	1
1	0	1
1	1	1

3) NOT Gate -

Operation - Output is the opposite of the input.

e.g.

a	output
0	1
1	0

4) XOR Gate -

Operation - Output is true(1) if exactly one input is true

e.g.

a	b	output
0	0	0
0	1	1
1	0	1
1	1	0

Q5 What non-linearly seperable problem? Explain XOR problems with sample data.

A non-linearly seperable problem is one where classes or categories cannot be seperated by a straight line. Instead a more complex decision boundary involving curves or non-linear shapes is required to accurately distinguish between different classes in the data set.

XOR problem -
The XOR problem is a classic example of a binary operation that returns true only if the number of true inputs are odd. The XOR problem arises when trying to find a linear decision boundary to seperate the two classes of XOR data points.

XOR Truth table

a	b	output
0	0	0
0	1	1
1	0	1
1	1	0

In the XOR truth table the o/p is true if the inputs are different and false if the inputs are the same. Attempting to use a single straight line to seperate the points with o/p 1 from those with o/p

is not possible.

If you plot the XOR data points in a 2D spare with input 1 on the x-axis and input 2 on the y-axis, you will find that no single straight line can accurately divide the data into two classes.

1	0	x
0	1	x
0	1	

Here x represents data points and you can see that a straigh line cannot seperate the point with output 1 from these with output 0.

Q. 6 State and explain preceptron learning algorithm

The perceptron learning algorithm is a supervised learning algorithm used for binary classification tasks. It was developed by Frank Rosenblatt in 1957. and is foundation for building neural networks. The perceptron is a simple model of biological neuron, and it's learning algorithm involves adjusting weights to learn a decision boundary that seperates different classes steps of perceptron learning algorithm -

1) Initialize weight and Bios.
Initialize the weight (w) and bios (b) with small random values.

2) Input and Activation -
For each training example, calculate the weighted sum of the input features and add the bios. This is the input to the activation function.

The activation function, traditionally a step function, determines the output (0 - 1) based on whether the weighted sum is above or below a threshold.

3) Update weights -
If the perceptron misclassifies an example update the weight and bios to reduce the error.

The weight update formula for the i-th weight (wi) is

wi = wi + learning_rate * (desired_output - actual_op) * input_i;

The bias is updated similarly:

b = b + learning_rate * (desired - actual_op)

9) Repeat -

Repeat step 2 and 3 for each learning example in the dataset

Iterate through the dataset multiple times achieves satisfactory performance.

The goal of perceptron learning algorithm is to find weights and a bias that allo the perceptron to correctly classify all training examples. It's important to note that the perceptron is limited to solving linearly separable problems. It can't learn non-linear decision boundries.