Bayesian deep learning

Deep Learning Practical Work

Aymeric Delefosse & Charles Vin

2023 – 2024

SCIENCES SORBONNE UNIVERSITÉ

Contents

Chapter 1

Bayesian Linear Regression

1.1 Linear Basis function model

1.1.1 Gaussian basis functions

1.2. Recall closed form of the posterior distribution in linear case. Then, code and visualize posterior sampling. What can you observe? Let N denote the number of training examples, K the number of dimensions of the outputs, and p the number of features (or predictors) in the input data. Consider $X \in \mathbb{R}^{N \times p}$, the input matrix, and $Y \in \mathbb{R}^{N \times K}$, the output matrix.

We typically define the posterior distribution as $p(w|X, Y)$. This distribution represents our updated beliefs about the parameters w after observing the data X and Y. According to Bayes' rule, we can express the posterior distribution as the product of two components:

1. $p(w)$, which represents our prior beliefs about the distribution of w before observing the data.
2. $p(Y|X, w)$, indicating the probability of observing the data Y given the parameters w and the data X. This quantifies how likely our data is under different hypotheses represented by w.

Therefore, the formula for the posterior distribution is given by:

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

In this formula, we start with our initial beliefs (priors) and then update these beliefs based on the new data (likelihood). In the case of our linear model, we know that $y_i = \Phi_i^T w + \epsilon$, with $\Phi \in \mathbb{R}^{N \times (p+1)}$ representing the design matrix and ϵ denoting the residual. Assuming that the error follows a centered Gaussian distribution with standard deviation $\beta^{-1} = 2\sigma^2$, meaning that $\epsilon \sim \mathcal{N}(0, \beta^{-1})$. Consequently, we can conclude that:

$$p(y_i|x_i, w) \sim \mathcal{N}(\Phi_i^T w, \beta^{-1})$$

Furthermore, we selected a centered Gaussian prior with a variance of $\alpha^{-1}I$ where α governs the prior distribution over the weights $\mathbf{w}$:

$$p(w|\alpha) \sim \mathcal{N}(0, \alpha^{-1}I)$$

In this specific case, we can demonstrate that the posterior distribution $p(w|X, Y)$ follows a Gaussian distribution as follows:

$$p(w|X, Y) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

The precision matrix $\boldsymbol{\Sigma}$, which is the inverse of the covariance matrix of the distribution parameters, is defined as:

$$\boldsymbol{\Sigma}^{-1} = \alpha I + \beta \Phi^T \Phi$$

The mean of the distribution parameters is given by:

$$\boldsymbol{\mu} = \beta \boldsymbol{\Sigma} \Phi^T Y$$

Parameters α and β serve analogous roles, with α governing the prior distribution and β regulating the likelihood.

Now, we can proceed to sample from the updated (with data) posterior distribution $p(w|X, Y)$. This process is demonstrated in Figure 1.1, with $\alpha = 2$ and $\beta = (2 \times 0.2^2)^{-1}$. It is worth noting that when $N = 0$, the posterior distribution $p(w|X, Y)$ simplifies to the prior distribution $p(w)$. As we increase the number of data points, we can observe how the model's certainty increase, demonstrated by the reduction in the variance of the parameters of the distribution. In other words, having more data points reduces aleatoric uncertainty.

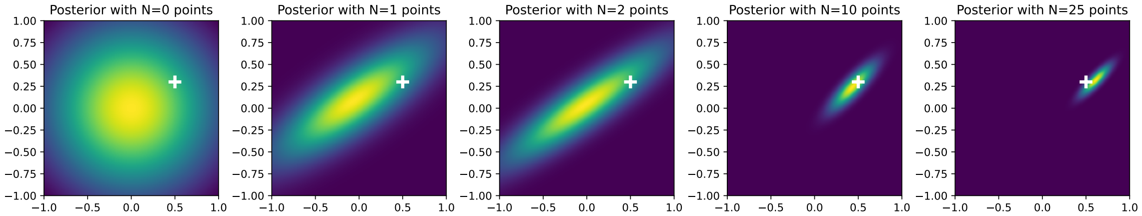


Figure 1.1: Evolution of a Bayesian posterior distribution with increasing data points: A visual representation of Bayesian inference, depicting how the posterior distribution updates as more data is incorporated. From left to right, the figures show the posterior with $N = 0$ (prior distribution), $N = 1$, $N = 2$, $N = 10$, and $N = 25$ data points, respectively. The cross mark represents the ground truth.

1.3. Recall and code closed form of the predictive distribution in linear case. Using the information from the previous question, we can now calculate the predictive distribution for new data point $x^\star$ by marginalizing over the parameter w. This is represented by the following equation where $\mathcal{D} = \{X, Y\}$ denotes the dataset:

$$p(y|x^\star, \mathcal{D}, \alpha, \beta) = \int p(y|x^\star, w, \beta) p(w|\mathcal{D}, \alpha, \beta)$$

By leveraging the property that the convolution of two Gaussian distributions results in another Gaussian distribution, we can demonstrate that the closed-form expression for the predictive distribution in the linear case is as follows:

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

We can see that the variance in the predictive distribution σ^2_{pred} for a new observation can be divided into two parts:

1. The aleatoric uncertainty, which represents the inherent noise in the data, that we fixed around β^{-1};
2. The epistemic uncertainty related to the model parameters w, characterized by $\mathbf{\Phi}(x^\star)^T \mathbf{\Sigma} \mathbf{\Phi}(x^\star)$.

It's worth noting that as the number of data points N approaches infinity ($\lim_{N\to\infty} \mathbf{\Phi}(x^\star)^T \mathbf{\Sigma} \mathbf{\Phi}(x^\star) = 0$), our understanding of the model parameters becomes nearly perfect. In this scenario, the only remaining source of uncertainty is the aleatoric uncertainty, stemming from the noise in the data.

1.4. Based on previously defined `f_pred()`, predict on the test dataset. Then visualize results using `plot_results()` defined at the beginning of the notebook. Figure 1.2 illustrates a comparison between the predictions made by a Bayesian Linear Regression model and the actual ground truth. In the left panel, you can see the model's linear fit to the training data, shown as blue points, alongside the true ground truth represented by the green line. To visualize the model's predictive uncertainty, shaded areas are used, ranging from dark to light, which correspond to one, two, and three standard deviation intervals, respectively.

The right panel of the figure focuses on the predictive variance σ^2_{pred} along the x-axis. This variance is depicted by a curve that widens as it moves away from the center of the training data, marked by the vertical dashed line. These visual elements together provide a comprehensive view of the model's confidence in its predictions across the entire domain.

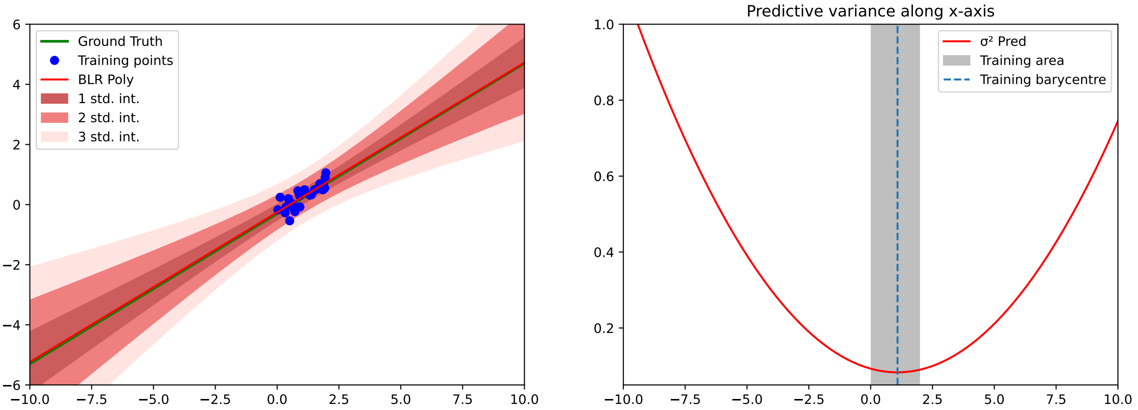


Figure 1.2: Visualization of predictive distribution of a linear dataset: The left panel illustrates the Bayesian Linear Regression fit to the training data (blue points) against the ground truth (green line). The shaded areas represent the predictive uncertainty, with one, two, and three standard deviation intervals shown in progressively lighter shades. The right panel displays the predictive variance σ^2_{pred} across the x-axis. The vertical dashed line indicates the center of the training data.

1.5. Analyse these results. Why predictive variance increases far from training distribution? Prove it analytically in the case where $\alpha = 0$ and $\beta = 1$. In the left panel of Figure 1.2, we can observe that the confidence intervals are narrowest near the cluster of training points. This suggests higher confidence in predictions within this area. As we move away from the center of the training data (towards the extremities of the x-axis), the confidence intervals become wider, indicating increasing uncertainty in the model's predictions.

Looking at the right panel of Figure 1.2, we notice that the variance remains low in the region where the training data is located (the grey shaded area). This correlates with the tight confidence intervals shown in the left panel. As expected, the predictive variance is lowest near the training barycentre, reflecting greater model certainty in this region due to the presence of more training data points.

As we move away from the training area on either side, the predictive variance increases significantly, which is consistent with the expanding confidence intervals in the left panel. This sharp increase in variance indicates a significant decrease in the model's confidence in its predictions outside the range of the training data. This is a common pattern in regression analysis, as the model has less data to rely on for predictions in these regions. Thus, the Bayesian Linear Regression model offers more reliable predictions within the training data range, with confidence diminishing as predictions move away from this range.

Let's prove it analytically. With $\alpha = 0$ and $\beta = 1$, the computation of $\mathbf{\Sigma}^{-1}$ simplifies as follows:

$$\begin{aligned}
\mathbf{\Sigma}^{-1} &= 0 \cdot I_3 + 1 \cdot \Phi^T \Phi \\
&= \Phi^T \Phi \\
&= \begin{pmatrix} 1 & \dots & 1 \\ x_1 & \dots & x_N \end{pmatrix} \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_N \end{pmatrix} \\
&= \begin{pmatrix} N & \sum x_i \\ \sum x_i & \sum x_i^2 \end{pmatrix}
\end{aligned}$$

To invert $\mathbf{\Sigma}$, we use the classic formula for a 2×2 matrix:

$$\mathbf{\Sigma} = \frac{1}{\det \mathbf{\Sigma}^{-1}} \begin{pmatrix} \sum x_i^2 & -\sum x_i \\ -\sum x_i & N \end{pmatrix}$$

Returning to our expression for $\sigma^2_{\text{pred}} = \Phi(x^\star)^T \mathbf{\Sigma}\Phi(x^\star)$, we have:

$$
\begin{aligned}
\sigma^2_{\text{pred}} = \Phi(x^\star)^T \mathbf{\Sigma}\Phi(x^\star) &= \frac{1}{\det \mathbf{\Sigma}^{-1}} \begin{pmatrix} 1 \\ x^\star \end{pmatrix} \begin{pmatrix} \sum x_i^2 & -\sum x_i \\ -\sum x_i & N \end{pmatrix} \begin{pmatrix} 1 & x^\star \end{pmatrix} \\
&= \frac{1}{\det \mathbf{\Sigma}^{-1}} \begin{pmatrix} 1 \\ x^\star \end{pmatrix} \begin{pmatrix} \sum_i x_i^2 - x^\star \sum_i x_i & x^\star N - \sum_i x_i \end{pmatrix} \\
&= \frac{1}{\det \mathbf{\Sigma}^{-1}} \left(\sum_i x_i^2 - x^\star \sum_i x_i + x^{\star 2} N - x^\star \sum_i x_i \right) \\
&= \frac{1}{\det \mathbf{\Sigma}^{-1}} \left(\sum_i x_i^2 - 2x^\star \sum_i x_i + x^{\star 2} N \right) \\
&= \frac{1}{\det \mathbf{\Sigma}^{-1}} \sum_i (x_i - x^\star)^2
\end{aligned}
$$

This formula reveals that the predictive variance is directly proportional to the squared differences between each training data point x_i and the prediction point $x^\star$. As $x^\star$ moves away from the region where the training data is concentrated, these squared differences increase, consequently leading to an increase in the predictive variance.

Bonus: What happens when applying Bayesian Linear Regression on the following dataset? Examining the right panel in Figure 1.3, an intriguing observation is made: the variance is unexpectedly minimized at the barycenter, i.e. the "hole" where there are no training data points. Normally, one would anticipate an increase in variance in data-scarce regions. This behavior indicates that the model exhibits overconfidence in its predictions, a notion supported by the lower variance at the endpoints when compared to our earlier model depicted in Figure 1.2. This overconfidence stems from our strong prior belief in the linearity of the data.

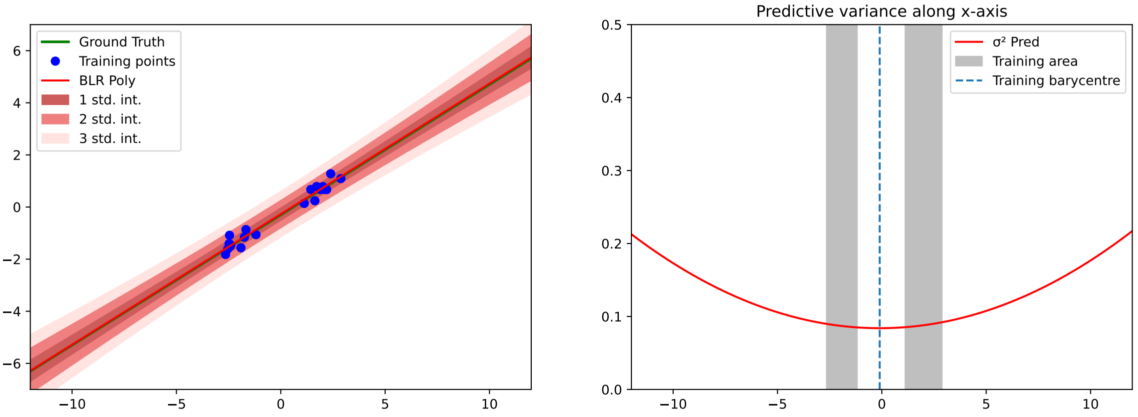


Figure 1.3: Visualization of predictive distribution of a linear dataset featuring a "hole".

1.2 Non Linear models

1.2.1 Polynomial basis functions

2.2. Code and visualize results on sinusoidal dataset using polynomial basis functions. What can you say about the predictive variance? Figure 1.4 and Figure 1.5 illustrate a comparison between the predictions made by a Bayesian Polynomial Regression model and the actual ground truth. Similar to the linear case, we can draw the same conclusions: as we move further away from our training data, uncertainty increases. However, in this case, our model demonstrates the ability to capture more intricate patterns, such as those resembling a sinusoidal function. It's worth noting that beyond the training data points, the model struggles to generalize, as it lacks the necessary data points to accurately capture the periodic nature of the sinusoidal ground truth.

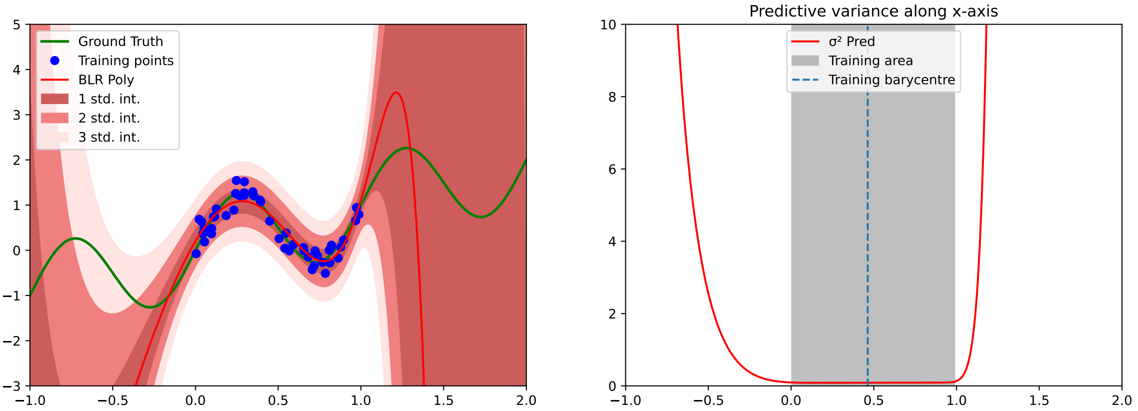


Figure 1.4: Visualization of predictive distribution of a sinusoidal dataset: The left panel illustrates the Bayesian Polynomial Regression fit to the training data (blue points) against the ground truth (green line). The shaded areas represent the predictive uncertainty, with one, two, and three standard deviation intervals shown in progressively lighter shades. The right panel displays the predictive variance σ^2_{pred} across the x-axis. The vertical dashed line indicates the center of the training data.

Furthermore, we decided to explore the effects of providing data points that are more dispersed and less abundant, as shown in Figure 1.5. Unsurprisingly, we observe that the variance around these small clusters is minimal, and as we move away from these regions, the variance increases. This phenomenon highlights the strength of these models: when given data, they become increasingly confident in specific areas. Therefore, in the context of uncertainty, we can gradually narrow down our desired outcomes by continuously adding more data points over time.

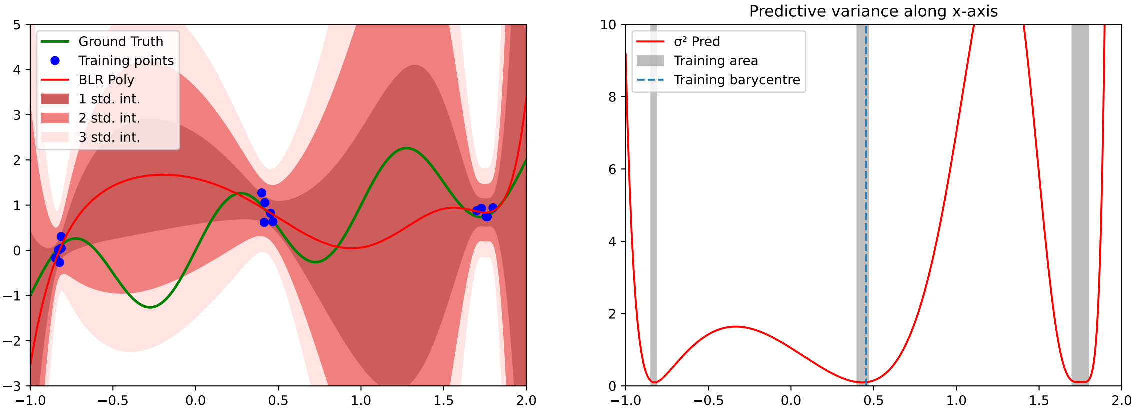


Figure 1.5: Visualization of predictive distribution of a sinusoidal dataset featuring "holes" and sparse data points.

1.2.2 Gaussian basis functions

2.4. Code and visualize results on sinusoidal dataset using Gaussian basis functions. What can you say this time about the predictive variance? Figure 1.6 and Figure 1.7 illustrate a comparison between the predictions made by a RBF Gaussian Regression model and the actual ground truth. Contrary to the last two models, we conclude different conclusions. Notably, the predictive variance isn't highest far from the training barycenter but at points where there is information. It fluctuates, with peaks and troughs corresponding to the areas where the density of training points varies. Thus, the model is closely aligned to the mean of the training data and for instance, we can see how it can impacts the sinusoidal learned curves, with a small shift .

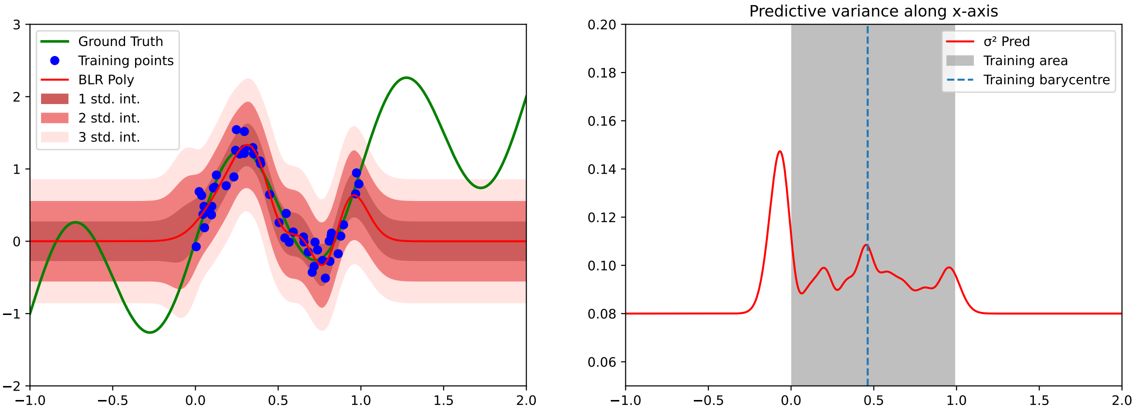


Figure 1.6: Visualization of predictive distribution of a sinusoidal dataset: The left panel illustrates the RBF Gaussian Regression fit to the training data (blue points) against the ground truth (green line). The shaded areas represent the predictive uncertainty, with one, two, and three standard deviation intervals shown in progressively lighter shades. The right panel displays the predictive variance σ^2_{pred} across the x-axis. The vertical dashed line indicates the center of the training data.

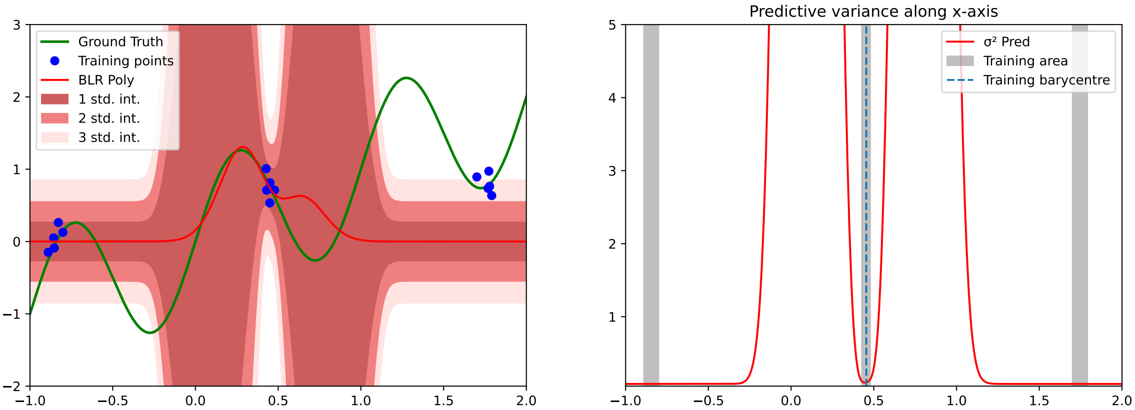


Figure 1.7: Visualization of predictive distribution of a sinusoidal dataset featuring "holes" and sparse data points.

2.5. Explain why in regions far from training distribution, the predictive variance converges to this value when using localized basis functions such as Gaussians.

Chapter 2

Approximate Inference in Classification

2.1 Bayesian Logistic Regression

2.1.1 Maximum-A-Posteriori Estimate

1.1. Analyze the results provided by Figure 2.1. Looking at $p(y = 1|x, \mathbf{w_{MAP}})$, what can you say about points far from train distribution? Approximating $p(\mathbf{w}|\mathbf{X}, \mathbf{Y})$ with a Dirac delta function is essentially akin to approximating the predictive distribution using $\mathbf{w}_{\text{MAP}}$, meaning $p(y = 1|x, \mathbf{w}_{\text{MAP}}) \approx p(y = 1|x, \mathbf{Y})$. This approximation is quite rudimentary. As depicted in Figure 2.1, the model's uncertainty doesn't significantly increase far from the training data. This limitation indicates that the point-wise estimate of the parameters can only confidently assign points to their respective classes but lacks the capacity to provide nuanced uncertainty measures for points that deviate far from the training data distribution. Thus, it lacks in providing uncertainty measures for outlying points.

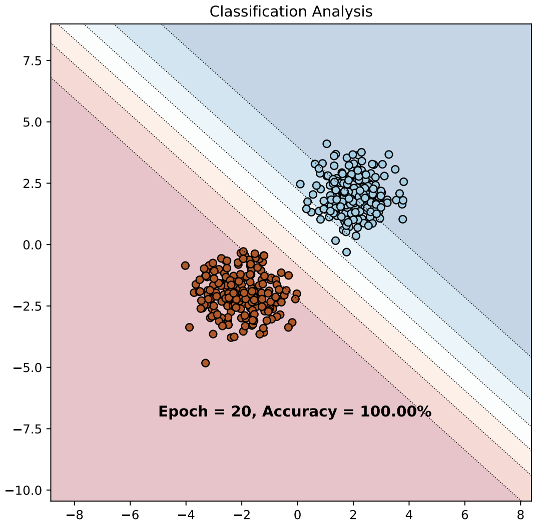


Figure 2.1: Illustration of a Bayesian Logistic Regression model applied to a binary classification task with uncertainty display, with a weight decay of 5×10^{-2}. Two distinct data point clusters — blue and red — represent separate classes, while the surrounding shaded areas reflect the model's predictive uncertainty, with lighter shades indicating lower confidence.

2.1.2 Laplace Approximation

1.2. Analyze the results provided by Figure 2.2. Compared to previous MAP estimate, how does the predictive distribution behave? Compared to the MAP estimate, the predictive distribution in Bayesian Logistic Regression with Laplace approximation captures the uncertainty about the model parameters. While the MAP estimate provides a single point estimate of the weights ($\mathbf{w}_{\text{MAP}}$), thus a single decision boundary, the Laplace approximation models the weights as a normal distribution centered around $\mathbf{w}_{\text{MAP}}$ with a covariance matrix derived from the Hessian of the log posterior, which allows uncertainty in the decision boundary, as clearly displayed in Figure 2.2.

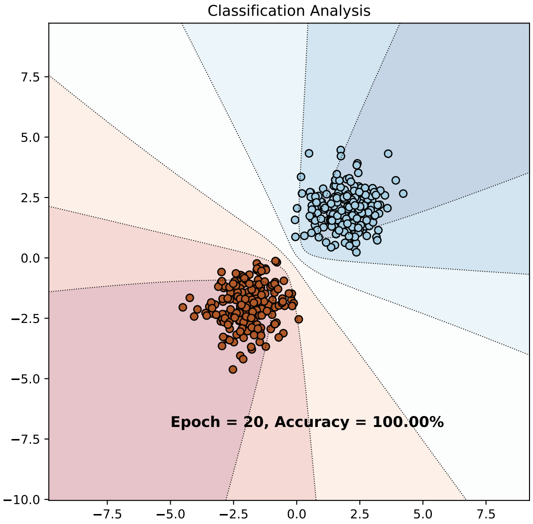


Figure 2.2: Illustration of a Bayesian Logistic Regression with Laplace approximation model applied to a binary classification task, with a weight decay of 5×10^{-2}.

1.3. Comment the effect of the regularisation hyper-parameter `WEIGHT_DECAY`. The weight decay hyper-parameter controls the complexity of the model. It adds a penalty to the loss function for large weights, effectively encouraging the model to maintain smaller weight values. This usually help prevent overfitting by discouraging the model from relying too heavily on any single feature or combination of features.

In Bayesian terms, weight decay corresponds to the precision (inverse variance) of the prior distribution over the weights. A higher weight decay value means a tighter prior, which pulls the weights closer to zero, unless the data provides strong evidence to the contrary. This can affect the predictive distribution by potentially making it more conservative. As a result, the decision boundary may be less flexible and the model may exhibit higher uncertainty, especially in regions far from the training data. This behavior is verified in Figure 2.3. When the weight decay is too high, it results in increased predictive uncertainty (wider shaded areas), whereas when it's too low, it results in high confidence (narrower shaded areas), which may not be justified for unseen data.

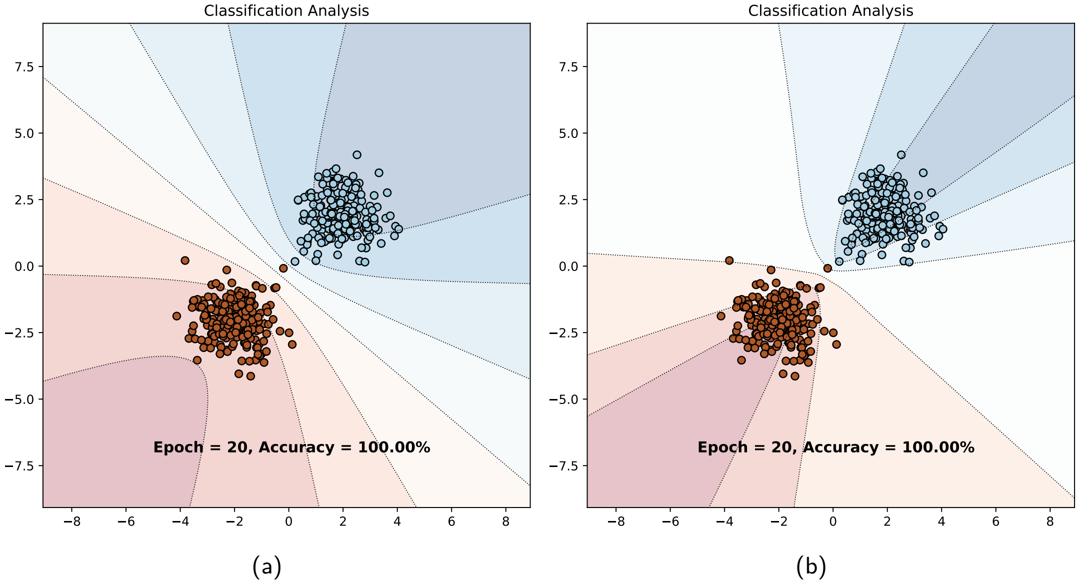


Figure 2.3: Illustration of a Bayesian Logistic Regression with Laplace approximation model applied to a binary classification task, with a weight decay of (a) 0.5 and (b) 5×10^{-5}.

2.1.3 Variational Inference

1.4. Comment the code of the `VariationalLogisticRegression` and `LinearVariational` classes. `LinearVariational` represents a single linear layer with variational inference applied. It approximates the weights and biases of the layer with distributions rather than fixed values.

- The class is initialized with the variational parameters for the weights (`w_mu`, `w_rho`) and bias (`b_mu`) of the layer, i.e. the parameters we want to learn. `prio_var` represents the variance of the prior distribution (σ_p^2), to specify our prior belief about the distribution of the weights.
- The `sampling` method uses the reparametrization trick to sample from the variational posterior distribution for the weights. The reparametrization trick allows the gradient of the loss function to backpropagate through the randomness of the sampling process.
- The `kl_divergence` method calculates the Kullback-Leibler divergence between the variational posterior and the prior distribution for the weights.
- The `forward` method defines the forward pass by sampling the weights then by computing the output of the layer using the sampled weights and the mean of the biases.

`VariationalLogisticRegression` represents a logistic regression model using variational inference:

- The class is initialized with one linear variational layer used to perform the linear transformation in logistic regression.
- The `forward` method defines the forward pass for the logistic regression model by returning the probabilities of the positive class.
- The `kl_divergence` method simply calls the same method of the `LinearVariational` layer to obtain the KL divergence term for the model.

1.5. Comment the code of the training loop, especially the loss computation. Analyze the results provided by Figure 2.4. Compared to previous MAP estimate, how does the predictive distribution behave? What is the main difference between the Variational approximation and the Laplace approximation? The loss function calculates the Evidence Lower Bound (ELBO), which we want to maximize. In theory, we aim to maximize the likelihood of the data directly, but this is often intractable due to the integral over the weights. Therefore, we compute the Kullback-Leibler divergence between the variational distribution $q_\theta(w)$ and the prior distribution $p(w)$. This acts as a regularization term, encouraging the variational distribution to be similar to the prior distribution. It represents the information lost when using $q_\theta(w)$ to approximate $p(w)$, which we want to minimize.

To ensure the model fits the data effectively, we compute the negative log-likelihood of the data under the model parameterized by the weights sampled from $q_\theta(w)$. This is done using a binary cross-entropy loss. Subsequently, we employ gradient descent (thus minimizing the negative ELBO) to update the parameters of the variational distribution to better approximate the true posterior.

Compared to a MAP estimate, the variational approach does not just find the most probable weights (as MAP does) but instead approximates the entire posterior distribution over the weights. In the variational approach, the predictive distribution captures the model's uncertainty about its predictions.

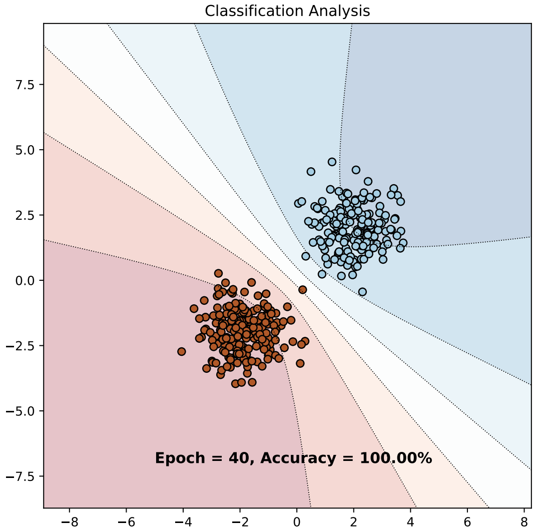


Figure 2.4: Illustration of a Variational Logistic Regression model applied to a binary classification task.

2.2 Bayesian Neural Networks

2.2.1 Variational Inference with Bayesian Neural Networks

2.1. Analyze the results showed on Figure 2.5. ...

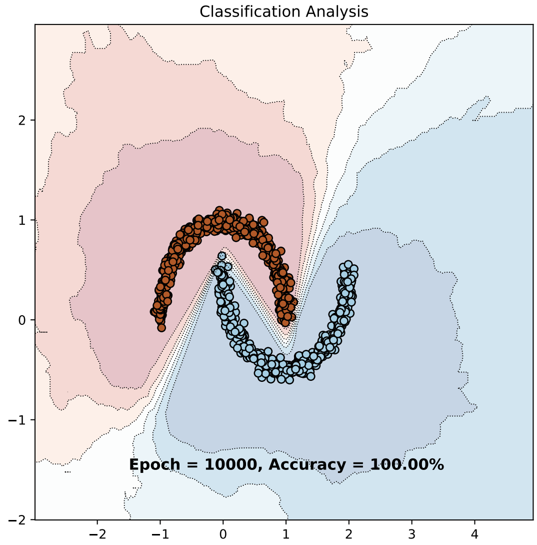
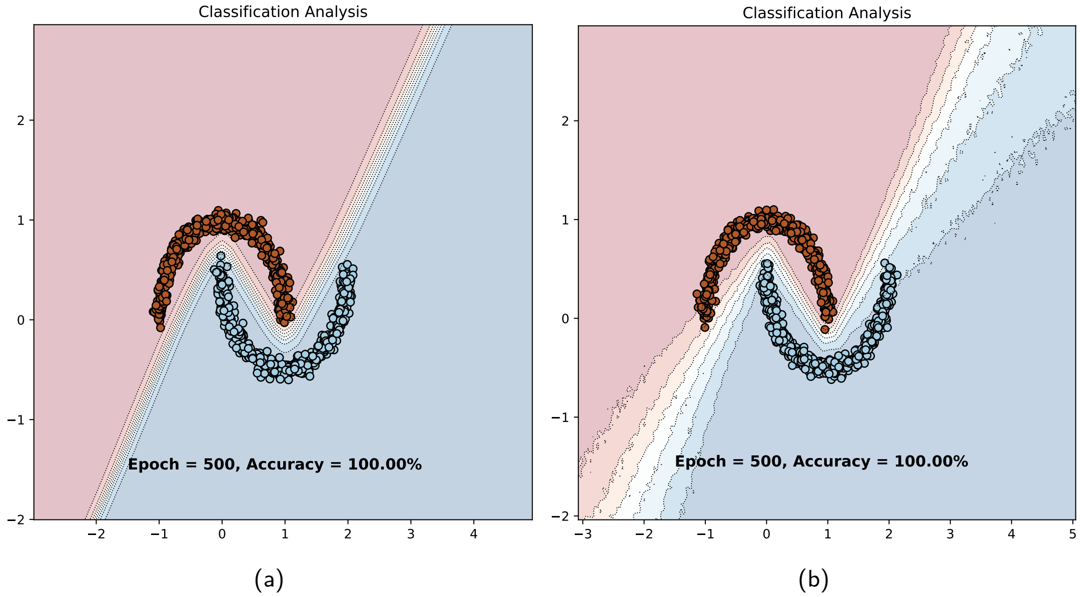


Figure 2.5: Illustration of a Bayesian Neural Network model applied to a binary classification task.

2.2.2 Monte Carlo Dropout

2.2. Again, analyze the results showed on Figure 2.6. What is the benefit of MC Dropout variational inference over Bayesian Logistic Regression with variational inference? ...



(a)

(b)

Figure 2.6: Illustration of a Bayesian Neural Network model applied to a binary classification task using (a) dropout and (b) Monte-Carlo dropout.

Chapter 3

Uncertainty Applications

3.1 Monte-Carlo Dropout on MNIST

1.1. What can you say about the images themselfs? How do the histograms along them helps to explain failure cases? Finally, how do probabilities distribution of random images compare to the previous top uncertain images?

3.2 Failure prediction

2.1. Compare the precision-recall curves of each method along with their AUPR values. Why did we use AUPR metric instead of standard AUROC?

3.3 Out-of-distribution detection

3.1. Compare the precision-recall curves of each OOD method along with their AUPR values. Which method perform best and why?