Neural Network Fundamentals

Assignment 1: Basic - Neural Network for AI Engineer Availability Prediction

Objective: Implement a simple feedforward neural network to predict whether an AI engineer is available to take on a new project based on their current workload metrics.

Dataset: *engineer_availability.csv* with the following fields:

- current_projects: Number of active projects (0-5)
- weekly_hours_committed: Hours already committed (0-40)
- response_time_hours: Average response time in hours (1-48)
- availability: Target variable (1 = available, 0 = unavailable)

Task:

1. Build a simple neural network with a single hidden layer using the concepts from the reading material:
    - Create the appropriate input layer based on the features
    - Add one hidden layer with ReLU activation (choose an appropriate size)
    - Add an output layer with sigmoid activation
    - Use binary cross-entropy loss for this classification task
2. Train your model for at least 10 epochs and evaluate its accuracy
3. Explain how each component of your neural network relates to the concepts covered in the self-reading material

Deliverables:

- Python script implementing the neural network
- Brief explanation of your implementation choices and results

Assignment 2: Intermediate - AI Service Type Classifier

Objective: Build a neural network that can classify the type of AI service needed based on the description of a problem submitted to the One Hour AI Solution platform.

Dataset: *service_classification.csv* containing:

- problem_description: Text features (already converted to numerical embeddings, 10 columns)
- service_type: Target variable with categories:
    - 0: Data preprocessing
    - 1: Model development
    - 2: Debugging/troubleshooting
    - 3: Performance optimization

Task:

1. Create a neural network with multiple hidden layers:
    - Input layer appropriate for the embedding dimensions
    - Two hidden layers with appropriate activations (refer to the activation functions in the reading)
    - Output layer suitable for multi-class classification
    - Apply an appropriate loss function
2. Implement dropout between the hidden layers as covered in the reading material
3. Train the model and evaluate its performance
4. Experiment with at least two different activation functions from the reading material and compare their impact

Deliverables:

- Complete Python script for the neural network
- Comparison of model performance with different activation functions
- Analysis of how the network architecture affects classification accuracy

Assignment 3: Advanced - Neural Network for Service Time Estimation

Objective: Develop a regression neural network to estimate the time needed to complete an AI problem on the One Hour AI Solution platform.

Dataset: service_time.csv containing:

- complexity_score: Numerical rating (1-10)
- data_volume: Size of data in MB
- iterations_required: Estimated iterations needed
- documentation_level: Level of documentation needed (1-5)
- tech_stack_complexity: Complexity of technology stack (1-10)
- service_hours: Target variable - actual hours needed to complete

Task:

1. Design and implement a neural network for this regression problem:
    - Normalize the input features appropriately
    - Create a network with at least two hidden layers
    - Use appropriate activation functions for each layer
    - Implement batch normalization as discussed in the reading
    - Use a suitable loss function for regression
2. Implement a model training loop with:
    - Learning rate scheduling
    - Early stopping based on validation loss
3. Analyze which features contribute most to the prediction
4. Compare your neural network's performance against a simpler model (e.g., linear regression)

Deliverables:

- Complete Python implementation
- Analysis of model performance and feature importance
- Discussion of how concepts from the reading material (activation functions, regularization techniques, etc.) influenced your solution