Uncertainty Estimation & Calibration

Akash Dhamasia, AI Software Solutions Engineer

Agenda

- ☐ Introduction & Motivation

- ☐ Uncertainty Estimation
    - ☐ Types of uncertainty estimation
    - ☐ Methods for uncertainty estimation

- ☐ Uncertainty Calibration
    - ☐ Techniques for uncertainty calibration
    - ☐ Methods for uncertainty calibration

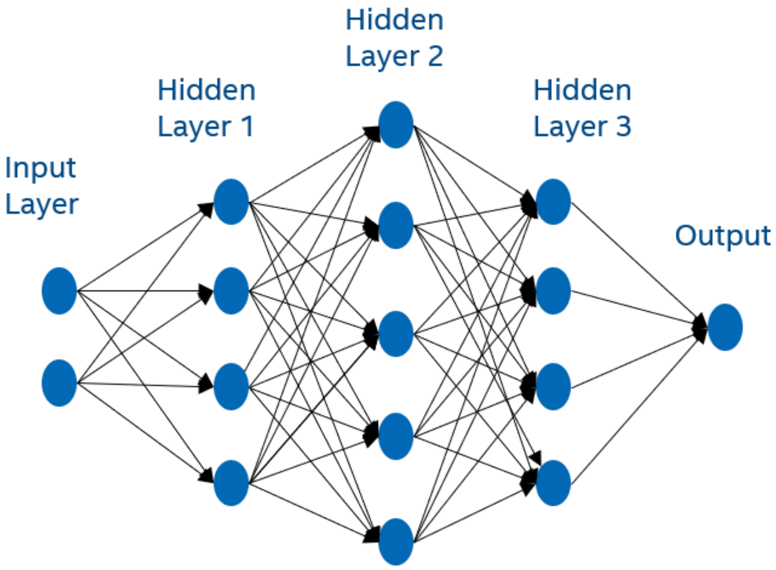
- ☐ Work @Intel on Uncertainty Estimation & Calibration

- ☐ Conclusion

- ☐ Demo

Introduction:
Deep Neural Architecture(DNN):

- A DNN consists of many neurons interconnected with each other to form a network similar to the structure of a Brain.
- A DNN can be divided into multiple layers each containing some fix sets of neurons as shown in Figure 2.1.
- There are multiple types of DNNs that are in use today, most popular being Convolution Neural Networks(CNNs), Recurrent Neural Networks (RNNs), Graph Neural Networks (GNN) and Transformer networks.



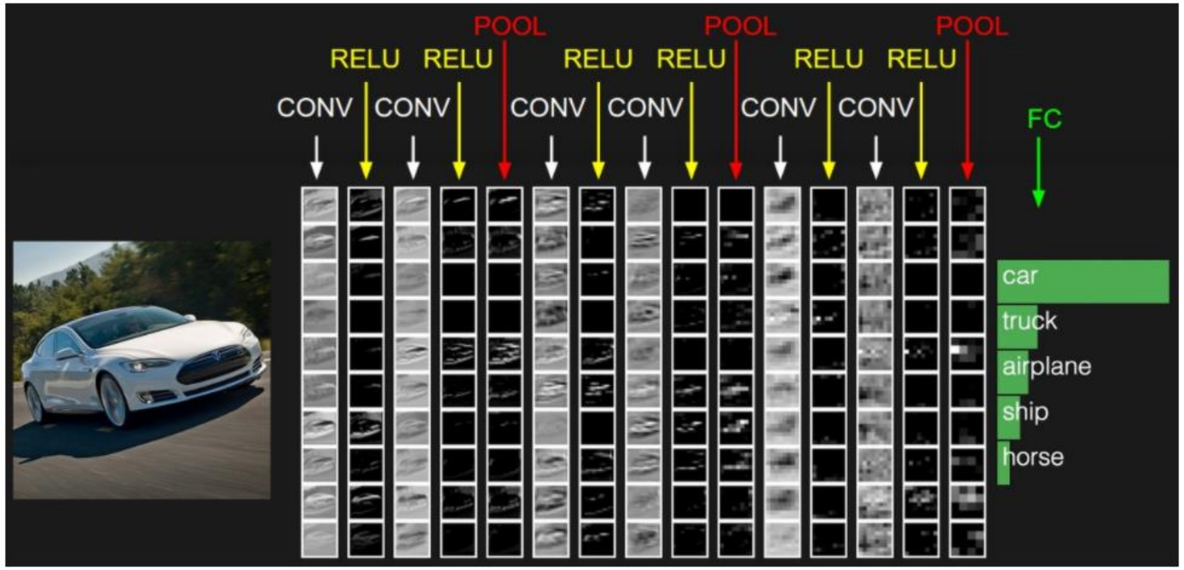


Image source: cs231n.stanford.edu

Introduction:
Safety Critical Applications

- Use of DNNs in safety-critical domains such as Autonomous Vehicles(AVs), Industrial Robotics and HealthCare are increasing, their safety and reliability is also becoming critical.

Image source: https://www.newworldai.com/deep-learning-self-driving-cars-2018-version-lecture-1/

Image source: https://link.springer.com/article/10.1007/s43154-020-00006-5

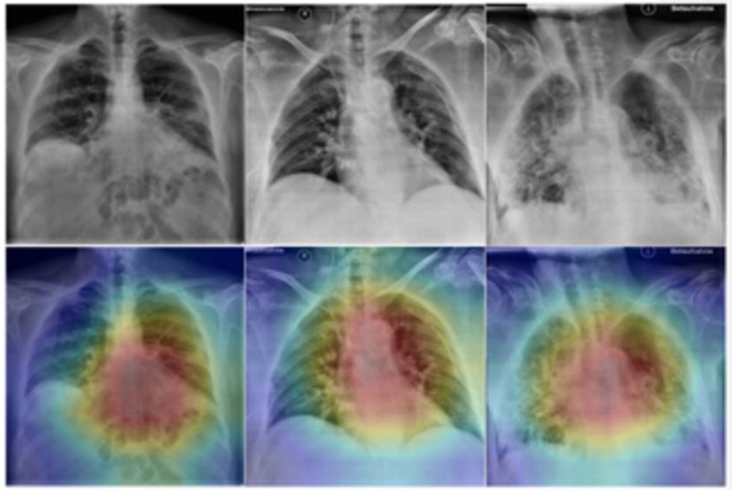


Image source: https://medium.com/@masteradd95/introduction-to-explainable-artificial-intelligence-in-medical-imaging-ad6bd919dd9f

- Safety critical applications require fast, scalable and calibrated uncertainty estimation

Introduction:
Motivation

A big part of reasons for these accidents is not acting when the DNN predictions are uncertain

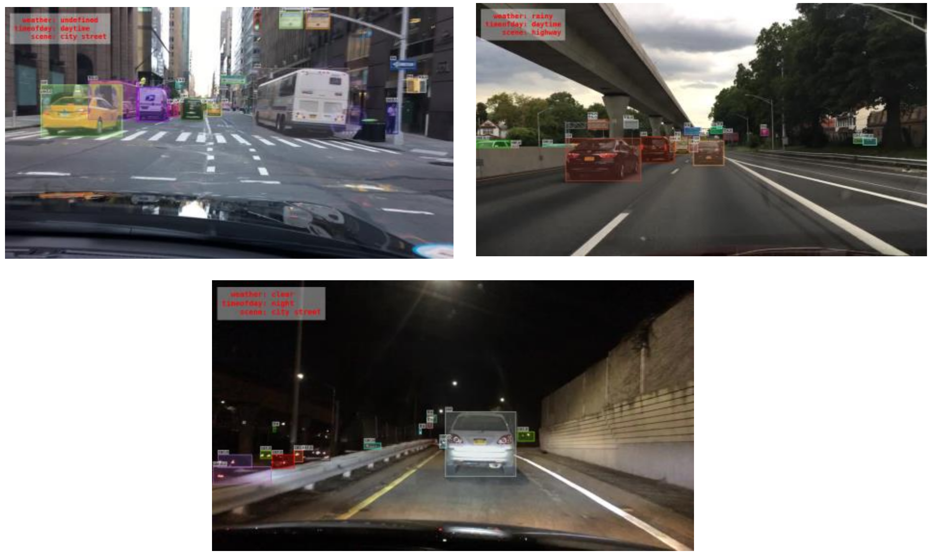
- On 23 March 2018 Tesla's Autopilot was involved in another deadly car crash.
- The automaker says its semi-autonomous system was engaged when a Model X SUV hit a freeway barrier last week in California, killing the driver.
- https://www.wired.com/story/tesla-autopilot-self-driving-crash-california/

Image source: https://www.ndtv.com/world-news/tesla-says-autopilot-was-engaged-during-fatal-crash-that-killed-apple-engineer-1831238

- On 14 Feb 2016, Google self-driving car hits a bus
- https://www.bbc.com/news/technology-35692845

Introduction:
Difference in Training dataset and Real Testing

Training Dataset



Source: https://www.bdd100k.com/

Real Scenario: Out of Distribution (O.O.D) data

Source:
https://www.vox.com/2016/4/21/11447838/self-driving-cars-challenges-obstacles
https://m.futurecar.com/1262/Autonomous-Cars-are-Having-Trouble-With-Animals
https://www.extremetech.com/extreme/189486-how-googles-self-driving-cars-detect-and-avoid-obstacles

Independent and Identically Distributed (I.I.D.) : $P_{train}(x,y) = P_{test}(x,y)$

Out of Distribution (O.O.D): $P_{train}(x,y) \neq P_{test}(x,y)$

Introduction:
Limitations of DNN networks

1. Adversarial Attacks
2. Out of Distribution data like unseen objects, covariance shift & label shift
3. Unseen Pose

Model fails to recognize out-of-distribution images of objects in unusual poses

[Alcorn et al., 2019]: Neural Networks Are Easily Fooled by Strange Poses of Familiar Objects
https://arxiv.org/pdf/1811.11553.pdf

Uncertainty Estimation:
In Classification & Regression network

Uncertainty Estimation:

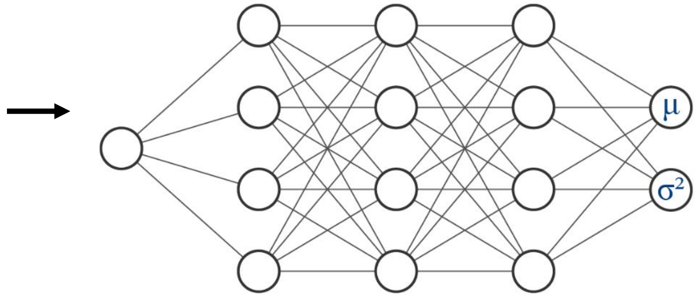
Returns a distribution of predictions rather than a single prediction.

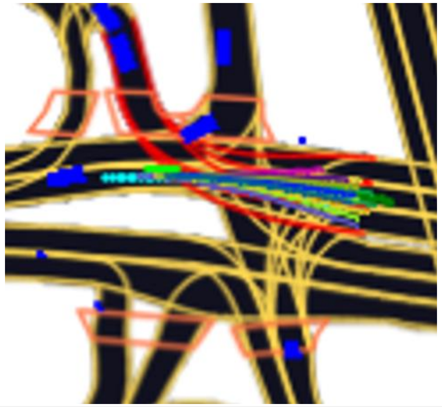
Classification Network:

With confidence score, return the output label that tells how certain the model is for a prediction. Simpler way to calculate predictive entropy of the combined outputs from multiple ensembles.

Regression Network:

Returns the mean and variance in the output.

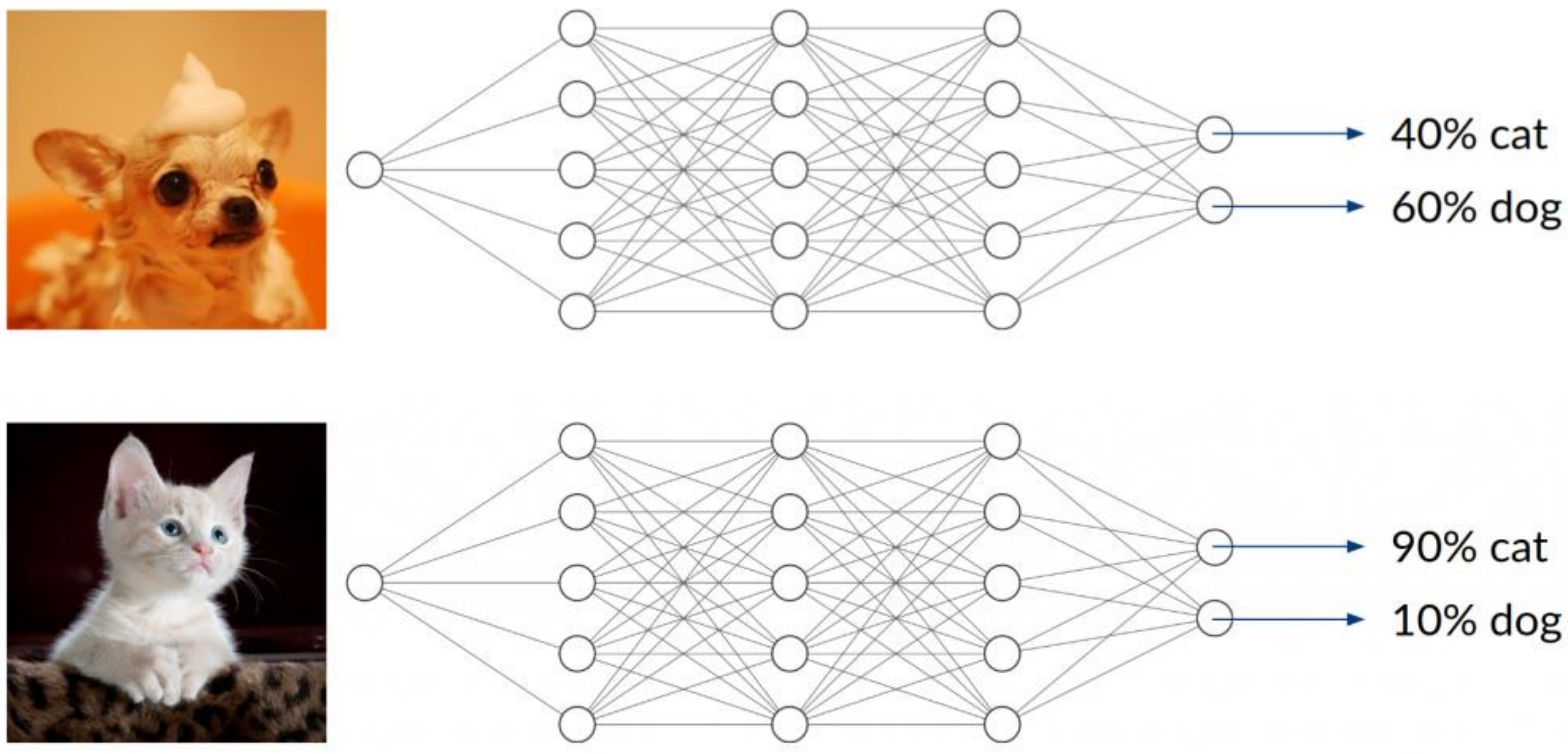




10 Monte-Carlo samples correspond to 10 probable trajectories

🟦 : History (10 frames)
🟩 : Prediction: Ground Truth (30 frames)
🟥 : Prediction: Mean Predicted Trajectory(30 frames)

Uncertainty Estimation:
In Classification network

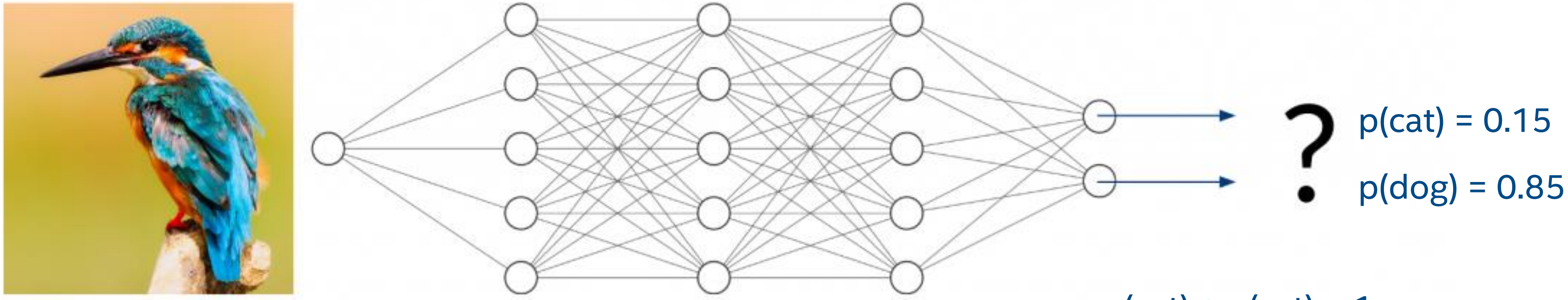


When the classification network is sure about what it sees, it assigns a high probability to the class

Source: https://www.inovex.de/de/blog/uncertainty-quantification-deep-learning/

Uncertainty Estimation:
In Classification network

Do not mistake output probability for model certainty



p(cat) + p(cat) = 1

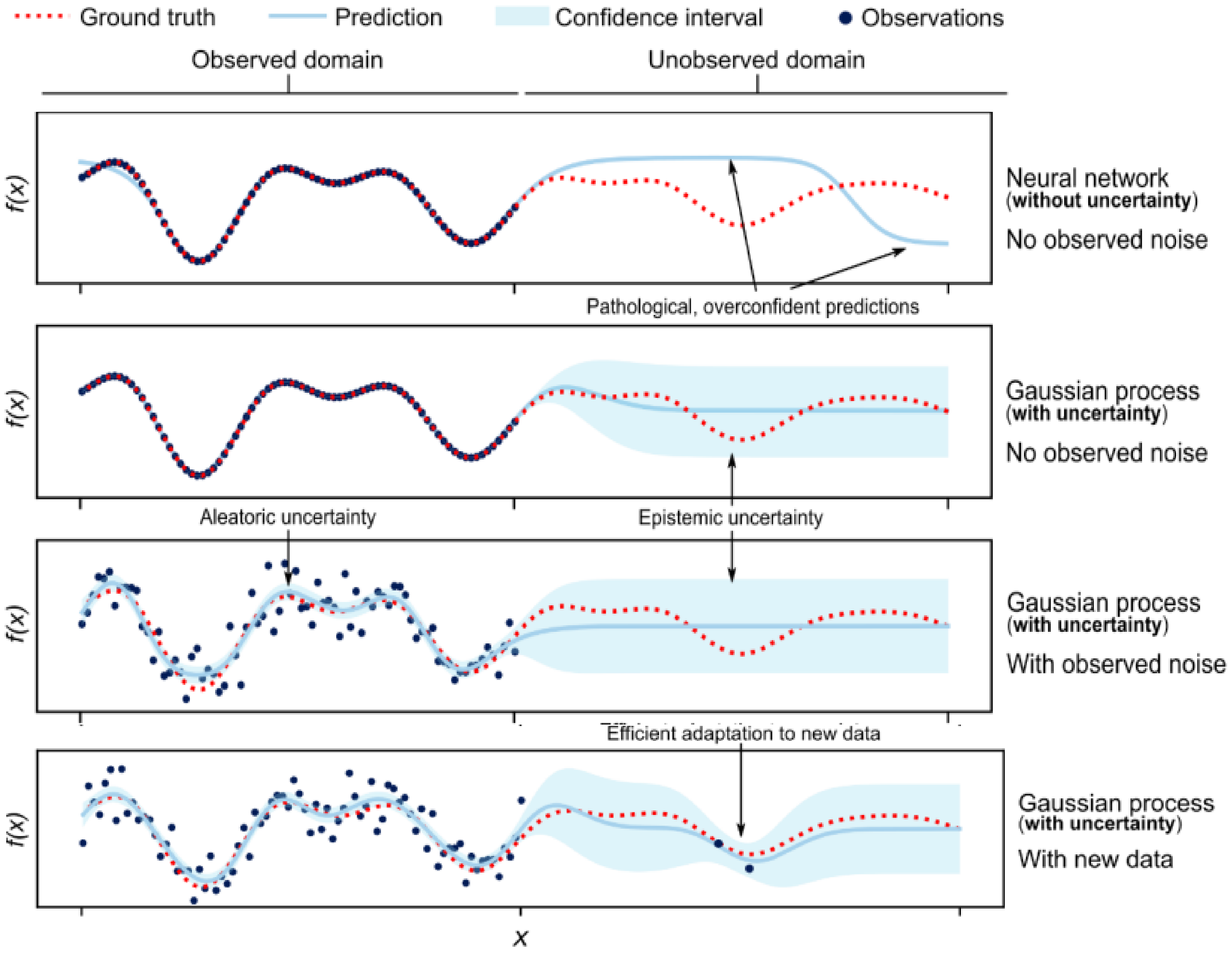
The output probabilities will be unreliable if the input is very different than training

Source: https://www.inovex.de/de/blog/uncertainty-quantification-deep-learning/

Uncertainty Estimation:
Types of uncertainty

Predictive uncertainty

- Aleatoric uncertainty
  - Data uncertainty
- Epistemic uncertainty
  - Model uncertainty



Source: https://www.biorxiv.org/content/10.1101/2020.08.11.247072v1.full.pdf

Uncertainty Estimation:
Aleatoric vs Epistemic Uncertainty

Aleatoric Uncertainty
(Data Uncertainty)

1. Describes the confidence in the input data
2. High when the input data is noisy
3. Cannot be reduced by adding more data

Epistemic Uncertainty
(Model Uncertainty)

1. Describes the confidence in the output prediction
2. High when missing training data
3. Can be reduced by adding more data

Credits: http://introtodeeplearning.com

Uncertainty Estimation:
Applications of uncertainty

Uncertainty

Safety

Better decision making: Calculating the risk vs reward associated with prediction (worst case vs average case)

Active Learning

The ability to “learn” from the data, usually by optimizing a model so it fits the data and its annotations

Reinforcement Learning

An uncertainty-aware model would let the agent know where it has a high chance of an undesired outcome

Trustworthy ML/DL

Knowing when to trust model’s predictions, especially under dataset shift

Bayesian Optimisation

Optimization of probability distribution to better estimate predictions.

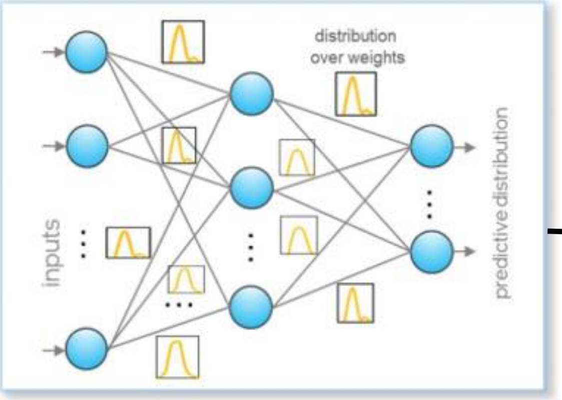
Uncertainty Estimation:
Methods to quantify uncertainty

Techniques for Uncertainty Quantization:

1. Variational Bayesian Inference
2. Monte Carlo Dropout
3. Ensembles
4. Monte Carlo Dropout Ensembles
5. Quantile Regression

Uncertainty Estimation:
Methods to quantify uncertainty

Technique:
1. Variational Bayesian Inference

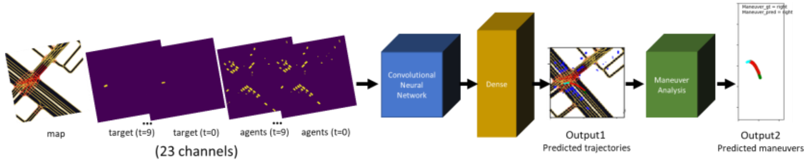


Monte Carlo Sampling

Captures uncertainty

- Bayesian neural networks differ from plain neural networks in that their weights are assigned a probability distribution instead of a single value or point estimate.
- These probability distributions describe the uncertainty in weights and can be used to estimate uncertainty in predictions.
- Training a Bayesian neural network via variational inference learns the parameters of these distributions instead of the weights directly.[3]

Trajectory Prediction network



Convert to Bayesian network

+

Monte Carlo Sampling

Captures uncertainty

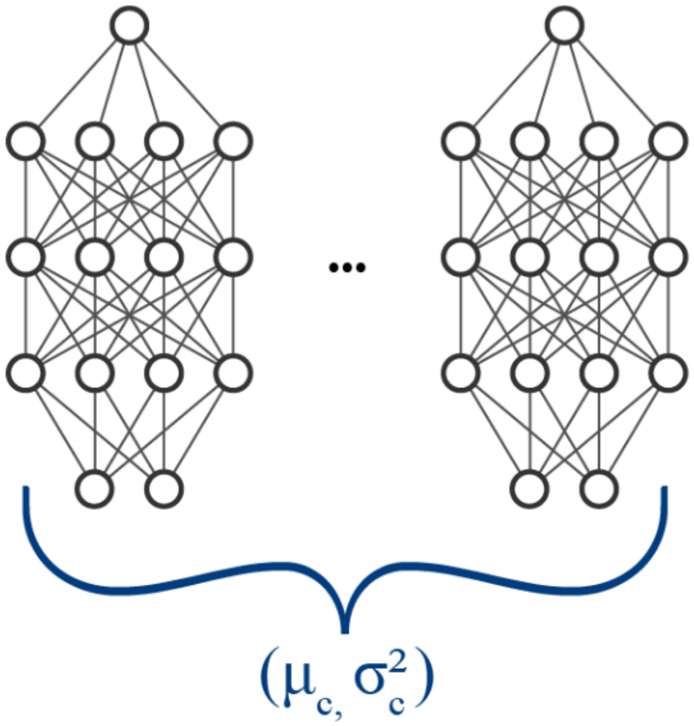
}Variance

[2]:

Uncertainty Estimation:
Methods to quantify uncertainty

Technique:

2. Ensemble Averaging

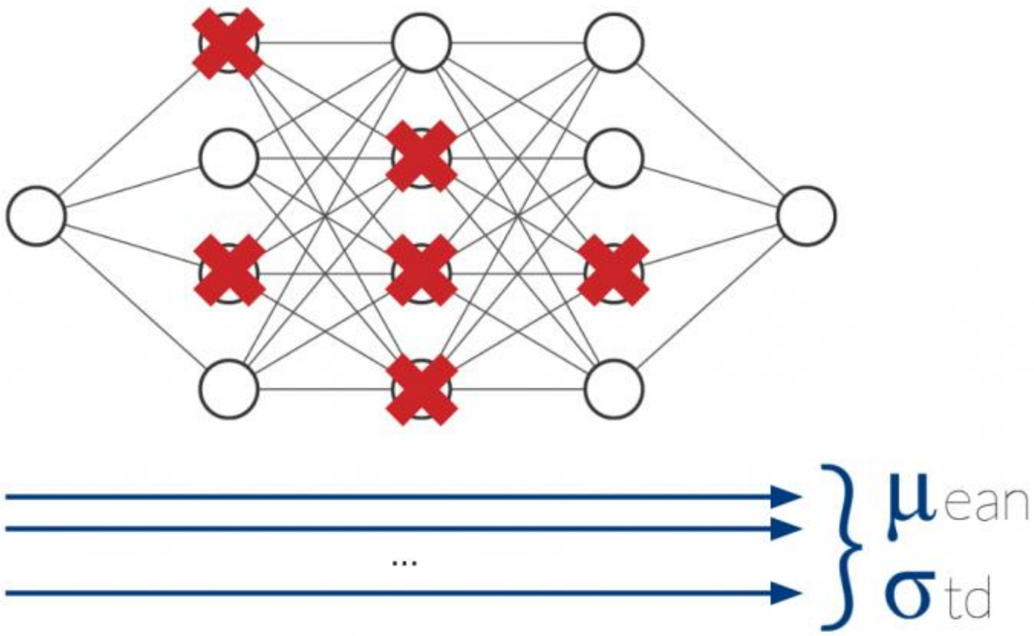


Ensemble Averaging

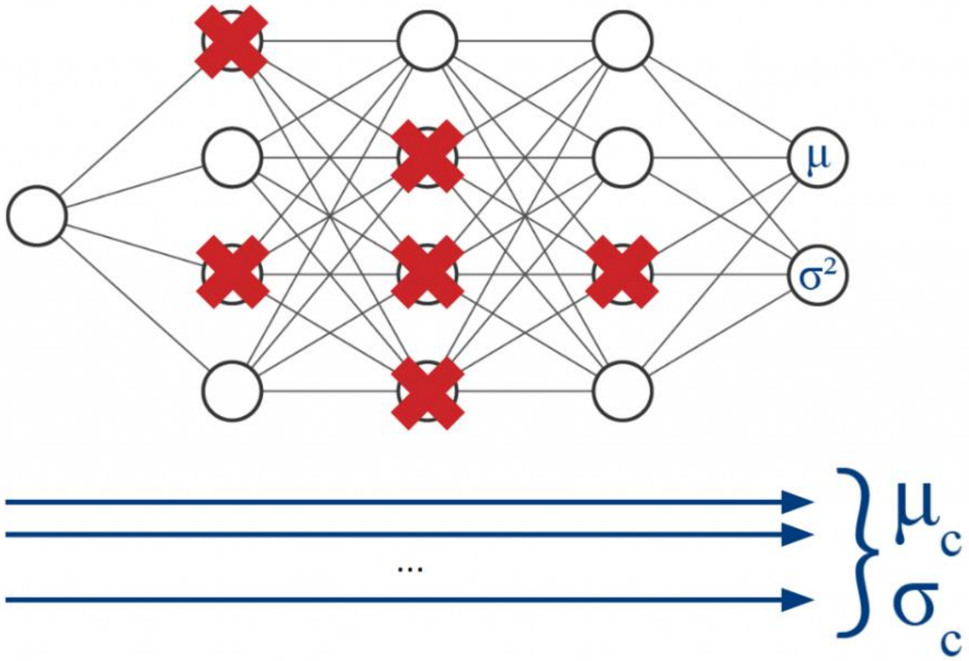
$$\begin{aligned} \hat{\mu}_c(x) &= \frac{1}{M}\sum_{i=1}^{M} \hat{\mu}_i(x), \\ \hat{\sigma}_c^2(x) &= \underbrace{\frac{1}{M}\sum_{i=1}^{M} \hat{\sigma}_i^2(x)}_{\text{aleatoric}} + \underbrace{\left[\frac{1}{M}\sum_{i=1}^{M} \hat{\mu}_i^2(x) - \hat{\mu}_c^2(x)\right]}_{\text{epistemic}} \end{aligned}$$

Source: https://www.inovex.de/de/blog/uncertainty-quantification-deep-learning/

Uncertainty Estimation:
Methods to quantify uncertainty



3. Monte-Carlo Dropout disables network nodes randomly during training and prediction time.



4. Dropout Ensembles

- Deep Ensembles generally produce the best results among the neural network-based approaches.
- Deep Ensembles, as well as Dropout Ensembles, provide the benefit of being able to separately determine aleatory and epistemic uncertainty.

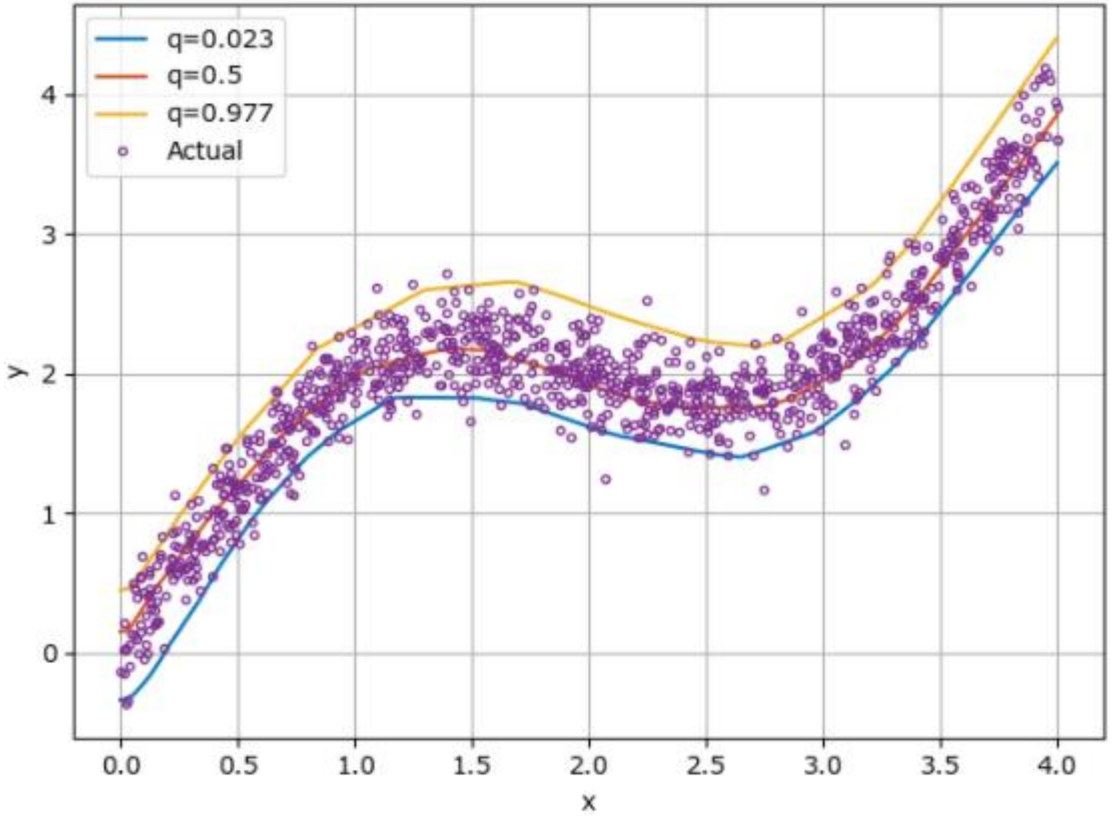
Source: https://www.inovex.de/de/blog/uncertainty-quantification-deep-learning/

Uncertainty Estimation:
Methods to quantify uncertainty

Techniques:
5. Quantile Regression

In this technique, we train our loss on multiple Quantiles. A quantile is the value below which a fraction of observations in a group falls. For example, a prediction for quantile 0.9 should over-predict 90% of the times. [4]

Variance could be the standard deviation between 2 quantile (0.001 & 0.999)

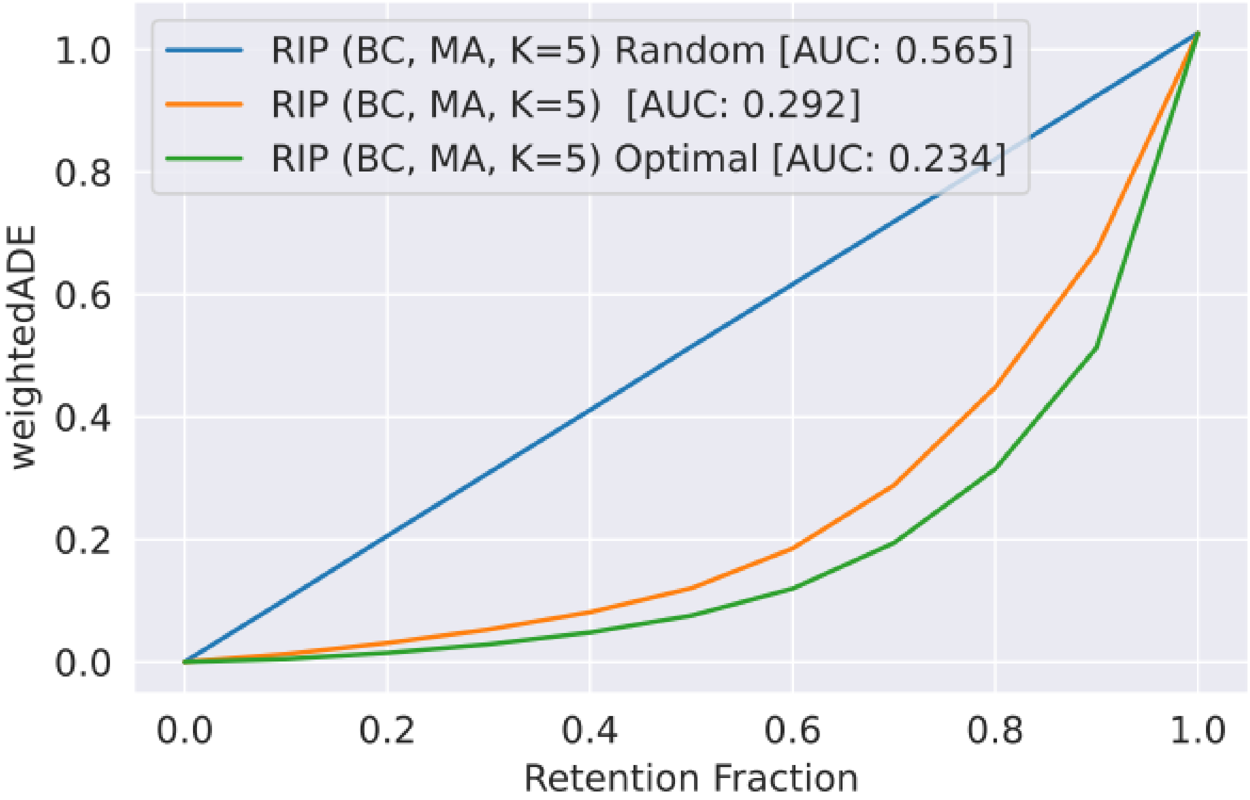


Source: https://medium.com/analytics-vidhya/quantile-regression-and-prediction-intervals-e4a6a33634b4

Uncertainty Estimation:
How do we measure the quality of uncertainty?

Measure: Certainty Score α Accuracy

Methods:
1. Retention graph
2. Brier Score
3. Expected Calibration Error (ECE)
4. Negative Log-Likelihood (NLL) (cross-entropy)



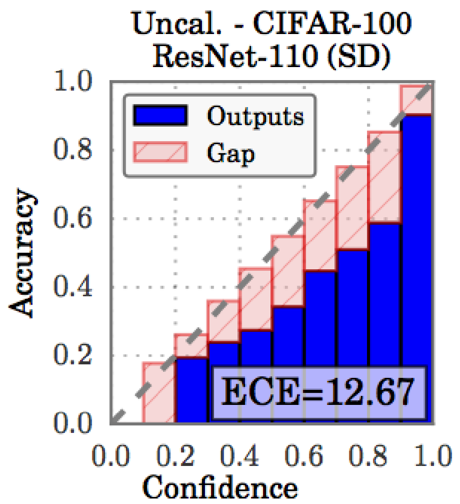
source: https://storage.yandexcloud.net/yandex-research/shifts/paper.pdf

Uncertainty Calibration:
Can You Trust Your Model's Uncertainty?

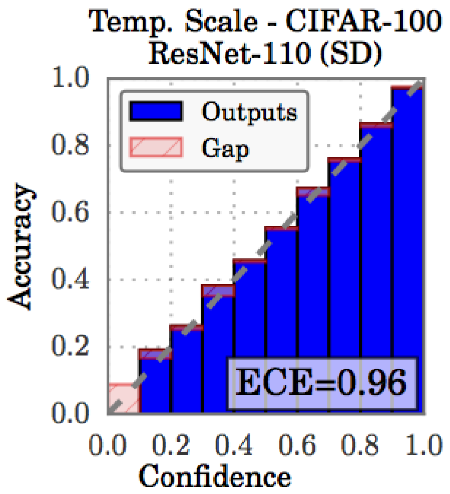
Methods to calibrate model's uncertainty:

1. Temperature Scaling
2. Isotonic Regression
3. AvUC (Accuracy versus Uncertainty Calibration) (From Intel)
4. Error Aligned Uncertainty Optimization (From Intel)

Temperature Scaling



An uncalibrated neural network, before temperature scaling. The reliability diagram indicates miscalibration.



Neural network after temperature scaling. The reliability diagram indicates a well-calibrated network

Source: https://github.com/gpleiss/temperature_scaling

AvUC :

		Uncertainty	
		certain	uncertain
Accuracy	accurate	**AC**	**AU**
	inaccurate	**IC**	**IU**

$$\mathrm{AvU} = \frac{n_{AC} + n_{IU}}{n_{AC} + n_{AU} + n_{IC} + n_{IU}}$$

$$\mathcal{L}_{\mathrm{AvUC}} := -\log\left(\frac{n_{AC} + n_{IU}}{n_{AC} + n_{IU} + n_{AU} + n_{IC}}\right) = \log\left(1 + \frac{n_{AU} + n_{IC}}{n_{AC} + n_{IU}}\right)$$

Source: https://papers.nips.cc/paper/2020/file/d3d9446802a44259755d38e6d163e820-Paper.pdf
ranganath.krishnan@intel.com

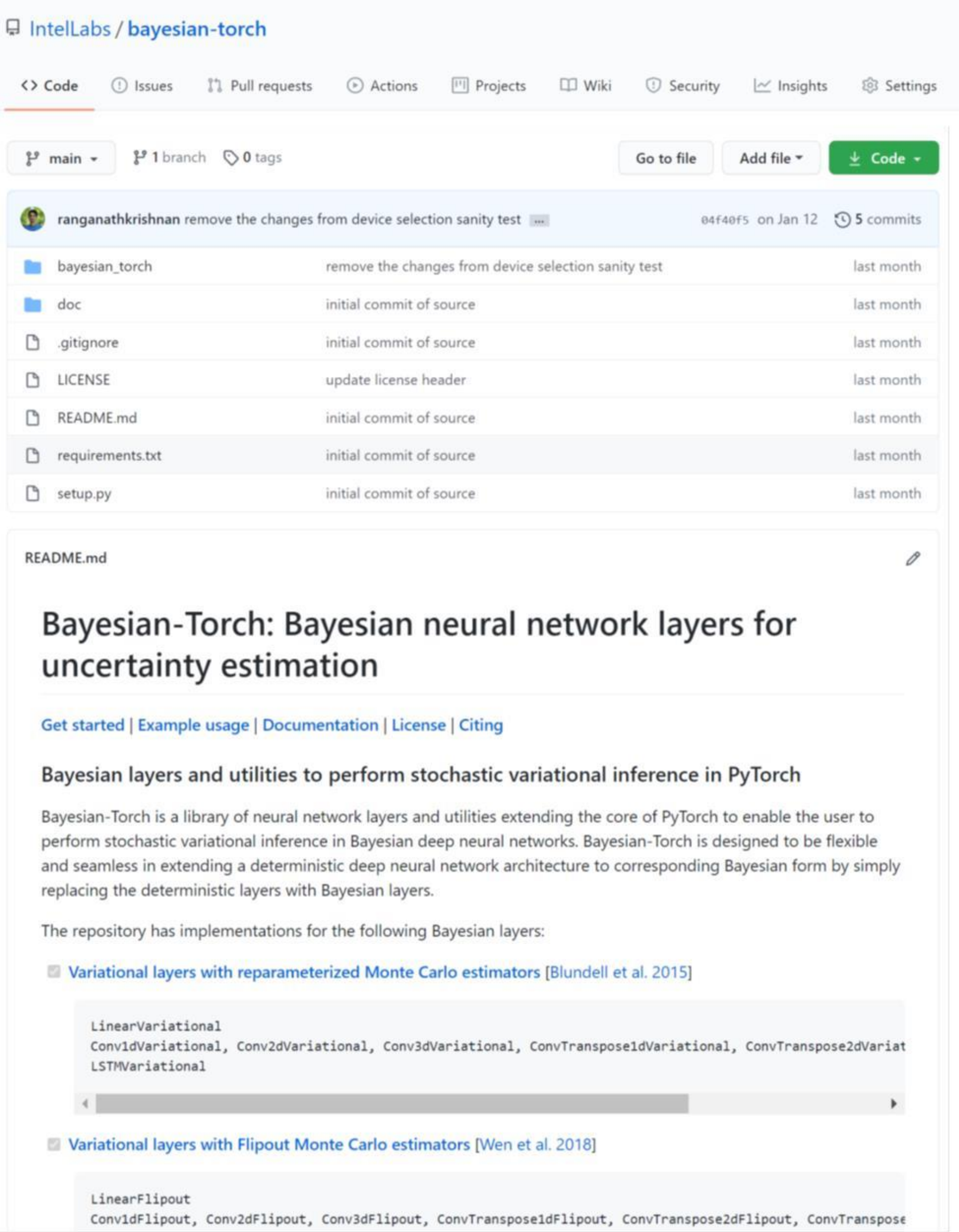
Work@Intel for Uncertainty Estimation and Calibration:

Bayesian torch:

•https://github.com/IntelLabs/bayesian-torch

- Open-source framework, extending the core of PyTorch to build Bayesian deep neural network models and perform scalable stochastic variational inference
- Bayesian convolutional, linear and LSTM layers
- Monte Carlo estimators
  - Reparameterization
  - Flipout (pseudo-independent perturbations within mini-batch)

Credits: ranganath.krishnan@intel.com



ranganathkrishnan remove the changes from device selection sanity test … 04f40f5 on Jan 12 5 commits

bayesian_torch	remove the changes from device selection sanity test	last month
doc	initial commit of source	last month
.gitignore	initial commit of source	last month
LICENSE	update license header	last month
README.md	initial commit of source	last month
requirements.txt	initial commit of source	last month
setup.py	initial commit of source	last month

README.md

Bayesian-Torch: Bayesian neural network layers for uncertainty estimation

Get started | Example usage | Documentation | License | Citing

Bayesian layers and utilities to perform stochastic variational inference in PyTorch

Bayesian-Torch is a library of neural network layers and utilities extending the core of PyTorch to enable the user to perform stochastic variational inference in Bayesian deep neural networks. Bayesian-Torch is designed to be flexible and seamless in extending a deterministic deep neural network architecture to corresponding Bayesian form by simply replacing the deterministic layers with Bayesian layers.

The repository has implementations for the following Bayesian layers:

☑ Variational layers with reparameterized Monte Carlo estimators [Blundell et al. 2015]

```
LinearVariational
Conv1dVariational, Conv2dVariational, Conv3dVariational, ConvTranspose1dVariational, ConvTranspose2dVariat
LSTMVariational
```

☑ Variational layers with Flipout Monte Carlo estimators [Wen et al. 2018]

```
LinearFlipout
Conv1dFlipout, Conv2dFlipout, Conv3dFlipout, ConvTranspose1dFlipout, ConvTranspose2dFlipout, ConvTranspose
```

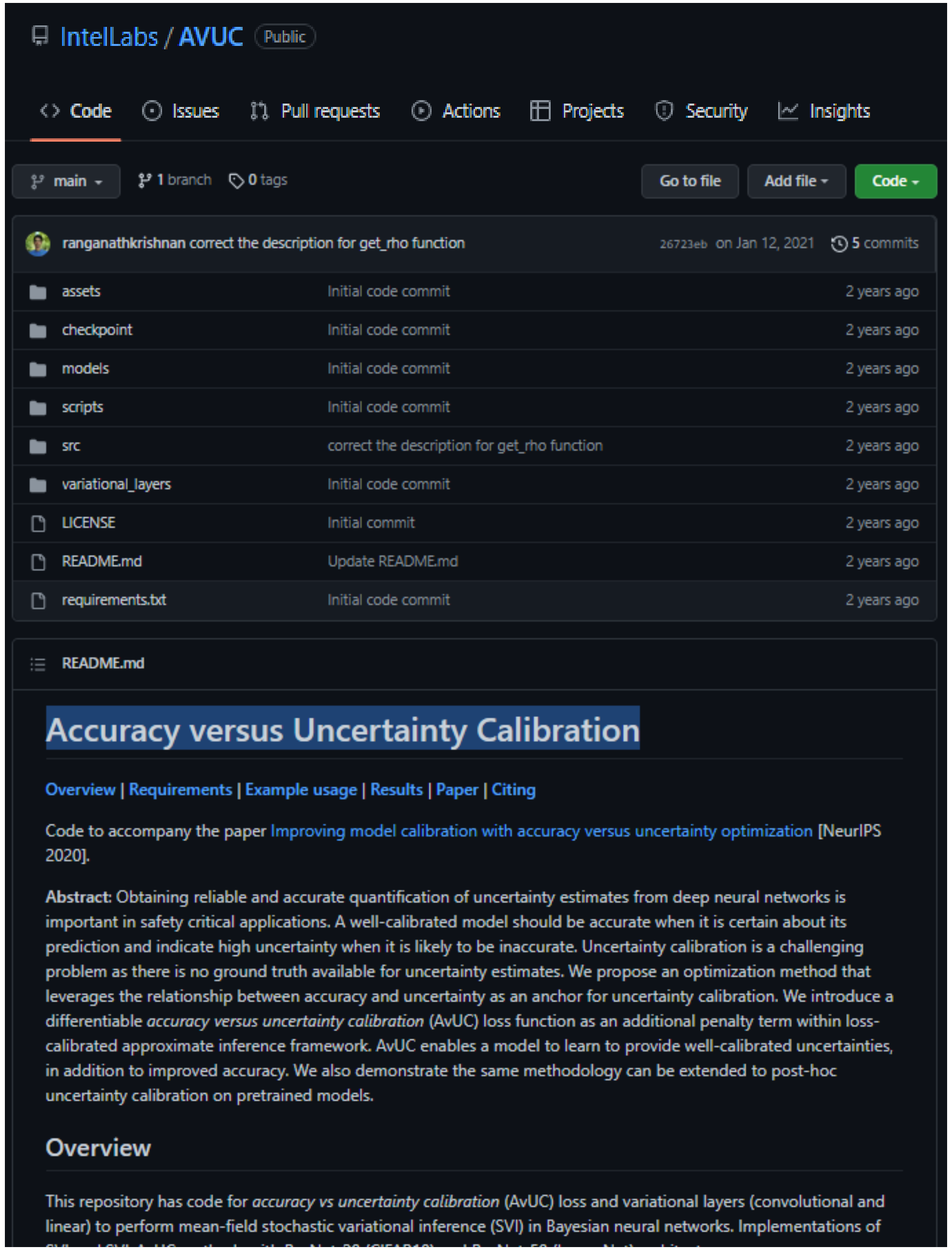
Work@Intel for Uncertainty Estimation and Calibration:

Accuracy versus Uncertainty Calibration (AvUC):
https://github.com/IntelLabs/AVUC

		Uncertainty	
		certain	uncertain
Accuracy	accurate	**AC**	**AU**
	inaccurate	**IC**	**IU**

$$\mathrm{AvU} = \frac{n_{AC} + n_{IU}}{n_{AC} + n_{AU} + n_{IC} + n_{IU}}$$

$$\mathcal{L}_{\mathrm{AvUC}} := -\log\left(\frac{n_{AC} + n_{IU}}{n_{AC} + n_{IU} + n_{AU} + n_{IC}}\right) = \log\left(1 + \frac{n_{AU} + n_{IC}}{n_{AC} + n_{IU}}\right)$$



ranganathkrishnan correct the description for get_rho function 26723eb on Jan 12, 2021 5 commits

assets	Initial code commit	2 years ago
checkpoint	Initial code commit	2 years ago
models	Initial code commit	2 years ago
scripts	Initial code commit	2 years ago
src	correct the description for get_rho function	2 years ago
variational_layers	Initial code commit	2 years ago
LICENSE	Initial commit	2 years ago
README.md	Update README.md	2 years ago
requirements.txt	Initial code commit	2 years ago

README.md

Accuracy versus Uncertainty Calibration

Overview | Requirements | Example usage | Results | Paper | Citing

Code to accompany the paper Improving model calibration with accuracy versus uncertainty optimization [NeurIPS 2020].

Abstract: Obtaining reliable and accurate quantification of uncertainty estimates from deep neural networks is important in safety critical applications. A well-calibrated model should be accurate when it is certain about its prediction and indicate high uncertainty when it is likely to be inaccurate. Uncertainty calibration is a challenging problem as there is no ground truth available for uncertainty estimates. We propose an optimization method that leverages the relationship between accuracy and uncertainty as an anchor for uncertainty calibration. We introduce a differentiable *accuracy versus uncertainty calibration* (AvUC) loss function as an additional penalty term within loss-calibrated approximate inference framework. AvUC enables a model to learn to provide well-calibrated uncertainties, in addition to improved accuracy. We also demonstrate the same methodology can be extended to post-hoc uncertainty calibration on pretrained models.

Overview

This repository has code for *accuracy vs uncertainty calibration* (AvUC) loss and variational layers (convolutional and linear) to perform mean-field stochastic variational inference (SVI) in Bayesian neural networks. Implementations of

Credits: https://papers.nips.cc/paper/2020/file/d3d9446802a44259755d38e6d163e820-Paper.pdf
ranganath.krishnan@intel.com

Work@Intel for Uncertainty Estimation and Calibration:

Select publications

1. Ranganath Krishnan, Omesh Tickoo “Improving model calibration with accuracy versus uncertainty optimization”. Advances in Neural Information Processing Systems (Vol. 33, pp. 18237-18248) (**NeurIPS 2020**) - **link**
2. Neslihan Kose Cihangir, Ranganath Krishnan, Akash Dhamasia, Omesh Tickoo, Michael Paulitsch. “Reliable Multimodal Trajectory Prediction via Error Aligned Uncertainty Optimization”. accepted for publication at SAIAD workshop(2022)
3. Bhatt, U., Zhang, Y., Antorán, J., Liao, Q.V., Sattigeri, P., Fogliato, R., Melançon, G.G., Krishnan, R., Stanley, J., Tickoo, O., Nachman, L. et al. “Uncertainty as a Form of Transparency: Measuring, Communicating, and Using Uncertainty”. Fourth AAAI/SCM Conference on Artificial Intelligence, Ethics and Society (**AIES 2021**) – **link**
4. Ranganath Krishnan, Mahesh Subedar, Omesh Tickoo “Specifying Weight Priors in Bayesian Deep Neural Networks with Empirical Bayes”. In Proceedings of the Thirty-fourth AAAI Conference on Artificial Intelligence. (**AAAI 2020**) - **link**
5. Mahesh Subedar, Ranganath Krishnan, Paulo Meyer, Omesh Tickoo, Jon Huang “Uncertainty-aware Audiovisual Activity Recognition using Deep Bayesian Variational Inference”. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 6300-6309, doi: 10.1109/ICCV.2019.00640) oral presentation (**ICCV 2019**) - **link**
6. Ranganath Krishnan, Alok Sinha, Nilesh Ahuja, Mahesh Subedar, Omesh Tickoo, Ravi Iyer “Mitigating Sampling Bias and Improving Robustness in Active Learning” Third workshop on Human-in-the-loop Learning (**ICML 2021**) - **link**
7. Mahesh Subedar, Ranganath Krishnan, Sidharth N Kashyap and Omesh Tickoo. “Quantization of Bayesian neural networks and its effect on quality of uncertainty”. Uncertainty & Robustness in Deep Learning workshop (**ICML 2021**) - **link**
8. Ranganath Krishnan, Mahesh Subedar, Omesh Tickoo, Angelos Filos, Yarin Gal “Improving MFVI in Bayesian Neural Networks with Empirical Bayes: a Study with Diabetic Retinopathy Diagnosis”. Fourth workshop on Bayesian Deep Learning (**NeurIPS 2019)** - **link**
9. Mahesh Subedar, Nilesh Ahuja, Ranganath Krishnan, Ibrahima Ndiour, Omesh Tickoo “Deep Probabilistic Models to Detect Data Poisoning Attacks”. Fourth workshop on Bayesian Deep Learning (**NeurIPS 2019**) - **link**
10. Ranganath Krishnan, Mahesh Subedar, Omesh Tickoo “Efficient Priors for Scalable Variational Inference in Bayesian Deep Neural Networks”. IEEE/CVF International Conference on Computer Vision Workshops (**ICCV 2019**) – **link**
11. Mahesh Subedar, Ranganath Krishnan, Paulo L. Meyer, Omesh Tickoo, Jon Huang “Uncertainty aware audiovisual activity recognition using deep Bayesian variational inference”. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops 2019 (pp. 103-108) (**CVPR 2019**)

Conclusion:

- We have learned about the importance of uncertainty estimation and why it is very important for safety critical applications ?

- While uncertainty estimates can be represented with a variety of methods, the chosen method should be one that is tested with the application.

- Uncertainty calibration becomes more important as input data-shift (noise) increases

- Intel offers various tools for uncertainty estimation and calibration
    - https://github.com/IntelLabs/bayesian-torch
    - https://github.com/IntelLabs/AVUC

Demo

Questions?