JAI HOO SATNARYAN.

SIX WEEKS INTERNSHIP AT IIT JAMMU

INTERNSHIP REPORT

1ST WEEK

Day 1: Visit the Institute (IIT Jammu)

My first day will involve familiarizing yourself with the institute's facilities, meeting with my mentors, and getting an overview of the internship program. This day sets the stage for the rest of my internship, giving me a chance to understand the environment and resources available.

Day 2: History of TinyML

1. Google's OK Google Team:

- In 2014, Google's OK Google team developed 14 KB neural networks to run on DSPs (Digital Signal Processors) in Android phones. These DSPs had very limited RAM and flash memory, which required the neural networks to be extremely small.
- The goal was to continuously listen for the "OK Google" wake words without draining the phone's battery. This was possible because DSPs use only a few milliwatts of power, unlike the main CPU which consumes more energy and is powered off to conserve battery.

2. TinyML Defined:

- TinyML refers to the practice of running machine learning models on devices that consume less than 1 milliwatt of power. This low energy consumption allows these devices to run on small batteries, such as coin batteries, for extended periods (up to a year).
- The concept of TinyML opens up new possibilities for smart devices that are low-cost, low-power, and can operate independently for long durations.

3. New Applications:

- TinyML enables innovative applications like peel-and-stick sensors. These sensors are small, battery-powered devices that can be attached to any surface and forgotten about because they require no battery changes and can function autonomously.

4. Importance of TinyML:

- The idea of TinyML is transformative because it allows intelligent data processing directly on small, energy-efficient devices. This reduces the need for continuous data transmission to centralized servers, which is often impractical due to high energy costs.

Day 3: Embedded Devices and Resource Constraints

1. Limited Resources:

- Embedded devices typically have very constrained resources, such as only a few hundred kilobytes of RAM and flash memory. They operate at low clock speeds (tens of megahertz) and are designed for specific tasks.

2. Development Tools:

- Despite these constraints, development on embedded devices has become more accessible thanks to standardized hardware platforms and user-friendly integrated development environments (IDEs) like Arduino. Arduino provides a simplified and consistent development experience, making it easier to prototype and deploy embedded applications.

3. Reliability Considerations:

- Embedded systems often avoid using dynamic memory allocation (functions like `new` or `malloc()`) to maintain reliability. Dynamic memory can lead to fragmentation and unpredictable behaviour over long periods, which is undesirable for devices that need to run continuously without human intervention.

Rapid Evolution of TinyML

1. Quick Advancements:

- The field of TinyML is rapidly evolving. Both hardware and software components are continuously improving, offering more capabilities and better performance over time. Research in this area is also progressing quickly, leading to new techniques and applications.

2. Using TensorFlow Lite:

- During my internship, i will be using TensorFlow Lite, a version of TensorFlow designed for mobile and embedded devices. TensorFlow Lite has a stable API.

Day 4: Getting Started with TinyML

1. Hardware Requirements

Computer:

- **Laptop or Desktop with USB Port:**
  - You will use your computer to write, edit, and compile the code for your TinyML projects.
  - The USB port is essential for connecting your computer to the embedded development board.

Embedded Development Boards.

- **Arduino Nano 33 BLE Sense:**
  - A versatile and widely-used development board.
  - Equipped with various sensors (microphone, accelerometer, etc.), which are useful for machine learning projects.
- **STM32F746G Discovery Kit:**
  - Another development board option with powerful features for embedded applications.
  - Suitable for more advanced projects requiring a bit more processing power and flexibility.

2. Software Requirements

Machine Learning Framework:

- **TensorFlow Lite for Microcontrollers:**
  - A lightweight version of TensorFlow Lite tailored for devices with limited computational resources.
  - Designed to run on microcontrollers with only a few tens of kilobytes of memory available.
  - Provides the necessary tools and libraries to develop and deploy ML models on low-power devices.

Command Line Interface:

- **Terminal (macOS/Linux) or Command Prompt (Windows):**
  - You will use the terminal or Command Prompt to enter commands for building and deploying your ML projects.
  - These interfaces allow you to interact with the development environment and execute scripts necessary for your projects.

Additional Software:

- **Development Board-Specific Software:**

- Depending on the development board I choose, additional software might be needed.
- For example, if I use the SparkFun Edge or Mbed devices, I might need to install Python and other tools for building and debugging my projects.
- The Arduino IDE includes most of what I need for Arduino boards, so additional software is minimal.

Day 5: Learning with Arduino TinyML Learning Kit

Today, I delved into understanding and working with the Arduino TinyML Learning Kit. Here are the key takeaways and activities:

1. **Overview of Arduino TinyML Learning Kit:**
   - **Purpose and Components:** Explored the kit's components and their functionalities, including the microcontroller, sensors, and necessary interfaces.
   - **Installation of Libraries:** Learned how to install essential libraries required for TinyML projects on the Arduino platform.
2. **Learning Process:**
   - **Hands-on Exercises:** Engaged in practical exercises to familiarize myself with basic operations and configurations using the Arduino TinyML Learning Kit.
   - **Library Usage:** Implemented various libraries specific to TinyML applications, understanding their role in model deployment and data processing.
3. **Functional Demonstrations:**
   - **Testing and Validation:** Conducted tests to validate functionality and performance of models deployed on the TinyML Learning Kit.

Day 6 and Day 7: Weekend Break

Given that these days were designated as weekend breaks, no specific technical learning or activities related to the project were conducted during this period.