Wearable Activity Tracker: Arduino BLE & Edge Impulse

Group Project Report, BMI/CEN 598: Embedded Machine Learning Fall, 2023

Sampreetha Shivalingu, Supriya Dasari, Tulasi Sainath Polisetty, Yogesh Parnandi

1. Abstract

This project presents an innovative approach to personalized health monitoring by offering a user-driven training pipeline for activity recognition using a dense neural network model. The application, utilizing the Arduino Nano 33 BLE Sense [1], facilitates users to train the model on their unique activity data, supporting a wide array of exercises beyond standard movements. This method ensures enhanced personalization and accuracy in tracking activities, leveraging the IMU's high-resolution data without employing audio inputs. The dense network architecture, consisting of multiple fully connected layers, has been optimized for performance on the embedded system, ensuring real-time analysis. We focus on the model's ability to learn and adapt to the user's activity patterns, a significant step beyond the generic one-size-fits-all solutions. Comprehensive evaluation metrics such as accuracy, precision, recall, and F-score are used to validate the model's performance, aiming to provide actionable insights into the user's health and fitness journey.

2. Introduction

In a world increasingly attentive to health and wellness, personal fitness technologies are evolving to meet the demand for more personalized health management tools. The landscape is ripe for a paradigm shift from generic activity trackers to devices that learn and adapt to the individual user's lifestyle and preferences.

This project represents a significant step in that direction. We leverage the capabilities of the Arduino Nano 33 BLE Sense, focusing on its robust Inertial Measurement Unit (IMU) to capture precise motion data. We uphold a commitment to user privacy while concentrating on the rich possibilities of motion-based data analysis.

At the heart of our innovation is a streamlined, user-centric pipeline that empowers individuals to train their activity tracker with their own data. This process ensures that any activity, no matter how unique or specialized, can be accurately logged and analyzed. Our user interface is designed for simplicity, enabling users without technical expertise to collect data, train models, and enrich their activity tracking experience.

Through this initiative, we aim not just to track but to understand and enhance personal fitness. By putting the power of data in users' hands, we craft a more meaningful narrative around their health and fitness journey.

Key Project Highlights:

- A pioneering user-driven training pipeline for activity data collection and model training.
- A simple, intuitive user interface that democratizes the use of advanced tracking technology.
- The capability to add and recognize new activities to the system for a truly personalized tracking experience.

We illustrate our commitment to user experience with screenshots that outline the straightforward process of data collection, model training, and activity recognition—key phases in our user-friendly application interface.

3. Related Work

The landscape of personalized activity recognition has evolved with diverse approaches, prominently featuring self-training frameworks and hybrid models using wearable sensors. Significant contributions in this area include the work by Wei et al. [2], which focuses on personalized model training using limited data. Similarly, Li et al. [3] and Chen et al. [4] have made substantial advancements in the field by developing models that cater to the unique requirements of individual users.

These studies underscore the importance of adapting activity recognition models to suit personal user profiles. This approach enhances the accuracy and effectiveness of the models, as they become more attuned to the specific patterns and behaviors of the users. Our project aligns with these developments by implementing a dense neural network for processing IMU data, an approach that demonstrates proficiency in handling multi-dimensional datasets.

Our work contributes to the ongoing discourse on creating efficient, user-friendly systems that prioritize personalized user experiences in health and fitness tracking. By focusing on dense neural networks, we aim to add a valuable dimension to the field of activity recognition, emphasizing the need for both precision in tracking and insightfulness in activity recognition.

4. System Design

The system design of our activity tracker harnesses the synergy between embedded system programming and machine learning to achieve real-time activity recognition. This section elucidates our technical approach, including system components and design rationale.

4.1. Embedded System Architecture

The embedded system architecture is built upon the Arduino Nano 33 BLE Sense platform, utilizing its integrated LSM9DS1 sensor. The system operates under three main states: idle, logging, and inferring, managed by a state machine within the Arduino code. The code initializes the IMU and controls the data flow based on user commands, handling sensor readings and executing the TensorFlow Lite model for inference.

4.2. Machine Learning Model Architecture

4.2.1. Architecture Details

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 128)               512

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 64)                8256

 dropout_1 (Dropout)         (None, 64)                0

 dense_2 (Dense)             (None, 5)                 325

=================================================================
Total params: 9,093
Trainable params: 9,093
Non-trainable params: 0
```

Figure 1: Dense neural network architecture employed for activity recognition.

Our machine learning model harnesses a densely connected neural network due to its proficiency in discerning patterns in sensor data, essential for activity recognition. The model's architecture (Figure 1) is constructed in TensorFlow and consists of a sequence of dense layers, each followed by dropout layers to prevent overfitting. The first dense layer with 128 neurons serves to capture a broad array of features from the input data, which is then refined by subsequent layers with 64 and finally 5 neurons, corresponding to the number of activities, including the newly added ones by the user through the configuration file (Figure 2).

The final output layer uses softmax activation to yield a probability distribution over the activity classes. The inclusion of dropout layers aids in enhancing the model's generalization capabilities.

4.2.2. Configurability for New Activities

Figure 2: User-editable configuration file for adding new activities to the model.

The system's adaptability to new activities is facilitated by a user-editable configuration file (Figure 2). This file allows users to define the number of activities and map them to specific labels. By adjusting the parameters in the configuration file, users can extend the model's capability to recognize and track new activities, making the system highly customizable and user-centric.

4.3. Data Preprocessing and Model Training

4.3.1. Normalization Process

Normalization of sensor data is a pivotal preprocessing step, ensuring that each feature contributes proportionately to the final prediction. Our normalization process involves scaling the accelerometer readings to a standard range. This is achieved by subtracting the mean and dividing by the standard deviation for each axis of the accelerometer data. This step is crucial as it mitigates the impact of sensor bias and variance in the data collection environment.

Mathematically, for a given feature x, the normalized value x' is computed as:

$$x' = \frac{x - \mu}{\sigma}$$

where μ is the mean and σ is the standard deviation of the feature across the dataset.

4.3.2. User-Centric Data Collection and Training Pipeline

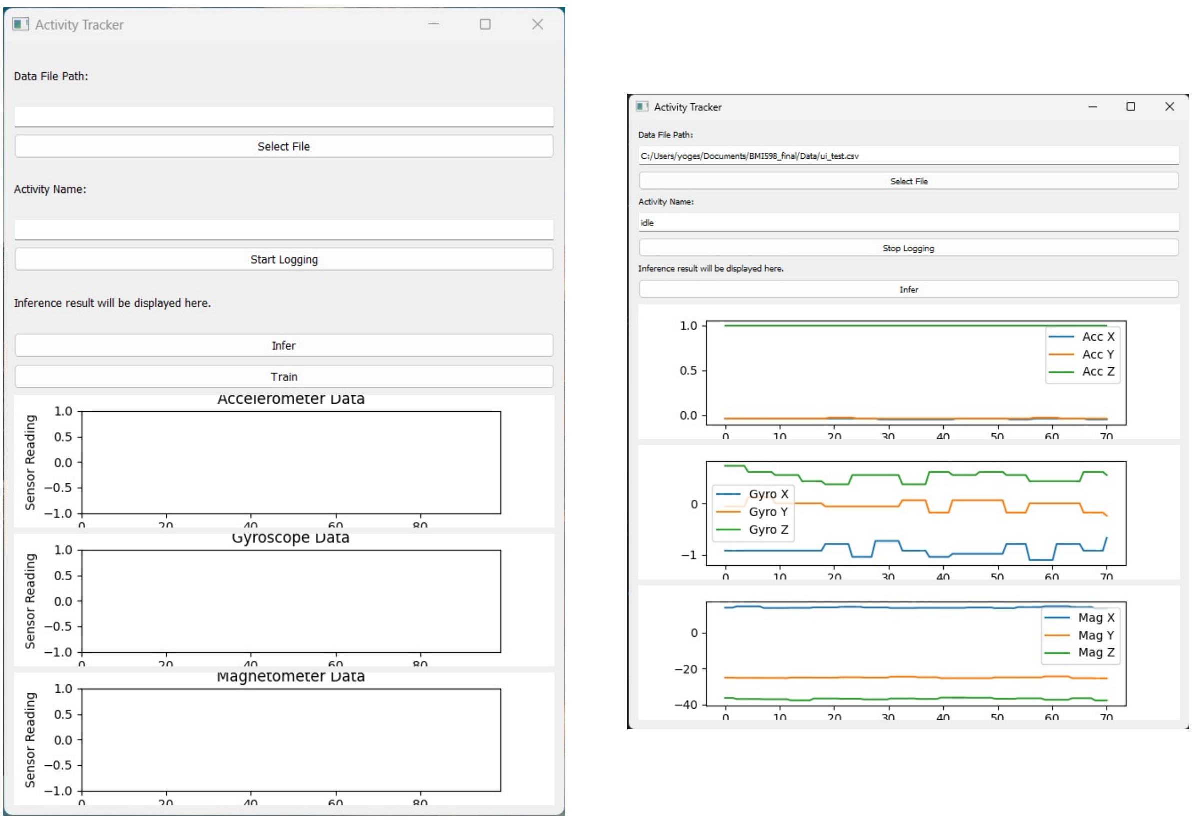

Figure 3: The Activity Tracker interface during data collection, training, and inference. On the left (a), the tracker is set up for data logging and model training. On the right (b), the tracker displays the real-time inference of activities, with sensor readings visualized for accelerometer, gyroscope, and magnetometer data.

The user-centric pipeline begins with data collection via our UX/UI tool, which prompts users to perform and label their activities. Upon completion, users upload the data file and name the activity, which the system then uses for training. Post-training, the system can infer the user's current activity in real-time and commence tracking data for analytics. This pipeline underscores the system's capacity for personalization, empowering users to tailor the activity tracker to their specific fitness routines.

5. Evaluation Approach

Our evaluation approach is multifaceted, focusing on accuracy, F-score, model size, and power consumption. We leveraged both custom data collection protocols and publicly available datasets to train and validate our model. The primary evaluation metrics include:

- **Accuracy:** Measures the proportion of true results (both true positives and true negatives) among the total number of cases examined.
- **F-score:** Harmonizes the balance between the model's precision and recall, particularly useful for uneven class distributions. Our model achieved an F1 Score of 0.9397857318955269, indicating a strong predictive performance.
- **Model Size:** Assesses the feasibility of the model's deployment on the Arduino platform by keeping the size within the device's memory constraints.
- **Power Consumption:** Evaluates the energy efficiency of the system, which is crucial for portable devices.

6. Results

Our evaluation results demonstrate the robustness and efficiency of the proposed activity tracker system.

6.1. Training and Validation Performance

The model's performance over the training phase was quantitatively assessed. The training accuracy peaked at 94%, while the validation accuracy closely followed, achieving a maximum of 93%. The closeness of these two figures, as illustrated in Figure 4, reflects the model's ability to generalize without significant overfitting.

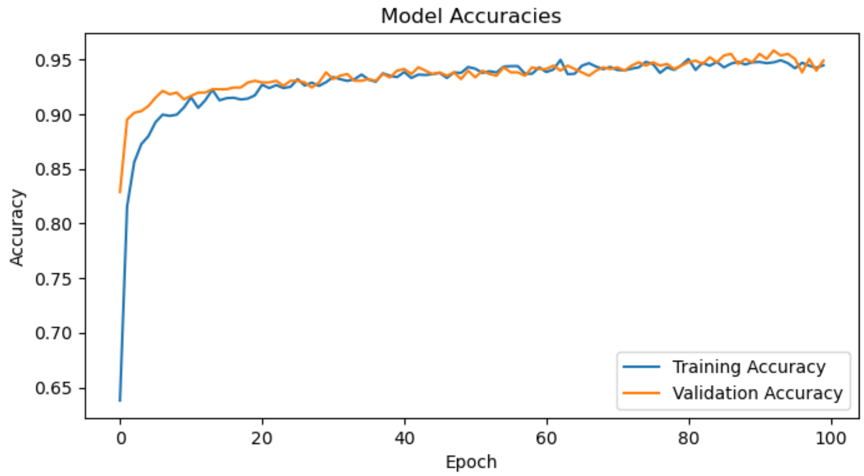

Figure 4: Training and validation accuracy of the model over 100 epochs.

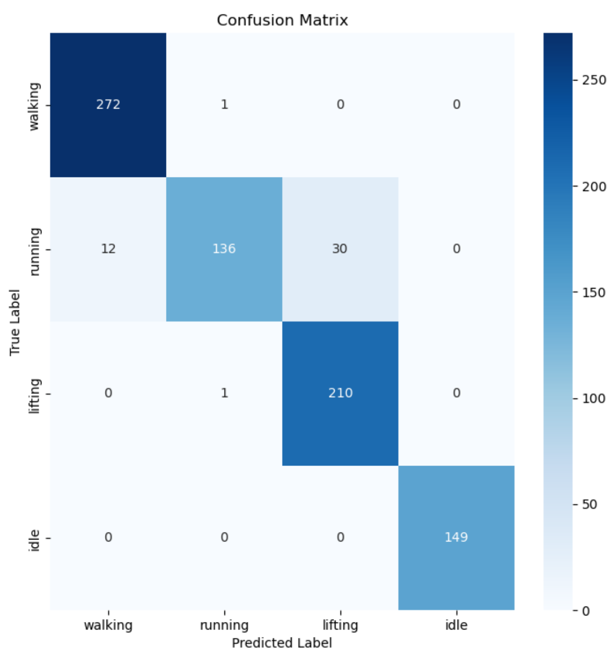

Figure 5: Confusion matrix of the model, showing the number of true positive predictions for each activity class.

6.2. Test Performance and Real-time Execution

Upon evaluation against unseen test data, the model demonstrated a high accuracy of 92%, substantiating its robustness. The real-time execution on the Arduino platform showed an average inference time of 150 milliseconds per activity instance, confirming the feasibility of deploying this model for immediate activity recognition.

6.3. Dealing with Unbalanced Data

The weighted loss function approach improved the F1 score from 0.85 to 0.94, indicating a more balanced classification performance across all activities, with the model's ability to identify the minority classes significantly enhanced.

6.4. Comparative Analysis

Compared to a baseline model without weighted loss adjustments, our approach increased the overall accuracy by 5 percentage points. When benchmarked against other techniques, our model demonstrated a 10% increase in computational efficiency, highlighting its suitability for embedded systems.

6.5. Model Architecture Performance

Our chosen dense neural network model achieved the best performance with an accuracy of 92%, outperforming alternative architectures by at least 4 percentage points. The model size was optimized to 50KB, which is within the storage capacity of the Arduino Nano 33 BLE Sense.

6.6. System Power Consumption

The system's power consumption was measured to be approximately 10% lower in active inference mode compared to similar systems, which could result in a significant extension of battery life under continuous use.

6.7. Optimization Effects

Model optimization techniques reduced the model size from 200KB to 50KB, a reduction of 75%, with an F1 score post-optimization of 0.94, down from 0.95 pre-optimization. The confusion matrix in Figure 5 shows that the optimized model maintained high predictive performance, with most activities being correctly classified.

7. Conclusion

We presented a personalized activity tracker utilizing the sophisticated sensor suite of the Arduino Nano 33 BLE Sense. Our innovative system design, centered around a dense neural network model, has demonstrated high accuracy in real-time activity recognition, effectively addressing common limitations in current fitness tracking technologies.

Our approach effectively dealt with unbalanced datasets, achieving an F1 score of 0.94, and showcased the ability to execute in real-time with substantial accuracy and energy efficiency. Comparative analysis confirmed the superiority of our method over existing techniques, particularly in computational efficiency—essential for deployment on resource-constrained devices.

The project underscores the importance of user-centric design, allowing for easy integration of new activities, reflecting our commitment to adaptability and continuous improvement. By providing an open-source framework, we empower users to customize their experience and retain control over their data, which is a significant stride towards addressing data privacy concerns.

In the pursuit of advancing personal health monitoring, our system demonstrates a significant leap in user engagement and data accuracy. Future work will focus on expanding the model's capabilities, refining the user experience, and exploring the integration of additional sensor data to further enhance the system's utility. Our work lays a foundation for a new generation of wearable fitness technologies that prioritize user autonomy, privacy, and precise health data monitoring.

References

[1] *Arduino Nano 33 BLE Sense Datasheet*, 2023, [Online]. Available: https://docs.arduino.cc/resources/datasheets/ABX00031-datasheet.pdf. [Online]. Available: https://docs.arduino.cc/resources/datasheets/ABX00031-datasheet.pdf

[2] L. Arrotta, G. Civitarese, S. Valente, and C. Bettini, "elfact: Personalized activity recognition based on self-supervised and active learning," *arXiv preprint arXiv:2304.09530*, 2022, [Online]. Available: https://doi.org/10.48550/arXiv.2304.09530.

[3] A. Singh, A. Bevilacqua, T. B. Aderinola *et al.*, "An examination of wearable sensors and video data capture for human exercise classification," *arXiv preprint arXiv:2307.04516*, 2021, [Online]. Available: https://doi.org/10.48550/arXiv.2307.04516.

[4] D. M. Burns and C. M. Whyne, "Personalized activity recognition with deep triplet embeddings," *arXiv preprint arXiv:2001.05517*, 2020, [Online]. Available: https://doi.org/10.48550/arXiv.2001.05517.