Advanced Machine Learning and Deep Learning
Consignes pour l'examen

Prof. Xavier Siebert et Prof. Sidi Ahmed Mahmoudi

2020-2021

L'évaluation de l'UE « Advanced Machine Learning and Deep Learning » se fera cette année sur base d'un projet en deux parties à réaliser par groupes de 3 étudiants maximum.

1. partie **AML (Advanced Machine Learning)** : un projet à choisir parmi ceux figurant dans ce document.
2. partie **ADL (Advanced Deep Learning)** : un projet à choisir parmi ceux dans le document annexe.

Comme expliqué au cours, il s'agit de combiner un projet **théorique** pour l'une des parties (**AML** ou **ADL**, au choix) avec un projet **pratique** pour l'autre.

- Veuillez nous communiquer par email pour le 18/12/2020
  - les membres du groupe (3 maximum)
  - 2 choix de combinaisons de projets AML-ADL. Nous essayerons de vous donner votre premier choix. Par exemple :
    1. premier choix : le projet 3-théorique pour AML combiné avec le projet 2-pratique pour ADL.
    2. deuxième choix : le projet 1-pratique pour AML combiné avec le projet 4-théorique pour ADL.
- Chaque partie (AML et ADL) fera l'objet d'un rapport de 10 pages maximum. Les codes de la partie pratique devront également être fournis. Le travail final est à remettre pour le **15 janvier 2021** au plus tard sur la boîte à dépôt sur la page `Moodle` du cours **Advanced Machine Learning**.

Les projets proposés pour la partie **Advanced Machine Learning** sont décrits dans ce document. Quelques références sont données, vous en trouverez facilement d'autres. N'oubliez pas de citer les références utilisées dans le rapport.

1 SLT1 - Méthode `Adaboost`

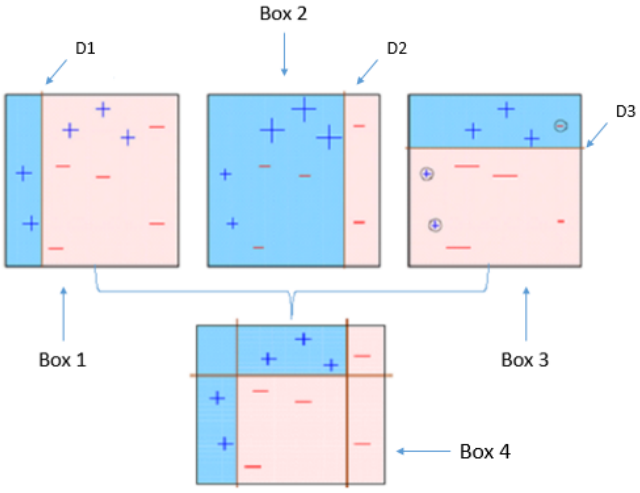


FIGURE 1 – Illustration de l'algorithme `Adaboost` (source : [6])

1. Partie théorique
   (a) Expliquez le principe du *boosting* et la méthode `Adaboost`.
   (b) Expliquez comment on obtient les bornes sur l'erreur de généralisation pour `Adaboost`
2. Partie pratique
   (a) Pour un jeu de données synthétique et un jeu de données réel, générer les courbes d'apprentissage avec la méthode `Adaboost`, en considérant comme classifieurs faibles
      - les classifieurs linéaires,
      - les arbres de décision.
   (b) Discuter les résultats :
      - La courbe empirique obtenue est-elle en accord avec la théorie ?
      - Les prédictions théoriques relatives sur les différents classifieurs reflètent-elles l'efficacité d'un classifieur par rapport à un l'autre ?

- Références : Chapitre 10 de [1]
- Données : [8]

2 SLT2 - Entropie de Vapnik-Chervonenkis

1. Partie Théorique
   (a) Expliquer l'utilisation de l'entropie de Vapnik-Chervonenkis. dans l'obtention de bornes de l'erreur de classification.
   (b) Comparer avec les bornes qui utilisent la dimension de Vapnik-Chervonenkis.
2. Partie Pratique
   (a) Pour un jeu de données synthétique et un jeu de données réel, générer les courbes d'apprentissage pour
      - un classifieur SVM
      - un classifieur kNN
   (b) Comparer avec les bornes théoriques.
      - avec la dimension de Vapnik-Chervonenkis
      - avec l'entropie de Vapnik-Chervonenkis
   (c) Discuter les résultats.
      - les prédictions théoriques absolues sont-elles proches des valeurs expérimentales ?
      - les prédictions théoriques relatives sur les différents classifieurs reflètent-elles l'efficacité d'un classifieur par rapport à un autre ?

- Références : [2]
- Données : [8]

3 SLT3 - Variables de Rademacher

1. Partie Théorique
   (a) Expliquer l'utilisation des variables de Rademacher dans l'obtention de bornes de l'erreur de classification.
   (b) Comparer avec les bornes qui utilisent la dimension de Vapnik-Chervonenkis.
2. Partie Pratique
   (a) Pour un jeu de données synthétique et un jeu de données réel, générer les courbes d'apprentissage pour
      - un classifieur SVM
      - un classifieur kNN
   (b) Comparer avec les bornes théoriques.
      - avec la dimension de Vapnik-Chervonenkis
      - obtenues grâce aux variables de Rademacher
   (c) Discuter les résultats.
      - les prédictions théoriques absolues sont-elles proches des valeurs expérimentales ?
      - les prédictions théoriques relatives sur les différents classifieurs reflètent-elles l'efficacité d'un classifieur par rapport à un autre ?

- Références : [2, 7]
- Données : [8]

4 SLT4 - Inégalités de Hoeffding et de Bernstein

1. Partie théorique
   (a) Démontrer en détails l'inégalité de Hoeffding pour les sommes de variables aléatoires bornées sur $[a, b]$
   (b) Enoncer l'inégalité de Bernstein. Quel est son intérêt par rapport à l'inégalité de Hoeffding ?
   (c) Donner un exemple d'application de l'inégalité de Bernstein.
2. Partie pratique : pas de partie pratique pour ce sujet

- Références : [2]

5 AL1 - Algorithme A^2

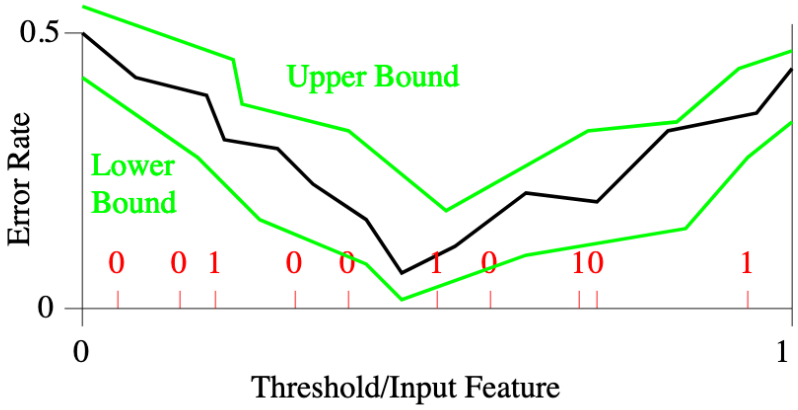


FIGURE 2 – Illustration de l'algorithme A^2 (source : [4])

1. Partie Théorique :
   (a) Expliquer le fonctionnement de l'algorithme A^2
   (b) Quelles sont les garanties théoriques sur le fonctionnement de cet algorithme et comment les obtient-on ?
2. Partie Pratique :
   (a) Implémenter l'algorithme A^2 sur un jeu de données synthétique et un jeu de données réel
   (b) Commenter les résultats.
   - l'algorithme est-il conforme aux prédictions théoriques ?
   - l'apprentissage actif avec A^2 est-il avantageux par rapport à un apprentissage « passif » ?
   - Quelles sont les possibles limitations de l'algorithme ?

- Références : [4]
- Données : [8]

6 AL2 - Algorithme `IWAL`

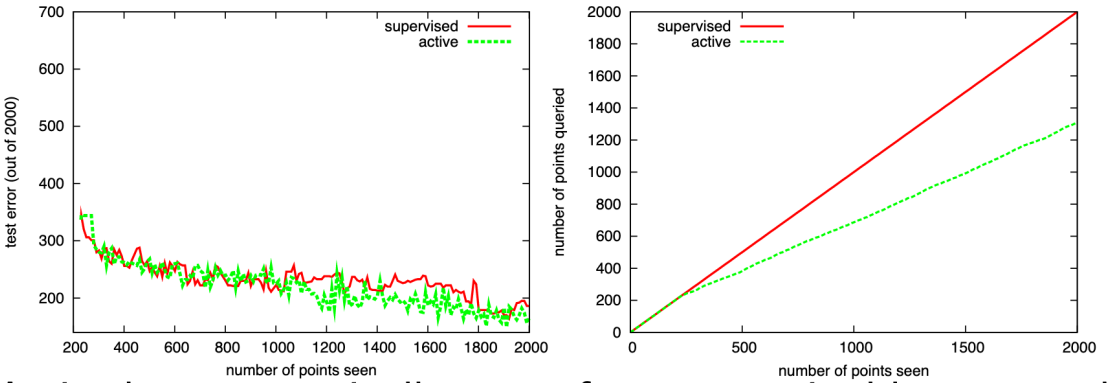


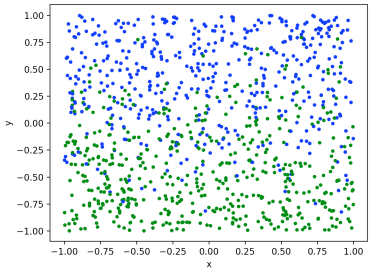
FIGURE 3 – Courbes d'apprentissage (gauche) et nombre de points choisis (droite) pour l'algorithme `IWAL` (source : [3])

1. Partie Théorique :
   (a) Expliquer le fonctionnement de l'algorithme `IWAL`.
   (b) Quelles sont les garanties théoriques sur le fonctionnement de cet algorithme et comment les obtient-on ?
2. Partie Pratique :
   (a) Implémenter l'algorithme `IWAL` sur un jeu de données synthétique et un jeu de données réel
   (b) Commenter les résultats.
   - l'algorithme est-il conforme aux prédictions théoriques ?
   - l'apprentissage actif avec A^2 est-il avantageux par rapport à un apprentissage « passif » ?
   - Quelles sont les possibles limitations de l'algorithme ?

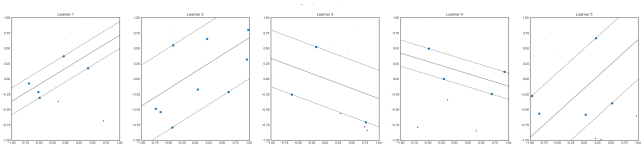
- Références : [3]
- Données : [8]

7 AL3 - Apprentissage actif par comité avec abstention

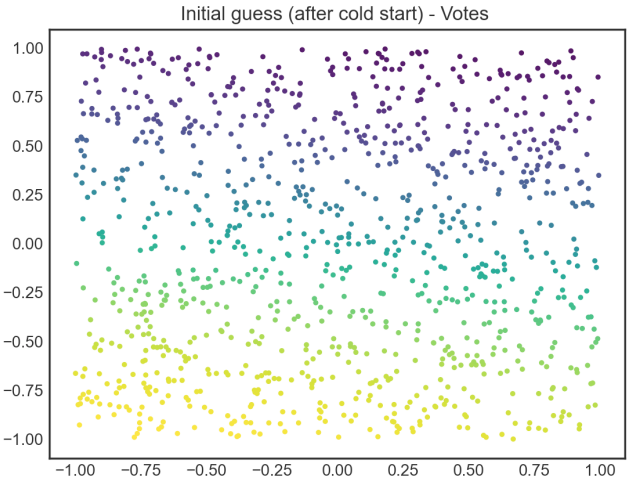
1. Partie Théorique : pas de partie théorique pour ce sujet
2. Partie Pratique : Implémenter l'algorithme QBC (vu au cours) pour une classification binaire, en y incorporant une possibilité pour l'oracle de s'abstenir si l'incertitude sur le label des points est trop grande, autrement dit :
   (a) Générer deux groupes de 500 points chacun, de sorte que chaque groupe corresponde à une classe, et que les classes se chevauchent



   (b) Construire un comité de 5 classifieurs linéaires, entraînés chacun sur 10 points aléatoires.



   (c) Faire voter ces 5 classifieurs sur les labels de tous les points, pour déterminer les points les plus incertains



   (d) Implémenter deux scénarios et comparer les courbes d'apprentissage :
   - en apprentissage QBC classique, les points les plus incertains (probabilité d'appartenir à une classe $\approx 0{,}5$) seraient considérés comme les plus informatifs et on demande leur label à l'oracle, puis on itère
   - en apprentissage QBC avec abstention, si ces points ont une probabilité d'appartenir à une classe trop proche de 0,5 (disons $0,5 \pm \lambda$ avec λ), l'oracle s'abstient. Essayer avec différentes valeurs de λ.

8 B1 - Bandits adversariels

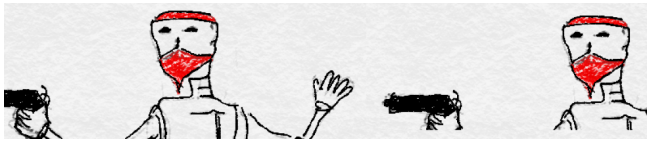


FIGURE 4 – Bandits (illustration de [5])

1. Partie Théorique :
   (a) Expliquer ce qu'est un problème de Bandits adversariels, par opposition au problème de bandit stochastique vu en classe
   (b) Expliquer le fonctionnement de l'algorithme `Exp3`.
2. Partie Pratique :
   (a) Implémenter l'algorithme `Exp3` sur un problème au choix, par exemple les données boursières [13] ou les données de traffic de sites web [14].
   (b) Commenter les résultats

- Références : [5, 12]
- Données : [13, 14]

9 B2 - Monte-Carlo Tree Search

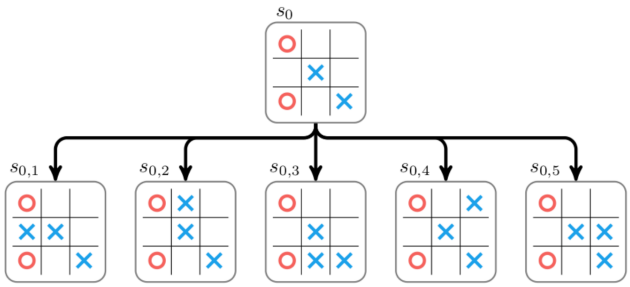


FIGURE 5 – Monte-Carlo Tree Search pour le jeu de Tic-Tac-Toe

1. Partie Théorique :
   (a) Expliquer l'algorithme `Monte-Carlo Tree Search` et son intérêt pratique
   (b) Expliquer la variante « Upper Confidence bound applied to Trees » (UCT), en faisant le lien avec l'algorithme UCB vu au cours.
2. Partie Pratique :
   (a) Implémenter l'algorithme `Monte-Carlo Tree Search` sur un problème au choix, par exemple sur un jeu simple comme Tic-Tac-Toe, que vous pouvez simuler en vous inspirant de [16]
   (b) Commenter les résultats

- Références : [5, 15]
- Données : [16, 17]

10 B3 - Bandits de Thompson

FIGURE 6 – Divers films du système de recommandation.

1. Partie Théorique :
   (a) Expliquer le principe du *Thompson Sampling*
   (b) Montrer comment on obtient des bornes sur le regret pour l'algorithme `Exp3`
2. Partie Pratique :

   Cette partie concerne la recommandation de films, proposé sous forme de concours par *Netflix* il ya quelques années. On ne vous demande pas ici de développer un système de recommandation complet, mais d'identifier le film le plus populaire, en supposant qu'au départ aucun film n'a été visionné.

   (a) Il faut donc développer une stratégie pour demander à des utilisateurs de visionner des films et de les noter. Pour ne pas fatiguer les utilisateurs, il faut parvenir le plus rapidement possible à identifier le meilleur film. Formalisez ce problème sous forme de problème de bandits.
   (b) Implémenter l'algorithme `Thompson Sampling` pour résoudre ce problème.
   (c) Commenter les résultats.

- Références : [5, 9, 12]
- Données : Netflix [10], MovieLens [11] ou autre...

Références

Articles et livres

[1] Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning : From theory to algorithms. Cambridge University P ress.

[2] *Introduction to Statistical Learning Theory*. O. Bousquet, S. Boucheron, and G. Lugosi, Lecture Notes in Artificial Intelligence, 3176 ; pp 169–207 (2004).

[3] *Importance weighted active learning*. Beygelzimer, A., Dasgupta, S., and Langford, J.. In Proceedings of the 26th annual international conference on machine learning, pp. 49-56, (2009)

[4] *Agnostic active learning* ; Balcan, M. F., Beygelzimer, A., and Langford, J. Journal of Computer and System Sciences, 75(1), 78-89. (2009)

[5] *Bandit algorithms*. Lattimore, Tor, and Csaba Szepesvári. Cambridge University Press, (2020).

Les ouvrages cités ci-dessus sont disponibles en format `pdf` sur `Moodle`.

Sites web

[6] `https://towardsdatascience.com/understanding-adaboost-2f94f22d5bfe`

[7] `https://www.youtube.com/watch?v=gR9Q8pS03ZE`

[8] UCI Machine Learning Repository (classification datasets)
`https://archive.ics.uci.edu/ml/datasets.php?format=&task=cla&att=&area=&numAtt=&numIns=&type=&sort=nameUp&view=table`

[9] `https://blog.insightdatascience.com/multi-armed-bandits-for-dynamic-movie-recommendatio`

[10] `www.kaggle.com/laowingkin/netflix-movie-recommendation/data`

[11] `https://www.kaggle.com/grouplens/movielens-20m-dataset`

[12] `https://jeremykun.com/2013/11/08/adversarial-bandits-and-the-exp3-algorithm`

[13] `https://jeremykun.com/2013/12/09/bandits-and-stocks`

[14] `https://sourceforge.net/projects/bandit/files/Datasets`

[15] `http://tim.hibal.org/blog/alpha-zero-how-and-why-it-works/`

[16] `https://medium.com/swlh/tic-tac-toe-at-the-monte-carlo-a5e0394c7bc2`

[17] `https://www.baeldung.com/java-monte-carlo-tree-search`

I. CNN & RNN pour la reconnaissance de mouvements/actions

1.1. Partie théorique :

- Expliquer la différence entre un réseau de neurones convolutionnel (CNN) et un réseau de neurones récurrent (RNN))
- Expliquer le problème de « Vanishing Gradient » rencontré avec les RNNs. Quelles solutions existent ?

1.2. Partie Pratique :

- Développer un réseau de neurones profond utilisant les CNNs et RNNs en vue de détecter les mouvements ou actions dans une séquence vidéo. Vous pouvez réaliser les expérimentations à l'aide de bases de données publiques.

1.3. Liens intéressants :

- Méthode TSM: Temporal Shift Module for Efficient Video Understanding
- Code : https://github.com/open-mmlab/mmaction2
- Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*.

II. Réseaux de neurones génératifs GAN pour la prédiction de mouvements

2.1. Partie théorique :

- Expliquer le principe des réseaux de neurones de type « Auto encoder »
- Expliquer le principe des réseaux de neurones génératifs « GAN »
- Quelle est la différence entre réseaux « Auto encoder » et « GAN » ?

2.2. Partie Pratique :

- Développer un réseau de neurones de type « GAN » pour prédire les mouvements et actions (frames futurs) dans une séquence vidéo dans le domaine de vidéosurveillance. Par exemple, si une personne se dirige vers une zone interdite, le réseau permettra de prédire que la personne se retrouvera en zone interdite dans quelques instants.

2.3. Liens intéressants :

- **Paper 1 "**Predicting Future Frames using Retrospective Cycle GAN **"**
- **Paper 2 : "**BiHMP-GAN: Bidirectional 3D Human Motion Prediction GAN**"**
- https://github.com/amoghadishesha/GAN-motion-Prediction
- https://paperswithcode.com/task/human-motion-prediction

III. Auto-encoders pour la segmentation d'images médicales « Covid-19 »

3.1. Partie théorique :

- Expliquer le principe des réseaux de neurones profonds utilisés pour la segmentation d'images ;
- Illustrer quelques architectures neuronales utilisées pour la segmentation d'images médicales.

3.2. Partie Pratique :

- Développer un réseau de neurones profond permettant de segmenter les poumons et lésions responsables de la pathologie Covid-19 en utilisant des images de scan (CT-scans).
- Vous pouvez travailler avec des bases de données publiques.

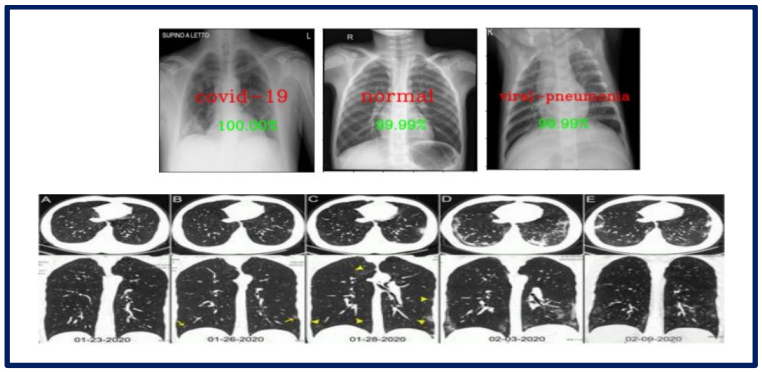


Figure 1: exemple d'images X-ray et CT-scans : covid-19, normal et viral-pneumonia

IV. Liens intéressants :

- Exemple de base de données : https://github.com/UCSD-AI4H/COVID-CT
- https://paperswithcode.com/task/covid-19-image-segmentation

---

IV. XAI pour la classification et reconnaissance d'actions

4.1. Partie théorique :

- Dans la littérature, les méthodes de Deep Learning sont appelées par le terme "**black-box model**". Quelle en est la signification ?
- Proposez une méthode d'explicabilité de l'intelligence artificielle permettant d'expliquer les décisions d'un réseau Deep Learning entraîné. Expliquer son fonctionnement.
- Un modèle Deep Learning entraîné à une précision de 100% sur un ensemble de test. Cela est-il suffisant pour justifier son utilisation directe dans un environnement réel ? Proposer quelques idées (liées ou non à l'explicabilité) et justifiez-les.

4.2. Partie Pratique :

- Développer une méthode d'explication et d'interprétation de modèles Deep Learning appliqués à une application de classification de vidéos et de reconnaissance d'actions. La particularité de ce type d'application est représentée par son besoin de combiner l'information spatiale (image)

et temporelle présentes dans une séquence vidéo ce qui requiert le développement de modèles combinant différents types de réseaux de neurones profonds (MLP, CNN, RNN, LSTM, etc.). L'objectif de ce mini-projet d'interpréter et expliquer ces modèles combinant différents DNNs afin de :

- Calculer et quantifier la contribution chaque frame (image) de la vidéo dans le résultat final du modèle : détecter les frames pertinents.
- Parmi les frames pertinents, calculer et quantifier la contribution des pixels (ou régions) dans le résultat final.

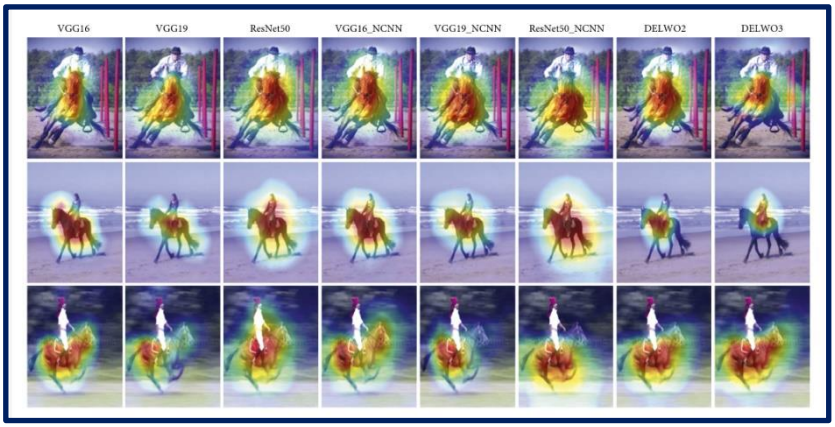


Figure 2: exemples de reconnaissances d'action avec carte d'attention [Paper1]

4.3. Liens intéressants :

- Paper 1
- Paper 2
- Paper 3
- Code

V. Système « Edge IA » pour villes intelligentes

5.1. Partie théorique :

- Présenter une analyse comparative entre les différents types de réseaux de neurones profonds

5.2. Partie pratique :

L'objectif de ce mini-projet est de développer un système IA embarqué sur une ressource Edge (nvidia Jetson Xavier). Le système s'appuiera sur les techniques et modèles Deep Learning de classification d'images et de localisation d'objets qui peuvent être utilisées dans un système de ville intelligente. Ces modèles seront combinés pour fournir un module « Edge IA » appliqué aux vidéos capturées en temps réel. Vous pouvez intégrer deux ou trois modèles parmi :

- Modèle de reconnaissance faciale

- Modèle de détection de feux de forêts : pour détecter les feux ou fumées dans des forêts proches de la ville intelligente en utilisant les images capturées via la caméra. Une base de données et une solution sera fournie
- Modèle de détection d'objets suspects : pour détecter la présence d'un ou plusieurs objets suspects ou non autorisés (ex. bâtons, sacs isolés, armes, etc.) dans une scène filmée par la caméra.

Le challenge sera de porter les réseaux de neurones sur la ressource Edge et fournir un calcul en temps réel. Le choix d'architectures neuronales devra prendre en compte la taille des modèles générés ainsi que leurs temps d'inférence.

Note : le matériel nécessaire (carte Jetson Xavier + accessoires) ainsi que deux modèles (classification d'images de feu et localisation d'objets personnels) seront fournis.

5.3. Liens intéressants :

- Modèle de reconnaissance faciale : https://github.com/davidsandberg/facenet

---

VI. Compression et Optimisation de réseaux de neurones profond

6.1. Partie théorique :

- Citer et expliquer les méthodes de compression de réseaux de neurones profonds

6.2. Partie pratique :

- Développer un algorithme de compression de réseaux de neurones profond (ex. Pruning, quantification, convolution séparable 1D) permettant de réduire sa taille mémoire, accélérer son temps d'inférence en garantissant le maintien de bonnes performances en termes de précision « accuracy » et perte « loss ».

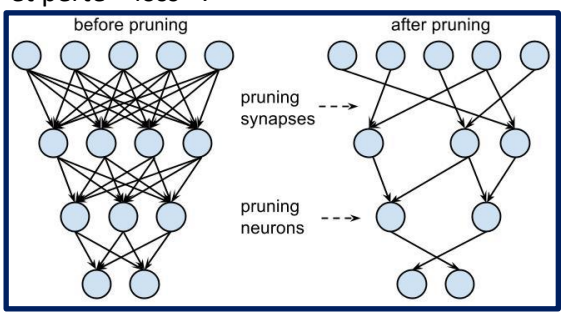


Figure 3: technique de Pruning

6.3. Liens intéressants :

- Code 1: https://github.com/arturjordao/PruningNeuralNetworks
- Code 2 : https://github.com/jiajuns/Neural-Network-Pruning-Keras
- Paper 1 : https://arxiv.org/pdf/1710.01878.pdf
- Paper 2: https://arxiv.org/pdf/1611.06440.pdf