Pokhara University
Faculty of Science and Technology

Course No.: CMP 342 (3 Credits)	Full marks: 100
Course title: Artificial Intelligence and Neural Network (3-1-3)	Pass marks: 45
Nature of the course: Theory and Practical	Total Lectures: 45 hrs
Level: Bachelor	Program: BE (Software)

1. Course Description

This course is designed to provide an in-depth study of Artificial Intelligence (AI) with a special focus on neural networks and their applications in software engineering. The course covers foundational AI concepts, problem-solving strategies, knowledge representation and expert systems. Students will also explore the design, implementation, and application of neural networks to solve complex software engineering problems.

2. General Objectives

- To provide the students with the foundational principles and techniques of AI and neural networks.
- To develop the skills in students to design and implement AI-based solutions in software engineering.
- To acquaint the students with the knowledge of various AI domains such as knowledge representation, expert systems and neural network.
- To acquaint the students with the knowledge of advanced topics in neural networks, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs).
- To provide the students with the knowledge to critically evaluate the ethical implications of AI and its impact on society.

3. Methods of Instruction

Lecture, Discussion, Readings, Practical works and Project works.

4. Contents in Detail

Specific Objectives	**Contents**

• Explain artificial intelligence, its approaches and its foundations. • Critically evaluate the ethical implications of AI and its impact on society.	1. **Introduction to Artificial Intelligence (4 hrs)** 1.1. Intelligence 1.1.1. Types of Intelligence 1.1.2. Components of Intelligence 1.2. Artificial Intelligence 1.2.1. Approaches of AI 1.2.1.1. Acting Humanly 1.2.1.2. Thinking Humanly 1.2.1.3. Thinking Rationally 1.2.1.4. Acting Rationally 1.2.2. Foundations of AI 1.2.3. History of AI 1.2.4. Risk and Benefits of AI 1.3. Ethics and Societal Implications 1.3.1. Ethical Implications of AI 1.3.2. AI and Society: Work and Automation, Employment, Privacy and Security 1.3.3. Governance and Regulation
● Design and implement intelligent agents.	2. **Intelligent Agents (5 hrs)** 2.1. Agents and Environments 2.2. Concept of Rationality 2.2.1. Performance Measures 2.2.2. Rationality and Rational Agent 2.3. Task environment and its properties 2.4. Structure of Agents 2.4.1. Agent programs 2.4.2. Types of agent programs 2.5. Learning Agents
● Formulate the real world problems and apply the search algorithms to solve them.	3. **Problem Solving and Search Algorithms (8 hrs)** 3.1. Problem Solving 3.1.1. Problem Solving Agents 3.1.2. Problem solving process 3.1.3. Production System 3.1.4. Well-defined and ill-defined problems 3.1.5. Problem formulation 3.2. Search Algorithms 3.2.1. Uninformed Search 3.2.1.1. Breadth- First Search 3.2.1.2. Depth-First Search 3.2.1.3. Iterative Deepening Search 3.2.2. Informed Search 3.2.2.1. Heuristics 3.2.2.2. Greedy Best-First Search 3.2.2.3. A* Search 3.3. Local Search and Optimization Problems 3.3.1. Hill-Climbing Search and its problems (Local maxima, plateaus, and ridges) 3.3.2. Simulated Annealing

	3.3.3. Genetic Algorithms 3.4. Adversarial Search and Game Playing 3.4.1. Minimax algorithm 3.4.2. Alpha-beta pruning
● Represent the knowledge of a domain and apply inference rules to draw conclusions.	4. **Knowledge Representation and Reasoning (4 hrs)** 4.1. Overview of Propositional and PredicateLogic 4.1.1. Syntax 4.1.2. Semantics 4.1.3. Inference and Resolution 4.2. Reasoning Under Uncertainty 4.2.1. Probabilistic Reasoning 4.2.1.1. Bayesian Networks 4.3. Other Approaches to Knowledge Representation 4.3.1. Semantic Nets and Frames 4.3.2. Ontological-Based Representation
● Design and develop an expert system to solve a real-world problem.	5. **Expert System (4 hrs)** 5.1. Definition and History of Expert System 5.2. Architecture of Expert Systems 5.3. Knowledge Representation in expert system 5.3.1. Logic based representation 5.3.2. Rule-based system 5.3.3. Semantic networks 5.3.4. Ontology-based Systems 5.3.5. Frame-based Systems 5.4. Inference Mechanisms 5.4.1. Forward Chaining 5.4.2. Backward Chaining 5.5. Knowledge Acquisition and Learning 5.6. Applications of Expert Systems

● Design, implement and evaluate neural networks. ● Apply the neural Network for software testing and debugging.	6. **Artificial Neural Network (12 hrs)** 6.1. Introduction to Machine Learning 6.2. Types of Machine Learning 6.2.1. Supervised Learning 6.2.2. Unsupervised Learning 6.2.3. Reinforcement Learning 6.3. Introduction to Neural Network 6.3.1. Biological inspiration: neurons and synapse 6.3.2. Structure of a Neuron 6.3.2.1. Artificial Neural Network 6.3.2.2. Components: weights, biases and activation functions 6.3.3. Neural Network Architectures 6.3.3.1. Feedforward 6.3.3.2. Convolution 6.3.3.3. Recurrent 6.3.4. Perceptrons 6.3.4.1. Single layer perceptron 6.3.4.2. Multilayer Perceptron 6.3.4.3. Backpropagation 6.4. Training Neural Network 6.4.1. Forward and Backward Propagation 6.4.1.1. Forward Propagation 6.4.1.2. Backpropagation and Gradient Descent 6.4.2. Loss functions 6.4.2.1. Role of loss function 6.4.2.2. Mean Squared Error (MSE) 6.4.2.3. Cross-Entropy Loss 6.4.3. Regularization Techniques 6.4.3.1. Overfitting and underfitting 6.4.3.2. Regularization methods: L1, L2, Dropout, Batch Normalization 6.5. Neural Network Optimization Techniques 6.5.1. Gradient Descent 6.5.2. Adam OPtimizer 6.6. Model Evaluation 6.6.1. Cross-validation, confusion matrix, 6.6.2. Precision, recall, F1 -score 6.7. Neural Networks in Software Engineering 6.7.1. Neural Networks for Software Testing and Debugging 6.7.2. AI-Driven Code Generation and Optimization 6.7.3. Neural Networks in Automated Software Maintenance

● Design and implement Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs).	7. **Deep Learning (8 hrs)** 7.1. Convolution Neural Network (CNNs) 7.1.1. CNNs and their components 7.1.2. Convolution, Pooling and fully connected layers 7.1.3. Applications in image processing and computer vision 7.2. Recurrent Neural Networks (RNNs) 7.2.1. Basics of RNNs 7.2.2. Long Short-Term Memory (LSTM) 7.2.3. Gradient Recurrent Units (GRU) 7.2.4. Applications in time-series prediction 7.3. Advanced Topics in Neural Network 7.3.1. Transfer Learning and Pretrained Models 7.3.2. Generative Adversarial Networks (GANs) 7.3.3. Reinforcement Learning and Neural Network

5. Practical Works

Laboratory work of 45 hours per group of maximum 24 students should cover implementation of the following lab works:

SN	Implementation Description
1	Implement search algorithms (e.g., BFS, DFS, A*) in Python.
2	Develop a simple expert system using rule-based reasoning.
3	Build and train CNNs for image classification.
4	Build and train RNNs for sequence prediction.
5	Design a neural network-based tool for a specific software engineering task (e.g., bug detection, code optimization).

Students should submit a project work that uses all the knowledge obtained from this course to solve any problem chosen by themselves. The marks for the practical evaluation must be based on the project work submitted by students.

6. List of Tutorials

The various tutorial activities that suit your course should cover all the content of the course to give students a space to engage more actively with the course content in the presence of the instructor. Students should submit tutorials as assignments or class works to the instructor for evaluation. The following tutorial activities of 15 hours per group of maximum 24 students should be conducted to cover the content of this course:

A. Discussion-based Tutorials: (3 hrs)

   a. Evolution of Artificial Intelligence (Class discussion).
   b. AI and Society: Employment, Privacy and Security (Class discussion).

B. Problem solving-based Tutorials: (6 hrs)
   a. Apply a search technique to solve a real world problem.
   b. Design and develop a simple expert system that solves a real world problem.

C. Review and Question/Answer-based Tutorials: (6 hrs)
   a. A detailed case study on Tools and Frameworks for implementing neural networks: TensorFlow, PyTorch and Python (Oral Presentation in class).
   b. Case study on real-world applications of Neural Network in Software Engineering
   c. Students ask questions within the course content, assignments and review key course content in preparation for tests or exams.

7. Evaluation System and Students' Responsibilities

Evaluation System

The internal evaluation of a student may consist of assignments, attendance, internal assessment, lab reports, project works etc. The internal evaluation scheme for this course is as follows:

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theory		30		
Attendance & Class Participation	10%			
Assignments	20%			
Presentations/Quizzes	10%			
Internal Assessment	60%		Semester-End examination	50
Practical		20		
Attendance & Class Participation	10%			
Lab Report/Project Report	20%			
Practical Exam/Project Work	40%			
Viva	30%			
Total Internal		50		
Full Marks: 50 + 50 = 100				

Student Responsibilities

Each student must secure at least 45% marks separately in internal assessment and practical evaluation with 80% attendance in the class in order to appear in the Semester End

Examination. Failing to get such a score will be given NOT QUALIFIED (NQ) to appear for the Semester-End Examinations. Students are advised to attend all the classes, formal exam, test, etc. and complete all the assignments within the specified time period. Students are required to complete all the requirements defined for the completion of the course.

8. Prescribed Books and References

Text Books

1. Russell, S. J., & Norvig, P. (2016). *Artificial Intelligence: A Modern Approach*. Pearson.
2. Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

References

1. Ian, G. (2016). *Deep learning/Ian Goodfellow, Yoshua Bengio and Aaron Courville*. The MIT Press.
2. Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow.* O'Reilly Media, Inc.
3. Aggarwal, C. C. (2018). *Neural Networks and Deep Learning*. Springer.