Layerwise Hebbian/anti-Hebbian (HaH) Learning In Deep Networks: A Neuro-inspired Approach To Robustness

Metehan Cekic[1] Can Bakiskan[1] Upamanyu Madhow[1]

Abstract

We propose a neuro-inspired approach for engineering robustness into deep neural networks (DNNs), in which end-to-end cost functions are supplemented with layer-wise costs promoting Hebbian (“fire together,” “wire together”) updates for highly active neurons, and anti-Hebbian updates for the remaining neurons. Unlike standard end-to-end training, which does not directly exert control over the features extracted at intermediate layers, Hebbian/anti-Hebbian (HaH) learning is aimed at producing sparse, strong activations which are more difficult to corrupt. We further encourage sparsity by introducing competition between neurons via divisive normalization and thresholding, together with implicit ℓ_2 normalization of neuronal weights, instead of batch norm. Preliminary CIFAR-10 experiments demonstrate that our neuro-inspired model, trained without augmentation by noise or adversarial perturbations, is substantially more robust to a range of corruptions than a baseline end-to-end trained model. This opens up exciting research frontiers for training robust DNNs, with layer-wise costs providing a strategy complementary to that of data-augmented end-to-end training.

1. Introduction

Since their original breakthrough in image classification performance, DNNs trained with backpropagation have attained outstanding performance in a wide variety of fields (Brown et al., 2020; Silver et al., 2018; Akkaya et al., 2019; Senior et al., 2020). Arguably, a key contributor to this explosive growth is the evolution of a powerful yet generic computational infrastructure for training DNNs with a very large number of parameters with variants of stochastic gradient descent on an end-to-end cost function. Yet there remain fundamental concerns regarding the lack of robustness in DNNs (e.g, to noise, distribution shifts, and adversarial perturbations). Within the existing training paradigm, the main recourses are modification of the end-to-end cost function and/or augmentation of the input data. For example, the state of the art defense against adversarial attacks is adversarial training (see (Madry et al., 2018) and variants thereof), which augments the input data with adversarial perturbations during training, while the cost function in (Zhang et al., 2019) seeks to trade off clean accuracy and attacked accuracy.

In this paper, we explore a complementary approach to robustness based on supplementing the end-to-end cost function with layer-wise costs aimed at shaping the features extracted by intermediate layers of the DNN. Specifically, while standard DNNs produce a large fraction of small activations at each layer, we seek architectures which produce a small fraction of strong activations, while continuing to utilize existing network architectures for feedforward inference and existing software infrastructure for stochastic gradient training. To this end, we introduce neuro-inspired mechanisms creating competition between neurons during both training and inference.

1.1. Approach and Contributions

In order to attain sparse, strong activations at each layer, we employ the following neuro-inspired strategy for modifying standard DNN training and architecture:
Hebbian/anti-Hebbian (HaH) Training: We supplement a standard end-to-end discriminative cost function with layer-wise costs at each layer which promote neurons producing large activations and demote neurons producing smaller activations. The goal is to develop a neuronal basis that produces a distributed sparse code, without requiring a reconstruction cost as in standard sparse coding (Olshausen & Field, 1997).
Neuronal Competition via Normalization: We further increase sparsity by introducing Divisive Normalization (DN), which enables larger activations to suppress smaller activations. In order to maintain a fair competition among neurons, we introduce Implicit ℓ_2 Normalization of the neu-

[1] Department of Electrical Engineering, UC Santa Barbara, Santa Barbara, CA, USA. Correspondence to: Metehan Cekic <metehancekic@ucsb.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

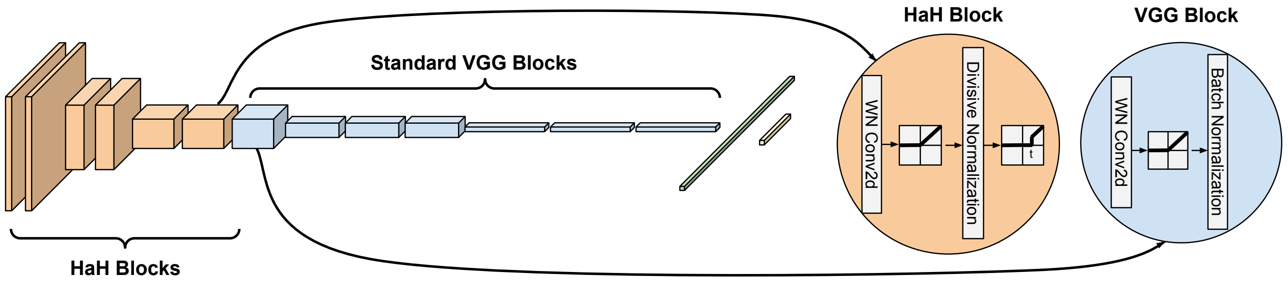

Figure 1: Our model consists of two different types of blocks: first 6 blocks are Hebbian-anti-Hebbian (HaH) while the rest are regular VGG blocks. HaH blocks use a weight normalized convolutional layer, followed by ReLU, divisive normalization and thresholding. Regular VGG blocks use a weight normalized convolutional layer followed by ReLU and batch norm.

ronal weights, so that each activation may be viewed as a geometric projection of the layer input onto the "direction" of the neuron. (Using implicit rather than explicit weight normalization in our inference architecture simplifies training.)

We report on experiments with CIFAR-10 image classification, comparing a baseline VGG-16 network trained end-to-end against the same architecture with HaH training and DN. Both architectures employ implicit weight normalization, which we have verified does not adversely impact accuracy. We demonstrate that the activations in our proposed architecture are indeed more sparse than for the baseline network. In order to isolate the impact of our training approach and inference architecture, we do not employ noise augmentation or adversarial training in these initial experiments. For CIFAR10 classification, we show that our model is significantly more robust than a baseline model against both noise and adversarial perturbations. Against the broader set of corruptions in the CIFAR10-C dataset (Common corruptions dataset), our model is generally more resilient than both the baseline model and an adversarially trained model.

1.2. Related Work

Hebbian learning has a rich history in artificial neural networks, dating back to the neocognitron (Fukushima et al., 1983), and including recent attempts at introducing it into deep architectures (Amato et al., 2019). However, to the best of our knowledge, ours is the first paper to clearly demonstrate gains in robustness from its incorporation in DNNs. Divisive normalization is a widely accepted concept in neuroscience (Carandini & Heeger, 2012; Burg et al., 2021), and versions of it have been shown to be competitive with other normalization techniques in deep networks (Ren et al., 2016). Our novel contribution is in showing that divisive normalization can be engineered to enhance sparsity and robustness. Finally, sparse coding with a reconstruction objective was shown to lead to neuro-plausible outcomes in a groundbreaking paper decades ago (Olshausen & Field, 1997). In contrast to the iterative sparse coding and dictionary learning in such an approach, our HaH-based training targets strong sparse activations in a manner amenable to standard stochastic gradient training.

Recent work showing potential robustness gains by directly including known aspects of mammalian vision in DNNs includes (Dapello et al., 2020), which employs Gabor filter blocks and stochasticity, and (Li et al., 2019), which employs neural activity measurements from mice for regularization in DNNs. Rather than incorporating specific features from biological vision, we use neuro-inspiration to extract broad principles that can be folded into data-driven learning and inference in DNNs.

2. Model

We now describe how we incorporate HaH training and divisive normalization into a standard CNN for image classification. We consider a "classical" CNN for our experiments– VGG-16 (Simonyan & Zisserman, 2014) applied to CIFAR-10, rather than variants of ResNet (He et al., 2015), because residual connections complicate our interpretation of building models from the bottom-up using HaH learning. Since we wish to build robustness from the bottom up, we modify the first few convolutional blocks to incorporate neuro-inspired principles. We term these modified blocks "HaH blocks."

Each HaH block employs convolution with implicit weight normalization, followed by ReLU, then divisive normalization, and then thresholding. Implicit weight normalization enables us to interpret the convolution outputs for each filter as projections, and we have verified that employing it in all blocks of a baseline VGG-16 architecture does not adversely impact accuracy (indeed, it slightly improves it). Each standard (non-HaH) block in our architecture therefore also employs convolution with implicit weight normalization, followed by ReLU, but uses batch norm rather than divisive normalization. Each HaH block contributes a HaH cost for training, so that the overall cost function used for training is the standard discriminative cost and the sum of the HaH costs from the HaH blocks.

We now describe the key components of our architecture, shown in Figure 1.

2.1. Inference in a HaH block

Implicit weight normalization: Representing the convolution output at a given spatial location from a given filter as a tensor inner product $\langle \cdot, \cdot \rangle$ between the filter weights $\mathbf{w}$ and the input $\mathbf{x}$, the output of the ReLU unit following the filter is given by

$$y = \text{ReLU}\left(\frac{\langle \mathbf{w}, \mathbf{x} \rangle}{||\mathbf{w}||_2}\right) \tag{1}$$

This effectively normalizes the weight tensor of each filter to unit ℓ_2 norm, without actually having to enforce an ℓ_2 norm constraint in the cost.

Divisive normalization: If we have N filters in a given HaH block, let $y_1(loc), ..., y_N(loc)$ denote the corresponding activations computed as in (Equation 1) for a given spatial location loc. Let $M(loc) = \frac{1}{N}\sum_{k=1}^{N} y_k(loc)$ denote the mean of the activations at a given location, and let $M_{max} = \max_{loc} M(loc)$ denote the maximum of this mean over all locations. We normalize each activation using these terms as follows:

$$z_k(loc) = \frac{y_k(loc)}{\sigma M_{max} + (1-\sigma) M(loc)}\ ,\ k = 1, ..., N \tag{2}$$

where $0 \leq \sigma \leq 1$ is a hyperparameter which can be separately tuned for each HaH block. Thus, in addition to creating competition among neurons at a given location by dividing by $M(loc)$, we also include M_{max} in the denominator in order to suppress contributions at locations for which the input is "noise" rather than a strong enough "signal" well-aligned with one or more of the filters. This particular implementation of divisive normalization ensures that the output of a HaH-block is scale-invariant (i.e., we get the same output if we scale the input to the block by any positive scalar).

Adaptive Thresholding: Finally, we ensure that each neuron is producing significant outputs by neuron-specific thresholding after divisive normalization. The output of the kth neuron at location loc is given by

$$o_k(loc) = \begin{cases} z_k(loc) & \text{if } z_k(loc) \geq \tau_k \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where the threshold τ_k is neuron and image specific. For example, we may set τ_k to the 90th percentile of the statistics of $z_k(loc)$ in order to get an activation rate of 10% for each neuron for every image. Another simple choice that works well, but gives higher activation rates, is to set τ_k to the mean of $z_k(loc)$ for each image.

2.2. HaH Training

For an N-neuron HaH block with activations $y_k(loc)$, $k = 1, ..., N$ at location loc, the Hebbian/anti-Hebbian cost seeks to maximize the average of the top K activations, and to minimize the average of the remaining $N - K$ activations, where K is a hyperparameter. Thus, sorting the activations $\{y_k(loc)\}$ so that $y^{(1)}(loc) \geq y^{(2)}(loc) \geq ... \geq y^{(N)}(loc)$, the contribution to the HaH cost (to be maximized) is given by

$$L_{block}(loc) = \frac{1}{K}\sum_{k=1}^{K} y^{(k)}(loc) - \lambda \frac{1}{N-K}\sum_{k=K+1}^{N} y^{(k)}(loc) \tag{4}$$

where $\lambda \geq 0$ is a hyperparameter determining how much to emphasize the anti-Hebbian component of the adaptation. The overall HaH cost for the block, L_{block}, which we wish to maximize, is simply the mean over all locations and images.

The overall loss function to be minimized is now given by

$$L = L_{disc} - \sum_{\text{HaH blocks}} \alpha_{block} L_{block} \tag{5}$$

where L_{disc} is the standard discriminative loss, and $\{\alpha_{block} \geq 0\}$ are hyper-parameters determining the relative weight of the HaH costs across blocks.

3. Experiments

We consider VGG-16 with the first 6 blocks (each block includes conv, ReLU, batch norm) replaced by HaH blocks (each block includes conv, ReLU, divisive norm, thresholding). In our training, we use Adam optimizer (Kingma & Ba, 2014) with an initial learning rate of 10^{-3}, multiplied by 0.1 at epoch 60 and again at epoch 80. We train all models for 100 epochs on CIFAR-10. We choose τ_k in Equation 3 to keep 20% of activations. We use $[4.5 \times 10^{-3}$, 2.5×10^{-3}, 1.3×10^{-3}, 1×10^{-3}, 8×10^{-4}, $5 \times 10^{-4}]$ for α in Equation 5. We use 0.1 for λ and set K to 10% of number of filters in each layer in Equation 4 and set $\sigma = 0.1$ in Equation 2. Details about other hyper-parameters can be found in our code in supplementary materials.

Sparser activations: To ensure that HaH blocks are operating as intended and achieving the sparse and strong activations we test the sparsity levels of intermediate representations and plot them in Figure 2. Sparsity is computed by the ratio of ℓ_1 norm to ℓ_2 norm (also known as Hoyer term (Hoyer, 2004)) of each spatial location's representation across the channel dimension. We then linearly normalize the values to lie in [0,1]. Lower values represent sparser representations. The activations in these first 6 blocks are indeed more sparse for our architecture than for baseline VGG.

Enhanced robustness to noise: We borrow the concept of signal-to-noise-ratio (SNR) from wireless communication to obtain a block-wise measure of robustness. Let $f_n(x)$ denote the input tensor at block n in response to clean image x, and $f_n(x + w)$ the input tensor when the image is

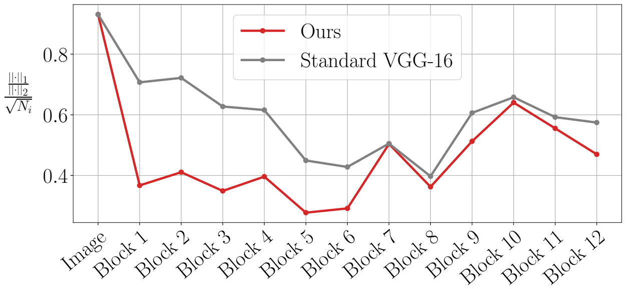

Figure 2: HaH blocks yield sparser activations than baseline. The measure of sparsity is the Hoyer ratio (Hoyer, 2004) of ℓ_1 norm to ℓ_2 norm of activations across channels, averaged across spatial locations, and then normalized to lie in [0,1] (lower values correspond to more sparsity).

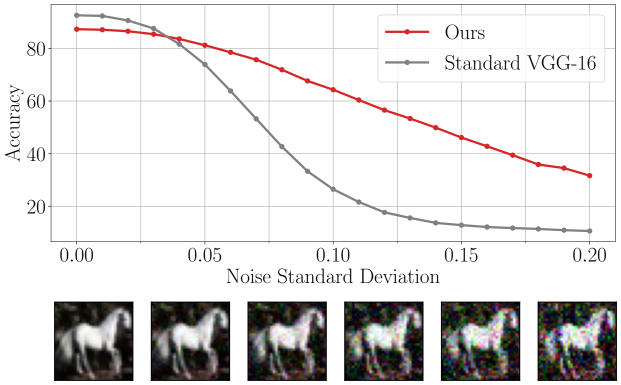

Figure 4: Comparison of classification accuracies as a function of noise σ. To provide a concrete sense of the impact of noise, noisy images at increasing values of σ are shown below the graph.

corrupted by noise w. As illustrated in Figure 3a, we define SNR as

$$\text{SNR}_n = 10 \log_{10} \left(\mathbb{E}_{x \sim \mathcal{D}_{test}} \left[\frac{||f_n(x)||_2^2}{||f_n(x+w) - f_n(x)||_2^2} \right] \right) dB \quad (6)$$

converting to logarithmic decibel (dB) scale as is common practice. Figure 3b shows that the SNR for our model comfortably exceeds that of the standard model, especially in the first 6 HaH blocks.

These higher SNR values also translate to gains in accuracy with noisy images: Figure 4 compares the accuracy of our model and the base model for different levels of Gaussian noise. There are substantial accuracy gains at high noise levels: 64% vs. 26% at a noise standard deviation of 0.1, for example.

Enhanced robustness to adversarial attacks: While we have not trained with adversarial examples, we find that, as expected, the noise rejection capabilities of the HaH blocks also translates into gains in adversarial robustness relative to the baseline VGG model. This holds for state-of-the-art gradient-based attacks (Madry et al., 2018; Pintor et al., 2021), as well as AutoAttack, an ensemble of parameter-free attacks suggested by RobustBench (Croce et al., 2020). We observe no additional benefit of using gradient-free attacks, and conclude that the robustness provided by our scheme is not because of gradient-masking. Because of space constraints, we only report on results from minimum-norm adversarial attacks and AutoAttack.

Figure 5 shows that the minimum distortion needed to flip the prediction of our model (computed using the recently proposed fast minimum norm computation method (Pintor et al., 2021)) is higher for our model for all the ℓ_p attacks considered.

We have also obtained substantial gains in adversarial accuracy against all four ℓ_p norm attacks ($p = 0, 1, 2, \infty$) used as benchmarks in adversarial machine learning. Table 1 displays a subset of results demonstrating accuracy gains against noise and adversarial perturbations, at the expense of a slight decrease in clean accuracy.

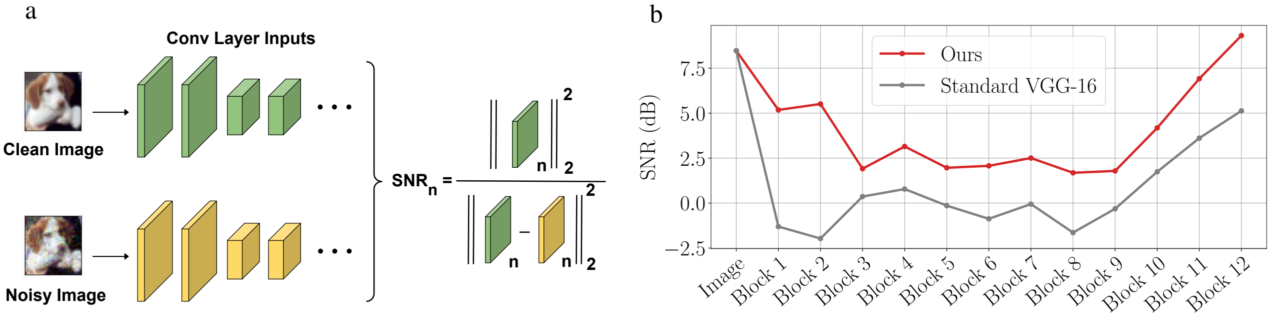

Figure 3: **a**: To compute the SNR at the n^{th} block inputs, we divide the ℓ_2 norm of the block input corresponding to clean image by the ℓ_2 norm of the difference of block corresponding to clean and noisy images. **b**: Comparison of SNR values of the block inputs for the standard base model (gray) and ours (red).

Table 1: Enhanced accuracy against noise and adversarial attacks.

	Clean	Noisy ($\sigma = 0.1$)	Adv (ℓ_∞) ($\epsilon = 2/255$)	Adv (ℓ_2) ($\epsilon = 0.25$)
Standard	**92.5%**	26.6%	10.4%	13.9%
Ours	87.3%	**64.0%**	**21.5%**	**27.6%**

Enhanced robustness to common corruptions: Finally, we evaluate our neuro-inspired framework for common corruptions suggested by (Hendrycks & Dietterich, 2018). These corruptions include noise injection, weather condition, common blur, and digital corruptions. Table 2 compares the accuracies obtained by our model with those for a standard model and an adversarially trained model. We see that our neuro-inspired design is effective in increasing robustness against these common corruptions. It is worth noting that, while adversarially trained models perform well against noise type corruptions, they perform drastically worse against more complex corruptions like fog and contrast (Machiraju et al., 2022; Kireev et al., 2021). In contrast, our HaH framework not only performs relatively well (performing substantially better than the standard model) for noise corruptions but also performs significantly better on more complex corruptions such as fog and contrast. Furthermore, the HaH-VGG16 outperforms both the standard model and adversarially trained model in terms of mean corruption accuracy. Given that such corruptions barely affect human vision, these results indicate that neuro-inspiration provides a valuable path towards general-purpose robustness against noise, adversarial perturbations, and common corruptions.

Ablation: Since we have different components in our HaH blocks, we explore the effectiveness of each component by doing an ablation study. Table 3 summarizes the contribution from each of the components. We see that all of the components (HaH training, divisive normalization, adaptive thresholding) play an important role in obtaining the reported gains in robustness to noise and adversarial attacks.

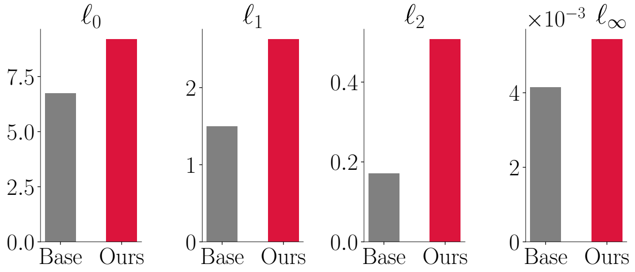

Figure 5: The average norm of minimum-norm adversarial attacks is higher for our model for all ℓ_p norms considered.

Furthermore, the number of HaH blocks plays a crucial role in obtaining robustness. Figure 6 shows the trade-off between clean accuracy and robust accuracy when the number of HaH blocks changes. Note that we successfully trained a model at most with 6 HaH blocks. Like earlier bio-inspired defenses, robustness through the HaH blocks also comes with a slight compromise on clean accuracy.

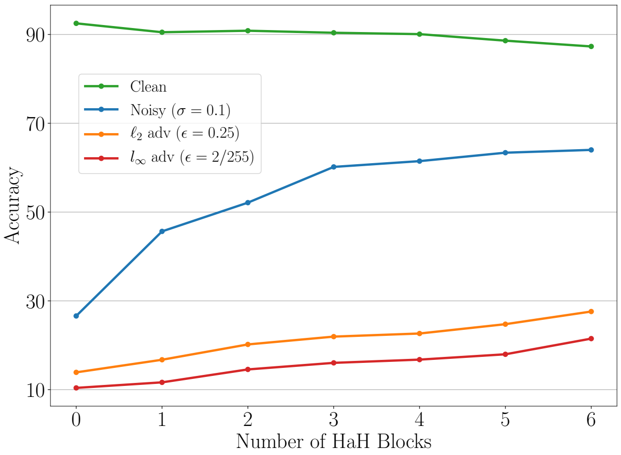

Figure 6: Ablation study for number of HaH blocks. Every additional HaH block contributes to the robustness of the model with a slight compromise on clean accuracy.

Corruptions →	Clean	Noise				Weather				Blur				Digital				Mean
Models ↓		Gauss.	Shot	Speckle	Impulse	Snow	Fog	Frost	Bright.	Defocus	Gauss.	Motion	Zoom	Contrast	Elastic	Pixelate	Spatter	of all
Standard	**92.5**	32.4	40.0	45.5	27.5	72.9	**64.5**	61.6	**87.4**	45.5	34.8	59.7	58.9	23.0	**74.8**	51.0	68.6	53.0
Adv(8/255)	78.7	**74.2**	**74.4**	**73.4**	**62.9**	62.3	29.7	59.0	60.4	**69.8**	**67.5**	**67.2**	**72.0**	18.0	72.5	**75.4**	71.5	63.1
HaH (Ours)	87.3	64.7	63.9	61.2	50.2	**74.4**	63.3	**73.3**	83.3	65.9	59.9	65.8	69.5	**76.3**	73.8	62.1	**76.3**	**67.7**

Table 2: Common corruption accuracies across different models. While standard and adversarially trained models are VGG16, HaH (ours) uses the aforementioned modified version of VGG16. Adversarially trained models perform poorly on fog and contrast corruptions while excelling on high-frequency corruptions like noise. On the other hand, the HaH framework consistently improves the robustness against all sorts of corruptions. Bright. stands for brightness, Gauss. stands for Gaussian, Elastic stands for elastic transformation

Table 3: Accuracies for ablation study.

	Clean	Noisy ($\sigma = 0.1$)	Adv (ℓ_∞) ($\epsilon = 2/255$)	Adv (ℓ_2) ($\epsilon = 0.25$)
All included	87.3%	**64.0%**	**21.5%**	**27.6%**
No HaH loss	89.7%	50.4%	8.8%	11.7%
Batch norm instead of divisive norm	**90.4%**	46.7%	12.3%	17.4%
No thresholding	89.9%	37.5%	3.7%	2.5%

4. Conclusion

Our preliminary results demonstrate the promise of enhancing the end-to-end training paradigm in DNNs with layer-wise costs in order to the features extracted by intermediate layers. In particular, our neuro-inspired approach to neuronal competition during training and inference demonstrably results in sparser, stronger activations and robustness against noise, common corruptions and adversarial perturbations than baseline models. Indeed, based our experiments with the CIFAR10-C (common corruptions) dataset, the robustness provided by our approach, trained in these preliminary results without any augmentation, appears to be more general-purpose than that obtained by adversarial training. We note that recent work on bio-inspired adversarial defenses appears to yield similar observations (Machiraju et al., 2022).

We hope that these results motivate a systematic inquiry into enhancing end-to-end training with layer-wise cost functions for a variety of architectures, training techniques (including unsupervised and semi-supervised learning, and data augmentation) and applications. In particular, for robust machine learning, a natural next step is to explore combination of data augmentation strategies (including adversarial training) with HaH architectures.

Acknowledgment

This work was supported in part by the Army Research Office under grant W911NF-19-1-0053, and by the National Science Foundation under grants CIF-1909320 and CIF-2224263.

References

Akkaya, I., Andrychowicz, M., Chociej, M., Litwin, M., McGrew, B., Petron, A., Paino, A., Plappert, M., Powell, G., Ribas, R., et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

Amato, G., Carrara, F., Falchi, F., Gennaro, C., and Lagani, G. Hebbian learning meets deep convolutional neural networks. In Ricci, E., Rota Bulò, S., Snoek, C., Lanz, O., Messelodi, S., and Sebe, N. (eds.), *Image Analysis and Processing – ICIAP 2019*, pp. 324–334, Cham, 2019. Springer International Publishing. ISBN 978-3-030-30642-7.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Burg, M. F., Cadena, S. A., Denfield, G. H., Walker, E. Y., Tolias, A. S., Bethge, M., and Ecker, A. S. Learning divisive normalization in primary visual cortex. *PLOS Computational Biology*, 17(6):e1009028, 2021.

Carandini, M. and Heeger, D. J. Normalization as a canonical neural computation. *Nature Reviews Neuroscience*, 13(1):51–62, 2012.

Croce, F., Andriushchenko, M., Sehwag, V., Debenedetti, E., Flammarion, N., Chiang, M., Mittal, P., and Hein, M. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

Dapello, J., Marques, T., Schrimpf, M., Geiger, F., Cox, D., and DiCarlo, J. Simulating a primary visual cortex at the front of cnns improves robustness to image perturbations. *bioRxiv*, 2020. doi: 10.1101/2020.06.16.154542. URL https://www.biorxiv.org/content/early/2020/06/17/2020.06.16.154542.

Fukushima, K., Miyake, S., and Ito, T. Neocognitron: A neural network model for a mechanism of visual pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(5):826–834, 1983. doi: 10.1109/TSMC.1983.6313076.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Hendrycks, D. and Dietterich, T. G. Benchmarking neural network robustness to common corruptions and surface variations. *arXiv preprint arXiv:1807.01697*, 2018.

Hoyer, P. O. Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research*, 5 (9), 2004.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kireev, K., Andriushchenko, M., and Flammarion, N. On the effectiveness of adversarial training against common corruptions. *arXiv preprint arXiv:2103.02325*, 2021.

Li, Z., Brendel, W., Walker, E., Cobos, E., Muhammad, T., Reimer, J., Bethge, M., Sinz, F., Pitkow, Z., and Tolias, A. Learning from brains how to regularize machines. *Advances in neural information processing systems*, 32, 2019.

Machiraju, H., Choung, O.-H., Herzog, M. H., and Frossard, P. Empirical advocacy of bio-inspired models for robust image recognition. *arXiv preprint arXiv:2205.09037*, 2022.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

Olshausen, B. A. and Field, D. J. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision research*, 37(23), 1997.

Pintor, M., Roli, F., Brendel, W., and Biggio, B. Fast minimum-norm adversarial attacks through adaptive norm constraints. *Advances in Neural Information Processing Systems*, 34, 2021.

Ren, M., Liao, R., Urtasun, R., Sinz, F. H., and Zemel, R. S. Normalizing the normalizers: Comparing and extending network normalization schemes. *arXiv preprint arXiv:1611.04520*, 2016.

Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Žídek, A., Nelson, A. W., Bridgland, A., et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792): 706–710, 2020.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. Theoretically principled trade-off between robustness and accuracy. *arXiv:1901.08573*, 2019.