EMG-Based Gesture Recognition and Character Prediction at Meta

Nihaar Shah

Project Overview

This project focused on developing and optimizing machine learning models for interpreting electromyography (EMG) and inertial measurement unit (IMU) signals to enable intuitive human-computer interaction. The work involved training deep learning models to categorize gestures and predict handwritten characters from biosignal data, with significant emphasis on improving training infrastructure efficiency and data pipeline flexibility.

Background

Electromyography (EMG) measures electrical activity produced by skeletal muscles, providing rich signals that can be decoded to understand user intent and motor actions. Combined with **IMU sensors** (accelerometers and gyroscopes), these biosignals enable the development of next-generation input devices for augmented reality, virtual reality, and assistive technologies.

The challenge lies in processing high-dimensional, noisy time-series data in real-time while maintaining accuracy across diverse users and contexts. This requires sophisticated machine learning models trained on large-scale datasets with efficient infrastructure to support rapid iteration and experimentation.

Technical Contributions

1. Model Development for Gesture Recognition and Character Prediction

Developed and trained deep learning models to perform two key tasks:

Gesture Categorization: Classification of discrete hand gestures and movements from EMG/IMU signal streams, enabling recognition of user actions such as pinching, swiping, pointing, and other interaction primitives.

Handwritten Character Prediction: Decoding continuous EMG and IMU signals to predict handwritten characters as users "write" in the air or on surfaces, enabling text input without physical keyboards.

The models leveraged temporal convolutional networks and recurrent architectures to capture the sequential dependencies in biosignal data, with careful attention to handling variable-length sequences and multi-modal sensor fusion.

---

2. Distributed Training Efficiency Optimization

Significantly improved the efficiency of distributed model training through systematic performance analysis and optimization:

Perfetto Trace Analysis: Utilized Perfetto, a production-grade tracing framework, to profile distributed training workflows and identify performance bottlenecks. This involved:

- Capturing detailed traces of GPU utilization, data loading, communication overhead, and computation time
- Analyzing timeline visualizations to detect idle GPU time, synchronization barriers, and inefficient data transfer patterns
- Quantifying the impact of various bottlenecks on overall training throughput

Dataloader Optimizations: Implemented targeted improvements to the data loading pipeline based on trace analysis findings:

- Optimized prefetching strategies to ensure GPUs remain saturated with data

- Tuned worker processes and batch preparation to minimize CPU-GPU transfer latency
- Reduced data loading overhead through efficient caching and preprocessing strategies
- Improved multi-worker coordination to eliminate stalls and maximize parallelism

These optimizations resulted in measurable improvements in training throughput, reducing time-to-convergence and enabling faster experimentation cycles for researchers.

---

3. Data Pipeline Architecture and Researcher Flexibility

Redesigned core components of the data pipeline to improve both efficiency and usability for research teams:

Custom Collate Functions: Designed flexible collate functions that handle complex data batching requirements:

- **Masking Support**: Implemented sophisticated masking mechanisms to handle variable-length sequences, missing data, and attention-based architectures. This enabled models to process sequences of different lengths within the same batch without padding overhead.
- **Label Filtering**: Created configurable filtering logic to selectively include or exclude specific labels, gestures, or signal segments based on experimental requirements. This allowed researchers to quickly test hypotheses on subsets of data without modifying upstream data generation.
- **Signal Filtering**: Developed signal-level filtering capabilities to apply frequency-domain filters, artifact removal, or feature extraction on-the-fly during training, reducing preprocessing overhead and storage requirements.

Researcher Flexibility: The modular design of collate functions enabled researchers to:

- Rapidly prototype new training configurations without modifying core data loading code
- Experiment with different masking strategies for self-supervised learning

- Test model robustness by selectively filtering challenging samples or signal conditions
- Compose multiple filtering and transformation operations declaratively

This architectural improvement accelerated the research workflow by reducing the friction between experimental ideas and implementation.

Technical Stack

- **Deep Learning Frameworks**: PyTorch for model development and training
- **Distributed Training**: Multi-GPU and multi-node training with efficient data parallelism
- **Performance Profiling**: Perfetto for distributed system tracing and bottleneck analysis
- **Signal Processing**: Time-series analysis, filtering, and feature extraction for EMG/IMU data
- **Data Pipeline**: Custom PyTorch DataLoader implementations with advanced collation logic

---

Impact

The work contributed to advancing Meta's capabilities in biosignal-based interaction technologies:

1. **Training Efficiency**: Reduced training time through systematic optimization, enabling faster iteration and experimentation
2. **Model Performance**: Improved gesture recognition and character prediction accuracy through better data handling and model architectures
3. **Research Velocity**: Enhanced researcher productivity by providing flexible, efficient data pipeline tools that reduced implementation overhead
4. **Scalability**: Established infrastructure patterns that scale to larger datasets and more complex models

---

Key Challenges Addressed

High-Dimensional Temporal Data: EMG and IMU signals are high-frequency, multi-channel time series that require careful handling to preserve temporal dependencies while managing computational complexity.

Distributed Training Bottlenecks: Identified and resolved performance issues in distributed training workflows, including data loading stalls, communication overhead, and GPU underutilization.

Variable-Length Sequences: Developed efficient batching strategies for sequences of varying lengths without excessive padding or memory waste.

Research Flexibility vs. Performance: Balanced the need for flexible experimentation tools with the requirement for high-performance training pipelines.

Skills Demonstrated

- Deep learning for time-series classification and sequence prediction
- Distributed systems optimization and performance engineering
- Production ML infrastructure development
- Signal processing and biosensor data analysis
- PyTorch ecosystem expertise (DataLoader, collate functions, distributed training)
- Performance profiling and trace analysis
- Research infrastructure design for ML teams

Future Directions

This work establishes foundations for continued advancement in biosignal-based interaction:

- **Real-time Inference**: Optimizing models for low-latency on-device inference
- **User Adaptation**: Developing personalization techniques to adapt models to individual users

- **Multimodal Fusion**: Exploring deeper integration of EMG, IMU, and other sensor modalities
- **Robustness**: Improving model performance under challenging conditions (motion artifacts, electrode placement variation)

Conclusion

This project demonstrates the intersection of machine learning research, systems engineering, and human-computer interaction. By combining model development with infrastructure optimization and thoughtful API design, the work enabled more efficient training workflows and accelerated research progress in biosignal-based interaction technologies at Meta.

The emphasis on both algorithmic innovation and engineering excellence reflects the multidisciplinary nature of modern ML research, where success requires not only developing effective models but also building the infrastructure that enables rapid experimentation and deployment at scale.