TINY-ML PROJECT REPORT

COMP 554

WIZARDING OWL'S MAGIC WAND

FALL 2022

NAYANA CHANDRASHEKARA SHARMA
Dept. Of Computer Science
Rice University
Houston, Texas
Net ID: **nc54**
nc54@rice.edu

SARA NABIYA
Dept. Of Computer Science
Rice University
Houston, Texas
Net ID: **sn62**
sn62@rice.edu

I. ELEVATOR PITCH

In the project, Wizarding Owl's Magic Wand, the Arduino Nano 33 BLE Board will be used to execute a deep learning model using TensorFlow Lite for Microcontrollers. The user can wave the microcontroller like a digital "magic wand" to cast "spells". As the user moves the wand, the model receives the three-dimensional data as input and returns a simple visual representation of the movement performed by the user.

To understand physical gestures, the wand application uses a device's accelerometer to collect information about its motion through 3D space. An accelerometer measures the degree of acceleration that it is currently experiencing. And based on this 3D data, the machine learning model infers and classifies the wand's movement.

The report covers what we are constructing in terms of machine learning, as well as its architecture, the tools required, the tests performed, the challenges we faced, and the deployment procedure.

II. HARDWARE AND SOFTWARE COMPONENTS

- Linux, MacOS, or Windows laptop
- Arduino Nano 33 BLE Microcontroller Board
- Micro USB-C Cable
- Arduino IDE, a software that is used to write and upload computer code to the physical board.
- Make Tool
- Google Colaboratory
- TensorFlow and Keras

III. STORY

A. Application Architecture

- Main loop: The application runs in a continuous loop. Since its model is small and simple and there's no preprocessing required.
- Accelerometer Handler (Input): This component captures data from the accelerometer and writes it to the model's input tensor. A 3-axis accelerometer outputs three values representing the amount of acceleration on the device's X, Y, and Z axes. The model takes these values directly as its input.
- TF Lite Interpreter: The interpreter runs the TensorFlow Lite model.
- Model: The model is included as a data array, and run by the interpreter. It is a trained Convolutional Neural Network (CNN), weighing in at 20 KB, that accepts raw accelerometer values as its input.
- Gesture predictor: This component takes the model's output and determines the gesture.
- Output Handler: Once data has been captured and inference has been run, the output handler prints the output on the terminal, and lights an LED.

B. Training the Model

We used the training scripts in the TensorFlow repository to train the model. The Jupyter Notebook can be run in Google Colabotary on a GPU runtime.

The scripts perform the following tasks:

- Prepare raw data for training
- Generate synthetic data
- Split the data for training, validation, and testing
- Perform data augmentation
- Define the model architecture
- Run the training process
- Convert the model into the TensorFlow Lite format
- Convert the model to a C file

C. Testing the Main Components

The tests are performed on the following main components - magic wand (model), accelerometer handler, gesture predictor, and output handler.

- Magic Wand: Feeds the data into the model, runs the inference, and compares the results with the expected value.
- Accelerometer Handler: Tests the one-time setup the accelerometer needs; tests only if 384 bytes of data are available in the array and then the inference begins to work.
- Gesture Predictor: Once inference has occurred, we can stipulate that for a gesture to be recognized, it must have been detected in at least a certain number of consecutive inferences. Since we run inference multiple times per second, we will quickly be able to decide whether a result was valid. This is the job of the gesture predictor.
- Output Handler: Tests that after getting the results from the gesture predictor, it can display all the results correctly to the terminal.

D. Deploying to Arduino

1. Download and install Arduino IDE to your system.

2. Add the following TensorFlowLite libraries:

Tool → Manage Libraries → In the window that appears, search and install the library named TensorFlowLite_ESP32 and/or Harvard_TinyMLx. Also, install the ArduinoBLE and the Arduino_LSM9DS1 libraries.

3. Load magic_wand files from Examples:
File → Examples → Harvard_TinyMLx → magic_wand

4. Next, take the output (model) generated during the training phase from the cell that ran xxd on our model and copy the contents of the array to the array present in the magic_wand_model_data.cpp file. So we deploy the model that we generated.

5. Then, make sure to use the correct device type and port:
Tools → Board → Arduino Nano 33 BLE
Tools → Port → COM6

6. Run the application: Hit the upload button to compile and upload the code to the Arduino device. Once the upload completes, open the Serial Monitor (Tools → Serial Monitor). Now make any gesture using the microcontroller (magic wand) and you will be able to visualize the same gesture on the screen. Refer to our video for more details.

IV. CHALLENGES

A. Issues while installing the libraries

While importing the libraries, there were runtime issues as the TensorFlow was not of the latest edition.

Error: pip's dependency resolver does not currently take into account all the packages that are installed. The issue was resolved using the latest version of TensorFlow.

B. Issues with the Make tool

We had to install the latest version of the Make tool and use the command gmake instead of make to execute the files.

C. Issues with the link of the given GitHub Repository

The GitHub repository mentioned in O'Reilly's TinyML book has been moved to the following new location https://github.com/tensorflow/tflite-micro. So we had to make use of this new repository and modify the commands while cloning it.

D. Installing the accelerometer driver

The ArduinoBLE and the Arduino_LSM9DS1 libraries had to be installed manually otherwise the code would error out.

IV. SUPPORTING LINKS

The link to our GitHub repository is as follows, which has the code to both the Hello World and the Magic Wand code. It includes all the files needed to build and execute the project.

GitHub Link to our Code:
https://github.com/saranabiya5/TinyML-Project

The Youtube video link of the output of the project is as follows:

Magic Wand Output:
https://youtube.com/shorts/X7xZITxPKVk?feature=share

V. CONCLUSION

Machine learning can be used to build intelligent tools that make consumers' life easier. But often, these tools demand a lot of processing power and resources, such as a strong cloud server or a desktop. However, it's now possible to use tiny, low-powered devices, such as microcontrollers, to run machine learning inference.

Microcontrollers are very widely used, use very little power, and are incredibly dependable. They are present in a wide range of home furnishings, including appliances, cars, toys, etc. In fact, some 30 billion devices that are powered through a microcontroller are produced every year. Without requiring expensive hardware or reliable internet connections, we can increase the intelligence of these gadgets we use every day by introducing machine learning to tiny microcontrollers.

VI. REFERENCES

[1] Warden, P., & Situnayake, D., *TinyML Book.* O'Reilly.

[2] Jain, Advait. *tflite-micro GitHub Repository*. Retrieved from Github: https://github.com/tensorflow/tflite-micro.git

[3] Warden, P., *TinyML Book Screencast #2*. Retrieved from Youtube: https://www.youtube.com/watch?v=AfAyHheBk6Y&t=1542s