Hochschule
Bonn-Rhein-Sieg

Digital Design

NN_RGB_FPGA Exercises

Prof. Dr. Marco Winzker

(Version: 04.01.2021)

Overview

Lecture Videos

- Machine Learning on FPGAs:
  1. Introduction
  2. Training the Neural Network
  3. Circuit Architecture and FPGA Implementation
  4. Advanced VHDL Implementation
- Video Streaming for FPGA Remote Lab

Exercises

- Improve training of neural network
- Octave verification
- VHDL simulation with testbench
- Increase clock frequency
- Sigmoid function

GitHub

- NN_RGB_FPGA
  - FPGA_plain corresponds to lecture 3
  - FPGA_generate corresponds to lecture 4

Remote Lab https://www.h-brs.de/fpga-vision-lab

Exercise: Improve Training of Neural Network

Task

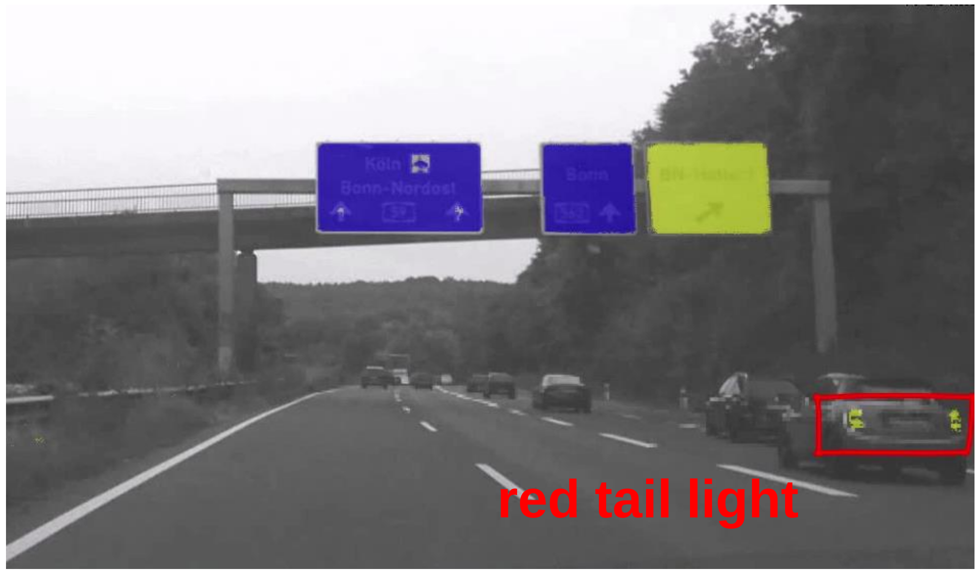
- Color detection should be improved
  - Red tail lights are detected as yellow
  - Green vegetation is detected as yellow
  - Light blue sky and dark blue signs are not distinguished

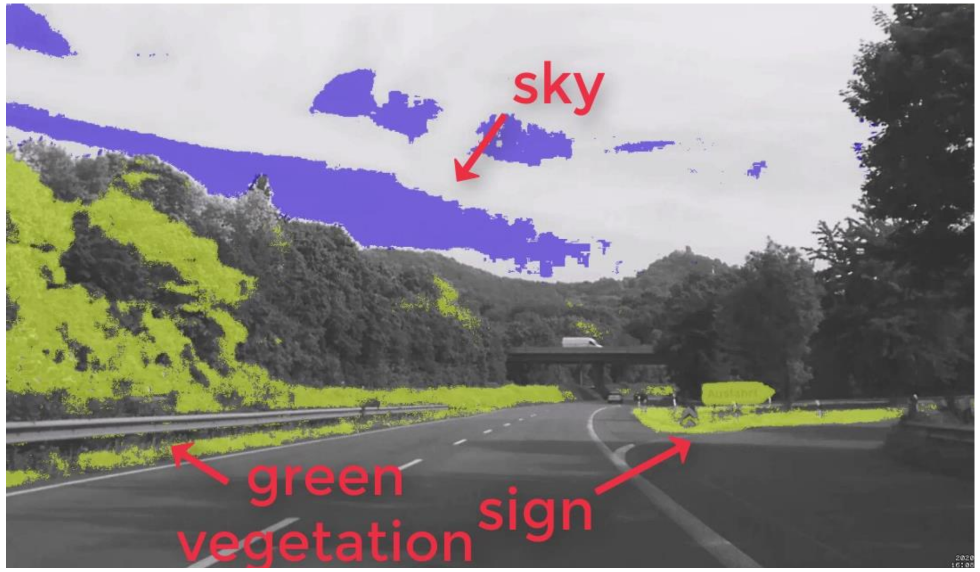
Approach

- Modify training data
  - “False yellow”: More red and light green
  - Labeling of blue and yellow without black/white letters

Alternative Task

- Training for detecting of green vegetation

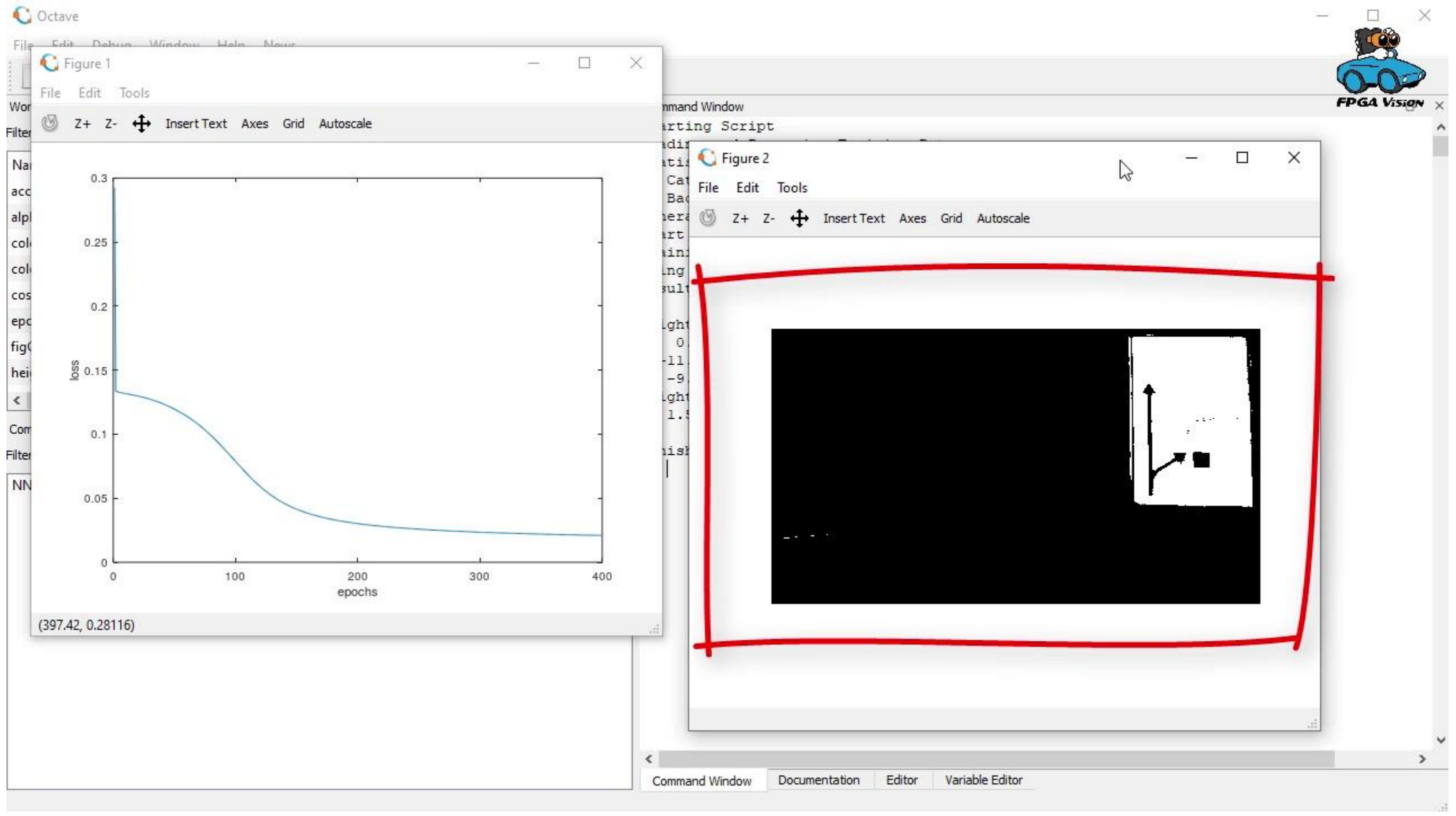




Exercise: Octave Verification

Task

- Octave uses training image to visualize training result
- Use different image



Task

- Octave uses floating-point accuracy for calculation of output image
- Use fixed-point accuracy to match FPGA implementation

Exercise: VHDL Simulation with Testbench

Task

- Perform VHDL simulation with a testbench
  - Template and explanations in video **“FPGA FIR Filter: Verification with VHDL Testbench”**
- Output images of testbench use no compression
  - They can be used to compare different circuit implementations

Task

- Extend to a self-checking testbench
  - Template and explanations in video **“FPGA FIR Filter: Self-Checking Testbench”**
- Reference image for expected results
  - Bit-true Octave implementation
  - Compare different VHDL versions

Exercise: Increase Clock Frequency

Task

- FPGA design has timing requirement for 720p video, i.e. 74.25 MHz
- Modify design for higher throughput, e.g. 200 MHz
  - Investigate capability of design
  - Remote lab will not require this throughput

Approach

- Set timing requirement in nn_rgb.sdc
  - create_clock … -period 13.47ns …   ← set to 5ns for 200 MHz
- Check “Timing Analysis” in Quartus
  - Add pipeline stages in design
  - Increase effort of synthesis in Quartus: Compiler Settings “Performance”
- Compare FPGA resources and power consumption in remote lab

Note: Additional pipeline stages require adjustment of “delay” for submodule control

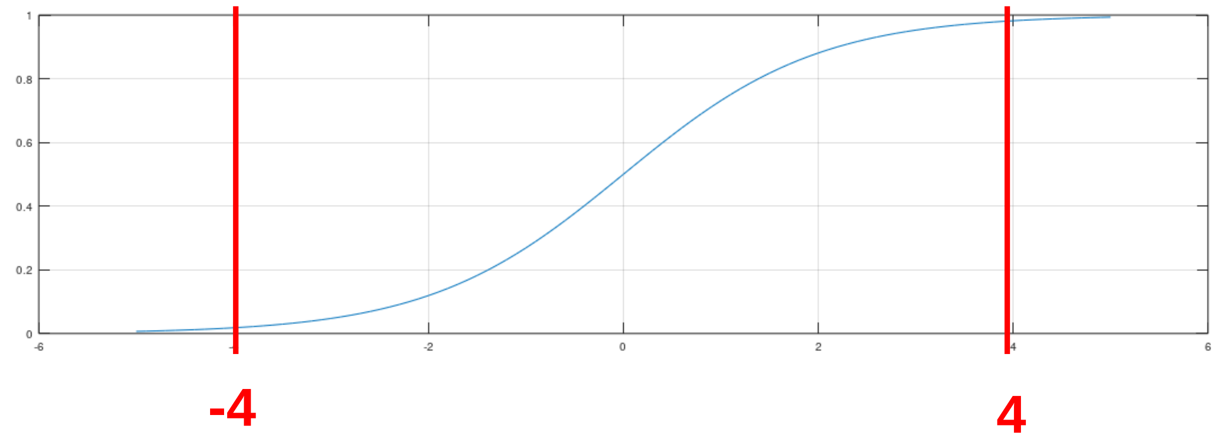
Sigmoid Function

Function of neuron

- z is factors times input values plus bias

  z = w1 * x1 + w2 * x2 + w3 * x3 + bias

- Sigmoid: $h = \dfrac{1}{1 + e^{-z}}$



Implementation

- Function table in ROM
- Values between ±4
- Limitation of values outside this range

Fixed-point implementation

- Values have factor of 2^13 = 8K (8192)
- Value ±4 corresponds to ±32K

Approach

5 binary places for w1, w2, w3
$\rightarrow$ shift parameter by 5 bit

input x1, x2, x3 have 8 bit, i.e. 0 to 255,
but correspond to values of 0 to 1

$\rightarrow$ must be considered for bias
$\rightarrow$ shift bias by another 8 bit

FPGA Implementation of Sigmoid Function

neuron.vhd in FPGA_plain

- Sum is limited to ±4 and shifted to positive range

  0 ≤ limit(sum+4) < 8
- Factor of 8K

  0 ≤ sumAdress < 64K

```
process
begin
    wait until rising_edge(clk);

    -- sum of input with factors and bias
    sum <= (w1 * x1 + w2 * x2 + w3 * x3 + bias);

    -- limiting and invoking ROM for sigmoid
    if (sum < -32768) then
        sumAdress <= (others => '0');
    elsif (sum > 32767) then
        sumAdress <= (others => '1');
    else
        sumAdress <= std_logic_vector(to_unsigned(sum + 32768, 16));
    end if;
end process;

sigmoid : entity work.sigmoid_IP
    port map (clock   => clk,
              address => sumAdress(15 downto 2),
              q       => afterActivation);

    -- format conversion
    output <= to_integer(unsigned(afterActivation));
```

Word width of ROM

- sumAdress has 16 bit
- ROM uses 14 bit

  address => sumAdress(15 downto 2)

- Other word width can be chosen
- Design FPGA_generate uses 12 bit

```
address => sumAdress(15 downto 4),
```

Definition of ROM Values

- ROM implemented as IP (Intellectual Property)
- Values defined in MIF (Memory Initialization File)

Definition of fixed-point values

- Input values corresponds to ±4 as discussed
- Output values correspond to range 0 … 1 with 8 bit word width
  - Range of 0 … 255
  - Same accuracy as input values x1, x2, x3

sigmoid_14_bit.mif

```
DEPTH = 16384;
WIDTH = 8;
ADDRESS_RADIX = DEC;
DATA_RADIX = DEC;
CONTENT
BEGIN
0 : 2;
1 : 2;
2 : 2;
3 : 2;
4 : 2;
5 : 2;
…
8000 : 121;
…
```

sigmoid_12_bit.mif

```
DEPTH = 4096;
WIDTH = 8;
ADDRESS_RADIX = DEC;
DATA_RADIX = DEC;
CONTENT
BEGIN
0 : 2;
1 : 2;
2 : 2;
…
2000 : 121;
…
```

ROM values for different word width

- 12 bit input values correspond to 14 bit values divided by 4

Exercise: Sigmoid Function

Task

- Implement ROMs with different input word width
  - Provided: 12, 13, 14, 16 Bit
  - Generate other word width, e.g. 8, 9, 10, 11 Bit
- Compare FPGA resources, power consumption and quality of color detection in remote lab
  - Please check: ROM size smaller than FPGA modules do not save resources

Task

- Select word width of ROM with generic parameter
  - Choose ROM with VHDL command „if-generate“
  - Change ROM to case-statement
  - Check functionality and resources for different word widths
- Choose parameter at top-level and forward to neuron and sigmoid function

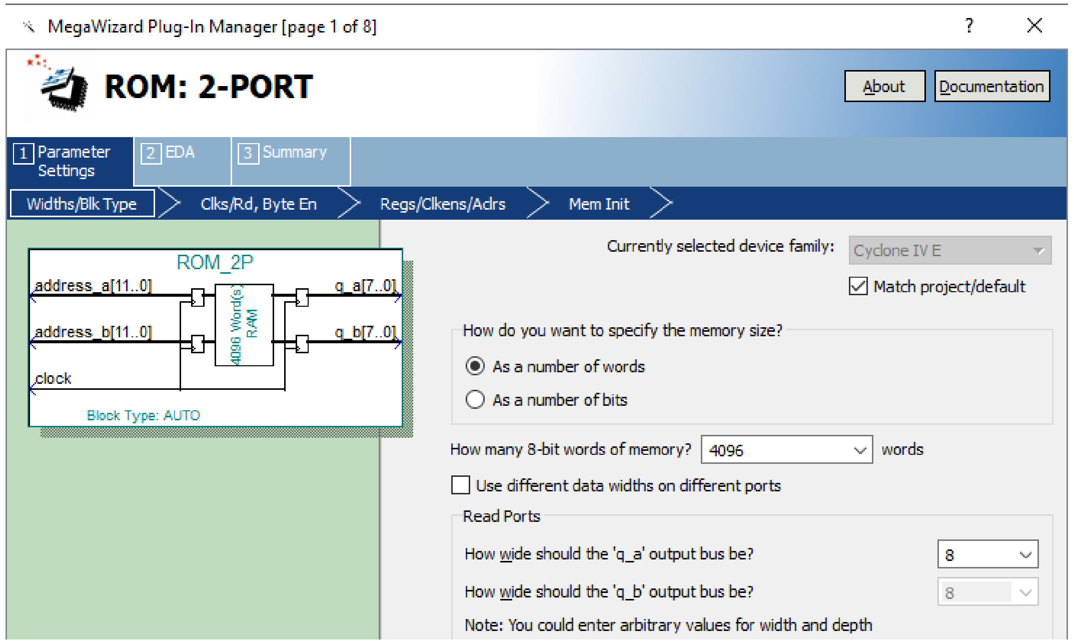
Exercise: Sigmoid Function (II)

Task

- FPGA RAMs have two ports
- All ROMs have identical content
  - Implement two sigmoid functions in one ROM
  - IP module “ROM: 2-PORT”

Approach

- Design new submodule with two neurons



Currently selected device family: Cyclone IV E

[✓] Match project/default

How do you want to specify the memory size?

(●) As a number of words

( ) As a number of bits

How many 8-bit words of memory? 4096 words

[ ] Use different data widths on different ports

Read Ports

How wide should the 'q_a' output bus be? 8

How wide should the 'q_b' output bus be? 8

Note: You could enter arbitrary values for width and depth

Exercise: Sigmoid Function (III)

Task

- Color mapping for output processing does not require sigmoid function
  - FPGA_plain: check if value > 127
  - FPGA_generate: check if value > 127 and
    compare values for different colors (yellow, blue)

Approach

- Output layer can use simplified function
  - output <= sumAdress(15 downto 8);
  - Can be selected by generic value parameter: output_layer (true/false)