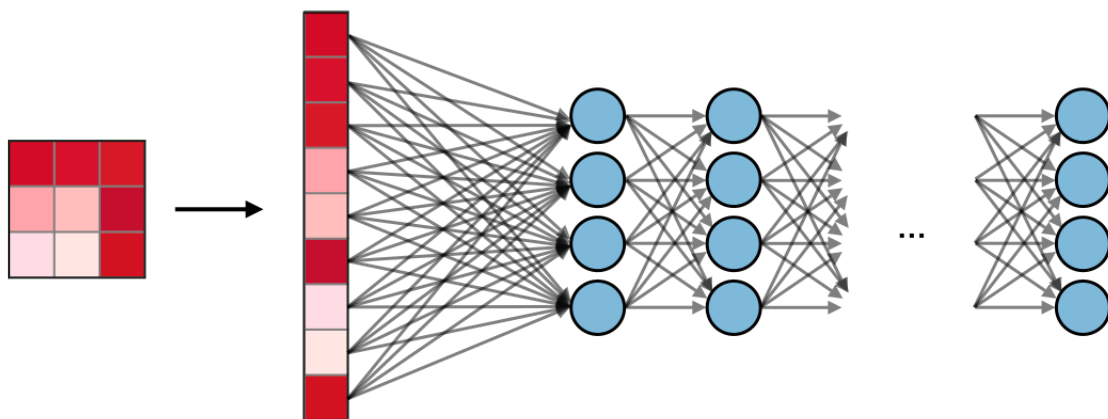


# Compte rendu de TAL n°1 Implémentation C de réseaux de neurones convolutifs sur systèmes embarqués

Ewann DELACRE

24 avril 2025  
Professeur référent : Hai Nam Tran



1ère année de Master Informatique  
Parcours Logiciels pour Systèmes Embarqués  
Promotion 2025  
Université de Bretagne Occidentale, Brest

## Table des matières

<b>1</b>	<b>Objectif du travail</b>	<b>2</b>
1.1	Du 3 au 18 avril . . . . .	2
<b>2</b>	<b>Notions étudiées</b>	<b>2</b>
2.1	Réseaux de neurones . . . . .	2
2.2	Machine Learning . . . . .	2
<b>3</b>	<b>Mise en place de l'environnement</b>	<b>2</b>
<b>4</b>	<b>Implémentations en langage C</b>	<b>2</b>
<b>5</b>	<b>Formation suivie</b>	<b>3</b>
<b>6</b>	<b>Bilan actuel et prochaines étapes</b>	<b>3</b>

# 1 Objectif du travail

## 1.1 Du 3 au 18 avril

Ce projet vise à explorer les bases de l'apprentissage automatique (machine learning), avec un accent particulier sur la régression et la classification, dans le but de construire un réseau de neurones convolutif (CNN) implémenté en langage C et destiné à être exécuté sur un microcontrôleur.

## 2 Notions étudiées

### 2.1 Réseaux de neurones

- Compréhension du neurone formel : couche d'entrée, couches cachées avec fonction d'activation, couche de sortie.
- Étude des fonctions d'activation : Heaviside, ReLU, logique bayésienne.
- Représentation logique et schématique du fonctionnement du neurone.

### 2.2 Machine Learning

Présentation générale du concept d'apprentissage automatique.

Classification des types d'apprentissage :

Apprentissage supervisé :

- Régression : prédire une valeur continue (ex. prix immobilier).
- Classification : prédire une catégorie (ex. détection de maladie).

Apprentissage non supervisé :

- Clustering : regroupement de données similaires (ex. Google News).
- Anomaly detection, réduction de dimensionnalité.

## 3 Mise en place de l'environnement

Installation de Jupyter Notebook et Python3.

Clonage du dépôt GitHub contenant les laboratoires (labs).

Prise en main des notebooks existants pour visualisation et compréhension des modèles.

## 4 Implémentations en langage C

Dans le but d'approcher le fonctionnement interne des modèles de machine learning, plusieurs laboratoires python ont été implémentés en langage C :

- Lab03W1 : prédiction linéaire simple
- Lab04W1 : fonction Cost
- Lab05W1 : fonction calcul erreur quadratique moyenne
- Lab01W2 : fonction calcul produit scalaire et prédiction vectorisée
- Lab02W2 : prédiction linéaire sur plusieurs entrées

- Lab06W2 : SK Learn
- Lab01W3 : prédiction linéaire binaire
- Lab02W3 : fonction sigmoïde
- Lab01W3 : prédiction logistiques

L'ensemble de ces implémentations a été publié sur GitLab :

<https://gitlab-depinfo.univ-brest.fr/e22306771/c-cnn>

## 5 Formation suivie

Cours : "Machine Learning" – Andrew Ng sur Coursera

Avancement par semaine :

Semaine 1 : terminé – régression, coût, descente de gradient

Semaine 2 : terminé – régression multiple, vectorisation

Semaine 3 : terminé – classification

Semaine 4 : en cours – algorithmes avancés, introduction à TensorFlow

## 6 Bilan actuel et prochaines étapes

Tous les concepts de base nécessaires à la prédiction ont été explorés, modélisés, implémentés et testés. Les versions C ont permis de simuler des environnements embarqués et de préparer la transition vers des implémentations plus complexes (réseaux multi-couches, CNN, classification multi-classes).

Les parties suivantes ne sont pas traitées ici et feront l'objet d'un second rapport :

- Utilisation de TensorFlow Lite Micro (TFLM)
- Portage sur système embarqué (Zephyr RTOS)
- Implémentation de CNN simplifié type U-NET