Tensorflow Assignment

Street-View House Numbers

Giannis Konstantinidis (BAPT1503)

Stamatis Pitsios (BAPT1502)

Lefteris Vazaios (BAPT1516)

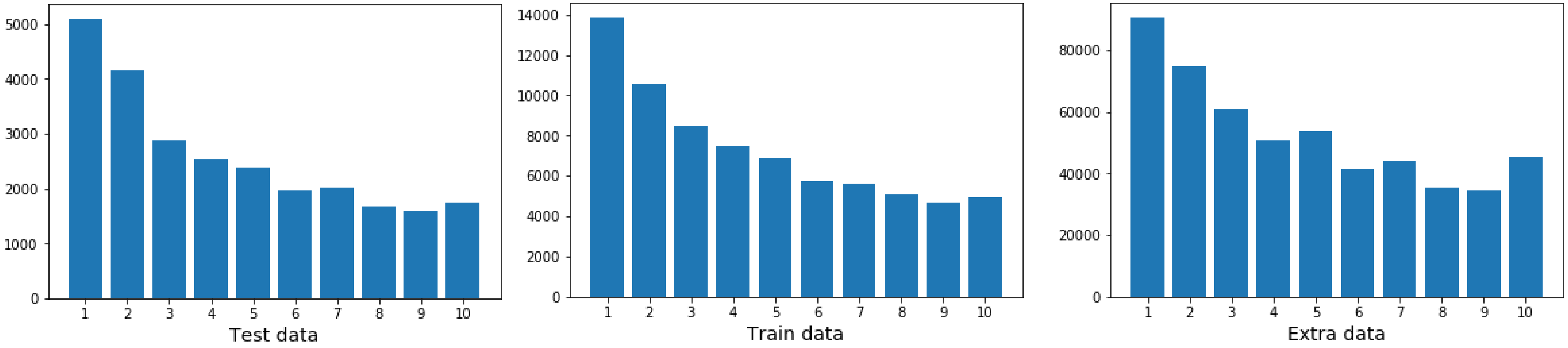
The data

32x32 images of house numbers

Pre-cropped to contain a single number

73k for training, 26k for testing, 531k easier cases

Counts per number and per dataset



Pre-processing

Normalization

- $(x/128) - 1$
- $(x - \mu) / \sigma$
- Batch normalization

Gray scaling

Hardware

1 PC with native Ubuntu 16.4

CPU Core i7 4720hq @3.6GHz (4 Cores/8Threads)

RAM 32GB ddr3

GPU NVidia 970M 3GB V-RAM

2 PCs with VM (1 used for testing)

CPU Core i5 4570 @3.6GHz (4 Cores/4 Threads)

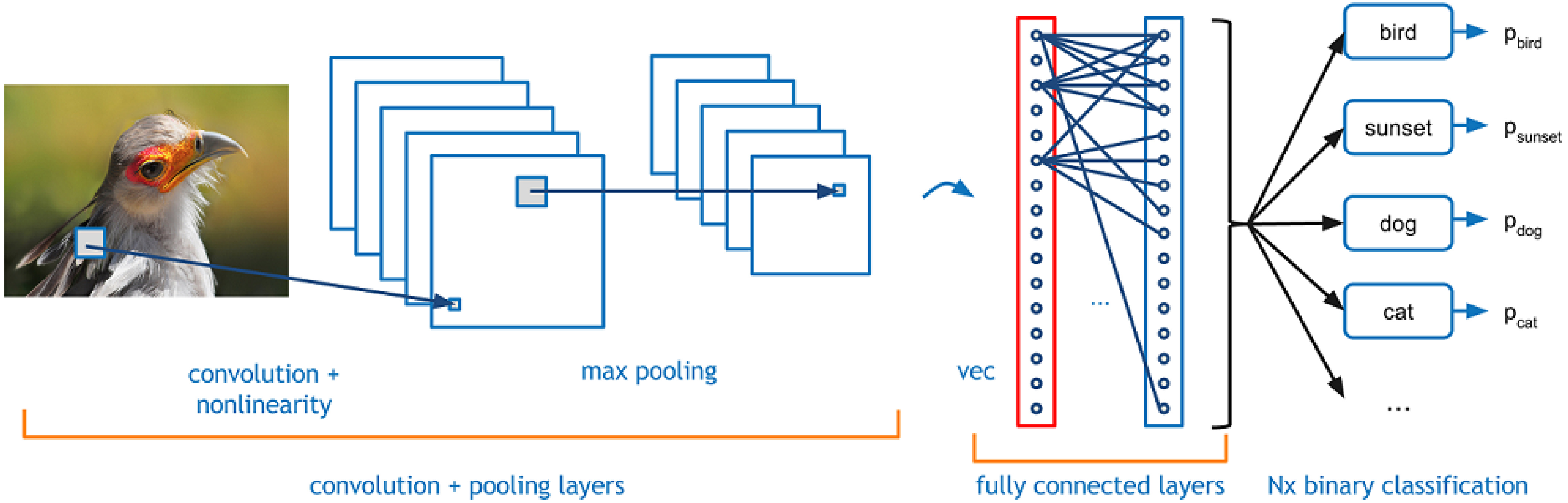
RAM 16GB ddr3

No GPU

Algorithms

Multilayer Perceptron (MLP)

Convolutional Neural Networks (CNN) ⬅



convolution + nonlinearity

max pooling

vec

...

bird

sunset

dog

cat

...

p_{bird}

p_{sunset}

p_{dog}

p_{cat}

convolution + pooling layers

fully connected layers

Nx binary classification

Our Approach

3 layers (convolution, ReLU, max pool)

window/filter size 5x5

max pool 2x2

variable depth 16-32-64

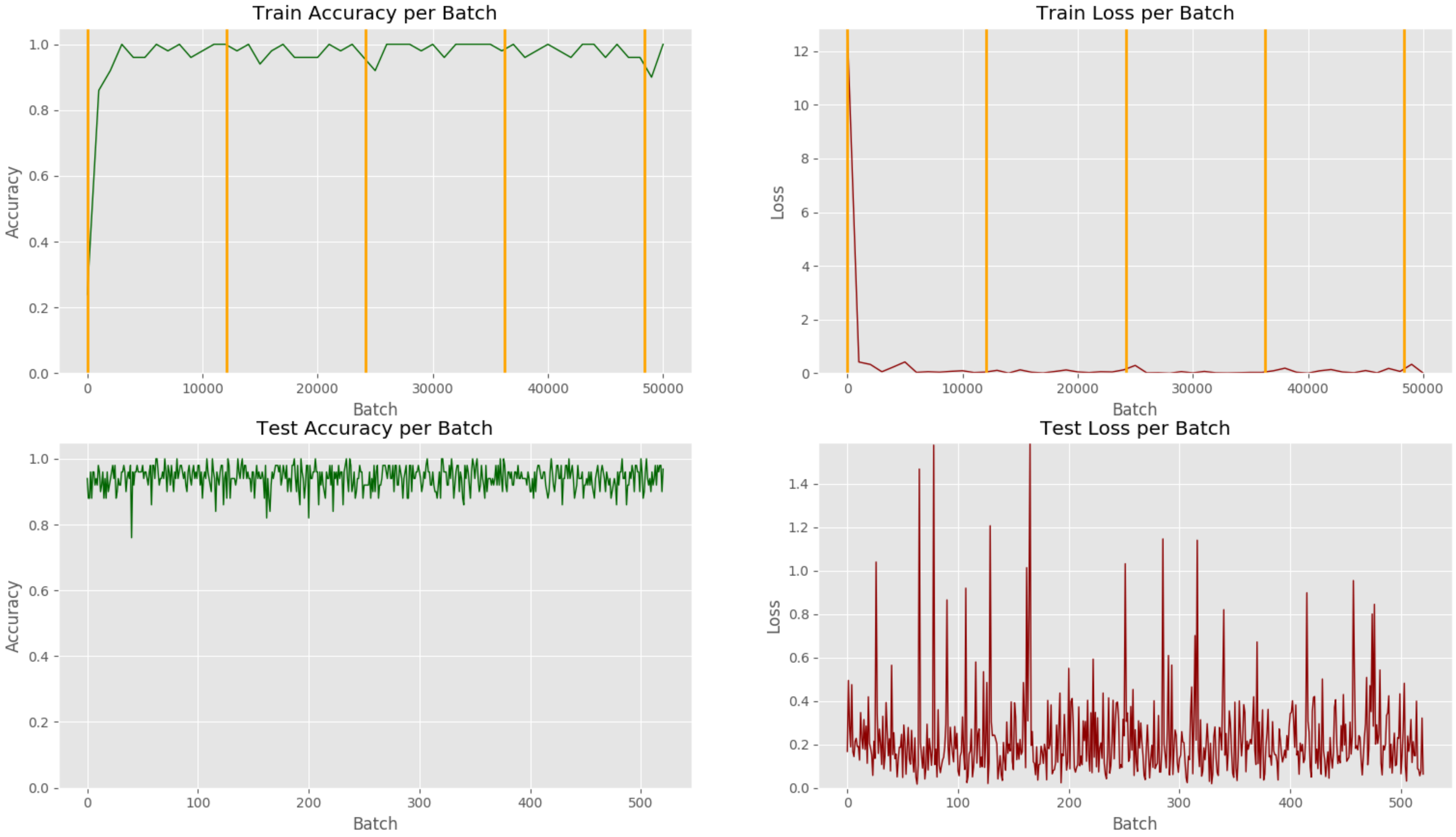
Fully connected layer

128 hidden layers

Dropout layer

0.8 keep probability

Normalization type	Gray-Scaling	Number of hidden layers	Filter(moving window) Size	Depth (layer 1, layer2, layer3)	Number of Iterations	Keep-Probability	Highest Accuracy Received
Batch Normalization(β = 0.0, γ = 1.0, ε = 0.001)	NO	128	5x5	(16, 32, 64)	50000	0.9	94.22%
Batch Normalization(β = 0.0, γ = 1.0, ε = 0.001)	NO	256	5x5	(16, 32, 64)	50000	0.9	94.09%
Batch Normalization(β = 0.0, γ = 1.0, ε = 0.001)	NO	128	5x5	(16, 32, 64)	50000	0.8	94.49%
None	NO	128	5x5	(16, 32, 64)	50000	0.8	91.21%
Batch Normalization(β = 0.0, γ = 1.0, ε = 0.001)	NO	128	5x5	(8, 16, 32)	50000	0.8	93.15%
Batch Normalization(β = 0.0, γ = 1.0, ε = 0.001)	NO	128	5x5	(16, 32, 64)	30000	0.8	93.01%
None	YES	128	5x5	(16, 32, 64)	50000	0.8	90.93%
Batch Normalization(β = 0.0, γ = 1.0, ε = 0.001)	YES	128	5x5	(16, 32, 64)	50000	0.8	94.29%
Batch Normalization(β = 0.0, γ = 1.0, ε = 0.001)	NO	128	3x3	(16, 32, 64)	50000	0.8	94.02%
None	NO	128	5x5	(32, 64, 128)	50000	0.8	20.05%


Train Accuracy per Batch
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
0
10000
20000
30000
40000
50000
Batch
Train Loss per Batch
Loss
12
10
8
6
4
2
0
0
10000
20000
30000
40000
50000
Batch
Test Accuracy per Batch
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
0
100
200
300
400
500
Batch
Test Loss per Batch
Loss
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
100
200
300
400
500
Batch

Conclusions

- Adding the extra data (containing easy cases) to the training set helps with accuracy quite considerably.
- CNN outperforms MLP (as expected since it specializes in image processing).
- Gray scaling reduces accuracy quite a lot (about 10% for MLP and 5% for CNN).
- Despite the presence of noise or multiple digits per image, the algorithm seems to be able to distinguish the central digit successfully.
- GPU increases speed of execution a lot compared to a traditional CPU.

THANK YOU

Questions?