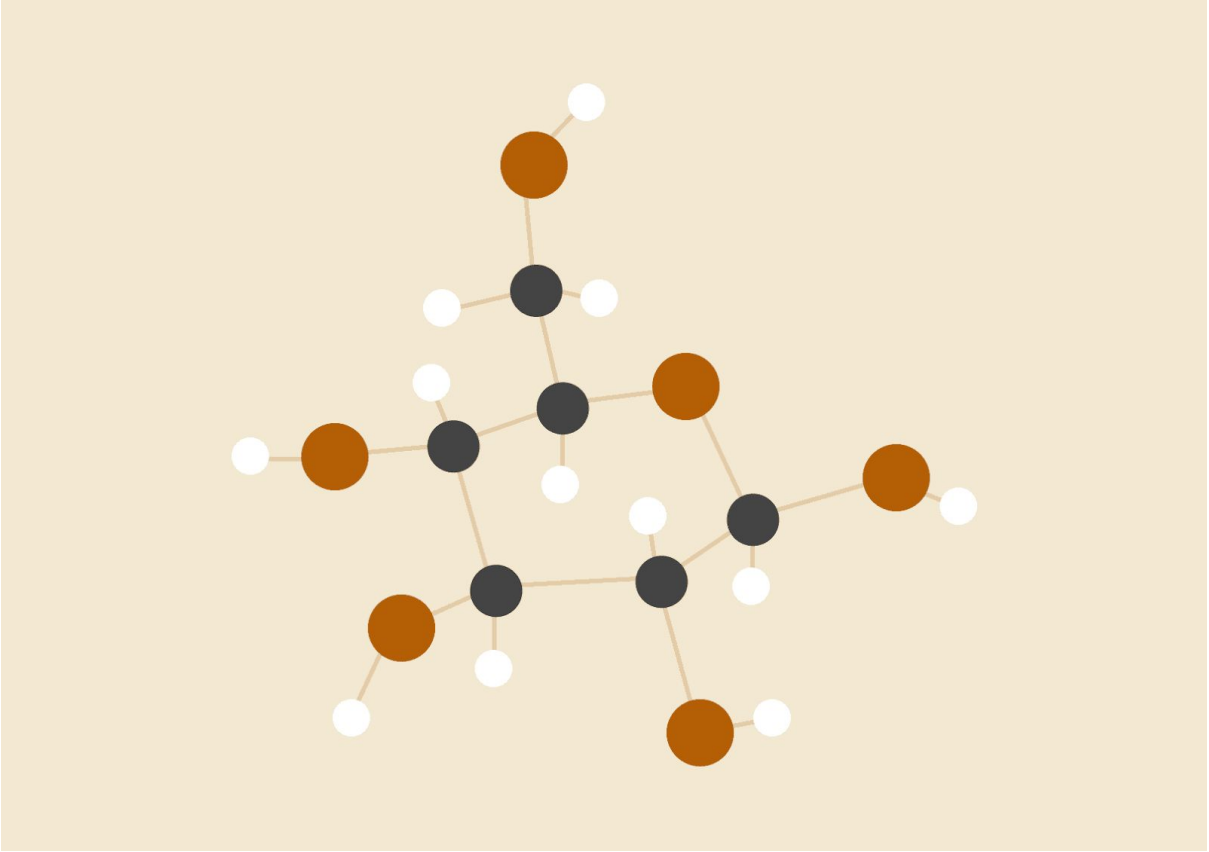
LAB 1 REPORT

Developing a neural network accelerator



Jannis Wolf - Julien Labarre

07.10.2020

PROCESSOR DESIGN

1. Selection criteria definition

Let's start with a quick project description. The goal of the project is to develop a neural network accelerator. Many design questions like the machine learning task and its corresponding network topology are still open and will get more concrete in the progression of the lab sessions. But it is safe to say, the accelerator should speed up the inference step of a neural network.

A major component which makes the development of accelerators time consuming is the communication interface of the software and hardware components. The TASTE framework (Taste) will be used to automate this. If the type of data that the accelerator expects from the CPU is specified, then the TASTE tool automatically generates the skeleton of the VHDL code for the accelerator, the implementation of the VHDL code which takes care of communicating with the CPU, as well as the implementation of the C code that will be used in the CPU to communicate with the accelerator. Figure 1 briefly describes the different parts and interfaces. First a feature vector has to be loaded on the software side. This data has to be sent to the accelerator (1), which is doing an inference step. Afterwards, the output of the neural net (2) is sent back and is further processed in the C Code on the CPU.

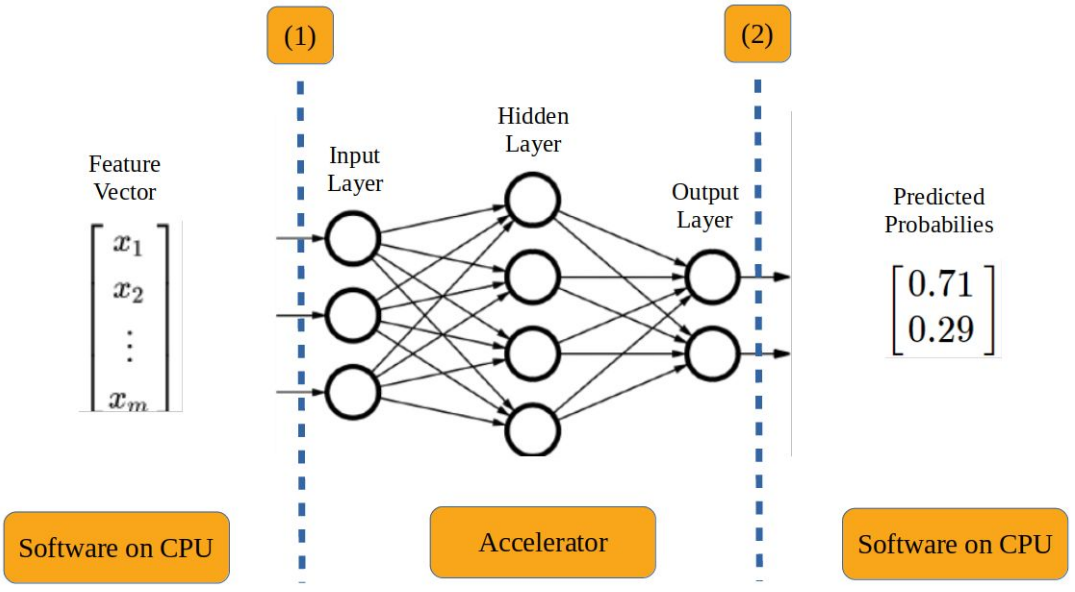


Figure 1: The communication between the software and the hardware (accelerator) side of the project. A feature vector is loaded on the software side. The TASTE toolchain provides the communication from the CPU and the accelerator in (1) and (2) before and after the inference step. The result is processed again on the CPU.

The project stands and falls with a good integration of the TASTE toolchain, so the focus in the development can be set on the accelerator. This sets

So in conclusion, the following selection criterias for the baseline processor have emerged:

- compatibility with the TASTE framework
- synthesis with the tools from Xilinx and Altera
- lightweight to fit on an FPGA together with the accelerator
- good documentation and availability of help from developers or community
- open source code in VHDL

2. Comparison of different designs

The TASTE tools are highly supporting the LEON 4 processor design (Gaisler). The LEON4 is a synthesizable VHDL model of a 32-bit processor compliant with the SPARC V8 architecture. The model is highly configurable. The LEON4 processor can also be synthesized with Xilinx and Altera. The core area (pipeline, cache controllers and mul/div units) requires only 30 kgates or 4000 LUT, depending on the configuration.

Gaisler has a second VHDL synthesizable model of a processor using the Risc-V architecture called NOEL.

3. Table summarizing the comparison (design VS criteria)

CRITERIA	DESIGN: Leon 4
Compatibility with the TASTE framework	
Synthesis with common tools	Xilinx, Altera and many more
Lightweight to fit on an FPGA	the core area needs only 30 kgates or 4000 LUT
good documentation and availability of help from developers or community	Mailing list from the Taste developers, documentation from the vendor Gaisler

open source code in VHDL	✔

4. Final decision on CPU used and justification

We decided for the LEON 4 model as a baseline processor. As we want to develop an accelerator the choice of the baseline model is strongly tied to the toolchain we want to use. Taste offers great support, tutorials and documentation for LEON 4. The model is lightweight and written in VHDL code. Projects we found on the internet, referenced here (dicear) , show that previous implementations of neural network accelerators can become quite big, because of the functional aspects of neural networks with the weights of fully connected layers.

5. References

dicear. *Neuron-VHDL*, https://github.com/dicearr/neuron-vhdl. Accessed 07 10 2020.

Gaisler. *LEON 4 Specifications*,

https://www.gaisler.com/index.php/products/processors/leon4. Accessed 7 10 2020.

Taste. *Taste Framework Website*, https://taste.tools/. Accessed 7 10 2020.