



## Rapport de laboratoire

Présenté à  
Alexandre Piché

Par  
Justine Pepin 1789244  
Gr. 1 de laboratoire

Dans le cadre du cours de I.A. : tech. probabilistes et d'apprentissage

INF8225

4 février 2018

## Observations

Dans les pages suivantes, à raison d'une expérience par page, seront présentés les résultats obtenus avec certaines combinaisons de taux d'apprentissage et de tailles de mini-batch lors de l'exécution d'un code répliquant une descente stochastique de gradients par mini-batches avec la fonction de régression logistique.

En règle générale, un taux d'apprentissage plus élevé offre en moyenne une meilleure précision sur l'ensemble de test. Par contre, les courbes sont sujettes à beaucoup plus de bruit car suite à chaque ajustement des gradients sur une mini-batch les poids sont augmentés ou diminués avec un plus grand impact.

Autre fait globalement observé : lorsque les mini-batches sont très petites (comme les tailles de un et vingt exemples), les pertes de l'ensemble de validation concordent beaucoup moins bien avec celles de l'ensemble d'entraînement, probablement car les poids sont mis à jour très souvent sur la base d'une poignée d'exemple et donc les poids sont spécifiquement bien adaptés à l'ensemble d'entraînement. Toutefois, la précision sur l'ensemble de test grimpe moins vite si les mini-batches sont constituées de plus d'exemples.

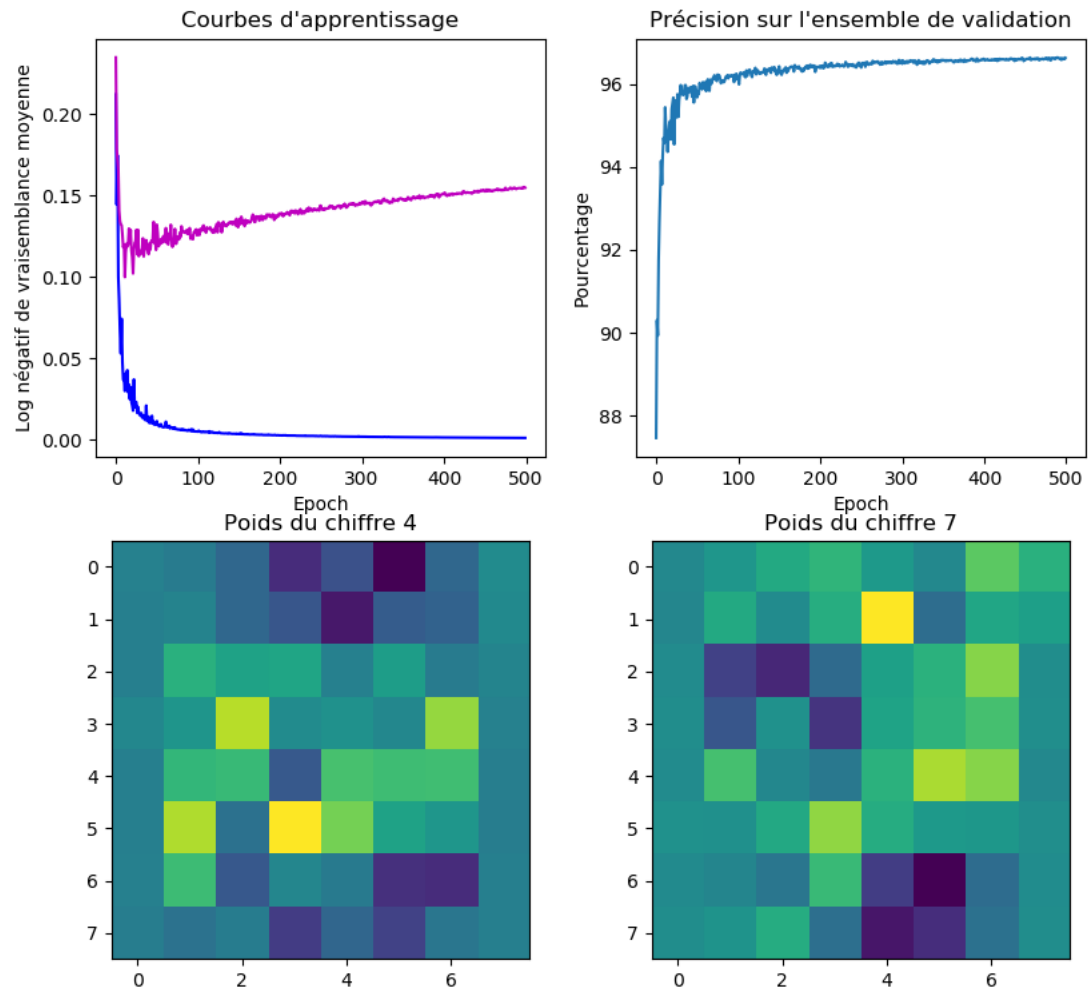
Finalement, on peut voir que les poids appris ont déjà l'allure générale des chiffres qu'ils représentent, et ce dans tous les cas. Autrement, puisque la précision sur l'ensemble de test de chacune des expériences est assez forte (aux alentours de 95% dans la plupart des cas), il serait intéressant de faire à nouveau les mêmes expériences mais en augmentant le nombre d'exemples artificiellement (en faisant par exemple de la distorsion d'images) et en ajoutant des fonctions de régularisation afin d'endurcir le système et d'obtenir une bonne précision sur des chiffres dans des conditions plus variées. Il serait intéressant aussi de proposer des images similaires de caractères alphabétiques afin d'introduire des situations où le système devrait répondre qu'aucune classe de chiffre ne correspond à ce qu'il perçoit.

### Interprétation des graphiques :

La première figure de chaque expérience comporte 4 graphes. Ceux-ci sont : le graphe des pertes observées par log négatif (où la courbe en bleu présente les pertes sur l'ensemble de d'entraînement et la courbe en magenta présente les pertes sur l'ensemble de validation), la précision (accuracy) sur l'ensemble de validation et les poids appris pour les chiffres 4 et 7.

La seconde figure est une capture d'écran de la console suite à l'exécution du programme. Elle présente la précision obtenue sur l'ensemble de test ainsi que les paramètres de l'expérience.

## Expérience 1 : taux d'apprentissage = 0.001, 1 élément/mini-batch

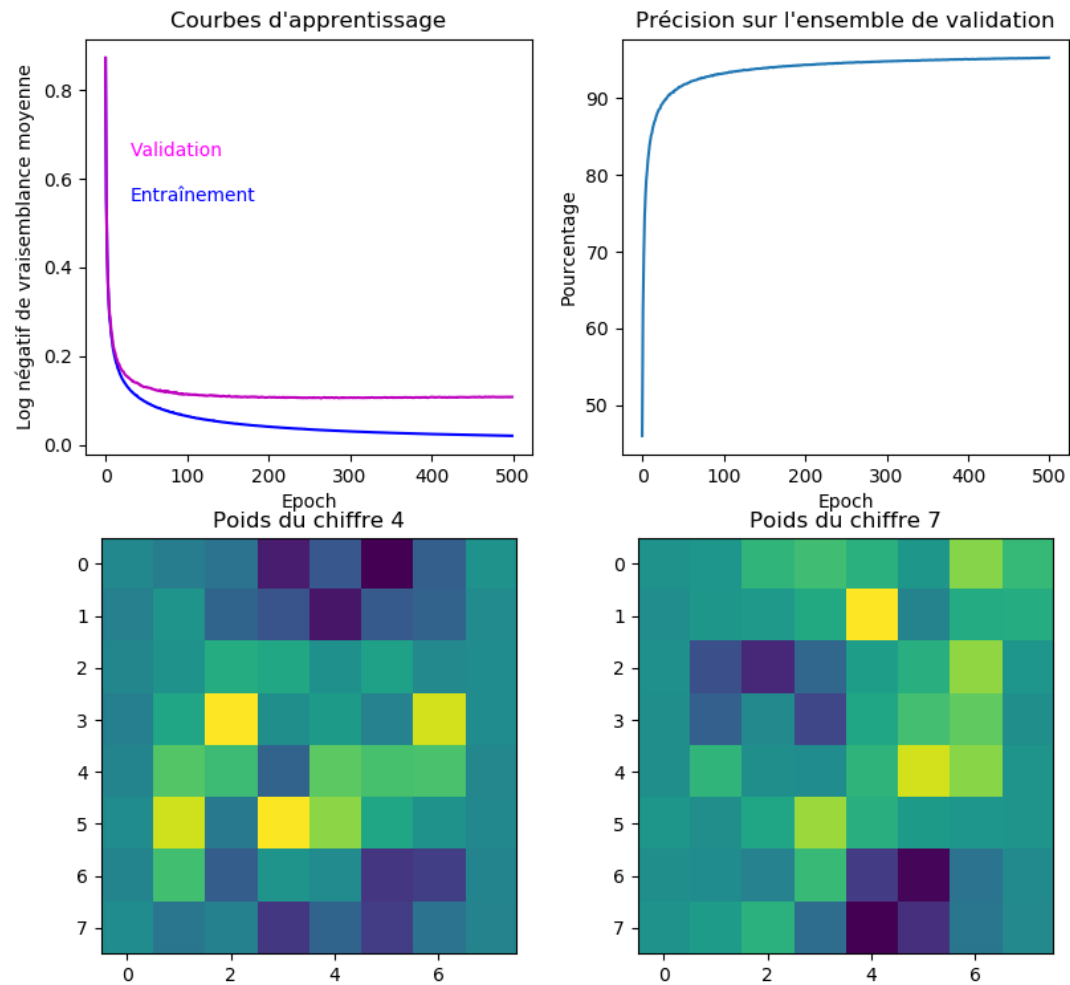


La précision obtenue = 95.9488996241

Pour un lr = 0.001

Avec une taille de mini-batch = 1

## Expérience 2 : taux d'apprentissage = 0.001, 20 éléments/mini-batch

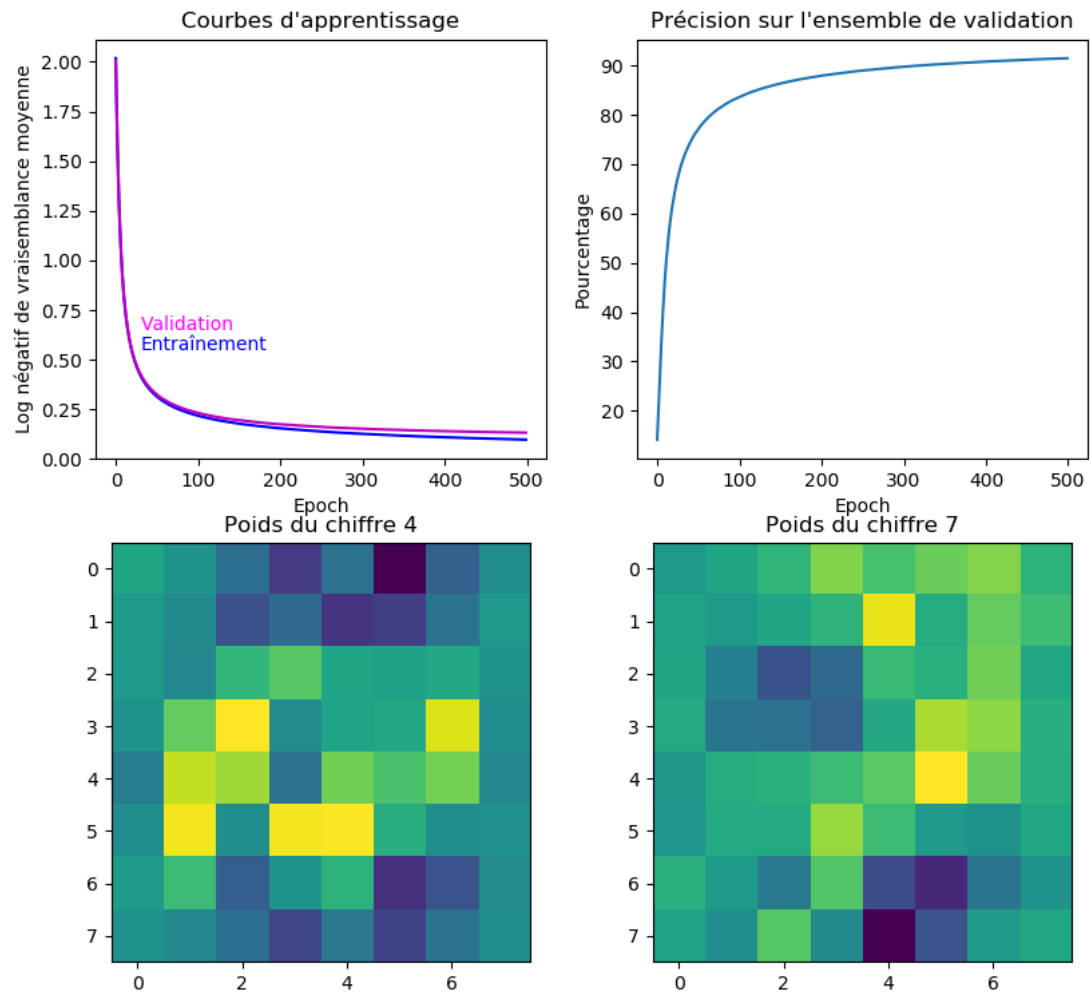


La précision obtenue = 95.1093056468

Pour un lr = 0.001

Avec une taille de mini-batch = 20

### Expérience 3 : taux d'apprentissage = 0.001, 200 éléments/mini-batch

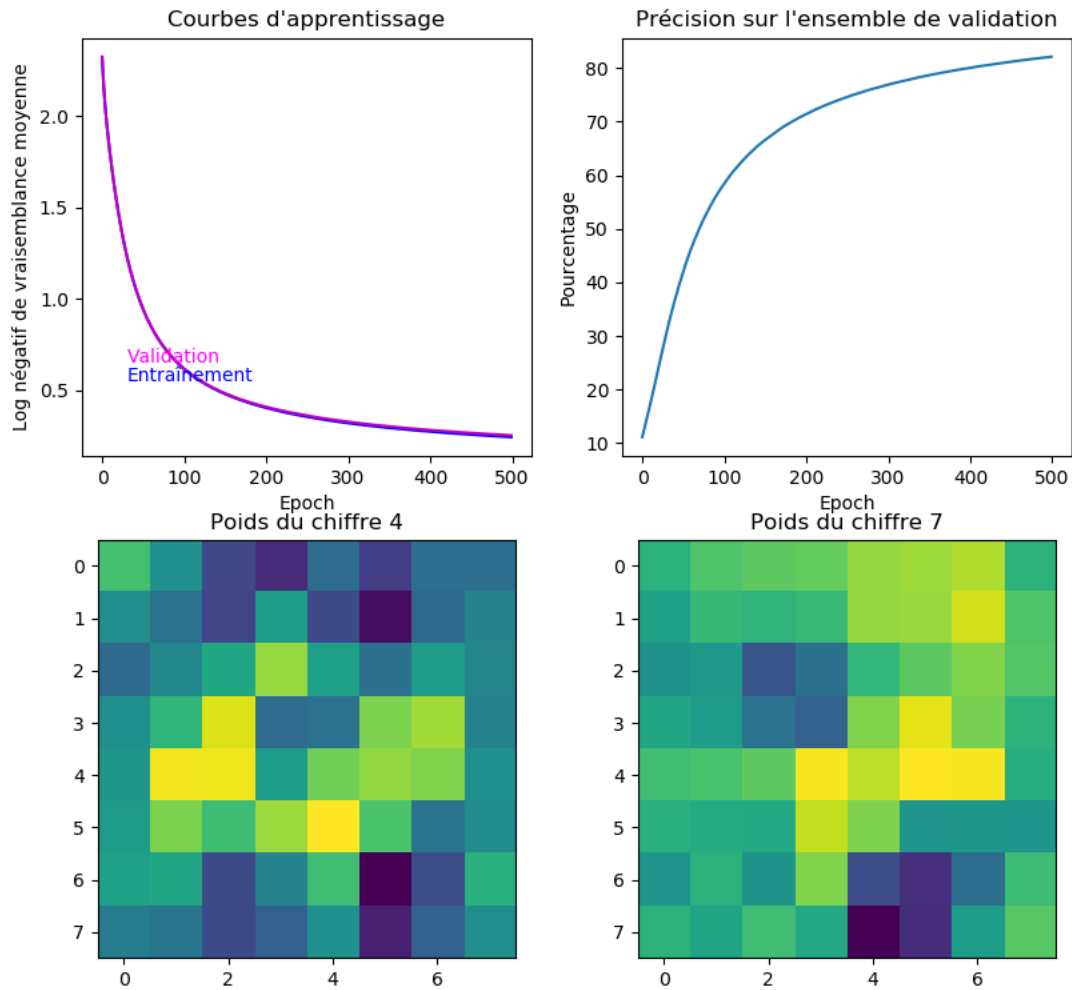


La précision obtenue = 91.6076835338

Pour un lr = 0.001

Avec une taille de mini-batch = 200

#### Expérience 4 : taux d'apprentissage = 0.001, 1000 éléments/mini-batch

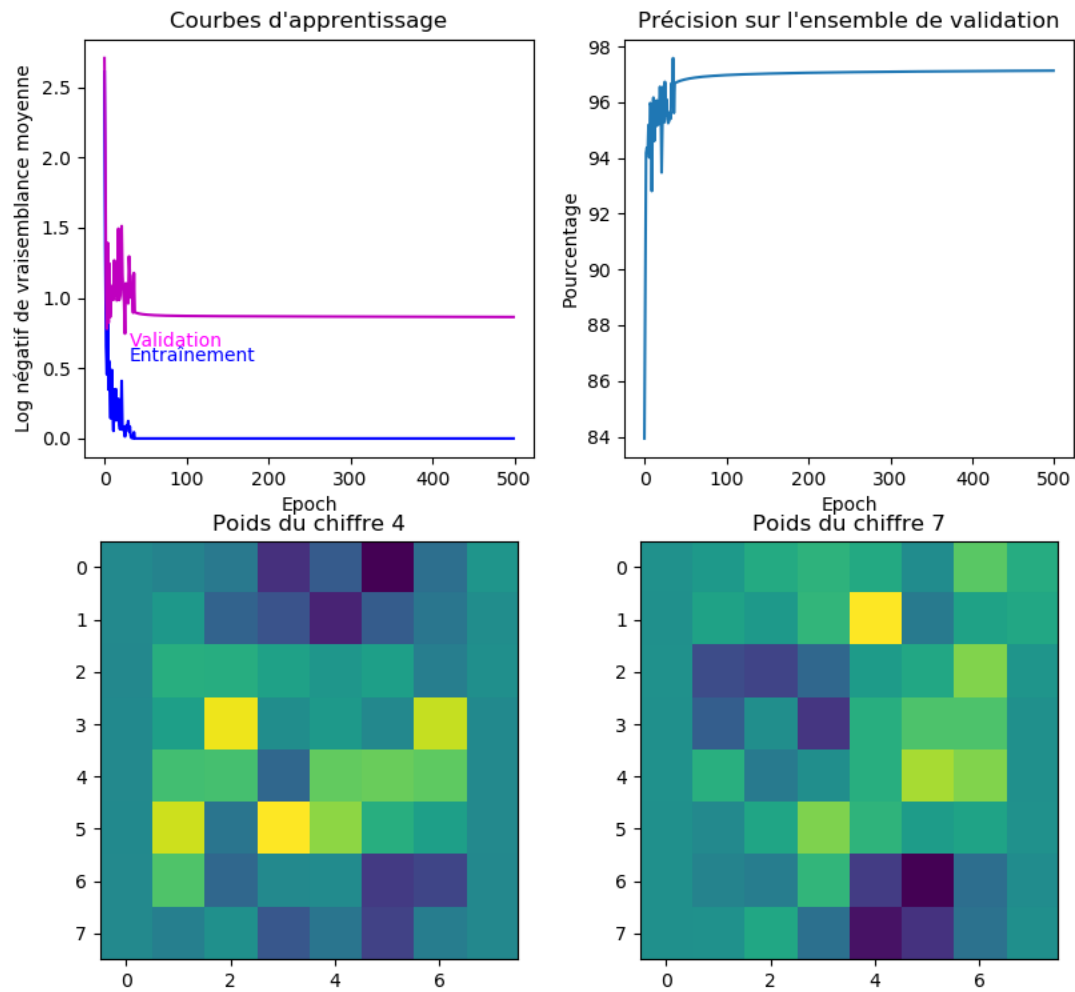


La précision obtenue = 82.0816971558

Pour un lr = 0.001

Avec une taille de mini-batch = 1000

## Expérience 5 : taux d'apprentissage = 0.01, 1 élément/mini-batch

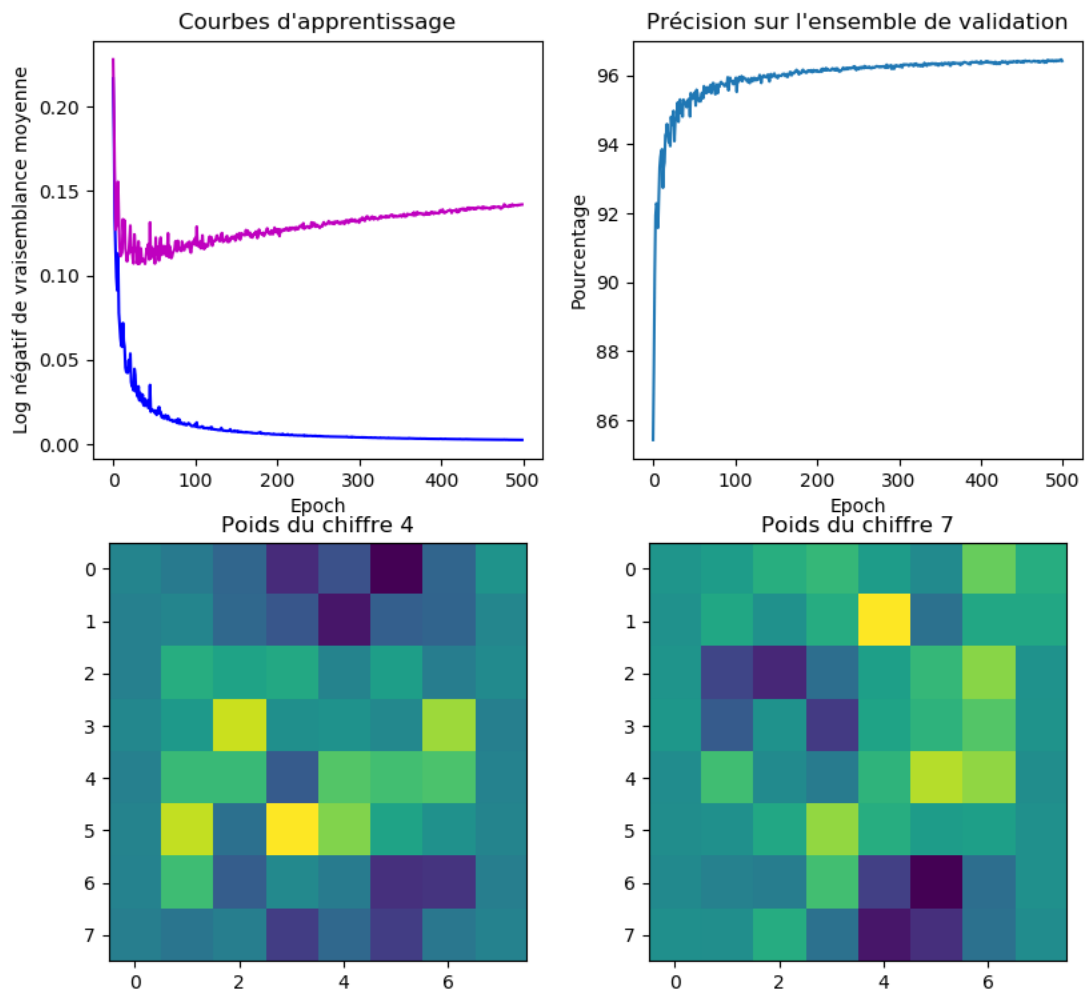


La précision obtenue = 95.9375503741

Pour un lr = 0.01

Avec une taille de mini-batch = 1

## Expérience 6 : taux d'apprentissage = 0.01, 20 éléments/mini-batch



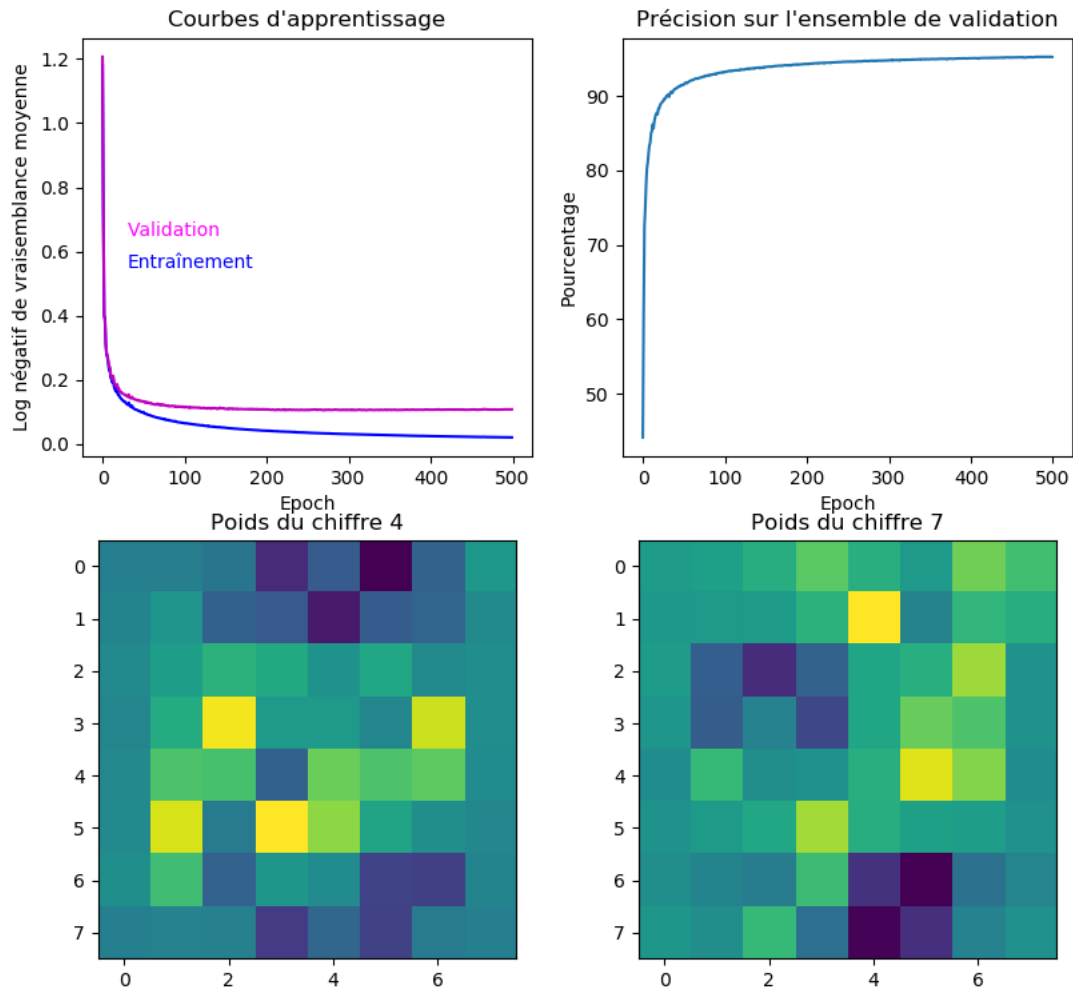
La précision obtenue = 95.8451663058

Pour un lr = 0.01

Avec une taille de mini-batch = 20



## Expérience 7 : taux d'apprentissage = 0.01, 200 éléments/mini-batch

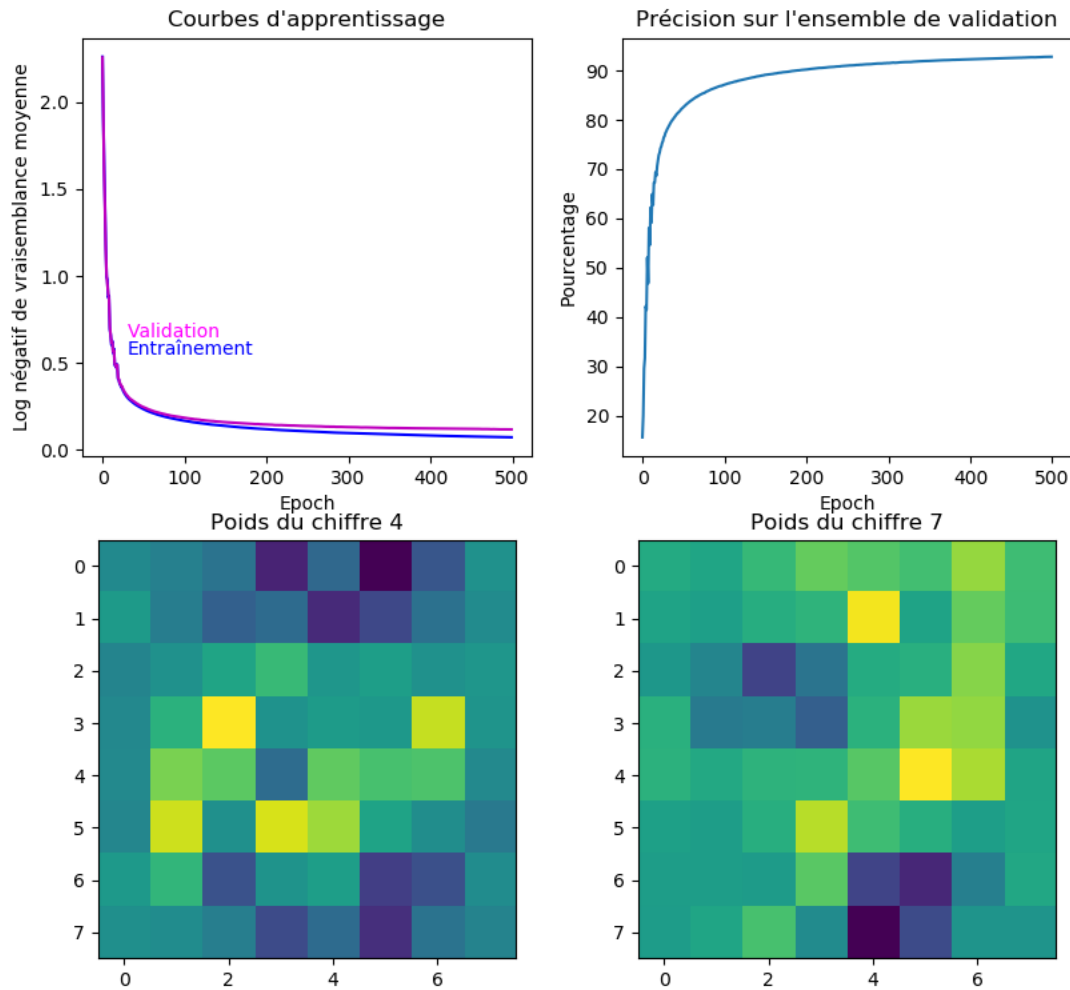


La précision obtenue = 95.1178993181

Pour un lr = 0.01

Avec une taille de mini-batch = 200

## Expérience 8 : taux d'apprentissage = 0.01, 1000 éléments/mini-batch

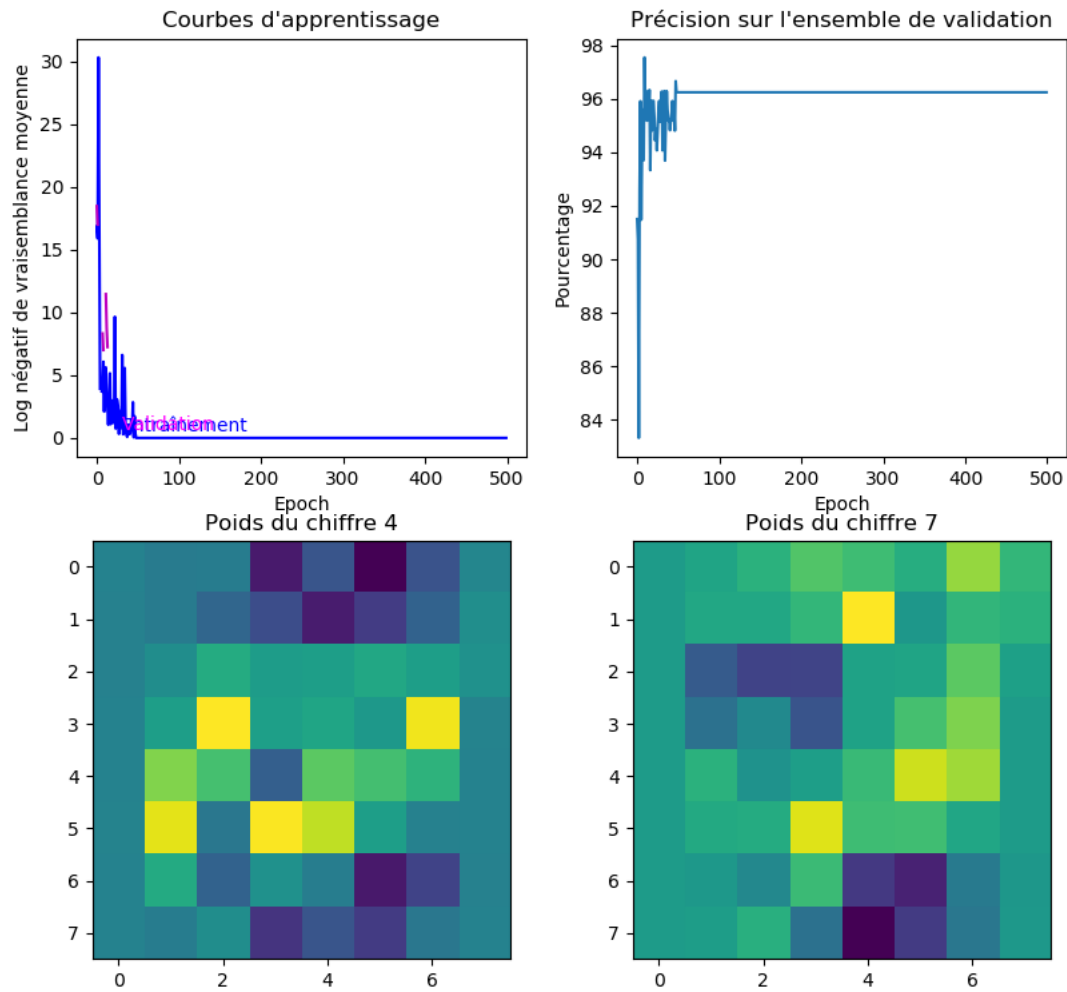


La précision obtenue = 92.9146203084

Pour un lr = 0.01

Avec une taille de mini-batch = 1000

## Expérience 9 : taux d'apprentissage = 0.1, 1 élément/mini-batch

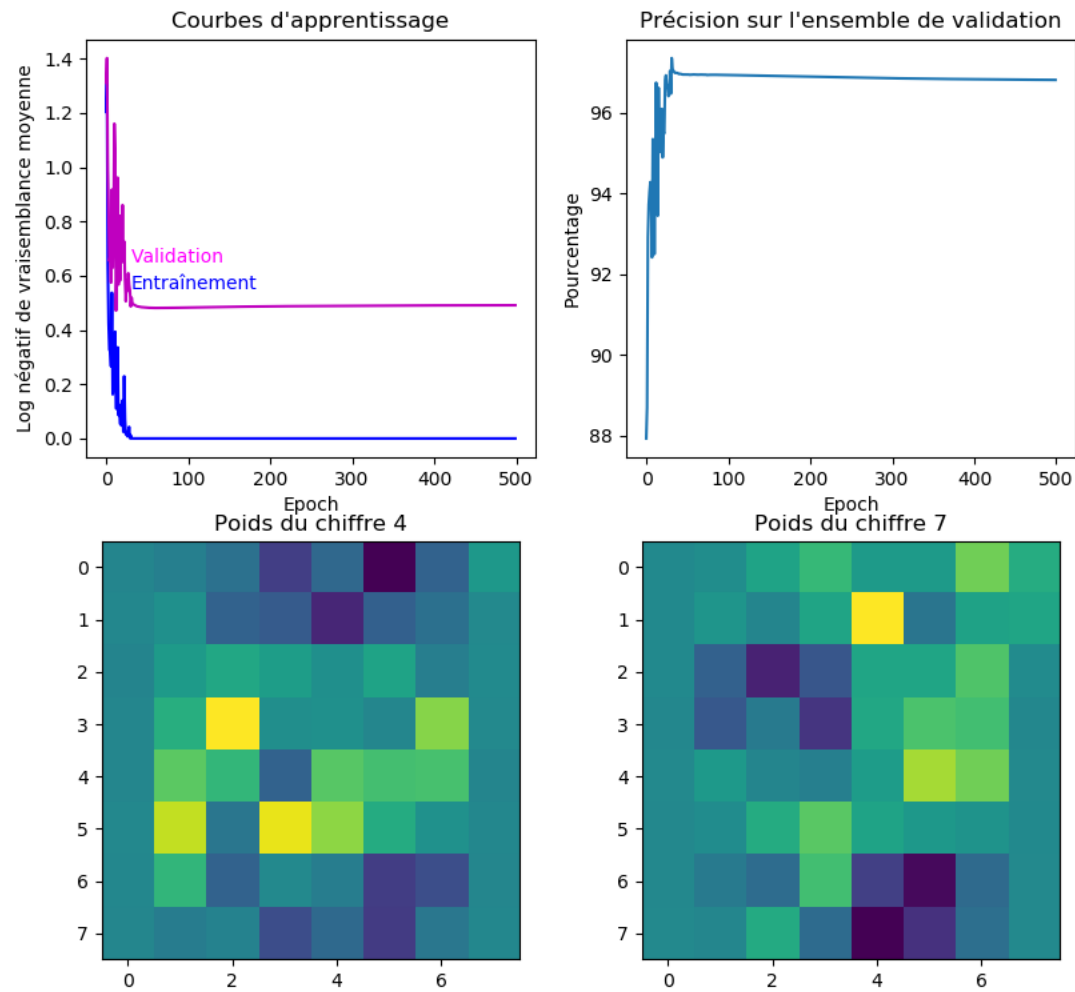


La précision obtenue = 94.9287788112

Pour un lr = 0.1

Avec une taille de mini-batch = 1

## Expérience 10 : taux d'apprentissage = 0.1, 20 éléments/mini-batch

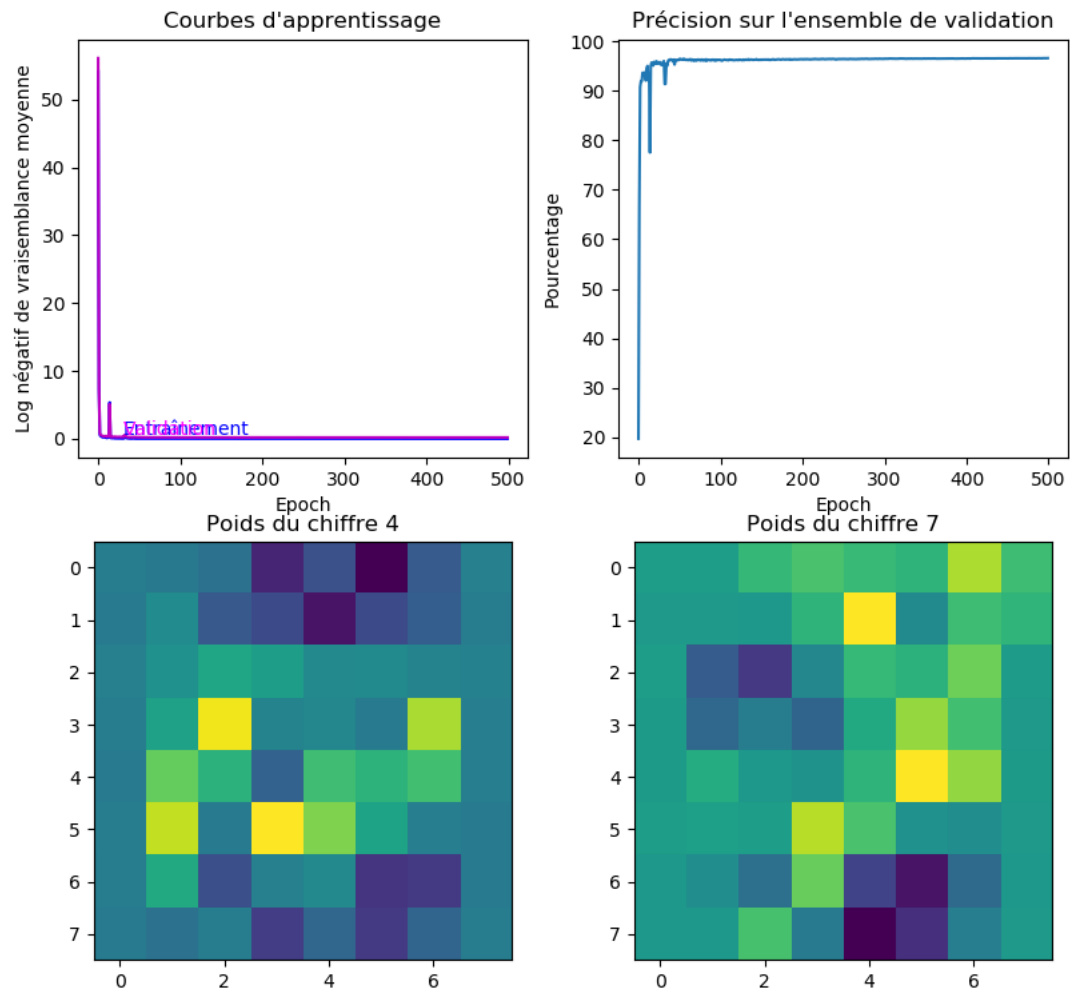


La précision obtenue = 95.7615324382

Pour un lr = 0.1

Avec une taille de mini-batch = 20

## Expérience 11 : taux d'apprentissage = 0.1, 200 éléments/mini-batch

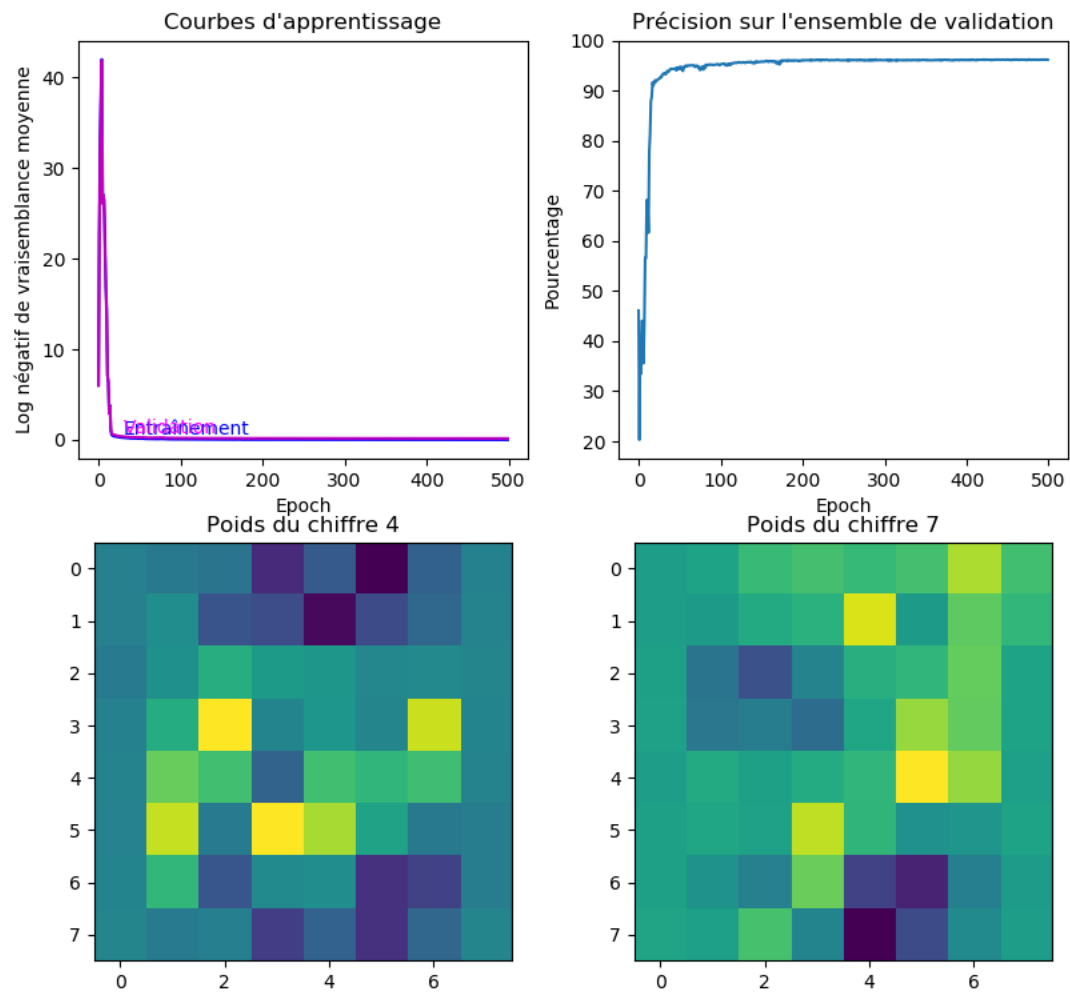


La précision obtenue = 96.2129104181

Pour un lr = 0.1

Avec une taille de mini-batch = 200

## Expérience 12 : taux d'apprentissage = 0.1, 1000 éléments/mini-batch



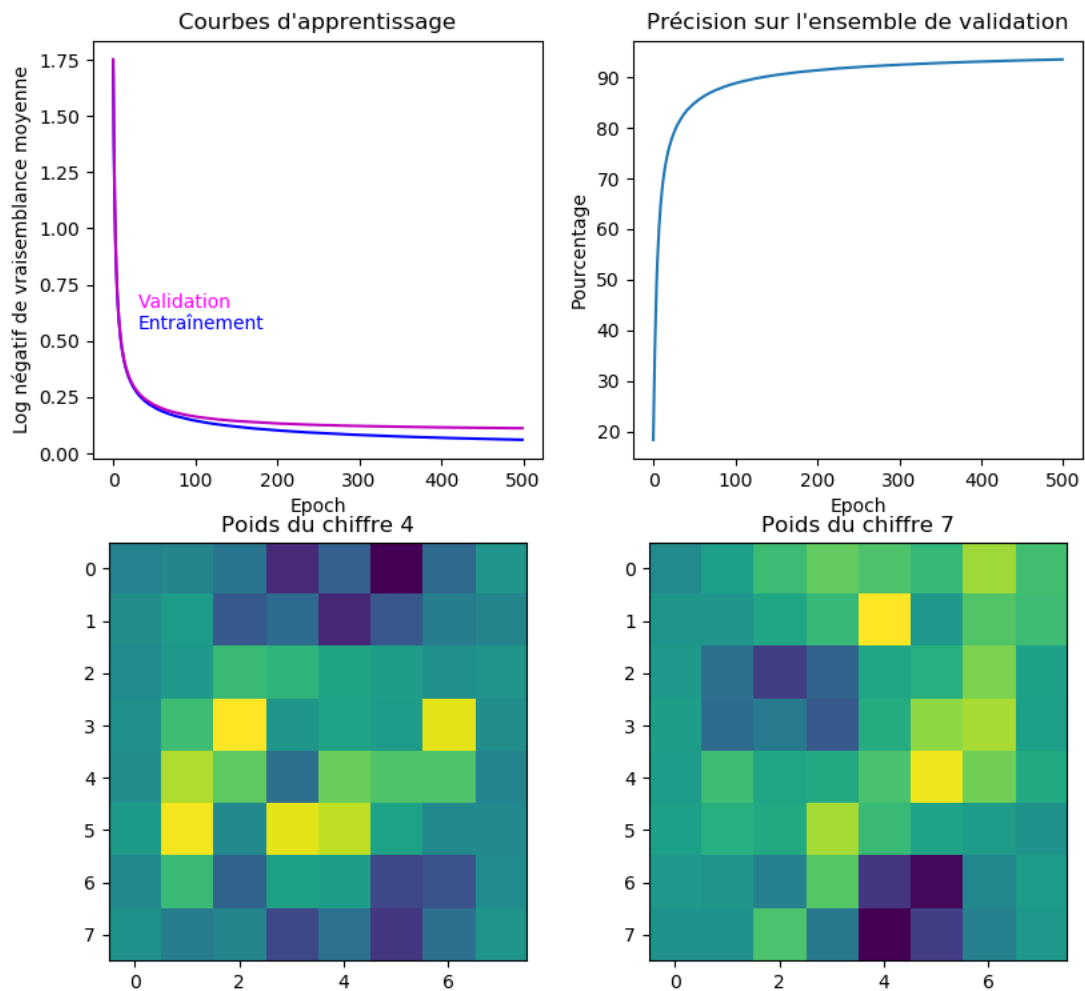
La précision obtenue = 96.1377368051

Pour un lr = 0.1

Avec une taille de mini-batch = 1000

### Expérience 13 : taux d'apprentissage = 0.001, 89 éléments/mini-batch

C'était les paramètres par défaut du fichier (avec `minibatch_size = len(y) // 20`).



La précision obtenue = 93.4034487219

Pour un lr = 0.001

Avec une taille de mini-batch = 89