LAB-9 REPORT

Subject: **Embedded Hardware Design**

Subject Code: **EL203**

Members: 1) Shantanu Tyagi **(201801015)**
2) Nikhil Mehta **(201801030)**
3) Sudhanshu Mishra **(201801114)**
4) Bhavana Kolli **(201701082)**

Introduction

Embedded artificial intelligence refers to the use of machines and deep learning in software at the device level (AI). The programme may be designed to give both predictive and reactive intelligence based on the data collected and reviewed. AI models may be run on devices having embedded AI, and the results can subsequently be used to achieve a purpose or function. TinyML (embedded machine learning) is a machine learning technique that is applied to embedded devices. Some of its benefits, when installed on systems, are listed below:

1) Less/No Bandwidth constraints
2) Improved latency
3) Cost-efficient

Standard computer programming addresses issues by expressing strategies directly in code and instructing computers to execute logic to process data and generate an output. Machine Learning (ML), on the other hand, is an AI technique for detecting patterns in data and, in essence, learning from it. Pre-labelling data (or not), using reinforcement learning to guide algorithm development, extracting features via statistical analysis, and then classifying input data against this trained data set to determine an output with a stated degree of confidence are all examples of how this can be accomplished.

In an ideal embedded Machine learning ecosystem, device providers, original equipment manufacturers (OEMs), and machine learning models are employed and deployed. It also expands the Machine Learning ecosystems by including approaches and tools for developing and deploying embedded devices, such as the Internet of Things (IoT), sometimes known as Artificial Intelligence on IoT devices (AIoT).

CODE:

main.c

```
// Multiple input Single output Neural Network model to predict
whether a person
// is affected by COVID-19 or or Healthy, given a particular
temperature, blood pressure, and blood sugar

#include "uart.h"
#include "simple_neural_network.h"

#define NUM_OF_INPUTS 3

double temperature[6] = {36,37,38,39,40,41};
double blood_pressure[6] = {110,120,130,140,150,160};
double blood_sugar[6] = {120,140,160,180,200,220};

double weight[3] = {-2,1,1};

double training_eg2[3] = {37,120,140};
double training_eg6[6] = {41, 160, 220};
//double training_eg2[3] = {temperature[2], blood_pressure[2],
blood_sugar[2]};

//hw  -> double training eg2[3] 3rd line me koi error hai, remove
that
int main(void){
     USART2_Init();

     printf(" Prediction of non-COVID or COVID: %f\r\n",
multiple_in_single_out(training_eg2, weight, NUM_OF_INPUTS));
     while(1){}
}
```

simple_neural_network.h

```
#ifndef __SIMPLE_NEURAL_NETWORK
#define __SIMPLE_NEURAL_NETWORK

#include <stdint.h>
```

```
double single_in_single_out(double input, double weight);
double weighted_sum(double *input, double *weight, uint32_t
INPUT_LEN);
double multiple_in_single_out(double *input, double *weight,
uint32_t INPUT_LEN);

#endif
```

uart.h

```
#ifndef __UART_H
#define __UART_H
#include "stm32f4xx.h"
#include <stdio.h>

void USART2_Init(void);
void test_setup(void);

#endif
```

simple_neural_network.c

```
#include "simple_neural_network.h"

double single_in_single_out(double input, double weight)
{
	return(input * weight);
}



double weighted_sum(double *input, double *weight, uint32_t
INPUT_LEN)
{
	double output=0;
	uint32_t i;
	for(i=0; i<INPUT_LEN; i++){
		output += input[i]*weight[i];
	}
	return output;
}
```

```
double multiple_in_single_out(double *input, double *weight,
uint32_t INPUT_LEN)
{
      double predicted_data;
      predicted_data = weighted_sum(input, weight, INPUT_LEN);
      return predicted_data;
}
```

uart.c

```
#include "uart.h"


void USART2_Init(void){
 //Enable clock for UART. at APB1


  /*ENABLE clock for USART2*/
      RCC->APB1ENR |=  0x20000;
      RCC->AHB1ENR |=  0x1;

        /* Configure PA2, PA3 for USART2 TX, RX */
  GPIOA->AFR[0] &= ~0xFF00;
  GPIOA->AFR[0] |=  0x7700;   /* alt7 for USART2 */

      GPIOA->MODER  &= ~0x00F0;
  GPIOA->MODER  |=  0x00A0;   /* enable alt. function for PA2, PA3
*/

  USART2->BRR = 0x0683;       /* 9600 baud @ 16 MHz */
  USART2->CR1 = 0x000C;       /* enable Tx, Rx, 8-bit data */
  USART2->CR2 = 0x0000;       /* 1 stop bit */
  USART2->CR3 = 0x0000;       /* no flow control */
  USART2->CR1 |= 0x2000;      /* enable USART2 */
}


int USART2_write(int ch){
      while(!(USART2->SR & 0x0080)){}
```

```
      USART2->DR =(ch &0xFF);
      return ch;

}

int USART2_read(void){
      while(!(USART2->SR & 0x0020)){}
      return USART2->DR;

}

struct __FILE { int handle; };
FILE __stdin  = {0};
FILE __stdout = {1};
FILE __stderr = {2};

int fgetc(FILE *f) {
    int c;

    c = USART2_read();      /* read the character from console */

    if (c == '\r') {        /* if '\r', after it is echoed, a '\n'
is appended*/
        USART2_write(c);    /* echo */
        c = '\n';
    }

    USART2_write(c);        /* echo */

    return c;
}

/* Called by C library console/file output */
int fputc(int c, FILE *f) {
    return USART2_write(c);  /* write the character to console */
}


int n;
char str[80];

void test_setup(void){
```

```
  printf("Please enter a number:");
      scanf("%d",&n);
      printf("The number you entered is : %d\r\n",n);
      printf("Please enter a character string :");
      gets(str);
      printf("The string you entered is : ");
      puts(str);
      printf("\r\n");

}
```

Screenshot:

Single input:

```
compiling system_stm32f4xx.c...
linking...
Program Size: Code=1160 RO-data=424 RW-data=44 ZI-data=1636
".\Objects\Lab9.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:02
```

Output = 36 * 2 = 72 => person not infectred with COVID

Multiple input:

```
linking...
Program Size: Code=6060 RO-data=600 RW-data=232 ZI-data=1632
".\Objects\Lab9.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:01
```

Output = 186 => The values less than these imply not infected with COVID

Conclusions/Observations:

1. From the last lab we created a driver which we will now use in this lab to implement said network.
2. One Network is single input and single output and another network is Multiple input and Single output.
3. A few other notable networks are single input and multiple output, multiple input and multiple output & multiple input and multiple output with hidden layers.

4. The main Idea behind the program is to see if a person is affected by COVID or they are a healthy person through the Single input and Single Output neural network, taking the temperature as input.
5. And we also implemented a situation where we take multiple input i.e temperature, blood pressure and blood sugar to give a single output.