Proposal for Neural Network Library Utilizing Fundamental Data Structures

Project Title:
Design and Implementation of a Neural Network Library Using Data Structures

Course:
Data Structures and Algorithms

Submitted by:
Muhammad Faisal Ishfaq

Roll No: 2023-CS-122

Section C

Objective

The goal of this project is to design and implement a simplified neural network library that incorporates key data structures such as **linked lists**, **stacks**, **queues**, and **graphs**. This library will allow users to define, train, and visualize a neural network while demonstrating the role of fundamental data structures in the efficient design of machine learning models.

Scope and Importance

Neural networks are fundamental to modern artificial intelligence and machine learning systems. However, their implementation is often abstracted by high-level libraries like TensorFlow and PyTorch, which obscure the underlying data structures and algorithms. This project will focus on building a neural network library from scratch to emphasize the importance of data structures in machine learning.

Key functionalities of the library will include:

1. **Layer Management:** Using a linked list to store and manage layers of the network.
2. **Weight and Bias Representation:** Using matrices to store weights and biases for efficient computation.
3. **Batch Management:** Using queues to handle mini-batches during training.
4. **Gradient Tracking:** Using stacks to manage gradients during backpropagation.

5. **Critical Layer Identification:** Using graphs to identify important layers and visualize network structures.

This project is an excellent integration of data structures and algorithms with real-world AI applications.

Proposed System Overview

1. Linked List for Layer Management

Functionality:

- Layers of the network (input, hidden, output) are represented as nodes in a linked list.
- Each node contains:
    - Layer type (e.g., input, dense, output).
    - Number of neurons.
    - Activation function.
    - Weight and bias matrices connecting to the next layer.

Benefits:

- Simplifies layer addition, removal, or modification.
- Facilitates forward and backward propagation by sequentially traversing the list.

---

2. Matrix for Weights and Biases

Functionality:

- Matrices represent connections (weights) and adjustments (biases) between neurons in consecutive layers.
- Operations such as matrix multiplication and addition are used in forward propagation, and gradient computation in backpropagation.

Benefits:

- Enables efficient computation and representation of neuron connections.
- Supports vectorized operations, which are central to neural network computations.

---

3. Queue for Batch Management

Functionality:

- Training data is processed in mini-batches stored in a queue, ensuring First-In-First-Out (FIFO) processing.
- Each batch contains input data and corresponding labels.

Benefits:

- Handles large datasets by breaking them into smaller, manageable chunks.
- Speeds up training by updating weights incrementally rather than processing the entire dataset at once.

---

4. Stack for Gradient Tracking

Functionality:

- Gradients calculated during backpropagation are stored in a stack (LIFO).
- Gradients are pushed onto the stack during computation and popped off during weight updates.

Benefits:

- Simplifies the reverse order required for gradient updates.
- Ensures efficient gradient tracking and application.

---

5. Graph for Critical Layer Identification

Functionality:

- The network is represented as a directed graph, where nodes are layers and edges represent connections (with weights).
- BFS or DFS algorithms are used to:
    - Analyze connectivity between layers.
    - Calculate the importance of each layer based on connectivity and edge weights.

Benefits:

- Identifies critical layers that influence training outcomes.
- Helps optimize training by prioritizing specific layers (e.g., adjusting learning rates).
- Enables visualization of the network structure for better interpretability.

Expected Outcomes

1. A functional neural network library that integrates data structures to perform training, backpropagation, and visualization.
2. A demonstration of how linked lists, stacks, queues, and graphs can be effectively utilized in neural networks.
3. A system that is both educational (highlighting data structure usage) and practical (implementing basic neural network functionality).
4. Improved understanding of how data structures impact performance and functionality in machine learning systems.

Note: This proposal is an approximate idea of what the project is about and what are its components. There can be little changes to the approach described above and more features can be added in it if time allows