Supplementary Material

Souvik Kundu, Massoud Pedram, Peter A. Beerel
University of Southern California, Los Angeles, CA, USA
{souvikku, pedram, pabeerel}@usc.edu

1. Model Details

We used variants of VGG5, VGG11, and ResNet12 models for our experimental evaluations. We followed recommendations from [3, 2] to design the ANN models that yield low conversion loss. In particular, our ANN models do not have any batch norm layer nor bias terms. To minimize the information loss of binary-spike-driven activation maps we avoided the use of any max-pool layers. The details of each model is presented in Table 1.

2. Training Hyperparameters

For ANN training, we used the SGD optimizer and trained with batches of 64 images. For SNN training, we used the ADAM optimizer with momentum of 0.9, a dropout rate of 0.2, and a batch-size of 32. For ANN-to-SNN conversion, we used a single batch of only 512 images to evaluate the threshold for each layer.

3. Memory Usage and Training Time

Fig. 1 shows the normalized memory for traditional SNN training (Φ^T_{SNN}) with rate-coded and direct inputs and the proposed SNN training (Φ^P_{SNN}). The time steps for rate-coded and direct input were 200 and 8, respectively. Due to memory limitations, we used a batch-size of 16 for rate-coded SNNs. For direct input SNNs the batch size is kept as 32. As we can see in the Fig. 1 compared to rate-coded SNN training, our proposed training requires up to 2.68× less memory.

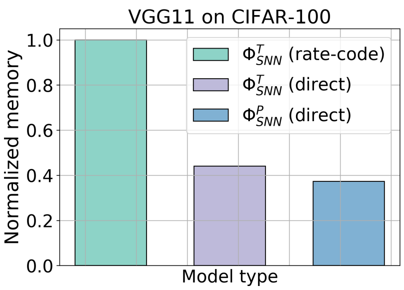

Figure 1. Comparison of memory usage for traditional vs. proposed SNN training.

Fig. 2 shows a comparison of the training time for the various SNN training strategies. Compared to traditional training with rate-coded inputs, the proposed training algorithm requires up to 30.8× less time[1].

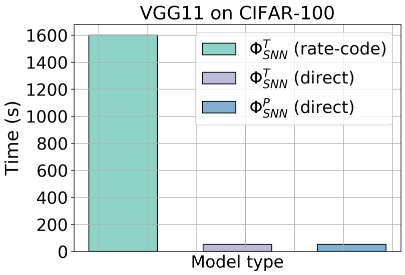

Figure 2. Comparison of training time for 6,400 images for traditional vs. proposed SNN training.

4. More Results

Attack strength vs test accuracy for VGG11 on CIFAR-100. Fig. 3 presents the model performance of VGG11 on CIFAR-100 under white box PGD attack with increasing attack strength. As Fig. 3(a) depicts, the performance against white box PGD generated images approaches $\sim$0% accuracy when the attack bound ϵ increases. We also tested our scheme with increasing iterations K and found that the adversarial performance reaches an asymptote as K increases beyond 40 (Fig. 3(b)). These trends are similar to our observations with VGG5 on CIFAR-10 (Fig. 8 of the original manuscript).

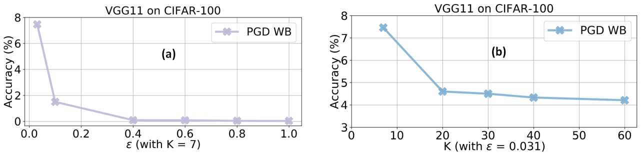

Figure 3. White-box PGD attack performance as a function of (a) bound ϵ and (b) attack iterations K with VGG11 on CIFAR-100.

Weight distribution visualization. HIRE-SNN models tend to have fewer weights having insignificant absolute

[1] All experiments were done on a NVIDIA 2080 Ti GPU

Model	VGG5	VGG11	ResNet12
# Parameters	50.6 M	61.36 M	4.99 M
Architecture	[C64/3 × 3], [AVG 2 × 2, 2] [C128/3 × 3], [Drop(pr)] [C128/3 × 3], [AVG 2 × 2, 2] 3FC	[C64/3 × 3], [AVG 2 × 2, 2] [C128/3 × 3], [Drop(pr)] [C256/3 × 3], [AVG 2 × 2, 2] [C512/3 × 3] × 2, [Drop(pr)]×2 [C512/3 × 3], [AVG 2 × 2, 2] [C512/3 × 3] × 2, [Drop(pr)]×2 3FC	[C64/3 × 3] × 2, [Drop(pr)]×4 [C64/3 × 3], [AVG 2 × 2, 2] [C64/3 × 3], [Drop(pr)] [C64/3 × 3] + I [C128/3 × 3], [Drop(pr)] [C128/3 × 3] + [C128/1 × 1] [C256/3 × 3], [Drop(pr)] [C256/3 × 3] + [C256/1 × 1] [C512/3 × 3], [Drop(pr)] [C512/3 × 3] + [C512/1 × 1] 1FC

Table 1. Description of the models used for our experiments. Activation layers are omitted for for the sake of brevity. Each of the convolution (C) layers are specified by its number of filters and kernel size. Average pooling layers (AVG) are specified with kernel and stride size. Dropout layers (Drop) are specified with corresponding drop probability pr. The repetition number of a layer is specified outside the corresponding bracket. For the 3×3 and 1×1 CONV layers strides are 1 and 2, respectively. I represents an identity layer. Parameters are computed for CIFAR-10 dataset (meaning the output classifier has 10 classes).

values compared to the ones produced by direct-input traditional SNN training, as exemplified in Fig. 4(a). This is a generally observed trend in adversarially robust models [1, 4]. Due to the comparatively higher weight magnitudes, the HIRE-SNN models do incur higher average SA compared to traditionally trained models on direct inputs. Layerwise this increment in SA can vary from ∼30% to ∼69% as is shown in Fig. 4 (b).

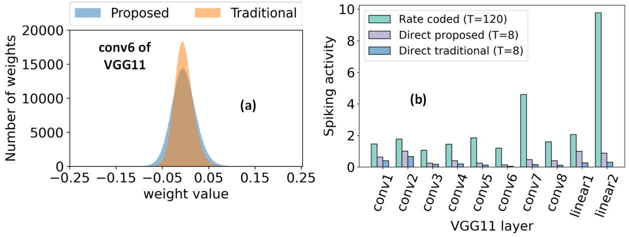

Figure 4. (a) Weight histogram plot of a model trained using proposed and traditional approaches, respectively. (b) Comparison of spiking activity for VGG11, averaged over 32 test samples.

Ablation with $\mathcal{N}$. Fig. 5(a) shows the ablation with different $\mathcal{N}$. The basic motivation to pick hyperparameters $\mathcal{N}$, ϵ_s, and ϵ_t is to ensure there is only an insignificant drop in the clean image accuracy while still improving the adversarial performance. Because different models produce perturbed images of different strengths, we hand-tuned ϵ_s (ϵ_t).

Performance comparison with an adversarially trained iso-architecture ANN. Fig. 5(b) shows the performance comparison of an adversarially trained ANN and corresponding HIRE-SNN requiring 2× and 1× training time compared to their respective baselines, respectively.

References

[1] Souvik Kundu, Mahdi Nazemi, Peter A Beerel, and Massoud Pedram. Dnr: A tunable robust pruning framework through dynamic network rewiring of dnns. In *Proceedings of the 26th Asia and South Pacific Design Automation Conference*, pages 344–350, 2021. 2

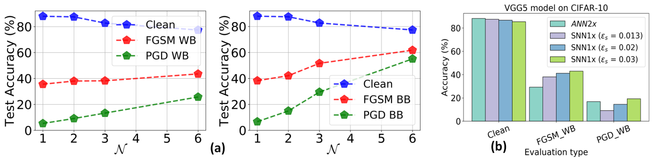

Figure 5. (a) Accuracy vs. iterations $\mathcal{N}$ ($T = 6$) for VGG5 on CIFAR-10 under both clean and adversarial images using WB and BB attacks. As the perturbations become stronger with increased $\mathcal{N}$, the clean-image classification performance drops. (b) Performance for ANN and HIRE-SNN trained with different ϵ_s.

[2] Nitin Rathi, Gopalakrishnan Srinivasan, Priyadarshini Panda, and Kaushik Roy. Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation. *arXiv preprint arXiv:2005.01807*, 2020. 1

[3] Abhronil Sengupta, Yuting Ye, Robert Wang, Chiao Liu, and Kaushik Roy. Going deeper in spiking neural networks: VGG and residual architectures. *Frontiers in Neuroscience*, 13:95, 2019. 1

[4] Shaokai Ye, Kaidi Xu, Sijia Liu, Hao Cheng, Jan-Henrik Lambrechts, Huan Zhang, Aojun Zhou, Kaisheng Ma, Yanzhi Wang, and Xue Lin. Adversarial robustness vs. model compression, or both? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2