

# Sorbonne Université, Computer Science Master Données, Apprentissage et Connaissances (DAC) Bayesian Deep Learning

**Nicolas Thome**  
Sorbonne Université  
ISIR Lab - Machine Learning Team (MLIA)



# Outline

Beyond Bayesian Linear Regression

Bayesian Logistic Regression

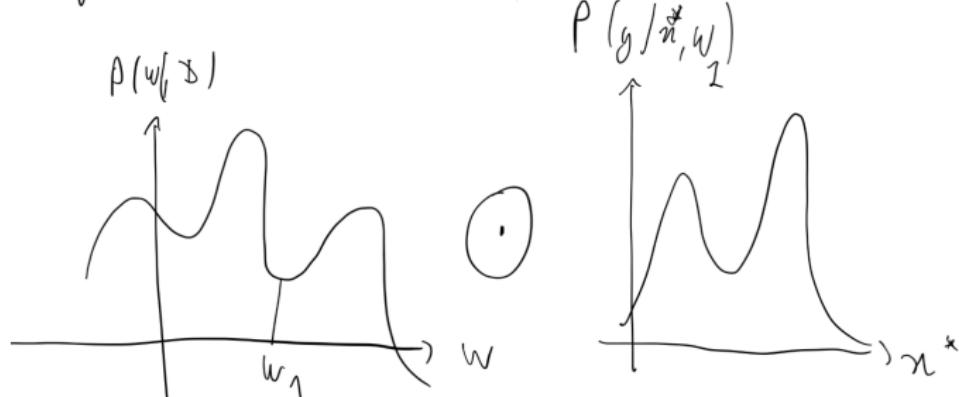
Bayesian Neural Networks

Monte Carlo Dropout

# Beyond Bayesian Linear Regression

- Posterior distribution for parameters  $\mathbf{w}$ :  $p(\mathbf{w}|\mathbf{X}, \mathbf{Y}) \propto p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})$
- Predictive distribution  $p(y|\mathbf{x}^*, \mathcal{D}) = \int p(y|\mathbf{x}^*, \mathbf{w})p(\mathbf{w}|\mathcal{D})d\mathbf{w}$ ,  $(\mathbf{X}, \mathbf{Y}) := \mathcal{D}$

$$p(y, w | \mathbf{x}^*, \mathcal{D}) \propto P(y | \mathbf{x}^*, w) P(w | \mathcal{D})$$



$$p(y | \mathbf{x}^*, \mathcal{D}) = \int p(w | \mathcal{D}) p(y | \mathbf{x}^*, w) dw$$

# Beyond Bayesian Linear Regression

- Posterior distribution for parameters  $\mathbf{w}$ :  $p(\mathbf{w}|\mathbf{X}, \mathbf{Y}) \propto p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})$
- Predictive distribution  $p(y|\mathbf{x}^*, \mathcal{D}) = \int p(y|\mathbf{x}^*, \mathbf{w})p(\mathbf{w}|\mathcal{D})d\mathbf{w}$ ,  $(\mathbf{X}, \mathbf{Y}) := \mathcal{D}$
- **Closed form for posterior  $p(\mathbf{w}|\mathcal{D})$  and predictive distribution  $p(y|\mathbf{x}^*, \mathcal{D})$ : more the exception than the rule!**
- Slightly more complicated models : no closed form solution
  - ▶ Bayesian Logistic Regression
    - ▶ Simplest linear classification model
    - ▶ Likelihood not Gaussian
  - ▶ Neural network with one hidden layer in general
    - ▶ No closed form for regression and classification
  - ▶ And of course deep neural networks

# Approximate Inference

No analytical expression for posterior  $p(\mathbf{w}|\mathcal{D})$  and  $p(y|\mathbf{x}^*, \mathcal{D})$  in general

⇒ Approximation needed!

- Gaussian approximation for  $p(\mathbf{w}|\mathcal{D})$ 
  - ▶ Ex: Laplace approximation [MacKay, 1992]
  - ▶ Historically used for bayesian logistic regression
- Monte Carlo methods: sampling to directly evaluate integral  $p(y|\mathbf{x}^*, \mathcal{D})$ 
  - ▶ Metropolis-Hastings, Hamiltonian Monte Carlo [Neal, 1996], Expectation propagation [Hernandez-Lobato and Adams, 2015, Jyläniemi et al., 2014]
- Variational inference [Hinton and van Camp, 1993, Graves, 2011, Blundell et al., 2015]: convert integration into optimization
  - ▶ Minimize KL divergence between  $p(\mathbf{w}|\mathcal{D})$  and a proposed parametric function

# Outline

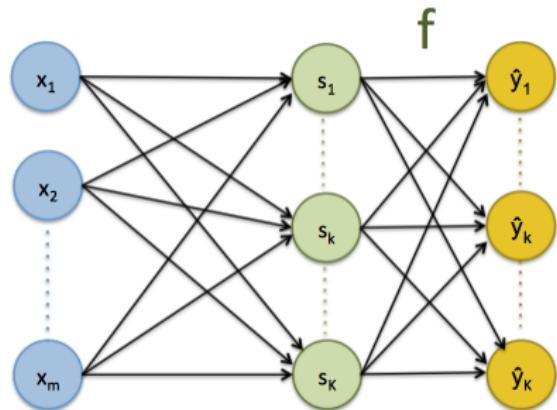
Beyond Bayesian Linear Regression

Bayesian Logistic Regression

Bayesian Neural Networks

Monte Carlo Dropout

# Bayesian Logistic Regression (BLR)



- $s_i = \mathbf{w}x_i$
- Multi-class:  $p(\mathbf{y}_i|\mathbf{x}_i, \mathbf{w}) = \hat{\mathbf{y}}_i$ 
  - ▶  $\hat{y}_{i,k} = \frac{\exp(s_i)}{\sum_k \exp(s_k)}$
- Binary case:  $p(\mathbf{y}_i = 1|\mathbf{x}_i, \mathbf{w}) = \sigma(s_i)$ 
  - ▶ σ sigmoid
  - ▶  $p(\mathbf{y}_i = -1|\mathbf{x}_i, \mathbf{w}) = 1 - \sigma(s_i)$

$$p(\mathbf{w}|\mathbf{X}, \mathbf{Y}) \propto p(\mathbf{Y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})$$

- $p(\mathbf{Y}|\mathbf{X}, \mathbf{w}) = \prod_{i=1}^N p(\mathbf{y}_i = 1|\mathbf{x}_i, \mathbf{w})$  not Gaussian anymore!
- ⇒ no closed-form on posterior distribution  $p(\mathbf{w}|\mathbf{X}, \mathbf{Y})$ !

# Bayesian Logistic Regression training (MAP)

$$\begin{aligned}\mathbf{w}_{\text{MAP}} &= \arg \max_{\mathbf{w}} p(\mathbf{Y}|\mathbf{X}, \mathbf{w}) p(\mathbf{w}) = \arg \max_{\mathbf{w}} \prod_{n=1}^N p(y_n|\mathbf{x}_n, \mathbf{w}) p(\mathbf{w}) \\ &= \arg \min_{\mathbf{w}} \sum_{n=1}^N -\log(p(y_n|\mathbf{x}_n, \mathbf{w})) - \log(p(\mathbf{w}))\end{aligned}$$

- Gaussian prior:  $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \mathbf{0}, \sigma_0^2 \mathbf{I})$  ;
- MAP BLR training with binary prediction:

$$\mathbf{w}_{\text{MAP}} = \arg \min_{\mathbf{w}} \sum_{n=1}^N \left( -y_n \log \sigma(\mathbf{w}^T \mathbf{x}_n + b) - (1-y_n) \log(1-\sigma(\mathbf{w}^T \mathbf{x}_n + b)) \right) + \frac{1}{2\sigma_0^2} \|\mathbf{w}\|_2^2$$

- Again: Gaussian prior  $\Leftrightarrow$  weight decay

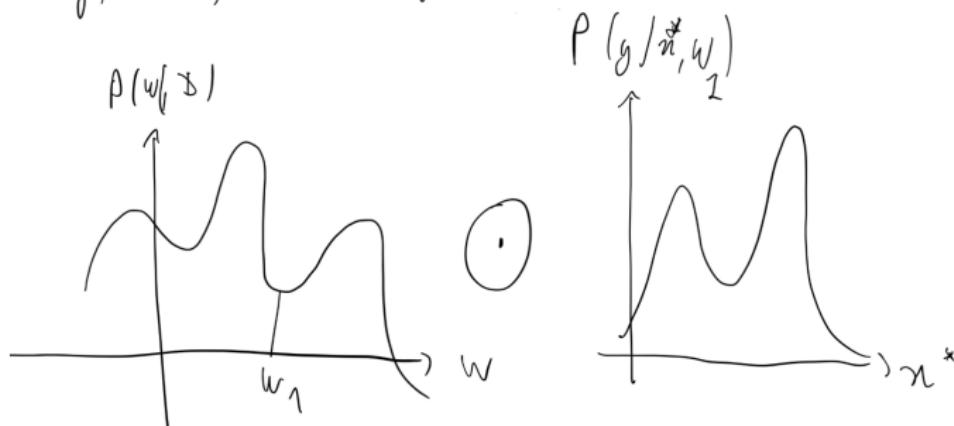
# Bayesian Logistic Regression training (MAP)

$$\mathbf{w}_{\text{MAP}} = \arg \min_{\mathbf{w}} \sum_{n=1}^N (-y_n \log \sigma(\mathbf{w}^T \mathbf{x}_n + b) - (1-y_n) \log(1-\sigma(\mathbf{w}^T \mathbf{x}_n + b))) + \frac{1}{2\sigma_0^2} \|\mathbf{w}\|_2^2$$

- $\mathbf{w}_{\text{MAP}}$  with gradient descent
- Recap: we want to estimate predictive distribution:

$$p(y=1|\mathbf{x}^*, \mathcal{D}) = \int p(y=1|\mathbf{x}^*, \mathbf{w}) p(\mathbf{w}|\mathcal{D}) d\mathbf{w}$$

$$p(y, w | \mathcal{D}) \propto p(y | \mathcal{D}, w) p(w | \mathcal{D})$$



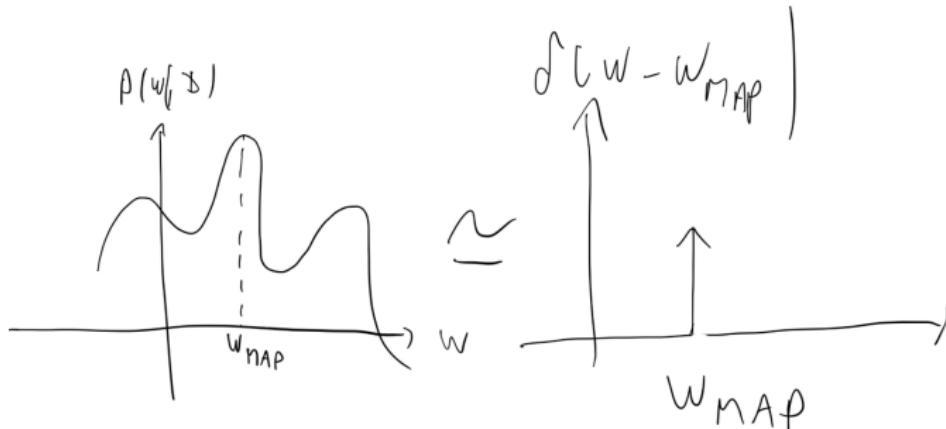
# Bayesian Logistic Regression training (MAP)

$$\mathbf{w}_{\text{MAP}} = \arg \min_{\mathbf{w}} \sum_{n=1}^N \left( -y_n \log \sigma(\mathbf{w}^T \mathbf{x}_n + b) - (1-y_n) \log(1-\sigma(\mathbf{w}^T \mathbf{x}_n + b)) + \frac{1}{2\sigma_0^2} \|\mathbf{w}\|_2^2 \right)$$

- $\mathbf{w}_{\text{MAP}}$  with gradient descent
- Recap: we want to estimate predictive distribution:

$$p(y = 1 | \mathbf{x}^*, \mathcal{D}) = \int p(y = 1 | \mathbf{x}^*, \mathbf{w}) p(\mathbf{w} | \mathcal{D}) d\mathbf{w}$$

- ▶ Need full posterior distribution  $p(\mathbf{w} | \mathbf{X}, \mathbf{Y})$ , but posterior intractable
- ▶  $p(\mathbf{w} | \mathbf{X}, \mathbf{Y}) \approx \delta(\mathbf{w} - \mathbf{w}_{\text{MAP}}) \Rightarrow p(y = 1 | \mathbf{x}^*, \mathcal{D}) \approx p(y = 1 | \mathbf{x}^*, \mathbf{w}_{\text{MAP}})$



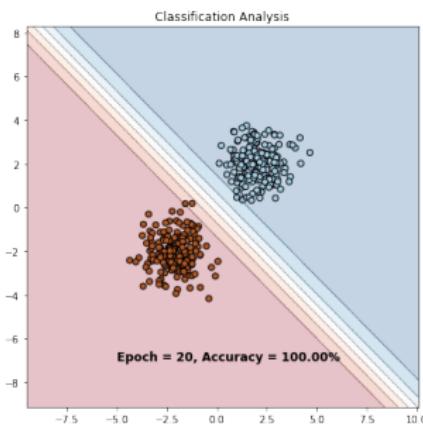
# Bayesian Logistic Regression training (MAP)

$$\mathbf{w}_{\text{MAP}} = \arg \min_{\mathbf{w}} \sum_{n=1}^N \left( -y_n \log \sigma(\mathbf{w}^T \mathbf{x}_n + b) - (1-y_n) \log(1-\sigma(\mathbf{w}^T \mathbf{x}_n + b)) + \frac{1}{2\sigma_0^2} \|\mathbf{w}\|_2^2 \right)$$

- $\mathbf{w}_{\text{MAP}}$  with gradient descent
- Recap: we want to estimate predictive distribution:

$$p(y = 1 | \mathbf{x}^*, \mathcal{D}) = \int p(y = 1 | \mathbf{x}^*, \mathbf{w}) p(\mathbf{w} | \mathcal{D}) d\mathbf{w}$$

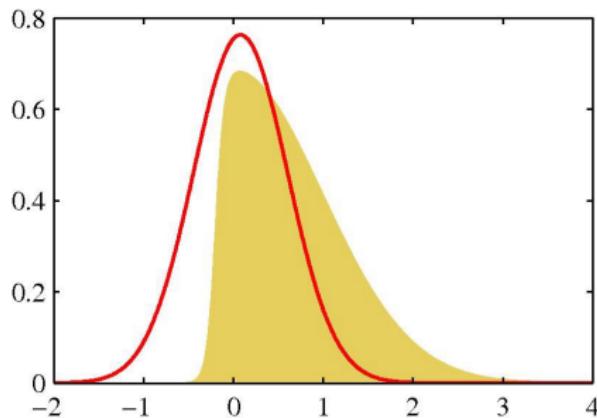
- ▶ Need full posterior distribution  $p(\mathbf{w} | \mathbf{X}, \mathbf{Y})$ , but posterior intractable
- ▶  $p(\mathbf{w} | \mathbf{X}, \mathbf{Y}) \approx \delta(\mathbf{w} - \mathbf{w}_{\text{MAP}}) \Rightarrow p(y = 1 | \mathbf{x}^*, \mathcal{D}) \approx p(y = 1 | \mathbf{x}^*, \mathbf{w}_{\text{MAP}})$



- $p(\mathbf{w} | \mathbf{X}, \mathbf{Y}) \approx \delta(\mathbf{w} - \mathbf{w}_{\text{MAP}})$ : very coarse approximation:
- Uncertainty does not increase far from training data
- Need for more accurate approximations

# Laplace Approximation for $p(\mathbf{w}|\mathbf{X}, \mathbf{Y})$

- Approximate  $p(\mathbf{w}|\mathbf{X}, \mathbf{Y})$  by a normal distribution  $q(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$
- Fit the mean  $\boldsymbol{\mu}$  of  $q(\mathbf{w})$  to the mode of  $p(\mathbf{w}|\mathbf{X}, \mathbf{Y})$ 
  - ▶ Mode of  $p(\mathbf{w}) \Rightarrow \nabla_{\mathbf{w}} p(\mathbf{w}) = 0$
  - ▶ In practice, maximize log posterior (e.g. gradient ascent)  $\Rightarrow$  MAP:  $\boldsymbol{\mu} = \mathbf{w}_{MAP}$
- Fit the inverse covariance  $\boldsymbol{\Sigma}^{-1}$  of  $q(\mathbf{w})$  to the Hessian of  $p(\mathbf{w}|\mathbf{X}, \mathbf{Y})$  at  $\boldsymbol{\mu} = \mathbf{w}_{MAP}$ :  $\boldsymbol{\Sigma}^{-1} = \nabla \nabla_{\mathbf{w}} p(\mathbf{w}|\mathbf{X}, \mathbf{Y})|_{\mathbf{w}=\mathbf{w}_{MAP}}$



from [Bishop, 2006]

- Laplace limitation: approximation at a single value of  $p(\mathbf{w}|\mathbf{X}, \mathbf{Y})$ , ignores global properties

# Predictive Distribution $p(y|\mathbf{x}^*, \mathcal{D})$ for BLR

RECAP, with  $\mathcal{D} = \mathbf{X}, \mathbf{Y}$ :

$$p(y|\mathbf{x}^*, \mathcal{D}) = \int p(y|\mathbf{x}^*, \mathbf{w})p(\mathbf{w}|\mathcal{D})d\mathbf{w}$$

- Posterior approximation by normal  $p(\mathbf{w}|\mathcal{D}) \approx q(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ :

$$p(y|\mathbf{x}^*, \mathcal{D}) \approx \int p(y|\mathbf{x}^*, \mathbf{w})q(\mathbf{w})d\mathbf{w}$$

- However, likelihood  $p(y|\mathbf{x}^*, \mathbf{w})$  still not Gaussian  
⇒ **Intractable posterior distribution**  $p(y|\mathbf{x}^*, \mathcal{D})$ !

# Predictive Distribution $p(y|x^*, \mathcal{D})$ for BLR

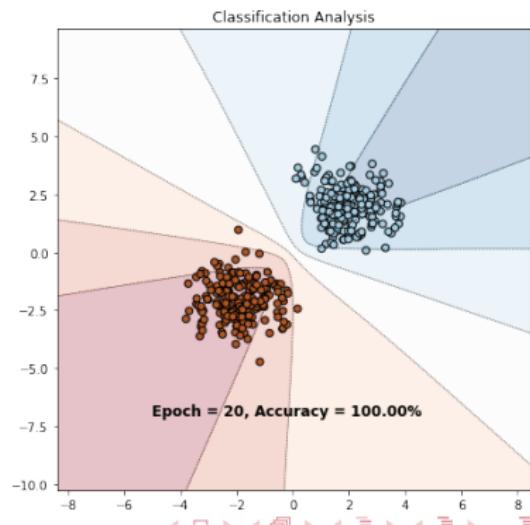
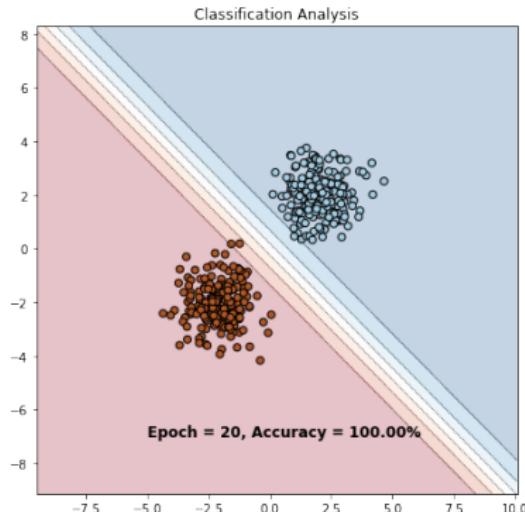
- Option 1: use Monte Carlo (MC) sampling

- ▶ Binary case (simple x-class extension):  $p(y = 1|x^*, \mathbf{w}) = \sigma(\mathbf{w}^T \mathbf{x}^*)$ ,  $\sigma$  sigmoid

$$p(y = 1|x^*, \mathcal{D}) \approx \sum_{s=1}^S \sigma\left((\mathbf{w}^s)^T \mathbf{x}^*\right) \quad \mathbf{w}^s \sim q(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- ▶ Easy to sample from Gaussian  $q(\mathbf{w})$

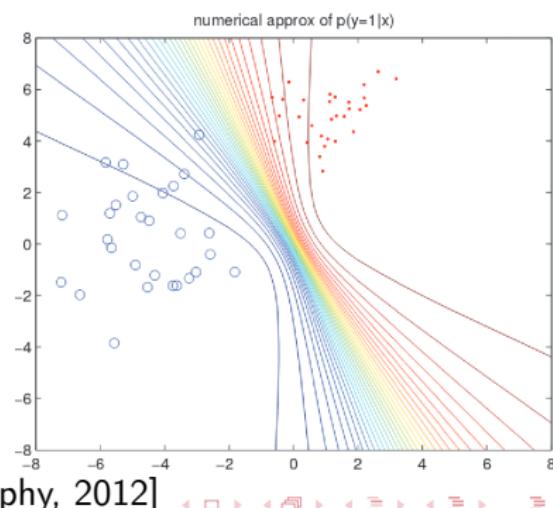
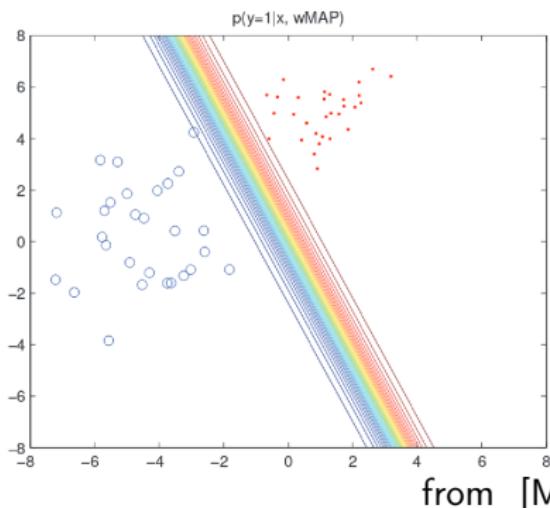
- **Practical session:** MAP solution for LR vs BLR (Laplace & MC sampling)



# Predictive Distribution $p(y|\mathbf{x}^*, \mathcal{D})$ for BLR

- $p(y|\mathbf{x}^*, \mathcal{D}) \approx \int p(y|\mathbf{x}^*, \mathbf{w})q(\mathbf{w})d\mathbf{w}$  intractable
- **Option 2:** (binary case):  $p(y|\mathbf{x}^*, \mathcal{D}) \approx \int \sigma(\mathbf{w}^T \mathbf{x}^*)q(\mathbf{w})d\mathbf{w}$ ;  $q(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$
- Convolution of sigmoid with Gaussian still intractable
  - ▶ Approximate  $\sigma(\mathbf{w}^T \mathbf{x}^*)$  by probit:  $\sigma(a) \approx \Phi(\lambda a)$ ,  $\lambda^2 = \pi/8$
  - ▶ Convolution of probit with Gaussian  $\Rightarrow$  probit:

$$p(y|\mathbf{x}^*, \mathcal{D}) \approx \int \Phi(\lambda \mathbf{w}^T \mathbf{x}^*) \mathcal{N}(\mathbf{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) d\mathbf{w} = \Phi\left(\frac{\mu_f}{\sqrt{\lambda^{-2} + \sigma_f^2}}\right) \quad \mu_f = \boldsymbol{\mu}^T \mathbf{x}^* \quad \sigma_f^2 = \mathbf{x}^{*T} \boldsymbol{\Sigma} \mathbf{x}^*$$



# Outline

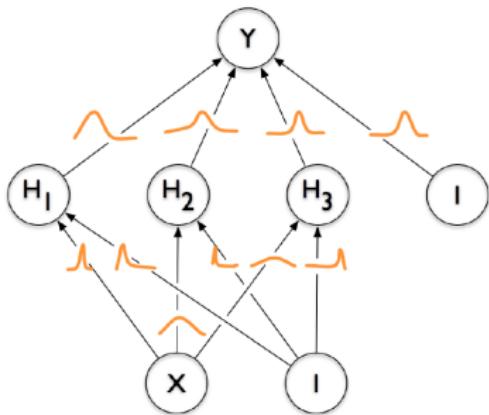
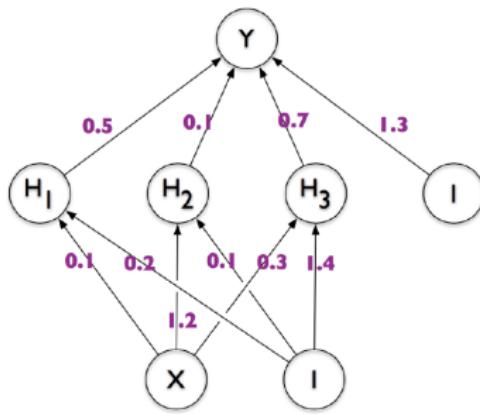
Beyond Bayesian Linear Regression

Bayesian Logistic Regression

Bayesian Neural Networks

Monte Carlo Dropout

# Bayesian Neural Networks (BNN)



Credit: [Blundell et al., 2015]

- Standard NN:  $\mathbf{y}_i = f^{\mathbf{w}}(\mathbf{x}_i)$ , Bayesian NN:  $p(\mathbf{y}_i|\mathbf{x}_i, \mathcal{D})$ 
  - ▶ Define prior over weights, e.g.  $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|0, \alpha^{-1}\mathcal{I})$  (point estimate for bias)
    - ▶ In practice, typically separate variance  $\sigma^2 = \alpha^{-1}$  for each layer
  - ▶ Define likelihood,  $p(\mathbf{y}_i|\mathbf{x}_i, \mathbf{w})$ , e.g. for regression  $p(\mathbf{y}_i|\mathbf{x}_i, \mathbf{w}) = \mathcal{N}(\mathbf{y}_i; f^{\mathbf{w}}(\mathbf{x}_i), \beta^{-1})$
  - ▶ **Goal:** compute posterior  $p(\mathbf{w}|\mathbf{X}, \mathbf{Y}) = \prod_{i=1}^N p(\mathbf{w}|\mathbf{x}_i, \mathbf{y}_i, \beta) \propto p(\mathbf{w}) \prod_{i=1}^N p(\mathbf{y}_i|\mathbf{x}_i, \mathbf{w})$

# Bayesian Neural Networks (BNN)

$$p(\mathbf{w}|\mathbf{X}, \mathbf{Y}) \propto p(\mathbf{w}) \prod_{i=1}^N p(y_i|x_i, \mathbf{w})$$

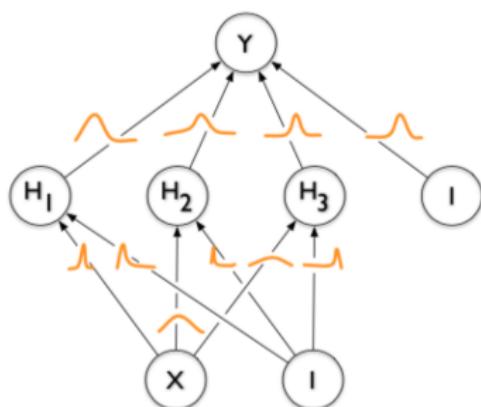
- With Bayesian Neural networks, even with:

- Gaussian prior  $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathcal{I})$
- Gaussian likelihood, e.g. regression  $p(y_i|x_i, \mathbf{w}) = \mathcal{N}(y_i; f^{\mathbf{w}}(x_i), \beta^{-1})$
- Posterior  $p(\mathbf{w}|\mathbf{X}, \mathbf{Y}, \beta) \propto p(\mathbf{w}) \prod_{i=1}^N p(y_i|x_i, \mathbf{w})$  is NOT Gaussian!!

- Non-linear dependence of  $f^{\mathbf{w}}(\mathbf{x})$  on  $\mathbf{w}$ !

- RECAP:

- $p(x) = \mathcal{N}(x|\mu_x, \Sigma_x)$
- $p(y|x) = \mathcal{N}(y|Ax + b, \Sigma_y)$ 
  - Linear dependence  $Ax + b$  required
- Then:  $p(x|y) = \mathcal{N}(x|\mu_{x|y}, \Sigma_{x|y})$ 
  - Not true for BNNs!



Credit: [Blundell et al., 2015]

# Posterior Inference: MCMC Carlo Sampling

The true predictive distribution  $p(y|\mathbf{x}^*, \mathcal{D})$  cannot be evaluated analytically

$$p(y|\mathbf{x}^*, \mathcal{D}) = \int p(y|\mathbf{x}^*, \mathbf{w})p(\mathbf{w}|\mathcal{D})d\mathbf{w}$$

- Monte Carlo estimation of the integral:

$$p(y|\mathbf{x}^*, \mathcal{D}) \approx \frac{1}{S} \sum_{s=1}^S p(y|\mathbf{x}^*, \mathbf{w}^s) \quad \mathbf{w}^s \sim p(\mathbf{w}|\mathcal{D})$$

- Can't sample exactly from  $p(\mathbf{w}|\mathcal{D})$ , **BUT approximate sampling using Markov chain Monte Carlo (MCMC) possible !**
  - ▶ Metropolis-Hastings (MH), Hamiltonian Monte Carlo (HMC) [Neal, 1996]
- **Works well, accurate posterior inference in BNNs**
- **Main drawback: does not scale to large datasets**
  - ▶ Computing likelihood for MH/HMC acceptance step requires the whole dataset

# Variational Inference (VI)

The true posterior  $p(\mathbf{w}|\mathbf{X}, \mathbf{Y})$  cannot usually be evaluated analytically

- Defining an **approximating variational distribution**  $q_\theta(\mathbf{w})$ , parameterized by  $\theta$
- Minimizing its KL divergence with the true posterior:**

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

- Computing approximate predictive distribution:**  $p(\mathbf{y}|\mathbf{x}^*, \mathbf{X}, \mathbf{Y}) \Leftarrow q_{\theta^*}(\mathbf{w})$ :

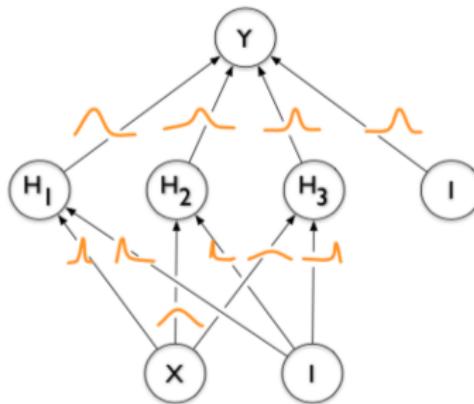
$$p(\mathbf{y}|\mathbf{x}^*, \mathbf{X}, \mathbf{Y}) \approx \int p(\mathbf{y}|\mathbf{x}^*, \mathbf{w}) q_{\theta^*}(\mathbf{w}) d\mathbf{w}$$

# Variational Inference (VI)

- Recap: BNN prior, e.g. Gaussian:  $p(\mathbf{w}) = \mathcal{N}(\mathbf{w} | 0, \alpha^{-1} \mathcal{I})$
- **Variational approximate posterior**  $q_{\theta}(\mathbf{w})$ , e.g. fully factorized Gaussian:

$$q_{\theta}(\mathbf{w}) = \mathcal{N}(\mathbf{w} | \theta) = \mathcal{N}(\mathbf{w} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{j=1}^D \mathcal{N}(w_j | \mu_j, \sigma_j)$$

- Each weight of the network  $w_j$  has its own mean  $\mu_j$  and variance  $\sigma_j$ 
  - ▶  $\theta = \{(\mu_j, \sigma_j)\}_{j \in \{1; D\}}$ : **variational parameters**



Credit: [Blundell et al., 2015]

## Variational Inference (VI): ELBO

$$\begin{aligned} KL(q_{\theta}(\mathbf{w}) \| p(\mathbf{w} | \mathbf{X}, \mathbf{Y})) &= \int q_{\theta}(\mathbf{w}) \log \frac{q_{\theta}(\mathbf{w})}{p(\mathbf{w} | \mathbf{X}, \mathbf{Y})} d\mathbf{w} = - \int q_{\theta}(\mathbf{w}) \log \frac{p(\mathbf{w} | \mathbf{X}, \mathbf{Y})}{q_{\theta}(\mathbf{w})} d\mathbf{w} \\ &= - \int q_{\theta}(\mathbf{w}) \log \frac{p(\mathbf{Y} | \mathbf{X}, \mathbf{w}) p(\mathbf{w})}{q_{\theta}(\mathbf{w}) p(\mathbf{Y} | \mathbf{X})} d\mathbf{w} \\ &= - \int q_{\theta}(\mathbf{w}) \log p(\mathbf{Y} | \mathbf{X}, \mathbf{w}) d\mathbf{w} + \int q_{\theta}(\mathbf{w}) \log \frac{q_{\theta}(\mathbf{w})}{p(\mathbf{w})} + \log p(\mathbf{Y} | \mathbf{X}) \\ &= - \int q_{\theta}(\mathbf{w}) \log p(\mathbf{Y} | \mathbf{X}, \mathbf{w}) d\mathbf{w} + KL(q_{\theta}(\mathbf{w}) \| p(\mathbf{w})) + \log p(\mathbf{Y} | \mathbf{X}) \end{aligned}$$

- $\Rightarrow KL(q_{\theta}(\mathbf{w}) \| p(\mathbf{w} | \mathbf{X}, \mathbf{Y})) = -\mathcal{L}_{VI}(\mathbf{X}, \mathbf{Y}, \theta) + \log p(\mathbf{Y} | \mathbf{X})$

►  $\mathcal{L}_{VI}(\mathbf{X}, \mathbf{Y}, \theta)$ : Evidence Lower Bound (ELBO)

$$\mathcal{L}_{VI}(\theta) = \int q_{\theta}(\mathbf{w}) \log p(\mathbf{Y} | \mathbf{X}, \mathbf{w}) d\mathbf{w} - KL(q_{\theta}(\mathbf{w}) \| p(\mathbf{w}))$$

►  $\mathcal{L}_{VI}(\theta) = \log p(\mathbf{Y} | \mathbf{X}) - KL(q_{\theta}(\mathbf{w}) \| p(\mathbf{w} | \mathbf{X}, \mathbf{Y})) \leq \log p(\mathbf{Y} | \mathbf{X})$

---

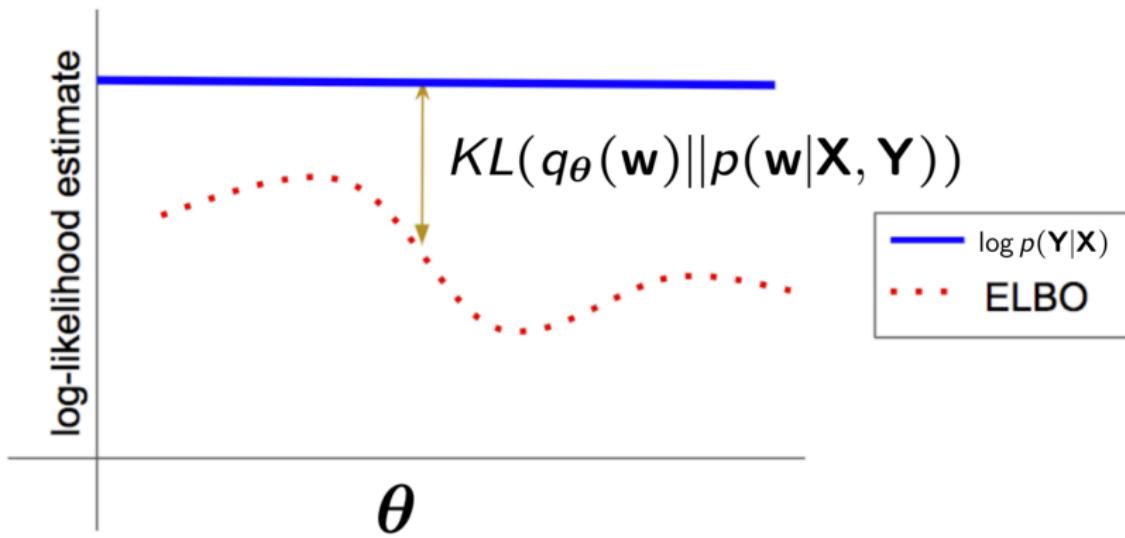
$$p(\mathbf{w} | \mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{Y} | \mathbf{X}, \mathbf{w}) p(\mathbf{w})}{p(\mathbf{Y} | \mathbf{X})}$$

# ELBO Illustration

- $\mathcal{L}_{VI}(\mathbf{X}, \mathbf{Y}, \theta)$ : Evidence Lower Bound (ELBO)

$$\mathcal{L}_{VI}(\theta) = \int q_{\theta}(\mathbf{w}) \log p(\mathbf{Y}|\mathbf{X}, \mathbf{w}) d\mathbf{w} - KL(q_{\theta}(\mathbf{w})||p(\mathbf{w}|\mathbf{X}, \mathbf{Y}))$$

- $\mathcal{L}_{VI}(\theta) = \log p(\mathbf{Y}|\mathbf{X}) - KL(q_{\theta}(\mathbf{w})||p(\mathbf{w}|\mathbf{X}, \mathbf{Y})) \leq \log p(\mathbf{Y}|\mathbf{X})$

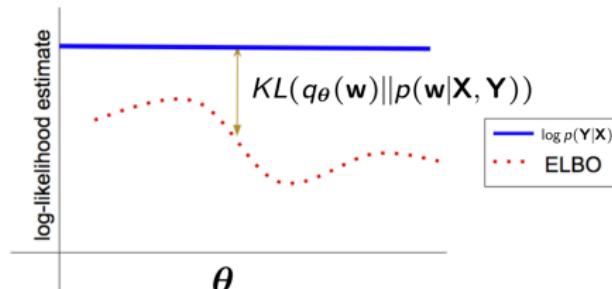


# Variational Inference (VI): ELBO

- $\mathcal{L}_{VI}(\theta) = \log p(\mathbf{Y}|\mathbf{X}) - KL(q_{\theta}(\mathbf{w})||p(\mathbf{w}|\mathbf{X}, \mathbf{Y}))$
- $\Rightarrow$  Minimizing  $KL(q_{\theta}(\mathbf{w})||p(\mathbf{w}|\mathbf{X}, \mathbf{Y})) \Leftrightarrow$  maximizing  $\mathcal{L}_{VI}(\theta)$  w.r.t  $q_{\theta}(\mathbf{w})$ :

$$\begin{aligned}\mathcal{L}_{VI}(\theta) &= \int q_{\theta}(\mathbf{w}) \log p(\mathbf{Y}|\mathbf{X}, \mathbf{w}) d\mathbf{w} - KL(q_{\theta}(\mathbf{w})||p(\mathbf{w})) \leq \log p(\mathbf{Y}|\mathbf{X}) \\ &= \mathbb{E}_{q_{\theta}(\mathbf{w})}[\log p(\mathbf{Y}|\mathbf{X}, \mathbf{w})] - KL(q_{\theta}(\mathbf{w})||p(\mathbf{w})) \\ &= \sum_{i=1}^N \int q_{\theta}(\mathbf{w}) \log p(\mathbf{y}_i|f^{\mathbf{w}}(\mathbf{x}_i)) d\mathbf{w} - KL(q_{\theta}(\mathbf{w})||p(\mathbf{w}))\end{aligned}$$

- **Exp. log likelihood**  $\mathbb{E}_{q_{\theta}(\mathbf{w})}[\log p(\mathbf{Y}|\mathbf{X}, \mathbf{w})]$ : max  $\Leftrightarrow q_{\theta}(\mathbf{w})$  explain data well
- **Prior KL**  $KL(q_{\theta}(\mathbf{w})||p(\mathbf{w}))$ : min  $\Leftrightarrow q_{\theta}(\mathbf{w})$  as close as possible to  $p(\mathbf{w})$  prior



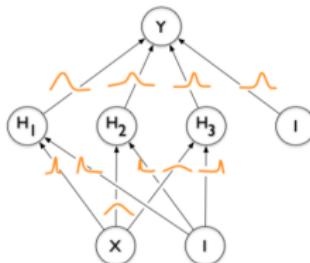
# Variational Inference: Training

- Variational Bayesian NN training: computing derivates of  $\mathcal{L}_{VI}$  w.r.t variational parameters  $\theta$

$$\mathcal{L}_{VI}(\theta) = \sum_{i=1}^N \int q_\theta(\mathbf{w}) \log p(\mathbf{y}_i | f^\mathbf{w}(\mathbf{x}_i)) d\mathbf{w} - KL(q_\theta(\mathbf{w}) || p(\mathbf{w}))$$

- RECAP: approximate variational posterior, e.g. fully factorize Gaussian:

$$q_\theta(\mathbf{w}) = \prod_{i=1}^D \mathcal{N}(w_i | \mu_j, \sigma_j)$$



- **Prior KL**  $KL(q_\theta(\mathbf{w}) || p(\mathbf{w}))$ : often can be integrated analytically, e.g. with Gaussian functions for prior  $p(\mathbf{w})$  and posterior approximation  $q_\theta(\mathbf{w})$
- **Expected log likelihood**  $\mathbb{E}_{q_\theta(\mathbf{w})}[\log p(\mathbf{Y} | \mathbf{X}, \mathbf{w})]$ : no close-form solution in general, requires tractable calculations over the entire dataset  
⇒ estimation by sampling

# Stochastic Variational Inference (VI): Training

$$\mathcal{L}_{VI}(\theta) = \sum_{i=1}^N \int q_\theta(\mathbf{w}) \log p(\mathbf{y}_i | f^\mathbf{w}(\mathbf{x}_i)) d\mathbf{w} - KL(q_\theta(\mathbf{w}) || p(\mathbf{w}))$$

- **Scalable gradient computation: batch sampling** [Graves, 2011]
  - ▶  $\mathcal{L}_{VI}$  linearly decomposes into training examples, unbiased gradient estimator
- **Modern solutions: approximate integral with MC integration**  $\hat{\mathbf{w}}_i \sim q_\theta(\mathbf{w})$ 
  - ▶ Sample  $\log p(\mathbf{y}_i | f^{\hat{\mathbf{w}}_i}(\mathbf{x}_i))$ ,  $\hat{\mathbf{w}}_i \sim q_\theta(\mathbf{w})$

$$\mathcal{L}_{VI}(\theta) = \sum_{i \in S} \log p(\mathbf{y}_i | f^{\hat{\mathbf{w}}_i}(\mathbf{x}_i)) - KL(q_\theta(\mathbf{w}) || p(\mathbf{w}))$$

- Issue: computing gradient wrt variational parameters  $\frac{\partial}{\partial \theta} \log p(\mathbf{y}_i | f^{\hat{\mathbf{w}}_i}(\mathbf{x}_i))$
- **Problem: sampling  $\hat{\mathbf{w}}_i \sim q_\theta(\mathbf{w})$  depends on variational parameters  $\theta$** 
  - ▶ Solution, easy cases: re-parametrization  $\mathbf{w} = g(\theta, \epsilon)$ 
    - ▶ Where  $g$  deterministic and  $\epsilon$  independent of  $\theta$  - As in VAE [Kingma and Welling, 2014]
    - ▶ Crucial point: sampling fully in  $\epsilon$ , independent of  $\theta$
  - ▶ Gaussian ex:  $\theta = \{\theta_j\}$ ,  $\theta_j = (\mu_j, \sigma_j)$ :  $\hat{\mathbf{w}}_j \sim \mathcal{N}(w_j | \mu_j, \sigma_j)$ 
    - ▶  $w_j = g((\mu_j, \sigma_j), \epsilon_j) = \mu_j + \sigma_j \epsilon_j$ :  $\hat{\mathbf{w}}_j \sim \mathcal{N}(w_j | \mu_j, \sigma_j) \Leftrightarrow \hat{\epsilon}_j \sim \mathcal{N}(\epsilon_j | 0, 1)$

$$\mathcal{L}_{VI}(\theta) = \sum_{i \in S} \log p(\mathbf{y}_i | f^{g(\theta, \hat{\epsilon}_i)}(\mathbf{x}_i)) - KL(q_\theta(\mathbf{w}) || p(\mathbf{w}))$$

# Stochastic Variational Inference (VI): Training

---

**Algorithm 1** Minimise divergence between  $q_\theta(\omega)$  and  $p(\omega|X, Y)$

---

- 1: Given dataset  $\mathbf{X}, \mathbf{Y}$ ,
  - 2: Define learning rate schedule  $\eta$ ,
  - 3: Initialise parameters  $\theta$  randomly.
  - 4: **repeat**
  - 5:   Sample  $M$  random variables  $\hat{\epsilon}_i \sim p(\epsilon)$ ,  $S$  a random subset of  $\{1, \dots, N\}$  of size  $M$ .
  - 6:   Calculate stochastic derivative estimator w.r.t.  $\theta$ :
$$\widehat{\Delta\theta} \leftarrow -\frac{N}{M} \sum_{i \in S} \frac{\partial}{\partial\theta} \log p(\mathbf{y}_i | \mathbf{f}^{g(\theta, \hat{\epsilon}_i)}(\mathbf{x}_i)) + \frac{\partial}{\partial\theta} \text{KL}(q_\theta(\omega) || p(\omega)).$$
  - 7:   Update  $\theta$ :
$$\theta \leftarrow \theta + \eta \widehat{\Delta\theta}.$$
  - 8: **until**  $\theta$  has converged.
-

# Bayesian Neural Networks: Predictive Distribution

- Gaussian approximation of posterior  $q_\theta(\mathbf{w}) \approx p(\mathbf{w}|\mathbf{X}, \mathbf{Y})$  e.g. VI or Laplace
- Predictive distribution:  $p(\mathbf{y}|\mathbf{x}^*, \mathcal{D}) \approx \int p(\mathbf{y}|\mathbf{x}^*, \mathbf{w}) q_\theta(\mathbf{w}) d\mathbf{w}$
- **Even with  $q_\theta(\mathbf{w})$  Gaussian, no closed-form for  $p(\mathbf{y}|\mathbf{x}^*, \mathcal{D})$ !**
  - ▶ Again due to non-linear dependence of  $y(\mathbf{x}^*, \mathbf{w})$  wrt  $\mathbf{w}$

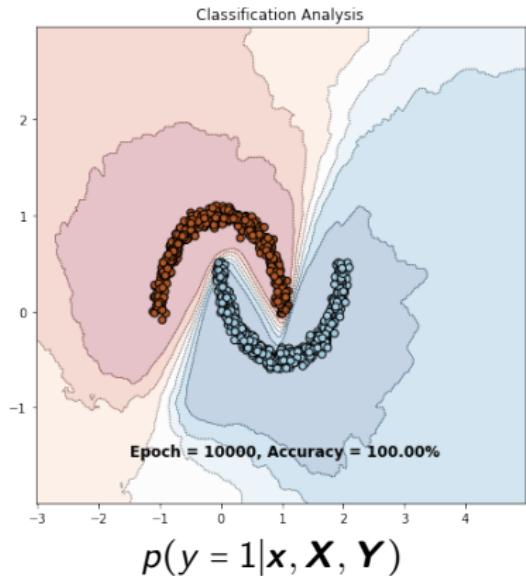
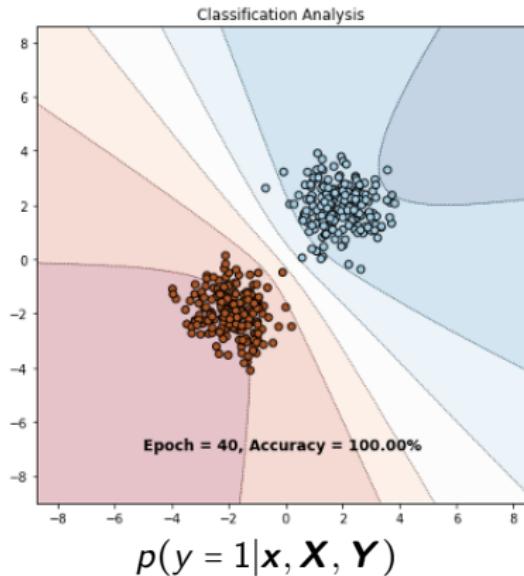
## Solutions:

1. MC sampling, easy to sample from  $q_\theta(\mathbf{w})$
  2. Perform Taylor expansion of  $y(\mathbf{x}^*, \mathbf{w})$  around  $\mathbf{w}_{MAP}$  for regression<sup>a</sup>:  
$$y(\mathbf{x}, \mathbf{w}) \approx y(\mathbf{x}, \mathbf{w}_{MAP}) + \frac{\partial y}{\partial \mathbf{w}}|_{\mathbf{w}=\mathbf{w}_{MAP}} (\mathbf{w} - \mathbf{w}_{MAP})$$
- $p(\mathbf{y}|\mathbf{x}^*, \mathbf{w}) \approx \mathcal{N}(y|\mu_y; \beta^{-1})$ ,  $\mu_y = y(\mathbf{x}, \mathbf{w}_{MAP}) + \frac{\partial y}{\partial \mathbf{w}}|_{\mathbf{w}=\mathbf{w}_{MAP}} (\mathbf{w} - \mathbf{w}_{MAP})$
  - Closed form solution for  $p(\mathbf{y}|\mathbf{x}^*, \mathcal{D})$ 
    - ▶  $p(\mathbf{y}|\mathbf{x}^*, \mathcal{D}) = \mathcal{N}(y|y(\mathbf{x}, \mathbf{w}_{MAP}); \sigma^2(x))$
    - ▶  $\sigma^2(x) = \beta^{-1} + \mathbf{g}^T \mathbf{A}^{-1} \mathbf{g}$ ,  $\mathbf{g}$  gradient and  $\mathbf{A}$  Hessian at  $\mathbf{w}_{MAP}$

<sup>a</sup>For classification, logit Taylor expansion, see 5.7.1 in [Bishop, 2006]

# Practical session

- Own implementation of VI for:
  - ▶ Bayesian logistic regression
  - ▶ Neural network for non-linear classification (moons)



# Outline

Beyond Bayesian Linear Regression

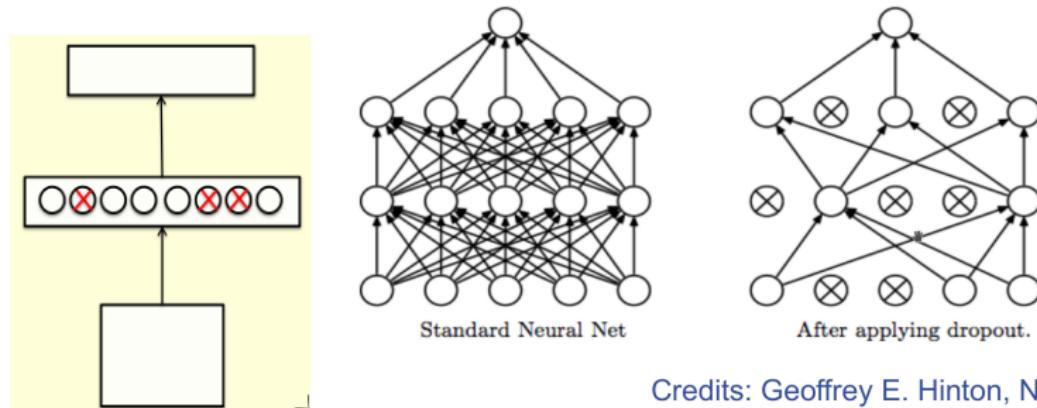
Bayesian Logistic Regression

Bayesian Neural Networks

Monte Carlo Dropout

# Dropout [Hinton et al., 2012]

- Randomly omit each hidden unit with probability  $p$ , e.g.  $p = 0.5$
- **Regularization technique**, limits over-fitting (better generalization)
  - ▶ Prevent co-adaptation
  - ▶ May be viewed as averaging over many NN
  - ▶ Slower convergence



Credits: Geoffrey E. Hinton, NIPS 2012

# Dropout as a variational inference [Gal, 2016]

- Input  $x \in \mathbb{R}^{D_a}$ , latent vector  $h \in \mathbb{R}^L$ 
  - ▶ First layer:  $h = \sigma(xW_1)$ ,  $\sigma$  non-linearity
- **Dropout sampling:** in input :  $x \odot \hat{\varepsilon}$ 
  - ▶  $\hat{\varepsilon} = \{\hat{\varepsilon}_i^1\}_{i \in \{1; D\}}$   $\varepsilon_i \sim \text{Bernoulli}(1 - p)$
  - ▶ First layer:  $h = \sigma((x \odot \hat{\varepsilon})W_1)$ 
    - ▶  $(x \text{ diag}(\hat{\varepsilon}))W_1 = x(\text{diag}(\hat{\varepsilon})W_1) = x\hat{W}_1$
    - ▶ Randomly setting to 0 rows of  $W_1$  (size  $((D, L))$  with probability  $p$

$$\begin{array}{c} \text{Input } x \\ \left( \begin{matrix} x_1 & \dots & x_D \\ 0 & \ddots & 0 \\ 0 & \dots & x_D \end{matrix} \right) \quad (D, 1) \end{array} \quad \left( \begin{matrix} w_1 \\ \vdots \\ w_D \\ (D, L) \end{matrix} \right) \quad \text{Dropout mask } \hat{\varepsilon} = \left( \begin{matrix} \hat{\varepsilon}_1 \\ \hat{\varepsilon}_2 \\ \vdots \\ \hat{\varepsilon}_D \end{matrix} \right) \quad = \quad \left( \begin{matrix} x_1 & \dots & x_D \\ 0 & \ddots & 0 \\ 0 & \dots & x_D \end{matrix} \right) \quad \left( \begin{matrix} w_1 \\ \vdots \\ w_D \\ (D, L) \end{matrix} \right)$$

<sup>a</sup>dimension (1,D)

# Dropout as a variational inference [Gal, 2016]

- **Illustration: dropout for a 2 layer NN** (1 hidden),  $\epsilon_i \sim \text{Bernoulli}(1 - p_i)$ :

$$\begin{aligned} & \boxed{\mathbf{h}_1 = \sigma(\hat{x}\mathbf{W}_1) = \sigma(\mathbf{x}\hat{\mathbf{W}}_1), \quad \hat{\mathbf{W}}_1 = \text{diag}(\hat{\epsilon}_1)\mathbf{W}_1} \\ & \boxed{\hat{\mathbf{y}} =: f^{\hat{\mathbf{W}}_1, \hat{\mathbf{W}}_2}(\mathbf{x}) = \hat{\mathbf{h}}_1\mathbf{W}_2 = \mathbf{h}_1\hat{\mathbf{W}}_2, \quad \hat{\mathbf{W}}_2 = \text{diag}(\hat{\epsilon}_2)\mathbf{W}_2 - \hat{\mathbf{W}} = \{\hat{\mathbf{W}}_1; \hat{\mathbf{W}}_2\}} \end{aligned}$$

- **MC Dropout sampling:**  $\frac{1}{S} \sum_{s \in S} p(\mathbf{y}_i | f^{\hat{\mathbf{W}}}(\mathbf{x}_i)) \approx \int p(\mathbf{y} | f^{\mathbf{W}}(\mathbf{x}^*)) q(\mathbf{W}) d\mathbf{w}$ 
  - ▶ ∀ layer  $l \in \{1; L\}$ ,  $\mathbf{W}_l$  random variable:  $\mathbf{W}_l \sim q(\mathbf{W}_l) = g(\mathbf{M}_l, \boldsymbol{\varepsilon}_l) = \text{diag}(\boldsymbol{\varepsilon}_l)\mathbf{M}_l$ 
    - ▶  $\boldsymbol{\varepsilon}_{l,i} \sim \text{Bernoulli}(1 - p_l)$ ,  $\mathbf{M}_l$  deterministic parameters

$$\qquad \qquad \qquad \boxed{q(\mathbf{W}) = \prod_{l=1}^L q(\mathbf{W}_l)}$$

- **Big result** (see next): **training NN with dropout  $\Leftrightarrow$  training BNN with variational posterior approximation  $q_M(\mathbf{W})$**  (and some prior  $p(\mathbf{W})$ )
  - ▶  $M = \{\mathbf{M}_l\}_{l \in \{1; L\}}$  variational parameters

- **MC dropout:** sampling several passes with dropout  $\Leftrightarrow$  **performing MC approximate inference with variational posterior  $q_M(\mathbf{W})$**

$$\boxed{\frac{1}{S} \sum_{s \in S} p(\mathbf{y}_i | f^{\hat{\mathbf{W}}}(\mathbf{x}_i)) \approx \int p(\mathbf{y} | f^{\mathbf{W}}(\mathbf{x}^*)) q(\mathbf{W}) d\mathbf{w} \approx p(y | \mathbf{x}^*, \mathbf{X}, \mathbf{Y})}$$

# Dropout as a variational inference [Gal, 2016]

Big result: proof sketch for a 2 layer NN (regression)

- **Prediction with dropout:**  $\hat{y} = \sigma(\mathbf{x} \hat{\mathbf{W}}_1) \hat{\mathbf{W}}_2 =: f^{\hat{\mathbf{W}}_1, \hat{\mathbf{W}}_2}(\mathbf{x})$ 
  - ▶  $\forall I \in \{1; 2\}$  :  $\hat{\mathbf{W}}_I = \text{diag}(\hat{\epsilon}_I) \mathbf{M}_I$  -  $\hat{\mathbf{W}} = \{\hat{\mathbf{W}}_1; \hat{\mathbf{W}}_2\}$
- **Training for regression,**  $\hat{\mathcal{L}}_{\text{dropout}}$  objective function:

$$\hat{\mathcal{L}}_{\text{dropout}}(\mathbf{M}_1, \mathbf{M}_2) = \frac{1}{M} \sum_{i \in S} \|f^{\hat{\mathbf{W}}}(\mathbf{x}_i) - y_i\|^2 + \lambda_1 \|\mathbf{M}_1\|^2 + \lambda_2 \|\mathbf{M}_2\|^2$$

- With Gaussian likelihood:  $p(y_i | f^{\hat{\mathbf{W}}}(\mathbf{x}_i)) = \mathcal{N}(y_i, f^{\hat{\mathbf{W}}}(\mathbf{x}), \tau^{-1}\mathbf{I})$ , we have:

$$\|f^{\hat{\mathbf{W}}}(\mathbf{x}_i) - y_i\|^2 = -\frac{1}{\tau} \log p(y_i | f^{g(\mathbf{M}, \hat{\epsilon}^i)}(\mathbf{x}))$$

- $\hat{\mathcal{L}}_{\text{dropout}}$  rewrites as follows:

$$\boxed{\hat{\mathcal{L}}_{\text{dropout}}(\mathbf{M}_1, \mathbf{M}_2) = \frac{1}{M\tau} \sum_{i \in S} \log p(y_i | f^{g(\mathbf{M}, \hat{\epsilon}^i)}(\mathbf{x})) + \lambda_1 \|\mathbf{M}_1\|^2 + \lambda_2 \|\mathbf{M}_2\|^2} \quad (1)$$

# Dropout as a variational inference [Gal, 2016]

- Big similarity between  $\hat{\mathcal{L}}_{\text{dropout}}$  in Eq (1) and algo 1!
- Same algorithms if:

$$\frac{\partial}{\partial \mathbf{M}} \text{KL}(q(\mathbf{W}) \| p(\mathbf{W})) = \frac{\partial}{\partial \mathbf{M}} N\tau(\lambda_1 \|\mathbf{M}_1\|^2 + \lambda_2 \|\mathbf{M}_2\|^2)$$

[Gal and Ghahramani, 2016] showed that this can be fulfilled for:

- $p(\mathbf{W}) = \prod_I p(\mathbf{W}_I) = \prod_I \mathcal{MN}(\mathbf{W}_I; 0; I/l_I^2, I)$  (prior factorized over layers)
- $q(\mathbf{W}_I) = \text{diag}(\hat{\varepsilon}_I) \mathbf{M}_I$ ,  $\varepsilon_{I,i} \sim \text{Bernoulli}(1 - p_i)$ ,  $q(\mathbf{W}) = \prod_I q(\mathbf{W}_I)$ 
  - ▶ Approximated by a mixture of two Gaussians with small std and one component fixed at zero  
 $q_{\theta_{i,k}}(w_{i,k}) = (1 - p_i)\mathcal{N}(w_{i,k}; m_{i,k}; \sigma^2 I) + p_i\mathcal{N}(w_{i,k}; 0; \sigma^2 I)$

⇒ A neural network with dropout can be interpreted as a variational Bayesian approximation

# Model uncertainty

Predictive prediction with variational inference approximated with:

$$\begin{aligned} p(y|\mathbf{x}^*, \mathbf{X}, \mathbf{Y}) &= \int p(y|f^W(\mathbf{x}^*)) p(W|\mathbf{X}, \mathbf{Y}) d\mathbf{w} \\ &\approx \int p(y|f^W(\mathbf{x}^*)) q(W) d\mathbf{w} := q_{\mathbf{w}^*}(y|\mathbf{x}^*) \end{aligned}$$

⇒ Estimate  $p(y|\mathbf{x}^*, \mathbf{X}, \mathbf{Y})$  by MC sampling of  $p(y|f^{\hat{W}}(\mathbf{x}^*)), \hat{W} \sim q(W)$

- $W = \{W_i\}_{i=1}^L$  our set of random variables
- $f^W(\mathbf{x}^*)$  our model's stochastic output
- $q_{\mathbf{w}^*}(W)$  our optimum of variational distribution

# Model uncertainty in regression

We will perform moment-matching and estimate the first two moments of the predictive distribution empirically.

## Proposition

Given  $p(y|f^W(\mathbf{x}^*) = \mathcal{N}(y; f^W(\mathbf{x}^*); \tau^{-1}I)$  for some  $\tau > 0$ ,  $\mathbb{E}_{q_{\mathbf{w}^*}(y|\mathbf{x}^*)}[\mathbf{y}]$  can be estimated with the unbiased estimator

$$\widetilde{\mathbb{E}}[y] := \frac{1}{T} \sum_{t=1}^T f^{\hat{\mathbf{W}}_t}(\mathbf{x}^*) \xrightarrow{T \rightarrow \infty} \mathbb{E}_{q_{\mathbf{w}^*}(y|\mathbf{x}^*)}[\mathbf{y}]$$

with  $\hat{\mathbf{W}}_t \sim q(\mathbf{W})$

⇒ equivalent to **performing  $T$  stochastic forward passes through the network and averaging the results.**

# Model uncertainty in regression

## Proposition

Given  $p(y|f^W(\mathbf{x}^*)) = \mathcal{N}(y; f^W(\mathbf{x}^*); \tau^{-1}I)$  for some  $\tau > 0$ ,  $\mathbb{E}_{q_{\mathbf{w}^*}(\mathbf{y}|\mathbf{x}^*)}[(y)^T(y)]$  can be estimated with the unbiased estimator, with  $\hat{\mathbf{W}}_t \sim q(\mathbf{W})$ :

$$\begin{aligned}\mathbb{E}[(y)^T(y)] &:= \tau^{-1}I + \frac{1}{T} \sum_{t=1}^T f^{\hat{\mathbf{W}}_t}(\mathbf{x}^*)^T f^{\hat{\mathbf{W}}_t}(\mathbf{x}^*) \\ &\xrightarrow{T \rightarrow \infty} \mathbb{E}_{q_{\mathbf{w}^*}(\mathbf{y}|\mathbf{x}^*)}[(y)^T(y)]\end{aligned}$$

## Corollary

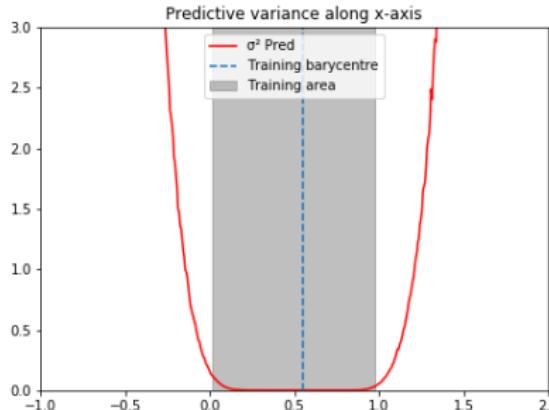
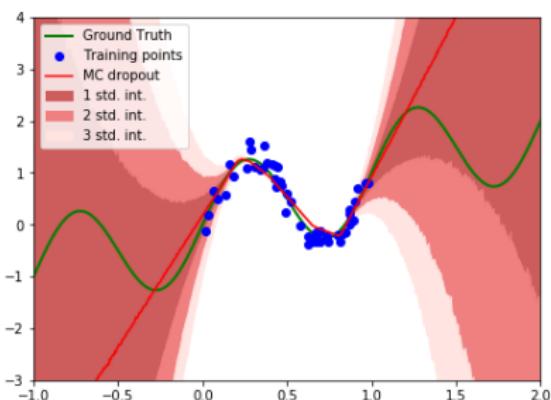
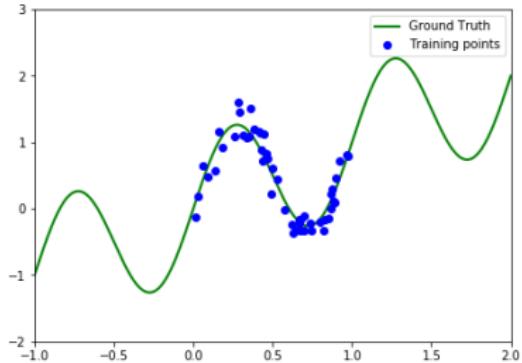
$\text{Var}_{q_{\mathbf{w}^*}(\mathbf{y}|\mathbf{x}^*)}[y]$  can be estimated with the unbiased estimator

$$\begin{aligned}\widetilde{\text{Var}}[(y)] &:= \tau^{-1}I + \frac{1}{T} \sum_{t=1}^T f^{\hat{\mathbf{W}}_t}(\mathbf{x}^*)^T f^{\hat{\mathbf{W}}_t}(\mathbf{x}^*) - \frac{1}{T} \left( \sum_{t=1}^T f^{\hat{\mathbf{W}}_t} \right)^T \left( \sum_{t=1}^T f^{\hat{\mathbf{W}}_t} \right) \\ &\xrightarrow{T \rightarrow \infty} \text{Var}_{q_{\mathbf{w}^*}(\mathbf{y}|\mathbf{x}^*)}[y]\end{aligned}$$

⇒ sample variance of T stochastic forward passes through the NN + the inverse model precision

# Application: MC dropout for predictive distribution

- MC dropout for regression:  $\sin(x) + x$

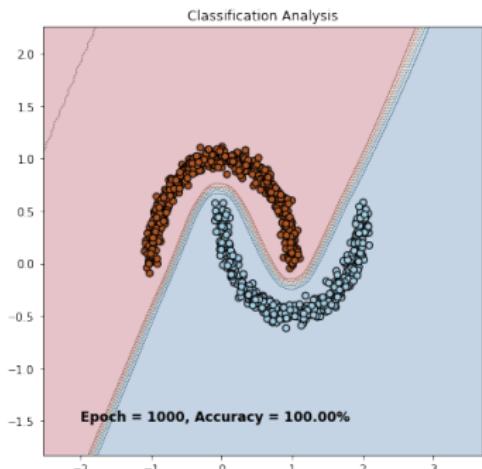


# Application: MC dropout for predictive distribution

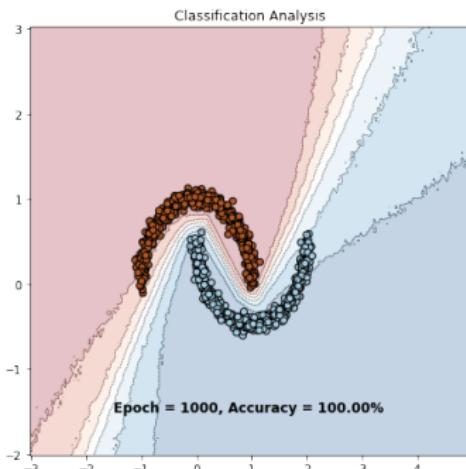
- MC dropout for non-linear classification

- As for BLR:  $p(y = 1|\mathbf{x}^*, \mathcal{D}) \approx \sum_{s=1}^S f_{\mathbf{w}^s}(\mathbf{x}^*)$ ,  $\mathbf{w}^s \sim q(\mathbf{W})$

Deterministic NN



Bayesian NN (MC dropout)



# References |

- [Bishop, 2006] Bishop, C. M. (2006).  
*Pattern Recognition and Machine Learning (Information Science and Statistics)*.  
Springer-Verlag, Berlin, Heidelberg.
- [Blundell et al., 2015] Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015).  
Weight uncertainty in neural network.  
In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1613–1622, Lille, France. PMLR.
- [Gal, 2016] Gal, Y. (2016).  
*Uncertainty in Deep Learning*.  
PhD thesis, University of Cambridge.
- [Gal and Ghahramani, 2016] Gal, Y. and Ghahramani, Z. (2016).  
Dropout as a bayesian approximation: Representing model uncertainty in deep learning.  
In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1050–1059. JMLR.org.
- [Graves, 2011] Graves, A. (2011).  
Practical variational inference for neural networks.  
In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 24*, pages 2348–2356. Curran Associates, Inc.
- [Hernandez-Lobato and Adams, 2015] Hernandez-Lobato, J. M. and Adams, R. (2015).  
Probabilistic backpropagation for scalable learning of bayesian neural networks.  
In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1861–1869, Lille, France. PMLR.
- [Hinton et al., 2012] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2012).  
Improving neural networks by preventing co-adaptation of feature detectors.  
*CoRR*, abs/1207.0580.
- [Hinton and van Camp, 1993] Hinton, G. E. and van Camp, D. (1993).  
Keeping the neural networks simple by minimizing the description length of the weights.  
In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, COLT '93, pages 5–13, New York, NY, USA. ACM.

# References II

- [Jylänki et al., 2014] Jylänki, P., Nummenmaa, A., and Vehtari, A. (2014).  
Expectation propagation for neural networks with sparsity-promoting priors.  
*Journal of Machine Learning Research*, 15:1849–1901.
- [Kingma and Welling, 2014] Kingma, D. P. and Welling, M. (2014).  
Auto-encoding variational bayes.  
In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.
- [MacKay, 1992] MacKay, D. J. C. (1992).  
A practical bayesian framework for backpropagation networks.  
*Neural Comput.*, 4(3):448–472.
- [Murphy, 2012] Murphy, K. P. (2012).  
*Machine Learning: A Probabilistic Perspective*.  
The MIT Press.
- [Neal, 1996] Neal, R. M. (1996).  
*Bayesian Learning for Neural Networks*.  
Springer-Verlag, Berlin, Heidelberg.