An introduction to Bayesian Neural Network and uncertainty quantification in Deep Learning

Inverted CERN School of Computing 2023

Jacopo Talpini

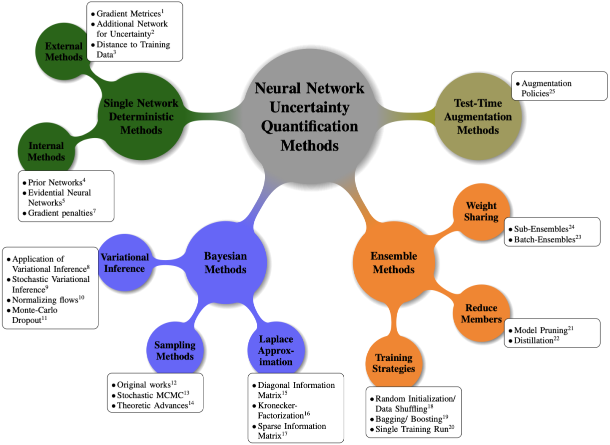
University of Milano-Bicocca

j.talpini@campus.unimib.it

Outline

Introduction

- Many advances in Deep Learning, deployed in real-life settings
- Safety-Critical domains requires reliable uncertainty estimates



[1]

[1] Jakob Gawlikowski et al. *A survey of uncertainty in deep neural networks*. 2021.

Plan

- Recap on Neural Network training from a probabilistic perspective
- Introduction to uncertainty
- Introduction to Bayesian Neural Network
- Introduction to Variational Inference
- Examples and other approaches to uncertainty quantification

Recap on Neural Network

- Given a dataset: $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ of input-output pairs
- We define a parametric function (aka a Neural Net) $\hat{y} \equiv f(\boldsymbol{x}; \boldsymbol{w})$ for describing $\mathcal{D}$
- **Problem**: how to chose $\boldsymbol{w}$ so that $\hat{y}_i(\boldsymbol{x}_i)$ is close to y_i for all input-output pairs of $\mathcal{D}$?

Neural Networks Training

- Introduce a loss function $\mathcal{L}(y;\hat{y})$ and minimize it:

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \mathcal{L}(y;\hat{y}) \tag{1}$$

  - A common choice for regression is the sum of squared error:

$$\mathcal{L}(\boldsymbol{w}) = \sum_{i=1}^{N} (y_i - f(\boldsymbol{x}_i;\boldsymbol{w}))^2 \tag{2}$$

- To control over-fitting add a regularization term:

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \left[\sum_{i=1}^{N} \mathcal{L}(y_i;\hat{y}_i) + \lambda |\boldsymbol{w}|_2^{\alpha} \right] \tag{3}$$

  - Setting $\alpha = 2$ leads to the L2 or Ridge regularization, $\alpha = 1$ to L1

Neural Networks Training: Probabilistic perspective

- We may explicitly model the **aleatoric noise** ϵ inherent to the data

$$y(\boldsymbol{x}) = f(\boldsymbol{x}; \boldsymbol{w}) + \epsilon(\boldsymbol{x}) \tag{4}$$

  - One common assumption is gaussian noise $\epsilon(\boldsymbol{x}) = \mathcal{N}(0, \sigma^2)$
- The loss function is viewed as the negative log likelihood $p(\mathcal{D}|\boldsymbol{w}, I)$:
- Under the assumption of i.i.d. additive gaussian noise the likelihood and the loss function are :

$$p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{w}, \sigma^2) = \prod_{i=1}^{N} \mathcal{N}(y_i | f(\boldsymbol{x}_i; \boldsymbol{w}), \sigma^2) \tag{5}$$

$$\mathcal{L}(\boldsymbol{w}) = \frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - f(\boldsymbol{x}_i; \boldsymbol{w}))^2 + \text{const.} \tag{6}$$

Neural Networks Training: Probabilistic perspective II

- Similarly, the regularizer is interpreted as a log-prior probability distribution over the models' parameters $p(\boldsymbol{w}|I)$.
- Using Bayes Theorem we obtain the posterior distribution over the parameters:

$$p(\boldsymbol{w}|\mathcal{D}, I) = \frac{p(\mathcal{D}|\boldsymbol{w}, I)p(\boldsymbol{w}|I)}{p(\mathcal{D})} \tag{7}$$

- The optimal $\hat{\boldsymbol{w}}$ is obtained by maximizing the log posterior :

$$\hat{\boldsymbol{w}}_{\mathsf{MAP}} = \arg\max_{\boldsymbol{w}} \left[\log(p(\mathcal{D}|\boldsymbol{w}, I)) + \log(p(\boldsymbol{w}|I)) + \text{const.}\right] \tag{8}$$

  - The adoption of a normal distribution as prior recovers the L2 regularization term, while a Laplace distribution recovers the L1.

Introduction to Bayesian Neural Networks

- At the end of the training we have a point estimate for the parameters $\hat{\boldsymbol{w}}$
- **Goal:** Quantifying uncertainty on the prediction of unseen inputs $\boldsymbol{x}^*$
- Deep Neural Networks do not fully capture uncertainty[2] [3]
- We have to take into account the **epistemic or model uncertainty** arising from the uncertainty associated to $\hat{\boldsymbol{w}}$

When combined with probability theory NN can capture uncertainty in a principled way: Bayesian Neural Network

[2] Yarin Gal and Zoubin Ghahramani. *Dropout as a bayesian approximation: Representing model uncertainty in deep learning*. 2016.

[3] Andrew G Wilson and Pavel Izmailov. *Bayesian deep learning and a probabilistic perspective of generalization*. 2020.

Outline

Predictive distribution

- Bayesian inference starts from a model $p(y|\boldsymbol{x}, \boldsymbol{w}, I)$ and the posterior $p(\boldsymbol{w}|\mathcal{D})$
- The prediction for a new input $\boldsymbol{x}^*$ is given by the predictive distribution:

$$p(y|\boldsymbol{x}^*, \mathcal{D}, I) = \int p(y|\boldsymbol{x}^*, \boldsymbol{w}, I)p(\boldsymbol{w}|\mathcal{D})d\boldsymbol{w} \tag{9}$$

- It is a Bayesian model average of many models, weighted by their posterior probabilities
- The non Bayesian predictions are recovered if $p(\boldsymbol{w}|\mathcal{D}) \sim \delta(\boldsymbol{w} - \hat{\boldsymbol{w}}_{\mathsf{MAP}})$

Predictive distribution II

- Eq. (9) is the core of Bayesian NN: marginalize over the posterior distribution of the weights rather than optimize it!
- **Problem**: It is highly non trivial to evaluate the predictive distribution
- **Warning**: We need to decouple the epistemic and aleatoric uncertainty in the predictive distribution
- **Warning**: Small uncertainties do not imply good predictive performance

To get some insights let's start from the simplest NN: Linear Regression

Interlude: Linear Regression

- Data : $y = 1 + x + x^2$ and noise $\mathcal{N}(0, \sigma^2 = 1)$
- Model: $y(x) = w_1 + w_2 x + w_3 x^2 = \boldsymbol{w}^T \boldsymbol{\phi}(x)$, homeschedastic gaussian noise, gaussian prior
- Log-posterior of the model:

$$\log(p(\boldsymbol{w}|\mathcal{D})) = \frac{1}{2}\sum_{i=1}^{N}(y_i - \boldsymbol{w}^T\boldsymbol{\phi}(x_i))^2 + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w};\ \ \lambda = 0.001 \tag{10}$$

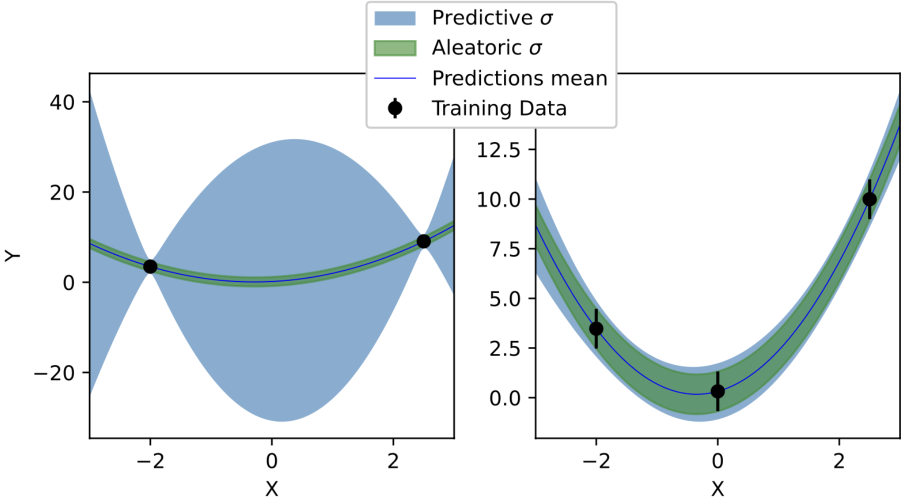
- It is possible to show that the predictive distribution is given by[4]:

$$p(y|x, \mathcal{D}, \sigma_{\mathsf{P}}^2) = \mathcal{N}(y|\hat{\boldsymbol{w}}_{\mathsf{MAP}}^T\boldsymbol{\phi}(x), \sigma_{\mathsf{P}}^2) \tag{11}$$

$$\sigma_{\mathsf{P}}^2 = \underbrace{\sigma^2}_{\text{Aleatoric}} + \underbrace{\boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{S}\boldsymbol{\phi}(\boldsymbol{x})}_{\text{Epistemic}} \tag{12}$$

[4] C. Bishop. *Pattern Recognition and Machine Learning*. 2006.

Interlude: Linear Regression II



Evaluating the predictive distribution

$$P(y|\boldsymbol{x}^*, \mathcal{D}) = \int p(y|\boldsymbol{w}, \boldsymbol{x}^*)p(\boldsymbol{w}|\mathcal{D})d\boldsymbol{w} \tag{13}$$

- **Problem**: It is highly non trivial to evaluate the predictive distribution

- Many possible **approaches**:
  - MCMC
  - Laplace approximation
  - Variational Inference

- **Variational Inference:** Approximate the posterior $p(\boldsymbol{w}|\mathcal{D})$ with a tractable p.d.f. $q(\boldsymbol{w}|\boldsymbol{\theta})$

- **Problem**: We need do adjust $\boldsymbol{\theta}$ to get the best aproximation

Variational Inference I

- The objective function for measuring the quality of the approximation may be derived from the Kullback-Leibler divergence:

$$\mathsf{KL}\left[q(\boldsymbol{w}|\boldsymbol{\theta})||p(\boldsymbol{w}|\mathcal{D})\right] = \int q(\boldsymbol{w}|\boldsymbol{\theta})\mathsf{log}\frac{q(\boldsymbol{w}|\boldsymbol{\theta})}{p(\boldsymbol{w}|\mathcal{D})}d\boldsymbol{w}$$

- Using Bayes' theorem $p(w|\mathcal{D}) = (p(\mathcal{D}|w)p(w))/p(\mathcal{D})$ and re-arranging the terms we obtain:

$$\int q(\boldsymbol{w}|\boldsymbol{\theta})\mathsf{log}p(\mathcal{D}|\boldsymbol{w})d\boldsymbol{w} - \mathsf{KL}\left[q(\boldsymbol{w}|\boldsymbol{\theta})||p(\boldsymbol{w})\right] = \mathsf{log}(p(\mathcal{D})) - \mathsf{KL}\left[q(\boldsymbol{w}|\boldsymbol{\theta})||p(\boldsymbol{w}|\mathcal{D})\right]$$

Variational Inference II

$$\int q(\boldsymbol{w}|\boldsymbol{\theta})\mathsf{log}p(\mathcal{D}|\boldsymbol{w})d\boldsymbol{w} - \mathsf{KL}\left[q(\boldsymbol{w}|\boldsymbol{\theta})||p(\boldsymbol{w})\right] = \mathsf{log}(p(\mathcal{D})) - \mathsf{KL}\left[q(\boldsymbol{w}|\boldsymbol{\theta})||p(\boldsymbol{w}|\mathcal{D})\right]$$

- The last term is positive and $\mathsf{log}(p(\mathcal{D}))$ is constant so:

$$\int q(\boldsymbol{w}|\boldsymbol{\theta})\mathsf{log}p(\mathcal{D}|\boldsymbol{w})d\boldsymbol{w} - \mathsf{KL}\left[q(\boldsymbol{w}|\boldsymbol{\theta})||p(\boldsymbol{w})\right] \leq \mathsf{log}(p(\mathcal{D}))$$

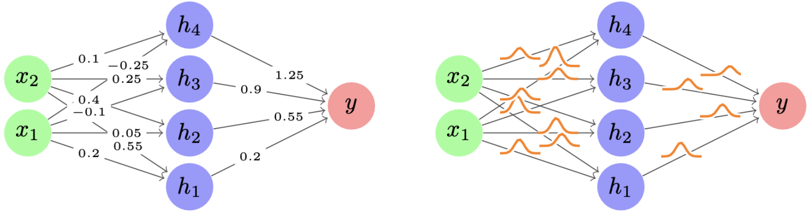
- The left hand side term will be our objective function, known as variational free energy or ELBO :

$$\mathcal{F}(\boldsymbol{\theta}) = \mathsf{KL}\left[q(\boldsymbol{w}|\boldsymbol{\theta})||p(\boldsymbol{w})\right] - \mathbb{E}_{q(\boldsymbol{w}|\boldsymbol{\theta})}[\mathsf{log}(p(\mathcal{D}|\boldsymbol{w}))]$$

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \mathcal{F}(\boldsymbol{\theta})$$

Variational Inference III

- Common choice is a diagonal gaussian distribution as approximant distribution (Mean Field Approximantion)
- Backpropagation-compatible algorithm[5]



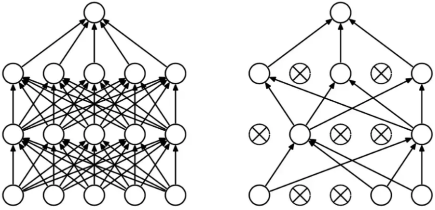
- The predictive distribution for a given input $\boldsymbol{x}^*$ is approximate as:

$$p(y|\boldsymbol{x}^*) \approx \frac{1}{N}\sum_{i=1}^{N} p(y|\boldsymbol{x}^*, \boldsymbol{w}_i); \quad \boldsymbol{w}_i \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}) \tag{14}$$

[5] Charles Blundell et al. *Weight uncertainty in neural network*. PMLR, 2015.

MC-Dropout

- Drop out each hidden unit by sampling from a Bernoulli distribution N times[6] [7] :



(a) Standard Neural Net (b) After applying dropout.

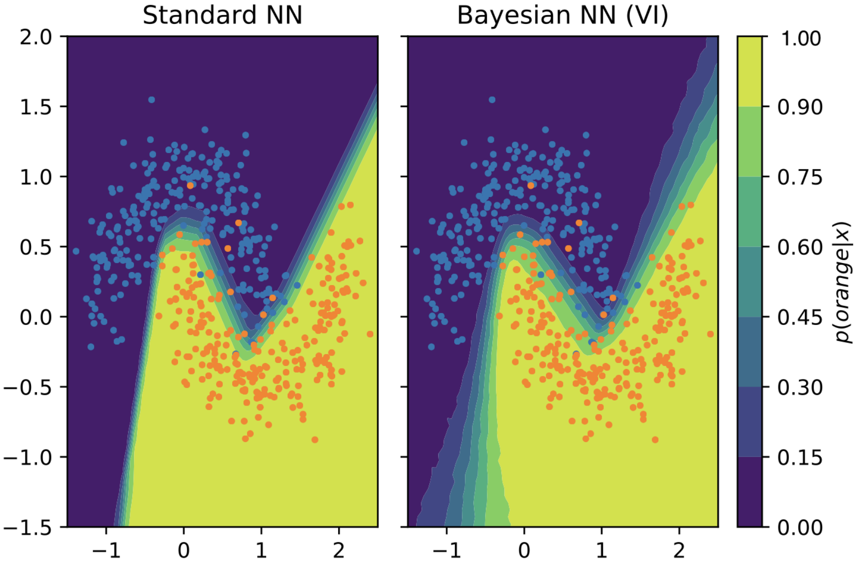
$$p(y|\boldsymbol{x}^*) \approx \frac{1}{N}\sum_{i=1}^{N} p(y|\boldsymbol{x}^*, \boldsymbol{w}_i) \tag{15}$$

- Applicable also to Recurrent Neural Networks

[6] Gal and Ghahramani, *Dropout as a bayesian approximation: Representing model uncertainty in deep learning*.

[7] Yarin Gal, Jiri Hron, and Alex Kendall. *Concrete dropout*. 2017.

Interlude: NN for classification



Epistemic and Aleatoric decoupling: Regression

- Denoting the predictive distribution as $p(y|\boldsymbol{x}, \mathcal{D})$ and the predictive variance as $\mathsf{Var}[y]$:

$$\mathsf{Var}[y] = \underbrace{\mathsf{Var}_{p(\boldsymbol{w}|\mathcal{D})}[\mathbb{E}_{p(y|\boldsymbol{x},\boldsymbol{w})}[y]]}_{\text{Epistemic}} + \underbrace{\mathbb{E}_{p(\boldsymbol{w}|\mathcal{D})}[\mathsf{Var}_{p(y|\boldsymbol{x},\boldsymbol{w})}[y]]}_{\text{Aleatoric}} \tag{16}$$

- For instance, homoschedastic gaussian noise and MC-dropout:

$$\mathbb{E}[y] \approx \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{x}, \boldsymbol{w}_i) \tag{17}$$

$$\mathsf{Var}[y] \approx \frac{1}{N}\sum_{i=1}^{N}(f(\boldsymbol{x}, \boldsymbol{w}_i) - \mathbb{E}[y])^2 + \sigma^2 \tag{18}$$

Epistemic and Aleatoric decoupling: Classification

- Typically a NN is trained to predictit the posterior distribution over K exclusive and exaustive classes, trough the softmax activation function
- The total uncertainty can be estimated trough the Shannon Entropy:

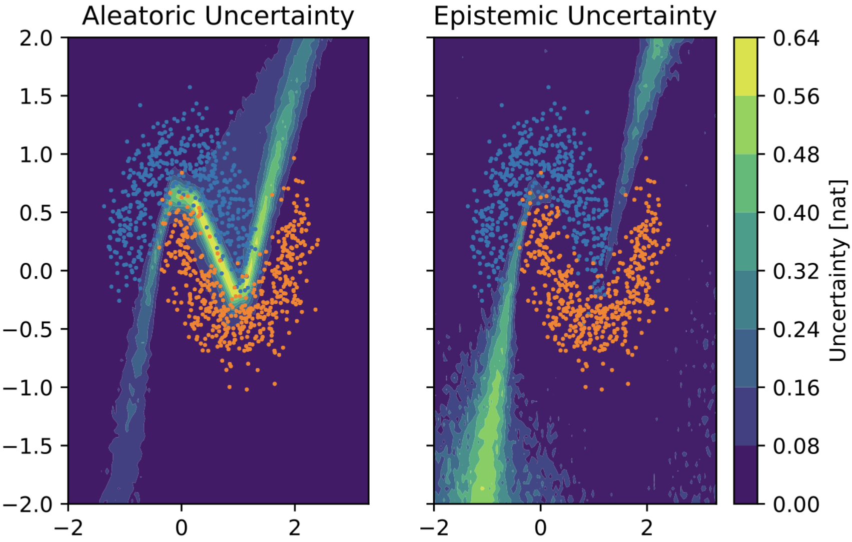
$$\mathbb{H}[\boldsymbol{y}] = \sum_i p(y_i)\ln(p(y_i)) \tag{19}$$

- Maximized in case of a flat distribution
- It can be decomposed as :

$$\mathbb{H}[p(\boldsymbol{y}|\boldsymbol{x},\mathcal{D})] = \underbrace{\mathbb{I}[\boldsymbol{y},\boldsymbol{w}|\boldsymbol{x},\mathcal{D}]}_{\text{Epistemic}} + \underbrace{\mathbb{E}_{p(\boldsymbol{w}|\mathcal{D})}[\mathbb{H}[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w})]]}_{\text{Aleatoric}} \tag{20}$$

where $\mathbb{I}[\boldsymbol{y},\boldsymbol{w}|\boldsymbol{x},\mathcal{D}]$ is the information gain about the model parameters

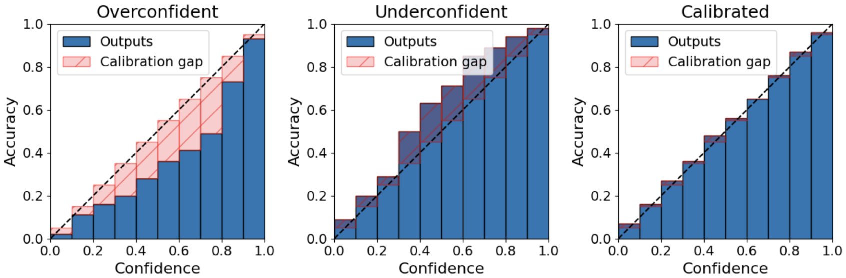
Epistemic and Aleatoric decoupling: Classification



Outline

Quality estimates

- Main idea: compute the predicted confidence interval, and count the percentage of ground-truth points that fall inside.
- For a calibrated model we expect, on average, $X\%$ of ground-truth points falling inside the predicted $X\%$ confidence intervals.
- For classification compare the predicted probabilities and the empirical frequency of correct labels.

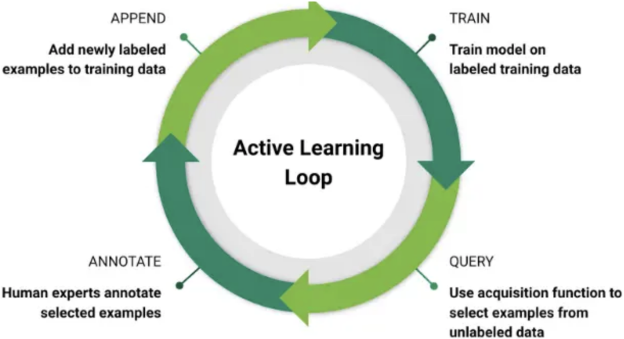


8

[8] Gawlikowski et al., *A survey of uncertainty in deep neural networks*.

Interlude: Active Learning

- Deep learning often requires large amounts of labelled data
- Train a model by querying as few labelled data as possible
- Active Learning:



- Label only informative points: **epistemic uncertainty** (BALD)

Interlude: Active Learning II

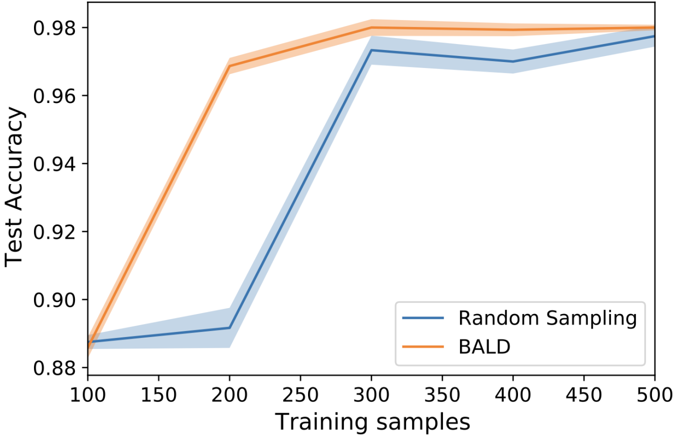


Figure: Mean test accuracy and standard deviation on the two moon dataset as a function of the training size. Results are averaged over multiple training loops.

Further Topics

- Deep Ensamble[9]
- MultiSWAG[10]
- Evidential Regression and classification[11] [12]
- Conformal prediction[13]
  - distribution-free uncertainty quantification method
  - provides prediction sets with guaranteed frequentist coverage probability. Even with a completely misspecified models!
  - cannot distinguish between epistemic and aleatoric uncertainty

[9] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. *Simple and scalable predictive uncertainty estimation using deep ensembles*. 2017.

[10] Wilson and Izmailov, *Bayesian deep learning and a probabilistic perspective of generalization*.

[11] Alexander Amini et al. *Deep evidential regression*. 2020.

[12] Murat Sensoy, Lance Kaplan, and Melih Kandemir. *Evidential deep learning to quantify classification uncertainty*. 2018.

[13] Anastasios N Angelopoulos and Stephen Bates. *A gentle introduction to conformal prediction and distribution-free uncertainty quantification*. 2021.

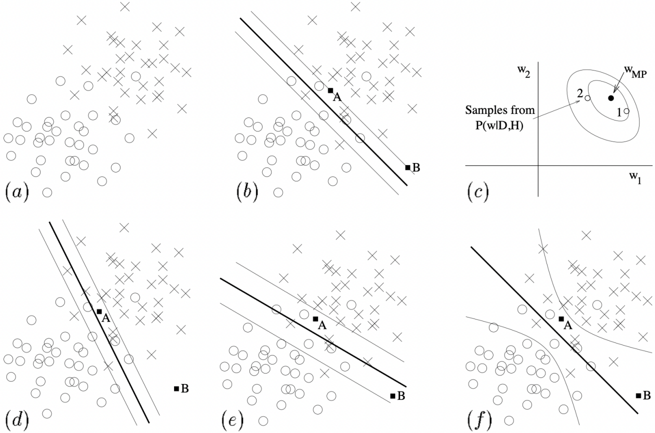
Further readings

- Books:
  - Information theory, inference and learning algorithms. MacKay, David JC. Cambridge university press, 2003.
  - Pattern Recognition and Machine Learning. Christopher M. Bishop. Springer New York, 2006
  - Probabilistic machine learning: an introduction. Kevin P. Murphy .MIT press, 2022.
  - Probabilistic machine learning: Advanced topics. Kevin P. Murphy. MIT Press, 2023.
- High Energy Physics applications:
  - Chapter 18 Artificial Intelligence for High Energy Physics. P. Calafiura, D. Rousseau, K. Terao. WorldScientific 2022
  - Bollweg, Sven, et al. "Deep-learning jets with uncertainties and more." SciPost Physics 8.1 2020
  - Araz, Jack Y., and Michael Spannowsky. "Combine and conquer: event reconstruction with Bayesian ensemble neural networks." Journal of High Energy Physics 2021.4

Thank you for your attention!

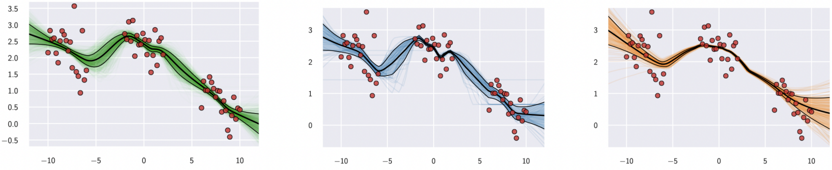
Backup: Overconfident classification

- From "Probable Networks and Plausible Predictions - A Review of Practical Bayesian Methods for Supervised Neural Networks" by David MacKay, 1996



Backup: Regression

- From "Bayesian Deep Learning and a Probabilistic Perspective of Generalization" by Andrew Gordon Wilson Pavel Izmailov, 2022



(a) Exact (b) Deep Ensembles (c) Variational Inference

Bayes by Backprop

- For applying back-propagation we have to replace the derivative of an expectation with the expectation of the derivative

Proposition 1

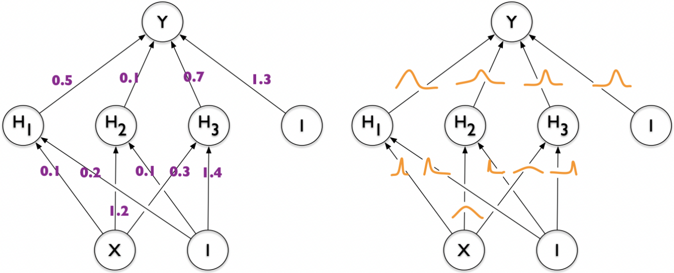
1. Let ϵ a random variable with p.d.f. $q(\epsilon)$ and $\boldsymbol{w} = t(\theta, \epsilon)$ where t is a deterministic function
2. Assuming that $q(\boldsymbol{w}|\theta)$ is such that $q(\epsilon)d\epsilon = q(\boldsymbol{w}|\theta)d\boldsymbol{w}$
3. Then, for a function f with derivatives in $\boldsymbol{w}$ we have:

$$\frac{\partial}{\partial\theta}\mathbb{E}_{q(\boldsymbol{w}|\theta)}[f(\boldsymbol{w}, \theta)] = \mathbb{E}_{q(\epsilon)}\left[\frac{\partial f(\boldsymbol{w}, \theta)}{\partial \boldsymbol{w}}\frac{\partial \boldsymbol{w}}{\partial\theta} + \frac{\partial f(\boldsymbol{w}, \theta)}{\partial\theta}\right]$$

- Basically, It is a generalization of the "Reparametrization Trick"

Gaussian Variational Posterior

- Variational posterior is a diagonal gaussian distribution
- Parametrization trick: a sample of $\boldsymbol{w}$ is given by a deterministic function of a random variable: $\boldsymbol{w} = t(\theta, \epsilon) = \mu + \log(1 + \exp(\rho))\epsilon$ where: $\epsilon \in \mathcal{N}(0, I)$
- The objective function is: $f(\boldsymbol{w}, \theta) = \log q(\boldsymbol{w}|\theta) - \log(P(\boldsymbol{w})P(\mathcal{D}|\boldsymbol{w}))$
- Update variational parameters: $\mu^* \leftarrow \mu - \alpha\Delta_\mu$; $\rho^* \leftarrow \rho - \alpha\Delta_\rho$



Backup: Reparametrization trick

- From "An Introduction to Variational Autoencoders" by Diederik P. Kingma and Max Welling, 2019