Bayesian Neural Networks - Uncertainty Quantification

- Rémi Emonet – Hubert Curien Laboratory
- Deep learning for medical imaging school
- 2021-04-21
- sources, updates and code available on github

Bayesian Neural Networks - Uncertainty Quantification (Overview)

1. Why we need to quantify uncertainty?
2. Some sources of uncertainty
3. Statistical machine learning approaches for general uncertainty modeling
4. Deep Learning practices for uncertainty modeling
5. Bayesian Neural Networks
    1. Bayesian view of machine learning
    2. Variational inference
    3. Variational Dropout
6. Applications and Openings

Why we need uncertainty quantification?

- Automated systems raise questions from experts
  - Can I trust the predictions?
  - Is the system confident in its prediction?
  - How was the decision taken?

- Good decision making is based on some assessment of uncertainties
  - Medical diagnosis
  - Asymmetric costs situations
  - Benefit/risk evaluation
  - Multi-factorial decisions
  - Self driving cars
  - …

What is uncertainty?

> *Uncertainty refers to epistemic situations involving* ***imperfect or unknown information****.*
>
> *It applies to predictions of* ***future events****, to* ***physical measurements*** *that are already made, or to the* ***unknown****.*
> *Uncertainty arises in partially observable and/or stochastic environments, as well as due to* ***ignorance, indolence, or both****.*
>
> *It arises in any number of fields, including insurance, philosophy, physics,* ***statistics****, economics, finance, psychology, sociology,* ***engineering****, metrology, meteorology, ecology and information science.* **wikipedia**

Sources of Uncertainty in Models

- Traditional ideal (deterministic) models, like rules in physics
  - e.g., $x_{t+1} = f(x_t)$ (e.g., dynamics, f includes gravity etc)
  - e.g., $Y = f(X)$ (e.g., ideal gas law, $PV = nRT$)
- Sources of uncertainty around statistical models?
  - Examples
    - stochastic (non-deterministic) environment
    - "sensors": partial observations, noisy observations
    - limited data (e.g., estimate the fairness of a dice from only 2 throws)
    - modeling: incomplete/partial/imprecise (or even wrong) model
      - imprecision in the model (e.g., value of π in $\mathcal{A} = \pi r^2$)
    - more...
  - Usual categories
    - Aleatoric (statistical) uncertainty... from things we cannot know
    - Epistemic (systematic) uncertainty... from things we could know

Uncertainty Types: a simplified view

- Aleatoric (statistical) uncertainty
  - "true" inherent randomness
  - w.r.t. our observables, inputs and "ground truth" outputs
  - no amount of data can remove this uncertainty
- Epistemic (systematic) uncertainty
  - not enough "training" data
  - wrong/simplified modeling assumptions

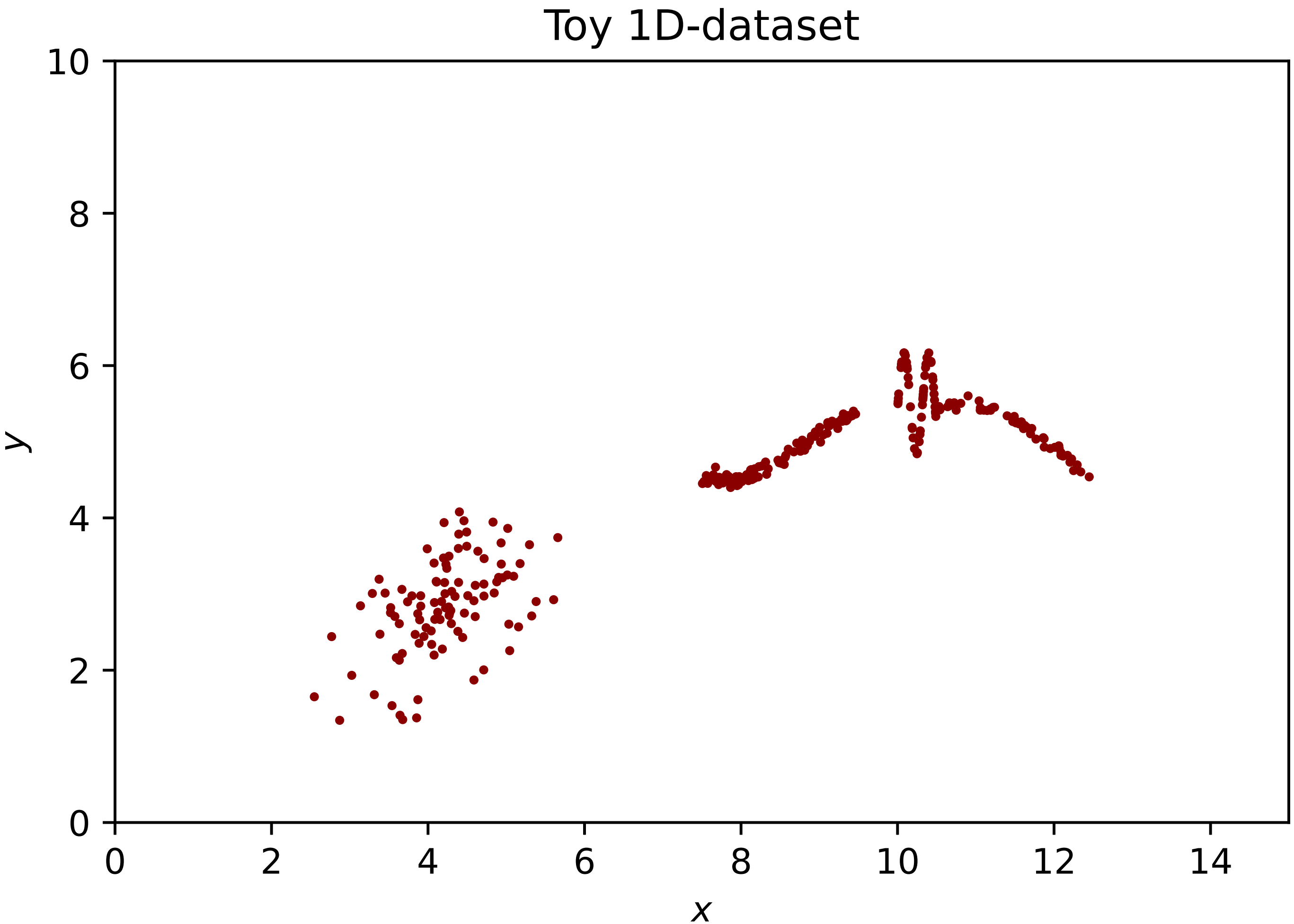
Today's challenge

Learn a model under **aleatoric** and **epsitemic** uncertainty

- NB: a task we won't cover in this talk, Uncertainty Propagation
  - given an actual model (manually specified / already learned)
  - propagate the uncertainty from a given input to the output prediction

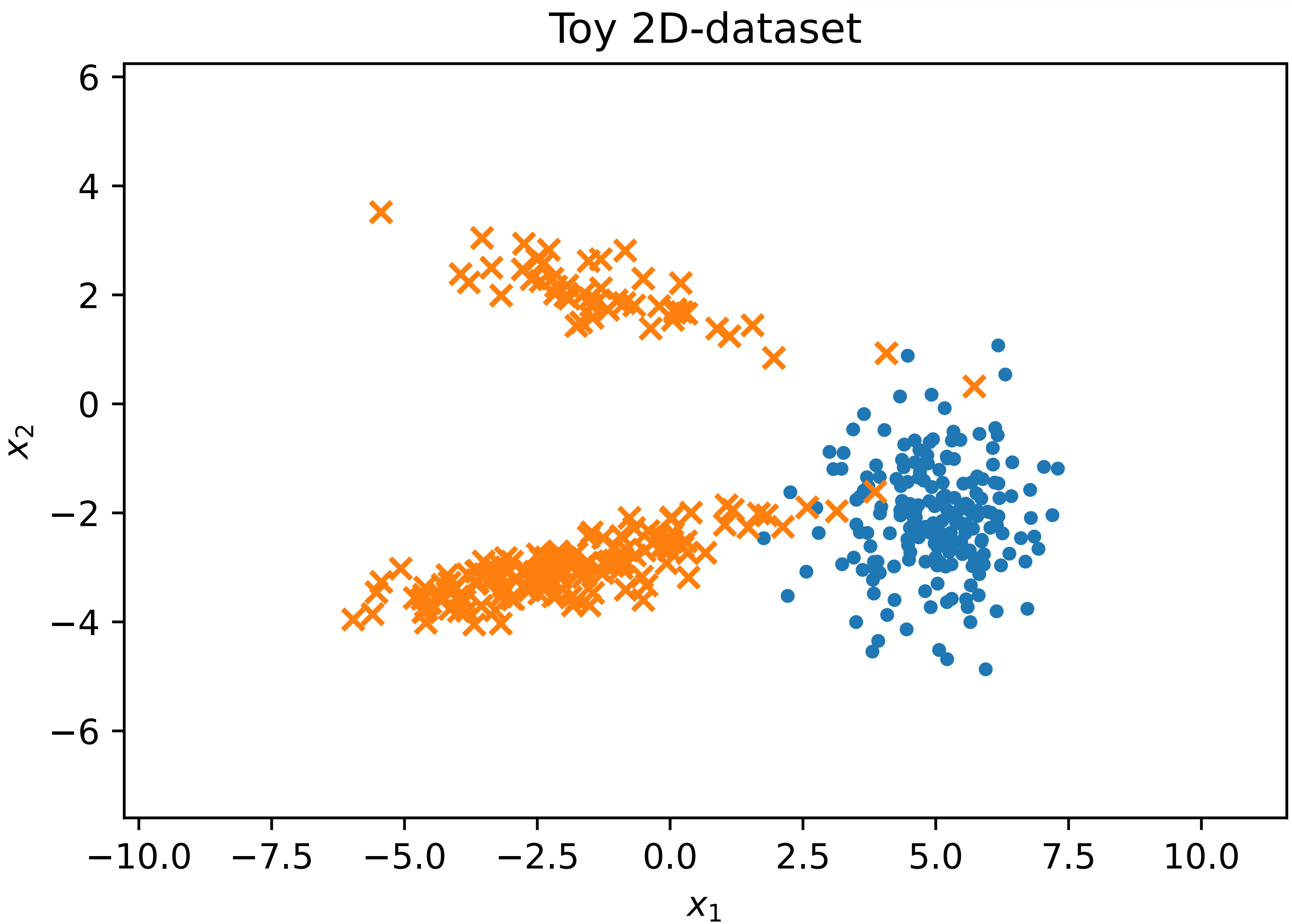
Uncertainty in Machine Learning: regression (1D "input", single "output")

-Aleatoric/statistical = "true" inherent randomness
-Epistemic/systematic = missing data, bad model



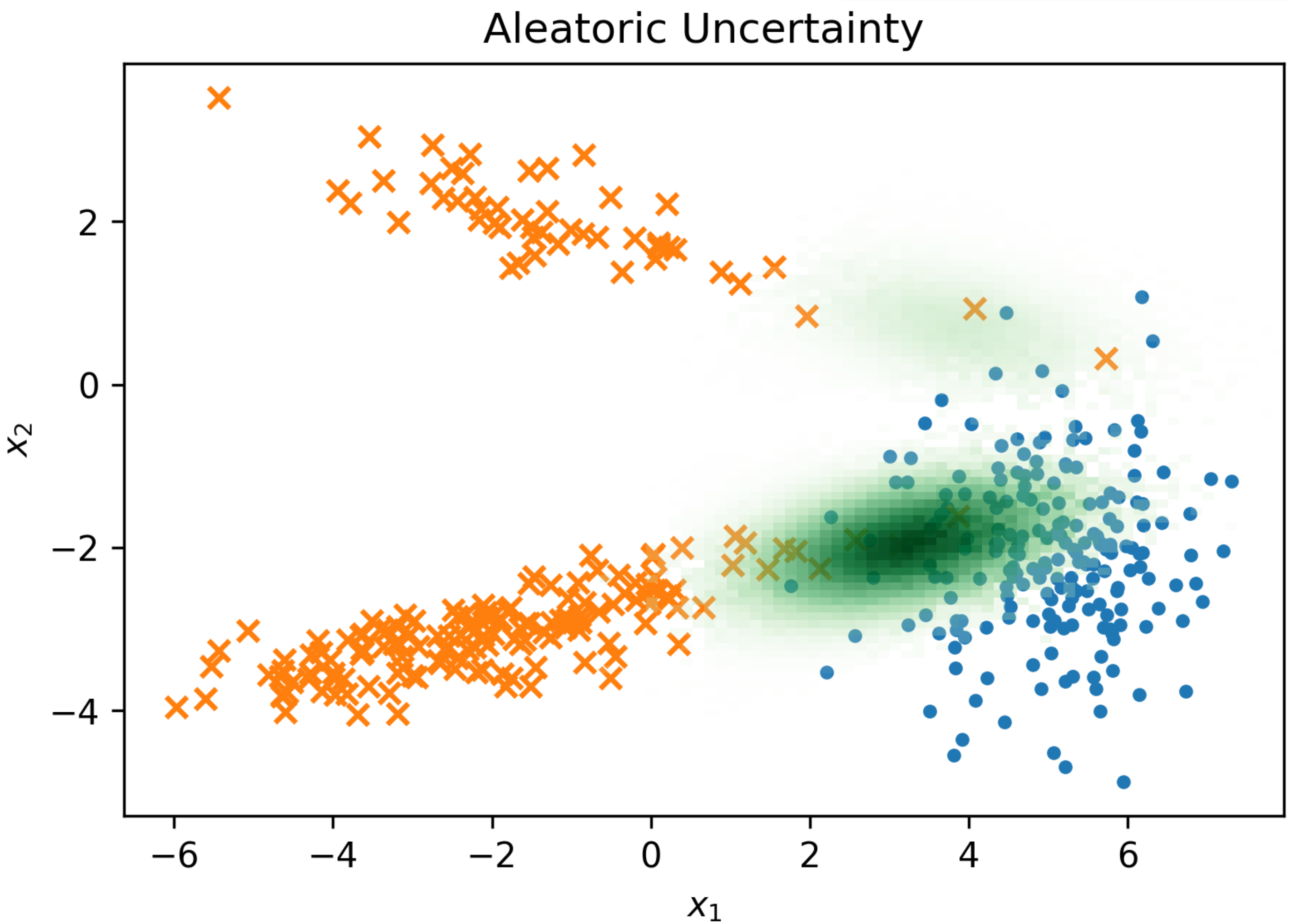
Uncertainty in Machine Learning: classification (2D "input", binary)

- Aleatoric/statistical = "true" inherent randomness
- Epistemic/systematic = missing data, bad model



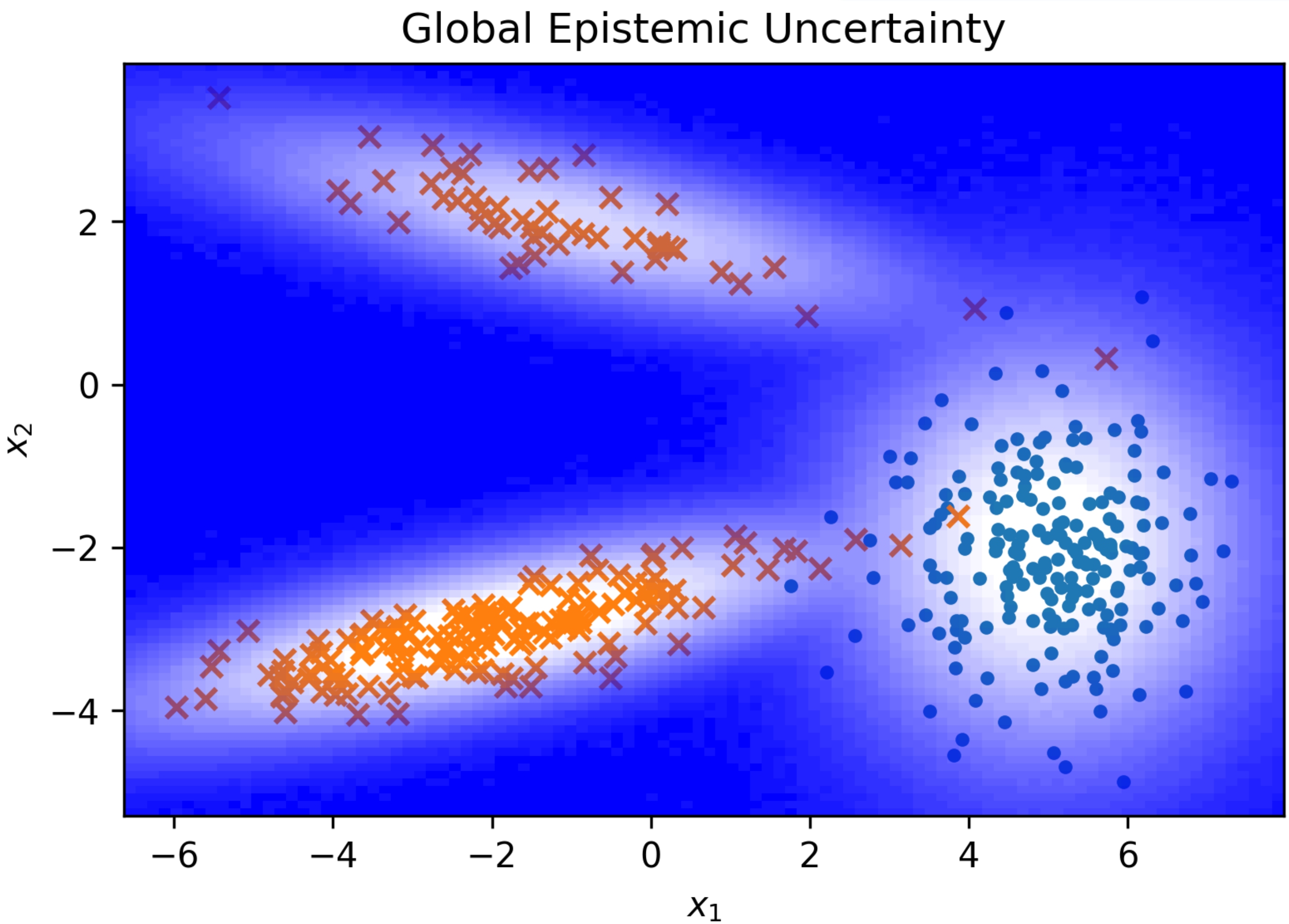
Toy 2D-Dataset: Aleatoric Uncertainty

Aleatoric/statistical = "true" inherent randomness
Epistemic/systematic = missing data, bad model



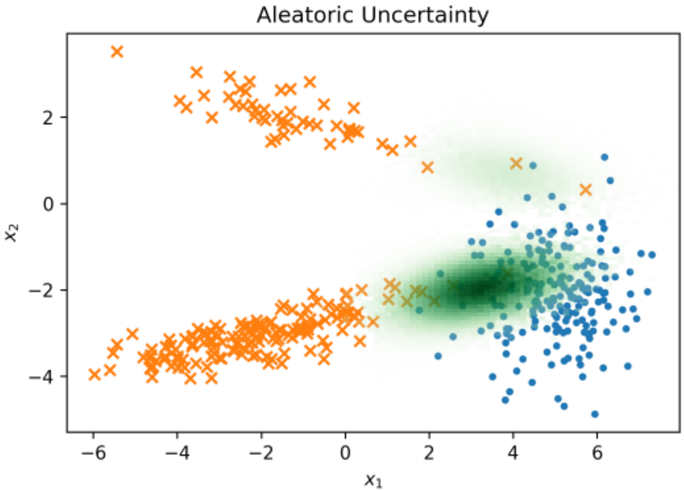
Toy 2D-Dataset: Epistemic Uncertainty

Aleatoric/statistical = "true" inherent randomness
Epistemic/systematic = missing data, bad model

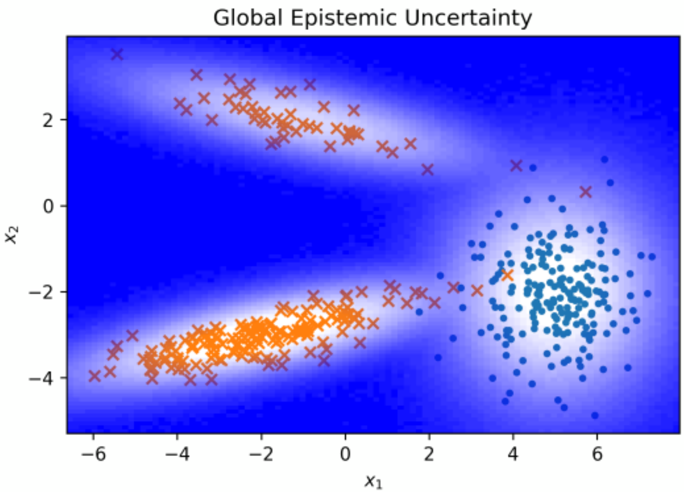


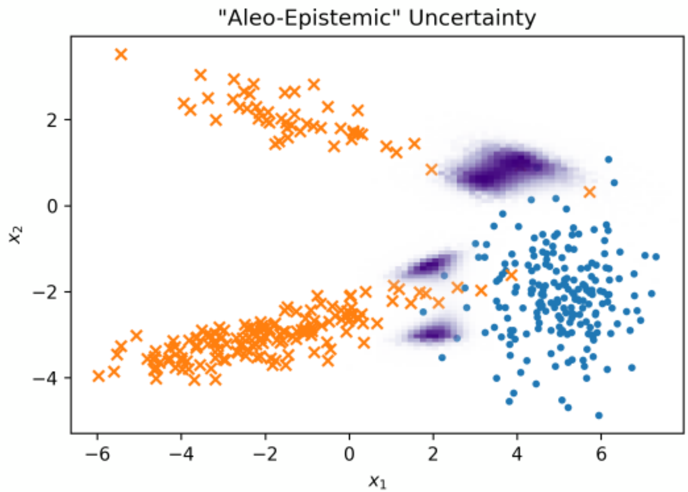
A Quick Visual Summary on Uncertainty

-Aleatoric/statistical = "true" inherent randomness
-Epistemic/systematic = missing data, bad model

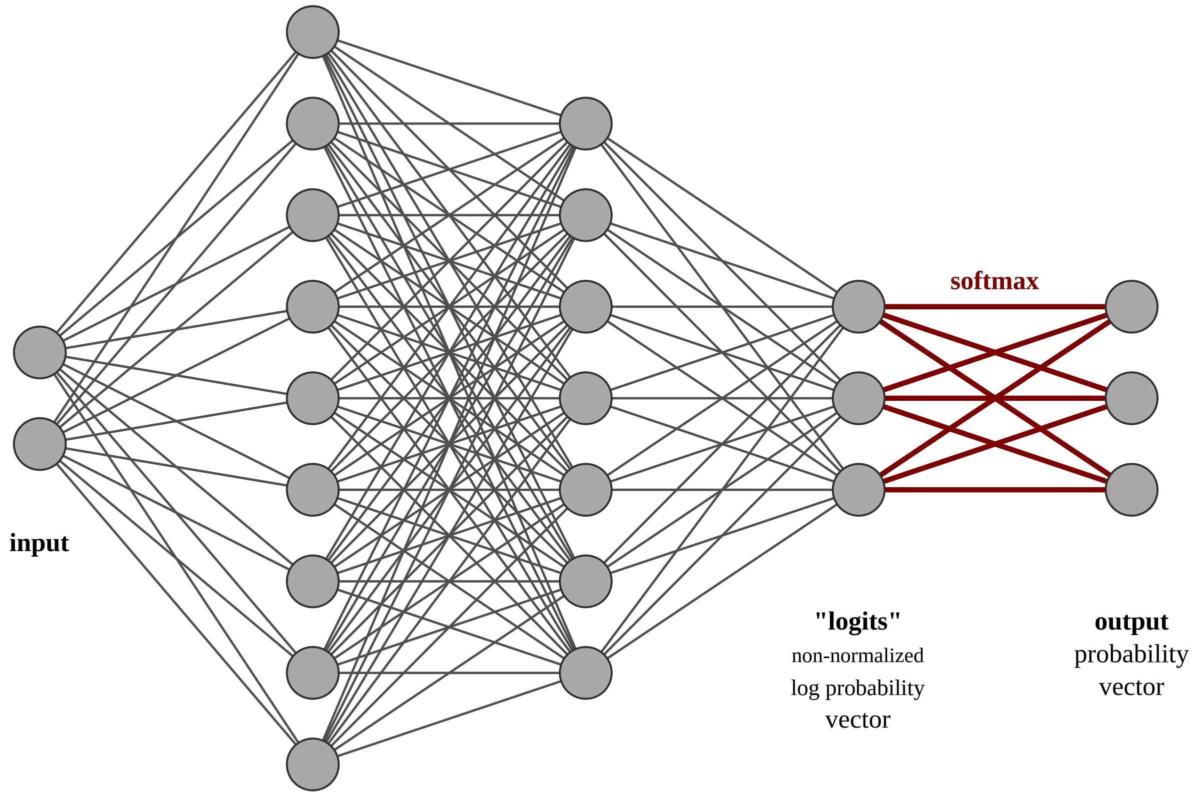


- Aleatoric uncertainty in regions of class overlap
- Epistemic uncertainty when OOD (out of distribution)
  - encompasses many different situations
  - NB: no perfect specification of what to do for OOD
- Epistemic uncertainty in regions of low data
  - especially with class imbalance, etc.
  - possibly combined with aleatoric



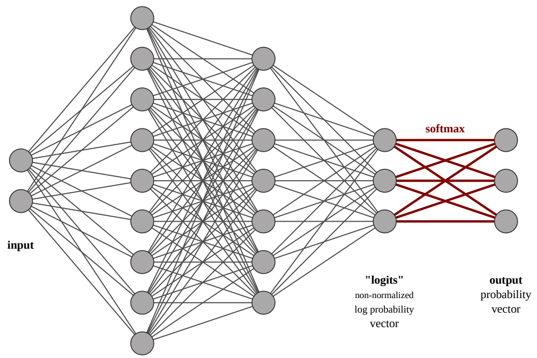


How Neural Nets do Classification? (reminder) (example with 3 classes)



- $o = softmax(l) \Leftrightarrow \forall i, o_i = \frac{\exp(l_i)}{\sum_j \exp(l_j)}$

A Probability for each class?



- A probability vector is better than just predicting a class
  - parallel with a regression setting
    - instead of predicting a single output value
    - predict a distribution (e.g., a mean and variance)
- But... ambiguity: aleatoric vs epistemic
  - what is actually uncertain (in the current representation space)
  - what the model doesn't know

A probability vector cannot convey all information

(but a good probability vector can be enough for some decision making)

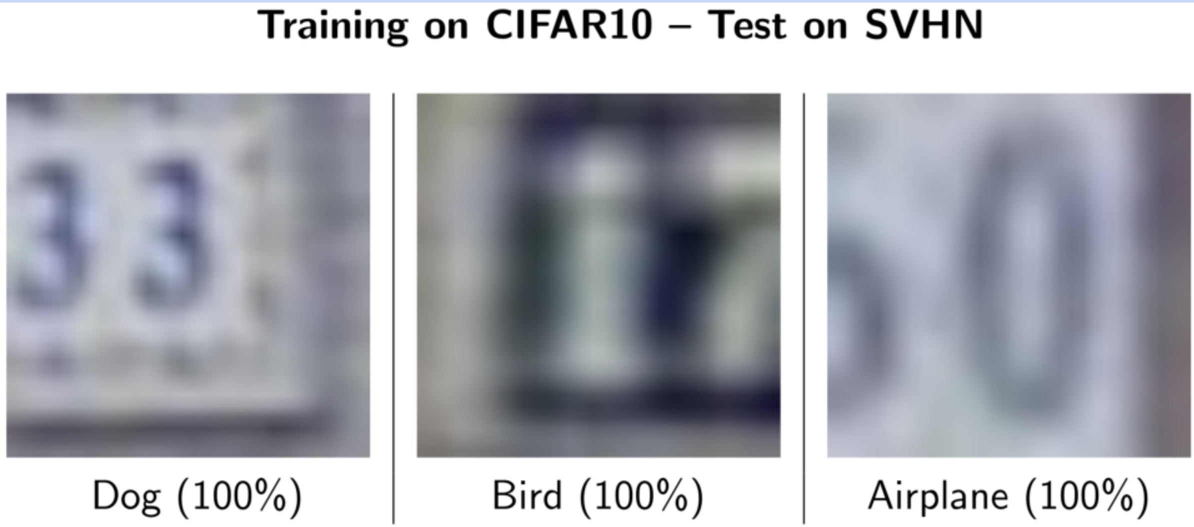
50% dog, 50% plane?

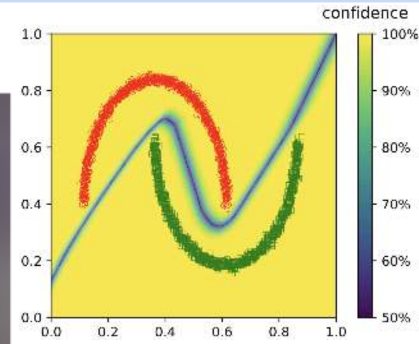
- aleatoric vs epistemic
- Example setup
  - a model trained to distinguish 2 classes, dog vs plane
  - on a new image, the network predicts 50%/50%
  - two possible situations

Case 1 !

Case 2 !

ReLU Networks are Overconfident (Hein et al., CVPR2019)





- Over-confident predictions

> A deep model doesn't know what it doesn't know

- NB: it is also over-confident in regions of inherent uncertainty

(Image from the companion-webpage https://github.com/max-andr/relu_networks_overconfident of:
Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem)

1. Why we need to quantify uncertainty?
2. Some sources of uncertainty
3. Statistical machine learning approaches for general uncertainty modeling
4. Deep Learning practices for uncertainty modeling
5. Bayesian Neural Networks
    1. Bayesian view of machine learning
    2. Variational inference
    3. Variational Dropout
6. Applications and Openings

(Re)Calibrating a Trained Model f

- Goal: properly quantifying aleatoric uncertainty
- Calibration = for every x, make the two following match,
  - the predicted output probably $f(x)$ from the model
  - and the actual class probability position $p(y|x)$
  - ⇒ "expected calibration error"
  - need binning (or density estimation) for estimation
- Possible solutions
  - re-fit/tune the likelihood/last layer (logistic, Dirichlet, ...)
  - e.g., fine tune a softmax temperature
    - $o = softmax_t(l) \Leftrightarrow \forall i, o_i = \dfrac{\exp(l_i/t)}{\sum_j \exp(l_j/t)}$
    - $t \to \infty$
      ⇒ $exp(.) \to 1$
      ⇒ $o \to uniform()$
    - $t \to 0$ ⇒ $softmax \to max$

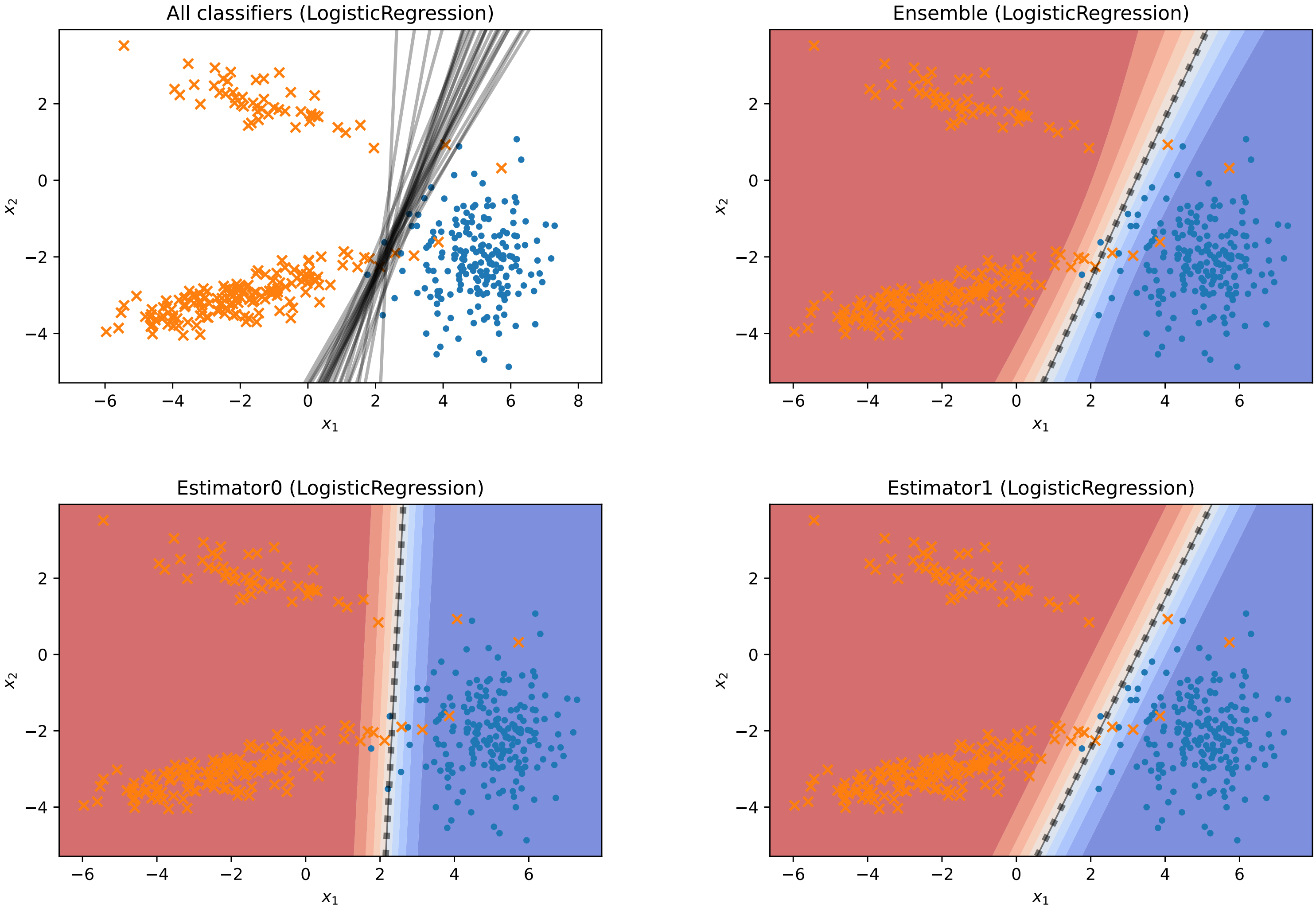
Dataset Shift, Domain Adaptation, Transfer Learning (not today's focus)

- Dataset shift
  - the "target" set is different from the training set
  - out of distribution situation
  - .
- Solution
  - Unsupervised domain adaptation (UDA)
    - use unlabeled target data to adapt
  - Usual approach
    - reduce the discrepancy
      between source and target datasets
    - a natural fit for the Optimal Transport theory
    - or, tune the classifier to become certain on data points

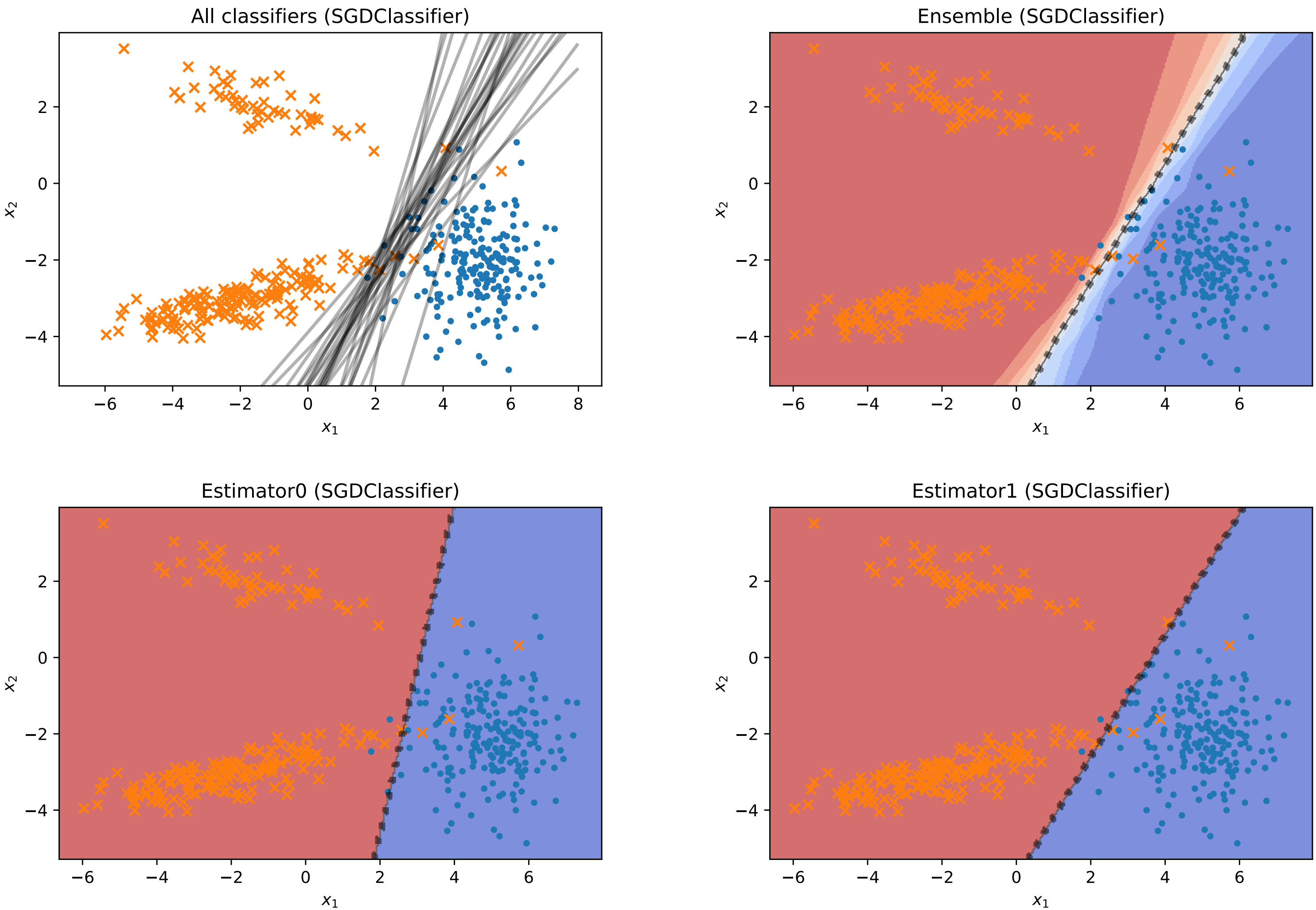
Ensemble methods

- General principle
  - learn several models
  - "average" their predictions
- Typical Approach: Bagging (bootstrap aggregating)
  - Sample several dataset, with replacement (bootstrap)
  - Average model predictions
  - Random Forests
    - maybe the most know bagging model
    - ensemble of decision trees
    - additionally use "feature bagging"
- NB on boosting (e.g., AdaBoost) (not a simple average)
  - use just-better-than-random models
  - iteratively train with re-weighted datasets
  - learn the weights of the models

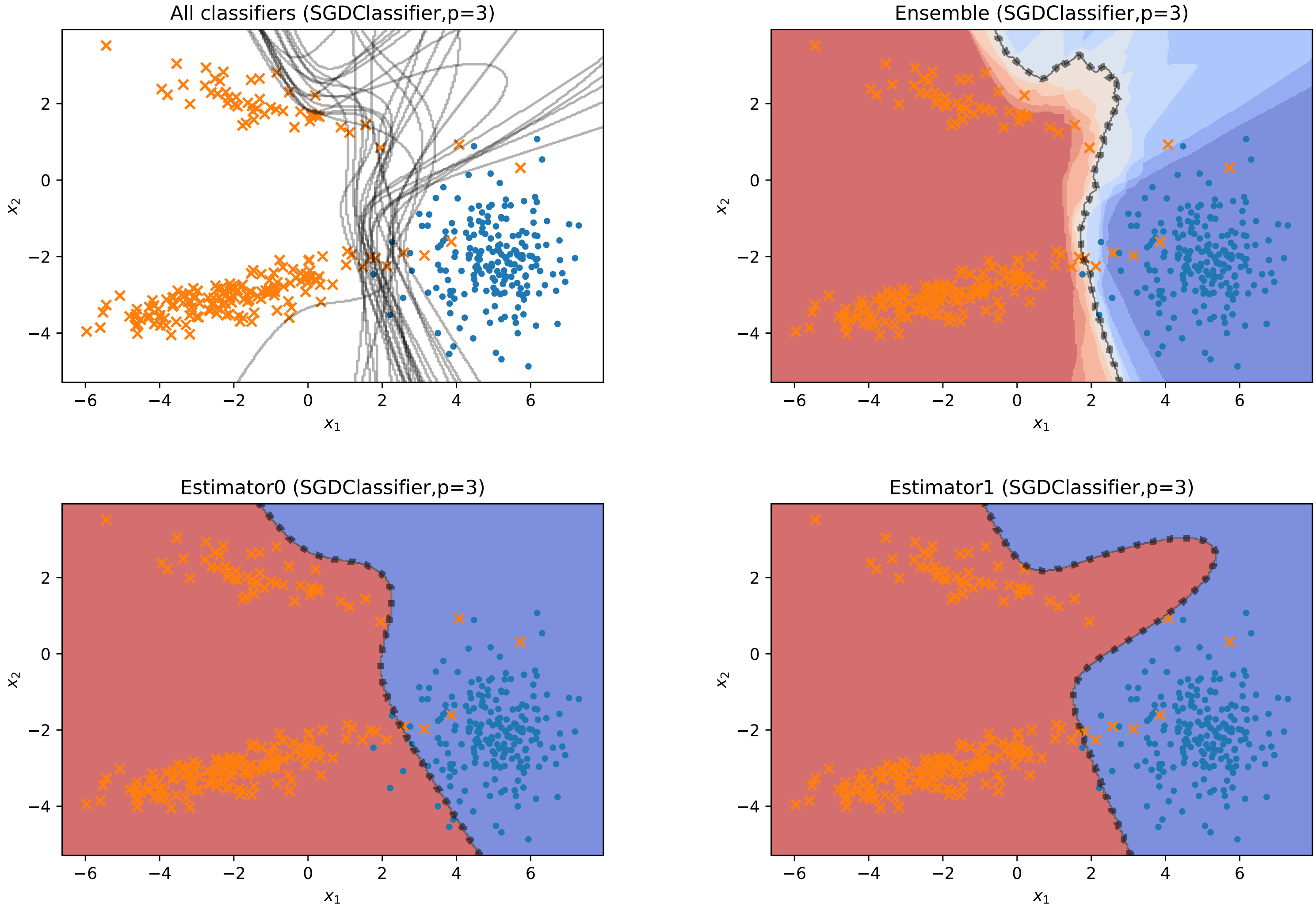
Logistic Regressor Bagging Example



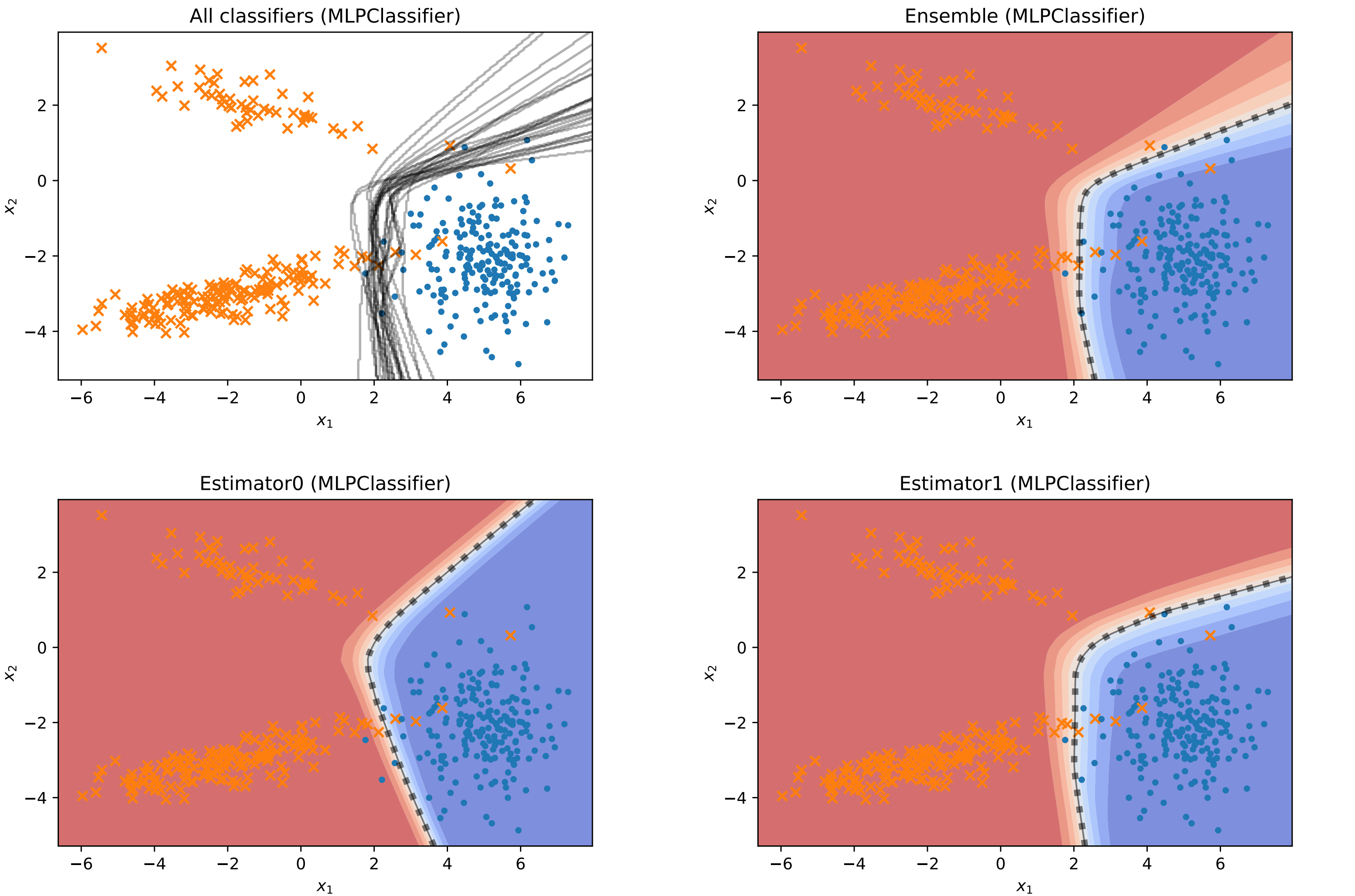
Support Vector Machines (SVM) Bagging Example



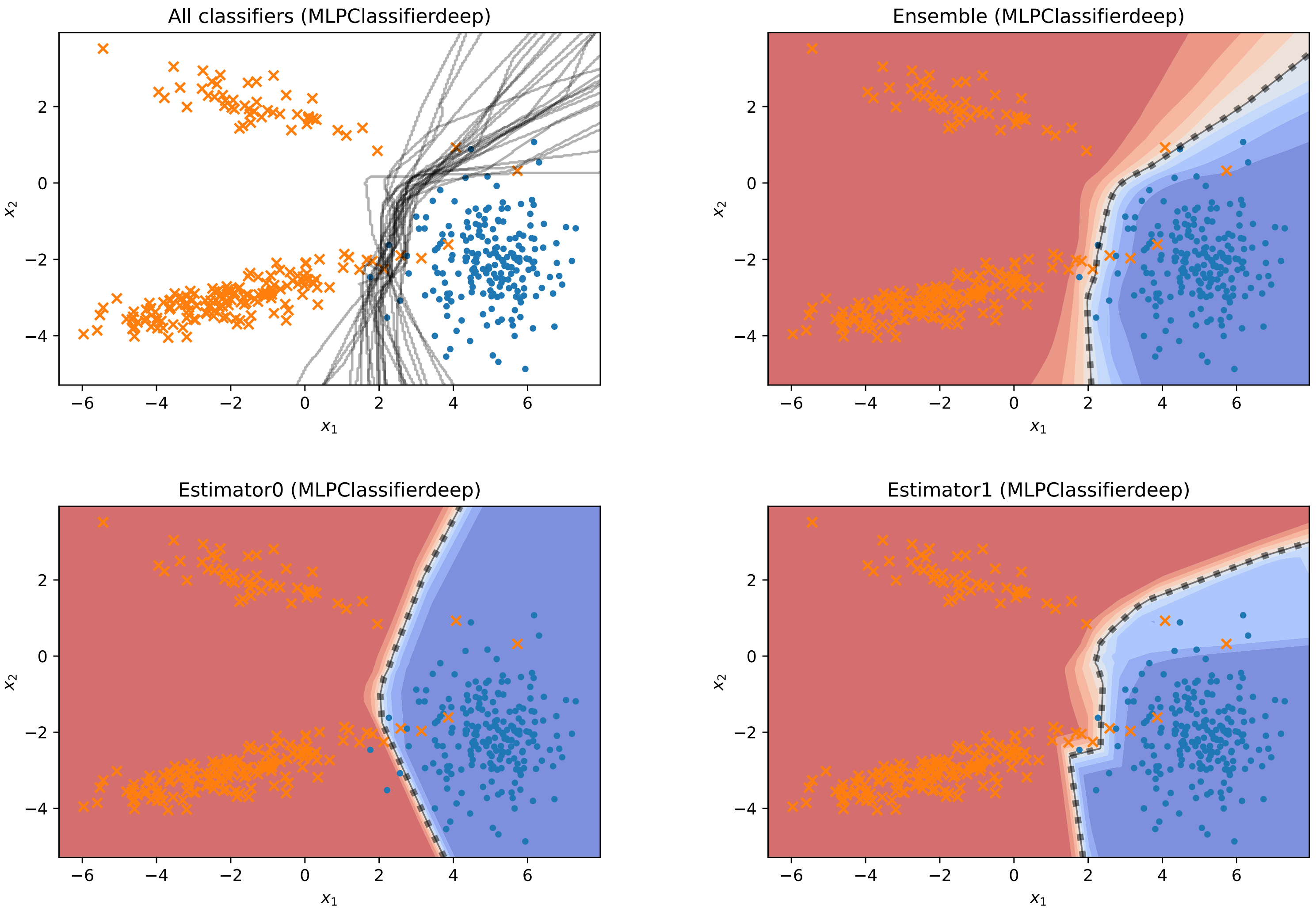
SVM Bagging Example with polynomial features (kernel)



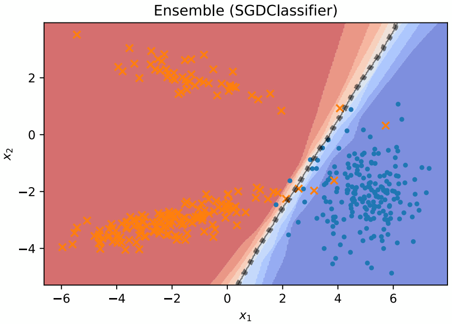
Multi-Layer Perceptron (MLP) Bagging Example

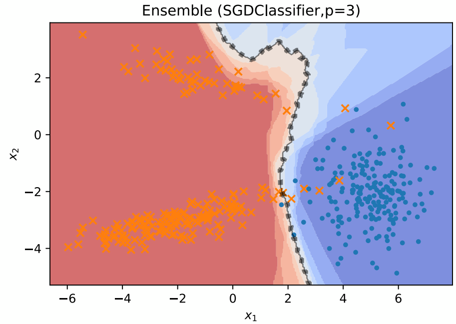


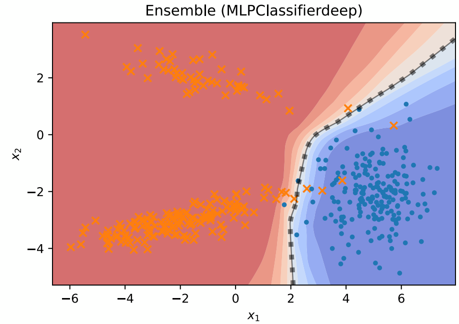
Very-Deep MLP Bagging Example



Summary on bagging







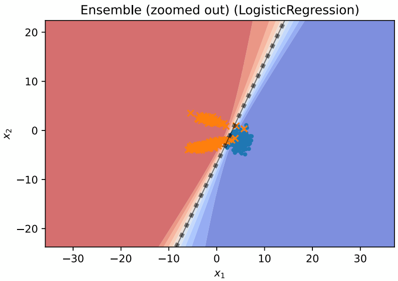
Bagging nicely handles quantifying aleatoric uncertainty

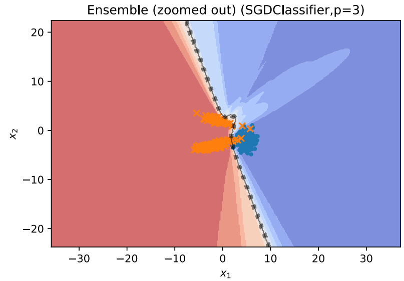
- It works even with overconfident base models
- The sampling creates the necessary noise in boundary regions

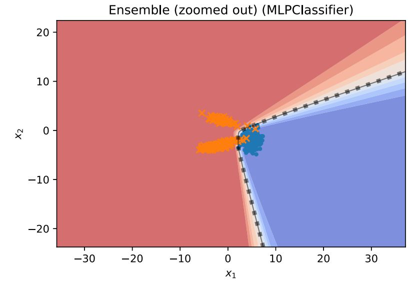
The model family controls Bagging's Out of Distribution behavior

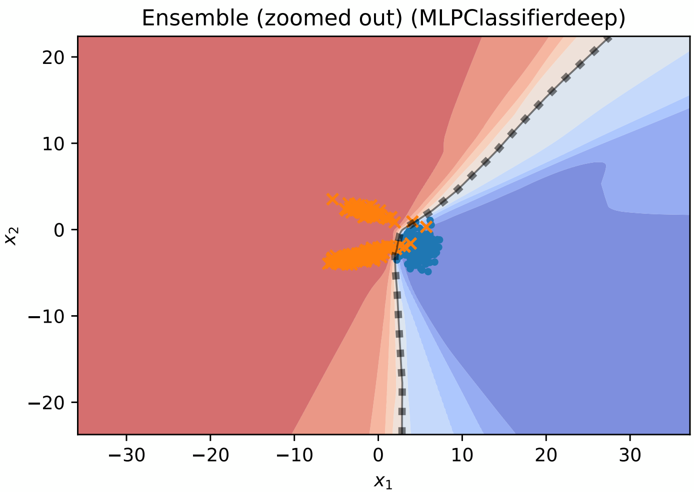
- Simple models have low OOD variety ⇒ 100% over-confidence
- Complex/varied models/features better assess epistemic uncertainty
- However, still a major OOD over-confidence (like most discriminative models)

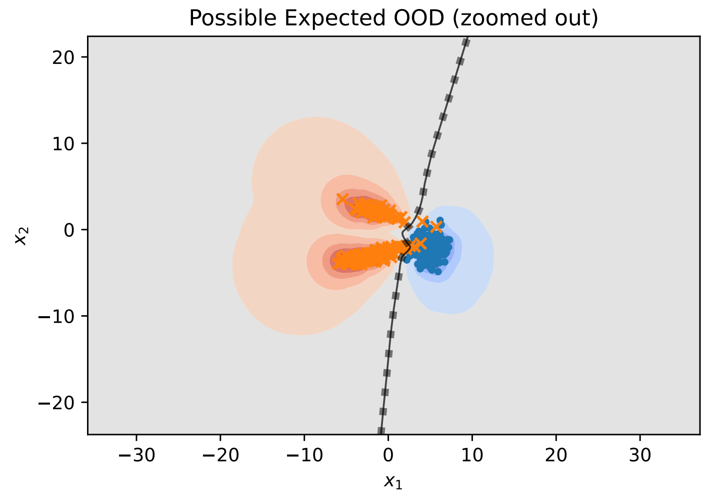
Zooming out to see Out Of Distribution Over-confidence

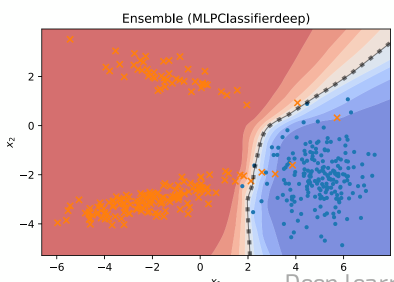






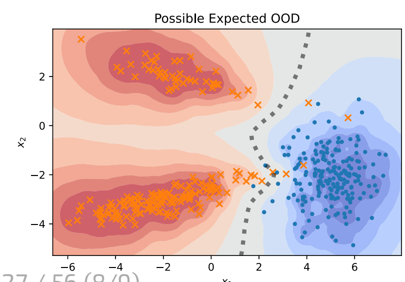






- Purely-discriminative models fail at OOD
- *We will ignore this issue for some time*

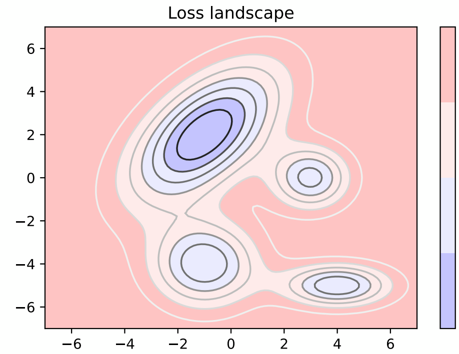
- Some solutions
  - Use Gaussian Processes
  - Combine with one-class / density est.
  - Force doubt on generated OOD samples



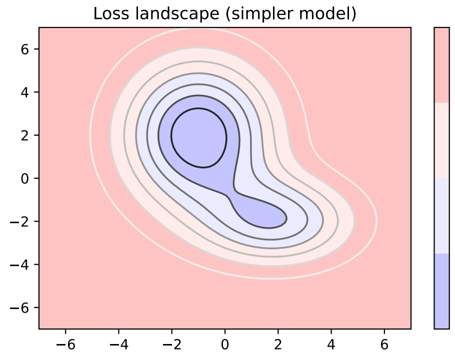
1. Why we need to quantify uncertainty?
2. Some sources of uncertainty
3. Statistical machine learning approaches for general uncertainty modeling
4. Deep Learning practices for uncertainty modeling
5. Bayesian Neural Networks
    1. Bayesian view of machine learning
    2. Variational inference
    3. Variational Dropout
6. Applications and Openings

Learning Ensembles of Deep Models

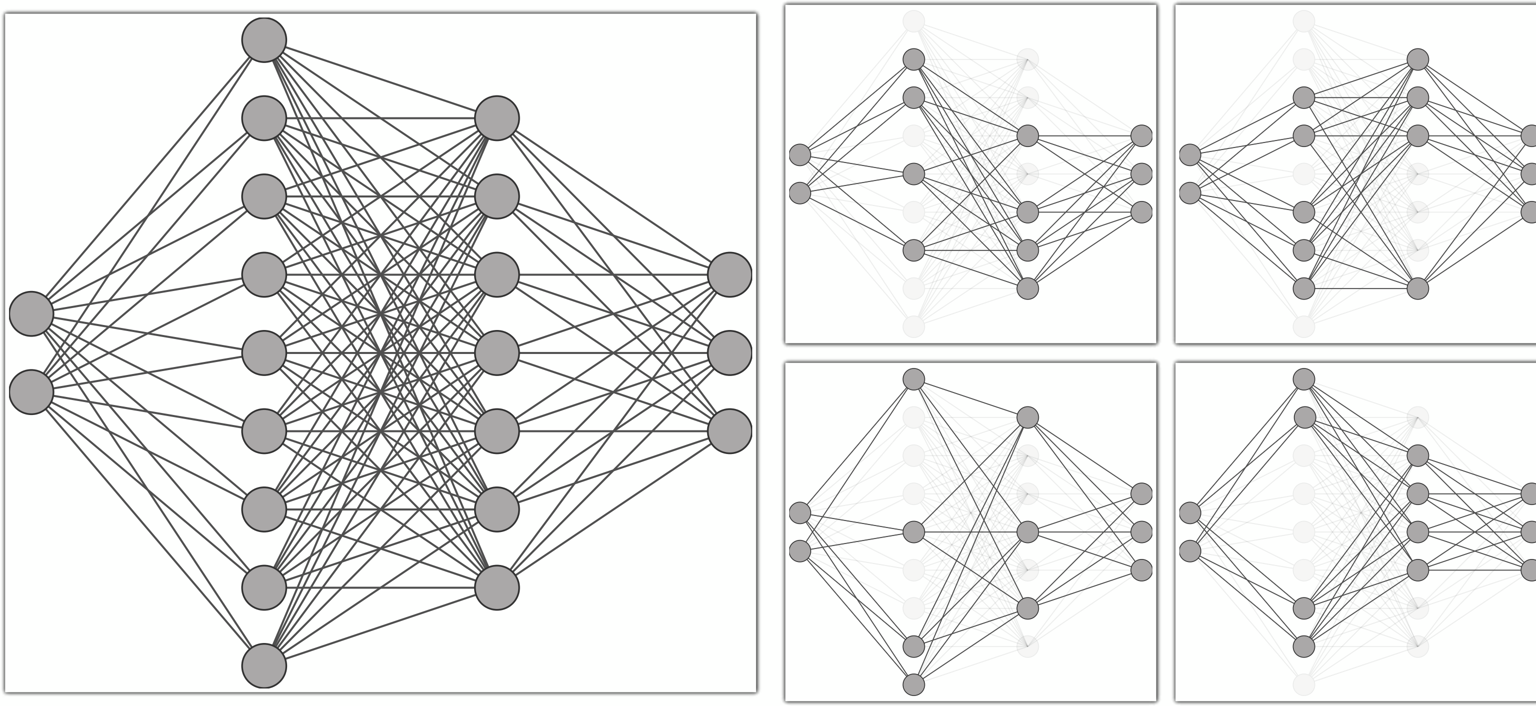
- Re-seeding for stochastic methods
  - works very well in practice
  - resource intensive (mem., process)
  - some variations/optimizations
    - snapshot + cyclic learning rate



- Different families of models
  - different hyper-parameters
  - different architectures



Stochastic Learning as Model Ensembling



- Dropout: simultaneously training "2^N models" (with shared parameters)
  - randomly set weights or activations to 0 (for every SGD sample)
  - NB: often, "weight scaling rule" at test time -> very bad, no uncertainty
  - NB: dropout should be applied at test time (costly)
- Other sources of stochasticity and ensembling
  - (mini-)Batch normalization
  - Stochastic (minibatch) Gradient Descent

Diverse vs Local Ensembling

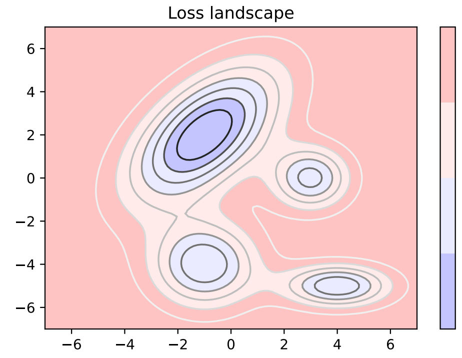
- Re-seeding, Dropout, ...
  ⇒ diverse ensemble

- Mode fitting (local diversity)

  1. learn a single model
  2. estimate the local loss landscape
  3. create several perturbations of the model
  4. use all the models as an ensemble

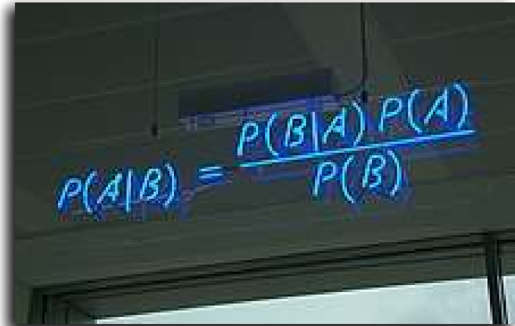
- Bayesian Neural Networks (BNN)

  - "dense" ensemble
  - can be both local or diverse



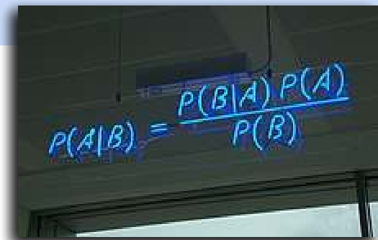
1. Why we need to quantify uncertainty?
2. Some sources of uncertainty
3. Statistical machine learning approaches for general uncertainty modeling
4. Deep Learning practices for uncertainty modeling
5. Bayesian Neural Networks
    1. Bayesian view of machine learning
    2. Variational inference
    3. Variational Dropout
6. Applications and Openings

1. Why we need to quantify uncertainty?
2. Some sources of uncertainty
3. Statistical machine learning approaches
   for general uncertainty modeling
4. Deep Learning practices for uncertainty modeling
5. Bayesian Neural Networks
   1. Bayesian view of machine learning
   2. Variational inference
   3. Variational Dropout
6. Applications and Openings



- "Bayesianism"
  - Everything as random variables
  - Use (conditional) probabilities ... a lot
- Two probability rules
  - Product rule: $P(A,B) = P(A|B)\ P(B)$ = $P(B|A)\ P(A)$
  - Marginalization, Sum rule: $P(B) = \sum_A P(A,B) \triangleq \sum_a P(A=a,B)$
- And in the Baye's rule[W] bind them
  - $$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)} = \frac{P(B|A)\ P(A)}{\sum_{A'} P(B|A')\ P(A')}$$
- NB: Exactly the same holds with probability densities (for continuous random variable)

Principle of Bayesian "Learning"

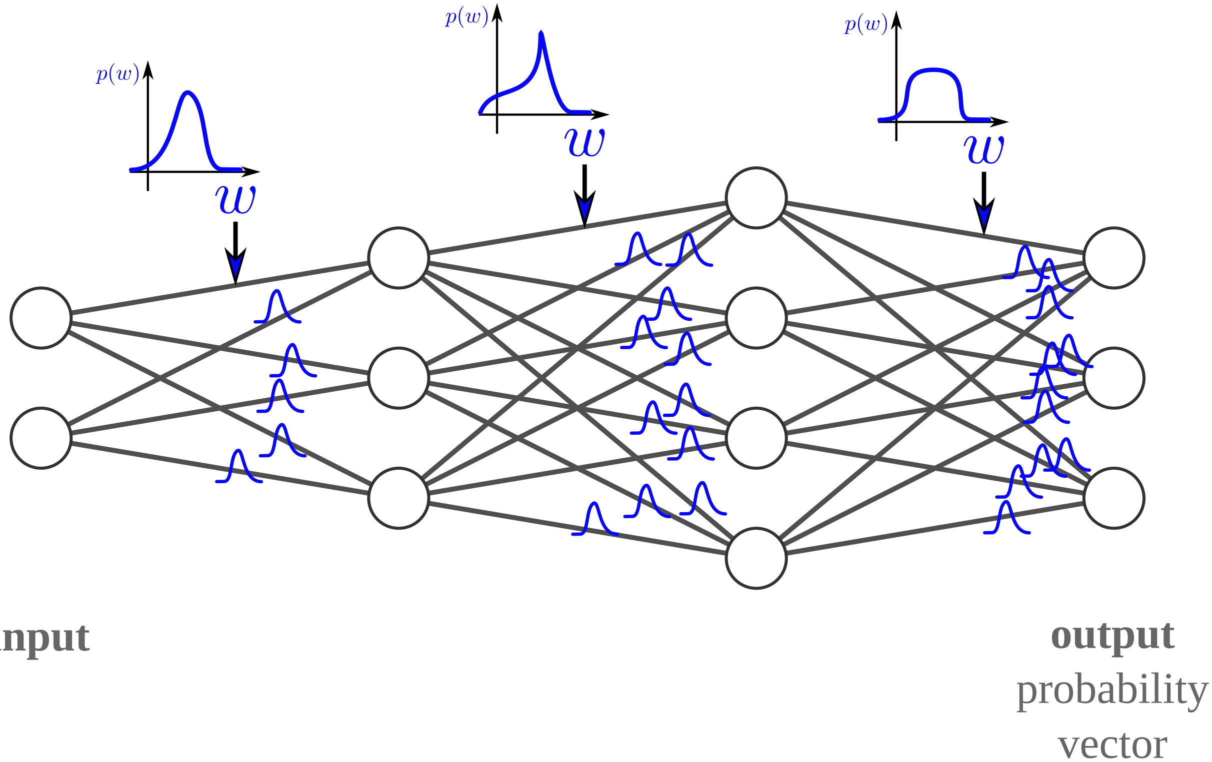


- Use probabilities to
  - represent non-deterministic laws
  - represent uncertainty (aleatoric and epistemic)
  - reason about uncertainty (do learning, inference)
- Considering
  - some parameters (e.g., weights of the network, W)
  - some dataset (e.g., both training inputs and labels, X)
- We have
  - $P(W|X) = \dfrac{P(X|W)\,P(W)}{P(X)} \propto P(X|W)\,P(W)$
- More verbosely

$$P_{posterior}(weights|trainset) = \frac{P_{likelihood}(trainset|weights)\,P_{prior}(weights)}{P_{constant}(trainset)}$$

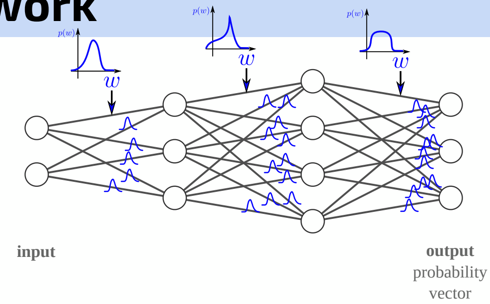
- Posterior probability
  - probability distribution of the parameters given the training set
  - i.e. what we know about the parameters after seeing the training set

Principle of Bayesian Neural Networks



- Typical BNN: have a 1D Normal distribution on each weight
  - 1 mean and 1 variance per weight
  - 1 billion weights ⇒ 2 billions parameters

Training a Bayesian Neural Network



- Bayesian "learning"
  - Goal: finding the *posterior distribution on the parameters*
  - $P_{posterior}(weights|trainset)$

$$= \frac{P_{likelihood}(trainset|weights)\; P_{prior}(weights)}{P_{constant}(trainset)}$$

  - Prediction for a new input x

$$f_{posterior}(x) = \int f_{weights}(x) \cdot P_{posterior}(weights|trainset)$$

- Variational Inference (VI) (or Stochastic Gradient Variational Bayes, SGVB)
  - Parameterize $P_{posterior}(weights|trainset)$, e.g., a $\mathcal{N}ormal$ per weight $\Rightarrow$ 1 billion means and variances
  - Do stochastic gradient descent (SGD)
  - Sample a weight at every forward pass i.e., approximate the $\int \cdots$ by a single sample

Variational Inference: BNNs vs VAE

- Compared to Variational Autoencoders (VAE)
  - Also use the "reparameterization trick"
    - change $w \sim \mathcal{N}(\mu, \sigma^2)$ to $\varepsilon \sim \mathcal{N}(0,1)\ ;\ w = \mu + \varepsilon.\sigma$
    - allow the gradient to "flow" to μ and σ
      (VAE: allow the gradient to flow to the encoder)
  - Model distribution on millions of parameters
    (VAE: distribution on latent variables, only a few, but for each data point)
  - No per-data (latent) variables ⇒ no need for amortization

Bayesian Neural Networks - Uncertainty Quantification (Overview)

1. Why we need to quantify uncertainty?
2. Some sources of uncertainty
3. Statistical machine learning approaches
   for general uncertainty modeling
4. Deep Learning practices for uncertainty modeling
5. Bayesian Neural Networks
    1. Bayesian view of machine learning
    2. Variational inference
    3. Variational Dropout
6. Applications and Openings

Dropout and Bayesian Neural Networks

- Traditional (weight) dropout
  - for each weight, "set" it to 0 with a probability $1-p$
  - at test time, multiply weights by p (weight scaling rule)

- Interpretation as a (fixed) distribution
  - $w_i \sim Mixture_p(0, v_i)$, or
    $\varepsilon_i \sim Bernoulli(p)\ ;\ w_i = \varepsilon_i \cdot v_i$ (reparameterized)
  - Bayesian implication: apply dropout at test time
    (sometimes called "monte carlo dropout")
    - $f_{posterior}(x) = \int f_w(x) \cdot P_{posterior}(w|trainset)$
    - $f_{posterior}(x) = \mathbb{E}_{w \sim P_{posterior}(w|trainset)} \left[f_w(x) \right]$
    - $f_{posterior}(x) \approx \frac{1}{D} \sum_{j=1}^{D} f_{w^j}(x)$ with $w^j \sim P_{posterior}(w|trainset)$
    - $f_{posterior}(x) \approx \frac{1}{D} \sum f_{v^j \cdot \varepsilon^j}(x)$ with $\varepsilon_i^j \sim Bernoulli(p)$

Bayesian Treatment of Dropout

- Bayesian approach, reminder: everything is a random variable
- Treat p as a random variable (during "training")
  - first, how many p do we want to use?
    - a single p for the whole network
    - **a p per layer**
    - a p per weight
  - learn the posterior on the parameters θ, including p
    - $\theta = \{W, p\}$
    - $P(\theta | trainset) \propto P_{likelihood}(trainset | \theta) \cdot P_{prior}(\theta)$
    - $\Rightarrow$ need a prior... that acts as a natural regularization
  - non trivial optimization
- A variety of dropout: Bernoulli, Gaussian (multiplicative) dropout, sparsifying prior, ...

(e.g. Learnable Bernoulli Dropout for Bayesian Deep Learning)

- Bayesian treatment of neural networks
  - consider each weight as a random variable
  - formulate a prior on the weights
  - observe some (training) data, and given that…
  - … infer the (posterior) knowledge on these weights
  - use this knowledge for prediction

- Several flavors, including
  - Dropouts
    - Bernouilli or gaussian
    - learnable dropout parameters
    - using prior to encourage network sparsity
  - $\mathcal{N}ormal$ modeling of each weight

- Learning, inference, testing
  - train by gradient descent (SGD) on the "variational" parameters
  - use sampling at test time to produce several predictions

1. Why we need to quantify uncertainty?
2. Some sources of uncertainty
3. Statistical machine learning approaches for general uncertainty modeling
4. Deep Learning practices for uncertainty modeling
5. Bayesian Neural Networks
    1. Bayesian view of machine learning
    2. Variational inference
    3. Variational Dropout
6. Applications and Openings

Estimating Uncertainty and Interpretability in Deep Learning for Coronavirus (COVID-19) Detection

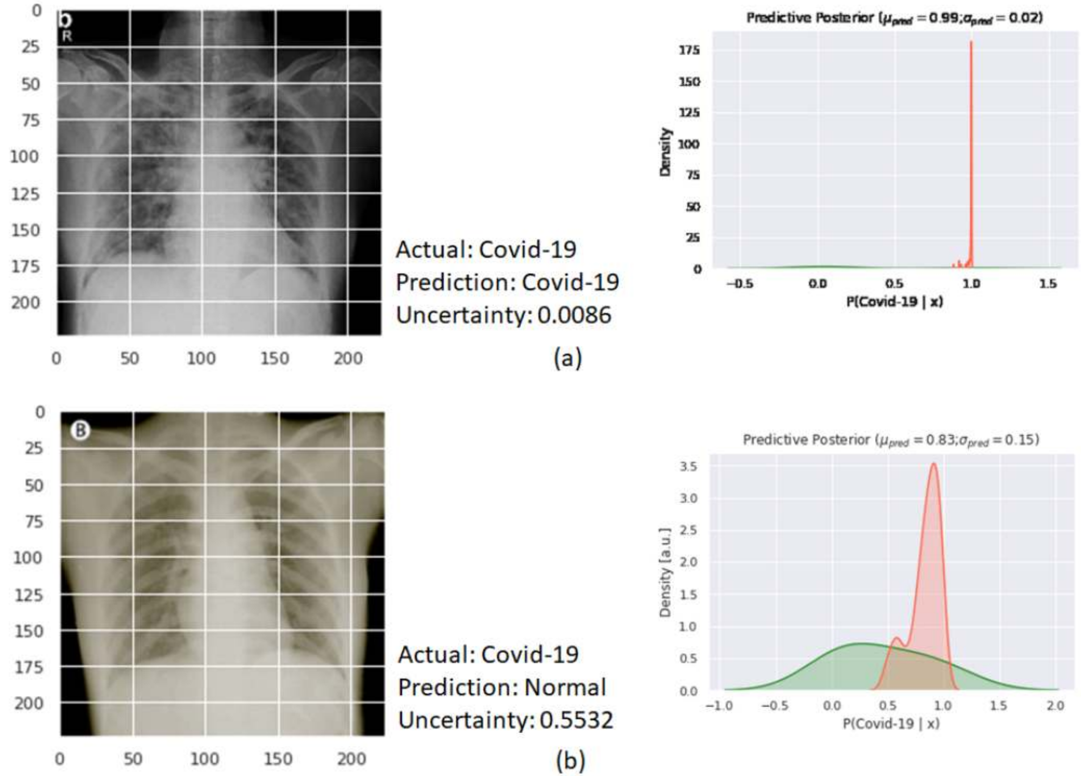


Fig. 1. Example input images with uncertainty and the corresponding prediction distributions generated by Bayesian DNN. Figure 1(a) shows a correctly classified image where the model is highly certain about its prediction (PH=0.0086). Whereas Figure 1(b) shows a miss-classified image where the model is uncertain (PH=0.5532) and wider posterior distributions.

- (Biraja Ghoshal, Allan Tucker)
- dropweight on a BCNN
- improve classification
- good quantification
- human/machine combination

Towards safe deep learning: accurately quantifying biomarker uncertainty in neural network predictions

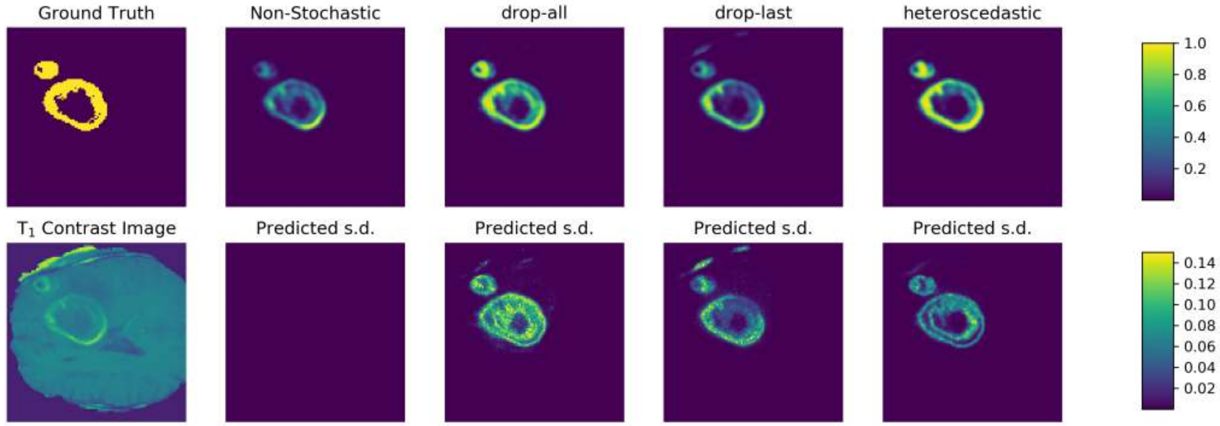
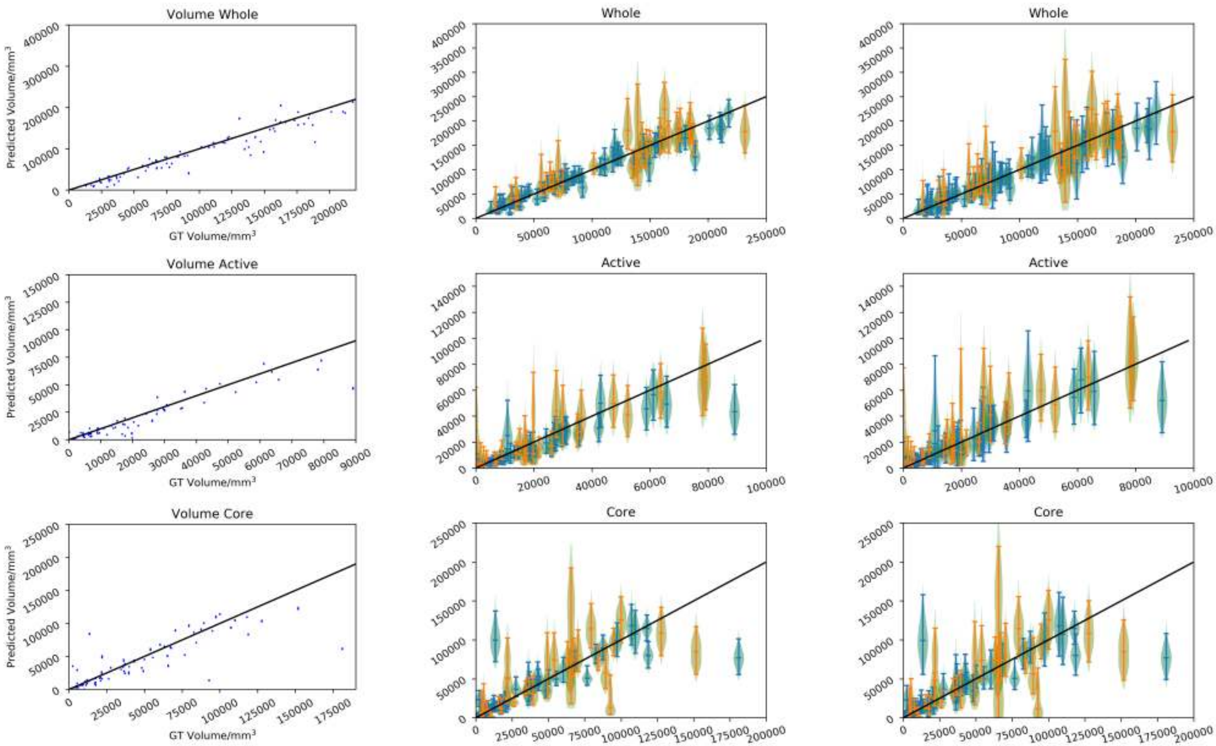


Fig. 1: Top row: mean confidences (over 20 forward passes) for the given model to belong to the 'Active' class. Bottom row: $\sigma_{predicted}$, the standard deviation across predictions. Different methods display similar predictions, but the level of uncertainty varies depending on the network used.

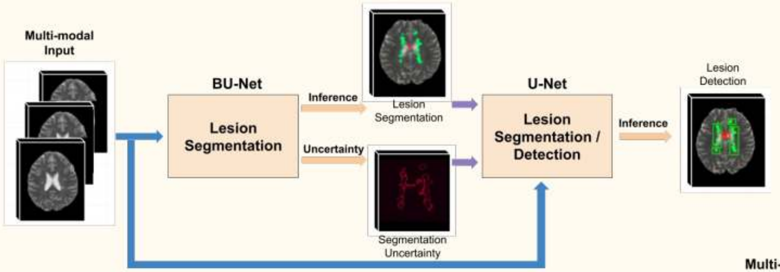
- (Zach Eaton-Rosen, Felix Bragman, Sotirios Bisdas, Sebastien Ourselin, M. Jorge Cardoso)



(a) Volume Estimates (b) Uncalibrated (c) Calibrated

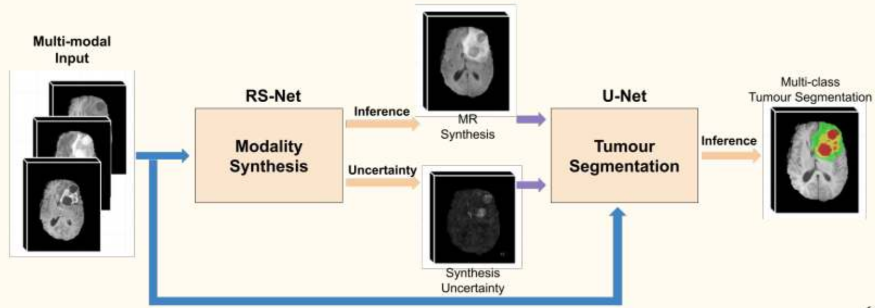
Fig. 3: In Figure 3a, we plot point estimates for 20 estimates of the volume from HR_{drop_all}. Despite 20 estimates being plotted, there is almost no variation that can be seen in the plots. This lack of variation is present in all methods (unshown). In 3b, we see the uncalibrated estimates of uncertainty, and in 3c we plot the calibrated estimates. Subjects tion' set are blue, and 'test' orange.

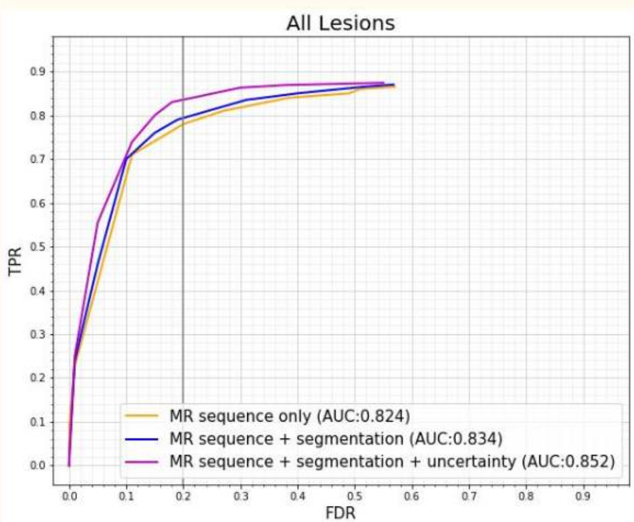
Propagating uncertainty across cascaded medical imaging tasks for improved deep learning inference

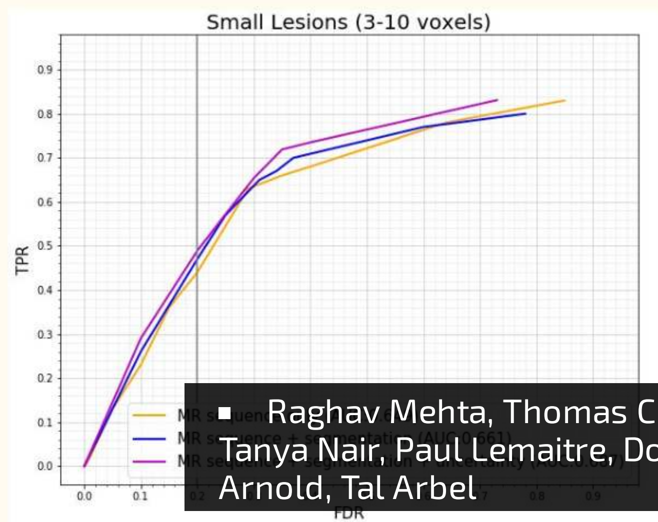


MS T2 lesion segmentation detection

Brain Tumour Segmentation



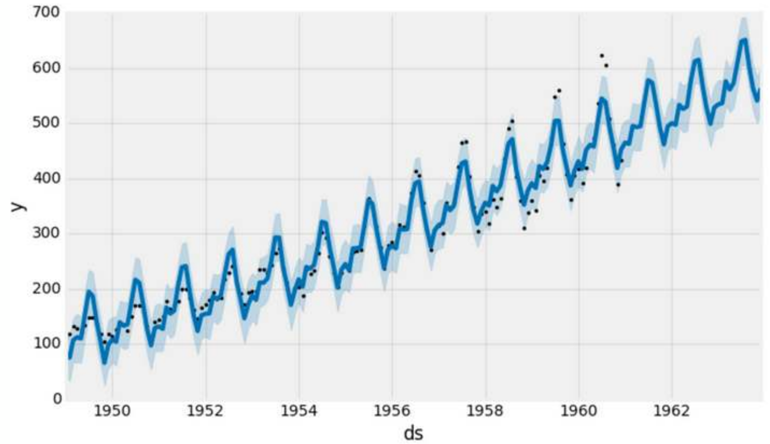




- Raghav Mehta, Thomas Christinck, Tanya Nair, Paul Lemaitre, Douglas L. Arnold, Tal Arbel

Gaussian processes (GP)

- GP = Infinitely-wide BNN
- Well formalized
  - closed forms
  - lot of works around scaling
- Allows to include prior learning
- Probabilistic, can be combined "easily"
- Deep GP avoid too much kernel choice



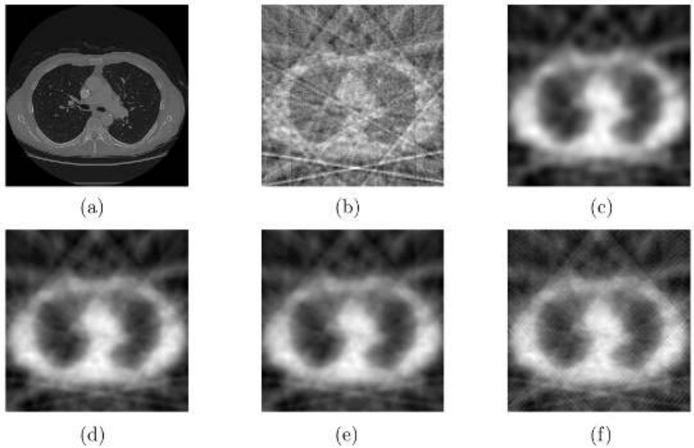
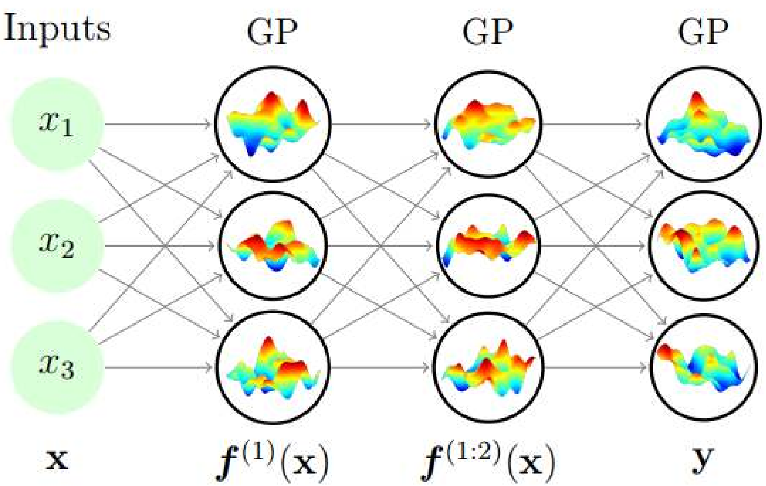


Figure 4. (a) A ground truth of 2D chest phantom. (b) Filtered backprojection reconstruction (Ram–Lak filter) from nine projections. (c) GP reconstruction using SE covariance, (d) GP reconstruction using Matérn covariance, (e) GP reconstruction using Laplacian covariance, (f) GP reconstruction using Tikhonov covariance. The GP reconstructions are using nine projections.



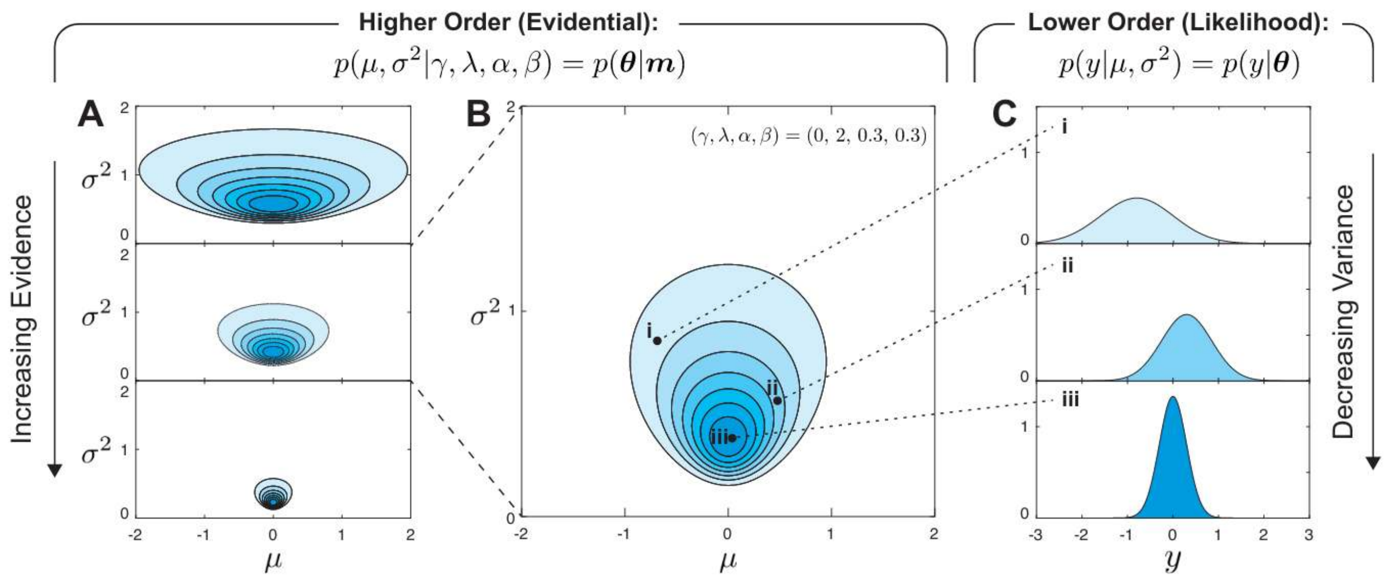
(Avoiding pathologies in very deep networks, Duvenaud et al.)

Going beyond probabilities?

- Probabilities are a way of represent belief
- It might be necessary to also represent confidence
- Some possible directions
  - Imprecise Probabilities [W]
  - Dempster–Shafer Theory (DST) [W]
    - theory of belief functions
    - evidence theory

Evidential Deep Learning

- Principle
  - instead of: predicting the parameters of an (aleatoric) distribution
  - do: predict a distribution over these parameters
- A way of learning/representing epistemic uncertainty on aleatoric uncertainty



(e.g., Deep Evidential Regression (above))
(e.g., Evidential Deep Learning to Quantify Classification Uncertainty)

Bayesian Neural Networks - Uncertainty Quantification

- Rémi Emonet – Hubert Curien Laboratory
- Deep learning for medical imaging school
- 2021-04-21

THANK YOU!

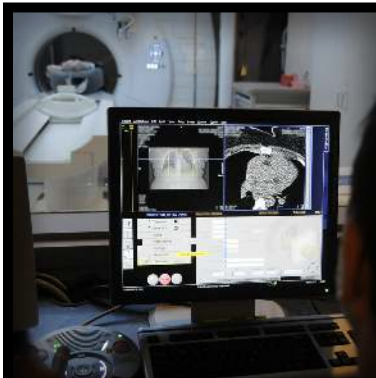
Questions?

SCAN ME

(presentation page)
or go to:
home.heeere.com

Attribution

CC by Official U.S. Navy Imagery (Flickr)



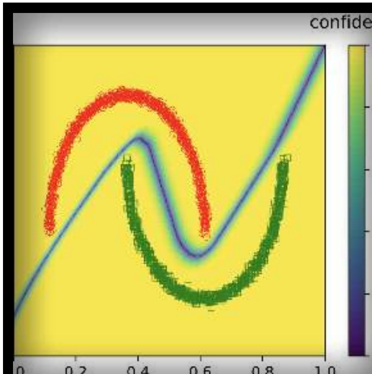
United States Government Work (Flickr)

emDocs 'Multiple Layers of Diagnostic Uncertainty'

Pediatric Ethiscope Embracing Diagnostic Uncertainty

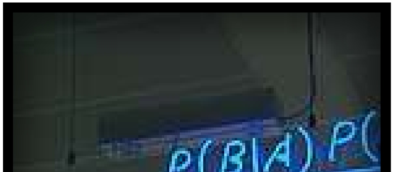
CC by DaPuglet (Flickr)

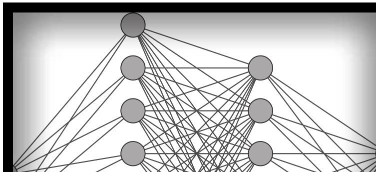
CC by mlinksva (Flickr)

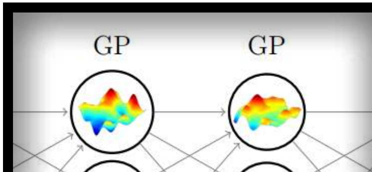


By @max-andr (github) /relu_networks_overconfident

By @max-andr (github) /relu_networks_overconfident







- Wikipedia articles
  - Bayesian Probabilities[w]
  - Imprecise Probabilities[w]
  - Dempster–Shafer Theory (DST)[w]
  - Proper Scoring Rules[w] (and calibration)
- Some (limited) pointers to research articles
  - Unclassified
    - The need for uncertainty quantification in machine-assisted medical decision making
  - On calibration
    - Well-calibrated regression uncertainty in medical imaging with deep learning
  - About dropout
    - Variational dropout and the local reparameterization trick
    - Variational dropout sparsifies deep neural networks
    - Variational Gaussian dropout is not Bayesian
    - Learnable Bernoulli dropout for Bayesian deep learning
  - Evidential deep learning
    - Evidential Deep Learning to Quantify Classification Uncertainty .
    - Deep Evidential Regression .
- Some online references
  - Susan Holmes on Bayesian statistics
  - NeurIPS lecture on evidential deep learning

1. Why we need to quantify uncertainty?
2. Some sources of uncertainty
3. Statistical machine learning approaches for general uncertainty modeling
4. Deep Learning practices for uncertainty modeling
5. Bayesian Neural Networks
    1. Bayesian view of machine learning
    2. Variational inference
    3. Variational Dropout
6. Applications and Openings

Bayesian Neural Networks - Uncertainty Quantification

- Rémi Emonet – Hubert Curien Laboratory
- Deep learning for medical imaging school
- 2021-04-21

THANK YOU!

Questions?

SCAN ME

(presentation page)
or go to:
home.heeere.com

LABEX PRIMES UNIVERSITÉ DE LYON · CREATIS · LABORATOIRE HUBERT CURIEN UMR • CNRS • 5516 • SAINT-ETIENNE · UNIVERSITÉ DE SHERBROOKE · ÉTS Le génie pour l'industrie · ANR