|  |  |  |
| --- | --- | --- |
| IQRA University (IU) | | |
| Faculty of Engineering Sciences and Technology (FEST) | | |
| Computer Science Department (CS) | | |
| Course Code | Course Name | Credit Hr |
| AIC 322- L | ARTIFICIAL NEURAL NETWORK | 3+1 |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Basic Information | | | |
| Instructor | Syed Muhammad Daniyal | Designation | Junior Lecturer |
| Prerequisite(s) | CMC112 | Semester | Fall 2024 |
| Email | Syed.daniyal@iqra.edu.pk | Phone | NA |
| Consulting Hours | 8:30 to 4:30 | Office Location | 601 |

|  |
| --- |
| 1. **Course Objective(s)** |
| This course aims to provide a comprehensive understanding of neural networks and deep learning, covering their history, architecture, and practical applications. Students will learn to design, implement, and analyze neural networks, from perceptrons to advanced architectures like CNNs, RNNs, GANs, and ResNet. Key topics include optimization techniques, machine learning principles, unsupervised learning, generative models, and overfitting mitigation strategies. By the end, students will be equipped to apply neural networks effectively to solve complex, real-world problems. |

|  |
| --- |
| 1. **Course Contents** |
| Introduction and history of neural networks, Basic architecture of neural networks, Perceptron and Adaline (Minimum Error Learning) for classification. Basics of deep learning, learning networks, Shallow vs. Deep learning etc.; Machine learning theory, training and test sets, evaluation, etc. Selected topics from: Gradient descent (Delta) rule, Hebbian, Neo-Hebbian and Differential Hebbian Learning, Drive Reinforcement Theory, Kohonen Self Organizing Maps, Associative memory, Bi-directional associative memory (BAM), Energy surfaces, The Boltzmann machines, Backpropagation Networks, Feedforward Networks; Theory of Generalization; Multi-layer perceptrons, error backpropagation; Deep convolutional networks, Computational complexity of feed-forward and deep convolutional neural networks; Unsupervised deep learning including auto-encoders; Deep belief networks; Restricted Boltzman Machines; Deep Recurrent Neural Networks (BPTT, LSTM, etc.); GPU programming for deep learning CuDNN; Generative adversarial networks (GANs); Sparse coding and auto-encoders; Data augmentation, elastic distortions, data normalization; Mitigating overfitting with dropout, batch normalization, drop connect; Novel architectures, ResNet, GoogleNet, etc |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. **Course Learning Outcomes** | | | | | | |
| **CLOs** | **CLO Statement** | **BT Level** | **Mapping** | | | **% Weight** |
| **GAs** | **ACM KA** | **SGDs** |
| CLO1 | **Design** and develop innovative artificial neural network architectures and learning algorithms using modern tools to solve complex problems across various domains. | C5 (Evaluation) | GA5 | #9  IS | 4 | 65% |
| CLO2 | **Discuss** the significance of artificial neural networks in solving real-world problems and actively assist in collaborative learning environments by conforming to ethical guidelines and fostering an inclusive perspective towards diverse applications of ANN technology. | A2  (Responds to phenomena) | GA7 | 35% |
| ***Note: On successful completion of course GA 1 (Academic Education) will automatically attain.*** | | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. **Course Textbook / Reference Books and Supplementary Reading Material** | | | |
| **S No** | **Book Title** | **Author(s)** | **Edition/ publication year/publisher** |
|  | Deep Learning | Ian Goodfellow, Yoshua Bengio, and Aaron Courville | 1st Edition 2016  Publisher: MIT Press |
|  | Generative Deep Learning | David Foster | 2nd Edition, June 2024  Publisher: O'Reilly Media |
|  | Neural Networks from Scratch in Python | Harrison Kinsley & Daniel Kukieła | 1st Edition, 2020  Publisher: Harrison Kinsley |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. **CLO Outcome Based Assessment (OBA) (Tentative)** | | | | | | |
| **Assessment Tool** | | **CLO Mapped** | **CLO Marks** | **% Weight** | **Total Marks** | **Assessment Date** |
| **Lab Manual**  **10** |  | ***CLO 1, 2*** | 10 | 100% | 10 | TBD |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| **Total Quizzes %** | | | **100%** | 10 |  |
|  | | | | | | |
| **Assignments/ Lab Task**  **15** | Assignment #1 | *CLO1* | 10 | 25% | 5 |  |
| Assignment #2 | *CLO2* | 10 | 25% | 5 |  |
| Assignment #3 | *CLO1* | 10 | 50% | 10 |  |
|  |  | | | **100%** | 20 |  |
|  | | | | | | |
| **Midterm**  **25** | Midterm Q1 | *CLO1* | 15 | **60%** | **15** |  |
| Midterm Q2 | *CLO2* | **10** | **40%** | **10** |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | **Total Midterm %** | | | **100%** | **25** |  |
|  | | | | | | |
| **Project/OEL**  **10** | OEL | *CLO2* | 10 |  |  |  |
|  | **Total Project /CCP %** | | | **100%** | **10** |  |
|  | | | | | | |
| **Final Exam**  **40** | Final Exam Q1 | *CLO1* | 20 | **50%** |  |  |
| Final Exam Q2 | *CLO2* | 20 | **50%** |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | **Total Final Exam %** | | | **100%** | **40** |  |
| **100** | **Total Marls** | | | | **100** |  |
| ***Note: Please make sure every CLO must be assessed at least 3 time.*** | | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. **Weekly Plan** | | | | |
| **Week**  **No** | **Lab No** | **Lab Description** | **Contact**  **Hr** | **CLO** |
| 1 | 1 | Understand the history and fundamental principles of neural networks.  Activities: Write a brief report on the evolution of neural networks.  Implement a simple neural network using Python/Numpy to classify linearly separable data. | 3 |  |
| 2 | 2 | Explore basic neural network architectures for classification.  Activities: Implement and train a Perceptron on a binary classification dataset.  Implement Adaline with the Least Mean Squares rule and compare its performance with the Perceptron. | 3 |  |
| 3 | 3 | Transition from shallow networks to deep learning models.  Activities:  Train and test shallow networks for classification.  Extend the network to include one or more hidden layers and observe performance improvements. | 3 |  |
| 4 | 4 | Learn various learning rules and optimization methods.  Activities: Implement gradient descent to minimize a simple cost function.  Explore the Delta Rule, Hebbian Learning, and Differential Hebbian Learning for updating weights. | 3 |  |
| 5 | 5 | Understand unsupervised learning with Kohonen Self-Organizing Maps.  Activities: Implement a SOM to cluster data points in 2D.  Visualize the trained SOM and interpret the results. | 3 |  |
| 6 | 6 | Explore associative memory and bidirectional associative memory (BAM).  Activities: Implement and test an associative memory model.  Extend the model to BAM and demonstrate its bidirectional recall capability. | 3 |  |
| 7 |  | **Open Ended Lab/Project Assigned** |  |  |
| 8 | **Midterm Exam** | | | |
| 9 | 7 | Train and evaluate multi-layer perceptrons using backpropagation.  Activities: Build a multi-layer perceptron using PyTorch or TensorFlow for a classification task. Visualize the training process, loss, and accuracy. | 3 |  |
| 10 | 8 | Introduce convolutional layers and their role in feature extraction.  Activities: Implement a simple CNN for image classification.  Analyze the computational complexity of the CNN architecture. | 3 |  |
| 11 | 9 | Explore dimensionality reduction and feature learning.  Activities:  Implement and train an autoencoder on an image dataset.  Visualize encoded features and reconstructed outputs. | 3 |  |
| 12 | 10 | Handle sequential data using RNNs and Long Short-Term Memory (LSTM) networks.  Activities: Implement a simple RNN for time-series prediction.  Extend it to an LSTM model and compare the results. | 3 |  |
| 13 | 11 | Understand and implement generative adversarial networks.  Activities: Build a simple GAN to generate synthetic data/images.  Train the GAN and evaluate the quality of generated samples. | 3 |  |
| 14 | 12 | Mitigate overfitting and explore advanced architectures.  Activities: Experiment with data augmentation, dropout, batch normalization, and elastic distortions. Implement and compare ResNet and GoogleNet architectures on an image dataset. | 3 |  |
| 15 |  | **Revision / Open Ended Lab/Project Assessment** |  |  |
| 16 |  | **Open Ended Lab/Project Assessment** |  |  |
| 17 | **Final Exam** | | | |

|  |  |
| --- | --- |
| 1. **IU Assessment / grading Policy** | **Instructor grading for course \*** |
| Lab Manual 0-10%  Labs Task Assessment 10-20%  Projects/OEL/PBL 5-20%  Mid Semester Examination/ 20-30%  End Semester Examination 40-50% | 10  15  10  25  40 |