Deep Learning Course Overview

Week 1: Foundations and Neural Network Fundamentals

Day 1: Python for Data Science (Numpy, Pandas, Matplotlib)

- Python programming concepts for data manipulation.
- NumPy for efficient numerical operations and array computing.
- Pandas for data structures (DataFrames).
- Data visualization (Matplotlib/Seaborn)
- **Daily Assignment:** Download Kaggle dataset, perform data loading, cleaning, basic transformations using Pandas, and create at least three insightful visualizations

Day 2: Introduction to ML & DL

- ML: supervised, unsupervised, and reinforcement learning.
- ML to DL: understanding the motivations and advantages.
- **Daily Assignment:** Research and present a concise summary (1-2 pages) on the key differences between traditional ML algorithms and basic neural networks, outlining scenarios where Deep Learning offers significant advantages.

Day 3: Neural Networks Basics

- Fundamental building block: the artificial neuron (perceptron) and its operation.
- Study of activation functions (ReLU, Sigmoid, Tanh, SoftMax) and its impact on NN.
- Forward propagation process: how input data flows through the network to produce an output.
- Backpropagation algorithm for efficient gradient calculation and weight updates.
- Loss functions (Mean Squared Error, Cross-Entropy).
- **Daily Assignment:** Implement a simple single-layer perceptron from scratch in Python to solve a basic binary classification problem (e.g., XOR gate). Train, it using gradient descent and visualize its decision boundary.

Day 4: Tensorflow

- Introduction to TensorFlow 2.x and Keras API.
- Tensors: Fundamental data structures in TensorFlow and performing basic tensor operations.
- Building sequential models in Keras: defining layers, input shapes, and output configurations.
- Data loading and preprocessing techniques using the tf.data API.
- Model compilation and training workflow: compile (), fit (), and evaluate ().
- **Daily Assignment:** Build a feedforward neural network using TensorFlow to classify the MNIST handwritten digits dataset. Experiment with different numbers of layers, neurons, and activation functions.

Day 5: Training Neural Networks

- Various optimization algorithms: Stochastic Gradient Descent (SGD), Adam, RMSprop, etc.
- Strategies for learning rate scheduling: step decay, exponential decay, etc.

- Regularization techniques to prevent overfitting: L1/L2 regularization and Dropout.
- Batch Normalization: how it stabilizes training and accelerates convergence.
- Hyperparameter tuning: grid search, random search etc.
- **Daily Assignment:** Take the MNIST classifier from Day 4. Implement at least two regularization techniques and a learning rate scheduler. Analyze and report the impact on training loss, validation loss, and model generalization.

Week 2: Advanced Architectures and Applications

Day 6: Deep Neural Networks (DNNs)

- Designing architectures for deeper networks: challenges and solutions.
- Vanishing/exploding gradients problem in deep networks.
- ReLU and Batch Normalization for stable training in deep models.
- Residual connections (ResNet) and their role in very deep architectures.
- **Daily Assignment:** Implement a Deep Neural Network (e.g., 5-10 layers) for the Fashion MNIST dataset. Focus on proper weight initialization, using Batch Normalization layers, and monitoring training progress to ensure stable convergence.

Day 7: CNN & Computer Vision

- Understanding Convolutional layers: filters, strides, padding, and feature extraction.
- Understanding pooling layers (max pooling, average pooling) for dimensionality reduction.
- CNN architectures: LeNet, AlexNet, VGG, ResNet, and Inception networks.
- Transfer learning and fine-tuning pre-trained CNN models for new tasks.
- Applications of CNNs in computer vision: image classification, object detection, and segmentation.
- **Daily Assignment:** Choose a small image dataset (e.g., CIFAR-10 or a custom dataset of 2-3 classes). Implement a CNN from scratch and fine-tune a pre-trained model (e.g., VGG16 or ResNet50) using transfer learning. Compare the performance of both approaches.

Day 8: Recurrent Neural Networks, LSTM, GRU

- Basic Recurrent Neural Network (RNN) architecture and its challenges.
- Long Short-Term Memory (LSTM) networks for capturing long-range dependencies.
- Gated Recurrent Units (GRU) as a simpler alternative to LSTMs.
- Applications of RNNs, LSTMs, and GRUs in NLP.
- **Daily Assignment:** Implement an LSTM network for a simple sequence prediction task.

Day 9: Autoencoders & GANs

- Autoencoder architecture: encoder, latent space, and decoder.
- Types of autoencoders: denoising autoencoders, variational autoencoders (VAEs).
- Generative Adversarial Networks (GANs): the Generator and Discriminator components.
- **Daily Assignment:** Implement a basic Autoencoder using TensorFlow to compress and reconstruct images. Visualize the original and reconstructed outputs.

Day 10: Modern Deep Learning Trends

- Attention mechanisms and the Transformer architecture for sequence modeling.
- Reinforcement Learning (RL) and Deep Q-Networks (DQN) for decision-making agents.
- Ethical considerations in AI: bias, fairness, transparency, and responsible AI development.
- **Daily Assignment:** Choose one modern DL trend (e.g., Transformers, RL in gaming, GNNs for social networks). Research its core idea, key applications, and future potential. Prepare a short presentation or a detailed report summarizing your findings.

Week 3: Hackathon / Mini-Capstone Topics

1. **AI for Climate Change Adaptation:** Develop a Deep Learning model to predict localized climate impacts based on satellite imagery and historical data, or optimize resource allocation for disaster response.
2. **Personalized Health Diagnostics:** Create a system that uses Deep Learning to analyze multimodal patient data for early detection of specific diseases or to recommend personalized treatment plans.
3. **Generative AI for Digital Content Creation:** Build a model to generate realistic human faces, unique architectural designs, short musical pieces in a specific style, or even simple animated characters.
4. **Smart City Traffic Optimization:** Implement a Computer Vision system to analyze real-time traffic flow from surveillance cameras and use Deep Reinforcement Learning to optimize traffic light timings for reduced congestion and emissions.
5. **Financial Market Prediction & Anomaly Detection:** Use time-series Deep Learning models to predict stock price movements, cryptocurrency trends, or detect fraudulent transactions in financial datasets.
6. **AI-Powered Educational Feedback System:** Create a Deep Learning model that analyzes student responses (textual, code, or visual) to provide automated, personalized feedback, identify common misconceptions, or generate adaptive practice problems**.**
7. **AI for Wildlife Conservation:** Develop a Deep Learning system to identify endangered species from camera trap images or detect poaching activity patterns.
8. **AI-Based Sign Language Translator:** Build a real-time system using computer vision and sequence models (e.g., CNN + LSTM) to recognize sign language gestures from video input and translate them into text or speech.
9. **Deep Learning in Agriculture:** Implement a model that classifies plant diseases from leaf images or predicts crop yield using satellite imagery, weather data, and soil conditions.
10. **AI Legal Assistant for Document Analysis:** Develop a system that summarizes long legal documents, identifies key clauses, and flags missing legal elements using transformer-based models like BERT