

图形化编程之嵌入式机器学习入门

用 Wio Terminal 玩转 TinyML



目录

前言

第 1 课 使用 Wio Terminal 和 Codecraft 学习 TinyML 1

第 2 课 使用内置加速度计进行运动识别 12

第 3 课 使用内置光线传感器进行手势识别 32

第 4 课 使用内置麦克风识别唤醒词 48

第 5 课 使用 Grove 多通道气体传感器进行气味识别 66

更多课程即将推出 80

后记 81

前言

课程名称：图形化编程之嵌入式机器学习入门——用 Wio Terminal 玩转 TinyML

课程介绍

在这个课程中，学生将会了解并使用以 Cortex-M 为核心的微控制器设备——Wio Terminal，运用 Codecraft 的图形编程工具可在其中编程进而训练和部署深度神经网络模型，最终达到学习掌握嵌入式机器学习（TinyML，机器学习的一个分支）知识的目的。此课程包含了七个详细的逐步递进的项目，可以让学生轻松了解并掌握如今高速发展的机器学习的基本概念，以及如何将其运用于低功耗且尺寸更小的微控制器并用来创建智能化、彼此相互连接的系统。

课程大纲

课程说明

此课程意在让学生了解并使用 Codecraft 的图形编程工具在 Wio Terminal 上编程，进而进行训练和部署深度神经网络模型。它包含七个详细的逐步递进的课程项目，让学生能掌握嵌入式机器学习的基本概念，并且懂得如何将其用于低功耗而尺寸更小的微控制器以创建智能化、相互连接的系统。完成本课程后，学生将能够在以 Cortex-M 为核心的微控制器上设计并实践他们自己的嵌入式机器学习项目，从定义问题到收集数据进而训练神经网络模型，再到最后将其部署到设备上以显示推理结果，或者根据推理数据控制其他硬件设备，这些都可以自己完成。

Codecraft 的使用，使得数据收集、模型训练的流程得到了简化。

本课程并不需要编程或电子相关的知识，它将带您从零开始逐步掌握必要的知识，并在每个项目中快速得到实践。



嵌入式机器学习项目的一般步骤如下：

1. 项目概述：介绍课程中要完成的项目目标及需要得到的结果。
2. 背景知识：介绍所要用到的新硬件及其相关的知识。
3. 练习与实践：
 - a. 创建与选择模型
 - b. 数据采集
 - c. 训练与部署

- d. 使用与编程
4. 项目有关的嵌入式机器学习理论知识

具体内容包括：

1. 使用 Wio Terminal 和 Codecraft 学习 TinyML
2. 使用内置加速度计进行运动识别
3. 使用内置光线传感器进行手势识别
4. 使用内置麦克风识别唤醒词
5. 使用 Grove–Multichannel 气体传感器进行气味识别
6. 使用内置加速度计进行运动识别
7. 使用内置光传感器识别条码
8. 使用热成像传感器进行人脸识别
9. 创意项目和总结

嵌入式机器学习知识：

1. 了解输入：输入标签、数据集
2. 了解输出：输出结果、训练表现
3. 了解不同的模型比例
4. 了解超参数
5. 对模型进行评估
6. 学习提高模型训练效果
7. 神经网络的进阶知识：层（也叫 layer）及其知识

课程要求：

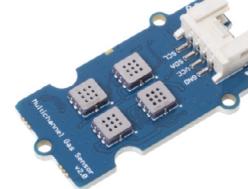
硬件需求（板卡、模块等）：



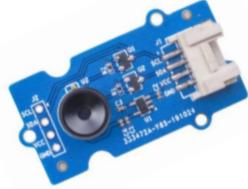
• Wio Terminal x1



• Grove cables x4



• Grove – Multichannel Gas Sensor v2



• Grove – Thermal Imaging Camera

软件需求：图形化编程软件 Codecraft

 **Codecraft**

<https://ide.tinkergen.com/>

注：Codecraft 机器学习的服务器支持，由合作伙伴 [Edge Impulse](#) 提供。

 **EDGE IMPULSE**

课程大纲

No.	名称	概述	硬件			
	入门					
1	使用 Wio Terminal 和 Codecraft 学习 TinyML	<ul style="list-style-type: none"> • TinyML 的基础理论知识 • Wio Terminal 和 Grove 的介绍 • 开始使用 Codecraft 	Wio Terminal	进阶项目 (即将提供)		
	初级项目			6	使用内置加速度计进行运动识别	<ul style="list-style-type: none"> • 背景知识: 运动识别的原理介绍 • 练习与实践: <ul style="list-style-type: none"> ◦ 创建与选择模型 ◦ 数据采集 ◦ 训练与部署 ◦ 使用与编程 • 机器学习理论 (模型评估)
2	使用内置加速度计进行运动识别	<ul style="list-style-type: none"> • 背景知识: 加速度计的原理介绍 • 练习与实践: <ul style="list-style-type: none"> ◦ 创建与选择模型 ◦ 数据采集 ◦ 训练与部署 ◦ 使用与编程 • 机器学习理论 (输入标签, 数据集) 	Wio Terminal	7	使用内置光线传感器识别条码	<ul style="list-style-type: none"> • 背景知识: 条码的原理介绍 • 练习与实践: <ul style="list-style-type: none"> ◦ 创建与选择模型 ◦ 数据采集 ◦ 训练与部署 ◦ 使用与编程 • 机器学习理论 (提高模型训练效果)
3	使用内置光线传感器进行手势识别	<ul style="list-style-type: none"> • 背景知识: 光线传感器的原理介绍 • 练习与实践: <ul style="list-style-type: none"> ◦ 创建与选择模型 ◦ 数据采集 ◦ 训练与部署 ◦ 使用与编程 • 机器学习理论 (输出, 训练表现) 	Wio Terminal	8	使用热成像传感器进行人脸识别	<ul style="list-style-type: none"> • 背景知识: 热成像传感器原理介绍及人脸识别 • 练习与实践: <ul style="list-style-type: none"> ◦ 创建与选择模型 ◦ 数据采集 ◦ 训练与部署 ◦ 使用与编程 • 机器学习理论 (层 - layer)
4	使用内置麦克风识别唤醒词	<ul style="list-style-type: none"> • 背景知识: 麦克风的原理介绍 • 练习与实践: <ul style="list-style-type: none"> ◦ 创建与选择模型 ◦ 数据采集 ◦ 训练与部署 ◦ 使用与编程 • 机器学习理论 (不同的模型比例) 	Wio Terminal	总结 (即将提供)		
5	使用 Grove 多通道气体传感器进行气味识别	<ul style="list-style-type: none"> • 背景知识: 气体传感器的原理介绍 • 练习与实践: <ul style="list-style-type: none"> ◦ 创建与选择模型 ◦ 数据采集 ◦ 训练与部署 ◦ 使用与编程 • 机器学习理论 (超参数) 	Wio Terminal Grove - Multichannel Gas Sensor v2	9	创意项目和总结	<ul style="list-style-type: none"> • ML 理论知识总结 • 创意项目的例子

第 1 课

使用 Wio Terminal 和 Codecraft
学习 TinyML



理论知识

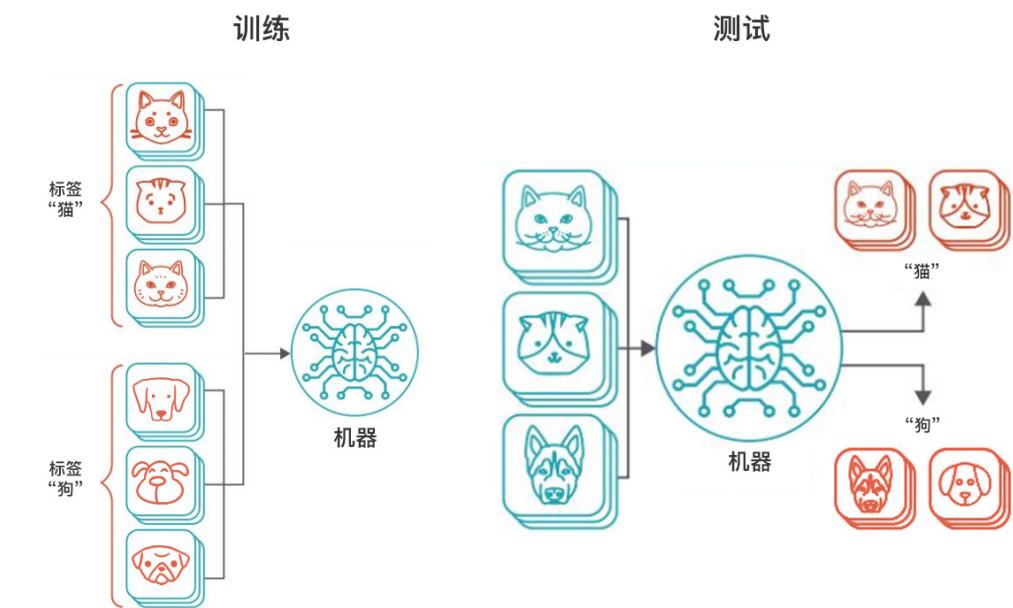
机器学习和深度学习

机器学习是人工智能 (AI – Artificial Intelligence) 的一个分支，主要是从数据中学习进而能输出分类并随着时间的推移、算法的增强、数据的增多提高其准确性的一个多领域交叉学科，它的编程方式并不适用于有规则的传统的编程。机器学习使机器能够像人类一样通过观察、分类数据来工作，并从错误中不断学习进步；而深度学习是机器学习的一个子集，它依靠深度人工神经网络（因此得名）使得机器可以从更加庞大的数据中学习。



人工神经网络 (ANN——Artificial Neural Network) 是人类试图模拟组成人类大脑的神经元网络所提出的一个方法，它是由计算机编程创建的，其行为方式就像相互连接的脑细胞一样，为了学习深度学习，人工神经网络是您所需要的必备的一个技能。

学习人工神经网络，首先您需要准备大量称为“训练集”的数据。当您尝试教人工神经网络如何区分猫和狗时，训练集将提供数千张标记狗和猫的图像供神经网络学习。一旦人工神经网络接受了大量数据的训练，它将能够根据它认为在不同单元中“看到或听到”（取决于训练集）的内容来对未来“看到或听到”的数据进行分类。在训练期间，机器的输出结果会与人类提供的对观察内容的描述进行比较，当它们相同时，机器验证为正确，而如果不正确，它会使用反向传播 (Backpropagation) 来调整其学习模式。反向传播的意思是不正确的信息会通过层 (layer) 返回从而调整数学方程（即算法）的过程，这可以称作“深度学习”的过程，这也是使网络智能化的原因。



什么是 TinyML (嵌入式机器学习)，为什么它如此重要？

通常来说，深度神经网络需要相当强大的计算资源来训练和部署。然而，最近出现了一个名为 TinyML (嵌入式机器学习) 的分支，它代表了机器学习和嵌入式系统中的一项技术（或研究领域），用来探索哪些机器学习应用程序（经过简化、优化和集成）可以在微控制器这类小型设备上运行。其常见的 Logo 如右图所示。



ML (Machine Learning) 表示机器学习，而 TinyML 中的 Tiny 是“微小”的意思，意味着 ML 模型经过优化，可以在极低功耗和小尺寸设备上运行，例如各种微型控制器 (MCU)。嵌入式设备有着各种形状和大小，他可以大到从“嵌入式超级计算机”NVIDIA Jetson Xavier AGX (如右图所示) 到最小的微控制器 ESP32 或者 Cortex M0。



右图是 GeeekNET ESP 32 的开发板：



为什么使用微型控制器的嵌入式机器学习被归入一个特殊类别，甚至有着自己的名字？

答案是它有着自己独特的优点以及特点。嵌入式机器学习的吸引力实际上在于 MCU 体积小可以无处不在，耗能少又相对便宜。以 ARM Cortex M0+ 和围绕它构造的小型 Seeeduino XIAO 板为例——该板只有拇指大小 (20x17.5 毫米)，仅消耗 1.33 毫安每时的电量，这意味着它可以连接 150 mA 电池之后可以连续工作 112 个小时（如果设备进入深度睡眠，这个时间还可以进一步增加）。做到这些，而它的成本仅为 5 美元。



由于最近模型优化的改进以及专门为在微控制器上运行机器学习模型而创建的框架的出现，为这些微型设备提供更多智能已经成为可能。我们现在可以在微控制器上部署神经网络，用于音频场景识别（例如，大象活动或玻璃破碎的声音）、热词检测（使用特定短语激活设备），甚至用于简单的图像识别任务。带有嵌入式微控制器的设备可为旧传感器赋予新的生命和意义。

借助 Codecraft 和 Wio Terminal，您现在就可以体验嵌入式机器学习的整个过程，而无需处理复杂的编程环境和丰富的编程知识。

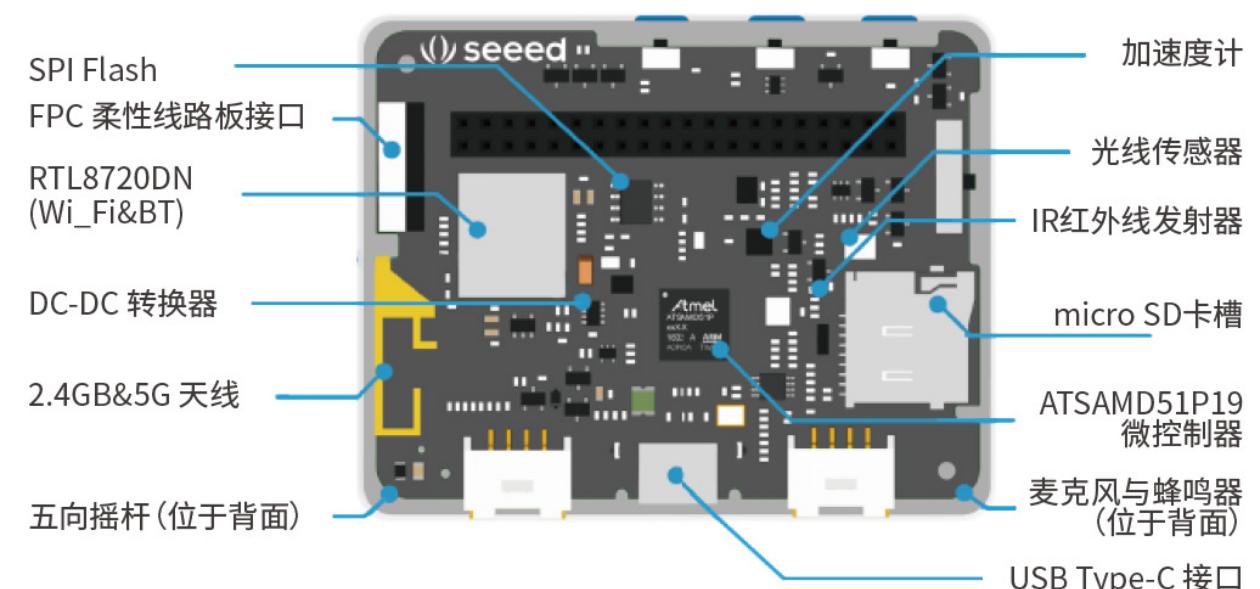
做好准备

Wio Terminal

Wio Terminal 对新手来说是学习物联网和嵌入式机器学习的一个完美的工具。它由矽递科技设计的并且支持多种嵌入式机器学习框架。



Wio Terminal 使用 ATSAMD51P19 微控制器和 ARM Cortex-M4F，运行频率为 120MHz（提升至 200MHz），它包含 4MB 外部闪存和 192kb RAM，并且支持 Realtek RTL8720DN 的无线连接，同时与 Arduino 和 MicroPython 兼容，支持蓝牙和 Wi-Fi，这为其物联网功能奠定了坚实的基础。Wio Terminal 有 2.4 英寸液晶屏、板载加速度计（LIS3DHTR）、麦克风、蜂鸣器、microSD 卡槽、光传感器和红外发射器（IR 940nm）。最重要的是，它有两个用于 Grove 生态系统的板载多功能 Grove 端口和与树莓派兼容的 40 针 GPIO 引脚，用来提供额外的附加支持。



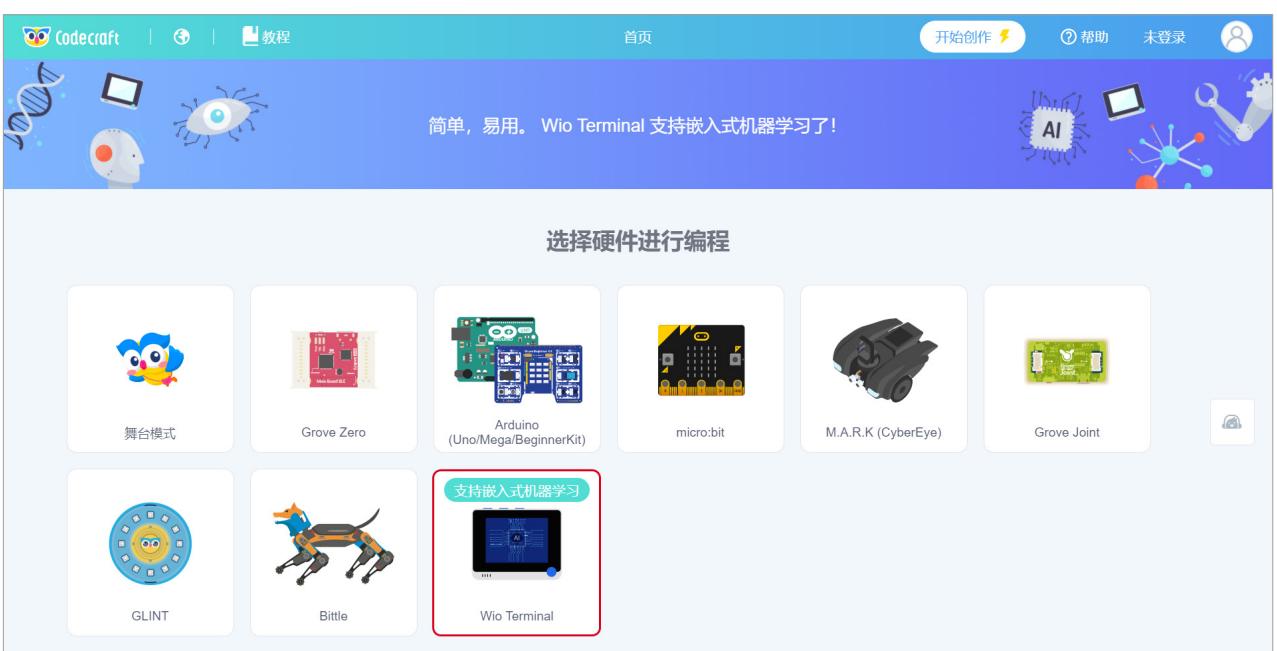
Codecraft

在软件方面，我们将使用图形化编程平台 Codecraft，由 Edge Impulse 为 Codecraft 提供机器学习的服务端技术支持，使用 Codecraft 的初学者可以轻松使用嵌入式机器学习。它为初学者从数据采集一直到模型部署到设备的整个过程都提供了支持，同时对整个 TinyML 的使用过程提供了友好又功能强大的 Web 界面和工具包。它是一个用户友好的开发平台，可以用于在嵌入式设备上进行机器学习。

所以，让我们快速了解如何使用并开始我们的机器学习之旅吧。



在 Codecraft 的主页点击 Wio Terminal 的图标，进入到 Wio Terminal 的嵌入式机器学习页面。



1、在开始创立和使用模型之前，您需要先登录（页面右上角）。新人使用则需要先注册。

2、点击“创建与选择模型”以进入“为嵌入式机器学习创建新模型”界面，这里可以看到Codecraft 给不同种类的机器学习框架，当中使用了不同种类的传感器。

3、所有机器学习模型处理部分分为四个步骤：

- 创建与选择模型：基于使用的传感器来创建模型
- 数据采集：使用 Wio Terminal 来采集数据
- 训练与部署：使用采集的数据来训练模型，等训练完成之后部署到 Wio Terminal 上
- 使用与编程：编程以使用模型

一、创建与选择模型

① 创建与选择模型：嵌入式机器学习的第一步。点击“创建与选择模型”就可以看见“为嵌入式机器学习创建新模型”的页面。

② 返回 Codecraft 主页：点击以返回 Codecraft 主页，如果您还没有保存当前程序，它会在您离开之前提示保存。

③ 语言：点击后会出现语言选项，您可以在此处更改语言。

④ 文件：您可以创建一个新的在线项目或打开一个本地项目，您也可以按照屏幕上的说明将项目从计算机保存到云端。

⑤ 教程：在这里您可以看到 Codecraft 的相关例程。

⑥ 保存：您可以修改项目名称并将其保存到云端（Codecraft 应该连接到互联网并且您应该使用您的 Codecraft 账户登录）。

⑦ 模块 / 代码：您可以在模块和代码之间切换，这里使用的编程语言是 C。

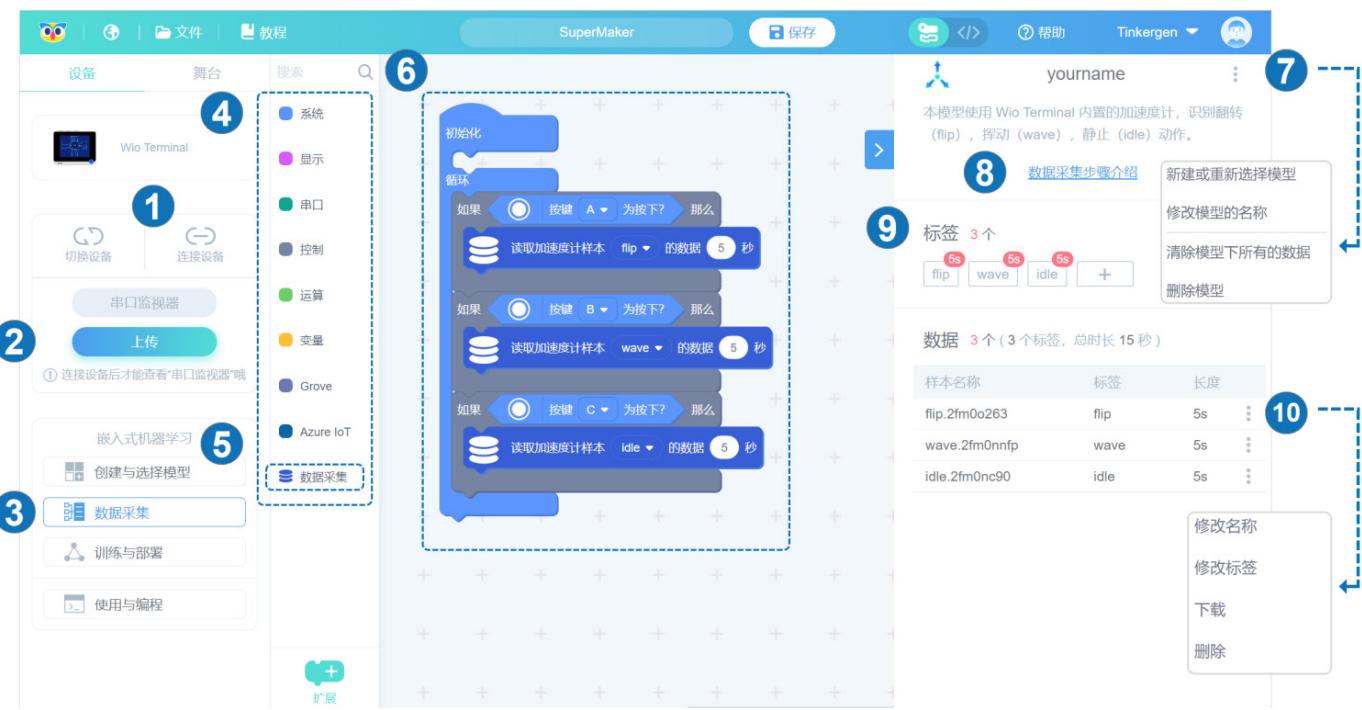
⑧ 帮助：我们始终感谢您对 Codecraft 的建议，有您的参与，我们会做得更好。

⑨ 登录：您可以看到您的云项目、账户设置、我的邀请码还有退出键，如果您还没有登录，它会提示您进行登录。

⑩ 据传感器选择模型框架：您可以根据您要使用的传感器在此处创建模型。单击相应的图标以使用对应的传感器创建模型，创建新模型后，界面会自动跳转到“数据采集”界面。

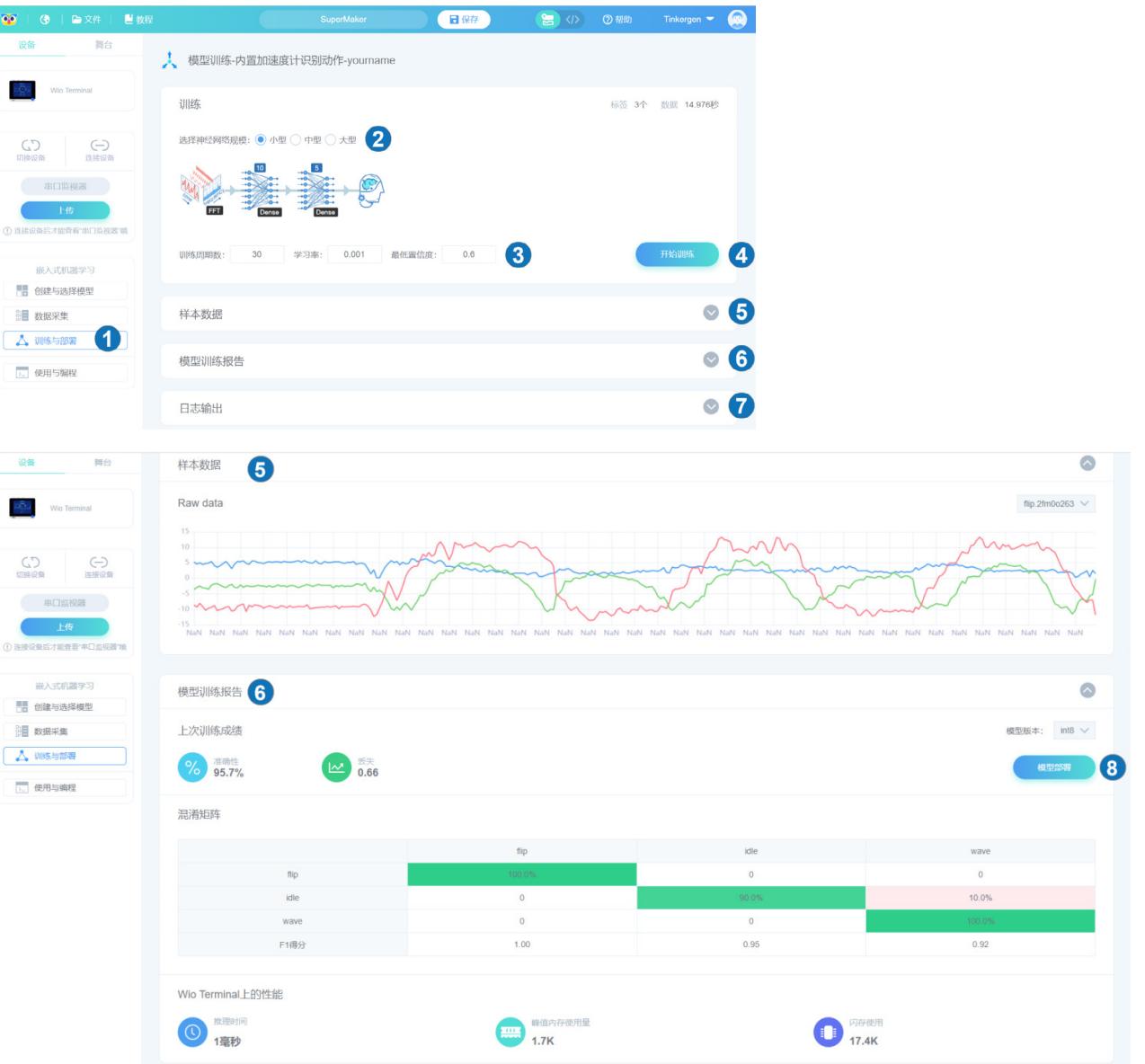
⑪ 我的模型：您可以单击此处查看您创建的所有模型。

二、数据采集



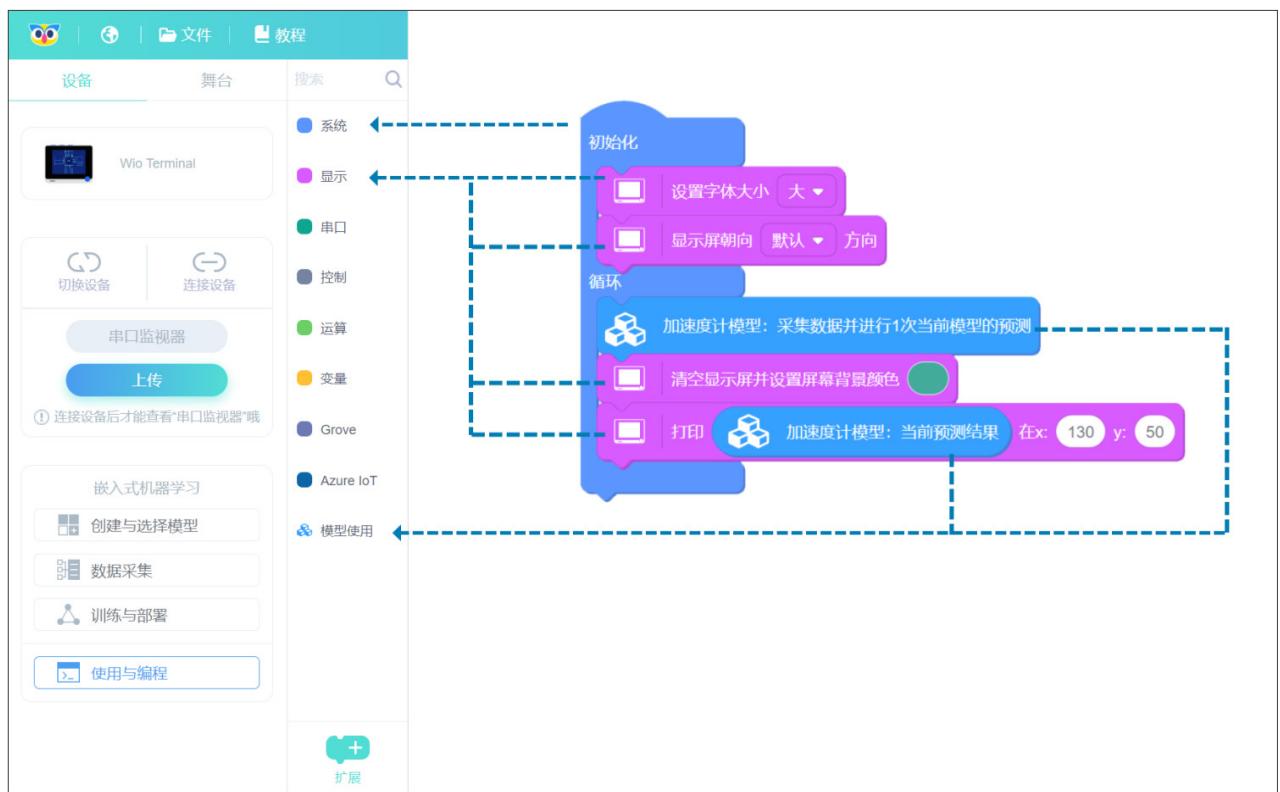
- ① **连接设备:** 您可以在这里连接您的设备。
- ② **上传:** 点击这里来上传您的程序。
- ③ **数据采集:** 嵌入式机器学习的第二步。
- ④ **模块类别:** 您可以在这里选择您的模块分类。
- ⑤ **数据采集:** 您可以获取“默认数据采集程序”以及与样本数据相关的模块。
- ⑥ **默认数据采集程序:** 您可以上传默认数据采集程序以快速开始您的数据采集。
- ⑦ **模块上的命令:** 在这里您可以重命名模块或者删除模块和它的数据。
- ⑧ **数据采集步骤介绍:** 单击此处将显示一个弹出窗口，其中包含数据收集过程的详细信息
- ⑨ **标签:** 数据采集界面右侧有默认标签，您在这里可以添加和删除标签。添加自定义标签：单击“+”，删除标签：将鼠标悬停在标签上，标签左上角会出现一个“x”，点击“x”则可以将其删除，修改标签：点击出现的标签就会弹出修改标签窗口。
- ⑩ **数据上的命令:** 在这里你可以删除，重命名和下载数据。

三、训练与部署



- ① **训练与部署:** 嵌入式机器学习的第三步。
- ② **选择神经网络规模:** 您可以选择合适的神经网络规模：小型、中型、大型。
- ③ **训练周期数:** Codecraft 为模型的每个比例提供默认参数值。您也可以自行设置训练周期数（正整数）、学习率（0~1 小数）和最低置信度（0~1 小数）。
- ④ **开始训练:** 点击“开始训练”来训练模型。
- ⑤ **样本数据:** 在这里可以观察收集到的数据样本。
- ⑥ **模型训练报告:** 您可以在 Wio Terminal 上观察模型的准确率、损失和性能等训练结果。如果训练结果不理想，您可以回到第一步的训练模型，选择另一个规模大小的神经网络或调整参数设置来重新训练，直到得到一个结果满意的模型。
- ⑦ **日志输出:** 在训练过程中，“日志”屏幕会显示训练的日志。通过观察“日志”即可推断模型的性能。
- ⑧ **模型部署:** 点击这里来部署理想的模型

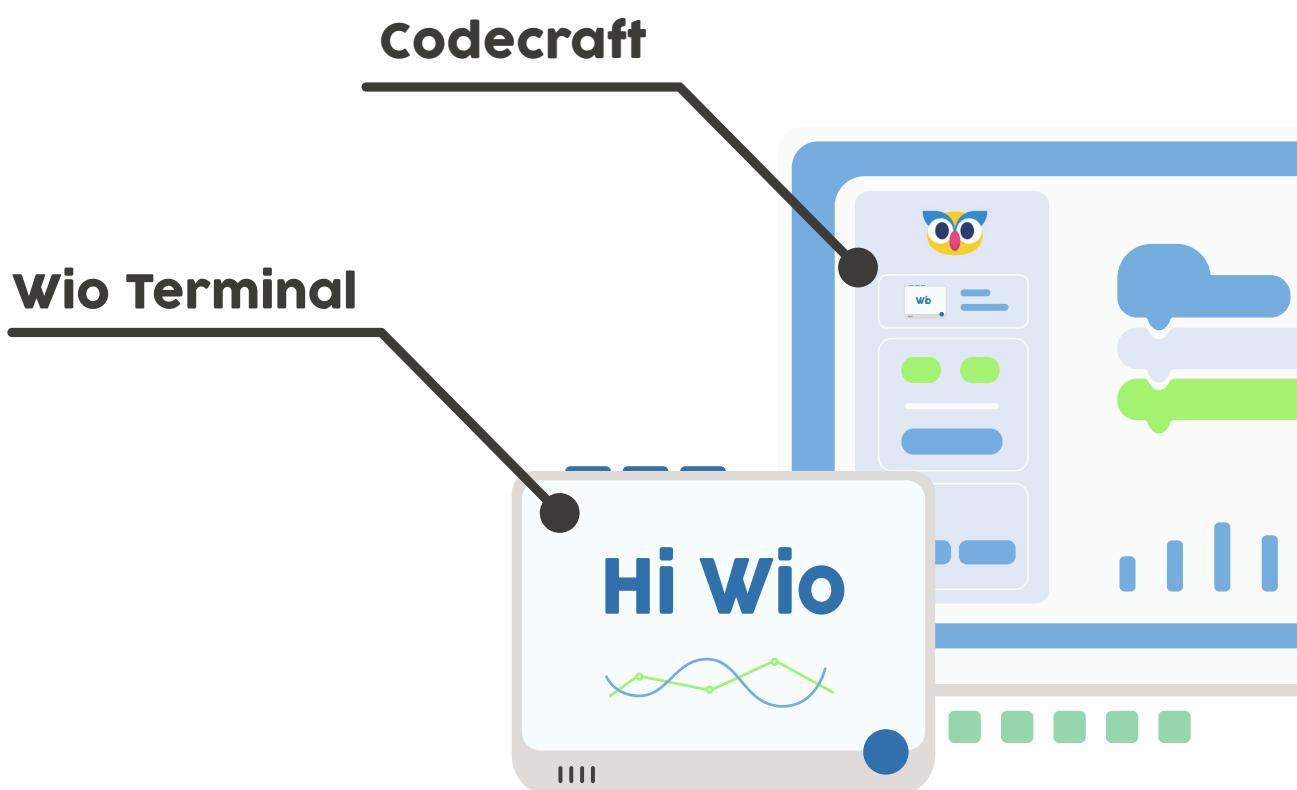
四、使用与编程



使用与编程：**嵌入式机器学习的第四步**，您可以通过拖动模块来编写程序，课程还提供了最简洁的示例代码，您可以根据颜色来找到相对应的模块。

★ 示例教程

为了帮助理解嵌入式机器学习的使用，我们提供 7 个模板项目。您可以根据这些模型示例来学习如何使用 Codecraft 进行嵌入式机器学习。我们希望这些示例可以帮助您构建更多更有趣的应用程序。



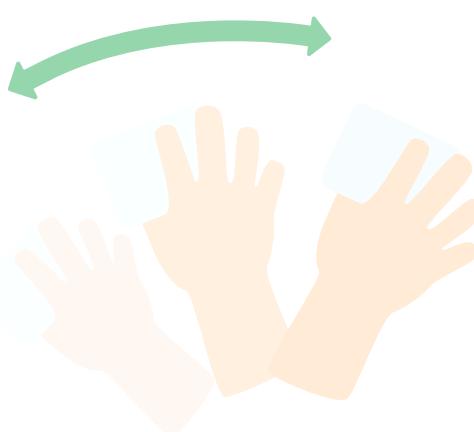
第 2 课

使用内置加速度计进行运动识别



项目概述

在这节课中，我们将要进行一个手势识别的任务，也就是尝试运用 Wio Terminal 的内置的三轴加速度计来识别三种手势，包括翻转，挥动以及静止。



如果使用基于算法的传统编程手段，这将会非常棘手。对于传统编程的机器来说，手势必须以完全相同的方式执行才能被识别，而这在现实当中几乎不可能，其中涉及众多因素，以运动因素为例：我们手势的摆动速度、转动角度以及静置方向，严格意义上来说都不会完全相同，而这些微小变化都需要使用传统编程来制订一套规则以减少识别错误。所以如果同样使用传统编程来解决这个手势识别问题时，就需要为每种操作模式制定数百条不同的规则。

值得庆幸的是，**机器学习**可以非常简单的处理这些变化，而不需要冗杂的规则程序。

预期结果

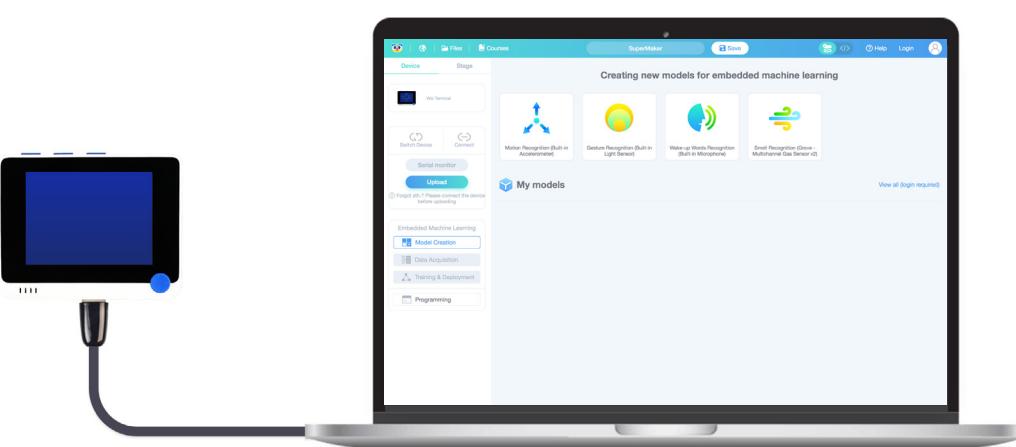
Wio Terminal 实时显示当前 Wio Terminal 动作的识别结果：翻转、挥动以及静止。



准备工作

硬件需求：Wio Terminal

连接方法：



背景知识

可能正如你想的那样，加速度计是用来测量一个物体加速度的传感器，它的输出结果来自物体的速度变化。加速度计可以测量速度随时间的变化，常见的加速度为重力加速度，一个重力加速度等于 9.8 米每平方秒 (m/s²)，表示每秒钟就有 9.8 米每秒 (m/s) 的速度在增加或减少。

世界上有很多种加速度的计算测量方法，最早的加速度是由一个机械建筑演变而来的。第一个加速度计是阿特伍德机，它是由英国物理学家乔治·阿特伍德发明的。

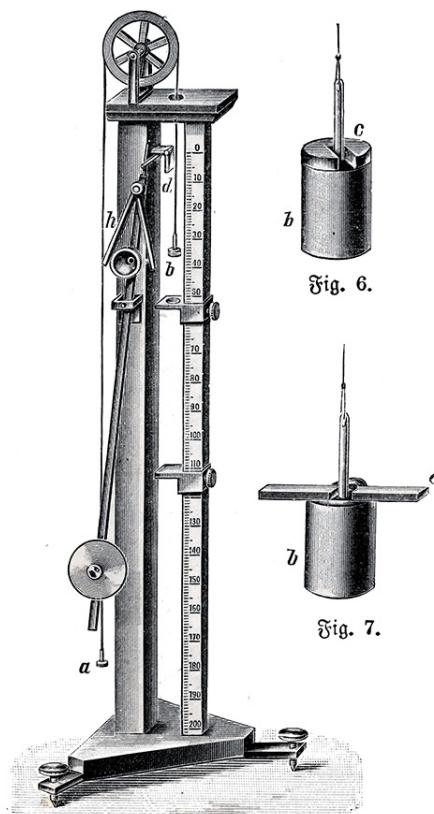
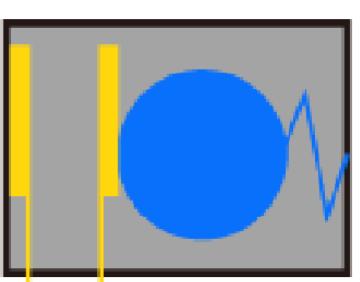


Fig. 5.

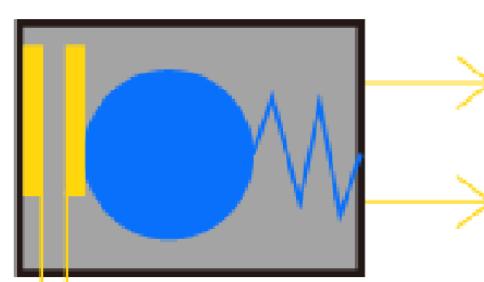
阿特伍德机

加速度计通常由质量块、阻尼器、弹性元件、敏感元件和适调电路等部分组成，根据传感器敏感元件的不同，加速度计可以分为电容式、压电式等；根据输入轴又可以分为单轴、双轴和三轴。一般使用在手机中的加速度计是微机电系统（Micro-electromechanical systems 简称 MEMS）加速度计，我们在 Wio Terminal 中使用的是就三轴数字加速度计（LIS3DHTR）。

加速度计的原理一般情况下是通过传感器感知受到力的作用，再经过内部电路（电容变化）将受到的力转换为电信号输出。比如 MEMS 加速度计，它的模型其实是一个由弹簧质量块组成的阻尼系统，当质量块由于惯性作用压缩弹簧的时候会产生一个弹性力，将弹簧与内部电路相连，电容会随着弹簧的移动而变化，通过胡克定律和电容变化我们可以知道弹性力与电信号的关系，而后用牛顿第二定律即可用电信号表示加速度。



1. 质量块与电极板静置



2. 质量块按压电极板使得电容变化，输出电信号

三轴数字加速度计在原理上与 MEMS 加速度计大同小异，另外可以让它测量更加精准的原因是它同时拥有三个传感器来识别三个轴向力的作用。

练习与实践

课程步骤

1. 创建与选择模型
2. 数据采集
3. 训练与部署
4. 使用与编程



步骤 1、创建与选择模型

1.1 建立“运动识别（内置加速度计）”模块

点击“创建与选择模型”，然后点击“内置加速度计识别动作”，如下面步骤 1 和步骤 2 所示。

给新建的模型命名。点击“确认”就会自动跳转到数据采集界面。

步骤 2、数据采集

2.1 默认标签

2.2 连接设备并上传 Codecraft 中的默认数据采集程序

Wio Terminal 连接后，在 Codecraft 界面，点击 **上传** 即可上传默认数据采集程序。

如右图所示将出现一个“上传”弹出窗口。

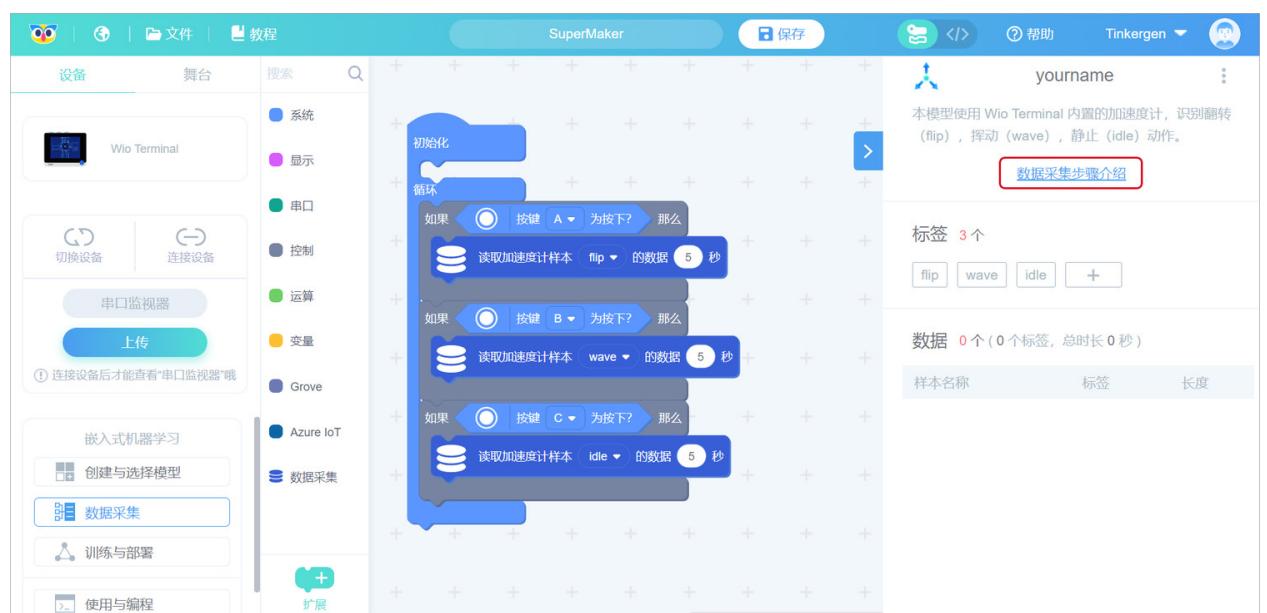
选择与当前 Wio Terminal 对应的串口号（不一定为下图的 COM14），点击“确定”按钮。





2.3 数据采集

在右上角的超链接中, 您将找到数据采集的分步介绍。请按照说明收集数据。



注意:

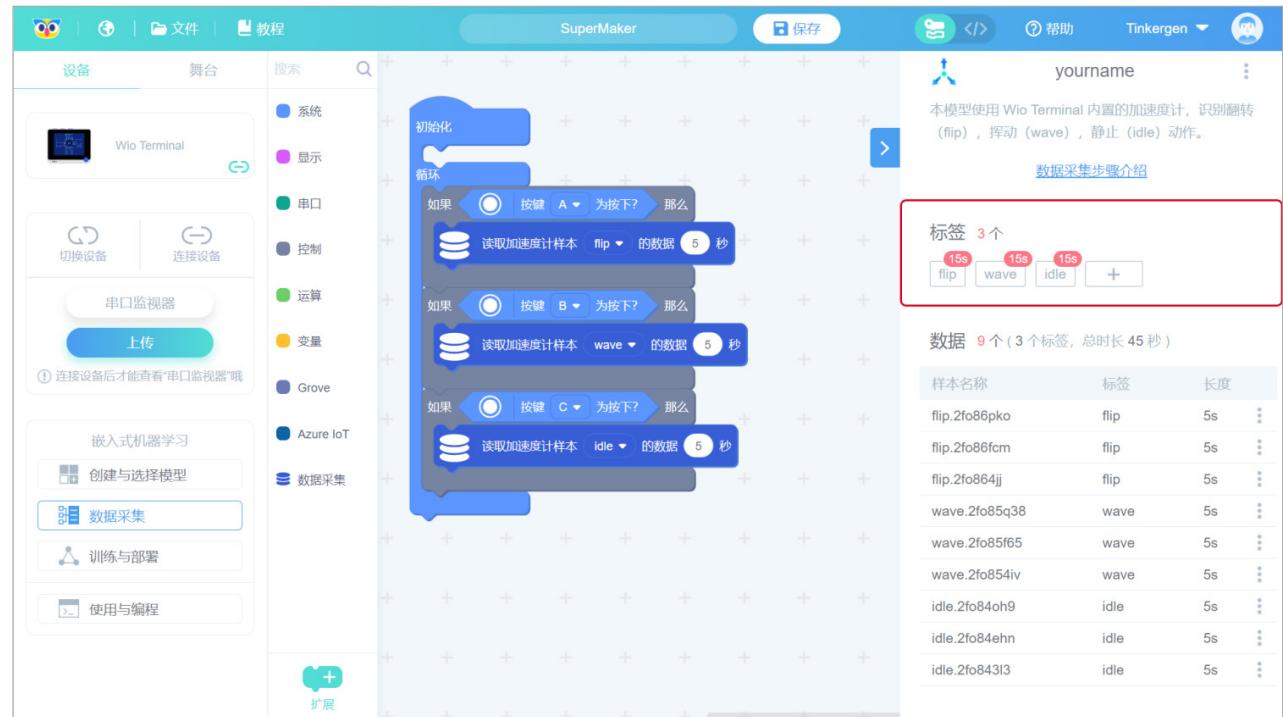
- Wio Terminal 按钮位置。
- 动画是已经加速过的, 实际动作可能有所缓慢。
- 请注意红色提示。
- 将光标指向描述文本以获得更详细的内容。



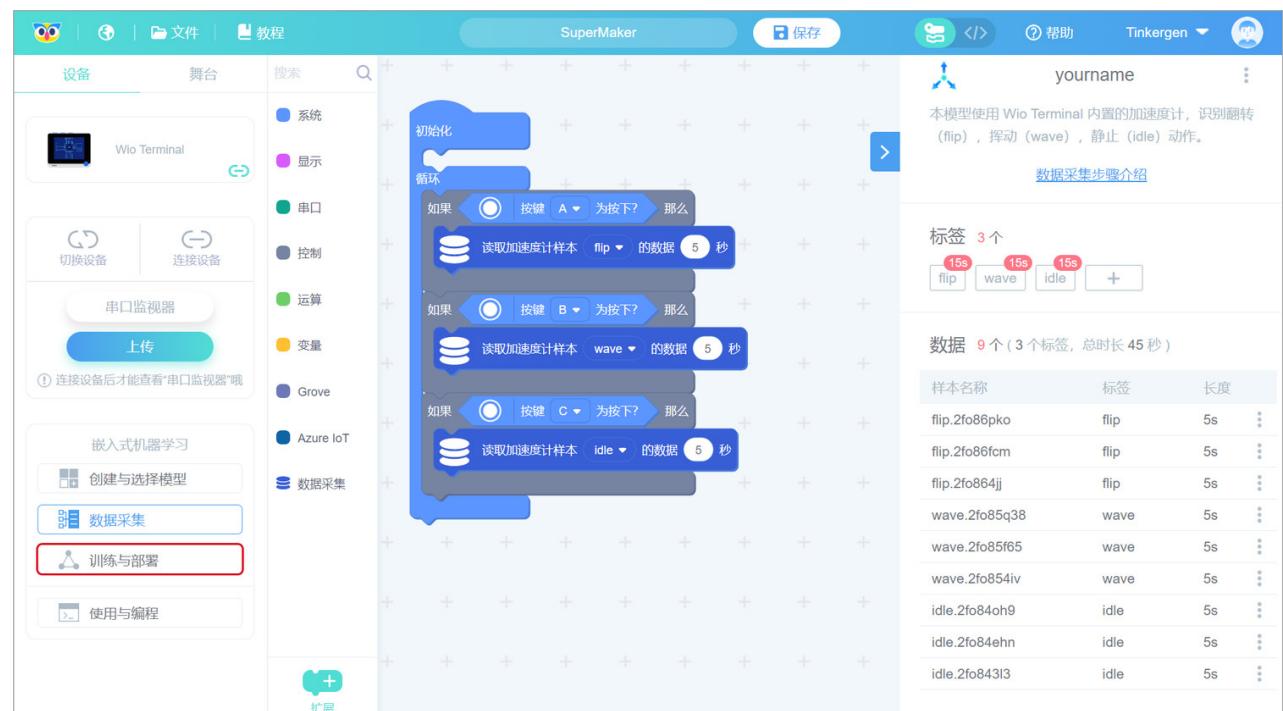


Wio Terminal 虽然显示数据采集完成，但 CodeCraft 仍在上传数据，数据从 Wio Terminal 传输到 CodeCraft 需要 1~2s。

至此，数据全部采集完成。



单击“训练与部署”。



步骤 3. 训练与部署

3.1 设置神经网络和参数

选择你觉得合适的神经网络规模：小型、中型、大型。

接着设置参数、训练周期数（正整数）、学习率（从 0 到 1 的数字）和最小置信度（从 0 到 1 的数字）。初始页面已提供一些默认参数值。



3.2 开始训练模型

点击“开始训练”。

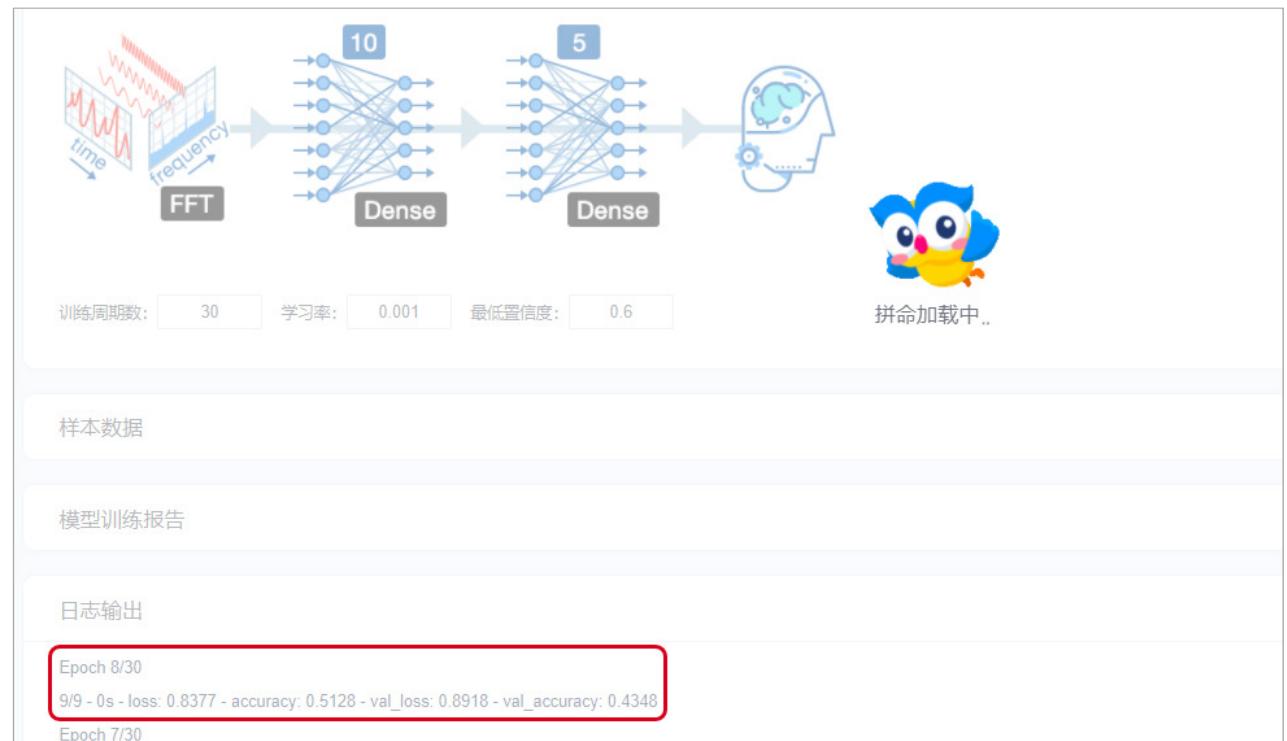


当您点击“开始训练”时，界面会显示“拼命加载中...”。

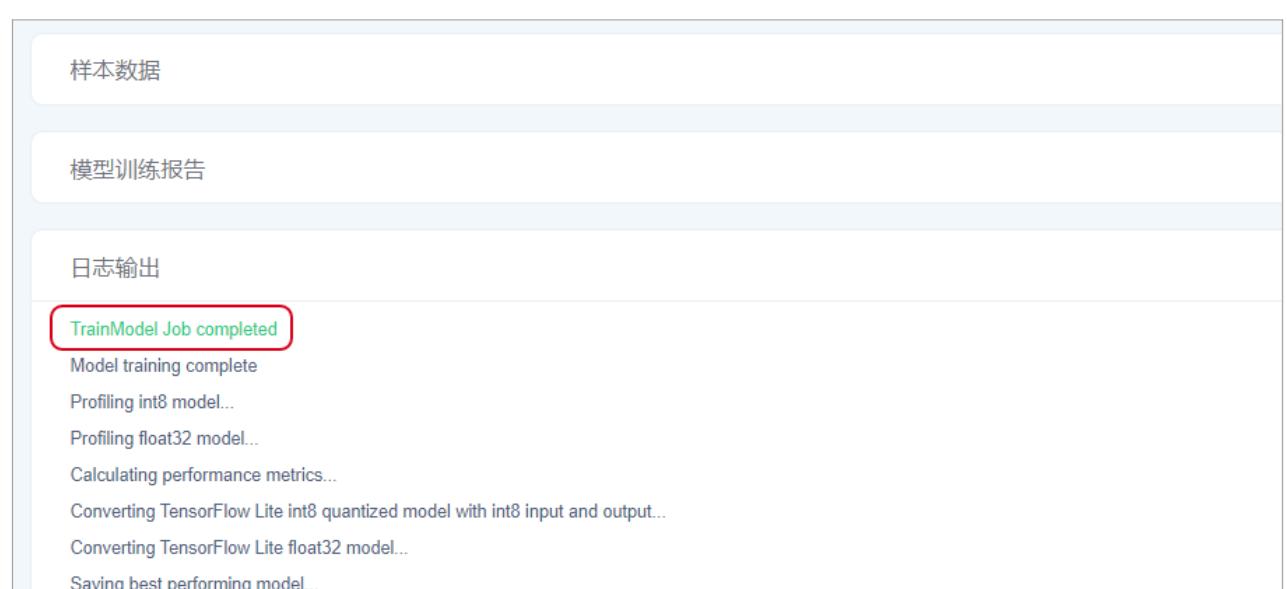


“拼命加载中...”的持续时间取决于所选神经网络的规模（小型、中型、大型）和训练周期数。通常来说，网络规模和训练周期数越大，所需的时间就越长。

您还可以通过观察“日志输出”来推断等待时间。下图中，“Epoch: 8/30”表示训练总轮数为 30 轮且已经训练了 8 轮。



等待加载完后，“日志输出”中会出现“TrainModel Job completed”表示训练完成，界面上会出现“模型训练报告”。



3.3 观察模型性能，选择理想模型

在“模型训练报告”窗口中，您可以观察训练结果，包括模型的准确率、损失和性能。

如果训练结果不理想，您随时可以回到第一步训练模型，选择另一个大小的神经网络，调整参数训练等，直到得到一个结果满意的模型。

上次训练成绩

%	准确性 98.6%	丢失 0.40
---	--------------	------------

混淆矩阵

	flip	idle	wave
flip	100.0%	0	0
idle	3.6%	96.4%	0
wave	0	0	100.0%
F1得分	0.98	0.98	1.00

Wio Terminal上的性能

推理时间 1毫秒	峰值内存使用量 1.7K	闪存使用 17.4K
-------------	-----------------	---------------

3.4 部署理想模型

在“模型训练报告”窗口中，点击“模型部署”。

上次训练成绩

%	准确性 98.6%	丢失 0.40
---	--------------	------------

混淆矩阵

	flip	idle	wave
flip	100.0%	0	0
idle	3.6%	96.4%	0
wave	0	0	100.0%
F1得分	0.98	0.98	1.00

Wio Terminal上的性能

推理时间 1毫秒	峰值内存使用量 1.7K	闪存使用 17.4K
-------------	-----------------	---------------

上次训练成绩

%	准确性 98.6%	丢失 0.40
---	--------------	------------

混淆矩阵

	flip	idle	wave
flip	100.0%	0	0
idle	3.6%	96.4%	0
wave	0	0	100.0%
F1得分	0.98	0.98	1.00

等到部署完成后，点击“确认”即可跳转到“使用与编程”窗口。



步骤 4. 使用与编程

4.1 编写使用模型的程序

在“使用与编程”界面，点击“模型使用”以调用部署的模型。

设备 舞台 搜索

Wio Terminal

切换设备 连接设备 串口监视器 上传

① 连接设备后才能查看“串口监视器”哦

嵌入式机器学习

- 创建与选择模型
- 数据采集
- 训练与部署
- 使用与编程

模型使用

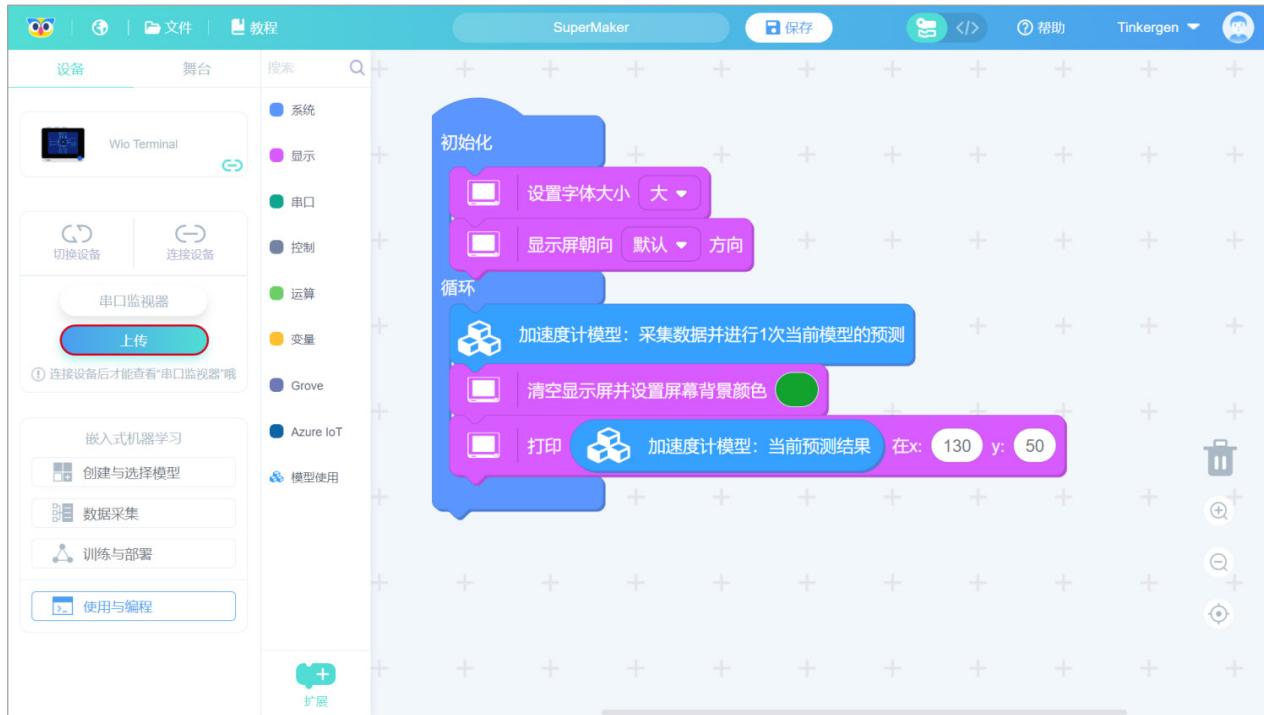
- 加速度计模型：采集数据并进行1次当前模型的预测
- 加速度计模型：当前预测结果
- 加速度计模型：当前 flip 的置信度
- 加速度计模型：当前识别结果为 flip ?

您可以尝试通过编写以下程序来调用您的模型。



4.2 将程序上传到 Wio Terminal

单击“上传”按钮。



选择与当前 Wio Terminal 对应的串口（不一定如图所示的 COM14），点击“确定”按钮。



第一次上传时间通常比较长，并且随着模型的复杂性时间会进一步增加。较小型号的上传时间约为 4 分钟，较大型号可能会消耗更长时间，这取决于您机器的性能。



4.3 Wio Terminal 测试模型

翻转您的 Wio Terminal，查看其屏幕是否显示“翻转”。可以再尝试其他的动作，看看 Wio Terminal 是否识别您的动作。

恭喜！您已经完成了第一个嵌入式机器学习模型。相信你一定有兴趣了解自己做的每一个操作的意义，那么让我们首先来学习机器学习的基础理论知识，先看看我们的输入（input）。

机器学习理论（输入，标签，数据集）：

输入（input）：

- 标签
- 数据集
- a. 训练集、验证集和测试集
- b. 数据量
- c. 获取数据的方式
- d. 数据集的质量

数据 9 个 (3 个标签, 总时长 45 秒)		
样本名称	标签	长度
flip.2fo86pk0	flip	5s
flip.2fo86fcm	flip	5s
flip.2fo864jj	flip	5s
wave.2fo85q38	wave	5s
wave.2fo85f65	wave	5s
wave.2fo854iv	wave	5s
idle.2fo84oh9	idle	5s
idle.2fo84ehn	idle	5s
idle.2fo843l3	idle	5s

输入：

1. 标签

1.1 什么是标签：

顾名思义，当您为数据添加有意义的标签或为您收集的原始数据分配类别时，数据将会被标记，这意味着您向数据设置了其对应目标，AI 会通过这些数据标签来学习。

现在，让我们看看这是如何做的。假设我们有一组标有“柠檬”、“橙子”或“香蕉”的水果，模型会通过这些标记的水果进行分类训练，当训练结束后，如果这时输入有新的水果且没有贴标签，则会被模型判断并贴上标签并归入这三组中的其中一组，注意输入元素有可能同时属于多个类别。



2. 数据集

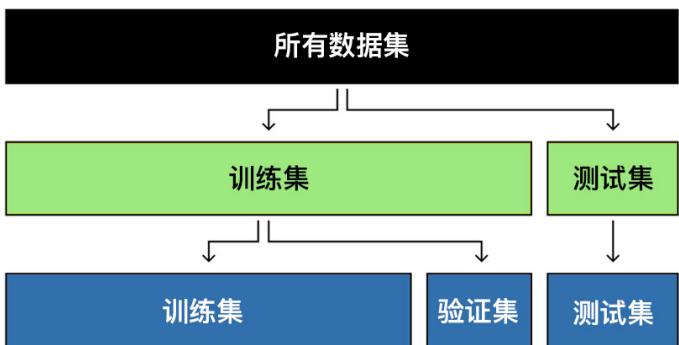
2.1 什么是数据集：

如果您想要一个可运行且可靠的模型，则需要大量相关的高质量数据。这意味着您必须通过提供数

万个示例来训练您的 ML 模型，并且让它记住您的数据片段之间的相关性。

完整数据集由训练集和测试集组成：

您提供给模型的数据称为训练数据集，而您提供给模型进行预测的数据则称为测试集。



此外，训练集可以分为两组。我们可以在开始训练后观察日志，发现训练数据集会被拆分为训练集和验证集。

```

9/9 - 1s - loss: 2.4400 - accuracy: 0.2711 - val_loss: 2.0213 - val_accuracy: 0.0000e+00
Epoch 1/30
Training model...
Training on 273 inputs, validating on 69 inputs
Splitting data into training and validation sets OK
Splitting data into training and validation sets...
  
```

这里的 Training inputs 即为训练集，而 validating inputs 为验证集。

训练集：训练集是模型用来学习的数据。

验证集：使用训练集进行训练后，使用验证集中的点来计算分类器的准确率或误差。这里的关键是我们知道验证集中每个点的真实标签，但我们暂时假装不知道，也就是我们假装将验证集中的点以没有标签的形式输入到模型分类器，然后我们会收到该点的分类和标签。完成后我们可以查看验证点的真实标签是否与收到的标签一致，看看我们是否做对了。如果我们对验证集中的每个点都这样做，我们就可以计算验证错误！

2.2 那多少训练数据才能够呢

如果要一个简短的回答是，这没有准确的答案。如果要稍微长一点的答案的话，那就是每个特定项目都需要不同的特定数量的训练数据，而知道需要多少训练数据的唯一方法就是反复试验。

这里有一些指南可以帮助您了解为什么这个问题并没有准确答案：

- 通常来说，具有更多属性和更多连接网络且更复杂的模型（如人工神经网络）将需要更多数据才能正确训练。
- 模型的应用范围，以及模型接受训练之后用来预测的现实生活的复杂性也会影响所需训练数据的多少，特别是异常和盲点。
- 随着时间的推移，您可能需要重新训练或调整您的模型，因为它预测的趋势会发生变化，从长远来看，这将需要更多数据。

训练集的数据量取决于多种因素，对经验丰富的工程师来说，他们大体上对训练模型的数据量有一定的了解，你可以去论坛、科学社区等一些地方来找到他们向他们请教从而精进学习。

2.3 如何获取数据



- 自己动手：将 Wio Terminal 与传感器或者其他设备一起使用。
- 开源社区：一些数据科学社区，如 Kaggle。
- 网络搜索：根据您的目标，在线搜索您想要的数据集。

2.4 如何判断数据集的质量：

准确性：准确性是指数据描述是否符合其在现实世界条件中的描述。如果您尝试教计算机识别猫，则需要将现实中的猫标记为“猫”而不是“狗”。

关联性：您收集的数据也应该对您准备将其用于的模型中有用。如果它与您的目标不相关，则它对您并没有用处，应当舍弃。如果您想教电脑通过图片识别猫，则并不需要提供猫粮的照片。总的来说，为您的数据收集设定目标很重要，这样才知道要什么样的数据才是有用的。

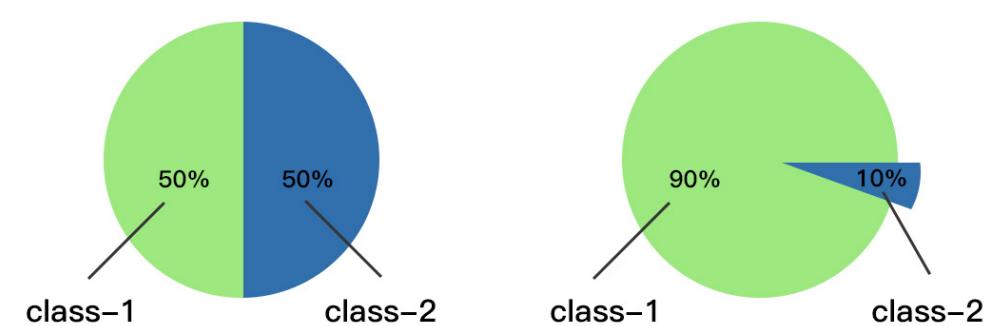
数据规模：如果您尝试教计算机识别猫，则需要显示数千张大小、颜色和形状各不同的猫的图像，以便您的模型能够准确识别猫。

平衡性：平衡性是指训练集中的数据分类是否均衡，不平衡尤为常见：大多数分类数据集在每个类别中的实例数量并不相同，但微小的差异并不重要。

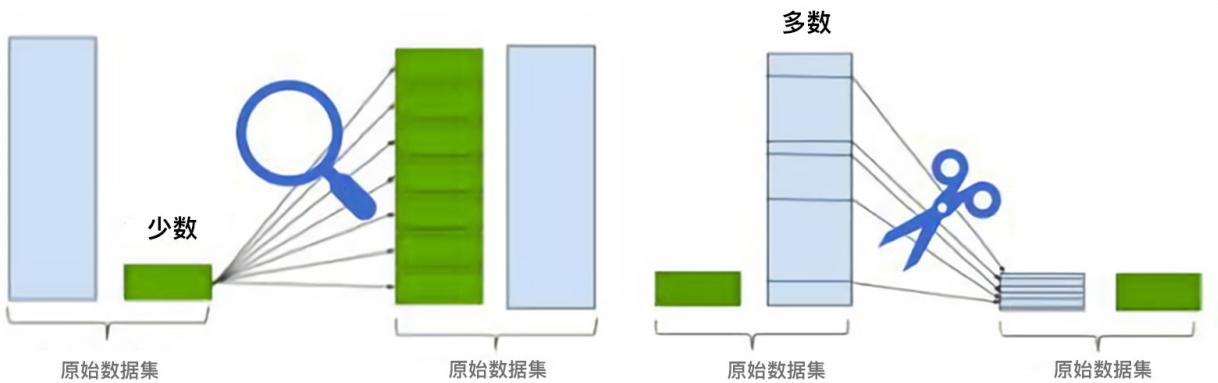
而当不平衡性足够大时，我们的模型会发生什么？

我们的模型通过学习数据并且“巧妙”地进行分类，得到的结果是预测“1 类”竟然达到 90% 的高准确度。

关于平衡和不平衡数据的例子



那么如何解决失衡呢?



过采样 (Up Sampling) : 增加训练集中少数类别的数量。

欠采样 (Down Sampling) : 减少多数样本的数量以平衡分类分布。

★ 总结

1. 背景理论知识:

- 加速度计

2. 嵌入式机器学习实践



3. 机器学习理论 (输入)

- 标签: 为您收集的原始数据贴上标签以分配类别。

· 数据集:

- a. 训练集、验证集和测试集: 训练集来输入模型, 验证集用于验证结果以改进模型, 测试集用于预测。
- b. 所需数据量: 合适的规模 (一般情况下越多越好)。
- c. 数据获取方式: 自己获取, 开源社区, 互联网搜索。
- d. 数据集的质量: 准确性、关联性、数据规模、平衡性。



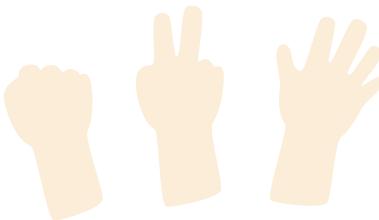
第3课

使用内置光线传感器进行手势识别



项目概述

在这节课当中我们将使用 Wio Terminal 来识别手势：利用内置光线传感器来识别石头、剪和布。



如果使用基于算法的传统编程手段，这将会非常棘手。对于传统编程的机器来说，手势必须以完全相同的方式执行才能被识别，而这在现实当中几乎不可能，其中涉及众多因素，以运动因素为例：我们手势的摆动速度、转动角度以及静置方向，严格意义上来说都不会完全相同，而这些微小变化都需要使用传统编程来制订一套规则以减少识别错误。所以如果同样使用传统编程来解决这个手势识别问题时，就需要为每种操作模式制定数百条不同的规则，但**机器学习**可以非常轻松地处理这些变化。

一般来说，用机器学习识别手势应当是使用相机传感器。但我们使用的是光线传感器，而使用光线传感器就相当于仅仅使用相机的一个像素点来识别手势，这简直是另一个等级的挑战，那现在就让我们用这个项目挑战一下吧！

预期结果

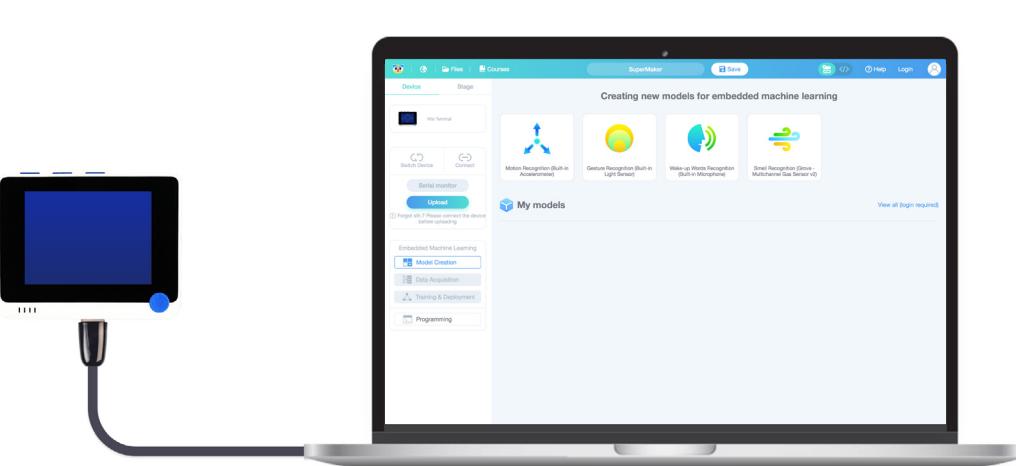
如下图所示，Wio Terminal 实时显示了当前手势识别的结果：石头、剪刀和布之一。



准备工作

硬件需求：Wio Terminal

连接方法：



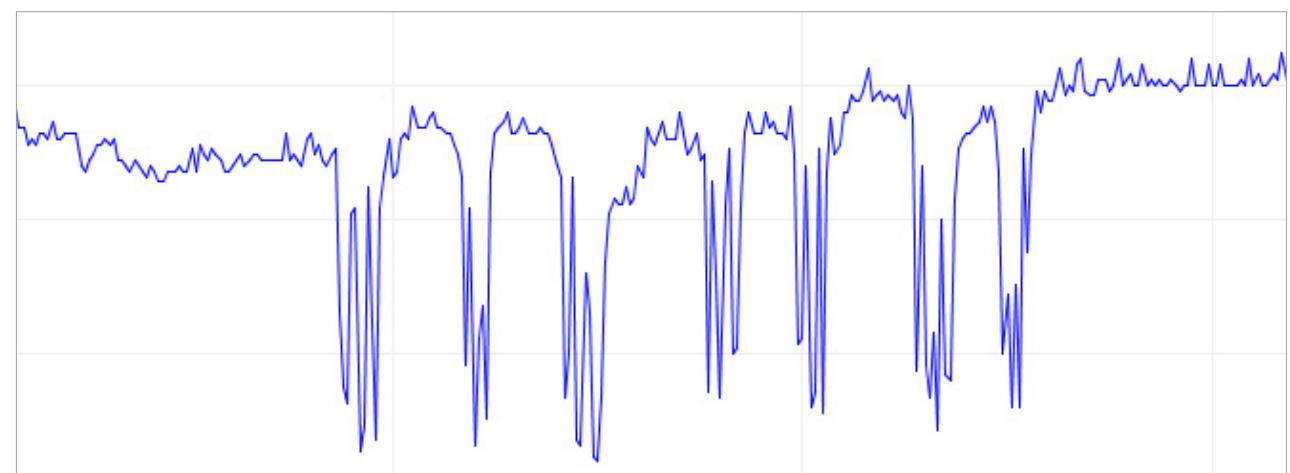
背景知识

在本课中，我们将训练和部署一个简单的神经网络，对石头、剪刀、布手势进行分类，借助仅能探测到一个像素点的光线传感器。

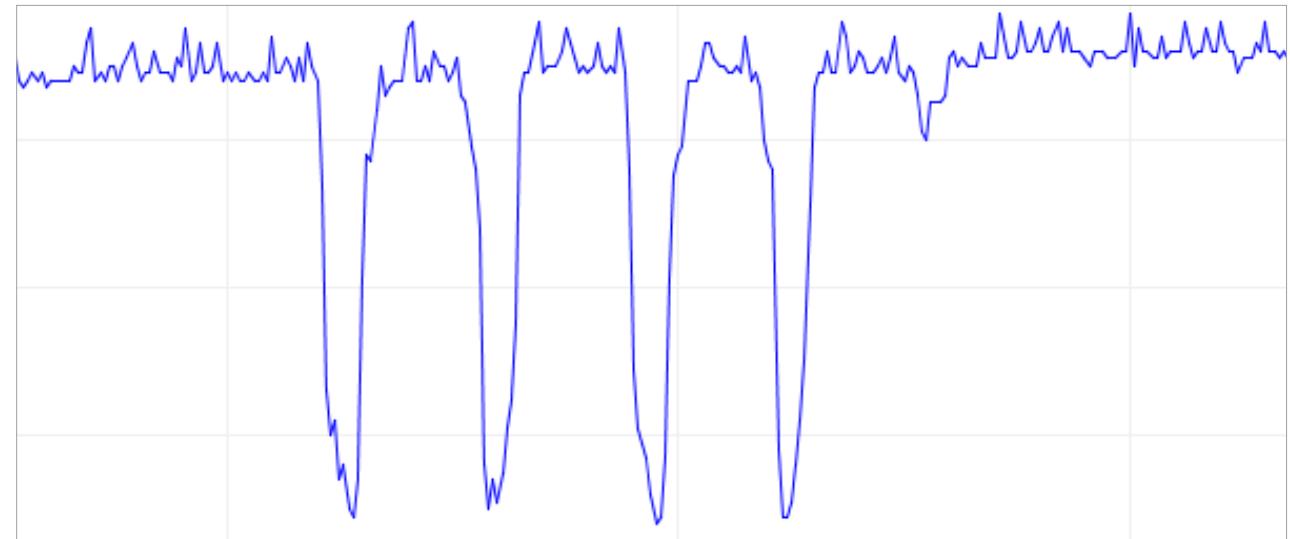


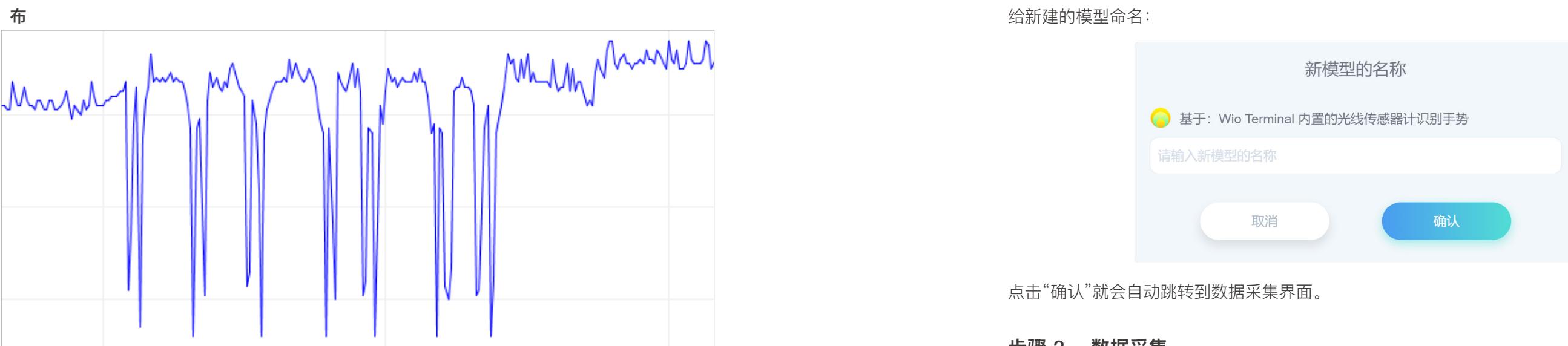
这个项目的工作原理其实很简单——在光线传感器上方移动的不同手势会在特定时间段内阻挡一定量的光。以手势的石头 🪨 为例子，传感器一开始会是高值（传感器上方没有东西），而当手经过传感器上方时值会降低，待其完全经过后又是同样的高值。同样的，在检测手势的布 🧺 时，其中的每个手指都会经过一遍传感器上方，我们就可以得到“高 – 低 – 高 – 低 – 高 – 低 – 高 – 低 – 高”的值。

剪刀



石头





这些手势通过传感器时的速度和幅度的在很大差异，这些差异对于传感器探测到的数据，是应当使用机器学习模型而不是传统编程的一个很好的案例。

练习与实践

课程步骤

1. 创建与选择模型
2. 数据采集
3. 训练与部署
4. 使用与编程

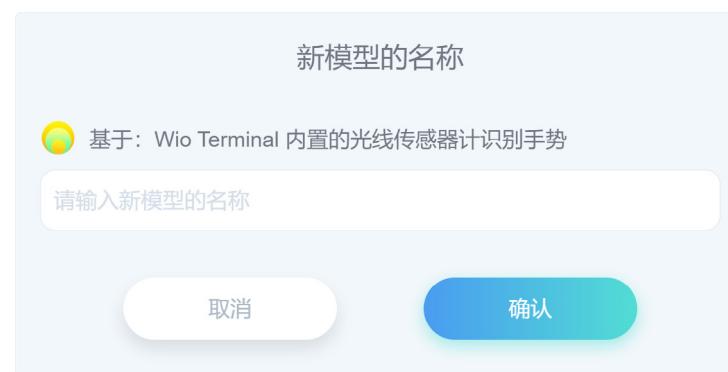
步骤 1、创建与选择模型

1.1 建立“手势识别（内置光线传感器）”模块

点击“创建与选择模块”，然后点击“内置光线传感器识别手势”，如下面步骤 1 和步骤 2 所示：



给新建的模型命名：



点击“确认”就会自动跳转到数据采集界面。

步骤 2、数据采集

2.1 默认标签

标签 3 个

rock paper scissors +

标签 3 个

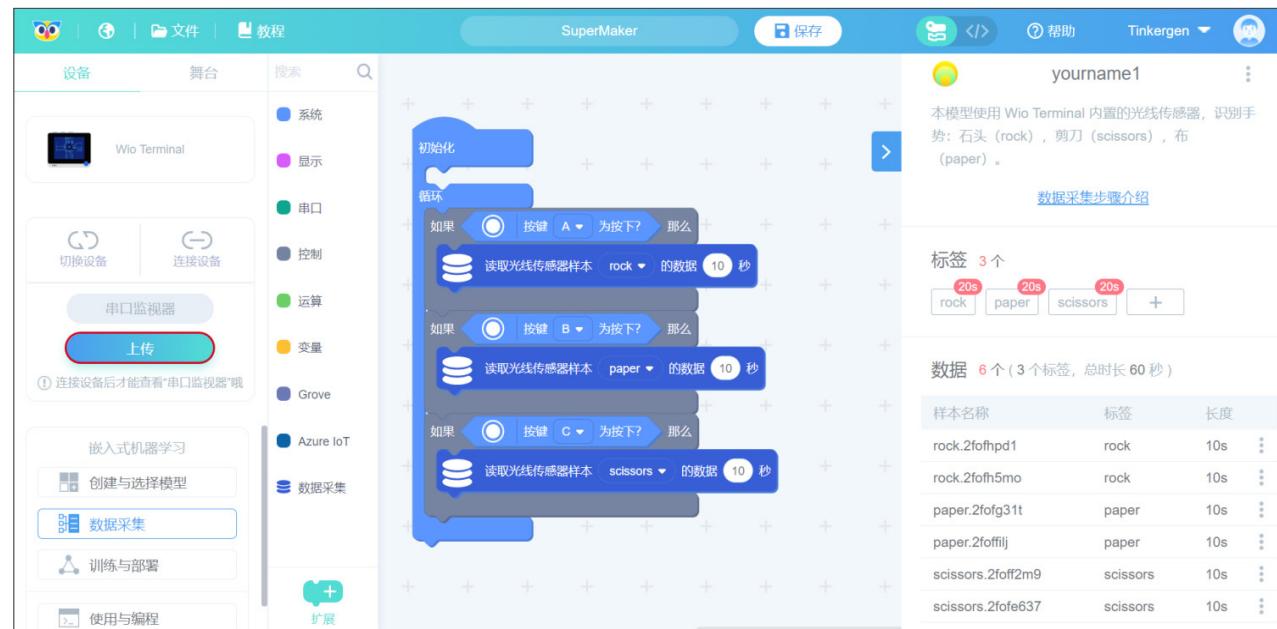
rock paper scissors +

数据 0 个 (0 个标签, 总时长 0 秒)

样本名称 标签 长度

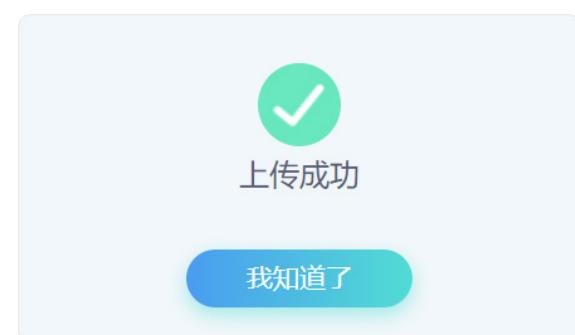
2.2 连接设备并上传 Codecraft 中的默认数据采集程序

Wio Terminal 连接后，在 Codecraft 界面，点击 即可上传默认数据采集程序。



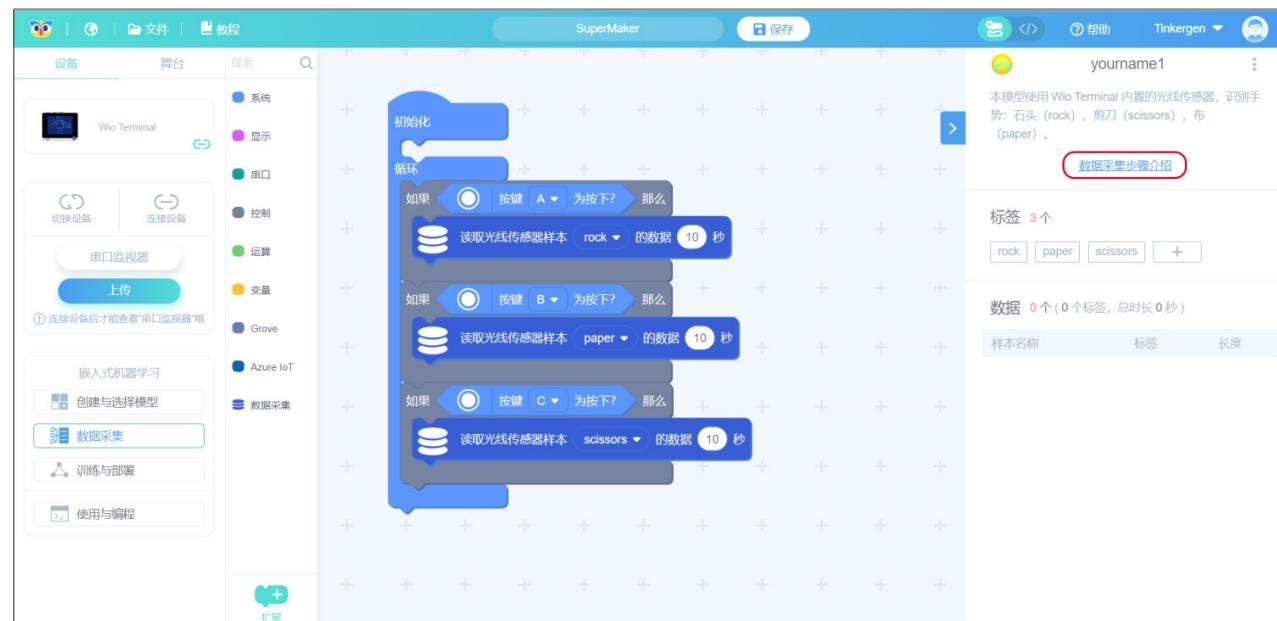
通常上传需要 10 秒钟。程序上传后，“上传成功”的窗口将出现在如下图所示的屏幕上。

点击“我知道了”关闭上传成功弹窗，返回界面。



2.3 数据采集

在右上角的超链接中，您将找到数据采集的分步介绍，请按照说明收集数据。



注意：

- Wio Terminal 按钮位置。
- 动画是已经加速过的，实际动作可能有所缓慢。

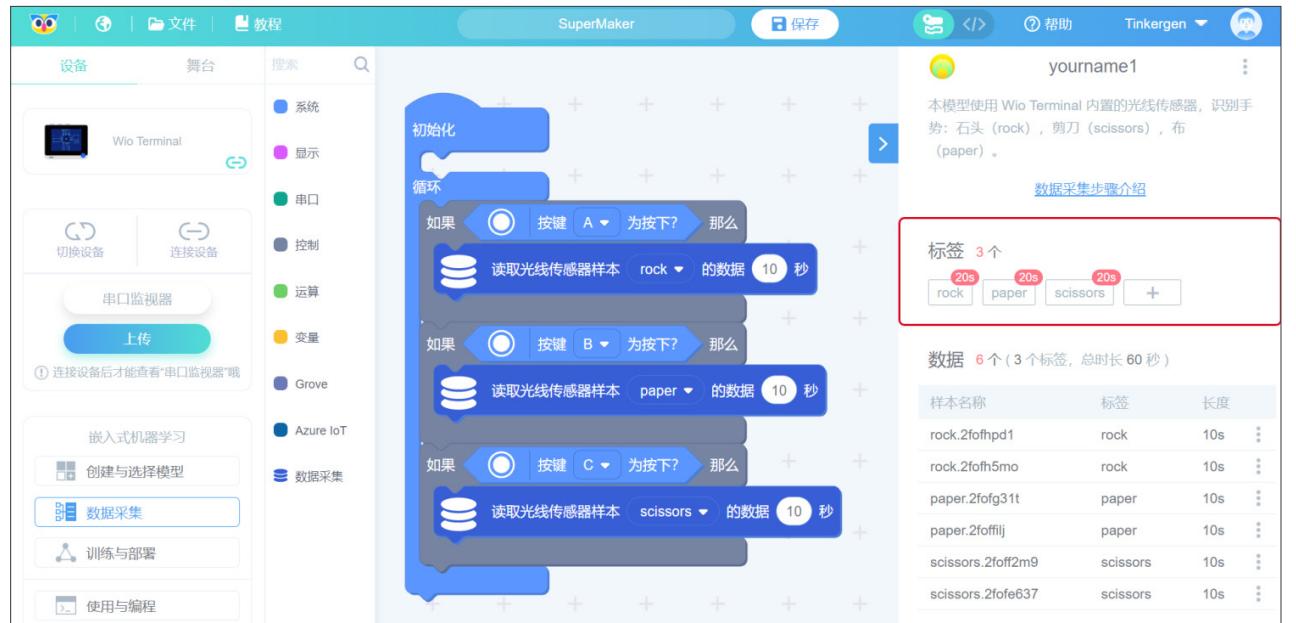
- 请注意红色提示。
- 将光标指向描述文本以获得更详细的内容。



在数据收集过程中，Wio Terminal 将会显示图标来表示当前进度，我们可以根据 Wio Terminal 显示图标来完成数据采集。

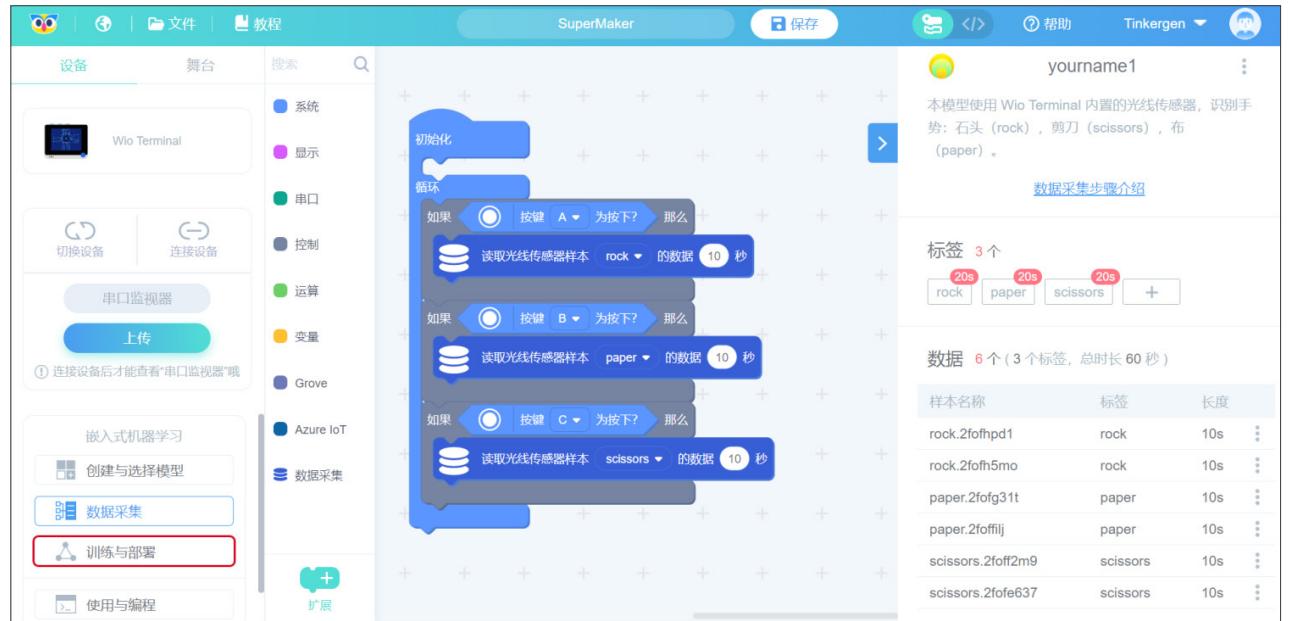
 这个信号表示正在 Wio Terminal 收集数据。

 OK 表示数据收集完成。



至此，数据全部采集完成。

单击“训练与部署”。



步骤 3. 训练与部署

3.1 设置神经网络和参数

选择你觉得合适的神经网络规模：小型、中型、大型。

接着设置参数、训练周期数（正整数）、学习率（从 0 到 1 的数字）和最小置信度（从 0 到 1 的数字）。初始页面已提供一些默认参数值。

在这里我们使用选择的规模为中，这会耗一段时间。



3.2 开始训练模型

点击“开始训练”。



当您点击“开始训练”时，界面会显示“拼命加载中 ...”。



“拼命加载中...”的持续时间取决于所选神经网络的规模（小型、中型、大型）和训练周期数。通常来说，网络规模和训练周期数越大，所需的时间就越长。

您还可以通过观察“日志”来推断等待时间。下图中，“Epoch: 78/500”表示训练总轮数为 500 轮且已经训练了 68 轮。



3.3 观察模型性能，选择理想模型

在“模型训练报告”窗口中，您可以观察训练结果，包括模型的准确率、损失和性能。

如果训练结果不理想，您随时可以回到第一步训练模型，选择另一个大小的神经网络，调整参数训练等，直到得到一个结果满意的模型。



3.4 部署理想模型

在“模型训练报告”窗口中，点击“模型部署”。



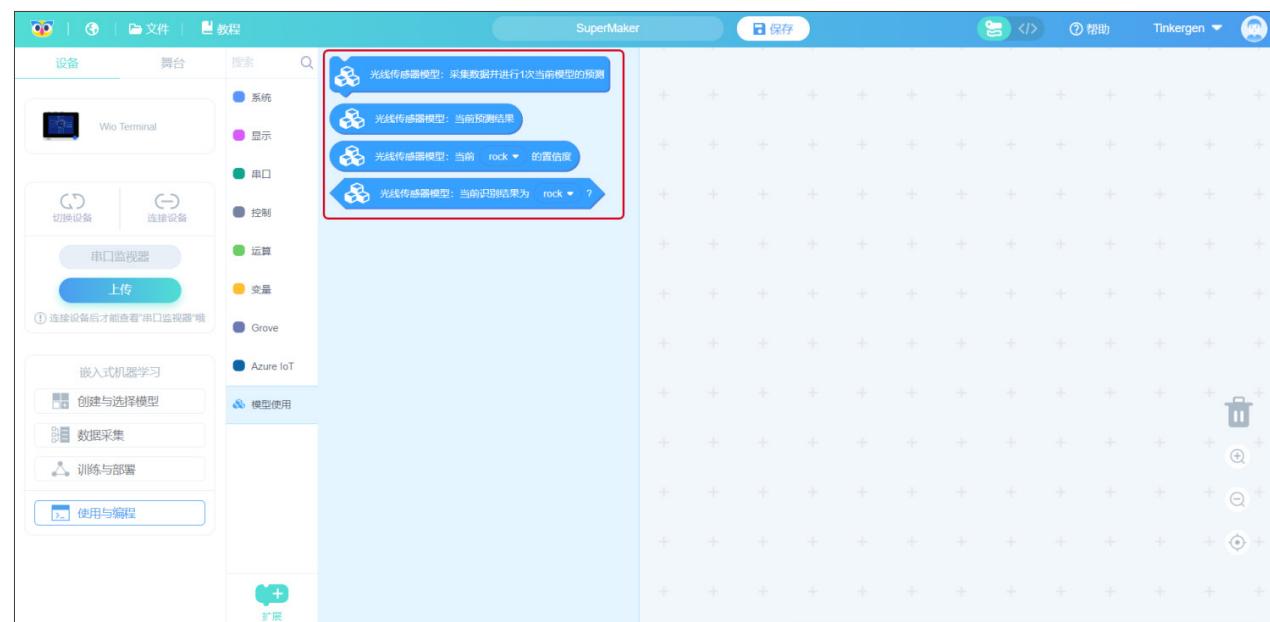
等到部署完成后，点击“确认”即可跳转到“使用与编程”窗口。



步骤 4. 使用与编程

4.1 编写使用模型的程序

在“编程”界面，点击“模型使用”以使用部署的模型。



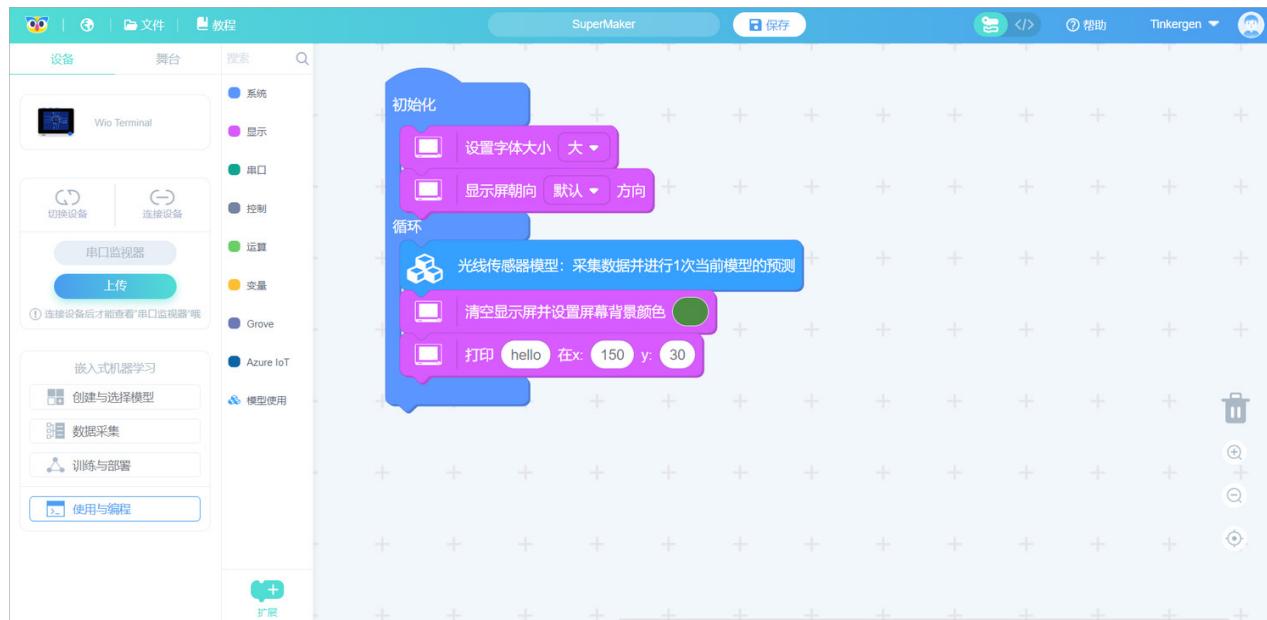
在“模型训练报告”窗口中，点击“模型部署”。

您可以尝试通过编写以下程序来使用您的模型。

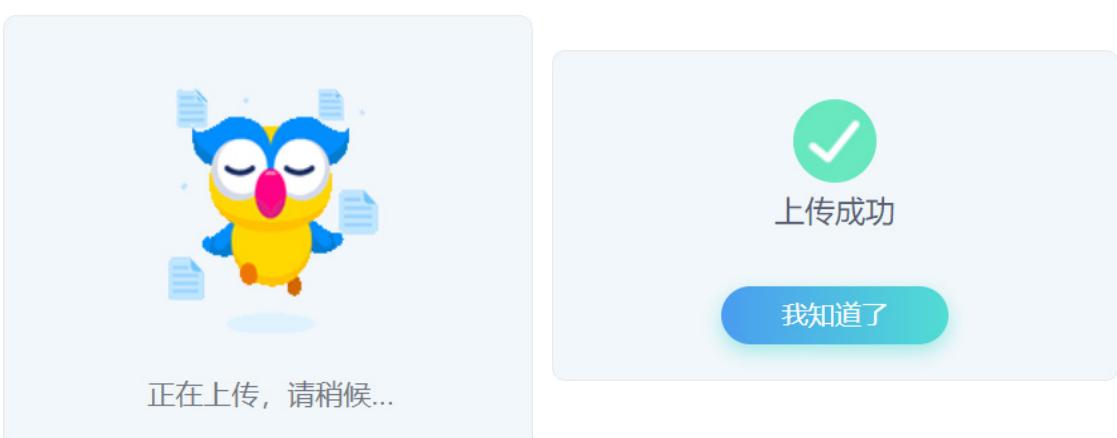


4.2 将程序上传到 Wio Terminal

单击“上传”按钮。



第一次上传时间通常比较长，并且随着模型的复杂性时间会进一步增加。较小型号的上传时间约为 4 分钟，较大型号可能会消耗更长时间，这取决于您机器的性能。



4.3 Wio Terminal 测试模型

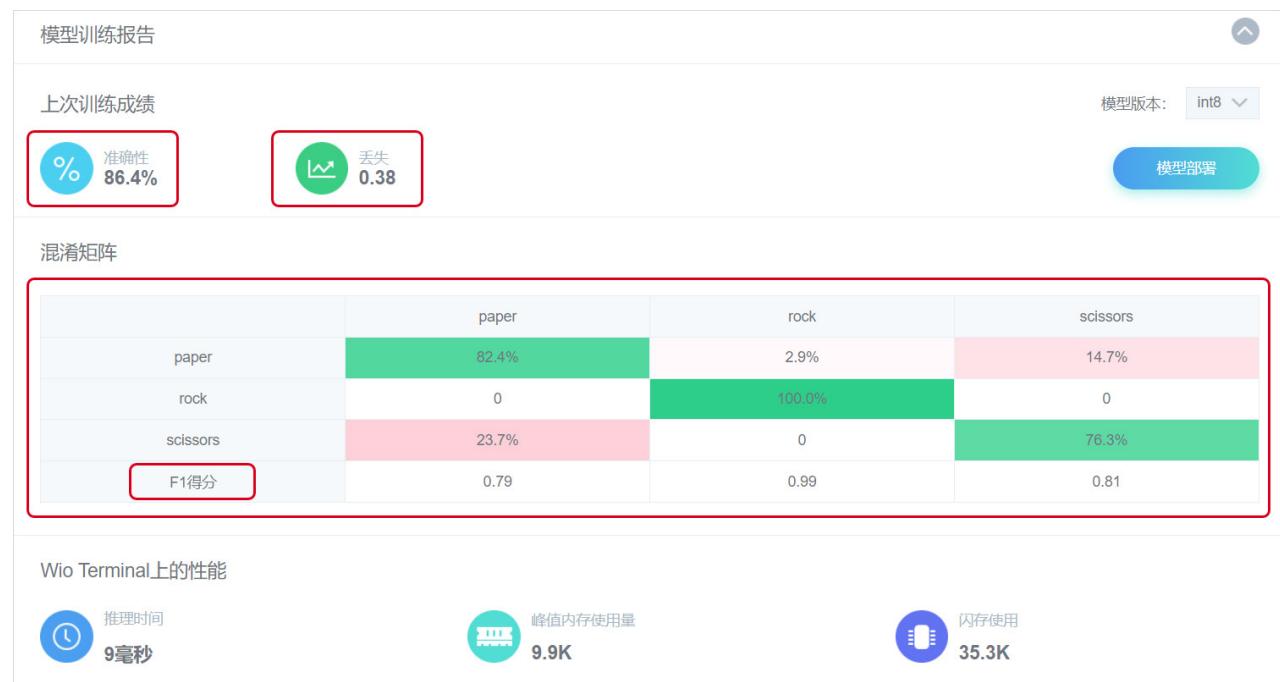
做一个“剪刀”的手势或者其他手势，看 Wio Terminal 能否识别。

恭喜！您已经完成了第二个嵌入式机器学习模型，相信您已经熟悉了输入（数据集和标签）并且对输出有了一定的了解。较大的“准确性”分数可以表示您的模型运行良好，但这并不足以评估模型的性能。那么现在让我们来看看输出吧，熟悉输出可以帮助你更好的理解和判断模型的性能。

机器学习理论（输出、训练表现）：

输出（训练表现）：

- 准确率
- 损失函数
- 混淆矩阵
- F1 分数



1. 准确率

准确率是衡量分类模型性能的一种方法，它通常用百分比表示，也就是预测值等于真实值的所占总预测值的比例。模型性能与整体的准确率相关联，而通常在训练阶段准确率就会不断变化。与损失相比，准确率更容易理解。

2. 损失函数

损失函数也称为成本函数或者代价函数，根据预测值与真实值的差异程度来确定预测的概率或不确定性。这能让我们可以更细致地了解模型的性能。

与准确率不同，损失不是百分比。它是训练集或验证集中每个样本出现的错误的总和。在训练过程中经常使用损失来寻找模型的“最佳”参数值（例如神经网络中的权重）。在训练过程中，这个最佳的参数值应当是最小的值。

* 准确率与损失函数的关系

大多数情况下，我们能观察到准确率会随着损失函数的减少而增加，但情况并非总是如此：准确度和损失函数有着不同的定义，它们测量的是不同的参数，虽然它们通常看起来成反比，但这两个指标之间是没有数学关系的。

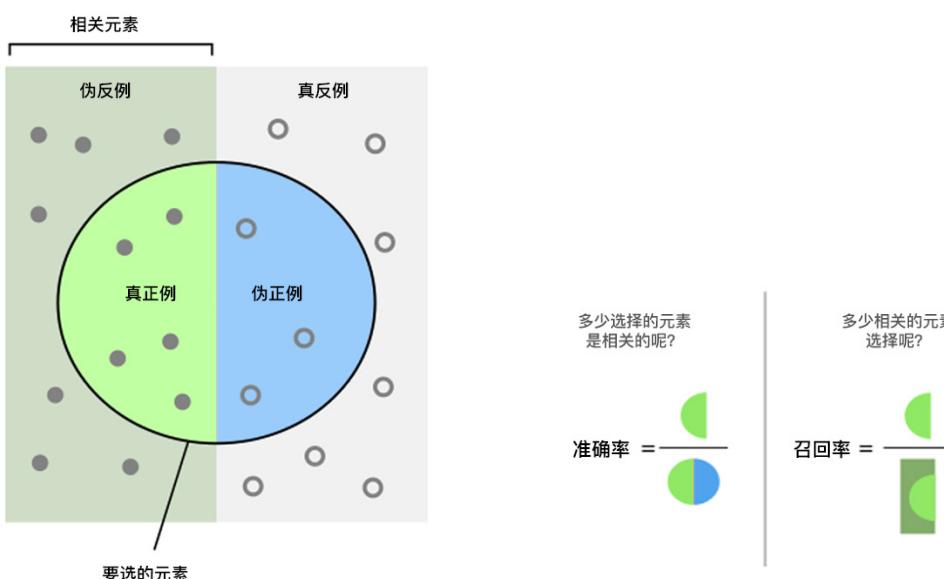
3. 混淆矩阵

- 混淆矩阵可以展示出一个学习算法的性能。
- 它是由一个简单的方阵组成，各个方阵为：
 - 真正例 (True Positive)
 - 真反例 (True Negative)
 - 伪正例 (False Postive)
 - 伪反例 (False Negative)

		预测结果	
		正例	假例
真正结果	正例	真正例 热狗	伪反例 不是热狗
	假例	伪正例 热狗	真反例 不是热狗

4. F1 分数

F1 分数是衡量模型在数据集上的准确性的指标，它与准确率和召回率有关。首先我们先认识一下准确率和召回率：



而 F1 分数与准确率和召回率相关的计算公式为：

$$F_1 = 2 \cdot \frac{\text{准确率} \cdot \text{召回率}}{\text{准确率} + \text{召回率}}$$

理想情况下，准确率和召回率都为 1，因此 F1 为 1 ——这可以表示机器学习算法非常完美，可以使机器模型完美分类，而一般来说 F1 分数是介于 0 和 1 之间的。

★ 总结

1. 背景理论知识：

- 光线传感器
- 2. 嵌入式机器学习实践



3. 机器学习理论（输出 / 训练表现）

- 准确度：一种衡量模型性能的方法。
- 损失：损失函数，也称为成本函数，根据预测与真实值的差异程度来判断预测的概率或不确定性。
- 混淆矩阵：展示学习算法性能的矩阵。
- F1 分数：

$$F_1 = 2 \cdot \frac{\text{准确率} \cdot \text{召回率}}{\text{准确率} + \text{召回率}}$$

第4课

使用内置麦克风识别唤醒词



项目概述

我们对于 iPhone 的“Hi, Siri”可能很熟悉，但其中的原理，手机是如何听取声音并实现反馈我们是否了解呢？在本课中，我们将试着用机器学习模型制作自己的“Siri”——“Wio”，并用“Hi, Wio”进行将其进行唤起。



我们将使用 Wio Terminal 内置麦克风来收集唤醒单词的语音以及说话环境的声音来训练模型。在本课结束后，不仅仅只是唤醒单词的语音识别，我们甚至可以建立机器学习模型来对其他类型的音频进行分类，例如“哭泣的婴儿”，“咳嗽”，“狗吠叫”等。

预期结果

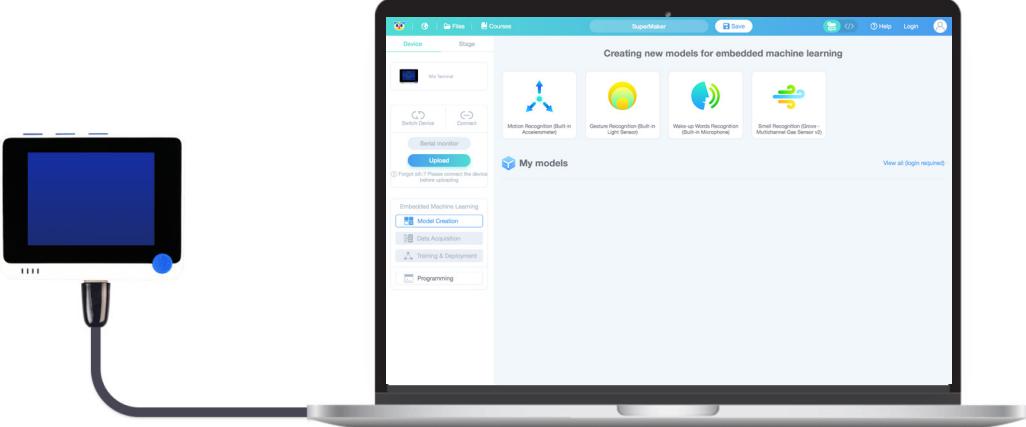
当 Wio Terminal 内置麦克风检测到“Hi, Wio”的声音时，Wio Terminal 可以做到实时响应。



准备工作

硬件需求：Wio Terminal

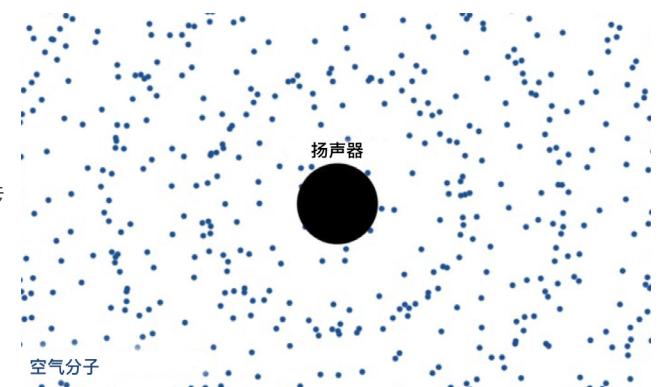
连接方法：



背景知识

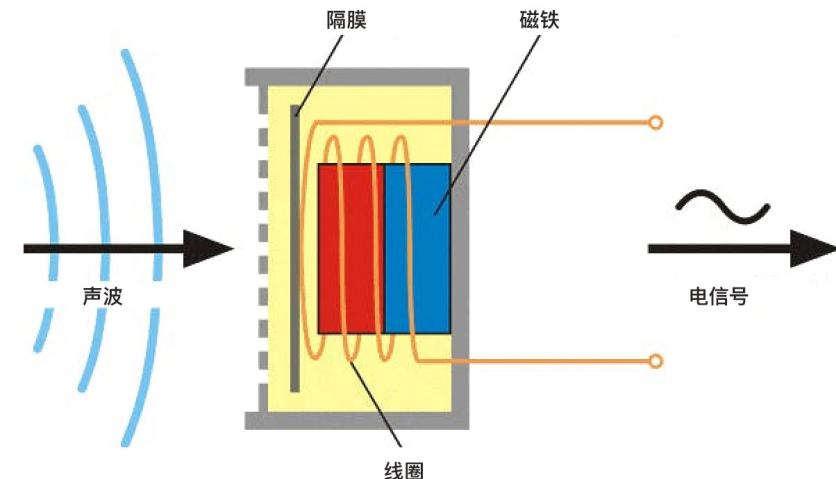
让我们更多地了解计算机如何处理声音：

首先，声音是一种震动，以声波的形式通过传输介质传播，介质包括气体，液体和固体。

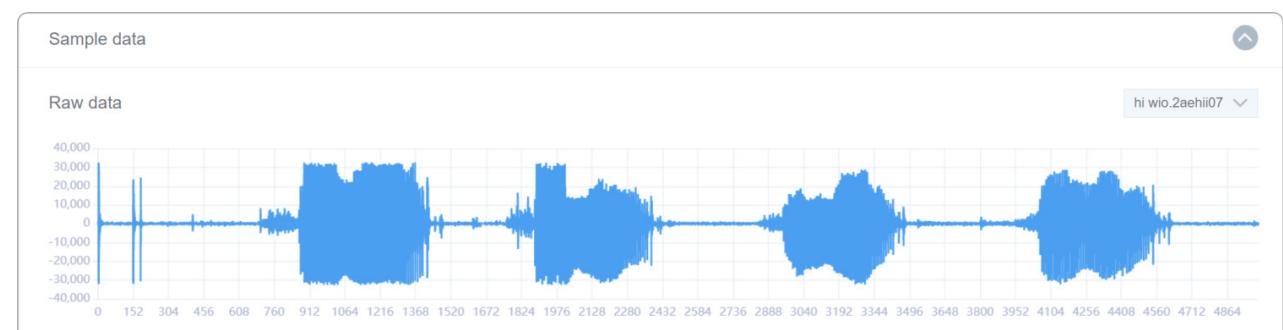


因为声音是一种震动，在声源处它会推动周围的介质分子，周围的介质分子又会推动它周围的介质分子，依此类推，到达其他分子后，它们都会略微振动。

我们在麦克风中使用的就是这个原理，麦克风膜会被介质分子向内推动而又回到原始位置，这期间会在电路中产生电压，其中电压与声音的幅度成正比。声音越响亮，推动膜的次数越多，输出端产生的电压就越高。



最后，我们用模 / 数转换器来读取这个电压，并以相等的间隔进行记录。

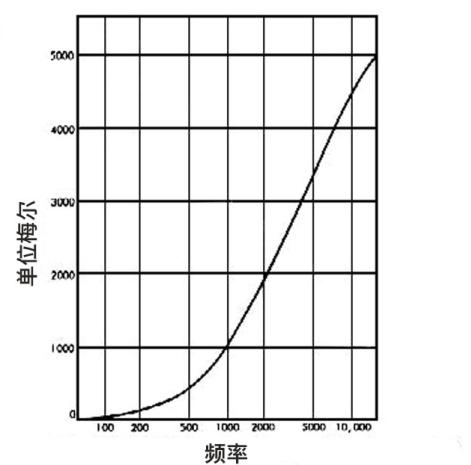


可是我们无法对这种原始记录的声音做太多事情，因为其包含了除我们想要的声音外，还有许多需要剔除的杂音，我们可以将其剪切成好几个部分使其更安静或更响亮以供机器能分辨。那么如何分辨呢？

研究表明，人类更擅长检测较低频率之间的差异而不是较高频率之间的。例如，我们可以很容易地分辨出 500 Hz 和 1000 Hz 之间的差异，但我们几乎无法分辨出 10000 Hz 和 10500 Hz 之间的差异。即使这两对声音频率之间的距离相同。

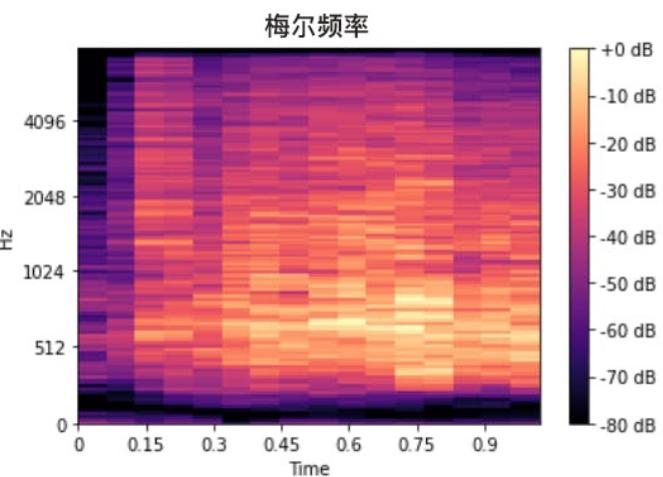
1937 年，史蒂文斯 – 沃尔克曼 (Stevens-Volkmann) 和纽曼 (Newmann) 提出了一个音高单位，使得声音的间距的音高单位能与听众的距离相等。这被称为 Mel 频谱图，同时他们提出梅尔倒谱系数特

征 (Mel Frequency Cepstral Co-efficients)。



梅尔倒谱系数特征 (MFCC) 是一种易于使用的内部音频表示格式，这类似于图像的 JPG 格式——那么我们就可以像处理图像一样来处理音频并教机器去如何分辨。

了解原理后，现在是时候开始实施我们的项目了。



练习与实践

课程步骤

1. 创建与选择模型
2. 数据采集
3. 训练与部署
4. 使用与编程

<https://ide.tinkergen.com/>

步骤 1、创建与选择模型

1.1 建立“唤醒词识别（内置麦克风）”模块

点击“创建与选择模型”，然后点击“内置麦克风识别唤醒词”，如下面步骤 1 和步骤 2 所示：



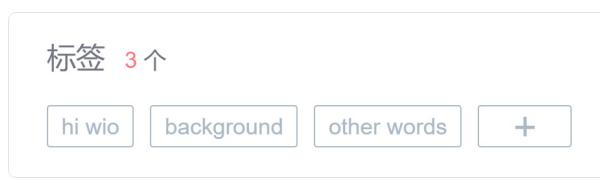
给新建的模型命名。



点击“确认”就会自动跳转到数据采集界面。

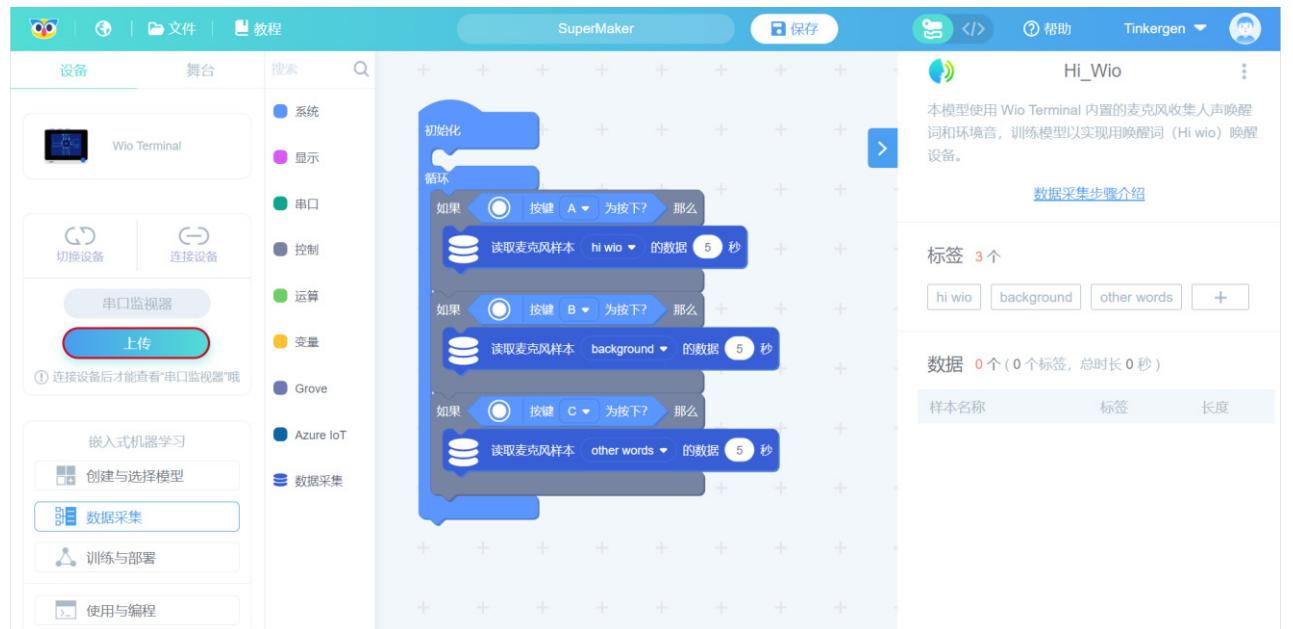
步骤 2、数据采集

2.1 默认标签



2.2 连接设备并上传 Codecraft 中的默认数据采集程序

Wio Terminal 连接后，在 Codecraft 界面，点击 即可上传默认数据采集程序。



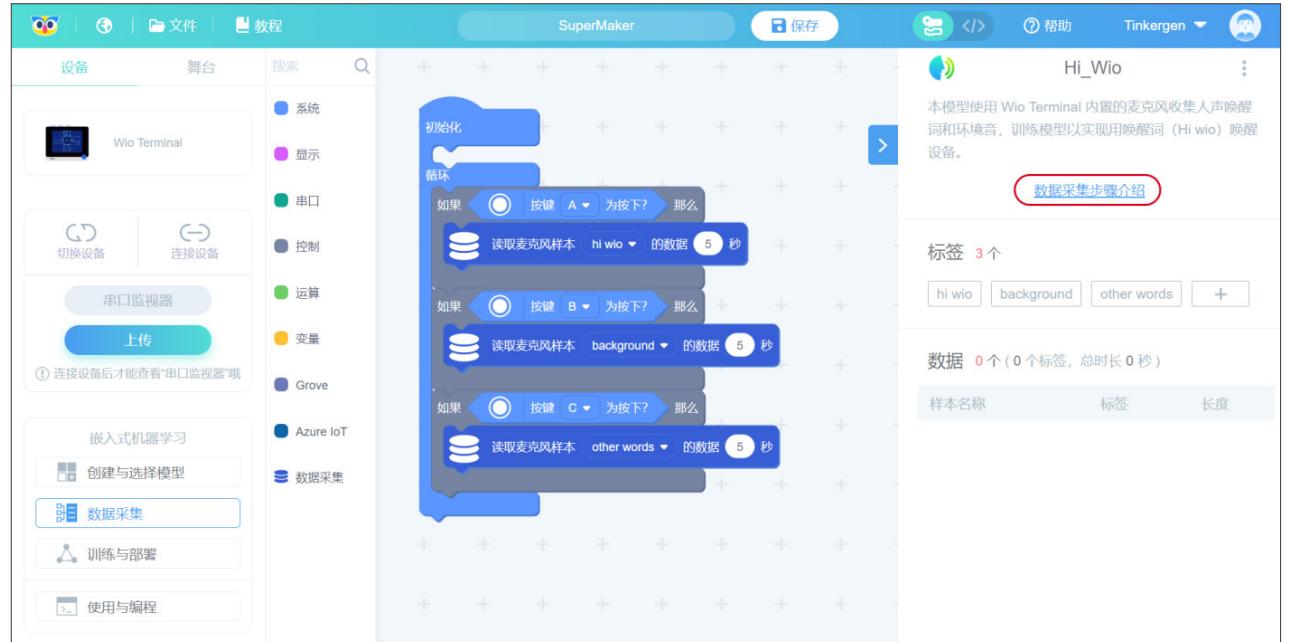
通常上传需要 10 秒钟。程序上传后，“上传成功”的窗口将出现在如下图所示的屏幕上。



点击“我知道了”关闭上传成功弹窗，返回界面。

2.3 数据采集

在右上角的超链接中，您将找到数据采集的分步介绍。请按照说明收集数据。



注意：

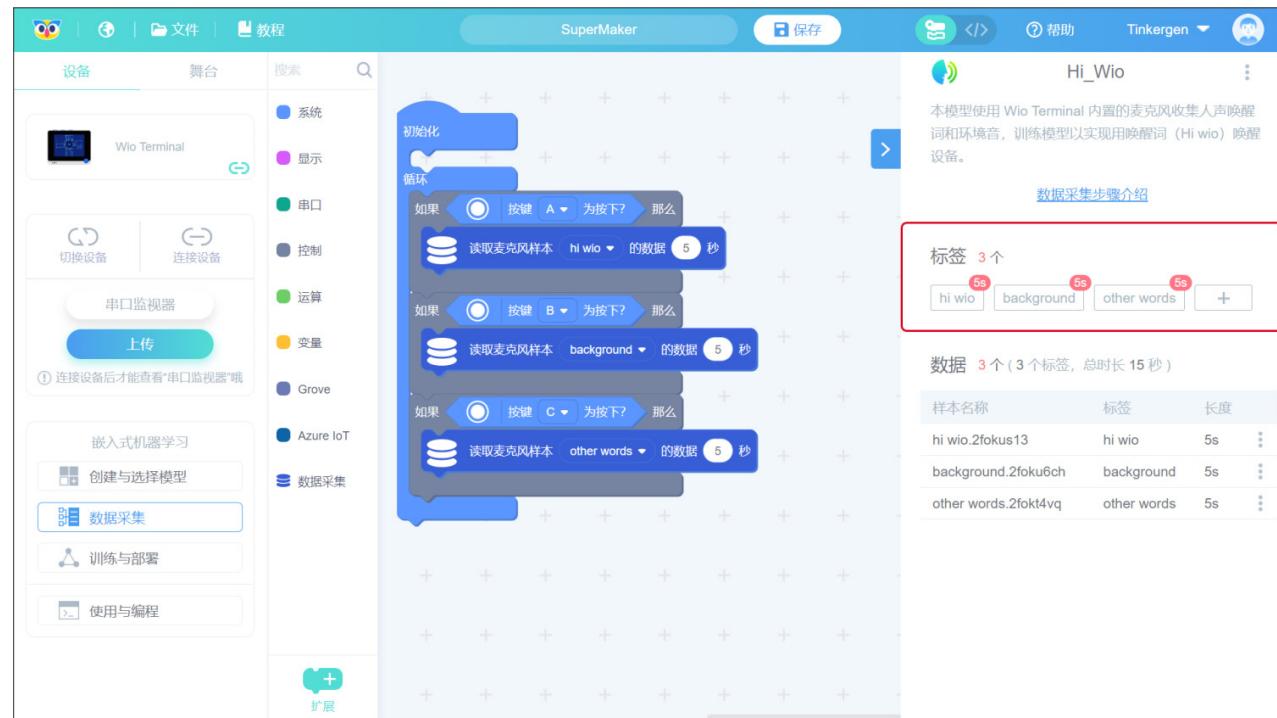
- Wio Terminal 按钮位置。
- 动画是已经加速过的，实际动作可能有所缓慢。
- 请注意红色提示。
- 将光标指向描述文本以获得更详细的内容。



在数据收集过程中，Wio Terminal 将会显示图标来表示当前进度，我们可以根据 Wio Terminal 显示图标来完成数据采集。

 这个信号表示正在 Wio Terminal 收集数据。

 OK 表示数据收集完成。



至此，数据全部采集完成。

单击“训练与部署”。



步骤 3. 训练与部署

3.1 设置神经网络和参数

选择你觉得合适的神经网络规模：小型、中型、大型。

接着设置参数、训练周期数（正整数）、学习率（从 0 到 1 的数字）和最小置信度（从 0 到 1 的数字）。初始页面已提供一些默认参数值。

在这里我们使用选择的规模为中，这会耗一段时间。



3.2 开始训练模型

点击“开始训练”。

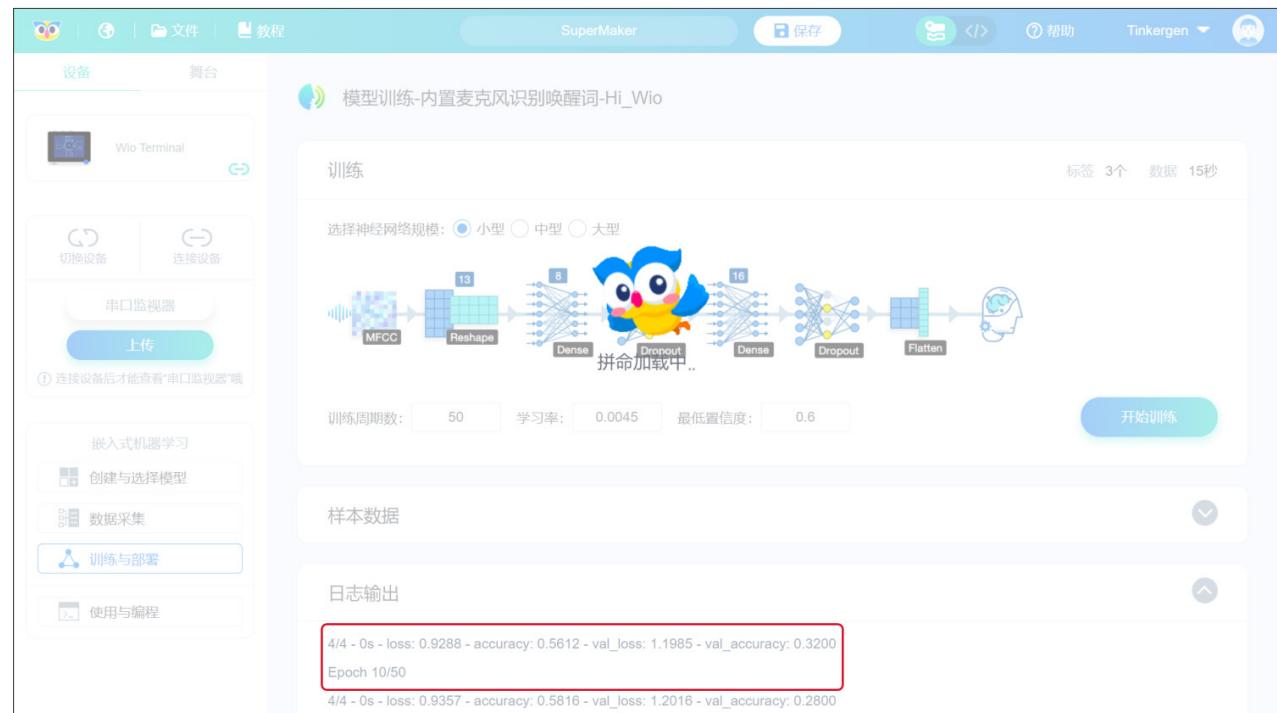


当您点击“开始训练”时，界面会显示“拼命加载中...”。



“拼命加载中...”的持续时间取决于所选神经网络的规模（小型、中型、大型）和训练周期数。通常来说，网络规模和训练周期数越大，所需的时间就越长。

您还可以通过观察“日志”来推断等待时间。下图中，“Epoch: 10/50”表示训练总轮数为 50 轮且已经训练了 10 轮。



3.3 观察模型性能，选择理想模型

在“模型训练报告”窗口中，您可以观察训练结果，包括模型的准确率、损失和性能。

如果训练结果不理想，您随时可以回到第一步训练模型，选择另一个大小的神经网络，调整参数训练等，直到得到一个结果满意的模型。



3.4 部署理想模型

在“模型训练报告”窗口中，点击“模型部署”。



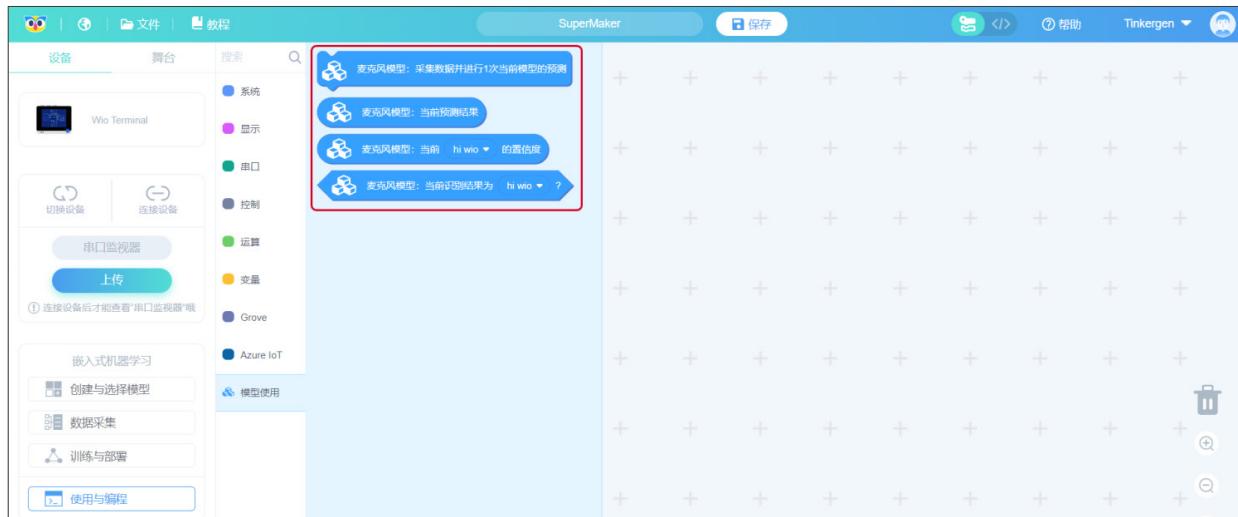
等到部署完成后，点击“确认”即可跳转到“使用与编程”窗口。



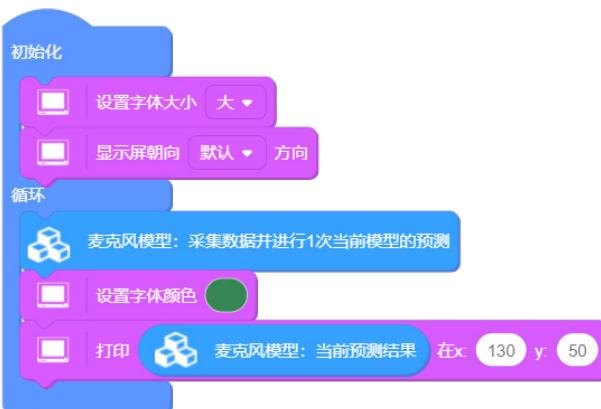
步骤 4. 使用与编程

4.1 编写使用模型的程序

在“使用与编程”界面，点击“模型使用”以使用部署的模型。

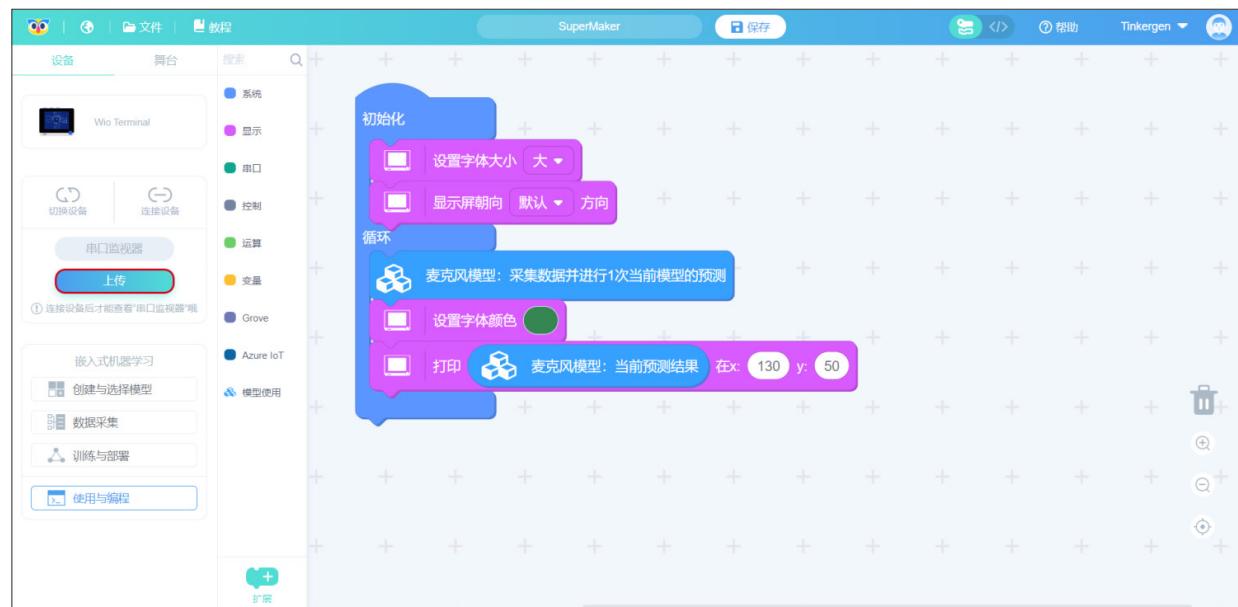


您可以尝试通过编写以下程序来使用您的模型。

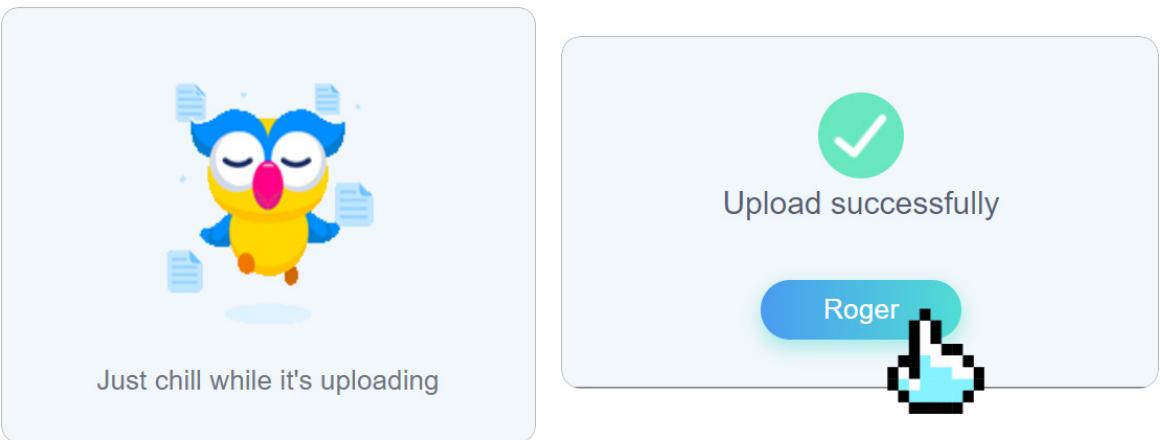


4.2 将程序上传到 Wio Terminal

单击“上传”按钮。



第一次上传时间通常比较长，并且随着模型的复杂性时间会进一步增加。较小型号的上传时间约为 4 分钟，较大型号可能会消耗更长时间，这取决于您机器的性能。



4.3 Wio Terminal 测试模型

尝试喊一下“Hi, Wio”，看能否唤醒机器。

祝贺您完成了第三个 TinyML 模型！相信您已经尝试了不同比例的模型，以便获得更好的模型性能。现在让我们对模型内部进行深究，深刻理解“差异”。

机器学习理论（不同的比例模型）：

- 人工神经网络结构
 - a. 节点
 - b. 层
 - c. 人工神经网络结构 (ANN)
- 不同尺度神经网络
 - a. 规模大小
 - b. 神经网络宽度
 - c. 神经网络深度
 - d. 模型容量
 - e. 模型架构



1. 人工神经网络结构 (ANN)

1.1 什么是节点

神经元节点是网络的处理数据的单元。每个神经元对不同的输入进行加权和求和，并通过激活函数传递它们，激活函数的作用是在数据送到下一层之前对其进行缓冲。

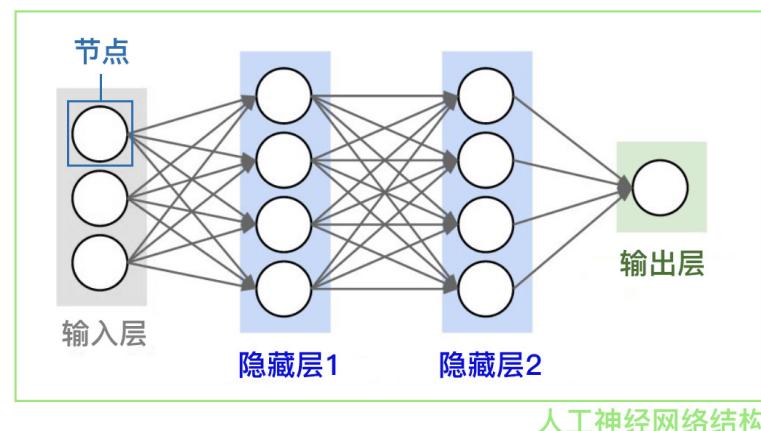
1.2 什么是层 (layer)

在深度学习模型中，层是模型的体系中的一种结构或者网络拓扑，它从上一层获取信息，然后将信息传递到下一层。我们可以认为一层是由一个一个节点构成。

1.3 人工神经网络结构 (ANN) 是什么？

人工神经网络由输入层、输出层和隐藏层三层组成。层的各个类型总结如下：

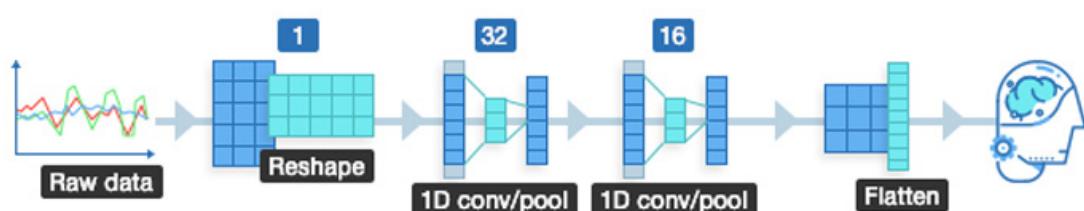
- **输入层**: 输入层用来输入变量，有时称为可见层。
- **输出层**: 产生输出变量的节点层。
- **隐藏层**: 输入层和输出层之间的节点层。这些层中可能有一个也可能有多个。隐藏层从输入层接收输入的信息并进行处理，得到的值就被传输到输出层。



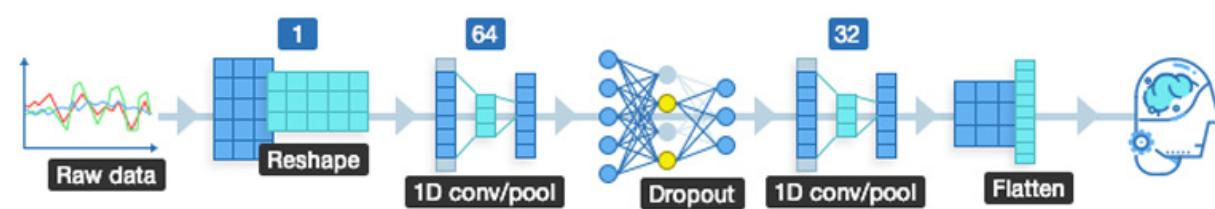
每个输入层的节点与隐藏层的节点，每个隐藏层节点与输出层的节点之间，一定会存在连接。

2. 不同规格的神经网络

SMALL



MEDIUM

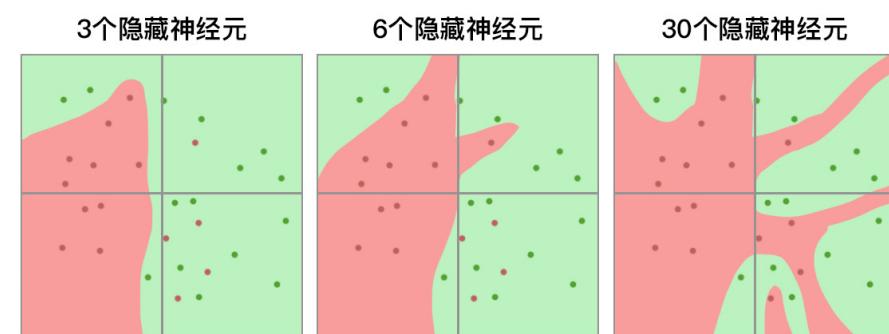


2.1 规模大小：整个模型中的节点数。

2.2 神经网络宽度：在特定层中的节点数。



下图告诉我们“宽度”在模型中起着重要作用。



3个隐藏神经元

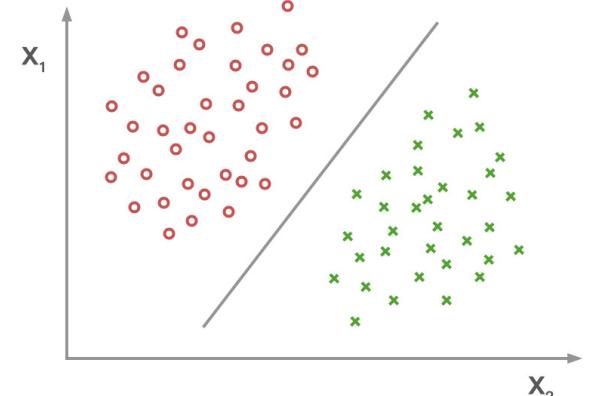
6个隐藏神经元

30个隐藏神经元

2.3 神经网络深度：神经网络中的层数。

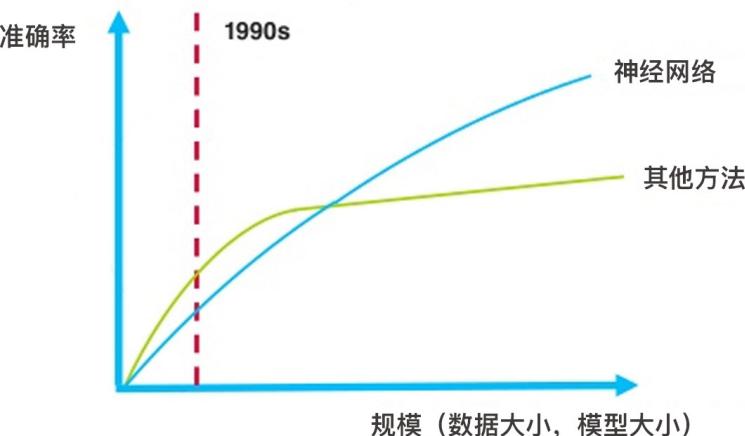
2.3.1 为什么要有多个层？

单层神经网络只能用来表示线性可分函数。这意味着一个分类问题中的两个类就只用一条线分开。



当然如果您的问题相对简单，也许单层神经网络就足够了，但很显然我们感兴趣的大多数问题都不是线性可分的，对于很多问题，我们可以使用多层感知器 (Multilayer Perceptron)。

更多的数据 + 更大的模型

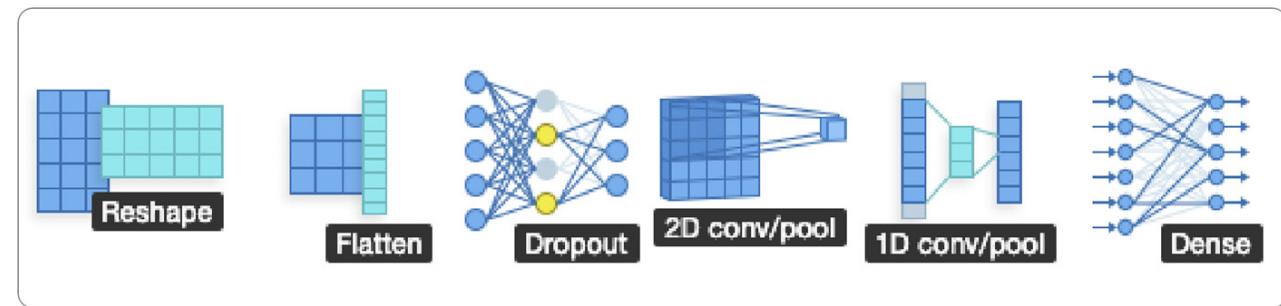


多层感知器 (MLP) 可以用来表示凸区域。这意味着他们可以学习在一些高维空间中围绕输入的数据来绘制形状，正如上图所示。它通过克服线性可分性的限制，利用这些绘制空间对输入的数据进行分离和分类。

* 事实上，利普曼在 1987 年发表的论文《神经网络计算导论》中，有一个理论表明具有两个隐藏层的 MLP 足以创建任何所需形状的分类区域。尽管应该注意的是，这里并没有指示每个层中要使用多少节点或如何学习权重，但毋庸置疑这是有指导意义的。进一步的理论发现和证明表明，MLP 是通用的近似器。这意味着每使用一个隐藏层，MLP 都可以近似我们需要的任何函数。

2.4 模型容量：可通过网络配置学习的功能类型或结构，有时被称为“表示容量 (representational capacity) ”。

2.5 模型架构：网络中各层和节点的具体排列。不同类型的层还有：Reshape 层，Flatten 层，Dropout 层，Conv2D/pool 层，Conv1D/pool 层，Dense 层等等。



* 为什么要有不同类型的层？

不同的层对其输入执行的转换是不同的，某些层比其他层更适合于某些特定任务。

例如，卷积层通常用于处理图像数据的模型，而递归层用于处理时间序列数据的模型。完全连接的层将其层内的每个输入连接到每个输出。

(第 8 课中会有更多关于层的细节。)

总之，小型、中型和大型模型都是有不同数量的隐藏层和神经元，还有许多不同类型的层。

★ 总结

1. 背景知识
 - 麦克风
2. 嵌入式机器学习实践



3. ML 理论 (模型规模)

- 人工神经网络结构
 - a. 节点：神经元节点是网络的处理单元。
 - b. 层：在深度学习模型中，层是模型架构中的一种结构或网络拓扑，它从上一层获取信息，然后将信息传递到下一层。
 - c. 人工神经网络 (ANN)：ANN 由三层组成，即输入层、输出层和隐藏层。
- 不同尺度神经网络
 - a. 大小：模型中的节点数。
 - b. 宽度：特定层中的节点数。
 - c. 深度：神经网络中的层数。
- d. 容量：可通过网络配置学习的功能类型或结构。
- e. 架构：网络中各层和节点的具体安排。

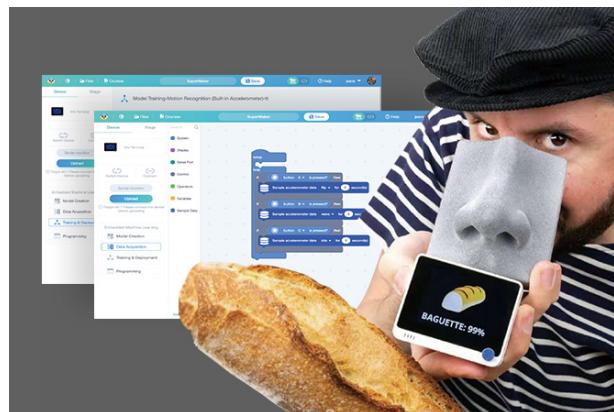
第 5 课

使用 Grove 多通道气体传感器进行
气味识别



项目概述

在这个项目中，我们将制造一个人工智能鼻子，它是由机器学习模块构成，可以被训练用来区分可乐中的酒精，或者识别你想叫它闻到的任何东西。我们将使用 Wio Terminal 的外接 Grove 多通道气体传感器（Grove – Multichannel Gas Sensor v2）来收集不同的气味数据，并将其用于训练模型以区分可乐、酒精和空气。



这受到了 [Benjamin Cabé 人工鼻子项目的启发](#)：

“我花了相当多的时间来试图完善我的面包配方，包括确定什么时候我的酵母发酵剂会处于理想状态，可以让我烤出完美的法棍。”凯布如是说。“几周后，我组装了一个完美的假鼻子。”

所以在项目完成后，你也可以尝试组装自己的一个假鼻子来检测其他气味。

预期结果

期望的结果显示如下，Wio Terminal 能实时检测到当前的酒精或可乐并显示：

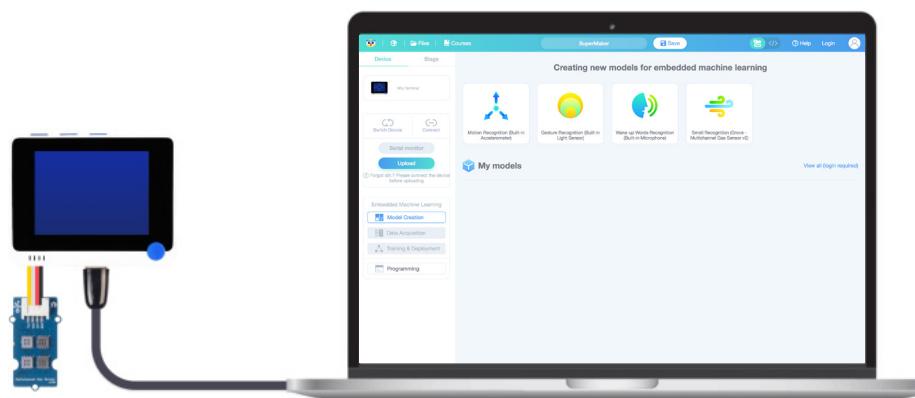


准备工作

硬件需求：

- Wio Terminal
- [Grove 多通道气体传感器](#)

连接方法：



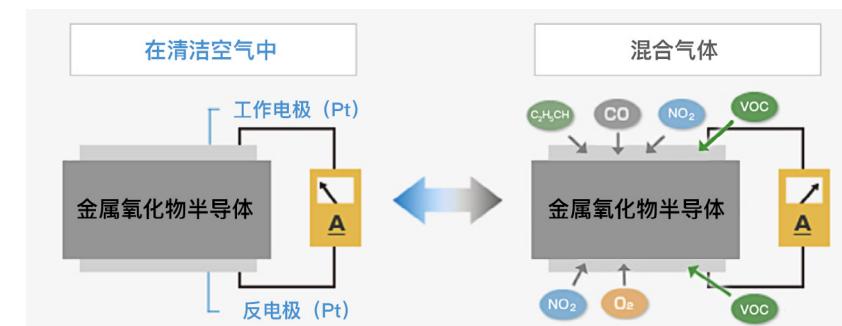
· 注意 ·

在使用之前我们需要对传感器进行预热，以达到内部化学平衡，这样才能使我们更好的用传感器识别气体。建议传感器存储时间和相应的预热时间如下：

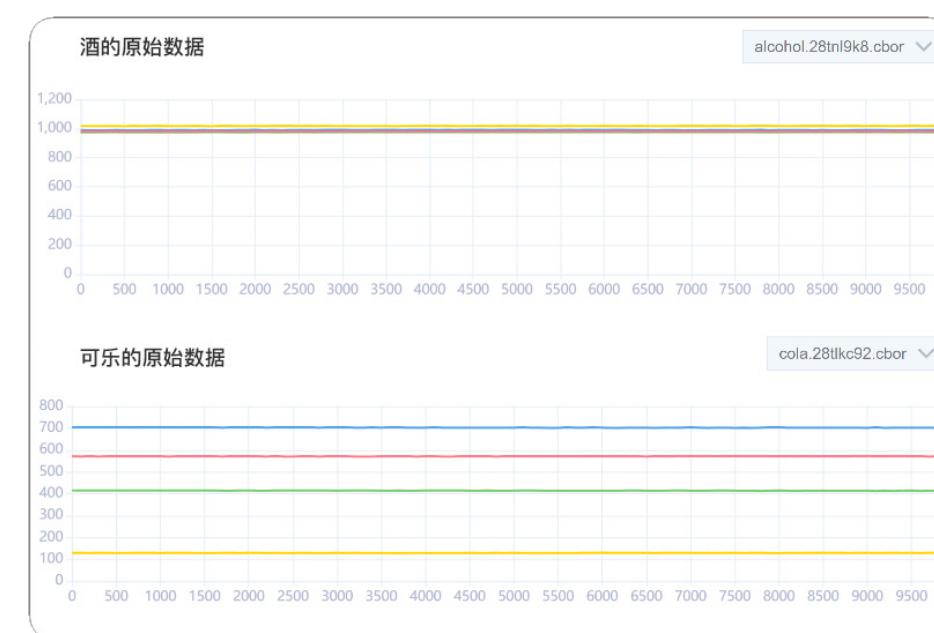
传感器存储时间	建议的预热时间
少于1个月	不少于24小时
1-6个月	不少于48小时
超过6个月	不少于72小时

背景知识

气体传感器使用 MEMS (Micro-Electro-Mechanical Systems) 工艺，也就是在硅底板上嵌入在清洁空气中导电率低的半导体材料（金属氧化物），以此来制造微热板，所以当传感器暴露在有特定气体气味的环境中，传感器的电导率会随着空气中的气体浓度升高而升高。这种电导率的变化可以通过一个简单的电路转换来把对应的气体浓度转换为电信号输出。



Grove 多通道气体传感器有 4 个测量单元，每个测量单元对各种气体都很敏感，这意味着我们可以同时获得四组数据，因为识别的气体里有着不同种类的气体，我们就可以通过这四组数据来判断。传感器可检测一氧化碳 (CO)、二氧化氮 (NO₂)、乙醇 (C₂H₅OH)、挥发性有机化合物 (VOC) 等多种气体。



如上图所示的酒和可乐，气体传感器在检测到其气体的含量和组合的值时是具有一定特征的，而这些特征值就可以让我们用来建立一个机器学习模型以识别不同物质排放的气体。

练习与实践

课程步骤

1. 创建与选择模型
2. 数据采集
3. 训练与部署
4. 使用与编程



步骤 1、创建与选择模型

1.1 建立“气体识别（Grove 多通道气体传感器）”模块

点击“创建与选择模型”，然后点击“外接多通道气体传感器识别味道”，如下面步骤 1 和步骤 2 所示：

The screenshot shows the 'Create New Model' screen in the Codecraft IDE. On the left, there's a sidebar with categories: '设备' (Devices), '舞台' (Stage), '串口监视器' (Serial Monitor), '上传' (Upload), and '我的模型' (My Models). Under '我的模型', there are three cards: '内置麦克风识别换醒词' (Built-in microphone recognition wake word) with icon, '内置光线传感器识别手势' (Built-in light sensor recognition gesture) with icon, and '外接多通道气体传感器识别味道' (External multi-channel gas sensor recognize flavor) with icon. A red circle labeled '2' is over the third icon. On the right, there's a large area for creating a new model, with a placeholder text '为嵌入式机器学习创建新模型' (Create a new model for embedded machine learning).

给新建的模型命名。点击“确认”就会自动跳转到数据采集界面。

The dialog box has a title '新模型的名称' (New Model Name). It contains a note: '基于：外接的 Grove 多通道气体传感器识别不同的气味' (Based on: External Grove multi-channel gas sensor identifies different odors). Below is a text input field with placeholder '请输入新模型的名称' (Enter new model name) and a note '请输入英文开头的1-30位英文、数字或者下划线组成的字符串' (Enter a string of 1-30 English letters, digits, or underscores starting with English). At the bottom are two buttons: '取消' (Cancel) and '确认' (Confirm).

步骤 2、数据采集

2.1 默认标签

The screenshot shows the 'Data Collection' steps in the Codecraft IDE. The steps are: '初始化' (Initialization), '如果 按键 A 为按下? 那么 读取多通道气体传感器样本 cola 的数据 10 秒', '如果 按键 B 为按下? 那么 读取多通道气体传感器样本 alcohol 的数据 10 秒', and '如果 按键 C 为按下? 那么 读取多通道气体传感器样本 air 的数据 10 秒'. To the right, there's a 'Labels' section with three boxes: 'cola', 'alcohol', and 'air'. Below it is a 'Data Collection Steps Introduction' box. The status bar at the bottom says '数据 0个(0个标签, 总时长0秒)'.

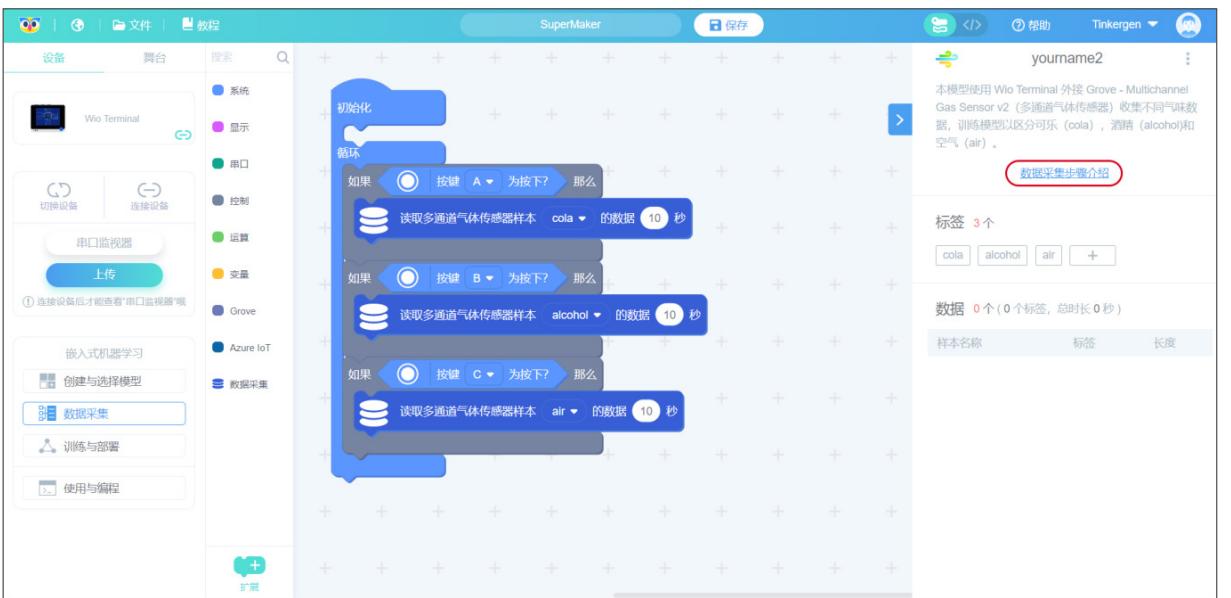
2.2 连接设备并上传 Codecraft 中的默认数据采集程序

Wio Terminal 连接后，在 Codecraft 界面，点击 **上传** 即可上传默认数据采集程序。

The screenshot shows the 'Data Collection' steps in the Codecraft IDE after upload. The steps remain the same: '初始化', '如果 按键 A 为按下? 那么 读取多通道气体传感器样本 cola 的数据 10 秒', '如果 按键 B 为按下? 那么 读取多通道气体传感器样本 alcohol 的数据 10 秒', and '如果 按键 C 为按下? 那么 读取多通道气体传感器样本 air 的数据 10 秒'. The status bar at the bottom says '数据 0个(0个标签, 总时长0秒)'.

2.3 数据采集

在右上角的超链接中，您将找到数据采集的分步介绍。请按照说明收集数据。

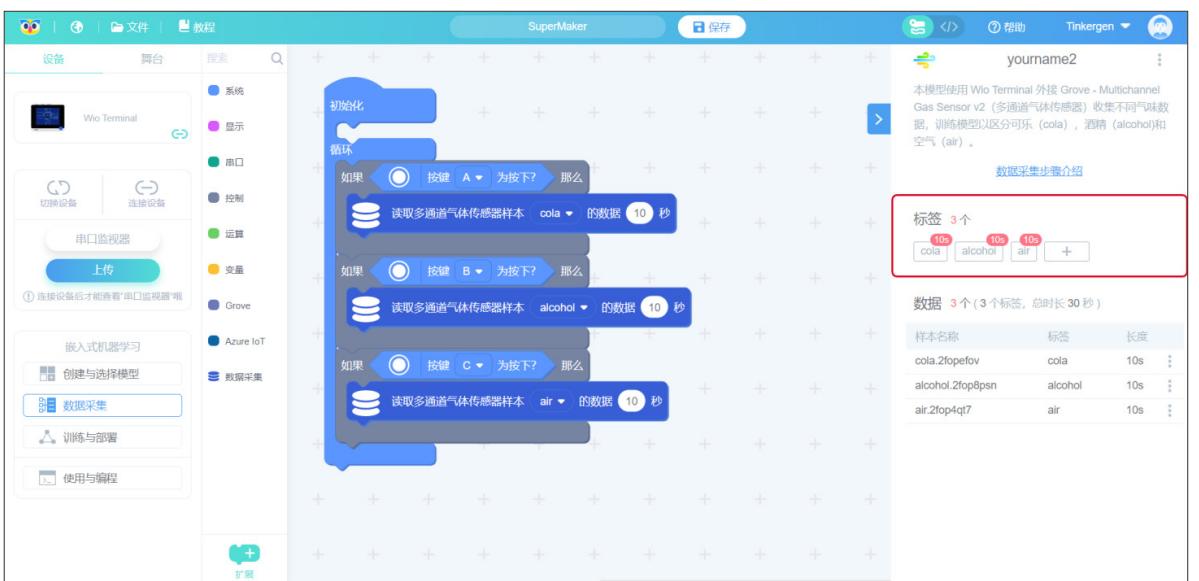


注意：

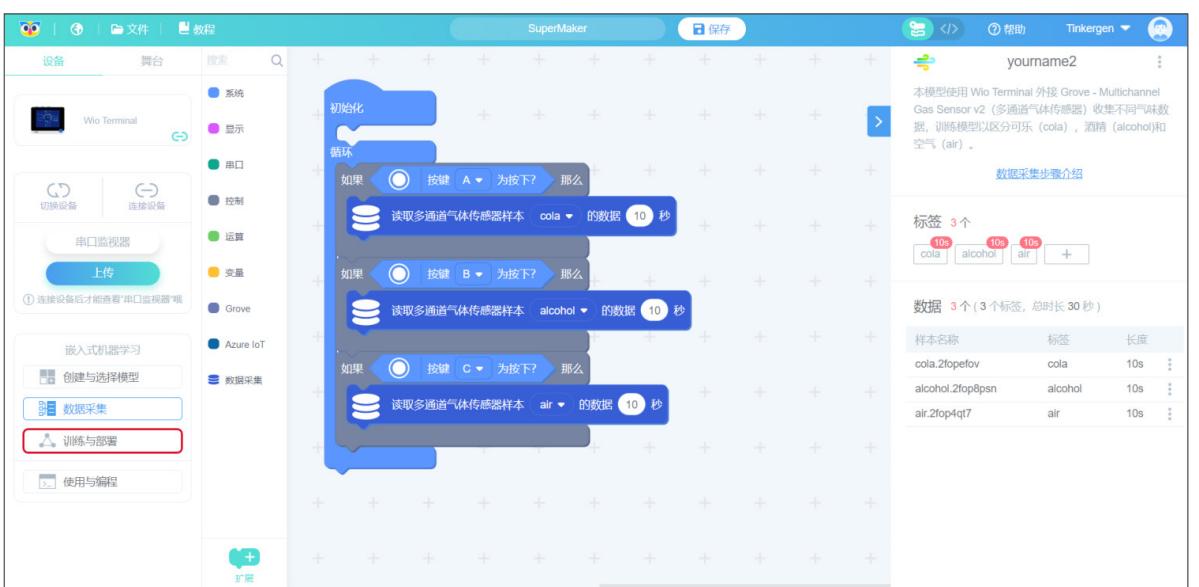
- Wio Terminal 按钮位置。
- 动画是已经加速过的，实际动作可能有所缓慢。
- 请注意红色提示。
- 将光标指向描述文本以获得更详细的内容。



数据全部采集完成。



单击“训练与部署”。



步骤 3. 训练与部署

3.1 设置神经网络和参数

选择你觉得合适的神经网络规模：小型、中型、大型。

接着设置参数、训练周期数（正整数）、学习率（从 0 到 1 的数字）和最小置信度（从 0 到 1 的数字）。初始页面已提供一些默认参数值。

在这里我们使用选择的规模为中，这会耗一段时间。



3.2 开始训练模型

点击“开始训练”。



当您点击“开始训练”时，界面会显示“拼命加载中...”。

“拼命加载中...”的持续时间取决于所选神经网络的规模（小型、中型、大型）和训练周期数。通常来说，网络规模和训练周期数越大，所需的时间就越长。



3.3 观察模型性能，选择理想模型

在“模型训练报告”窗口中，您可以观察训练结果，包括模型的准确率、损失和性能。



如果训练结果不理想，您随时可以回到第一步训练模型，选择另一个大小的神经网络，调整参数训练等，直到得到一个结果满意的模型。

3.4 部署理想模型

在“模型训练报告”窗口中，点击“模型部署”。



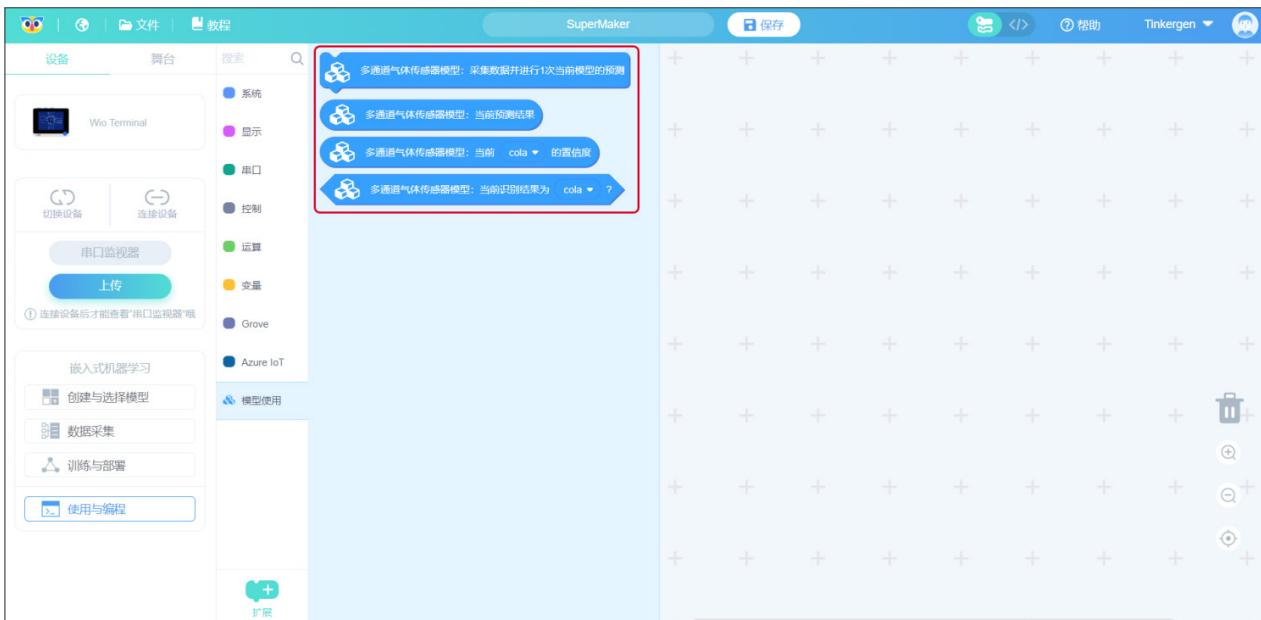
等到部署完成后，点击“确认”即可跳转到“使用与编程”窗口。



步骤 4. 使用与编程

4.1 编写使用模型的程序

在“使用编程”界面，点击“模型使用”以调用部署的模型。

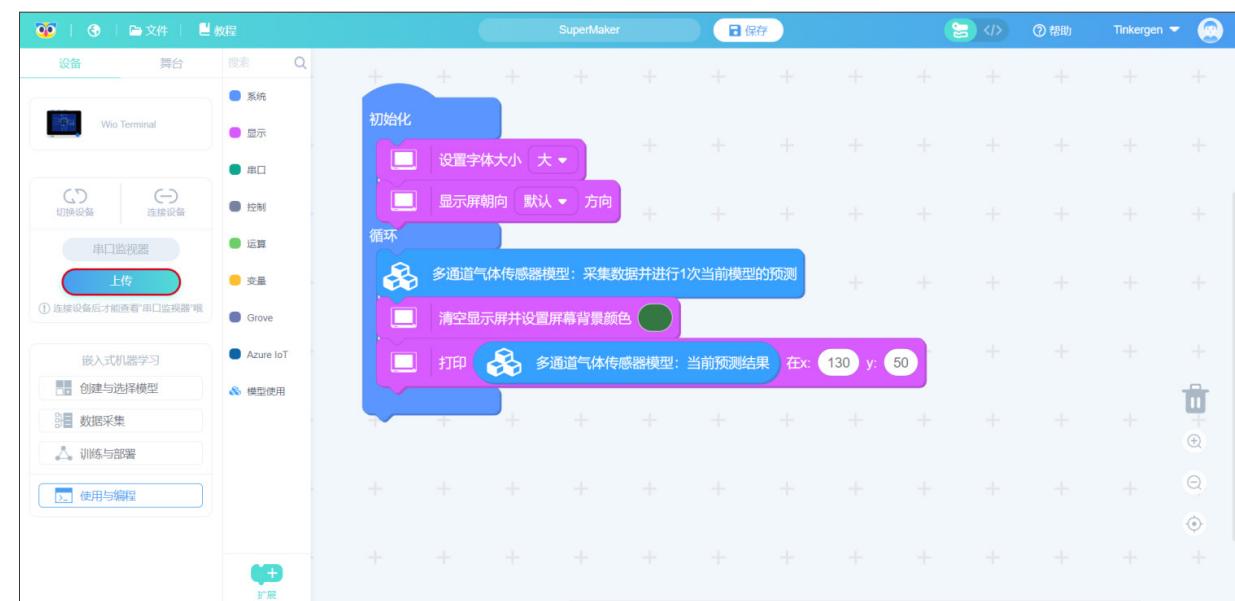


您可以尝试通过编写以下程序来调用您的模型。



4.2 将程序上传到 Wio Terminal

单击“上传”按钮。



第一次上传时间通常比较长，并且随着模型的复杂性时间会进一步增加。较小型号的上传时间约为 4 分钟，较大型号可能会消耗更长时间，这取决于您机器的性能。

4.3 Wio Terminal 测试模型

将你的 Grove 多通道气体传感器靠近可乐来看看 Wio Terminal 是否会显示“可乐”，再尝试一下其他的饮料，看看 Wio Terminal 是否可以识别。

恭喜你！您已经完成了第四个 TinyML 模型。相信你已经尝试设置过参数 (parameters)，训练循环次数 (training cycles)，学习率 (learning rate) 和最小置信度 (minimum confidence rating)，然而这些参数是如何影响我们的模型性能的？

那现在就让我们开始学习这些神奇的参数吧。

机器学习理论（超参数）：

超参数 (Hyperparameters)

超参数在我们训练我们的神经网络模型的时候尤为重要，在训练前设置合适的超参数可以优化权重

和偏差。它可以分为：

- 决定神经网络结构的变量
 - a. 隐藏单元数
 - b. 隐藏层数
- 决定神经网络如何训练的变量
 - a. 训练周期 / 时期数
 - b. 学习速率

在上节课中，我们学习了决定神经网络结构的变量。在这节课中，我们将学习决定神经网络训练的变量。



1. 训练周期的次数

训练周期数也被称为 Epoch，一个 Epoch 代表着将整个训练数据集输入模型进行一次训练的时间，很多情况下一个 Epoch 会对计算机负担很重，我们可以小批量地将训练数据输入到算法中，这样子就可以减少其负荷。

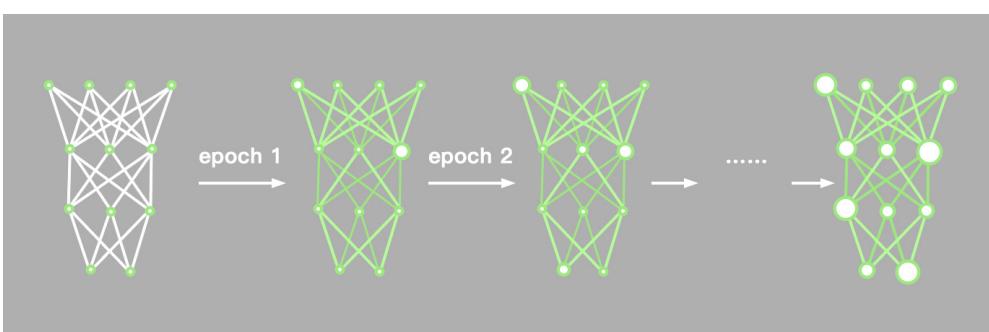
我们可以自己尝试观察训练周期数对模型训练的影响：

1.1 将神经网络模型设置“训练周期数”为“1”。这样就会限制训练只有单个迭代，然后单击“Start training”，观察结果。

1.2 接着将“训练周期数”改为“2”，观察结果你会发现模型性能得到提高。最后，试着将“训练周期数”改为“100”或更多，尝试让机器完成并输入其他测试集来观察结果。

Epoch 是算法的一个重要超参数。Epoch 的值具体代表的是训练数据集完整输入到机器学习模型中，对模型算法进行训练的次数。所以对于每个 Epoch，模型内部参数都会被更新且模型会更加拟合训练数据集里的点，但可能你也会发现，训练太多次之后反而训练出来的模型会有大的偏差，即模型太过于拟合训练集而无法对测试集作出令人满意的预测和分类。

选择合适的训练周期次数也是极其重要的。

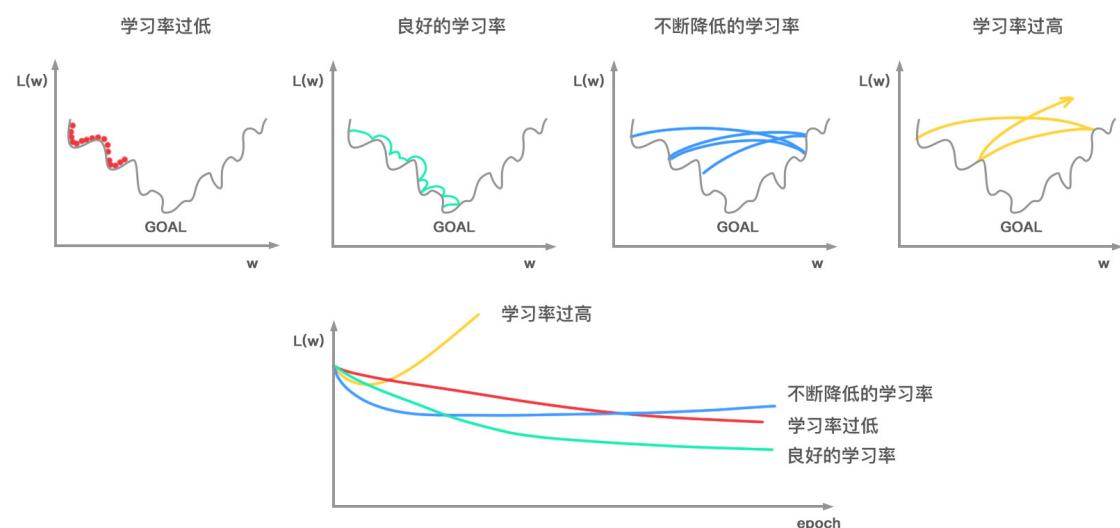


2. 学习率

学习率是一个超参数，在每次模型权重更新时，都可以通过它来调整模型权重以减少误差。选择学习率是同样具有挑战性的，如果学习率过小，训练时间就会过长，而过大的学习率则可能导致学习过快从而只能达到次优值而非最优值，也可能导致训练过程不稳定。

在训练神经网络模型时，学习率是最重要的超参数。因此，研究学习率对模型性能的影响，以及拥有能判断出符合模型性能的最佳学习率的直觉也是非常重要的。

学习率就相当于我们完成训练模型要走的步伐，我们既不想把步伐迈大了忽略了很多数据集的点，也不想迈小了导致我们要走的时间非常长，同时还可能误判“最优解”的值。



如上图所示，如果没有选到合适的学习率，将会带来一系列问题：

- 学习率太高：我们总是错过进入山谷最低（最优解）的路。
- 学习率太低：我们不知道什么时候能达到目标，时间消耗太长，并且我们可能会达到一个“假”的最低山谷，也就是假“最优解”。
- 良好的学习率：经过适当的时间就达到预期的目标。
- 不断降低的学习率：一步一步调整到达预期的目标。

★ 总结

1. 背景理论知识：

- 气体传感器
- 2. 嵌入式机器学习实践

3. 决定网络结构的变量

- a. 隐藏单元数
- b. 隐藏层数

决定网络如何训练的变量

- a. 训练周期数：将整个训练数据集输入模型的时间段称为 Epoch
- b. 学习率：在每次更新模型权重时，根据估计误差控制对模型的更改



更多课程即将推出

进阶项目			
6	使用内置加速度计进行运动识别	<ul style="list-style-type: none"> 背景知识：运动识别的原理介绍 练习与实践： <ul style="list-style-type: none"> 创建与选择模型 数据采集 训练与部署 使用与编程 机器学习理论（模型评估） 	Wio Terminal
7	使用内置光线传感器识别条码	<ul style="list-style-type: none"> 背景知识：条码的原理介绍 练习与实践： <ul style="list-style-type: none"> 创建与选择模型 数据采集 训练与部署 使用与编程 机器学习理论（提高模型训练效果） 	Wio Terminal
8	使用热成像传感器进行人脸识别	<ul style="list-style-type: none"> 背景知识：热成像传感器原理介绍及人脸识别 练习与实践： <ul style="list-style-type: none"> 创建与选择模型 数据采集 训练与部署 使用与编程 机器学习理论（层 - layer） 	Wio Terminal Grove - Thermal Imaging Camera (MLX90640)
总结			
9	创意项目和总结	<ul style="list-style-type: none"> ML 理论知识总结 创意项目的例子 	

后记

Seeed Wiki: Wio Terminal 入门教程

<https://wiki.seeedstudio.com/Wio-Terminal-Getting-Started/>

Codecraft

<https://ide.tinkergen.com/>

物品链接

Wio Terminal:

<https://www.seeedstudio.com/Wio-Terminal-p-4509.html>

Grove 多通道气体传感器：

<https://www.seeedstudio.com/Grove-Multichannel-Gas-Sensor-v2-p-4569.html>

Grove 热成像照相机：

<https://www.seeedstudio.com/Grove-Thermal-Imaging-Camera-IR-Array-MLX90640-110-degree-p-4334.html>

我们正寻求多语言翻译人才！

如果你可以给我们课程提供除中英文以外的翻译版本，请联系我们！

联系方式

电话: +86-0755-86716703

地址: 广东省深圳市南山区 1001 中山园路 TCL 国际 E 城 G3 栋 1002 室

网址: contact@chaihuo.org

www.tinkergen.com

www.seeedstudio.com

课程撰写人:

此课程为矽递科技股份有限公司的雇员联合撰写：

主编: 赖慧莹

设计: 孟依卉

校对: 于剑锋

翻译: 于剑锋



 **seeed**^{studio}

想了解关于矽递的更多信息，请访问：www.seeedstudio.com