Uncertainty Quantification in Deep Neural Networks

Alexey Zaytsev

14 March, 2025

The presentation extensively uses slides from Maxim Panov and Nikita Kotelevsky, and I am extremely grateful to them

Content

1. **Why do we need neural networks?**

2. **What is uncertainty**

- Definition of uncertainty
- Definition of uncertainty quantification
- Taxonomy: model and data uncertainty

3. **Road to uncertainty estimates for Neural Networks:**

- Linear models
- Pseudo-Bayesian methods: ensembles and sampling
- Deterministic uncertainty for neural networks

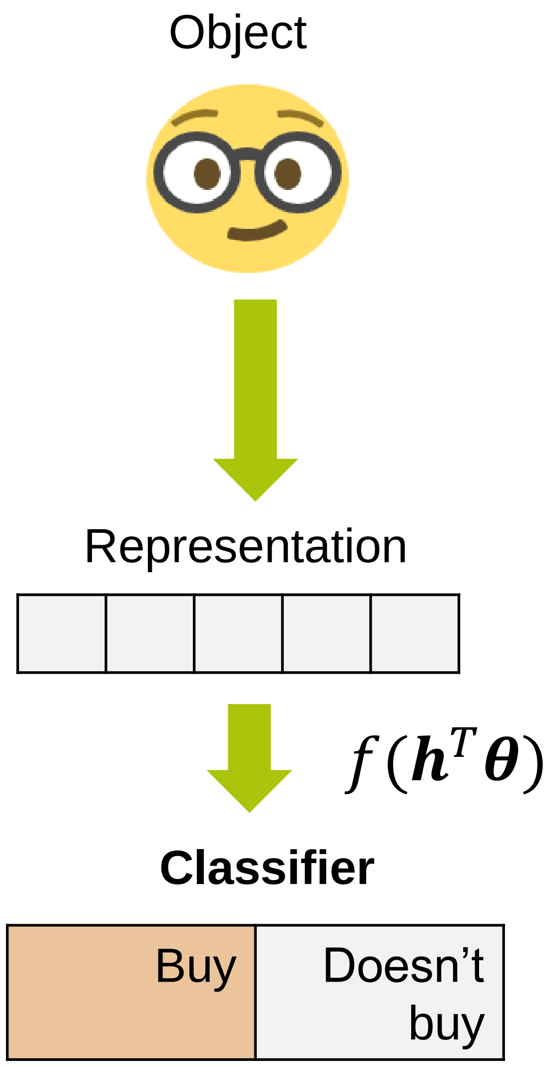
4. **Open problems**

Skoltech

Why do we need neural networks?

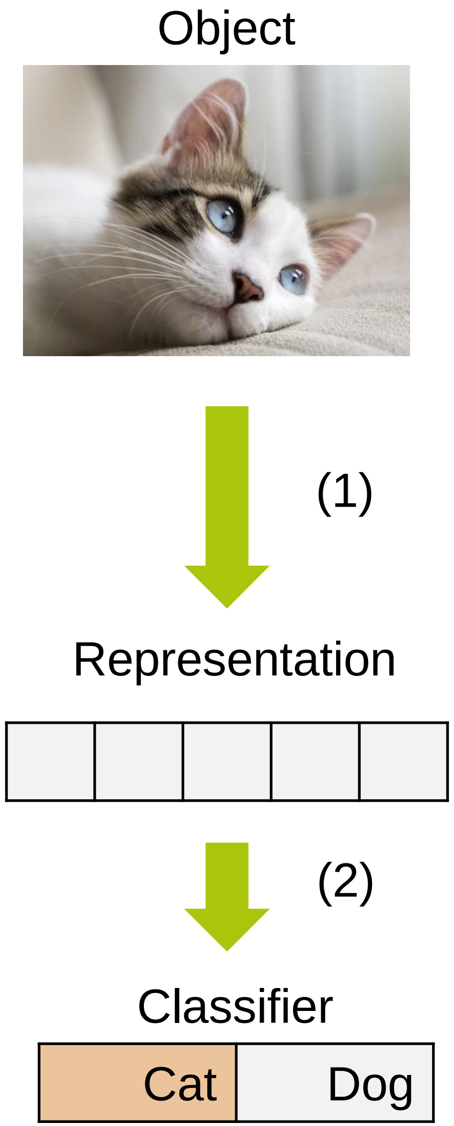
Classic machine learning, representations are available

$\boldsymbol{x}$ – an object	A human
y – a true label	Will buy a product in 3 months?
$\boldsymbol{h} = g(\boldsymbol{x})$ – an object representation	Salary, age
Problem: train a model that can identify the true class label $\approx y$	A logistic regression model $f(\boldsymbol{h}^T\boldsymbol{\theta})$ outputs a probability of a purchase



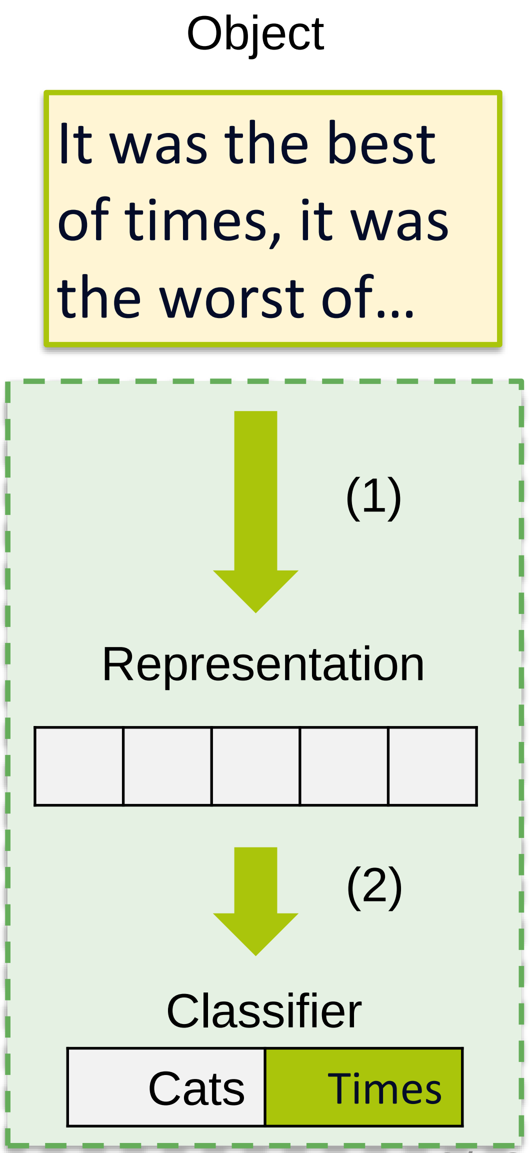
For structured data we need representation learning

$\boldsymbol{x}$ – an object	An image
y – a true label	Cat or dog?
$\boldsymbol{h} = g(\boldsymbol{x})$ – an object representation	NOT CLEAR, HOW TO DEFINE???
Problems: **(1) train a model's body that produces representation,** (2) train a model's head that predicts the true class;	<u>A neural network!</u>



Neural Large Language Models

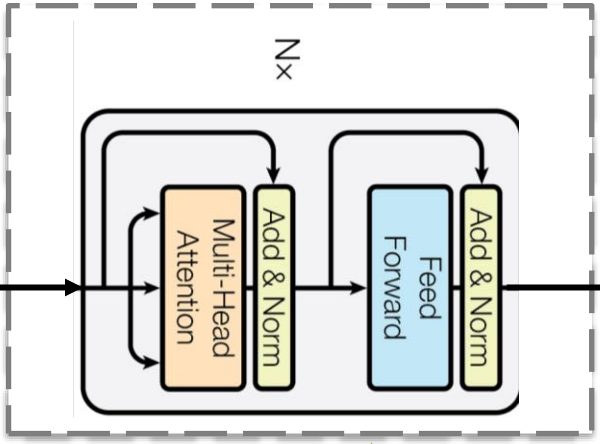
$\boldsymbol{x}$ – an object	A text
y – a true label	The next word
$\boldsymbol{h} = g(\boldsymbol{x})$ – an object representation	NEED TO TRAIN
Problems: **(1) train a model's body that produces representation,** (2) train a model's head that predicts the true class;	<u>A neural network language model!</u>



Structure of a neural network

Object, $\boldsymbol{x}$

It was the best of times, it was the worst of...



Body: $g_{\boldsymbol{\theta}_g}(\boldsymbol{x}) =$

$$g_k(\boldsymbol{\theta}_k; g_{k-1}(\boldsymbol{\theta}_{k-1}; \dots g_1(\boldsymbol{\theta}_1, \boldsymbol{x}) \dots))$$

Representation, $\boldsymbol{h}$

Head: $f_{\boldsymbol{\theta}_f}(\boldsymbol{h})$

Classifier

0.1	dog
0.3	cat
0.23	eye
0.14	ear
0.23	fox

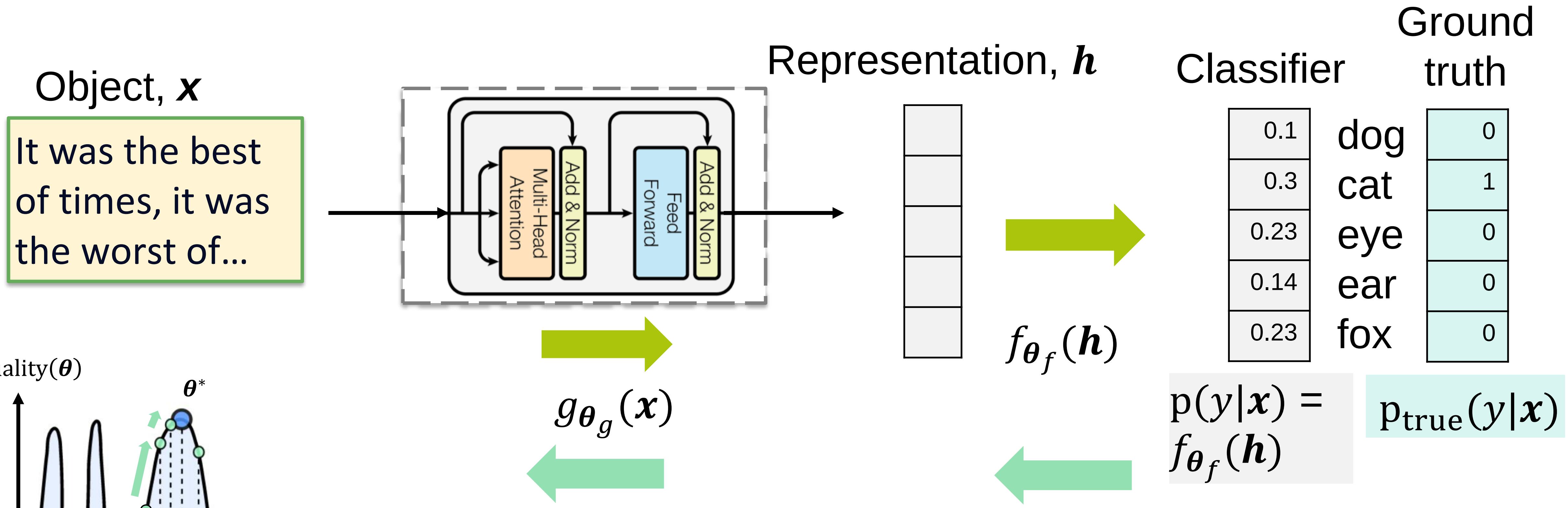
Output class probabilities

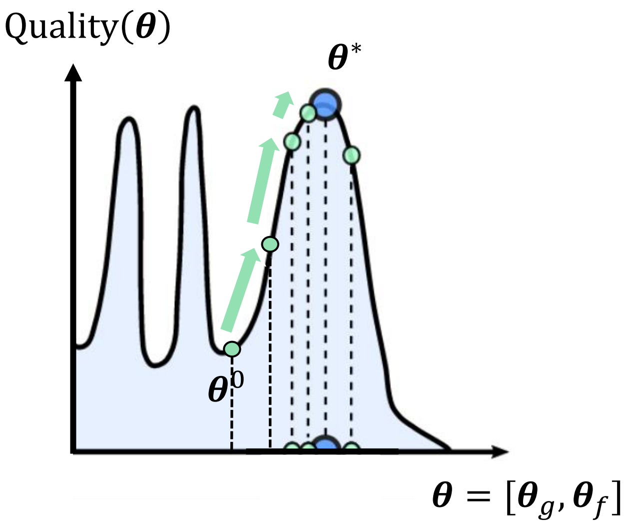
$$p(y|\boldsymbol{x})$$

BERT model: 345 millions of parameters in the body
Largest LLAMA model: 70 billions of parameters

Image credit: He, Kaiming, et al. "Deep residual learning for image recognition." *IEEE CVPR*. 2016.

Learning of a neural network

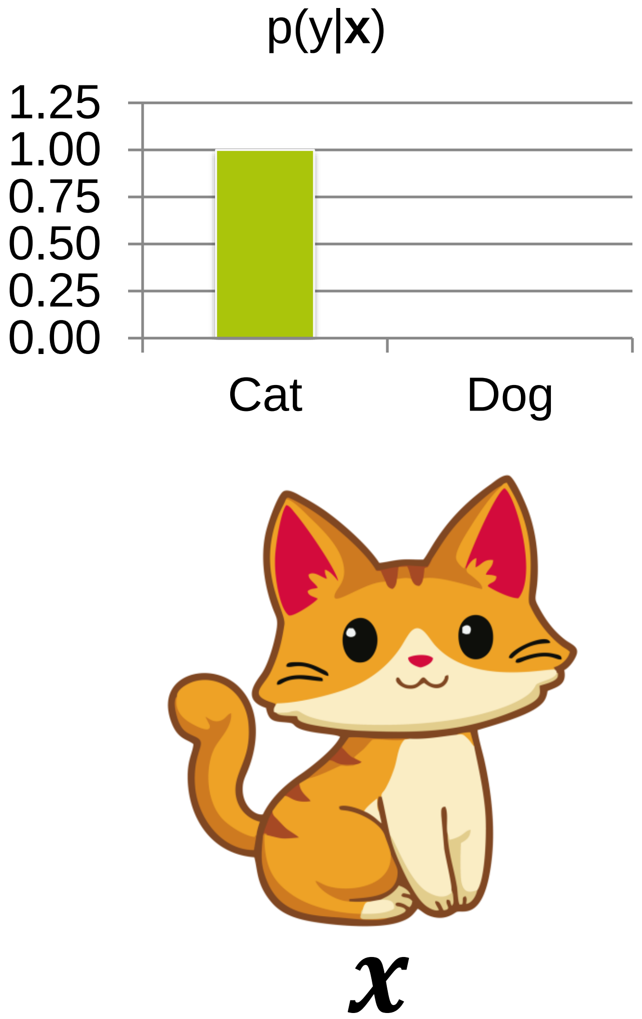


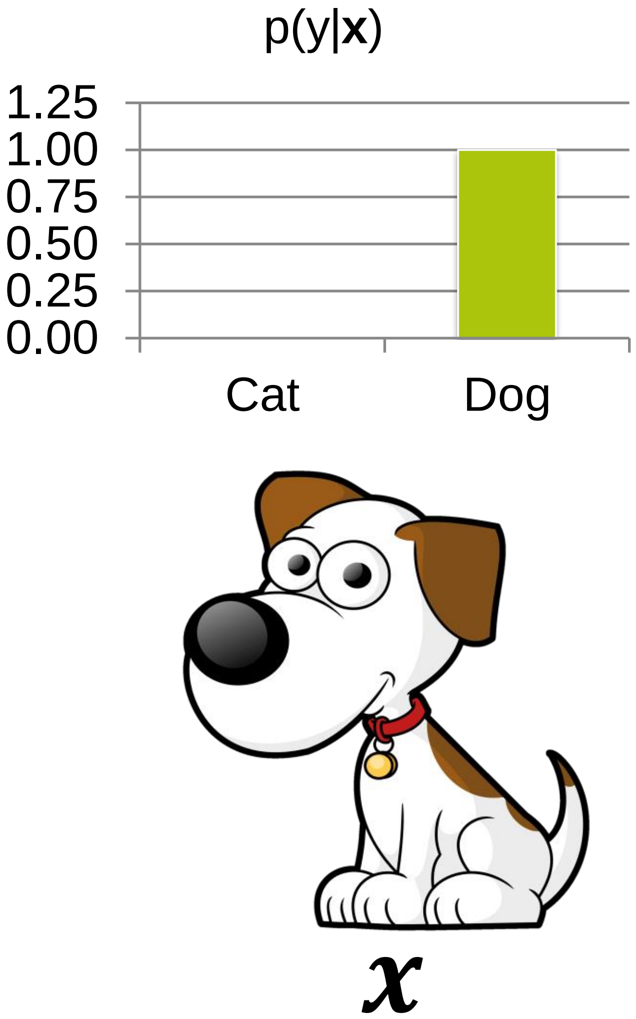


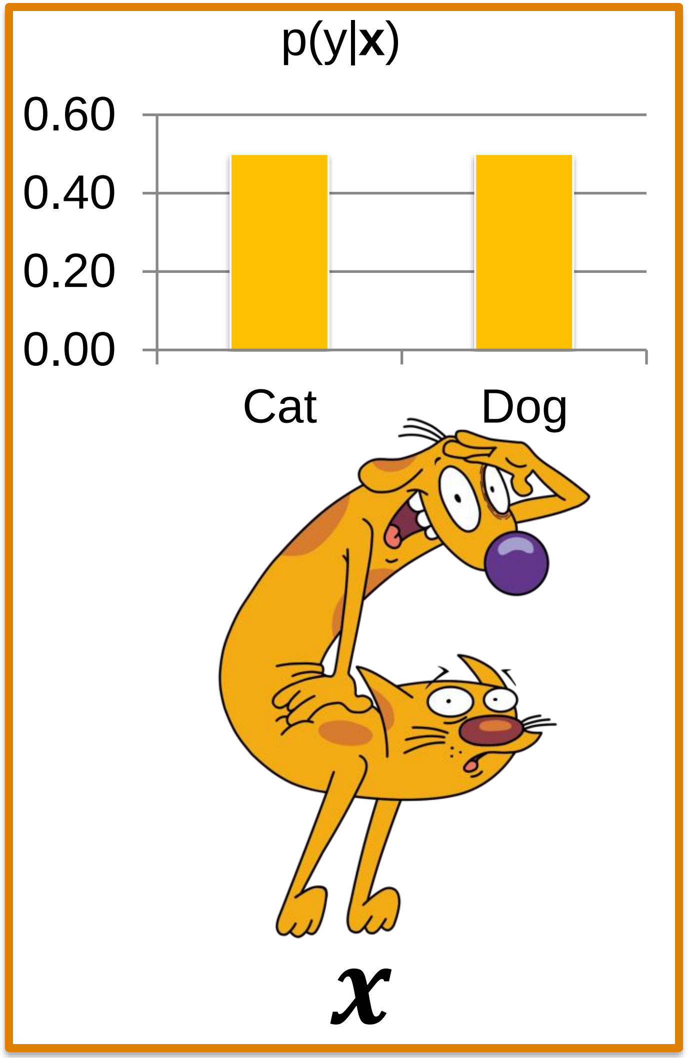
Learning is an adjustment of parameters, so the output distribution matches the true distribution

Image credit: He, Kaiming, et al. "Deep residual learning for image recognition." *IEEE CVPR*. 2016.

We (almost) never know the uncertainties in the ground truth







Credit: https://www.jamiesale-cartoonist.com/free-cartoon-dog-vector-clip-art
https://www.vecteezy.com/png/13078569-illustration-of-cute-colored-cat-cartoon-cat-image-in-png-format-suitable-for-children-s-book-design-elements-introduction-of-cats-to-children-books-or-posters-about-animal

Skoltech

What is uncertainty?

What is uncertainty quantification?

Life is uncertain

Image source: https://pixels.com/featured/the-tightrope-walker-g-roberto-weigand.html

Uncertainty in daily life examples

- **Weather forecasting** - is it going to rain?
- **Investment decisions** - will the stock go up or down?
- **Medical diagnosis** - what's the exact disease?

Skoltech

What is Uncertainty?

- **Uncertainty** refers to situations involving imperfect or unknown information.

- It’s a gap between what we know and what we don't know.

Skoltech

Go back to Neural Networks

- Consider a neural network making predictions: is it a cat or a dog in the image?

- Just like the examples mentioned, the network operates under unknown factors.

- For example, here the network predicts for irrelevant input.

$P(y = 1|x) = 0.9$

$y_{true} = 1$

$P(y = 1|x) = 0.2$

$y_{true} = 0$

$\boldsymbol{P(y = 1|x) = 0.8}$

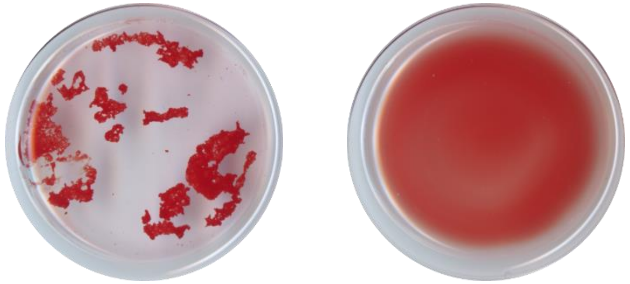
$y_{true} = ???$

Uncertainty in applications

- **Hallucinations in Large Language models** – how sure is a model about its’ output? What hallucinations we fight?

Source: OpenAI

- **Medical diagnosis** – should we retake a performed test?



Source: Synteco, IITP RAS

- **Active learning** – what data should we label to minimize labeling costs, if the labeling budget is limited?

Source: https://twitter.com/1evilidiot/status/794613309613821952

Why Uncertainty Quantification?

- By **quantifying** uncertainty, we can provide a measure of confidence in the network’s predictions.

- This measure helps users and developers trust the AI system and make informed decisions based on its outputs.

Skoltech

What do we mean by uncertainty?

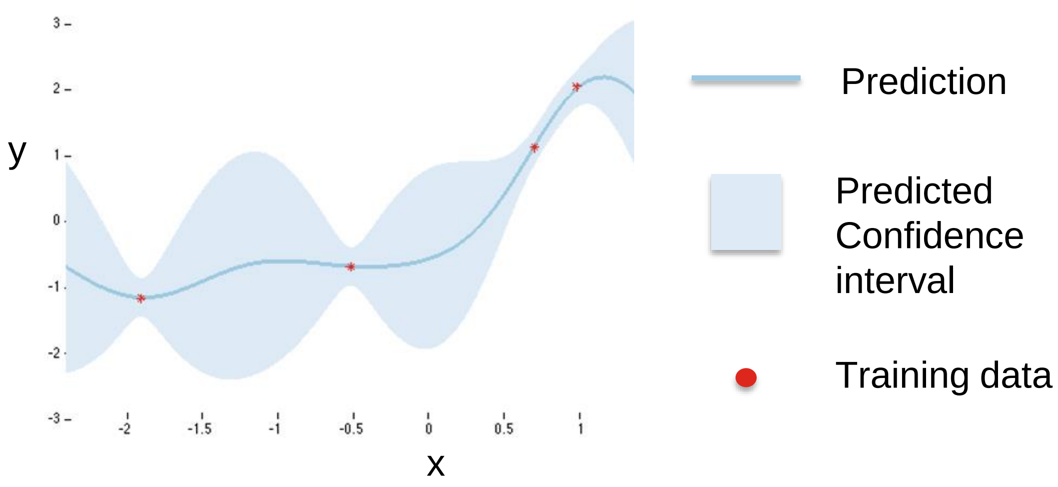
We would like to have **some value** *distribution over plausible predictions*, or *aggregated value*, which describes the "variability" over possible predictions.

Example:

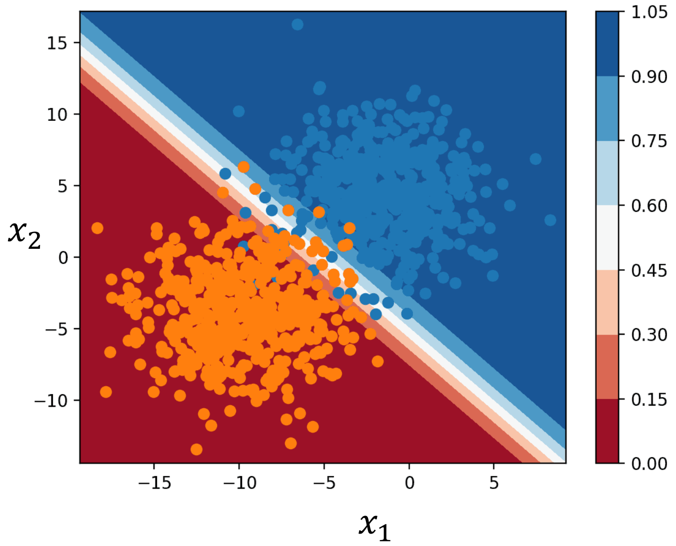
Regression: predict a mean with *the variance*

Classification: predict a label with *the confidence*

Regression



Classification

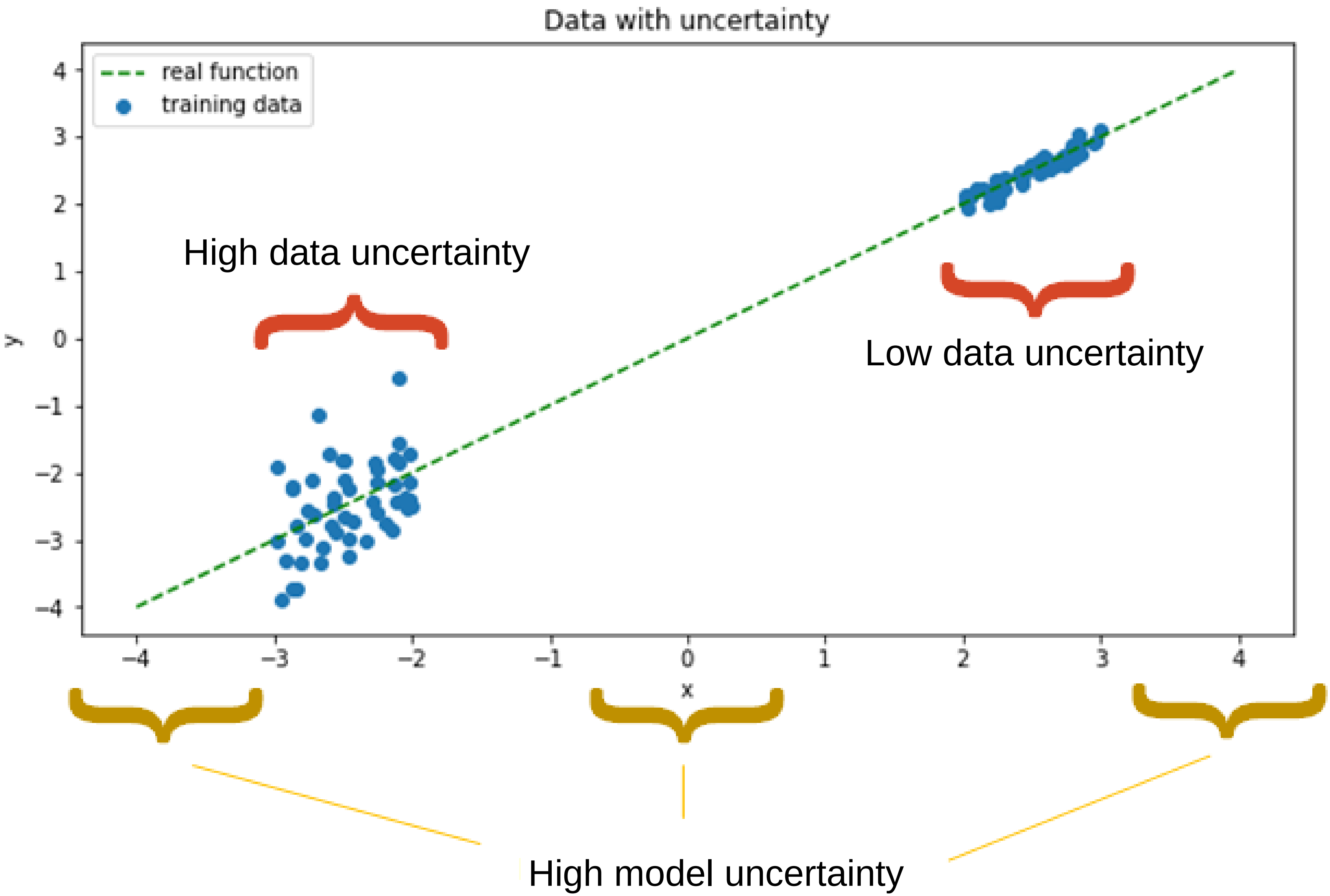


$$p(y = 1 \mid \boldsymbol{x})$$

Skoltech

Sources: https://jorisbaan.nl/2021/03/02/introduction-to-bayesian-deep-learning.html
https://machinelearningmastery.com/plot-a-decision-surface-for-machine-learning/

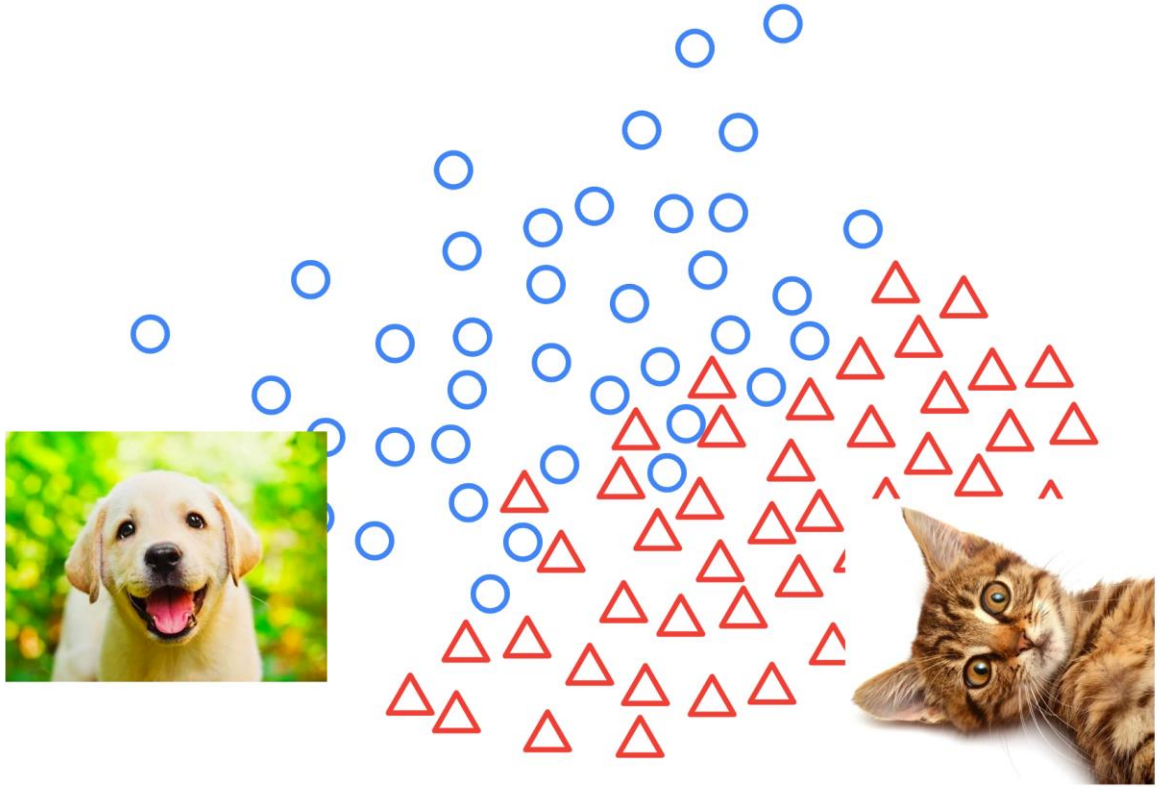
Model/data uncertainties for a regression problem



Skoltech

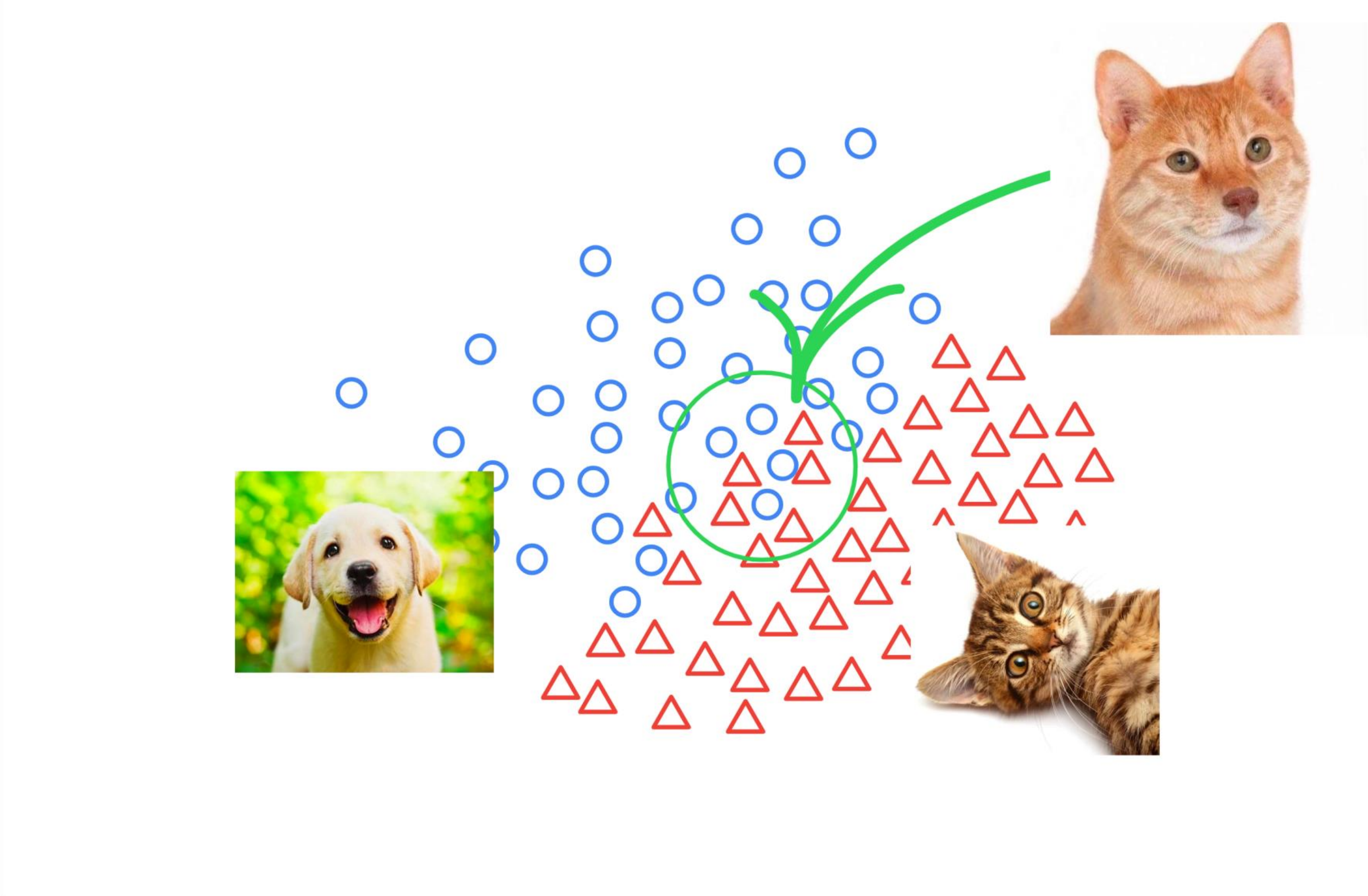
Source: https://towardsdatascience.com/my-deep-learning-model-says-sorry-i-dont-know-the-answer-that-s-absolutely-ok-50ffa562cb0b

Model/data uncertainties for a classification problem



Source: https://towardsdatascience.com/my-deep-learning-model-says-sorry-i-dont-know-the-answer-that-s-absolutely-ok-50ffa562cb0b

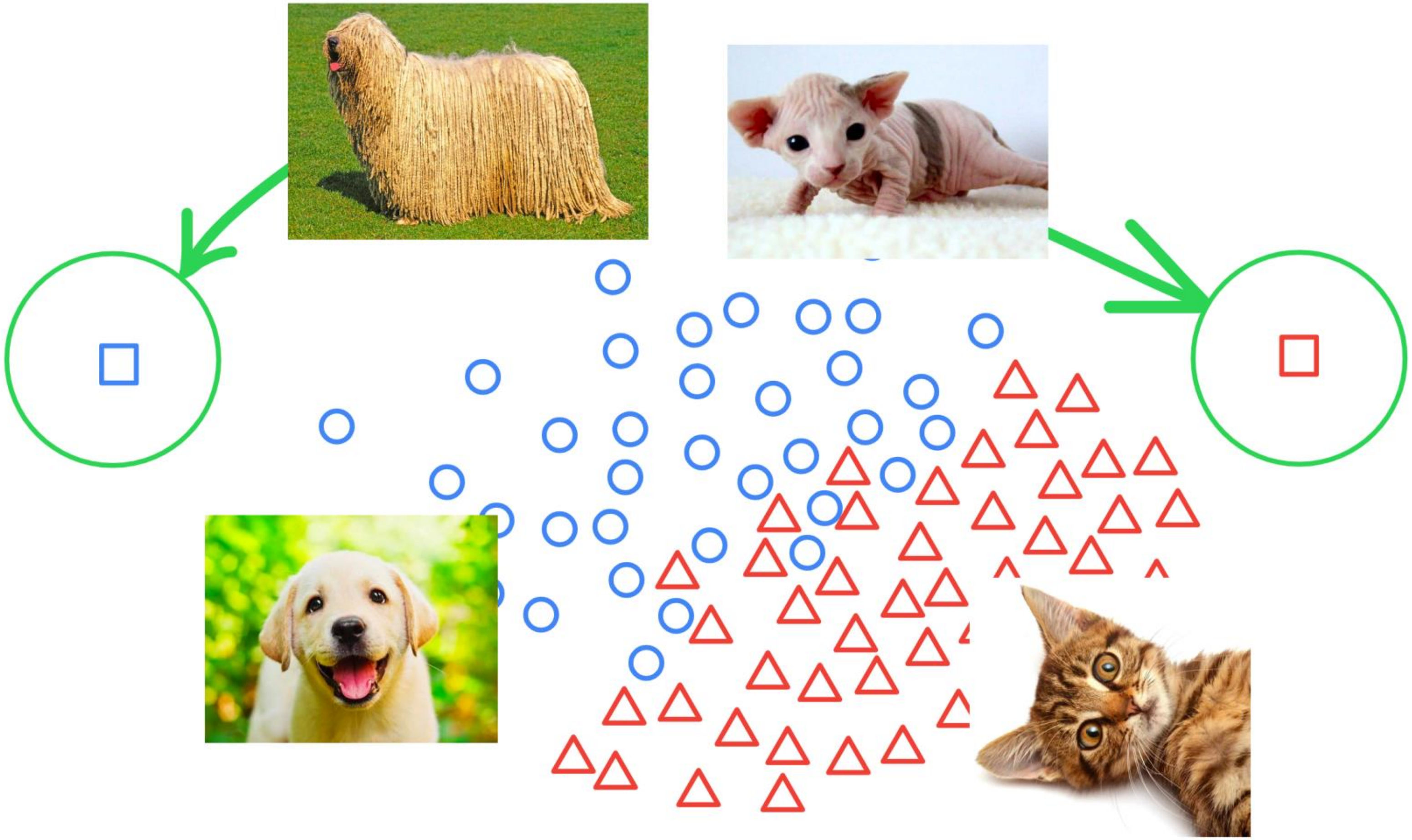
Model/data uncertainties for a classification problem



High data uncertainty

Source: https://towardsdatascience.com/my-deep-learning-model-says-sorry-i-dont-know-the-answer-that-s-absolutely-ok-50ffa562cb0b

Model/data uncertainties for a classification problem



High model uncertainty

What type of uncertainty do we need to estimate for active learning?

Skoltech

Source: https://towardsdatascience.com/my-deep-learning-model-says-sorry-i-dont-know-the-answer-that-s-absolutely-ok-50ffa562cb0b

Uncertainty evaluation for linear models

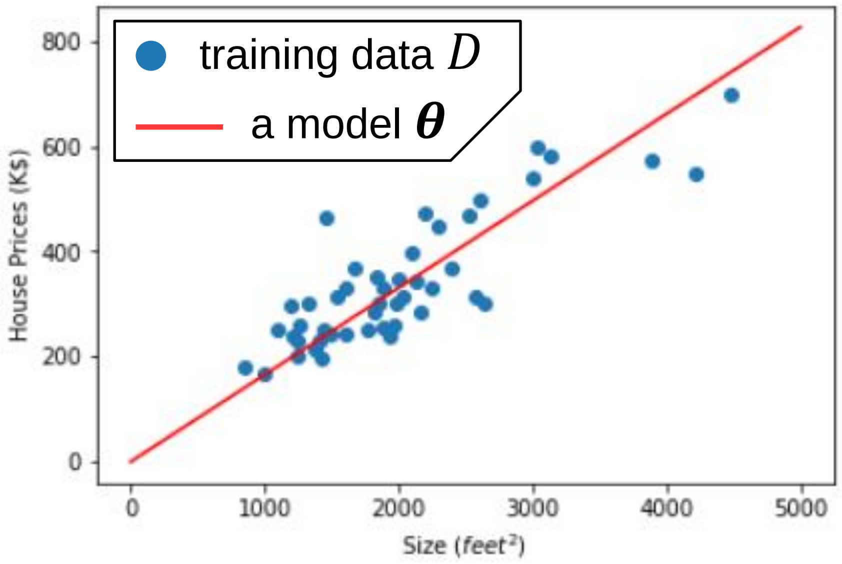
Bayesian linear regression model

We have training data $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n} = (X, \boldsymbol{y})$, $\boldsymbol{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R}$

Our assumptions about the model:

$$y = f_{\boldsymbol{\theta}}(\boldsymbol{x}) + \varepsilon = \boldsymbol{x}^T \boldsymbol{\theta} + \varepsilon, \varepsilon \sim N(0, \sigma^2), \boldsymbol{\theta} \sim N(\boldsymbol{0}, \sigma_\theta^2 \boldsymbol{I})$$

Example: a house price prediction y via the size of the house in squared feet $\boldsymbol{x}$



To define the model, we estimate $\boldsymbol{\theta}$ or a distribution of $\boldsymbol{\theta}$ given data D

Skoltech

Source: https://towardsai.net/p/l/univariate-linear-regression-from-scratch

A posterior distribution of parameters

For linear model $y = \boldsymbol{x}^T\boldsymbol{\theta} + \varepsilon$ we have prior and posterior distributions

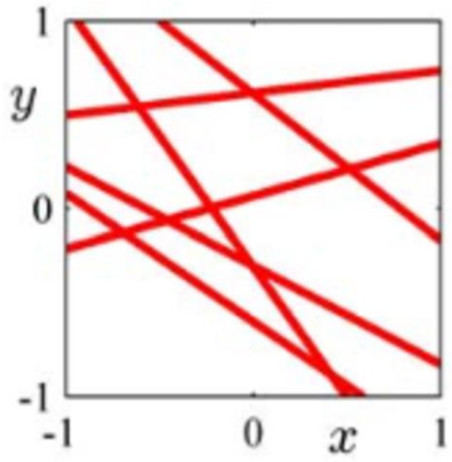
The prior distribution of parameters

$$\mathrm{p}(\boldsymbol{\theta}) = N(\boldsymbol{0}, \sigma_\theta^2 \boldsymbol{I})$$

The posterior distribution for parameters:

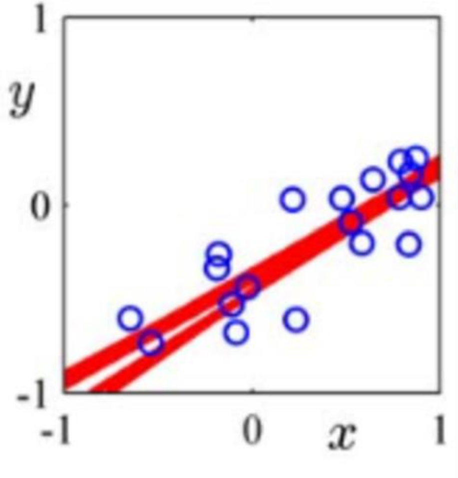
$$\mathrm{p}(\boldsymbol{\theta}|D) = N(\boldsymbol{\mu}, S),$$
$$\boldsymbol{\mu} = \sigma^{-2} S X^T \boldsymbol{y},$$
$$S^{-1} = \sigma^{-2} X^T X + \sigma_\theta^{-2} \boldsymbol{I}$$

The prior over functions



Training data D

The posterior over functions



Skoltech

Source: C.Bishop, Pattern recognition and machine learning, 2006

Predictive distribution

The predictive distribution corresponds to the predictions at a particular point, we predict not only a mean, but *a distribution.*

The predictive distribution:

$$\mathrm{p}(y|\boldsymbol{x}, D) = \int \mathrm{p}(y|\boldsymbol{x}, \boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta}|D)\mathrm{d}\boldsymbol{\theta}$$

As both distributions are Gaussian, we can take the integral to get:

$$\mathrm{p}(y|\boldsymbol{x}, D) = N\left(\mu(\boldsymbol{x}), \sigma^2(\boldsymbol{x})\right),$$

$$\mu(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{x}^T S X^T \boldsymbol{y},$$

$$\sigma^2(\boldsymbol{x}) = \boldsymbol{x}^T\left(\sigma^{-2}X^T X + \sigma_\theta^{-2}\boldsymbol{I}\right)^{-1}\boldsymbol{x} + \sigma^2$$

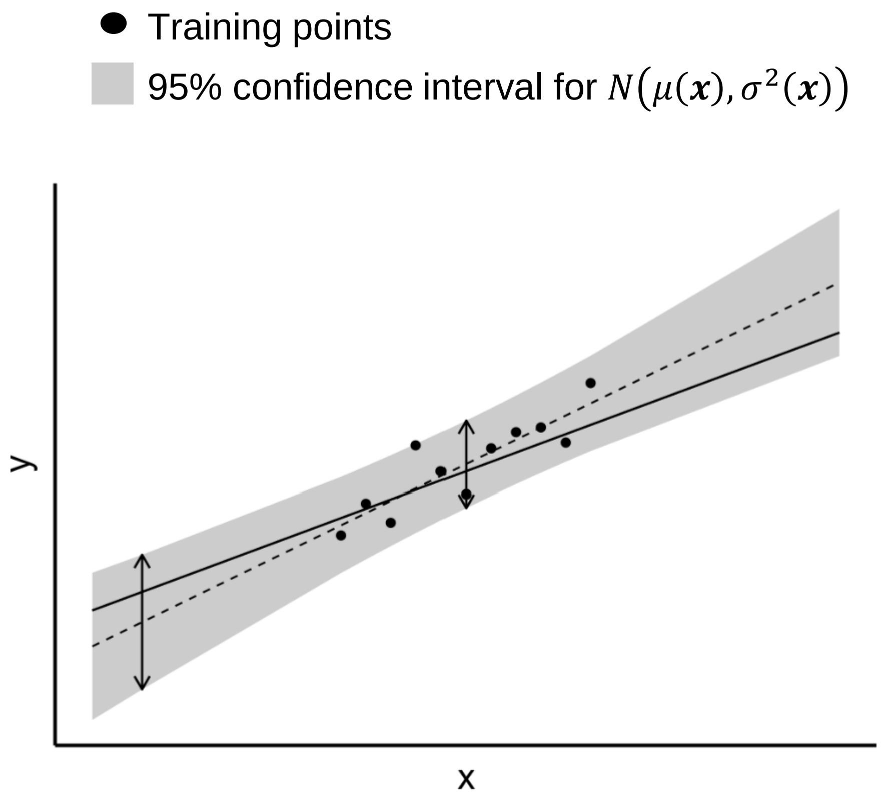
Skoltech

Uncertainty estimate for a predictive distribution

The variance has the analytical form for the linear regression model:

$$\sigma^2(\boldsymbol{x}) = \underbrace{\boldsymbol{x}^T\left(\sigma^{-2}X^TX + \sigma_\theta^{-2}\boldsymbol{I}\right)^{-1}\boldsymbol{x}}_{\text{Model uncertainty}} + \underbrace{\sigma^2}_{\text{Data uncertainty}}$$

We solved the problem, but the model is terribly wrong



Skoltech

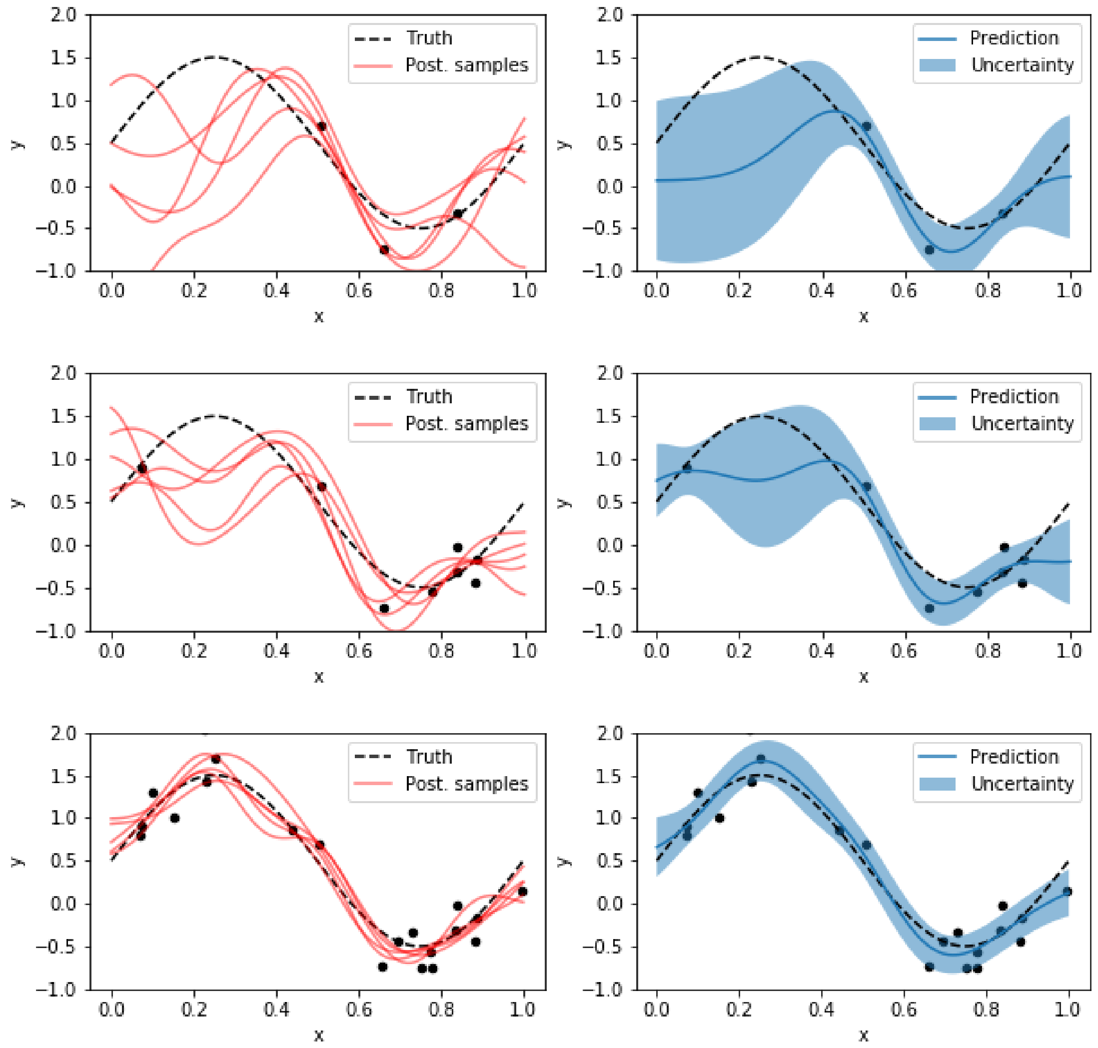
Source: Gruber, Cornelia, et al. "Sources of Uncertainty in Machine Learning--A Statisticians' View." *arXiv preprint arXiv:2305.16703* (2023).

Gaussian process regression

We define a distribution $\mathrm{p}(f)$ over a set of functions dense in $\mathrm{C}(\mathbb{R}^d)$

In the same way, using the prior distribution $\mathrm{p}(f)$ and the data D, we obtain a posterior over functions $\mathrm{p}(f|D)$

The predictive distribution has analytical form, if we approximate the posterior $\mathrm{p}(f|D)$ with a delta function



Samples from $\mathrm{p}(f|D)$

Posterior $\mathrm{p}(y|\boldsymbol{x}, \boldsymbol{\theta}, D)$

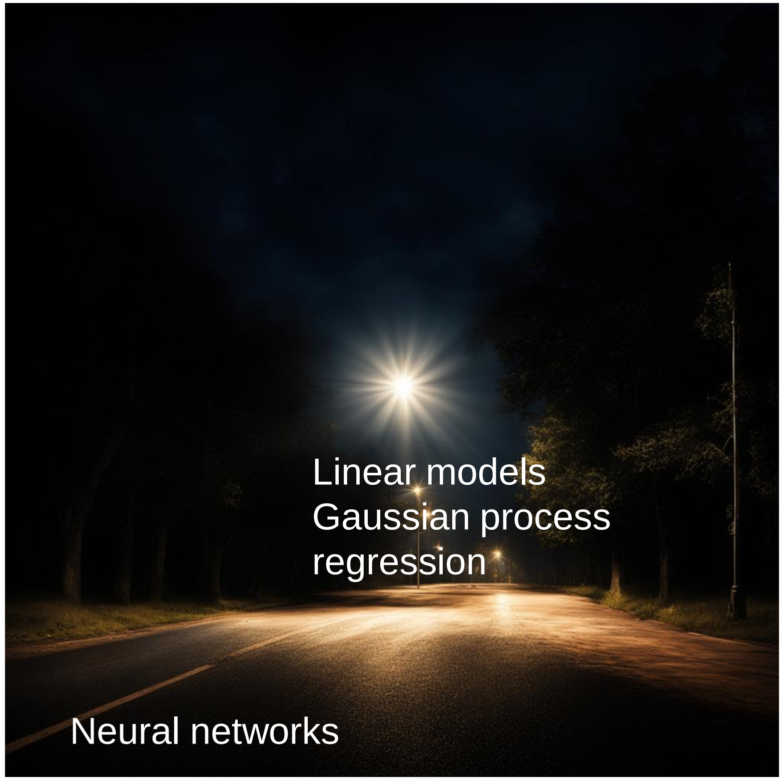
Sources: https://krasserm.github.io/2019/02/23/bayesian-linear-regression/
L. Vidhi, and C.E.Rasmussen. "Approximate inference for fully Bayesian Gaussian process regression." *Symposium on Advances in Approximate Bayesian Inference*. PMLR, 2020.

Good news

- Completely analytical form for the variance
- Rich space of functions

Bad news

- Approximation of the predictive distribution
- We model mostly *model* uncertainties
- **The model is wrong, neural networks work much better**



Streetlight effect for statistics

Skoltech

Pseudo-Bayesian methods via deep ensembles

Skoltech

Why?

In a Bayesian paradigm, we have distributions by construction.

They definitely reflect the uncertainties of predictions

Bayesian Inference for the predictive distribution

In Bayesian paradigm, inference is done via the Posterior Predictive Distribution:

$$\mathrm{p}(y \mid \boldsymbol{x}, D) = \int_\theta \mathrm{p}(y, \boldsymbol{\theta} \mid \boldsymbol{x}, D)\mathrm{d}\boldsymbol{\theta} = \int_\theta \mathrm{p}(y \mid \boldsymbol{\theta}, \boldsymbol{x})\mathrm{p}(\boldsymbol{\theta} \mid D)\mathrm{d}\boldsymbol{\theta}$$

However the integral is (almost) impossible to compute for neural networks.

Thus, **approximations should be used.**

Skoltech

Monte-Carlo approximation

To approximate

$$\mathrm{p}(y \mid \boldsymbol{x}, D) = \int \mathrm{p}(y \mid \boldsymbol{\theta}, \boldsymbol{x})\mathrm{p}(\boldsymbol{\theta} \mid D)\mathrm{d}\boldsymbol{\theta},$$

Monte-Carlo estimate is typically used:

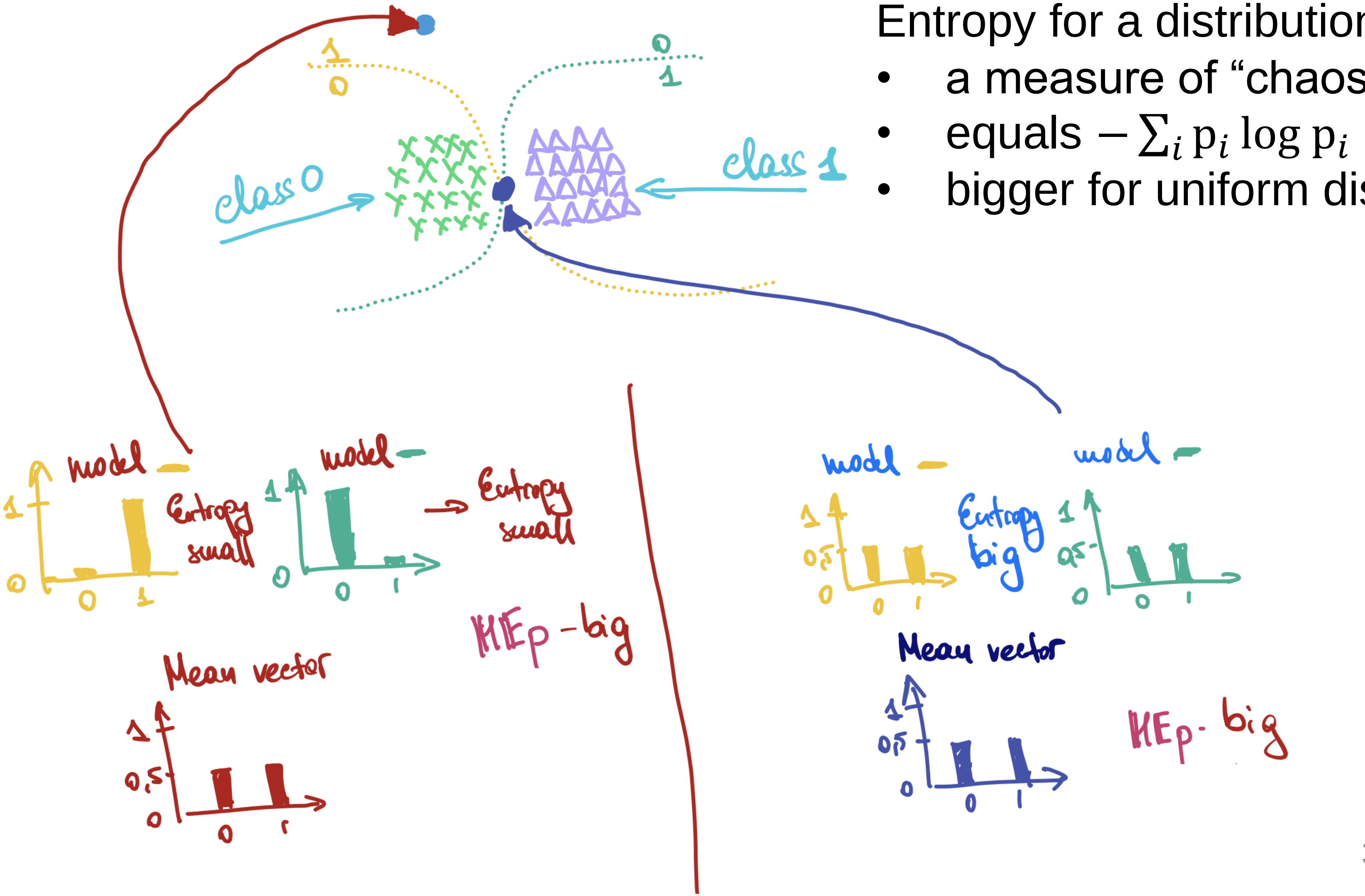
$$\mathrm{p}(y \mid x, D) \approx \frac{1}{K}\sum_{k=1}^{K}\mathrm{p}(y \mid \boldsymbol{\theta}_k, \boldsymbol{x}) = \int \mathrm{p}(y \mid \boldsymbol{\theta}, \boldsymbol{x})\mathrm{p}_K(\boldsymbol{\theta} \mid D)\mathrm{d}\boldsymbol{\theta},$$

where $\boldsymbol{\theta}_k$ are samples from the posterior distribution over model parameters to from the empirical distribution $\mathrm{p}_K(\boldsymbol{\theta} \mid D)$:

$$\boldsymbol{\theta}_k \sim \mathrm{p}(\boldsymbol{\theta} \mid D)$$

Skoltech

Example for measures of uncertainty



Entropy for a distribution $\{\mathrm{p}_i\}_{i=1}^{m}$:

- a measure of “chaos”
- equals $-\sum_i \mathrm{p}_i \log \mathrm{p}_i$
- bigger for uniform distributions

Skoltech

Different types of Uncertainty

We derive the following quantities:

Entropy of expected prediction - $\mathbb{H}_{\mathrm{p}}\mathbb{E}_{\boldsymbol{\theta}}\, \mathrm{p}(y \mid \boldsymbol{\theta}, \boldsymbol{x})$ - **total uncertainty**

Expected entropy - $\mathbb{E}_{\boldsymbol{\theta}}\mathbb{H}_{\mathrm{p}}\, \mathrm{p}(y \mid \boldsymbol{\theta}, \boldsymbol{x})$ - **data uncertainty**

BALD - $\mathbb{H}_{\mathrm{p}}\mathbb{E}_{\boldsymbol{\theta}}\mathrm{p}(y \mid \boldsymbol{\theta}, \boldsymbol{x}) - \mathbb{E}_{\boldsymbol{\theta}}\mathbb{H}_{\mathrm{p}}\mathrm{p}(y \mid \boldsymbol{\theta}, \boldsymbol{x})$ - **model uncertainty**

Note, that these measures can be computed only when we have different samples. Otherwise entropies coincide.

Skoltech

Sampling of models

Given samples $\boldsymbol{\theta}_k$, we can approximate all these expectations for uncertainty measures.

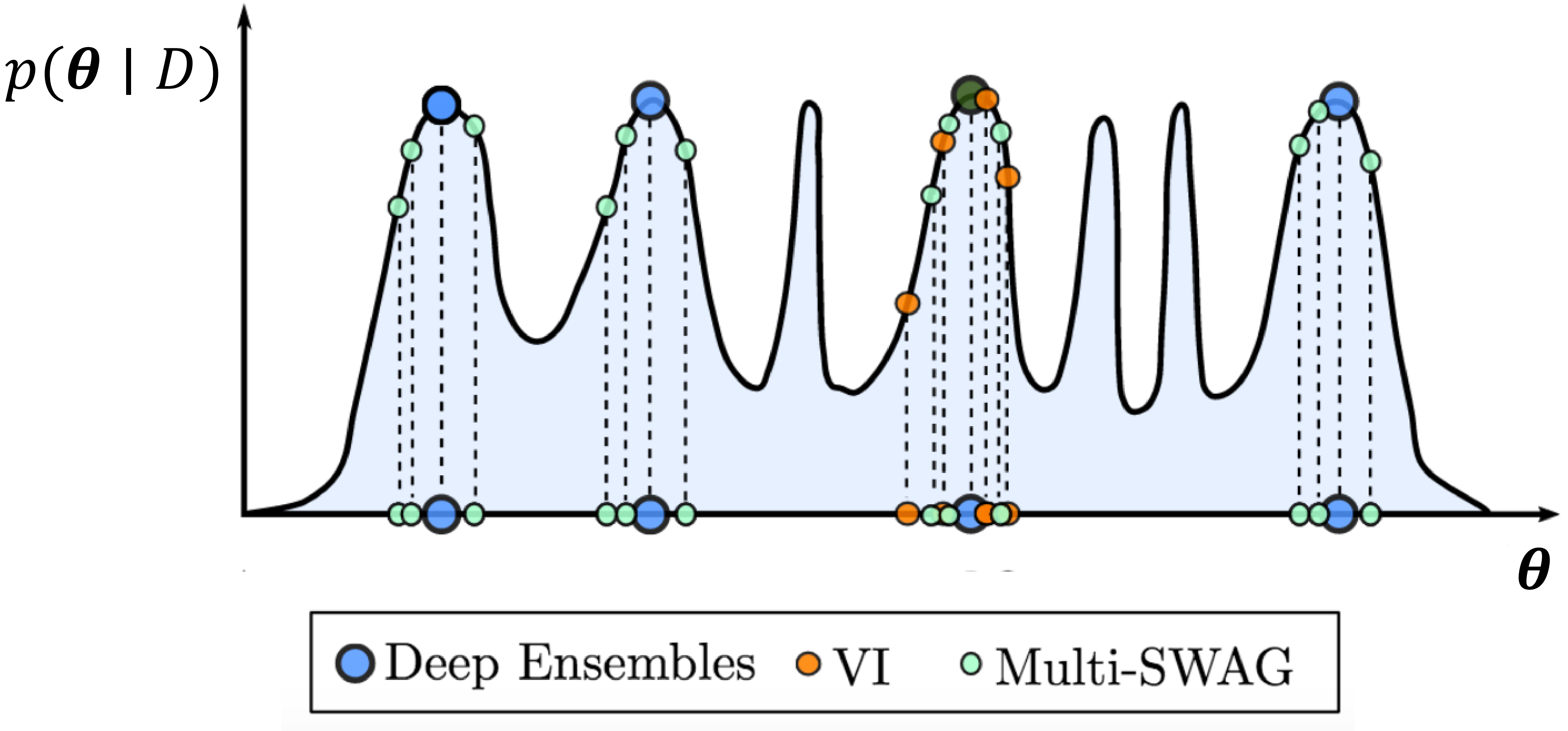
But how to receive these samples?

$$\boldsymbol{\theta}_k \sim \mathrm{p}(\boldsymbol{\theta} \mid D)$$

1. **Multistart from different seeds**
2. **Variational Inference (VI)**
3. **Markov Chain Monte Carlo (MCMC), SWAG**

Skoltech

Stochastic Weight Averaging Gaussian (SWAG) is a better way to sample an ensemble



Skoltech

Source: Wilson, A.G., and Izmailov P. "Bayesian deep learning and a probabilistic perspective of generalization." *NeurIPS*. 2020.

Deep Ensembles

Ensemble’s pros:

- Accuracy is better than for the single model
- Allows to compute measures of uncertainty

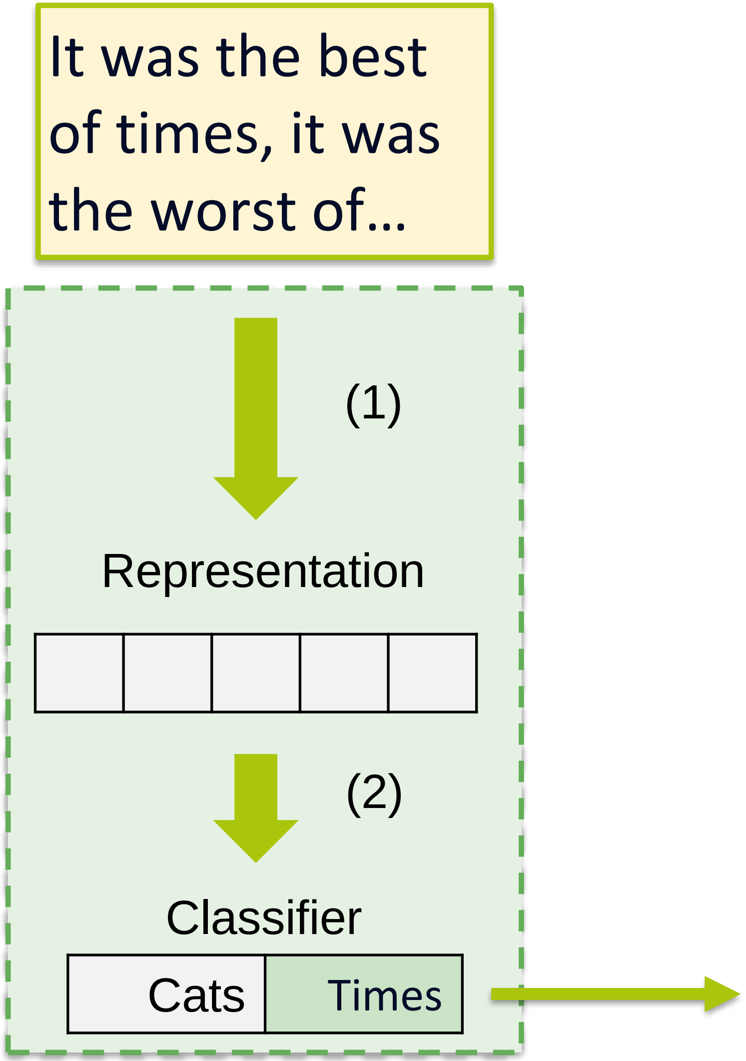
Ensemble’s cons:

- Training is K times **more expensive**
- Inference is K times **more expensive**
- Storage is K time **more expensive**

Skoltech

Uncertainty estimation via a single language model: an efficient alternative

Classification baseline, maxprob

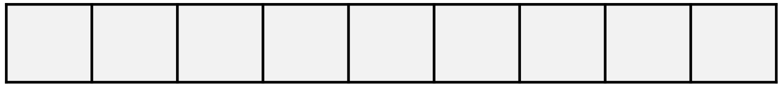


$$\text{uncertainty} = 1 - \max_i \mathrm{p}_i$$

Also: entropy $-\sum_i \mathrm{p}_i \log \mathrm{p}_i$

$$\text{certainty} = \max_i \mathrm{p}_i$$

$\{\mathrm{p}_i\}_{i=1}^{m}$ *m* is the dictionary size



Density-based UE Methods

Provide **high-quality UEs**, introduce **low computational overhead**, almost **no additional memory footprint**

- ↓ Deterministic uncertainty quantification (DUQ) [1]
- ↓ Mahalanobis distance [2]
- ↓ Spectral-normalized Neural Gaussian Process [3]

UNIVERSITY OF OXFORD

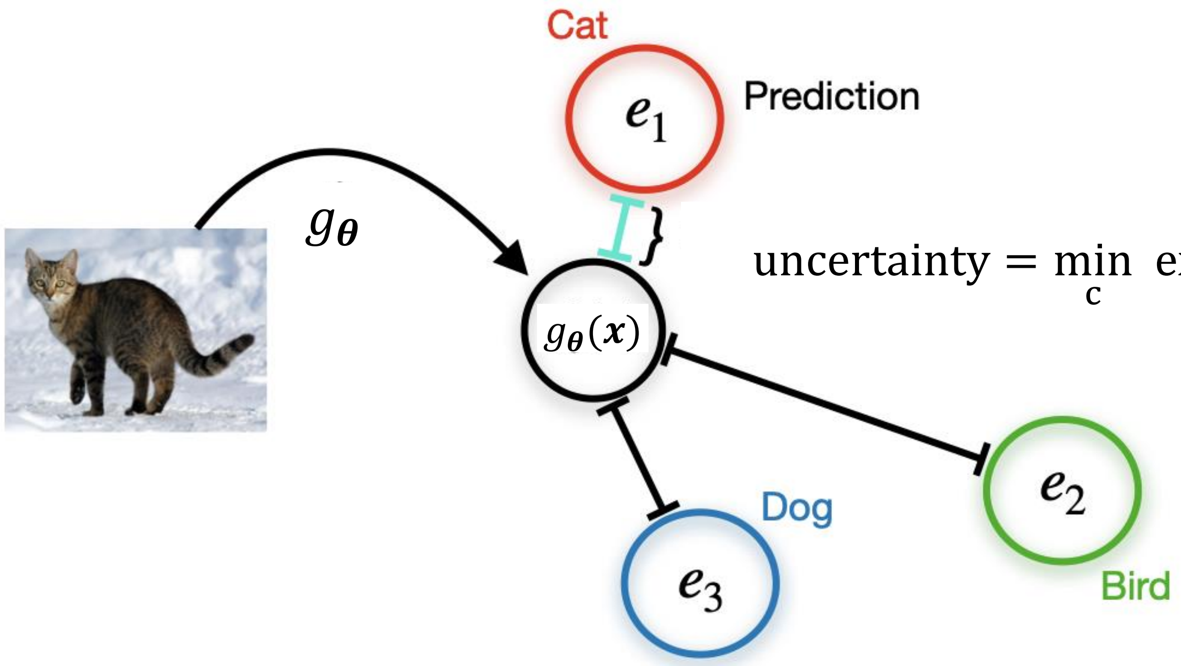
Google Research

Skoltech

1. Van Amersfoort, J., et al. "Uncertainty estimation using a single deep deterministic neural network." *ICML*, 2020.
2. Podolskiy, A., et al. "Revisiting mahalanobis distance for transformer-based out-of-domain detection." *AAAI*. 2021.
3. Liu et al. "Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness." *NeurIPS.* 2020

Uncertainty for Deterministic Networks (DUQ)

Common idea — **utilize representations of the training data.**



$$\text{uncertainty} = \min_c \exp\left[-\frac{\|g_\theta(x) - e_c\|_2^2}{2\sigma^2}\right]$$

A DUQ architecture. The input is mapped to the feature space, where it is assigned to the closest centroid. The distance to this centroid is uncertainty.

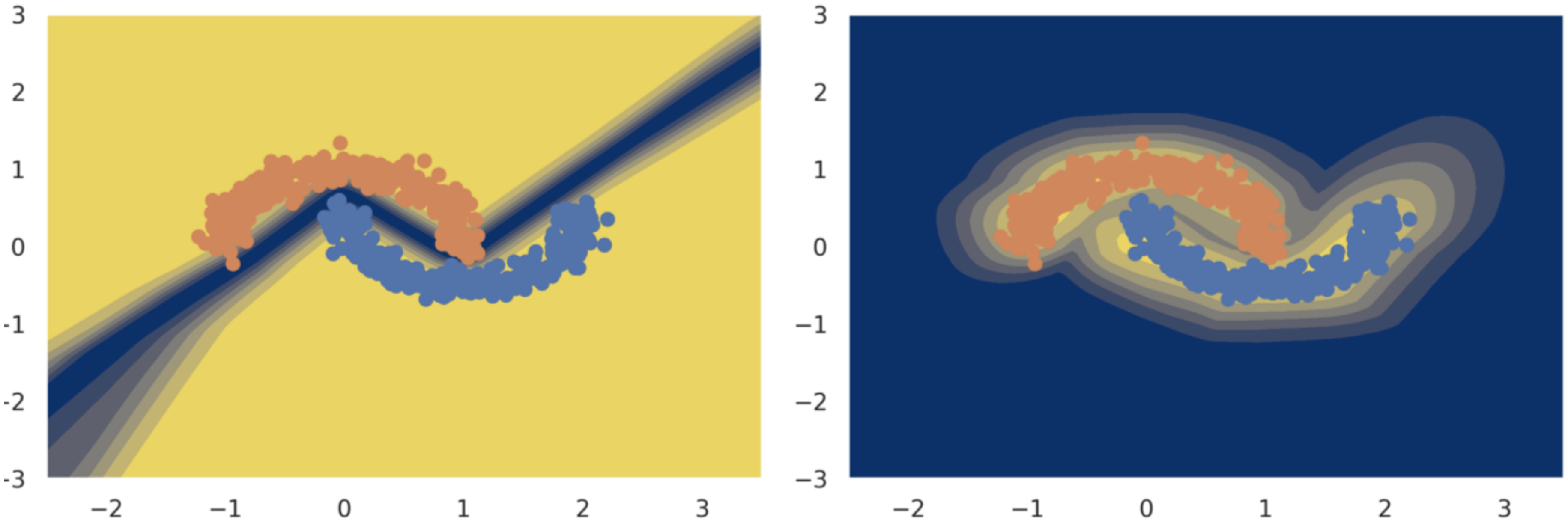
Source: Van Amersfoort, Joost, et al. "Uncertainty estimation using a single deep deterministic neural network." *ICML*, 2020.

Toy example for DUQ

High uncertainty

Low uncertainty

Training data



Deep ensembles | DUQ model

Skoltech

Source: Van Amersfoort, Joost, et al. "Uncertainty estimation using a single deep deterministic neural network." *ICML*, 2020.

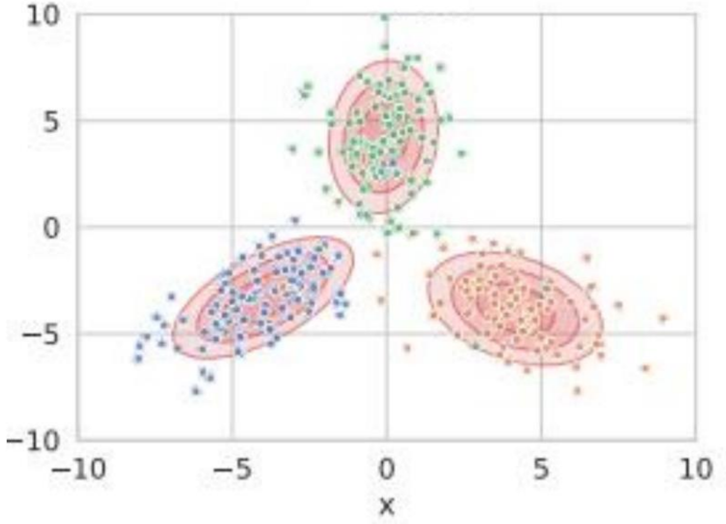
Density-based UE Methods: Deep Deterministic Uncertainty (DDU)

Fit a Gaussian Mixture Model (GMM) on the training data for $p\big(h(\boldsymbol{x})\big)$, where $h(\boldsymbol{x})$ – hidden representation of instance x.

$$\overline{U}_E^{DDU}(\boldsymbol{x}) = \sum_{c \in C} p(h(\boldsymbol{x})|y = c)p(y = c)$$

$$p(h(\boldsymbol{x})|y = c) \sim \mathcal{N}(h(\boldsymbol{x})|\mu_c, \Sigma_c)$$

$$p(y = c) = \frac{\sum_{(\boldsymbol{x}_i, y_i) \in D} \mathbf{1}[y_i = c]}{|D|}$$

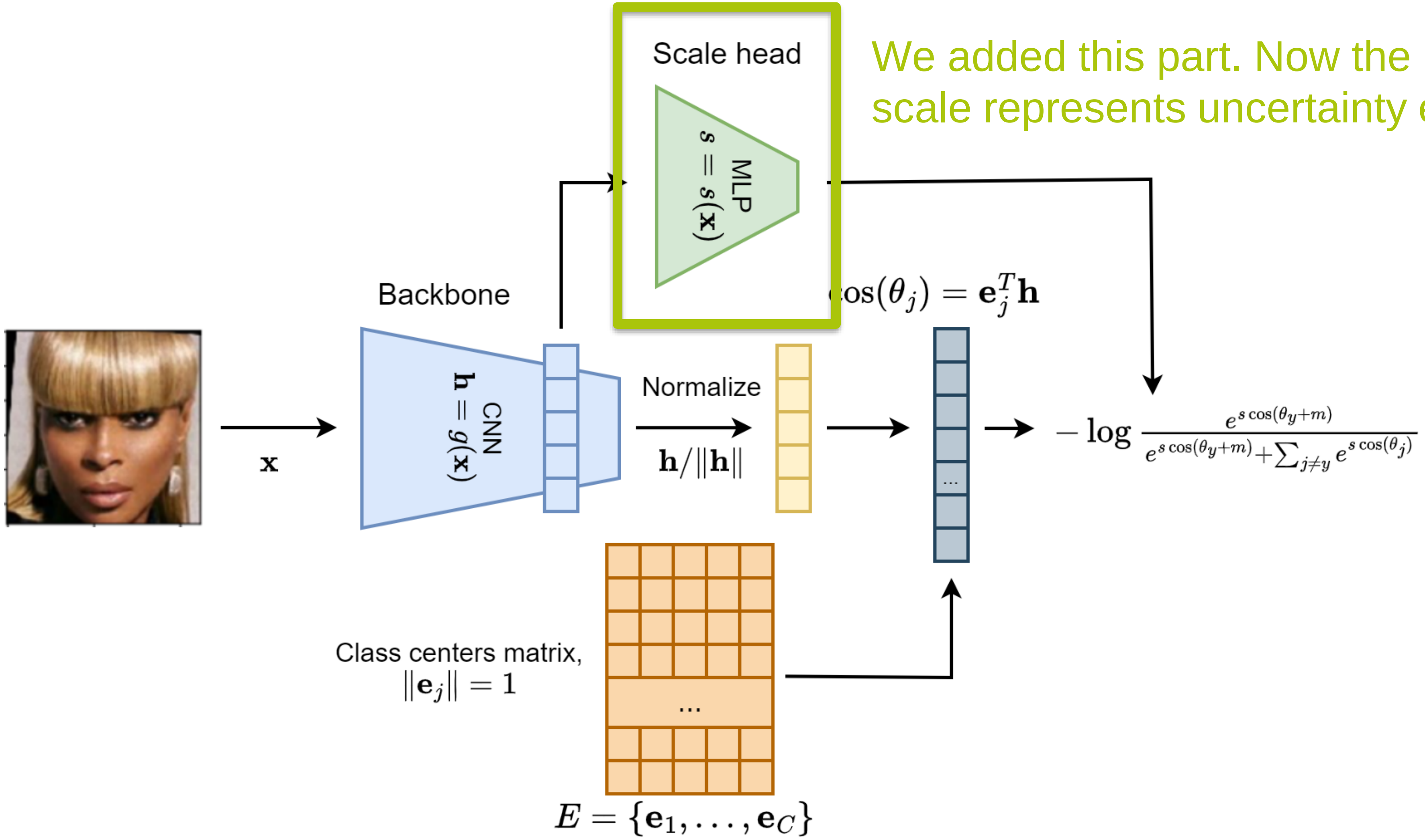


GMM with 3 components fitted to a dataset with 3 different classes

Skoltech

Mukhoti, J., et al. "Deep Deterministic Uncertainty: A New Simple Baseline." *CVPR*. 2023.

Scale tuning as an alternative



We added this part. Now the predicted scale represents uncertainty efficiently

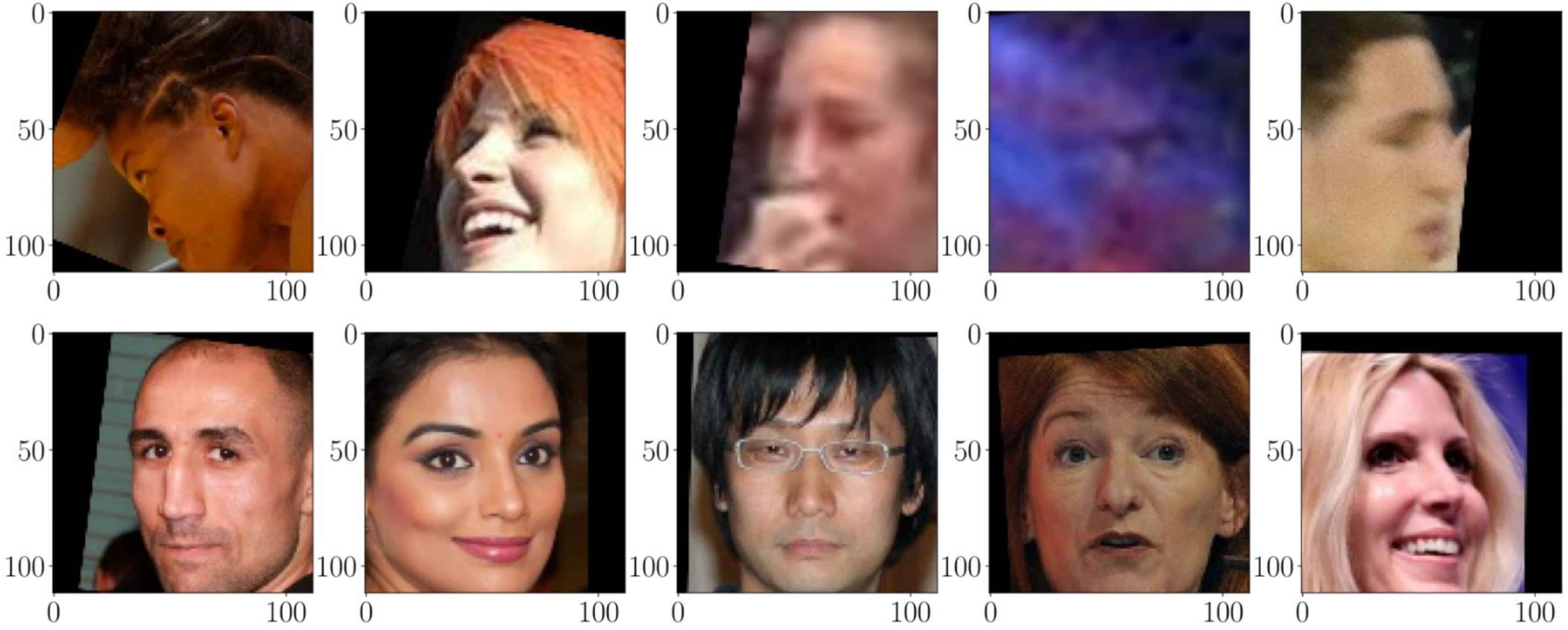
Skoltech

Source: Kail, Roman, et al. "ScaleFace: Uncertainty-aware deep metric learning." *IEEE DSAA*. IEEE, 2023.

ScaleFace judgements correlate with human for the Face Identification problem

From top10% uncertain w.r.t. ScaleFace

From top10% certain w.r.t. ScaleFace



Skoltech

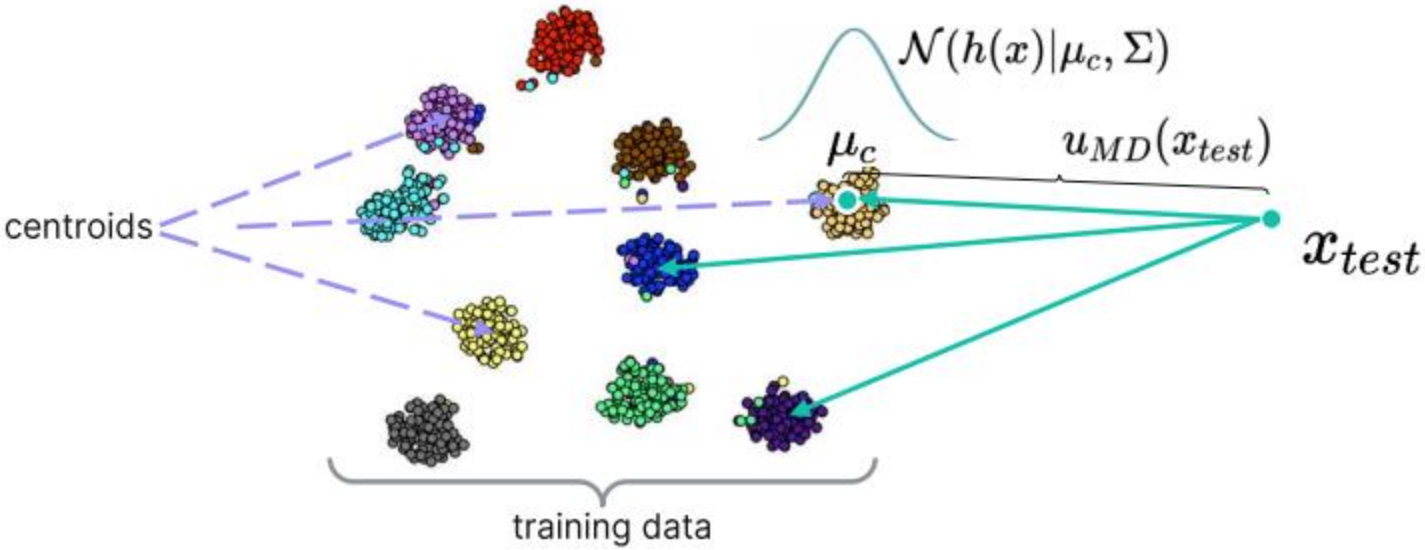
Source: Kail, Roman, et al. "ScaleFace: Uncertainty-aware deep metric learning." *IEEE DSAA*. IEEE, 2023.

Introducing Mahalanobis Distance

Mahalanobis distance is a generalization of the Euclidian distance. It takes into account the spreading of instances in the training set along various directions in a feature space:

$$u_{MD}(\boldsymbol{x}_i) = \min_{c \in C} (\boldsymbol{h}_i - \boldsymbol{\mu}_c)^T \Sigma^{-1} (\boldsymbol{h}_i - \boldsymbol{\mu}_c),$$

where $\boldsymbol{h}_i$ is a hidden representation of a i-th instance, $\boldsymbol{\mu}_c$ is a centroid of a class c, and Σ is a covariance matrix for hidden representations of training instances.



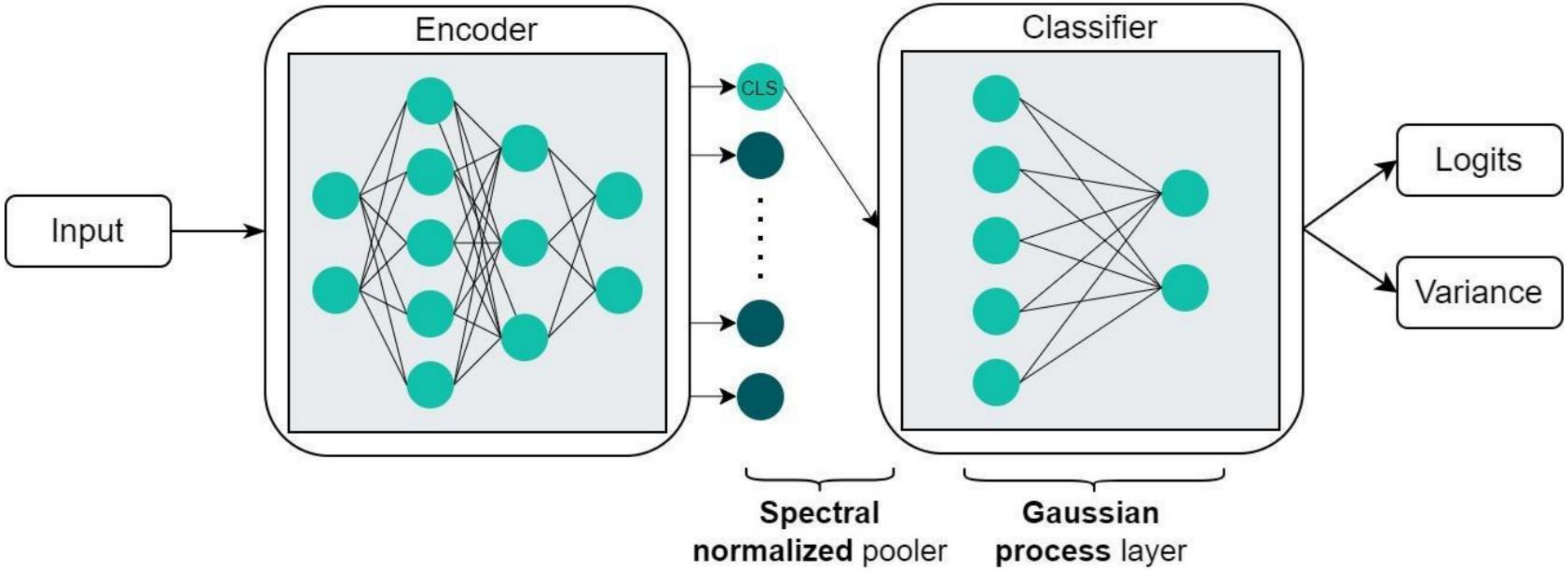
Podolskiy et al. Revisiting Mahalanobis Distance for Transformer-Based Out-of-Domain Detection. AAAI 2021
Lee et al. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks. NeurIPS 2018

Skoltech

Spectral-normalized Neural Gaussian Process

A. Use spectral normalization (SN) in the weight matrix of the one-but-last classification layer to preserve distance for hidden representations.

B. Replace the classic dense output layer of a network with a layer that implements a Gaussian process (GP) with an RBF kernel.



Liu et al. Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness, NeurIPS. 2020

A: Spectral normalization

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

$$W_l = \begin{cases} c\dfrac{W_l}{\hat{\lambda}}, & if\ c < \hat{\lambda} \\ W_l, & otherwise \end{cases}$$

We want bi-Lipschitz property for embeddings

To ensure this, we limit the norm of weights matrix, $\hat{\lambda} \approx |W_l|_2$

B: Random Fourier Features GP

- Squared exponential kernel
- Random Fourier Features (linear regression in a different feature space)
- Laplace approximation

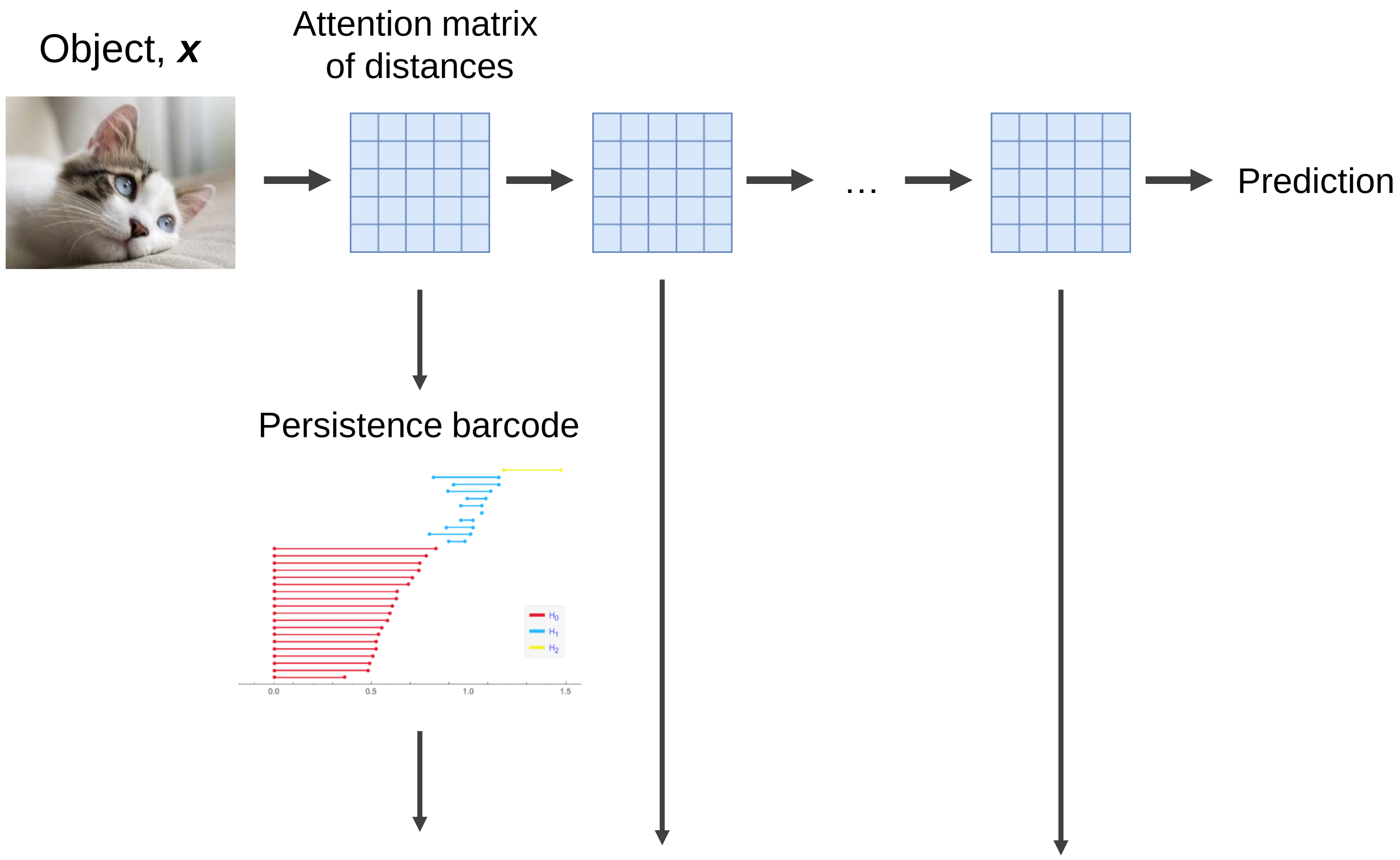
Skoltech

Liu et al. Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness, NeurIPS. 2020

Topological data analysis (TDA) representations

Existing models are either computationally demanding or look only at the last layer

- TDA helps to extract information about intermediate layers by examining barcodes of attention matrices $[\boldsymbol{h}_1, \boldsymbol{h}_2, \dots, \boldsymbol{h}_L]$ and cross barcodes $\boldsymbol{h}_{ij}$ via **persistent homologies**
- Now we have a representation $\boldsymbol{h}$ that takes into account all layers



$$\boldsymbol{h} = [\boldsymbol{h}_1, \quad \boldsymbol{h}_2, \dots, \quad \boldsymbol{h}_L] + [\boldsymbol{h}_{1,2}, \ \boldsymbol{h}_{1,3}, \dots, \ \boldsymbol{h}_{(L-1),L}]$$

Source: https://community.wolfram.com/groups/-/m/t/2029121

Usage of topological data analysis (TDA) for uncertainty estimation

Using the TDA-based representation $\boldsymbol{h}$ we recover uncertainty better than other methods (1, 2, 3)

We need both common TDA features and features from cross barcodes

Method type	Method name	En-CoLA	Ita-CoLA
Basic Methods	Softmax Response	0.068	0.085
	MC Dropout [1]	0.071	0.084
	Mahalanobis estimator [2]	0.083	0.091
	Embedding estimator [3]	0.075	0.090
Our methods	Topological estimator without cross-barcodes	<u>0.087</u>	<u>0.092</u>
	Topological estimator with cross-barcodes	**0.098**	**0.099**
Oracle Upper Bound		0.124	0.121

Area under rejection curve metric. We want to maximize it. The best value for each dataset (En-CoLA, Ita-CoLA) is highlighted with **bold font**, the second best values – with an <u>underscore</u>.

Skoltech

Source: Barannikov, Serguei, et al. "Manifold Topology Divergence: a Framework for Comparing Data Manifolds." *NeurIPS, 2021.*
Kostenok, Elizaveta, Daniil Cherniavskii, and Alexey Zaytsev. "Uncertainty Estimation of Transformers' Predictions via Topological Analysis of the Attention Matrices." *arXiv preprint,* 2023.

Semantic uncertainty

- Language models exhibit uncertainty at the token level, where different sentences can convey the same meaning.
- Our primal focus lies on understanding the intended meaning rather than the wording used to convey that meaning.
- Solution – semantic uncertainty

1. **Generation:** Sample M sequences from the predictive distribution of a large language model given a context $\boldsymbol{x}$.
2. **Clustering:** Cluster the sequences which mean the same thing using bidirectional entailment algorithm.
3. **Entropy estimation:** Approximate semantic entropy by summing probabilities that share a meaning and compute resulting entropy.

(a) Scenario 1: No semantic equivalence

Answer $\mathbf{s}$	Likelihood $p(\mathbf{s} \mid x)$	Semantic likelihood $\sum_{\mathbf{s}\in c} p(\mathbf{s} \mid x)$
Paris	0.5	0.5
Rome	0.4	0.4
London	0.1	0.1
Entropy	0.94	0.94

(b) Scenario 2: Some semantic equivalence

Answer $\mathbf{s}$	Likelihood $p(\mathbf{s} \mid x)$	Semantic likelihood $\sum_{\mathbf{s}\in c} p(\mathbf{s} \mid x)$
Paris	0.5	0.9
It's Paris	0.4	
London	0.1	0.1
Entropy	0.94	0.33

Kuhn, L., Gal, Y., & Farquhar, S. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *ICLR*. 2023

Open problems in uncertainty estimation

1. This area lacks theoretical explanations of observed phenomena.

2.Compared to other methods such as Gaussian process regression, uncertainties for neural networks are inferior.

3. The most effective techniques necessitate the use of ensembles, which leads to computational inefficiency.

4. Furthermore, these methods demonstrate suboptimal performance when applied to Large Language Models.

A musician or a face?

Skoltech