

Weekly Report: Imtiaz Alpha AI

Week - 02 (April 28 - May 2)

Machine Learning Specialization:

Course 02: Advance Machine Learning Algorithm

Week 1: Introduction to Neural Networks

- Gained foundational understanding of neural networks and their analogy with the neurons in the human brain.
- Explored real-world applications of neural networks, including demand forecasting and image recognition.
- Studied the architecture of neural networks, such as input, hidden, and output layers, and how activation functions add non-linearity to enhance learning capabilities.
- Learned about forward propagation, the key mechanism by which predictions are made in a neural network.
- Practiced implementing neural networks using TensorFlow, including:
 - Creating and connecting multiple layers
 - Representing and preprocessing data
 - Building a complete model for digit classification using a multilayer perceptron (MLP)
- Explored matrix multiplication and its role in optimizing neural network computations using NumPy for efficient vectorized operations.

Week 2: TensorFlow Implementation and Activation Functions

- Implemented neural networks using TensorFlow, focusing on loss functions and their role in model learning.
- Developed intuition for loss vs cost function, and how minimization drives learning.
- Investigated popular activation functions (ReLU, Sigmoid, SoftMax), their mathematical behavior, and use cases:
 - Sigmoid for binary classification output layers
 - SoftMax for multi-class classification
- Understood advanced optimization techniques like Adam Optimizer, and its advantages over basic gradient descent for faster convergence and handling sparse gradients.

Week 3: Model Evaluation and Regularization

- Learned strategies to select appropriate machine learning models for real-world problems.
- Covered the importance of splitting data into Training, Validation, and Test sets to measure model generalization.

- Diagnosed bias-variance trade-offs and used learning curves to analyze underfitting vs overfitting scenarios.
 - Introduced techniques like regularization (L1/L2) to reduce overfitting by penalizing large weights.
 - Implemented regularization in TensorFlow models to observe its effect on performance.
 - Studied the iterative process of ML development, including model selection, hyperparameter tuning, and continuous evaluation.
 - Introduced to Transfer Learning: fine-tuning a pre-trained model to solve related tasks with less data and time.
 - Deep dive into evaluation metrics for imbalanced datasets:
 - Confusion Matrix, Precision, Recall, F1 Score
 - Learned how to adjust precision-recall trade-off to meet specific performance goals depending on the use-case (e.g., fraud detection vs medical diagnosis).
-

Task 2: Session on Object-Oriented Programming (OOP)

- Refreshed the foundational concepts of OOP: Class, Object, Constructor, Method, Attribute, and how they help in modular programming.
 - Learned the four core pillars of OOP:
 - Abstraction: Hiding complex implementation details
 - Encapsulation: Binding data and methods together
 - Inheritance: Reusing code through parent-child class relationships
 - Polymorphism: Ability to use objects in different forms
 - Prepared a presentation summarizing OOP concepts and shared insights during the session.
 - Compiled comprehensive documentation for internal reference and knowledge sharing among team members.
-

Task 3: Session on Git and GitHub

- Learned the basics of version control and collaborative development using Git and GitHub.
 - Explored:
 - Creating and managing repositories
 - Using essential Git commands (init, clone, add, commit, push, pull)
 - Navigating Git Bash to work locally with repositories
 - Practiced branching, resolving conflicts, and collaborating using pull requests and merges to simulate team workflows.
-

Task 4: Session on Agile Methodology

- Attended an in-depth session on Agile frameworks, particularly Scrum and Kanban, used for modern project management.

- Learned about Agile principles of iterative development, team collaboration, and customer-centric delivery.
- Studied key Scrum events:
 - Sprint Planning: Defining work for the sprint
 - Daily Stand-ups: Team progress updates
 - Sprint Review & Retrospective: Reviewing work and improving processes
- Gained clarity on Scrum roles:
 - Product Owner - prioritizes features
 - Scrum Master - facilitates team processes
 - Development Team - executes tasks collaboratively
- Understood how Agile encourages transparency, adaptability, and consistent delivery in dynamic project environments.