



Dr CADIC Philippe

MD , MG

pcadic@gmail.com @sulfuroid


Des définitions

D'abord fut l'algorithmique

Définition : L'algorithmique est la science qui étudie la conception, la structuration et l'optimisation des algorithmes — des suites d'instructions logiques permettant de résoudre un problème ou d'effectuer une tâche donnée.

Usage médical : Utilisation d'algorithmes pour automatiser le tri de dossiers médicaux, gérer les files d'attente ou coder les actes médicaux selon des règles prédéfinies.

Référence logicielle : Excel avec macros VBA — exemple d'environnement simple d'exécution d'algorithmes pour automatiser des tâches répétitives.



```
    'replace_interests' => false,  
    'send_welcome'     => false,  
  });  
  
  on_array('error', {result}) {  
    result = array ('response'=>'error', 'mess  
  }  
  result = array ('response'=>'success');  
  on_message(result);
```

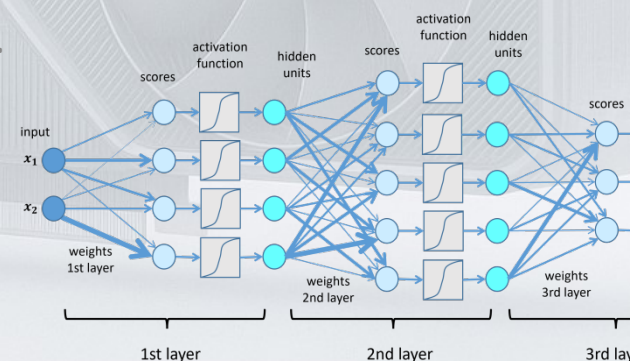
Des définitions

Puis vint le Deep learning

Définition : Le Deep Learning (apprentissage profond) est une branche de l'apprentissage automatique (machine learning) reposant sur des **réseaux de neurones artificiels multicouches**. Il permet d'apprendre à partir de grandes quantités de données, en mimant le fonctionnement du cerveau humain.

Usage médical : Diagnostic assisté par IA pour détecter des tumeurs sur imagerie médicale (scanner, IRM, fond d'œil...).

Référence logicielle : TensorFlow (Google) — une des bibliothèques les plus utilisées pour entraîner des réseaux de neurones.



Des définitions

Puis vint L'intelligence artificielle

Définition : L'intelligence artificielle désigne l'ensemble des techniques visant à reproduire des capacités humaines (raisonnement, prise de décision, reconnaissance visuelle ou vocale...) par des machines :

- **IA symbolique** (basée sur des règles logiques),
- **IA statistique** (machine learning),
- **IA connexionniste** (deep learning),
- **IA hybride** (combinaison de plusieurs approches).

Usage médical : Triage automatisé aux urgences, systèmes d'aide à la décision thérapeutique.

Référence logicielle : Watson Health (IBM) — solution d'IA intégrée dans la santé, notamment en oncologie (même si partiellement arrêtée, elle a marqué l'histoire de l'IA médicale).



Des définitions

Puis vint L'intelligence artificielle générative

Définition : L'IA générative est une catégorie d'intelligence artificielle capable de **créer du contenu original** (textes, images, sons, vidéos...) à partir de données d'entraînement. Elle repose sur des modèles de langage avancés comme les **LLM** (Large Language Models), utilisant des milliards de paramètres.

Usage médical : Génération automatique de courriers médicaux, résumés de consultations, notices explicatives pour les patients.

Référence logicielle : ChatGPT (OpenAI) — modèle emblématique de l'IA générative, utilisé pour répondre, écrire, résumer ou traduire des textes médicaux.

https://en.wikipedia.org/wiki/Generative_artificial_intelligence

Run LLMs Locally



with Ollama

Des définitions

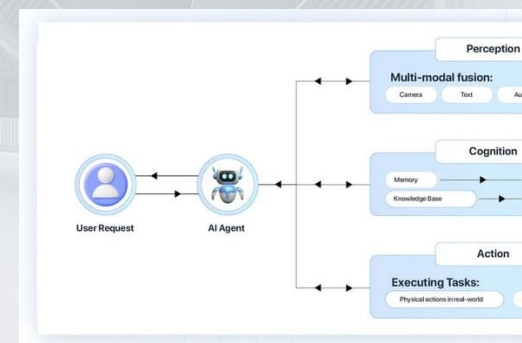
Puis vinrent les agents IA

Définition : Un **agent IA** est une entité logicielle autonome qui **utilise une IA générative**, enrichie de fonctions spécialisées, pour **remplir une mission précise** dans un contexte donné. Il peut interagir avec l'environnement, appeler des outils externes, consulter des bases de données ou suivre un protocole métier. Contrairement à une IA générique comme ChatGPT, un agent IA **est orienté vers un objectif** avec une mémoire et un comportement définis.

Usage médical : Un agent IA dédié à la **gestion pré-consultation**, qui interroge les patients sur leurs symptômes, vérifie les contre-indications, et prépare la consultation avec un résumé pour le médecin.

Référence logicielle : **AutoGPT, OpenAI GPTs ou LangChain Agents** — solutions permettant de créer des agents capables de prendre des décisions, exécuter des tâches en série, et manipuler des documents complexes.

https://en.wikipedia.org/wiki/Intelligent_agent



Des définitions

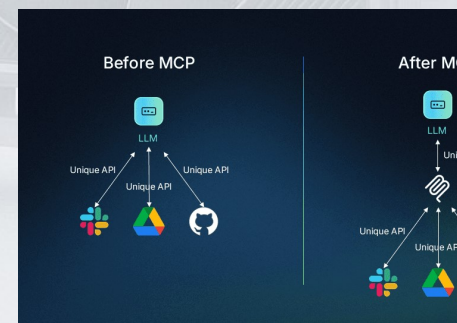
Puis vinrent les MCP

Définition : Les *Model Context Protocols* (MCP) sont des **cadres de structuration** permettant à une IA générative d'adapter ses réponses à un **contexte métier précis**. Ils organisent les données, les instructions et les rôles pour guider le comportement du modèle. Ils permettent ainsi de créer des IA spécialisées (juridique, médical, technique...).

Usage médical : Création d'un agent conversationnel spécialisé en dermatologie ou en télé-expertise pour orienter les généralistes.

Référence logicielle : **OpenAI GPTs (Custom GPT)** — chaque agent peut être configuré avec un *MCP* personnalisé (instructions, fonctions, documents, ton, etc.).

https://en.wikipedia.org/wiki/Model_Context_Protocol



Comment apprend une IA ?

Acquisition des données

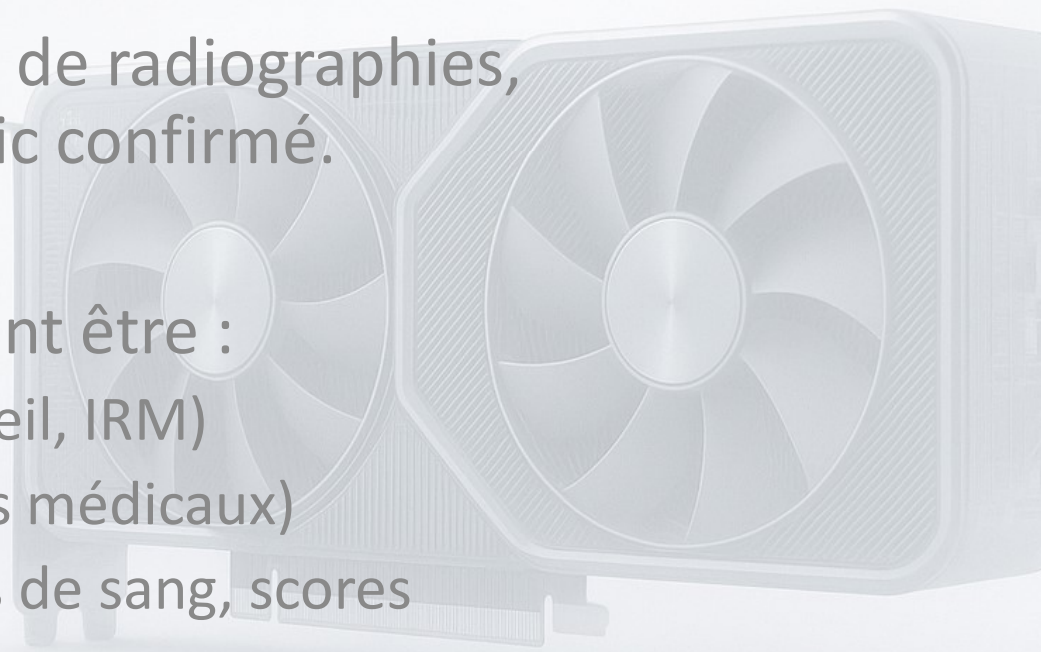
Exemple : 100 000 images de radiographies, chacune avec un diagnostic confirmé.

Les données brutes peuvent être :

Images (scanner, fond d'œil, IRM)

Textes (courriers, dossiers médicaux)

Données chiffrées (prises de sang, scores cliniques)



Comment apprend une IA ?

Prétraitement et étiquetage

Prétraitement et normalisation

Mise à la même taille, élimination des erreurs, standardisation des formats.

Exemple : transformer toutes les images en **256x256 pixels** en niveaux de gris.

Étiquetage (labelling)

Chaque donnée est associée à un **label connu**, par exemple :

Radiographie = *fracture présente* / *fracture absente*

Texte = *diagnostic A* ou *diagnostic B*

```
import tensorflow as tf
mnist = tf.keras.datasets.mnist

(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

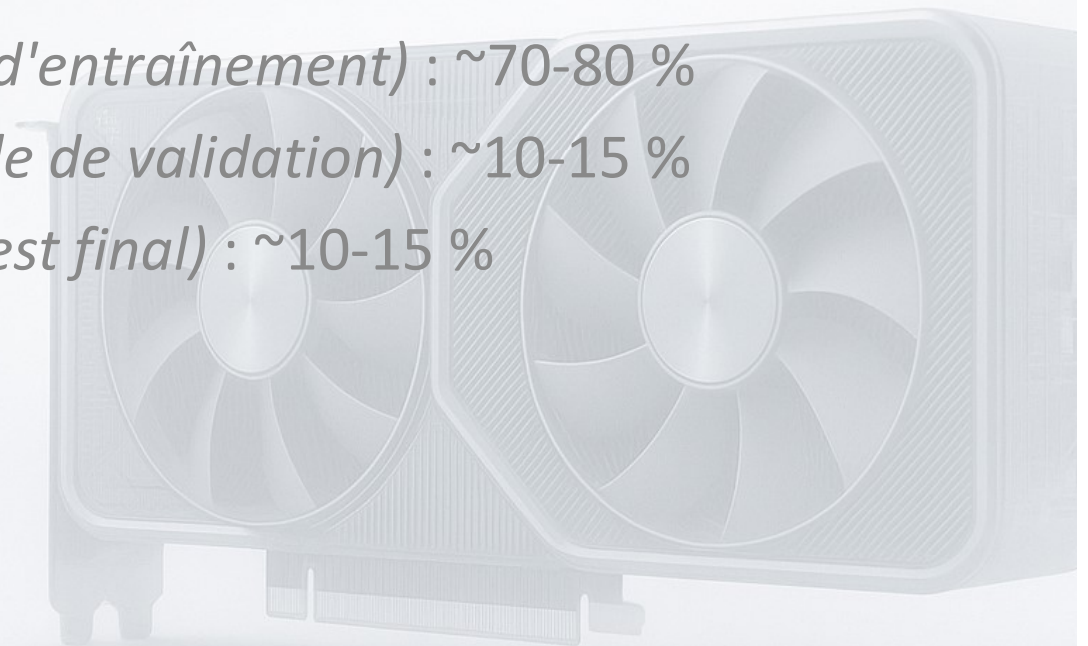
model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation=tf.nn.relu),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test)
```

Comment apprend une IA ?

Séparation des données en groupes

1. Training set (*ensemble d'entraînement*) : ~70-80 %
2. Validation set (*ensemble de validation*) : ~10-15 %
3. Test set (*ensemble de test final*) : ~10-15 %



Comment apprend une IA ?

Étapes d'apprentissage avec TensorFlow

1. Création du modèle

On construit un réseau de neurones artificiels : une **architecture avec des couches** de neurones connectés.

2. Phase d'entraînement ("training")

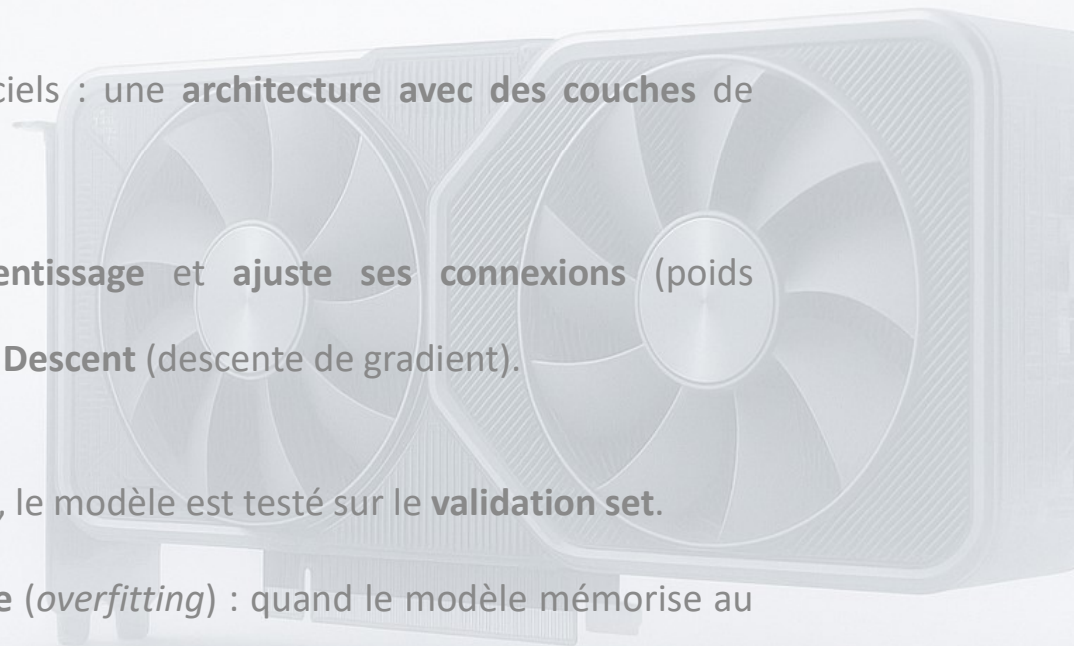
Le modèle **analyse les données d'apprentissage** et **ajuste ses connexions** (poids synaptiques) pour **minimiser les erreurs**.

Utilisation d'un algorithme comme **Gradient Descent** (descente de gradient).

3. Validation croisée

À chaque "époque" (boucle d'entraînement), le modèle est testé sur le **validation set**.

Cela permet de **détecter le surapprentissage** (*overfitting*) : quand le modèle mémorise au lieu de généraliser.

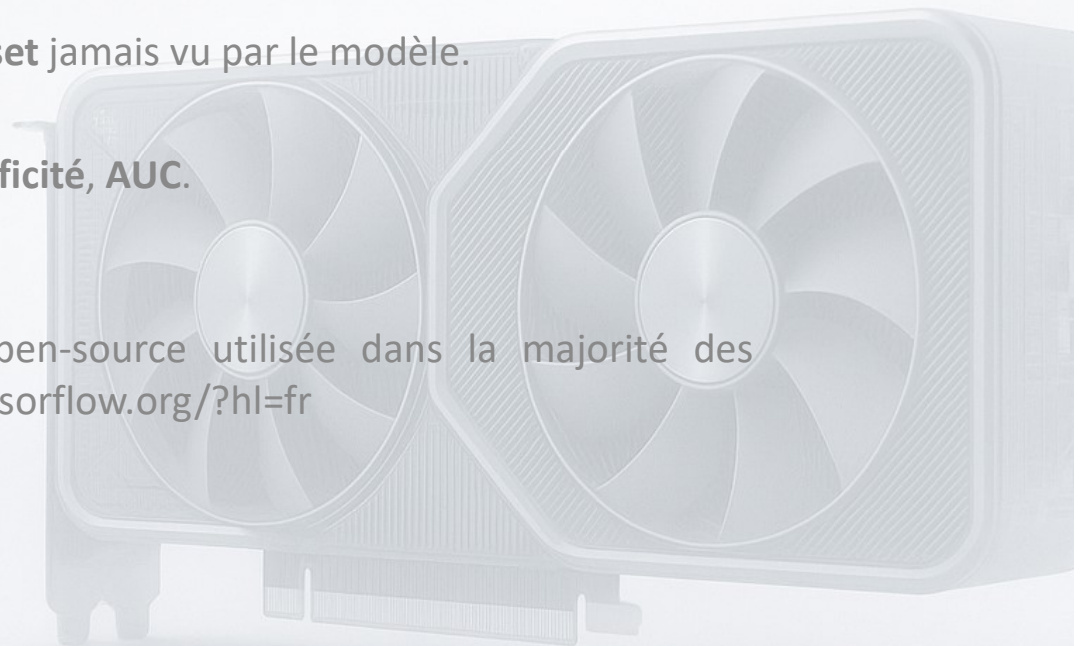


Comment apprend une IA ?

Étapes de test final

1. On évalue la performance sur le **test set** jamais vu par le modèle.
2. Mesures : **précision, sensibilité, spécificité, AUC.**

TensorFlow (Google) – Bibliothèque open-source utilisée dans la majorité des systèmes médicaux d'IA. <https://www.tensorflow.org/?hl=fr>



Quelques points de départ

Outils pour débutants en deep learning

1. Keras

Interface haut niveau pour TensorFlow

Très utilisé pour **prototyper rapidement** des réseaux de neurones

Idéal pour les débutants en deep learning

🔗 <https://keras.io>

2. TensorFlow (Google)

Bibliothèque complète pour l'apprentissage automatique et profond

Compatible avec CPU, GPU, TPU

Permet un **contrôle fin des modèles complexes**

🔗 <https://www.tensorflow.org>

3. PyTorch (Meta)

Alternative à TensorFlow, plus **flexible pour la recherche et le prototypage rapide**

Adoré par les chercheurs en IA

🔗 <https://pytorch.org>

4. Scikit-learn

Pour l'**apprentissage machine classique** (régression, arbres de décision, etc.)

Parfait pour les tâches simples ou les prétraitements

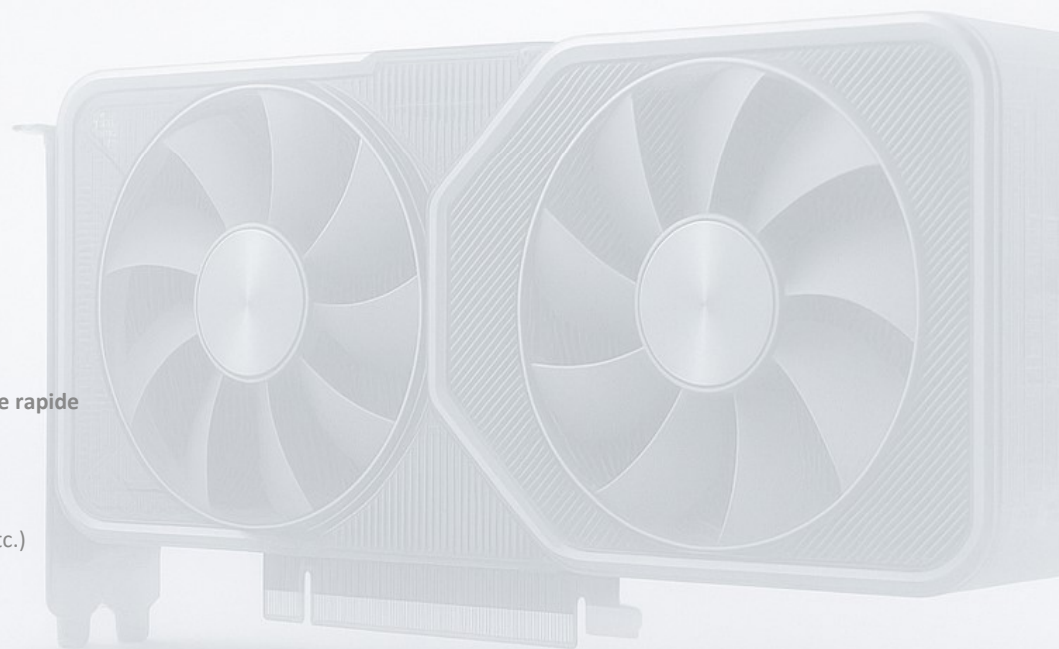
🔗 <https://scikit-learn.org>

5. Hugging Face Transformers

Bibliothèque spécialisée dans les **modèles de langage préentraînés** (comme ChatGPT, BERT, etc.)

Facile à intégrer dans des projets médicaux (analyse de texte, résumé de documents)

🔗 <https://huggingface.co>



Monter sa propre machine Deep Learning

Hardware nécessaire et suffisant pour expérimenter sans cloud /
RGPD Compliant

Configuration matérielle conseillée (budget maîtrisé) :

💻 **Système** : Ubuntu 24.04 LTS (stable, gratuit, bien supporté)

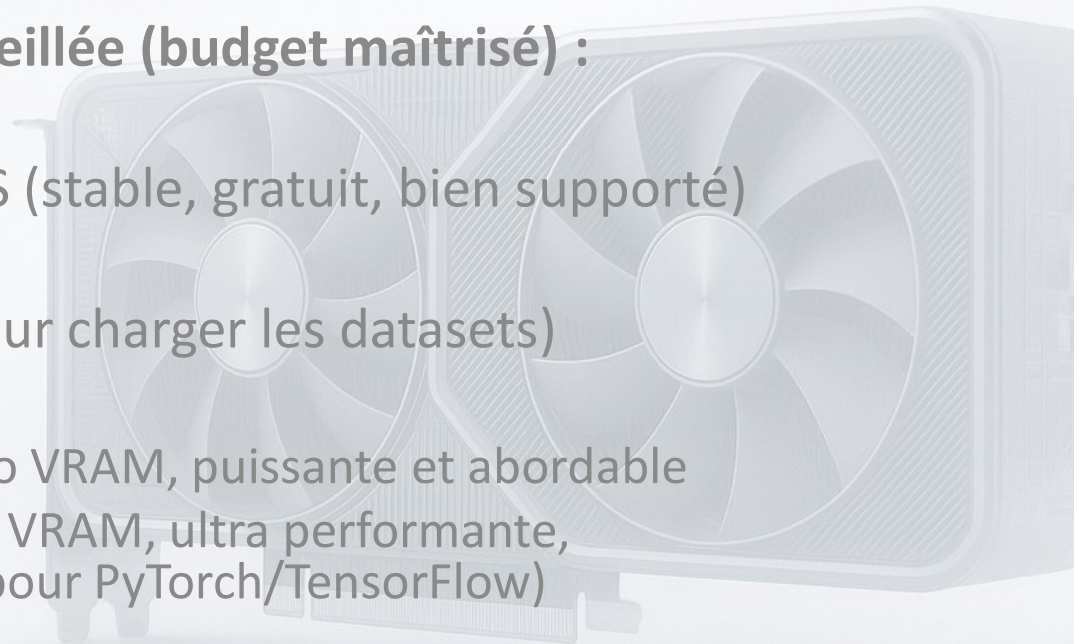
☁️ **RAM** : 32 à 64 Go

💾 **SSD** : 1 To NVMe (rapide pour charger les datasets)

🖥️ **Carte graphique (GPU)** :

◆ **AMD RX 7900 XTX** – 24 Go VRAM, puissante et abordable

◆ **NVIDIA RTX 4090** – 24 Go VRAM, ultra performante,
compatible CUDA (préférée pour PyTorch/TensorFlow)



Monter sa propre machine Deep Learning

Hardware nécessaire et suffisant pour expérimenter sans cloud /
RGPD Compliant



Monter sa propre machine Deep Learning

Hardware nécessaire et suffisant pour expérimenter sans cloud /
RGPD Compliant

Installation logicielle :

Pour AMD (RX7900XTX)

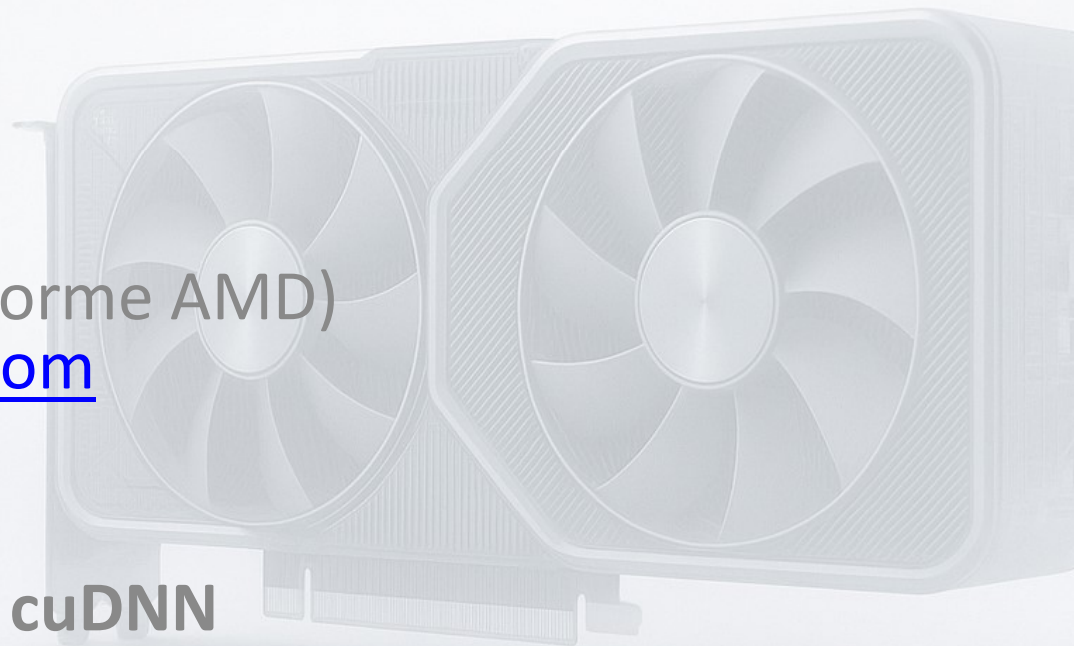
Utiliser ROCm 6.x (plateforme AMD)

<https://rocm.docs.amd.com>

Pour NVIDIA (RTX 4090)

Installer CUDA Toolkit et cuDNN

<https://developer.nvidia.com/cuda-toolkit>



Monter sa propre machine LLM / IA Generative

Un outil simple, opensource, local: ollama

Ollama est un outil open-source qui permet d'exécuter localement des **modèles de langage de grande taille (LLM)** sur votre propre ordinateur, sans dépendre d'Internet ni de services cloud. Conçu pour être simple à utiliser, il fonctionne sur **Linux, macOS et Windows**

Caractéristiques principales :

- **Installation facile** : une seule commande pour démarrer.
- **Exécution locale** : vos données restent sur votre machine, garantissant la confidentialité.
- **Support de nombreux modèles** : DeepSeek-R1, Qwen 3, Llama 3, Mistral, Phi-2, etc.
- **Interface en ligne de commande** : pour interagir avec les modèles.
- **Interface web optionnelle** : pour une utilisation plus conviviale

Pourquoi l'utiliser en médecine ?

- **Confidentialité renforcée** : idéal pour manipuler des données sensibles sans les exposer à des services externes.
- **Autonomie** : permet de tester et d'adapter des modèles selon les besoins spécifiques du praticien ou de l'établissement.
- **Économie** : évite les coûts liés aux services cloud.

 Me



ollama run 1

Monter sa propre machine LLM / IA Générative

BitNet – Encore moins gourmand

<https://github.com/microsoft/BitNet>

<https://huggingface.co/microsoft/bitnet-b1.58-2B-4T>

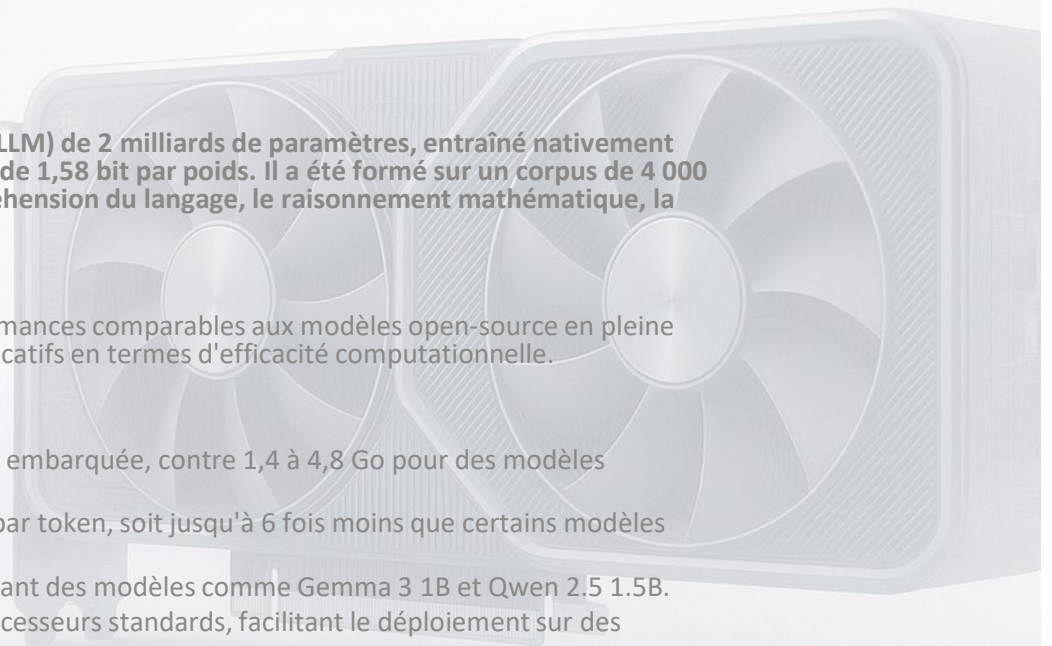
BitNet b1.58 2B4T est un modèle de langage de grande taille (LLM) de 2 milliards de paramètres, entraîné nativement avec des poids ternaires (-1, 0, +1), équivalant à une précision de 1,58 bit par poids. Il a été formé sur un corpus de 4 000 milliards de tokens, couvrant des domaines tels que la compréhension du langage, le raisonnement mathématique, la programmation et les conversations.

Performances :

Malgré sa faible précision, BitNet b1.58 2B4T atteint des performances comparables aux modèles open-source en pleine précision de taille similaire, tout en offrant des avantages significatifs en termes d'efficacité computationnelle.

Avantages clés :

- **Efficacité mémoire** : Utilise seulement 0,4 Go de mémoire non embarquée, contre 1,4 à 4,8 Go pour des modèles concurrents.
- **Consommation énergétique réduite** : Consomme 0,028 joule par token, soit jusqu'à 6 fois moins que certains modèles comparables.
- **Latence faible** : Temps de décodage de 29 ms sur CPU, surpassant des modèles comme Gemma 3 1B et Qwen 2.5 1.5B.
- **Compatibilité CPU** : Peut fonctionner efficacement sur des processeurs standards, facilitant le déploiement sur des appareils à ressources limitées.



Un outil pour développer des agents IA

N8N - Opensource

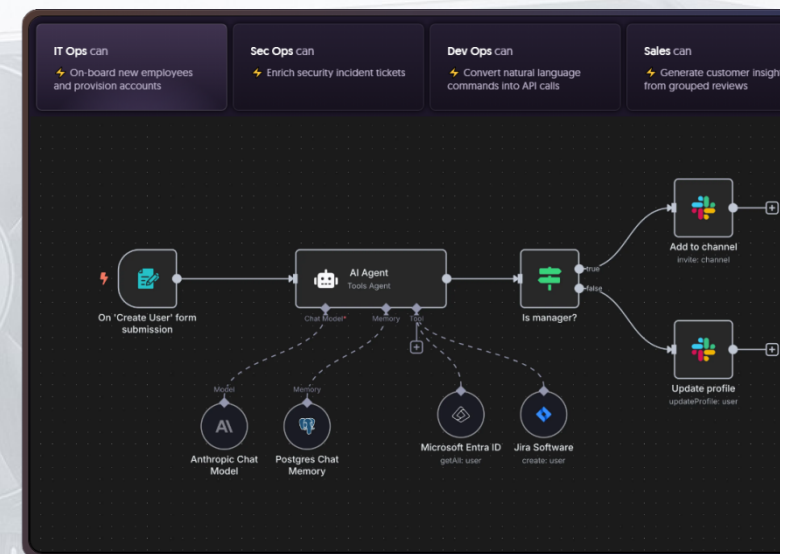
<https://n8n.io/>

N8N est une **plateforme open source d'automatisation de workflows**, créée en 2019.

Elle permet de connecter entre elles **plus de 350 applications** (comme Gmail, Slack, MySQL, Telegram, Power BI, etc.), pour automatiser des tâches répétitives sans écrire de code complexe.

Comment ça fonctionne ?

- Utilise des **nœuds (nodes)** pour créer des scénarios : lire un fichier, envoyer un mail, appeler une API...
- Chaque nœud représente une action dans un **workflow visuel**, que l'on peut assembler comme un schéma logique.
- Offre des **connecteurs HTTP personnalisés** pour intégrer des outils non supportés nativement (via API).



<https://n8n.io/workflows/3694-multi-agent-ai-clinic-management-with-whatsapp-telegram-and-google-calend>

Maintenant que c'est plus clair: Un job killer ?

Et si c'était plutôt un job enhancer ?



Un job replicator

Garder les meilleurs et les moins bons même quand ils sont partis

Exemple : Opportunités IA. Secretariat

Entraînement de systèmes de réponses et d'accueil par jumeau numérique des meilleures employés. adaptation au service et ses spécificités.

Des dizaines de milliers d'heures de conversations téléphoniques.

Avatarisation visuelle ou vidéo des interactions. Interfaçage temps réel avec dossier médical . **Entraînement avec le savoir faire local.**

Autre exemple: réplique/avatarisation des agents administratifs physiques pour multiplier la productivité, garder le savoir faire après départ retraite ou démission.

Un job switcher

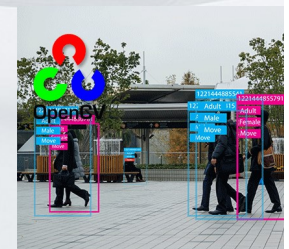
Optimiser le coût du travail et les ressources

- Sortir l'humain cher du travail bureaucratique sans valeur ajoutée
- Repositionner le travailleur sur la relation humaine
- **Permettre une progression de carrière (cycle humain, constitution de la data , deep learning , IA , changement de métier ou progression professionnelle)**
- Eventuellement développer un système de rémunération de l'individu pour avoir aidé à créer l'outil lors de l'avatarisation

Assurer des jobs qu'on ne peut se payer

Amélioration de la sécurité et de l'accueil

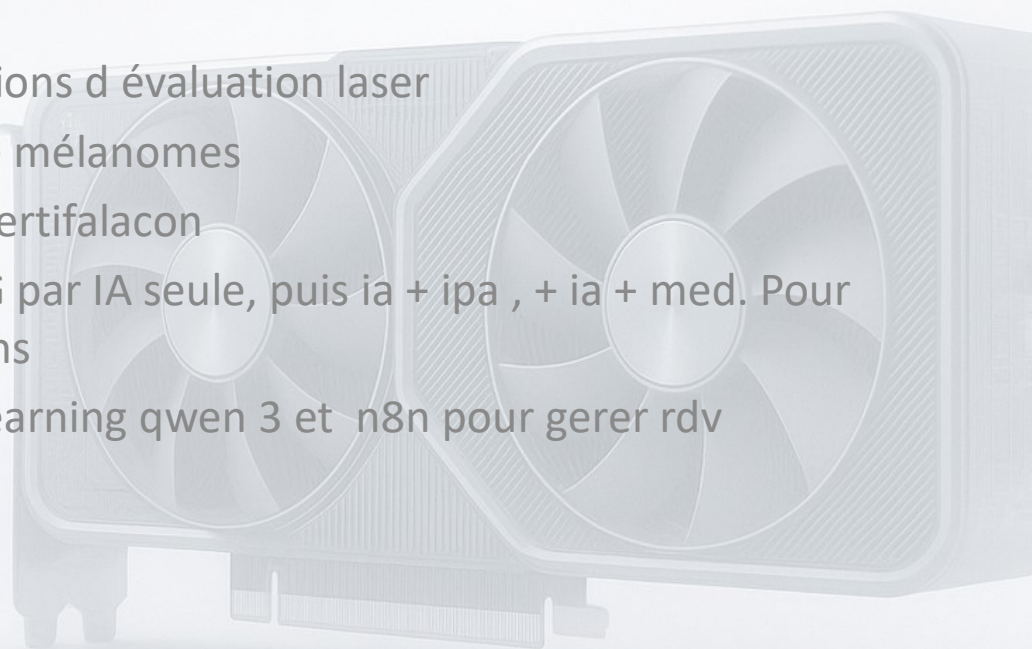
- Analyse auto de situations (vol, attroupement , personne au sol , ...)
- Detection d attente a l accueil (open cv, detection images) et reaction en consequence
- Detection de situations d urgence: chute, courses, violences (dans box / cam thermiques)
- Detection de files d attentes generatrices de tensions a desamorcer
- Detection de patients necessitant prise en charge specifiques ou ressources supplementaires : patient a mobilité reduite, non voyants....
- Detection de comportements anormaux : rixe, degradation salle attente. Detection malaise wc
- Detection degré de remplissage salle attente pour levee de pression et reduction du contentieux.



Mes projets au cabinet

Parce que c'est maintenant accessible

- Avatarisation des premières consultations d'évaluation laser
- Entraînement d'une IA pour dépistage mélanomes
- Développement d'un serveur de télécertification
- Recherche de généralité de télé CS MG par IA seule, puis ia + ipa, + ia + med. Pour multiplier les capacités d'accès aux soins
- Mise en place de mon serveur deep learning qwen 3 et n8n pour gérer rdv



Conclusion

Avec l'IA, on est mort Sauf si...

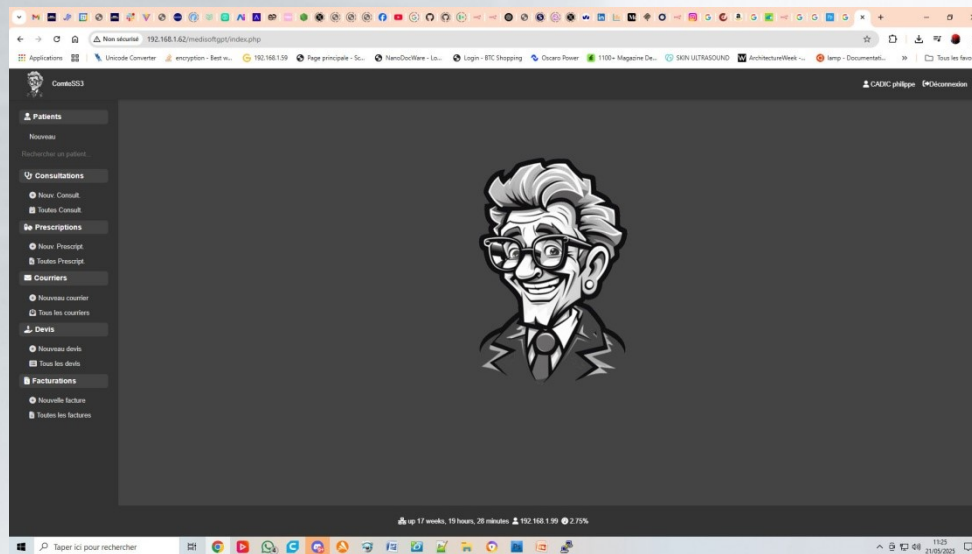
- On garde la **souveraineté des datas et de l'entraînement**
- Si on **développe des IA/Agent locaux avec le savoir faire et la qualité locale**
- Si on développe une **spécialité médicale de cybernétique/AI** pour les médecins (comme on a développé des spécialités médicale quand des progrès de la science ou été faits).
- **Garder la main** , l'éthique médicale, sur les orientations et ne pas laisser ça à des sociétés financières internationales. Avoir des ultra experts médicaux interlocuteurs pour discuter avec le législateur et les tutelles pour ne pas perdre la main.

Conclusion

Avec l'IA, on est bien vivant... parce que

- On peut libérer du temps pour le travail de bas niveau intellectuel
- On peut réduire le risque de faux négatifs pour le dépistage
- On peut améliorer la précocité des diagnostics
- On peut améliorer la sécurité physique et mentale des équipes
- On peut garder (photocopier) l'expertise de ceux qui nous quittent
- On peut avoir une armée de clones infatigables pour favoriser l'accès au soins sans exploser les budgets personnels.
- On peut analyser des données a posteriori pour améliorer les process, découvrir des améliorations, optimiser des coûts ou des profits.
- On peut développer / simuler des outils numériques sans connaissances informatiques et sans interfaçage tech couteux en temps et budget

Merci



Logiciel de gestion de cabinet opensource
Intégralement développé avec l'IA



<https://www.amazon.fr/dp/B0DY6HP6QP>