RDFIA: Homework 4-a,b,c (Bayesian deep learning)

GALMICHE Nathan PIRIOU Victor

February 13, 2023

Abstract

We have studied in previous lectures and practical works many ways to recognize shapes from images. We learned that there are technologies, such as convolutional neural networks, that are able to recognize objects with very high accuracy. The problem is that there are numbers of issues that come with those. For instance, neural networks are not perfectly interpretable. Also, in some use cases, it is desirable to know how confident the model is about its predictions as they may involve human lives. We thus studied a way to have a better understanding of our tools functioning and performances by looking at Bayesian models that estimate the whole posterior distribution whose variance represents the epistemic uncertainty whereas classic neural networks only output the mode of this distribution.

We first started by studying the Bayesian Linear regression. It is a good way to get introduced to them as it represents a very simple case where we can compute the posterior distribution thanks to the existence of its closed form.

During this practical work, this Bayesian framework was also applied on neural networks to do classification. The problem is that the solution is not a closed form anymore and we thus have to estimate it. We did it in different ways. A first way was variational inference which estimates the posterior distribution of the parameters by optimizing the model using a gradient descent. A second way was the use of the Laplace approximation that consists of approximating the posterior by a normal distribution. A third way was by using the Monte-Carlo dropout which involves sampling different parameter configurations where we ignore some of the weights at test time in order to obtain an approximation of the posterior distribution by averaging all of the outputs.

Finally, we applied uncertainty quantification methods to failure prediction and out-of-distribution detection. In order to better predict the uncertainty, we looked at different criteria, other than the most voted class. They include the variation ratio, the mutual information and the entropy.

1 Bayesian Linear Regression

1.1 Linear Basis function model

Question 1.2 : Recall closed form of the posterior distribution in linear case. Then, code and visualize posterior sampling. What can you observe?
The closed form of the posterior distribution in linear is:

$$p(\mathrm{w} \mid \mathrm{X}, \mathrm{Y}) = \mathcal{N}(\mathrm{w} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}),$$

with the mean of the parameters' distribution defined by:

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

and the inverse of the co-variance matrix of the parameters' distribution defined by:

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

where:

- $\mathrm{Y} \in \mathbb{R}^{N \times K}$ is the ground truth. Here N is the number of training examples while K is the number of dimensions of the ground truth,
- $\Phi \in \mathbb{R}^{N \times (p+1)}$ is the matrix containing the data of dimensionality p and the bias,
- $\mathbf{W} \in \mathbb{R}^{(p+1) \times K}$ is the matrix containing the parameters,
- α is the parameter controlling the prior $p(\mathrm{w} \mid \alpha) = \mathcal{N}\left(\mathrm{w}; 0, \alpha^{-1} I\right)$,
- $\beta = \frac{1}{2\sigma^2}$ is the parameter controlling the likelihood $p\left(\mathrm{y}_i \mid \mathrm{x}_i, \mathrm{w}\right) = \mathcal{N}\left(\Phi_i^T \mathrm{w}, \beta^{-1}\right)$.

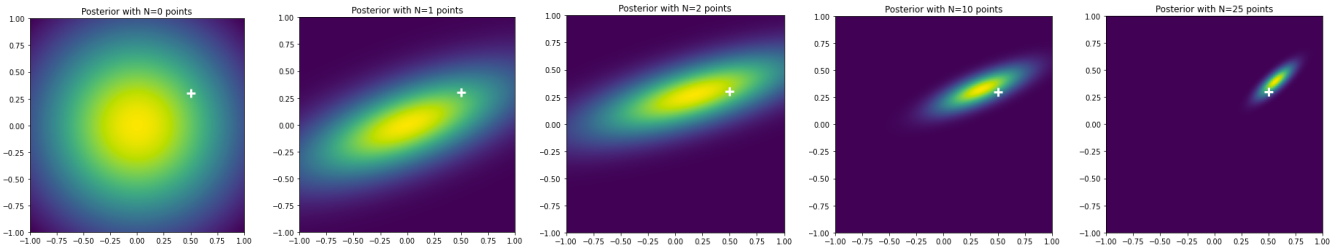


Figure (1) Posterior samplings with different number of points.

In Figure 1, the more data points we have the more the variance of the parameters' distribution is reduced and the closer the estimate gets to the ground truth (represented in white). In other words, the more data points we have the more the aleatoric uncertainty over the parameters is reduced.

Question 1.3 : Recall and code closed form of the predictive distribution in linear case.
The closed form of the predictive distribution in linear case is defined by:

$$p\left(y \mid x^*, \mathcal{D}, \alpha, \beta\right) = \mathcal{N}\left(y; \mu^T \boldsymbol{\Phi}\left(\mathrm{x}^*\right), \frac{1}{\beta} + \boldsymbol{\Phi}\left(\mathrm{x}^*\right)^T \boldsymbol{\Sigma} \boldsymbol{\Phi}\left(\mathrm{x}^*\right)\right),$$

where $\mathcal{D}$ is the dataset and x^* a new input.

Question 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.

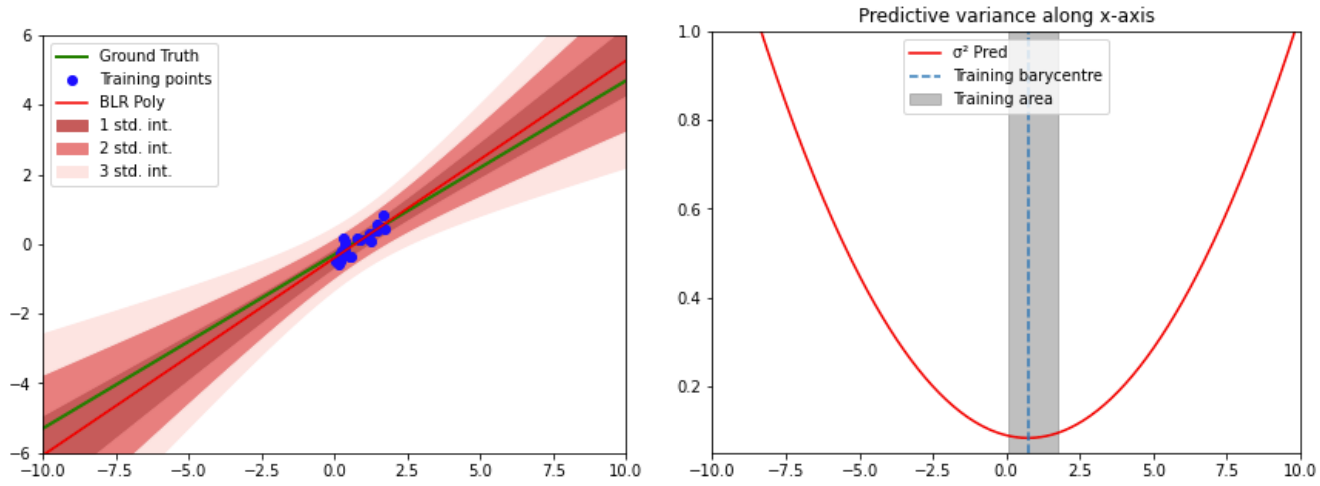


Figure (2) Visualization of the predictive distribution of a test dataset.

Question 1.5 : Analyse these results. Why predictive variance increases far from training distribution? Prove it analytically in the case where $\alpha = 0$ and $\beta = 1$.
We can notice that:

- the further we are from the data, the bigger the variance is,
- the variance is minimal at the barycenter of the training points.

Let's prove it analytically in the case where $\alpha = 0$ and $\beta = 1$.

We have:

$$\begin{aligned}\Sigma^{-1} &= \alpha \boldsymbol{I} + \beta \Phi^T \Phi \\ &= \Phi^T \Phi \\ &= \begin{pmatrix} 1 & \dots & 1 \\ x_1 & \dots & x_N \end{pmatrix} \cdot \begin{pmatrix} 1 & x_1 \\ \dots & \dots \\ 1 & x_N \end{pmatrix} = \begin{pmatrix} N & \sum_i x_i \\ \sum_i x_i & \sum_i x_i^2 \end{pmatrix}.\end{aligned}$$

This implies that:

$$\Sigma = \frac{1}{det(\Sigma^{-1})} \begin{pmatrix} \sum_i x_i^2 & -\sum_i x_i \\ -\sum_i x_i & N \end{pmatrix}.$$

Then, the variance of a new data point x^* is obtained by:

$$\begin{aligned}\Phi\left(\mathrm{x}^*\right)^T \Sigma \Phi\left(\mathrm{x}^*\right) &= \frac{1}{det(\Sigma^{-1})} \begin{pmatrix} 1 & x^* \end{pmatrix} \cdot \begin{pmatrix} \sum_i x_i^2 & -\sum_i x_i \\ -\sum_i x_i & N \end{pmatrix} \cdot \begin{pmatrix} 1 \\ x^* \end{pmatrix} \\ &= \frac{1}{det(\Sigma^{-1})} \cdot \begin{pmatrix} \sum_i x_i^2 - x^* \sum_i x_i & -\sum_i x_i + N x^* \end{pmatrix} \cdot \begin{pmatrix} 1 \\ x^* \end{pmatrix} \\ &= \frac{1}{det(\Sigma^{-1})} \cdot \Big(\sum_i x_i^2 - 2x^* \sum_i x_i + N (x^*)^2 \Big) \\ &= \frac{1}{det(\Sigma^{-1})} \cdot \Big(\sum_i (x_i - x^*)^2 \Big).\end{aligned}$$

As the determinant is constant, it is now clear that the further we are from the training examples the bigger $(x_i - x^*)^2$ is and the bigger the variance is.

Bonus Question : What happens when applying Bayesian Linear Regression on the following dataset?

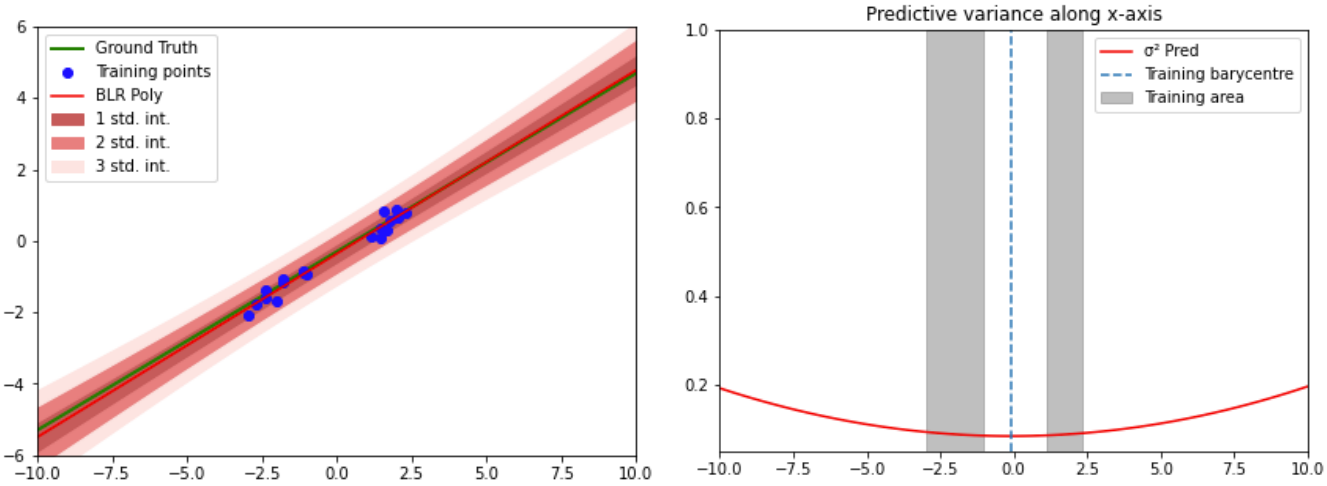


Figure (3) Visualization of the predictive distribution of a test dataset.

On Figure 3 we can see that the predictive variance is still minimized at the barycenter of the training points. This fact is not desirable this time. Indeed, we clearly have two separated clusters of points so the variance should be minimized around them and bigger the further we get away from these clusters.

1.2 Non Linear Models

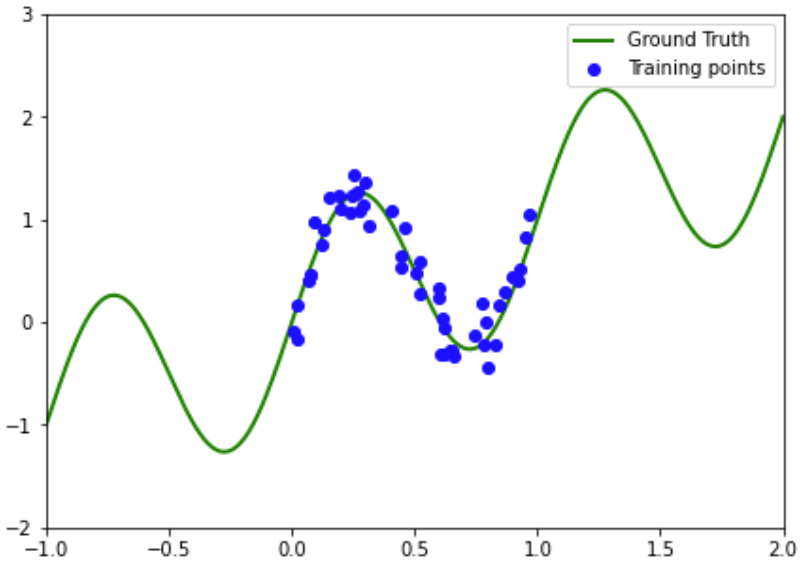


Figure (4) Dataset of an increasing sinusoidal curve.

1.2.1 Polynomial basis functions

Question 2.2 : Code and visualize results on sinusoidal dataset using polynomial basis functions. What can you say about the predictive variance?

On Figure 5, the predictive variance gets bigger as we move away from the training points. This is expected, however, it looks like the model does not capture the periodicity of the sinusoidal ground truth because the model has no

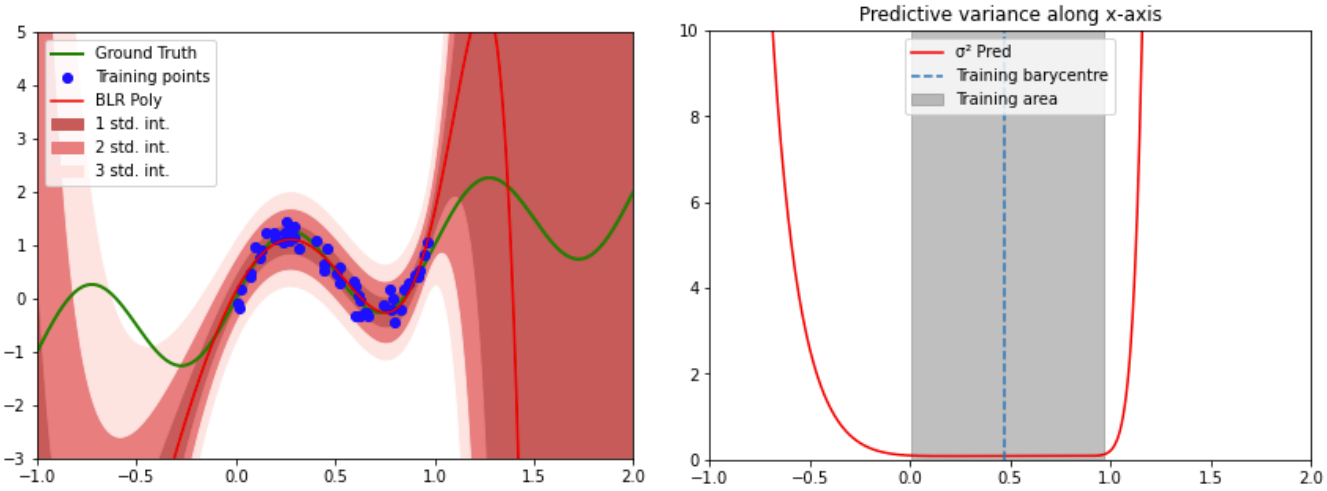


Figure (5) Visualization of the predictive distribution of a test dataset.

data to know which values to predict. It thus predicts a continuation of the first data that do not give information on the previous and next periods. We thus notice that the ground truth gets very far from the prediction to the point where the variance does not even include it which is very bad if we were to predict in this range.

1.2.2 Gaussian basis functions

Question 2.4 : Code and visualize results on sinusoidal dataset using Gaussian basis functions. What can you say this time about the predictive variance?

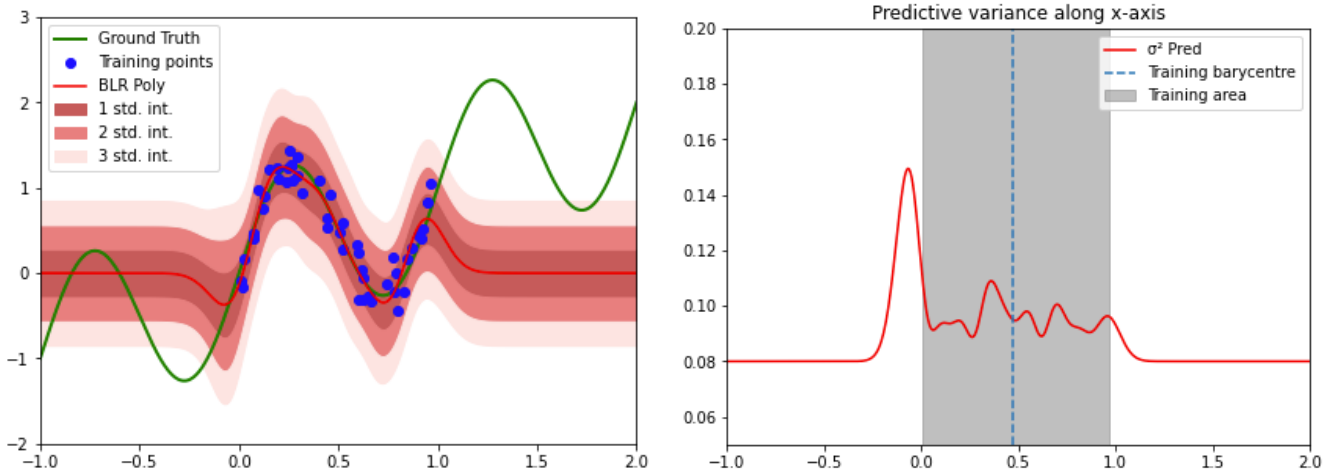


Figure (6) Visualization of the predictive distribution of a test dataset.

Figure 6 shows that compared to polynomial basis functions the use of the Gaussian basis functions is associated to a smaller variance and a constant prediction in the area where we are far from the training points. It fits very well to the data with the variance fluctuating depending on the concentration of the data. However this time, in the zones where we do not have any data, we see that the variance does not really increase and keeps a pretty low value while the prediction is constant. This not what we want because we see that we are getting really far from the ground truth.

Question 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.
This is explained by the fact that the epistemic uncertainty $\Phi(\mathrm{x}^*)^T \Sigma \Phi(\mathrm{x}^*)$ converges to 0 so $\sigma^2_{\text{pred}} = f(\mathrm{x}^*) =$

$\beta^{-1} + \Phi(\mathrm{x}^*)^T \Sigma \Phi(\mathrm{x}^*)$ becomes $\sigma^2_{\text{pred}} = f(\mathrm{x}^*) = \beta^{-1}$ and is therefore reduced to the aleatoric uncertainty. In our case, $\beta^{-1} = 0.08$ which fits the graph we obtained.

2 Approximate inference

2.1 Bayesian Logistic Regression

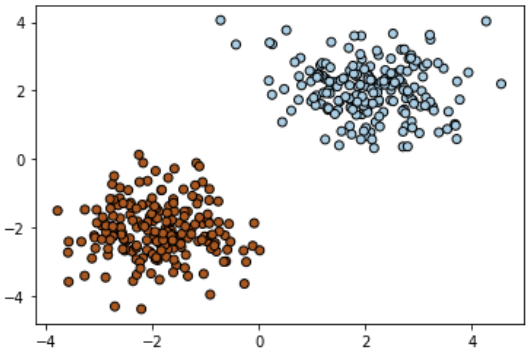


Figure (7) Visualization of the used dataset that is linearly separable.

Question 1.1 : Analyze the results provided by previous plot. Looking at $p(\mathrm{y} = 1|\boldsymbol{x}, \boldsymbol{w}_{\mathbf{MAP}})$, what can you say about points far from train distribution?
We can see on Figure 8 that the uncertainty does not increase when we are far from the training data. In fact, as we obtain a point wise estimate of the parameters, it is only able to say in which class each point belongs and thus cannot take into account the amount of information we originally had around.

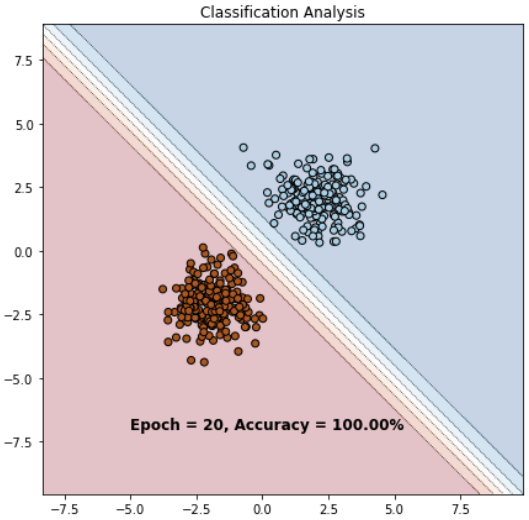


Figure (8) Decision boundary obtained after having trained a Bayesian logistic regression using the plug-in approximation.

Question 1.2 : Analyze the results provided by previous plot. Compared to previous MAP estimate, how does the predictive distribution behave?

We can see on Figure 9 that this time the uncertainty increases when we are far from the training data. Computing the Hessian matrix allows to obtain an evaluation of the variance at each point and we are thus able to show some sort of uncertainty. This gaussian approximation is not perfect since it is only done at the mode of the distribution. Consequently, the global properties of the distribution are ignored.

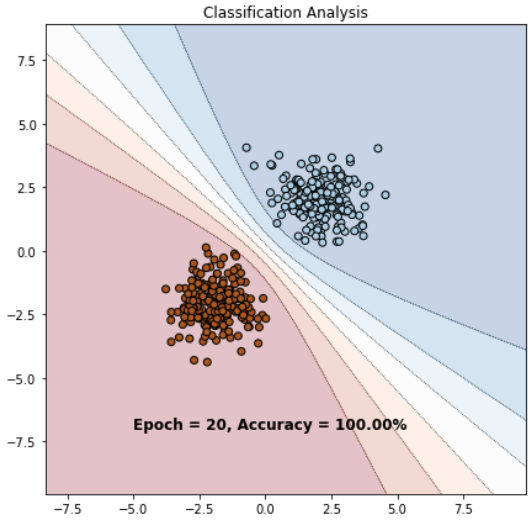


Figure (9) Decision boundary obtained after having trained a Bayesian logistic regression using the Laplace approximation.

Question 1.3 : Analyze the results provided by previous plot. Compared to previous MAP estimate, how does the predictive distribution behave?

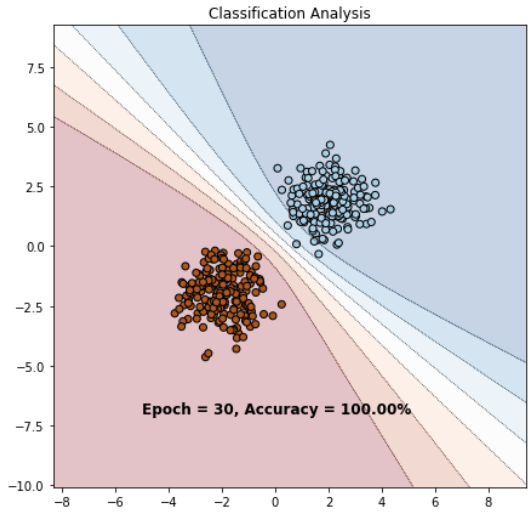


Figure (10) Decision boundary obtained after having trained a variational logistic regression.

This technique allows to estimate the posterior distribution that cannot be evaluated analytically. This means that we are able to evaluate for each point which class to put it in and how certain we are. We obviously find similar results as the previous technique because the data is pretty simple but the estimation is correct.

2.2 Bayesian Neural Networks

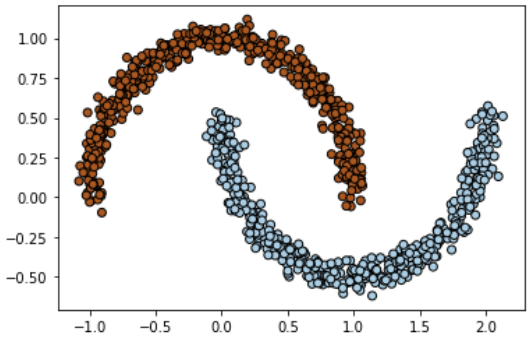


Figure (11) Visualization of the two moons' dataset whose classes are not linearly separable.

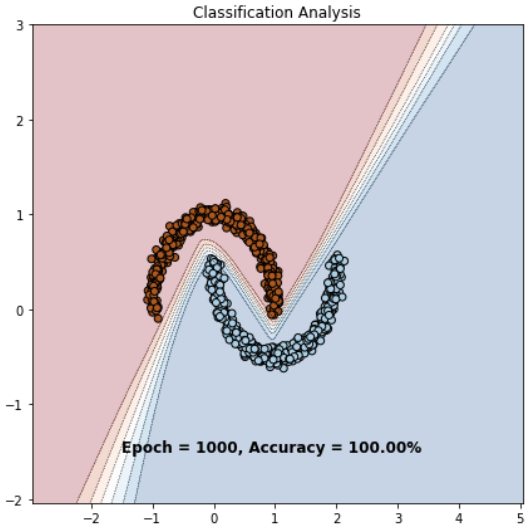
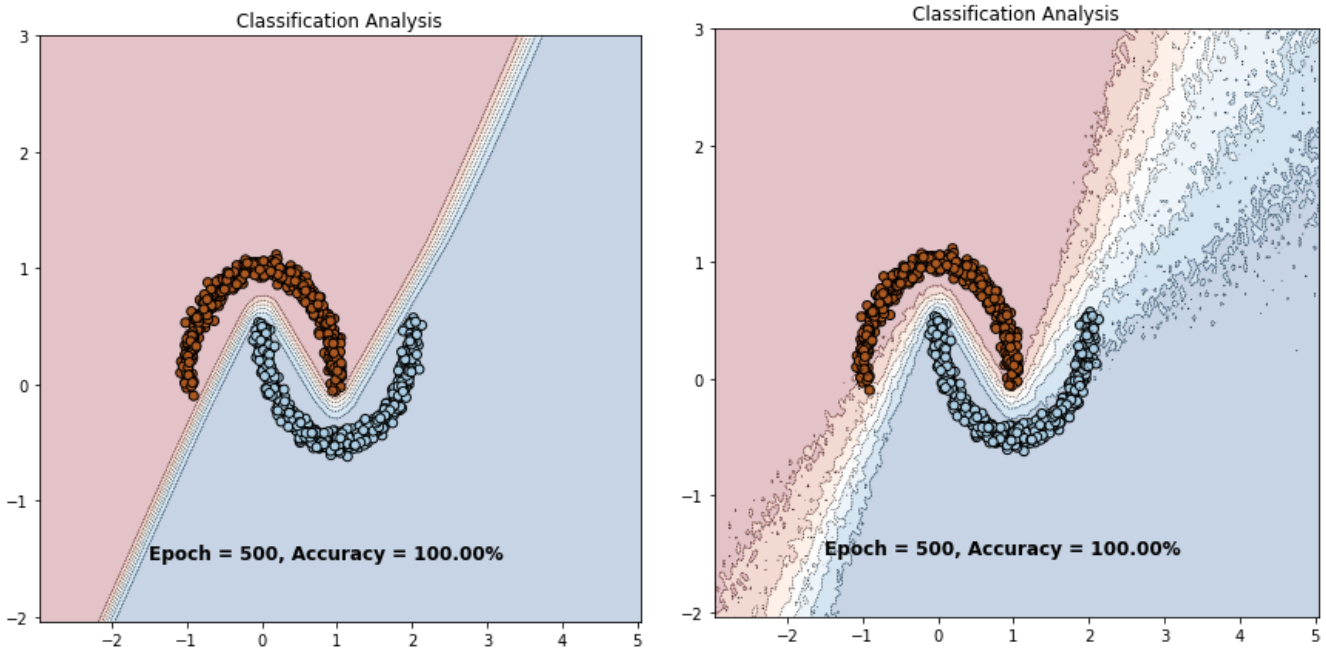


Figure (12) Decision boundary obtained after having trained a variational MLP.

2.2.1 Monte Carlo Dropout

Question 2.1 : Again, analyze the results showed on plot. What is the benefit of MC Dropout variational inference over Bayesian Logistic Regression with variational inference?
First we notice that the Bayesian Logistic Regression shows good results as we see that the uncertainty increases when we get further from the data. It is still precise where there is data.



(a) Without Monte Carlo Dropout

(b) With Monte Carlo Dropout

Figure (13) Decision boundaries obtained after having trained a MLP with dropout layers.

However using Monte Carlo dropout gives us very different results this time. The first thing we notice is that the limits between the classes is not as straight. We see that there are some irregularities that come from the dropout. Those are not very important but we also notice that the uncertainty is much more important with this technique as some is introduced right by the data that are at the extreme. This could seem interesting since the model cannot be precise when we are far from any data but when the data is separable like this, we should not need to introduce uncertainty close to the data. Here, we see that the extreme ends of the moons are less certain than the rest, which is not needed.

The rate of the dropout can also be modified to lessen or heighten this effect.

3 Applications of uncertainty

3.1 Monte-Carlo Dropout on MNIST

Question 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?

In general in this dataset, the number we are looking at in the image is pretty big and fully contrasted so it is clear. However when looking at the worst images, it can be hard even for our human brain to understand what number does it represent. We thus expect the model to struggle. This is very visible in histograms when you have different classes with similar probabilities. Also, when looking more into the details of each classes, we look at histograms showing the confidence of every prediction for the class. We notice that when the model struggles, the histogram is either pretty balanced which means it is sometimes confident and sometimes not, or the histogram is dominant on the lower confidences.

This is very different than the good images where we notice that there is only one bar on each histogram : only one class predicted, always the same level of confidence (very high for the right class, very low for the wrong ones).

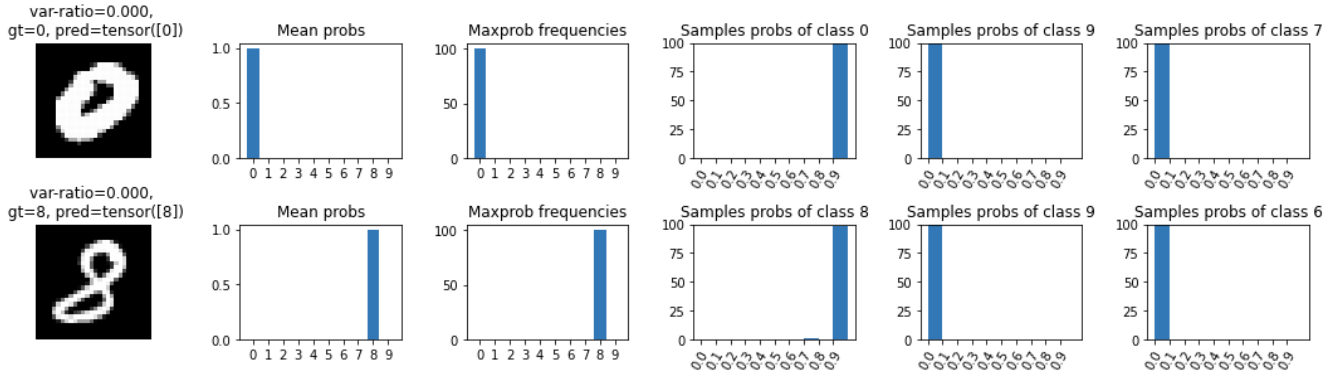


Figure (14) Random images with their var-ratios value.

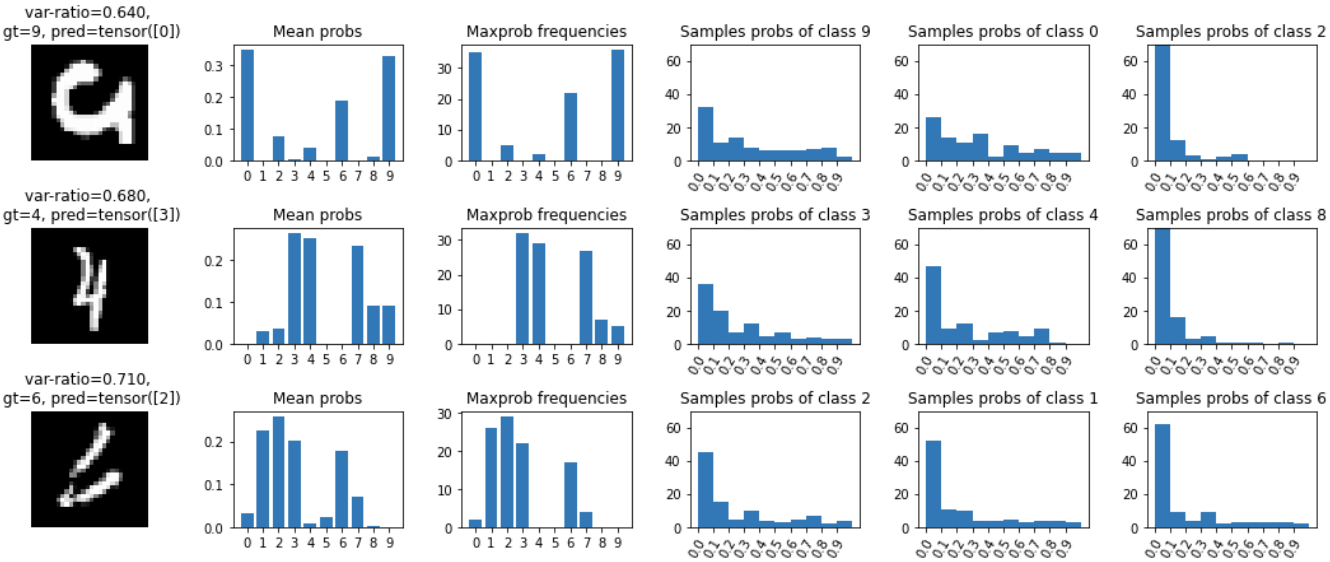


Figure (15) Top-3 most uncertain images along with their var-ratios value.

4 Failure prediction

Question 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?

Globally, we can see on Figure 16 that the area under the curve of ConfidNet is much more big than that of the other methods. More precisely, ConfidNet has both an higher precision and a higher recall values. In sum, it is much more successful than the MCP and MCPDropout methods.

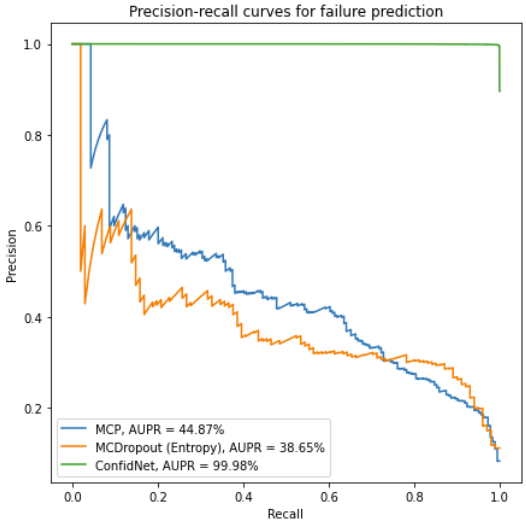


Figure (16) Precision-recall curves along with their AUPR values for failure prediction.

The AUPR metric does not make use of the True Negatives at all. Hence, in our case the AUPR metric is more suitable than the standard AUROC because the classes are imbalanced. The AUROC metric would ignore a lot of errors, which is not what we want.

5 Out-of-distribution detection

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

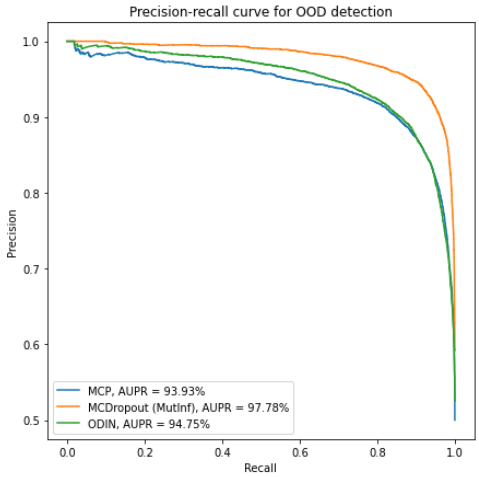


Figure (17) Precision-recall curves along with their AUPR values for out-of-distribution detection.

The best method is the MCdropout which is expected because we get more information thanks to the sampling. The ODIN idea improves a little bit the score of the simple MPC but it is still worst than the MCDropout.