POLITECNICO DI TORINO

Department of Electronics and Telecommunications

Politecnico
di Torino

Master's Degree Thesis in ICT for Smart Societies

Energy Sustainability Analysis of Machine Learning Algorithms

Supervisors	Student
Prof. Michela MEO	Jun YIN
Prof. Greta VALLERO	

Academic Year 2023/24

Abstract

AI applications are pervasive today and the significant environmental concerns has been posed by the energy consumption of large language models (LLMs), such as OpenAI's GPT-4, during their training processes. By developing and employing specialized tools, this study quantifies GPU energy consumption across various stages of deep learning model training, highlighting considerable energy usage during forward propagation, particularly in convolutional layers. The study systematically analyzes how factors like hardware variations, dataset characteristics, batch size, and model architecture influence energy consumption. From these insights, predictive models have been constructed to estimate energy usage, providing forecasts that consider both the models' multiply-accumulate operations (MACs) and their configuration without incorporating accuracy. Furthermore, the thesis explores the integration of renewable energy sources solar power and battery systems with the electrical grid to enhance the energy efficiency of GPU operations. An innovative energy dispatch model, tailored to optimize energy utilization and reduce costs, has been tested under various operational conditions. The findings advocate for more sustainable AI training practices by demonstrating substantial potential for cost reductions and energy savings.

Keywords: AI energy consumption, deep learning models, GPU efficiency, predictive energy model, energy dispatch model

Acknowledgements

I extend my deepest appreciation to my supervisors, Professor MICHELA MEO and Professor GRETA VALLERO, for their invaluable guidance and support throughout my research. Their mentorship has significantly enriched both my academic and personal growth.

I am also thankful to Politecnico di Torino and the ICT4SS platform, which provided a dynamic environment that greatly enhanced my skills and confidence. Special thanks go to the faculty and fellow students for their encouragement and support.

A heartfelt thanks to my wife, whose unwavering support and positive spirit have been pivotal throughout my academic journey. Her companionship and understanding have been my greatest motivations.

Lastly, I am grateful to all who have supported me in this journey, enabling me to pursue and achieve my academic goals.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

AI applications are ubiquitous in today's technological landscape, with most innovative applications embedding AI solutions to enhance their functionality. The release of the ChatGPT-4 model by OpenAI has brought large language models (LLMs) into the spotlight, driving significant attention and research in this area. These models, while powerful, require substantial computational resources, leading to considerable energy consumption. For instance, training OpenAI's GPT-3 model once demands approximately 1287 MWh of energy [**1**]. This substantial energy requirement underscores the importance of addressing the environmental impact of AI technologies.

Recent recommendations from the European Commission highlight the urgency of governing and reducing CO_2 emissions from software assets [**2**]. This directive is particularly pertinent to AI solutions, which are increasingly integrated into various applications and systems. Consequently, there is a critical need to develop tools and methodologies capable of providing reliable and realistic estimations of the energy consumption associated with AI technologies and their resultant environmental impacts. This necessity forms the impetus for our research, which seeks to contribute to the emerging field of green AI by focusing on energy-efficient AI model training and operation.

1.2 Objective

The primary objective of this research is to quantify and investigate the environmental impact of AI training through the development and application of specialized tools. These ad hoc tools represent a preliminary but crucial step towards the

implementation of green AI approaches, aimed at minimizing the electricity cost of AI technologies.

This master's thesis is structured into two main parts. The first part is to find out the key performance indicators (KPIs) that can affect energy consumption during the training of deep learning models. Then, considering two different preconditions, an energy consumption prediction model was proposed, and the proposed model was validated.

Based on the insights gained in the first part, the second part focuses on constructing a mathematical model designed to analysis the economic cost during training centralized AI algorithms. This model will elucidate the relationships between the power limit and running performance. Additionally, the model incorporates a dynamic allocation strategy for computational resources, which adjusts the resources dedicated to AI algorithm training according to the energy production from locally installed photovoltaic (PV) panels. This dynamic allocation aims to maximize the use of clean energy during the training process of AI models, thereby aligning the energy requirements of AI model training with environmentally sustainable practices which leads to the minimum cost.

In summary, this research aims to provide a comprehensive framework for assessing and reducing the economic cost of AI training. By developing reliable estimation tools and a dynamic energy allocation model, we seek to advance the field of green AI and promote the adoption of more sustainable AI practices.

1.3 Literature Review

Literature[**3**] defines the concept of GreenAI and introduces various methods to reduce carbon emissions during the training of AI models. In terms of energy consumption calculation, Literature [**4**] proposes a method for estimating the carbon footprint of Generative AI throughout its lifecycle. Literature[**5**] and literature[**6**] primarily focus on energy consumption and carbon emission estimation models for different models and hardware facilities in local and cloud computing clusters, respectively.

Regarding specific model energy consumption estimation, literature[**7**] analyzes the performance of different GPUs and CPUs in deep model training scenarios, verifying that GPUs outperform CPUs. Literature[**8**] analyzes the operational characteristics of models on different hardware devices. Literature[**9**] provides a

mathematical model of energy consumption for GPUs based on an analysis of GPU architecture at a low level. Literature[**10**] investigates the DVFS (Dynamic Voltage and Frequency Scaling) technology for GPUs and identifies the performance of GPUs under different operating conditions through DVFS adjustments. Literature[**11**] collects energy consumption data for different models on a single dataset and proposes a method for predicting energy consumption based on different model structures.

Considering data center operations, Literature[**12**] provides a detailed analysis and modeling of data center energy consumption, from subsystems such as servers, processors, memory, and storage to the entire data center. Literature[**13**] introduces a data center power supply strategy that considers multiple renewable energy sources and finds the cost-minimizing power supply mode through the optimal allocation of energy from different renewable sources. Literature[**14**] discusses an optimization algorithm for sizing the electric infrastructure using combinations of standard microgrid approaches and quantifying the level of grid utilization when data centers consume/export electricity from/to the grid to determine the required effort from the grid operator. Literature[**15**] achieves the goal of minimizing both the energy costs and overall carbon emissions of data center operations through an optimization algorithm that combines various renewable energy sources and grid power. This is accomplished by dynamically allocating arriving jobs to different clusters to minimize the overall energy cost of the data center.

Regarding the characteristics of GPUs during deep learning model training, Literature[**16**] identifies the operating speeds of GPUs under different power limits. Based on this, an optimization algorithm is proposed to adjust the power limits and batch size settings in subsequent model training to ensure that overall energy consumption remains as low as possible.

Considering all the referenced literature, this thesis firstly involved modeling multiple deep learning models and selecting appropriate measurement tools to collect energy consumption data for various parts of the training process. This provided energy consumption characteristics for each part and overall model energy data. Subsequently, the collected energy data was analyzed in relation to model structure, dataset size, and training parameter settings. Multiple potential factors influencing the energy consumption of deep learning model training were identified and analyzed using the collected data.

Next, based on the available data, energy consumption predictions were made for the models, both considering and not considering accuracy. The proposed models were then validated. Finally, a mathematical model was constructed to simulate the

power supply scenario involving solar panels and storage batteries, combined with grid energy, to power a GPU-centric computation center. Optimization algorithms were applied to identify the minimal cost and relevant factors influencing the cost of GPU training tasks.

1.4 Overall Structure

- **Chapter 2** compares different energy consumption testing tools and analyzes the energy consumption of different layers in the forward propagation process of a deep learning network based on the selected tool to have an deeper understanding of the model characteristics of deep learning.
- **Chapter 3** considers the prediction methods of energy consumption, firstly, the energy consumption data is collected by building several different deep learning models, then based on the collected data, two prediction methods are considered and the prediction methods are verified.
- **Chapter 4** shows the simulation results of the different models and the verification of the prediction methods.
- **Chapter 5** firstly analyzes the relationship between the energy consumption of the hardware and the operation speed. Then based on this, a model is built to test the effect of different training durations and corresponding training speeds on the final energy consumption and corresponding overheads in the case of multiple forms of energy supply.
- **Chapter 6** summarize the results and present the future works.

Chapter 2

Methodology

2.1 Hardware and Measurement Tools

This section introduces the hardware devices used in the thesis and the tools employed for measuring energy consumption.

2.1.1 Hardware

The hardware devices used in this thesis include a RTX3060 GPU and a RTX4090 GPU. The 3060 GPU has different power limits under different GPU drivers, which are 80W(under driver version 545) and 95W(under driver version 525) respectively. The 4090 GPU has a power limit of 450W(under driver version 545).

2.1.2 Measurement tools

For the energy consumption measurement tools, two tools at first were taken into consideration: codecarbon [**17**] and nvidia-smi [**18**].

Codecarbon is a lightweight python pip package, it can be used to estimate the energy consumption as well as the carbon emission during training deep learning models.

However, during the testing, it was found that codecarbon calculates energy consumption using an estimation method. Specifically, it assigns a fixed corresponding energy consumption value to each device and then multiplies it by the corresponding training time to obtain an approximate estimate. In our hardware tests, the power consumption of the 3060 GPU in codecarbon dataset was theoretically set

to a maximum of 130W, which differs from the actual power consumption 80W. Considering that this thesis requires more precise energy consumption data for each epoch, codecarbon is not very suitable for this purpose.

The NVIDIA System Management Interface (nvidia-smi) is a command line utility, based on top of the NVIDIA Management Library (NVML), intended to aid in the management and monitoring of NVIDIA GPU devices.It can record the power consumption per second during the training of deep learning models. By collecting energy consumption data of each epoch in this way, more accurate data can be obtained for this thesis compared to codecarbon.

2.2 Calculation Method of Energy Consumption Data

In this study, all subsequent energy consumption records are obtained using nvidia-smi, which records the average per-second energy consumption during each epoch of the training process of each round. For the energy consumption calculation of each part, the total energy consumption of each epoch is divided by the runtime of each epoch, resulting in the average energy consumption per second within an epoch. By multiplying the energy consumption per second by the corresponding runtime of a given part, the energy consumption for the part can be obtained. By calculating energy consumption in this way, the runtime can be equivalent to the energy consumption of each part during the training process.

2.3 Energy Consumption of Different Layers

This section primarily uses the testing tools mentioned earlier to analyze the time consumption of different layers during the forward propagation process of a deep learning model.

To test the energy consumption of different layers, which is the time consumed by different layers during the training process in thesis, we selected the AlexNet model [**19**], the architecture is shown in Figure 2.1. The dataset selected is FashionMNIST [**20**], in which the size of the image is 1*28*28. In this dataset, there are 60,000 training samples and 10,000 test samples. During the training process, we set the number of epochs to 20 and the batch size to 128.

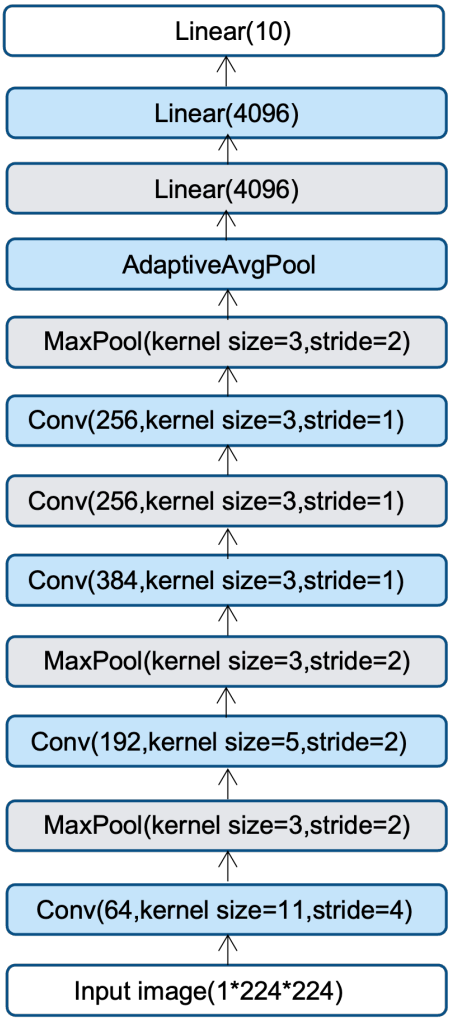


Figure 2.1: AlexNet

During the forward propagation process, the layers of the deep learning model were categorized based on their different functions into convolutional layers, activation layers, pooling layers, linear layers, dropout layers, and flatten layers. In each epoch, the runtime of layers with the same function was summed. Finally, the average energy consumption time over 20 epochs was calculated. The results are shown in the Figure 2.2.

It can be observed that during the forward propagation process of AlexNet, the convolutional layers consume most of the training time. This is because, during computation, the kernel function of the convolutional layers needs to compute row by row with the image, thus requiring the most computation and time. For

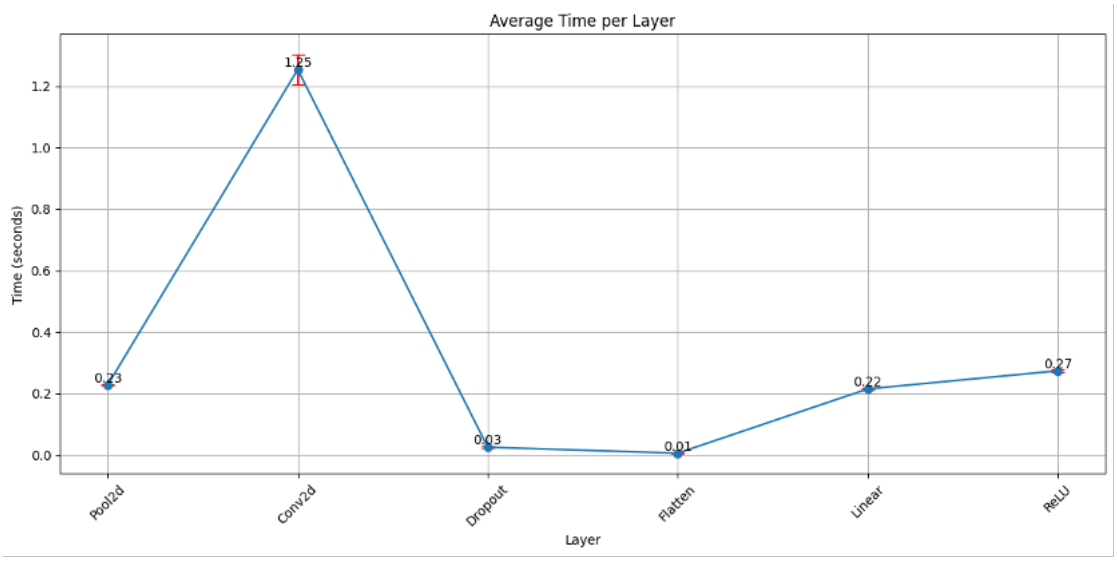


Figure 2.2: Average time per layer

the Dropout and Flatten part, the required time is very short and can even be considered negligible.

Additionally, there is no significant relationship between computation time and the number of parameters in each layer. Among all the layers, the fully connected layer has the largest number of parameters, but its computation time is almost the same to that of the max pooling layer, which has no parameters. The number of parameters in the convolutional layer is related to the size of the kernel and the number of channels. Although it has fewer parameters than the fully connected layer, it consumes the most time.

Chapter 3

Data Preparation

The content of this chapter and Chapter 4 is primarily aimed at building a model that can predict the energy consumption of deep learning models during the training process and validating the proposed model.

This chapter introduces the data collection steps undertaken to build the energy consumption prediction model. These steps include the building various deep learning models from scratch, the identification of potential factors that may influence energy consumption.

3.1 Building Deep Learning Models

In this section, multiple deep learning models were built from scratch, enabling the measurement of energy consumption (i.e., corresponding time consumption) of different layers during the forward propagation process in the training phase using the previously mentioned software tool, nvidia-smi, as well as the energy consumption (i.e., time consumption) data at different stages of the entire training process.

In addition to the previously mentioned AlexNet, ResNet[**21**], VGG [**22**], MobileNet [**23**] and GoogleNet [**24**] were also built to a better look of the energy consumption of training deep learning models.

3.1.1 VGG

The VGG model was proposed by the University of Oxford in 2014, and its architecture consists of a series of convolutional layers. In the data collection section, three VGG models were selected: VGG11, VGG13, and VGG16. Taking VGG11 as an example, its architecture is shown in the Figure 3.1.

3.1.2 MobileNet

MobileNet is a model proposed by Google in 2017, designed for image recognition on mobile and embedded devices, and it has a wide range of applications. In this thesis, 2 types of this model have been built: MobileNetV1 and MobileNetV2.

The MobileNet model structure is mainly composed of multiple depth-wise separable convolutions. The structure of depth-wise separable convolutions(DSC) is shown in the figure, and as for the model architecture, taking MobileNetV1 as an example, is illustrated in the Figure 3.2.

3.1.3 ResNet

ResNet is a highly influential model proposed by Kaiming He et al. in 2016 [**21**], and it won the championship in the 2015 ImageNet image recognition competition. ResNet is primarily composed of a series of residual blocks. The main architecture of a residual block is shown in the Figure 3.3.

In the energy consumption data collection section, three types of ResNet models are selected: ResNet18, ResNet34, and ResNet50. Taking ResNet18 as an example, its architecture is shown in the Figure 3.4.

3.1.4 GoogLeNet

GoogLeNet is a deep learning model proposed by Google in 2014, based on the inception module. Its characteristic feature is that the inception module is a multi-branch block. The structure of the inception module is shown in the Figure 3.5, and the architecture of the GoogLeNet model used in this thesis is also illustrated in the Figure 3.6.

3.2 Potential KPIs

Potential KPIs that may influence the energy consumption during the training process of deep learning models include hardware devices type, different datasets, settings of epoch and batch size during training, model's MACs (Multiply-Accumulate Operations), and model's multi-branch structure.

3.2.1 Hardware Devices

Different hardware devices may exhibit varying energy consumption levels even when using the same deep learning model, given the same dataset, identical initial settings, and the same epoch and batch size configurations. This is because with the iterative advancements in hardware technology, computational capabilities have been improving. Therefore, for the same deep learning model training tasks, newer versions of GPUs may have energy efficiency advantages over older versions. The GPUs selected for this thesis are the 3060 and 4090.

3.2.2 Different Datasets

In different datasets, the image size, number of channels, and the quantity of training and test samples all vary. Therefore, during the training process, the same deep learning model will exhibit different energy consumption levels when trained on different datasets.

In this section, the selected datasets are FashionMNIST and CIFAR-100. FashionMNIST has already been mentioned previously. The images in the CIFAR-100 dataset [**25**] have a size of 3*28*28. The dataset contains 100 classes, with 600 images per class. The total number of training samples is 50,000, and the test set comprises 10,000 images.

3.2.3 Epoch and Batch Size Settings

The number of epochs represents how many times the dataset needs to be trained. More epochs will increase the training duration and energy consumption. The batch size affects the number of images trained in each batch. If the batch size is too large, it may lead to insufficient GPU memory, slowing down the training speed or even interrupting the training task. Conversely, if the batch size is too small, more batches will be required to complete training in one epoch, increasing the

frequency of data exchange in the GPU memory, which may subsequently increase training duration and energy consumption.

3.2.4 MACs of Model

MACs refer to the number of multiply-accumulate operations in a model. Using ptflops [**26**], we can obtain the MACs for different models, indicating that the computational requirements for each image passing through the model vary. Models with higher computational requirements imply that more energy is needed for computation. Therefore, the differences in the number of MACs may affect the energy consumption of different deep learning models during the training process.

3.2.5 Multi-branch Structure

Even with similar MACs, different models may have significant differences in training energy consumption due to variations in their architectures. Therefore, in this section, the multi-branch structure design of GoogLeNet is considered as a potential factor influencing energy consumption.

3.3 Energy Consumption Prediction Models

Through the collection of the simulation data and the analysis of potential factors affecting energy consumption, this section primarily proposes two models for energy consumption prediction.

- **Energy consumption prediction model without considering accuracy**. The first prediction model is designed to predict the energy consumption during the training process, given a specified model, dataset, and the set parameters of epoch and batch size.

- **Energy consumption prediction model considering accuracy**. The second prediction model aims to build upon the first model, attempting to predict the energy consumption required for a model to achieve a given accuracy level during testing.

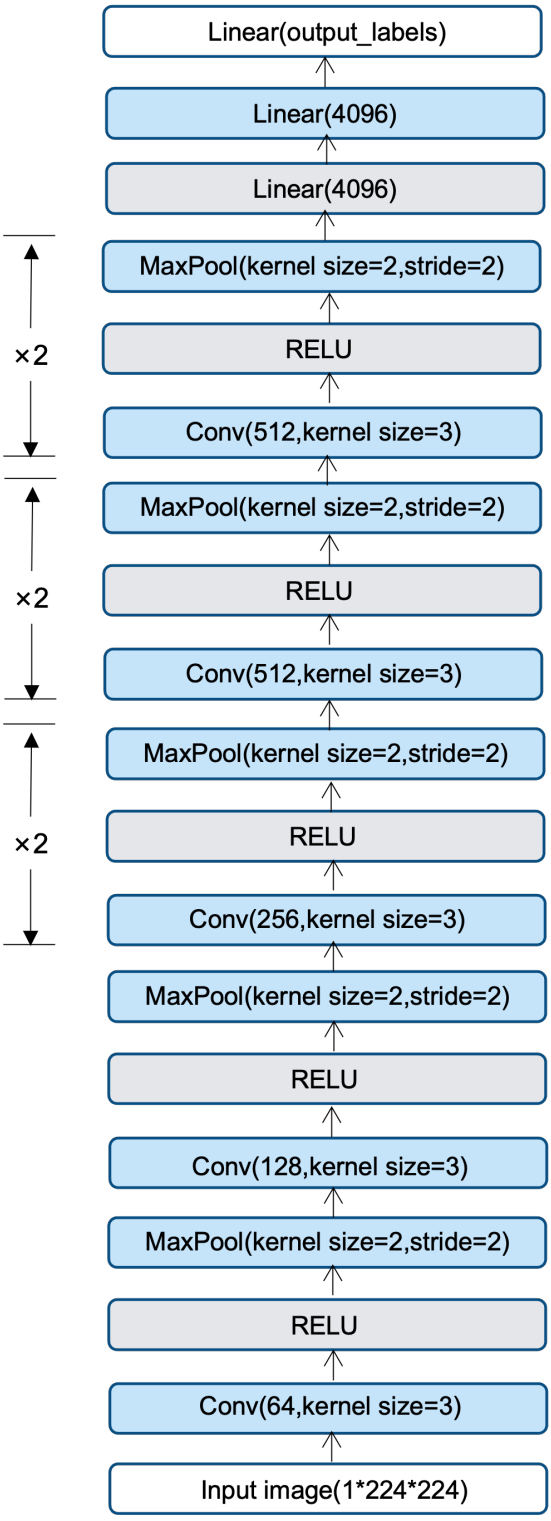


Figure 3.1: VGG11

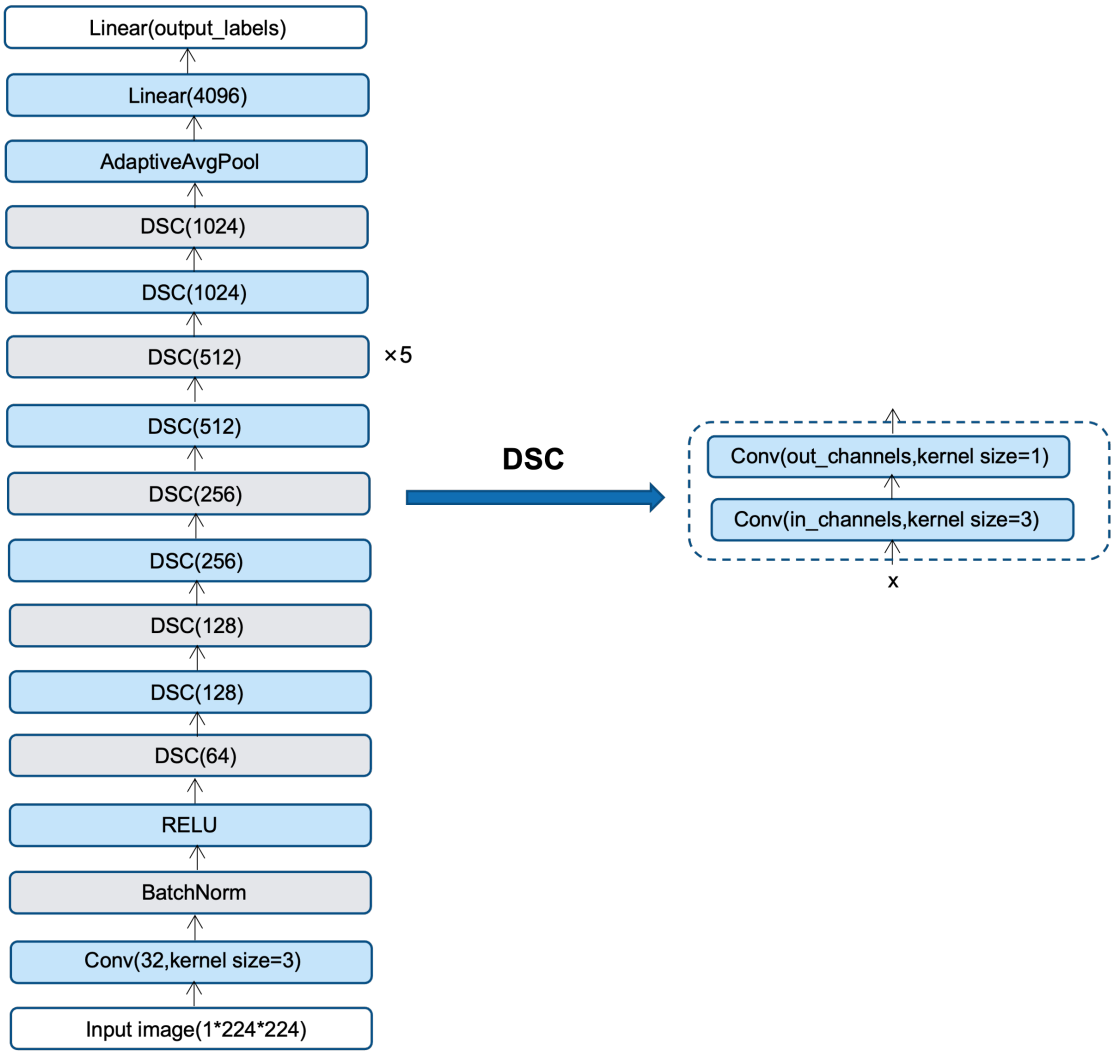


Figure 3.2: MobileNetV1

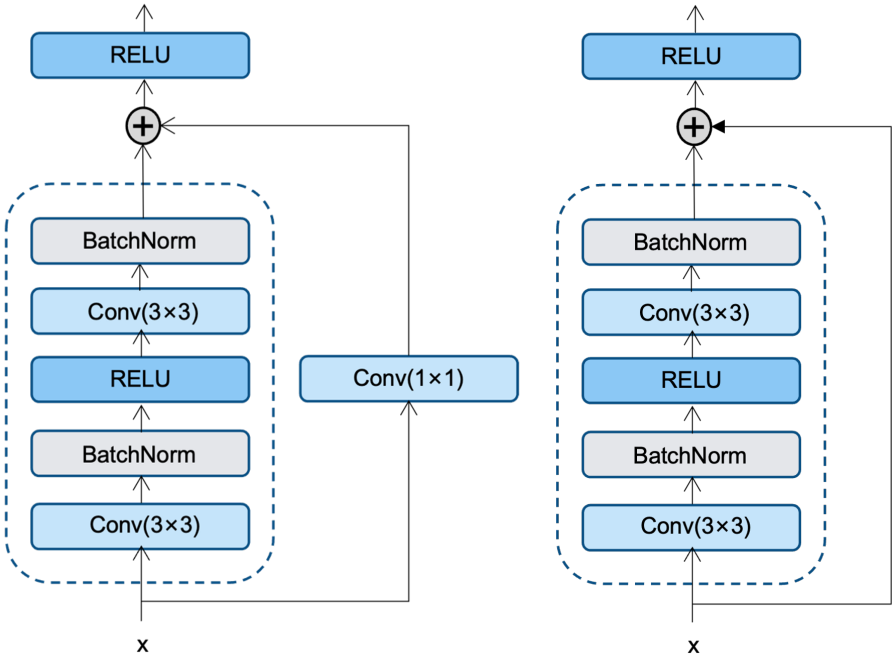


Figure 3.3: ResNet

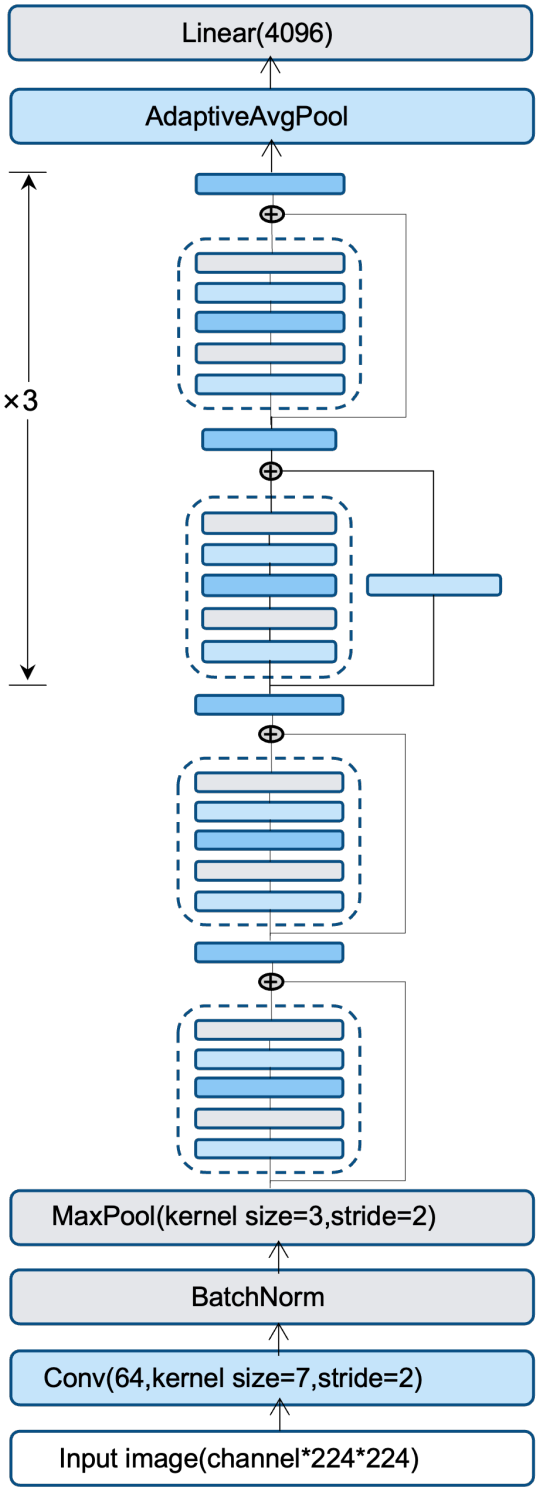


Figure 3.4: ResNet18

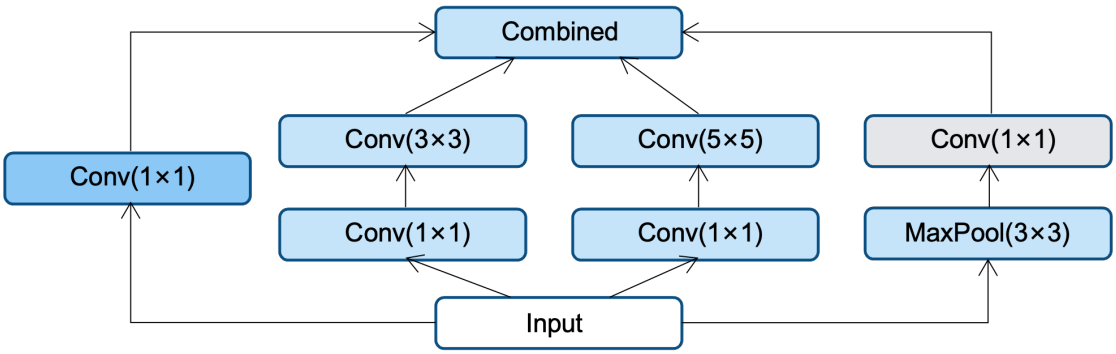


Figure 3.5: Inception

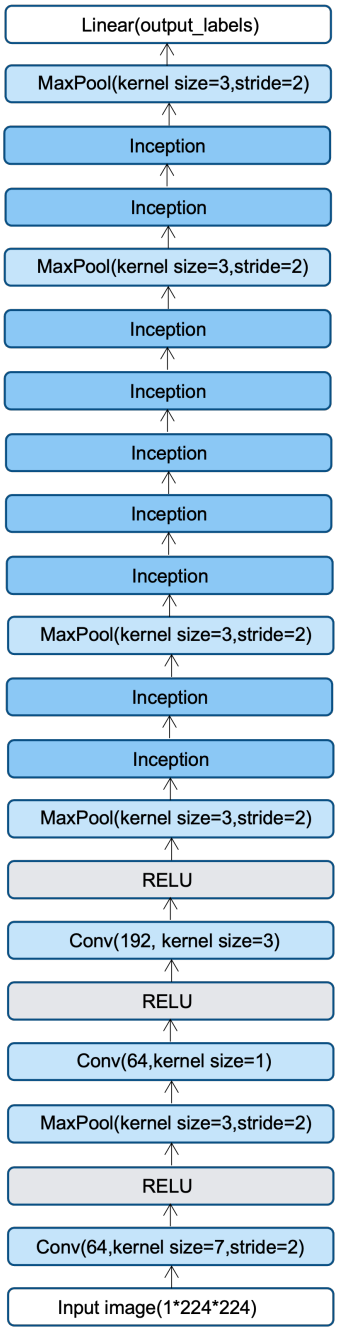


Figure 3.6: GoogLeNet

Chapter 4

Cases Study

By utilizing the various models mentioned in Chapter 3, simulation data is used to validate the potential factors affecting energy consumption discussed earlier, assessing their impact on energy consumption. Subsequently, energy consumption prediction models are proposed and validated.

4.1 Analysis of Potential KPIs

4.1.1 Hardware Devices

To analyze the differences in energy consumption for deep learning model training on different hardware devices, the selected hardware devices are the 3060 and 4090 GPUs. Simulations have been conducted for two deep learning models (AlexNet and ResNet18). The dataset used in this section is FashionMNIST. All models were set to run for 10 epochs during the training process, with a batch size of 128.The results are shown in the Figure 4.1.

It can be observed from the figure that the energy consumption difference between the two hardware devices when training the AlexNet model is not very significant. On average, the energy consumption per epoch for the 3060 GPU is only 212.36J higher than that for the 4090 GPU. However, in the ResNet model, the average energy consumption per epoch for the 3060 GPU is 3034.89J higher than that for the 4090 GPU.

This is because, compared to the 3060 GPU, the 4090 GPU has more advanced manufacturing technology and more computational units. During operation, its average power consumption per second is higher than that of the 3060 GPU, but

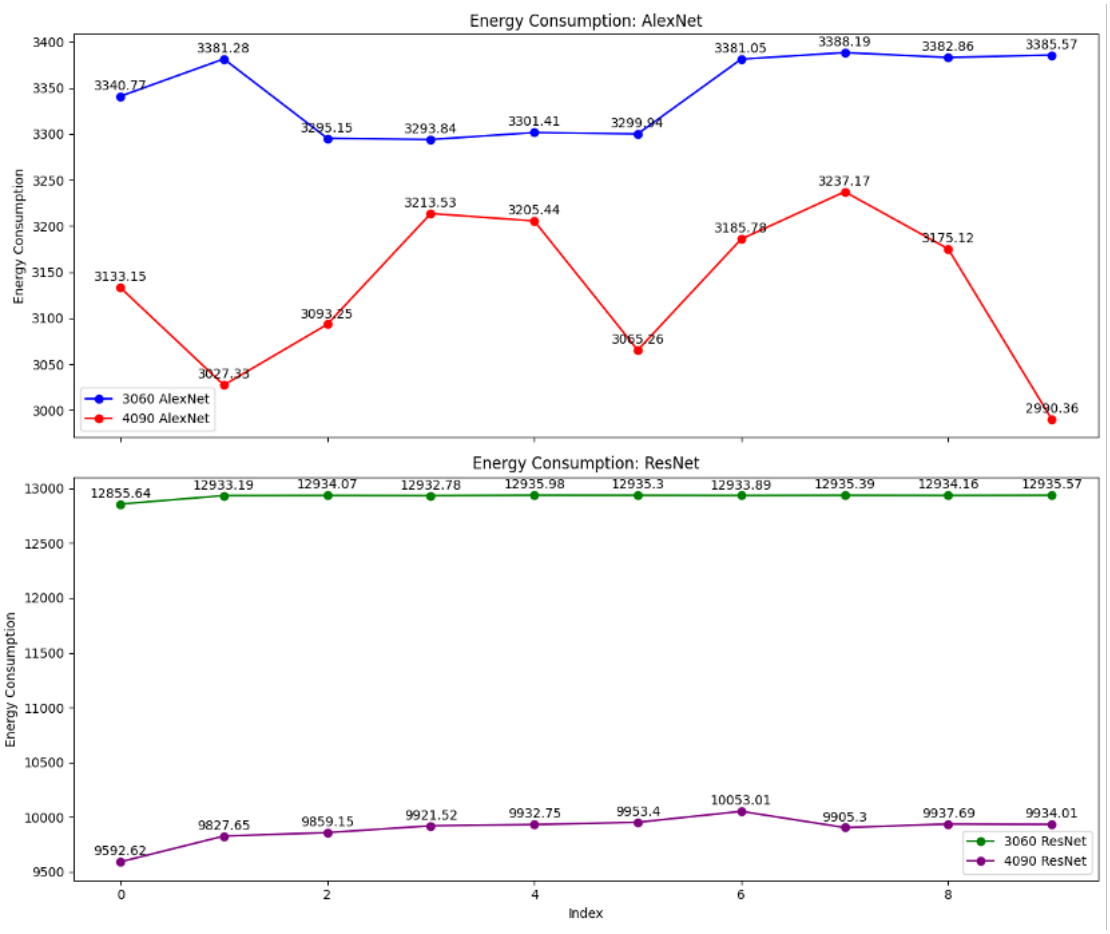


Figure 4.1: Energy consumption comparison between 3060 and 4090

its runtime is shorter. Considering the models, the number of MACs in the ResNet model is higher than in AlexNet. Therefore, when running AlexNet, the energy consumption difference between the two hardware devices is not significant. However, as the number of MACs increases, the difference in energy consumption between the hardware becomes more significant.

4.1.2 Different Datasets

To analyze the impact of different datasets, the 4090 GPU was selected as the hardware device. Two different datasets, FashionMNIST and CIFAR-100, were considered. To verify the differences in energy consumption caused by different datasets across various models, we selected AlexNet, ResNet18, GoogLeNet, and VGG11 for comparison. For each model, the number of epochs was set to 20 and

the batch size was set to 256 during the training process. The results are shown in the Figure 4.2.

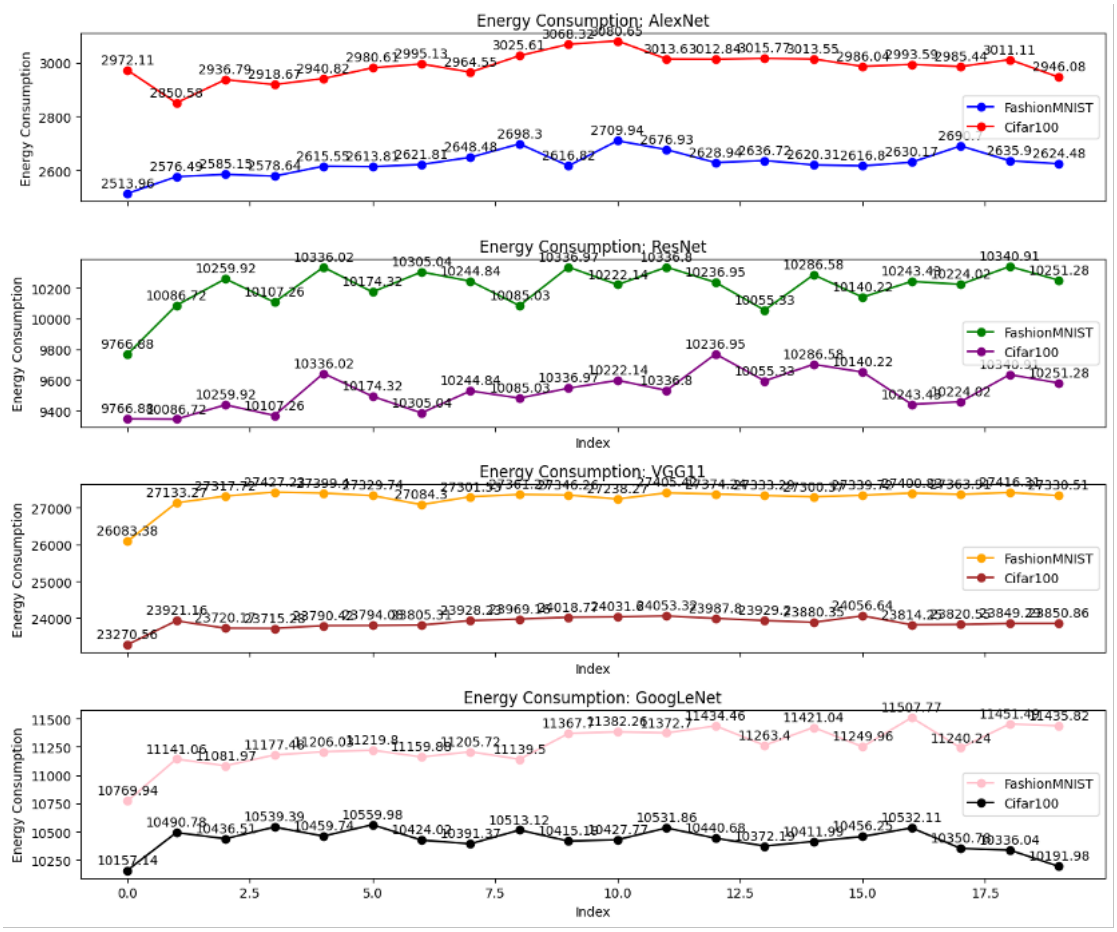


Figure 4.2: Energy consumption comparison between different datasets

It can be observed from the Figure 4.2 that for the relatively simple AlexNet model, the energy consumption on the CIFAR-100 training set is higher than that on the FashionMNIST dataset. However, for the other three models, the energy consumption for training on the FashionMNIST dataset is higher than that on the CIFAR-100 dataset.

The reason for this result is due to the relatively simple structure of the AlexNet model. When training on this model, the CIFAR-100 dataset, which has images with 3 channels, results in higher energy consumption compared to the FashionMNIST dataset, which has images with only 1 channel. After adjusting the input

channels for the AlexNet model, it can be observed that the MACs for the AlexNet model when training on the FashionMNIST dataset is 665.53 MMAC, whereas it is 712.39 MMAC when training on the CIFAR-100 dataset. Additionally, the FashionMNIST training set contains 60,000 images, while the CIFAR-100 training set contains 50,000 images. Calculations show that when using the AlexNet model for training, the MACs for the CIFAR-100 dataset is higher than that for the FashionMNIST dataset. This means that training the CIFAR-100 dataset with AlexNet requires more computation, and consequently, more energy consumption compared to training the FashionMNIST dataset.

As the models become more complex, the MACs for training the same model on two different datasets no longer show significant variation. Taking the VGG11 model, which has the highest number of MACs, as an example, its MACs is 7.58 GMACs when training on the FashionMNIST dataset and 7.64 GMACs when training on the CIFAR-100 dataset. In this case, compared to the AlexNet model, the primary factor influencing energy consumption when training different datasets shifts from the number of MACs to the number of images in the training set. This explains why, for the other three models, the energy consumption for training on the FashionMNIST dataset is higher than that for training on the CIFAR-100 dataset. Moreover, for these three models, the larger the number of MACs, the greater the difference in energy consumption results when training on the two datasets.

4.1.3 Epoch and Batch Size Settings

In this section, the AlexNet model is used for analysis. The selected dataset is FashionMNIST, and the hardware devices used for model training are the 3060 and 4090 GPUs.

To analyze the impact of different epochs and batch sizes, and considering the varying computational capabilities of different hardware, the epoch numbers on the 3060 GPU were set to [10, 20, 30], and the corresponding batch sizes were set to [64, 128, 256, 512], resulting in a total of 12 epoch and batch size combinations. During the training process, each epoch and batch size combination was repeated 5 times to obtain the final average energy consumption. The results are shown in Figure 4.3.

It can be observed that on the 3060 GPU, as the batch size increases, the average energy consumption per epoch gradually decreases. Additionally, with the increase in batch size, the rate of decrease in energy consumption also slows down. When the batch size increases from 256 to 512, the reduction in energy consumption per

epoch is much smaller compared to the increase from 64 to 128. This is because as the batch size increases, the number of images that need to be loaded into the GPU memory for computation in each batch increases. When the GPU memory is sufficient, increasing the batch size reduces the number of times data needs to be swapped in the memory, thus improving computational efficiency and reducing energy consumption. However, when the number of images becomes too large, it is limited by the physical constraints of the hardware, which prevents further improvements in computational speed, resulting in less significant reductions in energy consumption.

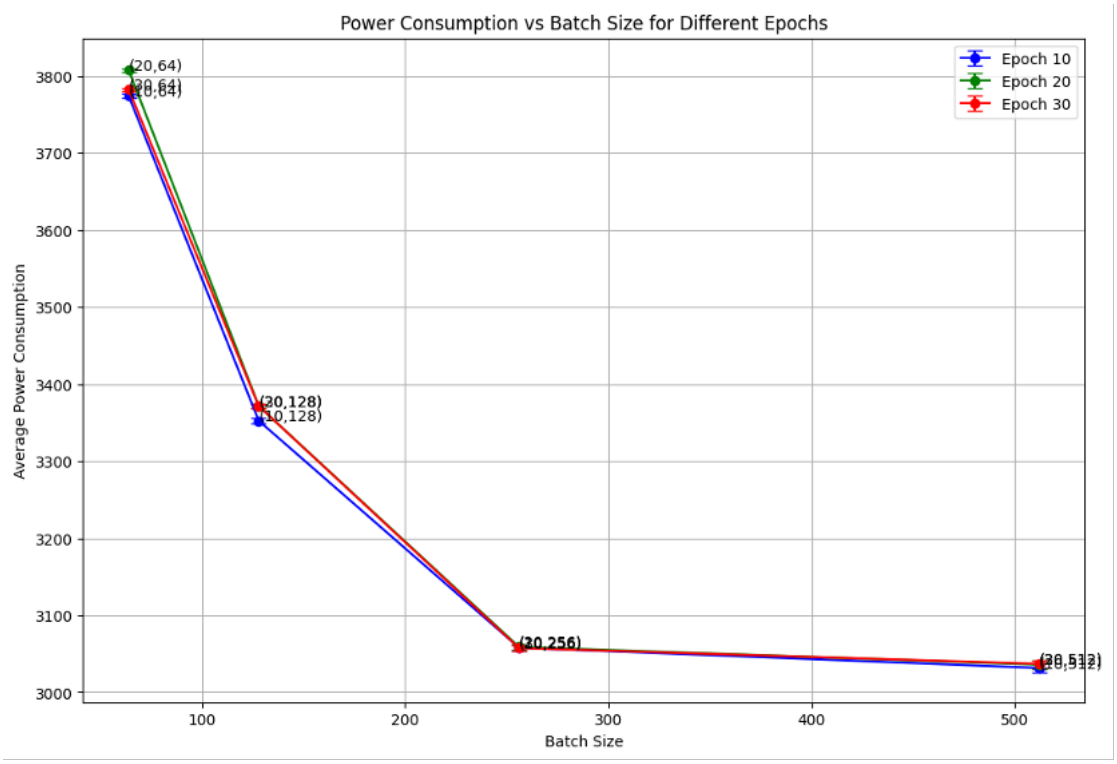


Figure 4.3: Energy consumption comparison between different epochs and batch sizes in 3060

On the 4090 GPU, the AlexNet model was also used for training. Considering that the 4090 GPU has higher hardware performance than the 3060 GPU, the epoch numbers were set to [10, 20, 30] with corresponding batch sizes of [64, 128, 256, 512, 1024, 2048]. For epoch numbers set to [40, 50], the corresponding batch sizes were [64, 128, 256, 512]. Therefore, a total of 26 epoch and batch size combinations were set on the 4090. Each combination was repeated 3 times to obtain the final average energy consumption. The results are shown in Figure 4.4.

It can be observed that on the 4090 GPU, when the batch size increases from 64 to 128, the average energy consumption for different epochs increases. However, with further increases in batch size, the average energy consumption per epoch begins to decrease.

This is because when the batch size is increased from 64 to 128 on the 4090 GPU, the utilization of the GPU's computational resources improves, with average power consumption increasing from 140W to 160W, leading to increased energy consumption. However, as the batch size continues to increase beyond 128, the reduction in energy consumption per epoch verifies the previous analysis on the 3060 GPU. Furthermore, it can be clearly seen that due to the improved hardware performance of the 4090 GPU, when the batch size increases from 256 to 512, there is a more significant overall decrease in energy consumption per epoch compared to the decrease observed on the 3060 GPU.

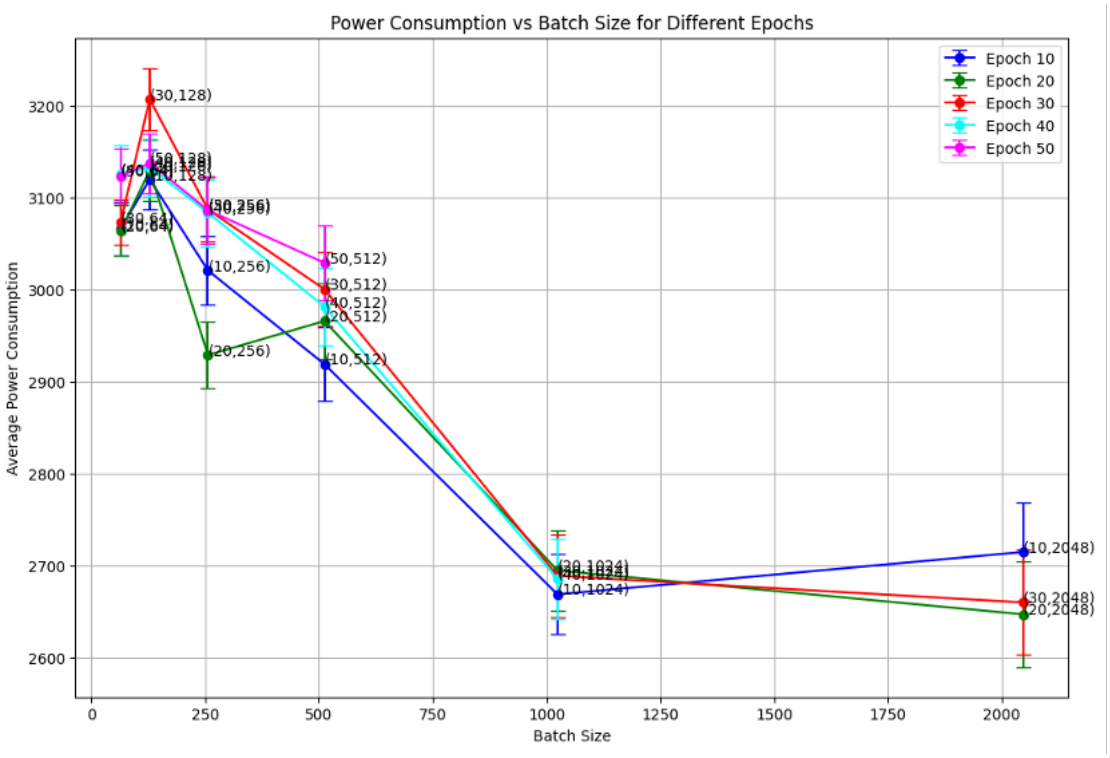


Figure 4.4: Energy consumption comparison between different epochs and batch sizes in 4090

Additionally, the impact of the number of epochs can be clearly observed: an

increase in epochs results in higher overall training energy consumption. The validation results on both the 3060 and 4090 GPUs are shown in Figure 4.5 and Figure 4.6.

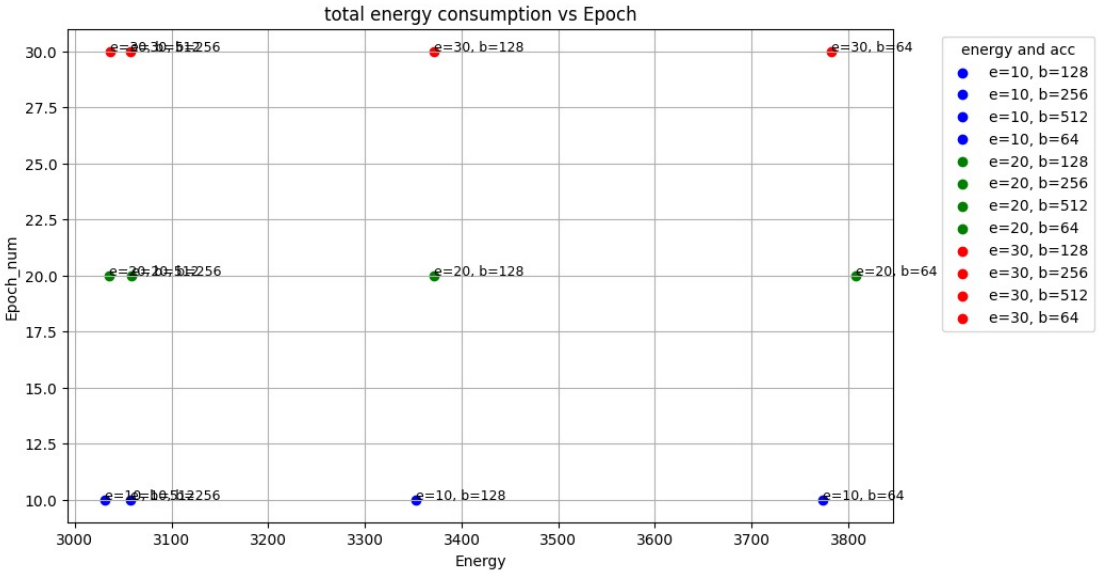


Figure 4.5: Energy consumption vs epoch number in 3060

4.1.4 MACs of Model

In the analysis of this section, the hardware used was the 4090 GPU. All models were set to 20 epochs and a batch size of 256 during the training process. The dataset is FashionMNIST, and all the selected models include all the models mentioned in Chapter 3, as well as some modified versions of GoogLeNet, making a total of 20 models. The final relationship between the MACs of the models and their training energy consumption is shown in Figure 4.7.

It can be observed from the figure that different models with varying numbers of MACs correspond to different final training energy consumptions. Moreover, in a logarithmic coordinate system, this relationship appears to be approximately linear.

4.1.5 Multi-branch Structure

This section primarily analyzes the impact of the multi-branch structure on the energy consumption of models during training.

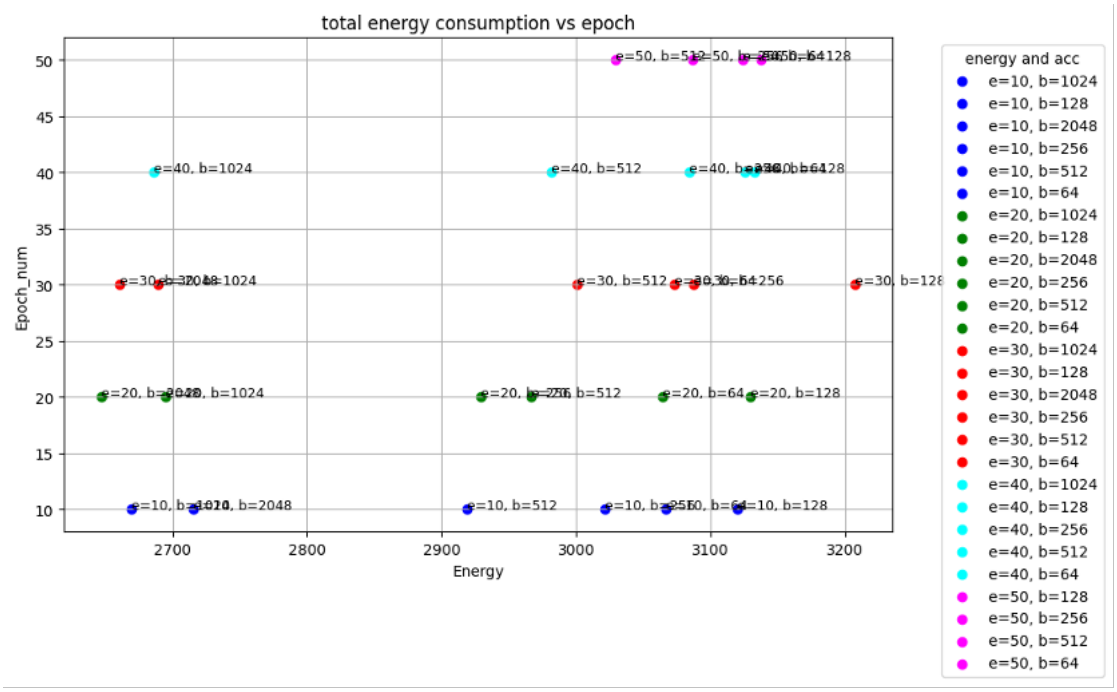


Figure 4.6: Energy consumption vs epoch number in 4090

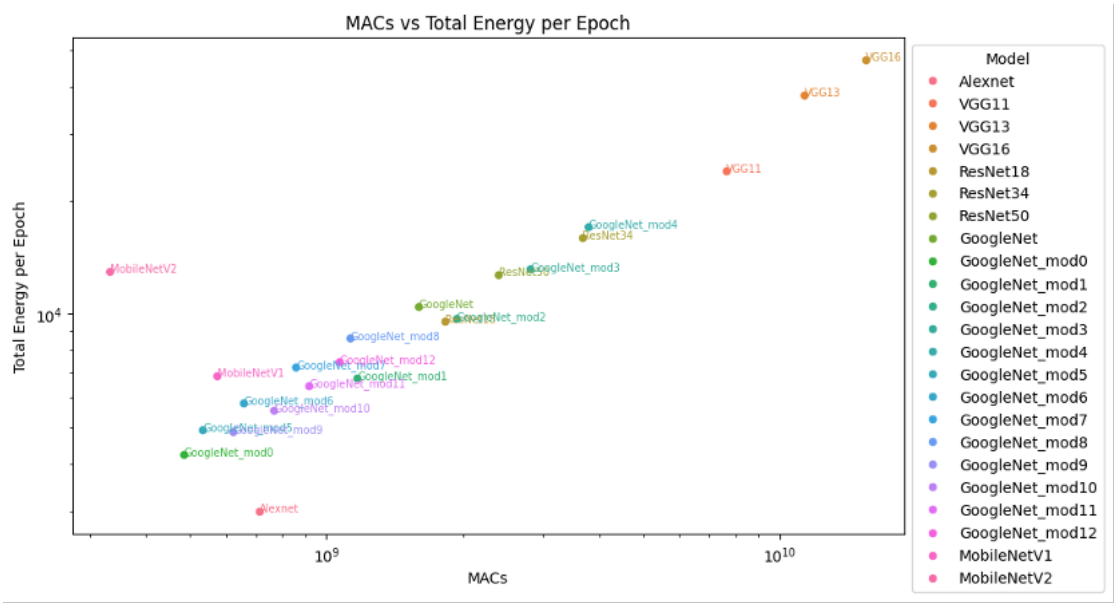


Figure 4.7: MACs vs Energy consumption per epoch of different models on FashionMNIST

In this section, the hardware used is the 4090 GPU. The model was set to 20 epochs and a batch size of 256 during the training process. The dataset used for training is FashionMNIST.

Considering the multi-branch structure in the original GoogLeNet model, which is the Inception block, we made some modifications to analyze the energy consumption of different modified Inception block, without taking the final accuracy of model training into account, given a fixed number of training epochs.

For the adjustments to the inception block, this thesis considers whether different numbers of branches would have an impact. Additionally, the structure within the inception block was modified to verify the trend of energy consumption changes under different MACs. The structure of the modified inception block is shown in Figure 4.8.

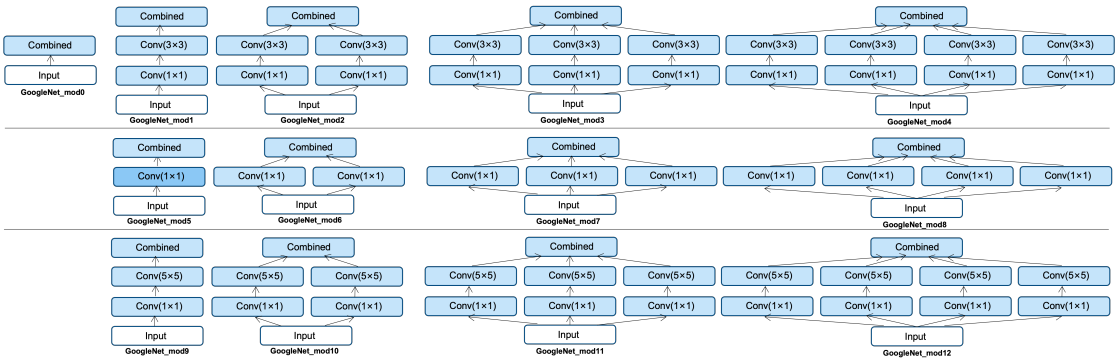


Figure 4.8: The structure of the modified inception block

The training energy consumption results of the GoogLeNet model and its modified versions on the FashionMNIST dataset are shown in Figure 4.9.

It can be observed from the figure that the multi-branch structure affects the model's energy consumption. For the GoogleNet_mod8 and GoogleNet_mod12 models, the number of branches in their modified inception block is 4, while the number of branches in the modified inception block of the GoogleNet_mod1 model is 1. Although these three models have similar MACs, the energy consumption during training for GoogleNet_mod8 and GoogleNet_mod12, which have fewer MACs, is higher than that of the GoogleNet_mod1 model. This demonstrates that increasing the number of branches in the multi-branch structure raises the model's energy consumption during training. However, from the overall trend, the relationship between MACs and energy consumption still approximately exhibits a

clear linear relationship in a logarithmic coordinate system.

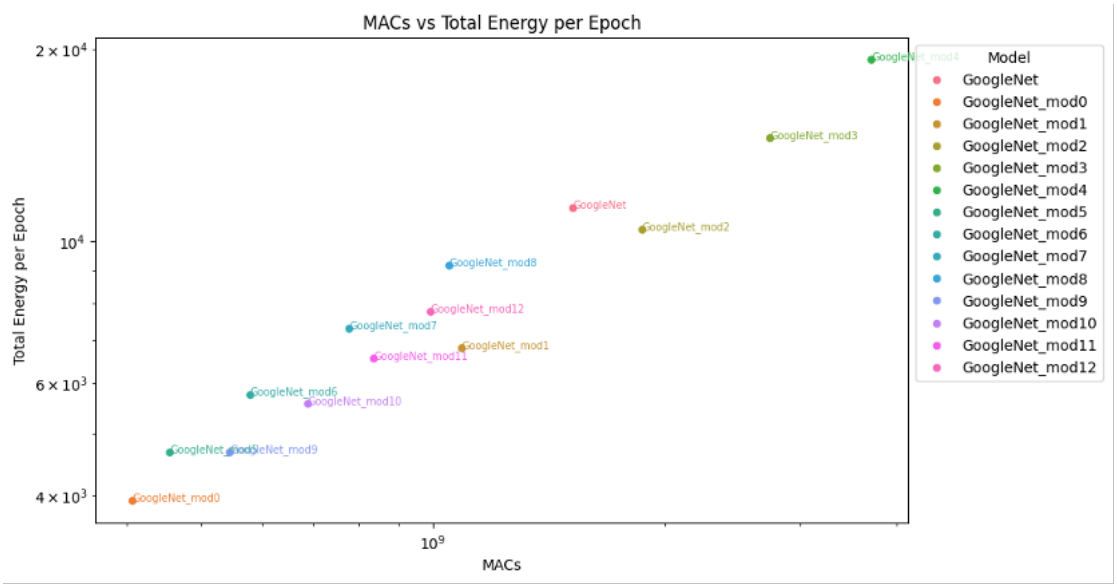


Figure 4.9: MACs vs Energy consumption per epoch of GoogLeNet model and its modified versions

4.2 Energy Consumption Prediction Models

4.2.1 Without Considering Accuracy

In this section, we use all the collected data to find out that all the potential factors mentioned earlier indeed affect the final energy consumption during model training.

However, it can also be observed that the magnitude of influence of different factors on energy consumption varies. During the initial phase of training, given a dataset and the set number of epochs and batch size, the factors analyzed earlier indicate that the final energy consumption is primarily influenced by the MACs of the selected model, the hardware used for training, and whether the model's structure includes elements like the multi-branch structure.

In this context, using the 4090 GPU, with the training set to 20 epochs and a batch size of 256, the 20 models mentioned earlier were trained on the FashionMNIST and CIFAR-100 datasets. Ignoring accuracy, the energy consumption data for different models were collected. The relationship between the number

of MACs of all models and the final energy consumption is shown in the figure. The relationship between the energy consumption data and the MACs of the models during training on the CIFAR-100 dataset is also illustrated in Figure **??**.

It can be observed that on different datasets, the MACs of the models and the final training energy consumption exhibit a linear relationship. Based on this, further analysis was conducted by timing each model during training in four segments: time to device, forward, backward. This was done to identify the running characteristics of each part during the model's training process, thereby aiding in the construction of a more accurate energy consumption prediction algorithm.

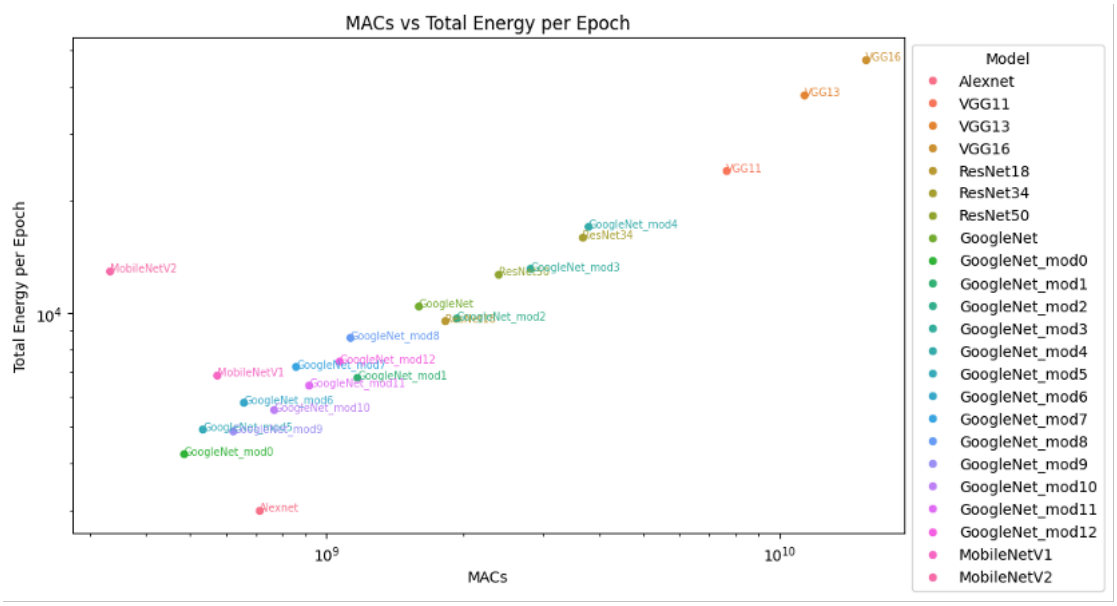


Figure 4.10: MACs vs Energy consumption per epoch of different models on CIFAR-100

Using the data from the training process on the FashionMNIST dataset as an example, the results are shown in Figure 4.11, Figure 4.12 and Figure 4.13.

It can be observed that there are no significant characteristics during the data import process. The time taken to import data does not show a clear relationship with the MACs of the models and is generally around 48 seconds. This is because the same hardware is used when training different models, so the hardware characteristics required to import data into the GPU at the start of model training are consistent. Therefore, the time taken for the "Time to device" part is generally consistent and unrelated to the model's characteristics.

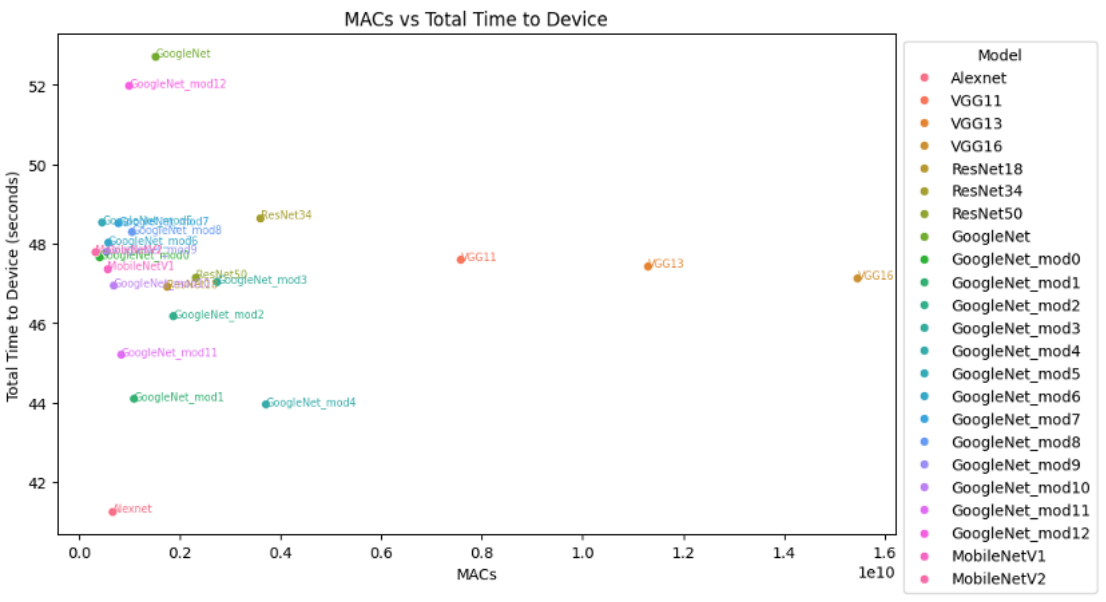


Figure 4.11: MACs vs Total time to device of different models on FashionMNIST

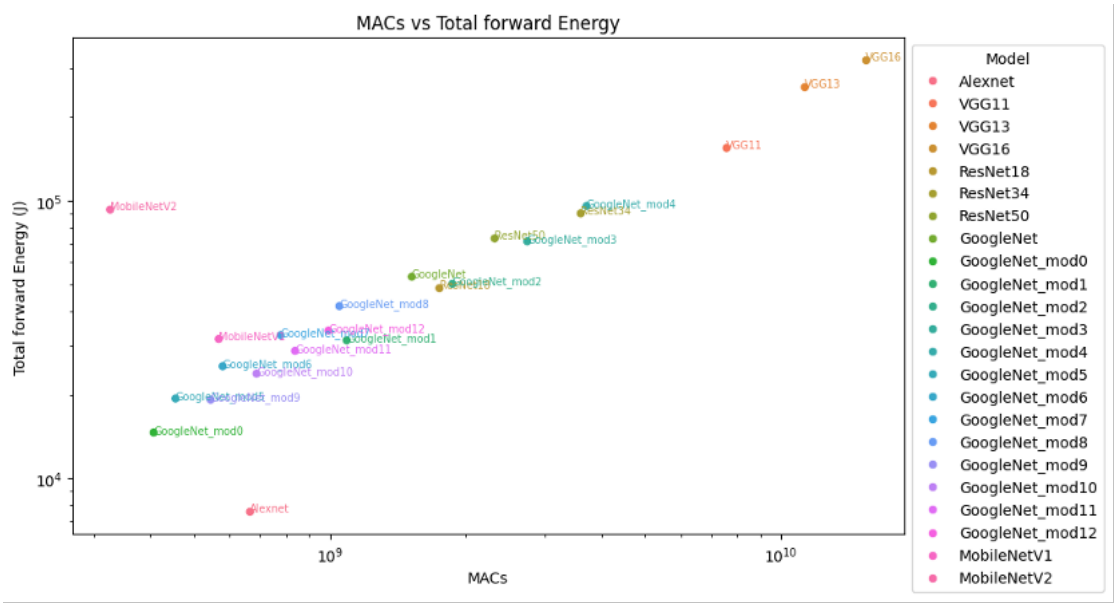


Figure 4.12: MACs vs Total forward energy of different models on FashionMNIST

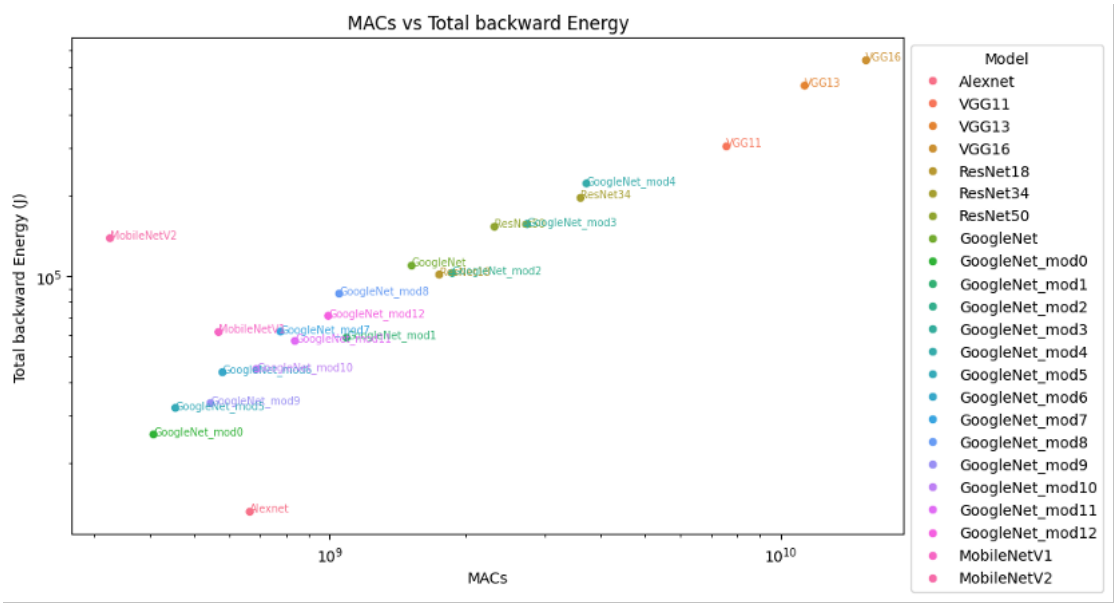


Figure 4.13: MACs vs Total backward energy of different models on FashionMNIST

However, in the energy consumption graphs for the forward and backward propagation processes, it is evident that the energy consumption for both forward and backward propagation shows a clear linear relationship with the number of MACs in a logarithmic coordinate system.

Based on the above analysis, this thesis proposes an energy consumption prediction model 4.1.

$$\text{EnergyConsumption} = HC \times MACs \tag{4.1}$$

where HC represents hardware characteristics, referring to the energy consumption required by the selected hardware to process each MAC operation during training. Different hardware will have different hardware characteristics.

4.2.2 Verification of Model

Considering that the collected data shows an approximately linear relationship between each model's MACs and its energy consumption during training, this thesis uses linear regression to estimate the approximate energy consumption value

for each MAC operation on the 4090 GPU.

Using the collected data for different models on the FashionMNIST and CIFAR-100 datasets, a linear regression model can be applied to obtain the predicted linear relationship between energy consumption and different models for each dataset. The results are shown in the Figure 4.14.

It can be observed that as the number of MACs increases, the predicted energy consumption for model training on the FashionMNIST dataset is higher than on the CIFAR-100 dataset. This result corroborates the analysis in section 4.1.2, which indicates that as the number of MACs grows, the impact of the training set's sample size on energy consumption becomes greater than the impact of the number of channels in the images.

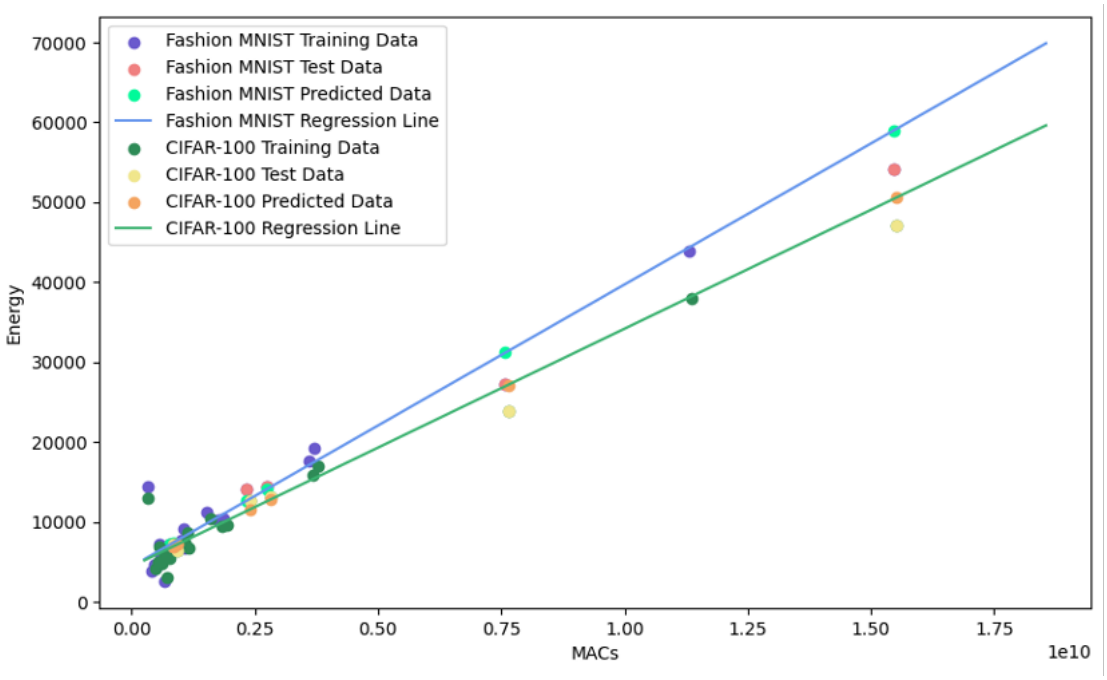


Figure 4.14: Predicted linear relationships between energy consumption and different models on FashionMNIST and CIFAR-100 dataset in 4090

Through the above analysis, it can be estimated that on the 4090 GPU, the computational cost for each MAC operation in a model is approximately 5.9×10^{-11} J.

To validate this value, the energy consumption of all models was recorded during training on the CIFAR-10 dataset [**27**]. Using our predicted HC for the 4090 GPU,

which represents the average energy consumption for each MAC operation during training, the energy consumption of each model on the CIFAR-10 dataset was predicted. The results of this energy consumption prediction model, along with the prediction results obtained through linear regression on other datasets, are shown in Figure 4.15. It can be observed from Figure 4.15 that the energy consumption prediction model closely matches the energy consumption prediction curve obtained through linear regression on the CIFAR-100 dataset.

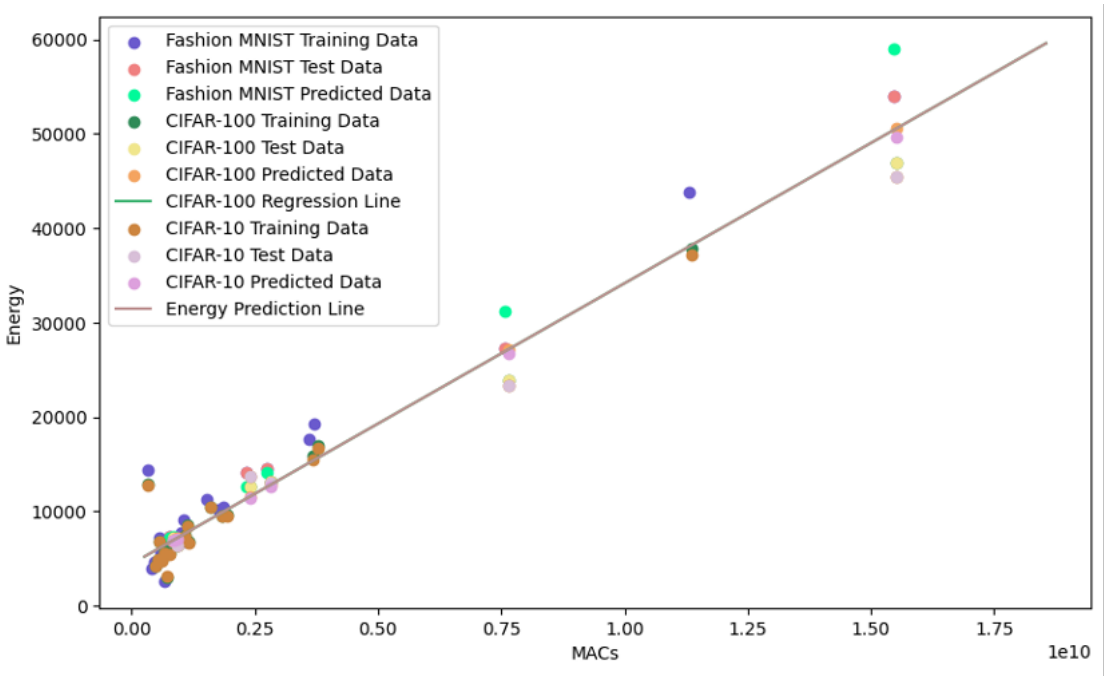


Figure 4.15: Comparison between prediction model result and linear regression result on different models on FashionMNIST, CIFAR-10 and CIFAR-100 dataset in 4090

Additionally, this thesis also applied a linear regression model to the training energy consumption data collected from the CIFAR-100 dataset to determine the relationship curve between training energy consumption and MACs. The results are shown in Figure 4.16.

It can be observed from Figure 4.16 that when training on the CIFAR-10 dataset, the relationship curve between MACs and energy consumption for each model has a slope that is almost identical to that of the energy consumption prediction model proposed in this thesis.

In summary, the energy consumption prediction model proposed in this thesis, which does not consider accuracy, is feasible.

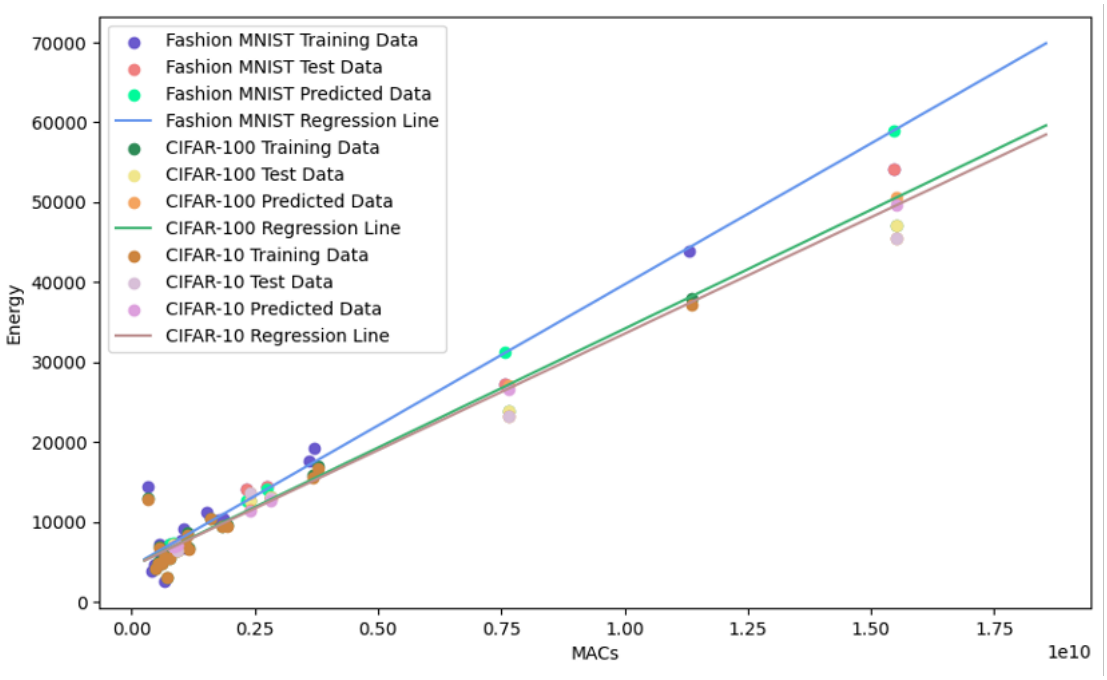


Figure 4.16: Linear regression result on different models on FashionMNIST, CIFAR-10 and CIFAR-100 dataset in 4090

4.2.3 With Considering Accuracy

When considering accuracy, our goal is to predict the approximate energy consumption required to achieve a given accuracy threshold for a model, given the training set, the number of epochs and batch size set during training, and the model itself.

To construct this model, we first trained AlexNet and ResNet18 on the FashionMNIST and CIFAR-100 datasets with different epoch settings in 4090 to observe the accuracy levels that these models could achieve.

From the data in Figure 4.17 and Figure 4.19, it can be observed that on the FashionMNIST dataset, both models can achieve high accuracy. However, on the CIFAR-100 dataset, the accuracy of both models hovers around 50%. Even with the number of epochs set to 100 for AlexNet, higher accuracy cannot be achieved. This is due to the relatively simple structure of the models. When faced with a more

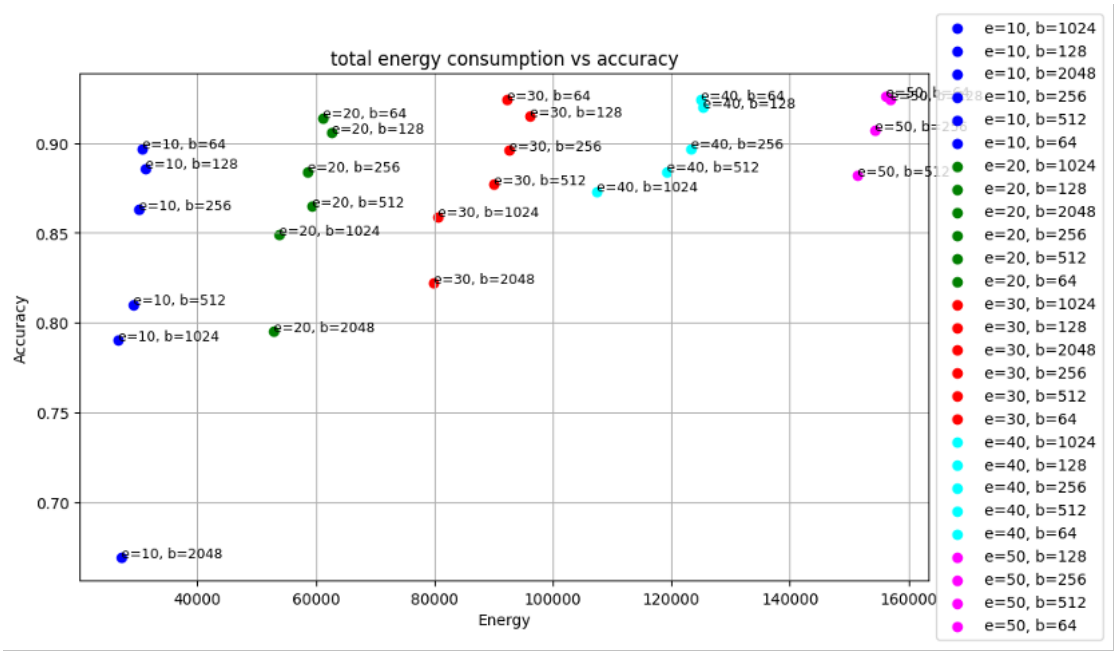


Figure 4.17: AlexNet on FashionMNIST

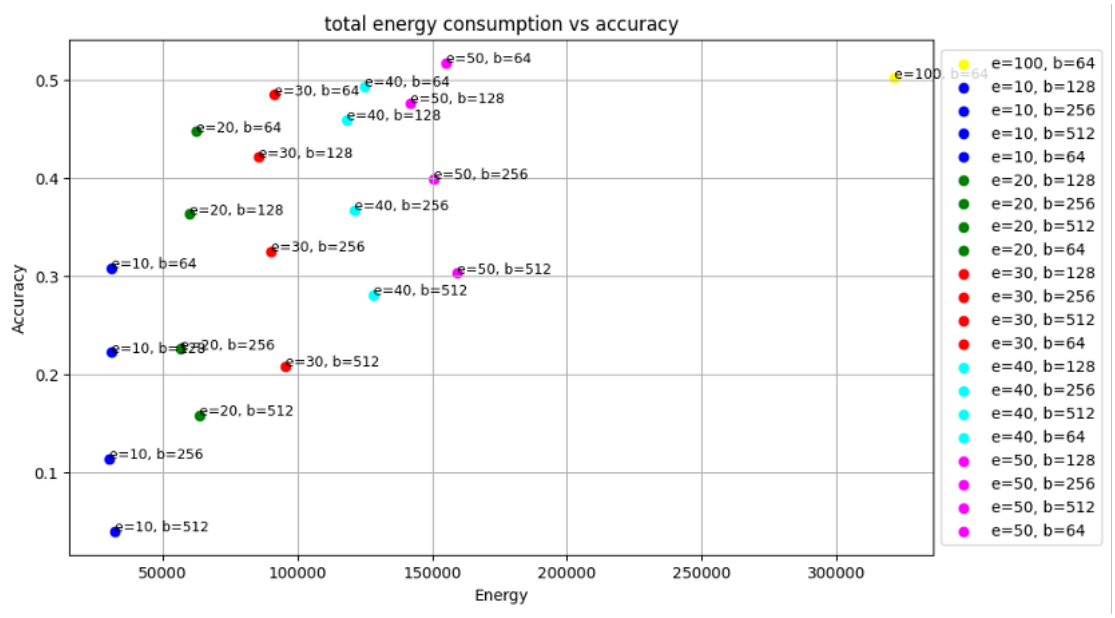


Figure 4.18: AlexNet on CIFAR-100

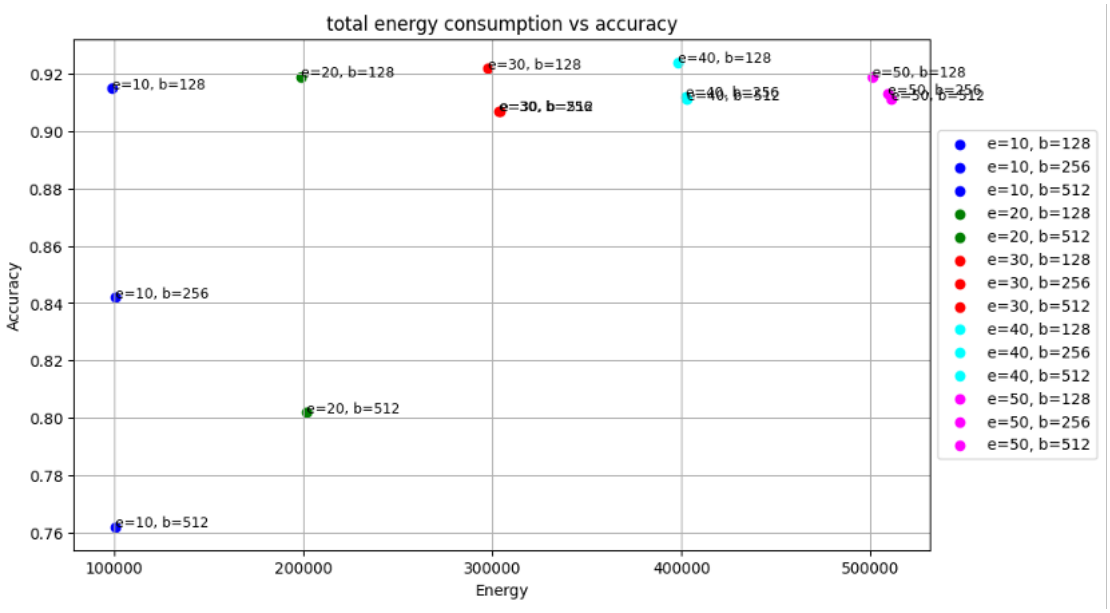


Figure 4.19: ResNet18 on FashionMNIST

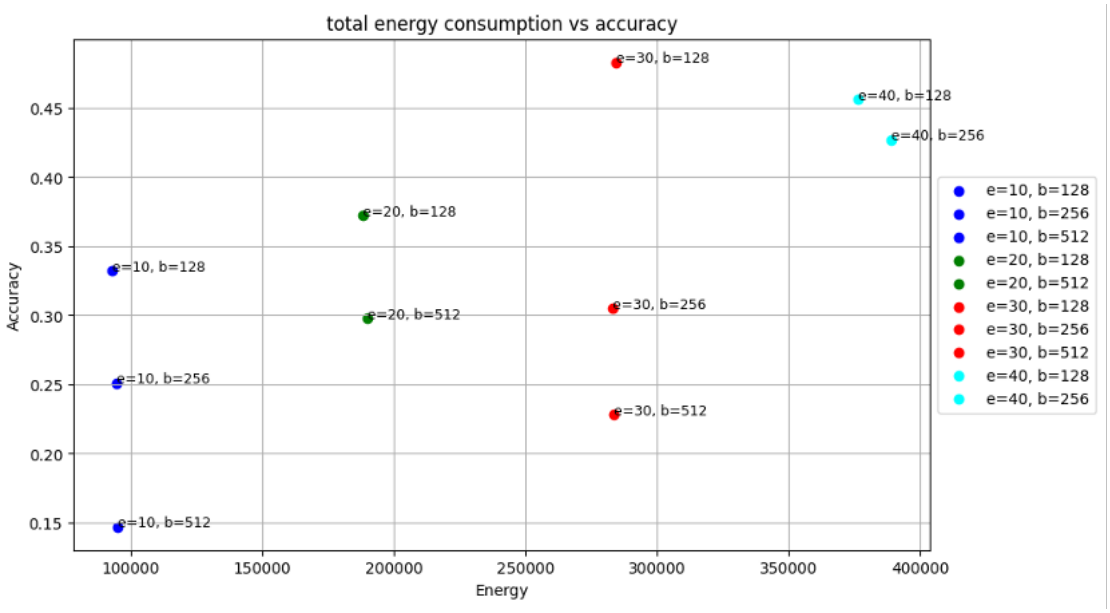


Figure 4.20: ResNet18 on CIFAR-100

complex dataset like CIFAR-100, the structural limitations of the AlexNet model prevent it from capturing all the features necessary for classification, resulting in an inability to further improve accuracy.

In the case of the ResNet18 model, due to GPU resource limitations, higher numbers of epochs were not run on this model. However, according to information available on this website, ResNet18 can achieve an accuracy of 79% with fine-tuning when the number of epochs reaches 300. Additionally, after reviewing relevant literature, no papers were found that discuss the correlation between energy consumption and accuracy. Currently, due to the lack of a clear mathematical relationship between accuracy and energy consumption, it is not yet possible to propose an energy consumption prediction model that considers model accuracy.

Chapter 5

Energy Dipatch Model

5.1 Power Limit and Running Speed

This section primarily explores the performance of GPUs under different power limits. In this section, we use the 3060 GPU. By changing its driver to Nvidia 525, we increased its operating power from 80W to 95W. We used the GreenWithEnvy [**28**] software to ensure that the GPU remains within a set power limit during training. During the testing process, we chose the AlexNet model with the epoch and batch size set to 10 and 128, respectively. The power limits were set to [40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95]. During the simulations, we recorded the average power consumption per epoch, the total energy consumption, and the final accuracy of the model under each power limit.

From Figure 5.1, it can be seen that when the power limit is set between 45W and 70W, the increase in running speed shows a linear relationship with the increase in power limit. However, as the power continues to increase beyond this range, the relationship between running speed and power no longer follows a linear pattern. The improvement in running speed with increasing power becomes smaller. This is because, as power increases, the GPU's computational units approach saturation, causing the rate of increase in computational speed to gradually diminish and eventually reach a maximum. At a power limit of 40W, the running time is significantly higher than at 45W, and the energy consumption is also higher. This is because the power is too low, causing the GPU frequency to drop significantly, which in turn reduces the running speed.

From Figure 5.2, it can be observed that as the power limit of the GPU increases, the overall energy consumption during the model's training process shows an increasing trend. However, the final accuracy of the model does not show significant

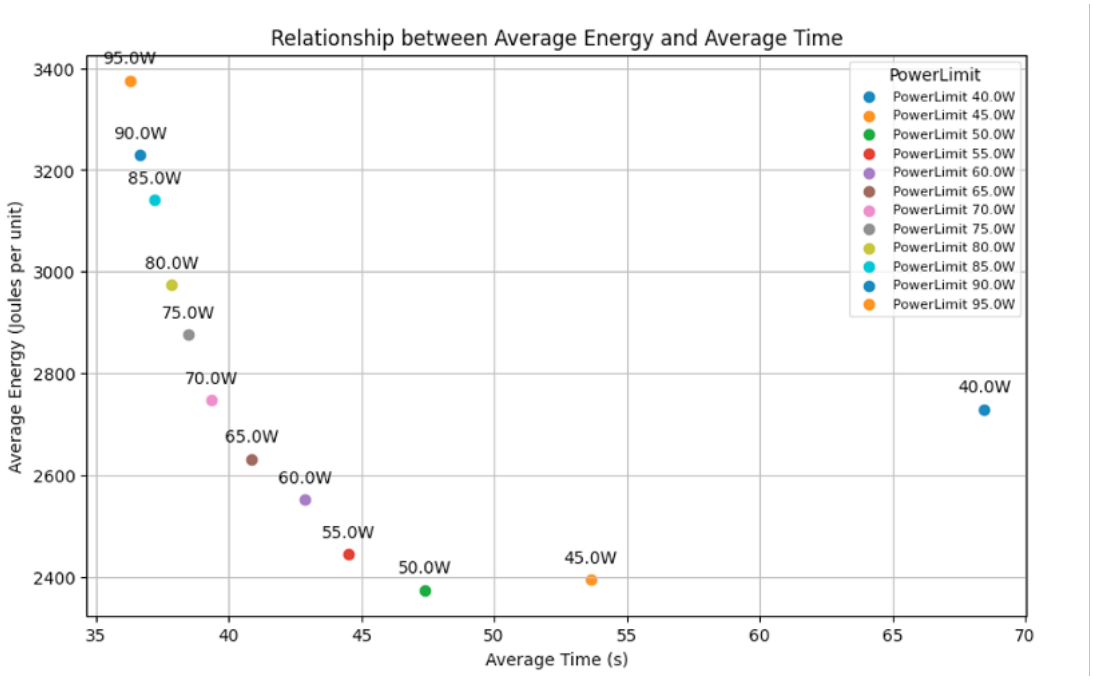


Figure 5.1: Relationship between average energy and average time

differences under different power limits. This indicates that the training results of the model per epoch remain consistent regardless of the GPU power limit settings during training.

5.2 Power Supply Data

This section focuses on constructing a model to simulate various power supply modes for powering a GPU executing a model training task. The primary power sources include three types: solar panels, storage batteries, and the power grid. The power supply logic is as follows:

- **Solar Panels**
  - When solar panels can generate sufficient power, they are prioritized for supplying power to the GPU.
  - Any excess power generated by the solar panels will be partially stored in the storage batteries as a backup.
  - If there is still surplus power, it can be sold back to the power grid.

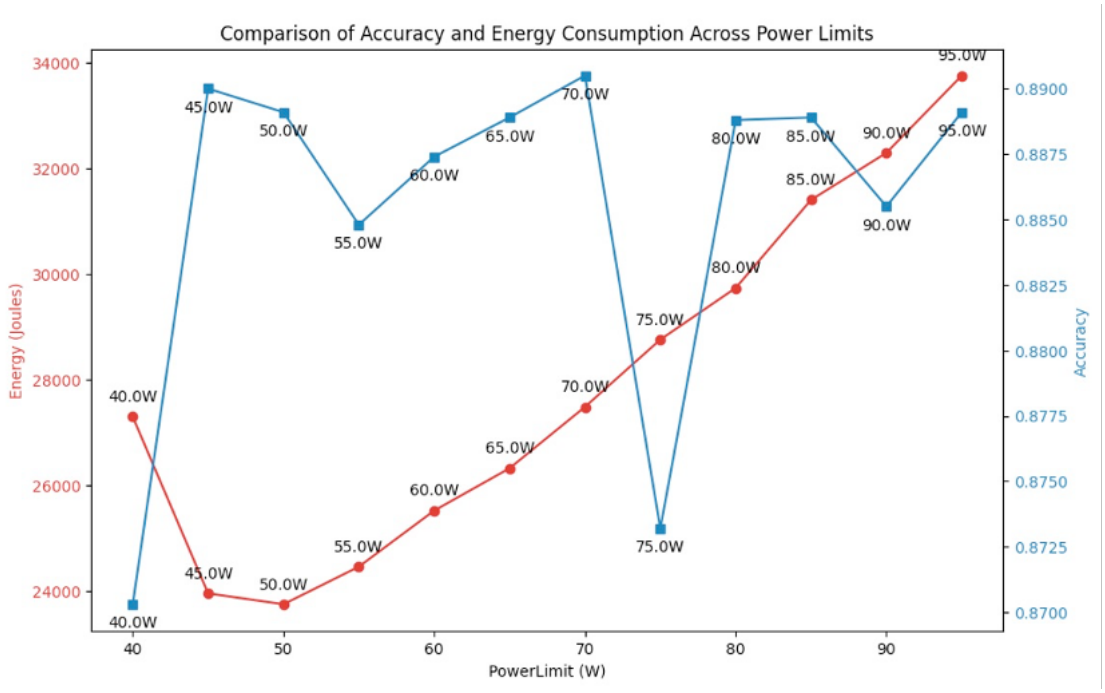


Figure 5.2: Comparison of accuracy and energy consumption across power limits

- **Storage Batteries**
  - When the solar panels cannot generate enough power to supply the GPU, the power stored in the storage batteries will be used.
- **Power Grid**
  - When neither the solar panels nor the storage batteries can provide sufficient power, electricity will be drawn from the power grid to supply the GPU.

By using this multi-source power supply model, we aim to minimize energy costs while ensuring a stable power supply for GPU operations during model training. The data for different GPU power limits and corresponding operating speeds, as described in the previous section, will be integrated into this model.

5.2.1 Parameters

- Start time of total jobs: T_s (h)
- End time of total jobs: T_e (h)

- Supplied power to data center: P_t (W)
- Power generated by solar panels: G_t^{solar} (W)
- Unit price generated by solar panels: Q_t^{solar} (€/kWh)
- Power generated by grid: G_t^{grid} (W)
- Unit price taken by grid: Q_t^{grid} (€/kWh)
- Power sold back to grid: W_t (W)
- Unit price of sold back to grid: $Q_t^{\text{grid}'}$ (€/kWh)
- Power consumption in time-slot t: C_t (W)
- Solar power to supply the data center: P_t^{solar} (W)
- Solar power to restore to the ESD: R_t^{solar} (W)
- Solar power sold back to grid: W_t^{solar} (W)
- Grid power to supply the data center: P_t^{grid} (W)
- Grid power to restore to the ESD: R_t^{grid} (W)
- Restored power in ESD: R_t (W)
- Power sold back to the grid: W_t^{grid} (W)
- ESD discharge to grid: D_t^{grid} (W)
- ESD discharge to data center: D_t^{dc} (W)
- ESD power in each time slot: ESD_t (W)
- ESD Maximum capacity: $\text{ESD}_{\max}$ (W)
- Model training time: J_t (h)

5.2.2 GPU Data

Based on the GPU performance characteristics at different power limits identified in the previous section, and incorporating data referenced in the literature [**16**] , we selected the GPU data and corresponding operating speeds mentioned in the paper. This data will be used for the GPU component in our cost-minimizing model. The data graph is shown in Table 5.1. The first row represents the different GPU power limits set. The second row shows the time required for the GPU to run a specific model training task under the given power limits.

Table 5.1: GPU data and corresponding operating speed

Max.GPU power P(W)	100	125	150	175	200	225	250
Appr.training time J(h)	32	28	26	25	24	23	22

5.2.3 Solar Panel Data

In this thesis, a solar panel system was configured using the PVWatts website [**29**], selecting Turin as the location for weather data. The operation interface is shown in Figure 5.3. On the website, considering the GPU power requirements in the model proposed in this chapter, a 4-square-meter solar panel was configured. The generated data includes the hourly energy production of the solar panel throughout the year. A screenshot of this data is shown in Figure 5.4. Here, we primarily focus on the last column of data, which represents the AC system output power generated by the solar panels each hour.

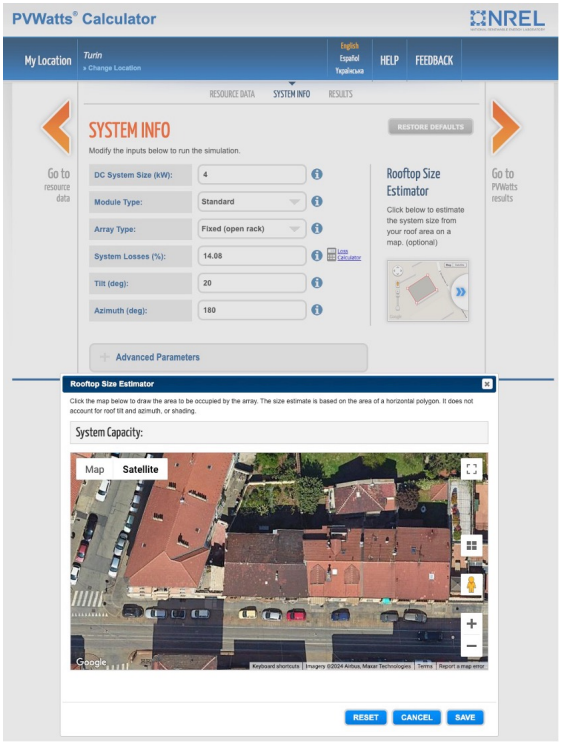


Figure 5.3: Solar panel model

Month	Day	Hour	Beam Irradia	Diffuse Irrac	Ambient Tei	Wind Speed	Albedo	Plane of Arr	Cell Temper	DC Array Output (W)	AC System Output (W)
1	1	0	0	0	-1.4	0	0.2	0	-1.4	0	0
1	1	1	0	0	-1.8	0	0.2	0	-1.8	0	0
1	1	2	0	0	-2.4	0	0.2	0	-2.4	0	0
1	1	3	0	0	-3.2	0	0.2	0	-3.2	0	0
1	1	4	0	0	-3.5	0	0.2	0	-3.5	0	0
1	1	5	0	0	-3.6	0	0.2	0	-3.6	0	0
1	1	6	0	0	-3.6	0	0.2	0	-3.6	0	0
1	1	7	0	0	-3	0	0.2	0	-3	0	0
1	1	8	30	7	-2.1	0	0.2	0	-2.1	0	0
1	1	9	137	26	-0.8	0	0.2	87.285	3.243	212.476	193.056
1	1	10	247	42	1	0	0.2	180.568	9.596	485.273	456.187
1	1	11	328	52	3.3	0	0.2	275.13	16.43	736.008	698.037
1	1	12	359	55	6	0	0.2	319.106	21.224	840.409	798.739
1	1	13	332	52	6.9	0	0.2	298.184	21.125	784.146	744.47
1	1	14	254	43	7.1	0	0.2	220.537	17.623	583.425	550.861
1	1	15	146	28	6.6	0	0.2	113.783	12.021	299.349	276.852
1	1	16	37	8	5.5	0	0.2	26.227	6.723	61.502	47.432
1	1	17	0	0	3.8	0	0.2	0	3.8	0	0
1	1	18	0	0	1.4	0	0.2	0	1.4	0	0
1	1	19	0	0	0.3	0	0.2	0	0.3	0	0
1	1	20	0	0	-0.5	0	0.2	0	-0.5	0	0
1	1	21	0	0	-1	0	0.2	0	-1	0	0
1	1	22	0	0	-1	0	0.2	0	-1	0	0

Figure 5.4: Solar panel data

5.2.4 Storage Battery Data

To construct the energy dispatch model, we need to integrate a model of the storage batteries. Using the parameters set on these websites [**30**][**31**], we configure the charge/discharge efficiency, self-discharge rate, and overall capacity of the batteries. In this thesis, the storage batteries are configured to provide 250Wh, which is enough to power the GPU at its maximum power for 1 hour. And the efficiency paramters can be found in Table 5.2.

	Charge efficiency α	**Discharge efficiency** β	**Self-discharge efficiency** θ
ESD	0.9	0.9	0.005

Table 5.2: ESD efficiency parameters

5.2.5 Grid Data

In this model, we primarily consider the price of electricity from the grid. We simplified the grid pricing by using a time-of-use pricing model for residential electricity in Table 5.3, where different time periods correspond to different prices. Additionally, the price for selling excess electricity back to the grid is simplified to a fixed value.

Tariff	Time slot	Q_t^{grid} €/kWh	$Q_t^{\text{grid}'}$ €/kWh
F1	08:00 – 20:00	0.45	0.3
F2	20:00 – 23:00	0.35	0.3
F3	23:00 – 08:00	0.25	0.3

Table 5.3: Tariff price

5.3 Building the Energy Dispatch Model

Based on the references and data mentioned in the previous sections, the mathematical model for energy dispatch is constructed and each formula is explained below. The schematic diagram of the energy dispatch model is shown in Figure 5.5.

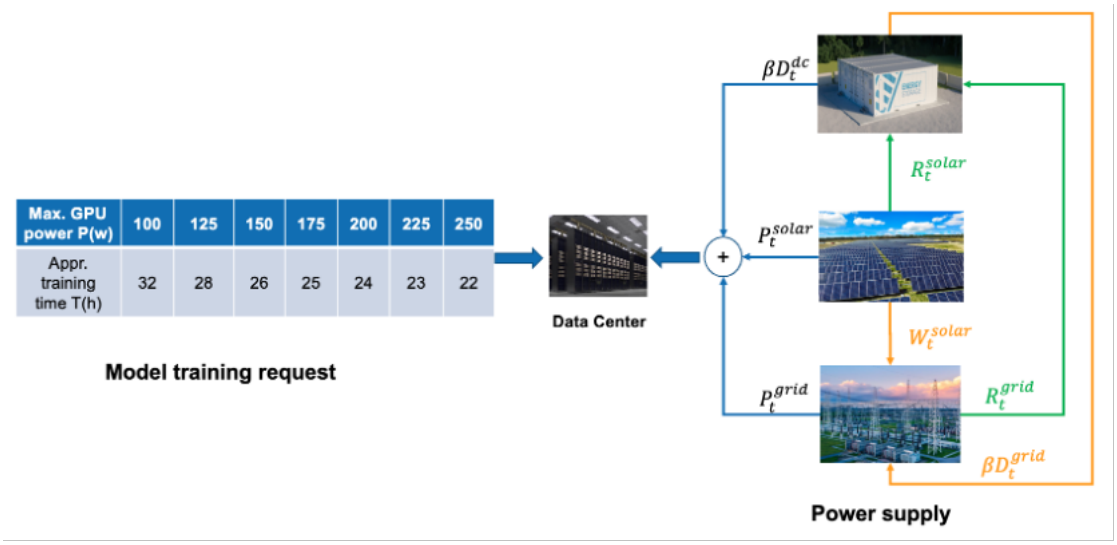


Figure 5.5: Energy dispatch model

As shown in Figure 5.5, during the model training process, various power limit settings correspond to different required working durations. In the model, it is assumed that setting different power limits for the GPU corresponds to varying completion times for the training task. When a higher power limit is set, the time required for the GPU to complete the training task under this power limit is shorter. Conversely, when a lower power limit is set, the time required for the GPU to complete the training task under this power limit is longer. Additionally, it is assumed that during the training process, the training task is completed uniformly, i.e., each hour completes a fraction of the task corresponding to one divided by the total working duration. the training task is completed uniformly, i.e., each

hour completes a fraction of the task corresponding to one divided by the total working duration. To ensure that there is sufficient power for the training task, three different power sources are considered: solar energy, storage batteries, and the power grid. The objective is to minimize the costs incurred during the training process.

5.3.1 Objective Function

We assume the data center has 3 kinds of power supply: power grid, solar panel, and ESD (Energy Storage Devices), which can be seen in the right part of Figure 1. The solar panel generates renewable energy G_t^{solar}, which can be used to power the data center P_t^{solar} directly, or stored into ESD R_t^{solar} for later use, or sold back to the power grid W_t^{solar} to finance part of the high energy expenditure of the data center.

An alternative option for greening the data centers is to buy electricity from the power grid P_t^{grid} when the solar energy is insufficient. To make the best use of renewable energy, we use ESD to store renewable energy R_t^{solar} when its supply is abundant, or store the electricity from the grid R_t^{grid} when the outside electricity price is low. The energy stored in ESD can be discharged to power the data center D_t^{dc} when the outside electricity price is high, or sold back to the power grid D_t^{grid} to lower the overall energy cost.

- The objective function:

$$\min \sum_{t=T_s}^{T_e} C_t \tag{5.1}$$

$$C_t = G_t^{\text{grid}} \cdot \Delta T \cdot Q_t^{\text{grid}} - W_t^{\text{grid}} \cdot \Delta T \cdot Q_t^{\text{grid}'} \tag{5.2}$$

$$\Delta T = 1 \text{ hour} \tag{5.3}$$

- The decision variable is:

$$y_{t,k} \in \{0, 1\} \tag{5.4}$$

  when the value is 1, means at time slot t the power level of data center is set to level k.

k is the power level number of data center:

$$k \in \{1,2,3,4,5,6,7\} \tag{5.5}$$

P_k is the power level of data center:

$$P_k \in \{100,125,150,175,200,225,250\} \tag{5.6}$$

At each time slot, only one level can be selected:

$$\sum_{k=1}^{7} y_{t,k} = 1 \quad \forall t \in [T_s, T_e] \tag{5.7}$$

The selected power level is:

$$P_t = y_{t,k} * P_k \quad \forall t \in [T_s, T_e] \tag{5.8}$$

5.3.2 Power Supply Constraints

- The power range generated by solar panels:

$$0 \leq G_t^{\text{solar}} \leq G_{t_{\max}}^{\text{solar}} \quad \forall t \in [T_s, T_e] \tag{5.9}$$

$$G_t^{\text{solar}} = P_t^{\text{solar}} + R_t^{\text{solar}} + W_t^{\text{solar}} \quad \forall t \in [T_s, T_e] \tag{5.10}$$

- The power range generated by power grid:

$$0 \leq G_t^{\text{grid}} \leq G_{\max}^{\text{grid}} \quad \forall t \in [T_s, T_e] \tag{5.11}$$

$$G_t^{\text{grid}} = P_t^{\text{grid}} + R_t^{\text{grid}} \quad \forall t \in [T_s, T_e] \tag{5.12}$$

- The power charged to ESD:

$$R_t = R_t^{\text{solar}} + R_t^{\text{grid}} \quad \forall t \in [T_s, T_e] \tag{5.13}$$

- The power sent back to grid:

$$W_t^{\text{grid}} = W_t^{\text{solar}} + \beta D_t^{\text{grid}} \quad \forall t \in [T_s, T_e] \tag{5.14}$$

- Update ESD power in each time slot:

$$\text{ESD}_t = (1-\theta) \cdot [\text{ESD}_{t-1} - D_{t-1}^{\text{dc}} - D_{t-1}^{\text{grid}} + \alpha R_{t-1}] \quad \forall t \in [T_s, T_e] \tag{5.15}$$

$$\text{ESD}_1 = 0, \quad D_1^{\text{dc}} = 0, \quad D_1^{\text{grid}} = 0 \quad \forall t \in [T_s, T_e] \tag{5.16}$$

$$0 \leq \text{ESD}_t \leq \text{ESD}_{\max} \quad \forall t \in [T_s, T_e] \tag{5.17}$$

$$0 \leq D_t^{\text{dc}} + D_t^{\text{grid}} \leq \text{ESD}_t \quad \forall t \in [T_s, T_e] \tag{5.18}$$

$$0 \leq R_t \leq \text{ESD}_{\max} - \text{ESD}_t \quad \forall t \in [T_s, T_e] \tag{5.19}$$

- Total power supply to data center:

$$P_t = P_t^{\text{solar}} + P_t^{\text{grid}} + \beta D_t^{\text{dc}} \quad \forall t \in [T_s, T_e] \tag{5.20}$$

- Power consumption in each time-slot should be less than or equal to the selected GPU power:

$$C_t \leq P_t \quad \forall t \in [T_s, T_e] \tag{5.21}$$

- Model training task progress:

$$\sum_{t=1}^{T} \frac{1}{J_t} \geq 1 \quad \forall t \in [T_s, T_e] \tag{5.22}$$

5.4 Simulation Scenarios

In the simulation of this mathematical model of energy dispatch, the following assumptions are made:

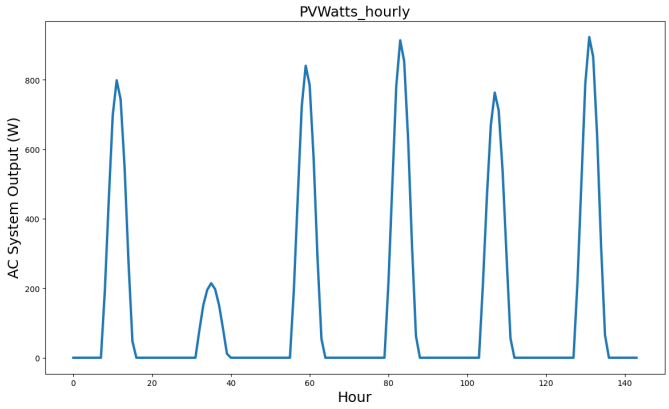
- Once the training task starts on the GPU, it runs continuously until the task is completed, without any interruptions.
- The start time of the training task is set to any hour between the 0th and 143rd hour of the year (out of a total of 8760 hours in a year).
- The duration of the training task is set between 23 and 32 hours.

The simulation results of solar power supply and corresponding electricity cost of every hour are shown in Figure 5.6. The red line in the figure represents the hourly energy generated by the solar panels during the first week, and the blue line represents the final cost data for starting the training task at different times within the week, with the task duration set to a maximum of 23 hours. It can be observed that, with the task duration set to 23 hours, the minimum cost for the training task is approximately 0.2 euros, with the optimal start time for the training task set at the 115th hour.

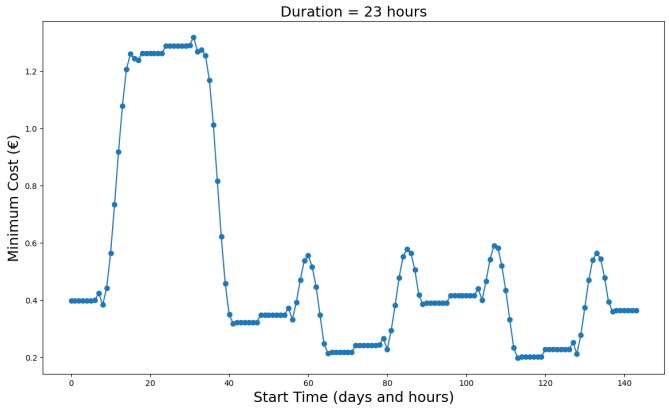
Furthermore, from the data patterns in Figure 5.6, it can be seen that when the weather conditions during the day are relatively good, meaning that solar energy is abundant, the total cost of the training task generally remains at a relatively low level. However, when the weather conditions are poor, and solar energy generated during the day is insufficient to provide adequate power for model training, it becomes necessary to draw power from the grid to ensure the GPU's operation, resulting in higher costs. Additionally, with the task duration set to 23 hours, starting the training task around noon each day leads to a local maximum in costs. This is because starting the task at this time means that the solar panels cannot provide enough energy for a significant portion of the training period, and the GPU must rely on grid power during the evening hours, thus incurring higher expenses.

Furthermore, when considering different training task completion times, the final cost scenario is illustrated in Figure 5.7.

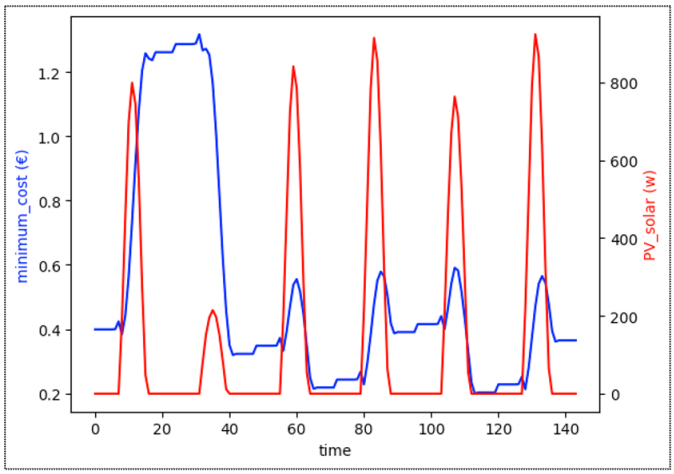
In this section, the training costs for maximum completion times of 23, 25, 27, 29, 31, and 32 hours are presented. It can be observed that as the maximum completion time increases, both the maximum and minimum costs for starting the training task at different times decrease. Specifically, under the 23-hour completion



(a) Solar power at different time



(b) Minimum cost under different start time



(c) Solar power supply and minimum cost

Figure 5.6: Solar power supply vs minimum cost_Duration=23 hours

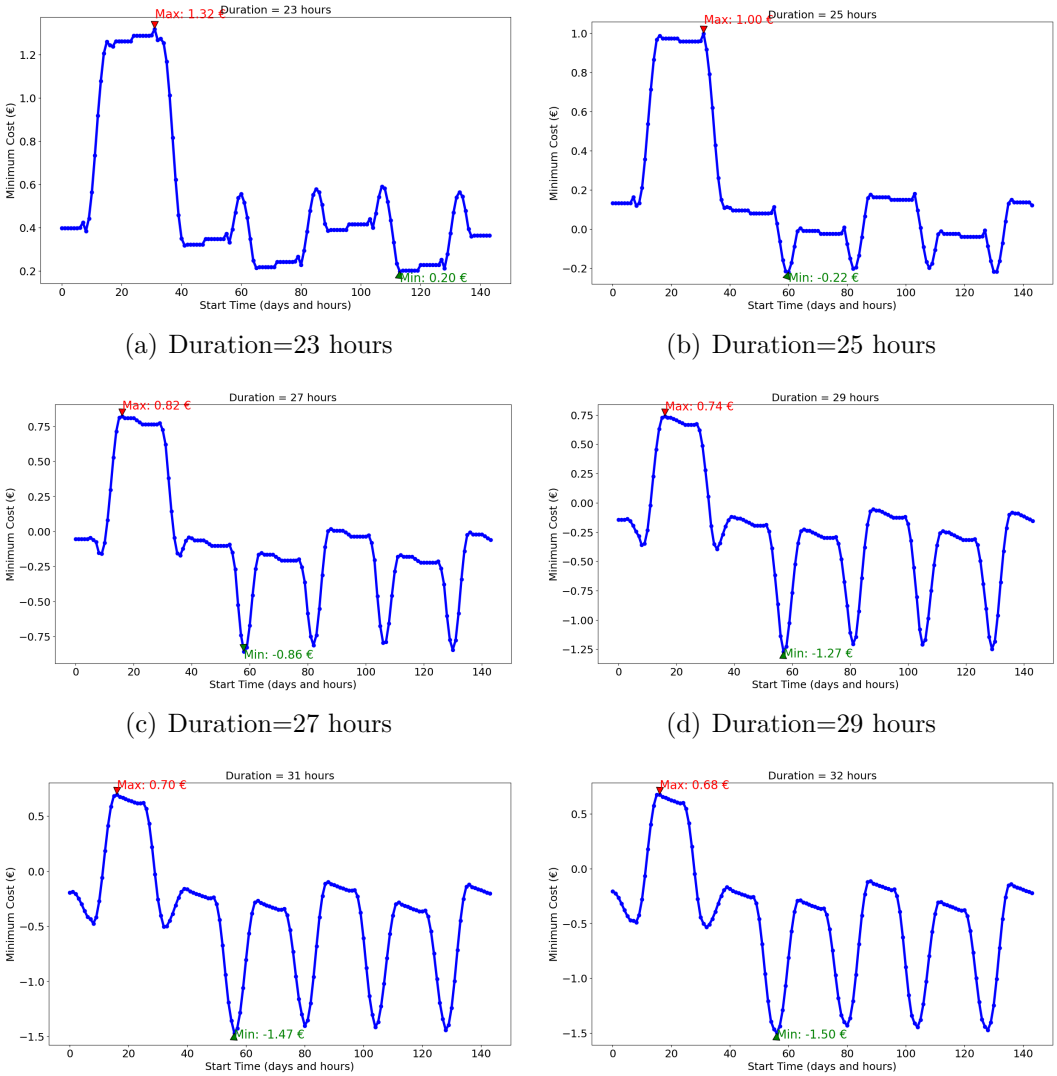
time condition, the maximum and minimum costs are 1.32 euros and 0.20 euros, respectively, which decrease to 0.68 euros and -1.5 euros under the 32-hour completion time condition.

Furthermore, it can be seen that as the maximum completion time increases, the shape of the cost curve for starting the training task at different times also changes. Under the previously mentioned condition of a 23-hour maximum completion time, starting the training task at noon results in a local maximum cost. However, when the maximum completion time is extended to 25 hours, starting the training task around noon becomes a local minimum in terms of cost. This is because, when the maximum completion time exceeds 24 hours, starting the training task at any time will ensure that the task spans a full day, allowing it to receive energy from the solar panels throughout the entire day. In this scenario, starting the training at noon on the first day enables the utilization of more solar energy, leading to a local minimum in costs.

5.5 Summary of Energy dispatch Model

In summary, the analysis conducted using the energy dispatch model developed in this study reveals several key insights:

- **Extended Completion Time**
  - The longer the maximum completion time for the training task, the lower the average cost during the training process. This trend is evident from the comparison of maximum and minimum costs across different completion times.
- **Primary Role of Solar Energy**
  - During the daytime, solar energy generated by the solar panels serves as the primary power source. Efficient utilization of solar energy significantly reduces the overall training cost.
- **Optimal Utilization of Solar Power**
  - Ensuring that solar energy is fully utilized can drastically lower the total cost of training tasks compared to scenarios where solar energy is not efficiently harnessed. This is particularly apparent when the training task spans over a complete day, allowing maximum use of available solar energy.



(a) Duration=23 hours

(b) Duration=25 hours

(c) Duration=27 hours

(d) Duration=29 hours

(e) Duration=31 hours

(f) Duration=32 hours

Figure 5.7: Minimum cost under different start time of model training

By optimizing the start times and completion durations of training tasks to align with periods of maximum solar energy availability, the energy dispatch model demonstrates that training costs can be minimized effectively. This underscores the importance of integrating renewable energy sources into computational tasks to achieve cost efficiency and sustainability.

Chapter 6

Conclusion

6.1 Summary

In this thesis, suitable tools for measuring the energy consumption of GPUs during model training were identified. Various deep learning models were built from scratch, allowing for the testing of energy consumption across different layers during forward propagation and the relationship between energy consumption in different parts of the model training process (time to device, forward, backward). Simulation results showed that during the forward propagation phase of model training, convolutional layers consumed the majority of the total energy.

Next, multiple potential factors that could influence the energy consumption of deep learning models during training were proposed and analyzed individually using the controlled variable method. The analysis revealed the following:

- **Hardware Differences**
  - Different hardware, due to their manufacturing processes and design specifications, results in varying energy consumption when training the same model. Newer products tend to have lower power consumption.
- **Dataset and Model Architecture**
  - The energy consumption of different deep learning models varies with the dataset and is influenced by the model architecture and the number of MACs. For smaller models, the primary factor affecting energy consumption across different datasets is the number of MACs, rather than the number of training samples. However, for more complex models with larger MACs, the primary factor becomes the dataset size.

- **Batch Size**
  - Increasing the batch size generally reduces the energy consumption per epoch during training. The amount of energy reduction is related to hardware performance. When the batch size is small, meaning there are fewer data per batch and the hardware is not a bottleneck for training speed, increasing the batch size can significantly reduce energy consumption (up to 19.71%). However, as the batch size continues to increase, leading to more images per batch and increasing memory pressure, the hardware becomes a bottleneck for training speed. In this case, the reduction in energy consumption from increasing the batch size becomes smaller (maximum 0.87%).
    Furthermore, the setting of batch size has a significant impact on energy consumption (close to 20%). However, the energy consumption prediction model proposed in this thesis is based on fixed epochs and batch size settings. Therefore, the impact of this factor is not considered in the construction of the energy consumption prediction model.
- **Model MACs**
  - As model complexity and the number of MACs increase, differences in energy consumption between different models become more pronounced. Models with similar numbers of MACs tend to have similar energy consumption.
- **Model Structure**
  - For the same dataset, models with a multi-branch structure tend to consume more energy during training. However, similar to the relationship between MACs and energy consumption, this also exhibits a near-linear relationship.

Next, energy consumption prediction models for training were proposed, both considering and not considering accuracy.

- **Without Considering Accuracy**
  - A model was proposed to predict energy consumption based solely on the relationship between the model's MACs and its energy consumption. This approach provided a good approximation. An energy consumption prediction model was developed and validated. The validation results indicated that the prediction model could accurately predict the energy consumption of different models on new datasets, proving the feasibility of the prediction model.

- **With Considering Accuracy**
  - When considering accuracy, proposing a prediction model proved more challenging. This was due to the lack of a clear mathematical relationship and the significant differences in model performance across different datasets. Consequently, it was difficult to establish a prediction model that accounted for accuracy.

Finally, the variation in GPU running speed under different power limits was analyzed. Testing revealed that GPU exhibit different running speeds at different power limits, with a nonlinear relationship. At lower power limits, there is a linear increase in speed as the power limit increases. However, as the power limit approaches the GPU's maximum, the increase in speed slows and eventually stabilizes.

Based on this characteristic, an energy dispatch model was constructed to minimize the cost of GPU training tasks when multiple power sources are available. Simulation results showed that extending the maximum completion time allows the model to run at lower power limits, reducing GPU power consumption and maximizing the sale of excess solar energy back to the grid. This minimizes training costs. Conversely, when the maximum completion time is constrained, minimizing costs requires starting the training task under optimal weather conditions, ideally around 8 am, to maximize the use of solar energy during the training phase.

6.2 Future Work

Future work needs to consider energy consumption analysis and prediction for large models. This involves predicting the energy consumption during the training process of large models based on the physical characteristics and operating conditions of hardware devices, to better allocate resources in advance.

Additionally, in the energy dispatch section, future considerations will include the actual operating conditions of the power grid. This involves assessing the behavior of selling electricity from solar panels and drawing power from the grid based on the load conditions of the power grid at different times.

Bibliography

[1] Patterson D, Gonzalez J, Hölzle U, Le Q, Liang C, Munguia LM, Rothchild D, So DR, Texier M, Dean J. The carbon footprint of machine learning training will plateau, then shrink. Computer. 2022 Jun 28;55(7):18-28.

[2] Vieira, Leticia Canal, Mariolina Longo, and Matteo Mura. "From carbon dependence to renewables: The European oil majors' strategies to face climate change." Business Strategy and the Environment 32, no. 4 (2023): 1248-1259.

[3] Schwartz, Roy, Jesse Dodge, Noah A. Smith, and Oren Etzioni. "Green ai." Communications of the ACM 63, no. 12 (2020): 54-63.

[4] Berthelot, Adrien, Eddy Caron, Mathilde Jay, and Laurent Lefèvre. "Estimating the environmental impact of Generative-AI services using an LCA-based methodology." Procedia CIRP 122 (2024): 707-712.

[5] Lannelongue, Loïc, Jason Grealey, and Michael Inouye. "Green algorithms: quantifying the carbon footprint of computation." Advanced science 8, no. 12 (2021): 2100707.

[6] Dodge, Jesse, Taylor Prewitt, Remi Tachet des Combes, Erika Odmark, Roy Schwartz, Emma Strubell, Alexandra Sasha Luccioni, Noah A. Smith, Nicole DeCario, and Will Buchanan. "Measuring the carbon intensity of ai in cloud instances." In Proceedings of the 2022 ACM conference on fairness, accountability, and transparency, pp. 1877-1894. 2022.

[7] https://developer.nvidia.com/blog/inference-next-step-gpu-accelerated-deep-learning/

[8] Wang, Yuxin, Qiang Wang, Shaohuai Shi, Xin He, Zhenheng Tang, Kaiyong Zhao, and Xiaowen Chu. "Benchmarking the performance and energy efficiency of AI accelerators for AI training." In 2020 20th IEEE/ACM International

Symposium on Cluster, Cloud and Internet Computing (CCGRID), pp. 744-751. IEEE, 2020.

[9] Hong, Sunpyo, and Hyesoon Kim. "An integrated GPU power and performance model." In Proceedings of the 37th annual international symposium on Computer architecture, pp. 280-289. 2010.

[10] Tang, Zhenheng, Yuxin Wang, Qiang Wang, and Xiaowen Chu. "The impact of GPU DVFS on the energy and performance of deep learning: An empirical study." In Proceedings of the Tenth ACM International Conference on Future Energy Systems, pp. 315-325. 2019.

[11] Cai, Ermao, Da-Cheng Juan, Dimitrios Stamoulis, and Diana Marculescu. "Neuralpower: Predict and deploy energy-efficient convolutional neural networks." In Asian Conference on Machine Learning, pp. 622-637. PMLR, 2017.

[12] Dayarathna, Miyuru, Yonggang Wen, and Rui Fan. "Data center energy consumption modeling: A survey." IEEE Communications surveys & tutorials 18, no. 1 (2015): 732-794.

[13] He, Wei, Qing Xu, Shengchun Liu, Tieying Wang, Fang Wang, Xiaohui Wu, Yulin Wang, and Hailong Li. "Analysis on data center power supply system based on multiple renewable power configurations and multi-objective optimization." Renewable Energy 222 (2024): 119865.

[14] Gnibga, Wedan Emmanuel, Anne Blavette, and Anne-Cécile Orgerie. "Renewable energy in data centers: the dilemma of electrical grid dependency and autonomy costs." IEEE Transactions on Sustainable Computing (2023).

[15] Gu, Chonglin, Longxiang Fan, Wenbin Wu, Hejiao Huang, and Xiaohua Jia. "Greening cloud data centers in an economical way by energy trading with power grid." Future Generation Computer Systems 78 (2018): 89-101.

[16] You, Jie, Jae-Won Chung, and Mosharaf Chowdhury. "Zeus: Understanding and optimizing GPU energy consumption of DNN training." In 20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23), pp. 119-139. 2023.

[17] https://codecarbon.io/

[18] https://developer.nvidia.com/system-management-interface

[19] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems 25 (2012).

[20] https://www.kaggle.com/datasets/zalando-research/fashionmnist

[21] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.

[22] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[23] Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).

[24] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1-9. 2015.

[25] https://huggingface.co/datasets/uoft-cs/cifar100

[26] https://github.com/sovrasov/flops-counter.pytorch

[27] https://huggingface.co/datasets/uoft-cs/cifar10

[28] https://github.com/dankamongmen/GreenWithEnvy

[29] https://pvwatts.nrel.gov/pvwatts.php

[30] https://www.eia.gov/todayinenergy/detail.php?id=46756

[31] https://www.monolithicpower.com/en/learning/mpscholar/battery-management-systems/introduction-to-battery-technology/battery-parameters