Learning with Hierarchical-Deep Models

Ruslan Salakhutdinov, Joshua B. Tenenbaum, and Antonio Torralba

Abstract—We introduce HD (or "Hierarchical-Deep") models, a new compositional learning architecture that integrates deep learning models with structured hierarchical Bayesian models. Specifically we show how we can learn a hierarchical Dirichlet process (HDP) prior over the activities of the top-level features in a Deep Boltzmann Machine (DBM). This compound HDP-DBM model learns to learn novel concepts from very few training examples, by learning low-level generic features, high-level features that capture correlations among low-level features, and a category hierarchy for sharing priors over the high-level features that are typical of different kinds of concepts. We present efficient learning and inference algorithms for the HDP-DBM model and show that it is able to learn new concepts from very few examples on CIFAR-100 object recognition, handwritten character recognition, and human motion capture datasets.

Index Terms—Deep Networks, Deep Boltzmann Machines, Hierarchical Bayesian Models, One-Shot Learning.

1 Introduction

The ability to learn abstract representations that support transfer to novel but related tasks, lies at the core of many problems in computer vision, natural language processing, cognitive science, and machine learning. In typical applications of machine classification algorithms today, learning a new concept requires tens, hundreds or thousands of training examples. For human learners, however, just one or a few examples are often sufficient to grasp a new category and make meaningful generalizations to novel instances [15], [25], [31], [44]. Clearly this requires very strong but also appropriately tuned inductive biases. The architecture we describe here takes a step towards this ability by learning several forms of abstract knowledge at different levels of abstraction, that support transfer of useful inductive biases from previously learned concepts to novel ones.

We call our architectures *compound HD models*, where "HD" stands for "Hierarchical-Deep", because they are derived by composing hierarchical nonparametric Bayesian models with deep networks, two influential approaches from the recent unsupervised learning literature with complementary strengths. Recently introduced deep learning models, including Deep Belief Networks [12], Deep Boltzmann Machines [29], deep autoencoders [19], and many others [9], [10], [21], [22], [26], [32], [34], [43], have been shown to learn useful distributed feature representations for many high-dimensional datasets. The ability to automatically learn in multiple layers allows deep models to construct sophisticated domain-specific features without the need to rely on precise human-crafted input representations, increasingly important with the proliferation of data sets and application domains.

- *Ruslan Salakhutdinov is with the Department of Statistics and Computer Science, University of Toronto, Toronto, ON, M5S 3G3, Canada. E-mail: rsalakhu@utstat.toronto.edu*
- *Joshua B. Tenenbaum is with Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139. E-mail: jbt@mit.edu*
- *Antonio Torralba is with Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, Cambridge, MA 02139. E-mail: torralba@mit.edu.*

While the features learned by deep models can enable more rapid and accurate classification learning, deep networks themselves are not well suited to learning novel classes from few examples. All units and parameters at all levels of the network are engaged in representing any given input ("distributed representations"), and are adjusted together during learning. In contrast, we argue that learning new classes from a handful of training examples will be easier in architectures that can explicitly identify only a small number of degrees of freedom (latent variables and parameters) that are relevant to the new concept being learned, and thereby achieve more appropriate and flexible transfer of learned representations to new tasks. This ability is the hallmark of hierarchical Bayesian (HB) models, recently proposed in computer vision, statistics, and cognitive science [8], [11], [15], [28], [44] for learning from few examples. Unlike deep networks, these HB models explicitly represent category hierarchies that admit sharing the appropriate abstract knowledge about the new class's parameters via a prior abstracted from related classes. HB approaches, however, have complementary weaknesses relative to deep networks. They typically rely on domain-specific handcrafted features [2], [11] (e.g. GIST, SIFT features in computer vision, MFCC features in speech perception domains). Committing to the a-priori defined feature representations, instead of learning them from data, can be detrimental. This is especially important when learning complex tasks, as it is often difficult to hand-craft high-level features explicitly in terms of raw sensory input. Moreover, many HB approaches often assume a fixed hierarchy for sharing parameters [6], [33] instead of discovering how parameters are shared among classes in an unsupervised fashion.

In this work we propose compound HD (hierarchical-deep) architectures that integrate these deep models with structured hierarchical Bayesian models. In particular, we show how we can learn a hierarchical Dirichlet process (HDP) prior over the activities of the top-level features in a Deep Boltzmann Machine (DBM), coming to represent both a layered hierarchy of increasingly abstract features, and a tree-structured hierarchy of classes. Our model depends minimally on domain-specific representations and achieves state-of-the-art performance by

unsupervised discovery of three components: (a) low-level features that abstract from the raw high-dimensional sensory input (e.g. pixels, or 3D joint angles) and provide a useful first representation for all concepts in a given domain; (b) high-level part-like features that express the distinctive perceptual structure of a specific class, in terms of class-specific correlations over low-level features; and (c) a hierarchy of super-classes for sharing abstract knowledge among related classes via a prior on which higher-level features are likely to be distinctive for classes of a certain kind and are thus likely to support learning new concepts of that kind.

We evaluate the compound HDP-DBM model on three different perceptual domains. We also illustrate the advantages of having a full generative model, extending from highly abstract concepts all the way down to sensory inputs: we cannot only generalize class labels but also synthesize new examples in novel classes that look reasonably natural, and we can significantly improve classification performance by learning parameters at *all levels jointly* by maximizing a joint log-probability score.

There have also been several approaches in the computer vision community addressing the problem of learning with few examples. Torralba et al. [42] proposed to use several boosted detectors in a multi-task setting, where features are shared between several categories. Bart and Ullman [3] further proposed a cross-generalization framework for learning with few examples. Their key assumption is that new features for a novel category are selected from the pool of features that was useful for previously learned classification tasks. In contrast to our work, the above approaches are discriminative by nature and do not attempt to identify similar or relevant categories. Babenko et al. [1] used a boosting approach that simultaneously groups together categories into several super-categories, sharing a similarity metric within these classes. They, however, did not attempt to address transfer learning problem, and primarily focused on large-scale image retrieval tasks. Finally, Fei-Fei et al. [11] used a hierarchical Bayesian approach, with a prior on the parameters of new categories that was induced from other categories. However, their approach was not ideal as a generic approach to transfer learning with few examples. They learned only a single prior shared across all categories. The prior was learned only from three categories, chosen by hand. Compared to our work, they used a more elaborate visual object model, based on multiple parts with separate appearance and shape components.

2 Deep Boltzmann Machines (DBMs)

A Deep Boltzmann Machine is a network of symmetrically coupled stochastic binary units. It contains a set of visible units $\mathbf{v} \in \{0,1\}^D$, and a sequence of layers of hidden units $\mathbf{h}^{(1)} \in \{0,1\}^{F_1}$, $\mathbf{h}^{(2)} \in \{0,1\}^{F_2}$,..., $\mathbf{h}^{(L)} \in \{0,1\}^{F_L}$. There are connections only between hidden units in adjacent layers, as well as between visible and hidden units in the first hidden layer. Consider a DBM with three hidden layers[1] (i.e. $L = 3$). The energy of the joint configuration $\{\mathbf{v}, \mathbf{h}\}$ is defined as:

$$E(\mathbf{v}, \mathbf{h}; \boldsymbol{\psi}) = -\sum_{ij} W_{ij}^{(1)} v_i h_j^{(1)} - \sum_{jl} W_{jl}^{(2)} h_j^{(1)} h_l^{(2)} - \sum_{lk} W_{lk}^{(3)} h_l^{(2)} h_k^{(3)},$$

where $\mathbf{h} = \{\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}\}$ represent the set of hidden units, and $\boldsymbol{\psi} = \{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{W}^{(3)}\}$ are the model parameters, representing visible-to-hidden and hidden-to-hidden symmetric interaction terms[2].

The probability that the model assigns to a visible vector $\mathbf{v}$ is given by the Boltzmann distribution:

$$P(\mathbf{v}; \boldsymbol{\psi}) = \frac{1}{\mathcal{Z}(\boldsymbol{\psi})} \sum_{\mathbf{h}} \exp\left(-E(\mathbf{v}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}; \boldsymbol{\psi})\right). \quad (1)$$

Observe that setting both $\mathbf{W}^{(2)}$=0 and $\mathbf{W}^{(3)}$=0 recovers the simpler Restricted Boltzmann Machine (RBM) model.

The conditional distributions over the visible and the three sets of hidden units are given by:

$$\begin{aligned}
p(h_j^{(1)} = 1 | \mathbf{v}, \mathbf{h}^{(2)}) &= g\left(\sum_{i=1}^{D} W_{ij}^{(1)} v_i + \sum_{l=1}^{F_2} W_{jl}^{(2)} h_l^{(2)}\right), \\
p(h_l^{(2)} = 1 | \mathbf{h}^{(1)}, \mathbf{h}^{(3)}) &= g\left(\sum_{j=1}^{F_1} W_{jl}^{(2)} h_j^{(1)} + \sum_{k=1}^{F_3} W_{lk}^{(3)} h_k^{(3)}\right), \\
p(h_k^{(3)} = 1 | \mathbf{h}^{(2)}) &= g\left(\sum_{l=1}^{F_2} W_{lk}^{(3)} h_l^{(2)}\right), \\
p(v_i = 1 | \mathbf{h}^{(1)}) &= g\left(\sum_{j=1}^{F_1} W_{ij}^{(1)} h_j^{(1)}\right), \quad (2)
\end{aligned}$$

where $g(x) = 1/(1 + \exp(-x))$ is the logistic function.

The derivative of the log-likelihood with respect to the model parameters $\boldsymbol{\psi}$ can be obtained from Eq. 1:

$$\begin{aligned}
\frac{\partial \log P(\mathbf{v}; \boldsymbol{\psi})}{\partial \mathbf{W}^{(1)}} &= \mathrm{E}_{P_{\mathrm{data}}}[\mathbf{v}\mathbf{h}^{(1)\top}] - \mathrm{E}_{P_{\mathrm{model}}}[\mathbf{v}\mathbf{h}^{(1)\top}], \quad (3) \\
\frac{\partial \log P(\mathbf{v}; \boldsymbol{\psi})}{\partial \mathbf{W}^{(2)}} &= \mathrm{E}_{P_{\mathrm{data}}}[\mathbf{h}^{(1)}\mathbf{h}^{(2)\top}] - \mathrm{E}_{P_{\mathrm{model}}}[\mathbf{h}^{(1)}\mathbf{h}^{(2)\top}], \\
\frac{\partial \log P(\mathbf{v}; \boldsymbol{\psi})}{\partial \mathbf{W}^{(3)}} &= \mathrm{E}_{P_{\mathrm{data}}}[\mathbf{h}^{(2)}\mathbf{h}^{(3)\top}] - \mathrm{E}_{P_{\mathrm{model}}}[\mathbf{h}^{(2)}\mathbf{h}^{(3)\top}],
\end{aligned}$$

where $\mathrm{E}_{P_{\mathrm{data}}}[\cdot]$ denotes an expectation with respect to the completed data distribution $P_{\mathrm{data}}(\mathbf{h}, \mathbf{v}; \boldsymbol{\psi}) = P(\mathbf{h}|\mathbf{v}; \boldsymbol{\psi}) P_{\mathrm{data}}(\mathbf{v})$, with $P_{\mathrm{data}}(\mathbf{v}) = \frac{1}{N} \sum_n \delta_{\mathbf{v}_n}$ representing the empirical distribution, and $\mathrm{E}_{P_{\mathrm{model}}}[\cdot]$ is an expectation with respect to the distribution defined by the model (Eq. 1). We will sometimes refer to $\mathrm{E}_{P_{\mathrm{data}}}[\cdot]$ as the *data-dependent expectation*, and $\mathrm{E}_{P_{\mathrm{model}}}[\cdot]$ as the *model's expectation*.

Exact maximum likelihood learning in this model is intractable. The exact computation of the data-dependent expectation takes time that is exponential in the number of hidden units, whereas the exact computation of the models expectation takes time that is exponential in the number of hidden and visible units.

1. For clarity, we use three hidden layers. Extensions to models with more than three layers is trivial.

2. We have omitted the bias terms for clarity of presentation. Biases are equivalent to weights on a connection to a unit whose state is fixed at 1.

2.1 Approximate Learning

The original learning algorithm for Boltzmann machines used randomly initialized Markov chains in order to approximate both expectations in order to estimate gradients of the likelihood function [14]. However, this learning procedure is too slow to be practical. Recently, [29]. proposed a variational approach, where mean-field inference is used to estimate data-dependent expectations and an MCMC based stochastic approximation procedure is used to approximate the models expected sufficient statistics.

2.1.1 A Variational Approach to Estimating the Data-dependent Statistics

Consider any approximating distribution $Q(\mathbf{h}|\mathbf{v};\boldsymbol{\mu})$, parameterized by a vector of parameters $\boldsymbol{\mu}$, for the posterior $P(\mathbf{h}|\mathbf{v};\boldsymbol{\psi})$. Then the log-likelihood of the DBM model has the following variational lower bound:

$$\begin{aligned}\log P(\mathbf{v};\boldsymbol{\psi}) &\geq \sum_{\mathbf{h}} Q(\mathbf{h}|\mathbf{v};\boldsymbol{\mu}) \log P(\mathbf{v},\mathbf{h};\boldsymbol{\psi}) + \mathcal{H}(Q) \quad (4)\\ &\geq \log P(\mathbf{v};\boldsymbol{\psi}) - \mathrm{KL}(Q(\mathbf{h}|\mathbf{v};\boldsymbol{\mu})||P(\mathbf{h}|\mathbf{v};\boldsymbol{\psi})),\end{aligned}$$

where $\mathcal{H}(\cdot)$ is the entropy functional, and $\mathrm{KL}(Q||P)$ denotes the Kullback-Leibler divergence between the two distributions. The bound becomes tight if and only if $Q(\mathbf{h}|\mathbf{v};\boldsymbol{\mu}) = P(\mathbf{h}|\mathbf{v};\boldsymbol{\psi})$.

Variational learning has the nice property that in addition to maximizing the log-likelihood of the data, it also attempts to find parameters that minimize the Kullback-Leibler divergence between the approximating and true posteriors.

For simplicity and speed, we approximate the true posterior $P(\mathbf{h}|\mathbf{v};\boldsymbol{\psi})$ with a fully factorized approximating distribution over the three sets of hidden units, which corresponds to so-called mean-field approximation:

$$Q^{MF}(\mathbf{h}|\mathbf{v};\boldsymbol{\mu}) = \prod_{j=1}^{F_1}\prod_{l=1}^{F_2}\prod_{k=1}^{F_3} q(h_j^{(1)}|\mathbf{v})q(h_l^{(2)}|\mathbf{v})q(h_k^{(3)}|\mathbf{v}), \quad (5)$$

where $\boldsymbol{\mu} = \{\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}, \boldsymbol{\mu}^{(3)}\}$ are the mean-field parameters with $q(h_i^{(l)} = 1) = \mu_i^{(l)}$ for $l = 1, 2, 3$. In this case the variational lower bound on the log-probability of the data takes a particularly simple form:

$$\begin{aligned}\log P(\mathbf{v};\boldsymbol{\psi}) &\geq \sum_{\mathbf{h}} Q^{MF}(\mathbf{h}|\mathbf{v};\boldsymbol{\mu}) \log P(\mathbf{v},\mathbf{h};\boldsymbol{\psi}) + \mathcal{H}(Q^{MF})\\ &\geq \mathbf{v}^\top \mathbf{W}^{(1)} \boldsymbol{\mu}^{(1)} + {\boldsymbol{\mu}^{(1)}}^\top \mathbf{W}^{(2)} \boldsymbol{\mu}^{(2)} +\\ &\quad + {\boldsymbol{\mu}^{(2)}}^\top \mathbf{W}^{(3)} \boldsymbol{\mu}^{(3)} - \log \mathcal{Z}(\boldsymbol{\psi}) + \mathcal{H}(Q^{MF}). \quad (6)\end{aligned}$$

Learning proceeds as follows. For each training example, we maximize this lower bound with respect to the variational parameters $\boldsymbol{\mu}$ for fixed parameters $\boldsymbol{\psi}$, which results in the mean-field fixed-point equations:

$$\mu_j^{(1)} \leftarrow g\Big(\sum_{i=1}^{D} W_{ij}^{(1)} v_i + \sum_{l=1}^{F_2} W_{jl}^{(2)} \mu_l^{(2)}\Big), \quad (7)$$

$$\mu_l^{(2)} \leftarrow g\Big(\sum_{j=1}^{F_1} W_{jl}^{(2)} \mu_j^{(1)} + \sum_{k=1}^{F_3} W_{lk}^{(3)} \mu_k^{(3)}\Big), \quad (8)$$

$$\mu_k^{(3)} \leftarrow g\Big(\sum_{l=1}^{F_2} W_{lk}^{(3)} \mu_l^{(2)}\Big), \quad (9)$$

where $g(x) = 1/(1+\exp(-x))$ is the logistic function. To solve these fixed-point equations, we simply cycle through layers, updating the mean-field parameters within a single layer. Note the close connection between the form of the mean-field fixed point updates and the form of the conditional distribution[3] defined by Eq. 2.

2.1.2 A Stochastic Approximation Approach for Estimating the Data-independent Statistics

Given the variational parameters $\boldsymbol{\mu}$, the model parameters $\boldsymbol{\psi}$ are then updated to maximize the variational bound using an MCMC-based stochastic approximation [29], [39], [46].

Learning with stochastic approximation is straightforward. Let $\boldsymbol{\psi}_t$ and $\mathbf{x}_t = \{\mathbf{v}_t, \mathbf{h}_t^{(1)}, \mathbf{h}_t^{(2)}, \mathbf{h}_t^{(3)}\}$ be the current parameters and the state. Then $\mathbf{x}_t$ and $\boldsymbol{\psi}_t$ are updated sequentially as follows:

- Given $\mathbf{x}_t$, sample a new state $\mathbf{x}_{t+1}$ from the transition operator $T_{\boldsymbol{\psi}_t}(\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t)$ that leaves $P(\cdot;\boldsymbol{\psi}_t)$ invariant. This can be accomplished by using Gibbs sampling (see Eq. 2).
- A new parameter $\boldsymbol{\psi}_{t+1}$ is then obtained by making a gradient step, where the intractable model's expectation $\mathrm{E}_{P_{\mathrm{model}}}[\cdot]$ in the gradient is replaced by a point estimate at sample $\mathbf{x}_{t+1}$.

In practice, we typically maintain a set of M "persistent" sample particles $X_t = \{\mathbf{x}_{t,1}, ..., \mathbf{x}_{t,M}\}$, and use an average over those particles. The overall learning procedure for DBMs is summarized in Algorithm 1.

Stochastic approximation provides asymptotic convergence guarantees and belongs to the general class of Robbins–Monro approximation algorithms [27], [46]. Precise sufficient conditions that ensure almost sure convergence to an asymptotically stable point are given in [45]–[47]. One necessary condition requires the learning rate to decrease with time, so that $\sum_{t=0}^{\infty} \alpha_t = \infty$ and $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$. This condition can, for example, be satisfied simply by setting $\alpha_t = a/(b+t)$, for positive constants $a > 0$, $b > 0$. Other conditions ensure that the speed of convergence of the Markov chain, governed by the transition operator $T_{\boldsymbol{\psi}}$, does not decrease too fast as $\boldsymbol{\psi}$ tends to infinity. Typically, in practice, the sequence $|\boldsymbol{\psi}^t|$ is bounded, and the Markov chain, governed by the transition kernel $T_{\boldsymbol{\psi}}$, is ergodic. Together with the condition on the learning rate, this ensures almost sure convergence of the stochastic approximation algorithm to an asymptotically stable point.

3. Implementing the the mean-field requires no extra work beyond implementing the Gibbs sampler.

Algorithm 1 Learning Procedure for a Deep Boltzmann Machine with Three Hidden Layers.

Given: a training set of N binary data vectors $\{\mathbf{v}\}_{n=1}^{N}$, and M, the number of persistent Markov chains (*i.e.* particles).
Randomly initialize parameter vector $\boldsymbol{\psi}_0$ and M samples: $\{\tilde{\mathbf{v}}_{0,1}, \tilde{\mathbf{h}}_{0,1}\}, ..., \{\tilde{\mathbf{v}}_{0,M}, \tilde{\mathbf{h}}_{0,M}\}$, where $\tilde{\mathbf{h}} = \{\tilde{\mathbf{h}}^{(1)}, \tilde{\mathbf{h}}^{(2)}, \tilde{\mathbf{h}}^{(3)}\}$.
for $t = 0$ to T (number of iterations) **do**

// Variational Inference:
for each training example $\mathbf{v}_n$, $n = 1$ to N **do**
Randomly initialize $\boldsymbol{\mu} = \{\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}, \boldsymbol{\mu}^{(3)}\}$ and run mean-field updates until convergence, using Eqs. 7, 8, 9.
Set $\boldsymbol{\mu}_n = \boldsymbol{\mu}$.
end for

// Stochastic Approximation:
for each sample $m = 1$ to M (number of persistent Markov chains) **do**
Sample $(\tilde{\mathbf{v}}_{t+1,m}, \tilde{\mathbf{h}}_{t+1,m})$ given $(\tilde{\mathbf{v}}_{t,m}, \tilde{\mathbf{h}}_{t,m})$ by running a Gibbs sampler for one step (Eq. 2).
end for

// Parameter Update:

$$\mathbf{W}_{t+1}^{(1)} = \mathbf{W}_t^{(1)} + \alpha_t \Big(\tfrac{1}{N}\textstyle\sum_{n=1}^{N} \mathbf{v}_n (\boldsymbol{\mu}_n^{(1)})^\top - \tfrac{1}{M}\sum_{m=1}^{M} \tilde{\mathbf{v}}_{t+1,m} (\tilde{\mathbf{h}}_{t+1,m}^{(1)})^\top \Big).$$

$$\mathbf{W}_{t+1}^{(2)} = \mathbf{W}_t^{(2)} + \alpha_t \Big(\tfrac{1}{N}\textstyle\sum_{n=1}^{N} \boldsymbol{\mu}_n^{(1)} (\boldsymbol{\mu}_n^{(2)})^\top - \tfrac{1}{M}\sum_{m=1}^{M} \tilde{\mathbf{h}}_{t+1,m}^{(1)} (\tilde{\mathbf{h}}_{t+1,m}^{(2)})^\top \Big).$$

$$\mathbf{W}_{t+1}^{(3)} = \mathbf{W}_t^{(3)} + \alpha_t \Big(\tfrac{1}{N}\textstyle\sum_{n=1}^{N} \boldsymbol{\mu}_n^{(2)} (\boldsymbol{\mu}_n^{(3)})^\top - \tfrac{1}{M}\sum_{m=1}^{M} \tilde{\mathbf{h}}_{t+1,m}^{(2)} (\tilde{\mathbf{h}}_{t+1,m}^{(3)})^\top \Big).$$

Decrease α_t.
end for

2.1.3 Greedy Layerwise Pretraining of DBMs

The learning procedure for Deep Boltzmann Machines described above can be used by starting with randomly initialized weights, but it works much better if the weights are initialized sensibly. We therefore use a greedy layer-wise pretraining strategy by learning a stack of modified Restricted Boltzmann Machines (RBMs) (for details see [29]).

This pretraining procedure is quite similar to the pretraining procedure of Deep Belief Networks [12], and it allows us to perform approximate inference by a single bottom-up pass. This fast approximate inference is then used to initialize the mean-field, which then converges much faster than mean-field with random initialization[4].

2.2 Gaussian-Bernoulli DBMs

We now briefly describe a Gaussian-Bernoulli DBM model, which we will use to model real-valued data, such as images of natural scenes and motion capture data. Gaussian-Bernoulli DBMs represent a generalization of a simpler class of models, called Gaussain-Bernoulli Restricted Boltzmann Machines, which have been successfully applied to various tasks including image classification, video action recognition, and speech recognition [16], [20], [23], [35].

4. The code for pretraining and generative learning of the DBM model is available at http://www.utstat.toronto.edu/∼rsalakhu/DBM.html

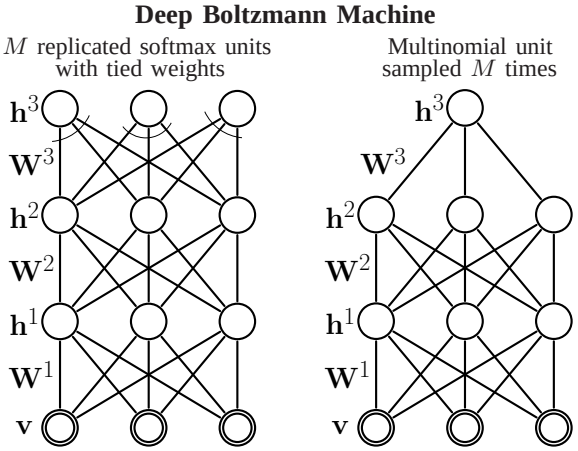

Fig. 1. **Left:** Multinomial DBM model: the top layer represents M softmax hidden units $\mathbf{h}^{(3)}$, which share the same set of weights. **Right:** A different interpretation: M softmax units are replaced by a single multinomial unit which is sampled M times.

In particular, consider modelling visible real-valued units $\mathbf{v} \in \mathbb{R}^D$ and let $\mathbf{h}^{(1)} \in \{0,1\}^{F_1}$, $\mathbf{h}^{(2)} \in \{0,1\}^{F_2}$, and $\mathbf{h}^{(3)} \in \{0,1\}^{F_3}$ be binary stochastic hidden units. The energy of the joint configuration $\{\mathbf{v}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}\}$ of the three-hidden-layer Gaussian-Bernoulli DBM is defined as follows:

$$E(\mathbf{v}, \mathbf{h}; \boldsymbol{\psi}) = \frac{1}{2}\sum_i \frac{v_i^2}{\sigma_i^2} - \sum_{ij} W_{ij}^{(1)} h_j^{(1)} \frac{v_i}{\sigma_i} - \sum_{jl} W_{jl}^{(2)} h_j^{(1)} h_l^{(2)} - \sum_{lk} W_{lk}^{(3)} h_l^{(2)} \hat{h}_k^{(3)}, \quad (10)$$

where $\mathbf{h} = \{\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}\}$ represent the set of hidden units, and $\boldsymbol{\psi} = \{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{W}^{(3)}, \boldsymbol{\sigma}^2\}$ are the model parameters, and σ_i^2 is the variance of input i. The marginal distribution over the visible vector $\mathbf{v}$ takes form:

$$P(\mathbf{v}; \boldsymbol{\psi}) = \sum_{\mathbf{h}} \frac{\exp\left(-E(\mathbf{v}, \mathbf{h}; \boldsymbol{\psi})\right)}{\int_{\mathbf{v}'} \sum_{\mathbf{h}} \exp\left(-E(\mathbf{v}, \mathbf{h}; \boldsymbol{\psi})\right) d\mathbf{v}'}. \quad (11)$$

From Eq. 10, it is straightforward to derive the following conditional distributions:

$$p(v_i = x | \mathbf{h}^{(1)}) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{\left(x - \sigma_i \sum_j h_j^{(1)} W_{ij}^{(1)}\right)^2}{2\sigma_i^2} \right),$$

$$p(h_j^{(1)} = 1 | \mathbf{v}) = g\left(\sum_i W_{ij}^{(1)} \frac{v_i}{\sigma_i} \right), \quad (12)$$

where $g(x) = 1/(1 + \exp(-x))$ is the logistic function. Conditional distributions over $\mathbf{h}^{(2)}$ and $\mathbf{h}^{(3)}$ remain the same as in the standard DBM model (see Eq. 2).

Observe that conditioned on the states of the hidden units (Eq. 12), each visible unit is modelled by a Gaussian distribution, whose mean is shifted by the weighted combination of the hidden unit activations. The derivative of the log-likelihood

with respect to $\mathbf{W}^{(1)}$ takes form:

$$\frac{\partial \log P(\mathbf{v};\boldsymbol{\psi})}{\partial W_{ij}^{(1)}} = \mathrm{E}_{P_{\mathrm{data}}}\left[\frac{1}{\sigma_i} v_i h_j^{(1)}\right] - \mathrm{E}_{P_{\mathrm{Model}}}\left[\frac{1}{\sigma_i} v_i h_j^{(1)}\right].$$

The derivatives with respect to parameters $\mathbf{W}^{(2)}$ and $\mathbf{W}^{(3)}$ remain the same as in Eq. 3.

As described in previous section, learning of the model parameters, including the variances $\boldsymbol{\sigma}^2$, can be carried out using variational learning together with stochastic approximation procedure. In practice, however, instead of learning $\boldsymbol{\sigma}^2$, one would typically use a fixed, predetermined value for $\boldsymbol{\sigma}^2$ ([13], [24]).

2.3 Multinomial DBMs

To allow DBMs to express more information and introduce more structured hierarchical priors, we will use a conditional multinomial distribution to model activities of the top-level units $\mathbf{h}^{(3)}$. Specifically, we will use M softmax units, each with "1-of-K" encoding, so that each unit contains a set of K weights. We represent the k^{th} discrete value of hidden unit by a vector containing 1 at the k^{th} location and zeros elsewhere. The conditional probability of a softmax top-level unit is:

$$P(h_k^{(3)}|\mathbf{h}^{(2)}) = \frac{\exp\left(\sum_l W_{lk}^{(3)} h_l^{(2)}\right)}{\sum_{s=1}^{K} \exp\left(\sum_l W_{ls}^{(3)} h_l^{(2)}\right)}. \quad (13)$$

In our formulation, all M separate softmax units will share the same set of weights, connecting them to binary hidden units at the lower-level (Fig. 1). The energy of the state $\{\mathbf{v}, \mathbf{h}\}$ is then defined as follows:

$$E(\mathbf{v},\mathbf{h};\boldsymbol{\psi}) = -\sum_{ij} W_{ij}^{(1)} v_i h_j^{(1)} - \sum_{jl} W_{jl}^{(2)} h_j^{(1)} h_l^{(2)} - \sum_{lk} W_{lk}^{(3)} h_l^{(2)} \hat{h}_k^{(3)},$$

where $\mathbf{h}^{(1)} \in \{0,1\}^{F_1}$ and $\mathbf{h}^{(2)} \in \{0,1\}^{F_2}$ represent stochastic binary units. The top layer is represented by the M softmax units $\mathbf{h}^{(3,m)}$, $m = 1,..,M$, with $\hat{h}_k^{(3)} = \sum_{m=1}^{M} h_k^{(3,m)}$ denoting the count for the k^{th} discrete value of a hidden unit.

A key observation is that M separate copies of softmax units that all share the same set of weights can be viewed as a single multinomial unit that is sampled M times from the conditional distribution of Eq. 13. This gives us a familiar "bag-of-words" representation [30], [36]. A pleasing property of using softmax units is that the mathematics underlying the learning algorithm for binary-binary DBMs remains the same.

3 Compound HDP-DBM model

After a DBM model has been learned, we have an undirected model that defines the joint distribution $P(\mathbf{v}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)})$. One way to express what has been learned is the conditional model $P(\mathbf{v}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}|\mathbf{h}^{(3)})$ and a complicated prior term $P(\mathbf{h}^{(3)})$, defined by the DBM model. We can therefore rewrite the variational bound as:

$$\log P(\mathbf{v}) \geq \sum_{\mathbf{h}^{(1)},\mathbf{h}^{(2)},\mathbf{h}^{(3)}} Q(\mathbf{h}|\mathbf{v};\boldsymbol{\mu}) \log P(\mathbf{v}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}|\mathbf{h}^{(3)}) + \mathcal{H}(Q) + \sum_{\mathbf{h}^{(3)}} Q(\mathbf{h}^{(3)}|\mathbf{v};\boldsymbol{\mu}) \log P(\mathbf{h}^{(3)}). \quad (14)$$

This particular decomposition lies at the core of the greedy recursive pretraining algorithm: we keep the learned conditional model $P(\mathbf{v}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}|\mathbf{h}^{(3)})$, but maximize the variational lower-bound of Eq. 14 with respect to the last term [12]. This maximization amounts to replacing $P(\mathbf{h}^{(3)})$ by a prior that is closer to the average, over all the data vectors, of the approximate conditional posterior $Q(\mathbf{h}^{(3)}|\mathbf{v})$.

Instead of adding an additional undirected layer (e.g. a restricted Boltzmann machine), to model $P(\mathbf{h}^{(3)})$ we can place a hierarchical Dirichlet process prior over $\mathbf{h}^{(3)}$, that will allow us to learn category hierarchies, and more importantly, useful representations of classes that contain few training examples.

The part we keep, $P(\mathbf{v}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}|\mathbf{h}^{(3)})$, represents a *conditional* DBM model[5]:

$$P(\mathbf{v}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}|\mathbf{h}^{(3)}) = \frac{1}{\mathcal{Z}(\boldsymbol{\psi}, \mathbf{h}^{(3)})} \exp\left(\sum_{ij} W_{ij}^{(1)} v_i h_j^{(1)} + \sum_{jl} W_{jl}^{(2)} h_j^{(1)} h_l^{(2)} + \sum_{lm} W_{lm}^{(3)} h_l^{(2)} h_m^{(3)}\right), \quad (15)$$

which can be viewed as a two-layer DBM but with bias terms given by the states of $\mathbf{h}^{(3)}$.

3.1 A Hierarchical Bayesian Prior

In a typical hierarchical topic model, we observe a set of N documents, each of which is modelled as a mixture over topics, that are shared among documents. Let there be K words in the vocabulary. A topic t is a discrete distribution over K words with probability vector $\boldsymbol{\phi}_t$. Each document n has its own distribution over topics given by probabilities $\boldsymbol{\theta}_n$.

In our compound HDP-DBM model, we will use a hierarchical topic model as a prior over the activities of the DBM's top-level features. Specifically, the term "document" will refer to the top-level multinomial unit $\mathbf{h}^{(3)}$, and M "words" in the document will represent the M samples, or active DBM's top-level features, generated by this multinomial unit. Words in each document are drawn by choosing a topic t with probability θ_{nt}, and then choosing a word w with probability ϕ_{tw}. We will often refer to topics as our learned *higher-level features*, each of which defines a topic specific distribution over DBM's $\mathbf{h}^{(3)}$ features. Let $h_{in}^{(3)}$ be the i^{th} word in document n, and x_{in} be its topic. We can specify the following prior over $\mathbf{h}^{(3)}$:

$$\begin{aligned}
\boldsymbol{\theta}_n|\boldsymbol{\pi} &\sim \mathrm{Dir}(\alpha\boldsymbol{\pi}), && \text{for each document } n{=}1,..,N \\
\boldsymbol{\phi}_t|\boldsymbol{\tau} &\sim \mathrm{Dir}(\beta\boldsymbol{\tau}), && \text{for each topic } t{=}1,..,T \\
x_{in}|\boldsymbol{\theta}_n &\sim \mathrm{Mult}(1,\boldsymbol{\theta}_n), && \text{for each word } i{=}1,..,M \\
h_{in}^{(3)}|x_{in},\boldsymbol{\phi}_{x_{in}} &\sim \mathrm{Mult}(1,\boldsymbol{\phi}_{x_{in}}),
\end{aligned}$$

5. Our experiments reveal that using Deep Belief Networks instead of DBMs decreased model performance.

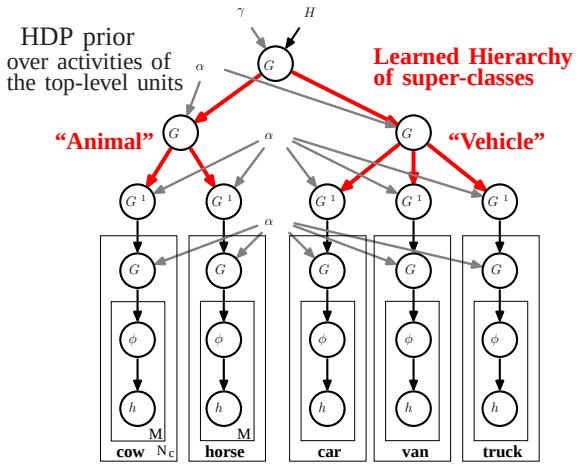

Fig. 2. Hierarchical Dirichlet Process prior over the states of the DBM's top-level features $\mathbf{h}^{(3)}$.

where $\boldsymbol{\pi}$ is the global distribution over topics, $\boldsymbol{\tau}$ is the global distribution over K words, and α and β are concentration parameters.

Let us further assume that our model is presented with a fixed two-level category hierarchy. In particular, suppose that N documents, or objects, are partitioned into C basic level categories (e.g. cow, sheep, car). We represent such partition by a vector $\mathbf{z}^b$ of length N, each entry of which is $z_n^b \in \{1, ..., C\}$. We also assume that our C basic-level categories are partitioned into S super-categories (e.g. animal, vehicle), represented by a vector $\mathbf{z}^s$ of length C, with $z_c^s \in \{1, ..., S\}$. These partitions define a *fixed two-level* tree hierarchy (Fig. 2). We will relax this assumption later by placing a nonparametric prior over the category assignments.

The hierarchical topic model can be readily extended to modelling the above hierarchy. For each document n that belongs to the basic category c, we place a common Dirichlet prior over $\boldsymbol{\theta}_n$ with parameters $\boldsymbol{\pi}_c^{(1)}$. The Dirichlet parameters $\boldsymbol{\pi}^{(1)}$ are themselves drawn from a Dirichlet prior with level-2 parameters $\boldsymbol{\pi}^{(2)}$, common to all basic-level categories that belong to the same super-category, and so on. Specifically, we define the following hierarchical prior over $\mathbf{h}^{(3)}$:

$$
\begin{aligned}
\boldsymbol{\pi}_s^{(2)}|\boldsymbol{\pi}_g^{(3)} &\sim \text{Dir}(\alpha^{(3)}\boldsymbol{\pi}_g^{(3)}), \quad \text{for each super-class } s{=}1,..,S \\
\boldsymbol{\pi}_c^{(1)}|\boldsymbol{\pi}_{z_c^s}^{(2)} &\sim \text{Dir}(\alpha^{(2)}\boldsymbol{\pi}_{z_c^s}^{(2)}), \quad \text{for each basic-class } c{=}1,..,C \\
\boldsymbol{\theta}_n|\boldsymbol{\pi}_{z_n^b}^{(1)} &\sim \text{Dir}(\alpha^{(1)}\boldsymbol{\pi}_{z_n^b}^{(1)}), \quad \text{for each document } n{=}1,..,N \\
x_{in}|\boldsymbol{\theta}_n &\sim \text{Mult}(1, \boldsymbol{\theta}_n), \quad \text{for each word } i{=}1,..,M \\
\boldsymbol{\phi}_t|\beta, \boldsymbol{\tau} &\sim \text{Dir}(\beta\boldsymbol{\tau}), \\
h_{in}^{(3)}|x_{in}, \boldsymbol{\phi}_{x_{in}} &\sim \text{Mult}(1, \boldsymbol{\phi}_{x_{in}}),
\end{aligned} \tag{16}
$$

where $\boldsymbol{\pi}_g^{(3)}$ is the global distribution over topics, $\boldsymbol{\pi}_s^{(2)}$ is the super-category specific and $\boldsymbol{\pi}_c^{(1)}$ is the class specific distribution over topics, or higher-level features. These high-level features, in turn, define topic-specific distribution over $\mathbf{h}^{(3)}$ features, or "words" in our DBM model. Finally, $\alpha^{(1)}$, $\alpha^{(2)}$, and $\alpha^{(3)}$ represent concentration parameters describing how close $\boldsymbol{\pi}$'s are to their respective prior means within the hierarchy.

For a fixed number of topics T, the above model represents a hierarchical extension of the Latent Dirichlet Allocation (LDA) model [4]. However, we typically do not know the number of topics a-priori. It is therefore natural to consider a nonparametric extension based on the HDP model [38], which allows for a countably infinite number of topics. In the standard hierarchical Dirichlet process notation, we have the following:

$$
\begin{aligned}
G_g^{(3)}|\beta, \gamma, \tau &\sim \text{DP}(\gamma, \text{Dir}(\beta\boldsymbol{\tau})), \\
G_s^{(2)}|\alpha^{(3)}, G^{(3)} &\sim \text{DP}(\alpha^{(3)}, G_g^{(3)}), \\
G_c^{(1)}|\alpha^{(2)}, G^{(2)} &\sim \text{DP}(\alpha^{(2)}, G_{z_c^s}^{(2)}), \\
G_n|\alpha^{(1)}, G^{(1)} &\sim \text{DP}(\alpha^{(1)}, G_{z_n^b}^{(1)}), \\
\boldsymbol{\phi}_{in}^*|G_n &\sim G_n, \\
h_{in}^3|\boldsymbol{\phi}_{in}^* &\sim \text{Mult}(1, \boldsymbol{\phi}_{in}^*),
\end{aligned} \tag{17}
$$

where $\text{Dir}(\beta\boldsymbol{\tau})$ is the base-distribution, and each $\boldsymbol{\phi}^*$ is a factor associated with a single observation $h_{in}^{(3)}$. Making use of topic index variables x_{in}, we denote $\boldsymbol{\phi}_{in}^* = \boldsymbol{\phi}_{x_{in}}$ (see Eq. 16). Using a stick-breaking representation we can write:

$$
\begin{aligned}
G_g^{(3)}(\phi) &= \sum_{t=1}^{\infty} \pi_{gt}^{(3)} \delta_{\phi_t}, \qquad G_s^{(2)}(\phi) = \sum_{t=1}^{\infty} \pi_{st}^{(2)} \delta_{\phi_t}, \\
G_c^{(1)}(\phi) &= \sum_{t=1}^{\infty} \pi_{ct}^{(1)} \delta_{\phi_t}, \qquad G_n(\phi) = \sum_{t=1}^{\infty} \theta_{nt} \delta_{\phi_t},
\end{aligned} \tag{18}
$$

that represent sums of point masses. We also place Gamma priors over concentration parameters as in [38].

The overall generative model is shown in Fig. 2. To generate a sample we first draw M words, or activations of the top-level features, from the HDP prior over $\mathbf{h}^{(3)}$ given by Eq. 17. Conditioned on $\mathbf{h}^{(3)}$, we sample the states of $\mathbf{v}$ from the conditional DBM model given by Eq. 15.

3.2 Modelling the number of super-categories

So far we have assumed that our model is presented with a two-level partition $\mathbf{z} = \{\mathbf{z}^s, \mathbf{z}^b\}$ that defines a fixed two-level tree hierarchy. We note that this model corresponds to a standard HDP model that assumes a fixed hierarchy for sharing parameters. If, however, we are not given any level-1 or level-2 category labels, we need to infer the distribution over the possible category structures. We place a nonparametric two-level nested Chinese Restaurant Prior (CRP) [5] over $\mathbf{z}$, which defines a prior over tree structures and is flexible enough to learn arbitrary hierarchies. The main building block of the nested CRP is the Chinese restaurant process, a distribution on partition of integers. Imagine a process by which customers enter a restaurant with an unbounded number of tables, where the n^{th} customer occupies a table k drawn from:

$$
P(z_n = k|z_1, ..., z_{n-1}) = \begin{cases} \frac{n^k}{n-1+\eta} & n^k > 0 \\ \frac{\eta}{n-1+\eta} & k \text{ is new} \end{cases}, \tag{19}
$$

where n^k is the number of previous customers at table k and η is the concentration parameter.

The nested CRP, nCRP(η), extends CRP to nested sequence of partitions, one for each level of the tree. In this case each

observation n is first assigned to the super-category z_n^s using Eq. 19. Its assignment to the basic-level category z_n^b, that is placed under a super-category z_n^s, is again recursively drawn from Eq. 19. We also place a Gamma prior $\Gamma(1,1)$ over η. The proposed model allows for both: a nonparametric prior over potentially unbounded number of global topics, or higher-level features, as well as a nonparametric prior that allow learning an arbitrary tree taxonomy.

Unlike in many conventional hierarchical Bayesian models, here we infer both the model parameters as well as the hierarchy for sharing those parameters. As we show in the experimental results section, both sharing higher-level features and forming coherent hierarchies play a crucial role in the ability of the model to generalize well from one or few examples of a novel category. Our model can be readily used in unsupervised or semi-supervised modes, with varying amounts of label information at different levels of the hierarchy.

4 INFERENCE

Inferences about model parameters at all levels of hierarchy can be performed by MCMC. When the tree structure $\mathbf{z}$ of the model is not given, the inference process will alternate between fixing $\mathbf{z}$ while sampling the space of model parameters, and vice versa.

Sampling HDP parameters: Given the category assignment vector $\mathbf{z}$, and the states of the top-level DBM features $\mathbf{h}^{(3)}$, we use the posterior representation sampler of [37]. In particular, the HDP sampler maintains the stick-breaking weights $\{\boldsymbol{\theta}\}_{n=1}^N$, $\{\boldsymbol{\pi}_c^{(1)}, \boldsymbol{\pi}_s^{(2)}, \boldsymbol{\pi}_g^{(3)}\}$, and topic indicator variables $\mathbf{x}$ (parameters $\boldsymbol{\phi}$ can be integrated out). The sampler alternates between: (a) sampling cluster indices x_{in} using Gibbs updates in the Chinese restaurant franchise (CRF) representation of the HDP; (b) sampling the weights at all three levels conditioned on $\mathbf{x}$ using the usual posterior of a DP.

Conditioned on the draw of the super-class DP $G_s^{(2)}$ and the state of the CRF, the posteriors over $G_c^{(1)}$ become independent. We can easily speed up inference by sampling from these conditionals in parallel. The speedup could be substantial, particularly as the number of the basic-level categories becomes large.

Sampling category assignments $\mathbf{z}$: Given the current instantiation of the stick-breaking weights, for each input n we have:

$$(\theta_{1,n}, ..., \theta_{T,n}, \theta_{new,n}) \sim \text{Dir}(\alpha^{(1)}\pi^{(1)}_{\mathbf{z}_n,1}, ..., \alpha^{(1)}\pi^{(1)}_{\mathbf{z}_n,T}, \alpha^{(1)}\pi^{(1)}_{\mathbf{z}_n,new}). \quad (20)$$

Combining the above likelihood term with the CRP prior (Eq. 19), the posterior over the category assignment can be calculated as follows:

$$p(\mathbf{z}_n|\boldsymbol{\theta}_n, \mathbf{z}_{-n}, \boldsymbol{\pi}^{(1)}) \propto p(\boldsymbol{\theta}_n|\boldsymbol{\pi}^{(1)}, \mathbf{z}_n)p(\mathbf{z}_n|\mathbf{z}_{-n}), \quad (21)$$

where $\mathbf{z}_{-n}$ denotes variables $\mathbf{z}$ for all observations other than n. When computing the probability of placing $\boldsymbol{\theta}_n$ under a newly created category, its parameters are sampled from the prior.

Sampling DBM's hidden units: Given the states of the DBM's top-level multinomial unit $\mathbf{h}_n^{(3)}$, conditional samples from $P(\mathbf{h}_n^{(1)}, \mathbf{h}_n^{(2)}|\mathbf{h}_n^{(3)}, \mathbf{v}_n)$ can be obtained by running a Gibbs sampler that alternates between sampling the states of $\mathbf{h}_n^{(1)}$ independently given $\mathbf{h}_n^{(2)}$, and vice versa. Conditioned on topic assignments x_{in} and $\mathbf{h}_n^{(2)}$, the states of the multinomial unit $\mathbf{h}_n^{(3)}$ for each input n are sampled using Gibbs conditionals:

$$P(\mathbf{h}_{in}^{(3)}|\mathbf{h}_n^{(2)}, \mathbf{h}_{-in}^{(3)}, \mathbf{x}_n) \propto P(\mathbf{h}_n^{(2)}|\mathbf{h}_n^{(3)})P(\mathbf{h}_{in}^{(3)}|\mathbf{x}_{in}), \quad (22)$$

where the first term is given by the product of logistic functions (see Eq. 15):

$$P(\mathbf{h}_n^{(2)}|\mathbf{h}_n^{(3)}) = \prod_l P(h_{ln}^{(2)}|\mathbf{h}_n^{(3)}), \quad \text{with} \quad (23)$$

$$P(h_l^{(2)} = 1|\mathbf{h}^{(3)}) = \frac{1}{1 + \exp\big(-\sum_m W_{lm}^{(3)} h_m^{(3)}\big)},$$

and the second term $P(\mathbf{h}_{in}^{(3)})$ is given by the multinomial: $\text{Mult}(1, \boldsymbol{\phi}_{x_{in}})$ (see Eq. 17). In our conjugate setting, parameters $\boldsymbol{\phi}$ can be further integrated out.

Fine-tuning DBM: Finally, conditioned on the states of $\mathbf{h}^{(3)}$, we can further *fine-tune* low-level DBM parameters $\boldsymbol{\psi} = \{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{W}^{(3)}\}$ by applying approximate maximum likelihood learning (see section 2) to the conditional DBM model of Eq. 15. For the stochastic approximation algorithm, since the partition function depends on the states of $\mathbf{h}^{(3)}$, we maintain one "persistent" Markov chain per data point (for details see [29], [39]). As we show in our experimental results section, fine-tuning low-level DBM features can significantly improve model performance.

4.1 Making predictions

Given a test input $\mathbf{v}_t$, we can quickly infer the approximate posterior over $\mathbf{h}_t^{(3)}$ using the mean-field of Eq. 6, followed by running the full Gibbs sampler to get approximate samples from the posterior over the category assignments. In practice, for faster inference, we fix learned topics ϕ_t and approximate the marginal likelihood that $\mathbf{h}_t^{(3)}$ belongs to category $\mathbf{z}_t$ by assuming that document specific DP can be well approximated by the class-specific[6] DP $G_t \approx G_{\mathbf{z}_t}^{(1)}$ (see Fig. 2). Hence instead of integrating out document specific DP G_t, we approximate:

$$P(\mathbf{h}_t^{(3)}|\mathbf{z}_t, G^{(1)}, \boldsymbol{\phi}) = \int_{G_t} P(\mathbf{h}_t^{(3)}|\boldsymbol{\phi}, G_t)P(G_t|G_{\mathbf{z}_t}^{(1)})dG_t \approx P(\mathbf{h}_t^{(3)}|\boldsymbol{\phi}, G_{\mathbf{z}_t}^{(1)}), \quad (24)$$

which can be computed analytically by integrating out topic assignments x_{in} (Eq. 17). Combining this likelihood term with nCRP prior $P(\mathbf{z}_t|\mathbf{z}_{-t})$ of Eq. 19 allows us to efficiently infer approximate posterior over category assignments. In all of our experimental results, computing this approximate posterior takes a fraction of a second, which is crucial for applications, such as object recognition or information retrieval.

6. We note that $G_{\mathbf{z}_t}^{(1)} = \text{E}[G_t|G_{\mathbf{z}_t}^{(1)}]$

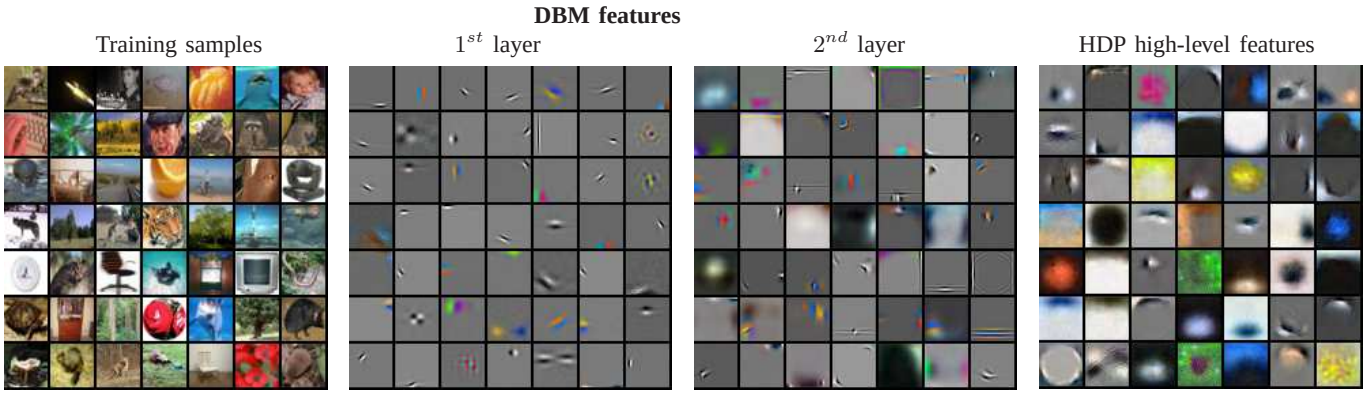

Fig. 3. A random subset of the training images along with the 1^{st} and 2^{nd} layer DBM features, and higher-level class-sensitive HDP features/topics. To visualize higher-level features, we first sample M words from a fixed topic ϕ_t, followed by sampling RGB pixel values from the conditional DBM model.

1. bed, chair, clock, couch, dinosaur, lawn mower, table, telephone, television, wardrobe
2. bus, house, pickup truck, streetcar, tank, tractor, train
3. crocodile, kangaroo, lizard, snake, spider, squirrel
4. hamster, mouse, rabbit, raccoon, possum, bear
5. apple, orange, pear, sunflower, sweet pepper
6. baby, boy, girl, man, woman
7. dolphin, ray, shark, turtle, whale
8. otter, porcupine, shrew, skunk
9. beaver, camel, cattle, chimpanzee, elephant
10. fox, leopard, lion, tiger, wolf
11. maple tree, oak tree, pine tree, willow tree
12 flatfish, seal, trout, worm
13 butterfly, caterpillar, snail
14 bee, crab, lobster
15 bridge, castle, road, skyscraper
16 bicycle, keyboard, motorcycle, orchid, palm tree
17 bottle, bowl, can, cup, lamp
18 cloud, plate, rocket 19. mountain, plain, sea
20 poppy, rose, tulip 21. aquarium fish, mushroom
22 beetle, cockroach 23. forest

Fig. 4. A typical partition of the 100 basic-level categories. Many of the discovered super-categories contain semantically coherent classes.

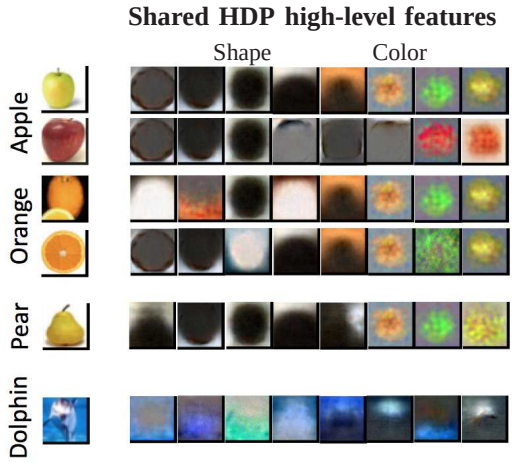

Fig. 5. Learning to Learn: training examples along with eight most probable topics ϕ_t, ordered by hand.

5 Experiments

We present experimental results on the CIFAR-100 [17], handwritten character [18], and human motion capture recognition datasets. For all datasets, we first pretrain a DBM model in unsupervised fashion on raw sensory input (e.g. pixels, or 3D joint angles), followed by fitting an HDP prior, which is run for 200 Gibbs sweeps. We further run 200 additional Gibbs steps in order to fine-tune parameters of the entire compound HDP-DBM model. This was sufficient to obtain good performance. Across all datasets, we also assume that the basic-level category labels are given, but no super-category labels are available. We must infer how to cluster basic categories into super-categories at the same time as we infer parameter values at all levels of the hierarchy. The training set includes many examples of familiar categories but only a few examples of a novel class. Our goal is to generalize well on a novel class.

In all experiments we compare performance of HDP-DBM to the following alternative models. The first two models, stand-alone Deep Boltzmann Machines and Deep Belief Networks (DBNs) [12] used three layers of hidden variables and were pretrained using a stack of RBMs. To evaluate classification performance of DBNs and DBMs, both models were converted into multilayer neural networks and were discriminatively fine-tuned using backpropagation algorithm (see [29] for details). Our third model, "Flat HDP-DBM", always used a single super-category. The Flat HDP-DBM approach, similar in spirit to the one-shot learning model of [11], could potentially identify a set of useful high-level features common to all categories. Our fourth model used a version of SVM that implements cost-sensitive learning[7]. The basic idea is to assign a larger penalty value for misclassifying examples that arise from the under-represented class. In our setting, this model performs slightly better compared to a standard SVM classifier. Our last model used a simple k nearest neighbours (k-NN) classifier. Finally, using HDPs on top of raw sensory input (i.e. pixels, or even image-specific GIST features) performs far worse compared to our HDP-DBM model.

7. We used LIBSVM software package of [7].

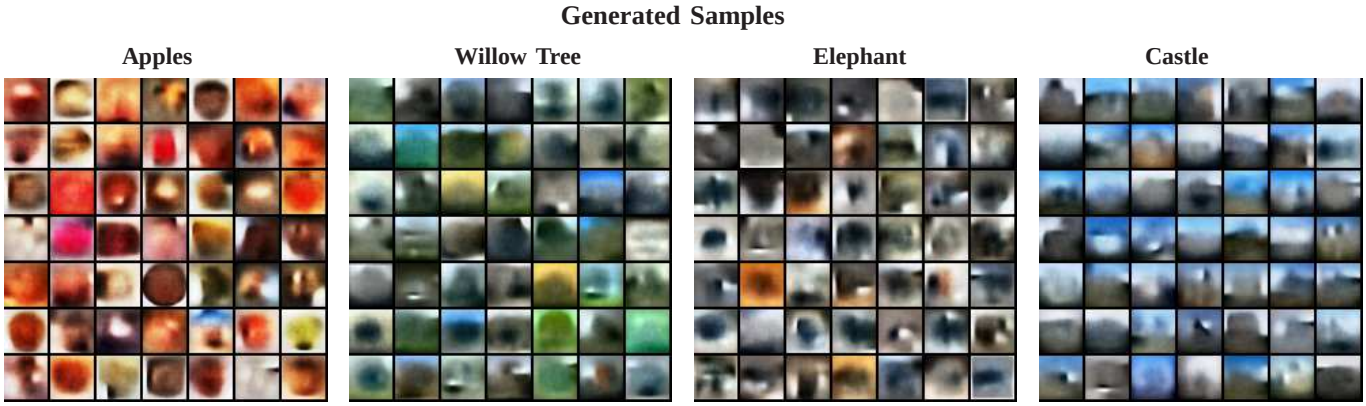

Fig. 6. Class-conditional samples generated from the HDP-DBM model. Observe that the model despite extreme variability, the model is able to capture a coherent structure of each class. See in colour for better visualization.

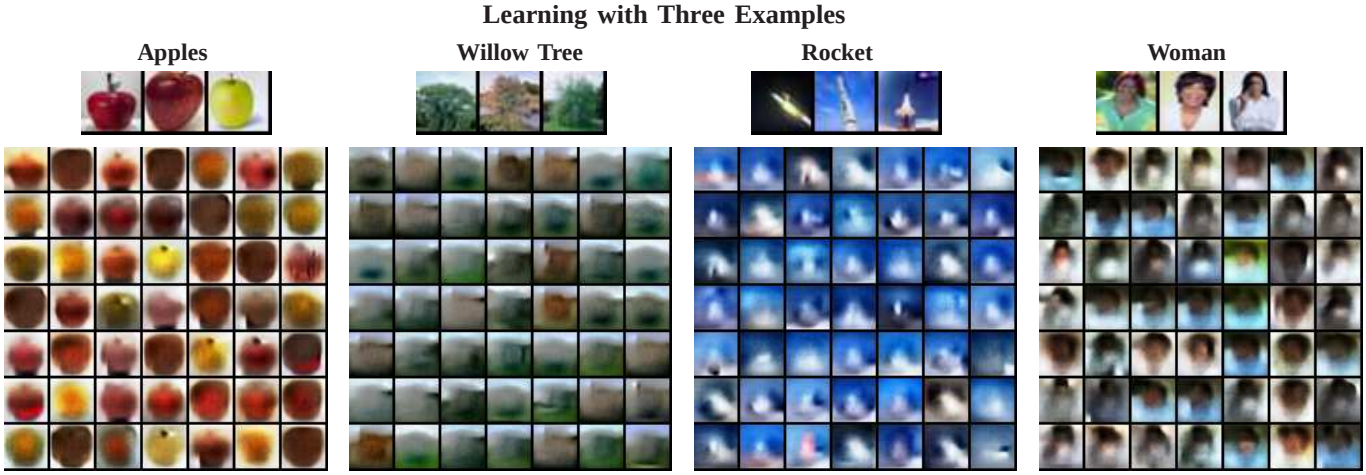

Fig. 7. Conditional samples generated by the HDP-DBM model when learning only with three training examples of a novel class: **Top:** three training examples, **Bottom:** 49 conditional samples. See in colour for better visualization

5.1 CIFAR-100 dataset

The CIFAR-100 image dataset [17] contains 50,000 training and 10,000 test images of 100 object categories (100 per class), with $32 \times 32 \times 3$ RGB pixels. Extreme variability in scale, viewpoint, illumination, and cluttered background makes the object recognition task for this dataset quite difficult. Similar to [17], in order to learn good generic low-level features, we first train a two-layer DBM in completely unsupervised fashion using 4 million tiny images[8] [40]. We use a conditional Gaussian distribution to model observed pixel values [13], [17]. The first DBM layer contained 10,000 binary hidden units, and the second layer contained M=1000 softmax units[9]. We then fit an HDP prior over $\mathbf{h}^{(2)}$ to the 100 object classes. We also experimented with a 3-layer DBM model, as well as various softmax parameters: $M = 500$ and $M = 2000$. The difference in performance was not significant.

Fig. 3 displays a random subset of the training data, 1^{st} and 2^{nd} layer DBM features, as well as higher-level class-sensitive features, or topics, learned by the HDP model. Second layer features were visualized as a weighted linear combination of the first layer features as in [21]. To visualize a particular higher-level feature, we first sample M words from a fixed topic ϕ_t, followed by sampling RGB pixel values from the conditional DBM model. While DBM features capture mostly low-level structure, including edges and corners, the HDP features tend to capture higher-level structure, including contours, shapes, colour components, and surface boundaries in the images. More importantly, features at all levels of the hierarchy evolve without incorporating any image-specific priors. Fig. 4 shows a typical partition over 100 classes that our model discovers with many super-categories containing semantically similar classes.

Table 1 quantifies performance using the area under the ROC curve (AUROC) for classifying 10,000 test images as belonging to the novel vs. all other 99 classes. We report 2*AUROC-1, so zero corresponds to the classifier that makes random predictions. The results are averaged over 100 classes using "leave-one-out" test format. Based on a single example, the HDP-DBM model achieves an AUROC of 0.36, significantly outperforming DBMs, DBNs, SVMs, and 1-NN

8. The dataset contains random images of natural scenes downloaded from the web.

9. The generative training of the DBM model using 4 million images takes about a week on the Intel Xeon 3.00GHz. Fitting an HDP prior to the DBMs top-level features on the CIFAR dataset takes about 12 hours. However, at test time, using variational inference and approximation of Eq. 24, it takes a fraction of a second to classify a test example into its corresponding category.

	CIFAR Dataset Number of examples					Handwritten Characters Number of examples				Motion Capture Number of examples				
Model	1	3	5	10	50	1	3	5	10	1	3	5	10	50
Tuned HDP-DBM	**0.36**	**0.41**	**0.46**	**0.53**	**0.62**	**0.67**	**0.78**	**0.87**	**0.93**	**0.67**	**0.84**	**0.90**	**0.93**	**0.96**
HDP-DBM	0.34	0.39	**0.45**	**0.52**	**0.61**	0.65	0.76	0.85	**0.92**	**0.66**	0.82	0.88	**0.93**	**0.96**
Flat HDP-DBM	0.27	0.37	0.42	0.50	**0.61**	0.58	0.73	0.82	0.89	0.63	0.79	0.86	0.91	**0.96**
DBM	0.26	0.36	0.41	0.48	**0.61**	0.57	0.72	0.81	0.89	0.61	0.79	0.85	0.91	**0.95**
DBN	0.25	0.33	0.37	0.45	**0.60**	0.51	0.72	0.81	0.89	0.61	0.79	0.84	0.92	**0.96**
SVM	0.20	0.29	0.32	0.39	**0.61**	0.43	0.68	0.78	0.87	0.55	0.78	0.85	0.91	**0.96**
1-NN	0.17	0.18	0.19	0.20	0.32	0.43	0.65	0.73	0.81	0.58	0.75	0.81	0.88	0.93
GIST	0.27	0.31	0.33	0.39	0.58	-	-	-	-	-	-	-	-	

TABLE 1

Classification performance on the test set using 2*AUROC-1. The results in bold correspond to ROCs that are statistically indistinguishable from the best (the difference is not statistically significant).

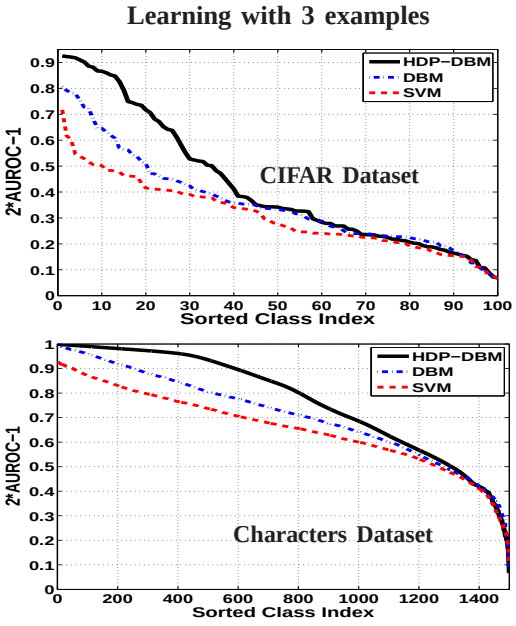

Fig. 8. Performance of HDP-DBM, DBM, and SVMs for all object classes when learning with 3 examples. Object categories are sorted by their performance.

using standard image-specific GIST features[10] that achieve an AUROC of 0.26, 0.25, 0.20 and 0.27 respectively. Table 1 also shows that fine-tuning parameters of *all layers jointly* as well as learning super-category hierarchy significantly improves model performance. As the number of training examples increases, the HDP-DBM model still outperforms alternative methods. With 50 training examples, however, all models perform about the same. This is to be expected, as with more training examples, the effect of the hierarchical prior decreases.

We next illustrate the ability of the HDP-DBM to generalize from a single training example of a "pear" class. We trained the model on 99 classes containing 500 training images each, but only one training example of a "pear" class. Fig. 5 shows the kind of transfer our model is performing, where we display training examples along with eight most probable topics ϕ_t, ordered by hand. The model discovers that pears are like apples and oranges, and not like other classes of images, such as dolphins, that reside in very different parts of the hierarchy. Hence the novel category can inherit the prior distribution over similar high-level shape and colour features, allowing the HDP-DBM to generalize considerably better to new instances of the "pear" class.

We next examined the generative performance of the HDP-DBM model. Fig. 6 shows samples generated by the HDP-DBM model for four classes: "Apple", "Willow Tree", "Elephant", and "Castle". Despite extreme variability in scale, viewpoint, and cluttered background, the model is able to capture the overall structure of each class. Fig. 7 shows conditional samples when learning only with three training examples of a novel class. For example, based on only three training examples of the "Apple" class, the HDP-DBM model is able to generate a rich variety of new apples. Fig. 8 further quantifies performance of HDP-DBM, DBM, and SVM models for all object categories when learning with only three examples. Observe that over 40 classes benefit in various degrees from both: learning a hierarchy as well as learning low and high-level features.

5.2 Handwritten Characters

The handwritten characters dataset [18] can be viewed as the "transpose" of the standard MNIST dataset. Instead of containing 60,000 images of 10 digit classes, the dataset contains 30,000 images of 1500 characters (20 examples each) with 28×28 pixels. These characters are from 50 alphabets from around the world, including Bengali, Cyrillic, Arabic, Sanskrit, Tagalog (see Fig. 9). We split the dataset into 15,000 training and 15,000 test images (10 examples of each class). Similar to the CIFAR dataset, we pretrain a two-layer DBM model, with the first layer containing 1000 hidden units, and the second layer containing M=100 softmax units. The HDP prior over $\mathbf{h}^{(2)}$ was fit to all 1500 character classes.

Fig. 9 displays a random subset of training images, along with the 1^{st} and 2^{nd} layer DBM features, as well as higher-level class-sensitive HDP features. The first-layer features capture low-level features, such as edges and corners, while the HDP features tend to capture higher-level parts, many of which resemble pen "strokes", which is believed to be a promising way to represent characters [18]. The model discovers approximately 50 super-categories, and Fig. 10 shows

10. Gist descriptors have previously been used for this dataset [41]

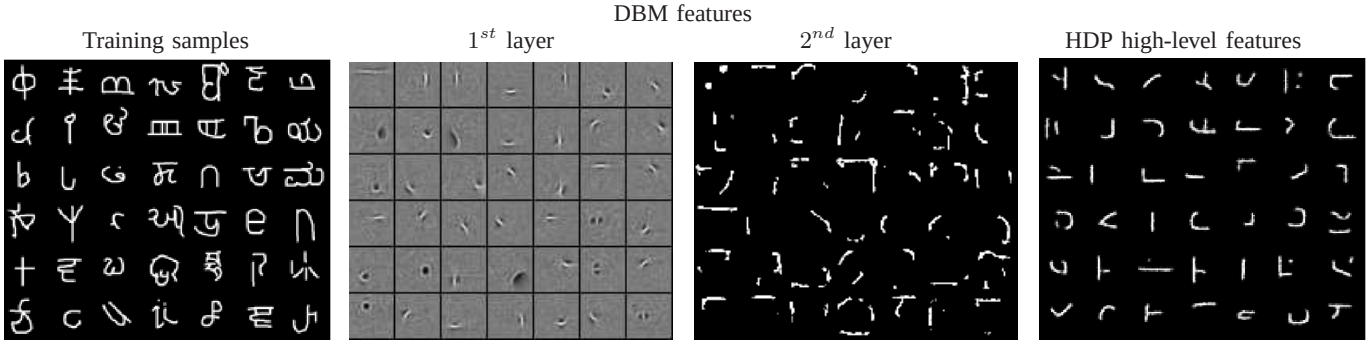

Fig. 9. A random subset of the training images along with the 1^{st} and 2^{nd} layer DBM features, as well as higher-level class-sensitive HDP features/topics. To visualize higher-level features, we first sample M words from a fixed topic ϕ_t, followed by sampling pixel values from the conditional DBM model.

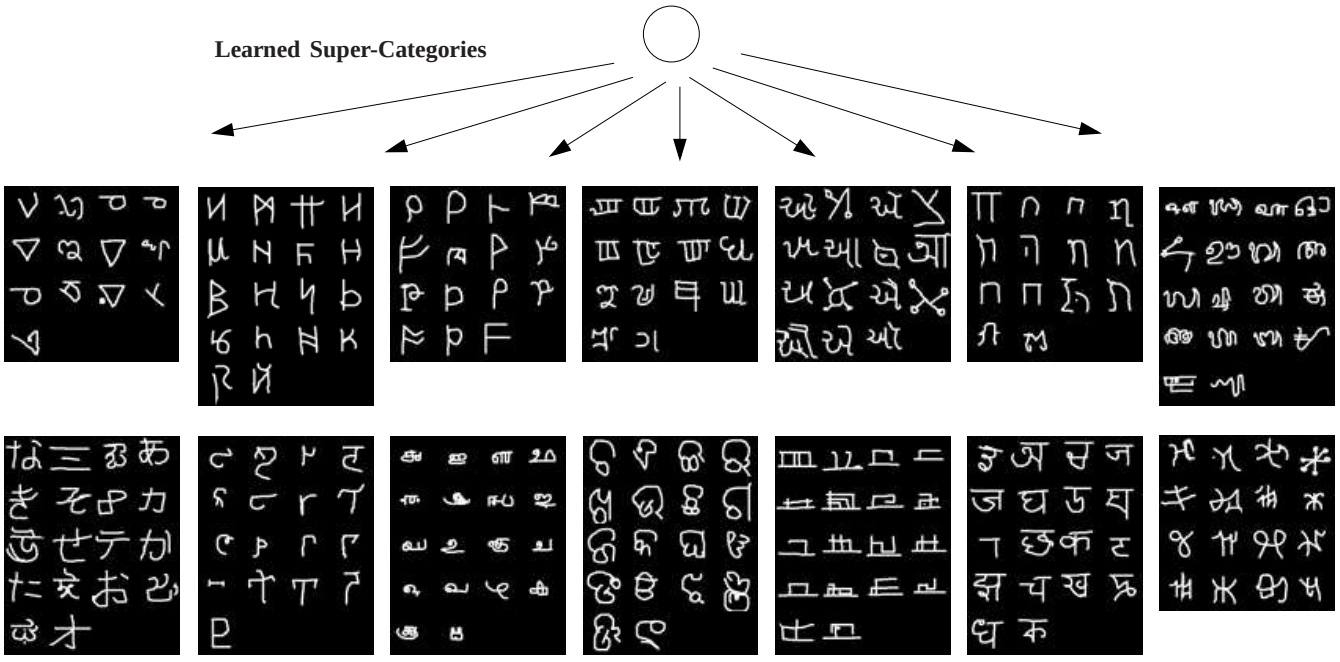

Fig. 10. Some of the learned super-categories that share the same prior distribution over "strokes". Many of the discovered super-categories contain meaningful groupings of characters.

a typical partition of some of the classes into super-categories, which share the same prior distribution over "strokes". Similar to the CIFAR dataset, many of the super-categories contain meaningful groups of characters.

Table 1 further shows results for classifying 15,000 test images as belonging to the novel vs. all other 1,499 character classes. The results are averaged over 200 characters chosen at random, using "leave-one-out" test format. The HDP-DBM model significantly outperforms other methods, particularly when learning characters with few training examples. This result demonstrates that the HDP-DBM model is able to successfully transfer appropriate prior over higher-level "strokes" from previously learned categories.

We next tested the generative aspect the HDP-DBM model. Fig. 11 displays learned super-classes along with examples of *entirely novel* characters that have been generated by the model for the same super-class. In particular, left panels show training characters in one super-category with each row displaying a different observed character and each column displaying a drawing produced by a different subject. Right panels show examples of novel synthesized characters in the corresponding super-category, where each row displays a different synthesized character, whereas each column shows a different example generated at random by the HDP-DBM model. Note that, many samples look realistic, containing coherent, long-range structure, while at the same time being different from existing training images.

Fig. 12 further shows conditional samples when learning only with three training examples of a novel character. Each panel shows three figures: 1) three training examples of a novel character class, 2) 12 synthesized examples of that class, and 3) samples of the training characters in the *same super-category* that the novel character has been grouped under. Many of the novel characters are grouped together with related classes, allowing each character to inherit the prior distribution over similar high-level "strokes", and hence generalizing better to new instances of the corresponding class (see Supplementary Materials for a much richer class of generated samples).

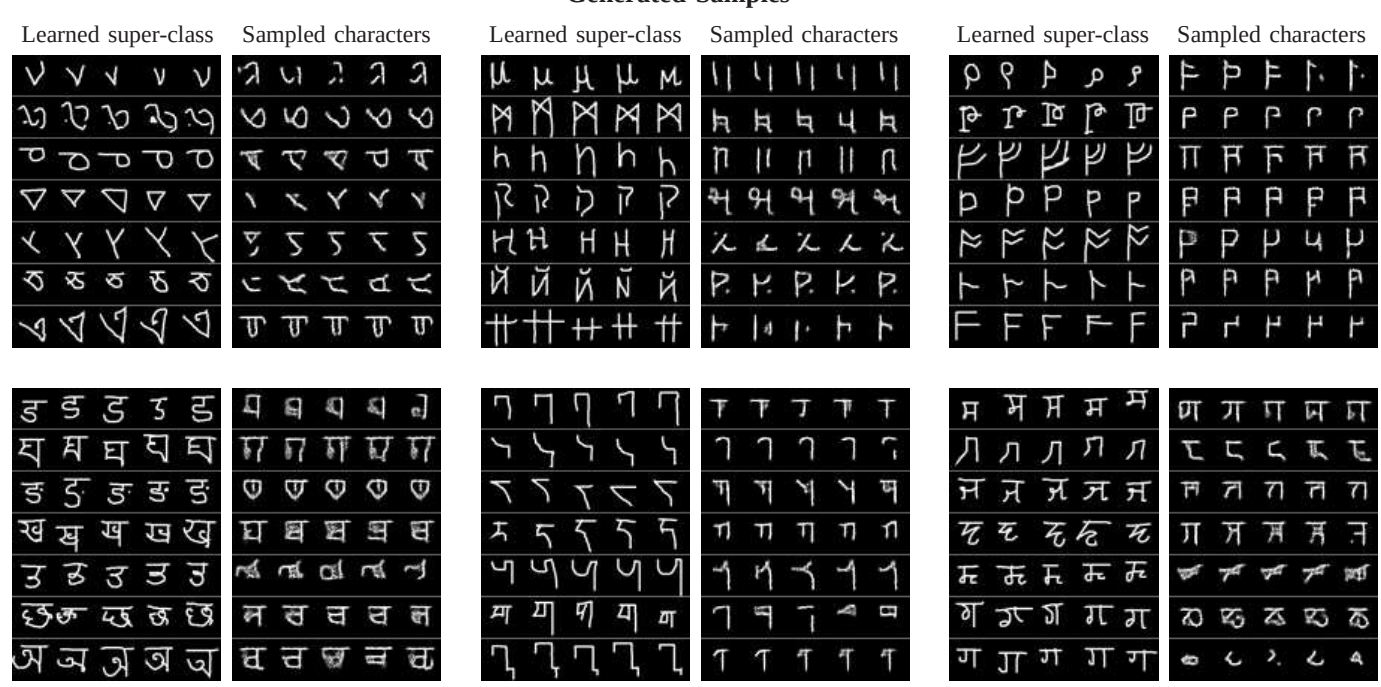

Fig. 11. Within each panel, **Left:** Examples of training characters in one super-category: each row is a different training character and each column is a drawing produced by a different subject. **Right**: Examples of *novel sampled characters* in the corresponding super-category: each row is a different sampled character, and each column is a different example generated at random by the model.

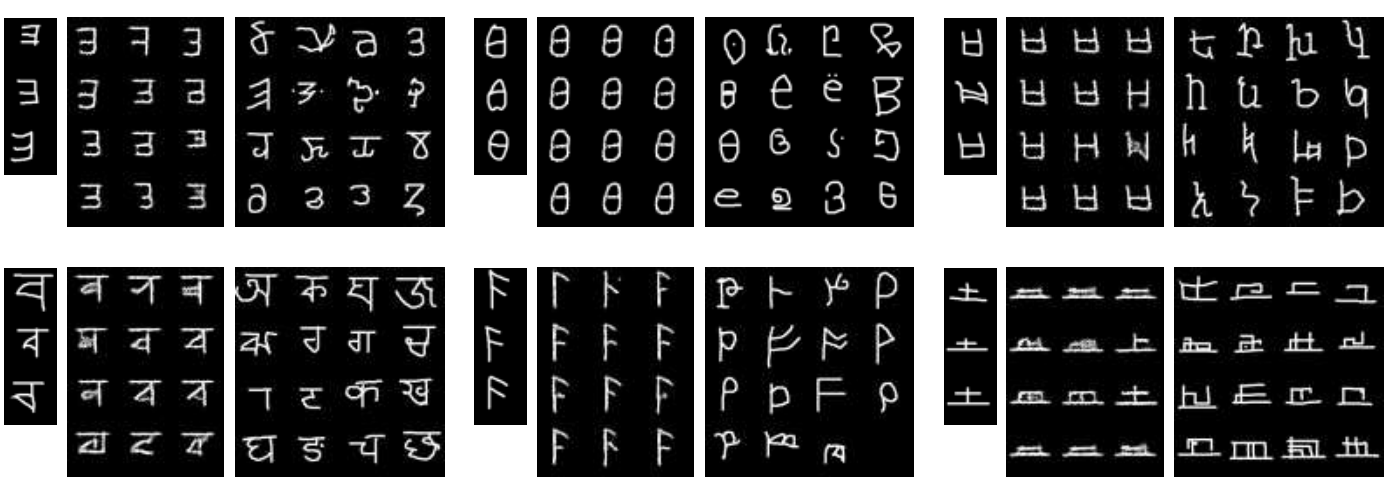

Fig. 12. Each panel shows three figures from left to right: 1) three training examples of a novel character class, 2) 12 synthesized examples of that class, and 3) training characters in the same super-category that the novel character has been assigned to.

Using Deep Belief Networks instead of DBMs produced far inferior generative samples when generating new characters as well as when learning from three examples.

5.3 Motion capture

Results on CIFAR and Character datasets show that the HDP-DBM model can significantly outperform many other models on object and character recognition tasks. Features at all levels of the hierarchy were learned without assuming any image-specific priors, and the proposed model can be applied in a wide variety of application domains. In this section, we show that the HDP-DBM model can be applied to modelling human motion capture data.

The human motion capture dataset consists of sequences of 3D joint angles plus body orientation and translation, as shown in Fig. 13, and was preprocessed to be invariant to isometries [34]. The dataset contains 10 walking styles, including normal, drunk, graceful, gangly, sexy, dinosaur, chicken, old person, cat, and strong. There are 2500 frames of each style at 60fps, where each time step was represented by a vector of 58 real-valued numbers. The dataset was split at random into 1500 training and 1000 test frames of each style. We further preprocessed the data by treating each window of 10

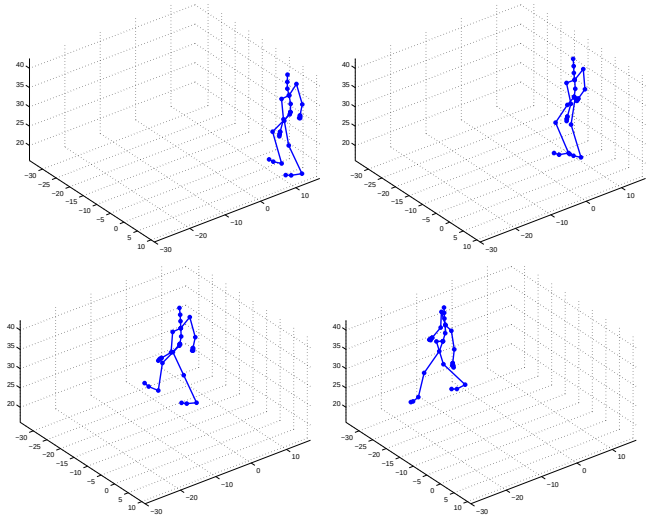

Fig. 13. Human motion capture data that corresponds to the "normal" walking style.

consecutive frames as a single $58 * 10 = 580$-d data vector.

For the two-layer DBM model, the first layer contained 500 hidden units, with the second layer containing M=50 softmax units. The HDP prior over the second-layer features was fit to various walking styles. Using "leave-one-out" test format, Table 1 shows that the HDP-DBM model performs much better compared to other models when discriminating between existing nine walking styles vs. novel walking style. The difference is particularly large in the regime when we observe only a handful number of training examples of a novel walking style.

6 Conclusions

We developed a compositional architecture that learns an HDP prior over the activities of top-level features of the DBM model. The resulting compound HDP-DBM model is able to learn low-level features from raw, high-dimensional sensory input, high-level features, as well as a category hierarchy for parameter sharing. Our experimental results show that the proposed model can acquire new concepts from very few examples in a diverse set of application domains.

The compositional model considered in this paper was directly inspired by the architecture of the DBM and HDP, but it need not be. Indeed, any other deep learning module, including Deep Belief Networks, sparse auto-encoders, or any other hierarchical Bayesian model can be adapted. This perspective opens a space of compositional models that may be more suitable for capturing the human-like ability to learn from few examples.

Acknowledgements

This research was supported by NSERC, ONR (MURI Grant 1015GNA126), ONR N00014-07-1-0937, ARO W911NF-08-1-0242, and Qualcomm.

References

[1] B. Babenko, S. Branson, and S. J. Belongie. Similarity functions for categorization: from monolithic to category specific. In *ICCV*, 2009.

[2] E. Bart, I. Porteous, P. Perona, and M. Welling. Unsupervised learning of visual taxonomies. In *CVPR*, pages 1–8, 2008.

[3] E. Bart and S. Ullman. Cross-generalization: Learning novel classes from a single example by feature replacement. In *CVPR*, pages 672–679, 2005.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[5] David M. Blei, Thomas L. Griffiths, and Michael I. Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *J. ACM*, 57(2), 2010.

[6] Kevin R. Canini and Thomas L. Griffiths. Modeling human transfer learning with the hierarchical dirichlet process. In *NIPS 2009 workshop: Nonparametric Bayes*, 2009.

[7] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[8] Bo Chen, Gungor Polatkan, Guillermo Sapiro, David B. Dunson, and Lawrence Carin. The hierarchical beta process for convolutional factor analysis and deep learning. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 361–368. Omnipress, 2011.

[9] Adam Coates, Blake Carpenter, Carl Case, Sanjeev Satheesh, Bipin Suresh, Tao Wang, and Andrew Y. Ng. Text detection and character recognition in scene images with unsupervised feature learning. In *In Proceedings of the 11th International Conference on Document Analysis and Recognition*, 2011.

[10] Aarron Courville, James Bergstra, and Yoshua Bengio. Unsupervised models of images by spike-and-slab rbms. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 1145–1152, New York, NY, USA, June 2011. ACM.

[11] Li Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(4):594–611, April 2006.

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

[13] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504 – 507, 2006.

[14] G. E. Hinton and T. Sejnowski. Optimal perceptual inference. In *IEEE conference on Computer Vision and Pattern Recognition*, 1983.

[15] C. Kemp, A. Perfors, and J. Tenenbaum. Learning overhypotheses with hierarchical Bayesian models. *Developmental Science*, 10(3):307–321, 2006.

[16] A. Krizhevsky. Learning multiple layers of features from tiny images, 2009.

[17] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Dept. of Computer Science, University of Toronto, 2009.

[18] Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Josh Tenenbaum. One-shot learning of simple visual concepts. In *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, 2011.

[19] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin. Exploring strategies for training deep neural networks. *Journal of Machine Learning Research*, 10:1–40, 2009.

[20] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Intl. Conf. on Machine Learning*, pages 609–616, 2009.

[21] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th International Conference on Machine Learning*, pages 609–616, 2009.

[22] Yuanqing Lin, Tong Zhangi, Shenghuo Zhu, and Kai Yu. Deep coding networks. In *Advances in Neural Information Processing Systems*, volume 23, 2011.

[23] A. Mohamed, G. Dahl, and G. Hinton. Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech, and Language Processing*, 2011.

[24] V. Nair and G. E. Hinton. Implicit mixtures of restricted Boltzmann machines. In *Advances in Neural Information Processing Systems*, volume 21, 2009.

[25] A. Perfors and J.B. Tenenbaum. Learning to learn categories. In *31st Annual Conference of the Cognitive Science Society*, pages 136–141, 2009.

[26] M. A. Ranzato, Y. Boureau, and Y. LeCun. Sparse feature learning for deep belief networks. *Advances in Neural Information Processing Systems*, 2008.

[27] H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Stat.*, 22:400–407, 1951.

[28] A. Rodriguez, D. Dunson, and A. Gelfand. The nested Dirichlet process. *Journal of the American Statistical Association*, 103:11311144, 2008.

[29] R. R. Salakhutdinov and G. E. Hinton. Deep Boltzmann machines. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 12, 2009.

[30] R. R. Salakhutdinov and G. E. Hinton. Replicated softmax: an undirected topic model. In *Advances in Neural Information Processing Systems*, volume 22, 2010.

[31] L.B. Smith, S.S. Jones, B. Landau, L. Gershkoff-Stowe, and L. Samuelson. Object name learning provides on-the-job training for attention. *Psychological Science*, pages 13–19, 2002.

[32] Richard Socher, Cliff Lin, Andrew Y. Ng, and Christopher Manning. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the Twenty-Eighth International Conference on Machine Learning*. ACM, 2011.

[33] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky. Describing visual scenes using transformed objects and parts. *International Journal of Computer Vision*, 77(1-3):291–330, 2008.

[34] G. Taylor, G. E. Hinton, and S. T. Roweis. Modeling human motion using binary latent variables. In *Advances in Neural Information Processing Systems*. MIT Press, 2006.

[35] Graham W. Taylor, Rob Fergus, Yann LeCun, and Christoph Bregler. Convolutional learning of spatio-temporal features. In *ECCV 2010*. Springer, 2010.

[36] Y. W. Teh and G. E. Hinton. Rate-coded restricted Boltzmann machines for face recognition. In *Advances in Neural Information Processing Systems*, volume 13, 2001.

[37] Y. W. Teh and M. I. Jordan. Hierarchical Bayesian nonparametric models with applications. In *Bayesian Nonparametrics: Principles and Practice*. Cambridge University Press, 2010.

[38] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[39] T. Tieleman. Training restricted Boltzmann machines using approximations to the likelihood gradient. In *ICML*. ACM, 2008.

[40] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

[41] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large image databases for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[42] Antonio B. Torralba, Kevin P. Murphy, and William T. Freeman. Shared features for multiclass object detection. In *Toward Category-Level Object Recognition*, volume 4170 of *Lecture Notes in Computer Science*, pages 345–361. Springer, 2006.

[43] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol. Extracting and composing robust features with denoising autoencoders. In William W. Cohen, Andrew McCallum, and Sam T. Roweis, editors, *Proceedings of the Twenty-Fifth International Conference*, volume 307, pages 1096–1103, 2008.

[44] Fei Xu and Joshua B. Tenenbaum. Word learning as bayesian inference. *Psychological Review*, 114(2), 2007.

[45] L. Younes. Parameter inference for imperfectly observed Gibbsian fields. *Probability Theory Rel. Fields*, 82:625–645, 1989.

[46] L. Younes. On the convergence of Markovian stochastic algorithms with rapidly decreasing ergodicity rates, March 17 2000.

[47] A. L. Yuille. The convergence of contrastive divergences. In *Advances in Neural Information Processing Systems*, 2004.

Ruslan Salakhutdinov received his PhD in machine learning (computer science) from the University of Toronto in 2009. After spending two post-doctoral years at the Massachusetts Institute of Technology Artificial Intelligence Lab, he joined the University of Toronto as an Assistant Professor in the Departments of Statistics and Computer Science. His primary interests lie in statistical machine learning, Bayesian statistics, probabilistic graphical models, and large-scale optimization. He is the recipient of the Early Researcher Award, Connaught New Researcher Award, and is a Scholar of the Canadian Institute for Advanced Research.

Joshua B. Tenenbaum received his Ph.D. in 1993 from MIT in the Department of Brain and Cognitive Sciences, where he is currently Professor of Computational Cognitive Science as well as a principal investigator in the Computer Science and Artificial Intelligence Laboratory (CSAIL). He studies learning, reasoning and perception in humans and machines, with the twin goals of understanding human intelligence in computational terms and bringing computers closer to human capacities. He and his collaborators have pioneered accounts of human cognition based on sophisticated probabilistic models and developed several novel machine learning algorithms inspired by human learning, most notably Isomap, an approach to unsupervised learning of nonlinear manifolds in high-dimensional data. His current work focuses on understanding how people come to be able to learn new concepts from very sparse data – how we 'learn to learn' – and on characterizing the nature and origins of people's intuitive theories about the physical and social worlds. His papers have received awards at the IEEE Computer Vision and Pattern Recognition (CVPR), NIPS, Cognitive Science, UAI and IJCAI conferences. He is the recipient of early career awards from the Society for Mathematical Psychology, the Society of Experimental Psychologists, and the American Psychological Association, along with the Troland Research Award from the National Academy of Sciences.

Antonio Torralba received the degree in telecommunications engineering from the Universidad Politecnica de Cataluna, Spain; he received the PhD degree in signal, image, and speech processing from the Institute National Polytechnique de Grenoble, France. Thereafter, he spent postdoctoral training at the Brain and Cognitive Science Department and the Computer Science and Artificial Intelligence Laboratory at the Massachusetts Institute of Technology (MIT). He is an associate professor of electrical engineering and computer science in the Computer Science and Artificial Intelligence Laboratory (CSAIL) at MIT. He is a member of the IEEE.