A Hybrid Neural Network-Latent Topic Model

Li Wan **Leo Zhu** **Rob Fergus**
Dept. of Computer Science, Courant institute, New York University
{wanli,zhu,fergus}@cs.nyu.edu

Abstract

This paper introduces a hybrid model that combines a neural network with a latent topic model. The neural network provides a low-dimensional embedding for the input data, whose subsequent distribution is captured by the topic model. The neural network thus acts as a trainable feature extractor while the topic model captures the group structure of the data. Following an initial pre-training phase to separately initialize each part of the model, a unified training scheme is introduced that allows for discriminative training of the entire model. The approach is evaluated on visual data in scene classification task, where the hybrid model is shown to outperform models based solely on neural networks or topic models, as well as other baseline methods.

1 Introduction

Probabilistic graphical models [4, 9, 1] and neural networks [10, 17, 14] are the two prevalent types of belief network in machine learning. In the first of these, explicit variables (usually corresponding to some tangible entity) are linked in a sparse dependency structure, which is typically specified by hand using domain knowledge. This is quite different to a neural network where variables lack explicit meaning and are densely connected. In principle, the model is less dependent on the details of the problem, but in practice selections must be made regarding the model architecture and the training protocol (e.g. [13]).

Each model is appropriate for different settings. Probabilistic graphical models are a natural way to represent the high level structure of a signal. Equally, deep neural networks have proven effective at automatically learning good feature representations from the raw signal, a situation where detailing the precise form of the dependencies is problematic.

This paper proposes a model that combines a deep neural network with a latent topic models and presents a joint learning scheme that allows the combined model to be trained in a discriminative fashion. The resulting model combines the strengths of the two approaches: the deep belief network provides a powerful non-linear feature transformation for the domain-appropriate topic model.

Our main technical contribution is a novel way of transforming the graphical model during inference to form additional neural network layers. This transformation allows the back-propagation [13] to be performed in a straightforward manner on the unified model. We demonstrate this transformation for a latent topic model but the operation is valid for any model where (approximate) inference is possible in closed-form.

We demonstrate our model in a computer vision setting, using it to perform scene classification. We choose this domain because (a) developing good feature representations for visual data is an active area of research [14, 17] and the neural network part of our model addresses this task and (b) latent topic models, such as latent Dirichlet allocation (LDA) [4], have been shown to be effective for image classification, using a bag-of-words image representations [5, 6].

The training procedure for our model involves a preliminary unsupervised phase where the parameters of the deep-belief network are initialized. This is performed using restricted Boltzmann machines (RBMs) in conjunction with contrastive divergence learning, in the style of Hinton and Salakhutdinov [8]. The same layer-wise greedy scheme of [8, 2] is used.

Our graphical model is a variant of latent Dirichlet allocation [4] and is closely related to the author-topic model [18], introduced to the vision community

Appearing in Proceedings of the 15^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume 22 of JMLR: W&CP 22. Copyright 2012 by the authors.

by Sudderth *et al.* [21]. However, unlike these unsupervised models, ours is discriminative and incorporates the class label, similar to the Disc-LDA model of Lacoste-Julien *et al.* [11] and the Supervised-LDA model of Blei and McAuliffe [3]. Ngiam *et al.* [16] show how a feedforward neural-network can be used as a deterministic transformation as input to a deep-belief network (DBN), showing results on MNIST and NORB datasets. In contrast, we use a topic model instead of the DBN and evaluate on a more complex scene dataset. Another related paper to ours is Salakhutdinov *et al.* [19], who combine a Gaussian process with a deep-belief network. However, this is a simpler graphical model than ours and lacks the modeling capabilities of latent topic models. The most closely related paper is the contemporaneous work of Salakhutdinov [20] who combine an Hierarchical Dirichlet Process with a Deep Boltzmann Machine.

2 The hybrid model

Our hybrid model is a combination of one particular probabilistic model, hierarchical topic model (HTM), and a neural network (NN). We consider a set of N images $\{I_1, \ldots, I_i, \ldots, I_N\}$ with labels y. From each we extract a set of SIFT descriptors [15] $\{v_1, \ldots, v_j, \ldots, v_{n_i}\}$. Each descriptor v_j is individually mapped by a neural network with parameters w to a feature vector x_j in R^d, d being the number of units in the top layer of the network. The transformation of $v \rightarrow x$ is denoted by $f_w(v)$. The structure of neural network is encoded by layers of hidden units h. The connections of hidden units are given in section 2.1.

Each image is represented by a topic model where each topic is a probability distribution over visual words in a vocabulary. In prevalent topic models such as LDA[4, 6] and its variants [21], the vocabulary is given by vector quantization, but not integrated into the topic model. In this paper, we want to learn the feature representation and the topic model jointly. We extend the topic model by adding an extra latent variable to encode visual vocabulary. Unlike other topic models, our model is directly defined over image features, and capable of learning vocabulary and topic distribution simultaneously. The new topic model is of hierarchical structure (see fig.1) whose distribution is given in section 2.2.

The hierarchical topic model is combined with the neural network to form a hybrid model by treating the output x of the network as the bottom nodes of the hierarchy.

2.1 Neural Network

In this paper, we consider a neural network with two hidden layers, as shown in Fig. 1(a). The first hidden one is a sigmoid layer which maps the input features v into a binary representation h via a sigmoid function, i.e. $h = \sigma(w_1 v + b_1)$ where $\sigma(t) = 1/(1 + exp(-t))$ and w_1, b_1 are the parameters of this layer. The second hidden layer performs linear dimension reduction $x = hw_2 + b_2$, with w_2, b_2 being parameters. The output of the units x correspond to the transformation $f_w(v)$ provided the whole network. An arbitrary number of extra hidden layers could be inserted between these two layers if more a complex transformation is preferred. Let $w = \{w_1, b_1, w_2, b_2\}$ denote all parameters of the network. Training the network is performed by back propagation [13] on w. The initialization of w is obtained by learning a Restricted Boltzmann Machine (RBM) [7] with the same structure of network. We will give details of the training procedure in section 3.2.

2.2 Hierarchical topic model

Given an image I_i, the neural network will transform each raw feature v_j into a vector x_j, which is input to the hierarchical topic model. We assume that each x_j is generated by a Gaussian distribution of the corresponding word cluster u_j. The Gaussian is parametrized by $\phi_{u_j} = \{\mu_{u_j}, \Sigma_{u_j}\}$. The word u_j is generated by a multinomial word distribution with parameters of η_{z_j}. The distribution of topics z_j is a multinomial parametrized by π_{y_i} where y_i is a label of scene category for image I_i. y_i is provided in the learning stage. The overall model is shown in Fig. 1(b). The hierarchical generative process is given by:

1. Draw latent topic $z_j \sim Multi(\pi_{y_i})$
2. Draw latent word $u_j \sim Multi(\eta_{z_i})$
3. Draw feature vector $x_j \sim Gaussian(\phi_{u_j})$.

The prior distributions on π, η, ϕ are defined as follows:

$$\begin{aligned}\pi_y &\sim Dir(\alpha) \text{ for } y \in \{1, 2, \ldots S\}\\ \eta_z &\sim Dir(\beta) \text{ for } z \in \{1, 2, \ldots M\}\\ \phi_u &\sim \mathcal{NIW}(\gamma) \text{ for } u \in \{1, 2, \ldots K\}\end{aligned}$$

where $Dir(.)$ is Dirichlet distribution, $\mathcal{NIW}(\gamma)$ is normal-inverse-Wishart distribution, parametrized by $\gamma = \{\mu_0, \kappa, \nu, \Lambda_0\}$. S is the number of scene categories (class labels), M is the number of latent topics, and K is the size of word vocabulary.

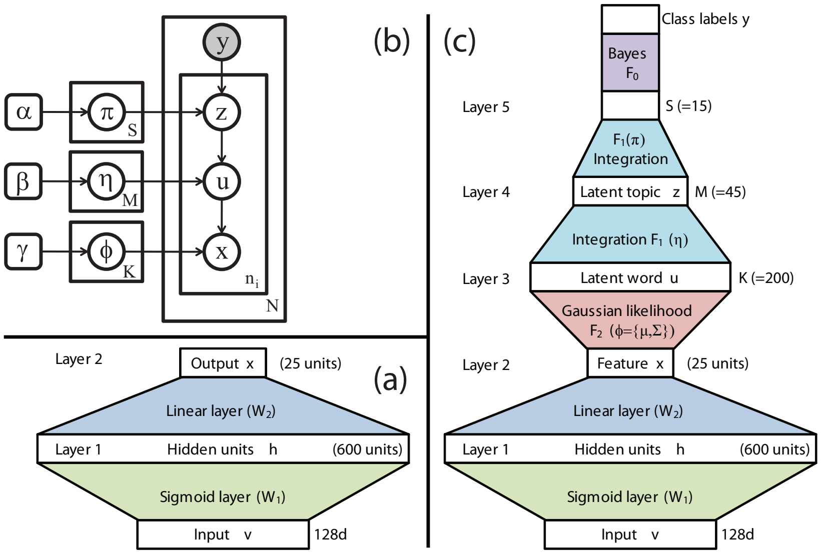

Figure 1: (a) Neural network (NN). (b) Hierarchical topic model (HTM). (c) Our hybrid model that combines the NN and HTM.

The hyper-parameters α, β, γ are set by hand, their exact value not being too important. Learning the model involves the estimation of the parameters π, η, ϕ. In section 3.3, we will discuss the learning of the hierarchical model.

Note that, unlike standard LDA[4], our model has no image (or document) specific prior of topic distribution. The most related representation is the scene model [21] which essentially is a author-topic model [18].

2.3 The hybrid model: coupling the neural network and topic model

The hybrid model is designed to combine the strengths of the two models defined above. The input x to the hierarchical topic model is the transformed output $f_w(v)$ of the neural network. Note, in the hybrid model, x is not observable. The topic model captures high-level scene structure of an image while the neural network offers approximate low-dimensional embedding of raw features. Typically, the probabilistic topic model is a sparse graph while neural network is a densely connected graph.

3 Learning the hybrid model

3.1 Brief description

The task of learning the hybrid model is to maximize the posterior distribution of class label y, given the input data v. The loss function is given by:

$$L(w, \pi, \eta, \phi) = -\log p(y|v, w, \pi, \eta, \phi) \qquad (1)$$

Since $x = f_w(v)$, $p(y|x, w, \pi, \eta, \phi) = p(y|v, w, \pi, \eta, \phi)$, and applying Bayes rule, we thus have:

$$L = -\log p(f_w(v)|y, \pi, \eta, \phi) + \log \sum_{\tilde{y}=1}^{S} p(f_w(v)|\tilde{y}, \pi, \eta, \phi) \qquad (2)$$

The likelihood function of the hierarchical topic model for an image I_i is defined as $p(f_w(v)|y)$:

$$\prod_{j=1}^{n_i} \sum_{z_j=1}^{M} \left(\sum_{u_j=1}^{K} p\left(f_w(v_j)|u_j, \phi\right) p\left(u_j|z_j, \eta\right) \right) p\left(z_j|y, \pi\right)$$

The optimization of the loss function is performed by gradient descent. The learning procedure consists of two steps:

1. We first initialize the parameters $\{w^0, \pi^0, \eta^0, \phi^0\}$ obtained by pre-training the neural network and the hierarchical topic model which will be introduced in sections 3.2 and 3.3.

2. The parameters are then updated according to the following rules (where c is a learning rate):

$$\begin{aligned} w^{t+1} &= w^t - c\frac{\partial L}{\partial w}, \ \pi^{t+1} = \pi^t - c\frac{\partial L}{\partial \pi} \\ \eta^{t+1} &= \eta^t - c\frac{\partial L}{\partial \eta}, \ \phi^{t+1} = \phi^t - c\frac{\partial L}{\partial \phi} \end{aligned} \qquad (3)$$

The major technical issues of learning are (i) how to obtain a good initialization; (ii) how to calculate the gradients of the loss function defined over the joint model. We will address these two issues in the following sections.

3.2 Pre-training of neural network

We first pre-train the neural network by learning RBMs with the same structure in an unsupervised manner. A RBM is an undirected graphical model with connections between visible units v and hidden units h. The joint distribution $p(v,h) \propto \exp(-E(v,h))$ with the normalization constant $Z = \sum_{v,h} \exp(-E(v,h))$. The transformation provided by neural network is defined as $p(h|v)$ in the corresponding RBM. We apply the contrastive divergence (CD) algorithm [7] to train the network. CD algorithm is a greedy layer-wise learning method for stack of Restricted Boltzmann machine (RBMs) which maximize the variational lower bound of the model likelihood. Hinton and Osindero [7] show that such initialization works well for discriminative training.

Each layer of NN is initialized by the corresponding RBM model. More precisely, the RBM energy for pre-training of the bottom layer is $E(v,h) = -\frac{1}{2}(v-b)^T(v-b) - h^T c - v^T W h$. The energy of RBM for pre-training of the second (top) layer is $E(h,x) = -\frac{1}{2}(x-c)^T(x-c) - h^T b - h^T W x$.

3.3 Pre-training of the hierarchical topic Model

Once we have pre-trained the neural network, the resulting $x = f_w(v)$ are fixed and used as input to the hierarchical topic model. The training of the graphical model is performed by Gibbs sampling in the style adopted by Sudderth *et al.* [21] in their similar scene model. The sampling procedure is given in Algorithm 1:

```
foreach image i do
    foreach feature j do
        • Remove $u_{ij}$ from the cached statistics
        for current latent variable $k = z_{ij}$ and
        $m = u_{ij}$: $\pi_{yk} \leftarrow \pi_{yk} - 1, \eta_{km} \leftarrow \eta_{km} - 1$
        and $\phi_m \leftarrow$ remove $u_{ij}$ from $\phi_m$.
        • Sample $z_{ij}$ and $u_{ij}$ jointly from the
        following multinomial distribution:
        $p(z_{ij}, u_{ij} | x_{ij}, \pi, \eta, \phi, \alpha, \beta, \gamma) \propto$
        $p(x_{ij} | u_{ij}, \phi, \gamma) p(u_{ij} | \eta_{z_{ij}}, \beta) p(z_{ij} | \pi_{y_i}, \alpha)$.
        • Add back feature $x_{ij}$ to the cached
        statistic for its new latent variable
        $k = z_{ij}$ and $m = u_{ij}$:
        $\pi_{yk} \leftarrow \pi_{yk} + 1, \eta_{km} \leftarrow \eta_{km} + 1$ and $\phi_m \leftarrow$
        add $u_{ij}$ to $\phi_m$.
    end
end
```

Algorithm 1: Gibbs sampler for the hierarchical topic model.

3.4 Joint optimization by gradient descent

We now have the full pre-trained model with all parameters initialized. Gradient descent is applied to update the parameters. The loss function defined in Equation (2) can be expanded as follows:

$$\begin{aligned} L &= -\sum_j \log p(f_w(v_j)|y,\pi,\eta,\phi) + \\ & \quad \log \sum_{i=1}^{S} \prod_j p(f_w(v_j)|y=i,\pi,\eta,\phi) \end{aligned}$$

Let $F_0(A,y) \equiv L$ where A is a $n_i \times S$ matrix, $A_{ji} = [p(f_w(v_j)|y=i,\pi,\eta,\phi]_{ji}$ and F_0 is a function which takes an input of a matrix A and a label information y. $F_0(A,y)$ is a form of:

$$F_0(A,y) = -\sum_j \log A_{jy} + \log \sum_i \prod_j A_{ji} \tag{4}$$

We can decompose each element of A as follows:

$$\begin{aligned} A_{ji} &= p(f_w(v_j)|y=i,\pi,\eta,\phi) \\ &= \sum_{z_j=1}^{M} p(f_w(v_j)|z_j,\eta,\phi)p(z_j|y=i,\pi) \\ &= [F_1(B,C)]_{ji} \\ & \quad \text{where } B_{jm} = p(f_w(v_j)|z=m,\eta,\phi) \\ & \quad \text{and } C_{mi} = p(z=m|y=i,\pi) \end{aligned}$$

Here matrix B is of size $n_i \times M$ and matrix C is of size $M \times S$. Function F_1 is a multiplication of two matrices, i.e. $A = F_1(B,C) = BC$.

The decomposition of matrix B is given by:

$$\begin{aligned} B_{jm} &= p(f_w(v_j)|z=m,\eta,\phi) \\ &= \sum_{u_j=1}^{K} p(f_w(v_j)|u_j,\phi)p(u_j|z=m,\eta) \\ &= [F_1(D,E)]_{jm} \\ & \quad \text{where } D_{jk} = p(f_w(v_j)|u=k) \\ & \quad \text{and } E_{km} = p(u_j=k|z=m,\eta) \end{aligned}$$

where E is the η matrix of size $K \times M$ and D is of size $n_i \times K$. We can decompose the matrix D:

$$D_{jk} = p(f_w(v_j)|u=k,\phi)$$

$$
\begin{aligned}
&= |\Sigma_k|^{-1}\exp\left(-\frac{1}{2}\left(f_w(v_j)-\mu_k\right)\Sigma_K^{-1}\left(f_w(v_j)-\mu_k\right)^T\right)\\
&= [F_2(x,\phi)]_{jk} \text{ where } \phi_k=\{\mu_k,\Sigma_k\}
\end{aligned}
$$

Function F_2 evaluates the likelihood (dropping out the normalization constant) of the input data v_i for different Gaussian centers $\phi_k = \{\mu_k, \Sigma_k\}$ for $k = 1, 2, \ldots, K$. The output of $F_2(f_w(v), \phi)$ is $|\Sigma_k|^{-1}\exp(-\frac{1}{2}(f_w(v_j)-\mu_k)\Sigma_k^{-1}(f_w(v_j)-\mu_k)^T)$.

After defining the functions F_0, F_1, F_2, the loss function can be rewritten:

$$
L = F_0(\underbrace{F_1(\underbrace{F_1(\overbrace{F_2\left(f_w(v),\phi\right)}^{D},\overbrace{\eta}^{E})}_{B},\overbrace{\pi}^{C})}_{A},y) \qquad (5)
$$

and

$$
\begin{aligned}
\frac{\partial L}{\partial \pi} &= \frac{\partial F_0(A,y)}{\partial A}\frac{\partial F_1(B,\pi)}{\partial \pi}\\
\frac{\partial L}{\partial \eta} &= \frac{\partial F_0(A,y)}{\partial A}\frac{\partial F_1(B,\pi)}{\partial B}\frac{\partial F_1(D,\eta)}{\partial \eta}\\
\frac{\partial L}{\partial \phi_k} &= \frac{\partial F_0(A,y)}{\partial A}\frac{\partial F_1(B,\pi)}{\partial B}\\
&\quad \frac{\partial F_1(D,\eta)}{\partial D}\frac{\partial F_2(f_w(v),\phi)}{\partial \phi_k}\\
\frac{\partial L}{\partial w} &= \frac{\partial F_0(A,y)}{\partial A}\frac{\partial F_1(B,\pi)}{\partial B}\frac{\partial F_1(D,\eta)}{\partial D}\\
&\quad \sum_{j=1}^{n_i}\frac{\partial F_2(f_w(v_j),\phi)}{\partial f_w(v_j)}\frac{\partial f_w(v_j)}{\partial w}
\end{aligned}
$$

The gradients of L w.r.t w, ϕ, η, π can be obtained by applying chain rule, provided the gradients of F_0, F_1, F_2:

1. Function $F_0(A, y)$ evaluation:

$$
F_0(A,y) = -\sum_j \log A_{jy} + \log\sum_{i=1}^{S}\prod_{j=1}^{n_i} A_{ji}
$$

and gradient computation:

$$
\frac{\partial F_0(A,y)}{A_{ji}} = -1 + \frac{\exp(F_0(A,y))}{A_{ji}}
$$

2. Function $F_1(B, \pi)$ evaluation:

$$
[F_1(B,\pi)]_{ik} = \frac{\sum_j B_{ij}\pi_{jk}}{\sum_k \pi_{jk}}
$$

and gradient computation:

$$
\frac{\partial [F_1(B,\pi)]_{ik}}{\partial B_{ij}} = \frac{\pi_{jk}}{\sum_t \pi_{jt}}
$$

and

$$
\frac{\partial [F_1(B,\pi)]_{ik}}{\partial \pi_{jk}} = \frac{B_{ij}}{\sum_t \pi_{jt}} - \frac{[F_1(B,\pi)]_{ik}}{\sum_t \pi_{jt}}
$$

The reason that we divide the output by $\sum_k R_{jk}$ is because the second parameter π of this function is always sufficient static of multinomial distribution, thus must always sum to one. The derivative for $F_1(D, \eta)$ takes a similar form.

3. Function $F_2(x, \phi)$,where $\phi = \{\phi_1, \phi_2, \ldots, \phi_K\}$ each of ϕ_k is a Gaussian center $\phi_k = \{\mu_k, \Sigma_k\}$ $[F_2(x,\phi)]_{jk} = |\Sigma_k|^{-1}\exp(-\frac{1}{2}(\mathbf{x}_j-\mu_k)\Sigma_k^{-1}(\mathbf{x}_j-\mu_k)^T)$ Taking gradients we obtain (omitting the Σ_k update for brevity):

$$
\begin{aligned}
\frac{\partial [F_2(x,\phi)]_{jk}}{\partial x_j} &= -[F_2(x,\phi)]_{jk}(x_j-\mu_k)\Sigma_k^{-1}\\
\frac{\partial [F_2(x,\phi)]_{jk}}{\partial \mu_k} &= [F_2(x,\phi)]_{jk}(x_j-\mu_k)\Sigma_k^{-1}
\end{aligned}
$$

3.5 Unifying probabilistic hierarchical model and neural network: back-propagation

The gradient descent scheme introduced in section 3.4 can be interpreted as a back-propagation algorithm on a new neural network which unifies the hierarchical topical model and the neural network. The decomposition of the loss function in Equation (5) allows us applying chain rule to calculate the gradients w.r.t. all the parameters in the hybrid model. It suggests a strong connection with the back-propagation algorithm if we define four consecutive layers from top to bottom in the following order:

1. *Bayesian Layer:* Function F_0
2. *Integration Layer on z:* Function F_1 with π as the second parameter
3. *Integration Layer on u:* Function F_1 with η as the second parameter
4. *Gaussian Likelihood Layer:* Function F_2

The transformation of each layer is defined by the corresponding function F and its parameters. We can simply add these four invented layers on the top of the original neural network, yielding a unified neural network. By back-propagating through this unified network, we can estimate the parameters of the hybrid model.

3.6 Inference of the hybrid model: feed-forward

Inference is performed as a feed-forward procedure on the unified neural network. Given a testing image v, the first two layers of the neural network produce encoded features $x = f_w(v)$. According to the definition of the four additional layers, the output at the top layer is $p(y|f_w(v))$ for $y = 1, 2, \ldots, S$. The task of inference $y^* = \arg\max_y p(y|f_w(v))$ is the same as passing x through Gaussian likelihood layer(F_2), Integration layer($F_1(\cdot, \eta)$), Integration layer($F_1(\cdot, \pi)$) and Bayesian layer F_0 (see Figure 1(c)). Note that our hybrid approach can be extended to any graphical model where the (approximate) inference can be performed in closed-form.

4 Toy experiments

We illustrate the different stages of training in our model with toy 2D data, as shown in Figure 2. The data is drawn from 4 classes, arranged in 5 crescent-shaped clusters (which cannot easily be separated by a Gaussian mixture model). Pre-training the NN (having a 2-50-2 architecture) for the most part preserves the structure of the input data (see Figure 2(middle)) in the feature space, thus the topic model, using Gaussian distributions makes many classification errors. But following back-propagation of the entire model (Figure 2(right)), the NN provides a significant warping of the input space, thus making it easier to separate the clusters. Note that the Gaussian clusters of the topic model do not lie directly on top of the features since the topic model optimizes a discriminative criterion, rather than a generative one.

5 Vision experiments

In this section, we provide a quantitative evaluation of our hybrid model on a vision dataset and compare its performance with alternative methods such as a standard neural network, hierarchical topic model, pLSA, LDA and their variants.

5.1 Dataset and image features

We evaluated our hybrid model on challenging image scene recognition dataset [12] where the experimental results of standard probabilistic graphical models such as pLSA and LDA have been reported. This dataset consists of 1500 training images and 2998 test images each of which is labeled by one of 15 scene categories such as street, kitchen, coast, etc. Each image is represented by a set of SIFT descriptors which are sampled every 16 pixels (giving ~ 240 per image). Each SIFT descriptor is a 128-dimensional vector which encodes the histogram of gradients of a local image patch with size of 32×32. Since we focus on the theoretical issues involved when combining NN with graphical models, the pyramid representation [12] which leads to better performance is not used.

5.2 Scene modeling: topic model, neural network and the hybrid

Topic model. The baseline model of image scene is standard LDA [4, 6]. We first form visual vocabulary with size of 200 (following [6]) where visual words are obtained by vector quantization of SIFT descriptors. An image scene is represented by LDA which learns the latent topic distributions of visual words. When labels are provided in training, the variants of LDA such as supervised LDA [3] and discriminative LDA [11] can be applied. The performance of LDA is directly based on [6].

Hierarchical topic model. Unlike LDA, the hierarchical topic model introduced in section 2.2 integrates the representation of dictionary. The observation data input to HTM is SIFT features. The HTM is capable of learning the visual vocabulary and the topic distribution jointly. Our HTM method which makes use of class labels is related to the discriminative version of LDA [11]. We also follow the standard way [12, 11] to study the discriminative power of the inferred latent topics by training a SVM with the assigned topics as classification features. Note that the input to all variants of both HTM and LDA models are fixed visual vocabulary without the ability of learning transformation of low-level feature representation, i.e. SIFT descriptors.

Neural network. We use the same architecture of the neural network as described in section 2.1. In order to predict scene labels, we extend the neural network by imposing a softmax layer on the top which performs logistic regression of the scene labels and the output of the feature transformation. Learning the extended neural network is performed by standard back-propagation [13]. Unlike the topic models, this approach lacks a high level model of the scene.

Hybrid model: The hybrid model is a combination of the hierarchical topic model and the neural network which integrate the ability of learning low-level feature transformation and high-level scene representation. The input to the model is SIFT features, used by the other approaches. The free parameters and hyper-parameters are set to $S = 15$ (categories), $M = 45$ (topics), $K = 200$ (words), $\alpha = 1/3, \beta = 1/3$. Let d denote the dimension of μ (=25). γ is set to $(\mu_0 = \mathbf{0}_d, \kappa = 0.1, \nu = d + 5, \Lambda_0 = \mathbf{I}_d)$. The NN has a $128 - 600 - 25$ architecture. Back-propagation is performed using conjugate gradients, with mini-batches of 75 images by ~ 200 features/image. Convergence occurs in about 70 iterations. The corresponding separate HTM and NN use the same parameter settings.

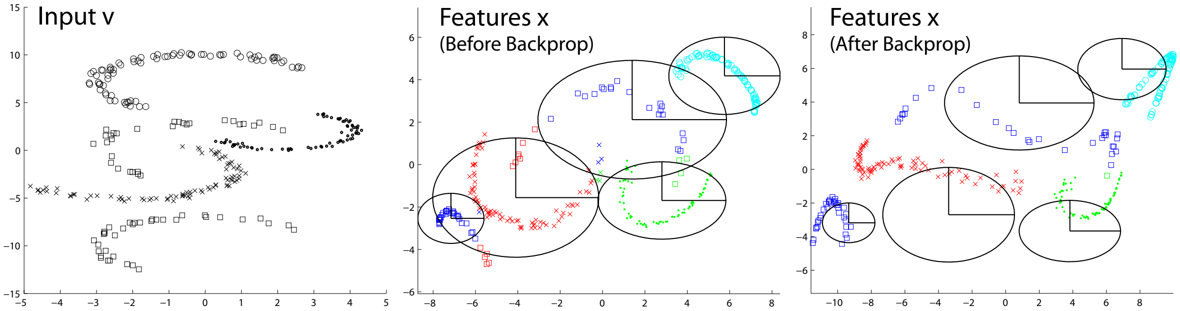

Figure 2: Toy experiment using 2D data (left), with 5 clusters drawn from 4 classes (cross, dot, square, circle). Note that there are multiple clusters per class. Middle: Visualization of 2D feature space x after NN pre-training and Gibbs sampling of supervised topic model with K=5 clusters and Y=4 classes. The ellipses show the mean and covariance of each Gaussian cluster in the HTM. In the HTM, the color indicates the predicted class of each data point (the labels: red=cross, green=dot, blue=square, cyan=circle). Note that several points are mislabeled. Right: The feature space after back-propagation of unified model. The NN has distorted the feature space to make classification easier for the topic model, with only a single data point now being misclassified.

pLSA+SVM [12]	LDA [6]	Supervised LDA [3]	Neural Network	HTM
63.3	65.2	67.0	51.6 ± 1.1	64.9 ± 1.2
HTM+ SVM	Hybrid model no pre-train	Hybrid model pre-train NN only	Hybrid model pre-trained	Hybrid model fully trained
65.5 ± 1.5	47.2	52.5	65.7 ± 0.4	70.1 ± 0.6

Table 1: Classification rates of our model and other approaches on a scene classification dataset [12]. Our implementation of discriminative hierarchical topic model (HTM) is similar to Sudderth's scene model[21]. The performance of the HTM alone is close to the other two probabilistic models (pLSA+SVM) reported in [12] and discriminative LDA [6] which is evaluated on 13 categories. The method of "HTM+SVM" is a multi-class SVM with the input features of the latent topic assignments of HTM. Our hybrid model is a combination of neural network and HTM. We report the results of both pre-training and joint optimization, with the latter achieving a performance of 70.1%.

5.3 Results

We report the classification accuracy of the three types of methods in Table 1. Each model is trained on 5 random splits of training and test sets. We can see that the neural network, which lacks high-level representation, performs badly with classification accuracy of 51.6%. We also tried adding an extra hidden layer to better match the capacity of hybrid model, which resulted in a performance of 50.7%. The baseline HTM achieves 64.9% which is significantly better than neural network, and is comparable with other latent topic models LDA [6] (65.2%) and pLSA [12] (63.3%). The method of "HTM+SVM" which is a multi-class SVM using latent topic assignments of HTM as classification features, provides slightly improved predictions (65.5% vs 64.9%).

The hybrid model is analyzed after: i) pre-training and ii) full training with joint optimization. The pretrained hybrid model achieves 65.7% , slightly better than HTM, which shows that simple pre-trained feature transformation offers similar predictions. The fully trained hybrid model further improves the classification accuracy to 70.1% which is significantly better than HTM. It shows that joint optimization is capable of learning better low-level feature transformations for high-level topic modeling.

6 Discussion

We have introduced a unified representation that unifies two distinct classes of model that are widely used in machine learning and an end-to-end training scheme for the model. A number of improvements to our model could easily be incorporated. For example, a convolutional form of NN [13] could be used to directly learning from image pixels, or a spatial structure could be incorporated into the topic model, in the style of Sudderth *et al.* [21].

Our approach for joint training of the two models is a simple one that can be applied to more complex types of graphical model, provided (approximate) inference is possible in closed form. Finally, our model is not limited to image data and could easily be applied to other modalities such as text or audio.

References

[1] R. P. Adams, H. M. Wallach, and Z. Ghahramani. Learning the structure of deep sparse graphical models. *Journal of Machine Learning Research - Proceedings Track*, 9:1–8, 2010.

[2] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *In NIPS*, 2007.

[3] D. Blei and J. McAuliffe. Supervised topic models. In *NIPS*, 2007.

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

[5] C. Dance, J. Willamowski, L. Fan, C. Bray, and G. Csurka. Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision.*, Prague, 2004.

[6] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. *CVPR*, pages 524–531, 2005.

[7] G. E. Hinton and S. Osindero. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:2006, 2006.

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

[9] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. In *Machine Learning*, page 2001, 2001.

[10] K. Kavukcuoglu, P. Sermanet, Y. lan Boureau, K. Gregor, M. Mathieu, and Y. Lecun. Learning convolutional feature hierarchies for visual recognition, 2010.

[11] S. Lacoste-Julien, F. Sha, and M. Jordan. Disclda: Discriminative learning for dimensionality reduction and classification. In *NIPS*, 2009.

[12] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, pages 2169–2178, 2006.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *IEEE*, 86(11):2278–24, 1998.

[14] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML*, pages 609–616, 2009.

[15] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[16] J. Ngiam, Z. Chen, P. W. Koh, and A. Ng. Learning deep energy models. In L. Getoor and T. Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 1105–1112, New York, NY, USA, June 2011. ACM.

[17] M. Ranzato and G. Hinton. Modeling pixel means and covariances using factorized third-order Boltzmann Machines. In *CVPR*, 2010.

[18] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Uncertainty in Artificial Intelligence 20*, pages 487–494, 2004.

[19] R. Salakhutdinov and G. E. Hinton. Using deep belief nets to learn covariance kernels for gaussian processes. In *NIPS*, 2008.

[20] R. Salakhutdinov, J. Tenenbaum, and A. Torralba. Learning to learn with compound hd models. In *NIPS*, 2011.

[21] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Describing visual scenes using transformed objects and parts. *Intl. Journal of Computer Vision*, 77, March 2008.