A Deep Learning System for Handwritten Digit Recognition

Anusha Kasi Vishwanathan

EECE.7110 HPC-GPU Project - Midterm Presentation

03-26-2018

System Overview, Dataset

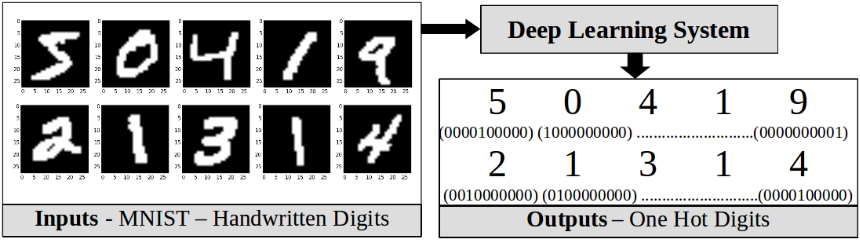


Figure 1: Overall System

- **Deep Learning System** - Based on a Convolutional Neural Network (CNN), implemented from scratch on CUDA
- **Goals** - Train the CNN to find digits in unseen test set images, achieve lower test error % than state-of-the-art - 0.23 [5]
- **Dataset** - MNIST [1] handwritten digits - 60000 train, 10000 test images, centered, scaled to 28×28 pixels, one-hot labels

Deep Learning System

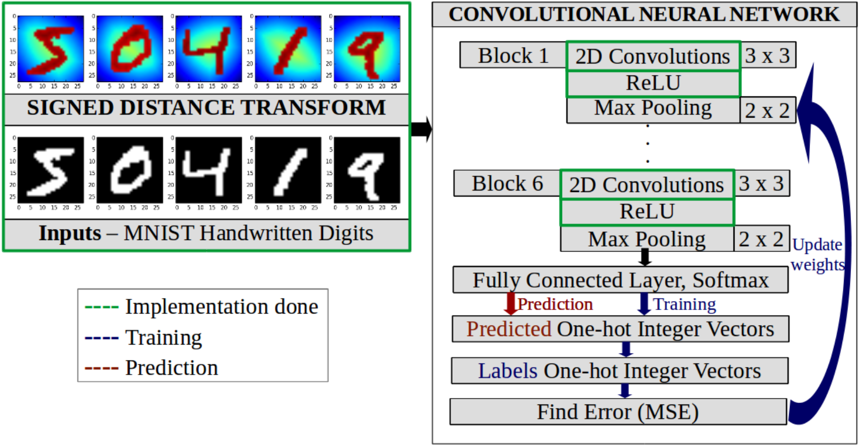
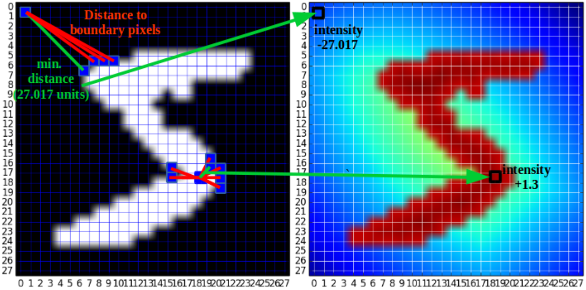


Figure 2: Deep Learning System

Signed Distance Transform Algorithm [2-4]

Algorithm 1 Signed Distance Transform

Inputs: Binary Images - Digit/Non-digit - **img**[n_{rows}][n_{cols}]
Outputs: Signed Distance Image **signedDist**[n_{rows}][n_{cols}]
for $i = 1$ to n_{rows} **do**
    for $j = 1$ to n_{cols} **do**
        $signedDist[i][j] = min(euclDist[x][y]) \forall (x, y) \ni I[x][y] = 1 - I[i][j]$
        $sign(signedDist[i][j]) =$ '+' if $I[i][j] = 1$ '-' if $I[i][j] = 0$
    end for
end for



Purpose:

-- Use as input to train and test the CNN

-- CNN learns to better distinguish the digit and non-digit pixels

--Could improve accuracy

Figure 3: Signed Distance Transform Algorithm

CUDA - Euclidean Distance

- to compute signed distances, we need euclDist between all pixel positions (denoted by i_n, j_n); calculated as

$$euclDist[i_1, j_1][i_2, j_2] = \sqrt{(i_2 - i_1)^2 + (j_2 - j_1)^2} \qquad (1)$$

- total number of operations required for this, $t = num_{rows} \times num_{cols} \times (num_{rows} \times num_{cols} - 1)/2$
- $num_{rows} = 28, num_{cols} = 28;\ t = 1 + 2 + \ldots + 784 = \frac{783 \times 784}{2}$
- $blockDim.x = 256, gridDim.x = 2$; each thread computes distances for 1 pixel

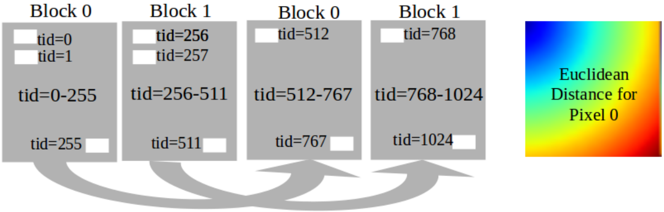


Figure 4: Euclidean Distance - CUDA Design

CUDA - Signed Distance Transform

- 60,000 train images; 784×784 operations for every image
- $blockDim.x = 784, gridDim.x = 1024$, shared $tmp\_dist[784]$
- each block deals with 1 pixel, each thread with 1 distance
- shared memory to store all boundary distances of current pixel
- find minimum distance using
  - parallel reduction
  - atomicMin() function - only supports int, convert float to int by $tmp\_dist \times 1000$, maintain this in a *signedDistInt* array
- use *__syncthreads*(), use thread 0 to assign the min to signedDist

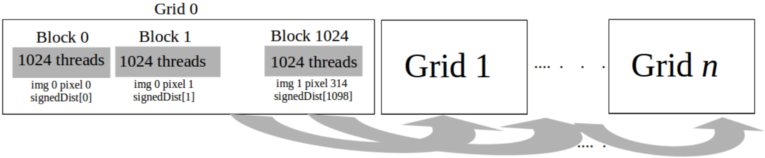


Figure 5: Euclidean Distance - CUDA Design

Signed Distance Transform - Outputs

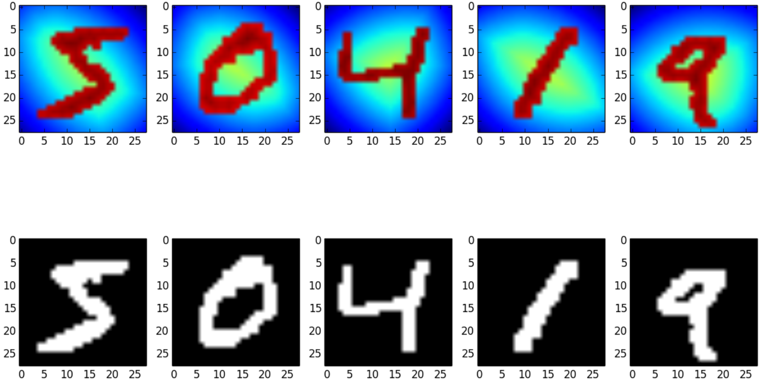


Figure 6: Euclidean Distance - CUDA Design

Convolution Operation

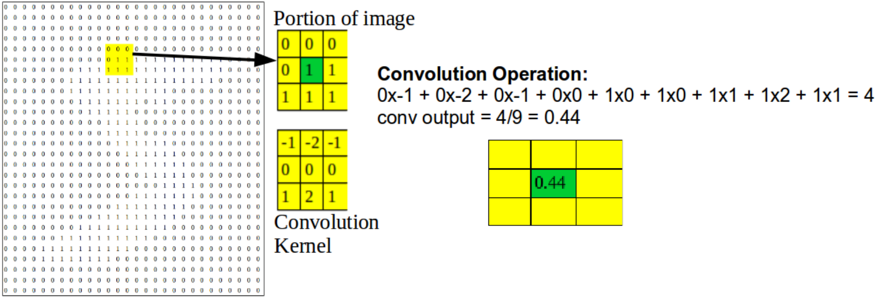


Figure 7: Convolution Operation

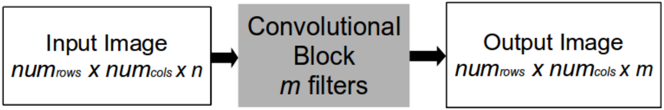


Figure 8: Convolution Layer

CUDA - Convolution Design

- a batch of train images (zero-padded), batch-size=32 -$32 \times 28 \times 28 \times 1$, kernel - 3×3
- total number of convolutions $t = 32 \times 28 \times 28$
- $blockDim.x = 1024$, $gridDim.x =$ batch-size=32
- each block deals with 1 image channel (784 convolutions)
- each thread performs one convolution
- *extern* shared memory - holds image channel of block
- the rows are repeatedly accessed for convolutions

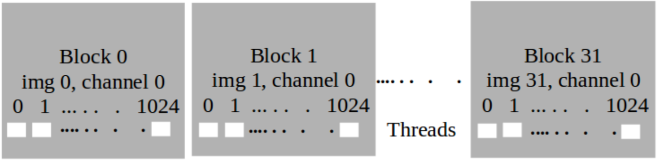


Figure 9: Convolution Layer

Convolution Sample Outputs

Using horizontal sobel edge detector [6-8] as weights (fixed)

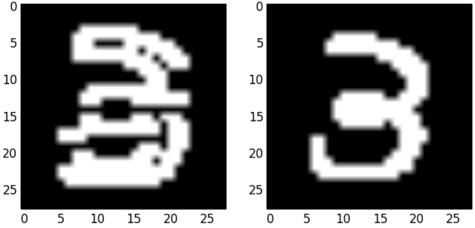


Figure 10: Convolved Output, Input Image

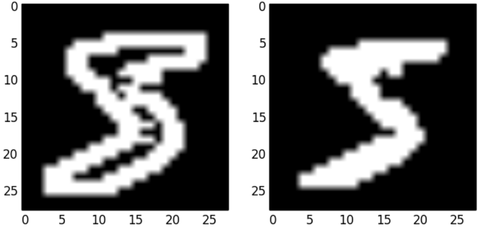


Figure 11: Convolved Output, Input Image

ReLU Layer, Performance

- activation function - to introduce non-linearity in the network
- applied to every pixel
- similar to a thresholding operation, defined as

$$ReLU(x) = max(0, x) \tag{2}$$

- not tested for now, will be tested after building the whole network
- CUDA implementation - same as convolutional blocks implementation

Task	Number of Operations	Time Taken
Euclidean Distance	$\frac{784 \times 783}{2}$	343 ms
Signed Distance (Parallel Reduction)	$60000 \times 784 \times 784$	229 ms
Signed Distance (Atomic Min)	$60000 \times 784 \times 784$	340 ms
Convolution, ReLU	32×784	$635us$

Table 1: Performance - Time Taken for Various Tasks Implemented

Progress

Tasks Completed:

- Signed Distance Transform
- Convolutional Layer
- ReLU Layer

Tasks Pending:

- Pooling Layer
- Fully Connected Layer
- Mean Squared Error Loss Function
- Gradient-based Training, Evaluation

References

[1]. LeCun, Yann, Corinna Cortes, and C. J. Burges. "MNIST handwritten digit database." ATT Labs [Online]. Available: http://yann. lecun. com/exdb/mnist 2 (2010).
[2]. Sigg, Christian, Ronald Peikert, and Markus Gross. "Signed distance transform using graphics hardware." In Proceedings of the 14th IEEE Visualization 2003 (VIS'03), p. 12. IEEE Computer Society, 2003.
[3]. Ye, Q-Z. "The signed Euclidean distance transform and its applications." In Pattern Recognition, 1988., 9th International Conference on, pp. 495-499. IEEE, 1988.
[4]. Fabbri, Ricardo, Luciano Da F. Costa, Julio C. Torelli, and Odemir M. Bruno. "2D Euclidean distance transform algorithms: A comparative survey." ACM Computing Surveys (CSUR) 40, no. 1 (2008): 2.
[5]. Cireșan, Dan, and Ueli Meier. "Multi-column deep neural networks for offline handwritten Chinese character classification." In Neural Networks (IJCNN), 2015 International Joint Conference on, pp. 1-6. IEEE, 2015.ii
[6]. Matthews, James. "An introduction to edge detection: The sobel edge detector." (2002).
[7]. Gao, Wenshuo, Xiaoguang Zhang, Lei Yang, and Huizhong Liu. "An improved Sobel edge detection." In Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on, vol. 5, pp. 67-71. IEEE, 2010.
[8]. Vincent, O. Rebecca, and Olusegun Folorunso. "A descriptive algorithm for sobel image edge detection." In Proceedings of Informing Science IT Education Conference (InSITE), vol. 40, pp. 97-107. California: Informing Science Institute, 2009.

References Contd.

[9]. Maas, Andrew L., Awni Y. Hannun, and Andrew Y. Ng. "Rectifier nonlinearities improve neural network acoustic models." In Proc. icml, vol. 30, no. 1, p. 3. 2013.
[10]. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." In Advances in neural information processing systems, pp. 1097-1105. 2012.
[11]. Wan, Li, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. "Regularization of neural networks using dropconnect." In International Conference on Machine Learning, pp. 1058-1066. 2013.
[12]. LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521, no. 7553 (2015): 436.
[13]. Schmidhuber, Jürgen. "Deep learning in neural networks: An overview." Neural networks 61 (2015): 85-117.
[14]. LeOu11, Y., L. Jackal, L. Bottou, A. Brunet, C. Cortes, J. Denker, H. Drucker et al. "COMPARISON OF LEARNING ALGORITHMS FOR HANDWRITTEN DIGIT RECOGNITION."
[15]. Belongie, Serge, Jitendra Malik, and Jan Puzicha. "Shape matching and object recognition using shape contexts." IEEE transactions on pattern analysis and machine intelligence 24, no. 4 (2002): 509-522.
[16]. Shilman, Michael, Zile Wei, Sashi Raghupathy, Patrice Simard, and David Jones. "Discerning structure from freeform handwritten notes." In Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on, pp. 60-65. IEEE, 2003.

THANK YOU!