Probabilistic Graphical Models for Image Analysis - Lecture 10

Stefan Bauer

23rd November 2018

Max Planck ETH Center for Learning Systems

Overview

1. Recall Autoencoding Variational Bayes

2. Evaluating Deep Representation Learning

Recall Autoencoding Variational Bayes

Black Box is easy to implement!

Black-Box Stochastic Variational Inference in Five Lines of Python

David Duvenaud
dduvenaud@seas.harvard.edu
Harvard University

Ryan P. Adams
rpa@seas.harvard.edu
Harvard University

Abstract

Several large software engineering projects have been undertaken to support black-box inference methods. In contrast, we emphasize how easy it is to construct scalable and easy-to-use automatic inference methods using only automatic differentiation. We present a small function which computes stochastic gradients of the evidence lower bound for any differentiable posterior. As an example, we perform stochastic variational inference in a deep Bayesian neural network.

Gradient Estimates of the ELBO

$$\mathrm{ELBO} = \mathbb{E}_{q_\nu}[\log p_\theta(z, x)] - \mathbb{E}_q[\log q_\nu(z)]$$

where ν are the parameters of the variational distribution and θ the parameters of the model (as before).

Aim: Maximize the ELBO

Problem: Need unbiased estimates of $\nabla_{\nu,\theta}\mathrm{ELBO}$.

Algorithm

Algorithm 1 Black Box Variational Inference

1: **Input:** data x, model p(x,z), variational family $q_\varphi(z)$,
2: **while** Stopping criteria is not fulfilled **do**
3: Draw L samples $z_l \sim q_\varphi(z)$
4: Update variational paramater using the collected samples

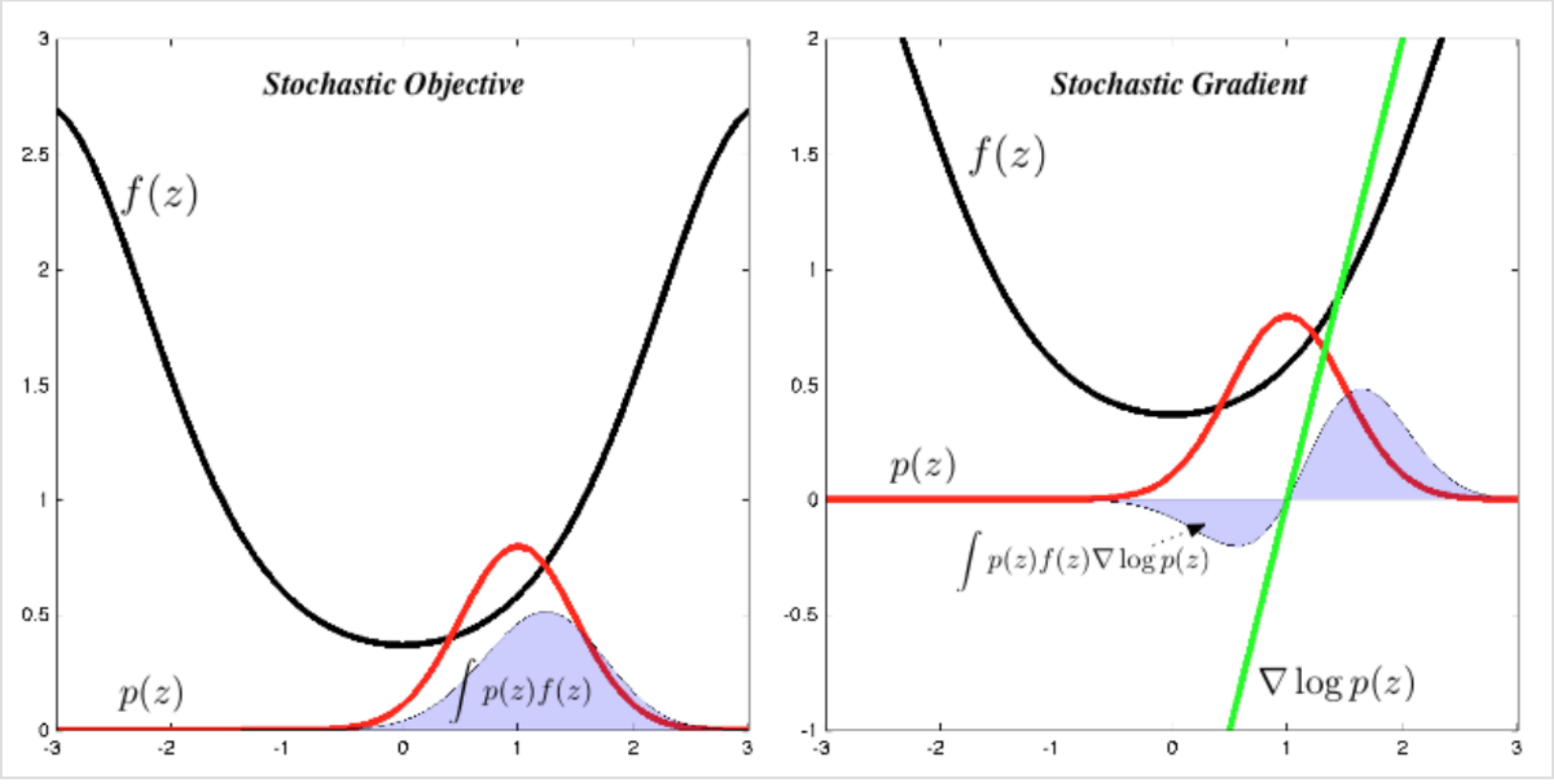
$$\varphi \leftarrow \varphi + \delta_t \frac{1}{L} \sum_{l=1}^{L} \nabla \log q(z_l)(\log p(x, z_l) - \log q(z_l))$$

5: Check step size and update if required!
6: **end while**

Note: Active research area (problem) is the reduction of the variance of the noisy gradient estimator.

Illustration*

$$\nabla_\theta \mathbb{E}_{p(z;\theta)}[f(z)] = \mathbb{E}_{p(z;\theta)}[f(z)\nabla_\theta \log p(z;\theta)]$$



*The Spectator Blog: ML Trick of the Day
http://blog.shakirm.com/2015/11/machine-learning-trick-of-the-day-5-log-derivative-trick/

Comparison

Score Function (reinforce)

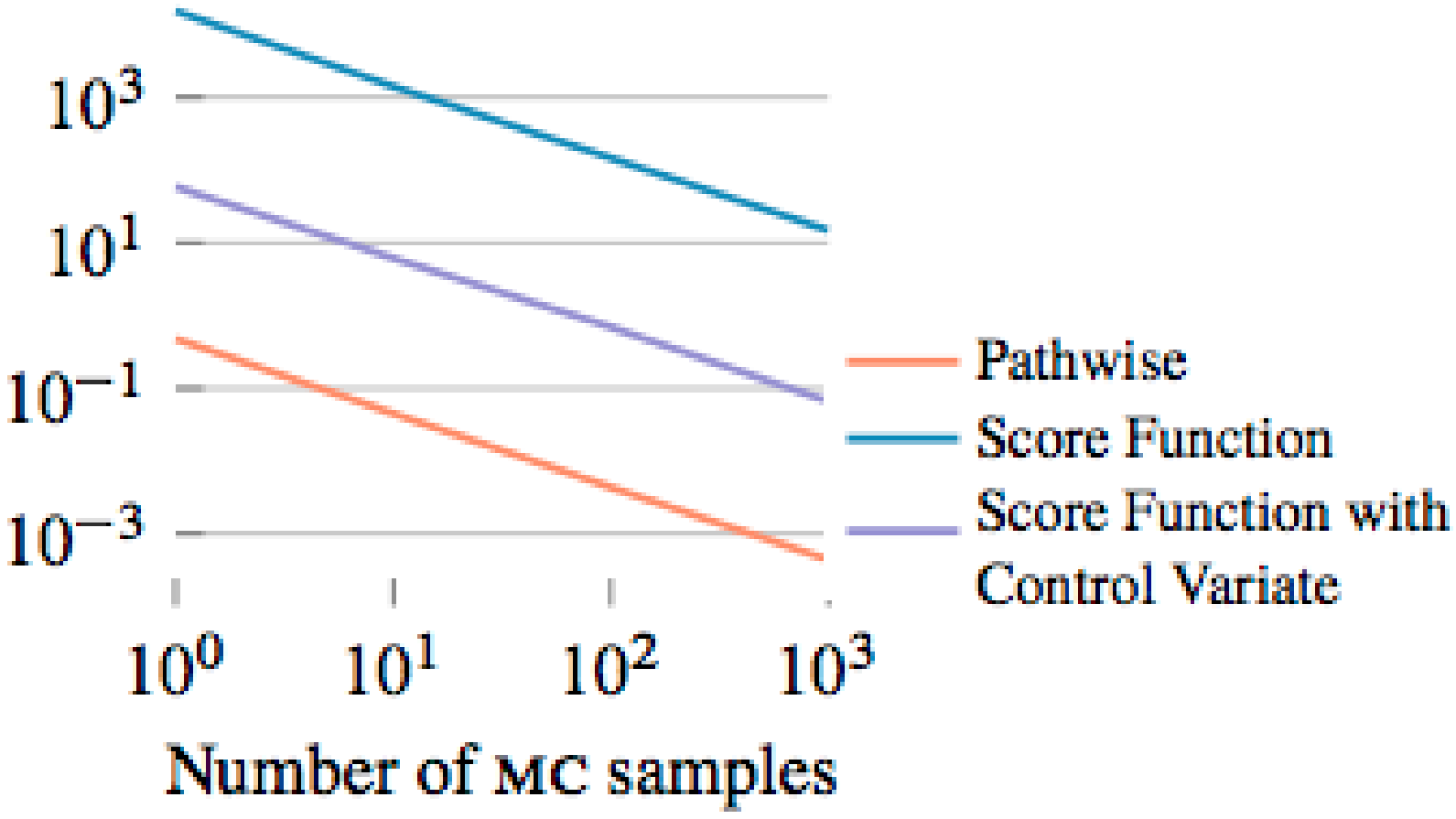
- Differentiates the density $\nabla_\nu q(z, \nu)$
- Works for discrete and continuous models
- Works for large class of variational approximations
- Variance is a big issue

Pathwise (reparameterization)

- Differentiates the function $\nabla_z[\log p(x, z) - \log q(z, \nu)]$
- requires differentiable models
- requires variational models to have special form
- In practice better behaved variance

Appendix D in `https://arxiv.org/pdf/1401.4082.pdf` provides a discussion about variance of both approaches.

Variance Comparison*



*NIPS Variational Inference Tutorial 2016
https://media.nips.cc/Conferences/2016/Slides/6199-Slides.pdf

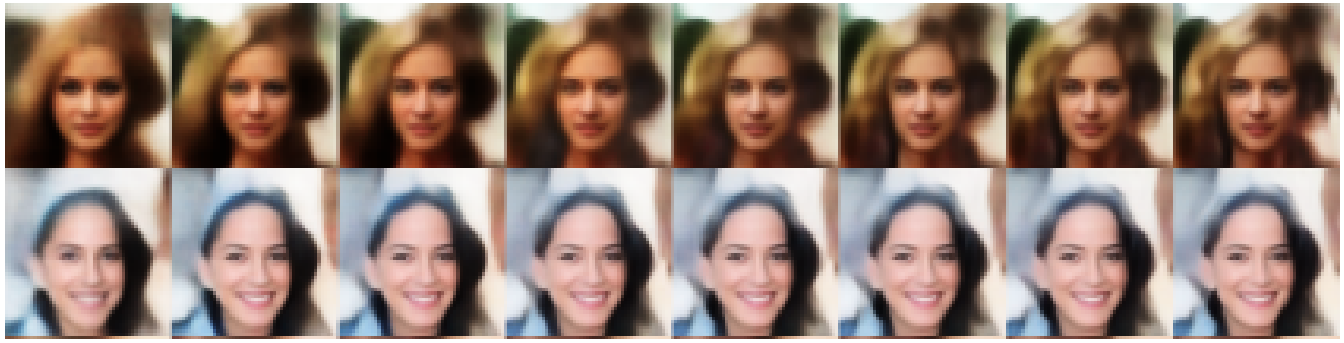
GAN Progress*

*Brundage et.al. 2018 https://img1.wsimg.com/blobby/go/3d82daa4-97fe-4096-9c6b-376b92c619de/downloads/1c6q2kc4v_50335.pdf

Progress GANs vs. State of the art VAE

GAN *

Recent VAE*



*Brundage et.al. 2018 https://arxiv.org/pdf/1802.07228.pdf
*Zhao et.al 2017 https://arxiv.org/pdf/1702.08658.pdf

Summary

Variational Autoencoders

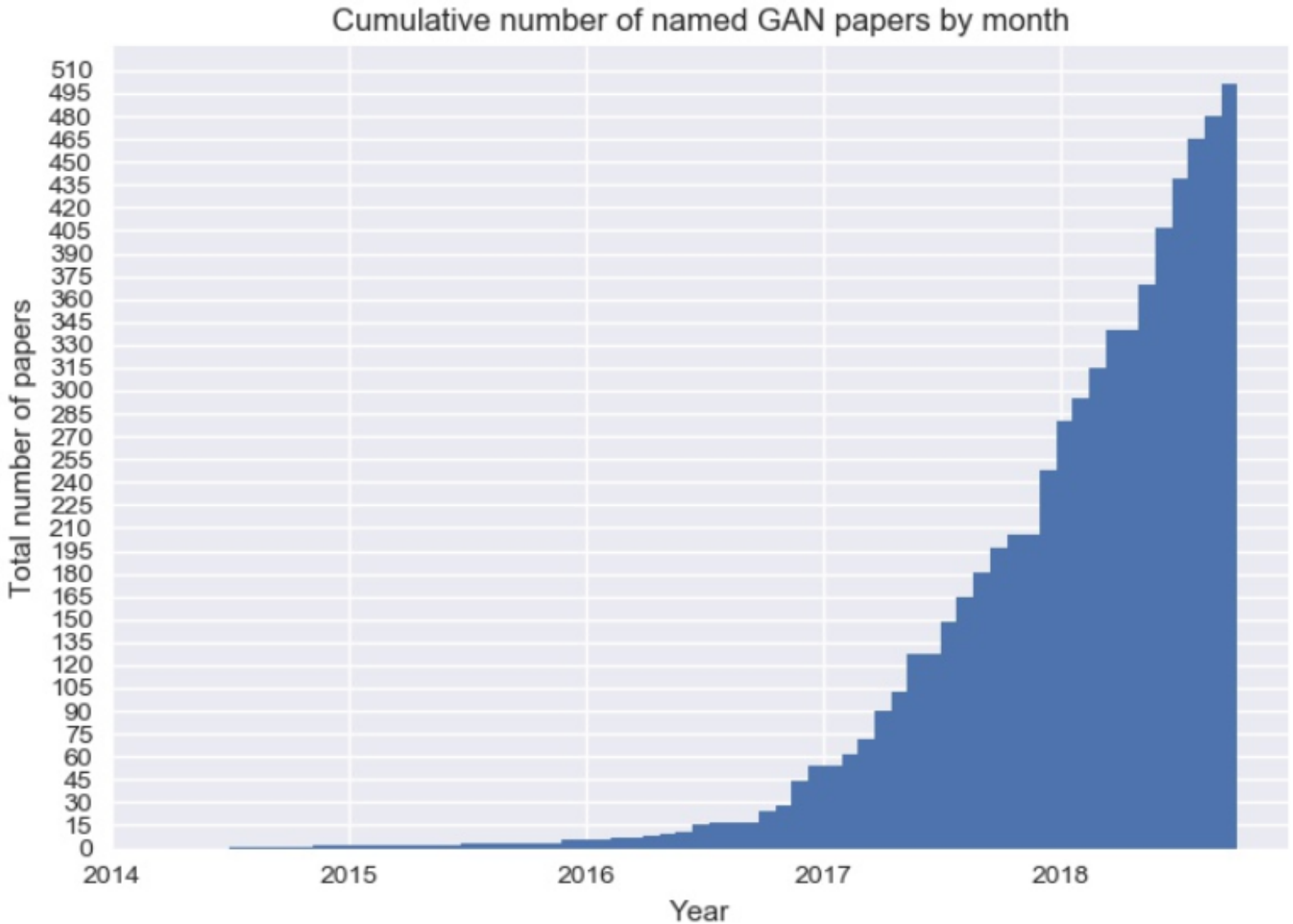
- for image generation, necessity to reconstruct each pixel
- reparametrization is not applicable to discrete latent variables
- usually only allows to use a fixed standard normal as a prior
- images are often blurry compared to high-fidelity samples generated by GANs
- allows for efficient Bayesian inference

Generative Adversarial Networks

- Instability of training
- mode collapse i.e. generated samples are often only from a few modes of the data distribution
- only visual inspection since GANs do not support inference (can additionally train an inference network)
- does not support discrete visible variables

Generally: We are unable to control the attributes of generated samples e.g. aim for regularization which enforces disentangled latent codes.

Gan Zoo*



*https://github.com/hindupuravinash/the-gan-zoo

Policy Gradient Methods for Reinforcement Learning with Function Approximation

Richard S. Sutton, David McAllester, Satinder Singh, Yishay Mansour
AT&T Labs – Research, 180 Park Avenue, Florham Park, NJ 07932

Abstract

Function approximation is essential to reinforcement learning, but the standard approach of approximating a value function and determining a policy from it has so far proven theoretically intractable. In this paper we explore an alternative approach in which the policy is explicitly represented by its own function approximator, independent of the value function, and is updated according to the gradient of expected reward with respect to the policy parameters. Williams's

Overview

Reinforcement learning of motor skills with policy gradients

Jan Peters [a,b,*], Stefan Schaal [b,c]

[a] Max Planck Institute for Biological Cybernetics, Spemannstr. 38, 72076 Tübingen, Germany
[b] University of Southern California, 3710 S. McClintoch Ave – RTH401, Los Angeles, CA 90089-2905, USA
[c] ATR Computational Neuroscience Laboratory, 2-2-2 Hikaridai, Seika-cho, Soraku-gun Kyoto 619-0288, Japan

ARTICLE INFO

Article history:
Received 11 September 2007
Received in revised form
24 February 2008
Accepted 24 February 2008

Keywords:
Reinforcement learning
Policy gradient methods
Natural gradients
Natural Actor-Critic
Motor skills
Motor primitives

ABSTRACT

Autonomous learning is one of the hallmarks of human and animal behavior, and understanding the principles of learning will be crucial in order to achieve true autonomy in advanced machines like humanoid robots. In this paper, we examine learning of complex motor skills with human-like limbs. While supervised learning can offer useful tools for bootstrapping behavior, e.g., by learning from demonstration, it is only reinforcement learning that offers a general approach to the final trial-and-error improvement that is needed by each individual acquiring a skill. Neither neurobiological nor machine learning studies have, so far, offered compelling results on how reinforcement learning can be scaled to the high-dimensional continuous state and action spaces of humans or humanoids. Here, we combine two recent research developments on learning motor control in order to achieve this scaling. First, we interpret the idea of modular motor control by means of motor primitives as a suitable way to generate parameterized control policies for reinforcement learning. Second, we combine motor primitives with the theory of stochastic policy gradient learning, which currently seems to be the only feasible framework for reinforcement learning for humanoids. We evaluate different policy gradient methods with a focus on their applicability to parameterized motor primitives. We compare these algorithms in the context of motor primitive learning, and show that our most modern algorithm, the Episodic Natural Actor-Critic outperforms previous algorithms by at least an order of magnitude. We demonstrate the efficiency of this reinforcement learning method in the application of learning to hit a baseball with an anthropomorphic robot arm.

© 2008 Elsevier Ltd. All rights reserved.

Gradient Estimation Using Stochastic Computation Graphs

John Schulman[1,2]
joschu@eecs.berkeley.edu

Nicolas Heess[1]
heess@google.com

Theophane Weber[1]
theophane@google.com

Pieter Abbeel[2]
pabbeel@eecs.berkeley.edu

[1] Google DeepMind

[2] University of California, Berkeley, EECS Department

Abstract

In a variety of problems originating in supervised, unsupervised, and reinforcement learning, the loss function is defined by an expectation over a collection of random variables, which might be part of a probabilistic model or the external world. Estimating the gradient of this loss function, using samples, lies at the core of gradient-based learning algorithms for these problems. We introduce

Policy Optimization*

Trust Region Policy Optimization

John Schulman JOSCHU@EECS.BERKELEY.EDU
Sergey Levine SLEVINE@EECS.BERKELEY.EDU
Philipp Moritz PCMORITZ@EECS.BERKELEY.EDU
Michael Jordan JORDAN@CS.BERKELEY.EDU
Pieter Abbeel PABBEEL@CS.BERKELEY.EDU
University of California, Berkeley, Department of Electrical Engineering and Computer Sciences

Abstract

We describe an iterative procedure for optimizing policies, with guaranteed monotonic improvement. By making several approximations to the theoretically-justified procedure, we develop a practical algorithm, called Trust Region Policy Optimization (TRPO). This algorithm is similar to natural policy gradient methods and is effective for optimizing large nonlinear policies such as neural networks. Our experiments demon-

Tetris is a classic benchmark problem for approximate dynamic programming (ADP) methods, stochastic optimization methods are difficult to beat on this task (Gabillon et al., 2013). For continuous control problems, methods like CMA have been successful at learning control policies for challenging tasks like locomotion when provided with hand-engineered policy classes with low-dimensional parameterizations (Wampler & Popović, 2009). The inability of ADP and gradient-based methods to consistently beat gradient-free random search is unsatisfying, since gradient-based optimization algorithms enjoy much better

*https://sites.google.com/site/trpopaper/

Some Scepticism*

arg min$_{blog}$ About

The Policy of Truth

Ben Recht • Feb 20, 2018

This is the sixth part of "An Outsider's Tour of Reinforcement Learning." Part 7 is here. Part 5 is here. Part 1 is here.

Our first generic candidate for solving reinforcement learning is *Policy Gradient*. I find it shocking that Policy Gradient wasn't ruled out as a bad idea in 1993. Policy gradient is seductive as it apparently lets one fine tune a program to solve any problem without any domain knowledge. Of course, anything that makes such a claim must be too general for its own good. Indeed, if you dive into it, **policy gradient is nothing more than random search dressed up in mathematical symbols and lingo**.

*http://www.argmin.net/2018/02/20/reinforce/

Evaluating Deep Representation Learning

The Problem - Metrics

How do we evaluate generative models?

- For tractable likelihood models: Evaluate generalization by reporting likelihoods on test data
- Proxy to likelihood might be available e.g. ELBO for VAEs.
- Visual Evaluation or e.g. using some metrics like Inception Scores, Frechet Inception Distance, Kernel Inception Distance based on heuristics like diversity, sharpness, similarities in feature representation ....

Difficult Problem (recall guest lecture) ....

Are GANs Created Equal? A Large-Scale Study

Mario Lucic* **Karol Kurach*** **Marcin Michalski** **Olivier Bousquet** **Sylvain Gelly**
Google Brain

How to evaluate latent representation

If available, we can evaluate the latent representation using the metric from a downstream task e.g. accuracy for semi-supervised learning.

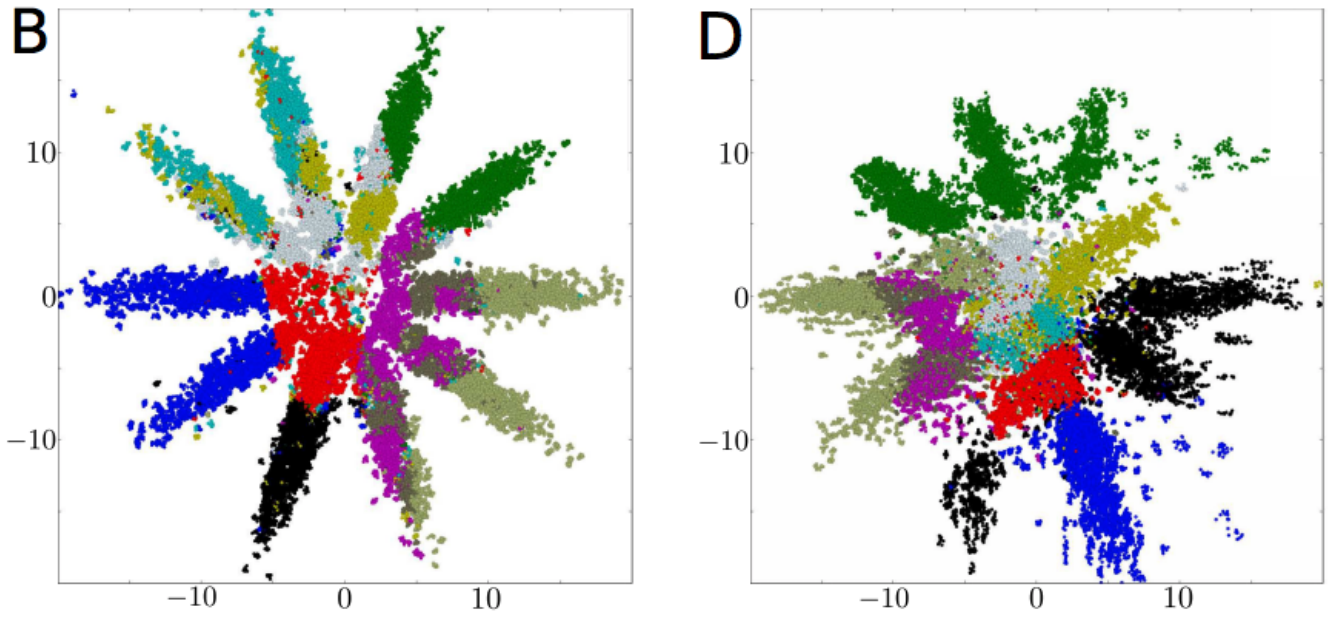
For unsupervised evaluations can be based on:

- Clustering using some attribute labels available e.g. color in MNIST.
- Compression
- ’Disentanglement’
- ....

Problem: At least partially a renaming of the problem i.e shifts the focus from how to evaluate representations to the evaluation of clusterings ...

Clustering*

Example: Cluster based on learned 2D representation, color denotes true labels.



Problem: How do we validate clustering? (Spring 2019: Prof. J. Buhmann *Statistical Learning Theory*)

*Makhzani et.al Adversarial Autoencoders 2016
https://arxiv.org/pdf/1511.05644.pdf

Next week

- Combination of deep representation learning with dynamics
- Disentanglement
- Summary

Questions?