

# NeuroMem

## RBF Decision Space Mapping

---

Version 4.3  
Revised 01/10/2019



This manual is copyrighted and published by General Vision Inc. All rights reserved.

## Contents

---

1	Introduction to a decision space .....	3
1.1	Example of a 2D decision space .....	3
1.2	Definitions.....	4
1.3	Examples of N-dimensional space .....	4
2	Construction with an RCE/RBF model generator.....	5
3	Recognition with the RCE/RBF classifier .....	6
3.1.1	Three possible classification status.....	7
3.1.2	Multi-level responses.....	7
4	Flexibility of the RCE/RBF model generator .....	8
4.1	Non-linear model generator .....	8
4.1.1	Modulating liberalism by teaching counter examples.....	9
4.1.2	Modulating liberalism with the Maximum Influence Field.....	9
4.2	Shaping the decision space with a Norm.....	10
4.2.1	Norm L1 (Manhattan distance).....	10
4.2.2	Norm L Sup .....	10
4.2.3	Norm L2 (Euclidian distance) .....	10
4.3	Building zones of uncertainty .....	11
4.3.1	Modulating uncertainty zones with the Minimum Influence Field .....	11
4.4	Benefits of Iterative learning .....	11
4.4.1	Incremental Decision Space mapping.....	11
4.4.2	Learning curve and feature relevancy .....	13
5	KNN (K-Nearest Neighbor) classifier .....	14
5.1	Fastest KNN classifier.....	14
5.2	KNN is not a model generator .....	14
5.3	Warning: KNN classification using RBF models.....	15
6	From decision space to knowledge base .....	16

## 1 Introduction to a decision space

Pattern recognition starts with the definition of a decision space suitable to discriminate distinct categories of objects. If the characterization of a population is composed of  $N$  measurements, these  $N$  values ( $X_1, X_2, \dots, X_n$ ) make a feature vector or signature vector. Depending on the application, the  $X_i$  measurements can be unrelated, or on the contrary represent a progression over time, or a variation in space, etc.

Once a certain number of feature vectors are plotted, the decision space starts taking shape and hopefully reveals clusters of objects belonging to distinct categories. In the case of a complex situation, a same category may be modeled by multiple disjoint clusters, or distinct categories can be modeled with intertwined and embedded clusters. The resulting decision space can then be used to over generalize and classify new vectors.

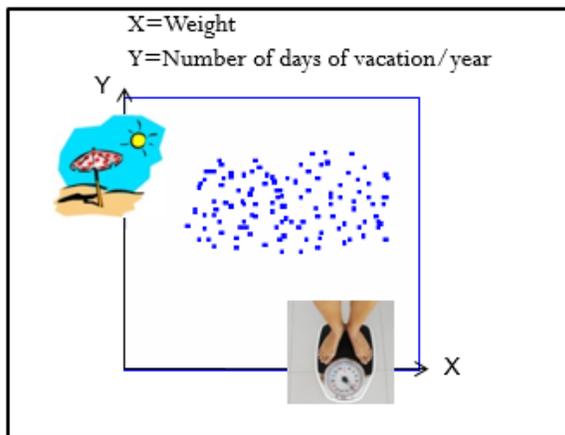
A NeuroMem network is a chain of identical neurons which have their own “genetic” material to learn and recall patterns. They can build a non-linear decision space following a Radial Basis Function (RBF) model generator and more specifically a Restricted Coulomb Energy (RCE) neural network. They can also behave as a simple K-Nearest Neighbor classifier.

This manual will use examples taken in 2D space for the sake of graphical illustrations, but the NeuroMem neurons can build a decision space with up to 256 dimensions.

### 1.1 Example of a 2D decision space

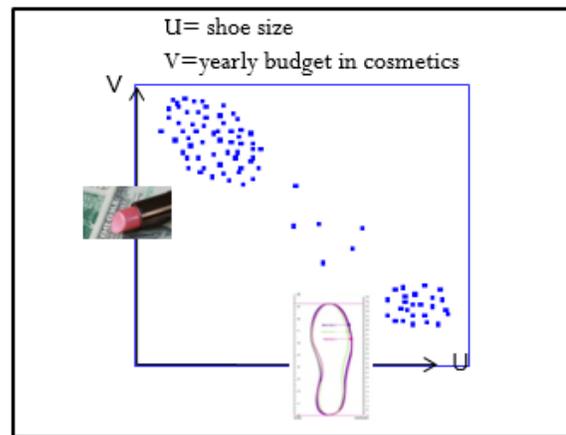
Given pairs of (X,Y) measurements collected over a population of people or objects, their spatial distribution can outline significant clusters of points. These clusters can be associated to categories or classes of objects.

#### Feature Set 1:



No specific cluster can be outlined. The measurements  $X$  and  $Y$  are either irrelevant or insufficient to reveal significant categories within the surveyed population.

#### Feature Set 2:



Two obvious and disjoint clusters appear: (1) persons with shoe size above 10 and spending less than \$800/year in cosmetics; (2) persons with shoe size under 10 and spending over \$1000/year in cosmetics. A few outliers can be observed between the two clusters.

If the intent of the survey is to discriminate males and females, the measurements (U,V) will do a better job than the measurements (X,Y).

## 1.2 Definitions

Lets' take the example of a dataset describing Iris flowers:

Example	Feature 1	Feature 2	Feature 3	Feature 4	Category
	Sepal length	sepal width	petal length	petal width	species
1	6.4	2.8	5.6	2.2	2
2	5.0	2.3	3.3	1.0	1
3	4.9	2.5	4.5	1.7	2
4	4.9	3.1	1.5	0.1	3
5	5.7	3.8	1.7	0.3	3

- The **Features** columns contain characteristics of an example
- The **Category** column (species) is the label to learn or predict; It is naturally a string, but the digital world relies on numeric values. Therefore, it is mapped to a number.
  - 1 = versicolor
  - 2 = virginica
  - 3 = setosa
- A **Feature Vector** is a series of features which can be heterogeneous (like in this example), or on the contrary represent a time or spatial series, etc.

## 1.3 Examples of N-dimensional space

### Dimension 4:

Feature vector = (Wave crest, Wave trough, Wavelength, Wind velocity)

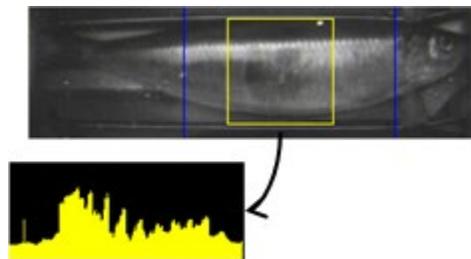
Category = Capillary waves, Chop, Swell, Tsunamis, Seiches



### Dimension 256:

Feature vector = Subsampling of a region of pixels

Category = Accept, Recycle, Reject

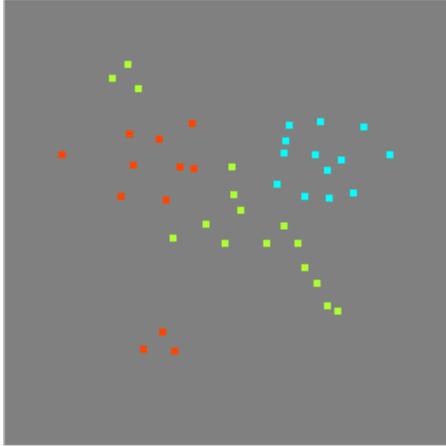


## 2 Construction with an RCE/RBF model generator

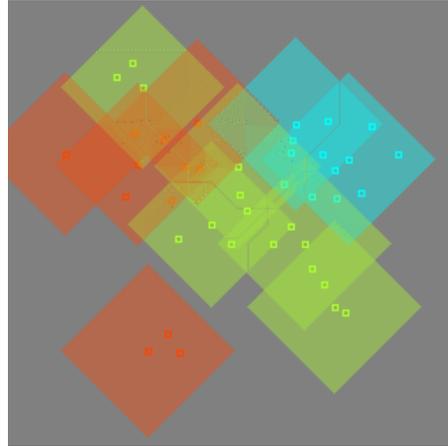
Neural networks are trainable and good at modeling decision spaces with some fuzzy logic. They can generalize what they are taught and therefore recognize patterns which they have never seen before.

The decision space is mapped by teaching a series of examples and labeling them with a category.

Graph 1: Examples of X,Y patterns and their associated categories represented by different colors



Graph 2: Resulting decision space modeled by the neurons by learning the examples

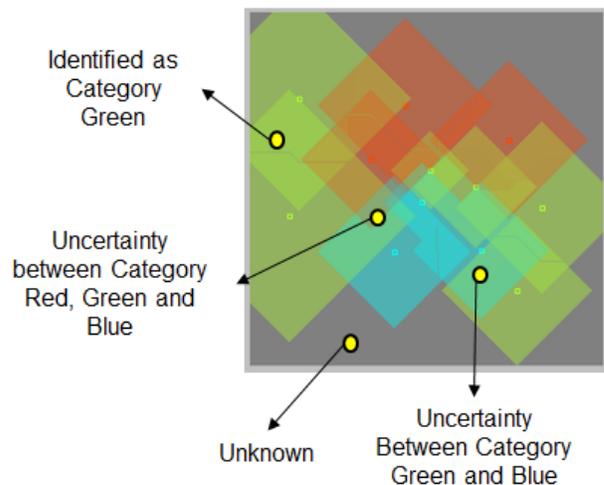


Each time a set  $(X, Y, Cat)$  is presented to the network, the neurons first verify if one of them already recognizes  $(X, Y)$  as belonging to category  $C$ . If this is the case, no action is taken. If, on the contrary,  $(X, Y, Cat)$  is not recognized by any neuron, a new one is created to store  $(X, Y)$  as a reference pattern and  $C$  as the category. In addition, the new neuron inherits an influence field which defines its area of attraction or similarity domain. As examples are taught, the decision space gets built accommodating smaller neurons which model the outliers.

A neuron is represented in the decision space by the point  $(X, Y)$  surrounded by a dotted area with a color associated to  $C$ . This area represents the influence field of the neuron and its shape is determined by the norm used for distance calculation (see next paragraph).

In recognition, broadcasting a new set  $(X, Y)$  will cause some neurons to fire if  $(X, Y)$  falls in their influence field.

- If all firing neurons have the same category, the set  $(X, Y)$  is positively identified.
- If the firing neurons do not have the same category, the set  $(X, Y)$  is recognized but with uncertainty.
- If no neuron fires, the set  $(X, Y)$  is unknown



### 3 Recognition with the RCE/RBF classifier

---

The classification of a vector consists of evaluating if it lies within the influence field of one or more neurons modeling the decision space. The outcome can have three possible classification status: Identified with certainty, Identified with uncertainty, Unknown.

As a result, the RCE/RBF classifier is very powerful since it allows managing uncertainty for a better, more refined diagnostic. It is also especially suited for anomaly or novelty detection where the “unknown” classification is the one of importance.

When a vector is broadcasted to the neural network, all the neurons calculate their distance between the input vector and the prototype stored in their memory. If the distance of a neuron is greater than its influence field, the neuron excludes itself from the list of neurons recognizing the vector. Otherwise it “fires” to indicate that it recognizes “somewhat” the vector. The similarity range is expressed with the distance value. Its dimension is a function of the data type stored in the vector and the norm in use to calculate the distance.

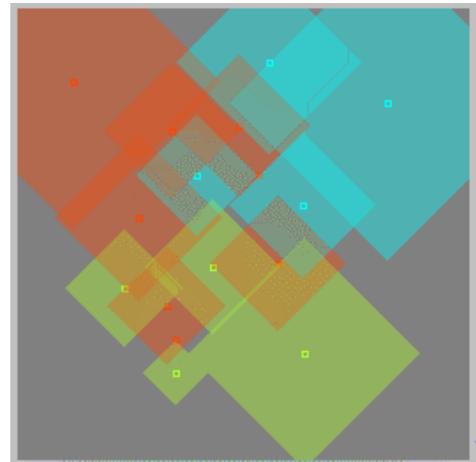
Several neurons can recognize the input vector. The one with the smallest distance value has a prototype in memory which is the closest to the input vector. Also, more than one neuron can fire with the same smallest distance. If they have identical categories, it reinforces the confidence level of the recognition. If they do not have the same category, they point a level of uncertainty in the recognition and potential ambiguities between certain categories. This uncertainty can be further considered by reading the categories recognized by the next firing neurons, that is with the next smallest distance value, etc.

The higher the distance, the lesser the similarity between the prototype and the input vector.

If a neuron has a distance equal to 0, it means that the input vector matches exactly its prototype.

In the example to the right, a 2D decision space is divided into three categories labeled with the colors R, G, B. It has been modeled with 19 neurons: 6 associated to the Red category, 4 to the Green category and 4 to the Blue category. The influence fields of the models have variable size and some are overlapping one another.

It is common that a decision space recognizes most of a population with a high accuracy after a minimal training and a few neurons, but performs poorly on the remaining population. Obviously the NeuroMem non-linear classifier can learn many examples of the corner cases and tune the decision space to recognize them by generating smaller neurons delimiting the edges of the decision space (edges between categories, and edges with the background).



### 3.1.1 Three possible classification status

The recognition of a vector can have three outcomes or classification status:

- Unknown: no neuron recognizes the input vector (ex p5)
- Identified: one or several neurons recognize the vector and they agree with its category value (ex: p1)
- Uncertain: several neurons recognize the vector and they are not all in agreement with its category value. (ex: p2, p3, p4)

### 3.1.2 Multi-level responses

When a recognition is uncertain, the user can take different actions depending on his application:

- Read the response of the neuron with the best match (i.e. smallest distance)
- Read the response of the N top firing neurons and apply some statistical or probabilistic rules to generate a single global response
- Discard the response but trigger the use of another network of neurons assigned to a different context and trained with a different feature which might be more relevant to waive the current existing uncertainty.

#### Best-match classification

The response of the firing neuron with the smallest distance value is equivalent to a best match. It can still be uncertain and in this case a detailed classification can be read out.

If a neuron fires with a distance 0, it means that the vector matches exactly the prototype stored in the neuron. The classification is an **exact match**.

#### Detailed K classification

Examining the distance and category of all the firing neurons can be of interest to reinforce the accuracy of a decision, especially in the cases of uncertainty. Rules must be established on a “per application” basis depending on the cost of a mistake, the requirements for a minimum throughput, minimum false negative, etc.

Let’s take the example of a recognition where the following 6 firing neurons recognize a vector:

Neuron 1	2	3	4	5	6	
Distance	12	13	15	38	39	
Category	8	7	7	7	3	5

The best match is a neuron with a category 8 reporting a distance equal to 5 between its prototype and the input vector. However, the next three firing neurons with a higher distance report a category 7.

If the cost of an inaccurate recognition is low, the response of the 1<sup>st</sup> neuron with category 8 is the simplest to retrieve (and very fast). On the contrary, if the application cannot afford a false-positive, it might be wiser to involve some statistics and assume that category 7 is the dominant category and should be the one selected for a final decision. More sophisticated rules can be deployed including the analysis of the histogram of the categories, and more. Some applications might even consider the generation of a “response” vector composed of all the “firing” categories (i.e. 8,7,7,7,3,5) and to be classified by another set of neurons taught to classify the “response” vectors. NeuroMem can handle up to 127 subsets of neurons trained for different purposes. These subsets are called [contexts](#).

## 4 Flexibility of the RCE/RBF model generator

---

- The mapping of the decision space can vary from conservative to moderate by using different values of the [Maximum Influence Field](#). or by teaching counter-examples.
- The decision space can be tuned by learning both examples and their counter examples
- The zones of uncertainty can be controlled by using different values of the [Minimum Influence Field](#).

### 4.1 Non-linear model generator

---

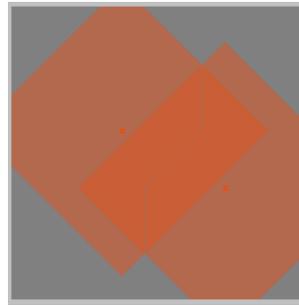
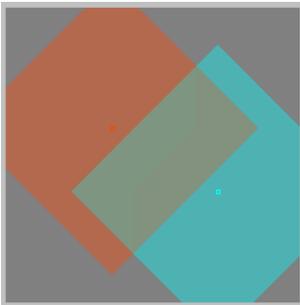
One of the strengths of the NeuroMem neurons is that they know when and how to adapt their influence field to comply with the teacher.

If a neuron containing a prototype of category A recognizes a vector V and V must be learned as category B, the neuron automatically reduces its influence field to exclude V. All neurons firing with a category different from B do the same and a new neuron is committed to store V as a new prototype with a category B. The influence field of this new neuron is set to the smallest distance of all the firing neurons.

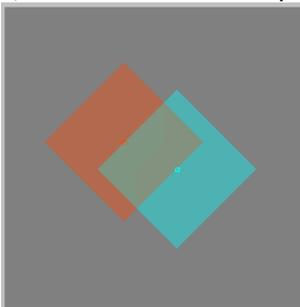
Learning new categories of vectors reduces the influence field of existing neurons and creates new smaller neurons. The category 0 has a special functionality in the sense that it reduces existing neurons, but does not create new ones. Learning a category 0 is equivalent to showing a counter-example, or a background example.

The following illustrations show how neurons adjust their influence field as new examples are taught.

If two examples are significantly different, two neurons are committed to store their (X, Y) vector and C category. Their influence fields are set to the default maximum value. They overlap slightly, thus defining a zone of redundancy for the category  $C_{blue}$ , or a zone of uncertainty between the categories  $C_{red}$  and  $C_{blue}$ ,



If two examples are close and with the same category, one neuron is sufficient to ensure their recognition. If they have different categories, two neurons are created, but their influence fields become equal to their distance minus 1, so the neurons mutually exclude the prototype of the other one.

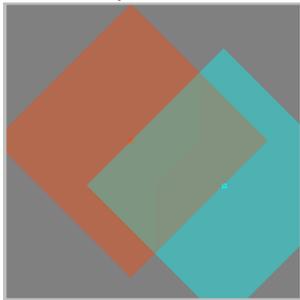


The tendency to create neurons with small or large influence fields can be controlled by teaching counter examples or by changing the Maximum Influence Field global register.

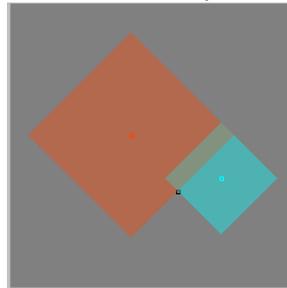
#### 4.1.1 *Modulating liberalism by teaching counter examples*

The definition of conservatism is to prevent the neurons from over generalizing what they are taught. The easiest way to avoid this behavior is to teach limits or examples which the neurons should not associate to their model. This is the purpose of the category 0 or Null category.

In this example, a red and blue example are taught and the 2 committed neurons share a large area of uncertainty.



Teaching a category 0 (back example) is only intended to force some neurons to reduce their influence field. It does not create any new neuron.



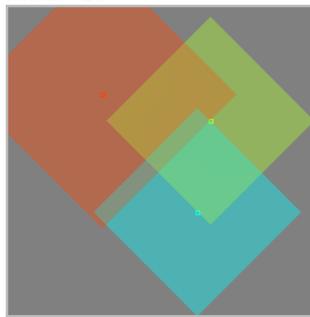
#### 4.1.2 *Modulating liberalism with the Maximum Influence Field*

The Maximum Influence Field is the default value of a newly committed neuron when no other neuron recognizes the vector to learn

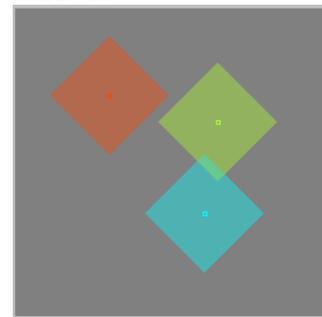
The higher the Maximum Influence Field, the larger the similarity domains of the neurons and the more liberal the recognition engine. The neurons with large influence fields have the tendency to over-generalize and recognize patterns with a high throughput. Liberalism is practical when the cost of an error is not critical and the number of neurons available is limited. If, on the other hand, an application has a very low tolerance to errors, it is best to set a small Maximum Influence Field or teaching many counter-examples. A conservative engine will require more teaching than a liberal engine because it must understand the diversity and subtle differences between many examples.

The graphs to the right illustrate how the Maximum Influence Field can help fill the decision space using the same set of examples.

Maxif=100



Maxif=50



Warning: Changing the Maximum Influence Field once neurons are already committed can affect the consistency of the knowledge under construction.

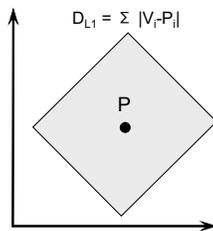
## 4.2 Shaping the decision space with a Norm

The Norm determines how to calculate the distance between the reference pattern or prototype stored in a neuron (P) and an input vector (V). If the calculated distance is less than the influence field of the neuron, the vector V is considered as similar to the prototype P.

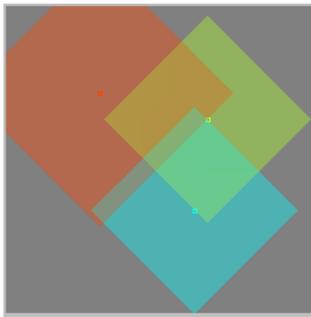
### 4.2.1 Norm L1 (Manhattan distance)

$$\text{Distance} = \text{Sum } |V_i - P_i|$$

The L1 distance emphasizes the drift of the sum of the all components between V and P.



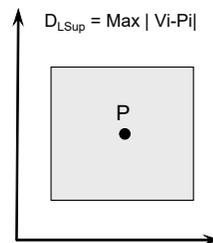
In a two dimensional space, the points located at a distance  $D_{L1}$  from P such that  $D_{L1} < C$  describe a diamond centered in P and with a side equal to  $\sqrt{2} \times C$ .



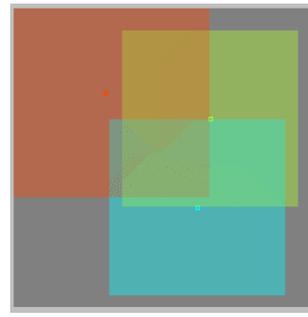
### 4.2.2 Norm L Sup

$$\text{Distance} = \text{Max. } |V_i - P_i|$$

The Lsup distance emphasizes the largest drift of the same component between V and P.



In a two dimensional space, the points located at a distance  $D_{Lsup}$  from P such that  $D_{Lsup} < C$  describe a square centered in P and with a side equal to  $2 \times C$ .



### 4.2.3 Norm L2 (Euclidian distance)

The L2 Norm calculating  $\text{Distance} = \sqrt{\text{Sum } (V_i - P_i)^2}$  would give circular shapes to the neurons' influence field. However, the little precision improvement of the L2 distance compared to the L1 distance did not justify adding this processing-intensive calculation in the ASIC which would have significantly impacted its size.

### 4.3 Building zones of uncertainty

---

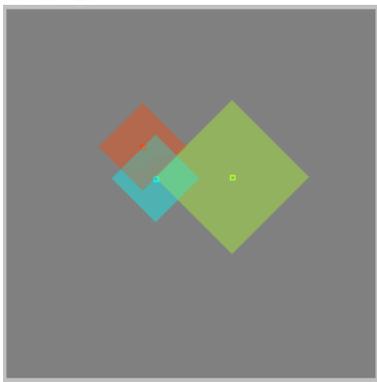
The existence of areas of uncertainty can be important to weigh a decision, especially if a mistake has a high cost. The belonging of a pattern to such an area can trigger the need for a thorough analysis in the current decision space and possibly require the usage of another decision space, if applicable, to give a second opinion.

#### 4.3.1 Modulating uncertainty zones with the Minimum Influence Field

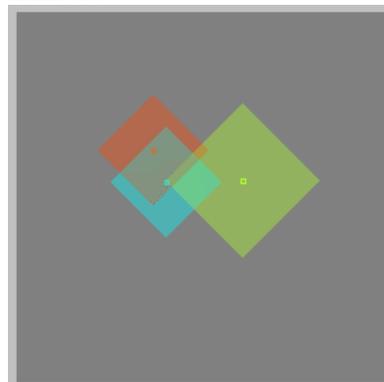
The Minimum Influence Field (MINIF) is the value below which the active influence field of a neuron cannot shrink. The Minif of a neuron is set at the time it is committed and cannot change after. If during a training session the neuron's Influence Field reaches its Minif, it is tagged as degenerated. This tag is a warning when the neuron fires, that its model lies close to the boundary of another neuron of a different category.

The graphs below show how a higher Minimum Influence Field can build larger zones of uncertainty. The neurons which are degenerated are shown as filled boxes.

Minif=10



Minif=50



### 4.4 Benefits of Iterative learning

---

The decision space is modeled as examples are taught and its shape depends on their sequence. This dependency is not desirable to build an accurate knowledge and it is recommended whenever possible to learn the examples repeatedly until the decision space is stable. This condition is established when no new neuron is committed between two iterations. Obviously, the ability to execute an iterative learning requires that the examples be archived which is technically possible in most applications.

#### 4.4.1 Incremental Decision Space mapping

If an example A is taught as category #1, the neural network first determines if it recognizes the example. If only one category is recognized and it is category #1, the network does not take any action and simply discard the example A. If several categories are recognized, including the category #1, the network does not commit a new neuron to store the example, but instructs the neurons responding with a category other than #1 to shrink their influence fields.

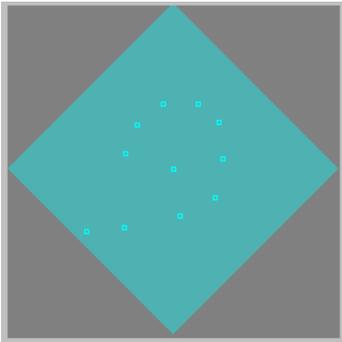
With the learning of additional examples, the decision space will continue to change and it is possible that Example A might no longer fall inside the influence field of a neuron recognizing the category #1. Therefore, it may no

longer be recognized, or fall in the influence field of a neuron created later and which overgeneralizes too much in the absence of counter examples forcing it to shrink. Teaching Example A again will commit a new neuron with the category #1.

The order in which the examples are taught has a direct impact on the shape of the decision space, so whenever possible it is best to learn the examples iteratively until no more neurons are committed to build a stable and accurate knowledge.

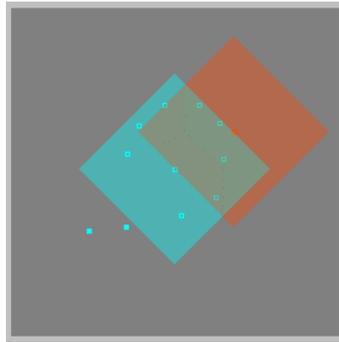
Sequence 1:

A single neuron recognizes the 10 initial examples of the “blue” category.



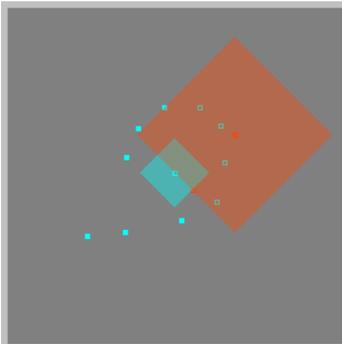
Sequence 2:

The first example of the “red” category causes the single neuron to shrink. Two of the former blue examples are no longer recognized.



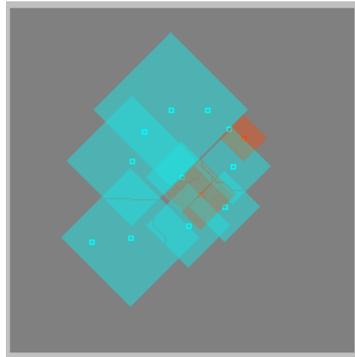
Sequence 3:

The second example of the “red” category causes the first neuron to shrink even more. 5 original blue examples are no longer recognized.



Iterative learning:

Re-learning all the examples will this time create 7 blue neurons and 2 red neurons

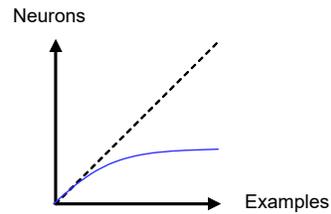


Whenever possible, it is recommended to record the examples so they can be learned in a repetitive loop until the knowledge gets steady and no more neurons are created or shrunken. The decision space is then the most accurate possible since all examples have been taken into consideration.

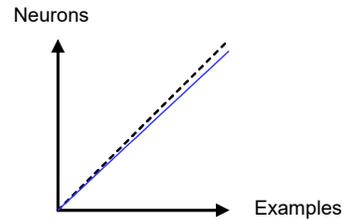
#### 4.4.2 Learning curve and feature relevancy

The learning curve plots the number of neurons created as new examples are taught.

If vectors learned by the neurons are significant for the intended classification, the neurons should generalize quickly and their number should reach a constant following a logarithmic curve



A linear learning curve with a slope close to 1 as shown to the right is a hint that the selected feature vector does not perform well since the neurons do not generalize.



If a feature is not performing as expected, studying the learning curves per categories of examples might be helpful to identify the categories which will be easy or hard to discriminate with the selected feature.

## 5 KNN (K-Nearest Neighbor) classifier

---

In KNN mode, the notion of influence field is discarded, and the network always returns K responses which are the first K closest matches.

For applications requiring a simple KNN classification, it is common to load the neurons with reference patterns and their associated categories if any, without any need for the RBF model generator, but rather as a regular memory (in Save\_and\_Restore mode).

### 5.1 Fastest KNN classifier

---

A NeuroMem network used as a KNN classifier has unique speed performances since the latency of each query is deterministic regardless of the number of neurons loaded with reference patterns.

Depending on the requested outputs, a single query will take 18 clock cycles for the distance plus 19 clock cycles for the category. Therefore, a query for the K nearest neighbors will take  $K * 37$  clock cycles, and the K responses are automatically sorted per order of increasing distance.

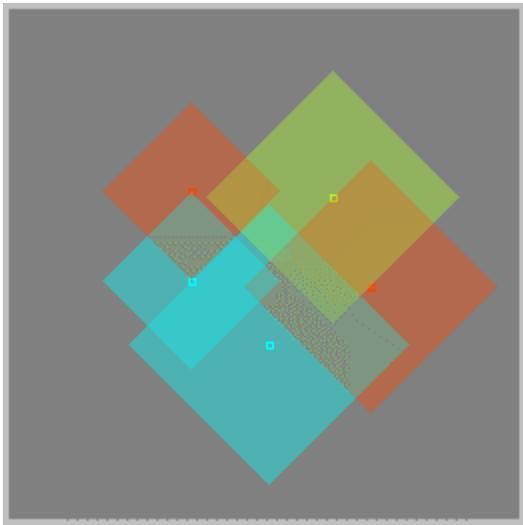
### 5.2 KNN is not a model generator

---

The following plots illustrate how a set of identical neurons can model different decision spaces when used in RCE/RBF or KNN recognition mode.

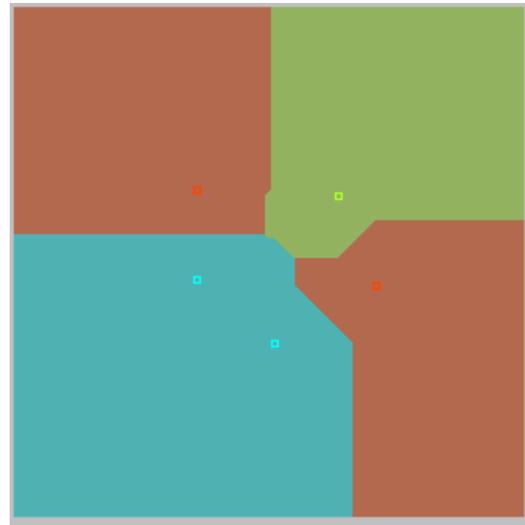
#### RCE/RBF

The space is mapped partially with certain zones being unclassified (i.e. black color). The zones with multiple colors are zones of uncertainty.

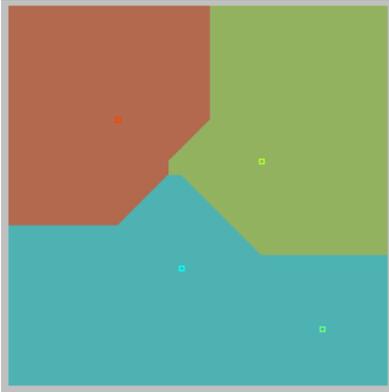


#### KNN

The entire space is mapped and with a single possible category (i.e. color code) per position. The notion of Unknown and Uncertainty is inexistent.



**Why KNN is a recognition mode ONLY, and should NOT be active while learning:**



Indeed, as soon as you have trained one neuron with a category A, it will fire along with all the other committed neurons and prevent the learning of new examples of category A by pulling the ID line of the network, even though this neuron might not be the closest firing neuron. As a result, if the network is set to KNN prior to learning, the decision space will have only one neuron per category. On the other end, a decision space featuring multiple models per category can be trained in RBF mode or loaded to the neurons using the SR mode and later be used for recognition in KNN mode.

**5.3 Warning: KNN classification using RBF models**

---

RBF is a powerful non-linear model generator entertaining area of Unknown and Uncertainty in the decision space and reporting these states of classification for a richer diagnostic.

Technically, it is possible to teach the neurons with the RBF model generator and switch their behavior to KNN for the classification. However, you must be cautious with possible artifacts derived from the shape of the decision space as illustrated below:

Given the decision space with the three neurons (red, green and blue) and the stimuli shown with the black dot:

RBF response:

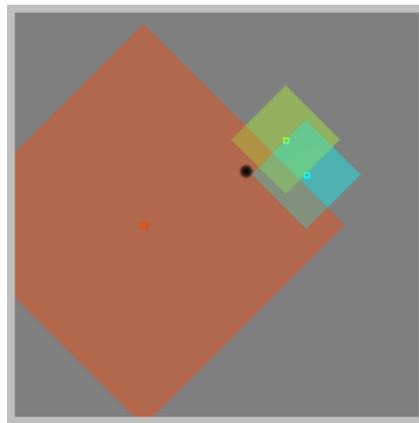
Red neuron, Distance 128

KNN response:

Green neuron, Distance 25

Blue neuron, Distance 26

Red neuron, Distance 128



## 6 From decision space to knowledge base

---

Once a decision space has been modeled and validated by recognizing many samples, the contents of the neurons can represent a valuable knowledge base and intellectual property. This information includes the prototype stored in the neuron, its context, category and influence field. Also the knowledge file must be associated to the description of the features used to train the neurons for each context in use.