Bayesian Inference for Deep Learning

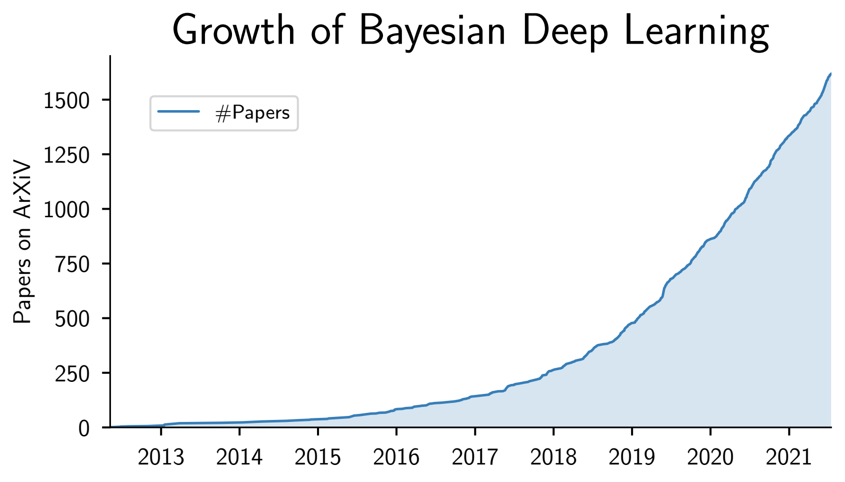
Introduction to Bayesian Inference for Deep Learning

Simone Rossi and Maurizio Filippone

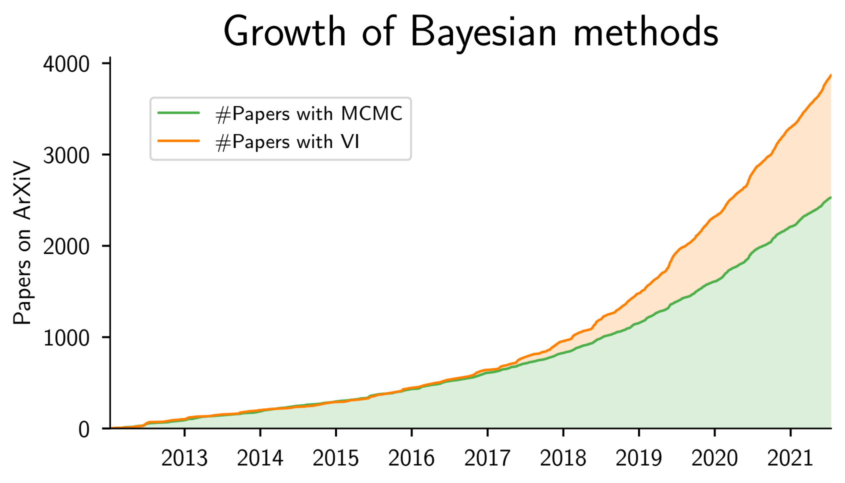
Data Science Department, EURECOM (France)

Introduction

Why a tutorial on Bayesian deep learning?

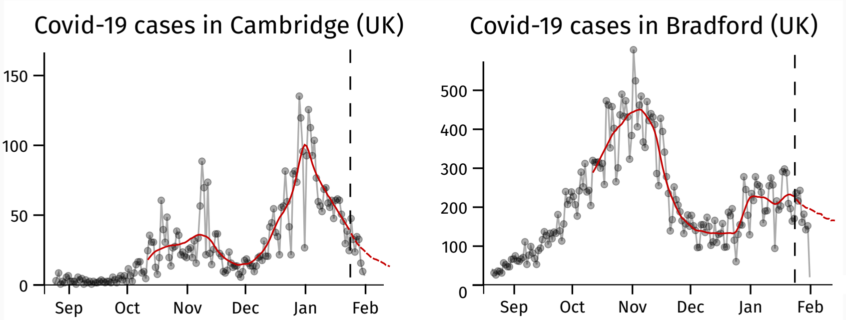


Why a tutorial on Bayesian deep learning?



Role of uncertainty today

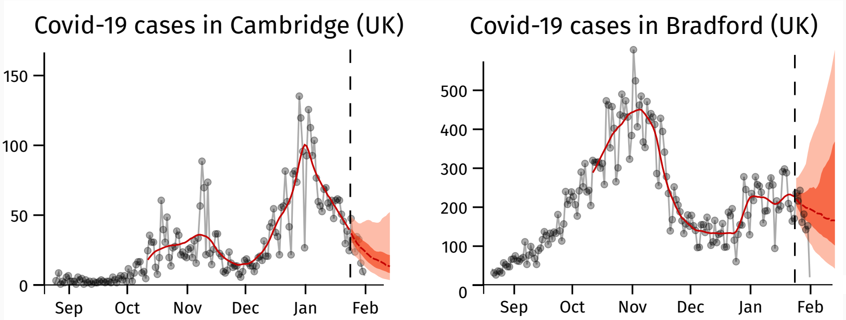
Accounting for uncertainty, if possible, is important



From https://localcovid.info/

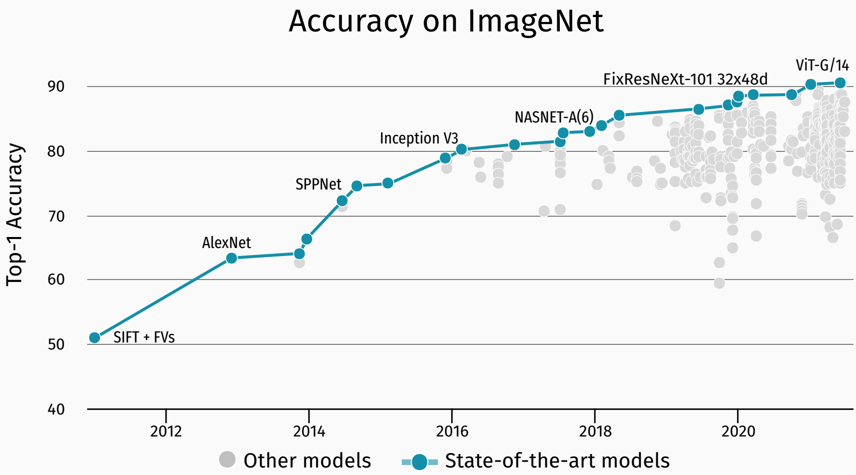
Role of uncertainty today

Accounting for uncertainty, if possible, is important



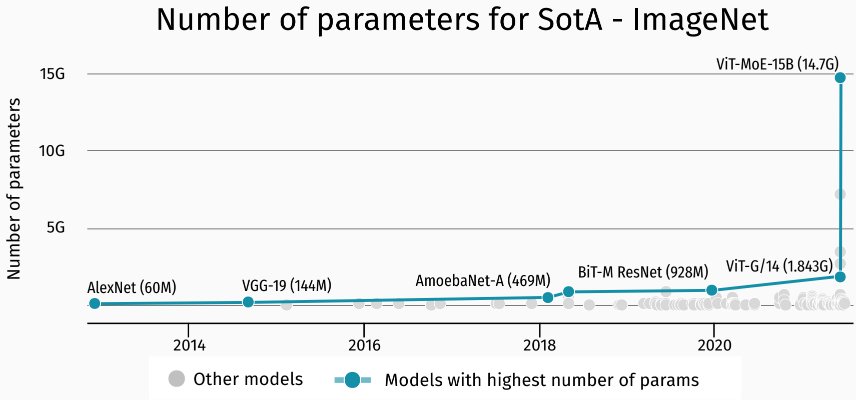
From https://localcovid.info/

The race to bigger and more accurate models



https://paperswithcode.com/sota/image-classification-on-imagenet

The race to bigger and more accurate models



https://paperswithcode.com/sota/image-classification-on-imagenet

Bayesian inference:
A quick cheat sheet

A quick introduction to Bayesian inference: Syntax of probabilities

Consider two continuous random variables a and b

- Sum rule:

$$p(a) = \int p(a, b) db$$

- Product rule:

$$p(a, b) = p(a|b)p(b) = p(b|a)p(a)$$

A quick introduction to Bayesian inference: Syntax of probabilities

Consider two continuous random variables a and b

- Sum rule:

$$p(a) = \int p(a, b) db$$

- Product rule:

$$p(a, b) = p(a|b)p(b) = p(b|a)p(a)$$

- Bayes' rule:

$$p(b|a) = \frac{p(a|b)p(b)}{p(a)}$$

Note: Bayes' rule is a direct consequence of the product rule

A quick introduction to Bayesian inference: Expectations

- Expectations:

$$\overline{f} = \mathsf{E}_{p(a)}\left[f(a)\right] = \int f(a)\, p(a)\, da$$

- Example: the mean

$$\mu = \mathsf{E}_{p(a)}\left[a\right] = \int a\, p(a)\, da$$

A quick introduction to Bayesian inference: Expectations

- Expectations:

$$\overline{f} = \mathrm{E}_{p(a)}\left[f(a)\right] = \int f(a)\, p(a)\, da$$

- Example: the mean

$$\mu = \mathrm{E}_{p(a)}\left[a\right] = \int a\, p(a)\, da$$

- Monte Carlo estimate by averaging over samples from $p(a)$:

$$\overline{f} \approx \frac{1}{N}\sum_{i=1}^{N} f(a_i) \qquad \text{with} \qquad a_i \sim p(a)$$

Definitions

- Data is a set of N inputs/labels pairs:

$$\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1,\ldots,N} \qquad \text{with}$$

$$\mathbf{x} \in \mathbb{R}^D, \qquad \mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^\top \qquad \text{and}$$
$$\mathbf{y} \in \mathbb{R}^O, \qquad \mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)^\top$$

Definitions

- Data is a set of N inputs/labels pairs:

$$\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1,\dots,N} \qquad \text{with}$$

$$\mathbf{x} \in \mathbb{R}^D, \qquad \mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)^\top \qquad \text{and}$$

$$\mathbf{y} \in \mathbb{R}^O, \qquad \mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_N)^\top$$

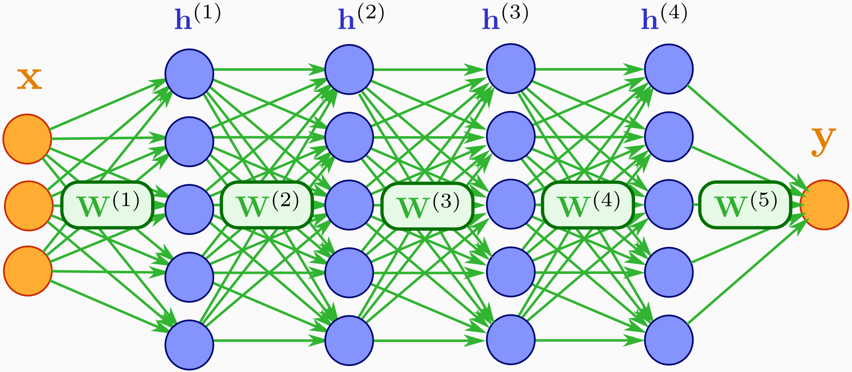
- **Goal**: Estimate a function

$$\mathbf{f}(\mathbf{x}) : \mathbb{R}^D \to \mathbb{R}^O$$

Deep Neural Networks

- Implement a composition of parametric functions

$$\mathbf{f}(\mathbf{x}) = \mathbf{h}^{(L)}\left(\mathbf{h}^{(L-1)}\left(\cdots \mathbf{h}^{(1)}(\mathbf{x})\right)\right) \quad \text{with} \quad \mathbf{h}^{(l)} = a\left(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)}\right)$$



Notation: the bias is included in $\mathbf{W}$

Loss Minimization – Regression

- Define $\mathbf{w} = \{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L)}\}$
- Definition of the quadratic loss function:

$$\mathcal{L} = \sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{f}(\mathbf{x}_i; \mathbf{w})\|^2$$

- Solution to the regression problem not available in closed form:

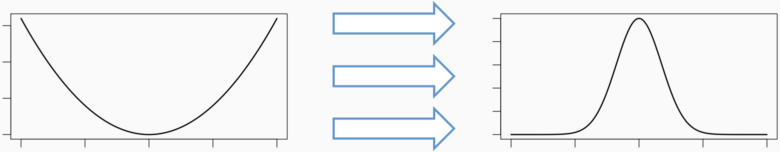
$$\nabla_{\mathbf{w}} \mathcal{L} = \mathbf{0}$$

- Back-propagation to calculate gradients to perform optimization of $\mathcal{L}$ wrt $\mathbf{w}$

Probabilistic Interpretation of Loss Minimization

- Consider a simple transformation of the loss function

$\|\mathbf{y}_i - \mathbf{f}(\mathbf{x}_i; \mathbf{w})\|^2$ $\quad\Rightarrow\quad$ $\exp\left(-\gamma\|\mathbf{y}_i - \mathbf{f}(\mathbf{x}_i; \mathbf{w})\|^2\right)$



- Minimizing the quadratic loss equivalent to maximizing the Gaussian likelihood function

$$\begin{aligned}\exp\left(-\gamma\mathcal{L}\right) &= \prod_i \exp\left(-\gamma\|\mathbf{y}_i - \mathbf{f}(\mathbf{x}_i; \mathbf{w})\|^2\right) \\ &\propto \mathcal{N}\left(\mathbf{Y} \,\middle|\, \mathbf{F}, \frac{1}{2\gamma}\mathbf{I}_{N\times O}\right) \qquad \text{Gaussian distribution}\end{aligned}$$

Probabilistic Interpretation of Loss Minimization

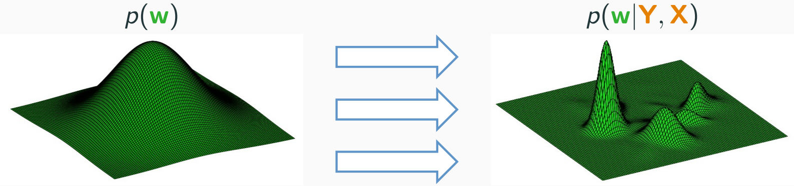
- The likelihood $\mathcal{N}\left(\mathbf{y}|\mathbf{X}\mathbf{w}, \frac{1}{2\gamma}\right)$ hints to the fact that we are assuming:

$$\mathbf{y}_i = \mathbf{f}(\mathbf{x}_i; \mathbf{w}) + \boldsymbol{\epsilon}_i$$

  with $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\varepsilon_i|0, \frac{1}{2\gamma}\mathbf{I}_O)$
- Remark: the likelihood is not a probability!

Bayesian Inference

- Viewing parameters and data as random variables, we can use Bayes Theorem as follows:



$$p(\mathbf{w}|\mathbf{Y}, \mathbf{X}) = \frac{p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{\int p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})d\mathbf{w}}$$

Bayesian Inference

- Bayes rule:

$$p(\mathbf{w}|\mathbf{X},\mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X},\mathbf{w})p(\mathbf{w})}{\int p(\mathbf{Y}|\mathbf{X},\mathbf{w})p(\mathbf{w})d\mathbf{w}} = \frac{p(\mathbf{Y}|\mathbf{X},\mathbf{w})p(\mathbf{w})}{p(\mathbf{Y}|\mathbf{X})}$$

Bayesian Inference

- Bayes rule:

$$p(\mathbf{w}|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{\int p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})d\mathbf{w}} = \frac{p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{p(\mathbf{Y}|\mathbf{X})}$$

- **Likelihood** : $p(\mathbf{Y}|\mathbf{X}, \mathbf{w})$
  - Measure of “fitness”

Bayesian Inference

- Bayes rule:

$$p(\mathbf{w}|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{\int p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})d\mathbf{w}} = \frac{p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{p(\mathbf{Y}|\mathbf{X})}$$

- **Likelihood** : $p(\mathbf{Y}|\mathbf{X}, \mathbf{w})$
  - Measure of "fitness"
- **Prior density**: $p(\mathbf{w})$
  - Anything we know about parameters *before* we see any data

Bayesian Inference

- Bayes rule:

$$p(\mathbf{w}|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{\int p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})d\mathbf{w}} = \frac{p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{p(\mathbf{Y}|\mathbf{X})}$$

- **Likelihood** : $p(\mathbf{Y}|\mathbf{X}, \mathbf{w})$
  - Measure of "fitness"
- **Prior density**: $p(\mathbf{w})$
  - Anything we know about parameters *before* we see any data
- **Posterior density**: $p(\mathbf{w}|\mathbf{X}, \mathbf{Y})$
  - Distribution over parameters *after* observing data

Bayesian Inference

- Bayes rule:

$$p(\mathbf{w}|\mathbf{X},\mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X},\mathbf{w})p(\mathbf{w})}{\int p(\mathbf{Y}|\mathbf{X},\mathbf{w})p(\mathbf{w})d\mathbf{w}} = \frac{p(\mathbf{Y}|\mathbf{X},\mathbf{w})p(\mathbf{w})}{p(\mathbf{Y}|\mathbf{X})}$$

- **Likelihood** : $p(\mathbf{Y}|\mathbf{X},\mathbf{w})$
  - Measure of "fitness"
- **Prior density**: $p(\mathbf{w})$
  - Anything we know about parameters *before* we see any data
- **Posterior density**: $p(\mathbf{w}|\mathbf{X},\mathbf{Y})$
  - Distribution over parameters *after* observing data
- **Marginal likelihood**: $p(\mathbf{Y}|\mathbf{X})$
  - It is a normalization constant – ensures $\int p(\mathbf{w}|\mathbf{X},\mathbf{Y})\, d\mathbf{w} = 1$.

Bayesian Inference - Predictive Distribution

- Predictions can be made in the form of distributions:

$$p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{Y}, \mathbf{X}) = \int p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{w})p(\mathbf{w}|\mathbf{Y}, \mathbf{X})d\mathbf{w}$$

- Notice how parameters disappear from the expression of the predictive distribution!

Bayesian Inference - Model Selection

- Marginal likelihood

$$p(\mathbf{Y}|\mathbf{X}) = \int p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})d\mathbf{w}$$

  depends on the modeling choice.

- For models M_1 and M_2 we have:

$$p(\mathbf{Y}|\mathbf{X}, M_1) \qquad \text{and} \qquad p(\mathbf{Y}|\mathbf{X}, M_2)$$

Bayesian Inference - Model Selection

- Marginal likelihood

$$p(\mathbf{Y}|\mathbf{X}) = \int p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})d\mathbf{w}$$

  depends on the modeling choice.

- For models M_1 and M_2 we have:

$$p(\mathbf{Y}|\mathbf{X}, M_1) \qquad \text{and} \qquad p(\mathbf{Y}|\mathbf{X}, M_2)$$

- We can pick the model with the largest marginal likelihood...

Bayesian Inference - Model Selection

- Marginal likelihood

$$p(\mathbf{Y}|\mathbf{X}) = \int p(\mathbf{Y}|\mathbf{X},\mathbf{w})p(\mathbf{w})d\mathbf{w}$$

  depends on the modeling choice.

- For models M_1 and M_2 we have:

$$p(\mathbf{Y}|\mathbf{X}, M_1) \qquad \text{and} \qquad p(\mathbf{Y}|\mathbf{X}, M_2)$$

- We can pick the model with the largest marginal likelihood...
- ...or we can assign priors $p(M_1)$ and $p(M_2)$ and use Bayes theorem to obtain:

$$p(M_i|\mathbf{Y},\mathbf{X}) = \frac{p(\mathbf{Y}|\mathbf{X}, M_i)p(M_i)}{\sum_j p(\mathbf{Y}|\mathbf{X}, M_j)p(M_j)}$$

Bayesian Linear Regression - Example

- Polynomial regression with Bayesian linear models:

$$f(x) = \sum_{i=0}^{k} w_i x^i$$

- The model is linear in the parameters but can model functions through polynomials
- Define:

$$\mathbf{\Phi} = \begin{bmatrix} \varphi_1(\mathbf{x}_1) & \dots & \varphi_D(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \varphi_1(\mathbf{x}_N) & \dots & \varphi_D(\mathbf{x}_N) \end{bmatrix}$$

Bayesian Linear Regression - Example

- Polynomial regression with Bayesian linear models:

$$f(x) = \sum_{i=0}^{k} w_i x^i$$

- The model is linear in the parameters but can model functions through polynomials
- Define:

$$\mathbf{\Phi} = \begin{bmatrix} \varphi_1(\mathbf{x}_1) & \dots & \varphi_D(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \varphi_1(\mathbf{x}_N) & \dots & \varphi_D(\mathbf{x}_N) \end{bmatrix}$$

- Assume a Gaussian likelihood:

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \mathcal{N}(\mathbf{y}|\mathbf{\Phi}\mathbf{w}, \sigma^2\mathbf{I})$$

- Assume a Gaussian prior:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{\Sigma})$$

Bayesian Linear Regression - Example

- The posterior is Gaussian:

$$p(\mathbf{w}|\mathbf{X}, \mathbf{y}, \sigma^2) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{S})$$

- Covariance:

$$\boldsymbol{\Sigma} = \left(\frac{1}{\sigma^2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \mathbf{S}^{-1}\right)^{-1}$$

- Mean:

$$\boldsymbol{\mu} = \frac{1}{\sigma^2}\boldsymbol{\Sigma}\boldsymbol{\Phi}^\top\mathbf{y}$$

- The predictive distribution is also Gaussian:

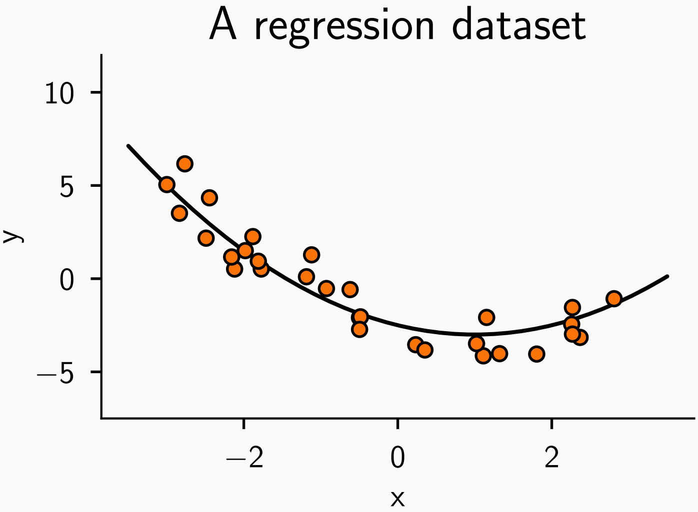
$$p(y_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*, \sigma^2) = \mathcal{N}(y_*|\varphi(\mathbf{x}_*)^\top\boldsymbol{\mu}, \sigma^2 + \varphi(\mathbf{x}_*)^\top\boldsymbol{\Sigma}\varphi(\mathbf{x}_*))$$

- The marginal likelihood is Gaussian:

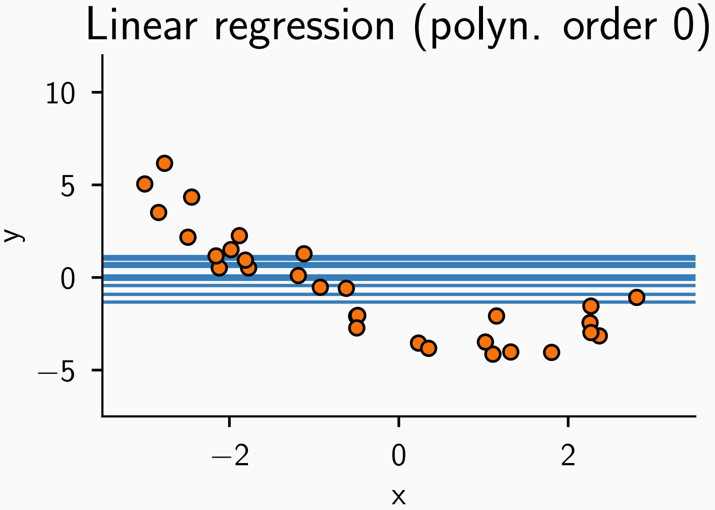
$$p(\mathbf{y}|\mathbf{X}, \mathbf{y}, \sigma^2) = \mathcal{N}(\mathbf{y}|\mathbf{0}, \sigma^2\mathbf{I} + \boldsymbol{\Phi}\mathbf{S}\boldsymbol{\Phi}^\top)$$

Bayesian Linear Regression - Example

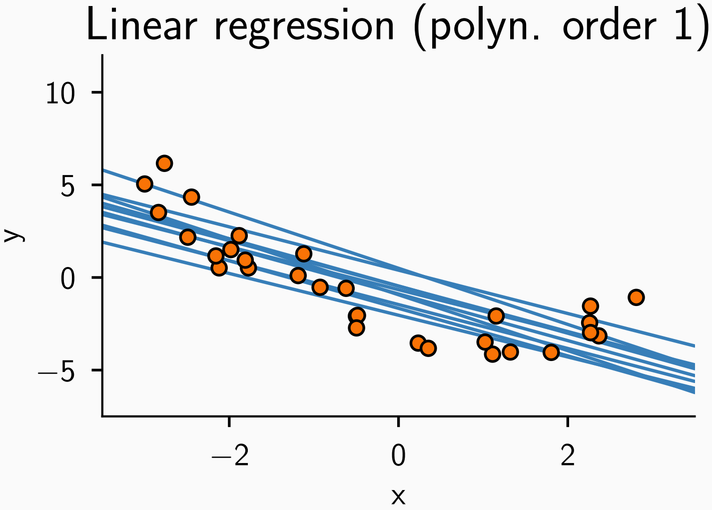
Some data generated from a known polynomial of order $k = 2$

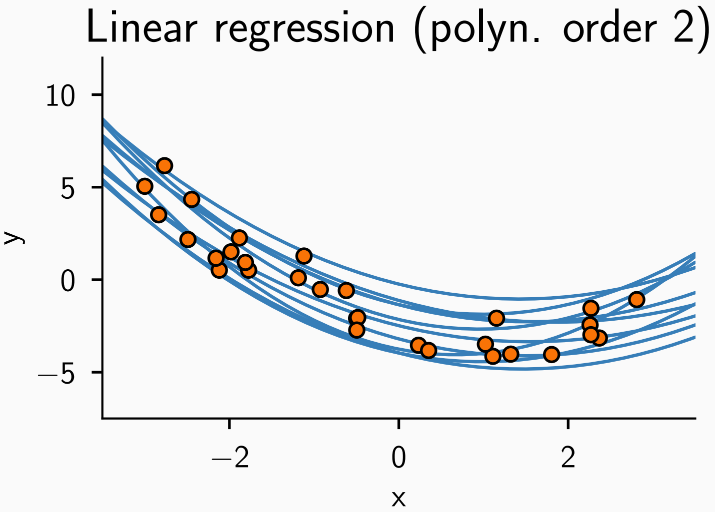


Bayesian Linear Regression - Example

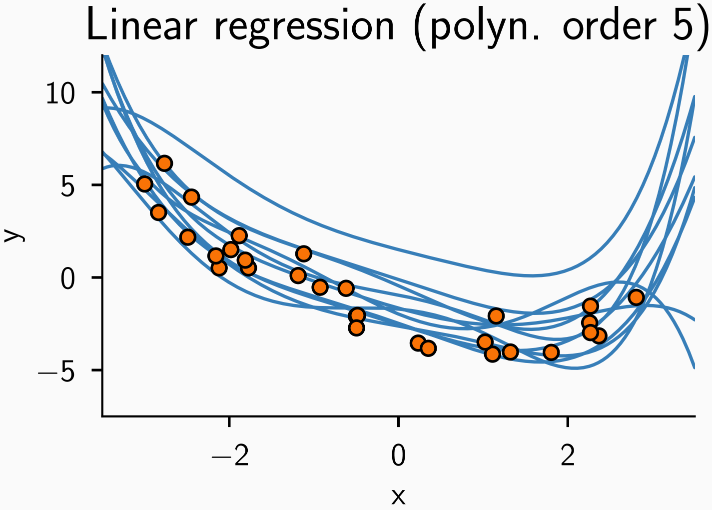


Bayesian Linear Regression - Example

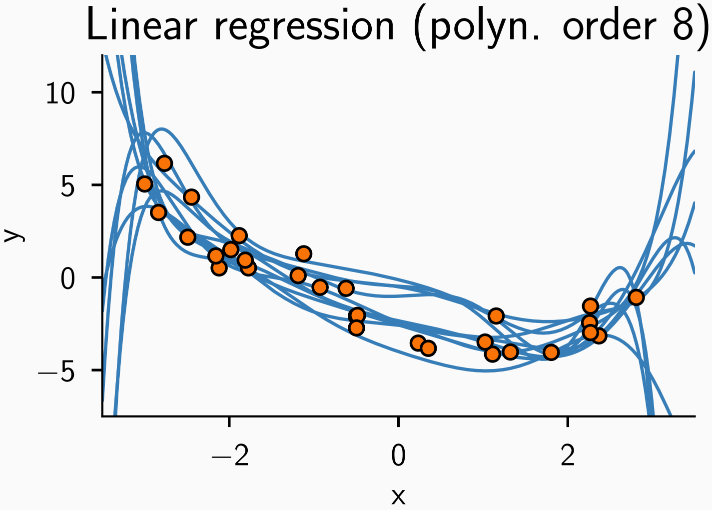


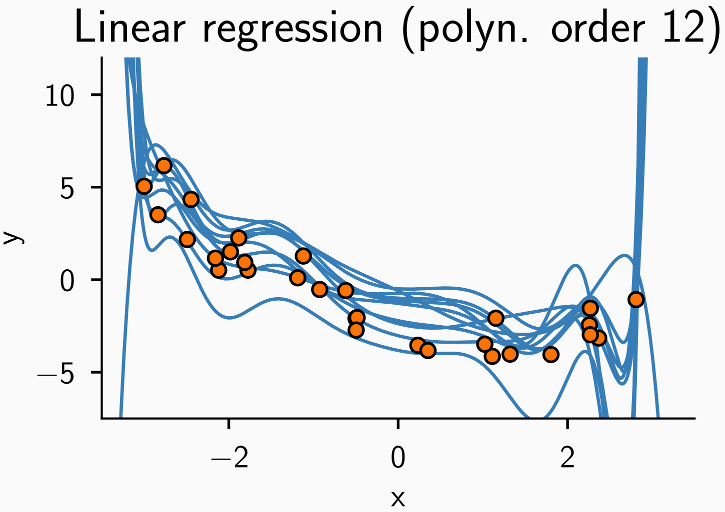

Linear regression (polyn. order 2)
10
5
0
−5
y
−2
0
2
x

Bayesian Linear Regression - Example



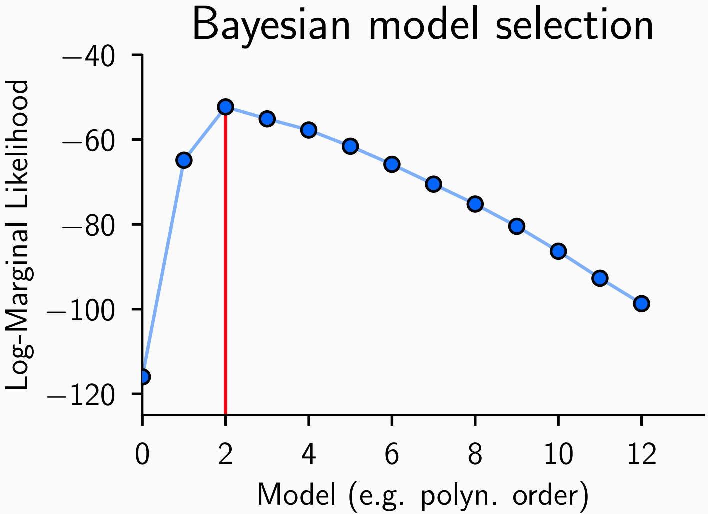
Bayesian Linear Regression - Example




Linear regression (polyn. order 12)
10
5
0
−5
y
−2
0
2
x

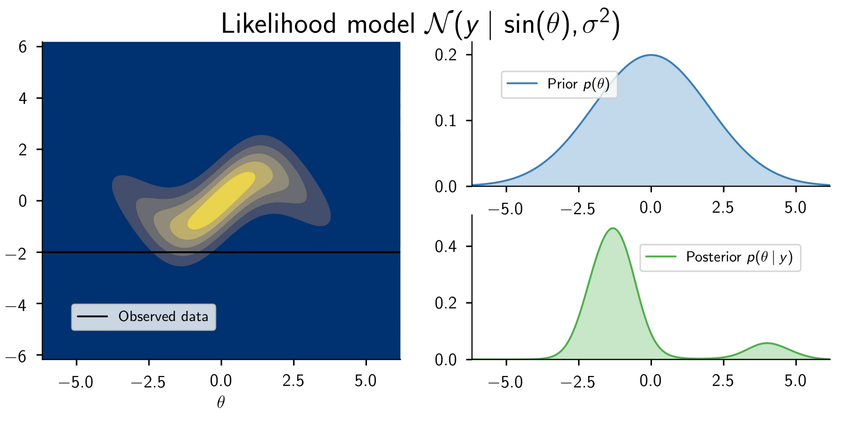
Bayesian Linear Regression - Example

Marginal likelihood as a way to choose the "best" model



Nonlinear Models

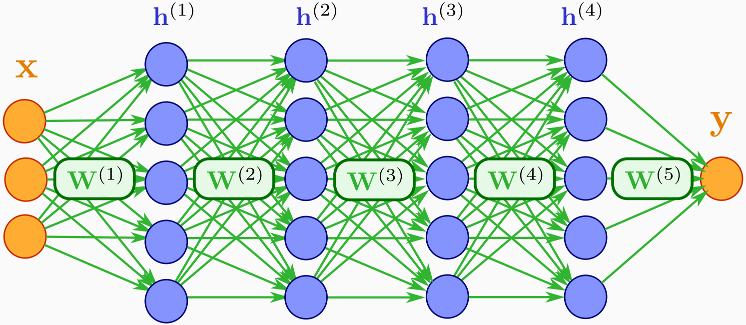
- Illustration of Bayesian inference for a simple nonlinear model.



Back to Deep Neural Networks

- The composition makes the dependence wrt parameter highly nontrivial.

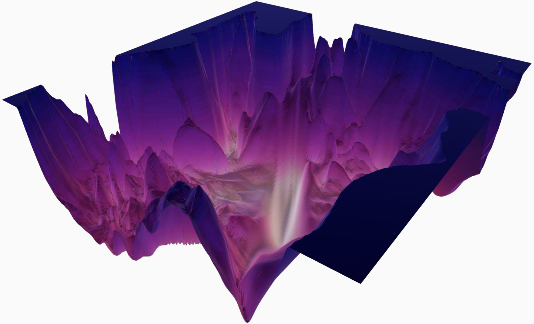
$$\mathbf{f}(\mathbf{x}) = \mathbf{h}^{(L)}\left(\mathbf{h}^{(L-1)}\left(\cdots\mathbf{h}^{(1)}(\mathbf{x})\right)\right) \quad \text{with} \quad \mathbf{h}^{(l)} = a\left(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)}\right)$$



Bayesian Inference for Deep Learning

Applying Bayesian inference to deep neural networks is extremely challenging.

1. How can we work with intractable posterior?
2. How can we handle millions to billions model parameters? What about scalability to big datasets?
3. What kind of priors should we use for these models?
4. Can we trust the uncertainty quantification of Bayesian inference?
5. ...



From losslandscape.com

Roadmap of the tutorial

- **Introduction to Bayesian Inference**
  - A refresh of probability theory and Bayes' theorem
  - Bayesian linear regression: prior, posterior and model selection
- **Bayesian inference as Optimization with Variational Inference**
- **Sampling with MCMC methods**
- **Alternatives for Approximate Bayesian Deep Learning**
- **Gaussian processes and Bayesian neural networks**
- **Priors and Model Selection**
- **Uncertainty Quantification with Bayesian Neural Networks**

Roadmap of the tutorial

- **Introduction to Bayesian Inference**
- **Bayesian inference as Optimization with Variational Inference**
  - Introduction to variational inference (objective and gradients)
  - Challenges and solutions for variational inference on Bayesian neural networks
- **Sampling with MCMC methods**
- **Alternatives for Approximate Bayesian Deep Learning**
- **Gaussian processes and Bayesian neural networks**
- **Priors and Model Selection**
- **Uncertainty Quantification with Bayesian Neural Networks**

Roadmap of the tutorial

- **Introduction to Bayesian Inference**
- **Bayesian inference as Optimization with Variational Inference**
- **Sampling with MCMC methods**
  - Markov-Chain Monte Carlo methods with Metropolis-Hastings
  - Extension to scalable MCMC methods with stochastic gradients
- **Alternatives for Approximate Bayesian Deep Learning**
- **Gaussian processes and Bayesian neural networks**
- **Priors and Model Selection**
- **Uncertainty Quantification with Bayesian Neural Networks**

Roadmap of the tutorial

- **Introduction to Bayesian Inference**
- **Bayesian inference as Optimization with Variational Inference**
- **Sampling with MCMC methods**
- **Alternatives for Approximate Bayesian Deep Learning**
  - Local approximation with the Laplace method
  - Ensembles and Bayesian Bootstrap
- **Gaussian processes and Bayesian neural networks**
- **Priors and Model Selection**
- **Uncertainty Quantification with Bayesian Neural Networks**

Roadmap of the tutorial

- **Introduction to Bayesian Inference**
- **Bayesian inference as Optimization with Variational Inference**
- **Sampling with MCMC methods**
- **Alternatives for Approximate Bayesian Deep Learning**
- **Gaussian processes and Bayesian neural networks**
    - The infinite-limit width of neural networks
    - Deep Gaussian processes and deep Bayesian neural networks
- **Priors and Model Selection**
- **Uncertainty Quantification with Bayesian Neural Networks**

Roadmap of the tutorial

- **Introduction to Bayesian Inference**
- **Bayesian inference as Optimization with Variational Inference**
- **Sampling with MCMC methods**
- **Alternatives for Approximate Bayesian Deep Learning**
- **Gaussian processes and Bayesian neural networks**
- **Priors and Model Selection**
  - Practical ways to choose priors and models
- **Uncertainty Quantification with Bayesian Neural Networks**

Roadmap of the tutorial

- **Introduction to Bayesian Inference**
- **Bayesian inference as Optimization with Variational Inference**
- **Sampling with MCMC methods**
- **Alternatives for Approximate Bayesian Deep Learning**
- **Gaussian processes and Bayesian neural networks**
- **Priors and Model Selection**
- **Uncertainty Quantification with Bayesian Neural Networks**
    - Calibration of uncertainty
    - Challenges of out-of-distribution data

Tutorials

- Iain Murray. *Monte Carlo Inference Methods*. NeurIPS 2015
- David Blei, Rajesh Ranganath, Shakir Mohamed. *Variational Inference: Foundations and Modern Methods*. NeurIPS 2016
- Mohammad Emtiyaz Khan. *Deep Learning with Bayesian Principles*. NeurIPS 2019
- Andrew G. Wilson. *Bayesian Deep Learning and a Probabilistic Perspective of Model Construction*. ICML 2020
- Marc Deisenroth, Cheng Soon Ong. *There and Back Again: A Tale of Slopes and Expectations*. NeurIPS 2020
- Dustin Tran, Balaji Lakshminarayanan, Jasper Snoek. *Practical Uncertainty Estimation and Out-of-Distribution Robustness in Deep Learning*. NeurIPS 2020

Books

- Christopher Bishop. *Patter Recognition and Machine Learning*
- Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*
- Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*