Experiment 05

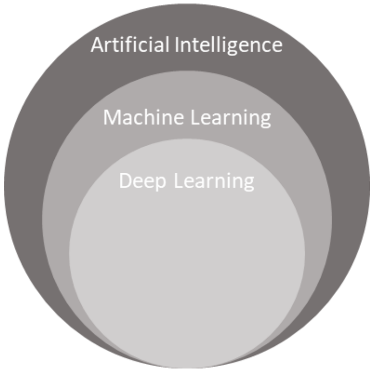
EDGE AI – Movement Recognition

In this experiment, we will create and run a machine-learning model that can identify a four types of movement using the Inertial Measurement Unit (IMU) on the IoT Board.

Background

Humans have designed machines for a long time. Earlier these machines were mechanical, they were designed specifically to solve a particular problem. Even devices like calculators used mechanized systems to run calculations. The software was introduced with the advent of electronic machines, that could run logical instructions efficiently when compared to mechanical machines. The software was programmed to perform predetermined tasks. Any user input outside the recognizable commands would not function and the software lacked “intelligence”. Softwares were not capable of making decisions on its own. Computers capable of mimicking human intelligence were later theorized and pursued as the next frontier.

As computing capabilities grew exponentially over the past few decades, it enabled the design of software that could “learn” and replicate the capacity to “think” at an elementary level. This ability to demonstrate human-like intelligence by machines was termed Artificial Intelligence (AI). In this modern age, AI can be categorized into Narrow Intelligence and General Intelligence. Artificial Narrow Intelligence or Weak AI are AI systems developed to perform a very specific task. For example, an AI that can play chess or an AI that can recognize a person in a video frame. AI such as smartphone assistants also falls under this category though they can perform quite a few tasks their entire purpose is to understand human speech. An AI trained to play chess cannot be deployed to recognize things in an image let alone play another board game at the same proficiency. Artificial General Intelligence or Strong AI or Super AI on the other hand is theorized as having capabilities to perform any given task equivalently or even better than humans. There are a lot of examples of such AI in science fiction, but researchers are exploring the development of such an AI. A close example today would be Chat GPT. It is not fully capable of performing multiple tasks with great effectiveness, but it could very well be a precursor to what a fully realized Super AI would look like.



Machine learning is a type of AI that is designed to learn and adapt from experience rather than programming it for every possible scenario. Machine learning is a methodology to realize AI software. Supervised learning, Unsupervised learning are the two approaches to broadly describe how a machine can be taught or how a machine learns from experience. In Supervised training, the experienced data is “labeled”. The model then creates an objective function that is optimized over several iterations to map the labeled training data to match the output. In unsupervised learning, the training data is unlabeled. The model optimizes the objective function iteratively using statistical approaches. Deep learning is a type of machine learning technique where the algorithm is modeled after a human brain. Similar to the structure of neurons, an artificial network of neurons is created called "artificial neural networks". Deep learning has proven to be very effective in the creation of powerful narrow AI systems. Modern computer vision and speech recognition AI systems are for example developed using deep learning techniques.

Edge AI, or Edge Artificial Intelligence, involves running AI algorithms and data processing on local devices, close to data sources rather than relying solely on distant cloud servers.

💻Experiment Set-up: Configuration

In this experiment, we will use the onboard IMU and the Node MCU to identify what kind of movement was performed by the user. The student will be moving the IoT board and the model will predict four movements: updown, wave, snake and iddle. Generally, such movements can be recognized using a traditional program. But even with a small deviation from the initially programmed movement, the accuracy of detection will fall drastically. Machine-learning-based detection on the hand can detect these movements even when there is a healthy deviation.

Machine learning model - For the class purposes, this has been already done for you, but the instructions are on next page. We encourage you to complete them for a better understanding of the experiment.

In this step we are collecting the training and test data, designing the machine learning model, tunning the model, and deploying it to an Arduino library so we can use it for real-time predictions using the onboard IMU and the Node MCU. All of these steps are done in Edge Impulse.

- Go to the Edge Impulse website and create an account.
- Login to your Edge Impulse account. Then on the user home page, click “Create new project”. Enter the name of your project in the popup field. Click Ok. Now click on your project to enter the Impulse setup page.
- Download the training and test datasets that are available on this link: https://cdn.edgeimpulse.com/datasets/gestures.zip
- On the "Getting started" option, select "Add existing data" and upload the data set from previous step. Select the option to upload individual files to upload the training and test data into its corresponding category using the file names as labels.
- The next process is to design an impulse. Click on the “Impulse Design” and then click on the Create Impulse button from the menu on the left-hand side. In the Time series data windows, set the window size to 2000 ms. Also, set the window increase to 80ms and frequency to 60hz. You should be able to see the three input axes X, Y, and Z.
- Add a processing block for spectral analysis.
- Add a learning block for classification. Name the classifier in the Neural Network window and select the spectral features. The output features should be wave, iddle, snake and updown. Save the impulse.
- The next step is to generate the features. Click on "Spectral features" and generate them. When the process is complete, you should be able to see the feature explorer with the points classified
- In the Neural network settings (Classifier), we can view the training settings. You can initially set the training cycles to 50 and the learning rate to 0.0005. Click on the “start training” button. After the training is completed, we can look at the confusion matrix and accuracy. This matrix/table helps us visualize how the model predicts movements and its confidence in those predictions.
- Next, we will deploy the impulse. In the deploy impulse page, select Arduino under Create Library and download the library. MAGIC JUST HAPPENED! All the learning model is now in the Arduino library.

Instructions

1. Generally, the libraries can be installed from the Library Manager available within the Arduino IDE. But for this experiment, we will use the library we generated in the Edge Impulse deployment. You can use the library provided in Canvas "ei-aramirez2-project-1-arduino-1.0.1" or generate yours following the instructions on previous page. Go to Sketch --> Include Library --> Add .Zip
2. After installing the libraries open the Arduino IDE and click on “File” from the Menu bar. Click on “Open” to open the “aimotion.ino” file that can be found in the course material for this module. Now the IDE displays the code that needs to be flashed onto the microcontroller.
3. Now compile the code and look for any errors. Then proceed to upload the code onto the microcontroller. Refer to the IoT Board Manual for flashing instructions.
4. If the AHA board is not detecting the IMU sensor, connect directly the NodeMCU with the IMU using the wires and the breadboard. The connections you have to wire are:
    a. Microcontroller GND to Breadboard GND
    b. Microcontroller 3.3V to Breadboard 3.3V
    c. Same two previous steps for the motion sensor (using the other side of the breadboard)
    d. SCL in the motion sensor to pin G22 in the microcontroller
    e. 5. SDA in the motion sensor to pin G21 in the microcontroller
5. Open the Serial Monitor from the dropdown menu under “Tools” on the Menu bar. Refer to the IoT Board Manual for setting up the Serial Monitor window.
6. In the Serial Monitor Window, the Node MCU will now display the detected movements (updown, wave, snake, iddle) as the user performs the movements.

☞ Deliverables

Demonstration:

1. Record a video demonstration explaining the outcome of the experiment. Refer to the title page for a brief description of the expected outcome. Make sure you talk over all observations and the video is presentable. Also, don't forget to show the data updating in real time on the Serial Monitor
2. In the video, explain why this experiment is about EdgeAI and, in general, how the machine learning model is generated, deployed and flashed into the microcontroller.

References and Further Reading

[1] https://www.ibm.com/topics/artificial-intelligence
[2] https://www.edgeimpulse.com/
[3] https://docs.edgeimpulse.com/docs/edge-impulse-cli/cli-installation
[4] https://docs.edgeimpulse.com/docs/deployment/arduino-library