Assignment 4

Embedded Machine Learning for Edge Comuputing
SkillSurf 2024

Introduction
In this assignment, you will get a manual approach to TinyML by implementing a simple machine learning model on the Arduino Uno v3. While the Arduino Uno has limitations in terms of memory and processing power, we will creatively work around these constraints to run a sine wave prediction model that controls the brightness of an LED. You will learn to achieve the desired functionality without relying on TensorFlow Lite, making it a valuable exercise in resource-constrained environments. The assignment has been divided into two sections,

- Section 1 - Training the model
- Section 2 - Microcontroller programming

Note
To implement the code on an Arduino Uno v3, you may use the online simulator platform Wokwi. Alternatively, you may use a physical Arduino Uno or a STM32 Nucleo device and other relevant components if you have your own.

Submission
Starter code for the assignment is included in the *starter_code.zip* folder. Complete the code where it is mentioned and submit the following files,

1. A pdf of the completed Google Colab (ipynb) notebook
2. Completed *model.cpp, model_data.cpp, sketch.ino* files
3. A screenshot of the Wokwi interface with serial plotter open

Compress the above files into a zip folder and rename it with your index number (i.e. *index_no.zip*) and submit it via the Google Form.

Section 1

Load the *assignment_4_notebook.ipynb* file to Google Colab or Jupyter Notebook. Make sure Tensorflow 2 is installed on the system. Continue running the notebook and fill the relevant cells with the correct code. After completing, export the completed notebook (with the plots and results included) as a pdf file.

Section 2

Set up a empty Wokwi Arduino project on https://wokwi.com/arduino. Follow this page to get a brief introduction to the Wokwi simulator platform. Alternatively, you may watch this video.

Exercise 1
Connect a LED to pin 11 of the Arduino Uno with a 220 Ω resistor. Make sure the LED is properly grounded.

Upload the *sketch.ino, model_data.h, model_data.cpp, model.h, model.cpp* files to the Wokwi editor. Complete the files *sketch.ino, model_data.cpp, model.cpp* as instructed as below. Copy and paste the weights and biases contents from the Colab notebook to the *model_data.cpp* file.

Exercise 2
Open the *model.cpp* file and complete the *ReLU* function and the forward pass through the hidden layers in the *predict()* function.

Remember that the ReLU activation function is defined as,

$$ReLU(x) = max\{0, x\}$$

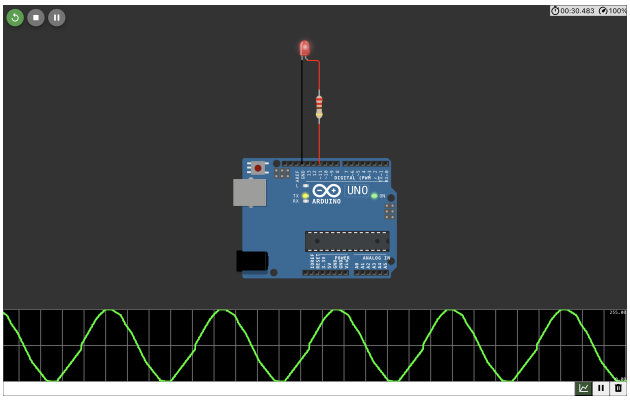
You may use **fmax()** from *math.h* library to implement this.

Remember that a forward pass through a hidden layer is like,

$$h = ReLU(W \cdot x + b)$$

Using your knowledge on matrix multiplication, get the outputs after first and second hidden layer and output layer. For convenience, try drawing a diagram and understanding how the matrices are multiplied. You may use **for loops** for this, vectorized implementation is not needed.

Exercise 3
Open the *sketch.ino* file. Call the *predict()* function to obtain the y_value. Write the brightness value using **AnalogWrite()** to the LED pin and also serial log the brightness value.
Run the Wokwi simulation and observe the LED pulsating in a sinusoidal pattern. Open the serial plotter and observe the sinusoidal plot. Take a screenshot of the interface with the waveform visible. The final output should look something similar to the below image.



End of Assignment