Rapport Projet ML: Réseau de neurones

AUTEURS:

AKNOUCHE ANIS
HADDADI HACENE

Année Universitaire: 2020 / 2021

Sommaire

Partie 1: Module Linéaire

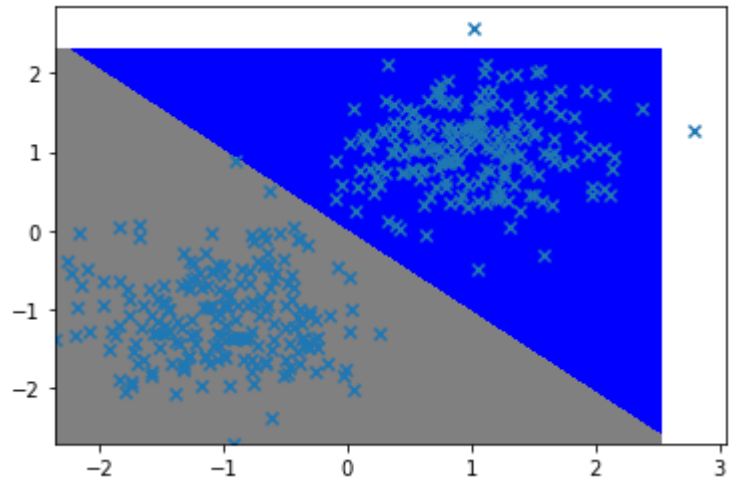

Figure 1 : Frontière de décision de données linéaires appartenant à deux classes (-1 , +1).

Utilisation d'un module linéaire de dimension d'entrées =2 et dimensions de sortie =1 avec une MSELoss (Mean Squared Error Loss)

Linear(2,1)

Remarque :

- La frontière de décision est très claire, on a une bonne séparation des données (cas linéaire).
- Obtention d'une précision en entraînement et test de 99% .

Partie 2: Module Non Linéaire

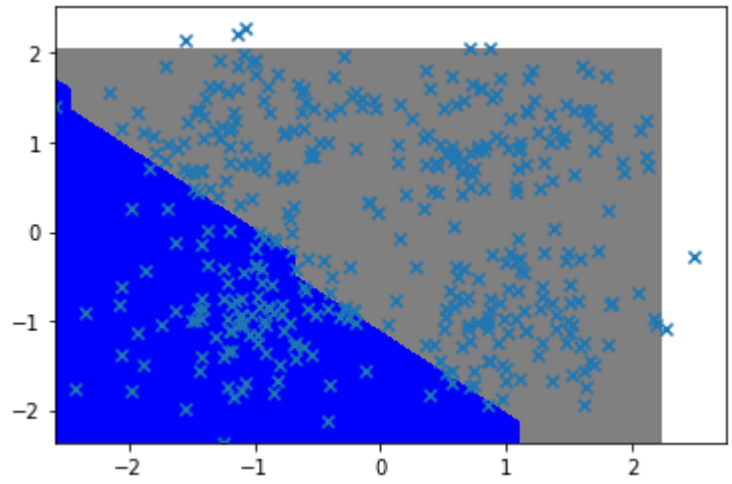

Figure 2 : Frontière de décision de données non linéaires.

Utilisation d'un module linéaire de dimension d'entrées =2 et dimensions de sortie =1 avec une fonction d'activation Non Linéaire Tanh avec une MSELoss (Mean Squared Error Loss)

Linear(2,1) -> TanH()

Remarque :

- La frontière de décision est moins précise que dans le cas linéaire.
- Obtention d'une précision en entraînement de 68% et une précision en test de 70%

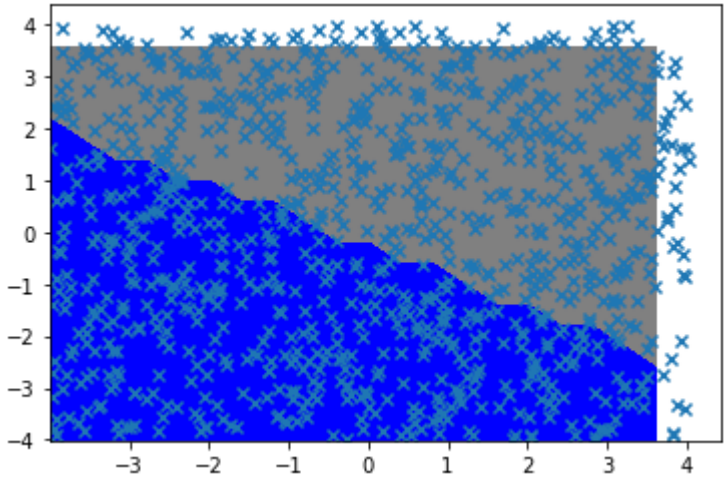

Figure 3 : Frontière de décision de données XOR.

Utilisation d'un réseau qui se compose d'un module Linéaire suivie d'une fonction d'activation TanH puis d'un autre module linéaire suivi d'une fonction d'activation TanH avec une MSELoss (Mean Squared Error Loss)

Linéaire(2,1) -> TanH() -> Linéaire(1,1) -> TanH()

Remarque :

- Obtention d'une précision en entraînement de 54% et d'une précision en test de 52%.

Partie 3: Encapsulation

Implémentation des classes : Sequentiel, Optim, SGD avec leur test.

Partie 4: Module Multiclasse

Utilisation d'un réseau qui se compose d'un module linéaire et d'une fonction d'activation softMax stabilisée (Normalisée):

Linear(256, 10) -> Sofmax_stable() et utilisation d'une fonction de coût cross-entropy

- Pour uniquement 2 classes de chiffres :

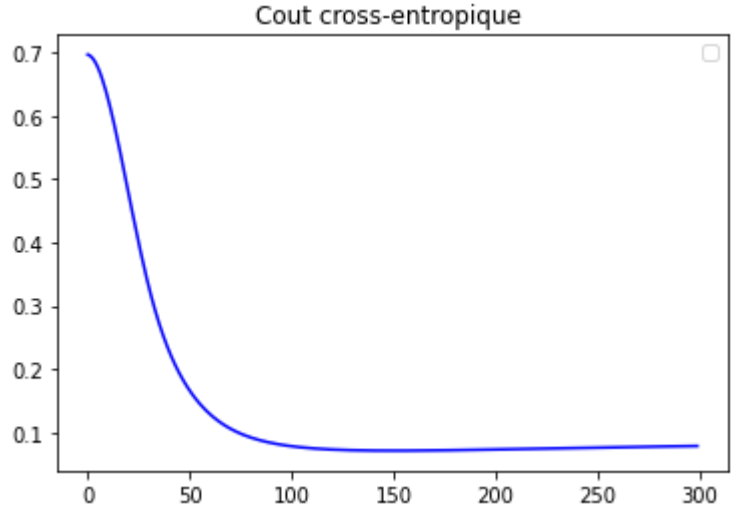

Figure 4 : Évolution du coût cross-entropique selon le nombre d'itérations de la descente de gradient.

Train accuracy = 0.978688524590164
Test accuracy = 0.9696969696969697

Remarque :

- Plus le nombre d'itérations de la descente de gradient augmente, plus le coût de la loss diminue ce qui signifie que le réseau apprend bien et converge.
- Obtention d'une précision en entraînement de 97% et d'une précision en test de 96%

- Pour les 10 classes de chiffres :

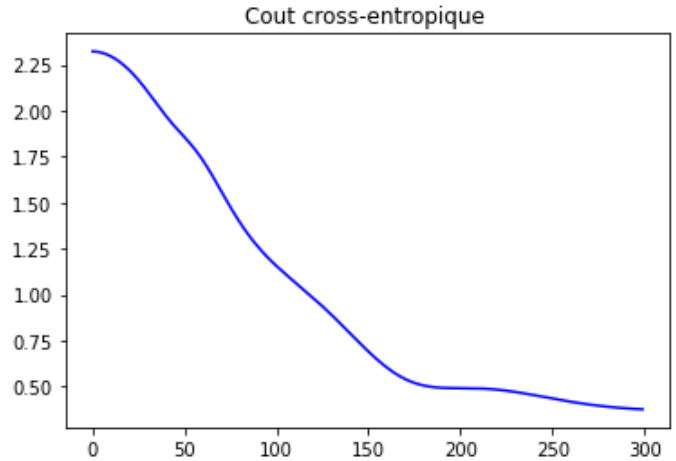

Figure 5 : Évolution du coût cross-entropique selon le nombre d'itérations de la descente de gradient.

Train accuracy = 0.8871211082156083
Test accuracy = 0.8390632785251619

Remarque :

- On remarque qu'à partir de 150 itérations en descente de gradient on obtient une loss inférieure à 0.5, le réseau converge.
- Obtention d'une précision en entraînement de 88% et une précision en test de 83%

Figure 6 : Matrice de Confusion.

Remarque :
On remarque que notre réseau se trompe très peu.
Certaines classes partagent quelques caractéristiques :

- Les classes 4 et 9
- Les classes 3 et 8
- Les classes 0 et 6
- Les classes 5 et 6
- Les classes 8 et 9
- Les classe 2 et 8
- les classe 7 et 8
- Les classe 5 et 3 et 2 et 4

Partie 5: Auto-Encodeur

Visualisation de quelques images après compression:

Avec le réseau suivant : (Espace Latent de taille 100)

Encodeur		Décodeur
Linear(256,100) -> TanH()	->	Linear(100,256) -> TanH()

avec :

- Initialisation des paramètres selon l'initialisation de Xavier
- Utilisation d'une cross-entropie binaire BCELoss

Les données apprises:

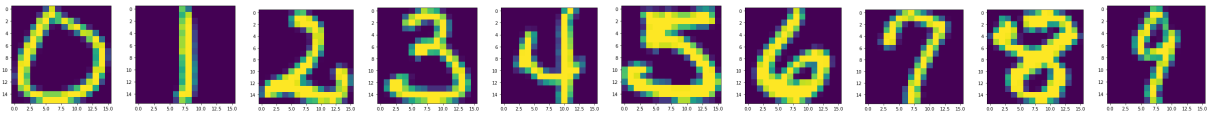

Figure 7 : Représentation de chiffres manuscrits.

Compression avec l'encodeur (Projection dans un espace latent de taille 100):

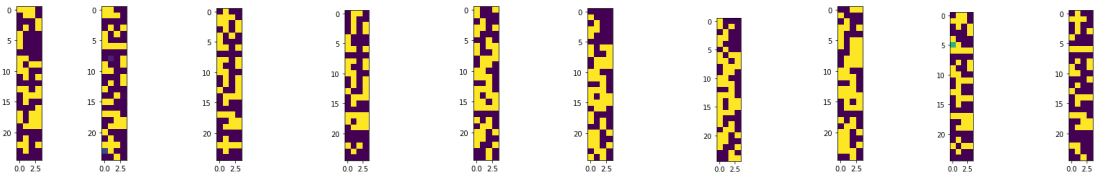

Figure 8 : Représentation des données compressées.

Décompression avec le décodeur vers l'espace d'origine:

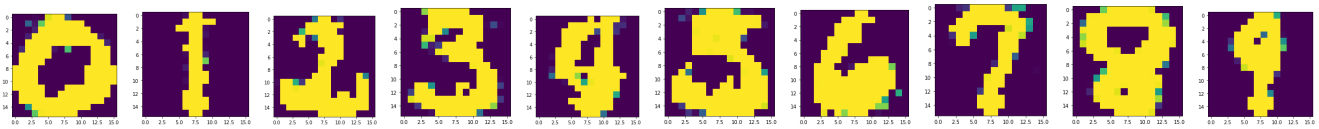

Figure 9 : Représentation des données décompressées.

Avec le réseau suivant : (Espace Latent de taille 10)

Encodeur		Décodeur
Linear(256,10) -> TanH()	->	Linear(10,256) -> TanH()

avec :

- Initialisation des paramètres selon l'initialisation de Xavier
- Utilisation d'une cross-entropie binaire BCELoss

Les données apprises:

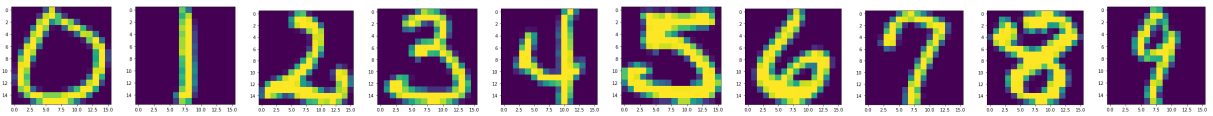

Figure 10 : Représentation de chiffres manuscrits.

Compression avec l'encodeur (Projection dans un espace latent de taille 10):

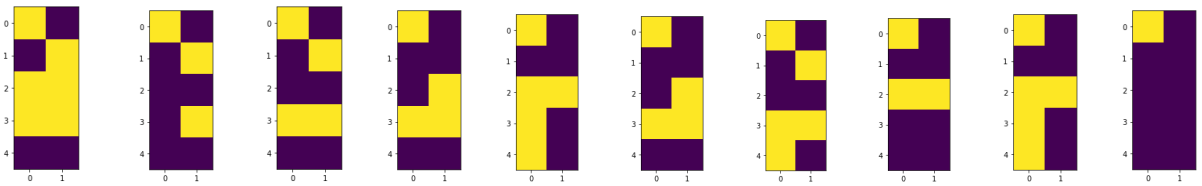

Figure 11 : Représentation des données compressées.

Décompression avec le décodeur vers l'espace d'origine :

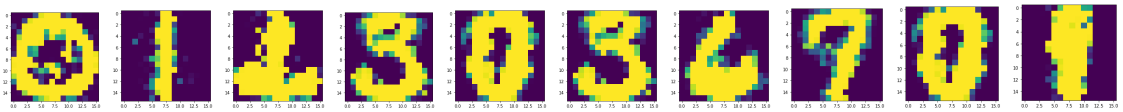

Figure 12 : Représentation des données décompressées.

Remarque:

- Plus l'espace latent est réduit, plus la précision diminue lors de la décompression de l'image à cause de la forte compression qui fait perdre de l'information.

Visualisation des données à l'aide d'une PCA dans l'espace d'origine :

Explained variation per principal component: [0.17884424 0.0896704 0.06571727]

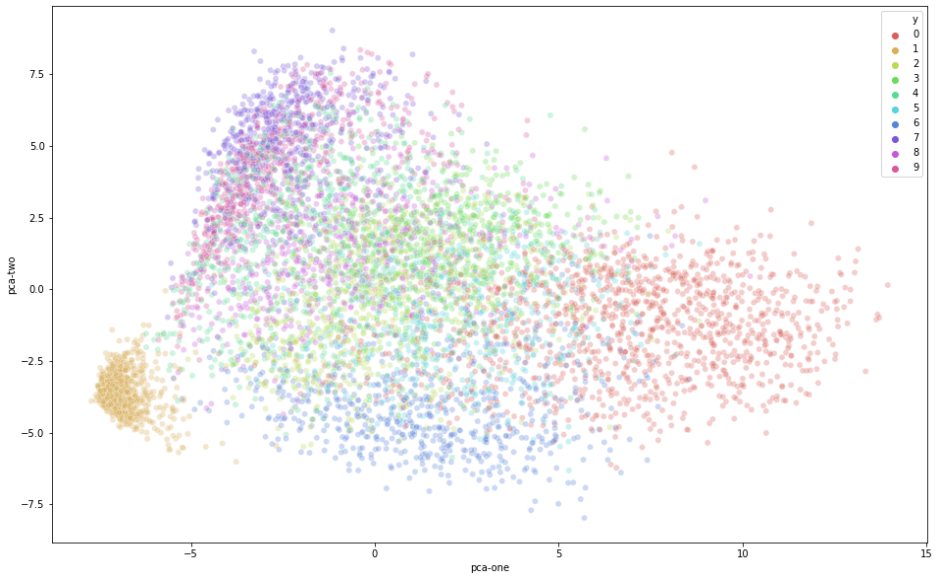

Figure 13 : Visualisation des données dans l'espace d'origine grâce aux composantes principales (2D).

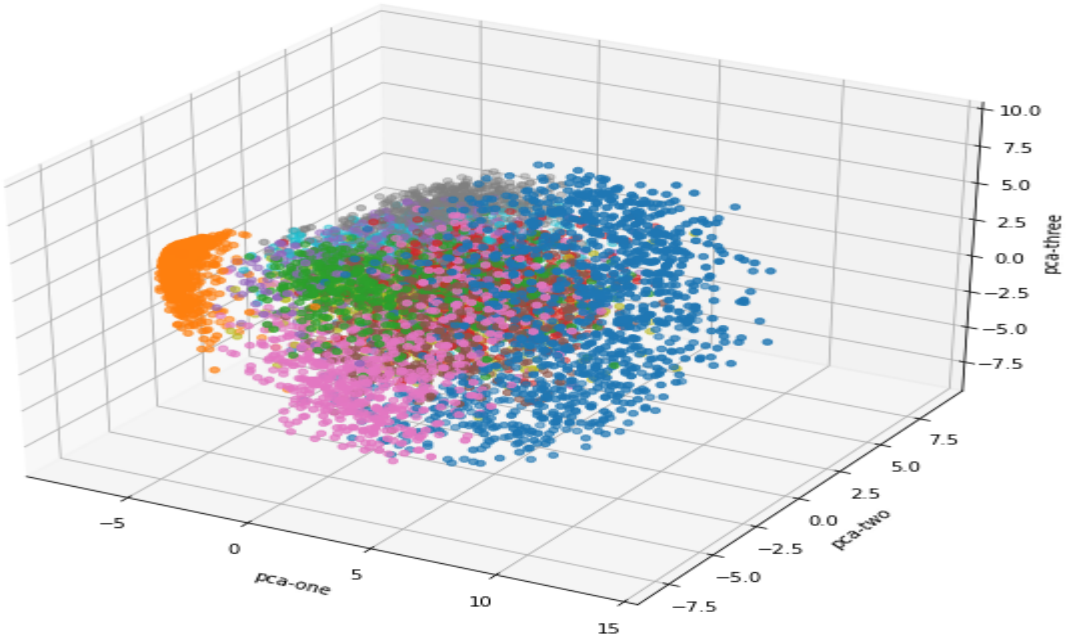

Figure 14 : Visualisation des données dans l'espace d'origine grâce aux composantes principales (3D).

Remarque:

- On peut remarquer que la classe 1 se distingue bien des autres classes.
- La classe 6 et la classe 8 ne se chevauchent pas trop.
- Les classes 2, 3, 4 et 5 se chevauchent beaucoup.

Visualisation des données à l'aide d'une PCA dans l'espace projeté de taille 100 :

Explained variation per principal component: [0.24684883 0.13735504 0.09038577]

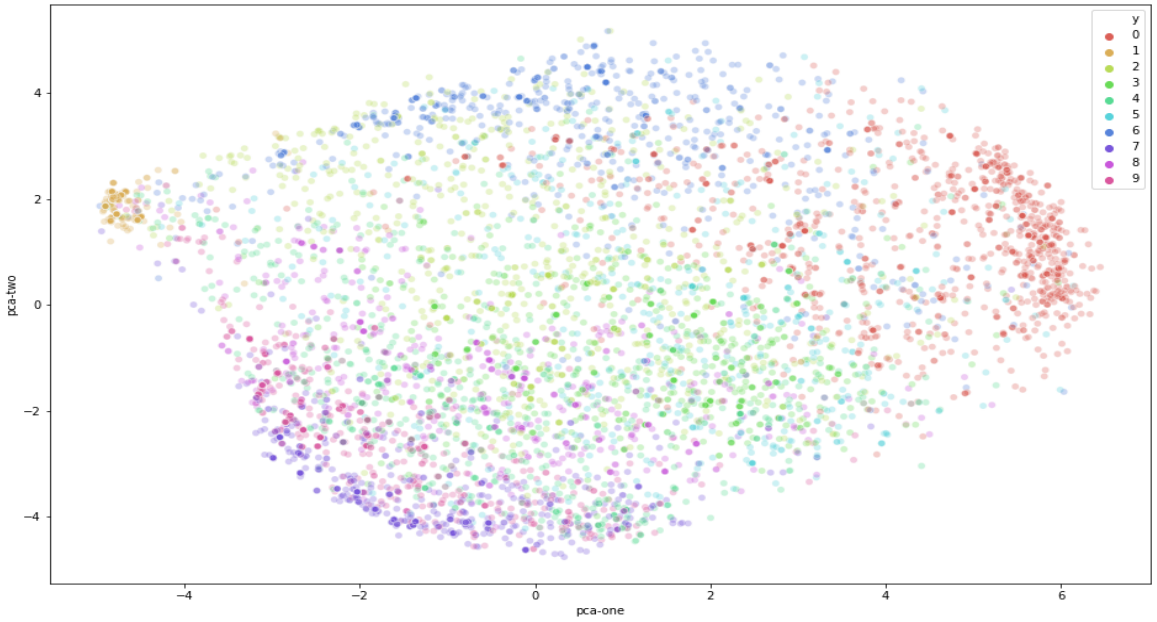

Figure 15 :Visualisation des données dans l'espace projeté grâce aux composantes principales (2D).

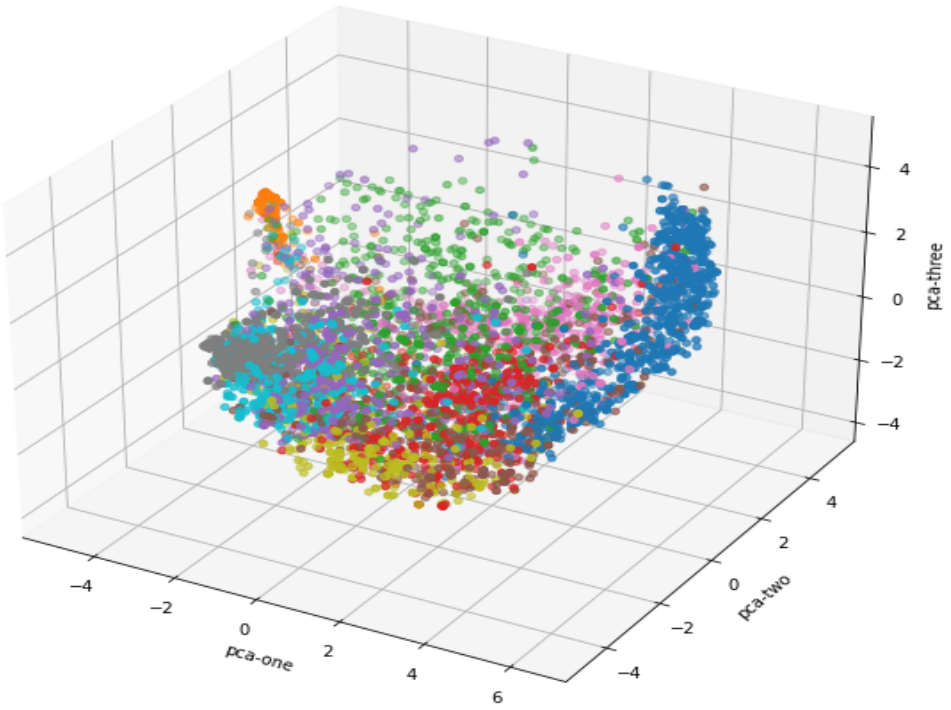

Figure 16 :Visualisation des données dans l'espace projeté grâce aux composantes principales (3D).

Remarque :

- On retrouve les mêmes informations sur les différents clusters de classe de chiffre qu'avec l'espace d'origine.
- Avec un espace latent de taille 100, la perte d'information est faible, grâce à la faible compression.

Visualisation des données à l'aide d'une PCA dans l'espace projeté de taille 10 :

Explained variation per principal component: [0.24684883 0.13735504 0.09038577]

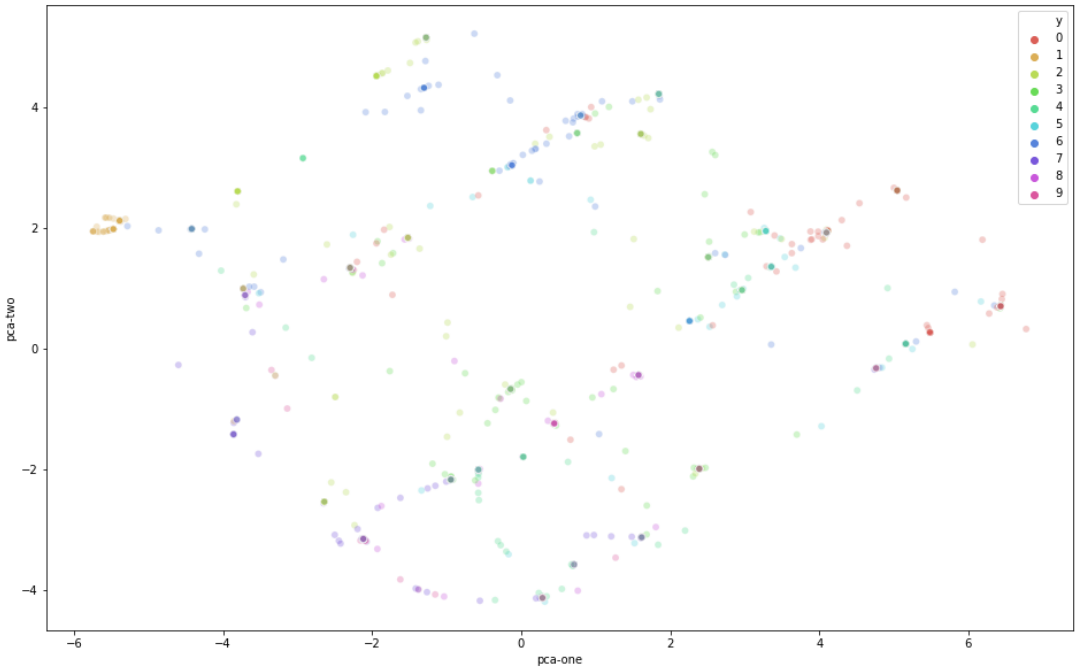

Figure 15 :Visualisation des données dans l'espace projeté grâce aux composantes principales (2D).

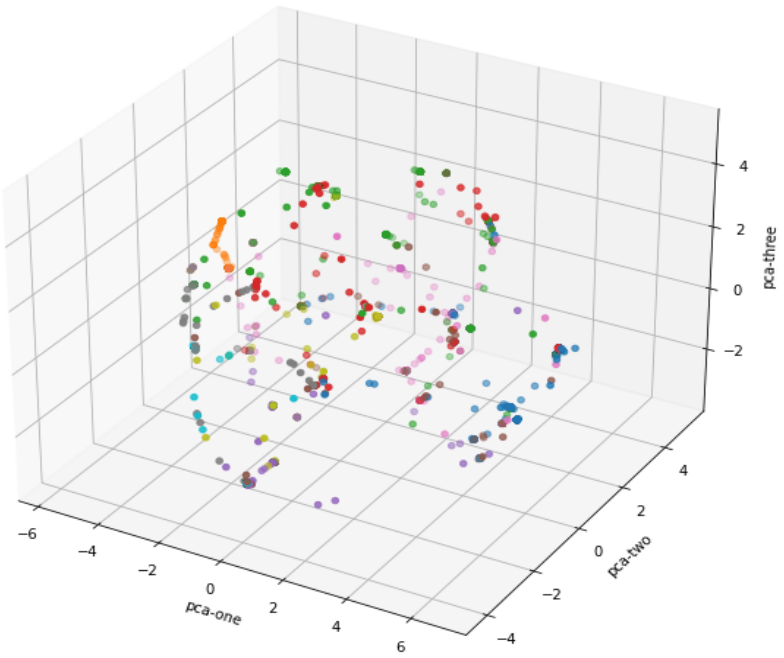

Figure 16 :Visualisation des données dans l'espace projeté grâce aux composantes principales (3D).

Remarque :

- On ne retrouve plus les mêmes informations sur les différents clusters de classe de chiffre qu'avec l'espace d'origine.
- Avec un espace latent de taille 10, la perte d'information est très forte à cause de la forte compression.