ML-Uncertainties and Bayesian Networks

Tilman Plehn

Universität Heidelberg

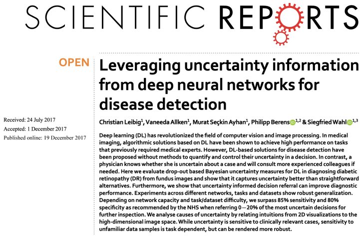
Berkeley Lab 8/2023

Neural networks and uncertainties

Neural networks

- nothing but numerically evaluated functions

  regression $x \to f(x)$
  classification $x \to p(x) \in [0,1]$
  generation $x \to p_X(x)$ with sampled $x \sim \mathcal{N}$
- constructed through minimization of loss function
- nothing like a Minut fit
- Error bars $x \to f(x) \pm \Delta f(x)$?



SCIENTIFIC REPORTS

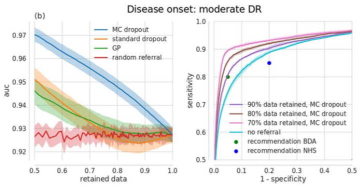
OPEN

Leveraging uncertainty information from deep neural networks for disease detection

Received: 24 July 2017
Accepted: 1 December 2017
Published online: 19 December 2017

Christian Leibig[1], Vaneeda Allken[1], Murat Seçkin Ayhan[1], Philipp Berens[1,2] & Siegfried Wahl[1,3]

Deep learning (DL) has revolutionized the field of computer vision and image processing. In medical imaging, algorithmic solutions based on DL have been shown to achieve high performance on tasks that previously required medical experts. However, DL-based solutions for disease detection have been proposed without methods to quantify and control their uncertainty in a decision. In contrast, a physician knows whether she is uncertain about a case and will consult more experienced colleagues if needed. Here we evaluate drop-out based Bayesian uncertainty measures for DL in diagnosing diabetic retinopathy (DR) from fundus images and show that it captures uncertainty better than straightforward alternatives. Furthermore, we show that uncertainty informed decision referral can improve diagnostic performance. Experiments across different networks, tasks and datasets show robust generalization. Depending on network capacity and task/dataset difficulty, we surpass 85% sensitivity and 80% specificity as recommended by the NHS when referring 0—20% of the most uncertain decisions for further inspection. We analyse causes of uncertainty by relating intuitions from 2D visualizations to the high-dimensional image space. While uncertainty is sensitive to clinically relevant cases, sensitivity to unfamiliar data samples is task dependent, but can be rendered more robust.



Neural networks and uncertainties

Neural networks

- nothing but numerically evaluated functions

  regression $x \to f(x)$
  classification $x \to p(x) \in [0, 1]$
  generation $x \to p_X(x)$ with sampled $x \sim \mathcal{N}$
- constructed through minimization of loss function
- nothing like a Minut fit
- Error bars $x \to f(x) \pm \Delta f(x)$?

NN with uncertainties

- regression: p_T of jet from constituents, error bar?
  classification: probability of Higgs event, error bar?
  generation: phase space density for large $\not{p}_T$, error bar?
- standard LHC approach

  train black box on Monte Carlo
  calibrate with reference data

$\to$ Try to do better?

A tale of four theses

David MacKay (1991)

- Bayesian methods [posterior=likelihood*prior/evidence]

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)}$$

- Bayesian networks for inference
  data modelling through parameters w

$$P(w|D,M) = \frac{P(D|w,M)P(w|M)}{P(D|M)}$$

- Occam factor for model evidence [posterior/prior volume]
- technically: Gaussian weight distributions?

for Adaptive Models

Thesis by

David J.C. MacKay

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

California Institute of Technology
Pasadena, California

©1992
(Submitted December 10, 1991)

Since the 1960's, the Bayesian minority has been steadily growing, especially in the fields of economics [89] and pattern processing [20]. At this time, the state of the art for the problem of speech recognition is a Bayesian technique (Hidden Markov Models), and the best image reconstruction algorithms are also based on Bayesian probability theory (Maximum Entropy), but Bayesian methods are still viewed with mistrust by the orthodox statistics community; the framework for model comparison is especially poorly known, even to most people who call themselves Bayesians. This thesis therefore takes some time to thoroughly review the flavour of Bayesianism that I am using. To some, the word Bayesian denotes

A tale of four theses

David MacKay (1991)

- Bayesian methods [posterior=likelihood*prior/evidence]

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)}$$

- Bayesian networks for inference
  data modelling through parameters w

$$P(w|D,M) = \frac{P(D|w,M)P(w|M)}{P(D|M)}$$

- technically: Gaussian weight distributions?

Chapter 3

A Practical Bayesian Framework for Backpropagation Networks

Abstract

A quantitative and practical Bayesian framework is described for learning of mappings in feedforward networks. The framework makes possible: (1) objective comparisons between solutions using alternative network architectures; (2) objective stopping rules for network pruning or growing procedures; (3) objective choice of magnitude and type of weight decay terms or additive regularisers (for penalising large weights, etc.); (4) a measure of the effective number of well–determined parameters in a model; (5) quantified estimates of the error bars on network parameters and on network output; (6) objective comparisons with alternative learning and interpolation models such as splines and radial basis functions. The Bayesian 'evidence' automatically embodies 'Occam's razor', penalising over–flexible and over–complex models. The Bayesian approach helps detect poor underlying assumptions in learning models. For learning models well matched to a problem, a good correlation between generalisation ability and the Bayesian evidence is obtained.

Thesis by

David J.C. MacKay

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

California Institute of Technology
Pasadena, California

©1992
(Submitted December 10, 1991)

A tale of four theses

David MacKay (1991)

- Bayesian methods [posterior=likelihood*prior/evidence]

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)}$$

- Bayesian networks for inference
  data modelling through parameters w

$$P(w|D,M) = \frac{P(D|w,M)P(w|M)}{P(D|M)}$$

- technically: Gaussian weight distributions?

Radford Neal (1995)

- deep Bayesian networks [regression, classification]
- beyond Gaussian approximation
- hybrid Monte Carlo sampling
- technically: avoid overtraining for large BNNs

→ Deep BNNs for inference

BAYESIAN LEARNING FOR NEURAL NETWORKS

by

Radford M. Neal

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy,
Graduate Department of Computer Science,
in the University of Toronto

© Copyright 1995 by Radford M. Neal

A tale of four theses

Yarin Gal (2016)

- deep learning and uncertainties
- active learning/reinforcement learning
- technically: variational inference
- technically: stochastic regularization

$\rightarrow$ BNNs for uncertainty

UNIVERSITY OF CAMBRIDGE

Uncertainty in Deep Learning

Yarin Gal

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Gonville and Caius College — September 2016

Other situations that can lead to uncertainty include

- noisy data (our observed labels might be noisy, for example as a result of measurement imprecision, leading to *aleatoric uncertainty*),
- *uncertainty in model parameters* that best explain the observed data (a large number of possible models might be able to explain a given dataset, in which case we might be uncertain which model parameters to choose to predict with),
- and *structure uncertainty* (what model structure should we use? how do we specify our model to extrapolate / interpolate well?).

The latter two uncertainties can be grouped under *model uncertainty* (also referred to as *epistemic uncertainty*). Aleatoric uncertainty and epistemic uncertainty can then be used to induce *predictive uncertainty*, the confidence we have in a prediction.

A tale of four theses

Yarin Gal (2016)

- deep learning and uncertainties
- active learning/reinforcement learning
- technically: variational inference
- technically: stochastic regularization

→ BNNs for uncertainty

UNIVERSITY OF CAMBRIDGE

Uncertainty in Deep Learning

Yarin Gal

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Gonville and Caius College — September 2016

> But fitting the posterior over the weights of a Bayesian NN with a unimodal approximating distribution does not mean the predictive distribution would be unimodal! imagine for simplicity that the intermediate feature output from the first layer is a unimodal distribution (a uniform for example) and let's say, for the sake of argument, that the layers following that are modelled with delta distributions (or Gaussians with very small variances). Given enough follow-up layers we can capture any function to arbitrary precision—including the inverse cumulative distribution function (CDF) of any multimodal distribution. Passing our uniform output from the first layer through the rest of the layers—in effect transforming the uniform with this inverse CDF—would give a multimodal predictive distribution.

A tale of four theses

Yarin Gal (2016)

- deep learning and uncertainties
- active learning/reinforcement learning
- technically: variational inference
- technically: stochastic regularization

$\rightarrow$ BNNs for uncertainty

Manuel Haußmann (2021)

- many proper derivations
- active learning, reinforcement learning
- stochastic differential equations
- technically: BNN variational inference

INAUGURAL – DISSERTATION

zur

Erlangung der Doktorwürde

der

Naturwissenschaftlich-Mathematischen Gesamtfakultät

der

RUPRECHT-KARLS-UNIVERSITÄT
HEIDELBERG

vorgelegt von

Manuel Haußmann, M.Sc.

geboren in Stuttgart, Deutschland

Jet regression

Jet properties with uncertainties

- train many networks
  different architectures/hyperparameters
  different trainings
  different initalizations
  different data sets
- histogram network output $f(x)$, use $f(x) \pm \Delta f(x)$
- remember NN function $f_\theta(x)$ described by weights θ

$\to$ Bayesian network $\quad \Delta f_\theta(x)$ from $\Delta \theta_j$

Energy measurement for jet j

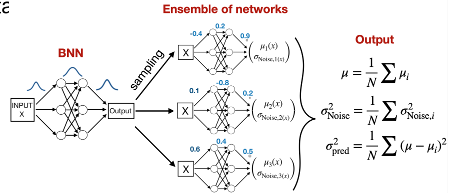
- expectation value from probability distribution

$$\langle E \rangle = \int dE \; E \; p(E)$$

- weighted by reproduced training data

$$p(E) = \int d\theta \; p(E|\theta) \; p(\theta|T)$$

$\to$ θ-distributions means BNN



Likelihood loss

Replacing the MSE

- start from variational approximation [think $q(\theta)$ as Gaussian with mean and width]

$$p(E) = \int d\theta \; p(E|\theta) \; p(\theta|T) \approx \int d\theta \; p(E|\theta) \; q(\theta)$$

- similarity through minimal KL-divergence [Bayes' theorem to remove unknown posterior]

$$\begin{aligned} D_\text{KL}[q(\theta), p(\theta|T)] &= \int d\theta \; q(\theta) \; \log \frac{q(\theta)}{p(\theta|T)} \\ &= \int d\theta \; q(\theta) \; \log \frac{q(\theta) p(T)}{p(T|\theta) p(\theta)} \\ &= D_\text{KL}[q(\theta), p(\theta)] - \int d\theta \; q(\theta) \; \log p(T|\theta) + \log p(T) \int d\theta \; q(\theta) \\ &= D_\text{KL}[q(\theta), p(\theta)] - \int d\theta \; q(\theta) \; \log p(T|\theta) + \log p(T) \end{aligned}$$

- well-defined evidence lower bound (ELBO)

$$\begin{aligned} \log p(T) &= D_\text{KL}[q(\theta), p(\theta|T)] - D_\text{KL}[q(\theta), p(\theta)] + \int d\theta \; q(\theta) \; \log p(T|\theta) \\ &\geq \int d\theta \; q(\theta) \; \log p(T|\theta) - D_\text{KL}[q(\theta), p(\theta)] \end{aligned}$$

$\to$ loss with likelihood $p(T|\theta)$ and prior $p(\theta)$

$$\mathcal{L} = - \int d\theta \; q(\theta) \; \log p(T|\theta) + D_\text{KL}[q(\theta), p(\theta)]$$

Relation to standard networks

Regularization and dropout

- Gaussian prior

$$D_{\mathrm{KL}}[q_{\mu,\sigma}(\theta), p_{\mu,\sigma}(\theta)] = \frac{\sigma_q^2 - \sigma_p^2 + (\mu_q - \mu_p)^2}{2\sigma_p^2} + \log \frac{\sigma_p}{\sigma_q}$$

- deterministic network $q(\theta) \to \delta(\theta - \theta_0)$

$$\mathcal{L} \approx - \log p(T|\theta_0) + \frac{(\mu_p - \theta_0)^2}{2\sigma_p^2} + \text{const}$$

standard network with fixed L2-regularization

$\to$ deterministic counterpart

- Monte-Carlo dropout

meant to reduce overfitting
remove random weights during training

loss with Bernoulli distribution [weight $x\theta_0 = 0, \theta_0$]

$$\mathcal{L} = - \int dx \, \left[\rho^x (1-\rho)^{1-x} \right]_{x=0,1} \log p(T|x\theta_0) \approx -\rho \, \log p(T|\theta_0)$$

$\to$ trivial version of variational training

Weight sampling

Weight space

- expectation value using trained network $q(\theta)$

$$\langle E \rangle = \int dE d\theta \; E \; p(E|\theta) \; q(\theta)$$

$$\equiv \int d\theta \; q(\theta) \overline{E}(\theta) \qquad \text{with} \qquad \overline{E}(\theta) = \int dE \; E \; p(E|\theta)$$

- output variance

$$\begin{aligned} \sigma_\text{tot}^2 &= \int dE d\theta \; (E - \langle E \rangle)^2 \; p(E|\theta) \; q(\theta) \\ &= \int d\theta \; q(\theta) \left[\overline{E^2}(\theta) - 2 \langle E \rangle \overline{E}(\theta) + \langle E \rangle^2 \right] \\ &= \int d\theta \; q(\theta) \left[\overline{E^2}(\theta) - \overline{E}(\theta)^2 + \left(\overline{E}(\theta) - \langle E \rangle \right)^2 \right] \equiv \sigma_\text{stoch}^2 + \sigma_\text{pred}^2 \end{aligned}$$

Two uncertainties

- contribution vanishing for $q(\theta) \to \delta(\theta - \theta_0)$

$$\sigma_\text{pred}^2 = \int d\theta \; q(\theta) \left[\overline{E}(\theta) - \langle E \rangle \right]^2$$

- contribution in weight space

$$\sigma_\text{stoch}^2 \equiv \sigma_\text{model}^2 = \int d\theta \; q(\theta) \left[\overline{E^2}(\theta) - \overline{E}(\theta)^2 \right] = \int d\theta \; q(\theta) \; \sigma_\text{stoch}(\theta)^2$$

Implementation

Approximations and implementation

- network output in weight and phase space

$$\text{BNN}: x, \theta \to \begin{pmatrix} \overline{E}(\theta) \\ \sigma_\text{stoch}(\theta) \end{pmatrix}$$

- Gaussian weights & likelihood

$$L = \int d\theta \; q_{\mu,\sigma}(\theta) \sum_{\text{jets } j} \left[\frac{\left| \overline{E}_j(\theta) - E_j^\text{truth} \right|^2}{2\sigma_{\text{stoch},j}(\theta)^2} + \log \sigma_{\text{stoch},j}(\theta) \right]$$
$$+ \frac{\sigma_q^2 - \sigma_p^2 + (\mu_q - \mu_p)^2}{2\sigma_p^2} + \log \frac{\sigma_p}{\sigma_q}$$

- heteroskedastic loss, deterministic network

$$L = \sum_{\text{jets } j} \left[\frac{\left| \overline{E}_j(\theta_0) - E_j^\text{truth} \right|^2}{2\sigma_{\text{stoch},j}(\theta_0)^2} + \log \sigma_{\text{stoch},j}(\theta_0) \right]$$

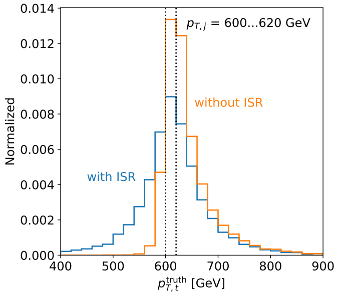
- supervised uncertainties

  training statistics
  stochastic training data
  systematics from data
  label augmentations
  model limitations

Jet measurements with error bars

Measure $p_{T,t}$ of hadronically decaying top [Kasieczka, Luchmann, Otterpohl, TP]

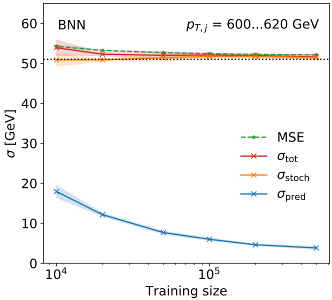
- BNN regression $p_{T,t}$
  p_T of (fat) jet decent estimate for $p_{T,t}^{\text{truth}}$
- non-Gaussian truth label
  symmetric in ISR-jet 'QCD heat bath'
  without ISR jets need for correction

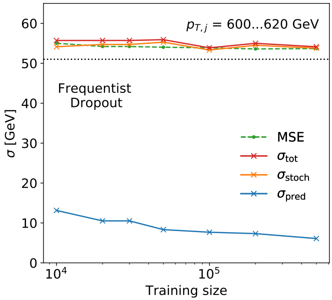


Jet measurements with error bars

Measure $p_{T,t}$ of hadronically decaying top [Kasieczka, Luchmann, Otterpohl, TP]

- BNN regression $p_{T,t}$
  p_T of (fat) jet decent estimate for $p_{T,t}^{\text{truth}}$
- non-Gaussian truth label
  symmetric in ISR-jet 'QCD heat bath'
  without ISR jets need for correction
- training sample size
  separate $\sigma_{\text{stoch}} \gg \sigma_{\text{pred}}$
  statistics not the problem [LHC theme]
  noisy label inherent limitation
  checked with deterministic networks

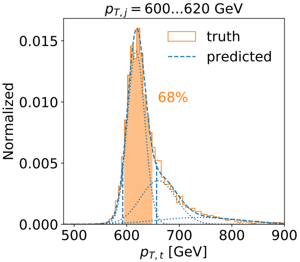




Jet measurements with error bars

Measure $p_{T,t}$ of hadronically decaying top [Kasieczka, Luchmann, Otterpohl, TP]

- BNN regression $p_{T,t}$
  p_T of (fat) jet decent estimate for $p_{T,t}^{\text{truth}}$
- non-Gaussian truth label
  symmetric in ISR-jet 'QCD heat bath'
  without ISR jets need for correction
- training sample size
  separate $\sigma_{\text{stoch}} \gg \sigma_{\text{pred}}$
  statistics not the problem [LHC theme]
  noisy label inherent limitation
  checked with deterministic networks
- non-Gaussian network output
  remember $p_{T,t}^{\text{truth}}$ non-Gaussian
  model $p(T|\theta)$ as Gaussian mixture
  weight distribution $q(\theta)$ still Gaussian



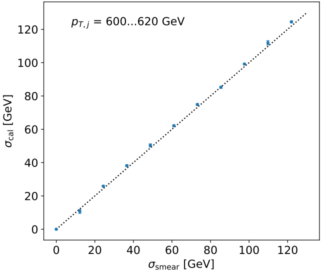
Data augmentation

Calibration means error propagation

- calibration means label measured elsewhere
- training on smeared data?
  training with smeared labels!
- Gaussian noise over label
- added to the stochastic uncertainty

$$\begin{aligned}\sigma_{\text{tot}}^2 &= \sigma_{\text{stoch}}^2 + \sigma_{\text{pred}}^2 \\ &= \sigma_{\text{stoch},0}^2 + \sigma_{\text{cal}}^2 + \sigma_{\text{pred}}^2\end{aligned}$$

$\rightarrow$ error extracted correctly



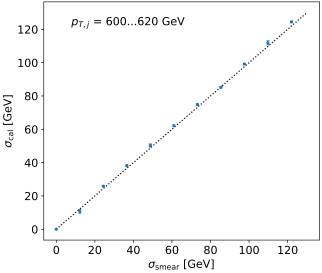
Data augmentation

Calibration means error propagation

- calibration means label measured elsewhere
- training on smeared data?
  training with smeared labels!
- Gaussian noise over label
- added to the stochastic uncertainty

$$\sigma^2_{\text{tot}} = \sigma^2_{\text{stoch}} + \sigma^2_{\text{pred}}$$
$$= \sigma^2_{\text{stoch},0} + \sigma^2_{\text{cal}} + \sigma^2_{\text{pred}}$$

$\rightarrow$ error extracted correctly



Jet regression bottom lines

- BNN regressionion working
- statistical uncertainty controlled
- stochastic uncertainty sizeable
- non-Gaussian output working
- training-data augmentation
- calibration straighforward

Precision amplitudes

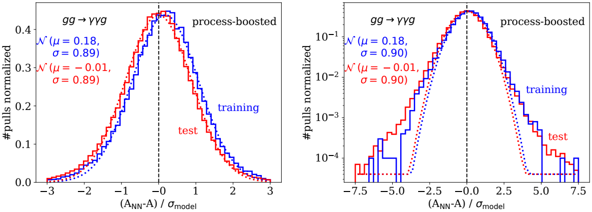
Loop amplitudes $gg \to \gamma\gamma g(g)$ [Badger, Butter, Luchmann, Pitz, TP]

- amplitudes A over phase space points x_j — simple regression
- weight-dependent pull

$$\frac{\overline{A}_j(\theta) - A_j^\text{truth}}{\sigma_{\text{model},j}(\theta)}$$

- training data exact in x and A
- improvement $\to$ interpolation by weighting [by pull or σ]

$$L = \int d\theta \; q_{\mu,\sigma}(\theta) \sum_{\text{points } j} n_j \times \left[\frac{\left| \overline{A}_j(\theta) - A_j^\text{truth} \right|^2}{2\sigma_{\text{model},j}(\theta)^2} + \log \sigma_{\text{model},j}(\theta) \right] \cdots$$



Precision amplitudes

Loop amplitudes $gg \to \gamma\gamma g(g)$ [Badger, Butter, Luchmann, Pitz, TP]

- amplitudes A over phase space points x_j — simple regression
- weight-dependent pull

$$\frac{\overline{A}_j(\theta) - A_j^{\text{truth}}}{\sigma_{\text{model},j}(\theta)}$$

- training data exact in x and A
- improvement $\to$ interpolation by weighting [by pull or σ]

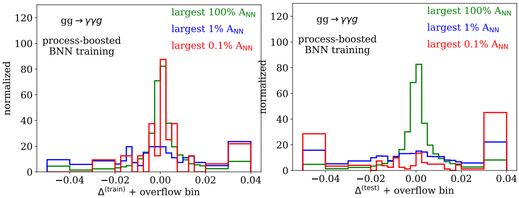
$$L = \int d\theta \; q_{\mu,\sigma}(\theta) \sum_{\text{points } j} n_j \times \left[\frac{\left| \overline{A}_j(\theta) - A_j^{\text{truth}} \right|^2}{2\sigma_{\text{model},j}(\theta)^2} + \log \sigma_{\text{model},j}(\theta) \right] \cdots$$

Precision regression

- quality of network amplitudes

$$\Delta_j^{(\text{train/test})} = \frac{\langle A \rangle_j - A_j^{\text{train/test}}}{A_j^{\text{train/test}}}$$

$\to$ Beyond fit-like regression



Precision amplitudes

Loop amplitudes $gg \to \gamma\gamma g(g)$ [Badger, Butter, Luchmann, Pitz, TP]

- amplitudes A over phase space points x_j — simple regression
- weight-dependent pull

$$\frac{\overline{A}_j(\theta) - A_j^{\text{truth}}}{\sigma_{\text{model},j}(\theta)}$$

- training data exact in x and A
- improvement $\to$ interpolation by weighting [by pull or σ]

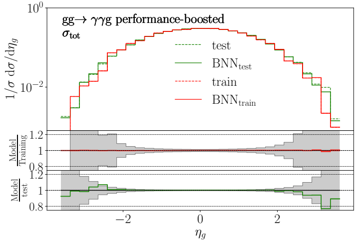
$$L = \int d\theta \; q_{\mu,\sigma}(\theta) \sum_{\text{points } j} n_j \times \left[\frac{\left| \overline{A}_j(\theta) - A_j^{\text{truth}} \right|^2}{2\sigma_{\text{model},j}(\theta)^2} + \log \sigma_{\text{model},j}(\theta) \right] \cdots$$

Precision regression

- quality of network amplitudes

$$\Delta_j^{(\text{train/test})} = \frac{\langle A \rangle_j - A_j^{\text{train/test}}}{A_j^{\text{train/test}}}$$

$\to$ Beyond fit-like regression



Generative networks

Unsupervised Bayesian networks [Bellagente, Haußmann, Luchmann, TP]

- data: event sample [points in 2D space]

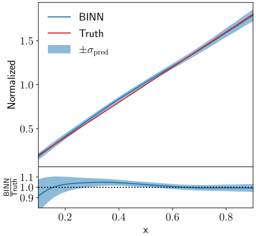
  learn phase space density
  normalizing flow mapping to latent space [INN]
  standard distribution in latent space [Gaussian]
  mapping bijective
  sample from latent space
- Bayesian version

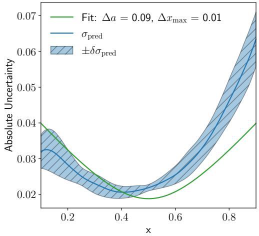
  allow weight distributions
  learn uncertainty map
- 2D wedge ramp

$$p(x) = ax + b = ax + \frac{1 - \frac{a}{2}(x_{\text{max}}^2 - x_{\text{min}}^2)}{x_{\text{max}} - x_{\text{min}}}$$

$$(\Delta p)^2 = \left(x - \frac{1}{2}\right)^2 (\Delta a)^2 + \left(1 + \frac{a}{2}\right)^2 (\Delta x_{\text{max}})^2 + \left(1 - \frac{a}{2}\right)^2 (\Delta x_{\text{min}})^2$$

  explaining minimum in $\sigma_{\text{pred}}(x)$





$\rightarrow$ INNs just (non-parametric) fits

Precision generator

Phase-space generators [typical LHC task]

- training from event samples
  no energy-momentum conservation
- every correlation counts
- $Z_{\mu\mu} + \{1, 2, 3\}$ jets [Z-peak, variable jet number, jet-jet topology]

Precision generator

Phase-space generators [typical LHC task]

- training from event samples
  no energy-momentum conservation
- every correlation counts
- $Z_{\mu\mu} + \{1, 2, 3\}$ jets [Z-peak, variable jet number, jet-jet topology]

INN-generator

- stable bijective mapping

$$\text{latent } r \sim p_\text{latent} \quad \underset{\leftarrow\ \overline{G}_\theta(x)}{\overset{G_\theta(r) \rightarrow}{\longleftrightarrow}} \quad \text{phase space } x \sim p_\text{data}$$

- tractable Jacobian

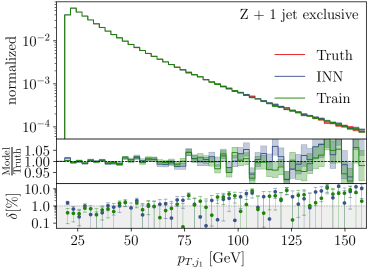
$$dx\ p_\text{model}(x) = dr\ p_\text{latent}(r)$$

$$p_\text{model}(x) = p_\text{latent}\left(\overline{G}_\theta(x)\right) \left| \frac{\partial \overline{G}_\theta(x)}{\partial x} \right|$$

- likelihood loss

$$\mathcal{L}_\text{INN} = -\Big\langle \log p_\text{model}(x) \Big\rangle_{p_\text{data}}$$

$\Rightarrow$ Per-cent precision possible



Controlled precision generator

Best of GANs: discriminator

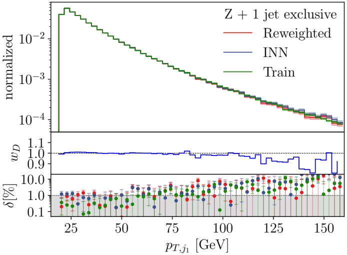
- $D = 0$ (generator) vs $D = 1$ (training)
- NP-optimal discriminator

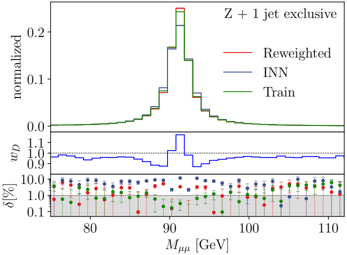
$$D(x) \to \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_{\text{model}}(x)} \to \frac{1}{2}$$

- learned event weight

$$w(x) \to \frac{D(x)}{1 - D(x)} = \frac{p_{\text{data}}(x)}{p_{\text{model}}(x)} \to 1$$

$\Rightarrow$ Dual purpose: control and reweight





Controlled precision generator

Best of GANs: discriminator

- $D = 0$ (generator) vs $D = 1$ (training)
- NP-optimal discriminator

$$D(x) \to \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_{\text{model}}(x)} \to \frac{1}{2}$$

- learned event weight $$w(x) \to \frac{D(x)}{1 - D(x)} = \frac{p_{\text{data}}(x)}{p_{\text{model}}(x)} \to 1$$

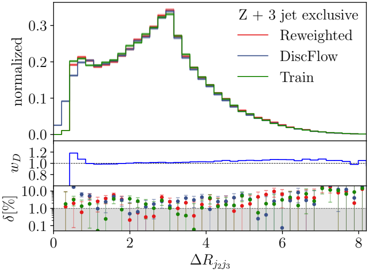
$\Rightarrow$ Dual purpose: control and reweight

Joint training [GAN inspiration]

- GAN-like training unstable [Nash equilibrium??]
- coupling through weights

$$\mathcal{L} = - \int dx \, \frac{p_{\text{data}}^{\alpha+1}(x)}{p_{\text{model}}^{\alpha}(x)} \, \log \frac{p_{\text{model}}(x)}{p_{\text{data}}(x)}$$

$\Rightarrow$ Unweighted, controlled events

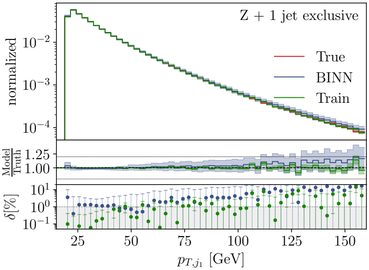


Precision generator with uncertainties

Bayesian network generator

- network with weight distributions [Gal (2016)]
  sample weights [defining error bar]
  working for regression, classification
  frequentist: efficient ensembling

$\Rightarrow$ Training-related error bars



Precision generator with uncertainties

Bayesian network generator

- network with weight distributions [Gal (2016)]
  sample weights [defining error bar]
  working for regression, classification
  frequentist: efficient ensembling

$\Rightarrow$ Training-related error bars

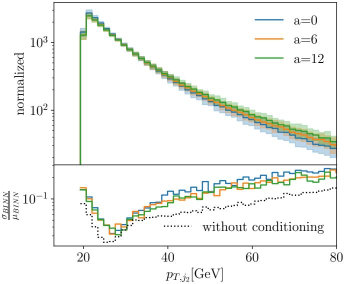
Theory uncertainties

- BNN regression/classification:
  systematics from data augmentation
- systematic uncertainties in tails

$$w = 1 + a \left(\frac{p_{T,j_1} - 15 \;\; \text{GeV}}{100 \;\; \text{GeV}} \right)^2$$

- augment training data [$a = 0 \ldots 30$]
- train conditionally on a
  error bar from sampling a

$\Rightarrow$ Systematic/theory error bars



Precision generator with uncertainties

Bayesian network generator

- network with weight distributions [Gal (2016)]
  sample weights [defining error bar]
  working for regression, classification
  frequentist: efficient ensembling

$\Rightarrow$ Training-related error bars

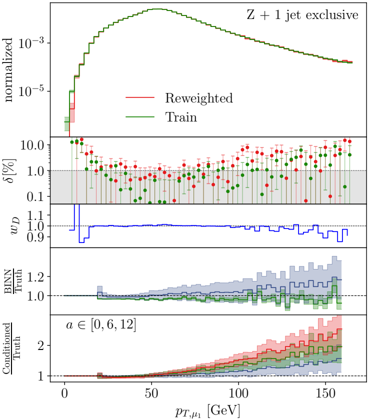
Theory uncertainties

- BNN regression/classification:
  systematics from data augmentation
- systematic uncertainties in tails

$$w = 1 + a \left(\frac{p_{T,j_1} - 15 \;\; \text{GeV}}{100 \;\; \text{GeV}} \right)^2$$

- augment training data [$a = 0 \dots 30$]
- train conditionally on a
  error bar from sampling a

$\Rightarrow$ Systematic/theory error bars



Testing generative networks

Compare network to training/test data

- supervised: histogram deviation [or pull]
- unsupervised density → histogram discriminator

$$w(x_i) = \frac{D(x_i)}{1 - D(x_i)} = \frac{p_{\text{data}}(x_i)}{p_{\text{model}}(x_i)}$$

→ Using interpretable phase space

Testing generative networks

Compare network to training/test data

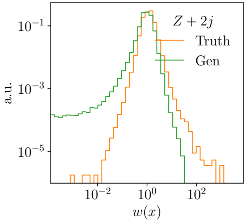
- supervised: histogram deviation [or pull]
- unsupervised density → histogram discriminator

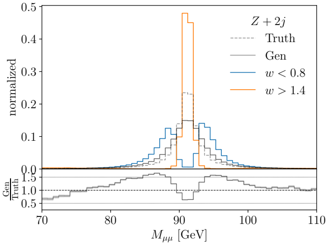
$$w(x_i) = \frac{D(x_i)}{1 - D(x_i)} = \frac{p_{\text{data}}(x_i)}{p_{\text{model}}(x_i)}$$

→ Using interpretable phase space

Applied to event generators [also jets, calorimeter showers]

- shape and width of w-histogram
- pattern in (interpretable) phase space?





Testing generative networks

Compare network to training/test data

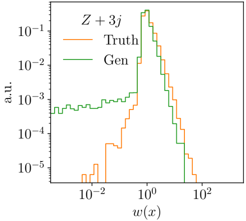
- supervised: histogram deviation [or pull]
- unsupervised density → histogram discriminator

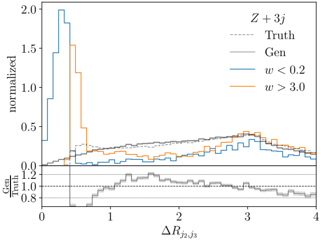
$$w(x_i) = \frac{D(x_i)}{1 - D(x_i)} = \frac{p_{\text{data}}(x_i)}{p_{\text{model}}(x_i)}$$

→ Using interpretable phase space

Applied to event generators [also jets, calorimeter showers]

- shape and width of w-histogram
- pattern in (interpretable) phase space?





→ Generative xAI for LHC physicists

Bayesian networks

Initially developed for inference they work for...

- ...regression with error bars
- ...classification with error bars
- ...generation with error bars

Modern Machine Learning for LHC Physicists

Tilman Plehn[a,*], Anja Butter[a,b], Barry Dillon[a], Claudius Krause[a,c], and Ramon Winterhalder[d]

[a] Institut für Theoretische Physik, Universität Heidelberg, Germany
[b] LPNHE, Sorbonne Université, Université Paris Cité, CNRS/IN2P3, Paris, France
[c] NHETC, Dept. of Physics and Astronomy, Rutgers University, Piscataway, USA
[d] CP3, Université Catholique de Louvain, Louvain-la-Neuve, Belgium

July 21, 2023

Abstract

Modern machine learning is transforming particle physics, faster than we can follow, and bullying its way into our numerical tool box. For young researchers it is crucial to stay on top of this development, which means applying cutting-edge methods and tools to the full range of LHC physics problems. These lecture notes are meant to lead students with basic knowledge of particle physics and significant enthusiasm for machine learning to relevant applications as fast as possible. They start with an LHC-specific motivation and a non-standard introduction to neural networks and then cover classification, unsupervised classification, generative networks, and inverse problems. Two themes defining much of the discussion are well-defined loss functions reflecting the problem at hand and uncertainty-aware networks. As part of the applications, the notes include some aspects of theoretical LHC physics. All examples are chosen from particle physics publications of the last few years. Given that these notes will be outdated already at the time of submission, the week of ML4Jets 2022, they will be updated frequently.