Machine Learning and Deep Learning Roadmap

(Weekly Schedule)

Prepared by ChatGPT

Contents

1 Introduction 2

2 Prerequisites (Weeks 1–3) 2

3 Part I: Machine Learning (Scikit-Learn) (Weeks 4–10) 3

4 Part II: Deep Learning (TensorFlow + Keras) (Weeks 11–16) 5

5 Conclusion 6

1 Introduction

This roadmap is designed for a beginner to intermediate learner aiming to master:

- Machine Learning (with Scikit-Learn)
- Deep Learning (with TensorFlow & Keras)
- Required mathematical and programming foundations

It follows the structure of:

- Part I: Fundamentals of Machine Learning
- Part II: Neural Networks and Deep Learning

The full plan spans **16 weeks** (4 months), with weekly content and tasks.

2 Prerequisites (Weeks 1–3)

Week 1: Python Foundations

- Review Python basics: loops, functions, conditionals
- Master data structures: lists, tuples, dictionaries
- OOP essentials: classes, objects, methods
- Practice small Python exercises

Mini-Tasks:

- Write a calculator program
- Implement a simple class (e.g., Student class)

Week 2: NumPy and Matplotlib

- Array creation, indexing, slicing
- Broadcasting and vectorization
- Random number generation
- Creating plots: line, scatter, bar, histogram

Mini-Tasks:

- Create random 3x3 matrices
- Visualize random distributions

Week 3: Pandas + Math for ML

- DataFrames, reading CSVs
- Handling missing data
- Groupby, merging, filtering
- Math topics:
  - Linear algebra (vectors, dot product)
  - Calculus intuition (derivatives, gradients)
  - Probability & statistics basics

3 Part I: Machine Learning (Scikit-Learn) (Weeks 4–10)

Week 4: Introduction to Machine Learning

- What is ML?
- Supervised vs. Unsupervised
- Key terms: features, labels, models
- ML workflow overview

Mini-Project: Classify simple toy datasets (Iris, Digits)

Week 5: Data Preparation

- Train/test split
- Data cleaning & preprocessing
- One-hot encoding
- Scaling: StandardScaler, MinMaxScaler

- Pipelines

Task: Create a preprocessing pipeline for a CSV dataset.

Week 6: Linear, Polynomial & Logistic Regression

- Linear Regression theory + implementation
- Polynomial features & overfitting
- Logistic Regression for classification
- Cost function + optimization

Task: Predict house prices (regression).

Week 7: Classification Algorithms

- k-Nearest Neighbors
- Support Vector Machines
- Decision Trees

Task: Train SVM and KNN on MNIST subset.

Week 8: Ensemble Methods

- Random Forest
- Bagging
- Boosting (AdaBoost, GradientBoosting)
- Stacking

Project: Titanic/Loan classification using ensembles.

Week 9: Model Evaluation and Tuning

- Cross-validation (K-Fold)
- GridSearchCV / RandomizedSearchCV
- Understanding bias/variance

Task: Tune hyperparameters of a RandomForest.

Week 10: Unsupervised Learning

- PCA (dimensionality reduction)
- Clustering (K-Means, DBSCAN)
- Anomaly detection

Project: Customer segmentation using clustering.

4 Part II: Deep Learning (TensorFlow + Keras) (Weeks 11–16)

Week 11: Neural Network Foundations

- Neurons, layers, activations
- Loss functions
- Gradient descent & backpropagation

Task: Train your first MLP on MNIST.

Week 12: Building Models with Keras

- Sequential and Functional API
- Callbacks (EarlyStopping, LearningRateScheduler)
- Saving Loading models

Project: Digit classifier with high accuracy.

Week 13: Convolutional Neural Networks (CNNs)

- Convolution, pooling, filters
- Common architectures (VGG, ResNet)
- Transfer learning

Project: Image classification (CIFAR-10 or Dogs vs Cats)

Week 14: Recurrent Neural Networks

- RNN, LSTM, GRU
- Time-series prediction
- Text sequence modeling

Project: LSTM-based text generator.

Week 15: Autoencoders and GANs

- Autoencoders for feature learning
- Denoising autoencoders
- GAN architecture (Generator + Discriminator)

Task: Train a simple GAN for MNIST digits.

Week 16: Advanced Topics and Final Projects

- Regularization (Dropout, BatchNorm)
- Handling large datasets
- Reinforcement Learning (basics)
- Bayesian Deep Learning (uncertainty estimation)

Final Project Ideas:

- Image classifier with data augmentation
- Fraud/anomaly detection system
- GAN for generating characters
- Time-series forecasting

5 Conclusion

This 16-week plan takes you from Python essentials to building advanced neural networks. You may extend each section based on your pace and depth of interest.