# Démêlage de représentations neuronales pour l'audio et la voix

Sous la tutelle de M. Artières

Par Killian Le Goff

# Contexte

Dans le cadre d'un projet de recherche du LIS en lien avec l'équipe de neurosciences de la Timone, on cherche à pouvoir altérer des données vocales.

Pour ce faire, ce projet propose d'explorer l'**espace latent** d'un réseau de neurones, afin d'altérer certains attributs de la donnée (essentiellement l'âge, le sexe, l'accent).

# Sommaire

1. Notions essentielles de machine learning
2. Approche du projet
3. Résultats
4. Conclusion et perspectives

# 1. Notions de Machine learning

**Machine learning** : optimiser une modèle sur un jeu de données

**Deep Learning** : approche de modélisation de plus haut niveau (représentations), inspiré du cerveau

Réseau de neurones **convolutionnels** : apprentissage de filtres de convolution (adapté à computer vision, et speech recognition) - identification de pattern

# Autoencodeur

-Réseau de neurone

-Apprentissage de représentation compressée (espace latent, faible dimension)

-Reconstruction de la donnée

-Encodeur et décodeur entrainés ensemble

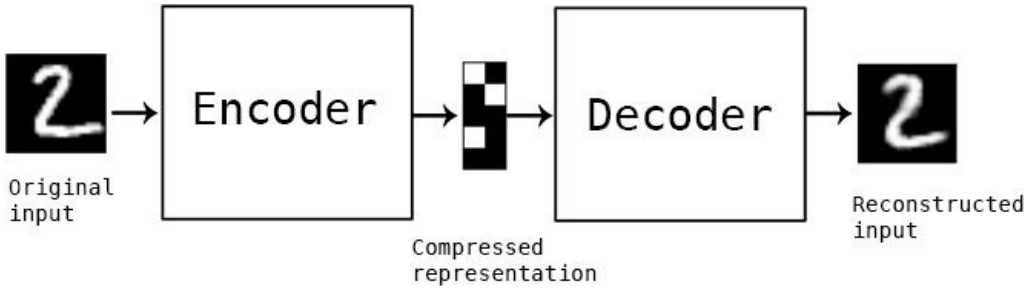



*fig.1 Illustration du fonctionnement d'un autoencodeur*

*source : https://blog.keras.io/building-autoencoders-in-keras.html*

# GAN : Generative Adversarial Network

- 2 réseaux concurrents : générateur / discriminateur (entraînement en plusieurs étapes)

-Espace latent en entrée du générateur, pour obtenir de la donnée aléatoire

Apprendre des directions avec le GAN : (faire coïncider les axes avec les directions de nos attributs)

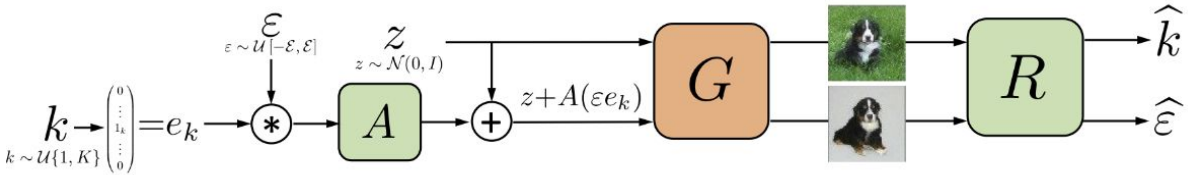



*fig.2 Illustration du fonction d'un GAN dans la méthode de l'article (1)*

source : https://arxiv.org/pdf/2002.03754.pdf

# Autoencodeur Variationnel

-Même architecture que l'autoencodeur classique

-Arithmétique dans l'espace latent :

On transforme les modalités de nos données

(ex : 6 -> 8 sur la fig.3)

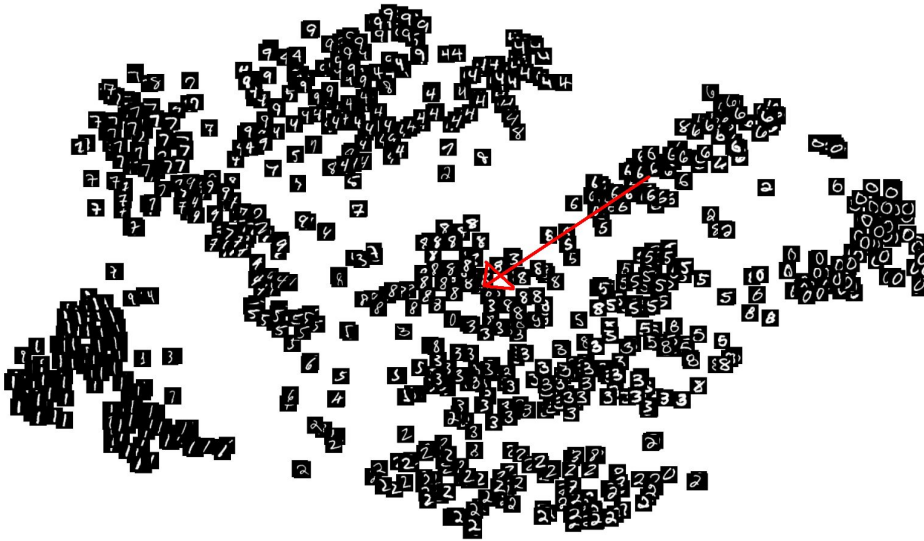


*fig.3 Représentation apprise du dataset "digit" de scikit-learn*

*source : https://hackernoon.com/latent-space-visualization-deep-learning-bits-2-bd09a469*

# 2. Approche du projet

Jeu de données collaboratif

Mozilla Common Voice (fig.4)

A l'obtention du jeu de données, nous avons :
-la phrase prononcée (original_sentence)
-références de l'enregistrement
(-les caractéristiques du locuteur (age, gender, accent)
-des fichiers audios (qques secondes)


*fig.4 Composition du dataset Common Voice*

*source : https://commonvoice.mozilla.org/fr/datasets*

# Prétraitement

-On cale l'audio sur le texte lu

-Spectrogramme = Calcul des transformées de Fourier discrètes pour chaque acquistion audio

(Autres approches : Mfcc)

=>Données rectangulaires (CNN)

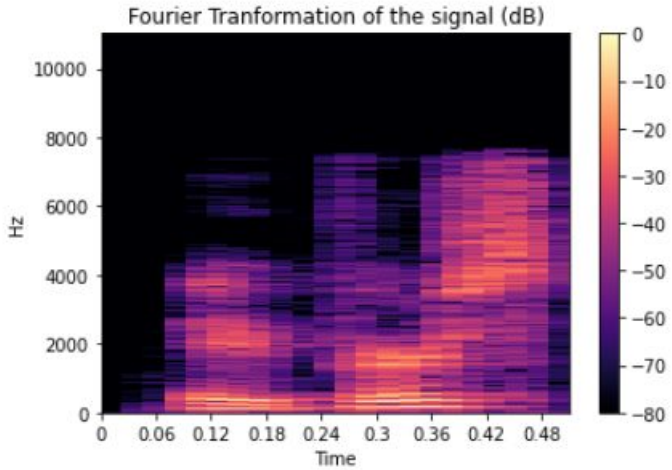



*fig.6 Allure d'un spectrogramme calculé sur un enregistrement de voix*

# Approche de machine learning

Autoencodeur pré-entrainé sur des données spectrogrammes

On ajoute la dimension “variationnelle” à l’autoencodeur

(projeter dans l’espace latent avec l’encodeur, puis modifier la donnée dans l’espace latent)

On reconstruit avec le décodeur

En parallèle, on entraine des classifieurs sur les différents attributs : age, gender

Le but est de déterminer si la modification de la donnée est crédible

(ex : on transforme une voix d'homme en femme, et on étudie le résultat de ces classifieurs)

# Exploration de l'espace latent

-Espace latent de dimension 264

Tracé graphique pour étudier la séprabilité

PB : trop de dimensions, et la variance est répartie selon les axes

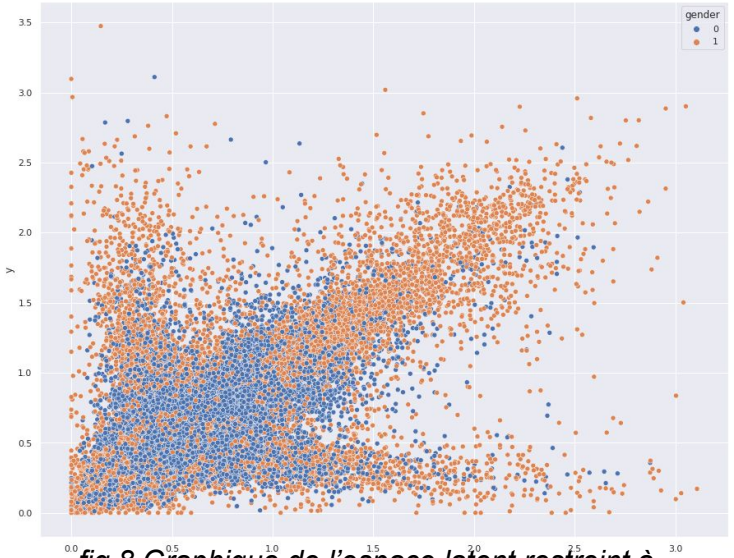



*fig.8 Graphique de l'espace latent restreint à 2 axes, avec indication de l'attribut "gender"*

-Approche par ACP :

rédution de dimension, condense la variance sur les premiers axes

PB : la variance est étalée sur les axes (pour 90% de la variance,

plus 100 axes)

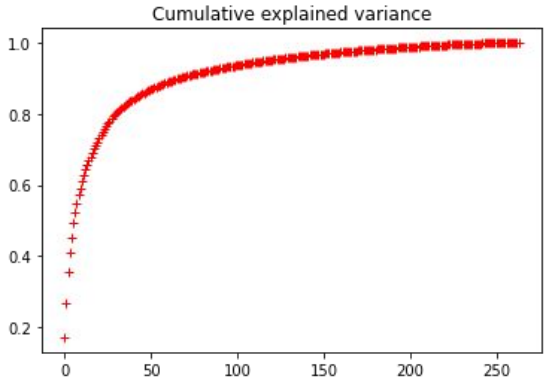



*fig.9 Variance explicative cumulée de l'ACP*

TSNE = méthode de réduction de dimension

Conserve la proximité entre les données

Les attributs ne sont pas séparables

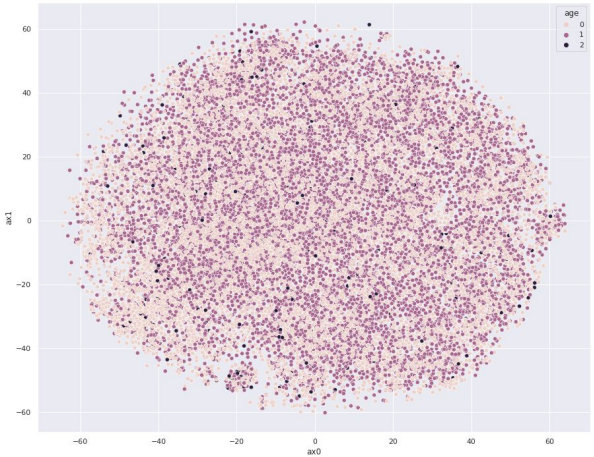


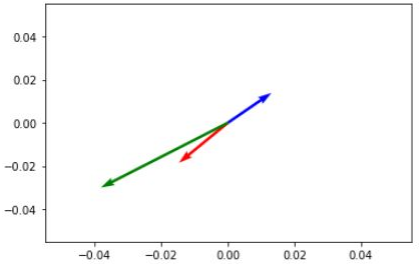

*fig.11 Représentation de la réduction de dimension par TSNE*

On calcule les vecteurs moyens des modalités (fig12)

La variance est largement supérieure à la valeur moyenne

=> Une majeure partie de la variance est due à la variabilité des textes lus

et des phonemes prononcés




```
Age 0 mean vector: [-2.0073302, -1.6413031] +/- [927.06366, 777.23
Age 1 mean vector: [1.7862424, 1.2266225] +/- [797.69507, 755.3502
Age 2 mean vector: [-5.2060485, -2.6586773] +/- [848.37604, 839.39
```

*fig.12 : Vecteurs moyens des modalités de la variable âge, dans l'espace obtenu par TSNE*

# Trucage de la donnée

On réalise l'opération suivante pour modifier l'attribut de notre donnée

$$\textit{shifted data} = \textit{latent data} + \textit{eps} * (\textit{end mod} - \textit{start mod})$$

Ensuite on challenge nos classifieurs d'attributs pour déterminer si le trucage est crédible.

# Classifieur d'attribut

Modèle CNN

Précision assez faible pour l'age

(question sur la pertinence de cet attribut

et sur la façon utilisée pour remplir les attributs)

| gender_prediction | 0 | 1 |
|---|---|---|
| gender | | |
| 0 | 2779 | 400 |
| 1 | 219 | 1613 |

Accuracy = 0.8765

| age_prediction | 0 | 1 |
|---|---|---|
| age | | |
| 0 | 213 | 1999 |
| 1 | 179 | 2558 |
| 2 | 3 | 72 |

Accuracy = 0.5516

fig.14 Matrice de confusion et précision des modèles de classification d'attributs

# Résultats

```
1    21480
0    17612
2      626
Name: age, dtype: int64

1    36531
0     3169
Name: age_shift10, dtype: int64

1    37274
0     2426
Name: age_shift01, dtype: int64
```

```
0    25190
1    14528
Name: gender, dtype: int64

0    21391
1    18309
Name: gender_shift10, dtype: int64
```