Memory-Optimal Direct Convolutions for Maximizing Classification Accuracy in Embedded Applications

Albert Gural, Boris Murmann

Introduction

In the age of Internet of Things (IoT), embedded devices ranging from ARM Cortex M0s with hundreds of KB of RAM to Arduinos with 2 KB RAM are expected to perform increasingly sophisticated classification tasks, such as voice and gesture recognition, activity tracking, and biometric security. While convolutional neural networks (CNNs), together with spectrogram preprocessing, are a natural solution to many of these classification tasks, storage of the network's activations often exceeds the hard memory constraints of embedded platforms.

In [1], a tree-based algorithm called "Bonsai" is proposed for performing classification on embedded devices. They suggest that CNNs are an inadequate solution, stating "Bonsai is not compared to deep convolutional neural networks as they have not yet been demonstrated to fit on such tiny IoT devices. . . . Implementing them on tiny microcontrollers is still an open research problem." They report on several classifiers and datasets, including a 2 KB classifier achieving 94.38% on 2-class MNIST and an 84 KB classifier achieving 97.01% accuracy on 10-class MNIST.

We suggest that dismissing CNNs for embedded applications due to memory constraints is premature and propose memory-optimal direct convolutions as a way to push classification accuracy as high as possible given strict hardware memory constraints at the expense of extra compute. We therefore explore the opposite end of the compute-memory trade-off curve from standard approaches that minimize latency at the expense of extra memory.

Technial Approach

The general observation that direct convolutions can be performed in a more memory-efficient manner compared to the "naive" approach of maintaining a separate area of memory for convolution outputs stems from the observation that convolutions are local operations, so pixels in input activations are computational dependencies of only a small number of output activations. Therefore, after an input pixel has satisfied all of its dependencies, the memory in that location can be deleted and used to store output pixels.

In more detail, activation memory is generally stored as shown at the top of Figure 1. When a convolution layer maps from a set of activations with more channels to one with the same or fewer channels, new activations can overwrite the old activations whose computational dependents are satisfied, as seen in the middle row of the Figure. However, in the more common case that the number of output channels increases, additional memory needs to be allocated, as seen in the bottom row of the Figure. For this latter case of $f_{out} > f_{in}$, we propose a novel method of processing output pixels, called "herringbone" (after the name of a tile that shares a similar traversal pattern), that we have proven is memory-optimal.

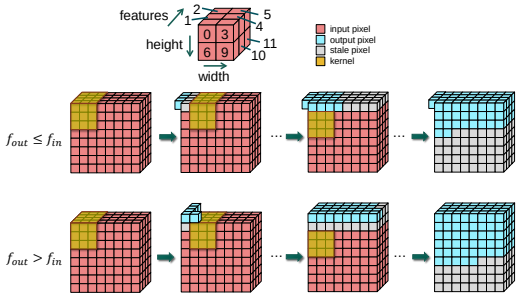

Fig. 1. Arrangement of pixels in memory (top) and placement of output activations based on how f_{in} compares to f_{out} (middle, bottom).

Results and Significance

We validate the memory-optimal CNN technique with an Arduino implementation of the 10-class MNIST classification task, fitting the network specification, weights, and activations entirely within 2 KB SRAM and achieving a state-of-the-art classification accuracy for small-scale embedded systems of 99.15%. For the neural network that achieves this accuracy, Figure 2 summarizes the impact of our herringbone method on required activation storage.

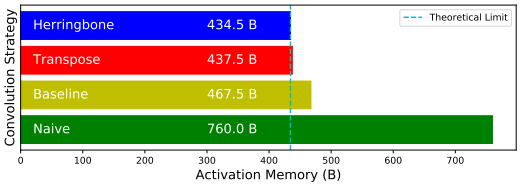

Fig. 2. Activation memory versus convolution strategy.

Herringbone is most helpful for smaller networks, but these are exactly the kinds of networks that would be deployed on tiny embedded devices which have the most to gain from memory improvements. Therefore, these results are a great match for the tinyML community when trying to implement 2D convolutions for classification.

References

[1] Kumar, Ashish, Saurabh Goyal, and Manik Varma. "Resource-efficient Machine Learning in 2 KB RAM for the Internet of Things." International Conference on Machine Learning. 2017.

A. Gural and B. Murmann are both from the Department of Electrical Engineering, Stanford University, Stanford, CA, 94305 USA