: N° d'ordre
: N° de série

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

---

UNIVERSITE ECHAHID HAMMA LAKHDAR - EL OUED
FACULTÉ DES SCIENCES EXACTES
Département D'Informatique

Mémoire de Fin D'étude
Présenté pour l'obtention du Diplôme de

MASTER ACADEMIQUE

Domaine : **Mathématique et Informatique**
Filière : **Informatique**
Spécialité : **Systèmes Distribués et Intelligence Artificielle**

Présenté par :

- **Abderrahmane Rezig**
- **Bachir Labbi**

Thème

Méthode stochastique d'augmentation des données pour améliorer l'apprentissage machine.

Soutenue le xx-xx- 2022 Devant le jury:

M.	……	MCA	Président
M.	……	MAA	Rapporteur
M.	Naoui Med Anouar	MAA	Encadreur

Année Universitaire: 2021/2022

Résumé :

L'objectif de notre travail est d’adresser le problème le plus courant en apprentissage automatique, à savoir la quantité insuffisante de données d'entraînement. Une façon de résoudre ce problème est appelée augmentation de données, qui est un ensemble de techniques utilisées pour augmenter artificiellement la quantité de données en créant de nouveaux points de données à partir de données existantes. Utile pour améliorer les performances et les résultats des modèles d'apprentissage automatique en formant des exemples nouveaux et différents d'ensembles de données de formation. Si l'ensemble de données d'un modèle d'apprentissage automatique est riche et suffisant, le modèle fonctionne mieux et est plus précis. Cela nous d’étudier l’utilisation de deux technique, Bootstrap et Jackknife, qui augmentent la quantité de données et donnent des résultats acceptables et quelque peu satisfaisants par rapport à d'autres technique plus largement utilisées et populaires, et nous avons choisi la technique shift comme modèle de comparaison.

Mots clés:

Apprentissage automatique, augmentation des données, données d'entraînement, amélioration des performances, techniques Bootstrap et Jackknife , comparé la techniques de shift.

Abstract :

The objective of our work is to address on the most common problem in machine learning, namely the insufficient amount of training data. One way to solve this problem is called data augmentation, which is a set of techniques used to artificially increase the amount of data by creating new data points from existing data. Useful for improving the performance and results of machine learning models by training new and different examples of training data sets. If the dataset for a machine learning model is rich and sufficient, the model performs better and is more accurate. This allowed us to develop two technique, Bootstrap and Jackknife, that augment the amount of data and give acceptable and somewhat satisfactory results compared to other more widely used and popular technique, and we chose the shift technique as the comparison model.

Keywords:

Machine Learning, Data Augmentation, Training Data, Performance Improvement, Bootstrap Techniques And Jackknife, compared the shift Techniques.

ملخص:

الهدف من عملنا هو العمل على المشكلة الأكثر شيوعًا في التعلم الآلي ، وهي عدم كفاية كمية بيانات التدريب. إحدى طرق حل هذه المشكلة تسمى زيادة البيانات ، وهي عبارة عن مجموعة من التقنيات المستخدمة لزيادة كمية البيانات بشكل مصطنع عن طريق إنشاء نقاط بيانات جديدة من البيانات الموجودة. مفيد لتحسين أداء ونتائج نماذج التعلم الآلي من خلال تدريب أمثلة جديدة ومختلفة لمجموعات بيانات التدريب. إذا كانت مجموعة البيانات الخاصة بنموذج التعلم الآلي غنية وكافية ، فإن النموذج يعمل بشكل أفضل وأكثر دقة. سمح لنا ذلك بتطوير تقنيتين ، Bootstrap و Jackknife ، تزيدان كمية البيانات وتعطي نتائج مقبولة ومرضية إلى حد ما مقارنة بالتقنيات الأخرى الأكثر استخدامًا والشعبية ، وقد اخترنا تقنية shift كنموذج مقارنة.

الكلمات المفتاحية:

التعلم الآلي , زيادة البيانات , بيانات التدريب, تحسين الأداء , تقنيات Bootstrap و Jackknife , مقارنة تقنية shift .

Sommaire

Chapitre II : Rééchantillonnage (bootstrap,Jackknife).

Chapitre III : Conception & affichage des resultats.

Liste des figures

Liste des tableaux

Introduction général

Introduction général :

Des progrès considérables ont été réalisés dans l'utilisation de modèles d'apprentissage en profondeur pour augmenter le nombre de données et améliorer la précision des modèles.

L'augmentation des données est une méthode de prétraitement importante qui s'est avérée efficace dans la formation de modèles d'apprentissage en profondeur très divergents. L'augmentation des données a été initialement généralisée grâce à des techniques nouvelles et innovantes pour rendre le modèle plus réalisable et plus précis. Dans le domaine de la vision par ordinateur, puisqu'il existe des millions voire des milliards de paramètres dans les CNN, l'augmentation des données est nécessaire pour collecter suffisamment de données pour obtenir des performances satisfaisantes. Plusieurs stratégies d'augmentation des données ont été proposées pour améliorer l'apprentissage automatique.

Bien que des recherches approfondies aient été effectuées sur les stratégies de mise à l'échelle des données, développer deux stratégies Bootstrap et Jackknife pour faire le travail, et avons choisi l'ensemble de données mnist pour nos tests, la bibliothèque tensorflow fournissant l'ensemble de données principal qui est un grand ensemble d'images manuscrites. Nombres. Il existe 60 000 images d'entraînement et 10 000 images de test [1].

Dans ce travail, des stratégies ont été programmées pour augmenter les cadres de données, et un certain nombre de tests ont été effectués démontrant la précision de l'apprentissage du modèle, la validation des résultats et la comparaison avec d'autres stratégies populaires. Enfin, nous montrons que les stratégies que nous proposons sont plus précises et efficaces.

Le travail considéré contient trois chapitres :

Le chapitre 1 définira les concepts de base utilisés.Au chapitre 2, nous donnerons une définition générale des stratégies proposées.Chapitre 3

Nous allons présenter la structure générale de notre système et montrer les résultats obtenus.

Chapitre I

1-Introduction:

La performance de la plupart des modèles machine learning (ML), et des modèles d'apprentissage profond en particulier, dépend de la qualité, de la quantité et de la pertinence des données d'entraînement. Cependant, l'insuffisance des données est l'un des défis les plus courants dans la mise en œuvre de l'apprentissage automatique dans l'entreprise. Cela s'explique par le fait que la collecte de ces données peut être coûteuse et prendre du temps dans de nombreux cas.

Les entreprises peuvent tirer parti de l'augmentation des données pour réduire la dépendance à la collecte et à la préparation des données de formation et pour construire plus rapidement des modèles d'apprentissage automatique plus précis.[1]

2- Définition de l'augmentation des données:

L'augmentation des données dans l'analyse des données est une technique utilisée pour accroître la quantité de données en ajoutant des copies légèrement modifiées de données existantes ou des données synthétiques nouvellement créées à partir de données existantes. Elle agit comme un régularisateur et aide à réduire l'overfitting lors de la formation d'un modèle d'apprentissage automatique[2]. Elle est étroitement liée au suréchantillonnage dans l'analyse des données.

3- Les importance de l'augmentation:

Les applications de l'apprentissage automatique, en particulier dans le domaine de l'apprentissage profond, continuent de se diversifier et d'augmenter rapidement. Les techniques d'augmentation des données peuvent être un bon outil pour relever les défis auxquels le monde de l'intelligence artificielle est confronté.

L'augmentation des données permet d'améliorer les performances et les résultats des modèles d'apprentissage automatique en ajoutant des exemples nouveaux et différents aux ensembles de données d'entraînement. Si l'ensemble de données d'un modèle d'apprentissage automatique est riche et suffisant, le modèle est plus performant et plus précis.

Pour les modèles d'apprentissage automatique, la collecte et l'étiquetage des données peuvent être des processus épuisants et coûteux. La transformation des ensembles de données à l'aide de techniques d'augmentation des données permet aux entreprises de réduire ces coûts opérationnels.

L'une des étapes d'un modèle de données est le nettoyage des données, qui est nécessaire pour les modèles de haute précision. Cependant, si le nettoyage réduit la représentabilité des données, le modèle ne peut pas fournir de bonnes prédictions pour les données du monde réel. Les techniques d'augmentation des données permettent aux modèles d'apprentissage automatique d'être plus robustes en créant des variations que le modèle peut voir dans le monde réel. [1]

4-Le chemin du travail:

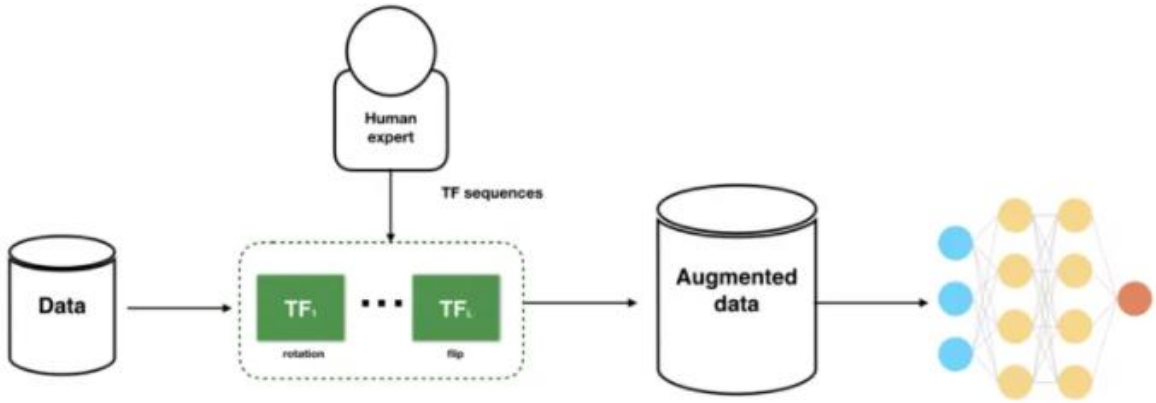


Figure 1: Augmentation des données [3]

Les schémas d'augmentation de données heuristiques reposent souvent sur la composition d'un ensemble de fonctions de transformation simples (TF) telles que les rotations et les retournements (voir la figure 1). Lorsqu'ils sont choisis avec soin, les schémas d'augmentation des données réglés par des experts humains peuvent améliorer les performances du modèle. Cependant, de telles stratégies heuristiques dans la pratique peuvent entraîner de grandes variations dans les performances du modèle final et peuvent ne pas produire les augmentations nécessaires pour les modèles de pointe.

5- Augmentation et la segmentation des images:

Pour l'augmentation des données, il est courant d'apporter de simples modifications aux données visuelles. En outre, les réseaux adversariens génératifs Generative Adversarial Networks (GAN) sont utilisés pour créer de nouvelles données synthétiques. Les activités classiques de traitement d'images pour l'augmentation des données sont les suivantes:

- le rembourrage
- la rotation aléatoire
- la remise à l'échelle
- retournement vertical et horizontal
- translation (l'image est déplacée dans les directions X et Y)
- recadrage
- zooming
- assombrissement et éclaircissement/modification des couleurs
- mise à l'échelle des gris
- modification du contraste
- ajout de bruit
- effacement aléatoire

Figure 2: Exemple d'augmentation des données [4]

6- Les modèles avancés d'augmentation des données sont:

- **Formation contradictoire/Apprentissage automatique contradictoire :** Il génère des exemples adverses qui perturbent un modèle d'apprentissage automatique et les injecte dans un ensemble de données pour l'entraîner.
- **Réseaux adversaires génératifs (GAN) :** Les algorithmes GAN peuvent apprendre des modèles à partir de jeux de données d'entrée et créer automatiquement de nouveaux exemples qui ressemblent aux données d'entraînement.
- **Transfert de style neuronal :** Les modèles neuronaux de transfert de style peuvent mélanger l'image du contenu et l'image du style et séparer le style du contenu.
- **Apprentissage par renforcement :** Les modèles d'apprentissage par renforcement entraînent des agents logiciels à atteindre leurs objectifs et à prendre des décisions dans un environnement virtuel. [1]

7 - L'augmentation des données pour le traitement du langage naturel (TAL):

L'augmentation des données n'est pas aussi populaire dans le domaine du traitement du langage naturel (TAL) que dans celui de la vision par ordinateur. L'augmentation des données textuelles est difficile, en raison de la complexité de la langue. Les méthodes courantes d'augmentation des données en TAL sont les suivantes :

- Opérations d'augmentation facile des données Easy Data Augmentation (EDA) : remplacement de synonymes, insertion de mots, échange de mots et suppression de mots.
- Retraduction : retraduction d'un texte de la langue cible vers sa langue d'origine.
- Incorporation de mots contextualisés. [1]

8- La différence avec les données synthétiques :

La génération de données synthétiques est une façon d'augmenter les données. Il existe d'autres approches (par exemple, apporter des modifications minimales aux données existantes pour créer de nouvelles données) pour l'augmentation des données.

9- Les avantages de l'augmentation des données :

Les avantages de l'augmentation des données sont les suivants

- Améliorer la précision des prédictions du modèle
  - en ajoutant plus de données de formation dans les modèles
  - la prévention de la pénurie de données pour de meilleurs modèles
  - la réduction de l'overfitting des données (une erreur en statistique, cela signifie qu'une fonction correspond trop étroitement à un ensemble limité de points de données) et la création de la variabilité dans les données
  - augmenter la capacité de généralisation des modèles
  - aider à résoudre les problèmes de déséquilibre des classes dans la classification.
- Réduire les coûts de collecte et d'étiquetage des données.
- Permet la prédiction d'événements rares.
- Prévient les problèmes de confidentialité des données.

10- Les défis de l'augmentation des données :

- Les entreprises doivent mettre en place des systèmes d'évaluation de la qualité des ensembles de données augmentées. À mesure que l'utilisation des méthodes d'augmentation des données augmente, l'évaluation de la qualité de leurs résultats sera nécessaire.
- Le domaine de l'augmentation des données doit développer de nouvelles recherches et études pour créer des données nouvelles/synthétiques avec des applications avancées. Par exemple, la génération d'images haute résolution à l'aide de GAN peut être un défi.
- Si un ensemble de données réel contient des biais, les données augmentées à partir de cet ensemble en contiendront également. Il est

donc important d'identifier la stratégie optimale d'augmentation des données.

11- Les application de l'augmentation des doonnées:

Les modèles de reconnaissance d'images et de langage naturel utilisent généralement des méthodes d'augmentation des données. Le domaine de l'imagerie médicale utilise également l'augmentation des données pour appliquer des transformations aux images et créer de la diversité dans les ensembles de données. [1]

12- Principales techniques d'augmentation des données :

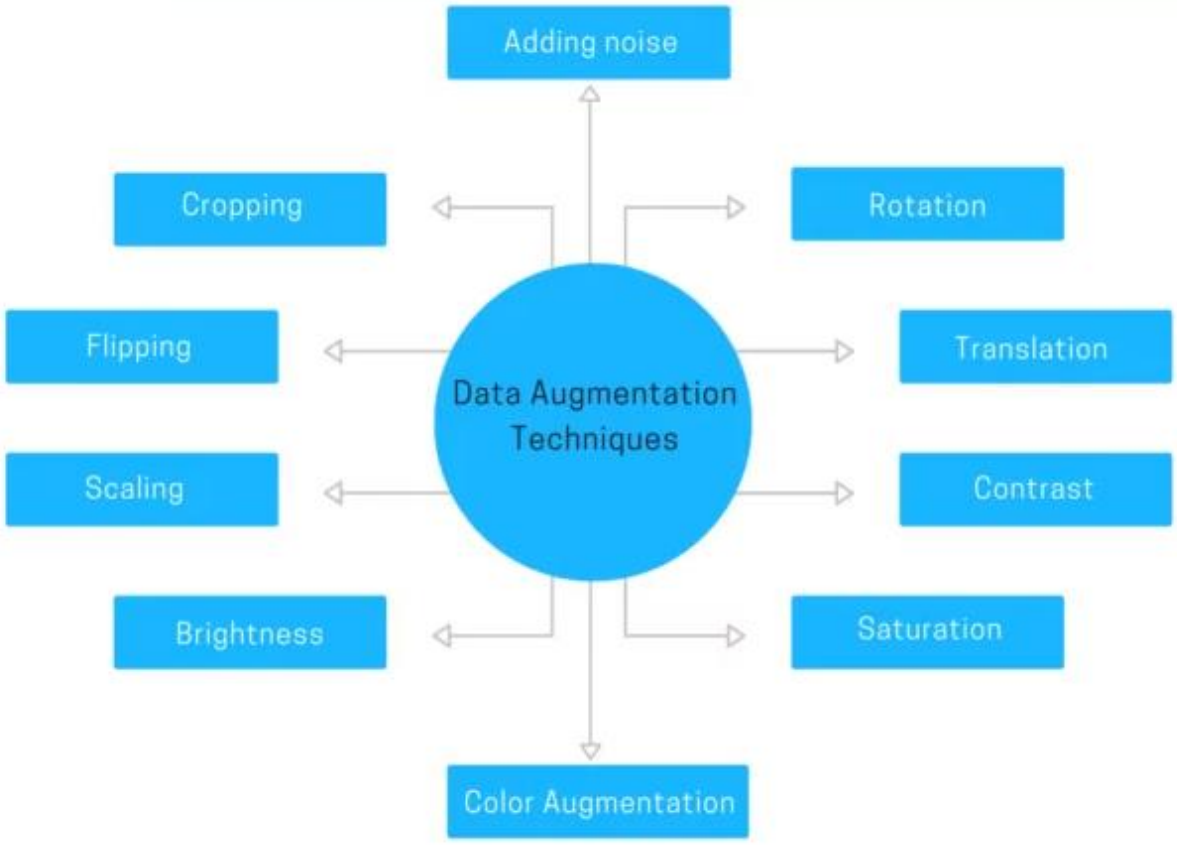


Figure 3: Principales techniques d'augmentation des données .

Les techniques d'augmentation des données génèrent artificiellement différentes versions d'un ensemble de données réelles pour en augmenter la taille. Les modèles de vision par ordinateur et de traitement du langage naturel (TAL) utilisent la stratégie d'augmentation des données pour faire face à la rareté des données et à leur diversité insuffisante.

Les algorithmes d'augmentation des données peuvent accroître la précision des modèles d'apprentissage automatique. Selon une expérience, un modèle d'apprentissage profond après augmentation de l'image est plus performant en termes de perte de formation (c'est-à-dire la pénalité pour une mauvaise prédiction) et de précision et de perte de validation et de précision qu'un modèle d'apprentissage profond sans augmentation pour une tâche de classification d'images. [20]

12-1- Techniques d'augmentation des données en vision par ordinateur:

Il existe des méthodes d'augmentation de l'espace géométrique et de l'espace couleur pour les images afin de créer une diversité d'images dans le modèle.

12-1-1- Ajouter du bruit:

Pour les images floues, l'ajout de bruit sur l'image peut être utile. Par "bruit poivre et sel", l'image ressemble à des points blancs et noirs.

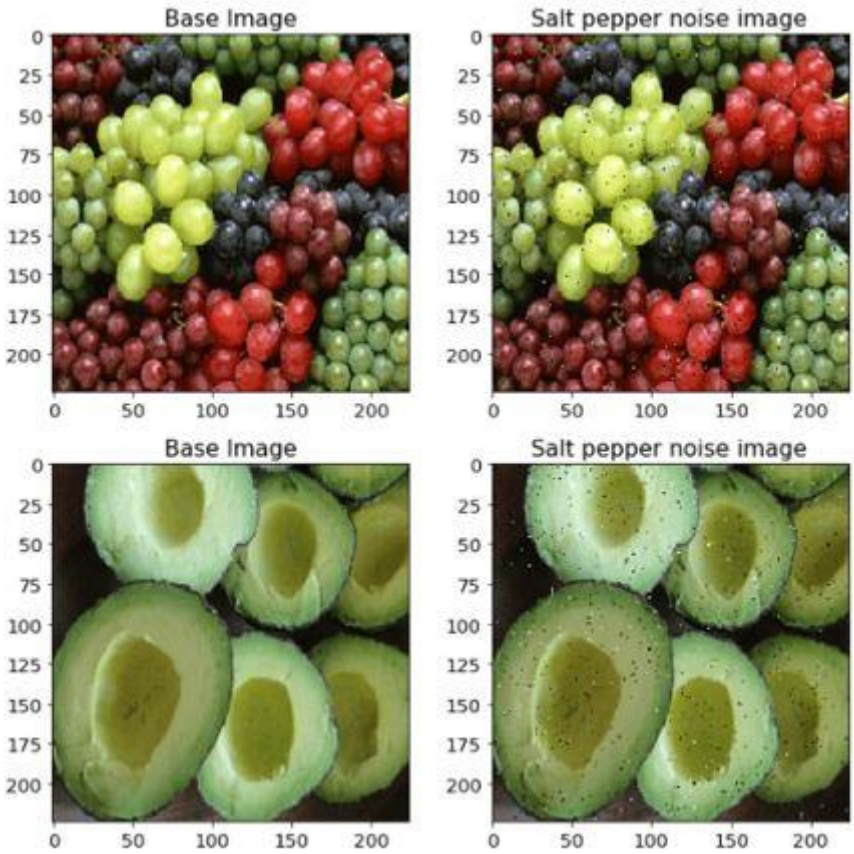


Figure 4: Exemple ajouter du bruit [5]

12-1-2- Culture:

Une section de l'image est sélectionnée, recadrée puis redimensionnée au format de l'image originale.

Figure 5: Exemple culture [6]

12-1-3- Flipping:

L'image est retournée horizontalement et verticalement. Le retournement réorganise les pixels tout en protégeant les caractéristiques de l'image. Le retournement vertical n'a pas de sens pour certaines photos, mais il peut être utile en cosmologie ou pour des photos microscopiques.

1:ORIGINAL IMAGE 2. HORIZONTAL FLIP 3.VERTICAL FLIP

Figure 6: Exemple Flipping [7]

12-1-4- Rotation:

L'image est tournée d'un degré entre 0 et 360 degrés. Chaque image pivotée sera unique dans le modèle.

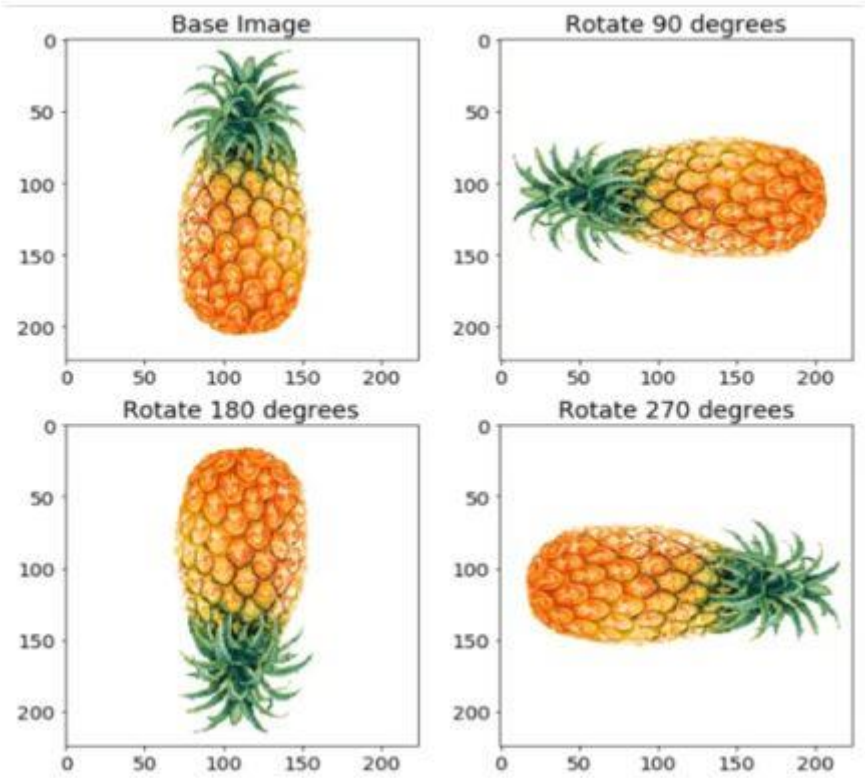


Figure 7: Exemple Rotation [5]

12-1-5- Mise à l'échelle:

L'image est mise à l'échelle vers l'extérieur et vers l'intérieur. Un objet dans la nouvelle image peut être plus petit ou plus grand que dans l'image originale grâce à la mise à l'échelle.

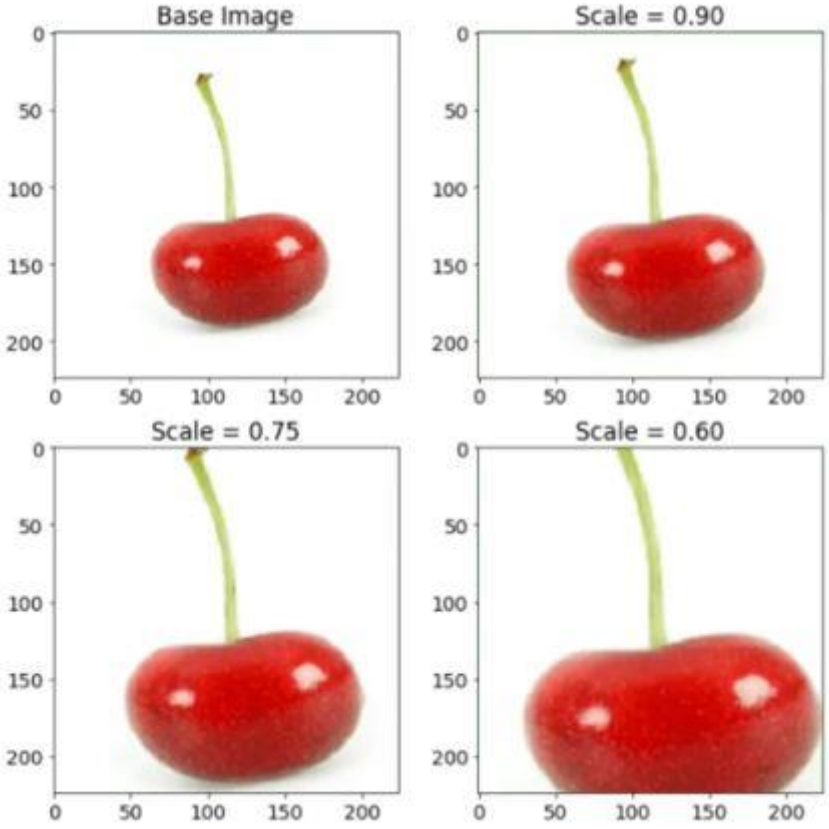


Figure 8: Exemple mise à l'échelle [9]

12-1-6- Traduction:

L'image est décalée dans différentes zones le long de l'axe des x ou de l'axe des y, de sorte que le réseau neuronal cherche partout dans l'image pour la capturer.

Figure 9: Exemple traduction [10]

12-1-7- Luminosité:

La luminosité de l'image est modifiée et la nouvelle image sera plus sombre ou plus claire. Cette technique permet au modèle de reconnaître l'image dans différents niveaux d'éclairage.

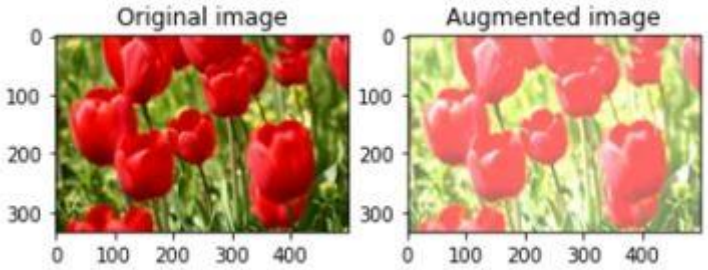


Figure 10: Exemple luminosité [11]

12-1-8- Contraste:

Le contraste de l'image est modifié et la nouvelle image sera différente en termes de luminance et de couleur. Le contraste de l'image suivante est modifié de manière aléatoire.

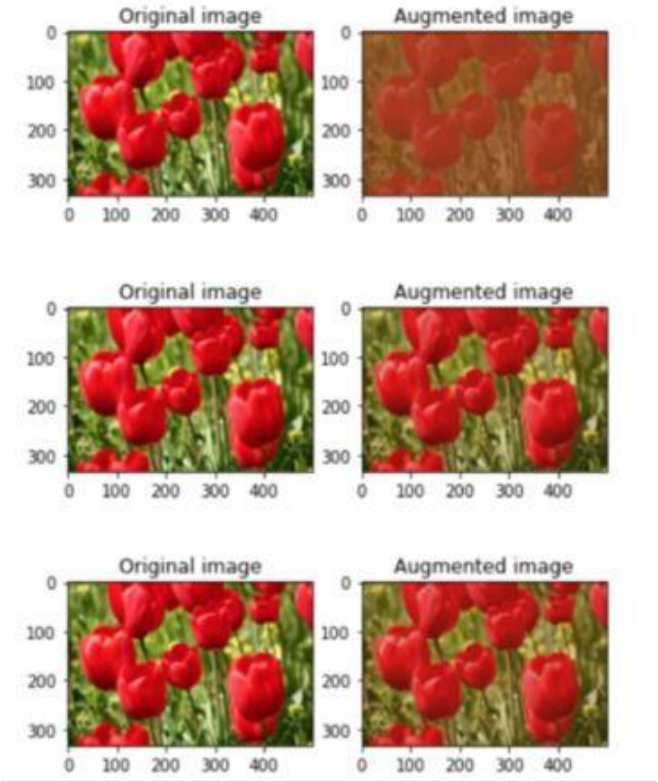


Figure 11: Exemple contraste [11]

12-1-9 -Augmentation de la couleur:

La couleur de l'image est modifiée par les nouvelles valeurs des pixels. Il existe un exemple d'image qui est en niveaux de gris.

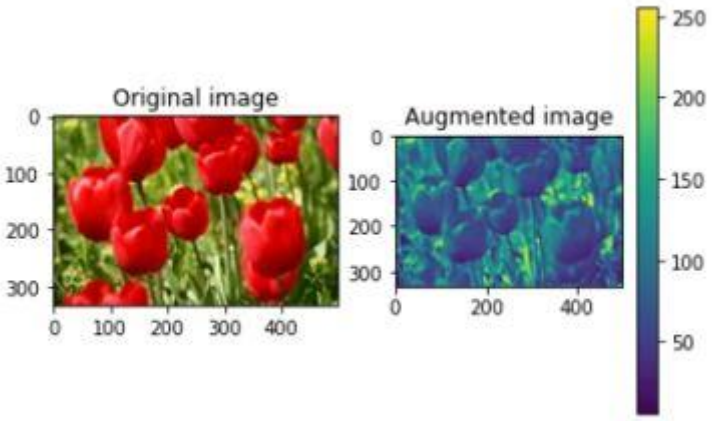


Figure 12: Exemple augmentation de la couleur [11]

12-1-10- Saturation:

La saturation est la profondeur ou l'intensité de la couleur dans une image. L'image suivante est saturée par la méthode d'augmentation des données.

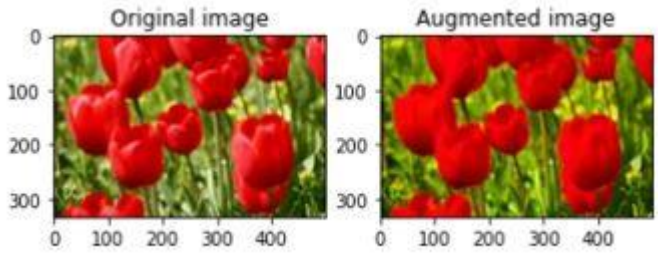


Figure 13: Exemple saturation [11]

13- Techniques d'augmentation des données dans les modèles de langage naturel:

Les techniques d'augmentation des données sont appliquées au niveau des caractères, des mots et des textes.

13-1-Méthodes d'augmentation facile des données (EDA):

Les méthodes facile des données (EDA) comprennent des transformations de texte faciles, par exemple un mot est choisi au hasard dans la phrase et remplacé par l'un de ses synonymes ou deux mots sont choisis et échangés dans la phrase. Les exemples de techniques EDA dans le traitement NLP sont

- Remplacement de synonymes
- Substitution de texte (basée sur les règles, sur le ML, sur le masque, etc.)
- Insertion aléatoire
- Échange aléatoire
- Suppression aléatoire
- Remaniement de mots et de phrases

13-2-Retraduction:

Une phrase est traduite dans une langue, puis la nouvelle phrase est retraduite dans la langue d'origine. Ainsi, des phrases différentes sont créées.

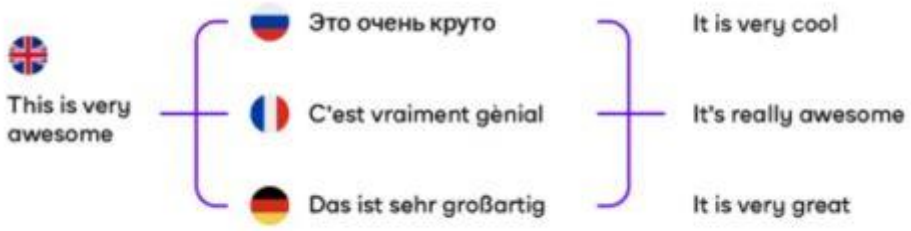


Figure 14: Exemple retraduction [15]

13-3- Génération de texte:

Un réseau adversarial génératif (GAN) est entraîné à générer du texte avec quelques mots.

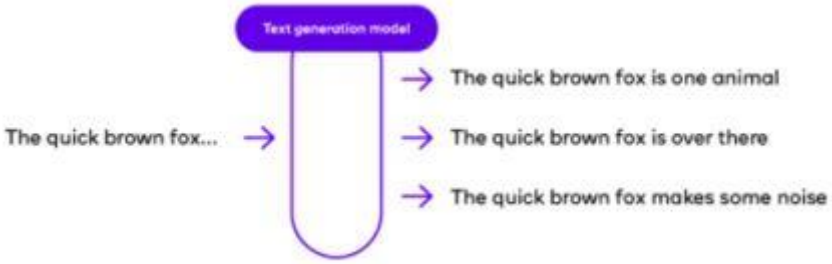


Figure 15: Exemple génération de texte [15]

14- Techniques d'augmentation des données pour les données audio:

Les méthodes d'augmentation des données audio comprennent le recadrage d'une partie des données, l'injection de bruit, le décalage temporel, le réglage de la vitesse, le changement de hauteur, le mélange de bruits de fond et le masquage de la fréquence.

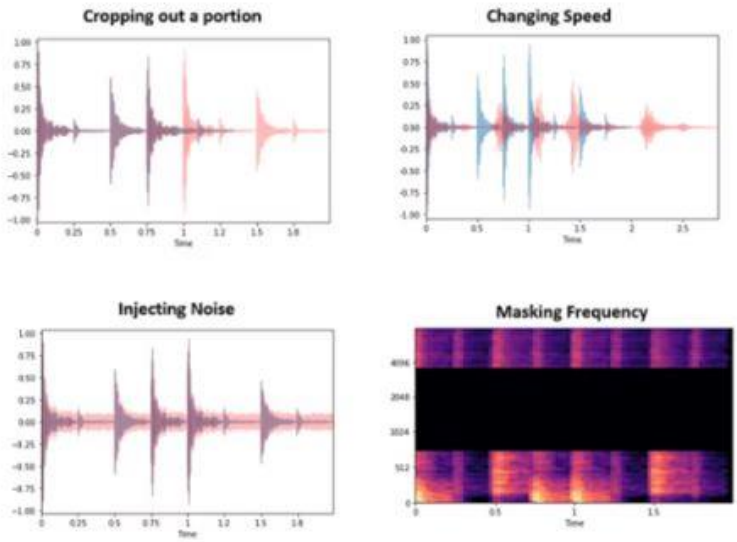


Figure 16: Exemple techniques d'augmentation des données pour les données audio [17]

15- Techniques avancées d'augmentation des données:

Les méthodes avancées d'augmentation des données sont couramment utilisées dans le domaine de l'apprentissage profond. Certaines de ces techniques sont:

- Formation contrariée
- Transfert de style neuronal
- l'augmentation basée sur les réseaux adversaires génératifs (GAN).

16- Data augmentation libraries:

Il existe des bibliothèques pour les développeurs, comme Albumentations, Augmentor, Imgaug, nlpaug, NLTK et spaCy. Ces bibliothèques comprennent des fonctions de transformation géométrique et de transformation de l'espace couleur, des filtres Kernel (c'est-à-dire des fonctions de traitement de l'image pour l'accentuation et le flou) et d'autres transformations de texte. Les bibliothèques d'augmentation des données utilisent différents cadres d'apprentissage profond, par exemple Keras, MxNet, PyTorch et TensorFlow. [20]

17- Les techniques d'augmentation des donnees pour l'apprentissage profond:

Nous avons examiné les applications de l'apprentissage profond ainsi que d'autres approches d'IA, comme l'apprentissage par renforcement, qui tendent à utiliser des structures d'apprentissage profond pour l'apprentissage.

17-1- Formation contradictoire:

L'entraînement contradictoire est une technique d'apprentissage automatique qui permet d'améliorer les performances d'un modèle en l'entraînant sur des tâches de prédiction difficiles à résoudre.

Dans la formation contradictoire, des exemples contradictoires sont créés et injectés dans l'ensemble de données de formation. Un modèle tente de tromper l'autre modèle en fournissant des entrées trompeuses (par exemple, en ajoutant du bruit aux échantillons de l'ensemble de données). Après l'attaque contradictoire, si le modèle classe l'entrée de manière erronée, les modèles

d'apprentissage profond sont formés à nouveau en utilisant ces exemples contradictoires pour améliorer les performances du modèle.

Cette approche rend les algorithmes d'apprentissage profond plus robustes. L'augmentation des données avec des exemples contradictoires enrichit les modèles d'apprentissage profond en fournissant des données diverses.

Un exemple contradictoire est présenté ci-dessous. Un bruit qui est difficilement compréhensible pour les gens est ajouté sur l'image du "panda". Après cette transformation, le modèle pense que l'image est "un gibbon".

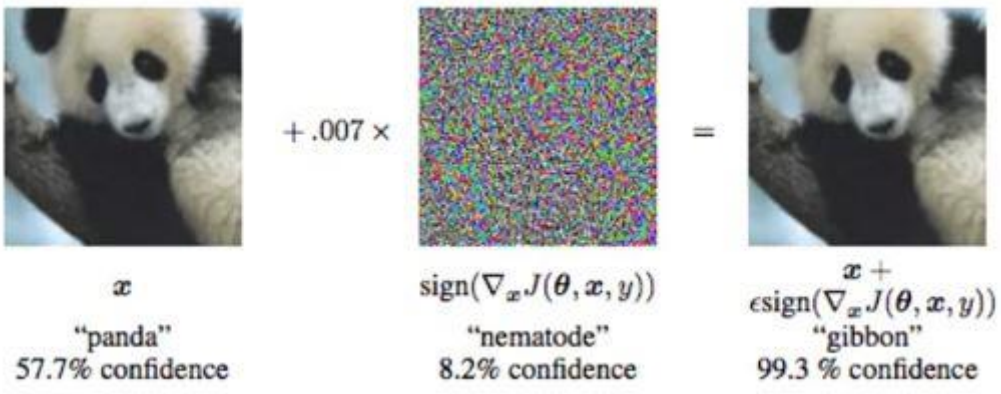


Figure 17: Exemple formation contradictoire [18]

L'apprentissage contradictoire est un sujet nouveau et peut s'avérer un processus coûteux. Il existe des zones d'ombre à son sujet, comme son avantage à réduire l'overfitting. La recherche dans ce domaine, non seulement pour les données d'image mais aussi pour les données textuelles et audio.

17-2- Augmentation basée sur les réseaux adversaires génératifs (GAN):

Les réseaux adversatifs génératifs ont la capacité d'augmenter les données pour l'entraînement des réseaux de neurones convolutifs (CNN). Les performances des modèles CNN dépendent de la quantité suffisante de données d'entraînement. Les GAN peuvent renforcer efficacement les capacités des CNN en générant de nouveaux échantillons dans l'ensemble des données d'entraînement par rapport aux techniques traditionnelles d'augmentation des données (rotations aléatoires, zoom, recadrage, etc.). Les GAN peuvent être utilisés pour différentes méthodes de génération d'images telles que le mélange d'images, la traduction d'images en images et la synthèse de textes en images.

17-3- Augmentation des données de méta-apprentissage:

Le méta-apprentissage ou "apprentissage par apprentissage" est un sous-domaine de l'apprentissage automatique. Les algorithmes de méta-apprentissage peuvent apprendre d'autres algorithmes d'apprentissage automatique. Dans le domaine de l'apprentissage profond, il s'agit de l'optimisation des réseaux neuronaux par d'autres réseaux neuronaux. Le méta-apprentissage peut être utilisé pour créer des éléments de haut niveau pour l'entraînement des réseaux neuronaux. Les algorithmes de méta-apprentissage peuvent avoir la capacité d'échantillonner des classes comme les images, de sorte que l'augmentation des données avec le méta-apprentissage peut fournir un avantage pour les modèles d'apprentissage profond. Une étude examine comment l'augmentation des données est utilisée pour accroître le nombre d'images pour chaque classe et créer de nouvelles classes dans l'ensemble de données.

17-4- Augmentation basée sur le transfert de style neuronal:

Le transfert de style neuronal est couramment utilisé dans les applications artistiques. Le style d'une image (par exemple, l'ambiance, la texture et la composition) créé dans les CNN est modifié grâce au transfert de style neuronal. Le style d'une image est mélangé au contenu d'une autre image. Le transfert de style neuronal peut être une méthode utile pour l'augmentation des données afin d'accroître la quantité de données de formation dans le domaine de l'apprentissage profond. Le transfert de style neuronal aide les transformations d'augmentation des données à décider du style tout en reproduisant de nouvelles images. Il existe quelques défis dans l'augmentation basée sur le transfert de style neuronal, tels que les efforts pour décider du style, la lenteur de l'exécution, la capacité de stockage et de mémoire élevée.

Figure 18: Exemple augmentation basée sur le transfert de style neuronal [19]

17-5- Augmentation basée sur l'apprentissage par renforcement:

L'apprentissage par renforcement learning (RL) peut être amélioré grâce à l'augmentation des données afin que les agents puissent être prêts à performer dans une grande variété de scénarios.

Si vous êtes prêt à utiliser l'augmentation des données dans votre entreprise, vous pouvez vous fier à nos listes de priorités pour les logiciels d'apprentissage profond et d'apprentissage automatique (ML). Une part importante de ces logiciels fournit des outils pour l'augmentation des données. [19]

18-Conclusion:

Dans cette partie dans le mémoire nous avons présenté quelque définitions de concepts de base dont l'objectif est d'expliquer le champs des termes utilisés, afin de rendre notre travail mieux compréhensible.

Chapitre II

1-Introduction:

Il existe plusieurs façons de rencontrer des problèmes en utilisant les méthodes statistiques paramétriques et non paramétriques traditionnelles. Par exemple, la taille de notre échantillon peut être trop petite pour que le théorème de la limite centrale garantisse que les moyennes de l'échantillon sont normalement distribuées, de sorte que les limites de confiance calculées classiquement peuvent ne pas être exactes. Nous ne souhaitons peut-être pas recourir à des tests non paramétriques de faible puissance. Nous intéresser à une statistique pour laquelle la distribution théorique sous-jacente est inconnue. Sans distribution, nous ne pouvons pas calculer les intervalles de confiance, les valeurs p ou les valeurs critiques.

Les méthodes de rééchantillonnage sont une solution à ces problèmes, et elles présentent plusieurs avantages. Elles sont flexibles et intuitives. Elles sont souvent plus puissantes que les méthodes non-paramétriques, et elles approchent et parfois dépassent la puissance des méthodes paramétriques. Deux d'entre elles (bootstrap, jackknife) ne font aucune hypothèse sur la forme de la distribution originale, si ce n'est que l'échantillon est un bon reflet de cette distribution, ce qui sera le cas si vous avez réuni un échantillon aléatoire grâce à un bon plan d'échantillonnage et si la taille de votre échantillon est suffisamment grande. Les deux mêmes peuvent également être appliquées à tout problème, même lorsqu'il n'existe pas de distribution théorique de la statistique. Elles sont moins sensibles aux valeurs aberrantes que les méthodes paramétriques. [1]

2- Objectif de rééchantillonnage:

Nous utilisons le rééchantillonnage parce que nous ne disposons que d'une quantité limitée de données - Les limites du temps et de l'économie, pour le moins. Qu'est-ce que le rééchantillonnage? Le rééchantillonnage est de prendre un échantillon, puis de prendre un échantillon de l'échantillon. Pourquoi faire ceci? Eh bien, cela vous permet de voir la variation qu'il aurait eu, cela vous permet d'avoir une compréhension différente de l'échantillon que vous avez prélevé. Disons que, par exemple, vous vouliez que 1 000 personnes prennent une enquête, mais vous n'en avez que 100. En sous-échantillonnant intelligemment les échantillons, nous pouvons obtenir une nouvelle distribution, ce qui nous permet d'en savoir un peu plus sur l'incertitude de l'échantillon, que nous supposons lié à l'incertitude de la population sous-jacente. [2]

-

Les méthodes de rééchantillonnage [3] tirent des échantillons des données observées afin de tirer certaines conclusions sur la population concernée. Deux méthodes de rééchantillonnage les plus populaires sont le jackknife et le bootstrap. Ces deux méthodes sont des exemples de méthodes statistiques non paramétriques. [3]

3-1-Bootstrap:

Le bootstrapping est une méthode statistique permettant d'estimer la distribution d'échantillonnage d'un estimateur par échantillonnage avec remplacement à partir de l'échantillon original, le plus souvent dans le but de dériver des estimations robustes des erreurs standard et des intervalles de confiance d'un paramètre de population comme une moyenne, une médiane, une proportion, un rapport de cotes, un coefficient de corrélation ou de régression. On l'a appelé le principe du plug-in[4], car il s'agit de la méthode d'estimation des

fonctionnelles d'une distribution de population en évaluant les mêmes fonctionnelles à la distribution empirique basée sur un échantillon.

Par exemple,[4] pour estimer la moyenne de la population, cette méthode utilise la moyenne de l'échantillon ; pour estimer la médiane de la population, elle utilise la médiane de l'échantillon ; pour estimer la ligne de régression de la population, elle utilise la ligne de régression de l'échantillon.

Elle peut également être utilisée pour construire des tests d'hypothèse. Elle est souvent utilisée comme une alternative robuste à l'inférence basée sur des hypothèses paramétriques lorsque ces hypothèses sont mises en doute, ou lorsque l'inférence paramétrique est impossible ou nécessite des formules très compliquées pour le calcul des erreurs standard. Les techniques de bootstrap sont également utilisées dans les transitions de mise à jour-sélection des filtres de particules, des algorithmes de type génétique et des méthodes de Monte Carlo de rééchantillonnage/reconfiguration connexes utilisées en physique computationnelle [5] [6]. Dans ce contexte, le bootstrap est utilisé pour remplacer les mesures de probabilité pondérées séquentiellement empiriques par des mesures empiriques. Le bootstrap permet de remplacer les échantillons à faible pondération par des copies des échantillons à forte pondération.

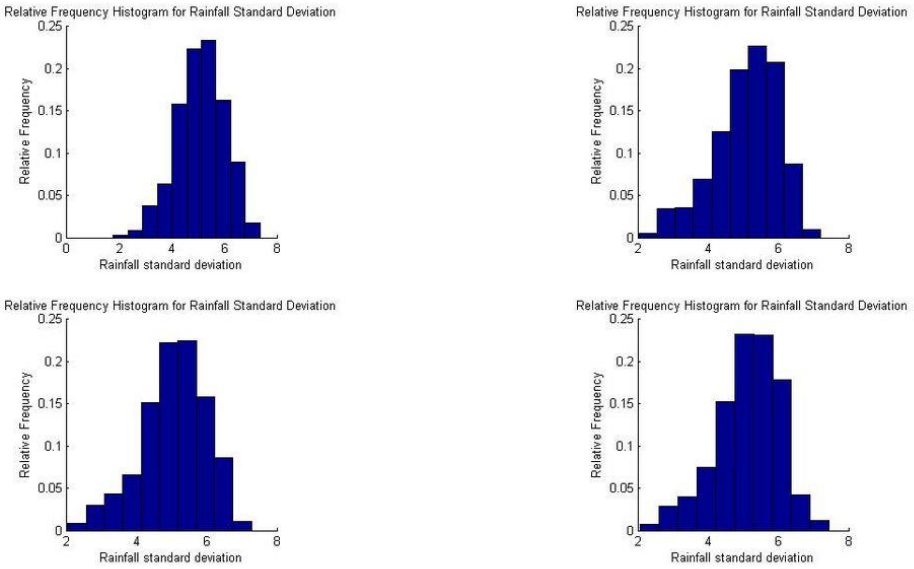


Figure 19: Exemple du principe du plug-in, la méthode du bootstrapping. [23]

3-1-1-Discussion:

3-1-1-1-Avantages:

Un grand avantage du bootstrap est sa simplicité. Il s'agit d'un moyen direct de dériver des estimations des erreurs standard et des intervalles de confiance pour des estimateurs complexes de la distribution, tels que les points percentiles, les proportions, les rapports de cotes et les coefficients de corrélation. Cependant, malgré sa simplicité, le bootstrap peut être appliqué à des plans d'échantillonnage complexes [7] Le bootstrap est également un moyen approprié pour contrôler et vérifier la stabilité des résultats. Bien que pour la plupart des problèmes, il soit impossible de connaître le véritable intervalle de confiance, le bootstrap est asymptotiquement plus précis que les intervalles standard obtenus à l'aide de la variance de l'échantillon et des hypothèses de normalité.[8] Le bootstrap est également une méthode pratique qui évite le coût de la répétition de l'expérience pour obtenir d'autres groupes de données d'échantillon.

3-1-1-2-Inconvénients:

Le bootstrapping dépend fortement de l'estimateur utilisé et, bien que simple, l'utilisation ignorante du bootstrapping ne donnera pas toujours des résultats asymptotiquement valides et peut conduire à une incohérence[9]. Bien que le bootstrapping soit (sous certaines conditions) asymptotiquement cohérent, il ne fournit pas de garanties générales sur les échantillons finis. Le résultat peut dépendre de l'échantillon représentatif. L'apparente simplicité peut cacher le fait que des hypothèses importantes sont faites lors de l'analyse bootstrap (par exemple, l'indépendance des échantillons ou une taille d'échantillon suffisamment grande) alors qu'elles seraient plus formellement énoncées dans d'autres approches. En outre, le bootstrap peut prendre beaucoup de temps et il n'existe pas beaucoup de logiciels pour le bootstrap car il est difficile à automatiser à l'aide des progiciels statistiques traditionnels[7].

3-1-1-3-Recommandations:

Les chercheurs ont recommandé d'utiliser davantage d'échantillons bootstrap à mesure que la puissance de calcul disponible augmentait. Si les résultats peuvent avoir des conséquences importantes dans le monde réel, il faut alors utiliser autant d'échantillons que possible, compte tenu de la puissance de calcul et du temps disponibles. L'augmentation du nombre d'échantillons ne peut pas augmenter la quantité d'informations dans les données d'origine ; elle ne peut que réduire les effets des erreurs d'échantillonnage aléatoires qui peuvent découler de la procédure bootstrap elle-même. De plus, il est prouvé que des nombres d'échantillons supérieurs à 100 entraînent des améliorations négligeables dans l'estimation des erreurs standard[10]. En fait, selon le développeur original de la méthode bootstrap, même en fixant le nombre d'échantillons à 50, il est probable que l'on obtienne des estimations d'erreurs standard assez bonnes[11].

Adèr et al. recommandent la procédure bootstrap pour les situations suivantes :[12]

- Lorsque la distribution théorique d'une statistique d'intérêt est compliquée ou inconnue. Puisque la procédure bootstrap est indépendante de la distribution, elle fournit une méthode indirecte pour évaluer les propriétés de la distribution sous-jacente à l'échantillon et les paramètres d'intérêt qui sont dérivés de cette distribution.
- Lorsque la taille de l'échantillon est insuffisante pour une inférence statistique directe. Si la distribution sous-jacente est bien connue, le bootstrapping permet de tenir compte des distorsions causées par l'échantillon spécifique qui peut ne pas être entièrement représentatif de la population.
- Lorsque des calculs de puissance doivent être effectués et qu'un petit échantillon pilote est disponible. La plupart des calculs de puissance et de taille d'échantillon dépendent fortement de l'écart type de la statistique concernée. Si l'estimation utilisée est incorrecte, la taille de l'échantillon requise sera également erronée. Une méthode pour avoir une impression de la variation de la statistique est d'utiliser un petit échantillon pilote et de réaliser un bootstrapping sur celui-ci pour avoir une impression de la variance.

Cependant, Athreya a montré [13] que si l'on effectue un bootstrap naïf sur la moyenne de l'échantillon lorsque la population sous-jacente n'a pas de variance finie (par exemple, une distribution en loi de puissance), alors la distribution bootstrap ne convergera pas vers la même limite que la moyenne de l'échantillon. Par conséquent, les intervalles de confiance sur la base d'une simulation de Monte Carlo du bootstrap pourraient être trompeurs. Athreya déclare que "à moins d'être raisonnablement sûr que la distribution sous-jacente n'est pas à queue lourde, on devrait hésiter à utiliser le bootstrap naïf".

3-1-2- Types de schéma d'amorçage:

- Rééchantillonnage de cas.
  - Estimation de la distribution de la moyenne de l'échantillon.
  - Régression.
- Bootstrap bayésien.
- Amorçage en douceur.
- Bootstrap paramétrique.
- Rééchantillonnage des résidus.
- Bootstrap de régression par processus gaussien.
- Le bootstrap sauvage.
- Amorce de bloc.
  - Série chronologique : Bootstrap par blocs simples.
  - Série chronologique : Bootstrap à blocs mobiles.
  - Séries chronologiques : Bootstrap à entropie maximale.
  - Données en grappe : bootstrap par blocs.

3-1-3-Méthodes pour améliorer l'efficacité des calculs:

- bootstrap de Poisson.
- Sac de petites bretelles. [14]

3-2-Jackknife:

Le jackknife est une méthode utilisée pour estimer la variance et le biais d'une grande population. Il s'agit de la plus ancienne méthode de rééchantillonnage. Elle implique une stratégie "leave-one-out" de l'estimation d'un paramètre (par exemple, la moyenne) dans un ensemble de données de N observations (ou

enregistrements). Idéalement, N - 1 modèles sont construits sur l'ensemble de données avec différents facteurs exclus de chaque modèle. Les estimations de tous les modèles sont ensuite agrégées en une seule estimation du paramètre. Le jackknife devient difficilement calculable lorsque $N \to \infty$. Le succès du jackknife dans le monde universitaire et de la recherche a conduit au développement de la méthode bootstrap.[15]

3-2-1-Estimation moyenne:

L'estimateur jackknife d'un paramètre est trouvé en excluant systématiquement chaque observation d'un ensemble de données et en calculant l'estimation du paramètre sur les observations restantes, puis en agrégeant ces calculs.

3-2-2-Estimation du biais d'un estimateur:

La technique du jackknife peut être utilisée pour estimer (et corriger) le biais d'un estimateur calculé sur l'ensemble de l'échantillon.

3-2-3-Estimation de la variance d'un estimateur:

La technique du jackknife peut également être utilisée pour estimer la variance d'un estimateur calculé sur l'ensemble de l'échantillon. [16]

4-Comparaison entre bootstrap et du jackknife:

Les deux méthodes, le bootstrap et le jackknife, estiment la variabilité d'une statistique à partir de la variabilité de cette statistique entre les sous-échantillons, plutôt qu'à partir d'hypothèses paramétriques. Pour le jackknife plus général, le jackknife à suppression d'observations, le bootstrap peut être considéré comme une approximation aléatoire de celui-ci. Les deux donnent des résultats numériques similaires, c'est pourquoi chacun peut être considéré comme une approximation de l'autre. Bien qu'il existe d'énormes différences théoriques dans leurs conceptions mathématiques, la principale différence pratique pour les utilisateurs de statistiques est que le bootstrap donne des résultats différents

lorsqu'il est répété sur les mêmes données, alors que le jackknife donne exactement le même résultat à chaque fois. Pour cette raison, le jackknife est populaire lorsque les estimations doivent être vérifiées plusieurs fois avant d'être publiées. D'autre part, lorsque cette fonction de vérification n'est pas cruciale et qu'il est intéressant de ne pas avoir un nombre mais juste une idée de sa distribution, le bootstrap est préféré.

L'utilisation du bootstrap ou du jackknife peut dépendre davantage des aspects opérationnels que des préoccupations statistiques d'une enquête. Le jackknife, utilisé à l'origine pour la réduction du biais, est une méthode plus spécialisée et n'estime que la variance de l'estimateur ponctuel. Cela peut suffire pour l'inférence statistique de base (par exemple, les tests d'hypothèse, les intervalles de confiance). Le bootstrap, quant à lui, estime d'abord la distribution entière (de l'estimateur ponctuel), puis calcule la variance à partir de celle-ci. Bien que cette méthode soit puissante et facile, elle peut devenir très exigeante en termes de calcul.

"Le bootstrap peut être appliqué à la fois aux problèmes d'estimation de la variance et de la distribution. Cependant, l'estimateur de variance bootstrap n'est pas aussi bon que le jackknife ou l'estimateur de variance par réplication répétée équilibrée (BRR) en termes de résultats empiriques. De plus, l'estimateur de variance bootstrap nécessite généralement plus de calculs que le jackknife ou le BRR. Ainsi, le bootstrap est principalement recommandé pour l'estimation de la distribution "[17].

Il y a une considération spéciale avec le jackknife, particulièrement avec le jackknife à 1 observation supprimée. Il ne devrait être utilisé qu'avec des statistiques lisses et différentiables (par exemple, les totaux, les moyennes, les proportions, les rapports, les rapports impairs, les coefficients de régression, etc. ; pas avec les médianes ou les quantiles). Cela pourrait devenir un inconvénient

pratique. Cet inconvénient est généralement l'argument en faveur du bootstrap par rapport au jackknife.

En général, le jackknife est plus facile à appliquer à des plans d'échantillonnage complexes que le bootstrap. Les plans d'échantillonnage complexes peuvent comporter une stratification, des étapes multiples (regroupement), des poids d'échantillonnage variables (ajustements pour non-réponse, calibrage, post-stratification) et des plans d'échantillonnage à probabilité inégale. Les aspects théoriques du bootstrap et du jackknife peuvent être trouvés dans Shao et Tu (1995),[18] tandis qu'une introduction de base est comptabilisée dans Wolter (2007).[19] L'estimation bootstrap du biais de prédiction du modèle est plus précise que les estimations jackknife avec des modèles linéaires tels que la fonction discriminante linéaire ou la régression multiple.[20]

5-Tests de permutation:

Un test de permutation (également appelé test de ré-randomisation) est un test d'hypothèse statistique exact faisant appel à la preuve par contradiction dans lequel la distribution de la statistique de test sous l'hypothèse nulle est obtenue en calculant toutes les valeurs possibles de la statistique de test sous des réarrangements possibles des données observées. Les tests de permutation sont donc une forme de rééchantillonnage.

Les tests de permutation peuvent être compris comme des tests de données de substitution où les données de substitution sous l'hypothèse nulle sont obtenues par des permutations des données originales[21].

Les tests de permutation ne doivent pas être confondus avec les tests aléatoires[22].

6-Conclusion:

Dans cette partie du mémoire, nous avons fourni une définition générale des stratégies proposées visant à donner un large aperçu et une vision pour mieux faire comprendre notre travail.

Chapitre III

1.Introduction:

Les applications de l'apprentissage automatique, en particulier dans le domaine de l'apprentissage profond, continuent de se diversifier et d'augmenter rapidement. Les techniques d'augmentation des données peuvent être un bon outil pour relever les défis auxquels le monde de l'intelligence artificielle est confronté.

L'augmentation des données permet d'améliorer les performances et les résultats des modèles d'apprentissage automatique en ajoutant des exemples nouveaux et différents aux ensembles de données d'entraînement. Si l'ensemble de données d'un modèle d'apprentissage automatique est riche et suffisant, le modèle est plus performant et plus précis.

Pour les modèles d'apprentissage automatique, la collecte et l'étiquetage des données peuvent être des processus épuisants et coûteux. La transformation des ensembles de données à l'aide de techniques d'augmentation des données permet aux entreprises de réduire ces coûts opérationnels.[1]

Dans notre travail, développer suggéré deux techniques (Bootstrap, Jackknife) pour augmenter le nombre de données pour améliorer l'éducation automatique qui est augmentée par le biais de données brutes MNIST, puis évaluées via le modèle SVM et comparé les résultats obtenus avant et après augmenter le nombre de données et avec la technique de shift.

2.L'architecture :

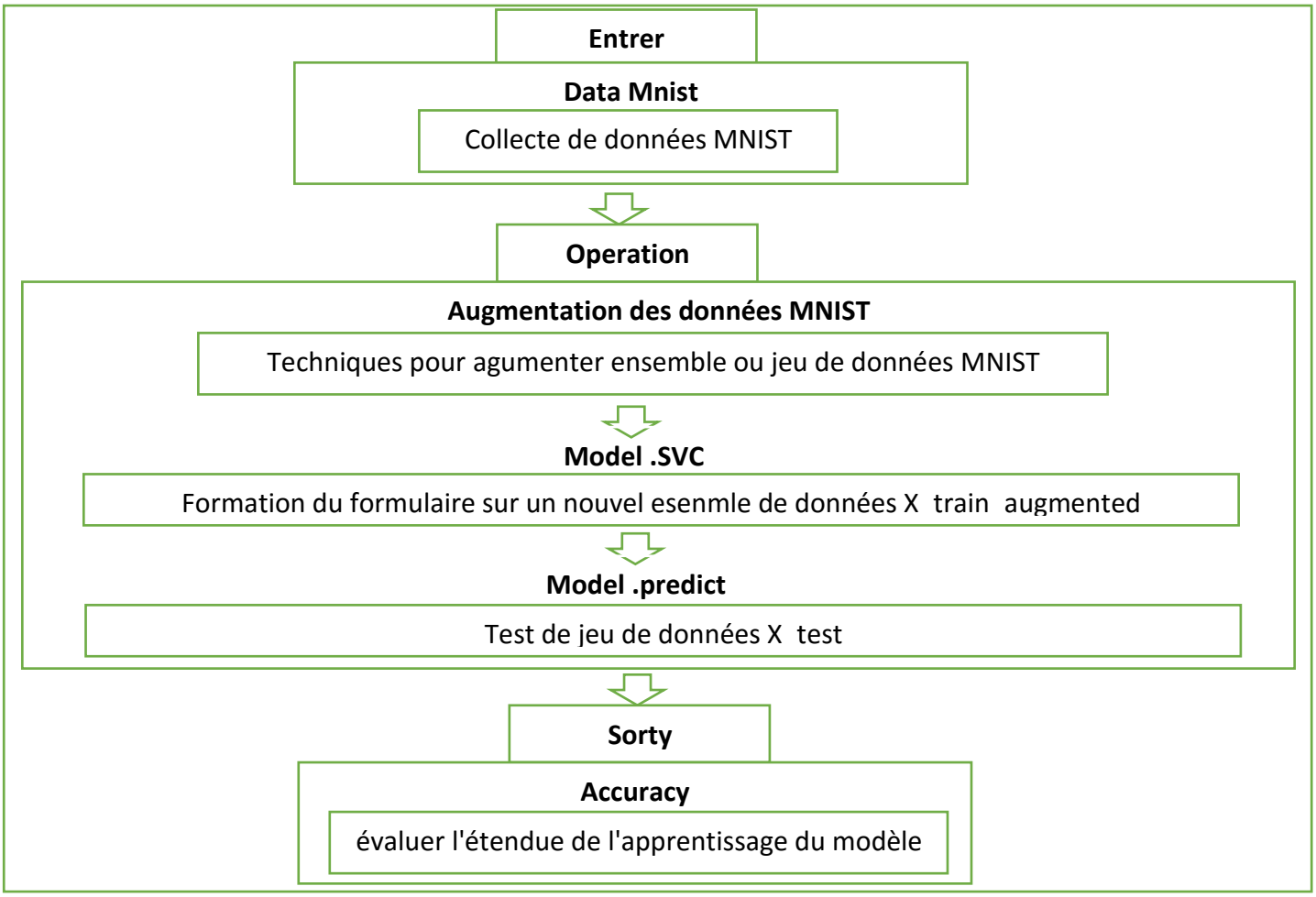
2-1- Choix mnist:

Il s'agit d'une ensemble ou jeu de données de chiffres manuscrits (0-9), avec laquelle vous pouvez tester quelques algorithmes d'apprentissage automatique.

De nombreuses bibliothèques d'apprentissage automatique, comme sklearn en python, offrent déjà un accès facile à l'ensemble de données MNIST.

2-2-L'architecture générale:

La figure ci-dessus montre les différentes étapes du système développé suivi par les détails d'implémentation pour les principales taches.



L'architecture générale

2-2-1- Ensemble ou jeu de données MNIST:

La ensemble ou jeu de données MNIST (Modified National Institute of Standards and Technology database[2]) est une grande ensemble ou jeu de données de chiffres manuscrits qui est couramment utilisée pour l'entraînement de divers systèmes de traitement d'images[3][4]. La ensemble ou jeu de données est également largement utilisée pour l'entraînement et le test dans le domaine de l'apprentissage automatique[5][6]. Elle a été créée en "remixant" les échantillons des ensembles de données originaux du NIST[7]. [Les créateurs ont estimé que, puisque l'ensemble de données d'entraînement du NIST provenait d'employés du Bureau du recensement américain, tandis que l'ensemble de données de test provenait de lycéens américains, il n'était pas bien adapté aux expériences d'apprentissage automatique[8] De plus, les images en noir et blanc du NIST ont été normalisées pour entrer dans une boîte de délimitation de 28x28 pixels et anticrénelées, ce qui a introduit des niveaux de gris[8].

La ensemble ou jeu de données MNIST contient 60 000 images d'entraînement et 10 000 images de test[9]. La moitié de l'ensemble d'entraînement et la moitié de l'ensemble de test proviennent de l'ensemble de données d'entraînement du NIST, tandis que l'autre moitié de l'ensemble d'entraînement et l'autre moitié de l'ensemble de test proviennent de l'ensemble de données de test du NIST[10].

2-2-2-Méthode d'augmentation de données:

Dans cette partie, nous parlons de la façon d'augmenter l'ensemble de données des nombres MNIST et d'utiliser une nouvelle technique Bootstrap.

2-2-2-1- Bootstrap:

En prenant 3 matrices avec 28 * 28 dimensions classées via la fonction d'index afin qu'elles soient droit, c'est-à-dire qu'elle a le même lable.

Lable = 0 Lable = 0 Lable = 0 ⟶ Lable = 0 ;…

Matrice(1) Matrice(2) Matrice(3) ⟶ Matrice(x);…

…..

…..

…..

Lable = 9 Lable = 9 Lable = 9 ⟶ Lable = 9 ;…

Matrice(1) Matrice(2) Matrice(3) ⟶ Matrice(y);…

Prenez le même numéro de cellule sur toutes les 3 matrices, puis choisissez un nombre aléatoire de trois nombres, via la fonction aléatoire. Ensuite, mettez-le dans le même numéro de cellule à partir de la matrice à créer.

Le processus de répétition est fait pour la préparation de lable jusqu'à la fin de la série de classification des nombres. Par exemple: lable = 0 toutes les 3 matrices lables sont prises = 0 ensemble jusqu'à la chaîne de matrices classifiées et ainsi de suite pour tous les chiffres jusqu'à atteindre le numéro = 9.

2-2-2-1-1-Image:

Index pexil: (1,1) (1,2) (1,3) ..... (1,28)

120				

Lable=0

(1,1) (1,2) (1,3) ..... (1,28)

118				

Lable=0

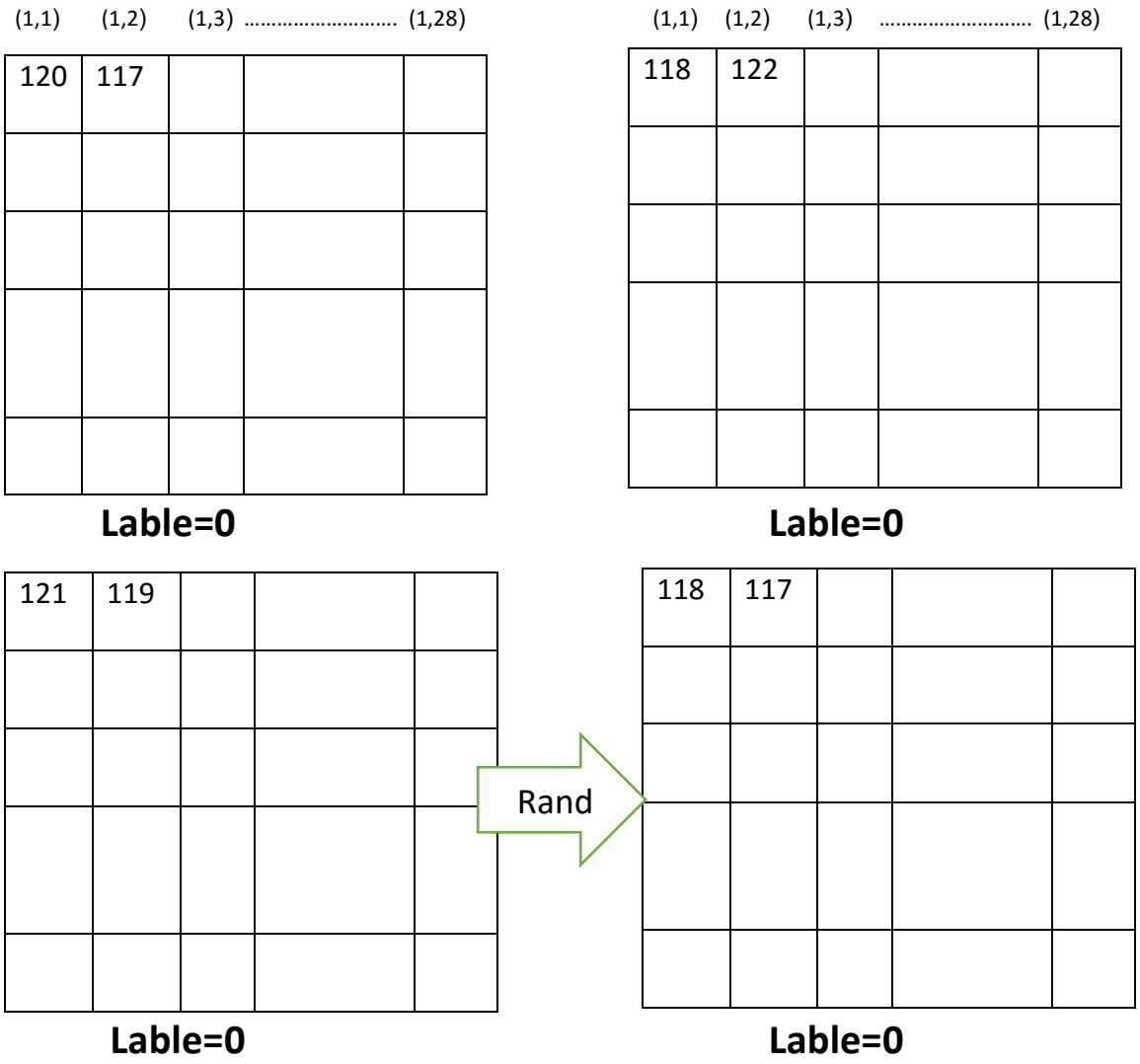
121				

Lable=0

Rand

118				

Lable=0



2-2-2-2-Jackknife:

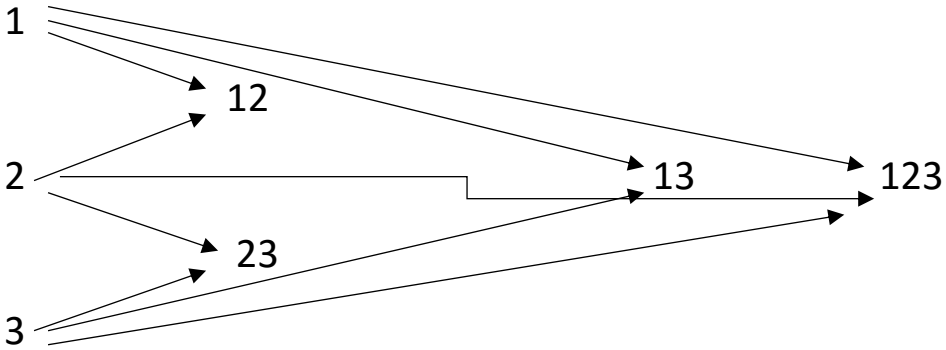
Prenez une matrice de 28 * 28 pixels.

Prenez le même numéro de cellule à partir de n (connectez un numéro de matrice) et il peut être formé, puis divisant le nombre de matrices prises.

Exemple: 3 matrices.

Pour calculer le nombre de possibilités de nouvelles matrices pour 3 ordonnances (photos).

Un arbre de possibilité sans répétition:



N : Le nombre de matrice.

K : Le nombre d'éléments tirés du groupe N.

K<=N

$$\frac{N!}{K!\,(N-K)!}$$

Perspective K=2 :

$$\frac{3!}{2!\,(3-2)!} = 3$$

Nombre de nouvelles possibilités : 12, 13, 23.

1 : numéro d'image.

(11) : numéro de cellule.

$$1 + 2 = \frac{5 + 9}{2} = 7 \quad m$$

Index pixel: (1,1) (1,2) (1,1) (1,2) (1,1) (1,2)

5	

9	

7	

$$1 \quad + \quad 3 \quad = \frac{1_{(11)} \quad +3_{(11)}}{2} \quad = y \quad w$$

3	

4	

y	

$$2 \quad + \quad 3 \quad = \frac{2_{(11)} \quad +3_{(11)}}{2} \quad = t \quad g$$

5	

6	

t	

Perspective K=3 :

$$\frac{3!}{3!\,(3-3)!} = 1$$

Nombre de nouvelles possibilités : 123

$$1 \quad + \quad 2 \quad + \quad 3 \quad = \frac{1_{(11)} \quad +2_{(11)} \quad +3_{(11)}}{3} \quad = l \quad f$$

7	

8	

9	

l	

….

….

….

….

….

$$1 \quad + \quad 2 \quad + \quad 3 \quad = \quad \frac{1_{(22)} \quad +2_{(22)} \; +3_{(22)}}{3} \quad = u \quad f$$

10	11
12	13

14	15
16	17

18	19
20	21

l	…
…	u

2-2-3-Model. SVC:

Le SVM offre une précision très élevée par rapport à d'autres classificateurs tels que la régression logistique et les arbres de décision. Il est connu pour son astuce de noyau pour gérer les espaces d'entrée non linéaires. Il est utilisé dans une variété d'applications telles que la détection des visages, la détection des intrusions, la classification des courriels, des articles de presse et des pages Web, la classification des gènes et la reconnaissance de l'écriture manuscrite.[11]

2-2-4-Model.predict:

La modélisation prédictive est le processus qui consiste à utiliser des résultats connus pour créer, traiter et valider un modèle qui peut être utilisé pour prévoir des résultats futurs. Il s'agit d'un outil utilisé dans l'analyse prédictive, une technique d'exploration de données qui tente de répondre à la question "que pourrait-il se passer à l'avenir ?". [12]

2-2-5-Accuracy:

La précision peut être définie comme étant, en moyenne, la distance entre vos mesures ou vos résultats et votre objectif. En d'autres termes, la précision est la mesure dans laquelle la moyenne des mesures s'écarte de la valeur réelle. [13]

3.Préparation de l'environnement de travail:

3-1- Environnement et Outils du développement:

3-1-1- Langage python:

Python est un langage de programmation multi-paradigme. Il favorise la programmation impérative structurée, et orientée objet. Il est doté d'un typage dynamique fort. développé par « Python Software Foundation ». Son implémentation a débuté en décembre 1989 [14]. Il à plusieurs versions Python 2. x ou Python 3. x et nous allons utiliser notre dernière version de Python 3.6.5 [14].

3-1-2- choix langage python :

Python propose des bibliothèques de haute qualité, les plus utilisées en intelligence artificielle et en apprentissage automatique. Il supporte à la fois TensorFlow, Tweepy et Python Pyspark. Dans le domaine de l'intelligence artificielle, nous trouvons que les bibliothèques les plus célèbres sont constituées du python, telles que Scikit-Learn et Pandas et NumPy [15].

3-1-3-Definition colab:

Colaboratory, ou "Colab" en abrégé, est un produit de Google Research. Colab permet à quiconque d'écrire et d'exécuter du code python arbitraire via le navigateur, et est particulièrement bien adapté à l'apprentissage automatique, à l'analyse de données et à l'éducation. Plus techniquement, Colab est un service de carnet de notes Jupyter hébergé qui ne nécessite aucune installation pour être utilisé, tout en offrant un accès gratuit aux ressources informatiques, y compris aux GPU. [16]

3-2-Nouveau module :

Nous allons également importer tensorflow ici sous le nom de tf.

Tensorflow est une autre bibliothèque d'apprentissage automatique avec de nombreuses utilisations. Pour l'instant, nous l'utilisons uniquement pour obtenir l'ensemble de données mnist.

Le jeu de données mnist est un grand ensemble d'images de chiffres manuscrits. Il y a 60.000 images d'entraînement et 10.000 images de test !

```
import pandas as pd
import numpy as np
from sklearn import svm
from sklearn import metrics
import tensorflow as tf
```

Il existe plusieurs façons d'obtenir le jeu de données mnist, mais nous allons l'obtenir à partir de Keras de TensorFlow. Kera est une bibliothèque de réseaux neuronaux que nous pouvons exécuter avec TensorFlow. Elle contient également de nombreux jeux de données tels que le jeu de données mnist que nous allons utiliser.

Heureusement, l'utilisation de ce jeu de données présente deux avantages majeurs :

Le jeu de données est déjà propre et prêt à être utilisé.

3-3-Data MNIST (Collecte de données MNIST):

Nous pouvons charger le jeu de données directement dans l'ensemble d'entraînement et l'ensemble de test, chacun avec les caractéristiques et les étiquettes séparées.

Nous allons utiliser la fonction load_data() de Keras qui renvoie deux tuples.

```
(X_train, y_train), (X_test, y_test) = tf.keras.datasets.mnist.load_data()
```

Le premier tuple contient les caractéristiques de l'ensemble de formation (X_train) et les étiquettes de l'ensemble de formation (y_train). Le second tuple contient les caractéristiques de l'ensemble de test (X_test) et les étiquettes de l'ensemble de test (y_test).

Ceci est similaire à la façon dont train_test_split() retourne ses tableaux pour le test et l'apprentissage.

X_train et X_test sont nos tableaux d'images tandis que y_train et y_test sont nos tableaux d'étiquettes pour chaque image.

Par exemple, si l'image montre un 7 écrit à la main, alors l'étiquette sera le 7 entier.

3-3-1-Corriger les données:

Ainsi, bien que l'ensemble de données soit propre, nous devons tout de même procéder à un traitement des données, comme tout bon spécialiste des données.

Plus précisément, nous devons reformater notre tableau X_train et notre tableau X_test car ils n'ont pas la forme correcte.

Imprimons la forme de tous nos tableaux pour voir à quoi ils ressemblent actuellement. Pour ce faire, nous allons utiliser l'attribut shape:

```
X_train shape (60000, 28, 28)
y_train shape (60000,)
X_test shape (10000, 28, 28)
y_test shape (10000,)
```

Vous pouvez voir ici que pour les ensembles de formation, nous avons 60 000 éléments et que les ensembles de test ont 10 000 éléments.

y_train et y_test n'ont que des formes unidimensionnelles car il s'agit simplement des étiquettes de chaque élément.

X_train et X_test ont des formes tridimensionnelles car ils ont une

largeur et une hauteur (28x28 pixels) pour chaque élément.

3-3-2-Le problème de la formep:

Chaque fois que nous ajustons notre modèle, nous devons passer deux arguments à la fonction fit() :

X : Données d'apprentissage de forme (n_échantillons, n_caractéristiques)

y : Valeurs des étiquettes d'apprentissage de la forme (n_échantillons, n_labels)

Chaque fois que nous prédisons avec notre modèle, nous devons passer un argument dans la fonction predict():

X : échantillons de test de la forme (n_échantillons, n_caractéristiques)

Fondamentalement, les algorithmes d'apprentissage supervisé de Scikit-learn s'attendent à ce que les données soient stockées dans des tableaux bidimensionnels.

Heureusement, les tableaux 1D tels que nos étiquettes dans y_train et y_test, sont automatiquement remodelés pour devenir des tableaux 2D. Ils seront donc transformés de (n_échantillons,) en (n_échantillons, 1).

Cependant, nos caractéristiques sont toujours en 3 dimensions avec une forme (n_samples, 28, 28). Nous devons les remodeler pour qu'elles ne soient plus que bidimensionnelles.

Pour ce faire, nous allons modifier les données des pixels pour qu'elles ne soient pas un tableau 2D de hauteur et de largeur, mais un tableau long de tous les pixels. Par exemple, 28 pixels sur 28 pixels deviendront 784 pixels (28 au carré).

Remodelons X_train et X_test. Rappelez-vous que X_train a 60 000 éléments, chacun avec 784 pixels au total, et deviendra donc la forme (60000, 784).

Tandis que X_test a 10 000 éléments, chacun avec 784 pixels au total, ce qui donnera la forme (10000, 784).

```
X_train = X_train.reshape(60000, 784)
X_test = X_test.reshape(10000, 784)
```

3-3-3-Comment préparer des données:

Nous récupérons la collecte de données MNIST à travers:

```
x_train_augmeted=[image for image in x_train]
y_train_augmeted=[image for image in y_train]
```

3-4-Augmentation des données mnist (Techniques pour augmenter les ensembles ou jeu de données MNIST):

Ici, nous utilisons les fonctions nouvellement développées pour augmenter les glandes de données.

Début

Data_mnist

Def methode index (Data_mnist.length(),nombre_recherche)

Por i=0 à Data_mnist. .length():

Liste= recherche index de nombre_recherche

Fin por

Returne liste;

Fin

Index Graduates Le numéro dans la collecte des données MNIST, puis nous renvoyons un tableau d'index.

Def methode_ Bootstrap(arr1,arr2,arr3)

Por i a 28

Por j a 28

arr4[0]= arr1[i][j]

arr4[1]= arr2[i][j]

arr4[2]= arr3[i][j]

fin por

Tab[i][j]= random(arr4,size=1)

Fin por

Returne tab;

Il faut au hasard un seul numéro dans tous les pixels 3

Puis ajoutez de nouvelles données au groupe de données cloné (y_train_augmeted).

```
X_train_augmented = np.append(X_train_augmented,[ tab],axis=0)
y_train_augmented = np.append(y_train_augmented,y)
```

*Nous augmentons data augmentation Matrice de tab Pour l'ensemble de données X_train_augmented .

*Nous augmentons l'identifiant (Y) Pour l'ensemble de donné y_train_augmented.

3-5- Model. SVC (Formation du formulaire sur un nouvel ensemble de données X_train_augmented):

Maintenant que nos tableaux sont en bon état, nous pouvons créer notre modèle et commencer l'apprentissage.

Créez le modèle de classification par vecteur de support en utilisant svm.SVC(). Ensuite, ajustez le modèle avec l'ensemble X_train_ augmented et l'ensemble y_train_augmented**.**

```
model = svm.SVC()
model.fit(X_train_augmented,y_train_augmented)
```

3-6- Model.predict(Test de jeu de données X_test):

Notre modèle est maintenant entraîné et prêt à être testé sur un nouvel ensemble de données. Utilisons notre ensemble d'images de test pour que notre modèle puisse prédire leurs étiquettes.

```
y_pred = model.predict(X_test)
```

3-7-Accuracy (Évaluer l'étendue de l'apprentissage du modèle):

Nous pouvons examiner la précision du modèle à l'aide de la fonction metrics.accuracy_score().

```
acc = metrics.accuracy_score(y_test, y_pred)
print('\nAccuracy: ', acc)
```

4-Comparaison des résultats avec d'autres technique:

methode	X_test	Y_test	X_train	y_train	X_train_augmented	y_train_augmented	Accuracy
Bootstrap	10000	10000	#60000	#10000	79997	79997	0.9807
Methode Normal	10000	10000	#60000	#10000	60000	60000	0.9782
shift	10000	10000	#60000	#10000	300000	300000	0.9882

Table 1: comparaison des résultats.

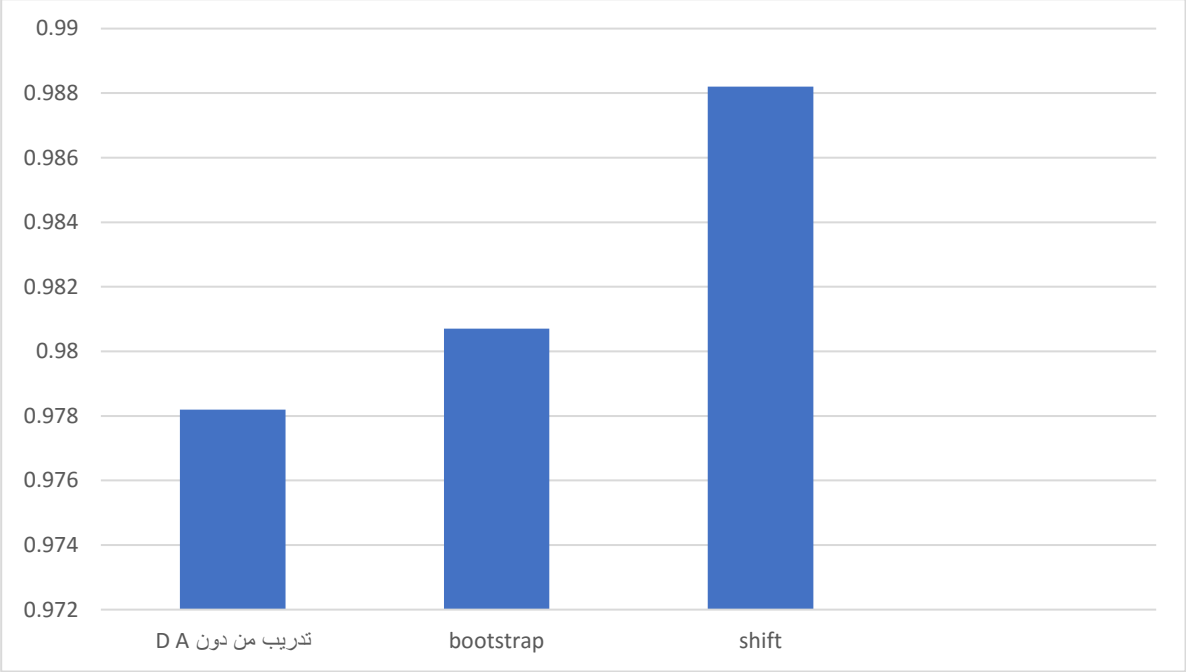


Figure 20: Diagramme en barre des resultats.

Nous notons que Shift a la note Bootstrap la plus élevée de 0,0075, où la moyenne de bootstrap était de 0,9807 et pour la technique Shift était de 0,9882, la différence n'est pas grande en synchronisation avec le nombre de données comme 300 000 X_train_augmented Shift après le nombre de données et X_train_augmented Bootstrap 79997 est une grande différence (Bootstrap augmenté de 19997, shift augmenté de 240000), mais la précision de notre modèle est acceptable dans une certaine mesure.

5-Conclusion:

Dans cette partie du mémoire, nous avons transformé les mécanismes et comment fabriquer des technique Bootstrap et Jackknife, puis afficher les résultats et les comparer avec la technique connue shift.

Conclusion général

Conclusion général :

L'augmentation des données permet d'améliorer les performances et les résultats des modèles d'apprentissage automatique. Le modèle d'apprentissage automatique fonctionne mieux et est plus précis lorsque l'ensemble de données est riche et suffisant. Après l'introduction des deux nouvelles technique (Jackknife, Bootstrap), nous avons pu effectuer cette tâche. Après plusieurs premières expériences avec la technique Bootstrap, cela a donné des résultats très acceptables et satisfaisants par rapport aux autres techniques.La technique la plus populaire et la plus utilisée est la technique de décalage. Malgré la grande différence dans le nombre de données d'enchères (Bootstrap 19997 ،shift 240000), le pourcentage de précision était très proche l'un de l'autre, et ce n'était pas une grande différence (0,0075). Il est possible qu'après des développements de cette technique, elle donne des résultats meilleurs et plus précis qui concurrencent d'autres techniques.Quant à la technique Jackknife, nous l'avons suggérée et expliquée comment elle fonctionne, mais nous n'avons pas eu le temps de la programmer et voir ses résultats.

Les résultats de cette recherche ne se sont pas arrêtés à ce point, il faudra peut-être plusieurs recherches et autres développements pour arriver à des résultats meilleurs et plus précis que ce que nous avons atteint maintenant.

Bibliographie

Bibliographes :

Introduction general

[1]: https://mgta.gmu.edu/courses/ml-with-python/handwrittenDigitRecognition.php

Chapitre I

[1]: https://research.aimultiple.com/data-augmentation/

[2] : Shorten, Connor; Khoshgoftaar, Taghi M. (2019). "A survey on Image Data Augmentation for Deep Learning". Mathematics and Computers in Simulation. springer. 6: 60.

[3]: The Stanford AI Lab Blog. https://ai.stanford.edu/blog/data-augmentation/

[4]:medıum.https://medium.com/analytics-vidhya/data-augmentation-is-it-really-necessary-b3cb12ab3c3f

[5]:medıum.https://medium.com/ymedialabs-innovation/data-augmentation-techniques-in-cnn-using-tensorflow-371ae43d5be9

[6]: Github. https://github.com/xkumiyu/numpy-data-augmentation

[7]:MEDIUM.https://odsc.medium.com/image-augmentation-for-convolutional-neural-networks-18319e1291c

[8]:https://www.limsi.fr/fr/actualites/847-tal-representations-vectorielles-et-apprentissage-automatique-pour-l-alignement-d-entites-textuelles-et-de-concepts-d-ontologie

[9]:Medium.https://medium.com/ymedialabs-innovation/data-augmentation-techniques-in-cnn-using-tensorflow-371ae43d5be9#8be0

[10]:KDnuggets.https://www.kdnuggets.com/2018/05/data-augmentation-deep-learning-limited-data.html

[11]:Tensorflow.org.https://www.tensorflow.org/tutorials/images/data_augmentation

[15]:Medium.https://medium.com/ideas-at-igenius/the-delicacy-of-data-augmentation-in-natural-language-processing-nlp-2ef07e9ad1c0

[17]: Github. https://github.com/makcedward/nlpaug

[18]: tnw. https://thenextweb.com/news/what-is-adversarial-machine-learning-syndication

[19]:TOWARDS DATA SCIENCE. https://towardsdatascience.com/advanced-data-augmentation-strategies-383226cd11ba

[20]https://research.aimultiple.com/data-augmentation-techniques/

Chapitre II

[1]: http://strata.uga.edu/8370/lecturenotes/resampling.html.

[2]: https://towardsdatascience.com/bootstrap-resampling-2b453bb036ec.

[3]: S. Sinharay, An Overview of Statistics in Education, Editor(s): Penelope Peterson, Eva Baker, Barry McGaw, International Encyclopedia of Education (Third Edition), Elsevier, 2010, Pages 1-11.

[4] :Quenouille, M. H. (1949). "Approximate Tests of Correlation in Time-Series". Journal of the Royal Statistical Society, Series B. 11 (1): 68–84.

[5] :Tukey, J. W. (1958). "Bias and Confidence in Not-quite Large Samples (Preliminary Report)". Annals of Mathematical Statistics. 29 (2): 614.

[6] :Mahalanobis, P. C. (1946). "Proceedings of a Meeting of the Royal Statistical Society held on July 16th, 1946". Journal of the Royal Statistical Society. 109 (4): 325–370.

[7]: "21 Bootstrapping Regression Models" .Archived *from the original on 2015-07-24.*

[8] :DiCiccio TJ, Efron B (1996) Bootstrap confidence intervals (with Discussion). Statistical Science 11: 189–228.

[9] : *Hinkley, David (1994-08-01).* "[Bootstrap: More than a Stab in the Dark?]". *Statistical Science. 9 (3).*

[10] :Goodhue, D.L., Lewis, W., & Thompson, R. (2012). Does PLS have advantages for small sample size or non-normal data? MIS Quarterly, 36(3), 981–1001.

[11] :Efron, B., Rogosa, D., & Tibshirani, R. (2004). Resampling methods of estimation. In N.J. Smelser, & P.B. Baltes (Eds.). International Encyclopedia of the Social & Behavioral Sciences (pp. 13216–13220). New York, NY: Elsevier.

[12] : Adèr, H. J., Mellenbergh G. J., & Hand, D. J. (2008). Advising on research methods: A consultant's companion. Huizen, The Netherlands: Johannes van Kessel Publishing.

[13] : Bootstrap of the mean in the infinite variance case Athreya, K.B. Ann Stats vol 15 (2) 1987 724–731.

[14] : https://en.wikipedia.org/wiki/Bootstrapping_(statistics).

[15]: Robert Nisbet, Gary Miner, Ken Yale, Chapter 11 - Model Evaluation and Enhancement, Editor(s): Robert Nisbet, Gary Miner, Ken Yale, Handbook of Statistical Analysis and Data Mining Applications (Second Edition), Academic Press,2018, Pages 215-233.

[16] :https://en.wikipedia.org/wiki/Jackknife_resampling.

[17] :Shao, J. and Tu, D. (1995). The Jackknife and Bootstrap. Springer-Verlag, Inc. pp. 281.

[18] :Shao, J.; Tu, D. (1995). The Jackknife and Bootstrap. Springer.

[19] :Wolter, K. M. (2007). Introduction to Variance Estimation (Second ed.). Springer.

[20] :Verbyla, D.; Litvaitis, J. (1989). "Resampling methods for evaluating classification accuracy of wildlife habitat models". Environmental Management. 13 (6): 783–787.

[21] :Moore, Jason H. "Bootstrapping, permutation testing and the method of surrogate data." Physics in Medicine & Biology 44.6 (1999): L11.

[22] :Onghena, Patrick (2017-10-30), Berger, Vance W. (ed.), "Randomization Tests or Permutation Tests? A Historical and Terminological Clarification", Randomization, Masking, and Allocation Concealment (1 ed.), Boca Raton, FL: Chapman and Hall/CRC, pp. 209–228, retrieved 2021-10-08.

[23].https://en.wikipedia.org/wiki/Resampling_(statistics)#/media/File:Bootstrapping.jpg

Chapitre III

[1] :AI research. aimultiple.com was first indexed by Google in April 2020, https:/research.aimultiple.com/data-augmentation]

[2] :"THE MNIST DATABASE of handwritten digits". Yann LeCun, Courant Institute, NYU Corinna Cortes, Google Labs, New York Christopher J.C. Burges, Microsoft Research, Redmond.

[3] :"Support vector machines speed pattern recognition - Vision Systems Design". Vision Systems Design. Retrieved 17 August 2013.

[4] :Gangaputra, Sachin. "Handwritten digit database". Retrieved 17 August 2013.

[5] :Qiao, Yu (2007). "THE MNIST DATABASE of handwritten digits". Retrieved 18 August 2013.

[6] :Platt, John C. (1999). "Using analytic QP and sparseness to speed training of support vector machines" Advances in Neural Information Processing Systems: 557–563. Archived from the original on 4 March 2016. Retrieved 18 August 2013.

[7] :Grother, Patrick J. "NIST Special Database 19 - Handprinted Forms and Characters Database" National Institute of Standards and Technology.

[8] :Jump up to:a b c d e f LeCun, Yann; Cortez, Corinna; Burges, Christopher C.J. "The MNIST Handwritten Digit Database". Yann LeCun's Website yann.lecun.com. Retrieved 30 April 2020.

[9] :Kussul, Ernst; Baidyk, Tatiana (2004). "Improved method of handwritten digit recognition tested on MNIST database". Image and Vision Computing. 22 (12): 971–981. doi:10.1016/j.imavis.2004.03.008.

[10] :Zhang, Bin; Srihari, Sargur N. (2004). "Fast k-Nearest Neighbor Classification Using Cluster-Based Trees" . IEEE Transactions on Pattern Analysis and Machine Intelligence. 26 (4): 525–528. doi:10.1109/TPAMI.2004.1265868. PMID 15382657. S2CID 6883417. Retrieved 20 April 2020.

[11] :https://www.datacamp.com/tutorial/svm-classification-scikit-learn-python

[12] :https://www.investopedia.com/terms/p/predictive-modeling.asp

[13] :https://www.isixsigma.com/dictionary/accuracy/

[14] : https://www.python.org/ , [Online; accessed 15/04/2019].

[15] : A. C. M. Sarah Guido – Le machine learning avec python, O'Reilly, First, 2018. https://www.amazon.fr/gp/product/241203446X/ref=as_li_qf_asin_il_tl?ie=UTF8&tag=pepsmultime0a21&creative=6746&linkCode=as2&creativeASIN=241203446X&linkId

=660e1082f58229153d3457241e5db28b#reader_B07HHM72D1,[Online;accessed 10/04/2019].

[16] : https://research.google.com/colaboratory/faq.html.