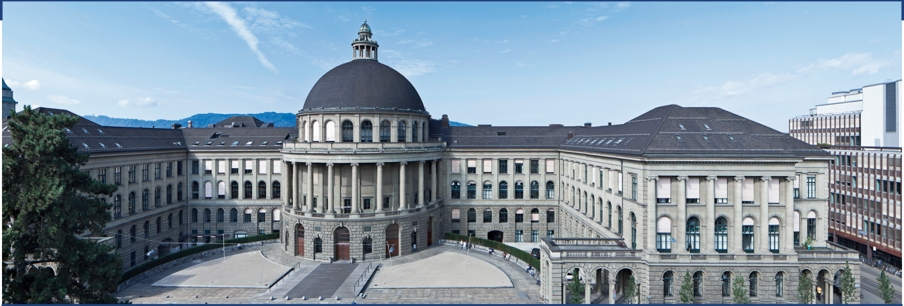

Seminar in Physics for CSE

Ardizzone et al.: Analyzing Inverse Problems with Invertible Neural Networks

Beat Hubmann

Outline

What Is an Inverse Problem?

INN-forward-process.png

[?]

What Is an Inverse Problem?

INN-inverse-process-direct.png

[?]

What Is an Inverse Problem?

INN-inverse-process-direct.png

ambiguous

[?]

What Is an Inverse Problem?

INN-inverse-process-direct.png

wanted: $p(x|y)$

[?]

Ambiguous Inverse Problem Go-To Nr. 1: MCMC

mcmc.jpg

- e.g. Metropolis-Hastings algorithm
- expensive

Inverse Problem Go-To Nr. 2:

Approximate Bayesian Computation

$$\pi(\theta|y) \propto p(y|\theta)\pi(\theta)$$

Inverse Problem Go-To Nr. 2:
Approximate Bayesian Computation

ABC_parameter_est.png

[?]

Inverse Problem Go-To Nr. 2:

Approximate Bayesian Computation

ABC_parameter_est.png

expensive

[?]

ABC.png

[?]

ABC1.png

ABC2.png

ABC3.png

Neural Network-based Approaches (1)

- predict fitting parameter of a distribution: restrictive

predict_fit.png

[?]

Neural Network-based Approaches (2)

- use variational network weights: still restrictive

uncertain_weights.png

Neural Network-based Approaches (3)

- use conditional Generative Adversarial Networks: yes, but ...

conditional_GAN.png

[?]

What is the difference between standard NN and invertible NN?

NNvsINN.png

Creating a Bijective Mapping: $x \leftrightarrow [y, z]$

INN-forward-process-with-z.png

Resolving the Ambiguity

INN-scheme.png

Resolving the Ambiguity

INN-scheme.png

Resolving the Ambiguity

INN-scheme.png

Invertible Neural Networks: Core Idea

$$s_1, s_2, t_1, t_2 : \mathbb{R} \to \mathbb{R}$$

$$u_1, u_2 \in \mathbb{R}$$

Invertible Neural Networks: Core Idea

$$s_1, s_2, t_1, t_2 : \mathbb{R} \to \mathbb{R}$$

$$u_1, u_2 \in \mathbb{R}$$

$$v_1 = u_1 \cdot \exp(s_2(u_2)) + t_2(u_2)$$

$$v_2 = u_2 \cdot \exp(s_1(v_1)) + t_1(v_1)$$

Invertible Neural Networks: Core Idea

$$s_1, s_2, t_1, t_2 : \mathbb{R} \to \mathbb{R}$$

$$u_1, u_2 \in \mathbb{R}$$

$$v_1 = u_1 \cdot \exp(s_2(u_2)) + t_2(u_2)$$

$$v_2 = u_2 \cdot \exp(s_1(v_1)) + t_1(v_1)$$

$$u_2 = \frac{v_2 - t_1(v_1)}{\exp(s_1(v_1))}$$

$$u_1 = \frac{v_1 - t_2(u_2)}{\exp(s_2(u_2))}$$

Invertible Neural Networks: Core Idea

$$s_1, s_2, t_1, t_2 : \mathbb{R} \to \mathbb{R}$$

$$u_1, u_2 \in \mathbb{R}$$

$$v_1 = u_1 \cdot \exp(s_2(u_2)) + t_2(u_2)$$

$$v_2 = u_2 \cdot \exp(s_1(v_1)) + t_1(v_1)$$

$$u_2 = \frac{v_2 - t_1(v_1)}{\exp(s_1(v_1))}$$

$$u_1 = \frac{v_1 - t_2(u_2)}{\exp(s_2(u_2))}$$

s_i and t_i can be arbitrarily complicated functions:
no need to be invertible themselves,
hence can use trainable functions

INN Main Building Block: Affine Coupling Layer

forward process:

INN-coupling-layer.png

Characterization of Invertible Neural Networks

1. mapping bijective: has inverse
2. forward and inverse mapping efficiently computable
3. forward and inverse mapping with tractable Jacobian

Training Scheme

INN-training-scheme.png

Training Scheme

INN-training-scheme.png

Loss function for backward loss $\mathcal{L}_z$:
Maximum Mean Discrepancy (MMD)

given:

$X = \{x_1, \dots, x_m\} \sim p$,

$Y = \{y_1, \dots, y_n\} \sim q$

test if $p = q$

MMD.png

[?]

- Kullback-Leibler divergence or L^1/L^2 distance compare $\hat{p}, \hat{q}$: indirect measure

Loss function for backward loss $\mathcal{L}_z$: Maximum Mean Discrepancy (MMD)

given:

$X = \{x_1, \ldots, x_m\} \sim p$,

$Y = \{y_1, \ldots, y_n\} \sim q$

test if $p = q$

MMD.png

[?]

- Kullback-Leibler divergence or L^1/L^2 distance compare $\hat{p}, \hat{q}$: indirect measure
- MMD uses kernel trick: direct measure

Outline

Toy Example: Gaussian Mixture Model

[**?**]

Toy Example: Gaussian Mixture Model

[**?**]

Real-World Example: Biological Tissue Parameters from Multispectral Image

tissue-data.png

Real-World Example: Biological Tissue Parameters from Multispectral Image

tissue-table.png

Outline

Benefits of the INN Method

- Very good quantitative and qualitative results
- Relatively easy, cheap and straightforward to train

Challenges of the INN Method

- How to decide the intrinsic dimension of the data?
- How to decide splitting x into u_1 and u_2?
- How to decide permutation of the streams u_1 and u_2 between coupling layers?

My Take

My Take

positive.png

References I

L. Ardizzone et al.
Analyzing Inverse Problems with Invertible Neural Networks.
ICLR 2019 conference paper, arXiv:1808.04730, 2019.

L. Ardizzone et al.
Analyzing Inverse Problems with Invertible Neural Networks.
Visual Learning Lab Heidelberg, Blog Post, 2018.

L. Dinh et al.
Density Estimation using Real NVP.
ICLR 2017 conference paper, arXiv:1605.08803, 2016.

References II

A. Smola
Maximum Mean Discrepancy.
ICONIP 2006 conference presentation, Alexander Smola's Personal Page (Retrieved Sep 26, 2019), 2006.

M. Mirza and S. Osindero
Conditional Generative Adversarial Nets.
arXiv:1411.1784, 2014.

C. Blundell et al.
Weight Uncertainty in Neural Networks.
Google DeepMind, arXiv:1505.05424, 2015.

References III

D. Nix and A. Weigend
Estimating the Mean and Variance of the Target Probability Distribution.
ICNN94 conference paper, DOI:10.1109/ICNN.1994.374138, 1994

M. Sunnåker et al.
Approximate Bayesian Computation.
PLoS Comput Biol 9(1): e1002803, DOI:10.1371/journal.pcbi.1002803, 2013