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

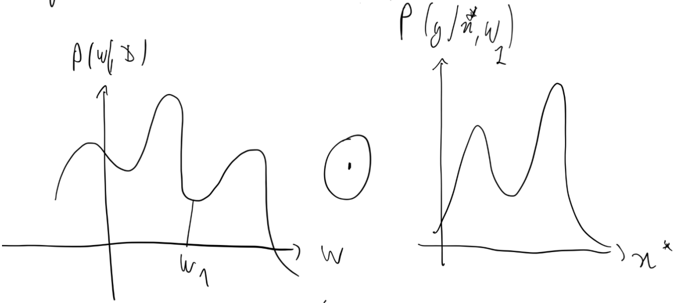
Bayesian Neural Networks

Monte Carlo Dropout

Beyond Bayesian Linear Regression

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

$$p(y, w | x^*, D) \propto p(y | x^*, w)\, p(w | D)$$



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

Beyond Bayesian Linear Regression

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

- **Closed form for posterior $p(\boldsymbol{w}|\mathcal{D})$ and predictive distribution $p(y|\boldsymbol{x}^*, \mathcal{D})$: 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

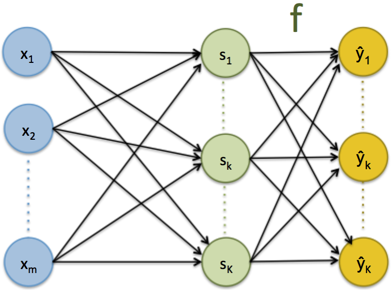
No analytical expression for posterior $p(\boldsymbol{w}|\mathcal{D})$ and $p(y|\boldsymbol{x}^*, \mathcal{D})$ in general

$\Rightarrow$ **Approximation nedded!**

- **Gaussian approximation for $p(\boldsymbol{w}|\mathcal{D})$**
  - **Ex: Laplace approximation** [MacKay, 1992]
  - Historically used for bayesian logistic regression
- **Monte Carlo methods**: sampling to directly evaluate integral $p(y|\boldsymbol{x}^*, \mathcal{D})$
  - **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 $p(\boldsymbol{w}|\mathcal{D})$ and a proposed parametric function

Outline

Bayesian Logistic Regression (BLR)



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

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

- $p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{w}) = \prod_{i=1}^{N} p(\boldsymbol{y}_i = 1|\boldsymbol{x}_i, \boldsymbol{w})$ not Gaussian anymore!

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

Bayesian Logistic Regression training (MAP)

$$\boldsymbol{w}_{\mathrm{MAP}} = arg\max_{\boldsymbol{w}} p(\boldsymbol{X},\boldsymbol{Y}|\boldsymbol{w})p(\boldsymbol{w}) = arg\max_{\boldsymbol{w}} \prod_{n=1}^{N} p(y_n|\boldsymbol{x}_n,\boldsymbol{w})p(\boldsymbol{w})$$
$$= arg\min_{\boldsymbol{w}} \sum_{n=1}^{N} -log\left(p(y_n|\boldsymbol{x}_n,\boldsymbol{w})\right) - log\left(p(\boldsymbol{w})\right)$$

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

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

  - Again: Gaussian prior $\Leftrightarrow$ weight decay

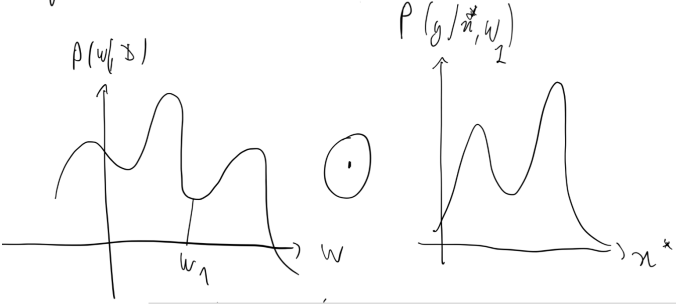
Bayesian Logistic Regression training (MAP)

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

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

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

$$P(y, w | x^*, D) \propto P(y | x^*, w) \, P(w | D)$$



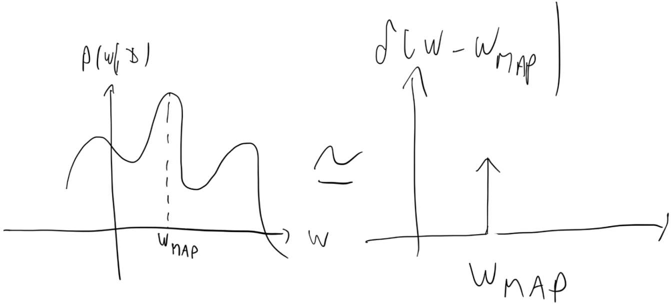
Bayesian Logistic Regression training (MAP)

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

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

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

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



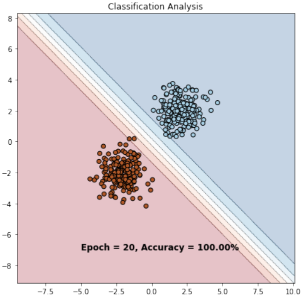
Bayesian Logistic Regression training (MAP)

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

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

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

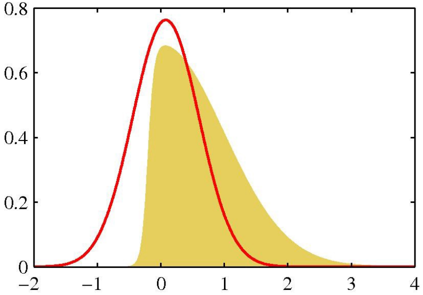
  - Need full posterior distribution $p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{Y})$, but posterior intractable
  - $p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{Y}) \approx \delta(\boldsymbol{w} - \boldsymbol{w}_{\text{MAP}}) \Rightarrow p(y = 1|\boldsymbol{x}^*, \mathcal{D}) \approx p(y = 1|\boldsymbol{x},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



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

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

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



from [Bishop, 2006]

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

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

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

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

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

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

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

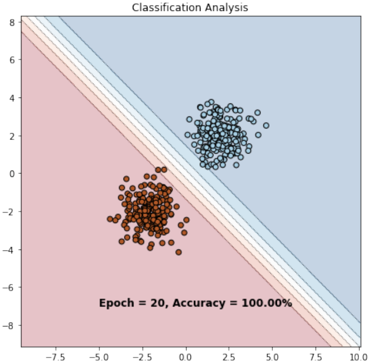
$\Rightarrow$ **Intractable posterior distribution** $p(y|\boldsymbol{x}^*, \mathcal{D})$**!**

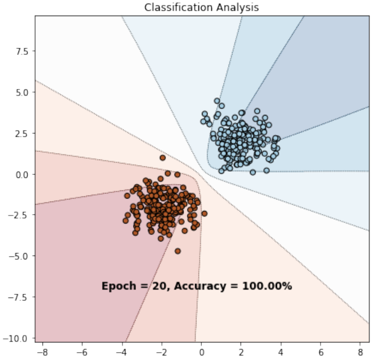
Predictive Distribution $p(y|\boldsymbol{x}^*, \mathcal{D})$ for BLR

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

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

    - Easy to sample from Gaussian $q(\boldsymbol{w})$
- **Practical session:** MAP solution for LR *vs* BLR (Laplace & MC sampling)

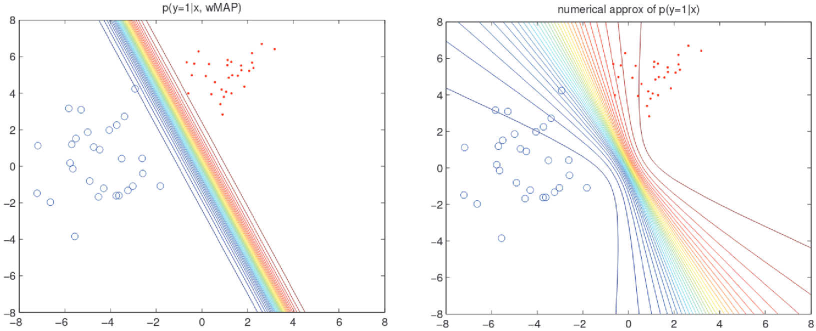




Predictive Distribution $p(y|\mathbf{x}^*, \mathcal{D})$ for BLR

- $p(y|\mathbf{x}^*, \mathcal{D}) \approx \int p(y|\mathbf{x}^*, \mathbf{w})q(\mathbf{w})dw$ intractable
- **Option 2:** (binary case): $p(y|\mathbf{x}^*, \mathcal{D}) \approx \int \sigma(\mathbf{w}^T\mathbf{x}^*)q(\mathbf{w})dw$; $q(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$
- 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:

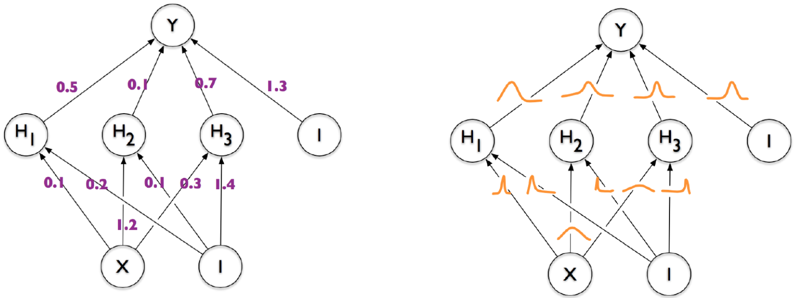
$$p(y|\mathbf{x}^*, \mathcal{D}) \approx \int \Phi\left(\lambda \mathbf{w}^T\mathbf{x}^*\right) \mathcal{N}(\mathbf{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \Phi\left(\frac{\mu_f}{\sqrt{\lambda^{-2}+\sigma_f^2}}\right) \quad \mu_f = \boldsymbol{\mu}^T\mathbf{x}^* \quad \sigma_f^2 = {\mathbf{x}^*}^T\boldsymbol{\Sigma}\mathbf{x}^*$$



from [Murphy, 2012]

Outline

Bayesian Neural Networks (BNN)



Credit: [Blundell et al., 2015]

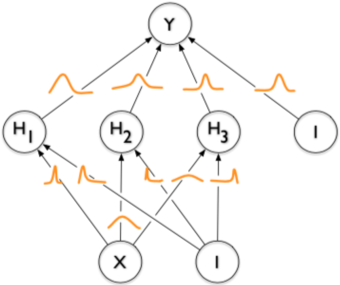
- $\boxed{\text{Standard NN: } \boldsymbol{y}_i = f^{\boldsymbol{w}}(\boldsymbol{x}_i), \textbf{ Bayesian NN: } p(\boldsymbol{y}_i|\boldsymbol{x}_i, \mathcal{D})}$
  - Define prior over weights, *e.g.* $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|0, \alpha^{-1}\mathcal{I})$ (point estimate for bias)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

Bayesian Neural Networks (BNN)

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

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

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



Credit: [Blundell et al., 2015]

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

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

- Monte Carlo estimation of the integral:

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

- Can't sample exactly from $p(\boldsymbol{w}|\mathcal{D})$, **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**
- **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 $p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{Y})$ cannot usually be evaluated analytically

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

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

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

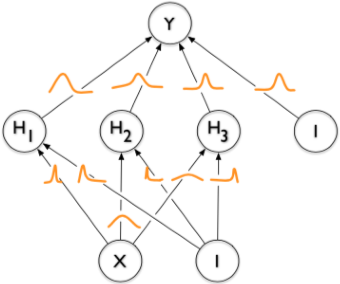
$$p(\boldsymbol{y}|\boldsymbol{x}^*, \boldsymbol{X}, \boldsymbol{Y}) \approx \int p(\boldsymbol{y}|\boldsymbol{x}^*, \boldsymbol{w}) q_{\boldsymbol{\theta}^*}(\boldsymbol{w}) d\boldsymbol{w}$$

Variational Inference (VI)

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

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

- 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

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

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

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

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

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

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

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

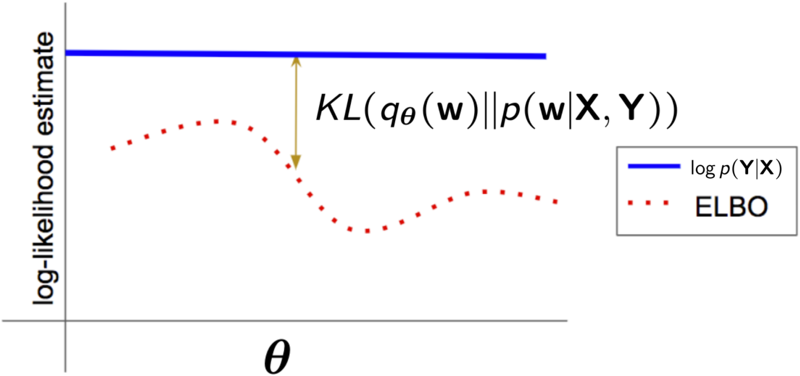
$p(\mathbf{w}|\mathbf{X},\mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X},\mathbf{w})p(\mathbf{w})}{p(\mathbf{Y}|\mathbf{X})}$

ELBO Illustration

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

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

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

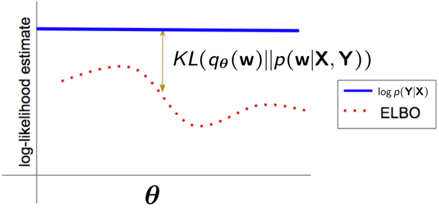


Variational Inference (VI): ELBO

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

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

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

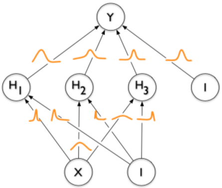


Variational Inference: Training

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

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

- RECAP: approximate variational posterior, *e.g.* fully factorize Gaussian:
  $q_{\boldsymbol{\theta}}(\boldsymbol{w}) = \prod_{i=1}^{D} \mathcal{N}(w_j | \mu_j, \sigma_j)$



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

Stochastic Variational Inference (VI): Training

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

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

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

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

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

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.

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

Solutions:

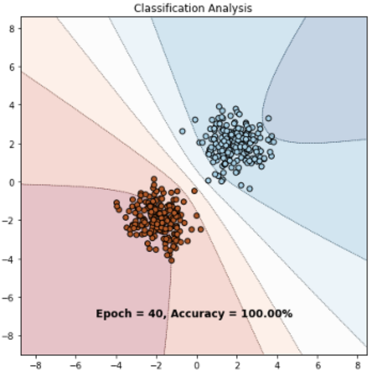
1. MC sampling, easy to sample from $q_\theta(\boldsymbol{w})$
2. Perform Taylor expansion of $y(\boldsymbol{x}^*, \boldsymbol{w})$ around $\boldsymbol{w}_{MAP}$ for regression[a]:
   $y(\boldsymbol{x}, \boldsymbol{w}) \approx y(\boldsymbol{x}, \boldsymbol{w}_{MAP}) + \frac{\partial y}{\partial \boldsymbol{w}}|_{\boldsymbol{w}=\boldsymbol{w}_{MAP}}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

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

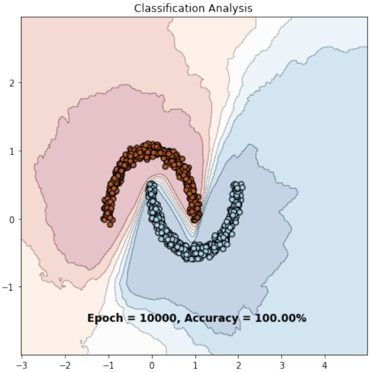
[a] For classification, logit Taylor expansion, see 5.7.1 in [Bishop, 2006]

Practical session

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



$p(y = 1|\boldsymbol{x}, \boldsymbol{X}, \boldsymbol{Y})$

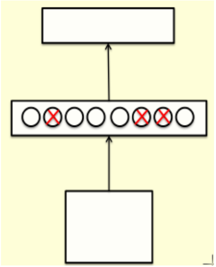


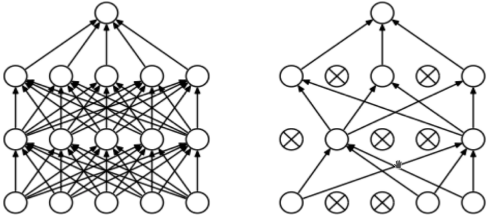
$p(y = 1|\boldsymbol{x}, \boldsymbol{X}, \boldsymbol{Y})$

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





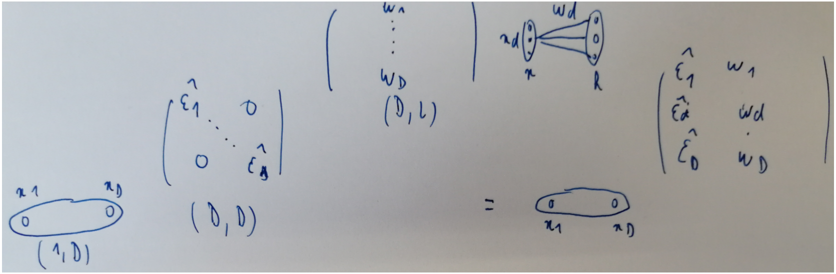
Standard Neural Net

After applying dropout.

Credits: Geoffrey E. Hinton, NIPS 2012

Dropout as a variational inference [Gal, 2016]

- Input $\boldsymbol{x} \in \mathbb{R}^{D\,a}$, latent vector $\boldsymbol{h} \in \mathbb{R}^{L}$
  - First layer: $\boldsymbol{h} = \sigma(\boldsymbol{x}\boldsymbol{W}_1)$, σ non-linearity
- **Dropout sampling:** in input : $\boldsymbol{x} \odot \hat{\boldsymbol{\varepsilon}}$
  - $\hat{\boldsymbol{\varepsilon}} = \left\{\hat{\varepsilon}_i^1\right\}_{i \in \{1;D\}}$ $\varepsilon_i \sim Bernoulli(1-p)$
  - First layer: $\boldsymbol{h} = \sigma((\boldsymbol{x} \odot \hat{\boldsymbol{\varepsilon}})\boldsymbol{W}_1)$
    - $(\boldsymbol{x}\,\mathrm{diag}(\hat{\varepsilon}))\boldsymbol{W}_1 = \boldsymbol{x}(\,\mathrm{diag}(\hat{\varepsilon})\boldsymbol{W}_1) = \boldsymbol{x}\hat{\boldsymbol{W}}_1$
    - Randomly setting to 0 rows of $\boldsymbol{W}_1$ (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), $\epsilon_i \sim Bernoulli(1 - p_i)$:
  - $\boxed{\boldsymbol{h}_1 = \sigma(\hat{\boldsymbol{x}}\mathrm{W}_1) = \sigma(\boldsymbol{x}\hat{\mathrm{W}}_1)}$, $\hat{\boldsymbol{W}}_1 = \mathrm{diag}(\hat{\epsilon_1})\mathrm{W}_1$
  - $\boxed{\hat{\boldsymbol{y}} =: f^{\hat{\boldsymbol{W}}_1, \hat{\boldsymbol{W}}_2}(\boldsymbol{x}) = \hat{\mathrm{h}_1}\mathrm{W}_2 = \mathrm{h}_1\hat{\mathrm{W}}_2}$, $\hat{\boldsymbol{W}}_2 = \mathrm{diag}(\hat{\epsilon_2})\mathrm{W}_2$ - $\hat{\boldsymbol{W}} = \{\hat{\mathrm{W}}_1; \hat{\mathrm{W}}_2\}$
- **MC Dropout sampling:** $\frac{1}{S}\sum\limits_{s \in S} p(\boldsymbol{y}_i | f^{\hat{\boldsymbol{W}}}(\boldsymbol{x}_i)) \approx \int p(\boldsymbol{y}|f^{\boldsymbol{W}}(\boldsymbol{x}^*)q(\boldsymbol{W})d\boldsymbol{w}$
  - $\forall$ layer $l \in \{1; L\}$, $\boldsymbol{W}_l$ random variable: $\boldsymbol{W}_l \sim q(\boldsymbol{W}_l) = g(\mathrm{M}_l, \boldsymbol{\varepsilon}_l) = \mathrm{diag}(\boldsymbol{\varepsilon_l})\mathrm{M}_l$
    - $\varepsilon_{l,i} \sim Bernoulli(1 - p_l)$, M_l deterministic parameters
  - $q(\boldsymbol{W}) = \prod\limits_{l=1}^{M} q(\boldsymbol{W}_l)$
- **Big result** (see next): **training NN with dropout** $\Leftrightarrow$ **training BNN with variational posterior approximation** $q_{\mathrm{M}}(\boldsymbol{W})$ (and some prior $p(\mathrm{W})$)
  - $\mathrm{M} = \{\mathrm{M}_l\}_{l \in \{1;L\}}$ variational parameters
- **MC dropout:** sampling several passes with dropout $\Leftrightarrow$ **performing MC approximate inference with variational posterior** $q_{\mathrm{M}}(\boldsymbol{W})$

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

Dropout as a variational inference [Gal, 2016]

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

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

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

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

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

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

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

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:

- $\boxed{p(\boldsymbol{W}) = \prod_l p(\boldsymbol{W}_l) = \prod_l \mathcal{MN}(\boldsymbol{W}_l; 0; I/l_l^2, I)}$ (prior factorized over layers)
- $\boxed{q(\boldsymbol{W}_l) = \mathrm{diag}(\hat{\boldsymbol{\varepsilon}}_{\boldsymbol{l}})\mathrm{M}_l,\ \varepsilon_{l,i} \sim Bernoulli(1-p_i),\ q(\boldsymbol{W}) = \prod_l q(\boldsymbol{W}_l)}$
  - 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:

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

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

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

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

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

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

Model uncertainty in regression

Proposition

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

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

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

Corollary

$Var_{q_{\boldsymbol{w}^*}(\boldsymbol{y}|\boldsymbol{x}^*)}[y]$ can be estimated with the unbiased estimator

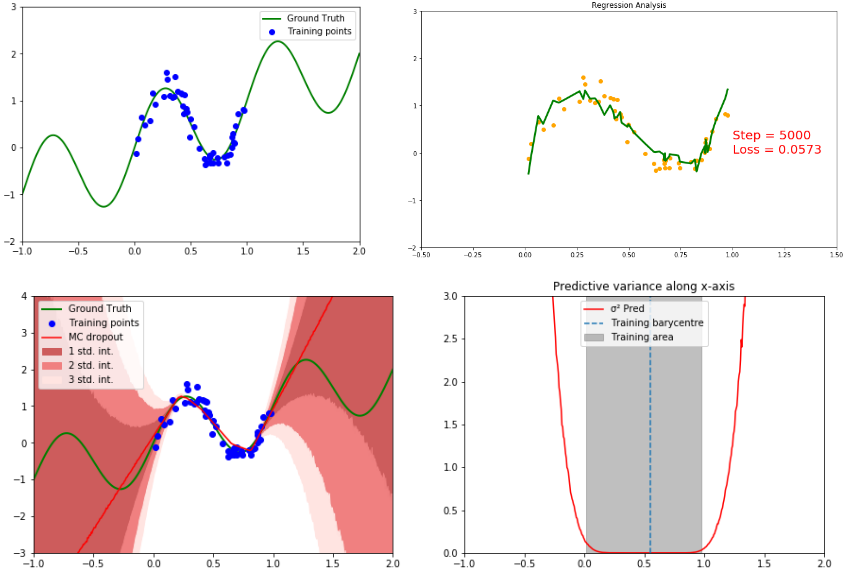
$$\widetilde{Var}[(y)] := \tau{-}1I + \frac{1}{T}\sum_{t=1}^{T} f^{\hat{\boldsymbol{W}}_t}(\boldsymbol{x}^*)^T f^{\hat{\boldsymbol{W}}_t}(\boldsymbol{x}^*) - \frac{1}{T}\left(\sum_{t=1}^{T} f^{\hat{\boldsymbol{W}}_t}\right)^T \left(\sum_{t=1}^{T} f^{\hat{\boldsymbol{W}}_t}\right)$$

$$\xrightarrow[T\to\infty]{} Var_{q_{\boldsymbol{w}^*}(\boldsymbol{y}|\boldsymbol{x}^*)}[y]$$

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

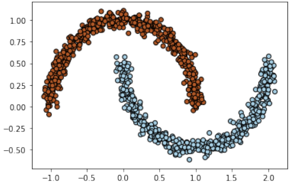
Application: MC dropout for predictive distribution

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



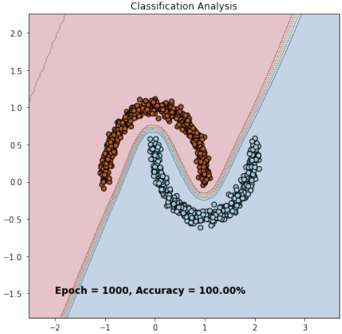
Application: MC dropout for predictive distribution

- MC dropout for non-linear classification

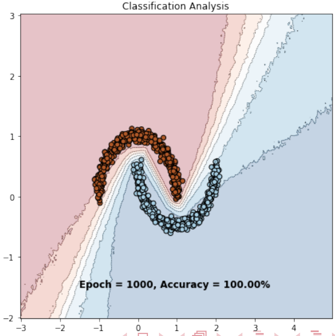


- As for BLR: $p(y=1|\boldsymbol{x}^*,\mathcal{D}) \approx \sum_{s=1}^{S} f_{\boldsymbol{w}^s}(\boldsymbol{x}^*)$, $\boldsymbol{w}^s \sim q(\boldsymbol{W})$

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.