Reconnaissance des formes pour l'analyse et l'interprétation d'images

Rapport TP 1-2-3: Bayesian Deep Learning

Etudiant :	**Numéro**
DJEGHRI Amine	3801757
MAMOU Idles	3803676

Fevrier 2021

TP 1: Bayesian Linear Regression

Q1.1 : **Closed-form for the posterior** :

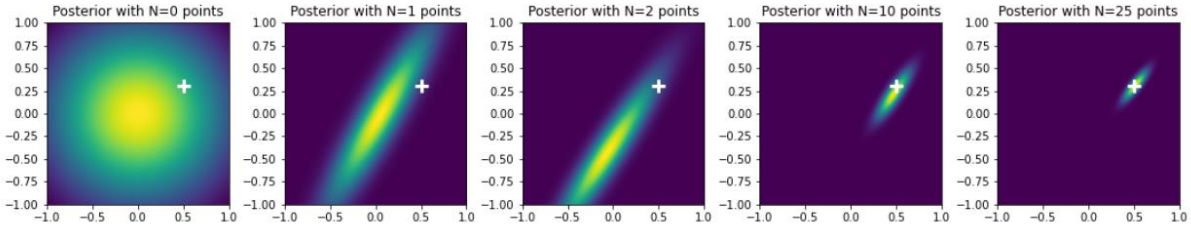
P(w |x,y) proportion p(w/x,w) p(w)

La forme du posterior est ici une gaussienne (car on a une vraisemblance et un prior gaussiens)

Cette forme peut être calculée analytiquement

$$p(w/X,Y) = \mathcal{N}(w|\mu,\Sigma)$$
$$\Sigma^{-1} = \alpha I + \beta \Phi^T \Phi$$
$$\mu = \beta \Sigma \Phi^T Y$$

Q1.2 : Looking at the visualization of the posterior above, what can you say?



La photo la plus à gauche est une gaussienne à priori, les 4 autres sont à posteriori. On remarque que dépendamment du nombre d'exemples qu'on a en entrée, plus on a d'exemples et plus la variance diminue et on tend vers les vrais paramètres.

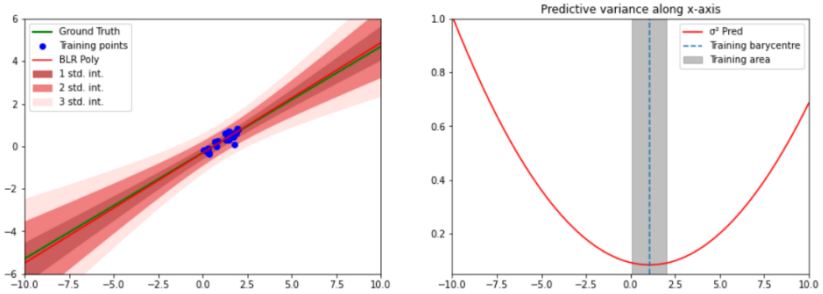
Q1.3 closed form of the predictive distribution in linear case

Pour calculer la distribution prédictive, on utilisera la distribution à posteriori et en marginalisera sur les poids w.

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

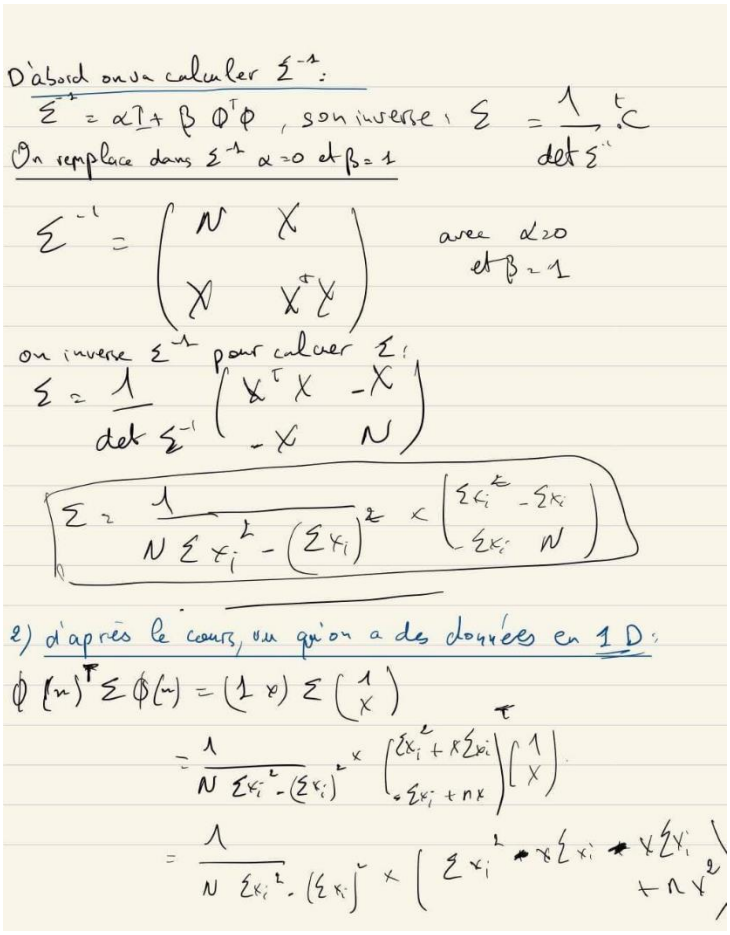
$$p(y^*|x^*,\mathcal{D},\alpha,\beta) = \mathcal{N}(y^*;\mu^T\Phi(x^*),\frac{1}{\beta} + \Phi(x^*)^T\Sigma\Phi(x^*))$$

Q1.4 Analyse these results. Describe the behavior of the predictive variance for points far from training distribution. Prove it analytically in the case where α=0 and β=1.

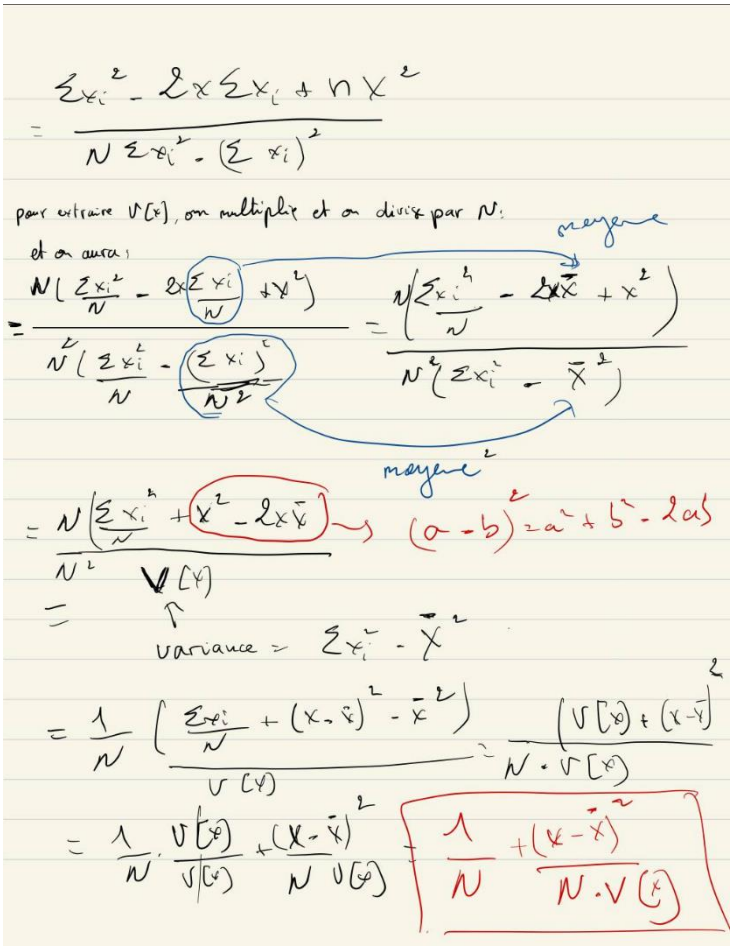


Plus on s'éloigne des points du dataset, plus la variance augmente (l'incertitude augmente) et plus on se rapproche des données du dataset et plus la variance diminue jusqu'à atteindre son minimum là où se trouve le centre des données $\bar{x}$ (le barycentre).

Quand on a Beta=1 et alpha = 0



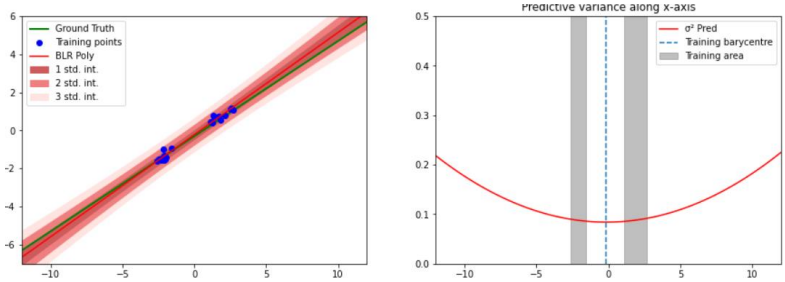
page 1



page 2

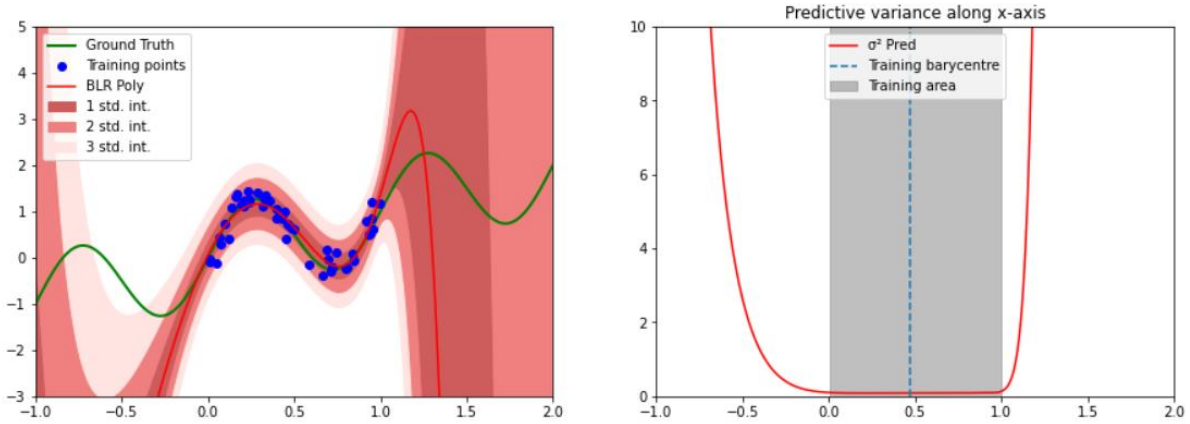
à la fin, on obtient variance= $\frac{1}{N} + \frac{(x-\bar{x})^2}{N.V[x]}$ ce qui nous affirme que plus la distance entre une donnée x et le centre de donnée $\bar{x}$ est minime plus la variance diminue, et inversement.

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



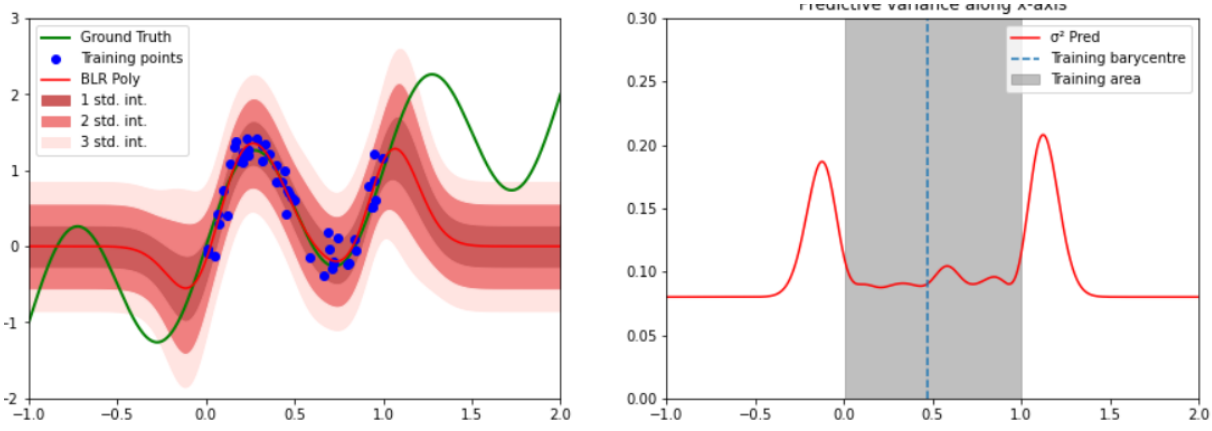
On remarque que même s'il n'y a pas de données dans le centre des données, la variance atteint quand même son minimum, alors qu'elle devrait être élevée vu qu'il n y a pas de données. Plus on s'éloigne du centre des données (barycentre) et plus la variance augmente mais très lentement. On peut conclure que notre modèle n'est pas adapté à ce type de données vu qu'il ne prend pas compte des positions des données.

[Q2.1] What can you say about the predictive variance?



Le minimum de la variance n'est plus le barycentre des données seulement (on voit que le minimum est entre 0 et 1 du x-axis) et plus on s'éloigne des données plus la variance augmente. On remarque aussi que notre modèle arrive bien à suivre nos données.

Q2.2 What can you say this time about the predictive variance? What can you conclude?



On remarque que plus on s'éloigne des données et plus la variance diminue et converge vers une valeur proche de 0.08 alors qu'on devrait normalement tendre vers une grande valeur voir l'infinie vu qu'on s'éloigne des données. Environ la même valeur de la variance est observée pour la zone où se trouve nos données. On remarque également que la fonction n'est pas convexe et qu'il y a un léger rebond dans sa valeur qu'on on est au bord de la zone où se trouve nos données.

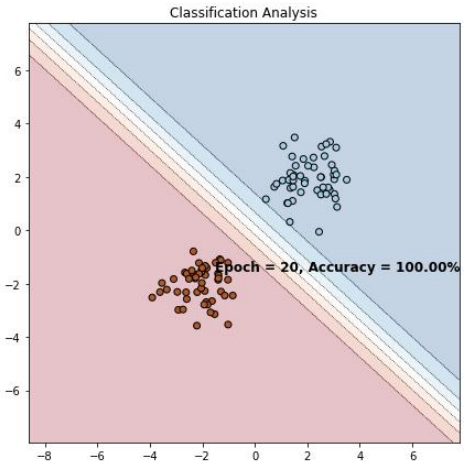
[Question 2.3] Explain why in regions far from training distribution, the predictive variance converges to this particular value when using localized basis functions such as Gaussians.

La variance converge vers cette valeur car on a une loi normale centrée et vu qu'on lui ajoute un écart type du bruit (ecart type = 0.2) et donc cette valeur de 0.08 correspond à 2x ecarttype²

TP 2: Approximate Inference in Classification

[Question 1.1] : Analyze the results provided by previous plot. Looking at p(y=1|x,wMAP), what can you say about points far from train distribution ?

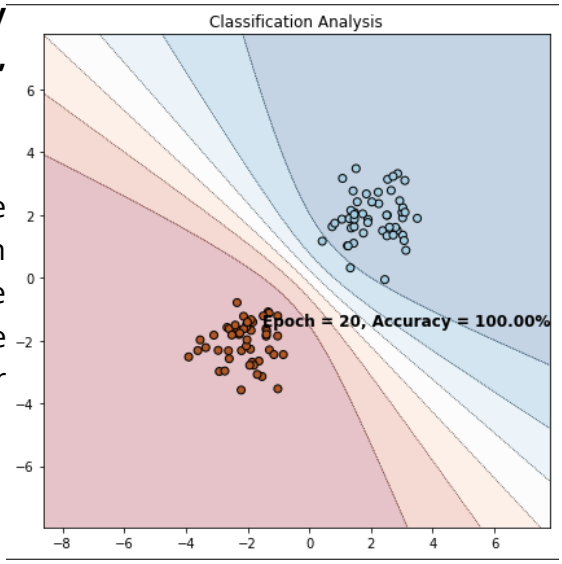
Quand on s'éloigne de la frontière de décision, le modèle reste confiant, on ne reflète donc pas l'incertitude épistémique ici. Le modèle donne la même valeur de certitude quel que soit la zone où on se trouve qu'on on est loin de la frontière de décision



En utilisant le MAP

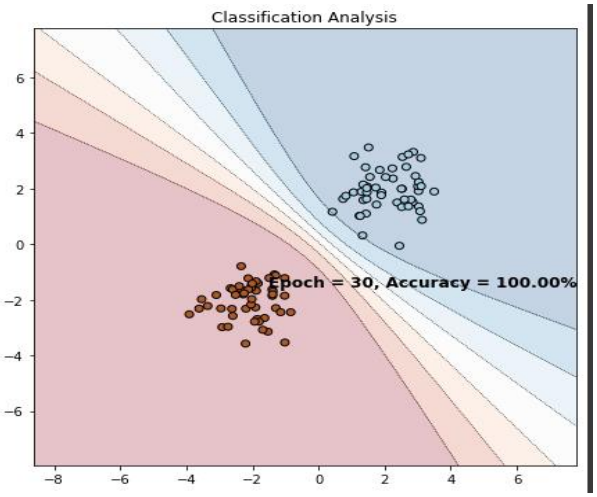
[Question 1.2]: Analyze the results provided by previous plot. Compared to previous MAP estimate, how does the predictive distribution behave?

En comparant à la figure précédente du MAP, la frontière de décision issue de la méthode Laplace Approximation n'est pas la même et on remarque que la certitude décroit quand on longe la frontière de décision à droite et à gauche, cependant, on a toujours la même valeur pour les points derrière les données d'entrainement



[Question 1.3] :

Analyze the results provided by previous plot(VI). Compared to previous MAP estimate, how does the predictive distribution behave ?



Commentaires sur LinearVariational :

Cette classe codée en Pytorch et qui hérite de nn.Module implémente une couche linéaire et ses paramètres suivent une loi variationnelle. La classe a différents attributs et méthodes :

Les attributs :

- **parent** : permet d'accumuler les KL-divergence des autres modules qui sont calculés dans le cas où on a plusieurs couches linéare variationnal
- **w_mu et b_mu** représentent respectivement les éspérences des lois des poids w, et le biais b respectivement.
- **w_rho et b_rho** représentent respectivement les variances des lois des poids w et le biais b respectivement.

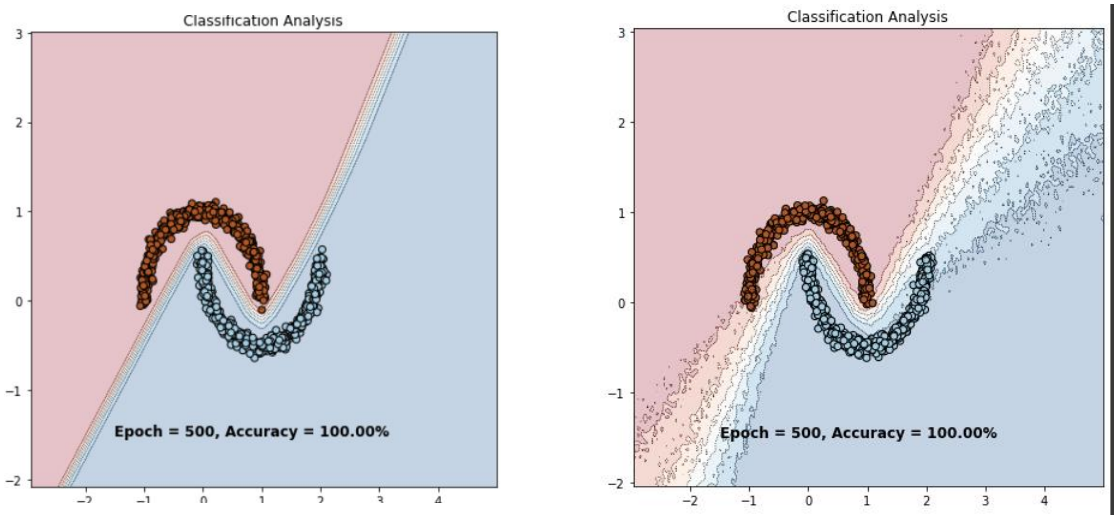
Les méthodes :

- **Samping** : prend en entrée mu et rho, au lieu d'échantillonner de la variational distribution, en utilisant le Reparametrization Trick. On va échantillonner un vecteur à partir de la distribution gaussienne qui a comme paramètre mu et ecart type = log(1+exp(rho)) , le log sert à avoir une variance positive.
- **Forward** : calcule des sorties (Tensors) à partir de Tensors fournis en entrée, dans cette méthode on échantillonne w et b qui sont des vecteurs de poids et de pied puis on fait le calcule d'une couche lineaire avec comme paramètre w et b. A la fin, on calcule la KL divergence
- **KL_divergence** : à partir des entrées mu_theta et rho_theta et z fournit en entrée et qui nous donnent une distribution normale définit par ces paramètres, on calcule la KL divergence entre cette distribution et un prior N(0,1) (loi normale centrée)

Analyse des résultats:

On remarque que nos données sont toujours bien séparés, la variance a augmenté autour de la frontière de décision donc l'incertitude est présente et augmente augmente un peu dans la zone des données d'entrainements , cependant pour la zone qui se trouve derrière nos données , la certitude reste la même (forte) (ici on parle de certitude et pas incertitude c'est pour cela qu'on a dit qu'elle est forte)

[Question 2.1] : Again, analyze the results showed on plot. What is the benefit of MC Dropout variational inference over BayesianLogistic Regression with variational inference ?

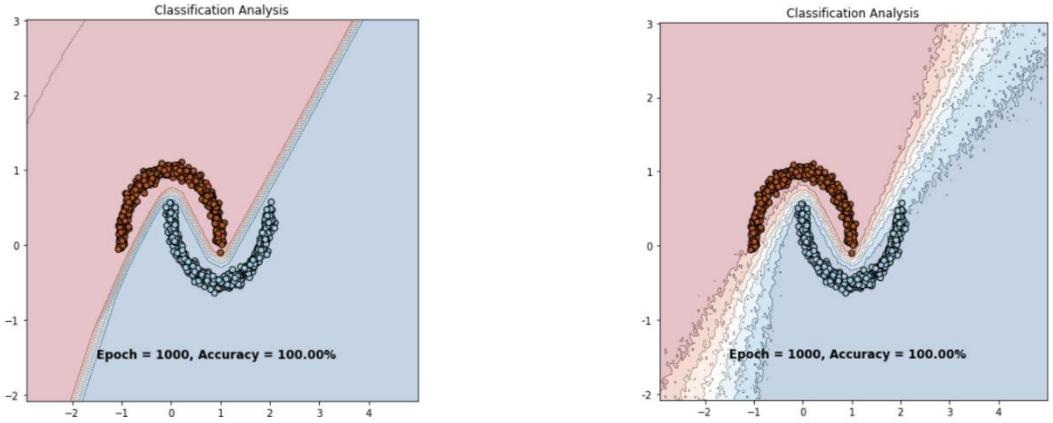


Entrainement 500 epochs

Après avoir entrainé le modèle pour un nombre d'époch égale à 500, on remarque que le modèle sépare parfaitement nos deux classes, la variance augmente de plus en plus quand on s'éloigne de nos données. Cette variance peut être utilisée pour la représentation de notre incertitude épistémique.

Si on entraine notre modèle pour plus d'epochs (par exemple 1000-1500 epochs), le modèle aura toujours une accuracy de 100% cependant la variance ne sera pas comme dans la figure figure, elle sera un peu plus petite comme on la voit dans la figure suivante.

Après 1000 epochs



Entrainements pour 1000 epochs

Les benefices :

- Possibilité de changer le dropout rate, aussi au niveau du code c'est plus facile à implémenter
- Obtenir des résultats sans augmenter la complexité

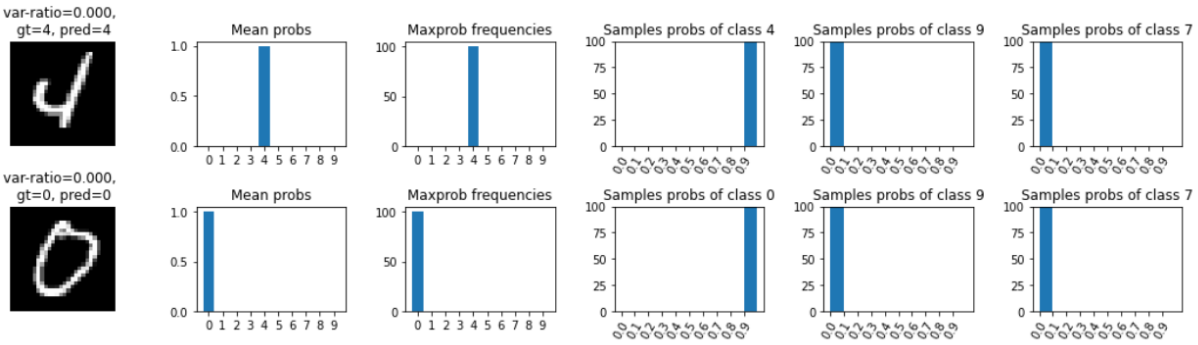
TP 3: Bayesian Deep Learning and Robustness

Expliquer le principe de la failure prediction

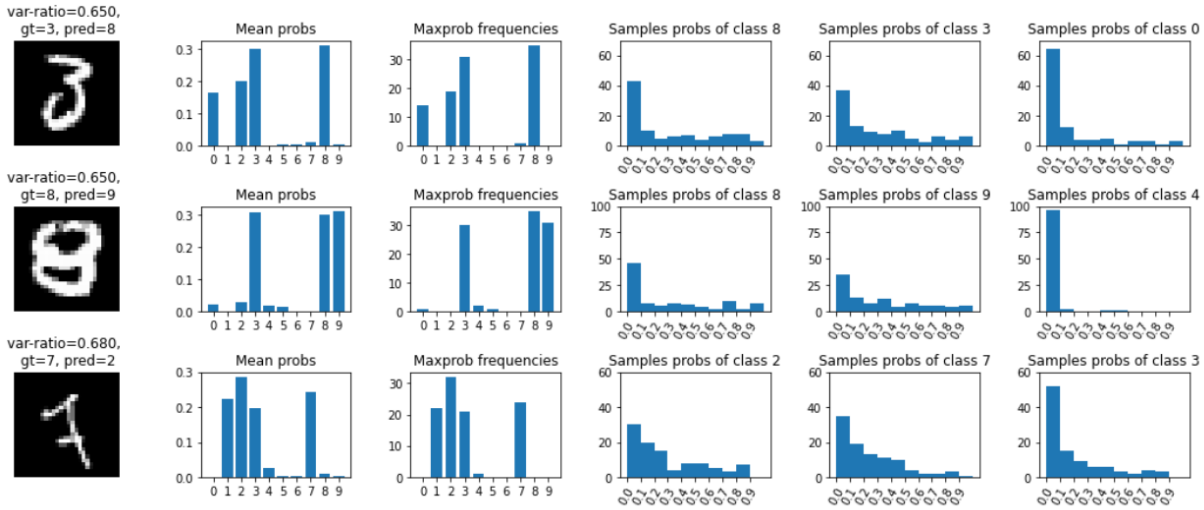
On utilise l'incertitude pour accepter ou rejeter des prédictions. A partir d'un seuil d'incertitude le modèle va décider de faire la classification ou non. Si on dépasse le seuil alors le modèle détectera que son incertitude est trop grande et donc il ne sait pas quoi prédire. Ceci va nous permettre de rejeter la prédiction.

L'objectif de cette détection est que le modèle se sert de cette mesure d'incertitude pour savoir quand il ne sait pas.

[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 ?



Random images



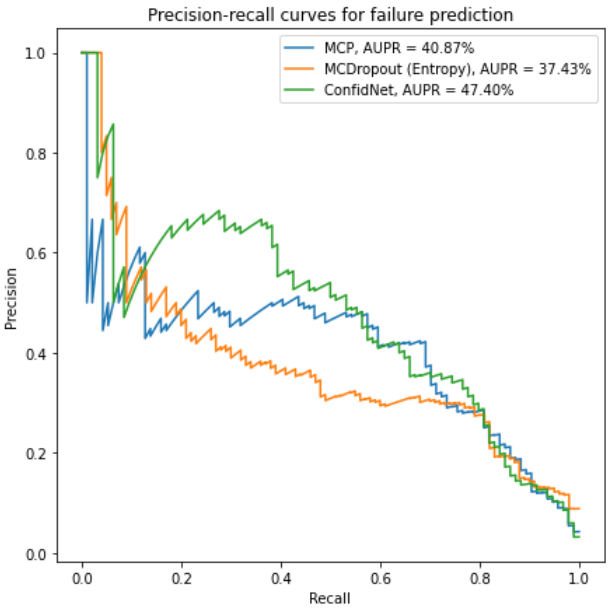
Top uncertain images

La distribution de probabilité pour les images qui sont incertaines possèdent plusieurs pics dans leur distribution tandis qu'il n'y a qu'un seul pic pour les images aléatoires.

Pour les images aléatoires dans la figure 1, le modèle n'a donc pas de doute sur la classe prédite vu que la probabilité de la classe prédite est de 1 pour l'unique pic qui est la classe prédite et est de 0 pour toutes les autres classes. Alors que pour les images qui sont les plus incertaines le modèle n'est pas sûr de lui et différentes classes ont des probabilités

(Dans la figure : mean probs représente le vecteur proba moyen et max probs représente combien de fois une classe a été prédite)

[Question 2.1] : Compare the precision-recall curves of each method along with their AUPR values.

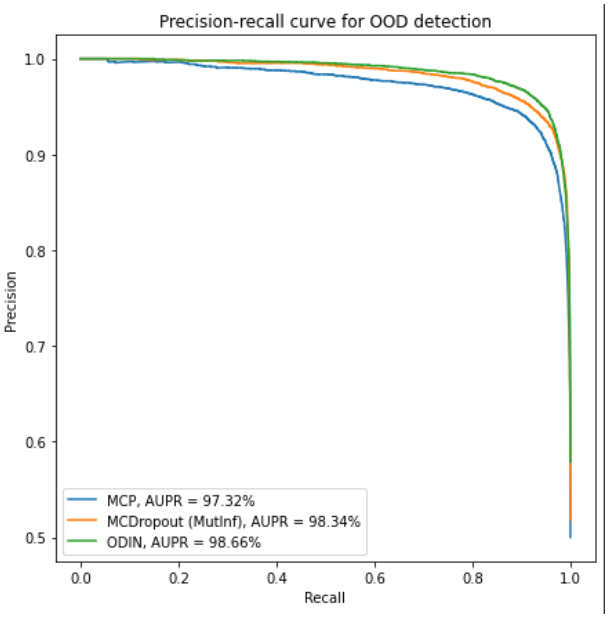


On peut observer que pour les trois méthodes utilisées, le réseau n'arrive pas à prédire ses erreurs (40% pour le MCP, 37% pour le MCDropout entropy et 47% pour confidnet). Et donc pour détecter 20% d'erreurs, la précision des différents réseau est entre 50-60%. L'idéal ici c'est d'avoir des courbes qui tendent plus vers la partie haute droite du graphe

Why did we use AUPR metric instead of standard AUROC ?

Les erreurs de classifications sont utilisées comme la détection positive de la classe , et ce qui nous intéresse ici. Donc AUPR par rapport à AUROC est plus sensible aux améliorations pour la class positives qui est notre erreur de classification. AUPR regarde la positive predictive value (PPV) et le true positif rate (TPR), alors que AUROC regarde le vrai positif et le faux positif rate donc AUROC est utilisé plutôt quand on se souci équitablement de la classe positive et la classe négative, et vu qu'on se souci ici des erreurs de classifications qui sont la classe positive alors on prend AUPR.

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



Pour cette tâche, les différentes méthodes présentent de très bons résultats, les courbes s'allongent presque parfaitement vers le coté haut-droit du graphe, pour un recall de 80%, on a une précision de plus de 97%

La méthode qui performe le mieux est ODIN, car c'est elle qui a la courbe la plus au-dessus et c'est la plus proche d'une courbe parfaite entre les 3 méthodes