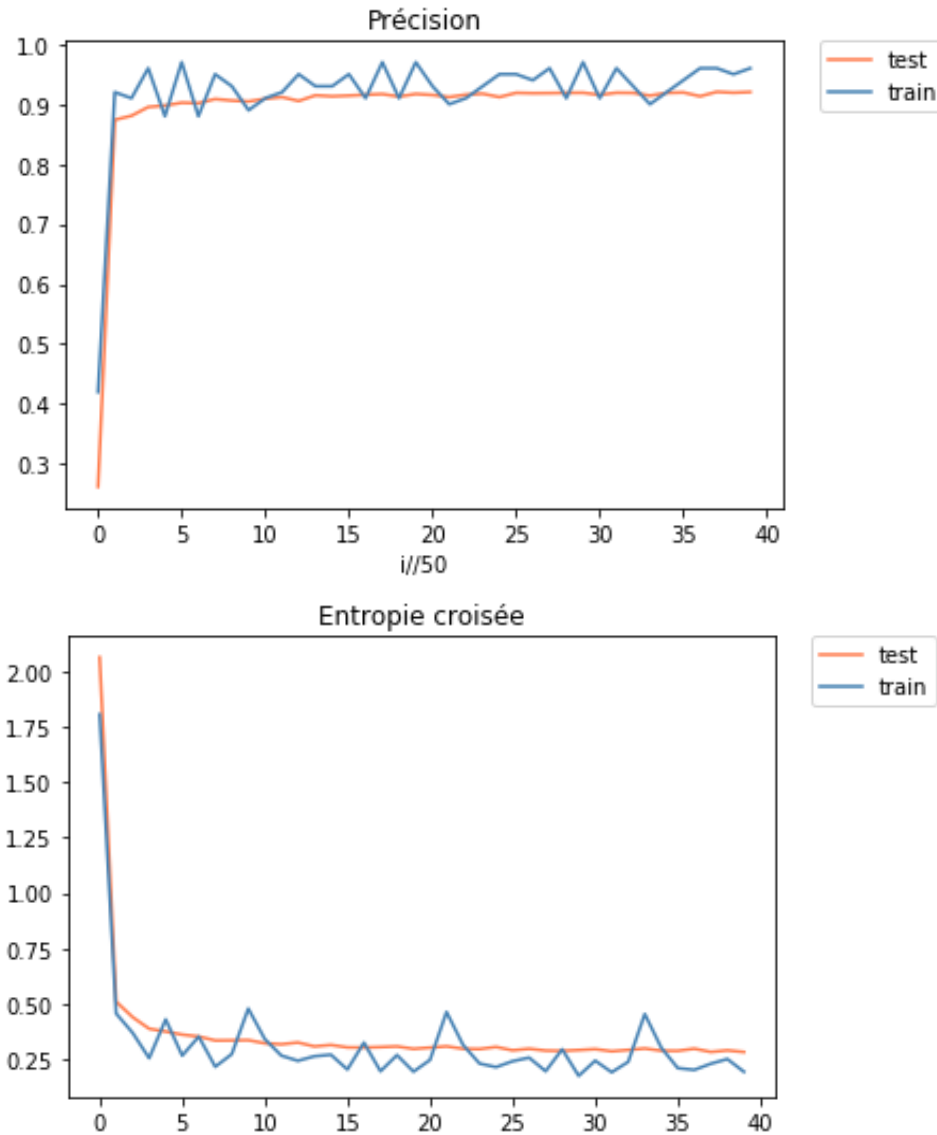


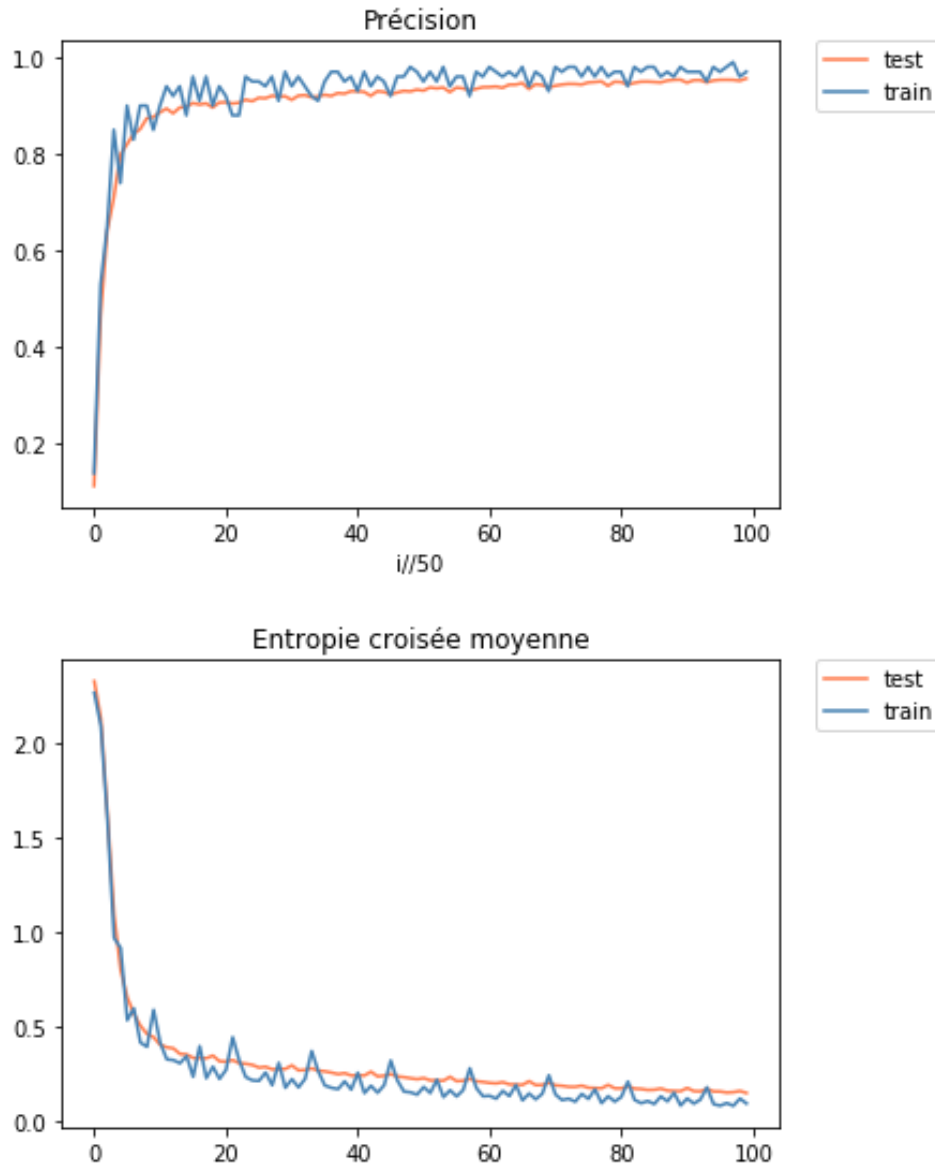
Les résultats de Simulation

➤ Les résultats de Réseau neuronal mono couche – Softmax :



On aura 92,06% de réussite sur un jeu de données jamais rencontrées lors de l'apprentissage.

- Si on ajout de couches de neurones intermédiaires avec sigmoïde comme fonction d'activation. La dernière couche de neurones reste la même que précédemment avec Softmax :

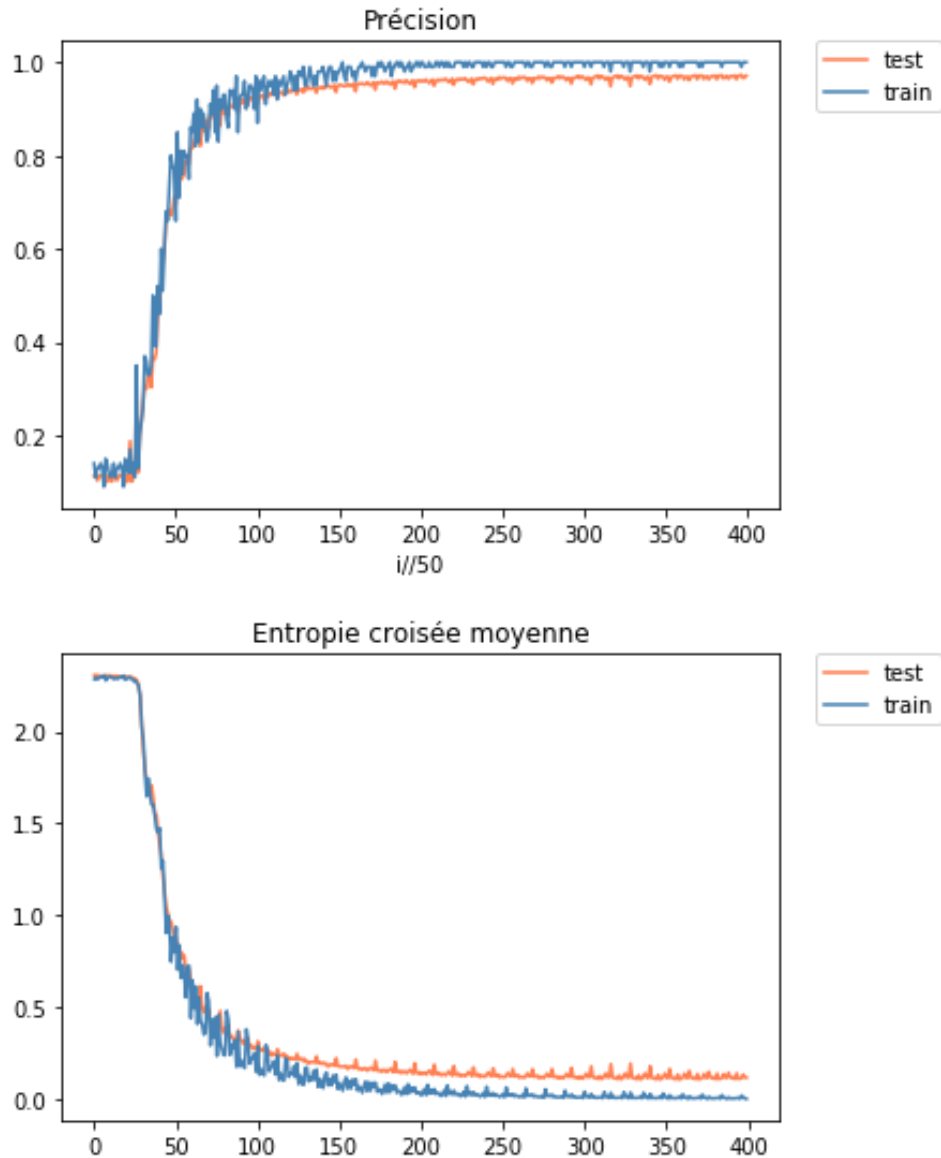


On aura 95,39% de réussite sur l'ensemble de test.

➤ **Réseau neuronal profonde:**

Deep neural network : 5 niveaux. Nécessite plus d'itérations pour converger.

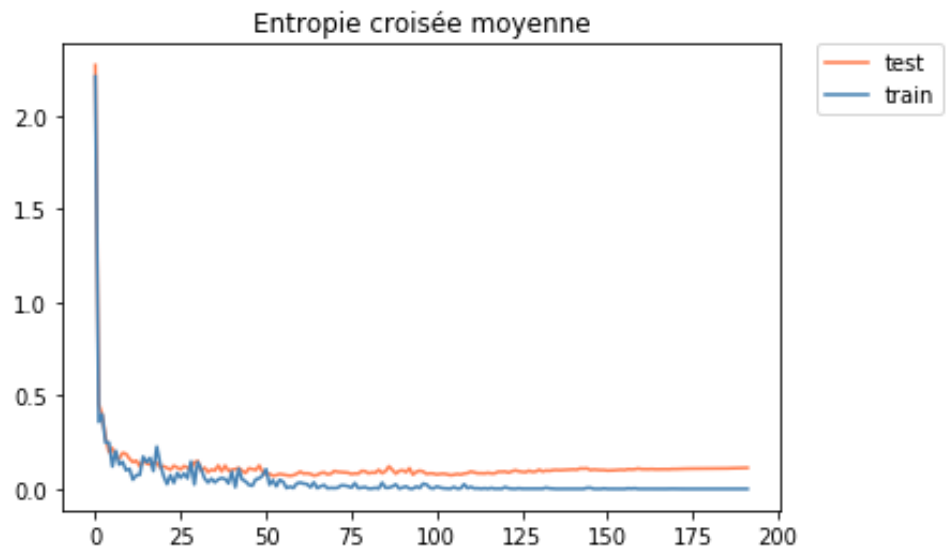
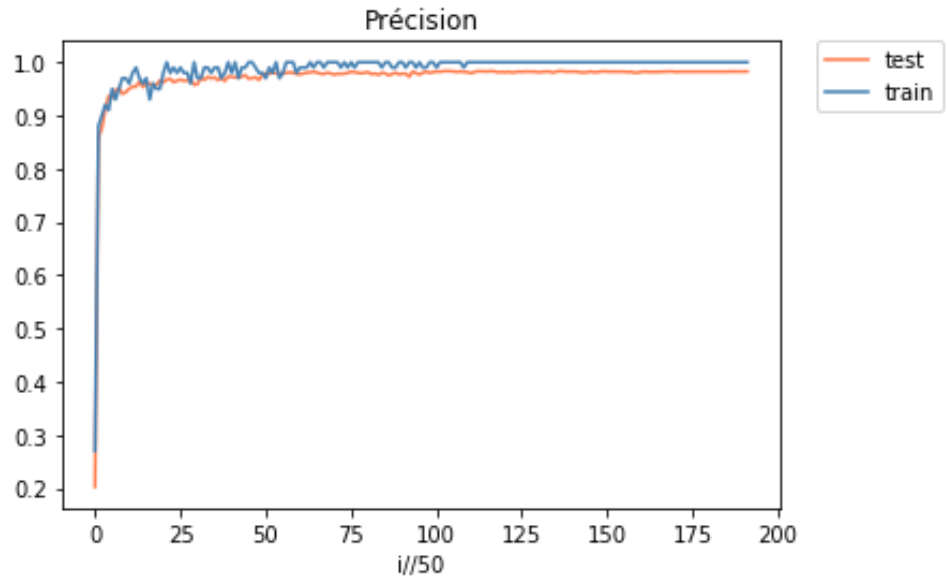
- Poids W initialisés aléatoirement selon une gaussienne
- Biais b initialisés à 0
- Optimisation GradientDescentOptimize



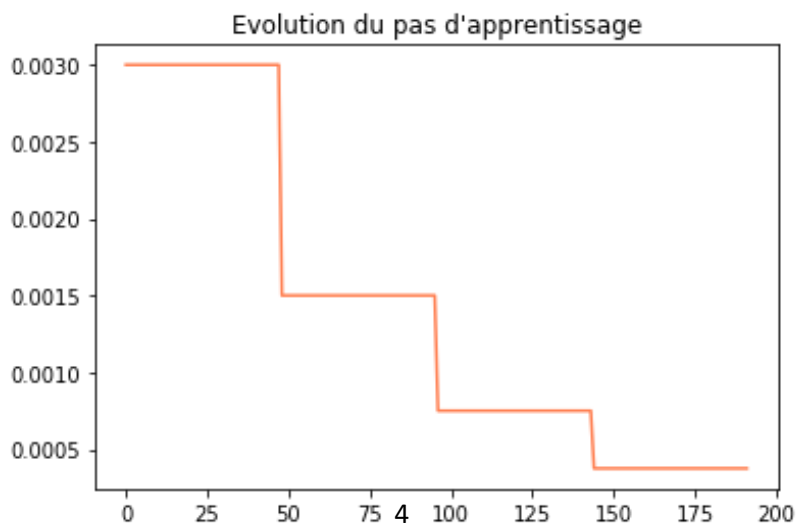
Meilleure précision au bout de 10000 itérations : 96,15, Au bout de 34 époques précision de 97,24%.

➤ Optimisations pour les réseaux neuronaux profonds :

- ReLU - Rectified Linear Unit :
L'utilisation de la fonction sigmoïde peut s'avérer problématique car toutes les sorties des neurones sont comprises entre 0 et 1, ce qui peut amener à des sorties et des gradients qui deviennent non significatifs. On préfère dorénavant utiliser la fonction ReLU `tf.nn.relu`.



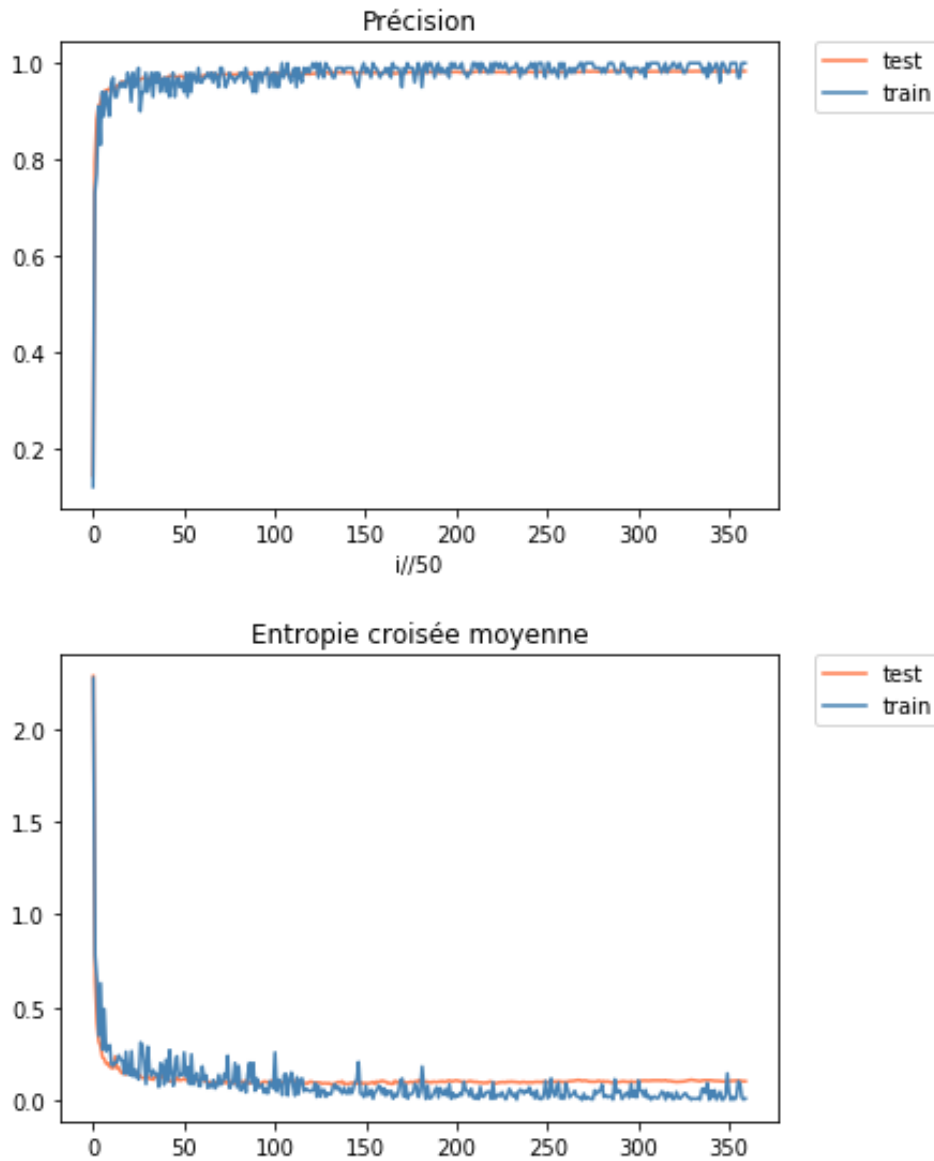
Meilleure précision au bout de 8800 itérations : 98,4%. Légèrement variable d'un coup à l'autre, certainement dû aux conditions initiales définies aléatoirement.



➤ **Overfitting et dropout :**

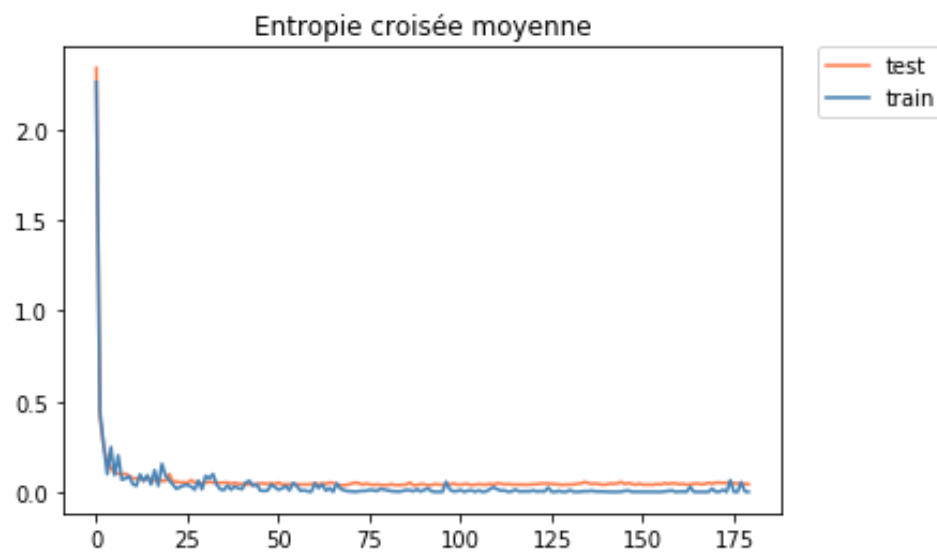
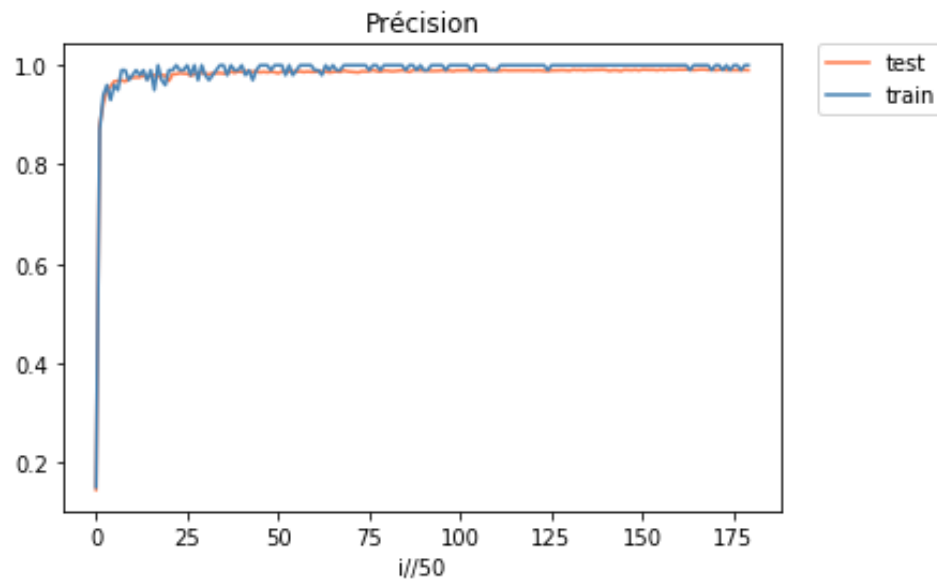
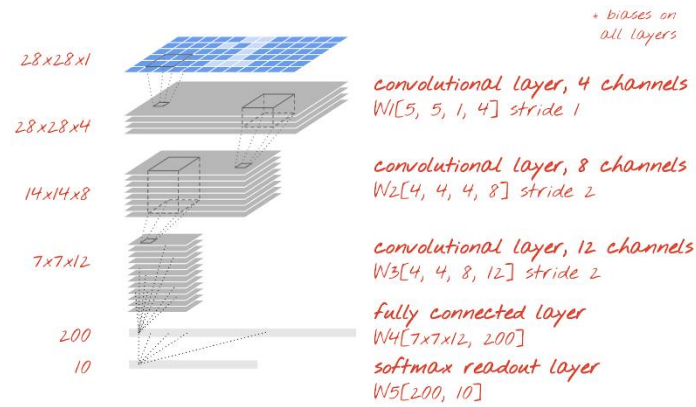
Gestion du sur-apprentissage avec la suppression de la réponse de certains neurones.

- Reformulation du réseau neuronal à 5 couches :



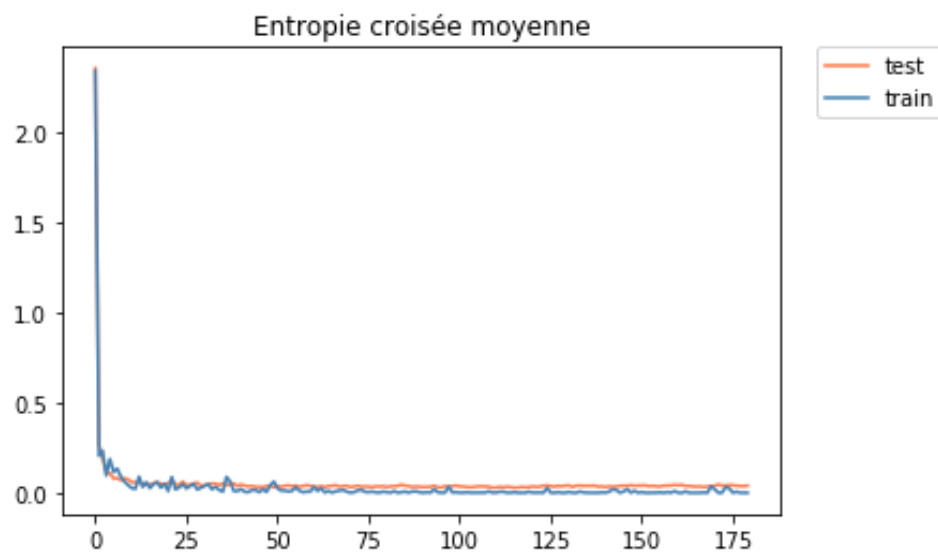
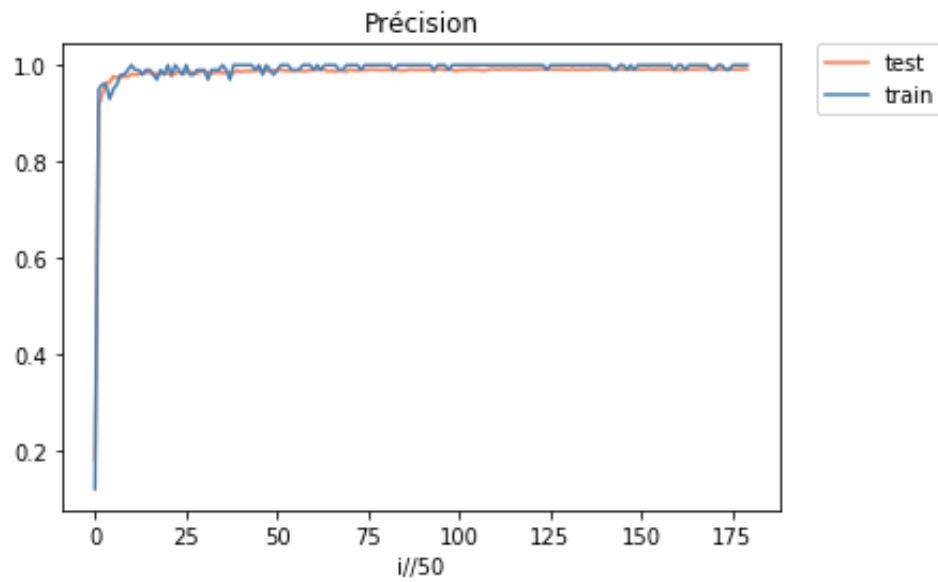
On aura une précision = 98,42 %. La progression est plus chaotique, mais on n'observe plus de divergence entre l'entropie des deux jeux de données. Par contre il n'y a pas d'amélioration notable par rapport au test précédent.

➤ Réseau neuronal convolutif :



Meilleur score pour l'itération 8100 avec une précision = 99,15%

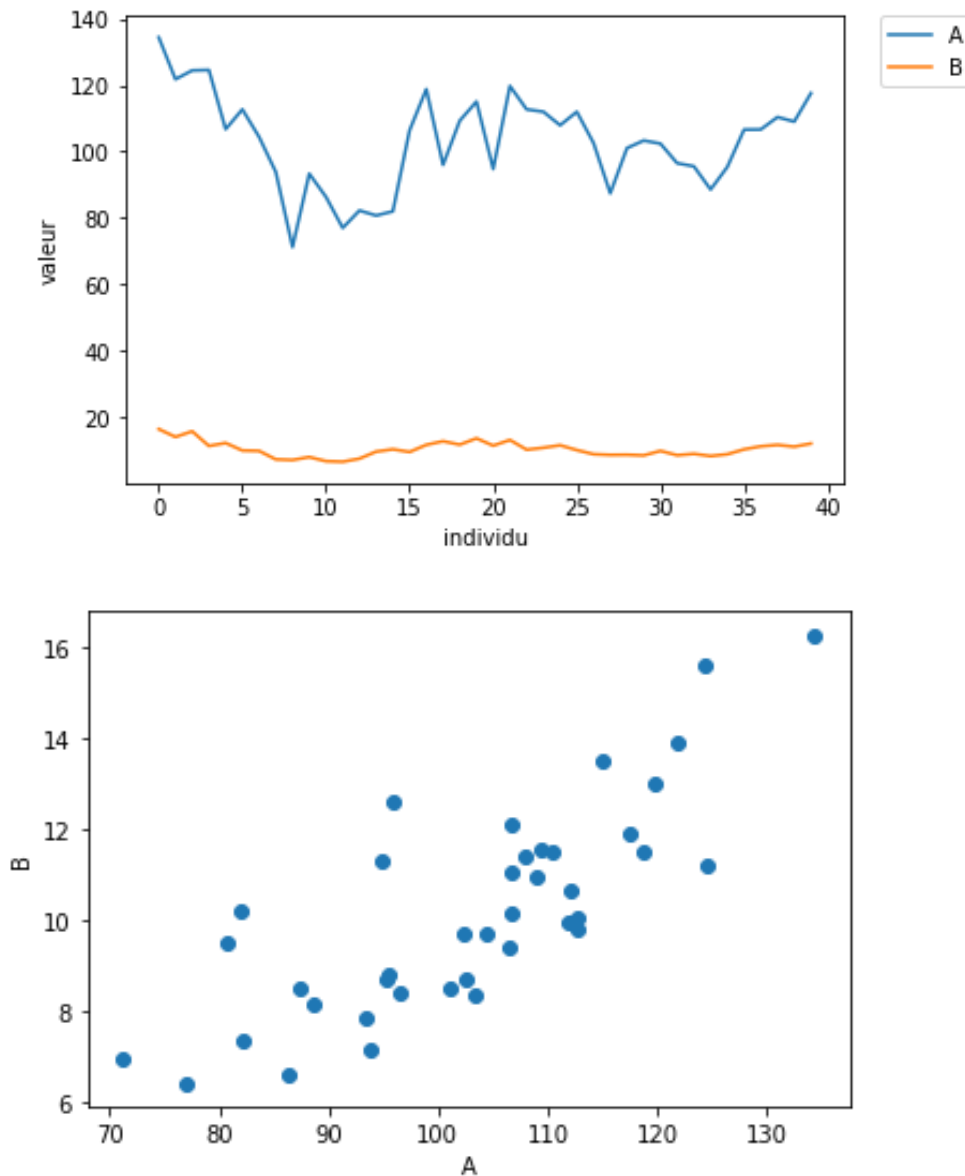
- Modification du réseau précédent pour augmenter le nombre de canaux de sortie de la première couche. Le but est de prendre en compte plus de traits caractéristiques de nos images d'entrée.



On remarque qu'on aura 99.33 % de taux de reconnaissance sur le jeu de test.

➤ Batch normalization :

La "Batch Norm" est une technique de régularisation qui va permettre d'avoir des ordres de grandeurs équivalents pour les entrées d'une fonction d'activation.



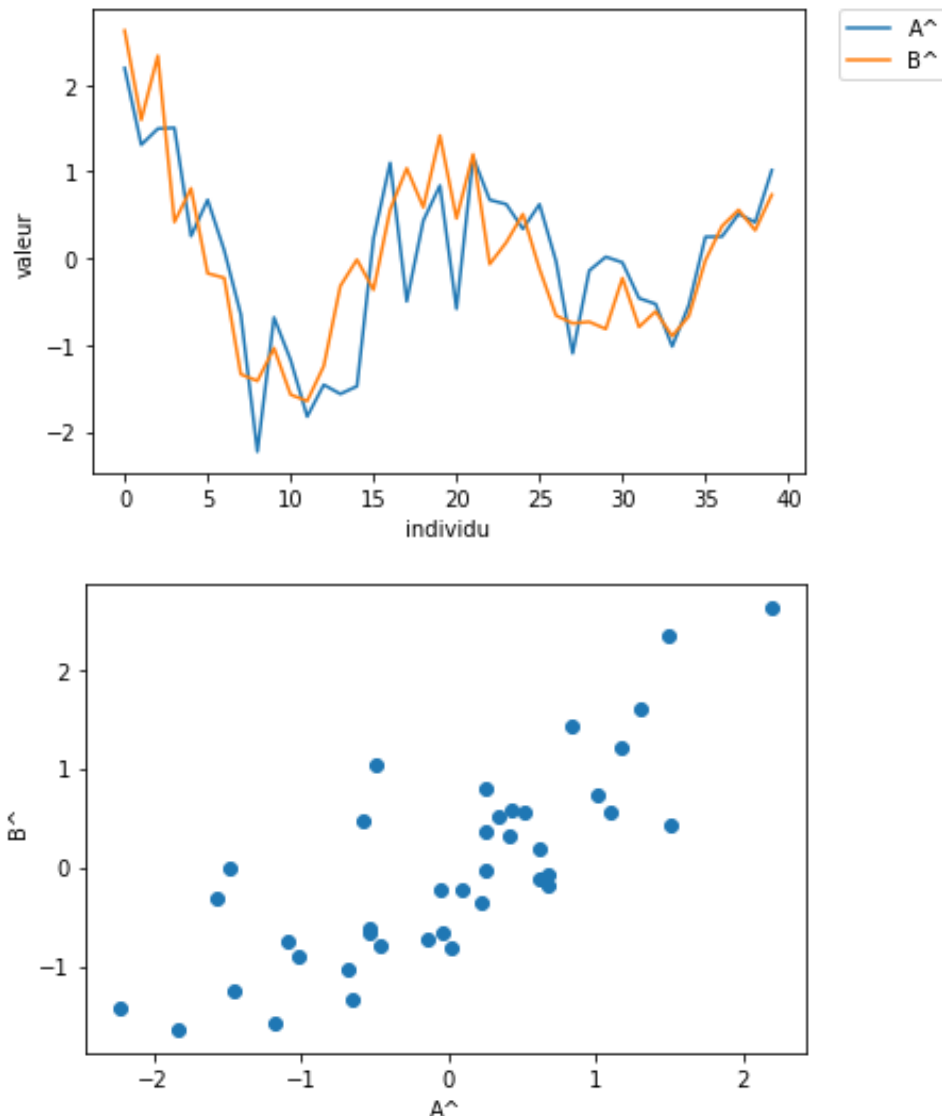
Si on a un jeu de données de ce type et que l'on cherche à créer un réseau de neurones basé sur ces éléments avec un vecteur d'entrée qui comporte $[A, B]$, on se heurte au problème d'avoir de très mauvaises données en entrée. Les ordres de grandeurs ne sont pas du tout similaires entre les caractéristiques A et B. Lors du passage dans un neurone comme on multiplie l'entrée par un poids w , l'activation sera bien plus importante lorsque la valeur d'entrée sera grande. Ce qui crée des déséquilibres dans le réseau neuronal et entraîne une perte d'efficacité.

Pour ramener les jeux de données à des ordres de grandeurs similaires et les centrer autour de zéro: on soustrait la moyenne et on divise par l'écart type.

$$A' = (A - \text{mean}(A)) / \text{std}(A)$$

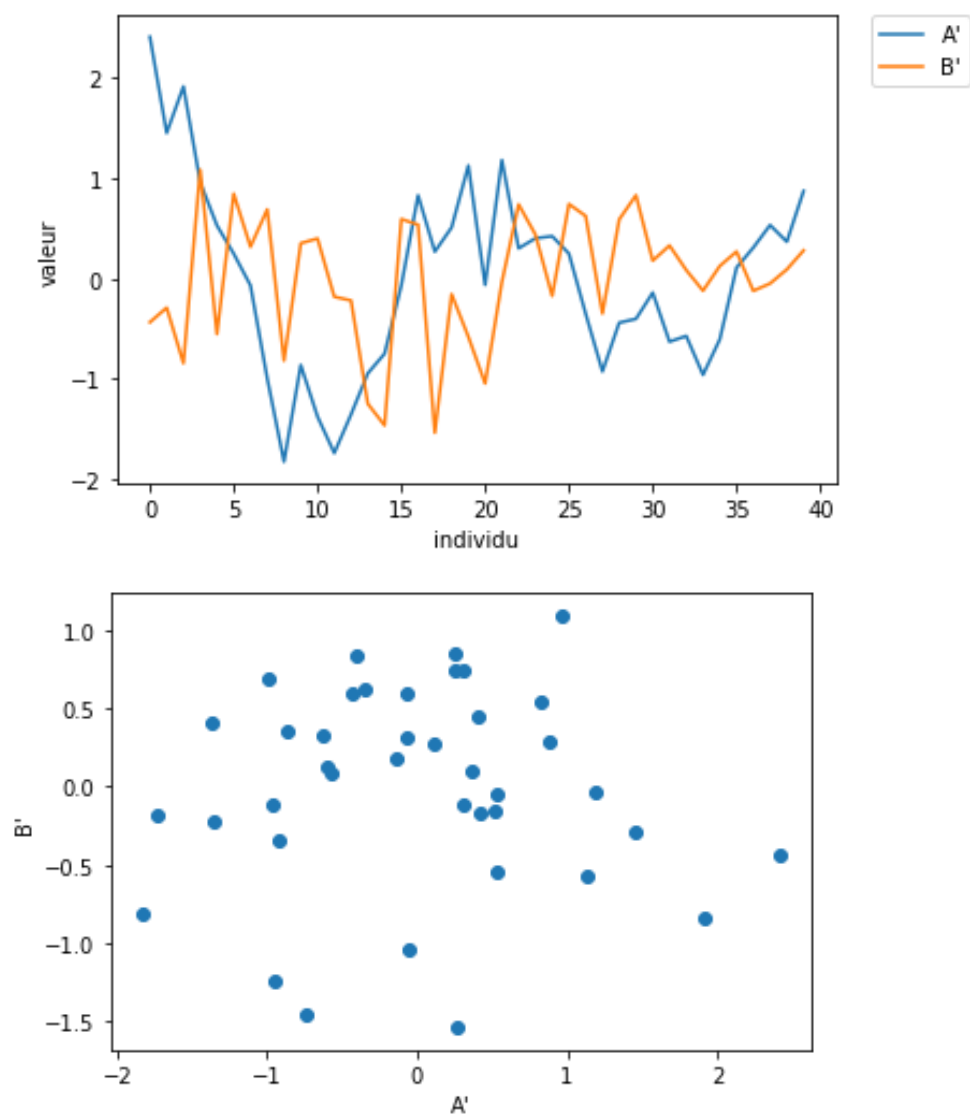
En notation mathématique :

$$A' = (A - \bar{A}) / \sigma_A$$

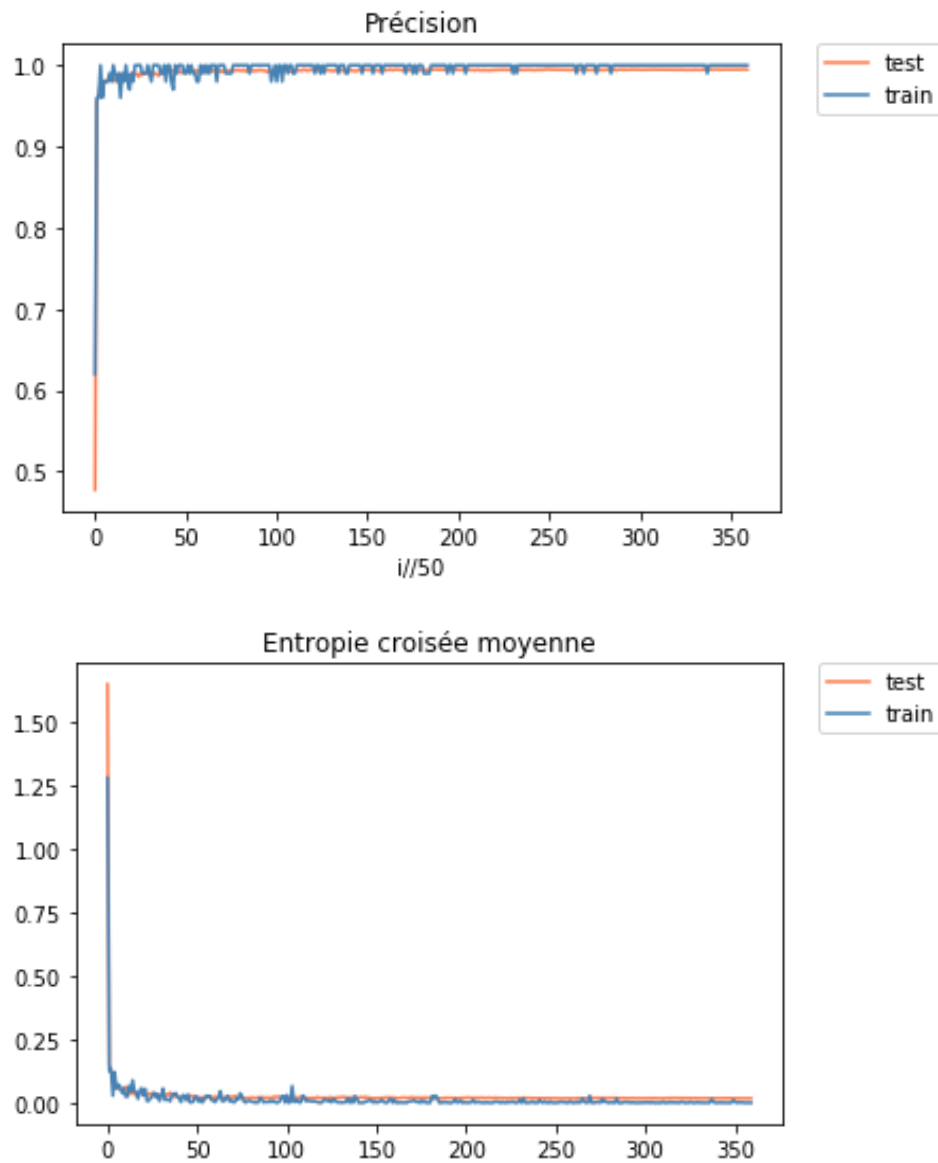


En faisant cette opération, on remarque que les variables semblent corrélées car elle évolue de façon similaire (et le nuage de points de B en fonction de A s'oriente principalement selon une diagonale).

On suppose qu'il y a de l'information à exploiter dans ces signaux, mais pour que le réseau soit efficace il faut que les variables d'entrée soient décorrélées (en tous cas que la corrélation ne soit pas évidente et que le nuage de points soit réparti de façon homogène). Pour cela on va suivre l'évolution des variables:



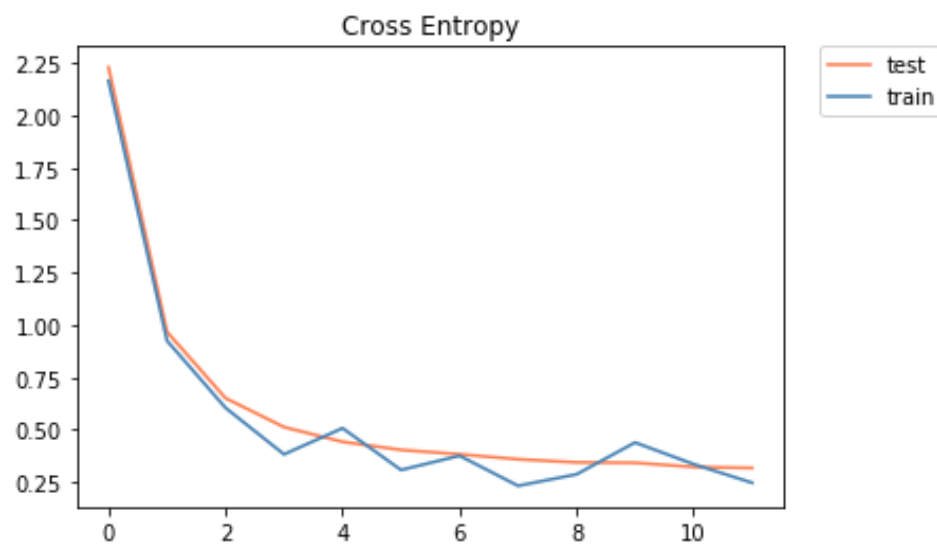
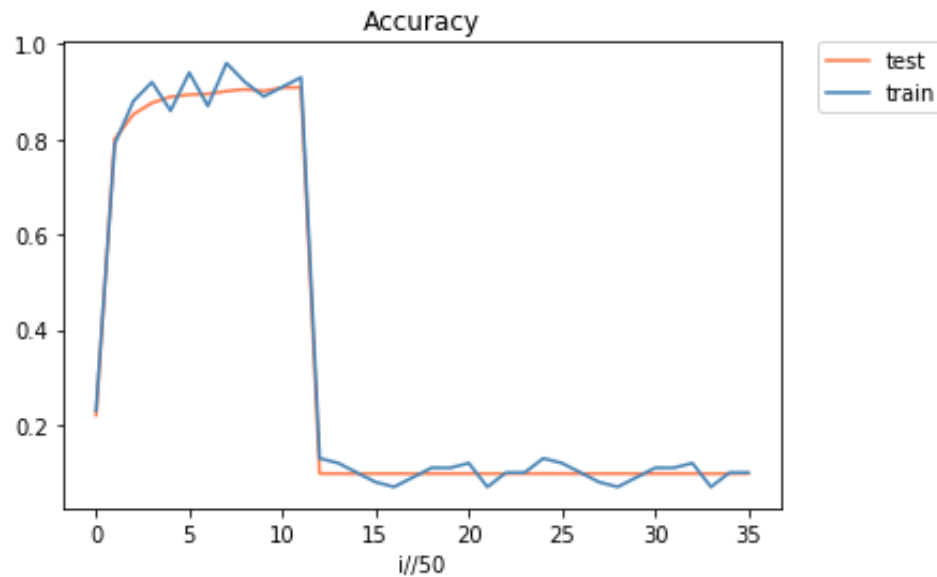
➤ **Modification du meilleur réseau convolutionnel pour intégrer la batch norm :**



Meilleur score pour l'itération 17850 avec précision égale à 0.9952, 99.52 % de taux de reconnaissance sur le jeu de test.

➤ **Test batch_normalization :**

Test des problèmes avec batch_normalization et quelques options de travail.



Meilleur score pour l'itération 550 | Précision = 0,9090.