Multi-Modal Convergence Map

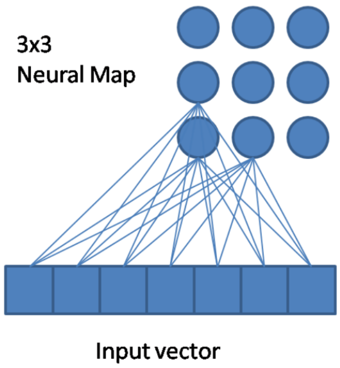
In this page I explain the model of multi modal convergence map, the related algorithms and the possible applications.

Model motivation

[NOM] shown that the brain is likely to contain convergence zones. These zones are responsibles for linking together the various sensory modalities of the same concepts, for example seeing a photo of a very dirty and wet dog can give you a feeling of its smell. Moreover, such structures doing the link beetween modalities can also represent functions rather than static concepts : synesthetes use colors instead of numbers to compute highly complex mathematical calculus. A model of these structures can be very usefull to mimic cognitive functionnalities. In this page I will describe a simple but powerfull model, inspired from Self Organizing Map, which can be used to merge different sensory modalities in a neuro-inspired map. I will explain how this model can be extended in a hierarchical way to represent a more global cognition system.

Self Organizing Map

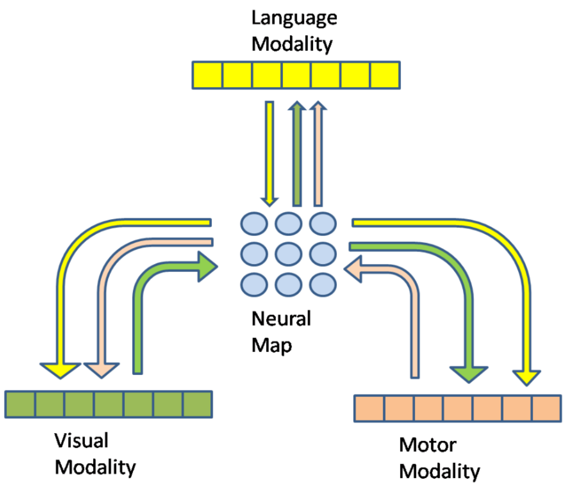
Self Organizing Maps (SOM) were introduced by Kohonen and are a well known model in connectionism. The main purpose of SOMs is to operate a vector quantization and to represent high dimensional data. I will explain briefly the concept of a classic SOM here.



A SOM is a 2 (or more) dimension map of artificial neurons. An input vector is connected to the map so that each component of this input vector is connected to each node of the map (see a partial reprensentation of the connections on figure 1). In this context, each neuron of the map own a vector of connections that has the size of the input vector, and each connection has a weight. The main idea is to fill the input vector with values and to compare these values with the vector of weights of each neuron. Each neuron is activated by the similarity between its weight vector and the input vector. One neurons will more activated than the other, we call it the winner.
The main idea is to train the map so that 2 neurons that are close on the map will encode similar input vectors. To do so we present input vectors from a training set, calculate the winner neuron and modify its weigthts so they will be closer to the input values. Moreover, we also modify the weights of the winner neighbours so that a region instead of a single neuron will learn to respond to this vector. After a while the map will converge and will capture the main caracteristics of vectors of the training set.
Detail of the algorithm and samples are already well documented and explained, see [REF]

Using maps as a convergence zone

In this section I will show how to use the self organizing property of SOM to do a link between 2 or more modalities.



Co-occurence wells

Each modality dig the map in the same direction.
When a SOM is trained at the same time using all the modalities, it will result in a topological mapping between all these different modalities. After convergence, regions will appears and be kind of receptive fields for vectors of the differents modalities that are often co-occurents. In other words it means that if you always say "rock" when you close your hand, and then you say "rock" it will activate a region of the map that would have been activated if you have closed your hand. This region beeing activated by the word "rock", it is possible to activate in return the motor modality and have for example the robot closing its hand. And the map can work in any direction : assuming that the robot closes its hand, it will predict the right word to be said.

Feedforward stimulation (modality to map)

There are mainly two way of stimulating the map. Single modality stimulation and multi modal stimulation.
Single modality : in this case the map activation depent only of a single modality input. For example we stimulate the motor modality and predict the visual modality. The fusion of the different modalities occurs only during learning and not during the simulation. This is the stimulation I used until now,

but it is quite limited and can be only use to test some system hypothesis.
Multi modal : the map is stimulated by all modalities, so both weights of the differents modalities and map activation are dependent from the co-occurent inputs of the system. I predict that the convergence of the map will be faster but I'm still not sure about the quality of the prediction in case of a non complete modalities stimulation. However, with the last method, the system is more intended to be use in the Enactive Prediction model explained below.

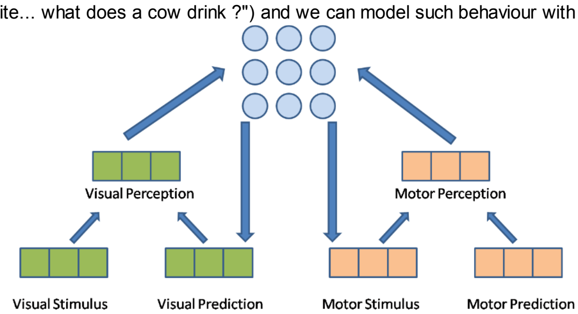
Feedback stimulation (map to modality)

Propagating activation from modalities to a map is easy and well defined, but the reverse propagation is not. Assuming that you have an activated map, how do you get back the modality that led you to this activated map ? We experimented different ways to achieve this purpose, we will present here their pro and con :
winner give all: only the most activated neuron from the map is used to get the modality. In fact the modality become equal to the weight vector of this neuron. It is a very fast but rought method, and you lose a lot of information and can even give you a high error (Figure of 2 completly different maps that have the same winner)
weighted mean: all the neurons contribute to calculate the modality, proportionnaly to their activation. So far we used a simple mean, which can be refined using ... (MOSAIC responsibilities formula).
treshold mean: we treshold the activation of the neurons, so it will become 0 or 1 and then we do a mean. It can removes a lot of the noise of the weighted mean, but the treshold has to be found manually. This method can be achieved in an elegant way by giving to the neurons a sigmoidal function, which will add a bit of non linearity to the system if it is used also during learning. This is the most promising method, but I havn't tested it yet.

Limitations
The map size is somehow limitative and related to the modalities dimensions and the aspect of the input space. The more modalities you try to link on a map, the more non linear it will have to be. (use a sheet of paper metaphore)

Model of Enactive Perception

Using the MMCM it is possible to link multiple different modalities according to the way we experienced them. It is possible to predict one modality from the activation of the map. We use this prediction capability to embbed the MMCM into a dynamic model which can represent the enaction. The main idea is that all that we perceive is not the reality, but our memories altered by the reality (REFERENCES ?). This can be experienced through many optical illusions or conditioning experiments ("white, white, white... what does a cow drink ?") and we can model such behaviour with the structure presend in figure below.



At any time the system experience the world, so the stimuli vector are set by the sensory input (St). However, this raw sensory data is not used purely used to stimulate the MMCM, instead we use a mixture of this sensory input and the prediction (Pr) made by the system at the previous time step. This perceived vector (Pe) is calculated using the formula :

$$Pe_t = \rho.St_t + (1-\rho).\mathrm{Pr}_{t-1}$$

Rho is a parameter of the system which which vary in [0,1] and represents the weight of the reality in the perception of the system. The smaller it is the more the system will be influenced by its memory.
The add of this prediction to the inputs turn the system into a dynamic one. The MMCM does not learn only static association between modalities, but it also becomes linked to their temporal relationship. Which means that the system should be able to learn sequences of stimuli. This has still to be tested and can be very much related to the value of Rho (if we set it to 1 for example the system loose completly this capability.)

Prediction quality
Since the system is able to predict what the simuli should be, we can test its error.
The difference between the prediction and the reality is called Prediction Quality (PQ) and is scaled in [0-1]. (Find the formula)
If PQ=1 then the system predict exactly the reality, if PQ = 0 then it's prediction is completly wrong.
The PQ can be used in many different ways, specialy to influence learning.
It influence :
- the size of the neighborhood :

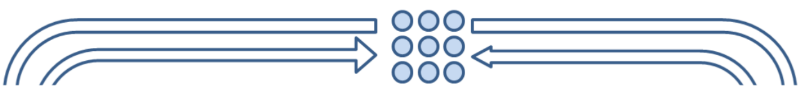
a good prediction should have a smaller neighborhood in order to refine the learning and to tackle catastrophic forgiving. Typically the neighborhood range function should be decreasing while PQ increases. I'm still not sure about what it should do when PQ is near to 0, it can also be good to have it decreasing (so the function will be a gaussian) so that the map will be modifed in a small area and that this new stimulus won't disturb the already defined wieghts.

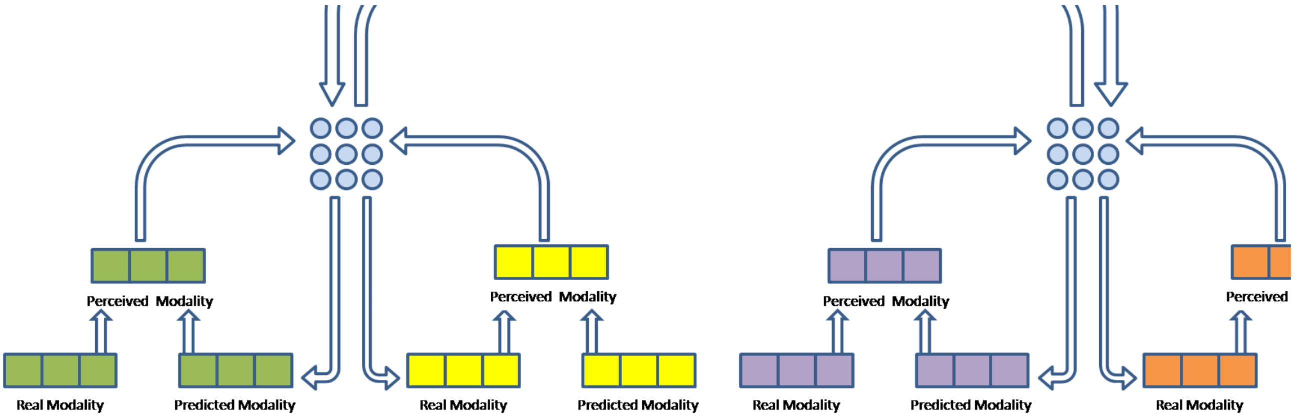
- the learning rate :

modifying the learning rate has more or less a similar effect as modifying the neighborhood. However, since the feedback stimulation may change (using a population code instead of only the winner for example) we can use it to avoid catastrophic forgiving regardless of the feedback stimulation chosen. If we modify the learning rate by a factor of a gaussian of PQ centered in 0.5, then we can choose to modify the weights only when the map can improve it's prediction without loosing too much of its knowledge. (It learns only if the map makes good but not perfect predictions). If the map makes very bad predictions on a particular stimulus, it means that it won't be able to learn it without forgiving something else. In this case we can choose to forgive it, or to increase the map size so that the new region will welcome this new stimulus.

Using a multi modal map hierarchy as a dynamic cognitive system

In this section I will propose a way of using a map hierarchy in a dynamic way.





Stimulation order

The system presented in the last figure link two enactive perception modules through another MMCM. However, even if this third map uses the same structure and learning algorithm as the lower level maps, the dynamic of the system can't be the same. In this case the lower level map act as a modality and stimulate directly the higher level map, we call that Feed Forward activation. On the other hand, the higher level map, being stimulated by both lower level map activate them in return, we call that Feed Back activation. Since this system is designed to be used on real time system, we choosed to use an asyncronous way of propagating the activation. It means that at each time step maps care only about receiving and sending activation at local scale, no matter which other map will use this sent activity. It leads to a dynamic system where higher level map will receive a stimulation that has been presented to the sensor a few time steps ago. However, sine the system is real time, the input flow should be a continum where two time close sensory inputs are similar.