INVESTIGATING ROBUSTNESS OF BIOLOGICAL VS. BACKPROP BASED LEARNING

Yanpeng Zhou[1], Maosen Wang[1], Manas Gupta[2], Arulmurugan Ambikapathi[2,3], Ponnuthurai Nagaratnam Suganthan[1], Savitha Ramasamy[2,3]

[1]School of Electrical and Electronic Engineering, Nanyang Technological University,Singapore
[2]Institute for Infocomm Research (I^{2}R), A*STAR, Singapore
[3]Artificial Intelligence, Analytics And Informatics (AI^3), A*STAR, Singapore

ABSTRACT

Robustness of learning algorithms remains an important problem to be solved from both the perspective of adversarial attacks and improving generalization. In this work, we investigate the robustness of biologically inspired Hebbian learning algorithm in depth. We find that Hebbian learning based algorithms outperform conventional learning algorithms like CNNs by a huge margin of upto 18% on the CIFAR-10 dataset under the addition of noise. We highlight that an important reason for this is the underlying representations that are being learnt by the learning algorithms. Specifically, we find that the Hebbian method learns the most robust representations compared to other methods that helps it to generalize better. We also conduct ablations on the Hebbian network and showcase that robustness of the model drops by upto 16% on the CIFAR-10 dataset if the representation capacity of the network is deteriorated. Hence, we find that the representations learnt play an important role in the resultant robustness of the models. We conduct experiments on multiple datasets and show that the results hold on all the datasets and at various noise levels.

Index Terms— Robustness, Hebbian learning algorithms, Representational learning, Biological plausible learning

1. INTRODUCTION

Robustness of neural networks is gaining increasing importance recently due to increasing number of adversarial attacks on neural networks [1]. Active research is being done to propose methods to increase robustness and give theoretical guarantees on the degrees of robustness of neural networks [2]. However, an important element has not been explored fully with regards to robustness of neural networks. By default, most of the neural networks today are trained using Backpropagation [3] and therefore, backprop is considered the default training technique. However, we question this assumption and study the robustness of backprop in comparison with other training techniques.

In particular, backprop is not considered very biologically plausible. We investigate this discrepancy and seek to study the performance and robustness of networks trained with backprop, against those trained with biological learning methods. We conduct a detailed range of experiments comparing biological systems like Hebbian learning to backprop based networks like CNNs. We find that Hebbian learning is surprisingly much more robust than CNNs. We note that when performance crashes in CNNs due to noise, Hebbian learning can still perform very well. Furthermore, we isolate a key factor that underlies these results. We note that the representations learnt by these two systems are quite different and on inspection, Hebbian networks learn much more robust representations compared to CNNs. We present detailed experiments on the MNIST and CIFAR-10 datasets along with additional ablations on Hebbian learning to provide evidence for the above.

2. RELATED WORK

Hebbian learning has developed from research in neuroscience and has been actively applied to neural networks for a long time [4, 5]. It is used in many fields, including continual learning [6, 7], biology [8, 6], and various real-world applications [9, 10]. Due to its biological and theoretical foundations, many works use it as an alternative to traditional backprop [11, 12], and explore the possibility of Hebbian learning as a viable alternative to backprop. Krotov and Hopfield [12] introduced some new rules to make hebbian learning competitive versus backprop. Amato et al. [13] extended the Hebbian framework to multiple layers to form deep architectures. To make the framework simpler and faster, Gupta et al. [14] focused on reducing algorithmic complexity while maintaining performance. Compared with other recent biologically plausible methods [15, 16], Hebbian learning is completely unsupervised and does not need any error or feedback signal to be relayed back. It is only dependent on the input and output of a neuron and thereby, learns very differently compared to feedback based methods.

Some early efforts were made to understand the robust-

ness of Hebbian based methods. However, these studies either utilized very simple networks and datasets [17, 18] or centered narrowly around certain application domains e.g., signal processing [19], biology [8], etc. Pedersen [20] proposed to increase the generalization ability of the model by reducing the number of rules, while improving the robustness of the model. However, there is a lack of recent literature investigating the robustness of biological learning algorithms in depth. In this work, we bridge this gap and also extend the analysis by highlighting the relationship between representations learnt by the neurons and the corresponding robustness performance of the models.

Algorithm 1 Algorithm for training HLF

```
# Initialize the Hebbian weight matrix W_1
W_1 ~ N(−0.5, 1)
# Initialize the classification layer matrix W_2
W_2 ← Kaiming initialization
epochs ← 20
batch_size ← 100
x, target ← load data
# Train the Hebbian layer
while epoch < epochs do
    Z = XW_1

    # Winner-takes-all
    pos ← index of neuron with max abs(Z) in each instance
    for i in range (0, len(pos)) do
        Act_new[i, pos[i]] = Activation_value
    end for
    ΔW = X^T Act_new

    # Threshold
    for i in range (0, len(pos) do
        T[:, pos[i]] += W[:, pos[i]]
    end for
    ΔW = ΔW − T
    ΔW = ΔW / max(ΔW)
    W_1 = W_1 + η · ΔW
end while

# Train the SGD layer
while epoch < epochs do
    Train W_2 using Adadelta
end while
```

3. APPROACH

To explore the relationship between the representations learnt by the neural network to its resultant robustness, we use three methods to benchmark our results - Hebbian learning, Oja's rule [21], and Convolutional Neural Networks (CNNs). We detail each method below.

Vanilla Hebbian learning The vanilla Hebbian rule states that a weight changes in accordance to the activations in the two neurons that it connects, such that:

$$\Delta w_{ij} = \eta x_i z_j \tag{1}$$

where Δw_{ij} is the weight connecting neuron i and neuron j, x_i and z_j are the input and output of neuron j respectively, and η is the learning rate. The magnitude of the weight update is proportional to the product of the input and output values. The higher the product of input and output, the stronger the connection between neurons, and vice-versa.

Hebbian Learning framework (HLF) The Hebbian framework implemented by Krotov and Hopfield was one of the first to achieve competitive performance against Backprop. It was able to learn the inputs very robustly. For our comparison, we use their framework along with some modifications. We modify their Winner-Takes-All (WTA) strategy. We set the activation value of the winning neuron to a pre-determined value to aid in the stable learning of representations. The full algorithm pseudo-code is described in Algorithm 1, where $Activation_value$ is the pre-determined value ,Act_{new} is the new output of hebbian layer after WTA and T in column 21 is the Threshold.

To provide an ablation for the learning capacity of the HLF, we design an ablated variant called HLF (5%). The aim is to have only 5% of the learning capacity of the original network and see if the network can still be as robust. To achieve the reduced capacity, we use the fully trained HLF network and then randomly reinitialize 95% of the neurons such that they forgot the learnt information. This essentially makes only 5% of the neurons retain their learnt patterns.

Modified Oja's rule (MOR) Oja's rule is a modification of the standard Hebbian rule [21]. It solves the problem of infinite growth of weights under the vanilla Hebbian rule by introducing a 'forgetting' or weight reduction term. The value of the forgetting term is not only proportional to the value of the weight, but also proportional to the square of the neuron output, as follows:

$$\Delta w_{ij} = \eta z_j (x_i - z_j w_{ij}) \tag{2}$$

where w_{ij} is the weight connecting neuron i in the input layer and neuron j in hidden layer, x_i and z_j are the input and the output of neuron j and η is the learning rate. In addition, we store the previous weight as a threshold, and linearly combine the previous weight with the current weight at each training iteration to obtain a new threshold, as shown in the equation:

$$T_{ij}^t = \begin{cases} x_i^t z_j^t & \text{if } t = 1 \\ (1-\lambda) T_{ij}^{t-1} + \lambda x_i^{t-1} z_j^t & \text{otherwise} \end{cases} \tag{3}$$

where T_{ij}^{t} refers to the threshold for neuron j in iteration t and λ is a hyper-parameter to adjust the impact of previous thresholds on the current threshold. Intuitively, this means that the weight update is not only limited by the forgotten items, but also affected by past updates. **Convolutional Neural Network** Convolutional neural networks (CNN) are the cornerstone of computer vision today. The convolutional layer is an important part of the convolutional neural network. It extracts features from the input data [22], as follows:

$$Z^{l+1}(i,j) = [Z^{l} \otimes w^{l+1}](i,j) + b \tag{4}$$

where $\otimes$ means convolution calculation, b is the bias, Z^{l},Z^{l+1} and w^{l+1} refer to the input, output and weights in layer $l+1$.

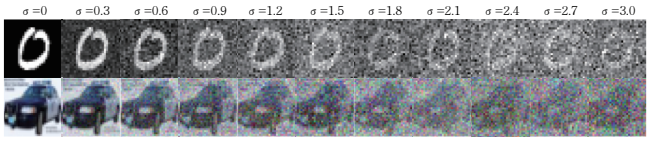


Fig. 1. Test set images on the MNIST and CIFAR-10 datasets, after the addition of Gaussian noise with mean 0 and standard deviation ranging from 0 to 3.

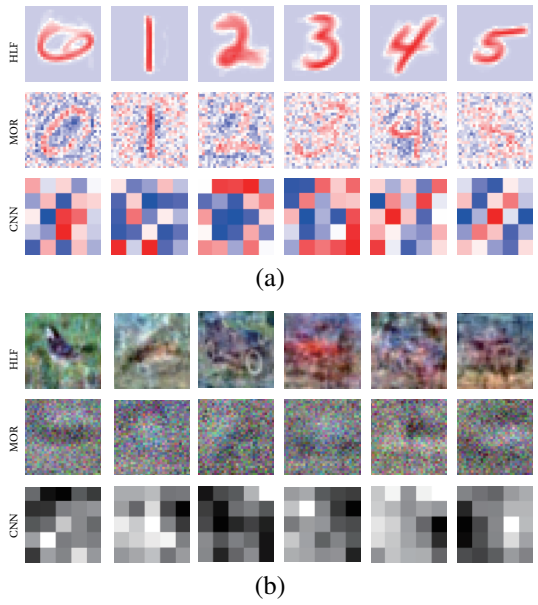
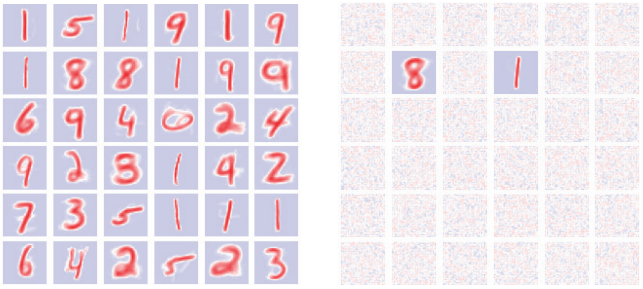


Fig. 2. (a) Representations learnt by the different methods on MNIST dataset. Red color indicates positive weights while blue indicates negative weights. (b) Representations learnt on the CIFAR-10 dataset (RGB color scale). HLF learns the most robust representations.



(a) HLF (b) HLF (5%)

Fig. 3. (a) Representations learnt by the vanilla HLF network utilizing all of the neurons (b) Representations learnt by a HLF network with only 5% neurons operational.

4. EXPERIMENTS

We perform a series of experiments on the different learning methods mentioned in section 3. We first perform the experiments on the MNIST dataset. For HLF, we use a two-layered network with the first layer learning representations through Hebbian learning and the second layer performing classification using SGD. The network structure of HLF (5%) is exactly the same as HLF. In MOR, we built a two-layer neural network consisting of a modified Oja rule layer and an SGD layer. The modified Oja rule layer learns representations, and the SGD layer implements classification. For CNN, we use two convolutional layers to extract features, and use a SGD layer to complete the classification. We train all the models for 50 epochs on the train set and then perform inference with the noise added to the test set (Fig. 1).

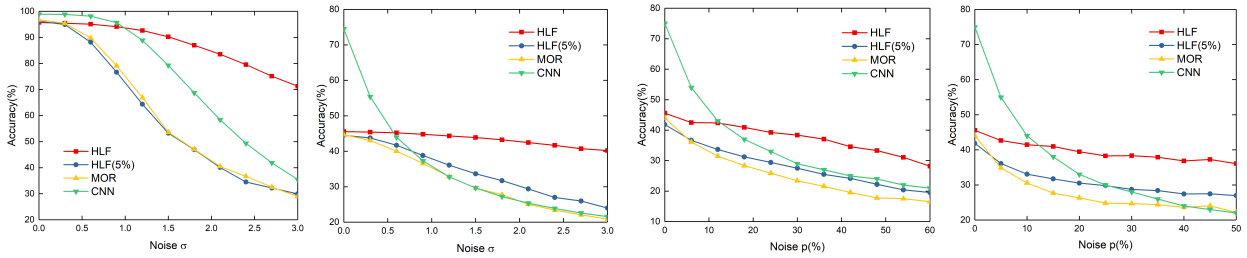
Table 1. Results with and without noise on the various methods on the MNIST dataset. HLF is the most robust and outperforms all the other methods by a huge margin under the noise.

Method	Acc. without noise	Acc. with noise (σ=3.0)
MOR	96.77%	28.97%
CNN	98.87%	35.46%
HLF (5%)	96.18%	29.94%
HLF	95.70%	**71.26%**

As can be seen from table 1, HLF outperforms all the other methods when noise is added and proves to be the most robust. It beats the CNN network by a 36% margin. We hypothesize that the reason why this happens is linked to the representations learnt by the different methods. Referring to Fig. 2(a), we can see that HLF learns the most robust patterns, clearly learning the entire digit. This is in contrast to the CNN network which learns less clear patterns. An im-

Table 2. Results with and without noise on the various methods on the CIFAR-10 dataset. HLF is the most robust and outperforms all the other methods by a huge margin under the noise.

Method	Acc. without noise	Acc. with Gaussian Noise (σ=3.0)	Acc. with Salt and Pepper Noise (p=60%)	Acc. with Rand noise (p=50%)
MOR	44.67%	20.88%	16.54%	22.17%
CNN	74.51%	21.56%	21.26%	22.00%
HLF (5%)	44.49%	23.98%	19.56%	26.98%
HLF	45.56%	**40.21**%	**28.21**%	**36.06**%



(a) (b) (c) (d)

Fig. 4. Accuracy curves for different methods under various noise types. (a) Gaussian Noise on MNIST, (b) Gaussian Noise on CIFAR-10, (c) Salt and Pepper Noise on CIFAR-10, (d) Random Noise on CIFAR-10. HLF proves to be the most robust under high noise ratios for both MNIST and CIFAR-10.

portant difference to note here is the learning objectives of the two methods. HLF learns in an unsupervised manner and therefore, learns the entire feature space of the input data i.e. the entire digit. However, the CNN is trained to minimize the classification loss and thereby learns discriminative features that help distinguish between the different classes. It has no requirement to learn the entire feature space and hence, learns sparser features that look unclear.

Another thing to note is the HLF compared to the HLF (5%). Continuing from our hypothesis that the robustness is linked to the representations learnt, we conduct an ablation to see what happens when we inactivate a large proportion of the neurons. Thus, in HLF (5%), only 5% of the neurons are retained (see section 3 for details). We compare the representations learnt by HLF and HLF(5%) in Fig. 3. We find that our hypothesis indeed holds, and HLF (5%) is not robust to noise achieving only 30% accuracy under noise. Thus, learning good representations helps in achieving good robustness. We also test the methods at different noise ratios and plot the accuracy curves (Fig. 4(a)). HLF outperforms the CNN and other methods at all noise ratios where $\sigma > 1$.

We then experiment on a tougher dataset i.e. CIFAR-10 (table 2). We find that HLF again outperforms the other methods under noise by a huge margin. We find that as in the case of MNIST, HLF learns the most robust representations compared to the other methods (Fig. 2(b)). In addition, we introduce two more noise types to our experiments i.e., Salt and Pepper Noise (Fig. 4(c)) and Random Noise (Fig. 4(d)) on the CIFAR-10 dataset. In random noise, we randomly select some pixels (p% of all pixels) as white noise points, which satisfy the uniform distribution. In salt and pepper noise, we also randomly select some pixels ($\frac{p}{2}$% of all pixels) as white noise (salt noise) points and the remaining $\frac{p}{2}$% of all pixels as black noise (pepper noise) points. As can be seen from Fig. 4, HLF can maintain higher robustness compared to other methods for all types of noise.

5. CONCLUSION

We have presented a systematic analysis of the robustness of various learning algorithms under the influence of noise. In particular, we showed that Hebbian based biological learning algorithms learn the most robust representations compared to other back propagation based learning algorithms like CNNs. This indicates that the biological algorithm HLF is more robust than CNNs and other methods while handling the influence of noise, which is inevitable in reality and is the major limitation of other existing learning algorithms. Even at high noise levels, when the performance of other algorithms deteriorates rapidly, HLF is still able to sustain its classification performance and achieves 18% higher accuracy than CNN on the CIFAR-10 dataset. We believe that this work fosters new areas for researchers to improve robustness of models by understanding the quality of representations being learnt by the underlying learning algorithm.

6. REFERENCES

[1] Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu, "Adversarial attacks and defenses in deep learning," *Engineering*, vol. 6, no. 3, pp. 346–360, 2020.

[2] Ravi Mangal, Aditya V. Nori, and Alessandro Orso, "Robustness of neural networks: A probabilistic and practical approach," *ICSE-NIER 2019*, 2019.

[3] Jun Haeng Lee, Tobi Delbruck, and Michael Pfeiffer, "Training deep spiking neural networks using backpropagation," *Frontiers in neuroscience*, vol. 10, pp. 508, 2016.

[4] E Awh, L Anllo-Vento, SA Hillyard, JS Baizer, LG Ungerleider, and R Desimone, "Allport, da (1985). distributed memory, modular subsystems and dysphasia. in sk newman & r. epstein (eds.), current perspectives in dysphasia (pp. 32–60). edinburgh: Churchill livingstone. anderson, jr (1976). language, memory, and," *science*, vol. 12, no. 5, pp. 840–847, 1991.

[5] A Harry Klopf, *Brain function and adaptive systems: a heterostatic theory*, Number 133. Air Force Cambridge Research Laboratories, Air Force Systems Command, United . . . , 1972.

[6] Lukasz Kuśmierz, Takuya Isomura, and Taro Toyoizumi, "Learning with three factors: modulating hebbian plasticity with errors," *Current opinion in neurobiology*, vol. 46, pp. 170–177, 2017.

[7] Vithursan Thangarasa, Thomas Miconi, and Graham W Taylor, "Enabling continual learning with differentiable hebbian plasticity," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[8] Jannis Born, Juan M Galeazzi, and Simon M Stringer, "Hebbian learning of hand-centred representations in a hierarchical neural network model of the primate visual system," *PloS one*, vol. 12, no. 5, pp. e0178304, 2017.

[9] Rohit Abraham John, Fucai Liu, Nguyen Anh Chien, Mohit R Kulkarni, Chao Zhu, Qundong Fu, Arindam Basu, Zheng Liu, and Nripan Mathews, "Synergistic gating of electro-iono-photoactive 2d chalcogenide neuristors: coexistence of hebbian and homeostatic synaptic metaplasticity," *Advanced Materials*, vol. 30, no. 25, pp. 1800220, 2018.

[10] Mirko Hansen, Finn Zahari, Hermann Kohlstedt, and Martin Ziegler, "Unsupervised hebbian learning experimentally realized with analogue memristive crossbar arrays," *Scientific reports*, vol. 8, no. 1, pp. 1–10, 2018.

[11] Noureddine Kermiche, "Contrastive hebbian feedforward learning for neural networks," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 6, pp. 2118–2128, 2019.

[12] Dmitry Krotov and John J Hopfield, "Unsupervised learning by competing hidden units," *Proceedings of the National Academy of Sciences*, vol. 116, no. 16, pp. 7723–7731, 2019.

[13] Giuseppe Amato, Fabio Carrara, Fabrizio Falchi, Claudio Gennaro, and Gabriele Lagani, "Hebbian learning meets deep convolutional neural networks," in *International Conference on Image Analysis and Processing*. Springer, 2019, pp. 324–334.

[14] Manas Gupta, ArulMurugan Ambikapathi, and Savitha Ramasamy, "Hebbnet: A simplified hebbian learning framework to do biologically plausible learning," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3115–3119.

[15] Timothy P Lillicrap, Daniel Cownden, Douglas B Tweed, and Colin J Akerman, "Random synaptic feedback weights support error backpropagation for deep learning," *Nature communications*, vol. 7, no. 1, pp. 1–10, 2016.

[16] Nikolay Manchev and Michael W Spratling, "Target propagation in recurrent neural networks.," *J. Mach. Learn. Res.*, vol. 21, pp. 7–1, 2020.

[17] Erkki Oja and Liuyue Wang, "Neural fitting: robustness by anti-hebbian learning," *Neurocomputing*, vol. 12, no. 2-3, pp. 155–170, 1996.

[18] Tibor Fomin and A Lórincz, "Robustness of hebbian and anti-hebbian learning," in *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*. IEEE, 1994, vol. 2, pp. 731–735.

[19] Bing Lu, Alireza Dibazar, and Theodore W Berger, "Nonlinear hebbian learning for noise-independent vehicle sound recognition," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, 2008, pp. 1336–1343.

[20] Joachim Winther Pedersen and Sebastian Risi, "Evolving and merging hebbian learning rules: increasing generalization by decreasing the number of rules," *arXiv preprint arXiv:2104.07959*, 2021.

[21] Erkki Oja, "The nonlinear pca learning rule in independent component analysis," *Neurocomputing*, vol. 17, no. 1, pp. 25–45, 1997.

[22] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep learning*, MIT press, 2016.