
Guide

Challenge MNIST par caractéristiques moyennes

Stéphane Mallat, John Zarka

27 janvier 2023

Le challenge MNIST de classification d'images demande de développer un algorithme qui reconnaît un chiffre y entre 0 et 9 apparaissant dans une image \mathbf{x} , avec le moins d'erreur possible. C'est l'un des challenges qui a eu le plus d'impact pour la recherche en apprentissage et intelligence artificielle. Il a été proposé en 1998 par Corinna Cortes, Yann LeCun et Christopher Burges. Il est à l'origine de milliers d'articles scientifiques qui ont progressivement réduit le pourcentage d'erreurs avec des algorithmes de plus en plus élaborés. Nous proposons ici une approche du challenge, en cherchant des caractéristiques de l'image dont les moyennes permettent d'identifier chaque chiffre.

Les données et le score La base de données MNIST comprend 60 000 images d'entraînement \mathbf{x}_{train} associées au chiffre représenté y_{train} pour chaque image. On utilisera cette base de données d'entraînement pour chercher et programmer un algorithme qui calcule une estimation $f(\mathbf{x})$ du chiffre y dans l'image \mathbf{x} , qui est une fonction de \mathbf{x} . L'algorithme fait une erreur si $f(\mathbf{x}) \neq y$. Le challenge fournit aussi des images \mathbf{x}_{test} de test. Le score de votre algorithme est le pourcentage d'erreurs qu'il fait sur ces images de test.

Vos prédictions $f(\mathbf{x}_{test})$ effectuées sur les images de test \mathbf{x}_{test} doivent être envoyées sur la page dédiée au challenge sur la plateforme Challenge Data à l'adresse <https://challengedata.ens.fr>. La page vous renvoie alors votre score en comparant vos prédictions $f(\mathbf{x}_{test})$ aux chiffres y_{test} , et vous positionne sur le "leaderboard" de la classe en utilisant votre meilleur score.

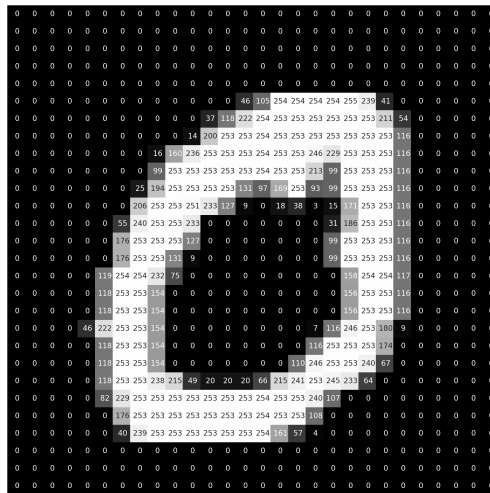


Figure 1: Exemple d'image \mathbf{x} de la base de données MNIST représentant le chiffre $y = 0$. \mathbf{x} est un tableau de 28×28 pixels codés par des entiers compris entre 0 et 255.

Images Une image \mathbf{x} est un tableau informatique de $N \times M$ entiers compris entre 0 et 255. Chaque valeur $\mathbf{x}[i, j]$ donne le niveau de gris d'un point de l'image à la position (i, j) , appelé pixel. Si $\mathbf{x}[i, j] = 0$, le pixel apparaît noir tandis que si $\mathbf{x}[i, j] = 255$, il apparaît blanc. Une image numérique a typiquement 1000×1000 pixels. Les images de chiffres de MNIST sont des petites images de 28×28 pixels. La Figure 1 donne un exemple d'image \mathbf{x} dont le chiffre est $y = 0$. On note en caractère gras les images et les tableaux de nombres.

1 Challenge MNIST-2 : classifier les chiffres 0 et 1

Cette première section vous guidera pour développer un algorithme de classification, en se concentrant dans un premier temps sur les images \mathbf{x} représentant seulement les chiffres $y = 0$ ou $y = 1$. On nommera MNIST-2 ce sous-ensemble d'images de MNIST et challenge MNIST-2 le problème de classification associé. La Figure 2 montre des exemples d'images de 0 et de 1 au sein de MNIST-2.

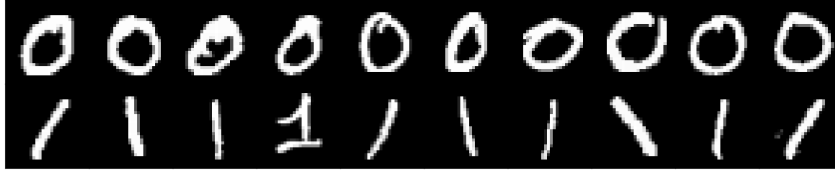


Figure 2: Exemples d'images de 0 et de 1 dans la base de données d'entraînement de MNIST-2.

1.1 Caractéristique discriminante d'une image

Nous allons commencer par calculer une valeur caractéristique $\phi(\mathbf{x})$ pour chaque image \mathbf{x} qui doit permettre de discriminer si \mathbf{x} est une image de 0 ou de 1.

En regardant les images de 0 et de 1 de la Figure 2, proposez une façon de calculer une caractéristique discriminante $\phi(\mathbf{x})$ à partir des 28×28 entiers du tableau $\mathbf{x}[i, j]$. Faites un programme qui calcule et imprime la valeur de $\phi(\mathbf{x})$ pour une image \mathbf{x} , et essayez-le sur des images de 0 et de 1. La caractéristique $\phi(\mathbf{x})$ semble-t-elle suffisamment discriminante ? Pouvez-vous l'améliorer ?

1.2 Discrimination par moyennes statistiques

On sépare les images d'entraînement \mathbf{x}_{train} associées aux chiffres $y = 0$ et $y = 1$ qui forment deux "populations" différentes, et on calcule la moyenne de la caractéristique $\phi(\mathbf{x})$ sur chaque population.

On note $\bar{\phi}(y)$ la moyenne de toutes les images \mathbf{x}_{train} d'entraînement correspondant à un chiffre y . Faites un programme informatique qui calcule $\bar{\phi}(0)$ pour $y = 0$ et $\bar{\phi}(1)$ pour $y = 1$.

On évalue la discrimination d'une caractéristique par le rapport Δ de la distance au carré entre $\bar{\phi}(0)$ et $\bar{\phi}(1)$ sur la somme de leurs carrés :

$$\Delta = \frac{|\bar{\phi}(0) - \bar{\phi}(1)|^2}{\bar{\phi}(0)^2 + \bar{\phi}(1)^2}. \quad (1)$$

Vérifiez que si $\phi(\mathbf{x})$ est toujours positif, alors $\Delta \leq 1$. Pouvez-vous utiliser cette valeur pour améliorer votre caractéristique ?

1.3 Classification de chiffres

Cherchez et programmez un algorithme de classification qui calcule une estimation $f(\mathbf{x})$ de la valeur y du chiffre d'une image \mathbf{x} , à partir de sa caractéristique $\phi(\mathbf{x})$, et des valeurs des moyennes $\bar{\phi}(0)$ et $\bar{\phi}(1)$.

1.4 Erreur d'entraînement

Faites un programme qui calcule le score égal au pourcentage d'erreur de votre algorithme sur les images \mathbf{x}_{train} de la base d'entraînement.

Quel est le score de votre algorithme sur les images d'entraînement ?

Pouvez-vous utiliser ce résultat pour améliorer votre caractéristique ou votre algorithme de classification ?

1.5 Erreur de test

Soumettez les estimations $f(\mathbf{x}_{test})$ calculées par votre algorithme sur les images de test \mathbf{x}_{test} sur la page du challenge sur la plateforme Challenge Data.

Quel est le score de votre algorithme ?

Ce score est-il le même que celui que vous avez obtenu obtenu sur les images d'entraînement ?

Pourquoi ?

1.6 Comprendre les erreurs

On veut comprendre d'où viennent les erreurs pour essayer de les réduire. À l'aide de la fonction `hist` de la bibliothèque `matplotlib` dont le fonctionnement est illustré dans le notebook d'utilitaires associé au challenge, calculez l'histogramme des caractéristiques $\phi(\mathbf{x}_{train})$ pour la population d'images pour lesquelles $y = 0$, et l'histogramme des caractéristiques $\phi(\mathbf{x}_{train})$ pour la population d'images pour lesquelles $y = 1$. Expliquez pourquoi votre algorithme fait une erreur en comparant ces deux histogrammes.

Donnez une condition sur les histogrammes pour que l'erreur puisse être nulle.

Pouvez-vous utiliser les histogrammes que vous avez calculés pour améliorer votre algorithme de classification ?

2 Challenge MNIST-4 : classifier les chiffres 0, 1, 4, 8

Le Challenge MNIST-4 propose de reconnaître les images \mathbf{x} de la base de données représentant les quatre chiffres $y \in \{0, 1, 4, 8\}$, sous-ensemble d'images de MNIST que l'on nomme MNIST-4. La Figure 3 montre des exemples d'images de 0, 1, 4 et 8 dans MNIST-4.

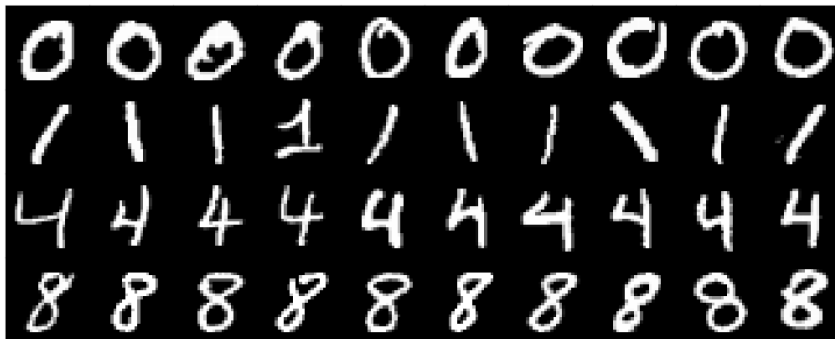


Figure 3: Exemples d'images de 0, 1, 4 et 8 dans la base de données d'entraînement de MNIST-4.

2.1 Classification avec une seule caractéristique

Nous allons utiliser la caractéristique $\phi(\mathbf{x})$ des images \mathbf{x} que vous avez définie dans le Challenge MNIST-2 de la section précédente en question 1.1, et calculer sa moyenne sur les quatre populations d'images \mathbf{x}_{train} pour lesquelles $y = 0, 1, 4$ ou 8 .

Calculez la moyenne $\bar{\phi}(y)$ de $\phi(\mathbf{x}_{train})$ pour chaque population $y \in \{0, 1, 4, 8\}$.

Cherchez et programmez un algorithme qui calcule une estimation $f(\mathbf{x})$ du chiffre y d'une image \mathbf{x} , à partir de $\phi(\mathbf{x})$ et des moyennes $\bar{\phi}(0), \bar{\phi}(1), \bar{\phi}(4), \bar{\phi}(8)$.

Soumettez les estimations $f(\mathbf{x}_{test})$ calculées par votre algorithme sur les images de test du challenge MNIST-4 sur la plateforme Challenge Data. Quel est votre score ?

2.2 Comprendre les erreurs avec une caractéristique

Pour comprendre d'où vient ce pourcentage d'erreurs, on va calculer les histogrammes des caractéristiques dans les quatre populations $y = 0, 1, 4$ et 8 .

A l'aide de la fonction `hist`, calculez les quatre histogrammes des caractéristiques $\phi(\mathbf{x}_{train})$ des images d'entraînement pour chaque $y \in \{0, 1, 4, 8\}$.

Expliquez pourquoi votre algorithme fait beaucoup d'erreurs en comparant ces quatre histogrammes.

Donnez une condition sur les histogrammes pour que l'erreur puisse être nulle.

En regardant ces histogrammes, pouvez-vous identifier les deux chiffres qui sont les plus difficiles à distinguer et le chiffre le plus facile à reconnaître ?

Pensez-vous qu'il soit possible de beaucoup réduire cette erreur avec une seule caractéristique ? Pourquoi ?

2.3 Deux caractéristiques et leurs moyennes

Pour améliorer le résultat de classification, nous allons calculer deux caractéristiques complémentaires, que l'on met dans un tableau ayant 2 variables : $\phi(\mathbf{x}) = [\phi_1(\mathbf{x}), \phi_2(\mathbf{x})]$. Ces deux caractéristiques doivent nous permettre de distinguer si le chiffre y de l'image \mathbf{x} est un $0, 1, 4$ ou 8 , avec peu d'erreurs.

En regardant les images de chiffres dans la Figure 3, proposez une façon de calculer ces caractéristiques.

Faites un programme qui calcule $\phi(\mathbf{x}) = [\phi_1(\mathbf{x}), \phi_2(\mathbf{x})]$ pour chaque image \mathbf{x} .

Faites un programme qui calcule la moyenne $\bar{\phi}_1(y)$ de $\phi_1(\mathbf{x}_{train})$ et la moyenne $\bar{\phi}_2(y)$ de $\phi_2(\mathbf{x}_{train})$, pour les quatre populations d'images correspondant aux chiffres $y = 0, 1, 4$ et 8 .

Pour chaque y , on peut représenter le tableau $\bar{\phi}(y) = [\bar{\phi}_1(y), \bar{\phi}_2(y)]$ comme un point dans un plan, dont on donne les deux coordonnées dans un repère. Quelle propriété doivent avoir ces points pour pouvoir faire de la classification avec ces caractéristiques ?

2.4 Classification avec deux caractéristiques

On interprète aussi le tableau $\phi(x) = [\phi_1(\mathbf{x}), \phi_2(\mathbf{x})]$ comme un point dont on donne les deux coordonnées dans le repère d'un plan. On rappelle que la distance au carré de deux points de coordonnées $\mathbf{a} = [a_1, a_2]$ et $\mathbf{b} = [b_1, b_2]$ est :

$$d(\mathbf{a}, \mathbf{b})^2 = (a_1 - b_1)^2 + (a_2 - b_2)^2$$

Cherchez et programmez un algorithme qui calcule une estimation $f(\mathbf{x})$ du chiffre représenté par l'image \mathbf{x} , à partir des distances $d(\phi(\mathbf{x}), \bar{\phi}(y))^2$ pour chaque $y \in \{0, 1, 4, 8\}$.

Soumettez les estimations $f(\mathbf{x}_{test})$ calculées par votre algorithme sur les images de test du challenge MNIST-4. Avez-vous amélioré votre score ? Pourquoi ?

2.5 Comprendre les erreurs avec deux caractéristiques

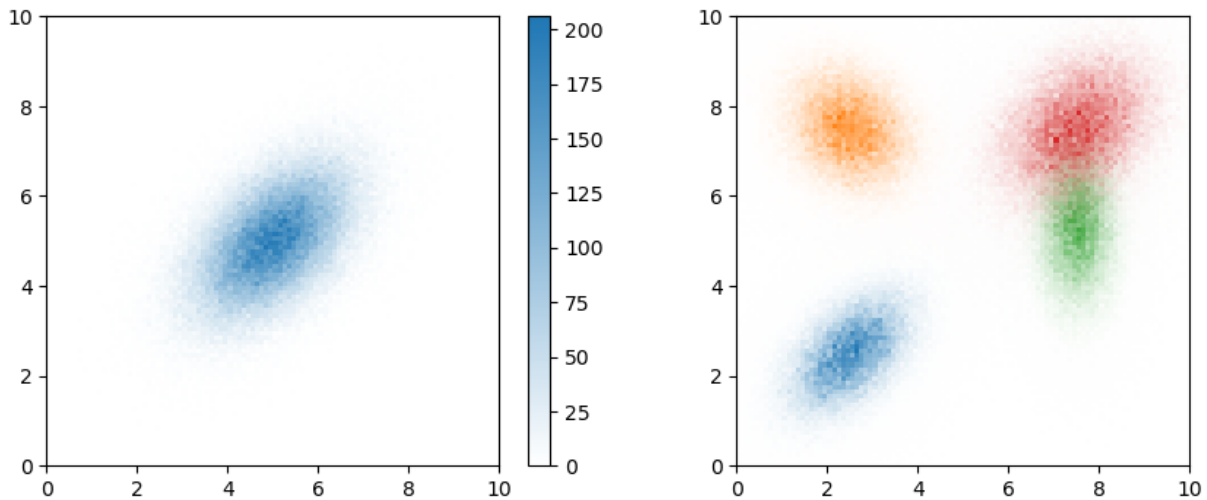
On veut à nouveau comprendre d'où viennent les erreurs en regardant les histogrammes en deux dimensions des caractéristiques dans chaque population. Chaque $[\phi_1(\mathbf{x}), \phi_2(\mathbf{x})]$ est représenté par un point dans un plan. On quadrille le plan avec des petits carrés (*bins* en anglais). Un histogramme en deux dimensions donne le nombre de caractéristiques $\phi(\mathbf{x}) = [\phi_1(\mathbf{x}), \phi_2(\mathbf{x})]$ dans chaque carré du quadrillage, comme l'explique la Figure 4a. La fonction `hist2d` de la bibliothèque `matplotlib` dont le fonctionnement est illustré dans le notebook d'utilitaires associé au challenge permet de calculer et de visualiser sous forme d'image l'historgramme en deux dimensions associé aux caractéristiques $\phi(\mathbf{x}) = [\phi_1(\mathbf{x}), \phi_2(\mathbf{x})]$, pour chaque population d'images de chiffres. La Figure 4b donne un exemple de superposition d'histogrammes en deux dimensions associés à des populations différentes.

Utilisez la fonction `hist2d` pour visualiser les histogrammes de vos caractéristiques, sur les quatre populations d'images de chiffres \mathbf{x}_{train} pour lesquelles $y = 0, 1, 4$ ou 8 .

Donnez une condition sur ces histogrammes pour que votre algorithme ne fasse aucune erreur.

En regardant ces histogrammes, pouvez-vous identifier le chiffre le plus difficile à distinguer et le chiffre le plus facile à reconnaître ?

Comment pourriez vous améliorer votre algorithme en rajoutant des caractéristiques ?



(a) Histogramme d'une population de 100 000 points du plan

(b) Histogramme de quatre populations de 100 000 points du plan

Figure 4: L'historgramme de $\phi(\mathbf{x}) = [\phi_1(\mathbf{x}), \phi_2(\mathbf{x})]$ donne sur une population de \mathbf{x} le nombre de $\phi(\mathbf{x})$ dans chaque petit carré du plan. La Figure (a) montre à titre d'exemple illustratif l'historgramme d'une population de points du plan, sous la forme d'une image. La couleur de chaque petit carré de l'historgramme correspond au nombre de points de la population qui se trouvent dans le carré. Plus ce nombre est élevé, plus le bleu est intense, comme indiqué par l'échelle des couleurs sur la droite. La Figure (b) montre quatre histogrammes superposés sur une même image avec des couleurs différentes de quatre populations différentes.

3 MNIST-10 : Classifier des chiffres de 0 à 9

Le Challenge MNIST-10 propose de reconnaître les dix chiffres $y \in \{0, \dots, 9\}$ qui apparaissent dans les images \mathbf{x} de la base de données MNIST. La Figure 5 montre des exemples d'images de chiffres de 0 à 9 de MNIST.

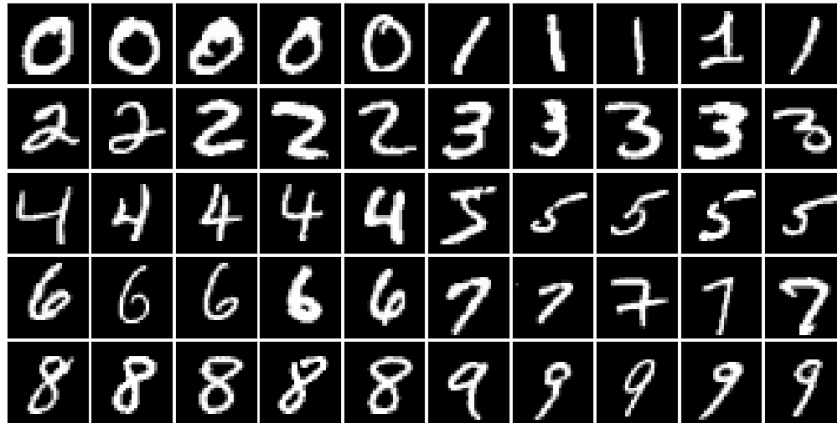


Figure 5: Exemples d'images de chiffres de 0 à 9 dans la base de données d'entraînement de MNIST.

3.1 Classification avec deux caractéristiques

En utilisant deux caractéristiques $\phi(\mathbf{x}) = [\phi_1(\mathbf{x}), \phi_2(\mathbf{x})]$, calculez la moyenne $\bar{\phi}(y)$ des $\phi(\mathbf{x}_{train})$ pour chaque population d'images d'entraînement \mathbf{x}_{train} correspondant à chaque $y \in \{0, \dots, 9\}$.

Utilisez l'algorithme de la question 2.4 pour estimer le chiffre y d'une image \mathbf{x} . Soumettez les estimations $f(\mathbf{x}_{test})$ calculées par votre algorithme sur les images de test \mathbf{x}_{test} de MNIST. Quel est votre score ?

3.2 Quatre caractéristiques et leurs moyennes

Le problème étant plus difficile, on va améliorer l'algorithme en augmentant le nombre de caractéristiques. Proposez quatre caractéristiques différentes et complémentaires : $\phi(x) = [\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \phi_3(\mathbf{x}), \phi_4(\mathbf{x})]$ et faites un programme qui calcule ce tableau de caractéristiques.

Calculez le tableau de moyennes $\bar{\phi}(y) = [\bar{\phi}_1(y), \bar{\phi}_2(y), \bar{\phi}_3(y), \bar{\phi}_4(y)]$ de $\phi(x) = [\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \phi_3(\mathbf{x}), \phi_4(\mathbf{x})]$, pour chaque population d'images d'entraînement \mathbf{x}_{train} correspondant à chaque $y \in \{0, \dots, 9\}$.

3.3 Classification avec quatre caractéristiques

La distance au carré entre des tableaux de 4 coordonnées $\mathbf{a} = [a_1, a_2, a_3, a_4]$ et $\mathbf{b} = [b_1, b_2, b_3, b_4]$ est :

$$d(\mathbf{a}, \mathbf{b})^2 = (a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2 + (a_4 - b_4)^2 \quad (2)$$

Utilisez le même algorithme qu'à la question 2.4 pour calculer une estimation $f(\mathbf{x})$ du chiffre représenté par l'image \mathbf{x} , à partir des distances $d(\phi(\mathbf{x}), \bar{\phi}(y))^2$ pour chaque $y \in \{0, \dots, 9\}$.

Soumettez les estimations $f(\mathbf{x}_{test})$ calculées par votre algorithme sur les images de test \mathbf{x}_{test} de MNIST pour obtenir votre score.

3.4 Comprendre les erreurs avec quatre caractéristiques

La distance relatives de vos caractéristiques moyennes $\bar{\phi}(y)$ et $\bar{\phi}(y')$ pour des paires de classes (y, y') est définie par :

$$\Delta(y, y') = \frac{d(\bar{\phi}(y), \bar{\phi}(y'))^2}{d(\bar{\phi}(y), 0)^2 + d(\bar{\phi}(y'), 0)^2}, \quad (3)$$

Montrez que si $\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \phi_3(\mathbf{x}), \phi_4(\mathbf{x})$ sont positifs, alors $\Delta(y, y') \leq 1$.

Faites un programme qui calcule $\Delta(y, y')$ pour les différentes paires de chiffres, et identifiez les deux chiffres que vos caractéristiques vont le plus confondre.

Comment pouvez-vous améliorer votre algorithme et réduire l'erreur de classification ?

3.5 Chiffres moyens

On propose maintenant d'utiliser comme tableau de caractéristiques l'image elle-même : $\phi(\mathbf{x}) = \mathbf{x}$. Ce tableau de caractéristiques a donc $28^2 = 784$ entiers.

Calculez le tableau des caractéristiques moyennes $\bar{\phi}(y)$ pour les 10 populations d'images \mathbf{x}_{train} correspondant à chaque chiffre $y \in \{0, \dots, 9\}$.

Chaque $\bar{\phi}(y)$ est un tableau de 28×28 pixels qui correspond à l'image moyenne de chaque chiffre y sur la base de données d'entraînement : on appellera "chiffre moyen" une telle image. Utilisez la fonction `imshow` de la bibliothèque `matplotlib` dont le fonctionnement est illustré dans le notebook d'utilitaires associé au challenge pour visualiser chaque $\bar{\phi}(y)$ comme une image.

À quoi ressemblent ces 10 images ?

3.6 Classification par chiffre moyen

Généralisez la distance (2) pour définir une distance $d(\mathbf{a}, \mathbf{b})^2$ entre deux tableaux $\mathbf{a}[i, j]$ et $\mathbf{b}[i, j]$ de 28×28 variables.

Utilisez le même algorithme qu'à la question 3.3 pour calculer une estimation $f(\mathbf{x})$ du chiffre représenté par \mathbf{x} , à partir des distances $d(\phi(\mathbf{x}), \bar{\phi}(y))^2$ pour chaque $y \in \{0, \dots, 9\}$.

Soumettez les estimations $f(\mathbf{x}_{test})$ calculées par votre algorithme sur les images de test de MNIST pour obtenir votre score. Comparez votre score avec celui obtenu avec quatre caractéristiques.

3.7 Chiffres non centrés

Imaginez que la position des chiffres ne soit plus centrée mais qu'elle soit arbitraire dans l'image. À quoi ressembleront les chiffres moyens $\bar{\phi}(y)$?

L'algorithme de classification par chiffre moyen de la question 3.6 précédente fonctionnera-t-il bien ?

Montrez qu'on obtient le même score de classification avec une seule caractéristique $\phi(\mathbf{x})$ égale à la somme des pixels de l'image. Donnez le score que l'on obtient sur MNIST avec cette caractéristique.

Proposez une deuxième caractéristique qui puisse améliorer le score lorsque les chiffres sont non centrés.