

Résumé

29 Juin
(Ajouté au github online)

class_names_2.csv

```
1 id_species,species_name,genus,Images,genus_images
2 1363227,Daucus carota,Daucus,7208,7227
3 1392475,Alliaria petiolata,Alliaria,6337,6857
4 1356022,Hypericum perforatum,Hypericum,6140,11946
5 1364099,Centranthus ruber,Centranthus,5334,5607
6 1355937,Cirsium vulgare,Cirsium,5178,12075
7 1359517,Trifolium pratense,Trifolium,5063,14622
```

class_names_2.csv

Informations sur les plantes



Train

Ensemble de données

user_paths *



Pl@ntNet_main_menu_window.py

Programme principal

functions.py

Sous-programme

Fonctions diverses

- user_paths *
- draw_text
- possibility_to_return_to_menu
- list_of_groups
- id_species_per_group
- decipher_coord

user_paths *



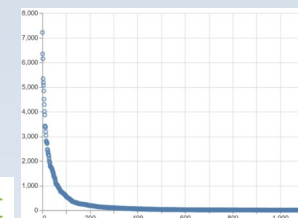
Pygames_elements

Dossier présent dans le dossier
Github *Projet-L3*

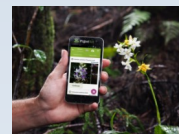
images



Pl@ntNet



| Ce PC > Documents > train | | |
|---------------------------|------------------|---------------------|
| Nom | Modifié le | Type |
| 1355868 | 15/06/2021 06:40 | Dossier de fichiers |
| 1355920 | 15/06/2021 05:03 | Dossier de fichiers |
| 1355932 | 15/06/2021 06:40 | Dossier de fichiers |
| 1355936 | 15/06/2021 06:40 | Dossier de fichiers |
| 1355937 | 15/06/2021 06:41 | Dossier de fichiers |
| 1355955 | 15/06/2021 06:02 | Dossier de fichiers |
| 1355959 | 15/06/2021 06:41 | Dossier de fichiers |



Installation et fonctionnement des programmes

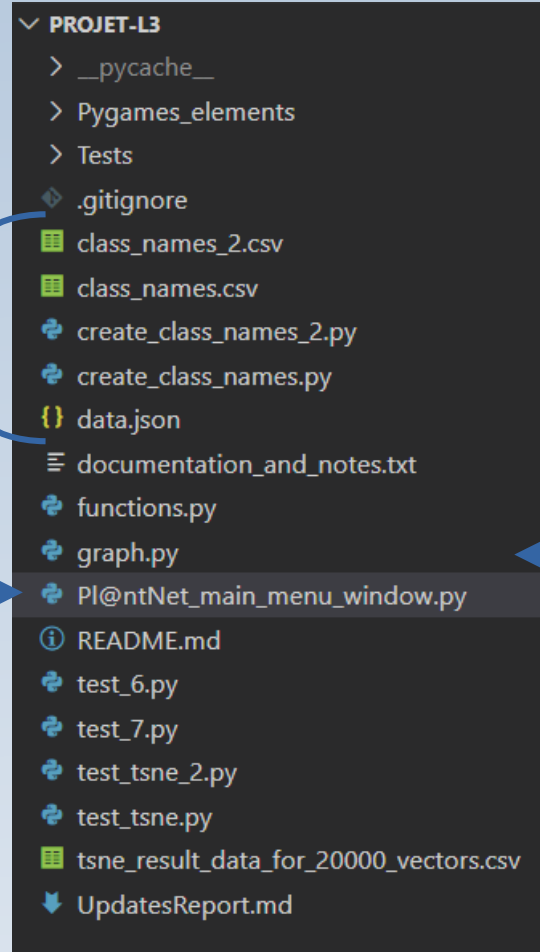
(Plus de détails dans le mail du 29 Mai)

- Télécharger notre Github.
(<https://github.com/GFouilhe-AThirit/Projet-L3>)
- Installer Pygame (`pip install Pygame`).
- Renseigner dans le fichier *functions.py* les chemins *path_to_train* et *path_to_folder* menant respectivement au dossier "*Train*" et à notre répertoire github téléchargé.

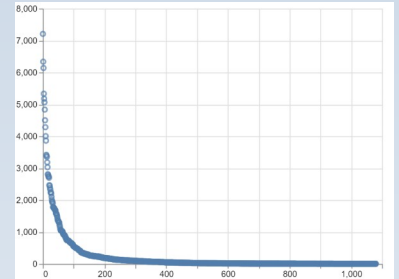
Programmes

Création de
class_names_2.csv

Programme principal



Programme qui affiche le
graphique dont l'image a été
sauvegardée en *png* dans
Pygames_elements



```
{ data.json > ...
```

```
1 { "1355868": "Lactuca virosa L.", "1355920":  
  "Pelargonium capitatum (L.) L'Hér.",  
  "1355932": "Pelargonium graveolens L'Hér.",  
  "1355936": "Cirsium arvense (L.) Scop.",  
  "1355937": "Cirsium vulgare (Savi) Ten.",  
  "1355955": "Pelargonium odoratissimum (L.)  
  L'Hér.", "1355959": "Pelargonium peltatum (L.)  
  L'Hér.", "1355961": "Pelargonium  
  quercifolium (L. f.) L'Hér.", "1355978":
```

```
class_names_2.csv
```

```
1 id_species,species_name,genus,Images,genus_images  
2 1363227,Daucus carota,Daucus,7208,7227  
3 1392475,Alliaria petiolata,Alliaria,6337,6857  
4 1356022,Hypericum perforatum,Hypericum,6140,11946  
5 1364099,Centranthus ruber,Centranthus,5334,5607  
6 1355937,Cirsium vulgare,Cirsium,5178,12075  
7 1359517,Trifolium pratense,Trifolium,5063,14622
```

```
class_names.csv
```

```
1 id_species,species_name  
2 1355868,Lactuca virosa L.  
3 1355920,Pelargonium capitatum (L.) L'Hér.  
4 1355932,Pelargonium graveolens L'Hér.  
5 1355936,Cirsium arvense (L.) Scop.  
6 1355937,Cirsium vulgare (Savi) Ten.  
7 1355955,Pelargonium odoratissimum (L.) L'Hér.
```

create_class_names.py

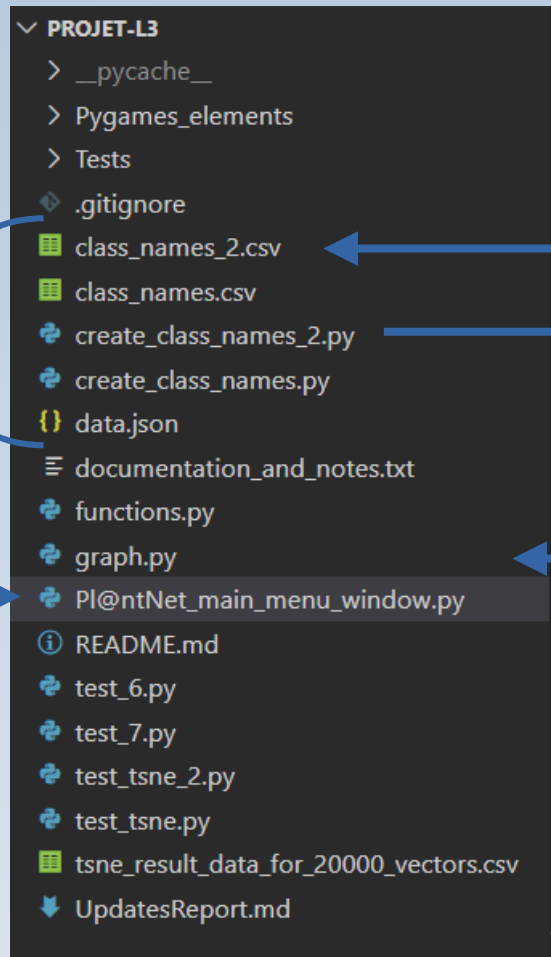
create_class_names_2.py

data.json → class_names.csv → class_names_2.csv

Programmes

Création de
class_names_2.csv

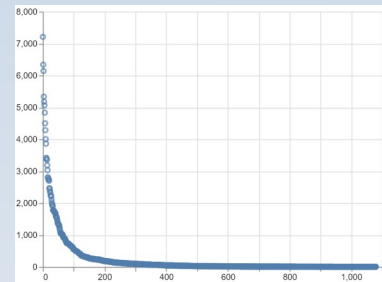
Programme principal



```
class_names.csv
1 id_species,species_name
2 1355868,Lactuca virosa L.
3 1355920,Pelargonium capitatum (L.) L'Hér.
4 1355932,Pelargonium graveolens L'Hér.
5 1355936,Cirsium arvense (L.) Scop.
6 1355937,Cirsium vulgare (Savi) Ten.
7 1355955,Pelargonium odoratissimum (L.) L'Hér.
```

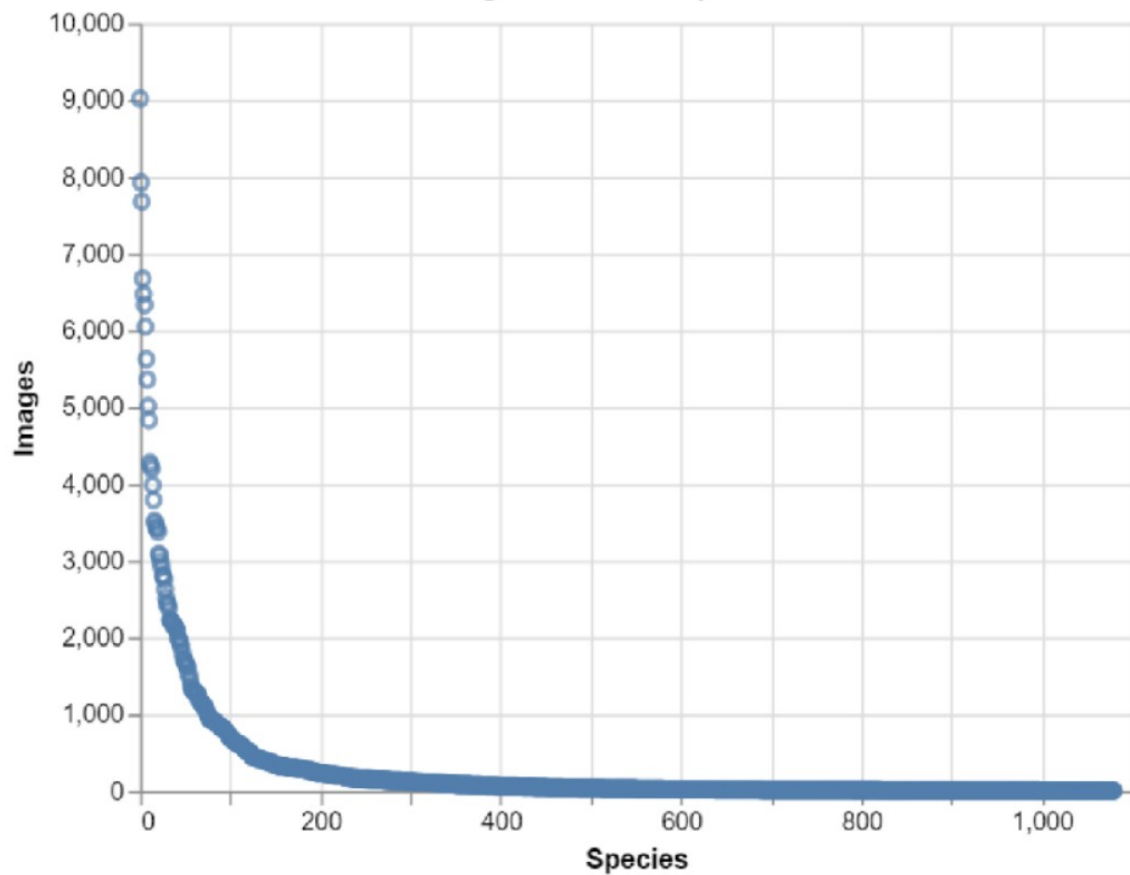
class_names.csv

Programme qui affiche le
graphique dont l'image a été
sauvegardée en *png* dans
Pygames_elements

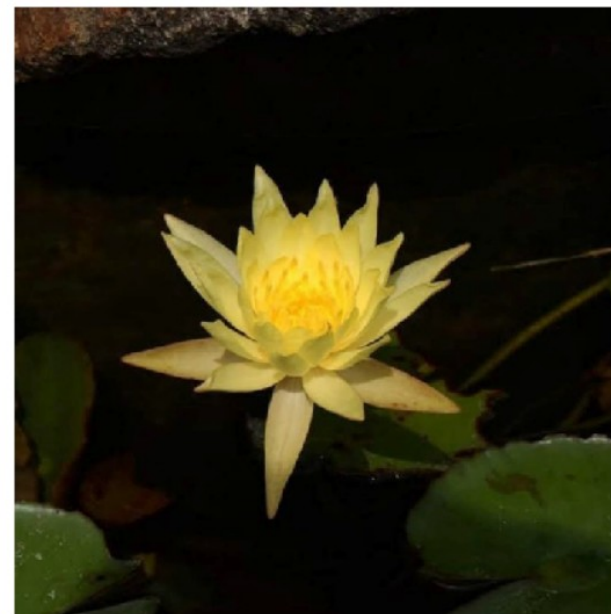
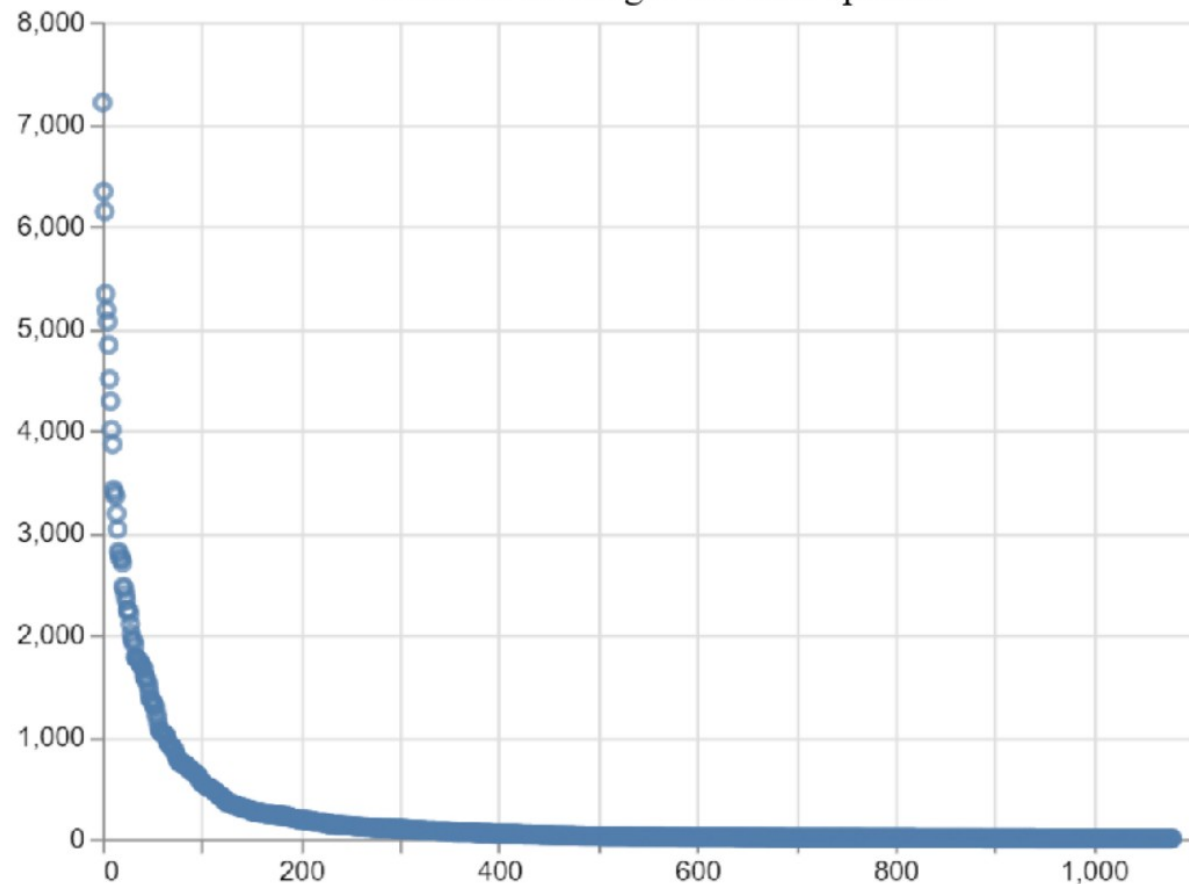




Number of Images for each species in the Dataset



Number of images for each species



Species : *Nymphaea mexicana*
Number of Images : 14

Click on a group name



Pl@ntNet



Groups

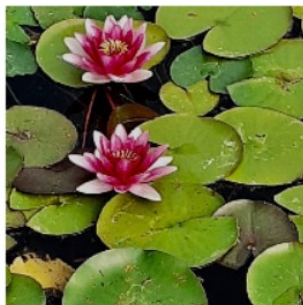
Liriodendron
 Humulus
 Trachelospermum
 Tradescantia
 Pelargonium
 Angelica
 Zamiodulcas
 Perovskia
 Schefflera
 Ophrys
 Cucurbita
 Melilotus
 Helminthotheca
 Mercurialis
 Smilax
 Barbarea
 Fittonia
 Dryopteris
 Pancratium
 Acacia
 Kniphofia
 Epipactis
 Nandina
 Nymphaea
 Chaerophyllum
 Althaea
 Anthericum
 Daphne
 Nephrolepis
 Phalaris
 Hebe
 Dendrobium
 Thapsia
 Lithodora
 Lathraea
 Galega
 Hippophae
 Peperomia
 Metasequoia

Group : Nymphaea.

Number of species : 10



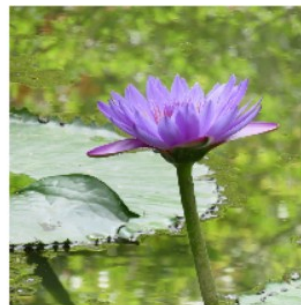
Nymphaea_alba



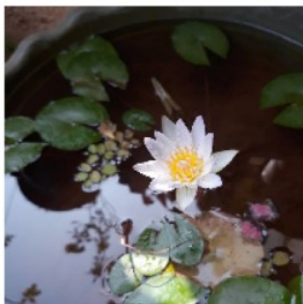
Nymphaea_candida



Nymphaea_odorata



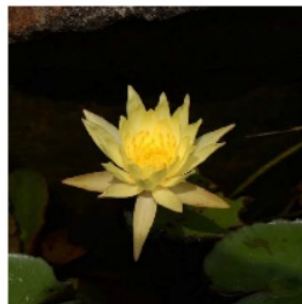
Nymphaea_nouchali



Nymphaea_lotus



Nymphaea_ampla



Nymphaea_mexicana



Nymphaea_tetragona



Genus

ABC



PlantNet



Genus : Acacia.

Number of species : 39.



Acacia dealbata



Acacia retinodes



Acacia longifolia



Acacia mearnsii



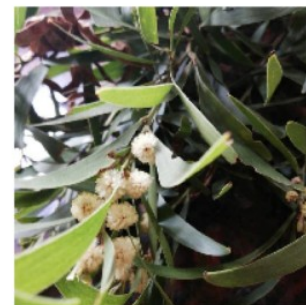
Acacia melanoxylon



Acacia baileyana



Acacia saligna



Acacia pycnantha



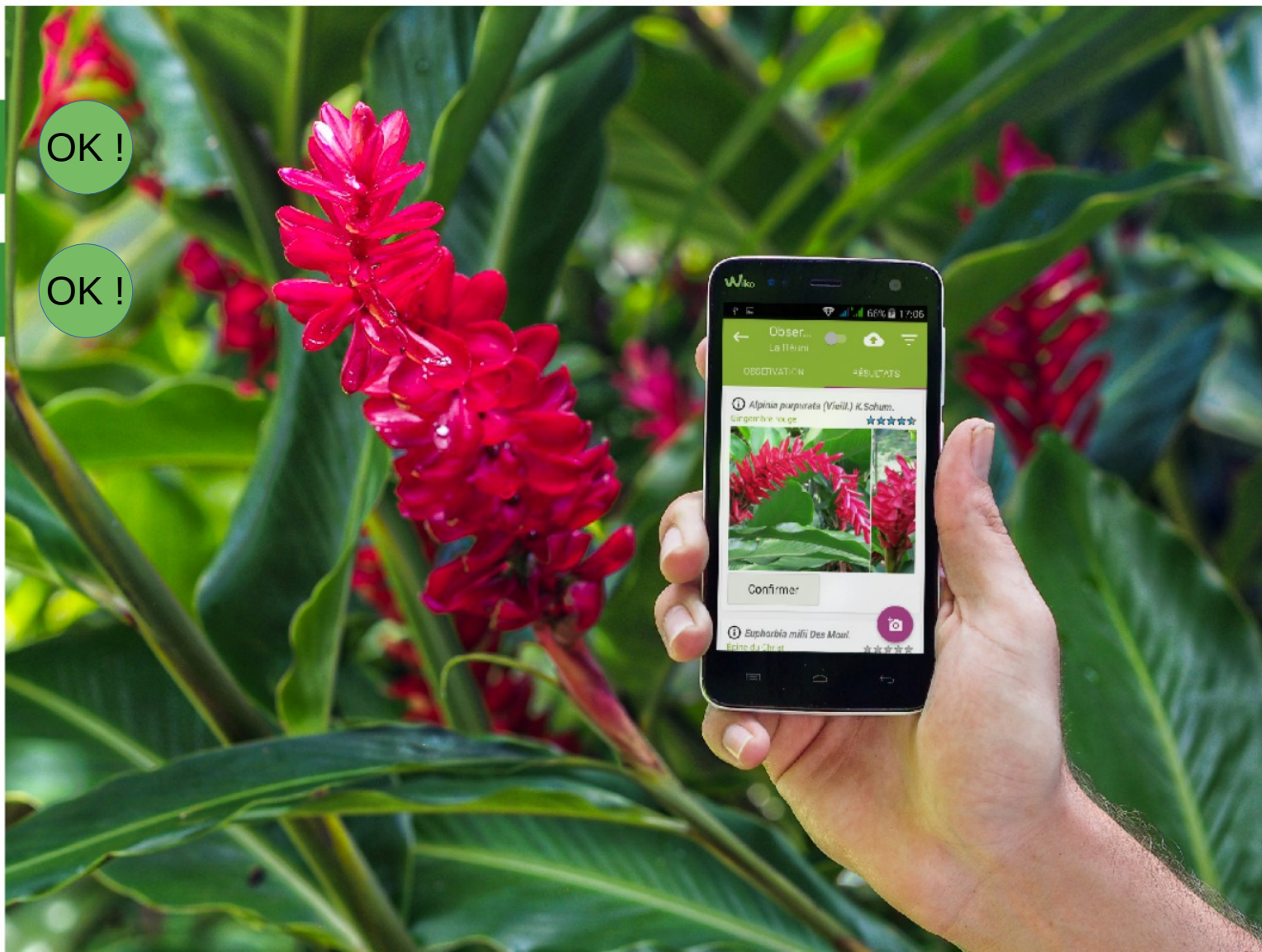
Abeliophyllum
 Acacia
 Acalypha
 Achyranthes
 Aciotis
 Adenostyles
 Adlumia
 Adonis
 Aegopodium
 Aeschynomene
 Aextoxicon
 Aizoanthemum
 Aizoon
 Alcea
 Alibertia
 Alliaria
 Alocasia
 Althaea
 Anemone
 Angelica
 Angostura
 Anisocampium
 Ansellia
 Anthericum
 Anthurium
 Antirhea
 Aphelandra
 Aralia
 Aristea
 Arthraxon
 Aspidia
 Astydamia
 Asystasia
 Atocion
 Atractocarpus
 Balsamorhiza
 Barbarea
 Barringtonia
 Bellium
 Berula
 Bismarckia
 Bocoa
 Bonafousia
 Boscia
 Bourreria

Pareto Effect

OK !

Genus

OK !



Un point sur le calendrier



A calendar interface for June 2021. The title 'juin 2021' is at the top left, with navigation arrows (up and down) to its right. The days of the week are abbreviated as 'lu', 'ma', 'me', 'je', 've', 'sa', 'di'. The dates are arranged in a grid. The date '28' is highlighted with a blue border. The date '31' is shown in the first row, indicating the month ends on a Friday. The next row shows dates from 7 to 13, then 14 to 20, then 21 to 27. The highlighted '28' is in the row starting with 21. The final row shows dates 5 to 11, which are likely from the following month.


| juin 2021 | | | | | | | ^ | v |
|-----------|----|----|----|----|----|----|---|---|
| lu | ma | me | je | ve | sa | di | | |
| 31 | 1 | 2 | 3 | 4 | 5 | 6 | | |
| 7 | 8 | 9 | 10 | 11 | 12 | 13 | | |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 | | |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | | |
| 28 | 29 | 30 | 1 | 2 | 3 | 4 | | |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 | | |


Documentation consultée


- L'article sur l'algo PCA
- Les cinq conférences sur la classification d'images
- Un peu de documentation informatique sur le TSNE et l'affichage graphique 3D

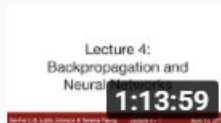
Lecture Collection | Convolutional Neural Networks for
Stanford University School of Engineering - 1/16

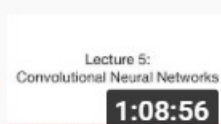
↺ ↻

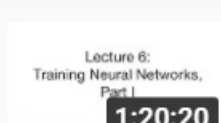
▶  Lecture 1: Introduction
57:57
Lecture 1 | Introduction to Convolutional Neural Networks for...
Stanford University School of Engineering

2  Lecture 2:
Image Classification pipeline
59:32
Lecture 2 | Image Classification
Stanford University School of Engineering

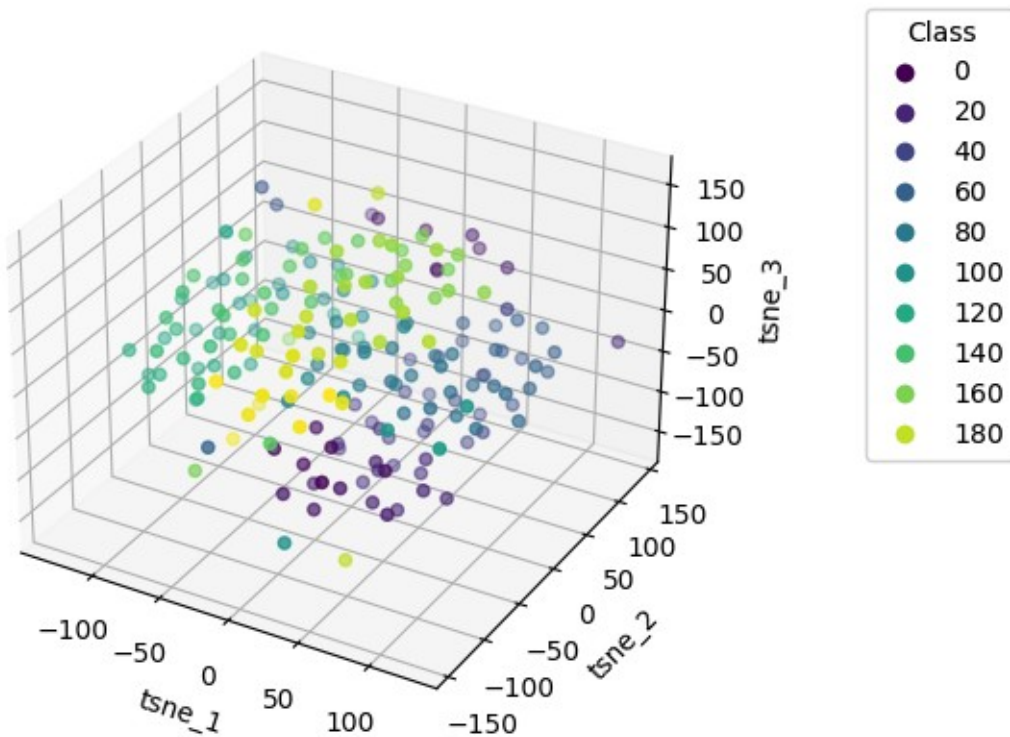
3  Lecture 3:
Loss Functions
and Optimization
1:14:40
Lecture 3 | Loss Functions and Optimization
Stanford University School of Engineering

4  Lecture 4:
Backpropagation and
Neural Networks
1:13:59
Lecture 4 | Introduction to Neural Networks
Stanford University School of Engineering

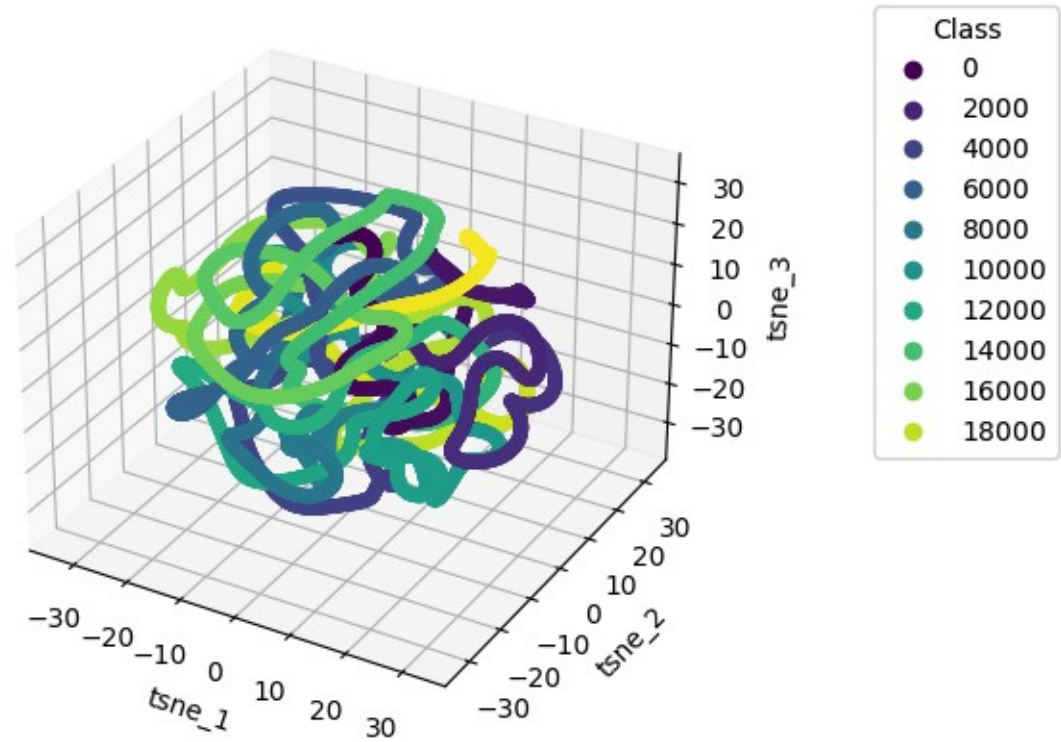
5  Lecture 5:
Convolutional Neural Networks
1:08:56
Lecture 5 | Convolutional Neural Networks
Stanford University School of Engineering

6  Lecture 6:
Training Neural Networks,
Part I
1:20:20
Lecture 6 | Training Neural Networks I
Stanford University School of Engineering

TSNE pour 200 vecteurs



TSNE pour 20 000 vecteurs



Ai-je bien codé la transformation TSNE de la matrice A ?

A (20 000x100)

```
[[0.0000e+00 0.0000e+00 0.0000e+00 ... 0.0000e+00 0.0000e+00 0.0000e+00]
 [1.0000e+00 1.0000e+00 1.0000e+00 ... 1.0000e+00 1.0000e+00 1.0000e+00]
 [2.0000e+00 2.0000e+00 2.0000e+00 ... 2.0000e+00 2.0000e+00 2.0000e+00]
 ...
 [1.9997e+04 1.9997e+04 1.9997e+04 ... 1.9997e+04 1.9997e+04 1.9997e+04]
 [1.9998e+04 1.9998e+04 1.9998e+04 ... 1.9998e+04 1.9998e+04 1.9998e+04]
 [1.9999e+04 1.9999e+04 1.9999e+04 ... 1.9999e+04 1.9999e+04 1.9999e+04]]
```

$$\in \mathbb{R}^{100} \left\{ \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \begin{pmatrix} 2 \\ 2 \\ \vdots \\ 2 \end{pmatrix} \dots \begin{pmatrix} 19\,999 \\ 19\,999 \\ \vdots \\ 19\,999 \end{pmatrix} \right.$$
$$\xrightarrow{\text{TSNE}} \begin{pmatrix} \vdots \\ \vdots \\ \vdots \end{pmatrix} \begin{pmatrix} \vdots \\ \vdots \\ \vdots \end{pmatrix} \begin{pmatrix} \vdots \\ \vdots \\ \vdots \end{pmatrix} \dots \begin{pmatrix} \vdots \\ \vdots \\ \vdots \end{pmatrix}$$

```
label = list(i for i in range(20000))
→ ligne i : classe « i »
```

```
tsne = TSNE(3)
tsne_result= tsne.fit_transform(A)
```


Quelques questions

Y a-t-il une bijection entre l'ensemble des vecteurs avant TSNE et l'ensemble des vecteurs réduits après TSNE ?

Forme serpent.

Pourquoi utiliser la méthode de descente de gradient plutôt que la méthode PCA pour déterminer la matrice W ? De même, pourquoi utiliser TSNE si l'algo PCA est le meilleur pour la « recovery » ? Brefs quels liens entre PCA, TSNE, descente gradient, Neural Network et Convolutional Network ?

(→ couches « layers » ?)

Where we are now...

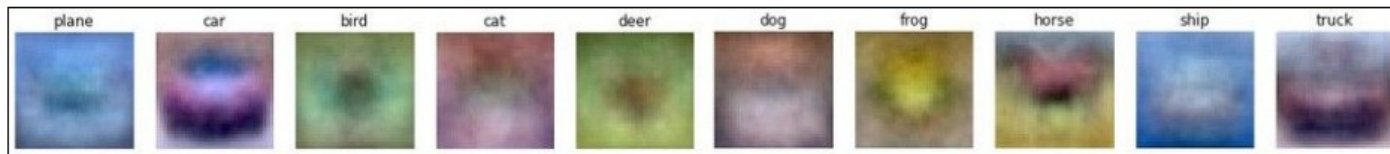
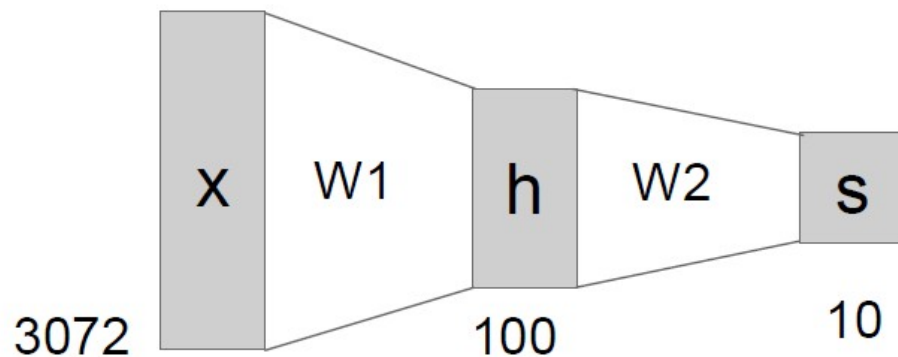
Neural Networks

Linear score function:

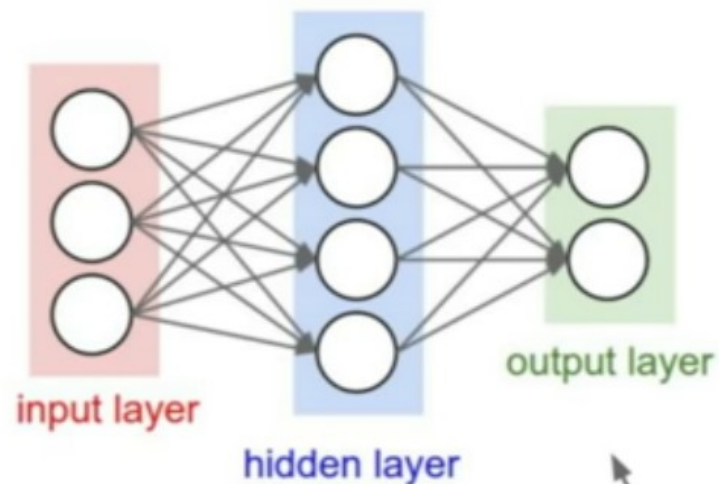
$$f = Wx$$

2-layer Neural Network

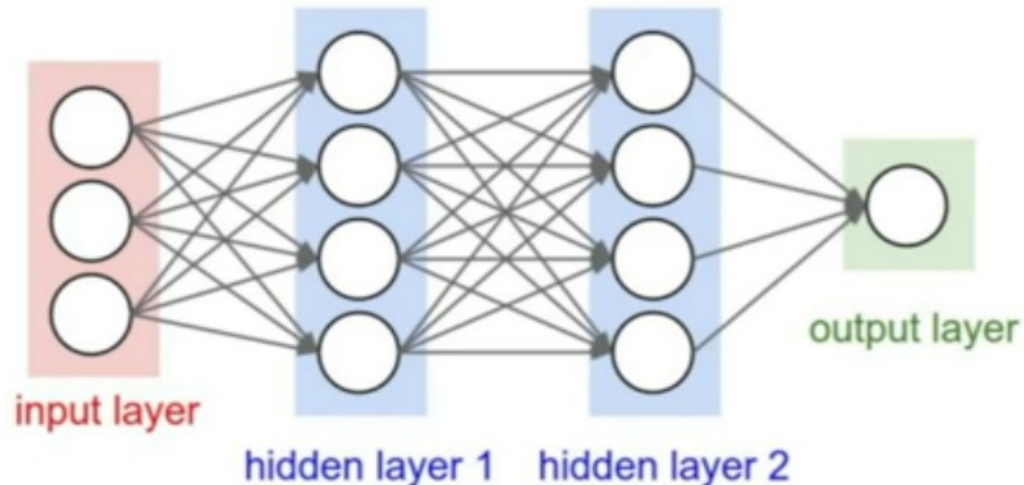
$$f = W_2 \max(0, W_1 x)$$



Neural networks: Architectures



"2-layer Neural Net", or
"1-hidden-layer Neural Net"



"3-layer Neural Net", or
"2-hidden-layer Neural Net"

"Fully-connected" layers

Stanford

Where we are now...

Convolutional Neural Networks

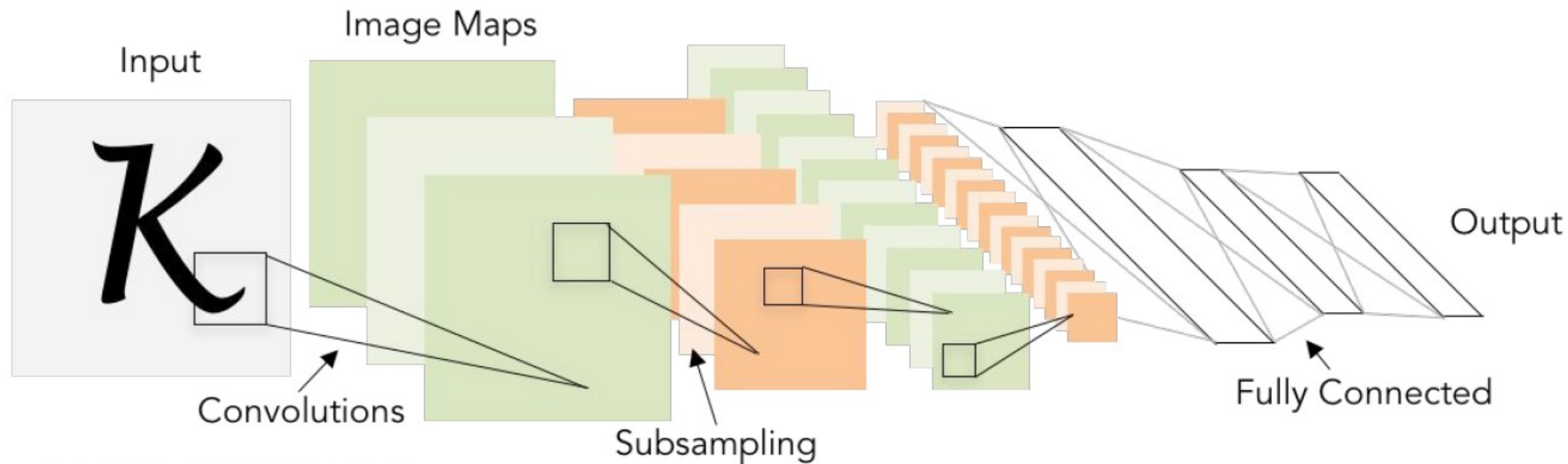


Illustration of LeCun et al. 1998 from CS231n 2017 Lecture 1

In PCA, we find the compression matrix W and the recovering matrix U so that the total squared distance between the original and recovered vectors is minimal; namely, we aim at solving the problem

$$\operatorname{argmin}_{W \in \mathbb{R}^{n,d}, U \in \mathbb{R}^{d,n}} \sum_{i=1}^m \|\mathbf{x}_i - UW\mathbf{x}_i\|_2^2. \quad (23.1)$$

To solve this problem we first show that the optimal solution takes a specific form.

Y a-t-il une bijection entre l'ensemble des vecteurs avant TSNE et l'ensemble des vecteurs réduits après TSNE ?

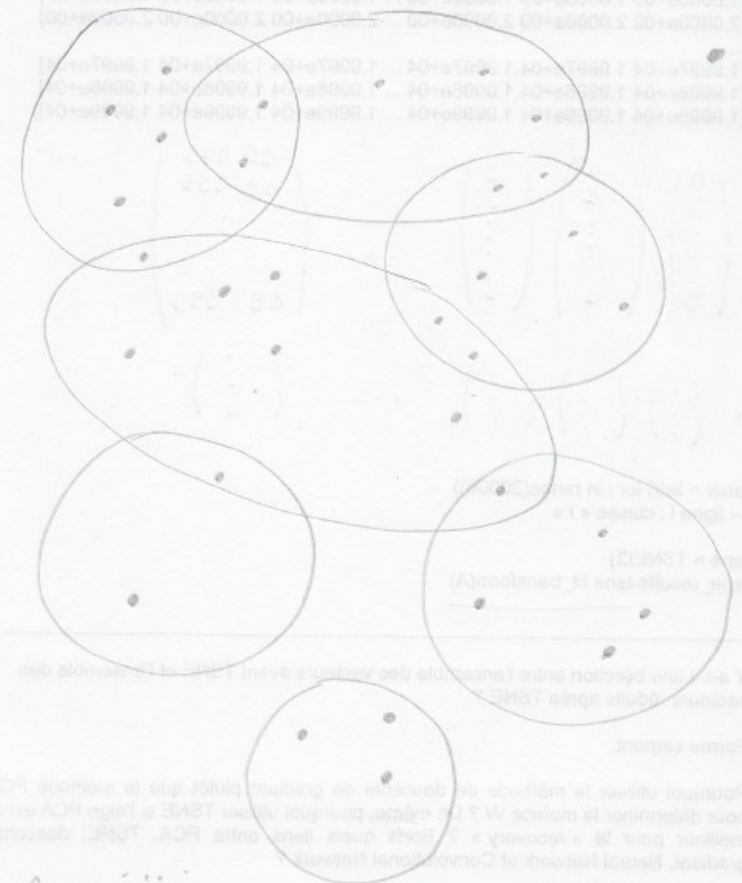
Forme serpent.

Pourquoi utiliser la méthode de descente de gradient plutôt que la méthode PCA pour déterminer la matrice W ? De même, pourquoi utiliser TSNE si l'algo PCA est le meilleur pour la « recovery » ? Brefs quels liens entre PCA, TSNE, descente gradient, Neural Network et Convolutional Network ?

(→ couches « layers » ?)

« Meta-groupes » : ensembles de groupes

○ Meta-groupes
● espèces



- Meta groupe 1
- Meta groupe 2
- Meta groupe 10

↑ gradation en couleurs

