A Deconvolutional Competitive Algorithm for Building Sparse Hierarchical Representations

D.M. Paiton [*]
Vision Science Grad. Group
Univ. of California, Berkeley
dpaiton@berkeley.edu

S.Y. Lundquist
Dept. of Computer Science
Portland State University

W.B. Shainin
The New Mexico Consortium

X. Zhang [†]
The University of New Mexico

P.F. Schultz
The New Mexico Consortium

G.T. Kenyon [‡]
Los Alamos National Lab
garkenyon@gmail.com

ABSTRACT

Sparse coding methods have been used to study how hierarchically organized representations in the visual cortex can be learned from unlabeled natural images. Here, we describe a novel Deconvolutional Competitive Algorithm (DCA), which explicitly learns non-redundant hierarchical representations by enabling competition both within and between sparse coding layers. All layers in a DCA are trained simultaneously and all layers contribute to a single image reconstruction. Because the entire hierarchy in a DCA comprises a single dictionary, there is no need for dimensionality reduction between layers, such as MAX pooling. We show that a 3-layer DCA trained on short video clips obtained from hand-held cameras exhibits a clear segregation of image content, with features in the top layer reconstructing large-scale structures while features in the middle and bottom layers reconstruct progressively finer details. Compared to lower levels, the representations at higher levels are more invariant to the small image transformations between consecutive video frames recorded from hand-held cameras. The representation at all three hierarchical levels combine synergistically in a whole image classification task. Consistent with psychophysical studies and electrophysiological experiments, broad, low-spatial resolution image content was generated first, primarily based on sparse representations in the highest layer, with fine spatial details being filled in later, based on representations from lower hierarchical levels.

[*]Secondary affiliation: The Redwood Center for Theoretical Neuroscience, Univ. of Calif., Berkeley

[†]Secondary affiliation: The New Mexico Consortium

[‡]Secondary affiliation: The New Mexico Consortium

Categories and Subject Descriptors

I.4.10 [**Image Representation**]: Algorithms

Keywords

sparse coding, visual cortex, hierarchical model, deep learning, convolutional neural network, deconvolution

1. INTRODUCTION

Convolutional deep networks have become an integral tool for both machine learning applications and for constructing models of the visual cortex [13, 21]. However, many deep learning algorithms require a large amount of labeled data, which can be costly to acquire and often introduce biases from human-labeling methods. An ongoing challenge for deep learning researchers as well as vision neuroscientists has been to develop deep, hierarchical networks that can learn from unlabeled data, the quantity and variety of which vastly exceeds the limited quantity of labeled data. To this end, features are learned from visual experience alone with a goal of representing data in a manner that best promotes visually-guided behaviors. One widely used approach to building such representations that supports a variety of signal classification tasks is sparse coding.

Sparse coding techniques have been employed to model the high-order statistical structure of sensory inputs [26]. Although sparse coding has been associated with multiple brain areas, including auditory networks, olfactory networks, and the retina [8, 4, 18], sparse approximation methods have been most extensively used to model the responses of visual neurons [35], including classical receptive field properties of V1 simple cells [16, 17, 3, 20, 36]. Using a Locally Competitive Algorithm (LCA), a dynamic sparse solver implemented as a neural network with symmetric lateral inhibition, it has been demonstrated that sparse coding methods can describe *non-classical* contrast-invariant tuning [22, 33, 34]. Sparse coding thus provides a neurally-plausible framework for understanding how sensory cortices can self-organize in response to unlabeled sensory data.

Here, we describe a novel Deconvolutional Competitive Algorithm (DCA) that combines a deconvolutional approach to constructing deep, sparse, hierarchical representations with local competition among neurons throughout the hierarchy so as to generate less redundant multi-scale encodings.

BICT 2015, December 03-05, New York City, United States

DOI 10.4108/eai.3-12-2015.2262428

We employ a novel connectivity method, where each layer in the hierarchy receives a projection of the image reconstruction error convolved up through lower layers. Each layer generates a pixel-level image reconstruction by deconvolving its sparse representation back through the lower layers in the hierarchy. The modeled inhibitory feedback connections from higher layers draw inspiration from the ample feedback connections present from higher visual cortical areas to lower, consistent with the Rao and Ballard predictive coding framework [19].

To find a sparse approximation of an image, we use gradient descent to minimize an energy function that penalizes the sum of the squared pixel-by-pixel errors between the input and the reconstruction. The reconstruction is generated from the activations of all layers in the network. This is accomplished via serial deconvolution, with each deconvolutional step consisting of a weighted sum of features (also commonly referred to as basis vectors, filters, or weights). The energy function further penalizes the number of features used in the reconstruction, thereby ensuring that the resulting representations are sparse. The features are learned iteratively via stochastic gradient descent on the same energy function, resulting in a neurally-plausible, local Hebbian weight update rule. Because sparse coding introduces competition between all features that overlap a given image patch, the corresponding activation coefficients will tend to be statistically independent, both within and between layers. Within each layer, we adopt a convolutional approach in which a reduced set of learned kernels is replicated with a stride that is much smaller than the patch size, allowing for the construction of overcomplete dictionaries using a relatively small number of learned features [23]. Because the entire hierarchy in a DCA forms a single dictionary, the degree of overcompleteness is determined by the total number of neurons in the hierarchy, thereby obviating the need for a dimensionality reduction step between layers. Our sparse solver is highly compatible with hardware implementations [12, 24, 30] and allows for considerable flexibility in the implementation of different network topologies.

Below, we show that image content is naturally segregated by a 3-layer DCA such that large, low spatial-frequency structures are mostly represented at the top level in the hierarchy while lower levels represent progressively finer spatial detail. We further show that there is less change in the sparse representations between consecutive video frames at the top-most level than at lower levels, indicating that representations are more stable at the top of the hierarchy under the typically small image transformations that occur between consecutive video frames. To determine how much independent information each layer encodes from the scene, we trained a linear classifier on a whole image labeling task using activities produced by a 3-layer DCA network. The performance supported by concatenating activations from the entire hierarchy greatly exceeds the performance supported by any single layer indicating that non-redundant information is represented at each hierarchical level. Finally, we explored the feature redundancy between and within layers for the DCA network against a comparable network that utilized MAX pooling for dimensionality reduction, demonstrating that our model has lower representational redundancy.

2. RELATED WORK

There are several published deep convolutional models for unsupervised feature learning. Many of them [7, 31, 15, 6, 29] learn each layer iteratively, such that each layer is only learned once the previous layer's features have converged. Unlike these models, DCA learns all layer features simultaneously and activities for the whole network are updated together in a dynamic, recurrent fashion. Our architecture also does not employ any pooling operations when learning unsupervised features, a method that is used widely in deep network models [7, 32, 15, 11, 10, 14]. Finally, for many architectures (e.g. [10, 11]), each layer only reconstructs the output of the layer below. With DCA, all layers are globally competing to reconstruct the input image.

Our model was originally inspired by LCA [22], a generative online single-layer neural network for sparse approximation. As originally described, the LCA model was non-convolutional and did not include a rule for learning features. A single layer in DCA can be thought of as a convolutional LCA with a Hebbian weight learning step. When placed in a hierarchy, the network architecture bears much similarity to the 2-layer predictive coding network described by Rao and Ballard [19], although their originally described model follows different thresholding dynamics and lacks global competition for reconstructing the input data.

Our model can also be readily compared to the Adaptive Deconvolutional Networks (ADNs) described by Zeiler et al. [32]. ADNs alternate convolutional sparse coding layers with MAX pooling; however, instead of learning a sparse representation of the MAX pooled output of the previous layer, each layer attempts its own reconstruction of the input image. Conversely, our model attempts to encode information that other layers fail to represent. The DCA is distinct in that it enforces all layers in the hierarchy to compete for the image representation so as to generate less redundant multi-scale encodings.

3. METHODS

We trained a DCA consisting of 3 sparse coding layers on a series of still images or short video clips. Figure 1 shows a circuit diagram of our 3-layer DCA. Each sparse coding layer participates in a predictive coding feedback loop, receiving input from the image reconstruction error layer convolved through all lower levels in the hierarchy and sending output back to the image reconstruction error layer via deconvolutions implemented as transposes of the forward pathway. The video dataset consisted of Twitter Vine video clips that were sampled and organized using the ViPar toolkit [2].

Except where otherwise noted, all subscripts denote layer number. Formally, we represent the feature set Φ_n as $K_{n-1} \times K_n$ matrix and activations S_n as a K_n-dimensional vector for layer n. The input image I is represented as a K_0-dimensional vector. In this work, we consider only non-negative activations, $S_n \geq 0$. Our energy function follows what is typically used in sparse coding literature:

$$E = \overbrace{\frac{1}{2} \parallel I - I' \parallel_2^2}^{\text{Reconstruction Error}} + \overbrace{\sum_n^N \lambda_n C(S_n)}^{\text{Sparsity Constraint}}. \quad (1)$$

However, we include contributions from all layers in the DCA hierarchy to form the image reconstruction, I':

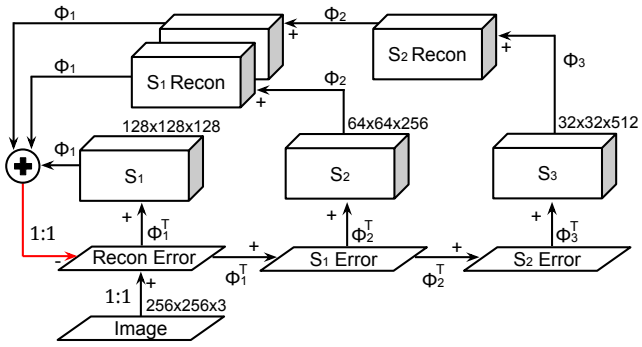

Figure 1: Schematic illustrating a deconvolutional competitive algorithm (DCA). Positive connections are indicated in black, negative connections in red. The Recon Error layer outputs the difference between Image and a sum of reconstructions from $\{S_1, S_2, S_3\}$. The reconstructions are computed using the features $\{\Phi_1, \Phi_2, \Phi_3\}$, which are learned via a local Hebbian learning rule that reduces the total image reconstruction error. The receptive fields at S_3 are twice the diameter of those at S_2 and are spaced twice as far apart, while the receptive fields at S_2 are in turn twice the diameter and spaced two times as far apart as those at S_1. See text for additional implementation details. The 1:1 notation indicates a connection with identity weights.

$$I' = \Phi_1 S_1 + \Phi_1 \Phi_2 S_2 + \Phi_1 \Phi_2 \Phi_3 S_3 + \cdots + \prod_{n=1}^{N} \Phi_n S_N.$$

In equation 1, the λ_n parameters control the trade-off between reconstruction error and sparseness, and the $C(S_n)$ are cost functions that encourage sparsity. Here, we use the l_1 cost function: $C(S_n) = \sum_j^{K_n} S_{n,j}$, which corresponds to a rectified soft-threshold transfer function on an membrane potential [22]:

$$T_{\lambda_n}(u_n) = \left\{ \begin{array}{ll} u_n - \lambda_n & , u_n > \lambda_n \\ 0 & , u_n \leq \lambda_n \end{array} \right\}. \quad (2)$$

Where $S_n = T_{\lambda_n}(u_n)$. Thus, the energy function can also be written as:

$$E = \frac{1}{2} \parallel I - I' \parallel_2^2 + \sum_n^N \lambda_n T_{\lambda_n}(u_n),$$

where the activation function is replaced with a thresholded internal state, u_n. The internal state models the neuron membrane potential, which does not influence other neurons except through the transfer function $T_\lambda(.)$.

For given features $\Phi_1, \Phi_2, \Phi_3, \ldots, \Phi_N$ and input image I, DCA finds sparse activations that minimize equation 1. The columns of Φ_n are local kernels that are replicated with a stride that varies from layer to layer [23]. Each layer reconstructs a vector of P pixels using kernels that are replicated across the input with a stride of J_n. Although the dimensions of Φ_n are formally K_{n-1}-by-K_n, the number of nonzero elements, determined by the local support for each kernel or kernel patch size, is much smaller. We adjust the kernel patch size between layers so that the number of pixels in the reconstructed image to which any given neuron can contribute increases as a function of hierarchical level. The strides J_n similarly increase with hierarchical level. Our model can be extended to any number of layers; however, all DCA networks presented here consist of three sparse coding layers.

Following the approach used in LCA [22] and Schultz et al. [23], a minimum of E with respect to the sparse activation variables S_n is found by solving a discrete version of the following differential equation:

$$\tau_n \frac{\partial u_n}{\partial t} = -\frac{\partial E}{\partial S_n} = \left\{ \begin{array}{l} u_n + \prod_{i=1}^{n} \Phi_i^{\mathsf{T}} \left(I - I'\right), u_n \leq \lambda_n \\ -\lambda_n + \prod_{i=1}^{n} \Phi_i^{\mathsf{T}} \left(I - I'\right), u_n > \lambda_n \end{array} \right. . \quad (3)$$

Each $u_{n,k}$ for the k^{th} neuron in the n^{th} layer varies as a function of time, which we model as discrete 1ms steps, with time constants τ_n. To update the features on time step $t+1$, we chose to begin with a local Hebbian learning rule instead of directly minimizing the energy function with respect to the features:

$$d\Phi_n^{Hebb} = \left(\left(\prod_{m=1}^{n-1} \Phi_m^{\mathsf{T}} \right) \left(I - I'\right) \right) S_n^{\mathsf{T}}. \quad (4)$$

As described previously, the Φ_n matrix is composed of smaller replicated kernels that share features across the input image. $d\Phi_n^{Hebb}$ is then averaged across the contributions from all such replicas. Our feature update applies this Hebbian rule, augmented by a momentum term to accelerate learning:

$$\begin{aligned} \Phi_n^{(t+1)} &= \Phi_n^{(t)} + \Delta\Phi_n^{(t+1)} \\ \Delta\Phi_n^{(t+1)} &= \eta_n d\Phi_n^{Hebb} + e^{\frac{-1}{\chi_n}} \Delta\Phi_n^{(t)}, \end{aligned} \quad (5)$$

where η_n is the learning rate for the Φ_n connection and χ_n is a dimensionless rate that determines how quickly the momentum decays between feature update steps. The update is applied once per image after a sparse approximation has been found for that image.

All simulations were performed using PetaVision [1], an open-source, high-performance neural simulation toolbox. The differential equations for the u_n were solved in discrete time using an explicit method. To accelerate convergence, the time step size was increased over the course of obtaining a sparse hierarchical representation for any given image in a manner that was controlled by the image reconstruction error. Parameter files and associated executables for generating the results reported here are hosted as publicly available Amazon Machine Instances (AMIs) from Amazon Web Services (AWS), which can be accessed via the PetaVision website [1].

4. RESULTS

When trained on short video clips of $256 \times 256 \times 3$ pixels, the 3-layer DCA produced a hierarchical segmentation of image content (Figure 2). The video frames were individually modified to have a mean of zero and standard deviation

of one. Each video frame was presented for 1,200 time steps to allow the network to converge on a sparse approximation. We used strides of 2, 4, and 8 pixels, time constants of 400, 800, and 1,200 ms, and spatial kernel patch sizes of 18×18, 36×36, and 72×72 pixels for sparse coding layers S_1, S_2 and S_3, respectively. Within an S_1 kernel, each of the 128 neurons received an $18 \times 18 \times 3$ image patch as input, with the third dimension representing color channels. A given S_2 neuron received input from a $10 \times 10 \times 128$ patch of S_1 neurons. Taking into account the 2 pixel stride of the S_1 kernels, this results in the S_2 kernels having an edge size of $1*18+2*9 = 36$ image pixels. From this, one can determine that the Φ_1 dictionary contains 124,416 free parameters, the Φ_2 dictionary contains 3,276,800 free parameters and the Φ_3 dictionary contains 13,107,200 free parameters. To compute the overcompleteness of each layer, we compare the total number of elements in the layer against the total number of inputs to the layer, multiplied by a scaling factor of 0.5 to account for rectification. Thus, S_1 has 2,097,152 rectified neurons and $196,608$ pixel inputs, which results in a $\sim 5.3\times$ overcomplete representation. The S_2 and S_3 layers are $0.5\times$ undercomplete with respect to their input layers, and thus the image-wise overcompleteness decreases by a factor of $1/2$ as one ascends the hierarchy. The threshold parameters λ_n were adjusted to ensure that each layer contributed to the combined reconstruction. Final values for λ_n were 0.003125, 0.0125 and 0.05, which produced per-image mean activations of 2.3%, 1.2% and 1.3% (percentage of neurons with non-zero activation coefficients) for layers S_1, S_2 and S_3, respectively. We assigned the ratio of values for λ_n to match the ratio of strides for the corresponding S_n layers and then followed a coarse grid search procedure to find the relative scaling. We reserve systematically exploring the parameter space, such as setting λ_n to force mean activations for each layer to be equal, for future work.

The sparse representations at the top of the hierarchy reconstructed large, low spatial frequency image components while still preserving semantic information. Sparse representations generated by the middle hierarchical layer reconstructed elongated, high spatial frequency image contours that were essential for defining object boundaries. Finally, reconstructions generated by the lowest hierarchical level emphasized high spatial frequency, pixel-level structure. The spatial frequency segmentation is quantified in Figure 3, which shows the rotationally (phase) averaged power spectrum of the individual reconstructions deconvolved from the S_1, S_2 and S_3 representations for 2000 different natural scenes.

Numerical stability requires that we set the base time constants for $S_3 : S_2 : S_1$ to a ratio of $3 : 2 : 1$, respectively. However, even for a single sparse coding layer, the effective time constants are determined by both the base constant and by the eigenvalues of the symmetric weight matrix, $\Phi^{\mathsf{T}}\Phi$. For a hierarchical sparse model, such as DCA, similar considerations apply, such that the eigenvalues are determined by symmetric combinations of weight matrices. Intuitively, the negative feedback employed by DCA can greatly affect how quickly the membrane potential of a given neuron charges to its stationary value for any given image. Analysis of retinal response curves [27] and psychophysical studies [28, 5] have established an expectation that the lower spatial resolution cells should respond most quickly. Figure 4 shows that, in spite of their longer base time constants, the neu-

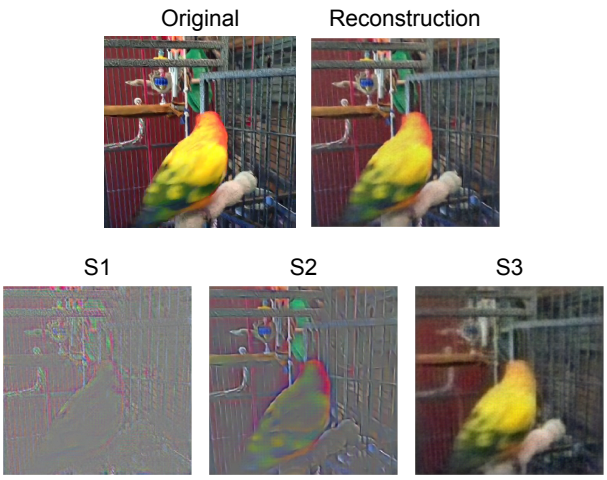

Figure 2: Hierarchical segregation of image content across DCA layers. *Top row:* The original image is on the left. The reconstruction combining contributions from all three layers is on the right. *Bottom row:* Reconstructions from representations in each associated layer. Vertical competition in combination with progressively larger strides and patch sizes results in the highest layer (S_3) capturing large-scale, low-resolution image components while lower layers (S_2 and S_1) capture progressively finer spatial details.

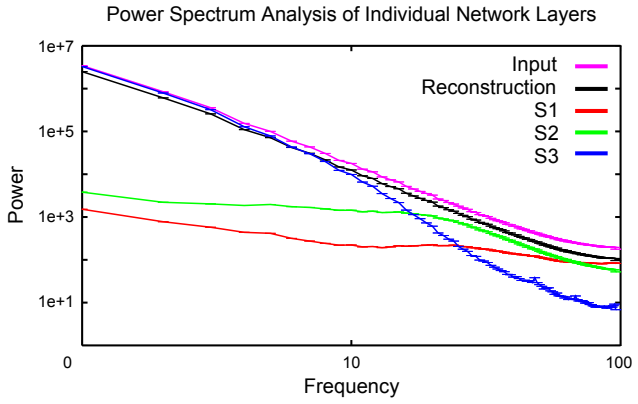

Figure 3: Power spectrum analysis. Values indicate rotationally (phase) averaged power from 0 to 96 cycles per image. Error bars indicate standard error of the mean from 2000 non-consecutive natural images. *Magenta:* Input frames. *Black:* Combined reconstruction using all layers. *Red:* S_1 reconstructions. *Green:* S_2 reconstructions. *Blue:* S_3 reconstructions. S_3 has the most power at lower spatial frequencies, S_2 has the most power in the middle range and S_1 has the most power at frequencies greater than 61 cycles per image. The maximum detectable frequency is 96 cycles per image, which is limited by the smallest dimension of the image.

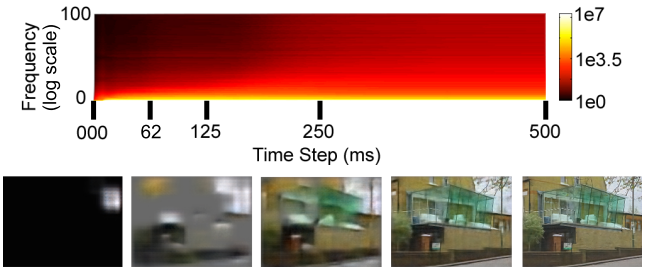

Figure 4: Reconstruction spectrogram. The spectrogram plot shows that lower spatial frequencies have more power earlier on, and the expected 1/f structure doesn't arise until the S_1 and S_2 neurons activate. The five frames pictured are combined reconstructions corresponding to the labeled time step values on the spectrogram plot. Color indicates power in the fourier space. We have ignored the first 28 time steps because none of the model neurons were above threshold before this point.

rons representing the lowest spatial-frequency information are the first to respond. From our analysis of the power spectra of reconstructions arising from each layer, we conclude that the relevant eigenvalues controlling the dynamics of our 3-layer DCA network adapt in such a manner so that features with the lowest spatial resolution, primarily drawn from layer S_3, activate first, reconstructing large, diffuse image components.

Each hierarchical layer reconstructs image content from its sparse representations, which are consistent with the learned kernels (Figure 5) that have been deconvolved to image space for visualization. The hierarchical segregation of image content produced by the 3-layer DCA appears to be a direct result of the different strides, time constants, and patch sizes employed at the 3 different hierarchical levels. The larger strides, slower time constants and larger patch sizes encourage higher layers to encode broad, diffuse structures that persisted across video frames. The global competition causes "explaining away" to happen between layers; layers must only explain features that are not already accounted for by the other layers. As a result, the capture of low spatial frequency structure at the top of the hierarchy forces lower layers to capture progressively finer spatial detail.

Videos recorded from hand-held cameras introduce small transformations between consecutive frames. Given the relatively high frame-rate, the overall content of the scene is largely unchanged from one frame to the next. Such transformations provide natural sequences for assessing the stability of sparse representations at different hierarchical levels. Frame-to-frame stability in representing natural movie scenes also gives an indication of invariances to transforms that are likely to occur, such as changes in object position and scale. On average, the sparse representations generated by S_3 features are 5.2% more stable across successive video frames than are the sparse representations generated by S_2 features, which are in turn 0.5% more stable than S_1 representations. Stability over time is computed by measuring the change in the corresponding sparse representations across consecutive frames. The change in the representation is defined here as the ratio of the number of elements that change activation state (from 0 to a non-zero value or vice versa) divided by the total number of active elements in either frame, equivalent to a sliding XOR/OR logical operation applied to the binarized activations for each successive pair of frames. Averaged over all frames, the mean percent change in representation was 52.6%, 52.1% and 46.9% for layers S_1, S_2 and S_3, respectively.

To directly assess the absence of redundancy in the representations generated by DCA layers, we trained a 3-layer network on the CIFAR10 database. Because CIFAR10 images are tiny (32×32 pixels), the strides and patch sizes of each sparse coding layer were reduced from the values used previously. See the table 1 description for the values used. The sparse representations generated by each layer in the DCA were averaged over a 5×5 spatial neighborhood centered on each neuron. The average-pooled output of each layer, either individually or concatenated with other layers, was then used to train a linear SVM. We computed classification accuracy via cross-validation on the official test set. The pooling step was only done for classification purposes and had no influence on the features learned. The classification results show that the performance of any combination of layers was always greater than any of the component layers and the best performance was obtained when all three layers were concatenated together (Table 1). Thus, DCA generated non-redundant sparse coding hierarchical representations using an unsupervised learning rule in which the features encoded by any given layer could be combined synergistically with the features encoded by other layers to support improved performance on an image classification task.

As previously stated, many alternative approaches to constructing hierarchical representations train each layer to represent the MAX pooled activity in the previous layer. We have hypothesized that the representations generated by inserting a MAX pooling step between each sparse coding layer will be more redundant than the representations generated by DCA. To test this hypothesis, we measured the degree of orthogonality within and between layers for the features learned in a 3-layer DCA network and compared the results to a control network based on conventional MAX pooling. Both networks contained the same number of sparse coding layers, with each sparse coding layer containing the same number and density of neurons. Each network also employed identical strides and equivalent patch sizes for the connections between layers (the patch sizes in the MAX pooled network were reduced by a factor of two in the second and third layers in order to match the pixel-level field of view). The thresholds in the MAX pooling network were adjusted so as to approximately match the level of sparsity in the corresponding layers in the DCA network. We measured orthogonality by deconvolving each feature back to image space, scaling the l_2 norm of the resulting images to unity, then convolving the normalized image back up through the network and recording the activations at each point in the hierarchy. The resulting Gramian matrix represents the activity correlation for each feature compared to each other feature, which is analogous to the cross-correlation of the features in pixel space. Our results, displayed in Figure 6, indicate that the features learned by DCA are less redundant than the features learned via conventional MAX pooling.

5. DISCUSSION

DCA uses local competition both within and between hierarchical layers to avoid the need for combinatorially increas-

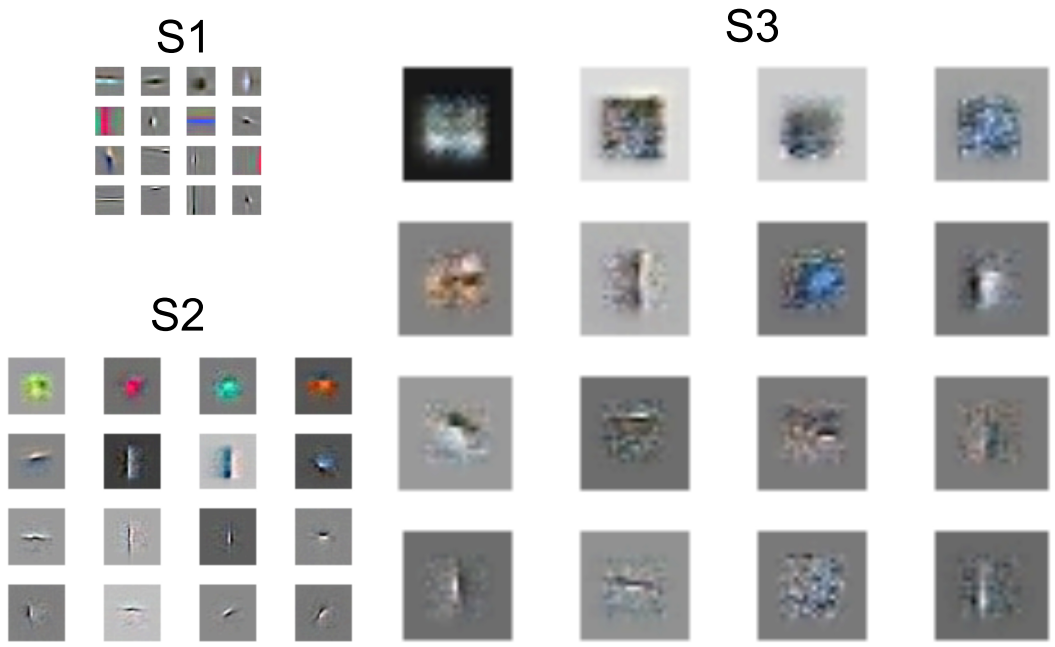

Figure 5: Representative features. ***Top left:*** S_1 **features (patch size =** $18 \times 18 \times 3$ **pixels).** ***Bottom left:*** S_2 **features (spanning** 36×36 **pixels).** ***Right:*** S_3 **features (spanning** 72×72 **pixels).** S_2 **and** S_3 **features have been deconvolved to image space for purposes of visualization. Features exhibit more large-scale, low-frequency structure at progressively higher levels in the hierarchy. The subset of elements shown represent a sampling of features for the most active neurons, excluding the top** 5%**, which tend to be diffuse whitening elements. The region of inactivation around the** S_3 **features is a result of edge effects when using large patch sizes in convolutional networks, which suppresses structure in that portion of the patch that falls outside of the image margins for** S_3 **columns withing a patch radius of the border.**

Table 1: Image Classification Result on CIFAR-10 Dataset. S_1**:** 32 **kernels (stride =** 1 **, patch size =** 5×5**);** S_2**:** 32 **kernels (stride =** 2**, patch size =** 10×10**);** S_3**:** 128 **kernels (stride =** 4**, patch size =** 20×20**); patch sizes expressed in pixels after deconvolution to image space. For classification, we used average pooling over each layer (stride =** 1 **and patch size =** 5×5**). Feature vectors for classification are composed of either a single layer or a concatenation of multiple layers. Accuracy increased when more layers were used, showing that each layer encoded non-redundant information.** S_2 **had the most discriminative representation.**

	S1	S2	S3	S1+S2	S2+S3	S1+S3	S1+S2+S3
Acc.	42.39%	59.84%	56.45%	61.89%	65.91%	60.92%	66.65%

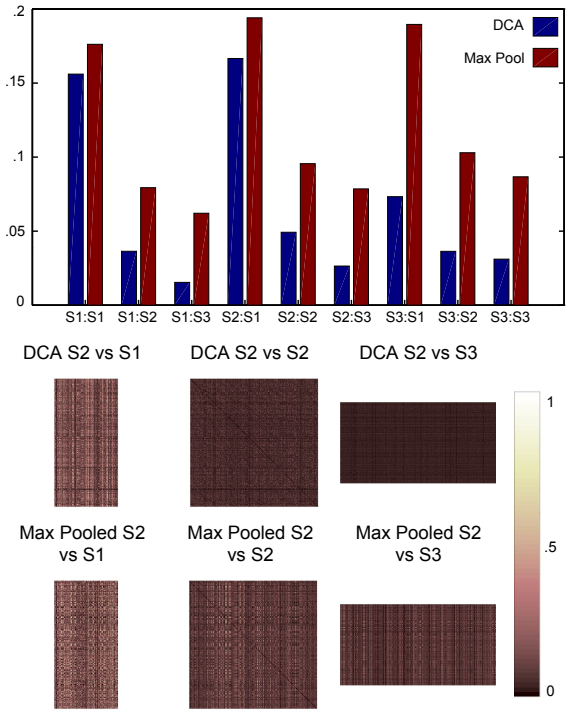

Figure 6: Representational redundancy comparison for DCA and MAX pooled networks. The bar graph shows the degree of orthogonality within and between layers for a 3-layered DCA network and a control network based on conventional MAX pooling, with the bar height indicating activity correlation. Representative heat maps show element-by-element correlation comparisons for layer S_2 in the DCA (*top row*) and MAX pooling (*bottom row*) networks, where lighter colors indicate higher correlation and thus more redundancy. In all cases, the DCA network has a lower mean activation correlation than the MAX pooling network, indicating reduced redundancy (and thus increased orthogonality) between features. See text for implementation details.

ing the number of neurons in each layer as one ascends the hierarchy (the so-called combinatorial explosion problem). The novel tree-like structure imposed on the DCA dictionary provides a single multi-scale representation of the entire image, where each scale level is linearly combined. Each level in the hierarchy only encodes those image components that are not efficiently represented at other levels, allowing for "explaining away" to occur across the hierarchy. Prediction errors are sent as forward projections, while sparse representations are deconvolved to image space in backward projections. This forward pattern of connectivity is inspired from research on the role of the corticothalmocortical connections via the pulvinar nucleus of the thalamus [25], which could serve as a central relay to project prediction information from lower cortical areas to higher areas. In addition to layer-layer competition, model neurons within each layer compete to build a representation of the input. Although we do not formally investigate how such local competition between cortical neurons might be implemented in the primate cortex, we refer the reader to a theory implicating lateral inhibition [22, 34, 9]. By eliminating the need for a dimensional reduction step between layers, local competition acts as a replacement for the MAX pooling step commonly inserted between coding layers in unsupervised deep architectures. MAX pooling also provides a non-linearity in the construction of deep architectures to reduce filter redundancy among layers at the cost of destroying some potentially useful spatial relationships. Because sparse coding and rectification inherently provide additional non-linearities in the projection to representation space, we were able to use a simple top-down competition term in our model to reduce redundancy in representation among layers without losing any spatial data. Our network's convolutional nature allows us to construct a hierarchical dictionary for sparse coding of large images, where each dictionary layer is composed in terms of the dictionaries at lower levels. We believe that this can provide a powerful method for learning unsupervised features that could be combined with traditional deep-network approaches for large-scale semi-supervised learning problems.

6. CONCLUSION

Published unsupervised learning algorithms have previously been described in a hierarchical and convolutional framework to learn efficient representations that encode different spatial scales. There has been an increasing interest in such models to leverage the massive amount of unlabeled image data in order to improve supervised network performance on image description tasks. We present a novel deep, hierarchical and deconvolutional sparse coding model. The deconvolutional competitive algorithm (DCA) learns explicitly non-redundant hierarchical representations by enabling competition both between and within layers. Our hierarchy is built without the need for lossy pooling steps between layers, and demonstrates stability for encoding time-varying input. The DCA is also more constrained by current theories of visual cortical function than other related models, resulting in a more biologically plausible architecture. The model exhibits response properties that are reflected in biological systems, incuding the timecourse of representations spanning different spatial resolutions as well as learning gabor-like and color opponent neuronal receptive fields at multiple spatial scales.

7. ACKNOWLEDGMENTS

We would like to thank Daniel Delott and Brian Broom-Peltz for curating our video dataset and Max Theiler for contributions to the writing of the manuscript. Much of this work was inspired and supported by discussions with Professor Bruno Olshausen. This work was funded by the DARPA UPSIDE program.

8. REFERENCES

[1] Petavision, 2015. https://petavision.github.io/.

[2] Petavision vipar toolkit, 2015. https://github.com/PetaVision/ViPar-Toolkit.

[3] M. V. Albert, A. Schnabel, and D. J. Field. Innate visual learning through spontaneous activity patterns. *PLoS Computational Biology*, 4(8):e1000137, 2008.

[4] M. J. Berry, D. K. Warland, and M. Meister. The structure and precision of retinal spike trains. *Proceedings of the National Academy of Sciences*, 94(10):5411–5416, 1997.

[5] K. Gish, G. L. Shulman, J. B. Sheehy, and H. W. Leibowitz. Reaction times to different spatial frequencies as a function of detectability. *Vision Research*, 26(5):745–747, 1986.

[6] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.

[7] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[8] T. Hromádka, M. R. DeWeese, and A. M. Zador. Sparse representation of sounds in the unanesthetized auditory cortex. *PLoS Biol*, 6(1):e16, 2008.

[9] T. Hu, C. Pehlevan, and D. B. Chklovskii. A hebbian/anti-hebbian network for online sparse dictionary learning derived from symmetric matrix factorization. *arXiv preprint arXiv:1503.00690*, 2015.

[10] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2146–2153. IEEE, 2009.

[11] K. Kavukcuoglu, P. Sermanet, Y.-L. Boureau, K. Gregor, M. Mathieu, and Y. LeCun. Learning convolutional feature hierarchies for visual recognition. In *Advances in neural information processing systems*, pages 1090–1098, 2010.

[12] J. K. Kim, P. Knag, T. Chen, and Z. Zhang. Efficient hardware architecture for sparse coding. *Signal Processing, IEEE Transactions on*, 62(16):4173–4186, 2014.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[14] Q. V. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. S. Corrado, J. Dean, and A. Y. Ng. Building high-level features using large scale unsupervised learning. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 81–88, 2012.

[15] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *Artificial Neural Networks and Machine Learning–ICANN 2011*, pages 52–59. Springer, 2011.

[16] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[17] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

[18] B. A. Olshausen and D. J. Field. Sparse coding of sensory inputs. *Current opinion in neurobiology*, 14(4):481–487, 2004.

[19] R. P. Rao and D. H. Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87, 1999.

[20] M. Rehn and F. T. Sommer. A network that uses few active neurones to code visual input predicts the diverse shapes of cortical receptive fields. *Journal of computational neuroscience*, 22(2):135–146, 2007.

[21] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature neuroscience*, 2(11):1019–1025, 1999.

[22] C. J. Rozell, D. H. Johnson, R. G. Baraniuk, and B. A. Olshausen. Sparse coding via thresholding and local competition in neural circuits. *Neural computation*, 20(10):2526–2563, 2008.

[23] P. F. Schultz, D. M. Paiton, W. Lu, and G. T. Kenyon. Replicating kernels with a short stride allows sparse reconstructions with fewer independent kernels. *arXiv preprint arXiv:1406.4205*, 2014.

[24] S. Shapero, A. S. Charles, C. J. Rozell, and P. Hasler. Low power sparse approximation on reconfigurable analog hardware. *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, 2(3):530–541, 2012.

[25] S. M. Sherman and R. Guillery. The role of the thalamus in the flow of information to the cortex. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 357(1428):1695–1708, 2002.

[26] E. P. Simoncelli and B. A. Olshausen. Natural image statistics and neural representation. *Annual review of neuroscience*, 24(1):1193–1216, 2001.

[27] D. C. Van Essen and C. H. Anderson. Information processing strategies and pathways in the primate visual system. *An introduction to neural and electronic networks*, 2:45–76, 1995.

[28] A. Vassilev and D. Mitov. Perception time and spatial frequency. *Vision research*, 16(1):89–92, 1976.

[29] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[30] W. Woods, J. Burger, and C. Teuscher. Synaptic weight states in a locally competitive algorithm for neuromorphic memristive hardware. *Nanotechnology, IEEE Transactions on*, pp, 2015.

[31] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2528–2535. IEEE, 2010.

[32] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2018–2025. IEEE, 2011.

[33] M. Zhu and C. J. Rozell. Visual nonclassical receptive field effects emerge from sparse coding in a dynamical system. *PLoS Comput Biol*, 9:e1003191, 2013.

[34] M. Zhu and C. J. Rozell. Modeling inhibitory interneurons in efficient sensory coding models. *PLOS Comput Biol*, 11(7):e1004353, 2015.

[35] J. Zylberberg and M. R. DeWeese. Sparse coding models can exhibit decreasing sparseness while learning sparse codes for natural images. *PLoS computational biology*, 9(8):e1003182, 2013.

[36] J. Zylberberg, J. T. Murphy, and M. R. DeWeese. A sparse coding model with synaptically local plasticity and spiking neurons can account for the diverse shapes of v1 simple cell receptive fields. *PLoS Comput Biol*, 7(10):e1002250, 2011.