Beyond supervision

What happens when there is no supervision

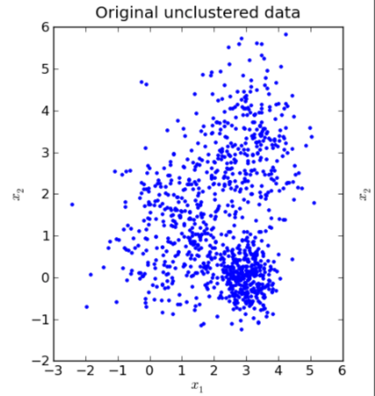
- number / structure of classes is unknown
- Yd can not be known in advance
- Error gradient can not be computed
- back propagation can not be computed

Beyond supervision

we can still split the space in k classes : (but is k appropriate ?)

K-means algorithm will categorize without supervision.

- Let suppose you want to categorize n samples into a given number of k categories
- you only know the number of categories you wish to obtain : k
- but you have no idea about the distribution of the samples

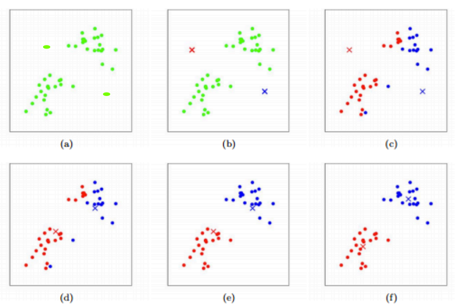


source : http://hduongtrong.github.io/2016/02/10/KMean-KMedoid-KernelPCA/

Beyond supervision

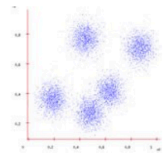
we can still split the space in k classes : (but is k appropriate ?)

1. *Choose randomly K points among the possible samples* $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$
2. *Assign each sample to the closest centroid :* $c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$
3. *When all sample have been assigned, recalculate the positions of the K centroids.* $\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}$
4. *Repeat Steps 2 and 3 until the centroids no longer move.*

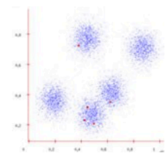


Beyond supervision

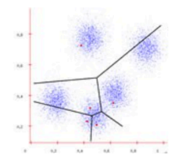
we can still split the space in k classes : (but is k appropriate ?)



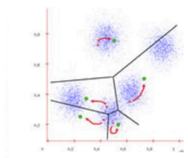
Step 0: Get the dataset



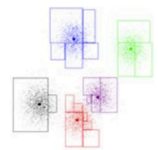
Step 1: Assign random centers



Step 2: “Own” datapoints



Step 3: Shift the centers



Step 4: The end!

Beyond supervision

we can still split the space in k classes : (but is k appropriate ?)

K-means algorithm

- The only information provided is the number of classes expected

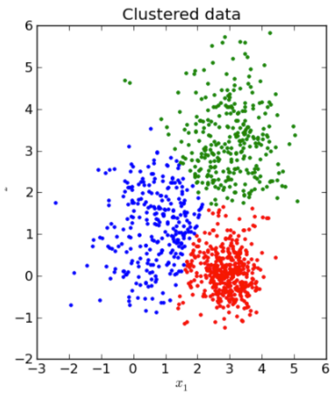
- forms convex classes

- unsupervised (no need to provide correction)

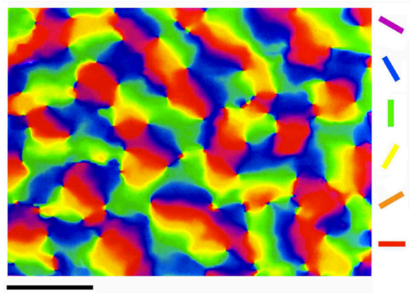
-information compression : any new sample belong to one of the 3 classes, and can be linked to one of the 3 centroids

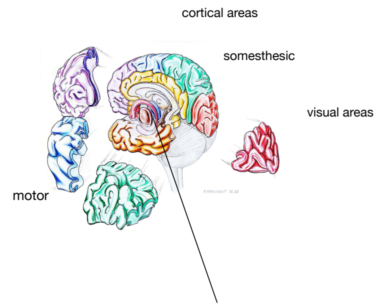
- be careful : different possible results according to the initialization (random) of the centroids.

- this is not neural networks :(



What about biology ?



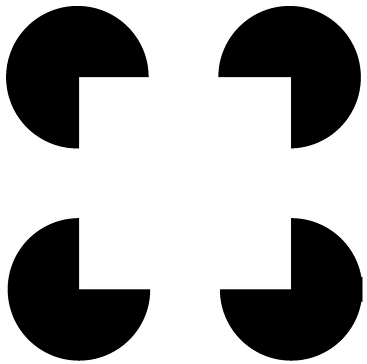


Hubel&Wiesel (1962, 1974):

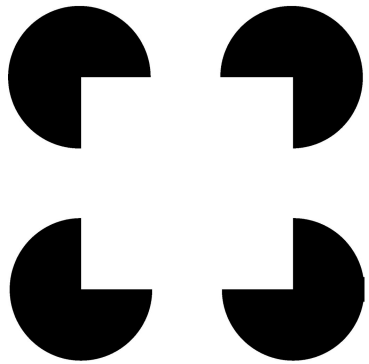
Map of the orientation detection in the primary visual area

«kind of continuity» : topology

What about biology ?



What about biology ?

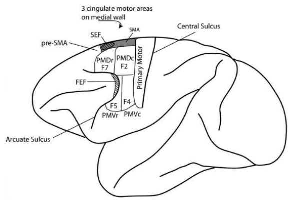


Lateral cooperation : activation of the closed neurons coding for the same orientation

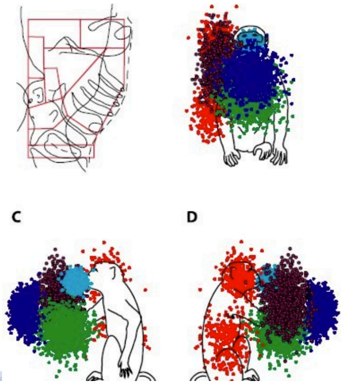
+ Competition : inhibition of the neurons coding for opposite interactions

= your brain perceives lines that don't exists

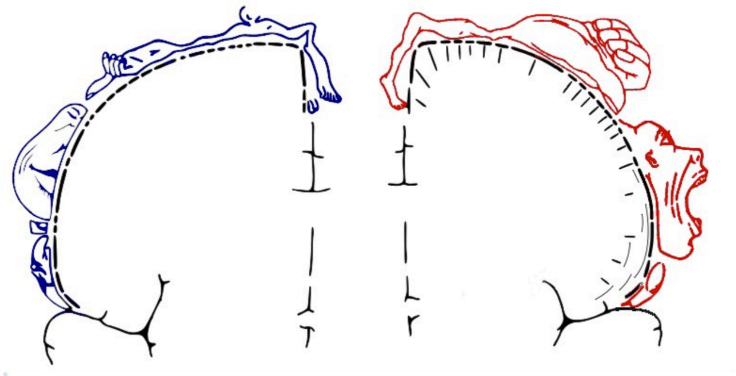
What about biology ?



Graziano (1997~)



What about biology ?

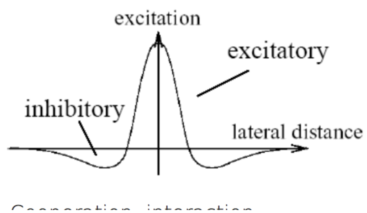


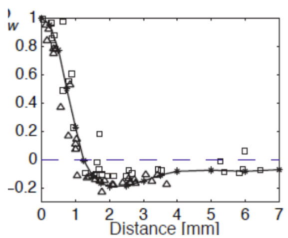
What about biology ?

homonculus :

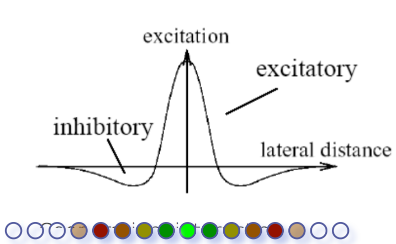
What about biology ?

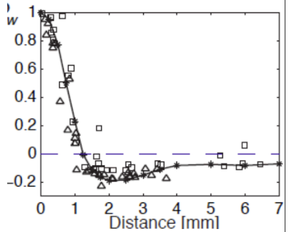
Working hypothesis : maps (i.e groups of neighbor) neurons are auto-organizing themselves thanks to lateral connections, co-activation and competition.





What about biology ?

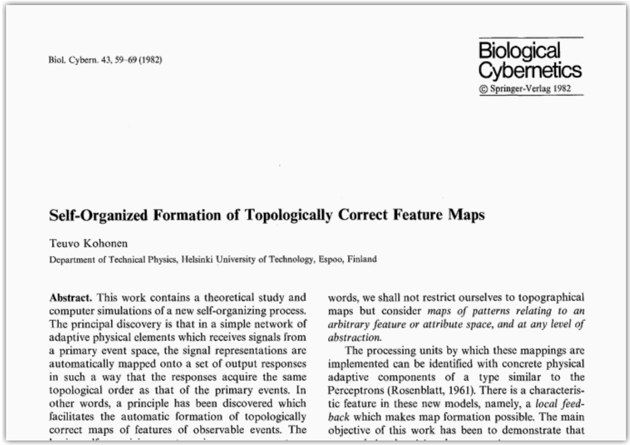




In a topological map :

- each neuron has a function of discrimination of an input pattern
- the most active neuron is the winning neuron (competition)
- the most active neuron can stimulate its closest neighbors (cooperation)
- the most active neuron can inhibit the surrounding neighborhood (competition)

self-organizing maps (SOM)



Biol. Cybern. 43, 59–69 (1982)

Biological Cybernetics

Self-Organized Formation of Topologically Correct Feature Maps

Teuvo Kohonen

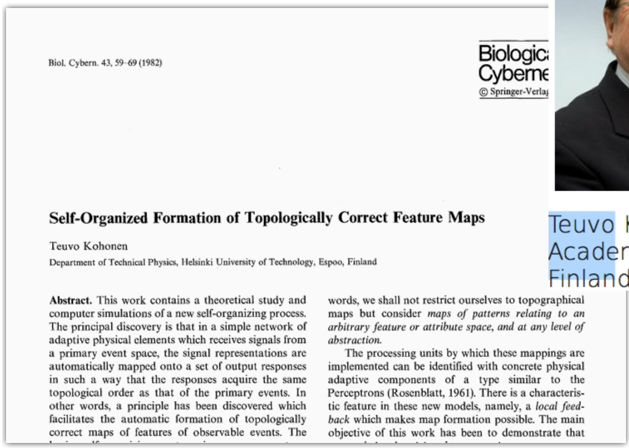
Department of Technical Physics, Helsinki University of Technology, Espoo, Finland

Abstract. This work contains a theoretical study and computer simulations of a new self-organizing process. The principal discovery is that in a simple network of adaptive physical elements which receives signals from a primary event space, the signal representations are automatically mapped onto a set of output responses in such a way that the responses acquire the same topological order as that of the primary events. In other words, a principle has been discovered which facilitates the automatic formation of topologically correct maps of features of observable events. The

words, we shall not restrict ourselves to topographical maps but consider *maps of patterns relating to an arbitrary feature or attribute space, and at any level of abstraction.*

The processing units by which these mappings are implemented can be identified with concrete physical adaptive components of a type similar to the Perceptrons (Rosenblatt, 1961). There is a characteristic feature in these new models, namely, a *local feed-back* which makes map formation possible. The main objective of this work has been to demonstrate that

self-organizing maps (SOM)



Biol. Cybern. 43, 59–69 (1982)

Biologic
Cyberne
© Springer-Verla

Self-Organized Formation of Topologically Correct Feature Maps

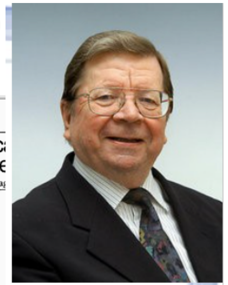
Teuvo Kohonen

Department of Technical Physics, Helsinki University of Technology, Espoo, Finland

Abstract. This work contains a theoretical study and computer simulations of a new self-organizing process. The principal discovery is that in a simple network of adaptive physical elements which receives signals from a primary event space, the signal representations are automatically mapped onto a set of output responses in such a way that the responses acquire the same topological order as that of the primary events. In other words, a principle has been discovered which facilitates the automatic formation of topologically correct maps of features of observable events. The

words, we shall not restrict ourselves to topographical maps but consider maps of patterns relating to an arbitrary feature or attribute space, and at any level of abstraction.

The processing units by which these mappings are implemented can be identified with concrete physical adaptive components of a type similar to the Perceptrons (Rosenblatt, 1961). There is a characteristic feature in these new models, namely, a local feed-back which makes map formation possible. The main objective of this work has been to demonstrate that



Teuvo Kohonen
Academy of Finland

self-organizing maps (SOM)

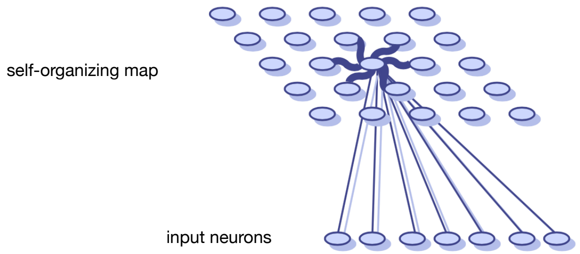
In a self-organizing map :

- each neuron has a function of discrimination of an input pattern
- the most active neuron is the winning neuron (competition)
- the most active neuron can stimulate its closest neighbors (cooperation) : **they learn at the same time the input pattern (specialization)**
- the most active neuron can inhibit the surrounding neighborhood (competition): **they learn the opposite of the input pattern**

self-organizing maps (SOM)

In a self-organizing map :

- each neuron has a function of discrimination of an input pattern
- the most active neuron is the winning neuron (competition)
- the most active neuron can stimulate its closest neighbors (cooperation) : **they learn at the same time the input pattern (specialization)**
- the most active neuron can inhibit the surrounding neighborhood (competition): **they learn the opposite of the input pattern**



self-organizing maps (SOM)

1. Randomize the weights between the input and the self-organizing map
2. Select randomly an input I in the learning set
3. Compute the activity of each node in the map :
   Use the Euclidean distance formula to find the similarity between the input vector I and the map's node's weight vector

$$d_i = \sum_{j=1}^{M} (I_j - W_{ij})^2$$

   Select the winning neuron : the one that have the smallest d_i

4. Update the neurons in the neighborhood of the winning neuron by pulling them closer (or farther) to the input vector

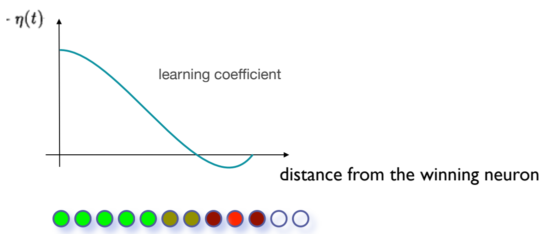
$$W_{kj}(t+1) = W_{kj}(t) + \eta(t) \cdot (I_j(t) - W_{kj}(t))$$

5. repeat from step 2 while stabilization of the weights

Be careful : $\eta(t)$ is decreasing with time (learning slows down)
$\eta(t)$ change the learning according to the distance of the current neuron from the winning neuron

self-organizing maps (SOM)

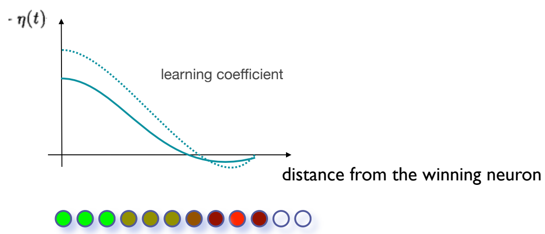
- $\eta(t)$ change with time



self-organizing maps (SOM)

- $\eta(t)$ change with time

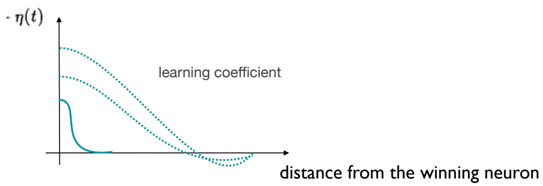
later...



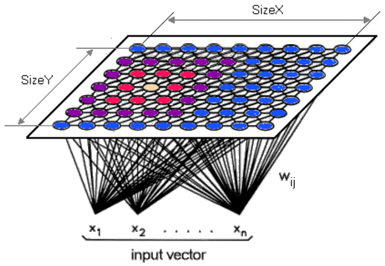
self-organizing maps (SOM)

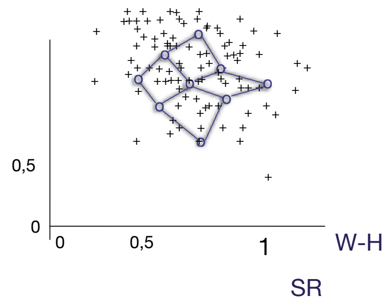
- $\eta(t)$ change with time

...and later....



self-organizing maps (SOM)

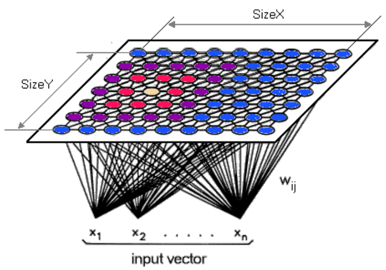


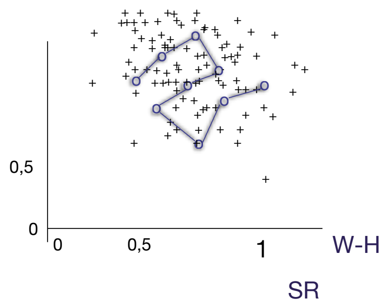


self-organizing maps (SOM)

Let the network learn the topology of the input space:

kohonen, SOM



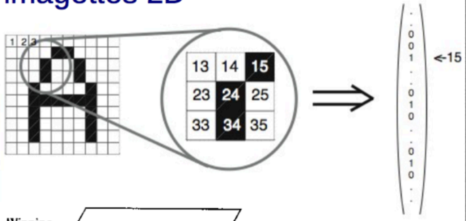


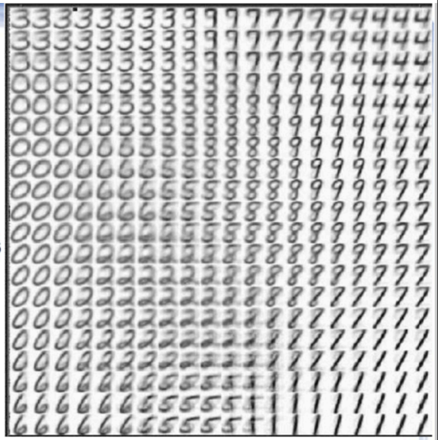
self-organizing maps (SOM)

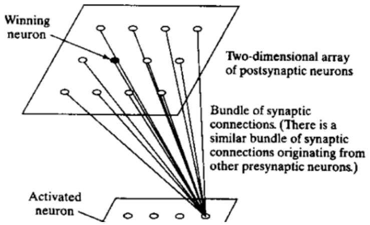
```
%% Two dimensional self-organizing feature map al la Kohonen
 clear; nn=10; lambda=0.2; sig=2; sig2=1/(2*sig^2);
 [X,Y]=meshgrid(1:nn,1:nn); ntrial=0;

 % Initial centres of prefered features:
 c1=0.5-.1*(2*rand(nn)-1);
 c2=0.5-.1*(2*rand(nn)-1);

%% training session
 while(true)
    if(mod(ntrial,100)==0) % Plot grid of feature centres
        clf; hold on; axis square; axis([0 1 0 1]);
        plot(c1,c2,'k'); plot(c1',c2','k');
        tstring=[int2str(ntrial) ' examples']; title(tstring);
        waitforbuttonpress;
    end
    r_in=[rand;rand];
    r=exp(-(c1-r_in(1)).^2-(c2-r_in(2)).^2);
    [rmax,x_winner]=max(max(r)); [rmax,y_winner]=max(max(r'));
    r=exp(-((X-x_winner).^2+(Y-y_winner).^2)*sig2);
    c1=c1+lambda*r.*(r_in(1)-c1);
    c2=c2+lambda*r.*(r_in(2)-c2);
    ntrial=ntrial+1;
 end
```

Reconnaisse des formes

Vecteur d'entrées : dim N x N
imagettes 2D







self-organizing maps (SOM)

TABLE 9.3 Animal Names and Their Attributes

Animal		Dove	Hen	Duck	Goose	Owl	Hawk	Eagle	Fox	Dog	Wolf	Cat	Tiger	Lion	Horse	Zebra	Cow
is	small	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0
	medium	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	big	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
has	2 legs	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	4 legs	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	hair	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	hooves	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
	mane	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0
	feathers	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
likes to	hunt	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0
	run	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0
	fly	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
	swim	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

self-organizing maps (SOM)

dog	dog	fox	fox	fox	cat	cat	cat	eagle	eagle
dog	dog	fox	fox	fox	cat	cat	cat	eagle	eagle
wolf	wolf	wolf	fox	cat	tiger	tiger	tiger	owl	owl
wolf	wolf	lion	lion	lion	tiger	tiger	tiger	hawk	hawk
wolf	wolf	lion	lion	lion	tiger	tiger	tiger	hawk	hawk
wolf	wolf	lion	lion	lion	owl	dove	hawk	dove	dove
horse	horse	lion	lion	lion	dove	hen	hen	dove	dove
horse	horse	zebra	cow	cow	cow	hen	hen	dove	dove
zebra	zebra	zebra	cow	cow	cow	hen	hen	duck	goose
zebra	zebra	zebra	cow	cow	cow	duck	duck	duck	goose

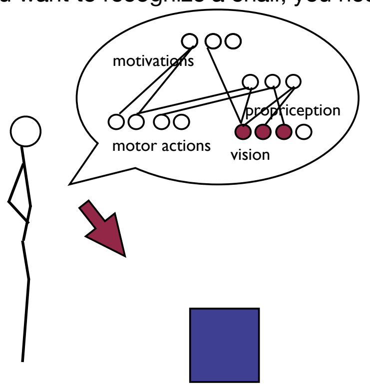
self-organizing maps (SOM)

tu sum up : SOM

+ no supervision : unsupervised leaning
+ to use when we do not have *a priori* information about the categories (number, shape, etc...)
- long time to convergence
- more parameters to tune (number of neurons, learning speed, lateral interactions)
+ categorization
+ topological organization
+ compression

Action matters...to produce perception

If you want to recognize a chair, you need to :



- **Build** the experience of seating
- to **categorize** it
  - *you need to have legs*
  - *you need to need to seat*
  - *merge the perceptions, the motor actions*
  - *to associate the objects vision with the whole*
  - *in order to generalize and recognize*

Action matters...to produce perception

Connectionism is a part of this perception-action philosophy...

and neural networks is a tool to achieve such adaptive categorization

PerAc

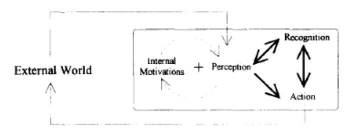
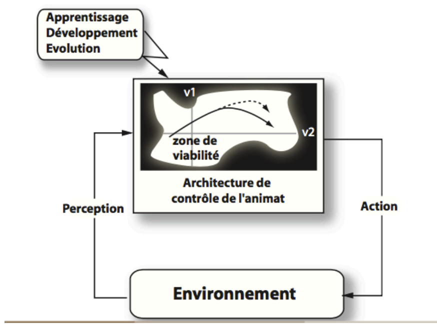


Fig. 3. The concept of PerAc architecture to control autonomous robots.



PerAc

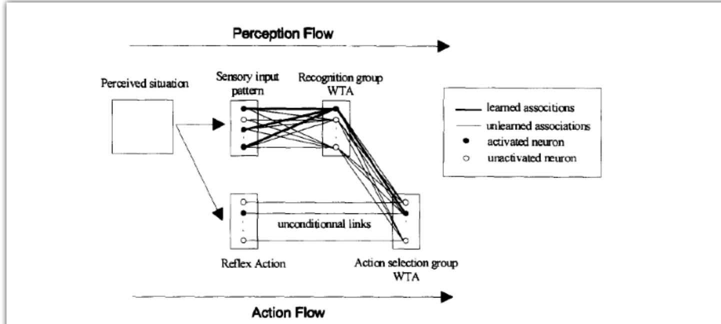


Fig. 5. Architecture of a PerAc Block. From the perceived situation a reflex action and a sensory input pattern are extracted. The action group learns to associate the recognized situation with the unconditional input. The system can then recognize a situation and react correctly even if the reflex mechanism is not activated.

PerAc

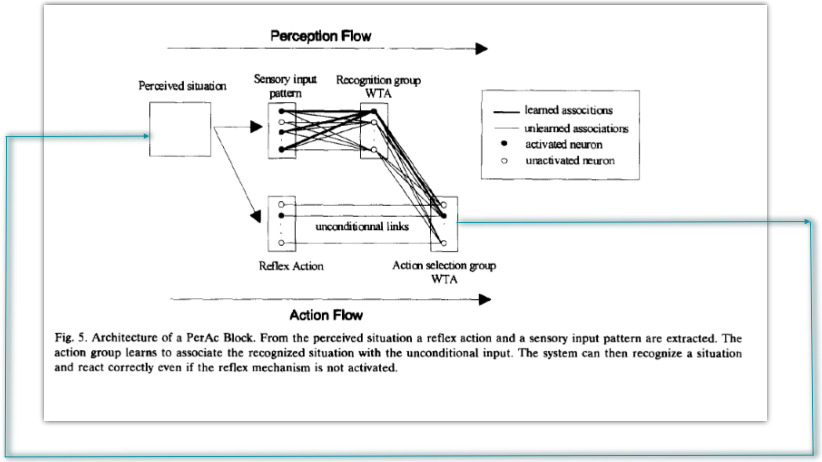


Fig. 5. Architecture of a PerAc Block. From the perceived situation a reflex action and a sensory input pattern are extracted. The action group learns to associate the recognized situation with the unconditional input. The system can then recognize a situation and react correctly even if the reflex mechanism is not activated.

Rassemblons...

Counter Propagation [Nielsen]

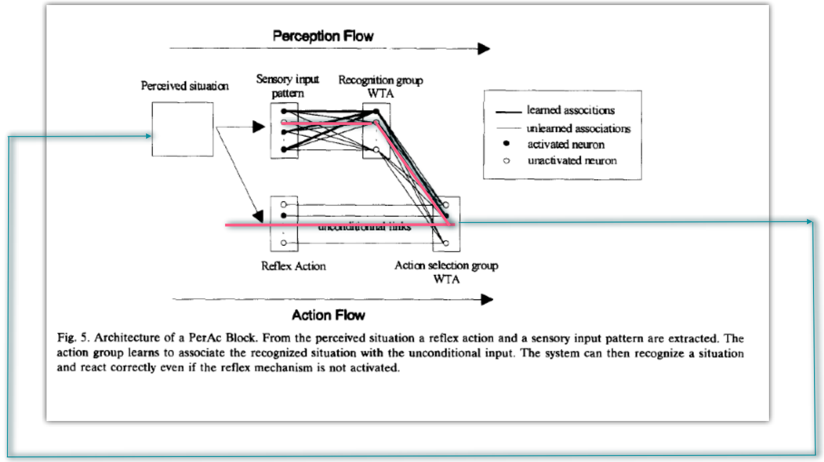


Fig. 5. Architecture of a PerAc Block. From the perceived situation a reflex action and a sensory input pattern are extracted. The action group learns to associate the recognized situation with the unconditional input. The system can then recognize a situation and react correctly even if the reflex mechanism is not activated.

Rassemblons...

Counter Propagation [Nielsen]

WTA, Kohonen, INSTAR

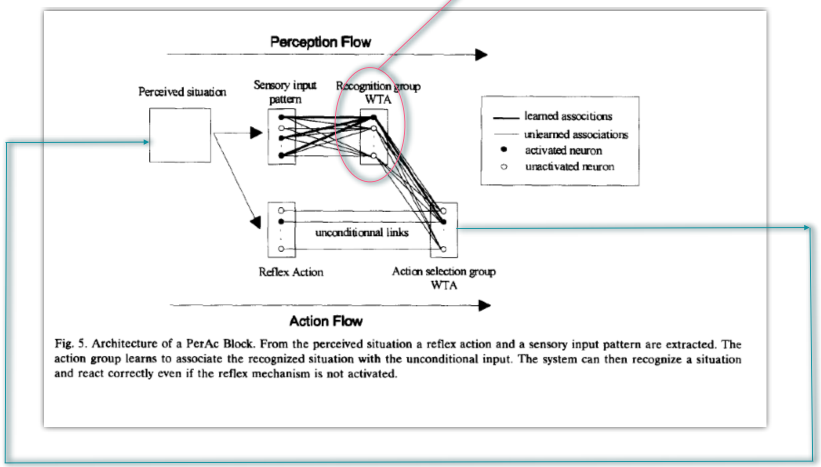


Fig. 5. Architecture of a PerAc Block. From the perceived situation a reflex action and a sensory input pattern are extracted. The action group learns to associate the recognized situation with the unconditional input. The system can then recognize a situation and react correctly even if the reflex mechanism is not activated.

Rassemblons...

Counter Propagation [Nielsen]

WTA, Kohonen,

outstar - LMS - ART (SAW)

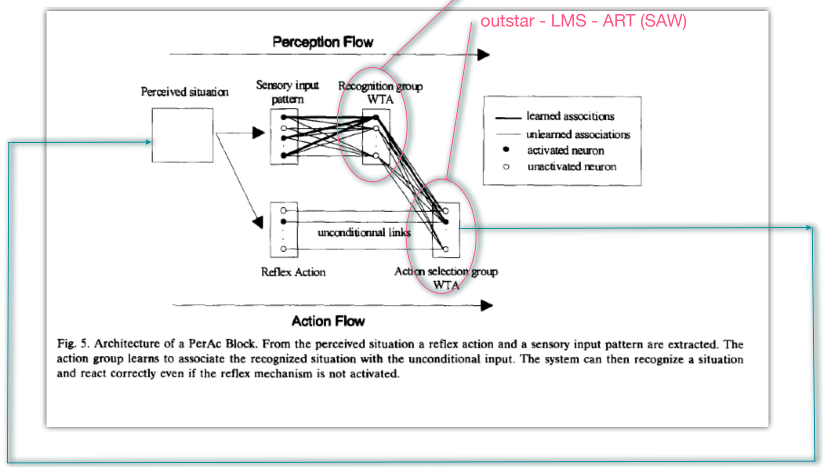


Fig. 5. Architecture of a PerAc Block. From the perceived situation a reflex action and a sensory input pattern are extracted. The action group learns to associate the recognized situation with the unconditional input. The system can then recognize a situation and react correctly even if the reflex mechanism is not activated.

Rassemblons...

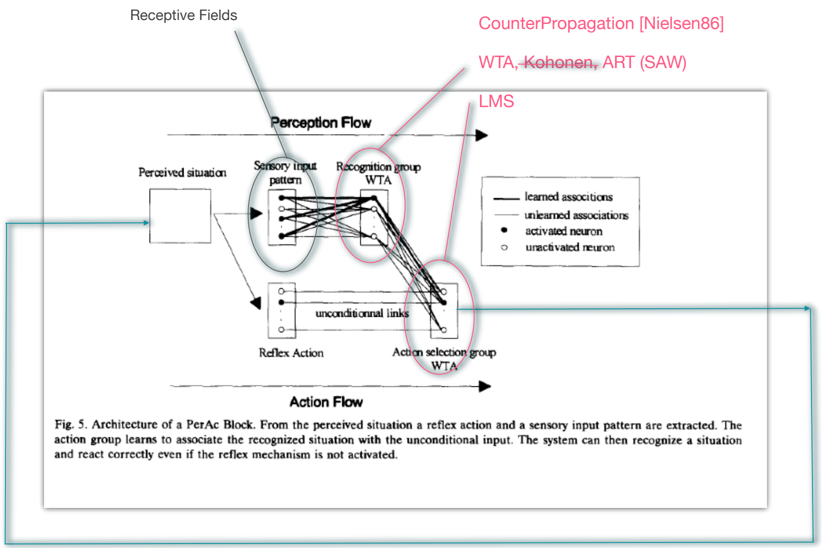


Fig. 5. Architecture of a PerAc Block. From the perceived situation a reflex action and a sensory input pattern are extracted. The action group learns to associate the recognized situation with the unconditional input. The system can then recognize a situation and react correctly even if the reflex mechanism is not activated.

application

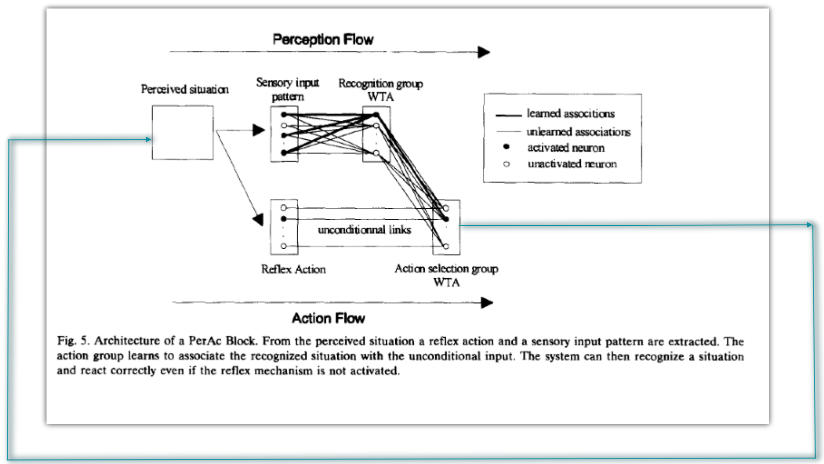
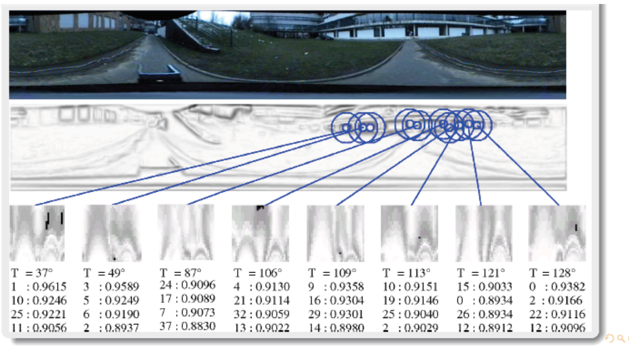
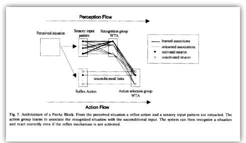


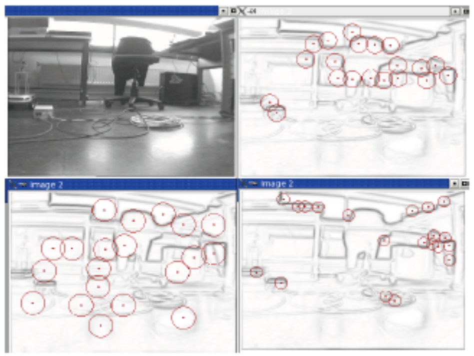
Fig. 5. Architecture of a PerAc Block. From the perceived situation a reflex action and a sensory input pattern are extracted. The action group learns to associate the recognized situation with the unconditional input. The system can then recognize a situation and react correctly even if the reflex mechanism is not activated.

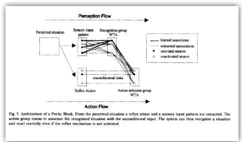
application



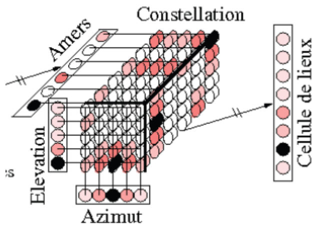


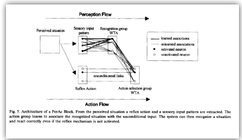
application



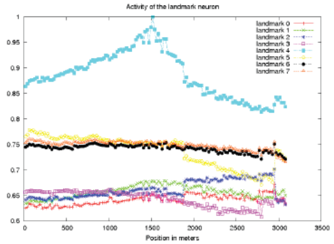


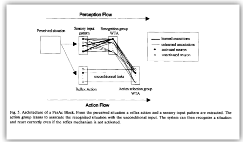
application



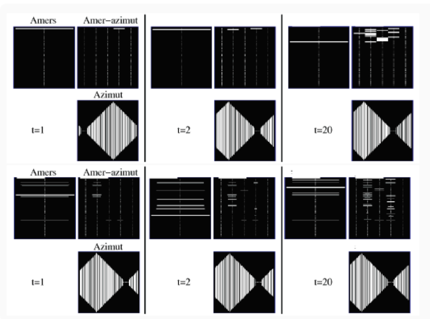


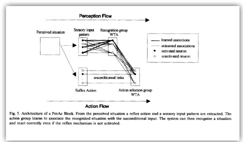
application



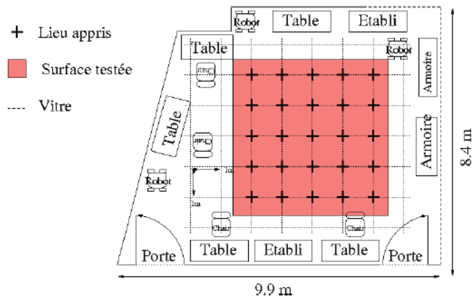


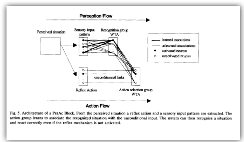
application



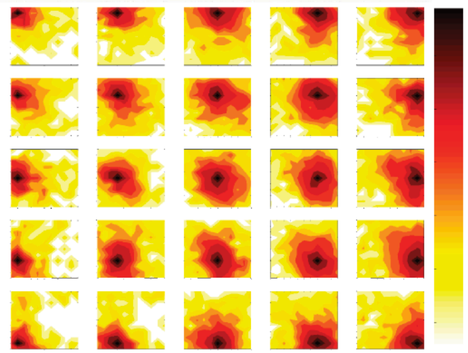


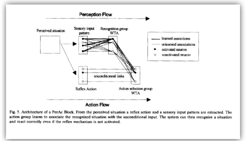
application





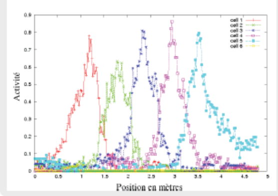
application



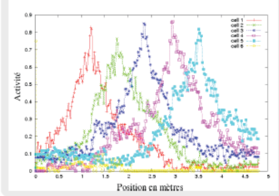


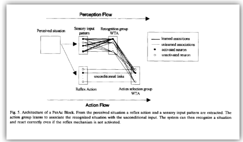
application

Compétition dure (1 gagnant)

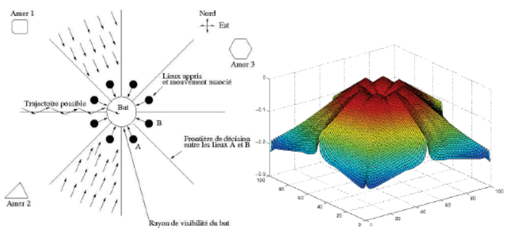


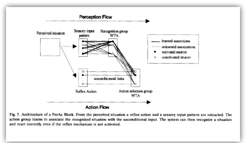
Compétition molle (4 gagnants)



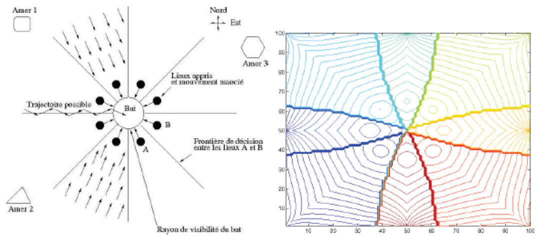


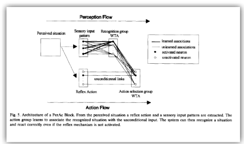
application



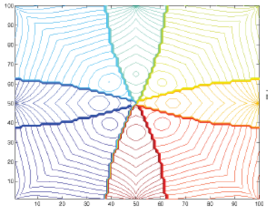


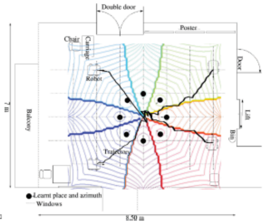
application





application





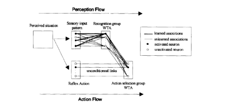


Fig. 5. Architecture of a PerAc Block. From the perceived situation a reflex action and a sensory input pattern are extracted. The action group learns to associate the recognized situation with the unconditional input. The system can then recognize a situation and react correctly even if the reflex mechanism is not activated.