Neural Networks need Power

- Number of operations and error rate:
  - AlexNet (240 MB): 720 MFLOPs
  - GoogleNet: 1550
  - VGG-19 (550MB): MFLOPs
  - ResNet-152 (240MB): 19.6

BFLO

11.3

BFLO

Neural Networks need Power

- Number of operations and error rate:
  - AlexNet (240 MB): 720 MFLOPs
  - GoogleNet: 1550
  - VGG-19 (550MB): MFLOPs
  - ResNet-152 (240MB): 19.6

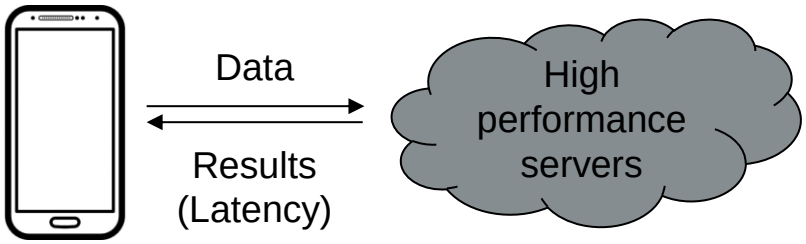
BFLO

11.3

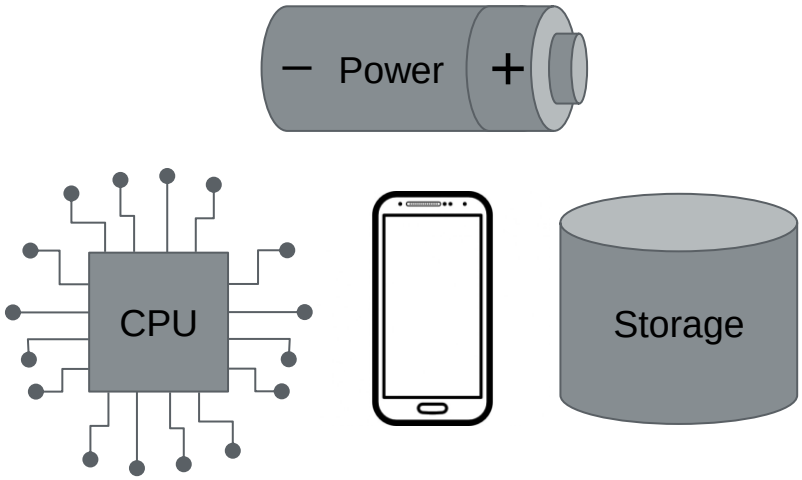
BFLO

- Inference time on CPU:
  - AlexNet: 3 fps
  - VGG-10: 0.25 fps

Binary Neural Networks

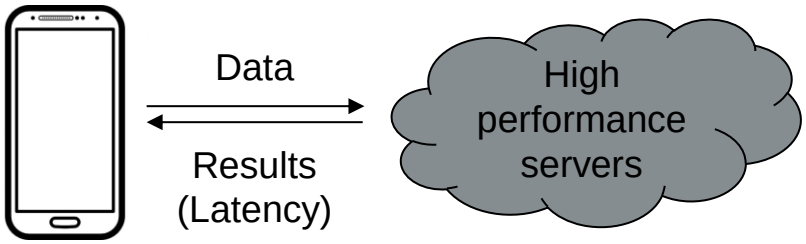


Processing in the cloud



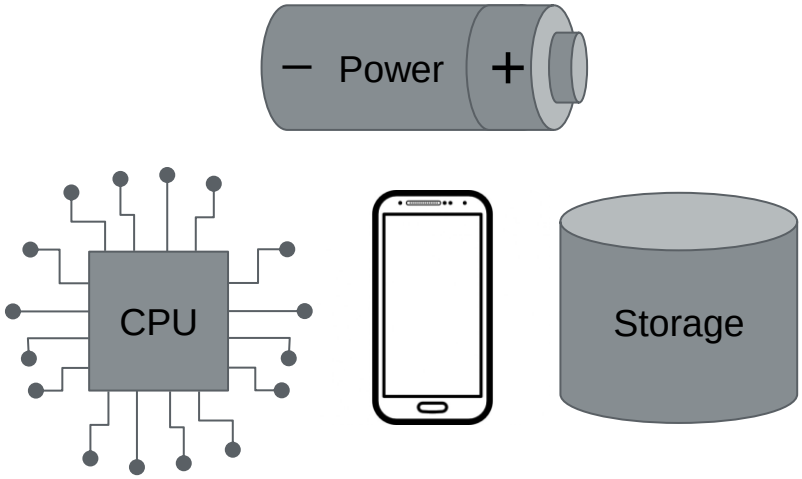
Processing on device

Binary Neural Networks



Processing in the cloud

Limitations: availability, latency, privacy



Processing on device

Limitations: computational power, memory, energy

How does it work?

- Neural Networks learn weights (numbers)
  - Usually floating point numbers (full-precision)

Full-precision

Input		Weight		
0.1		0.5	-0.5	-0.1
-0.7		-0.1	0.5	0.5
0.5	*	-0.4	-0.7	0.3
0.3		0.3	-0.1	-0.7
Result		0.01	-0.78	-0.42

How does it work?

- Neural Networks learn weights (numbers)
  - Usually floating point numbers (full-precision)

Full-precision

Input		Weight		
0.1	*	0.5	-0.5	-0.1
-0.7		-0.1	0.5	0.5
0.5		-0.4	-0.7	0.3
0.3		0.3	-0.1	-0.7
Result		0.01	-0.78	-0.42

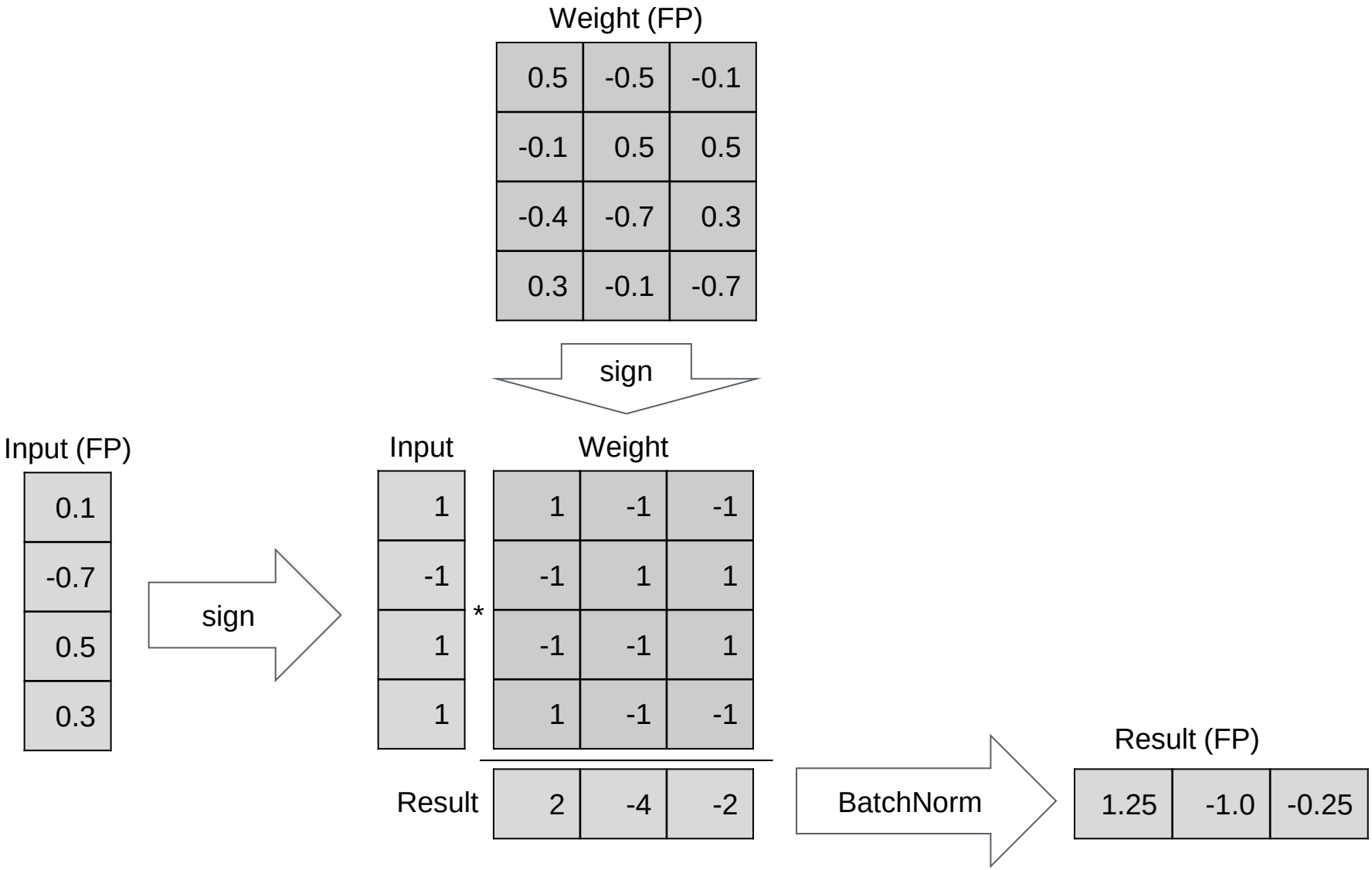
Binary

Input		Weight		
1	*	1	-1	-1
-1		-1	1	1
1		-1	-1	1
1		1	-1	-1
Result		2	-4	-2

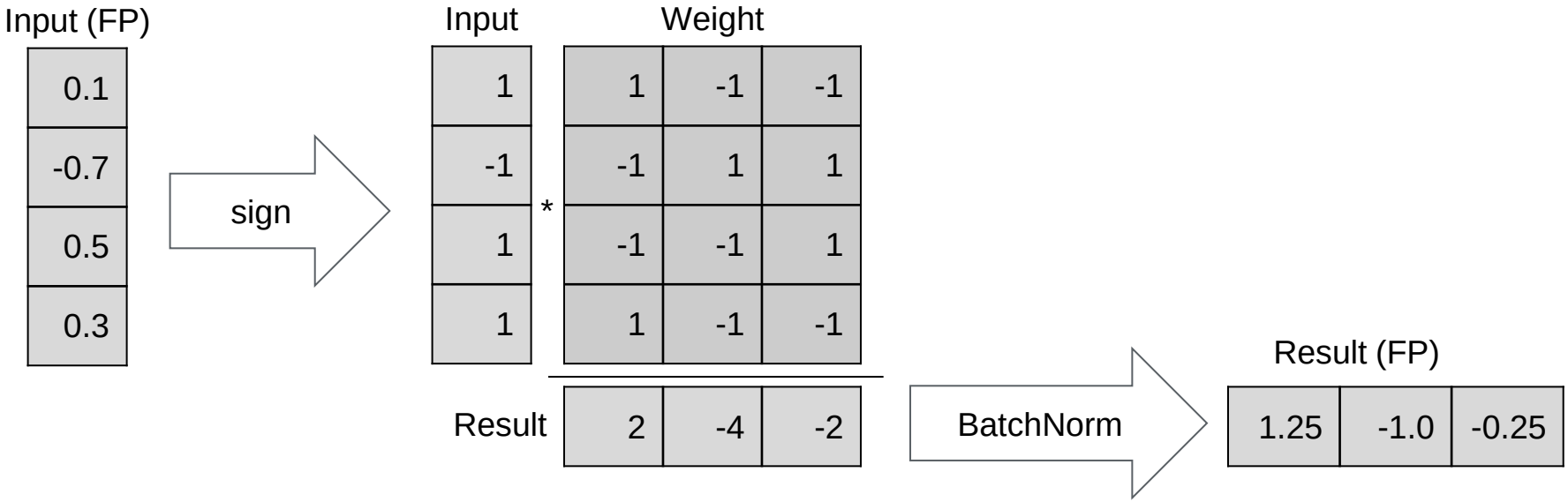
- In a binary neural network we use binary values instead

  → 32x saving of memory, theoretical speedup

Training


Weight (FP)
0.5 -0.5 -0.1
-0.1 0.5 0.5
-0.4 -0.7 0.3
0.3 -0.1 -0.7
sign
Input (FP)
0.1
-0.7
0.5
0.3
sign
Input
Weight
1
-1
1
1
*
1 -1 -1
-1 1 1
-1 -1 1
1 -1 -1
Result
2 -4 -2
BatchNorm
Result (FP)
1.25 -1.0 -0.25

Inference



- Speedup with XNOR and popcount (+ scaling)

x	y	x*y
-1	-1	1
-1	1	-1
1	-1	-1
1	1	1

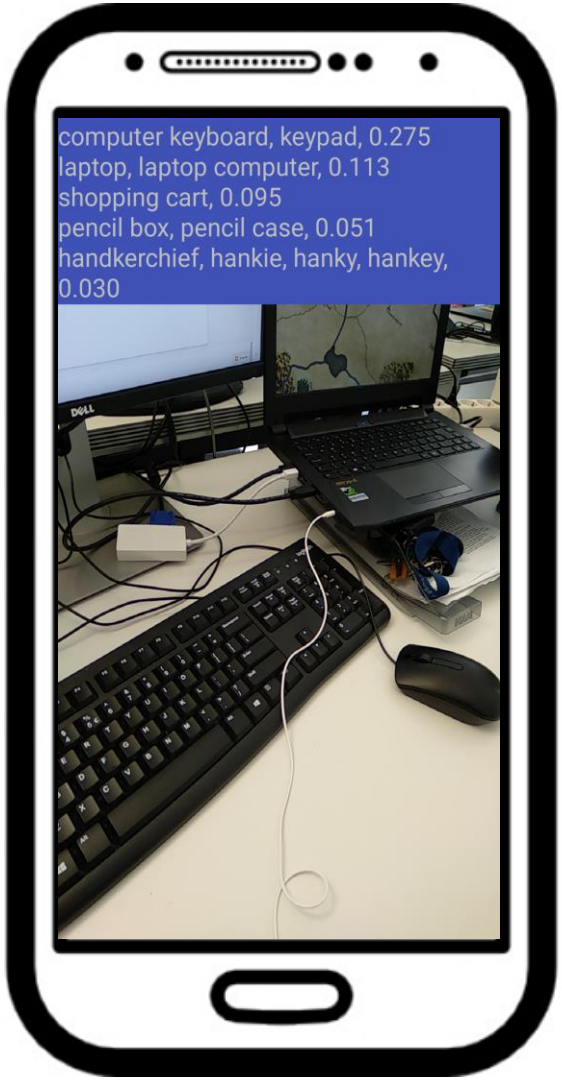
x	y	xnor(x,y)
0	0	1
0	1	0
1	0	0
1	1	1

Open Source Implementation BMXNet 2

- Source code is on github:

  https://github.com/hpi-xnor/BMXNet-v2

- Demo of first BMXNet version:
  - Binary ResNet18 (1.5 MB)



Binary Neural Networks

- Based on BMXNet 2
- Develop a deep learning application

  (low latency, data privacy and/or network independency)
- Specific application is open for discussion, we have a few ideas prepared
- Deploy on Raspberry Pi