TINYML MODEL MINIMIZATION FOR ARDUINO NANO 33 BLE SENSE

Bradley Voracek, RET

***Abstract* – The Arduino Nano 33 BLE Sense board is used by the TinyML community for running TensorFlow algorithms. There is a limited amount of memory on the board, which means we need to minimize our model size in order to make on board predictions with a large ML algorithm. I generate a dataset of 100 gestures and try to find optimal features that maximize accuracy while minimizing model size. It is found that the skew of the slope of our data is a highly predictive feature, and can maintain perfect accuracy on my test set while having the model be 45kB, compared to the raw data model size of 911kB.**

***Index Terms*—Embedded ML, TinyML, gesture recognition, TensorFlow**

1. PROJECT DESCRIPTION

The TinyML community has optimized TensorFlowLite algorithms for the Arduino Nano 33 BLE Sense board [1]. This makes this board an ideal one to use for research on embedded machine learning. The board has 1MB of CPU Flash Memory, so our models and libraries all need to fit under that maximum limit. One of the basic examples on this board is the "Magic Wand" or gesture recognition example. While it is simple to generate a model small enough to predict between the two classes in the sample, having more gestures to select between can run into size issues. For this reason, my research tries to find optimal features for gesture classification, so that the size of the model is minimized while maintaining accuracy. With this research it becomes possible to classify between many different gestures without running into model size constraints on the Arduino board.

I started my research by collecting a dataset of 100 gestures, 50 punch and 50 flex gestures. I collected the xyz data on the accelerometer and the gyroscope from the board. The other sensors did not seem relevant to the gesture recognition problem. Each sample collected 119 data points of each of these 6 sensor values over a 1 second period. The baseline TinyML model was using this raw data to feed into the Keras Sequential Neural Network algorithm [2]. This generated a model that was 911kB in size, which could predict between two gestures with 100% accuracy on my test data, and performed extremely well on test gestures I did by hand on the board. My research goal then was to see how far I could decrease this model size while maintaining 100% accuracy.

I generated statistical features on each of the sensor values to see which ones could predict better than just purely raw data. I used the max, min, skew, kurtosis, sum, mean, standard deviation, and the variance. Initial testing with just these features did not yield ideal results, so I went a step further and took the difference between data points, effectively yielding the slope or derivative of the sensor data. I generated the same statistical features for this slope data as well, and then attempted to compute the best predictive features. Using the random forest, I generated feature importances of these 96 features [3].

The most predictive features were from the skew of the slope in the xy of the gyroscope, and z of the accelerometer. Variance was also highly ranked. Using the variance of the xyz of both sensors, and the skew of the slope of the xyz of both sensors, I was able to generate my best model which maintained perfect accuracy at only 45kB model size, over 10 times smaller than the raw data model. Figure 1 shows the different models I ran along with this size in bytes and their corresponding accuracies. Using just the best feature, the skew of the slope of the y gyroscope value, I found a lower bound for model size is 32kB.

This research shows that feature generation and selection can create a TinyML model for gesture recognition that predicts as well as the raw data while making the model over 10 times smaller. The skew of the slope of the xyz gyroscope and accelerometer are found to be highly important features. Future research should see if the skew of the slope generalizes to be predictive of more gestures than just the flex and punch gestures.

Model Name	Model Size	Accuracy	Misclasifications
Features & Raw Data	8,249,802	1	0
Raw Data Only	911,468	1	0
Delta raw data	911,468	1	0
Delta Features	149,268	0.9	2
1/10 Raw Data	112,268	1	0
Features Only	90,068	0.95	1
42 Selected Features	82,668	0.95	1
1/20 Raw Data	75,268	0.95	1
1/20 Slope Data	75,268	1	0
Hybrid	60,468	1	0
22/42 Selected	58,002	0.9	2
Max & Var & SlopeSkev	53,068	1	0
Max & Var	45,668	0.85	3
Var & SlopeSkew	45,668	1	0
Slope Min/Max	45,446	0.9	2
Top 10 Slope Features	43,202	0.8	4
Top 3 Overall Features	34,568	0.95	1
Top 3 Skew Features	34,568	0.85	3
gYSlopeSkew	32,102	0.9	2

Figure 1: Summary Results

References

[1] Warden, Petee. *TINYML: Machine Learning with Tensorflow on Arduino, and Ultra-Low Power Micro-Controllers.* O'REILLY MEDIA, 2020.

[2] Ketkar, Nikhil. "Introduction to keras." *Deep learning with Python*. Apress, Berkeley, CA, 2017. 97-111.

[3] Rogers, Jeremy, and Steve Gunn. "Identifying feature relevance using a random forest." *International Statistical and Optimization Perspectives Workshop" Subspace, Latent Structure and Feature Selection"*. Springer, Berlin, Heidelberg, 2005.