




**G.S. Mandal's
Maharashtra Institute of Technology, Aurangabad
Department of Computer Science and Engineering**

LAB MANUAL

**CSE 421: Artificial Intelligent and Deep
Learning
(2019-20 Part 1)**

Maharashtra Institute of Technology, Aurangabad
NH-211, MIT Campus, Satara Village Road, Aurangabad- 431 010 (M.S.); India.
Phone: (0240) 2375222; Fax: (0240) 2376618, E-mail: principalmitt@mit.asia
Website: www.btech.mit.asia

	G.S. MANDAL'S	
	MAHARASHTRA INSTITUTE OF TECHNOLOGY, AURABGABAD	
	LAB WORK INSTRUCTION SHEET	
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING		
CLASS: B. TECH PART: 1 (2019-20)	LAB: 506B	SUBJECT: CSE 421 Artificial Neural Network And Deep Learning

Index

Contents		Page No.
Vision & Mission		2
Program Educational Objectives		3
Program Specific Objectives		3
Program Outcomes		4
Course Outcomes		5
Mapping		5
Exp No.	Title of Experiment	Page No.
1	To Write a program to implement Perceptron.	6
2	To write a program to implement AND OR gates using Perceptron.	11
3	To implement Crab Classification using pattern net	13
4	To write a program to implement Wine Classification using Back propagation.	16
5	To write a MatLab Script containing four functions Addition, Subtraction, Multiply and Divide functions	19
6	Write a program to implement classification of linearly separable Data with a perceptron	21
7	To study Long Short Term Memory for Time Series Prediction	23
8	To study Convolutional Neural Network and Recurrent Neural Network	26
9	To study ImageNet, GoogleNet, ResNet convolutional Neural Networks	26
10	To study the use of Long Short Term Memory / Gated Recurrent Units to predict the stock prices based on historic data	27



G.S. MANDAL'S

MAHARASHTRA INSTITUTE OF TECHNOLOGY, AURANGABAD

LAB WORK INSTRUCTION SHEET

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CLASS: B. TECH
PART: 1 (2019-20)

LAB: 506B


SUBJECT: CSE 421 Artificial Neural Network And Deep Learning

Vision

To develop the department as a center of excellence in the field of computer science and engineering by imparting knowledge & training to the students for meeting growing needs of the industry & society.

Mission

Providing quality education through a well-designed curriculum in tune with the challenging needs of software industry by providing state of the art facilities and to impart knowledge in the thrust areas of computer science and engineering.

	G.S. MANDAL'S	
	MAHARASHTRA INSTITUTE OF TECHNOLOGY, AURANGABAD	
	LAB WORK INSTRUCTION SHEET	
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING		
CLASS: B. TECH PART: 1 (2019-20)	LAB: 506B	SUBJECT: CSE 421 Artificial Neural Network And Deep Learning

Program Educational Objectives

PEO1: To prepare the students to achieve success in Computing Domain to create individual careers, innovations or to work as a key contributor to the private or Government sector and society.

PEO2: To develop the ability among the students to understand Computing and mathematical fundamentals and apply the principles of Computer Science for analyzing, designing and testing software for solving problems.

PEO3: To empower the students with ability to quickly reflect the changes in the new technologies in the area of computer software, hardware, networking and database management.


PEO4: To promote the students with awareness for lifelong learning, introduce them to professional practice, ethics and code of professionalism to remain continuous in their profession and leaders in technological society.

Program Specific Objectives

PSO1: Identify appropriate data structures and algorithms for a given contextual problem and develop programs to design and implement applications.

PSO3: Design and manage the large databases and develop their own databases to solve real world problems and to design, build, manage networks and apply wireless techniques in mobile based applications.

PSO3: Design a variety of computer-based components and systems using computer hardware, system software, systems integration process and use standard testing tools for assuring the software quality.

	G.S. MANDAL'S	
	MAHARASHTRA INSTITUTE OF TECHNOLOGY, AURABGABAD	
	LAB WORK INSTRUCTION SHEET	
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING		
CLASS: B. TECH PART: 1 (2019-20)	LAB: 506B	SUBJECT: CSE 421 Artificial Neural Network And Deep Learning

Program Outcomes

PO1: Apply knowledge of mathematics, science, and engineering fundamentals to solve problems in Computer science and Engineering.

PO2: Identify, formulate and analyze complex problems.

PO3: Design system components or processes to meet the desired needs within realistic constraints for the public health and safety, cultural, societal and environmental considerations.

PO4: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data for valid conclusions.

PO5: Select and apply modern engineering tools to solve the complex engineering problem.

PO6: Apply knowledge to assess contemporary issues.

PO7: Understand the impact of engineering solutions in a global, economic, environmental, and societal context.


PO8: Apply ethical principles and commit to professional ethics and responsibilities.

PO9: Work effectively as an individual, and as a member or leader in diverse teams and in multidisciplinary settings.

PO10: Communicate effectively in both verbal and written form.

PO11: Demonstrate knowledge and apply engineering and management principles to manage projects and in multi-disciplinary environment.

PO12: To engage in life-long learning to adopt to the technological changes.

	G.S. MANDAL'S	
	MAHARASHTRA INSTITUTE OF TECHNOLOGY, AURABGABAD	
	LAB WORK INSTRUCTION SHEET	
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING		
CLASS: B. TECH PART: 1 (2019-20)	LAB: 506B	SUBJECT: CSE 421 Artificial Neural Network And Deep Learning

Course Outcomes:

After Completing the course students will be able to

CO1: Understand the characteristics and types of artificial neural network and remember working of biological Neuron and Artificial Neural Network.

CO2: Apply learning algorithms on perceptron and apply back propagation learning on Neural Network.

CO3: Apply Feedback NN and plot a Boltzmann machine and associative memory on various application

CO4: Apply different types of auto encoders with dimensionality reduction and regularization.

CO5: Design Convolutional Neural Network and classification using Convolutional Neural Network.

CO6: Solve sequence learning problem and implement long short term memory and gated recurrent units.



CLASS: B. TECH
PART: 1 (2019-20)

LAB: 506B

SUBJECT: CSE 421 Artificial Neural Network And Deep Learning

Experiment#1

Aim: To Write a program to implement Perceptron.

Objective: To study the structure, contents and working principle of basic Perceptron.

Outcomes: To design a perceptron and train and test it for an application

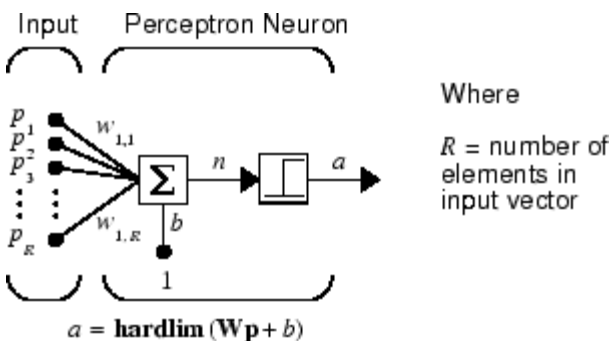
Theory:

Perceptron Neural Networks

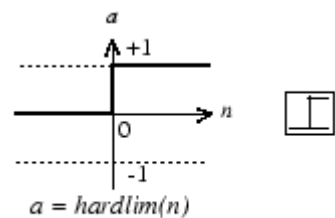
Rosenblatt created many variations of the perceptron. One of the simplest was a single-layer network whose weights and biases could be trained to produce a correct target vector when presented with the corresponding input vector. The training technique used is called the perceptron learning rule. The perceptron generated great interest due to its ability to generalize from its training vectors and learn from initially randomly distributed connections. Perceptrons are especially suited for simple problems in pattern classification. They are fast and reliable networks for the problems they can solve. In addition, an understanding of the operations of the perceptron provides a good basis for understanding more complex networks.

Neuron Model

A perceptron neuron, which uses the hard-limit transfer function `hardlim`, is shown below.



Each external input is weighted with an appropriate weight w_{1j} , and the sum of the weighted inputs is sent to the hard-limit transfer function, which also has an input of 1 transmitted to it through the bias. The hard-limit transfer function, which returns a 0 or a 1, is shown below.



Hard-Limit Transfer Function



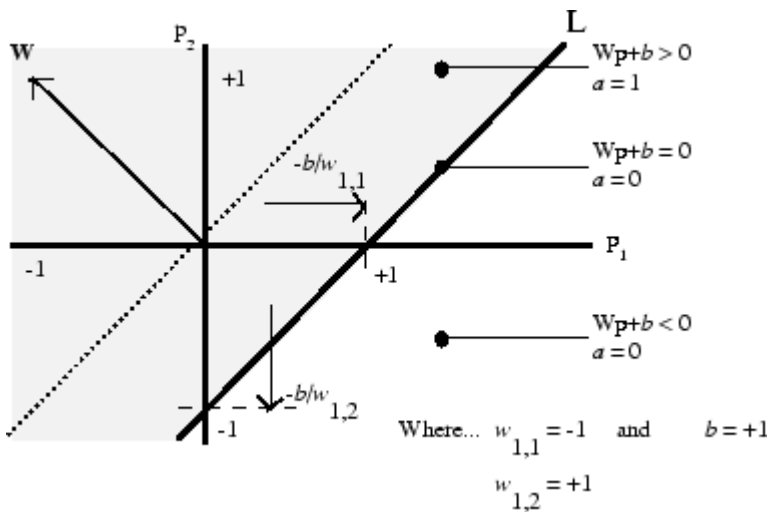
CLASS: B. TECH
PART: 1 (2019-20)

LAB: 506B

SUBJECT: CSE 421 Artificial Neural Network And Deep Learning

The perceptron neuron produces a 1 if the net input into the transfer function is equal to or greater than 0; otherwise it produces a 0.

The hard-limit transfer function gives a perceptron the ability to classify input vectors by dividing the input space into two regions. Specifically, outputs will be 0 if the net input n is less than 0, or 1 if the net input n is 0 or greater. The following figure show the input space of a two-input hard limit neuron with the weights $w_{1,1} = -1$, $w_{1,2} = 1$ and a bias $b = 1$.



Two classification regions are formed by the *decision boundary* line L at $\mathbf{Wp} + b = 0$. This line is perpendicular to the weight matrix \mathbf{W} and shifted according to the bias b . Input vectors above and to the left of the line L will result in a net input greater than 0 and, therefore, cause the hard-limit neuron to output a 1. Input vectors below and to the right of the line L cause the neuron to output 0. You can pick weight and bias values to orient and move the dividing line so as to classify the input space as desired.

Hard-limit neurons without a bias will always have a classification line going through the origin. Adding a bias allows the neuron to solve problems where the two sets of input vectors are not located on different sides of the origin. The bias allows the decision boundary to be shifted away from the origin, as shown in the plot above.

You might want to run the example program nnd4db. With it you can move a decision boundary around, pick new inputs to classify, and see how the repeated application of the learning rule yields a network that does classify the input vectors properly.

Perceptron Architecture

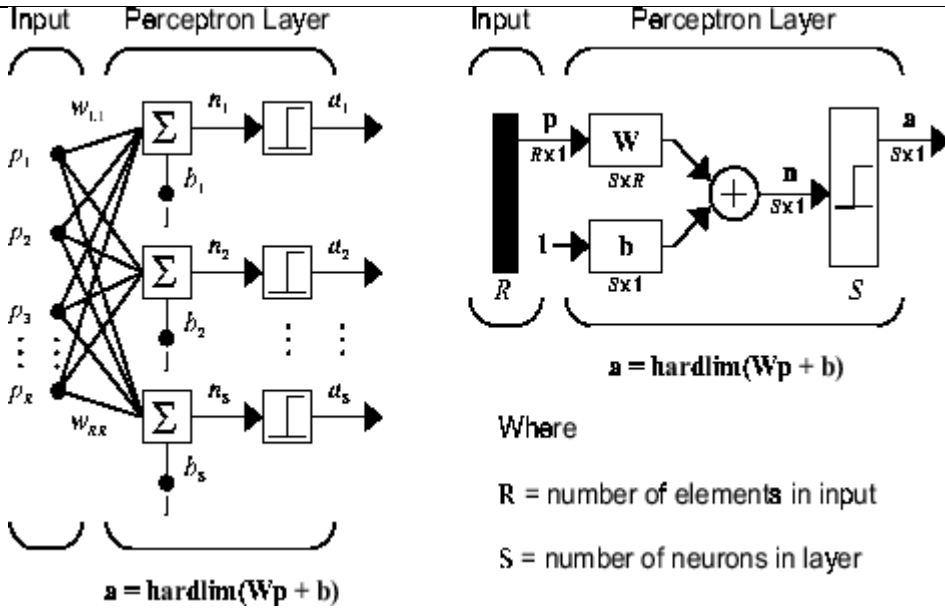
The perceptron network consists of a single layer of S perceptron neurons connected to R inputs through a set of weights $w_{i,j}$, as shown below in two forms. As before, the network indices i and j indicate that $w_{i,j}$ is the strength of the connection from the j th input to the i th neuron.



CLASS: B. TECH
PART: 1 (2019-20)

LAB: 506B

SUBJECT: CSE 421 Artificial Neural Network And Deep Learning



The perceptron learning rule described shortly is capable of training only a single layer. Thus only one-layer networks are considered here. This restriction places limitations on the computation a perceptron can perform.

Perceptron Learning Rule (learnp)

Perceptrons are trained on examples of desired behavior. The desired behavior can be summarized by a set of input, output pairs


$$p_1t_1, p_2t_1, \dots, pQt_1$$

where \mathbf{p} is an input to the network and \mathbf{t} is the corresponding correct (target) output. The objective is to reduce the error \mathbf{e} , which is the difference $\mathbf{t} - \mathbf{a}$ between the neuron response \mathbf{a} and the target vector \mathbf{t} . The perceptron learning rule `learnp` calculates desired changes to the perceptron's weights and biases, given an input vector \mathbf{p} and the associated error \mathbf{e} . The target vector \mathbf{t} must contain values of either 0 or 1, because perceptrons (with `hardlim` transfer functions) can only output these values.

Each time `learnp` is executed, the perceptron has a better chance of producing the correct outputs. The perceptron rule is proven to converge on a solution in a finite number of iterations if a solution exists.

If a bias is not used, `learnp` works to find a solution by altering only the weight vector \mathbf{w} to point toward input vectors to be classified as 1 and away from vectors to be classified as 0. This results in a decision boundary that is perpendicular to \mathbf{w} and that properly classifies the input vectors.

There are three conditions that can occur for a single neuron once an input vector \mathbf{p} is presented and the network's response \mathbf{a} is calculated:

	G.S. MANDAL'S	
	MAHARASHTRA INSTITUTE OF TECHNOLOGY, AURABGABAD	
	LAB WORK INSTRUCTION SHEET	
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING		
CLASS: B. TECH PART: 1 (2019-20)	LAB: 506B	SUBJECT: CSE 421 Artificial Neural Network And Deep Learning

CASE 1. If an input vector is presented and the output of the neuron is correct ($\mathbf{a} = \mathbf{t}$ and $\mathbf{e} = \mathbf{t} - \mathbf{a} = 0$), then the weight vector \mathbf{w} is not altered.

CASE 2. If the neuron output is 0 and should have been 1 ($\mathbf{a} = 0$ and $\mathbf{t} = 1$, and $\mathbf{e} = \mathbf{t} - \mathbf{a} = 1$), the input vector \mathbf{p} is added to the weight vector \mathbf{w} . This makes the weight vector point closer to the input vector, increasing the chance that the input vector will be classified as a 1 in the future.

CASE 3. If the neuron output is 1 and should have been 0 ($\mathbf{a} = 1$ and $\mathbf{t} = 0$, and $\mathbf{e} = \mathbf{t} - \mathbf{a} = -1$), the input vector \mathbf{p} is subtracted from the weight vector \mathbf{w} . This makes the weight vector point farther away from the input vector, increasing the chance that the input vector will be classified as a 0 in the future.

The perceptron learning rule can be written more succinctly in terms of the error $\mathbf{e} = \mathbf{t} - \mathbf{a}$ and the change to be made to the weight vector $\Delta\mathbf{w}$:

CASE 1. If $\mathbf{e} = 0$, then make a change $\Delta\mathbf{w}$ equal to 0.

CASE 2. If $\mathbf{e} = 1$, then make a change $\Delta\mathbf{w}$ equal to \mathbf{p}^T .

CASE 3. If $\mathbf{e} = -1$, then make a change $\Delta\mathbf{w}$ equal to $-\mathbf{p}^T$.

Training (train)

If `sim` and `learnp` are used repeatedly to present inputs to a perceptron, and to change the perceptron weights and biases according to the error, the perceptron will eventually find weight and bias values that solve the problem, given that the perceptron *can* solve it. Each traversal through all the training input and target vectors is called a *pass*.

The function `train` carries out such a loop of calculation. In each pass the function `train` proceeds through the specified sequence of inputs, calculating the output, error, and network adjustment for each input vector in the sequence as the inputs are presented.

Note that `train` does not guarantee that the resulting network does its job. You must check the new values of \mathbf{W} and \mathbf{b} by computing the network output for each input vector to see if all targets are reached. If a network does not perform successfully you can train it further by calling `train` again with the new weights and biases for more training passes, or you can analyze the problem to see if it is a suitable problem for the perceptron.

Reference: <https://in.mathworks.com/help/deeplearning/ug/perceptron-neural-networks.html>

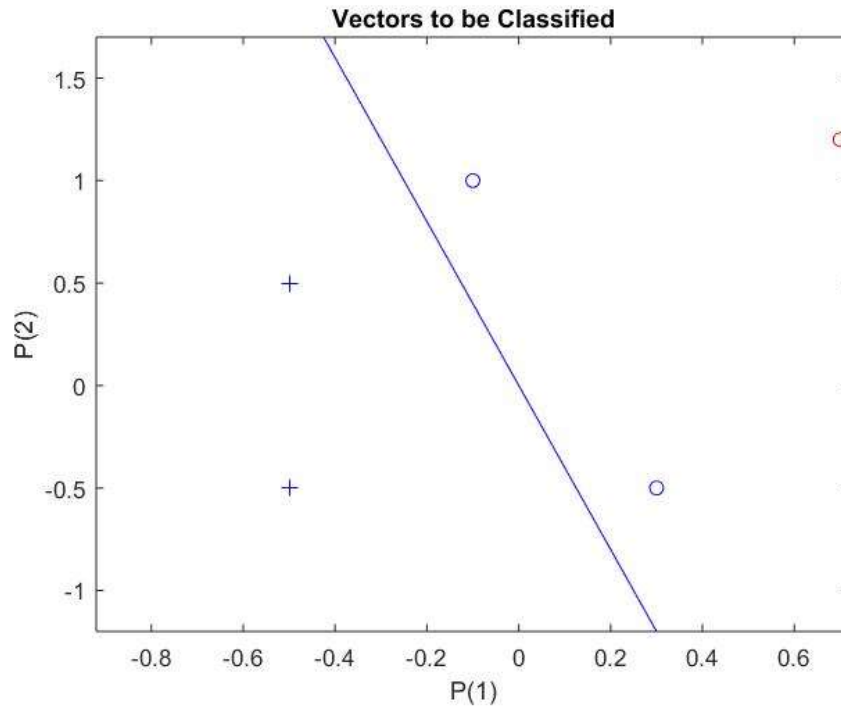


CLASS: B. TECH
PART: 1 (2019-20)

LAB: 506B

SUBJECT: CSE 421 Artificial Neural Network And Deep Learning

Input/Output:



Assessment Questions:

1. What is a Perceptron?
2. What are Weights, bias, activation function and learning rule?
3. How perceptron can be used to classify vectors?



CLASS: B. TECH
PART: 1 (2019-20)

LAB: 506B

SUBJECT: CSE 421 Artificial Neural Network And Deep Learning

Experiment#2

Aim: To write a program to implement AND OR gates using Perceptron.

Objective: To study the design of a neural network that simulates the truth tables of AND & OR Gate

Outcomes: To Simulate AND & OR Logic Gates

Theory: The AND gate is a basic digital logic gate that implements logical conjunction - it behaves according to the truth table.

A HIGH output (1) results only if all the inputs to the AND gate are HIGH (1).


If none or not all inputs to the AND gate are HIGH, a LOW output results. The function can be extended to any number of inputs. The Truth Table for AND gate is as shown,

INPUT		OUTPUT
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

The OR gate is a digital logic gate that implements logical disjunction – it behaves according to the truth table. A HIGH output (1) results if one or both the inputs to the gate are HIGH (1). If neither input is high, a LOW output (0) results. In another sense, the function of OR effectively finds the *maximum* between two binary digits, just as the complementary AND function finds the *minimum*.

INPUT		OUTPUT
A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Reference: https://en.wikipedia.org/wiki/OR_gate

	G.S. MANDAL'S	
	MAHARASHTRA INSTITUTE OF TECHNOLOGY, AURABGABAD	
	LAB WORK INSTRUCTION SHEET	
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING		
CLASS: B. TECH PART: 1 (2019-20)	LAB: 506B	SUBJECT: CSE 421 Artificial Neural Network And Deep Learning

Implementation steps

- 1: type 'nntool' on Matlab command line
- 2: Select 'New'
- 3: Select Data and create input Train Data
- 4: Select Data and create target Data
- 5: Select Data as 'Input' and create Test Data
- 6" Select Network and create Perceptron Network using above data
- 7: Train the Network
- 8: Simulate the Desired Logic Gates

Input/Output:


Input=[0 0 1 1 ;0 1 0 1]

Target=[0 0 0 1] (AND Gate) Target=[0 1 1 1] (OR Gate)

Test=[0 1]

Assessment Questions:

1. What is a formula for activation function?
2. What is role of weights in AND Gate simulation?
3. How perceptron can be used to predict AND gate or OR gate output?

	G.S. MANDAL'S	
	MAHARASHTRA INSTITUTE OF TECHNOLOGY, AURABGABAD	
	LAB WORK INSTRUCTION SHEET	
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING		
CLASS: B. TECH PART: 1 (2019-20)	LAB: 506B	SUBJECT: CSE 421 Artificial Neural Network And Deep Learning

Experiment#3
Aim: To implement Crab Classification using pattern net
Objective: To study the concept of pattern matching and implement for Crab Classification
Outcomes: Implement Pattern Matching
<p>Theory: Artificial neural networks are useful for pattern matching applications. Pattern matching consists of the ability to identify the class of input signals or patterns. Pattern matching ANN are typically trained using supervised learning techniques. One application where artificial neural nets have been applied extensively is optical character recognition (OCR). OCR has been a very successful area of research involving artificial neural networks. An example of a pattern matching neural network is that used by VISA for identifying suspicious transactions and fraudulent purchases. When input symbols do not match an accepted pattern, the system raises a warning flag that indicates a potential problem. The Crab classification can be implemented by using the feature attributes of the crab as a pattern and accordingly the crab could be categorized into a male or a female crab.</p> <p>Crab Classification This example illustrates using a neural network as a classifier to identify the sex of crabs from physical dimensions of the crab. In this example we attempt to build a classifier that can identify the sex of a crab from its physical measurements. Six physical characteristics of a crab are considered: 1)species, 2)fontallip, 3)rear width, 4)length, 5)width and 6)depth. The problem on hand is to identify the sex of a crab given the observed values for each of these 6 physical characteristics.</p> <p>Why Neural Networks? Neural networks have proven themselves as proficient classifiers and are particularly well suited for addressing non-linear problems. Given the non-linear nature of real world phenomena, like crab classification, neural networks is certainly a good candidate for solving the problem. The six physical characteristics will act as inputs to a neural network and the sex of the crab will be target. Given an input, which constitutes the six observed values for the physical characteristics of a crab, the neural network is expected to identify if the crab is male or female.</p> <p>This is achieved by presenting previously recorded inputs to a neural network and then tuning it to produce the desired target outputs. This process is called neural network training.</p> <p>Preparing the Data Data for classification problems are set up for a neural network by organizing the data into two matrices, the input matrix X and the target matrix T. Each ith column of the input matrix will have six elements representing a crabs species, fontallip, rearwidth, length, width and depth. Each corresponding column of the target matrix will have two elements. Female crabs are represented with a one in the first element, male crabs with a one in the second element. (All other elements are zero).</p>



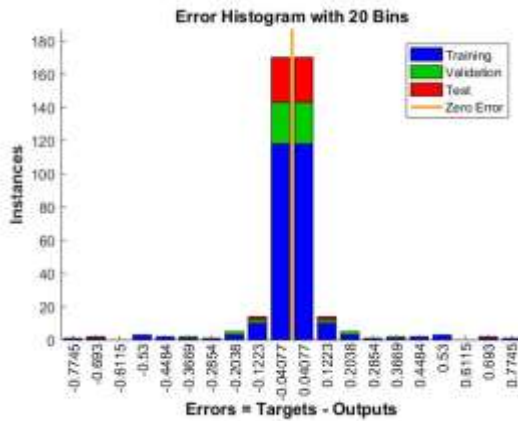
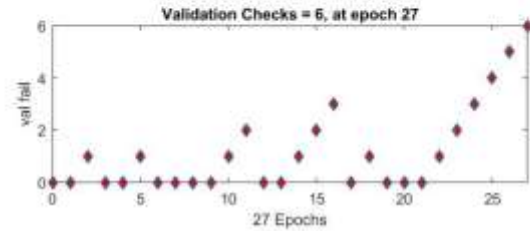
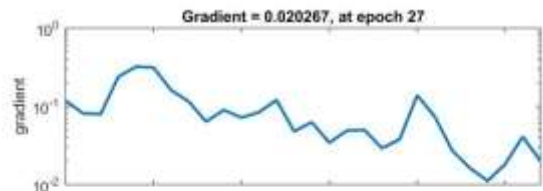
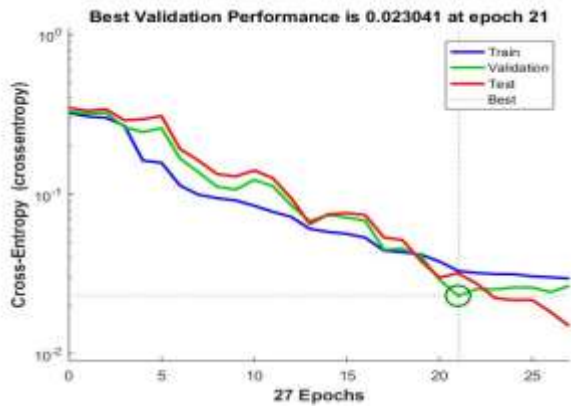
G.S. MANDAL'S
MAHARASHTRA INSTITUTE OF TECHNOLOGY, AURANGABAD
LAB WORK INSTRUCTION SHEET

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CLASS: B. TECH
PART: 1 (2019-20)

LAB: 506B

SUBJECT: CSE 421 Artificial Neural Network And Deep Learning



Training Confusion Matrix

Output Class \ Target Class	1	2	
1	74 92.9%	2 1.4%	97.4%
2	1 0.7%	63 48.0%	64.7%
	36.7%	55.3%	67.6%
	1.2%	1.7%	2.1%

Validation Confusion Matrix

Output Class \ Target Class	1	2	
1	12 40.0%	0 0.0%	100%
2	0 0.0%	18 60.0%	100%
	100%	100%	100%
	0.0%	0.0%	0.0%

Test Confusion Matrix

Output Class \ Target Class	1	2	
1	12 40.0%	0 0.0%	100%
2	1 3.3%	17 58.7%	62.0%
	36.7%	59.0%	62.7%
	0.7%	1.0%	1.7%

All Confusion Matrix

Output Class \ Target Class	1	2	
1	86 60.0%	2 1.0%	96.0%
2	2 1.0%	66 40.0%	68.0%
	68.0%	66.0%	68.0%
	0.0%	0.0%	0.0%



G.S. MANDAL'S
MAHARASHTRA INSTITUTE OF TECHNOLOGY, AURANGABAD

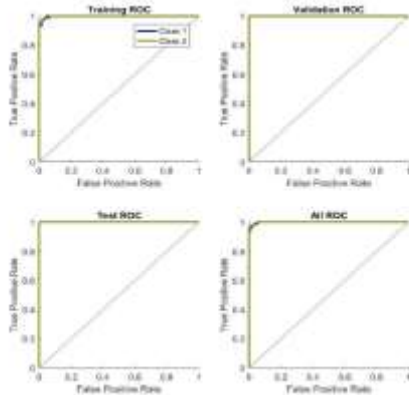
LAB WORK INSTRUCTION SHEET

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CLASS: B. TECH
PART: 1 (2019-20)

LAB: 506B


SUBJECT: CSE 421 Artificial Neural Network And Deep Learning



Reference: Wikipedia and Matlab toolbox Demo

Questions:

1. What is a pattern in Neural Network?
2. How Neural Networks are trained using patterns?
3. How do you set Target for a selected pattern?
4. How do you calculate the accuracy of a pattern net?

	G.S. MANDAL'S	
	MAHARASHTRA INSTITUTE OF TECHNOLOGY, AURABGABAD	
	LAB WORK INSTRUCTION SHEET	
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING		
CLASS: B. TECH PART: 1 (2019-20)	LAB: 506B	SUBJECT: CSE 421 Artificial Neural Network And Deep Learning

Experiment#4

Aim: To write a program to implement Wine Classification using Back propagation.

Objective: To study the concept of Back propagation and implement for Wine Classification
--

Outcomes: To design and implement a Backpropagation application using neural network.
--

Backpropagation, short for "backward propagation of errors," is an algorithm for supervised learning of artificial neural networks using gradient descent. Given an artificial neural network and an error function, the method calculates the gradient of the error function with respect to the neural network's weights. It is a generalization of the delta rule for perceptrons to multilayer feedforward neural networks.

The "backwards" part of the name stems from the fact that calculation of the gradient proceeds backwards through the network, with the gradient of the final layer of weights being calculated first and the gradient of the first layer of weights being calculated last. Partial computations of the gradient from one layer are reused in the computation of the gradient for the previous layer. This backwards flow of the error information allows for efficient computation of the gradient at each layer versus the naive approach of calculating the gradient of each layer separately.

Backpropagation's popularity has experienced a recent resurgence given the widespread adoption of deep neural networks for image recognition and speech recognition. It is considered an efficient algorithm, and modern implementations take advantage of specialized GPUs to further improve performance.

Backpropagation was invented in the 1970s as a general optimization method for performing automatic differentiation of complex nested functions. However, it wasn't until 1986, with the publishing of a paper by Rumelhart, Hinton, and Williams, titled "Learning Representations by Back-Propagating Errors," that the importance of the algorithm was appreciated by the machine learning community at large.

Researchers had long been interested in finding a way to train multilayer artificial neural networks that could automatically discover good "internal representations," i.e. features that make learning easier and more accurate. Features can be thought of as the stereotypical input to a specific node that activates that node (i.e. causes it to output a positive value near 1). Since a node's activation is dependent on its incoming weights and bias, researchers say a node has learned a feature if its weights and bias cause that node to activate when the feature is present in its input.

By the 1980s, hand-engineering features had become the de facto standard in many fields, especially in computer vision, since experts knew from experiments which features (e.g. lines, circles, edges, blobs in computer vision) made learning simpler. However, hand-engineering successful features requires a lot of knowledge and practice. More importantly, since it is not automatic, it is usually very slow.

Backpropagation was one of the first methods able to demonstrate that artificial neural networks could learn good internal representations, i.e. their hidden layers learned nontrivial features. Experts examining multilayer feedforward networks trained using backpropagation actually found that many nodes learned features similar to those designed by human experts and those found by neuroscientists investigating biological neural networks in mammalian brains (e.g. certain nodes learned to detect edges, while others computed Gabor filters). Even more



G.S. MANDAL'S
MAHARASHTRA INSTITUTE OF TECHNOLOGY, AURABGABAD

LAB WORK INSTRUCTION SHEET

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CLASS: B. TECH
PART: 1 (2019-20)

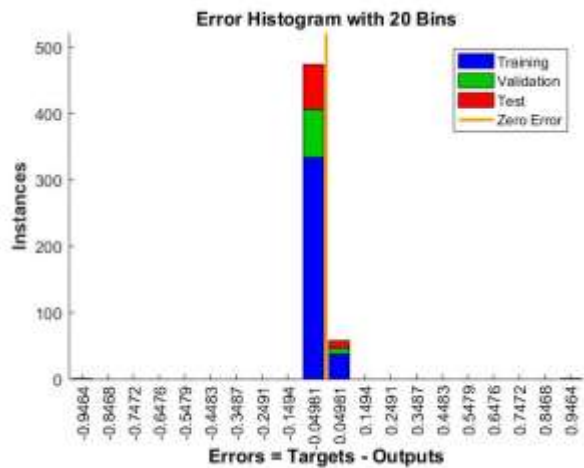
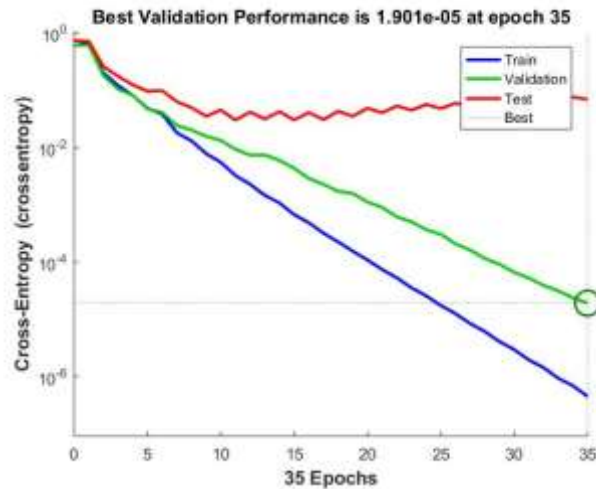
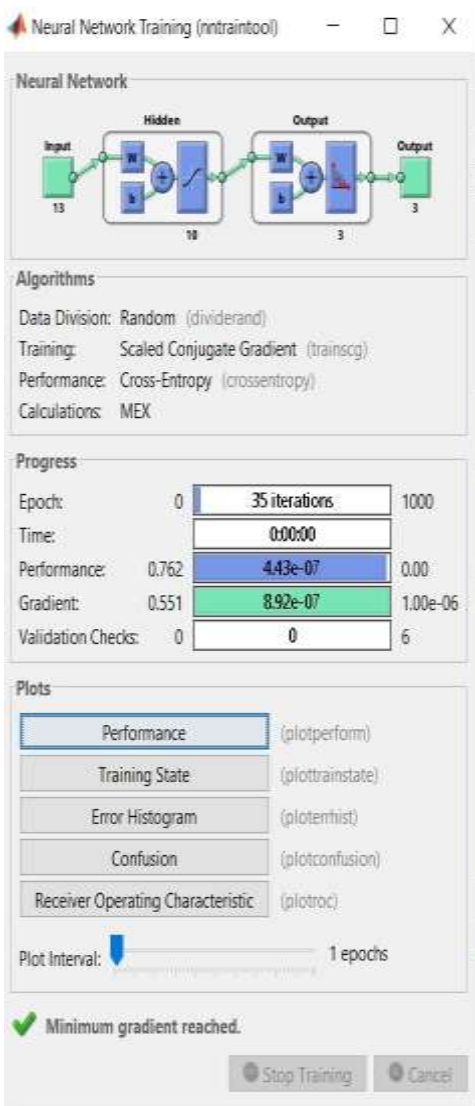
LAB: 506B

SUBJECT: CSE 421 Artificial Neural Network And Deep Learning

importantly, because of the efficiency of the algorithm and the fact that domain experts were no longer required to discover appropriate features, backpropagation allowed artificial neural networks to be applied to a much wider field of problems that were previously off-limits due to time and cost constraints.

Reference: <https://brilliant.org/wiki/backpropagation/>

Input output



Reference: Matlab toolbox Demo



**G.S. MANDAL'S
MAHARASHTRA INSTITUTE OF TECHNOLOGY, AURABGABAD**

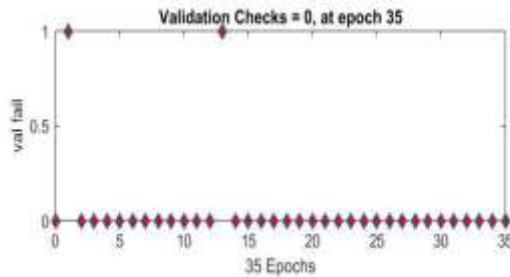
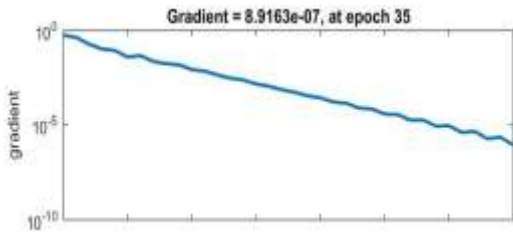
LAB WORK INSTRUCTION SHEET

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CLASS: B. TECH
PART: 1 (2019-20)

LAB: 506B

SUBJECT: CSE 421 Artificial Neural Network And Deep Learning



Training Confusion Matrix

Output Class \ Target Class	1	2	3	
1	45 36.3%	0 0.0%	0 0.0%	100%
2	0 0.0%	49 39.5%	0 0.0%	100%
3	0 0.0%	0 0.0%	30 24.2%	100%
	100%	100%	100%	100%

Validation Confusion Matrix

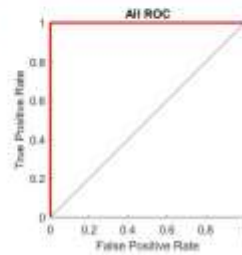
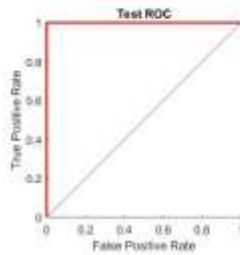
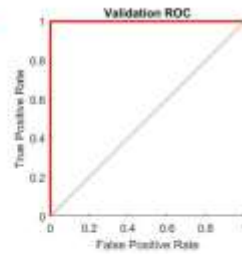
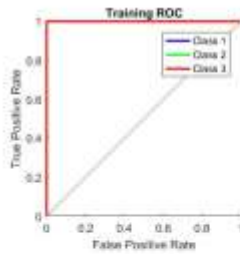
Output Class \ Target Class	1	2	3	
1	5 18.0%	0 0.0%	0 0.0%	100%
2	0 0.0%	11 40.7%	0 0.0%	100%
3	0 0.0%	0 0.0%	11 40.7%	100%
	100%	100%	100%	100%

Test Confusion Matrix

Output Class \ Target Class	1	2	3	
1	9 33.3%	0 0.0%	0 0.0%	100%
2	0 0.0%	10 37.0%	0 0.0%	100%
3	0 0.0%	1 3.7%	7 25.9%	87.3%
	100%	96.8%	100%	96.3%


All Confusion Matrix

Output Class \ Target Class	1	2	3	
1	59 33.1%	0 0.0%	0 0.0%	100%
2	0 0.0%	70 25.3%	0 0.0%	100%
3	0 0.0%	1 0.6%	44 27.0%	88.5%
	100%	96.8%	100%	99.4%



Questions :

1. What is back propagation?
2. How to calculate errors ?
3. Why the errors are back propagated?
4. How to train and test a back propagation neural network?
5. How to calculate Gradient in back propagation neural network?

	G.S. MANDAL'S	
	MAHARASHTRA INSTITUTE OF TECHNOLOGY, AURABGABAD	
	LAB WORK INSTRUCTION SHEET	
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING		
CLASS: B. TECH PART: 1 (2019-20)	LAB: 506B	SUBJECT: CSE 421 Artificial Neural Network And Deep Learning

Experiment#5
Aim: Write a MatLab Script containing four functions Addition, Subtraction, Multiply and Divide functions.
Objective: To study the script of MatLab and use of functions
Outcomes: To design a MatLab script that uses different function calls.
<p>The simplest type of MATLAB® program is called a <i>script</i>. A script is a file that contains multiple sequential lines of MATLAB commands and function calls. You can run a script by typing its name at the command line.</p> <p>Scripts</p> <p>To create a script, use the edit command, edit myspherecg</p> <p>This command opens a blank file named mysphere.m. Enter some code that creates a unit sphere, doubles the radius, and plots the results:</p> <pre>[x,y,z] = sphere; r = 2; surf(x*r,y*r,z*r) axis equal</pre> <p>Next, add code that calculates the surface area and volume of a sphere:</p> <pre>A = 4*pi*r^2; V = (4/3)*pi*r^3;</pre> <p>Whenever you write code, it is a good practice to add comments that describe the code. Comments enable others to understand your code and can refresh your memory when you return to it later. Add comments using the percent (%) symbol.</p> <pre>% Create and plot a sphere with radius r. [x,y,z] = sphere; % Create a unit sphere. r = 2; surf(x*r,y*r,z*r) % Adjust each dimension and plot. axis equal % Use the same scale for each axis.</pre> <p>% Find the surface area and volume.</p> <pre>A = 4*pi*r^2; V = (4/3)*pi*r^3;</pre> <p>Save the file in the current folder. To run the script, type its name at the command line: mysphere</p>



**G.S. MANDAL'S
MAHARASHTRA INSTITUTE OF TECHNOLOGY, AURABGABAD**

LAB WORK INSTRUCTION SHEET

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CLASS: B. TECH
PART: 1 (2019-20)

LAB: 506B

SUBJECT: CSE 421 Artificial Neural Network And Deep Learning

You can also run scripts from the Editor using the **Run** button,  .

Reference: https://in.mathworks.com/help/matlab/learn_matlab/scripts.html

Questions:

1. What is the difference between a Matlab Script and a function?
2. How to debug a Matlab Script?
3. How will you design a graphical user interface using a Matlab Script?



CLASS: B. TECH
 PART: 1 (2019-20)

LAB: 506B

SUBJECT: CSE 421 Artificial Neural Network And Deep Learning

Experiment#6

Aim: Write a program to implement classification of linearly separable Data with a perceptron.

Objective: To study Linearly Separable Data and method to separate the data into classes

Outcomes: To design a classifier that separates the data linearly.

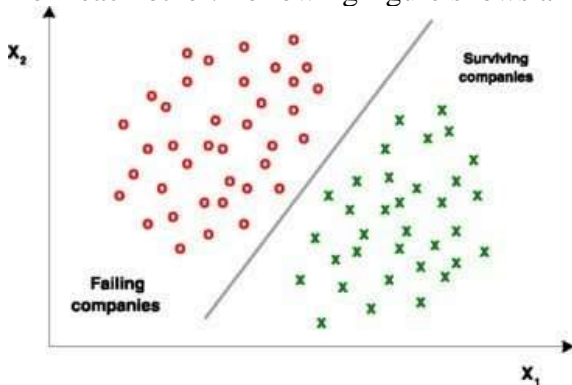
Theory: Two clusters of data, belonging to two classes, are defined in a 2-dimensional input space. Classes are linearly separable. The task is to construct a Perceptron for the classification of data.

In Euclidean geometry, **linear separability** is a property of two sets of points. This is most easily visualized in two dimensions (the Euclidean plane) by thinking of one set of points as being colored blue and the other set of points as being colored red. These two sets are *linearly separable* if there exists at least one line in the plane with all of the blue points on one side of the line and all the red points on the other side.

This idea immediately generalizes to higher-dimensional Euclidean spaces if line is replaced by hyperplane. The problem of determining if a pair of sets is linearly separable and finding a separating hyperplane if they are, arises in several areas. In statistics and machine learning, classifying certain types of data is a problem for which good algorithms exist that are based on this concept.

The idea of linearly separable is easiest to visualize and understand in 2 dimensions. Let the two classes be represented by colors red and green.

A dataset is said to be linearly separable if it is possible to draw a line that can separate the red and green points from each other. Following figure shows an example of linearly separable data:



References:

https://en.wikipedia.org/wiki/Linear_separability

<https://www.commonlounge.com/discussion/6caf49570d9c4d0789afbc544b32cdf>

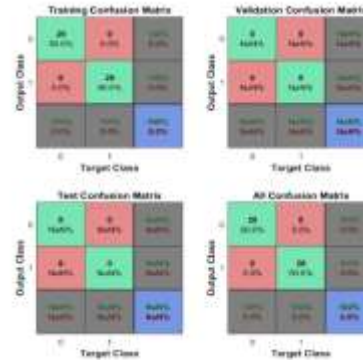
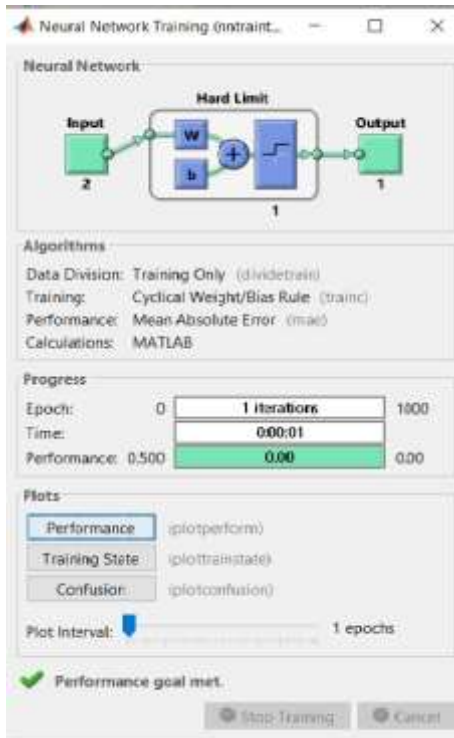


CLASS: B. TECH
PART: 1 (2019-20)

LAB: 506B


SUBJECT: CSE 421 Artificial Neural Network And Deep Learning

Input output



Questions:

1. Which data is said to be linearly separable?
2. Why the linearly non separable data can't be separated linearly?
3. How will you extend the logic used to separate linearly separable data to separate linearly non separable data?
4. What confusion matrix signifies in this example?

	G.S. MANDAL'S	
	MAHARASHTRA INSTITUTE OF TECHNOLOGY, AURANGABAD	
	LAB WORK INSTRUCTION SHEET	
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING		
CLASS: B. TECH PART: 1 (2019-20)	LAB: 506B	SUBJECT: CSE 421 Artificial Neural Network And Deep Learning

Experiment#7

Aim: To study Long Short Term Memory for Time Series Prediction

Objective: To study Long Short Term Memory

Outcomes: To design a neural network that uses long short term memory for Time Series Prediction

Theory: Time series analysis comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data. Time series forecasting is the use of a model to predict future values based on previously observed values. Time series are widely used for non-stationary data, like economic, weather, stock price, and retail sales in this post. Time series forecasting is an important area of machine learning that is often neglected.

It is important because there are so many prediction problems that involve a time component. These problems are neglected because it is this time component that makes time series problems more difficult to handle.

Time does play a role in normal machine learning datasets.

Predictions are made for new data when the actual outcome may not be known until some future date. The future is being predicted, but all prior observations are almost always treated equally. Perhaps with some very minor temporal dynamics to overcome the idea of “*concept drift*” such as only using the last year of observations rather than all data available.

A time series dataset is different.

Time series adds an explicit order dependence between observations: a time dimension.

Time Series Forecasting

Making predictions about the future is called extrapolation in the classical statistical handling of time series data.

More modern fields focus on the topic and refer to it as time series forecasting.

Forecasting involves taking models fit on historical data and using them to predict future observations.

Descriptive models can borrow for the future (i.e. to smooth or remove noise), they only seek to best describe the data.

An important distinction in forecasting is that the future is completely unavailable and must only be estimated from what has already happened.

The purpose of time series analysis is generally twofold: to understand or model the stochastic mechanisms that gives rise to an observed series and to predict or forecast the future values of a series based on the history of that series

We use the data set `chickenpox_dataset`. The example trains an LSTM network to forecast the number of chickenpox cases given the number of cases in previous months. Following are the steps to be followed to implement.


1. Load Sequence Data

Load the example data. `chickenpox_dataset` contains a single time series, with time steps corresponding to months and values corresponding to the number of cases. The output is a cell array, where each element is a single time step. Reshape the data to be a row vector.

Partition the training and test data. Train on the first 90% of the sequence and test on the last 10%.

2. Standardize Data

For a better fit and to prevent the training from diverging, standardize the training data to have zero mean and unit variance. At prediction time, you must standardize the test data using the same parameters as the training data.

	G.S. MANDAL'S	
	MAHARASHTRA INSTITUTE OF TECHNOLOGY, AURABGABAD	
	LAB WORK INSTRUCTION SHEET	
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING		
CLASS: B. TECH PART: 1 (2019-20)	LAB: 506B	SUBJECT: CSE 421 Artificial Neural Network And Deep Learning

3. Prepare Predictors and Responses

To forecast the values of future time steps of a sequence, specify the responses to be the training sequences with values shifted by one time step. That is, at each time step of the input sequence, the LSTM network learns to predict the value of the next time step. The predictors are the training sequences without the final time step.

4. Define LSTM Network Architecture

Create an LSTM regression network. Specify the LSTM layer to have 200 hidden units.

Specify the training options. Set the solver to 'adam' and train for 250 epochs. To prevent the gradients from exploding, set the gradient threshold to 1. Specify the initial learn rate 0.005, and drop the learn rate after 125 epochs by multiplying by a factor of 0.2.

5. Train LSTM Network

Train the LSTM network with the specified training options by using trainNetwork.

6. Forecast Future Time Steps

To forecast the values of multiple time steps in the future, use the predictAndUpdateState function to predict time steps one at a time and update the network state at each prediction. For each prediction, use the previous prediction as input to the function. Standardize the test data using the same parameters as the training data.

To initialize the network state, first predict on the training data XTrain. Next, make the first prediction using the last time step of the training response YTrain(end). Loop over the remaining predictions and input the previous prediction to predictAndUpdateState. For large collections of data, long sequences, or large networks, predictions on the GPU are usually faster to compute than predictions on the CPU. Otherwise, predictions on the CPU are usually faster to compute. For single time step predictions, use the CPU.

To use the CPU for prediction, set the 'ExecutionEnvironment' option of predictAndUpdateState to 'cpu'. Unstandardize the predictions using the parameters calculated earlier. The training progress plot reports the root-mean-square error (RMSE) calculated from the standardized data. Calculate the RMSE from the unstandardized predictions. Plot the training time series with the forecasted values.

References:

<https://machinelearningmastery.com/time-series-forecasting/>

<https://in.mathworks.com/help/deeplearning/examples/time-series-forecasting-using-deep-learning.html>



G.S. MANDAL'S
MAHARASHTRA INSTITUTE OF TECHNOLOGY, AURANGABAD

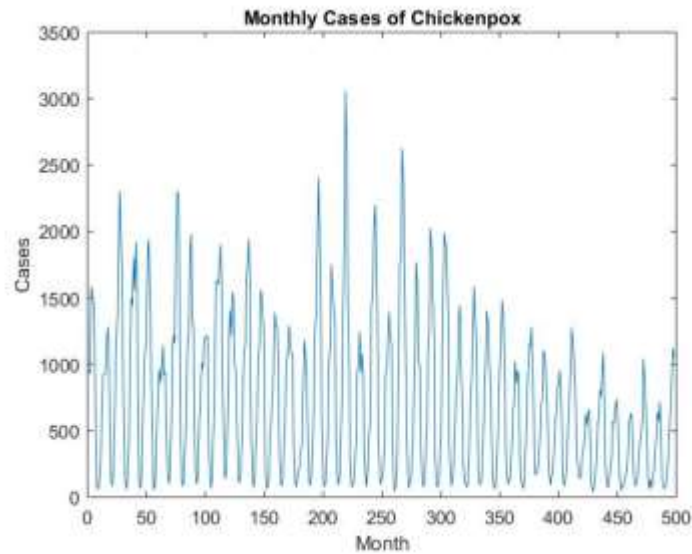
LAB WORK INSTRUCTION SHEET

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CLASS: B. TECH
PART: 1 (2019-20)


LAB: 506B

SUBJECT: CSE 421 Artificial Neural Network And Deep Learning




Questions:

1. What is the working principle of LSTM?
2. What are the applications of LSTM?
3. What is the difference between Architecture of LSTM and FFBBNN?

	G.S. MANDAL'S	
	MAHARASHTRA INSTITUTE OF TECHNOLOGY, AURABGABAD	
	LAB WORK INSTRUCTION SHEET	
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING		
CLASS: B. TECH PART: 1 (2019-20)	LAB: 506B	SUBJECT: CSE 421 Artificial Neural Network And Deep Learning

Experiment#8
Aim: To study Convolutional Neural Network and Recurrent Neural Network
Objective: To study the construction and working of Convolutional Neural Network and Recurrent Neural Network
Outcomes: To design and implement CNN and RNN for given applications.
Theory: <ol style="list-style-type: none"> 1. Convolutional Neural Network 2. Recurrent Neural Network <p>Students are expected to study the text books and other web references provided in the syllabus and write in their own words about Construction, Basic working principles, network design, learning algorithms, and applications of Convolutional Neural Network and Recurrent Neural Network.</p> <p>Students are also instructed to generate plagiarism report of the write-ups they would submit.</p>

Experiment#9
Aim: To study ImageNet, GoogleNet, ResNet convolutional Neural Networks
Objective: To study the construction and working of Different Convolutional Neural Network examples
Outcomes: To learn data science and design and implement various convolutional Neural Networks
Theory: <ol style="list-style-type: none"> 1. ImageNet 2. GoogleNet 3. ResNet <p>Students are expected to study the text books and other web references provided in the syllabus and write in their own words about Construction, Basic working principles, network design, learning algorithms, and applications of different types of Convolutional Neural Networks. Students are expected to compare these types of CNN and write their applications in detail.</p> <p>Students are also instructed to generate plagiarism report of the write-ups they would submit.</p>

	G.S. MANDAL'S	
	MAHARASHTRA INSTITUTE OF TECHNOLOGY, AURABGABAD	
	LAB WORK INSTRUCTION SHEET	
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING		
CLASS: B. TECH PART: 1 (2019-20)	LAB: 506B	SUBJECT: CSE 421 Artificial Neural Network And Deep Learning

Experiment#10
Aim: To study the use of Long Short Term Memory / Gated Recurrent Units to predict the stock prices based on historic data
Objective: To study the construction and working of LSTM and GRU
Outcomes: To learn data science and design and implement various LSTM/GRU for Predictions.
Theory: <ol style="list-style-type: none"> 1. Long Short Term Memory 2. Gated Recurrent Units <p>Students are expected to study the text books and other web references provided in the syllabus and write in their own words about Construction, Basic working principles, network design, learning algorithms, and applications of Long Short Term Memory and Gated Recurrent Units. Students should write how LSTM / GRU can be used to predict the stock prices based on historic data.</p> <p>Students are also instructed to generate plagiarism report of the write-ups they would submit.</p>

-----X-----