



Télécom Paris

TP1 SI221

Étudiant

Maxime Thauvin

Encadrant

Aslan Tchamkerten

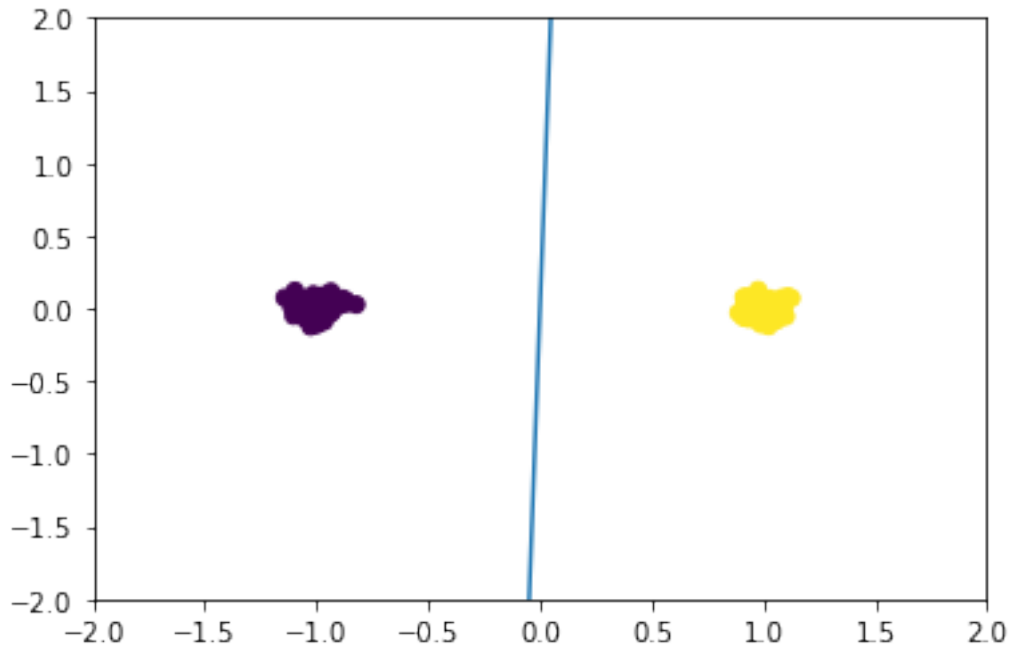
10 mars 2020

Exercice 1

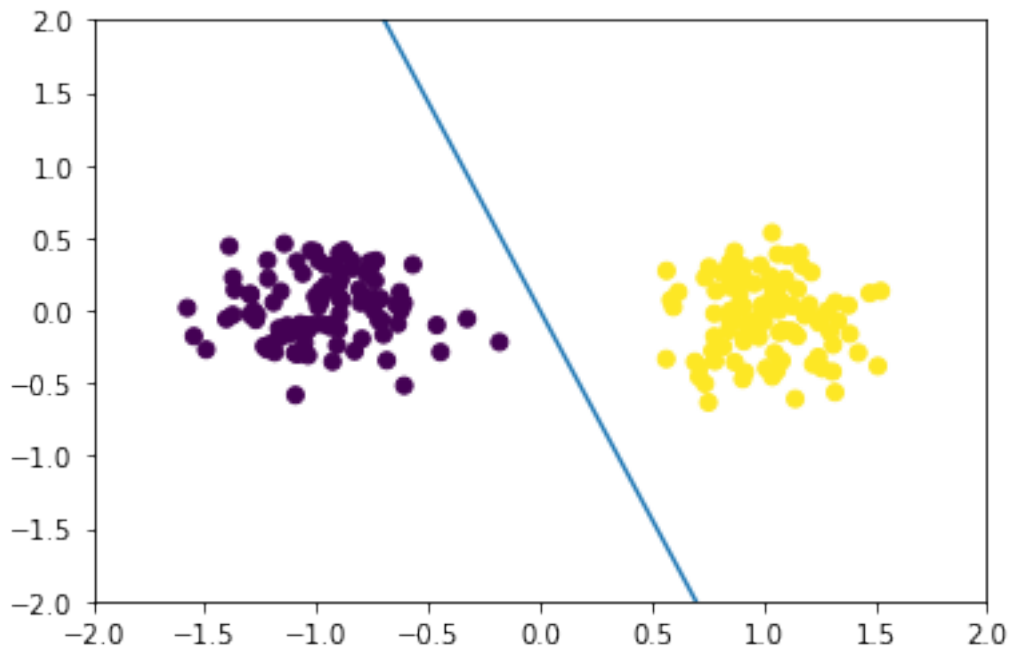
Question 1

En regardant les données générées et la droite séparatrice que l'on trace avec les poids calculés avec le perceptron, on remarque assez facilement que quand σ augmente, les deux sets de données sont de moins en moins séparables. En effet les points sont de plus en plus étalés et se mélangent.

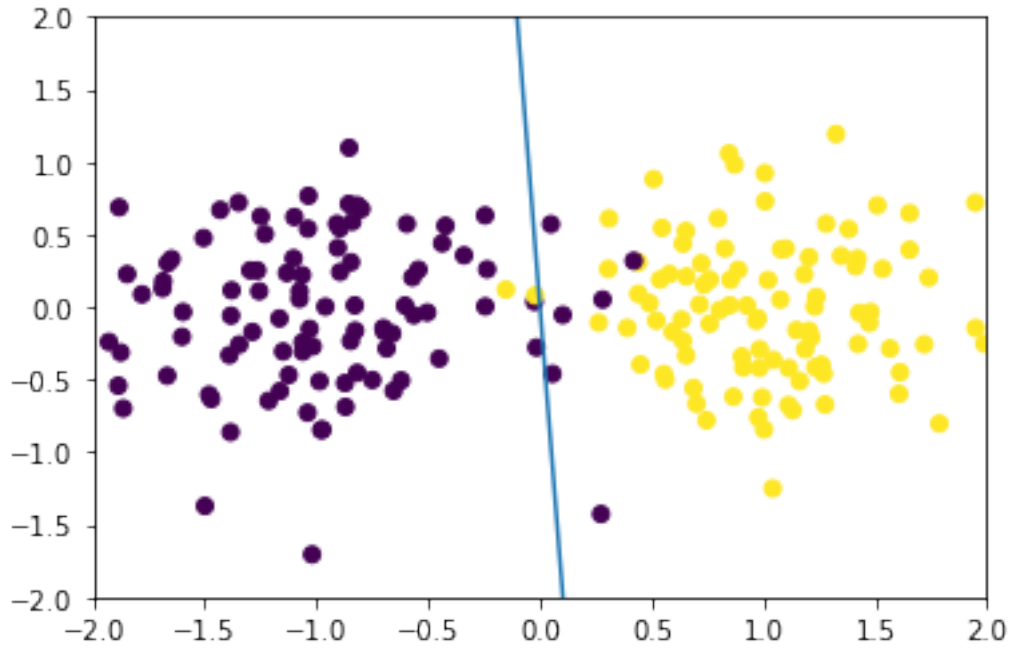
$\sigma = 0.05$



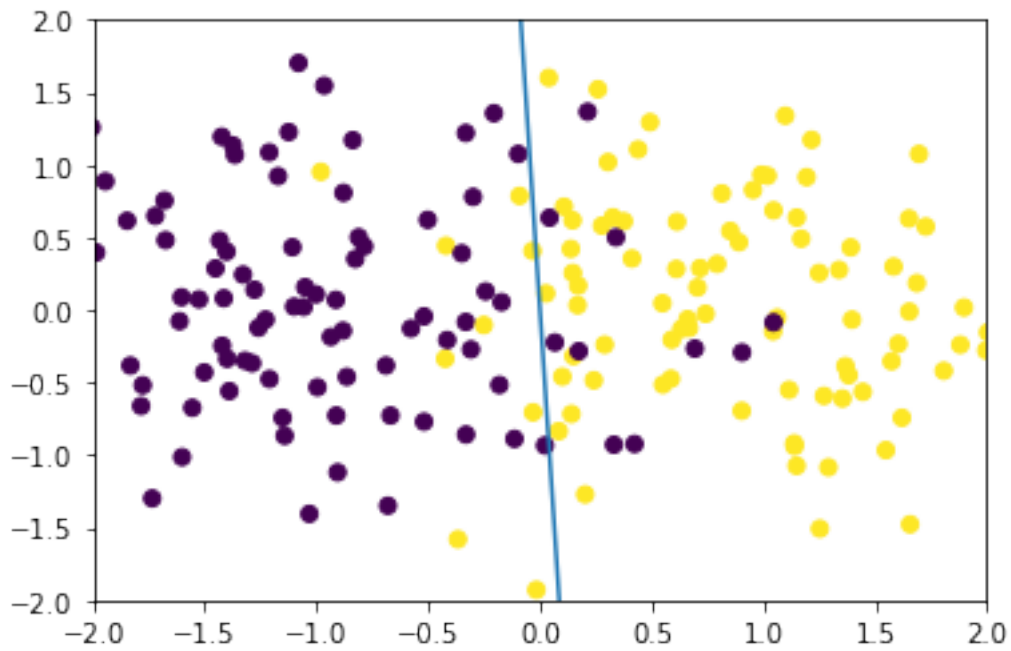
$\sigma = 0.25$



$\sigma = 0.50$

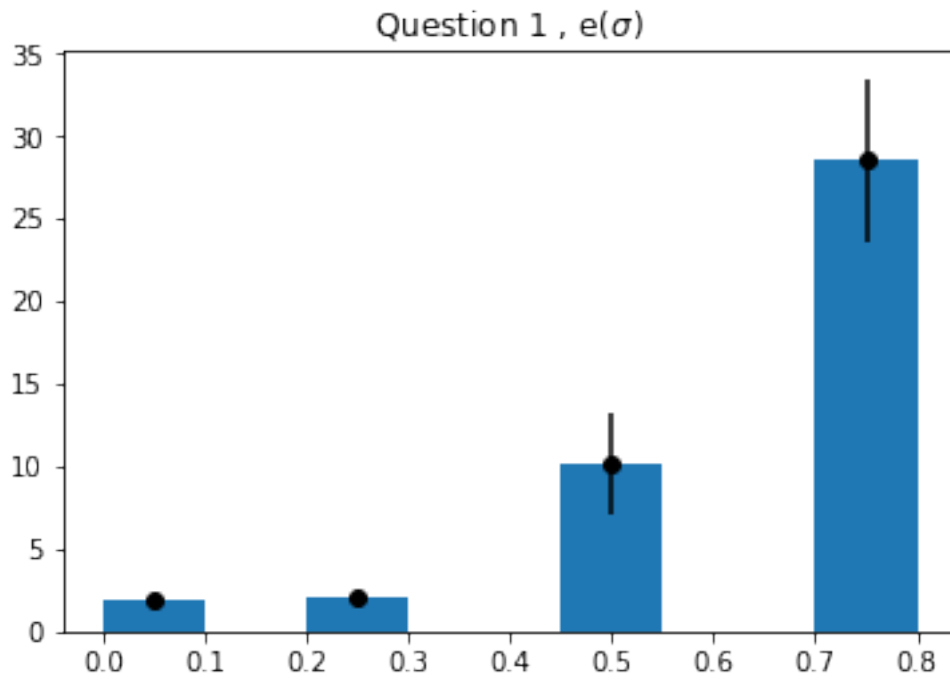


$\sigma = 0.75$



Résultats obtenus

En calculant les erreurs et la déviation pour les différentes valeurs de $\sigma = \{0.05, 0.25, 0.50, 0.75\}$ on obtient le error bar suivant :

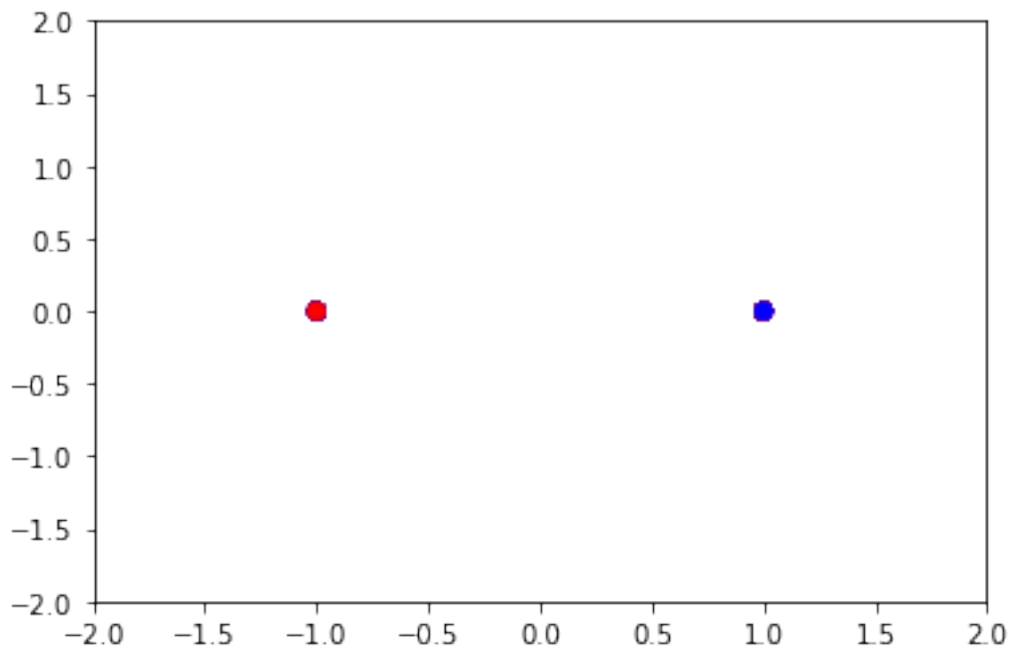


On voit bien sur ce graphique que pour des valeurs de σ inférieure à 0.50 l'erreur est quasi nulle. Mais pour des valeurs supérieures à 0.50 elle augmente rapidement, du fait que les deux ensembles sont de moins en moins séparables. La déviation augmente également car il y a plus de différences entre deux dataset puisque les points sont plus en plus mélangés, et donc la frontière entre les deux est de plus en plus dur à calculer pour le perceptron.

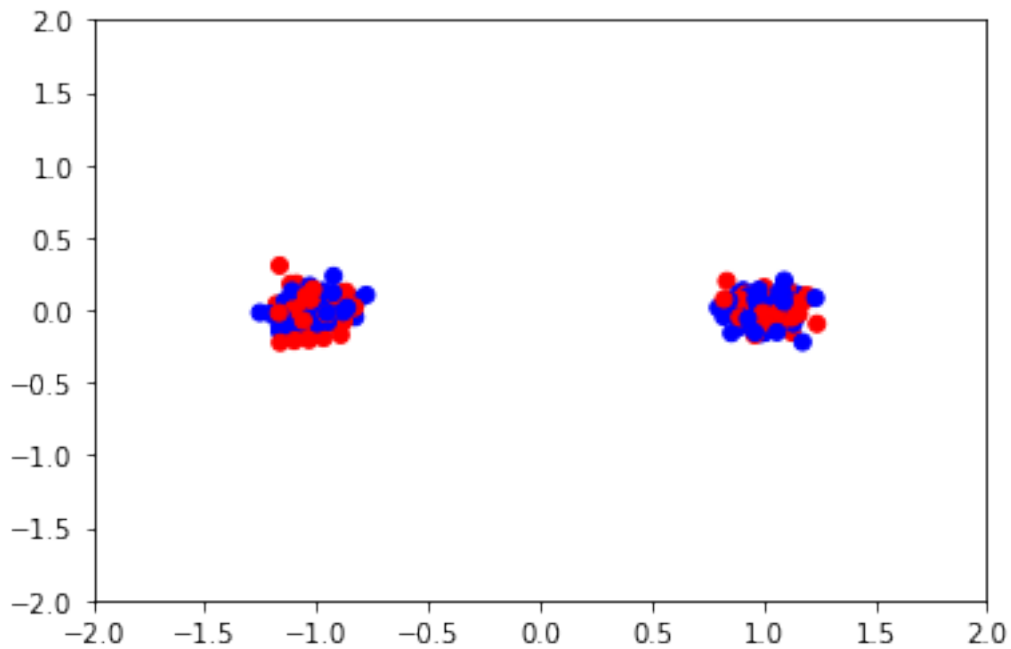
Question 2

Dans cette question, σ est fixé à 0.15 et des points changent de label avec une probabilité p . Plus p est grand, plus il y a de points qui changent de label et se retrouvent avec le mauvais label.

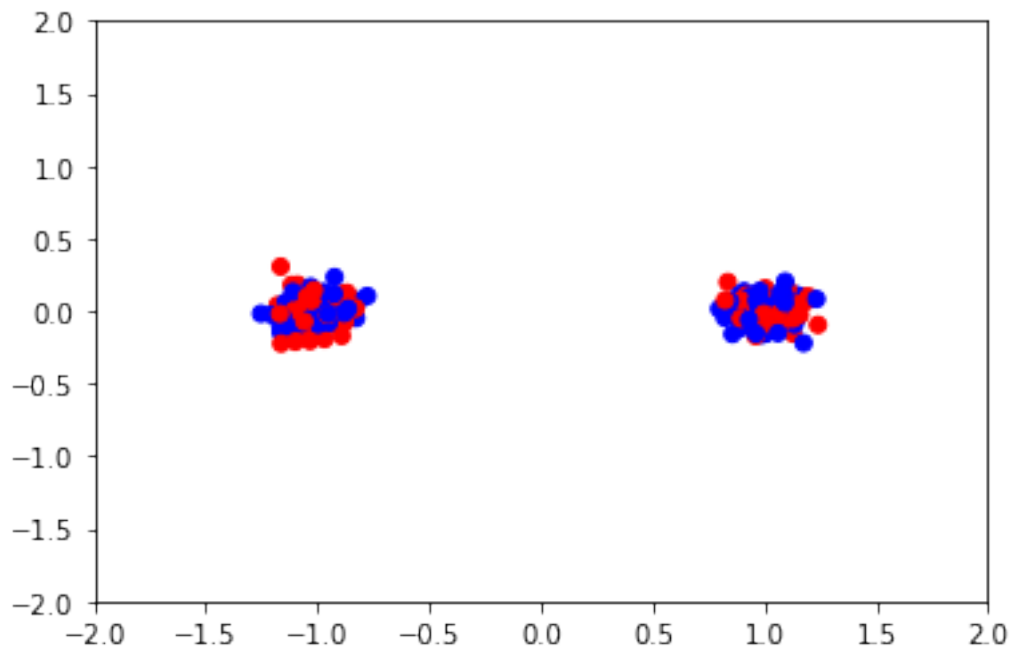
$p = 0$



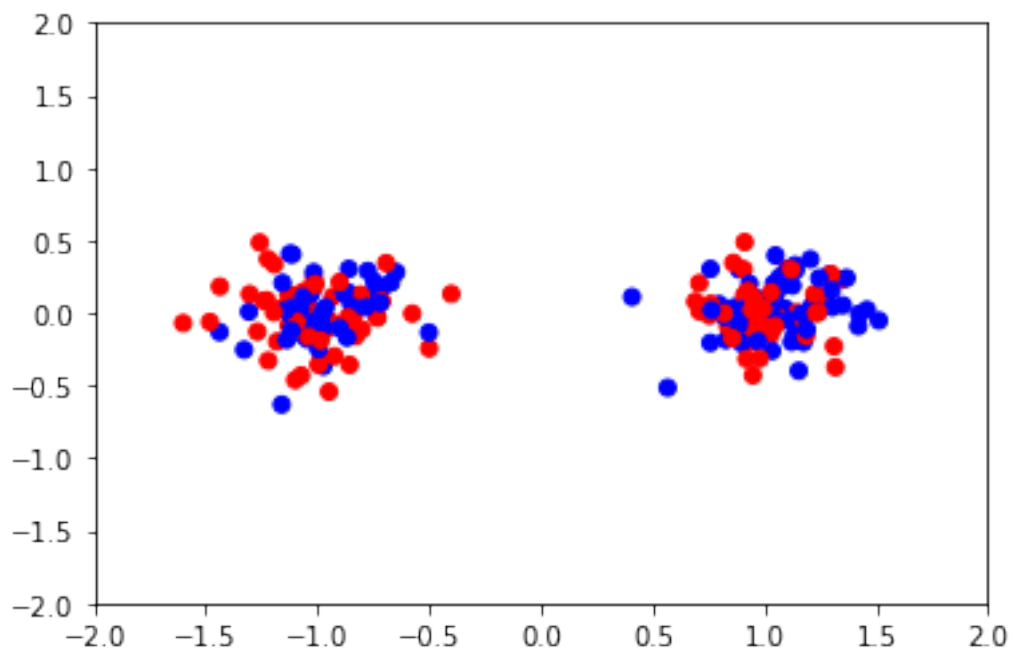
$p = 0.05$



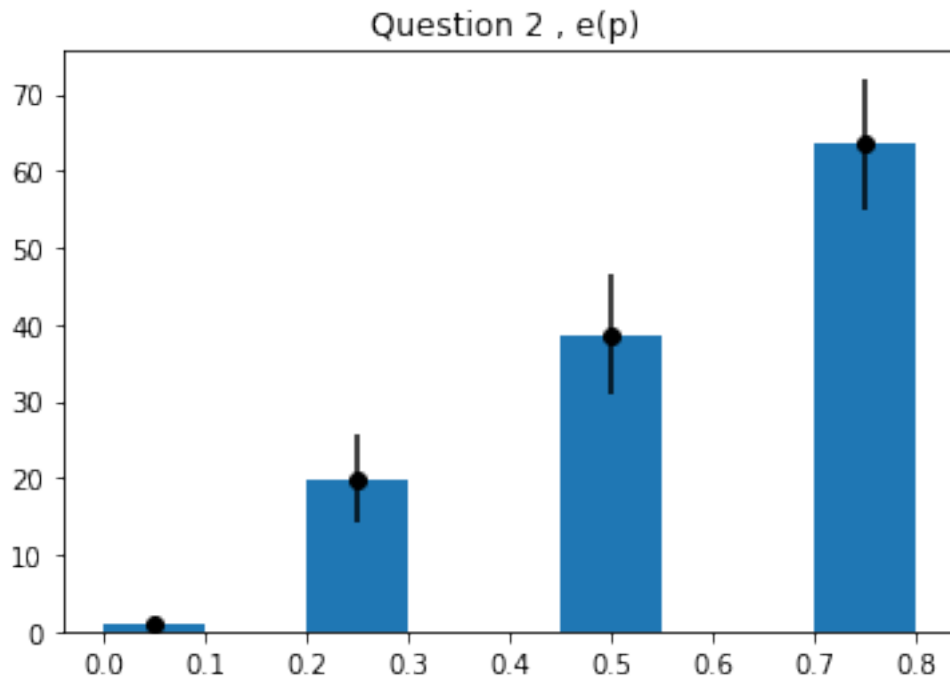
$p = 0.10$



$p = 0.20$



Résultats obtenus

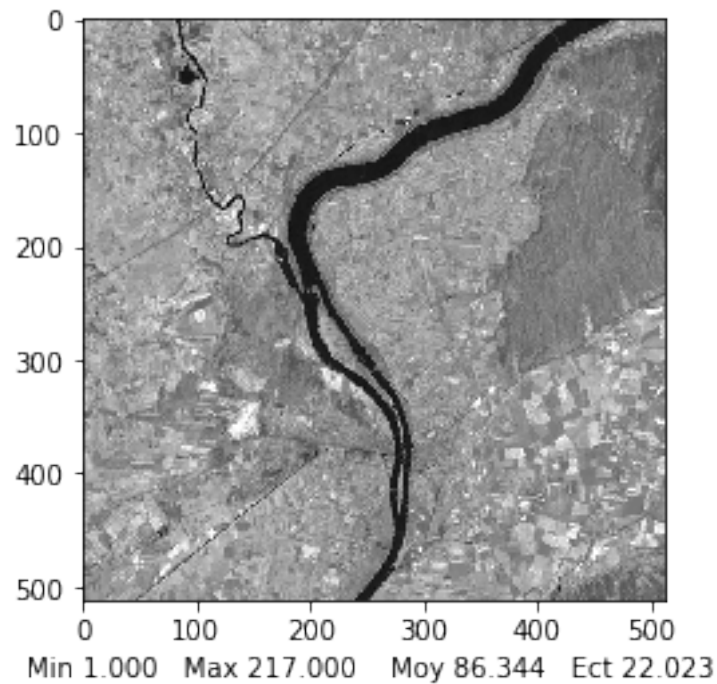


Quand p augmente, le perceptron a de plus en plus de mal à trouver une solution car les deux ensembles sont de moins séparables. Le nombre d'erreurs et la déviation semblent proportionnel par rapport à la valeur de p , ce qui est normal car ces erreurs interviennent pour les points qui ont changés de label.

En changeant le label de certains points, on a simulé du "bruit" dans nos dataset, et on peut alors en conclure que le perceptron est sensible au bruit et aura du mal à fonctionner en présence de bruit.

Exercice 2

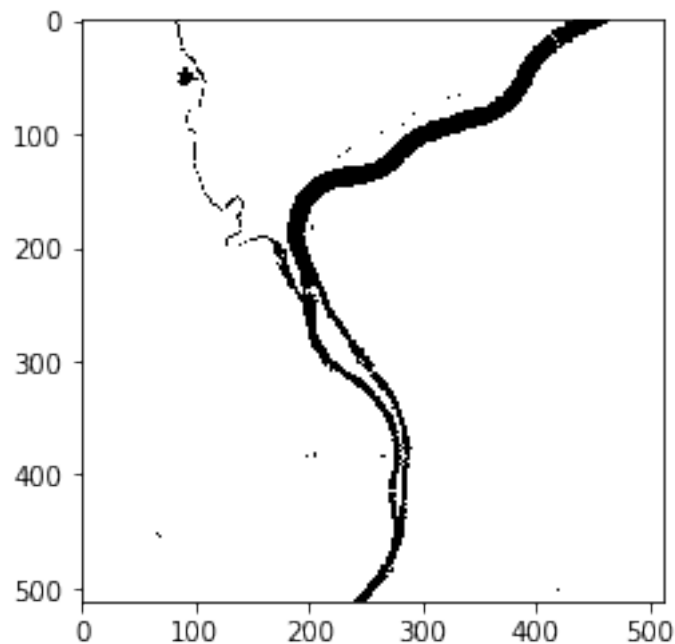
Voici l'image qui va servir de données dans cet exercice :



Partie 1 - label 1 pour les pixels avec une valeur inférieure à 30, label 2 sinon

Dans cette partie les labels 1 correspondent au pixel qui ont une valeur inférieure à 30, et qui représentent des points d'eau sur l'image satellite précédente.

Voici la matrice de label :

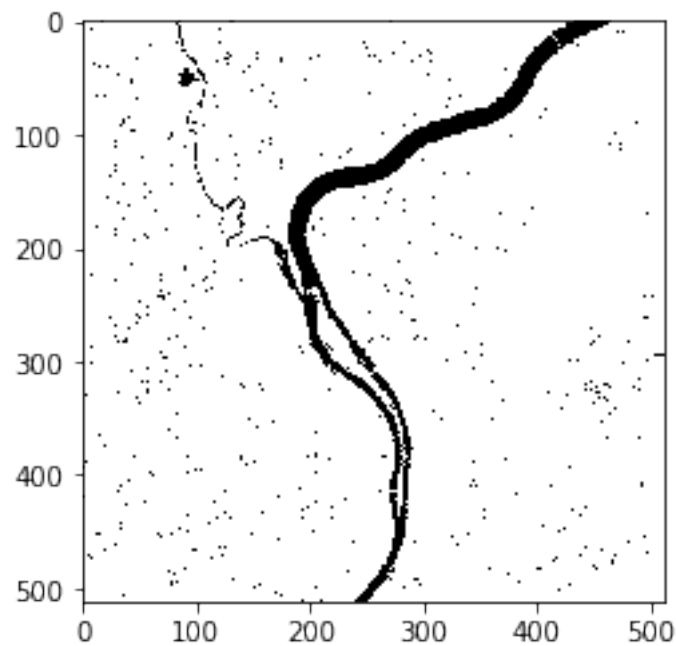


Les deux ensembles sont ici séparables (label 1 pour $[0, 30]$ et label 2 pour $] 30, \infty[$). Le perceptron trouve converge alors assez rapidement en une ou deux epochs.

Partie 2 - ajout des pixels avec une valeur égale à 110 aux label 1

Sans cette partie on rajoute les pixels de valeur 110 au label 1, cela représente du bruit, comme si le capteur photo ne fonctionnait pas pour une luminosité de 110.

On obtient alors la matrice de label suivante, sur laquelle on voit bien une "image" bruitée :

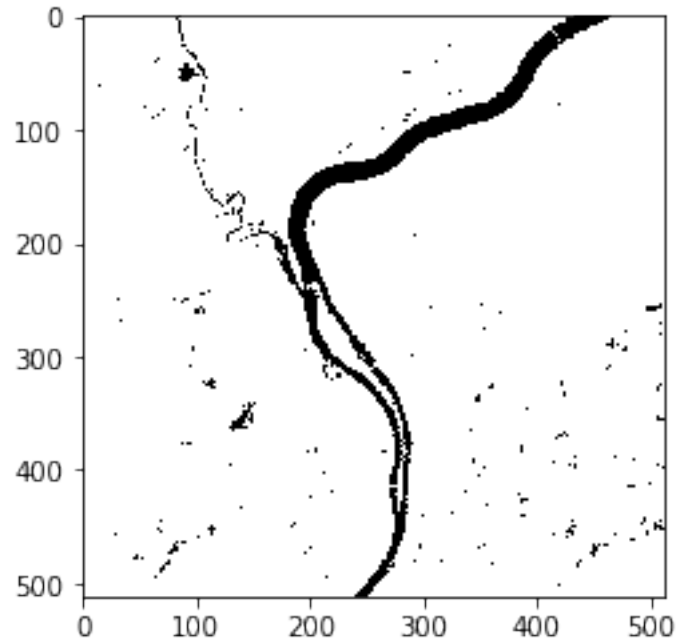


2391 pixels sont concernés par ce changement de label. Au bout de 50 epochs le perceptron n'a toujours pas convergé. Il y a alors 5391 erreurs à la fin des 50 epochs.

Partie 3 - ajout des pixels avec une valeur supérieure à 140 aux label 1

Sans cette partie on rajoute les pixels de valeur supérieure à 140 au label 1, cela représente du bruit, comme si le capteur photo ne fonctionnait pas pour une luminosité supérieure à 140.

On obtient alors la matrice de label suivante, sur laquelle on voit bien une "image" bruitée :



1593 pixels sont concernés par ce changement de label. Au bout de 50 epochs le perceptron n'a toujours pas convergé. Il y a alors 2476 erreurs à la fin des 50 epochs.

Conclusion

On peut conclure de ce TP que le perceptron est un algorithme très efficace quand les deux ensembles sont facilement séparables, mais qu'il est très sensible au bruit, et devient alors peu efficace