Uncertainty Quantification - Bayesian Deep Models

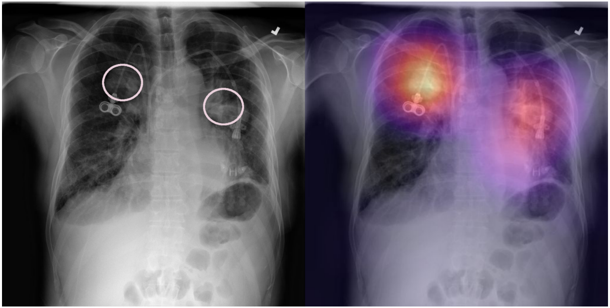
Clément Rambour

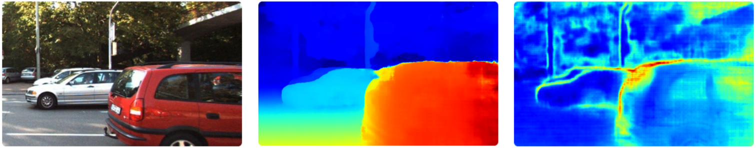
Sorbonne Université
ISIR - équipe MLIA

Robustness in Deep Learning

Deep Learning : huge gain in average performance, *e.g.* precision for classification

- Need for **performance certification in safety-critical applications : robustness**
  - Healthcare, autonomous steering, nuclear, defense, *etc*

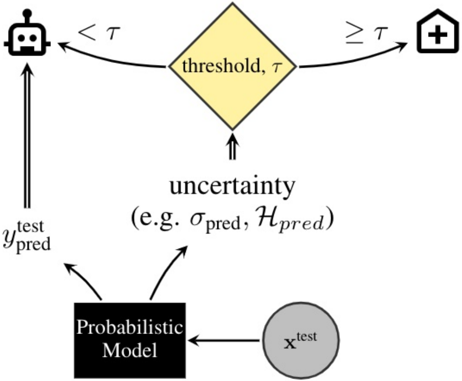




Robustness in Deep Learning

Performance certification

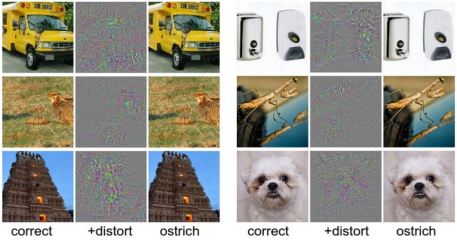
- **Reliable confidence / uncertainly estimation** of the decision process
  - "Know when you do not know"



Robustness in Deep Learning

Performance certification

- <u>**Stability of the decision function**</u>, *e.g.* robustness to adversarial attacks



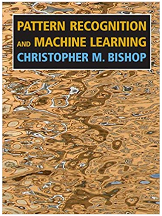
[Evtimov et al., 2017]

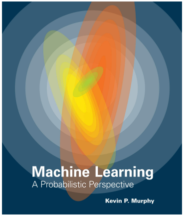
Robustness in Deep Learning : RDFIA Section 4

2 seances on robust deep learning

1. **Week 1 :** Bayesian models and uncertainty estimation
2. **Week 2 :** Application of uncertainty : failure prediction, OOD detection

- **References :**
  - Pattern Recognition and Machine Learning [Bishop, 2006]
  - Machine Learning : A Probabilistic Perspective [Murphy, 2012]





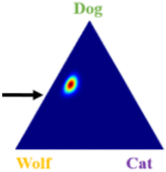
Outline

Type of uncertainties

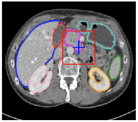
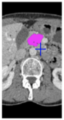
Aleatoric uncertainty

- *Aleator* (lat.) = dice player
- **Noise inherent in the observations**, *i.e.* natural randomness.
  - Inherent stochasticity, *e.g.* class confusion



  - Ambiguous data (*e.g.* medical images), or sensor quality

*Rain drops**


Lack of visual features


Glare

Type of uncertainties

Aleatoric uncertainty

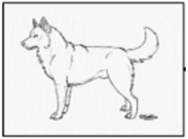
- *aleator* (lat.) = dice player
- **Noise inherent in the observations**, *i.e.* natural randomness.
- Cannot be reduced (need to change input/sensor), but can be estimated/learned
- Two (sub)-types of aleatoric uncertainty :
  1. **homoscedastic uncertainty** : stays constant for different input values, limited, captures 'average' uncertainty
  2. **heteroscedastic uncertainty** : depends on the input, learned from data

Type of uncertainties

Epistemic uncertainty

- *Episteme* (gr.) : knowledge $\Rightarrow$ **model uncertainty**
- Lack of knowledge about the generating process y (class) $\rightarrow$ input (image)



- **Main feature :** detects samples far from the training distribution
- Can be explained away given enough data
  - Epistemic uncertainty $\downarrow$ when number of data N (evidence) $\uparrow$
  - Epistemic uncertainty $\rightarrow 0$ when $N \rightarrow \infty$

Outline

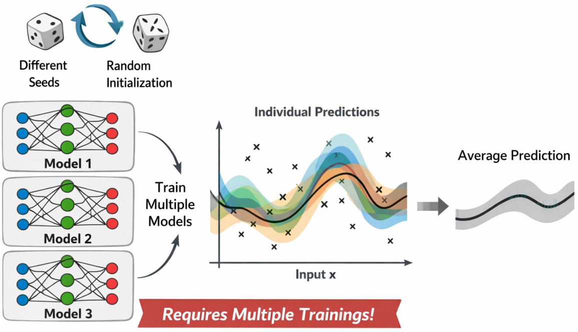
Uncertainty quantification

Intuition

How to assess the stability of a given model with respect to a training set ?

Naively :

- train multiple models with varying seed / initialization
- Average predictions on the input space to obtain an empirical distribution over the model outputs
  $\hookrightarrow$ Requires multiple training



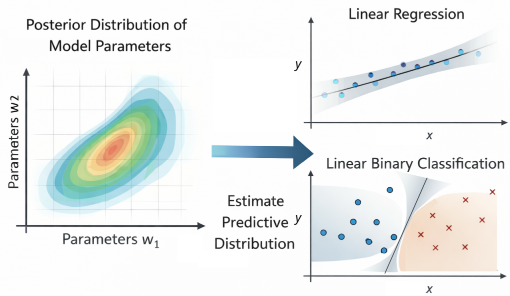
Bayesian Models

Intuition

How to assess the stability of a given model with respect to a training set ?

Bayesian approach :

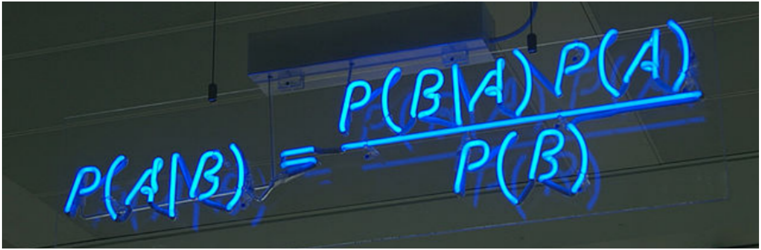
- Estimate the model distribution ⇔ the distribution of the model's parameters
- Estimate the predictive ditribution ⇔ distribution of the prediction in the input space



Bayesian Models

- Dataset $\mathcal{D} = (\mathbf{X}, \mathbf{Y})$ with inputs $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ and outputs $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^N$
  - $\mathbf{x}_i \in \mathbb{R}^d$, $\mathbf{y}_i^* \in \mathbb{R}^K$ (classification or regression)
  - Model with parameters $\mathbf{w}$: $\hat{\mathbf{y}}_i = f_{\mathbf{w}}(\mathbf{x}_i)$
- **Bayes rule :** $p(\mathbf{Y}, \mathbf{w}/\mathbf{X}) = p(\mathbf{Y}/\mathbf{X}, \mathbf{w})p(\mathbf{w}) = p(\mathbf{w}/\mathbf{X}, \mathbf{Y})p(\mathbf{Y}/\mathbf{X})$

$$\Rightarrow \boxed{p(\mathbf{w}/\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{Y}/\mathbf{X},\mathbf{w})p(\mathbf{w})}{p(\mathbf{Y}/\mathbf{X})} \propto p(\mathbf{Y}/\mathbf{X}, \mathbf{w})p(\mathbf{w})}$$



Bayesian Models

Deterministic *vs.* Bayesian learning

- **Deterministic models** learn the model's parameters $\mathbf{w}$ on the training set $\mathcal{D}$ that minimizes the expected loss $\mathcal{L}$:

$$\mathbf{w} = \underbrace{\operatorname{argmin}_{\mathbf{w}}}_{\substack{\text{find best} \\ \text{fixed parameters}}} \quad \underbrace{\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathcal{D})}}_{\substack{\text{data sampled} \\ \text{from } \mathcal{D}}} \quad \mathcal{L}\big(f_{\mathbf{w}}(\mathbf{x}), \mathbf{y}\big)$$

Bayesian Models

Deterministic *vs.* Bayesian learning

- **Deterministic models** learn the model's parameters $\mathbf{w}$ on the training set $\mathcal{D}$ that minimizes the expected loss $\mathcal{L}$:

$$\mathbf{w} = \underbrace{\underset{\mathbf{w}}{\text{argmin}}}_{\substack{\text{find best} \\ \underline{\text{fixed}}\ \text{parameters}}} \quad \underbrace{\underset{(\mathbf{x},\mathbf{y})\sim p(\mathcal{D})}{\mathbb{E}}}_{\substack{\text{data sampled} \\ \text{from } \mathcal{D}}} \quad \underbrace{-\log p(\mathbf{y}|\mathbf{x},\mathbf{w})}_{\substack{\text{loss=} \\ \text{negative log likelihood}}}$$

Bayesian Models

Deterministic *vs.* Bayesian learning

- **Deterministic models** learn the model's parameters $\mathbf{w}$ on the training set $\mathcal{D}$ that minimizes the expected loss $\mathcal{L}$:

$$\mathbf{w} = \underbrace{\operatorname*{argmin}_{\mathbf{w}}}_{\substack{\text{find best}\\ \text{fixed parameters}}} \underbrace{\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathcal{D})}}_{\substack{\text{data sampled}\\ \text{from } \mathcal{D}}} \underbrace{-\log p(\mathbf{y}|\mathbf{x},\mathbf{w})}_{\substack{\text{loss=}\\ \text{negative log likelihood}}}$$

- **Bayesian models** learn the distribution of $\mathbf{w}$ given the training set $p(\mathbf{w}|\mathcal{D})$. The best model $\mathbf{w}^*$ is called the maximum a posteriori (MAP) :

$$\mathbf{w}^* = \underbrace{\operatorname*{argmin}_{\mathbf{w}}}_{\substack{\text{find best parameters}\\ \text{from their distribution}}} -\log \underbrace{p(\mathbf{w}|\mathcal{D})}_{\substack{\text{parameters conditional}\\ \text{distribution}}}$$

Bayesian Models

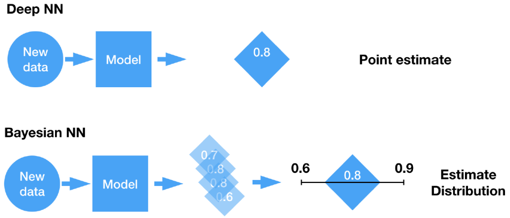
Deterministic *vs.* Bayesian learning

- **Deterministic models** learn the model's parameters $\mathbf{w}$ on the training set $\mathcal{D}$ that minimizes the expected loss $\mathcal{L}$:

$$\mathbf{w} = \underset{\mathbf{w}}{\text{argmin}} \underset{(\mathbf{x},\mathbf{y})\sim p(\mathcal{D})}{\mathbb{E}} \mathcal{L}(f_{\mathbf{w}}(\mathbf{x}), \mathbf{y})$$

- **Bayesian models** learn the distribution of $\mathbf{w}$ given the training set $p(\mathbf{w}|\mathcal{D})$. The best model $\mathbf{w}^*$ is called the maximum a posteriori (MAP) :

$$\mathbf{w}^* = \underset{\mathbf{w}}{\text{argmin}} \ -\log p(\mathbf{w}|\mathcal{D})$$



Bayesian Models & Uncertainty

Predictive distribution

From posterior $p(\mathbf{w}|\mathcal{D})$ $\Rightarrow$ compute **predictive distribution** given new input $\mathbf{x}^*$:

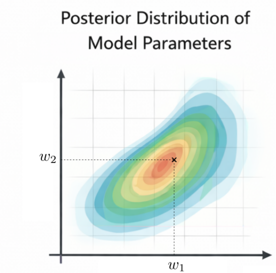
$$\forall y, \quad p(\mathbf{y}|\mathbf{x}^*, \mathcal{D}) = \int \boxed{p(\mathbf{y}, \mathbf{w}|\mathbf{x}^*, \mathcal{D})} d\mathbf{w} \quad \text{marginalization}$$

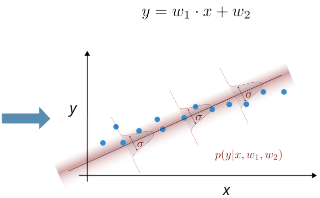
Bayes

$$= \int \boxed{p(\mathbf{y}|\mathbf{x}^*, \mathbf{w})} \; \boxed{p(\mathbf{w}|\mathcal{D})} d\mathbf{w} \quad \text{Bayes rule}$$

$$= \mathbb{E}_{p(\mathbf{w}|\mathcal{D})}[p(y|\mathbf{x}^*, \mathbf{w})] \quad \text{expectation definition}$$

Naturally gives a measure of uncertainty





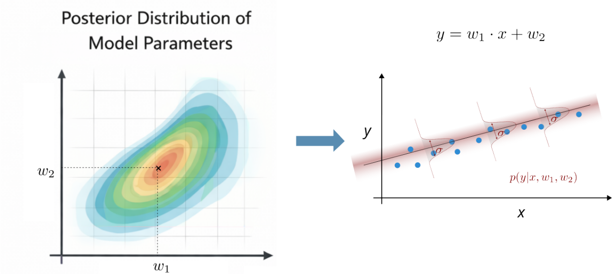
Bayesian Models & Uncertainty

Predictive distribution

From posterior $p(\mathbf{w}|\mathcal{D}) \Rightarrow$ compute **predictive distribution** given new input $\mathbf{x}^*$:

$$
\begin{aligned}
\forall y, \quad p(\mathbf{y}|\mathbf{x}^*, \mathcal{D}) &= \int \boxed{p(\mathbf{y}, \mathbf{w}|\mathbf{x}^*, \mathcal{D})} d\mathbf{w} && \text{marginalization} \\
& \qquad \text{Bayes} \\
&= \int \boxed{p(\mathbf{y}|\mathbf{x}^*, \mathbf{w})} \; \boxed{p(\mathbf{w}|\mathcal{D})} d\mathbf{w} && \text{Bayes rule} \\
&= \mathbb{E}_{p(\mathbf{w}|\mathcal{D})}\left[p(y|\mathbf{x}^*, \mathbf{w})\right] && \text{expectation definition}
\end{aligned}
$$

Naturally gives a measure of uncertainty



Bayesian Models & Uncertainty

Predictive distribution

From posterior $p(\mathbf{w}|\mathcal{D})$ $\Rightarrow$ compute **predictive distribution** given new input $\mathbf{x}^*$:

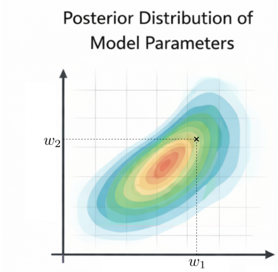
$$\forall y, \quad p(\mathbf{y}|\mathbf{x}^*, \mathcal{D}) = \int \boxed{p(\mathbf{y}, \mathbf{w}|\mathbf{x}^*, \mathcal{D})} d\mathbf{w} \quad \text{marginalization}$$

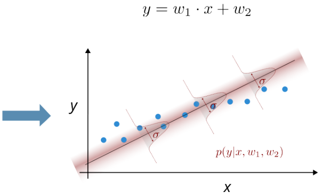
Bayes

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

$$= \mathbb{E}_{p(\mathbf{w}|\mathcal{D})} [p(y|\mathbf{x}^*, \mathbf{w})] \quad \text{expectation definition}$$

Naturally gives a measure of uncertainty





Bayesian Models & Uncertainty

Predictive distribution

From posterior $p(\mathbf{w}|\mathcal{D})$ $\Rightarrow$ compute **predictive distribution** given new input $\mathbf{x}^*$:

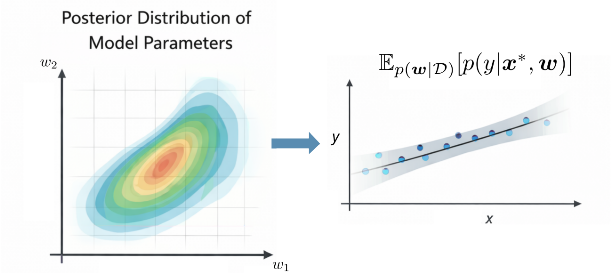
$$\forall y, \quad p(\mathbf{y}|\mathbf{x}^*, \mathcal{D}) = \int \boxed{p(\mathbf{y}, \mathbf{w}|\mathbf{x}^*, \mathcal{D})} d\mathbf{w} \quad \text{marginalization}$$

Bayes

$$= \int \boxed{p(\mathbf{y}|\mathbf{x}^*, \mathbf{w})} \; \boxed{p(\mathbf{w}|\mathcal{D})} d\mathbf{w} \quad \text{Bayes rule}$$

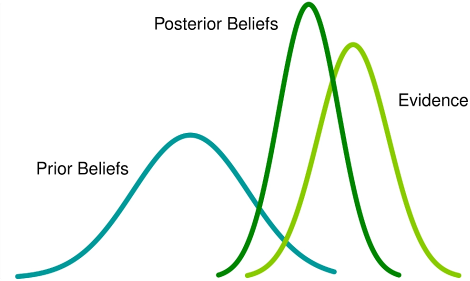
$$= \mathbb{E}_{p(\mathbf{w}|\mathcal{D})}\left[p(y|\mathbf{x}^*, \mathbf{w})\right] \quad \text{expectation definition}$$

Naturally gives a measure of uncertainty



Bayesian Models & Uncertainty

- RECAP : for uncertainty estimate with Bayesian models
  1. Define prior $p(\mathbf{w})$ and likelihood $p(\mathbf{Y}/\mathbf{X}, \mathbf{w})$
  2. Compute posterior distribution $p(\mathbf{w}/\mathbf{X}, \mathbf{Y})$
  3. Compute predictive distribution $p(\mathbf{y}^*/\mathbf{x}^*, \mathbf{Y}, \mathbf{X})$
- Easy ? NO ! ! $\Rightarrow$ steps 2 and 3 computationally hard in general !
  - Typically no closed form for step 2
  - High-dimensional integration for step 3



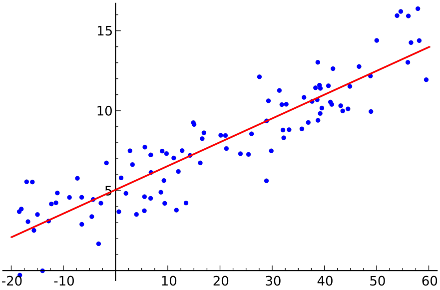
Probabilistic Linear Regression

- N training examples $(\mathbf{x}_i, \mathbf{y}_i)_{i \in \{1;N\}}$; $\mathbf{x}_i \in \mathbb{R}^p$, $\mathbf{y}_i \in \mathbb{R}^K$
- Matrix notation including bias in $\mathbf{w}$: $\mathbf{\Phi}$ of size $N \times (p+1)$

$$\mathbf{\Phi} = \begin{pmatrix} 1 & x_{11} & \dots & x_{1p} \\ 1 & x_{21} & \dots & x_{2p} \\ \dots & \dots & \dots & \dots \\ 1 & x_{N1} & \dots & x_{Np} \end{pmatrix}$$

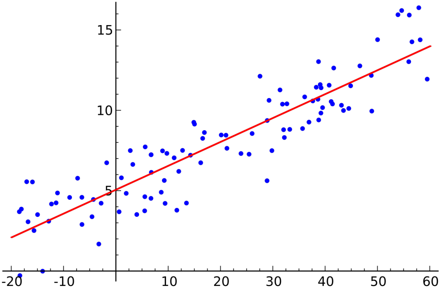
- $\boxed{\mathbf{Y} = \mathbf{\Phi}\mathbf{w} + \boldsymbol{\varepsilon}}$, $\boldsymbol{\varepsilon} \in \mathbb{R}^{N \times K}$, $\varepsilon_{i,k} \sim \mathcal{N}(0, \sigma^2)$
- $\mathbf{Y} \in \mathbb{R}^{N \times K}$, $\mathbf{\Phi} \in \mathbb{R}^{N \times (p+1)}$, $\mathbf{w} \in \mathbb{R}^{(p+1) \times K}$
  - $\mathbf{\Phi}_i^T := \begin{pmatrix} 1 & x_{i1} & \dots & x_{ip} \end{pmatrix}^T \in \mathbb{R}^{1 \times (p+1)}$
  - $\mathbf{y}_i = \mathbf{\Phi}_i^T \mathbf{w} + \varepsilon_i$, $\mathbf{y}_i \in \mathbb{R}^{1 \times K}$, $\varepsilon_i \in \mathbb{R}^{1 \times K}$

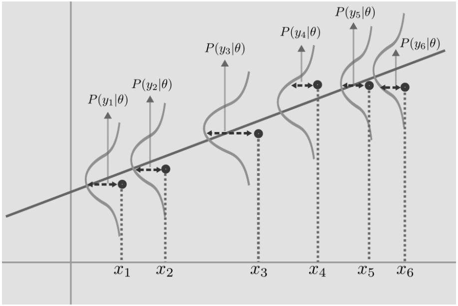
- Ex : scalar inputs and outputs $p = 1$, $K = 1 \Rightarrow \mathbf{\Phi} \in \mathbb{R}^{N \times 2}$, $\mathbf{w} \in \mathbb{R}^2$:



Probabilistic Linear Regression

- N training examples $(\mathbf{x}_i, \mathbf{y}_i)$ $\mathbf{y}_i = \mathbf{\Phi}_i^T \mathbf{w} + \varepsilon_i$, with $\epsilon_i \sim \mathcal{N}(0,1)$
- $p(\mathbf{y}_i / \mathbf{x}_i, \mathbf{w}) \sim \mathcal{N}(\mathbf{\Phi}_i^T \mathbf{w}, \sigma^2)$ or $p(\mathbf{y}_i - \mathbf{\Phi}_i^T \mathbf{w}) \sim \mathcal{N}(0, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma}\, e^{\frac{(\mathbf{y}_i - \mathbf{\Phi}_i^T \mathbf{w})^2}{2\sigma^2}}$
  - $p(\mathbf{y}_i / \mathbf{x}_i, \mathbf{w})$: likelihhod, $\sigma \sim$ aleatoric uncertainty
  - σ independent of $\mathbf{x} \Rightarrow$ homoscedastic uncertainty





Probabilistic Linear Regression

- Trained with Maximum Likelihood (ML)
  - Examples assumed to be *i.i.d* (independent and identically distributed)
    $\Rightarrow p(\mathbf{Y}/\mathbf{X}, \mathbf{w}, \sigma) = \prod_{i=1}^{N} p(y_i/x_i, \mathbf{w}, \sigma)$
  - MLE : $(\hat{\mathbf{w}}, \hat{\sigma}) = \arg\max_{(\mathbf{w},\sigma)} p(\mathbf{X}, \mathbf{Y}/\mathbf{w}, \sigma) = \arg\min_{\mathbf{w},\sigma} - \sum_{i=1}^{N} log\left[p(\mathbf{y}_i/\mathbf{x}_i, \mathbf{w})\right]$

Probabilistic Linear Regression

- Trained with Maximum Likelihood (ML)
  - Examples assumed to be *i.i.d* (independent and identically distributed)
    $\Rightarrow p(\mathbf{Y}/\mathbf{X},\mathbf{w},\sigma) = \prod_{i=1}^{N} p(y_i/x_i,\mathbf{w},\sigma)$
  - MLE : $(\hat{\mathbf{w}},\hat{\sigma}) = \arg\max_{(\mathbf{w},\sigma)} p(\mathbf{X},\mathbf{Y}/\mathbf{w},\sigma) = \arg\min_{\mathbf{w},\sigma} \; - \sum_{i=1}^{N} log\left[p(\mathbf{y}_i/\mathbf{x}_i,\mathbf{w})\right]$
- MLE solution $\mathbf{w}$: $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} C(\mathbf{w}) = \arg\min_{\mathbf{w}} \sum_{i=1}^{N} (\mathbf{y}_i - \mathbf{\Phi}_i^T \mathbf{w})^2$

  $\Rightarrow$ Ordinary least square problem (closed form) :
  - $C(\mathbf{w}) = ||\mathbf{Y} - \mathbf{\Phi w})||^2 = (\mathbf{Y} - \mathbf{\Phi w})^T(\mathbf{Y} - \mathbf{\Phi w}),\ \nabla_{\mathbf{w}} C = 2\mathbf{\Phi}^T(Y - \mathbf{\Phi w})$
  - $\nabla_{\mathbf{w}} C = 0 \Leftrightarrow \mathbf{\Phi}^T\mathbf{\Phi w}^{=}\mathbf{\Phi}^T\mathbf{Y}$

$$\boxed{\hat{\mathbf{w}} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{Y}}$$

Probabilistic Linear Regression

- Trained with Maximum Likelihood (ML)
  - Examples assumed to be *i.i.d* (independent and identically distributed)
    $\Rightarrow p(\mathbf{Y}/\mathbf{X},\mathbf{w},\sigma) = \prod_{i=1}^{N} p(y_i/x_i,\mathbf{w},\sigma)$
  - MLE : $(\hat{\mathbf{w}},\hat{\sigma}) = \arg\max_{(\mathbf{w},\sigma)} p(\mathbf{X},\mathbf{Y}/\mathbf{w},\sigma) = \arg\min_{\mathbf{w},\sigma} \; -\sum_{i=1}^{N} log\left[p(\mathbf{y}_i/\mathbf{x}_i,\mathbf{w})\right]$
- MLE solution $\mathbf{w}$: $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} C(\mathbf{w}) = \arg\min_{\mathbf{w}} \sum_{i=1}^{N} (\mathbf{y}_i - \mathbf{\Phi}_i^T \mathbf{w})^2$

  $\Rightarrow$ Ordinary least square problem (closed form) :
  - $C(\mathbf{w}) = ||\mathbf{Y} - \mathbf{\Phi}\mathbf{w})||^2 = (\mathbf{Y} - \mathbf{\Phi}\mathbf{w})^T(\mathbf{Y} - \mathbf{\Phi}\mathbf{w}),\ \nabla_{\mathbf{w}} C = 2\mathbf{\Phi}^T(Y - \mathbf{\Phi}\mathbf{w})$
  - $\nabla_{\mathbf{w}} C = 0 \Leftrightarrow \mathbf{\Phi}^T\mathbf{\Phi}\mathbf{w}^{=}\mathbf{\Phi}^T\mathbf{Y}$

$$\boxed{\hat{\mathbf{w}} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{Y}}$$

- ML solution σ : $\arg\min_{\sigma}[\ N\ log(\sigma) + \frac{1}{2\sigma^2} \sum_{i=1}^{N} (\mathbf{y}_i - \mathbf{\Phi}_i^T \mathbf{w})^2]$
  - Closed form solution : $\hat{\sigma^2} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{y}_i - \mathbf{\Phi}_i^T \mathbf{w})^2 \Rightarrow$ interpretation : data std

Limits of MLE

- Learning model to predict coin toss with MLE
  - Bernoulli variable $\mathbf{X}$ with param p : $P(\mathbf{x}|p) = \prod_{i=1}^{N} P(x_i|p) = \prod_{i=1}^{N} p^{x_i}(1-p)^{1-x_i}$
  - MLE : $\ln P(\mathbf{x}|p) = \sum_{i=1}^{N} [x_i \ln p + (1-x_i)\ln(1-p)] \Rightarrow \boxed{p_{MLE} = \frac{1}{N}\sum_{i=1}^{N} x_i}$
  - MLE : predict $P(X|p_{MLE}) = 1$ for all futures tosses !

- Using prior knowledge on p, *e.g.* $P(p) = 0.5$ or $P(p) = 0.3 \Rightarrow$ MAP

Bayesian Linear Regression

$$\boxed{\text{Include prior } p(\mathbf{w}) \Rightarrow \text{biased posterior } p(\mathbf{w}/\mathbf{X},\mathbf{Y}) \text{ (MAP)}}$$

Bayesian Linear Regression

$$\boxed{\text{Include prior } p(\mathbf{w}) \Rightarrow \text{biased posterior } p(\mathbf{w}/\mathbf{X},\mathbf{Y}) \text{ (MAP)}}$$

- Choose **Prior**, *e.g.* $p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}; 0, \alpha^{-1}I)$

Bayesian Linear Regression

$$\boxed{\text{Include prior } p(\mathbf{w}) \Rightarrow \text{biased posterior } p(\mathbf{w}/\mathbf{X},\mathbf{Y}) \text{ (MAP)}}$$

- Choose **Prior**, *e.g.* $p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}; 0, \alpha^{-1}I)$
- **Likelihood**, as before : $p(\mathbf{y}_i/\mathbf{x}_i, \mathbf{w}) = \mathcal{N}(\mathbf{\Phi}_i^T \mathbf{w}, \beta^{-1})$ with $\beta = \frac{1}{2\sigma^2}$
  $\Rightarrow$ **Compute posterior** $p(\mathbf{w}/\mathbf{x}_i, \mathbf{y}_i) \propto p(\mathbf{y}_i/\mathbf{x}_i, \mathbf{w})p(\mathbf{w})$

Bayesian Linear Regression

$$\boxed{\text{Include prior } p(\mathbf{w}) \Rightarrow \text{biased posterior } p(\mathbf{w}/\mathbf{X},\mathbf{Y}) \text{ (MAP)}}$$

- Choose **Prior**, *e.g.* $p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}; 0, \alpha^{-1}I)$
- **Likelihood**, as before : $p(\mathbf{y}_i/\mathbf{x}_i, \mathbf{w}) = \mathcal{N}(\mathbf{\Phi}_i^T\mathbf{w}, \beta^{-1})$ with $\beta = \frac{1}{2\sigma^2}$
  $\Rightarrow$ **Compute posterior** $p(\mathbf{w}/\mathbf{x}_i, \mathbf{y}_i) \propto p(\mathbf{y}_i/\mathbf{x}_i, \mathbf{w})p(\mathbf{w})$
  - Here, prior same form as likelihood (Gaussian), *i.e.* **Prior conjugate to likelihood**
    $\Rightarrow$ **Posterior as the same form as the prior**
    $\Rightarrow$ **Closed-form for the posterior, which is also Gaussian !**

Bayesian Linear Regression

$$\boxed{\text{Include prior } p(\mathbf{w}) \Rightarrow \text{biased posterior } p(\mathbf{w}/\mathbf{X},\mathbf{Y}) \text{ (MAP)}}$$

- Choose **Prior**, *e.g.* $p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}; 0, \alpha^{-1}I)$
- **Likelihood**, as before : $p(\mathbf{y}_i/\mathbf{x}_i, \mathbf{w}) = \mathcal{N}(\mathbf{\Phi}_i^T\mathbf{w}, \beta^{-1})$ with $\beta = \frac{1}{2\sigma^2}$
  $\Rightarrow$ **Compute posterior** $p(\mathbf{w}/\mathbf{x}_i, \mathbf{y}_i) \propto p(\mathbf{y}_i/\mathbf{x}_i, \mathbf{w})p(\mathbf{w})$
  - Here, prior same form as likelihood (Gaussian), *i.e.* **Prior conjugate to likelihood**
    $\Rightarrow$ **Posterior as the same form as the prior**
    $\Rightarrow$ **Closed-form for the posterior, which is also Gaussian !**
  - With $p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}; 0, \alpha^{-1}I)$ and $p(\mathbf{y}_i/\mathbf{x}_i, \mathbf{w}) = \mathcal{N}(\mathbf{\Phi}_i^T\mathbf{w}, \beta^{-1})$, we can show that :

$$\boxed{\begin{aligned} p(\mathbf{w}/\mathbf{X},\mathbf{Y}) &= \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}, \mathbf{\Sigma}) \\ \mathbf{\Sigma}^{-1} &= \alpha \boldsymbol{I} + \beta \mathbf{\Phi}^T \mathbf{\Phi} \\ \boldsymbol{\mu} &= \beta \mathbf{\Sigma} \mathbf{\Phi}^T \mathbf{Y} \end{aligned}}$$

- Closed form solution for MAP ($\boldsymbol{\mu}$, median $\leftrightarrow$ mode)
- $\alpha \to 0 \Rightarrow$ recover ML ; $N = 0 \Rightarrow$ recover prior

Bayesian Linear Regression

Come from general results [Bishop, 2006]

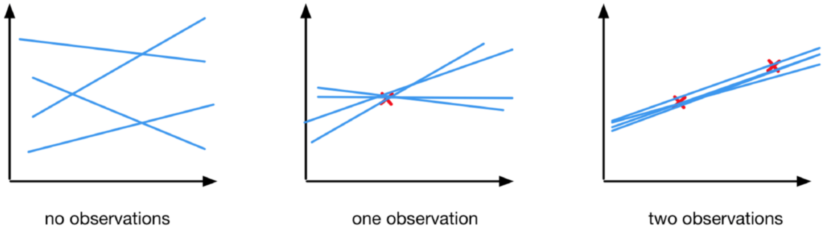
- Assume that :
  - $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$
  - $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|A\mathbf{x} + b, \boldsymbol{\Sigma}_y)$
- Then : $p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{x|\mathbf{y}}, \boldsymbol{\Sigma}_{x|\mathbf{y}})$, with :

$$\boldsymbol{\Sigma}_{\boldsymbol{x}|\mathbf{y}}^{-1} = \boldsymbol{\Sigma}_x^{-1} + \boldsymbol{A}^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{A}$$

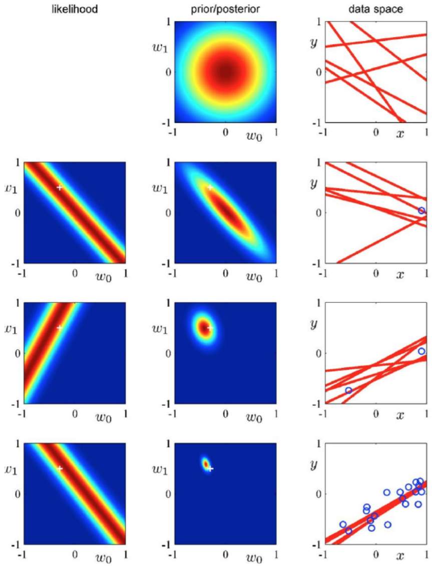
$$\boldsymbol{\mu}_{x|\mathbf{y}} = \boldsymbol{\Sigma}_{\boldsymbol{x}|\mathbf{y}}[A^T \boldsymbol{\Sigma}_y^{-1}(\mathbf{y} - b) + \boldsymbol{\Sigma}_x^{-1} \boldsymbol{\mu}_x]$$

Bayesian Linear Regression : Posterior Sampling

- $p(\mathbf{w}/\mathbf{X},\mathbf{Y}) = \mathcal{N}(\mathbf{w};\boldsymbol{\mu},\boldsymbol{\Sigma}) \Rightarrow$ we can sample from posterior
  - No observation : $p(\mathbf{w}/\mathbf{X},\mathbf{Y}) = p(\mathbf{w})$
  - $N \geq 1$: prior biased by data likelihood
  - Ex : $p(\mathbf{w}|\mathbf{x}_0,\mathbf{y}_0) \propto p(\mathbf{w})p(\mathbf{y}_0|\mathbf{x}_0,\mathbf{w})$
  - $p(\mathbf{w}|(\mathbf{x}_0,\mathbf{y}_0),(\mathbf{x}_1,\mathbf{y}_1)) \propto p(\mathbf{w}|\mathbf{x}_0,\mathbf{y}_0)p(\mathbf{y}_1|\mathbf{x}_1,\mathbf{w}) \propto p(\mathbf{w})p(\mathbf{y}_0|\mathbf{x}_0,\mathbf{w})p(\mathbf{y}_1|\mathbf{x}_1,\mathbf{w})$ ....



Bayesian Linear Regression : Posterior Sampling



0 data points are observed.

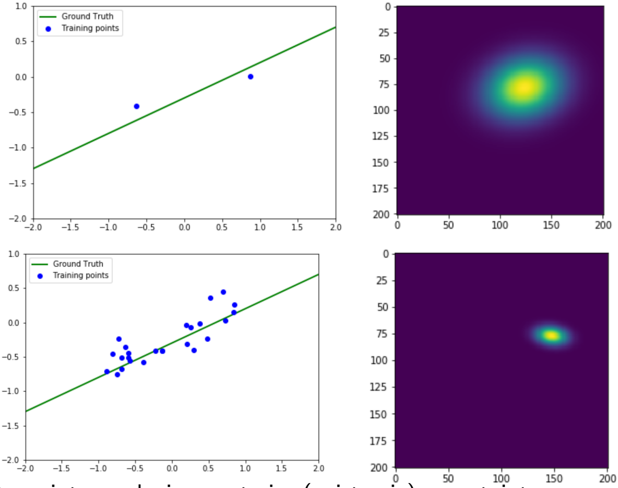
1 data point is observed.

2 data points are observed.

20 data points are observed.

Bayesian Linear Regression : Posterior Sampling

Practical session : compute posterior



- More data points : reducing posterior (epistemic) uncertainty
- $N \to \infty$ posterior uncertainty $\Leftrightarrow$ aleatoric uncertainty

Bayesian Linear Regression : Predictive Distribution

$p(\mathbf{w}|\mathcal{D},\alpha,\beta) \Rightarrow$ compute predictive distribution by marginalizing over $\mathbf{w}$:

- $p(y|\mathbf{x}^*,\mathcal{D},\alpha,\beta) = \int p(y|\mathbf{x}^*,\mathbf{w},\beta)p(\mathbf{w}|\mathcal{D},\alpha,\beta)dw$
  - $p(y|\mathbf{x}^*,\mathbf{w},\beta) = \mathcal{N}(y;\mathbf{\Phi}(\mathbf{x}^*)^T\mathbf{w},\beta^{-1})$: likelihood
  - $p(\mathbf{w}|\mathcal{D},\alpha,\beta) = \mathcal{N}(\mathbf{w};\boldsymbol{\mu},\mathbf{\Sigma})$: $\mathbf{w}$ posterior
    - $\mathbf{\Sigma}^{-1} = \alpha \boldsymbol{I} + \beta\mathbf{\Phi}^T\mathbf{\Phi}$
    - $\boldsymbol{\mu} = \beta\mathbf{\Sigma}\mathbf{\Phi}^T\mathbf{Y}$
- $p(y|\mathbf{x}^*,\mathcal{D},\alpha,\beta)$: convolution of two Gaussians $\Rightarrow$ Gaussian
  - Mean of predictive distribution $\mu^T\mathbf{\Phi}(\mathbf{x}^*)$
  - Variance of predictive distribution $\sigma^2_{pred}(\mathbf{x}^*) = \frac{1}{\beta} + \mathbf{\Phi}(\mathbf{x}^*)^T\mathbf{\Sigma}\mathbf{\Phi}(\mathbf{x}^*)$

$$\boxed{p(y|\mathbf{x}^*,\mathcal{D},\alpha,\beta) = \mathcal{N}(y;\mu^T\mathbf{\Phi}(\mathbf{x}^*),\frac{1}{\beta} + \mathbf{\Phi}(\mathbf{x}^*)^T\mathbf{\Sigma}\mathbf{\Phi}(\mathbf{x}^*))}$$

Bayesian Linear Regression : Predictive Distribution

$$\boxed{p(y|\mathbf{x}^*, \mathcal{D}, \alpha, \beta) = \mathcal{N}(y; \mu^T \mathbf{\Phi}(\mathbf{x}^*), \frac{1}{\beta} + \mathbf{\Phi}(\mathbf{x}^*)^T \mathbf{\Sigma} \mathbf{\Phi}(\mathbf{x}^*))}$$

- $\sigma^2_{pred}(\mathbf{x}^*) = \frac{1}{\beta} + \mathbf{\Phi}(\mathbf{x}^*)^T \mathbf{\Sigma} \mathbf{\Phi}(\mathbf{x}^*)$
- β is actually our noise representation (**aleatoric**)
- $\mathbf{\Phi}(\mathbf{x}^*)^T \mathbf{\Sigma} \mathbf{\Phi}(\mathbf{x}^*)$ is uncertainty over parameters $\mathbf{w}$ (**epistemic**)
- $\sigma^2_{pred}(\mathbf{x}^*)$ actually depends on N, $\sigma^2_{pred}(\mathbf{x}^*, N)$
  - $\sigma^2_{pred}(\mathbf{x}^*, N+1)$ ¡ $\sigma^2_{pred}(\mathbf{x}^*, N)$
  - $\lim_{N \to \infty} \mathbf{\Phi}(\mathbf{x}^*)^T \mathbf{\Sigma} \mathbf{\Phi}(\mathbf{x}^*) = 0$: epistemic uncertainty removed by adding data samples

Bayesian Linear Regression : Predictive Distribution

$$p(y|\mathbf{x}^*, \mathcal{D}, \alpha, \beta) = \mathcal{N}(y; \mu^T \mathbf{\Phi}(\mathbf{x}^*), \sigma^2_{pred}(\mathbf{x}^*))$$

- $\sigma^2_{pred}(\mathbf{x}^*) = \beta^{-1} + \mathbf{\Phi}(\mathbf{x}^*)^T \mathbf{\Sigma} \mathbf{\Phi}(\mathbf{x}^*)$
- $\mathbf{\Phi}(\mathbf{x}^*)^T \mathbf{\Sigma} \mathbf{\Phi}(\mathbf{x}^*)$ is uncertainty over parameters $\mathbf{w}$ **(epistemic)**
- **Practical session :** in case of 1D inputs & outputs, *i.e.* $x_i \in \mathbb{R}$, $\mathbf{X} \in \mathbb{R}^{N \times 1}$, $\mathbf{X} \in \mathbb{R}^{N \times 1}$, $\mathbf{w} \in \mathbb{R}^2$, $\mathbf{\Phi} \in \mathbb{R}^{N \times 2}$:

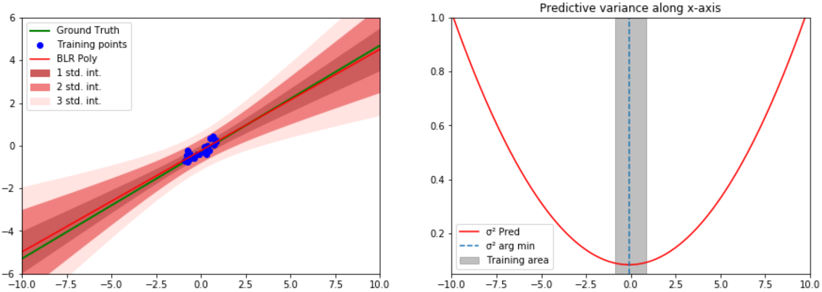
$$\mathbf{\Sigma}^{-1} = \alpha \boldsymbol{I} + \beta \mathbf{\Phi}^T \mathbf{\Phi} = \begin{pmatrix} \alpha + \beta N & \beta \mathbf{1}^T \mathbf{X} \\ \beta \mathbf{1}^T \mathbf{X} & \alpha + \beta \mathbf{X}^T \mathbf{X} \end{pmatrix}$$

$\Rightarrow$ $\mathbf{\Phi}(\mathbf{x}^*)^T \mathbf{\Sigma} \mathbf{\Phi}(\mathbf{x}^*)$ **Increases when $\mathbf{x}^*$ far from training data**

- $\mathbf{x}^*_{min} = \frac{\sum_i x_i}{N + \alpha / \beta}$

Bayesian Linear Regression : Predictive Distribution

Practical session : predictive distribution and uncertainty



Bayesian Linear Regression

- Note on MAP estimate :

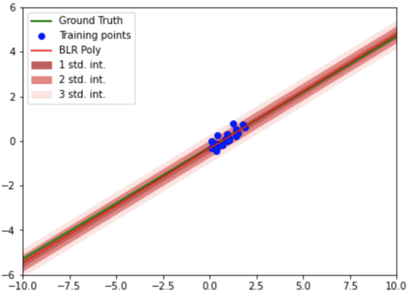
$$
\begin{aligned}
\mathbf{w}_{MAP} &= \arg\min_{\mathbf{w}} - \sum_{n=1}^{N} \log p(\mathbf{w}|\mathbf{x}_n, y_n, \beta, \alpha) \\
&= \arg\min_{\mathbf{w}} - \sum_{n=1}^{N} \log p(y_n|\mathbf{x}_n, \mathbf{w}, \beta, \alpha) - \log p(\mathbf{w}|\alpha) \\
&= \arg\min \frac{\beta}{2} \sum_{i=1}^{N} ||y_n - f^{\mathbf{w}}(\mathbf{x}_n)||^2 + \frac{\alpha}{2} \mathbf{w}^T \mathbf{w}
\end{aligned}
$$

$\Rightarrow$ **adding a Gaussian prior with precision on weights α acts like L_2 regularisation (weight decay) with $\lambda = \alpha/\beta$**

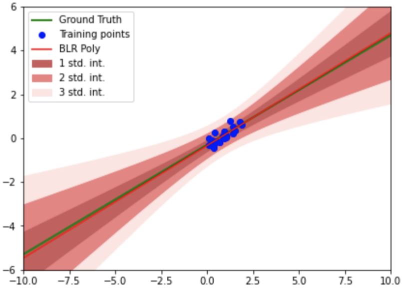
Bayesian Linear Regression : Predictive Distribution

Prediction distribution $\neq$ likelihood at $w = w_{MAP}$

- **Likelihood at** $w = w_{MAP}$: $p(y|\mathbf{x}^*, w_{MAP}) = \mathcal{N}(y; \mu^T \mathbf{\Phi}(\mathbf{x}^*), \sigma^2))$
  - $\sigma^2 = Cte \quad \forall \mathbf{x}^*$
- **Prediction distribution :** $p(y|\mathbf{x}^*, \mathcal{D}, \alpha, \beta) = \int p(y|\mathbf{x}^*, \mathbf{w}, \beta) p(\mathbf{w}|\mathcal{D}, \alpha, \beta) dw = \mathcal{N}(y; \mu^T \mathbf{\Phi}(\mathbf{x}^*), \sigma^2_{pred}(\mathbf{x}^*))$
  - $\sigma^2_{pred} = f(\mathbf{x}^*) = \beta^{-1} + \mathbf{\Phi}(\mathbf{x}^*)^T \mathbf{\Sigma} \mathbf{\Phi}(\mathbf{x}^*)$



$p(y|\mathbf{x}^*, w_{MAP})$

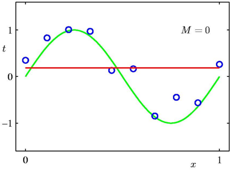


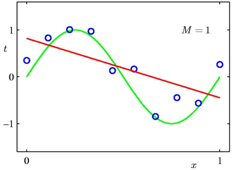
$p(y|\mathbf{x}^*, \mathcal{D}, \alpha, \beta)$

Non-Linear Regression

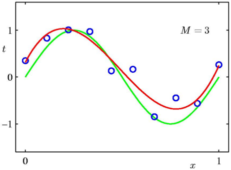
- Linear regression : limited in many datasets
- Non-linear extension by designing explicit non-linear feature maps $\mathbf{\Phi}$
  - Ex : Polynomial regression for 1D input, *i.e.* $x_i \in \mathbb{R}$:

$$\mathbf{\Phi} = \begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^M \\ 1 & x_2 & x_2^2 & \dots & x_2^M \\ \dots & \dots & \dots & \dots & \\ 1 & x_N & x_N^2 & \dots & x_N^M \end{pmatrix} \qquad \mathbf{Y} = \mathbf{\Phi} \mathbf{w} + \boldsymbol{\varepsilon} \qquad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 & \dots & \varepsilon_N \end{pmatrix}^T, \ \varepsilon_i \sim \mathcal{N}(0, \sigma^2)$$

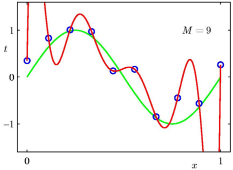




← Poor representations of $\sin(2\pi x)$



← Best Fit to $\sin(2\pi \mathrm{x})$



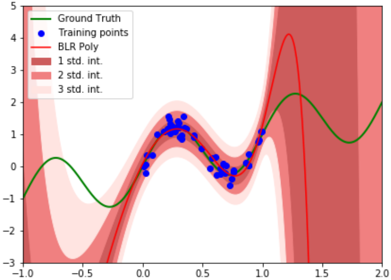
Over Fit
Poor representation of $\sin(2\pi \mathrm{x})$

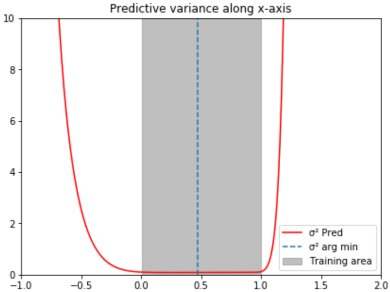
Bayesian Polynomial Regression

- Polynomial regression for 1D input, *i.e.* $x_i \in \mathbb{R}$:

$$\mathbf{\Phi} = \begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^M \\ 1 & x_2 & x_2^2 & \dots & x_2^M \\ \dots & \dots & \dots & \dots & \\ 1 & x_N & x_N^2 & \dots & x_N^M \end{pmatrix} \qquad \mathbf{Y} = \mathbf{\Phi}\mathbf{w} + \boldsymbol{\varepsilon} \qquad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 & \dots & \varepsilon_N \end{pmatrix}^T, \ \varepsilon_i \sim \mathcal{N}(0, \sigma^2)$$

- Apply Bayesian linear regression in non-linear feature space $\mathbf{\Phi}$
  - Same closed-form solution for posterior and predictive distribution in $\mathbf{\Phi}$
- Practical session : regression for $f(x) = x + sin(2\Pi x)$, $M = 10$, $\alpha = 0.05$



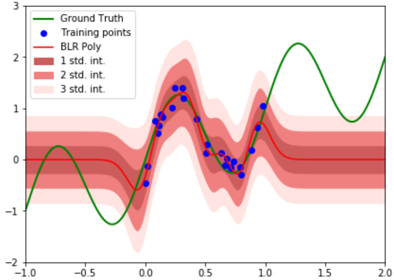


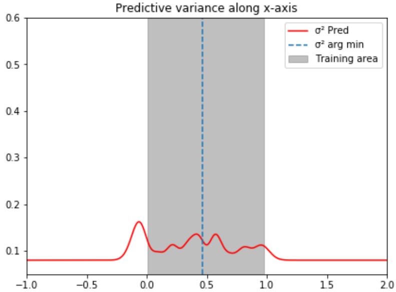
RBF Gaussian Regression

- Gaussian regression for 1D input, *i.e.* $x_i \in \mathbb{R}$, $\Phi_j(x_i) = exp\left(-\frac{(x_i - \mu_j)^2}{2s^2}\right)$:

$$\boldsymbol{\Phi} = \begin{pmatrix} \Phi_1(x_1) & \Phi_2(x_1) & \dots & \Phi_M(x_1) \\ \dots & \dots & \dots & \dots \\ \Phi_1(x_N) & \Phi_2(x_N) & \dots & \Phi_M(x_N) \end{pmatrix} \quad \mathbf{Y} = \boldsymbol{\Phi}\mathbf{w} + \varepsilon, \; \varepsilon_i \sim \mathcal{N}(0, \sigma^2)$$

- Apply Bayesian linear regression in non-linear feature space $\boldsymbol{\Phi}$





- Practical session : issue with localized features, epistemic uncertainty $\boldsymbol{\Phi}(\mathbf{x}^*)^T \boldsymbol{\Sigma} \boldsymbol{\Phi}(\mathbf{x}^*) \rightarrow 0$ far from trainig data (μ_j)

Outline

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

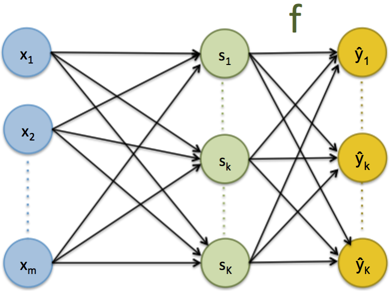
No analytical expression for posterior $p(\mathbf{w}|\mathcal{D})$ and $p(y|\mathbf{x}^*, \mathcal{D})$ in general

$\Rightarrow$ **Approximation nedded !**

- **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-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 $p(\mathbf{w}|\mathcal{D})$ and a proposed parametric function

Outline

Bayesian Logistic Regression (BLR)



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

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

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

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

Bayesian Logistic Regression training (MAP)

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

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

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

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

  - Gaussian prior $\Leftrightarrow$ weight decay

Bayesian Logistic Regression training (MAP)

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

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

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

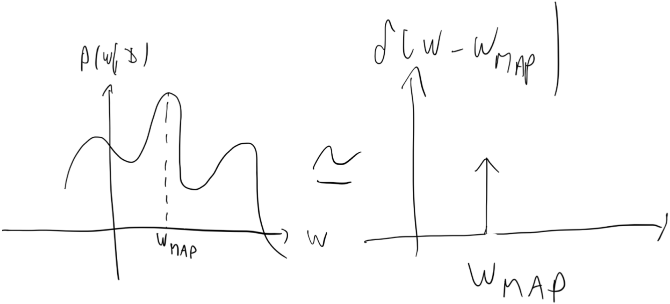
Bayesian Logistic Regression training (MAP)

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

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

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

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



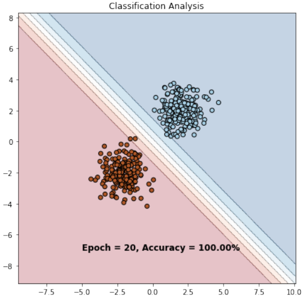
Bayesian Logistic Regression training (MAP)

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

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

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

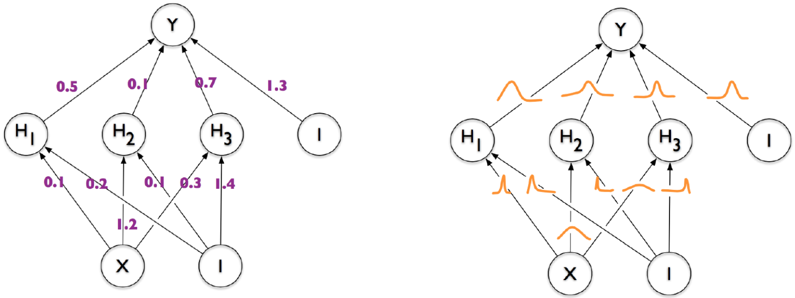
  - Need full posterior distribution $p(\mathbf{w}|\mathbf{X},\mathbf{Y})$, but posterior intractable
  - $p(\mathbf{w}|\mathbf{X},\mathbf{Y}) \approx \delta(\mathbf{w}-\boldsymbol{w}_{\mathrm{MAP}}) \Rightarrow p(\mathbf{y}=1|\boldsymbol{x}^{*},\mathcal{D}) \approx p(y=1|\boldsymbol{x},\boldsymbol{w}_{\mathrm{MAP}})$



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

Outline

Bayesian Neural Networks (BNN)



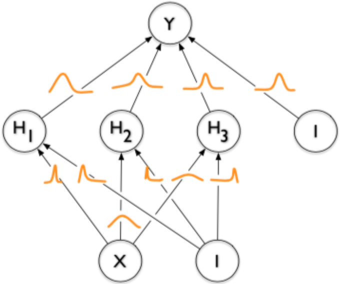
Credit : [Blundell et al., 2015]

- $\boxed{\text{Standard NN : } \mathbf{y}_i = f^{\mathbf{w}}(\mathbf{x}_i), \textbf{ Bayesian NN : } p(\mathbf{y}_i|\mathbf{x}_i, \mathcal{D})}$
  - Define prior over weights $p(\mathbf{w})$, *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 $p(y_i|\mathbf{x}_i, \mathbf{w}) = \mathcal{N}(y_i; f^w(\mathbf{x}_i), \beta^{-1})$
  - **<u>Goal :</u> 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)

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

- **With Bayesian Neural networks, even with :**
  - Gaussian prior $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|0, \alpha^{-1}\mathcal{I})$
  - Gaussian likelihood, *e.g.* regression $p(\mathbf{y}_i|\mathbf{x}_i, \mathbf{w}) = \mathcal{N}(y_i; f^w(\mathbf{x}_i), \beta^{-1})$
  - Posterior $p(\mathbf{w}|\mathbf{X}, \mathbf{Y}, \beta) \propto p(\mathbf{w}) \prod_{i=1}^{N} p(\mathbf{y}_i|\mathbf{x}_i, \mathbf{w})$ **is NOT Gaussian ! !**
- **Non-linear dependence of $f^w(\mathbf{x})$ on $\mathbf{w}$!**
- RECAP :
  - $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$
  - $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|A\mathbf{x} + b, \boldsymbol{\Sigma}_y)$
    - Linear depedence $A\mathbf{x} + b$ required
  - Then : $p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{x|\mathbf{y}}, \boldsymbol{\Sigma}_{x|\mathbf{y}})$
    - Not true for BNNs !



Credit : [Blundell et al., 2015]

Posterior Inference : MCMC Sampling

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

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

- Monte Carlo estimation of the intergral :

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

- Can't sample exactly from $p(\mathbf{w}|\mathcal{D})$, **BUT approximate sampling using Markov chain Monte Carlo (MCMC) possible !**
  - Metropolis-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 $p(\mathbf{w}|\mathbf{X},\mathbf{Y})$ cannot usually be evaluated analytically

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

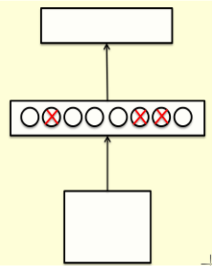
$$KL(q_{\boldsymbol{\theta}}(\mathbf{w})||p(\mathbf{w}|\mathbf{X},\mathbf{Y})) = \int q_{\boldsymbol{\theta}}(\mathbf{w}) \log \frac{q_{\boldsymbol{\theta}}(\mathbf{w})}{p(\mathbf{w}|\mathbf{X},\mathbf{Y})} d\mathbf{w}$$

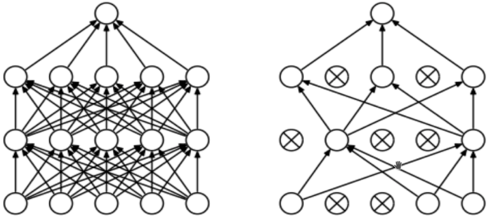
- **Computing approximate predictive distribution :** $p(\mathbf{w}|\mathbf{X},\mathbf{Y}) \Leftarrow q_{\boldsymbol{\theta}^*}(\mathbf{w})$:

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

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





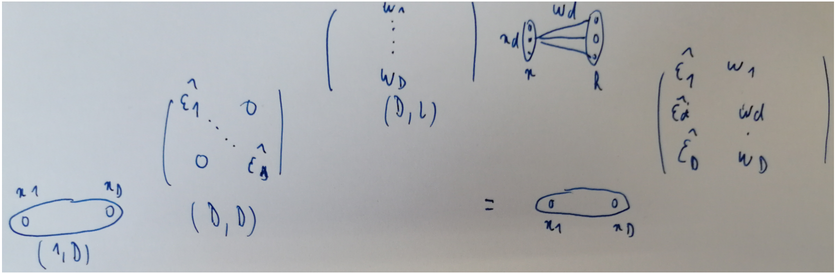
Standard Neural Net

After applying dropout.

Credits: Geoffrey E. Hinton, NIPS 2012

Dropout as a variational inference [Gal, 2016]

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



a. dimension (1,D)

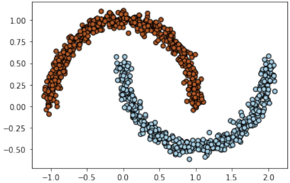
Dropout as a variational inference [Gal, 2016]

- **Illustration : dropout for a 2 layer NN** (1 hidden), $\epsilon_i \sim Bernoulli(1-p_i)$:
  - $\boxed{\mathbf{h_1} = \sigma(\hat{\mathbf{x}}\mathbf{W_1}) = \sigma(\mathbf{x}\hat{\mathbf{W}}_\mathbf{1})}$, $\hat{\mathbf{W}}_1 = \text{diag}(\hat{\epsilon_1})\mathbf{W_1}$
  - $\boxed{\hat{\mathbf{y}} =: f^{\hat{\mathbf{W}}_1, \hat{\mathbf{W}}_2}(\mathbf{x}) = \hat{\mathbf{h_1}}\mathbf{W_2} = \mathbf{h_1}\hat{\mathbf{W}}_\mathbf{2}}$, $\hat{\mathbf{W}}_2 = \text{diag}(\hat{\epsilon_2})\mathbf{W_2}$ - $\hat{\mathbf{W}} = \{\hat{\mathbf{W}}_\mathbf{1}; \hat{\mathbf{W}}_\mathbf{2}\}$
- **MC Dropout sampling :** $\frac{1}{S}\sum_{s\in S} p(\mathbf{y}_i|f^{\hat{\mathbf{W}}}(\mathbf{x}_i)) \approx \int p(\mathbf{y}|f^{\mathbf{W}}(\mathbf{x}^*)q(\mathbf{W})d\mathbf{w}$
  - $\forall$ layer $l \in \{1; L\}$, $\mathbf{W}_l$ random variable : $\mathbf{W}_l \sim q(\mathbf{W}_l) = g(\mathbf{M}_l, \varepsilon_l) = \text{diag}(\boldsymbol{\varepsilon_l})\mathbf{M}_l$
    - $\varepsilon_{l,i} \sim Bernoulli(1-p_l)$, $\mathbf{M}_l$ deterministic parameters
  - $q_\mathbf{M}(\mathbf{W}) = \prod_{l=1}^{M} q_{\mathbf{M_l}}(\mathbf{W}_l)$
- **Big result** (see [Gal, 2016]) : **training NN with dropout** $\Leftrightarrow$ **training BNN with variational posterior approximation** $q_\mathbf{M}(\mathbf{W})$ (and some prior $p(\mathbf{W})$)
  - $\mathbf{M} = \{\mathbf{M}_l\}_{l\in\{1;L\}}$ variational parameters
- **MC dropout :** sampling several passes with dropout $\Leftrightarrow$ **performing MC approximate inference with variational posterior** $q_\mathbf{M}(\mathbf{W})$

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

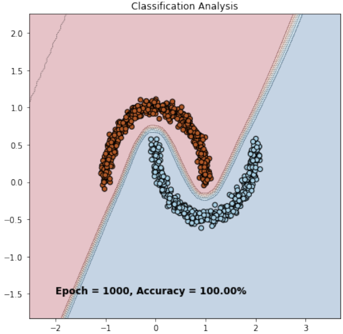
Application : MC dropout for predictive distribution

- MC dropout for non-linear classification

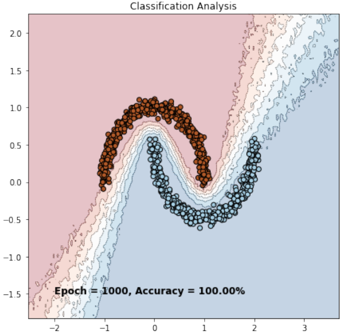


- As for BLR : $p(y=1|\mathbf{x}^*, \mathcal{D}) \approx \sum_{s=1}^{S} f_{\mathbf{w}^s}(\mathbf{x}^*)$, $\mathbf{w}^s \sim q(\mathbf{W})$

Deterministic NN



Bayesian NN (MC dropout)



References I

Bishop, C. M. (2006).
Pattern Recognition and Machine Learning (Information Science and Statistics).
Springer-Verlag, Berlin, Heidelberg.

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, Y. (2016).
Uncertainty in Deep Learning.
PhD thesis, University of Cambridge.

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, 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, 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, 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, P., Nummenmaa, A., and Vehtari, A. (2014).
Expectation propagation for neural networks with sparsity-promoting priors.
Journal of Machine Learning Research, 15 :1849–1901.

MacKay, D. J. C. (1992).
A practical bayesian framework for backpropagation networks.
Neural Comput., 4(3) :448–472.

Murphy, K. P. (2012).
Machine Learning : A Probabilistic Perspective.
The MIT Press.

Neal, R. M. (1996).
Bayesian Learning for Neural Networks.
Springer-Verlag, Berlin, Heidelberg.