Graph Neural Network-Based Personalized Workout Optimization System

Rameez Malik
North Carolina State University
ECE 592 / CSC 591: Advanced Topics in Deep Learning
remalik@ncsu.edu

1 Introduction

1.1 Goal and Task

The objective of this project is to design a Graph Neural Network (GNN)-based system capable of optimizing an individual's existing workout routine rather than recommending new exercises. The system analyzes user-recorded training data including exercises, repetitions, sets, rest intervals, intensity, and session frequency to determine how the user can modify their current workouts for improved muscle development and progressive overload. The model aims to predict optimal increases in load, adjustments in rest time, and changes in repetition range to maximize training efficiency while reducing risk of overtraining or injury.

1.2 Importance of the Problem

2 Dataset

This project utilizes both publicly available data and self-collected training logs.

Primary Dataset: Weightlifting Dataset (Kaggle) - includes exercises, weights, sets, reps, and timestamps.

Personal Data: Custom logs of sets, reps, rest times, and weekly progress will be incorporated to simulate longitudinal performance tracking.

In this setting:

- **Nodes:** represent exercises or training sessions.
- **Edges:** capture relationships among exercises (shared muscle groups, same-day combinations, or sequential temporal order).

- **Node features:** include training volume, rest time, perceived intensity, and recent performance changes.
- **Task:** predict optimal incremental changes in load or rest duration for subsequent sessions.
- **Metrics:** Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) for prediction accuracy; Precision@K for recommendation quality.

3 Machine Learning Techniques

3.1 Proposed Method

The initial prototype will implement a simplified GraphSAGE architecture using mean aggregation and fixed depth. This approach balances relational expressiveness with computational feasibility. Temporal and attention-based enhancements are proposed as future extensions and will be discussed in the final report. Given an input graph $G = (V, E)$, each node $v \in V$ represents an exercise, and edges $(u, v) \in E$ represent functional or temporal dependencies.

At layer k, node embeddings are updated as:

$$h_v^{(k)} = \sigma\left(W^{(k)} \cdot \text{AGGREGATE}\left(\{h_u^{(k-1)} : u \in \mathcal{N}(v)\}\right)\right)$$

where $h_v^{(k)}$ is the hidden representation of exercise v after k message-passing iterations, and $\mathcal{N}(v)$ denotes its neighbors.

While attention-based aggregation and adaptive depth mechanisms offer promising improvements for capturing nuanced exercise relationships, they are beyond the scope of the current prototype. These enhancements will be explored conceptually and reserved for future implementation based on feasibility of training.

3.2 Challenges

Key challenges include:

1. Constructing a coherent graph from heterogeneous sequential data.
2. Ensuring meaningful aggregation without over-smoothing in a fixed-depth model.

3.3 Justification of Model Choice

Graph Neural Networks are uniquely suited for relational learning tasks. A user's training efficiency depends on multi-exercise interactions rather than independent metrics; thus, a GNN can naturally model inter-exercise influence and session-to-session progression. Conventional MLPs or CNNs lack the capability to capture this structural dependency, making GNNs a more appropriate choice for this application.

4 Computational Resources

- **Platform:** Google Colab (free GPU and notebook integration).
- **Frameworks:** PyTorch Geometric for GNN implementation; NetworkX for graph construction; Scikit-Learn for metric evaluation.
- **Estimated runtime:** Approximately one hour for training and evaluation on a subset of 1000 samples.

5 Key Deliverables

Deliverable	**Description**	**Evaluation Metric**
Data Preprocessing	Graph/Model Construction	Node/Edge Statistics
Baseline Model	GraphSAGE/GCN performance benchmark	MAE, RMSE
Proposed Model	Simplified GraphSAGE with tuned hyperparameters	Improving baseline metrics
Ablation Study	Reserved for work if feasible (e.g., attention or depth control)	N/A
Visualization	Plot embeddings and sample optimization recommendations	Qualitative interpretability

Table 1: Key Deliverables and Evaluation Metrics