Can the Network be the AI Accelerator?

Davide Sanvito
Politecnico di Milano, NEC Laboratories Europe

Giuseppe Siracusano
NEC Laboratories Europe

Roberto Bifulco
NEC Laboratories Europe

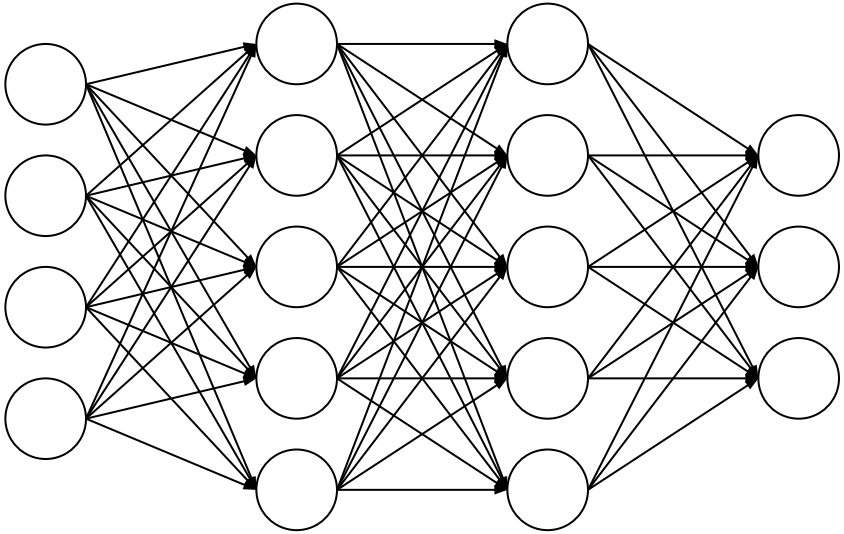
POLITECNICO MILANO 1863

NEC

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 761493 5Gtango

Artificial Neural Networks (ANN)

- Machine Learning tool
- Sequence of interconnected layers of neurons
  - Activation
  - Hyperparameters
  - MLP / CNN / RNN
- Training vs Inference



AI accelerators

- General-purpose GPU (GPGPU)
  - Parallel computing on large amount of data (SIMD)
  - High efficiency with large batches
  - Data movement overhead
  - Best suited for training
- Tensor Processing Unit (TPU)
  - Custom ASIC dedicated to inference
  - Data transfer up to 76% of processing time

For latency sensitive services, NN inference is performed on CPUs!

Our contributions

- Programmable Network Devices
  - Network cards and switches
  - More than pure forwarding
  - Privileged position in an end-to-end system
- Profiling the computation required by a NN inference on a CPU
- Analysis of the options to offload (a subset of) NN's layers from the CPU to a different processor
- Evaluating under which conditions NIC and switches are suitable to work as CPUs co-processors for NN inference

Neural Networks inference workload

- NN inference workload in Google's data centers

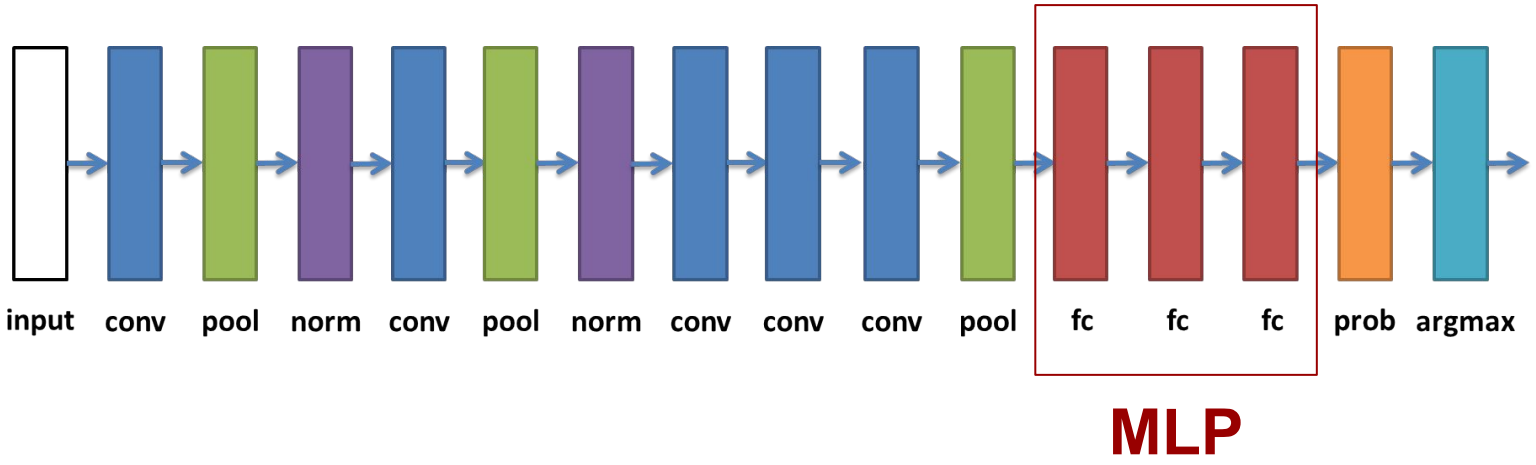
NN	Workload
MLP	61%
RNN	29%
CNN	5%

- Final portion of a CNN is a set of *fc* layers (MLP)

N. P Jouppi, et al. 2017. *In-datacenter performance analysis of a tensor processing unit*. 44th Annual International Symposium on Computer Architecture

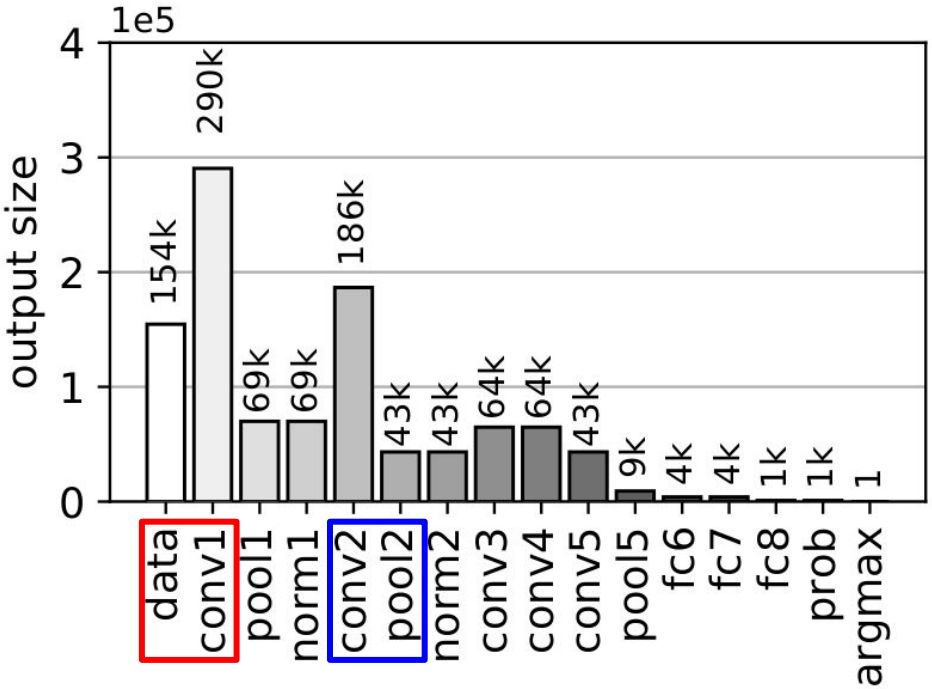
AlexNet

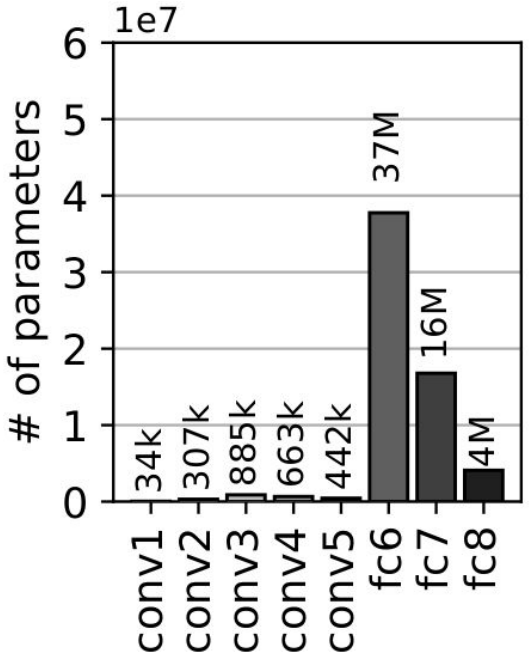
- CNN for image classification
- Winner of ImageNet Large Scale Visual Recognition (ILSVRC) 2012



A. Krizhevsky, et al. 2012. *Imagenet classification with deep convolutional neural networks*. In Advances in neural information processing systems

Structural analysis

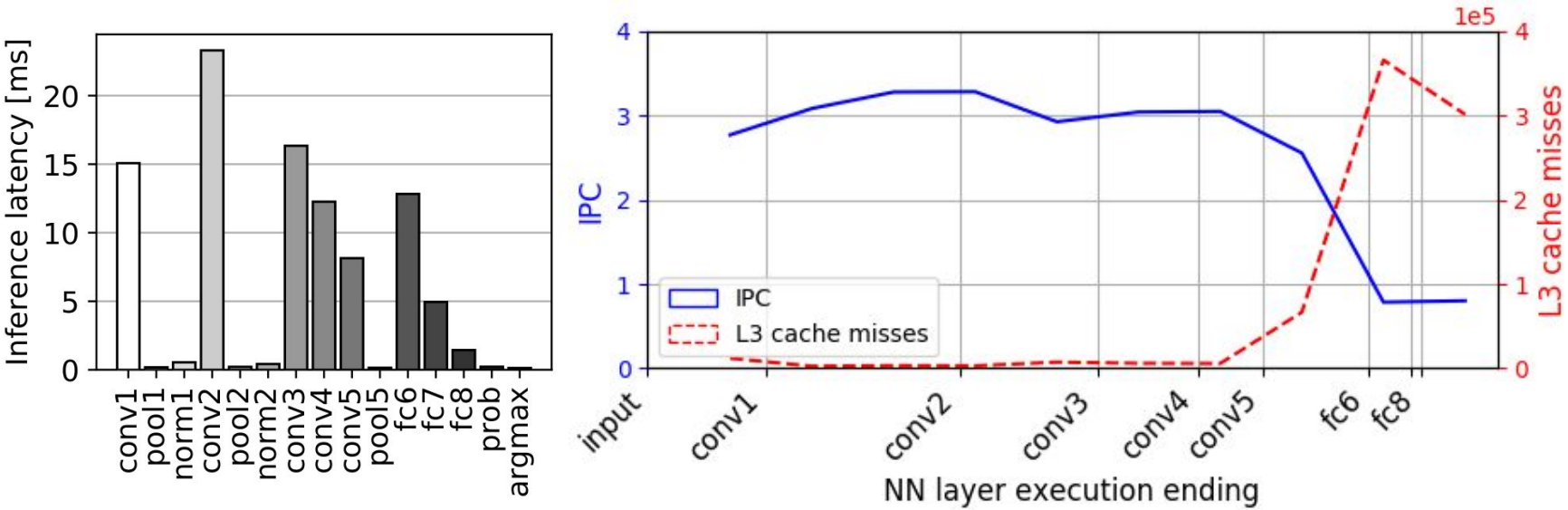




Runtime analysis

- Testbed
  - dual-socket (NUMA) machine with two CPUs Intel Xeon E5-2650 (8 cores@2.4GHz)
  - hyperthreading disabled
  - 16GB of RAM per socket
  - L1 data and instruction caches: 32KB per core
  - L2 cache: 256KB per core
  - L3 cache: 20MB shared by all the CPU's cores
- Intel Caffe running on a single isolated core on a dedicated CPU
- Total and per layer inference latency
- Linux *perf* tool
  - Instructions Per Cycle rate (IPC)
  - Stalled cycles
  - L1 (data), L2, L3 cache misses

Inference latency



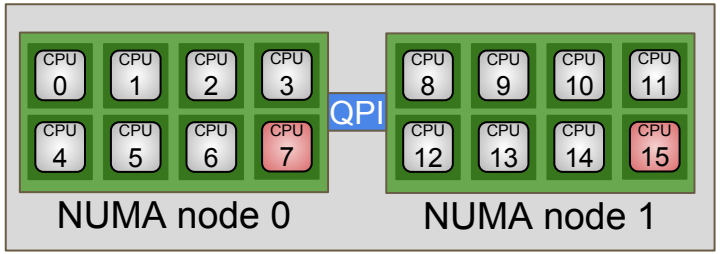
- *conv* layers processing is computation-bound
- *fc* layers processing is memory-bound

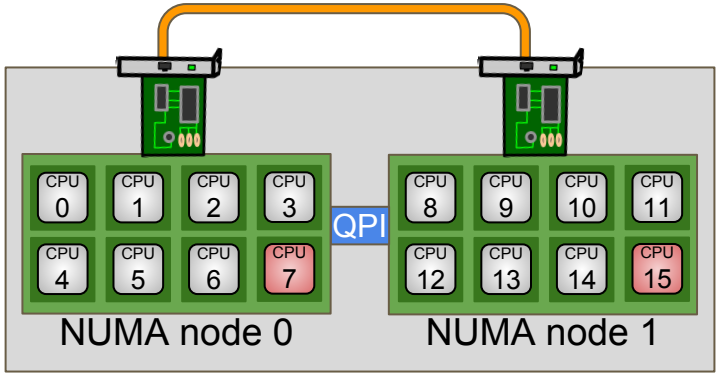
NN splitting

- CPUs are efficient executors for *conv*, *pool* and *norm* layers
- During *fc* layers, CPU's pipeline is stalled for a large fraction of time
- Moving the execution of *fc* layers to another executor can:
  - reduce NN inference latency
  - free CPU resources for a better suited workload

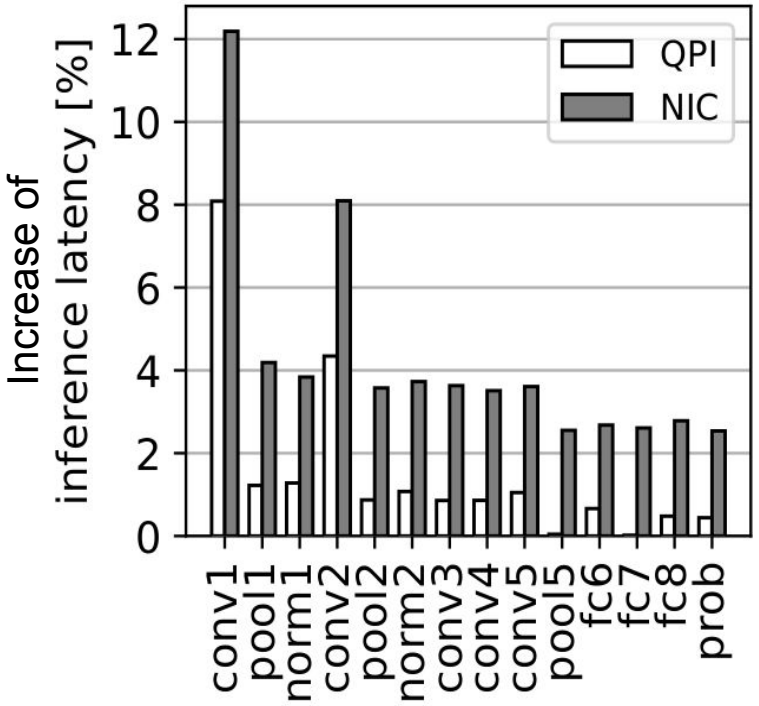
NN splitting (2)

- NN execution has been splitted on two homogeneous executors
  - What's the impact of splitted execution on inference latency?
  - What's the communication overhead?



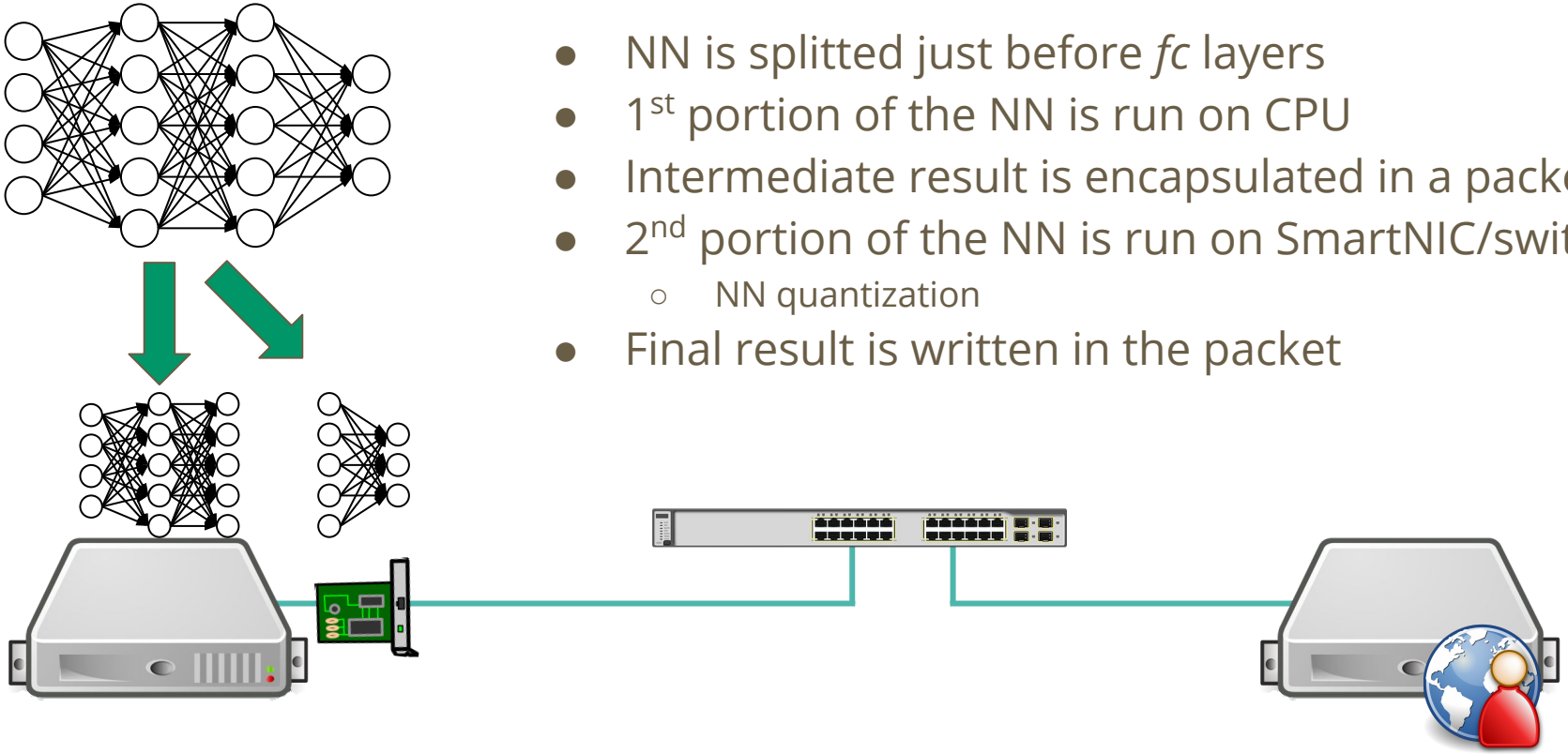


NN splitting overhead



- Higher overhead if split point is not carefully selected
- What if on-path network devices could perform NN processing?

BaNaNa Split



- NN is splitted just before *fc* layers
- 1st portion of the NN is run on CPU
- Intermediate result is encapsulated in a packet
- 2nd portion of the NN is run on SmartNIC/switch
  - NN quantization
- Final result is written in the packet

PoC implementation

- BNN (Binary Neural Network)
  - activation and parameters are represented with 1 bit
  - bitwise logical operations and popcount only
- Extension of N2Net for network processor-based SmartNICs support
  - compiler from BNN description to P4 program to configure an RMT-like switch pipeline
  - popcount implementation leverages built-in support
- Micro-benchmark
  - single binarized *fc* layer with 4096 neurons (*fc6*)
  - activation vector and output (512B) fit a network packet
  - layer's parameters (2MB) pre-configured in SmartNIC memory
  - execution takes 1 ms

G. Siracusano and R. Bifulco. 2018. In-network Neural Networks.arXiv preprint arXiv:1801.05731

Conclusion

- Analysis of suitability of current programmable network devices to work as NN accelerators
- BaNaNa Split: split the NN to execute computation-bound layers on CPU and offload memory-bound layers on a SmartNIC
  - Take advantage of a system's heterogeneous components!
  - Free CPU for more compute-intensive tasks, improving the efficiency of the overall infrastructure
- Open points
  - NN quantization accuracy
  - Network device's memory shared with classic network functions

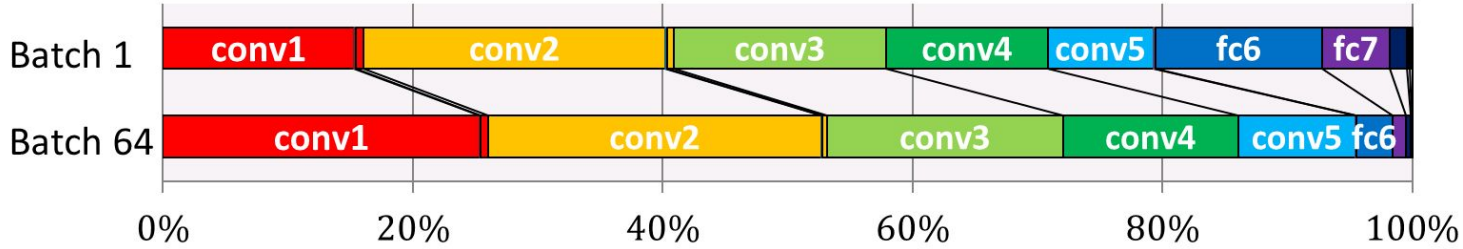
Thanks!

davide.sanvito@polimi.it

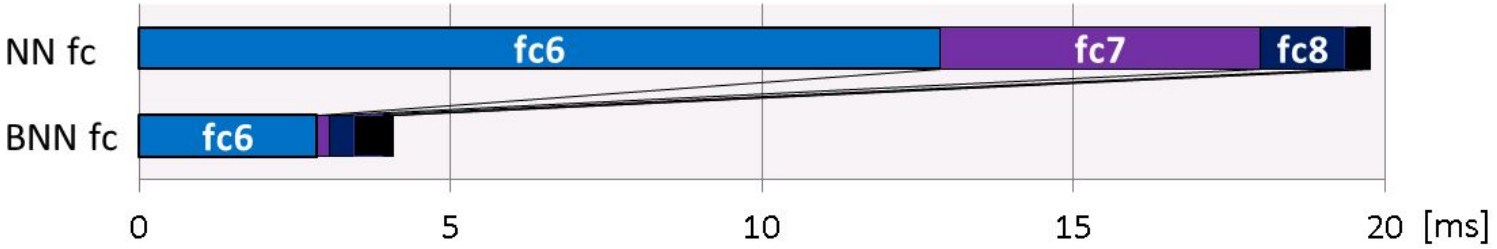
Inference latency with image batching

- Improved efficiency
- *fc* layers get most benefit
  - IPC increases
  - Cache misses are reduced
- Increased mean response time

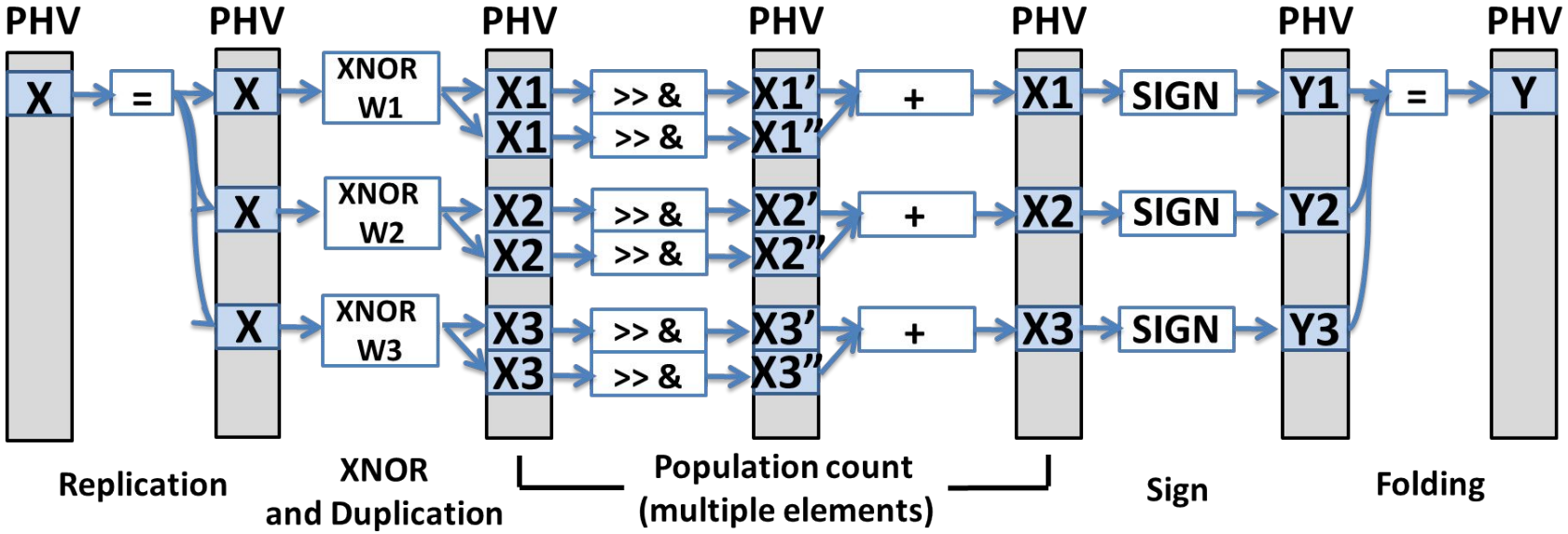
Batch size	1	16	32	64
Batched proc. latency [ms]	96	1334	2724	5585
Sequential proc. latency [ms]	96	1536	3071	6143
Batched proc. saving [%]	-	13.15	11.30	9.08
Batched proc. saving fc only [%]	-	69.74	73.75	80.32



BNN *fc* execution



N2Net


PHV
X
=
PHV
X
X
X
XNOR
W1
XNOR
W2
XNOR
W3
PHV
X1
X1
X2
X2
X3
X3
>> &
>> &
>> &
>> &
>> &
>> &
PHV
X1'
X1''
X2'
X2''
X3'
X3''
+
+
+
PHV
X1
X2
X3
SIGN
SIGN
SIGN
PHV
Y1
Y2
Y3
=
PHV
Y
Replication
XNOR
and Duplication
Population count
(multiple elements)
Sign
Folding