

Rapport de l'activité 4 sur l'introduction à la machine learning du Groupe 4

ALLAMADJI DJIMRASSEM Innocent
NDILNODJI Djimadoumbaye MODJIDON Exaucé
BEMADJI Arsène DJETOURALAR Yannick

28 mai 2023



Université de n'djamena

République du Tchad

Unité-Travail-Progress

Université de N'Djamena

Faculté des sciences exactes et appliquées

Département d'Informatique

Projet Tech4tchad



THEME : *l'algorithme t-SNE(t-distributed Stochastic Neighbor Embedding)*

Table des matières

0.1	Introduction	1
0.2	Organisation de l'équipe	1
0.2.1	Répartition des différentes taches des membres pour la réalisation du travail	1
0.3	Présentation du travail des membres de l'équipe	1
0.3.1	Notion de réduction de dimension en Machine Learning	1
0.3.2	Principe Et Fonctionnement de t-SNE	2
0.3.3	Quand et comment utiliser le t-SNE ?	2
0.4	Un exemple de code avec l'utilisation de t-SNE	3
0.4.1	Resultat apres execution du code	3
0.5	Conclusion	4

0.1 Introduction

Si par hasard vous avez déjà travaillé avec un ensemble de données comportant beaucoup de points forts, vous pouvez comprendre qu'il est si difficile de comprendre ou d'étudier les liens entre les points forts. Cela rend la procédure de l'AED problématique et influence la présentation du modèle d'IA, car il y a de fortes chances que vous sur ajustiez votre modèle ou que vous endommagiez une partie des soupçons du calcul, comme c'est le cas pour l'autonomie des points saillants en cas de rechute directe. C'est là que la diminution de la dimensionnalité entre en jeu. Dans l'IA, la diminution de la dimensionnalité est le moyen de réduire le nombre de facteurs irréguliers viables en obtenant beaucoup de facteurs de tête. En diminuant la composante de l'espace de vos éléments, vous avez moins de connexions entre les mèches à prendre en compte, qui peuvent être étudiées et représentées efficacement, et en outre vous êtes plus réticent à surdimensionner votre modèle.

0.2 Organisation de l'équipe

0.2.1 Répartition des différentes tâches des membres pour la réalisation du travail

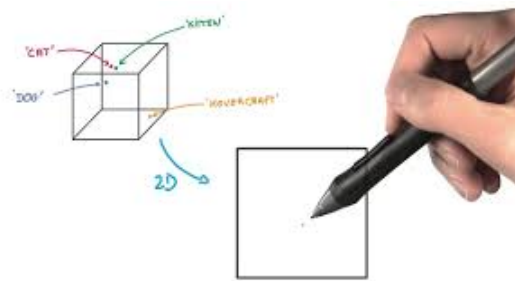
Pour réaliser cette 4ème activité sur la machine learning qui nous ait été soumis voici l'organisation dont nous avons adopté

Nom et Prénom	Tache
ALLAMADJI DJIMRASSEM Innocent	fonctionnement de t-SNE
NDILNODJI Djimadoubaye	réduction des Dimensions
MODJIDON Exaucé	????
BEMADJI Arsène	Quand et Comment utiliser t-SNE
DJETOURALAR Yannick	exemple de code

0.3 Présentation du travail des membres de l'équipe

0.3.1 Notion de réduction de dimension en Machine Learning

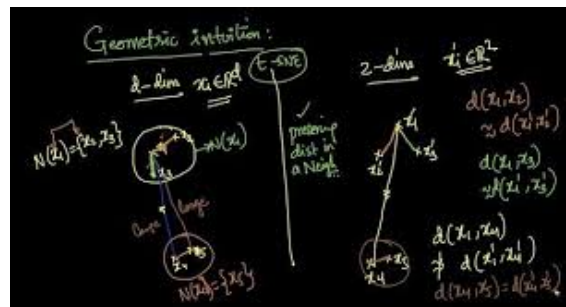
La réduction de dimension est un processus étudié en mathématiques et en informatique. Il consiste à prendre des données dans un espace de grande dimension, et à les remplacer par d'autres dans un espace de dimension inférieure, mais qui contiennent encore la plupart des informations contenues dans le grand ensemble. Autrement dit, on cherche à construire moins de variables tout en conservant le maximum d'informations possible. En Machine Learning, ce processus de traitement de données est crucial dans certains cas, parce que les jeux de données plus petits sont plus faciles à explorer, exploiter et à visualiser, et rendent l'analyse des données beaucoup plus facile et plus rapide. Cette étape est importante aussi dans les cas du sur-apprentissage et des données très éparses (fléau de la dimensionnalité), qui nécessitent beaucoup de temps et de puissance de calcul pour les étudier. En utilisant un espace de plus petite dimension, on obtient des algorithmes plus efficaces, ainsi qu'un panel de solutions plus réduit.



0.3.2 Principe Et Fonctionnement de t-SNE

Le principe de t-SNE : il consiste à représenter chaque point de données comme un nuage dans l'espace. Les points les plus proches sont regroupés en un seul nuage, tandis que les points plus éloignés sont séparés en différents nuages. Cette technique permet de mettre en évidence les structures locales et globales des données.

Fonctionnement : Pour comprendre l'utilité de t-SNE, il est important de comprendre que lorsque nous travaillons avec des données de grande dimension, la visualisation de ces données peut être difficile, car notre cerveau peut difficilement traiter autant d'informations en même temps. En utilisant t-SNE, nous pouvons réduire le nombre de dimensions de nos données tout en conservant les relations entre les points dans notre ensemble de données original.



0.3.3 Quand et comment utiliser le t-SNE ?

Le t-SNE est un algorithme très intelligent qui permet d'éviter le sur-apprentissage et de simplifier les données. Il s'adapte à des situations de réductions de dimensions précises. Cet algorithme gère efficacement et intelligemment les données non linéaires et les données qui présentent des valeurs aberrantes ou extrêmes contrairement à L'ACP qui est un algorithme linéaire.. Il faut faire attention à ne pas utiliser le t-SNE avec des jeux de données volumineux, parce qu'il implique beaucoup de calculs. Il est particulièrement lent et gourmand en matière de puissance de calcul. Le t-SNE est non déterministe, parfois, on n'obtiendra pas exactement la même sortie chaque fois qu'on l'exécute, bien que les résultats sont susceptibles d'être similaires. Il est donc efficace pour comprendre et décortiquer des données complexes, mais pas vraiment pour l'entraînement des modèles Machine Learning puisqu'il peut changer de résultats d'une exécution à une autre. Il peut être utilisé pour comprendre et interpréter les réseaux de neurones convolutifs (CNN) qui sont généralement des boîtes noires.

0.4 Un exemple de code avec l'utilisation de t-SNE

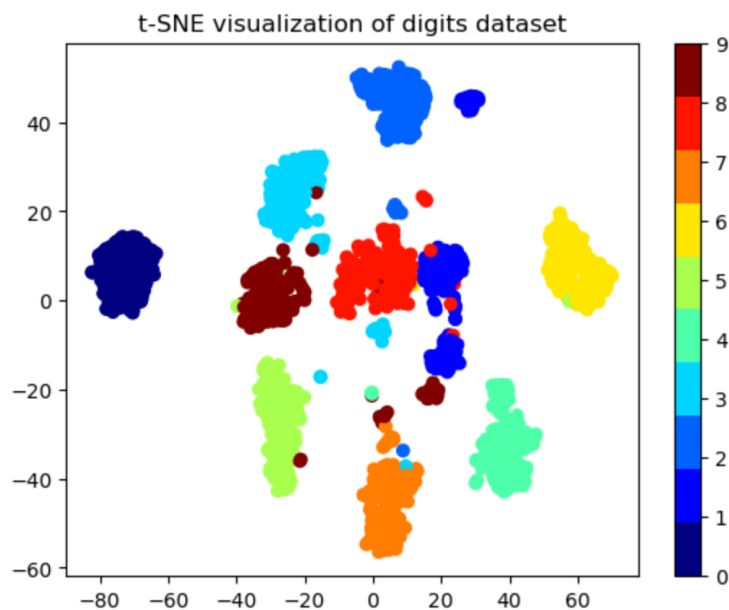
```
#on commence par importer les bibliothèques nécessaires
import numpy as np
import matplotlib.pyplot as plt
from sklearn.manifold import TSNE
from sklearn.datasets import load_digits

#ensuite on Charge les jeu de donnée intégrés dans sklearn
digits = load_digits()
X = digits.data
y = digits.target

#Utilisons la c t-SNE pour transformer les données en seulement deux dimensions
tsne = TSNE(n_components=2, random_state=0)
X_2d = tsne.fit_transform(X)

#Affichons la visualisation en deux dimensions
plt.scatter(X_2d[:, 0], X_2d[:, 1], c=y, cmap=plt.cm.get_cmap('jet', 10))
plt.colorbar(ticks=range(10))
plt.title('t-SNE visualization of digits dataset')
plt.show()
```

0.4.1 Resultat apres execution du code



0.5 Conclusion

l'algorithme t-SNE est un excellent outil pour comprendre les ensembles de données de grande dimension. Cela peut être moins utile lorsque vous souhaitez effectuer une réduction de dimensionnalité pour la formation ML (ne peut pas être réappliqué de la même manière). Ce n'est pas déterministe et itératif, donc chaque fois qu'il s'exécute, il peut produire un résultat différent. Mais même avec ces inconvénients, elle reste l'une des méthodes les plus populaires dans le domaine.

Références

<https://chat-gpt/chat>
<https://blent.ai/blog/a/algorithmetsne>