Algorithms for ML and AI

Winter Course 25-26

INTRODUCTION

This intensive course introduces engineering students to core algorithms in Machine Learning (ML) and Artificial Intelligence (AI). Students explore the foundational theory and practical applications of learning algorithms, with hands-on implementation using Python and standard libraries. The course also integrates a company visit to Multiverse Computing, where students observe how ML/AI are applied in quantum computing and finance.

Prerequisites:

- Basic programming knowledge (Python preferred).
- Understanding of basic statistics and linear algebra.

Course Objectives:

- Understand key supervised, unsupervised, and deep learning algorithms.
- Implement core ML/AI models using Python libraries (Scikit-learn, TensorFlow).
- Optimize and evaluate models with practical datasets.
- Explore AI applications through real-world case studies.
- Learn about quantum computing and ML/AI convergence during a company visit.

Teaching Methodology

The course will be conducted using a blended learning approach that combines theoretical lectures, hands-on programming labs, interactive discussions, and practical projects. This methodology ensures a comprehensive understanding of both the foundational concepts and practical implementations of machine learning algorithms.

1. **Lectures**
    - **Purpose:** Introduce and explain theoretical concepts.
    - **Approach:** Interactive presentations, real-world examples, Q&A sessions.
2. **Hands-on Labs**
    - **Purpose:** Provide practical experience in implementing ML/AI algorithms.
    - **Approach:** Guided coding sessions, lab assignments, and code reviews.
3. **Interactive Discussions**

- **Purpose:** Facilitate deeper understanding through peer-to-peer and instructor-student interactions.
- **Approach:** Online discussion forums, in-class group discussions, and guest lectures from industry experts.

4. **Practical Projects**
   - **Purpose:** Apply learned concepts to real-world problems and develop project management skills.
   - **Approach:** Mid-term and final projects involving end-to-end machine learning pipelines, with presentations and peer feedback.
5. **Assessment and Evaluation**
   - **Purpose:** Evaluate students' understanding and application of course material.
   - **Approach:** Weekly quizzes, coding assignments, project evaluations, and participation assessment.
6. **Learning Resources**
   - **Purpose:** Provide additional materials to support learning.
   - **Approach:** Recommended textbooks, online resources, Kaggle competitions, and software tools like Jupyter Notebook, Scikit-learn, TensorFlow, etc.
7. **Feedback Mechanism**
   - **Purpose:** Continuously improve the course based on student feedback.
   - **Approach:** Regular surveys, feedback forms, and office hours for personalized guidance.

CONTENTS OF THE COURSE

Module 1: Intro to ML & Python for AI (4 hours)

Lecture 1: Introduction to Machine Learning (1 hours)

Definition and types (supervised, unsupervised, reinforcement learning)
Applications of machine learning
Overview of the course structure

Lecture 2: Basics of Python for ML (3 hours)

Python programming basics
Introduction to libraries: NumPy, Pandas, Matplotlib

Module 2: Supervised Learning Algorithms (8 hours)

Lecture 3: Supervised Learning I (4 hours)

Simple and multiple linear regression
Gradient descent algorithm

Evaluation metrics: MSE, RMSE, R^2 score

Lecture 4: Supervised Learning II (4 hours)

Decision Trees
Random Forests
k-Nearest Neighbors

Module 3: Unsupervised Learning Algorithms (4 hours)

Lecture 5: Clustering (2 hours)

K-Means Clustering
Hierarchical Clustering

Lecture 6: Principal Component Analysis (PCA) (2 hours)

Dimensionality reduction
Eigenvalues and eigenvectors
Covariance matrix

Module 4: Deep Learning Fundamentals (4 hours)

Lecture 7: Neural Networks (2 hours)

Perceptron and multi-layer perceptron
Backpropagation algorithm
Activation functions

Lecture 8: Deep Learning Basics (2 hours)

Convolutional neural networks (CNNs)
Recurrent neural networks (RNNs)
Introduction to deep learning frameworks (TensorFlow, Keras, PyTorch)
Fine-tuning techniques

Module 5: Model Evaluation and Optimization (4 hours)

Lecture 9: Model Evaluation Techniques (2 hours)

Cross-validation
ROC and AUC
Precision, recall, F1 score

Lecture 10: Hyperparameter Tuning (2 hours)

Grid search
Random search
Bayesian optimization

Lecture 11: Model Interpretability (1 hours)

SHAP values
LIME (Local Interpretable Model-agnostic Explanations)

Module 6: AI Applications (8 hours)

Lecture 12: Image, Text, and Time Series (4 hours)

CNNs for image recognition
NLP pipeline: Tokenization, embeddings, RNNs/LSTMs
Time Series: ARIMA, LSTM basics

Lecture 13: Special Topics in ML & AI (4 hours)

Reinforcement Learning: Q-learning, exploration vs. exploitation
Transfer Learning and fine-tuning
AutoML overview
AI Ethics: Fairness, bias, privacy

Module 7: Course Challenge based on Project-based Learning (4 hours)

Lecture 14: Mini Project & Presentation (4 hours)

Final team project work
Model development and testing
Preparation of presentation and technical summary

Module 8: Company Visit: Multiverse Computing (4 hours)

Lecture 30: Course Review and Application (4 hours)

Introduction to quantum-enhanced ML
Financial applications of AI
Demos of ML integration in quantum computing tools
Industry Q&A with engineers and researchers

Attendance is obligatory as well as any required reading or assignments. Throughout the course, in lecture or discussion sections we expect respect of one another, a positive environment in this class is up to each of the participants. To that point, if computers/tablets

or other devices are used during this time, they should be used solely for course material and note-taking, so as not to distract others.

At least 2 years of study in an Engineering degree completed before the program start.

SCHEDULE AND CREDITS

- Contact hours: 40 h.
- Estimated time for homework and study: 10 h.
- ECTS Credits: 4 (2 credits in USA).

LEARNING OUTCOMES

Upon successful completion of this course, engineering students will be able to:

1. Understand Fundamental Concepts of Machine Learning

- **Define and Distinguish:** Clearly define machine learning and distinguish between its various types (supervised, unsupervised, and reinforcement learning).
- **Recognize Applications:** Identify and explain the applications of machine learning across different industries.

2. Implement and Apply Machine Learning Algorithms

- **Code Algorithms:** Write and implement machine learning algorithms in Python using libraries such as NumPy, Pandas, Scikit-learn, TensorFlow, and PyTorch.
- **Use Preprocessing Techniques:** Apply data preprocessing techniques including data cleaning, normalization, feature extraction, and dimensionality reduction.

3. Analyze and Evaluate Model Performance

- **Choose Evaluation Metrics:** Select appropriate evaluation metrics for different types of machine learning models.
- **Interpret Results:** Interpret and analyze the results of machine learning models to assess their performance and identify potential improvements.

4. Optimize Machine Learning Models

- **Tune Hyperparameters:** Perform hyperparameter tuning using techniques such as grid search, random search, and Bayesian optimization to improve model performance.
- **Prevent Overfitting:** Implement strategies like cross-validation, regularization, and pruning to prevent overfitting and enhance model generalization.

5. Develop Advanced Machine Learning Solutions

- **Implement Ensemble Methods:** Utilize ensemble methods (bagging, boosting, and stacking) to build robust and accurate models.
- **Construct Neural Networks:** Design, implement, and train neural networks including convolutional neural networks (CNNs) and recurrent neural networks (RNNs) for complex tasks such as image and text processing.

6. Apply ML/AI to Real-world Problems

- **Work on Practical Projects:** Complete practical projects that involve end-to-end machine learning pipelines, including data collection, preprocessing, model training, and deployment.
- **Collaborate Effectively:** Work effectively in teams to tackle complex machine learning problems, demonstrating project management and teamwork skills.

7. Stay Informed About Emerging Trends

- **Explore Advanced Topics:** Gain insights into advanced topics like reinforcement learning, transfer learning, explainable AI (XAI), and automated machine learning (AutoML).
- **Ethical Awareness:** Understand and articulate the ethical implications of machine learning, including issues related to bias, fairness, and privacy.

8. Recognize the integration of ML/AI in advanced industries (e.g., quantum computing)

- **Present Findings:** Present technical findings and project outcomes clearly and effectively, both in written reports and oral presentations.
- **Explain Concepts:** Explain complex machine learning concepts and techniques to both technical and non-technical audiences.

EVALUATION AND GRADING CRITERIA

The overall evaluation would be carried out as follows:

- Assignments and homework: 30% (Weekly coding assignments)
- Quizzes and individual evaluation tests: 20% (Weekly quizzes)
- Mid-term Project: 30% (Comprehensive project with report and presentation)
- Participation: 20% (Active participation during the classes and visits)

The specific topics evaluated are based on the sessions content and the weight varies depending on the dedication.