Experimenting with Neural Network Implementation on Embedded devices

Ashok Mohan, Electrical Engineering
Mentor: Dr. Chao Wang, Senior Lecturer
Ira A. Fulton Schools of Engineering

Introduction

- In the last few years, there has been a digital revolution with many Internet of Things (IoT) devices connected to the cloud.
- These IoT appliances, such as health monitors, and smart factory equipment, typically generate a lot of data that needs to be processed.
- These IoT devices typically have an embedded microcontroller
- Microcontrollers are limited in the computing and RAM capacity
- Simple AI/ML Neural Network models are implemented in these microcontrollers for detecting motion, voice, and images and taking action
- This research involves an experimental approach to develop an understanding of the constraints of training NN models to be deployed on an embedded microcontroller such as Arduino Nano 33 BLE Sense

Objective

- Collect and/or Obtain training datasets for two types of data - accelerometer and voice commands
- For each dataset, train an appropriate NN model to be deployed on Arduino Nano 33 BLE Sense
- Conduct experiments to determine accuracy, loss, inference time, and RAM usage as a function of the number of layers, number of neurons, and learning rate (hyperparameters)
- Document general observation of the sensitivity of the hyperparameters on the performance

Background

Neural Networks [1,2]

- Neural networks are a subset of machine learning at the core of Deep Learning Algorithms.
- Inspired by the human brain, mimicking the way biological neurons signal to each other
- NN consists of node layers: an input layer, one or more hidden layers, and an output layer.
- Each neuron in one layer connects to another neuron in the next layer and has associated weight and threshold
- If the output of any individual node is above the threshold, the node is activated, sending data to the next layer; Otherwise, no data is passed to the next layer.
- There are many types of NN networks - Convolution Neural Networks (CNN), Recurrent Neural Networks (RNN) etc. NN rely on training data to learn and improve their accuracy.

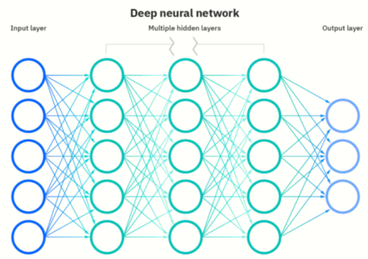

Figure 1: A Deep Neural Network Model

Machine Learning models (NN) can be trained to be deployed on an embedded microcontroller. The efficacy of the trained model is determined by performance metrics. A few of interest are:

- Accuracy - Defined as how the model performs across all classes. It is calculated as the ratio between the number of correct predictions to the total number of predictions
- Inference time - The amount of time it takes to do one forward propagation through a trained model. The inference time also depends on the precision used (e.g. Float32, Quant(Int8))
- Loss refers to the difference in prediction vs actual values
- RAM - The amount of memory that is needed for the trained model

The training model performance depends on hyperparameters. Hyperparameters are parameters whose values control the training process and determine the values of the model parameters during training. A validation set is used with the hyperparameters. A few of interest are:

- Learning Rate – A tuning parameter that determines the step size at each iteration while moving towards a minimum of a loss function. Represents the speed at which a machine learning model "learns"
- Training Cycles or Epochs - Defined as when an entire dataset is passed forward and backward through the neural network only once.
- Layers - The number of layers in the Neural Network or Convolution Neural network
- Neurons - The number of neurons per layer

Experimental Procedure

The NN model training was conducted on two separate datasets.

1) Accelerometer: The goal was to train a model to differentiate the vibrations of devices
2) Voice commands: The goal was to train a model to spot certain keywords

The trained models were then deployed on the Arduino Nano 33 BLE Sense for inference.

Data Acquisition - Vibrations

- Accelerometer data were gathered from three devices – 1) Food Blender 2) Vacuum Cleaner, and 3) Dryer
- Data was acquired using a smartphone and the Edge Impulse tool (Figure 2)
- The smartphone was attached to the devices
- Each device when in operation, a sample of 10 seconds was collected and recorded in Edge Impulse tool
- For each device, 20 samples were collected
- In all, a total of 60 samples were collected

Figure 2: Smartphone attached to Vacuum cleaner for data acquisition

Data Acquisition – Keyword Spotting

The goal of the trained model was to recognize the word "Go" and "Stop"

- This data was acquired from another Edge Impulse project[6]
- The dataset consists of audio recordings from three classes - GO, STOP, and Unknown
- The dataset consisted of
  - 2800+ samples of keyword "Go" totaling 48 min
  - 2800+ samples of keyword 'Stop" totaling 48 min
  - 3900+ samples of Unknown totaling 45 min

Training

The training procedure followed:

- Load the training data into the Edge Impulse tool
- Select the processing block (Figure 3)
  - Spectral Analysis - Analyzes repetitive motion such as data from accelerometers. Extracts frequency and power characteristics of a signal over time (Vibrations dataset)
  - Spectrogram for extracting spectrogram for audio or sensor data (keyword spotting dataset)
- Select the learning block (Figure 3)
  - Keras - the NN classifier (Vibrations dataset & keyword spotting)
- Train the model with different hyperparameters
- For each hyperparameter configuration, record accuracy, loss, inference time, and RAM usage with the validation set

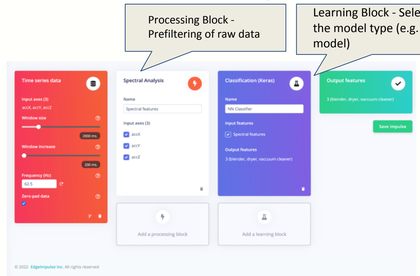

Figure 3: Edge Impulse Design for the Vibrations dataset

Results - Training

Figure 4

Figure 5

Vibrations Dataset

Figure 6

Figure 7

Keyword Spotting Dataset

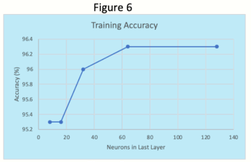

Figure 8

Figure 9

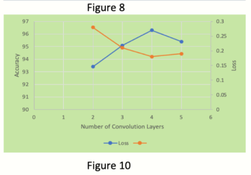

Figure 10

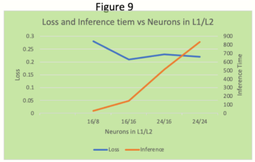

Figure 11

Precision	RAM Usage	Inference time
Quant(Int8)	16.5 KB	27 ms
Float32	51.6 KB	57 ms

Table 1: RAM Usage, Inference time vs Precision for Keyword Spotting trained dataset (Note: No difference in accuracy was observed)

I would like to thank my mentor, Dr. Wang for her guidance and support for this project

Inference

- The trained models for both the Vibrations and Keyword spotting were deployed on the Arduino Nano 33 BLE Sense for inference
- After the model was deployed, inference experiments were conducted for each class of the dataset
- A live inference experiment was conducted; 20 experiments per class for both the datasets; Average accuracy is reported below

Results - Inference

Voice Command	Accuracy
GO	97.6
STOP	99.6
UNKNOWN	99.6

Keyword Spotting Dataset

Accelerometer	Accuracy
Food Blender	92.0
Vacuum Cleaner	92.1
Dryer	93.1

Vibrations Dataset

Findings/Observations

- Learning rate determines the step size at each iteration
- When the learning rate is high, the loss gets stuck in an undesirable local minimum while the accuracy drops significantly. This is probably because of the underfitting of the model
- Increasing training cycles improves accuracy and reduces loss
- However, there is diminishing return as the training cycles are increased beyond a certain value
- Increasing layers and neurons do improve accuracy and reduce loss
- However, the inference time goes up
- One can tradeoff a slight decrease in accuracy if inference time is important
- The datasets trained were not memory constrained on the Arduino Nano 33 BLE sense
- Live Inference shows very good accuracy on the trained model

Future Work

- Conduct research on DNN implementation challenges on microcontrollers
- Conduct experimental research on more complex data sets for object detection and image recognition on Arduino 33 BLE Sense that stresses the limit of the memory
- Develop an optimized DNN model for the datasets

References

1. Sergio Branco , Andre G. Ferreira * and Jorge Cabral, Algoritmi Center, University of Minho, 4800-058 Guimar es, Portugal; "Machine Learning in Resource-Scarce Embedded Systems, FPGAs, and End-Devices: A Survey"
2. "What are Neural Networks"
   1. https://www.ibm.com/cloud/learn/neural-networks
3. "What is Convolutional Neural Network"
   1. https://www.i2tutorials.com/what-is-convolutional-neural-network-what-are-all-the-layers-used-in-it/
4. Learning Rate in Machine Learning - https://en.wikipedia.org/wiki/Learning_rate#:~:text=In%20machine%20learning%20and%20statistics,minimum%20of%20a%20loss%20function.
5. Edge Impulse tool
   1. https://studio.edgeimpulse.com/studio
6. Syntiant RC-Go-Stop Project
   1. https://studio.edgeimpulse.com/public/42868/latest

FURI

Ira A. Fulton Schools of Engineering
Arizona State University