

# Intelligence Artificielle pour l'embarqué



## Présentation

**Code interne :** EE9IT398

## Description

Responsables :

Guillaume Bourmaud, ENSEIRB-MATMECA

Frédéric Druillolle, CENBG

Cours :

Première partie : IA classique

Introduction à l'apprentissage supervisé.

Approches paramétriques.

Réseaux de neurones.

Perceptron multicouche MLP.

Apprentissage des paramètres d'un réseau de neurones.

Fonctions de coût.

Optimisation des paramètres d'un réseau de neurones par rétropropagation du gradient. Descente de gradient stochastique.

ADAM.

Généralisation / Arrêt prématuré.

Architecture de réseaux de neurones.

Couche de convolution.

BatchNorm.

Connexion résiduelle.

ResNet.

Spécialisation d'un réseau de neurones.

Augmentation de données.

Introduction à PyTorch.

Deuxième partie : IA embarquée

Après avoir décidé le type de réseau de neurones, le nombre de couches et le nombre de neurones par couche, après avoir défini les fonctions d'activation de chaque couche, le nombre d'epochs et le choix de la fonction de perte, vous avez un réseau de neurone qui a appris. Mais comment exploiter son potentiel dans une architecture embarquée ? En effet, le nombre de coefficients et donc le nombre d'opérations matricielles à effectuer est à priori trop volumineux pour être intégrable dans un système tel qu'un FPGA. L'objectif est donc de découvrir les outils nécessaires et les étapes pour essayer de valider un modèle d'IA appris pour FPGA à



partir d'une sauvegarde en format json et hd5. Il faut transformer le modèle à travers des opérations de pruning et de quantification de créer des lookUp tables selon la fonction d'activation utilisée.

Après avoir défini la plateforme et donc la technologie du système embarqué :

GPU.

Multi-purpose multi-processor component.

Circuits neuromorphiques.

System On Chip (Xilinx).

Nous regarderons les frameworks d'optimisation des modèles IA et leur transformation en code VHDL synthétisable à travers des outils disponibles sur le marché :

Vivado HLS et Vitis.

HLS4ML.

Nous exploiterons HLS4ML et Vivado HLS/Vitis à travers d'exemples simple puis dans un exemple de catégorisation de signaux temporels issus d'un capteur. Nous verrons ainsi l'ensemble des étapes en créant un modèle d'IA pour une cible FPGA Zynq. Cela nous permettra d'explorer l'ensemble des fonctions que nous offre ce FPGA type SoC contenant 2 processeurs ARM9 intégrables.

TP :

Perceptron MLP avec numpy.

Introduction à PyTorch. Perceptron MLP avec PyTorch.

Perceptron MLP et CNN avec la base de données MNIST.

Projet sur la spécialisation d'un réseau de neurones.

IA embarquée autour des outils HLS4ML et Vivado HLS/Vitis.

---

## Syllabus

Responsables :

Guillaume Bourmaud, ENSEIRB-MATMECA

Frédéric Druillolle, CENBG

Cours :

Première partie : IA classique

Introduction à l'apprentissage supervisé.

Approches paramétriques.

Réseaux de neurones.

Perceptron multicouche MLP.

Apprentissage des paramètres d'un réseau de neurones.

Fonctions de coût.

Optimisation des paramètres d'un réseau de neurones par rétropropagation du gradient. Descente de gradient stochastique.

ADAM.

Généralisation / Arrêt prématuré.

Architecture de réseaux de neurones.

Couche de convolution.

BatchNorm.

Connexion résiduelle.

ResNet.

Spécialisation d'un réseau de neurones.

Augmentation de données.

Introduction à PyTorch.



Deuxième partie : IA embarquée

Après avoir décidé le type de réseau de neurones, le nombre de couches et le nombre de neurones par couche, après avoir défini les fonctions d'activation de chaque couche, le nombre d'epochs et le choix de la fonction de perte, vous avez un réseau de neurone qui a appris. Mais comment exploiter son potentiel dans une architecture embarquée ? En effet, le nombre de coefficients et donc le nombre d'opérations matricielles à effectuer est à priori trop volumineux pour être intégrable dans un système tel qu'un FPGA. L'objectif est donc de découvrir les outils nécessaires et les étapes pour essayer de valider un modèle d'IA appris pour FPGA à partir d'une sauvegarde en format json et hd5. Il faut transformer le modèle à travers des opérations de pruning et de quantification de créer des lookUp tables selon la fonction d'activation utilisée.

Après avoir défini la plateforme et donc la technologie du système embarqué :

GPU.

Multi-purpose multi-processor component.

Circuits neuromorphiques.

System On Chip (Xilinx).

Nous regarderons les frameworks d'optimisation des modèles IA et leur transformation en code VHDL synthétisable à travers des outils disponibles sur le marché :

Vivado HLS et Vitis.

HLS4ML.

Nous exploiterons HLS4ML et Vivado HLS/Vitis à travers d'exemples simple puis dans un exemple de catégorisation de signaux temporels issus d'un capteur. Nous verrons ainsi l'ensemble des étapes en créant un modèle d'IA pour une cible FPGA Zynq. Cela nous permettra d'explorer l'ensemble des fonctions que nous offre ce FPGA type SoC contenant 2 processeurs ARM9 intégrables.

TP :

Perceptron MLP avec numpy.

Introduction à PyTorch. Perceptron MLP avec PyTorch.

Perceptron MLP et CNN avec la base de données MNIST.

Projet sur la spécialisation d'un réseau de neurones.

IA embarquée autour des outils HLS4ML et Vivado HLS/Vitis.

---

## Modalités de contrôle des connaissances

### Évaluation initiale / Session principale - Épreuves

Type d'évaluation	Nature de l'épreuve	Durée (en minutes)	Nombre d'épreuves	Coefficient de l'épreuve	Note éliminatoire de l'épreuve	Remarques
Contrôle Continu Intégral	Participation Active			1		
Contrôle Continu Intégral	Compte-Rendu			1		



# Infos pratiques

---

## Contacts

Guillaume Bourmaud

✉ [Guillaume.Bourmaud@bordeaux-inp.fr](mailto:Guillaume.Bourmaud@bordeaux-inp.fr)