Development of Energy Efficient Machine Learning Models – A Survey

Yuhang Song
University of Liverpool
L69 3BX
sgyson10@liverpool.ac.uk

Abstract

Rapid evolving of the Artificial Intelligence mechanisms brings significant challenges to the global energy consumptions, due to the better performance of huge models like Deep Neural Networks and complex Reinforcement Learning Control model, standard model designs are becoming more expensive to execute, thus brought impacts on both the energy consumptions and the availability of those techniques. In this paper, contributions are made to summarize the DNN energy consumption problem as a whole and analyze the current available solutions and their limitations. Firstly, I have analyzed some mechanisms that reduce the complexity of DNN models, then further developed that the reduction of calculations and parameters is not necessarily proportional to the energy consumption. I then research though the source of energy consumption as well as the amount and find out is more dependent on the lower-level computing configurations. Lastly, some analyzes are performed on current energy-efficient solutions, together with my conclusions and research target in this field of study.

1. Introduction

Since the DNN and Reinforcement Learning are now widely used in many fields of research and commercial purpose, the development of energy-friendly models will not just helping address the global energy issues, but most importantly radiating its lower cost throughout all fields that uses Deep Neural Networks as major component, energy-friendly models would reduce research and commercial costs and thus promoting the research more rapid in fields like Computer Vision, NPL, Reinforcement Learning, and general DNN models development.

Most huge but effective models proposed recently, like GPT-3, are widely used in offline training, yet still difficult to be applied in online training due to the high inference complexity and training complexity of such models. As a commonly known trend in the filed of machine learning, online training brings more flexibilities comparing to offline training models. The development of the Energy-Efficient models would potentially help to make big models eligible to be used in online-training. Moreover, the studies of energy consumptions on Convolutional Neural Networks might help us understand better the feature extraction abilities of DNN architectures.

1.1. Motivations { Energy Consumption of Complex Models

The classic model ResNet101 [45] used in Computer Visions contains roughly 100,000,000 parameters, whereas for Natural Language Processing networks, depends on different models, from BERT, Megatron, Turing NLG, to the most famous GPT3, the number of parameters varies from 340,000,000 all the way to incredibly 175,000,000,000 [75], not to mention the newest Switch Transformer Model with trillions of parameters. Training such big models brings undoubtedly enormous challenges to the global energy consumption, together with the trends of Block Chain Mining, it has becoming a significant problems that we need to concern about during the development of the computer science.

Despite the influences on global energy, the expensive energy requirements are becoming the major cause that makes such big, yet relatively accurate models that are still not realistic to access from lighter devices.

1.2. Motivations { Embedded Devices

As a result of continual reduction in cost of computing capability, many devices are accessible to the processing power, which once been considered uneconomical for those devices, therefore with such processing power, we intend to build autonomous systems upon the devices. These devices typically have less computational powers.

With Artificial Intelligence techniques, Computer Vision [56, 93], Natural Language Processing [90], Deep Neural Network, Reinforcement Learning, and so on, due to their high performances, becoming backbones of autonomous systems, Deep Neural Networks, as core of all above mentioned techniques, after AlexNet won the ILSVRC 2012 [54], are made deeper and more complex to increase the

accuracy.

However, although there are some works showing the testing phase of above methods can be light-weighted, those Artificial Intelligence processing still consumes significant amount of time, [102] computational power, and memory in training. Therefore, such methods would typically not be afforded by light devices.

1.3. Terminology

- **FLOPs(Floating Point Operation):** It is essentially a quantity to measure the computing resources required by a specific DNN model or algorithm, representing the computational needs for the forward propagation. It is obtained by summarizing the number of additions and multiplying in each layer, can be used to quantify the complexity of a certain model.

- **Inference Power:** The computational power and energy needed for forward propagation.

- **Energy Consumption:** Indicated by the computing complexity and many other hard-ware level operation measurements of models(or algorithms), generally means the amount of usage of electricity and memory resources during training and inferencing process.

2. Literature Reviews

Generally, Deep Neural Networks and Reinforcement Learning [66] has 2 major phases, Testing and Training, both of which are not energy efficient when performing large machine learning models, where typically requires significant amounts of time, computational power, and memory in both phases. The Testing phase, due to the large number of parameters in complex network architectures or Reinforcement Control Models, can still be expensive, earlier studies shows that by using knowledge distilling and Asymmetry Architecture, light-weight models can be achieved in both Deep Neural Network and Reinforcement Learning. [20, 96].

Nevertheless, the general Training energy consumption issues in these 2 fields remains not well-addressed and has always been the recent research focus. Intuitively, people tend to think that the energy consumptions could be related to the complexity of the model, that is, in terms of the number of parameters or the number of parameters. There are some proven effective solutions on reducing the complexity of models.

2.1. Reducing Model Parameters and/or Calculation Complexity

2.1.1 Composite Kernels

There are some preliminaries studies indicated some directions of addressing such problem by reducing the complexity of big models, one of them is to use Composite Kernels together with Structured Convolutional Layers [13], in which the mathematical approach is performed to reduce the number of multiplies in convolutional layers. First by defining a set of composite Bias

$$B = \{\beta_1, \beta_2, ..., \beta_M\}$$

which represents a set of binary tensors of the shape of original Convolution Kernel, where for each β, it is consisting of either 0 or 1, that is formally:

$$\beta_m \in \{0, 1\}$$

Each convolutional layer has a unique B which divides the kernel into partitions with the location of validate elements of a certain partition be represented as 1s. Then assign to each β_m a parameter α_m, indicating learnable parameter, then the original kernel can be treated as the product of those two parameters, as shown in figure.1 With that being said, the convolution operation will now turn to a simpler way:

$$\begin{aligned} X * W = X * \sum_{m=1}^{M} \alpha_m \beta_m &= \sum_{m=1}^{M} \alpha_m (X * \beta_m) \\ &= \sum_{m=1}^{M} \alpha_m sum(X \bullet \beta_m) \\ &= \sum_{m=1}^{M} \alpha_m E_m \end{aligned}$$

As indicated in above equations, number of multiplications reduces from CN2 to simply M, where the M is called the degree of freedom of the parameters, example shown in Figure.1 has 4 degrees of freedoms. Moreover, the paper also shows that by using such methods to perform convolutions, the number of additions can also be reduced into $\sum sum(\beta_m - 1)$.

Such operations are equivalent to a sum-pooling followed by a smaller convolution operation. The method effectively transfers the original complex convolution operations into simpler sum pooling and a smaller convolution and yet keeps the loss of accuracy at low. Worth to mention that this mechanism can be integrated into further effective convolution models since it does not change any traditional

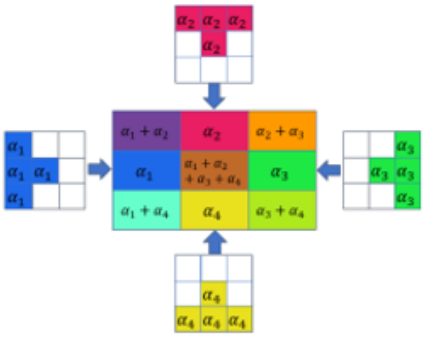

Figure 1. Colored region represents the partitions, with a associated with each partition.

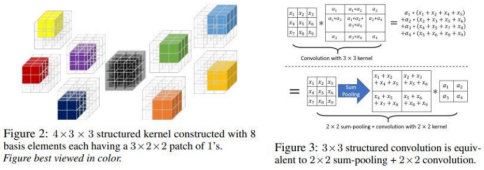

Figure 2: $4\times3\times3$ structured kernel constructed with 8 basis elements each having a $3\times2\times2$ patch of 1's. *Figure best viewed in color.*

Figure 3: 3×3 structured convolution is equivalent to 2×2 sum-pooling + 2×2 convolution.

Figure 2. The Structure Convolution from Qualcomm AI Research's paper.

convolution structure. Other alternative mathematical approaches are also available to reduce the number of parameters of large models during training for wide fields of computer visons or Natural Language Processing, as well as the Reinforcement Learning. [85, 86, 91]

2.1.2 Light Networks

Another previously proposed methods is called Light Networks, The core of lightweight networks is to lighten the network in terms of both size and speed while maintaining as much accuracy as possible, and this paper provides a brief description of lightweight networks, mainly involving SqueezeNet, ShuffleNet, ManasNet, MobileNet [2, 49] etc.

The most famous light network is MobileNet[6], proposed by Google Inc. in 2017, which is an architecture designed specifically for the lighter computing platforms like smartphones and general embedded devices, for them to use the high-performance Artificial Intelligence ability, both time complexity and space complexity along with the calculation complexity for the models are strictly limited.

To start with, Google proposed 2 new approaches to perform lighter convolution as alternative to normal convolution. Firstly, convolute the original input(of shape CxWxH) with a kernel of shape 1xNxN for each channel of original input separately and combining the outputs together, this operation is named the Depth-wise Convolution. As shown

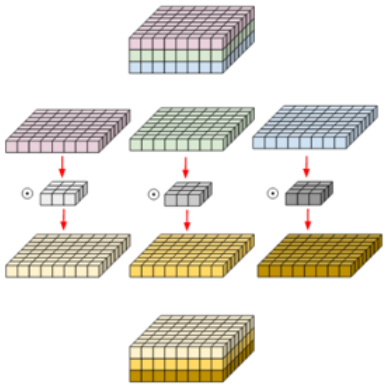

Figure 3. Depth-wise Convolution[13].

in Figure.3[13].Depth-wise Convolution are more efficient than the normal convolution. If we have an image with three channels. A regular convolution network applies the filter to the whole input while in depth-wise convolution, a single filter is applied to each input channel. The output of a standard convolution is a single layer while in the depth-wise convolution, the number of output layers equals the number of input channels.Then, to ensure the freedom of number of channels for the output for convolutional layers, since the result of Depth-wise Layer will be having exactly same number of channels as input, the paper then proposed that by making use of 1x1 filter, perform another convolution namely the Point-wise Convolution to recover this freedom, by defining number of 1x1 Point-wise kernels, we can obtain whatever numbers of channels we want in output. MobileNet is very popular since it was proposed, it reduces number of computations drastically. According to the original paper, this reduction is given by the following in comparison of the normal convolution:

$$\frac{1}{N} + \frac{1}{D_k^2}$$

It uses depth-wise separable convolutions that save the computation cost up to 8 to 9 times while the reduction in accuracy is minor according to the original paper. This network model is commonly used in ResNet and has been proven significantly effective in such a complex Deep Neural Networks. The detailed architecture of MobileNet V1 is given by the following:

MobileNet is later evolved into V3, with BottomNeck architectures involved and some AutoML mechanisms, removed the batch normalization operation and uses h-swish as activation function instead of RuLu6 in V1, further reduced the calculation complexity and number of parameters. Not like the previous Composite Kernels method mentioned in 2.1, the MobileNet changes the standard convo-

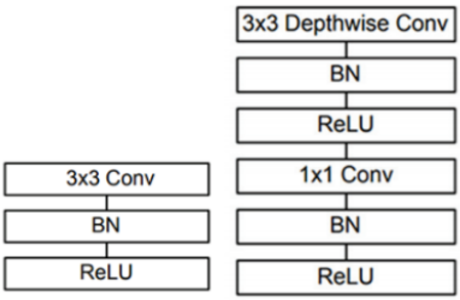

Figure 4. Left: Normal Convolution Blocks — Right: Standard MobileNet Blocks.

lutional operations by introducing the Depth-wise Convolution, that makes it less general to be deployed into real-life cases since it needs to be supported by the common AI programming frames for easier access. Moreover, since it proposes perform the convolution separately by channels, comparing to the normal convolution operation, it extracts less local information from the original input, hence causes the loss of accuracy.

2.1.3 Model Compression-Knowledge Distilling

Apart from actively convert the convolutional calculation progress into simpler additions or use different convolution structure, similar to the LfD mechanisms indicated in Reinforcement Learning, it is proposed that Knowledge Distilling [47] can also be used to reduce the Training time.

Knowledge Distilling was proposed firstly because of the inconsistency between training models and deploying models. During the training process we need to use complex models with large amounts of computational resources in order to extract information from very large, highly redundant datasets. In experiments, the models that work best tend to be very large in size, or even obtained by integrating multiple models. Whereas large models are not easy to deploy to services, there are common limitations like the slow inference speed for big models, or the limited resources in deploying environment. Therefore, the model compression problem becomes important, Knowledge Distilling was then proposed as one of the model compression mechanisms.

It defines two kinds of networks, one is called "Teacher Model", one is called "Student Model". Usually, there are two stages to perform in a general classification task:

> **1. Train the Teacher Model** , alternatively called Net-T, which is characterized by a relatively complex model and can be integrated from several separately trained models. The only requirement is that for each input X, the output Y, where Y is mapped by SoftMax, corresponds to the probability of the corresponding category.
>
> **2. Train the Student Model** , alternatively called Net-S, it is a single model with a small number of parameters and a relatively simple model structure. Similarly, for input X, it can all output Y. Y is mapped by SoftMax to output the same probability values corresponding to the corresponding categories.

With Knowledge Distilling, by compressing the model by Knowledge Distilling, the training for student models can be accelerated and it is still valid. The output of the SoftMax layer, in addition to the positive examples, also carries a lot of information about the negative labels, e.g., some negative labels correspond to a much higher probability than others. In contrast, in the traditional training process (hard target), all negative labels are treated uniformly. In other words, the KD training approach makes each sample bring more information to Net-S than the traditional training approach.

2.1.4 Network Pruning

Lastly, the acceleration in training can also be achieved via Network Pruning [67, 94], which indicates that some redundant are presented in the original Convolutional Neural Networks, by removing such unnecessary parameters, the parameters of the network are reduced and hence obtain a lighter model with more energy-friendly training progress. It is also shown that with proper pruning, the performance of the network increases rather than drops due to avoidance of overfitting.

2.2. Less Parameters Implies vs. Power Consumption

However, the with all above mentioned methods of reducing the complexity of big models, the main question here would be, does the number of calculations and/or parameters proportional to the energy consumption? Recent studies have investigated this question [21], energy consumptions as whole means the cost of electricity power, the number of parameters and the number of calculations along are not sufficient to quantify the energy consumption by a specific model, since it is commonly known that the energy costs in transferring data is more than calculations itself [48, 68, 84]. Some of the mechanisms mentioned above does reduced the energy consumption levels on certain computing platform, as will be discussed in section 3, but with very limited performance on that, it does, however, provide us some directions of study to reduce the energy consumption. There is still a long way to go to find a general solution to reduce the energy consumption of big models.

Model	Hardware	Power (W)	Hours	kWh·PUE	CO_2e	Cloud compute cost
$Transformer_{base}$	P100x8	1415.78	12	27	26	$41–$140
$Transformer_{big}$	P100x8	1515.43	84	201	192	$289–$981
ELMo	P100x3	517.66	336	275	262	$433–$1472
$BERT_{base}$	V100x64	12,041.51	79	1507	1438	$3751–$12,571
$BERT_{base}$	TPUv2x16	—	96	—	—	$2074–$6912
NAS	P100x8	1515.43	274,120	656,347	626,155	$942,973–$3,201,722
NAS	TPUv2x1	—	32,623	—	—	$44,055–$146,848
GPT-2	TPUv3x32	—	168	—	—	$12,902–$43,008

Figure 5. Estimated cost of training a model in terms of CO2 emissions (lbs) and cloud compute cost (USD). Power and carbon footprint are omitted for TPUs due to lack of public information on power draw for this hardware. [90] .

Consumer	Renew.	Gas	Coal	Nuc.
China	22%	3%	65%	4%
Germany	40%	7%	38%	13%
United States	17%	35%	27%	19%
Amazon-AWS	17%	24%	30%	26%
Google	56%	14%	15%	10%
Microsoft	32%	23%	31%	10%

Figure 6. Percent energy sourced from: Renewable (e.g. hydro, solar, wind), natural gas, coal and nuclear for the top 3 cloud compute providers (Cook et al., 2017) [7, 23], compared to the United States,4 China5 and Germany (Burger, 2019) [17]

2.3. How Much Energy is Used During DNN's Training? What is the impact caused?

According to the latest study [90] from the University of Massachusetts, Amherst, which they performed a life cycle assessment for large models on training process. [17, 19, 29] It is found that training such models typically requires averagely 3.4 kW electricity powers, and meanwhile it releases averagely 626,000 pounds of carbon dioxide, this amount of waste is equivalent to nearly five times the lifetime emissions of the average American car (and that includes manufacture of the car itself). The study also analyzed different energy consuming sources of different nationals from there top 3 cloud computing providers, as shown in Figure.6 below. Spot that while training a single model is relatively cheap, the cost of tuning a model for a new dataset, which we estimate here to require 24 jobs, or performing the full R&D required to develop this model, it then becomes extremely expensive. [33, 50, 59, 62, 65]

In conclusion, the study found that the energy consumption and environmental costs of training process grew proportionally to the size of model and dramatically increases then performing additional tuning process for accuracy improvement, due to the repeated executions of model in whole during tuning.

2.4. Where Does the Consumptions Comes From?

With all above mentioned methods of reducing the complexity of big models, we say that the complexity of the models does not necessarily relate to the energy consumption. Plus, the scary records of energy consuming data, the main question that needs to be clarify is where exactly those energy consumptions comes from during the computing process. Let's first take a deep look from what is going on in the lower level.

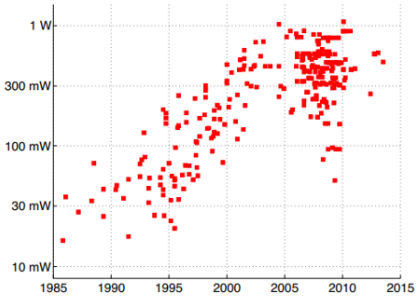

Figure 7. Power density in mW/mm2 vs. year [34]

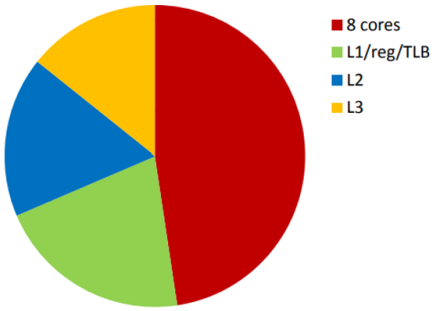

Figure 8. Power breakdown of an 8-core server chip. [25, 34]

According to a related study from Stanford University [34], as shown in figure.7, rapid development of computing processors is being made smaller [27], but the power density of the processors does not scale as the size of them according to plan. Higher performance of processers with increasingly smaller size brought large power density. Apart from the energy consumed in the processor, the study also spots the significant power consumptions from the memory [26, 64].

Figure.8 illustrates the power breakdown of a recent 40nm, 8-core superscalar processor with an 8MB last-level cache. Over 50% of the processor die energy is dissipated in the caches and register files in this machine. To a even clearer understanding of where does this energy comes from in terms of programming, the study breakdown the power consumption data of common assemble-language-level operations, shown in Figure.9.

The data from Figure.9 is recorded based on a 45nm technology, the data does show that additions comparing to multiplying costs less power, but calculations are not the only facts that determining the result, most importantly, the

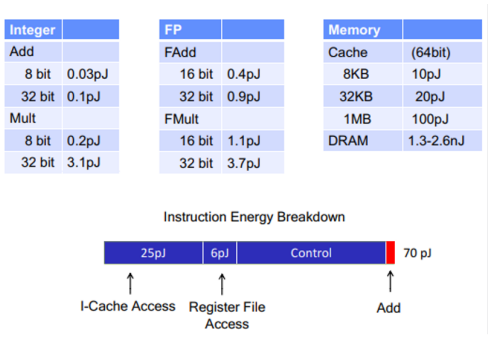

Figure 9. Rough energy costs for various operations in 45nm 0.9V. [34, 74]

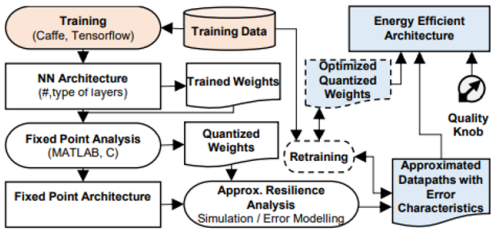

Figure 10. Energy-efficient and adaptive neural network accelerator-based architectures for Machine Learning and AI systems proposed by [80]

data suggests that the application must have very good data locality to maintain high energy efficiency, since a cache fetch is 20pJ. Fetches from the first level cache have an energy cost that is a significant fraction of that of an instruction, so if data needs to be fetched often, the energy improvement will be modest. [34] That tackles the present DNN model designs, most of the big models fetches parameters very often, by reconsider the design particularly in relation to the data locality would be a direction of study.

As computer scientists, in addition to learning about what is happening in lower level, the propriate measurement from higher level algorithms would also be critical. Present methods of evaluating the energy consumptions for Machine Learning algorithms based deeply on the computing platforms, as discussed above, lower-level energy consumptions studies are the main course when comes to talk about energy-efficiency, different computing platforms tend to have different lower-level architectures. Although it is now lack of a common method to directly measure the energy consumption of all computing platforms, some mechanisms are proposed to bridge the gap. [8, 35, 37–39] It is proposed that since the lack of explanation of traditional power meters places at different places, the method of using either simulated hardware or performance monitoring counters [36] can be use as alternative. Using the hardware simulator allows the researchers to have a detailed view of how each hardware component is accessed by the program, however the hardware simulator added overhead, which makes it not suitable for obtaining the real-time energy measurement. [11]

PMC(Performance Monitoring Counters) are now accessible in almost all modern processors; they provide the ability to count microarchitectural events of processor at run time. [35] Both hardware simulations and PMC are capable of measuring the power consumptions for computing programs at eighter architecture level or instruction level or both [35]. Since our purpose is focusing on the large models of machine learning training progress, typically with large scale datasets, the study suggests that using one of [9, 12, 26, 32, 36–38, 50, 53, 76, 82, 83, 87] modelling methods to measure the energy consumption.

In order to develop a energy-efficient model, works still needs to be done for a general energy consumption measurement.

2.5. Energy-Efficient Adaptive Hardware Accelerators for Neural Networks

Proposed firstly on 2017 IEEE Computer Society Annual Symposium on VLSI [80]. It clarified that most of the DNN applications are error resilient, which can be attributed to following causes: Firstly, since all the vectors in a specific model are being deal identically, therefore there is a potential that some neurons are redundant and by simplifying or even removing them does not reduce the quality of processing but could indeed reduce power consumption. Secondly, this error resilient can also be the result of the DNNs training is an iterative process, which is stoppable when a good result is obtained, and by retraining the model can even mitigate the effects of the lack of accuracy. Detailed design is shown in Figure.10. Training block performs pre-training for a specific model, then the result is delivered into the Fixed-Point Analysis [55, 61], which selects the an suitable fixed point format for keeping the minimum number of bits for both integer and fraction.

Then the selected weights are processed into Error Resilience Analysis, in which those weight are being tested either via Monte-Carlo based simulations where a certain randomly selected training data is used to evaluate the suitability of each Datapath for approximation or the analytical modeling. [44] Then by using adaptive hardware [14, 22, 80, 97, 103] which adjust the configurations accordingly, based on the analyzed error characteristics and accuracy requirements, the quality knob making tradeoff between these two and decide a suitable hardware sets for specific selected data. The paper shows some experiments did for selecting the approximate arithmetic configuration. [40, 42, 51, 69] Works been done to show that the efficiency improvement while using approximate mod-

els. [24, 30, 41, 52]

The mechanism reduced the power consumptions using adaptive hardware accelerators and configure the settings accordingly depending on specific data. However, there are still some questions remains unsolved, for example how to decide which to concedes? Memory or Computation? For the current method, it might significantly reduce the calculation power consumption, what about memory power consumption? Furthermore, a propriate mechanism to decide how to perform tradeoff between the accuracy and energy consumption is still needed.

2.6. Other Readings Related

Note that this work & some mentioned papers is also indirectly referred from the following related works: [**?**, 1, 3–6, 9, 12, 15, 16, 18, 28, 31, 46, 53, 57, 58, 60, 63, 70–73, 77–79, 81, 83, 87–89, 92, 95, 98–101].

3. Research Significance & My Focus

As discussed in section 2, despite that many great mechanisms were proposed to achieve the lighter weight models, which are mainly about either reduce the number of weights and operations(Multiplying and Additions MACs) or reshape the filters in some way to achieve mathematically less FLOPs, that does not mean that this reduction will be reflecting the reduction on energy consumptions [21]. Similarly, energy-efficient models also do not intuitively imply lighter models, because the number of parameters within a model and number of calculations are insufficient for quantifying the energy consumptions of DNNs.

3.1. Significance

Present complex Artificial Intelligence Training progress particularly for Deep Neural Networks and Reinforcement Learning are still requires significant amount of energy, for real-life problems usually train a model needs professional GPU groups to accelerate the progress into an acceptable period. Works are being done to show that while SqueezeNet requires 50 times [54] fewer weights than AlexNet, it consumes more energy than AlexNet on different platforms[12].

Specifically, the Composite Kernels & Structured Convolution can be used in my research as fundamental model compression unit, but still there are space to improve, for example looking for even simpler mathematical convention that reduces the number of multiples or/and additions further at the mean time recover the accuracy lost in original propose.

MobileNets does spot a direction of research to reduce the network complexity, the Googles MobileNets proposals are only focused on its own mobile computing platform Android, hence the MobileNets are altered to fit more on Android, might be optimal on that, however not so true as a general model. It is designed particularly for mobile platforms, therefore does not represent the general energy-efficient model solution on general platforms. Also, according to MobileNets V3, the proposed nonlinear activation function h-swish does not work well in low-dimensional spaces, can still be improved.

Moreover, as for Knowledge Distilling, it is only efficient in saving inference power, although using it as pre-trained model in new model training does accelerates the training progress, but it does not save any training energy, energy consumptions for training were just performed at the same quantity level in pre-training stage. The Network Pruning, however, is certainly a direction of research, it does indicate a path of addressing energy consuming issue in training, however, deciding which parameters to be pruned, depends on specific tasks and is competitively complex, it would also be one my research focus.

The analysis above in section 2 and Energy-Efficient approach discussed in 2.5 shows that in order to reduce the energy consumption of models we need to paying specific attentions on lower-level hardware designs [43], but with the newer generations of platforms have been invented, the differences between different platforms matters a lot when talk about energy-efficiency, the research on a general solution on both lower-level hardware and higher-level programming [10]to address the energy consumption issue of training large DNN models is still a significant gap.

3.2. My Focus

In conclusion, my research will mainly focus on firstly find a way to erase the difference between lower-level architectures in abstract level, well quantifies the energy consumption in Training stage amount all kinds of computing platforms, paying attention not only on the energy overhead for processing the data to the MAC engines but the calculation consumptions.

Then start the development of a general method or model to reduce the training energy consumptions for Deep Neural Networks and Reinforcement Learning Control mechanisms, but still ensure the performance/accuracy of the model, where the less energy consumptions mains fewer computing powers, less memory, and hence less training time. Particularly, from the lower level, referencing from DNNs focused hardware designs like Tensor Processing Unit, Neural network Processing Unit trying to overcome the difficulties brought from data transferring and processing.

Moreover, from higher-level computing perspective, in cooperate with the proposing lower-level architecture, propose a new model or configuration to allow less energy consumptions on large DNN model training.

4. Methodology

It is preliminarily scheduled that during my 4 years PhD time, spend one and half year to read through all the existing methods and mechanisms as mentioned above, and implement them as indicated on the papers by myself to understand deeper about those important thoughts, they all point a clear direction of thinking for later researchers working on related filed. Then spend half a year to apply those methods on real-life applications, accumulates practice experiences, which I consider to be critical for me to underlining the problems and pros & cons of the current methods further.

After above, for the remaining 2 years, I will try to propose my own model or mechanisms, the thinking direction of such propose will based on my own thoughts combined with my previous practices and readings. Roughly the question will be addressed from the following aspects, personally, I don't tend to go into model design because it currently lacks theoretical support and is more like experimental science, hence the uncertainty is greater; pruning can be investigated; distillation works on almost all tasks and adds no extra reasoning overhead; quantification to reduce the inference power is currently the most applied from my point of view, and of course, these views are still waiting for the validation in my 2 years of experiments.

The last thing I want to mention is, during my time of research, I would like to also pay attention to the fundamental methods used in Deep Neural Networks, querying questions regarding the basic ideas in Computer Vision, Reinforcement Learning and Artificial Intelligence in general, for example, current Artificial Intelligence are largely based on Artificial Neural Networks, since its ability of high-dimensional function fitting, thus can be used in feature extraction and learning, but instead of current Neural Network model, is there a better approach that can perform high-dimensional function fitting but with less costs.

References

[1] berkeleyuni. https://chisel.eecs.berkeley.edu/. Accessed 28, Jul, 2022. 7

[2] codingninjas. https://www.codingninjas.com/codestudio/library/mobilenet. Accessed 28, Jul, 2022. 3

[3] getpebble. https://getpebble.com. Accessed 28, Jul, 2022. 7

[4] googleglass. http://www.google.com/glass/start/. Accessed 28, Jul, 2022. 7

[5] stanforduni. http://genesis2.stanford.edu/. Accessed 28, Jul, 2022. 7

[6] Mohamed M Sabry Aly, Mingyu Gao, Gage Hills, Chi-Shuen Lee, Greg Pitner, Max M Shulaker, Tony F Wu, Mehdi Asheghi, Jeff Bokor, Franz Franchetti, et al. Energy-efficient abundant-data computing: The n3xt 1,000 x. *Computer*, 48(12):24–33, 2015. 7

[7] Rhonda Ascierto. Uptime institute global data center survey. *Seattle: Uptime Insitute. https://uptimeinstitute.com/uptime_assets/f7bb01a900c060cc9abe42bb084609f63f02e448f5df1ca7ba7fdebb746cd1c4-2018-data-center-industry-survey. pdf*, 2018. 5

[8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 6

[9] Frank Bellosa, Andreas Weissel, Martin Waitz, and Simon Kellner. Event-driven energy accounting for dynamic thermal management. In *Proceedings of the Workshop on Compilers and Operating Systems for Low Power (COLP'03)*, volume 22, 2003. 6, 7

[10] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011. 7

[11] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012. 6

[12] Ramon Bertran, Marc Gonzalez, Xavier Martorell, Nacho Navarro, and Eduard Ayguade. Decomposable and responsive power models for multicore processors using performance counters. In *Proceedings of the 24th ACM International Conference on Supercomputing*, pages 147–158, 2010. 6, 7

[13] Yash Bhalgat, Yizhe Zhang, Jamie Menjay Lin, and Fatih Porikli. Structured convolutions for efficient neural network design. *Advances in Neural Information Processing Systems*, 33:5553–5564, 2020. 2

[14] Albert Bifet and Ricard Gavalda. Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM international conference on data mining*, pages 443–448. SIAM, 2007. 6

[15] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016. 7

[16] David Brooks, Vivek Tiwari, and Margaret Martonosi. Wattch: A framework for architectural-level power analysis and optimizations. *ACM SIGARCH Computer Architecture News*, 28(2):83–94, 2000. 7

[17] Bruno Burger. Net public electricity generation in germany in 2018. *Fraunhofer Institute for Solar Energy Systems ISE: Freiburg im Breisgau, Germany*, 2019. 5

[18] Doug Burger and Todd M Austin. The simplescalar tool set, version 2.0. *ACM SIGARCH computer architecture news*, 25(3):13–25, 1997. 7

[19] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2016. 5

[20] Chin-Jui Chang, Yu-Wei Chu, Chao-Hsien Ting, Hao-Kang Liu, Zhang-Wei Hong, and Chun-Yi Lee. Reducing the deployment-time inference control costs of deep reinforcement learning agents via an asymmetric architecture. In

2021 IEEE International Conference on Robotics and Automation (ICRA), pages 4762–4768. IEEE, 2021. 2

[21] Y Chen, Tien-Ju Yang, Joel Emer, and Vivienne Sze. Understanding the limitations of existing energy-efficient design approaches for deep neural networks. *Energy*, 2(L1):L3, 2018. 4, 7

[22] Yu-Hsin Chen, Tushar Krishna, Joel S Emer, and Vivienne Sze. Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. *IEEE journal of solid-state circuits*, 52(1):127–138, 2016. 6

[23] Gary Cook, Jude Lee, Tamina Tsai, Ada Kong, John Deans, Brian Johnson, and Elizabeth Jardim. Clicking clean: who is winning the race to build a green internet. *Greenpeace Inc., Washington, DC*, 5, 2017. 5

[24] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Training deep neural networks with low precision multiplications. *arXiv preprint arXiv:1412.7024*, 2014. 7

[25] Andrew Danowitz, Kyle Kelley, James Mao, John P Stevenson, and Mark Horowitz. Cpu db: recording microprocessor history. *Communications of the ACM*, 55(4):55–63, 2012. 5

[26] Howard David, Eugene Gorbatov, Ulf R Hanebutte, Rahul Khanna, and Christian Le. Rapl: Memory power estimation and capping. In *2010 ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED)*, pages 189–194. IEEE, 2010. 5, 6

[27] Robert H Dennard, Jin Cai, and Arvind Kumar. A perspective on today's scaling challenges and possible future directions. In *Handbook of Thin Film Deposition*, pages 3–18. Elsevier, 2018. 5

[28] Pedro Domingos and Geoff Hulten. Mining high-speed data streams. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 71–80, 2000. 7

[29] Timothy Dozat and Christopher D Manning. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*, 2016. 5

[30] Zidong Du, Krishna Palem, Avinash Lingamneni, Olivier Temam, Yunji Chen, and Chengyong Wu. Leveraging the error resilience of machine-learning applications for designing highly energy efficient accelerators. In *2014 19th Asia and South Pacific design automation conference (ASP-DAC)*, pages 201–206. IEEE, 2014. 7

[31] Michel Dubois, Murali Annavaram, and Per Stenström. *Parallel computer organization and design*. cambridge university press, 2012. 7

[32] Dimitris Economou, Suzanne Rivoire, Christos Kozyrakis, and Partha Ranganathan. Full-system power analysis and modeling for server environments. International Symposium on Computer Architecture (IEEE), 2006. 6

[33] EPA Emissions. Generation resource integrated database (egrid). *US Environmental Protection Agency: Washington, DC*, 2018. 5

[34] Eva García Martín. *Energy Efficiency in Machine Learning: Approaches to Sustainable Data Stream Mining*. PhD thesis, Blekinge Tekniska Högskola, 2020. 5, 6

[35] Eva García-Martín, Niklas Lavesson, Håkan Grahn, Emiliano Casalicchio, and Veselka Boeva. How to measure energy consumption in machine learning algorithms. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 243–255. Springer, 2018. 6

[36] C Gilberto and M Margaret. Power prediction for intel xscale processors using performance monitoring unit events power prediction for intel xscale processors using performance monitoring unit events. In *ISLPED*, volume 5, pages 8–10, 2005. 6

[37] Bhavishya Goel and Sally A McKee. A methodology for modeling dynamic and static power consumption for multicore processors. In *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 273–282. IEEE, 2016. 6

[38] Bhavishya Goel, Sally A McKee, Roberto Gioiosa, Karan Singh, Major Bhadauria, and Marco Cesati. Portable, scalable, per-core power estimation for intelligent resource management. In *International Conference on Green Computing*, pages 135–146. IEEE, 2010. 6

[39] Bhavishya Goel, Sally A McKee, and Magnus Själander. Techniques to measure, model, and manage power. In *Advances in Computers*, volume 87, pages 7–54. Elsevier, 2012. 6

[40] Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In *International conference on machine learning*, pages 1737–1746. PMLR, 2015. 6

[41] Vaibhav Gupta, Debabrata Mohapatra, Sang Phill Park, Anand Raghunathan, and Kaushik Roy. Impact: Imprecise adders for low-power approximate computing. In *IEEE/ACM International Symposium on Low Power Electronics and Design*, pages 409–414. IEEE, 2011. 7

[42] Philipp Matthias Gysel. *Ristretto: Hardware-oriented approximation of convolutional neural networks*. University of California, Davis, 2016. 6

[43] Md Rokib Hasan. Influence of device performance of sub-10 nm gan-based dg-mosfets over conventional si-based sg-mosfets. In *2017 4th International Conference on Advances in Electrical Engineering (ICAEE)*, pages 697–702. IEEE, 2017. 7

[44] Soheil Hashemi, Nicholas Anthony, Hokchhay Tann, R Iris Bahar, and Sherief Reda. Understanding the impact of precision quantization on the accuracy and energy of neural networks. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, pages 1474–1479. IEEE, 2017. 6

[45] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[46] John L Hennessy and David A Patterson. *Computer architecture: a quantitative approach*. Elsevier, 2011. 7

[47] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015. 4

[48] Mark Horowitz. 1.1 computing's energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pages 10–14. IEEE, 2014. 4

[49] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 3

[50] Russ Joseph and Margaret Martonosi. Run-time power estimation in high performance microprocessors. In *Proceedings of the 2001 international symposium on Low power electronics and design*, pages 135–140, 2001. 5, 6

[51] Patrick Judd, Jorge Albericio, Tayler Hetherington, Tor Aamodt, Natalie Enright Jerger, Raquel Urtasun, and Andreas Moshovos. Reduced-precision strategies for bounded memory in deep neural nets. *arXiv preprint arXiv:1511.05236*, 2015. 6

[52] Patrick Judd, Jorge Albericio, Tayler Hetherington, Tor M Aamodt, Natalie Enright Jerger, and Andreas Moshovos. Proteus: Exploiting numerical precision variability in deep neural networks. In *Proceedings of the 2016 International Conference on Supercomputing*, pages 1–12, 2016. 7

[53] Jonathan Koomey, Stephen Berard, Marla Sanchez, and Henry Wong. Implications of historical trends in the electrical efficiency of computing. *IEEE Annals of the History of Computing*, 33(3):46–54, 2010. 6, 7

[54] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 1, 7

[55] Liangzhen Lai, Naveen Suda, and Vikas Chandra. Deep convolutional neural network inference with floating-point weights and fixed-point activations. *arXiv preprint arXiv:1703.03073*, 2017. 6

[56] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2, 1989. 1

[57] Benjamin C Lee and David M Brooks. Accurate and efficient regression modeling for microarchitectural performance and power prediction. *ACM SIGOPS operating systems review*, 40(5):185–194, 2006. 7

[58] Jinsu Lee, Sanghoon Kang, Jinmook Lee, Dongjoo Shin, Donghyeon Han, and Hoi-Jun Yoo. The hardware and algorithm co-design for energy-efficient dnn processor on edge/mobile devices. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 67(10):3458–3470, 2020. 7

[59] Da Li, Xinbo Chen, Michela Becchi, and Ziliang Zong. Evaluating the energy efficiency of deep convolutional neural networks on cpus and gpus. In *2016 IEEE international conferences on big data and cloud computing (BDCloud), social computing and networking (SocialCom), sustainable computing and communications (SustainCom)(BDCloud-SocialCom-SustainCom)*, pages 477–484. IEEE, 2016. 5

[60] Sheng Li, Jung Ho Ahn, Richard D Strong, Jay B Brockman, Dean M Tullsen, and Norman P Jouppi. The mcpat framework for multicore and manycore architectures: Simultaneously modeling power, area, and timing. *ACM Transactions on Architecture and Code Optimization (TACO)*, 10(1):1–29, 2013. 7

[61] Darryl Lin, Sachin Talathi, and Sreekanth Annapureddy. Fixed point quantization of deep convolutional networks. In *International conference on machine learning*, pages 2849–2858. PMLR, 2016. 6

[62] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015. 5

[63] Hamid Reza Mahdiani, Ali Ahmadi, Sied Mehdi Fakhraie, and Caro Lucas. Bio-inspired imprecise computational blocks for efficient vlsi implementation of soft-computing applications. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 57(4):850–862, 2009. 7

[64] Krishna T Malladi, Frank A Nothaft, Karthika Periyathambi, Benjamin C Lee, Christos Kozyrakis, and Mark Horowitz. Towards energy-proportional datacenter memory with mobile dram. In *2012 39th Annual International Symposium on Computer Architecture (ISCA)*, pages 37–48. IEEE, 2012. 5

[65] E Matthew. Peters, mark neumann, mohit iyyer, matt gardner, christopher clark, kenton lee, luke zettlemoyer. deep contextualized word representations. In *Proc. of NAACL*, 2018. 5

[66] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013. 2

[67] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11264–11272, 2019. 4

[68] David Moloney. Embedded deep neural networks:"the cost of everything and the value of nothing". In *2016 IEEE Hot Chips 28 Symposium (HCS)*, pages 1–20. IEEE, 2016. 4

[69] Vojtech Mrazek, Syed Shakib Sarwar, Lukas Sekanina, Zdenek Vasicek, and Kaushik Roy. Design of power-efficient approximate multipliers for approximate artificial neural networks. In *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–7. ACM, 2016. 6

[70] Philip J Mucci, Shirley Browne, Christine Deane, and George Ho. Papi: A portable interface to hardware performance counters. In *Proceedings of the department of defense HPCMP users group conference*, volume 710. Citeseer, 1999. 7

[71] Kenneth O'brien, Ilia Pietri, Ravi Reddy, Alexey Lastovetsky, and Rizos Sakellariou. A survey of power and energy predictive models in hpc systems and applications. *ACM Computing Surveys (CSUR)*, 50(3):1–38, 2017. 7

[72] Priyadarshini Panda, Abhronil Sengupta, Syed Shakib Sarwar, Gopalakrishnan Srinivasan, Swagath Venkataramani, Anand Raghunathan, and Kaushik Roy. Cross-layer approximations for neuromorphic computing: From devices

to circuits and systems. In *Proceedings of the 53rd Annual Design Automation Conference*, pages 1–6, 2016. 7

[73] C Pavlatos and V Vita. Linguistic representation of power system signals. In *Electricity Distribution*, pages 285–295. Springer, 2016. 7

[74] John W Poulton, William J Dally, Xi Chen, John G Eyles, Thomas H Greer, Stephen G Tell, and C Thomas Gray. A 0.54 pj/b 20gb/s ground-referenced single-ended short-haul serial link in 28nm cmos for advanced packaging applications. In *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pages 404–405. IEEE, 2013. 6

[75] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 1

[76] Karthick Rajamani, Heather Hanson, Juan Rubio, Soraya Ghiasi, and Freeman Rawson. Application-aware power management. In *2006 IEEE International Symposium on Workload Characterization*, pages 39–48. IEEE, 2006. 6

[77] Shankar Ganesh Ramasubramanian, Rangharajan Venkatesan, Mrigank Sharad, Kaushik Roy, and Anand Raghunathan. Spindle: Spintronic deep learning engine for large-scale neuromorphic computing. In *2014 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pages 15–20. IEEE, 2014. 7

[78] Efraim Rotem, Alon Naveh, Avinash Ananthakrishnan, Eliezer Weissmann, and Doron Rajwan. Power-management architecture of the intel microarchitecture code-named sandy bridge. *Ieee micro*, 32(2):20–27, 2012. 7

[79] Mariagiovanna Sami, Donatella Sciuto, Cristina Silvano, and Vittorio Zaccaria. An instruction-level energy model for embedded vliw architectures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 21(9):998–1010, 2002. 7

[80] Muhammad Shafique, Rehan Hafiz, Muhammad Usama Javed, Sarmad Abbas, Lukas Sekanina, Zdenek Vasicek, and Vojtech Mrazek. Adaptive and energy-efficient architectures for machine learning: Challenges, opportunities, and research roadmap. In *2017 IEEE Computer society annual symposium on VLSI (ISVLSI)*, pages 627–632. IEEE, 2017. 6

[81] Muhammad Shafique, Rehan Hafiz, Semeen Rehman, Walaa El-Harouni, and Jörg Henkel. Cross-layer approximate computing: From logic to architectures. In *Proceedings of the 53rd Annual Design Automation Conference*, pages 1–6, 2016. 7

[82] Yakun Sophia Shao and David Brooks. Energy characterization and instruction-level energy model of intel's xeon phi processor. In *International Symposium on Low Power Electronics and Design (ISLPED)*, pages 389–394. IEEE, 2013. 6

[83] Karan Singh, Major Bhadauria, and Sally A McKee. Real time power estimation and thread scheduling via performance counters. *ACM SIGARCH Computer Architecture News*, 37(2):46–55, 2009. 6, 7

[84] Evgeny A Smirnov, Denis M Timoshenko, and Serge N Andrianov. Comparison of regularization methods for imagenet classification with deep convolutional neural networks. *Aasri Procedia*, 6:89–94, 2014. 4

[85] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012. 3

[86] David So, Quoc Le, and Chen Liang. The evolved transformer. In *International Conference on Machine Learning*, pages 5877–5886. PMLR, 2019. 3

[87] Vasileios Spiliopoulos, Andreas Sembrant, and Stefanos Kaxiras. Power-sleuth: A tool for investigating your program's power behavior. In *2012 IEEE 20th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 241–250. IEEE, 2012. 6, 7

[88] Gopalakrishnan Srinivasan, Parami Wijesinghe, Syed Shakib Sarwar, Akhilesh Jaiswal, and Kaushik Roy. Significance driven hybrid 8t-6t sram for energy-efficient synaptic storage in artificial neural networks. In *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 151–156. IEEE, 2016. 7

[89] W Nick Street and YongSeog Kim. A streaming ensemble algorithm (sea) for large-scale classification. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 377–382, 2001. 7

[90] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019. 1, 5

[91] Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. Linguistically-informed self-attention for semantic role labeling. *arXiv preprint arXiv:1804.08199*, 2018. 3

[92] David Stutz, Nandhini Chandramoorthy, Matthias Hein, and Bernt Schiele. Bit error robustness for energy-efficient dnn accelerators. *Proceedings of Machine Learning and Systems*, 3:569–598, 2021. 7

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

[94] Lucas Theis, Iryna Korshunova, Alykhan Tejani, and Ferenc Huszár. Faster gaze prediction with dense networks and fisher pruning. *arXiv preprint arXiv:1801.05787*, 2018. 4

[95] Vivek Tiwari, Sharad Malik, Andrew Wolfe, and Mike Tien-Chien Lee. Instruction level power analysis and optimization of software. In *Technologies for wireless computing*, pages 139–154. Springer, 1996. 7

[96] Minh Q Tran, Tuong Do, Huy Tran, Erman Tjiputra, Quang D Tran, and Anh Nguyen. Light-weight deformable registration using adversarial learning with distilling knowledge. *IEEE Transactions on Medical Imaging*, 2022. 2

[97] James W Tschanz, James T Kao, Siva G Narendra, Raj Nair, Dimitri A Antoniadis, Anantha P Chandrakasan, and Vivek De. Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage. *IEEE Journal of Solid-State Circuits*, 37(11):1396–1402, 2002. 6

[98] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 7

[99] Swagath Venkataramani, Kaushik Roy, and Anand Raghunathan. Substitute-and-simplify: A unified design paradigm for approximate and quality configurable circuits. In *2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1367–1372. IEEE, 2013. 7

[100] Neil HE Weste and David Harris. *CMOS VLSI design: a circuits and systems perspective*. Pearson Education India, 2015. 7

[101] Wu Ye, Narayanan Vijaykrishnan, Mahmut Kandemir, and Mary Jane Irwin. The design and use of simplepower: A cycle-accurate energy estimation tool. In *Proceedings of the 37th Annual Design Automation Conference*, pages 340–345, 2000. 7

[102] Yang You, Zhao Zhang, Cho-Jui Hsieh, James Demmel, and Kurt Keutzer. Imagenet training in minutes. In *Proceedings of the 47th International Conference on Parallel Processing*, pages 1–10, 2018. 2

[103] Chen Zhang, Di Wu, Jiayu Sun, Guangyu Sun, Guojie Luo, and Jason Cong. Energy-efficient cnn implementation on a deeply pipelined fpga cluster. In *Proceedings of the 2016 International Symposium on Low Power Electronics and Design*, pages 326–331, 2016. 6