L03 Deploying a Simple AI Model on a Simulated Edge Device using Visual Studio Code

Start Assignment

- Due Sunday by 11:59pm
- Points 100
- Submitting a file upload
- File Types docx, doc, odt, pptx, pdf, ipynb, and nb

Deploying a Simple AI Model on a Simulated Edge Device

INDIVIDUAL Assignment

Objective:

Deploy a simple AI model on a simulated edge device to understand the basics of edge computing and AI integration.

Using Visual Studio Code (VS Code)

Step-by-Step Instructions:

Step 1: Set Up the Environment

1. **Install Python:**
   - Download and install Python from **Python's official website (https://www.python.org/)** .
2. **Install VS Code:**
   - Download and install Visual Studio Code from **VS Code's official website (https://code.visualstudio.com/)** .
3. **Install TensorFlow:**
   - Open a terminal in VS Code and run:

bashCopy codepip install tensorflow

4. **Install Edge Impulse CLI:**
   - Install Node.js and npm from **Node.js official website (https://nodejs.org/)** .
   - Run the following command in the terminal:

bashCopy codenpm install -g edge-impulse-cli

Step 2: Prepare the Dataset

1. **Load and Preprocess the Data:**

- Create a new Python file in VS Code and write the following code to load and preprocess the MNIST dataset:

pythonCopy codeimport tensorflow as tffrom tensorflow.keras.datasets import mnist (x_train, y_train), (x_test, y_test) = mnist.load_data()x_train, x_test = x_train / 255.0, x_test / 255.0 x_train = x_train.reshape((-1, 28, 28, 1))x_test = x_test.reshape((-1, 28, 28, 1))

Step 3: Train a Simple AI Model

1. **Define the Model:**

pythonCopy codemodel = tf.keras.Sequential([tf.keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)), tf.keras.layers.MaxPooling2D((2, 2)), tf.keras.layers.Flatten(), tf.keras.layers.Dense(128, activation='relu'), tf.keras.layers.Dense(10, activation='softmax')])

2. **Compile the Model:**

pythonCopy codemodel.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])

3. **Train the Model:**

pythonCopy codemodel.fit(x_train, y_train, epochs=5, validation_data=(x_test, y_test))

Step 4: Convert and Deploy the Model

1. **Convert the Model to TFLite:**

pythonCopy codeconverter = tf.lite.TFLiteConverter.from_keras_model(model)tflite_model = converter.convert() with open('model.tflite', 'wb') as f: f.write(tflite_model)

2. **Upload the Model to Edge Impulse:**
   - Open a terminal in VS Code:

bashCopy codeedge-impulse-uploader --api-key <your-api-key> model.tflite

3. **Simulate the Edge Device:**
   - Follow the instructions provided by Edge Impulse to simulate the edge device environment and test the deployed model.

Step 5: Test and Validate the Model

1. **Run Inference on the Simulated Edge Device:**
   - Use the Edge Impulse platform to run inference and validate the model performance.
   - Analyze the results and make necessary adjustments.

2. **Reflective Journal:**
   - Keep a reflective journal throughout the exercise.

- Include detailed snippets of the simulation, observations, and personal reflections on the process and outcomes.

3. **Document the Results:**
   - Record the accuracy, latency, and any observations during the testing phase.
   - Provide screenshots and explanations in the Python file or a separate document.

Deliverables:

1. **Simulation Documentation:**
   - A report detailing the setup, deployment process, testing, and validation results.
   - Include screenshots or logs to support your documentation.
2. **Reflective Journal:**
   - Submit an Individual reflective journal with detailed snippets of the simulation, observations, and personal reflections on the exercise. as A02
   - Name your file as usual L03_Student_ Name_ITAI3377