AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY

Boosting FPGA efficiency with modified representation of data

Michał Karwatowski, Maciej Wielgosz, Marcin Pietroń, Kazimierz Wiatr

Zakopane 08.03.2018

Agenda

» The need for precision reduction

» Test database – MNIST

» K nearest neighbors

» Distance calculation

» FPGA implementation

» Experiments results

Why reduce precision?

» High cost of floating point operations

» AI algorithms respond well

» We don’t always need high accuracy

MNIST

» Training – 60000

» Test – 10000

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, 86(11):2278-2324, November 1998.

K nearest neighbors

» Calculate distance

$$d_{L2\,(x,y)} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

» Sort results

» Choose first K

» Vote

MNIST:

97.17% accuracy

99.37% accuracy

(with preprocessing)

```
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  51 159 253 159  50   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0  48 238 252 252 252 237   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0  54 227 253 252 239 233 252  57   6   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0  10  60 224 252 253 252 202  84 252 253 122   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0 163 252 252 252 253 252 252  96 189 253 167   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0  51 238 253 253 190 114 253 228  47  79 255 168   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0  48 238 252 252 179  12  75 121  21   0   0 253 243  50   0   0   0   0   0
0   0   0   0   0   0   0   0  38 165 253 233 208  84   0   0   0   0   0   0 253 252 165   0   0   0   0   0
0   0   0   0   0   0   0   7 178 252 240  71  19  28   0   0   0   0   0   0 253 252 195   0   0   0   0   0
0   0   0   0   0   0   0  57 252 252  63   0   0   0   0   0   0   0   0   0 253 252 195   0   0   0   0   0
0   0   0   0   0   0   0 198 253 190   0   0   0   0   0   0   0   0   0   0 255 253 196   0   0   0   0   0
0   0   0   0   0   0  76 246 252 112   0   0   0   0   0   0   0   0   0   0 253 252 148   0   0   0   0   0
0   0   0   0   0   0  85 252 230  25   0   0   0   0   0   0   0   0   7 135 253 186  12   0   0   0   0   0
0   0   0   0   0   0  85 252 223   0   0   0   0   0   0   0   0   7 131 252 225  71   0   0   0   0   0   0
0   0   0   0   0   0  85 252 145   0   0   0   0   0   0   0  48 165 252 173   0   0   0   0   0   0   0   0
0   0   0   0   0   0  86 253 225   0   0   0   0   0   0 114 238 253 162   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0  85 252 249 146  48  29  85 178 225 253 223 167  56   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0  85 252 252 252 229 215 252 252 252 196 130   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0  28 199 252 252 253 252 252 233 145   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0  25 128 252 253 252 141  37   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

FPGA implementation

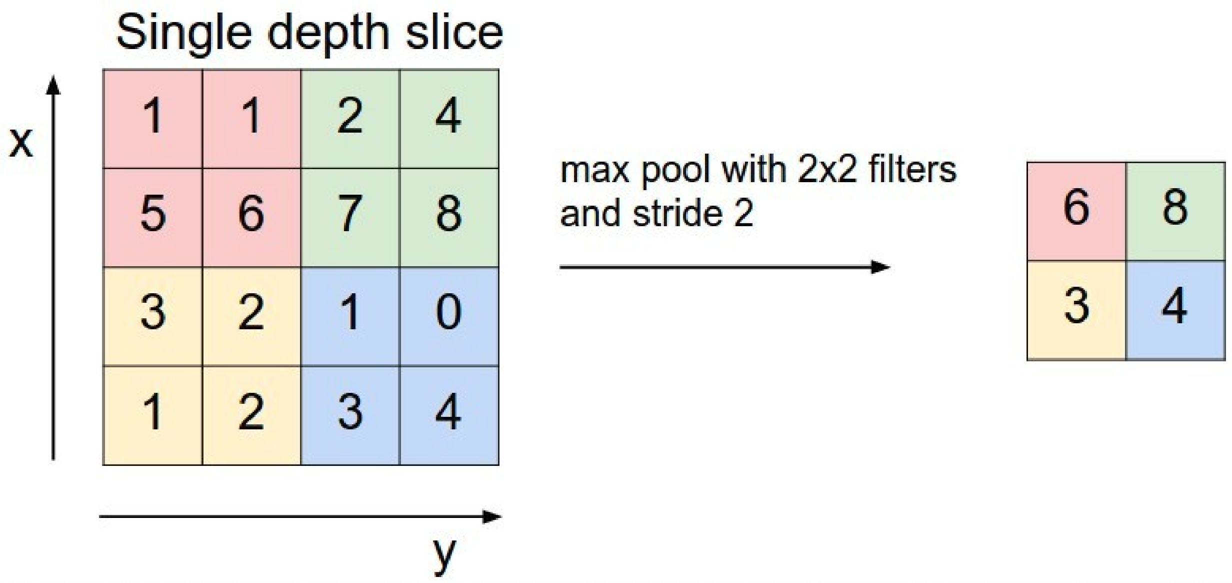
» Delegate calculations to hardware

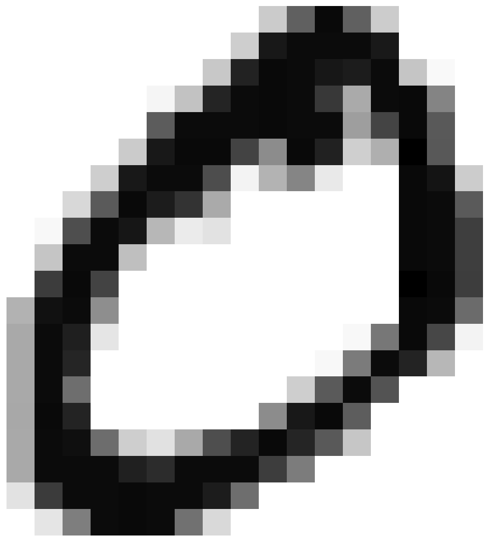
$$d_{L2(x,y)} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

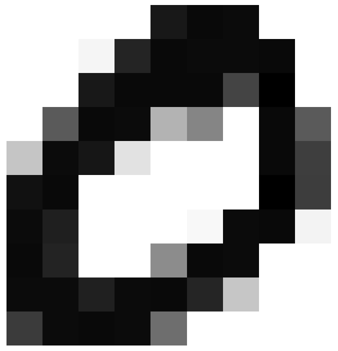
```
void dist_L2(hlsml::Tensor3D &tensorA, hlsml::Tensor3D &tensorB, dist_t &distance) {
        sum_t sum = 0;
        for (unsigned d = 0; d < tensorA.depth(); d++) {
                for (unsigned h = 0; h < tensorA.height(); h++) {
                        for (unsigned w = 0; w < tensorA.width(); w++) {
                                diff_t diff = diff_t(tensorA(d, h, w)) - diff_t(tensorB(d, h, w));
                                sum += sum_t(pow_t(diff) * pow_t(diff));
                        }
                }
        }
        distance = dist_t(hls::sqrt(sum));
}
```

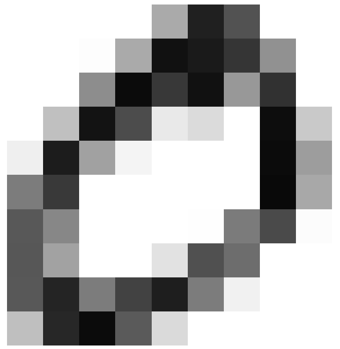
Resolution reduction

- » Max pooling
- » Average pooling









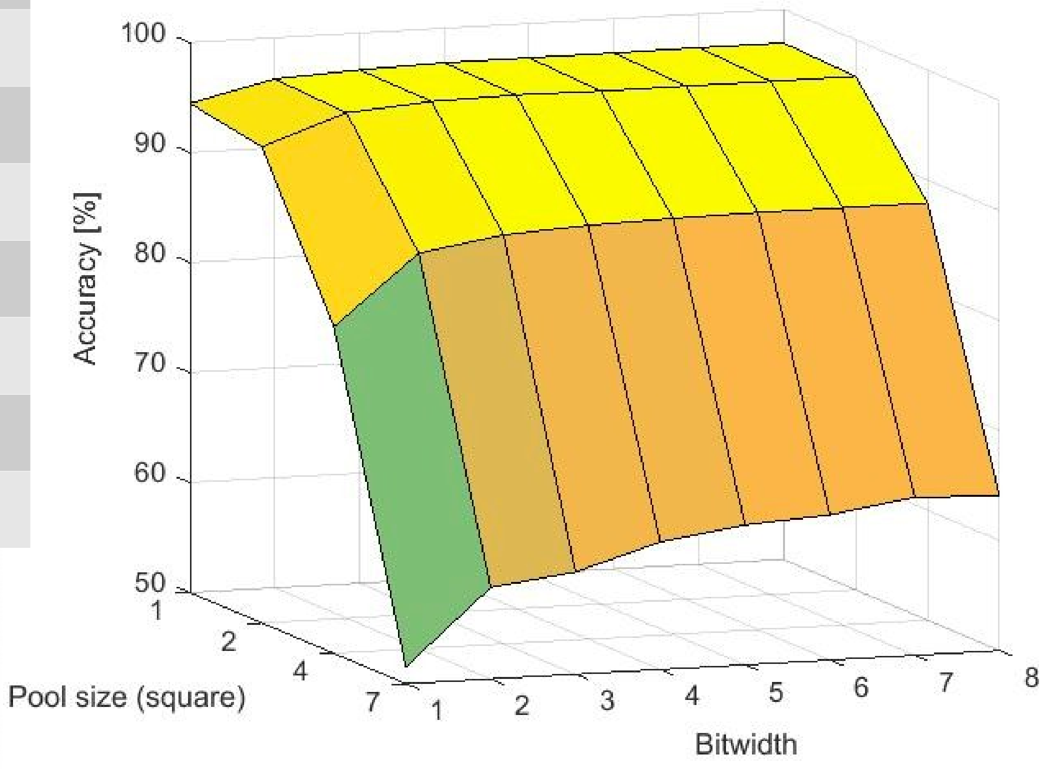
Distance precision

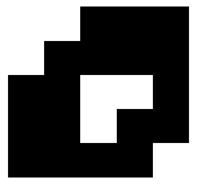
			28x28		14x14		7x7		4x4	
mnist	diff	pow	sum	dist	sum	dist	sum	dist	sum	dist
8	9	17	26	13	24	12	22	11	21	11
7	8	15	24	12	22	11	20	10	19	10
6	7	13	22	11	20	10	18	9	17	9
5	6	11	20	10	18	9	16	8	15	8
4	5	9	18	9	16	8	14	7	13	7
3	4	7	16	8	14	7	12	6	11	6
2	3	5	14	7	12	6	10	5	9	5
1	1	1	10	5	8	4	6	3	5	3

$$d_{L2(x,y)} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Bitwidth + max pooling

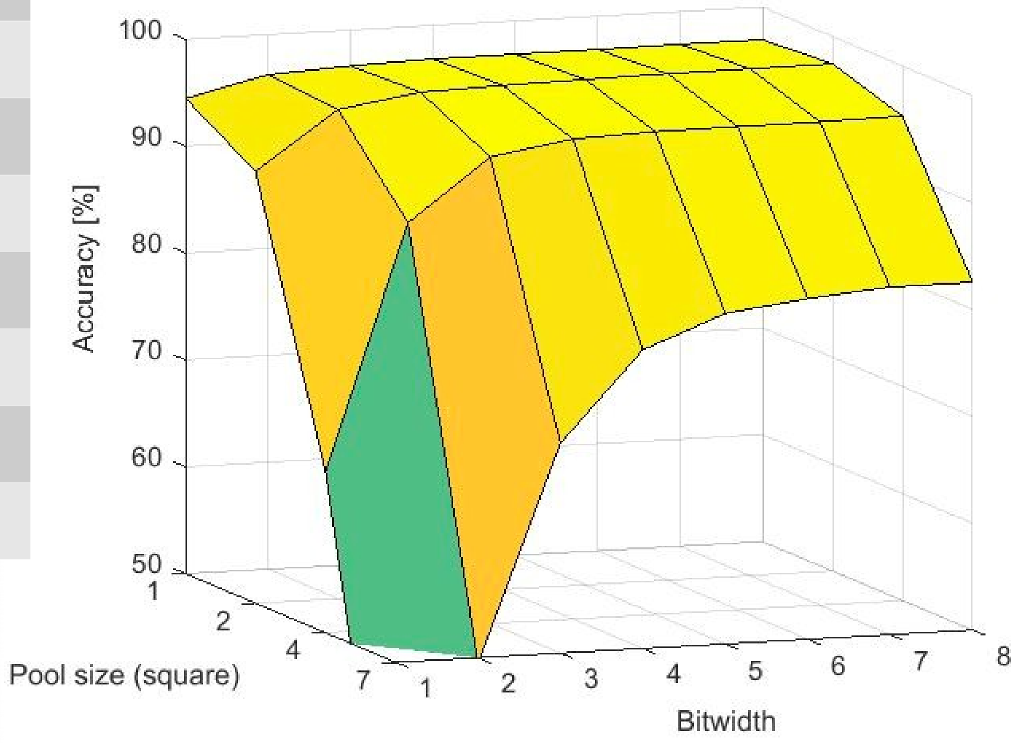
bitwidth	28x28	14x14	7x7	4x4
1	94.47	93.29	79.68	51.53
2	96.25	95.94	85.94	58.33
3	96.65	96.54	87.17	59.3
4	96.87	96.7	87.61	61.59
5	96.88	96.62	87.81	62.71
6	96.89	96.65	87.92	63.2
7	96.95	96.65	87.91	64.38
8	96.94	96.66	87.93	64.04

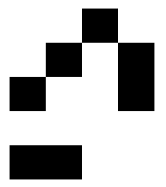




Bitwidth + average pooling

bitwidth	28x28	14x14	7x7	4x4
1	94.47	90.39	65.06	23.15
2	96.25	95.76	87.92	49.51
3	96.65	96.95	93.65	69.62
4	96.87	97.15	94.91	77.93
5	96.88	97.28	95.16	80.92
6	96.89	97.39	95.2	81.86
7	96.95	97.41	95.22	82.53
8	96.94	97.45	95.32	82.55

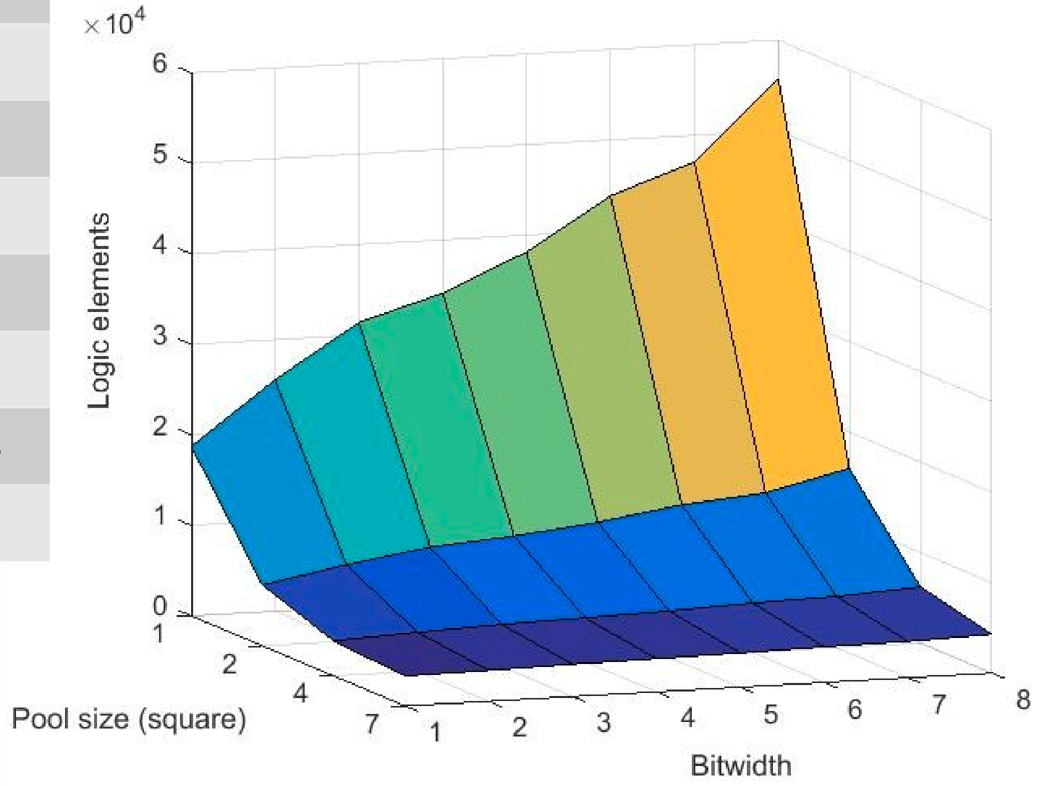




AGH

Logic utilization

bitwidth	28x28	14x14	7x7	4x4
1	18723	6732	3879	3301
2	25570	8429	4299	3423
3	31381	9883	4663	3582
4	34092	10578	4858	3677
5	38139	11589	5106	3788
6	43818	13008	5460	3947
7	47049	13815	5661	4034
8	55720	15982	6202	4281



AGH

Results

» Baseline:

96.94% accuracy, 55720 logic elements

» Reduction:

Accuracy						
97%	96.5%	96%	95.5%	95%	94.5%	94%
aver2-4bit	aver2-3bit	aver2-3bit	aver2-2bit	aver4-5bit	aver4-4bit	aver4-4bit
10578	9883	9883	8429	5106	4858	4858

AGH

Questions

?