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

Nicolas Thome
Conservatoire National des Arts et Métiers (Cnam)
Laboratoire CEDRIC - équipe MSDMA

Outline

Beyond Bayesian Linear Regression

- Posterior distribution for parameters w: $p(\mathrm{w}|\mathrm{X},\mathrm{Y}) \propto p(\mathrm{Y}|\mathrm{X},\mathrm{w})p(\mathrm{w})$
- Predictive distribution $p(y^*|\mathrm{x}^*,\mathcal{D}) = \int p(y^*|\mathrm{x}^*,\mathrm{w})p(\mathrm{w}|\mathcal{D})dw$
- **Closed form for posterior $p(\mathrm{w}|\mathrm{X},\mathrm{Y})$ and predictive distribution $p(y^*|\mathrm{x}^*,\mathcal{D})$: 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

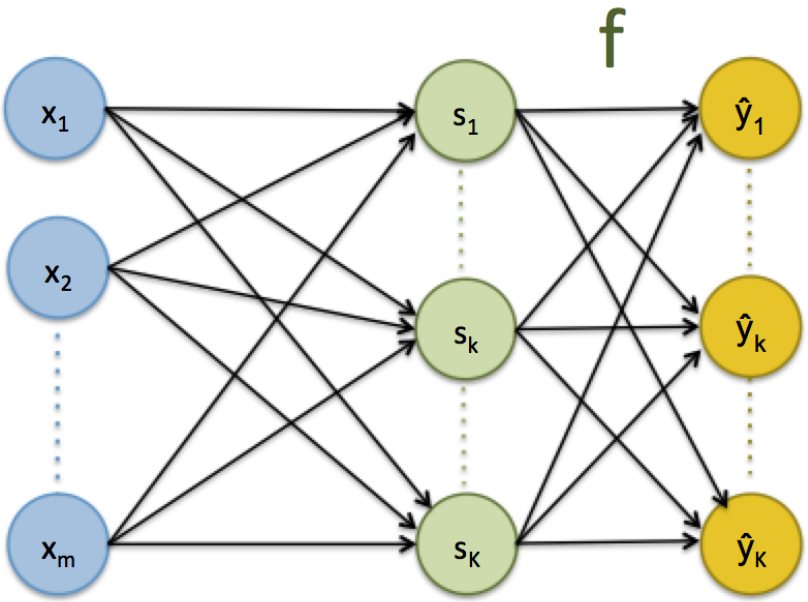
No analytical expression for posterior $p(\mathrm{w}|\mathrm{X},\mathrm{Y})$ and $p(y^*|\mathrm{x}^*,\mathcal{D})$ in general

$\Rightarrow$ **Approximation nedded!**

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

Outline

Bayesian Logistic Regression (BLR)



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

$$p(w|X, Y) \propto p(Y|X, w)p(w)$$

- $p(Y|X, w) = \prod_{i=1}^{N} p(y_i = 1|x_i, w)$ not Gaussian anymore!

$\Rightarrow$ **no closed-form on posterior distribution** $p(w|X, Y)$**!**

Bayesian Logistic Regression training (MAP)

$$\boldsymbol{w}_{\text{MAP}} = arg\max_{\boldsymbol{w}} p(\mathrm{X}, \mathrm{Y}|\boldsymbol{w})p(\boldsymbol{w}) = arg\max_{\boldsymbol{w}} \prod_{n=1}^{N} p(y_n|\boldsymbol{x}_n, \boldsymbol{w})p(\boldsymbol{w})$$

$$= arg\min_{\boldsymbol{w}} \sum_{n=1}^{N} -log\left(p(y_n|\boldsymbol{x}_n, \boldsymbol{w})\right) - log\left(p(\boldsymbol{w})\right)$$

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

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

  - Again: Gaussian prior $\Leftrightarrow$ weight decay

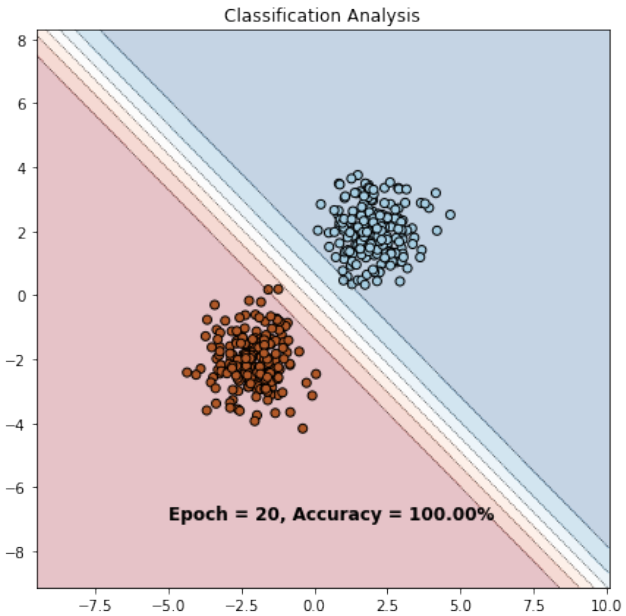
Bayesian Logistic Regression training (MAP)

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

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

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

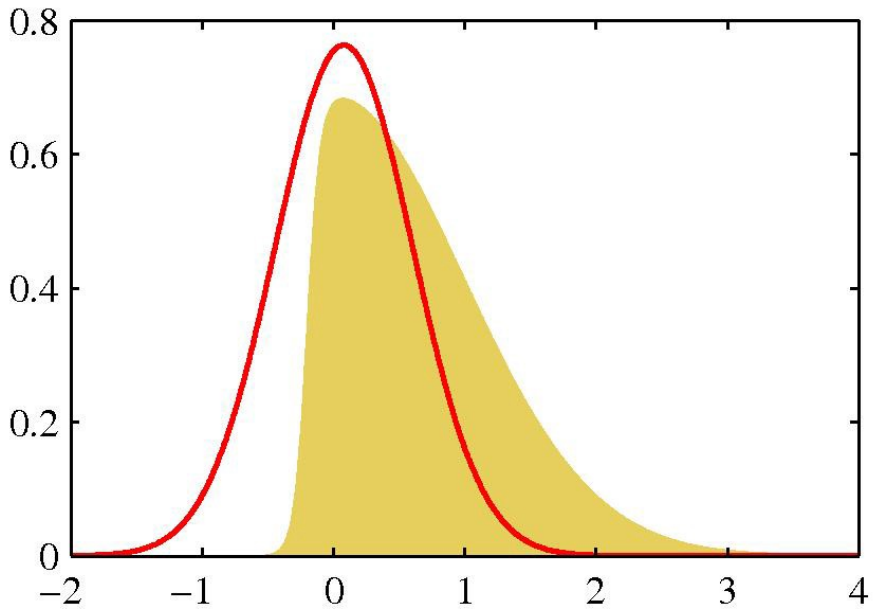
  - Need full posterior distribution $p(\text{w}|\text{X}, \text{Y})$, but posterior intractable
  - $p(\text{w}|\text{X}, \text{Y}) \approx \delta(\text{w} - \boldsymbol{w}_{\text{MAP}}) \Rightarrow p(\text{y} = 1|\boldsymbol{x}^*, \mathcal{D}) \approx p(y = 1|\boldsymbol{x}, \boldsymbol{w}_{\text{MAP}})$



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

Laplace Approximation for $p(w|X,Y)$

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



from [Bishop, 2006]

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

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

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

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

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

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

- However, likelihood $p(y^*|\mathrm{x}^*, \mathcal{D})$ still not Gaussian

$\Rightarrow$ **Intractable posterior distribution** $p(y^*|\mathrm{x}^*, \mathcal{D})$**!**

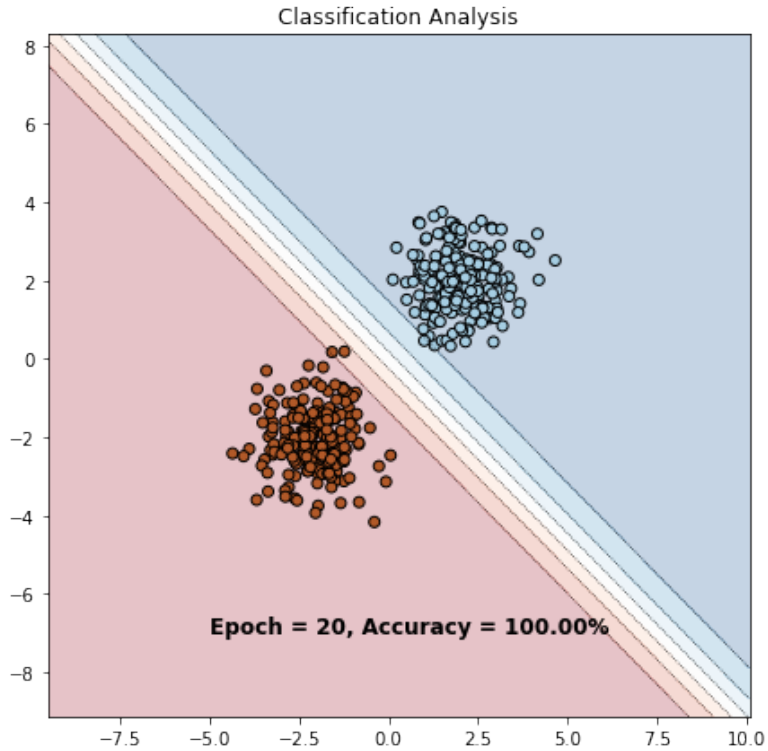
Predictive Distribution $p(y^*|x^*, \mathcal{D})$ for BLR

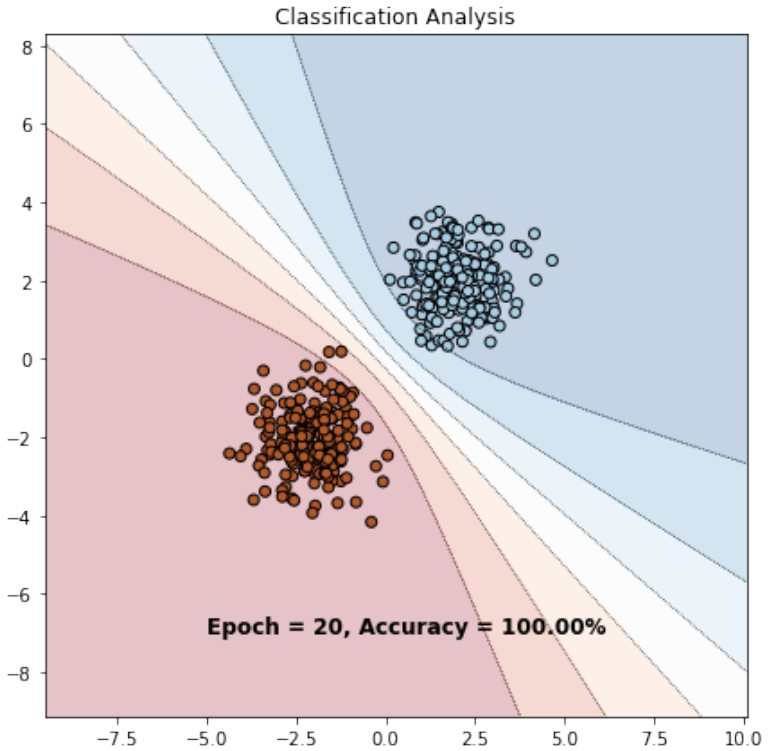
- **Option 1:** use Monte Carlo (MC) sampling
  - Binary case (simple x-class extension): $p(y^* = 1|x^*, w) = \sigma(w^T x^*)$, σ sigmoid

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

  - Easy to sample from Gaussian $q(w)$

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

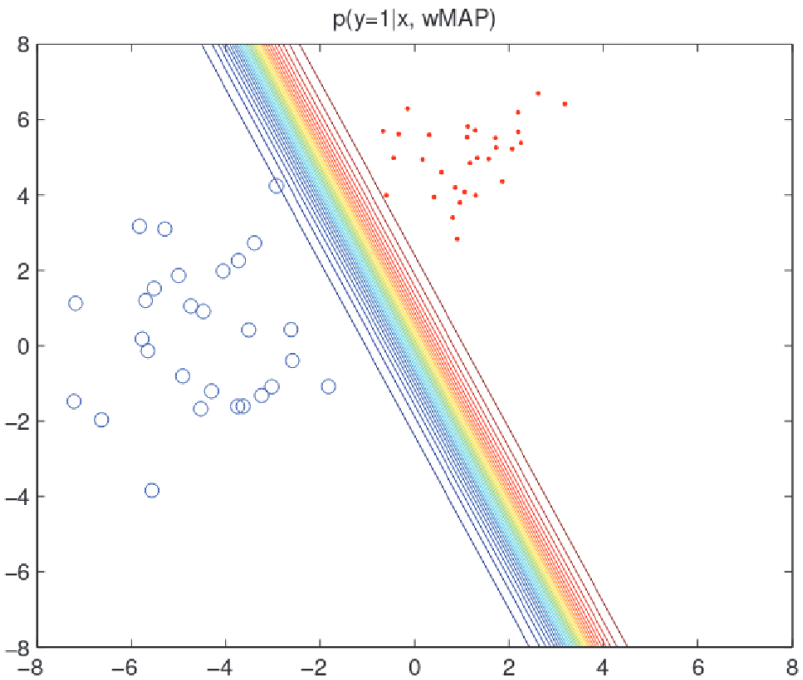


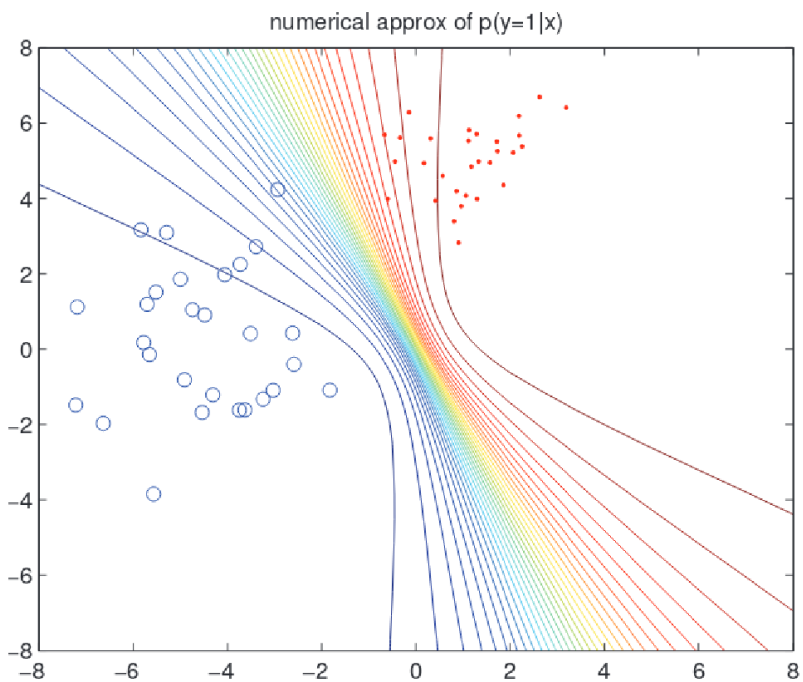


Predictive Distribution $p(y^*|x^*, \mathcal{D})$ for BLR

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

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

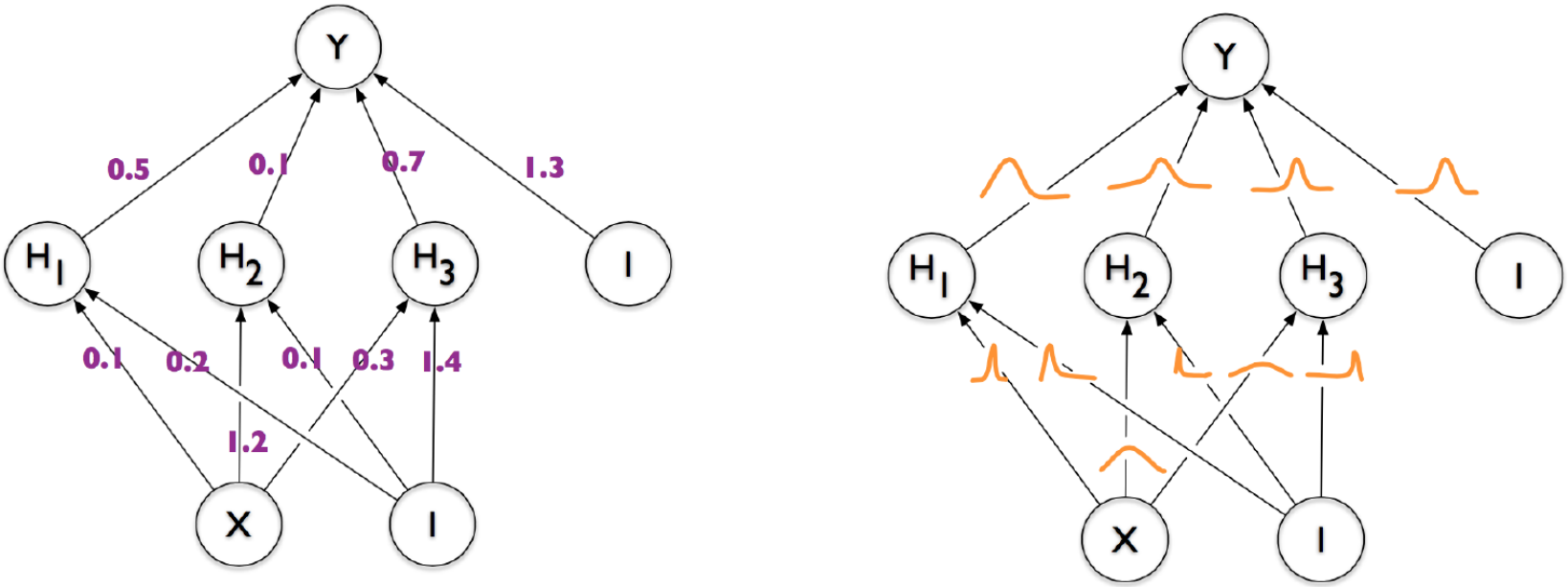




from [Murphy, 2012]

Outline

Bayesian Neural Networks (BNN)



Credit: [Blundell et al., 2015]

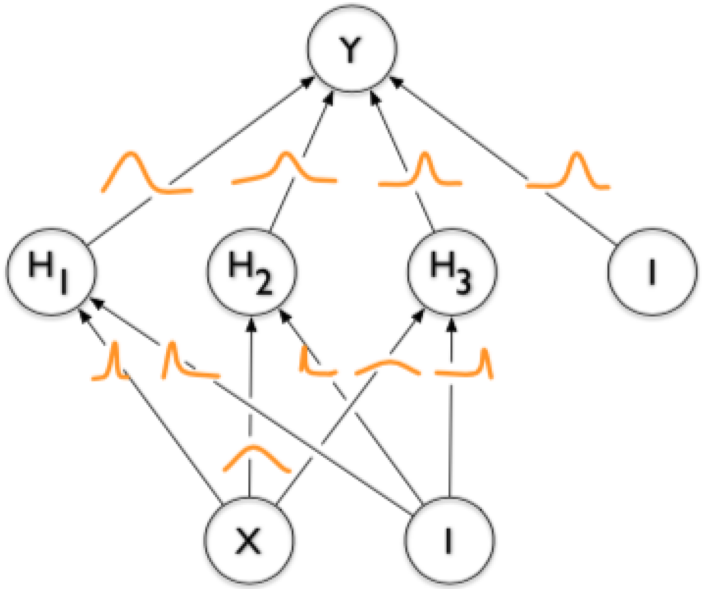
- Standard NN: $y_i = f^w(x_i)$, **Bayesian NN:** $p(y_i|x_i, \mathcal{D})$
  - Define prior over weights $p(w)$, *e.g.* $p(w) = \mathcal{N}(w|0, \alpha^{-1}\mathcal{I})$ (point estimate for bias)
    - In practice, typically separate variance $\sigma^2 = \alpha^{-1}$ for each layer
  - Define likelihood,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

Bayesian Neural Networks (BNN)

$$p(\mathrm{w}|\mathrm{X},\mathrm{Y}) \propto p(\mathrm{w}) \prod_{i=1}^{N} p(\mathrm{y}_i|\mathrm{x}_i,\mathrm{w})$$

- **With Bayesian Neural networks, even with:**
  - Gaussian prior $p(\mathrm{w}) = \mathcal{N}(\mathrm{w}|0,\alpha^{-1}\mathcal{I})$
  - Gaussian likelihood, *e.g.* regression $p(\mathrm{y}_i|\mathrm{x}_i,\mathrm{w}) = \mathcal{N}(y_i; f^w(\mathrm{x}_i),\beta^{-1})$
  - Posterior $p(\mathrm{w}|\mathrm{X},\mathrm{Y},\beta) \propto p(\mathrm{w}) \prod_{i=1}^{N} p(\mathrm{y}_i|\mathrm{x}_i,\mathrm{w})$ **is NOT Gaussian!!**

- **Non-linear dependence of $f^w(\mathrm{x})$ on w!**
- RECAP:
  - $p(\mathrm{x}) = \mathcal{N}(\mathrm{x}|\boldsymbol{\mu}_x,\Sigma_x)$
  - $p(\mathrm{y}|\mathrm{x}) = \mathcal{N}(\mathrm{y}|A\mathrm{x}+b,\Sigma_y)$
    - Linear depedence $A\mathrm{x}+b$ required
  - Then: $p(\mathrm{x}|\mathrm{y}) = \mathcal{N}(\mathrm{x}|\boldsymbol{\mu}_{x|\mathrm{y}},\Sigma_{x|\mathrm{y}})$
    - Not true for BNNs!



Credit: [Blundell et al., 2015]

Posterior Inference: MCMC Carlo Sampling

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

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

- Monte Carlo estimation of the intergral:

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

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

Variational Inference (VI)

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

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

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

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

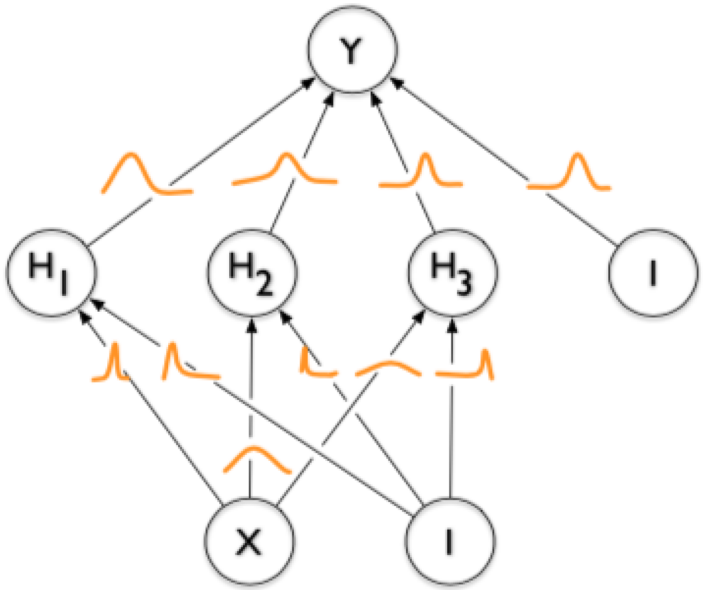
$$p(\mathrm{y}^*|\mathrm{x}^*,\mathrm{X},\mathrm{Y}) \approx \int p(\mathrm{y}^*|x^*,\mathrm{w}) q_{\boldsymbol{\theta}^*}(\mathrm{w}) d\mathrm{w}$$

Variational Inference (VI)

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

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

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



Credit: [Blundell et al., 2015]

Variational Inference (VI): ELBO

$$KL(q_\theta(\mathrm{w})||p(\mathrm{w}|\mathrm{X},\mathrm{Y})) = \int q_\theta(\mathrm{w}) \log \frac{q_\theta(\mathrm{w})}{p(\mathrm{w}|\mathrm{X},\mathrm{Y})} dw = -\int q_\theta(\mathrm{w}) \log \frac{p(\mathrm{w}|\mathrm{X},\mathrm{Y})}{q_\theta(\mathrm{w})} dw^1$$

$$= -\int q_\theta(\mathrm{w}) \log \frac{p(\mathrm{Y}|\mathrm{X},\mathrm{w})p(\mathrm{w})}{q_\theta(w)p(\mathrm{Y}|\mathrm{X})} dw$$

$$= -\int q_\theta(\mathrm{w}) \log p(\mathrm{Y}|\mathrm{X},\mathrm{w}) dw + \int q_\theta(\mathrm{w}) \log \frac{q_\theta(\mathrm{w})}{p(\mathrm{w})} + \log p(\mathrm{Y}|\mathrm{X})$$

$$= -\int q_\theta(\mathrm{w}) \log p(\mathrm{Y}|\mathrm{X},\mathrm{w}) dw + KL(q_\theta(\mathrm{w})||p(\mathrm{w})) + \log p(\mathrm{Y}|\mathrm{X})$$

- $\Rightarrow \boxed{KL(q_\theta(\mathrm{w})||p(\mathrm{w}|\mathrm{X},\mathrm{Y})) = -\mathcal{L}_{VI}(\mathrm{X},\mathrm{Y},\theta) + \log p(\mathrm{Y}|\mathrm{X})}$
  - $\mathcal{L}_{VI}(\mathrm{X},\mathrm{Y},\theta)$: Evidence Lower Bound (ELBO)

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

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

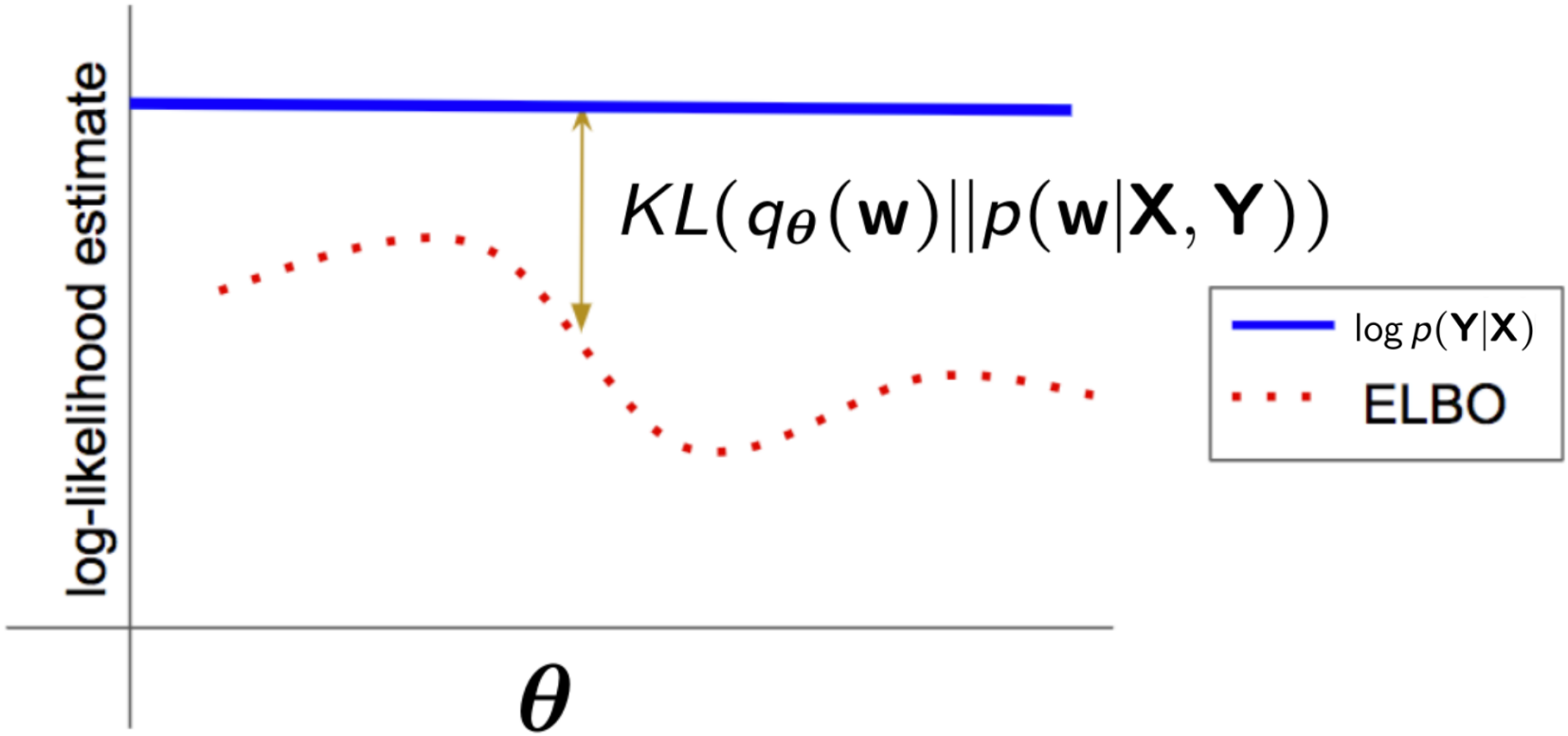
$p(\mathrm{w}|\mathrm{X},\mathrm{Y}) = \frac{p(\mathrm{Y}|\mathrm{X},\mathrm{w})p(\mathrm{w})}{p(\mathrm{Y}|\mathrm{X})}$

ELBO Illustration

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

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

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

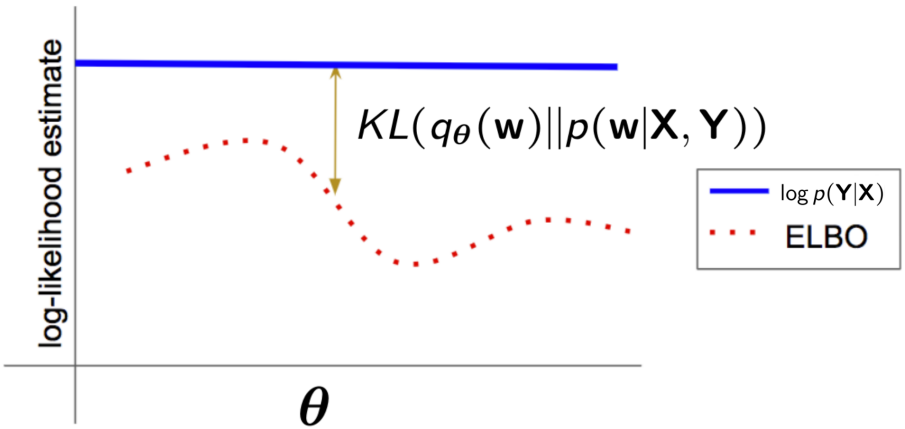


Variational Inference (VI): ELBO

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

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

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



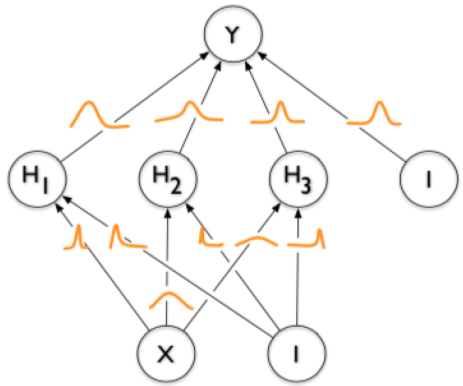
Variational Inference: Training

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

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

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

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



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

Stochastic Variational Inference (VI): Training

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

- **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** $\hat{\mathrm{w}}_i \sim q_{\boldsymbol{\theta}}(\mathrm{w})$
  - Sample $\log p(\mathrm{y}|f^{\hat{\mathrm{w}}_i}(\mathrm{x}_i))$, $\hat{\mathrm{w}}_i \sim q_{\boldsymbol{\theta}}(\mathrm{w})$

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

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

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

Stochastic Variational Inference (VI): Training

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

1: Given dataset $\mathbf{X}, \mathbf{Y}$,
2: Define learning rate schedule η,
3: Initialise parameters θ randomly.
4: **repeat**
5: Sample M random variables $\widehat{\boldsymbol{\epsilon}}_i \sim p(\boldsymbol{\epsilon})$, S a random subset of $\{1, .., N\}$ of size M.
6: Calculate stochastic derivative estimator w.r.t. θ:

$$\widehat{\Delta\theta} \leftarrow -\frac{N}{M}\sum_{i\in S}\frac{\partial}{\partial\theta}\log p(\mathbf{y}_i|\mathbf{f}^{g(\theta,\widehat{\boldsymbol{\epsilon}}_i)}(\mathbf{x}_i)) + \frac{\partial}{\partial\theta}\mathrm{KL}(q_\theta(\boldsymbol{\omega})||p(\boldsymbol{\omega})).$$

7: Update θ:

$$\theta \leftarrow \theta + \eta\widehat{\Delta\theta}.$$

8: **until** θ has converged.

Bayesian Neural Networks: Predictive Distribution

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

Solutions:

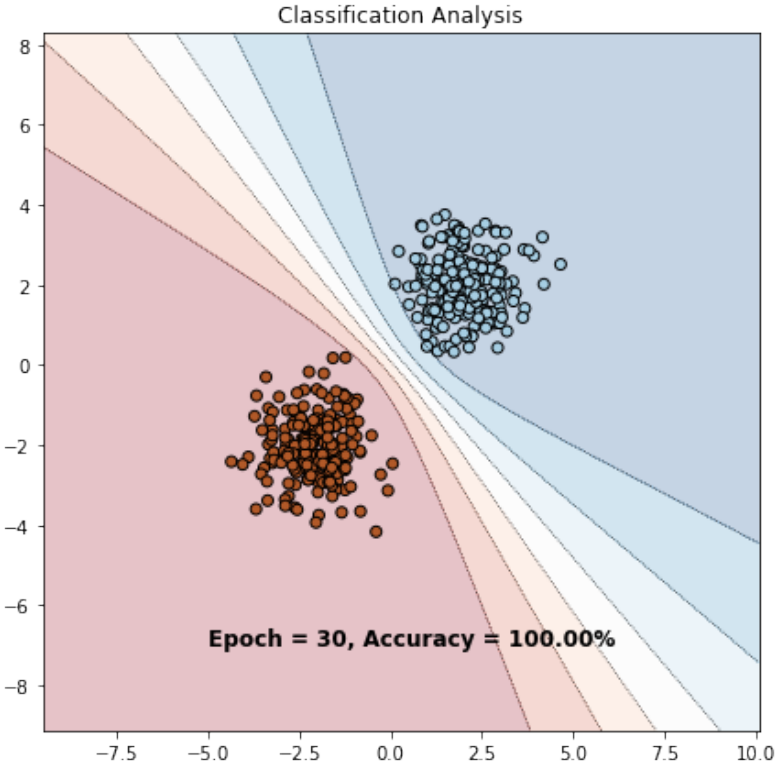
1. MC sampling, easy to sample from $q_\theta(\mathrm{w})$
2. Perform Taylor expansion of $y^*(\mathrm{x}^*, \mathrm{w})$ around w_{MAP} for regression[a]:
   $y(\mathrm{x}, \mathrm{w}) \approx y(\mathrm{x}, \mathrm{w}_{MAP}) + \frac{\partial y}{\partial \mathrm{w}}|_{\mathrm{w}=\mathrm{w}_{MAP}} (\mathrm{w} - \mathrm{w}_{MAP})$

- $p(y^*|\mathrm{x}^*, \mathrm{w}) \approx \mathcal{N}(y^*|\mu_y; \beta^{-1})$, $\mu_y = y(\mathrm{x}, \mathrm{w}_{MAP}) + \frac{\partial y}{\partial \mathrm{w}}|_{\mathrm{w}=\mathrm{w}_{MAP}} (\mathrm{w} - \mathrm{w}_{MAP})$
- Closed form solution for $p(\mathrm{y}^*|\mathrm{x}^*, \mathcal{D})$
  - $p(\mathrm{y}^*|\mathrm{x}^*, \mathcal{D}) = \mathcal{N}\left(y^*|y(\mathrm{x}, \mathrm{w}_{MAP}); \sigma^2(x)\right)$
  - $\sigma^2(x) = \beta^{-1} + \boldsymbol{g}^T \boldsymbol{A}^{-1} \boldsymbol{g}$, $\boldsymbol{g}$ gradient and $\boldsymbol{A}$ Hessian at w_{MAP}

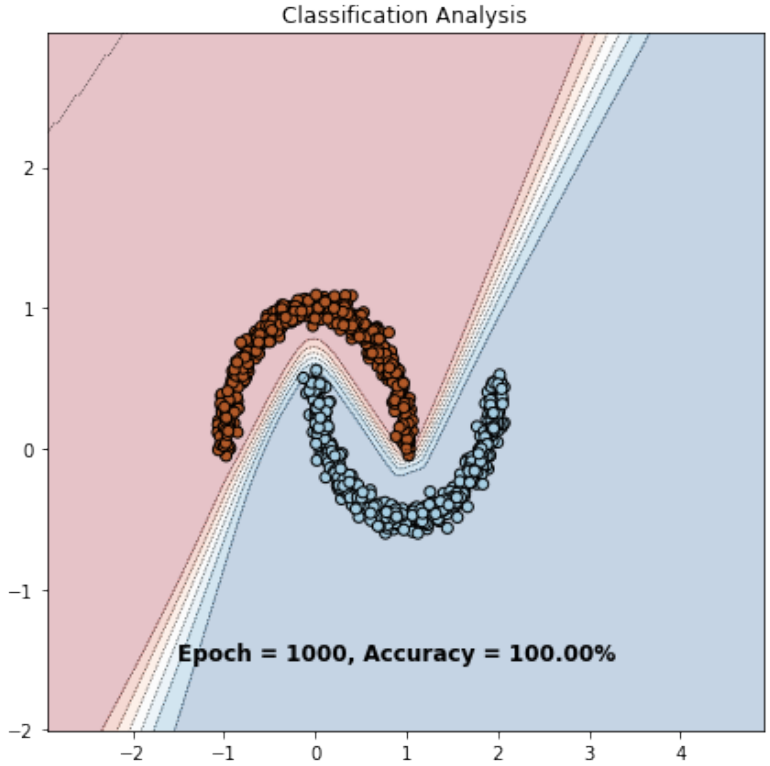
[a] For classification, logit Taylor expansion, see 5.7.1 in [Bishop, 2006]

Practical session

- Own implementation of VI for:
  - Bayesian logistic regression
  - Neural network for non-linear classification (moons)
- Option: pyro `https://pyro.ai/` $\Rightarrow$ stochastic process sampling, posterior approximation (VI), predictive distribution with MC sampling



$p(y = 1|x, X, Y)$

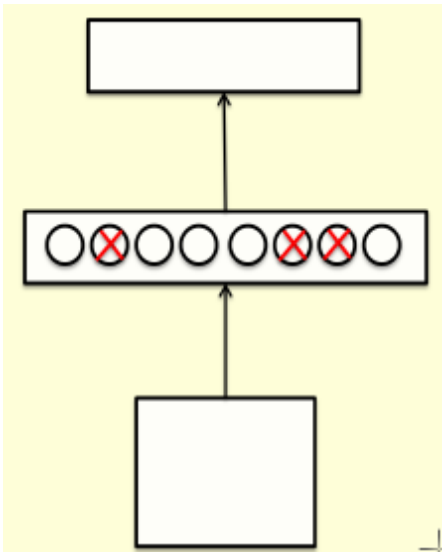


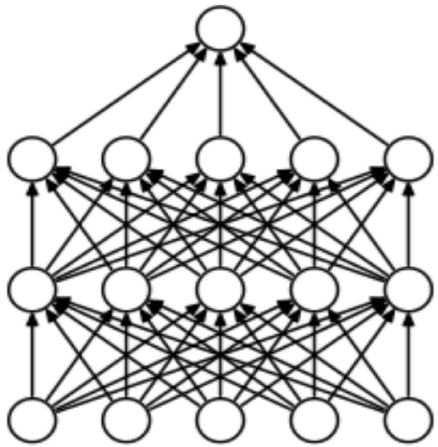
$p(y = 1|x, X, Y)$

Outline

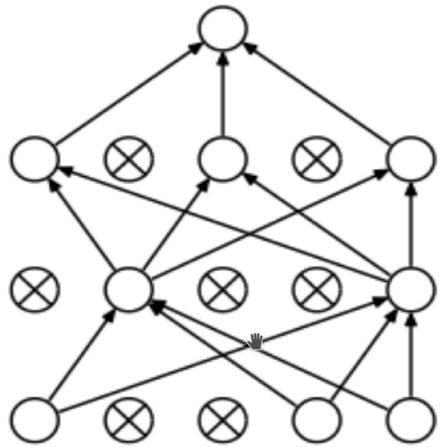
Dropout [Hinton et al., 2012]

- 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





Standard Neural Net

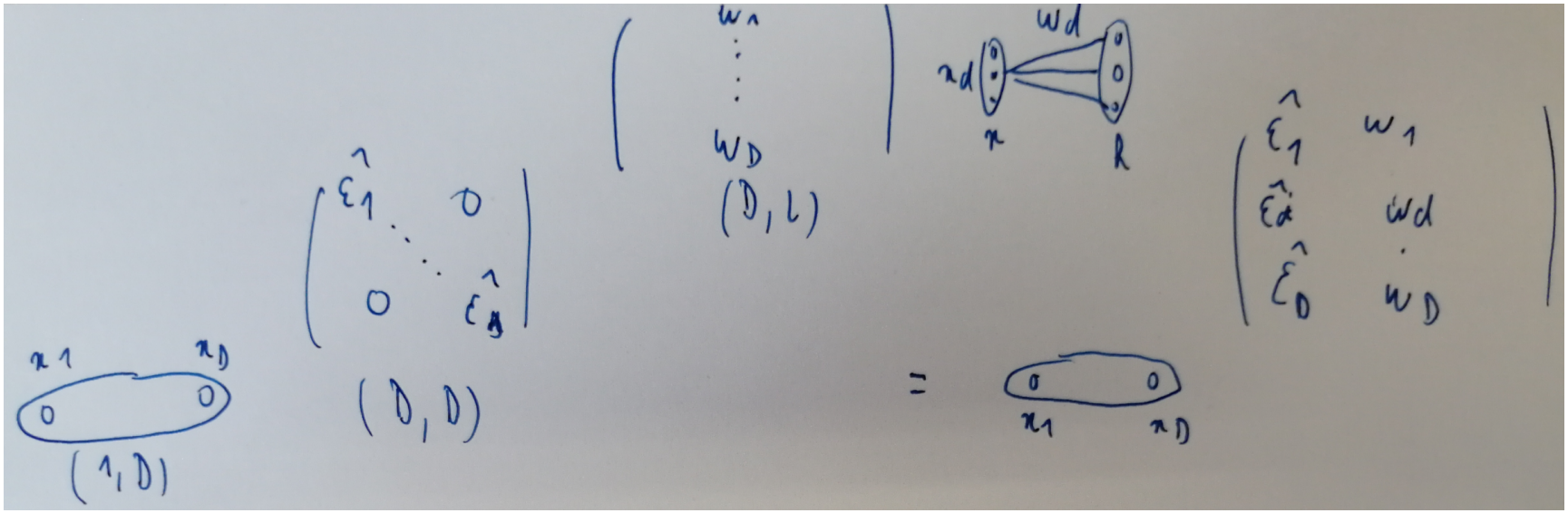


After applying dropout.

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



[a] dimension (1,D)

Dropout as a variational inference [Gal, 2016]

- **Illustration: dropout for a 2 layer NN** (1 hidden), $\epsilon_i \sim Bernoulli(1-p_i)$:
  - $\boxed{h_1 = \sigma(\hat{x}W_1) = \sigma(x\hat{W}_1)}$, $\hat{W}_1 = \text{diag}(\hat{\epsilon_1})W_1$
  - $\boxed{\hat{y} =: f^{\hat{W_1},\hat{W_2}}(x) = \hat{h_1}W_2 = h_1\hat{W}_2}$, $\hat{W}_2 = \text{diag}(\hat{\epsilon_2})W_2$ - $\hat{W} = \{\hat{W}_1; \hat{W}_2\}$
- **MC Dropout sampling:** $\frac{1}{S}\sum_{s\in S} p(y_i|f^{\hat{W}}(x_i)) \approx \int p(y^*|f^W(x^*)q(W)dw$
  - $\forall$ layer $l \in \{1; L\}$, W_l random variable: $W_l \sim q(W_l) = g(M_l, \varepsilon_l) = \text{diag}(\varepsilon_l)M_l$
    - $\varepsilon_{l,i} \sim Bernoulli(1-p_l)$, M_l deterministic parameters
  - $q(W) = \prod_{l=1}^{M} q(W_l)$
- **Big result** (see next): **training NN with dropout $\Leftrightarrow$ training BNN with variational posterior approximation** $q_M(W)$ (and some prior $p(W)$)
  - $M = \{M_l\}_{l\in\{1;L\}}$ variational parameters
- **MC dropout:** sampling several passes with dropout $\Leftrightarrow$ **performing MC approximate inference with variational posterior** $q_M(W)$

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

Dropout as a variational inference [Gal, 2016]

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

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

$$\hat{\mathcal{L}}_{dropout}(M_1, M_2) = \frac{1}{M}\sum_{i \in S} ||f^{\hat{W}}(x_i) - y_i||^2 + \lambda_1||M_1||^2 + \lambda_2||M_2||^2$$

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

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

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

$$\hat{\mathcal{L}}_{dropout}(M_1, M_2) = \frac{1}{M\tau}\sum_{i \in S} \log p(y_i|f^{g(M,\hat{\epsilon}^i)}(x) + \lambda_1||M_1||^2 + \lambda_2||M_2||^2 \quad (1)$$

Dropout as a variational inference [Gal, 2016]

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

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

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

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

$\Rightarrow$ **A neural network with dropout can be interpreted as a variational Bayesian approximation**

Model uncertainty

Predictive prediction with variational inference approximated with:

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

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

- $\mathrm{W} = \{W_l\}_{i=l}^{L}$ our set of random variables
- $f^{\mathrm{W}}(\mathrm{x}^*)$ our model's stochastic output
- $q_{\mathrm{W}^*}(\mathrm{W})$ 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 $p(y^*|f^{\mathsf{W}}(\mathsf{x}^*) = \mathcal{N}(y^*; f^{\mathsf{W}}(\mathsf{x}^*); \tau^{-1} I)$ for some $\tau > 0$,
$\mathbb{E}_{q_{\mathbf{w}^*}(y^*|\mathsf{x}^*)}[y^*]$ can be estimated with the unbiased estimator

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

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

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

Model uncertainty in regression

Proposition

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

$$\widetilde{\mathbb{E}}[(y^*)^T(y^*)] := \tau^{-1}I + \frac{1}{T}\sum_{t=1}^{T} f^{\hat{\mathrm{W}}_t}(\mathrm{x}^*)^T f^{\hat{\mathrm{W}}_t}(\mathrm{x}^*)$$

$$\xrightarrow[T\to\infty]{} \mathbb{E}_{q^*_{\mathbf{w}}(y^*|\mathrm{x}^*)}[(y^*)^T(y^*)]$$

Corollary

$Var_{q^*_{\mathbf{w}}(y^*|\mathrm{x}^*)}[(y^*)]$ can be estimated with the unbiased estimator

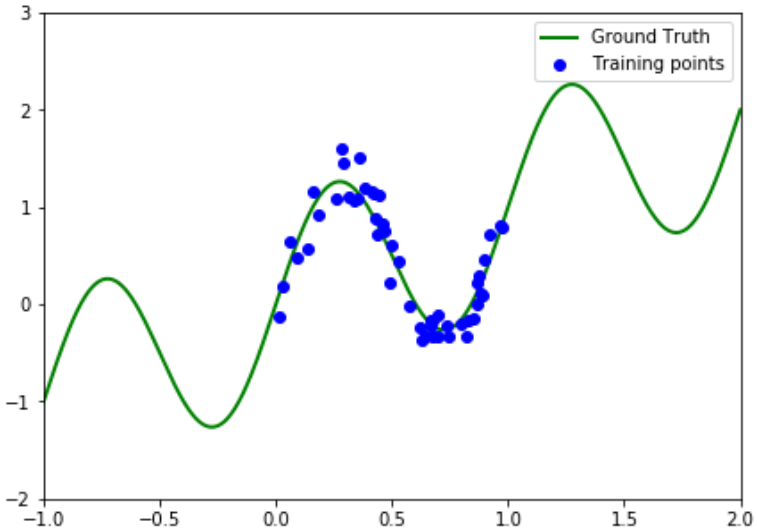
$$\widetilde{Var}[(y^*)] := \tau^{-1}I + \frac{1}{T}\sum_{t=1}^{T} f^{\hat{\mathrm{W}}_t}(\mathrm{x}^*)^T f^{\hat{\mathrm{W}}_t}(\mathrm{x}^*) - \frac{1}{T}\left(\sum_{t=1}^{T} f^{\hat{\mathrm{W}}_t}\right)^T \left(\sum_{t=1}^{T} f^{\hat{\mathrm{W}}_t}\right)$$

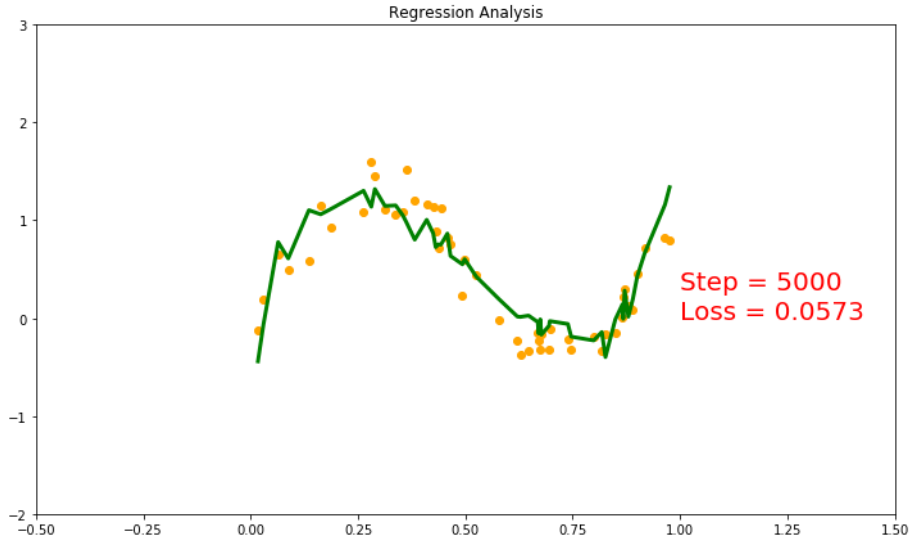
$$\xrightarrow[T\to\infty]{} Var_{q^*_{\mathbf{w}}(y^*|\mathrm{x}^*)}[y^*]$$

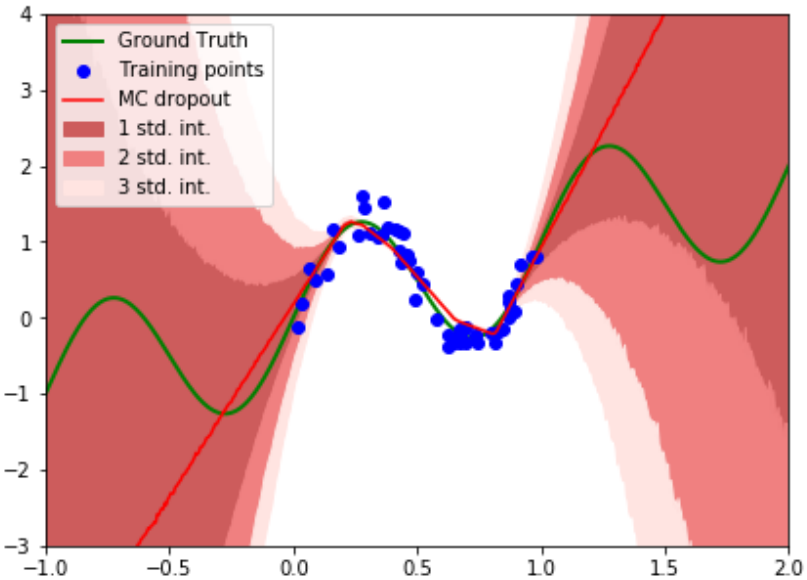
$\Rightarrow$ **sample variance of T stochastic forward passes** through the NN + the **inverse model precision**

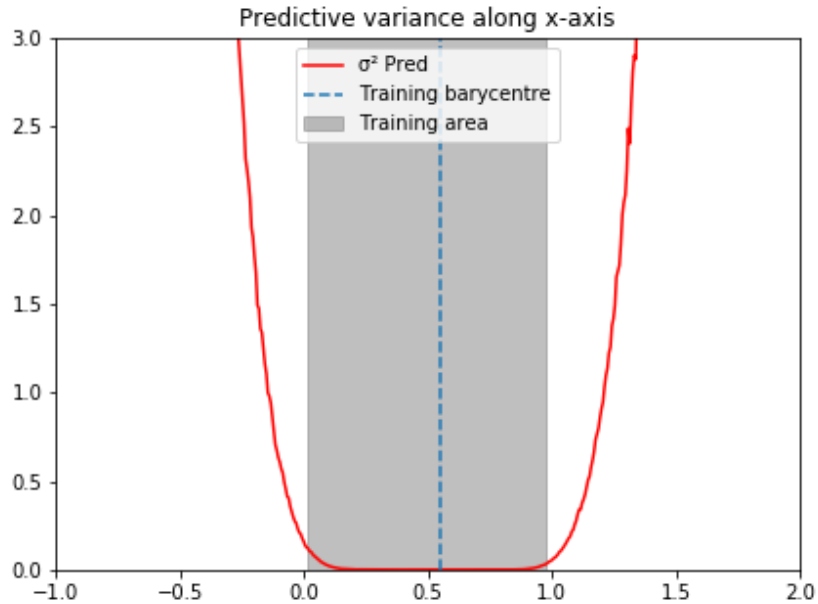
Application: MC dropout for predictive distribution

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

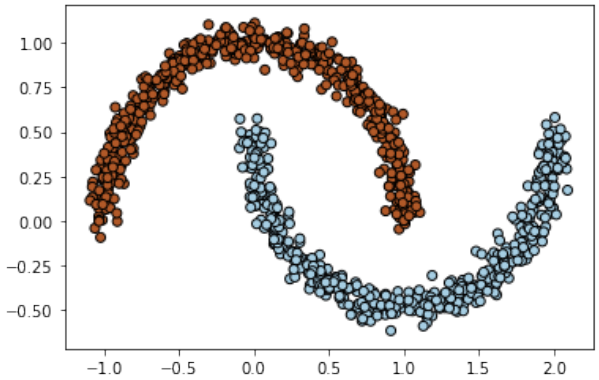








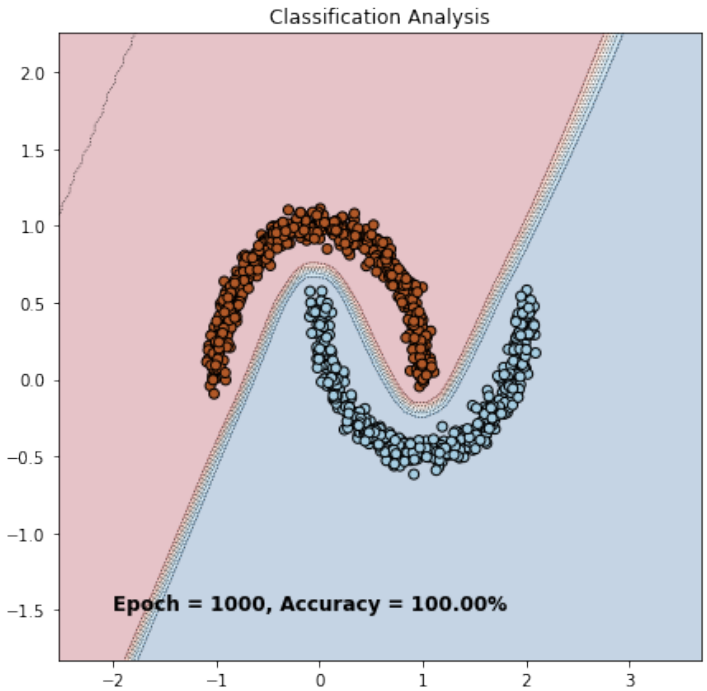
Application: MC dropout for predictive distribution



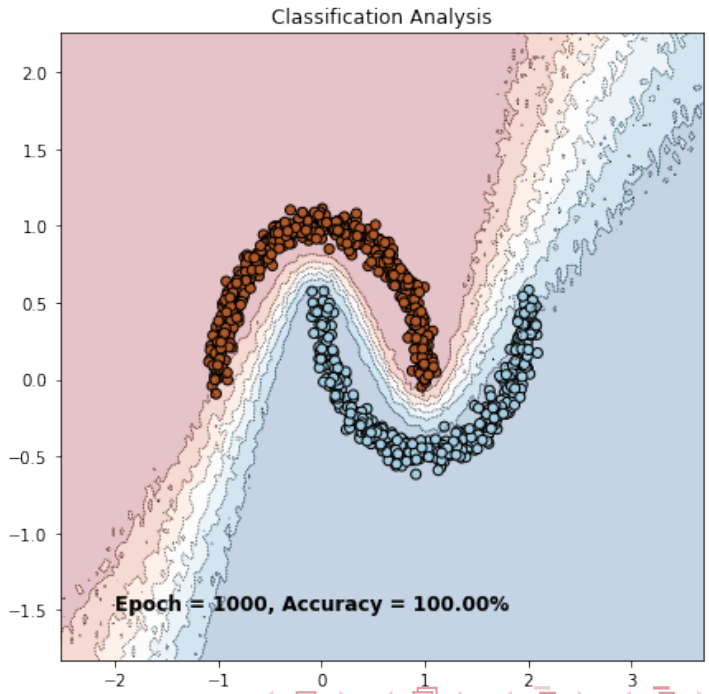
- MC dropout for non-linear classification

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

Deterministic NN



Bayesian NN (MC dropout)



References I

[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.