Priors for Bayesian Deep Learning

Maurizio Filippone

Department of Data Science, EURECOM

August 30^{th}, 2023

Outline

Motivation

Bayesian Deep Learning

Some Emerging Trends in Bayesian Deep Learning

Ongoing Work

Motivation

Decision-Making

Decision-making is a critical step in several domains [**Norvig and Russell**, **1995**]:

- Policy-making for the environment
- Healthcare
- Society
- ...

Decision-Making

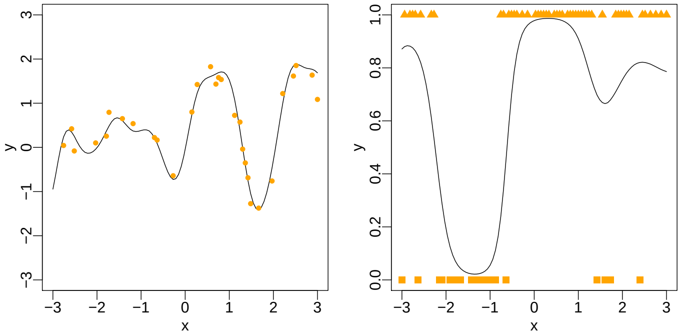
Decision-making is a critical step in several domains [**Norvig and Russell**, **1995**]:

- Policy-making for the environment
- Healthcare
- Society
- ...

$$\text{Decision Theory} = \text{Probabilistic reasoning} + \text{Utility theory}$$

Learning from Data – Function Estimation

- Consider these two examples



- We are interested in estimating a function $\mathbf{f}(\mathbf{x})$ from data
- Many problems in Statistics/Machine Learning can be cast this way!

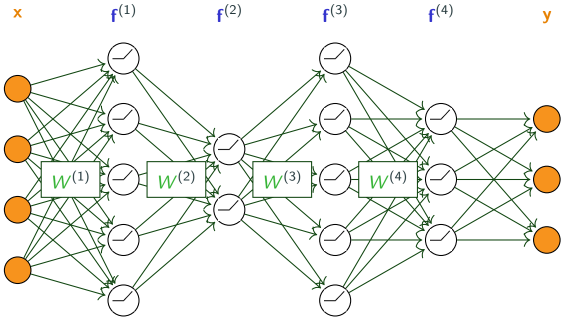
Deep Neural Networks

- Implement a composition of parametric functions

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}^{(L)}\left(\mathbf{f}^{(L-1)}\left(\cdots \mathbf{f}^{(1)}(\mathbf{x})\cdots\right)\right)$$

with

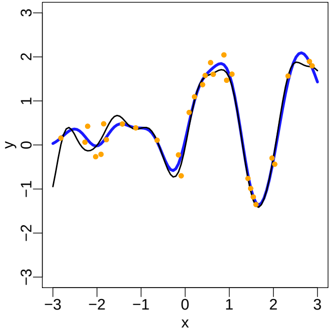
$$\mathbf{f}^{(l)}(\mathbf{h}) = \mathbf{g}\left(\mathbf{W}^{(l)}\mathbf{h}\right)$$

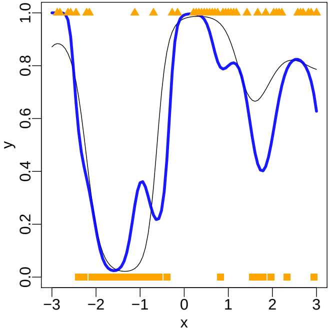


Optimizing Deep Nets

- Quadratic Loss Minimization (regression case):

$$\hat{\mathbf{W}} = \arg\min_{\mathbf{W}} \sum_i \|\mathbf{y}_i - \mathbf{f}(\mathbf{x}_i)\|^2 + \text{regularization}$$





Over-confidence of Deep Learning Models

> *"What we know is that the vehicle was on a divided highway with Autopilot engaged when a tractor trailer drove across the highway perpendicular to the Model S. Neither Autopilot nor the driver noticed the white side of the tractor trailer against a brightly lit sky, so the brake was not applied. The high ride height of the trailer combined with its positioning across the road and the extremely rare circumstances of the impact caused the Model S to pass under the trailer, with the bottom of the trailer impacting the windshield of the Model S."*

Over-confidence of Deep Learning Models

Uber suspends self-driving car testing after cyclist is killed

The company says it is "fully co-operating with local authorities in their investigation of this incident" and offers condolences.

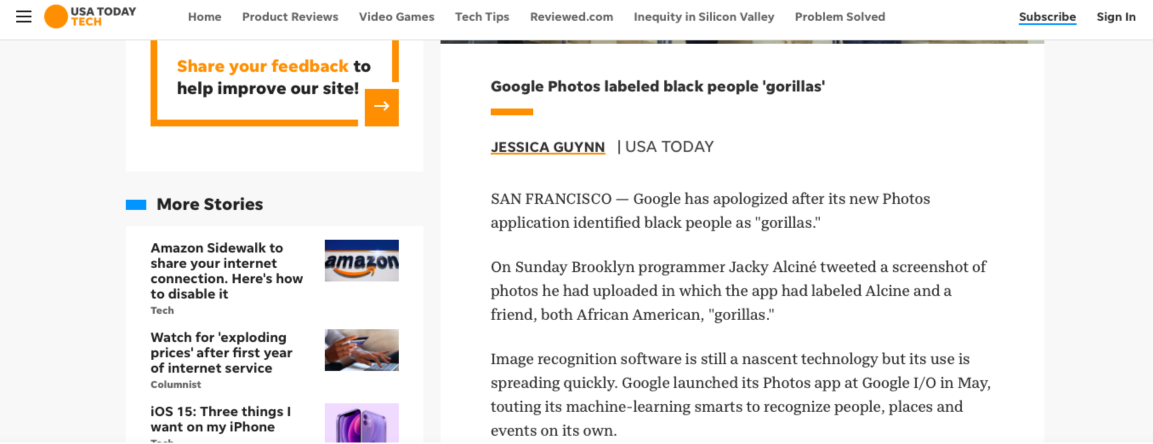
Tuesday 20 March 2018 06:07, UK

Damage on the front of the self-driving car

Why you can trust Sky News ›

Uber has suspended testing of its self-driving cars after one struck and killed a female cyclist in Phoenix.

Over-confidence of Deep Learning Models



USA TODAY TECH

Home | Product Reviews | Video Games | Tech Tips | Reviewed.com | Inequity in Silicon Valley | Problem Solved

Subscribe | Sign In

Share your feedback to help improve our site!

More Stories

Amazon Sidewalk to share your internet connection. Here's how to disable it
Tech

Watch for 'exploding prices' after first year of internet service
Columnist

iOS 15: Three things I want on my iPhone

Google Photos labeled black people 'gorillas'

JESSICA GUYNN | USA TODAY

SAN FRANCISCO — Google has apologized after its new Photos application identified black people as "gorillas."

On Sunday Brooklyn programmer Jacky Alciné tweeted a screenshot of photos he had uploaded in which the app had labeled Alcine and a friend, both African American, "gorillas."

Image recognition software is still a nascent technology but its use is spreading quickly. Google launched its Photos app at Google I/O in May, touting its machine-learning smarts to recognize people, places and events on its own.

Over-confidence of Deep Learning Models - Online Meme

Image prediction: ping-pong ball
Confidence: 99.99%

Illustration: Dianna “Mick” McDougall, Photo: ResNeXtGuesser

Over-confidence of Deep Learning Models - Online Meme

Image prediction: pineapple
Confidence: 99.3%

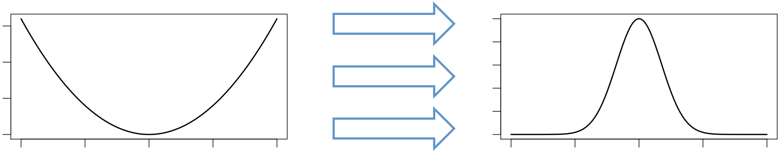
Illustration: Dianna "Mick" McDougall, Photo: ResNeXtGuesser

Bayesian Deep Learning

Back-propagation – Probabilistic Interpretation Loss

- Inputs : $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$
- Labels : $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_N\}$
- Weights : $\mathbf{W} = \{\mathbf{W}^{(1)}, \dots, \mathbf{W}^{(L)}\}$

Quadratic Loss $\qquad\qquad$ $p(\mathbf{Y}|\mathbf{X}, \mathbf{W}) \propto \exp(-\mathrm{Loss})$

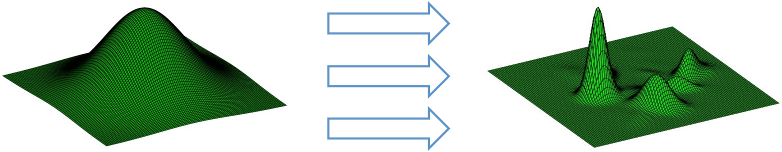


- Back-propagation minimizes a loss function
- … equivalent as optimizing likelihood $p(\mathbf{Y}|\mathbf{X}, \mathbf{W})$

Bayesian Inference

- Inputs : $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$
- Labels : $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$
- Weights : $\mathbf{W} = \{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L)}\}$

$p(\mathbf{W})$ $\qquad$ $p(\mathbf{W}|\mathbf{Y}, \mathbf{X})$

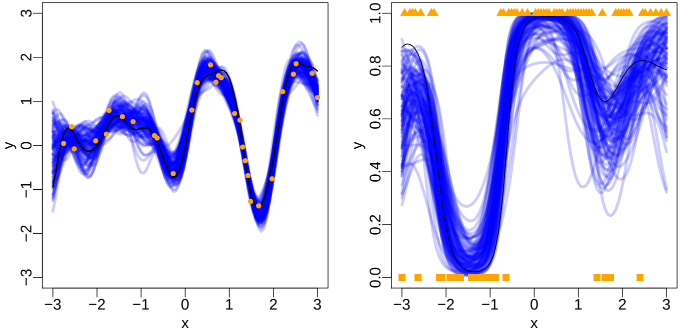


$$p(\mathbf{W}|\mathbf{Y}, \mathbf{X}) = \frac{p(\mathbf{Y}|\mathbf{X}, \mathbf{W})p(\mathbf{W})}{\int p(\mathbf{Y}|\mathbf{X}, \mathbf{W})p(\mathbf{W})d\mathbf{W}}$$

Bayesian Deep Nets

- Predictions consider an infinite number of parameter configurations

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{Y}, \mathbf{X}) = \int p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{W})p(\mathbf{W}|\mathbf{Y}, \mathbf{X})d\mathbf{W}$$



Bayesian Deep Learning Time-line

- Bayesian Deep Nets have been though about since the nineties [**MacKay**, **1992**]
- Deep Nets as Gaussian processes [**Neal**, **1996**]

Bayesian Deep Learning Time-line

- Bayesian Deep Nets have been though about since the nineties [**MacKay**, **1992**]
- Deep Nets as Gaussian processes [**Neal**, **1996**]
- Mini-Batch variational inference for Deep Nets [**Graves**, **2011**]
- Mini-Batch MCMC sampling [**Chen et al.**, **2014**]
- TensorFlow is released [**Abadi et al.**, **2016**]

Bayesian Deep Learning Time-line

- Bayesian Deep Nets have been though about since the nineties [**MacKay**, **1992**]
- Deep Nets as Gaussian processes [**Neal**, **1996**]
- Mini-Batch variational inference for Deep Nets [**Graves**, **2011**]
- Mini-Batch MCMC sampling [**Chen et al.**, **2014**]
- TensorFlow is released [**Abadi et al.**, **2016**]
- Dropout as Variational Inference [**Gal and Ghahramani**, **2016**]

First ever practical approach for **approximate** Bayesian Conv Nets

Bayesian Deep Learning Time-line

- Bayesian Deep Nets have been though about since the nineties [**MacKay**, **1992**]
- Deep Nets as Gaussian processes [**Neal**, **1996**]
- Mini-Batch variational inference for Deep Nets [**Graves**, **2011**]
- Mini-Batch MCMC sampling [**Chen et al.**, **2014**]
- TensorFlow is released [**Abadi et al.**, **2016**]
- Dropout as Variational Inference [**Gal and Ghahramani**, **2016**]

  First ever practical approach for **approximate** Bayesian Conv Nets

- First workshop on Bayesian Deep Learning at NeurIPS 2016

Challenges with Bayesian Deep Learning

- **Urban legend**: Slow and cumbersome to tune/implement compared to optimization

Challenges with Bayesian Deep Learning

- **Urban legend**: Slow and cumbersome to tune/implement compared to optimization
- Predictive **performance is usually worse** than non-Bayesian solutions
  - People started questioning the optimality of Bayesian principles 😱
  - Literature flooded with alternative approaches

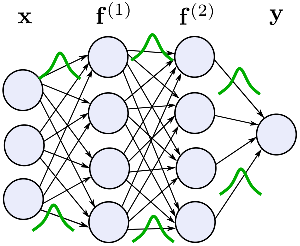
Challenges with Bayesian Deep Learning

- **Urban legend**: Slow and cumbersome to tune/implement compared to optimization
- Predictive **performance is usually worse** than non-Bayesian solutions
  - People started questioning the optimality of Bayesian principles 😱
  - Literature flooded with alternative approaches
  - Improvements to Variational Inference for deep models
    [**Rossi et al.**, **ICML 2019**, **NeurIPS 2020**]

Challenges with Bayesian Deep Learning

- **Urban legend**: Slow and cumbersome to tune/implement compared to optimization
- Predictive **performance is usually worse** than non-Bayesian solutions
    - People started questioning the optimality of Bayesian principles 😱
    - Literature flooded with alternative approaches
    - Improvements to Variational Inference for deep models [**Rossi et al.**, **ICML 2019**, **NeurIPS 2020**]
- The problem of choosing sensible priors has been overlooked!

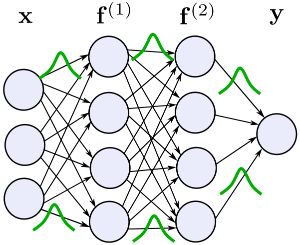
Prior for Bayesian Neural Networks



$$\mathbf{w}_i \sim \mathcal{N}(0, \mathbf{I})$$

Specifying a sensible prior for Bayesian neural networks (BNNs) is difficult!

Prior for Bayesian Neural Networks



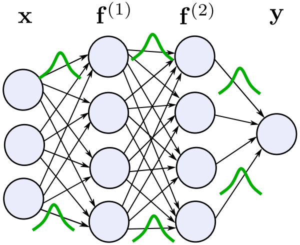
$$\mathbf{w}_i \sim \mathcal{N}(0, \mathbf{I})$$

Specifying a sensible prior for Bayesian neural networks (BNNs) is difficult!

- Neural networks are extremely **high-dimensional** and **nonidentifiable**.

  $\longrightarrow$ Reasoning about parameters is very challenging.

Prior for Bayesian Neural Networks



$$\mathbf{w}_i \sim \mathcal{N}(0, \mathbf{I})$$

Specifying a sensible prior for Bayesian neural networks (BNNs) is difficult!

- Neural networks are extremely **high-dimensional** and **nonidentifiable**.
  $\longrightarrow$ Reasoning about parameters is very challenging.
- Most work has resorted to priors of convenience.
  $\longrightarrow$ Gaussian priors such as $\mathcal{N}(0, 1)$ and $\mathcal{N}(0, 1/D_{l-1})$ are the most popular priors for BNN.

Prior for Bayesian Neural Networks

The prior on the parameters of a BNN induces an *unpredictable prior over functions*.

$$p(\mathbf{f}) = \int p(\mathbf{f} \mid \mathbf{w}) p(\mathbf{w}) d\mathbf{w}$$

Some Emerging Trends in Bayesian Deep Learning

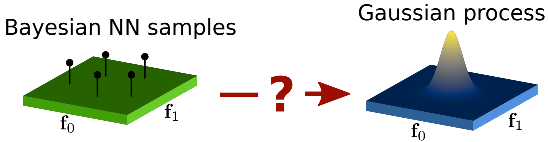
Gaussian Process Priors

- Gaussian Processes (GPs) are a useful tool for choosing *sensible priors* on *functions we intend to model*.
- A popular covariance function is the radial basis function (RBF):

$$\kappa_{\alpha,l}(\mathbf{x}, \mathbf{x}') = \alpha^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{l^2}\right).$$

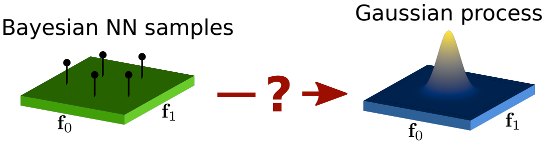
Research Question

How to impose functional priors on BNNs exhibit interpretable properties, similar to GPs?



Research Question

How to impose functional priors on BNNs exhibit interpretable properties, similar to GPs?

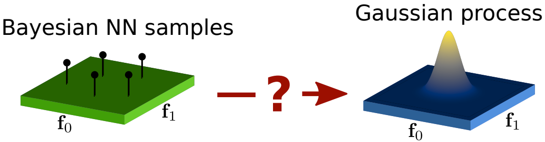


This is a challenging task!

- We aim at matching two stochastic processes $\rightarrow$ infinite-dimensional distributions.
- We don't know closed-form of the density of BNNs.

Research Question

How to impose functional priors on BNNs exhibit interpretable properties, similar to GPs?



This is a challenging task!

- We aim at matching two stochastic processes $\rightarrow$ infinite-dimensional distributions.
- We don't know closed-form of the density of BNNs.
  - Minimize the KL divergence between BNN and GP priors.

$$\mathrm{KL}\left[p_{nn} \,\|\, p_{gp}\right] = -\int p_{nn}(\mathbf{f};\psi) \log p_{gp}(\mathbf{f})\, \mathrm{d}\mathbf{f} + \underbrace{\int p_{nn}(\mathbf{f};\psi) \log p_{nn}(\mathbf{f};\psi)\, \mathrm{d}\mathbf{f}}_{\text{Entropy -- intractable!}} .$$

Wasserstein distance

Definition
Given a measurable space Ω, the Kantorovich dual form of the 1-Wasserstein distance between two Borel's probability measures π and ν in $\mathcal{P}(\Omega)$ is

$$W_1(\pi, \nu) = \sup_{\|\phi\|_L \leq 1} \mathbb{E}_{\pi}[\phi(\mathbf{x})] - \mathbb{E}_{\nu}[\phi(\mathbf{x})]\,,$$

where ϕ is a 1-Lipschitz function.

Wasserstein distance

Definition
Given a measurable space Ω, the Kantorovich dual form of the 1-Wasserstein distance between two Borel's probability measures π and ν in $\mathcal{P}(\Omega)$ is

$$W_1(\pi, \nu) = \sup_{\|\phi\|_L \leq 1} \mathbb{E}_{\pi}[\phi(\mathbf{x})] - \mathbb{E}_{\nu}[\phi(\mathbf{x})]\,,$$

where ϕ is a 1-Lipschitz function.

✓ No need to know the closed-form of π and ν as we can estimate expectations with samples.

✓ The 1-Lipschitz function ϕ can be parameterized by a neural network.

Proposed Method

- Minimize the 1-Wasserstein distance between the BNN functional prior and a GP prior

✓ The objective is *fully sampled-based*!

Proposed Method

- Minimize the 1-Wasserstein distance between the BNN functional prior and a GP prior

✓ The objective is *fully sampled-based*!

$\longrightarrow$ Not necessary to know the closed-form of the marginal density $p_{nn}(\mathbf{f}; \psi)$.

Proposed Method

- Minimize the 1-Wasserstein distance between the BNN functional prior and a GP prior

✓ The objective is *fully sampled-based*!

$\longrightarrow$ Not necessary to know the closed-form of the marginal density $p_{nn}(\mathbf{f}; \psi)$.

$\longrightarrow$ Can consider any stochastic process as a target prior over functions.

Proposed Method

- Minimize the 1-Wasserstein distance between the BNN functional prior and a GP prior

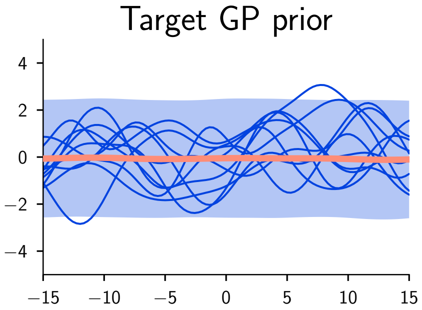
✓ The objective is *fully sampled-based*!

$\longrightarrow$ Not necessary to know the closed-form of the marginal density $p_{nn}(\mathbf{f}; \psi)$.

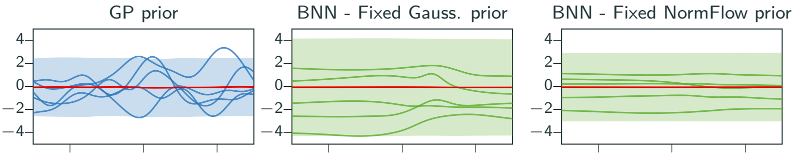
$\longrightarrow$ Can consider any stochastic process as a target prior over functions.

✓ The objective can be optimized with gradient descent algorithms with back-propagation.

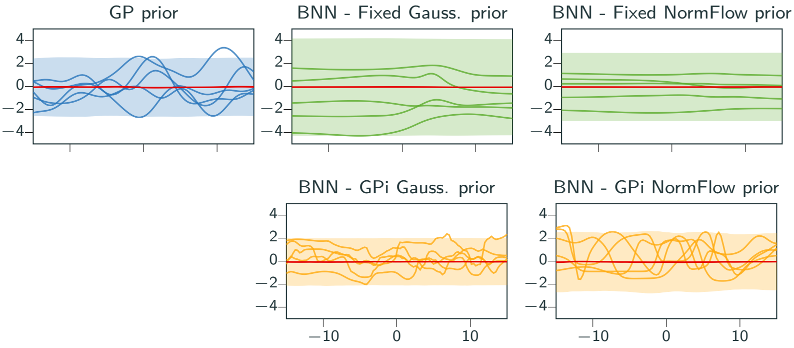
Matching BNN Prior to GP Prior



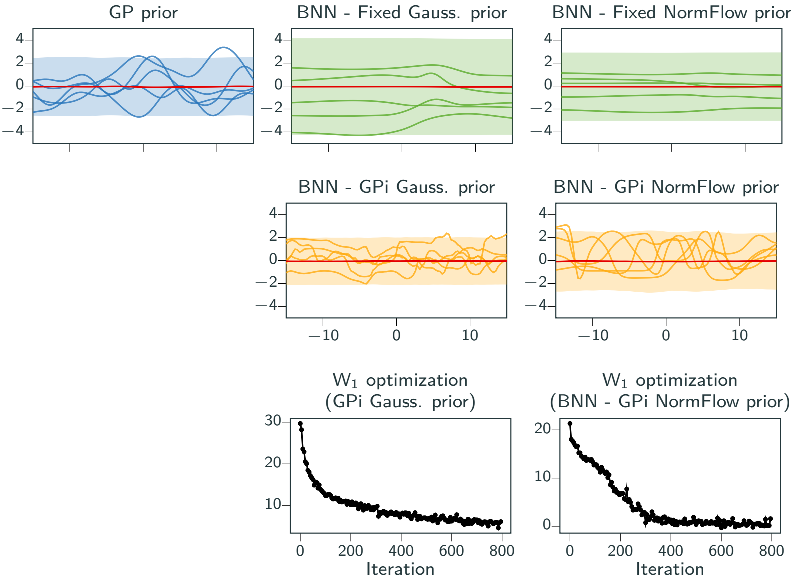
1D Regression Synthetic Data



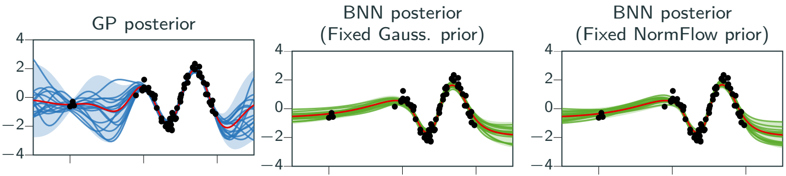
1D Regression Synthetic Data



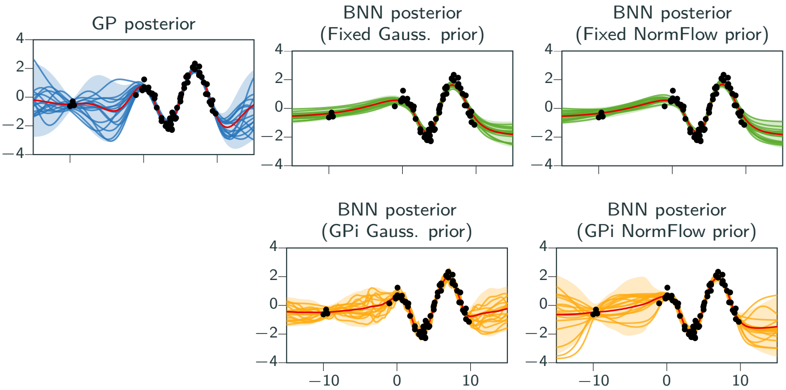
1D Regression Synthetic Data



1D Regression Synthetic Data



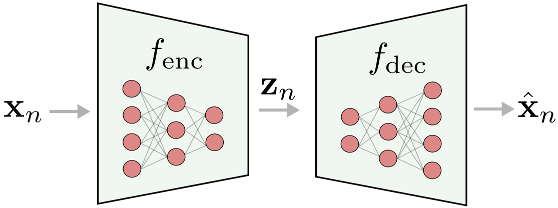
1D Regression Synthetic Data



Bayesian Convolutional Neural Networks - CIFAR-10

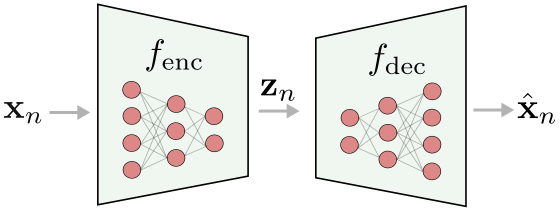
Architecture	Method	Accuracy - % (↑)	NLL (↓)
VGG16	Deep Ensemble	81.96 $\pm$ 0.33	0.7759 $\pm$ 0.0033
	Fixed Gauss. prior	81.47 $\pm$ 0.33	0.5808 $\pm$ 0.0033
	Fixed Gauss. prior + Temp. Scaling	82.25 $\pm$ 0.15	0.5398 $\pm$ 0.0015
	GPi Gauss. prior (**ours**)	83.34 $\pm$ 0.53	0.5176 $\pm$ 0.0053
	Fixed Hierar. prior	86.03 $\pm$ 0.20	0.4345 $\pm$ 0.0020
	GPi Hierar. prior (**ours**)	**87.03** $\pm$ **0.07**	**0.4127** $\pm$ **0.0007**
PRERESNET20	Deep Ensemble	87.77 $\pm$ 0.03	0.3927 $\pm$ 0.0003
	Fixed Gauss. prior	85.34 $\pm$ 0.13	0.4975 $\pm$ 0.0013
	Fixed Gauss. prior + Temp. Scaling	87.70 $\pm$ 0.11	0.3956 $\pm$ 0.0011
	GPi Gauss. prior (**ours**)	86.86 $\pm$ 0.27	0.4286 $\pm$ 0.0027
	Fixed Hierar. prior	87.26 $\pm$ 0.09	0.4086 $\pm$ 0.0009
	GPi Hierar. prior (**ours**)	**88.20** $\pm$ **0.07**	**0.3808** $\pm$ **0.0007**

Autoencoders



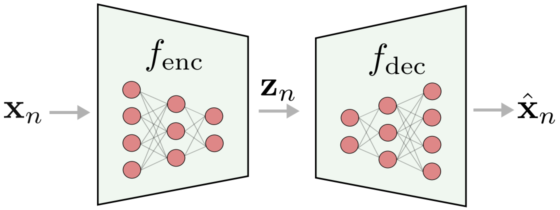
- An autoencoder (AE) is a neural network used for *unsupervised learning*

Autoencoders



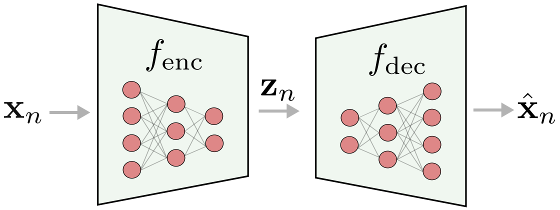
- An autoencoder (AE) is a neural network used for *unsupervised learning*
- *Encoder*: transforms an unlabelled dataset, $\mathbf{x} := \{\mathbf{x}_n\}_n^N$, into latent codes, $\mathbf{z} := \{\mathbf{z}_n\}_n^N$
- *Decoder*: transforms latent codes into reconstructions, $\hat{\mathbf{x}} := \{\hat{\mathbf{x}}_n\}_n^N$

Autoencoders



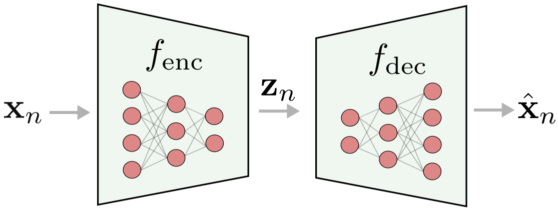
- An autoencoder (AE) is a neural network used for *unsupervised learning*
- *Encoder*: transforms an unlabelled dataset, $\mathbf{x} := \{\mathbf{x}_n\}_n^N$, into latent codes, $\mathbf{z} := \{\mathbf{z}_n\}_n^N$
- *Decoder*: transforms latent codes into reconstructions, $\hat{\mathbf{x}} := \{\hat{\mathbf{x}}_n\}_n^N$
- We can do Bayesian Autoencoders! [**Tran et al.**, **NeurIPS**, **2021**]

Bayesian Autoencoders



✓ Breaking away from Variational Autoencoders – separating modeling from inference

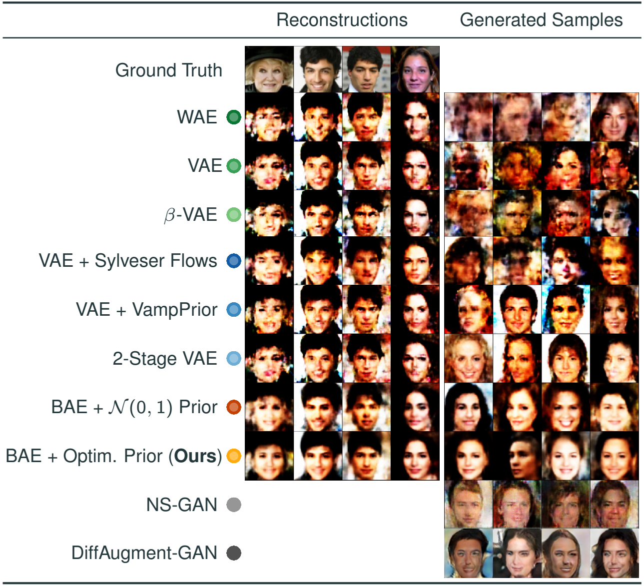
Bayesian Autoencoders

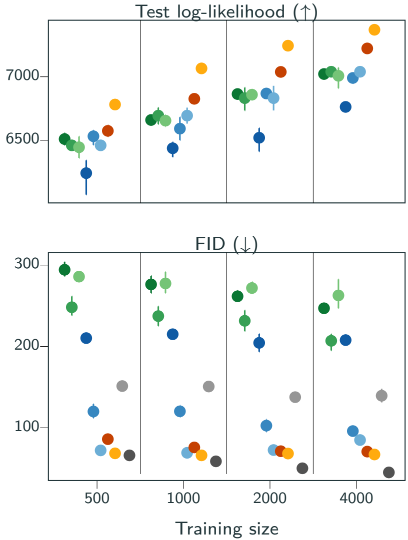


✓ Breaking away from Variational Autoencoders – separating modeling from inference

✗ Lack of generative modeling – Easy to bypass by modeling distribution of the latent codes

Experiments on CelebA Dataset



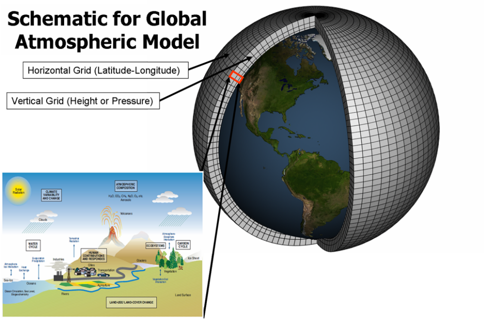


Ongoing Work

Ongoing Work

Bayesian Deep Learning and Physics

- Emulation
- Physics-based priors
- Tackling identifiability issues of Bayesian calibration

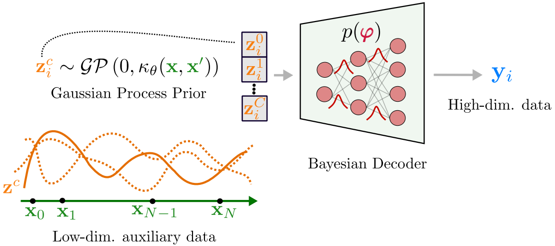


[Lorenzi and Filippone, ICML 2018 – Marmin and Filippone, Bayesian Analysis 2022]

Ongoing Work

Structured priors for Bayesian Autoencoders

- Beyond Score-based Diffusion Models
- Interpretability
- Causality



[**Tran et al., ICML 2023**]

Ongoing Work

Applications to problems and where decision-making matters

- Environment and Sustainability
- Life Sciences

Thank you!

Questions?