Deep Learning Practical Work 4

SIHAMDI Mostefa, BOUSBA Abdellah

UE RDFIA 2021-2022, Encadrants : Nicolas Thome , Charles Corbiere, Remy Sun

M2 DAC

I. Bayesian Linear Regression:

A. Linear Basis function model:

- Question 1.1:

The closed form of the posterior distribution in linear case is (cf. course)

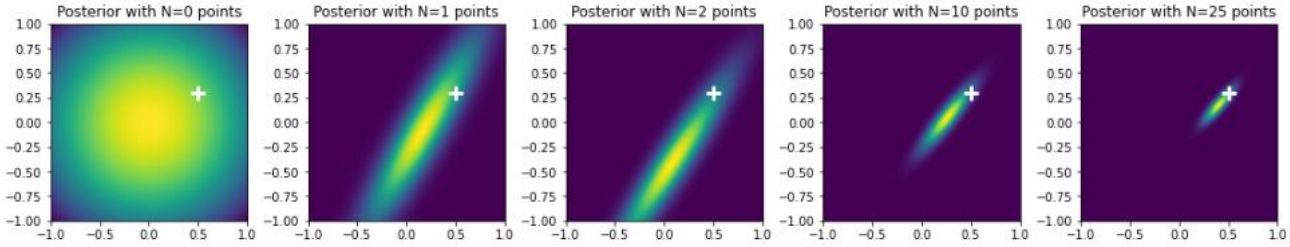
$$p(w/xi, yi) \propto p(yi/xi, w)p(w)$$

$$p(w/X, Y) = N(w \mid \mu, \Sigma)$$

with : $\Sigma^{-1} = \alpha I + \beta \Phi^T \Phi$ $\mu = \beta \Sigma \Phi^T Y$

- Question 1.2:

The images below represent the weights distribution (posterior), the white cross is the optimal parameters. When the umber of points is 0 prior and posterior are equals and the more the number of points increases the more the variance decreases and we get closer to the optimal parameters.



- Question 1.3:

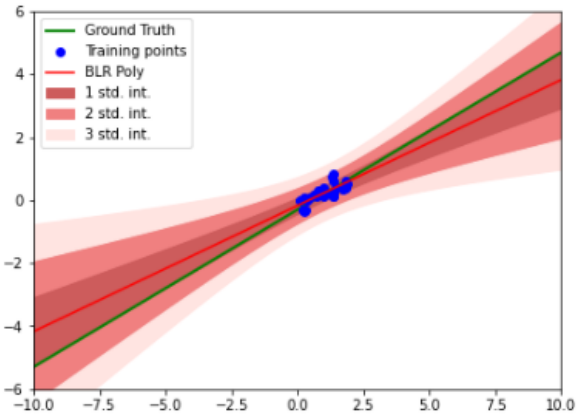
the closed form of the predictive distribution in linear case is (cf. course)

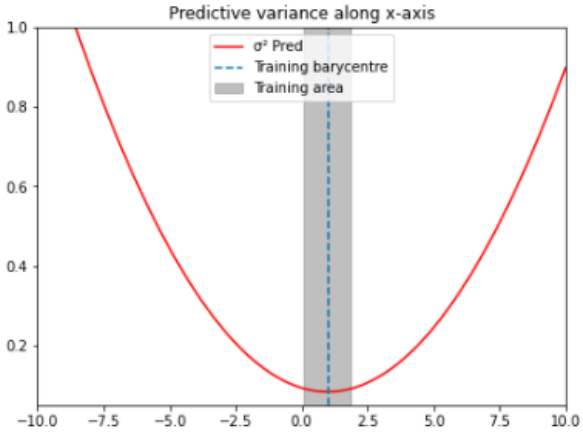
p(w|D, α, β) ⇒ compute predictive distribution by marginalizing over w

$$p(y^* \mid x^*, D, \alpha, \beta) = \int p(y^* \mid x^*, w, \beta) p(w \mid D, \alpha, \beta) dw$$

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

- Question 1.4:





On the left we can see the prediction line with red and the ground truth with green, it's clear that the more we distance our self from the training points the more we have more error and it's due to the fact that the variance increases the further we are from them. As for the right figure we project the variance on one dimension for more clarity, we can notice the same thing the model is more confident of the prediction on the barycenter of the data as the variance is minimal.

As for the analytic proof in the case where α=0 and β=1:

$$\Sigma^{-1} = \alpha I + \beta \Phi^T \Phi = \Phi^T \Phi = \begin{pmatrix} n & \Sigma xi \\ \Sigma xi & \Sigma xi^2 \end{pmatrix}$$

So : $$\Sigma = \frac{1}{\det \Sigma^{-1}} \begin{pmatrix} \Sigma xi^2 & -\Sigma xi \\ -\Sigma xi & n \end{pmatrix} = \frac{1}{n\Sigma xi^2 - (\Sigma xi)^2} \begin{pmatrix} \Sigma xi^2 & -\Sigma xi \\ -\Sigma xi & n \end{pmatrix}$$

And the predictive variance is : $\sigma^2(x^*) = \frac{1}{\beta} + \Phi(x^*)^T \Sigma \Phi(x^*) = \Phi(x^*)^T \Sigma \Phi(x^*)$

$$\sigma^2(x^*) = \begin{pmatrix} 1 & x^* \end{pmatrix} \Sigma \begin{pmatrix} 1 \\ x^* \end{pmatrix} = \frac{\Sigma xi^2 - x^* \Sigma xi - \Sigma xi + n x^* \begin{pmatrix} 1 \\ x^* \end{pmatrix}}{n\Sigma xi^2 - (\Sigma xi)^2}$$

$$\frac{\Sigma xi^2 - x^* \Sigma xi + x^*(-\Sigma xi + n x^*)}{n\Sigma xi^2 - (\Sigma xi)^2} = \frac{\Sigma xi^2 - 2x^* \Sigma xi + n(x^*)^2}{n\Sigma xi^2 - (\Sigma xi)^2}$$

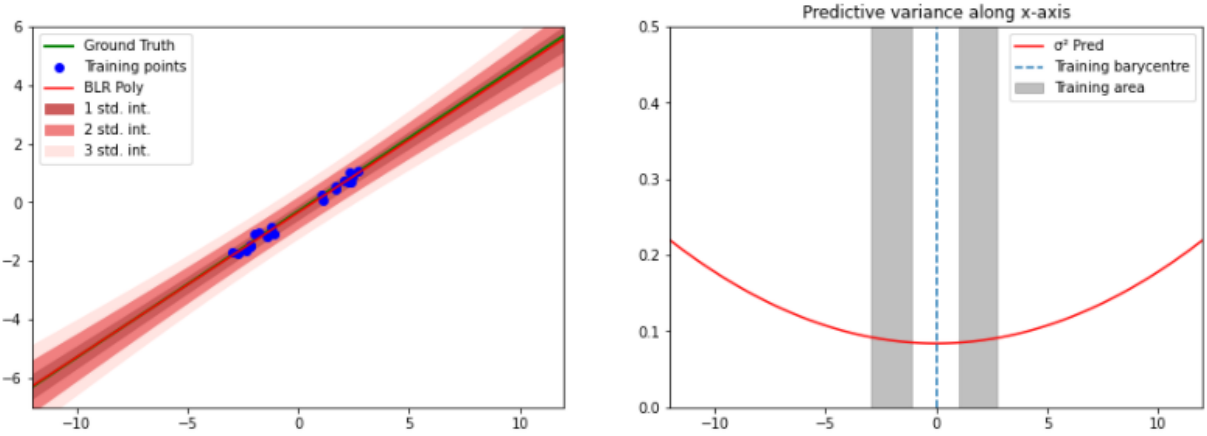
$$= \frac{n\left(\frac{\Sigma xi^2}{n} - \frac{2x^* \Sigma xi}{n} + (x^*)^2\right)}{n^2\left(\frac{\Sigma xi^2}{n} - \frac{(\Sigma xi)^2}{n^2}\right)} = \frac{\left(\frac{\Sigma xi^2}{n} - 2x^* \bar{x} + (x^*)^2\right)}{n\left(\frac{\Sigma xi^2}{n} - \bar{x}^2\right)}$$

$$= \frac{\frac{\Sigma xi^2}{n} - \bar{x}^2 + (x^* - \bar{x})^2}{n\, var(X)} = \frac{var(X) + (x^* - \bar{x})^2}{n\, var(X)}$$

$$= \frac{1}{n} + \frac{1}{n\, var(X)} (x^* - \bar{x})^2$$

According to this result the variance is at its minimum when $x^* = \bar{x}$ the center of the data.

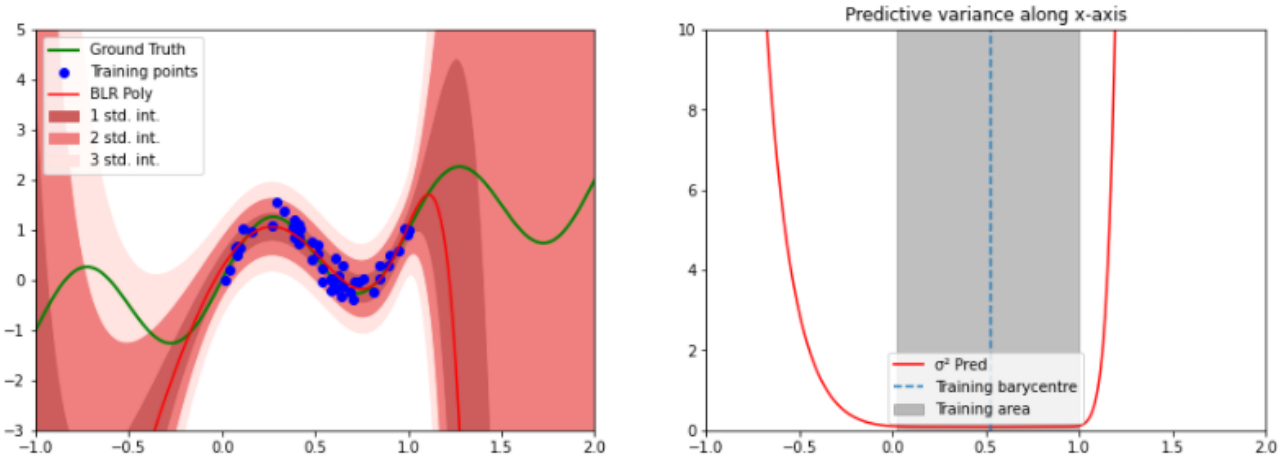
- Bonus question:



We can notice that by applying the model on data separated on two groups the variance decreases but still at its minimum on the center of the data even though there is no data at the center which is not the expected behavior and thus we conclude that this model isn't adapted for this kind of problem.

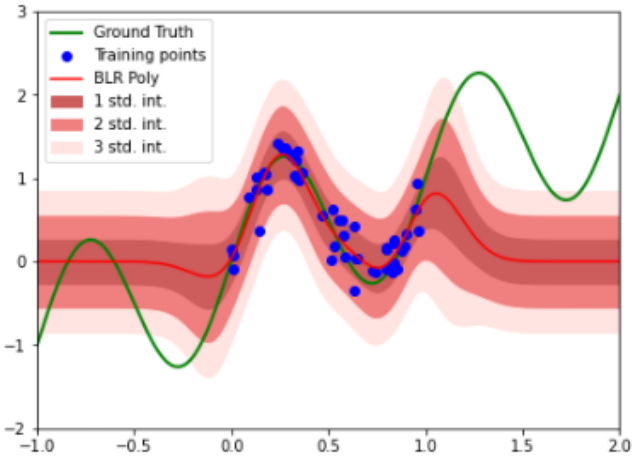
B. Non-Linear models:

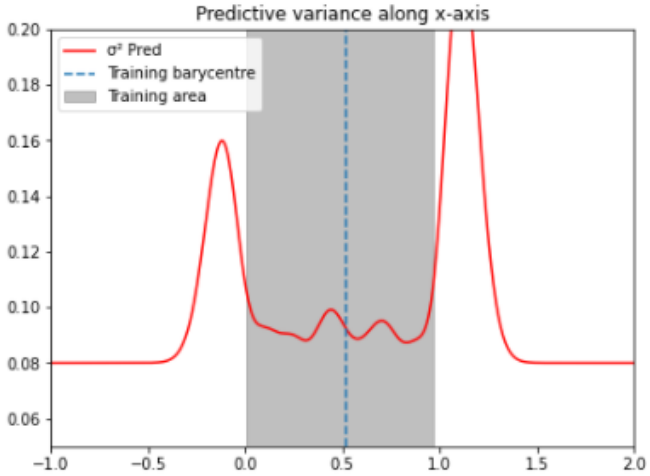
- Question 2.1:



We can notice that the predictive variance increases a lot faster when we get away from the training data and the error between the prediction and the ground truth is a lot bigger. On the other hand, the minimum variance is not longer the center of data but the range 0 to 1.

- Question 2.2:





The prediction is still near perfect around the data and gets worst when we get away. But the variance is low around the data, makes some high spikes the moment we get outside the range of data then decreases again to around 0.08. This behavior isn't expected and the variance should have increased.

- Question 2.3:

When using gaussian features the variance converges to 1/ β this is due to the radial base kernel.

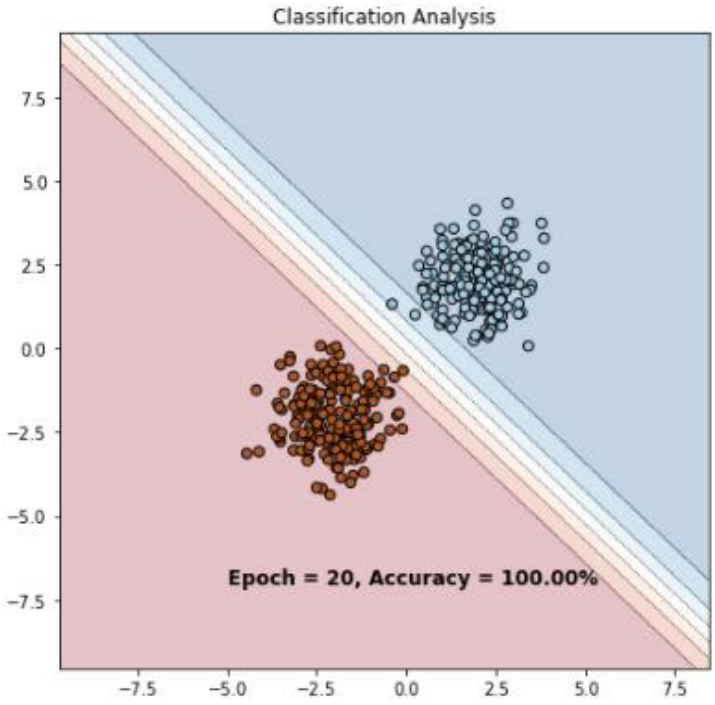
β = $1/2\sigma^2$ with $\sigma = 0.2$, 1/ β = 0.08 which is the value of random uncertainty found at the previous question.

II. Approximate Inference in Classification:

A. Bayesian Logistic Regression:

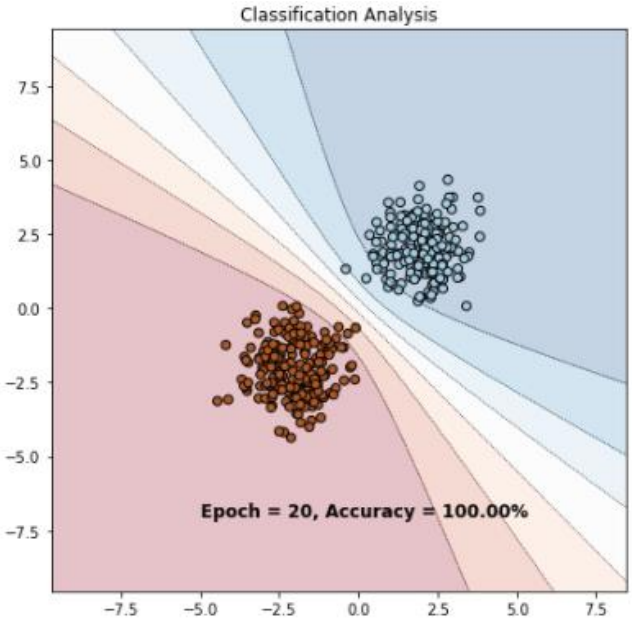
- Question 1.1:

We can notice that the decision boundary stays linear regardless of the training data. Although the prediction is perfect but the model is very confident even if we get away from the decision boundary which is not the desired outcome.



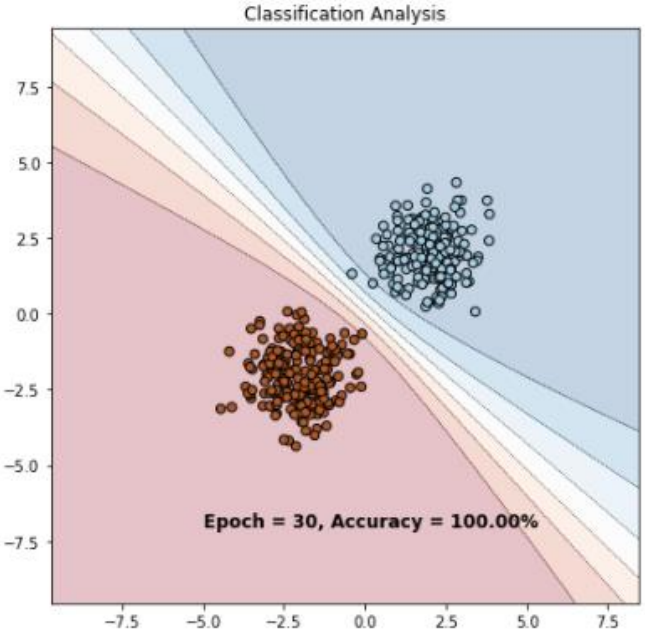
- Question 1.2:

Compared to the previous MAP the accuracy is the same but less colored (less confident) regions around the decision boundary appeared on the form of a curve which gets wider the more we get further away from the training data.



- Question 1.3:

This time we used a different method but we have similar results. The same accuracy and more or less the same decision boundary slightly shifted towards the right and the less confident regions are smaller



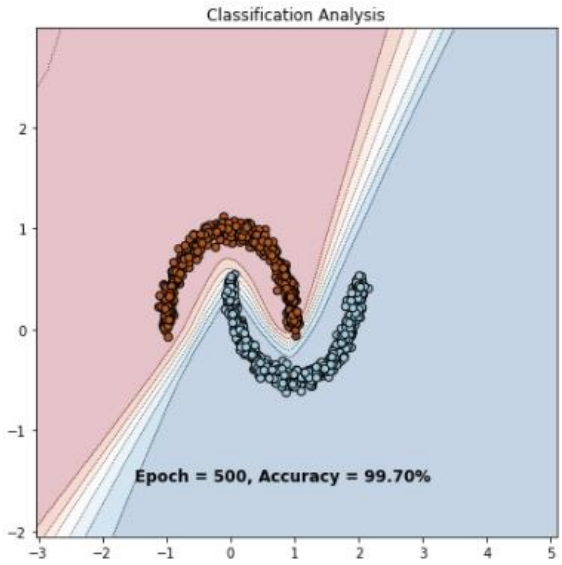
- About LinearVariational class:

It's a class that has a linear layer and parameters according to the variational inference. It contains attributes such as the variance and expectation of both *w* and *b*, also the *parent* attribute that keeps track of the KL divergence in case of multiple layers. We want to learn variational distribution of the weights with minimal distance (KL divergence), so at each forward pass we sample the weights using the reparameterization trick that will be used for the linear calculation. Then using its *output*, *mu_theta* and *rho_theta* we calculate the KL divergence.

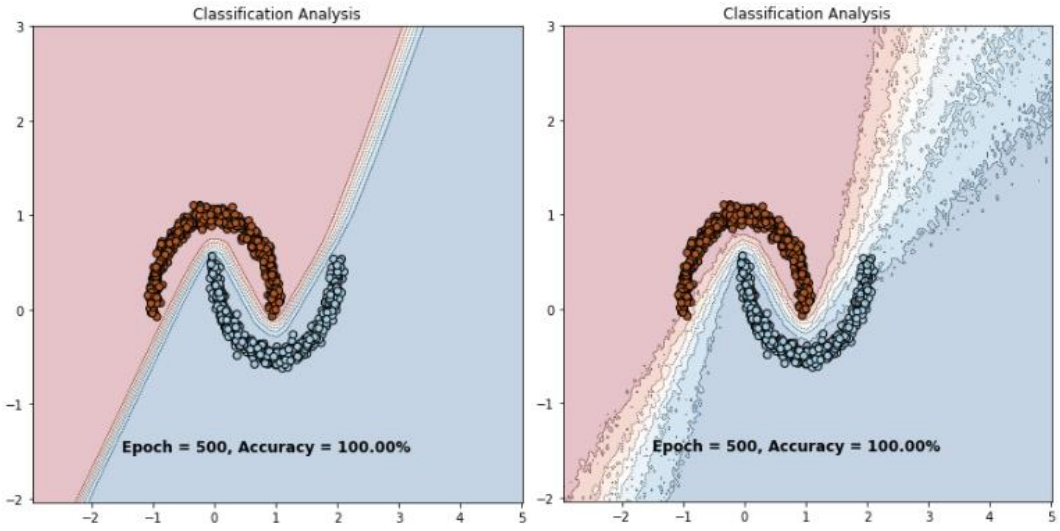
B. Bayesian Neural Networks:

- Question 2.1:

Using Bayesian Neural Networks, the prediction is near perfect but the non-confident regions are less spread compared to the linear problem



Now using the dropout we obtain perect accuracy. The incertenty regions are very tight and close to the boundary region. On the other hand using MC dropout shows good results, the incertenty regions are more spread the further we get from the data.

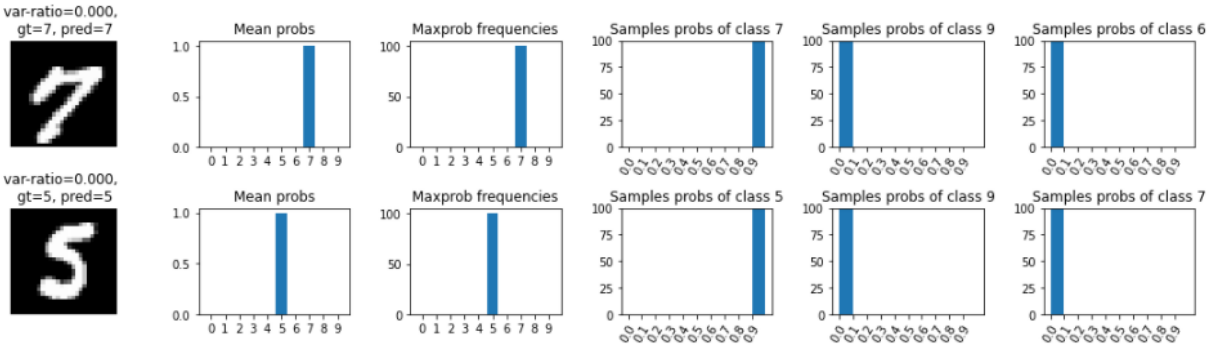


The benefits od the MCDropout method is that we can control the dropout rate, it is easier to implement since we no longer have complex calculations on the weights.

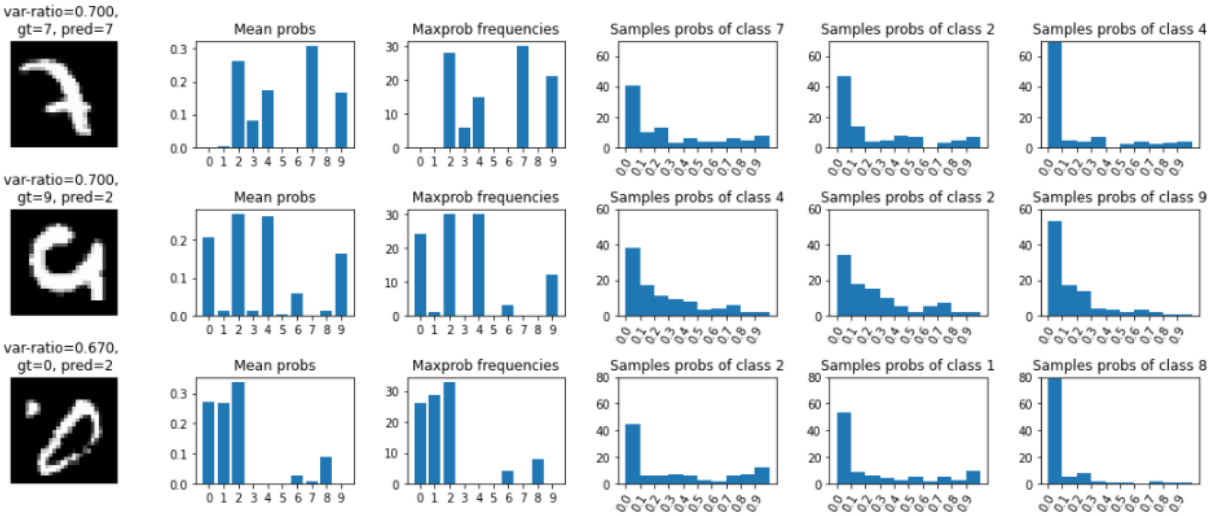
III. Uncertainty Applications:

A. Monte-Carlo Dropout on MNIST:

- Question 1.1:



Images taken randomly have zero variance and it's easy to predict perfectly, they also look very easy to identify with the human eye.



As for the most uncertain images the variance is of course the highest 2 out of 3 are wrong. The images are more or less easy to confuse it with another digit using the human eye, or example the 9 is very close to 4,0 and 2 which is shown on the mean probs plot. As shown for the first image even though it's classified correctly the model was only 30% sure of it. We can also notice that compared to the random images the samples probs have some spikes on multiple labels which shows the uncertainty on these images.

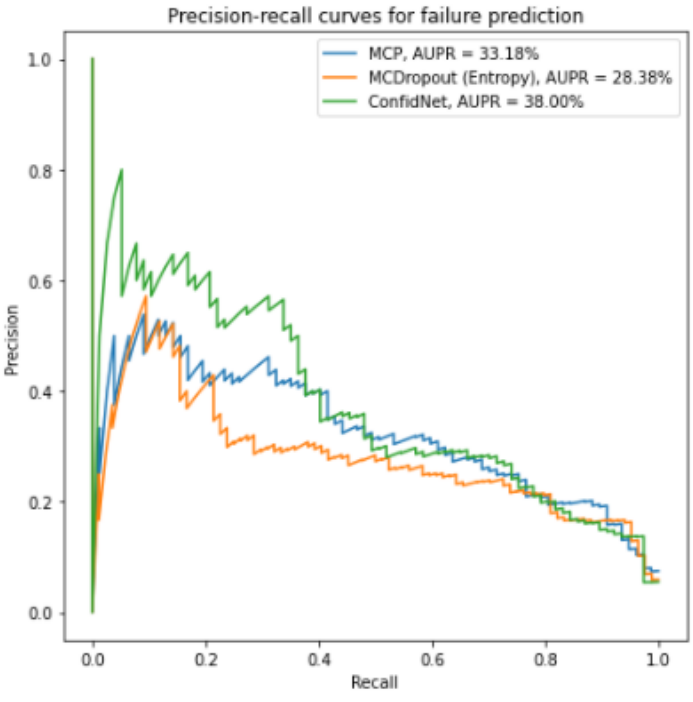
B. Failure prediction:

- The goal of failure prediction:

We use uncertainty to accept or reject some predictions. The model can judge if the uncertainty is so high that he can't correctly predict, and this is based on an uncertainty threshold, for example if the true class probability is less than 1/n with n number of classes its very likely that the model is wrong and that's what we can failure prediction. So, it is used to monitor and predict the potential failure occurrences.

- Question 2.1:

The ROC curve makes it possible to visualize a balance between the rate of true positives and the rate of false positives in a classification. But with an unbalanced data like our case there will be a bias because of the high number of true negatives. In consequence, we will get the impression that the model has good performance but in reality, it is not. The alternative is the PR curve that is used to check for only the positive class (which is the errors in our case).



We can notice that all three methods have weak results, the precisions decline the more data is used. ConfidNet has the highest score specially when the recall is low. The expected result is to have high recall and precision at the same time.

C. Out-of-distribution detection:

- Question 3.1:

The results this time are almost perfect, and the model that we used has very close performances. All three curves are leaned towards the top right which means the precision and recall are very high.

The best model was ODIN with 98.69%, and that is probably because it is easier for this model to differentiate between the in-distribution elements by increasing its MCP higher than out-distribution MCP.