DS-GA-1008 Assignment 1

LongCat: Catherine Olsson and Long Sha and Kevin Brown

February 9, 2015

1 Submission details

TODO what does our submission consist of

2 Summary of Modifications

Overall, we tried three significant modifications to the tutorial code provided:

1. We added a windowing step in preprocessing, designed to eliminate distracting information (including adjacent house numbers unrelated to the target digit) by obscuring the edges of the images. This modification is described in the Preprocessing section. It ultimately did not improve performance, so in the submitted model we set the width of the window to have no effect, although the ability to decrease the window size is present in the submitted code.

2. We increased the dimensionality of the first two layers, from 64 to 128. This modification is described in the Architecture section. It improved our performance from [TODO what to what?].

3. We tried varying sizes of normalization kernel [TODO what?]; however, we were able to try out these modification only on the CPU, not on the GPU, and the tradeoff with speed was not worth it in the last few days before the deadline, so we did not include these modifications in our submitted model.

3 Preprocessing

We preprocessed each image in the following manner. First, we converted the images from RGB representations to YUV, to separate luminance from color, with the working assumption that pixels that are part of address numbers will have different luminance, but possibly similar color. (Note: because torch's rgb2yuv requires floating point representation, we recast the original representation of the data.

Next, we normalized each feature globally, by subtracting the global mean for each channel, and dividing by the standard deviation (zscore), for the train data. We then used this mean and standard deviation to transform the test data (to keep training and testing data separate).

CAT ADD PREPROCESSING W/ GAUSSIAN KERNEL DETAILS:

4 Architecture

TODO summary paragraph. In particular:

- Number and type of layers
- Number of neurons
- Size of input

5 Learning Techniques

- Data augmentations?
- Dropout: We used a dropout probability of 0.5 on the third layer to regularize neurons and prevent 'coadaptation' [1]

6 Training Procedure

We used standard stochastic gradient descent, since the optimization problem is non-convex for our architecture.

- Learning Rate 1×10^{-7}
- Momentum of 0
- Loss function: negative log likelihood (nll)
- Train/validation split: 73257/26032
- Training/validation error: CAT

References

[1] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580 [cs]*, July 2012. arXiv: 1207.0580.