The Need for Energy-Efficient Machine Learning: A Review

By Tom Potter
(Student ID: 21182741)

Abstract

The field of machine learning offers significant potential in delivering computational innovations across science and industry, with cutting-edge systems using advanced methods, such as deep neural networks, to improve performance across the board every year. However, much of this work strives for competitive performance on global leaderboards without considering consequences further afield. This has led to today's models inflicting a significant environmental, financial, and social cost. The subsequent review consolidates these impacts, describing the current issues surrounding machine learning systems; suggested hardware and software solutions aiming to minimise this cost are then explored, demonstrating how the energy efficiency and sustainability of advanced machine learning models can be improved.

Contents

1 Introduction **3**
- 1.1 Background . . . 3
- 1.2 Overview of Review . . . 4

2 Literature Review **5**
- 2.1 The Issue of Energy Consumption . . . 5
  - 2.1.1 Quantifying the Issue . . . 5
  - 2.1.2 Environmental Impacts . . . 6
  - 2.1.3 Social Impacts . . . 7
  - 2.1.4 Research Issues . . . 7
  - 2.1.5 Benefits of Red AI . . . 8
- 2.2 Promoting Energy Efficiency . . . 8
  - 2.2.1 Reporting Resources . . . 8
  - 2.2.2 Efficiency Metrics . . . 9
  - 2.2.3 Designing Efficient Systems . . . 9
  - 2.2.4 Data Efficiency . . . 10
  - 2.2.5 Quantifying Energy Usage . . . 10
- 2.3 Improving Energy Efficiency through Hardware . . . 11
  - 2.3.1 GPU Accelerators . . . 11
  - 2.3.2 TPU Accelerators . . . 12
  - 2.3.3 Efficient Hardware and Green AI . . . 12
- 2.4 Improving Energy Efficiency through Software . . . 13
  - 2.4.1 Efficient ML Packages . . . 13
  - 2.4.2 Lightweight Network Architectures . . . 14
  - 2.4.3 Lightweight Operations . . . 15
  - 2.4.4 Progressive Training . . . 16
  - 2.4.5 Transfer Learning . . . 16
  - 2.4.6 Hyperparameter Optimisation . . . 17
  - 2.4.7 Reducing Precision and Accuracy . . . 18
  - 2.4.8 Data Efficiency . . . 18
  - 2.4.9 Future Directions . . . 19

3 Future Work **19**

4 Conclusions **20**

1 Introduction

The field of machine learning (ML) has long been deeply synonymous with cutting-edge computational innovations, pushing the boundaries of what is deemed possible by computer systems. This has been the case since the inception of machine learning, with early systems such as *Samuel's Checkers Player* (Samuel, 1959)—the first program that could learn to play the board game checkers—defining what we now refer to as *artificial intelligence*. In recent years, machine learning has become ubiquitous in almost every area of computing, with ML research producing cutting-edge results across the board: from Meta AI's *M2M-100* becoming the first language processing model that can directly translate between any pair of 100 different languages from across the globe (Fan, Bhosale, Schwenk, Ma, El-Kishky, Goyal, Baines, Celebi, Wenzek, Chaudhary, Goyal, Birch, Liptchinsky, Edunov, Grave, Auli and Joulin, 2020), to DeepMind's *AlphaFold* becoming the first computational method to solve the complex *protein folding problem* that has challenged Biologists for years (Jumper et al., 2021).

The vast majority of these innovative contributions to ML research prioritise optimising model performance solely on accuracy-based metrics, competing for dominance on leaderboards such as the *ImageNet Benchmark*. In one of the most extensive evaluations of the negative impacts of performant ML systems to date, Schwartz et al. (2020) surveyed sixty recent state-of-the-art ML papers, discovering that 90% of ACI papers, 80% of NeurIPS papers, and 75% of CVPR papers cite an accuracy improvement as the main contribution of their work. Whilst some of these advancements have been induced by innovations in model architectures and methods, many typically rely upon increasing computational resources and larger models, such as large neural network architectures, extreme dataset sizes, and lengthy network training processes. However, these large computational loads do not come for free; research has shown an increased reliance on energy-intensive hardware and high-complexity software has a significant financial, environmental, and social cost. Furthermore, this is a trend likely to continue; Devlin et al. (2019) note that while scaling computational loads continue to produce "strictly increasing results" the development of resource-hungry performant ML models is unlikely to down scale. Bender et al. (2021) share a similar view, asserting that while performance boosts continue, further resource scaling is inevitable, summarising the current direction of ML research as following a philosophy of "there's no data like more data". Hence an increased focus on the energy efficiency of leading ML research is necessary to mitigate these negative impacts.

1.1 Background

Machine learning (ML) is a form of *artificial intelligence* (AI) focused on developing computer programs that learn how to perform certain tasks directly from data, without any human intervention (Samuel, 1959). To achieve this goal, models are developed through a *training* process during which they learn how to perform a task T from some provided experience E (extracted from the *training dataset* consisting of many instances E_i). During *inference*, these trained models can subsequently be applied to additional data not exploited during the training process to infer novel results. Typically, this involves the evaluation of a *prediction* task, where the future value of a given variable is inferred, or a *classification* task, where input data points are assigned labels corresponding to a distinct class. Once training has been completed, the learning process can be judged by evaluating some *performance metric*: for example, the *accuracy* of a model is a commonly chosen metric, assessing the proportion of correct predictions or classifications compared to the total number of inferences made.

Recently, much work in the field of ML has favoured *deep learning* (DL). DL methods take inspiration from the human brain, constructing an *artificial neural network* (ANN) that processes information through a collection of internal neurons arranged in *layers* and connected

through *weighted channels*. The arrangement of these neurons and layers is known as the *network architecture*. Early neural networks relied upon shallow architectures consisting of only a few layers, such as the *structured perceptron* of McDonald et al. (2010); however, recent work has pushed the boundaries of *deep neural networks* (DNNs) that use larger architectures with many internal layers and parameters to perform more complex and intensive computations. The design of these architectures is typically chosen through a process known as *hyperparameter optimisation* (HPO) where *neural architecture search* (NAS) is conducted to find the most beneficial formulation of neurons and layers for the task at hand. Schwartz et al. (2020) classify the resulting architectures from this search into two categories: *Red AI* and *Green AI*. Red AI concerns those models that employ large, computationally-intensive DNNs to push state-of-the-art accuracy at the expense of other financial, environmental, and social impacts. Alternatively, Green AI attempts to rebuke this trend, prioritising energy-efficient ML models that generate novel results without a significant computational cost.

Two popular fields within DL are *natural language processing* (NLP) and *computer vision* (CV), both of which focus on executing incredibly computationally expensive tasks. NLP encompasses a wide spectrum of ML modelling applications that focus on the processing and understanding of language: for example, *language models* (LMs) make predictions about the characters and words in a text to conduct text generation and classification (e.g. Cer et al. (2018)). CV is a similarly expansive field, exploring the derivation of information from visual inputs, allowing computers to *see* a particular domain; for example, the classification of subjects within an image (e.g. Mahajan et al. (2018)), or the mapping of an environment for autonomous vehicles. ML models in CV typically rely upon *convolutional neural networks* (CNNs), a variant of the DNN specialied to image analysis, sequentially applying *convolution* filters to break down an image into its features. Due to the large knowledge bases these methods are learning from, such as entire dictionaries of words or databases of images, the DNNs exploited during NLP and CV often have an incredibly high computational cost. However, likely for this reason, these two fields constitute much of the current research into Green AI, withmultiple surveys such as Bender et al. (2021) highlighting the impacts of large NLP models, and works like Iandola et al. (2016) attempting to reduce the computational load of CV network architectures.

Mobile computing and *edge computing* are two further domains within which the goals of Green AI are ubiquitous, sharing many similar ideas about the energy efficiency and resource load of ML. Mobile computing focuses on the computational practices utilised in mobile devices, and edge computing concerns the local processing of data, typically onboard the device it was collected without relying upon external cloud computing servers. Both of these applications typically require ML algorithms and ANNs to be implemented on small, embedded devices with limited power and memory budgets. Hence, whilst not explicitly sharing the sustainability motivations of Green AI, many concepts and developments improving the energy efficiency of ML are commonly referenced and explored in these fields.

1.2 Overview of Review

In this review, the current landscape surrounding energy-efficient technologies and practices for machine learning will be surveyed. Initially, the motivations surrounding the development of energy-efficient ML and Green AI will be explored, tackling how the problems associated with state-of-the-art ML models—including their financial, environmental, and social impacts—can be quantified. Current trajectories tackling the energy efficiency of ML systems will then be analysed. This will begin by focussing on how passive and retroactive measures can help reduce the impacts of ML research, such as the reporting of efficiency metrics, use of quantification tools, and promotion of efficiency-centric research. Then approaches that directly offer energy-efficient adaptations and alternatives to Red AI will be explored, starting with possible changes

to ML hardware that promise to reduce the computational load of training and inference, and later extending to software implementations that aim to produce more lightweight model architectures, training algorithms, and datasets. This exploration will conclude by identifying the gaps in research remaining within the field of energy-efficient ML, and an insight into the future direction of my research in this field.

2 Literature Review

2.1 The Issue of Energy Consumption

2.1.1 Quantifying the Issue

Recently, researchers in the ML fields such as NLP have begun to highlight this trend of scaling resources, quantifying its impact. Bender et al. (2021) have conducted one such study, analysing the changes to state-of-the-art models in recent years. They discovered that the 2019 model *BERT* (Devlin et al., 2019) used 340 million parameters and a 16GB dataset to generate class-leading performance, whereas only two years later the leading model Switch-C (Fedus et al., 2021) required 1.57 trillion parameters and a 745GB dataset. Whilst analysing the relationship between deep learning and computing power, Amodei and Hernandez (2021) assert that the amount of processing necessary to train cutting-edge DNNs (measured in PetaFPO/s-days) has increased by a factor of 3×10^5 in the six years between 2012 and 2018. The authors put this into context against the two year doubling period of Moore's Law—which was witnessed in pre-2012 computing power—showing that recent trends instead suggest the resources required by state-of-the-art DNNs are exponentially increasing with a 3.4-month doubling time.

Referring to these performance-driven ML systems as *Red AI*, Schwartz et al. (2020) quantify their computational load through Equation 1, breaking down the total cost of training a ML model M as being proportional to 1) the cost of evaluating a single training instance E, 2) the size of the training dataset D, and 3) the number of hyperparameters H being optimised during HPO.

$$Cost(M) \propto E * D * H \tag{1}$$

The computational demand of modern ML techniques can be analysed from these three perspectives. Firstly, the computational cost of evaluating each training data instance rises proportionally to the depth and parameter count of the implemented DNN. Namely, deeper models require longer training times and draw more energy from their hardware, as both passing data through the network's layers and optimising the value of each parameter are more computationally demanding. Schwartz et al. (2020) exemplify how computationally demanding deep networks can be through DeepMind's *AlphaGo* (Silver et al., 2016), the first system to exhibit super-human performance in the board game Go. This DNN-based system required 1920 CPUs and 280 GPUs to play each game, with the total training process requiring 430 million evaluations of data instances. However, it is important to note that more recent implementations of this system, such as *AlphaGo Zero* (Silver et al., 2017), have improved upon this computational efficiency. Secondly, Schwartz et al. (2020) share the view of Bender et al. (2021) that significant increases in the size of datasets utilised by performance-driven models are a major contributor to their increased computational cost. This trend is demonstrated by models such as Mahajan et al.'s image classifier, which exploited 3.5 billion images to produce record-breaking accuracy (evaluated through the *ImageNet top-1 accuracy* score)—a training set three orders of magnitude larger than previous leading models (which typically utilised the *Open Images Dataset* of 9 million instances). Thirdly, the large number of hyperparameter optimisation experiments

exploited to find the most performant architecture for the given task, such as Google's language model (Cer et al., 2018) which searched over 12,800 different architectures, is cited as another significant source of computational cost in terms of both time and energy. Whilst these three aspects form the major culprits when identifying the cause of the increased computational demand of recent cutting-edge ML systems, it's important to note that they are not the sole causes. In particular, Lacoste et al. (2019) raise the type of hardware on which training is implemented and the location of the cloud computing and data servers relied upon as key factors in determining the cost of ML models.

2.1.2 Environmental Impacts

As the computational cost of a ML model increases, the energy it draws also scales. Due to the majority of computing centres and energy providers not solely relying upon carbon-neutral energy sources (Strubell et al., 2019), this energetic cost comes alongside a significant impact on the environment, a factor that has increasingly been raised by concerned researchers. Through inspecting recent NLP models, Strubell et al. (2019) exemplified this connection between computational cost, energy consumption, and carbon emissions. In their paper, they estimated the power consumption of models such as BERT (Devlin et al., 2019); from this, the electricity cost and CO_2 emissions were approximated, contextualising the substantial emissions associated with ML: for example, GPU-based training of BERT was estimated to release 1438lbs of CO_2, which roughly equates to the same emissions as a trans-American flight.

However, the energy consumption associated with high-performance computing applications such as ML research has been shown to range significantly between different cloud computing providers: a (2017) Greenpeace report found that whilst 56% of *Google Cloud*'s estimated electricity demand is powered by renewable energy, *Amazon Web Services* has an equivalent figure of only 12% (Cook et al., 2017). Since this report, Google claims to have offset 100% of their annual energy demands every year with renewable sources—declaring in 2021 that they had retroactively offset the carbon footprint of the company's entire operating history (Google Sustainability, 2021). Alongside this report, Google Cloud released a breakdown of the average percentage of carbon-free energy used by each of its computing centres, showcasing that this ranges between an impressive 94% (Finland) to a disappointing 4% (Singapore) and hence demonstrating that whilst the current sustainability trajectory is promising, they still have a lot of work to do. Furthermore, through cross-referencing the locations of computing servers and the resources used by them, Lacoste et al. (2019) demonstrate how carbon emissions vary depending on the geographical location of the resources used by a ML model. Their data highlights the variation of carbon emissions associated with particular regions: for example, North American computing servers emit anywhere between 20 and 737 grams of CO_2 per kWh of energy. Hence, large efficiency gaps between servers can contribute to significant differences in carbon emissions over the lengthy training procedures used by performant ML models.

Schwartz et al. (2020) note that a large proportion of the companies claiming to rely upon carbon-neutral energy only achieve this through purchasing carbon credits, arguing that this has a negligible impact on improving their sustainability on the whole. Bender et al. (2021) similarly argue that the renewable resources used by computing providers have a continued environmental impact, referencing a 2020 article by *The Herald Scotland* that highlights how 14 million trees have been cut down across Scotland since 2000 to erect wind farms. However, the points raised by this article have been contested: *Scottish Forestry* figures show that within this same time frame the number of trees planted vastly outnumbered those felled. Additionally, Bender et al. (2021) question the ethics behind utilising a significant portion of the limited clean energy resources currently available solely for computing tasks, suggesting that these use cases could be taking away from other potential applications that would provide greater benefits

to communities and the environment: for example, in 2017 Microsoft purchased 100% of the renewable energy produced by the Vattenfall wind farm in the Netherlands—a site that could have greatly benefitted local communities (Microsoft News Center, 2017).

To quantify and compare the varying sustainability of different computing centres, Lacoste et al. (2019) suggest evaluating their Power Usage Effectiveness (PUE); this metric quantifies the proportion of energy used for overhead tasks like cooling and power conversion instead of on the computational task at hand. *The Uptime Institute* approximates the average PUE for the largest data centres at 1.59, meaning for every watt of power used for computing, an additional 0.59 watts are expended on overhead costs (Bizo et al., 2021). This suggests that a significant part of the energy drawn by ML systems is wasted on auxiliary tasks. However, this is also a factor that ranges drastically between different centres: for example, Google Cloud boasts an average annual PUE of 1.1 (Google Sustainability, 2021), a 31% improvement on the Uptime Institute's approximated average.

It is important to remember that computing hardware alone is not the sole cause of ML research's unsustainability. Bender et al. (2021) note that for many models the inference process can be equally, or even more, energy-intensive than training. Jain et al. (2019) reiterate this view, estimating that 90% of the computational cost associated with DNNs can be attributed to inference, not training. Additionally, Al-Jarrah et al. (2015) highlight that the data centres utilised in experiments, such as the 3.5 billion images utilised by Mahajan et al. (2018), are a huge hidden contributor to carbon emissions. Concern is raised over how the significant increase in data requirements of Red AI is rapidly increasing both the number of servers in each data centre (e.g. disks and solid-state drives used for storage) and the electricity drawn by each server. However, it is important to remember that many researchers typically reuse the same datasets (such as Google's Open Images Dataset), meaning the cost of storing this data is shared between its users, spreading the environmental impact of data centres.

2.1.3 Social Impacts

Both Schwartz et al. (2020) and Strubell et al. (2019) raise concerns about the social impacts of increasing the cost of ML. They argue that as resource loads scale, the financial cost is prohibitively expensive, limiting who can engage in cutting-edge research and reducing the breadth of ML research. For example, it has been estimated that the experimental cost of DeepMind's AlphaGo Zero (Silver et al., 2017) totalled approximately $35 million (Schwartz et al., 2020). This lack of accessibility accelerates the "rich get richer" cycle of research funding (Strubell et al., 2019), stifling creativity and hindering the number and type of research opportunities available. Moreover, Bender et al. (2021) consider the imbalance between those who benefit from state-of-the-art models and those who bear their environmental effects. They highlight that climate change more rapidly and severely affects the world's most marginalised communities (United Nations Department of Economic and Social Affairs, 2016), questioning "is it fair or just, for example, that the residents of the Maldives [...] pay the environmental price of training and deploying ever larger English LMs".

2.1.4 Research Issues

Additionally, research has suggested that relying on scaling resources alone to deliver performance enhancements is not adequate for the continued advancement of ML performance. Both Schwartz et al. (2020) and Buchanan and Tachet des Combes (2021) highlight the diminishing returns of this approach; they affirm that current trends show model sizes are growing at a rate that far exceeds their performance gains. Mahajan et al. (2018) extend this analysis, suggesting there exists a logarithmic relationship between model accuracy and training dataset size. Furthermore, it's asserted by Al-Jarrah et al. (2015) that current ML methods do not deal with

this significant increase in data load intelligently or efficiently, resulting in significant energy wastage.

Walsh et al. (2021) add that a large proportion of this energy is used only to get the final bits of performance out of a system, as the accuracy of a model plateaus after the initial stages of training. To exemplify this, the authors train a simple model over the *Iris dataset*: they found that to gain a training accuracy of 96.17% only 964 joules of energy were consumed, but improving the accuracy from here to 98.67% required an additional 15,077 joules (15 times more than the initial stage). Strubell et al. (2019) examine So et al.'s language model to demonstrate a similar conclusion: this model improves state-of-the-art accuracy in English-German translation by 0.1 *BLEU*, however, training the model is approximated to cost up to $3.2 million and generate 626,255lbs of CO_2 (nearly five times the emissions associated with the average car over its lifetime). Bender et al. (2021) take this further, arguing that there is little evidence to suggest that these small performance increases bring the field any closer to the goal of *general AI*, instead only teaching models to "cheat [their] way through tests" through brute-force processing.

However, it's important to note that accuracy metrics can be misleading, as the seemingly small accuracy improvements at the final stages of training do not necessarily only reflect a small boost in real-world performance. This is because accuracy levels around 90% can make a system seem deceptively intelligent: for example, a model making a binary classification of the digits 0-9 as 'zero' or 'non-zero' will generate 90% accuracy with only a trivial model that solely guesses 'non-zero' on all instances (as digits 1-9 will always be non-zero).

2.1.5 Benefits of Red AI

Despite these issues, both Al-Jarrah et al. (2015) and Schwartz et al. (2020) highlight that there are several applications where ML has been used to benefit environmental conservation and sustainability: for example, ML models have been used to improve the efficiency of harnessing renewable energy sources (Daniel et al., 2021), reduce CO_2 emissions in industries such as cement manufacturing (Acharyya et al., 2019), and aid the development of sustainable products (Marwah et al., 2011). Al-Jarrah et al. (2015) use such examples to explain that improving energy efficiency would not only reduce the impacts of ML research but further the scope and impact of these sustainable applications.

2.2 Promoting Energy Efficiency

Despite the highlighted issues associated with Red AI, the trend of exponentially increasing resource requirements is still common across ML research, and only appears to be accelerating further. However, in recent years several proposals have been made that attempt to improve the energy efficiency of ML models; these range from promoting increased transparency and an improved development process, to hardware and algorithmic innovations directly minimising a model's energy consumption over its lifetime. Schwartz et al. (2020) describe such methods as *Green AI*. Whilst surveying the current landscape of Green AI, Xu et al. (2021) describe its motivations as threefold: mitigating the environmental and financial costs of ML, making ML research more inclusive, and improving performance on low-resource edge devices.

2.2.1 Reporting Resources

A practice raised by multiple proponents of Green AI to raise awareness about the impacts of ML research is an increased transparency around the computational resources used to generate state-of-the-art results. Almost all papers addressing the issues of energy inefficiency in leading

models highlight transparency as an important factor in raising awareness about the environmental, economic, and social impacts of ML. Strubell et al. (2019) argue that reporting training time and resource load of models can facilitate a cost-benefit comparison between models and allow more informed decisions to be made during the design process of new implementations. They also argue that reporting how the performance of proposed models scales with training dataset size allows future models using smaller training budgets to be compared, allowing these lower complexity models to compete, decreasing training costs, and increasing accessibility.

2.2.2 Efficiency Metrics

In their analysis, Strubell et al. (2019) note that a standardised hardware-independent measure of training cost is necessary to properly evaluate model efficiencies. Similarly, Buchanan and Tachet des Combes (2021) identify that efficiency metrics, whilst being crucial to improving the sustainability of ML, have not received the same attention as accuracy metrics, and work is needed to expose a globally viable measure. This is nicely summarised by Schwartz et al. (2020), who assert "you can't improve what you can't measure".

Several promising efficiency metrics are later explored by Schwartz et al. (2020), who settle upon *Floating-Point Operations count* (FPO) as the most reliable and informative measure. FPO reports the total count of floating-point operations (i.e. addition and multiplication operations) that a program uses to generate a result. This metric measures the computational work done by a program, allowing any abstract ML function to be objectively compared; it is also directly correlated to model runtime, penalising inefficient models that are executed over long periods, and has been successfully utilised by Veniat and Denoyer (2017) to quantify the energy footprint of DNNs. Schwartz et al. (2020) demonstrate the benefit of reporting FPO through analysis of two competing models: *ResNet* (He et al., 2015) and *ResNeXt* (Xie et al., 2016). Whilst ResNeXt improves the *ImageNet top-1 accuracy* of ResNet by 0.5%, it exhibits a 35% increase in FPO, contextualising the performance boost provided by this model against its computational cost. However, it is important to note the limitations of utilising FPO as a universal efficiency metric; Veniat and Denoyer (2017) highlight that FPO can be misleading as it does not consider additional computational tasks performed by a system, such as communications or memory accesses, and so does not always exactly correlate with elapsed runtime or energy consumption. Likely for this reason, the field is still fragmented when it comes to choosing a universal metric; in the same year as Strubell, Lacoste et al. (2019) propose reporting CO_2-equivalents, a measure enumerating the amount of CO_2 that would have the equivalent impact on global warming as the energy expended to train a particular model.

2.2.3 Designing Efficient Systems

Whilst several surveys highlight the need for improving the sustainability of ML research, little has been done by those on the bleeding edge of ML innovations. At the time of his discussion of sustainable data modelling practices, Al-Jarrah et al. (2015) describes how very limited research had been conducted on efficient ML methods, citing only Yoo et al.'s clustering model and Cheng et al.'s support vector machine-based model as noteworthy systems prioritising energy-efficiency, neither of which relied upon DNNs. Whilst the field has expanded since then, these works typically remain on the fringes of the ML research landscape, such as in mobile devices with low-energy components. For this reason, Bender et al. (2021) emphasise that it's "past time" for energy efficiency to become a core consideration during the development of large-scale ML systems, calling for energy, environmental, and financial costs to become a major factor in design decisions. They also suggest that *pre-mortem* evaluation exercises (Klein, 2008) would be an effective method of analysing the construction of ML systems to undertake this

goal. Along these lines, Walsh et al. (2021) assert that a compromise must be met between accuracy and energy consumption, determined by the nature and priority of the work; medical image classification, for example, is raised as a domain where accuracy is critical and thus should still be prioritised. Similarly, Lacoste et al. (2019) advise that when designing an ML system, the environmental impact of varying techniques should be considered: for example, Lacoste suggests that during hyperparameter optimisation a random search should be chosen over a grid search, as it is proven to be more efficient and still generate effective results. Walsh et al. (2021) compare this accuracy-efficiency compromise to the common software engineering compromise when searching for software bugs, where the risk of a bug is balanced against the time and effort associated with removing it. Hence, it is likely this compromise would be an easy and widely accepted addition to the development process that could aid in increasing the efficiency of ML systems.

2.2.4 Data Efficiency

Al-Jarrah et al. (2015) additionally identify data curation and processing as an area where increased efficiency would drastically improve the sustainability of ML. Bietti and Vatanparast (2019) and Bender et al. (2021) share this view, explaining that current systems contain a significant amount of wasted data; they assert that more time should be spent carefully curating training data instead of simply ingesting expansive datasets, most of which is not beneficial for the learning process and only wastes time and energy. For this reason, Bender et al. (2021) suggest that researchers should document the amount, properties, and risks of data used during training.

2.2.5 Quantifying Energy Usage

Due to the lack of a cohesive, globally viable efficiency metric, several varying tools quantifying the energy consumption of a ML system have been proposed. These range from real-time online benchmarking tools, such as Lottick et al.'s analysis package for Python (that interprets energy consumption to approximate carbon emissions) and Henderson et al.'s *experiment-impact-tracker* framework (for summarising energy, carbon, and compute costs on Linux systems), to advisor tools like Walsh et al.'s *Green Cloud Advisor* that highlights the sustainability implications of programmatic designs.

Lacoste et al. (2019) recognise that the most effective way developers can begin to improve the sustainability of their work is to be informed about the carbon emissions associated with certain designs and implementations; for this reason, they propose the *Machine Learning Emissions Calculator* to help developers understand the environmental impacts of their models (outputting the CO_2-equivalent of a system). However, Lacoste demonstrates a few notable limitations of their work: they highlight the margin of error introduced by a lack of transparency from organisations and governments about the energy used by their infrastructure and its carbon impacts, as the tool relies heavily upon publicly available data. For this reason, it is likely that organisations with more in-depth access to information about the energy and computing infrastructure used by the resources relied upon by ML models, such as the companies running data centres, would be more qualified to produce such quantification tools. This is a core view of Dodge et al. (2019), who believe large cloud computing companies must do more to report the energy costs of their resources, highlighting to users the hidden costs of their computing and data services. The resource metrics included in Microsoft's *Azure Machine Learning* (AzureML) platform are a good example of such a system, providing enterprise-grade metrics analysing the computational costs (e.g. total CPU/GPU and memory utilisation, and energy expended) of programs hosted on Microsoft's servers. Microsoft claims that by providing developers with the ability to measure the computational cost of their work, they aim to

compel ML developers to make more effective use of the resources available to them, reducing their environmental impact (Buchanan and Tachet des Combes, 2021). *The Allen Institute*, a key proponent of Green AI, views Microsoft's tools as a key milestone toward the "goal of more transparency around the power usage required by computation" (Dodge et al., 2019), allowing developers to make crucial adjustments to improve the sustainability of their ML models. However, it is important to highlight the deep connections between the Allen Institute and Microsoft, which were both (co-)founded by Paul Allen and worked in partnership to develop these energy quantification tools; hence, their endorsement should be considered with caution. Despite this, the utility of quantification tools such as Microsoft's to addressing the energy inefficiency of ML is undeniable.

2.3 Improving Energy Efficiency through Hardware

Throughout history, pushing the performance limits of processing hardware has been at the forefront of the computing industry; this is exemplified by Moore's Law, a commonly cited observation that the per-chip transistor count of high-end computer hardware is exponentially increasing, doubling every two years (Amodei and Hernandez, 2021). As demonstrated by Lacoste et al.'s analysis of scaling resource loads, a similar story is true of ML hardware; hence, concern must be raised about the impacts of increasing energy consumption. Motivated by this issue and a need for improving the cost-performance efficiency of high-end computing hardware, a recent push into specialised chips for ML has been seen. Both Lacoste et al. (2019) and Walsh et al. (2021) argue that specialised ML hardware is a major area of improvement for increasing energy efficiency and reducing financial and carbon costs.

2.3.1 GPU Accelerators

Training and inference of ML models typically involve computing similar multiplication and summation operations over many different data and parameter instances. The most fundamental of these operations is the *multiply and accumulate* (MAC) function of neural networks, where to compute the value taken by neuron n in layer $i+1$ a sum is computed over the values of all neurons connected to n in the previous layer i, each multiplied by the weight of the connection between these neurons (Burr et al., 2021). These MAC operations are implemented as matrix-vector products between a matrix of weights and a vector of neuron values; therefore, improving the speed and efficiency of matrix-vector products is a core focus when attempting to improve the energy efficiency of ML through hardware.

These operations can be accelerated through the use of *Graphics Processing Units* (GPUs); these processors split workloads into smaller tasks and exploit parallelisation to increase speed and efficiency (Kumar et al., 2020). In their analysis, Lacoste et al. (2019) demonstrate how using tuned domain-specific GPU hardware can provide significant efficiency benefits: for example, *Jetson AGX Xavier* GPUs (Seznec et al., 2021) can be 10-20 times more efficient than an equivalent CPU for certain tasks (in terms of peak FPO per watt of power). Similarly, Walsh et al. (2021) assert that a renewed focus on the development and improvement of hardware accelerators such as GPGPUs (*General Purpose GPUs*) should be given, as these offer higher performance and lower energy consumption than traditional approaches. However, whilst individual GPUs benefit energy efficiency, current state-of-the-art models often train over many GPUs in tandem; this can multiply the energy and carbon cost of such systems, negating efficiency improvements (de Leeuw, 2021).

2.3.2 TPU Accelerators

Despite the performance and efficiency gains of GPU hardware, the exponential increase in the computational load of ML models has put increased pressure on these systems. For this reason, Google noticed the need for purpose-built hardware accelerators for DNNs, advancing the performance and reducing the cost of computations beyond what's possible with GPUs (Jouppi et al., 2017). To achieve this, Google designed the *Tensor Processing Unit* (TPU) which split the workload of neural networks into parallelisable processing units, using on-chip memory and high data throughput to improve the efficiency of matrix multiplications. The chip minimised the distance between data storage and processing areas, reducing the energy expended on reading and writing to memory, and improving the energy efficiency of computations (Kumar et al., 2020).

To exemplify the benefits of TPUs, Jouppi et al. (2017) compare the cost-performance ratio and energy proportionality of the 1st-generation TPU against an equivalent CPU (*Intel Haswell* CPU) and GPU (*Nvidia K80* GPU). By comparing the total performance per watt of power for the TPU, GPU, and CPU, Jouppi et al. found that whilst the GPU produces a 1.2-2.1 times improvement relative to the CPU, the TPU significantly exceeds this with a 17-34 times improvement. Hence, the study showcased the 14-16 times better performance-power ratio of TPUs compared to the GPUs, demonstrating the significant energy efficiency benefit of this updated processor.

Jouppi et al. (2017) also explored how factors such as thermal design affect the power consumed by processing units, analysing how energy consumption scales with workload size (known as the *energy proportionality*) in TPUs, GPUs, and CPUs. It was found that whilst the TPU utilises the lowest power of the three, it has poor energy proportionality: at 10% workload, 88% of the total power that would be used at 100% workload is consumed; in comparison, the GPU and CPU generated equivalent figures of 66% and 56% respectively. Hence, for less demanding tasks that do not require the full resources of the TPU, utilising more traditional computational hardware would likely prove to be more energy efficient. This reinforces Lacoste et al.'s call for selecting hardware contextually and optimising energy efficiency based on the requirements of the task at hand. Whilst analysing the performance of Google's 3rd-gen TPUs, Lacoste et al. conducted a similar comparison, determining that TPUs are approximately 4-12 times more efficient than comparable GPUs (in terms of FPOs/W): specifically, the TPU v3 could achieve 11.6 times more GigaFPOs/W than an *AMD RX480* GPU, and 4.2 times more than a *Nvidia GeForce RTX 2080 Ti* GPU. Whilst still exemplifying the impressive energy efficiency of the TPU, this provides a more realistic view of the current improvements associated with Google's latest processor in the context of more recent competition from GPU hardware. Namely, the 2016 TPU v1 used a 28nm fabrication process that matched the 28nm process of the 2014 Nvidia GPU used for Jouppi et al.'s comparison. However, the 16nm fabrication process of the 2018 TPU v3 falls behind the 14nm and 12nm processes of the AMD and GeForce offerings released in 2016 and 2018. Although this is set to change again with Google's TPU v4 chip announced in 2021, which claims to provide 2 times the computing power of the TPU v3 and utilises a 7nm fabrication process (Wang and Selvan, 2021).

2.3.3 Efficient Hardware and Green AI

Recent developments such as TPUs and GPUs demonstrate that hardware accelerators offer much promise to improve the energy efficiency of ML implementations, reducing their carbon emissions. However, it's important to remember that the majority of this work is driven by reducing financial costs—Jouppi et al. (2017) cite a reduction in data centre costs as the main driving factor behind developing the TPU. Thus, as improving environmental sustainability is not at the heart of their motivation, it is possible that these innovations may lead to even

more performance-heavy systems, using an increased energy capacity to simply exploit more resources. Therefore, it is important that the potential for improving the energy efficiency of ML provided by innovative hardware is combined with other techniques to promote Green AI, such as the quantification methods previously discussed.

2.4 Improving Energy Efficiency through Software

To take advantage of energy-efficient ML hardware, we must also use software that minimises its energetic cost. This is a core motivation of Kumar et al.'s research into energy-efficient ML for edge devices, which asserts that regardless of how efficient ML hardware becomes, inefficient ML models will still result in a high energy cost. Furthermore, in circumstances where hardware choice is out of the developer's control (such as individual researchers relying upon Google Cloud servers), energy-efficient software is even more important to mitigating environmental costs. Despite this importance, energy-efficient software solutions have only been widely addressed in edge computing applications, limiting overheating and improving battery life when exerting high computational loads on low-resource components. However, it is likely improvements within this domain can aid the development and promote the use of Green AI further afield by progressing the maturity of the field of energy-efficient software as a whole.

Kumar et al. (2020) assert that energy efficiency can be optimised in software through a variety of routes: the choice of programming language, the choice of compiler for that language, and the choice of methods being programmed. In the case of ML, the language and compilation choice mainly reduces to the choice of which ML framework will be used to implement models. Once this choice is made, the efficiency of the software is determined by which ML methods are relied upon (such as the choice of model architecture, training, and hyperparameter optimisation procedures), and hence developments that improve the energy efficiency of ML models and methodologies constitute the bulk of this field.

2.4.1 Efficient ML Packages

The efficiency of the framework through which ML methods are implemented is a core contributing factor to the energetic cost of a software solution. When developing ANNs for instance, there is a spectrum of packages providing network architectures and functions to utilise in ML research; however, each of these has varying simplicity, performance, and efficiency priorities. Google's *TensorFlow* (Abadi et al., 2016) and Meta's *PyTorch* (Paszke et al., 2019) are popular choices for such a framework, both providing a simple but flexible collection of models and tools for performing ML computations, training, inference, and debugging. In a recent comparison of the two platforms, Chirodea et al. (2021) highlighted the areas in which they are divergent: PyTorch offers a simpler, more user-friendly experience, with faster training and inference speeds, but sacrifices accuracy and number of available options; alternatively, TensorFlow offers more performant methods and customisability, but uses a more challenging interface and complex dataflow. This impacts the energy efficiency of the two frameworks, as the increased complexity and runtime of TensorFlow models (and a greater reliance upon external dependencies) incurs a higher energy cost. However, whilst PyTorch does support GPU acceleration, the close partnership between TensorFlow and Google Cloud allows an increased offloading of work to efficient GPU and TPU hardware, decreasing the cost of any dependence on external cloud computation. Furthermore, Google released *TensorFlow Lite* (David et al., 2020) for edge ML applications; this alternative framework prioritises high efficiency and low memory through optimised functions, reducing the overall energy cost of ML models. This direction is promising for the field of Green AI, as whilst the changes exhibited in TensorFlow Lite are motivated by the low-power demands of edge devices, it is likely the reduced cost associated with these

energy-efficient methods will promote the usage of similar technologies further afield (such as within the main TensorFlow framework), as they not only reduce carbon emissions but also reduce financial costs for those running such frameworks. Additionally, the development of efficiency-focused ML packages outside of this duopoly, such as the DNN library *MXNet* (Chen et al., 2015) that prioritises minimising computation and memory loads, is likely to increase the pressure on Google and Meta to adopt more energy-efficient measures within their frameworks.

2.4.2 Lightweight Network Architectures

Schwartz et al. (2020) identify how the cost of passing an instance through a model directly affects how resource-intensive an ML system is, and Bender et al. (2021) highlight how increasing parameter counts is scaling energy costs; hence, it's clear ensuring ML models are as lightweight as possible is a promising path towards improving energy efficiency. The concept of lightweight networks was set in motion by Han et al. (2015), who moved away from the commonly used *fully-connected network* architecture of the time in search of more efficient neural networks. Motivated by reducing the computational complexity and memory requirements of DNNs, Han et al. proposed *network pruning*, a training process that built upon Hanson and Pratt's theory that not all network parameters are equally responsible for generating a result. Han et al.'s method took a pre-trained model and identified the importance of each weight to the network's performance; it then removed redundant connections that fell below an importance threshold, producing a network that maintained accuracy but reduced the computational and memory cost. Starting from the model *AlexNet* (the leading image classification model of the time, winning the 2012 *ImageNet top-5 accuracy* competition by an 11% margin), the process claimed to produce a 9.1 times reduction in parameter count from 61 to 6.7 million without any accuracy reduction. This inspired several similar approaches tackling the issue of intensive resource loads, such as *SqueezeNet* by Iandola et al. (2016). Motivated to improve efficiency for deployment in low-memory embedded systems, SqueezeNet achieved AlexNet accuracy (beating AlexNet's *ImageNet top-5* and *top-1* scores by 0.15% and 0.68% respectively) with a 50 times reduction in parameter count, requiring only 1.24 million parameters held in 0.47MB of storage. This was a dramatic reduction from the 240MB of storage required by AlexNet and 27MB required for Han et al.'s pruned model. Similarly, Mehta et al. (2018) demonstrated that lightweight CNNs could be run on edge devices to preserve memory and battery life whilst preserving performance. Their model *ESPNetv2* used 25 times fewer FPOs to achieve the same performance as *YOLOv2* (Redmon and Farhadi, 2016), a real-time object detention model that represented state-of-the-art performance only two years prior.

In their discussion of Green AI, Xu et al. (2021) reference *network pruning* as an effective way to reduce the energetic cost of a ML model, outlining the techniques and benefits of *structured* and *unstructured pruning*. The authors assert that *unstructured pruning* methods, where individual parameters are evaluated and removed, improve efficiency but result in sparse networks that make computing underlying operations more challenging and expensive. Xu et al. explain that *structured pruning*, where entire parameter blocks of the model are removed at a time, rectifies this issue by cutting entire parameter matrices, layers, and their associated operations from the network, producing a more significant improvement to efficiency. This technique is exemplified by Luo et al. (2019), who exploit structured pruning in their model *ThiNet-Tiny* to exceed AlexNet performance (by 0.23% and 0.31% in *ImageNet top-1* and *top-5* scores) with only 695 thousand parameters: an 88 times reduction over AlexNet, 9.6 times reduction over Han et al.'s model, and 1.8 times reduction over SqueezeNet. However, Xu et al. (2021) note that there are downsides to using networking pruning to improve efficiency; namely, the iterative computation of importance scores, and subsequent retraining necessary for a cohesive pruned model, introduce their own computational load that can build up and

partially negate the efficiency benefits introduced.

Beyond network pruning, Xu et al. (2021) categorise a variety of further approaches to developing lightweight models into two distinct classes: efficient network architectures, and efficient network operations. Efficient network architectures follow the approach of Han et al. (2015), attempting to improve energy efficiency by manipulating parameter configurations. One commonly used approach is *weight sharing*, where the same parameters are reused over multiple layers of a network, or multiple networks addressing different tasks. Due to weight sharings simplification of network design and implementation, and the computational savings generated, much research has favoured this method for reducing the computational complexity of DNNs. For instance, Dehghani et al. (2018) reused the same parameter set for all network layers in their *Universal Transformer*, aiming to improve the speed and generalisability of their model, and Plummer et al. (2020) proposed *neural parameter allocation search* as a method through which a network architecture can automatically learn the most efficient arrangement of shared parameters. These both proved successful applications of *cross-layer sharing*, however, *cross-task sharing* has proved more challenging and requires more complex implementations.

Hence, *dynamic weight sharing* methods were produced to tackle this complexity by exploiting flexible networks that execute different layers and operations dependent on the task at hand; *BranchyNet* (Teerapittayanon et al., 2017) and *SkipNet* (Wang et al., 2017) are helpful examples demonstrating work in this field. BranchyNet introduced the concept of *early-exiting networks* that exploit multiple internal predictors on each input to compute intermediate predictions; dependent on the evaluation of a given confidence metric, these predictions are then either output or passed to a subsequent predictor layer. This method allowed the network to produce outputs as soon as a confident result was observed, eliminating redundant computations and improving time and resource efficiency. SkipNet took a similar dynamic approach; this method used an additional policy network to inform whether data should be passed into the next layer of the network or whether this layer should be *skipped* (repeating this process on the layer after that).

2.4.3 Lightweight Operations

Xu et al. (2021) highlight that improving the energy efficiency of ML operations is another route to mitigating the cost of research. In particular, *convolution* and *softmax* operations have been identified as two of the most expensive aspects of ANN execution. The efficiency-focused models SqueezeNet (Iandola et al., 2016) and ESPNetv2 (Mehta et al., 2018) both experimented with adapted convolution operations to reduce their computational cost. Iandola et al. (2016) proposed the alternative *fire convolution* module, and Mehta et al. (2018) exploited a *depth-wise separable convolution*; these are two alternative convolution operations that split the application of the convolution filter intrinsic to CNNs into multiple consecutive steps to increase efficiency and reduce computational costs. Applying the softmax function (used to convert multiple discrete outputs into a probability distribution) is another area of notable computational load, its complexity scaling proportional to the number of output labels. This dependence on the label set means improving the efficiency of softmax is a task-dependent issue. For example, language modelling tasks with a high vocabulary require a large number of softmax operations (one per possible output word); hence, operations to reduce the output space (such as the sub-word level vocabulary of Sennrich et al. (2015)) have been proposed to reduce the cost of evaluating softmax. Additionally, research has been conducted into efficiency-focussed variants of softmax, such as the *hierarchical softmax* (Morin and Bengio, 2005) that relies upon a tree-like data structure to reduce computational complexity from $O(N)$ to $O(\log N)$ for a set of N labels.

2.4.4 Progressive Training

Whilst the efficiency of each component of the model is paramount to ensuring that each pass through the network expends as little energy as possible, minimising the number of passes executed during training is also important. This has led to the development of energy-efficient training algorithms, such as *progressive training*. Whilst this method was conceived by Akhand et al. (2008) and its potential advantages shown, it has only recently been extended to large-scale DNNs. Instead of collectively training all layers of a network at each training pass, progressive training constructs the architecture sequentially layer-by-layer, first training lower levels early on in the network (focussing on the low-level features of an input) then training later higher levels (for higher-level feature constructions) based on the parameters settled upon for those preceding layers. Hence, higher levels optimise parameter values faster as values are updated based on information already known about the features of the input. This method was utilised by Yang et al. (2020) to improve the training time of BERT (Devlin et al., 2019); their NLP model generated a speedup of over 110%, reducing the training time from 85 hours to 40 hours whilst maintaining equivalent performance (producing an accuracy of 84.3 on the *Multi-Genre Natural Language Inference* dataset, compared to BERT's score of 84.4). Similarly, Belilovsky et al. (2018) demonstrated that progressive learning can be applied to train deep CNNs to exceed the performance of AlexNet and approach that of *VGG-19* (Simonyan and Zisserman, 2014)—a successor to AlexNet that vastly increased the number of layers and parameters. Hence, progressive training for DNNs shows much promise in addressing the impacts of DL through reducing training times and consequently minimising the energy and carbon cost of learning effective models.

2.4.5 Transfer Learning

Another method for reducing the cost of training a new ML model is known as *transfer learning*. Analogous to the cross-task weight sharing discussed by Xu et al. (2021), transfer learning exploits a single network across a spectrum of tasks, discarding the need to learn a model from scratch. However, whilst cross-task sharing uses the same parameters across these varying tasks, transfer learning instead starts from a general *base network* (pre-trained over some base dataset and task) which is re-trained for use on a new task. The parameters of the new *target network* are initialised from this base network, and a short additional training round is used to repurpose the general features learned to the specific target task. This method was used by Wang et al. (2020) to repurpose the base model *ResNet-101* (He et al., 2015) originally trained on the large ImageNet dataset for use on the much smaller *Stanford Dogs 120* dataset (consisting of only 20,580 images compared to the 1.2 million of ImageNet). Their model had 11.07% higher accuracy than any other competing model on this dataset whilst using 10 times fewer operations. Although typically exploited for image classification problems, such as that of Wang et al. (2020), transfer learning has recently shown great promise within the field of NLP, where base models, like Google's *Universal Sentence Encoder* (Cer et al., 2018), learn general language representations that are applied by target models. Ruder (2018) and Conneau et al. (2017) both discuss the importance of these models to the field of NLP, asserting that the success of transfer learning for image classification demonstrates the potential of this approach to improving performance and reducing training costs further afield.

In their analysis of energy-efficient ML, Walsh et al. (2021) argue that the ability of transfer learning to employ a single base network over a wide range of target domains drastically saves on the energy consumption associated with these target networks. This is because the cost of training the base network is shared amongst all new networks that use it, and each retraining round only requires a fraction of the resources required to train the models from scratch. This view is shared by Schwartz et al. (2020), who advocate for more base networks to be released

publicly to promote transfer learning, and Xu et al. (2021) who similarly assert that parameter initialisation from pre-trained models both reduces training time and improves the ability of a model to generalise well to unseen data. However, some questions have been raised about the hidden inefficiencies of transfer learning; Strubell et al. (2019) argue that unless the base and target tasks are directly synonymous, repurposing learned features can become extremely computationally demanding, requiring a large number of parameter evaluations and updates that have their own energy cost. Despite this, transfer learning remains a promising path toward more energy-efficient neural networks, and the increased popularity of research within this domain offers a great opportunity to reduce the impacts of ML.

2.4.6 Hyperparameter Optimisation

The optimisation of hyperparameters is another aspect of the training process that bears a significant energy cost; both Schwartz et al. (2020) and Lacoste et al. (2019) cite the number of hyperparameters, and the time spent optimising them, as a major contributing factor toward the cost of Red AI. The search for an optimal neural network architecture during HPO, known as *neural architecture search* (NAS), constitutes the bulk of this process. Basic implementations of NAS, such as finding a local optimum value for the hyperparameters through randomly minimising a given error metric, typically require many computationally expensive operations (as the error space is non-convex and so has multiple minima) and thus have a significant cost in time, energy, and carbon emissions. However, Xu et al. (2021) explain that this inefficiency can be tackled by decomposing the NAS process into its three constituent parts: the architecture search space, the search strategy for this space, and the evaluation method of architecture instances selected. Efficiency-focussed research addressing these three stages has made significant progress in reducing the energy expended during NAS compared to the basic approach.

Firstly, efficiency can be improved by reducing the architecture search space, limiting the number of possible architectures that have to be surveyed. These approaches typically build a *structured search space*, exploiting some prior knowledge to manipulate the search space into a more structured form. One such method is constructing a *cell-based search space*, proposed by Zoph et al. (2017) and used by many models including Liu et al. (2018) and Real et al. (2019). This method recognises that there is significant repetition in the search space as large chunks of networks can be very similar across many different architectures; thus, architectures are broken down into *cells*, each containing a distinct part of a network architecture. Hence, the search space is reduced from formulations of whole architectures to searching different cells that can be composed to make a complete architecture. Zoph et al. (2017) assert that this smaller search space drastically speeds up NAS, claiming a 7 times speedup over basic searches over full architectures. Secondly, the search strategy through which we pick possible instances from the search space provides an opportunity for increased efficiency, as the faster the optimal candidate can be found the shorter the search time. Improving on the most basic random search method (where instances are drawn at random), several potential alternatives have been suggested. These range from *reinforcement learning based search*, where an architecture generator network constructs possible instances to be evaluated, to *evolution based search* where instances are selected and then mutated to optimise their architecture (a technique used by So et al. (2019) and Real et al. (2019)). Thirdly, architecture evaluation can be the most resource-hungry element of NAS, as candidate architectures are typically fully trained before performance is measured. To improve efficiency, network training adaptations such as weight sharing and early-exiting can be exploited, reducing the time and energy consumed tuning a network's parameters before the architecture's worth can be judged.

Despite these approaches to improving the energy efficiency of HPO showing much poten-

tial, currently, efficient NAS has only widely been adopted in CV applications (to reduce the computational load of intensive CNNs in works such as Real et al. (2019)). Hence, whilst some interest in this field has been generated by other works (e.g. So et al.'s NLP model), more work is required to increase the adoption of efficient architecture search.

2.4.7 Reducing Precision and Accuracy

Most approaches to improving the energy efficiency of ML training aim to limit the impact on performance as much as possible, striking a compromise between efficiency and performance that significantly favours the latter. However, in many applications the upper limits of performance are neither necessary nor beneficial, meaning this compromise can be shifted to produce even greater cost reductions. One such application is edge devices, where low power and memory usage must be prioritised. Both Kumar et al. (2020) and Xu et al. (2021) suggest that reducing the accuracy of operations can be incredibly beneficial in this case; Sze et al. (2017) similarly note that DL models using lower precision data representations can drastically decrease computation and memory costs. This reduction in accuracy and precision is commonly implemented through *quantisation*, where continuous values are mapped to discrete quantisation levels. This compresses the number of bits used to store variables, reducing the memory requirements of a network, the energy used fetching variables from both internal memory and cloud data storage, and the time and energy expended performing calculations.

Quantisation can take two forms: *deterministic quantisation*, computing an explicit map between higher and lower resolution representations, and *stochastic quantisation*, where no such deterministic mapping exists. There are several varying implementations of deterministic quantisation, which range from *uniform methods* that perform a uniform conversion between floating-point values and fixed-point values (such as Zhou et al.'s reduction from 32-bit to 8-bit representations), to *clustering methods* that group together similar values into clusters and replace each with the mean value of their group (e.g. Gong et al., 2014). Alternatively, *random rounding* (used by Courbariaux et al.'s image classifier BinaryConnect) is an example of stochastic quantification, where quantised values are sampled from a discrete probability distribution. In all of these implementations, the performance-efficiency tradeoff can be manipulated to provide varying compromises between the number of bits used to store each variable and the accuracy of the model. For example, Judd et al. (2016) focus on reducing bit count whilst only inflicting an accuracy loss of 1%, whilst BinaryConnect only uses single bit representations but inflicts a 19% accuracy drop. Furthermore, quantisation-aware training can provide a promising alternative to simply reducing the bit representations of pre-trained models; this method is used by Fan, Stock, Graham, Grave, Gribonval, Jegou and Joulin (2020) to produce a model that uses only 3.3MB of memory to achieve an *ImageNet top-1 accuracy* score of 80.0. Hence, quantisation offers much scope to reduce memory and energy costs in circumstances where state-of-the-art accuracy is not essential (which, in reality, is most commercial applications of ML).

2.4.8 Data Efficiency

Al-Jarrah et al. (2015) highlight that the efficient handling of data lies at the heart of developing energy-efficient ML models. This is an area often neglected by models aiming to improve energy efficiency, as while reducing parameter counts they often still rely on large training datasets. For example, *ALBERT* (Lan et al., 2019) attempts to recreate the performance of BERT using a smaller parameter set, however, does so by exploiting a large dataset, resulting in the model retaining a high computational load and energy consumption.

Xu et al. (2021) raise *active learning* as one possible resolution to this problem. Active learning restricts the number of data instances exploited during training to only those that

provide the most utility to the learning process. Since not all training instances will be equally useful to learning the parameters of a mode, if the training dataset can be constrained to only those data points that benefit the model's performance, the time and energy wasted on non-beneficial updates can be minimised, improving the energy-efficiency of training. There are several different ways in which active learning has been implemented; these methods can be categorised through the *importance function* used to select between training instances. For example, *uncertainty based active learning* selects instances through an uncertainty metric, such as the entropy of a distribution of data points (Joshi et al., 2009). Alternatively, *expected-model-change based active learning* chooses training instances expected to generate the greatest change of value in the model's parameters (Freytag et al., 2014). Besides active learning, some research has also focussed on leveraging pre-trained models to facilitate the use of small datasets—drawing parallels to the utilisation of transfer learning by Wang et al. (2020) to perform classification over a small dataset. Popular approaches include *self-supervised learning* (Jing and Tian, 2019), *contrastive learning* (Chen et al., 2020) , and *prompt learning* (Liu et al., 2021), all of which attempt to leverage their datasets in a more informed, efficient manner. Whilst the benefits of these approaches have been demonstrated in both NLP and CV, widespread adoption by the most data-hungry, inefficient systems (such as those used by Google and Meta) has not yet been seen, suggesting the benefits of this data-efficient training require further exploration and promotion outside of these applications.

2.4.9 Future Directions

This work shows there is significant potential for improving the energy efficiency of ML through the development of resource-conscious network architectures and algorithms that minimise the number and computational load of operations intrinsic to ML systems. These systems reduce both energy consumption and carbon emissions, providing more sustainable methods for reducing the environmental, financial, and social impact of research. However, work in this field has largely remained on the fringes of ML research. Whilst the benefits to CV and NLP systems have been shown—such as the training speedup of Yang et al.'s language model and reduced memory requirements of Iandola et al.'s SqueezeNet—these methods typically are only implemented in mobile and edge devices where the importance of efficiency outweighs that of performance (Kumar et al., 2020). Therefore, it will take an increased effort to facilitate the use of such energy-efficient software in the state-of-the-art performant models that currently constitute the majority of the ML's carbon footprint.

3 Future Work

The work explored in this review substantiates the view that the field of energy-efficient ML offers much promise to minimise the cost of research (and commercial applications) within this field. The importance of increased adoption of these ideas is clear; hence, when designing a large-scale ML system one should recall the advice of Walsh et al. (2021), who advocate for the incorporation of energy efficiency within the design process of any ML model, and use software engineering methodologies such as a project pre-mortum (Klein, 2008) to evaluate and mitigate the impacts of such work through energy-conscious hardware and software solutions. Several commercial applications of ML, such as mobile and edge computing, already incorporate evaluations of resource budgets and energy efficiency into the design process. This is also true of a limited number of further ML applications, such as So et al. (2019) in the field of NLP, and Iandola et al.'s image classifier. However, whilst concern has been raised by multiple developers in these fields, energy-efficient practices have not yet seen widespread adoption.

One such application of ML that has received little attention from the Green AI community is the field of *financial modelling*. In recent years, ML has been increasingly deployed by financial service institutions, building financial models for applications like algorithmic trading, asset management, and financial monitoring. Many of these models leverage DNNs to tackle the large datasets common in finance, such as transaction databases that typically take up petabytes of storage (OECD, 2021). For example, neural networks have been employed to perform risk prediction over financial statements (Green and Choi, 1997), and model the volatility of financial market prices (Cao et al., 2019). Furthermore, the industry surrounding financial modelling has exhibited tremendous growth in recent years, with the world's largest management consulting firm *McKinsey and Co.* estimating that employing ML could deliver up to $1 trillion of additional value per annum for the financial services industry (Kumar et al., 2022). However, the opportunity presented by leveraging DNNs for financial modelling comes alongside a plethora of risks, such as bias and interpretability difficulties (LaPlante and Rubtsov, 2019). The large datasets being exploited and rapid expansion into more complex modelling applications within the financial industry are also likely to come alongside a significant computational cost. Hence, the financial, environmental, and social consequences of large-scale ML highlighted in this review are likely to be just as prevalent (and detrimental) in this domain.

Recent attention to the environmental and social impacts of the financial industry has led to the emergence of *sustainable finance*, which aims to promote the incorporation of environmental, social, and governance factors (ESGs) into financial and investment decisions. This field encompasses research directions such as *climate financing*, *carbon financing*, and *impact investing* (Kumar et al., 2022). Machine learning has also been proved highly beneficial in sustainable finance: for example, automating the processing and analysis of public ESG data to provide better insights into where carbon emissions can be reduced by clients, partners, and suppliers (Allen et al., 2021). Sustainable finance has lately received a lot of industry attention, with the investment manager *BlackRock* pledging net-zero emissions by 2050 (Bodnar et al., 2022) and accounting firm *PwC* promising to hire 100,000 new employees by 2026 to tackle ESG related issues (DiNapoli and Lewis, 2021); however, there are still many major issues surrounding this field. These issues have been highlighted by Cunha et al. (2021), who describe the current state of sustainable finance as "excessively fragmented", and Fatemi and Fooladi (2013), who assert that much work in this field favours short-term solutions that simply outsource their social and environmental costs. Furthermore, the utilisation of ML in sustainable finance typically involves the application of existing expensive techniques, such as the application of NLP models for data analysis (e.g. *ESG-BERT* by Mehra et al. (2022)). This inevitably results in the same pitfalls as typical applications of Red AI, such as expensive computational loads and a cost in terms of energy and carbon emissions, in part counteracting any environmental benefit provided by these systems. Hence, there is an increased need for *sustainable ML for sustainable finance*, where the Green AI and energy-efficient approaches discussed in this review are leveraged to help improve the sustainability of financial modelling. For this reason, *energy-efficient machine learning for finance* will be a personal focus of future work within this field.

4 Conclusions

This review has demonstrated that the current trend of large architectures and computationally expensive datasets has led to state-of-the-art work in machine learning having a concerningly high environmental, financial, and social impact. However, it has been shown that newly proposed alternatives within the field of Green AI and energy-efficient ML promise to reduce this impact, limiting the energy and carbon cost of such systems. These adapted methods provide both efficient and performant implementations of common hardware and software used

in ML research. Therefore, to improve the energy efficiency of ML and reduce its negative impacts, such implementations should be considered during the design process of any ML system. Personally, the avenues explored by this review will be leveraged in future work to tackle the current environmental issues associated with the use of ML for sustainable finance, attempting to advance the field of *energy-efficient machine learning for finance*.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y. and Zheng, X. (2016). TensorFlow: A system for large-scale machine learning, *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283.
URL: *https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf*

Acharyya, P., Rosario, S. D., Flor, R., Joshi, R., Li, D., Linares, R. and Zhang, H. (2019). Autopilot of cement plants for reduction of fuel consumption and emissions, *ICML 2019 Workshop on Climate Change: How Can AI Help?*
URL: *https://www.climatechange.ai/papers/icml2019/26*

Akhand, M. A. H., Islam, M. M. and Murase, K. (2008). Training of neural network ensemble through progressive interaction, *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 2120–2126.

Al-Jarrah, O. Y., Yoo, P. D., Muhaidat, S., Karagiannidis, G. K. and Taha, K. (2015). Efficient machine learning for big data: A review, *Big Data Research* **2**(3): 87–93. Big Data, Analytics, and High-Performance Computing.
URL: *https://www.sciencedirect.com/science/article/pii/S2214579615000271*

Allen, T., Aubertel, E., Basirov, A., Craig, M., Delvoye, V., Hammond, L., Hecquet, J.-P., Spencer, K. and Paris, A. S. (2021). The Path to ESG: No Turning Back For Asset Owners and Managers, *The ESG Global Survey*, BNP Paribas.

Amodei, D. and Hernandez, D. (2021). Ai and compute.
URL: *https://openai.com/blog/ai-and-compute/*

Belilovsky, E., Eickenberg, M. and Oyallon, E. (2018). Greedy Layerwise Learning Can Scale to ImageNet, *arXiv e-prints* p. arXiv:1812.11446.

Bender, E. M., Gebru, T., McMillan-Major, A. and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big?, *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, Association for Computing Machinery, New York, NY, USA, p. 610–623.
URL: *https://doi.org/10.1145/3442188.3445922*

Bietti, E. and Vatanparast, R. (2019). Data Waste, *Harvard International Law Journal* **61**.
URL: *https://ssrn.com/abstract=3584251*

Bizo, D., Ascierto, R., Lawrence, A. and Davis, J. (2021). Uptime Institute Global Data Center Survey 2021, *Technical report*, Uptime Institute Intelligence.
URL: *https://uptimeinstitute.com/2021-data-center-industry-survey-results*

Bodnar, P., Boivin, J., Brazier, A., Gill, N., Paul, V. and Weber, C. (2022). Managing the net-zero transition, BlackRock Investment Institute.
URL: *https://www.blackrock.com/corporate/insights/blackrock-investment-institute/publications/net-zero-transition*

Buchanan, W. and Tachet des Combes, R. (2021). Charting the path towards sustainable AI with Azure Machine Learning Resource Metrics, *Microsoft Azure Machine Learning* .
URL: *https://techcommunity.microsoft.com/t5/green-tech-blog/charting-the-path-towards-sustainable-ai-with-azure-machine/ba-p/2866923*

Burr, G. W., Sebastian, A., Ando, T. and Haensch, W. (2021). Ohm's Law + Kirchhoff's Current Law = Better AI: Neural-Network Processing Done in Memory with Analog Circuits will Save Energy, *IEEE Spectrum* **58**(12): 44–49.

Cao, J., Chen, J. and Hull, J. (2019). A neural network approach to understanding implied volatility movements, *ERN: Volatility (Topic)* .

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., St. John, R., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y.-H., Strope, B. and Kurzweil, R. (2018). Universal Sentence Encoder, *arXiv e-prints* p. arXiv:1803.11175.

Chen, T., Kornblith, S., Norouzi, M. and Hinton, G. (2020). A Simple Framework for Contrastive Learning of Visual Representations, *arXiv e-prints* p. arXiv:2002.05709.

Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., Xiao, T., Xu, B., Zhang, C. and Zhang, Z. (2015). MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems, *arXiv e-prints* p. arXiv:1512.01274.

Cheng, H., Tan, P.-N. and Jin, R. (2010). Efficient algorithm for localized support vector machine, *IEEE Transactions on Knowledge and Data Engineering* **22**(4): 537–549.

Chirodea, M. C., Novac, O. C., Novac, C. M., Bizon, N., Oproescu, M. and Gordan, C. E. (2021). Comparison of tensorflow and pytorch in convolutional neural network - based applications, *2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1–6.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L. and Bordes, A. (2017). Supervised Learning of Universal Sentence Representations from Natural Language Inference Data, *arXiv e-prints* p. arXiv:1705.02364.

Cook, G., Lee, J., Tsai, T., Kongn, A., Deans, J., Johnson, B., Jardim, E. and Johnson, B. (2017). Clicking Clean: Who is winning the race to build a green internet?, *Technical report*, Greenpeace.

Courbariaux, M., Bengio, Y. and David, J.-P. (2015). BinaryConnect: Training Deep Neural Networks with binary weights during propagations, *arXiv e-prints* p. arXiv:1511.00363.

Cunha, F. A. F. d. S., Meira, E. and Orsato, R. J. (2021). Sustainable finance and investment: Review and research agenda, *Business Strategy and the Environment* **30**(8): 3821–3838.
URL: *https://onlinelibrary.wiley.com/doi/abs/10.1002/bse.2842*

Daniel, C., Shukla, A. K. and Sharma, M. (2021). Applications of machine learning in harnessing of renewable energy, *in* P. V. Baredar, S. Tangellapalli and C. S. Solanki (eds), *Advances in Clean Energy Technologies*, Springer Singapore, Singapore, pp. 177–187.

David, R., Duke, J., Jain, A., Janapa Reddi, V., Jeffries, N., Li, J., Kreeger, N., Nappier, I., Natraj, M., Regev, S., Rhodes, R., Wang, T. and Warden, P. (2020). TensorFlow Lite Micro: Embedded Machine Learning on TinyML Systems, *arXiv e-prints* p. arXiv:2010.08678.

de Leeuw, O. (2021). Towards More Energy-Efficient Neural Networks, *Open Data Science Conference West 2021*.
URL: *https://odsc.com/blog/towards-more-energy-efficient-machine-learning-models/*

Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J. and Kaiser, Ł. (2018). Universal Transformers, *arXiv e-prints* p. arXiv:1807.03819.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding, *ArXiv* **abs/1810.04805**.

DiNapoli, J. and Lewis, M. (2021). Pwc planning to hire 100,000 over five years in major esg push.
URL: *https://www.reuters.com/business/sustainable-business/pwc-planning-hire-100000-over-five-years-major-esg-push-2021-06-15/*

Dodge, J., Gururangan, S., Card, D., Schwartz, R. and Smith, N. A. (2019). Show Your Work: Improved Reporting of Experimental Results, *arXiv e-prints* p. arXiv:1909.03004.

Fan, A., Bhosale, S., Schwenk, H., Ma, Z., El-Kishky, A., Goyal, S., Baines, M., Celebi, O., Wenzek, G., Chaudhary, V., Goyal, N., Birch, T., Liptchinsky, V., Edunov, S., Grave, E., Auli, M. and Joulin, A. (2020). Beyond english-centric multilingual machine translation.

Fan, A., Stock, P., Graham, B., Grave, E., Gribonval, R., Jegou, H. and Joulin, A. (2020). Training with Quantization Noise for Extreme Model Compression, *arXiv e-prints* p. arXiv:2004.07320.

Fatemi, A. M. and Fooladi, I. J. (2013). Sustainable finance: A new paradigm, *Global Finance Journal* **24**(2): 101–113.
URL: *https://www.sciencedirect.com/science/article/pii/S1044028313000252*

Fedus, W., Zoph, B. and Shazeer, N. (2021). Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity, *arXiv e-prints* p. arXiv:2101.03961.

Freytag, A., Rodner, E. and Denzler, J. (2014). Selecting influential examples: Active learning with expected model output changes, *European Conference on Computer Vision*, pp. 562–577.

Google Sustainability (2021). Google Environmental Report 2021, *Technical report*, Google LLC.
URL: *https://www.gstatic.com/gumdrop/sustainability/google-2021-environmental-report.pdf*

Green, B. and Choi, J. (1997). Assessing the risk of management fraud through neural network technology, *Auditing* **16**.

Han, S., Pool, J., Tran, J. and Dally, W. J. (2015). Learning both Weights and Connections for Efficient Neural Networks, *arXiv e-prints* p. arXiv:1506.02626.

Hanson, S. and Pratt, L. (1988). Comparing Biases for Minimal Network Construction with Back-Propagation, *in* D. Touretzky (ed.), *Advances in Neural Information Processing Systems*, Vol. 1, Morgan-Kaufmann.

URL: *https://proceedings.neurips.cc/paper/1988/file/1c9ac0159c94d8d0cbedc973445af2da-Paper.pdf*

He, K., Zhang, X., Ren, S. and Sun, J. (2015). Deep Residual Learning for Image Recognition, *arXiv e-prints* p. arXiv:1512.03385.

Henderson, P., Hu, J., Romoff, J., Brunskill, E., Jurafsky, D. and Pineau, J. (2020). Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning, *arXiv e-prints* p. arXiv:2002.05651.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J. and Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and ¡0.5MB model size, *arXiv e-prints* p. arXiv:1602.07360.

Jain, P., Mo, X., Jain, A., Tumanov, A., Gonzalez, J. E. and Stoica, I. (2019). The OoO VLIW JIT Compiler for GPU Inference, *ArXiv* **abs/1901.10008**.

Jing, L. and Tian, Y. (2019). Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey, *arXiv e-prints* p. arXiv:1902.06162.

Joshi, A. J., Porikli, F. and Papanikolopoulos, N. (2009). Multi-class active learning for image classification, *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2372–2379.

Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., Boyle, R., Cantin, P.-l., Chao, C., Clark, C., Coriell, J., Daley, M., Dau, M., Dean, J., Gelb, B., Vazir Ghaemmaghami, T., Gottipati, R., Gulland, W., Hagmann, R., Ho, C. R., Hogberg, D., Hu, J., Hundt, R., Hurt, D., Ibarz, J., Jaffey, A., Jaworski, A., Kaplan, A., Khaitan, H., Koch, A., Kumar, N., Lacy, S., Laudon, J., Law, J., Le, D., Leary, C., Liu, Z., Lucke, K., Lundin, A., MacKean, G., Maggiore, A., Mahony, M., Miller, K., Nagarajan, R., Narayanaswami, R., Ni, R., Nix, K., Norrie, T., Omernick, M., Penukonda, N., Phelps, A., Ross, J., Ross, M., Salek, A., Samadiani, E., Severn, C., Sizikov, G., Snelham, M., Souter, J., Steinberg, D., Swing, A., Tan, M., Thorson, G., Tian, B., Toma, H., Tuttle, E., Vasudevan, V., Walter, R., Wang, W., Wilcox, E. and Yoon, D. H. (2017). In-Datacenter Performance Analysis of a Tensor Processing Unit, *arXiv e-prints* p. arXiv:1704.04760.

Judd, P., Albericio, J., Hetherington, T., Aamodt, T. M. and Moshovos, A. (2016). Stripes: Bit-serial deep neural network computing, *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 1–12.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P. and Hassabis, D. (2021). Highly accurate protein structure prediction with alphafold, *Nature* **596**(7873): 583–589.
URL: *https://doi.org/10.1038/s41586-021-03819-2*

Klein, G. (2008). Performing a project premortem, *Engineering Management Review, IEEE* **36**: 103–104.

Kumar, M., Zhang, X., Liu, L., Wang, Y. and Shi, W. (2020). Energy-Efficient Machine Learning on the Edges, *2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 912–921.

Kumar, S., Sharma, D., Rao, S., Lim, W. M. and Mangla, S. K. (2022). Past, present, and future of sustainable finance: insights from big data analytics through machine learning of scholarly research, *Annals of Operations Research* .
URL: *https://doi.org/10.1007/s10479-021-04410-8*

Lacoste, A., Luccioni, A., Schmidt, V. and Dandres, T. (2019). Quantifying the Carbon Emissions of Machine Learning, *arXiv e-prints* p. arXiv:1910.09700.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P. and Soricut, R. (2019). ALBERT: A Lite BERT for Self-supervised Learning of Language Representations, *arXiv e-prints* p. arXiv:1909.11942.

LaPlante, A. and Rubtsov, A. (2019). Artificial Neural Networks in Financial Modelling, Global Risk Institute.
URL: *https://globalriskinstitute.org/publications/artificial-neural-networks-in-financial-modelling/*

Liu, H., Simonyan, K. and Yang, Y. (2018). DARTS: Differentiable Architecture Search, *arXiv e-prints* p. arXiv:1806.09055.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H. and Neubig, G. (2021). Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing, *arXiv e-prints* p. arXiv:2107.13586.

Lottick, K., Susai, S., Friedler, S. A. and Wilson, J. P. (2019). Energy Usage Reports: Environmental awareness as part of algorithmic accountability, *arXiv e-prints* p. arXiv:1911.08354.

Luo, J.-H., Zhang, H., Zhou, H.-Y., Xie, C.-W., Wu, J. and Lin, W. (2019). ThiNet: Pruning CNN Filters for a Thinner Net, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **41**(10): 2525–2538.

Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A. and van der Maaten, L. (2018). Exploring the Limits of Weakly Supervised Pretraining, *arXiv e-prints* p. arXiv:1805.00932.

Marwah, M., Shah, A., Bash, C., Patel, C. and Ramakrishnan, N. (2011). Using data mining to help design sustainable products, *IEEE Computer* **44**: 103–106.

McDonald, R., Hall, K. and Mann, G. (2010). Distributed training strategies for the structured perceptron, *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, Association for Computational Linguistics, USA, pp. 456–464.

Mehra, S., Louka, R. and Zhang, Y. (2022). ESGBERT: Language Model to Help with Classification Tasks Related to Companies Environmental, Social, and Governance Practices, *arXiv e-prints* p. arXiv:2203.16788.

Mehta, S., Rastegari, M., Shapiro, L. and Hajishirzi, H. (2018). ESPNetv2: A Light-weight, Power Efficient, and General Purpose Convolutional Neural Network, *arXiv e-prints* p. arXiv:1811.11431.

Microsoft News Center (2017). Microsoft announces one of the largest wind deals in the Netherlands with Vattenfall, *Microsoft Media Relations, WE Communications for Microsoft, (425) 638-7777* .
URL: *https://news.microsoft.com/2017/11/02/microsoft-announces-one-of-the-largest-wind-deals-in-the-netherlands-with-vattenfall/*

Morin, F. and Bengio, Y. (2005). Hierarchical Probabilistic Neural Network Language Model, *in* R. G. Cowell and Z. Ghahramani (eds), *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, Vol. R5 of *Proceedings of Machine Learning Research*, PMLR, pp. 246–252. Reissued by PMLR on 30 March 2021.
URL: *https://proceedings.mlr.press/r5/morin05a.html*

OECD (2021). Artificial Intelligence, Machine Learning and Big Data in Finance: Opportunities, Challenges, and Implications for Policy Makers, *Technical report*, The Organisation for Economic Co-operation and Development.
URL: *https://www.oecd.org/finance/artificial-intelligence-machine-learning-big-data-in-finance.htm*

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. and Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library, *arXiv e-prints* p. arXiv:1912.01703.

Plummer, B. A., Dryden, N., Frost, J., Hoefler, T. and Saenko, K. (2020). Neural Parameter Allocation Search, *arXiv e-prints* p. arXiv:2006.10598.

Real, E., Aggarwal, A., Huang, Y. and Le, Q. V. (2019). Aging evolution for image classifier architecture search, *AAAI 2019*.

Redmon, J. and Farhadi, A. (2016). YOLO9000: Better, Faster, Stronger, *arXiv e-prints* p. arXiv:1612.08242.

Ruder, S. (2018). NLP's ImageNet moment has arrived, *The Gradient* .

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers, *IBM J. Res. Dev.* **3**(3): 210–229.
URL: *https://doi.org/10.1147/rd.33.0210*

Schwartz, R., Dodge, J., Smith, N. and Etzioni, O. (2020). Green ai, *Communications of the ACM* **63**: 54 – 63.

Sennrich, R., Haddow, B. and Birch, A. (2015). Neural Machine Translation of Rare Words with Subword Units, *arXiv e-prints* p. arXiv:1508.07909.

Seznec, M., Gac, N., Orieux, F. and Sashala Naik, A. (2021). Real-Time Optical Flow Processing on Embedded GPU: anHardware-Aware Algorithm to Implementation Strategy, *Journal of Real-Time Image Processing* .
URL: *https://hal.archives-ouvertes.fr/hal-03457011*

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T. P., Hui, F., Sifre, L., van den Driessche, G., Graepel, T. and Hassabis, D. (2017). Mastering the game of go without human knowledge, *Nature* **550**: 354–359.

Simonyan, K. and Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition, *arXiv e-prints* p. arXiv:1409.1556.

So, D. R., Liang, C. and Le, Q. V. (2019). The Evolved Transformer, *arXiv e-prints* p. arXiv:1901.11117.

Strubell, E., Ganesh, A. and McCallum, A. (2019). Energy and Policy Considerations for Deep Learning in NLP, *arXiv e-prints* p. arXiv:1906.02243.

Sze, V., Chen, Y.-H., Yang, T.-J. and Emer, J. (2017). Efficient Processing of Deep Neural Networks: A Tutorial and Survey, *arXiv e-prints* p. arXiv:1703.09039.

Teerapittayanon, S., McDanel, B. and Kung, H. T. (2017). BranchyNet: Fast Inference via Early Exiting from Deep Neural Networks, *arXiv e-prints* p. arXiv:1709.01686.

United Nations Department of Economic and Social Affairs (2016). *World Economic and Social Survey 2016: Climate Change Resilience - An Opportunity for Reducing Inequalities*, World Economic and Social Survey (WESS), UN.
URL: *https://books.google.co.uk/books?id=JlvjDwAAQBAJ*

Veniat, T. and Denoyer, L. (2017). Learning Time/Memory-Efficient Deep Architectures with Budgeted Super Networks, *arXiv e-prints* p. arXiv:1706.00046.

Walsh, P., Bera, J., Sharma, V. S., Kaulgud, V., Rao, R. M. and Ross, O. (2021). Sustainable ai in the cloud: Exploring machine learning energy use in the cloud, *2021 36th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW)*, pp. 265–266.

Wang, K., Gao, X., Zhao, Y., Li, X., Dou, D. and Xu, C.-Z. (2020). Pay attention to features, transfer learn faster cnns, *International Conference on Learning Representations*.
URL: *https://openreview.net/forum?id=ryxyCeHtPB*

Wang, T. and Selvan, A. (2021). Google wins MLPerf benchmarks with TPU V4, *Google Cloud Blog* .
URL: *https://cloud.google.com/blog/products/ai-machine-learning/google-wins-mlperf-benchmarks-with-tpu-v4*

Wang, X., Yu, F., Dou, Z.-Y., Darrell, T. and Gonzalez, J. E. (2017). SkipNet: Learning Dynamic Routing in Convolutional Networks, *arXiv e-prints* p. arXiv:1711.09485.

Xie, S., Girshick, R., Dollár, P., Tu, Z. and He, K. (2016). Aggregated Residual Transformations for Deep Neural Networks, *arXiv e-prints* p. arXiv:1611.05431.

Xu, J., Zhou, W., Fu, Z., Zhou, H. and Li, L. (2021). A Survey on Green Deep Learning, *ArXiv* **abs/2111.05193**.

Yang, C., Wang, S., Yang, C., Li, Y., He, R. and Zhang, J. (2020). Progressively Stacking 2.0: A Multi-stage Layerwise Training Method for BERT Training Speedup, *arXiv e-prints* p. arXiv:2011.13635.

Yoo, P. D., Ng, J. W. and Zomaya, A. Y. (2011). An energy-efficient kernel framework for large-scale data modeling and classification, *2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum*, pp. 404–408.

Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H. and Zou, Y. (2016). DoReFa-Net: Training Low Bitwidth Convolutional Neural Networks with Low Bitwidth Gradients, *arXiv e-prints* p. arXiv:1606.06160.

Zoph, B., Vasudevan, V., Shlens, J. and Le, Q. V. (2017). Learning Transferable Architectures for Scalable Image Recognition, *arXiv e-prints* p. arXiv:1707.07012.