Convolutional neural networks with extra-classical receptive fields

Anonymous Author(s)
Affiliation
Address
email

Abstract

In the visual system, neurons respond to a patch of the input known as their classical receptive field (RF), and can be modulated by stimuli in the surround. These interactions can be mediated by lateral connections, giving rise to extra-classical RFs. We use supervised learning via backpropagation to learn feedforward connections, combined with an unsupervised learning rule to learn lateral connections between units within a convolutional neural network. These connections allow each unit to integrate information from its surround, generating extra-classical receptive fields for the units in our new proposed model (CNNEx). We demonstrate that these connections make the network more robust and achieve better performance on noisy versions of the MNIST and CIFAR-10 datasets. Although the image statistics of MNIST and CIFAR-10 differ greatly, the same unsupervised learning rule generalized to both datasets. Our framework can potentially be applied to networks trained on other tasks, with the learned lateral connections aiding the computations implemented by feedforward connections when the input is unreliable.

1. INTRODUCTION

While feedforward convolutional neural networks have resulted in many practical successes [1], they are highly susceptible to adversarial attacks [2]. In contrast, the brain makes use of extensive recurrent connections, including lateral and feedback connections, which may provide some level of immunity to these attacks (for results on human adversarial examples, see [3]). Additionally, the brain is able to build rich internal representations of information with little to no labeled data, which is a form of unsupervised learning, in contrast to the supervised learning required by most models.

We present a model incorporating lateral connections (learned using a modified Hebbian rule) into convolutional neural networks, with feedforward connections trained in a supervised manner. When applying different noise perturbations to the MNIST [4] and CIFAR-10 [5] datasets, lateral connections in our model improve the overall performance and robustness of these networks. Our results suggest that integration of lateral connections into convolutional neural networks is an important area of future research.

2. METHODS

Neuroscience-inspired extra-classical receptive field model: Let $fc_{j,x}^{m,(l)}$ denote the classical receptive field (RF) responses of a unit representing feature j in layer l at image location m, given image x. The full response of a unit representing feature j at image location m in layer l, given image x can then be written as:

$$f_{j,x}^{m,(l)} = fc_{j,x}^{m,(l)} \Big(1 + \alpha \sum_{k} \sum_{n \neq m} W_{jk}^{mn,(l)} fc_{k,x}^{n,(l)}\Big) \tag{1}$$

where the second term on the right side represents the contribution from the extra-classical RF, α represents a hyperparameter that tunes the strength of the lateral connections, and $W_{jk}^{mn,(l)}$ are the synaptic weights from surrounding units n on to unit m within layer l. These weights are learned in

Submitted to 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Do not distribute.

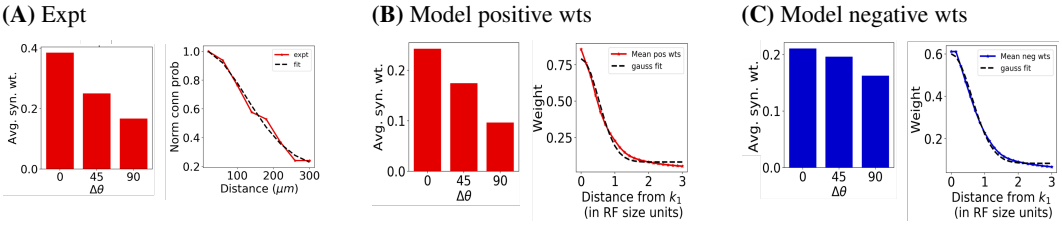

Figure 1: Orientation and distance dependence of lateral connections. **A)** Left: Connection probability as a function of difference in preferred orientation between excitatory neurons observed experimentally (from [8]). Right: Normalized connection probability between excitatory neurons as a function of inter-somatic distance as reported experimentally in mouse auditory cortex [9]. **B, C):** Model predictions for orientation and distance dependence (k_1 represents the target neuron) of positive (B) and negative (C) lateral connection weights for filters constructed using estimates of spatial receptive field (RF) sizes from in-vivo recordings in mouse V1 [7]. Red (blue) bars/lines represent positive (negative) weights and dashed black lines represent Gaussian fits for distance dependence (standard deviations $\sigma_{expt} = 114\mu m$, $\sigma_{pos} = 120\ \mu m$ and $\sigma_{neg} = 143\ \mu m$ for experiment, model positive and negative weights respectively). Predicted connections qualitatively match with experimental data.

an unsupervised manner using the rule:

$$W_{jk}^{mn,(l)} = \frac{\left\langle fc_j^{m,(l)} fc_k^{n,(l)} \right\rangle_x}{\left\langle fc_j^{m,(l)} \right\rangle_x \left\langle fc_k^{n,(l)} \right\rangle_x} - 1 \tag{2}$$

where $\langle . \rangle_x$ represents an average over the set of training images. Note that this formula differs from a Hebbian learning rule, in that only the covariance between the feedforward responses of units leads to changes in the lateral connections. A derivation of the above equations is sketched in the Appendix.

Using natural images from the Berkeley Segmentation Dataset [6] and a dictionary of classical RF features parameterized from mouse V1 electrophysiological responses [7], we computed lateral connection weights using Eq. 2. Figure 1 shows that predicted lateral connections qualitatively match the orientation and distance dependence of connectivity observed in mouse cortex.

Datasets: We used the standard train/test splits, holding out 10% of the training data for validation. We also added two types of noise to the original images: additive white Gaussian noise (AWGN) and salt-and-pepper noise (SPN). The mean of the AWGN was set to zero and the standard deviation varied in increasing levels from 0.1-0.5. For the SPN, the fraction of noisy pixels varied in increasing levels from 0.1-0.5. Example stimuli (original and noisy images) are shown in Figure 2.

Network architecture and training: The model architectures used are described in Table 1. Models were trained for 10 epochs with a minibatch size of 64 using stochastic gradient descent with a learning rate of 0.01 and a momentum value of 0.5. We trained 10 different instantiations using different random seeds to ensure the robustness of our results. All experiments were performed using Pytorch (v. 0.3.1) on a NVIDIA GTX 1080 Ti GPU.

Model	**Network architecture**							
CNN	conv5-10		maxpool	conv5-20		maxpool	FC-50	FC-10	soft-max
CNNEx	conv5-10	*conv7-10*	maxpool	conv5-20	*conv3-20*	maxpool	FC-50	FC-10	soft-max
CNN-PM	conv5-13		maxpool	conv5-26		maxpool	FC-50	FC-10	soft-max

Table 1: Model architectures used. CNN is the baseline model, CNNEx is the model with optimal lateral connections, and CNN-PM is a parameter-matched model to control for differences in network capacity. Convolutional layers are denoted as "conv<receptive field size>-<number of channels>". Convolutional layers in italics represent lateral connections learned in an unsupervised manner. "maxpool" denotes max pooling using a 2x2 window and a stride of 2. "FC" denotes fully connected layers with the given number of units. The ReLU activation function is not shown for brevity.

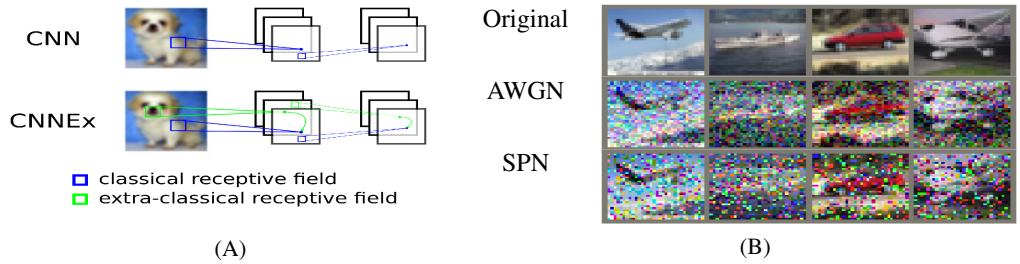

Figure 2: Overview of models and experiments. (A) Network architectures with feedforward classical receptive fields (RFs) (blue) and extra-classical RFs (green), termed CNN (top) and CNNEx (bottom), respectively. Extra-classical RFs modulate activity of other units within the network via lateral connections learned in an unsupervised manner. (B) Example images used in the experiments. Top row: original images. Middle row: AWGN stimuli. Bottom row: SPN stimuli. AWGN stimuli shown here had a mean of zero and a standard deviation of 0.2. SPN stimuli here had a fraction of changed pixels set to 0.2. Original images are reproduced from the CIFAR-10 [5] dataset.

Lateral connections: We first trained feedforward weights in the network using supervised learning. After freezing the feedforward weights, we introduced lateral connections given by eqn. 2 between units in the convolutional layers.

Network regularization: We chose a weight decay value of 0.005 and a dropout fraction of 0.5. Weight decay acted on all non-bias parameters of the model, while dropout was applied after each convolutional layer in the model, as well as after the first fully connected layer. We also tested the combination of these regularization techniques with the lateral connections.

Validation and testing: Lateral connections had a spatial extent of 7x7 (3x3) pixels in the first (second) convolutional layers, with connections from the same spatial location set to zero. Hyperparameters α for each of the two layers were chosen based on a grid search over the parameter range $\{0.1, 0.01, 0.001, 0.0001\}$ using the validation dataset, followed by a finer search starting from this coarse value. We did not use lateral connections for the two fully-connected layers.

3. RESULTS

Model performance: Table 2 summarizes model test accuracy with and without regularization (weight decay + dropout) for the network architectures described in Table 1. For both datasets, lateral connections provide improvement over the baseline model at higher levels of AWGN (above 0.3) and SPN (above 0.1). Lateral connections combined with regularization often resulted in even better performance at high noise levels for MNIST, but do not provide much additional benefit on CIFAR-10. This may be due to our models being somewhat underfit, as even our best performing model does not achieve close to state-of-the-art test accuracies on the CIFAR-10 dataset.

Effect of lateral connections: Lateral connections had two complimentary roles: redundancy reduction leading to sparsification of feature activations on images without noise, and noise reduction on noisy images leading to feature activations closer to those on the original images. The use of contextual information to modulate unit responses may underlie the ability of the CNNEx model to achieve higher accuracies under noisy conditions.

APPENDIX

Sketch of derivation of optimal lateral connection weights

We subdivide an image im_x into N patches $(im_x^1, ..., im_x^N)$ corresponding to the sizes of classical RFs. We assume a simple neuronal code in which unit activation is linearly related to the probability of the coded feature (F_k) to be present in patch n: $f_{k,x}^n = p(F_k^n | im_x)$ and define the classical RF response as $fc_{k,x}^n = p(F_k^n | im_x^n)$. We require that the sum of probabilities of all features in a patch is one, for every image so that $\sum_k fc_{k,x}^n = 1$ for all (n, x). For two patches ($N = 2$), assuming that the set of features represents a complete basis, we have

$$f_{j,x}^1 = p(F_j^1 | im_x^1, im_x^2) = \sum_k p(F_j^1 | im_x^1, im_x^2, F_k^2)\, p(F_k^2 | im_x^1, im_x^2) \tag{3}$$

	Models	Original	AWGN					SPN				
		-	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5
MNIST	CNN	98.71	98.61	98.21	**96.88**	92.03	81.78	**97.28**	92.01	80.85	65.29	48.28
	CNNEx	97.25	97.17	96.83	95.86	**93.34**	**88.24**	96.06	**93.45**	**87.97**	**77.99**	**63.04**
	CNN-PM	**98.79**	**98.67**	**98.31**	96.70	91.26	80.72	97.19	91.48	79.71	63.87	46.78
	CNN (wd+d)	97.39	97.30	97.01	96.43	95.23	92.82	96.63	95.16	92.01	85.61	73.98
	CNNEx (wd+d)	97.17	97.10	96.84	96.32	95.35	**93.59**	96.49	95.30	**93.11**	**88.53**	**79.30**
	CNN-PM (wd+d)	**97.69**	**97.61**	**97.32**	**96.82**	**95.69**	93.37	**97.04**	**95.67**	92.78	86.62	74.96
CIFAR-10	CNN	60.46	53.36	37.92	27.04	21.35	18.18	39.40	27.31	21.11	17.48	15.29
	CNNEx	58.32	53.20	39.78	**29.14**	**22.92**	**19.52**	**41.34**	**29.32**	**22.65**	**18.70**	**16.29**
	CNN-PM	**61.76**	**54.76**	**39.88**	28.90	22.65	19.13	41.06	28.85	22.32	18.39	15.91
	CNN (wd+d)	42.93	42.10	39.86	36.58	32.75	29.23	40.04	36.52	32.53	28.08	**24.01**
	CNNEx (wd+d)	42.94	42.14	39.89	36.57	32.77	29.24	40.08	36.55	32.55	28.09	23.99
	CNN-PM (wd+d)	**45.68**	**44.92**	**42.53**	**38.87**	**34.54**	**30.02**	**42.85**	**38.95**	**34.41**	**28.95**	23.79

Table 2: Model accuracy (%) on the MNIST and CIFAR-10 datasets. We separate results for the original images and the two types of noise perturbations by columns (AWGN: additive white gaussian noise, SPN: salt-and-pepper noise). The results for the baseline model (CNN), the model with lateral connections (CNNEx), and the parameter-matched model (CNN-PM) are shown in separate rows. wd corresponds to models trained with weight decay = 0.005. d corresponds to models trained with a dropout fraction of 0.5. All reported values are averages over 10 random initializations.

where the sum is over the k neurons with classical receptive fields in patch im_x^2. We make some simplifying assumptions (amounting to including only first order and excluding higher-order extra-classical RF influences): (i) $p(F_j^1|im_x^1, im_x^2, F_k^2) \approx p(F_j^1|im_x^1, F_k^2)$, (ii) $p(F_k^2|im_x^1, F_j^1) \approx p(F_k^2|F_j^1)$, (iii) $p(F_k^2|im_x^1) \approx p(F_k^2)$ and (iv) $p(F_k^2|im_x^1, im_x^2) \approx p(F_k^2|im_x^2)$. Using Bayes' rule and assumptions (ii-iv), we can write $p(F_j^1|im_x^1, F_k^2) = \big[p(F_j^1|im_x^1)\, p(F_k^2 \cap F_j^1)\big] / \big[p(F_j^1)\, p(F_k^2)\big]$. This combined with assumption (i) in Eq. 3 and the normalization condition gives $p(F_j^1|im_x^1, im_x^2) = p(F_j^1|im_x^1)\left(1 + \sum_k W_{jk}^{12} p(F_k^2|im_x^2)\right)$ where $W_{jk}^{12} = \big[p(F_j^1 \cap F_k^2) - p(F_j^1)\, p(F_k^2)\big] / \big[p(F_j^1)\, p(F_k^2)\big]$. Going from two to N patches and rewriting using the neuronal code above, if each patch provides independent information, then $f_{j,x}^1 = fc_{j,x}^1 \prod_{n \neq 1}^N \left(1 + \sum_k W_{jk}^{12} fc_{k,x}^2\right)$. Interpreting W_{jk}^{12} as a synaptic weight, the full unit response can thus be obtained by summing weighted lateral contributions from k neurons in each patch n and multiplying contributions from all patches with the classical RF response $fc_{j,x}^1$. This simplifies to $f_{j,x}^1 = fc_{j,x}^1 \sum_k \sum_{n \neq 1}^N \left(1 + \sum_k W_{jk}^{12} fc_{k,x}^2\right)$ for small contributions from each patch. Given the code, weights can be estimated as $W_{jk}^{mn} = \big[\big\langle fc_{j,x}^m c f_{k,x}^n \big\rangle_x\big] / \big[\big\langle fc_{j,x}^m \big\rangle_x \big\langle fc_{k,x}^n \big\rangle_x\big] - 1$. The last two expressions correspond to equations 1 and 2 used in the CNNEx model.

References

[1] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. Recent advances in convolutional neural networks. *Pattern Recognition*, 2017.

[2] Jiawei Su, Danilo Vasconcellos Vargas, and Sakurai Kouichi. One pixel attack for fooling deep neural networks. *arXiv preprint arXiv:1710.08864*, 2017.

[3] Gamaleldin F Elsayed, Shreya Shankar, Brian Cheung, Nicolas Papernot, Alex Kurakin, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial examples that fool both human and computer vision. *arXiv preprint arXiv:1802.08195*, 2018.

[4] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

[5] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[6] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. *Proc. 8th Int'l Conf. Computer Vision*, pages 416–423, 2001.

[7] S. Durand, R. Iyer, K. Mizuseki, S. De Vries, S. Mihalas, and R.C. Reid. A comparison of visual response properties in the lateral geniculate nucleus and primary visual cortex of awake and anesthetized mice. *Journal of Neuroscience*, 36(48), 2016.

[8] Ho Ko, Sonja B Hofer, Bruno Pichler, Katherine A Buchanan, P Jesper Sjöström, and Thomas D Mrsic-Flogel. Functional specificity of local synaptic connections in neocortical networks. *Nature*, 473(7345):87–91, 5 2011.

[9] Robert B. Levy and Alex D. Reyes. Spatial Profile of Excitatory and Inhibitory Synaptic Connectivity in Mouse Primary Auditory Cortex. *Journal of Neuroscience*, 32(16), 2012.