Vidya Pratishthan's
Kamalnayan Bajaj Institute of Engineering and Technology, Baramati
(An Autonomous Institute)

Board of Studies: Artificial Intelligence and Data Science

Syllabus: Honor in Computational Intelligence 2025 Pattern w.e.f 2026-2027
(As per NEP 2020)

Vidya Pratishthan's
Kamalnayan Bajaj Institute of Engineering and Technology, Baramati
Board of Studies: Artificial Intelligence and Data Science
Syllabus: Honor in Computational Intelligence 2025 Pattern w.e.f 2026-2027

Sem	Course Code	Course Name	Teaching Scheme			Examination Scheme and Marks							Credits			
			TH	PR	TUT	CA	ISE	ESE	TW	PR	OR	Total	TH	PR	OR	Total
III	AI25281TH	Introduction to Computational Intelligence	2	-	-	10	-	60	-	-	-	70	2		-	3
	AI25281PR	Introduction to Computational Intelligence	-	2	-	-	-	-	30	-	-	30	-	1	-	
IV	AI25291TH	Soft Computing: Intelligent Problem-Solving Techniques	2	-	-	10	-	60	-	-	-	70	2	-	-	3
	AI25291PR	Soft Computing: Intelligent Problem-Solving Techniques	-	2	-	-	-	-	30	-	-	30	-	1	-	
V	AI25381TH	Intelligent Optimization Algorithms	3	-	-	10	30	60	-	-	-	100	3	-	-	4
	AI25381PR	Intelligent Optimization Algorithms	-	2	-	-	-	-	30	-	-	30	-	1	-	
VI	AI25391TH	Quantum Artificial Intelligence	3	-	-	10	30	60	-	-	-	100	3	-	-	4
	AI25391PR	Quantum Artificial Intelligence	-	2	-	-	-	-	30	-	-	30	-	1	-	
VII	AI25481TH	Quantum Machine Learning	3	-	-	10	30	60	-	-	-	100	3	-	-	4
	AI25481PR	Quantum Machine Learning	-	2	-	-	-	-	30	-	-	30	-	1	-	
Total			13	10	-	50	90	300	150	-	-	590	13	05	-	18

28/03/2026

Mr.P.N. Shendage
Academic Coordinator

Dr.C.S. Kulkarni
Head of Department

Dr.S.M. Bhosale
Dean Academics

Dr.A.H.. Kolekar
Controller of Examination

Dr.S.B. Lande
Principal

Vidya Pratishthan's
Kamalnayan Bajaj Institute of Engineering and Technology, Baramati
(Autonomous Institute)

AI25281- Introduction to Computational Intelligence

Teaching Scheme:	Credits	Examination Scheme:
Theory: 2 Hours/Week **Practical: 2Hours/Week**	**03**	**CAA: 10 Marks** **ESE: 60 Marks** **TW: 30 Marks**

Prerequisites: Python Programming

Course Objectives:

- To provide a strong foundation on fundamental concepts in Computational Intelligence.
- To apply these techniques in applications which involve perception, reasoning and learning.
- To apply Computational Intelligence techniques for pattern recognition.
- To apply Computational Intelligence techniques primarily for machine learning activities like classification, clustering etc.

Course Outcomes (COs): The students will be able to learn:
CO1: Apply concepts of computational intelligence to real-world problems.
CO2: Analyze and implement machine learning models on various data sets.
CO3: Interpret the results of probabilistic models and inference techniques in practical applications.
CO4: Evaluate the effectiveness of various computational intelligence techniques in case studies, including IoT applications and decision-making processes.

Course Contents

Unit I Introduction to Computational Intelligence (06 Hours)
History, Biological neurons & Artificial models, intelligence machine, man-machine interaction, data mining for IoT, Relation between AI, ML, DL, data science and CI. Types of data analytic – predictive, prescriptive, descriptive, and diagnostic, Data analytic rule, web scrapping

Unit II Classification (6 Hours)
Forms of Learning, Supervised Learning , Learning Decision Trees , Regression and Classification with Linear Models, Artificial Neural Networks, Non parametric Models, Support Vector Machines, Statistical Learning, Learning with Complete Data, Learning with Hidden Variables- The EM Algorithm, Reinforcement Learning.

Unit III Probabilistic Models (6 Hours)
Probability basics, Bayes Rule and its Applications, Bayesian Networks, Exact and Approximate Inference in Bayesian Networks**.**

Unit IV Deep Learning and case studies
Convolution Neural Network, Recurrent Neural Network, Case Study-Hand Written Digit Recognition, VGG16 pre-trained model for image classification, Restricted Boltzmann Machines (RBM)

Text Books:

1. Andreis P. Engelbrecht, "Computational Intelligence an introduction", 2nd edition, Wiley publication
2. Nazmul Siddique, Hojjat Adeli, "Computational Intelligence, Synergies of Fuzzy logic, Neural Networks and Evolutionary computing", Wiley publication
3. Stuart Russell, Peter Norvig, ―Artificial Intelligence: A Modern Approac, Third Edition, Pearson Education / Prentice Hall of India, 2010.
4. Elaine Rich and Kevin Knight, ―Artificial Intelligence, Third Edition, Tata McGrawHill, 2010.

Reference Books:

1. James M. Keller, Derong Liu, David B. Fogel, “Fundamentals of Computational Intelligence: Neural Networks, Fuzzy Systems, and Evolutionary Computation”, John Wiley & Sons, 2016.
2. Mitchell Melanie, “An Introduction to Genetic Algorithms”, The MIT Press Cambridge, Massachusetts, MIT Press paperback edition, 1998.
3. Dan W.Patterson, ―Introduction to Artificial Intelligence and Expert Systems‖, PHI, 2006.
4. Nils J. Nilsson, ―Artificial Intelligence: A new Synthesis‖, Harcourt Asia Pvt. Ltd., 2000.

E-Resources:

1. **https://nptel.ac.in/courses/106102220**
2. **https://onlinecourses.nptel.ac.in/noc23_cs87/preview**
3. **https://onlinecourses.nptel.ac.in/noc22_ee21/preview**

List of Assignments

Comparison of regression models on predicting the medical expenses, In this assignment you will be implementing a learning model of regression which will predict the expenses through insurance data. You will be writing a code in jupyter noebook. Grading of this assignment will be based on following points

1. Data cleaning
2. Feature scaling
3. Gradient Descent

Link for dataset https://raw.githubusercontent.com/MintForever/CS4774/master/insurance.csv

2. Non linear classification with Support Vector Machine, In this assignment will be responsible to implement your own SVM not the provided by scikitlean library.
3. Clustering with K-Mean Algorithm link to download the dataset https://www.kaggle.com/camnugent/california-housing-prices
4. Build a neural network of XOR dataset, In this assignment you will be able to implement your own neural network for XOR dataset.

   The link for this dataset https://www.kaggle.com/datasets/bipinmaharjan/xor-dataset
5. Build a CNN model to recognize the classes of the vehicle, In this assignment you will be able write a program which will identify the class of the vehicle based upon learning experience. The dataset used can be downloaded from https://www.kaggle.com/datasets/dataclusterlabs/indian-vehicle-dataset

Vidya Pratishthan's
Kamalnayan Bajaj Institute of Engineering and Technology, Baramati
(Autonomous Institute)

AI25291- Soft Computing: Intelligent Problem-Solving Techniques

Teaching Scheme:	Credits	Examination Scheme:
Teaching Scheme: **Theory: 2 Hours/Week** **Practical: 2 Hours/Week**	**Credits** **03**	**Examination Scheme:** **CAA: 10 Marks** **ESE: 60 Marks** **TW: 30 Marks**

Prerequisites: Python Programming

Course Objectives:

- To explain the concept of soft computing techniques with their applications in product design. Manufacturing and operations with case studies
- To expose students to the concept of fuzzy logic and their applications in mechanical system.
- To familiarize the deep learning model development using artificial neural network.
- To introduce the concept of genetic algorithm and various advanced algorithms.
- To apply Markov models for system and process modeling and optimization

Course Outcomes (COs): The students will be able to learn:

CO1. Apply soft fuzzy logic approach for solving different problems in absence of sufficient data and using expert judgments.

CO 2. Develop deep learning model using artificial neural network.

CO 3. Apply genetic algorithms other random search procedures useful while seeking global optimum in self learning situations.

CO 4. Apply reinforcement and deep learning models for different data sets and optimize the system performance

Course Contents

Unit I Fuzzy Logic (7 Hours)

Fuzzy Systems Fuzzy set theory: Fuzzy sets, Operations, Membership Functions, Fuzzy relations and their composition, Measures, Rules, Propositions, Implications, and inferences, Defuzzification techniques, Logic controller design, Some applications of fuzzy logic.

Unit II Artificial Neural Network (7 Hours)

Artificial Neural Network (ANN) Neuron, Nerve structure and synapse, Biological and artificial neurons, Architectures – single layer and multilayer feed forward networks, recurrent networks. Back propagation algorithm, Working principle, Types of ANN, Activation functions – linear, Sigmoid, Tanh, supervised and unsupervised learning, Training techniques for ANNs, Applications, advantages, and limitations.

Unit III Genetic Algorithms (7)

Basic Genetics, Concepts, Working Principle, Creation of Offspring, Encoding, Fitness Function, Selection Functions, Genetic Operators-Reproduction, Crossover, Mutation; Genetic Modeling, Advantages, limitations and applications, Comparison between GA and traditional algorithms

Unit IV Reinforcement Learning (RL) and Deep Learning (7)

What is reinforcement learning? Terms used; Key features; Working process; Approaches – value-based, policy-based, and model-based; Elements – policy, reward signal, value function, model of the environment; The Bellman equation; Types – positive and negative; RL algorithms; Q-learning; Comparison between RL and supervised learning.

Text Books:

1. Neural Networks: A Comprehensive Foundation by S. Haykin, Pearson.
2. Fuzzy Logic with Engineering Application by T. J. Ross, John Wiley and Sons.
3. Evolutionary Computation by D.B. Fogel, IEEE Press.
4. D. K. Pratihar, Soft Computing, Narosa Publishing House, 2008.
5. An Introduction to Genetic Algorithm Melanic Mitchell (MIT Press).

Reference books:

1. Evolutionary Algorithm for Solving Multi-objective, Optimization Problems (2nd Edition), Collelo, Lament, Veldhnizer (Springer).
2. Fuzzy Logic with Engineering Applications Timothy J. Ross (Wiley).
3. Neural Networks and Learning Machines Simon Haykin (PHI).
4. Sivanandam, Deepa, Principles of Soft Computing, Wiley.
5. Jang J.S.R, Sun C.T. and Mizutani E, "Neuro-Fuzzy and soft computing", Prentice Hall. 6. Timothy J. Ross, "Fuzzy Logic with Engineering Applications", McGraw Hill

E-Resources:

1. https://nptel.ac.in/courses/106102220
2. https://onlinecourses.nptel.ac.in/noc23_cs87/preview
3. https://onlinecourses.nptel.ac.in/noc22_ee21/preview

List of Assignments

1. Design triangular, trapezoidal, Gaussian, and sigmoid membership functions for a given problem (e.g., temperature classification: Cold, Warm, Hot). Plot these membership functions and analyze their behavior. **Tools:** Python libraries like skfuzzy, MATLAB.

2. Implement a simple neural network to solve the non-linear XOR problem. **Task:** Use an MLP with at least one hidden layer to classify the XOR dateset. Train the network using back-propagation. Compare the results with and without activation functions (e.g., ReLU, Sigmoid). **Tools:** Python libraries - NumPy, TensorFlow/Keras

3. Build a neural network to classify handwritten digits. **Task:** Load the MNIST dataset. Design and train a feedforward neural network with multiple hidden layers. Experiment with different optimizers (e.g., SGD, Adam). Visualize training loss and accuracy over epochs. **Tools:** Keras, TensorFlow, Matplotlib

4. Optimize the selection of items to maximize value while respecting weight constraints. **Task:** Implement a genetic algorithm to solve the **0/1 knapsack problem**. Define chromosomes as binary strings where 1 indicates selecting an item. Apply selection, crossover, and mutation operations. Compare results with a greedy algorithm. **Tools:** Python libraries: NumPy, Matplotlib.

5. Implement **Q-Learning**, a model-free RL algorithm. **Task:** Create a 4x4 grid world with obstacles and a goal state. Implement Q-Learning to learn the optimal policy. Visualize the Q-table, optimal actions, and rewards after convergence. **Tools:** Python libraries: NumPy, Matplotlib.

Vidya Pratishthan's Kamalnayan Bajaj Institute of Engineering and Technology, Baramati (Autonomous Institute)		
AI25381- Intelligent Optimization Algorithms		
Teaching Scheme: **Theory: 3 Hours/Week** **Practical: 2 Hours/Week**	**Credits** **04**	**Examination Scheme:** **CAA:10 Marks** **ISE: 30 Marks** **ESE: 60 Marks** **TW: 30Marks**
Prerequisites: Discrete Mathematics (AI23202), Data Structures (AI23201)		

Course Objectives:

- To understand the need of optimization Algorithms
- To apply the optimization techniques while solving the problems
- To understand the constraints applied and optimization of the algorithm
- To optimize searching strategies
- To understand and use Self Optimizing algorithms

After completion of the course, learners should be able to

CO1: Identify Optimization Concepts to incorporate in problem solving in effective way

CO2: To formulate given optimization problem mathematically precisely

CO3: To create model using Optimization Techniques, like linear programming, integer programming and dynamic programming

CO4: To select an optimization strategy to tackle complex optimization problems and evaluate the optimization algorithms

CO5: To distinguish strategies of Optimization Algorithms selected

CO6: To construct an Optimization strategy to solve different problems

Course Contents

Unit-I Introduction (7)

Introduction, Fundamentals of Optimization, general structure of the optimization algorithms, types of optimization problems, examples of optimization, formulation of optimization problem, classification of optimization algorithms, traveling salesman and knapsack problem

Unit-II Classical Optimization (7)

Introduction, Mathematical model of optimization, Optimality conditions, Solution techniquesPenalty function, Linear programming (LP)-Formulation of LP Problem Optimality conditions, Integer Linear Programming, LP duality-Farkas Lemma, Quadratic Programming (QP)-Convex QP problems, Convex Programming, general constraint optimization problem

Unit-III Constraint Optimization (7)

Introduction Linear Programming-Simplex Method, Revised Simplex Method, Karmarkar's Method, Duality Theorem and Transportation Problem, Non-linear Programming-Quadratic and Geometric Programming, Karush–Kuhn–Tucker (KKT) conditions test as necessary condition. Dynamic Programming- Continuous vs Discrete dynamic programming, multistage graph problem, traveling salesman and knapsack problem

Unit-IV Search Optimization (7)

Introduction, Genetic Algorithms-Initialize population, Fitness Evaluation, Reproduction, Crossover andMutation, Multimodel test function, Solving linear equations with genetic algorithm, Simulated Annealing(SA)-Annealing and Boltzmann Distribution, Parameters, SA Algorithm, Unconstrained Optimization, Basic Convergence Properties, SA Behavior in Practice and Stochastic Tunneling, Particle Swarm Optimization(PSO)-Introduction, Swarm Behavior, PSO Algorithm, Variants of PSO Algorithm

Unit-V Differential Evolution and Swarm Optimization (7)

Introduction, Differential Evolution-Introduction, Differential Evolution, Variants, Choice of Parameters Convergence Analysis and Implementation. Swarm Optimization-Swarm Intelligence, PSO Algorithm, Accelerated PSO, Convergence Analysis-PSO, Binary PSO, Multiobjective

Optimization- Pareto Optimality, Constraint Methods, Weight Methods, Preference Elicitation, Ant colony optimization(ACO)

Unit-VI Self Tuning Algorithms (7)

Introduction, Algorithm Analysis and Parameter Tuning, Framework for Self-Tuning Algorithms, A Self-Tuning Firefly Algorithm- Firefly Behavior, Standard Firefly Algorithm, Variations of Light Intensity and Attractiveness, Controlling Randomization Variants of the Firefly Algorithm, Firefly Algorithms in Applications, Bat Algorithm- Echolocation of Bats, Bat Algorithms, Binary Bat Algorithms, Convergence Analysis, Applications

Text Books:

1. Andreas Antoniou, Wu-Sheng Lu, “Practical optimization algorithms and engineering applications”, Springer, 2007
2. Vasuki A., “Nature Inspired Optimization Algorithms”, CRC Press, 2020
3. Mykel J. Kochenderfer, Tim A. Wheeler, “Algorithms for Optimization”, MIT Press, 2019

Reference Books:

1. Rajesh Kumar Arora, “Optimization Algorithms and Applications”, Chapman & Hall, CRC, 2015

2. A Schrijver, “Theory of Linear and Integer Programming” (Wiley Series in Discrete Mathematics & Optimization)

3. V. Chvatal, “Linear Programming” W. H. Freeman **ISBN-13 :** 978-0716715870

MOOC Courses:

1.https://www.coursera.org/learn/optimization-for-decision-making

2. https://www.coursera.org/learn/solving-algorithms-discrete-optimization

List of Assignments

1. A mechanical industry has three warehouses in the Solapur area and needs to deliver camshafts to its three shops in and around for tomorrow. The three shops demand 10, 20, and 40 units respectively. The current stock level of shafts in the three warehouses are 80, 62, and 32 respectively. Delivery costs from each warehouse to each store are different due to different distances. Find the least expensive way to deliver the chairs to the stores. The delivery cost Matrix is represented below. Use Linear Programming to write a program in python.

Warehouse	Shop 1	Shop 2	Shop 3
Warehouse 1	3000/-	2000/-	5000/-
Warehouse 2	2000/-	7000/-	3000/-
Warehouse 3	2200/-	2400/-	1000/-

2. Write a python program to maximize the function with constraints find out the values of such a that it maximizes the given objective function using Quadratic Programming

3. Write a python program to minimize the flow from source S to the destination D in a multi-stage graph with a property , Here and are the partitions of the graph G and no connecting edge in the same partition. Find out a path from S to the D with minimum cost.

4. A linear equation of the form is to be solved with the help of Genetic Algorithms applying Initialize

population, Fitness Evaluation, Reproduction, Crossover and Mutation. Find out the approximate values of the coefficients with python programming

5. A delivery vehicle delivers the items to the different cities, it starts from his own city and visits all other cities once except his city of residence. You have to suggest a tour of shortest distance using Simulated Annealing

6. There is a dataset D over , supplied to the machine learning algorithm for classification purposes. We are cautious about the selection of the attributes for training and testing the model. Use Particle Swarm Optimization for feature section and show that the performance of a classification algorithm is improved over the use of PSO.

7. A Binary Particle Swarm Optimization algorithm to be applied on a dataset D for selection of the features to be used for training a binary class classifier. Mine the performance of the classifier when Binary PSO is applied.

8. A CNN based classifier uses a set of images for training and efficient testing of the model, it has the property of self tuning its parameters such that the classification accuracy reaches to the maximum possible. Use tensorflow, keras or Pytorch to write the program

9. There are different jobs to be executed on a machine, each of the job comes with a triplet {job, enter_time, exit_time}, Prepare a schedule of the jobs using `firefly algorithm to maximize the profit

10. A machine learning task requires a dataset D, has features, not necessarily all the features to be used during training and testing of the algorithm/model. Select an optimization technique like Firefly algorithm to choose the important features to be used during training of an algorithm. Write a python program with suitable libraries to carry out mentioned task

11. Mini Project:- Design and develop a mini project for classification of images into different categories using CNN along with Particle Swarm Optimization/Firefly/Binary PSO. The group of students developing this application need to use different datasets. Priority must be given for self data creation, publishing and using it in this project.

Vidya Pratishthan's Kamalnayan Bajaj Institute of Engineering and Technology, Baramati (Autonomous Institute)		
AI25391- Quantum Artificial Intelligence		
Teaching Scheme: **Theory: 3 Hours/Week** **Practical: 2 Hours/Week**	**Credits** **04**	**Examination Scheme:** **CAA:10 Marks** **ISE: 30 Marks** **ESE: 60 Marks** **TW: 30 Marks**

Prerequisites: Introduction to Computational Intelligence, Stat and Linear Algebra, Physics(Quantum Mechanics Unit)

Course Objectives:

- To get acquainted with the principles of quantum computing and the usage of Linear algebra in Quantum Computing
- To understand the Architecture of Quantum computing and solve examples of Quantum Fourier Transforms
- To understand the concepts of basic and advanced Quantum Algorithms and apply them to various problems.
- To study quantum machine learning and apply these to develop hybrid solutions
- To study the Quantum Theory with Fault-Tolerant Quantum techniques
- To understand Problem-Solving using various peculiar search strategies for AI

Course Outcomes (COs): The students will be able to:
CO1: Understand quantum requirements and formulate design solutions using quantum circuits.
CO2: Illustrate applicable solutions in one or more application domains using a quantum architecture that integrates ethical, social, and legal concerns
CO3: Apply the Advanced Quantum Algorithms on real time problem
CO4: Analyze the quantum machine learning algorithms and their relevant application
CO5: Analyze quantum information processing & its relevant algorithms
CO6: Evaluate suitable algorithms for AI problems

Course Contents

Unit I: Introduction to Quantum Computation (7 Hours)
Overview of Quantum Computation: Single qubit gates, Multiple qubit gates, Measurements in bases Vs computational basis, Quantum circuits, Qubit copying circuit, Example: Bell states & quantum teleportation. Basics of Linear Algebra: Hilbert Spaces, Products and Tensor Products, Matrices, Graphs, and Sums Over Paths, Example.

Unit II: Knowledge Representation and Reasoning **(7 Hours)**
The Framework of Quantum Mechanics: The State of a Quantum System, Time-Evolution of a Closed System, Composite Systems, Mixed States and General Quantum Operations, Universal Sets of Quantum Gates, Quantum measurement and quantum entanglement The quantum Fourier transform and its Applications- The quantum Fourier transform, Phase estimation, order-finding and factoring, General applications of the quantum Fourier transform- Period-finding, Discrete logarithms, The hidden subgroup problem

Unit III: Quantum Algorithms (7 Hours)
Probabilistic Versus Quantum Algorithms, Phase Kick-Back, The Deutsch Algorithm, The Deutsch–Jozsa Algorithm, Simon's Algorithm, Shor's Algorithm, Factoring Integers, Grover's Algorithm

Unit IV: Quantum Machine Learning (7 Hours)
Quantum Enhanced Machine Learning: Quantum Algorithms for Linear Algebra, Regression, Clustering, Nearest Neighbour Search, Classification. Quantum Boosting, Quantum Support Vector Machines, Quantum Neural Networks, Variational Quantum Algorithms.

Unit V: Quantum Information Processing (7 Hours)
Classical Error Correction: The Error Model Encoding, Error Recovery, The Classical Three-Bit Code, Fault Tolerance. Quantum Information: Quantum Teleportation, Quantum Dense Coding, Quantum

Key Distribution, Noise and error models in quantum systems, Quantum cryptography and secure communication.

Unit VI: Quantum Problem Solving & AI applications (7 Hours)

Quantum Problem Solving: Heuristic Search, Quantum Tree Search, Quantum Production System, Tarrataca's Quantum Production System Quantum AI Application: Introduction to Pennylane: a cross-platform Python library, Quantum Neural Computation, Quantum Walk – Random insect, Walk on graph, Case studies on Quantum-centric supercomputing: The next wave of computing, Quantum computing for data sciences

Text Books:

1. Nielsen, M. & Chuang I., "Quantum Computation and Quantum Information", 2002

2. Lipton and Reagan, "Quantum Algorithms via Linear Algebra: A Primer"

3. Kaye, LaFlamme and Mosca's, "Introduction to Quantum Computing"

4. Biamonte, J. et al., "Quantum Machine Learning", Nature, 2017

5. Andreas Wichert, "Principles Of Quantum Artificial Intelligence"

Reference Books:

1. Rieffel, E. G. & Polak W. H., "Quantum computing: A gentle introduction", MIT Press, 2011

2. Farhi, E., Goldstone, J. & Gutmann, S., "A quantum approximate optimization algorithm", arXiv preprint arXiv:1411.4028, 2014

3. Kuttler, "Elementary Linear Algebra", 2012

4. Kepner and Gilbert, "Graph Algorithms in the Language of Linear Algebra", 2011

5. Russell, S. & Norvig, P., "Artificial Intelligence: A modern approach", 4 th edition, Pearson Education, 2021

E-Resources:

1.http://mmrc.amss.cas.cn/tlb/201702/W020170224608149940643.pdf
2.https://arxiv.org/pdf/1611.09347.pdf
3.http://mmrc.amss.cas.cn/tlb/201702/W020170224608150244118.pdf
4.https://www.researchgate.net/publication/282378154_FPGA-based_quantum_circuit_emulation
5.Microsoft Quantum Development Kit: https://www.microsoft.com/enus/quantum/development-kit Forest
6. Learn quantum programming: https://pennylane.ai/qml/
7. Quantum machine learning: https://qiskit.org/learn/course/machine-learning-course/
8. Center for Excellence in Quantum Technology: https://research.ibm.com/blog/next-wave-quantumcentric-supercomputing

List of Assignments

1. In quantum computing, Bell states are a fundamental example of quantum entanglement. Quantum teleportation allows transferring quantum states between qubits using entanglement and classical communication.

Tasks:

- Implement a **quantum circuit** to generate the four Bell states using Hadamard and CNOT gates.
- Simulate **quantum teleportation** of an arbitrary qubit state using an entangled Bell pair.
- Measure the output and verify that the original state is reconstructed at the receiver.

Expected Outcome:

- Correct generation of Bell states.
- Successful transmission of quantum states using teleportation.
- Visualization of measurement results.

2. Quantum circuits use multiple qubit operations such as controlled gates (CNOT, Toffoli) to create entanglement. This assignment explores the effect of different gates on quantum states.

Tasks:

- Construct a quantum circuit using **Hadamard, Pauli-X, and CNOT gates**.
- Implement a **Toffoli (CCNOT) gate** and demonstrate its effect on a 3-qubit system.
- Simulate the circuit using Qiskit and analyze the measurement outcomes.

Expected Outcome:

- Correct implementation of multi-qubit gate operations.
- Understanding of quantum entanglement and state evolution.

3. The Quantum Fourier Transform (QFT) is a fundamental operation in quantum algorithms such as phase estimation. This assignment requires implementing QFT and inverse QFT using Qiskit.

Tasks:

- Implement **QFT** for a 2-qubit and 3-qubit system.
- Verify the correctness of QFT by applying its inverse and checking if the input state is recovered.
- Analyze the measurement results to understand the frequency domain representation.

Expected Outcome:

- Correct implementation of QFT and inverse QFT.
- Verification of transformation properties through simulation.

4. Quantum measurement collapses superposition states. This assignment explores quantum measurement and entanglement using Qiskit.

Tasks:

- Construct a **quantum circuit** that demonstrates **superposition and measurement** using Hadamard and measurement gates.
- Implement **quantum entanglement** between two qubits and demonstrate Bell inequality violation.
- Simulate the circuit and visualize measurement results.

Expected Outcome:

- Understanding of quantum measurement and state collapse.
- Demonstration of entanglement using Qiskit.

5. The Deutsch-Jozsa algorithm **determines if a function is** constant **or** balanced **using quantum parallelism. This assignment implements the algorithm for a 2-qubit system.**

Tasks

- Implement the Deutsch-Jozsa algorithm for a 2-qubit system in Qiskit.
- Define **constant and balanced functions** and verify their classification.
- Measure the output and analyze the probability distribution.

Expected Outcome:

- Efficient determination of function type in a single quantum evaluation.

6. Shor's algorithm is a quantum factorization algorithm **that efficiently finds the prime factors of a composite number. This assignment involves simulating Shor's algorithm using Qiskit.**

Tasks

- Implement a **quantum circuit** for Shor's algorithm for factoring **15** (3 × 5).
- Simulate the algorithm and verify its ability to find factors.
- Analyze the effect of quantum phase estimation in the process.

Expected Outcome

- Understanding of **period finding** in integer factorization.
- Verification of quantum speedup for factorization.

7. Support Vector Machines (SVMs) are widely used for classification. Quantum Support Vector Machines (QSVMs) use quantum kernels to classify data more efficiently.

Tasks

- Implement **QSVM** using Qiskit's quantum machine learning module.
- Train QSVM on a **binary classification dataset** (e.g., distinguishing handwritten digits 0 and 1).
- Compare the performance of QSVM with a classical SVM.

Expected Outcome

- Successful classification using QSVM.
- Performance comparison with classical SVM.

8. Variational Quantum Classifiers (VQCs) are quantum neural networks that use parameterized quantum circuits for machine learning.

Tasks

- Implement a **VQC** using Qiskit's qml module.
- Train the model on a simple dataset and classify test samples.
- Analyze the impact of quantum circuit depth on classification accuracy.

Expected Outcome

- Quantum-enhanced classification using **variational circuits**.

9. Quantum cryptography enables secure communication using quantum key distribution (QKD). The BB84 protocol **ensures secure key exchange between Alice and Bob.**

Tasks

- Implement the **BB84 quantum cryptography protocol** using Qiskit.
- Simulate an eavesdropper attack and measure its effect on security.
- Analyze the probability of successful key distribution.

Expected Outcome

- Secure quantum key exchange using BB84.
- Detection of eavesdropping attempts.

10. Quantum walks provide an alternative to classical random walks and are useful in AI search algorithms**.**

Tasks

- Implement a **Quantum Walk** on a **graph** using Qiskit.
- Simulate a **random walk of an insect on a 1D line** and measure probabilities at different positions.
- Compare quantum and classical random walk behavior.

Expected Outcome

- Understanding of **quantum walk dynamics**.
- Speedup in search algorithms using quantum walks.

Vidya Pratishthan's
Kamalnayan Bajaj Institute of Engineering and Technology, Baramati
(Autonomous Institute)

AI25481- Quantum Machine Learning

Teaching Scheme:	Credits	Examination Scheme:
Theory: 3 Hours/Week Practical: 2 Hours/Week	04	CAA:10 Marks ISE: 30 Marks ESE: 60 Marks TW: 30Marks

Prerequisites: Quantum Artificial Intelligence, Machine Learning, Linear Algebra

Course Objectives:

- To understand the theoretical foundations of quantum machine learning.
- To study quantum algorithms used in machine learning tasks.
- To explore hybrid quantum–classical learning models.
- To implement QML algorithms using frameworks such as Qiskit and PennyLane.
- To analyze the advantages and limitations of QML algorithms.
- To apply quantum models for solving real-world AI problems.

Course Outcomes (COs): The students will be able to:
CO1: Understand the mathematical foundations of quantum machine learning.
CO2: Explain quantum data encoding and quantum feature spaces.
CO3: Implement variational quantum circuits for machine learning.
CO4: Apply quantum algorithms for classification, clustering, and regression.
CO5: Analyze hybrid quantum–classical machine learning models.
CO6: Develop practical QML solutions using quantum programming frameworks.

Course Contents

Unit I: Introduction to Quantum Computation (6 Hours)
Introduction to Quantum Machine Learning, Relationship between Quantum Computing and Machine Learning, Limitations of Classical Machine Learning, Quantum advantage in machine learning, Quantum data representation and encoding methods - Basis encoding, Amplitude encoding, Angle encoding. Quantum feature spaces and kernel methods, Overview of quantum datasets and benchmark problems.

Unit II: Quantum Linear Algebra for Machine Learning (7 Hours)
Quantum algorithms for linear algebra - Matrix representation of quantum states, Quantum inner products and distances, Swap test for similarity measurement. Quantum algorithms for: Linear systems solving (HHL algorithm), Principal component analysis (QPCA), Quantum recommendation systems, Applications of linear algebra in machine learning models.

Unit III: Quantum Kernel Methods and Classification (7 Hours)
Quantum feature maps, Quantum kernel estimation, Quantum Support Vector Machines (QSVM), mparison with classical SVM, Kernel-based quantum classification, Quantum nearest neighbor algorithms, Applications in pattern recognition and classification tasks.

Unit IV: Variational Quantum Machine Learning (7 Hours)
Parameterized quantum circuits (PQC), Variational quantum algorithms, Variational Quantum Classifier (VQC), Variational Quantum Eigensolver (VQE), Gradient evaluation in quantum circuits, Hybrid quantum–classical optimization, Training quantum models with classical optimizers, Applications in classification and regression.

Unit V: Quantum Neural Networks and Deep Learning (7 Hours)
Quantum neural network architectures - Quantum perceptron, Quantum convolutional neural networks, Quantum recurrent neural networks. Expressibility and trainability of quantum circuits Barren plateau problem, Hybrid deep learning models combining classical and quantum networks

Unit VI: Practical Quantum Machine Learning Applications (6 Hours)
Implementation of QML using – Qiskit, PennyLane. Applications: Quantum classification, Quantum

clustering, Quantum regression. Case studies - Image classification using quantum kernels, Drug discovery with quantum ML, Quantum optimization problems, Future directions of quantum machine learning.

Text Books:

1. **Quantum Computation and Quantum Information**, Michael A. Nielsen and Isaac L. Chuang, 10th Anniversary Edition, 2010, Cambridge University Press.
2. **Supervised Learning with Quantum Computers**, Maria Schuld and Francesco Petruccione, 1st Edition, 2018, Springer International Publishing.
3. **Machine Learning with Quantum Computers**, Maria Schuld, 1st Edition, 2021, Springer International Publishing.
4. **Quantum Machine Learning: What Quantum Computing Means to Data Mining**, Peter Wittek, 1st Edition, 2014, Academic Press (Elsevier).

Reference Books:

1. **Quantum Computing: A Gentle Introduction**, Eleanor G. Rieffel and Wolfgang H. Polak, 1st Edition, 2011, MIT Press.
2. **Quantum Algorithms via Linear Algebra: A Primer**, Richard J. Lipton and Kenneth W. Regan, 1st Edition, 2014, MIT Press.
3. **Principles of Quantum Artificial Intelligence**, Andreas Wichert, 1st Edition, 2014, World Scientific Publishing.
4. **Artificial Intelligence: A Modern Approach**, Stuart Russell and Peter Norvig, 4th Edition, 2021, Pearson Education.

E-Resources:

1. **Quantum Machine Learning Course**, IBM Quantum,
Available at: https://qiskit.org/learn/course/machine-learning-course/
2. **PennyLane Quantum Machine Learning Tutorials**, Xanadu Quantum Technologies,
Available at: https://pennylane.ai/qml/
3. **Quantum Algorithm Implementations and Tutorials**, Qiskit,
Available at: https://qiskit.org/documentation/
4. **Quantum Machine Learning Review Paper**, Jacob Biamonte et al., *Quantum Machine Learning*, Nature, 2017.
Available at: https://arxiv.org/pdf/1611.09347.pdf
5. **Microsoft Quantum Development Kit Documentation**, Microsoft,
Available at: https://learn.microsoft.com/en-us/azure/quantum/
6. **Next Wave of Quantum-Centric Supercomputing**,
IBM Research,
Available at: https://research.ibm.com/blog/next-wave-quantum-centric-supercomputing
7. **Introduction to Quantum Machine Learning**, Massachusetts Institute of Technology OpenCourse are,
Available at: https://ocw.mit.edu

List of Assignments

1. **Quantum Data Encoding -**

Implement different quantum data encoding techniques such as **basis encoding, amplitude encoding, and angle encoding**. Prepare quantum states representing classical datasets. Analyze the advantages and limitations of different encoding schemes.

Expected Outcome:

Understanding how classical data can be mapped to quantum states for machine learning applications.

2. **Quantum State Similarity and Distance Measurement**

Implement the **Swap Test** to compute similarity between quantum states. Use the swap test for **distance estimation between feature vectors**.

Expected Outcome:

Ability to measure similarity between quantum data representations.

3. Quantum Kernel Estimation

Implement quantum feature maps, Estimate **quantum kernels** using quantum circuits, Compare quantum kernel results with classical kernel methods.

Expected Outcome:

Understanding kernel-based quantum learning approaches.

4. Quantum Support Vector Machine (QSVM)

Implement Quantum Support Vector Machine using Qiskit machine learning modules, Train QSVM on a simple dataset such as binary digit classification, Compare accuracy with classical SVM.

Expected Outcome:

Implementation and evaluation of a quantum classification algorithm.

5. **Variational Quantum Classifier**

Implement **Parameterized Quantum Circuits (PQC),** Train a **Variational Quantum Classifier (VQC)** using classical optimization methods. Analyze the effect of circuit depth on model accuracy.

Expected Outcome:

Understanding hybrid quantum–classical learning models.

6. **Quantum Principal Component Analysis**

Implement **Quantum PCA** for dimensionality reduction, Apply the algorithm to analyze dataset structure.

Expected Outcome:

Understanding quantum algorithms for data compression and feature extraction.

7. Quantum Neural Networks

Design a **Quantum Neural Network (QNN)** using parameterized circuits, Train the network on a small dataset, Analyze training behavior and optimization challenges.

Expected Outcome:

Understanding quantum neural architectures and training techniques.

8. Quantum Clustering Algorithms

Implement clustering using **quantum distance estimation methods,** Compare results with classical clustering algorithms such as K-means.

Expected Outcome:

Understanding unsupervised learning using quantum techniques.

9. Hybrid Quantum–Classical Learning Models

Combine classical neural networks with quantum circuits, Train hybrid models using classical optimizers.

Expected Outcome:

Understanding integration of quantum processors with classical machine learning pipelines.

10. Mini Project: Quantum Machine Learning Application, Students will design and implement a **mini-project** applying quantum machine learning techniques to a real-world problem. Possible applications include:

- Image classification
- Financial prediction models
- Optimization problems
- Drug discovery simulations
- Pattern recognition problems