

Projet UCIA

COPIL du 6 janvier 2025
Partie technique

Jean-Luc.Charles@mailo.com

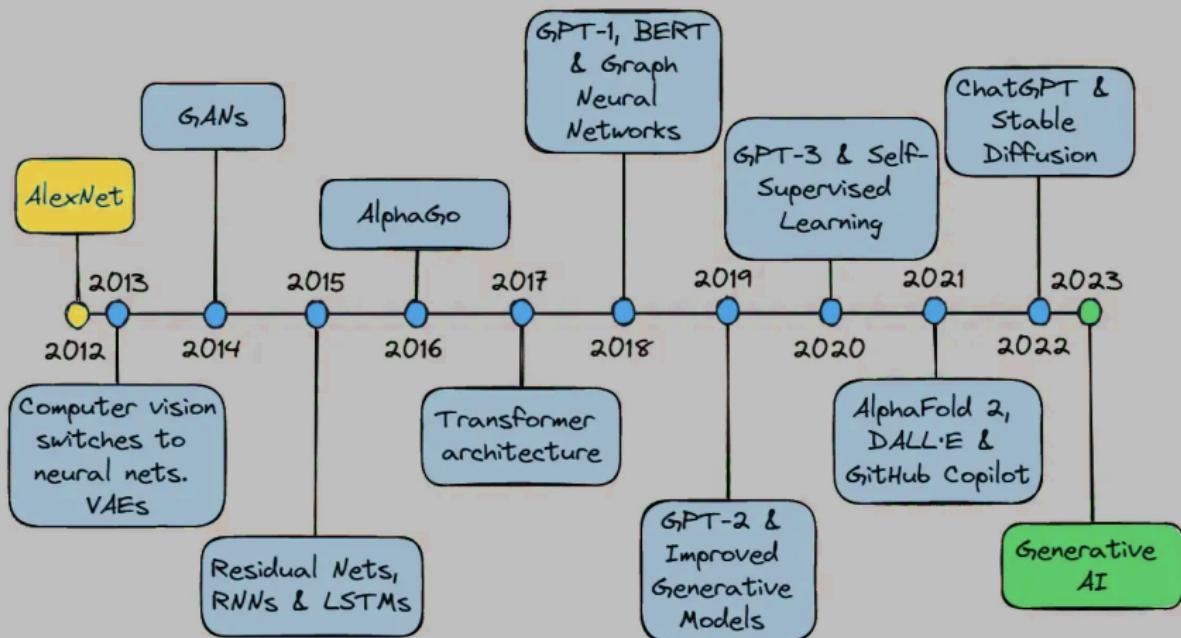




Partie technique

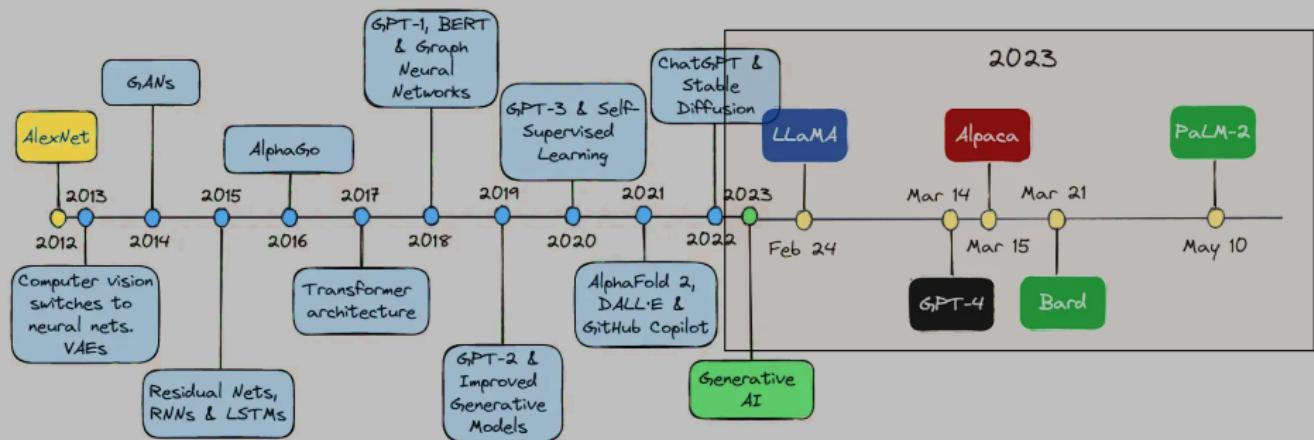
- Concepts et outils de l'IA
- Études menées
- Mise en oeuvre avec la carte RPi4
- Démonstration...

IA : Aspects historiques... l'accélération post 2012



Source: Thomas A Dorfe: "Ten Years of AI in Review", medium.com

IA : Aspects historiques post 2023 : accélération



Source: Thomas A Dorf: "Ten Years of AI in Review", medium.com

Intelligence Artificielles ?



Terme historiquement "mal choisi" ^a

→ le sens actuel reste ambigu : de nombreuses définitions (contradictoires) existent selon les périodes et les auteurs...

^autilisé la première fois en 1956 par John McCarthy, chercheur à Stanford, lors de la conférence de Dartmouth

Intelligence Artificielles ?



Terme historiquement "mal choisi" ^a

→ le sens actuel reste ambigu : de nombreuses définitions (contradictoires) existent selon les périodes et les auteurs...

^autilisé la première fois en 1956 par John McCarthy, chercheur à Stanford, lors de la conférence de Dartmouth.

- "...the science of making computers do things that require intelligence when done by humans." Alan Turing, 1940
 - "the field of study that gives computers the ability to learn without being explicitly programmed." Arthur Samuel, 1960
 - "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E ." Tom Mitchell, 1997
 - Notion of *intelligent agent, rational agent*
"...agent that acts in such a way as to reach the best solution or, in an uncertain environment, the best predictable solution."

Intelligences Artificielles ?

Strong / General AI (IA forte / générale)

Weak / Narrow AI (IA faible / spécialisée)

Multimodal AI (IA multi-modale)

Intelligences Artificielles ?

Strong / General AI (IA forte / générale)

- Système IA qui **pensent comme** les êtres humains, avec la capacité de raisonner en général.
 - Tente aussi d'expliquer comment les êtres humains pensent.
 - **nous n'en sommes pas encore là ?**

Weak / Narrow AI (IA faible / spécialisée)

Multimodal AI (IA multi-modale)

Intelligences Artificielles ?

Strong / General AI (IA forte / générale)

Weak / Narrow AI (IA faible / spécialisée)

- Système IA qui semble **se comporter comme** des être humains.
- Système IA conçu pour des tâches spécifiques.
- Ne renseigne pas sur la façon dont les êtres humains pensent.
- **Nous en sommes déjà là...** nous l'utilisons tous les jours !
(anti-spam, reconnaissance voix/faciale, traduction...)

Multimodal AI (IA multi-modale)

Intelligences Artificielles ?

Strong / General AI (IA forte / générale)

Weak / Narrow AI (IA faible / spécialisée)

Multimodal AI (IA multi-modale)

- Système IA conçu pour traiter des entrées de nature multiple (texte, images, sons...).

Intelligences Artificielles ?



Machine Learning: un champ prometteur de l'IA

Source [medium.com/machine-learning-for-humans/...](https://medium.com/machine-learning-for-humans/)

Machine learning ⊆ artificial intelligence

ARTIFICIAL INTELLIGENCE

Design an intelligent agent that perceives its environment and makes decisions to maximize chances of achieving its goal.

Subfields: vision, robotics, machine learning, natural language processing, planning, ...

MACHINE LEARNING

Gives "computers the ability to learn without being explicitly programmed" (Arthur Samuel, 1959)

SUPERVISED LEARNING

Classification, regression

UNSUPERVISED LEARNING

Clustering, dimensionality reduction, recommendation

REINFORCEMENT LEARNING

Reward maximization

Branches du *Machine Learning*

Entraînement Supervisé (*Supervised learning*)

Un jeu de **données labellisées** est utilisé pour l'entraînement :

- **Classification**

- Classification d'images
- Détection d'objets dans des images
- Reconnaissance de la parole...

- **Régression**

- Prévision de valeur...

- **Détection d'anomalie**

- Anti-spam
- Reconnaissance de défauts (appris)
- Prévision météo...

- ...

Branches du *Machine Learning*

Entraînement Supervisé (*Supervised learning*)

Un jeu de **données labellisées** est utilisé pour l'entraînement :

- **Classification**

- Classification d'images
- Détection d'objets dans des images
- Reconnaissance de la parole...

- **Régression**

- Prévision de valeur...

- **Détection d'anomalie**

- Anti-spam
- Reconnaissance de défauts (appris)
- Prévision météo...

- ...

Branches du *Machine Learning*

Entraînement Supervisé (*Supervised learning*)

Un jeu de **données labellisées** est utilisé pour l'entraînement :

- **Classification**

- Classification d'images
- Détection d'objets dans des images
- Reconnaissance de la parole...

- **Régression**

- Prévision de valeur...

- **Détection d'anomalie**

- Anti-spam
- Reconnaissance de défauts (appris)
- Prévision météo...

- ...

Branches du *Machine Learning*

Entraînement Supervisé (*Supervised learning*)

Un jeu de **données labellisées** est utilisé pour l'entraînement :

- **Classification**

- Classification d'images
- Détection d'objets dans des images
- Reconnaissance de la parole...

- **Régression**

- Prévision de valeur...

- **Détection d'anomalie**

- Anti-spam
- Reconnaissance de défauts (apris)
- Prévision météo...

- ...

Branches du *Machine Learning*

Entraînement non-supervisé (*Unsupervised learning*)

Analyse et groupement de **données non-labellisées**:

● Groupement

- Exploration de données (*Data mining*)
- Regroupement de données WEB
- Analyse de marché
- Analyse de données astronomiques...

● Détection d'anomalie

- Fabrication : détection de défauts (même nouveaux)
- Surveillance d'activité : fraude, défaillance, hacking
- Détection de *fake account* sur Internet...

● Réduction de dimensionnalité

- Compression de données...

● ...

Branches du *Machine Learning*

Entraînement non-supervisé (*Unsupervised learning*)

Analyse et groupement de **données non-labellisées**:

- **Groupement**

- Exploration de données (*Data mining*)
- Regroupement de données WEB
- Analyse de marché
- Analyse de données astronomiques...

- **Détection d'anomalie**

- Fabrication : détection de défauts (même nouveaux)
- Surveillance d'activité : fraude, défaillance, hacking
- Détection de *fake account* sur Internet...

- **Réduction de dimensionnalité**

- Compression de données...

- ...

Branches du *Machine Learning*

Entraînement non-supervisé (*Unsupervised learning*)

Analyse et groupement de **données non-labellisées**:

● Groupement

- Exploration de données (*Data mining*)
- Regroupement de données WEB
- Analyse de marché
- Analyse de données astronomiques...

● Détection d'anomalie

- Fabrication : détection de défauts (même nouveaux)
- Surveillance d'activité : fraude, défaillance, hacking
- Détection de *fake account* sur Internet...

● Réduction de dimensionnalité

- Compression de données...

- ...

Branches du *Machine Learning*

Entraînement non-supervisé (*Unsupervised learning*)

Analyse et groupement de **données non-labellisées**:

● Groupement

- Exploration de données (*Data mining*)
- Regroupement de données WEB
- Analyse de marché
- Analyse de données astronomiques...

● Détection d'anomalie

- Fabrication : détection de défauts (même nouveaux)
- Surveillance d'activité : fraude, défaillance, hacking
- Détection de *fake account* sur Internet...

● Réduction de dimensionnalité

- Compression de données...

- ...

Branches du *Machine Learning*

Entraînement par renforcement

Deep Reinforcement Learning (DRL)

Un *agent* apprend à piloter un *environment* en maximisant une **récompense reward** :

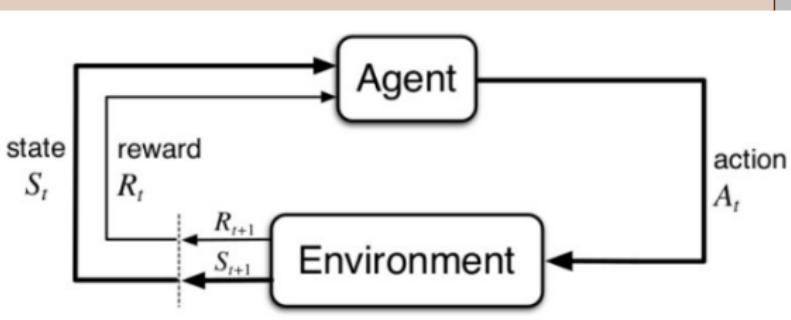
● Contrôle des systèmes

- Contrôle de robots, drones
- Optimisation de process de fabrication
- Négociation financière (boursière)...

● Prise de décision

- jeux vidéo
- Analyse financière...

- ...



Branches du *Machine Learning*

Entraînement par renforcement

Deep Reinforcement Learning (DRL)

Un *agent* apprend à piloter un *environment* en maximisant une **récompense reward** :

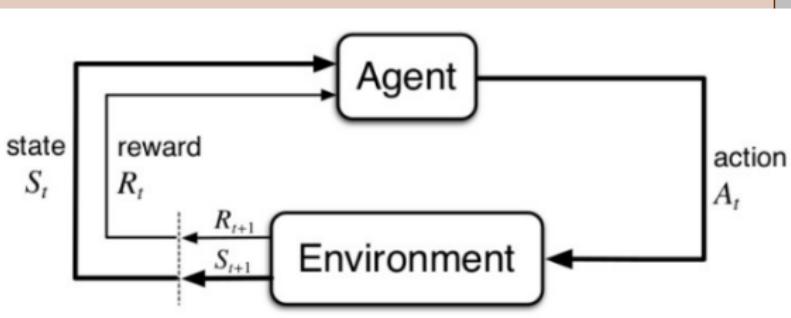
● Contrôle des systèmes

- Contrôle de robots, drones
- Optimisation de process de fabrication
- Négociation financière (boursière)...

● Prise de décision

- jeux vidéo
- Analyse financière...

- ...



Branches du *Machine Learning*

Entraînement par renforcement

Deep Reinforcement Learning (DRL)

Un *agent* apprend à piloter un *environment* en maximisant une **récompense reward** :

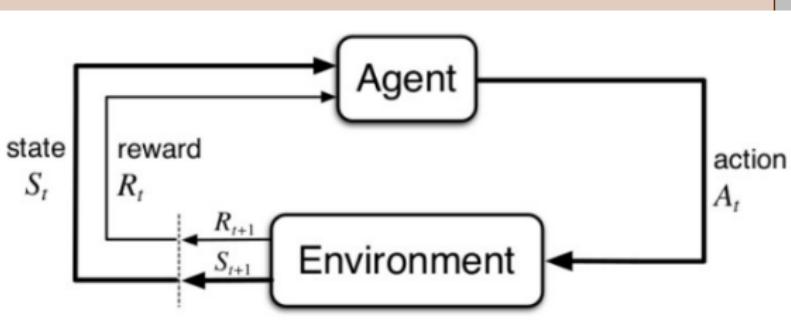
● Contrôle des systèmes

- Contrôle de robots, drones
- Optimisation de process de fabrication
- Négociation financière (boursière)...

● Prise de décision

- jeux vidéo
- Analyse financière...

- ...



Différents algorithmes du *Machine Learning*

Apprentissage Supervisé :

- Réseau de neurones
- Inférence Bayésienne
- Random forest
- Decision Tree
- Support Vector Machine (SVM)
- K-Nearest Neighbor
- Régression Linéaire
- Régression Logistique...

Apprentissage non-supervisé :

- Réseau de neurones
- Principal Composant Analysis
- Singular Value Decomposition
- K-mean & Prob. clustering...

DRL :

- Réseau de neurones (Q-learning, Actor-Critic, DDPG, PPO...)
- Monte Carlo
- SARSA...

Différents algorithmes du *Machine Learning*

Apprentissage Supervisé :

- Réseau de neurones
- Inférence Bayésienne
- Random forest
- Decision Tree
- Support Vector Machine (SVM)
- K-Nearest Neighbor
- Régression Linéaire
- Régression Logistique...

Apprentissage non-supervisé :

- Réseau de neurones
- Principal Composant Analysis
- Singular Value Decomposition
- K-mean & Prob. clustering...

DRL :

- Réseau de neurones
- Monte Carlo
- SARSA...

Différents algorithmes du *Machine Learning*

Apprentissage Supervisé :

- Réseau de neurones
- Inférence Bayésienne
- Random forest
- Decision Tree
- Support Vector Machine (SVM)
- K-Nearest Neighbor
- Régression Linéaire
- Régression Logistique...

Apprentissage non-supervisé :

- Réseau de neurones
- Principal Composant Analysis
- Singular Value Decomposition
- K-mean & Prob. clustering...

DRL :

- Réseau de neurones
- Monte Carlo
- SARSA...

Grande variété de domaines \leadsto succès
des Réseaux de Neurones artificiels

Applications du *Machine Learning*

Vision par ordinateur

- Classification d'images
- Détection d'objects
- Segmentation (sémantique)
- Génération d'images

Les 23 meilleurs générateurs d'images par IA (Gratuits et Pros)

- Estimation de pose
- Transfer de style
- Reconnaissance optique de caractères
(Optical Character Recognition: OCR)
- ...

Applications du *Machine Learning*

Vision par ordinateur

- Classification d'images
- Détection d'objects
- Segmentation (sémantique)
- Génération d'images

Les 23 meilleurs générateurs d'images par IA (Gratuits et Pros)

- Estimation de pose
- Transfer de style
- Reconnaissance optique de caractères
(Optical Character Recognition: OCR)
- ...



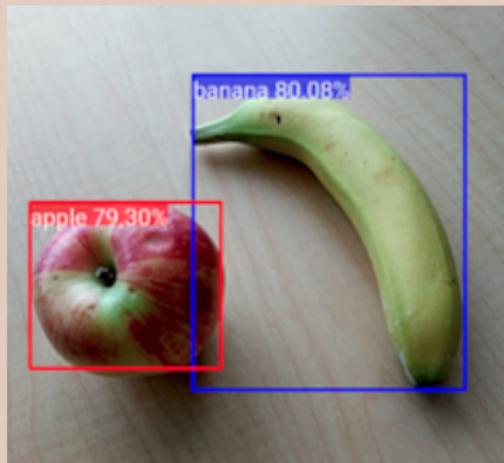
Applications du *Machine Learning*

Vision par ordinateur

- Classification d'images
- Détection d'objects
- Segmentation (sémantique)
- Génération d'images

Les 23 meilleurs générateurs d'images par IA (Gratuits et Pros)

- Estimation de pose
- Transfer de style
- Reconnaissance optique de caractères
(Optical Character Recognition: OCR)
- ...



source : [Tensorflow](#)

Applications du *Machine Learning*

Vision par ordinateur

- Classification d'images
- Détection d'objects
- Segmentation (sémantique)
- Génération
- Les 23 meilleurs g
- Estimation
- Transfer d
- Reconnaiss
- (Optical Char
- ...



source : [Tensorflow](#)

Applications du *Machine Learning*

Vision par ordinateur

- Classification d'images
- Détection d'objects
- Segmentation (sémantique)
- Génération d'images

Les 23 meilleurs générateurs d'images par IA (Gratuits et Pros)

- Estimation de pose
- Transfer de style
- Reconnaissance optique de caractères
(Optical Character Recognition: OCR)
- ...



source : [stylegan](#)

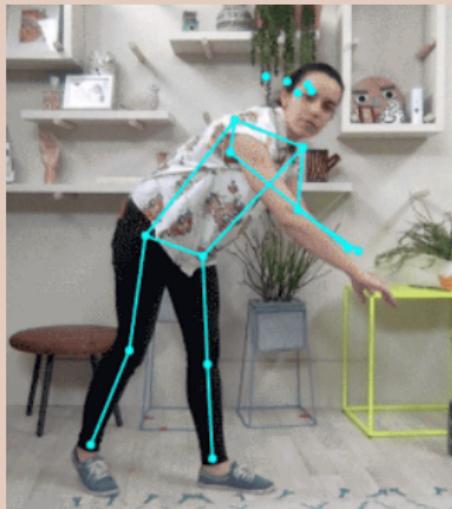
Applications du *Machine Learning*

Vision par ordinateur

- Classification d'images
- Détection d'objects
- Segmentation (sémantique)
- Génération d'images

Les 23 meilleurs générateurs d'images par IA (Gratuits et Pros)

- Estimation de pose
- Transfer de style
- Reconnaissance optique de caractères
(Optical Character Recognition: OCR)
- ...



source : [Tensorflow](#)

Applications du *Machine Learning*

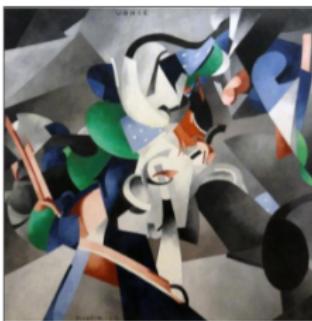
Vision par ordinateur

- Classification
- Détection
- Segmentation
- Génération

Les 23

- Estimation

Content



Style



Pastiche

- Transfer de style
- Reconnaissance optique de caractères
(Optical Character Recognition: OCR)
- ...

source : [Tensorflow](#)

Applications du *Machine Learning*

Vision par ordinateur

- Classification d'images
- Détection d'objets
- Segmentation (sémantique)
- Génération d'images

Les 23 meilleurs générateurs d'images par IA (Gratuits et Pros)

- Estimation de pose
- Transfer de style
- Reconnaissance optique de caractères
(Optical Character Recognition: OCR)
- ...



source : [Tensorflow](#)

Applications du *Machine Learning*

Traitement du langage naturel (*Natural Language Processing*)

- Natural Language Understanding (NLU)
- Reconnaissance de la parole (*Speech recognition*)
- Natural Language Generation (NLG)
- Synthèse de la parole (*Speech Synthesis, Text To Speech*)
- Traduction de langues (*Machine Translation (languages)*)
- Agent Conversationnel (*LLM ChatBots*)
- ...

Applications du *Machine Learning*

Traitement du langage naturel (*Natural Language Processing*)

- Natural Language Understanding (NLU)
- Reconnaissance de la parole (*Speech recognition*)
- Natural Language Generation (NLG)
- Synthèse de la parole (*Speech Synthesis, Text To Speech*)
- Traduction de langues (*Machine Translation (languages)*)
- Agent Conversationnel (*LLM ChatBots*)
- ...

Applications du *Machine Learning*

Traitement du langage naturel (*Natural Language Processing*)

- Natural Language Understanding (NLU)
- Reconnaissance de la parole (*Speech recognition*)
- Natural Language Generation (NLG)
- Synthèse de la parole (*Speech Synthesis, Text To Speech*)
- Traduction de langues (*Machine Translation (languages)*)
- Agent Conversationnel (*LLM ChatBots*)
- ...

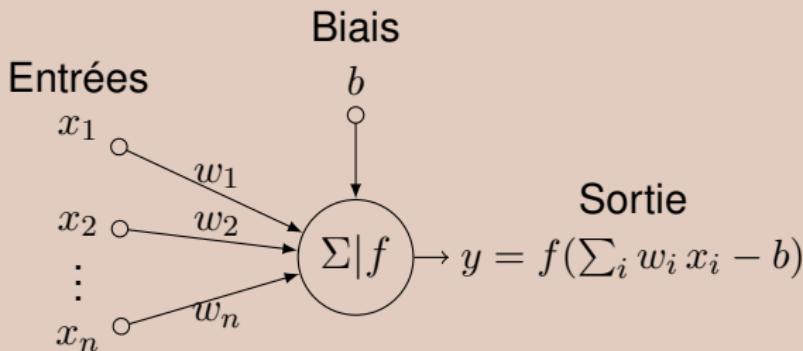
Applications du *Machine Learning*

Traitement du langage naturel (*Natural Language Processing*)

- Natural Language Understanding (NLU)
- Reconnaissance de la parole (*Speech recognition*)
- Natural Language Generation (NLG)
- Synthèse de la parole (*Speech Synthesis, Text To Speech*)
- Traduction de langues (*Machine Translation (languages)*)
- Agent Conversationnel (*LLM ChatBots*)
- ...

Aspects infomatiques

Le modèle du neurone artificiel

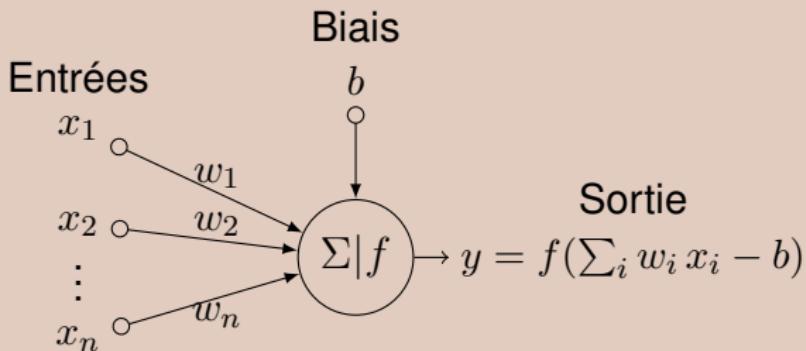


Le **neurone artificiel**:

- reçoit les **entrées** $(x_i)_{i=1..n}$ affectées des **poids** $(w_i)_{i=1..n}$
- calcule la **somme pondérée** de ses entrées : $\sum_i w_i x_i - b$
- donne en **sortie** son activation : $f(\sum_i w_i x_i - b)$ calculée avec sa **activation function** f .

Aspects infomatiques

Le modèle du neurone artificiel

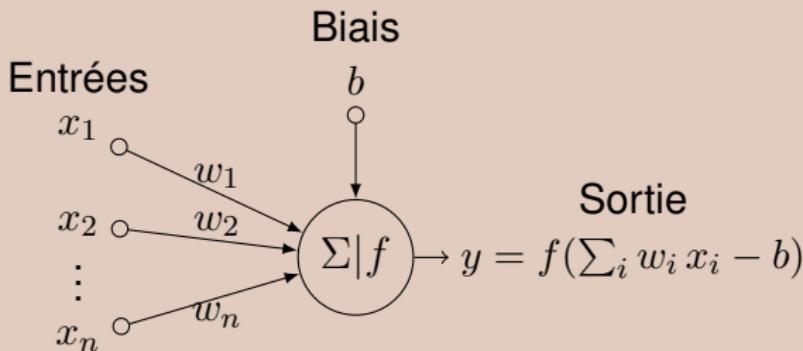


Le **neurone artificiel**:

- reçoit les **entrées** $(x_i)_{i=1..n}$ affectées des **poids** $(w_i)_{i=1..n}$
- calcule la **somme pondérée** de ses entrées : $\sum_i w_i x_i - b$
- donne en **sortie** son activation : $f(\sum_i w_i x_i - b)$ calculée avec sa **activation function** f .

Aspects infomatiques

Le modèle du neurone artificiel

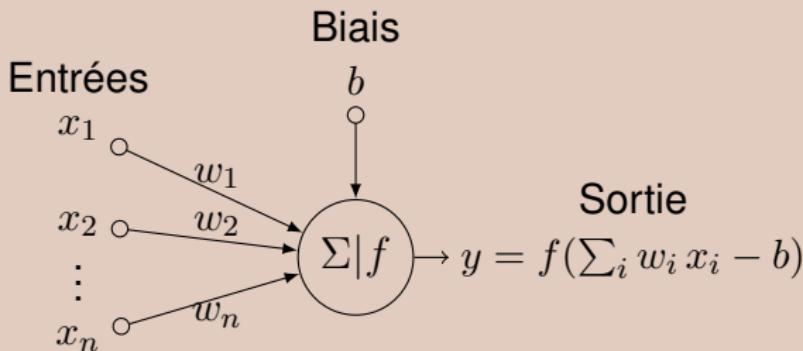


Le **neurone artificiel**:

- reçoit les **entrées** $(x_i)_{i=1..n}$ affectées des **poids** $(w_i)_{i=1..n}$
- calcule la **somme pondérée** de ses entrées : $\sum_i w_i x_i - b$
- donne en **sortie** son activation : $f(\sum_i w_i x_i - b)$ calculée avec sa **activation function** f .

Aspects infomatiques

Le modèle du neurone artificiel

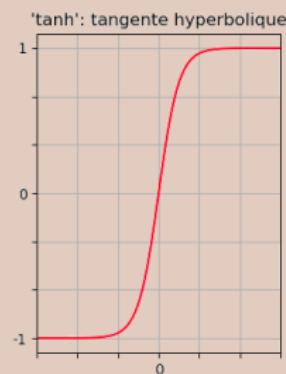
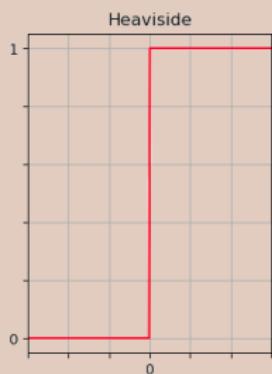


Le **neurone artificiel**:

- reçoit les **entrées** $(x_i)_{i=1..n}$ affectées des **poids** $(w_i)_{i=1..n}$
- calcule la **somme pondérée** de ses entrées : $\sum_i w_i x_i - b$
- donne en **sortie** son activation : $f(\sum_i w_i x_i - b)$ calculée avec sa **activation function** f .

Aspects informatiques

Exemples de fonctions d'activation courantes

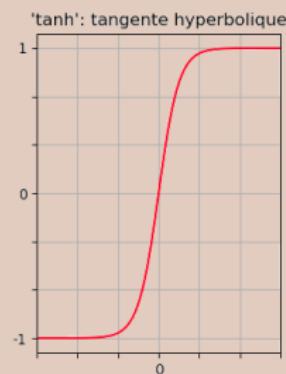
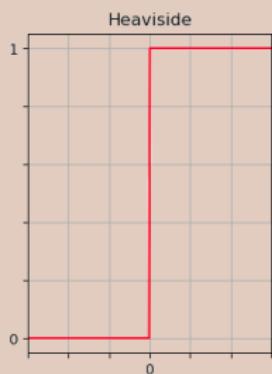


La fonction d'activation:

- Introduit un **comportement non-linéaire**, indispensable pour la réussite de l'apprentissage.
- Fixe la plage de sortie du neurone : $[-1, 1]$, $[0, 1]$, $[0, \infty[$...

Aspects informatiques

Exemples de fonctions d'activation courantes

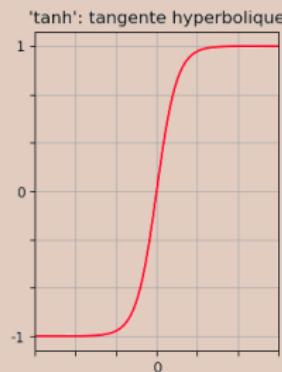


La fonction d'activation:

- Introduit un **comportement non-linéaire**, indispensable pour la réussite de l'apprentissage.
- Fixe la plage de sortie du neurone : $[-1, 1]$, $[0, 1]$, $[0, \infty[$...

Aspects informatiques

Exemples de fonctions d'activation courantes



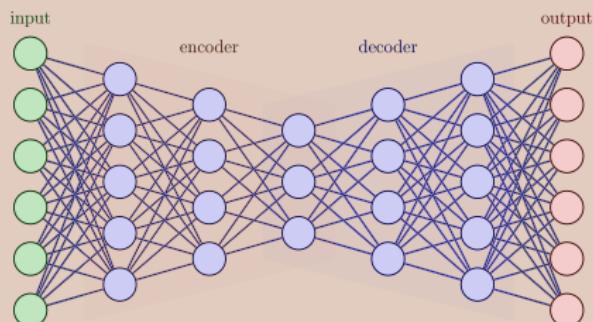
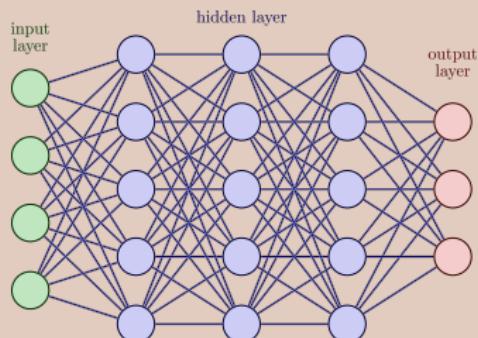
La fonction d'activation:

- Introduit un **comportement non-linéaire**, indispensable pour la réussite de l'apprentissage.
- Fixe la plage de sortie du neurone : $[-1, 1]$, $[0, 1]$, $[0, \infty[$...

Aspects informatiques

Réseau de neurones artificiels

Les neurones sont regroupés en couches pour former un **Réseaux de Neurones** artificiels (RN)



Architecture des Réseaux de Neurones

Architectures dédiées à des applications spécifiques

- *Dense Neural Network*
- *Convolutional (CNN)*
- *Recurrent (RNN)*
- *Auto Encoder (AEN)*
- *Generative Adversarial (GAN)*
- *Transformers*
- *Large Language Model (LLM)*

[Graphique synthétique animé : Saul Dobillas sur Medium]

Architecture des Réseaux de Neurones

Architectures dédiées à des applications spécifiques

- **Dense Neural Network**

La plus simple des architectures faite de couches successives de neurones, utilisant les algorithmes *Feed Forward* et *Back Propagation*

- **Convolutional (CNN)**

- **Recurrent (RNN)**

- **Auto Encoder (AEN)**

- **Generative Adversarial (GAN)**

- **Transformers**

- **Large Language Model (LLM)**

[Graphique synthétique animé : Saul Dobillas sur Medium]

Architecture des Réseaux de Neurones

Architectures dédiées à des applications spécifiques

- **Dense Neural Network**
- **Convolutional (CNN)** - Réseau de Neurones Convolutionnel
Principalement utilisé pour la classification d'images.
- **Recurrent (RNN)**
- **Auto Encoder (AEN)**
- **Generative Adversarial (GAN)**
- **Transformers**
- **Large Language Model (LLM)**

[Graphique synthétique animé : Saul Dobillas sur Medium]

Architecture des Réseaux de Neurones

Architectures dédiées à des applications spécifiques

- **Dense Neural Network**
- **Convolutional (CNN)**
- **Recurrent (RNN)** - Réseau de Neurones Récurent
Traitement des **séries temporelles** (Exemple: *Long Short-Term Memory (LSTM)*).
- **Auto Encoder (AEN)**
- **Generative Adversarial (GAN)**
- **Transformers**
- **Large Language Model (LLM)**

[Graphique synthétique animé : Saul Dobillas sur Medium]

Architecture des Réseaux de Neurones

Architectures dédiées à des applications spécifiques

- **Dense Neural Network**
- **Convolutional (CNN)**
- **Recurrent (RNN)**
- **Auto Encoder (AEN)**- Réseau de Neurones Auto-encodeur
Réduction de dimensionnalité, compression, débruitage, détection d'anomalie...
- **Generative Adversarial (GAN)**
- **Transformers**
- **Large Language Model (LLM)**

[Graphique synthétique animé : Saul Dobillas sur Medium]

Architecture des Réseaux de Neurones

Architectures dédiées à des applications spécifiques

- **Dense Neural Network**
- **Convolutional (CNN)**
- **Recurrent (RNN)**
- **Auto Encoder (AEN)**
- **Generative Adversarial (GAN)**
génération de texte, images, musique...
- **Transformers**
- **Large Language Model (LLM)**

[Graphique synthétique animé : Saul Dobillas sur Medium]

Architecture des Réseaux de Neurones

Architectures dédiées à des applications spécifiques

- **Dense Neural Network**
- **Convolutional (CNN)**
- **Recurrent (RNN)**
- **Auto Encoder (AEN)**
- **Generative Adversarial (GAN)**
- **Transformers**
(2017) Traitement du langage naturel, puis aussi pour la classification d'images.
- **Large Language Model (LLM)**

[Graphique synthétique animé : Saul Dobillas sur Medium]

Architecture des Réseaux de Neurones

Architectures dédiées à des applications spécifiques

- **Dense Neural Network**
- **Convolutional (CNN)**
- **Recurrent (RNN)**
- **Auto Encoder (AEN)**
- **Generative Adversarial (GAN)**
- **Transformers**
- **Large Language Model (LLM)**

lecture de texte, sons, images... génération de texte, livres, images, parole, musique (Exemple: *ChatGPT, LLama...*)

[Graphique synthétique animé : Saul Dobillas sur Medium]

Architecture des Réseaux de Neurones

Architectures dédiées à des applications spécifiques

- **Dense Neural Network**
- **Convolutional (CNN)**
- **Recurrent (RNN)**
- **Auto Encoder (AEN)**
- **Generative Adversarial (GAN)**
- **Transformers**
- **Large Language Model (LLM)**

[Graphique synthétique animé : [Saul Dobillas sur Medium](#)]

Enjeux sociétaux de l'IA : Explicabilité

devenue rapidement une priorité dans la recherche :

- ~ **xAI : Explicable Artificial Intelligence**
- ~ **iML : Interpretable Machine learning**

Explicabilité des Réseaux de Neurones (RN)

- **Inexplicabilité** des résultats calculés par les RN ~ obstacle à leur diffusion.
- L'apprentissage profond avec les RN souvent dénigré comme une "boîte noire" par les scientifiques ayant une "approche cartésienne"...
- La complexité grandissante des RN (LLM par exemple) rend l'explication simple de leurs décisions extrêmement difficile.

Enjeux sociétaux de l'IA : Prise de décision

Prise de décision

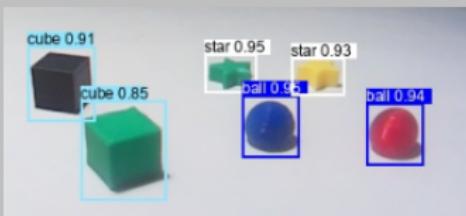
- Un nombre croissant de décisions dans des domaines sensibles (justice pénale, santé, assurance, défense...) confiées aux algorithmes ML source d'inquiétude...
- Les algorithmes de prise de décision reposent inévitablement sur des hypothèses (qualité des données d'apprentissage par exemple) souvent difficile à vérifier.
- Se pose la question de la transparence des algorithmes, des données d'entraînement (*Open Source*).

Enjeux sociaux de l'IA : Certification

Certification

- **Évaluation et Certification** des systèmes IA
 - ~ enjeu majeur pour leur intégration dans l'industrie.
- **Certification** formelle des algorithmes de ML
 - ~ reste aujourd'hui un sujet de recherche :
 - LNE : Certification des processus pour l'IA
 - Fraunhofer : Audit et certification des systèmes d'IA

Étude : entraînement d'un réseau de neurones YOLO pour la détection de petits objets 3D



Détection d'objets 3D

Deux études menées en novembre-décembre 2024:

Étude préliminaire (v1)

Objectifs :

- Faisabilité d'entraîner un réseau YOLO à détecter les objets 3D du projet UCIA
- Tester l'exploitation sur RPi4 d'un réseau YOLO entraîné.

Détection d'objets 3D

Deux études menées en novembre-décembre 2024:

Étude préliminaire (v1)

Objectifs :

- Faisabilité d'entraîner un réseau YOLO à détecter les objets 3D du projet UCIA
- Tester l'exploitation sur RPi4 d'un réseau YOLO entraîné.

Détection d'objets 3D

Deux études menées en novembre-décembre 2024:

Étude préliminaire (v1)

Objectifs :

- Faisabilité d'entraîner un réseau YOLO à détecter les objets 3D du projet UCIA
- Tester l'exploitation sur RPi4 d'un réseau YOLO entraîné.

Configuration :

- Caméra Raspberry standard montée sur trépied
- Entraînement des réseaux sur PC + carte graphique Nvidia
- Exploitation sur RPi4 4Go RAM des réseaux entraînés.

Détection d'objets

Deux études réalisées :

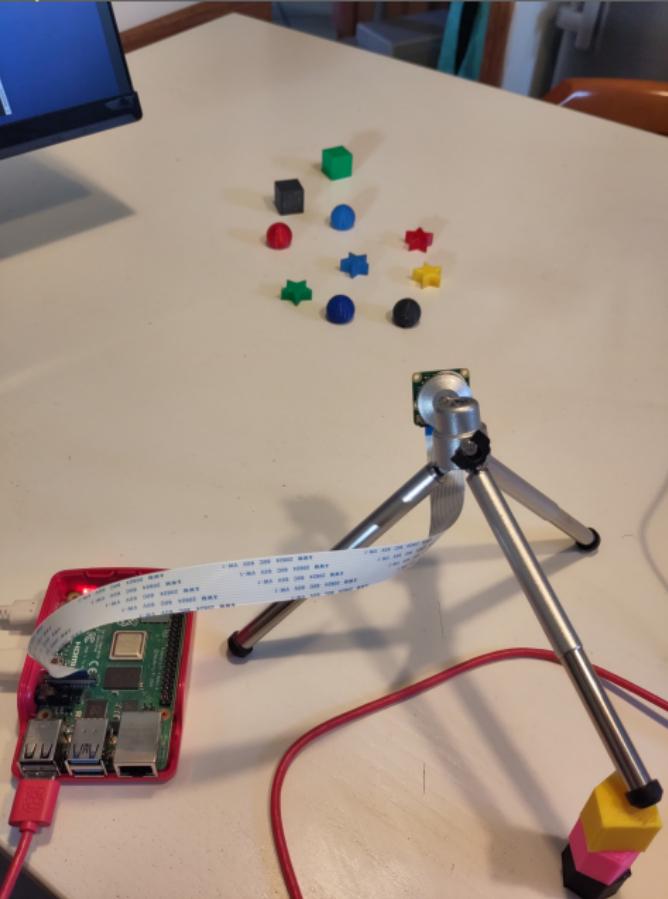
Étude préliminaire

Objectifs :

- Faisabilité de la détection 3D du projet
- Tester l'expérimentation

Configuration :

- Caméra Raspberry Pi
- Entraînement avec GPU
- Exploitation avec GPU



Préparer les objets

en entraîné.

Technique Nvidia entraînés.

Détection d'objets 3D

Étude consolidée (v2)

Objectifs :

- Consolidation de l'entraînement de réseaux YOLO dans les conditions matérielles du CDC du projet UCIA
- Visualisation à distance (navigateur WEB)
- Prototypage Python du déplacement du Thymio.

Détection d'objets 3D

Étude consolidée (v2)

Objectifs :

- Consolidation de l'entraînement de réseaux YOLO dans les conditions matérielles du CDC du projet UCIA
- Visualisation à distance (navigateur WEB)
- Prototypage Python du déplacement du Thymio.

Détection d'objets 3D

Étude consolidée (v2)

Objectifs :

- Consolidation de l'entraînement de réseaux YOLO dans les conditions matérielles du CDC du projet UCIA
- Visualisation à distance (navigateur WEB)
- Prototypage Python du déplacement du Thymio.

Détection d'objets 3D

Étude consolidée (v2)

Objectifs :

- Consolidation de l'entraînement de réseaux YOLO dans les conditions matérielles du CDC du projet UCIA
- Visualisation à distance (navigateur WEB)
- Prototypage Python du déplacement du Thymio.

Configuration :

- Caméra Raspberry grand angle, fixée sur le support Thymio.
- Entraînement des réseaux sur PC + carte graphique Nvidia
- Exploitation sur RPi4 4Go RAM fixée sur le support Thymio.

Détection d'objets 3D

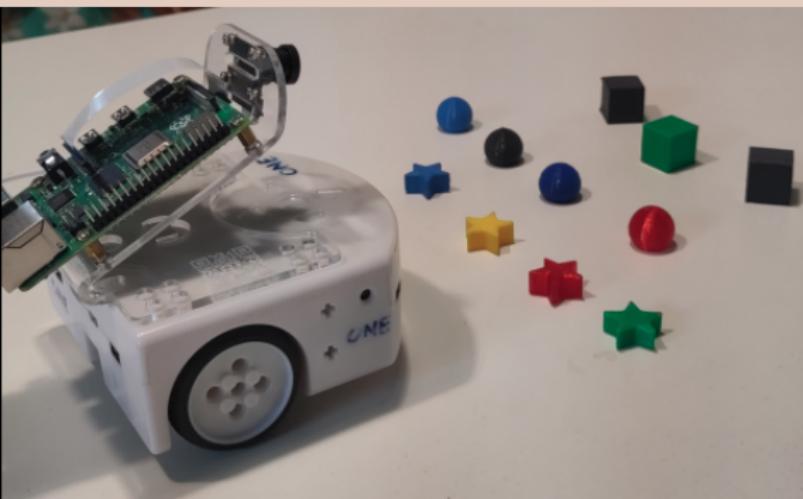
Étude consolidée (v2)

Objectifs :

- Consolider les connaissances sur les conditions d'apprentissage.
- Visualiser les résultats.
- Prototyper une application.

Configuration du robot

- Caméra et GPU.
- Entraînement et exploitation.



dans les

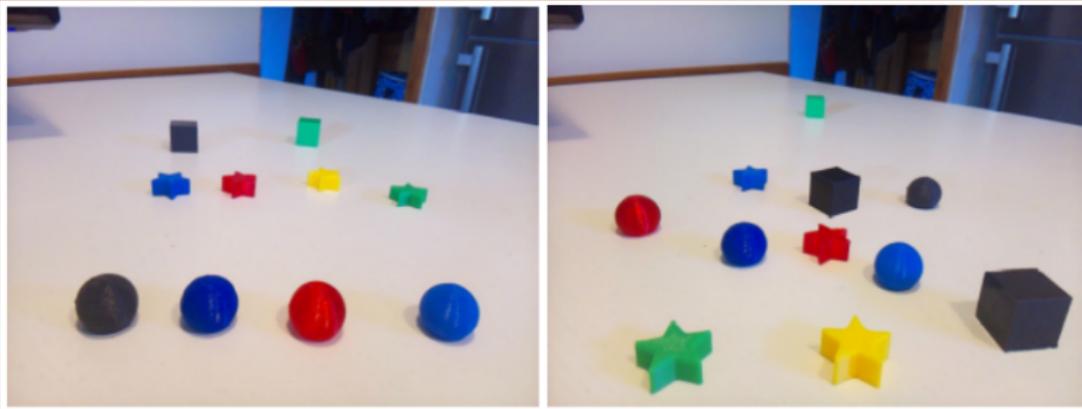
et Thymio.
de Nvidia
Thymio.

Détection d'objets 3D : Banque d'images annotées

Images de l'étude préliminaire (caméra Standard)

Caméra Rasberry Standard, 62 images, 10 objets/images

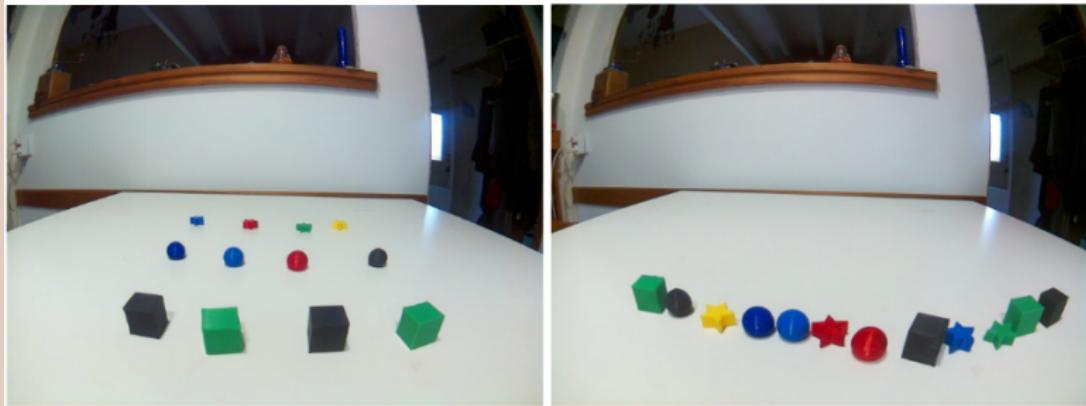
~ 620 objets



Détection d'objets 3D : Banque d'images annotées

Images de l'étude consolidée (caméra Grand Angle)

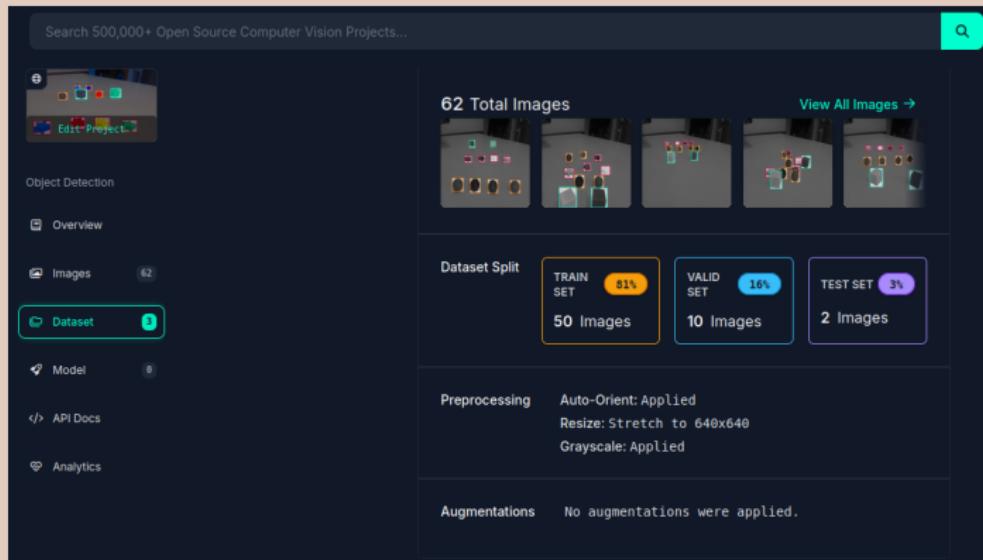
Caméra Raspberry Grand Angle, 200 images, 12 objets/images
~ 2400 objets



Détection d'objets 3D : Banque d'images annotées

Annotation des images (v1)

Images annotées sur le site Roboflow
→ projet [ucia-ia-object-detection](#)



Détection d'objets 3D : Banque d'images annotées

Annotation des images (v2)

Images annotées sur le site Roboflow
→ projet [ucia-ia-object-detection-v2.0](#)

The screenshot shows the Roboflow web interface for managing computer vision datasets. On the left, a sidebar lists project navigation options: Object Detection, Overview, Images (282), Dataset (selected), Model (0), API Docs, and Analytics. The main content area displays the following information:

- 202 Total Images**: Shows five thumbnail images of a scene with colored bounding boxes and labels.
- View All Images →**
- Dataset Split**:
 - TRAIN SET**: 81% (163 Images)
 - VALID SET**: 19% (38 Images)
 - TEST SET**: 0% (1 Images)
- Preprocessing**:
 - Auto-Orient: Applied
 - Resize: Stretch to 640x640
 - Grayscale: Applied
- Augmentations**: No augmentations were applied.

Détection d'objets 3D : réseau de neurones YOLO

Réseau de neurones **YOLO** (*You Only Look Once*)

- Réseau de neurones populaire de **détection d'objets** et de **segmentation** d'images.
- Lancé en 2015, très utilisé (rapidité et précision).
- Réseaux **pré-entraînés** avec les images du WEB :
 - Détection, Segmentation & Pose \sim **MS-COCO** (328000 images, 80 classes d'objets)
 - Classification \sim ImageNet (plus d'un million d'images)
- Les versions de YOLO sont sur le site Ultralytics :
docs.ultralytics.com/fr

Détection d'objets 3D : réseau de neurones YOLO

Versions du réseau YOLO retenues pour l'étude

- **YOLOv8** : version aboutie, stable, parfaitement connue.
- **YOLO11** : inclut les dernières innovations.

Détection d'objets 3D : réseau de neurones YOLO

Versions du réseau YOLO retenues pour l'étude

- **YOLOv8** : version aboutie, stable, parfaitement connue.
- **YOLO11** : inclut les dernières innovations.
- 5 versions de complexité croissante **n, s, m, l et x** :

YOLO...n	→ nano	pour les tâches petites et légères.
YOLO...s	→ small	mise à niveau de nano, meilleure précision.
YOLO...m	→ medium	pour une utilisation à usage général.
YOLO...l	→ large	meilleure précision au prix d'un calcul plus lourd.
YOLO...x	→ Extra-large	pour une précision et des performances maximales.

Détection d'objets 3D : réseau de neurones YOLO

Versions du réseau YOLO retenues pour l'étude

- **YOLOv8** : version aboutie, stable, parfaitement connue.
- **YOLO11** : inclut les dernières innovations.
- 5 versions de complexité croissante **n, s, m, l et x** :

Model	size (pixels)	mAP val 50-95	Speed CPU ONNX (ms)	Speed T4 TensorRT10 (ms)	params (M)	FLOPs (B)
YOLO11n	640	39.5	56.1 ± 0.8	1.5 ± 0.0	2.6	6.5
YOLO11s	640	47.0	90.0 ± 1.2	2.5 ± 0.0	9.4	21.5
YOLO11m	640	51.5	183.2 ± 2.0	4.7 ± 0.1	20.1	68.0
YOLO11l	640	53.4	238.6 ± 1.4	6.2 ± 0.1	25.3	86.9
YOLO11x	640	54.7	462.8 ± 6.7	11.3 ± 0.2	56.9	194.9

Détection d'objets 3D : réseau de neurones YOLO

Versions du réseau YOLO retenues pour l'étude

- **YOLOv8** : version aboutie, stable, parfaitement connue.
- **YOLO11** : inclut les dernières innovations.
- 5 versions de complexité croissante **n**, **s**, **m**, **l** et **x** :

YOLO...n	→ nano	pour les tâches petites et légères.
YOLO...s	→ small	mise à niveau de nano, meilleure précision.
YOLO...m	→ medium	pour une utilisation à usage général.
YOLO...l	→ large	meilleure précision au prix d'un calcul plus lourd.
YOLO...x	→ Extra-large	pour une précision et des performances maximales.

~ Les versions **n** et **s** sont retenues (cible RPi4) :

YOLOv8n, YOLOv8s, YOLO11n, YOLO11s

Détection d'objets 3D : entraînement du réseau YOLO

Choix des méta-paramètres d'entraînement

Méta-paramètres batch et epochs

étude v1

Paramètre	Description	Plage de valeurs
epochs	Nombre de répétitions du processus complet d'entraînement pour converger vers le meilleur état de réseau entraîné	20, 40, 60, 80, 100
batch	Nombre d'images fournies dans un lot d'images d'entraînement	2, 4, 8, 10, 16, 20, 30

étude v2

Paramètre	Description	Plage de valeurs
epochs	Nombre de répétitions du processus complet d'entraînement pour converger vers le meilleur état de réseau entraîné	20, 40, 60, 80, 100
batch	Nombre d'images fournies dans un lot d'images d'entraînement	2, 4, 8, 16, 32

Détection d'objets 3D : entraînement du réseau YOLO

Choix des paramètres d'entraînement

Paramètres imgz, pretrained et seed:

Paramètre	Description	Valeur
imgz	Taille des image (en pixels)	640
patience	Nombre d'époques à attendre sans amélioration des mesures de validation avant d'arrêter l'entraînement. Permet d'éviter le sur-entraînement en arrêtant le processus lorsque les performances atteignent un plateau.	100
pretrained	Détermine s'il faut utiliser un modèle pré-entraîné.	True
seed	Définit la graine aléatoire pour l'entraînement pour garantir la reproductibilité des résultats d'une exécution à l'autre avec les mêmes configurations.	1234
workers	Nombre de threads de travail pour le chargement des données.	0

Détection d'objets 3D : entraînement du réseau YOLO

La combinaison des méta-paramètres donne de nombreux entraînements à stocker :

$4 \times 7 \times 5 = 140$ entraînements pour l'étude v1, 100 pour l'étude v2.

Détection d'objets 3D : entraînement du réseau YOLO

La combinaison des méta-paramètres donne de nombreux entraînements à stocker :

$4 \times 7 \times 5 = 140$ entraînements pour l'étude v1, 100 pour l'étude v2.

Nommage des dossiers d'entraînement

Training/YOLO-trained-V2/UCIA-YOLOvvv/batch-BB_epo-EEE

vvv ~ version du réseau YOLO : (**v8n, v8s, 11n, 11s**)

BB ~ valeur du méta-paramètre batch : (**02, 04, 08, 16, 32**)

EEE ~ valeur du méta-paramètre epochs : (**020, 040, 060, 080, 100**)

Détection d'objets 3D : entraînement du réseau YOLO

La combinaison des méta-paramètres donne de nombreux entraînements à stocker :

$4 \times 7 \times 5 = 140$ entraînements pour l'étude v1, 100 pour l'étude v2.

Nommage des dossiers d'entraînement

Training/YOLO-trained-V2/UCIA-YOLOvvv/batch-BB_epo-EEE

vvv ~ version du réseau YOLO : (**v8n, v8s, 11n, 11s**)

BB ~ valeur du méta-paramètre batch : (**02, 04, 08, 16, 32**)

EEE ~ valeur du méta-paramètre epochs : (**020, 040, 060, 080, 100**)

Programmes Python d'entraînement des réseaux YOLO

train_YOLOv8.py et **train_YOL011.py** développés pour l'étude V1 puis V2 (github.com/cjlux/UCIA_ObjectDetection).

Détection d'objets 3D : entraînement du réseau YOLO

Les entraînements sont faits sur un PC équipé d'une carte graphique **Nvidia Quadro TRX8000**.

Contenu des dossiers

UCIA-YOLOvvv/batch-BB_epo-EEE/

- Extraits des images d'entraînement
- Résultats avec les images de validation
- Fichier `confusion_matrix.png` : matrice de confusion
- Fichier `results.png` : tracé des statistiques et métriques d'entraînement

Détection d'objets 3D : entraînement du réseau YOLO

Les entraînements sont faits sur un PC équipé d'une carte graphique **Nvidia Quadro TRX8000**.

Contenu des dossiers

UCIA-YOLOvvv/batch-BB_epo-EEE/

- Extraits des images d'entraînement
- Résultats avec les images de validation
- Fichier `confusion_matrix.png` : matrice de confusion
- Fichier `results.png` : tracé des statistiques et métriques d'entraînement

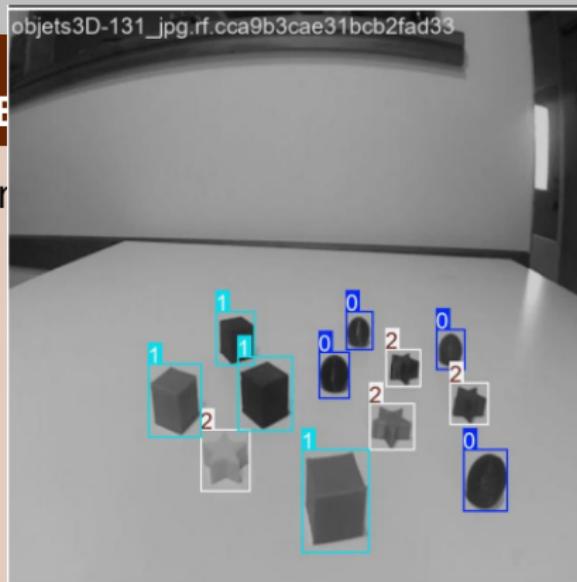
Détection d'objets 3D : entraînement du réseau YOLO

Les entraînements sont faits sur un PC équipé d'une carte graphique **Nvidia Quadro TRX8000**.

Contenu des dossiers

UCIA-YOLOvvv/batch-BB_epo-EEE

- Extraits des images d'entraînement



- Résultats avec les images de validation

- Fichier `confusion_matrix.png` : matrice de confusion

Détection d'objets 3D : entraînement du réseau YOLO

Les entraînements sont faits sur un PC équipé d'une carte graphique **Nvidia Quadro TRX8000**.

Contenu des dossiers

UCIA-YOLOvvv/batch-BB_epo-EEE/

- Extraits des images d'entraînement
- Résultats avec les images de validation
- Fichier `confusion_matrix.png` : matrice de confusion
- Fichier `results.png` : tracé des statistiques et métriques d'entraînement

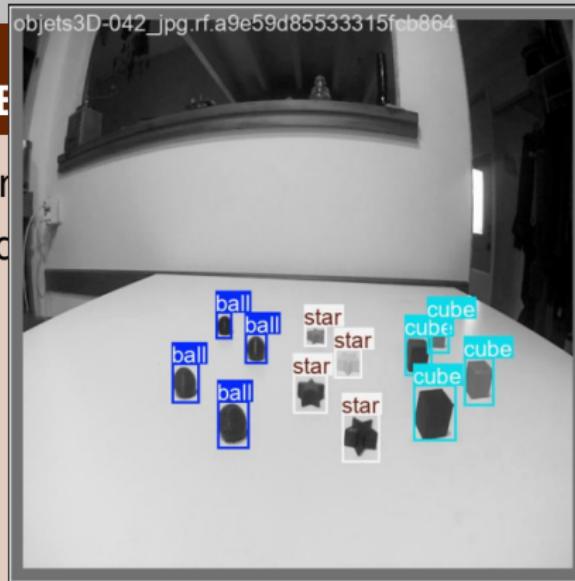
Détection d'objets 3D : entraînement du réseau YOLO

Les entraînements sont faits sur un PC équipé d'une carte graphique **Nvidia Quadro TRX8000**.

Contenu des dossiers

UCIA-YOLOvvv/batch-BB_epo-EEE

- Extraits des images d'entraînement
- Résultats avec les images de validation



- Fichier `confusion_matrix.png` : matrice de confusion
- Fichier `results.png` : tracé des statistiques et métriques

Détection d'objets 3D : entraînement du réseau YOLO

Les entraînements sont faits sur un PC équipé d'une carte graphique **Nvidia Quadro TRX8000**.

Contenu des dossiers

UCIA-YOLOvvv/batch-BB_epo-EEE/

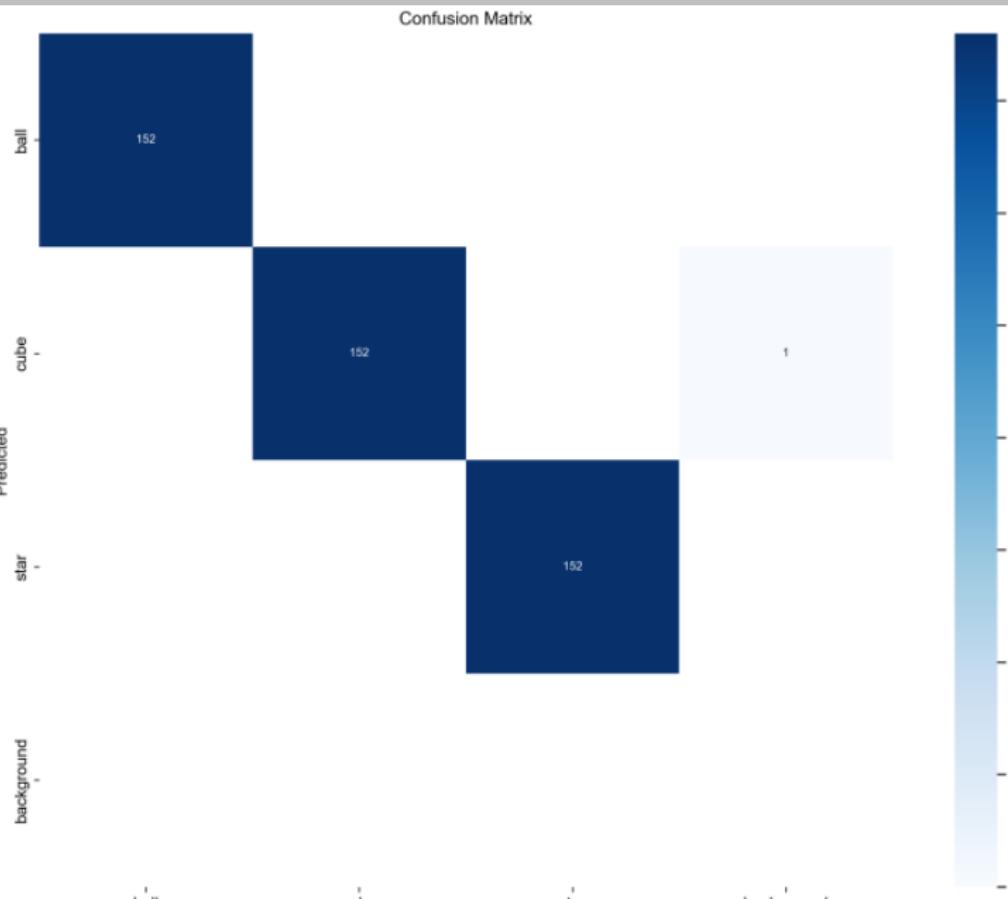
- Extraits des images d'entraînement
- Résultats avec les images de validation
- Fichier `confusion_matrix.png` : matrice de confusion
- Fichier `results.png` : tracé des statistiques et métriques d'entraînement

Détection d'objets

Les entraîneurs graphique N

Contenu de
UCIA-YOLO

- Extraits de code
- Résultats
- Fichier config



Détection d'objets 3D : entraînement du réseau YOLO

Les entraînements sont faits sur un PC équipé d'une carte graphique **Nvidia Quadro TRX8000**.

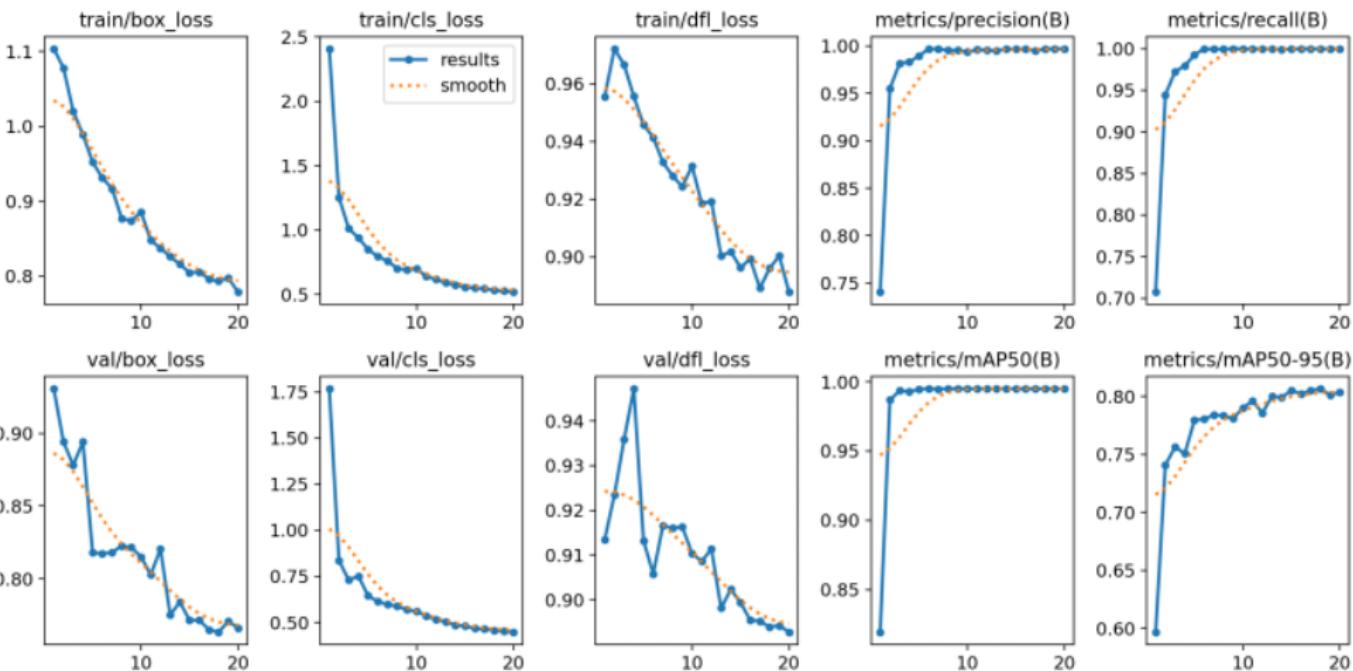
Contenu des dossiers

UCIA-YOLOvvv/batch-BB_epo-EEE/

- Extraits des images d'entraînement
- Résultats avec les images de validation
- Fichier `confusion_matrix.png` : matrice de confusion
- Fichier `results.png` : tracé des statistiques et métriques d'entraînement

Détection d'objets 3D : entraînement du réseau YOLO

Les entraînements sont faits sur un PC équipé d'une carte



Détection d'objets 3D : entraînement du réseau YOLO

Les entraînements sont faits sur un PC équipé d'une carte graphique **Nvidia Quadro TRX8000**.

Contenu des dossiers

UCIA-YOLOvvv/batch-BB_epo-EEE/

- Extraits des images d'entraînement
- Résultats avec les images de validation
- Fichier `confusion_matrix.png` : matrice de confusion
- Fichier `results.png` : tracé des statistiques et métriques d'entraînement

Détection d'objets 3D : entraînement du réseau YOLO

Contenu des dossiers

UCIA-YOLOvvv/batch-BB_epo-EEE/weights

- **best.pt** : fichier binaire des poids du réseau entraîné (au format **pytortch**).
- Formats optimisés pour RPi4 :
 - **best.onnx** : poids du réseau entraîné au format **ONNX**
 - **best.ncnn** : poids du réseau entraîné au format **NCNN**

Détection d'objets 3D : Performances des réseaux entraînés

- Le module Python **ultralytics** fournit des fonctions pour évaluer les métriques de performance de détection d'objets.
- Programmes Python `eval_YOLOv8.py` et `eval_YOLO11.py` développés (V1 et V2) pour comparer les versions d'entraînement des réseaux YOLO.
- ~ 4 fichiers résultats (exemple pour V2):
`results_yolov8n-V2.txt`, `results_yolov8s-V2.txt`,
`results_yolo11n-V2.txt`, `results_yolo11s-V2.txt`,
contenant chacun les métriques des entraînement pour les différentes valeurs des méta-paramètres.

Détection d'objets 3D : Performances des réseaux entraînés

Temps d'inférence

Temps moyen d'inférence sur le PC de calcul (carte Nvidia)

Réseau	Temps moyen d'inférence[ms]
yolov8n	2
yolo11n	2
yolov8s	4
yolo11s	4

~> Architecture s deux fois plus lente que n

Détection d'objets 3D : Performances des réseaux entraînés

Précision des réseaux entraînés

Temps moyen d'inférence sur le PC de calcul (carte Nvidia)

Réseau	Temps moyen d'inférence[ms]
yolov8n	2
yolo11n	2
yolov8s	4
yolo11s	4

~> Architecture **s** deux fois plus lente que **n**

Configuration RPi4-UCIA

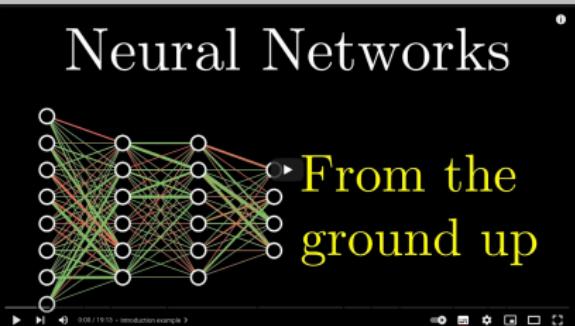
Études pour le projet UCIA

Deux études menées en novembre-décembre 2024:

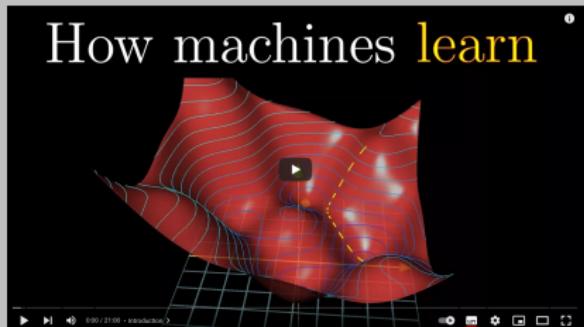
Videos



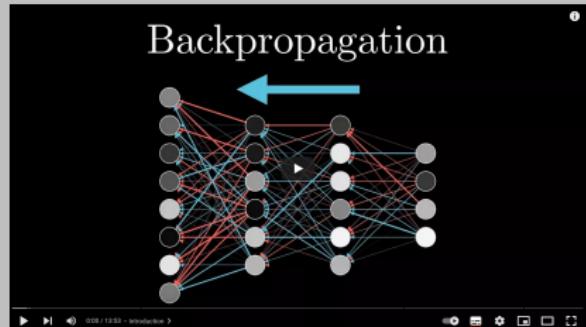
1/ Local: "Le deep learning - YouTube.webm"



2/ local : "But what is a neural network.webm"



3/ Local: "Gradient descent how neural networks learn.webm"



4/ Local: "What is backpropagation really doing .webm"

References

[1] *Artificial Intelligence: A Modern Approach (4th Edition)*, By Stuart Russell & Peter Norvig. Pearson, 2020. ISBN 978-0134610993. aima.cs.berkeley.edu

Intelligence artificielle – Une approche moderne – 4e éd., By Stuart Russell & Peter Norvig. Translated by L. Miclet, F. Popineau, & C. Cadet. Paris: Pearson Education France, 2021. ISBN 978-2326002210.

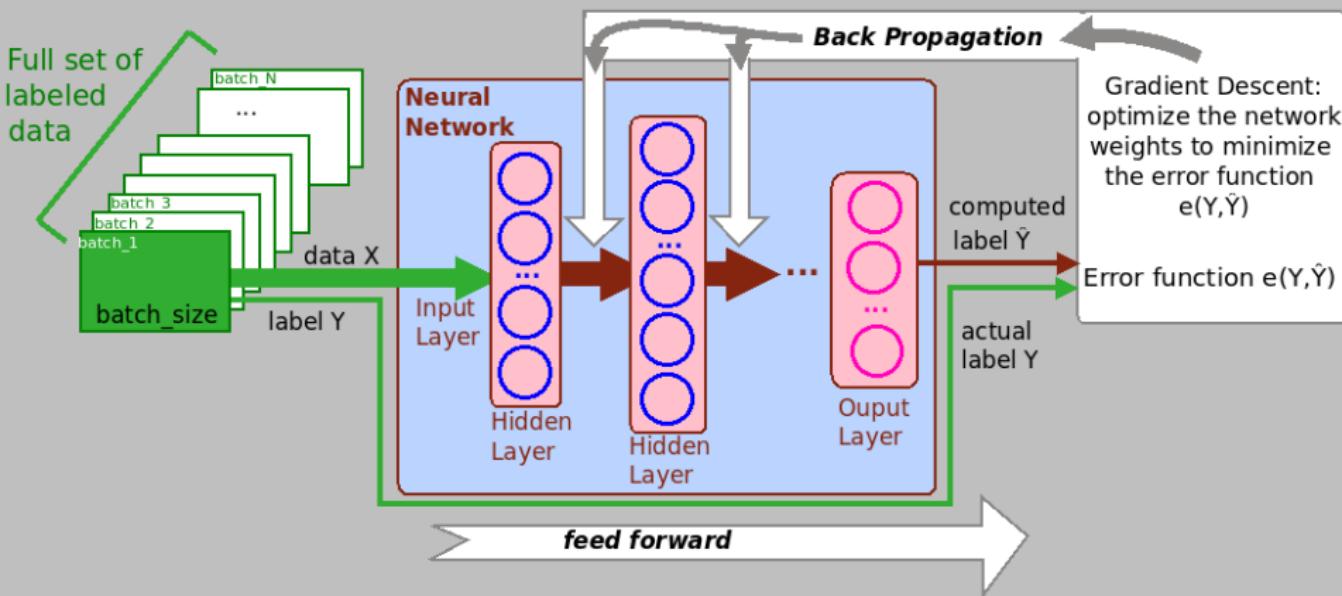
[2] *What is artificial intelligence (AI), and what is the difference between general AI and narrow AI?*, Kris Hammond, 2015
www.computerworld.com/article/2906336/what-is-artificial-intelligence.html

[3] *Stanford Encyclopedia of Philosophy*, plato.stanford.edu/entries/artificial-intelligence

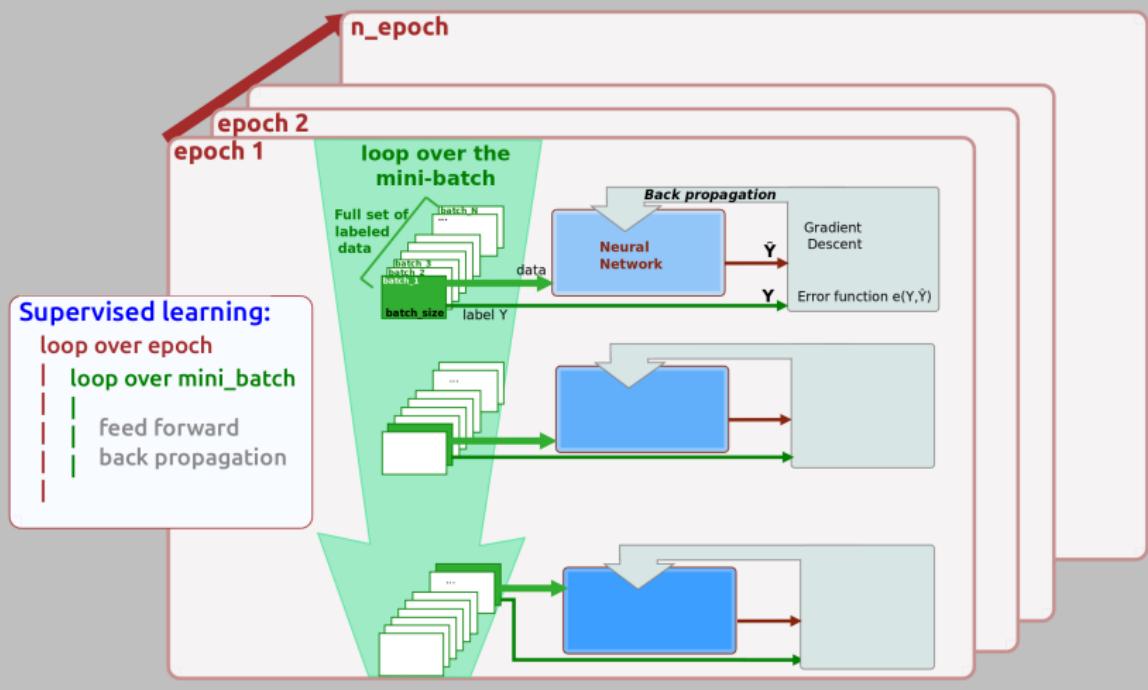
[4] *Deep Learning.*, Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron (2016), MIT Pres, ISBN 9780262035613

Annexes techniques

Supervised learning : Feed Forward and Back Propagation



- The dataset is split into (mini) **batches** of size **batch_size**
- After each *feed forward* the *Back Propagation* algorithm modifies the weights neurons to minimize the error e .



- Training with the whole dataset is repeated n_epoch times,
- The network state at the end of epoch n becomes the initial state for epoch $n+1$.