Discriminative Recurrent Sparse Auto-Encoders

Jason Tyler Rolfe & Yann LeCun
Courant Institute of Mathematical Sciences, New York University
719 Broadway, 12th Floor
New York, NY 10003
{rolfe, yann}@cs.nyu.edu

Abstract

We present the discriminative recurrent sparse auto-encoder model, comprising a recurrent encoder of rectified linear units, unrolled for a fixed number of iterations, and connected to two linear decoders that reconstruct the input and predict its supervised classification. Training via backpropagation-through-time initially minimizes an unsupervised sparse reconstruction error; the loss function is then augmented with a discriminative term on the supervised classification. The depth implicit in the temporally-unrolled form allows the system to exhibit far more representational power, while keeping the number of trainable parameters fixed.

From an initially unstructured network the hidden units differentiate into categorical-units, each of which represents an input prototype with a well-defined class; and part-units representing deformations of these prototypes. The learned organization of the recurrent encoder is hierarchical: part-units are driven directly by the input, whereas the activity of categorical-units builds up over time through interactions with the part-units. Even using a small number of hidden units per layer, discriminative recurrent sparse auto-encoders achieve excellent performance on MNIST.

1 Introduction

Deep networks complement the hierarchical structure in natural data (Bengio, 2009). By breaking complex calculations into many steps, deep networks can gradually build up complicated decision boundaries or input transformations, facilitate the reuse of common substructure, and explicitly compare alternative interpretations of ambiguous input (Lee, Ekanadham, & Ng, 2008; Zeiler, Taylor, & Fergus, 2011). Leveraging these strengths, deep networks have facilitated significant advances in solving sensory problems like visual classification and speech recognition (Dahl, et al., 2012; Hinton, Osindero, & Teh, 2006; Hinton, et al., 2012).

Although deep networks have traditionally used independent parameters for each layer, they are equivalent to recurrent networks in which a disjoint set of units is active on each time step. The corresponding representations are sparse, and thus invite the incorporation of powerful techniques from sparse coding (Glorot, Bordes, & Bengio, 2011; Lee, Ekanadham, & Ng, 2008; Olshausen & Field, 1996, 1997; Ranzato, et al., 2006). Recurrence opens the possibility of sharing parameters between successive layers of a deep network.

This paper introduces the *Discriminative Recurrent Sparse Auto-Encoder* model (DrSAE), comprising a recurrent encoder of rectified linear units (ReLU; Coates & Ng, 2011; Glorot, Bordes, & Bengio, 2011; Jarrett, et al., 2009; Nair & Hinton, 2010; Salinas & Abbott, 1996), connected to two linear decoders that reconstruct the input and predict its supervised classification. The recurrent encoder is unrolled in time for a fixed number of iterations, with the input projecting to each resulting layer, and trained using backpropagation-through-time (Rumelhart, et al., 1986). Training initially minimizes an unsupervised sparse reconstruction error; the loss function is then augmented with a discriminative term on the supervised classification. In its temporally-unrolled form, the network can be seen as a deep network, with parameters shared between the hidden layers. The temporal depth allows the system to exhibit far more representational power, while keeping the number of trainable parameters fixed.

Interestingly, experiments show that DrSAE does not just discover more discriminative "parts" of the form conventionally produced by sparse coding. Rather, the hidden units spontaneously differentiate into two types: a small number of categorical-units and a larger number of part-units. The categorical-units have decoder bases that look like prototypes of the input classes. They are weakly influenced by the input and activate late in the dynamics as the result of interaction with the part-units. In contrast, the part-units are strongly influenced by the input, and encode small transformations through which the prototypes of categorical-units can be reshaped into the current input. Categorical-units compete with each other through mutual inhibition and cooperate with relevant part-units. This can be interpreted as a representation of the data manifold in which the categorical-units are points on the manifold, and the part-units are akin to tangent vectors along the manifold.

1.1 Prior work

The encoder architecture of DrSAE is modeled after the Iterative Shrinkage and Threshold Algorithm (ISTA), a proximal method for sparse coding (Chambolle, et al., 1998; Daubechies, Defrise, & De Mol, 2004). Gregor & LeCun (2010) showed that the sparse representations computed by ISTA can be efficiently approximated by a structurally similar encoder with a less restrictive, learned parameterization. Rather than learn to approximate a precomputed optimal sparse code, the LISTA autoencoders of Sprechmann, Bronstein, & Sapiro (2012a,b) are trained to directly minimize the sparse reconstruction loss function. DrSAE extends LISTA autoencoders with a non-negativity constraint, which converts the shrink nonlinearity of LISTA into a rectified linear operator; and introduces a unified classification loss, as previously used in conjunction with traditional sparse coders (Bradley & Bagnell, 2008; Mairal, et al., 2009; Mairal, Bach, & Ponce, 2012) and other autoencoders (Boureau, et al., 2010; Ranzato & Szummer, 2008).

DrSAEs resemble the structure of deep sparse rectifier neural networks (Glorot, Bordes, & Bengio, 2011), but differ in that the parameter matrices at each layer are tied (Bengio, Boulanger-Lewandowski, & Pascanu, 2012), the input projects to all layers, and the outputs are normalized. DrSAEs are also reminiscent of the recurrent neural networks investigated by Bengio & Gingras (1996), but use a different nonlinearity and a heavily regularized loss function. Finally, they are similar to the recurrent networks described by Seung (1998), but have recurrent connections amongst the hidden units, rather than between the hidden units and the input units, and introduce classification and sparsification losses.

2 Network architecture

In the following, we use lower-case bold letters to denote vectors, upper-case bold letters to denote matrices, superscripts to indicate iterative copies of a vector, and subscripts to index the columns (or rows, if explicitly specified by the context) of a matrix or (without boldface) the elements of a vector. We consider discriminative recurrent sparse auto-encoders (DrSAEs) of rectified linear units with the architecture shown in figure 1:

$$\mathbf{z}^{t+1} = \max\left(0, \mathbf{E} \cdot \mathbf{x} + \mathbf{S} \cdot \mathbf{z}^t - \mathbf{b}\right) \tag{1}$$

for $t = 1, \ldots, T$, where n-dimensional vector $\mathbf{z}^t$ is the activity of the hidden units at iteration t, m-dimensional vector $\mathbf{x}$ is the input, and $\mathbf{z}^{t=0} = 0$. Unlike traditional recurrent autoencoders (Bengio, Boulanger-Lewandowski, & Pascanu, 2012), the input projects to every iteration. We call the $n \times m$ parameter matrix $\mathbf{E}$ the encoding matrix, and the $n \times n$ parameter matrix $\mathbf{S}$ the explaining-away matrix. The n-element parameter vector $\mathbf{b}$ contains a bias term. The parameters also include the $m \times n$ decoding matrix $\mathbf{D}$ and the $l \times n$ classification matrix $\mathbf{C}$.

We pretrain DrSAEs using stochastic gradient descent on the unsupervised loss function

$$L^U = \frac{1}{2} \cdot \left|\left|\mathbf{x} - \mathbf{D} \cdot \mathbf{z}^T\right|\right|_2^2 + \lambda \cdot \left|\left|\mathbf{z}^T\right|\right|_1, \tag{2}$$

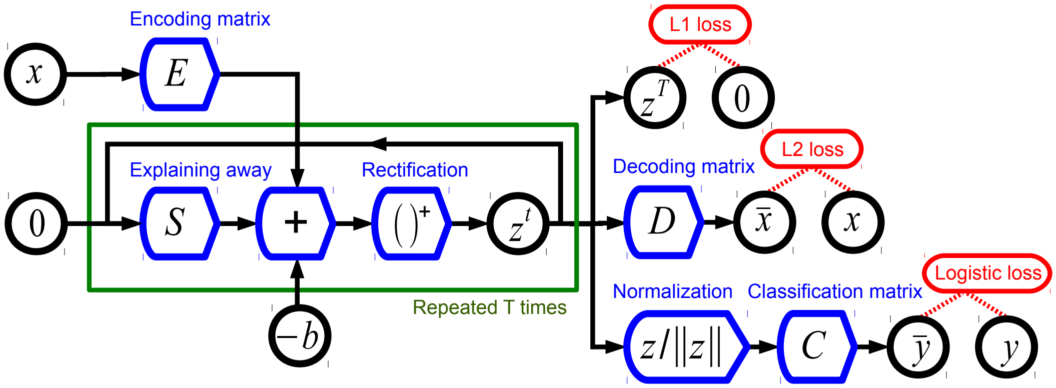

Figure 1: The discriminative recurrent sparse auto-encoder (DrSAE) architecture. $\mathbf{z}^t$ is the hidden representation after iteration t of T, and is initialized to $\mathbf{z}^0 = 0$; $\mathbf{x}$ is the input; and $\mathbf{y}$ is the supervised classification. Overbars denote approximations produced by the network, rather than the true input. $\mathbf{E}$, $\mathbf{S}$, $\mathbf{D}$, and $\mathbf{b}$ are learned parameters.

with the magnitude of the columns of $\mathbf{D}$ bounded by 1,[1] and the magnitude of the rows of $\mathbf{E}$ bounded by $\frac{1.25}{T}$.[2] We then add in the supervised classification loss function

$$L^S = \text{logistic}_y \left(\mathbf{C} \cdot \frac{\mathbf{z}^T}{||\mathbf{z}^T||} \right) , \tag{3}$$

where the multinomial logistic loss function is defined by

$$\text{logistic}_y(\mathbf{z}) = z_y - \log \left(\sum_i e^{z_i} \right) ,$$

and y is the index of the desired class.[3] Starting with the parameters learned by the unsupervised pretraining, we perform discriminative fine-tune by stochastic gradient descent on $L^U + L^S$, with the magnitude of the rows of $\mathbf{C}$ bounded by 5.[4] The learning rate of each matrix is scaled down by the number of times it is repeated in the network, and the learning rate of the classification matrix is scaled down by a factor of 5, to keep the effective learning rate consistent amongst the parameter matrices.

We train DrSAEs with $T = 11$ recurrent iterations (ten nontrivial passes through the explaining-away matrix $\mathbf{S}$)[5] and 400 hidden units on the MNIST dataset of 28×28 grayscale handwritten digits (LeCun, et al., 1998), with each input normalized to have ℓ_2 magnitude equal to 1. We use a training set of 50,000 elements, and a validation set of 10,000 elements to perform early-stopping. Encoding, decoding, and classification matrices learned via this procedure are depicted in figure 2.

The dynamics of equation 1 are inspired by the Learned Iterative Shrinkage and Thresholding Algorithm (LISTA) (Gregor & LeCun, 2010), an efficient approximation to the sparse coding Iterative Shrinkage and Threshold Algorithm (ISTA) (Chambolle, et al., 1998; Daubechies, Defrise, & De

[1] This sets the scale of $\mathbf{z}$; otherwise, the magnitude of $\mathbf{z}$ will shrink to zero and the magnitude of the columns of $\mathbf{D}$ will explode. This and all other such constraints are enforced by a projection after each SGD step.

[2] The size of each ISTA step must be sufficiently small to guarantee convergence. As the step size grows large, the input will be over-explained by multiple aligned hidden units, leading to extreme oscillations. This bound serves the same function as ℓ_2 weight regularization (Hinton, 2010). The particular value of the bound is heuristic, and was determined by an informal search of parameter space.

[3] Consistent with standard autoencoders but unlike traditional applications of backpropagation-through-time, the loss functions L^U and L^S only depend directly on the final iteration of the hidden units $\mathbf{z}^T$.

[4] As in the case of the encoder, this serves the same function as ℓ_2 weight regularization (Hinton, 2010). The particular value of the bound is heuristic, and was determined by an informal search of parameter space.

[5] The chosen number of recurrent iterations achieves a heuristic balance between representational power and computational expense. Experiments were conducted with $T \in \{2, 6, 11, 21\}$.

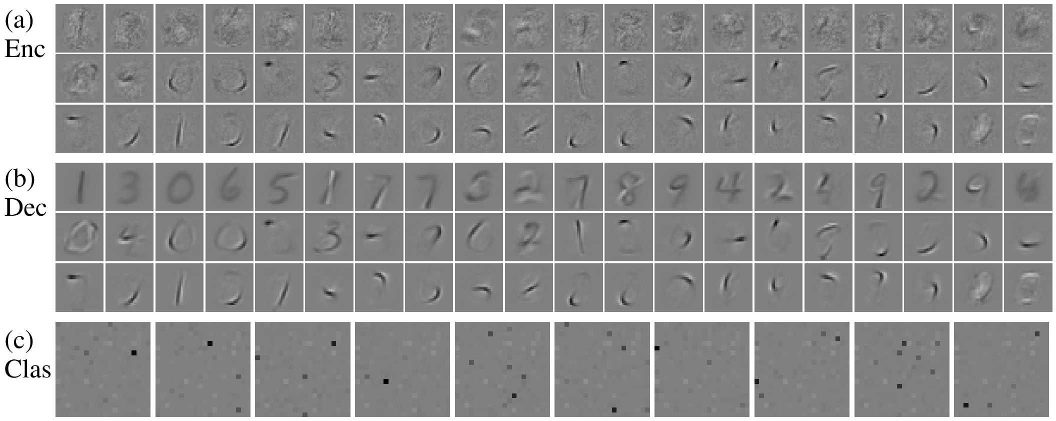

Figure 2: The hidden units differentiate into spatially localized part-units, which have well-aligned encoders and decoders; and global prototype categorical-units, which have poorly aligned encoders and decoders. A subset of the rows of encoding matrix $\mathbf{E}$ (a) and the columns of decoding matrix $\mathbf{D}$ (b), and all rows of the classification matrix $\mathbf{C}$ (c) after training. The first row of (a,b) shows the most categorical units; the last row contains the least categorical units; and the middle row evenly steps through the remaining units in order of categoricalness. Gray pixels denote connections with weight 0; darker pixels indicate more positive connections.

Mol, 2004). ISTA is an algorithm for minimizing the ℓ_1-regularized reconstruction loss function L^U of equation 2 with respect to $\mathbf{z}^T$. It is defined by the iterative step

$$\mathbf{z}^{t+1} = h_{\alpha \cdot \lambda} \left(\alpha \cdot \mathbf{D}^\top \cdot \mathbf{x} + \left(\mathbf{I} - \alpha \cdot \mathbf{D}^\top \cdot \mathbf{D} \right) \cdot \mathbf{z}^t \right) \,,$$

where $[h_\theta(\mathbf{x})]_i = \text{sign}(x_i) \cdot \max(0, |x_i| - \theta)$ and α is a small step-size parameter. With non-negative units, ISTA is equivalent to projected gradient descent of L^U of equation 2. As the number of iterations $T \to \infty$, a DrSAE defined by equation 1 becomes a non-negative version of ISTA if it satisfies the restrictions:

$$\mathbf{E} = \alpha \cdot \mathbf{D}^\top, \qquad \mathbf{S} = \mathbf{I} - \alpha \cdot \mathbf{D}^\top \cdot \mathbf{D}\,, \qquad b_i = \alpha \cdot \lambda\,, \qquad \text{and} \qquad z_i^t \geq 0\,, \tag{4}$$

where the positive scale factor α is less than the maximal eigenvalue of $\mathbf{D}^\top \cdot \mathbf{D}$, and $\mathbf{I}$ is the $n \times n$ identity matrix.

As in LISTA, but unlike ISTA, the encoding matrix $\mathbf{E}$ and explaining-away matrix $\mathbf{S}$ in a DrSAE are independent of the decoding matrix $\mathbf{D}$. Connections from the input to the hidden units, and recurrent connections between the hidden units, are all-to-all, so the network structure is agnostic to permutations of the input. DrSAEs can also be understood as deep, feedforward networks with the parameter matrices tied between the layers.

3 Analysis of the hidden unit representation

Discriminative fine-tuning naturally induces the hidden units of a DrSAE to differentiate into a hierarchy-like continuum. On one extreme are part-units, which perform an ISTA-like sparse coding computation; on the other are categorical-units, which use a sophisticated form of pooling to integrate over matching part-units, and implement winner-take-all dynamics amongst themselves. Converging lines of evidence indicate that these two groups use distinct computational mechanisms and serve different representational roles.

In the ISTA algorithm, each row of the encoding matrix $\mathbf{E}_i$ (which we sometimes call the encoder of unit i) is proportional to the corresponding column of the decoding matrix $\mathbf{D}_i$ (which we call the decoder of unit i), and each row $(\mathbf{S} - \mathbf{I})_i$ is proportional to $(\mathbf{D}_i)^\top \cdot \mathbf{D}$, as in equation 4. As a result, the angle between $\mathbf{E}_i$ and $\mathbf{D}_i$, and the angle between the rows of $\mathbf{S} - \mathbf{I}$ and $\mathbf{D}^\top \cdot \mathbf{D}$, are both simple measures of the degree to which a unit's dynamics follow the ISTA algorithm, and thus perform

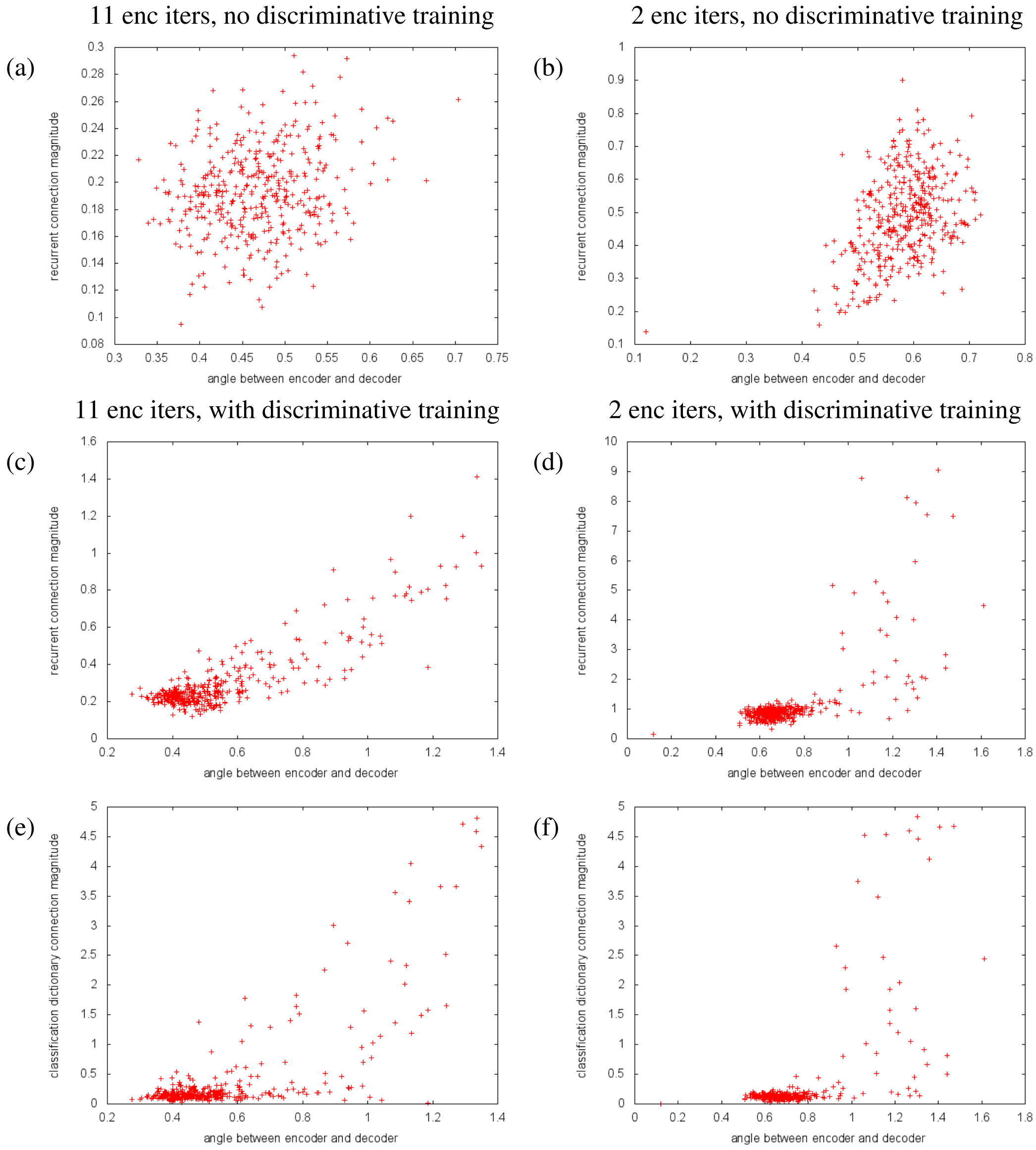

Figure 3: The hidden units differentiate into two populations after discriminative fine-tuning. The magnitude of row $(\mathbf{S}-\mathbf{I})_i$ (a,b,e) and $\mathbf{C}_i$ (c,d), versus the angle between encoder row and decoder column, for each unit from networks using 11 (a,c,e) and 2 (b,d,f) iterations. All plots are from discriminatively fine-tuned networks except (a,b), which are only subject to unsupervised pretraining. We call the dense cloud in the bottom-left part-units, and the tail extending to the top-right categorical-units.

sparse coding.[6] These quantities are equal to 0 in the case of perfect ISTA, and grow larger as the network diverges from ISTA. Of these two angles, the explaining-away matrix comparison is more difficult to interpret, since a distortion of any one unit's decoding column $\mathbf{D}_i$ will affect all rows of $\mathbf{D}^\top \cdot \mathbf{D}$, whereas the angle between the encoder row $\mathbf{E}_i$ and decoder column $\mathbf{D}_i$ only depends upon a single unit. For this reason, we use the angle between the encoder row and decoder column as a measure of the position of each unit on the part/categorical continuum.

[6] We always use $\mathbf{S}-\mathbf{I}$ when plotting recurrent connection strength, since it governs the perturbation of the otherwise stable hidden unit activations, as in projected gradient descent of L^U; i.e., ISTA.

Figure 3 plots, for each unit i, the magnitude of row $(\mathbf{S} - \mathbf{I})_i$ and column $\mathbf{C}_i$, versus the angle between row $\mathbf{E}_i$ and column $\mathbf{D}_i$. Before discriminative fine-tuning, there are no categorical-units; the angle between the encoder row and decoder column is small and the incoming recurrent connections are weak for all units, as in figure 3(a,b). After discriminative fine-tuning, there remains a dense cloud of points for which the angle between the encoder row and decoder column is very small, and the incoming recurrent and outgoing classification connections are weak. Abutting this is an extended tail of points that have a larger angle between the encoder row and decoder column, and stronger incoming recurrent and outgoing classification connections. We call units composing the dense cloud *part-units*, since they have ISTA-compatible connections, while we refer to those making up the extended tail as *categorical-units*, since they have strong connections to the classification output.[7] When trained on MNIST, part-units have localized, pen stroke-like decoders, as can be seen in the bottom rows of figure 2(a,b). Categorical-units, in contrast, tend to have whole-digit prototype-like decoders, as in the top rows of figure 2(a,b). Discriminative fine-tuning induces the differentiation of categorical-units regardless of the depth of the encoder.

3.1 Part-units

Examination of the relationship between the elements of $\mathbf{S} - \mathbf{I}$ and $\mathbf{D}^\top \cdot \mathbf{D}$ confirms that part-units with an encoder-decoder angle less than 0.5 radians abide by ISTA, and so perform sparse coding on the residual input after the categorical-unit prototypes are subtracted out. The prominent diagonals with matching slopes in figure 4(a,b), which plot the value of $S_{i,j} - \delta_{i,j}$ versus $\mathbf{D}_i \cdot \mathbf{D}_j$ for connections between part-units, and from categorical-units to part-units, respectively, demonstrate that part-units receive ISTA-consistent connections from all units. The fidelity of these connections to the ISTA ideal is not strongly dependent upon whether the afferent units are ISTA-compliant part-units, or ISTA-ignoring categorical-units. As a result, the part-units treat the categorical-units as if they were also participating in the reconstruction of the input, and only attempt to reconstruct the residual input not explained by the categorical-unit prototypes.

As can be seen in figure 4(c), the degree to which the encoder conforms to the ISTA algorithm is strongly correlated with the degree to which the explaining-away matrix matches the ISTA algorithm. Figure 5 shows the decoders associated with the strongest recurrent connections to three representative part-units. As expected, the decoders of these afferent units tend to be strongly aligned or anti-aligned with their target's decoder, and include both part-units and categorical-units.

3.2 Categorical-units

In contrast, the recurrent connections to categorical-units with an encoder-decoder angle greater than 0.7 radians are not strongly correlated with the values predicted by ISTA. Rather than analyzing connections to the categorical-units only based upon their destination, it is more informative to consider them organized by their source. Part-units are compatible with categorical-units of certain classes,[8] and not with others, as shown by figure 6(a). Part-units generally have positive connections to categorical-units with parallel prototypes, independent of offset, and negative connections to categorical-units with orthogonal prototypes, as shown in figure 7(a). This corresponds to a sophisticated form of pooling (Jarrett, et al., 2009), with a single categorical-unit drawing excitation from a large collection of parallel but not necessarily perfectly aligned part-units, as in figure 6(c). It is also suggestive of the standard Hubel and Wiesel model of complex cells in primary visual cortex (Hubel & Wiesel, 1962). ISTA would instead predict a connection proportional to the inner product, which is zero for orthogonal prototypes and negative for anti-aligned prototypes.

Part-units use sparse coding dynamics, and so are not disproportionately suppressed by categorical-units that represent any particular class. However, each part-unit is itself compatible with (i.e., has positive connections to) categorical-units of only a subset of the classes. As a result, the categorical-

[7] For the purpose of constructing figures characterizing the difference between part-units and categorical-units, we consider units with encoder-decoder angle less than 0.5 radians to be part-units, and units with encoder-decoder angle greater than 0.7 radians to be categorical-units. These thresholds are heuristic, and fail to reflect the continuum that exists between part- and categorical-units, but they facilitate analysis of the extremes.

[8] Categorical-units have strong, sparse classification matrix projections, as shown in figures 2(c) and 3(e,f), and can be identified with the output class to which they have the strongest projection.

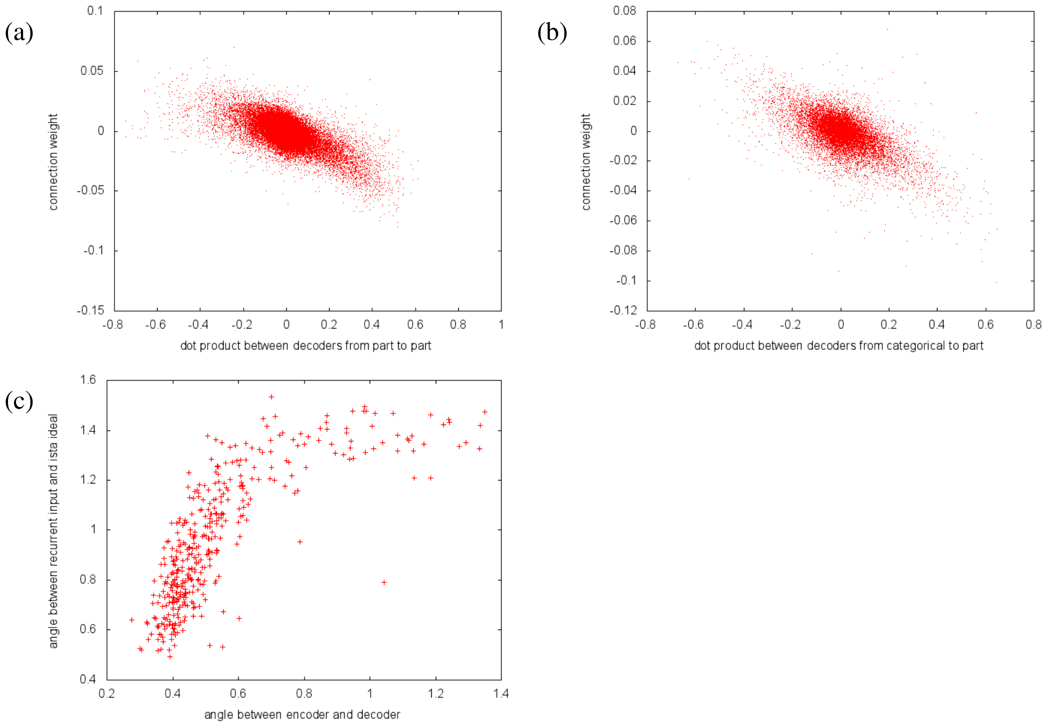

Figure 4: Part-units have connections consistent with ISTA. The actual connection weights $\mathbf{S} - \mathbf{I}$ versus the ISTA-predicted weights $\mathbf{D}^\top \cdot \mathbf{D}$, for connections from part-units to part-units (a) and categorical-units to part-units (b); and the angle between the rows of $\mathbf{S} - \mathbf{I}$ and the ISTA-ideal $\mathbf{D}^\top \cdot \mathbf{D}$ versus the angle between the encoder rows and decoder columns (c). Units are considered part-units if the angle between their encoder and decoder is less than 0.5 radians, and categorical-units if the angle between their encoder and decoder is greater than 0.7 radians.

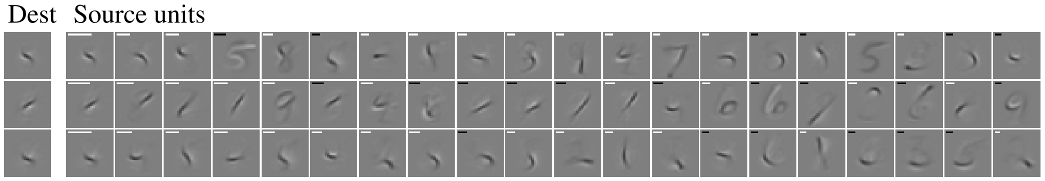

Figure 5: Part-units receive ISTA-compatible connections and thus perform sparse coding on the residual input after the contribution of the categorical-units is subtracted out. The decoders of the twenty units with the strongest explaining-away connections $|S_{i,j} - \delta_{i,j}|$ to three typical part-units, sorted by connection magnitude. The left-most column depicts the decoder of the recipient part-unit. The bars above the decoders in the remaining columns indicate the strength of the connections. Black bars are used for positive connections, and white bars for negative connections.

units and thus the class chosen are determined by the part-unit activations. In particular, only a subset of the possible deformations implemented by part-unit decoders are freely available for each prototype, since part-units with a strong negative connection to a categorical-unit will tend to silence it, and so cannot be used to transform the prototype of that categorical-unit.

Categorical-units implement winner-take-all-like dynamics amongst themselves, as shown in figure 6(b), with negative connections to most other categorical-units. Positive total self-connections $S_{i,i}$ facilitate the integration of inputs over time.

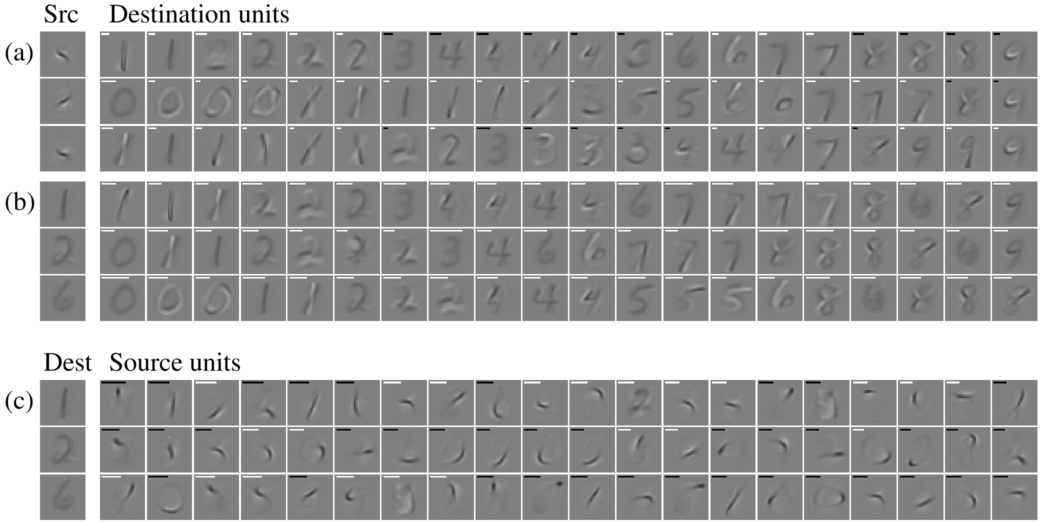

Figure 6: Categorical-units execute a sophisticated form of pooling over part-units, and have winner-take-all dynamics amongst themselves. The decoders of the categorical-units receiving the twenty strongest connections $|S_{i,j} - \delta_{i,j}|$ from representative part-units (a) and categorical-units (b), and the decoders of the part-units sending the twenty strongest projections to representative categorical-units (c). The connections are sorted first by the class of their destination, and then by the magnitude of the connection. The left-most column depicts the decoder of the source (a,b) or destination (c) unit. The bars above the decoders in the remaining columns indicate the strength of the connections. Black bars are used for positive connections, and white bars for negative connections.

When activated, the categorical-units make a much larger contribution to the reconstruction than any single part-unit, as can be seen in figure 7(b). Since, the projections from categorical-units to part-units are consistent with ISTA, the magnitude of the categorical-unit contribution to the reconstruction need not be tightly regulated. The part-units adjust accordingly to accommodate whatever residual is left by the categorical-units.

The units form a rough hierarchy, with part-units on the bottom and categorical-units on the top. Categorical-units receive strong recurrent connections, as shown in figure 3(c,d) implying that their activity is more determined by other hidden units and less by the input (since the magnitude of the input connections is bounded), and thus they are higher in the hierarchy. As shown in figure 7(c), part-units receive most of their input from other part-units; categorical-units receive a larger fraction of their input from other categorical-units. Whereas part-units have well-structured encoders and are generally activated directly by the input on the first iteration, categorical-units are more likely to first achieve a non-zero activation on the second iteration, as shown in figure 7(d), suggesting that they require stimulation from part-units. The immediate response of part-units in contrast to the gradual refinement of categorical-units is apparent in figure 8, which shows the optimal decoding matrix for selected units, inferred from their observed activity at each iteration.

4 Performance

The comparison of MNIST classification performance in table 1 demonstrates the power of the hierarchical representation learned by DrSAEs. Rather than learn to minimize the sum of equations 2 and 3, Gregor & LeCun (2010) train the LISTA encoder to approximate the code generated by a traditional sparse coder. While they do not report classification performance using LISTA, Gregor and LeCun do evaluate MNIST classification error using the related learned coordinate descent algorithm. Sprechmann, Bronstein, & Sapiro (2012a,b) extend this approach by training a LISTA auto-encoder to reconstruct the input directly, using loss functions similar to equation 2. Although they identify the possibility of using regularization dependent upon supervised information, Sprechmann and colleagues do not consider a parameterized classifier operating on a common hidden

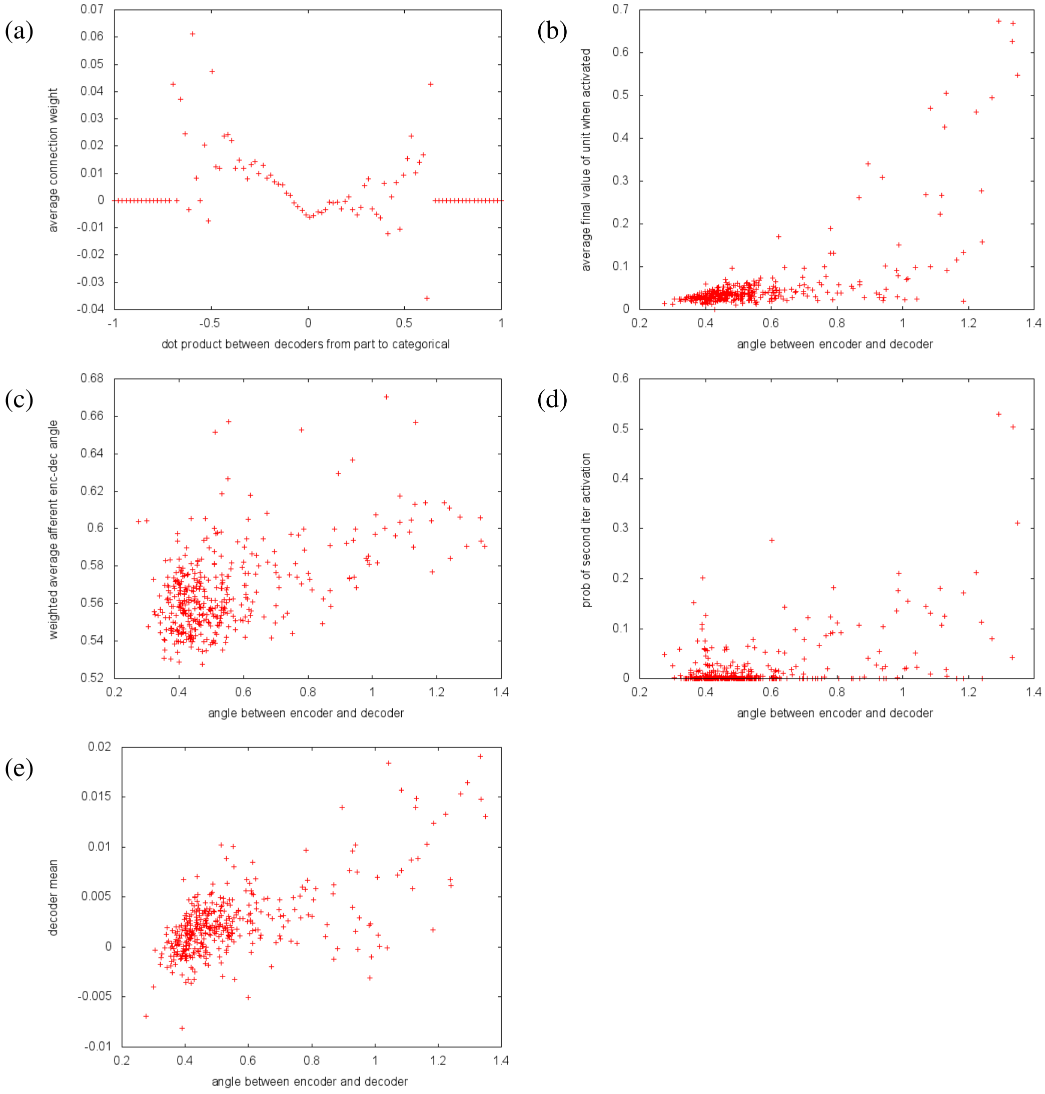

Figure 7: Statistics of connections indicate the presence of a rough hierarchy, with categorical-units on the top integrating over part-units on the bottom. Average explaining-away connection weight S_{ij}, binned by alignment between decoders, for connections from part-units to categorical-units (a). If no units fall in a given bin, the average is set to zero. Average final value of a unit $z_i^{t=T}$, given that $z_i^{t=T} > 0$, versus the angle between the encoder row E_i and decoder column D_i (b). Average angle between encoder row E_j and decoder column D_j of afferents to unit i, weighted by the strength of the connection to unit i, versus the angle between encoder row E_i and decoder column D_i (c). Probability that $z_i^1 = 0$ and $z_i^2 > 0$, versus the angle between the encoder row E_i and decoder column D_i (d). Average value of the decoder column $\overline{D_i}$ versus the angle between the encoder row E_i and the decoder column D_i (e).

representation. Instead, they train a separate encoder for each class, and classify each input based upon the encoder with the lowest sparse coding error. DrSAEs significantly outperform these other techniques based upon a LISTA encoder.

DrSAEs also perform well compared to other techniques using encoders related to LISTA. Deep sparse rectifier neural networks (Glorot, Bordes, & Bengio, 2011) combine discriminative training with an encoder similar to LISTA, but do not tie the parameters between the layers and only allow the input to project to the first layer. Differentiable sparse coding (Bradley & Bagnell, 2008) and

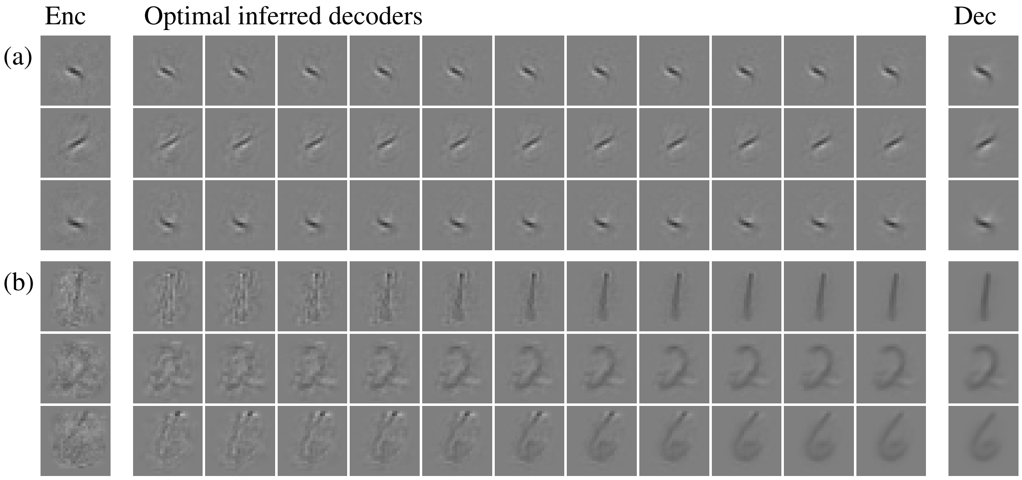

Figure 8: Part-units (a) respond to the input quickly, while the activity of categorical-units (b) refines slowly. Columns of the optimal decoding matrices $\mathbf{D}^t$ minimizing the input reconstruction error $||\mathbf{x} - \mathbf{D}^t \cdot \mathbf{z}^t||_2^2$ from the hidden representation $\mathbf{z}^t$ for $t = 1, \ldots, T$. The first and last columns show the corresponding encoder and decoder for the chosen representative units. Intermediate columns represent successive iterations t.

LISTA auto-encoder, $10 \times \left(289\text{-}100^5\right)$ (Sprechmann, Bronstein, & Sapiro, 2012a)	3.76	(5.98 with 289 hidden units)
Learned coordinate descent, 784-784^{50}-10 (Gregor & LeCun, 2010)	2.29	
Differentiable sparse coding, 180-256^*-10 (Bradley & Bagnell, 2008)	1.30	
Deep sparse rectifier neural network 784-1000-1000-1000-10 (Glorot, Bordes, & Bengio, 2011)	1.20	(1.16 with tanh nonlinearity)
Deep belief network, 784-500-500-2000-10 (Hinton, et al., 2012)	1.18	(0.92 with dropout)
Discriminative recurrent sparse auto-encoder **784-$\mathbf{400^{11}}$-10**	**1.08**	(1.21 with 200 hidden units)
Supervised dictionary learning, $45 \times (784\text{-}24^*)$ to $45 \times (784\text{-}96^*)$ (Mairal, et al., 2009)	1.05	(3.56 without contrastive loss)

Table 1: MNIST classification error rate (%) for pixel-permutation-agnostic encoders without boosting-like augmentations. The first column indicates the size of each layer in the specified encoder, separated by hyphens. Exponents specify the number of recurrent iterations; asterisks denote repetition to convergence. $10 \times (\cdots)$ indicates that a separate encoder is trained for each input class; $45 \times (\cdots)$ indicates that a separate encoder is trained for each pairwise binary classification problem. Further performance improvements have been reported with regularization techniques such as dropout, architectures that enforce translation-invariance, and datasets augmented by deformations, as discussed in the main text.

supervised dictionary learning (Mairal, et al., 2009) also train discriminatively, but effectively use an infinite-depth ISTA-like encoder, and are thus much less computationally efficient than DrSAEs. Supervised dictionary learning achieves performance statistically indistinguishable from DrSAEs using a contrastive loss function. A similar technique achieves MNIST classification error as low as 0.54% when the dataset is augmented with shifted copies of the inputs (Mairal, Bach, & Ponce, 2012).

Additional regularizations and boosting-like techniques can further improve performance of networks with LISTA-like encoders. Recent examples include dropout, which trains and then averages over a large set of random subnetworks formed by removing a constant fraction of the hidden units from the original network (Goodfellow, et al., 2013; Hinton, et al., 2012). Deep belief networks and deep Boltzmann machines fine-tuned with dropout are the current state-of-the-art for pixel-permutation-agnostic handwritten digit recognition (Hinton, et al., 2012), and can achieve MNIST classification error as low as 0.79% with a carefully tuned network structure and multi-step training procedure. Deep convex networks, which iteratively refine the classification by successively training a stack of classifiers, with the output of the $i - 1$st classifier provided as input to the ith classifier, can achieve an MNIST error of 0.83% (Deng & Yu, 2011). Regularizing by explicit modeling of the data manifold, and then minimizing the square of the Jacobian of the output along the tangent bundle around the training datapoints, can reduce MNIST error to 0.81% (Rifai, et al., 2011). Further performance improvements are possible if translation invariance is built directly into the network via a convolutional architecture, and deformations of the inputs are included in the training set (LeCun, et al., 1998), yielding error as low as 0.23% (Ciresan, Meier, & Schmidhuber, 2012). These regularizations and augmentations are potentially compatible with DrSAE, but we defer their exploration to future work.

Recurrence is essential to the performance of DrSAEs. If the number of recurrent iterations is decreased from eleven to two, MNIST classification error in a network with 400 hidden units increases from 1.08% to 1.32%. With only 200 hidden units, MNIST classification error increases from 1.21% to 1.49%, although the hidden units still differentiate into part-units and categorical-units, as shown in figure 3(d,f).

5 Discussion

It is widely believed that natural stimuli, such as images and sounds, fall near a low-dimensional manifold within a higher-dimensional space (the *manifold hypothesis*) (Bengio, Courville, & Vincent, 2012; Lee, Pedersen, & Mumford, 2003; Olshausen & Field, 2004). The low-dimensional data manifold provides an intuitively compelling and empirically effective basis for classification (Rifai, et al., 2011; Simard, LeCun, & Denker, 1993; Simard, et al., 1998). The continuous deformations that define the data manifold usually preserve identity, whereas even relatively small invalid transformations may change the class of a stimulus. For instance, the various handwritten renditions of the digit 3 in in the last column of figure 9(c) barely overlap, and so the Euclidean distance between them in pixel space is greater than that to the nearest 8 formed by closing both loops. Nevertheless, smooth deformations of one 3 into another correspond to relatively short trajectories along the data manifold,[9] whereas the transformation of a 3 into an 8 requires a much longer path within the data manifold. A prohibitive amount of data is required to fully characterize the data manifold (Narayanan & Mitter, 2010), so it is often approximated by the set of linear submanifolds tangent to the data manifold at the observed datapoints, known as the *tangent spaces* (Ekanadham, Tranchina, & Simoncelli, 2011; Rifai, et al., 2011; Simard, et al., 1998). DrSAEs naturally and efficiently form a tangent space-like representation, consisting of a point on the data manifold indicated by the categorical-units, and a shift within the tangent space specified by the part-units.

Before discriminative fine-tuning, DrSAEs perform a traditional part-based decomposition, familiar from sparse coding, as shown in figure 9(a). The decoding matrix columns are class-independent, local pen strokes, and many units make a comparable, small contribution to the reconstruction. After discriminative fine-tuning, the hidden units differentiate into sparse coding local part-units, and global prototype categorical-units that integrate over them. As shown in figure 9(b,c), the input is decomposed into a prototype, corresponding to a point on the data manifold; and a set of deformations from this prototype along the data manifold, corresponding to shifts within the tangent space. The same prototype can be used for very different inputs, as demonstrated in figure 9(c), since the space of deformations is rich enough to encompass diverse transformations without moving off the data

[9] In particular, figure 9(c) shows how each input can be produced by identity-preserving deformations from a common prototype, using the tangent space decomposition produced by our network.

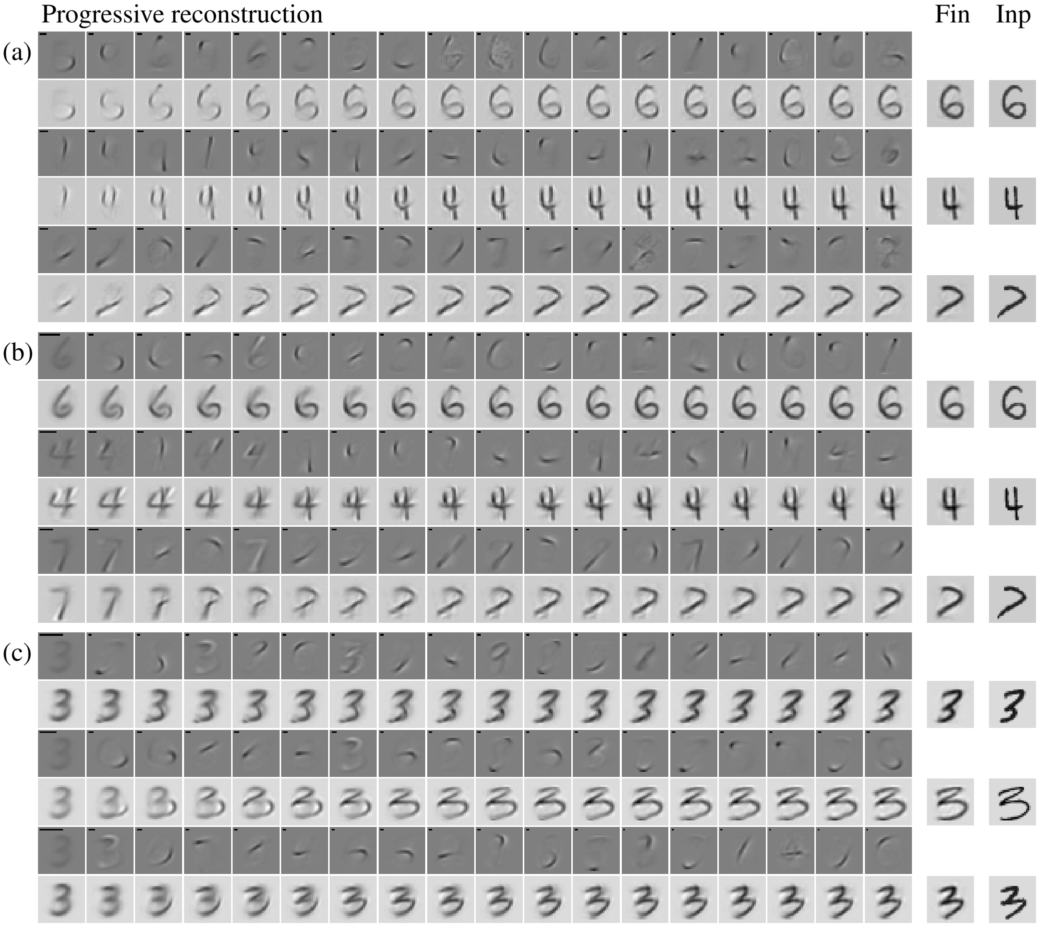

Figure 9: Discriminative recurrent sparse auto-encoders decompose the input into a prototype and deformations along the data manifold. The progressive reconstruction of selected inputs by the hidden representation before (a) or after (b,c) discriminative fine-tuning. The columns from left to right depict either the components of the reconstruction (top row of each pair), or the partial reconstruction induced by the first n parts (bottom row of each pair). Parts are added to the reconstruction in order of decreasing contribution magnitude; smoother transformations are possible with an optimized sequence. The last two columns show the final reconstruction with all parts (Fin), and the original input (Inp). Bars above the decoding matrix columns indicate the scale factor/hidden unit activity associated with the column.

manifold. Even when the prototype is very different from the input, all steps along the reconstruction trajectories in figure 9(b,c) are recognizable as members of the same class.

The prototypes learned by the categorical-units for each class are not simply the average over the elements of the class, as depicted in figure 10. Each class includes many possible input variations, so its average is blurry. The prototypes, in contrast, are sharp, and look like representative elements of the appropriate class. Many categorical-units are available for each class, as shown in figure 6. Not all categorical-units correspond to full prototypes; some capture global transformations of a prototype, such as rotations (Simard, et al., 1998).

Consistent with prototypes for the non-negative MNIST inputs, the decoding matrix columns of the categorical-units are generally positive, as shown in figure 7(e). In contrast, the decoders of the part-units are approximately mean-zero and so cannot serve as prototypes themselves. Rather, they shift and transform prototypes, moving activation from one region in the image to another, as demonstrated in figure 9(b,c).

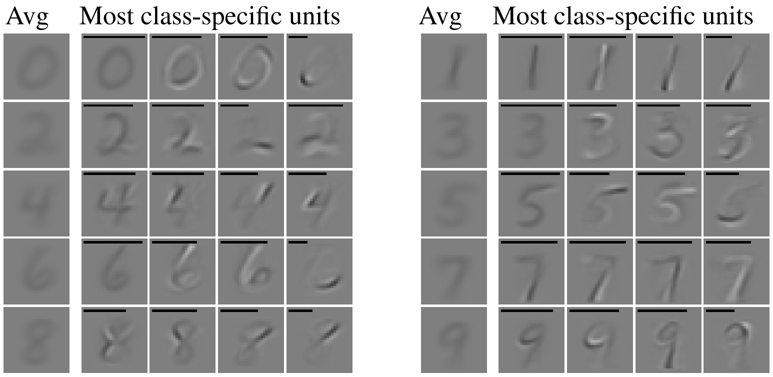

Figure 10: The prototypes learned by categorical-units resemble representative instances of the appropriate class, and are sharper than the average over all members of the class in the dataset. The left-most column in each group depicts the average over all elements of each of the ten MNIST digit classes. The other columns show the decoders of the associated units with the largest-magnitude columns in the classification matrix **C**. Bars above the decoders indicate the angle between the encoder and the decoder for the displayed unit. The most prototypical unit always makes the strongest contribution to the classification, and has a large (but not necessarily the largest) angle between its encoder and decoder. Some units that make large contributions to the classification represent global transformations, such as rotations, of a prototype (Simard, et al., 1998).

Discrepancies between the prototype and the input due to transformations along the data manifold are explained by class-consistent part-units, and only serve to further activate the categorical-units of that class, as in figure 6(a,c). Discrepancies between the prototype and the input due to deformations orthogonal to the data manifold are explained by class-incompatible part-units, and serve to suppress the categorical-units of that class, both directly and via activation of incompatible categorical-units.

If the wrong prototype is turned on, the residual input will generally contain substantial unexplained components. Part-units obey ISTA-like dynamics and thus function as a sparse coder on the residual input, so part-units that match the unexplained components of the input will be activated. These part-units will have positive connections to categorical-units with compatible prototypes, and so will tend to activate categorical-units associated with the true class (so long as the unexplained components of the input are diagnostic). The spuriously activated categorical-unit will not be able to sustain its activity, since few compatible part-units will be required to capture the residual input.

The classification approach used by DrSAEs is different from one based upon a traditional sparse coding decomposition: it projects into the space of deviations from a prototype, which is not the same as the space of prototype-free parts, as is clear from figure 9(a,b). For instance, a 5 can easily be constructed using the parts of a 6, making it difficult to distinguish the two. Indeed, the first seven progressive reconstruction steps of the 6 in figure 9(a) could just as easily be used to produce a 5. However, starting from a 6 prototype, the parts required to break the bottom loop are outside the data manifold of the 6 class, and so will tend to change the active prototype.

DrSAEs naturally learn a hierarchical representation within a recurrent network, thereby implementing a deep network with parameter sharing between the layers.

References

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, *2*(1), 1–127.

Bengio, Y., Boulanger-Lewandowski, N., & Pascanu, R. (2012). Advances in optimizing recurrent networks. arXiv:1212.0901v2 [cs.LG]

Bengio, Y., Courville, A., & Vincent, P. (2012). Representation learning: A review and new perspectives. arXiv:1206.5538 [cs.LG]

Bengio, Y., & Gingras, F. (1996). Recurrent neural networks for missing or asynchronous data. In D. Touretzky, M. Mozer, & M. Hasselmo (Eds.) *Advances in Neural Information Processing Systems (NIPS 8)* (pp. 395–401).

Bradley, D. M., & Bagnell, J. A. (2008). Differentiable sparse coding. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.) *Advances in Neural Information Processing Systems (NIPS 21)* (pp. 113–120).

Boureau, Y., Bach, F., LeCun, L., & Ponce, J. (2010). Learning mid-level features for recognition In *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010).*

Chambolle, A., De Vore, R. A., Lee, N. Y., & Lucier, B. J. (1998). Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Transactions on Image Processing*, *7*(3), 319–335.

Ciresan, D., Meier, U., & Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In *Proceedings of the 25th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012)* (pp. 3642–3649).

Coates, A., & Ng, A. Y. (2011). The importance of encoding versus training with sparse coding and vector quantization L. Getoor & T. Scheffer (Eds.) *Proceedings of the 28th International Conference on Machine Learning (ICML 2011)* (pp. 921–928).

Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, *20*(1), 30–42.

Daubechies, I., Defrise, M., & De Mol, C. (2004). An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, *57*(11), 1413–1457.

Ekanadham, C., Tranchina, D., & Simoncelli, E. P. (2011). Recovery of sparse translation-invariant signals with continuous basis pursuit. *IEEE Transactions on Signal Processing*, *59*(10), 4735–4744.

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. In G. Gordon, D. Dunson, & M. Dudik (Eds.) *JMLR W&CP: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011)* (pp. 315–323).

Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., & Bengio, Y. (2013). Maxout networks arXiv:1302.4389v3 [stat.ML]

Gregor, K., & LeCun, Y. (2010). Learning fast approximations of sparse coding. In J. Fürnkranz & T. Joachims (Eds.) *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)* (pp. 399–406).

Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*(7), 1527–1554.

Hinton, G. (2010). A practical guide to training restricted Boltzmann machines (UTML TR 2010-003, version 1). Toronto, Canada: University of Toronto, Department of Computer Science.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors arXiv:1207.0580v1 [cs.NE]

Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160(1), 106–154.

Jarrett, K., Kavukcuoglu, K., Ranzato, M. A., & LeCun, Y. (2009). *What is the best multi-stage architecture for object recognition?* In *Proceedings of the 12th International Conference on Computer Vision (ICCV 2009)* (pp. 2146–2153).

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Lee, A. B., Pedersen, K. S., & Mumford, D. (2003). The nonlinear statistics of high-contrast patches in natural images. *International Journal of Computer Vision*, *54*(1), 83–103.

Lee, H., Ekanadham, C., & Ng, A. (2008). Sparse deep belief net model for visual area V2. In J. C. Platt, D. Koller, Y. Singer & S. Roweis (Eds.) *Advances in Neural Information Processing Systems (NIPS 20)*, (pp. 873–880).

Mairal, J., Bach, F., Ponce, J., Sapiro, G., & Zisserman, A. (2009). Supervised dictionary learning. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.) *Advances in Neural Information Processing Systems (NIPS 21)* (pp. 1033–1040).

Mairal, J., Bach, F., & Ponce, J. (2012). Task-driven dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(4), 791–804.

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In J. Fürnkranz & T. Joachims (Eds.) *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)* (pp. 807-814).

Narayanan, H. & MItter, S. (2010). Sample complexity of testing the manifold hypothesis. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, & A. Culotta (Eds.) *Advances in Neural Information Processing Systems (NIPS 23)* (pp. 1786–1794).

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

Olshausen, B. A., & Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by VI? *Vision Research*, *37*(23), 3311–3326.

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

Ranzato M., Poultney, C., Chopra, S., & LeCun, Y. (2006). Efficient learning of sparse representations with an energy-based model. In B. Schölkopf, J. Platt, & T. Hoffman (Eds.) *Advances in Neural Information Processing Systems (NIPS 19)*, (pp. 1137–1144).

Ranzato, M., & Szummer, M. (2008). Semi-supervised learning of compact document representations with deep networks In A. McCallum & S. Roweis (Eds.), *Proceedings of the 25th Annual International Conference on Machine Learning (ICML 2008)* (pp. 792–799).

Rifai, S., Dauphin, Y., Vincent, P., Bengio, Y., & Muller, X. (2011). The manifold tangent classifier. In J. Shawe-Taylor, R. S. Zemel, P. Bartlett, F. C. N. Pereira, & K. Q. Weinberger (Eds.) *Advances in Neural Information Processing Systems (NIPS 24)* (pp. 2294–2302).

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, and the PDP Research Group (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Vol. 1. Foundations* (pp. 318–362). Cambridge, MA: MIT Press.

Salinas, E., & Abbott, L. F. (1996). A model of multiplicative neural responses in parietal cortex. *Proceedings of the National Academy of Sciences of the United States of America*, *93*(21), 11956–11961.

Seung, H. S. (1998). Learning continuous attractors in recurrent networks. In M. I. Jordan, M. J. Kearns, & S. A. Solla (Eds.) *Advances in Neural Information Processing Systems (NIPS 10)* (pp. 654–660).

Simard, P., LeCun, Y., & Denker, J. S. (1993). Efficient pattern recognition using a new transformation distance. In S. J. Hanson, J. D. Cowan, & C. L. Giles (Eds.) *Advances in Neural Information Processing Systems (NIPS 5)* (pp. 50–58).

Simard, P., LeCun, Y., Denker, J., & Victorri, B. (1998). Transformation invariance in pattern recognition: Tangent distance and tangent propagation. In G. Orr, & K. Muller (Eds.), *Neural networks: Tricks of the trade*. Berlin: Springer.

Sprechmann, P., Bronstein, A., & Sapiro, G. (2012). Learning efficient structured sparse models. In J. Langford & J. Pineau (Eds.) *Proceedings of the 29th International Conference on Machine Learning (ICML 12)* (pp. 615–622).

Sprechmann, P., Bronstein, A., & Sapiro, G. (2012). Learning efficient sparse and low rank models. arXiv:1212.3631 [cs.LG]

Deng, L. & Yu, D. (2011). Deep convex net: A scalable architecture for speech pattern classification. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association (INTERSPEECH 2011)* (pp. 2285-2288).

Zeiler, M. D., Taylor, G. W., & Fergus, R. (2011). Adaptive deconvolutional networks for mid and high level feature learning. In *Proceedings of the 13th International Conference on Computer Vision (ICCV 2011)* (pp. 2018–2025).