

Chapitre 13

Synthèse – L'intelligence artificielle

L'histoire du traitement de l'information → Unité 1

- Jusqu'au début du XX^e siècle, les machines traitant l'information sont limitées à une ou quelques tâches prédéterminées : tisser grâce à un ruban ou des cartes perforées, trier un jeu de cartes perforées, séparer des cartes selon un critère, sommer des valeurs indiquées sur ces cartes, etc. Comme le montre l'exemple de la machine analytique imaginée par Babbage (1834) et celle du métier à tisser configurable de Jacquard (1801), les progrès techniques du traitement de l'information se sont construits en utilisant et en adaptant les techniques antérieures.
- Au début du XX^e siècle, des questions mathématiques nouvelles émergent : peut-on tout calculer ? Pour toute propriété mathématique, peut-on décider si elle est vraie ou fausse ? Pour répondre à ces questions, Turing (1936) est le premier à proposer le concept de machine universelle. En s'inspirant des travaux de Babbage et de Turing, Von Neumann décrit l'architecture d'une telle machine dont la matérialisation est effective au début des années 1950 avec les premiers ordinateurs.
- Conformément à l'architecture de Von Neumann, les premiers ordinateurs sont constitués à minima d'un processeur et d'une mémoire vive.

Savoir-faire

- Analyser des documents historiques relatifs au traitement de l'information et à son automatisation.

- Recenser les différentes situations de la vie courante où sont utilisés les ordinateurs, identifier lesquels sont programmables et par qui (thermostat d'ambiance, smartphone, box internet, ordinateur de bord d'une voiture...).

La numérisation de l'information → Unité 2

- La première machine analytique de Babbage utilisait des cartes perforées. Pour effectuer et programmer (Lovelace, 1843) que des calculs numériques, un ordinateur actuel manipule des données de natures diverses une fois qu'elles ont été numérisées : textes, images, sons. Pour contrôler les actions qu'un ordinateur exécute, on utilise des programmes écrits avec des langages de programmation comme Python ou Scratch.
- Les programmes sont donc également des données : comme les données numérisées, ils peuvent être stockés, transportés, et traités par des ordinateurs. En particulier, un programme écrit dans un langage de programmation de haut niveau (Python, Scratch) peut être traduit en instructions spécifiques à chaque type de processeur.

Savoir-faire

- Savoir distinguer les fichiers exécutables des autres fichiers sous un système d'exploitation donné.
- Connaître l'ordre de grandeur de la taille d'un fichier image, son, vidéo. Savoir calculer la taille en octets d'une page de texte (en ASCII et non compressé).

Les bugs → Unité 3

- Comme c'est le cas par exemple pour une voiture autonome, un programme peut comporter jusqu'à plusieurs centaines de millions de lignes de code. La présence d'erreurs, de défauts de conception, appelées bogues (ou bugs) est donc très probable.
- Ces erreurs peuvent conduire un programme à avoir un comportement inattendu et gênant et, dans certains cas, entraîner des conséquences graves. L'utilisation de jeux de données est un des outils qui peut permettre d'éviter l'apparition de bugs.

Savoir-faire

- Étant donné un programme très simple, proposer des jeux de données d'entrée permettant d'en tester toutes les lignes.
- Corriger un algorithme ou un programme bogué simple.

Mots clés

Fichier exécutable : un fichier exécutable est un fichier contenant un programme et identifié comme tel par le système d'exploitation. Le chargement d'un fichier exécutable entraîne la création d'un processus et l'exécution du programme.

Octet : un octet est composé de huit bits.

Bit : contraction de Binary digit, qui signifie chiffre binaire et apparaît en 1948. Le bit représente la plus petite unité de mémoire utilisable sur un ordinateur. Cette mémoire ne peut prendre que deux valeurs, symbolisées le plus souvent par 0 ou 1.