

SUBJECT: DEEP LEARNING AND PYTHON BASICS (DLPB)

**CLASS XII
SEMESTER III**

THEORY

FULL MARKS – 20

(MCQ Type Question)

UNIT	Topic	No. of periods assigned	Marks
Unit 1	Introduction to Neural Networks	9	4
Unit 2	Perceptron's & Multi-layer Perceptron's (MLPs)	18	8
Unit 3	Deep Learning Fundamentals	18	8

DETAIL SYLLABUS

UNIT	Topic / Sub Topic	No. of periods assigned
Unit 1	1.1 Basic concept of neurons and neural networks. 1.2 Historical background and development of neural networks. 1.3 Applications of neural networks in various fields.	9
Unit 2	2.1 Definition and structure of perceptron's, Perceptron learning algorithm, Limitations of single-layer perceptron's. 2.2 Structure and architecture of MLPs, Activation functions (e.g., sigmoid, ReLU). 2.3 Backpropagation algorithm for training MLPs.	18
Unit 3	3.1 Definition and significance of deep learning. 3.2 Introduction to deep neural networks. 3.3 Advantages of deep learning over shallow models. 3.4 Benefits and challenges of Deep Learning	18

**CLASS XII
SEMESTER IV**

THEORY

FULL MARKS – 30

(SAQ AND LAQ Type Question)

UNIT	Topic	No. of periods assigned	Marks
Unit 4	Introduction to Python	18	8
Unit 5	Convolutional Neural Networks (CNNs)	15	8
Unit 6	Training Deep Learning Models	15	10
Unit 7	Applications of Deep Learning	15	4

DETAIL SYLLABUS

UNIT	Topic / Sub Topic	No. of periods assigned
Unit 4	4.1 Overview of Python programming language. 4.2 Installation of Python and Integrated Development Environments (IDEs) like Anaconda, Jupyter Notebook, or Google Colab. 4.3 Introduction to Libraries – Introduction to NumPy for numerical computing and arrays, Basic operations with NumPy arrays, Introduction to Pandas for data manipulation and analysis, Basic operations with Pandas Data Frames. 4.4 Deep Learning Frameworks – Overview of popular deep learning frameworks: TensorFlow and Py Torch, Installation and setup of TensorFlow and PyTorch, Building simple neural networks using TensorFlow/Keras or PyTorch.	18
Unit 5	5.1 Architecture of CNNs and their components (e.g., convolutional layers, pooling layers). 5.2 Applications of CNNs in image recognition and computer vision tasks. 5.3 Recurrent Neural Networks (RNNs) – Structure and working principle of RNNs, Applications of RNNs in sequential data processing (e.g., natural language processing, time series prediction).	15

UNIT	Topic / Sub Topic	No. of periods assigned
Unit 6	6.1 Gradient descent optimization and variants (e.g., stochastic gradient descent, mini- batch gradient descent). 6.2 Regularization techniques (e.g., dropout, L2 regularization), Hyperparameter tuning and model Evaluation (Confusion Matrix, Accuracy, Precision, F1-Score).	15
Unit 7	7.1 Data classification, object detection, Data Segmentation. 7.2 Healthcare applications (e.g., medical imaging analysis, disease diagnosis). 7.3 Smart Agriculture (Plant health monitoring), Horticulture monitoring with deep learning-Quality Gradation of fruits, Classification of farm produces	15

**PRACTICAL
CLASS XII**

FULL MARKS – 40

NO OF PERIODS ASSIGNED – 84

DETAIL SYLLABUS

Sl. No.	Practical	Classes
1	Introduction to Deep Learning	10
	Overview of artificial intelligence, machine learning, and deep learning. Introduction to neural networks and their applications. Understanding the role of deep learning in various domains. Basic concepts of supervised learning, unsupervised learning, and reinforcement learning.	
2	Python Basics for Deep Learning	16
	Introduction to Python programming language. Basic data types, operators, and control structures in Python. Introduction to NumPy for numerical computing. Hands-on exercises to implement basic operations using NumPy.	
3	Introduction to Artificial Neural Networks (ANNs)	16
	Implement a basic neural network from scratch using Python and NumPy. Start with a simple feed forward neural network with one hidden layer. Students can experiment with different activation functions (like sigmoid, ReLU) and loss functions (like mean squared error, cross-entropy).	
4	Image Classification	16
	Use a dataset like MNIST or Fashion-MNIST for a hands-on experience in image classification. Guide students to build a convolutional neural network (CNN) using frameworks like TensorFlow or PyTorch. They can experiment with different architectures, optimization algorithms (SGD, Adam), and hyper parameters to observe their impact on model performance.	
5	Text Classification	16

Sl. No.	Practical	Classes
	Introduce natural language processing (NLP) concepts by working on text classification tasks. Utilize datasets like sentiment analysis datasets or news categorization datasets. Students can build recurrent neural networks (RNNs) or long short-term memory networks (LSTMs) for this task. Transfer Learning: Teach the concept of transfer learning by using pre-trained models like VGG, ResNet, or BERT. Students can fine-tune these models on a smaller dataset relevant to their interest or project.	
6	Generative Adversarial Networks (GANs)	10
	Introduce the concept of GANs for generating realistic images. Start with simple examples like generating handwritten digits. Students can then move on to more complex tasks like generating human faces.	

**PROJECT
CLASS XII**

FULL MARKS – 10

NO OF PERIODS ASSIGNED – 60

DETAIL SYLLABUS

Name of the Project: Smart Farming- Crop Disease Detection and Yield Prediction using Machine Learning		
Objective – The objective of this project is to develop a smart farming system that utilizes machine learning techniques for crop disease detection and yield prediction. By leveraging sensor data and image analysis, the system will help farmers monitor crop health, detect diseases early, and make informed decisions to optimize yield.		
Sl. No.	Tasks	Classes
1	Data Collection	8
	- Research and identify datasets containing historical crop health data, weather information, soil conditions, and images of diseased crops. - Acquire permission or access to the datasets. - Clean and preprocess the data to ensure quality and consistency.	
2	Exploratory Data Analysis (EDA)	8
	- Perform descriptive statistics to understand the distribution of data. - Visualize the data to identify patterns, trends, and correlations. - Explore relationships between weather conditions, soil characteristics, and crop health.	
3	Feature Engineering	8
	- Extract relevant features from the dataset, including weather variables, soil properties, and crop characteristics. - Engineer new features if necessary, such as derived metrics or transformations. - Select appropriate features for training the machine learning models.	
4	Crop Disease Detection Model	12
	- Research and experiment with image classification algorithms suitable for detecting crop diseases from images. - Preprocess and augment the image data to enhance model performance. - Train and evaluate the image classification model using techniques like Convolutional Neural Networks (CNNs). - Validate the model's performance on a separate test dataset.	

5	Yield Prediction Model	8
	- Research and experiment with regression algorithms for yield prediction, considering factors such as weather conditions, soil quality, and crop health. - Split the dataset into training and testing sets. - Train and evaluate the yield prediction model using techniques like Random Forest Regression or Gradient Boosting Regression. - Fine-tune the model hyper-parameters to optimize performance.	
6	Integration and Deployment	8
	- Integrate the crop disease detection and yield prediction models into a unified smart farming system. - Develop an intuitive user interface for farmers to input data, view crop health status, and receive predictions. - Deploy the smart farming system on a cloud platform or a local server.	
7	Documentation and Presentation	8
	- Document the entire project, including data sources, methodologies, and implementation details. - Create a comprehensive report summarizing the findings, challenges, and recommendations. - Prepare a presentation to showcase the project outcomes, including demonstrations of the smart farming system.	