Variational Autoencoder for Deep Learning of Images, Labels and Captions

Yunchen Pu[†], Zhe Gan[†], Ricardo Henao[†], Xin Yuan[‡], Chunyuan Li[†], Andrew Stevens[†] and Lawrence Carin[†]

[†]Department of Electrical and Computer Engineering, Duke University; {yp42, zg27, r.henao, cl319, ajs104, lcarin}@duke.edu

[‡]Nokia Bell Labs, Murray Hill; xyuan@bell-labs.com

Introduction

The main contribution of this paper:

(i) A new VAE-based method for deep deconvolutional learning, with a CNN employed within a recognition model (encoder) for the posterior distribution of the parameters of the image generative model (decoder);

(ii) Demonstration that the fast CNN-based encoder applied to the DGDN yields accuracy comparable to that provided by Gibbs sampling and MCEM based inference, while being much faster at test time;

(iii) Applied semi-supervised CNN classification to large-scale image datasets;

(iv) Extensive experiments on image-caption modeling, in which we demonstrate the advantages of jointly learning the image features and caption model.

model

Image Decoder: Deep Deconvolutional Generative Model

Consider N images $\{\mathbf{X}^{(n)}\}_{n=1}^{N}$, with $\mathbf{X}^{(n)} \in \mathbb{R}^{N_x \times N_y \times N_c}$;

$$\text{Layer 2:} \quad \tilde{\mathbf{S}}^{(n,2)} = \sum_{k_2=1}^{K_2} \mathbf{D}^{(k_2,2)} * \mathbf{S}^{(n,k_2,2)} \tag{1}$$

$$\text{Unpool:} \quad \mathbf{S}^{(n,1)} \sim \text{unpool}(\tilde{\mathbf{S}}^{(n,2)}) \tag{2}$$

$$\text{Layer 1:} \quad \tilde{\mathbf{S}}^{(n,1)} = \sum_{k_1=1}^{K_1} \mathbf{D}^{(k_1,1)} * \mathbf{S}^{(n,k_1,1)} \tag{3}$$

$$\text{Data Generation:} \quad \mathbf{X}^{(n)} \sim \mathcal{N}(\tilde{\mathbf{S}}^{(n,1)}, \alpha_0^{-1}\mathbf{I}) \tag{4}$$

Image Encoder: Deep CNN

While the two-layer decoder in (1)-(4) is top-down, starting at layer 2, the encoder is bottom-up, starting at layer 1 with image $\mathbf{X}^{(n)}$:

$$\text{Layer 1:} \quad \tilde{\mathbf{C}}^{(n,k_1,1)} = \mathbf{X}^{(n)} *_s \mathbf{F}^{(k_1,1)} \quad , \quad k_1 = 1, \ldots, K_1 \tag{5}$$

$$\text{Pool:} \quad \mathbf{C}^{(n,1)} \sim \text{pool}(\tilde{\mathbf{C}}^{(n,1)}) \tag{6}$$

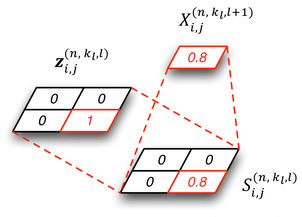
$$\text{Layer 2:} \quad \tilde{\mathbf{C}}^{(n,k_2,2)} = \mathbf{C}^{(n,1)} *_s \mathbf{F}^{(k_2,2)} \quad , \quad k_2 = 1, \ldots, K_2 \tag{7}$$

$$\text{Code Generation:} \quad \mathbf{s}_n \sim \mathcal{N}\left(\boldsymbol{\mu}_\phi(\tilde{\mathbf{C}}^{(n,2)}), \text{diag}(\boldsymbol{\sigma}_\phi^2(\tilde{\mathbf{C}}^{(n,2)}))\right) \tag{8}$$

$\boldsymbol{\mu}_\phi(\tilde{\mathbf{C}}^{(n,2)})$ and $\boldsymbol{\sigma}_\phi^2(\tilde{\mathbf{C}}^{(n,2)})$ in (8) is obtained by feeding $\tilde{\mathbf{C}}^{(n,2)}$ to an MLP.

Stochastic Unpooling/Pooling

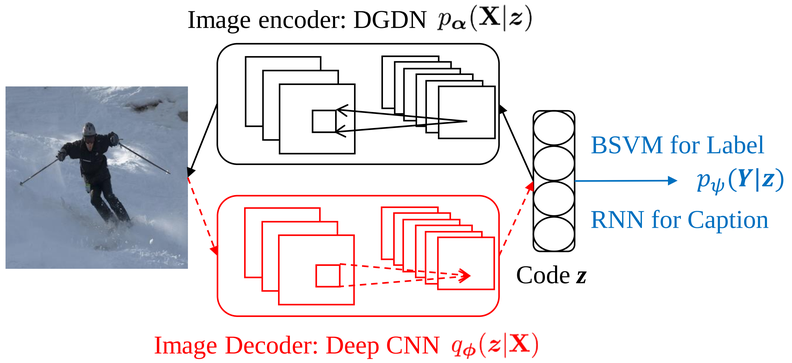
Stochastic Unpooling for Encoder: $\mathbf{S}^{(n,k_1,1)}$ is partitioned into contiguous spatial pooling blocks. Each pooling block of $\mathbf{S}^{(n,k_1,1)}$ is all-zeros except one non-zero element, with the value defined in $\mathbf{X}^{(n,k_1,2)}$ and location defined by $z_{i,j}^{(n,k_1,1)} \in \{0,1\}^{p_x p_y}$ which is a vector of all zeros, and a single one



$$z_{i,j}^{(n,k_1,1)} \sim \text{Mult}(1, 1/(p_x p_y), \ldots, 1/(p_x p_y)) \tag{9}$$

Stochastic Pooling for Decoder: $\tilde{C}_{i,j}^{(n,k_1,1)}$ reflect the $p_x p_y$ components in pooling block (i,j) of $\tilde{\mathbf{C}}^{(n,k_1,1)}$. Using a multi-layered perceptron (MLP), this is mapped to the $p_x p_y$-dimensional real vector. The pooling vector is drawn:

$$\boldsymbol{\eta}_{i,j}^{(n,k_1,1)} = \text{MLP}(\tilde{C}_{i,j}^{(n,k_1,1)}) \qquad z_{i,j}^{(n,k_1,1)} \sim \text{Mult}(1; \text{Softmax}(\boldsymbol{\eta}_{i,j}^{(n,k_1,1)})) \tag{10}$$



Label and Captions

Bayesian SVM for labels: Given a label $\ell_n \in \{1, \ldots, C\}$ associated with $\mathbf{X}^{(n)}$, we design C one-versus-all binary SVM classifiers, with $y_n^{(\ell)} \in \{-1, 1\}$. If $\ell_n = \ell$ then $y_n^{(\ell)} = 1$, and $y_n^{(\ell)} = -1$ otherwise. The goal of the SVM is to find an $f(\mathbf{s})$ that minimizes the objective function

$$\gamma \sum_{n=1}^{N} \max(1 - y_n f(\mathbf{s}_n), 0) + R(f(\mathbf{s})), \tag{11}$$

which is equivalent to estimating the mode of the pseudo-posterior of $\boldsymbol{\beta}$

$$p(\boldsymbol{\beta}|\mathbf{S}, \mathbf{y}, \gamma) \propto \prod_{n=1}^{N} \mathcal{L}(y_n|\mathbf{s}_n, \boldsymbol{\beta}, \gamma) p(\boldsymbol{\beta}|\cdot), \tag{12}$$

$$\mathcal{L}(y_n|\mathbf{s}_n, \boldsymbol{\beta}, \gamma) = e^{-2\gamma \max(1 - y_n \boldsymbol{\beta}^T \mathbf{s}_n, 0)} = \int_0^\infty \frac{\sqrt{\gamma}}{\sqrt{2\pi\lambda_n}} \exp\left(-\frac{(1 + \lambda_n - y_n \boldsymbol{\beta}^T \mathbf{s}_n)^2}{2\gamma^{-1}\lambda_n}\right) d\lambda_n. \tag{13}$$

RNN for captions: The probability of a cpation $\mathbf{Y}^{(n)} = (\mathbf{y}_1^{(n)}, \ldots, \mathbf{y}_{T_n}^{(n)})$ is defined as

$$p(\mathbf{Y}^{(n)}|\mathbf{s}_n) = p(\mathbf{y}_1^{(n)}|\mathbf{s}_n) \prod_{t=2}^{T_n} p(\mathbf{y}_t^{(n)}|\mathbf{y}_{<t}^{(n)}, \mathbf{s}_n) \tag{14}$$

$p(\mathbf{y}_t^{(n)}|\mathbf{y}_{<t}^{(n)}, \mathbf{s}_n)$ is specified as $\text{softmax}(\mathbf{V}\mathbf{h}_t^{(n)})$, where $\mathbf{h}_t^{(n)}$ is recursively updated through $\mathbf{h}_t^{(n)} = \mathcal{H}(\boldsymbol{w}_{t-1}^{(n)}, \mathbf{h}_{t-1}^{(n)})$ and $\mathcal{H}(\cdot)$ is implemented with GRU.

Learning and Inference

Given an image $\mathbf{X}$ and associated label/caption $\mathbf{Y}$, the variational lower bound is

$$\mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\alpha},\boldsymbol{\psi}}(\mathbf{X}, \mathbf{Y}) = \xi\{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{s}|\mathbf{X})}[\log p_{\boldsymbol{\psi}}(\mathbf{Y}|\mathbf{s})]\} + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{s},\boldsymbol{z}|\mathbf{X})}[\log p_{\boldsymbol{\alpha}}(\mathbf{X}, \mathbf{s}, \boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{s}, \boldsymbol{z}|\mathbf{X})]$$

where ξ is a tuning parameter that balances the two components of $\mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\alpha},\boldsymbol{\psi}}(\mathbf{X}, \mathbf{Y})$. The lower bound for the entire dataset is then:

$$\mathcal{J}_{\boldsymbol{\phi},\boldsymbol{\alpha},\boldsymbol{\psi}} = \sum_{(\mathbf{X},\mathbf{Y}) \in \mathcal{D}_c} \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\alpha},\boldsymbol{\psi}}(\mathbf{X}, \mathbf{Y}) + \sum_{\mathbf{X} \in \mathcal{D}_u} \mathcal{U}_{\boldsymbol{\phi},\boldsymbol{\alpha}}(\mathbf{X}) \tag{15}$$

where $\mathcal{D}_c$ denotes the set of training images with associated captions, and $\mathcal{D}_u$ is the set of training images that are uncaptioned (and unlabeled).

To optimize $\mathcal{J}_{\boldsymbol{\phi},\boldsymbol{\alpha},\boldsymbol{\psi}}$ w.r.t. $\boldsymbol{\phi}$, $\boldsymbol{\psi}$ and $\boldsymbol{\alpha}$, we utilize Monte Carlo integration to approximate the expectation, $\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{s},\boldsymbol{z}|\mathbf{X})}$, and stochastic gradient descent (SGD) for parameter optimization. We use the variance reduction techniques in VAE and NVIL to compute the gradients.

Experimental Results

Table 1 : Classification error (%) and testing time (ms per image) on benchmarks.

Method	MNIST		CIFAR-10		CIFAR-100		Caltech 101		Caltech 256		ImageNet 2012		
	test error	test time	test error	test time	test error	test time	test error	test time	test error	test time	top-1 error	top-5 error	test time
Gibbs	0.37	3.1	8.21	10.4	34.33	10.4	12.87	50.4	29.50	52.3	-	-	-
MCEM	0.45	0.8	9.04	1.1	35.92	1.1	13.51	8.8	30.13	8.9	37.9	16.1	14.4
VAE (Ours)	0.38	0.007	8.19	0.02	35.01	0.02	11.99	0.3	29.33	0.3	38.2	15.7	1.0

Table 2 : Semi-supervised classification error (%) on MNIST. N is the number of labeled images per class.

N	TSVM	Deep generative model		Ladder network		Our model	
		M1+TSVM	M1+M2	Γ-full	Γ-conv	$\xi = 0$	$\xi = N_x/(C\rho)$
10	16.81	11.82± 0.25	3.33 ± 0.14	1.06 ± 0.37	**0.89**±0.50	5.83 ± 0.97	1.49 ± 0.36
60	6.16	5.72± 0.05	2.59 ±0.05	-	0.82 ± 0.17*	2.19 ± 0.19	**0.77** ± 0.09
100	5.38	4.24± 0.07	2.40 ±0.02	0.84 ± 0.08	0.74 ± 0.10*	1.75 ± 0.14	**0.63** ± 0.06
300	3.45	3.49± 0.04	2.18 ±0.04	-	0.63 ± 0.02*	1.42 ± 0.08	**0.51** ± 0.04

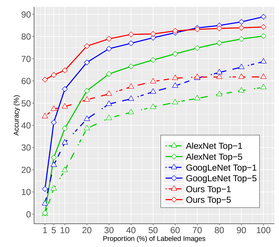


Figure 1 : Semi-supervised classification accuracy on ImageNet 2012.

Figure 2 : Examples of generated caption from unseen images on ImageNet.

Table 3 : Image captioning results on Flickr8k, Flickr 30k and MS COCO datasets.

Method	Flickr8k		Flickr30k		COCO			
	B-4	$\mathcal{PPL}$	B-4	$\mathcal{PPL}$	B-4	METEOR	CIDEr	$\mathcal{PPL}$
VggNet+RNN	0.16	15.71	0.17	18.83	0.19	0.19	0.56	13.16
GoogLeNet+RNN	0.16	15.71	0.17	18.77	0.17	0.19	0.55	14.01
Our two step model	0.17	15.82	0.17	18.73	0.18	0.20	0.58	13.46
Hard-Attention	0.21	-	0.20	- 16.17	0.25	0.23	-	-
Our joint model	0.22	15.24	0.22		0.26	0.22	0.89	11.57
Our joint model with ImageNet	0.25	13.24	0.25	15.34	0.28	0.24	0.90	11.14
Attributes-CNN+RNN	0.27	12.60	0.28	15.96	0.31	0.26	0.94	10.49

Conclusions

A recognition model has been developed for the Deep Generative Deconvolutional Network (DGDN). The model is learned using a variational autoencoder setup and achieved results competitive with state-of-the-art methods on several tasks and novel semi-supervised results.

Conference on Neural Information Processing Systems 2016 BARCELONA SPAIN