Large-Scale Spike-and-Slab Sparse Coding for Unsupervised Feature Discovery

Ian J. Goodfellow, Aaron Courvile, Yoshua Bengio. Dept. IRO, U. Montreal

We consider the problem of using a factor model we call *spike-and-slab sparse coding* (S3C) to learn features for a classification task. The S3C model resembles both the spike-and-slab RBM and sparse coding. Since exact inference in this model is intractable, we derive a structured variational inference procedure and employ a variational EM training algorithm. Prior work on approximate inference for this model has not prioritized the ability to exploit parallel architectures and scale to enormous problem sizes. We present an inference procedure appropriate for use with GPUs which allows us to dramatically increase both the training set size and the amount of latent factors.

The S3C model The S3C model consists of latent binary *spike* variables $h \in \{0,1\}^N$, latent real-valued *slab* variables $s \in \mathbb{R}^N$, and real-valued D-dimensional visible vector $v \in \mathbb{R}^D$ generated according to this process: $\forall i \in \{1,\dots,N\}, d \in \{1,\dots,D\}$,

$$p(h_i = 1) = \sigma(b_i),\ \ p(s_i \mid h_i) = \mathcal{N}(s_i \mid h_i\mu_i, \alpha_{ii}^{-1}),\ \ \ p(v_d \mid s, h) = \mathcal{N}(v_d \mid W_{d:}(h \circ s), \beta_{dd}^{-1}) \tag{1}$$

where σ is the logistic sigmoid function, b is a set of biases on the spike variables, μ and W govern the linear dependence of s on h and v on s respectively, α and β are diagonal precision matrices of their respective conditionals, and $h \circ s$ denotes the element-wise product of h and s.

To avoid overparameterizing the distribution, we constrain the columns of W to have unit norm, as in sparse coding. We restrict α to be a diagonal matrix and β to be a diagonal matrix or a scalar. We refer to the variables h_i and s_i as jointly defining the i^{th} hidden unit, so that there are are total of N rather than $2N$ hidden units. The state of a hidden unit is best understood as $h_i s_i$, that is, the spike variables gate the slab variables.

Outside the context of unsupervised feature discovery for supervised, semi-supervised and self-taught learning, the basic form of the S3C model (i.e. a spike-and-slab latent factor model) has appeared a number of times in different domains (Lücke and Sheikh, 2011; Garrigues and Olshausen, 2008; Mohamed *et al.*, 2011; Titsias and Lázaro-Gredilla, 2011). However, the existing inference schemes have at most been applied to models with hundreds of bases and hundreds of thousands of examples. To this literature, we contribute an inference scheme that scales to the kinds of object classifications tasks that we consider, which require training thousands of bases on millions of examples.

Variational inference for S3C The goal of variational inference is to select a distribution Q over the latent variables that minimizes the Kullback–Leibler divergence:

$$\mathcal{D}_{KL}(Q(h,s) \| P(h,s|v)) \tag{2}$$

where $Q(h,s)$ is drawn from a tractable family of distributions. We choose $Q(h,s) = \Pi_i Q(h_i, s_i)$, which implies a solution of the form

$$Q(h_i) = \hat{h}_i,\ Q(s_i \mid h_i) = \mathcal{N}(s_i \mid h_i \hat{s}_i,\ (\alpha_i + h_i W_i^T \beta W_i)^{-1}) \tag{3}$$

where $\hat{h}_i$ and $\hat{s}_i$ must be found by an iterative process. Previous methods obtained poor runtimes on parallel architectures such as GPUs or suffered from instability. We propose a fast, stable method. First, we partially minimize the KL divergence with respect to $\hat{s}$. The terms of the KL divergence that depend on $\hat{s}$ make up a quadratic function so this can be minimized via conjugate gradient descent. We implement conjugate gradient descent efficiently by using the R-operator to perform Hessian-vector products rather than computing the entire Hessian explicitly (Schraudolph, 2002). This step is guaranteed to improve the KL divergence on each iteration.

We next update $\hat{h}$ in parallel, shrinking the update by a damping coefficient. This approach is not guaranteed to decrease the KL divergence on each iteration but it is a widely applied approach that works well in practice (Koller and Friedman, 2009). We find empirically that we can obtain a faster algorithm that reaches equally good solutions by replacing the conjugate gradient update to $\hat{s}$ with a more heuristic approach. The resulting algorithm is summarized in Algorithm 1.

CIFAR-10 Results We evaluated our features by using them in the object recognition pipeline of Coates and Ng (2011). On CIFAR-10, S3C achieves a test set accuracy of 78.3 $\pm$ 0.9 % with 95% confidence. Coates and Ng (2011) do not report test set accuracy for sparse coding with "natural encoding" (i.e., extracting

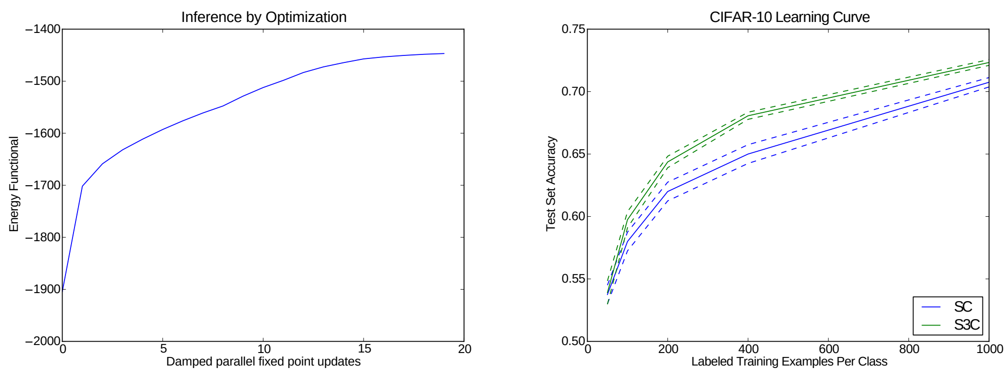

Figure 1: (Left) The energy functional of a batch of 5000 image patches increases during the E-step. (Right) Semi-supervised classification accuracy on CIFAR-10. In both cases the hyperparameters for the unsupervised stage were optimized for performance on the full CIFAR-10 dataset, not re-optimized for each point on the learning curve.

Algorithm 1 `Fixed-Point Inference`

Initialize $\hat{h}^{(0)} = \sigma(b)$ and $\hat{s}^{(0)} = \mu$.
for k=0:K **do**
Compute the individually optimal value $\hat{s}_i^*$ for each i simultaneously:

$$\hat{s}_i^* = \frac{\mu_i \alpha_{ii} + v^T \beta W_i - W_i \beta \left[\sum_{j \neq i} W_j \hat{h}_j \hat{s}_j^{(k)}\right]}{\alpha_{ii} + W_i^T \beta W_i}$$

Clip reflections by assigning

$$c_i = \rho \text{sign}(\hat{s}_i^*) |\hat{s}_i^{(k)}|$$

for all i such that $\text{sign}(\hat{s}_i^*) \neq \text{sign}(\hat{s}_i^{(k)})$ and $|\hat{s}_i^*| > \rho |\hat{s}_i^{(k)}|$, and assigning $c_i = \hat{s}_i^*$ for all other i.
Damp the updates by assigning

$$\hat{s}_i^{(k+1)} = \eta c + (1 - \eta) \hat{s}^{(k)}$$

where $\eta \in (0, 1]$.
Compute the individually optimal values for $\hat{h}$:

$$\hat{h}_i^* = \sigma \left(\left(v - \sum_{j \neq i} W_j \hat{s}_j^{(k+1)} \hat{h}_j^{(k)} - \frac{1}{2} W_i \hat{s}_i^{(k+1)} \right)^T \beta W_i \hat{s}_i^{(k+1)} + b_i \right.$$
$$\left. - \frac{1}{2} \alpha_{ii} (\hat{s}_i^{(k+1)} - \mu_i)^2 - \frac{1}{2} \log(\alpha_{ii} + W_i^T \beta W_i) + \frac{1}{2} \log(\alpha_{ii}) \right)$$

Damp the update to $\hat{h}$:

$$\hat{h}^{(k+1)} = \eta \hat{h}^* + (1 - \eta) \hat{h}^{(k)}$$

end for

features in a model whose parameters are all the same as in the model used for training) but sparse coding with different parameters for feature extraction than training achieves an accuracy of $78.8 \pm 0.9\%$ (Coates and Ng, 2011). Since we have not enhanced our performance by modifying parameters at feature extraction time these results seem to indicate that S3C is roughly equivalent to sparse coding for this classification task. S3C also outperforms ssRBMs, which achieve $76.7 \pm 0.9\%$ accuracy.

We also used CIFAR-10 to evaluate S3C's semi-supervised learning performance by training the SVM on small subsets of the CIFAR-10 training set, but using features that were learned on the entire CIFAR-10 train set. The results, summarized in Figure 1 (right) show that S3C is most advantageous for medium amounts of labeled data. S3C features thus include an aspect of flexible regularization– they improve generalization for smaller training sets yet do not cause underfitting on larger ones.

Transfer Learning Challenge For the NIPS 2011 Workshop on Challenges in Learning Hierarchical Models (Le *et al.*, 2011), the organizers proposed a transfer learning competition. This competition used a dataset consisting of 32×32 color images, including 100,000 unlabeled examples, 50,000 labeled examples of 100 object classes not present in the test set, and 120 labeled examples of 10 object classes present in the test set. The test set was not made public until after the competition. We disregarded the 50,000 labels and treated this as a semi-supervised learning task. We applied the same approach as on CIFAR-10 and won the competition, with a test set accuracy of 48.6 %.

References

Coates, A. and Ng, A. Y. (2011). The importance of encoding versus training with sparse coding and vector quantization. In *ICML 28*.

Garrigues, P. and Olshausen, B. (2008). Learning horizontal connections in a sparse coding model of natural images. In *NIPS'07*, pages 505–512. MIT Press, Cambridge, MA.

Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

Le, Q. V., Ranzato, M., Salakhutdinov, R., Ng, A., and Tenenbaum, J. (2011). *NIPS Workshop on Challenges in Learning Hierarchical Models: Transfer Learning and Optimization*. `https://sites.google.com/site/nips2011workshop`.

Lücke, J. and Sheikh, A.-S. (2011). A closed-form EM algorithm for sparse coding.

Mohamed, S., Heller, K., and Ghahramani, Z. (2011). Bayesian and l1 approaches to sparse unsupervised learning.

Schraudolph, N. N. (2002). Fast curvature matrix-vector products for second-order gradient descent. *Neural Computation*, **14**(7), 1723–1738.

Titsias, M. K. and Lázaro-Gredilla, M. (2011). Spike and slab variational inference for multi-task and multiple kernel learning. In *Advances in Neural Information Processing Systems 24*.