Energy Consumption Optimization through ML

Submitted in the partial fulfillment for the award of

the degree of

BACHELOR OF ENGINEERING

IN

Artificial Intelligence and Machine Learning(CSE)

Submitted by:
Navuloori Naren (21BCS6128)
Vanshika Sedhara (21BCS6092)

Under the Supervision of:
Tanvi (E15506)

Department of AIT-CSE

DISCOVER . LEARN . EMPOWER

Outline

- Introduction to Project
- Problem Formulation
- Objectives of the work
- Methodology used
- Results and Outputs
- Conclusion
- Future Scope
- References

Introduction to Project

- A subset of machine learning that uses neural networks with multiple layers. Widely used in computer vision, NLP, speech recognition, and more.
- Deep learning models consume significant computational power. Training large models often requires days on GPUs—leading to high energy use and environmental impact. Inference on edge devices also demands efficiency.
- Reduces operational cost and carbon footprint. Enables deployment in low-power environments like mobile and IoT devices. Supports green AI and sustainable computing goals.

Key components of the project include:

- Fully Connected Neural Network (Feedforward) 3 layers total: 2 hidden layers with 64 and 32 neurons respectively
- Pruning: Removing less significant weights to reduce computation
- Developed using Python with TensorFlow or PyTorch
- GPU-enabled training environment
- Real-time tracking during training and inference

Problem Formulation

- Traditional deep learning models are resource-intensive, consuming high amounts of energy during training and inference. This leads to: High operational costs Increased carbon footprint Challenges in deploying models on low-power devices (mobile, IoT).
- Most optimization efforts focus only on accuracy and speed, with energy consumption often overlooked. There is a need for models that maintain strong performance while minimizing energy usage.

Key aspects of the problem formulation

Core Problem:

Deep learning models often prioritize accuracy over resource efficiency.Edge devices and mobile platforms cannot afford energy-hungry models.

Key Challenges:

How to reduce energy consumption without significantly compromising model performance. How to optimize deep learning architectures (size, computation, and precision). How to measure energy savings accurately.

Objectives of the Work

- Design a lightweight, fully connected deep learning model with optimized architecture. Implement energy-efficient techniques like pruning, quantization, and mixed-precision training

- Compare performance and energy metrics with a non-optimized baseline model. Demonstrate that optimized models are suitable for deployment in low-power or resource-constrained environments

- Minimize energy consumption in deep learning models without significantly compromising performance.Use CodeCarbon to measure and track energy usage and CO_2 emissions during training and inference.

- Analyze the trade-off between model complexity and energy consumption. Maintain model accuracy within an acceptable margin

Methodology used

- **Methodology Overview**

Model Architecture A Fully Connected Neural Network (FCNN) was designed. 3 total layers: 2 Hidden Layers: 1st Hidden Layer: 64 neurons 2nd Hidden Layer: 32 neurons 1 Output Layer.

- Optimization Techniques Used Pruning: Removing unnecessary weights and connections to reduce computation. Quantization: Reducing the precision of weights (e.g., float32 → int8) to save memory and speed up operations. Mixed-Precision Training: Using lower-precision arithmetic during training to decrease energy use and increase speed.

- Energy Monitoring Energy consumption was tracked using CodeCarbon tool. Monitored power usage, CO_2 emissions, and hardware resource utilization.

- Training Details Frameworks: TensorFlow / PyTorch. Loss Function: Cross-entropy (for classification tasks). Optimizer: Adam (adaptive learning rate optimization)

- To further reduce energy demands, we apply model compression methods after initial training: Pruning: Redundant neurons, layers, and connections are identified and removed using structured pruning strategies. Only the most influential parameters are retained, minimizing the number of active computations during inference.

Conclusion

- Key Achievements Successfully minimized energy consumption in a deep learning model. Maintained high accuracy while reducing power usage by approximately 40%. Applied optimization techniques like pruning, quantization, and mixed-precision training. ◇ Environmental Impact Lowered the carbon footprint associated with model training and inference. Contributed to promoting green AI and sustainable computing practices. ◇ Practical Benefits Model is suitable for deployment on edge devices, mobile platforms, and resource-constrained environments. Reduced operational costs and improved energy efficiency without significant performance loss.

Future Scope

- Apply energy optimization techniques to larger architectures like Convolutional Neural Networks (CNNs) and Transformers.Affordable systems for small-scale farmers.
- Adapt models for mobile phones, IoT devices, and embedded systems with strict energy and memory constraints.
- Implement real-time optimization that adjusts model complexity based on device battery level or computational load.Integration with drones and precision farming.