Introducción a TinyML con Keras y TensorFlow Lite

Robótica

Alberto Díaz Álvarez y Guillermo Iglesias Hernández

Departamento de Sistemas Informáticos - Universidad Politécnica de Madrid

8 de octubre de 2023

Introducción a TinyML

¿Qué es esto de TinyML?

El término se refiere a la implementación de modelos de aprendizaje automático (ML) en dispositivos de recursos limitados

- Vamos, en microcontroladores y otros dispositivos de hardware de baja potencia y poca memoria

El objetivo principal es llevar la inteligencia artificial *al borde*

- Es decir, a los dispositivos que interactúan con el mundo real, como robots, drones, wearables, etc.

TensorFlow Lite, Keras, y otros frameworks proporcionan las herramientas necesarias para desarrollar y desplegar modelos de ML en microcontroladores[1][2][3]

1 https://www.sciencedirect.com/science/article/pii/S1319157821003335
2 https://analyticsindiamag.com/how-tensorflow-lite-fits-in-the-tinyml-ecosystem/
3 https://www.tensorflow.org/lite/microcontrollers?hl=es-419

Importancia en el *edge computing*

Los modelos de ML en el *borde* tienen varias ventajas, entre las que detacan:

- **Latencia**: Pueden procesar datos y tomar decisiones en tiempo real sin depender de una conexión a un servidor centralizado
- **Privacidad**: Al procesar los datos localmente en el dispositivo, pueden preservar la privacidad del usuario ya que los datos sensibles no necesitan ser transmitidos
- **Conectividad**: Pueden operar de manera efectiva incluso en entornos con conectividad limitada o nula, algo crítico para muchas aplicaciones en tiempo real
- **Consumo de Energía**: Los microcontroladores son conocidos por su eficiencia energética, lo que permite implementaciones de ML de baja potencia, extendiendo la vida útil de la batería en dispositivos portátiles
- **Costo**: La computación en el borde puede reducir los costos asociados con la transmisión de datos y el procesamiento en la nube

TensorFlow Lite (TFLite)

¿Qué es TensorFlow Lite?

Solución ligera de TensorFlow diseñada para dispositivos móviles y embebidos

- Permite la inferencia en local[4], lo que mejora latencia, privacidad y eficiencia

Principales características:

- **Eficiencia de memoria**: Diseñado para usar solo algunos KiB de memoria
- **Independencia del sistema operativo**: No requiere compatibilidad con el sistema operativo, ni bibliotecas C o C++ estándar, ni asignación de memoria dinámica
- **Soporte para diversos microcontroladores**: P.ej. Arduino, ESP32, y otros basados en la arquitectura ARM Cortex-M
- **Herramientas** para la optimización del tamaño y rendimiento del modelo

Está **diseñado específicamente para ejecutar modelos de ML en microcontroladores** y otros dispositivos de recursos limitados

[4] Computación en el *borde* (*edge computing*).

Flujo de trabajo con Keras y TensorFlow Lite

Kera es un API de alto nivel para redes neuronales escrita en Python

- Se ejecuta sobre TensorFlow (y soporta otros backends como Theano)

El flujo de trabajo para crear un modelo de ML con Keras y TensorFlow Lite es el siguiente:

1. **Entrenar** el modelo con Keras,
2. **Convertir** el modelo a TensorFlow Lite,
3. **Optimizar** el modelo para su uso en microcontroladores,
4. **Desplegar** el modelo en el microcontrolador[5].

Keras para el desarrollo y TensorFlow Lite para la implementación facilitan el desarrollo de aplicaciones de TinyML

5 https://stackoverflow.com/questions/68132431/load-tensorflow-lite-models-in-python

¡GRACIAS!