Embedded Audio Classifier,
An Application of Deep Learning of Light Control

Student ID: 17014730

Candidate Number: LXWG7

Candidate Name: Yuge Wang

28-04-2021

Word count: 1096

Github link for Documentation: https://github.com/YugeWang/DeepLearning_final

Edge Impulse Link for the Project: https://studio.edgeimpulse.com/public/21959/latest

This report is submitted in fulfilment of the project for

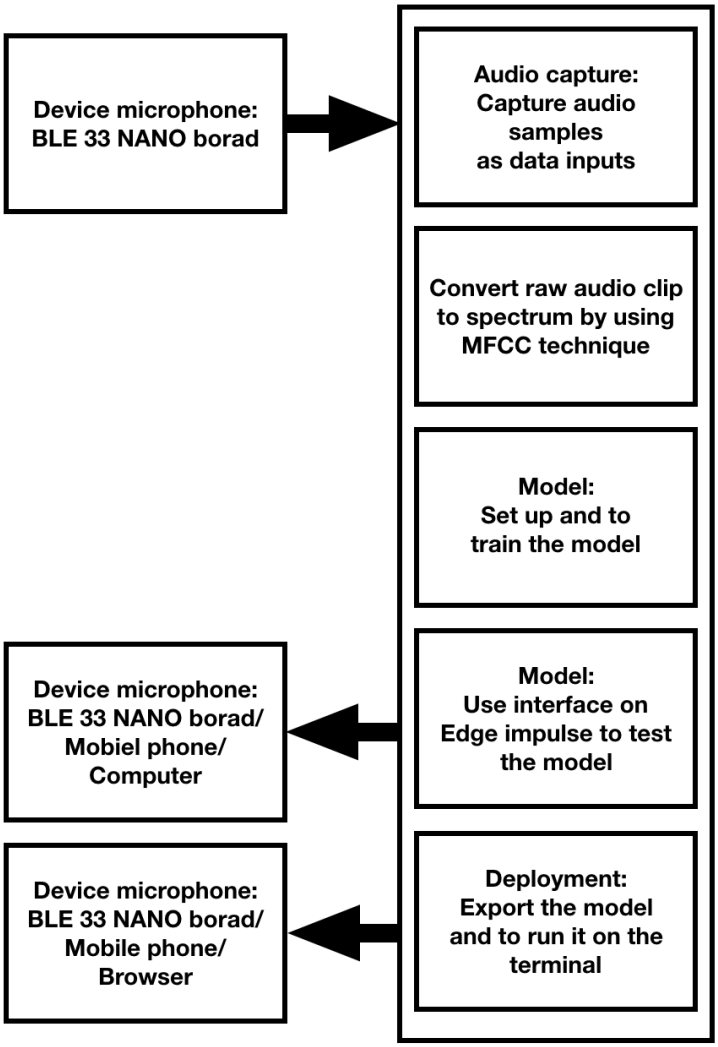
CASA0018:Deep Learning for Sensor Networks(20/21)

Application introduction

Inspired by a series of ‘smart home’ products in the current market, including voice control speakers, televisions and lights, as well as some voice-control user, interface like the Apple HomeKit which allows user to control all their smart devices at one time, ensuring a unified home environment.

This is an audio classification project that aims to identify five categories of words that would be used for a smart home device that can control the light switch and its brightness. The five categories are turn on, turn off, lighter, silence and noise which are the most commonly used command for a home-use light.

This report will discuss essential steps in detail, including data collection, model architecture, and lastly the model will be tested and to show its deployments on mobile phones and Arduino Nano boards. Most of the steps are completed using EdgeImpulse. And here is a flowchart for an overview of the application.



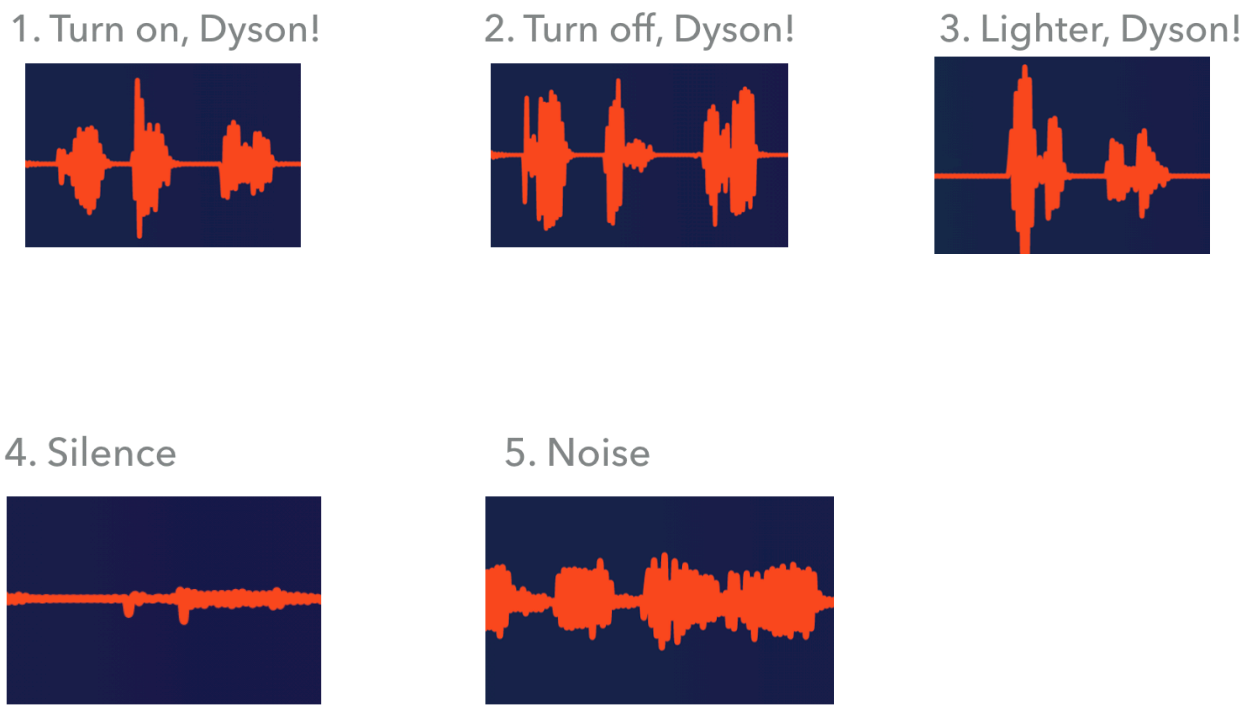
Graph.1 application flowchart

Data description

Data is collected using the BLE 33 nano board. Each audio clip is 5 seconds long and every five categories consist of 30 clips. In order to simulate the various environment in real life and to train the model better at recognising voice in a different situation. The “turn on”, “turn off” and “lighter” command recorded voice from both gender and various accent. And around 1/5 of the clip consist of clips with obvious background noise including dog’s barking, music and TV playing closely, etc. Similarly the background noise also recorded and gathered into the “noise” category. Most voice intelligence systems like google or Siri use a name as a wake-up word. Here to avoid mis-recognition, a name to the system called ‘Dyson’. I hoping the system will only respond when it hears the name and command together.

To make the audio input ready for model to use, firstly a spectrum was produced by using the technique of MFCC. a window of 3s and step forward at a rate of 10ms gave out the best accuracy. It moves to capture features which then being transformed into the spectrogram. And in the process of transformation, lower frequencies would have more resolutions which help with the noise reduction. And finally, we get the frequency bucket which holds features of audio input that is ready for use.

Here it shows the waveform of audio clips.



Graph.2 audio clips of five categories

Model Architecture

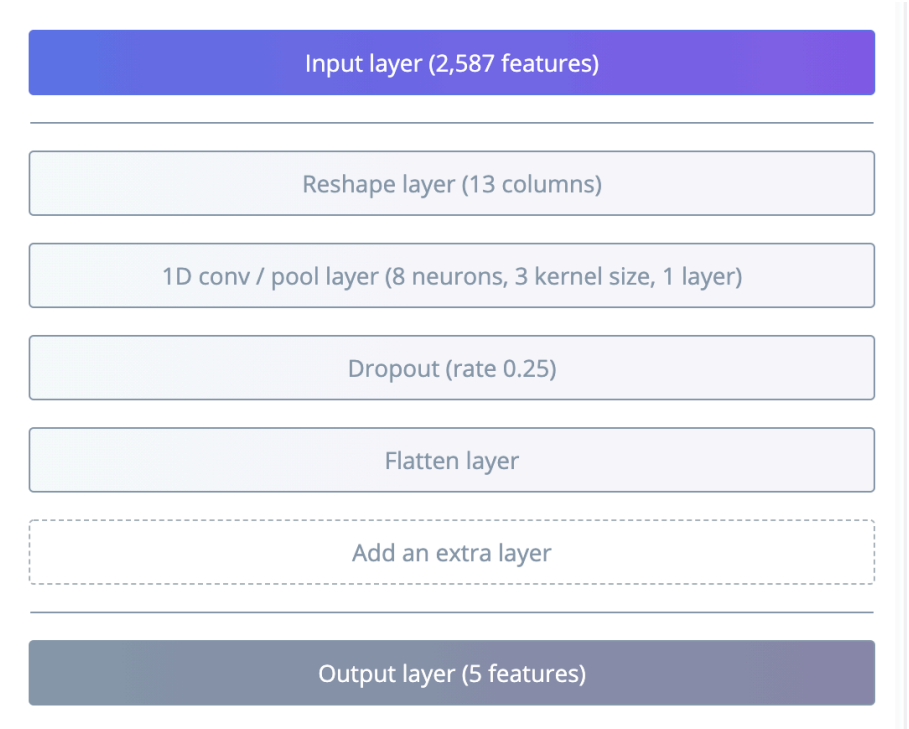
After a few attempt, the best combination of layers is described. This model used the Convolutional Nerone Network also known as the CNN method. There are mainly four types of layer used. So after the input and reshape layer, there are two 1D convolutional layers with 8 and 16 neurones respectively and each followed by a max-pooling layer that could capture the most active features. And lastly, a fully connected layer using softmax as an activation method to give out the probability of classifications of the audio.

The 2D convolutional layer was also tried but gave out much lower accuracy, it may fit better with more complex input. In 1D convolution, the filters move only one direction, that is, from left to right while in 2D convolution the filters move in two directions both left to right and top to bottom. And the 'softmax' activation method looks at all the probabilities in that layer of neurons and sets the highest value to 1 and all the others to 0. This makes it programmatically easier to find the most likely solution.

Experiment and Results

The performance of the model mainly evaluated based on classification accuracy that presented using a confusion matrix. And the loss will also be considered.

Two adjustments would be made. The first one is the size of the window which captures audio features and its window increase. The second is the model architecture. I will first set up an initial model of both and then make the adjustment to find out the best combination. The final model selects the better adjustment of the two options. The better one is filled with green colour. The initial model is as shown below:



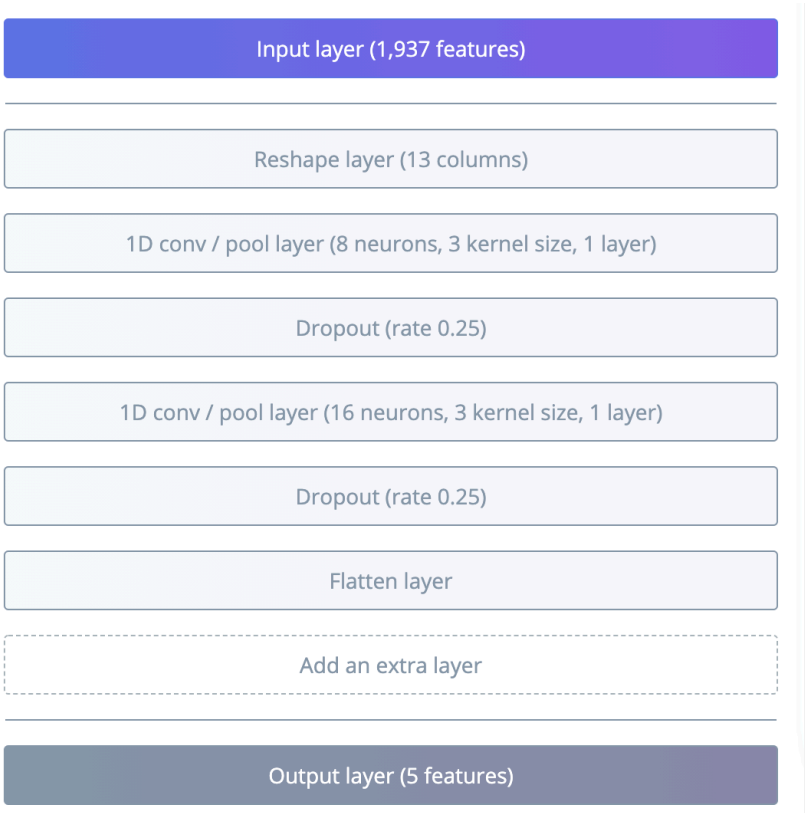
Graph.3 initial model architecture

	Adjustment	Accuracy	Loss
Window size and window increase	Window size is 3s and window increase is 1000ms	40%	1.60
	Window size is 3s and window increase is 100ms	83.5	0.51
Numbers of Neurons	One layer with 8 neurone	83.5%	0.51
	First layer 8 and second layer 16 neurone	96.7%	0.11
1D or 2D convolutional layers	1 Dimension	96.7%	0.11
	2 Dimension	83.3%	0.51

Table.1 comparison of model architecture with adjustments

Here in the table shows a summary of different combinations. The adjustment made based on the initial model. As shown in the table above, the best model is with a size window of 3 seconds and an increase with a 100ms window. And with two 1 dimension layers with 8 and 16 neurones respectively. It gives out an accuracy rate of 96.7%.

Here is the final model layer:



Graph.4 final version of the model

As shown in the matrix, there are a few mis-recognitions. For example, 4.4% of silence was classified as noise and 3.6% of turn on classified as turn off. But those mis-recognitions are much less essential as recognising turn off as turn on, or recognise silence as turn on and those are less than 1.5% probabilities.

	LIGHTER	NOISE	OFF	ON	SILEN...
LIGHTER	98.7%	1.3%	0%	0%	0%
NOISE	0%	98.3%	0%	1.7%	0%
OFF	1.5%	0%	97.0%	1.5%	0%
ON	0%	0%	3.6%	96.4%	0%
SILENCE	2.9%	4.4%	0%	0%	92.6%
F1 SCORE	0.97	0.96	0.97	0.96	0.96

Table. 2 confusion matrix of final model

Device deployment and observations

The model would be deployed on mobile devices and run independently without an internet connection. In this study, I have tried to deploy it on the mobile phone, computer interface, Arduino and terminal. Except for Arduino, the compelling process took a very long time, the other three platforms had successfully run the model and classified audio inputs. On the computer interface each audio need to be uploaded then classified. While on mobile phones and terminal, the data would be upload and classified continuously without user interventions. And the result will be in a format that indicates the probability of each sample to be classified into five categories. Below shows the confusion matrix for testing dataset. The accuracy compared with training model was lower but reasonable as I had captured testing audios data with more variations. The model is weaker in classifying ‘lighter’ and ‘turn on’ commands than others.

ACCURACY
78.77%

	LIGHTER	NOISE	OFF	ON	SILENCE	UNCERTAIN
LIGHTER	72.4%	0%	2.9%	2.9%	0%	21.9%
NOISE	2.9%	76.2%	0%	0%	0%	21.0%
OFF	1.9%	3.8%	81.0%	0%	0%	13.3%
ON	2.4%	1.2%	1.2%	76.2%	1.2%	17.9%
SILENCE	0%	0%	1.0%	0%	87.6%	11.4%

Table. 3 confusion matrix of testing dataset

Graph.5 live classification with variations

Here is a clip of some results. The test data here used considered different accent, gender and some of them with background noise. The result had shown a decent classification. And to point out that when replacing 'Dyson' with another word, like 'Mike' or 'Lisa', the model classified it as noise regardless of its 'turn off' command which is what I aimed designed for. However, if replacing 'Dyson' with other similar words, like 'Tyson', the command still works.

Bibliography

Situnayake, D. and Warden, P., 2019. *TinyML*. O'Reilly Media, Inc. Chapter 4 - Chapter 8. https://learning.oreilly.com/library/view/tinyml/9781492052036/.

Wilson, D. And Dejode, M.,(2021). Lecture Materials of CASA0018 - Deep Learning for Sensor Networks. Week 1 - Week 6. https://moodle.ucl.ac.uk/course/view.php?id=19367&section=0#tabs-tree-start

Declaration of Authorship

I, Yuge Wang, confirm that the work presented in this assessment is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Yuge Wang

28/04/2021