

# NeuroMem<sup>®</sup> Knowledge Builder

*Knowledge Building and Pattern Classification  
with a NeuroMem<sup>®</sup> neural network*



**GENERAL VISION**

Version 4.1

Rev 05/30/2019

## CONTENTS

2	Introduction.....	4
2.1	What can I do with NeuroMem Knowledge Builder? .....	4
2.2	Complete Machine Learning Framework.....	5
2.3	Simple User Interface.....	5
2.4	Definitions .....	6
2.4.1	Accuracy code .....	6
2.4.2	Status code .....	6
3	Prepare your datasets .....	6
3.1	Survey your datasets.....	7
4	Learning .....	7
4.1	Learning diagnostics at a glance .....	8
4.2	Check learning curve .....	8
4.3	Observe the learning compression factor per category.....	9
4.4	Knowledge profile in details.....	9
4.5	Validate the knowledge .....	10
4.6	Learning Settings.....	12
4.6.1	Maximum Influence Field .....	12
4.6.2	Minimum Influence Field .....	13
4.6.3	L1 versus LSup norm .....	13
5	Review knowledge.....	13
6	Classify a new testing set and observe .....	15
6.1	Results at a glance.....	15
6.2	Change consolidation rules .....	15
6.2.1	K best responses .....	15
6.2.2	Single Category Out .....	16
6.3	Details per category .....	16
6.4	Details per vector .....	17
6.5	Corrective Actions and Improvements.....	17
7	Change between RBF and KNN.....	17
7.1	RBF .....	17

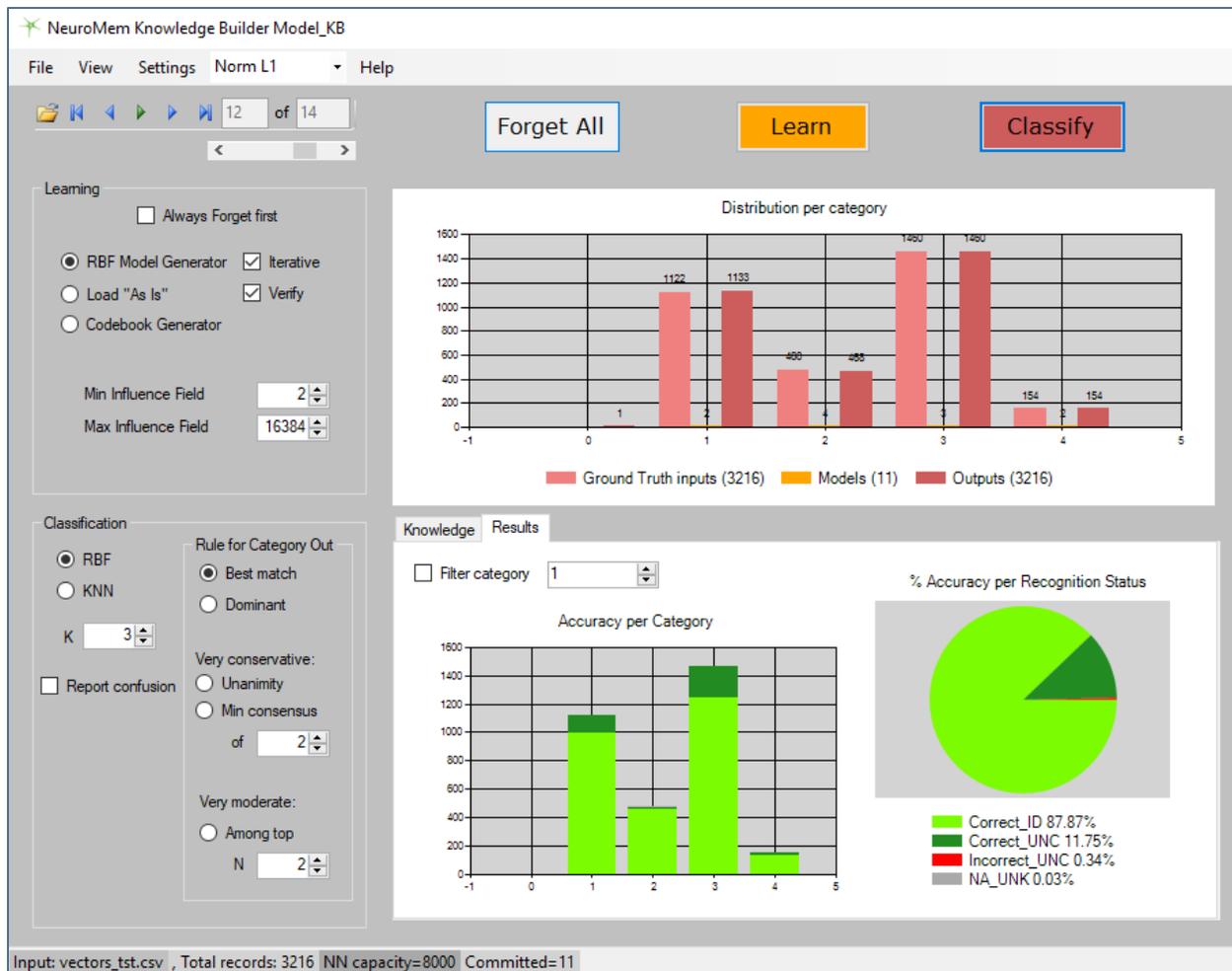
7.2	KNN .....	18
7.2.1	The Confusion Matrix .....	18
8	Load testing set and observe .....	20
9	export results.....	20
9.1	Output format .....	21
10	Save and export the knowledge .....	21
11	Timings .....	22
12	Batch Learning and Classification .....	23
13	What is New?.....	24
13.1	Version 4.0 .....	24

## 2 INTRODUCTION

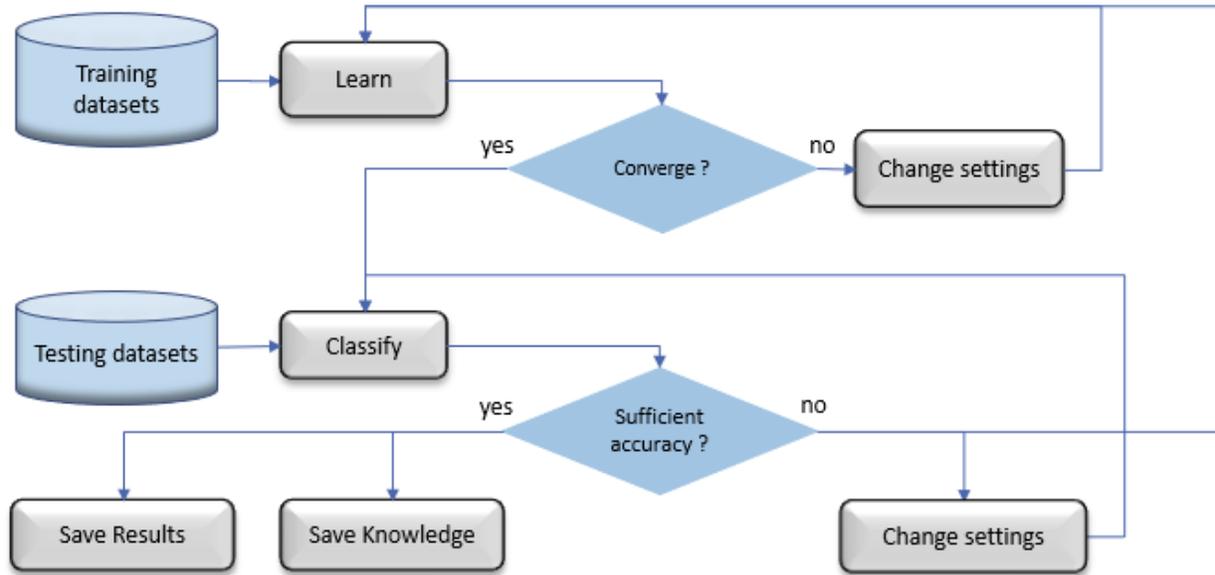
NeuroMem Knowledge Builder is a simple framework to train and validate a NeuroMem network to classify your datasets.

### 2.1 What can I do with NeuroMem Knowledge Builder?

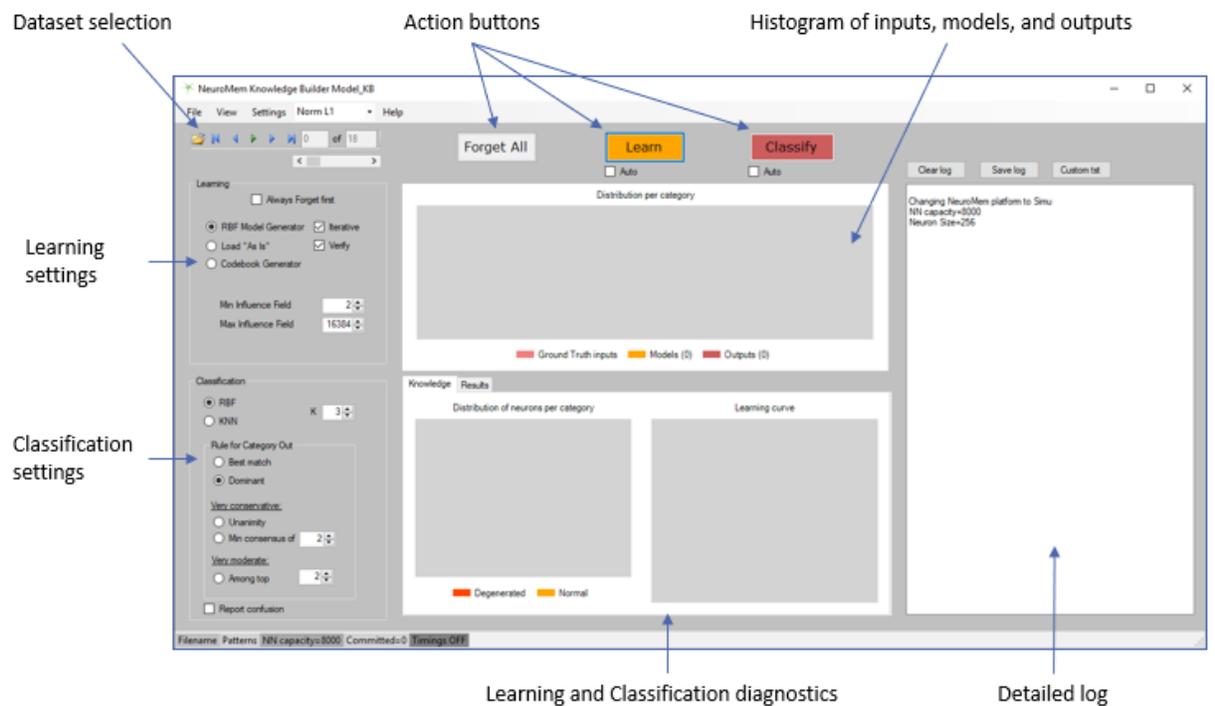
- Import dataset saved in text or csv format
- Experiment with the learning of your training sets
- Compare convergence, data reduction, categories subject to confusion
- Experiment with the classifications of your testing sets
- Compare RBF or K-NN classifications, K value and consolidation rules to handle cases of uncertainty
- Qualify accuracy per category and vector, confusion matrix, etc.
- Export classification results
- Review, Save and Load the neurons' content (aka Knowledge)
- Interface to NeuroMem USB devices such as the NeuroMem dongle and NeuroShield
- In the absence of a NeuroMem hardware, simulate a cycle-accurate NeuroMem network of 8000 neurons



## 2.2 Complete Machine Learning Framework



## 2.3 Simple User Interface



## 2.4 Definitions

### 2.4.1 Accuracy code

The Accuracy is based on the comparison of the Category Out (CatOut) and the Ground Truth category (CatGT). The Category Out is a function of the selected consolidation rule (best match, dominant, unanimity, minimum consensus).

- Correct      CatOut == CatGT
- Incorrect    CatOut != CatGT
- NA            Case of a non-existent Ground Truth Category

### 2.4.2 Status code

The Status code is based on the responses of the top N firing neurons. In the case of a KNN classification, the value N is equal to the select K value. In the case of an RBF classification, N can be less than K.

- ID, Identified All firing neurons recognize the same category
- UNC, Uncertain      All firing neurons do not recognize the same category
- UNK, Unknown      CatOut is equal to zero

## 3 PREPARE YOUR DATASETS

The application can load files saved in comma delimited format and complying with the simple format described below. Examples of datasets are supplied with the application.

Because the NeuroMem neurons each have a memory capable of holding patterns of 256 bytes, your pattern data P must be converted to fit in an array of dimension L with  $L < 256$  and with values ranging between [0, 255].

PatternID (optional content)	ParentID (optional content)	Context (*) (optional content)	CatGT	Length	P[0]	P[1]	...	P[L]
Index of the pattern in the input list	User-defined index (a file name, XY position, time stamp, etc.)	Value identifying the type of feature encrypted in the vector.	Ground Truth Category value assigned to the vector	Number of components L in the vector	L values of the vector components			
Default= automatic increment of 1, starting at 0	Default =0 Range=[0, 65535]	Range=[1, 127] Default = 1	Range=[1,32768] 0 = "background" 0xFFFF = "unspecified"	Default Range=[1,256] for all NeuroMem platform, except Intel	Values ranging between [0,255]			

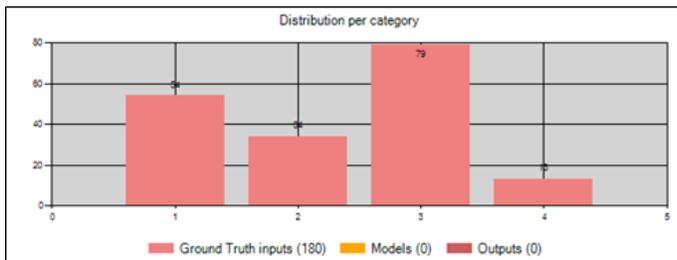
Range=[0, 65535]				QuarkSE/Curie = [1,128]	
------------------	--	--	--	-------------------------	--

### 3.1 Survey your datasets

Whenever a new dataset is loaded, the distribution of the input vectors per Ground Truth category is plotted in the main panel.

If the dataset is intended for a training, it might be of interest to notice if all categories are evenly represented, or not.

If a knowledge already resides in the NeuroMem network, the distribution of its models per category is also plotted (see the Learn paragraph).



Inputs:  
Vectors, with associated  
Ground Truth categories

## 4 LEARNING

Click the Learn button to submit all the input vectors and their Ground Truth categories to the NeuroMem neurons which will autonomously model the decision space storing only the relevant models into their memories and adjusting autonomously their influence fields to build a consistent knowledge.



Inputs:  
Vectors, with associated  
Ground Truth categories

**Learn**

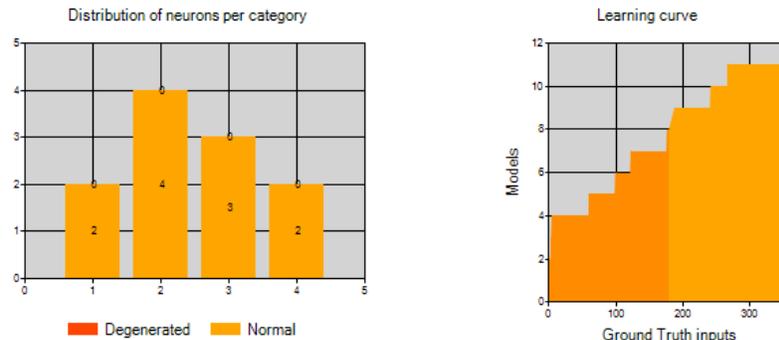


Models:  
Inputs retained by the neurons  
to describe the decision space

## 4.1 Learning diagnostics at a glance

Upon learning, the Knowledge Tab is updated and reports the distribution of the committed neurons per category and the learning curve.

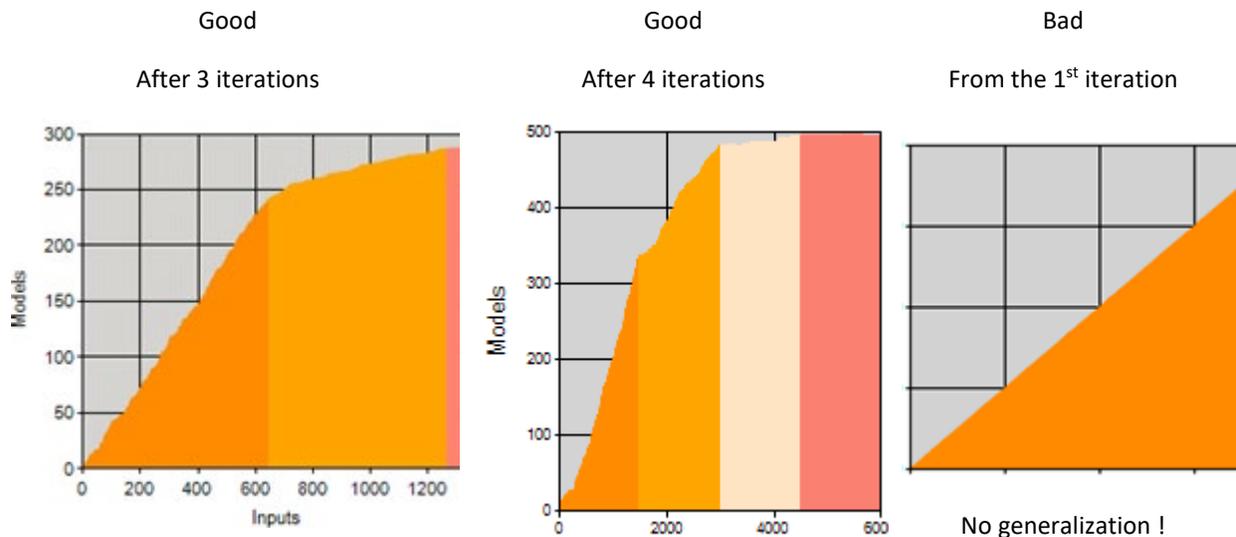
- 1) The reporting of degenerated neurons in the histogram is a flag that some of the categories in the learned dataset have overlap in the decision space and you can expect uncertainties between them.
- 2) The learning curve reports how many neurons are committed as input patterns are learned. This curve should be asymptotic (see next paragraph). Provided that the "Iterative" checkbox is marked, the Number of colors in the plot indicates how many iterations were necessary until the learning converges and no more neurons need to be committed to model the decision space.



## 4.2 Check learning curve

Verify at a glance that the learning curve is correct. An asymptotic curve reflects that the neurons can model your training set while over-generalizing and converging.

- X-axis: learned vectors
- Y-axis: retained vectors to model the knowledge

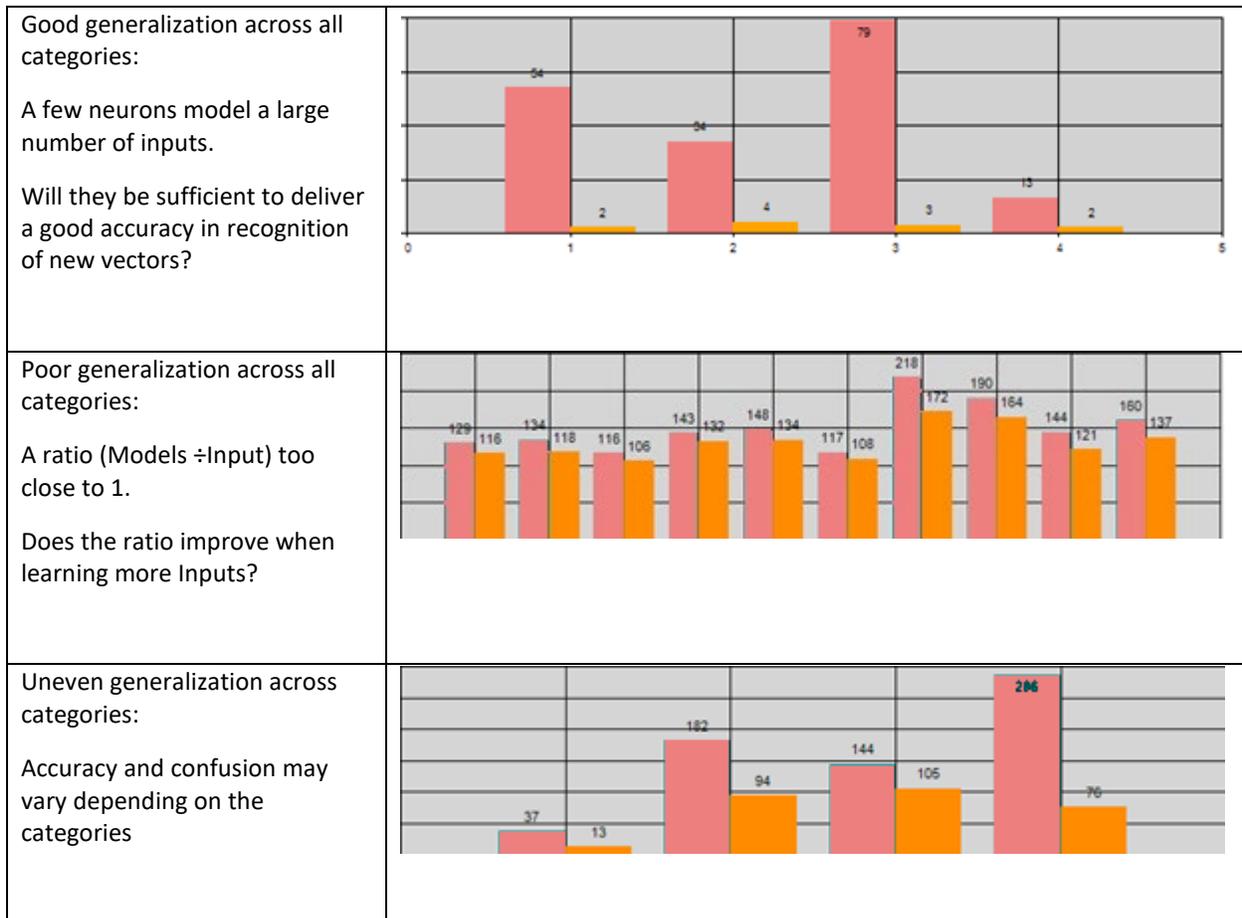


### 4.3 Observe the learning compression factor per category

The bars of the histogram of the Models should be significantly lower than the ones of the Ground truth categories. This means that the neurons are capable of generalization and validates the relevancy of the input vectors to model the GT categories.

The ratio between the number of input patterns (green) and the number of models retained by the neurons (orange) is an indicator of how well the neurons can model a specific category given the learned training set.

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



In addition to the histogram of the committed neurons per category, the text log reports statistics about their maximum dimensions of the AIFs per category, the existence of degenerated neurons per category of any, and the level of generalization/compression per category.

### 4.4 Knowledge profile in details

The knowledge is the content of the neurons including the model stored in their memories, their associated Ground Truth categories and their influence fields which they autonomously adjust during learning.

The Log panel reports the detailed distribution of the neurons per category in absolute and percentage values. Note that this report is function of the values of the Min and Max Influence fields defined in the Learning Settings.

Learn_RBF, Minif=2, Maxif=16384													
Neurons	Total	Cat0	Cat1	Cat2	Cat3	Cat4	Cat5	Cat6	Cat7	Cat8	Cat9	Cat10	Cat11
Percentage	100	0	25.64	7.69	2.56	10.26	7.69	2.56	2.56	10.26	5.13	5.13	20.51
Count	39	0	10	3	1	4	3	1	1	4	2	2	8
Max AIFs	6005	0	4904	5458	5254	5031	5133	6005	5133	5205	5445	5637	5404
Neurons/Inputs	8.88	n/a	11.76	5.77	3.33	16.67	10.34	3.03	14.29	11.11	5.41	3.64	15.69

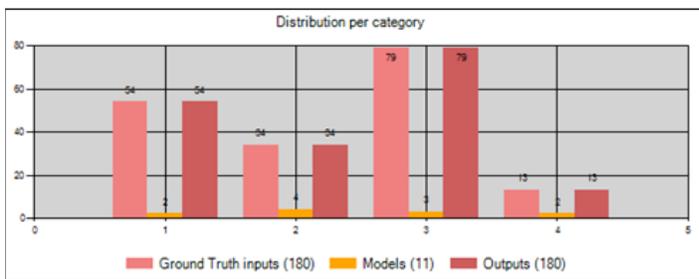
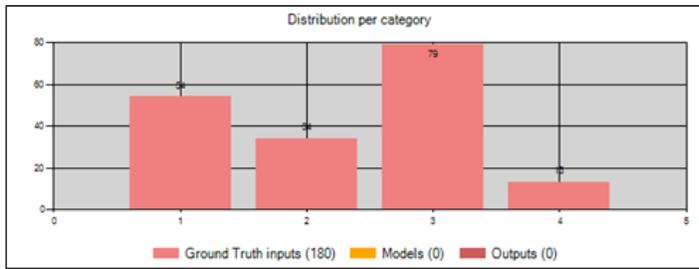
The Max AIFs and the ratio Neurons/Inputs can give an indication of the ease or complexity to model a particular category. For example, if the Max AIF of a category p is much smaller than the other Max AIFs of the other categories and at the same time the Neurons/Inputs ratio of this category p is much higher than the other ratios, we can assume that p is represented with a variety of models which are not easy to aggregate together and the recognition of vectors belonging to the category p might be more difficult than the other categories.

#### 4.5 [Validate the knowledge](#)

After a learning operation, it is important to verify before going any further that the presently committed neurons recognize correctly the dataset which was just learned. Any other result indicates that corrective actions might be needed such as:

- Verify accuracy of the GT categories in the dataset
- Learn additional training sets
- Re-consider the relevancy of the input vectors to properly model and discriminate the intended population of objects or events

The classification of the training dataset is executed automatically if the “Verify” box displayed under the Learning Settings is checked. Otherwise, simply click the Classify button.



Inputs:  
Vectors, with associated  
Ground Truth categories

**Learn**



Models:  
Inputs retained by the neurons  
to describe the decision space

**Classify**



Outputs:  
Vectors and their recognized  
categories

- Accuracy should be 100% since the neurons have learned the entire training set.
- A certain level of uncertainty is acceptable.
- Unknown are unacceptable

Ideal	Acceptable	Exceptional (**)
<p>% Accuracy per Recognition Status</p> <p>Correct_ID 100.00%</p>		<p>% Accuracy per Recognition Status</p> <p>Correct_ID 87.87% Correct_UNC 11.75% Incorrect_UNC 0.34% NA_UNK 0.03%</p>

The histograms of the Ground Truth and Output categories should be identical. If not, go to Corrective Actions.

If uncertainties exist in moderate amount, you can identify the categories which are close to one another and may introduce confusion in future classifications:

- Check "Confusion matrix"
- Open the View\Details\Uncertain and observe the recognized categories

If uncertainties exist in significant amount, consider corrective actions:

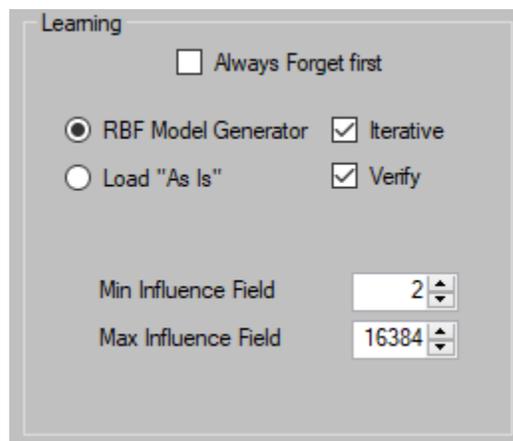
- Verify accuracy of the GT categories in the dataset
- Learn additional training set
- Re-consider the relevancy of the dataset (i.e. feature vectors) to classify and discriminate the intended population of objects or events

(\*\*) Unknown recognition of vectors from the training set should not occur except if the consolidation rule requires a minimum number of consensus.

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.

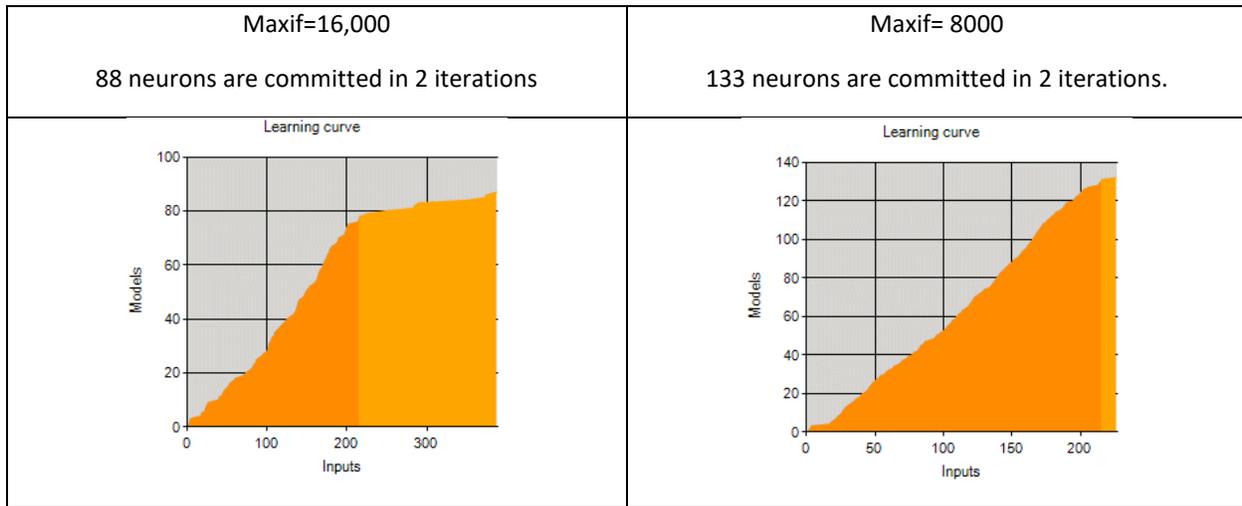
## 4.6 [Learning Settings](#)

NeuroMem Knowledge Builder makes it easy to use a same training set to build a multiplicity of knowledge bases by changing the few input parameters described below. The qualification of these knowledge bases can be quickly estimated by classifying new testing sets and comparing their diagnostics reports in term of throughput and accuracy.



### 4.6.1 Maximum Influence Field

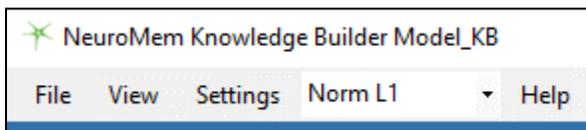
- The higher, the more liberalism and over-generalization. Tendency to model the decision space with few neurons with large influence fields.
- The lower the more conservatism. Tendency to model the decision space with more neurons, but with smaller influence fields.



#### 4.6.2 Minimum Influence Field

- The higher, the more uncertainties since the neurons are prevented to shrink their influence field under this Min value
- Increasing the Minif helps separate the categories with and without confusion, flag outliers and reveal non discriminating features
- The categories with possible confusion may later be handled differently by introducing the use of a different datasets representing different feature in order to waive uncertainties

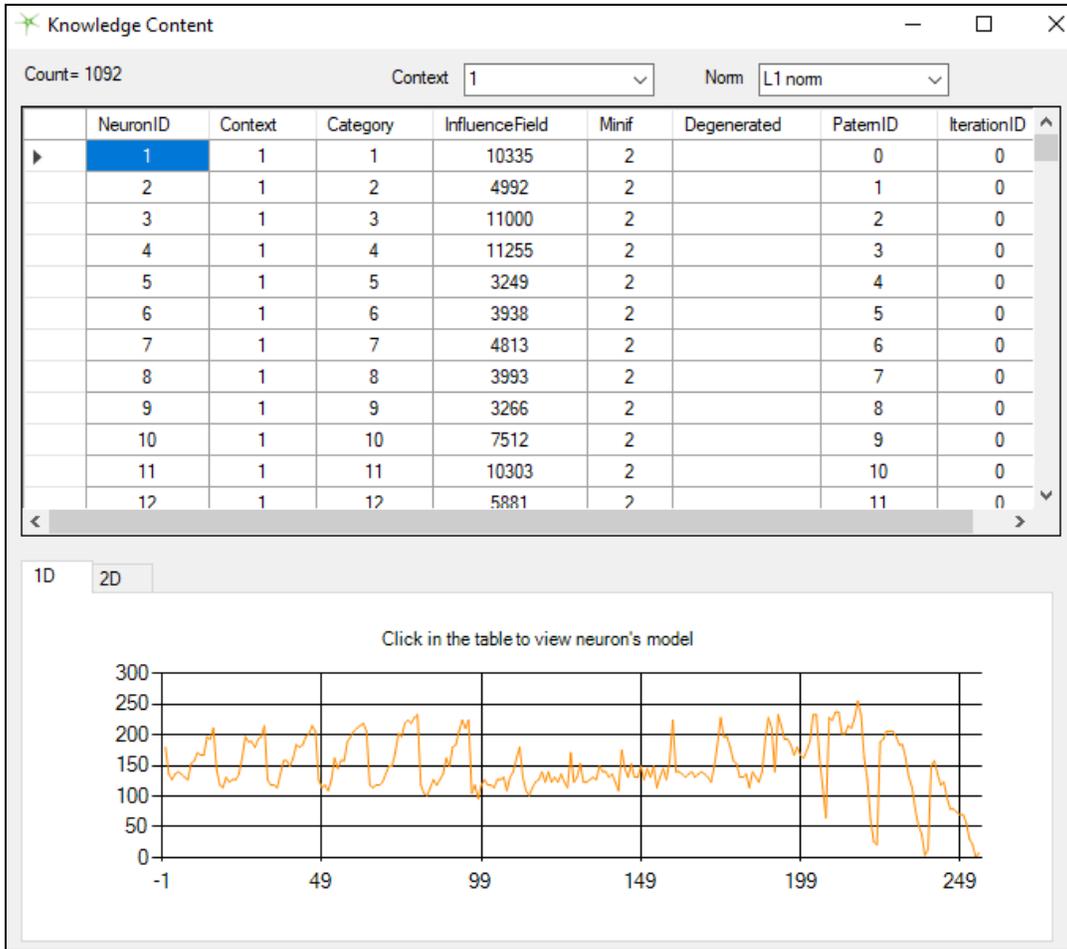
#### 4.6.3 L1 versus LSup norm



- The norm defines how to calculate the distance between the input patterns and the models stored in the neurons.

## 5 REVIEW KNOWLEDGE

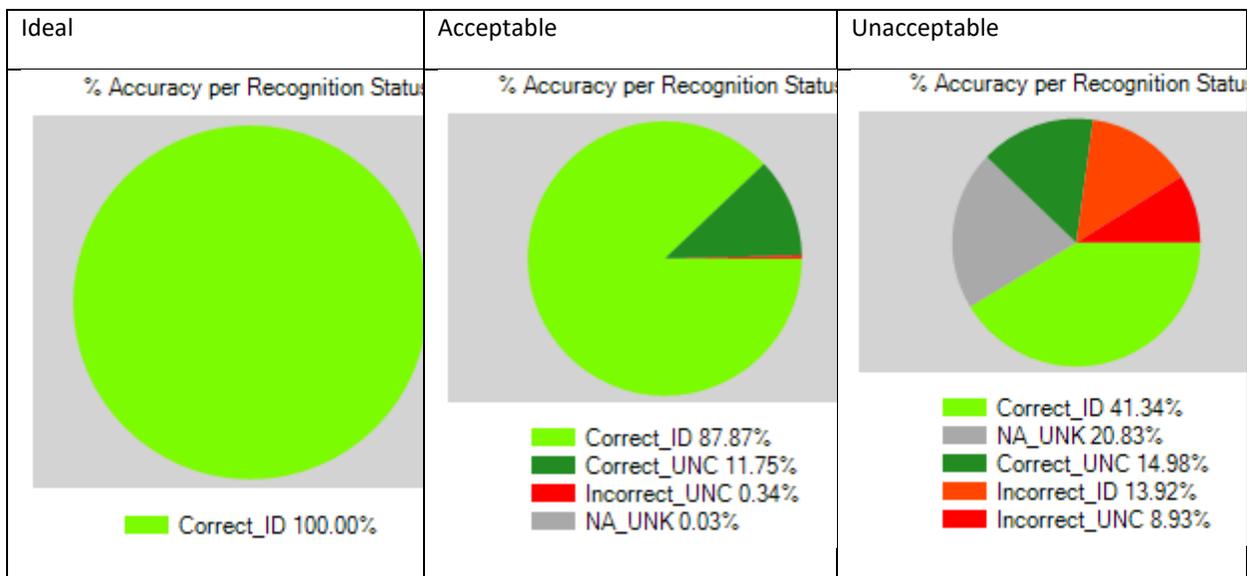
The knowledge is nothing more than the content of the neurons. It can be built by learning a single or multiple training sets.



## 6 CLASSIFY A NEW TESTING SET AND OBSERVE

### 6.1 Results at a glance

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.



- The ideal case is 100% Correct with 100% Identified.
- The worst case is to have any Identified Incorrect.
- A remedy to minimize the ID\_Incorrect consists of turning them into UNC (correct or incorrect) so they can be isolated and processed differently.

### 6.2 Change consolidation rules

Multiple neurons can recognize a same input vector and agree with its classification, or not.

In case of disagreement between firing neurons, a consolidation rule is applied to output a single category value, Cat Out.

Cat 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.

#### 6.2.1 K best responses

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. Typical values are 1, 3 or 5.

### 6.2.2 Single Category Out

Out of the K best responses, the global response or category Out is defined by one of the following consolidation rules:

Rule for Category Out

Best match

Dominant

Very conservative:

Unanimity

Min consensus of

Very moderate:

Among top

- **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 if it has a minimum count; category Unknown otherwise
- **Among top N:** This rule is purely academic and not practical in real life. It was recently added to fairly compare the performances of a NeuroMem network with Deep Learning/CNN benchmarks. It actually discards the value of the Category Out and assumes that the classification of the input vector is correct is one of the top N firing neurons returns the Ground Truth category regardless of its position in the firing list.

	Cat GT	Length	Status	Accuracy	CatOut	Cat 1	Dist 1	Nid 1	Cat 2	Dist 2	Nid 2	Cat 3	Dist
	1	192	UNC	Incorrect	2	1	1017	6	2	1740	10	2	254
	1	192	UNC	Incorrect	2	1	1475	6	2	1986	10	2	125
	1	192	UNC	Incorrect	2	1	479	6	2	1966	10	2	235
	1	192	UNC	Incorrect	2	1	435	6	2	2018	10	2	238
	1	192	UNC	Incorrect	2	1	337	6	2	2416	7	2	260
	1	192	UNC	Incorrect	2	1	302	6	2	2485	7	2	264
	2	192	UNC	Incorrect	1	1	2003	6	2	11519	2		
	1	192	UNC	Incorrect	2	1	1348	6	2	1857	10	2	123
	1	192	UNC	Incorrect	2	1	495	6	2	1746	10	2	254
	2	192	UNC	Incorrect	1	1	2016	6	2	11520	2		

In the above example, the selected vector has a Ground Truth category of 1 and the first 3 firing neurons recognize respectively the categories 1, 2 and 2

With Dominant Category rule, CatOut=2, → Accuracy Status = Incorrect

With Closest Category rule, CatOut=1 → Accuracy Status = Correct

### 6.3 [Details per category](#)

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%
> ID	65.13%	0%	8.25%	9.39%	6.40%	6.45%	5.63%	5.66%	7.92%	5.69%	4.77%	4.97%
> UNC	15.23%	0%	1.52%	2.46%	0.79%	1.78%	1.24%	1.22%	1.24%	2.31%	0.46%	2.21%
Incorrect	10.94%	0%	0.58%	0.23%	1.14%	1.40%	1.45%	1.24%	0.81%	1.24%	0.74%	2.11%
> ID	5.43%	0%	0.30%	0.05%	0.69%	0.53%	0.71%	0.58%	0.51%	0.66%	0.56%	0.84%
> UNC	5.51%	0%	0.28%	0.18%	0.46%	0.86%	0.74%	0.66%	0.30%	0.58%	0.18%	1.27%
NA	8.71%	0%	0.66%	0.18%	1.83%	0.81%	0.86%	1.04%	0.46%	0.81%	1.27%	0.79%
> UNK	8.71%	0%	0.66%	0.18%	1.83%	0.81%	0.86%	1.04%	0.46%	0.81%	1.27%	0.79%

## 6.4 Details per vector

Global response 1<sup>st</sup> firing neuron 2<sup>nd</sup> firing neuron → Kth firing neuron

	Cat GT	Length	Status	Accuracy	CatOut	Cat 1	Dist 1	Nid 1	Cat 2	Dist 2	Nid 2	Cat 3	Dist
	1	192	UNC	Incorrect	2	1	1017	6	2	1740	10	2	254
	1	192	UNC	Incorrect	2	1	1475	6	2	1986	10	2	125
	1	192	UNC	Incorrect	2	1	479	6	2	1966	10	2	235
	1	192	UNC	Incorrect	2	1	435	6	2	2018	10	2	238
	1	192	UNC	Incorrect	2	1	337	6	2	2416	7	2	260
	1	192	UNC	Incorrect	2	1	302	6	2	2485	7	2	264
	2	192	UNC	Incorrect	1	1	2003	6	2	11519	2		
	1	192	UNC	Incorrect	2	1	1348	6	2	1857	10	2	123
	1	192	UNC	Incorrect	2	1	495	6	2	1746	10	2	254
	2	192	UNC	Incorrect	1	1	2016	6	2	11520	2		

## 6.5 Corrective Actions and Improvements

- Try different values for K and different consolidation rules (see below)
- Verify accuracy of the GT categories in the dataset
- Learn additional training set
- Re-consider the relevancy of the dataset (i.e. feature vectors) to classify and discriminate the intended population of objects or events

## 7 CHANGE BETWEEN RBF AND KNN

### 7.1 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			
Correct	1	1	1401	6	1	3700	1			

## 7.2 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
Correct	1	1	2100	6	2	2317	10	2	2850	7

### 7.2.1 The Confusion Matrix

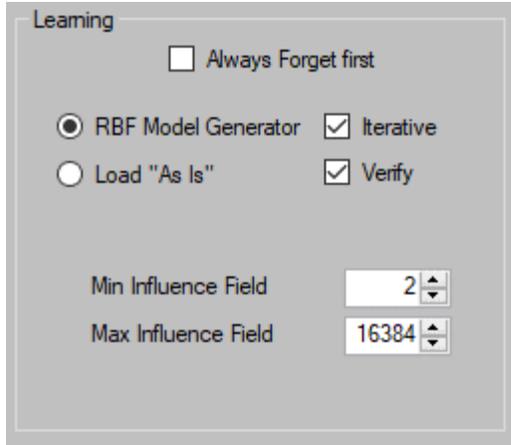
Each row corresponds to a Ground Truth category and the columns report the counts per recognized category. For example Confusion[A,B]=K means that K patterns assigned with the Ground Truth category A are actually recognized a category B. The numbers along the diagonal (Confusion[A,A]) are the numbers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

## 8 LOAD TESTING SET AND OBSERVE

For applications which do not require RBF learning, it is possible to load the neurons with the training set “as is”. In such case, the neurons behave as dummy memories storing all input patterns and associated categories without any processing logic.



Learning

Always Forget first

RBF Model Generator  Iterative

Load "As Is"  Verify

Min Influence Field

Max Influence Field

If your training set has N inputs, loading it will commit N neurons.

If your training set includes redundancies, the neurons will not perform any data reduction nor de-duplication.

If your training set includes inconsistencies, the neurons will not flag them with their “degenerated” status.

All the committed neurons will have a same influence field equal to the value of the Maximum Influence Field at the time of the loading.

## 9 EXPORT RESULTS

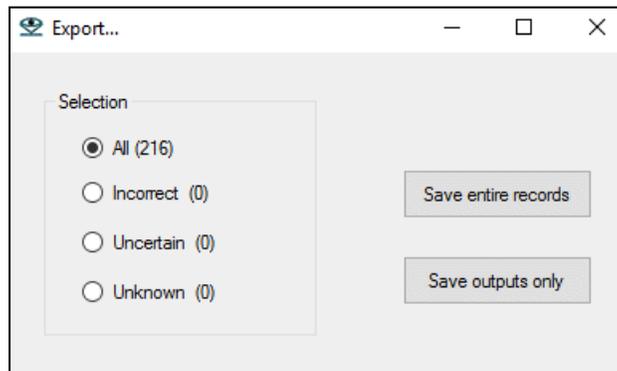
The results generated by the Classify command are appended to the input dataset in the format described below.

## 9.1 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.

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  Choice of BestMatch or Dominant category	Response of the 1st firing neuron including its distance, category and identifier.			Response of the 2nd firing neuron				Response of the K <sup>th</sup> firing neuron		
			In KNN mode, there are always K firing neurons In RBF mode, K is a maximum applicable									

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



## 10 SAVE AND EXPORT THE KNOWLEDGE

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.

## 11 TIMINGS

The latencies of the instructions executed by the NeuroMem network can be reported in the status bar shown at the bottom of the main panel.

Such reporting adds an overhead to the real overall latencies of execution, so it can be turned ON/OFF under the Options menu.

The timings are calculated using the exact number of clock cycles spent in the NeuroMem chips times a clock cycle of 55 ns or 18 Mhz which is the recommended clock for a NeuroMem network composed of multiple NM500 chips.

These timings do not include the time spent to transmit the USB or SPI packets back and forth between the host and the platform hosting the NeuroMem chips.

## 12 BATCH LEARNING AND CLASSIFICATION

NeuroMem Knowledge Builder can learn an entire folder of training datasets and classify an entire folder of testing datasets.

- Load a first image from the desired folder
- Click Auto under the Learn or the Classify button
- The current file is immediately processed, and you can verify that your settings are correct by observing the detailed log to the right
- When ready, click the Play button in the navigation bar at the top left of the main panel.
- Histograms and plots are disabled except for the learning curve which is cumulated across the learned datasets
- The results of the classification are appended to the right in the input file

## 13 WHAT IS NEW?

### 13.1 Version 4.0

- Expansion of the [consolidation rules](#)
- [Addition of the batch process](#)
- The path to the last dataset is kept as the last reference
- [Report of the timings](#) spent in the NeuroMem neurons