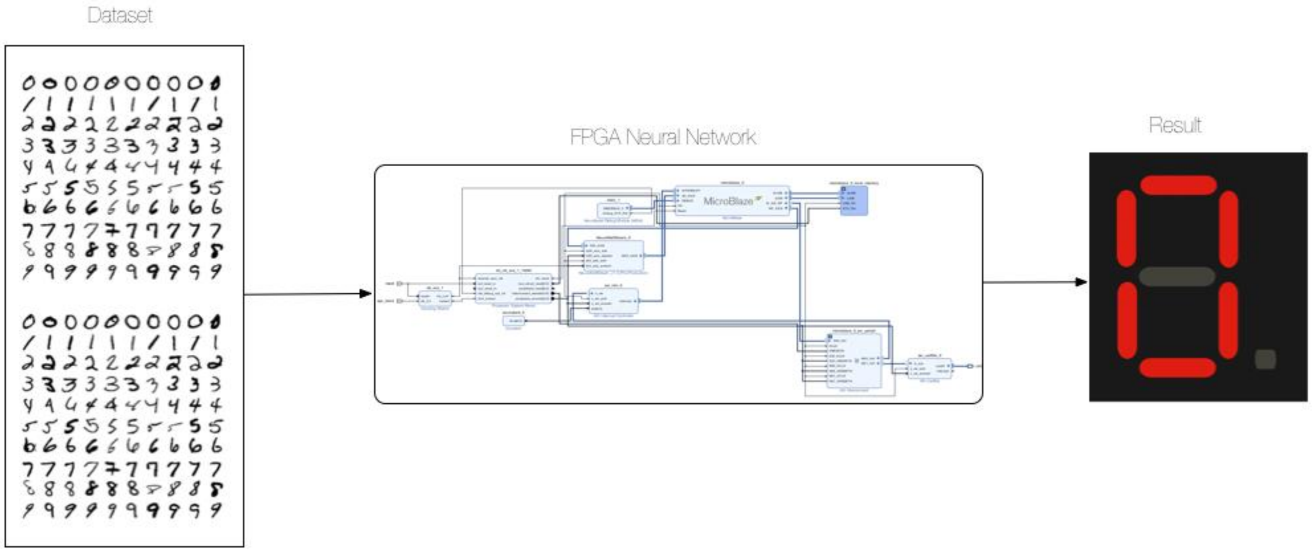
Neural Network

Reconfigurable Computing

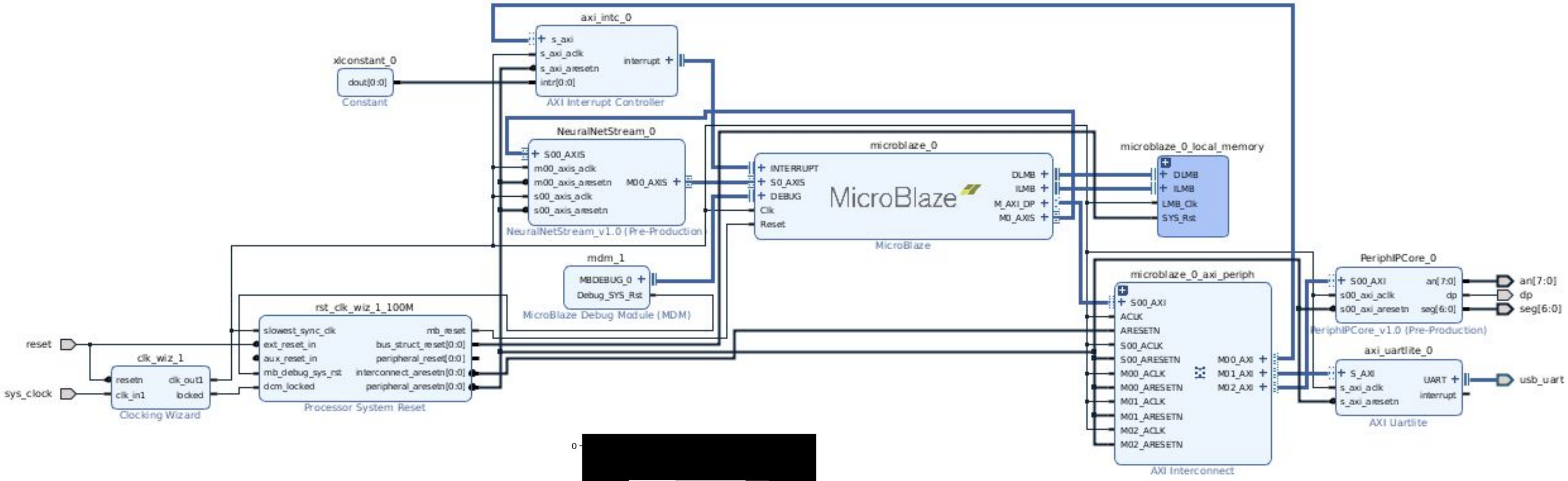
Silvério Pereira 76505

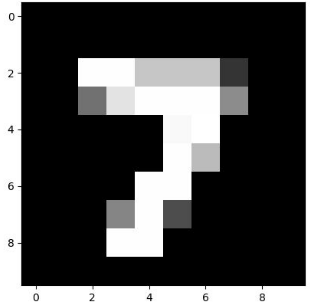


Dataset

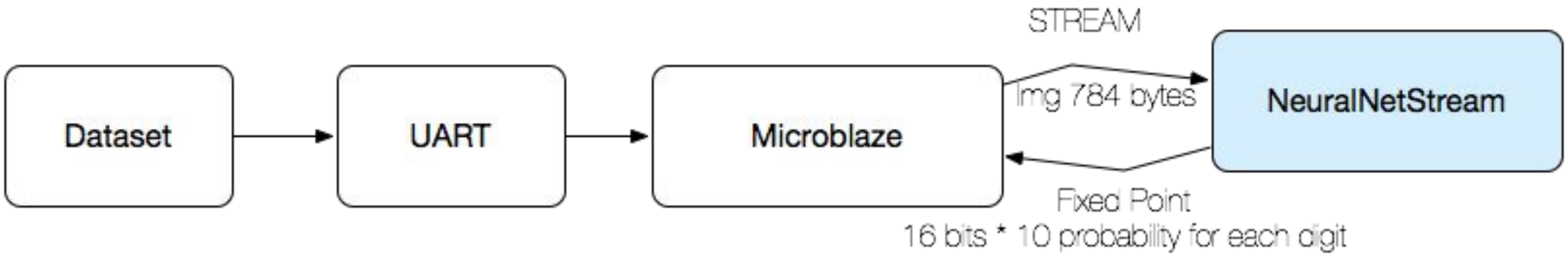
FPGA Neural Network

Result

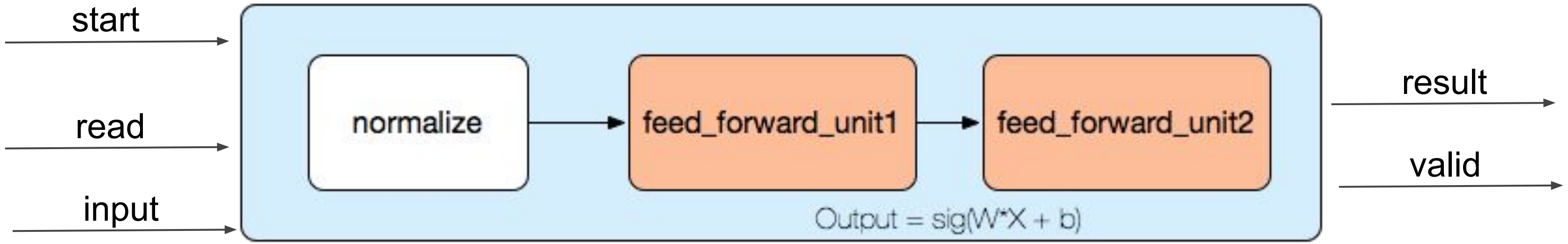

axi_intc_0
s_axi
s_axi_aclk
s_axi_aresetn
intr[0:0]
interrupt
AXI Interrupt Controller
xlconstant_0
dout[0:0]
Constant
NeuralNetStream_0
S00_AXIS
m00_axis_aclk
m00_axis_aresetn
s00_axis_aclk
s00_axis_aresetn
M00_AXIS
NeuralNetStream_v1.0 (Pre-Production)
microblaze_0
INTERRUPT
S0_AXIS
DEBUG
Clk
Reset
MicroBlaze
DLMB
ILMB
M_AXI_DP
M0_AXIS
MicroBlaze
microblaze_0_local_memory
DLMB
ILMB
LMB_Clk
SYS_Rst
mdm_1
MBDEBUG_0
Debug_SYS_Rst
MicroBlaze Debug Module (MDM)
rst_clk_wiz_1_100M
slowest_sync_clk
ext_reset_in
aux_reset_in
mb_debug_sys_rst
dcm_locked
mb_reset
bus_struct_reset[0:0]
peripheral_reset[0:0]
interconnect_aresetn[0:0]
peripheral_aresetn[0:0]
Processor System Reset
reset
sys_clock
clk_wiz_1
resetn
clk_in1
clk_out1
locked
Clocking Wizard
microblaze_0_axi_periph
S00_AXI
ACLK
ARESETN
S00_ACLK
S00_ARESETN
M00_ACLK
M00_ARESETN
M01_ACLK
M01_ARESETN
M02_ACLK
M02_ARESETN
M00_AXI
M01_AXI
M02_AXI
AXI Interconnect
PeriphIPCore_0
S00_AXI
s00_axi_aclk
s00_axi_aresetn
an[7:0]
dp
seg[6:0]
PeriphIPCore_v1.0 (Pre-Production)
an[7:0]
dp
seg[6:0]
axi_uartlite_0
S_AXI
s_axi_aclk
s_axi_aresetn
UART
interrupt
AXI Uartlite
usb_uart

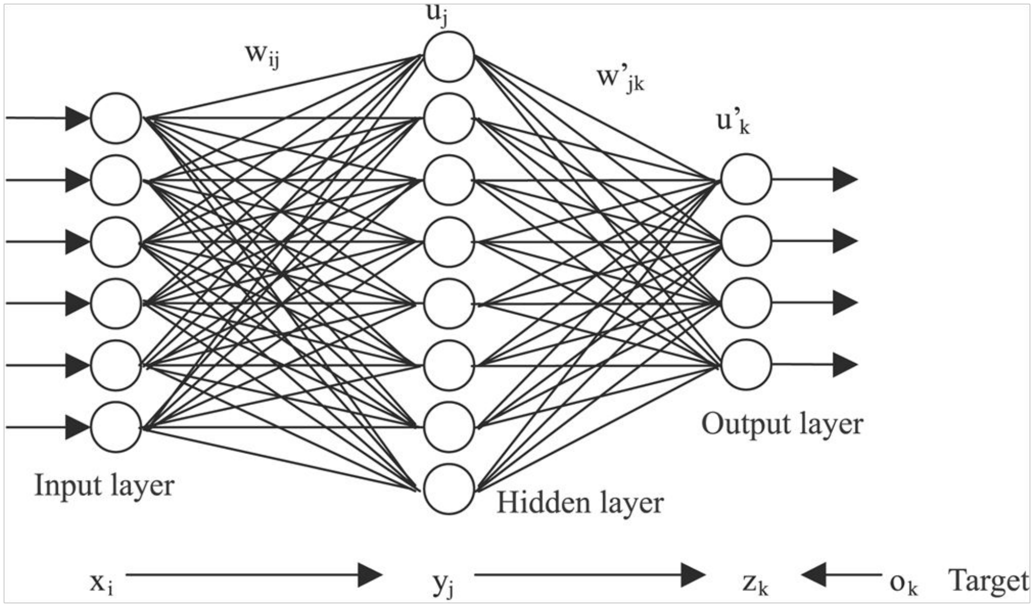

0
2
4
6
8
0
2
4
6
8

Architecture



NeuralNetStream

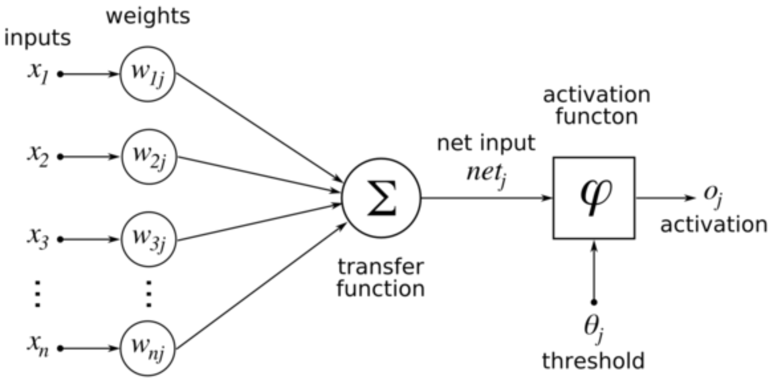




In our example:

100 input layers
10 hidden layers
10 output layers

About the sigmoid function



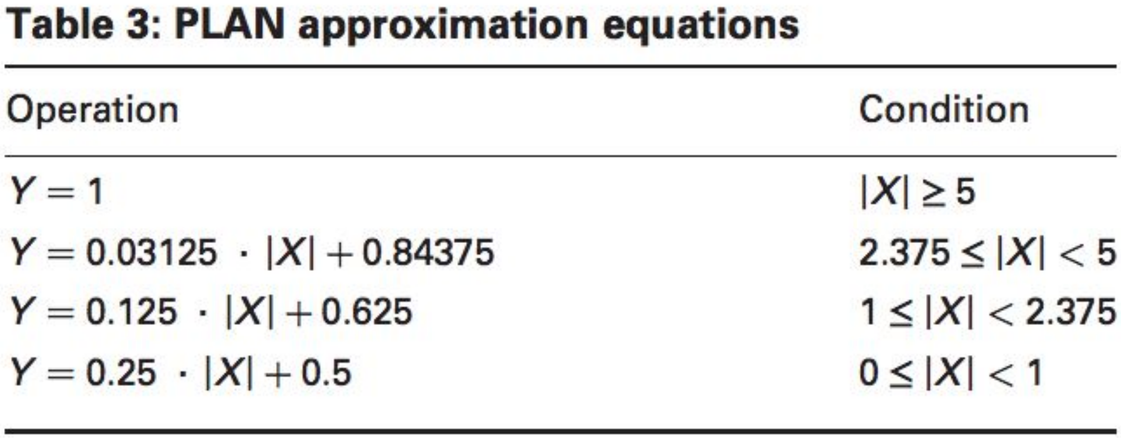
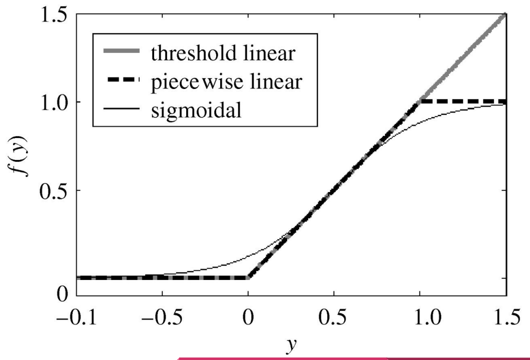
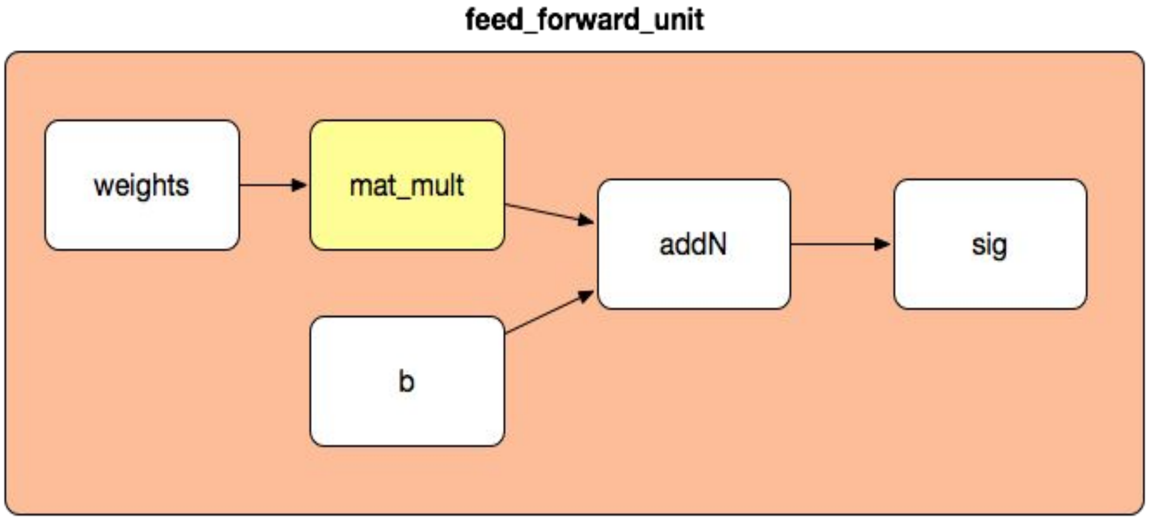


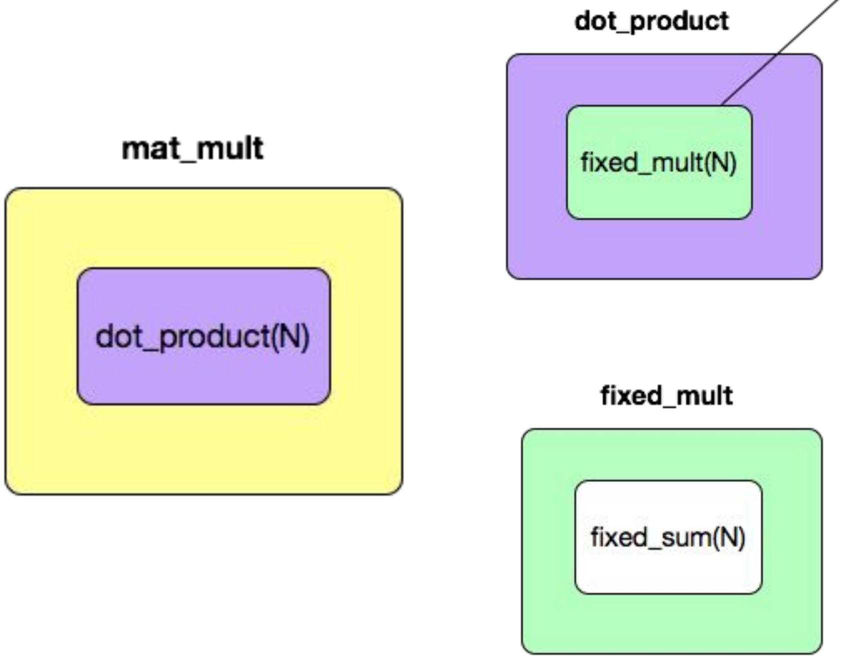
Table 3: PLAN approximation equations

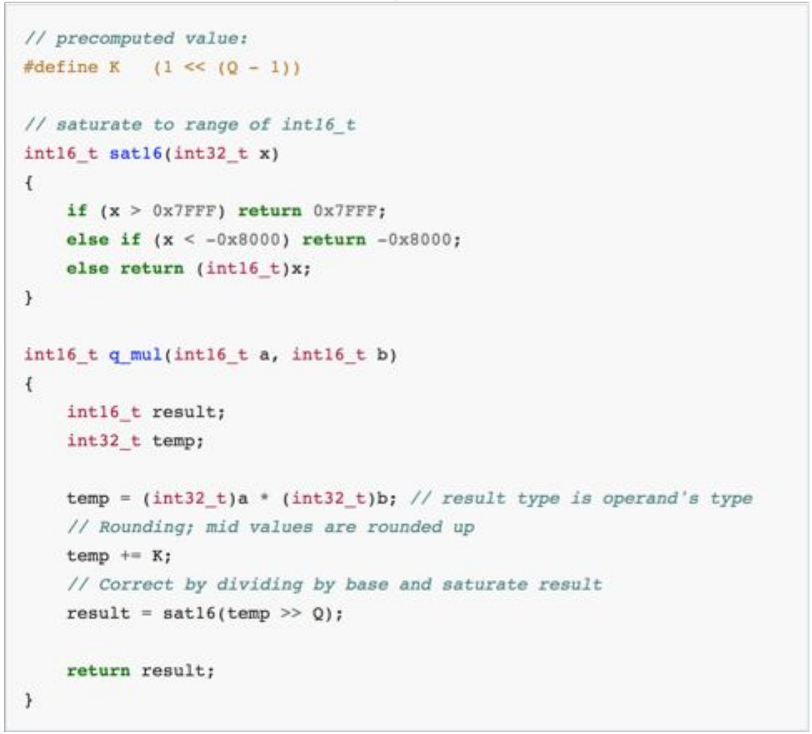
Operation	Condition
$Y = 1$	$\lvert X\rvert \geq 5$
$Y = 0.03125 \cdot \lvert X\rvert + 0.84375$	$2.375 \leq \lvert X\rvert < 5$
$Y = 0.125 \cdot \lvert X\rvert + 0.625$	$1 \leq \lvert X\rvert < 2.375$
$Y = 0.25 \cdot \lvert X\rvert + 0.5$	$0 \leq \lvert X\rvert < 1$





Output = sig(W*X + b)





```
// precomputed value:
#define K   (1 << (Q - 1))

// saturate to range of int16_t
int16_t sat16(int32_t x)
{
    if (x > 0x7FFF) return 0x7FFF;
    else if (x < -0x8000) return -0x8000;
    else return (int16_t)x;
}

int16_t q_mul(int16_t a, int16_t b)
{
    int16_t result;
    int32_t temp;

    temp = (int32_t)a * (int32_t)b; // result type is operand's type
    // Rounding; mid values are rounded up
    temp += K;
    // Correct by dividing by base and saturate result
    result = sat16(temp >> Q);

    return result;
}
```

Hardware

- Matrix operations
- Fixed point operations
- Sigmoid function
- Displays

Software

- Image reception through UART
- Send image to hardware
- Receiving hardware output
- Selection highest probability
- Send it to displays