1 INF 1901 - Module 2 - Apprentissage machine

1.1 Quels sont les objectifs de ce module?

1.2 Qu'est-ce que l'apprentissage machine?

L'apprentissage machine (AM) est un ensemble de techniques mathématiques qui permettent de résoudre des problèmes ardus en informatique, souvent associés à l'intelligence artificielle (IA) : classifier ou reconnaître des images (est-ce un chien ou un chat?), prédire la valeur d'une maison, jouer aux échecs, converser en anglais et résoudre des problèmes généraux, etc.

Ces problèmes sont considérés difficiles car il serait ardu d'écrire un programme classique pour les résoudre. Un programme classique encode essentiellement une série de règles et de procédures logiques pour résoudre un problème, tandis qu'un modèle d'AM dérive plutôt sa solution à partir d'exemples. Le fait qu'on parle d'intelligence de manière plus explicite dans le cas d'un modèle d'AM (par rapport à un programme classique) est un peu arbitraire, et matière à débat. Il reste que fondamentalement, l'AM est associée à des courants philosophiques, comme le connexionnisme par exemple, qui sont généralement associés à l'étude de l'intelligence humaine ou animale.

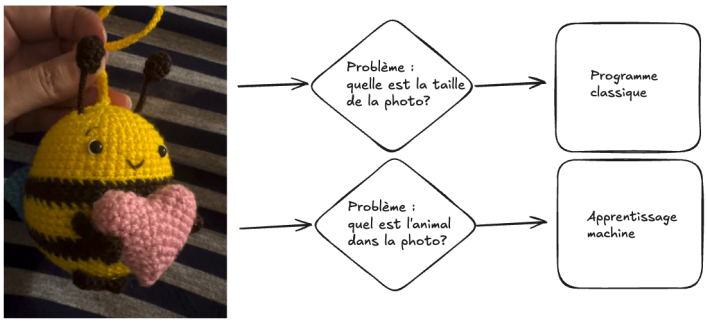

Figure 1: Problème classique versus problème d'apprentissage

En général, l'apprentissage machine utilise des données (qu'on appelle parfois des exemples) pour "entrainer" un modèle à l'aide d'un algorithme d'apprentissage. Une fois l'entrainement accompli, on peut utiliser l'algorithme dans un contexte où c'est utile. Le modèle est dynamique et changeant seulement dans la phase d'entrainement, à la phase d'utilisation, il est un objet statique.

La notion probablement la plus profonde et philosophique de l'AM, et celle qui fait en sorte qu'on rattache ce domaine à l'IA, est qu'un algorithme d'apprentissage devrait être en mesure de généraliser : si j'ai entrainé un modèle à distinguer entre un chien et un chat avec 1000 images d'entrainement, je ne suis pas intéressé par la performance du modèle sur l'une des images particulières qui ont servi à l'entrainement. Par construction et quasiment par définition, cette classification particulière devrait être correcte. Je suis plutôt intéressé par la classification de la 1001ième image, qui n'a pas servi à l'entrainement du modèle, et qui est donc entièrement nouvelle. Si le modèle a été entrainé avec succès, il devrait pouvoir généraliser à n'importe quelle image (par contre, la question se pose à savoir ce qui devrait arriver si je lui présente une image d'une vache!). Une bonne capacité de généralisation est le but fondamental de l'AM et de l'IA en général, et est reliée à ce qu'on entend par intelligence, scientifiquement parlant.

1.3 L'apprentissage machine dans un vrai scénario industriel

1.3.1 Situation

Imaginez une compagnie où il y a une chaine de montage où on assemble des téléviseurs.

1.3.2 Le problème

Supposons que dans un endroit particulièrement délicat de la chaine de montage, un problème survienne parfois, que l'on aimerait détecter le plus rapidement possible.

1.3.3 Une solution possible

On pourrait imaginer placer une caméra vidéo dont le but serait de visionner le flot des appareils en cours d'assemblage, pour tenter de détecter les problèmes. Pour ce faire, la caméra pourrait transmettre, à intervalles réguliers, les pixels de ce qu'elle capte, en tant que donnée à un modèle d'apprentissage machine qui roulerait (en tant que programme) sur un serveur, pas très éloigné. Ce modèle convertirait les pixels de la caméra en tant que données numériques (les entrées, "inputs"), et effectuerait un calcul complexe sur ces valeurs, en vue nde produire une valeur de sortie simple ("outputs") : "oui il y a un problème avec cette image", ou "non il n'y a pas de problème avec cette image". Sur la base de cette valeur de sortir, on pourrait agir et envoyer un technicien, en cas de besoin.

1.3.4 Quelle sera la nature de ce modèle?

Ce modèle sera essentiellement un modèle au sens statistique classique du terme : une série de paramètres déterminant une fonction particulière. Par analogie avec une fonction classique qu'on apprend au secondaire :

$$f(x) = mx + b$$

1.3.5 Comment calculer les paramètres de ce modèle?

Avant d'expliquer comment calculer les paramètres, nous devons déterminer plus précisément

Nous devons tout d'abord définir et clarifier certaines notions.

1. Que sont les paramètres

2. Qu'est-ce qu'un ensemble d'entrainement?

   Nous avons tout d'abord besoin d'un ensemble d'entrainement, qui est constitué d'une série d'images, accompagnées d'une étiquette "oui c'est un problème", ou "non ce n'est pas un problème". Évidemment dans la réalité les problèmes sont plus rares, mais il faudrait idéalement que cet ensemble d'entrainement (de disons 1000 images) soit constitué à 50% de cas problématiques, et 50% de cas non-problématiques.

3. Qu'est-ce que la fonction de coût?

   Une ensuite

   Mais comment a-t-on fait pour établir les paramètres du calcul effectué? Comment distinguer un problème d'un non-problème, à partir d'une longue série de valeurs numériques seulement (les pixels)? Il a fallu entrainer le modèle, et pour ce faire, on a dû tout d'abord amasser une grande quantité d'images de cas problématiques, ainsi que de cas non-problématiques. Pour chaque cas problématique, on l'associe à une étiquette "problème", et on fait de même pour les cas non-problématiques. Supposons qu'on a une banque de 10,000 cas d'entrainement (la moitié problématiques, l'autre moitié non-problématiques). Si on effectue le calcul sur ces données, il sera facile de calculer la "fonction d'erreur" : le pourcentage de "bonnes réponses". Avant tout entrainement, le modèle est essentiellement comme quelqu'un qui utiliserait une pièce de monnaie pour déterminer la réponse : il donnerait la "bonne" réponse (problématique ou non) dans seulement 50% des cas.

Ensuite on utilise ces données d'entrainement pour effectuer un calcul encore plus complexe et couteux, qui aura le but d'optimiser

1.4 En quoi l'AM diffère de la programmation traditionnelle?

Bien que l'apprentissage machine requiert de la programmation, il s'agit d'un paradigme entièrement différent de celui de la programmation.

Un programme traditionnel spécifie une série d'instructions que l'ordinateur exécute pour résoudre un problème. Normalement, ce programme fait son travail en relation avec des données fournies par l'utilisateur. Le programme dans ce cas est une série d'instructions symboliques dans un langage de programmation.

Un modèle d'AM (déjà entrainé) va prendre en entrée des données fournies par l'utilisateur, et va fournir une réponse appropriée après avoir effectué une série d'opérations mathématiques. Si on veut absolument parler de "programme" dans ce cas, on peut parler des opérations mathématiques (pas nécessairement symboliques) qui sont effectuées sur les données, pour les transformer en réponse. Il est important de comprendre que même si un modèle d'AM est avant tout un objet mathématique (un modèle avec ses paramètres), son implémentation concrète se fait quand même toujours avec un langage de programmation.

1.5 En quoi l'AM diffère de l'IA?

L'intelligence artificielle est le domaine plus vaste, qui englobe l'apprentissage machine. Les deux ont des méthodes profondément différentes, et l'histoire de leur développement est entièrement différente. Dans un certain sens, l'AM est une forme plus spécialisée et un peu plus récente d'IA, plus mathématique, moins symbolique, et clairement celle qui domine la période actuelle.

1.6 En quoi l'AM diffère des statistiques?

L'apprentissage machine, conceptuellement, est pratiquement identique aux statistiques. Dans les deux cas on parle de modèles, d'entrainement (ou recherche des paramètres), d'inférence, etc. Toutefois l'AM est plus axée sur les problèmes dont la modélisation se fait en très haute dimension, comme l'analyse d'images ou le traitement du langage. De plus, l'accent en AM est davantage mis sur les aspects computationnels, par opposition aux mathématiques (bien que le AM demeure très mathématique en substance).

1.7 Comment représenter les données

Un problème crucial qui se pose en AM est comment adéquatement représenter les données, pour qu'elles soient traitables et compréhensibles à la fois par l'ordinateur ainsi que le modèle (ou algorithme) d'apprentissage qu'on veut utiliser. Il existe de nombreuses manières de faire cela, mais un thème récurrent est l'utilisation d'espaces vectoriels pour représenter les données, ce qui est très étroitement relié au fait que la plupart des techniques d'AM touche de près ou de loin l'algèbre linéaire. Une image, par exemple, sera un point dans un espace vectoriel à très haute dimension (autant de dimensions qu'il y a de pixels!), et un mot pourrait être un point dans un espace vectoriel extrêmement épars (sparse) pour représenter la présence ou l'absence d'un mot. Il est également possible de représenter le sens des mots à l'aide d'un espace vectoriel, dont les grands modèles de langage (GML) font usage.

On parle souvent de "features" en AM, qui sont les caractéristiques, souvent numériques, mais pas toujours, des instances, ou des objets que l'on tente de traiter. Classiquement, on fait de l'ingénierie de features sur les données, pour tenter de les transformer de manière à améliorer les performances d'un algorithme. Le AM très moderne qui utilise les réseaux de neurones profonds tend à faire en sorte qu'on a moins besoin de ce genre de techniques, car les transformations sont faites automatiquement, par le réseau de neurones lui-même.

1.8 Les différents paradigmes de l'AM

Il existe plusieurs manières de catégoriser les algorithmes d'apprentissage machine, selon la nature et la structure des problèmes qu'ils tentent de résoudre. La catégorisation suivante est très classique.

1.8.1 Apprentissage supervisé (classification, regression)

L'apprentissage supervisé fonctionne à partir de données pour lesquelles la "bonne réponse" (i.e. celle qu'on aimerait que l'algorithme donne systématiquement) est fournie, en tant que donnée d'entrainement.

1. Régression

   Une régression est une famille d'algorithmes d'apprentissage supervisé (ou plus classiquement, de modélisation statistique) dont le but est de découvrir une fonction numérique continue, au sens classique mathématique (dans sa forme la plus simple, une fonction associe une valeur numérique du domaine X vers l'image Y).

- Régression linéaire (ex. nombre de pièces, année de construction -> prix d'une maison)

2. Classification

   Une autre famille d'algorithmes d'apprentissage supervisé tente plutôt de découvrir une fonction de classification, qui associe une série de features à une catégorie particulière (dont le nombre est fini et connu d'avance).

   - Régression logistique (ex. nombre d'heures étudiées, nombre de cours -> étudiant a gradué ou non)
   - k-NN
   - Arbres de décision
   - Naive Bayes

1.8.2 Apprentissage non-supervisé

L'apprentissage non-supervisé fonctionne à partir de données pour lesquelles la "bonne réponse" n'est pas fournie. Les algorithmes de cette famille doivent donc découvrir la structure inhérente aux données, de manière autonome, tout en étant guidé possible par des hypothèses et ses "biais inductifs".

1. Partitionnement (clustering)

   Avec un algorithme de partitionnement, on peut découvrir des "agrégats", ou des groupes naturels dans les données.

   - k-Means
   - DBScan
   - Hierarchical clustering

2. Réduction de la dimensionnalité

   En tentant de réduire la dimensionnalité des données, on peut découvrir sa structure inhérente, ce qui est souvent utile en visualisation (par exemple, une donnée exprimée en très haute dimension peut être plus facile à comprendre en 2d ou 3d).

   - PCA

1.8.3 Apprentissage par renforcement (RL)

L'apprentissage par renforcement (APR) est un paradigme différent des deux précédents. Si les apprentissages supervisé et non-supervisé pourraient être qualifiés de perceptifs (quelle est la nature de ce que je perçois?), l'apprentissage par renforcement pourrait être compris en tant que modélisation behaviorale (quelle action devrait être posée dans ce contexte particulier). L'APR est souvent utilisé dans les jeux et la robotique.

1.9 Réseaux de neurones

Les réseaux de neurones sont un algorithme d'apprentissage classiquement supervisé (mais cela va au-delà) extrêmement puissant et versatile, qui est l'élément clé à la base des révolutions de l'apprentissage profond et de l'IA génératif des temps récents. L'idée est de faire passer les données représentées à travers une série de couches de neurones, connectées par des matrices de poids (nombres réels), de manière à les transformer de manière extrêmement complexe et non-linéaire, afin de pouvoir découvrir des associations extrêmement sophistiquées et subtiles entre les données d'entrée (par exemple le prompt de ChatGPT) et les données de sortie (sa réponse). Le nombre de couches internes fait en sorte que ces réseaux sont qualifiés de "profonds", ce qui mène à l'apprentissage profond (deep learning).

1.10 Les applications de l'AM

- Modélisation
- Tests médicaux
- Jeux
- Chatbot
- Etc.

1.11 Concepts

1.11.1 Données

1.11.2 Représentation

1.11.3 Paramètres

1.11.4 Fonction objective (d'erreur)

1.11.5 Entrainement

1.11.6 Généralisation

1.11.7 Algorithme

1.11.8 Implémentation

1.11.9 Ingénierie des caractéristiques (feature engineering)

Quelles sont les composantes d'un NN?

- Noeuds
- Poids
- Fonction d'erreur
- Optimiseur (qui opt la fonction d'erreur)