Handwritten Digit Recognition using Hierarchical Temporal Memory

BRUCE BOBIER
bbobier@uoguelph.ca
Department of Computing and Information Science
University of Guelph, Guelph, ON, N1G 2W1

August 9, 2007

Abstract

This article describes an experiment to test the hypothesis that Numenta's NuPIC framework can achieve a recognition accuracy of 95% or greater on the USPS handwritten digit dataset. Using a training set of 7291 images and a testing set of 2007 images, NuPIC was configured to recognize 96.26% of the binary test images, and thus the hypothesis is accepted.

1 Introduction

Hierarchical Temporal Memory (HTM) was introduced by Hawkins and Blakeslee [10], and currently there have been few peer reviewed experiments that have tested the theoretical soundness and recognition ability of HTM [7]. Presently it is not known if HTMs are able to achieve or surpass the capabilities of existing approaches for simple object and character recognition in binary images.

The tree-shaped multi-level structure of an HTM reflects the nested hierarchical structures found in the world, following the notion that these hierarchies exist in both spatial and temporal dimensions. HTMs are similar to Bayesian Networks in that they both employ constant sharing of information between nodes and Belief Propagation, but differ in how hierarchy, time, and attention are used.

There are two basic functions that an HTM network performs: discovering causes and inferring the causes of novel input, where a "cause" refers to a persistent and repeating structure in the world, physical or otherwise (e.g. buildings, cats, words, songs, etc.) [11]. The first function, discovering causes, involves examining the input data in order to identify patterns that recur in both spatial and temporal dimensions. The second function involves classifying objects in the input data as belonging to a previously discovered cause and, when novel input is encountered, the attempting to determine the mostly likely high-level cause that is responsible for the input's occurrence.

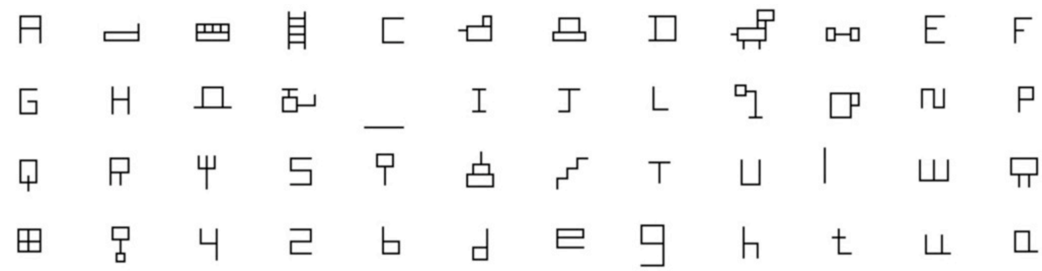

Figure 1: Examples of images that the NuPIC "Pictures" application was trained to recognize.

Every node in an HTM shares a common algorithm, regardless of its position in the hierarchy, although the hierarchical organization of the network causes the nodes at different levels to operate on different phenomenon. The nodes in lower levels deal with simple events that change quickly and occupy smaller spatial areas, while nodes in higher levels deal with multiple events that were sensed by lower level nodes, and thus are influenced by a greater range of data. Specifically, higher level nodes sense more complex causes, namely patterns of patterns, which evolve less quickly and exist in larger spatial areas. This property is similar to that of Slow Feature Analysis, which demonstrate that invariant features can be learned by employing a hierarchy that uses temporal "slowness" as a fundamental principle for learning.

HTMs are generative in nature, in that they have the ability to generate data that can be used to make predictions about what will occur in the future. Predicting what is likely to happen at the next time step is performed by combining known sequences of patterns with the current input and using this information to determine what is likely to happen next.

Currently, there have been relatively few implementations of HTMs beyond Numenta's initial proof of concept model (NuPIC). As such, in this article, an experiment is presented that aims to measure the ability of Numenta's NuPIC platform to recognize handwritten digits [8]. The dataset used for this experiment is the United States Postal Service (USPS) handwritten digit dataset, which consists of 9298 16x16 4-bit grey scale images. As the system used for the experiment is based on a modified version of the the NuPIC "Pictures" application which was designed to recognize objects such as ladders, cats, stairs, and characters from 32x32 pixel binary images (Figure 1), the dataset was binarized and resized.

The hypothesis to be tested by the experiment is that the NuPIC implementation of HTM can achieve a recognition rate of 95% or better on the USPS handwritten digit dataset.

The remainder of this article is organized as follows. The following section provides a introduces the USPS dataset. Section 3 presents the operational definitions and experimental parameters and Section 4 describes the experimental methodology. Section 5 presents the results of the experiment and Section 6 discusses the results and concludes.

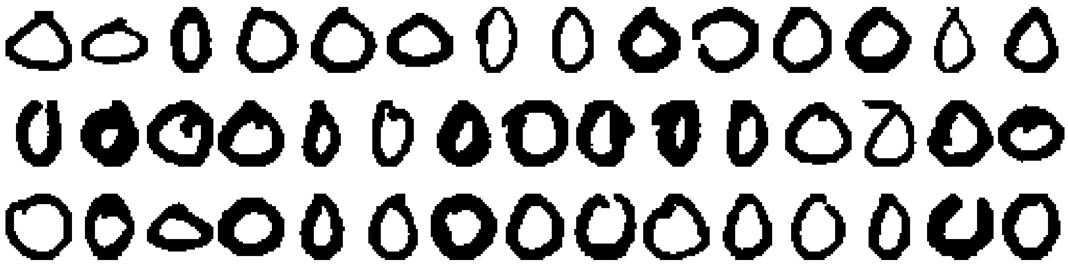

Figure 2: Examples of the digit 0 in the binarized USPS training dataset.

Digit	Training Set	Testing Set
0	1194	359
1	1005	264
2	731	198
3	658	166
4	652	200
5	556	160
6	664	170
7	645	147
8	542	166
9	644	177

Table 1: Number of instances for each digit in the training and testing sets.

2 Data Set

The United States Postal Service (USPS) dataset is comprised of ten classes of handwritten digits (0-9), with 7291 images in the training set and 2007 images in the testing set. Each image is 16x16 pixels in 4-bit grey scale (256 colours) in the original dataset. The current release of the NuPIC "Pictures" application is designed to work with 32x32 pixel binary images, and thus the dataset was binarized using Otsu's binarization algorithm [14] as it was found to provide the most accurate results [1], and resized to 32x32 pixels using Matlab. Figure 2 shows examples of the digit 0 from the training set after binarization and resizing.

The NuPIC pictures application requires two tarred and gzipped files (`.tgz`): one for training and one for testing. Each `.tgz` file is organized to have a directory for each object class, and the directory name is used as the class name (e.g. directories "0","1",... "9"). Each folder contains uniquely named bitmapped binary images, although the number of images varies greatly between classes. Table 1 illustrates that there are nearly twice as many training instances of the digit 0 as there are of the digit 9.

As the majority of related recognition experiments that use the USPS dataset for testing recognition approaches us the 16x16 grey scale (c.f. [2, 4, 12, 13]), direct comparison of results is not reliable. However, the dataset used for this experiment allows the digit recognition of the NuPIC implementation of HTM to be tested, and a similar experiment may be conducted on a future releases of the NuPIC platform that can operate on grey scale images.

3 Operational Definitions and Parameters

For this experiment, recognition accuracy is the primary operational definition that is measured. Recognition accuracy is defined as the number of incorrectly classified images in the testing set divided by the total number of test images.

Another important term in the following discussion is "coincidence" which, in the context of HTM, refers to a specific combination of two or more patterns that are likely to occur together at a single point in time.

With the NuPIC HTM implementation, there are numerous parameters that can be adjusted to alter the network topology and performance. For this experiment, the following parameters were altered and their influence measured (in terms of recognition accuracy): `maxDistance`, `sigma` and `topNeighbors`.

`maxDistance` is a parameter that controls the behaviour of the spatial pooling algorithm, by specifying the maximum permissible Euclidean distance that can occur between two new coincidences in order for them to be grouped together. By increasing the value of `maxDistance`, the number of coincidences is decreased, which in turn decreases the number of groups. To use a geometric analogy, in each bottom level node (i.e. L1 nodes), the node's the spatial pooler defines a hyper-sphere with a radius equal to the square root of `maxDistance` that is centred at the new coincidence. If the new input lies within an existing hyper-sphere, then the pattern is pooled with the coincidence represented by the centre of the sphere. However, if the new pattern is not within any existing hyper-sphere, then a new coincidence is learned and its hyper-sphere is added to the input space. Thus, a larger value of `maxDistance` causes fewer but larger hyper-spheres to compose the input space.

`sigma` is a parameter that is closely related to `maxDistance`, except that whereas `maxDistance` is a learning parameter, `sigma` is used during inference. To continue the previous geometric analogy for `sigma` during inference, when a pattern is provided to an L1 pooler, the belief value for each coincidence is computed using an L_2-norm between the input and the centre of each coincidence's hyper-sphere. The belief value for each coincidence decays exponentially as the distance between the input pattern and coincidence centre increases, and it is this rate of decay that is governed by `sigma`. Configuring the system with a small value for `sigma` (i.e. a fast decay) results in only the coincidences that are very close to the input point getting non-negligible belief values. Further, using a large value for `sigma` causes a slow decay, thereby causing coincidences that are distant from the input pattern to have non-negligible beliefs.

`topNeighbors` is used to control the behaviour of the spatial pooling algorithm, such that by increasing the value of the parameter, the "aggressiveness" of the grouping algorithm is increased. Generally, a larger value results in each node having fewer groups, with each group containing more coincidences.

4 Methodology

This experiment aims to test the hypothesis that the NuPIC implementation of HTM can achieve a recognition accuracy equal to or greater than 95%. The methodology to test this hypothesis consisted of first establishing the parameter values with which the network will be tested. Three parameters, namely `maxDistance`, `sigma` and `topNeighbors` were varied based on the default settings for the "Pictures" application and the recommendations provided through personal correspondences with Numenta's Jeff Edwards [6]. The default and evaluated parameter values are given in Table 2.

Parameter	Default Value	Evaluated Value	Evaluated Value
maxDistance	0.0	1.0	1.5
sigma	1.08	1.2	1.0
topNeighbors	3.0	2.0	4.0

Table 2: Default and evaluated parameter values for the NuPIC HTM.

Once the parameter values were selected, the next step involved transforming the dataset into a format suitable for NuPIC. In the first phase of experimentation, the HTM was configured with all parameters set to their default values and the results of testing (i.e. recognition accuracy and which digits were incorrectly classified) were recorded. During the second and third runs, `maxDistance` was set to 1.0 and 1.5 and the results testing were recorded. The parameter value of these three that produced the highest recognition accuracy was then used for the subsequent phases, wherein the influence of altering `topNeighbors` was evaluated in the same manner. Lastly, the influence of altering `sigma` was examined. It is acknowledged that the tournament styling of this methodology does not ensure that optimal settings are being discovered, and that the network may be more finely tuned for this problem. However, due to the considerable computational cost of each run, only a limited number of parameter values could be tested.

Additionally, the network topology of this experiment remains fixed across each run. The topology for the HTM network used for the USPS dataset is given in Figure 3 and described in the remainder of this section.

At the bottom level of the network (L1), each node is divided using an 8x8 grid that covers the entire image without overlap, where each node can see a 4x4 pixel viewing window of the image at any one time. That is, the image is divided into 64 viewing windows, with each node at that level seeing one window at a time. During the training process, each node "looks through" sequential viewing windows, in the order of top-left to bottom-right.

At the second level of the network (L2), each node receives input from the 16 L1 nodes. Specifically, a 2x2 grid is used to divide the input space, with each viewing window seeing 16x16 pixels. At the top level of the network, a single node receives input from all of the viewing windows at the previous level.

Although additional topologies were not tested in this experiment, future work may examine the influence of different network topologies. For example, using a 4x4 grid

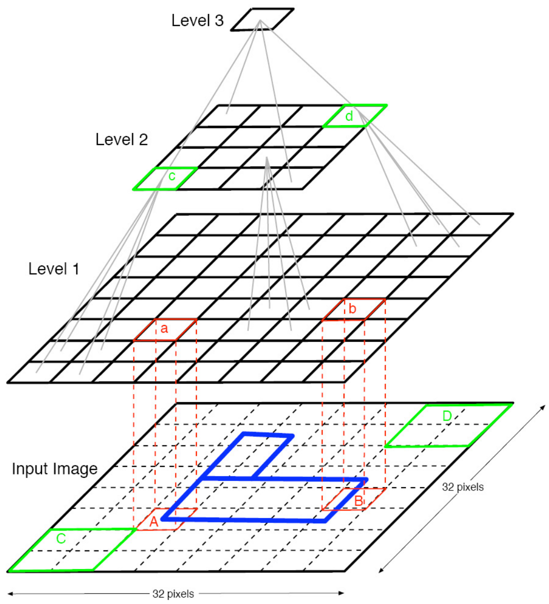

Figure 3: An HTM with 3 levels. (adopted from [9])

in L2, or adding another level of nodes for learning may provide better performance. However, for this to be tested, additional computational resources are recommended.

5 Results

In the first phase of experimentation, the influence of changing the value of `maxDistance` was observed. By default, the NuPIC "Pictures" application has `maxDistance` set to 0.0, `topNeighbors` set to 3.0 and `sigma` set to 1.08. Using these parameter values for the first run, the system achieved a recognition accuracy of 89.24% (1791 images correctly recognized). For the second run, `maxDistance` was increased to 1.0 and the other parameters were left unchanged. This resulted in a recognition accuracy of 96.26% (1932 images). For the third run, the value of `maxDistance` was further increased to 2.0, which resulted in a recognition accuracy of 93.92% (1885 images).

The next phase of the experiment involved observing the influence of changing the value of `topNeighbors`, with `maxDistance` set to 1.0 (as this value produced the highest recognition accuracy). As with the previous phase, the default setting of `topNeighbors`=3 resulted in a recognition accuracy of 96.26% (1932 images). Decreasing the value of `topNeighbors` to 2.0 resulted in a recognition accuracy of 91.28% (1832 images), while setting `topNeighbors` to 4.0 resulted in a recognition accuracy of 87.39% (1754 images).

The final phase of experimentation involved observing the influence of changing `sigma`, with `maxDistance`=1.0 and `topNeighbors`=3. With `sigma` set to its default value of 1.08, a recognition accuracy of 96.26% (1932 images) was achieved. By increas-

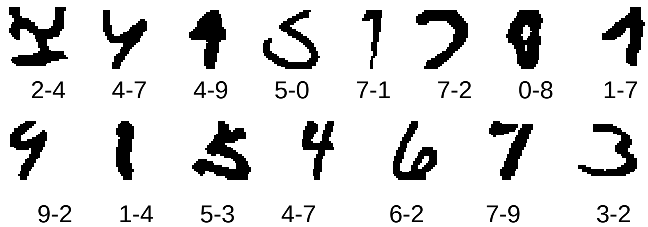

Figure 4: Examples of the errors for the USPS test set. The number in the bottom-left corner of each image indicates the true class, and the second digit is the recognized digit label.

ing `sigma` to 1.20, the recognition accuracy was 93.47% (1876 images), while setting `sigma` to 1.0 resulted in a recognition accuracy of 92.48% (1856 images).

Figure 4 shows examples of the digits that were misclassified by the best performing network, configured with `maxDistance`=1.0, `topNeighbors`=3 and `sigma`=1.08. Below each image, the number on the left indicates the true class of the digit and the number on the right indicates the class label as which the image was misclassified. From this figure, the images in the bottom row are likely more apparent to the reader as being as recognition errors. For example, the first digit in the bottom row looks more like a 9 than a 2, and the fourth digit more closely resembles a 4 than a 7, although they were both misclassified. Examining the images in the top row, many of the digits can be easily misclassified by the reader, and it is not surprising that the system misclassified them (e.g. the fifth digit being misclassified as a 1 rather than a 7).

6 Conclusion

In the previous section, it was shown that the NuPIC implementation of HTM is able to achieve a recognition accuracy of 96.26% on the binarized USPS handwritten digit data set. From this, the hypothesis that the NuPIC implementation of HTM can achieve a recognition rate of at least 95% on the USPS handwritten digit dataset is accepted.

Comparing these results to those of previous studies, it was found that NuPIC's implementation of HTM does not provide results that are statistically significantly more accurate than any of the five leading approaches to digit recognition on the USPS dataset. The results of these comparisons is given in Table 3.

However, each of these studies used the 16x16 grey scale image dataset, and it is possible that similar or better results could be obtained using a future version of NuPIC, although this is left as future work.

While implementing the NuPIC system, several limitations were encountered. The first limitation is that training a network on the 9298 images is a time consuming process; on a Windows laptop with 1GB RAM and a 2.13 MHz processor running NuPIC through VmWare, the training process typically took between 7 and 14 hours to complete. Secondly, the current release of NuPIC does not support many environments

Method	Recognition Accuracy
Cho [2]	96.05
NuPIC HTM [11]	92.26
S.W. Lee [12]	97.8
Local Learning Framework [4]	98.1
Virtual SVM [5]	98.7
SVC-rbf [13]	98.9

Table 3: Comparison of the accuracy of other approaches with the USPS dataset.

and configurations. Specifically, NuPIC requires either Suse Linux Enterprise 10, Fedora Core 6, Mac OS 10.4, or Red Hat Enterprise Linux 4 update 4, and does not work on any other operating systems without a considerable amount of modifications to the source code. In addition to the strict operating system requirements, NuPIC also requires Python version 1.4.4 (as NuPIC does not run properly with versions 1.4.1 and 1.5), and that version 1.4.4 be "hacked" to run with 4-bit unicode. Attempts to install the platform on the graduate lab's computers under Linux and Windows (through VmWare) and the undergraduate Mac computers was unsuccessful because none of the computers available to the University of Guelph Computer Science graduate students satisfy the 1GB minimum RAM requirements and software requirements.

An additional limitation of the NuPIC platform is that the NuPIC virtual machine (nuvm-1) is configured to only work on the VmWare software (which does not run on the lab computers due to the hardware requirements not being met), and does not run in other virtual machines such as Qemu.

Because of the considerable limitations of NuPIC, only a limited number of runs could be conducted for this experiment. Until NuPIC is able to run stably in additional environments with less stringently specified software versions, the potential for further experimentation is greatly limited. Further, the computational time requirements of the current version are considerably larger than of other approaches (e.g. SVM [3, 13]), which further reduces the feasibility of the NuPIC platform for large scale projects.

References

[1] Bruce Bobier. Evaluation of binarization algorithms for preprocessing elevation drawings. *Unpublished*, University of Guelph, Guelph, ON, March 2007.

[2] S. B. Cho and J. H. Kim. Combining multiple neural networks by fuzzy integral for robust classification. *SMC*, 25(2):380–384, 1995.

[3] Jian-Xiong Dong. HeroSvm 2.1, 2007. http://www.cenparmi.concordia.ca/ people/jdong/HeroSvm.html. [Online; accessed 08-August-2007].

[4] Jian-Xiong Dong, Adam Krzyzak, and Ching Y. Suen. Local learning framework for handwritten character recognition. *Engineering Applications of Artificial Intelligence*, 15(2):603–618, April 2002.

[5] Jian-Xiong Dong, Adam Krzyzak, and Ching Y. Suen. Fast svm training algorithm with decomposition on very large data sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):603–618, 2005.

[6] Jeff Edwards. Private communication, 2007.

[7] Saulius Garalevicius. Memory-prediction framework for pattern recognition: Performance and suitability of the Bayesian model of visual cortex. Paper accepted for FLAIRS-20, 2007.

[8] Dileep George and Bobby Jaros. Hierarchical temporal memory – zeta1 algorithms reference. Numenta Inc. Whitepaper, 2007.

[9] Dileep George and Bobby Jaros. The htm learning algorithm. Numenta Inc. Whitepaper, 2007.

[10] Jeff Hawkins and Sandra Blakeslee. *On Intelligence*. Times Books, 2004.

[11] Jeff Hawkins and Dileep George. Hierarchical temporal memory – concepts, theory, and terminology. Numenta Inc. Whitepaper, 2007.

[12] Seong-Whan Lee. Multilayer cluster neural network for totally unconstrained handwritten numeral recognition. *Neural Networks*, 8(5):783–792, 1995.

[13] Cheng-Lin Liu, Kazuki Nakashima, Hiroshi Sako, and Hiromichi Fujisawa. Handwritten digit recognition using state-of-the-art techniques. In *IWFHR '02: Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02)*, page 320, Washington, DC, USA, 2002. IEEE Computer Society.

[14] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.