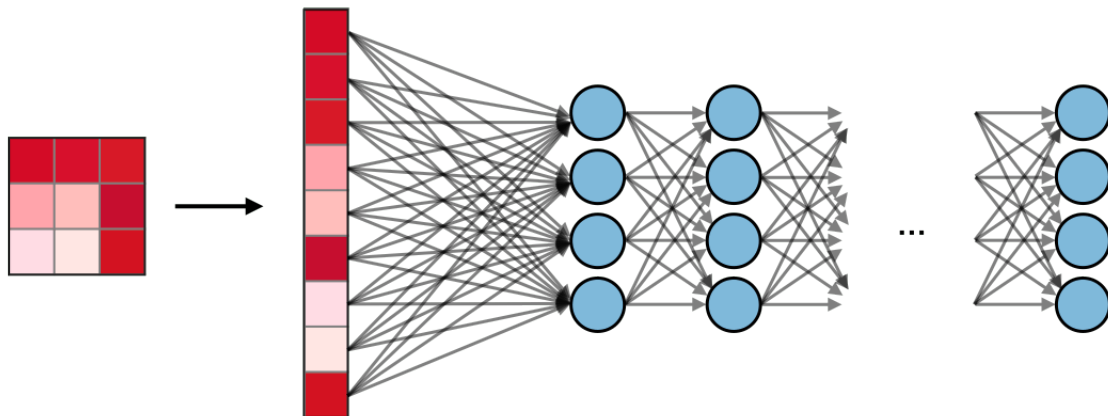


Compte rendu de TAL n°2 Implémentation C de réseaux de neurones convolutifs sur systèmes embarqués

Ewann DELACRE

28 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

1	Objectif du travail	2
1.1	Du 21 au 25 avril	2
2	Notions étudiées	2
2.1	TensorFlow Lite Micro	2
2.2	Zephyr RTOS	2
2.3	Architecture UNET	2
3	Mise en place de l'environnement	2
4	Implémentations en langage C	2
5	Formation suivie	3
6	Bilan actuel et prochaines étapes	3

1 Objectif du travail

1.1 Du 21 au 25 avril

Poursuivre l'évolution du projet en s'orientant vers l'utilisation de TensorFlow Lite Micro pour rendre les modèles de prédiction compatibles avec les contraintes des systèmes embarqués (notamment Raspberry Pi Pico WH), en préparation du futur portage sur Zephyr OS.

2 Notions étudiées

2.1 TensorFlow Lite Micro

- Présentation de TensorFlow Lite Micro (TFLM) : version de TensorFlow dédiée aux microcontrôleurs.
- Concepts de modèle .tflite, de conversion d'un modèle standard vers un modèle optimisé.
- Découverte de la structure MicroInterpreter, MicroMutableOpResolver pour alléger les opérations utilisées.

2.2 Zephyr RTOS

Introduction au Zephyr Real-Time Operating System pour microcontrôleurs :

- Architecture modulaire
- Gestion des threads, timers, périphériques
- Ciblage Raspberry Pi Pico WH avec environnement ARM Cortex-M0+

Spécificités d'un environnement Zephyr pour le déploiement de réseaux de neurones.

2.3 Architecture UNET

- Découverte du modèle de réseau de neurones U-Net appliqué à la segmentation d'images.
- Introduction à l'idée de déploiement de versions allégées de U-Net sur microcontrôleurs pour tâches de classification ou détection.

3 Mise en place de l'environnement

Installation locale de TensorFlow Lite Micro sans dépendance Python lourde.

Construction semi-manuelle de l'arborescence tflm_build/ à partir des sources officielles pour support de la compilation C++ embarquée.

Configuration de Makefile compatible pour simulation locale sur PC.

4 Implémentations en langage C

- Écriture d'un module C++ (pas le choix avec l'implémentation de TF Lite Micro) capable de charger un modèle .tflite intégré en binaire.
- Remplacement de bibliothèques (de AllOpsResolver à MicroMutableOpResolver) pour une

optimisation embarquée et par souci de dépendances.

- Rédaction d'un fichier de test et d'appel de code C++ en langage C "main.c" pour simplifier le test sur PC avant le portage sous Zephyr RTOS
- Simulation en environnement local en cours, problèmes de dépendances et de récupération de bibliothèques par chemin absolu à modifier.

L'ensemble de ces implémentations est publié sur GitLab :

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

5 Formation suivie

- Documentation TensorFlow Lite Micro API C
- Documentation officielle Zephyr OS (Developer Guide)
- Tutoriel officiel TensorFlow pour l'implémentation sur microcontrôleurs.
- Avancement : maîtrise de la chaîne complète depuis un modèle jusqu'à son exécution sur MCU.

6 Bilan actuel et prochaines étapes

Le projet est dans une étape de transition entre le développement sur environnement local (PC, Linux simple) avant d'être porté sur système embarqué (mais toujours en phase de développement via le debugger Raspberry Pi)

Prochaines étapes :

- Génération automatique de fichiers .pb puis .tflite depuis modèle d'apprentissage (à confirmer).
- Intégration réelle sous Zephyr RTOS avec optimisation mémoire.
- Première démonstration fonctionnelle de classification multi-classes sur Raspberry Pi Pico WH.