

## TP2

November 28, 2019

### Apprentissage par perceptron multi-couche

Construire un modèle de classification ayant comme paramètre : `hidden_layer_sizes = (50)`, puis calculez la précision du classifieur

Score: 88.23809523809524

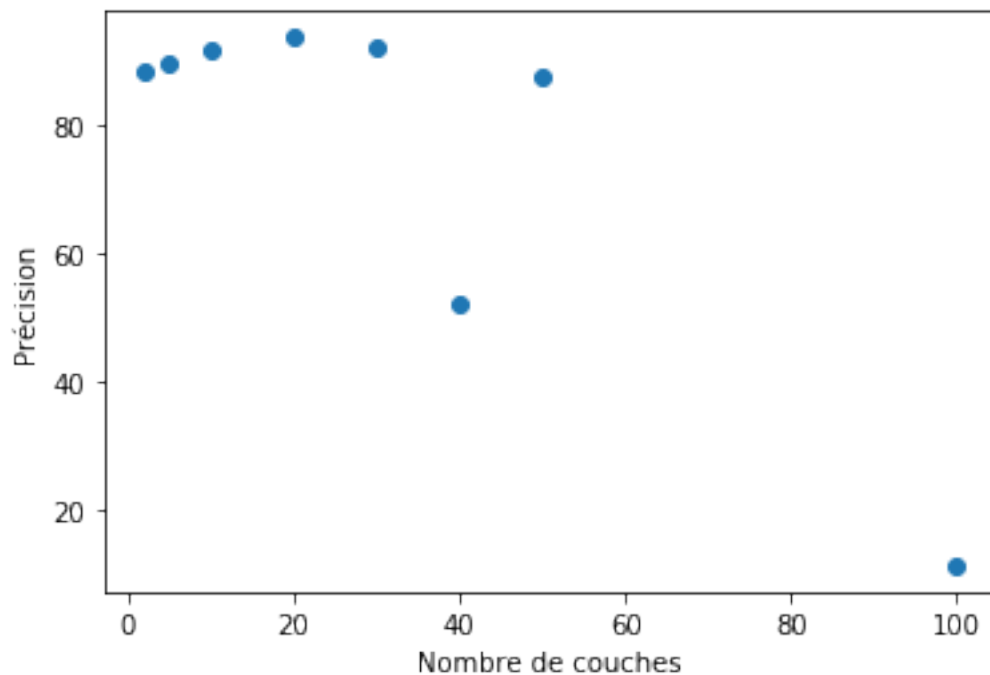
Error: 11.76190476190476

Afficher la classe de l'image 9 et sa classe prédite

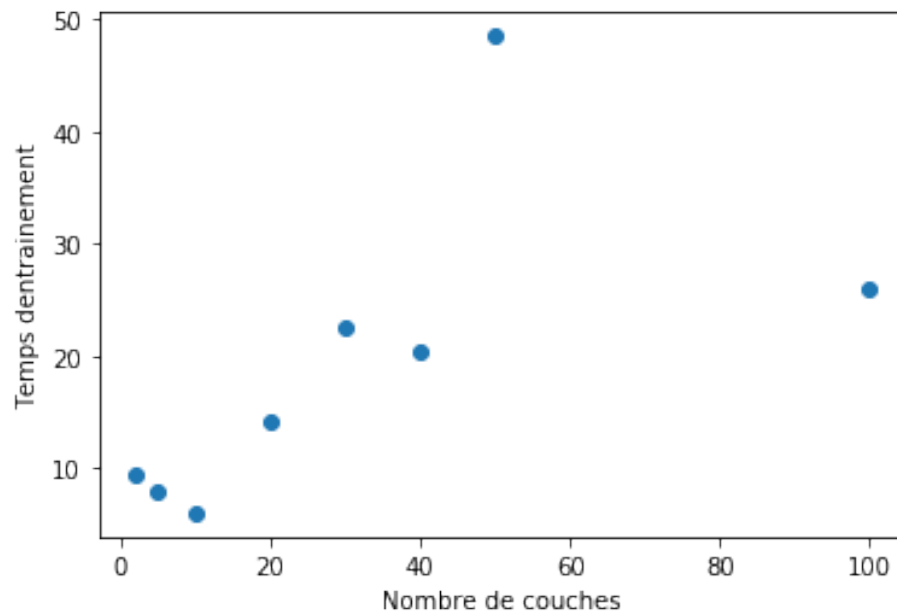
Prediction = ['4'] et nombre réel = 9

Varier le nombre de couches de 1 entre (2 et 100) couches, et recalculer la précision du classifieur

Précision du Classificateur en fonction du nombre de couches



### Temps d'entraînement du Classificateur en fonction du nombre de couches



**Construire 2 modèles de classification des données mnist, avec des réseaux qui ont respectivement :**

- Modèle1: 50 couches cachées, et des tailles de couches de manière régressives de 60 à 11 par pas de -1
- Modèle2: 20 couches cachées, et des tailles de couches de manière régressives de 60 à 33 par pas de -3, puis -2

Quelles sont les performances en taux de bonne classification et en temps d'apprentissage obtenus pour chaque modèle ? Utilisez la fonction `time()` du package `time` pour mesurer le temps d'apprentissage d'un modèle.

- Modèle 1 :

Score = 11.33333333333332

Temps d'entraînement = 13.117745876312256

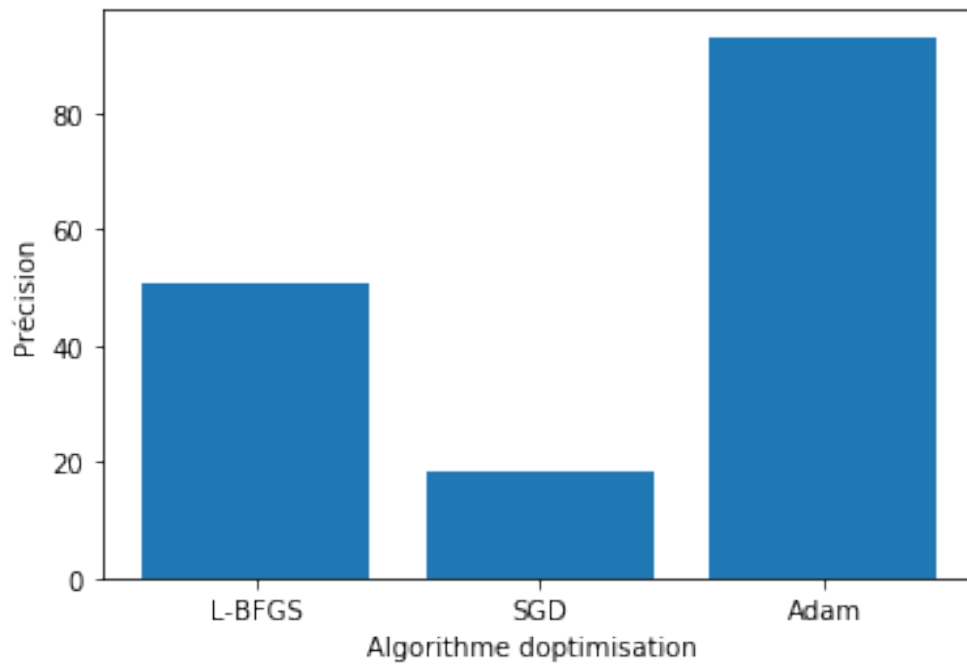
- Modèle 2 :

Score = 92.61904761904762

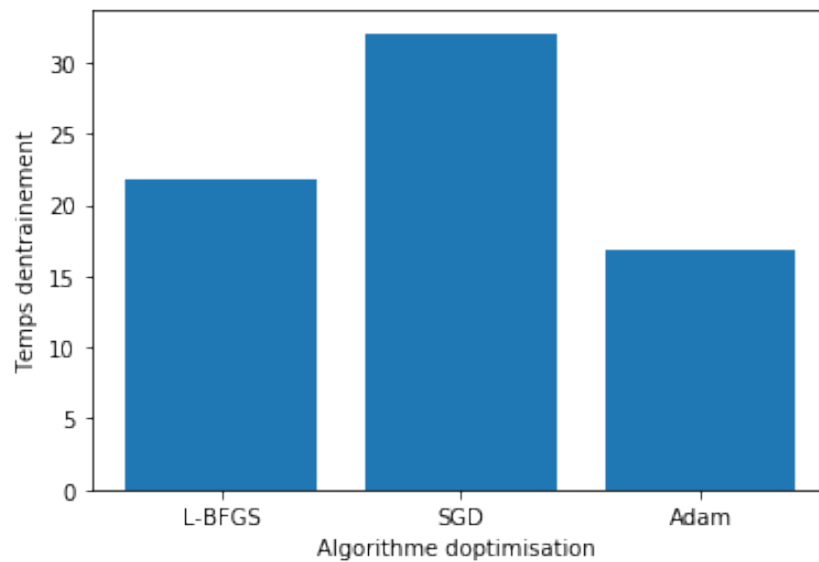
Temps d'entraînement = 20.750596523284912

**Etudier la convergence des algorithmes d'optimisation disponibles : L-BFGS, SGD et Adam**

Précision du Classificateur en fonction de l'algorithme d'optimisation

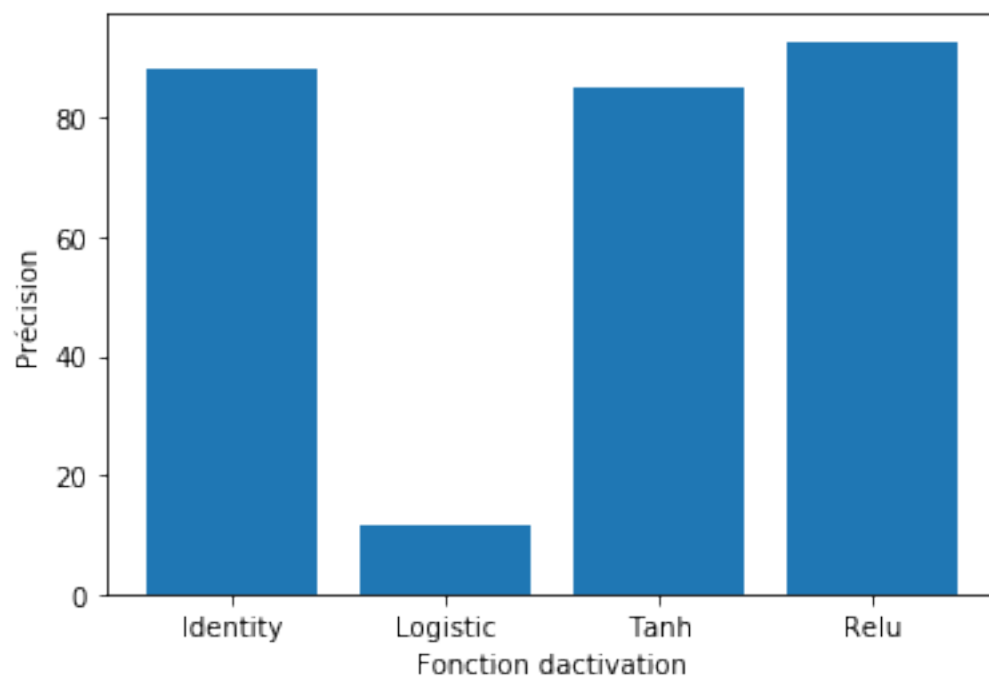


Temps d'entraînement du Classificateur en fonction de l'algorithme d'optimisation

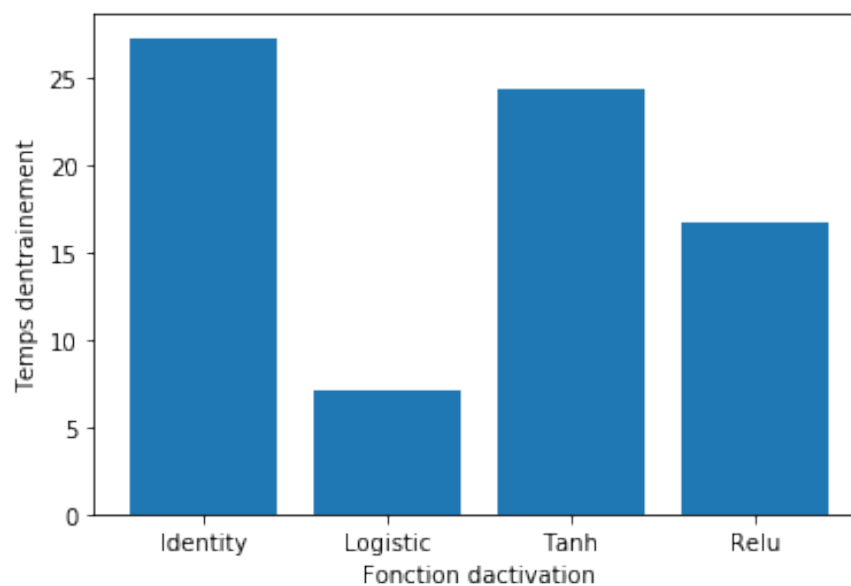


Varier les fonctions d'activation {'identity', 'logistic', 'tanh', 'relu'}

Précision du Classificateur en fonction de la fonction d'activation

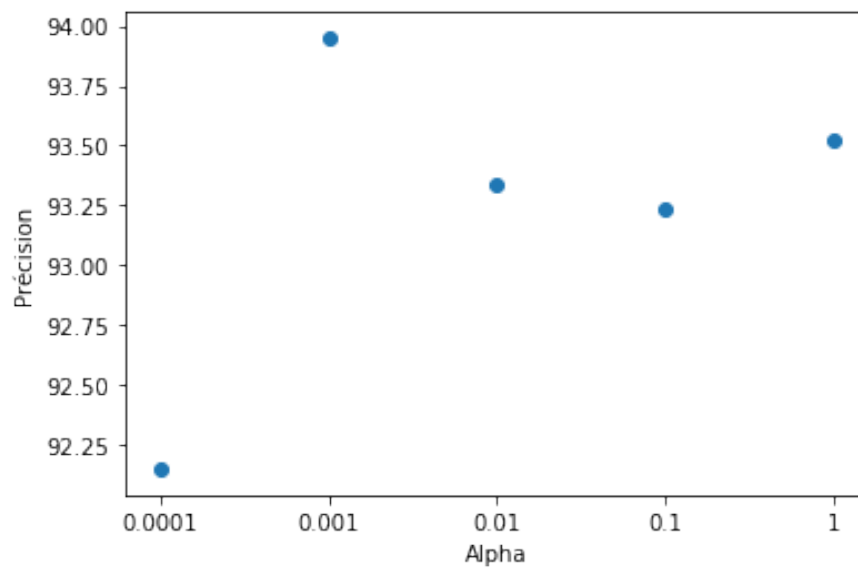


Temps d'entraînement du Classificateur en fonction de la fonction d'activation

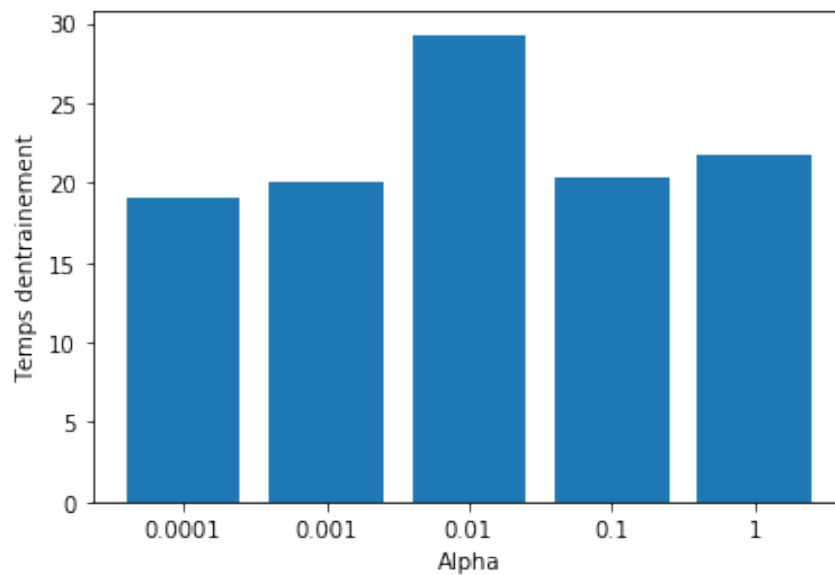


Changer la valeur de la régularisation L2 (paramètre  $\alpha$ )

Précision du Classificateur en fonction du paramètre de régularisation L2 ( $\alpha$ )



Temps d'entraînement du Classificateur en fonction de régularisation L2 ( $\alpha$ )



**Choisissez le modèle qui propose de meilleurs résultats** Au vu de toutes les expérimentations vues jusqu'à présent, le modèle qui propose de meilleurs résultats est un modèle ayant comme

paramètres: - Un nombre de couches compris aux alentours de 20 (entre 15 et 25 environ, selon le 1er graphique), en méthode régressive avec un nombre de neurones compris entre 60 et 10 même avec un pas d'environ -2 ou -3 - Adam, comme algo d'optimisation - Relu, comme fonction d'activation. Les autres sont assez proches en terme de résultat, à l'exception de 'logistic' qui n'est pas précise - Un paramètre de régulation le plus petit possible (environ 0.0001 ou moins)

**A votre avis, quels sont les avantages et les inconvénients des A-nn : optimalité ? temps de calcul ? passage à l'échelle ?**

**Avantages:**

- Tolérance aux fautes: Même si un ou plusieurs neurones sont corrompus, cela n'empêche pas la génération
- Capacité à faire du calcul en parallèle
- Capacité à générer des estimations pertinentes à partir de peu d'exemples

**Inconvénients:**

- Architecture: Il n'existe pas vraiment de moyens permettant de définir l'architecture optimale du
- Décisions prises qui sont difficiles à analyser ou expliquer
- Difficultés à estimer les temps de calcul du A-nn