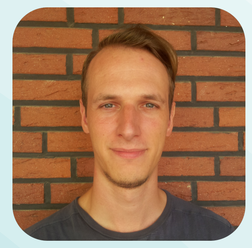
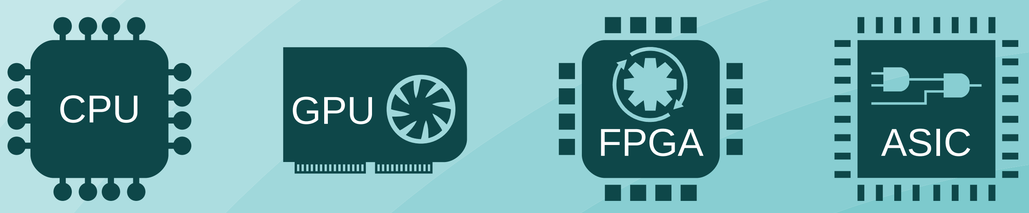
High-Level Synthesis of Neural Networks for FPGAs

Christof Schlaak – Andrej Ivanis – Christophe Dubach

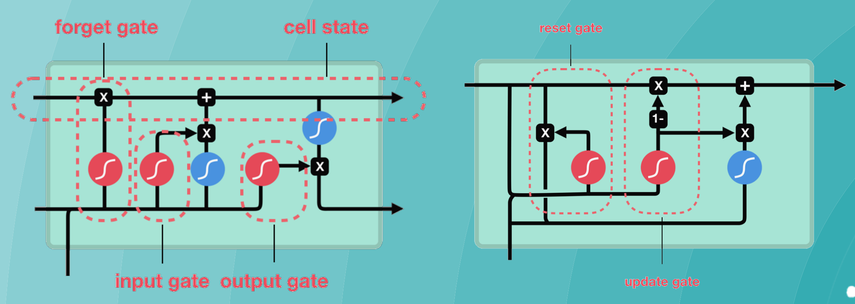


Motivation

- Hardware platforms for NN Accelerators:



- FPGAs are **reconfigurable**
  - ➡ can exploit different types of NNs
  - ➡ can adapt to evolving NN implementations



Problem

- FPGAs are not easy to use:
  - Require **hardware design expertise**
  - Require use of low level hardware language
  - **Steep learning curve** for tools
- Workflow is **not portable**

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use work.common.all;

architecture behavioral of add is
begin

    process (clk)
    begin
        if rising_edge(clk) then
            data_out <= std_logic_vector(
                    unsigned(data_in_1) +
                    unsigned(data_in_2));
            data_out_valid <= data_in_1_valid and
                    data_in_2_valid;
            data_in_1_ready <= data_out_ready and
                    data_in_1_valid and
                    data_in_2_valid;
            data_in_2_ready <= data_out_ready and
                    data_in_1_valid and
                    data_in_2_valid;
        end if;
    end process;

end behavioral;
```

Existing solutions

- OpenCL
  - Outperformed by hand-written HDL
- High-level tools provided by vendors (e.g. Xilinx SDAccel)
  - Not as flexible as HDL
  - May not support upcoming NN architectures
- HDL generation based on pre-built RTL components
  - Not flexible enough

The Lift approach

- Specify behaviour in a **high-level functional language**
- Optimise using **rewrite rules**
  - On algorithmic level &
  - On hardware-specific level
- **Generate hardware** implementation
- Estimate design quality using a **performance model**
  - Feedback results into new design generation

```
map( λ arow .
   map( λ bcol .
      reduce( +, 0 ) ∘
         map( X ) ∘
         zip( arow, bcol )
      , transpose( B )
   )
   , A
)
```

Advantages

- Target CPU, GPU and FPGAs
- Support arbitrary NN architectures
- **Portable** across many FPGAs
- Automatically **optimised**

Microsoft Research

LIFT
lift-project.org

THE UNIVERSITY of EDINBURGH
informatics