SÛRETÉ DE FONCTIONNEMENT

« DEEP LEARNING : FORCES, FAIBLESSES & DÉFIS »

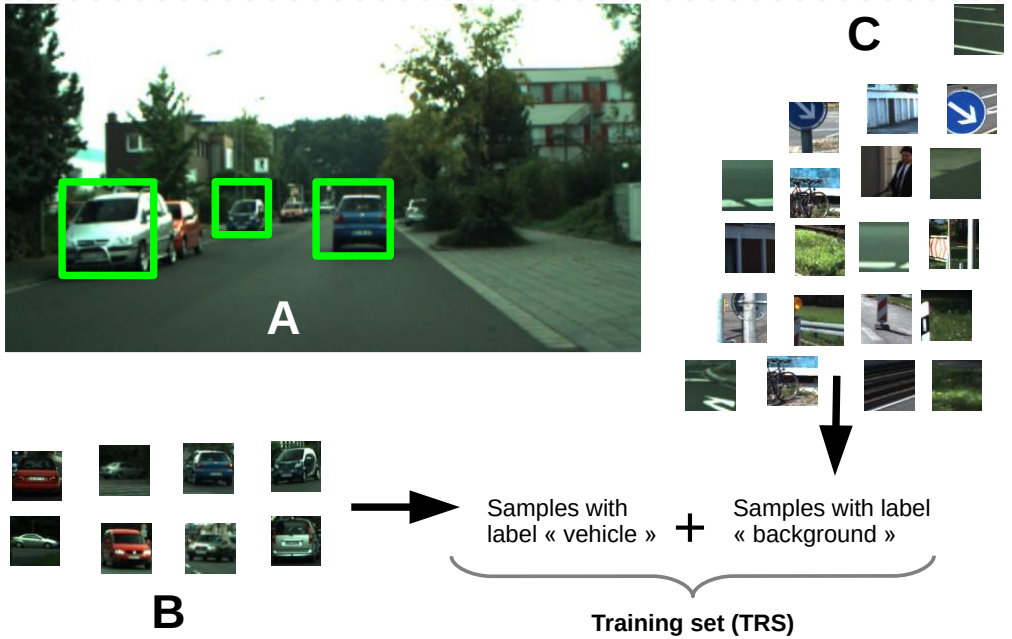
B. Monsuez
ENSTA PT

PRÉSENTATION DE L'APPRENTISSAGE & DU « DEEP LEARNING »

Apprentissage automatique

Apprentissage supervisé

- Construire une fonction Y=f(X) à partir d'exemples
- Ex: Classification d'images



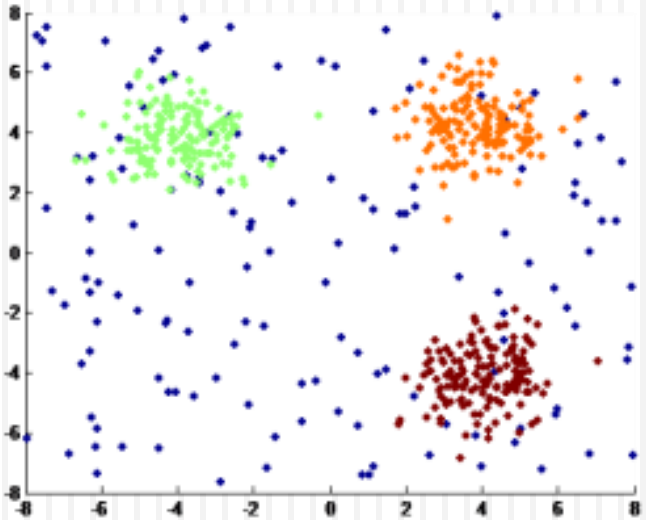
- En général pré-traitement des données -> caractéristiques
- Nombreux algorithmes : SVM, boosting, Réseaux de neurones …

Apprentissage automatique

Autre types d'apprentissage

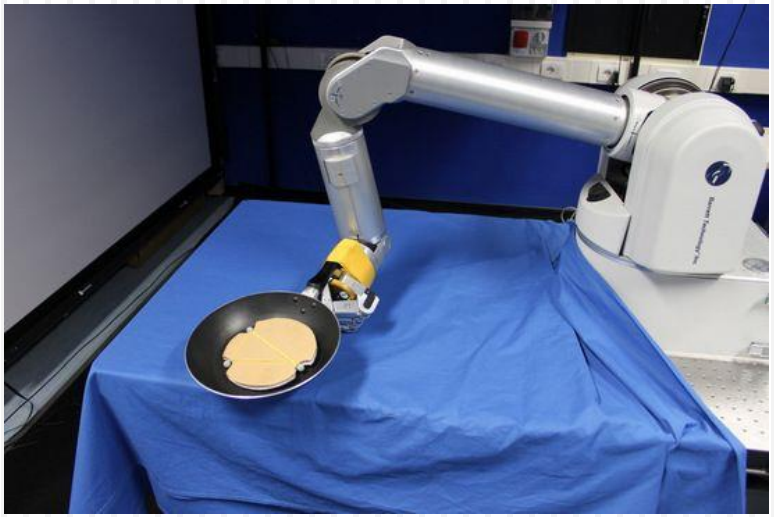
- Non supervisé :

recherche de structure
dans les données



- Par renforcement :

recherche de comportement
optimisant un cout par
essais/erreurs



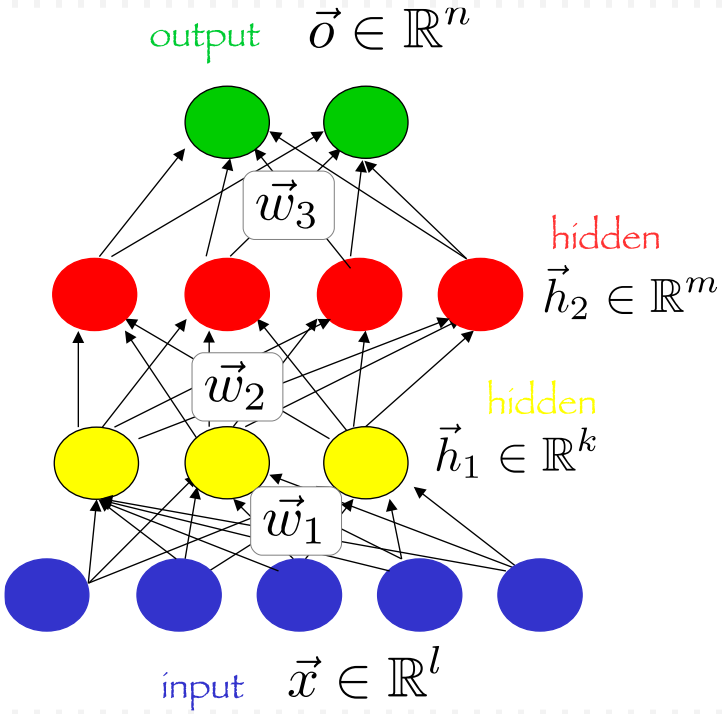
Réseaux de neurones

Famille d'algorithmes (~1960)

- Neurones : unité de calcul élémentaire (somme + non linearité)
- Réseau : neurones connectés par des poids
- Apprentissage : modification des poids
- Méthode : descente de gradient

Différentes structures

- Perceptron multi-couches (Feed forward)
- Réseaux récurrents
- Extrême learning machines
- ….



Deep Learning

Retour des réseaux de neurones (~2006)

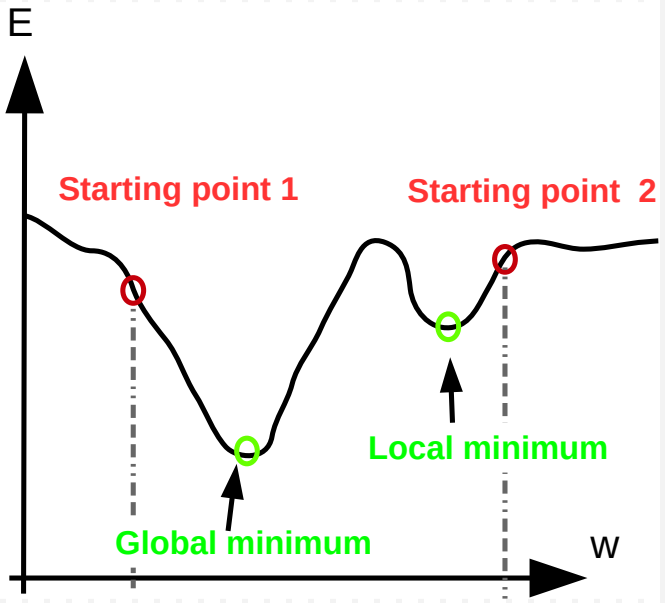
- Perceptron avec « beaucoup » de couches
- Ex: Resnet -> 152 couches
- Base théorique *très* similaire aux perceptrons

Avantages

- Permet des fonctions plus complexes
- Etat de l'art sur de nombreux pb.

Problèmes pré 2006

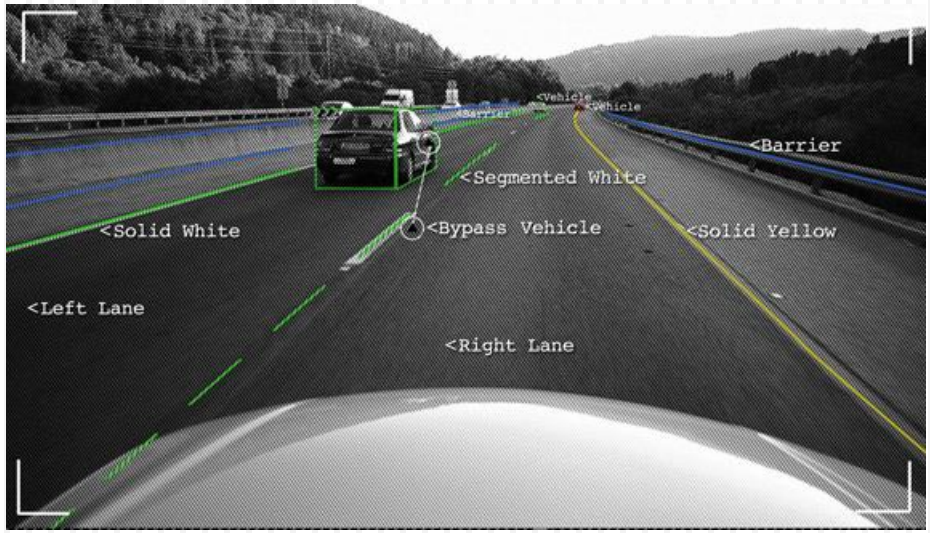
- Puissance de calcul nécessaire
- Apprentissage (min. locaux)
- Données d'apprentissage



Deep Learning : Forces

Excellentes performances applicatives, beaucoup d'applications

- Nombreuses tâches de vision : record en détection, reconnaissance
- Algorithmes de jeux
- Contrôle de robots
- Reconnaissance de la parole
- Traduction automatique
- Description d'images
- …



Deep Learning : Faiblesses

Questions théoriques

- Choix des modèles largement empiriques
- Hyper-paramètres long et complexes à régler

Questions pratiques

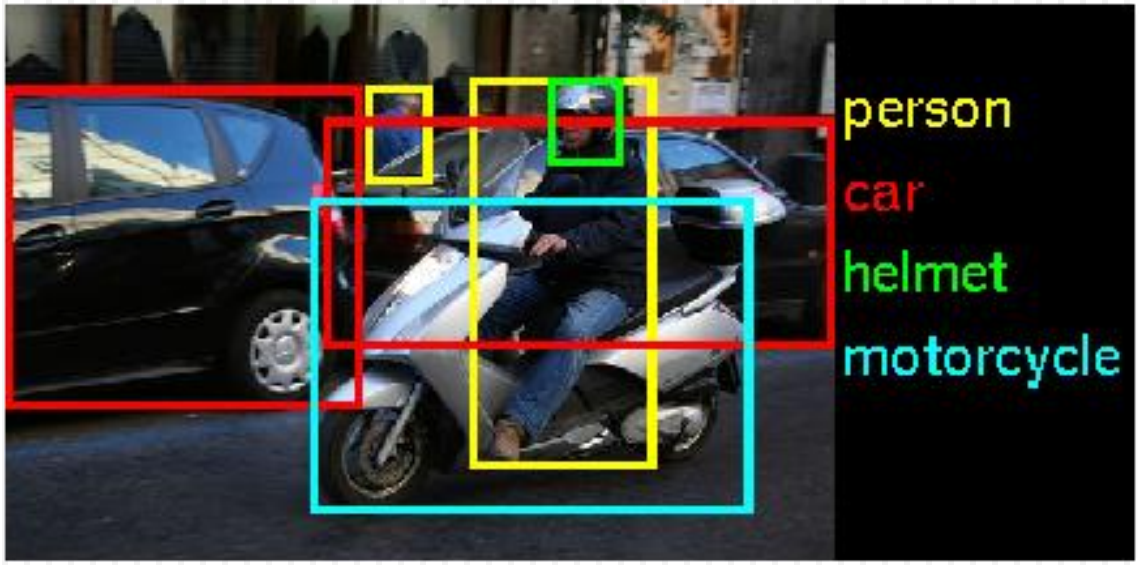
- Pas d'interprétation probabiliste (estimation de la confiance)
- Exemples inquiétants (problème de validation)

Deep Learning : Faiblesses

Questions pratiques

- Besoin de très grandes quantité de données
  - Réseaux entrainés sur ImageNet (14 million d'images)
  - Mobileye emploie 500 personnes pour annoter des données

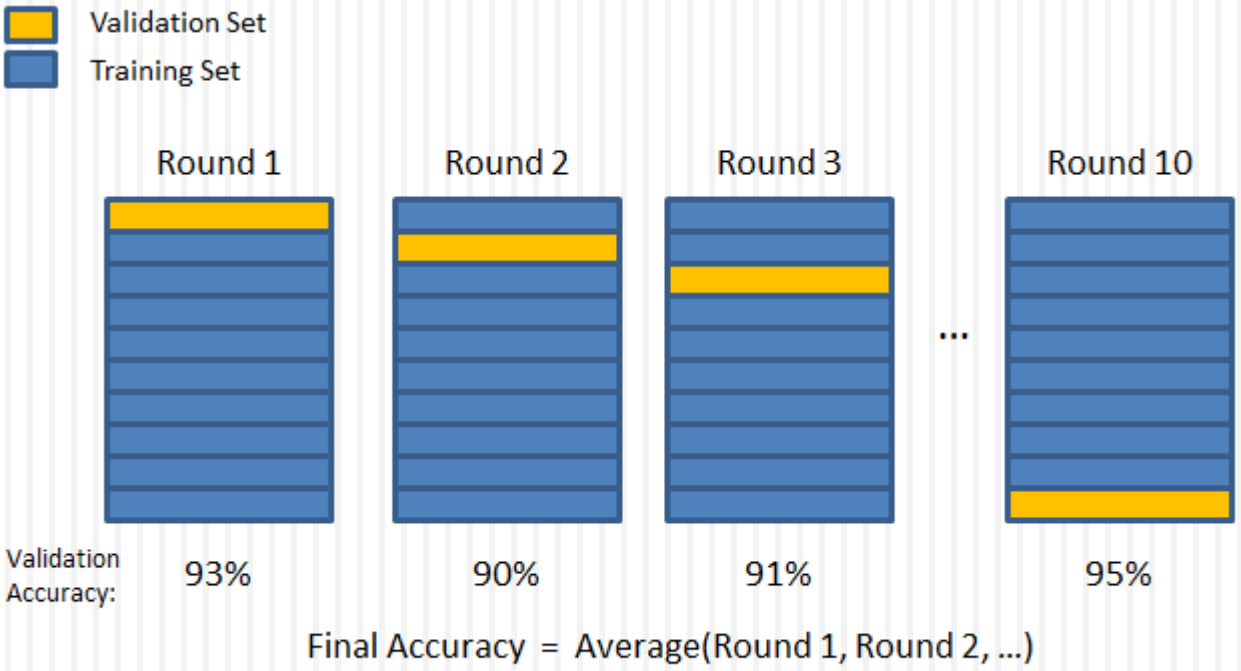
- Puissance de calcul
  - Utilisation extensive de GPU
  - Couteux en embarqué
  - Circuits dédiés



Deep Learning : Applications

Validation des performances

- Pas de possibilité de valider dans l'absolu
- Validation possible sur un jeu de test
- Besoin de beaucoup de données ou cross-validation



Quand ne pas utiliser le deep-learning ?

Problèmes sur des données en petites dimensions

- Une force du deep learning est d'intégrer la détection de caractéristiques pour des données complexes (images, flux audio), ce n'est pas forcement utile
- Exemples :
  - classifications de données statiques simples (télémétrie, gestion de stocks)
  - Problèmes pour lesquels les caractéristiques pertinentes sont connues/simples (détection d'objets connus dans un cadre contraint)
- Alternatives possibles:
  - Séparateurs à Vaste Marge (SVM)
  - Forêt d'arbre aléatoires (Random Forest)
  - Boosting

Quand ne pas utiliser le deep-learning ?

Problèmes pour lesquels l'analyse de la solution est intéressante

- Le deep learning fournit des réseaux de grande taille, souvent difficilement analysables
- La sortie du réseau n'est en général pas interprétable en terme de probabilité
- Pour des problèmes de petites tailles, il peut être utile de comprendre sur quoi se base le résultat et connaître son incertitude
- Alternatives
  - Processus Gaussiens: donnent une solution avec une variance associée
  - Réseaux Bayésien : fournissent un modèle explicite des dépendances entre variables et permettent d'expliquer sur quoi repose le résultat

Quand ne pas utiliser le deep-learning ?

Problème ou la performance n'est pas critique

- Sur certain problèmes, le gain du deep learning peut être réel mais faible au regard de son cout de calcul
- Exemple:
  - Classification de chiffres manuscrits sur la base de données MNIST
  - Meilleurs modèle SVM : 0,56% d'erreur
  - Meilleur modèle Deep Learning : 0,23% d'erreur
  - Gain de 0,33% pour un cout computationnel > 10x
- Alternatives
  - Boosting/cascades
  - Forêt d'arbre aléatoires (Random Forest)

PROBLEMATIQUES D'IMPLANTATIONS ET DE VALIDATIONS

Deep Learning : Implémentation

Principe de mise-en-œuvre

- **Entrainement du système :**

  Détermination des coefficients par exposition à de très grands échantillons de données
  de plusieurs millions à plusieurs milliards de paramètres à ajuster.

- **Déploiement et calcul :**

  Calcul en temps-réel

  Nombre de calculs importants sur des réseaux profonds et des structures complexes.

Possibilité de scinder les plateformes :

Une plateforme d'apprentissage

Une plateforme de reconnaissance

Deep Learning : Implémentation

Spécificité du Deep Neural Network

- Recours à des modèles de réseaux de neurones impliquant une même opération sur un ensemble de neurones (ex: CNN)
- Possibilité de paralléliser par des GPU
- Repose sur des multiplications de matrices
- Ne requiert pas systématiquement une précision importante.

DNN et conception modulaire

- Principe de boîte à outils de différents modèles de réseaux de neurones adaptés à certaines classes ce problèmes.
- Principe de modèle de composition entre les différentes couches et sous-réseaux.

Deep Learning : Implémentation

Utilisation de Hardware Spécifique

- GPU : accélération notable de l'ensemble des opérations matricielles, convolution…
- DSP : accélération des calculs, modèle VLIW.
- FPGA : synthèse de réseau au niveau du composant ou accélérateur de calcul lors de l'apprentissage.
- TPU (Tensor Processor Unit) : ASIC ad hoc pour augmenter le débit des opérations
  - Réduction de la précision de calcul
  - Optimisation du flot d'échange de données
  - Minimisation du contrôle
  - Hybride SIMD et VLIW

Deep Learning : Implémentation

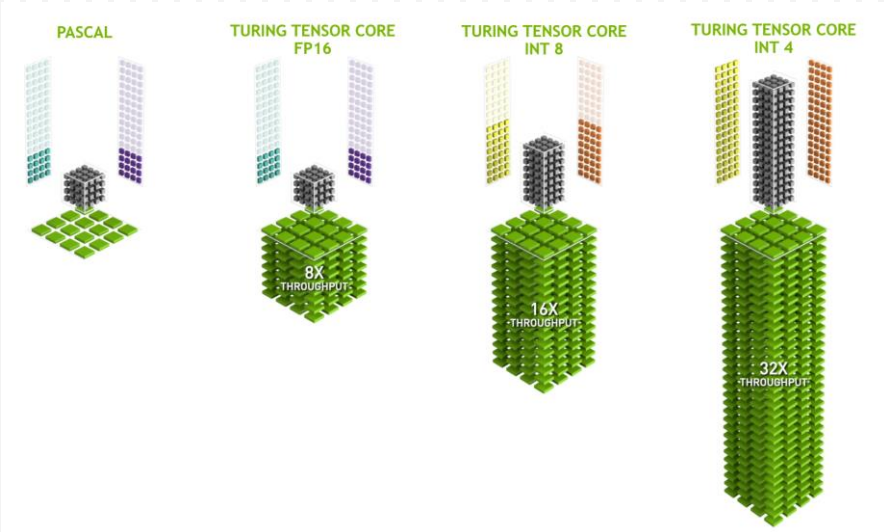
Forces & Faiblesse des briques hardware

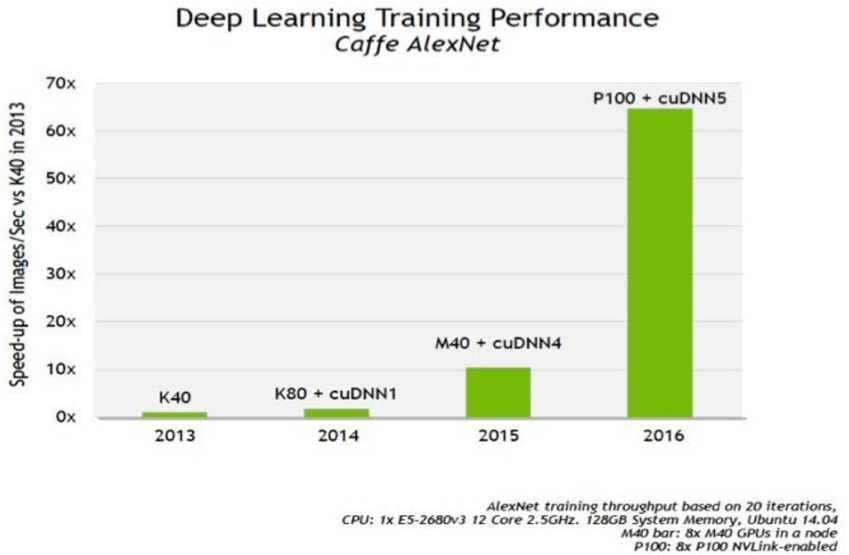
- CPU : Faible ratio Performance/Watt.
- GPU : Ratio Performance/Watt élevé.
  Accélération des phases d'apprentissage et des phases de déploiement.
  Adapté plus à certains types de réseau.
  Speedup : x160 pour 130 W, Tesla K40, DnnWeawer
- FPGA : Ratio Performance/Watt élevé (voir très élevé)
  Configuration différente entre apprentissage et déploiement.
  Possibilité de « synthétise » le réseau sur le composant.
  Plus flexible mais plus complexe à mettre en œuvre.
  Speedup : x45 pour 25 W, Arria 10 DnnWeawer
- TPU : Usage pour l'instant uniquement dans la phase de déploiement. Technologie propriétaire adaptée à un Framework.

Deep Learning : Implémentation

Tendances actuelles (au niveau matériel)

- GPU : évolution des GPU pour prendre en compte les DNN et les DCN.
  Augmentation des débits et du nombre de Teraflops
  Accélération d'un facteur 50
  dans les 3 dernières années !
  Diminution des latences
  et Augmentation du Débit
  Enveloppe énergétique
  reste élevée.





- Démonstrabilité du comportement Complexe

Deep Learning : Implémentation

- Tendances actuelles (au niveau matériel)
  - CPU : Architecture ManyCore.
    Augmentation de l'efficacité énergétique
    Amélioration des Interconnexions
    Difficilement Qualifiable en terme de SdF

  - FPGA : intégration Processeur + FPGA pour SOC
    DSP sur FPGA
    Développement d'outils de synthèses adaptés au DNN et DCN.
    Enveloppe énergétique modérée.
    Problématique de qualification de l'implantation sur FPGA
    Utilisation de Cœurs Démontrables & Temps-réels

Deep Learning : Développement

Emergence de nombreuses solutions d'implantations

Attention 3 phases dans le développement :

Prototype

Apprentissage

Déploiement

Avec des contraintes, des acteurs et des normes différentes.

Deep Learning : Validation

Problèmes :

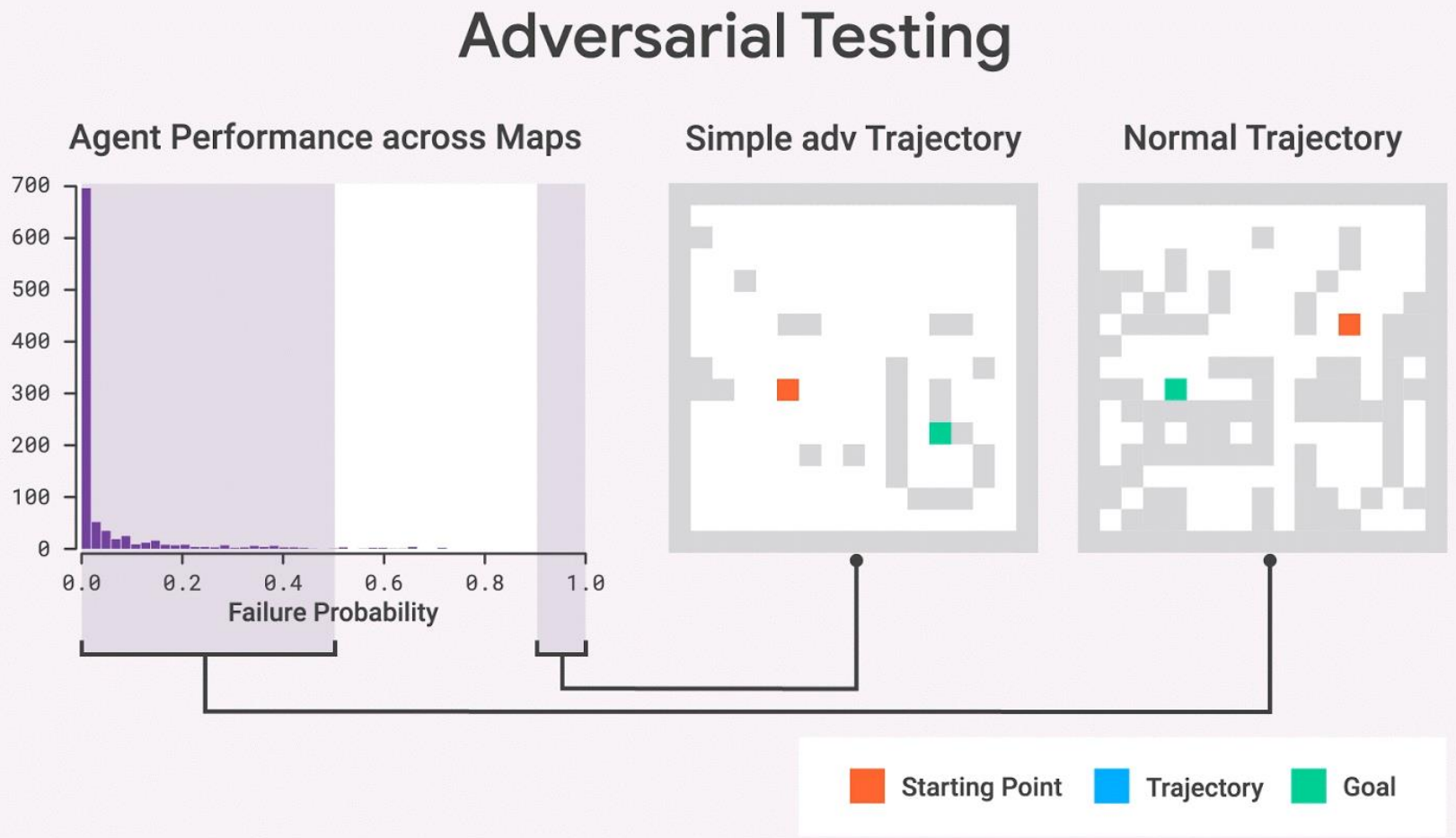
- Aucune garantie sur la qualité du résultat.
- Absence de « sûreté intrinsèque »

Problèmes complémentaires :

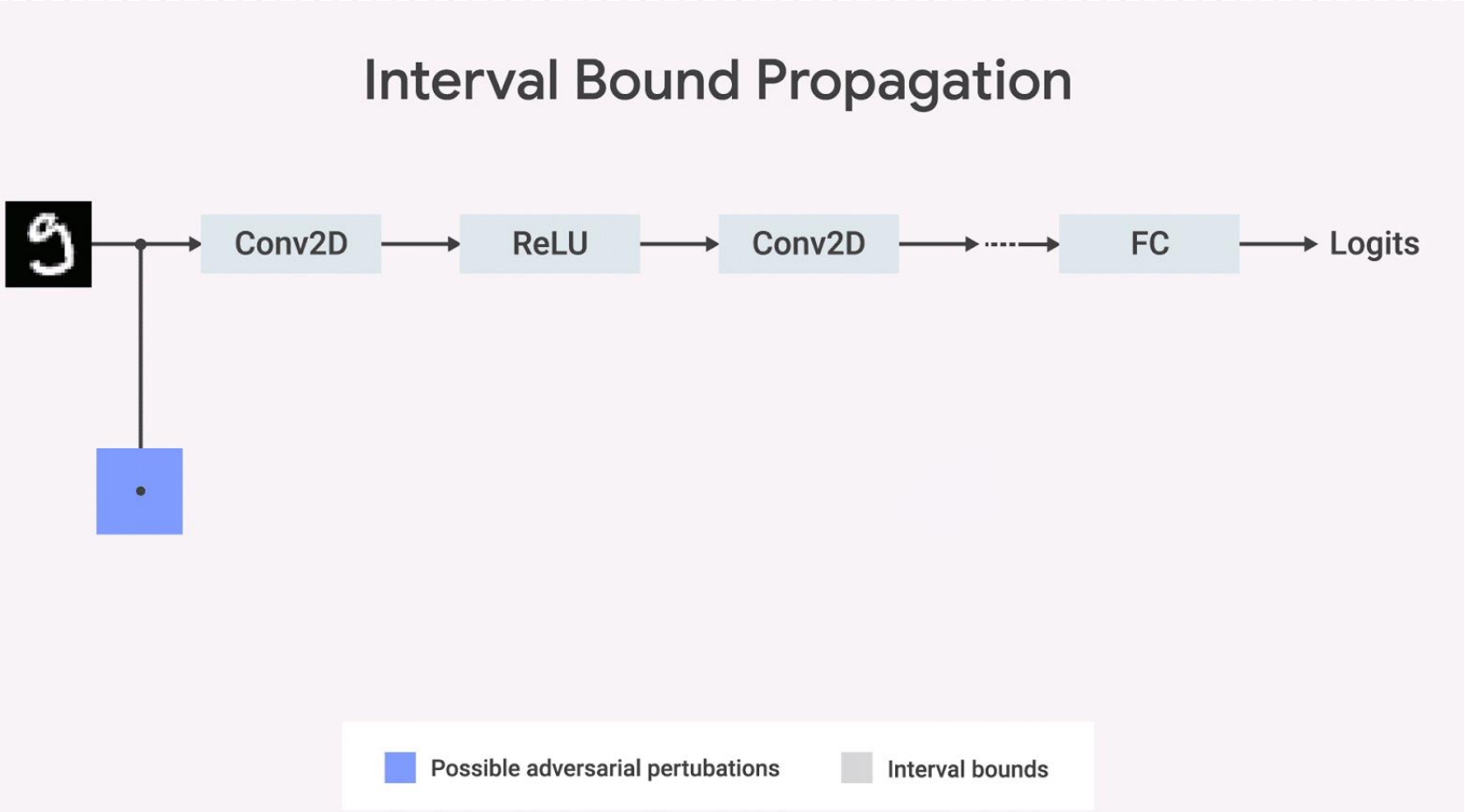
- Apprentissage en ligne : comportement se modifiant dans le temps ? Comment qualifier ?
- Rédaction du code, des bibliothèques (conformité avec Misra C/C++?)

La validation d'un tel algorithme est un problème ouvert à ce jour !

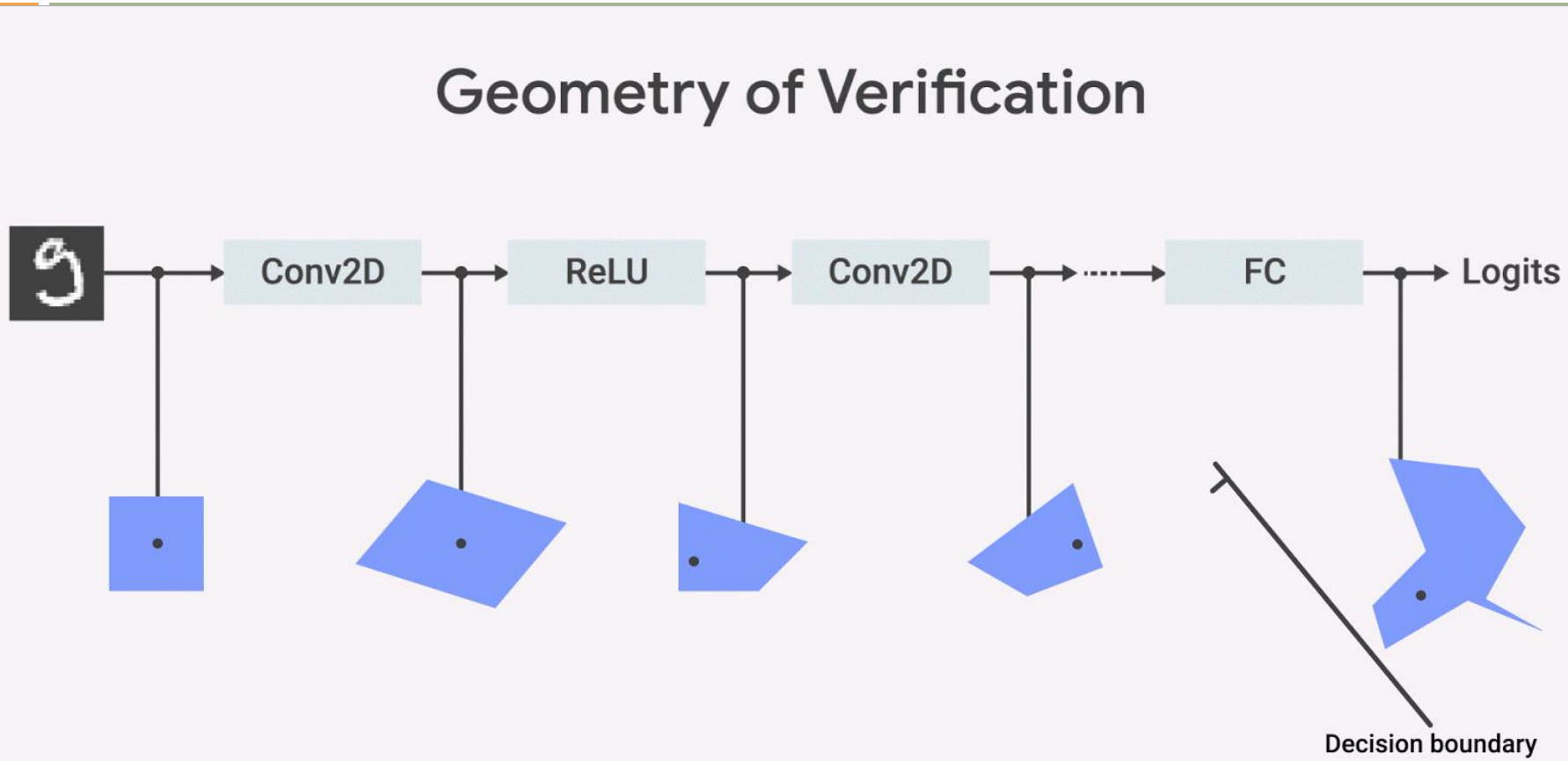
Deep Learning : Validation – Solutions explorées



Deep Learning : Validation – Solutions explorées



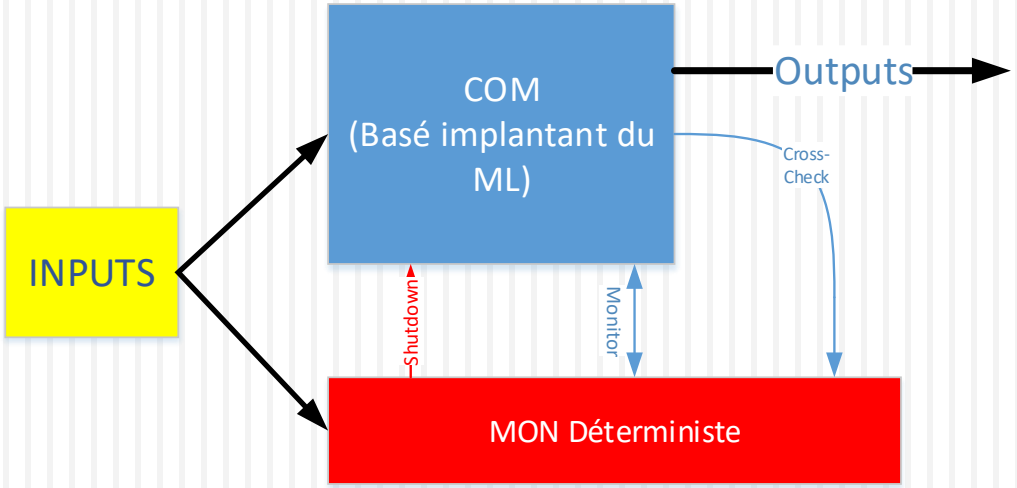
Deep Learning : Validation – Solutions explorées



Deep Learning : Validation – Solutions explorées

Architecture de Systèmes basées sur l'autonomie décisionnelle

- Disposer pour chacune des opérations d'une approche COM-MON



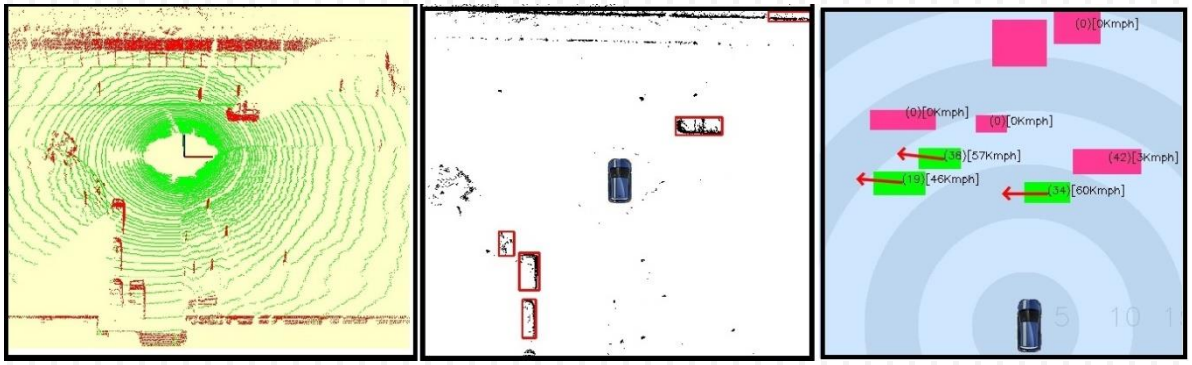
Défis

- Disposer pour les opérations d'un algorithme de vérification de pertinence de la solution.

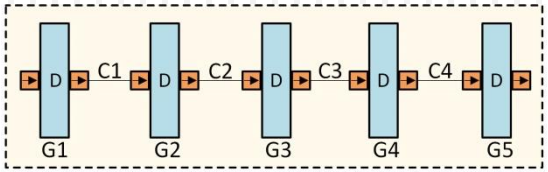
Deep Learning : Validation – Solutions explorées

Nécessité d'une approche Model-Based compositionnelle combinant plusieurs flots de traitement

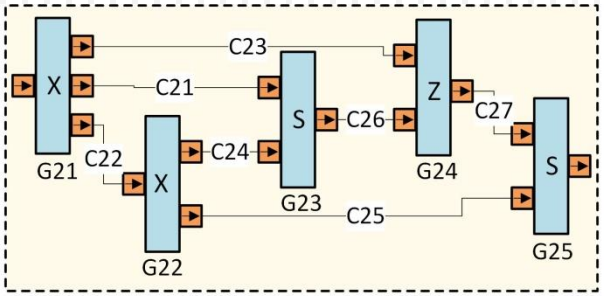
Test Vehicle



Tracking Results



C1:Data Preprocessing, C2: Segmentation, C3: Data Association, C4: State Estimation



C21: 2.5D Projection, C22: Ground Classifier, C24: 2D Projection, C25: Euclidean Clustering, C26:Image Segmentation, C27: Box Mapping

Conclusions

- Le « Machine Learning » ouvre des opportunités nouvelles
  - Efficacité
  - Qualité des solutions
- Le « Machine Learning » pose des problèmes en termes de validation
  - Niveau Logiciel
    - Algorithmes non-déterministes
    - Aucune garantie sur la pertinence des solutions
    - Impossible de qualifier le code
  - Niveau matériel
    - Briques de calcul complexes fortement parallèles
- L'implantation du « Machine Learning » impose de repenser la validation
  - Au niveau Système
  - Au niveau « Exigences de Vérifications »