

JANVIER - 2026

AUTONOMOUS CAR DEV

POWERED BY AI



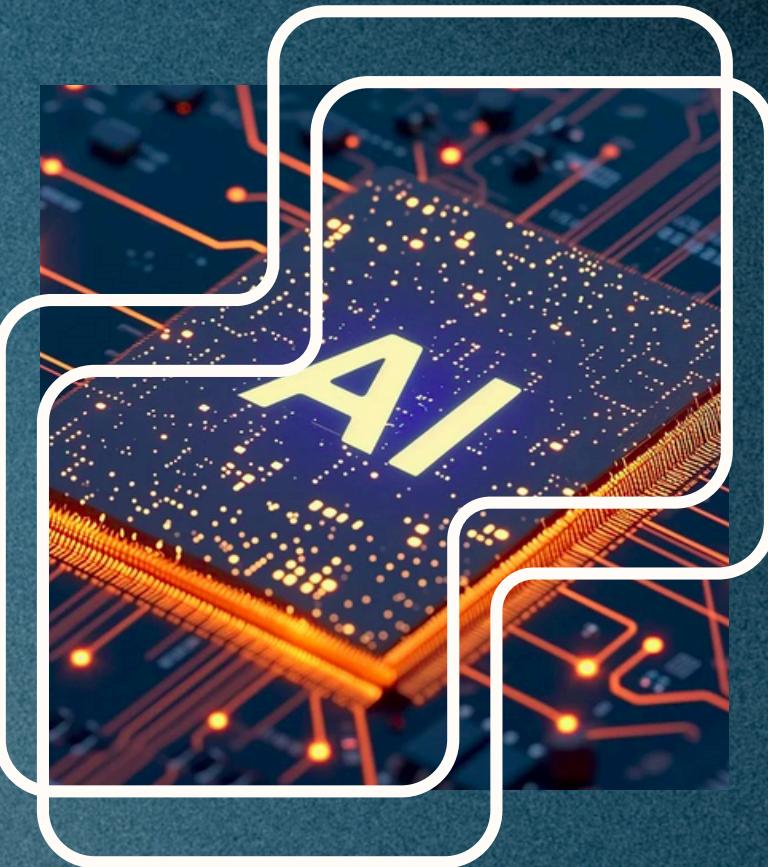
SOMMAIRE

- 1.** Introduction du projet
- 2.** Description du scope
- 3.** Pipeline d'entraînement
- 4.** Simulation et résultats
- 5.** Quantization
- 6.** Pros & Cons

INTRO

du projet

- Créer un système autonome
- Analyser l'environnement et prendre la décision
- Application au domaine automobile
- Embarquement des modèles sur un véhicule miniature (1/10)
 - Image (Unet pattern recognition)
 - Image (Unet 3D monocular)
 - Contrôle du véhicule (RL)
 - Commande vocale (voice learning)
 - Détection d'obstacles (Lidar)
 - GPS imu (fusion de capteurs)



DESCRIPTION

du scope: RL

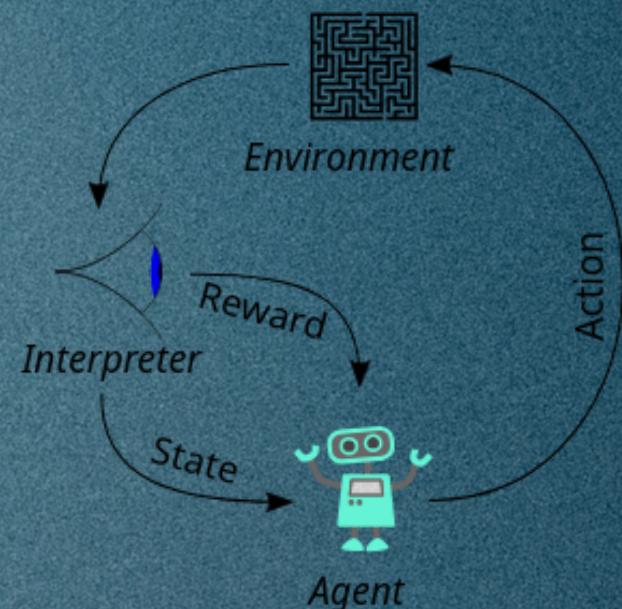
Point central du système

- Utiliser toutes les données fournies par les autres modèles pour décrire l'environnement
- Analyser cet environnement afin de pouvoir prendre une décision
- Exécuter cette action pour contrôler le véhicule

Reinforcement Learning

Reinforcement learning (RL) is a type of machine learning process in which autonomous agents learn to make decisions by interacting with their environment. An autonomous agent is any system that can make decisions and act in response to its environment independent of direct instruction by a human user.

<https://www.ibm.com/think/topics/reinforcement-learning>

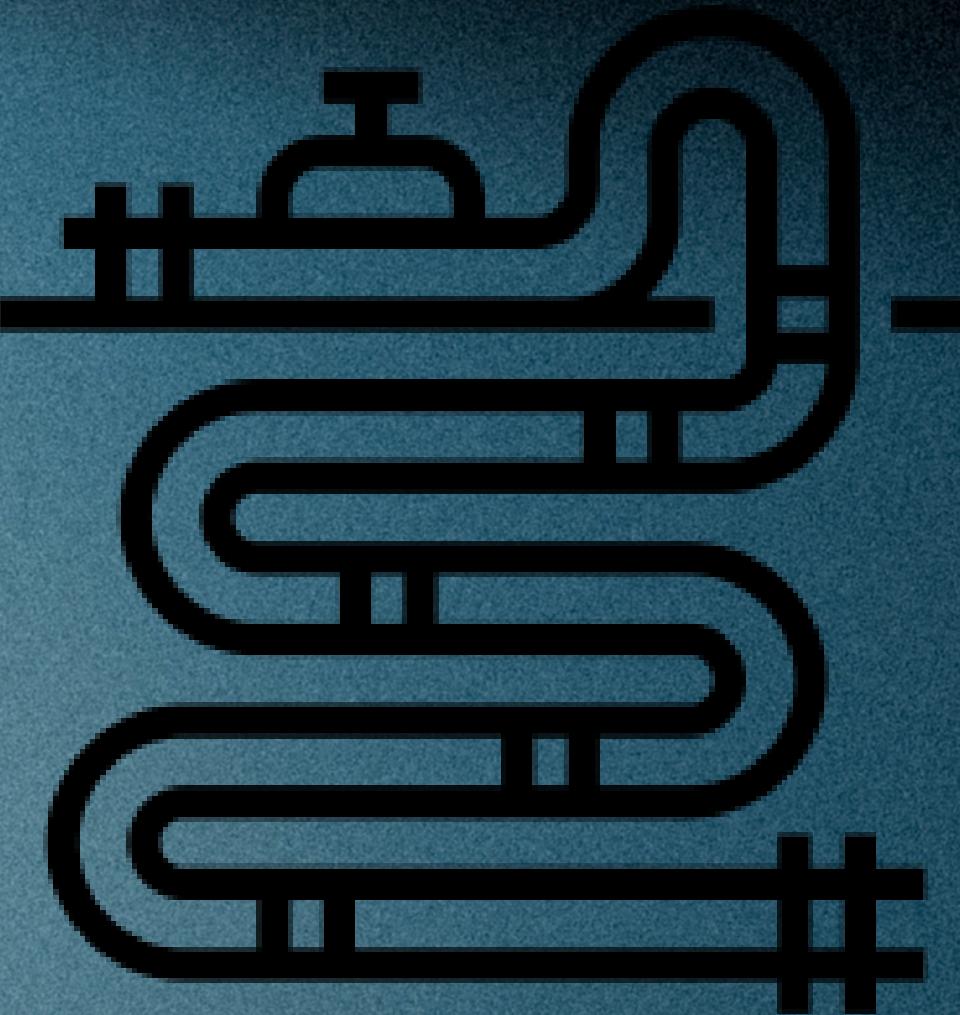


STRATÉGIE

d'entraînement

Pipeline prévu

- Entraînement en simulation en utilisant pytorch
- Vérification de la progression
- Quantization et export onnx
- TensorRT inference et embarquement sur la carte



ENTRAINEMENT

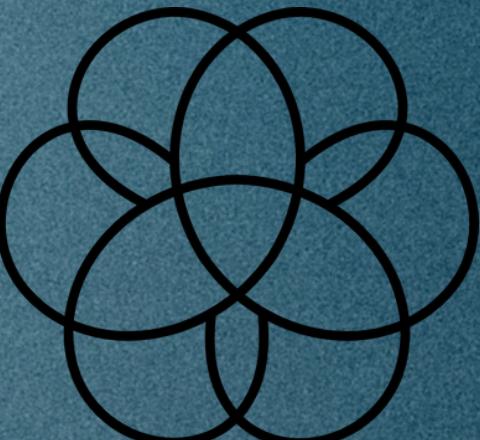
en simulation

Entrainement

- RL: méthode adaptée pour la situation
- PPO: stabilité, robustesse, et adaptabilité
- Policy CNN: Le réseau apprend automatiquement les features utiles (piste, virages, bords) depuis les images RGB
- Reward shaping: reward modulaire basé sur:
 - Rester sur la piste
 - Avancer
 - Éviter les blocages
 - Prioriser une position au centre de la piste
- Entrainement long avec sauvegarde régulière

Simulation

- Environnement : CarRacing-v3
- Pistes générées aléatoirement: pas de base de données d'entraînement
- Étude de l'environnement basée sur l'image
 - Détection du piste/hors piste par proportion de pixels verts



RÉSULTATS

de la simulation

Début d'entraînement

- Un des premiers checkpoints
- Phase d'exploration
- Score négatif
- Comportement aléatoire
- Décisions... redoutables

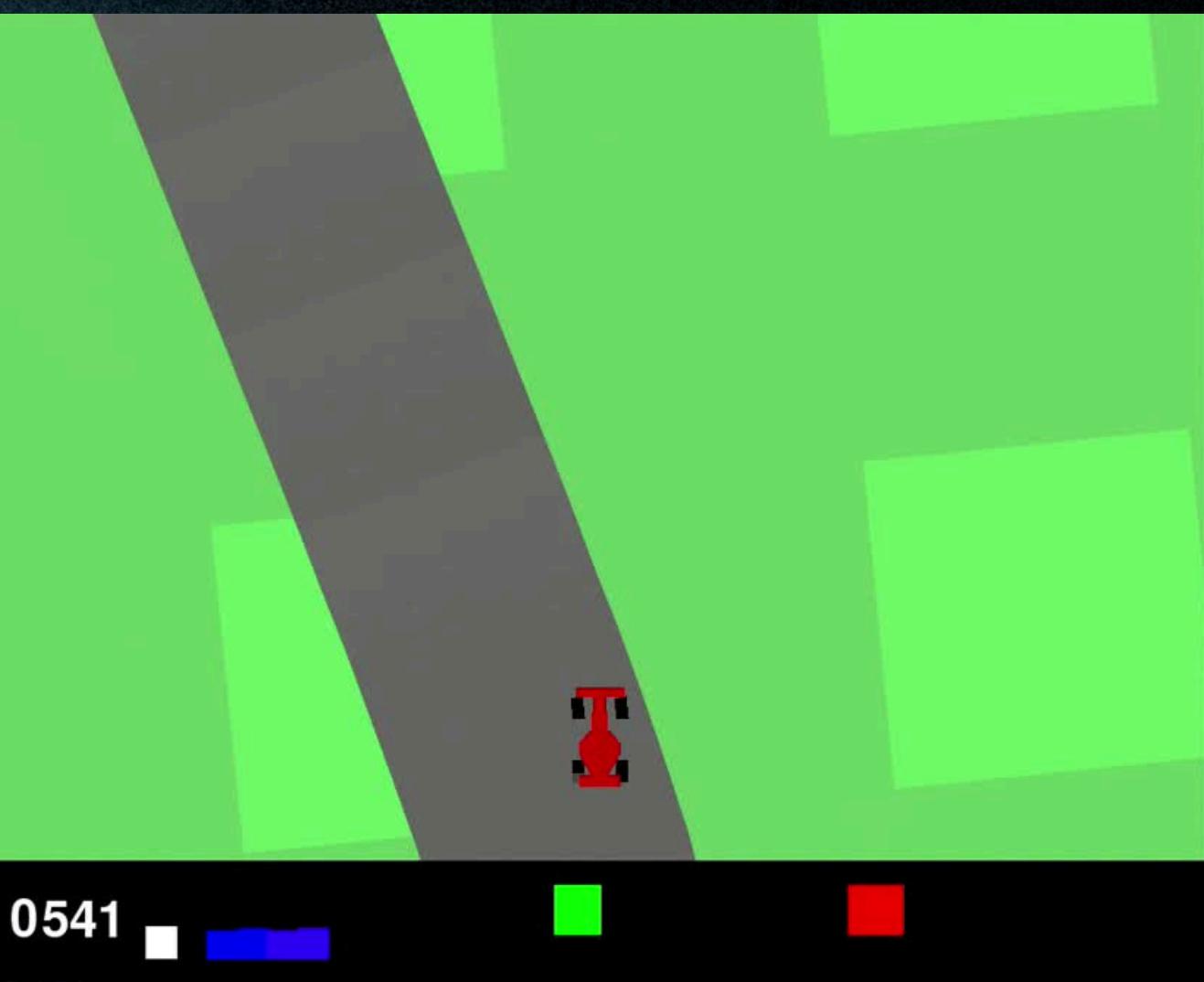


RÉSULTATS

de la simulation

Peak d'entraînement

- Un des meilleurs checkpoints
- Score maximal
- Comportement très correct
- Décisions fiables



RÉSULTATS

de la simulation

Surentrainement

- Un des derniers checkpoints
- Score pas si élevé
- Comportement bizarre
- Accélération pour profiter des récompenses
“avancer”
- Mise à jour des “rewards” pour la sortie hors piste



QUANTIZATION

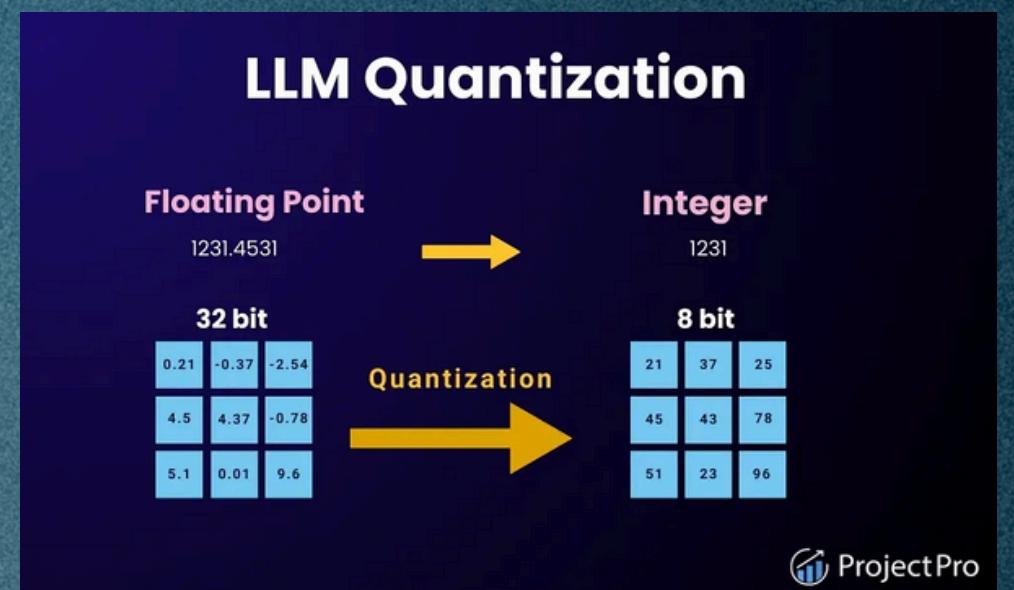
du modèle

Objectif

- Faire tourner un modèle IA plus vite, avec moins de mémoire et moins d'énergie, sur du matériel limité (embarqué).
- Consiste à représenter les poids et/ou activations avec moins de bits (passer de FP32 à INT8)
- Recommandé pour notre contrainte de quota de mémoire, et vis à vis la consommation d'énergie
- Avec la quantization, notre modèle est plus réactif, plus léger et moins énergivore

Mise en place

1. FP32 training
2. QAT preparation (fake quant)
3. QAT training (en réduisant le nombre d'epochs)
4. convert() → vrai INT8
5. Export (ONNX)
6. Déploiement embarqué



PROBLÈMES

& points d'amélioration

Problèmes

- Passage direct au ONNX
- Fichier engine en cours de création

Points d'amélioration

- Détection d'obstacles
- Optimisation de la vitesse
- Ajout de comportement aléatoire (autre voiture)

MERCI