VIVIDSPARKS
CASE STUDY

POSIT™ Arithmetic for Autonomous Driving

We have seen how Posit is a suitable drop-in replacement for IEEE-754 standard, and its potentialities in autonomous driving applications. Achieved results when combining Posit arithmetic with DNN are promising in terms of tradeoff between accuracy and processing time.

Prof. Marco Cococcioni,
University of Pisa,
Italy

Introduction

The use of deep neural networks (DNNs) as a general tool for signal and data processing is increasing in automotive industry. Due to strict constrains in terms of latency, dependability and security of autonomous driving, machine perception (i.e. detection or decisions tasks) based on DNN can not be implemented relying on a remote cloud access. These tasks must be performed in real-time on embedded systems on-board the vehicle, particularly for the inference phase (considering the use of DNNs pre-trained during an off-line step). When developing a DNN computing platform, the choice of the computing arithmetics matters. Moreover, functional safe applications like autonomous driving pose severe constraints on the effect that signal processing accuracy has on final rate of wrong detection/decisions.

Challenges

Autonomous driving is a safety critical application, as specified also in functional safety standard like ISO26262, with strict requirements in terms of real time (both throughput and latency). The effort in computing these artificial intelligence algorithms is an open challenge in the field of computing platforms nowadays. In particular, when considering strict requirements, such as lowering the power consumption, maximizing the throughput and minimizing the latency the computational complexity becomes more and more critical. Moreover, with the modern achievements in sensor components, the complexity and requirements further scale with data coming in higher volumes and dimensions and at higher speed.

Approach

We assemble a custom DNN architecture that is parametrized by data width, data type, and DNN hyper parameters (e.g. number of layers, neurons per layer, etc.), as shown in Figure. 1. Each layer contains dedicated EMAC units with local memory blocks for weights and biases. Storing DNN parameters in this manner minimizes latency by avoiding off-chip memory accesses. The compute cycle of each layer is triggered when its directly preceding layer has terminated computation for an input. This flow performs inference in a parallel streaming fashion. The ReLU activation is used throughout the network, except for the affine readout layer. A main control unit controls the flow of input data and activations throughout the network using a finite state machine.

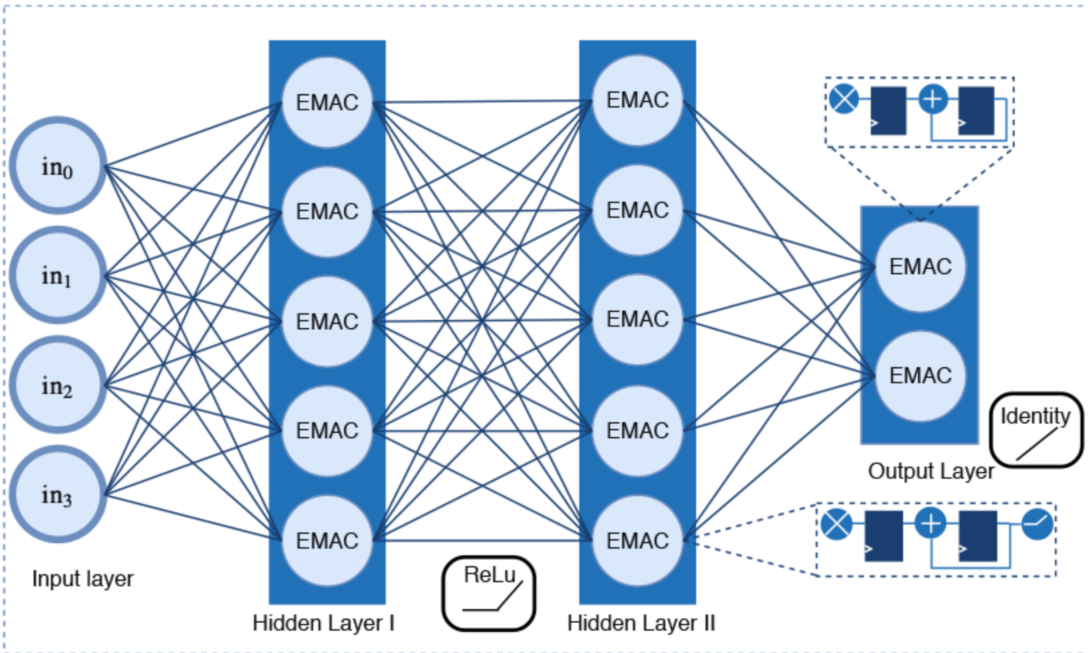

Figure 1: An overview of a simple Deep Positron architecture embedded with the exact multiply-and-accumulate blocks (EMACs).

We have considered different standard datasets, like the one shown in Figure 2, and standard CNN architectures, like the one shown in Figure 3. In particular, for the MNIST and German Traffic Road Sign Recognition Benchmark (GTSRB) benchmarks we trained customized CNN variants of that reported in Figure 3, including Posit related optimizations to convolutional and activation layers. For the Fashion-MNIST benchmark we used a pre-trained model with starting accuracy of 95%. For CIFAR-10 we used VGG16 pre-trained model. All the networks were initially trained using Float32 and then tested on the corresponding test sets, converting each Float32 trained model using different Posit configurations. Furthermore, in order to provide a fair timing-accuracy trade-off comparison, the Float32 model has been tested exploiting the SoftFloat library for software-emulated floating point numbers.

Figure 2: GTSRB.

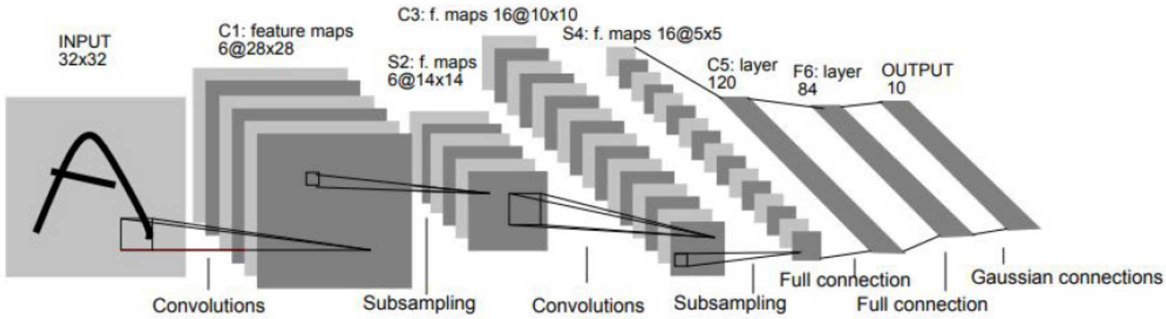

Figure 3: LeNet5 architecture.

Benefits

Table 1 presents the results obtained on three well known classification benchmarks: MNIST, Fashion-MNIST and CIFAR-10. MNIST is a digit recognition problem, while Fashion-MNIST has been designed as more complex drop in replacement for the MNIST dataset, providing more general classes to be recognized (such as fashion products). Furthermore, CIFAR-10 consists in an even more complex task, bringing 3-channel images in the dataset. As reported the tests on the model with the different types show that Posits with zero exponent bits and sized from 12 to 14 bits can be a perfect, replacement for Float32, while with 10 and 8 bits can replace Float32 with some drop in accuracy. The same holds for the Fashion-MNIST dataset.

The GTSRB is a baseline benchmark for road sign recognition, being very interesting as automotive task. Table 2 shows that also in this case Posits from 12 to 16 bits and even 10 bits can be a perfect replacement for Float32 while Posit 8,0 performs good with a little drop in accuracy.

The *k*-Nearest Neighbours (k-NN) algorithm is ubiquitous in pattern recognition problems. It can be used to segment images, or to compute the normal vectors to each point of a point cloud obtained by a lidar sensor mounted on a car. The *k*-NN algorithm algorithms finds the K nearest neighbours of a given point, from those in a given dataset. We have compared the performance of the *k*-NN when using Posits and Floats and, again, we have found that the accuracy of a Posit16,0 is very close to that of Float32 (see Figure. 4), and that a Posit8,0 outperforms a Float16.

Type	MNIST			Fashion-MNIST			CIFAR-10		
	Acc. (%)	Time (ms)	NCT[1]	Acc. (%)	Time (ms)	NCT[1]	Acc. (%)	Time (s)	NCT[1]
SoftFloat32	99.4%	8.8	–	95.0%	41.9	–	93.75%	7.75	–
Posit16,0	99.4%	5.2	0.59	95.0%	13.6	0.32	93.75%	2.55	0.32
Posit14,0	99.4%	4.6	0.52	95.0%	13.5	0.32	93.75%	2.49	0.32
Posit12,0	99.4%	4.6	0.52	95.0%	13.5	0.32	93.75%	2.44	0.31
Posit10,0	99.3%	4.6	0.52	95.0%	13.4	0.32	93.75%	2.40	0.30
Posit8,0	98.5%	3.8	0.43	94.0%	13.4	0.32	85.0%	2.34	0.30

Table 1: Accuracy and processing time obtained on MNIST, Fashion-MNIST and CIFAR-10 datasets. Processing time is evaluated as the mean per-sample inference time on the testset of the relative dataset.

Type	GTRSB		
	Acc. (%)	Time (ms)	NCT[1]
SoftFloat32	94.0%	15.86	–
Posit16,0	94.0%	6.37	0.40
Posit14,0	94.0%	5.21	0.32
Posit12,0	94.0%	5.08	0.32
Posit10,0	94.0%	5.0	0.31
Posit8,0	93.8%	4.0	0.25

Table 2: Accuracy-processing time trade-off obtained on the GTRSB dataset.

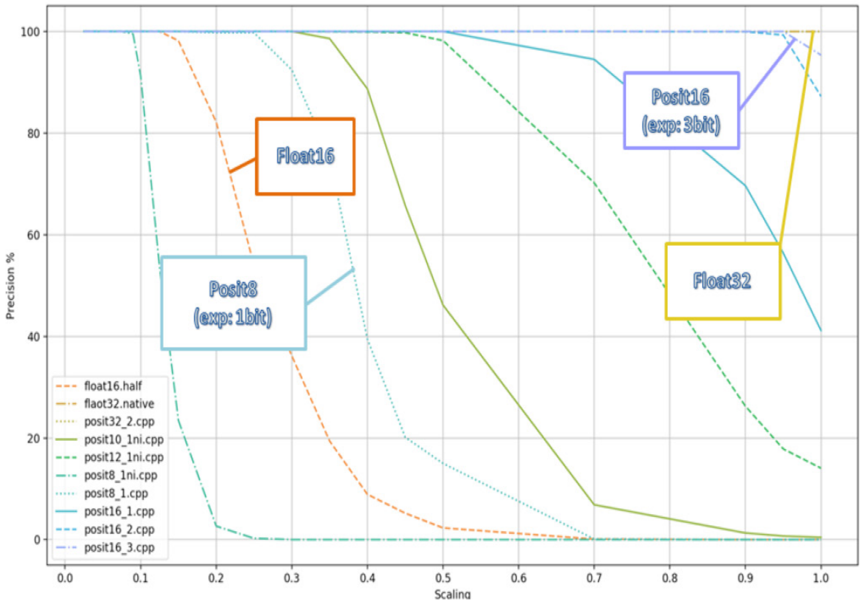

Figure 4: Performance of the k-NN using different data types, on a single dataset using different values for the scaling factor.

3

University Description

The Department of Information Engineering (DII) of the University of Pisa is a center of excellence for research in the field of Information Technologies (ICT), Robotics and Bioengineering. DII has promoted six spinoffs, and collaborates with public and private institutions to provide innovative solutions in key ICT fields, and to bridge the gap between academic and industrial research. The Department is involved in around 20 European projects (around 50 in the last three years), of which 6 coordinated by DII teachers, 3 ERC Grants and 25 regional projects, as well as 25 commercial projects from companies and almost 700 Collaborations with companies.