CNVLUTIN: Ineffectual-neuron-free DNN computing

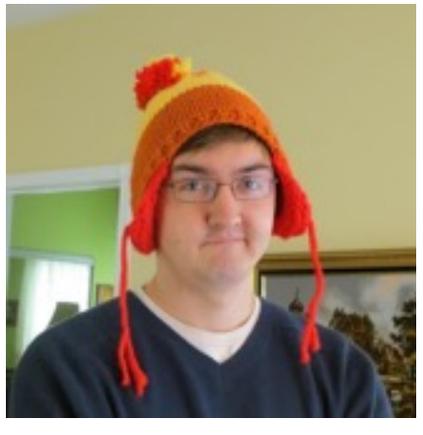
J. Albericio, P. Judd, T. Hetherington*,
T. Aamodt*, N. E. Jerger, A. Moshovos

UNIVERSITY OF TORONTO

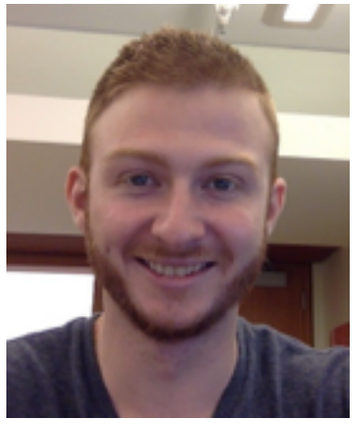
UBC *

CNVLUTIN: Ineffectual-neuron-free DNN computing

J. Albericio



P. Judd



T. Hetherington*

T. Aamodt*

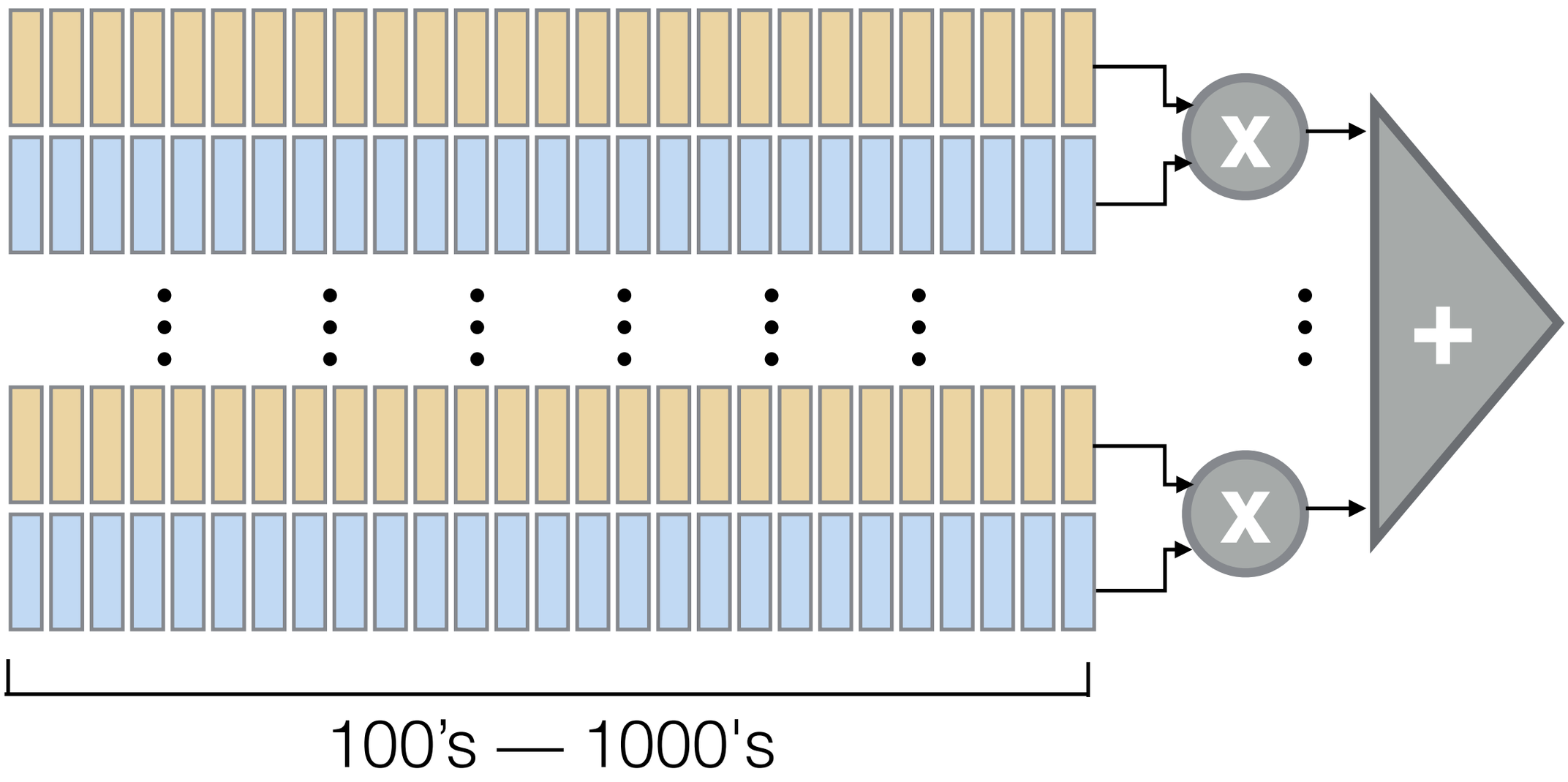
N. Enright Jerger

A. Moshovos

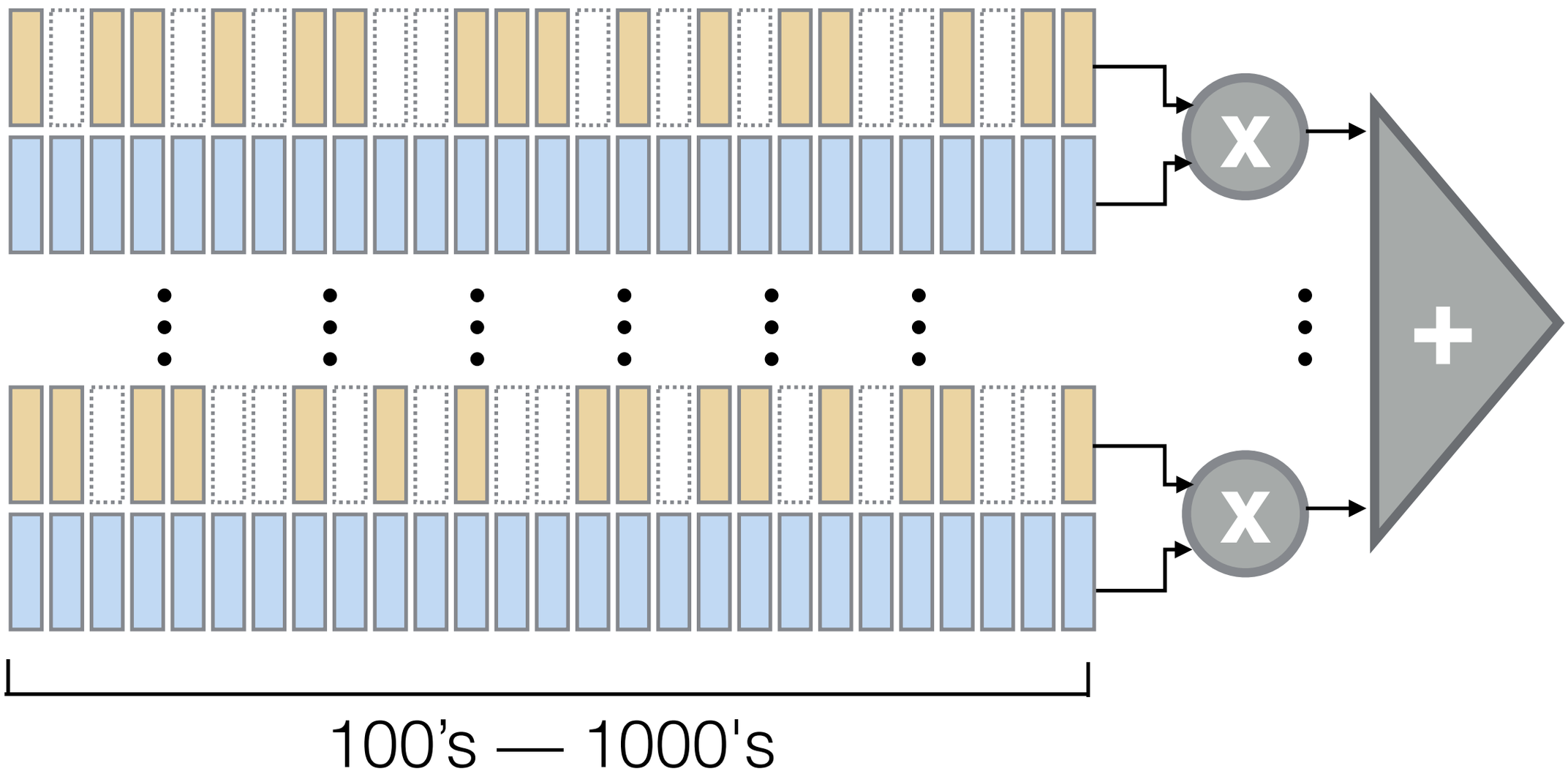
UNIVERSITY OF TORONTO

* UBC

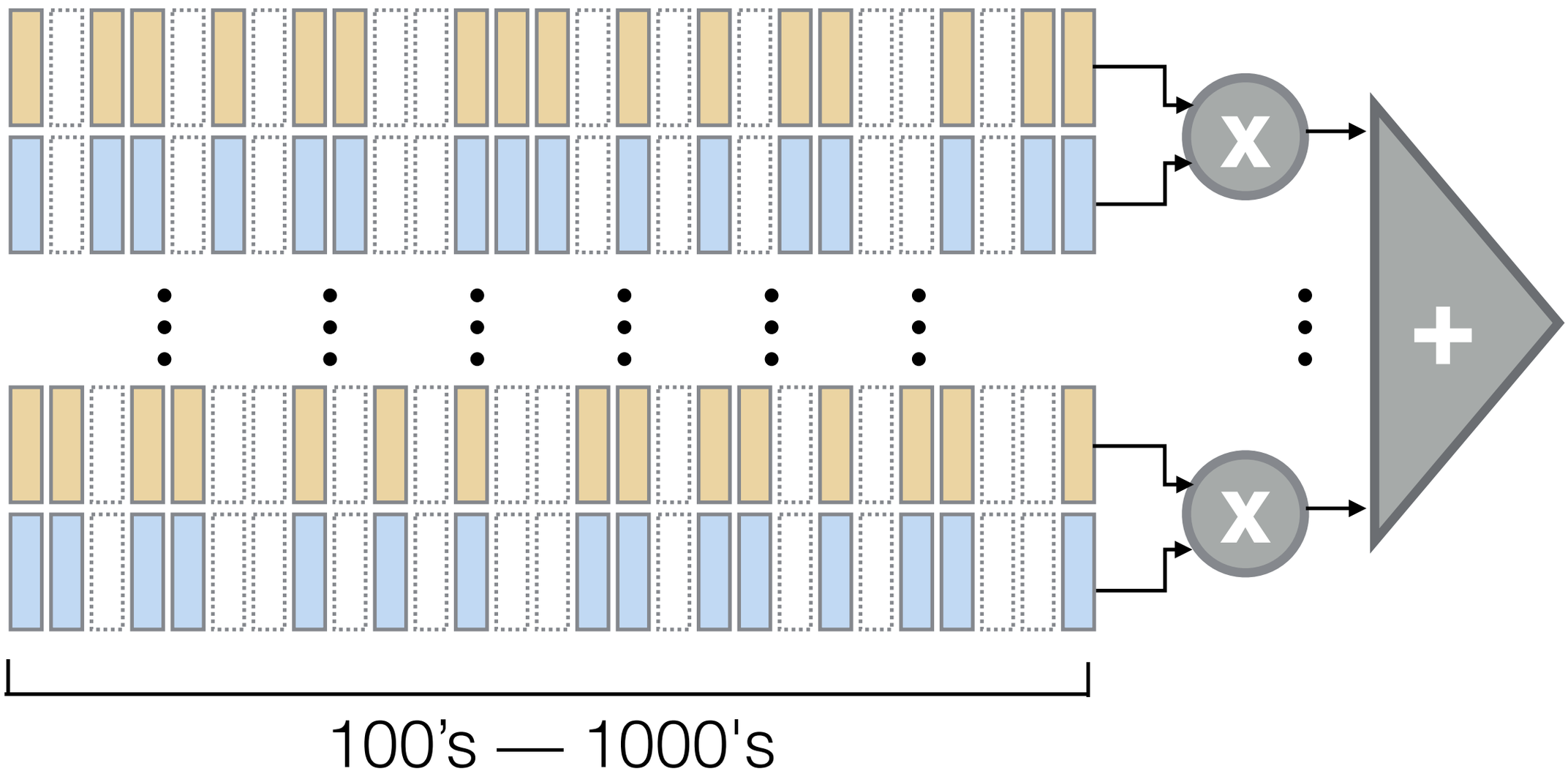
DNNs = SIMD Heaven



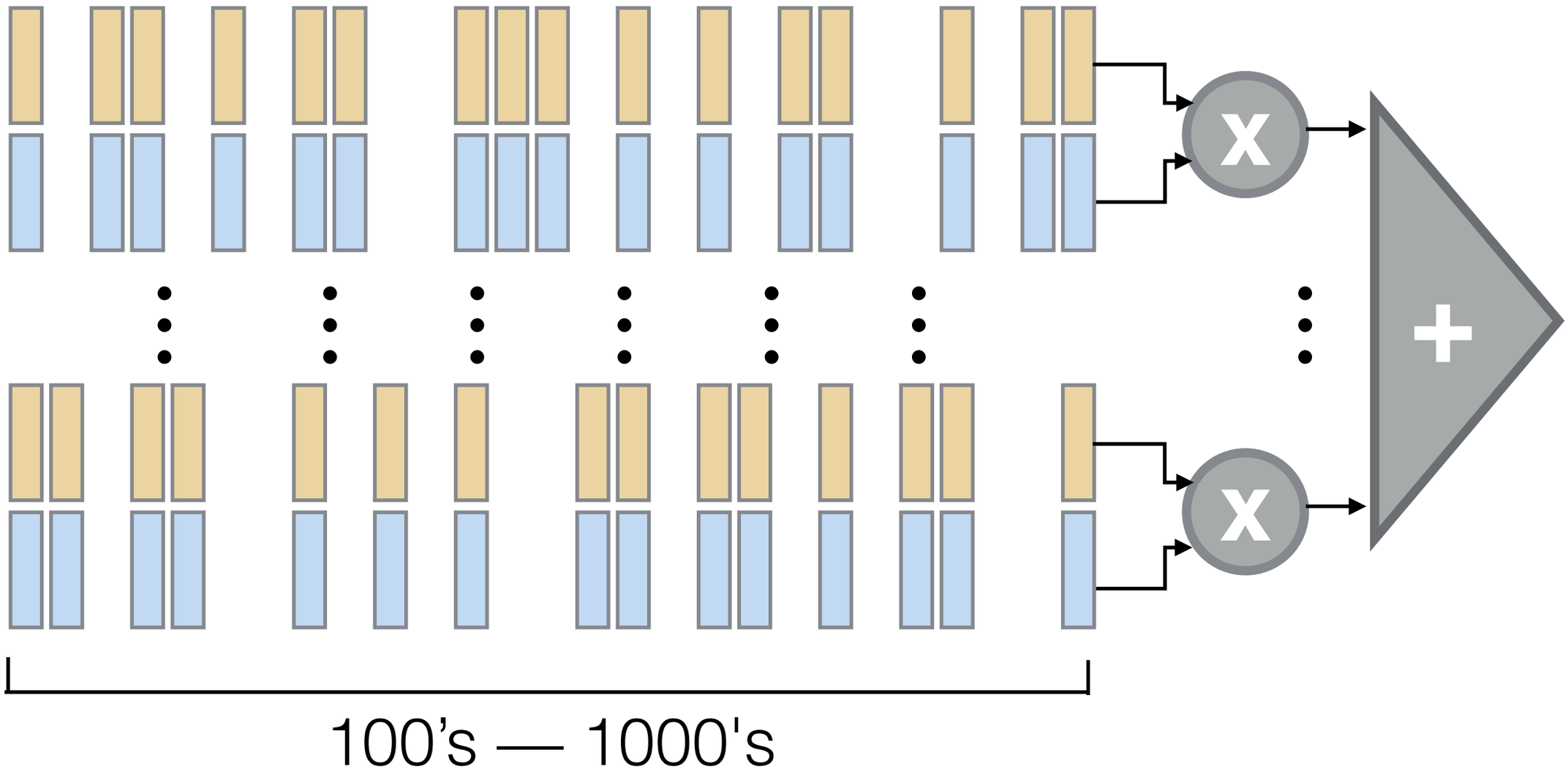
DNNs = SIMD Heaven



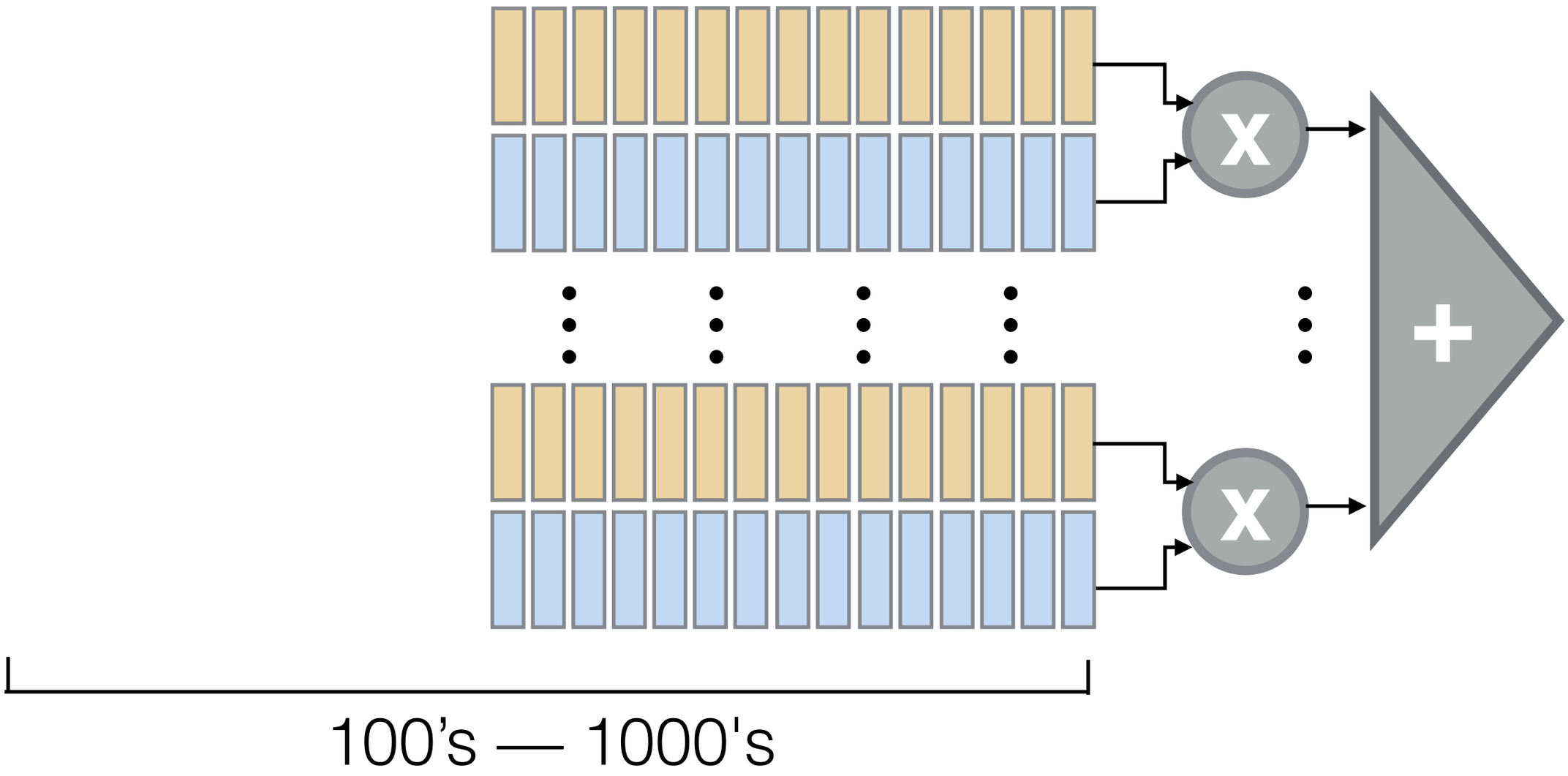
DNNs = SIMD Heaven



DNNs = SIMD Heaven



DNNs = SIMD Heaven



CNVLUTIN: Smarter SIMD

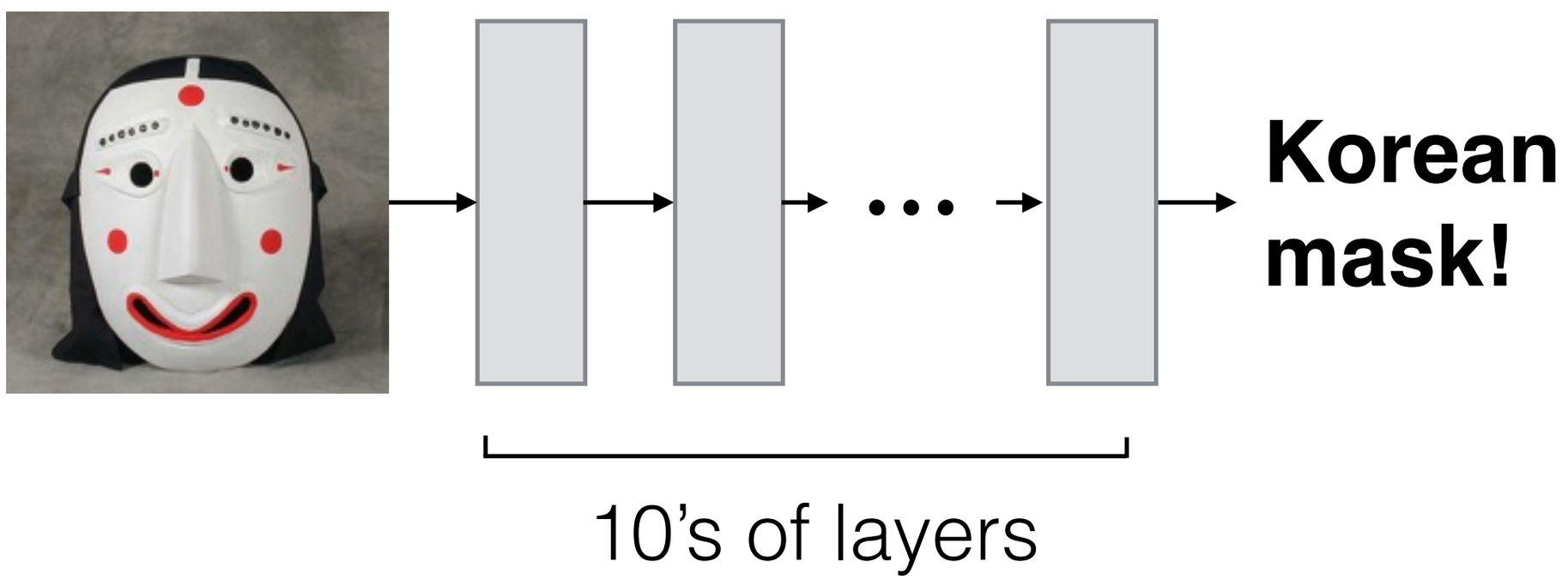
52% Performance — 2x ED²P

Out-of-the-box networks

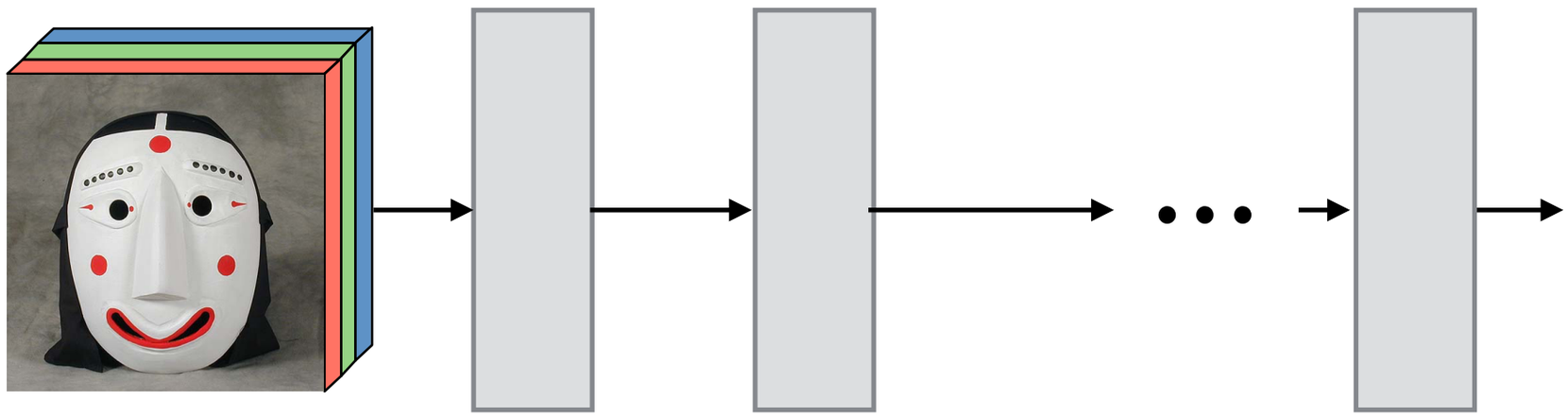
Outline

1. What's a CNN?
2. A wide SIMD design
3. CNVLUTIN: Skipping neurons in a wide SIMD design
4. Evaluation
5. Our approach

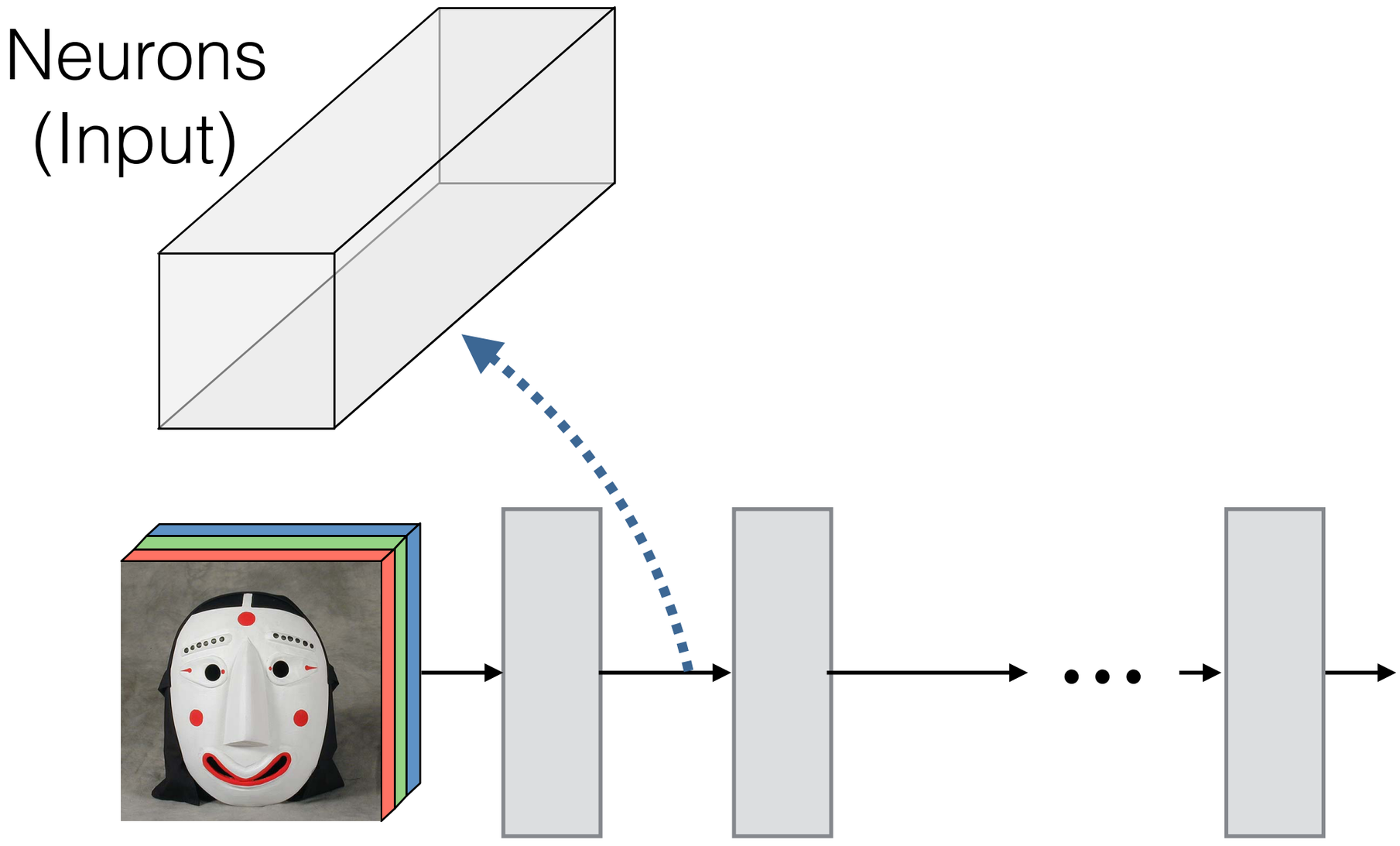
What's a CNN?



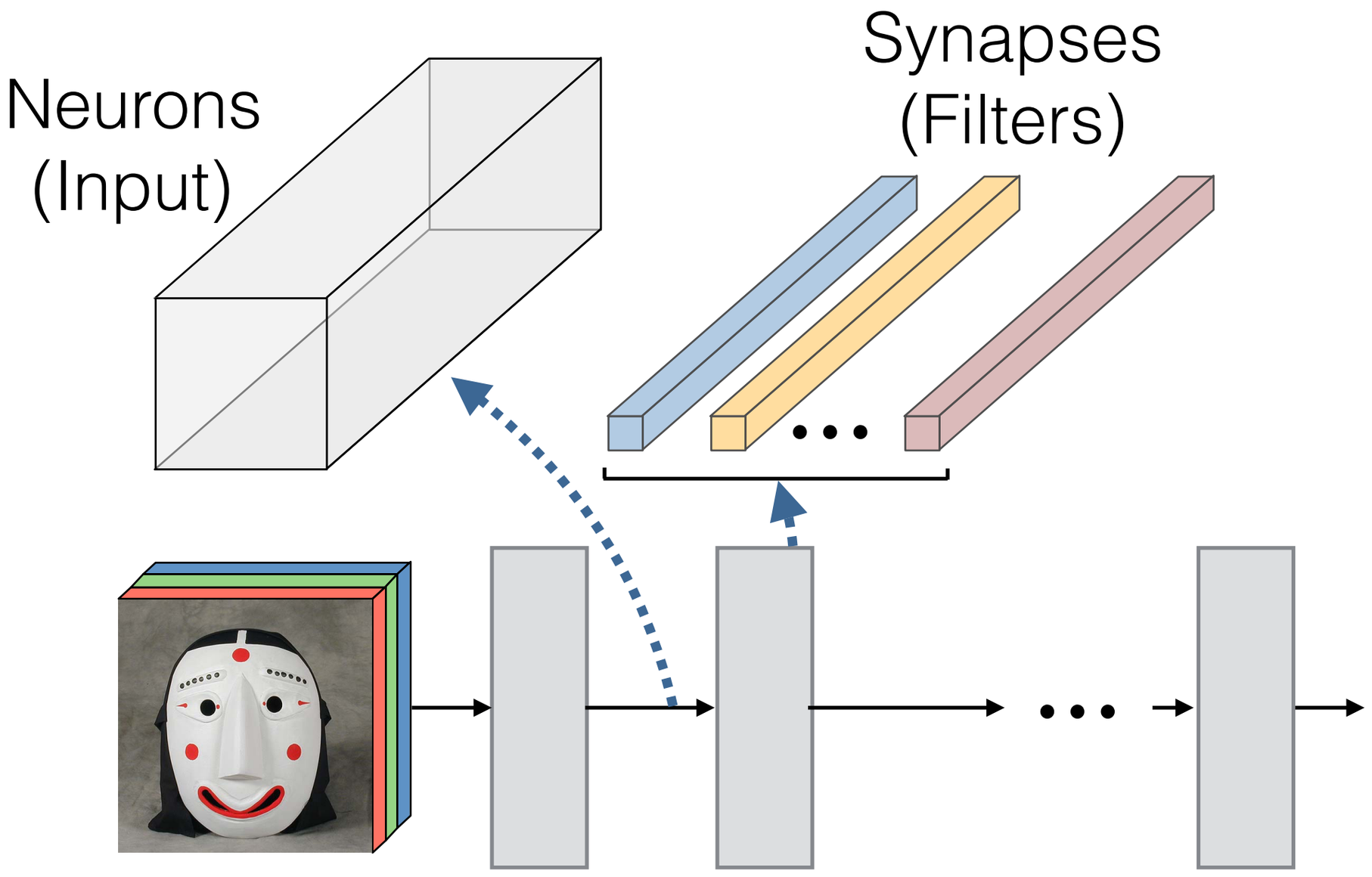
What's a CNN?



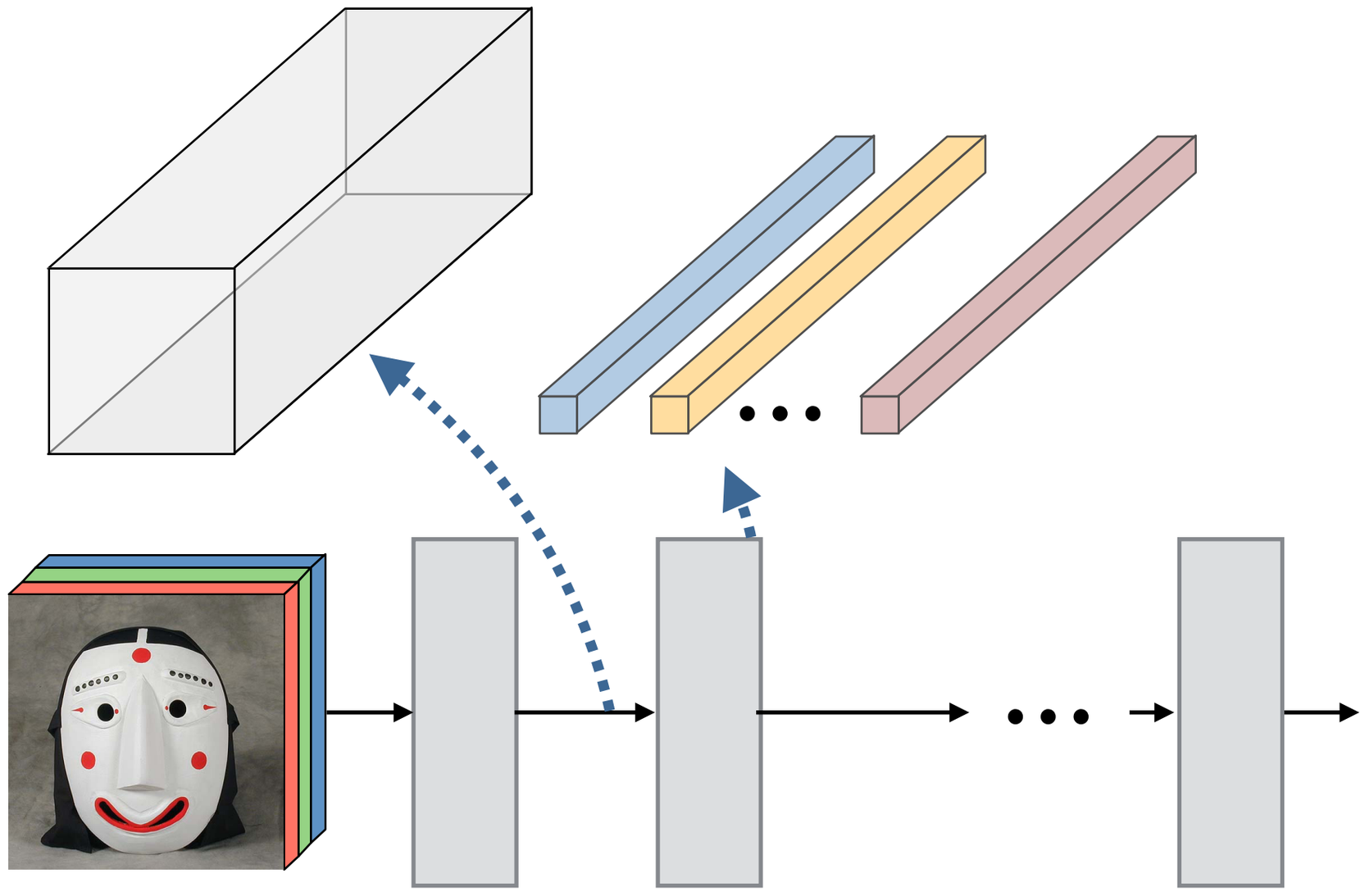
What's a CNN?



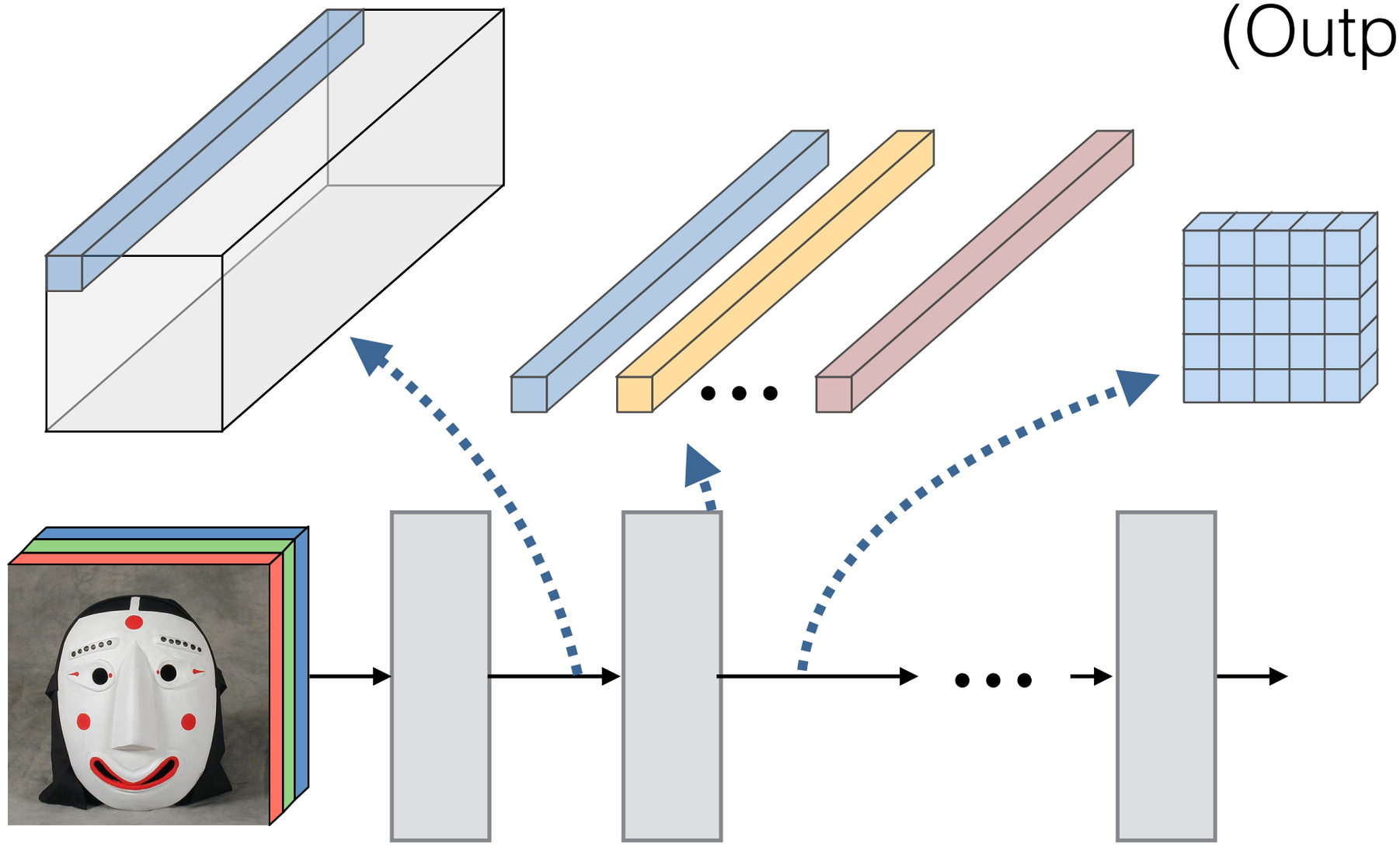
What's a CNN?



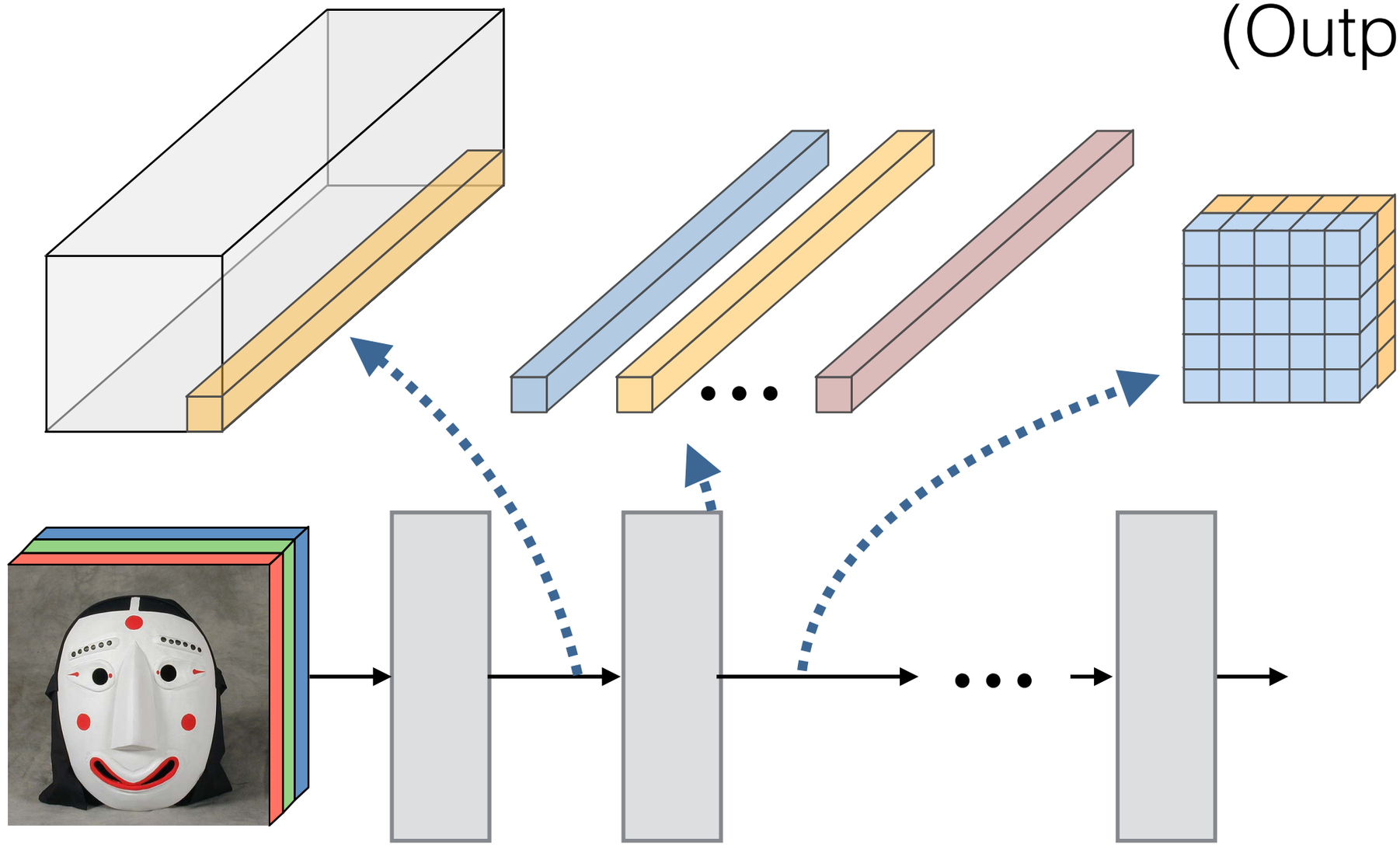
What's a CNN?



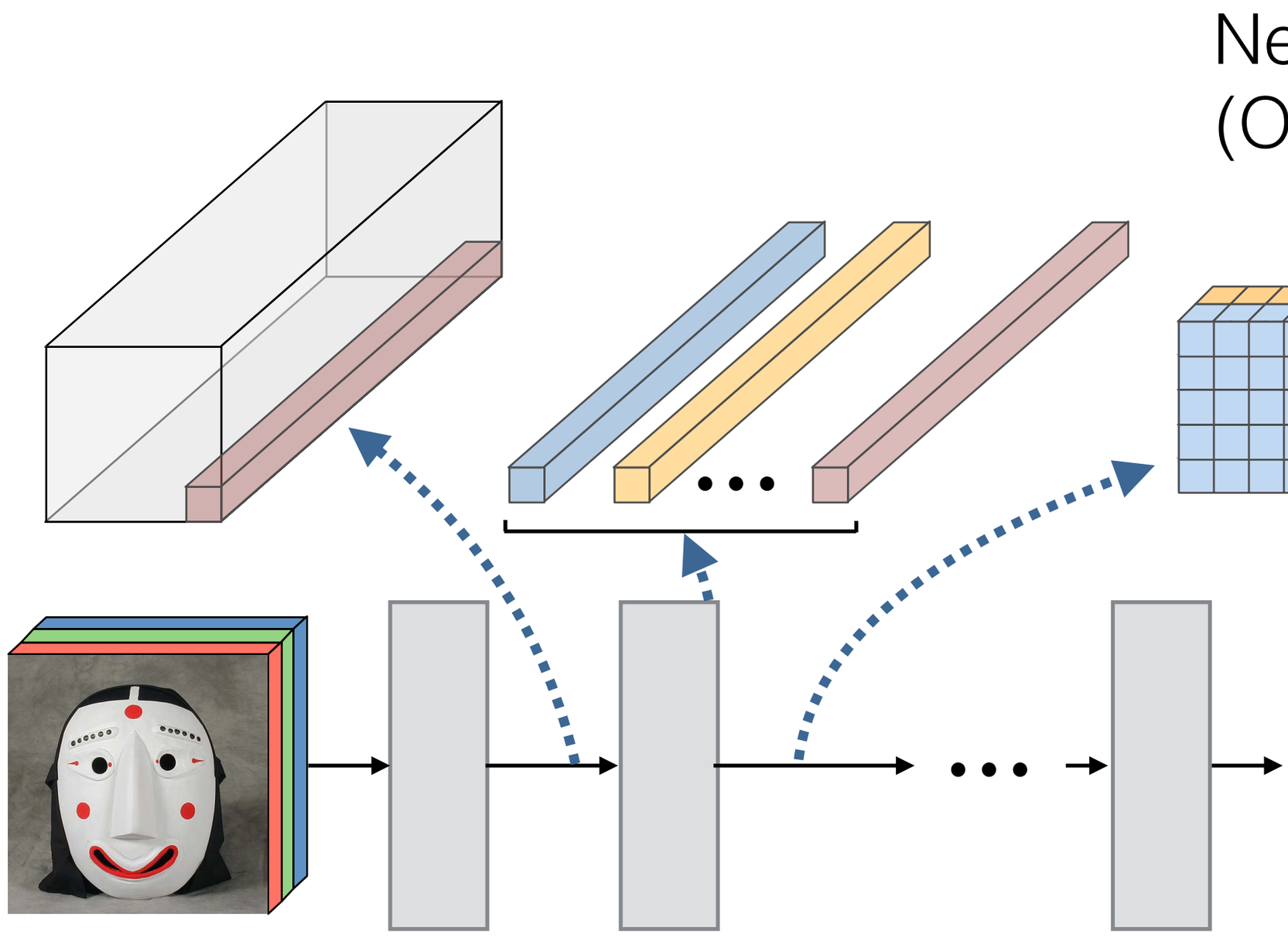
What's a CNN?



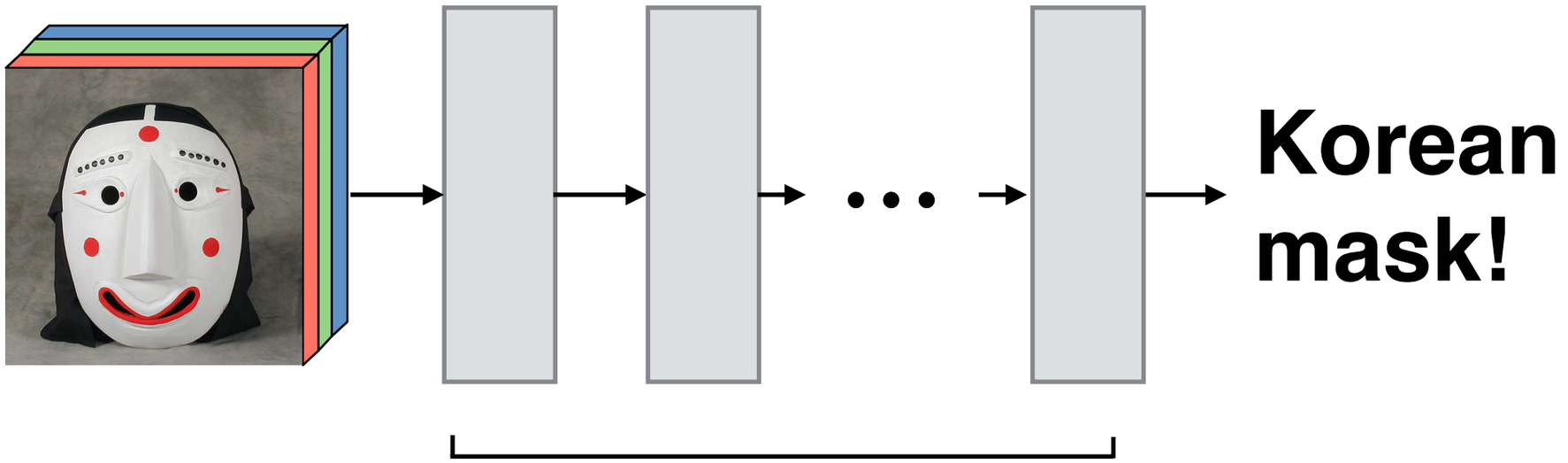
What's a CNN?



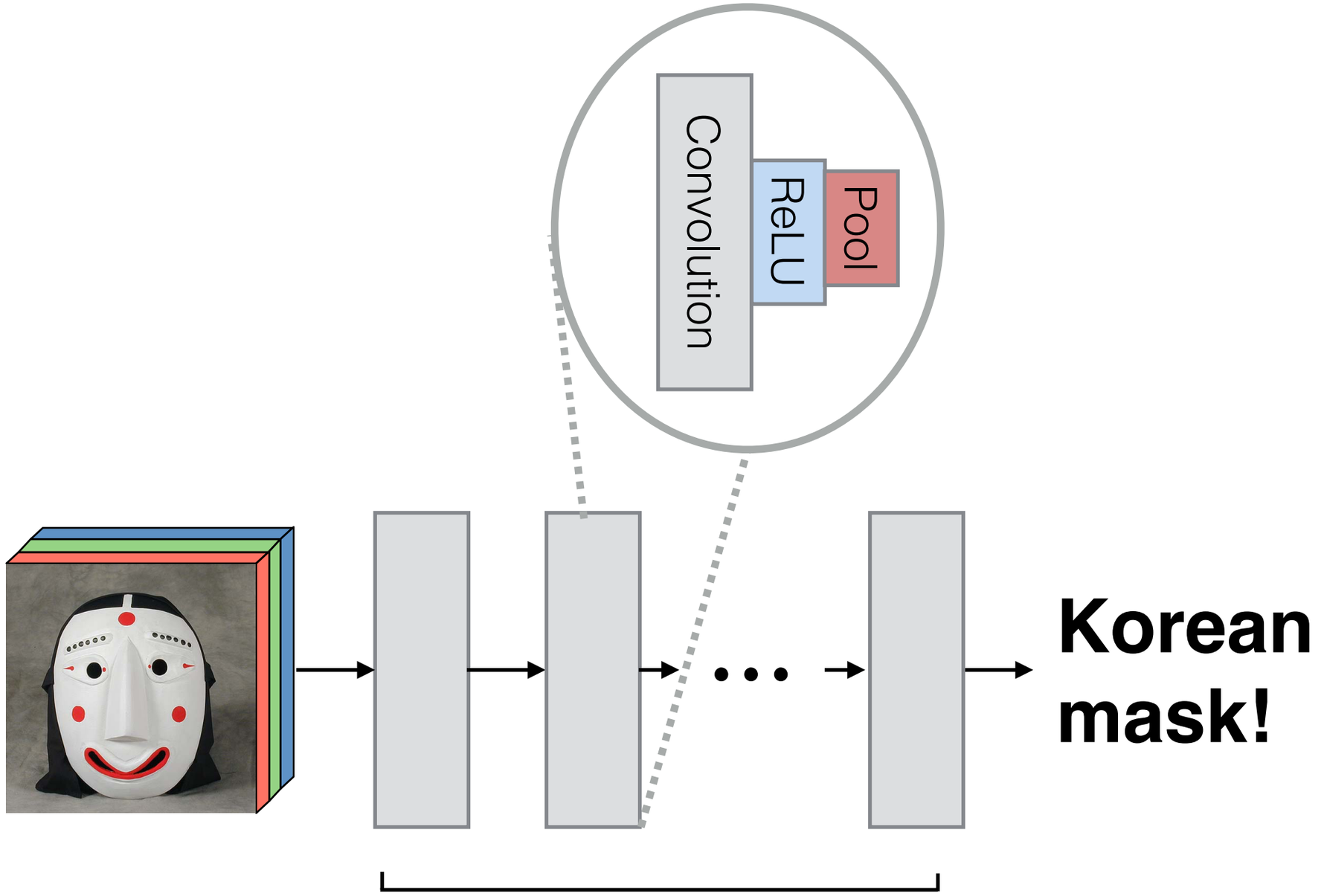
What's a CNN?



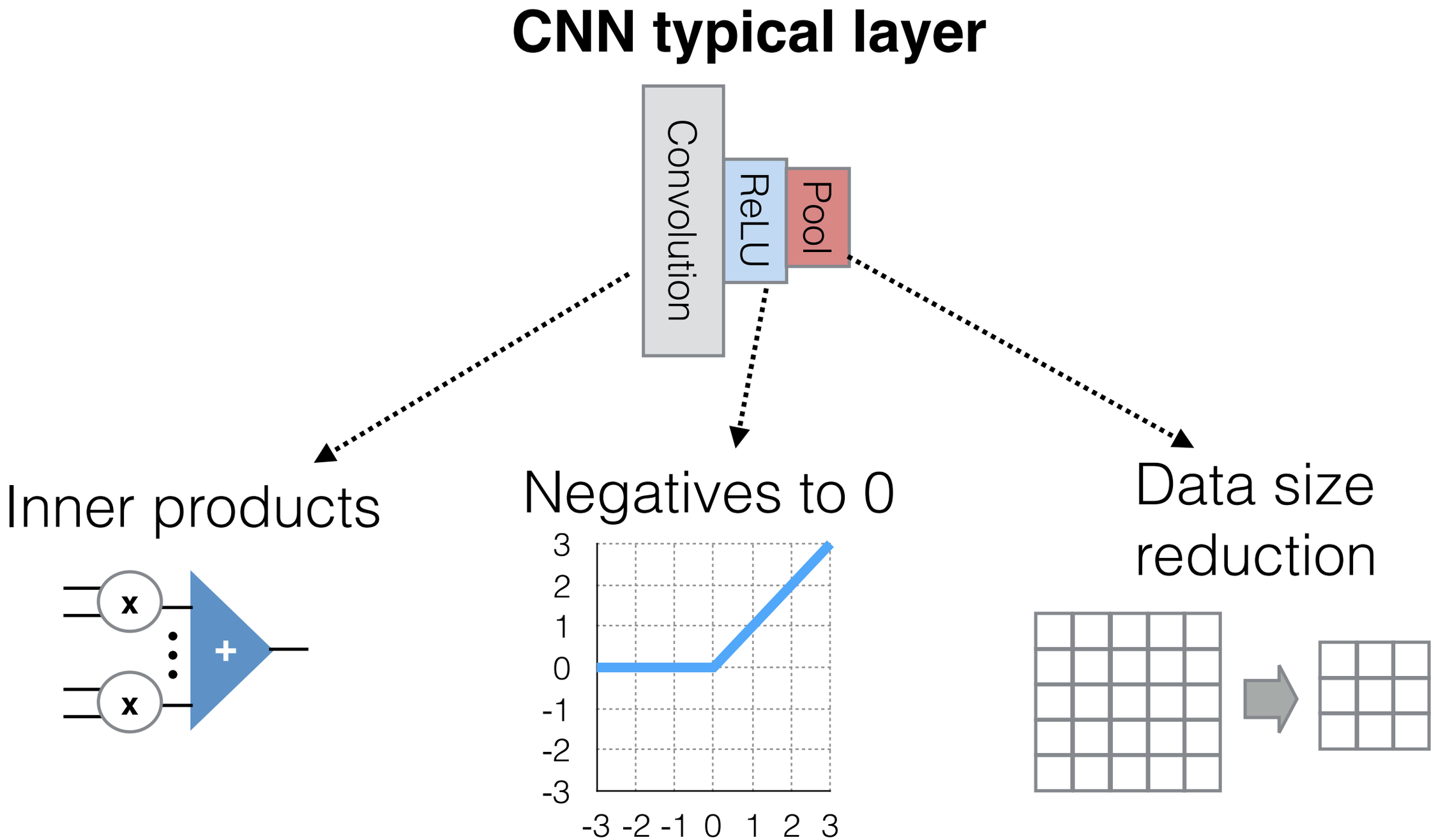
What's a CNN?



What's a CNN?



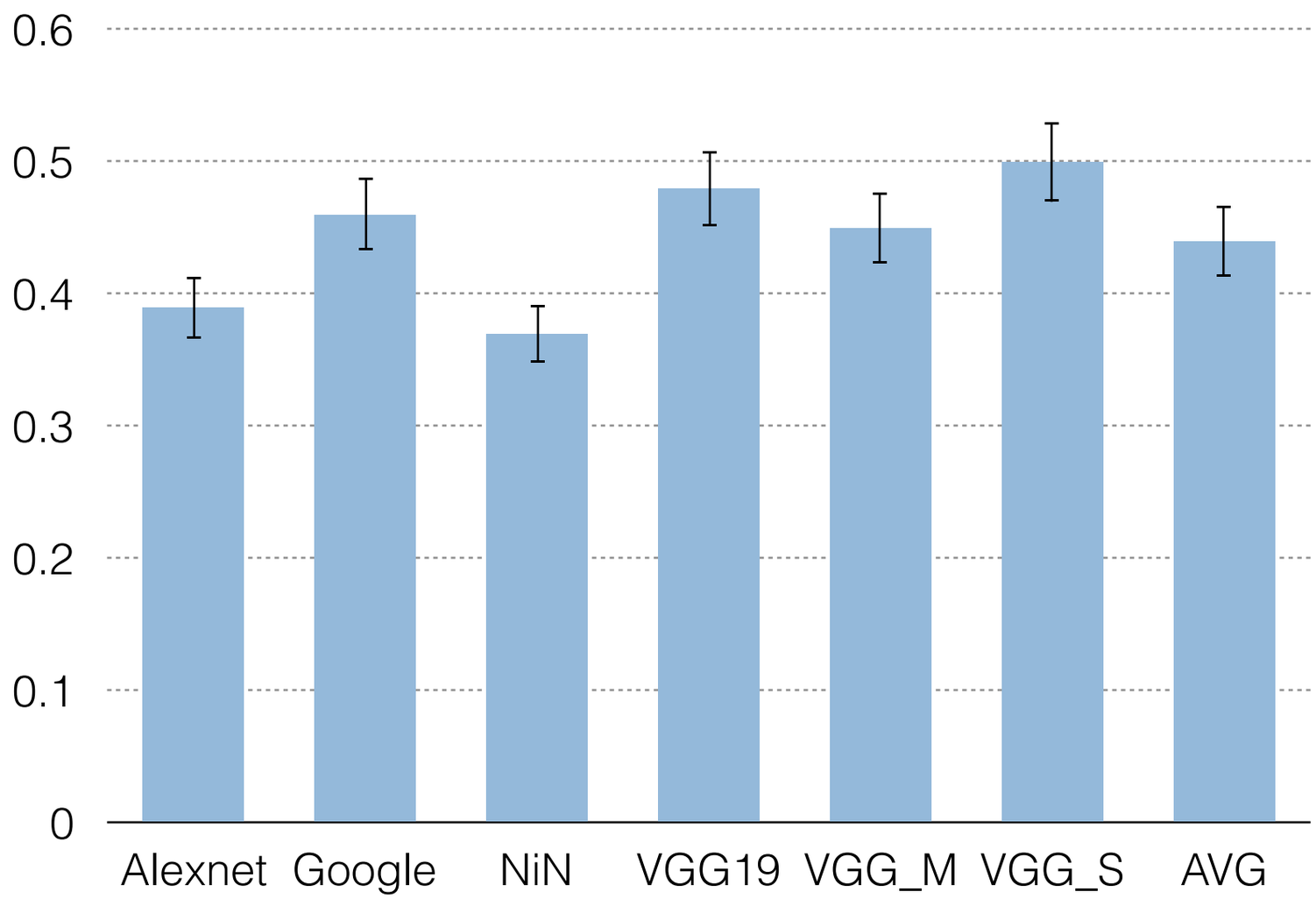
What's a CNN?


CNN typical layer
Convolution
ReLU
Pool
Inner products
x
x
+
Negatives to 0
3
2
1
0
-1
-2
-3
-3 -2 -1 0 1 2 3
Data size reduction

~90%

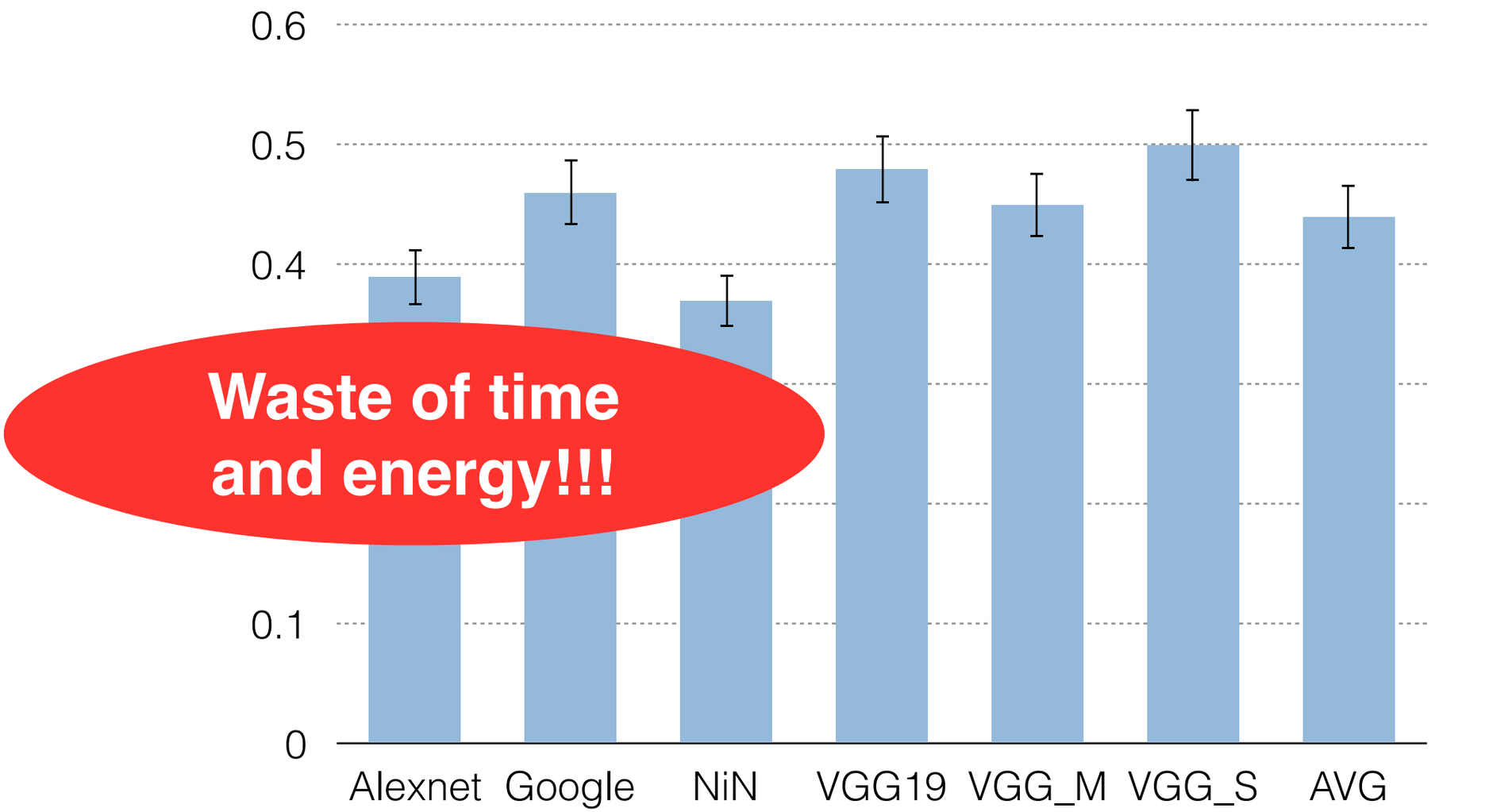
Time spent in convolutions

Lots of Runtime Zeroes



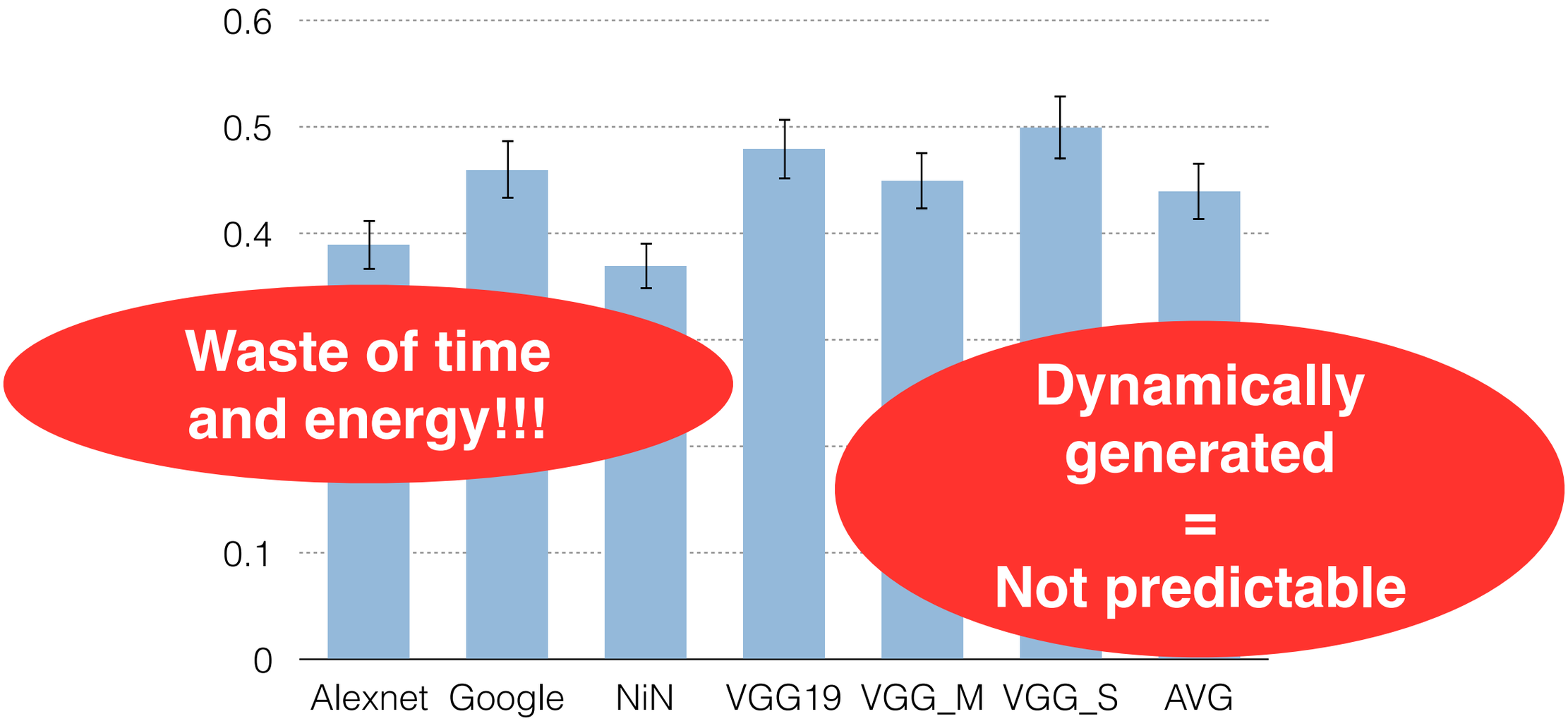
Fraction of zero neurons in multiplications

Lots of Runtime Zeroes



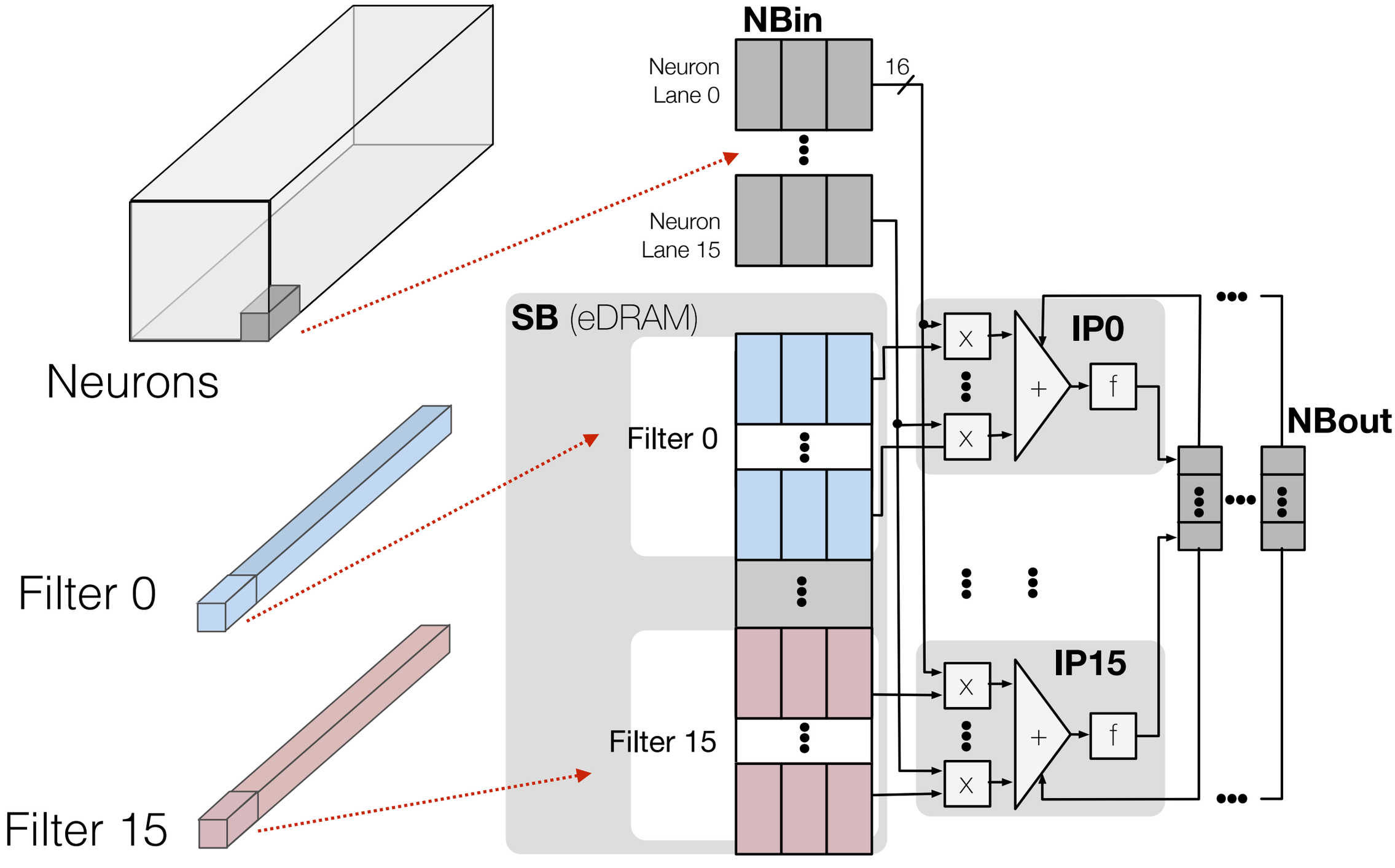
Fraction of zero neurons in multiplications

Lots of Runtime Zeroes



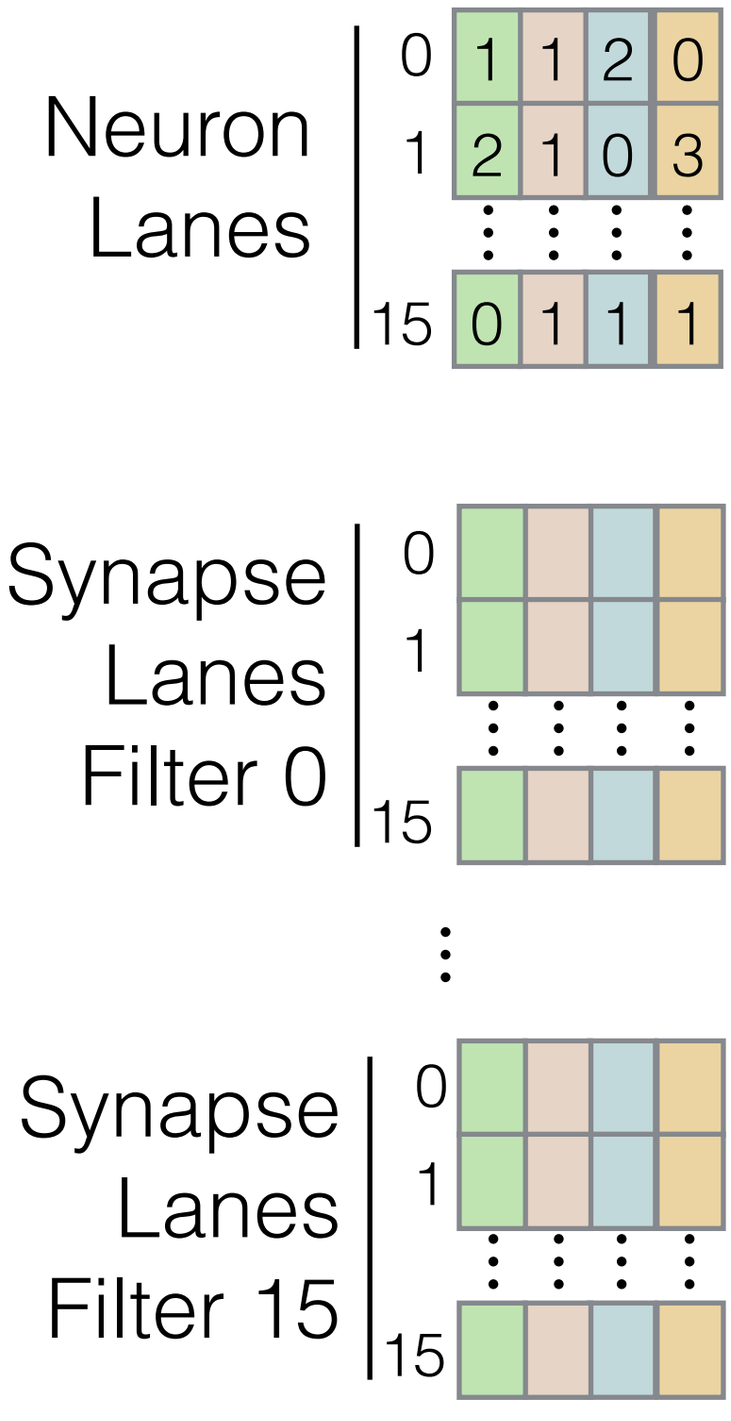
Fraction of zero neurons in multiplications

How to compute DNNs: DaDianNao*

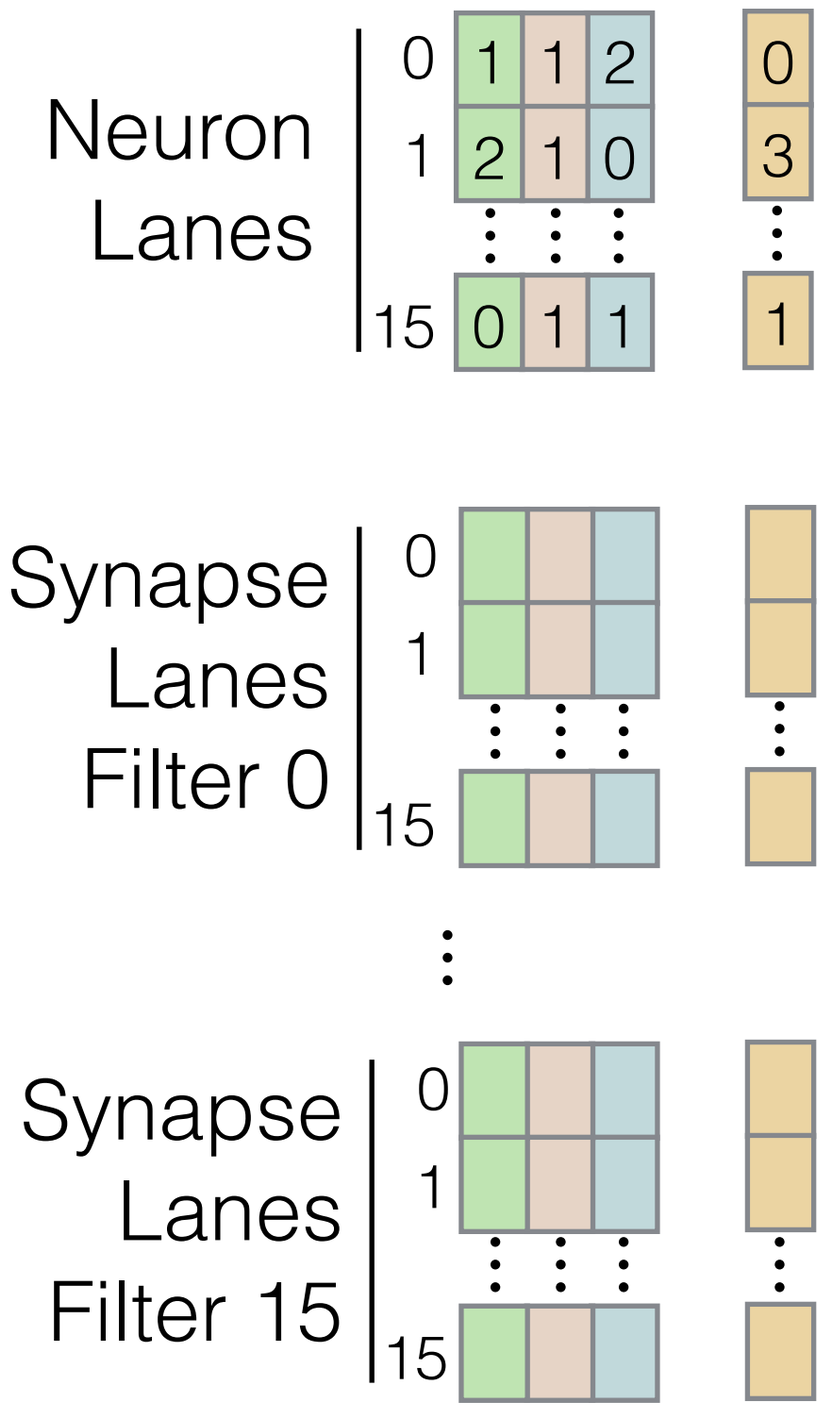


*Chen et al. MICRO 2014

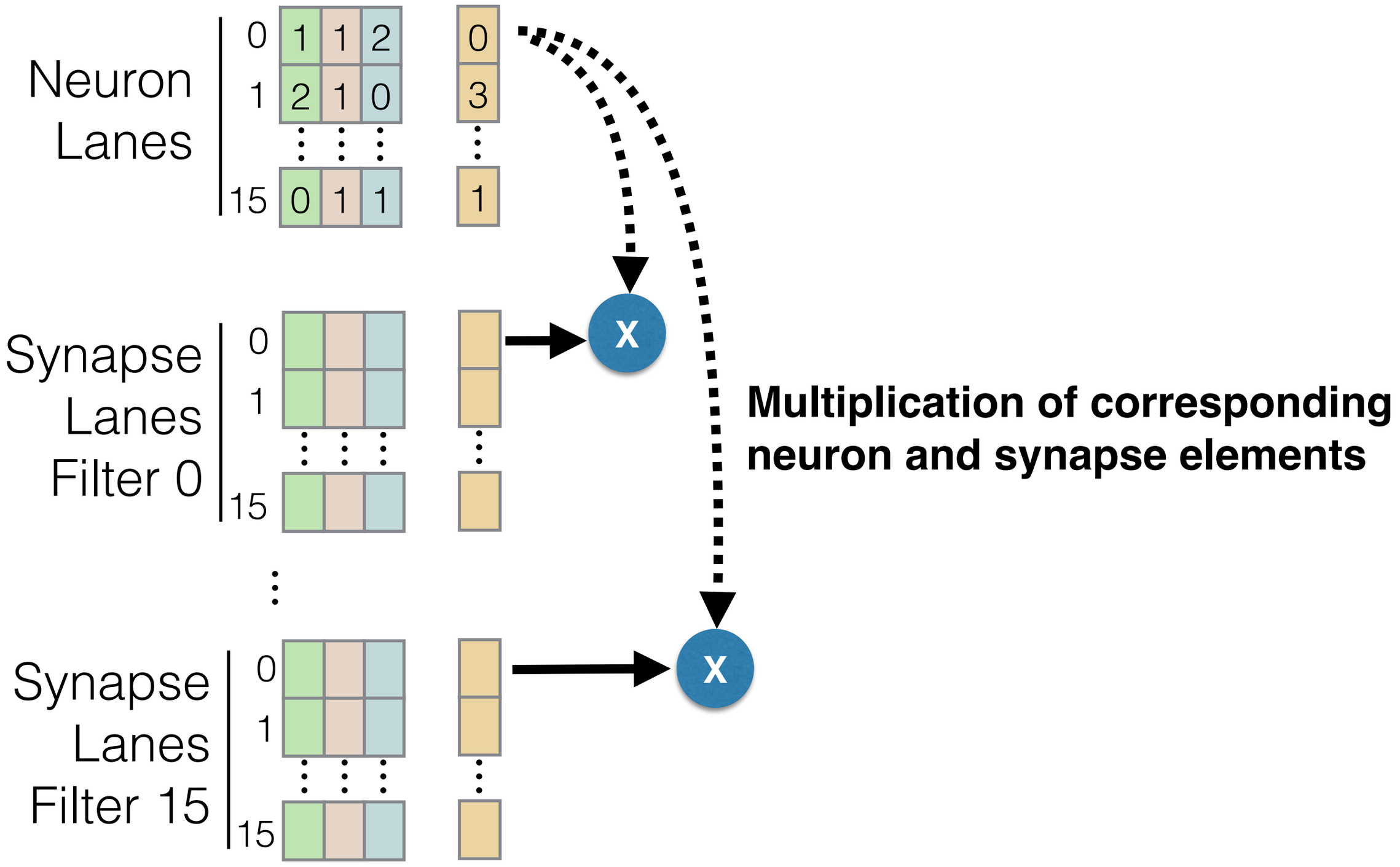
Processing in DaDianNao



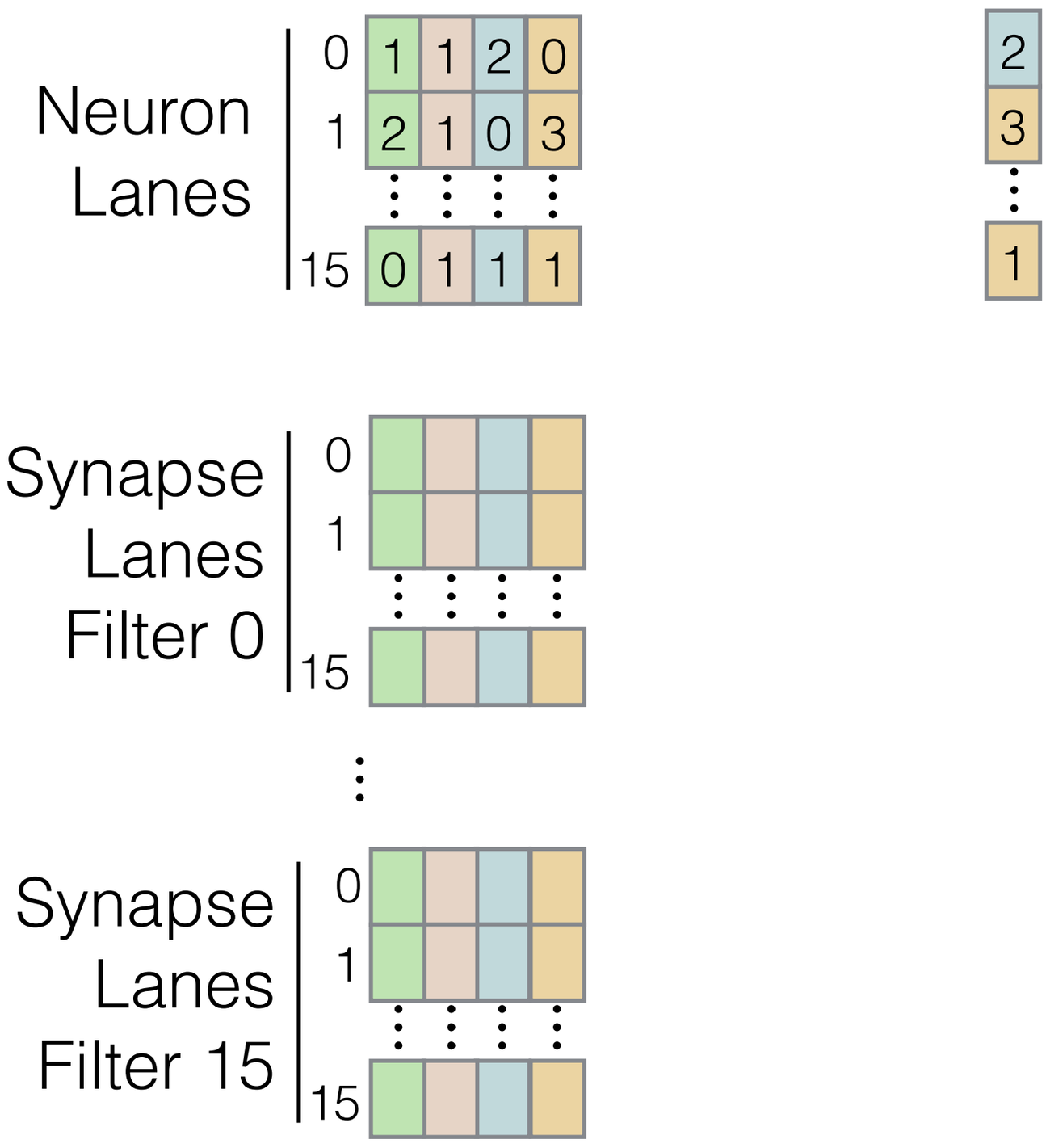
Processing in DaDianNao



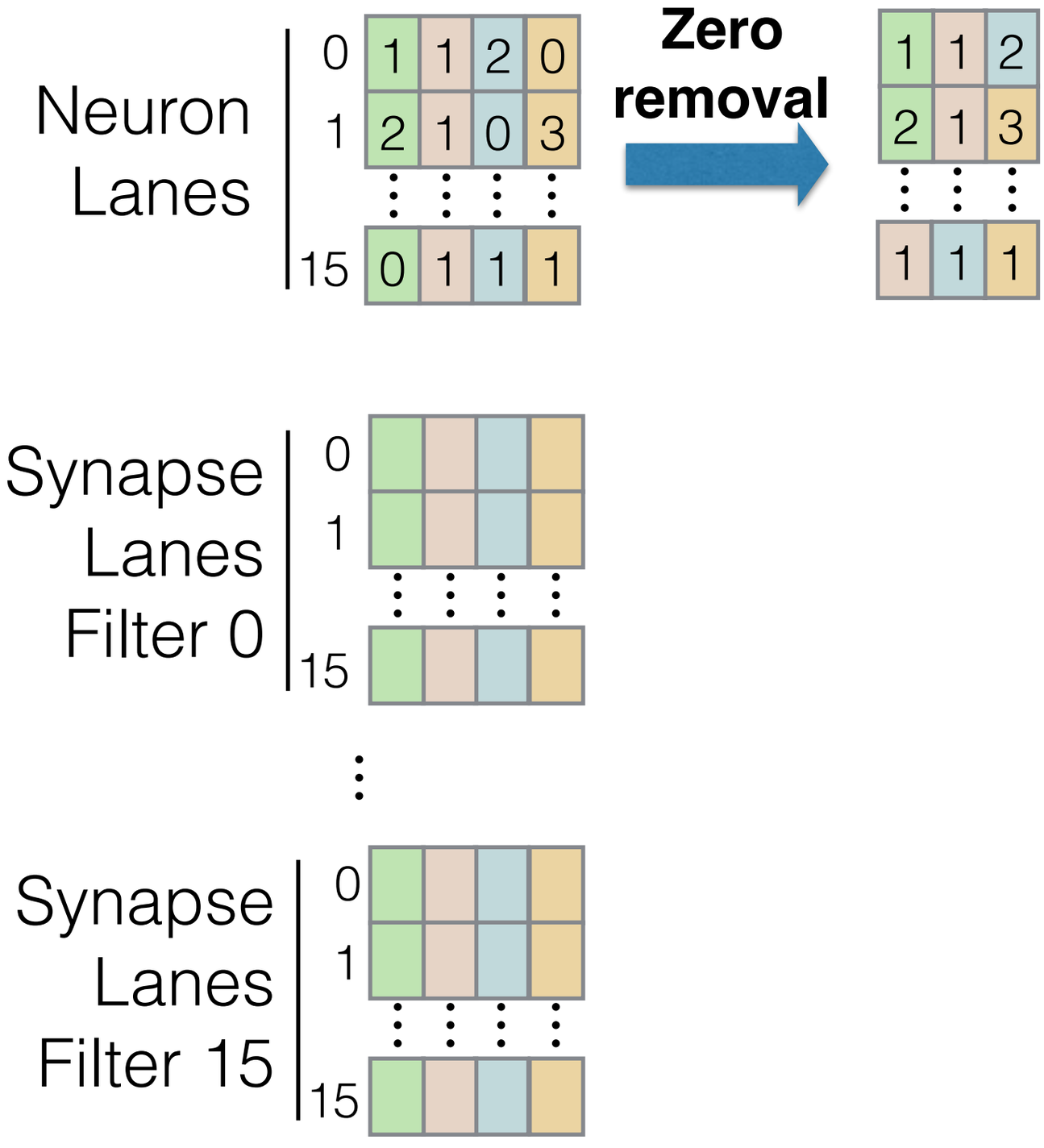
Processing in DaDianNao



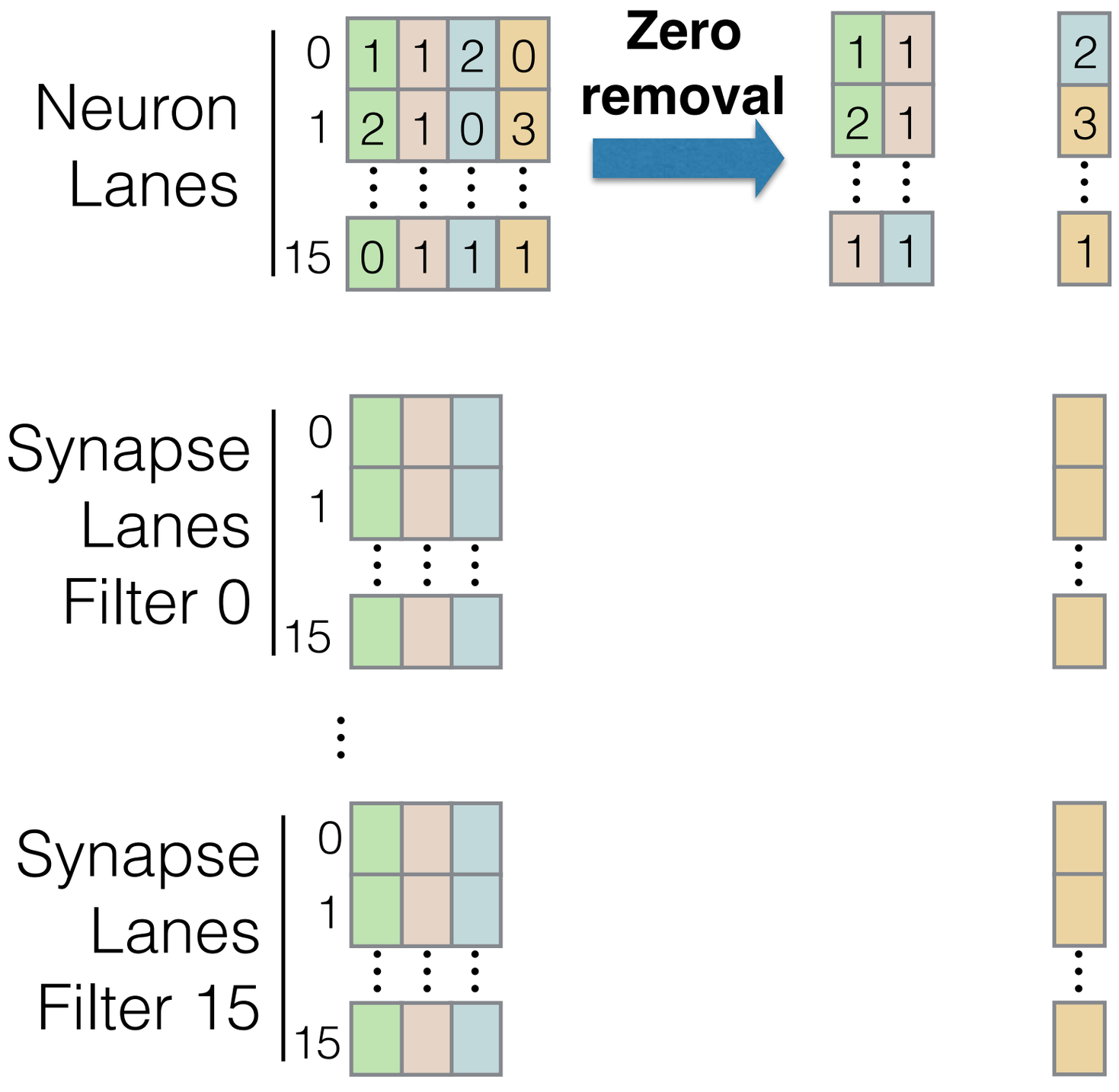
Zero-skipping in DaDianNao?



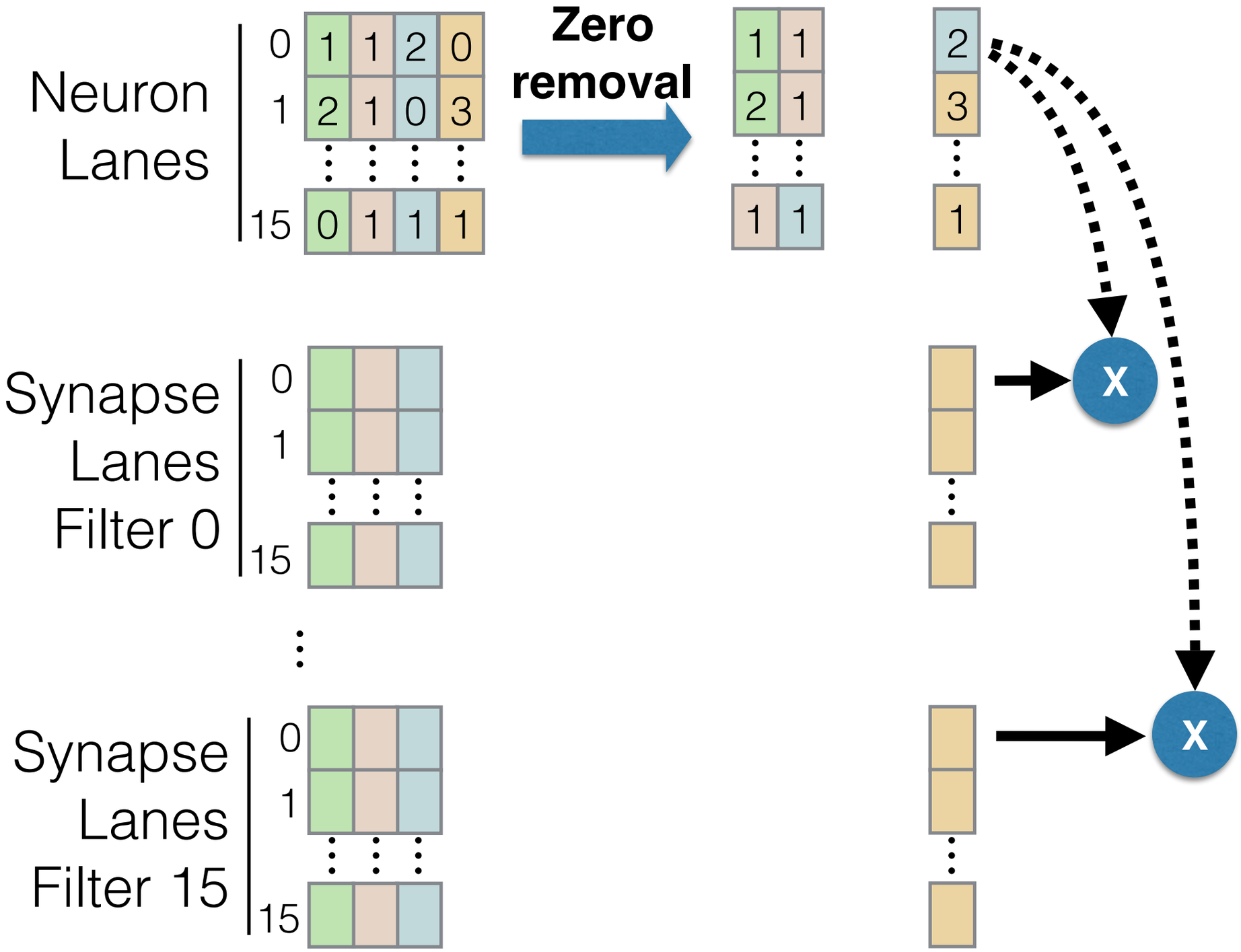
Zero-skipping in DaDianNao?



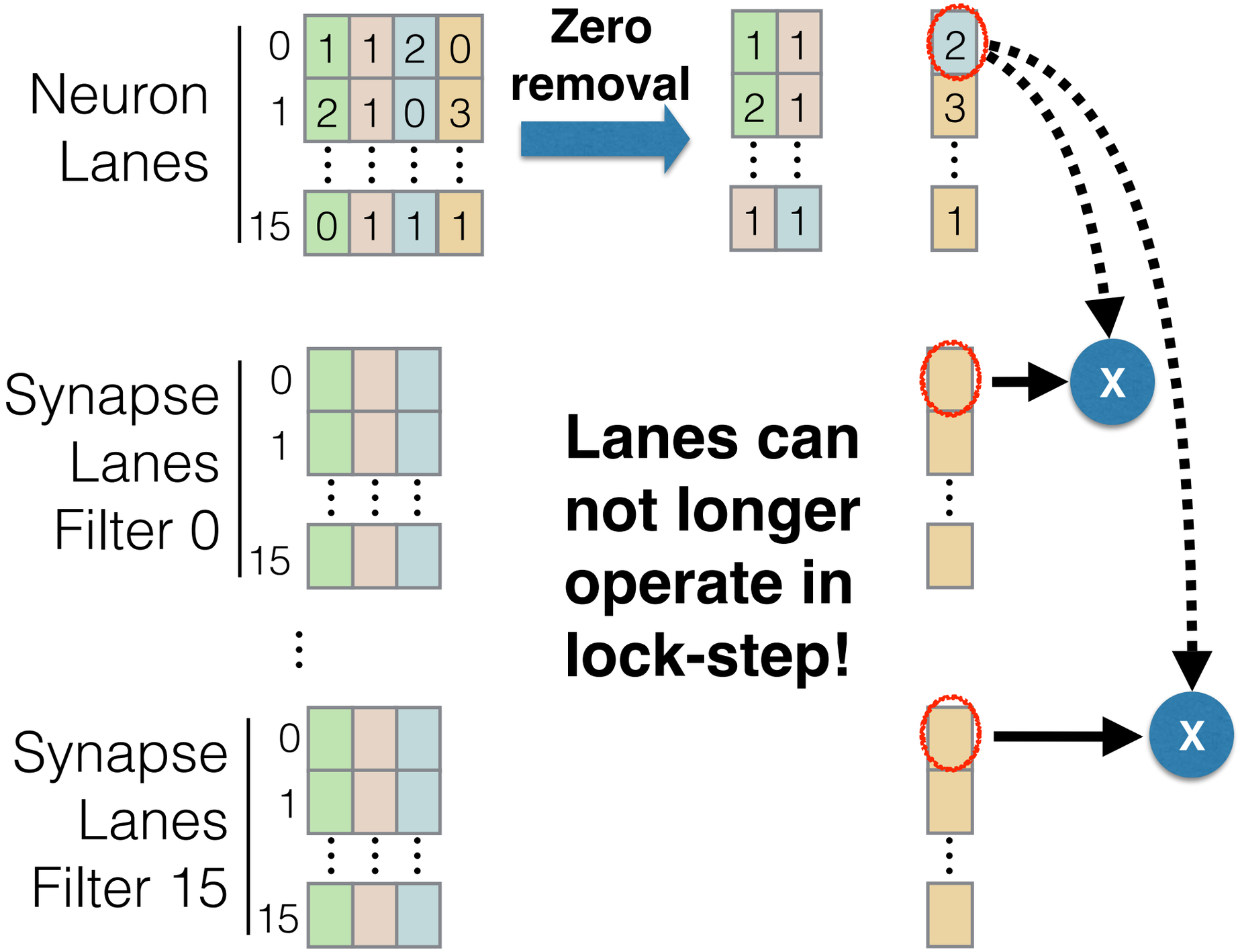
Zero-skipping in DaDianNao?



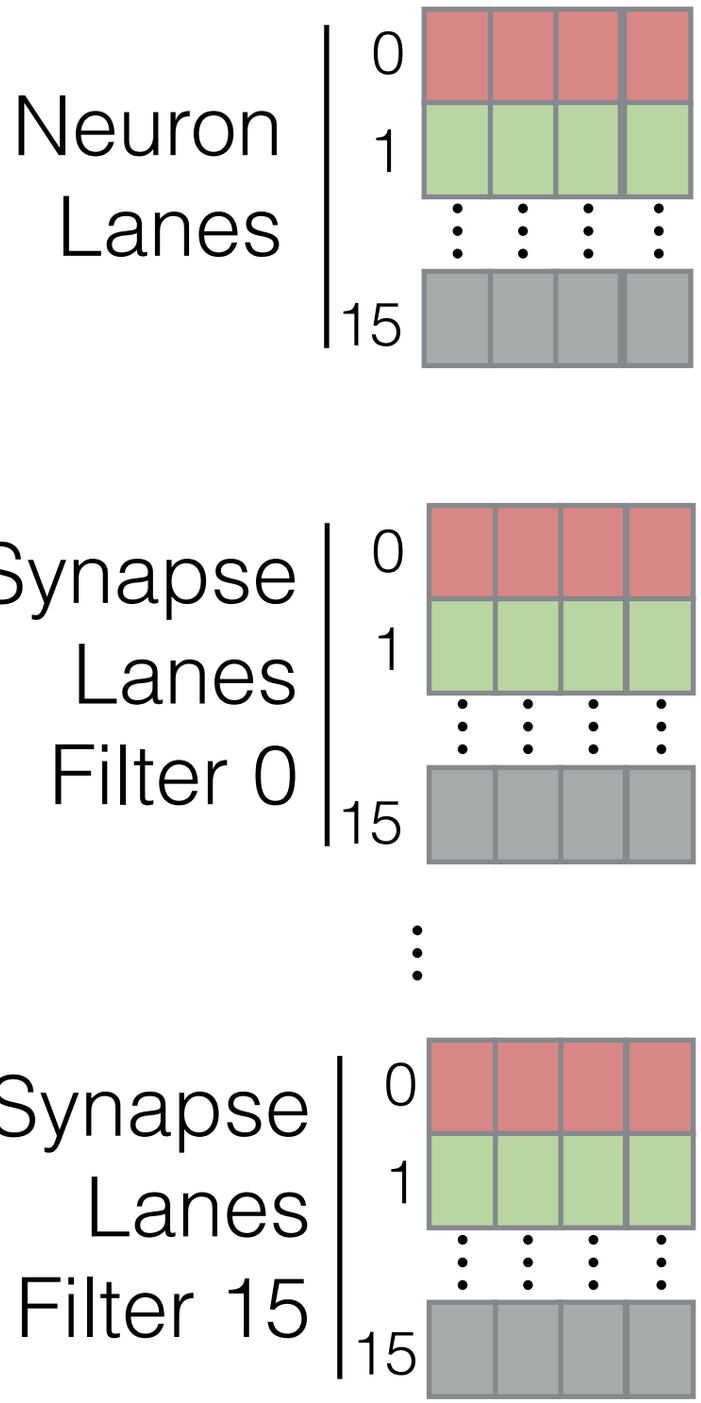
Zero-skipping in DaDianNao?

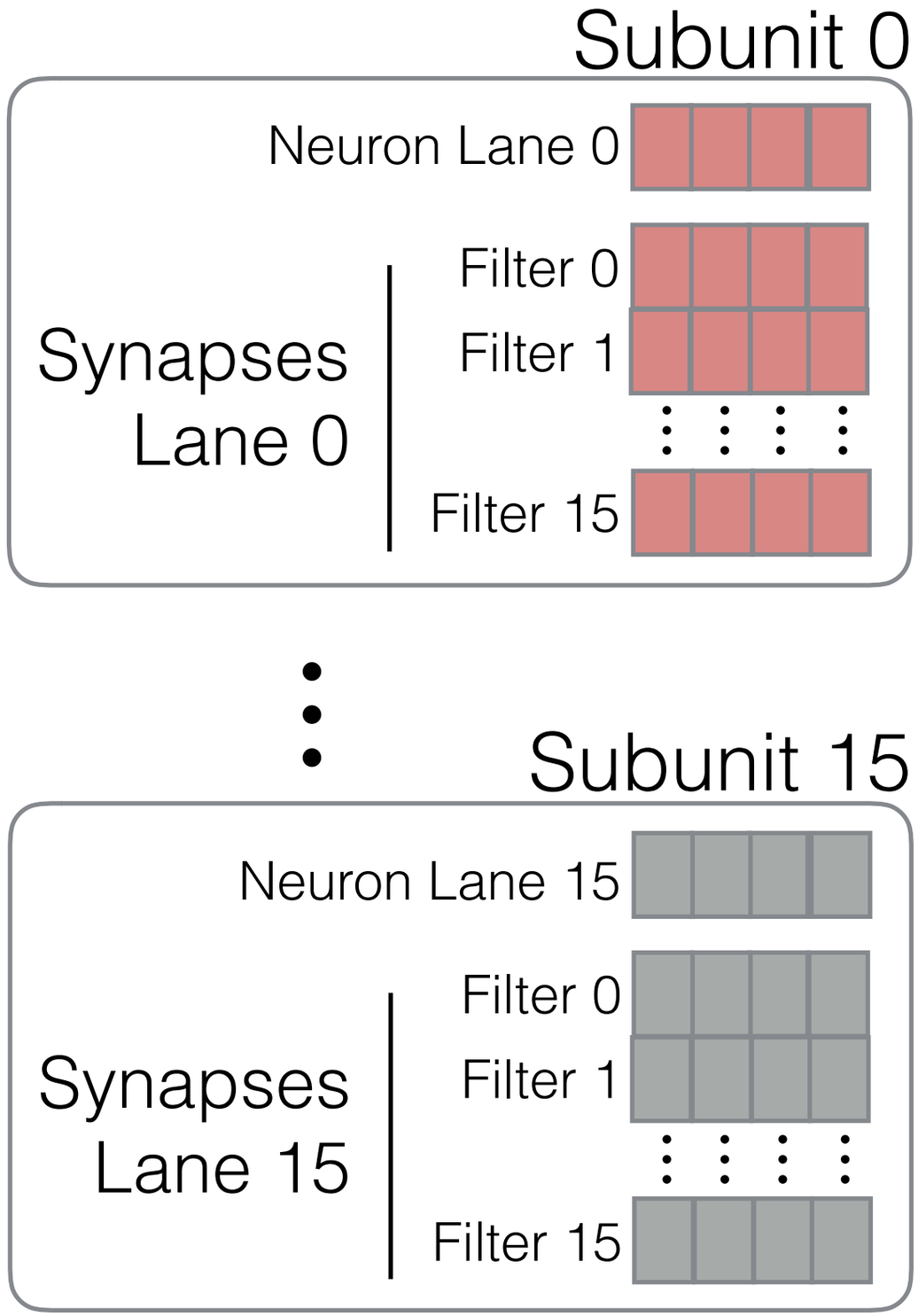


Zero-skipping in DaDianNao?



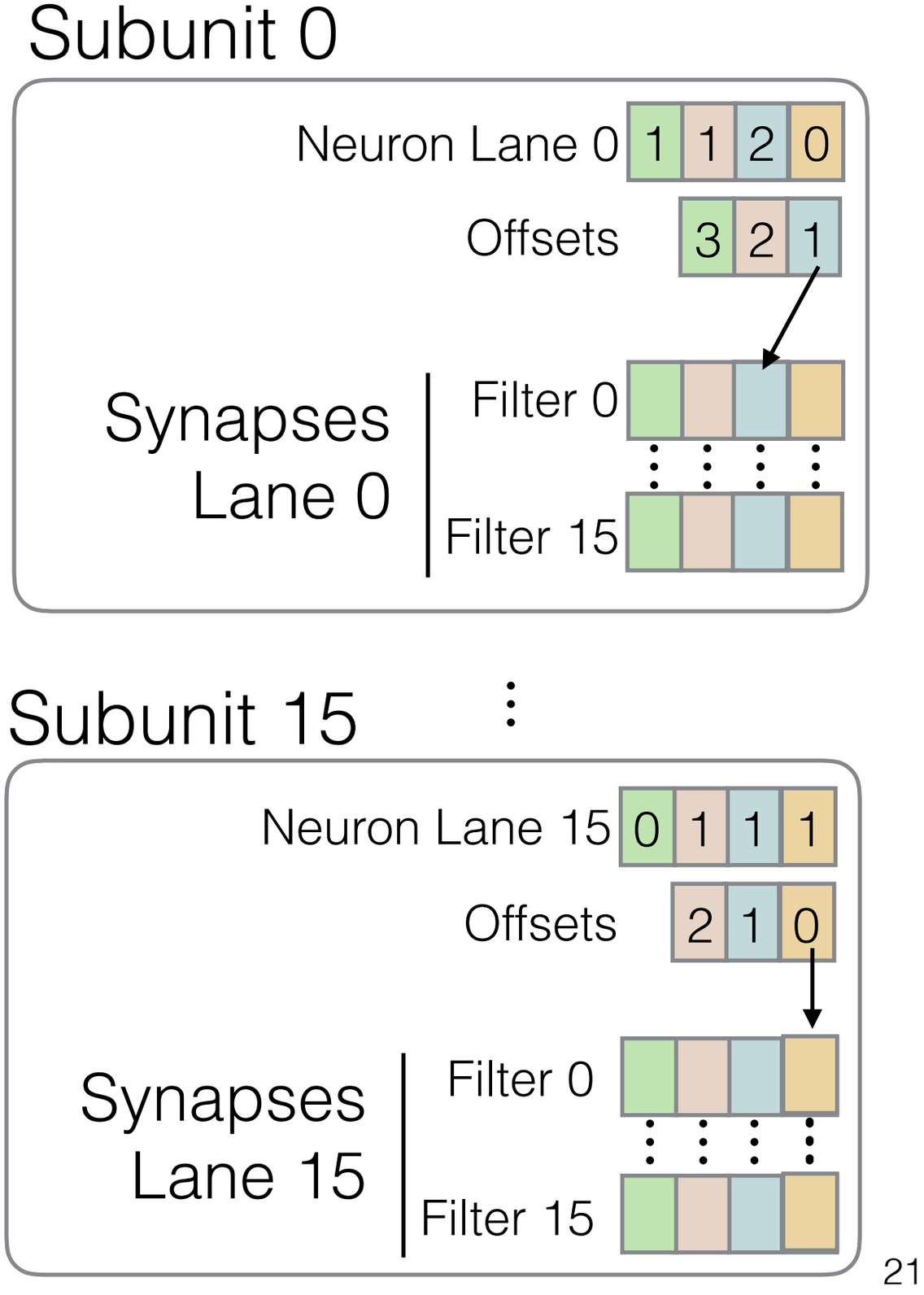
CNVLUTIN: Decoupling Lanes



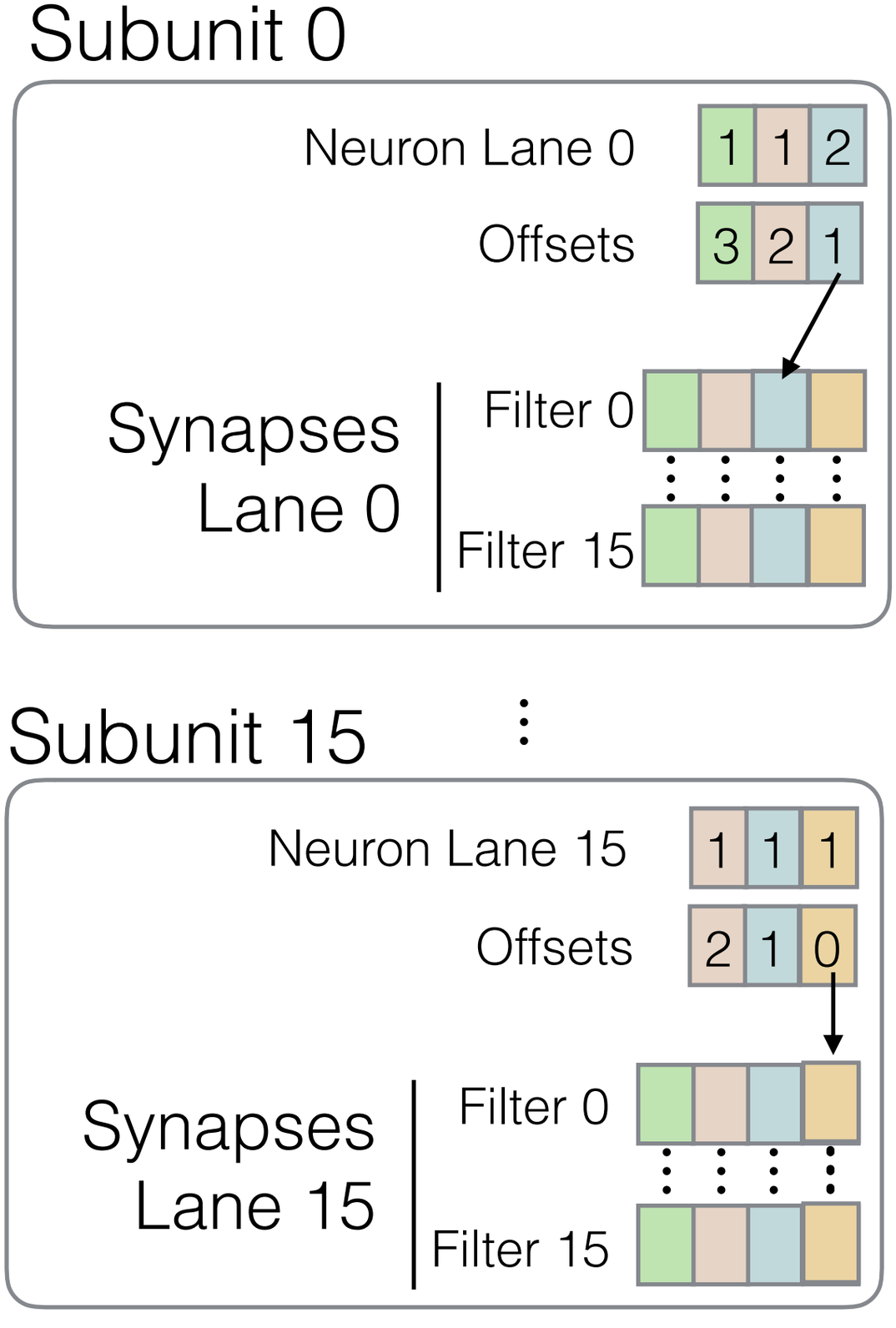


DaDianNao **CNVLUTIN**

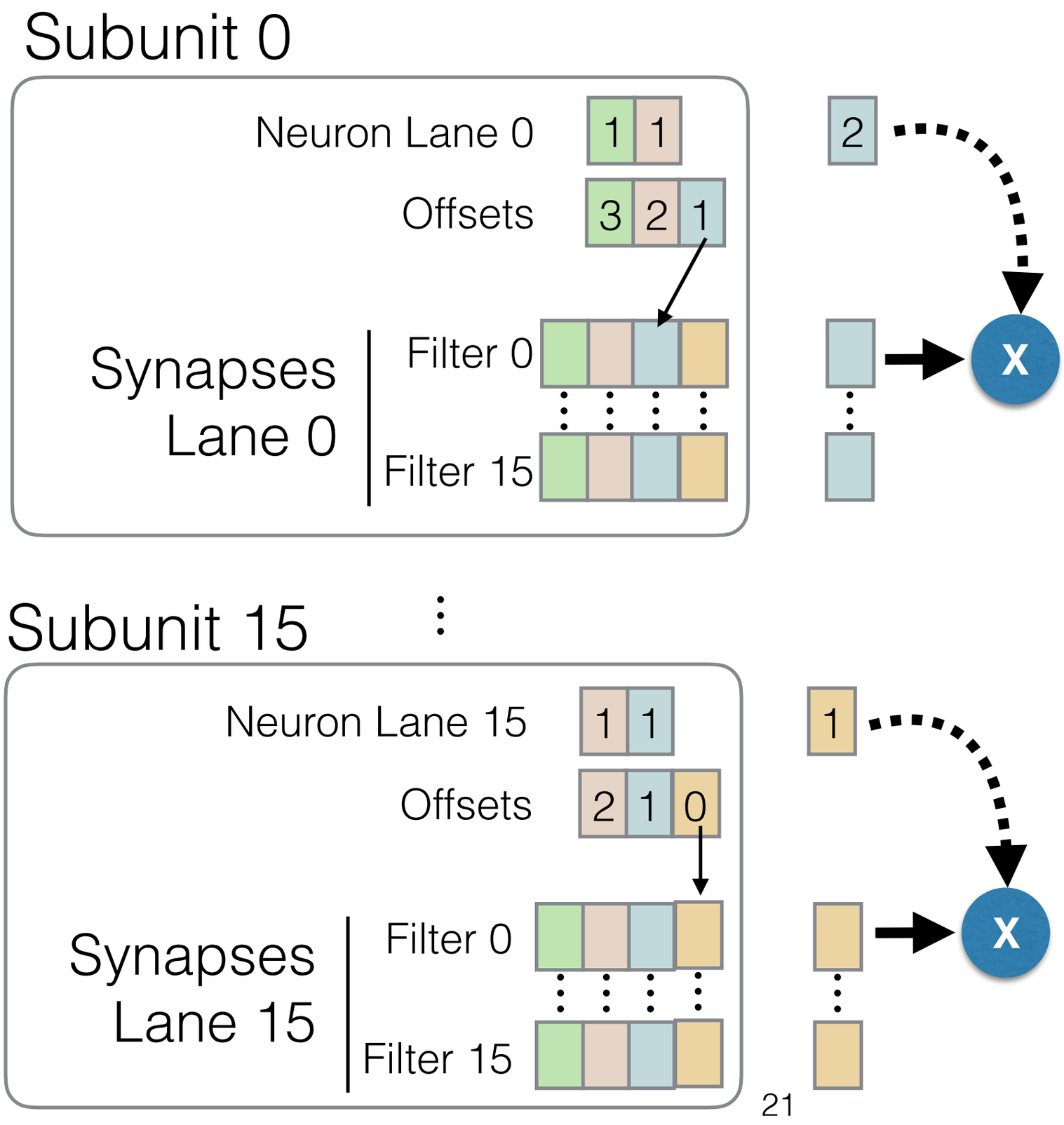
CNVLUTIN: Decoupling Lanes



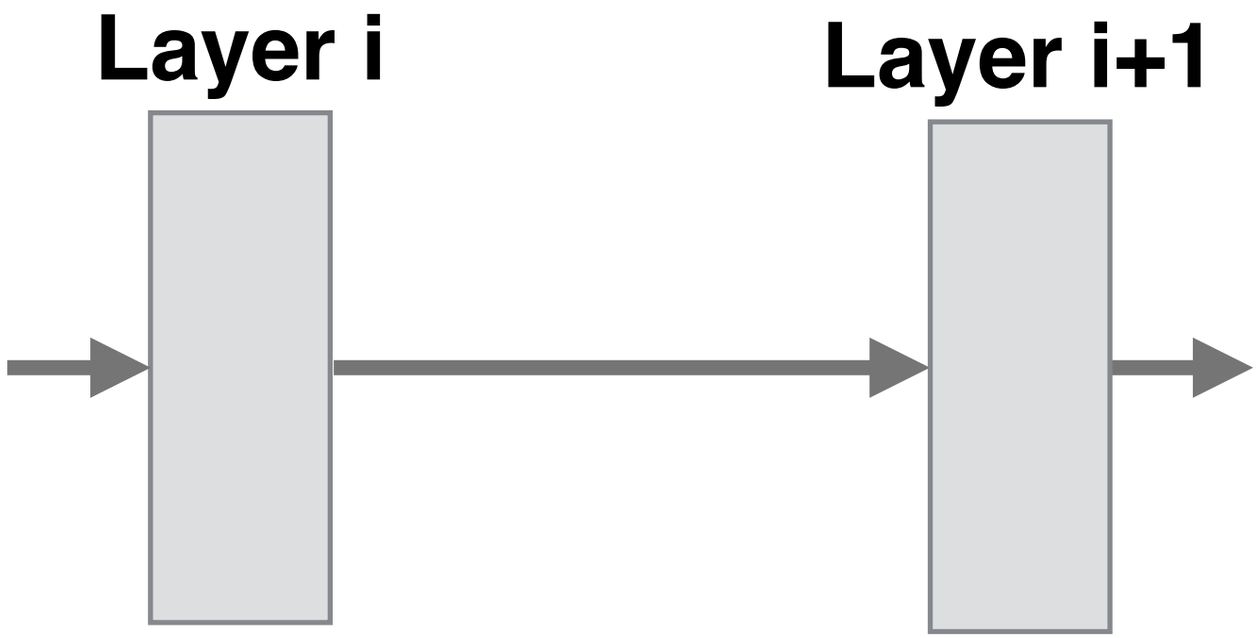
CNVLUTIN: Decoupling Lanes



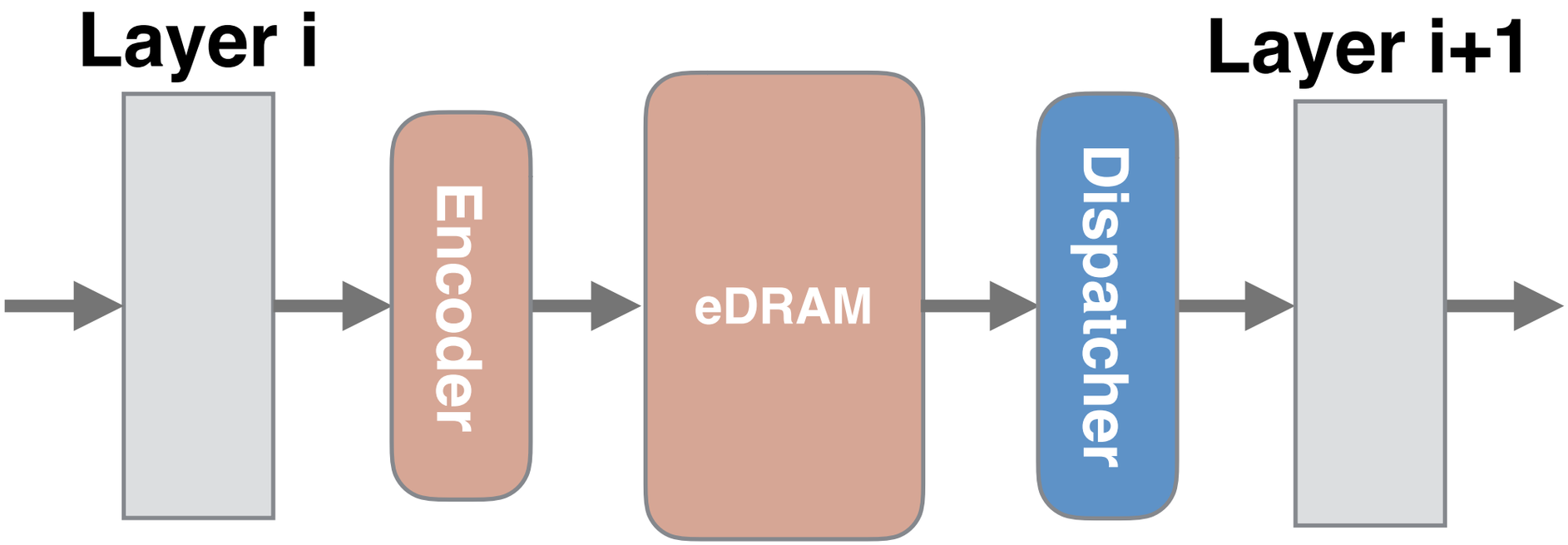
CNVLUTIN: Decoupling Lanes



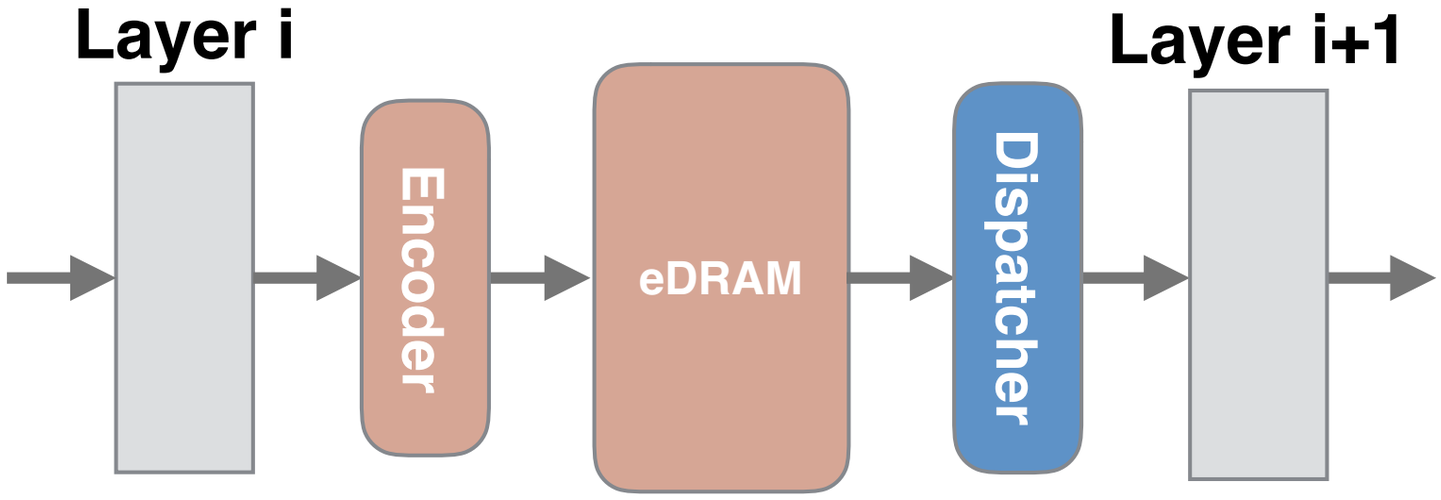
CNVLUTIN: Ineffectual-neuron Filtering



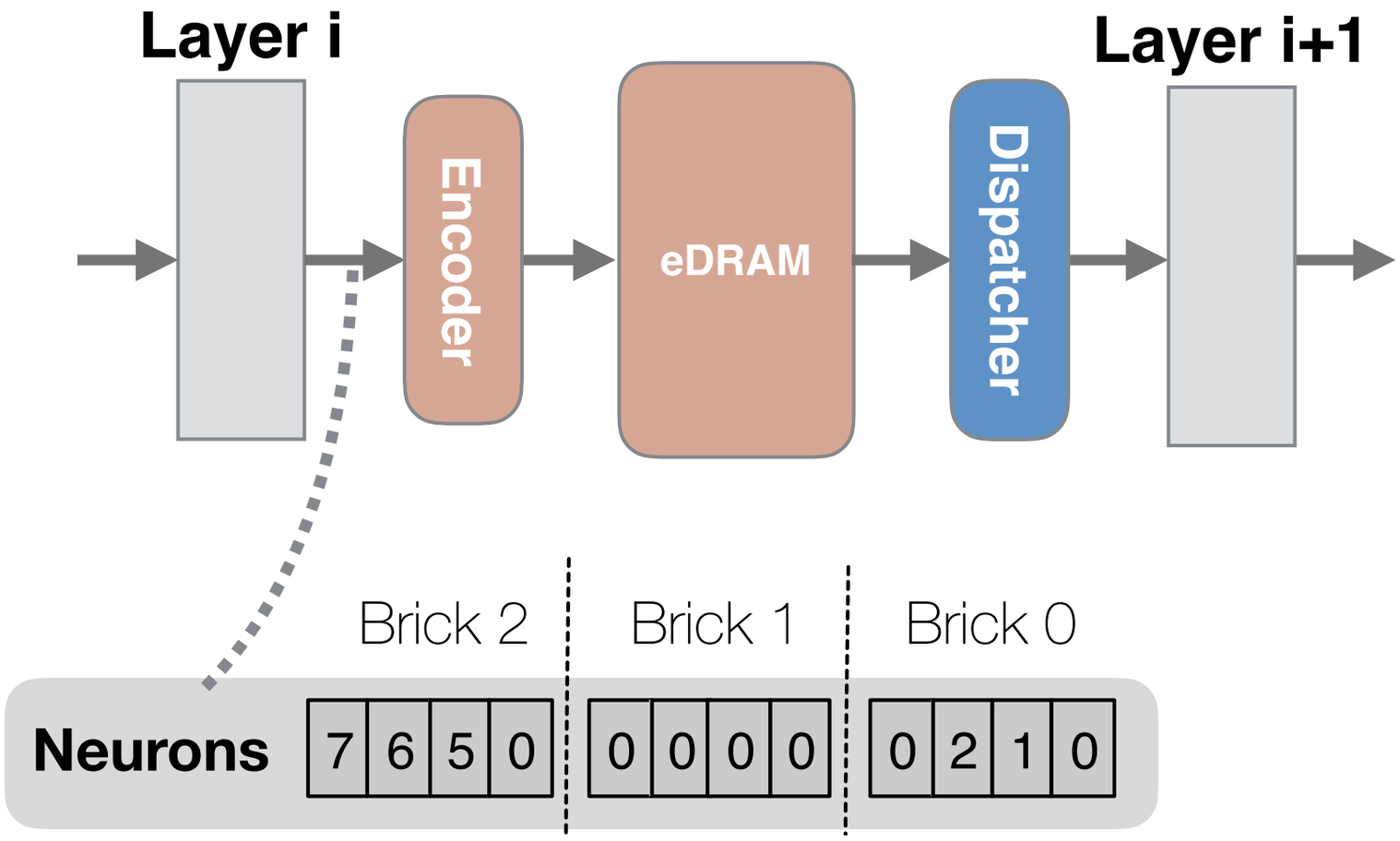
CNVLUTIN: Ineffectual-neuron Filtering



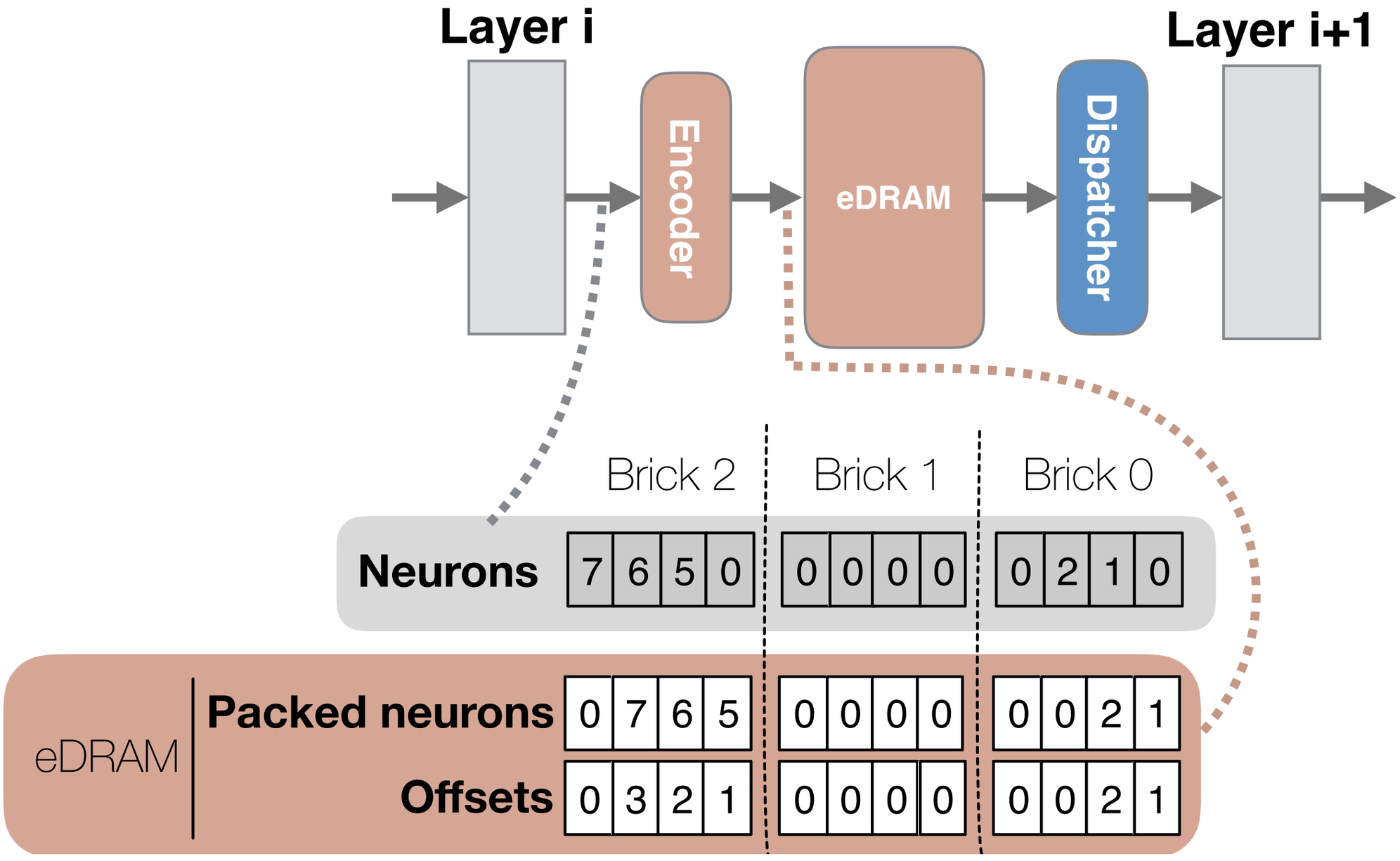
CNVLUTIN: Ineffectual-neuron Filtering



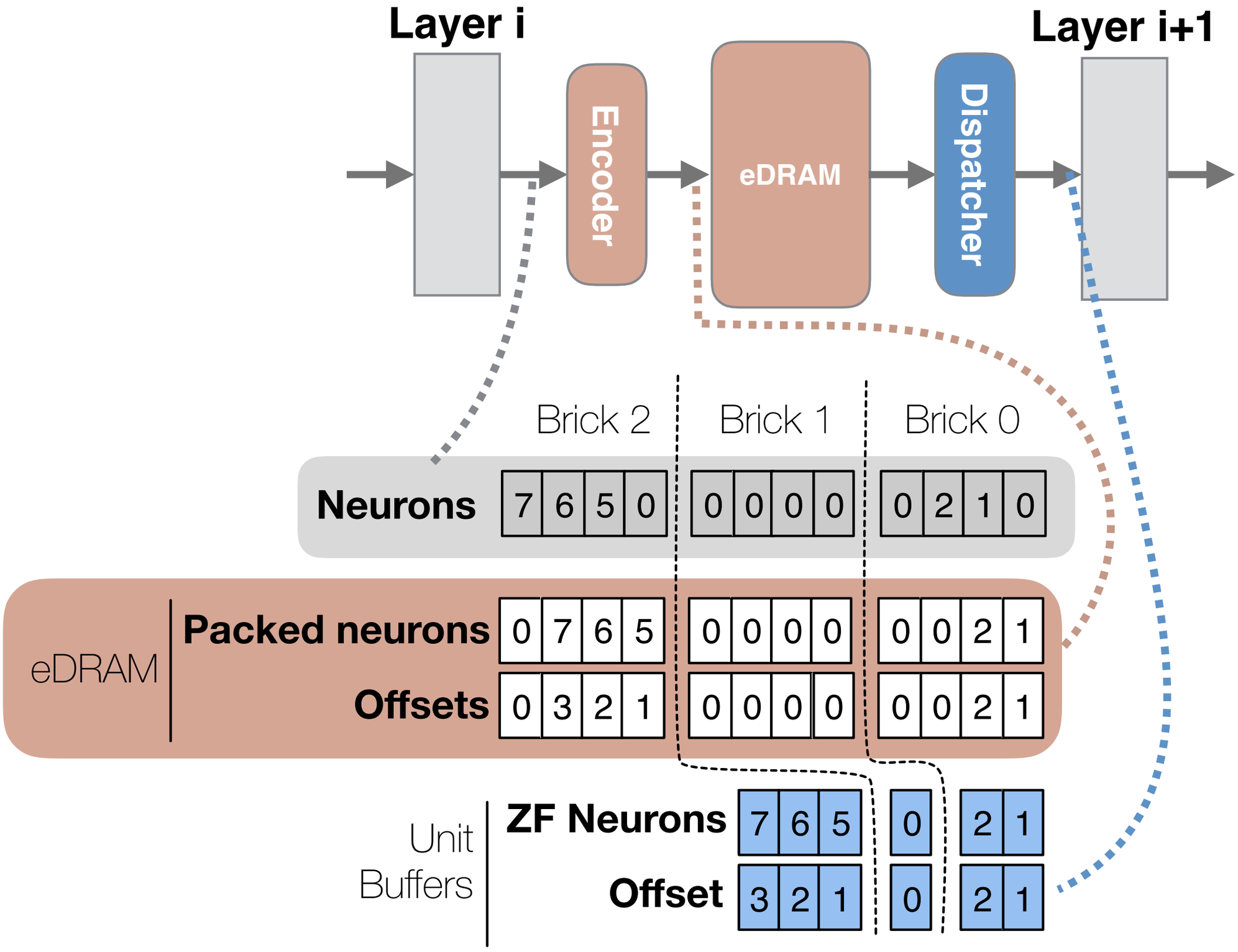
CNVLUTIN: Ineffectual-neuron Filtering



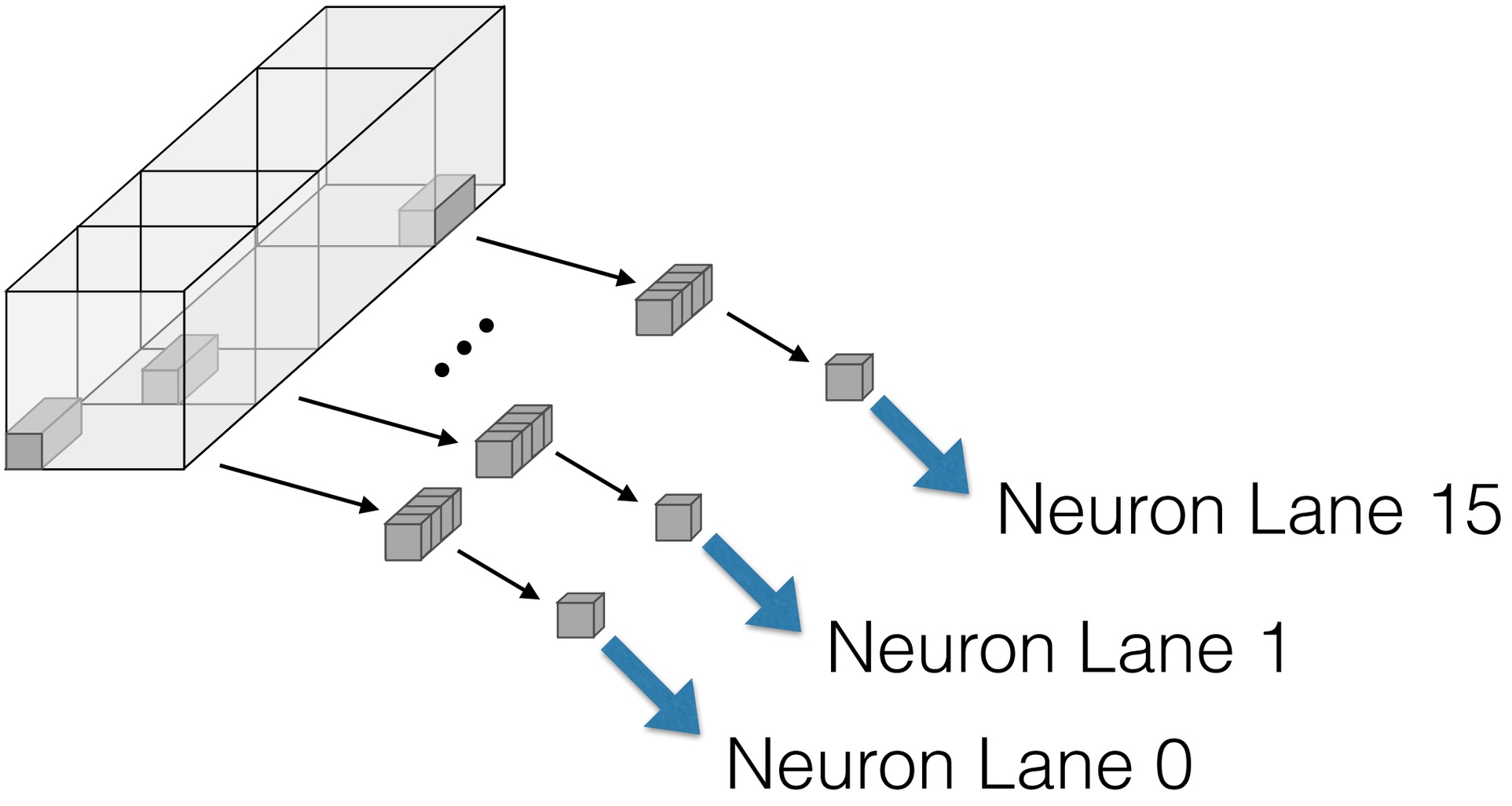
CNVLUTIN: Ineffectual-neuron Filtering



CNVLUTIN: Ineffectual-neuron Filtering



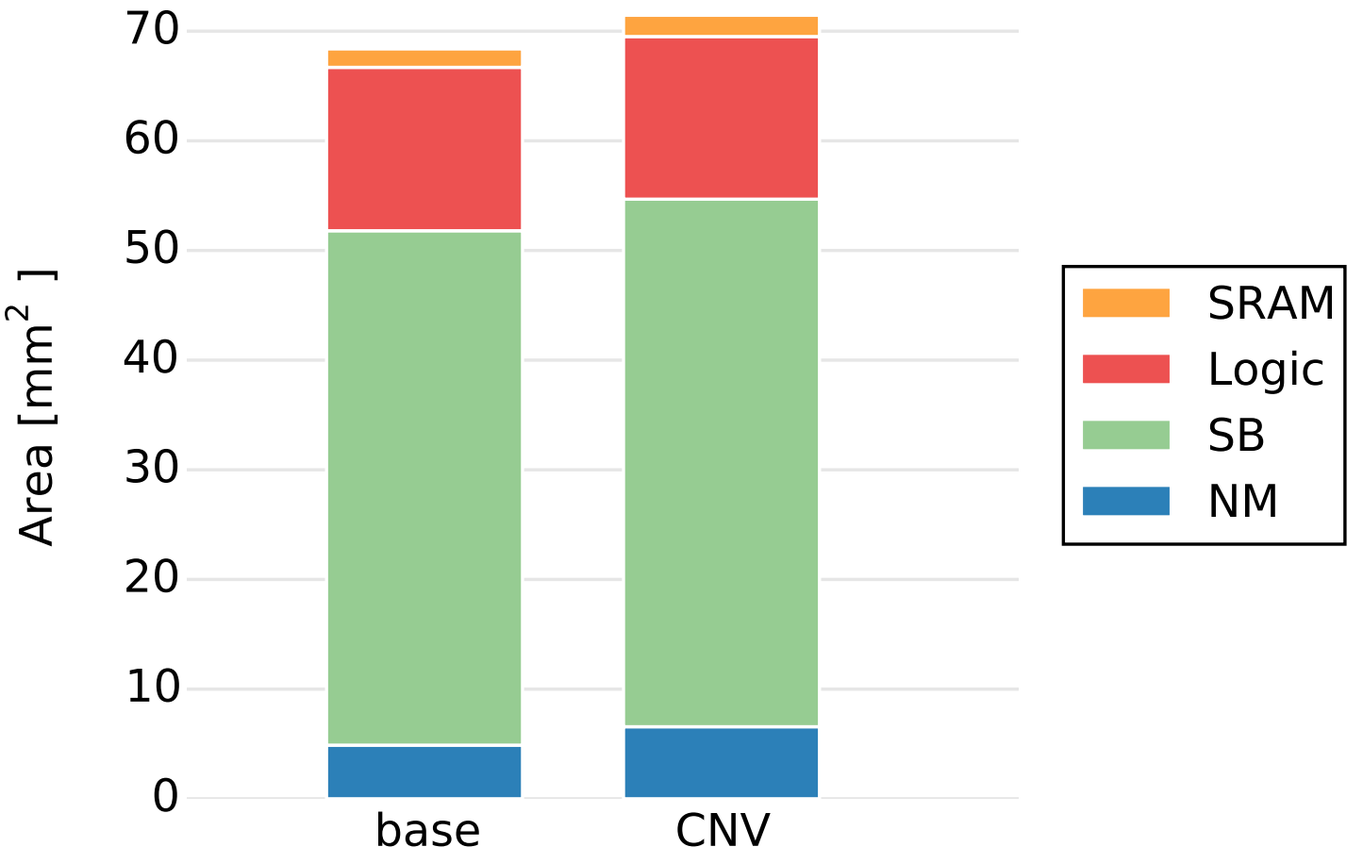
CNVLUTIN: Computation Slicing



Methodology

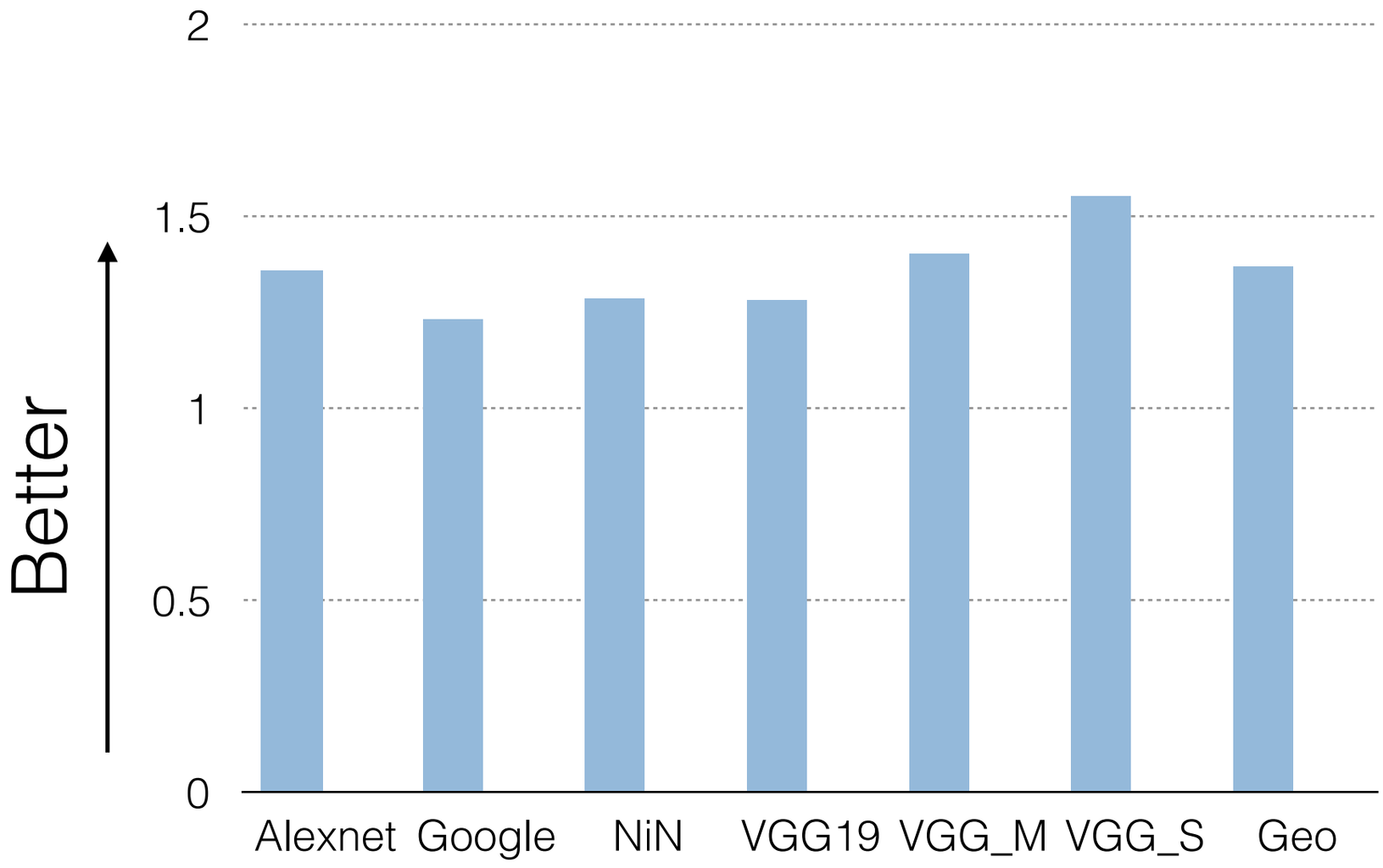
- In-house timing simulator: baseline + CNVLUTIN
- Logic + SRAM: Synthesis on 65nm TSMC
- eDRAM model: Destiny
- DNNs: Trained models from Caffe model zoo

Area

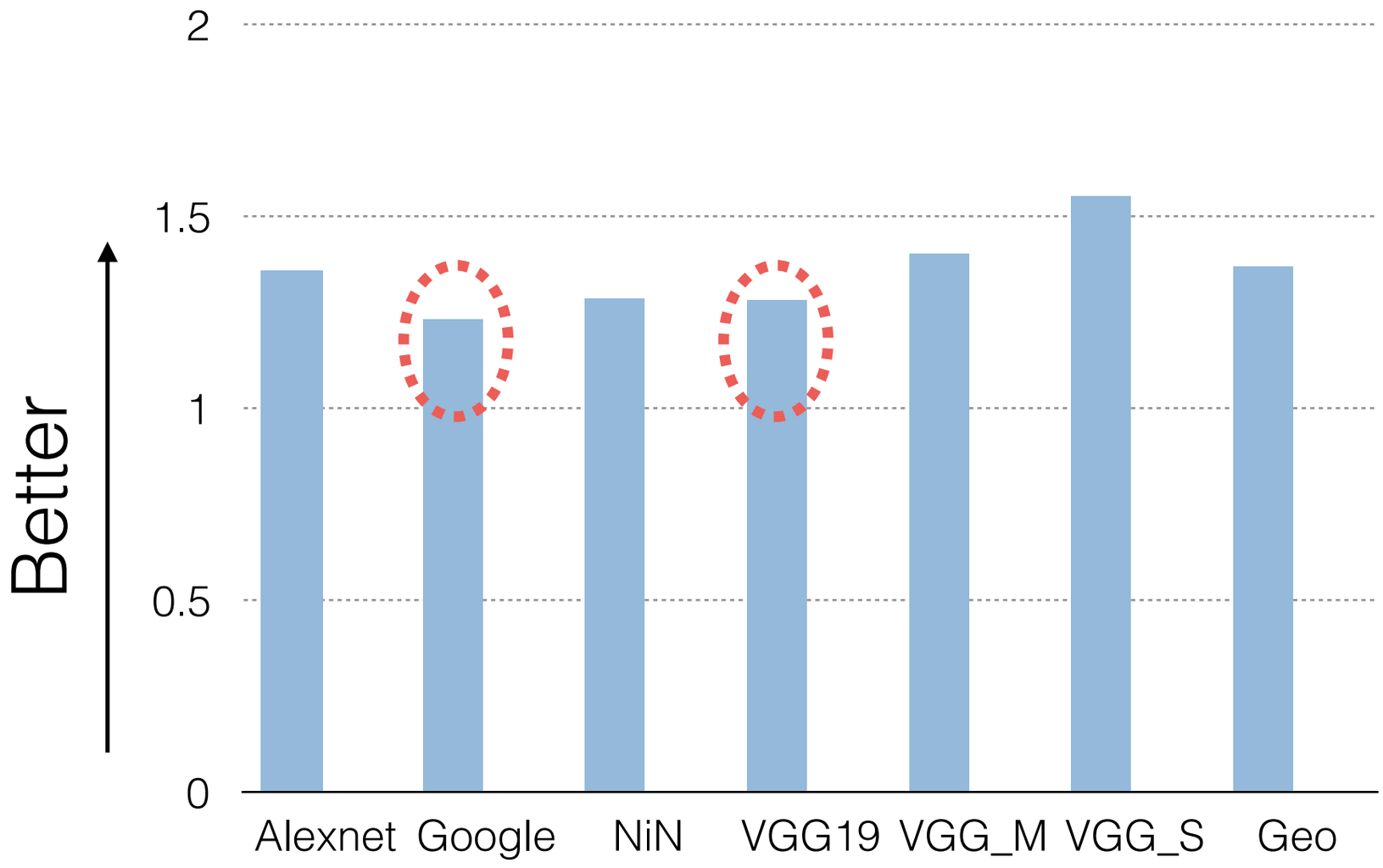


Only +4.5% in area overhead

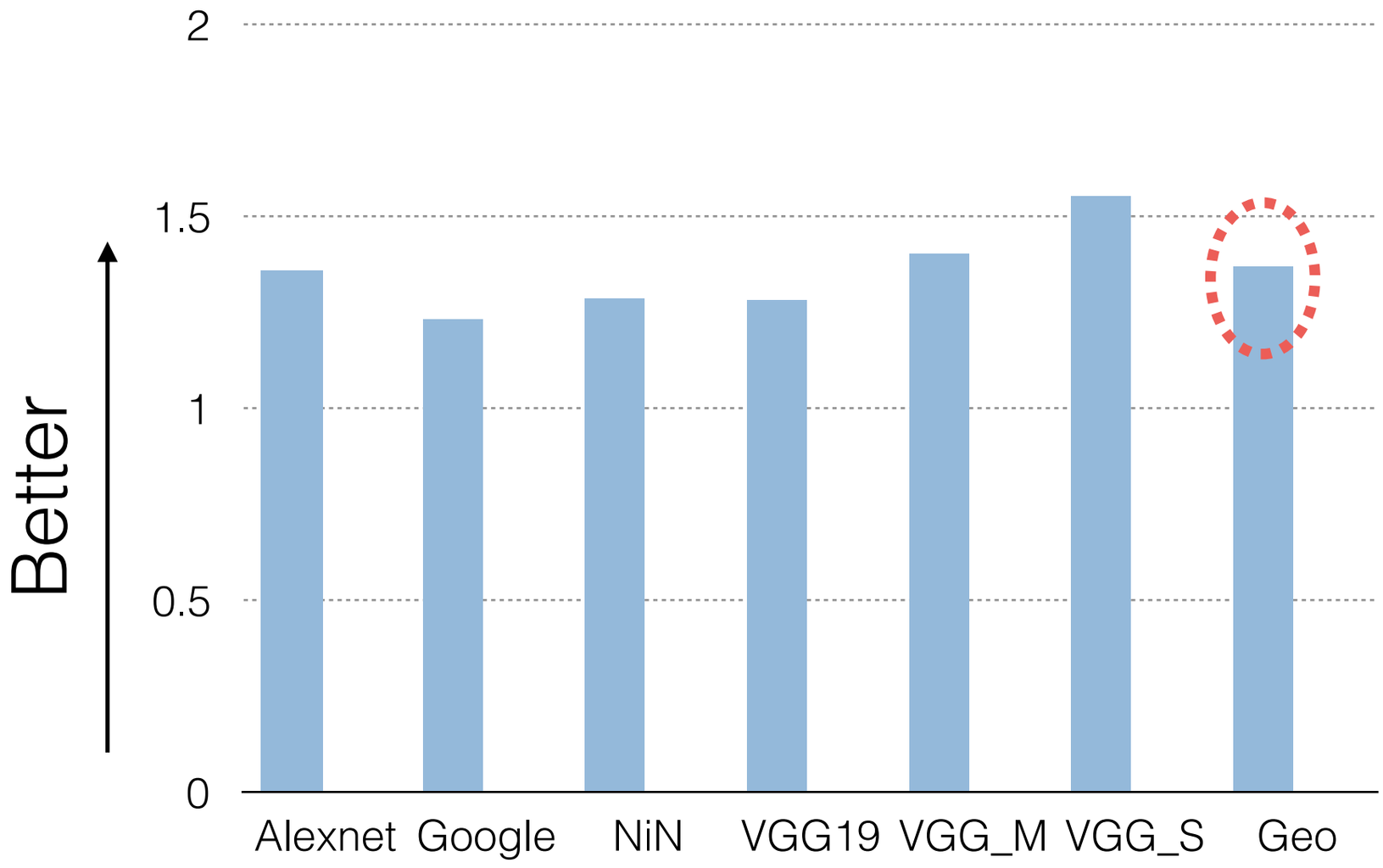
Speedup: ineffectual = 0



Speedup: ineffectual = 0



Speedup: ineffectual = 0



1.37x Performance on average

Loosening the Ineffectual Neuron Criterion

CNVLUTIN **zero**

“If all you have is ~~a hammer~~, everything looks like a ~~nail~~”

(Maslow’s hammer)

37	0	13	10
15	1	123	0
0	7	1	3
0	1	20	0
18	31	0	33

Loosening the Ineffectual Neuron Criterion

CNVLUTIN **zero**

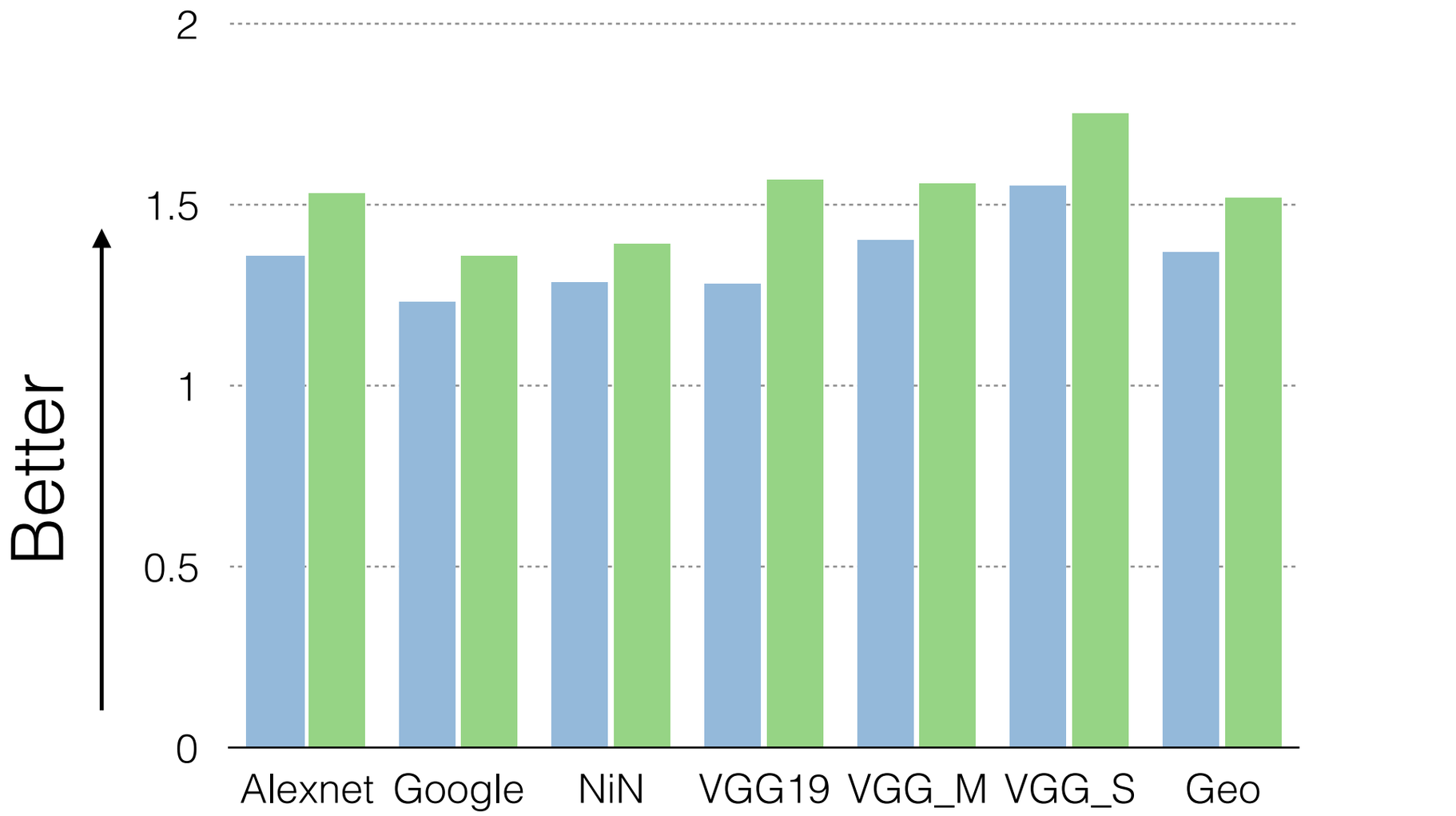
“If all you have is ~~a hammer~~, everything looks like a ~~nail~~”

(Maslow’s hammer)

37	0	13	10
15	1	123	0
0	7	1	3
0	1	20	0
18	31	0	33

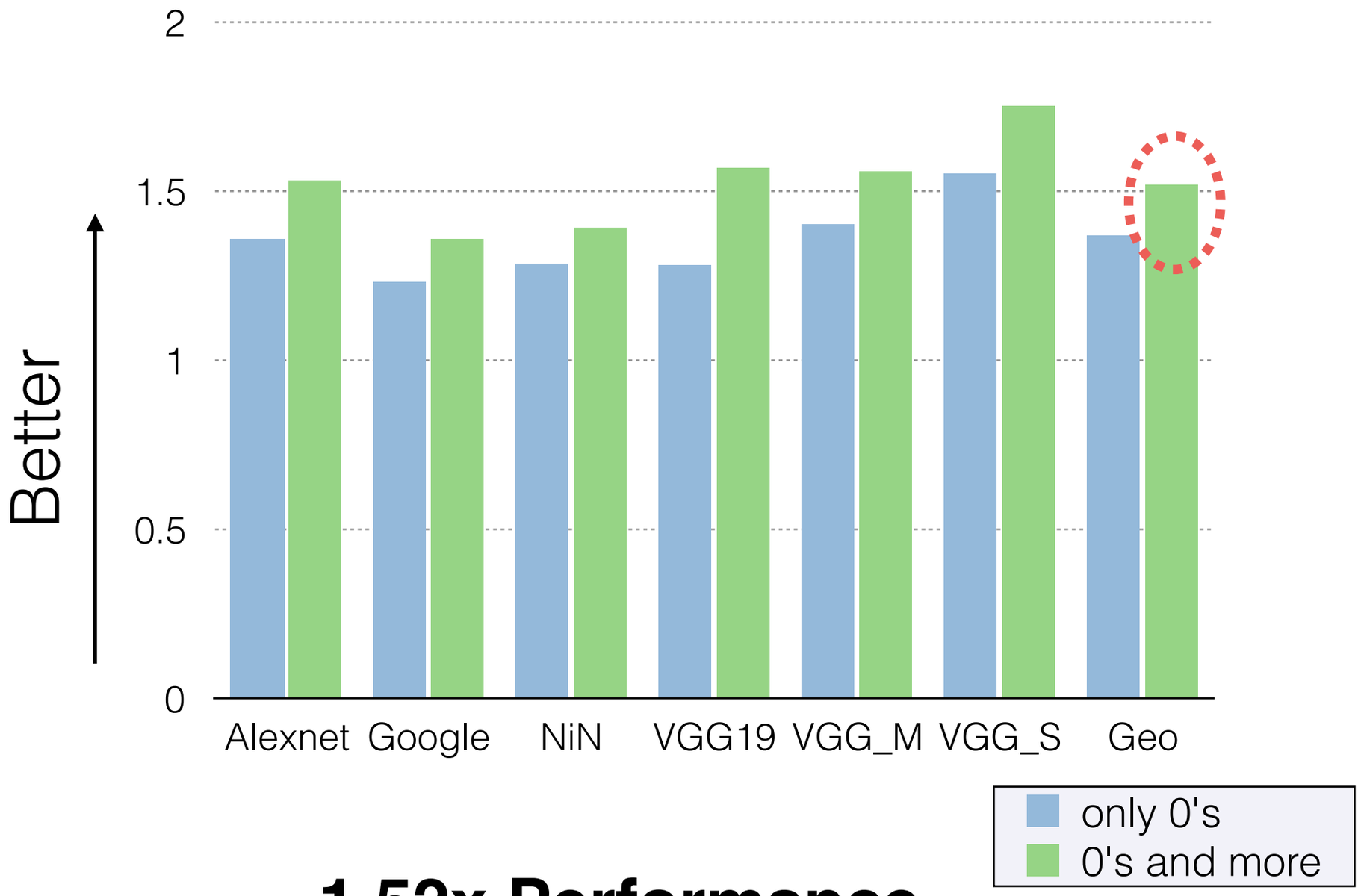
Example: consider ineffectual if value<2

Speedup: ineffectual >= 0



1.52x Performance
No accuracy lost

Speedup: ineffectual >= 0



1.52x Performance
No accuracy lost

Loosening the Ineffectual Neuron Criterion

CNVLUTIN **zero**

“If all you have is ~~a hammer~~, everything looks like a ~~nail~~”

(Maslow’s hammer)

37	0	13	10
15	1	123	0
0	7	1	3
0	1	20	0
18	31	0	33

Example: consider ineffectual if value<2

Loosening the Ineffectual Neuron Criterion

CNVLUTIN **zero**

“If all you have is ~~a hammer~~, everything looks like a ~~nail~~”

(Maslow’s hammer)

37	0	13	10
15	1	123	0
0	7	1	3
0	1	20	0
18	31	0	33

Example: consider ineffectual if value<8

Loosening the Ineffectual Neuron Criterion

CNVLUTIN **zero**

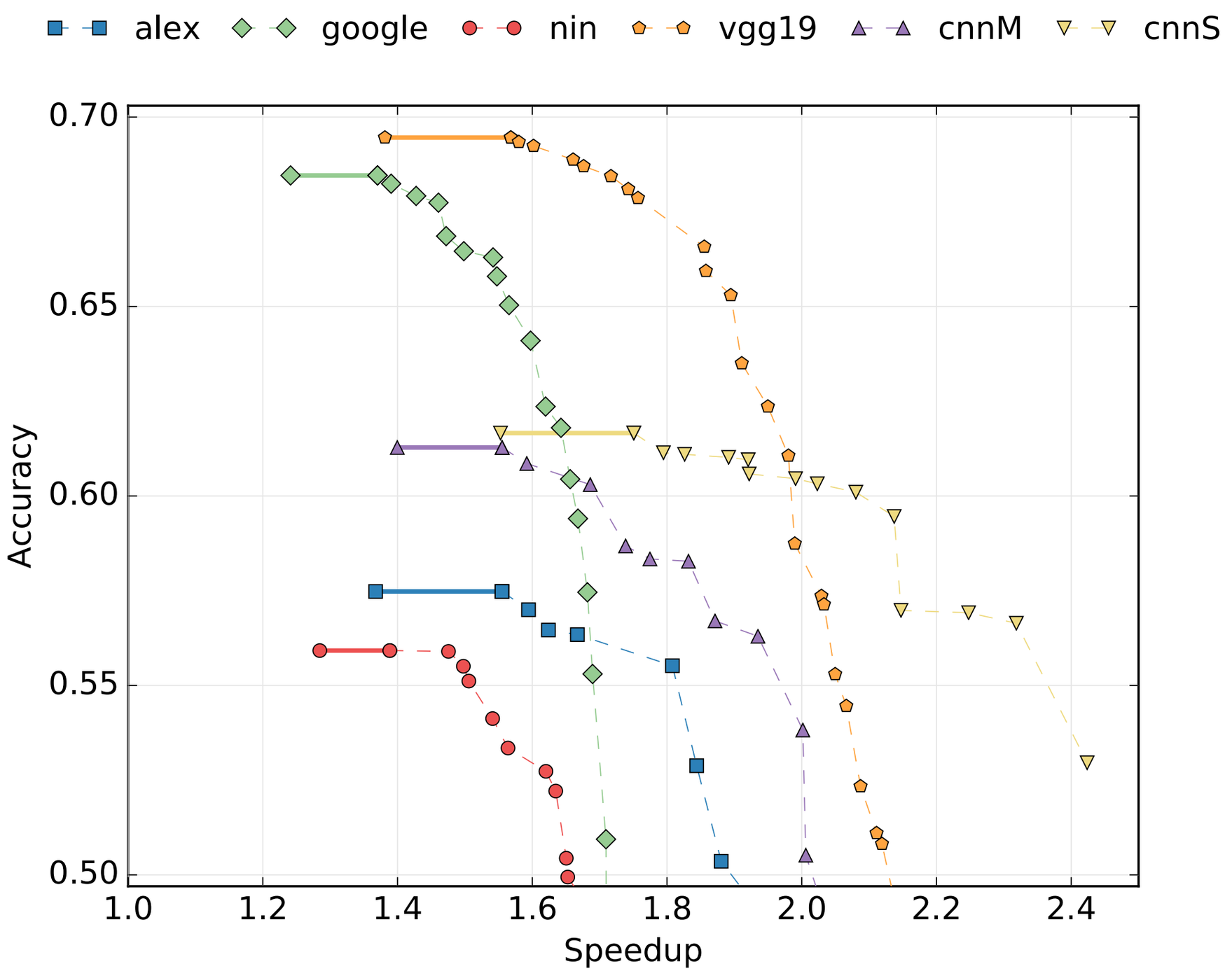
“If all you have is ~~a hammer~~, everything looks like a ~~nail~~”

(Maslow’s hammer)

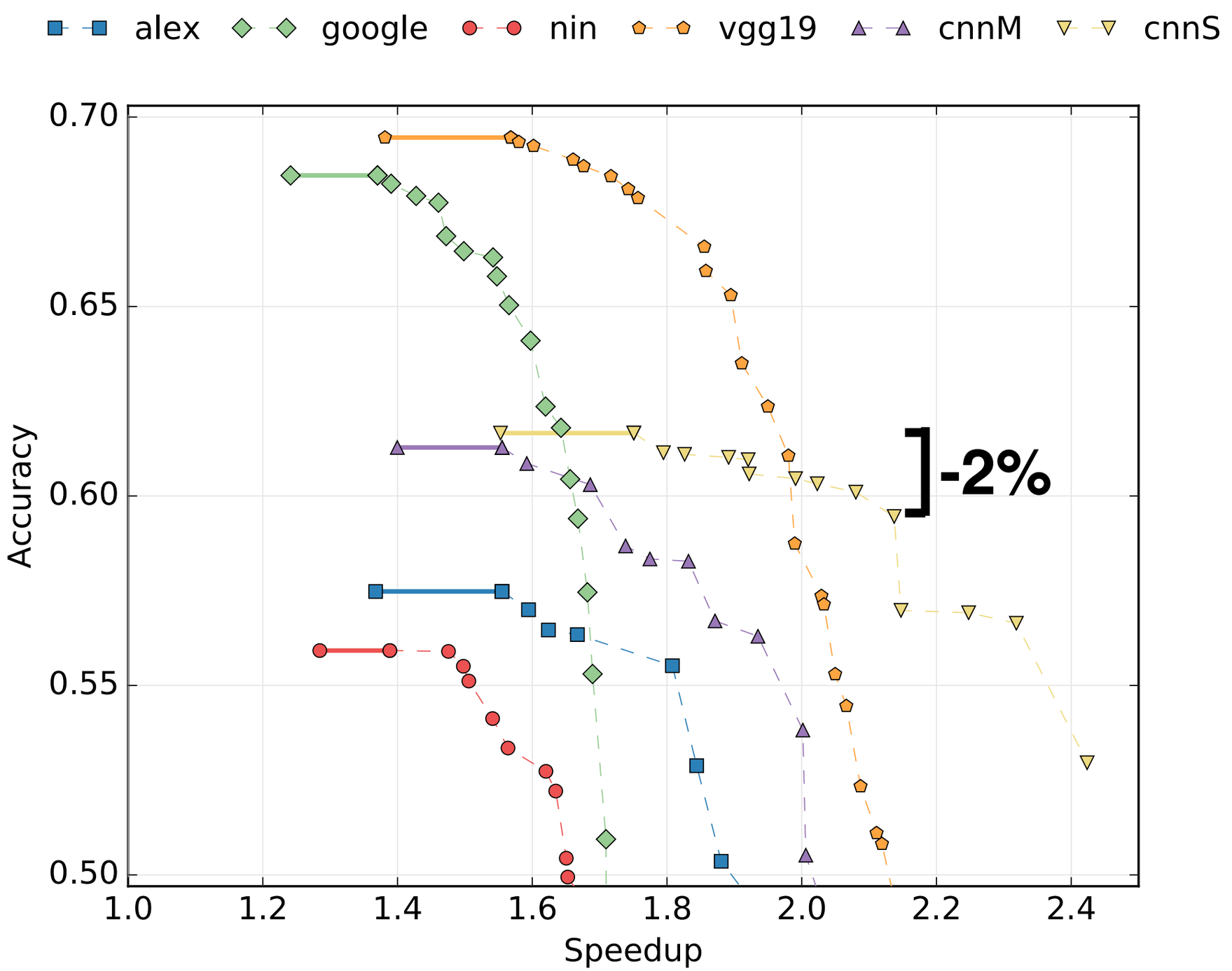
37	0	13	10
15	1	123	0
0	7	1	3
0	1	20	0
18	31	0	33

Example: consider ineffectual if value<32

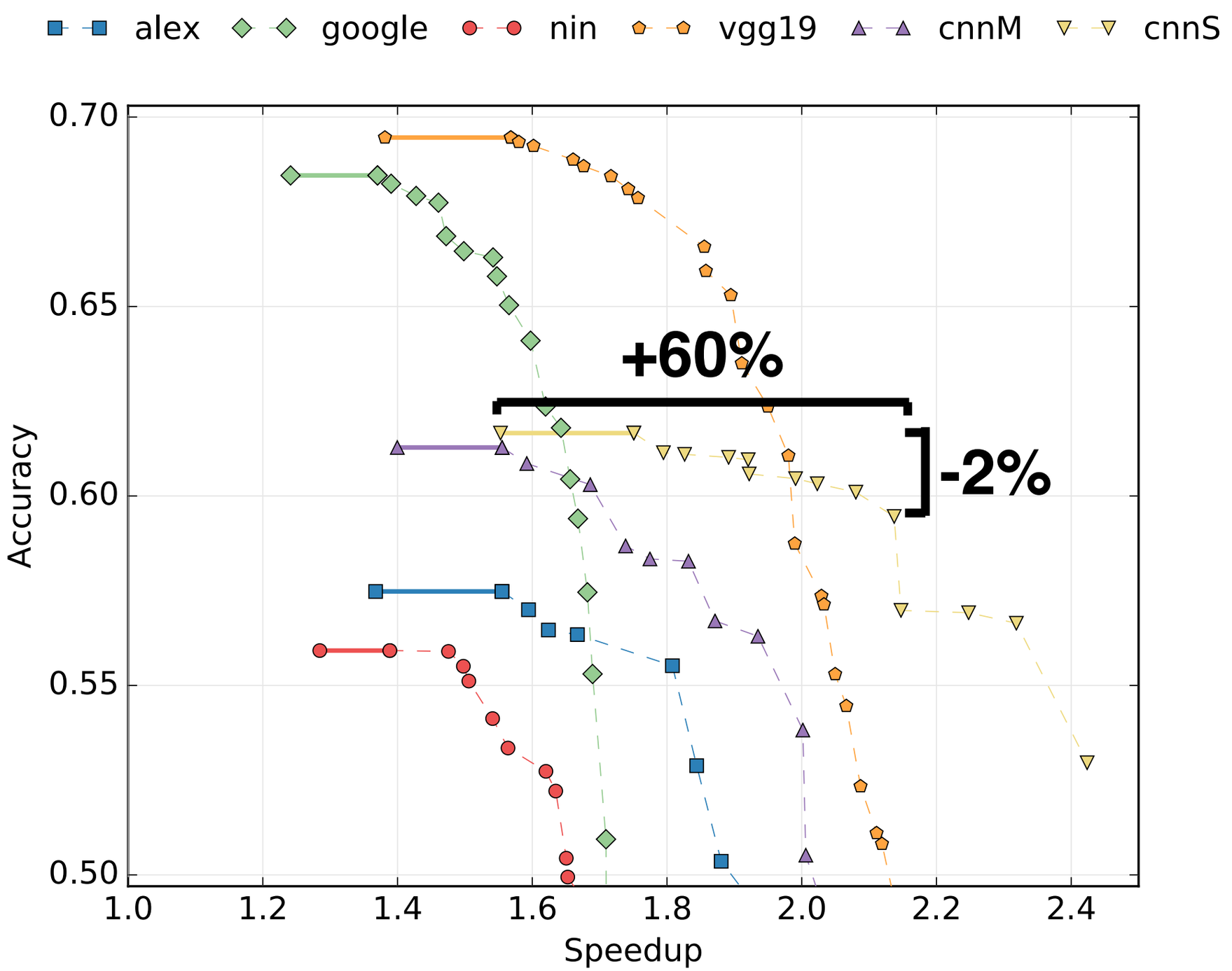
Trading accuracy for performance



Trading accuracy for performance



Trading accuracy for performance



CNVLUTIN: Smarter SIMD

52% Performance — 2x ED²P
No accuracy lost

out-of-the-box networks

Our Approach

Value-Aware Deep Learning Acceleration

arXiv, a while ago:

Reduced-Precision Strategies for Bounded Memory in DNNs

ICS 2016

Proteus: Exploiting Numerical Precision Variability in DNNs

Today

CNVLUTIN: Ineffectual-Neuron-Free DNN Computing

CAL (to appear)

How to offer performance that scales linearly with required numerical precision

More things coming soon :-)

Additional Material

CNVLUTIN: Ineffectual-neuron-free DNN computing

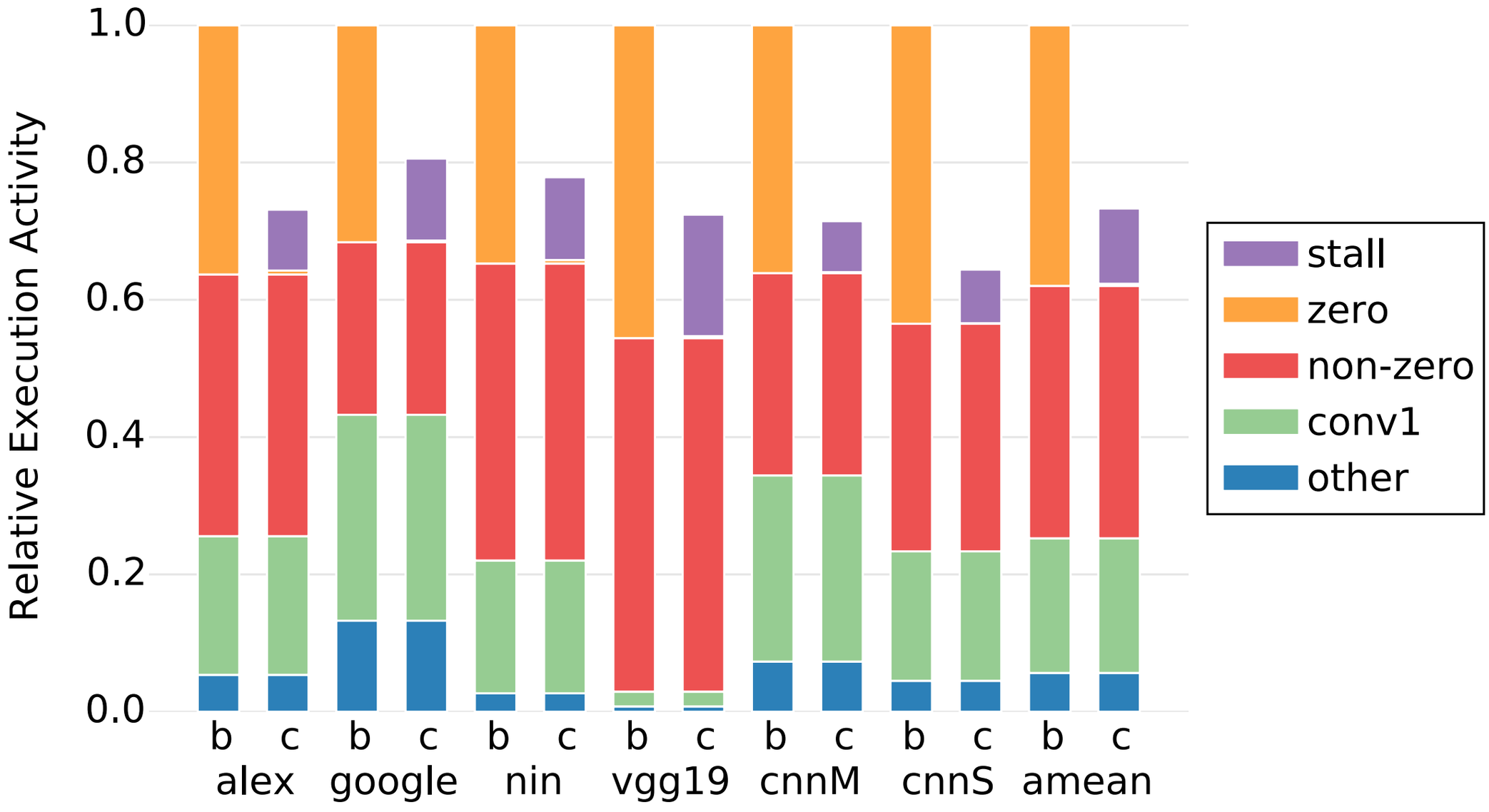
J. Albericio, P. Judd, T. Hetherington*,
T. Aamodt*, N. E. Jerger, A. Moshovos

UNIVERSITY OF TORONTO

UBC*

Please cite the original source.

Execution Activity Breakdown

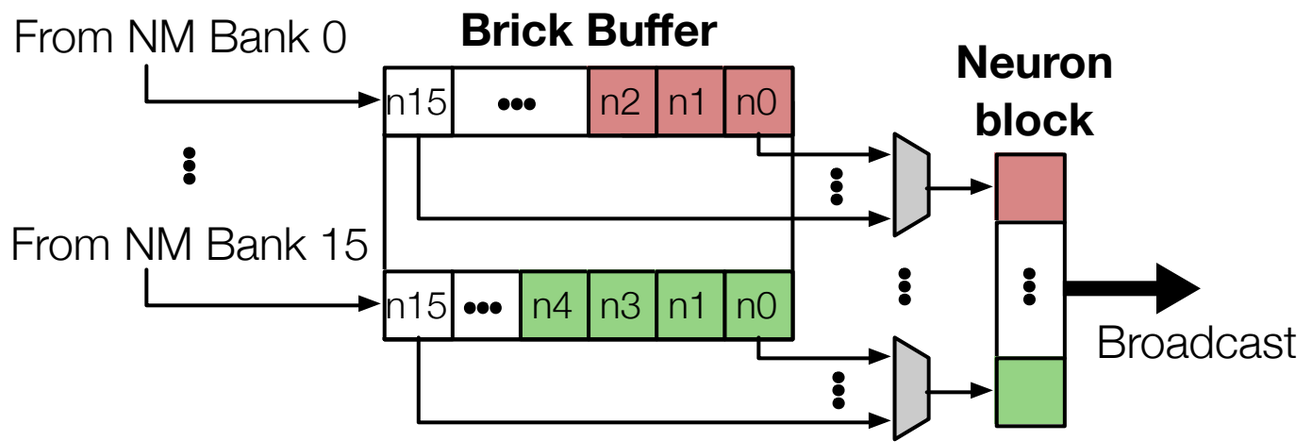


Cnvlutin: Maintaining Wide NM accesses

Dispatcher:

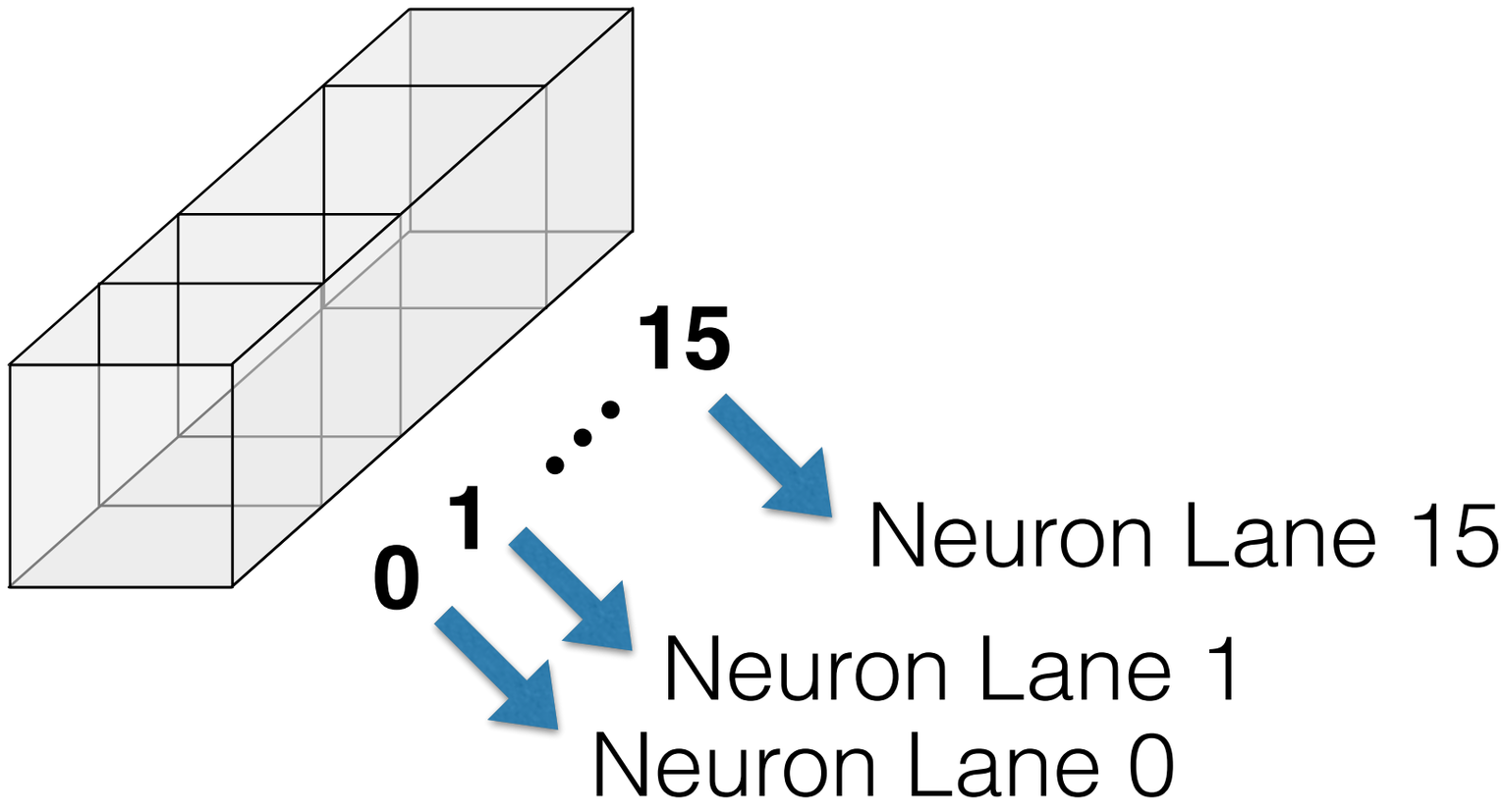
Reads Neuron Bricks - up to 16 neuron

Maintain one per Neuron Lane



Cnvlutin: Maintaining Wide NM accesses

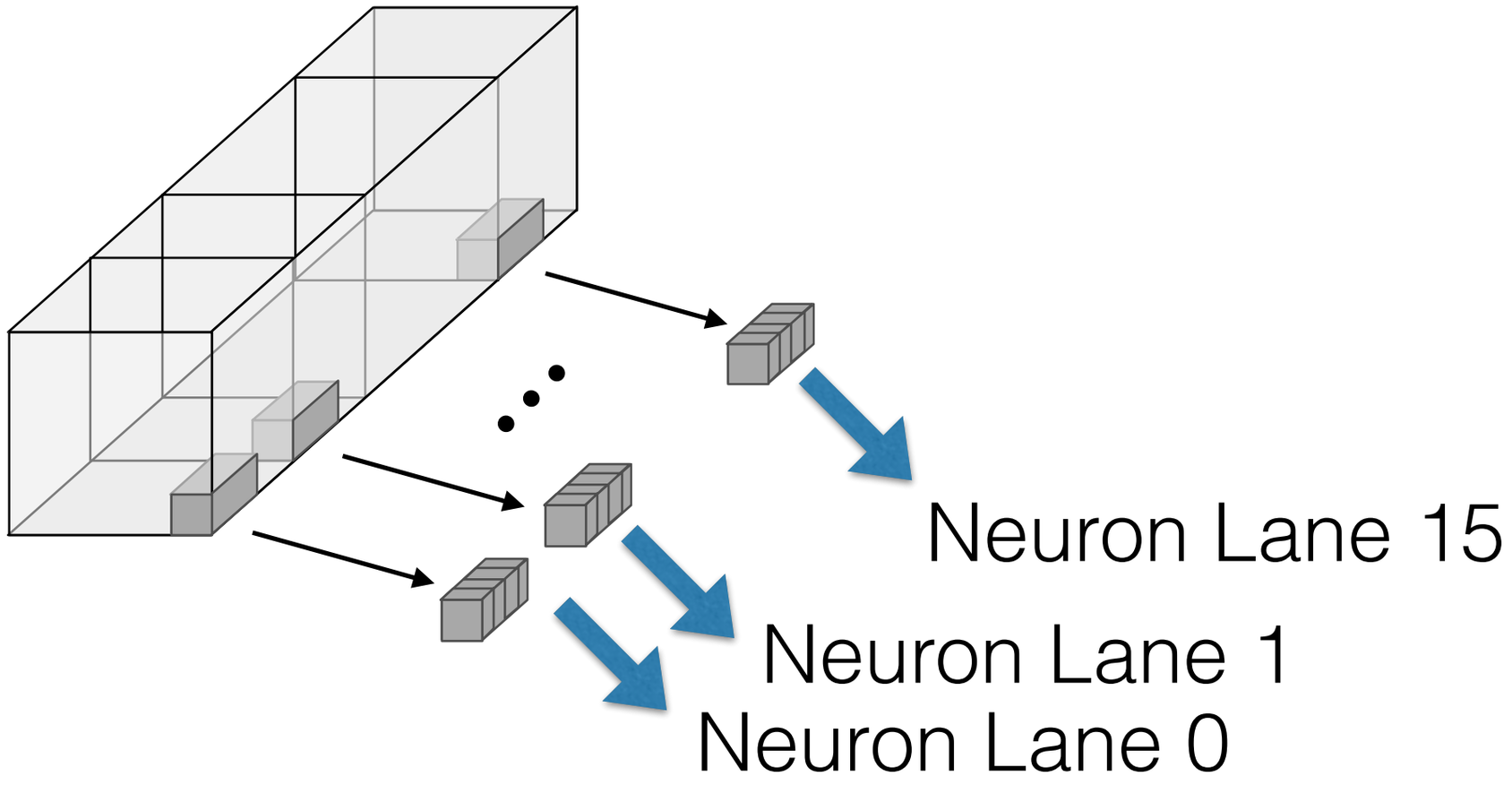
#1: Partition NM in 16 Slices over 16 banks



Cnvlutin: Maintaining Wide NM accesses

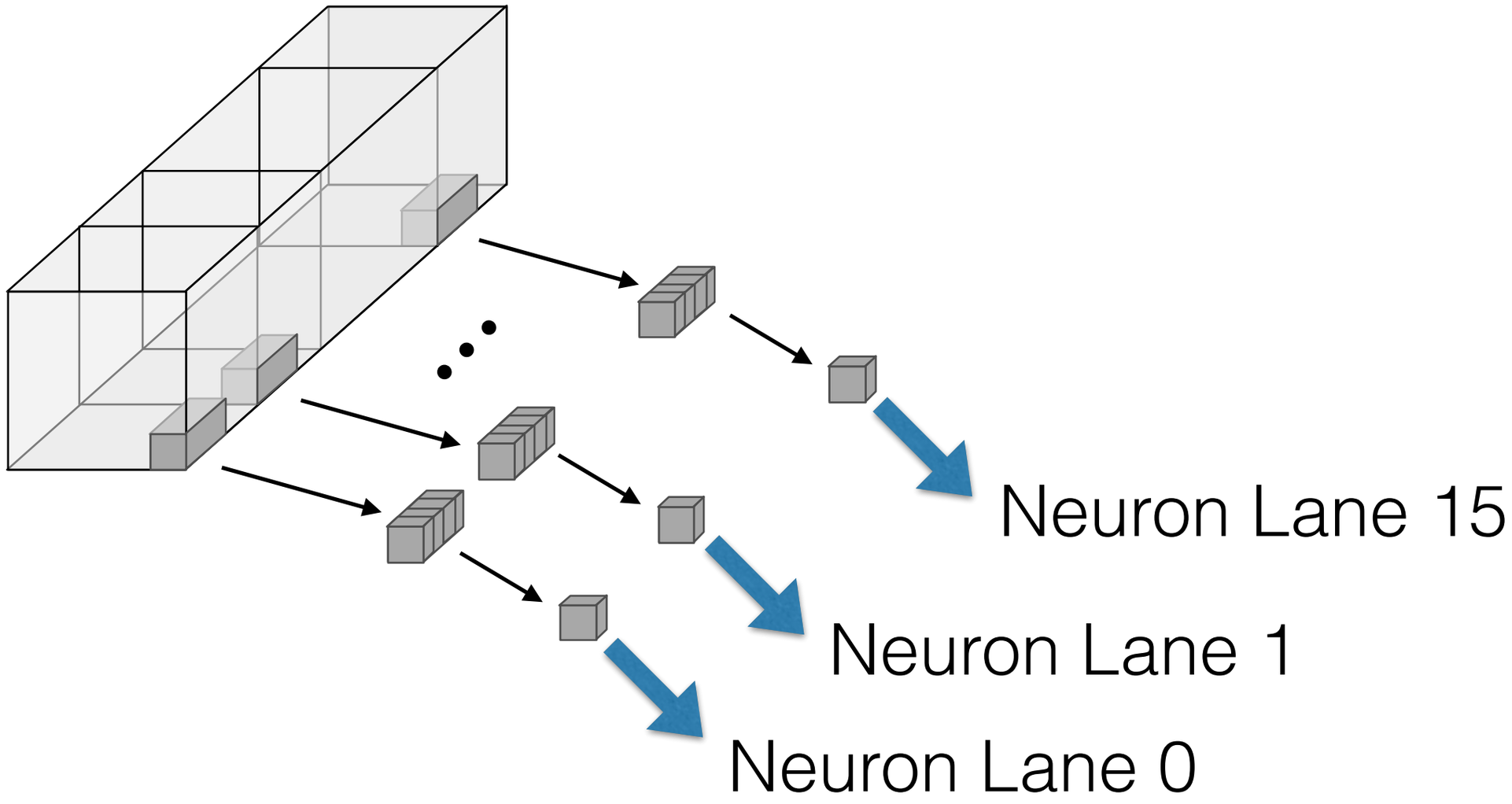
#2: Fetch and Maintain One Container per Slice

Container: up to 16 non-zero neurons

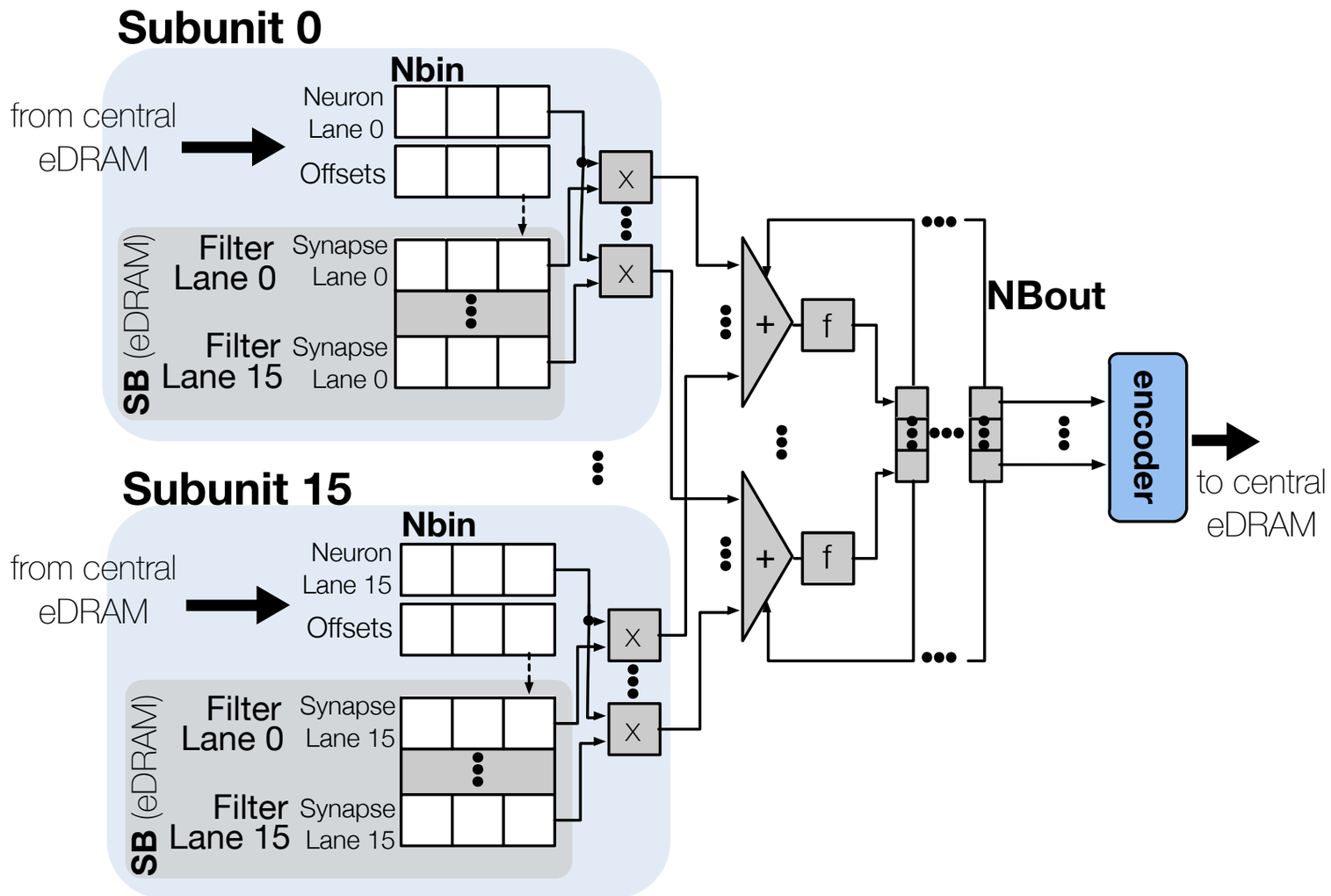


Cnvlutin: Maintaining Wide NM accesses

#3: Keep neuron lanes supplied with one neuron per cycle



Cnvlutin: Summary



Decoupled Neuron Lanes:

Neuron + coordinate

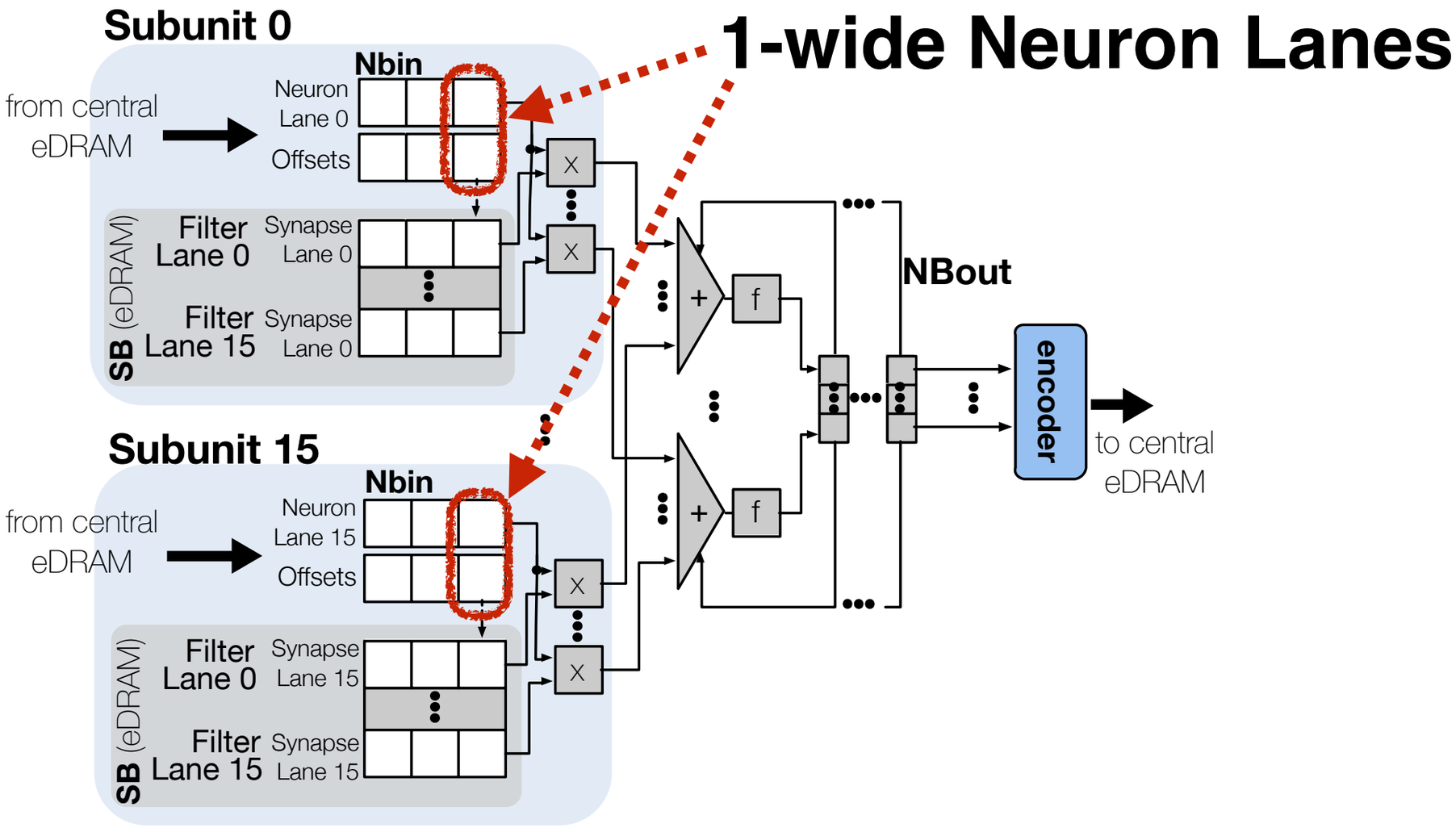
Proceed independently

Partitioned SB:

16-wide accesses

1 synapse per filter

Cnvlutin: Summary



Decoupled Neuron Lanes:

Neuron + coordinate

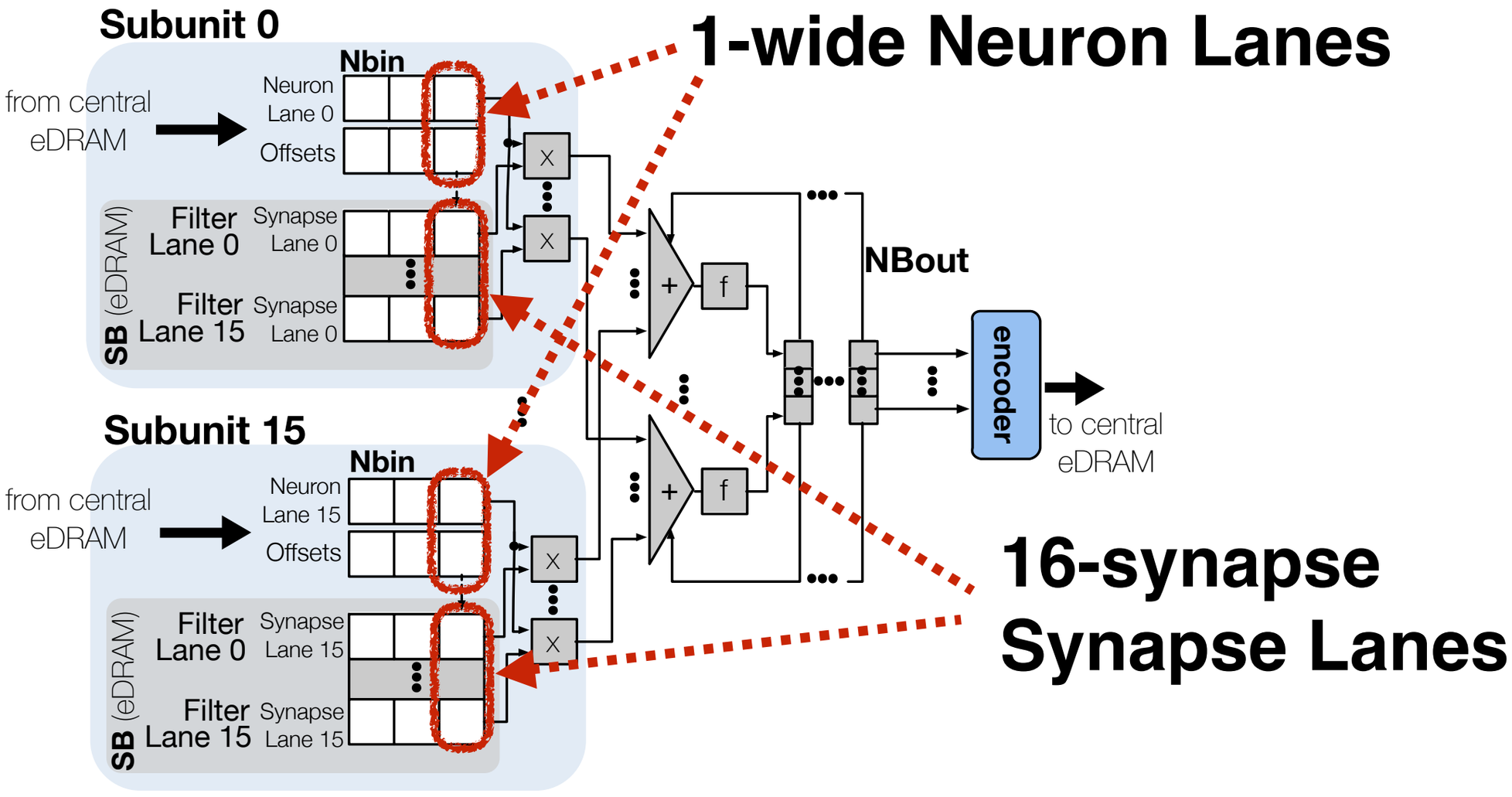
Proceed independently

Partitioned SB:

16-wide accesses

1 synapse per filter

Cnvlutin: Summary



Decoupled Neuron Lanes:

Neuron + coordinate

Proceed independently

Partitioned SB:

16-wide accesses

1 synapse per filter