



Organisms modeling: The question of radial basis function networks

Alexandre Muzy, Lauriane Massardier, Patrick Coquillard

► To cite this version:

Alexandre Muzy, Lauriane Massardier, Patrick Coquillard. Organisms modeling: The question of radial basis function networks. Workshop Activity-Based Modeling & Simulation (ACTIMS 2014), Centre National de la Recherche Scientifique (CNRS). FRA., Jan 2014, Zürich, Switzerland. pp.03002, 10.1051/itmconf/20140303002 . hal-01315184

HAL Id: hal-01315184

<https://hal.science/hal-01315184>

Submitted on 17 May 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Organisms modeling: The question of radial basis function networks

Alexandre Muzy^{1,a}, Lauriane Massardier^{2,b}, and Patrick Coquillard^{2,c}

¹IS UMR CNRS 7271, CS 40121 - 06903 Sophia-Antipolis CEDEX - France.

²ISA UMR 7254 INRA - CNRS - Université de Nice-Sophia Antipolis, 400 route des Chappes, BP 167, 06903 Sophia Antipolis - France.

Abstract. There exists usually a gap between bio-inspired computational techniques and what biologists can do with these techniques in their current researches. Although biology is the root of systems-theory and artificial neural networks, computer scientists are tempted to build their own systems independently of biological issues. This publication is a first-step re-evaluation of an usual machine learning technique (radial basis function(RBF) networks) in the context of systems and biological reactive organisms.

1 Introduction

Connectionist approaches are well defined and established to achieve supervised learning and classification of real-case data. The goal of this manuscript is to reevaluate these techniques in the context of activity [1] and reactive organisms [2].

The behavior of reactive organisms is considered as a set of input/output pairs. The structure of such organisms consists of an usual sensor-to-actuator network where each node can be an analogy of a gene or a sensory neuron. These analogies between artificial and biological entities aim at defining an intuitive and simple way to digitally mimic the behavior of organisms while keeping a certain level of biological structural plausability. Also, it can be considered as a biological application and so perspective of usual recent machine learning techniques [3].

The rest of the manuscript is organized as follows: Section 2 introduces sensory neurons and genes as nodes of a radial basis function (RBF) network for reactive organisms. Section 3 presents an RBF network and corresponding calibrating algorithm. Section 4 tries to extend the limits of this model to ecological applications. Finally, a conclusion sums up the article and embeds RBF model in a computational framework to be further tested.

2 Genes, neurons, nodes and actions

In this section, analogies between reinforcement learning, artificial neural networks, optimization, neuroscience and genetics are explored to define a coherent computational framework for implement-

^ae-mail: alexandre.muzy@cnrs.fr

^be-mail: lauriane.massardier@sophia.inra.fr

^ce-mail: patrick.coquillard@unice.fr

ing the reactions of organisms to different signals from the environment. Machine learning techniques and code are mainly extracted/modified from [3].

2.1 Systemic decomposition and genetic analogy

Actions of organisms correspond to behaviors that can be assimilated to phenotypes [2]. Phenotypes are resulting from genes' expression. and genes are activated (respec. inhibited) by one (or more) signal from the environment. Actions aim at exploiting resources (energy) necessary for the organism to live and reproduce. A feedback loop is established between environment/resources and organism. From a system perspective, behavior is a function of inputs (of the environment) that produces outputs (actions). A direct analogy can then be drawn between usual structures/behaviors systemic aspects and genes/phenotypes biological aspects. Figure 1 presents the analogy between biological and dynamic systems.

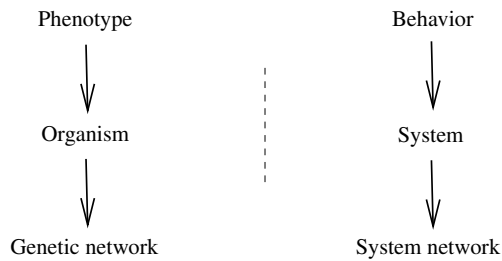


Figure 1. Analogy between biological and dynamic systems. Each arrow corresponds to a specification link, in the sense that more information is known about the structure of the system.

Genes activation depends on environment inputs. For a particular input pattern, some genes will be activated, others will not. For those activated, they exhibit different activation levels characterizing the reactivity of the organism to the state of its surrounding environment. Modeling the reactivity of organisms a major question arises: What are the values of the environment the organism detects and what are the related actions?

2.2 Environment sensing and neuron analogy

In neuroscience, the reactivity of sensory neurons is modeled using *receptive fields*. A receptive field consists of the area/space where a stimulus leads to the activation of particular sensory neurons. For a particular location on the receptive field, particular neurons are activated. A receptive field consists of the space of values activating the sensory components (neurons) of the system (the organism). Each neuron then has a particular activation range - corresponding to a particular area in the space of input values.

From the genetic perspective, the problem then is to determine the contribution of all genes to a phenotype (or action) according to the activation range of genes. Usually, the activation function of genes is represented by a *saturation function*.

In the following, neuron and gene entities are considered as nodes of a network corresponding to the organism. Activation of nodes is modeled using Gaussian functions. These functions can be implemented in a *radial basis function (RBF)* network:

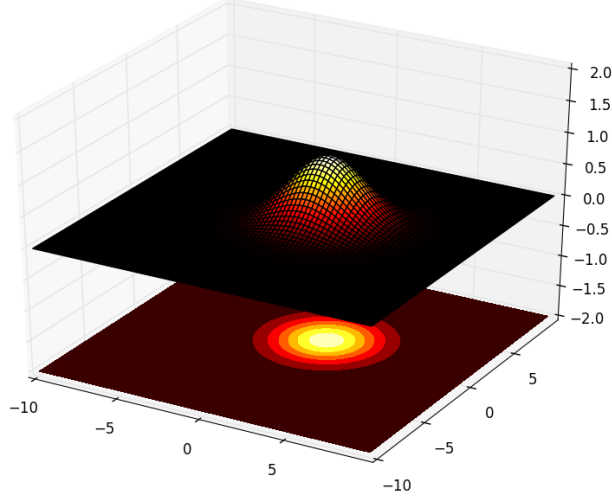


Figure 2. 2D and 3D RBF kernel representation for $\sigma_k = 2$ and $w_{k,1} = w_{k,2} = 1$.

$$g_k(\mathbf{x}, \mathbf{w}_k, \sigma_k) = \exp\left(\frac{-\|\mathbf{x} - \mathbf{w}_k\|}{2\sigma_k^2}\right) \quad (1)$$

where k is an *RBF node kernel*, \mathbf{x} is an *input vector*, \mathbf{w}_k is the *vector of weights* of node k (to every input component $x_i \in \mathbf{x}$ corresponds a weight $w_i \in \mathbf{w}_k$), $\|\mathbf{x} - \mathbf{w}_k\|$ is the *Euclidian distance between input and weight vectors*, and σ_k is the *activation width* (or standard deviation) of node k (e.g., the velocity parameter of a gene).

Equation 1 can be used in normalized form (whose shape is equivalent to saturation functions of genes or softmax functions used in neural networks and reinforcement learning):

$$g_k(\mathbf{x}, \mathbf{w}_k, \sigma_k) = \frac{\exp(-\|\mathbf{x} - \mathbf{w}_k\| / 2\sigma_k^2)}{\sum_{i=1}^n \exp(-\|\mathbf{x} - \mathbf{w}_i\| / 2\sigma_k^2)} \quad (2)$$

where n is the *number of RBF nodes*.

At each node, the distance between any input and node's weights represents its activation level. The closer input values are to node's weights, the more the node is activated. Also, the activation depends on gaussian width (controlled by the standard deviation σ_k). Therefore, as input and weight spaces are of same dimension and in same units, both spaces are equivalent. Then, it is convenient to represent nodes positions in the *weight column space* $C(\mathbf{W}_1) = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_m]$ with \mathbf{W}_1 the *matrix of input weights* of elements $w_{k,i}$ with n rows (corresponding to the number of nodes) and m columns (corresponding to inputs) and $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_m \in \mathbb{R}^m$. Figure 5 represents an example of positions of a RBF node with *activation width* $\sigma_k = 2$, a two-dimension input vector $\mathbf{x} \in \mathbb{R}^2$ and corresponding weight components $w_{k,1} = w_{k,2} = 1$. Figure 2 represents an example of RBF kernel in 3D and 2D.

2.2.1 Example: Two input sensors with uniform nodes activation

The width value of nodes can be set to $\forall k, \sigma_k = \frac{d}{\sqrt{2n}}$, where d is the *maximum distance between the locations of the two extrema nodes* and n is the number of nodes.

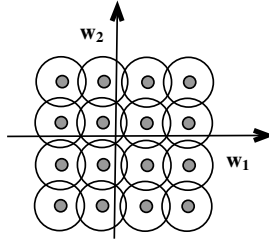


Figure 3. Activations cover uniformly the weight space.

2.3 Action selection and reinforcement learning analogy

In reinforcement learning, the same kind of formula as Equation 2 is used for action selection, i.e., an action a is chosen on the th play with probability [4]:

$$P(a) = \frac{\exp(Q_t(a)/\tau)}{\sum_i \exp(Q_t(i)/\tau)} \quad (3)$$

where, $Q_t(a)$ is the expected reward from action a , i.e., $Q_t(a) = \frac{r_1 + r_2 + \dots + r_p}{p}$, if at the th play action a has been chosen p times prior to t , yielding rewards r_1, r_2, \dots, r_p ; τ is equivalent to $2\sigma_k^2$ in Equation 2, it is a positive parameter called the *temperature*. High temperatures cause the actions to be all (nearly) equiprobable (when $\tau \rightarrow \infty$). Low temperatures cause a greater difference in selection probability for actions that differ in their value estimates. When $\tau \rightarrow 0$, only one action has greater probability and can be selected.

3 Reactive organism network

Organism reaction can now be represented as an RBF-network (cf. Figure 4). *Output layer level* is defined with: \mathbf{G} the *matrix of the activations of RBF nodes* (where each element g_{ik} corresponds to the *activation of RBF node k for input $x_i \in \mathbf{x}$*); \mathbf{W}_0 the *matrix of output weights* (where each element w_{kj} is the *weight between RBF node k and output $y_j \in \mathbf{y}$*), it is a matrix of n rows (corresponding to the number of nodes) and p columns (corresponding to the number of inputs). The *output of the network* then consists of $\mathbf{Y} = \mathbf{GW}_0$. However, since the *target outputs \mathbf{t}* are known, it is possible to analytically compute the *output weights of the network* as $\mathbf{W}_0 = \mathbf{G}^+ \mathbf{t}$, with \mathbf{G}^+ the *pseudo inverse matrix* of \mathbf{G} .

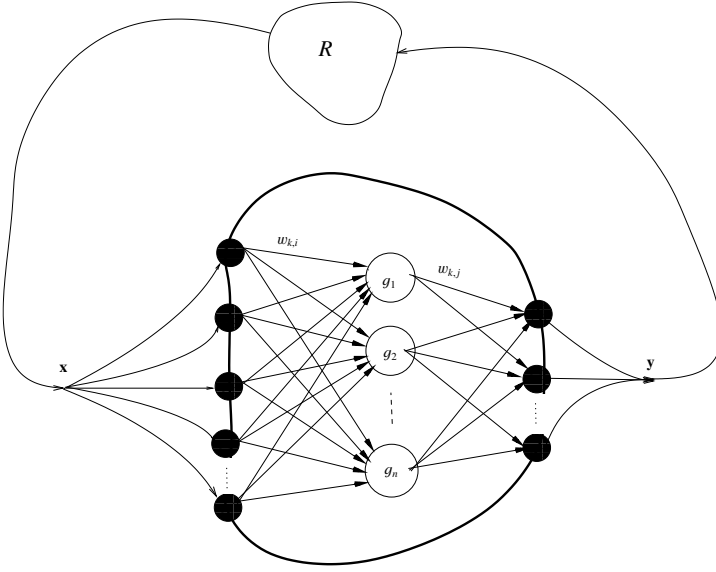


Figure 4. Structure of the organism system interacting with a resource R .

According to the value \mathbf{x} of the environment, output \mathbf{y} consists simply of a vector of boolean values whose components correspond to the achievement of a particular action:

- Only one action a_j (corresponding to a component y_j of output vector \mathbf{y}) can be achieved, e.g., $\mathbf{y} = (1, 0, 0)$ corresponds to the achievement of action a_1 , or
- A set of actions $\{a_j\}$ (corresponding to components $\{y_j\}$ of output vector \mathbf{y}) can be achieved jointly, e.g., $\mathbf{y} = (1, 0, 1)$ corresponds to the achievement of actions a_1 and a_3 .

In the RBF-network, *hidden layer* consists of RBF nodes to find a non-linear representation of inputs while *output layer* constitute a linear combination of hidden nodes achieving action classification. The problem can now be decomposed in two sub-problems: (i) for the hidden layer: find the centres (weights) of the nodes and the value of the activation width (σ), (ii) for the output layer: find the weights.

At *hidden layer level*, the whole range of inputs should be captured through the activation of the hidden nodes. Therefore, regularities have to be found in the different input values. Finding the positions of the nodes can be implemented using an unsupervised *k-means algorithm*. Finally the hybrid Algorithm 1 is obtained.

Algorithm 1 Radial basis functions hybrid algorithm (modified from [3]).

- 1: **Initialization** of RBF nodes ▷ use Calinski's criterion
 - 2: **run** k-means to initialise the positions in weight space
 - 3: **assign** uniform nodes activation: $\forall k, \sigma_k = \frac{d}{\sqrt{2n}}$
 - 4: **calculate** the actions of the RBF nodes using Equation 1
 - 5: **train** the output weights using the pseudo-inverse of the activations of the RBF centres
-

4 Organisms as optimal adaptive systems

4.1 Non-uniform activation of nodes

Contrary to RBF networks used as universal approximators, an organism cannot be sensible to any environment inputs. A trade-off should be achieved between the expected reward (resources) corresponding to an action engaged for a particular environment and the internal resources (energy) of the organism. This leads to non-uniform activations of nodes as described in this example.

4.1.1 Example: Two input sensors with non-uniform nodes activation

Non-uniform nodes activation requires modifying Algorithm 1 and leads to many *complex* questions:

- How to set the activation width σ_k of Gaussians (corresponding to, e.g., gene's *velocity parameter*)?
- Considering an unlimited external resource, how the resource acquired (energy) is distributed among nodes (e.g., genes)?
- What would be a plausible target set of environment inputs and organism's output actions?
- In uniform nodes activation, the goal of the RBF network is to be as close as possible to the target function. Then, Calinski's criterion can be used to determine the optimal number of nodes to minimize error. In non-uniform nodes activation, how to link error to the number of nodes? In other words what is the impact for the organism to do not re-act to some environmental signals?

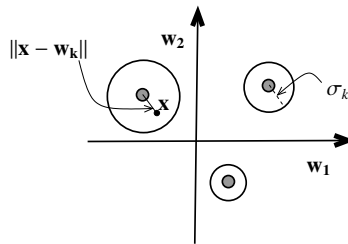


Figure 5. Activations cover non-uniformly the weight space.

4.2 Marginal cost

In Economics theory, an interesting notion called *marginal cost* allows deciding when to stop the production of a good to *optimize* the production process. “*Marginal cost* is the change in total costs from increasing output by one extra unit”. Formally, marginal cost C_m depends on both variations of *total cost* C_T and *quantity of units* produced q :

$$C_m = \frac{dC_T}{dq}$$

Marginal cost is relatively high at small quantities of output; then as production increases, marginal cost declines, reaches a minimum value, then rises. The *marginal cost* is shown in relation

to *marginal revenue*, the incremental amount of sales revenue that an additional unit of the product or service will bring to the firm. This shape of the marginal cost curve is directly attributable to increasing, then decreasing marginal returns (and the *law of diminishing marginal returns*: the decrease in the marginal (per-unit) output of a production process as the amount of a single factor of production is increased). In the law of diminishing marginal returns, first actions are usually of maximum immediate profit. Secondary actions are usually achieved only when necessary. This is coherent with a malthusian view of the problem: An other way of saying the same thing is that if the population increases the resource decreases and the costs increase.

As for marginal costs, formally, marginal revenue R_m depends on both variations of *total revenue* R_T and *quantity of units* produced q :

$$R_m = \frac{dR_T}{dq}$$

Marginal profit is the difference between marginal profit and marginal cost:

$$P_m = R_m - C_m$$

Best scenario is $P_m = 0$ (Figure 6)

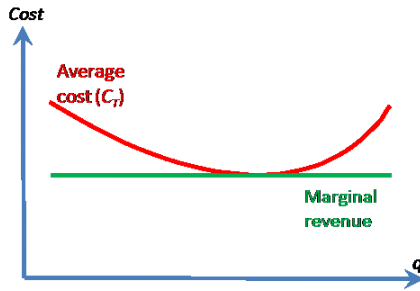


Figure 6. Cost and marginal revenues as functions of quantities.

In a biological context, an additional production unit should be equivalent to an additional gene. In previous works, we included in C_T the cost of plasticity of genes in addition to the cost of production, this leading us to minimize the ratio $\frac{C_T}{R_T}$ which coincides with a particular value of the energy Z used by the system. Let us denote Z^* this optimal value. After optimization of the system (as described in [2]), the curves of *Costs* and *Scores* in the space of Z show that at $Z^* \approx 1.25$ the two tangent lines cross the abscissa at the same point leading to the relation:

$$\frac{E'(Z^*)}{E(Z^*)} = \frac{S'(Z^*)}{S(Z^*)} \quad (4)$$

Equation (4) indicates that in the vicinity of Z^* a variation of the energy Z corresponds to a variation of the scores S which induces an almost proportional variation of the cost E in such a way that gains or losses are nearly negligible. Consequently Z^* defines a pseudo-equilibrium.

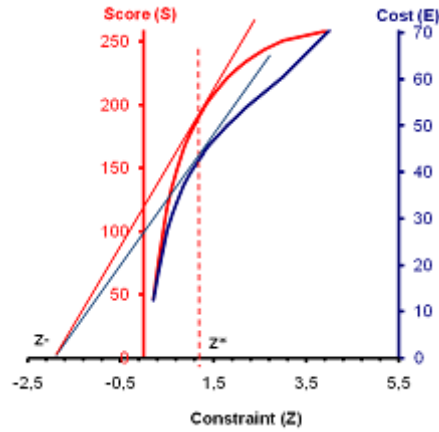


Figure 7. Scores and costs of a 3 genes system responses, in the space of the energy Z .

4.2.1 Example: Fur seals in Kerguelen islands

From Figure 7 it can be deduced that the calculation of the *Cost/Score* ratio leads to a convex function of Z with Z^* the abscissa of the minimum. As an example of this calculation, the variation of the body size of antartic fur seals has been simulated. Few days after having given birth, females of this species leave their colony (settled on Kerguelen island) and start foraging (small lantern fishes) for feed in the south of Indian Ocean. Foraging needs to be done as quickly as possible to return on time and feed pups by lactation. Over an attendance period of 120 days, the fur seals has to make an average number of 17 trips. The *Cost/Fitness* ratio has been computed from simulation results as a function of the *distances* the fur seals have to travel to find feedings, and their *body length*. Figure 8 shows that for each distance they have to travel there exists an optimal *Energy/Fitness* ratio, where $Fitness = (1 - D_m)(1 - D_p)$ is the probability for a female to successfully raise its pup with D_m and D_p the respective *probabilities of death of mothers and pups*.

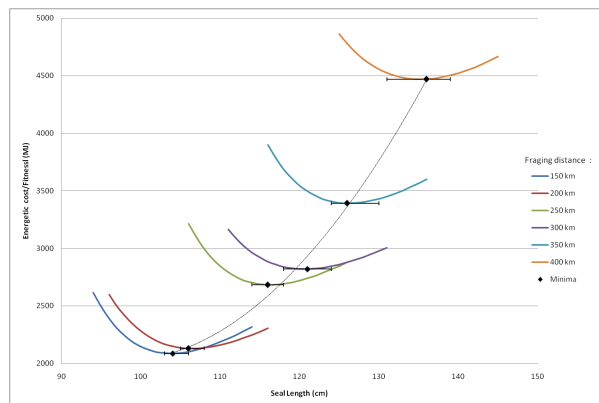


Figure 8. Optimal body size of simulated antartic fur seals.

5 Conclusion

The modeling of reactive organisms has been discussed using the connectionist approach of RBF networks. Analogies with genes and neurons have been pinpointed. The use of RBF networks in a bio-artificial framework proved to require pushing RBF usual boundaries (of universal approximators) to include input-output error control.

Activity levels of nodes is used at sensory level. This activity is equivalent to the strength of input signals of sensory neurons and can be converted into a latency [5]. Then, for each sensory signal of input types, equivalent (inversely proportional) latencies can be computed achieving an activity-to-latency conversion. This conversion leads to single spike neurons much more efficient than pulse train neurons for timed decisions. Finally, a reverse latency-to-activity conversion can be achieved for determining output actuator activations. Activity of both effectors and nodes are then a direct measure of the energy consumed by the organism for computing the *Cost/Benefit* ratio (*Benefit* being the resource acquired). This balance proved to be the driving force constraining the metabolism of organisms [2] interacting with their fluctuating environment in an evolutionary context. The minimization of this ratio can be used for the exploration of the parameter values of metabolic structures under partial information (neurons, genes, etc.) in real-case experiments.

Acknowledgments

This work was partially funded by the Centre National de la Recherche Scientifique (CNRS) and the Provence-Alpes-Cote d’Azur région (France).

References

- [1] A. Muzy, B.P. Zeigler, *Activity-based credit assignment (ACA) in hierarchical simulation*, in *SpringSim (TMS-DEVS)* (2012), p. 5
- [2] P. Coquillard, A. Muzy, F. Diener, *Ecological Modelling* **242**, 28 (2012)
- [3] S. Marsland, *Machine Learning: An Algorithmic Perspective*, 1st edn. (Chapman & Hall/CRC, 2009), ISBN 1420067184, 9781420067187
- [4] R.S. Sutton, A.G. Barto, *Introduction to Reinforcement Learning*, 1st edn. (MIT Press, Cambridge, MA, USA, 1998), ISBN 0262193981
- [5] L. Booker, S. Forrest, M. Mitchell, R. Riolo, eds., *Discrete event abstraction: an emerging paradigm for modeling complex adaptive systems* (Oxford University Press, Sante Fe Institute, 1989)