FPGA IGNITE 2023 SUMMER COURSE

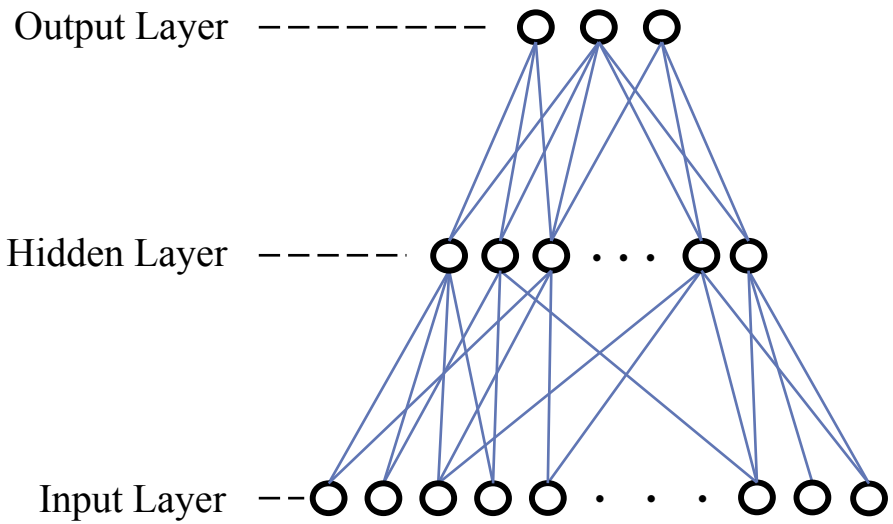
HLS Lab Portion

Brief Recap of CNNs

- CNNs are producing state-of-the-art image recognition accuracies
- Acceleration of CNN inference is important
  - Training can be done once offline
  - Increased emphasis on performing CNN inference in an embedded computing context
- CNNs are evolving fast
  - FPGAs are flexible enough to take advantage of recent research such as low-precision CNNs

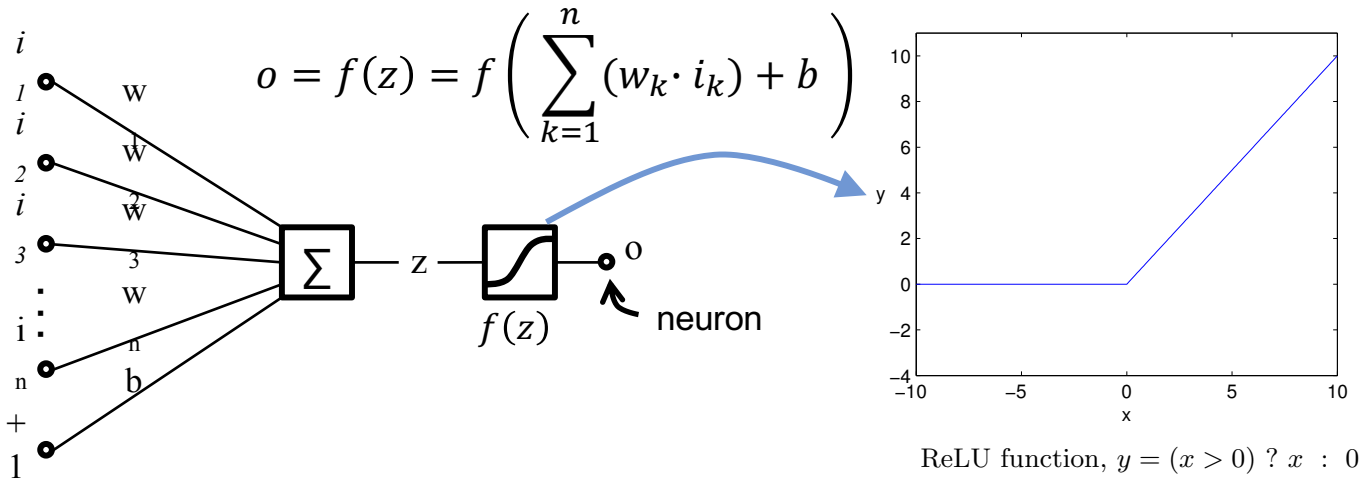
Deep Neural Networks

- Deep neural networks
  - A multi-layer structure
  - Can model more complex problems



A Neuron

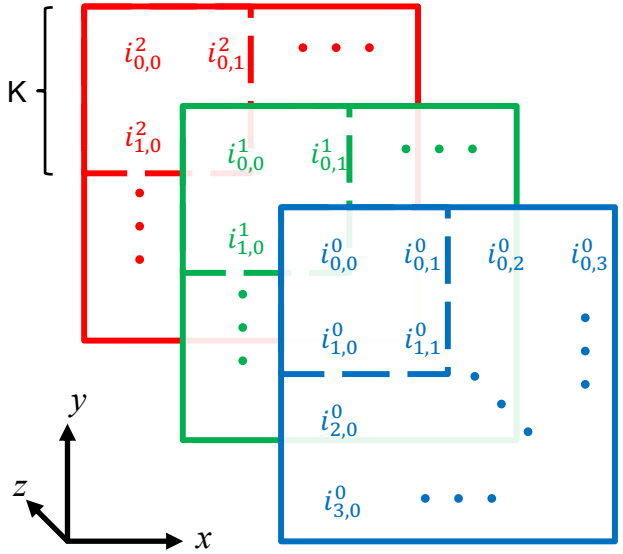
- The basic element in neural networks



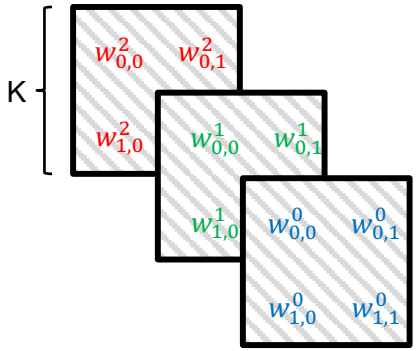
Key computation is MAC (multiply-accumulate)

Convolutional Layers

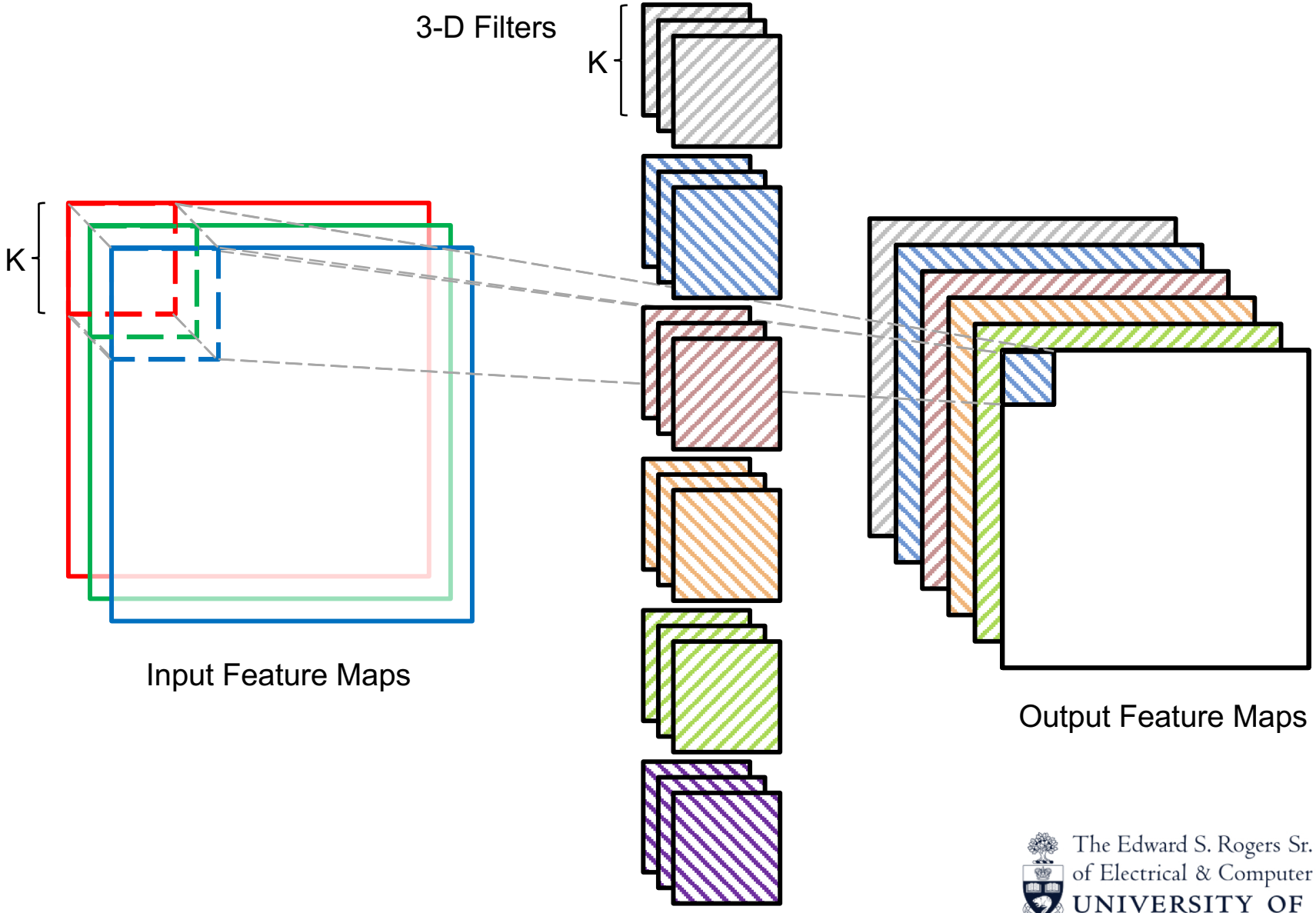
- Neurons are organized as a set of feature maps



- Filters are used to extract new features from input feature maps
- Applied on a local region (a.k.a. receptive field)



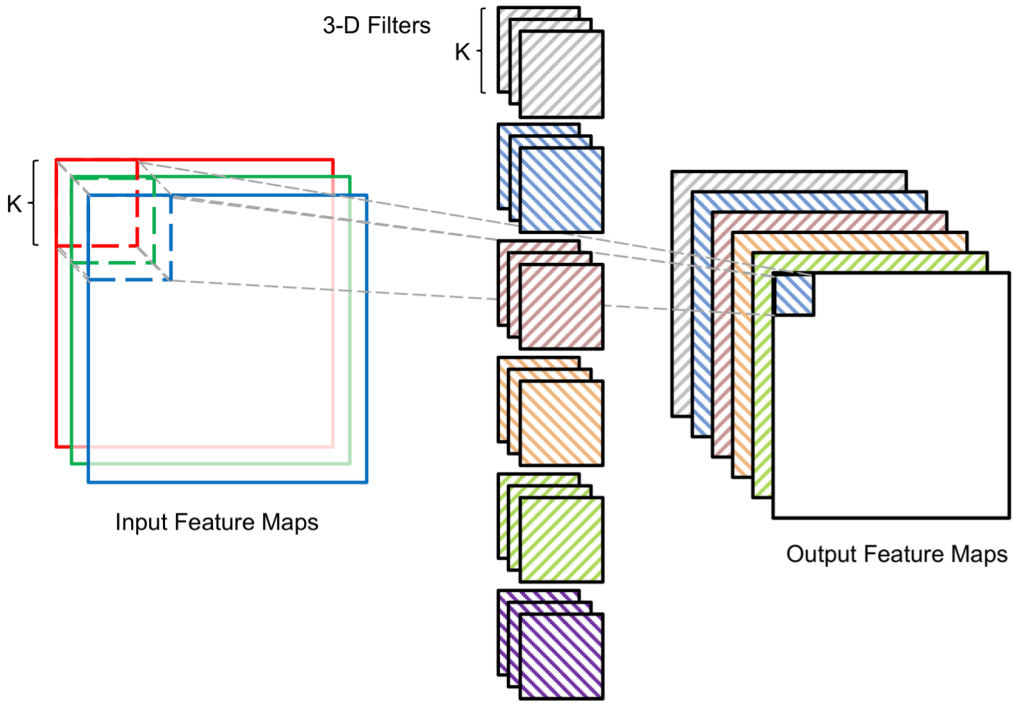
Convolutional Layers



Lab Part

- You will synthesize the computations for a CNN's convolution layer from C to hardware
  - Basic HW synthesis
    - From a "vanilla" C implementation
  - Loop unrolling and pipelining
  - Memory partitioning
  - Spatial parallelism with Pthreads
- Observe successively better performance with each step

Convolution Layer in Lab



- **16 input feature maps**
- **8 filters ➔ 8 output feature maps**
- **Filter dim: 3x3 (K = 3)**
- **Input feature map dim: 30x30**
- **Output feature map dim: 28x28**

LegUp HLS Demo for Part 1 of Lab

The Edward S. Rogers Sr. Department
of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Part 2 of Lab

- Apply loop unrolling and pipelining to optimize CNN circuit performance

- Loop unrolling concept to expose parallelism:

```
for (im = 0; im < NUM_INPUT_MAPS; im+=1) {
   output_fm_value += weights[om][i][j][im] * input_fms[row+i][col+j][im];
}
```

```
for (im = 0; im < NUM_INPUT_MAPS; im+=2) {
          // two MACs per loop iteration
          output_fm_value += weights[om][i][j][im] * input_fms[row+i][col+j][im];
          output_fm_value += weights[om][i][j][im+1] * input_fms[row+i][col+j][im+1];
}
```

Parts 3 and 4 of Lab

- Memory partitioning & spatial parallelism improve speed
  - Pthreads flow in LegUp HLS
- Memory partitioning: why?
  - Each array in program implemented in a RAM in FPGA
  - FPGA RAMs are dual port
  - At **most** two memory accesses/clock cycle
    - Can limit hardware performance and loop piplining II
- Memory partitioning idea:
  - Partition original array in to multiple arrays
  - Each array partition in separate RAM ➔ can be accessed in parallel
- Part 3: partition filter weights array, input feature map array into two arrays each