EnergyNet: Energy-Efficient Dynamic Inference

Yue Wang[*1], **Tan Nguyen**[*1], **Yang Zhao**[2], **Zhangyang Wang**[3], **Yingyan Lin**[1], **and Richard Baraniuk**[1]

[1]Rice University, Houston, TX 77005, USA
[2]UC Santa Barbara, Santa Barbara, CA 93106, USA
[3]Texas A&M University, College Station, TX 77843, USA

Abstract

The prohibitive energy cost of running high-performance Convolutional Neural Networks (CNNs) has been limiting their deployment on resource-constrained platforms including mobile and wearable devices. We propose a CNN for energy-aware dynamic routing, called *EnergyNet*, that achieves adaptive-complexity inference based on the inputs, leading to an overall reduction of run time energy cost while actually improving accuracy. This is achieved by proposing an energy loss that captures both computational and data movement costs. We combine it with the accuracy-oriented loss, and learn a dynamic routing policy for skipping certain layers in the networks that optimizes the hybrid loss. Our empirical results demonstrate that, compared to the baseline CNNs, *EnergyNet* can trim down the energy cost by up to 40% and 65%, during inference on the CIFAR10 and Tiny ImageNet testing sets, respectively, while maintaining the same testing accuracy. It is further encouraging to observe that the energy awareness might serve as a training regularization that can improve the prediction accuracy: our models can achieve 0.7% higher top-1 testing accuracy than the baseline on CIFAR-10 when saving up to 27% energy, and 1.0% higher top-5 testing accuracy on Tiny ImageNet when saving up to 50% energy, respectively.

1 Introduction

While deep learning-powered Internet of Things (IoT) devices promise to dramatically revolutionize the way we live and work by enhancing our ability to recognize, analyze, and classify the world around us, this revolution has yet to be unleashed due to many fundamental challenges. Edge devices, such as smart phones, smart sensors, drones and robots, have limited energy and computation resources since they are battery-powered and have a small form factor. On the other hand, high-performance Convolutional Neural Networks (CNNs) come at a cost of prohibitive energy consumption [1]. The CNNs with the highest accuracy have hundreds of layers and tens of millions of parameters. When deployed in practice, such networks drain the battery very quickly [2].

Recently, there have been a number of methods proposed to reduce energy cost in CNNs, while not hampering their predictive power. Most of them aim to reduce the model size or the number of computations [3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. However, [2] shows that a smaller model size and fewer operations might not necessarily lead to a lower energy cost. [2] uses energy cost to guide the pruning process, where the layer with the highest energy cost is pruned first. [13] formulates the CNN training process as an optimization problem under a certain energy budget constraint. While both methods [2, 13] show promising results towards pursuing more energy-efficient CNN models, they do not incorporate energy costs into the training loss function to explicitly learn a more energy-efficient model. Furthermore, once their model structures are learned from training, it can only be fixed during the inference time, and there is no room for input-dependent adaptivity.

This paper proposes a new CNN model that combines energy cost with a dynamic routing strategy to enable adaptive energy-efficient inference. Our proposed model, termed as *EnergyNet*, is a gated CNN architecture which employs conditional computing to route the input data through the network

[*] Yue Wang and Tan Nguyen contributed equally to this work.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

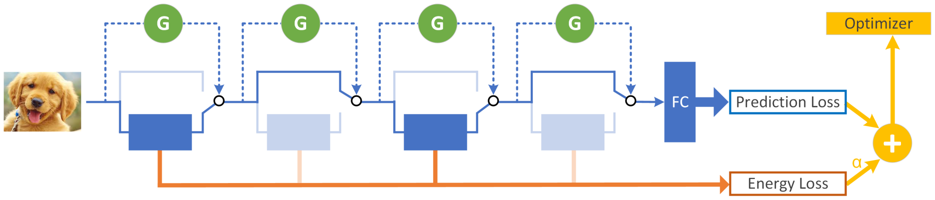

Figure 1: *EnergyNet* Structure: each green circle G indicates an RNN gate and each blue square under G indicates one block of layers in the base model. To reduce the energy cost, the RNN gates generate routing strategies dynamically for different input images. By sharing the parameters between all RNN gates, they will have only 0.04% of the energy cost of the base CNN model, which is negligible. In this specific example, only the first and third blocks get executed.

in an efficient path. Built on a *base network* (such as ResNet-34 or ResNet-50 [14]), *EnergyNet* uses an additional *gating network* [11] to decide whether the current input should skip certain layers in the network or not. It optimizes a weighted combination of an accuracy loss and an energy loss which captures both the computational and memory data movement costs, under which *EnergyNet* is trained to find the optimal routing policy to reduce the energy cost of the model without degrading the prediction accuracy. Our empirical results demonstrate that, compared to the base network without gating nor dynamic inference, *EnergyNet* can trim down the energy cost up to 40% and 65%, during inference on the CIFAR10 and Tiny ImageNet testing sets, respectively, while maintaining almost the same testing accuracy. Interestingly enough, we find the energy-aware *EnergyNet* can even achieve win-win, by simultaneously improving the prediction accuracy and saving energy, potentially due to its equivalent effect as a training regularization to avoid overfitting. For example, our models achieve 0.7% higher top-1 testing accuracy than the baseline on CIFAR-10 when saving up to 27% energy, and 1.0% higher top-5 accuracy on Tiny ImageNet when saving up to 50% energy, respectively.

2 Proposed *EnergyNet* Model

Overview: *EnergyNet* implements an effective dynamic routing algorithm using a set of gating networks, which shares similar ideas with [11], as depicted in Figure 1. Each gating network associates with a block of layers in the *EnergyNet*. Given an input image, the gating networks decide if the corresponding block should be skipped or not. The input to each block is first sent to the gating network G, whose output is either 0 or 1. If it is 0, the block will be skipped; otherwise, it will process the input normally as in the base model. If the input and output of the block have different dimensions, then we can perform a linear projection using a shortcut connection to match the dimensions as in [14]. The **core innovation** in *EnergyNet* is the adoption of a new energy-aware loss function for learning the gating (skipping) policy, whose details we defer to the next subsection.

In our implementation, we adopt the recurrent gates (RNNGates) as in [11] (see Figure 2). It is composed of a global average pooling followed by a linear projection that reduces the features to a 10-dimensional vector. A Long Short Term Memory (LSTM) [15] network that contains a single layer of dimension 10 is applied to generate a binary scalar. As mentioned in [11], this RNNGate design incurs a negligible overhead compared to its feed-forward counterpart (0.04% vs. 12.5% of the computation of the residual blocks when the baseline architecture is a ResNet). In order to further reduce the energy cost due to loading parameters into the memory, all RNNGates in the *EnergyNet* share the same weights.

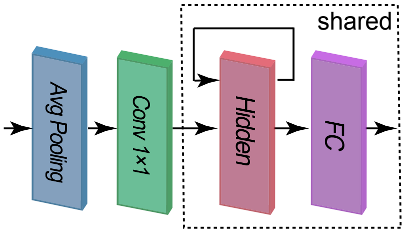

Figure 2: Gating networks in *EnergyNet* are RNNs that share weights (RNNGates). The RNNGates incurs a negligible overhead.

Energy-aware Learning for Dynamic Routing: The dynamic routing in *EnergyNet* is learned by minimizing an energy cost together with the accuracy loss. In particular, the learning goal in the *EnergyNet* is defined as:

$$\min_{W,G} \; L(W,G) \; + \alpha E(W,G) \tag{1}$$

Here, α is a weighting coefficient of the energy loss, and W and G denote the parameters of the base model and the gating network, respectively. Also, $L(W,G)$ denotes the prediction loss, and $E(W,G)$ denotes the energy cost of the CNN model associated with W and G, which is calculated by accumulating the energy cost of the layers that are not skipped. In order to compute the energy cost of each layer, we adopt the following energy model:

$$E = \sum_{i=1}^{N} \#_{acc_i} \times e_i + \#_{MAC} \times e_{MAC} \tag{2}$$

where e_i and e_{MAC} denote the energy costs of accessing the i-th memory hierarchy and one multiply-and-accumulate (MAC) operation [16], respectively, while $\#_{MAC}$ and $\#_{acc_i}$ denote the total number of MAC operations and accesses to the i-th memory hierarchy, respectively. Note that state-of-the-art CNN accelerators commonly employ such a hierarchical memory architecture for minimizing the dominant memory access and data movement costs. In this work, we consider the most commonly used design of three memory hierarchies including the main memory, the cache memory, and local register files [16], and employ a state-of-the-art simulation tool called "SCALE-Sim" [17] to calculate the number of memory accesses $\#_{acc_i}$ and the total number of MACs $\#_{MAC}$.

3 Experiments

Summary: We show that *EnergyNet* saves more energy than the baseline ResNet after training on CIFAR10 and Tiny ImageNet [18]. In particular, compared to the baseline ResNet, *EnergyNet* saves up to 40% and 65% energy cost without degrading the prediction accuracy, when processing CIFAR10 and TinyImageNet images, respectively. More encouragingly, our models can achieve 0.7% higher top-1 testing accuracy than the baseline on CIFAR-10 when saving up to 27% energy, and 1.0% higher top-5 testing accuracy on Tiny ImageNet when saving up to 50% energy, respectively.

Architectures and Training Details: We use the ResNet-38 and ResNet-50 in [14] as the baseline models for constructing and evaluating *EnergyNet* models on CIFAR-10 and Tiny ImageNet, with the resulting models denoted as *EnergyNet-38* and *EnergyNet-50*, respectively. The training process contains three steps. In step I, we set the weighting coefficient α to a small value (e.g., 0.1), which helps the model converge to the baseline accuracy first. In step II, we increase α to a larger value (e.g., 0.9) and retrain the model obtained from step I. For step III , it is only triggered if the model sees an accuracy loss larger than a threshold (default 0.1%) from step II: we then set α to a small value (e.g., 0.1) again to retrain the resulting model from step II for restoring the accuracy. Such a three-step strategy proves to help stabilize training and gain better performance.

Discussion: We use energy savings as a metric to quantify the resulting energy efficiency improvement of *EnergyNet*. The energy savings is defined as E_s/E_{total}, where E_{total} and E_s are the energy cost of the baseline model and the skipped layers due to *EnergyNet*. From Figure 3, we can conclude that *EnergyNet* achieves the goal of reducing energy cost while preserving or even improving the prediction accuracy. In particular, the accuracy of *EnergyNet-38* and *EnergyNet-50* will not drop when the energy savings is as high as 40% and 65%, respectively. To confirm that these experimental results are not just a coincidence, we performed 20 trials of experiments using *EnergyNet-38* and observed that the confidence interval with a 95% confidence level for the mean of the prediction accuracy and the energy savings are [92.47%, 92.58%] and [39.55%, 40.52%], respectively, verifying the reproducibility of *EnergyNet*'s prediction accuracy and resulting energy savings.

We observe that *EnergyNet* can achieve a higher accuracy than the original ResNet model. We conjecture that this is because *EnergyNet* can overcome overfitting when performing the dynamic routing for energy savings. Further *EnergyNet* can aggressively reduce energy cost by about $4\times$, over both the ResNet-38 and ResNet-50 baselines, at the cost of 3% and 4% testing accuracy losses , respectively.

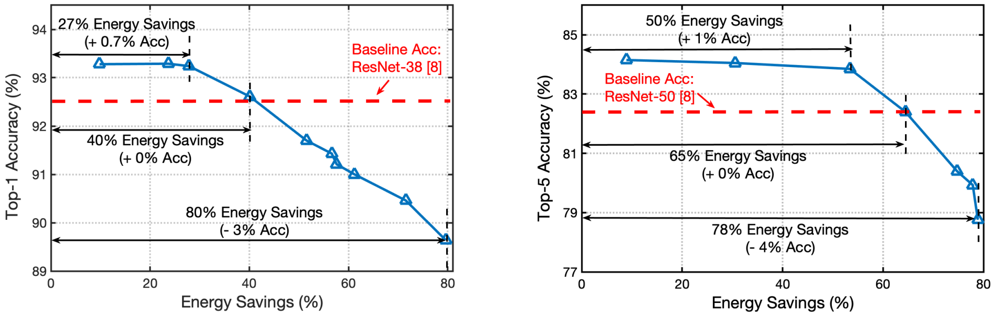

Figure 3: Top-1 accuracy (Acc) vs. energy savings for *EnergyNet-38* (left) and Top-5 accuracy (Acc) vs. energy savings for *EnergyNet-50* (right).

Acknowledgements

We would like to acknowledge Dr. Edwin Park and Qualcomm Technologies, Inc for providing funding support in this research.

References

[1] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[2] Tien-Ju Yang, Yu-Hsin Chen, and Vivienne Sze. Designing energy-efficient convolutional neural networks using energy-aware pruning. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5687–5695, 2017.

[3] Sourav Bhattacharya and Nicholas D Lane. Sparsification and separation of deep learning layers for constrained resource inference on wearables. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, pages 176–189. ACM, 2016.

[4] Zhangyang Wang, Jianchao Yang, Hailin Jin, Eli Shechtman, Aseem Agarwala, Jonathan Brandt, and Thomas S. Huang. Deepfont: Identify your font from an image. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 451–459. ACM, 2015.

[5] Soravit Changpinyo, Mark Sandler, and Andrey Zhmoginov. The power of sparsity in convolutional neural networks. *arXiv preprint arXiv:1702.06257*, 2017.

[6] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *International Conference on Learning Representations*, 2016.

[7] 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.

[8] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50× fewer parameters and$<$ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[9] Nicholas D. Lane, Sourav Bhattacharya, Petko Georgiev, Claudio Forlivesi, Lei Jiao, Lorena Qendro, and Fahim Kawsar. DeepX: A software accelerator for low-power deep learning inference on mobile devices. In *Proceedings of the 15th International Conference on Information Processing in Sensor Networks (IPSN)*, page 23. IEEE Press, 2016.

[10] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[11] Xin Wang, Fisher Yu, Zi-Yi Dou, and Joseph E. Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. *arXiv preprint arXiv:1711.09485*, 2017.

[12] Junru Wu, Yue Wang, Zhenyu Wu, Zhangyang Wang, Ashok Veeraraghavan, and Yingyan Lin. Deep k-means: Re-training and parameter sharing with harder cluster assignments for compressing deep convolutions. In *International Conference on Machine Learning*, pages 5359–5368, 2018.

[13] Haichuan Yang, Yuhao Zhu, and Ji Liu. End-to-end learning of energy-constrained deep neural networks. *arXiv preprint arXiv:1806.04321*, 2018.

[14] 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.

[15] Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with LSTM. In *Neural Computation*, volume 12, pages 2451–2471, 1999.

[16] Yu-Hsin Chen, Joel Emer, and Vivienne Sze. Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks. In *ACM SIGARCH Computer Architecture News*, volume 44, pages 367–379. IEEE Press, 2016.

[17] Ananda Samajdar, Yuhao Zhu, and Paul Whatmough. Systolic CNN AcceLErator Simulator (SCALE Sim). 2017.

[18] Leon Yao and John Miller. Tiny imagenet classification with convolutional neural networks. *CS 231N*, 2015.