NeuroMem

Knowledge Builder

Pattern Learning and Classification with a NeuroMem network

Version 2.3.1
Revised 08/23/2018

GENERALVISION

1 Contents

NeuroMem Knowledge Builder is a product of General Vision, Inc. (GV)

2 Introduction

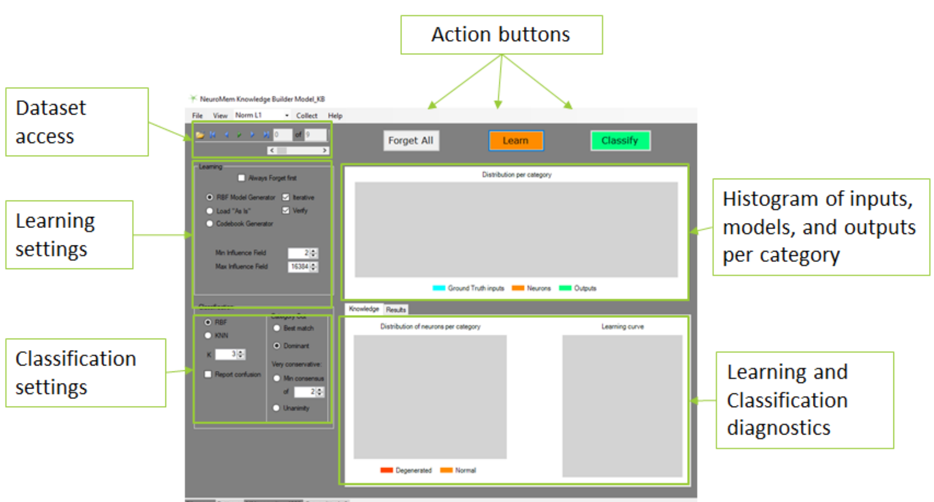
The NeuroMem Knowledge Builder is an application which allows learning, recognizing and clustering patterns, or feature vectors, using a NeuroMem neural network. The software can interface to a single NeuroStack board, or a stack of boards. In the absence of NeuroMem hardware it will simulate a chain of four CM1K chips (or 4096 neurons).

2.1 What can I do?

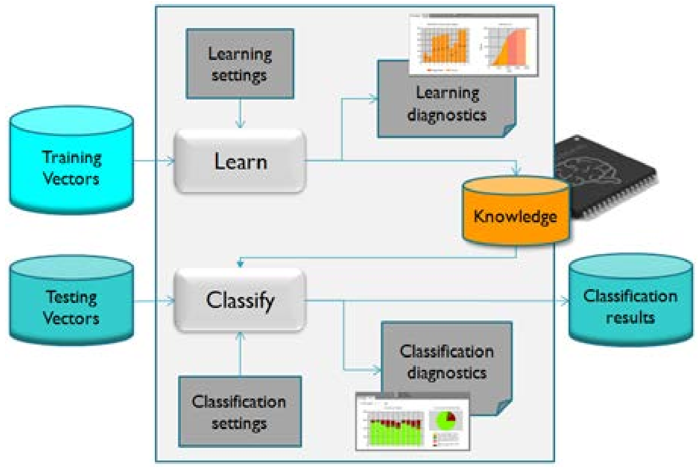
- Learn a training set and review the diagnostics report
  - Monitor the learning curve
  - Compare the histogram of the input vectors and learned models per category
  - Identify the categories which are easy to discriminate just by observing the ratio between the input vectors and learned models
- Recognize the training set and review the diagnostics report
  - Identify the categories with high accuracy and high throughput, as well as the categories recognized with poor accuracy
  - Identify the categories with confusion
  - Poor accuracy at this stage means that (1) the knowledge has been built on an insufficient number of vectors, or (2) a non-representative number of vectors.
- Recognize a testing set and review the diagnostics report
  - Identify the categories with high accuracy and high throughput, as well as the categories recognized with poor accuracy
  - Identify the categories with confusion
- Classify a new set and output its recognized categories
- Clusterize a dataset to group its vectors into significant categories (version Pro only)

2.2 A simple panel

The application is very easy to use with access to the functions and their diagnostics in a simple panel, and ability to review and save a data logger.



2.3 Simple dataset manipulations



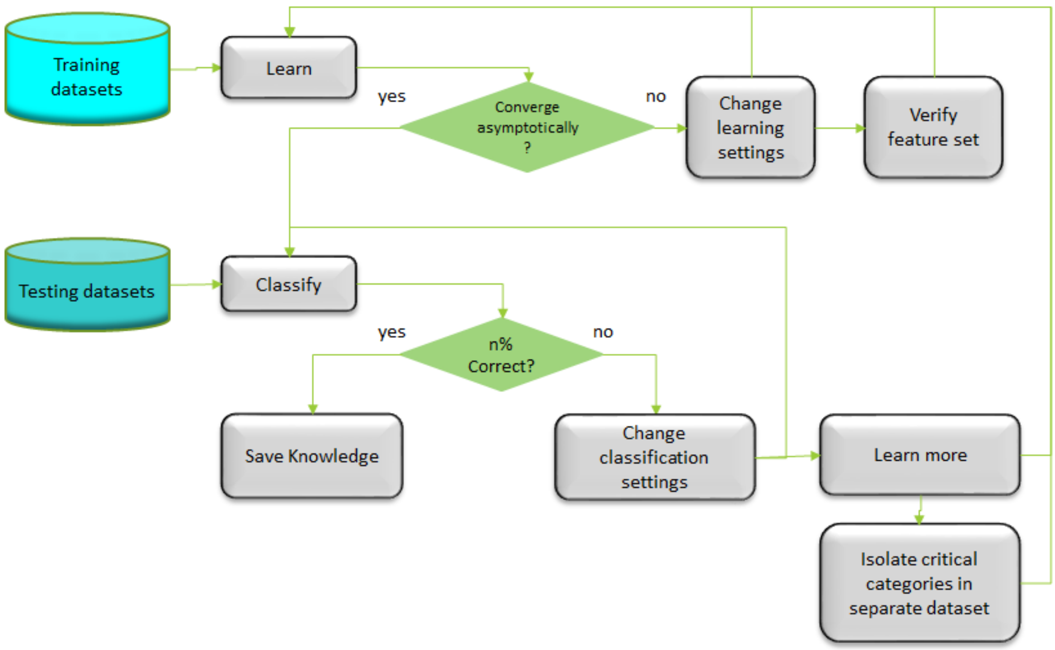
A Training file is composed of vectors which are annotated with a category (Ground Truth category). This dataset can be learned or loaded into the neurons and produces a knowledge base.

A Testing file is composed of vectors which are also annotated with a category. These vectors can be classified by the neurons and the resulting category is compared to the annotated category to quantify the accuracy of the knowledge base.

Diagnostics reports generated during both the training and the classification help identify simple and critical cases of classification. If the accuracy is unsatisfactory, you can easily try (a) learning more training vectors, (b) changing the learning parameters to be more or less conservative, or (c) consider changing the input vectors in use because they do not discriminate properly or sufficiently the annotated categories.

Once a knowledge is validated, the NeuroMem Knowledge Builder can be used to classify new datasets.

2.4 Typical Workflow



1) Prepare your training and testing datasets so their format is readable by NeuroMem Knowledge Builder
   a) Convert your patterns to fit in an array of dimension 256 and with values ranging between [0, 255]
   b) Save your patterns in csv format according to the format described in the next chapter
2) Learn training set: vectors with Ground Truth (GT) categories
3) Verify that the neurons can over generalize
   a) Learning curve must be asymptotic (proof that the neurons can generalize)
   b) The ratio between created models and input vectors must be less than 1 to confirm that the neurons can over-generalize
   c) Remedy1: verify accuracy of the GT categories
   d) Remedy2: consider the classification of the same objects using a different feature or signature
4) Identify categories of objects more complex than others
   a) The ratio between created models and input vectors per category can pinpoint at categories difficult to model
   b) Observe the distribution of degenerated neurons per category, if any
5) Verify that the neurons properly recognize the learned vectors
   a) Accuracy should 100% correct. Correct Uncertain are acceptable.
   b) Verify that all vectors are recognized properly
   c) Observe that the number of uncertainties is minimal
   d) Identify categories introducing confusion
6) Recognize a new testing set with GT categories
   a) Verify that all vectors are recognized properly
   b) The acceptable n% accuracy is specific to your application
   c) Verify that the number of Identified Incorrect are null
      i) Remedy1: Expand the learning set
      ii) Remedy2: Increase the MINIF to convert Identified_Correct into Uncertain cases
   d) Identify the Uncertain cases and fine tune their classification with a second feature extraction, if applicable
7) Save the knowledge in a portable file format

3 The Datasets

The NeuroMem Knowledge Builder support text or csv files saved in the comma delimited format described below. This format is the same for training and testing sets.

The NeuroMem neurons are a powerful model generator and non-linear classifier. Today, their commercial implementation as digital neural networks limit the memory size of the neurons to 256 bytes on the General Vision CM1K chip and 128 bytes on the Intel QuarkSE chip.

This means that the patterns to learn and classify must be formatted to fit on this dimension prior to learning or classification: The length of the pattern must be less than or equal to 256/128. The amplitude of the vector must range between 0 and 255.

Examples of datasets with the expected header are supplied in the folder myDocuments/General Vision/Datasets.

3.1 Input format

The application can load files saved in comma delimited format and complying with the simple format described below.

PatternID (optional)	ParentID (optional)	Context (*) (optional)	CatGT	Length	v(0)	v(1)	…	v(L)
Index of the pattern in the input list	User-defined index identifying the origin of the pattern (a file name, XY position, time stamp, etc.)	Value identifying the type of feature encrypted in the vector.	Ground Truth Category assigned to the vector, if applicable 0xFFFF otherwise	Number of components L in the vector	L values of the vector components			
Default= automatic increment of 1, starting at 0	Default =0	Range=[1, 127] Default = 1	Range=[1,32768] 0 = "background" 0xFFFF = "unspecified"	Default Range=[1,256] for all NeuroMem platform, except Intel QuarkSE/Curie =[1,128]	Values ranging between [0,255]			

The block of the first 3 columns is optional and will be filled automatically if the header of the file starts with "CatGT".

If you format your dataset to include the 3 columns "PatternID, ParentID, Context", suggested default values are a numeric series with increment of 1 for PatternID, zeros for ParentID and ones for Context.

(*) Refer to the NeuroMem Reference Guide for information about the notion of Context and how to use it. The NeuroMem Knowledge Builder supports multiple contexts, but the diagnostics are generated for one context at a time. This means that your datasets can include patterns representing different physical dimensions and have different lengths. The Learn and Classify functions will execute on specific context at a time and produce diagnostics for this context. You have the option to Learn and Classify all contexts combined, but the interpretation of the diagnostics must be made with caution.

3.2 Output format

The results generated by the Classify command are appended to the input dataset in the format described below. The value K is defined in the Preferences menu (default is 3) and determines the length of the output record.

Results can be saved along with the Input data using the Save Data & Results command, or solely using the Save Results Only command.

Status code	Accuracy code	CatOut	Cat 1	Dist 1	Nid1	Cat2	Dist2	Nid2	…	CatK	DistK	NidK
UNK=unknown ID=identified UNC=uncertain	Correct Incorrect N/A	Global response of the firing neurons	Response of the 1st firing neuron including its distance, category and identifier.			Response of the 2nd firing neuron				Response of the Kth firing neuron		
		Choice of BestMatch or Dominant category	In KNN mode, there are always K firing neurons In RBF mode, K is a maximum applicable									

4 The Main UI

4.1 File menu

Load dataset	Load a dataset saved in the comma delimited format described below (*.txt or *.csv)
Import dataset	Load a dataset deriving from knowledge files saved with the NeuroMem (*.knf) or CogniSight API (*.csp)
Save dataset	Save a dataset along with the result of the last recognition in a comma delimited format, with optional filter per recognition status
Load knowledge	Load an existing knowledge file to the neurons overwriting the previous knowledge
Save knowledge	Save the content of the committed neurons to a knowledge file

4.2 View menu

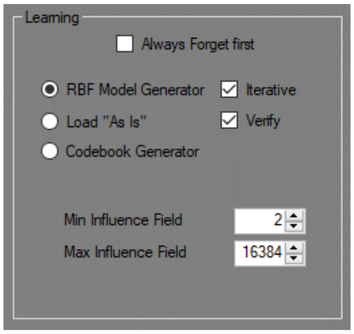
Knowledge	View the contents of the neurons and trace to the patterns which triggered their commitment
Dataset in detail	Access the information of each entry of the dataset before or after recognition and plot its vector profile
Platform	Select a hardware platform with NeuroMem silicon neurons or a simulation platform.

4.3 Action buttons

Forget All	Clear the content of the neurons
Learn	Learn the vectors of the selected Context using the method and settings selected in the Learning frame.
Classify	Classify the vectors of the selected Context using the method and settings selected in the Classification frame.

5 Learning

5.1 The Learning Methods



The NeuroMem neurons can be trained with various methods and settings:

- RBF Model Generator (supervised learning)
- Load As Is (for KNN classification only)
- Codebook Generator (Pro version only)

For more information about Learning, refer to the NeuroMem Technology Reference Guide and the NeuroMem Decision Space mapping manual.

Minimum and Maximum Influence Fields

The Minimum and Maximum Influence Fields are also global variables of the NeuroMem network and their values should be set in relation to the dimension represented by the feature vectors and the distance calculation in use (L1 or LSup).

MAXIF:
Control the initial level of moderation of the neurons. This limit applies to the neurons which are not yet committed, but ready to learn. It is a maximum value assigned to a newly committed neuron if it does not fall within the influence field of already committed neurons. Otherwise its influence field is set to its distance to the closest committed neuron.

- Range [2, 65534]; Default 16384
- The larger the MAXIF, the more generalization or over-fitting. Tendency to create lesser liberal neurons.
- The lesser the MAXIF, the less generalization. Tendency to create many conservative neurons.

MINIF:
Control the final level of uncertainty of the committed neurons. This limit applies to the neurons which are not yet committed, but ready to learn. It is the minimum value below which the influence field of a newly committed cannot shrink.

- Range [2, MAXIF]; Default 2
- The larger the MINIF, the more uncertain in the decision space. Tendency to create degenerated neurons which overlap on one another. Can help flag outliers, reveal non discriminating features and more.

RBF Model Generator

Learn repeatedly the input patterns with a Ground Truth Category different from 0xFFFF and stops when no more neuron gets committed between two passes. This stop condition ensures that the resulting knowledge is independent from the order in which the vectors are stored in the file.

Important Options:

- Iterative (default ON); if OFF, the patterns are broadcasted only once and consequently the knowledge is built faster and requires less neurons, but might not be as accurate since the order in which the patterns are learned affects the decision space
- Always Forget first; to automatically clear the neurons prior to the next learning operation
- Verify (default ON), Automatically attempts to recognize the learned vectors to verify that accuracy is 100% and uncertainties are minimum, if any

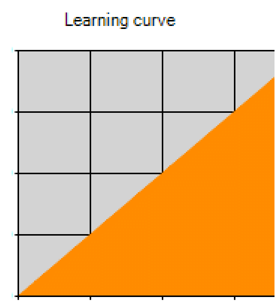
- Min Influence Field (default 2), if desirable to moderate the cases of uncertainty cases
- Max Influence Field (default 16384), if desirable to converge faster by generating conservative neurons from the beginning of the learning (as opposed to starting with moderate neurons and using the learning iterations to fine tune the accuracy)

Load As Is

This function is not a learning method, but rather a loading method during which the neurons are turned into dummy memories. Their learning logic is disabled meaning that pattern duplication can occur and their Influence Fields are all set to the current value of the Maximum Influence Field.

This function is of interest to execute a pure KNN classification to evaluate distances between the input patterns. It clears the neurons currently committed and load them with the input patterns and their Ground Truth Category, discarding any entry where the GT category is equal to 0xFFFF.

The learning curve is a straight line with a slope of 1 as shown to the right.



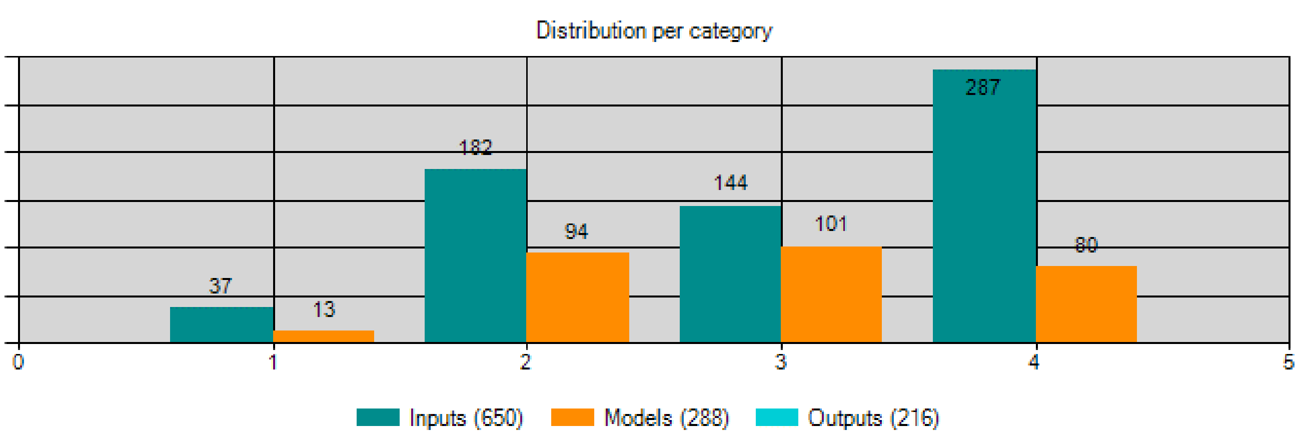
Codebook Generator (Pro version only)

Broadcast the input patterns to the neurons and let them learn autonomously the ones detected as novelty assigning their category incrementally.

Important Options:
- Max Influence Field (default 16384), important parameter to moderate the occurrence of novelties
- Always Forget first; to automatically clear the neurons prior to the next learning operation
- Min Influence Field (default 2)

5.2 The Learning Diagnostics

Ratio Models over Input patterns



The ratio between the number of input patterns submitted to the neurons for learning and the number of models retained by the neurons is an indicator of how well the neurons can over generalize the surveyed population.

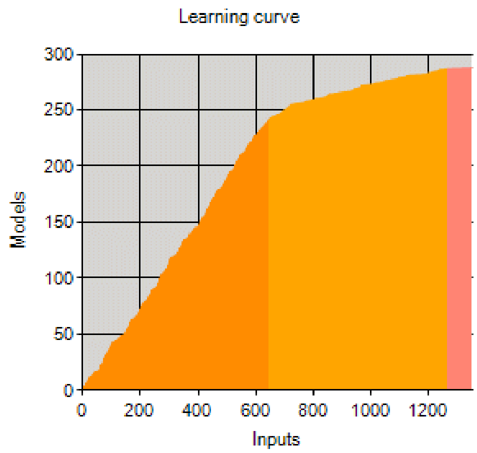
The smaller the ratio (Neurons/Inputs), the better the generalization performed by the neurons. Note that the accuracy of this generalization still has to be verified with a Diagnostics after recognition.

The closer to 1 the ratio (Neurons/Inputs), the less generalization occurs. This may have different interpretations:

- Not enough vectors have been learned
- The feature or signature encoded in the vector is not relevant to characterize the category
- Another category is very close in the decision space

The Learning curve

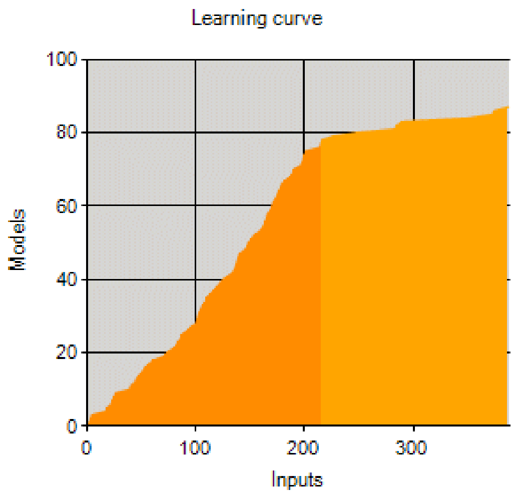
The learning curve should be asymptotic meaning that after enough examples, the neurons are capable of generalization, and hopefully capable of recognizing similar patterns never seen.



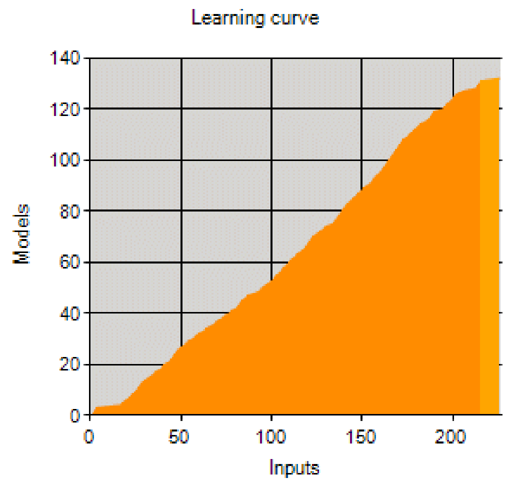
In the example to the left, the input patterns were submitted three times each before the knowledge became stable. The number of neurons committed at each iteration is shown in a different color.

Impact of the maximum influence field (selected in the Preferences menu)

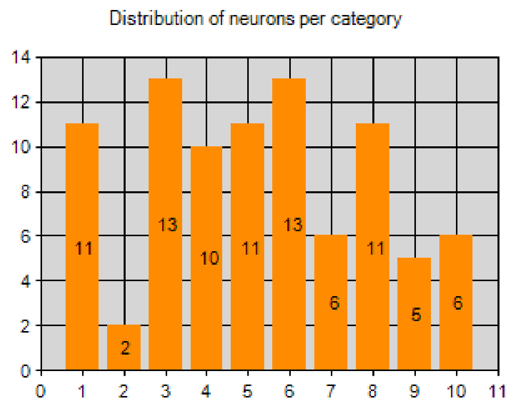
Maxif=16,000
88 neurons are committed in 2 iterations

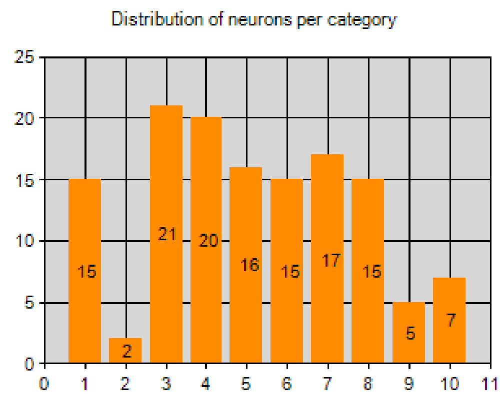


Maxif= 8000
133 neurons are committed in 2 iterations.



The neurons created with the smaller Maxif are smaller and consequently more numerous to cover the same decision space. Their histogram per category do not have the same amplitude but have the same profile. The neurons to the right will behave in a more conservative manner, generating less false positive but producing a lesser throughput.





Remark about speed performances

The Learn function not only trains the neurons but also generates a wealth of information for diagnostics purpose such as the association of which pattern commits which neuron, the learning curve or rate of commitment of the neurons through the broadcast of the patterns and more. This data is valuable for diagnostics purposes, but obviously implies some computing overhead.

Please note that this overhead will not exist in a standard learning operation where patterns and GT categories are broadcasted to the neurons with no need for feedback at each commitment of a new neuron.

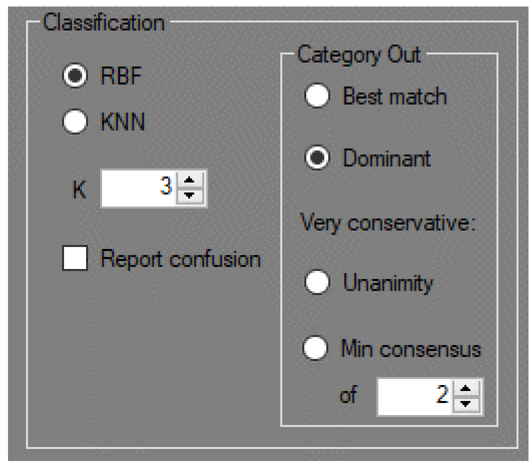
6 Classification

The Recognize function broadcasts each vector to the neurons and reads the response of the K closest firing neurons per vector. In addition, the function generates a global response, or Category Out, based on a rule also selected by the User.

The responses of the neurons are composed of the following triplets: a category (Cat), a distance (Dist) and an identifier (Nid). The distance represents the difference between the input pattern and the model held by the neuron. It is inverse proportional to a confidence factor.

6.1 Classification Settings

RBF or KNN



Choice to classify the patterns using the RBF classifier or a K-Nearest Neighbor.

K best responses

K is a value selected by the user and can range between 1 and the number of committed neurons. Typical values are 1, 3 or 5.

The recognition of a vector will return up to K triplet values (Cat, Dist, Nid) in the case of an RBF classification, and exactly K triplet values (Cat, Dist, Nid) in the case of a KNN classification.

1 global response: Category Out

Out of the K best responses, the function returns a global category per pattern, Category Out.

Category Out is the value used to determine if the recognition is accurate or not by comparing it to the Ground Truth category. Changing the decision rule can affect the percentages of correct and incorrect.

The global response can be defined as

- Best Match: The category of the first firing neuron (Cat1)
- Dominant: The dominant category among the K firing neurons
- Unanimous: The category unanimously recognized by the K firing neurons; category Unknown otherwise
- Min consensus: The dominant category provided that it has a minimum count; category Unknown otherwise

The last two rules are conservative and produces more Unknown and therefore Incorrect than the first 2 rules. They can be useful to locate the patterns and categories difficult to discriminate. The Export function of the File menu lets you separate the entries of your datasets into subsets for further studies. The latter can be divided into subsets such as “Easy to discriminate”, “Difficult to discriminate”, “High confusion”, etc.

RBF versus KNN

A NeuroMem network behaves by default as a Radial Basis Function and more precisely a Restricted Coulomb Energy classifier. It can also be set as a K-Nearest neighbor classifier. This option can be set in the View/Preferences menu as well as the value K. For more information about the KNN classifier, please refer to the NeuroMem Technology Reference Guide.

RBF

The number of firing neurons is variable. The program reads the response of the closest firing neurons stopping after K of them.

In the report below, each row corresponds to an input pattern and you can observe that some vectors are recognized by a single firing neuron (Cat 1, Dist 1, Nid 1), others by two or three firing neurons.

Accuracy	CatOut	Cat 1	Dist 1	Nid 1	Cat 2	Dist 2	Nid 2	Cat 3	Dist 3	Nid 3
Correct	2	2	3212	2						
Correct	2	2	3518	2						
Correct	2	2	4519	2						
Correct	2	2	7510	2						
Correct	1	1	1865	6	2	1956	10			
Correct	1	1	1930	6						
Correct	1	1	1758	6	2	1765	10			
Correct	1	1	1543	6	2	1952	10	1	3828	1
Correct	1	1	1540	6	1	3651	1			
Correct	1	1	1492	6	1	3701	1			

KNN

All neurons always fire. The program reads the response of the K closest firing neurons.

The Status code can never be Unknown. Note that the amount of incorrect classifications can considerably increase since the accuracy is ranked by comparing the Ground Truth category to the Category of the closest match regardless of its distance value.

Accuracy	CatOut	Cat 1	Dist 1	Nid 1	Cat 2	Dist 2	Nid 2	Cat 3	Dist 3	Nid 3
Correct	1	1	2056	6	2	2271	10	2	2961	7
Correct	1	1	2096	6	2	2273	10	2	2997	7
Correct	1	1	2200	6	2	2323	10	1	2989	1
Incorrect	2	2	2332	10	1	2347	6	1	2842	1
Incorrect	2	2	2350	10	1	2433	6	1	2756	1
Incorrect	2	2	2363	10	1	2434	6	1	2755	1
Incorrect	2	2	2364	10	1	2393	6	1	2796	1
Incorrect	2	2	2342	10	1	2383	6	1	2806	1
Correct	1	1	2334	6	2	2335	10	1	2855	1
Correct	1	1	2282	6	2	2327	10	1	2907	1

6.2 Classification Diagnostics

The Status and Accuracy codes are based on the aggregated category, CatOut, calculated with the selected rule (best match, dominant, unanimity, minimum consensus).

Status code

- ID, identified
  - CatOut assigns a non-zero category to the pattern which is approved by all top K firing neurons
- UNC, uncertain
  - CatOut assigns a non-zero category to the pattern but this category results from a rule aggregating multiple categories. This is a flag that some of the top K firing neurons disagree with the category.
- UNK, unknown
  - CatOut is equal to zero. This can occur if the pattern is not recognized by any neuron, or if it is recognized but the consolidation rule requires to discard the responses of the neurons due to lack of consensus.

Accuracy code

- Correct= CatOut matches the Ground Truth Category, if any
- Incorrect= CatOut does not match the Ground Truth Category, if any
- NA= non-applicable, the Ground Truth Category is not known so accuracy cannot be verified

AccuracyPerStatus code

The classification diagnostics reports the distribution of the Correct and Incorrect classifications for the entire dataset and per Ground Truth category.

- The ideal case is 100% Correct with 100% Identified.
- The worst case is to have any Incorrect Identified. This is a flag that the training set may have erroneous Ground Truth categories, or that the vectors are not an appropriate feature to classify and discriminate the intended population of objects or events.

Accuracy	Total	Cat0	Cat1	Cat2	Cat3	Cat4	Cat5	Cat6	Cat7	Cat8	Cat9	Cat10
Correct	80.36%	0%	9.77%	11.85%	7.18%	8.22%	6.88%	6.88%	9.16%	7.99%	5.23%	7.18%
> Identified	81.05%	0%	10.27%	11.69%	7.96%	8.02%	7.01%	7.04%	9.85%	7.08%	5.94%	6.19%
> Uncertain	18.95%	0%	1.9%	3.06%	0.98%	2.21%	1.55%	1.52%	1.55%	2.87%	0.57%	2.75%
Incorrect	19.64%	0%	1.24%	0.41%	2.97%	2.21%	2.31%	2.28%	1.27%	2.06%	2.01%	2.89%
> Identified	27.65%	0%	1.55%	0.26%	3.49%	2.71%	3.62%	2.97%	2.58%	3.36%	2.84%	4.26%
> Uncertain	28.04%	0%	1.42%	0.9%	2.33%	4.39%	3.75%	3.36%	1.55%	2.97%	0.9%	6.46%
> Unknown	44.32%	0%	3.36%	0.9%	9.3%	4.13%	4.39%	5.3%	2.33%	4.13%	6.46%	4.01%

The Confusion Matrix

The confusion matrix highlights the categories which can be mismatched together.

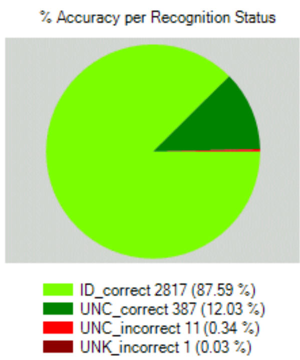
Each row corresponds to a Ground Truth category and the columns report the counts per recognized category. For example Confusion[A,B]=K means that K patterns assigned with the Ground Truth category A are actually recognized a category B. The numbers along the diagonal (Confusion[A,A]) are the numbers of correct responses. Any other number reports uncertainties between the category A and B (Confusion[A,B]).

In the following example, the highest confusion resides between the categories [3 and 8], [3 and 5], [4 and 9].

GT	Unknown	Cat 0	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5	Cat 6	Cat 7	Cat 8	Cat 9
Cat 0:	38	926	0	1	4	0	2	6	0	3	0
Cat 1:	32	2	945	2	0	1	0	4	8	4	2
Cat 2:	99	3	0	871	9	1	1	0	8	7	1
Cat 3:	109	2	0	6	835	0	19	0	3	20	6
Cat 4:	76	1	2	0	0	861	0	6	2	6	28
Cat 5:	107	2	1	0	28	0	737	5	0	11	1
Cat 6:	50	2	3	2	0	11	7	881	1	0	1
Cat 7:	83	0	11	5	8	3	0	1	866	1	22
Cat 8:	134	6	4	5	21	4	12	2	3	775	8
Cat 9:	103	1	3	2	4	30	6	0	17	8	826

6.3 Interpretation

Anticipated results when recognizing a training set



A first validation of a knowledge consists of recognizing the learned patterns.

Expected results are a vast majority of Correct Identification (ID_correct) and a few Correct Uncertainties (UNC_correct).

You do NOT want any Identification Incorrect (ID_incorrect). Such occurrences are warning about the following errors:

- Some Ground Truth categories might be erroneous
- Some vectors are incorrect
- Vectors do not all represent the same dimension

In KNN mode, it is possible to observe incorrect recognition if a pattern falls within the influence field of the correct neuron, but is at a closer distance from another neuron with a different category.

Anticipated results when recognizing a testing set

The true accuracy of a knowledge is rated by observing its classification of examples never seen before. It is realistic to expect false classifications at first, but NeuroMem can easily correct these inaccuracies by learning new examples.

Multiple iterations between learning and validation might be necessary to obtain a robust knowledge which can adapt to the variations of your objects.

An incorrect classification may be due to one of the following factors:

- Insufficient training
- Erroneous training

ID_Incorrect must be minimized. The remedy consists of turning them into UNC (correct or incorrect) so they can be processed differently. For example, by triggering the aggregation of another feature vector extracted from the same object. Cases of uncertainty are generated by re-learning the training set from scratch but with a higher value of the MINIF.

6.4 Remark about speed performances

The Classify function not only uses the neurons to recognize patterns but also generates a wealth of information for diagnostics purpose such as the retrieval and storage of all the attributes of the top K firing neurons, the comparison with GT categories and more. This data is valuable for diagnostics purposes, but obviously implies some computing overhead.

Please note that this overhead will not exist in a standard classification operation where patterns are broadcasted to the neurons and only the necessary attributes are read and formatted.

7 The Results

7.1 Result format

The results generated by the Classify command are appended to the input dataset in the format described below. The value K is defined in the Preferences menu (default is 3) and determines the length of the output record.

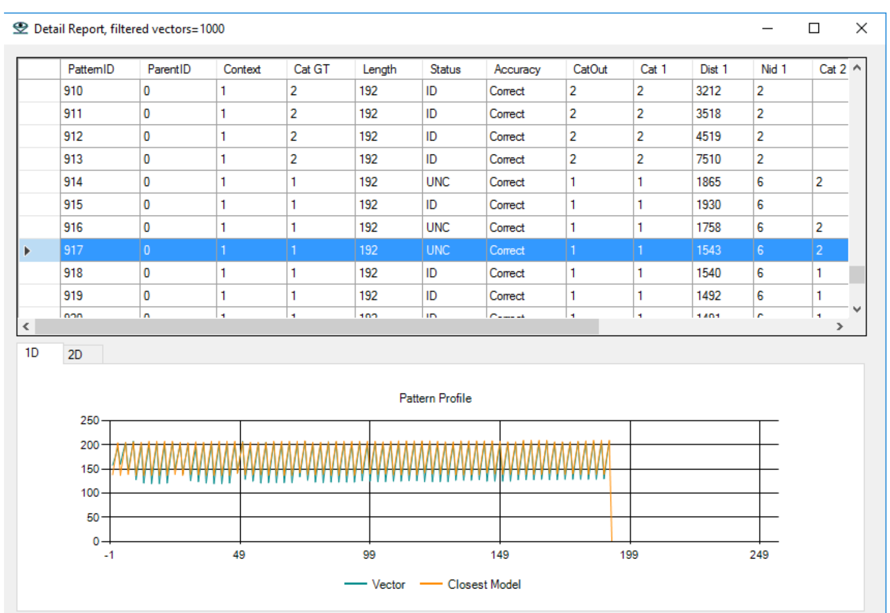
Results can be saved along with the Input data using the Save Data & Results command, or solely using the Save Results Only command.

Status code	Accuracy code	CatOut	Cat 1	Dist 1	Nid1	Cat2	Dist2	Nid2	…	CatK	DistK	NidK
UNK=unknown ID=identified UNC=uncertain	Correct Incorrect N/A	Global response of the firing neurons	Response of the 1st firing neuron including its distance, category and identifier.			Response of the 2nd firing neuron				Response of the Kth firing neuron		
		Choice of BestMatch or Dominant category	In KNN mode, there are always K firing neurons In RBF mode, K is a maximum applicable									

7.2 Results in detail

The input vectors can be studied in detail using the View/Details menu.

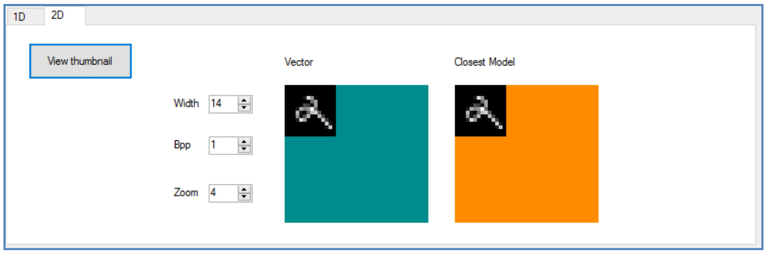
The prompted panel looks as follows and the plot profile is updated each time you click at a new row in the table. The 1D plot reports the input vector and, if applicable, the model of the neuron holding the closest match. The index of this neuron is displayed in the column nid1. If you click on a row where dist1=0, the vector and its best match have the same profile. Otherwise you can observe difference.



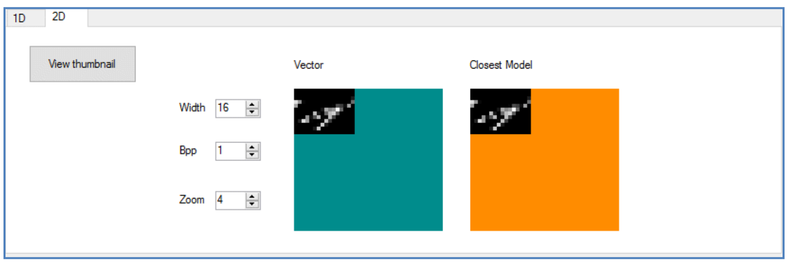
7.3 Optional 2D View

If the vectors represent sequences of pixel values extracted from a W*H patch, they can be displayed as images on the 2D tab.

This implies that you know how the width of the image W an also if each byte of a vector is a single intensity value (bpp=1), or if they should be read per triplet representing a red, green and blue intensity of a same pixel (bpp=3).

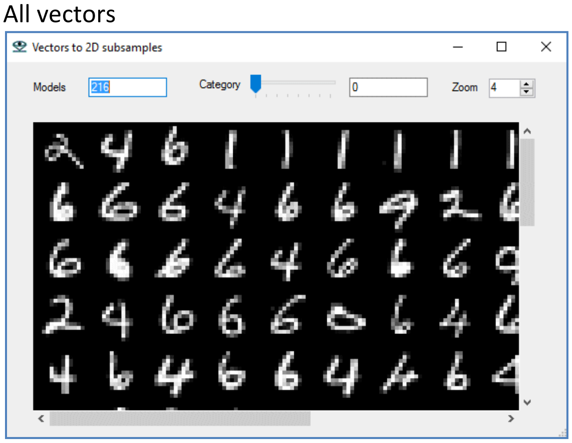


Note that if you do not have the proper values Width and Bpp, the images will look distorted as illustrated below:



7.4 Thumbnails of vectors as images

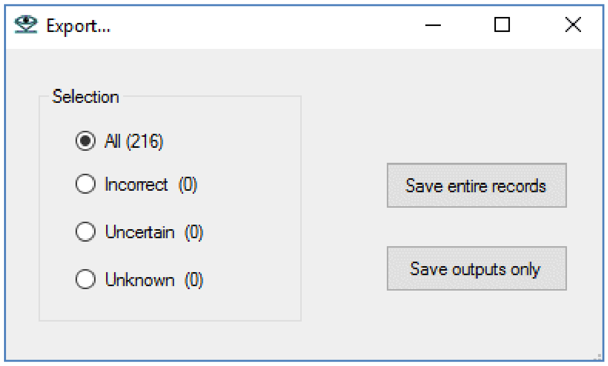
All vectors



Optional filter by category (in this example category 2 has been assigned to the digit "1")

7.5 Saving results

The results can be exported in two formats and for the entire set of vectors or with the filters listed in the panel below:



Entire record

"Entire record" means the assembly of the input format + results format:
PatternID , Descriptor , Context , Ground Truth Category , Vector length L , Vector(0) , Vector(1) , … , Vector(L) , Status code , Accuracy code , Cat 1 , Dist 1 , Nid1 , Cat2 , Dist2 , Nid2 , … , CatK , DistK , NidK

Outpus only

Outputs only means results format only:
PatternID , Descriptor , Context , Ground Truth Category , Status code , Accuracy code , Cat 1 , Dist 1 , Nid1 , Cat2 , Dist2 , Nid2 , … , CatK , DistK , NidK

8 The Knowledge

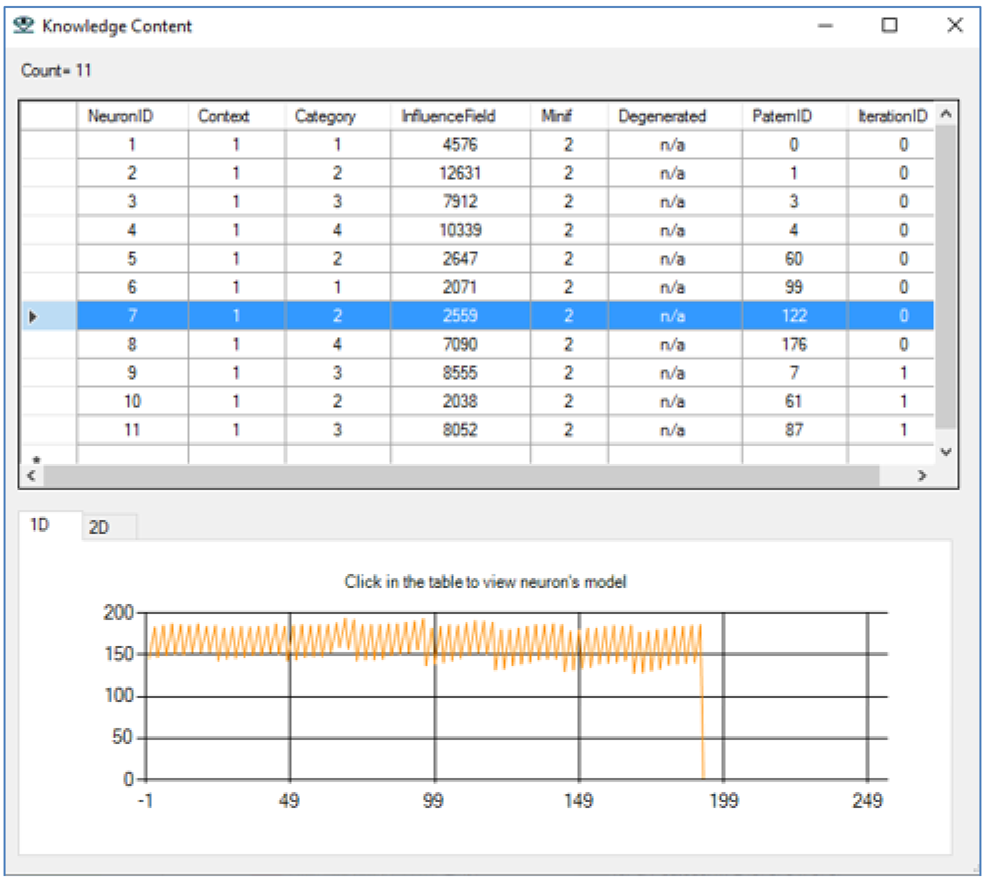
8.1 Content in detail

Reviewing the content of the neurons can help understand and track why a knowledge does not perform according to expectations. Sometimes it may reveal the existence of erroneous Ground truth categories, etc.

This panel is accessible through the View menu and reports information about the committed neurons belonging to the selected context.

The top table lists common registers of the neurons such as the context, influence field, minimum influence field, and if the neuron is degenerated or not. It also lists which input vector triggered the commitment of the neuron and at which iteration.

The bottom plot shows the model stored in the memory of the neuron highlighted in the table.

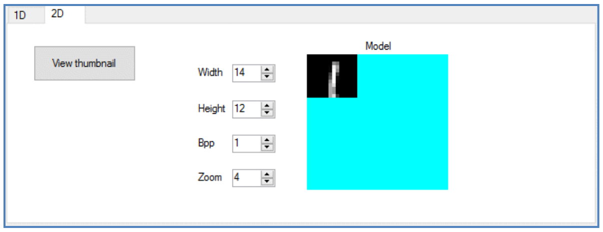


NeuronID	Context	Category	InfluenceField	MinIF	Degenerated	PatternID	IterationID
1	1	1	4576	2	n/a	0	0
2	1	2	12631	2	n/a	1	0
3	1	3	7912	2	n/a	3	0
4	1	4	10339	2	n/a	4	0
5	1	2	2647	2	n/a	60	0
6	1	1	2071	2	n/a	99	0
7	1	2	2559	2	n/a	122	0
8	1	4	7090	2	n/a	176	0
9	1	3	8555	2	n/a	7	1
10	1	2	2038	2	n/a	61	1
11	1	3	8052	2	n/a	87	1

8.2 Optional 2D View

If the models represent sequences of pixel values extracted from a W*H patch, they can be displayed as images on the 2D tab.

This implies that you know how the width of the image W an also if each byte of a vector is a single intensity value (bpp=1), or if they should be read per triplet representing a red, green and blue intensity of a same pixel (bpp=3).



Note that if you do not have the proper values Width and Bpp, the images will look distorted as illustrated below:

8.3 Thumbnails of models as images

All models

Optional filter by category (in this example category 2 has been assigned to the digit "1")

8.4 Knowledge for distribution

A validated knowledge can be saved for export to other hardware platforms featuring a NeuroMem network and programmed to generate the same type of vectors as the ones used for the training of the knowledge.

The program lets you save the knowledge as a binary file or as a comma delimited file format.