Adversarial Fisher Vectors For Unsupervised Representation Learning

Shuangfei Zhai, Walter Talbott, Carlos Guestrin, Joshua M. Susskind
Apple Inc.

Questions about GANs

- Is the discriminator useful at test time?
- Do GANs learn representations of data?
- Do you need to train an additional encoder?

Energy Based Model Interpretation of GANs

- The WGAN formulation

$$\max_G \min_D \mathrm{E}_{x \sim p_{data}(x)}[-D(x)] + \mathrm{E}_{z \sim p(z)}[D(G(z))] \quad (1)$$

- EBM with variational training has a dual form to a WGAN

$$\min_D \max_G \mathrm{E}_{x \sim p_{data}(x)}[-D(x)] + \mathrm{E}_{z \sim p(z)}[D(G(z))] + Entropy(p_G),\ s.t.\ p(x) = \frac{e^{D(x)}}{\int_x e^{D(x)} dx} \quad (2)$$

- Equation (1) and (2) can amount to the same practical implementation!

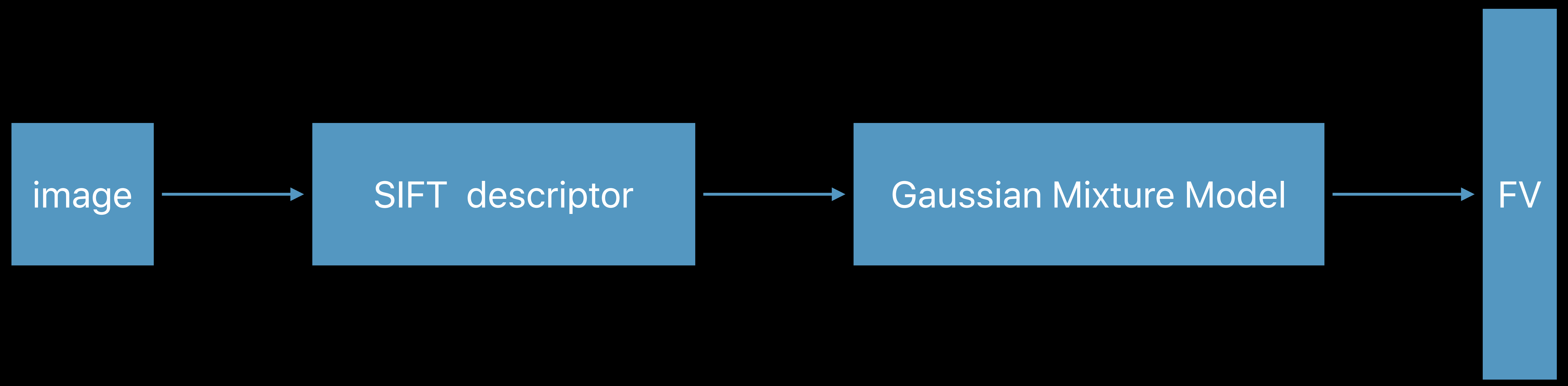
Generative Adversarial Networks as Variational Training of Energy Based Models, Zhai et. al.

Fisher Vectors

- Fisher vectors provide a way to represent an example given a probabilistic model

$$V_x = \mathrm{I}^{-\frac{1}{2}} \nabla_\theta \log p_\theta(x),\ s.t.,\ \mathrm{I} = \mathrm{E}_{x \sim p_\theta(x)}[\nabla_\theta \log p_\theta(x) \nabla_\theta \log p_\theta(x)^T]$$

- Has seen successful applications in computer vision



Exploiting Generative Models In Discriminative Classifiers, Jaakkola and Haussler

Adversarial Fisher Vectors

- Step 1: train a GAN and treat it as an EBM

- Step 2: compute the Adversarial Fisher Vector via:

$$V_{\mathbf{x}} = (diag(\mathrm{I})^{-\frac{1}{2}})U_{\mathbf{x}}$$

$$s.t.\ U_{\mathbf{x}} = \nabla_{\theta} D(\mathbf{x};\theta) - \mathrm{E}_{\mathbf{z}\sim p(\mathbf{z})} \nabla_{\theta} D(G(\mathbf{z});\theta),\ \mathrm{I} = \mathrm{E}_{\mathbf{z}\sim p(\mathbf{z})}[U_{G(\mathbf{z})} U^{T}_{G(\mathbf{z})}]$$

- Step 3: use V_x as the representation for downstream tasks (e.g., classification)

State-of-the-art Results on Linear Classification

Method	CIFAR10	CIFAR100	Method	#Features
Examplar CNN [29]	84.3	-	Unsupervised	-
DCGAN [38]	82.8	-	Unsupervised	-
Deep Infomax [39]	75.6	47.7	Unsupervised	1024
RotNet Linear [30]	81.8	-	Self-Supervised	$\sim$25K
AET Linear [32]	83.3	-	Self-Supervised	$\sim$25K
D-pool-128-50000	65.3	-	Unsupervised	512
AFV-128-50000	86.2	-	Unsupervised	1.5M
AFV-128-50000 + augment	87.1	-	Unsupervised	1.5M
AFV-256-50000 + augment	88.5	-	Unsupervised	5.9M
AFV-256-50000 + C100 + augment	**89.1**	**67.8**	Unsupervised	5.9M
D + BN supervised training	92.7	70.3	Supervised	-

Checkout the Paper and Code!