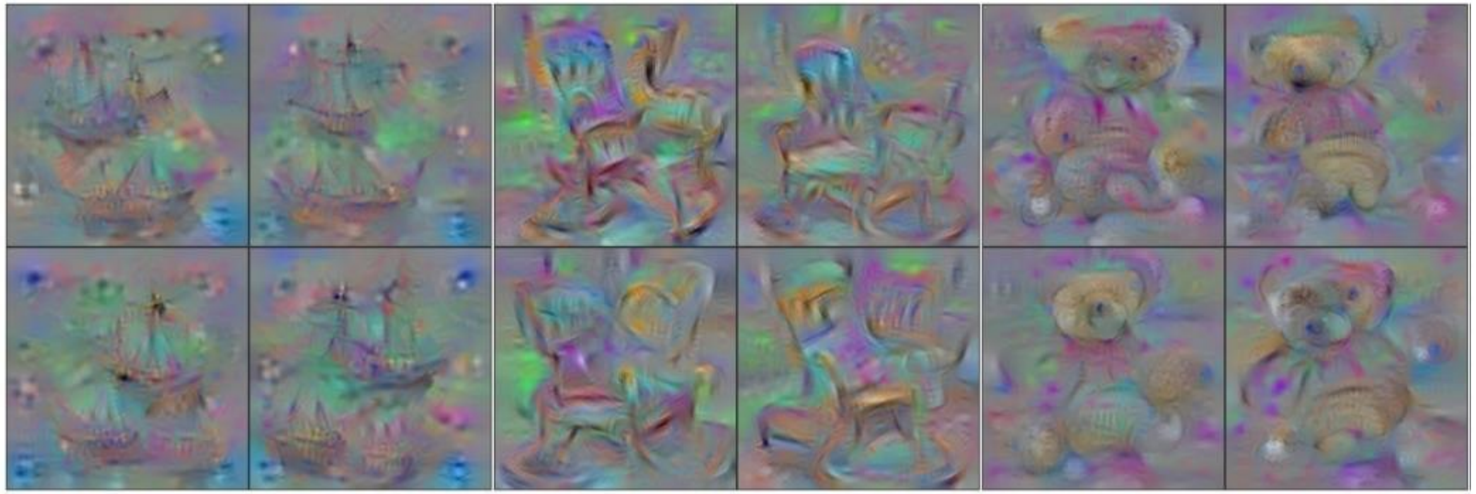
ECE 4252/8803: Fundamentals of Machine Learning (FunML)
Fall 2024

Lecture 27: Uncertainty Quantification in Neural Networks



Logistics

Finals

- Friday Dec 6, 90 mins
- Timeslot that the exam will be available on Canvas: 8 – 10 AM
  - **You can start at any time before 8.30 AM to have the full 90 mins window**
- Lectures 1 – 27 will be covered in the finals
  - Additional emphasis on Lecture 13 onwards
- All other logistics are the same as Midterms

Overview

In this Lecture..

Introduction and Motivation

Two Main Types of Uncertainty

- Aleatoric Uncertainty
- Epistemic Uncertainty

Iterative Uncertainty Estimation

Single Pass Uncertainty Estimation

Performance Metrics

Examples of Human Uncertainty

Blue or Gold?

Introduction
Uncertainty Quantification

Examples of Human Uncertainty

Pneumonia or Healthy?

For humans, uncertainty is frequently associated with imperfect decisions.

Examples of Human Uncertainty

Pneumonia or Healthy?

Your ideal response: let's get a doctor!

Introduction

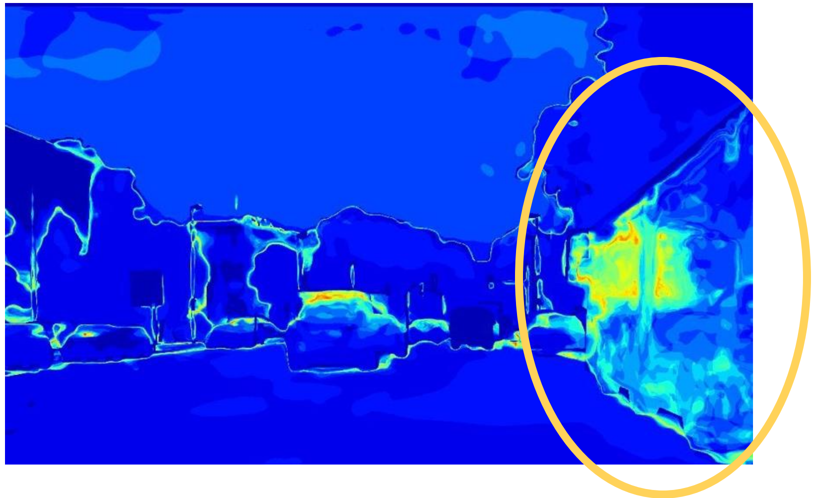
Uncertainty Quantification in Neural Networks

For machine learning models, uncertainty quantification is a function of their predictions

Input Image

Neural Network Output

Uncertainty Heatmap



Uncertainty Quantification is Crucial in Real-world Deployment

Undesirable Consequences

DOT report on fatal 2016 Tesla crash with tractor-trailer blames limitations of Autopilot mode

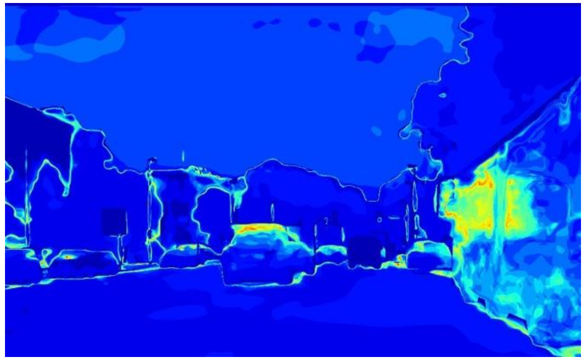
James Jaillet
Feb 2, 2017 | Updated Feb 21, 2017

An NTSB photo of the Freightliner Cascadia involved in the May 7 crash.

Ideal Expectations

Input Image

Uncertainty Heatmap



Knowing what a model does not know is essential for establishing reliability

NHTSA. PE 16-007. Technical report, U.S. Department of Transportation, National Highway Traffic Safety Administration, Jan 2017. Tesla Crash Preliminary Evaluation Report.

Overview

In this Lecture..

Introduction and Motivation

Two Main Types of Uncertainty

- Aleatoric Uncertainty
- Epistemic Uncertainty

Iterative Uncertainty Estimation

Single Pass Uncertainty Estimation

Performance Metrics

Uncertainty Quantification

Factors that Cause Uncertainty

- Noise during Data Acquisition and Measurement
- Variations among Model Configurations
- Unknown Data

Uncertainty Quantification

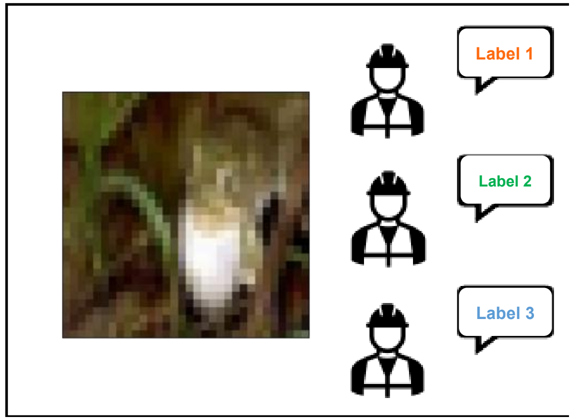
Factors that Cause Uncertainty

- **Noise during Data Acquisition and Measurement**
  - Sample and/or label noise[1,2]
- Variations among Model Configurations
- Unknown Data

Data distortion

Gaussian Blur | Noise | Rain | Shadow | Snow

Variations among labels

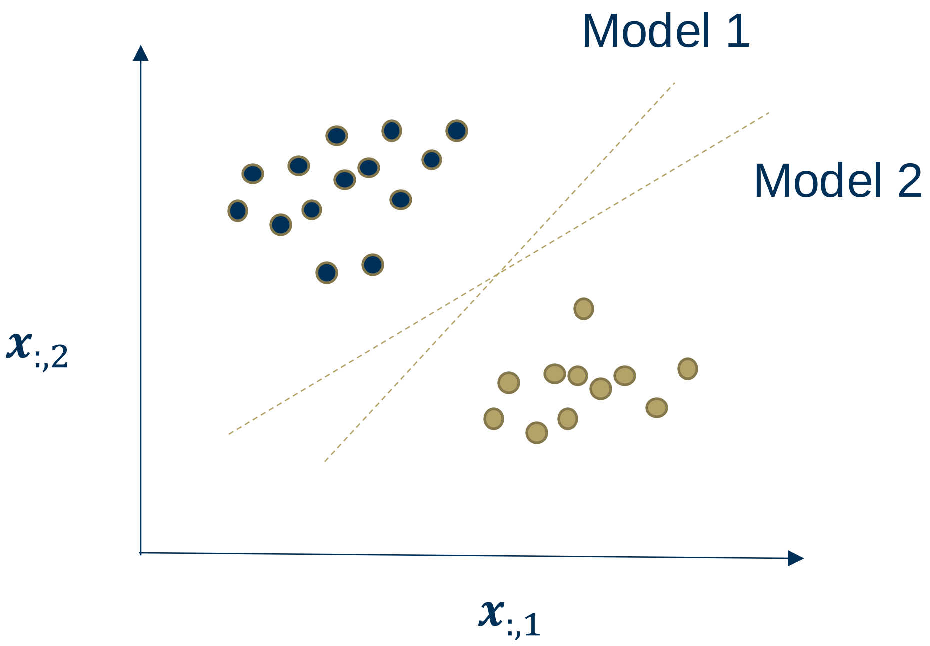


[1] Temel, Dogancan, et al. "CURE-TSR: Challenging unreal and real environments for traffic sign recognition." in NeurIPS Workshop on Machine Learning for Intelligent Transportation Systems, 2017
[2] C. Zhou, M. Prabhushankar, and G. AlRegib, "On the Ramifications of Human Label Uncertainty," in NeurIPS Workshop on Human in the Loop

Uncertainty Quantification

Factors that Cause Uncertainty

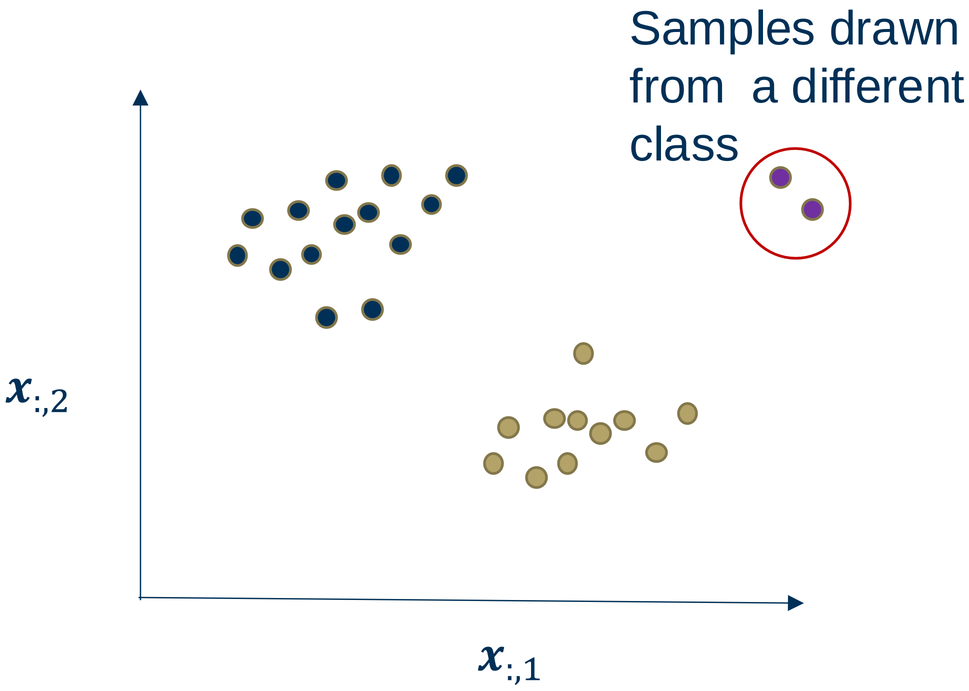
- Noise during Data Acquisition and Measurement
- **Variations among Model Configurations**
  - Different model configurations, parameters, training procedures etc.
- Unknown Data



Uncertainty Quantification

Factors that Cause Uncertainty

- Noise during Data Acquisition and Measurement
- Variations among Model Configurations
- **Unknown Data**
  - Lack of knowledge about the input data, e.g., unknown classes, out-of-distribution samples



OLIVES @GeorgiaTech

Georgia Tech

Types of Uncertainty

Inherent noise in data

- Noise during Data Acquisition and Measurement
  - Sample and/or label noise

Insufficient model knowledge

- Variations among Model Configurations
  - Different model configurations, parameters, training procedures etc.
- Unknown Data
  - Lack of knowledge about the input data, e.g., unknown classes, out-of-distribution samples

The study of Uncertainty is conducted based on the source – data or model uncertainty

Uncertainty in ML

Aleatoric (Data) Uncertainty

Inherent noise in observation data – also called **irreducible uncertainty**.

Epistemic (Model) Uncertainty

Insufficient model knowledge. **Can be reduced** based on good design choices.

Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? Structural Safety, 31 (2):105–112, 2009.

Overview

In this Lecture..

Introduction and Motivation

Two Main Types of Uncertainty

- Aleatoric Uncertainty
- Epistemic Uncertainty

Iterative Uncertainty Estimation

Single Pass Uncertainty Estimation

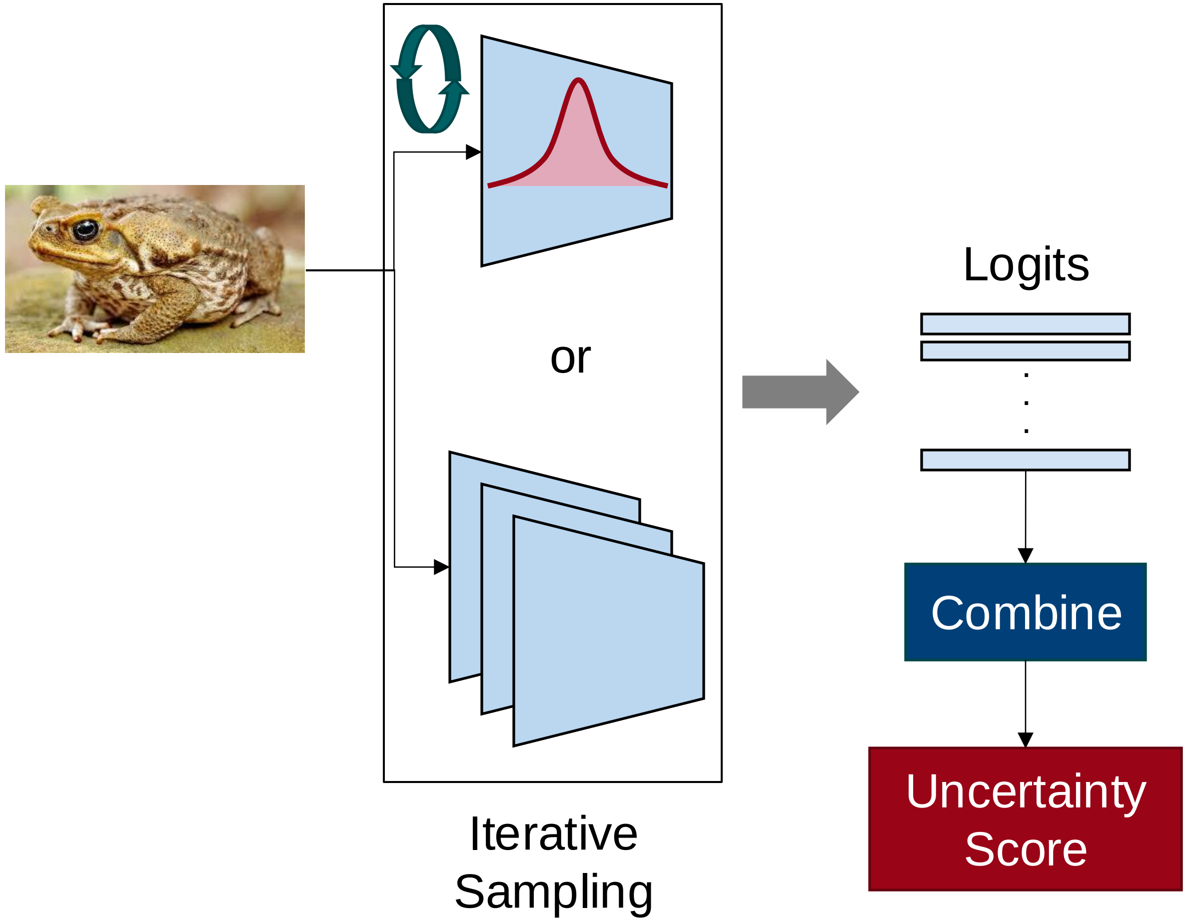
Performance Metrics

Epistemic Uncertainty Quantification

Iterative Uncertainty Methods

- **Definition:**
  1. Iteratively generate multiple model outputs with different parameter constellations
  2. Combine outputs into single uncertainty score
  3. Equivalent to exploring the parameter space

- **Examples:**
  1. Deep Ensembles
  2. Monte Carlo (MC)-Dropout:

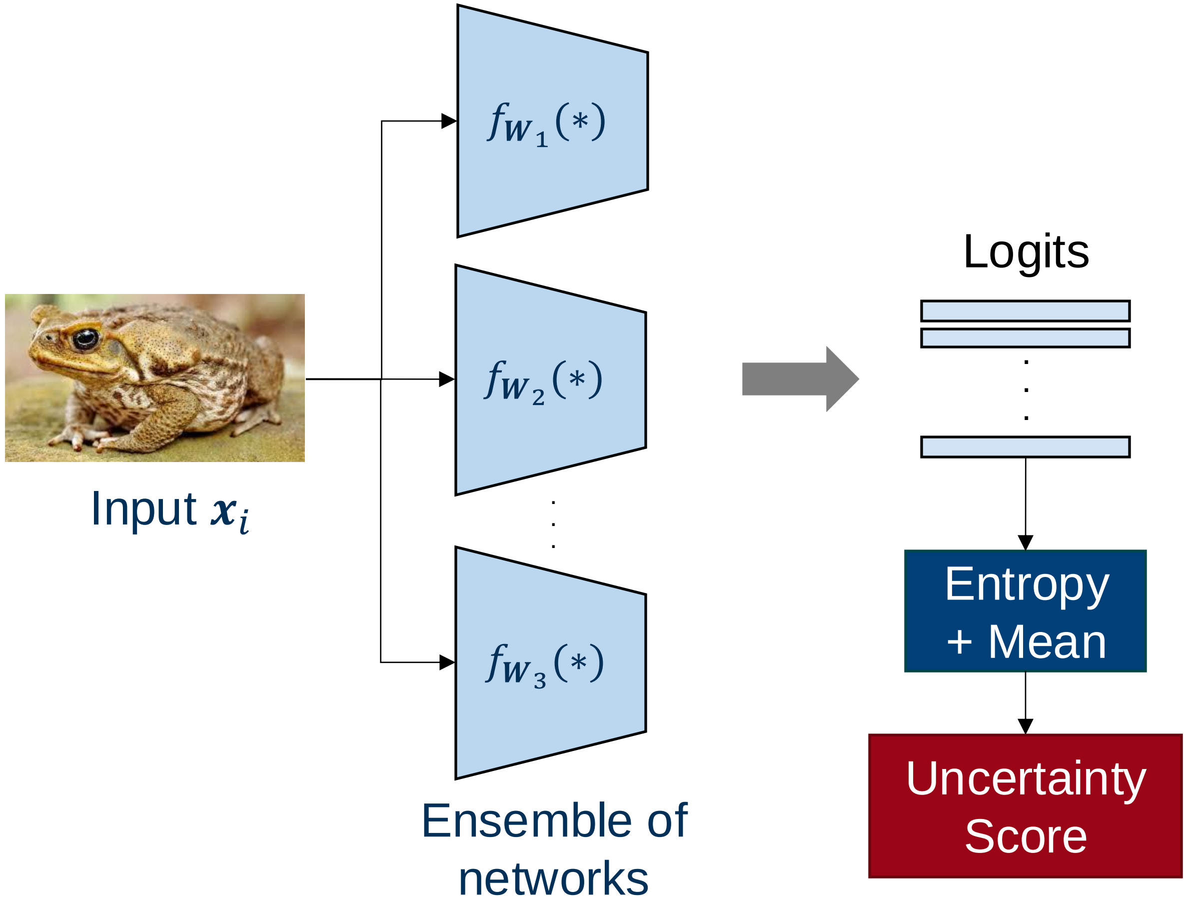


Iterative Uncertainty Methods

Deep Ensembles

- Different initialization parameters provides various network configurations, and different outputs.

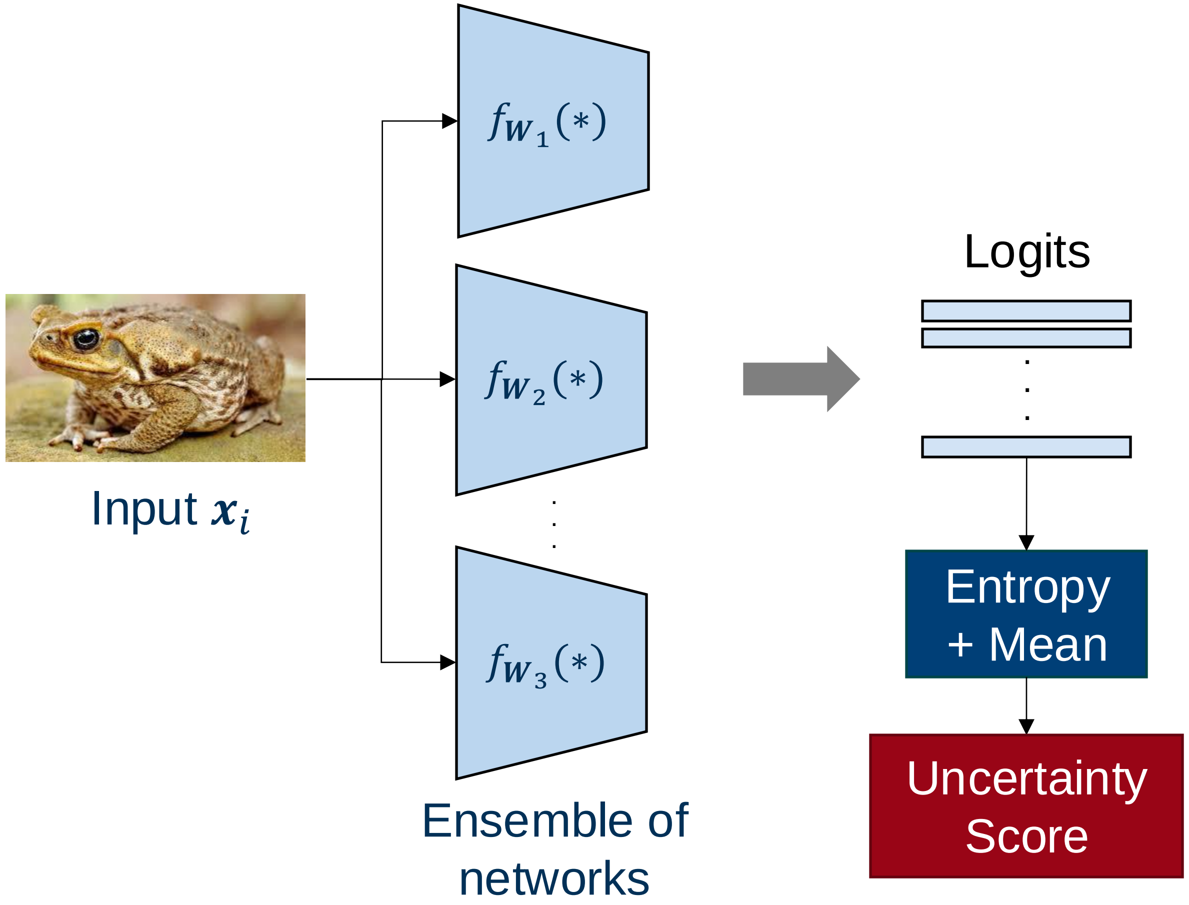
- Final prediction: average
- Uncertainty scores: entropy + mean (Formula in slide 26)



Iterative Uncertainty Methods

Deep Ensembles

- Different initialization parameters provides various network configurations, and different outputs.
- Specifically, use randomness in initialization and/or training data.
    - **Commonly used method: Initialize weights** with a **different random seed.**
    - Less commonly used method: Randomly divide the **data** into **different partitions to** train same network parameters.
- However, **deep ensembles** require training multiple networks and becomes **computationally infeasible**



Iterative Uncertainty Methods

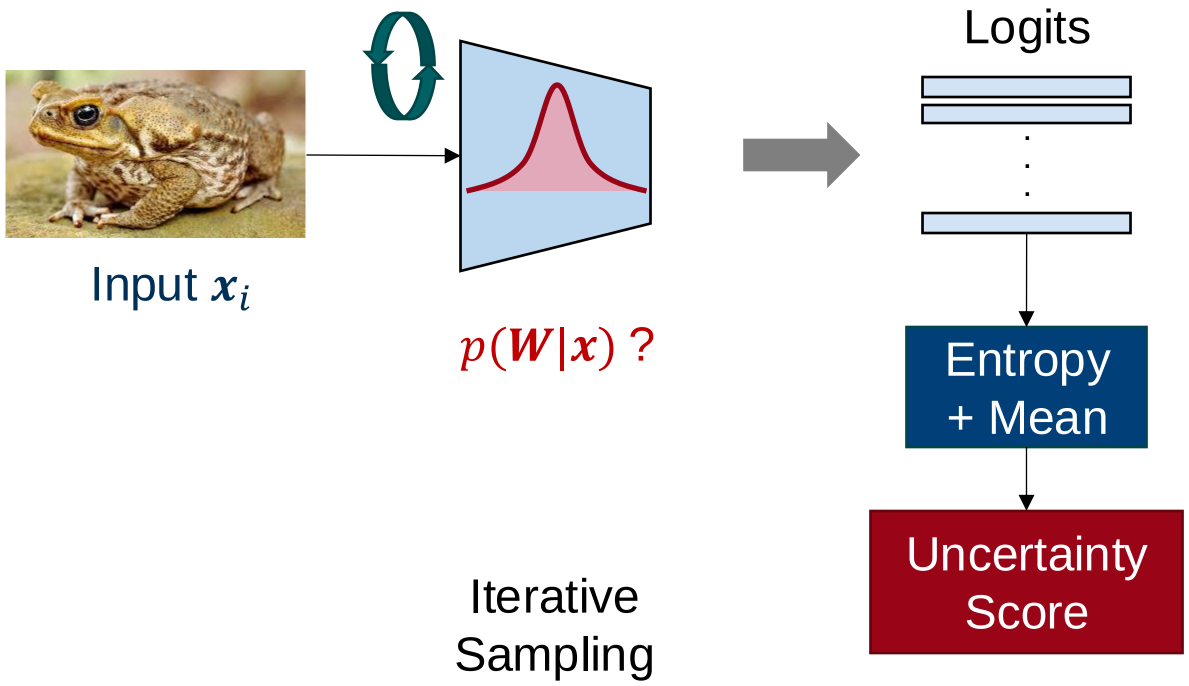
Approximating Deep Ensembles

- Weight Posterior:

$$p(\boldsymbol{W}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{W})p(\boldsymbol{W})}{\int p(\boldsymbol{x}|\boldsymbol{W})p(\boldsymbol{W})d\boldsymbol{W}}$$

  - intractable denominator

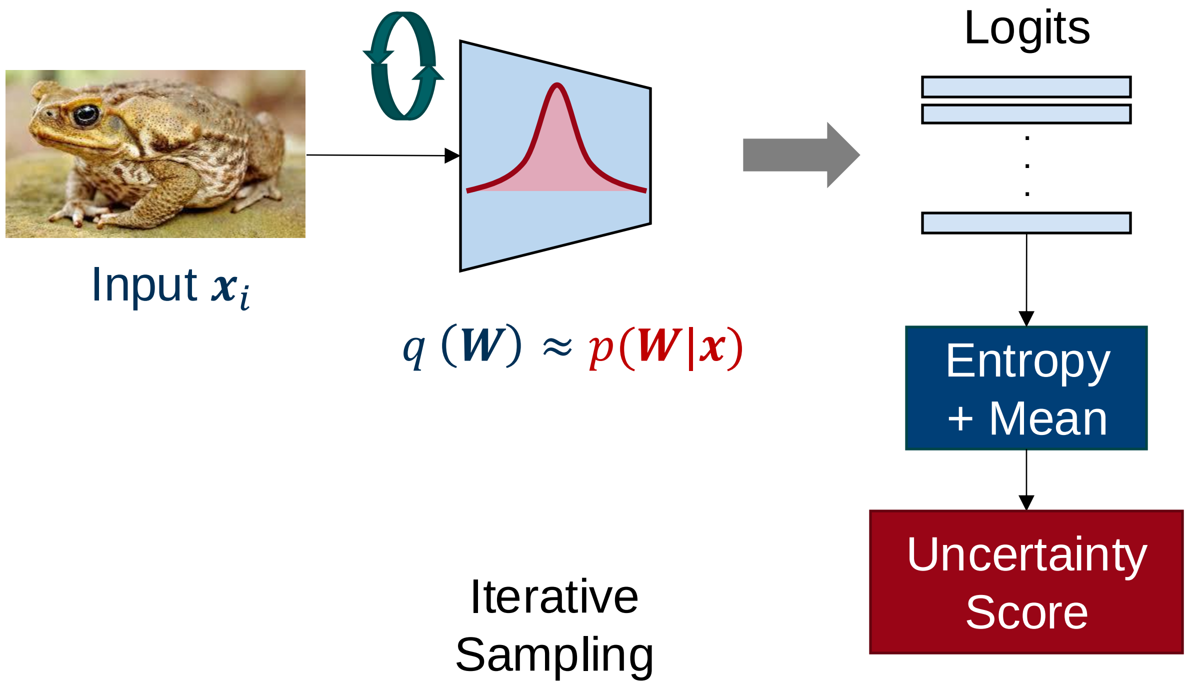
- We approximate the posterior



Iterative Uncertainty Methods

Monte Carlo (MC)-Dropout

- Sample parameter constellations from approximated weight posterior
- Dropout during training
  - from a Bernoulli distribution
- Apply dropout at test time
  - Sampling is done in a Monte Carlo fashion. Hence, it is called Monte Carlo dropout



Y Gal, Z Ghahramani, “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”, ICML 2016

Iterative Uncertainty Methods

Monte Carlo (MC)-Dropout

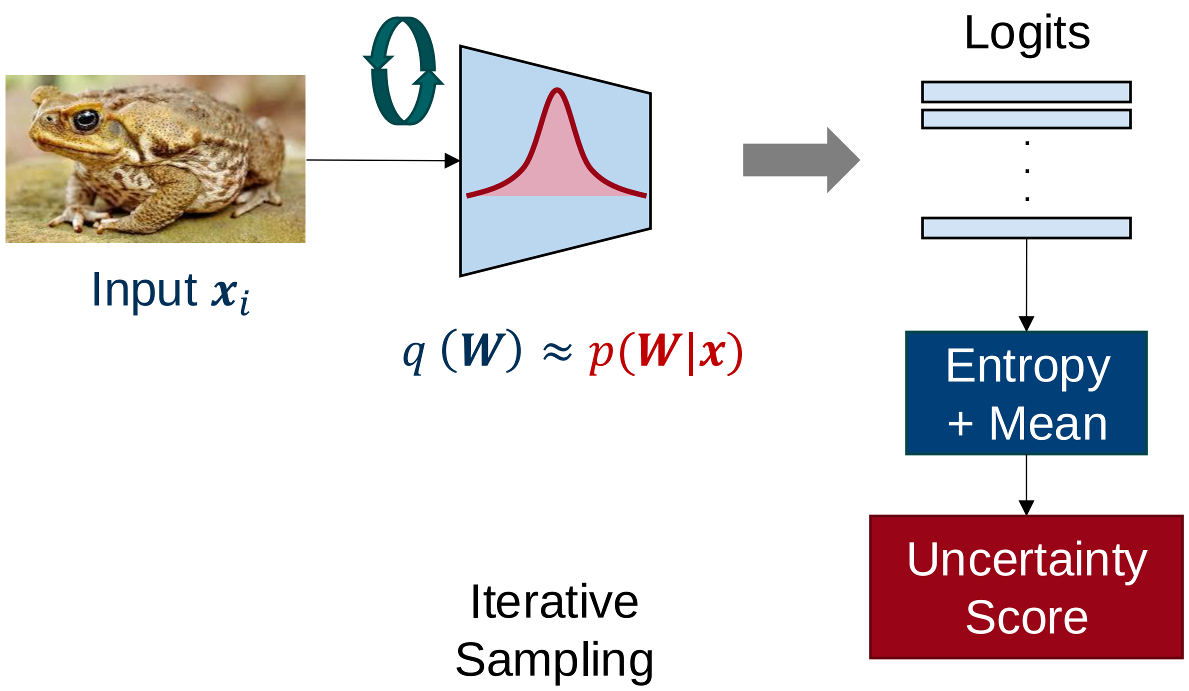
- Sample parameter constellations from approximated weight posterior

- Final prediction: average

$$p(y = c|\boldsymbol{x}) \approx \frac{1}{T}\sum_{t=1}^{T} Softmax(f_{\widehat{\boldsymbol{W}}_t}(\boldsymbol{x}))$$

where $\widehat{\boldsymbol{W}}_t \sim q(\boldsymbol{W})$ (dropout distribution)

- Uncertainty scores: spread, mean



Y Gal, Z Ghahramani, “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”, ICML 2016

Iterative Uncertainty Methods

Monte Carlo (MC)-Dropout

Standard Dropout

- Dropout applied during **training**
- During inference, the activations are multiplied by $(1 - p)$ to represent the “average behavior”

MC Dropout

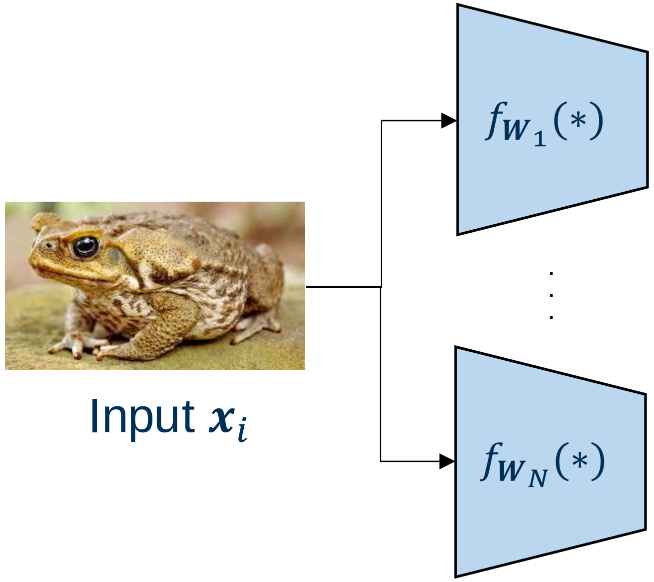
- Dropout is applied both during **training** and **inference**
- Mean as the prediction and entropy + mean as the uncertainty scores

Y Gal, Z Ghahramani, “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”, ICML 2016

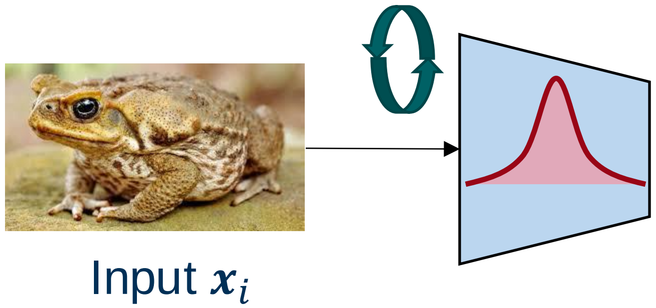
Iterative Uncertainty Methods

Summary

Ensembles



MC-Dropout



$q\,(\boldsymbol{W}) \approx p(\boldsymbol{W}|\boldsymbol{x})$

Multiple Predictions

Mean and Spread as Uncertainty Scores

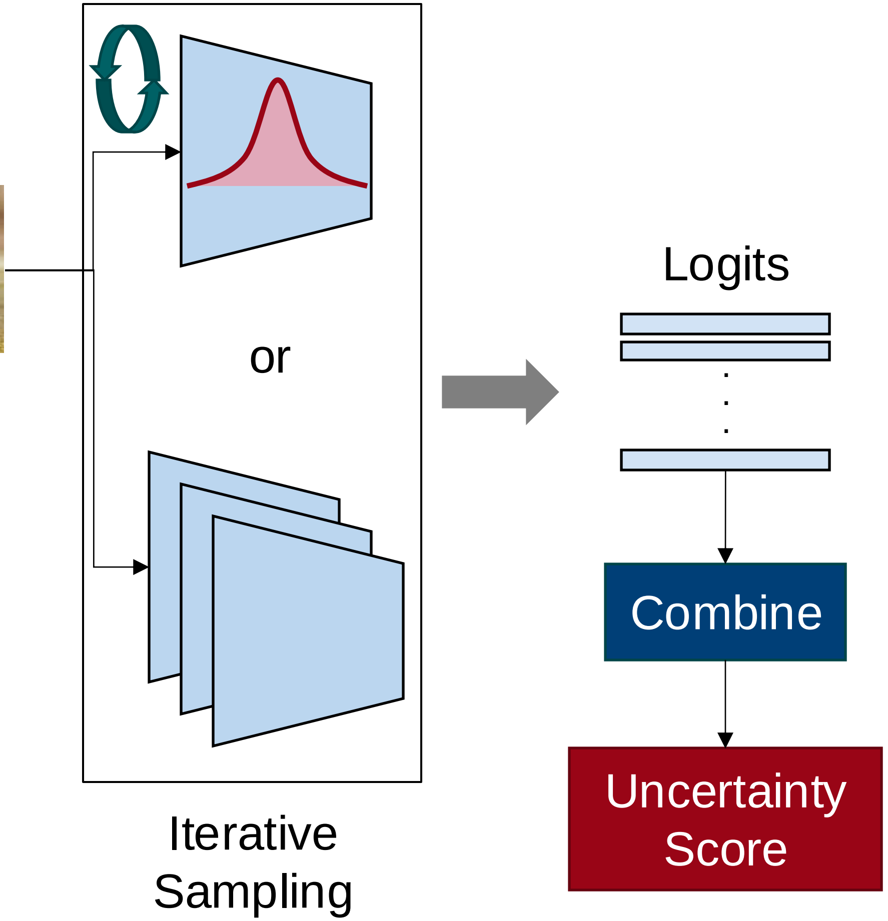
Iterative Uncertainty Estimation

- **Definition:**
  1. Iteratively generate multiple model outputs with different parameter constellations
  2. Combine outputs into single uncertainty score

- **Examples:**
  1. Deep Ensembles
  2. Monte Carlo (MC)-Dropout:

- Pros: Accurate (measured by predictive) uncertainty scores.

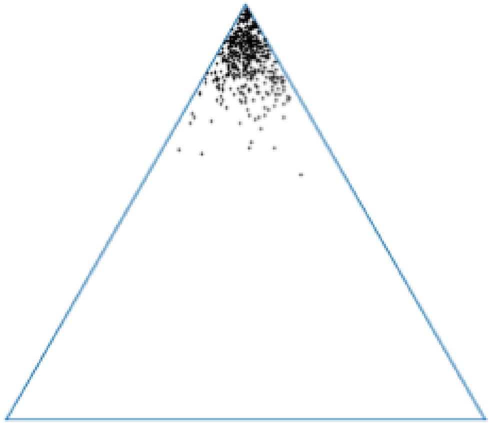
- Cons: High computation/latency cost.



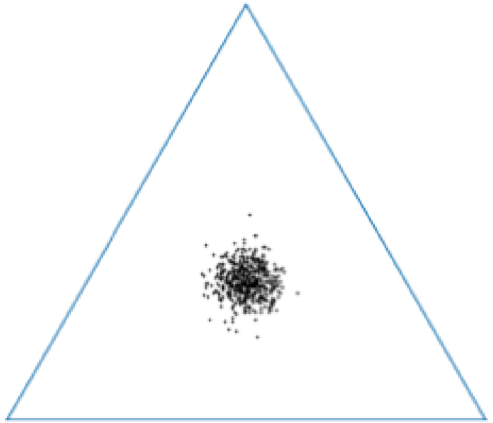
Iterative Uncertainty Methods

Calculating Uncertainty

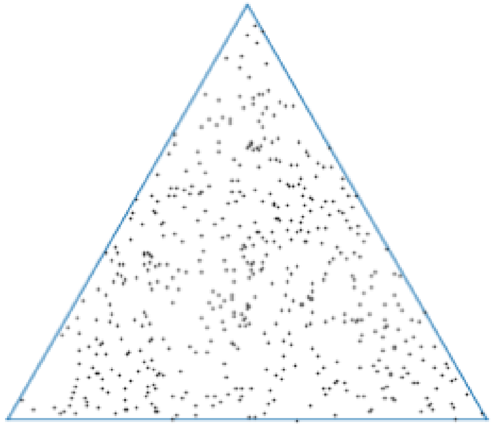
Confident



High Aleatoric Uncertainty



High Epistemic Uncertainty

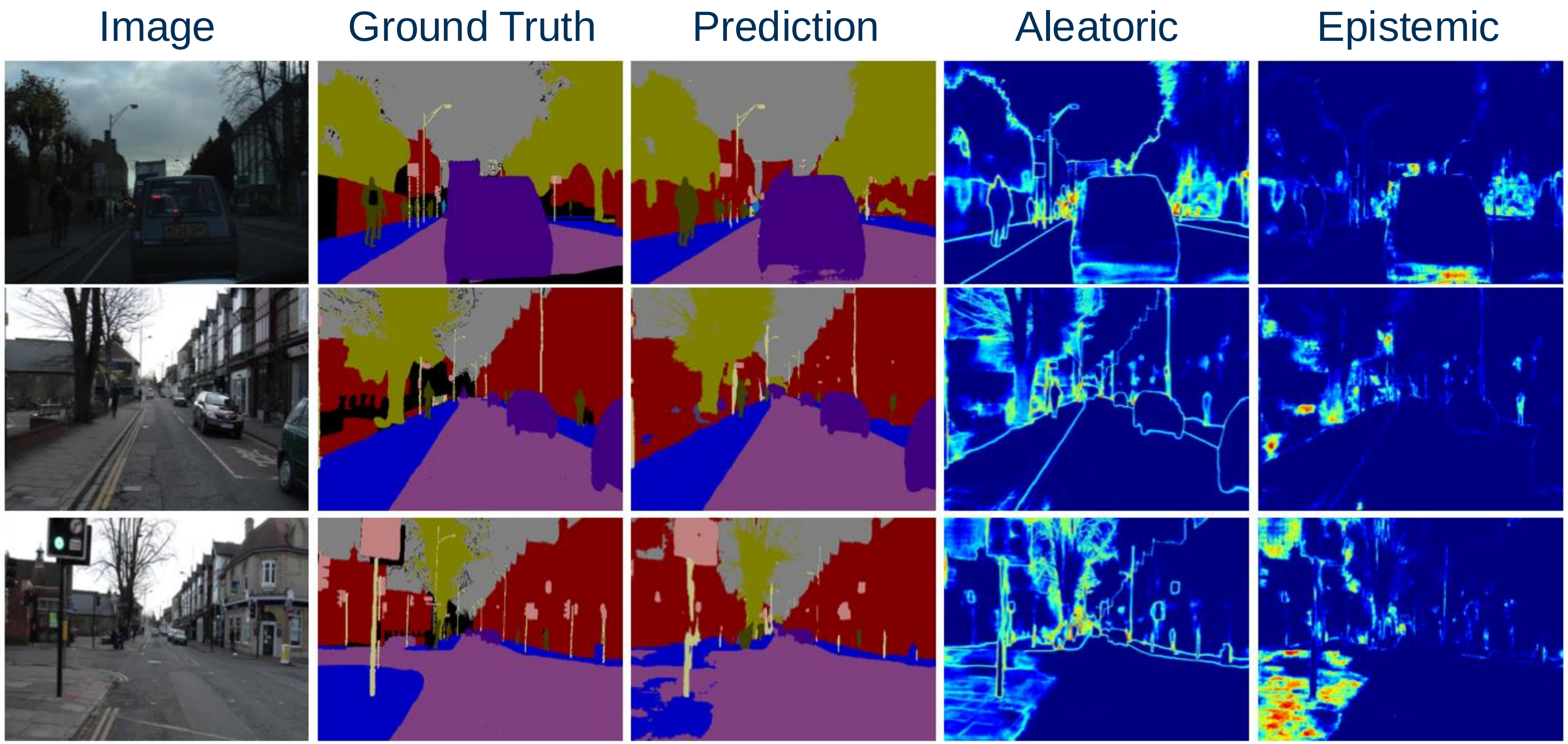


$$U_{epistemic} = \underbrace{H\left(\frac{1}{T}\sum_{t=1}^{T} Softmax\left(f_{\widehat{\boldsymbol{W}}_t}(\boldsymbol{x})\right)\right)}_{U_{total}} - \underbrace{\frac{1}{T}\sum_{t=1}^{T} H\left(Softmax\left(f_{\widehat{\boldsymbol{W}}_t}(\boldsymbol{x})\right)\right)}_{U_{aleatoric}}$$

Malinin, Andrey, Bruno Mlodozeniec, and Mark Gales. "Ensemble distribution distillation." *arXiv preprint arXiv:1905.00076* (2019).

Iterative Uncertainty Methods

Application: Uncertainty Quantification in Segmentation Applications



Kendall, Gal “What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision." *NIPS 2017*

Overview

In this Lecture..

Introduction and Motivation

Two Main Types of Uncertainty

- Aleatoric Uncertainty
- Epistemic Uncertainty

Iterative Uncertainty Estimation

Single Pass Uncertainty Estimation

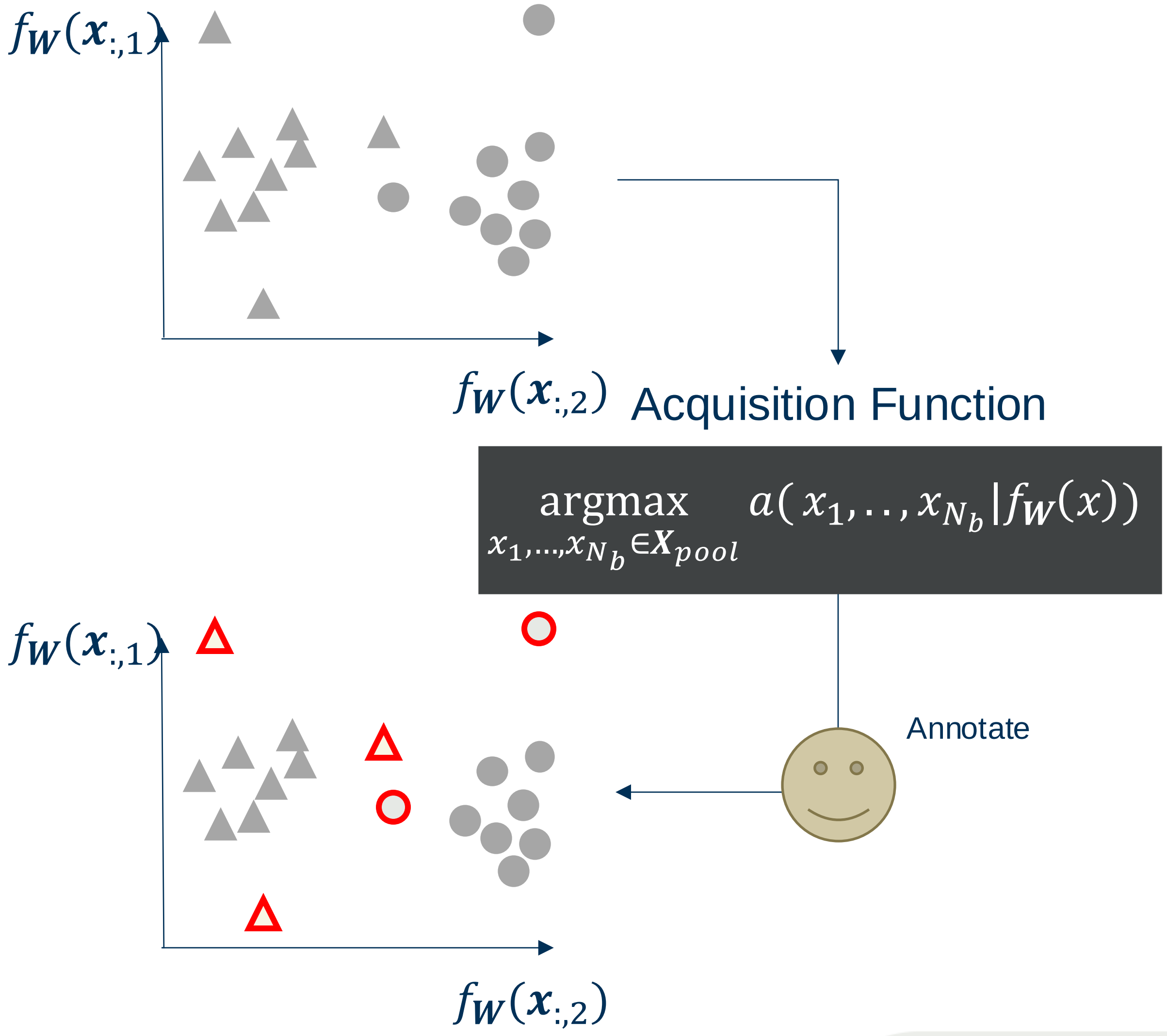
Performance Metrics

Single Pass Uncertainty Quantification Methods

Difficulty-based Methods

- The acquisition functions from Difficulty-based Active Learning measure uncertainty.

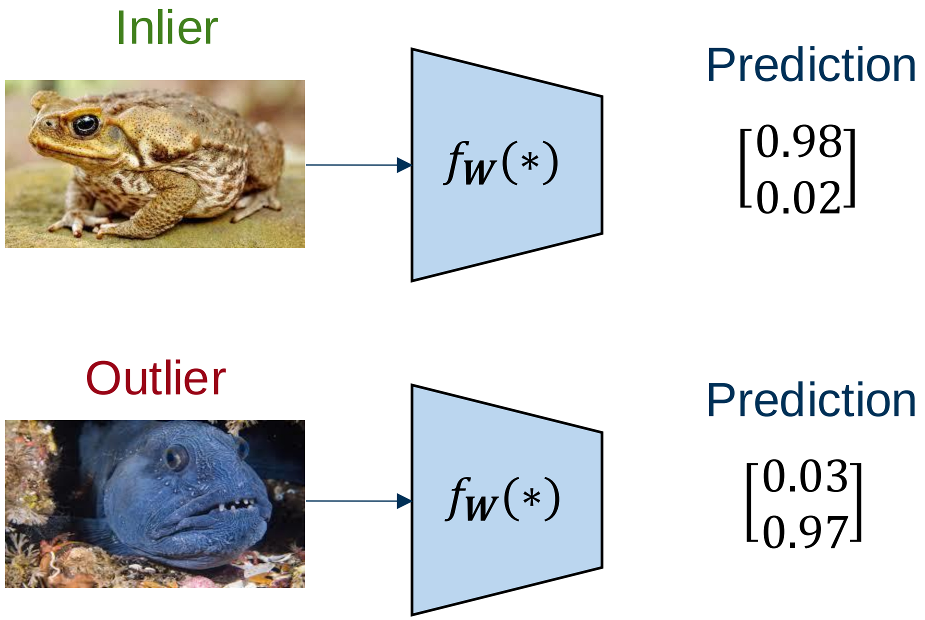
- Examples:
  - Entropy
  - Least confidence
  - Margin



Single Pass Uncertainty Quantification Methods

Difficulty-based Methods

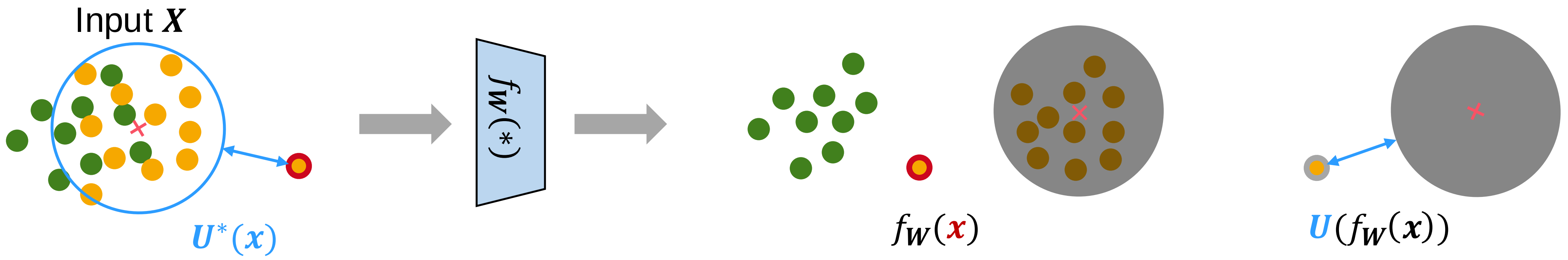
- The acquisition functions from Difficulty-based Active Learning measure uncertainty.
- Issues: difficulty-based scores are less effective when exposed to outlier samples. We say models are ”uncalibrated”



Single Pass Uncertainty Quantification Methods

Difficulty-based Methods

Typical Paradigm of Single Pass Uncertainty



$\boldsymbol{U}^*(\boldsymbol{x})$: uncertainty estimation of test point $\boldsymbol{x}$ drawn from the input space.
$f_W(\boldsymbol{x})$: latent representations of test point $\boldsymbol{x}$.
$\boldsymbol{U}(f_W(\boldsymbol{x}))$: uncertainty estimation from output space.

Liu, Jeremiah, et al. "Simple and principled uncertainty estimation with deterministic deep learning via distance awareness." Advances in Neural Information Processing Systems (2020)

Single Pass Uncertainty Reduction Methods

Distance-preservation Methods

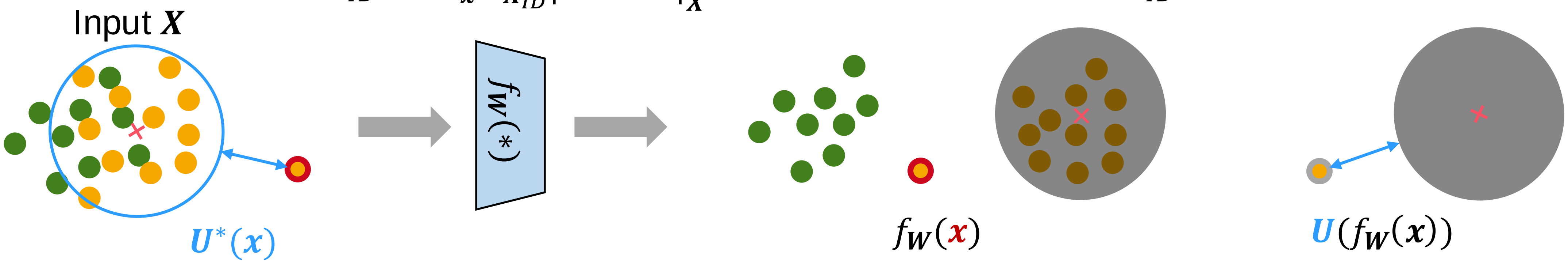
This is FYI only

Distance Awareness: Let $f_W(\boldsymbol{x})$ be a model conditioned on parameters $\boldsymbol{W} \sim p_o(\boldsymbol{W}|\boldsymbol{X}_{ID}, \boldsymbol{Y}_{ID})$ where p_o represents the weight posterior. We say f_W is **distance aware** if there exists a summary statistic $\boldsymbol{u_f}(*)$ of $f_W(\boldsymbol{x})$ that **quantifies uncertainty by reflecting the distance between testing x and the training data within the input manifold $\boldsymbol{X}_{ID}$**

$$u_f(\boldsymbol{x}) = v\big(d(\boldsymbol{x}, \boldsymbol{X}_{ID})\big)$$

v: Monotonic function

$d(\boldsymbol{x}, \boldsymbol{X}_{ID}) = E_{\boldsymbol{x}' \sim \boldsymbol{X}_{ID}} \big|\big|\boldsymbol{x} - \boldsymbol{x}'\big|\big|_{\boldsymbol{X}}$: Distance between $\boldsymbol{x}$ and $\boldsymbol{X}_{ID}$



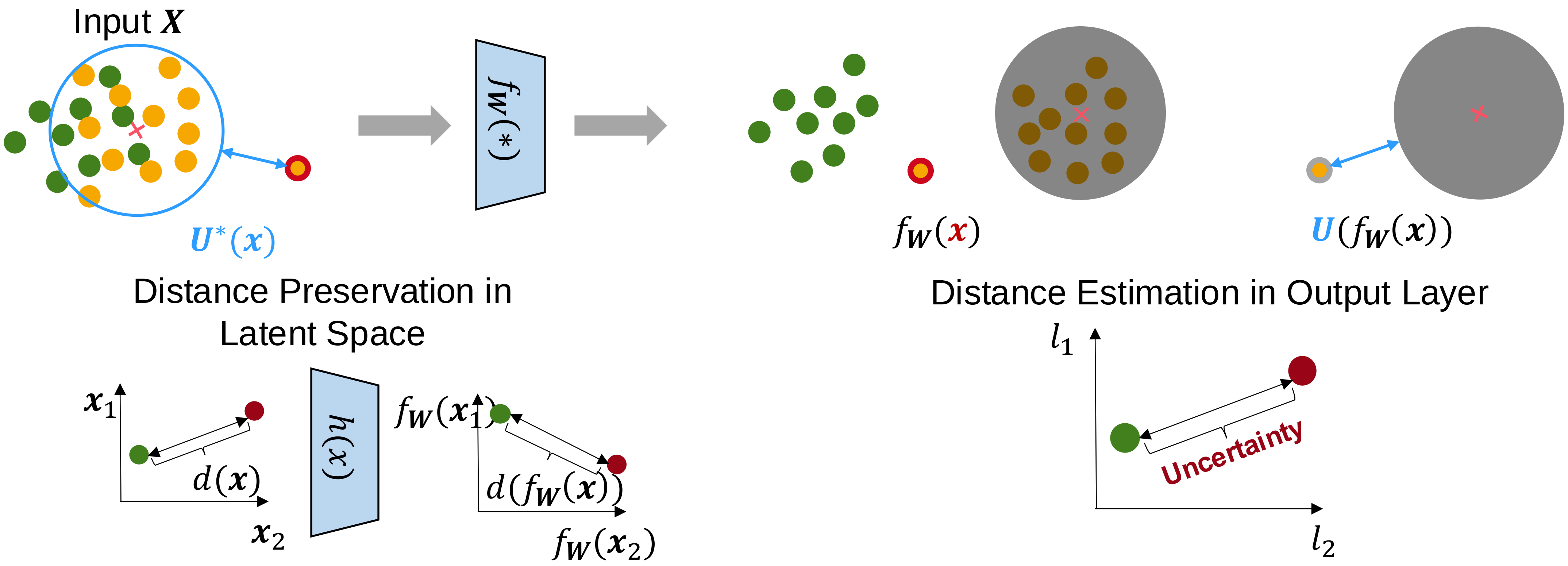
Liu, Jeremiah, et al. "Simple and principled uncertainty estimation with deterministic deep learning via distance awareness." Advances in Neural Information Processing Systems (2020)

Single Pass Uncertainty Reduction Methods

Distance-preservation Methods

This is FYI only

Requirements for High-quality Single Pass Uncertainty Estimation



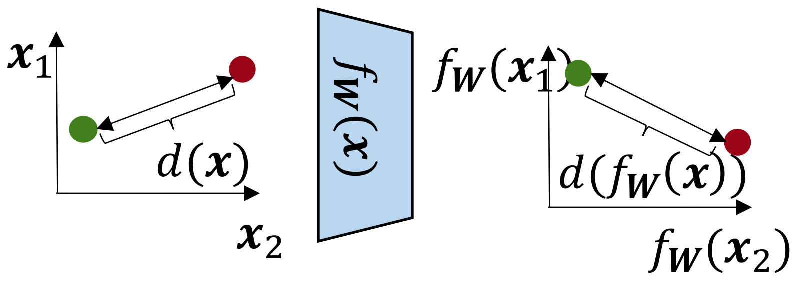
Liu, Jeremiah, et al. "Simple and principled uncertainty estimation with deterministic deep learning via distance awareness." Advances in Neural Information Processing Systems (2020)

Single Pass Uncertainty Reduction Methods

Distance-preservation Methods

This is FYI only

Distance Preservation

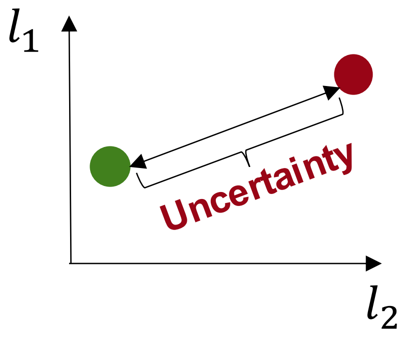


Enforce Bi-Lipschitz

$$L_1\|\boldsymbol{x}-\boldsymbol{x}'\|_X \leq \|f_W(\boldsymbol{x})-f_W(\boldsymbol{x}')\|_H \leq L_2\|\boldsymbol{x}-\boldsymbol{x}'\|_X$$

Ordinary DNNs need **improvement in distance-awareness**

Distance Estimation



$$U(\boldsymbol{x}) = v\big(d(\boldsymbol{x}, \boldsymbol{X}_{ID})\big)$$

v: Monotonic function

$d(\boldsymbol{x}, \boldsymbol{X}_{ID}) = E_{\boldsymbol{x}'\sim \boldsymbol{X}_{ID}}\left|\left|\boldsymbol{x}-\boldsymbol{x}'\right|\right|_{\boldsymbol{X}}$: Distance between $\boldsymbol{x}$ and $\boldsymbol{X}_{ID}$

Solution: Non-DL Algorithm
Most existing techniques in this area

Single Pass Uncertainty Reduction Methods

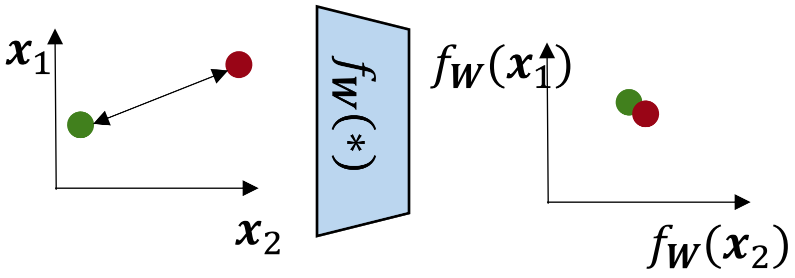
Sensitivity and Smoothness

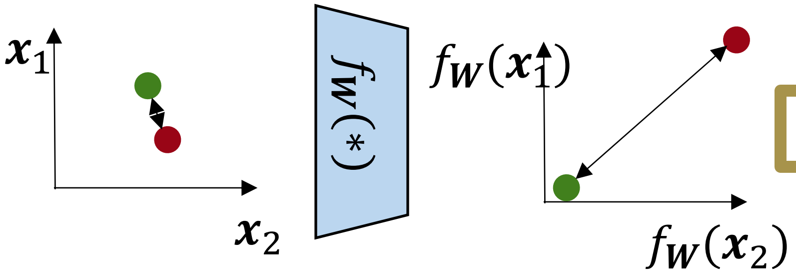
This is FYI only

Distance Preservation with Bi-Lipschitz Constraint

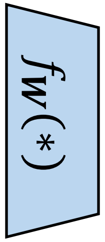
$$\underbrace{L_1\|\boldsymbol{x}-\boldsymbol{x}'\|_X}_{\text{Sensitivity}} \leq \|f_W(\boldsymbol{x})-f_W(\boldsymbol{x}')\|_H \leq \underbrace{L_2\|\boldsymbol{x}-\boldsymbol{x}'\|_X}_{\text{Smoothness}}$$

Not Sensitive





Not Smooth



$\begin{bmatrix}0.98\\0.02\end{bmatrix}$ Toad / Frog

1024 x 1024 x 3

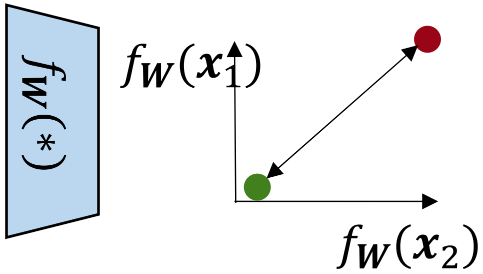
2 x 1

Midwife Toad [1]

Toad

False Toad [1]

Frog



Single Pass Uncertainty Reduction Methods

Sensitivity and Smoothness

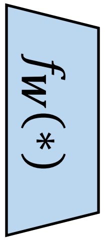
This is FYI only

Distance Preservation with Bi-Lipschitz Constraint

$$\underbrace{L_1\|\boldsymbol{x}-\boldsymbol{x}'\|_X}_{\text{Sensitivity}} \leq \|f_W(\boldsymbol{x})-f_W(\boldsymbol{x}')\|_H \leq \underbrace{L_2\|\boldsymbol{x}-\boldsymbol{x}'\|_X}_{\text{Smoothness}}$$

DNNs not always Sensitive

1024 x 1024 x 3



2 x 1

$\begin{bmatrix}0.98\\0.02\end{bmatrix}$ Toad / Frog

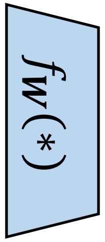
DNNs not always Smooth

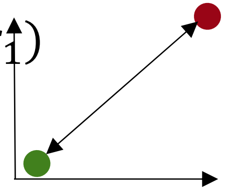
Midwife Toad [1]

Toad

False Toad [1]

Frog



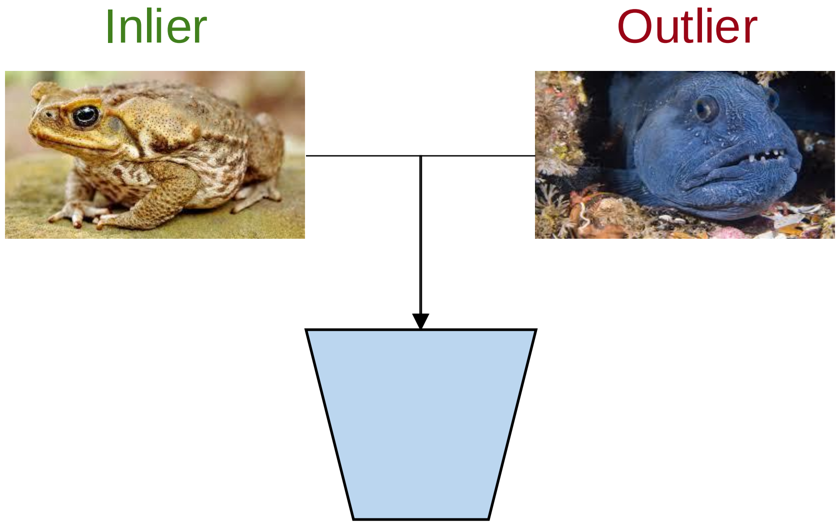


Sensitivity and **Smoothness** are not necessarily intended and possibly **hinder generalization**. Ideal methods manage distance preservation by **avoiding feature collapse of uncertainty information**

Single Pass Uncertainty Reduction Methods

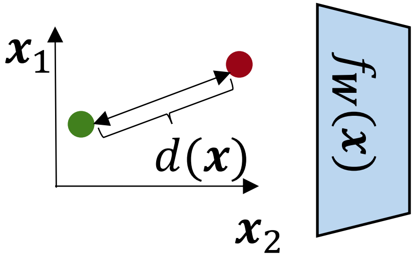
Sensitivity and Smoothness

This is FYI only

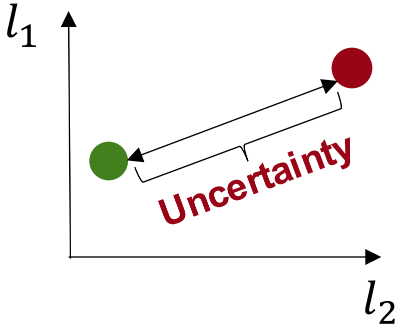


- **Definition:** Estimate uncertainty by enforcing **distance awareness** within the output representation.
- Idea: Enforce distance awareness within the representation through clever normalizations
- Examples: DUQ, SNGP, DUE, DEUP

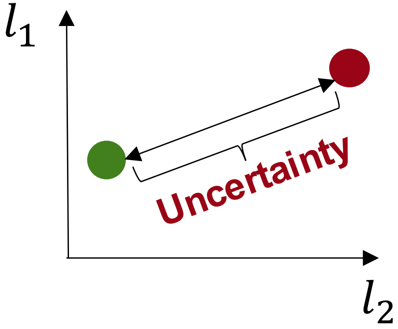
Distance Preservation in Latent Space



Distance Estimation in Output Layer



Representation



Spectral Normalization

This is FYI only

Distance Preservation with Bi-Lipschitz Constraint

$$\underbrace{L_1\|\boldsymbol{x}-\boldsymbol{x}'\|_X}_{\text{Sensitivity}} \leq \|f_W(\boldsymbol{x})-f_W(\boldsymbol{x}')\|_H \leq \underbrace{L_2\|\boldsymbol{x}-\boldsymbol{x}'\|_X}_{\text{Smoothness}}$$

Spectrally Normalized ResNet

$$(1-\alpha)^L\|\boldsymbol{x}-\boldsymbol{x}'\|_X \leq \|f_W(\boldsymbol{x})-f_W(\boldsymbol{x}')\|_H \leq (1+\alpha)^L\|\boldsymbol{x}-\boldsymbol{x}'\|_X$$

L: Number of Layers

$0 < \alpha < 1$

Single Pass Uncertainty Reduction Methods

Spectral Normalization

This is FYI only

Distance Preservation with Bi-Lipschitz Constraint

$$\underbrace{L_1\|\boldsymbol{x}-\boldsymbol{x}'\|_X}_{\text{Sensitivity}} \leq \|f_W(\boldsymbol{x})-f_W(\boldsymbol{x}')\|_H \leq \underbrace{L_2\|\boldsymbol{x}-\boldsymbol{x}'\|_X}_{\text{Smoothness}}$$

Spectrally Normalized ResNet

$$(1-\alpha)^L\|\boldsymbol{x}-\boldsymbol{x}'\|_X \leq \|f_W(\boldsymbol{x})-f_W(\boldsymbol{x}')\|_H \leq (1+\alpha)^L\|\boldsymbol{x}-\boldsymbol{x}'\|_X$$

L: Number of Layers

$$0 < \alpha < 1$$

Issues:

1. Spectral normalization enforces Bi-Lipschitz for **residual connections exclusively**
2. Lipschitz constants **scale** with the **number of layers** and **constrain less** for deeper models

Overview

In this Lecture..

Introduction and Motivation

Two Main Types of Uncertainty

- Aleatoric Uncertainty
- Epistemic Uncertainty

Iterative Uncertainty Estimation

Single Pass Uncertainty Estimation

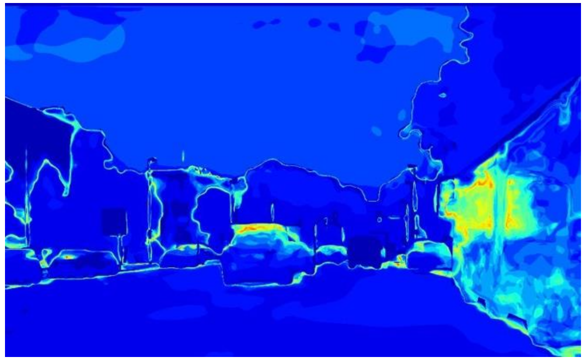
Performance Metrics

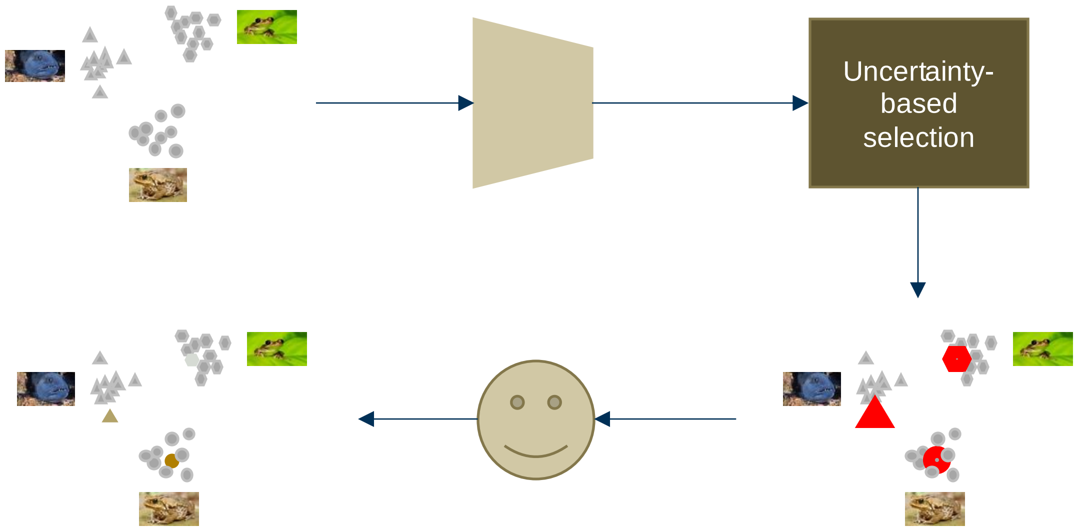
Performance Metrics

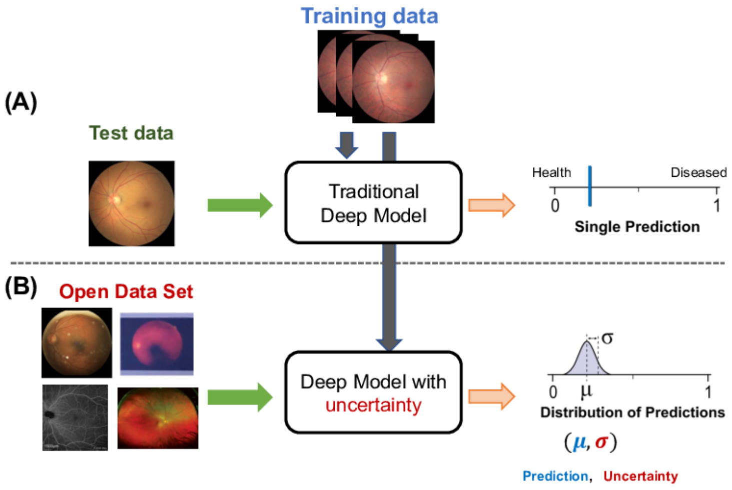
Summary

Active area of Research and no all-encompassing performance metrics

- Per-sample: NLL and Brier Score
- Per-dataset: Misprediction Detection
- Applications: Active Learning, Open-set Recognition etc.







Performance Metrics

Per-sample Uncertainty

NLL and Brier Score quantify uncertainty of prediction, when the true prediction is known

- Negative log-likelihood (NLL): Shows how likely a prediction is based on the evidence (data) conditioned on the learned parameters
  - Lower the NLL, lower the uncertainty
  - Derived from a Bayesian definition of probability
  - Typically works well for regression problems, but has since been adopted to deep neural networks with mixed results
  - Measures the ‘spread’ of the distribution

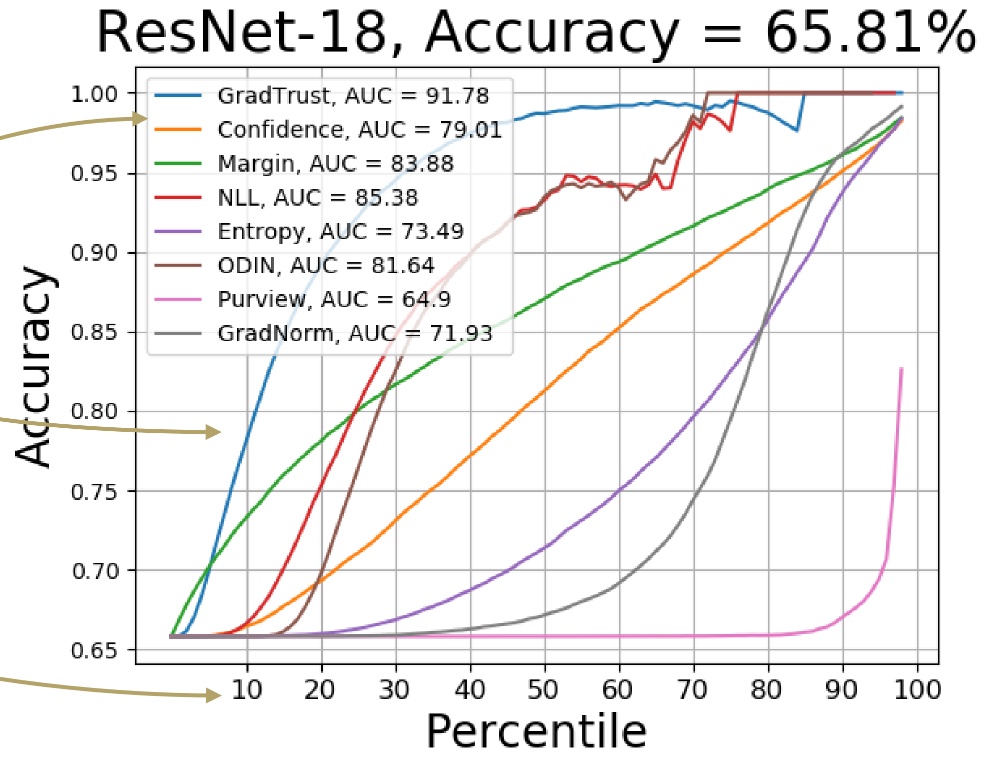
- Brier score: MSE between prediction and ground truth
  - MSE is generally the ‘hardest’ empirical loss function
  - Brier score measures how far away the prediction is from the GT
  - Higher the brier score, more the uncertainty

Performance Metrics

Per-dataset Uncertainty: Mispredicton Detection

For **ImageNet dataset** (with 50,000 validation set images):

1. **Run inference on all 50,000 images** and obtain GradTrust along with comparison trust scores
   - We compare against 8 other methods
2. **For each uncertainty,** order images in **ascending order**
3. For a given x **percentile**, calculate the **Accuracy** and F1 scores of all images above that percentile
4. Plot Area Under Accuracy Curve (AUAC) and Area Under F1 Curve (AUFC)
5. Repeat for multiple networks



Performance Metrics

Per-dataset Uncertainty: Mispredicton Detection

For Image Recognition Applications

Architecture	AUAC / AUFC								
	Softmax	Entropy	NLL	Margin [27]	ODIN [28]	MCD [12]	GradNorm [5]	Purview [4]	GradTrust
AlexNet [29]	72.86/68.43	65.02/62.14	**83.21/79.37**	79.04/73.3	79.22/75.89	54.2/51.59	58.85/55.28	50.14/48.92	**92.09/89.5**
MobileNet [30]	77.91/74.96	71.72/69.9	**84.02/81.37**	83.13/79.1	75.95/72.81	61.1/59.46	70.3/67.28	61.85/61.32	**93.37/90.58**
ResNet-18 [17]	79.01/76.13	73.49/71.71	**85.38/82.73**	83.88/79.87	81.64/79.26	62.91/61.4	71.93/69.29	64.9/64.01	**91.78/88.65**
VGG-11 [31]	79.95/77.02	74.33/72.52	**90.55/88.42**	84.85/80.77	85.08/83.33	63.19/61.62	73.16/70.06	65/63.84	**91.79/89.18**
ResNet-50 [17]	81.63/79.69	77.47/76.32	**89.23/86.47**	85.7/82.83	84.13/82.21	66.35/65.37	77.37/75.64	71.68/71.01	**92.24/90.09**
ResNeXt-32 [32]	81.56/79.97	78.11/77.15	**89.83/87.37**	85.16/82.81	82.77/80.43	66.9/66.09	78.61/77.28	74.06/73.05	**91.55/89.18**
WideResNet [33]	82.25/80.79	78.96/78.1	**90.84/88.42**	85.76/83.57	84.5/82.26	67.72/66.89	78.62/77.5	74.55/73.85	**91.36/89.12**
Efficient-v2 [34]	**91.49/87.84**	80.12/76.69	71.44/66.03	85.13/81.59	54.16/51.53	81.8/79.38	61.43/57.53	77.79/77.48	**93.57/89.61**
ConvNeXt-t [35]	88.17/86.21	85.56/83.88	79.19/76.85	**90.68/88.26**	62.51/60.74	85.43/83.82	70.86/66.25	79.16/78.91	**89.08/87.23**
ResNeXt-64 [32]	88.95/84.69	85.9/80.71	**90.04/87.06**	**91**/86.62	76.61/72.94	75.3/70.86	73.5/71.64	80.2/79.96	89.15/**87.41**
Swin-v2-t [36]	86.05/84.27	83.79/82.43	86.33/83.14	**88.75/86.29**	79.85/77.09	84.64/83.17	82.23/80.29	77.76/77.39	**87.45/85.23**
VIT-b-16 [37]	85.97/84.38	84.5/82.9	82.94/80.3	**88.67/86.5**	62.74/61,03	84.33/82.81	78.53/74.6	78.02/77.73	**87.77/85.85**
Swin-b [38]	86.18/84.49	84.77/83.14	79.18/75.52	**88.5/86.21**	68.07/64.59	84.69/83.17	83.09/81.52	80.71/80.45	**88.44/86.51**
MaxViT-t [39]	84.08/82.66	79.23/78.21	80.6/78.85	**85.84/84.02**	47.6/46.27	80.07/79.08	70.35/68.12	80.99/80.7	**90.19/88.48**

- **Negative Log Likelihood** (NLL) works well on smaller networks with **less accuracy** while **Margin classifier** works better with **high accuracy** networks
- **GradTrust performs well on all networks**

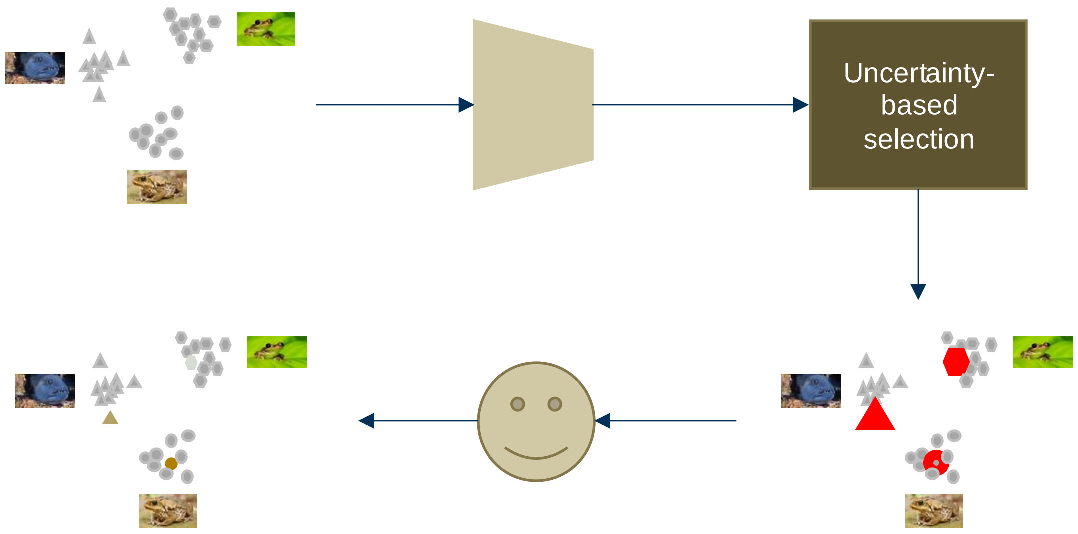
Prabhushankar, M., & AlRegib, G. (2024). Counterfactual Gradients-based Quantification of Prediction Trust in Neural Networks. *arXiv preprint arXiv:2405.13758.*

Performance Metrics

Applications: Active Learning

Many applications indirectly measure uncertainty

- Acquisition function is an ordered function of uncertainty quantification techniques
- Definition similar to 'difficulty-based' sampling from Lecture 25
- The performance metrics for AL indirectly measure the 'goodness' of acquisition function and hence Uncertainty quantification



- Other common applications: Open-set Recognition, Uncertainty visualization, latent space reconstruction etc.

Terminology

- *Distribution*: (sample space) the set of all possible samples
- *Dataset*: a set of samples drawn from a distribution
- *Batch*: a subset of samples drawn from the dataset
- *Sample*: a single data object represented as a set of features
- *Feature*: value of a single attribute, property, in a sample. Could be numeric or categorical.

Appendix A: Notations

- x_i: a single feature
- $\boldsymbol{x}_i$: feature vector (a data sample)
- $\boldsymbol{x}_{:,i}$: feature vector of all data samples
- $\boldsymbol{X}$: matrix of feature vectors (dataset)
- $\boldsymbol{W}$: weight matrix
- $\boldsymbol{Z}$: latent representation
- E_θ: encoding function
- G_ϕ: decoding function
- $\widehat{\boldsymbol{X}}$: reconstruction of data
- $\boldsymbol{\Omega}(\boldsymbol{Z})$: sparsity constraint
- $\hat{\rho}_{\boldsymbol{j}}$: average activation of neuron z_{ij}
- $\widetilde{\boldsymbol{X}}$:corrupted input
- N: number of data samples
- P: number of features in a feature vector
- $P^{(k)}$: the number of neurons in layer k
- α: learning rate
- Bold letter/symbol: vector
- Bold capital letters/symbol: matrix