MNIST BENCHMARK
WITH NEUROMEM® NETWORKS

The MNIST database is a large database of handwritten digits that is commonly used to validate Machine Learning systems and Convolutional Neural Networks (CNN).

As of today, CNNs deliver error rates ranging between 1.7% and 0.21% depending on their complexity and numbers of layers. However, their common validation criteria are to learn a dataset which is six times larger than the testing set and to as correct a good response among the top N responses (usually N=5).

This document describes a series of experiments made with a NeuroMem neural network to learn and classify the MNIST database. It is not a final benchmark, but rather a demonstration of the promising performances of a multiplicity of NeuroMem NNs trained on simple features and modeling complementary or redundant decision spaces.

To provide a fair comparison for NeuroMem, one experiment follows the standard criteria and delivers 99.32% accuracy with a single NeuroMem network (one-hidden layer) trained on 60K subsamples of the images.

However, our main goal is to demonstrate that the NeuroMem technology is a practical, explainable, and responsible AI technology.

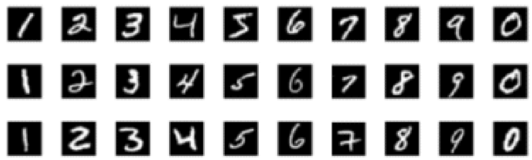
- **Practical** because it does not need large amount of training data and its latencies to learn and recognize depend on the number of samples and their length, but not on their content nor their relationship to one another. Consequently, all our experiments except one use the 10K images for learning.
- **Responsible** because our classification criteria are never Top N, but rather based on dominant category within a single network or preferably a consensus of responses from multiple networks. Not knowing or being uncertain is an acceptable output for a single network paving the way, if necessary, for more training or the recourse of another opinion. Consequently, all our experiments except one use the 60K images for testing and we compare RBF and KNN with Top1 and Dominant consolidations.
- **Explainable** because the feature vectors identified as discriminant and retained by the neurons are actually stored in their memory. If necessary, to trace results, these models can be retrieved when the neurons fire. This is how we found out that at least one image of the 60K dataset is labeled incorrectly.

1 CONTENTS

2 The dataset 3
2.1 Side note about the dataset accuracy 3
2.1.1 Understanding the challenge of the MNIST classification 3
3 The NeuroMem Experiments 4
3.1.1 Multiple simple features 4
3.1.2 2 scenarios: 4
3.1.3 3 consolidation rules to produce a single Output category 5
3.1.4 3 definitions of “Accuracy” 5
3.1.5 2 classifiers 5
4 Result Overview and Observations 7
5 Results in detail 8
5.1 Simplest feature 8
5.1.1 Remark about the “Incorrect” cases 8
5.2 complementary features? 9
5.2.1 Subsample 24x20 with blocks 1x2 9
5.2.2 Subsample 24x20 with blocks 2x1 9
5.2.3 Using a combination of the two feature sets 10
5.3 Shortest feature 10
5.4 More complex feature 11
6 Timings 11
6.1 Feature extraction 11
6.2 Feature recognition 11

2 THE DATASET

The MNIST database of handwritten digits has a training set of 60,000 examples, and a test set of 10,000 examples. The digits have been size-normalized and centered in a fixed-size image of 28x28 pixels. The resulting images contain contrasted grey levels because of the anti-aliasing technique used by the normalization algorithm. A survey of the data shows that all digits are included in a rectangle of 19x19 pixels at the largest.



2.1 SIDE NOTE ABOUT THE DATASET ACCURACY

One of the diagnostics reports of NeuroMem Knowledge Builder framework has pinpointed the incorrect labeling of at least one image as shown below.


train-4-5831.png


train-4-5832.png


train-4-5833.png


train-4-5834.png


train-4-5835.png

Consequently, learning the image train-4-5834.png will introduce inconsistencies in the knowledge build by the neurons causing unnecessary uncertainties, especially on the classification of the digit 4.

2.1.1 Understanding the challenge of the MNIST classification

The NeuroMem neurons learn what they are taught. Would you have annotated the following reference images as such?

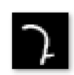
4 versus 9 ?


train-4-1216.png


train-9-5376.png

2 versus 7 ?


train-7-3249.png


train-2-2601.png

3 THE NEUROMEM EXPERIMENTS

Our experiments were executed in a few mouse clicks with the General Vision's NeuroMem Knowledge Builder framework. For more information, https://www.general-vision.com/documentation/TM_NeuroMem_KB.pdf

The parameters of the different experiments are the following:

3.1.1 Multiple simple features

Using a script written in MatLab, several feature vectors were extracted from the 28x28 images of the 10K and 60K datasets.

- Context1: SubsampleGrey(img, ctrX, ctrY, 24, 24, 2, 2); % 12 x 12 blocks
- Context2: SubsampleGrey(img, ctrX, ctrY, 24, 20, 1, 2); % 24 x 10 blocks
- Context3: SubsampleGrey(img, ctrX, ctrY, 24, 20, 2, 1); % 12 x 20 blocks
- Context4: SubsampleGrey(img, ctrX, ctrY, 21, 20,3, 4); % 7x5 blocks
- Context5: Hog(img, CtrX, CtrY, 24,24)

3.1.2 2 scenarios:

- **Learning 10K images and Validation on 60K images** (default scenario)
  - In order to prove that the NeuroMem neurons can model the decision space properly with satisfying generalization, it is essential to learn a dataset significantly smaller than the testing set.
- Learning 60K images and Validation on 10K images (to give a fair comparison to NeuroMem benchmark)

- The scenario using a learning set 6 times bigger than the testing set is the standard for most MNIST benchmarks, so we wanted to give a fair trial to our neurons on at least one of the feature set.
- As noted earlier, we know that at least one of the images of the 60K dataset is labeled incorrectly, so this scenario shows how the neurons can still adapt and model a proper decision space. The accuracy is slightly better than in our recommended default scenario but the number of vectors used for the validation is 6 times lesser!

3.1.3 3 consolidation rules to produce a single Output category

- Best match or Top1
- Dominant category among the top K responses (*)
- TopK (correct category if it is listed among the top K responses(*). **This 3rd consolidation rule is used for benchmarking purposes only. It is not deployable in real application since the Ground Truth is not known.**

We used K=5

(*) Note that in the RBF mode there might be less than K firing neurons.

(**) Note that in NeuroMem Knowledge Builder we offer additional more conservative rules such as a Minimum Consensus between the firing neurons. Such approach can deliver high accuracy when using multiple and complementary classifiers (refer to the conclusion paragraph).

3.1.4 3 definitions of "Accuracy"

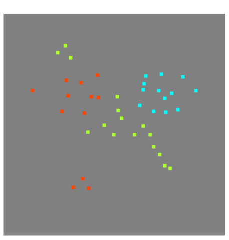
- Correct= Output category matches the Ground Truth Category
- Incorrect= Output category does not match the Ground Truth Category
- Unknown or N/A (*)

(*) In the case of the RBF mode, there can be cases of Unknown classification. They are not accounted as Correct nor Incorrect, but rather as "Wise" response, implying the need for more training or the use of another network (aka feature) to discriminate the digit number.

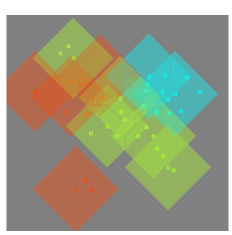
3.1.5 2 classifiers

- Radial Basis Function (RBF)
- K-Nearest Neighbor (KNN)

Given a training dataset (2D in this example)…



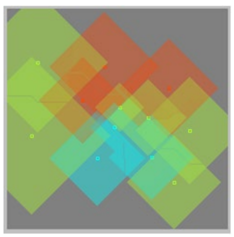
The neurons model the decision space adjusting autonomously their influence field to include new examples and never contradict the teacher.



The neurons entertain zones of Unknown (grey color) and zones of Uncertainty (overlapping influence field).

Classification with RBF mode

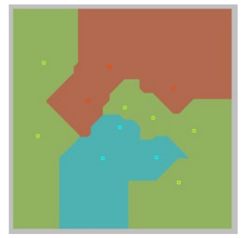
For a given input vector, the neurons can produce 3 types of classification status:
- **Identified** (the vector falls into the influence field of one or more neurons with the same category),
- **Uncertain** (the vector falls into the influence field of multiple neurons with different categories),
- **Unknown** (the vector falls outside any neuron's influence field)



The RBF "honesty" and level of details can be very powerful to flag the non-relevancy or non-efficiency of a feature to discriminate between specific categories and to imply the need for more training or the use of a different feature.

Classification with KNN mode

All the neurons respond. The classification status can be Identified or Uncertain, BUT the range of similarity between the input vector and the K closest models can be quite stretched sometimes.



KNN never reports any Unknown classification, but rather dispatches the inputs which would be classified as Unknown in RBF mode to the "closest but still possibly far" matches. Furthermore, KNN can report closer matches with incorrect categories in the case of convex decision space for example.

The accuracy in KNN is always higher than in RBF because the portion of Unknown is distributed between Identified and Uncertain, but it can resort to throwing a dice.

4 RESULT OVERVIEW AND OBSERVATIONS

Learning a feature set extracted from the 10K dataset commits in average between 2000 and 2300 for the 5 different feature sets.

Learning a subsample12x12 from the 60K dataset committed more than 8000 neurons. The resulting accuracy is not far superior. We can suspect that approximately 2K neurons model the bulk of the dataset and the remaining neurons are describing exceptions.

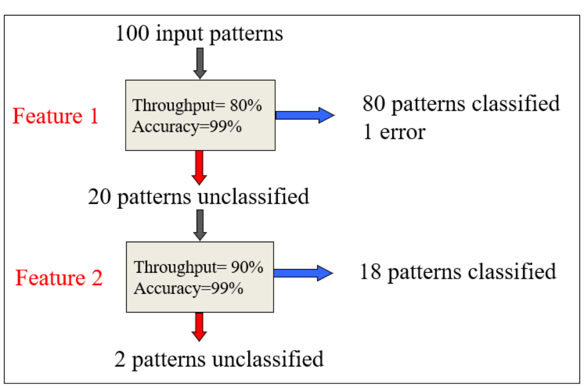
Learning 60K ➔ **99.32%** accuracy on the 10K dataset **in KNN Top5** with the subsample 12x12.

For the reason explained in the introduction, all the other tests use a practical approach to learn on 10K and validate on the much larger set of 60K.

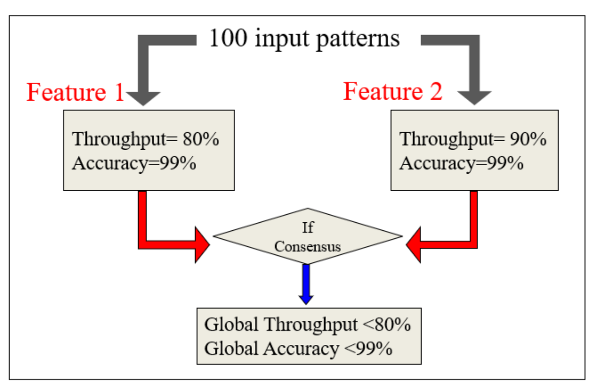
Learning 10K Reco 60K	**Academic Best accuracy (KNN Top 5)**	**Realistic accuracy (RBF Dominant, excluding Unknown)**	Remark
Subsample 12x12	98.50%	95.20%	
Subsample 24x10	98.43%	95.40%	These 2 feature sets can be combined to handle mutually exclusive cases of unknown and uncertainty and increase accuracy.
Subsample 12x20	98.12%	95.46%	
Subsample 7x5	98.26%	93.27%	Surprisingly good performance for such short vector. Can be a 1st classifier for high-speed discrimination, triggering a 2nd classifier to handle its cases of Unknown and Uncertainty
Hog	98.61%	93.63%	Feature more complex to calculate, and not delivering a significant advantage over subsample 12x12

The Unknown and Uncertain classifications can be waived by combining NeuroMem networks trained on different features emphasizing different aspects of the digit patterns.

Example of 2 cascaded networks



Example of 2 complementary and parallel networks



5 RESULTS IN DETAIL

5.1 SIMPLEST FEATURE

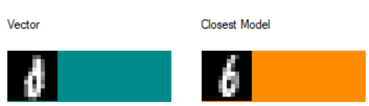
Subsample 24x24 with blocks 2x2
The resulting image is a compressed version of the image down to 12x12

Subsample 12x12	Consolidation rule	Neurons	Correct	Incorrect	Unknown
Learn Data_10K		2009			
Reco Data_10K	RBF Top1		100%		
Reco Data_60K	RBF Top1		82.58%	5.04%	12.39%
Reco Data_60K	RBF Dominant with K=3		83.37%	4.25%	12.39%
Reco Data_60K	RBF Dominant with K=5		83.41%	4.2%	12.39%
Reco Data_60K	RBF Top5		84.88%	2.74%	12.39%
Reco Data_60K	KNN Top1		87.66%	12.34%	N/A
Reco Data_60K	KNN Dominant with K=5		90.51%	9.49%	N/A
Reco Data_60K	KNN Top5		98.50%	1.50%	N/A

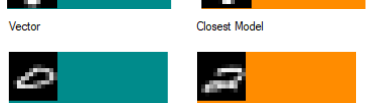
5.1.1 Remark about the "Incorrect" cases

When learning, the NeuroMem neurons act as an RBF model generator, only storing the novel and significant patterns into their memories and adjusting their respective influence fields if necessary. When classifying a new pattern, the firing neurons can return their category, but also distance or confidence level, and their identifier. Consequently, it is possible to pull the content of the firing neurons if necessary.

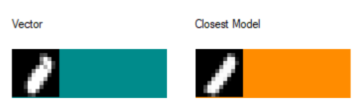
The NeuroMem Knowledge Builder features a utility to filter the vectors which are misclassified and trace the model of the closest firing neuron. This utility helps understand why the classification of some digits can be incorrect as shown in the examples selected below:



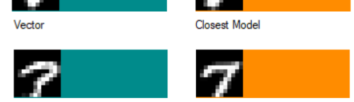
0 recognized as a 6



0 recognized as a 2



0 (?) recognized as a 1



3 (?) recognized as a 7

The next experiment is the only one learning of the 60K dataset. As noted earlier, we know that at least one image is labeled incorrectly. Results demonstrate that the neurons can still adapt and model the decision space with 89.67% accuracy in RBF with dominant category (a practical output) and 99.28% accuracy in KNN Top5.

Subsample 12x12	Consolidation rule	Neurons	Correct	Incorrect	Unknown
Learn Data_60K		8285			
Reco Data_60K	RBF Top1		100%		
Reco Data_10K	RBF Top1		88.83%	3.55%	7.62%
Reco Data_10K	RBF Dominant with K=5		89.76%	2.62%	7.62%
Reco Data_10K	RBF Top5		90.83%	1.55%	7.62%
Reco Data_10K	KNN Top1		91.63%	8.37%	N/A
Reco Data_10K	KNN Dominant with K=5		94.99%	5.01%	N/A
Reco Data_10K	KNN Top5		**99.32%**	0.68%	N/A

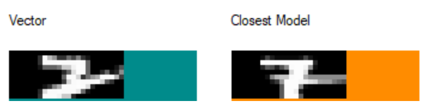
5.2 COMPLEMENTARY FEATURES?

5.2.1 Subsample 24x20 with blocks 1x2

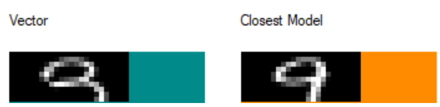
Compresses the pattern vertically into 24x10 pixels

Subsample 12x10	Consolidation rule	Neurons	Correct	Incorrect	Unknown
Learn Data_10K		2060			
Reco Data_10K	RBF Top1		100%		
Reco Data_60K	RBF Top1		82.79%	4.88%	12.33%
Reco Data_60K	RBF Dominant with K=5		83.63%	4.03%	12.33%
Reco Data_60K	RBF Top5		85.02%	2.65%	12.33%
Reco Data_60K	KNN Top1		87.58%	12.42%	N/A
Reco Data_60K	KNN Dominant with K=5		90.84%	9.16%	N/A
Reco Data_60K	KNN Top5		98.43%	1.57%	N/A

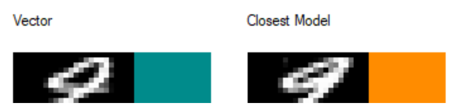
Some of the Incorrect responses



2 (?) recognized as a 7



3 (?) recognized as a 9



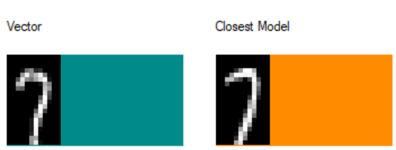
4 (?) recognized as a 9

5.2.2 Subsample 24x20 with blocks 2x1

Compresses the pattern horizontally into 12x20 pixels

Subsample 12x10	Consolidation rule	Neurons	Correct	Incorrect	Unknown
Learn Data_10K		2079			
Reco Data_10K	RBF Top1		100%		
Reco Data_60K	RBF Top1		82.89%	4.68%	12.44%
Reco Data_60K	RBF Dominant with K=5		83.59%	3.97%	12.44%
Reco Data_60K	RBF Top5		85.07%	2.5%	12.44%
Reco Data_60K	KNN Top1		87.72%	12.28%	N/A
Reco Data_60K	KNN Dominant with K=5		90.11%	9.89%	N/A
Reco Data_60K	KNN Top5		98.12%	1.88%	N/A

Some of the Incorrect responses



2 (?) recognized as a 7

3 (?) recognized as a 1

5.2.3 Using a combination of the two feature sets

The following tables report the recognition and accuracy status for the two features and their complementary to waive some unknown and uncertainties.

The classification was RBF dominant category K=3

Recognition Status

Subsample 24x10 \ Subsample 12x20	**UNK**	**ID**	**UNC**
UNK	7.6%	4.5%	0.3%
ID	4.4%	71.1%	3.9%
UNC	0.3%	3.5%	4.4%

7.6% of the images of the 60K dataset are not recognized by either feature set.
71.1% are identified by both feature sets.
The interesting information is that the 2 feature sets have partially exclusive domains of unknown and uncertainty:
3.5% recognized with uncertainty with the Subsample 24x10 are positively identified with the Subsample 12x20.
3.9% recognized with uncertainty with the subsample 12x20 are positively identified by the subsample 24x10.

The current NeuroMem Knowledge Builder framework does not support the experimentation between inter-feature consolidation rules, but this is under development.

5.3 SHORTEST FEATURE

Subsample 21x20 with blocks 3x4
The resulting image is a compressed version of the image down to 12x12

Subsample 7x5	Consolidation rule	Neurons	Correct	Incorrect	Unknown
Learn Data_10K		2253			
Reco Data_10K	RBF Top1		100%		
Reco Data_60K	RBF Top1		80.44%	6.84%	12.72%
Reco Data_60K	RBF Dominant with K=5		81.42%	5.87%	12.72%
Reco Data_60K	RBF Top5		83.37%	3.92%	12.72%
Reco Data_60K	KNN Top1		85.65%	14.35%	N/A
Reco Data_60K	KNN Dominant with K=5		89.49%	10.51%	N/A
Reco Data_60K	KNN Top5		98.26%	1.74%	N/A

5.4 MORE COMPLEX FEATURE

Histogram of Gradient (HOG) of the 24x24 pixels

Hog	Consolidation rule	Neurons	Correct	Incorrect	Unknown
Learn Data_10K		2197			
Reco Data_10K	RBF Top1		100%		
Reco Data_60K	RBF Top1		81.26%	6.58%	12.16%
Reco Data_60K	RBF Dominant with K=5		82.25%	5.59%	12.16%
Reco Data_60K	RBF Top5		84.15%	3.69%	12.16%
Reco Data_60K	KNN Top1		86.34%	13.67%	N/A
Reco Data_60K	KNN Dominant with K=5		90.43%	9.57%	N/A
Reco Data_60K	KNN Top5		98.61%	1.39%	N/A

6 TIMINGS

Unlike any other system, the learning and recognition latency of a NeuroMem network is directly proportional to the number of samples and their length. This means that the latency does NOT depend at all on the content of the samples nor their relationship to one another.

The timing for recognition can be decomposed in 2 tasks:
- Feature extraction
- Feature recognition

6.1 FEATURE EXTRACTION

The Subsampling is a simple feature extraction which involves a simple averaging of blocks of pixels within a region of interest. In a FPGA this feature can be assembled while reading the pixel values.

The HOG or Histogram of Gradients is a more complex feature extraction involving rotations and histogram binnings.

6.2 FEATURE RECOGNITION

The time to recognize a vector is non related to the number of neurons committed in the network, but it is related to the length of the vector to broadcast to the neurons and the value K which indicates how many queries to read the categories of the closest firing neurons.

Timings are supplied for a system clock of 18 Mhz which is the recommended clock for a chain of multiple NM500 chips (576 neurons/chip)

Consolidation rule	Subsample 12x12	Subsample 24x10 or 12x20	Subsample 12x10	Subsample 5x7 or 7x5
Vector length	244	240	120	35
K=1	267 cc / 14.8 us	263 cc / 14.6 us	143 cc / 7.9 us	58 cc / 3.2 us
K=5	351 cc / 19.5 us	347 cc / 19.3 us	227 cc / 12.6 us	142 cc / 7.9 us