3rd SEMESTER: M.Sc. (IT) Batch 2023

Course Code:
Credits: 4 (L- 3, T- 0, P-1)
Contact Hrs: 75 (Theory: 45, Tutorial: 30)

Title: Deep Learning [DSE 2] Max. Marks: 100
Theory External: 60; Min Marks: 24
Theory Internal (Continuous Assessment): 15 Marks, Min Marks: 06
Practical Experimental Basis= 15, Min. Marks: 06
Practical Experimental (Continuous assessment) =10, Min. Marks: 04

Objectives:
The objective of this course is to provide an overview of deep learning concepts, including its evolution from traditional machine learning, discuss the various stages involved in a deep learning project, and enable students to create and train different types of neural networks, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), generative adversarial networks (GANs), and auto-encoders.

Learning Outcomes:
After successful completion of the course, the students should be able to:

1. Articulate the fundamental principles of deep learning, including the differences between machine learning and deep learning, and the various stages involved in a deep learning project.
2. Build and train various types of neural networks, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), generative adversarial networks (GANs), and auto-encoders.
3. Implement regularization methods such as dropout and L2 regularization, and use optimization algorithms like SGD, Adam, and RMSprop to improve model performance.
4. Implement and train neural networks using popular deep learning frameworks such as TensorFlow, Keras, or PyTorch.

Syllabus

Unit I
Introduction to Deep Learning: History and Evolution of Machine Learning to Deep Learning, Stages in ML/DL project, Applications of Deep Learning,

Introduction to Neural Networks: ANN, Perceptron, Activation Functions (ReLU, Sigmoid, Tanh), Loss Functions (MSE, Cross-Entropy) and Gradient Descent and Backpropagation; Feedforward Neural Networks; Regularization Techniques (Dropout, L2 Regularization); Optimization Algorithms: SGD, Adam, RMSprop.

Unit II

CONVOLUTIONAL NEURAL NETWORKS (CNNS): Introduction to CNNs and their applications, Basic structure of Convolutional Networks, Types of Pooling layers, Different CNN architectures: LeNet, AlexNet, VGG, PlacesNet; Training a CNNs: weights initialization, batch normalization, hyperparameter optimization; Understanding and visualizing CNNs.

RECURRENT NEURAL NETWORKS (RNNS): Introduction to RNNs and their applications, Sequence modelling using RNNs, Backpropagation through time(BPTT), Vanishing and Exploding Gradients, Long Short Term Memory (LSTM), Bidirectional LSTMs, Bidirectional RNNs, Gated RNN Architecture.

Unit III
AUTO-ENCODERS: Auto-encoders, Architecture, and components of auto-encoders (encoder and decoder), Training an auto-encoder for data compression and reconstruction.

GENERATIVE ADVERSARIAL NETWORKS (GANS): Generative models, Concepts and principles of GANs, Architecture of GANs (generator and discriminator networks), Comparison between discriminative and generative models, Generative Adversarial Networks (GANs), Relationship between Auto-encoders and GANs.

Deep learning frameworks and libraries (TensorFlow/Keras, PyTorch).

Lab Course (Practical Work) **(1 Credit:30 hrs)**

Unit IV
1. Implement a Single layer perceptron with Python using Numpy.
2. Implement a basic Multi-Layer Perceptron TensorFlow/Keras or PyTorch.
3. Experiment with different activation functions (ReLU, Sigmoid, Tanh) and observe their impact on performance.
4. Implement Mean Squared Error (MSE) and Cross-Entropy loss functions.
5. Implement dropout and L2 regularization in a neural network model.
6. Experiment with different optimization algorithms (SGD, Adam, RMSprop).
7. Build a basic CNN from scratch for image classification, apply weight initialization and batch normalization techniques, and Perform hyperparameter optimization to enhance model performance.
8. Compare different CNN architectures: LeNet, AlexNet, VGG, and PlacesNet.
9. Implement a basic RNN for sequence prediction tasks.
10. Implement Long Short-Term Memory (LSTM) networks and Bidirectional LSTMs and compare their performance.
11. Design and implement a basic auto-encoder with encoder and decoder components to perform data compression and reconstruction.
12. Build a basic GAN with a generator and discriminator network.

13. Create a simple feedforward neural network to classify handwritten digits from the MNIST dataset.
14. Implement “Boston housing price prediction” problem using Deep Neural network.

Recommended Books

1. CharuC.Aggarwal. Neural Networks and Deep Learning, Springer International
2. Publishing, 2018.
3. Ian Goodfellow and Yoshua Bengio Aaron Courville. Deep Learning, An MIT Press Book
4. Michael Nielson. Neural Networks and Deep Learning, Determination Press,2015
5. Francois Chollet. Deep Learning with Python, Manning publications 2018