Sound Classification with CNN on Low Power Embedded platforms

Zurich University of Applied Sciences
School of Engineering
ISC Institute for Signal Processing and Wireless Communications

Simon Vogel, ZHAW

ISC Institute for Signal Processing and Wireless Communications

Abstract

In this project, we explored different low power embedded platforms concerning their efficiency by analyzing the clock-cycles needed to perform a given CNN-classification task. By using the new ARM CMSIS Neural-Net-Library, we implemented the CNN on a Cortex-M7 microcontroller, which offers DSP like performance and optimizations. For comparison purposes, we implemented the net on a standard Cortex-M0+ microcontroller and on the GAP8 processor, which uses 8 RISC-V cores to perform efficient signal processing and neural net tasks. The results showed that between the Cortex-M0+, M7 and GAP8 an efficiency increase of factor 100 can be achieved. This leads to reduction in energy consumption of the same order of magnitude, which is crucial for battery powered hearing instruments. The CNN performs audio scene classification by analyzing 32x32 mel-spectrogram data, using two convolutional and four fully connected layers. During the training phase, a large dataset providing data from 6 different scene-classes was used. Our best CNN with over 1 million parameters was able to classify audio samples with an accuracy of 92%. To allow real-time classification on a lower power embedded system, the net size was reduced to 6000 parameters what decreased the accuracy to 86%. This accuracy is still sufficient for the intended use in hearing instruments. Audio samples of 3 seconds length are classified within 16.5 milliseconds by the Cortex-M7. For further research, the benefits and implementation of RNN networks like LSTM structures could be analyzed on embedded systems.

Introduction

Sound classification is an important part of audio scene analysis. In this project sound classification was used to determine the environment that surrounds a wearer of a hearing instrument. By distinguishing between 6 different classes like speech, noise or driving in a car it is possible to adjust the behavior of the hearing instrument to the environment.
The sound data is converted in to Mel-spectrograms of 3 seconds length. The resulting data is then further analysed by a CNN. By using a spectrogram of 230 by 256 pixels and a CNN with 1 million parameters, it is possible to achieve a classification accuracy of 92% over all test samples. When filtering the classification results over a longer period of time, it possible to classify the test samples with an accuracy of almost 100%. However the number of 1 million parameters and 450 million multiplications needed by this network to classify a three-second audio sample, does not suit the computing capability of a low power embedded system. Therefore the input size and net size had to be reduced.

1 Architecture

On the embedded system the spectrogram size was reduced to 32 by 32 pixel. Figure 1 shows an overview of the preprocessing applied to the audio signal.

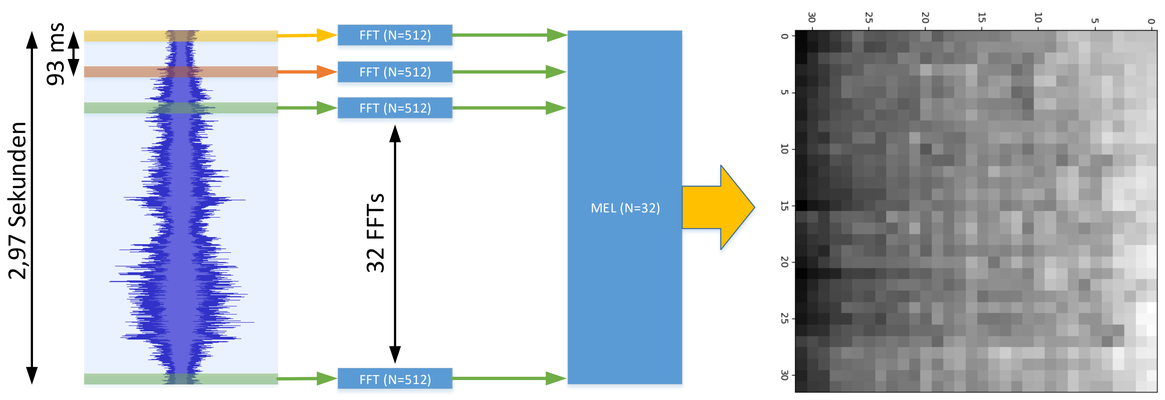

Figure 1: Preprocessing of a 3-second audio sample, resulting in a 32 x 32 spectrogram

The CNN was reduced to a network with 2 convolutional layers and 4 fully-connected layers as shown in figure 2. This network uses only 6000 parameters and 190 000 multiplications to classify three seconds of sound.

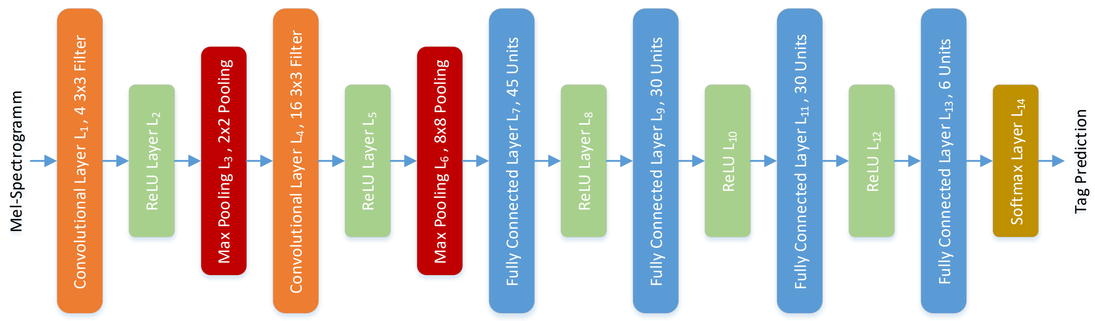

Figure 2: Structure of the implemented CNN

2 Performance

The performance of the CNN was evaluated directly on the Cortex-M7. Figure 3 shows the confusion matrix on sample basis on the left whereas the confusion matrix on the right shows the results based on the filtered labels. The filtered label corresponds to the label with the highest probability averaged over a full test file. The Cortex-M7 running at 200 MHz needs 16.5 ms to pre-process and classify an audio sample of 3 seconds length.

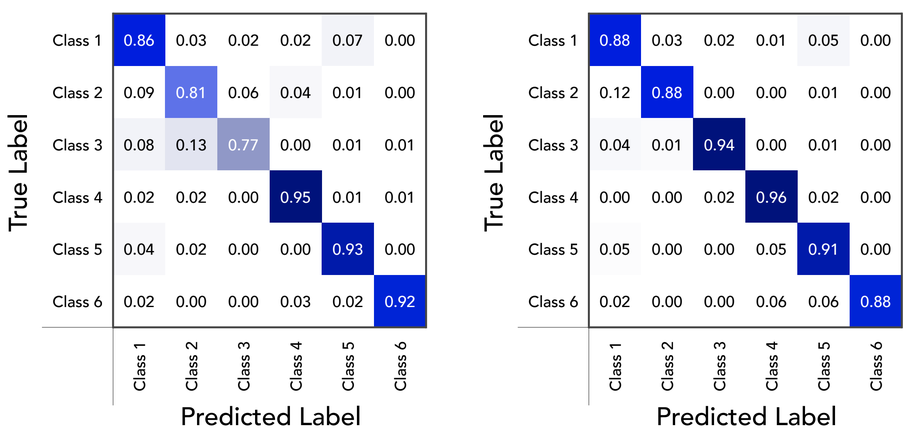

Figure 3: Confusion Matrix calculated on the Cortex-M7: Labels for 3-second samples on the left, Filtered labels on the right

3 CMSIS NN Library

The CMSIS NN-Library was used to implement the CNN on the Cortex-M7. This NN-Library is optimized for Cortex-M processors and makes use of the DSP and SIMD capabilities of the STM32F7. Figure 4 shows an overview of the available functions.

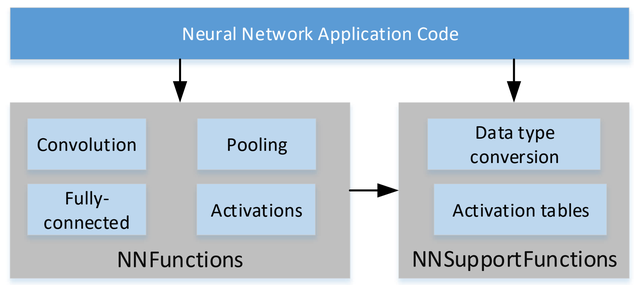

Figure 4: Overview of the CMSIS NN-Library

4 8-Bit Fixed Point vs Float

The NN-Library only uses 8-Bit and 16-Bit values for inference, allowing it to support all Cortex-M devices without using the FPU. The 8-Bit values are interpreted in fixed-point format, but with adjustable configuration of the fixed-point position in every layer. The fixed-point format of each layer is defined after training by simulating the quantisation effects on a computer. The confusion matrix stays nearly the same with 8-Bit inference and only small changes are noticeable. Figure 5 shows on the left an example of activation data after the first convolutional layer. The three-dimensional data is reduced to one dimension in this plot to show the quantisation effects. The first 200 activations values are saturated in the 8-bit format due to the quantisation. Figure 5 on the right shows the values of the last fully connected layer before the softmax function. Some differences between the float and 8-bit values are noticeable for the classes with low probability. The strongest class (2) gets nearly the same value in both implementations. This is a rather worst case example of the quantisation effect.

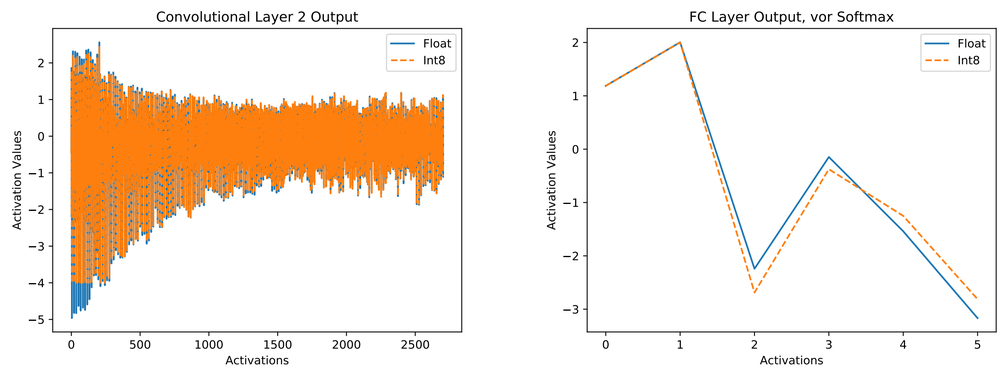

Figure 5: Comparison float and 8-Bit inference

5 GAP8-Processor

The GAP8-Processor is a new microcontroller, especially made for signal processing and machine learning. It features a cluster of 8 RISC-V cores with an instruction-set optimised for machine learning tasks. Figure 6 shows an overview of the internal structure of the GAP8.

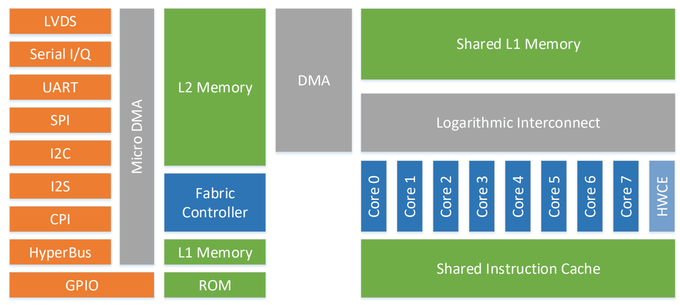

Figure 6: Overview of the GAP8 processor

6 Embedded platform Comparison

The CNN was also implemented on a Cortex-M0+ and the GAP8-Processor to compare the different embedded platforms. Figure 7 shows the clock cycles needed by the different embedded platforms to run one forward pass of the CNN. The GAP8 processor needs 100x less clock cycles to calculate the same result as the Cortex-M0+.

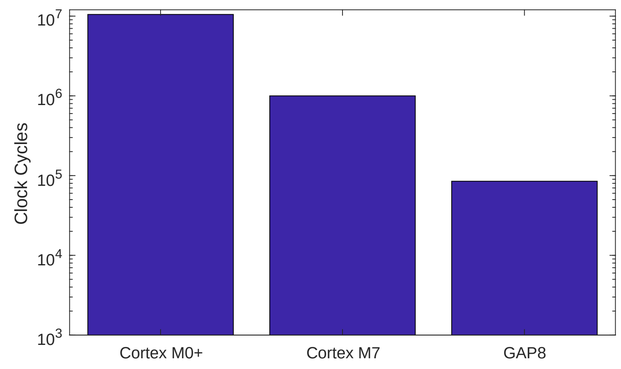

Figure 7: Comparison of the clock cycles needed for inference of 3 seconds audio data, on the different platforms

7 Conclusions

A CNN is well suited for acoustic scene classification. Further it is possible to implement a CNN of proper size on a low power embedded system. The most crucial limitation is the number of parameters and thereby the memory size. The computational power of the embedded system was sufficient for sound classification. The Cortex-M7 was able to classify a three-second audio sample in 16.5 milliseconds. Optimized instruction-set and processor structure can make a difference of factor 100 in calculation efficiency. Low power embedded systems with multiple cores further allow the usage of low clock frequencies to increase energy efficiency. Using 8-Bit values for inference gives comparable performance results to the floating implementation.