Introduction to Bayesian methods and Bayesian Neural Networks

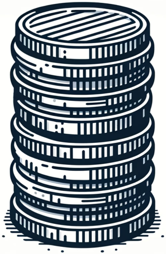
Tomasz Kuśmierczyk

2024-06-14

This research is part of the project No. ***2022/45/P/ST6/02969*** *co-funded by the National Science Centre and the European Union Framework Programme for Research and Innovation Horizon 2020 under the Marie Skłodowska-Curie grant agreement No. 945339. For the purpose of Open Access, the author has applied a CC-BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission;*

Coin tossing example

Task: for a biased (=unfair) coin, find the probability of heads θ :



Coin tossing example

Task: for a biased (=unfair) coin, find the probability of heads θ :

1. Observations $\mathcal{D} = \{H, H, H, T, H, H, T, H, H, H, H, T\}$

Coin tossing example

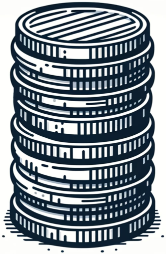
Task: for a biased (=unfair) coin, find the probability of heads θ :

2. Observations $\mathcal{D} = \{H, H, H, T, H, H, T, H\}$

Coin tossing example

Task: for a biased (=unfair) coin, find the probability of heads θ :

3. Observations $\mathcal{D} = \{H, H, H, T\}$



Coin tossing example

Task: for a biased (=unfair) coin, find the probability of heads θ :

4. Observations $\mathcal{D} = \{T, T\}$

Maximum Likelihood Estimate

1. Specify model.
2. Select loss.
3. Find parameters minimizing loss (=maximizing data likelihood).

Maximum Likelihood Estimate

1. Specify model.
2. Select loss.
3. Find parameters minimizing loss (=maximizing data likelihood).

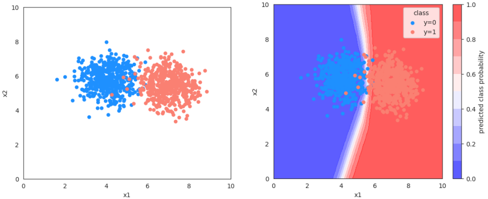
Coin tossing example:

- Parameter: θ
- Option 1: MSE loss: $\mathbb{L}(D,\theta) = \sum_{y\in\mathcal{D}}(y-\theta)^2$
- Option 2: Bernoulli negative log-likelihood: $\mathbb{L}(D,\theta) = -\sum_{y\in D}\log p(y|\theta)$ where $p =$ Bernoulli pmf
- Solution: $\hat{\theta} = \text{argmin}_\theta \mathbb{L}(D,\theta)$
  $\implies \hat{\theta} = 0.75$ for 1.,2.,3.
  $\implies \hat{\theta} = 0.0$ for 4.

Few problems with MLE

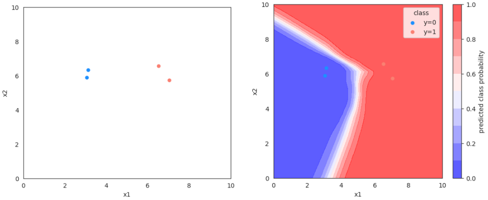
- **Overfitting with Small Sample Size**: If we toss the coin a very small number of times (e.g., once or twice), MLE can give misleading results.
- **Bias in Estimation with Limited Data**: MLE is highly sensitive to the sample size. With few observations, the estimation can be heavily biased.
- **Zero Probability Issue**: If an outcome is not observed in the sample, MLE assigns it a probability of zero.
- **Doesn't Account for Prior Knowledge**: MLE only uses the observed data and does not incorporate any prior knowledge or beliefs.
- **Variance in Estimates**: The variance of the MLE estimate is high for small sample sizes, leading to unstable predictions.
- **Sensitivity to Outliers**: Outliers or rare events can disproportionately affect the MLE.

MLE solution $p(y|x, \mathcal{D})$ for a NN trained on 1k data points



Based on: `https://github.com/wiseodd/last_layer_laplace`

MLE solution $p(y|x, \mathcal{D})$ for a NN trained on 4 data points



Based on: `https://github.com/wiseodd/last_layer_laplace`

Goal

We want ML models that:

- are uncertain about unseen things
- become more certain with more data

MLE vs Bayes: Latent variable inference

- point-wise - find one value $\hat{\theta}$,
  e.g., by minimizing loss = maximizing likelihood (MLE) / maximum a posteriori (MAP): $\cancel{\hat{\theta} = \text{argmin}_{\theta} \mathcal{L}(\mathcal{D}, \theta)}$
- distributional - find a posterior $p(\theta|\mathcal{D})$
  - we get uncertainty about θ
    (e.g., variance in addition to the mean)

Bayes Theorem

Everything follows from two basic rules:

- Sum rule: $p(A) = \sum_b p(A, B = b)$
- Product rule: $p(A, B) = p(B|A) \cdot p(A) = p(A|B) \cdot p(B)$

Bayes Theorem

Everything follows from two basic rules:

- Sum rule: $p(A) = \sum_b p(A, B = b)$
- Product rule: $p(A, B) = p(B|A) \cdot p(A) = p(A|B) \cdot p(B)$

Bayes' Theorem:

$$p(A|B) = \frac{p(B|A) \cdot p(A)}{p(B)} = \frac{p(B|A) \cdot p(A)}{\sum_a p(B, A = a)} = \frac{p(B|A) \cdot p(A)}{\sum_a p(B|A = a)p(A = a)}$$

Basic Concepts of Bayesian Methods

$$p(A|B) = \frac{p(B|A) \cdot p(A)}{p(B)}$$

Let's rename $A \to \theta$, $B \to \mathcal{D}$:

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta) \cdot p(\theta)}{p(\mathcal{D})}$$

- **Prior** $p(\theta)$**:** Initial belief on parameters before seeing data
- **Likelihood** $p(\mathcal{D}|\theta)$**:** Probability of data given parameters of the model
- **Posterior** $p(\theta|\mathcal{D})$**:** Updated belief after seeing (more) data

Bayes Theorem Example: Determining Stroke Type Based on MRI Scan Intensity

- The MRI scan of a patient shows an intensity value of $y = 140$ units ($\mathcal{D} = \{140\}$).

Bayes Theorem Example: Determining Stroke Type Based on MRI Scan Intensity

- The MRI scan of a patient shows an intensity value of $y = 140$ units ($\mathcal{D} = \{140\}$).
- Ischemic Stroke (Ischemic): In an ischemic stroke, affected brain areas may show lower signal intensity due to reduced blood flow. Assume these intensity values follow a normal distribution with a mean of 100 units and a standard deviation of 20:
  $y \sim \mathcal{N}(100, 20^2)|_{\theta=I}$.
- Hemorrhagic Stroke (Hemorrhagic): In a hemorrhagic stroke, affected areas show higher signal intensity due to bleeding. Assume these intensity values follow a normal distribution with a mean of 180 units and a standard deviation of:
  $y \sim \mathcal{N}(180, 30^2)|_{\theta=H}$.

Bayes Theorem Example: likelihood

$$p(y|\theta) = \mathcal{N}(y|100, 20^2) \cdot \mathbb{I}[\theta = l] + \mathcal{N}(y|180, 30^2) \cdot \mathbb{I}[\theta = H]$$

Bayes Theorem Example: priors

Prior Beliefs about Parameters:

- Ischemic Stroke (Ischemic): This is the most common type of stroke, accounting for about 85% of all strokes (prior probability = 0.85):
  $p(\theta = I) = 0.85$
- Hemorrhagic Stroke (Hemorrhagic): Less common, these strokes account for the remaining 15% of stroke cases (prior probability = 0.15).
  $p(\theta = H) = 0.15$

Bayes Theorem Example: solution

- Data: $\mathcal{D} = \{140\}$
- Likelihood: $\mathcal{N}(y|100, 20^2)|_{\theta=I}$, $\mathcal{N}(y|180, 30^2)|_{\theta=H}$
- Priors: $p(\theta = I) = 0.85$, $p(\theta = H) = 0.15$

Bayes Theorem Example: solution

- Data: $\mathcal{D} = \{140\}$
- Likelihood: $\mathcal{N}(y|100, 20^2)|_{\theta=I}$, $\mathcal{N}(y|180, 30^2)|_{\theta=H}$
- Priors: $p(\theta = I) = 0.85$, $p(\theta = H) = 0.15$
- Computation:
  - $p(\mathcal{D}|\theta = I) = \mathcal{N}(140|100, 20^2) = 0.0027$
  - $p(\mathcal{D}|\theta = H) = \mathcal{N}(140|180, 30^2) = 0.0055$

Bayes Theorem Example: solution

- Data: $\mathcal{D} = \{140\}$
- Likelihood: $\mathcal{N}(y|100, 20^2)|_{\theta=I}$, $\mathcal{N}(y|180, 30^2)|_{\theta=H}$
- Priors: $p(\theta = I) = 0.85$, $p(\theta = H) = 0.15$
- Computation:
  - $p(\mathcal{D}|\theta = I) = \mathcal{N}(140|100, 20^2) = 0.0027$
  - $p(\mathcal{D}|\theta = H) = \mathcal{N}(140|180, 30^2) = 0.0055$
  - $p(\mathcal{D}) = 0.0027 * 0.85 + 0.15 * 0.0055 = 0.002295 + 0.000825 = 0.00312$
  - $p(\theta = I|\mathcal{D}) = 0.0027 * 0.85/0.00312 = 0.736$
  - $p(\theta = H|\mathcal{D}) = 0.15 * 0.0055/0.00312 = 0.264$

Understanding priors: posteriors as priors

$$p(\theta|\mathcal{D}_0) = \frac{p(\mathcal{D}_0|\theta) \cdot p(\theta)}{p(\mathcal{D}_0)}$$

Understanding priors: posteriors as priors

$$p(\theta|\mathcal{D}_0) = \frac{p(\mathcal{D}_0|\theta) \cdot p(\theta)}{p(\mathcal{D}_0)}$$

$$p(\theta|\mathcal{D}_1 \cup \mathcal{D}_0) = \frac{p(\mathcal{D}_1 \cup \mathcal{D}_0|\theta) \cdot p(\theta)}{p(\mathcal{D}_1 \cup \mathcal{D}_0)} =$$

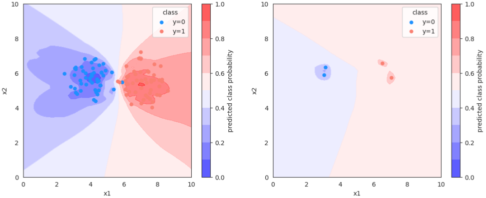
$$= \frac{p(\mathcal{D}_1|\theta)p(\mathcal{D}_0|\theta)p(\theta)}{p(\mathcal{D}_1)p(\mathcal{D}_0)} = \frac{p(\mathcal{D}_1|\theta)p(\theta|\mathcal{D}_0)}{p(\mathcal{D}_1)}$$

Understanding priors: posteriors as priors

$$p(\theta|\mathcal{D}_0) = \frac{p(\mathcal{D}_0|\theta)\cdot p(\theta)}{p(\mathcal{D}_0)}$$

$$p(\theta|\mathcal{D}_1 \cup \mathcal{D}_0) = \frac{p(\mathcal{D}_1 \cup \mathcal{D}_0|\theta)\cdot p(\theta)}{p(\mathcal{D}_1 \cup \mathcal{D}_0)} =$$

$$= \frac{p(\mathcal{D}_1|\theta)p(\mathcal{D}_0|\theta)p(\theta)}{p(\mathcal{D}_1)p(\mathcal{D}_0)} = \frac{p(\mathcal{D}_1|\theta)p(\theta|\mathcal{D}_0)}{p(\mathcal{D}_1)}$$



$$p(\theta|\mathcal{D}_1 \cup \mathcal{D}_0) \quad \leftarrow \quad p(\theta|\mathcal{D}_0) \leftarrow \quad p(\theta) = p(\theta|\emptyset)$$

Bayes Theorem Example: repeated measurement

Previous solution for $\mathcal{D}_0 = \{140\}$:

- $p(\theta = I|\mathcal{D}) = 0.0027 * 0.85/0.00312 = 0.736$
- $p(\theta = H|\mathcal{D}) = 0.15 * 0.0055/0.00312 = 0.264$

After additional measurement $y = 160$: $\mathcal{D}_1 = \{160\}$; $\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_1 = \{140, 160\}$:

- $p(\mathcal{D}_1|\theta = I) = \mathcal{N}(160|100, 20^2) = 0.0002$
- $p(\mathcal{D}_1|\theta = H) = \mathcal{N}(160|180, 30^2) = 0.0106$
- $p(\mathcal{D}) = 0.0002 * 0.736 + 0.0106 * 0.264 = 0.0029456$
- $p(\theta = I|\mathcal{D}) = 0.0002 * 0.736/0.0029456 = 0.05$
- $p(\theta = H|\mathcal{D}) = 0.0106 * 0.264/0.0029456 = 0.95$

Conjugate priors

- Problem: how to find $p(\theta|\mathcal{D})$?
- For some pairs of prior+likelihood, the posterior takes the same form as prior

Conjugate priors: whiteboard example

Back to the coin-tossing example:

Let's consider Beta-Bernoulli model:

- prior $p(\theta) = Beta(\theta|\alpha, \beta) = \frac{\theta^{\alpha-1} \cdot (1-\theta)^{\beta-1}}{B(\alpha,\beta)}$
- likelihood
  $p(\mathcal{D}|\theta) = \prod_{y \in \mathcal{D}} Bernoulli(y|\theta) = \prod_{y} \left(\theta^{y} \cdot (1-\theta)^{(1-y)}\right)$
- example data: $\mathcal{D} = \{T, T\}$

Conjugate priors: whiteboard example

Back to the coin-tossing example:

Let's consider Beta-Bernoulli model:

- prior $p(\theta) = Beta(\theta|\alpha, \beta) = \frac{\theta^{\alpha-1} \cdot (1-\theta)^{\beta-1}}{B(\alpha,\beta)}$
- likelihood
  $p(\mathcal{D}|\theta) = \prod_{y \in \mathcal{D}} Bernoulli(y|\theta) = \prod_y \left(\theta^y \cdot (1-\theta)^{(1-y)}\right)$
- example data: $\mathcal{D} = \{T, T\}$

Find $p(\theta|\mathcal{D})$:

1. Let's start with $p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta)$
2. ...
3. `https://homepage.divms.uiowa.edu/~mbognar/applets/beta.html`
4. Importance of priors: $\alpha = \beta = 1$ vs $\alpha = \beta = 10$

Conjugate priors

Conjugate priors:
`https://en.wikipedia.org/wiki/Conjugate_prior`

Likelihood $p(x_i\|\theta)$	Model parameters θ	Conjugate prior (and posterior) distribution $p(\theta\|\Theta), p(\theta\|\mathbf{x},\Theta) = p(\theta\|\Theta')$	Prior hyperparameters Θ	Posterior hyperparameters[note 1] Θ'	Interpretation of hyperparameters	Posterior predictive[note 2] $p(\tilde{x}\|\mathbf{x},\Theta) = p(\tilde{x}\|\Theta')$
Bernoulli	p (probability)	Beta	$\alpha,\, \beta \in \mathbb{R}$	$\alpha + \sum_{i=1}^{n} x_i,\, \beta + n - \sum_{i=1}^{n} x_i$	α successes, β failures[note 3]	$p(\tilde{x}=1) = \frac{\alpha'}{\alpha'+\beta'}$ (Bernoulli)

Finding latent variable vs predictive distribution

Two tasks:

- find latent variables θ (e.g. clusters in data) i.e. find $p(\theta|\mathcal{D})$
- make predictions for a new y (often based on features x) i.e. find $p(y|\mathcal{D}, x)$

NNs: Typical Regression/Classification Setting

- Data: $\mathcal{D} = \{(x_i, y_i)\}$
  but x_i (e.g., features vector) are not modeled (=are fixed)
- Task: predict unknown y for some input x
- Model:
  - parameters vector: θ
  - likelihood i.i.d.: $p(\mathcal{D}|\theta) = \prod_i p(y_i|\theta, x_i)$

NNs: likelihood $p(\mathcal{D}|\theta) = \prod_i p(y_i|\theta, x_i)$

- NN structure (layers, activations etc.) may be hidden inside of likelihood or we can write explicitly:
  $p(y|\theta, x) = p(y|\text{NN}(\theta^{\text{NN}}, x), \theta^{lik})$
  - $\theta = \theta^{\text{NN}} \cup \theta^{lik}$

NNs: likelihood $p(\mathcal{D}|\theta) = \prod_i p(y_i|\theta, x_i)$

- NN structure (layers, activations etc.) may be hidden inside of likelihood or we can write explicitly:
  $p(y|\theta, x) = p(y|\text{NN}(\theta^{\text{NN}}, x), \theta^{lik})$
  - $\theta = \theta^{\text{NN}} \cup \theta^{lik}$
  - where NN is the network
  - p "interprets" logits as parameters of a probability distribution e.g. softmax, sigmoid, normal
  - θ^{lik} are additional likelihood parameters not included in NN
- $\text{NN}(\theta, x) = \phi^L(\theta^L, \phi^{L-1}(\theta^{L-1}, \dots, \phi^1(\theta^1, x)))$
  - where $\theta^{\text{NN}} = \theta^1 \cup \dots \cup \theta^L$ consists of weights and biases in NN
  - ϕ^l are layers
    e.g. $\phi^l(\text{weights} \cup \text{biases}, \text{inputs}) = a^l(\text{weights} \cdot \text{inputs} + \text{biases})$
  - a^l are activations

Point-wise (MLE/MAP) solution

- Parameters: find one value $\hat{\theta}$,
  e.g., by minimizing loss = maximizing likelihood (MLE) / maximum a posteriori (MAP):

$$\hat{\theta} = \mathrm{argmin}_{\theta} \mathbb{L}(\mathcal{D}, \theta)$$

- Predictions:

$$\underbrace{p(y|x, \mathcal{D})}_{\text{predictive distribution}} = \underbrace{p(y|\hat{\theta}, x)}_{\text{likelihood}}$$

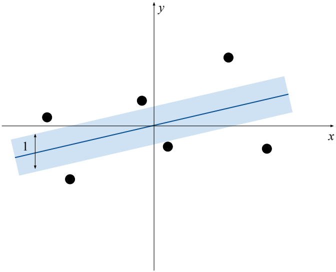
Example: Homoscedastic Gaussian regression

- $p(y_i|\mathrm{NN}(\theta, x_i), \sigma) = \mathcal{N}(y_i|\mu_i, \sigma)$
- $\mu_i := \mathrm{NN}(x_i, \theta) = \phi^L(\theta^L, \phi^{L-1}(\theta^{L-1}, \ldots, \phi^1(\theta^1, x_i)))$
- $\theta^{lik} = \{\sigma\}$
- $a^L = \text{identity (i.e., } a^L(v) = v)$

Note: we look for $\mathbb{E}_{p(y_i|\mathrm{NN}(\theta,x_i),\sigma)}[y_i] = \mu_i$, and it is coincidental that (for Gaussian regression) the NN is returning exactly μ_i.

1D linear regression: $\mathrm{NN}(x, \theta) = \theta \cdot x$ and $\sigma = 1$ (i.e. $y = \theta \cdot x + \epsilon, \epsilon \sim \mathcal{N}(0, 1)$)

- point-wise - find $\hat{\theta}$,
  e.g., maximizing likelihood (MLE) or maximum a posteriori (MAP)



Bayesian (distributional) solution

- Parameters: find a posterior $p(\theta|\mathcal{D})$
  - we get uncertainty about θ (e.g., variance in addition to mean)
- Predictions: Bayesian Model Averaging (average of all possible models weighted by the posterior):

$$\underbrace{p(y|x,\mathcal{D})}_{\text{posterior predictive}} = \int \underbrace{p(y|\theta,x)}_{\text{likelihood}} \underbrace{p(\theta|\mathcal{D})}_{\text{posterior}} d\theta$$

Bayesian (distributional) solution

- Parameters: find a posterior $p(\theta|\mathcal{D})$
  - we get uncertainty about θ (e.g., variance in addition to mean)
- Predictions: Bayesian Model Averaging (average of all possible models weighted by the posterior):

$$\underbrace{p(y|x,\mathcal{D})}_{\text{posterior predictive}} = \int \underbrace{p(y|\theta,x)}_{\text{likelihood}} \underbrace{p(\theta|\mathcal{D})}_{\text{posterior}} d\theta$$

  - ultimate goal: find posterior predictive $p(y|x,\mathcal{D})$
  - intermediate goal: find posterior $p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta)$

Bayesian solution: Making predictions with MC

- Bayesian Model Averaging (average of all possible models weighted by the posterior):

$$\underbrace{p(y|x,\mathcal{D})}_{\text{posterior predictive}} = \int \underbrace{p(y|\theta,x)}_{\text{likelihood}} \underbrace{p(\theta|\mathcal{D})}_{\text{posterior}} d\theta$$

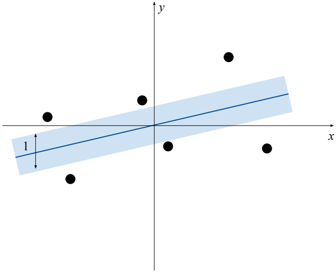
Statistics of $p(y|x,\mathcal{D})$ can be obtained using the Monte-Carlo estimates by two-step sampling:

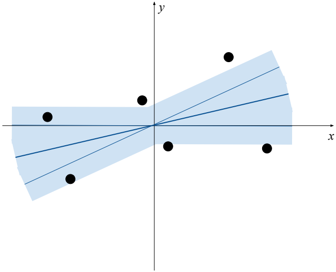
- sample $\theta \sim p(\theta|\mathcal{D})$
- for the fixed θ, sample $y \sim p(y|\theta,x)$

For example, $\mathbb{E}_{p(y|x,\mathcal{D})}[y] \approx \frac{1}{S_\theta}\frac{1}{S_y}\sum_{\theta\sim p(\theta|\mathcal{D})}\sum_{y\sim p(y|\theta,x)} y$

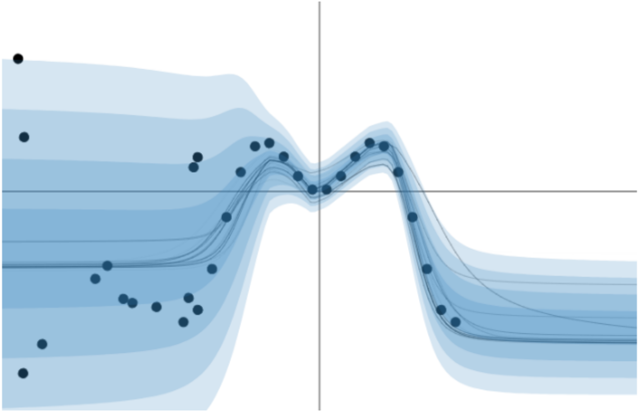
1D linear regression: $\mathrm{NN}(x, \theta) = \theta \cdot x$ and $\sigma = 1$ (i.e. $y = \theta \cdot x + \epsilon, \epsilon \sim \mathcal{N}(0, 1)$)

- point-wise - find $\hat{\theta}$,
  e.g., maximizing likelihood (MLE) or maximum a posteriori (MAP)
- distributional - find a posterior $p(\theta|D)$





1D non-linear regression example



http://mlg.eng.cam.ac.uk/yarin/blog_2248.html#demo

Figure: Multiple draws of a regression model.

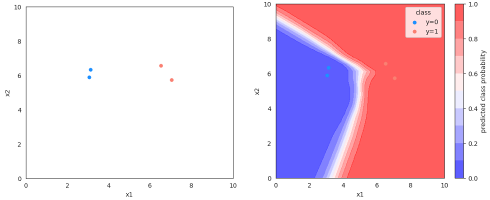
Example: Classification

- $p(y_i|\mathrm{NN}(\theta, x_i)) = \mathrm{Bernoulli}(y_i|p_i)$
- $p_i := \mathrm{NN}(x_i, \theta) = \phi^L(\theta^L, \phi^{L-1}(\theta^{L-1}, \ldots, \phi^1(\theta^1, x_i)))$
- $\theta^{lik} = \emptyset$
- $a^L(v) = \mathrm{sigmoid}(v)$
  sigmoid makes sure $p_i \in [0, 1]$

MLE (likelihood=Bernoulli) for a NN Classifier

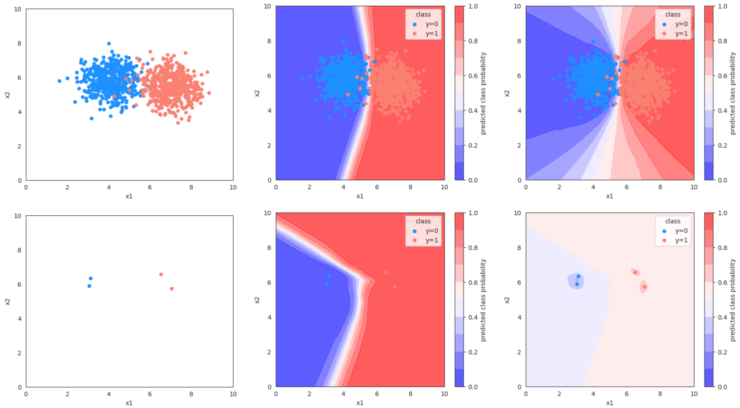
```python
class Model(nn.Module):

    def __init__(self):
        super(Model, self).__init__()

        self.layers = nn.Sequential(
            nn.Linear(n, h),
            nn.BatchNorm1d(h),
            nn.ReLU(),
            nn.Linear(h, h),
            nn.BatchNorm1d(h),
            nn.ReLU(),
            nn.Linear(h, 1),
        )

    def forward(self, x):
        x = self.layers(x)
        return self.torch.sigmoid(x)

model = Model()
opt = optim.SGD(model.parameters(), lr=1e-3, momentum=0.9, weight_decay=5e-4)

for it in range(5000):
    y_pred = model(X_train).squeeze()
    l = F.binary_cross_entropy(y_pred, y_train)
    l.backward()
    opt.step()
    opt.zero_grad()
```

MLE solution $p(y|x, \mathcal{D})$ for 4 data points



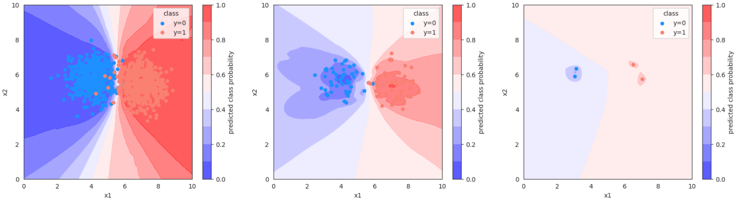
Based on: `https://github.com/wiseodd/last_layer_laplace`
accompanying: *Kristiadi et al. "Being bayesian, even just a bit, fixes overconfidence in relu networks." ICML 2020.*

Classification: MLE vs Bayesian (LLLA) solution



Based on: `https://github.com/wiseodd/last_layer_laplace`

Classification: Bayesian (LLLA) solution for varying data sizes



Based on: `https://github.com/wiseodd/last_layer_laplace`

Challenges in Bayesian learning

Design:

- likelihood and network structure
- priors $p(\theta)$

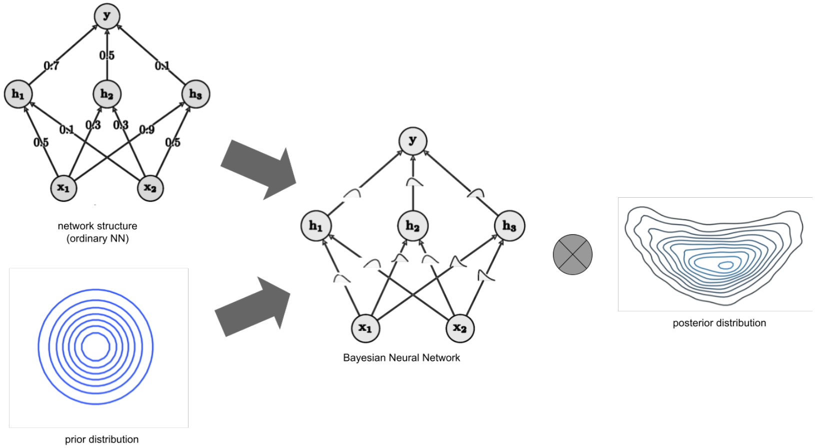
Learning:

- posterior $p(\theta|\mathcal{D}) = \frac{p(\mathcal{D},\theta)}{p(\mathcal{D})}$
- evidence $p(\mathcal{D}) = \int p(\mathcal{D}|\theta)p(\theta)d\theta$
- posterior predictive $p(y|\mathcal{D}) = \int p(y|\theta)p(\theta|\mathcal{D})d\theta$
- model selection = hyperparameters

Priors

- $\theta = \theta^L \cup \ldots \theta^1$
- Factorized priors $p(\theta) = \prod_d p(\theta_d)$
  Note: lower vs upper indices
- often $p(\theta_d) = N(\theta_d | 0, 1)$

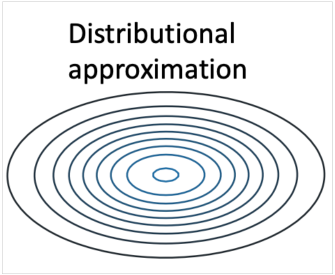
(Basic Feed-Forward) Bayesian Neural Network



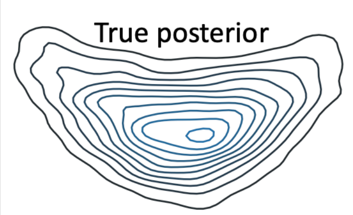
Objective: find the posterior distribution $p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta) \cdot p(\theta)}{p(\mathcal{D})}$

Finding posterior: Variational Inference

- Postulate $q(\theta|\lambda) \approx p(\theta|\mathcal{D})$

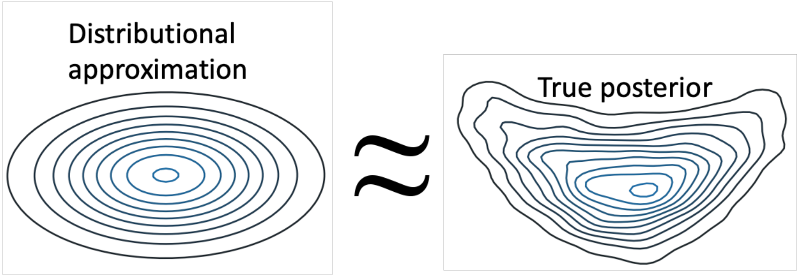


$\approx$



Finding posterior: Variational Inference

- Postulate $q(\theta|\lambda) \approx p(\theta|\mathcal{D})$



- Minimize KL between the approximation and the true posterior:
  VI: $q(\theta|\lambda) = \mathbf{argmin}_q KL(q|p)$

Approximate inference: variational objective - ELBO

$$
\begin{aligned}
KL(q|p) &= \int q(\theta|\lambda) \log\left(\frac{q(\theta|\lambda)}{p(\theta|\mathcal{D})}\right) d\theta = \int q(\theta|\lambda) \log\left(\frac{q(\theta|\lambda)p(\mathcal{D})}{p(\theta|\mathcal{D})p(\mathcal{D})}\right) d\theta \\
&= \int q(\theta|\lambda) \log\left(\frac{q(\theta|\lambda)}{p(\theta,\mathcal{D})}\right) d\theta + \log p(\mathcal{D}) \\
&= -\underbrace{\int q(\theta|\lambda)\left[\log\left(p(\mathcal{D}|\theta)p(\theta)\right) - \log q(\theta|\lambda)\right] d\theta}_{ELBO} + \log p(\mathcal{D}) \\
&= -\underbrace{\left(\mathbb{E}_q \log\left(p(\mathcal{D}|\theta)p(\theta)\right) \underbrace{-\mathbb{E}_q \log q(\theta|\lambda)}_{H(q)}\right)}_{ELBO} + \log p(D)
\end{aligned}
$$

Since $\log p(D) = const$: $\text{argmin}_q KL(q|p) = \text{argmax}_q ELBO$

Summary: variational inference for practitioners

We look for $q(\theta|\lambda) \approx p(\theta|\mathcal{D})$:

1. Choose model: likelihood $p(\mathcal{D}|\theta)$ and prior $p(\theta)$
2. Assume $q(\theta|\lambda)$ in some family parametrized by λ;
   often: $q(\theta|\lambda) \equiv N(\mu, diag(\sigma))$ so $\lambda \equiv \{\mu\} \cup \{\sigma\}$
3. Optimize with gradients: $\lambda_{next} := \lambda + \eta \nabla_\lambda \mathcal{L}$;
   usually $\mathcal{L} \equiv ELBO$
4. Use $q(\theta|\lambda)$ in place of $p(\theta|\mathcal{D})$ wherever needed

Appendix

Disclaimer on notation

Continuous r.v.-s $\longleftrightarrow$ Discrete r.v.-s:

- integral $\int$ $\longleftrightarrow$ sum $\sum$
- probability density p $\longleftrightarrow$ probability mass P

We may abuse notation by writing e.g. $r.v. \sim p(r.v.|params)$ (instead of $r.v. \sim p(params)$) to explicitly inform about $r.v.$

Loss: $\mathbb{L}(\mathcal{D}, \theta)$ = negative log-likelihood: $-\mathcal{L}(\mathcal{D}, \theta)$

Distributions: $q \equiv q(\theta|\lambda)$, $p \equiv p(\theta|\mathcal{D})$
e.g. $KL(q|p) \equiv KL(q(\theta|\lambda)|p(\theta|\mathcal{D})) = \int q(\theta|\lambda) \log \frac{q(\theta|\lambda)}{p(\theta|\mathcal{D})} d\theta$,
$H(q) \equiv H(q(\theta|\lambda)) = -\int q(\theta|\lambda) \log q(\theta|\lambda) d\theta$

Some useful identities

- $\log e^A = A$
- $\log(A \cdot B) = \log(A) + \log(B)$
- $\log(\prod_i A_i) = \sum_i \log(A_i)$
- $\log(A/B) = \log(A) - \log(B)$
- $\int g(B)p(A)dA = g(B)$