

实验 4：手势识别——边缘机器学习入门

简介与目标:在本实验中，您将使用开发板内置的陀螺仪收集不同手势的运动数据，通过 TensorFlow Lite 利用这些数据训练神经网络（NN），然后运行神经网络的推理来识别您的手势。您应该通过这个简单的实验获得有关嵌入式机器学习的知识。

1 简介

在这个实验室里，您将被引导构建一个嵌入式机器学习（ML）系统。嵌入式机器学习系统将机器学习集成到资源有限的小型设备中，例如物联网设备、可穿戴设备、机器人和边缘计算设备。其目标是让这些设备能够本地做出智能决策和预测，而无需持续的云连接。这些系统使用针对低功耗处理器和有限内存进行优化的轻量级高效算法。它们能够在本地执行诸如图像识别、语音识别、异常检测和预测性维护等任务，从而确保更快的响应速度、更好的隐私保护以及降低带宽需求。我们构建的嵌入式 ML 系统将用于手势识别。

2 寻找解决方案

基于本课程前三个实验，我们熟悉了矽递科技（Seeed Studio）开发的 XIAO 开发板系列以及用于机器学习的 TensorFlow 平台。手势识别需要惯性测量单元（IMU）来捕捉手部动作。我们可以选择内置 6 轴 IMU LSM6DS3TR-C 的 XIAO nRF52840 Sense 开发板。此外，嵌入式机器学习需要针对低功耗和有限内存的微处理器进行算法优化。这一要求应通过使用 TensorFlow Lite for microcontroller 来满足，它能在仅消耗几 KB 内存的微控制器上运行由 TensorFlow 训练的神经网络模型。因此，嵌入式机器学习的硬件和软件选择为 XIAO nRF52840 Sense 和 TensorFlow Lite。

3 准备并测试开发板

在构建嵌入式机器学习系统之前，我们应先学习并测试新的开发板。矽递科技（Seeed Studio）编写的维基页面（https://wiki.seeedstudio.com/XIAO_BLE/）包含了有关 XIAO nRF52840 Sense 开发板的信息。请阅读该页面的“规格对比”部分，并在答题纸的表 3.1 中填写 XIAO nRF52840 的相应信息。

3.1 XIAO nRF52840 Sense 信息表

阅读“硬件概述”部分。在答题纸上指出表 3.2 中对应功能的引脚编号。

3.2 XIAO nRF52840 Sense 引脚配置表。

请阅读“两个 Arduino 库”部分，了解 Seeed nRF52 Boards 和 Seeed nRF52 mbed-enabled Boards 之间的区别。在实验室中，我们应使用 Seeed nRF52 mbed-enabled Boards 来进行嵌入式机器学习应用。请阅读“入门”部分，以安装库并运行闪烁示例。请注意，对于 mbed-enabled 库，您应选择“XIAO nRF52840 Sense（无更新）”这块板子。如果在查找板子和端口时遇到问题，请按复位按钮两次以启动板子。将闪烁 LED 的亮灭时间分别改为 0.2 秒和 0.2 秒。

3.3 拍摄闪烁电路板的视频,并以“lab4_y33”为文件名上传至 Moodle 平台。

XIAO nRF52840 Sense 板载有一个可由用户编程的三合一 LED。请阅读“玩转内置三合一 LED”部分，并编写代码使该 LED 以“红-蓝-绿-红”的循环方式运行，每种颜色持续 0.2 秒。

3.4 拍摄彩色闪烁板的视频,并以“lab4_y34”为文件名上传至 Moodle 平台。

4 惯性测量单元（IMU）运动监测

在训练神经网络之前，我们应先使用 IMU 捕获手部动作。参考 Wiki 页面 (<https://wiki.seeedstudio.com/XIAO-BLE-Sense-TFLite-Getting-Started/>)，您可以构建一个系统来捕获并识别挥拳或弯曲动作。按照“软件设置”部分，您需要安装用于动作捕捉的所需库和示例。将“IMU_Capture”代码上传至 XIAO nRF52840 Sense，该 nRF52840 开发板在上电或复位时会通过串口打印“aX,aY,aZ,gX,gY,gZ”。之后，板载的 IMU（LSM6DS3，数据手册 https://files.seeedstudio.com/wiki/Grove-6-Axis_AccelerometerAndGyroscope/res/LSM6DS3TR.pdf）将不断捕获开发板的加速度。当开发板的加速度超过阈值时，开发板会将以下动作数据打印到串口。这些数据可以通过 Arduino IDE 的“串口监视器”显示，或者绘制到 Arduino IDE 的“串口绘图仪”中。上电后晃动开发板，将前 5 行动作数据复制到答题纸上。

4.1 运动数据的前 5 行,包括“aX,aY,...”等信息。

列出每一列的单位。您可能在 LSM6DS3 数据表中找到它。

4.2 列出每个数据列的单位。

打开“串行绘图仪”，晃动开发板。截取数据图并将其粘贴到答题纸上。

4.3 摆动数据的图表。

您应该会发现这些运动数据不容易识别。我们应该训练一个神经网络来进行手势识别。

5 构建训练数据集

Arduino IDE 2.X.X 版本的串行监视器存在问题。您应下载 1.8.19 版本 (<https://www.arduino.cc/en/software>)，并使用该版本的串行监视器进行复制操作。

运动数据集。请勿使用 Arduino IDE 1.8.19 版本上传代码。仅将其用于监测。

阅读维基页面中的“训练数据”部分。按照该部分的步骤逐步操作，创建两个手势的数据集。您可以自行设计手势，但不要更改其名称。为您的两个手势拍摄视频。

5.1 录制第一个手势的视频;将其以文件名“lab4_v51”上传至 Moodle 平台。

5.2 录制第二个手势的视频;将其以“lab4_v52”为文件名上传至 Moodle 平台。

5.3 将“punch.csv”和“flex.csv”这两个文件上传至 Moodle 平台。

6 训练神经网络模型并生成精简模型

请阅读 Wiki 页面中“生成 TensorFlow Lite 模型文件”部分。按照该部分的步骤依次导入数据集、定义并训练神经网络，然后将神经网络转换为适用于微控制器处理的 TFLite 格式。在 Colab 笔记本中，您应首先导入在前面部分创建的 csv 文件。在笔记本的“绘制数据”部分，绘制了十组拳击数据中加速度 (a_x, a_y, a_z) 以及角速度 ($\omega_x, \omega_y, \omega_z$) 的三个分量。将这些图形粘贴到答题纸上。

6.1 冲头装置的加速度和角速度曲线图。

然后定义并训练神经网络。在答题纸上列出训练数据集和训练好的神经网络的一些参数。

6.2 培训参数列表

使用训练好的模型预测测试数据，将输出结果粘贴到答题纸上，并计算准确率。

6.3 测试数据的预测与准确性。

最后，Colab 笔记本将创建一个名为“model.h”的 C 头文件，其中存储了神经网络。请分别写出以 Lite 格式和 C 头文件格式存储的模型文件的大小（以字节为单位）。

7 微控制器的推断

返回维基页面，阅读并遵循“推理”部分的内容。请注意，您应使用 Arduino IDE 2.X.X 版本进行上传。演示您的神经网络模型如何识别您的出拳和弯曲动作，并录制演示视频。

7.1 为手势识别录制一段视频;将其以“lab4_v71”为文件名上传至 Moodle 平台。

8 手势切换彩色灯

作为练习，设计三种不同的手势，并训练一个神经网络模型来识别这些手势。每个手势都与三种颜色之一相关联，即红色、蓝色和绿色。修改上一个中的 Arduino 代码

在该部分，通过手势将三合一 LED 切换到相应的颜色。

8.1 为手势变色灯拍摄一段演示视频;将其以“lab4_y81”为文件名上传至 Moodle 平台。

9 结论

总之，我们学习了如何通过传感器收集数据、训练神经网络模型以及在微控制器上运行模型。作为演示，我们构建了一个手势控制灯，内置的 LED 可以通过不同的手势进行切换。您可以使用 TensorFlow Lite 和开发板来进行嵌入式机器学习应用。