Unsupervised Features Extracted using Winner-Take-All Mechanism Lead to Robust Image Classification

Devdhar Patel
College of Computer and Information Sciences
University of Massachusetts Amherst
Amherst, MA, USA
devdharpatel@cs.umass.edu

Robert Kozma
College of Computer and Information Sciences
University of Massachusetts Amherst, Amherst, MA, USA
& Department of Mathematical Sciences
University of Memphis, Memphis, TN, USA
rkozma@cs.umass.edu

***Abstract*—Leading mainstream image processing approaches produce excellent performance using convolutional neural networks trained by backpropagation (BP) learning rules. Unsupervised learning approaches have been popular due to their biological significance, though they typically underperform compared to BP results. In this work, we demonstrate that features extracted in an unsupervised manner using the biologically inspired Hebbian learning rule in a winner-take-all setting, perform competitively with BP on the image classification task. The convolutional filters learned by Hebbian rule are smoother than filters learned using BP. The quality of the two training approaches is compared based on metrics such as the speed of training and classification accuracy. We demonstrate that the extracted features of unsupervised learning are more robust to noise as compared to BP.**

***Index Terms*—Unsupervised learning, CIFAR-10, Hebbian, Winner-take-all, Robust**

I. Introduction

In recent years, deep neural networks trained using backpropagation have achieved great success in computer vision tasks like image classification, segmentation, image generation, and even playing video games. However, there are still significant concerns regarding their performance. Deep networks trained using backpropagation are known to be vulnerable to adversarial attacks [1]–[3]. Also, the features learned by deep learning do not generalize well to minor changes in the task or different tasks [4], [5].

The biological brain, on the other hand, is much more robust to noisy input, adversarial examples. The brain is also capable of transfer learning and can transfer relevant knowledge from one task to another. In this work, we show that biologically inspired local learning rules can extract features from images in an unsupervised manner. We examine the differences between a network trained using backpropagation and a network trained using local learning. We demonstrate that the network trained using unsupervised feature extraction and local learning has comparable performance to a neural network trained using backpropagation on the CIFAR-10 dataset. Furthermore, we compare the two networks on various metrics like speed and training data. Finally, we show that the local learning networks are more robust to random noise and image occlusion and thus more generalized without any significant difference in the performance on the original task.

II. Background and Related work

Biologically inspired learning is an active area of interest; most of the work in this area can be categorized into three broad categories: Biologically inspired unsupervised learning, biologically inspired supervised learning, and reinforcement-based learning rules. Since research in this area takes its inspiration from biology and neuroscience, often, the research is conducted on spiking neural networks (SNN) [6]–[13]. However, in our work, we only show preliminary results on spiking neural networks. Our work primarily focuses on artificial neurons.

Much recent work has focused on unsupervised feature learning on spiking neural networks. It is essential to note the significance of the differences in these works. [14] introduced a method of training SNNs using convolutions over time instead of spatial convolutions. Their work leverages the vital property of SNN of encoding time in spikes. [15] uses a similar approach to the one proposed in this work. However, they showed that their approach gives competitive results on images of faces. Their work also shows that STDP based unsupervised feature learning can be used to train a classifier using fewer labeled training data. We test the reverse of this phenomenon in our work and find that training a classifier on features extracted from a larger dataset does not provide any advantage to the network's accuracy. [16] demonstrated the best accuracy so far on MNIST (99.28%). However, their work combines backpropagation on SNN with unsupervised pre-training using STDP. [17] used Hebbian and anti-Hebbian learning rules to achieve unsupervised feature learning on artificial neural networks (ANNs). They train a non-linear classifier to achieve an accuracy of 98.54% on MNIST. Furthermore, [17] is one of the few works to report an unsupervised accuracy on CIFAR-10 (50.75%). [18] further improved on this result by using the local learning networks in a convolutional manner. The convolutional filters extract

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

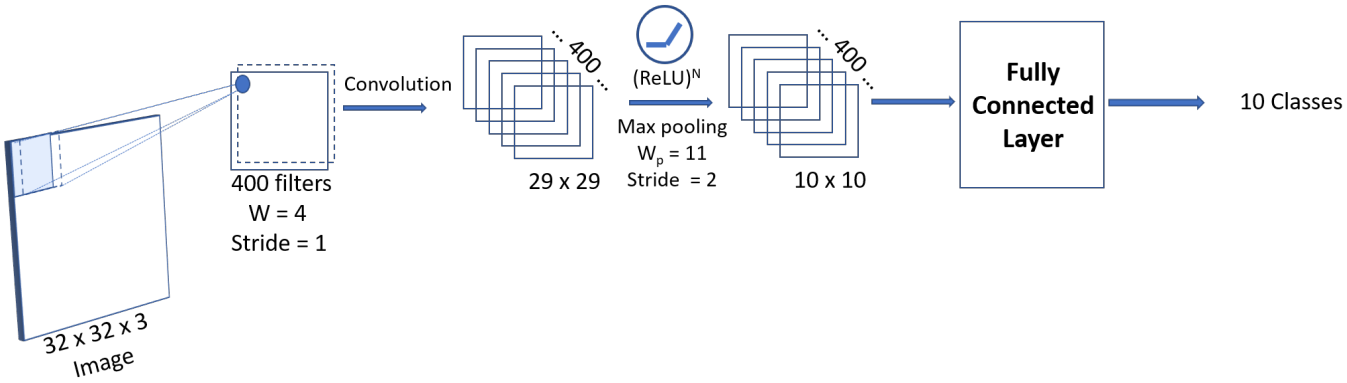


Fig. 1: Simple neural network containing 1 convolution layer with 400 filters, kernel size of 4 and stride of 1. The convolution layer is followed by ReLU non-linearity. The steepness of the ReLU is controlled by the factor N. For our experiments, $N = 1$ for the backpropagation network and $N = 40$ for the Hebbian learning network. The a max pooling layer of kernel size 11 and stride of 2.

position invariant features and further improve the CIFAR-10 accuracy. In this work, we use the network similar to [18]'s work to test our hypothesis on the properties of local Hebbian learning.

III. Experiment design and Results

While biologically inspired learning is an area of active research, most of the work has primarily used the MNIST dataset as the testbed. While unsupervised local learning has performed very well on MNIST [6], [7], [9], the same approach performs very poorly on more complex datasets like CIFAR-10 and ImageNet. One of the reasons for this is that the MNIST dataset is a straightforward dataset which can be classified by filters that look like the classes itself [6]. However, more complex datasets require more generalized features that are not class-specific. For this purpose, using convolutional filters that extract position invariant features is essential. While [9] showed an improved performance using fixed position filters that extract features from only one area of the images, since the filters did not convolve, it did not perform well on more complex datasets.

Similarly, [17] showed poor performance on CIFAR-10 without convolution; however, when the same approach is used with convolution filters, [18] achieve performance comparable to backpropagation on CIFAR-10. Since its architecture can be replicated for backpropagation, it is easy to compare its performance. For this reason, we choose the convolution network with Hebbian-anti-Hebbian learning to test several hypotheses about local learning. From our frontier exploration, we came across many common claims about local learning that are tested on the MNIST dataset. However, it remains to be seen if the same results hold for more complex datasets like CIFAR-10. In this work, we test the following hypothesis:

1) **The features extracted using this method are better than those found using backpropagation.** While [18] showed comparable performance on local learning network in terms of accuracy, accuracy does not fully characterize the performance of a network. For example, many neural networks surpass human performance [19]. However, they require large amounts of repeated updates and multiple epochs to reach that accuracy. Thus we measure the accuracy, training speed, and robustness to noise of the networks to characterize the performance of the networks.
2) **Features extracted from a larger dataset would perform better on a smaller labeled dataset** Since the first layer of the network is trained in an unsupervised manner, the features learned can potentially be extracted from a larger dataset of unlabeled data. Unsupervised feature extraction is useful, considering the time and cost of labeling images. Many research works assume that features extracted in an unsupervised manner from a larger dataset would give higher accuracy. However, it is seldom tested. We design an experiment to test this hypothesis.
3) **The features extracted using this method are more robust and generalized due to their smoothness.** [18] compared the features extracted using Hebbian learning with the features from deep networks like AlexNet [20] and noted that the features extracted from Hebbian learning appeared to be smoother than the ones in AlexNet. The smooth filters were further compared to biological features. It seems logical that smooth filters would be more generalized than filters with sharp edges. However, the smoothness of the filters is not measured in previous work, and it has not been compared to a network with similar architecture. We devise a measure for smoothness and compare the features of local learning and backpropagation with similar architecture. Furthermore, we test the generalization and robustness of the networks to the noise by testing their performance on noisy input.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

A. Network Architecture

The focus of this research work is on the properties of local learning. Therefore, we do not search for network architecture and use a simple network architecture. Figure 1 shows the network architecture. The CIFAR-10 images are of the dimension 32x32x3, which are the input to the network. The network contains a single convolution layer with 400 filters, kernel size 4, and stride of 1. The convolution filter is followed by a max-pooling layer of kernel size 11 and stride 2. The pooling layer is finally connected to a fully connected classifier.

For the backpropagation network, the convolution layer has the ReLU non-linearity. For the local learning network, the convolution filters are extracted using Hebbian-anti-Hebbian learning in an unsupervised manner from the training dataset. The learning rule is given as follows:

$$\Delta M_{\mu i} = \varepsilon \sum_{A \in \texttt{minibatch}} g\big[\texttt{Rank}\big(\sum_j M_{\mu j} v_j^A\big)\big] \times \big[v_i^A - \big(\sum_k M_{\mu k} v_k^A\big) M_{\mu i}\big] \quad (1)$$

Where ε is the learning rate. The activation function $g[]$ simulates winner-take-all learning where it is equal to 1 for the channel with the highest activation, equal to a small negative constant for the m largest activation (anti-Hebbian) and 0 otherwise. For our experiments, m is set to 2. v_i^A represents the patch of the image the filters are looking at where i represents the pixels for patch A. $M_{\mu i}$ is the weight matrix.

The convolution filters learned using the above learning rule are then substituted into the network and fixed. The final fully connected layer is then trained using backpropagation. Since only one layer is trained using backpropagation, the weights are only changed as a result of the inputting layer and the outputting layer. Thus it is local learning.

The Hebbian network has two major differences from the ANN:

1) The ReLU after the first convolution layer is a steep ReLU. That is, the output of the ReLU is $ReLU(x)^n$, where x is the output of the convolution layer. The power $n = 40$ for the experiments in this work.
2) Each convolution patch is normalized to be a unit vector for the Hebbian network. This normalization is termed as patch normalization by [18].

Without these two changes, the Hebbian network performs very poorly. The reason for this is because the weights trained by Hebbian learning converge to a unit vector. Thus, they cannot handle input that is not normalized. After patch normalization, the output of the convolution layer becomes a cosine similarity between the filters and the image patches. To improve their contrast, we need to use steep ReLU.

B. Training data and validation

For our experiments, we parameter search the learning rate using 10% of the training data as a validation set. For the rest of the experiments, we used the entire training data to train the networks and present our results on the test data.

C. Experiment 1: Learning Curve

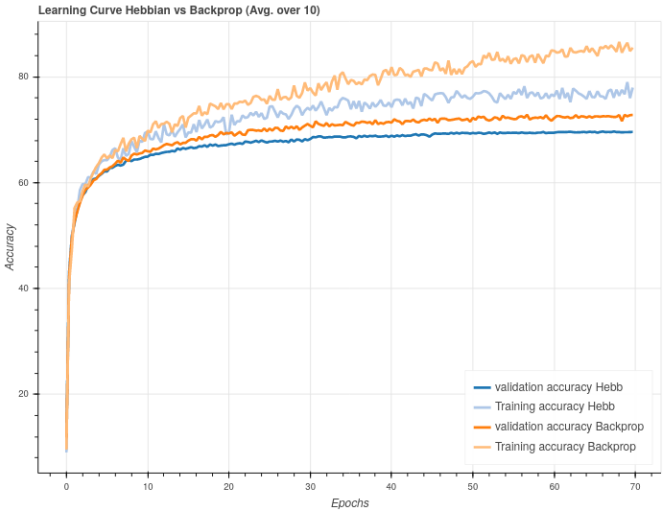


Fig. 2: Learning curve of the Back-propagation network vs. the Hebbian Network. Each network is trained for 70 epochs. Each learning curve is averaged over 10 different runs.

To compare the efficacy of the networks, we compare their learning curve. A better network would learn faster.

The learning curves are shown in Figure 2. Each network is trained for 70 epochs; however, for the Hebbian network, only the last layer is trained in a supervised manner. As we can see, both the networks have a similar learning curve, but the backpropagation-network has slightly higher accuracy than the Hebbian network. The accuracy is an expected result already stated in [18]'s work. The final test accuracy of the backpropagation-network is 72.631%. The final test accuracy of the Hebbian network is 69.463%.

From the learning curve, none of the networks learn significantly faster than the other. Therefore, we conclude that the learned filters do not affect the speed of training the classifier.

D. Experiment 2: Reduced training data

While the Hebbian network performs similarly to the BP network, it can still outperform the BP network when there are fewer labeled samples for supervised training. To test the second hypothesis, we trained the supervised part of the networks on partial training data of the CIFAR-10 dataset. Note that the unsupervised features are still extracted from the entire training dataset. Thus, the Hebbian network has an advantage since it extracts features in an unsupervised manner when labeled data is scarce. Figure 3 shows the test accuracy of the networks compared to the size of the training dataset. Each network is trained on different training data sizes and averaged over five runs. Surprisingly, both the networks perform well even with 5% of the training data. However, the Hebbian network does not outperform the backpropagation network with limited labeled data. This result is surprising and suggests that the final fully connected layer is responsible for much of the performance of the Hebbian network. From this

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

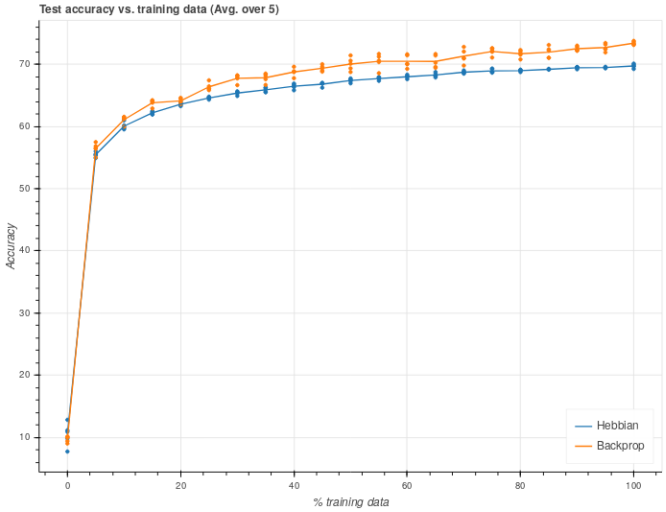


Fig. 3: Performance of the networks with respect to size of the training dataset.

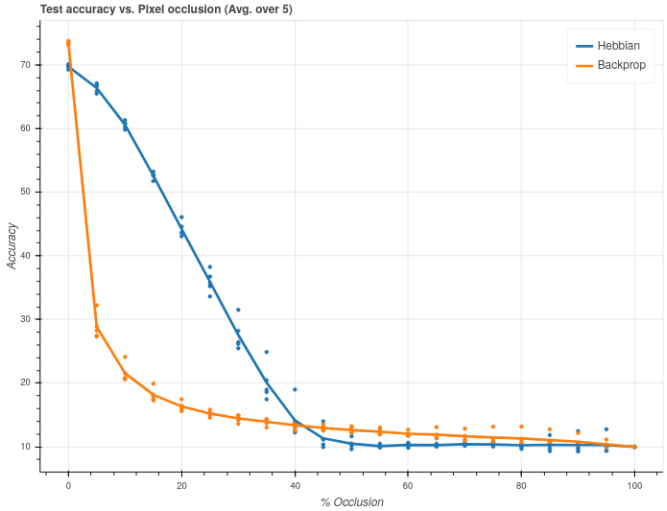


Fig. 4: Robustness of the networks with respect randomly removed pixels from the test data. The graph shows the accuracy of the networks with respect to the percentage of pixels removed. Both the plots are averaged over 5 trials. The individual trials are also plotted.

experiment, we can conclude that the Hebbian filters extracted from large amounts of unlabeled data do not contribute to better accuracy of the network. The final fully-connected classifier is responsible for the classification accuracy, and therefore, both the networks have similar performance in this experiment.

E. Experiment 3: Robustness and Generalization

To test the generalization performance of the filters, we test the accuracy of the networks with missing pixels from the test dataset. In this experiment, we removed a percentage of pixels (all three channels) from the test images. This test would help us gauge the generalization of the network and its robustness against noise and occlusion. Figure 4 shows the performance of the networks against the pixel occlusion. Each network was tested with varying percentages of pixels occluded, and the performance is averaged over five trials.

This test shows that the Hebbian network is significantly more robust than the backpropagation-network. The backpropagation-network suffers a drastic decrease in performance with just 5% of the pixels occluded while the Hebbian network is much more robust to the occlusion. While the accuracy of both the networks eventually falls, note that it is even tough for humans to classify the images with 40% of the pixels occluded.

This result is significant and suggests that while the final fully connected layer is essential for the classification accuracy, the extracted features are responsible for the robustness.

The second experiment tests the robustness of the networks when a square area of the image is occluded. Figure 5 shows the results of this experiment. In this experiment also, the Hebbian network outperforms the BP network. This result reinforces the conclusion that the Hebbian features are more robust and generalized than the features learned by backpropagation.

An interesting observation here is that the networks are much more robust to the square occlusion than the random pixels occlusion. The robustness could be due to the convolution and max-pooling layers mitigating the occlusion located in an area to some extent. Another interesting observation is that in both the cases, the performance of the Hebbian network degrades linearly as the percentage of the image occluded increases while the performance of the BP network degrades exponentially. This phenomenon needs further investigation and is left to future work.

IV. Filters and Smoothness

One apparent difference between the filters learned by Hebbian learning and the filters learned by backpropagation is that the filters learned by Hebbian network are smoother than those learned by backpropagation. Figure 6 shows the features extracted by the unsupervised learning and backpropagation. Note that the Hebbian filters contain color filters and color insensitive orientation filters. However, a lot of the filters are also color-sensitive orientation filters not mentioned in [18]. Also, note that 22 filters are unused during training and therefore are random.

In contrast, the filters learned by backpropagation look like random filters. There does not seem to be any pattern. The reason for this could be because the kernel size of 4 and 2 layered networks are too small to learn any visually discernible features.

To compare the smoothness of the filters, we design a measure that would represent the existence of sharp edges in the filter. The sharpness index is equal to the average L2-norm of each filter, with the same filter shifted by one pixel in each of the four directions. We define the sharpness-index

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

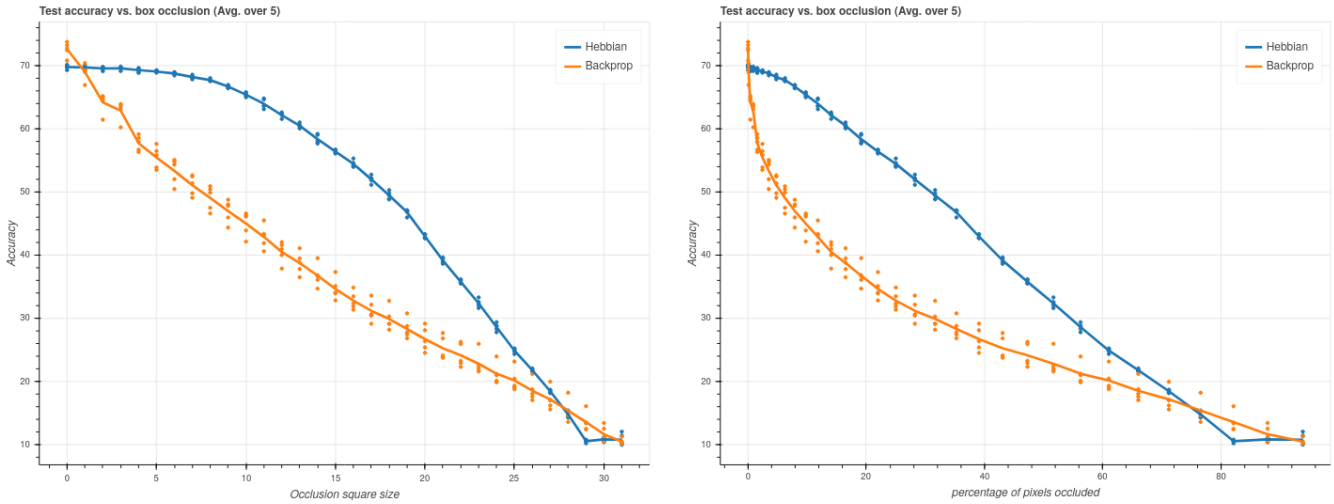
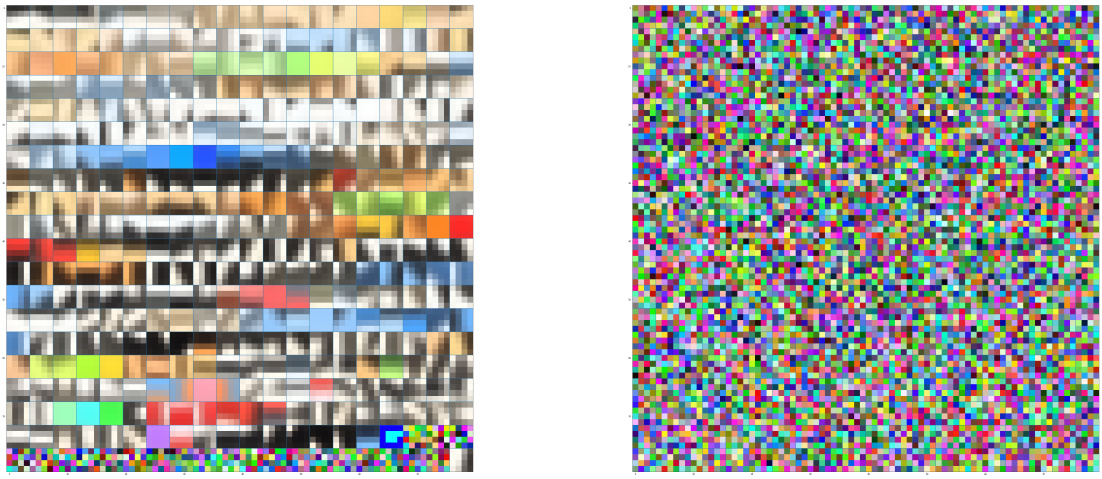


Fig. 5: Robustness of the networks with respect to a square area occluded from the test data. The graph shows the accuracy of the networks with respect to the size of the square (in pixels). For each size of square, the result is displayed as an average of 5 trials. The individual trials are also plotted. The plot on the left displays the size of the square occluded on the x-axis while the figure on the right displays the percentage of pixels occluded on the x-asis.



(a) Unsupervised learning.

(b) Back-propagation.

Fig. 6: Filters extracted from CIFAR-10 using different learning mechanisms.

as the average of this number for all 400 filters. Note that this method only captures the differences between adjacent pixels; thus, if a filter is smooth, it would have a lower sharpness index. In order to calculate the sharpness-index, it is crucial to normalize the filters before the calculation since the filters extracted from backpropagation and Hebbian learning have different magnitudes. Table I shows the sharpness-index of the two networks. The Hebbian filters are much smoother than the backpropagation filters. The smoothness could contribute to the Hebbian filters' robustness since they would experience a lesser change in their output as a change in the input due to their smoothness.

Network	Sharpness-index
Backpropagation	5.42
Hebbian	0.23

TABLE I: Smoothness-index of the filters learned by the two networks.

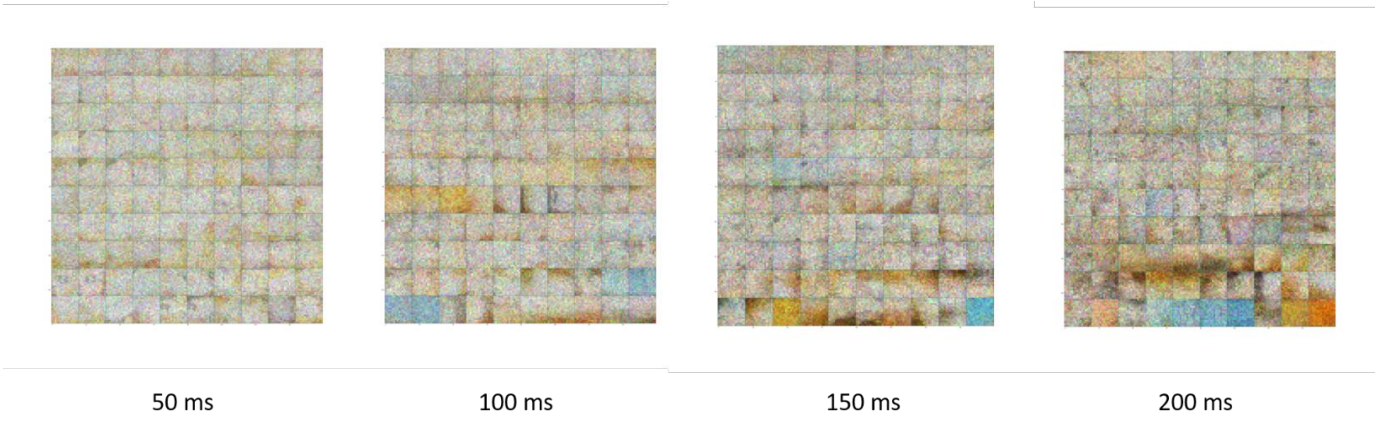


Fig. 7: Features extracted from a network with 100 neurons with varying runtime per input. Note the formation of color-sensitive and color-blind orientation-sensitive filters as we increase the runtime.

V. Spiking Neural Networks

Finally, we demonstrate that we can construct a similar network in Spiking Neural Networks (SNN). Spiking neural networks, while take longer to simulate, can be implemented on neuromorphic hardware to run much faster and be much more energy-efficient.

To implement a winner-take-all approach on SNN, we used a modified network architecture based on the work of [6]. [6] showed competitive performance on the MNIST dataset using this architecture. We note that our approach is similar to [9]. However, their approach used local connections rather than convolutional connections.

To demonstrate the capability of our approach, we present the results of our experiments on the Tiny ImageNet dataset. Note that the time to simulate spiking neural networks is longer by order of magnitudes than the time to simulate an artificial neural network. Therefore, we used limited data for our experiments. For this task, a toy dataset was created from the two classes of the Tiny ImageNet dataset. Each class contains 500 images of dimension 64x64x3 (pixel x pixel x RGB color). We used 450 images as training data and 50 images as testing data from each class. In order to facilitate learning from convolution patches, each image is split into multiple patches using convolution patches.

The network is exposed to each image for a fixed amount of time. We performed experiments with varying sizes of neural networks and varying amounts of the time window for the spiking activity. The kernel size of the convolution window was fixed to 16x16 with a stride of 2. After extracting the filter features, we trained a simple single-layer neural network classifier on the features in a supervised manner.

The filters learned using this method are shown in Fig. 7, and they can be used for the classification of images. During testing, we run the neural network on each convolution of images and use the sum of spikes of neurons as features for the classification. Using an SNN with 100 spiking units (neurons) and exposure time of 50 ms for each training image, we obtain training and testing accuracy of 81% and 79%, respectively. To test the transferability of these results for new data, we tested the filters obtained for two classes to classify the first three classes of the tiny ImageNet dataset, and we obtain 77% accuracy. This result suggests that the extracted filters are generalized and have transfer learning capability. More studies are in progress to scale up the results to more extensive data.

Some of the critical points to note in our results are:

1) As the exposure time increases, we see the formation of color-sensitive and color-blind orientation-sensitive filters as we increase the runtime. These results are similar to [18], and suggests that our model behaves similar to theirs despite the many changes. The changes include the use of spiking neural networks, the use of Spike-timing-dependent plasticity (STDP) instead of Hebbian learning, and weight normalization.
2) The network was trained for one epoch as opposed to 400 epochs in the Hebbian network, as simulating SNN is much slower. One training epoch takes between 18-48 hours on a GPU taking depending on the exposure time.
3) The filters look more pixelated since the input is converted into spikes when training the network. The spikes discretize the input and, as a result, the filters.

VI. Conclusion

In this work, we study unsupervised learning approaches, in particular Hebbian learning, to extract features from image data. Unsupervised learning approaches have been popular due to their biological significance, though they typically underperform BP results. We show that this is not always the case and compare the performance of unsupervised learning and supervised learning trained by backpropagation (BP). Specifically, we have the following main conclusions:

1) While features extracted using Hebbian learning cannot outperform BP learning, but they can achieve a comparable result. Features extracted in an unsupervised manner are limited in accuracy by the classifier trained in a supervised manner.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

2) While both the Hebbian network and the BP network converge to a set of weights, the weights extracted by Hebbian learning are more robust and generalize better. Previous work showed that there might be many different local minima's for a neural network, all of which give similar performance but have different adversarial robustness [1]. Hebbian learning helps extract robust and more generalized features.
3) Filters extracted using Hebbian learning are much smoother than the ones extracted by backpropagation. We hypothesize that this may be the reason for their robust performance. Moreover, the smooth filters help to explain the obtained results, thus turn the black-box model more transparent. It is the objective of ongoing and future studies to test this hypothesis.
4) We demonstrate a similar approach for training a spiking neural network in an unsupervised manner. We leave the complete implementation of SNN on the CIFAR-10 dataset for future work.

ACKNOWLEDGMENTS

This work has been supported in part by grant of the Defense Advanced Research Project Agency Grant, DARPA/MTO HR0011-16-l-0006.

REFERENCES

[1] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.
[2] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," *arXiv preprint arXiv:1712.04248*, 2017.
[3] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel, "Adversarial attacks on neural network policies," *arXiv preprint arXiv:1702.02284*, 2017.
[4] S. Witty, J. K. Lee, E. Tosch, A. Atrey, M. Littman, and D. Jensen, "Measuring and characterizing generalization in deep reinforcement learning," *arXiv preprint arXiv:1812.02868*, 2018.
[5] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.
[6] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers in computational neuroscience*, vol. 9, p. 99, 2015.
[7] H. Hazan, D. Saunders, D. T. Sanghavi, H. Siegelmann, and R. Kozma, "Unsupervised learning with self-organizing spiking neural networks," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–6.
[8] H. Hazan, D. J. Saunders, D. T. Sanghavi, H. Siegelmann, and R. Kozma, "Lattice map spiking neural networks (lm-snns) for clustering and classifying image data," *Annals of Mathematics and Artificial Intelligence*, pp. 1–24, 2019.
[9] D. J. Saunders, D. Patel, H. Hazan, H. T. Siegelmann, and R. Kozma, "Locally connected spiking neural networks for unsupervised feature learning," *arXiv preprint arXiv:1904.06269*, 2019.
[10] D. Patel, H. Hazan, D. J. Saunders, H. T. Siegelmann, and R. Kozma, "Improved robustness of reinforcement learning policies upon conversion to spiking neuronal network platforms applied to atari breakout game," *Neural Networks*, vol. 120, pp. 108–115, 2019.
[11] S. R. Kheradpisheh, M. Ganjtabesh, and T. Masquelier, "Bio-inspired unsupervised learning of visual features leads to robust invariant object recognition," *Neurocomputing*, vol. 205, pp. 382–392, 2016.
[12] T. Masquelier and S. J. Thorpe, "Unsupervised learning of visual features through spike timing dependent plasticity," *PLoS computational biology*, vol. 3, no. 2, p. e31, 2007.
[13] P. Ferré, F. Mamalet, and S. J. Thorpe, "Unsupervised feature learning with winner-takes-all based stdp," *Frontiers in computational neuroscience*, vol. 12, p. 24, 2018.
[14] G. Srinivasan, P. Panda, and K. Roy, "Stdp-based unsupervised feature learning using convolution-over-time in spiking neural networks for energy-efficient neuromorphic computing," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 14, no. 4, p. 44, 2018.
[15] P. Panda, G. Srinivasan, and K. Roy, "Convolutional spike timing dependent plasticity based feature learning in spiking neural networks," *arXiv preprint arXiv:1703.03854*, 2017.
[16] C. Lee, P. Panda, G. Srinivasan, and K. Roy, "Training deep spiking convolutional neural networks with stdp-based unsupervised pre-training followed by supervised fine-tuning," *Frontiers in neuroscience*, vol. 12, 2018.
[17] D. Krotov and J. J. Hopfield, "Unsupervised learning by competing hidden units," *Proceedings of the National Academy of Sciences*, vol. 116, no. 16, pp. 7723–7731, 2019.
[18] L. Grinberg, J. Hopfield, and D. Krotov, "Local unsupervised learning for image analysis," *arXiv preprint arXiv:1908.08993*, 2019.
[19] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data," *arXiv preprint arXiv:1805.09501*, 2018.
[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.