FTML practical session 10: 2023/06/01

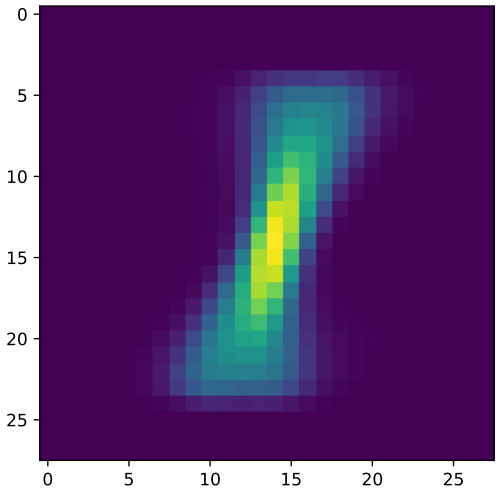

TABLE DES MATIÈRES

INTRODUCTION

1 APPLICATION OF UNSUPERVISED LEARNING TO CLASSIFICATION

In this section, we build classifiers based on an unsupervised preprocessing of the data.

1.1 Meteorological data : dimensionality reduction and visualization

A meteorological station has gathered 1600 data samples in dimension 6, thanks to 6 sensors. The operators of the station would like to predict the risk of a tempest the next day, but first, they need to reduce the dimensionality of the data, in order to apply a supervised learning algorithm on the reduced data.

The data are stored in the **data** folder.

Find a dimensionality reduction method and a dimension (2 or 3), that seems to allow to predict the label based on the projected components only, first by making scatter plots of the projected data. Verify this by training a classifier that learns to predict the labels based on the projections only.

https://scikit-learn.org/stable/modules/unsupervised_reduction.html

For a discussion on nonlinear dimensionality reduction, see section 2.

1.2 Application of vector quantization to classification

We will apply **vector quantization** to the MNIST classification problem. Namely, we will compute average representers of each class, and for a new sample, predict the class of the nearest representer among the classes.

https://en.wikipedia.org/wiki/Vector_quantization

You can fetch the data using vector_quantization/fetch_data.py.

A **Gaussian blur** of the input digits (see figures 3, 4, 5) might slightly improve the classification performance.

Implement a classification method based on vector quantization in order to classify the MNIST dataset, and use a hyperparameter optimizatino method in order to find a relevant value of the σ parameter of the GaussianBlur method of OpenCV.

https://docs.opencv.org/4.x/d4/d86/group__imgproc__filter.html#gaabe8c836e97159a9193fb0b11a

For this exercise, a decomposition of the dataset into a train/validation/test might be enough, but you can also use a cross validation.

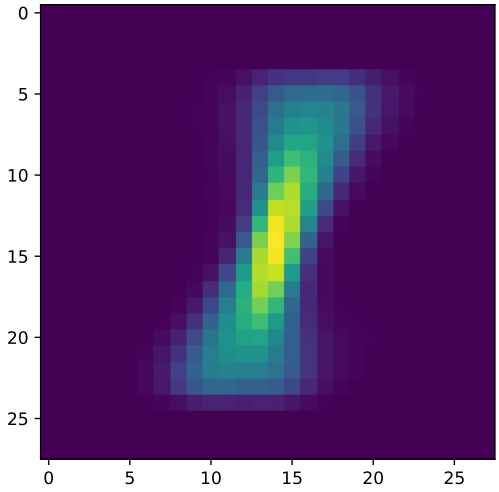

FIGURE 1 – Average of the 1 class

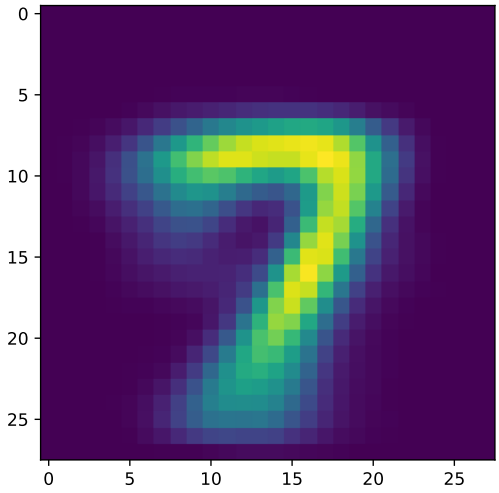

FIGURE 2 – Average of the 7 class

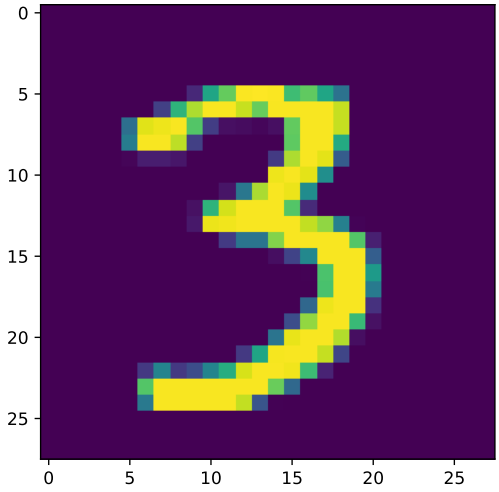

FIGURE 3 – A digit from the dataset.

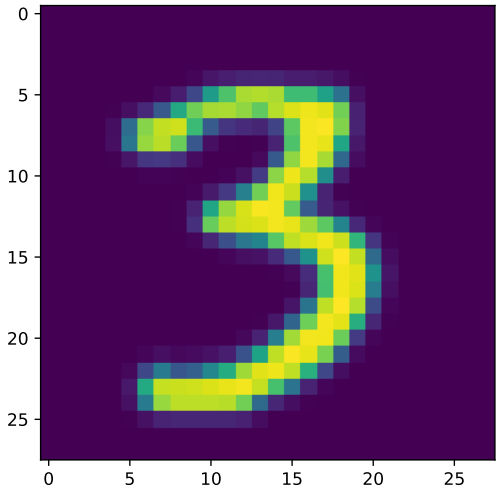

FIGURE 4 – The same digit, blurred with $\sigma = 0.5$

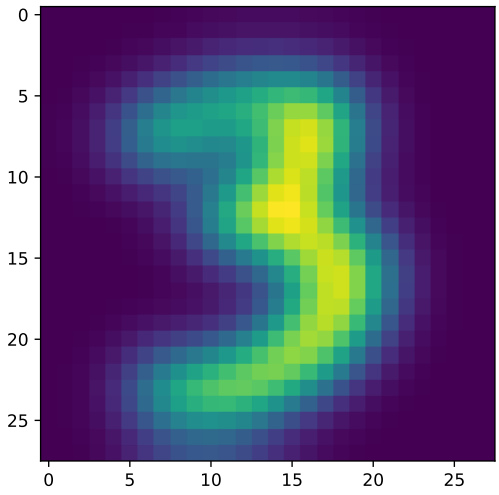

FIGURE 5 – The same digit, blurred with $\sigma = 2$

2 MANIFOLDS

In a previous session we saw an example of data that were lying in a linear subspace of dimension 3, inside $\mathbb{R}^{30}$. We could test this with a PCA.

However, in some situations the data also lie on subsets of $\mathbb{R}^d$, that are not linear subspace but also have a lower dimension, in a sense that is slightly different than the linear case. This subsets are called **manifolds** (variété, sous-variété) and the process of looking for such manifolds is called nonlinear dimensionality reduction or manifold learning.

https://en.wikipedia.org/wiki/Nonlinear_dimensionality_reduction#Locally-linear_embedding

To properly define the dimension of a manifold, some mathematical background is necessary, however the intuition is that locally, this subsets can be perfectly described by a number p of parameters, that corresponds to this dimension.

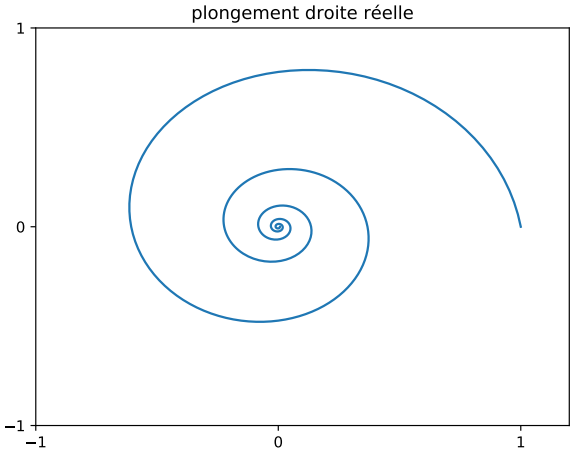

FIGURE 6 – 1 dimensional manifold, in $\mathbb{R}^2$. Locally, it is possible to parametrize this spiral with 1 number.

The goal of manifold learning is to project the data in a smaller subspace trying to preserve information on the dataset. Then, an algorithm might run more easily on the projected data.

You can find resources on this topic in these pages

https://scikit-learn.org/stable/modules/manifold.html

https://scikit-learn.org/stable/auto_examples/manifold/plot_compare_methods.html

In figure 7, you can find an example of manifold learning of the MNIST dataset with the t SNE method. The data are initially in $\mathbb{R}^{28\times 28}$.

https://lvdmaaten.github.io/tsne/

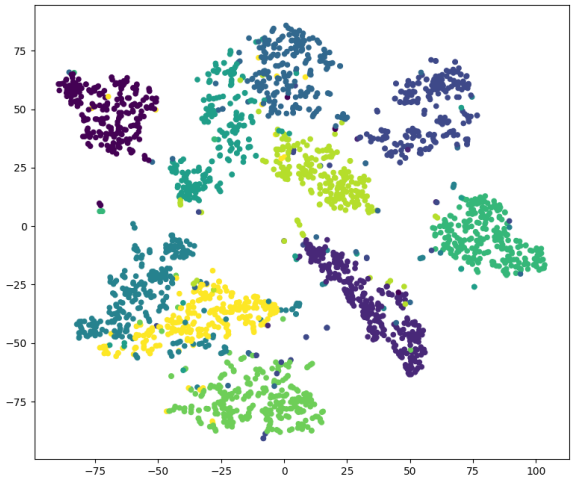

FIGURE 7 – Projection of the MNIST data to $\mathbb{R}^2$ with t-SNE. The colors correspond to the class of the sample, but this is just for visualization : the class is unknown by the dimensionality reduction method.

Some disadvantages of non linear manifold include the following facts :

— It is hard to determine a good output dimension (whereas in PCA we can use explained variance) and it is hard to interpret the embedded dimensions (whereas in PCA we know what they mean).

— They depend on the number of neighbors chosen (if relevant, as several of them involve a nearest neighbor search, often in the geodesic sense)

— They are often computationally slower.

You can experiment with scikit and try to reduce the dimension of some datasets with a nonlinear method.