DE LA RECHERCHE À L’INDUSTRIE


Druva

Dealing with uncertainties in Machine Learning

Workshop Machine Learning 2022
IN2P3/Irfu

Geoffrey DANIEL, CEA/DES/ISAS/DM2S/STMF/LGLS

Plan

Dealing with uncertainties in Machine Learning

1. Different levels of uncertainties
2. Uncertainties estimation in Deep Learning
3. Validation of uncertainties

Works in collaboration with Jean-Marc MARTINEZ (DES/ISAS/DM2S/STMF/LGLS)
And three interns:

- Mohamed Bahi YAHIAOUI (Mines St Etienne)
- Clément RIBES (IMT Atlantique)
- Olivier LAURENT (ENSTA)

Plan

Dealing with uncertainties in Machine Learning

1. Different levels of uncertainties
2. Uncertainties estimation in Deep Learning
3. Validation of uncertainties

Works in collaboration with Jean-Marc MARTINEZ (DES/ISAS/DM2S/STMF/LGLS)
And three interns:

- Mohamed Bahi YAHIAOUI (Mines St Etienne)
- Clément RIBES (IMT Atlantique)
- Olivier LAURENT (ENSTA)

Predictive, aleatoric and epistemic uncertainties

Let x an input point, f_ω a predictive model with parameters ω

Objective: Quantifying the uncertainty on the prediction $f_\omega(x)$

→ **Predictive uncertainty**

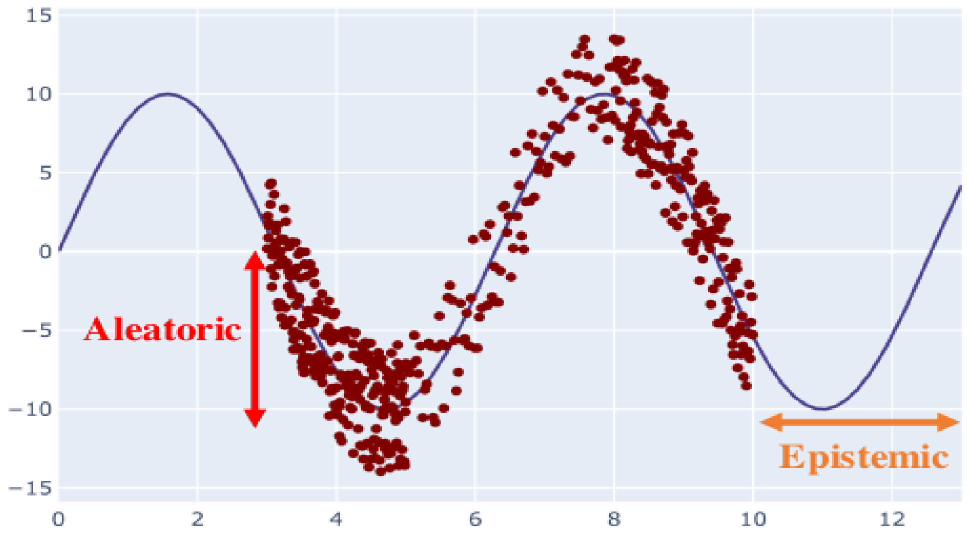
Aleatoric uncertainty

→ Uncertainty related to the data/the phenomenon

Epistemic uncertainty

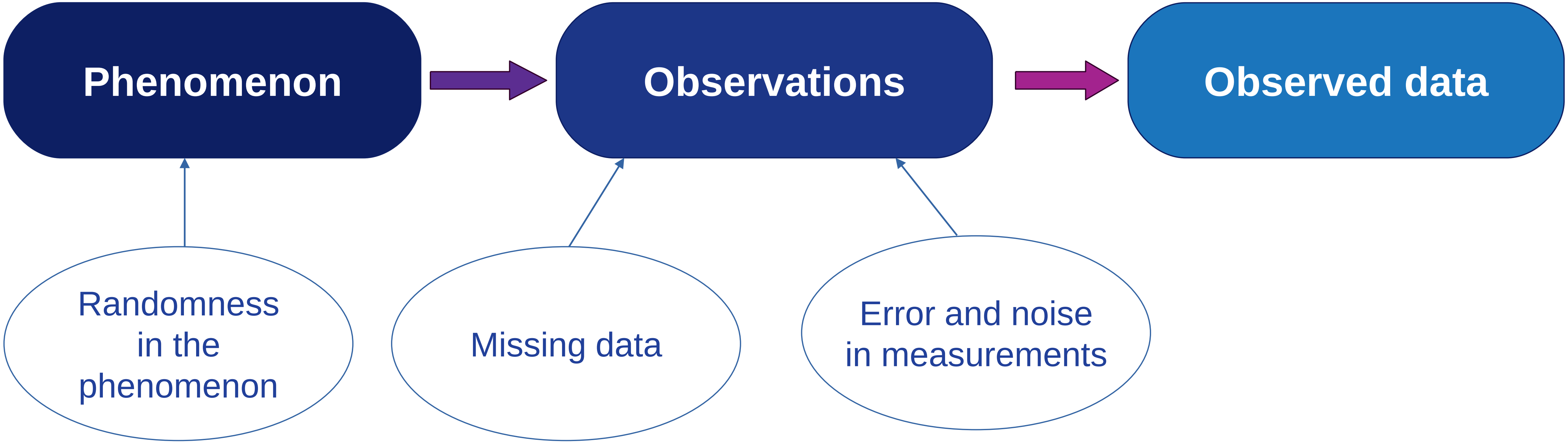
→ Uncertainty related to the model

A review of uncertainty quantification in deep learning: Techniques, applications and challenges, M. Abdar et al., https://doi.org/10.1016/j.inffus.2021.05.008



Visualisation of the two types of uncertainty

Aleatoric uncertainty



Phenomenon

Observations

Observed data

Randomness in the phenomenon

Missing data

Error and noise in measurements

Uncertainty intrinsic within the data, irreducible by improving the model or increasing the dataset
→ **A larger dataset does not reduce aleatoric uncertainty, but it helps to give a better estimation!**

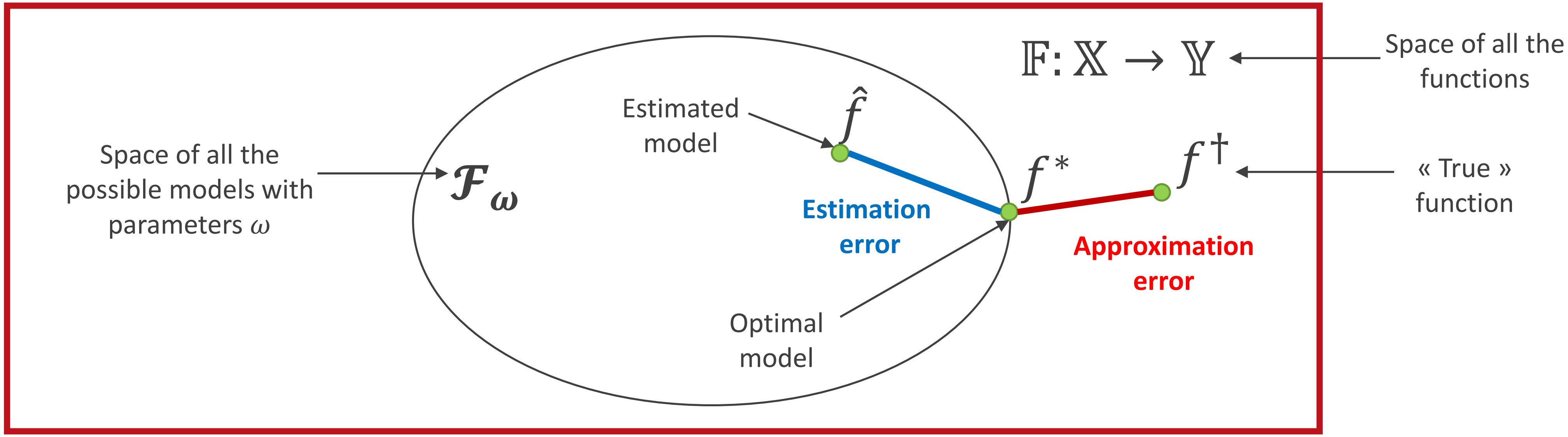
We can reduce the aleatoric uncertainty by improving the measurement (reducing the error or noise) for instance.

Epistemic uncertainty

Represents the lack of « knowledge » or « understanding » of a model on a specific input data point

Two main origins of epistemic uncertainty for machine learning models:

- **Estimation error**: the training dataset is just a sample of all the possible observable data
- **Approximation error**: no model can approximate perfectly the unknown « true » function



It can be possible to reduce epistemic uncertainty by using more data and increasing the model complexity

Probabilistic formulation

x : input data point
ω : model parameters
y^* : possible output
D : Training dataset

Representation of the total **predictive uncertainty** by a probability distribution

$$\boldsymbol{p(y^*|x, D) = \int_{\omega} \underbrace{p(y^*|x, \omega)}_{\textbf{Aleatoric part}} \underbrace{p(\omega|D)}_{\textbf{Epistemic part}} d\omega}$$

Intractable in practice → What quantifications and methods in deep learning?

Plan

Dealing with uncertainties in Machine Learning

1. Different levels of uncertainties
2. Uncertainties estimation in Deep Learning
3. Validation of uncertainties

Works in collaboration with Jean-Marc MARTINEZ (DES/ISAS/DM2S/STMF/LGLS)
And three interns:

- Mohamed Bahi YAHIAOUI (Mines St Etienne)
- Clément RIBES (IMT Atlantique)
- Olivier LAURENT (ENSTA)

Classification problems (basics)

Prediction of a class for an input x among c possible classes, represented by a binary vector $\left(y^{(i)}\right)_{i=1}^{c}$

Conventional representation: a model f_ω predicts the probability that x belongs to each class i:

$$\boldsymbol{f_\omega^{(i)}(x) = p\left(y^{(i)} = 1 \middle| x\right)}$$

Corresponding likelihood: multinoulli distribution (generalized Bernoulli)

$$L(\omega|x) = \prod_{i=1}^{c} p\left(y^{(i)} \middle| x; \omega\right)^{y^{(i)}} = \prod_{i=1}^{c} f_\omega^{(i)}(x)^{y^{(i)}}$$

Mean negative log-likelihood on the whole dataset: **categorical cross-entropy**

$$l(\omega) = -\frac{1}{N} \sum_j \sum_i y_j^{(i)} \log\left(f_\omega^{(i)}(x_j)\right) \rightarrow \text{Learning: } \omega^* = \underset{\omega}{\operatorname{argmin}}\, l(\omega)$$

Uncertainties quantification

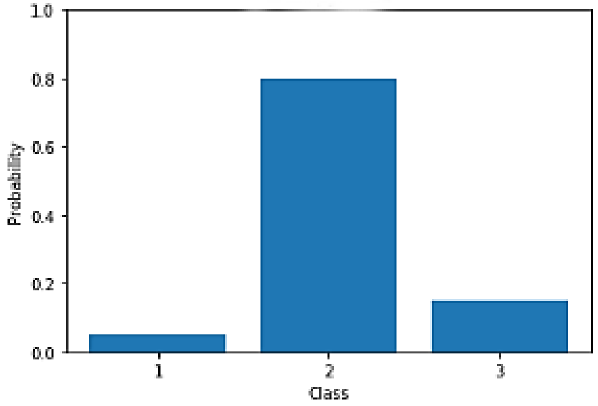
Classification problems: Aleatoric uncertainty

Let f_ω, a model with parameters ω, x an input data

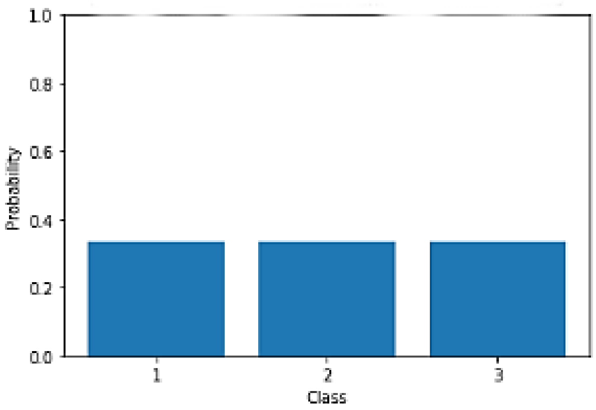
The aleatoric uncertainty can be interpreted as the uncertainty in the decision between each class.

Example with 3 classes:

Low uncertainty



High uncertainty
(fully random)



Quantity to synthetise this knowledge?

Classification problems: Aleatoric uncertainty

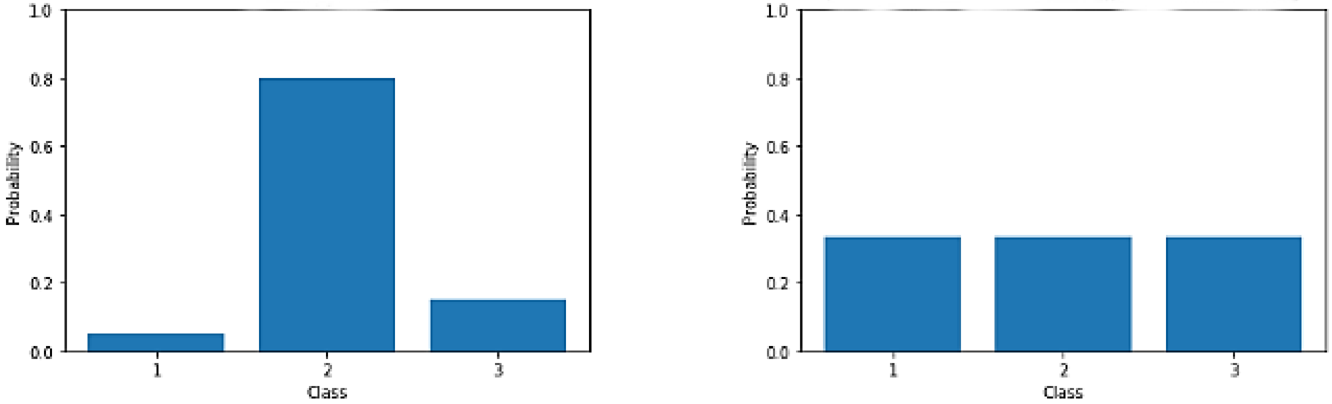
Quantification from information theory: **Shannon entropy**

$$\mathcal{H}\left(p(y|x;\omega)\right) = -\sum_i p\left(y^{(i)}\middle|x;\omega\right)\log\left(p\left(y^{(i)}\middle|x;\omega\right)\right) = -\sum_i f_\omega^{(i)}(x)\log\left(f_\omega^{(i)}(x)\right)$$

Example with 3 classes:

Low uncertainty

High uncertainty (fully random)



Shannon entropy: 0,61 $1{,}1 = \log(3)$ (maximal entropy)

Uncertainties quantification

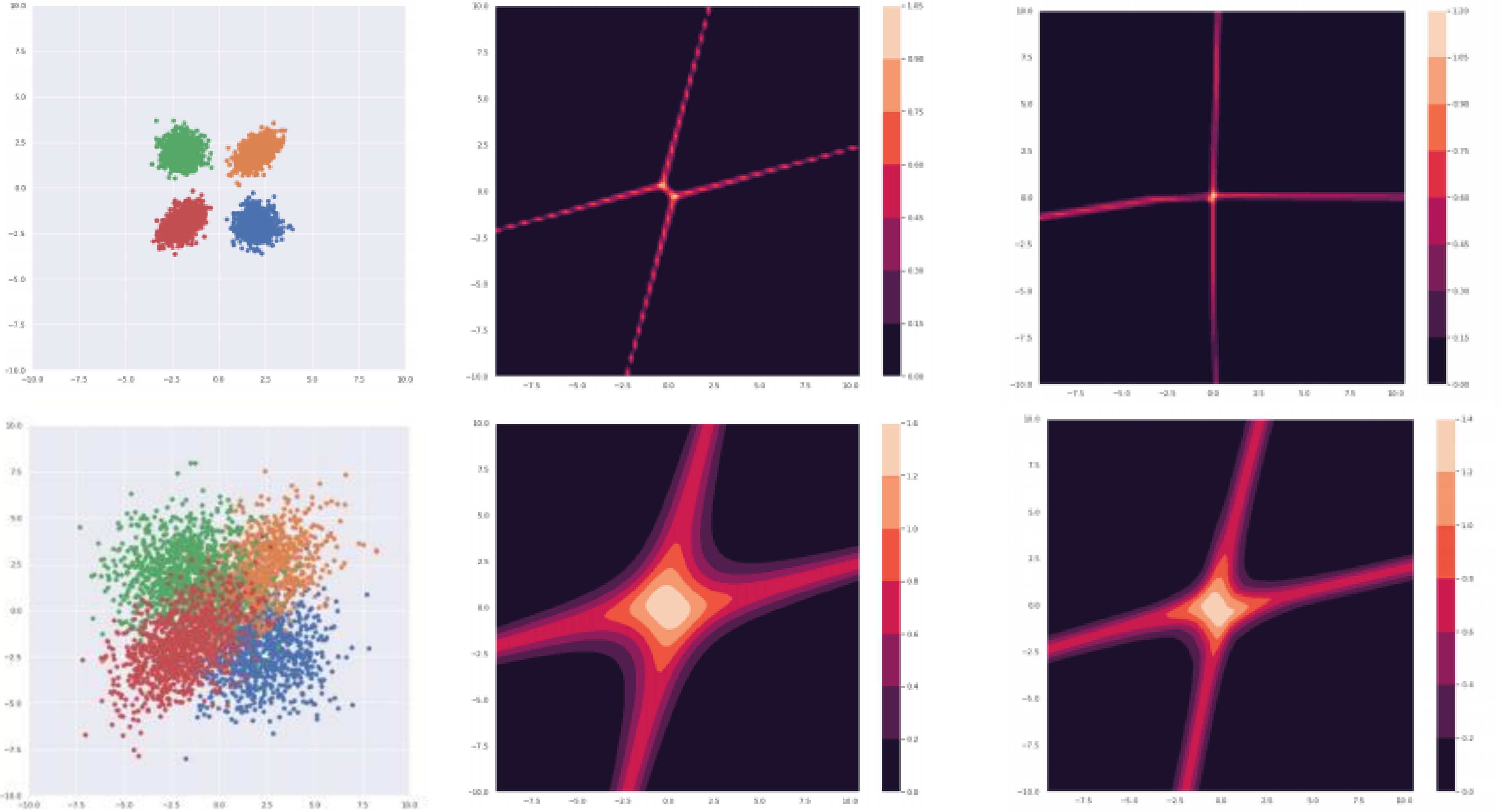
Example

Toy dataset generated by four 2D gaussian distributions

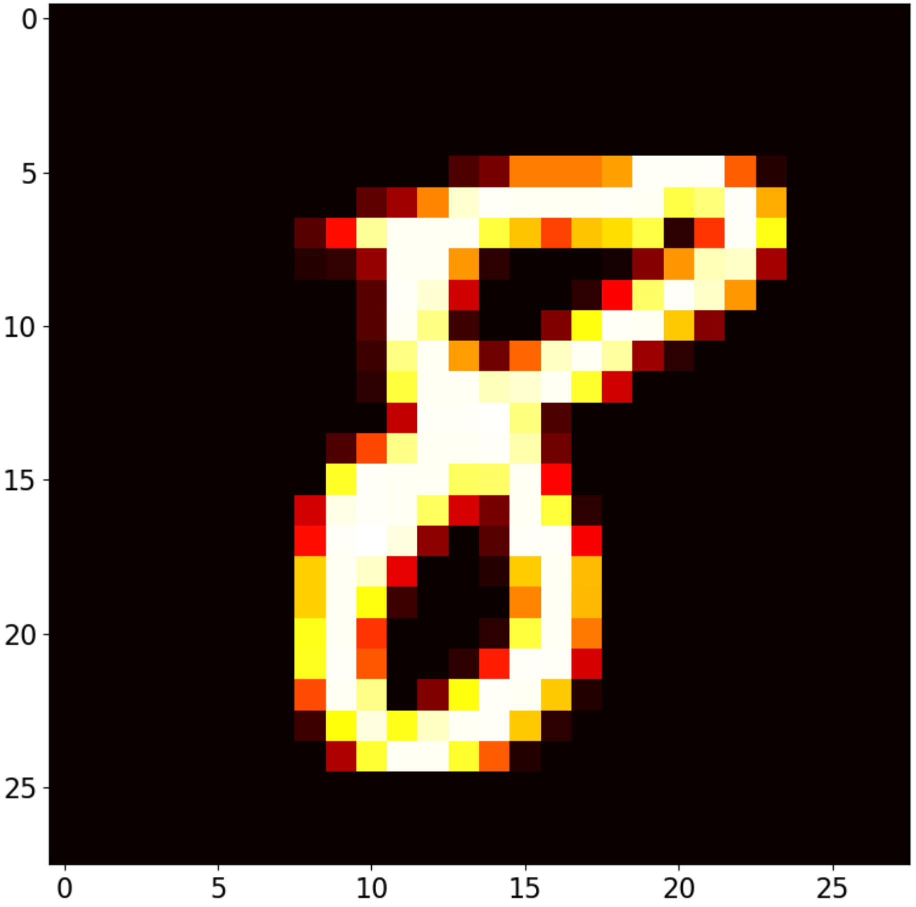
Generated dataset | **Expected entropy (Bayes theorem)** | **Predicted entropy by a neural network**

Separated classes

Overlapped classes

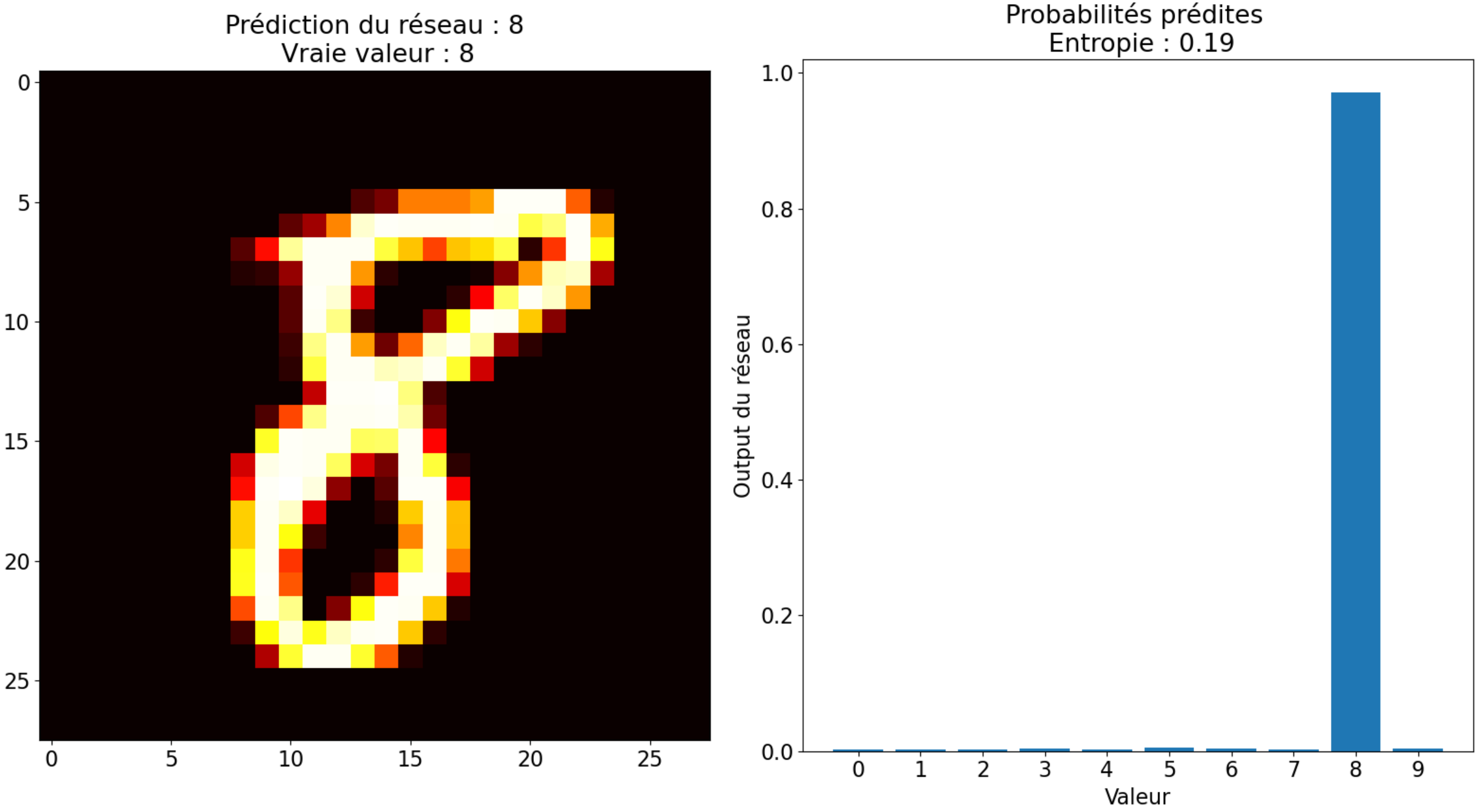


Example with MNIST

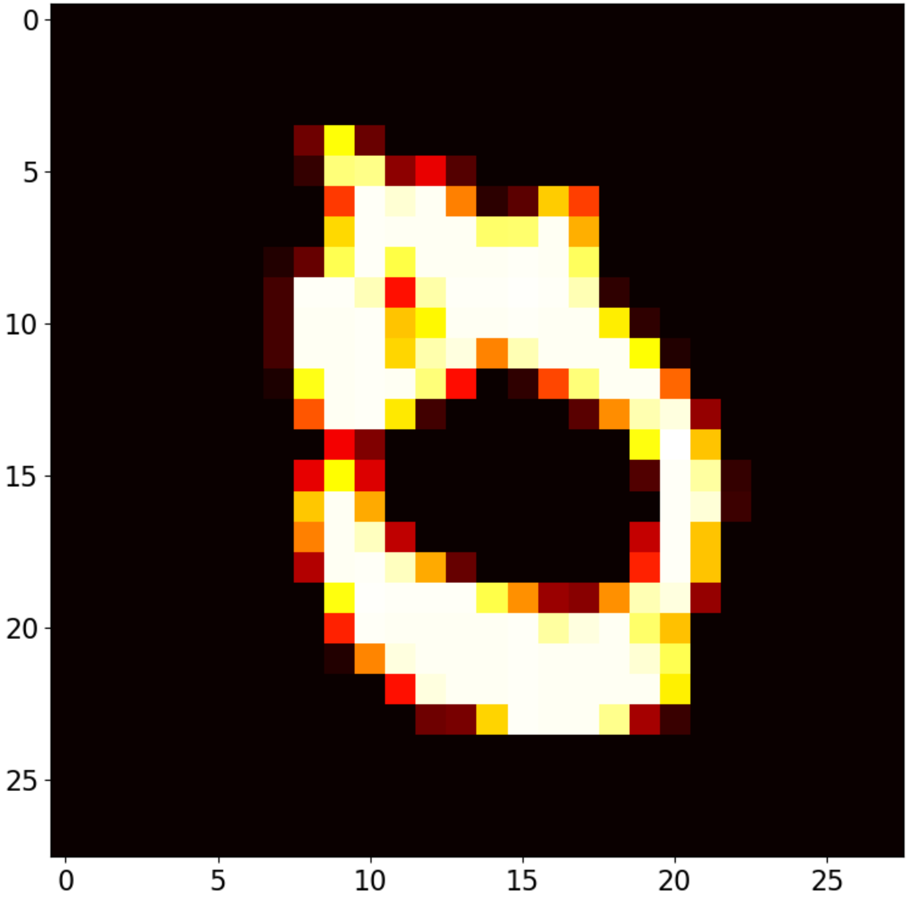


?

Example with MNIST

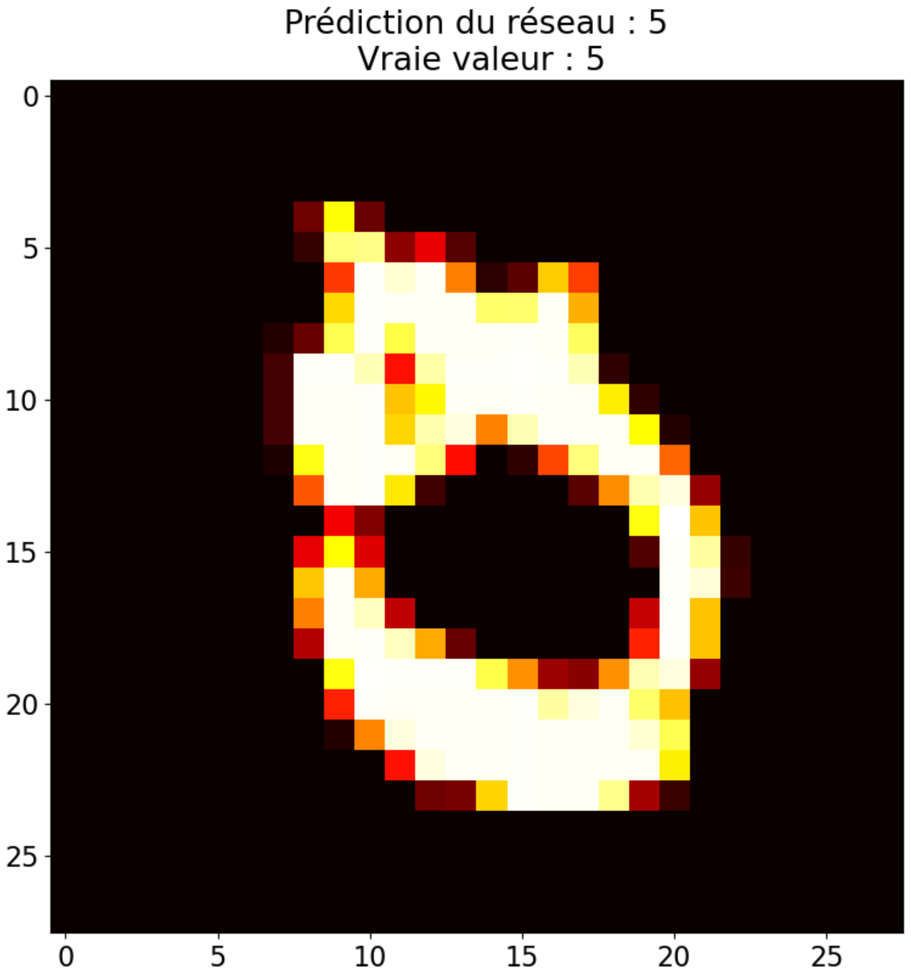


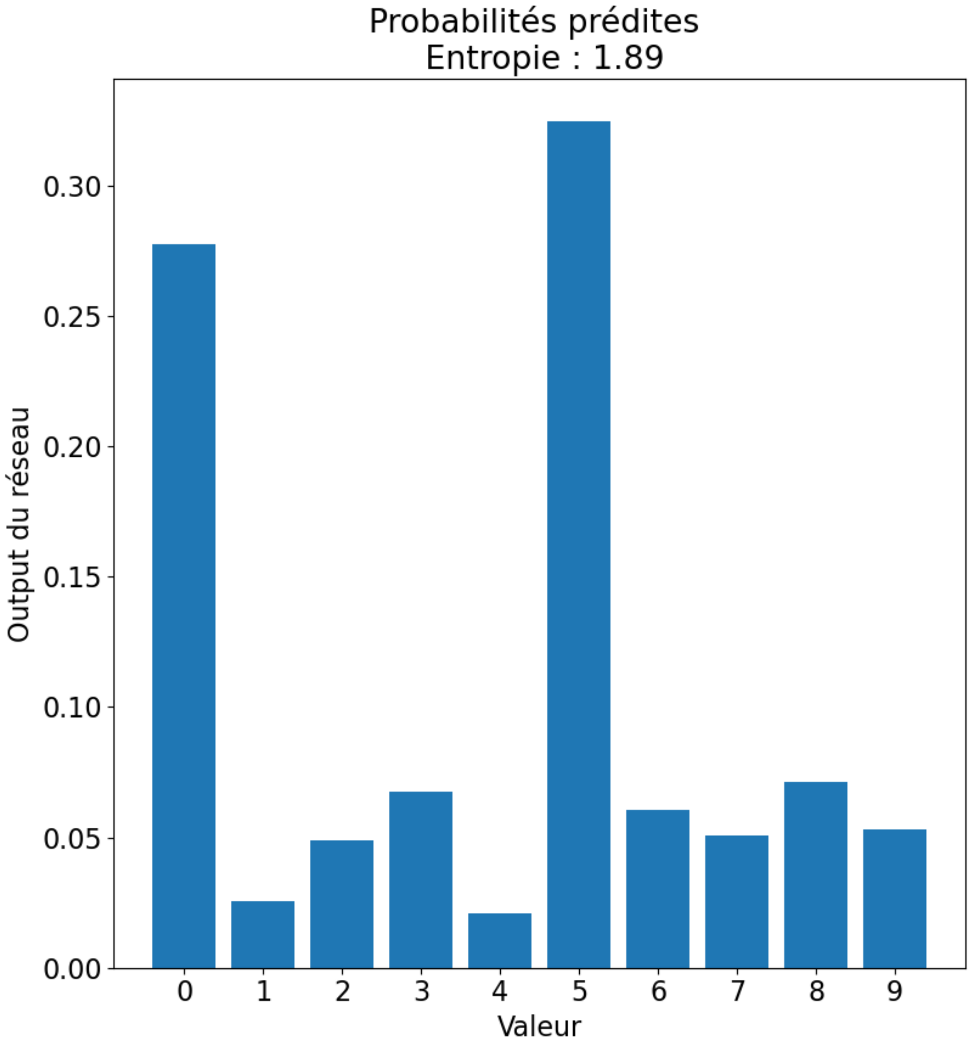
Example with MNIST



?

Example with MNIST





Classifcation problems: epistemic uncertainty

Epistemic uncertainty can be interpreted as the part of uncertainty related to the paramaters of the model.

The network parameters ω (weights) are represented by random variables, following a (unknown) probability distribution $\boldsymbol{p(\omega|D)}$**,** with the training dataset D.

Quantification of the epistemic uncertainty using the mutual information between the outputs y and the network parameters ω, conditionned by the input x and the training set D :

$$\underbrace{\boldsymbol{\mathcal{I}(y;\omega|x;D)}}_{\textbf{Epistemic uncertainty}} = \underbrace{\boldsymbol{\mathcal{H}}\left(\mathbb{E}_{\boldsymbol{\omega \backsim p(\omega|D)}}\boldsymbol{(p(y|x;\omega))}\right)}_{\textbf{Total uncertainty}} - \underbrace{\mathbb{E}_{\boldsymbol{\omega \backsim p(\omega|D)}}\left(\boldsymbol{\mathcal{H}(p(y|x;\omega))}\right)}_{\textbf{Aleatoric uncertainty}}$$

Mutual information = Entropy of the mean prediction – Mean entropy of the predictions

Classification problems: epistemic uncertainty, in practice

Main difficulty: estimation of $p(\omega|D)$ → still an open problem in Deep (and Machine) Learning, numerous methods are proposed that we will present later

Let suppose that $p(\omega|D)$ is known (or estimated at least)
Mutual information is still analytically intractable → use Monte-Carlo sampling

Drawing T samples $\{\omega_1, \omega_2, \dots, \omega_T\}$ from $p(\omega|D)$

Mean prediction approximation: $$\mathbb{E}_{\omega \backsim p(\omega|D)}\left(p\left(y^{(i)}\middle|x;\omega\right)\right) \cong \hat{p}\left(y^{(i)}\middle|x\right) = \frac{1}{T}\sum_t p\left(y^{(i)}\middle|x;\omega_t\right)$$

Total entropy approximation: $$\mathcal{H}\left(\hat{p}(y|x)\right) = -\sum_i \hat{p}\left(y^{(i)}\middle|x\right)\log\left(\hat{p}\left(y^{(i)}\middle|x\right)\right)$$

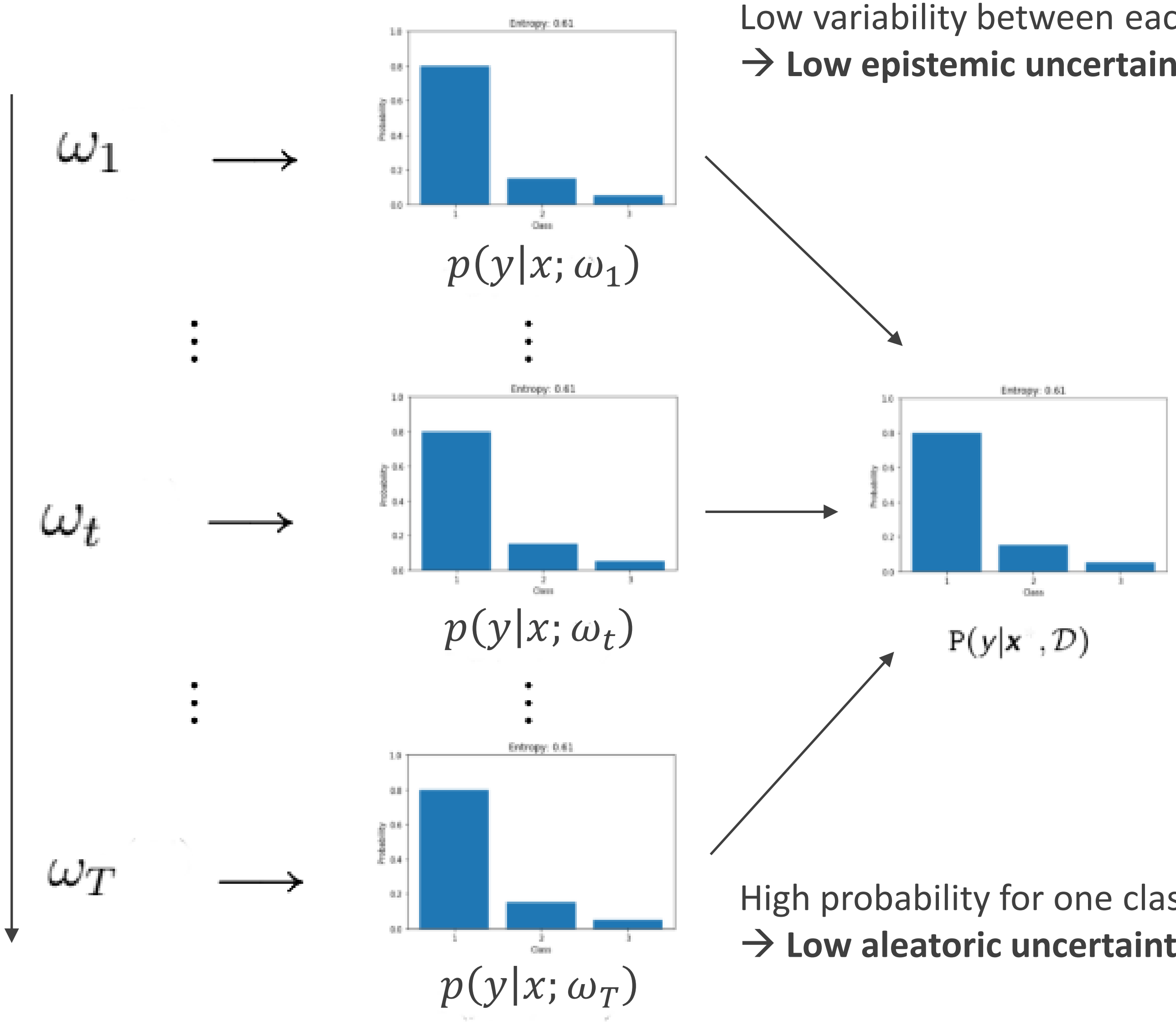
Aleatoric entropy approximation: $$\widehat{\mathcal{H}}\left(p(y|x;\omega)\right) = -\frac{1}{T}\sum_t\sum_i p\left(y^{(i)}\middle|x;\omega_t\right)\log\left(p\left(y^{(i)}\middle|x;\omega_t\right)\right)$$

Mutual information approximation: $$\hat{\mathcal{I}}(y;\omega|x,D) = \mathcal{H}\left(\hat{p}(y|x)\right) - \widehat{\mathcal{H}}\left(p(y|x;\omega)\right)$$

Uncertainties quantification

Intuition

Different samples ω_i



Low variability between each model
→ **Low epistemic uncertainty**

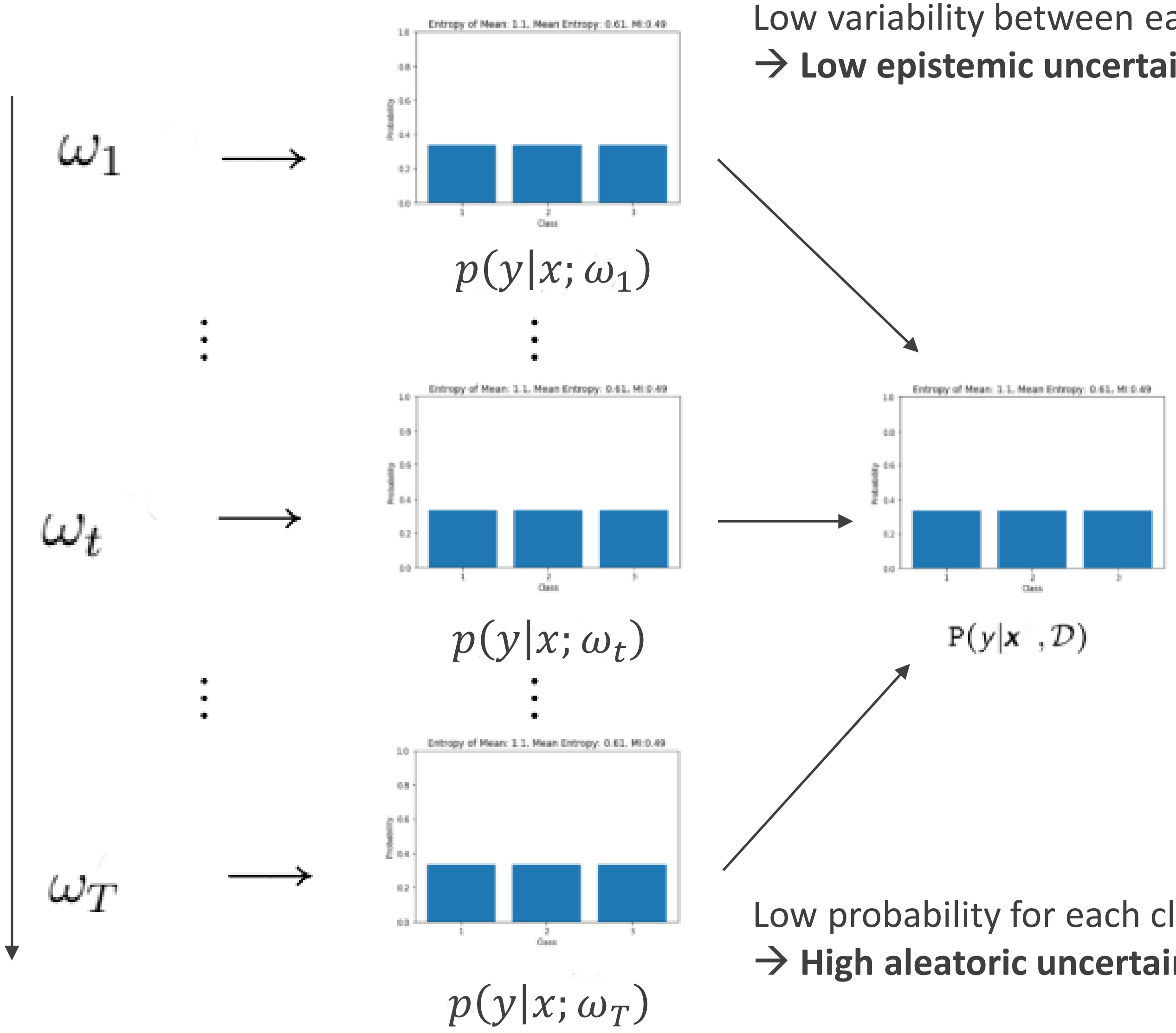
High mean probability for one class
→ **Low total uncertainty**

High probability for one class in each model
→ **Low aleatoric uncertainty**

Uncertainties quantification

Intuition

Different samples ω_i



$p(y|x;\omega_1)$

$p(y|x;\omega_t)$

$p(y|x;\omega_T)$

$\mathrm{P}(y|\mathbf{x}\ ,\mathcal{D})$

Low variability between each model
→ **Low epistemic uncertainty**

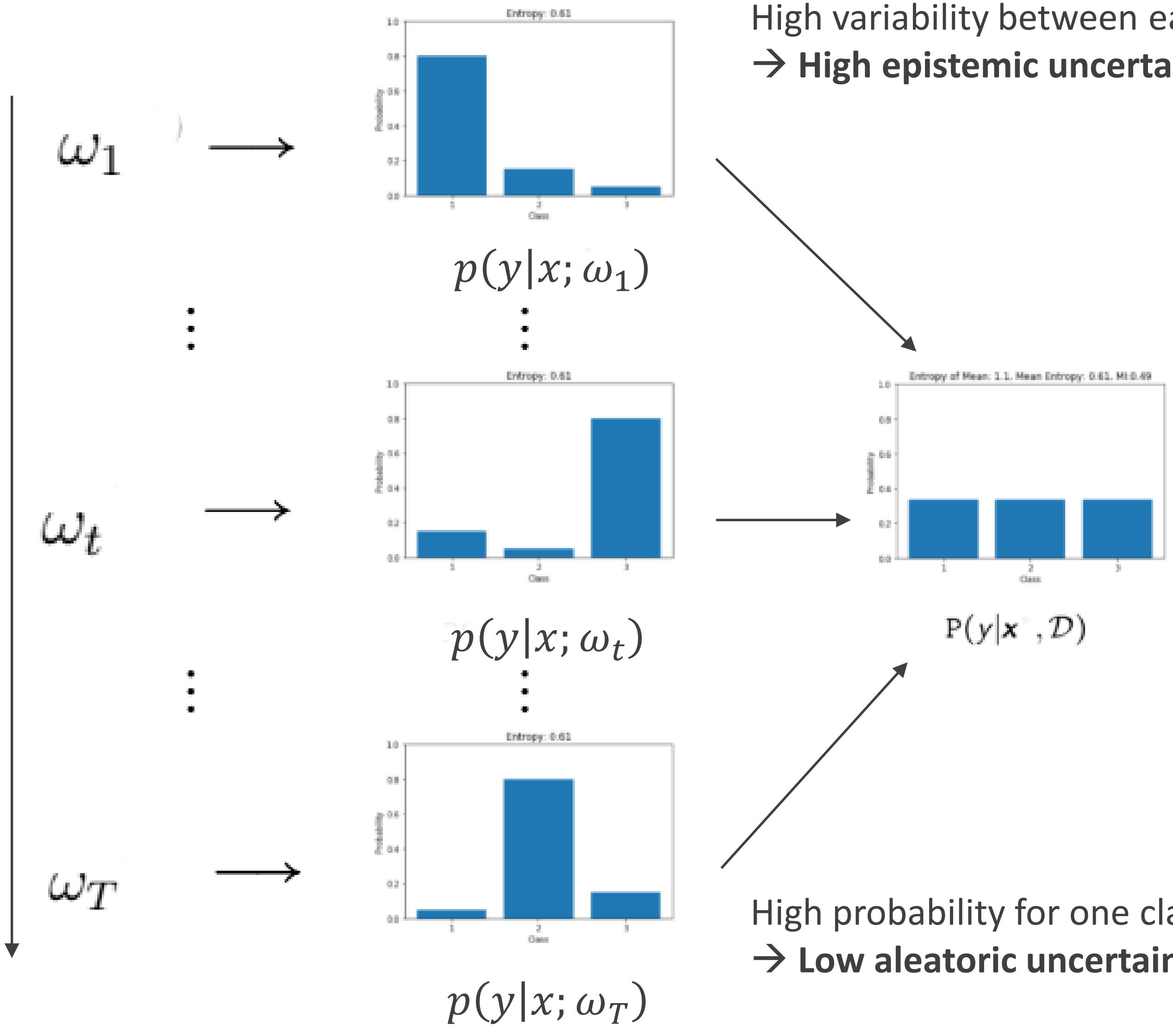
Low mean probability for each class
→ **High total uncertainty**

Low probability for each class in each model
→ **High aleatoric uncertainty**

Uncertainties quantification

Intuition

Différents paramètres tirés



High variability between each model
→ **High epistemic uncertainty**

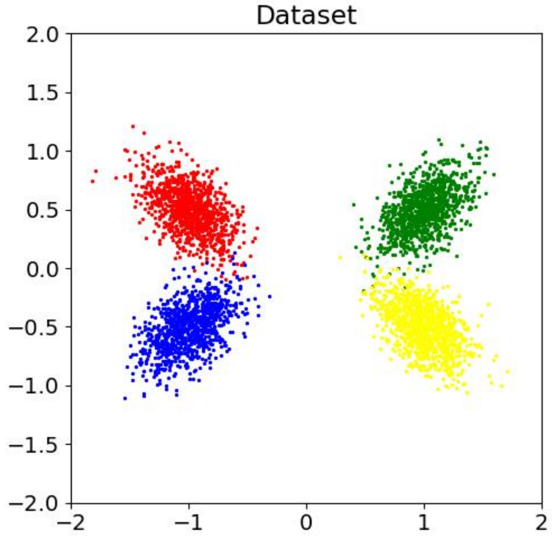
Low mean probability for each class
→ **High total uncertainty**

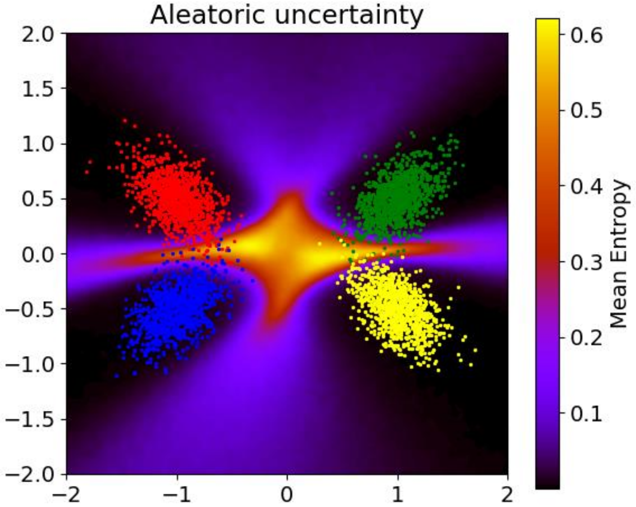
High probability for one class in each model
→ **Low aleatoric uncertainty**

Uncertainties quantification

Example

Back to the toy dataset (four 2D gaussian distributions)

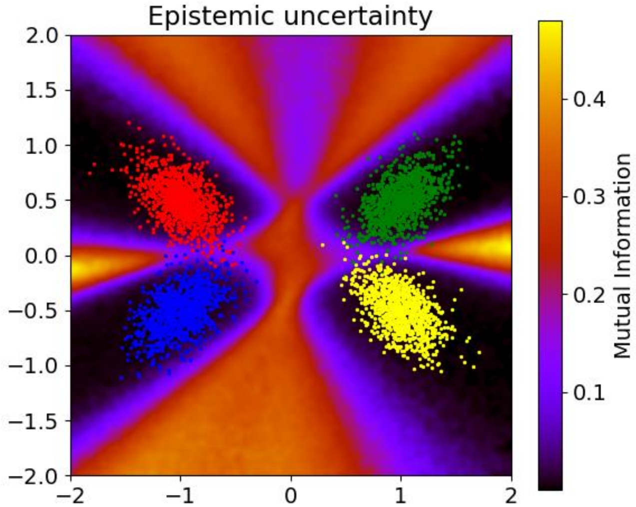

Dataset

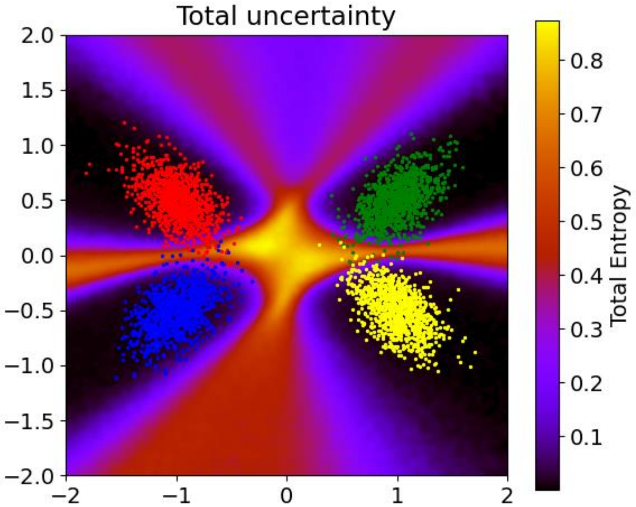

Aleatoric uncertainty

Predicted aleatoric uncertainty

Use of Monte-Carlo Dropout as approximation of bayesian Neural Network

Predicted epistemic uncertainty


Epistemic uncertainty


Total uncertainty

Predicted total uncertainty

Regression problems: conventional approach

Let f_ω a model with parameters ω, $\left(x_j, y_j\right)_j$ training data.

Loss function: Mean Squared Error

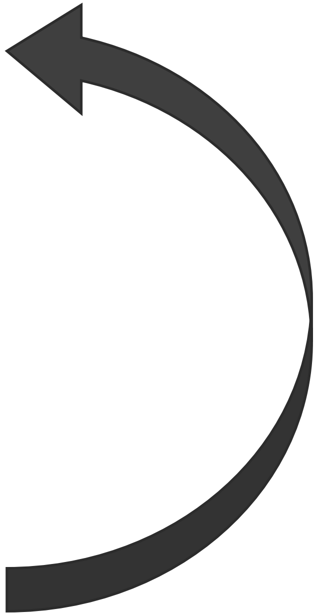
$$\boldsymbol{\omega^* = \underset{\omega}{\mathrm{argmin}} \frac{1}{N} \left\| y_j - f_\omega(x_j) \right\|_2^2}$$

Implicit hypothesis:

- The output data follow a normal distribution $y_j \sim \mathcal{N}\left(\mu_j, \sigma_j^2 \mathbb{I}\right)$
- Data noise is homoscedastic, σ_j is uniform for every example j

These hypothesis lead to the following likelihood (d is the output dimension) :

$$L(\omega) = \prod_j \frac{1}{\left(2\pi\sigma_j^2\right)^{\frac{d}{2}}} \exp\left(-\frac{\left\| y_j - f_\omega(x_j) \right\|_2^2}{2\sigma_j^2}\right)$$



Negative log-likelihood
+ σ_j constant

The estimated mean $\hat{\mu}$ is the output of the model f_ω

Regression problems: aleatoric uncertainty

Density neural networks

We model the aleatoric uncertainty using a distribution $p(y|\theta(x))$ with parameters $\theta(x)$.
The outputs of the neural network correspond to these parameters θ : $f_\omega(x) = \theta(x)$
Remark: we can use one dedicated model (neural network) for each parameter
Objective : minimization of the negative log-likelihood

$$\boldsymbol{\omega}^* = \underset{\boldsymbol{\omega}}{\mathbf{argmin}} - \sum_{\mathbf{j}} \mathbf{log}\left(\boldsymbol{p}\left(\boldsymbol{y_j} | \boldsymbol{f_\omega}(\boldsymbol{x_j})\right)\right)$$

Example: 1D gaussian distribution, $\theta(x) = (\mu(x), \sigma(x)\ )$.
The homoscedasticity hypothesis is no longer required, σ depends on x!

$$p(y|f_\omega(x)) = \frac{1}{\sqrt{2\pi} f_\omega^{(2)}(x)} \exp\left(-\frac{1}{2}\frac{\left(y - f_\omega^{(1)}(x)\right)^2}{f_\omega^{(2)}(x)^2}\right)$$

Negative log-likelihood:

$$\boldsymbol{l}(\boldsymbol{\omega}) = \sum_{\boldsymbol{j}} \mathbf{log}\left(\sqrt{\mathbf{2}\boldsymbol{\pi}}\boldsymbol{f}_{\boldsymbol{\omega}}^{(\mathbf{2})}(\boldsymbol{x_j})\right) + \frac{\mathbf{1}}{\mathbf{2}}\frac{\left(\boldsymbol{y_j} - \boldsymbol{f}_{\boldsymbol{\omega}}^{(\mathbf{1})}(\boldsymbol{x_j})\right)^{\mathbf{2}}}{\boldsymbol{f}_{\boldsymbol{\omega}}^{(\mathbf{2})}(\boldsymbol{x_j})^{\mathbf{2}}}$$

Remarks:
- The « real » standard deviation is not required in the training dataset
- Other distributions can be considered, but could raise convergence problems (such as divergence of the likelihood)

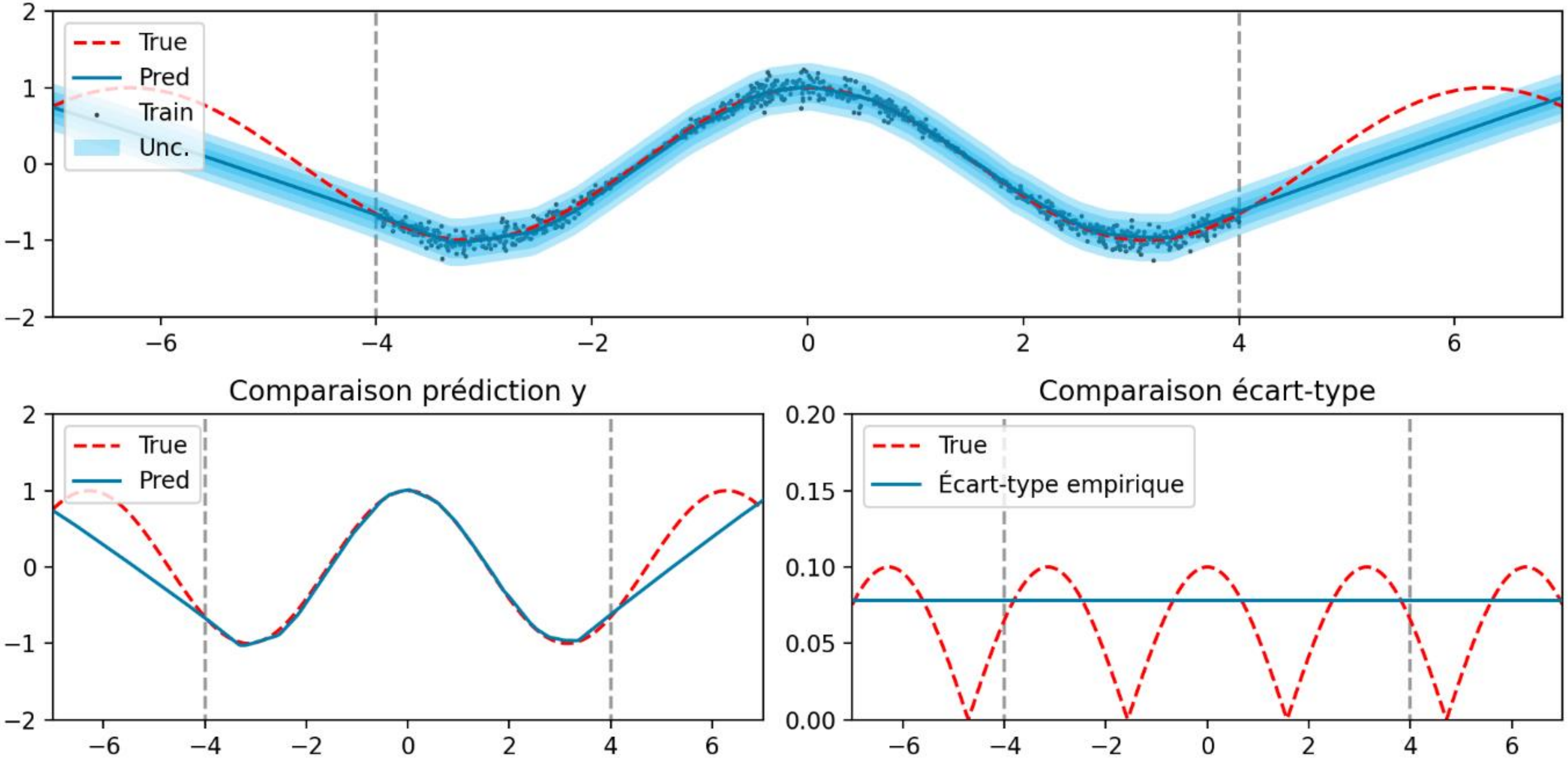
Regression problem: aleatoric uncertainty

Density neural networks: example

Generated data with heteroscedastic gaussian noise

Conventional appraoch:

- Least squares estimation
- Empirical estimation of the standard deviation, uniform regardless the input

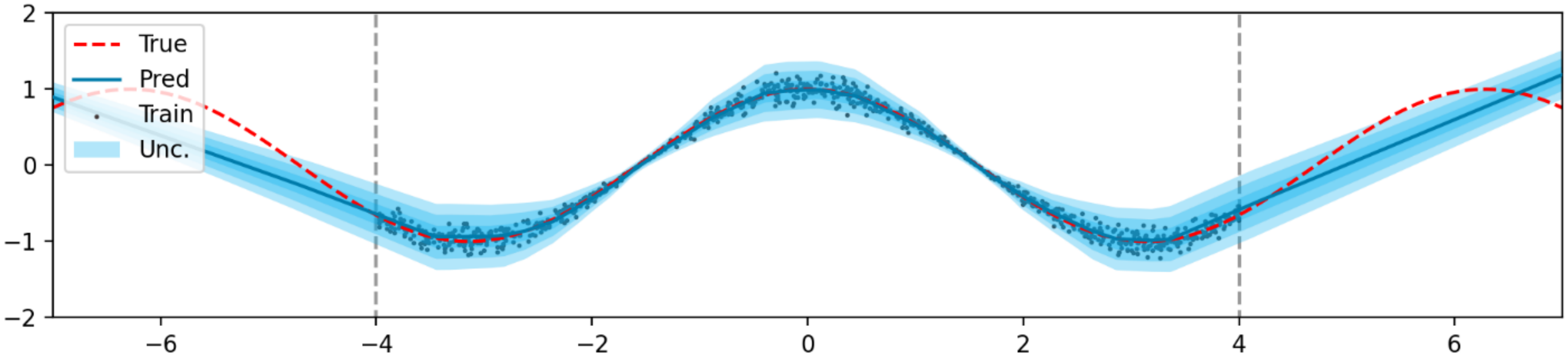


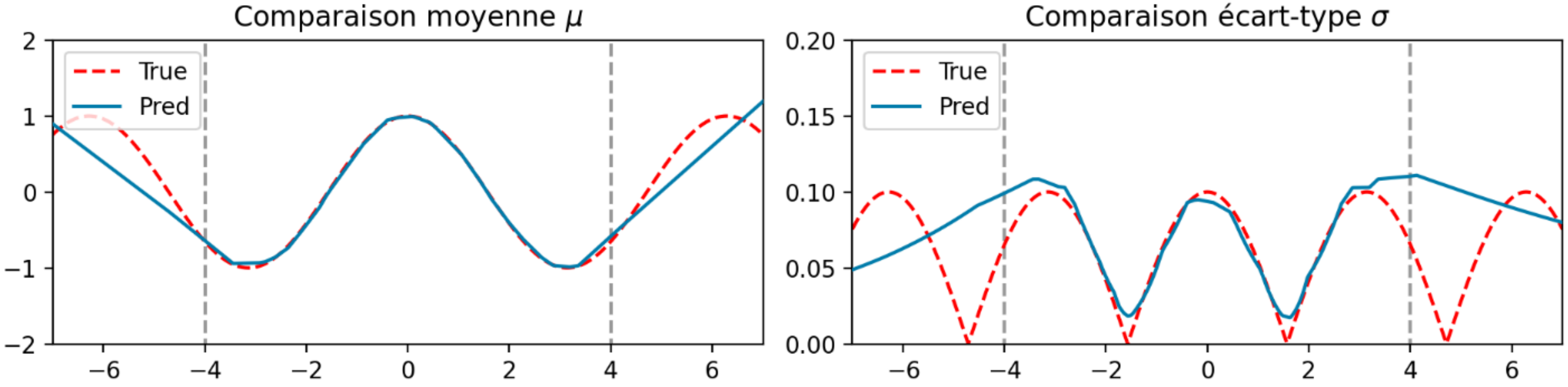
Regression problems: aleatoric uncertainty

Density neural networks: example

Generated data with heteroscedastic gaussian noise

Density neural networks




Comparaison moyenne μ

Comparaison écart-type σ

Uncertainties quantification

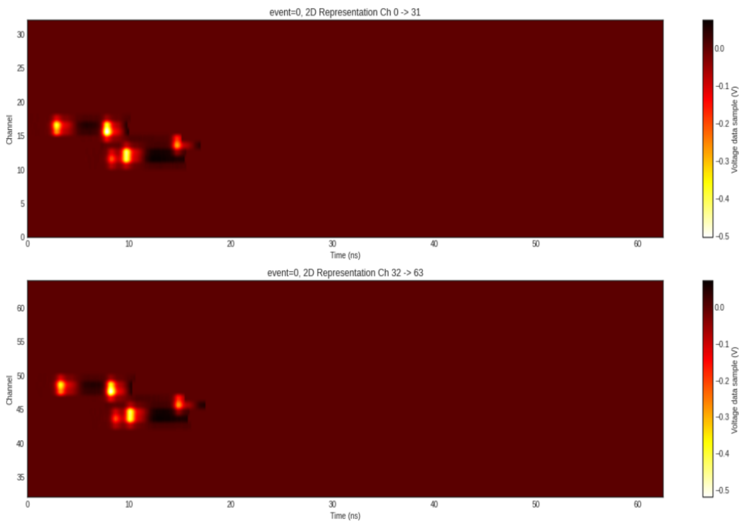
Regression problems: aleatoric uncertainty

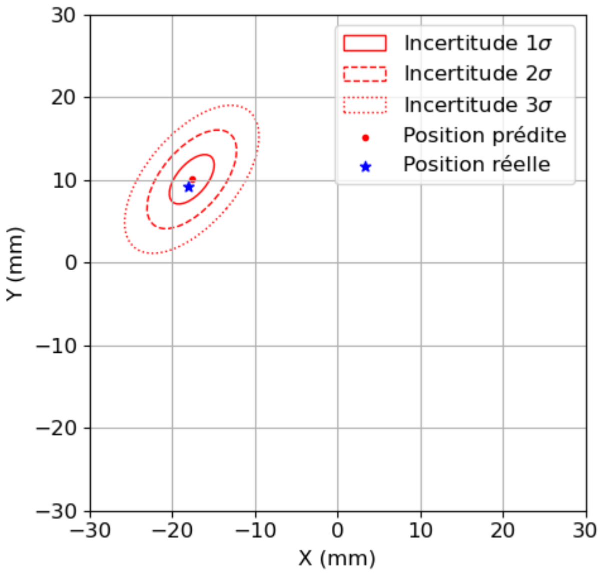
Density neural networks: application to a detector for medical imaging
Transverse project (DRF/Irfu, DRF/SHFJ, DES/LGLS) within the ClearMind project
See Chi-Hsun Sung's talk, *Machine Learning Algorithms for the Gamma Conversion Reconstruction in the ClearMind Project*

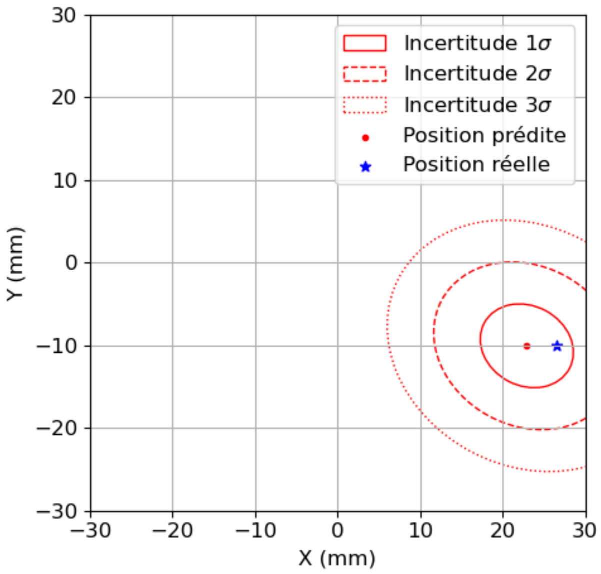
Input: Electrical impulse shapes recorded by the detector (high dimension 400x64)
Output: Gamma interaction position

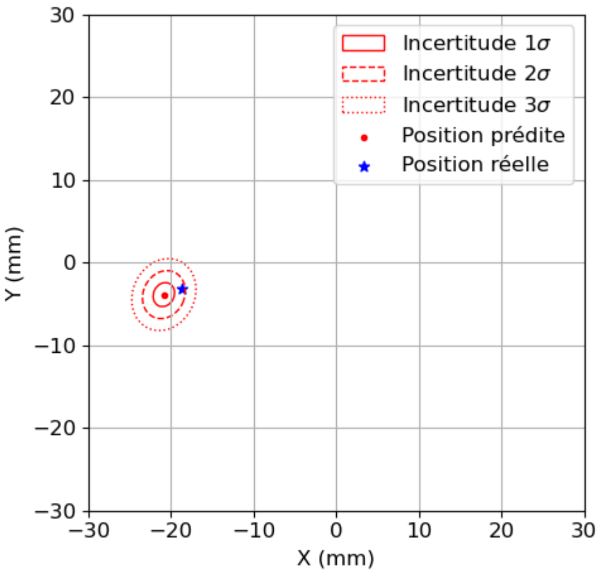
Density neural network with a 2D gaussian distribution (with a correlation factor)

Input example









Regression problems: aleatoric uncertainty

Quantile regression: a non-parametric approach

For an expected output $y(x)$, the quantile $q_\alpha(x)$ with probability α is the value such as:

$$p\big(q_\alpha(x) \geq y(x)\big) = \alpha$$

Quantile regression: prediction of the quantile, agnostically to the density of y

Loss function: Pinball loss, for a probability level α and $f_\omega(x)$ is trained to represent $q_\alpha(x)$

$$\boldsymbol{\omega}^* = \underset{\boldsymbol{\omega}}{\mathbf{argmin}} \frac{\mathbf{1}}{\boldsymbol{N}} \sum_{\boldsymbol{x} \in \boldsymbol{D}} \mathbf{max}\Big(\boldsymbol{\alpha}\big(\boldsymbol{y}(\boldsymbol{x}) - \boldsymbol{f}_{\boldsymbol{\omega}}(\boldsymbol{x})\big), (\mathbf{1} - \boldsymbol{\alpha})\big(\boldsymbol{f}_{\boldsymbol{\omega}}(\boldsymbol{x}) - \boldsymbol{y}(\boldsymbol{x})\big)\Big)$$

Advantages:
- No model of the expected noise (aleatoric uncertainty) is necessary
- Multi-modal distributions can be approximated

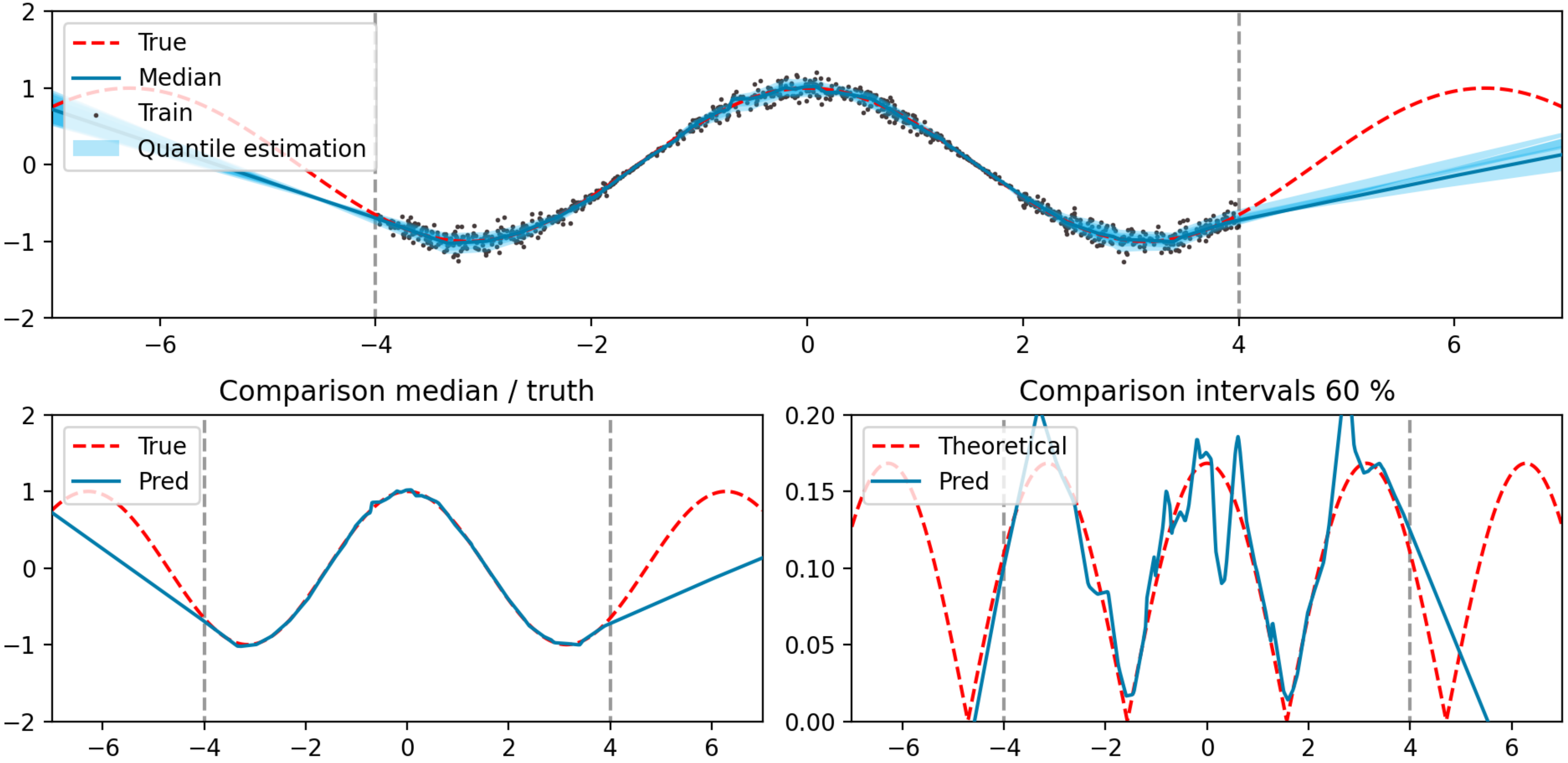
Limitations:
- Large amount of data required for a correct estimation of extreme quantiles
- Local or global interpretation of quantiles?

→ Study on conformalized quantile regression to try to get reliable quantile estimations

Regression problems: aleatoric uncertainty

Quantile regression: Example

Generated data with heteroscedastic gaussian noise



Regression problems: epistemic uncertainty

Same interpretation as classification: dependance of the output on the parameters ω with a distribution $p(\omega|D)$

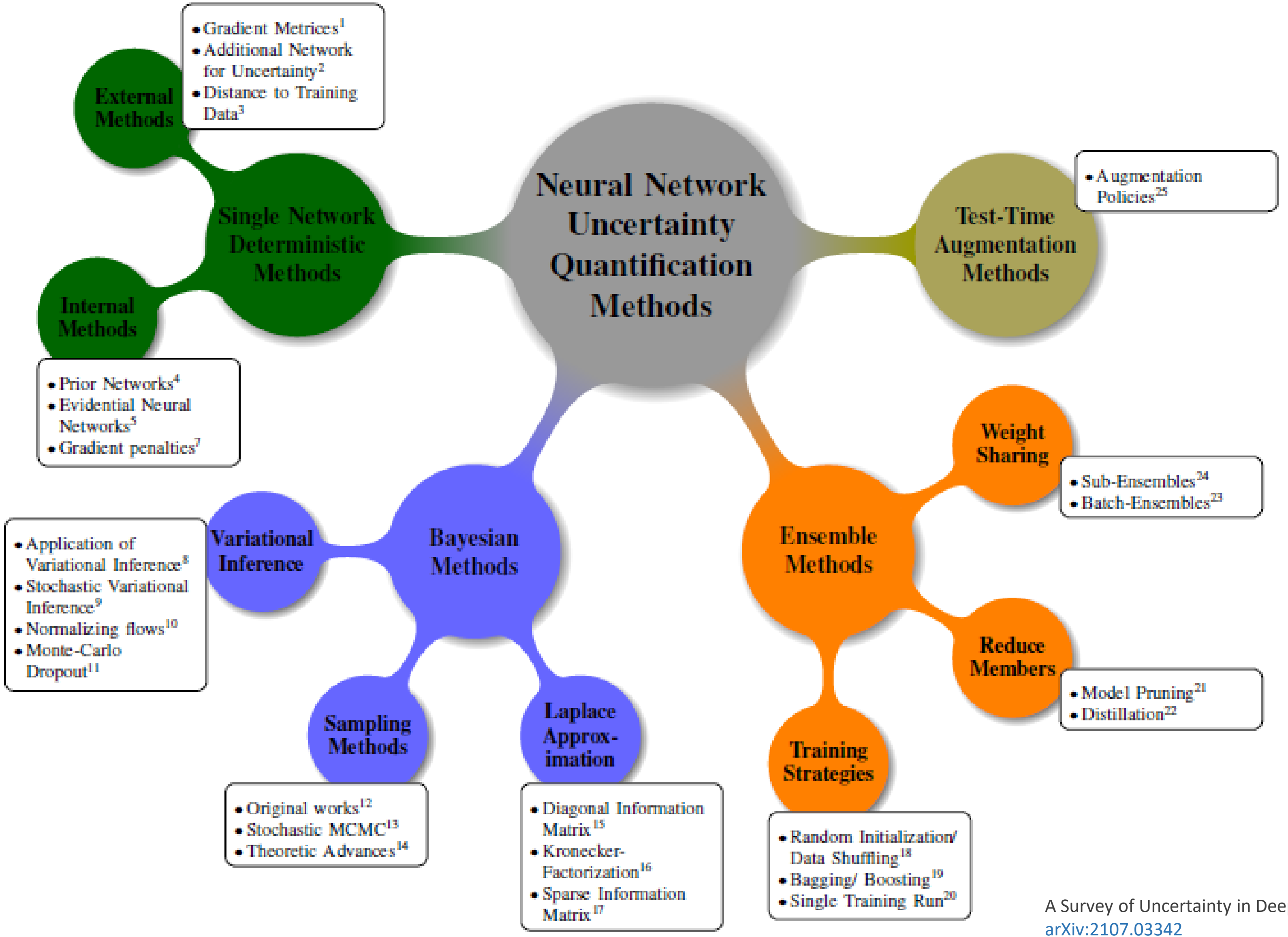
Same main difficulty as classification: estimation of $p(\omega|D)$

Quantity to characterize the epistemic uncertainty in regression: **variance**

$$\underbrace{\mathbb{V}(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{D})}_{\text{Total uncertainty}} = \underbrace{\mathbb{V}_{\boldsymbol{\omega}\sim \boldsymbol{p}(\boldsymbol{\omega}|\boldsymbol{D})}\left(\mathbb{E}_{\boldsymbol{y}\sim \boldsymbol{p}\left(\boldsymbol{y}|\boldsymbol{f}_{\boldsymbol{\omega}}(\boldsymbol{x})\right)}(\boldsymbol{y})\right)}_{\text{Epistemic uncertainty}} + \underbrace{\mathbb{E}_{\boldsymbol{\omega}\sim \boldsymbol{p}(\boldsymbol{\omega}|\boldsymbol{D})}\left(\mathbb{V}_{\boldsymbol{y}\sim \boldsymbol{p}\left(\boldsymbol{y}|\boldsymbol{f}_{\boldsymbol{\omega}}(\boldsymbol{x})\right)}(\boldsymbol{y})\right)}_{\text{Aleatoric uncertainty}}$$

Differential entropy (extension of Shannon entropy to continuous cases) instead of variance?

Numerous approaches in Deep Learning



Neural Network Uncertainty Quantification Methods

Single Network Deterministic Methods

External Methods
- Gradient Metrices[1]
- Additional Network for Uncertainty[2]
- Distance to Training Data[3]

Internal Methods
- Prior Networks[4]
- Evidential Neural Networks[5]
- Gradient penalties[7]

Test-Time Augmentation Methods
- Augmentation Policies[25]

Bayesian Methods

Variational Inference
- Application of Variational Inference[8]
- Stochastic Variational Inference[9]
- Normalizing flows[10]
- Monte-Carlo Dropout[11]

Sampling Methods
- Original works[12]
- Stochastic MCMC[13]
- Theoretic Advances[14]

Laplace Approx-imation
- Diagonal Information Matrix[15]
- Kronecker-Factorization[16]
- Sparse Information Matrix[17]

Ensemble Methods

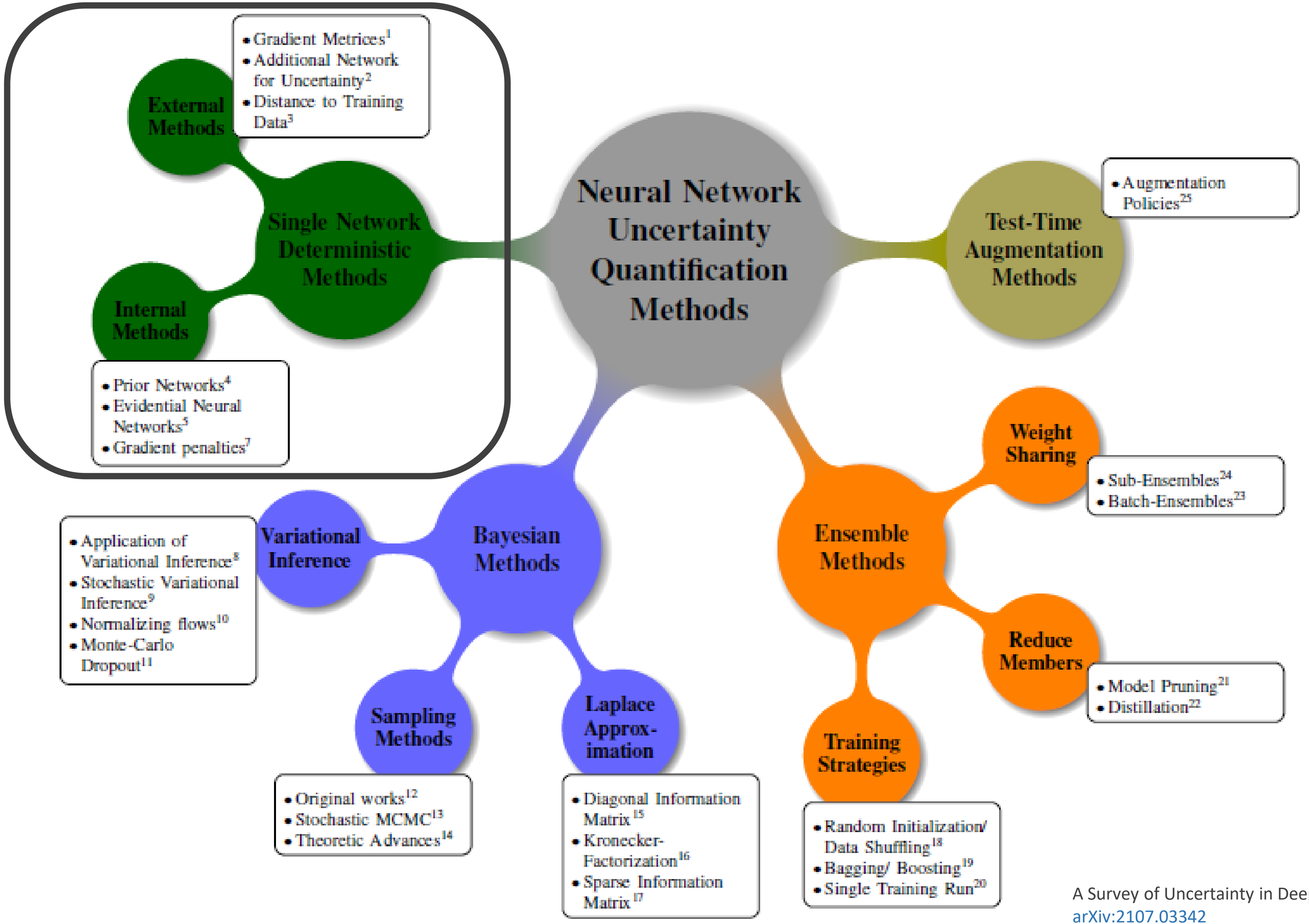
Weight Sharing
- Sub-Ensembles[24]
- Batch-Ensembles[23]

Reduce Members
- Model Pruning[21]
- Distillation[22]

Training Strategies
- Random Initialization/ Data Shuffling[18]
- Bagging/ Boosting[19]
- Single Training Run[20]

A Survey of Uncertainty in Deep Neural Networks, J. Gawlikowski et al,, arXiv:2107.03342

Numerous approaches in Deep Learning



Neural Network Uncertainty Quantification Methods

Single Network Deterministic Methods

External Methods
- Gradient Metrices[1]
- Additional Network for Uncertainty[2]
- Distance to Training Data[3]

Internal Methods
- Prior Networks[4]
- Evidential Neural Networks[5]
- Gradient penalties[7]

Test-Time Augmentation Methods
- Augmentation Policies[25]

Bayesian Methods

Variational Inference
- Application of Variational Inference[8]
- Stochastic Variational Inference[9]
- Normalizing flows[10]
- Monte-Carlo Dropout[11]

Sampling Methods
- Original works[12]
- Stochastic MCMC[13]
- Theoretic Advances[14]

Laplace Approx-imation
- Diagonal Information Matrix[15]
- Kronecker-Factorization[16]
- Sparse Information Matrix[17]

Ensemble Methods

Weight Sharing
- Sub-Ensembles[24]
- Batch-Ensembles[23]

Reduce Members
- Model Pruning[21]
- Distillation[22]

Training Strategies
- Random Initialization/ Data Shuffling[18]
- Bagging/ Boosting[19]
- Single Training Run[20]

A Survey of Uncertainty in Deep Neural Networks, J. Gawlikowski et al,, arXiv:2107.03342

Numerous approaches in Deep Learning

Single Network Deterministic methods :

The name is explicit: using one single neural network with a deterministic prediction

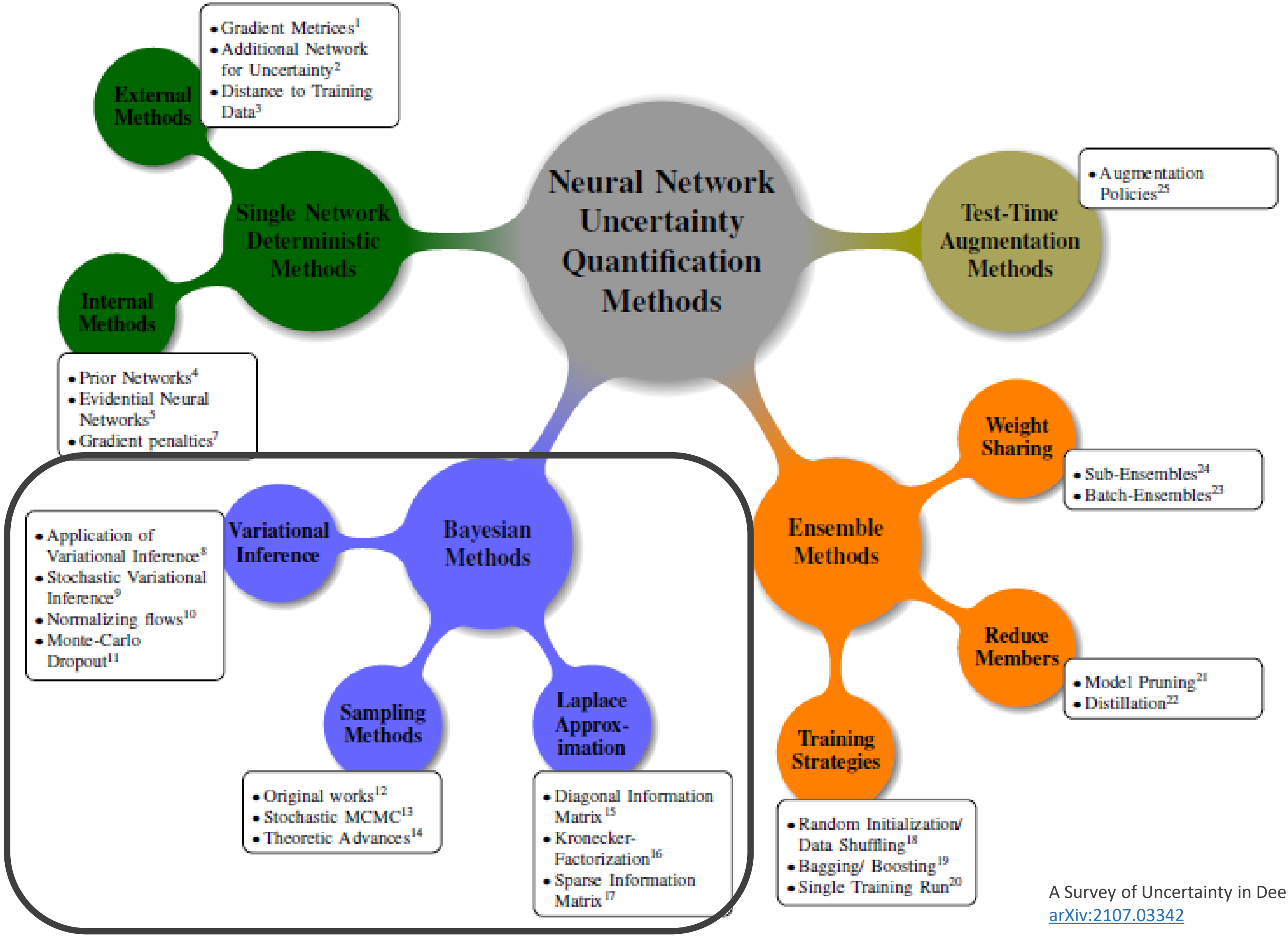
Examples:

- **Additionnal Network for Uncertainty**: in classification problems, we have several answers for a single input point. For instance, several doctors give possibly different diagnostics for a same image, we compute the corresponding frequency of positive and negative answers and the networks is trained on these frequencies.

- **Evidential Neural Network**: extension of Density Neural Networks with a probability distribution on the parameters of the probability distribution

  Example :
  - Gaussian distribution on the output: $y \sim \mathcal{N}(\mu, \sigma^2)$
  - The parameters μ et σ are considered as random variables:
    - $\mu \sim \mathcal{N}(\gamma, v^{-1}\sigma)$
    - $\sigma \sim \Gamma^{-1}(\alpha, \beta)$
  - The neural network estimates the four parameters $(\gamma, v, \alpha, \beta)$
  - The direct prediction is given by $\mathbb{E}(\mu) = \gamma$
  - The aleatoric uncertainty is given by $\mathbb{E}(\sigma^2) = \frac{\beta}{\alpha - 1}$ and the epistemic uncertatinty $\mathbb{V}(\mu) = \frac{\beta}{v(\alpha - 1)}$
  - Interesting formalism but some issues in practice, especially for Out-of-distribution points

Numerous approaches in Deep Learning



A Survey of Uncertainty in Deep Neural Networks, J. Gawlikowski et al,, arXiv:2107.03342

Numerous approaches in Deep Learning

Bayesian approaches:

Main idea: achieving a sampling of ω according to the unknown distribution $p(\omega|D)$

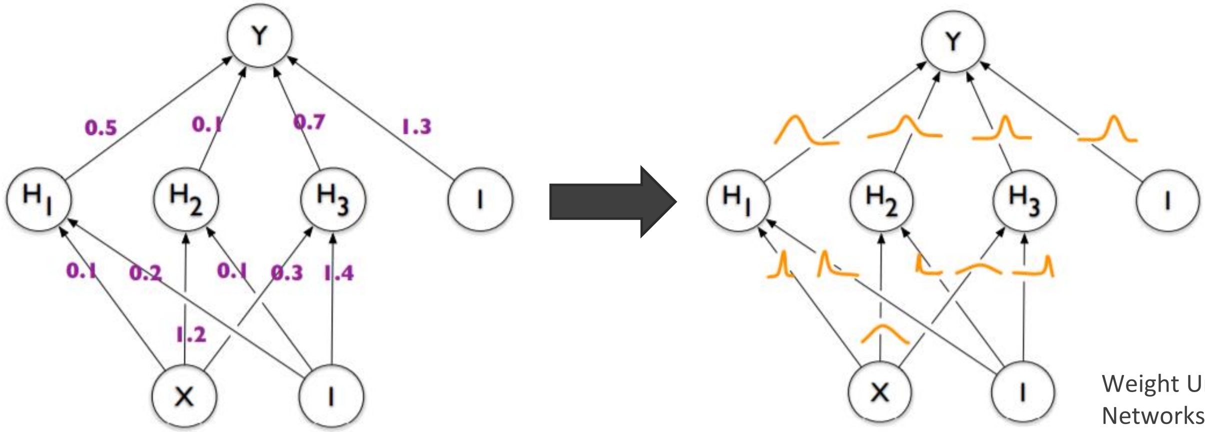
Several approaches:

Bayes by backprop: variational inference

$p(\omega|D)$ is intractable

→ Approximation by a distribution $q(\omega|\Theta)$

For instance : $\Theta_l = (\mu_l, \sigma_l)$ parameters of a gaussian



Weight Uncertainty in Neural Networks, Blundell et al.

Objective function:

$$\Theta^* = \underset{\Theta}{\text{argmin}}\,\text{KL}[q(\omega|\Theta)||p(\omega|D)]$$

KL: Kullback-Leibler divergence

$$\Leftrightarrow \boldsymbol{\Theta^*} = \underset{\boldsymbol{\Theta}}{\textbf{argmin}}\,\textbf{KL}[\boldsymbol{q(\omega|\Theta)}||\boldsymbol{p(\omega)}] - \mathbb{E}_{\boldsymbol{q(\omega|\Theta)}}[\textbf{log}(\boldsymbol{p(D|\omega)})]$$

ELBO : Evidence Lower Bound

Regularization term

Prior $p(\omega)$ to be defined

→ Limitation?

Likelihood maximization

Monte-Carlo sampling for estimation

→ Limitations in the training procedure

Numerous approaches in Deep Learning

Bayesian approaches :

Main idea: achieving a sampling of ω according to the unknown distribution $p(\omega|D)$

Several approaches:

Monte-Carlo Dropout

Deactivation or noise on neurons or weights
→ Conventional method for regularization
→ Applied at test time

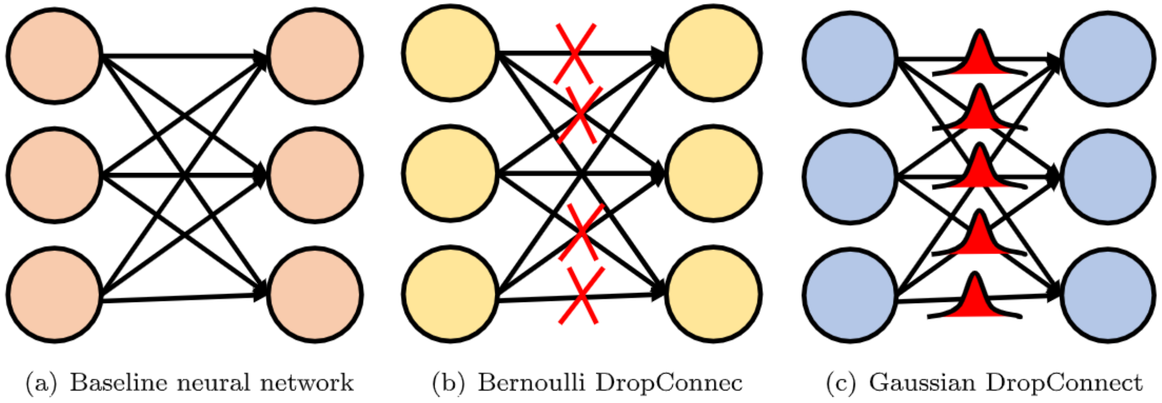
Deactivation of weights: each component of ω follows a Bernoulli distribution

Gaussian noise: each componant of ω follow a gaussian distribution, we optimize the mean (variance is fixed)

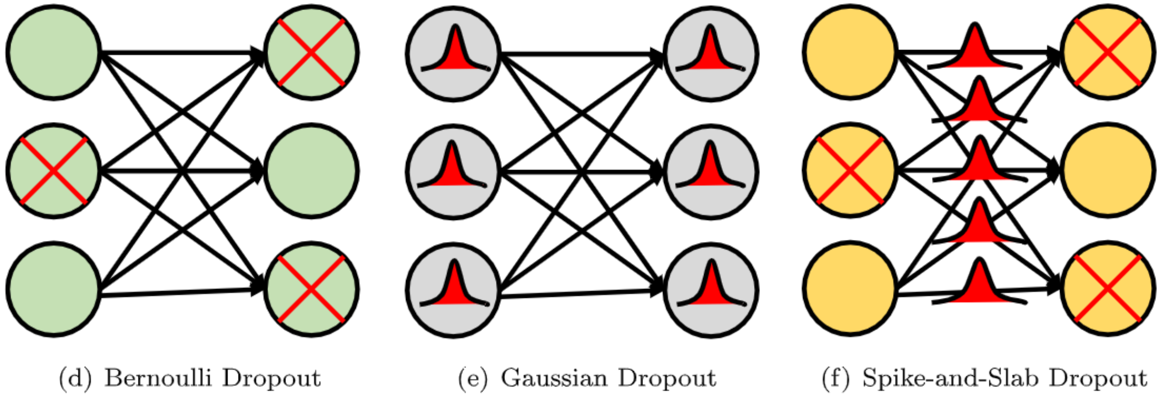
Prior $p(\omega)$ chosen as $\mathcal{N}(0, l)$ → L_2-regularization

Limitations:

- Weak theoretical guarantees
- Dropout rate/Variance of the gaussian noise?

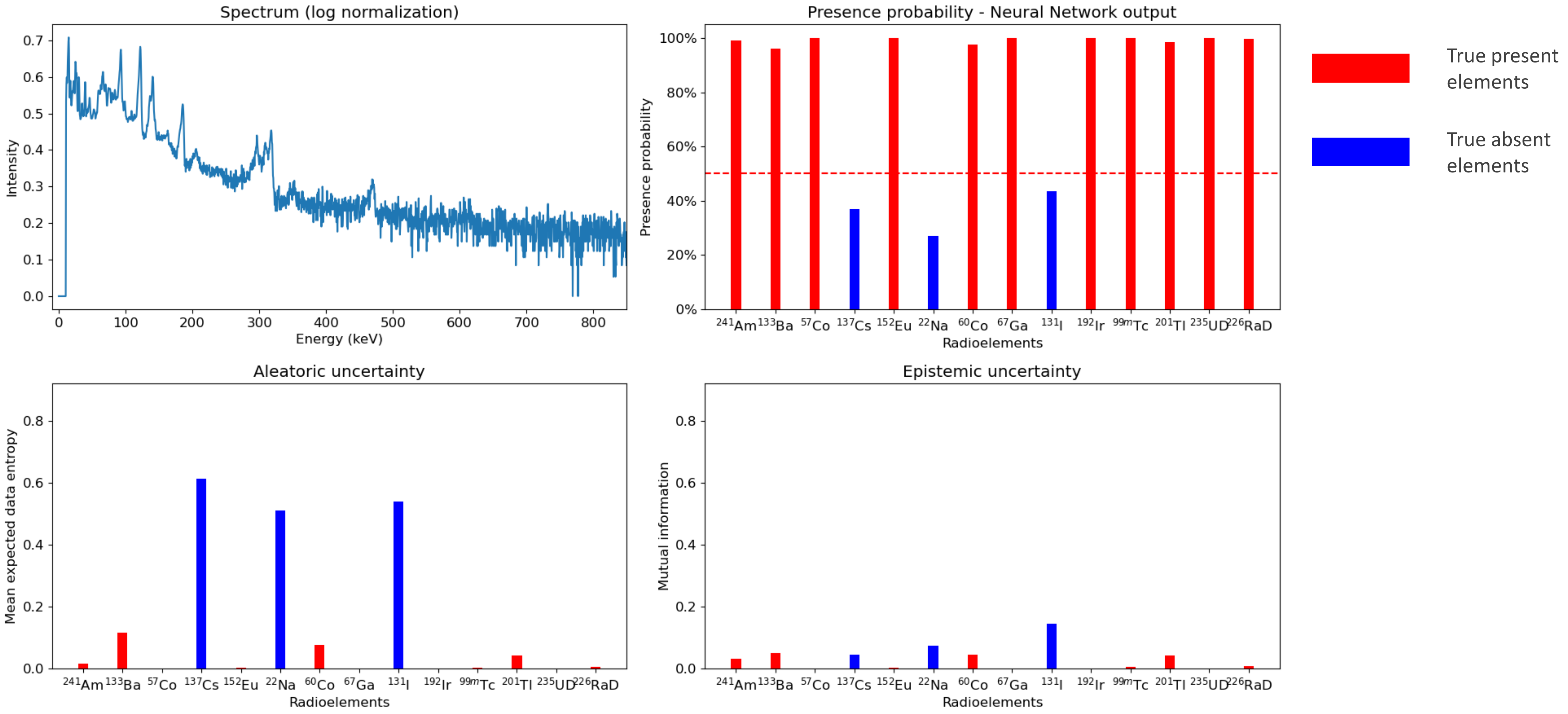


(a) Baseline neural network (b) Bernoulli DropConnec (c) Gaussian DropConnect



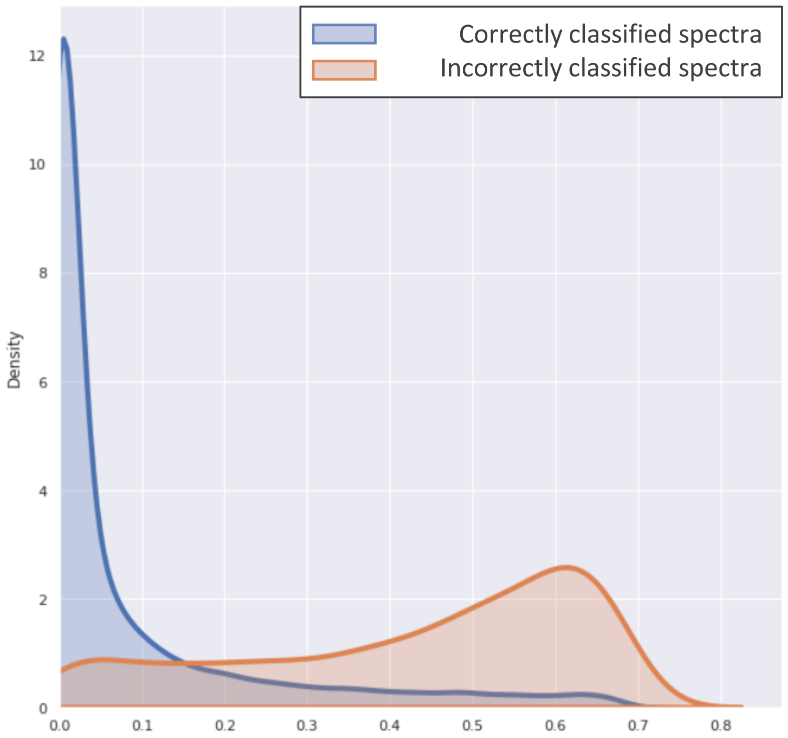
(d) Bernoulli Dropout (e) Gaussian Dropout (f) Spike-and-Slab Dropout

Example: application to gamma spectroscopy

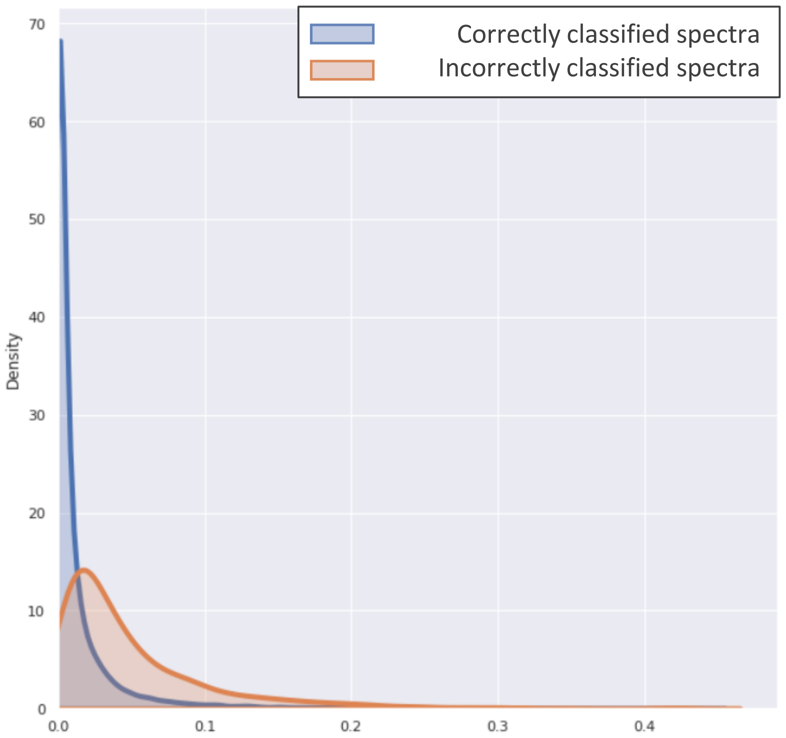


Numerous approaches in Deep Learning

Use uncertainty to « identify » wrong predictions?



Mean entropy
= Aleatoric uncertainty

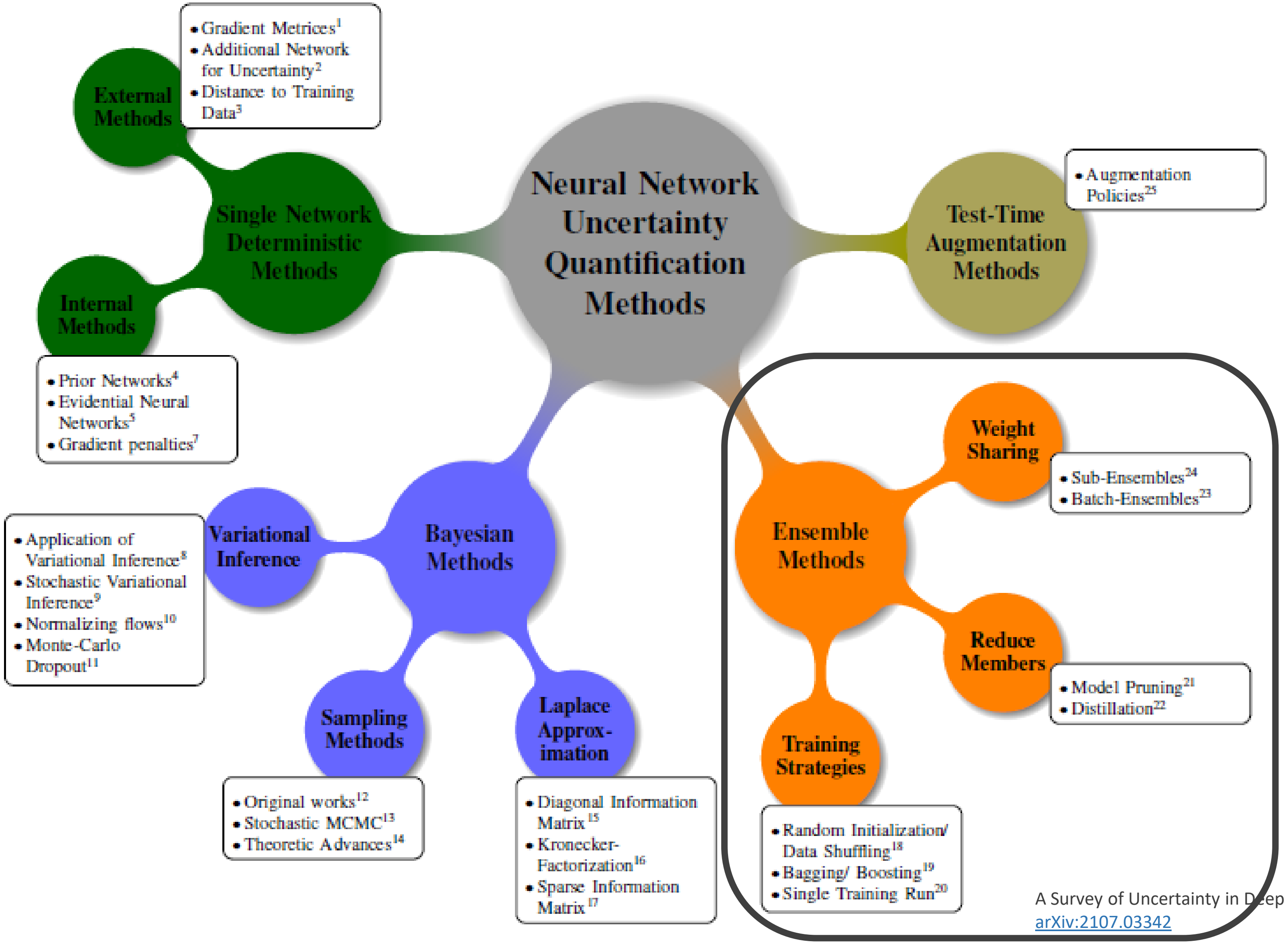


Mutual information
= Epistemic uncertainty

Wrong predictions: high aleatoric uncertainty
Correct predictions: low aleatoric uncertainty

Wrong predictions: slightly higher uncertainty than correct predictions

Numerous approaches in Deep Learning



- Gradient Metrices[1]
- Additional Network for Uncertainty[2]
- Distance to Training Data[3]

- Prior Networks[4]
- Evidential Neural Networks[5]
- Gradient penalties[7]

- Application of Variational Inference[8]
- Stochastic Variational Inference[9]
- Normalizing flows[10]
- Monte-Carlo Dropout[11]

- Original works[12]
- Stochastic MCMC[13]
- Theoretic Advances[14]

- Diagonal Information Matrix[15]
- Kronecker-Factorization[16]
- Sparse Information Matrix[17]

- Sub-Ensembles[24]
- Batch-Ensembles[23]

- Model Pruning[21]
- Distillation[22]

- Random Initialization/ Data Shuffling[18]
- Bagging/ Boosting[19]
- Single Training Run[20]

- Augmentation Policies[25]

A Survey of Uncertainty in Deep Neural Networks, J. Gawlikowski et al,, arXiv:2107.03342

Numerous approaches in Deep Learning

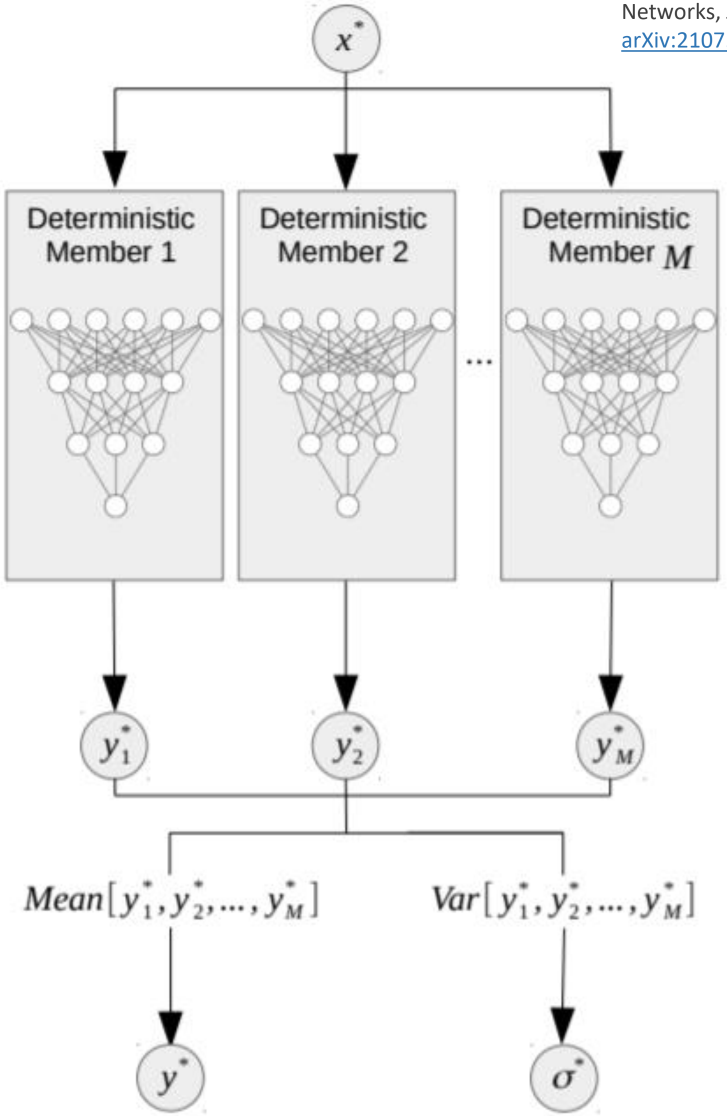
Ensemble methods :

Main idea: train several neural networks in order to provide a variability on the output

Several approaches:
- Random initialization of the network parameters
- Database shuffling
- Random noise in the gradient descent

Limitations:
- Requires the learning of several models
- Storing several models

A Survey of Uncertainty in Deep Neural Networks, J. Gawlikowski et al,, arXiv:2107.03342



Numerous approaches in Deep Learning

Ensemble methods : some illustrations for a regression

Function to fit:

$$f: \mathbb{R}^2 \to \mathbb{R}$$
$$(x_1, x_2) \to \cos(x_1) + \sin(x_2)$$

Training points only on a manifold (circle):

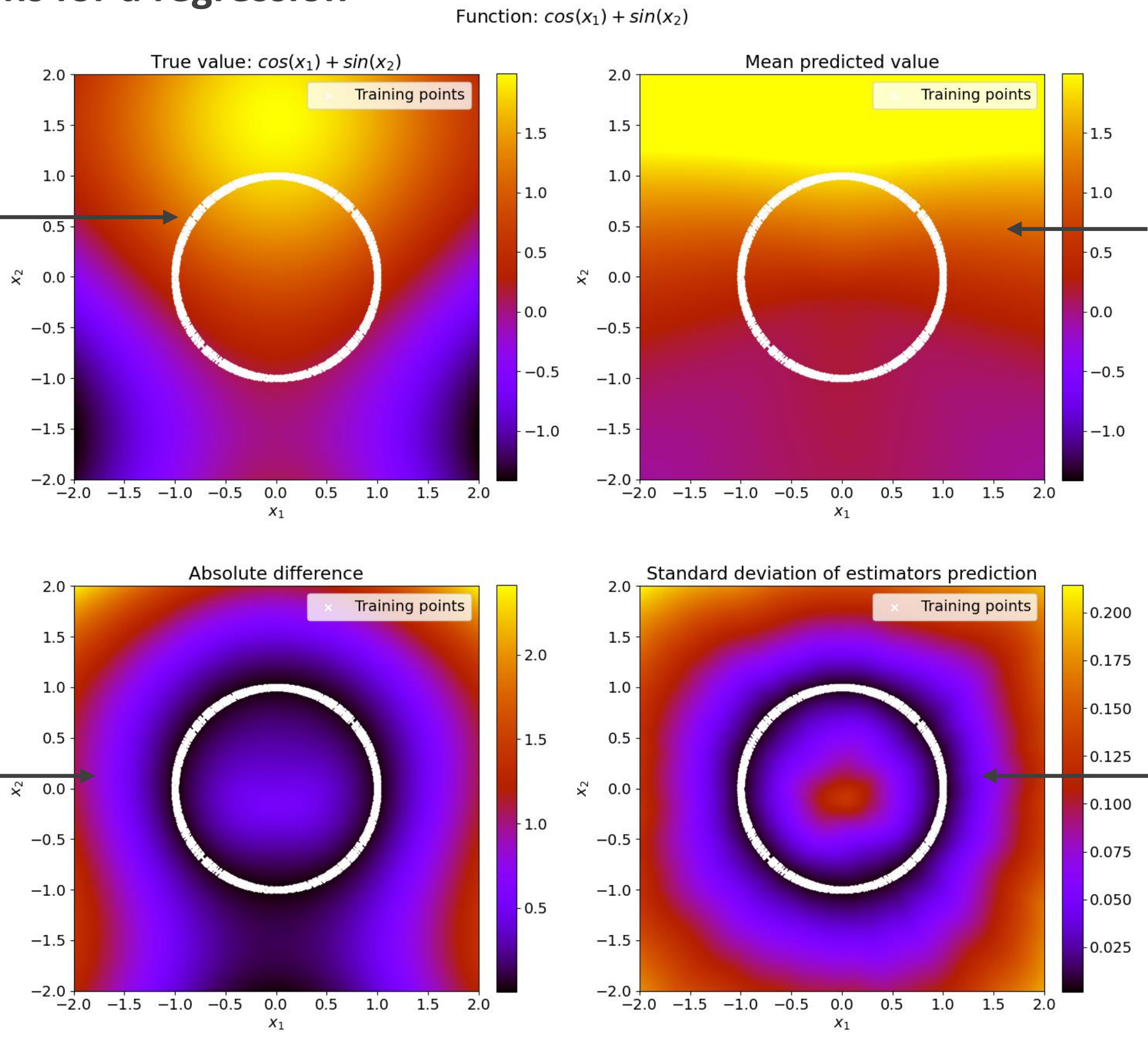
$$x_1^2 + x_2^2 = 1$$

Ensemble with $N = 50$ estimators $f^{(i)}$

Absolute difference:

$$|\hat{y} - f(x_1, x_2)|$$

Correct predictions on the training points
Bad predictions far from the training points



Mean predicted value:

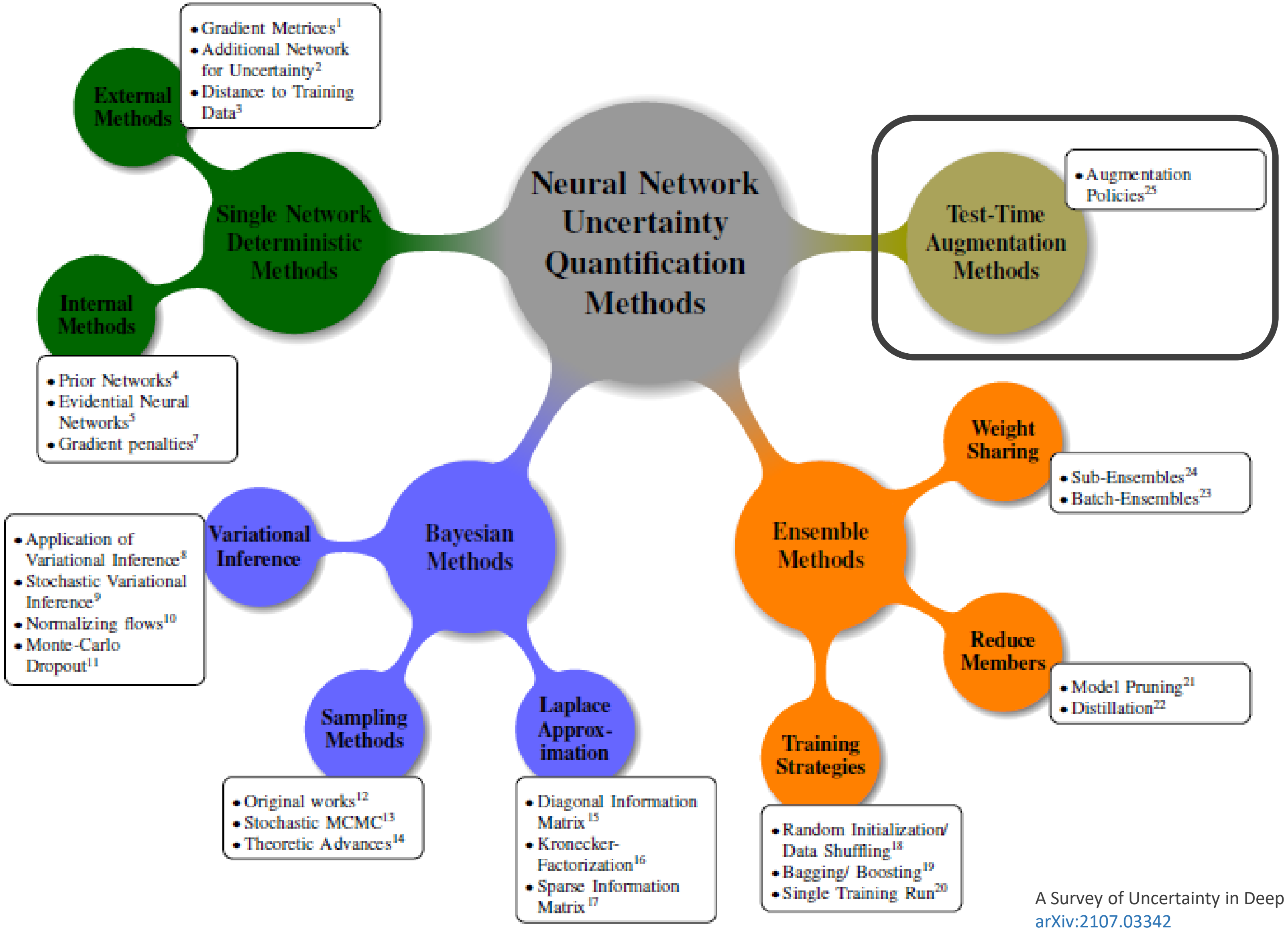
$$\hat{y} = \frac{1}{N}\sum_i f^{(i)}(x_1, x_2)$$

Standard deviation:

$$\sqrt{\frac{1}{N}\sum_i \left(f^{(i)}(x_1, x_2) - \hat{y}\right)^2}$$

Estimators have the same behavior on the manifold (training points)
They « disagree » far from the training set

Numerous approaches in Deep Learning



Neural Network Uncertainty Quantification Methods

Single Network Deterministic Methods

External Methods
- Gradient Metrices[1]
- Additional Network for Uncertainty[2]
- Distance to Training Data[3]

Internal Methods
- Prior Networks[4]
- Evidential Neural Networks[5]
- Gradient penalties[7]

Test-Time Augmentation Methods
- Augmentation Policies[25]

Bayesian Methods

Variational Inference
- Application of Variational Inference[8]
- Stochastic Variational Inference[9]
- Normalizing flows[10]
- Monte-Carlo Dropout[11]

Sampling Methods
- Original works[12]
- Stochastic MCMC[13]
- Theoretic Advances[14]

Laplace Approx-imation
- Diagonal Information Matrix[15]
- Kronecker-Factorization[16]
- Sparse Information Matrix[17]

Ensemble Methods

Weight Sharing
- Sub-Ensembles[24]
- Batch-Ensembles[23]

Reduce Members
- Model Pruning[21]
- Distillation[22]

Training Strategies
- Random Initialization/ Data Shuffling[18]
- Bagging/ Boosting[19]
- Single Training Run[20]

A Survey of Uncertainty in Deep Neural Networks, J. Gawlikowski et al,, arXiv:2107.03342

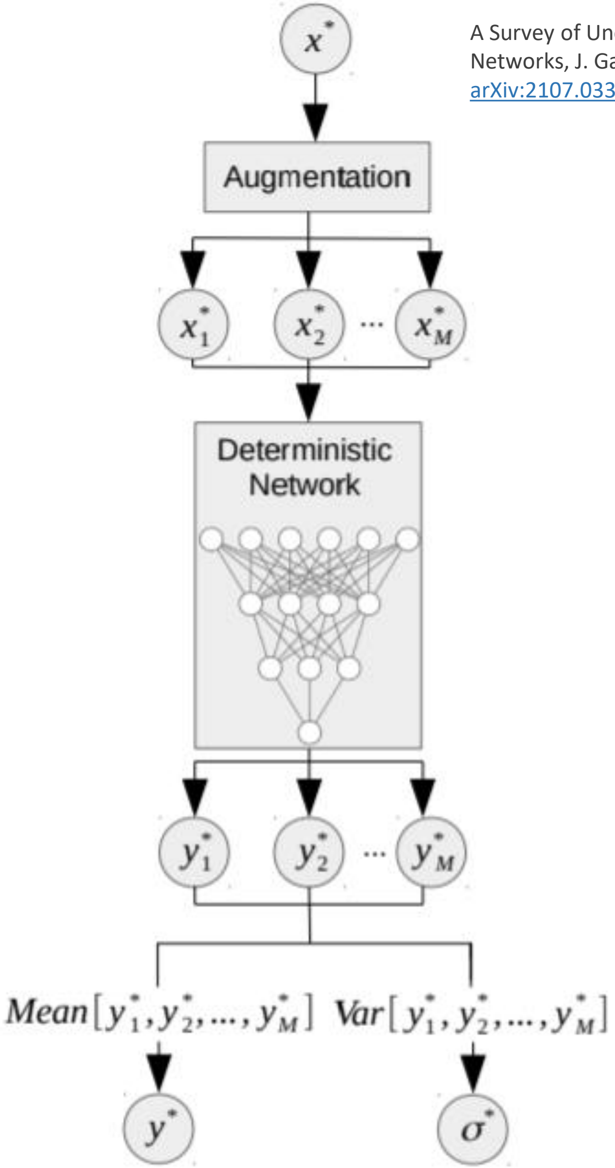
Numerous approaches in Deep Learning

Test-time augmentation methods:

Main idea: Introducing noise or distortion in the input in order to obtain a variability in the prediction

Several distortions:
- Gaussian noise or other noises
- For images: zoom, rotations, deactivate pixels in the image…

Limitations:
- Selection of the noise level, the type of distortions
- Real representation of uncertainties?



A Survey of Uncertainty in Deep Neural Networks, J. Gawlikowski et al,, arXiv:2107.03342

Plan

Dealing with uncertainties in Machine Learning

1. Different levels of uncertainties
2. Uncertainties estimation in Deep Learning
3. Validation of uncertainties

Works in collaboration with Jean-Marc MARTINEZ (DES/ISAS/DM2S/STMF/LGLS)
And three interns:

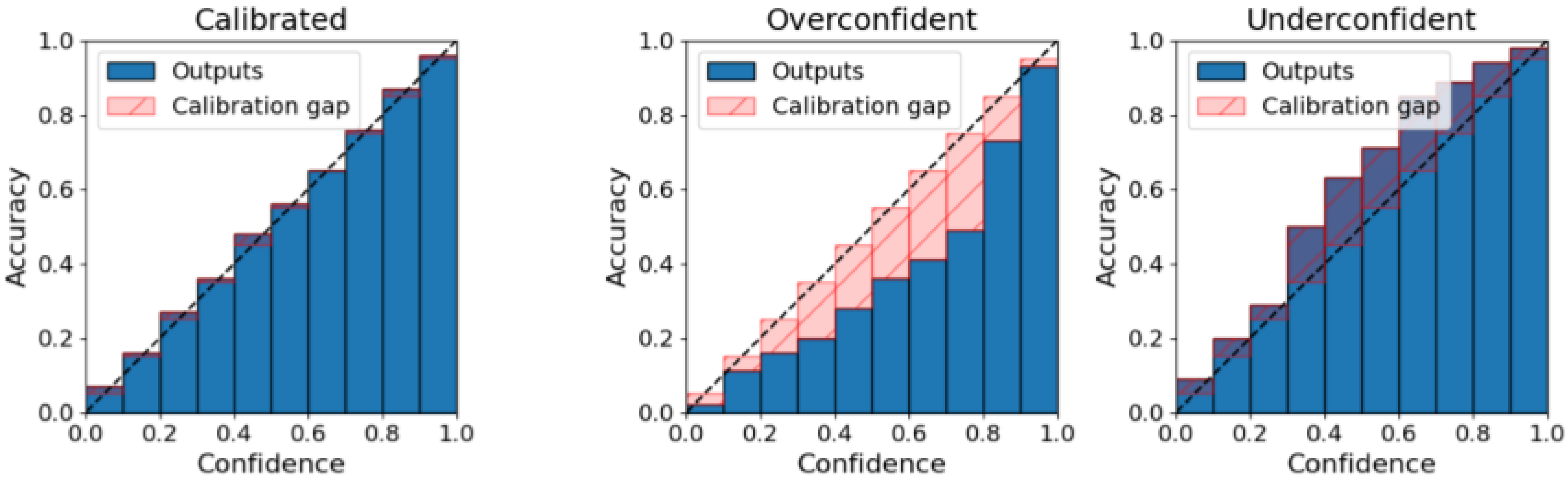
- Mohamed Bahi YAHIAOUI (Mines St Etienne)
- Clément RIBES (IMT Atlantique)
- Olivier LAURENT (ENSTA)

Reliability Plots

Main idea: probabilistic interpretation of the neural network output:

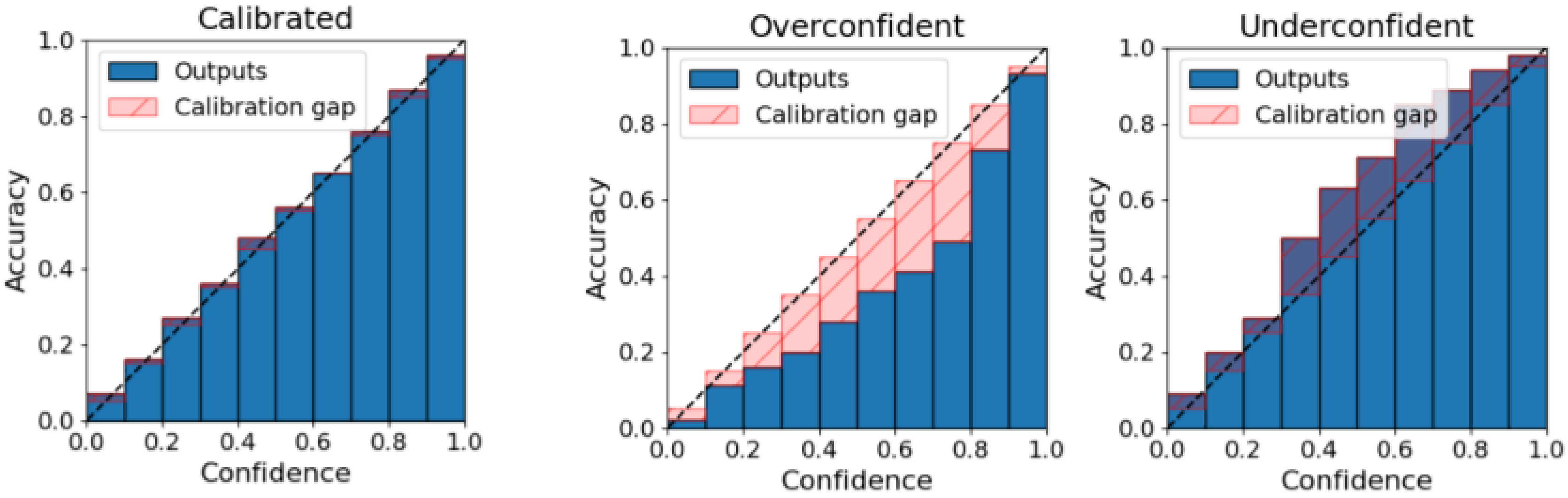
$$f_\omega^{(i)} = p\left(y^{(i)} \middle| x\right)$$

Comparison in the test set between the frequency of correct answers and the associated probabilities
Example: the frequency of correct classifications with a predicted probability 40% should be 40%
Expected Calibration Error (ECE) : difference between Accuracy and Confidence



A Survey of Uncertainty in Deep Neural Networks, J. Gawlikowski et al,,
arXiv:2107.03342

Reliability Plots



A Survey of Uncertainty in Deep Neural Networks, J. Gawlikowski et al,, arXiv:2107.03342

Recalibration technics:

- Regularization methods at training time
- Post-processing with a temperature factor (temperature scaling)

$$f_{\omega}^{(i)}(x) = p\left(y^{(i)}\middle|x\right) = \sigma\left(\frac{z^{(i)}}{T}\right)$$

- Calibration of hyperparameters: Dropout rate for MC-Dropout, for instance
- Conformal prediction

Calibration or coverage curves in regression

Prediction interval $I_\alpha(x)$ for a data point x with a probability α :

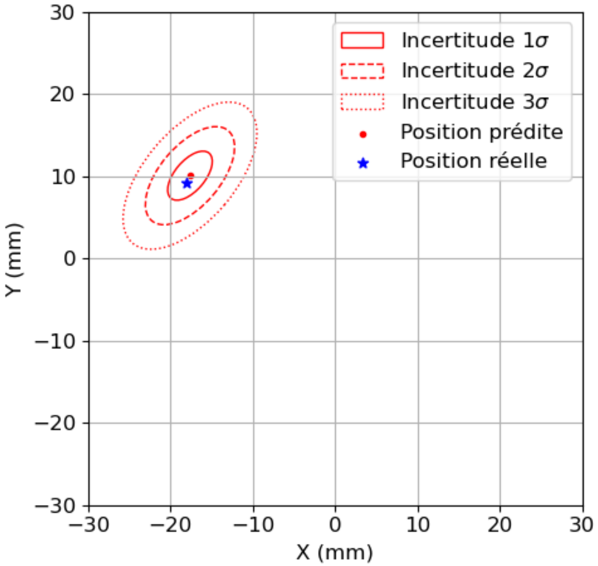
$P\left(I_\alpha(x) \ni y(x)\right) = \alpha$, where $y(x)$ is the expected output for x

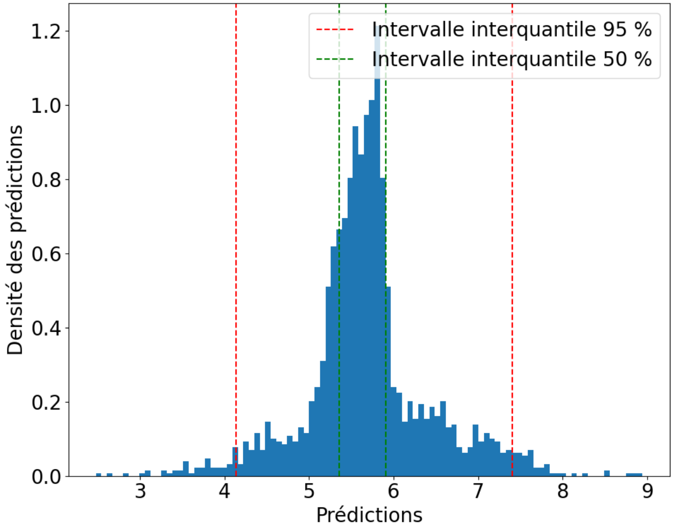
By using methods for un certainty quantification, we can estimate interquantile intervals $\hat{I}_\alpha$:

- Analytically in the case of Density Neural Networks:

  Direct estimation of the probability density $p(y|\theta)$ where $\theta = f_\omega(x)$

- Direct computation by using Monte-Carlo sampling of the output





Calibration or coverage curves in regression

Does estimated interquantile intervals $\hat{I}_\alpha$ with probability α correspond to prediction intervals I_α with probability α ?

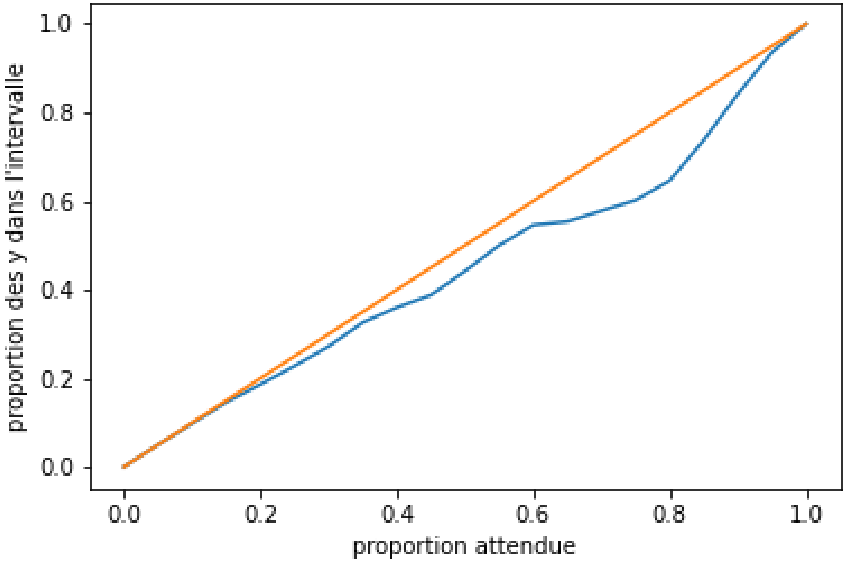
Verification procedure: mean evaluation on the test set

$$\mathbb{E}_x\left[P\left(\hat{I}_\alpha(x) \ni y(x)\right)\right] = \int_x P\left(\hat{I}_\alpha(x) \ni y(x)\right) p(x)dx \underset{?}{=} \alpha$$

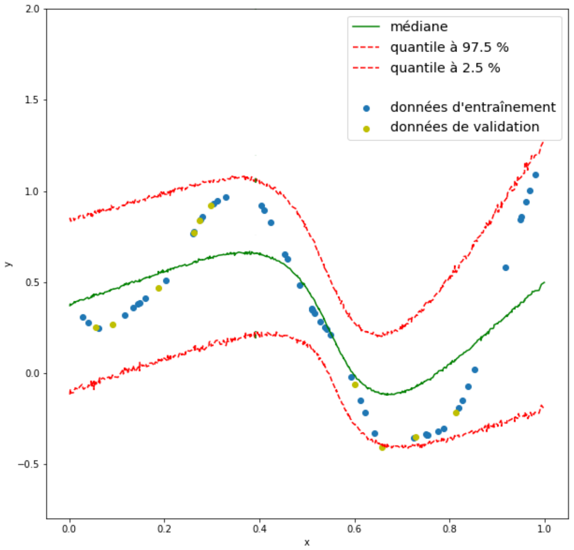
Calibration curve: frequency of exepected output in the predicted interquantile interval $\hat{I}_\alpha$ compared to the probability α

Recalibration :

- Modification of the density model in Density Neural Networks
- Hyperparameters tuning (Dropout rate in MC-Dropout, prior for Bayesian Neural Networks…)
- Conformal prediction (MAPIE)



If the uncertainties are well calibrated, it does not imply that the model has good predictive performance



Calibration or coverage curves in regression

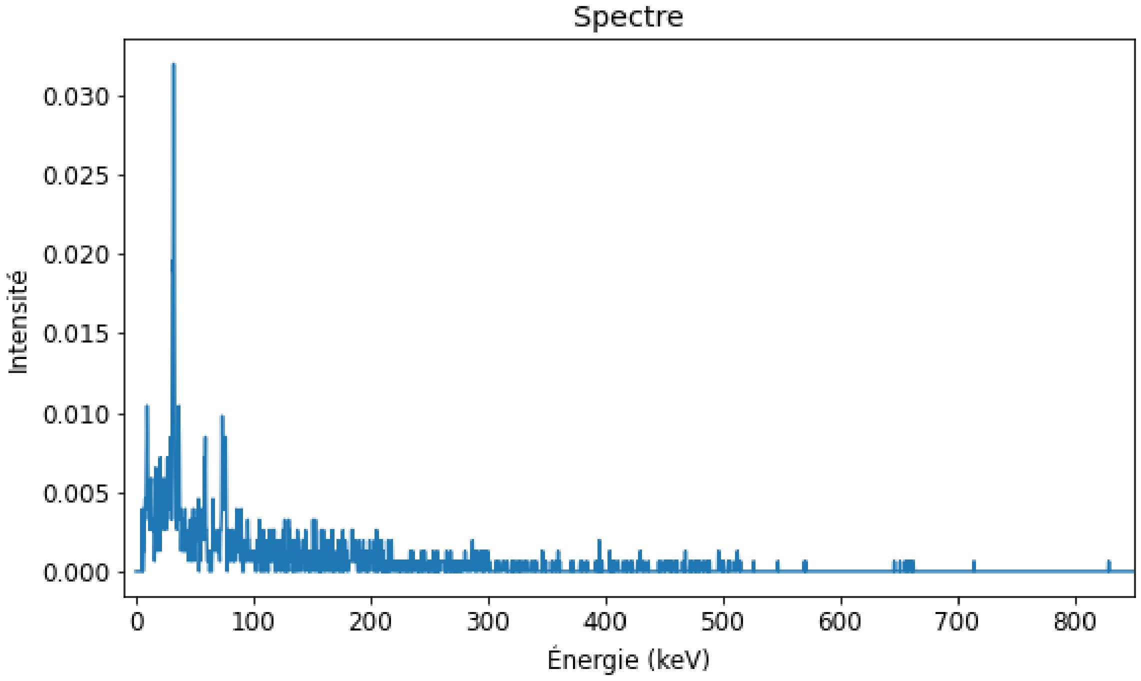
Application: examples for gamma spectroscopy

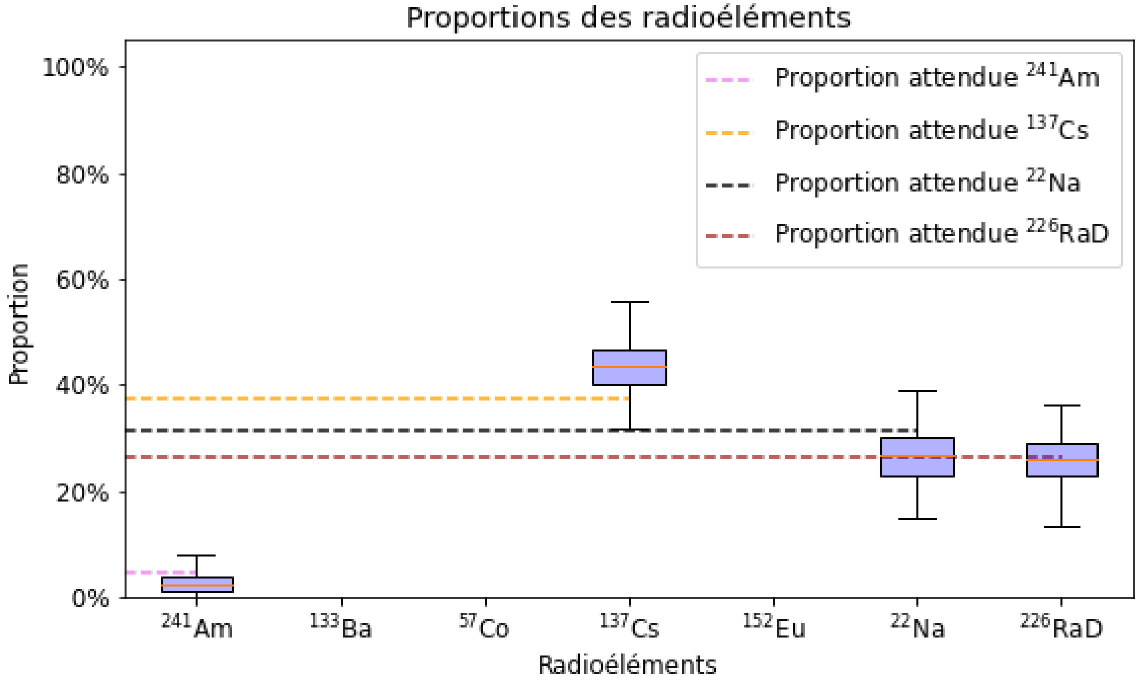
Pay attention: prediction of the proportions for each radioelement (not their presence)

→ Regression problem

→ Uncertainty prediction by MC-Dropout

Example:

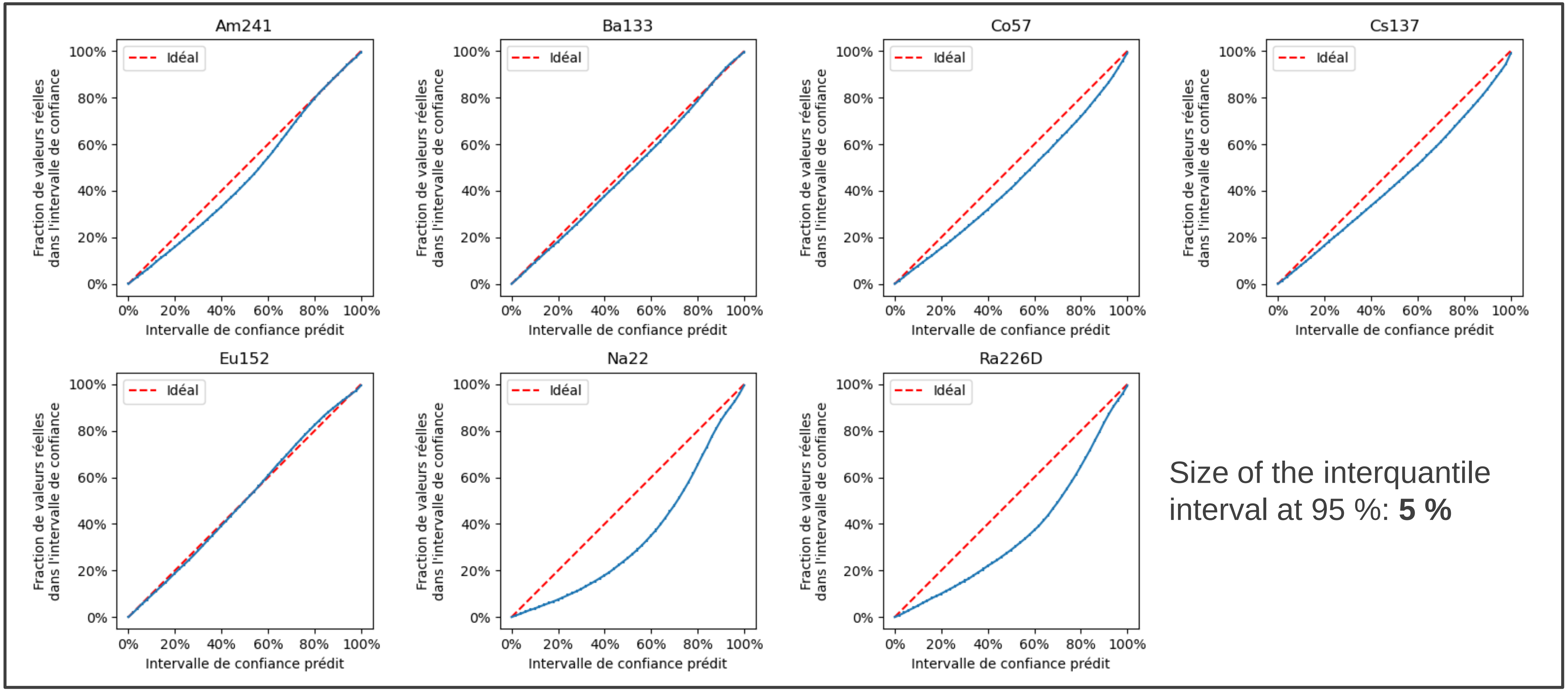

Spectre


Proportions des radioéléments

Calibration or coverage curves in regression

Application: examples for gamma spectroscopy

Calibration curves for a Dropout rate of 0,25



Size of the interquantile interval at 95 %: **5 %**

Q-Q plots for Density Neural Networks

Other method to validate uncertainties in the case of Density Neural Networks

Example: we consider a model that estimates, for each input x, the mean $\hat{\mu}(x)$ and the standard deviation $\hat{\sigma}(x)$

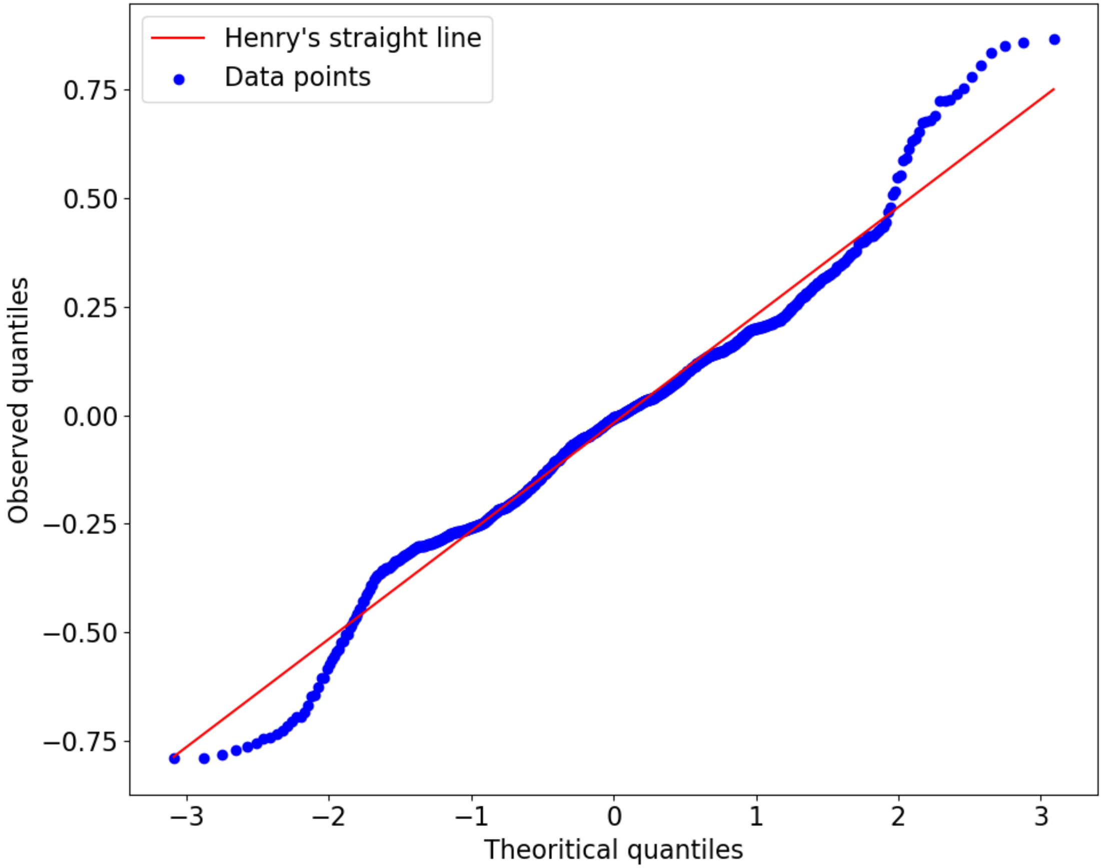
We compute the normalized error for each data point (x_i, y_i):

$$t_i = \frac{(y_i - \hat{\mu}(x_i))}{\hat{\sigma}(x_i)}$$

If y_i follows a normal distribution $\mathcal{N}(\hat{\mu}(x_i), \hat{\sigma}(x_i))$, then t_i follows a normal distribution $\mathcal{N}(0,1)$.

We can compute the quantile-quantile plot of $(t_i)_i$ on the dataset.

Example of Q-Q plot



Observed quantiles: sorted values of t_i

Theoritical quantiles: $\Phi^{-1}\left(\frac{j}{N}\right)$ where j is the j-th sorted index and N is the size of the dataset

Conclusion

Methods for uncertainties quantification in Deep Learning:
- New state-of-the-art
- Rapid evolution in the literature

Be cautious with the interpretation → Estimated uncertainties must be validated

Simplified implementation thanks to dedicated libraries and frameworks : Tensorflow Probabilities, MAPIE…

→ Importance to consider uncertainties notions in scientific applications of Deep (Machine) Learning algorithms
→ Connections to notions of robustness (sensibility of neural networks to small perturbations), out-of-distribution detection…

Thanks for your attention!
Questions?

References

Reviews on uncertainties in Deep Learning

- *A Survey of Uncertainty in Deep Neural Networks*, J. Gawlikowski et al. (2021)
  arXiv:2107.03342

- *A review of uncertainty quantification in deep learning: Techniques, applications and challenges*, M. Abdar et al. (2020), Information Fusion
  https://doi.org/10.1016/j.inffus.2021.05.008

Mixture Density Network (extension of Density Neural Networks)

- *Mixture Density Networks*, Christopher M. Bishop (1994), NCRG/94/004
  https://publications.aston.ac.uk/id/eprint/373/1/NCRG_94_004.pdf

Conformalized quantile regression

- *Conformalized Quantile Regression,* Y. Romano et al. (2019)
  https://arxiv.org/pdf/1905.03222.pdf

Bayes by backprop

- *Weight Uncertainty in Neural Networks*, Blundell et al. (2015), Proceedings of the 32 nd International Conference on Machine Learning
  arXiv:1505.05424v2

Monte-Carlo Dropout

- *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning,* Gal & Ghahramani (2016), Proceedings of the 33 rd International Conference on Machine Learning
  arXiv:1506.02142v6