|  |  |
| --- | --- |
|  | **PIR MEHR ALI SHAH ARID AGRICULTURE UNIVERSITY**  **University Institute of Information Technology** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Artificial Neural Networks (CS-405)** | | | | |
| **Credit Hours:** | 3(2-3) | **Prerequisites:** | Programming in AI | |
| **Teacher:** | **Dr. Muhammad Aqib** | | | |
| **Course Learning Outcomes (CLOs)** | | | | |
| At the end of course the students will be able to: | | | **Domain** | **BT Level\*** |
| 1. Understand the fundamentals of neural networks in AI | | | C | 2 |
| 1. Explain how simple ANNs can be designed | | | C | 2 |
| 1. Understand the different layers and their operation as well as back propagation | | | C | 3 |
| 1. Apply ANNs for the classification problems | | | C | 3 |
| 1. Given a dataset, be able to design a neural network appropriate for the problem | | | C | 4 |
| \*BT- Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A=Affective domain | | | | |

|  |
| --- |
| **Course Contents:** |
| Introduction and history of neural networks, basic architecture of neural networks, perceptron and introduction to regression and classification problems. Basics of machine learning and its theory – training, test sets, and evaluation. Types of learning -supervised, unsupervised. Introduction of deep learning, shallow learning vs deep learning, architecture of multi-layer perceptrons, parameters of network, activation functions, feed forward networks and back propagation. Overview of underfitting and overfitting and optimization algorithms. Artificial Neural Networks (ANNs) for regression and classification problems. Performance metrics for evaluation of ANNs. Overview of CNNs and RNNs. Explanation and working of LeNet. |
| **Course Objective:** |
| (a) Understand the history and significance of neural networks with grasp of the basic structure of neural networks and their components. (b) Define deep learning and its contrast with shallow learning and explore multi-layer perceptron architecture and the impact of hidden layers. (c) Study network parameters, activation functions, and their effects. Learn about feedforward networks, backpropagation, and optimization techniques. (d) Apply artificial neural networks to regression and classification tasks (e) Develop practical skills in implementing and evaluating neural networks for real-world problems. |
| **Teaching Methodology:** |
| Lectures, Assignments, labs, Projects, Presentations, etc. Major component of the course should be covered using conventional lectures. |
| **Courses Assessment:** |
| Exams, Assignments, Quizzes, Project, Presentations. Course will be assessed using a combination of written examinations and project(s). |
| **Reference Materials:** |
| 1. An Introduction to Neural Networks, James A Anderson 2. Neural Network Design, 2nd Edition, Martin T. Hagan 3. Neural Networks from Scratch in Python, Harrison Kinsley & Daniel Kukieła, 2020 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Week/Lecture #** | | **Theory** | **Practical** |
| Week 1 | Lecture-I | Introduction to Artificial Neural Networks (ANNs) and their history  Basic Sturcture of Neural Networks and brief overview of layers  Artificial Neural Networks: Weights, biases and their impact | Setting up Python and Jupyter notebooks |
| Python Introduction,  Loops, conditions, lists etc |
| Lecture-II | Artificial Neural Networks: Activation function, Overfitting, Generalization, Loss | Exploring and reading about scikit-learn library |
| Week 2 | Lecture-I | Single Neuron With 3 and 4 inputs  Layer of neurons  Tensors, arrays and vectors  Dot product and vector additions | Implementing single neuron with 3 inputs |
| Data manipulation in tensors, application of dot product etc. |
| Lecture-II | Neurons implementation with Numpy Matrix Product  Matrix Transposition | Implementation of layers of neuron |
| Week 3 | Lecture-I | Adding Layers to Network Training Data Data Visualization Data Generation | Working with multiple layers, Addition of hidden layers |
| Implentation of Tensors, arrays and vectors |
| Lecture-II | Working with Dense Layer Dead Neurons | Neurons implementation with Numpy |
| Week 4 | Lecture-I | Activation Functions (Linear Activation - Sigmoid Activation)Rectified Linear Activation | Coding simple activation functions e.g. step function. |
| Use of activation functions including step, linear, sigmoid |
| Lecture-II | Softmax Activation | Coding a simple script for applying ReLU |
| Week 5 | Lecture-I | What is Loss in NN?Categorical Cross-Entropy Loss |  |
| Training Data - Data Visualization - Data Generation |
| Lecture-II | Accuracy Calculation | Working with Dense Layer |
| Week 6 | Lecture-I | What is NN OptimizationOptimization algorithms: SGD and Adam |  |
| Implementation of Linear Activation Functions |
| Lecture-II | Experimentation of NN Optimization | Implementation of Sigmoid Activation Functions |
| Week 7 | Lecture-I | What is derivative Partial Derivative (Sum – Multiplication)  Gradient  Optimizers | Calculation of derivatives |
| Application of partial derivatives, chain rules etc. |
| Lecture-II | Type of Optimizer Granient Descent | Implementation of Gradient Descent |
| Week 8 | Lecture-I | Application of ReLU to a Neuron NetworkBackpropagartion Working of BackPropagation | Application of Relu in hidden layers |
| Lecture-II | Backpropagartion (Static -Recurrent) |
| **Midterm Exam** | | | |
| Week 10 | Lecture-I | Introduction to performance metrics.  Accuracy Calculation | Accuracy calculation |
| Use of python libraries for calculation of error and plotting |
| Lecture-II | Confusion Matrix, Precision, Recall, F1-score, IoU etc | Use of built-in functions for Precision recall etc. |
| Week 11 | Lecture-I | Introduction to major frameworks for ANNs – PyTorch and Tensorflow  A brief overview of Keras framework | Introduction to TensroFlow |
| Installation and working with Colab |
| Lecture-II | Designing of ANNs in Keras framework | Implementation of MNIST dataset for training using TensorFlow |
| Week 12 | Lecture-I | Introduction to YOLO,  YOLO versions  Images annotation  Annotation tools |  |
| Working with YOLO |
| Lecture-II | Working on Roboflow  Model Training, Validation, Prediction, and Tracking using YOLOv8 | Data labeling and training |
| Week 13 | Lecture-I | Concept of overfitting  How to | Tensorflow/YOLO example of overfitting |
| TensorFlow/YOLO example of underfitting |
| Lecture-II | Concept of underfitting | Working with underfit models and to improve their accuracy and prediction score |
| Week 14 | Lecture-I | Introduction to Deep Learning – Convolutional Neural Networks (CNNs) and applications of CNNs  A brief overview of Recurrent Neural Networks (RNNs) and applications | Implementation of CNN model |
| Lecture-II | CNNs: convolutions, activations, pooling, fully connect layer |
| Week 15 | Lecture-I | Case study: What is LeNet? Case study: In-depth explanation of LeNet | Implementation of Alexnet, ZFNet, VGGNet |
| Lecture-II | Overview of architectures – Alexnet, ZFNet, VGGNet, GoogleNet, ResNet, YOLO | Implementaion of GoogleNet, ResNet |
| Week 16 | Lecture-I | Project demonstration |  |
|  |
| Lecture-II | Project demonstration |  |
| **Finalterm Exam** | | | |