

---

# A Theory of Generative ConvNet

---

Jianwen Xie <sup>†</sup>

Yang Lu <sup>†</sup>

Song-Chun Zhu

Ying Nian Wu

Department of Statistics, University of California, Los Angeles, CA, USA

JIANWEN@UCLA.EDU

YANGLV@UCLA.EDU

SCZHU@STAT.UCLA.EDU

YWU@STAT.UCLA.EDU

## Abstract

We show that a generative random field model, which we call generative ConvNet, can be derived from the commonly used discriminative ConvNet, by assuming a ConvNet for multi-category classification and assuming one of the categories is a base category generated by a reference distribution. If we further assume that the non-linearity in the ConvNet is Rectified Linear Unit (ReLU) and the reference distribution is Gaussian white noise, then we obtain a generative ConvNet model that is unique among energy-based models: The model is piecewise Gaussian, and the means of the Gaussian pieces are defined by an auto-encoder, where the filters in the bottom-up encoding become the basis functions in the top-down decoding, and the binary activation variables detected by the filters in the bottom-up convolution process become the coefficients of the basis functions in the top-down deconvolution process. The Langevin dynamics for sampling the generative ConvNet is driven by the reconstruction error of this auto-encoder. The contrastive divergence learning of the generative ConvNet reconstructs the training images by the auto-encoder. The maximum likelihood learning algorithm can synthesize realistic natural image patterns.

## 1. Introduction

The convolutional neural network (ConvNet or CNN) (Le-Cun et al., 1998; Krizhevsky et al., 2012) has proven to be a tremendously successful discriminative or predictive learning machine. Can the discriminative ConvNet be turned

into a generative model and an unsupervised learning machine? It would be highly desirable if this can be achieved because generative models and unsupervised learning can be very useful when the training datasets are small or the labeled data are scarce. It would also be extremely satisfying from a conceptual point of view if both discriminative classifier and generative model, and both supervised learning and unsupervised learning, can be treated within a unified framework for ConvNet.

In this conceptual paper, we show that a generative random field model, which we call generative ConvNet, can be derived from the commonly used discriminative ConvNet, by assuming a ConvNet for multi-category classification and assuming one of the categories is a base category generated by a reference distribution. The model is in the form of exponential tilting of the reference distribution, where the exponential tilting is defined by the ConvNet scoring function. If we further assume that the non-linearity in the ConvNet is Rectified Linear Unit (ReLU) (Krizhevsky et al., 2012) and that the reference distribution is Gaussian white noise, then we obtain a generative ConvNet model that is unique among energy-based models: The model is piecewise Gaussian, and the means of the Gaussian pieces are defined by an auto-encoder, where the filters in the bottom-up encoding become the basis functions in the top-down decoding, and the binary activation variables detected by the filters in the bottom-up convolution process become the coefficients of the basis functions in the top-down deconvolution process (Zeiler & Fergus, 2014). The Langevin dynamics for sampling the generative ConvNet is driven by the reconstruction error of this auto-encoder. The contrastive divergence learning (Hinton, 2002) of the generative ConvNet reconstructs the training images by the auto-encoder. The maximum likelihood learning algorithm can synthesize realistic natural image patterns.

The main purpose of our paper is to explore the properties of the generative ConvNet, such as being piecewise Gaussian with auto-encoding means, where the filters in the bottom-up operation take up a new role as the basis

---

<sup>†</sup> Equal contributions.

functions in the top-down representation. Such an internal representational structure is essentially unique among energy-based models. They are the results of the marriage between the piecewise linear structure of the ReLU ConvNet and the Gaussian white noise reference distribution. The auto-encoder we elucidate is a harmonious fusion of bottom-up convolution and top-down deconvolution.

The reason we choose Gaussian white noise as the reference distribution is that it is the maximum entropy distribution with given marginal variance. Thus it is the most featureless distribution. Another justification for the Gaussian white noise distribution is that it is the limiting distribution if we zoom out any stochastic texture pattern, due to the central limit theorem (Wu et al., 2008). The ConvNet seeks to search for non-Gaussian features in order to recover the non-Gaussian distribution before the central limit theorem takes effect. Without the Gaussian reference distribution that contributes the  $\ell_2$  norm term to the energy function, we will not have the auto-encoding form in the internal representation of the model. In fact, without Gaussian reference distribution, the density of the model is not even integrable. The Gaussian reference distribution is also crucial for the Langevin dynamics to be driven by the reconstruction error.

The ReLU is the most commonly used non-linearity in modern ConvNet (Krizhevsky et al., 2012). It makes the ConvNet scoring function piecewise linear (Montufar et al., 2014). Without the piecewise linearity of the ConvNet, we will not have the piecewise Gaussian form of the model. The ReLU is the source of the binary activation variables because the ReLU function can be written as  $\max(0, r) = 1(r > 0) * r$ , where  $1(r > 0)$  is the binary variable indicating whether  $r > 0$  or not. The binary variable can also be derived from the derivative of the ReLU function. The piecewise linearity is also crucial for the exact equivalence between the gradient of the contrastive divergence learning and the gradient of the auto-encoder reconstruction error.

## 2. Related Work

The model in the form of exponential tilting of a reference distribution where the exponential tilting is defined by ConvNet was first proposed by (Dai et al., 2015). They did not study the internal representational structure of the model. (Lu et al., 2016) proposed to learn the FRAME (Filters, Random field, And Maximum Entropy) models (Zhu et al., 1997) based on the pre-learned filters of existing ConvNets. They did not learn the models from scratch. The hierarchical energy-based models (LeCun et al., 2006) were studied by the pioneering work of (Hinton et al., 2006b) and (Ngiam et al., 2011). Their models do not correspond directly to modern ConvNet and do not posses the internal representational structure of the generative ConvNet.

The generative ConvNet model can be viewed as a hierarchical version of the FRAME model (Zhu et al., 1997), as well as the Product of Experts (Hinton, 2002; Teh et al., 2003) and Field of Experts (Roth & Black, 2005) models. These models do not have explicit Gaussian white noise reference distribution. Thus they do not have the internal auto-encoding representation, and the filters in these models do not play the role of basis functions. In fact, a main motivation for this paper is to reconcile the FRAME model (Zhu et al., 1997), where the Gabor wavelets play the role of bottom-up filters, and the Olshausen-Field model (Olshausen & Field, 1997), where the wavelets play the role of top-down basis functions. The generative ConvNet may be seen as one step towards achieving this goal. See also (Xie et al., 2014).

The relationship between energy-based model with latent variables and auto-encoder was discovered by (Vincent, 2011) and (Swersky et al., 2011) via the score matching estimator (Hyvärinen, 2005). This connection requires that the free energy can be calculated analytically, i.e., the latent variables can be integrated out analytically. This is in general not the case for deep energy-based models with multiple layers of latent variables, such as deep Boltzmann machine with two layers of hidden units (Salakhutdinov & Hinton, 2009). In this case, one cannot obtain an explicit auto-encoder. In fact, for such models, the inference of the latent variables is in general intractable. In generative ConvNet, the multiple layers of binary activation variables come from the ReLU units, and the means of the Gaussian pieces are always defined by an explicit hierarchical auto-encoder.

Compared to hierarchical models with explicit binary latent variables such as those based on the Boltzmann machine (Hinton et al., 2006a; Salakhutdinov & Hinton, 2009; Lee et al., 2009), the generative ConvNet is directly derived from the discriminative ConvNet. Our work seems to suggest that in searching for generative models and unsupervised learning machines, we need to look no further beyond the ConvNet.

## 3. Generative ConvNet

To fix notation, let  $\mathbf{I}(x)$  be an image defined on the square (or rectangular) image domain  $\mathcal{D}$ , where  $x = (x_1, x_2)$  indexes the coordinates of pixels. We can treat  $\mathbf{I}(x)$  as a two-dimensional function defined on  $\mathcal{D}$ . We can also treat  $\mathbf{I}$  as a vector if we fix an ordering for the pixels. For a filter  $F$ , let  $F * \mathbf{I}$  denote the filtered image or feature map, and let  $[F * \mathbf{I}](x)$  denote the filter response or feature at position  $x$ .

A ConvNet is a composition of multiple layers of linear filtering and element-wise non-linear transformation as ex-

pressed by the following recursive formula:

$$[F_k^{(l)} * \mathbf{I}](x) = h \left( \sum_{i=1}^{N_{l-1}} \sum_{y \in \mathcal{S}_l} w_{i,y}^{(l,k)} [F_i^{(l-1)} * \mathbf{I}](x+y) + b_{l,k} \right) \quad (1)$$

where  $l \in \{1, 2, \dots, \mathcal{L}\}$  indexes the layer.  $\{F_k^{(l)}, k = 1, \dots, N_l\}$  are the filters at layer  $l$ , and  $\{F_i^{(l-1)}, i = 1, \dots, N_{l-1}\}$  are the filters at layer  $l-1$ .  $k$  and  $i$  are used to index filters at layers  $l$  and  $l-1$  respectively, and  $N_l$  and  $N_{l-1}$  are the numbers of filters at layers  $l$  and  $l-1$  respectively. The filters are locally supported, so the range of  $y$  is within a local support  $\mathcal{S}_l$  (such as a  $7 \times 7$  image patch). At the bottom layer,  $[F_k^{(0)} * \mathbf{I}](x) = \mathbf{I}_k(x)$ , where  $k \in \{R, G, B\}$  indexes the three color channels. Sub-sampling may be implemented so that in  $[F_k^{(l)} * \mathbf{I}](x)$ ,  $x \in \mathcal{D}_l \subset \mathcal{D}$ . For notational simplicity, we do not make local max pooling explicit in (1).

We take  $h(r) = \max(r, 0)$ , the Rectified Linear Unit (ReLU), that is commonly adopted in modern ConvNet (Krizhevsky et al., 2012).

Let  $(F_k^{(\mathcal{L})})$  be the top layer filters. The filtered images are usually  $1 \times 1$  due to repeated sub-sampling. Suppose there are  $C$  categories. For category  $c \in \{1, \dots, C\}$ , the scoring function for classification is

$$f_c(\mathbf{I}; w) = \sum_{k=1}^{N_{\mathcal{L}}} w_{c,k} [F_k^{(\mathcal{L})} * \mathbf{I}], \quad (2)$$

where  $w_{c,k}$  are the category-specific weight parameters for classification.

**Definition 1** *Discriminative ConvNet: We define the following conditional distribution as the discriminative ConvNet:*

$$p(c|\mathbf{I}; w) = \frac{\exp[f_c(\mathbf{I}; w) + b_c]}{\sum_{c=1}^C \exp[f_c(\mathbf{I}; w) + b_c]}. \quad (3)$$

where  $b_c$  is the bias term, and  $w$  collects all the weight and bias parameters at all the layers.

The discriminative ConvNet is a multinomial logistic regression (or soft-max) that is commonly used for classification (LeCun et al., 1998; Krizhevsky et al., 2012).

**Definition 2** *Generative ConvNet (fully connected version): We define the following random field model as the fully connected version of generative ConvNet:*

$$p(\mathbf{I}|c; w) = p_c(\mathbf{I}; w) = \frac{1}{Z_c(w)} \exp[f_c(\mathbf{I}; w)] q(\mathbf{I}), \quad (4)$$

where  $q(\mathbf{I})$  is a reference distribution or the null model, assumed to be Gaussian white noise in this paper.  $Z(w) = E_q\{\exp[f_c(\mathbf{I}; w)]\}$  is the normalizing constant.

In (4),  $p_c(\mathbf{I}; w)$  is obtained by the exponential tilting of  $q(\mathbf{I})$ , and is the conditional distribution of image given category,  $p(\mathbf{I}|c, w)$ . The model was first proposed by (Dai et al., 2015).

**Proposition 1** *Generative and discriminative ConvNets can be derived from each other:*

(a) Let  $\rho_c$  be the prior probability of category  $c$ , if  $p(\mathbf{I}|c; w) = p_c(\mathbf{I}; w)$  is defined according to model (4), then  $p(c|\mathbf{I}; w)$  is given by model (3), with  $b_c = \log \rho_c - \log Z_c(w) + \text{constant}$ .

(b) Suppose a base category  $c = 1$  is generated by  $q(\mathbf{I})$ , and suppose we fix the scoring function and the bias term of the base category  $f_1(\mathbf{I}; w) = 0$ , and  $b_1 = 0$ . If  $p(c|\mathbf{I}; w)$  is given by model (3), then  $p(\mathbf{I}|c; w) = p_c(\mathbf{I}; w)$  is of the form of model (4), with  $b_c = \log \rho_c - \log \rho_1 + \log Z_c(w)$ .

If we only observe unlabeled data  $\{\mathbf{I}_m, m = 1, \dots, M\}$ , we may still use the exponential tilting form to model and learn from them. A possible model is to learn filters at a certain convolutional layer  $L \in \{1, \dots, \mathcal{L}\}$  of a ConvNet.

**Definition 3** *Generative ConvNet (convolutional version): we define the following Markov random field model as the convolutional version of generative ConvNet:*

$$p(\mathbf{I}; w) = \frac{1}{Z(w)} \exp \left[ \sum_{k=1}^K \sum_{x \in \mathcal{D}_L} [F_k^{(L)} * \mathbf{I}](x) \right] q(\mathbf{I}), \quad (5)$$

where  $w$  consists of all the weight and bias terms that define the filters  $(F_k^{(L)}, k = 1, \dots, K = N_L)$ , and  $q(\mathbf{I})$  is the Gaussian white noise model.

Model (5) corresponds to the exponential tilting model (4) with scoring function

$$f(\mathbf{I}; w) = \sum_{k=1}^K \sum_{x \in \mathcal{D}_L} [F_k^{(L)} * \mathbf{I}](x). \quad (6)$$

For the rest of the paper, we shall focus on the model (5), but all the results can be easily extended to model (4).

## 4. A Prototype Model

In order to reveal the internal structure of the generative ConvNet, it helps to start from the simplest prototype model. A similar model was studied by (Xie et al., 2016).

In our prototype model, we assume that the image domain  $\mathcal{D}$  is small (e.g.,  $10 \times 10$ ). Suppose we want to learn a dictionary of filters or basis functions from a set of observed image patches  $\{\mathbf{I}_m, m = 1, \dots, M\}$  defined on  $\mathcal{D}$ . We denote these filters or basis functions by  $(\mathbf{w}_k, k = 1, \dots, K)$ , where each  $\mathbf{w}_k$  itself is an image patch defined on  $\mathcal{D}$ . Let

$\langle \mathbf{I}, \mathbf{w}_k \rangle = \sum_{x \in \mathcal{D}} \mathbf{w}_k(x) \mathbf{I}(x)$  be the inner product between image patches  $\mathbf{I}$  and  $\mathbf{w}_k$ . It is also the response of  $\mathbf{I}$  to the linear filter  $\mathbf{w}_k$ .

**Definition 4 Prototype model:** We define the following random field model as the prototype model:

$$p(\mathbf{I}; w) = \frac{1}{Z(w)} \exp \left[ \sum_{k=1}^K h(\langle \mathbf{I}, \mathbf{w}_k \rangle + b_k) \right] q(\mathbf{I}), \quad (7)$$

where  $b_k$  is the bias term,  $w = (\mathbf{w}_k, b_k, k = 1, \dots, K)$ , and  $h(r) = \max(r, 0)$ .  $q(\mathbf{I})$  is the Gaussian white noise model,

$$q(\mathbf{I}) = \frac{1}{(2\pi\sigma^2)^{|\mathcal{D}|/2}} \exp \left[ -\frac{1}{2\sigma^2} \|\mathbf{I}\|^2 \right], \quad (8)$$

where  $|\mathcal{D}|$  counts the number of pixels in the domain  $\mathcal{D}$ .

**Piecewise Gaussian and binary activation variables:** Without loss of generality, let us assume  $\sigma^2 = 1$  in  $q(\mathbf{I})$ . Define the binary activation variable  $\delta_k(\mathbf{I}; w) = 1$  if  $\langle \mathbf{I}, \mathbf{w}_k \rangle + b_k > 0$  and  $\delta_k(\mathbf{I}; w) = 0$  otherwise, i.e.,  $\delta_k(\mathbf{I}; w) = 1(\langle \mathbf{I}, \mathbf{w}_k \rangle + b_k > 0)$ , where  $1()$  is the indicator function. Then  $h(\langle \mathbf{I}, \mathbf{w}_k \rangle + b_k) = \delta_k(\mathbf{I}; w)(\langle \mathbf{I}, \mathbf{w}_k \rangle + b_k)$ . The image space is divided into  $2^K$  pieces by the  $K$  hyper-planes,  $\langle \mathbf{I}, \mathbf{w}_k \rangle + b_k = 0$ ,  $k = 1, \dots, K$ , according to the values of the binary activation variables  $(\delta_k(\mathbf{I}; w), k = 1, \dots, K)$ . Consider the piece of image space where  $\delta_k(\mathbf{I}; w) = \delta_k$  for  $k = 1, \dots, K$ . Here we abuse the notation slightly where  $\delta_k \in \{0, 1\}$  on the right hand side denotes the value of  $\delta_k(\mathbf{I}; w)$ . Write  $\delta(\mathbf{I}; w) = (\delta_k(\mathbf{I}; w), k = 1, \dots, K)$ , and  $\delta = (\delta_k, k = 1, \dots, K)$  as an instantiation of  $\delta(\mathbf{I}; w)$ . We call  $\delta(\mathbf{I}; w)$  the activation pattern of  $\mathbf{I}$ . Let  $A(\delta; w) = \{\mathbf{I} : \delta(\mathbf{I}; w) = \delta\}$  be the piece of image space that consists of images sharing the same activation pattern  $\delta$ , then the probability density on this piece

$$\begin{aligned} p(\mathbf{I}; w, \delta) &\propto \exp \left[ \sum_{k=1}^K \delta_k b_k + \langle \mathbf{I}, \sum_{k=1}^K \delta_k \mathbf{w}_k \rangle - \frac{\|\mathbf{I}\|^2}{2} \right] \\ &\propto \exp \left[ -\frac{1}{2} \|\mathbf{I} - \sum_{k=1}^K \delta_k \mathbf{w}_k\|^2 \right], \end{aligned} \quad (9)$$

which is  $N(\sum_k \delta_k \mathbf{w}_k, \mathbf{1})$  restricted to the piece  $A(\delta; w)$ , where the bold font  $\mathbf{1}$  is the identity matrix. The mean of this Gaussian piece seeks to reconstruct images in  $A(\delta; w)$  via the auto-encoding scheme  $\mathbf{I} \rightarrow \delta \rightarrow \sum_k \delta_k \mathbf{w}_k$ .

**Synthesis via reconstruction:** One can sample from  $p(\mathbf{I}; w)$  in (7) by the Langevin dynamics:

$$\mathbf{I}_{\tau+1} = \mathbf{I}_\tau - \frac{\epsilon^2}{2} \left[ \mathbf{I}_\tau - \sum_{k=1}^K \delta_k(\mathbf{I}_\tau; w) \mathbf{w}_k \right] + \epsilon Z_\tau, \quad (10)$$

where  $\tau$  denotes the time step,  $\epsilon$  denotes the step size, assumed to be sufficiently small throughout this paper, and

$Z_\tau \sim N(0, \mathbf{1})$ . The dynamics is driven by the auto-encoding reconstruction error  $\mathbf{I}_\tau - \sum_{k=1}^K \delta_k(\mathbf{I}_\tau; w) \mathbf{w}_k$ , where the reconstruction is based on the binary activation variables  $(\delta_k)$ . This links synthesis to reconstruction.

**Local modes are exactly auto-encoding:** If the mean of a Gaussian piece is a local energy minimum  $\hat{\mathbf{I}}$ , we have the exact auto-encoding  $\hat{\mathbf{I}} = \sum_{k=1}^K \delta_k(\hat{\mathbf{I}}; w) \mathbf{w}_k$ . The encoding process is bottom-up and infers  $\delta_k = \delta_k(\hat{\mathbf{I}}; w) = 1(\langle \hat{\mathbf{I}}, \mathbf{w}_k \rangle + b_k > 0)$ . The decoding process is top-down and reconstructs  $\hat{\mathbf{I}} = \sum_k \delta_k \mathbf{w}_k$ . In the encoding process,  $\mathbf{w}_k$  plays the role of filter. In the decoding process,  $\mathbf{w}_k$  plays the role of basis function.

## 5. Internal Structure of Generative ConvNet

In order to generalize the prototype model (7) to the generative ConvNet (5), we only need to add two elements: (1) Horizontal unfolding: make the filters ( $\mathbf{w}_k$ ) convolutional. (2) Vertical unfolding: make the filters ( $\mathbf{w}_k$ ) multi-layer or hierarchical. The results we have obtained for the prototype model can be unfolded accordingly.

To derive the internal representational structure of the generative ConvNet, the key is to write the scoring function  $f(\mathbf{I}; w)$  as a linear function  $\alpha + \langle \mathbf{I}, \mathbf{B} \rangle$  on each piece of image space with fixed activation pattern. Combined with the  $\|\mathbf{I}\|^2/2$  term from  $q(\mathbf{I})$ , the energy function will be quadratic, i.e.,  $\|\mathbf{I}\|^2/2 - \langle \mathbf{I}, \mathbf{B} \rangle - \alpha$ , and the probability distribution will be truncated Gaussian with  $\mathbf{B}$  being the mean. In order to write  $f(\mathbf{I}; w) = \alpha + \langle \mathbf{I}, \mathbf{B} \rangle$  for fixed activation pattern, we shall use vector notation for ConvNet, and derive  $\mathbf{B}$  by a top-down deconvolution process.  $\mathbf{B}$  can also be obtained by  $\partial f(\mathbf{I}; w)/\partial \mathbf{I}$  via back-propagation computation.

For filters at level  $l$ , the  $N_l$  filters are denoted by the compact notation  $\mathbf{F}^{(l)} = (F_k^{(l)}, k = 1, \dots, N_l)$ . The  $N_l$  filtered images or feature maps are denoted by the compact notation  $\mathbf{F}^{(l)} * \mathbf{I} = (F_k^{(l)} * \mathbf{I}, k = 1, \dots, N_l)$ .  $\mathbf{F}^{(l)} * \mathbf{I}$  is a 3D image, containing all the  $N_l$  filtered images at layer  $l$ . In vector notation, the recursive formula (1) of ConvNet filters can be rewritten as

$$[F_k^{(l)} * \mathbf{I}](x) = h \left( \langle \mathbf{w}_{k,x}^{(l)}, \mathbf{F}^{(l-1)} * \mathbf{I} \rangle + b_{l,k} \right), \quad (11)$$

where  $\mathbf{w}_{k,x}^{(l)}$  matches the dimension of  $\mathbf{F}^{(l-1)} * \mathbf{I}$ , which is a 3D image containing all the  $N_{l-1}$  filtered images at layer  $l-1$ . Specifically,

$$\langle \mathbf{w}_{k,x}^{(l)}, \mathbf{F}^{(l-1)} * \mathbf{I} \rangle = \sum_{i=1}^{N_{l-1}} \sum_{y \in \mathcal{S}_l} w_{i,y}^{(l,k)} [F_i^{(l-1)} * \mathbf{I}](x+y). \quad (12)$$

The 3D basis functions ( $\mathbf{w}_{k,x}^{(l)}$ ) are locally supported, and they are spatially translated copies for different positions  $x$ ,

i.e.,  $\mathbf{w}_{k,x,i}^{(l)}(x+y) = w_{i,y}^{(l,k)}$ , for  $i \in \{1, \dots, N_{l-1}\}$ ,  $x \in \mathcal{D}_l$  and  $y \in \mathcal{S}_l$ . For instance, at layer  $l = 1$ ,  $\mathbf{w}_{k,x}^{(1)}$  is a Gabor-like wavelet of type  $k$  centered at position  $x$ .

Define the binary activation variable

$$\delta_{k,x}^{(l)}(\mathbf{I}; w) = 1 \left( \langle \mathbf{w}_{k,x}^{(l)}, \mathbf{F}^{(l-1)} * \mathbf{I} \rangle + b_{l,k} > 0 \right). \quad (13)$$

According to (1), we have the following bottom-up process:

$$[F_k^{(l)} * \mathbf{I}](x) = \delta_{k,x}^{(l)}(\mathbf{I}; w) \left( \langle \mathbf{w}_{k,x}^{(l)}, \mathbf{F}^{(l-1)} * \mathbf{I} \rangle + b_{l,k} \right). \quad (14)$$

Let  $\delta(\mathbf{I}; w) = (\delta_{k,x}^{(l)}(\mathbf{I}; w), \forall k, x, l)$  be the activation pattern at all the layers. The active pattern  $\delta(\mathbf{I}; w)$  can be computed in the bottom-up process (13) and (14) of ConvNet.

For the scoring function  $f(\mathbf{I}; w) = \sum_{k=1}^K \sum_{x \in \mathcal{D}_L} [F_k^{(L)} * \mathbf{I}](x)$  defined in (6) for the generative ConvNet, we can write it in terms of lower layers ( $l \leq L$ ) of filter responses:

$$\begin{aligned} f(\mathbf{I}; w) &= \alpha_l + \langle \mathbf{B}^{(l)}, \mathbf{F}^{(l)} * \mathbf{I} \rangle \\ &= \alpha_l + \sum_{k=1}^{N_l} \sum_{x \in \mathcal{D}_l} \mathbf{B}_k^{(l)}(x) [F_k^{(l)} * \mathbf{I}](x), \end{aligned} \quad (15)$$

where  $\mathbf{B}^{(l)} = (\mathbf{B}_k^{(l)}(x), k = 1, \dots, N_l, x \in \mathcal{D}_l)$  consists of the maps of the coefficients at layer  $l$ .  $\mathbf{B}^{(l)}$  matches the dimension of  $\mathbf{F}^{(l)} * \mathbf{I}$ . When  $l = L$ ,  $\mathbf{B}^{(L)}$  consists of maps of 1's, i.e.,  $\mathbf{B}_k^{(L)}(x) = 1$  for  $k = 1, \dots, K = N_L$  and  $x \in \mathcal{D}_L$ . According to equations (14) and (15), we have the following top-down process:

$$\mathbf{B}^{(l-1)} = \sum_{k=1}^{N_l} \sum_{x \in \mathcal{D}_l} \mathbf{B}_k^{(l)}(x) \delta_{k,x}^{(l)}(\mathbf{I}; w) \mathbf{w}_{k,x}^{(l)}, \quad (16)$$

where both  $\mathbf{B}^{(l-1)}$  and  $\mathbf{w}_{k,x}^{(l)}$  match the dimension of  $\mathbf{F}^{(l-1)} * \mathbf{I}$ . Equation (16) is a top-down deconvolution process, where  $\mathbf{B}_k^{(l)}(x) \delta_{k,x}^{(l)}$  serves as the coefficient of the basis function  $\mathbf{w}_{k,x}^{(l)}$ . The top-down deconvolution process (16) is similar to but subtly different from that in (Zeiler & Fergus, 2014), because equation (16) is controlled by the multiple layers of activation variables  $\delta_{k,x}^{(l)}$  computed in the bottom-up process of the ConvNet. Specifically,  $\delta_{k,x}^{(l)}$  turns on or off the basis function  $\mathbf{w}_{k,x}^{(l)}$ , while  $\delta_{k,x}^{(l)}$  is determined by  $F_k^{(l)}$ . The recursive relationship for  $\alpha_l$  can be similarly derived.

In the bottom-up convolution process (14),  $(\mathbf{w}_{k,x}^{(l)})$  serve as filters. In the top-down deconvolution process (16),  $(\mathbf{w}_{k,x}^{(l)})$  serve as basis functions.

Let  $\mathbf{B} = \mathbf{B}^{(0)}$ ,  $\alpha = \alpha_0$ . Since  $\mathbf{F}^{(0)} * \mathbf{I} = \mathbf{I}$ , we have  $f(\mathbf{I}; w) = \alpha + \langle \mathbf{I}, \mathbf{B} \rangle$ . Note that  $\mathbf{B}$  depends on the activation pattern  $\delta(\mathbf{I}; w) = (\delta_{k,x}^{(l)}(\mathbf{I}; w), \forall k, x, l)$ , as well as  $w$  that collects the weight and bias parameters at all the layers.

On the piece of image space  $A(\delta; w) = \{\mathbf{I} : \delta(\mathbf{I}; w) = \delta\}$  of a fixed activation pattern (again we slightly abuse the notation where  $\delta = (\delta_{k,x}^{(l)} \in \{0, 1\}, \forall k, x, l)$  denotes an instantiation of the activation pattern),  $\mathbf{B}$  and  $\alpha$  depend on  $\delta$  and  $w$ . To make this dependency explicit, we denote  $\mathbf{B} = \mathbf{B}_{w,\delta}$  and  $\alpha = \alpha_{w,\delta}$ , thus

$$f(\mathbf{I}; w) = \alpha_{w,\delta} + \langle \mathbf{I}, \mathbf{B}_{w,\delta} \rangle. \quad (17)$$

See (Montufar et al., 2014) for an analysis of the number of linear pieces.

**Theorem 1** *Generative ConvNet is piecewise Gaussian: With ReLU  $h(r) = \max(0, r)$  and Gaussian white noise  $q(\mathbf{I})$ ,  $p(\mathbf{I}; w)$  of model (5) is piecewise Gaussian. On each piece  $A(\delta; w)$ , the density is  $N(\mathbf{B}_{w,\delta}, \mathbf{1})$  truncated to  $A(\delta; w)$ , i.e.,  $\mathbf{B}_{w,\delta}$  is an approximated reconstruction of images in  $A(\delta; w)$ .*

Theorem 1 follows from the fact that on  $A(\delta; w)$ ,

$$\begin{aligned} p(\mathbf{I}; w, \delta) &\propto \exp \left[ \alpha_{w,\delta} + \langle \mathbf{I}, \mathbf{B}_{w,\delta} \rangle - \frac{\|\mathbf{I}\|^2}{2} \right] \\ &\propto \exp \left[ -\frac{1}{2} \|\mathbf{I} - \mathbf{B}_{w,\delta}\|^2 \right], \end{aligned} \quad (18)$$

which is  $N(\mathbf{B}_{w,\delta}, \mathbf{1})$  restricted to  $A(\delta; w)$ .

For each  $\mathbf{I}$ , the binary activation variables in  $\delta = \delta(\mathbf{I}; w)$  are computed by the bottom-up convolution process (13) and (14), and  $\mathbf{B}_{w,\delta}$  is computed by the top-down deconvolution process (16).  $\mathbf{B}_{w,\delta}$  seeks to reconstruct images in  $A(\delta; w)$  via the auto-encoding scheme  $\mathbf{I} \rightarrow \delta \rightarrow \mathbf{B}_{w,\delta}$ .

One can sample from  $p(\mathbf{I}; w)$  of model (5) by the Langevin dynamics:

$$\mathbf{I}_{\tau+1} = \mathbf{I}_{\tau} - \frac{\epsilon^2}{2} [\mathbf{I}_{\tau} - \mathbf{B}_{w,\delta(\mathbf{I}_{\tau}; w)}] + \epsilon Z_{\tau}, \quad (19)$$

where  $Z_{\tau} \sim N(0, \mathbf{1})$ . Again, the dynamics is driven by the auto-encoding reconstruction error  $\mathbf{I} - \mathbf{B}_{w,\delta(\mathbf{I}; w)}$ .

The deterministic part of the Langevin equation (19) defines an attractor dynamics that converges to a local energy minimum (Zhu & Mumford, 1998).

**Proposition 2** *The local modes are exactly auto-encoding: Let  $\hat{\mathbf{I}}$  be a local maximum of  $p(\mathbf{I}; w)$  of model (5), then we have exact auto-encoding of  $\hat{\mathbf{I}}$  with the following bottom-up and top-down passes:*

$$\begin{aligned} \text{Bottom-up encoding: } \delta &= \delta(\hat{\mathbf{I}}; w); \\ \text{Top-down decoding: } \hat{\mathbf{I}} &= \mathbf{B}_{w,\delta}. \end{aligned} \quad (20)$$

The local energy minima are the means of the Gaussian pieces in Theorem 1, but the reverse is not necessarily true because  $\mathbf{B}_{w,\delta}$  does not necessarily belong to  $A(\delta; w)$ . But if  $\mathbf{B}_{w,\delta} \in A(\delta; w)$ , then  $\mathbf{B}_{w,\delta}$  must be a local mode and is exactly auto-encoding.

Proposition 2 can be generalized to general non-linear  $h()$ , whereas Theorem 1 is true only for piecewise linear  $h()$  such as ReLU.

Proposition 2 is related to the Hopfield network (Hopfield, 1982). It shows that the Hopfield minima can be represented by a hierarchical auto-encoder.

## 6. Learning Generative ConvNet

The learning of  $w$  from training images  $\{\mathbf{I}_m, m = 1, \dots, M\}$  can be accomplished by maximum likelihood. Let  $L(w) = \sum_{m=1}^M \log p(\mathbf{I}_m; w)/M$ , with  $p(\mathbf{I}_m; w)$  defined in (5), then

$$\frac{\partial L(w)}{\partial w} = \frac{1}{M} \sum_{m=1}^M \frac{\partial}{\partial w} f(\mathbf{I}_m; w) - \mathbb{E}_w \left[ \frac{\partial}{\partial w} f(\mathbf{I}_m; w) \right]. \quad (21)$$

The expectation can be approximated by Monte Carlo samples (Younes, 1999) from the Langevin dynamics (19). See Algorithm 1 for a description of the learning and sampling algorithm.

---

### Algorithm 1 Learning and sampling algorithm

---

**Input:**

- (1) training images  $\{\mathbf{I}_m, m = 1, \dots, M\}$
- (2) number of synthesized images  $\tilde{M}$
- (3) number of Langevin steps  $L$
- (4) number of learning iterations  $T$

**Output:**

- (1) estimated parameters  $w$
- (2) synthesized images  $\{\tilde{\mathbf{I}}_m, m = 1, \dots, \tilde{M}\}$

- 1: Let  $t \leftarrow 0$ , initialize  $w^{(0)} \leftarrow 0$ .
  - 2: Initialize  $\tilde{\mathbf{I}}_m \leftarrow 0$ , for  $m = 1, \dots, \tilde{M}$ .
  - 3: **repeat**
  - 4: For each  $m$ , run  $L$  steps of Langevin dynamics to update  $\tilde{\mathbf{I}}_m$ , i.e., starting from the current  $\tilde{\mathbf{I}}_m$ , each step follows equation (19).
  - 5: Calculate  $H^{\text{obs}} = \sum_{m=1}^M \frac{\partial}{\partial w} f(\mathbf{I}_m; w^{(t)})/M$ , and  $H^{\text{syn}} = \sum_{m=1}^{\tilde{M}} \frac{\partial}{\partial w} f(\tilde{\mathbf{I}}_m; w^{(t)})/\tilde{M}$ .
  - 6: Update  $w^{(t+1)} \leftarrow w^{(t)} + \eta(H^{\text{obs}} - H^{\text{syn}})$ , with step size  $\eta$ .
  - 7: Let  $t \leftarrow t + 1$
  - 8: **until**  $t = T$
- 

If we want to learn from big data, we may use the contrastive divergence (Hinton, 2002) by starting the Langevin

dynamics from the observed images. The contrastive divergence tends to learn the auto-encoder in the generative ConvNet.

Suppose we start from an observed image  $\mathbf{I}^{\text{obs}}$ , and run a small number of iterations of Langevin dynamics (19) to get a synthesized image  $\mathbf{I}^{\text{syn}}$ . If both  $\mathbf{I}^{\text{obs}}$  and  $\mathbf{I}^{\text{syn}}$  share the same activation pattern  $\delta(\mathbf{I}^{\text{obs}}; w) = \delta(\mathbf{I}^{\text{syn}}; w) = \delta$ , then  $f(\mathbf{I}; w) = a_{w,\delta} + \langle \mathbf{I}, \mathbf{B}_{w,\delta} \rangle$  for both  $\mathbf{I}^{\text{obs}}$  and  $\mathbf{I}^{\text{syn}}$ . Then the contribution of  $\mathbf{I}^{\text{obs}}$  to the learning gradient is

$$\frac{\partial}{\partial w} f(\mathbf{I}^{\text{obs}}; w) - \frac{\partial}{\partial w} f(\mathbf{I}^{\text{syn}}; w) = \langle \mathbf{I}^{\text{obs}} - \mathbf{I}^{\text{syn}}, \frac{\partial}{\partial w} \mathbf{B}_{w,\delta} \rangle. \quad (22)$$

If  $\mathbf{I}^{\text{syn}}$  is close to the mean  $\mathbf{B}_{w,\delta}$  and if  $\mathbf{B}_{w,\delta}$  is a local mode, then the contrastive divergence tends to reconstruct  $\mathbf{I}^{\text{obs}}$  by the local mode  $\mathbf{B}_{w,\delta}$ , because the gradient

$$\frac{\partial}{\partial w} \|\mathbf{I}^{\text{obs}} - \mathbf{B}_{w,\delta}\|^2/2 = -\langle \mathbf{I}^{\text{obs}} - \mathbf{B}_{w,\delta}, \frac{\partial}{\partial w} \mathbf{B}_{w,\delta} \rangle. \quad (23)$$

Hence contrastive divergence learns Hopfield network which memorizes observations by local modes.

We can establish a precise connection for one-step contrastive divergence.

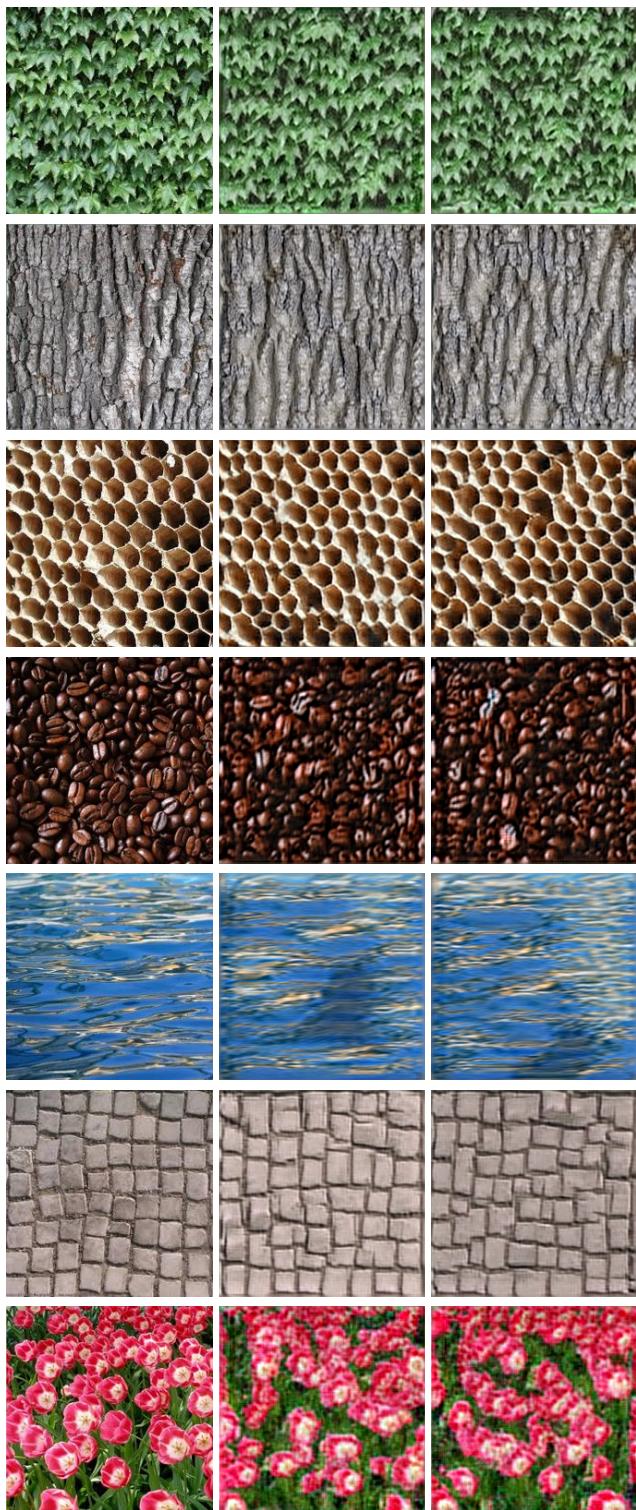
**Proposition 3** *Contrastive divergence learns to reconstruct: If the one-step Langevin dynamics does not change the activation pattern, i.e.,  $\delta(\mathbf{I}^{\text{obs}}; w) = \delta(\mathbf{I}^{\text{syn}}; w) = \delta$ , then the one-step contrastive divergence has an expected gradient that is proportional to the reconstruction gradient:*

$$\mathbb{E} \left[ \frac{\partial}{\partial w} f(\mathbf{I}^{\text{obs}}; w) - \frac{\partial}{\partial w} f(\mathbf{I}^{\text{syn}}; w) \right] \propto \frac{\partial}{\partial w} \|\mathbf{I}^{\text{obs}} - \mathbf{B}_{w,\delta}\|^2. \quad (24)$$

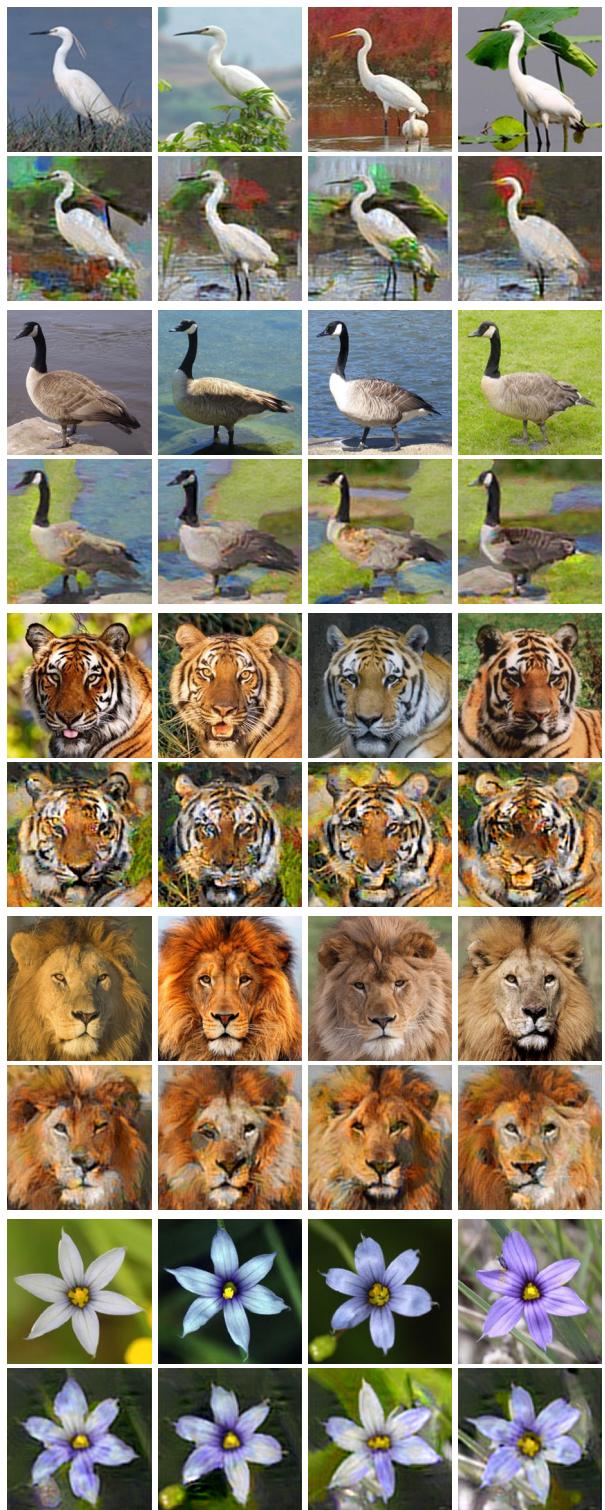
This is because  $\mathbf{I}^{\text{syn}} = \mathbf{I}^{\text{obs}} - \frac{\epsilon^2}{2} [\mathbf{I}^{\text{obs}} - \mathbf{B}_{w,\delta}] + \epsilon Z$ , hence  $\mathbb{E}_Z [\mathbf{I}^{\text{obs}} - \mathbf{I}^{\text{syn}}] \propto \mathbf{I}^{\text{obs}} - \mathbf{B}_{w,\delta}$ , and Proposition 3 follows from (22) and (23).

The contrastive divergence learning updates the bias terms to match the statistics of the activation patterns of  $\mathbf{I}^{\text{obs}}$  and  $\mathbf{I}^{\text{syn}}$ , which helps to ensure that  $\delta(\mathbf{I}^{\text{obs}}; w) = \delta(\mathbf{I}^{\text{syn}}; w)$ .

Proposition 3 is related to score matching estimator (Hyvärinen, 2005), whose connection with contrastive divergence based on one-step Langevin was studied by (Hyvärinen, 2007). Our work can be considered a sharpened specialization of this connection, where the piecewise linear structure in ConvNet greatly simplifies the matter by getting rid of the complicated second derivative terms, so that the contrastive divergence gradient becomes exactly proportional to the gradient of the reconstruction error, which is not the case in general score matching estimator. Also, our work gives an explicit hierarchical realization of auto-encoder based sampling (Alain & Bengio, 2014). The connection with Hopfield network also appears new.



*Figure 1.* Generating texture patterns. For each category, the first image is the training image, and the rest are 2 of the images generated by the learning algorithm.



*Figure 2.* Generating object patterns. For each category, the first row displays 4 of the training images, and the second row displays 4 of the images generated by the learning algorithm.



Figure 3. Reconstruction by one-step contrastive divergence. The first row displays 4 of the training images, and the second row displays the corresponding reconstructed images.

## 7. Synthesis and Reconstruction

We show that the generative ConvNet is capable of learning and generating realistic natural image patterns. Such an empirical proof of concept validates the generative capacity of the model. We also show that contrastive divergence learning can indeed reconstruct the observed images, thus empirically validating Proposition 3.

The code in our experiments is based on the MatConvNet package of (Vedaldi & Lenc, 2015).

Unlike (Lu et al., 2016), the generative ConvNets in our experiments are learned from scratch without relying on the pre-learned filters of existing ConvNets.

When learning the generative ConvNet, we grow the layers sequentially. Starting from the first layer, we sequentially add the layers one by one. Each time we learn the model and generate the synthesized images using Algorithm 1. While learning the new layer of filters, we also refine the lower layers of filters by back-propagation.

We use  $\tilde{M} = 16$  parallel chains for Langevin sampling. The number of Langevin iterations between every two consecutive updates of parameters is  $L = 10$ . With each new added layer, the number of learning iterations is  $T = 700$ . We follow the standard procedure to prepare the training images of size  $224 \times 224$ , whose intensities are within  $[0, 255]$ , and we subtract the mean image. We fix  $\sigma^2 = 1$  in the reference distribution  $q(\mathbf{I})$ .

For each of the 3 experiments, we use the same set of parameters for all the categories without tuning.

### 7.1. Experiment 1: Generating texture patterns

We learn a 3-layer generative ConvNet. The first layer has 100  $15 \times 15$  filters with sub-sampling size of 3. The second layer has 64  $5 \times 5$  filters with sub-sampling size of 1. The third layer has 30  $3 \times 3$  filters with sub-sampling size of 1. We learn a generative ConvNet for each category from

a single training image. Figure 1 displays the results. For each category, the first image is the training image, and the rest are 2 of the images generated by the learning algorithm.

### 7.2. Experiment 2: Generating object patterns

Experiment 1 shows clearly that the generative ConvNet can learn from images without alignment. We can also specialize it to learning aligned object patterns by using a single top-layer filter that covers the whole image. We learn a 4-layer generative ConvNet from images of aligned objects. The first layer has 100  $7 \times 7$  filters with sub-sampling size of 2. The second layer has 64  $5 \times 5$  filters with sub-sampling size of 1. The third layer has 20  $3 \times 3$  filters with sub-sampling size of 1. The fourth layer is a fully connected layer with a single filter that covers the whole image. When growing the layers, we always keep the top-layer single filter, and train it together with the existing layers. We learn a generative ConvNet for each category, where the number of training images for each category is around 10, and they are collected from the Internet. Figure 2 shows the results. For each category, the first row displays 4 of the training images, and the second row shows 4 of the images generated by the learning algorithm.

### 7.3. Experiment 3: Contrastive divergence learns to auto-encode

We experiment with one-step contrastive divergence learning on a small training set of 10 images collected from the Internet. The ConvNet structure is the same as in experiment 1. For computational efficiency, we learn all the layers of filters simultaneously. The number of learning iterations is  $T = 1200$ . Starting from the observed images, the number of Langevin iterations is  $L = 1$ . Figure 3 shows the results. The first row displays 4 of the training images, and the second row displays the corresponding auto-encoding reconstructions with the learned parameters.

## 8. Conclusion

This paper derives the generative ConvNet from the discriminative ConvNet, and reveals an internal representational structure that is unique among energy-based models.

The generative ConvNet has the potential to learn from big unlabeled data, either by contrastive divergence or some reconstruction based methods.

## Code and data

The code and training images can be downloaded from the project page: <http://www.stat.ucla.edu/~ywu/GenerativeConvNet/main.html>

## Acknowledgement

The code in our work is based on the Matlab code of MatConvNet of (Vedaldi & Lenc, 2015). We thank the authors for making their code public.

We thank the three reviewers for their insightful comments that have helped us improve the presentation of the paper. We thank Jifeng Dai and Wenze Hu for earlier collaborations related to this work.

The work is supported by NSF DMS 1310391, ONR MURI N00014-10-1-0933 and DARPA SIMPLEX N66001-15-C-4035.

## References

- Alain, Guillaume and Bengio, Yoshua. What regularized auto-encoders learn from the data-generating distribution. *The Journal of Machine Learning Research*, 15(1): 3563–3593, 2014.
- Dai, Jifeng, Lu, Yang, and Wu, Ying Nian. Generative modeling of convolutional neural networks. In *ICLR*, 2015.
- Hinton, Geoffrey E. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.
- Hinton, Geoffrey E., Osindero, Simon, and Teh, Yee-Whye. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006a.
- Hinton, Geoffrey E, Osindero, Simon, Welling, Max, and Teh, Yee-Whye. Unsupervised discovery of nonlinear structure using contrastive backpropagation. *Cognitive Science*, 30(4):725–731, 2006b.
- Hopfield, John J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8): 2554–2558, 1982.
- Hyvärinen, Aapo. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.
- Hyvärinen, Aapo. Connections between score matching, contrastive divergence, and pseudolikelihood for continuous-valued variables. *Neural Networks, IEEE Transactions on*, 18(5):1529–1531, 2007.
- Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *NIPS*, pp. 1097–1105, 2012.
- LeCun, Yann, Bottou, Léon, Bengio, Yoshua, and Haffner, Patrick. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- LeCun, Yann, Chopra, Sumit, Hadsell, Rata, Ranzato, Mare’Aurelio, and Huang, Fu Jie. A tutorial on energy-based learning. In *Predicting Structured Data*. MIT Press, 2006.
- Lee, Honglak, Grosse, Roger, Ranganath, Rajesh, and Ng, Andrew Y. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML*, pp. 609–616. ACM, 2009.
- Lu, Yang, Zhu, Song-Chun, and Wu, Ying Nian. Learning FRAME models using CNN filters. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- Montufar, Guido F, Pascanu, Razvan, Cho, Kyunghyun, and Bengio, Yoshua. On the number of linear regions of deep neural networks. In *NIPS*, pp. 2924–2932, 2014.
- Ngiam, Jiquan, Chen, Zhenghao, Koh, Pang Wei, and Ng, Andrew Y. Learning deep energy models. In *ICML*, 2011.
- Olshausen, Bruno A and Field, David J. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37(23):3311–3325, 1997.
- Roth, Stefan and Black, Michael J. Fields of experts: A framework for learning image priors. In *CVPR*, volume 2, pp. 860–867. IEEE, 2005.
- Salakhutdinov, Ruslan and Hinton, Geoffrey E. Deep boltzmann machines. In *AISTATS*, 2009.
- Swersky, Kevin, Ranzato, Marc’Aurelio, Buchman, David, Marlin, Benjamin, and Freitas, Nando. On autoencoders and score matching for energy based models. In *ICML*, pp. 1201–1208. ACM, 2011.
- Teh, Yee-Whye, Welling, Max, Osindero, Simon, and Hinton, Geoffrey E. Energy-based models for sparse overcomplete representations. *The Journal of Machine Learning Research*, 4:1235–1260, 2003.
- Vedaldi, A. and Lenc, K. Matconvnet – convolutional neural networks for matlab. In *Proceeding of the ACM Int. Conf. on Multimedia*, 2015.
- Vincent, Pascal. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.
- Wu, Ying Nian, Zhu, Song-Chun, and Guo, Cheng-En. From information scaling of natural images to regimes of statistical models. *Quarterly of Applied Mathematics*, 66:81–122, 2008.

Xie, Jianwen, Hu, Wenz, Zhu, Song-Chun, and Wu, Ying Nian. Learning sparse FRAME models for natural image patterns. *International Journal of Computer Vision*, pp. 1–22, 2014.

Xie, Jianwen, Lu, Yang, Zhu, Song-Chun, and Wu, Ying Nian. Inducing wavelets into random fields via generative boosting. *Journal of Applied and Computational Harmonic Analysis*, 41:4–25, 2016.

Younes, Laurent. On the convergence of markovian stochastic algorithms with rapidly decreasing ergodicity rates. *Stochastics: An International Journal of Probability and Stochastic Processes*, 65(3-4):177–228, 1999.

Zeiler, Matthew D and Fergus, Rob. Visualizing and understanding convolutional neural networks. In *ECCV*, 2014.

Zhu, Song-Chun and Mumford, David. Grade: Gibbs reaction and diffusion equations. In *ICCV*, pp. 847–854, 1998.

Zhu, Song-Chun, Wu, Ying Nian, and Mumford, David. Minimax entropy principle and its application to texture modeling. *Neural Computation*, 9(8):1627–1660, 1997.