Competitive Repetition-suppression (CoRe) Learning

Davide Bacciu[1,2] and Antonina Starita[2]

[1] IMT Lucca School for Advanced Studies,
Via San Micheletto 3, 55100 Lucca, Italy
davide.bacciu@imtlucca.it,
http://www.imtlucca.it

[2] Dipartimento d'Informatica, Università degli Studi di Pisa,
Largo B. Pontecorvo 3, 56127 Pisa, Italy
starita@di.unipi.it,
http://ciml.di.unipi.it/

Abstract. The paper introduces Competitive Repetition-suppression (CoRe) learning, a novel paradigm inspired by a cortical mechanism of perceptual learning called repetition suppression. CoRe learning is an unsupervised, soft-competitive [1] model with conscience [2] that can be used for self-generating compact neural representations of the input stimuli. The key idea underlying the development of CoRe learning is to exploit the temporal distribution of neurons activations as a source of training information and to drive memory formation. As a case study, the paper reports the CoRe learning rules that have been derived for the unsupervised training of a Radial Basis Function network.

1 Introduction

The present work introduces a novel learning algorithm inspired by a cortical mechanism of perceptual learning called repetition suppression. The fundamental aspects of this cortical mechanism have been modeled in a competitive learning schema, the *Competitive Repetition-suppression (CoRe) Learning*, for the unsupervised generation of compact neural representations of the input stimuli.

We propose the idea that mere stimuli repetition acts as a fundamental resource for efficient memory formation. In particular we explore the neurophysiological hypothesis [3] that the repetition suppression mechanism serves as an unsupervised mean for reducing the size of stimuli representation, that is the number of neurons coding a given stimulus, while strengthening the responses of the most selective neurons, i.e. those showing sharp responses to particular classes of input stimuli.

For the sake of the present paper, we focus on the application of the proposed model to the automated construction of a Radial Basis Function Network (RBFN) [4]. In particular, we show how CoRe learning can be used to optimize the number, position and shape of RBFN gaussian kernels by means of a completely unsupervised training procedure.

S. Kollias et al. (Eds.): ICANN 2006, Part I, LNCS 4131, pp. 130–139, 2006.

In the following sections we introduce the neurophysiological findings that have inspired this work together with the computational model of CoRe learning. The paper ends with the results of the tests conducted on the Iris classification dataset.

2 Neurophysiological Foundations of Repetition Suppression

The ability of humans and animals to learn from experience is recognized to be supported by multiple memory systems with different functional characteristics and neural basis. Our work has been inspired by the characteristics of a particularly interesting learning scheme, called *perceptual learning* [5]. Perceptual learning supports the formation of non-declarative memory at the level of the visual cortex, providing a mean for improving the performance on several sensory tasks following practice.

In particular, we focused on a phenomenon known as *repetition suppression* (RS), which appears to be fundamental for mediating perceptual learning [3]. Repetition suppression induces long-lasting changes to the visual cortex, decreasing the neural activity as a consequence of the repeated presentation of similar stimuli. This cortical mechanism is strictly item-specific and appears at an abstract representational level. For instance, neurophysiological evidences have shown that its effects can be observed also when the repeated stimulus is presented at different retinal locations and in case of variations to the visual stimulus geometry. Repetition suppression does not depend on the behavioral significance of the stimuli, i.e. it is not specifically linked to the active maintenance of a sample in memory, nor it requires any form of response/reward signal. Furthermore, RS is an unconscious process, since its effects can be recorded also in anesthetized subjects [3].

In brief, repetition suppression involves sharpening the neural representation of items by means of an overall reduction of the number of active neurons which is counterbalanced by the steepening of the response of the most item-selective neurons. This process seems to be aimed at the selection of neurons that act as detectors of the most informative features. Moreover it appears that such a process may facilitate novelty detection, since more familiar stimuli experience more suppression than unfrequent items.

We suggest that repetition suppression, and in general perceptual learning, may provide interesting hints for the development of innovative learning schemes and mechanisms. The literature in this field offers very few works that try to exploit the neurophysiological issues described so far. Mozer [6] proposes a computational model reproducing the repetition suppression phenomenon by means of a network of binary-hypothesis neurons which uses blind equalization to suppress the irrelevant inputs and to enhance the most active neurons, by steepening the curve of the sigmoid driving their responses. Another interesting model is that proposed by French and called *activation sharpening* [7]. This algorithm extends the standard backpropagation with an extra step in which activations patterns at the hidden layer are sharpened, i.e., the activation level of the most active hidden nodes is increased slightly for each pattern, while the other nodes activations are decreased. The activation sharpening process, although not mediated by repetition, offered an interesting starting point for the development of our repetition suppression-based model. However, the learning scheme we propose is intended to have a broader scope than those described so far and to be applicable to a wide range of neural networks and learning systems on real world tasks (e.g. machine vision).

3 Competitive Repetition-suppression (CoRe) Learning

3.1 A Soft-Competitive Approach to Learning

The general idea underlying the proposed model is to make the neural population evolve in the direction of maximum selectivity by means of a procedure that penalizes or enhances the responses of the neurons on the basis of the stimuli frequency. This approach falls into the family of the competitive learning [8] schemes, in which units compete to be active during training. Hard-competitive approaches such as Winner Takes All (WTA), allow only the winning unit, i.e. the one with the highest activation, to learn on each case. The soft-competitive approach [1], on the other hand, allows each unit, or the units from a selected subset, to adapt its weights in proportion to its activation strength.

CoRe learning falls into the family of soft-competitive approaches. At each step, it selects the most active neurons to form the *winners pool*, while the remainder of the units forms the *losers pool*. We define the winners pool for the input x_k as the set of units u_i that fires more than θ, that is

$$win_k = \{i \mid y_i(x_k) \geq \theta,\ u_i \in U\} \tag{1}$$

where $y_i(x_k)$ is the activation of the i-th unit on the k-th input pattern. The definition of the losers pool can be obtained by mirroring the inequality in (1). The competition is engaged between the units of two pools: the winners gets rewarded and their activity is strengthened, while the losers are penalized depending on the amount of the repetition suppression generated. In this aspect, CoRe extends the *rival penalized competitive learning* (RPCL) algorithm [9]. The key idea of RPCL is that for each stimulus not only the winner is learned to approach the input pattern, but also the second winner (the rival) is de-learned away from it for a bit. We extended this approach by defining a *soft-rival* algorithm, in which winner and rival refer to pools of units.

3.2 CoRe Learning

The primary issue for implementing a repetition suppression learning scheme is modeling the stimuli repetition. We build our scheme on a parameter, the

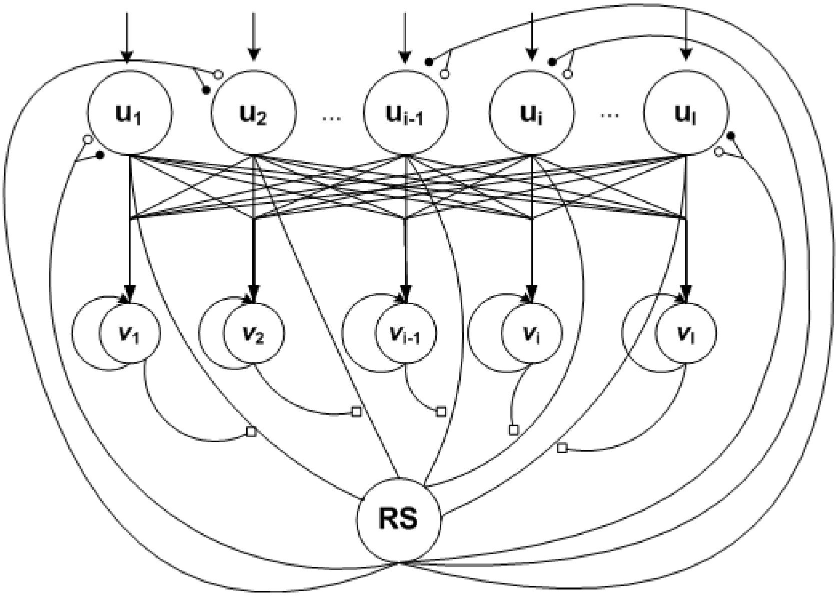


Fig. 1. General form of a CoRe learning layer of neurons: units u_i are the feature detectors; ν_i accumulates the conscience [2] related to unit u_i, while the RS unit generates the repetition suppression effect which inhibits (*empty-dot connections*) or enhance (*filled-dot connections*) the feature detectors.

stimulus predominance, that gives a soft measure of the stimuli frequency. The stimulus predominance at the time t is defined as

$$\nu_i^t = \frac{1}{|\chi_t|} \sum_{x_k \in \chi_t} \frac{y_i(x_k)}{z_U^k}, \tag{2}$$

where χ_t is the set of the input stimuli presented to the network up to time t, while $y_i(x_k)$ is the output of the i-th unit on the k-th input pattern and z_U^k is (an approximation of) the output of the maximally active unit, from the set U, on the pattern x_k. For instance, it can be approximated, by means of the softmax, as

$$z_U^k = \sum_{u_j \in U} y_j(x_k) \frac{e^{q y_j(x_k)}}{\sum_l e^{q y_l(x_k)}}, \tag{3}$$

where q is a parameter which regulates the sharpness of the approximation.

The idea underlying the definition in (2) is to associate each unit $u_i \in U$ with a prototype p_i that identifies the reference stimuli for the neuron. Then, each unit uses eq. 2 to measure the frequency with which stimuli similar to the prototype p_i have been shown to the network. Here we assume that the unit output y_i is an increasing function of the similarity between the input vector x_k and the unit's prototype p_i. This approach resembles competitive learning with *conscience* [2]. The conscience mechanism was proposed for making frequently winning representatives less likely to win in the future because of their heavier conscience [10]. In our model we use a different conscience mechanism for suppressing the responses of the less selective neurons. Bienenstock, Cooper and Munro tackled with the selectivity issue in their notable BCM model [11]. Their Hebb-like learning rule is aimed at training neurons with a maximal response on one particular pattern, while retaining a very low response on the other patterns.

CoRe learning pursues neuron selectivity by defining a penalty factor, the repetition suppression, which is mediated by the stimulus predominance of the winning neurons. The amount of repetition suppression generated at time t in response to the pattern x_k is calculated as

$$RS_k^t = \frac{1}{|win_k|} \sum_{i \in win_k} \nu_i^t y_i(x_k), \tag{4}$$

where the winners pool win_k is calculated as in (1).

Figure 1 gives a visual interpretation of the model. The ν_i units accumulate the history of the feature detector (u_i) activations in order to generate the stimulus predominance for the prototype p_i. The activation of the winning neurons, scaled by their stimulus predominance (the empty-square connections in figure 1), is then used to generate the repetition suppression factor in the RS unit.

The neurons in the losers pool have a different learning rule with respect to those in the winners pool, although in both cases we define a pseudo-target to be used as a reference signal for the training procedure. The target activation for neurons in the losers pool ($u_i \in lose_k$) is defined as $\hat{y}_i^t(x_k) = y_i(x_k)(1 - RS_k^t)$, where x_k is the current input pattern. This expression forces the loser neurons to shrink their activation proportionally to the amount of repetition suppression they receive. The representation error of the i-th loser can be written as

$$\underline{E}_{i,k}^t = \frac{1}{2}(\hat{y}_i^t(x_k) - y_i(x_k))^2 = \frac{1}{2}(-y_i(x_k) RS_k^t)^2. \tag{5}$$

Conversely, the target activation for the neurons $u_i \in win_k$ is set to 1 in order to strengthen their activation (assuming 1 to be the the maximal output of a neuron). The representation error, in this case, is simply

$$\overline{E}_{i,k}^t = \frac{1}{2}(1 - y_i(x_k))^2. \tag{6}$$

The network parameters can be adapted by means of a supervised learning algorithm that minimize the error functions defined in (5) and (6). In section 3.3 we show an example of a Radial Basis Function Network whose gaussian units have been trained by CoRe learning and gradient descent. Notice that although CoRe learning resort to supervised learning for training the network parameters, it is a completely unsupervised algorithm since all the reference signals it uses are self-generated on the basis of the input stimuli distribution over space and time.

The repetition suppression phenomenon produces a compact neural representation by evolving a set of highly selective neurons from a large pool of units. Hence, it is important to define a metric for identifying the most significant neurons which have been produced by the learning process. We define the *relevance factor* for the unit u_i as

$$\hat{\nu}_i^t = \frac{1}{|win_{u_i}^t|} \sum_{x_k \in win_{u_i}^t} \frac{y_i(x_k)}{z_{win_k}^k}, \tag{7}$$

where $z_{win_k}^k$ follows the definition in (3) and $win_{u_i}^t$ is the set of patterns $x_k \in \chi_t$ for which unit u_i was in the winners pool, i.e.

$$win_{u_i}^t = \{x_k \mid y_i(x_k) \geq \theta,\ x_k \in \chi_t\}. \tag{8}$$

In other words, the relevance factor defines a soft-measure of the frequency with which u_i was the most active unit in the winners pool. This measure can be used to prune those neurons which are less significant for the stimuli representation (e.g. see RBF pruning in section 3.3).

3.3 RBFN Structure Optimization

A radial basis function network can be interpreted as a composition of localized receptive fields that measure the similarity of incoming patterns x_k to the prototype p_i they represent. In case of gaussian basis functions, the activation of the i-th RBF unit is defined as

$$y_i(x_k) = \frac{1}{2} e^{\frac{\|x - c_i\|^2}{\sigma_i^2}} \tag{9}$$

where the gaussian center c_i corresponds to the prototype vector p_i, while the gaussian variance σ_i modulates the steepness of the units' response.

We applied CoRe learning to solve the structure optimization problem of a gaussian RBFN, i.e, defining number, position and shape of the radial basis functions. In order to do this, we used gradient descent to derive the CoRe learning rules for the parameters of the gaussian kernels. The parameter increments for the units $u_i \in lose_k$ can be derived by differentiating the error function in (5) with respect to the parameters c_i and σ_i, that is

$$\triangle \underline{c}_{i,k}^t = \frac{\partial \underline{E}_{i,k}^t}{\partial c_i} = -y_i RS_k^t \frac{\partial(-y_i RS_k^t)}{\partial c_i} = \left(\frac{y_i RS_k^t}{\sigma_i}\right)^2 (x_k - c_i) \tag{10}$$

$$\triangle \underline{\sigma}_{i,k}^t = \frac{\partial \underline{E}_{i,k}^t}{\partial \sigma_i} = -y_i RS_k^t \frac{\partial(-y_i RS_k^t)}{\partial \sigma_i} = (y_i RS_k^t)^2 \frac{\|x_k - c_i\|^2}{\sigma_i^3}. \tag{11}$$

Similarly, the parameter increments for the units $u_i \in win_k$ can be calculated as

$$\triangle \overline{c}_{i,k}^t = \frac{\partial \overline{E}_{i,k}^t}{\partial c_i} = -(1 - y_i) y_i \frac{(x - c_i)}{\sigma_i^2} \tag{12}$$

$$\triangle\overline{\sigma}^{t}_{i,k} = \frac{\partial \overline{E}^{t}_{i,k}}{\partial \sigma_i} = -(1-y_i)y_i\frac{\|x_k - c_i\|^2}{\sigma_i^3} \tag{13}$$

where $\overline{E}^{t}_{i,k}$ follows the definition in (6).

The update rules for the RBF parameters are

$$c_i^t = c_i^{t-1} - \alpha_c \triangle c^{t}_{i,k} \tag{14}$$

$$\sigma_i^t = \sigma_i^{t-1} - \alpha_\sigma \triangle \sigma^{t}_{i,k} \tag{15}$$

where $\triangle c^{t}_{i,k} = \triangle \overline{c}^{t}_{i,k}$ if $u_i \in win_k$ and $\triangle c^{t}_{i,k} = \triangle \underline{c}^{t}_{i,k}$ if $u_i \in lose_k$ (similarly for $\triangle \sigma^{t}_{i,k}$).

The sign of the increments $\triangle c^{t}_{i,k}$ and $\triangle \sigma^{t}_{i,k}$ is coherent with the expected repetition suppression effect. Units in the losers pool, for instance, experience the displacement of their centers away from the current input as well as the enlargement of their widths. Conversely, winner neurons have their centers moved closer to the current stimuli and their responses steepened by the reduction of their receptive field's width.

CoRe learning starts with a large RBF network and incrementally trains the RBF units at each pattern presentation. If an input parameter x_k does not produce a sufficient activation in any of the units in the network, then CoRe learning triggers a search procedure to find the neuron with the lowest relevance factor $\hat{\nu}_i^t$ and trains this unit to memorize the stimulus x_k. Moreover, at the end of each learning epoch, the least significant neurons, i.e those with a relevance factor under a certain threshold θ_{pr}, are pruned from the network. This allows to generate networks with a compact structure, while retaining an high representational power.

The outputs of the N RBF units are linearly combined in the output units o_j by the weighted summation

$$o_j(x_k) = \sum_{i=1}^{N} w_{ij} y_i(x_k). \tag{16}$$

The linear parameters w_{ij} are trained by a supervised algorithm independently and in parallel to CoRe learning [12].

4 Results

We evaluated the performance of CoRe learning on a classification task based on the IRIS dataset [13]. This dataset contains 150 samples of dimension 4 equally partitioned into three different IRIS classes: setosa, versicolor, and virginica. Among these, one is linearly separable from the others, while two of them are not. The dataset was split into a training and a test sets, with 75 samples each; 25 samples from each class were randomly selected to be presented to the network and thirty different partitions of the dataset were generated randomly.

Table 1. Performance of various RBF models applied to the IRIS problem: mean classification score on the test set, variance, number of initial RBF (when applicable) and average number of RBF generated during training

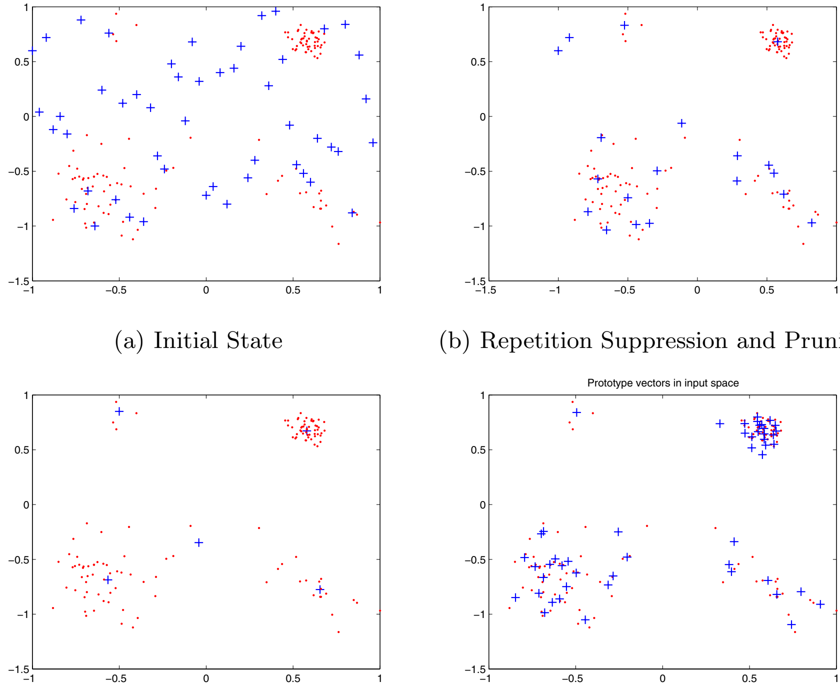
Model	Mean Score	Variance	Initial RBF Number	Final RBF Number
CGA-RBF [14]	97.04%	$\pm 1.97\%$	N.A.	6.4
CoRe-RBF	95.91%	$\pm 1.19\%$	50	6.6
RBF-DDA [15]	94.50%	$\pm 1.50\%$	N.A.	11.8

To evaluate the capabilities of the CoRe approach we implemented a radial basis function network with CoRe learning rules for training the gaussian kernels and a standard gradient descent for adapting the output weights w_{ij}. The network was trained and tested separately on each of the thirty random partitions of the dataset.

Table 1 shows a comparison of the results of the test phase for the CoRe-RBF network and for two constructive RBF models. The CoRe-RBF network had an initial population of 50 neurons and, as a result of CoRe learning, reduced its size to an average number of 6 RBF. As it can be seen from Table 1, our model achieved an higher score than the RBF-DDA model [15] and generated a more compact network structure. The evolutive CGA-RBF [14], on the other hand, achieved better results than CoRe-RBF with respect to the test-set error. However, the CGA-RBF learning scheme needs a-priori knowledge about the number of classes, while our model is completely unsupervised and does not require any a-priori information about the distribution of the input patterns. In addition, the CoRe model achieved a very stable learning, demonstrated by the low error variance (see Table 1) over the 30 task repetitions.

Figure 2 shows the performance of CoRe learning on a clustering task: the dataset consists of 125 datapoints generated by four gaussians with different mean, variance and density. The small dots represent the 2-dimensional input stimuli, while the pluses (+) identify the neurons prototypes. The network evolves from the initial random prototype allocation (Fig. 2.a) to a sparser input coverage (Fig. 2.b) as a result of the repetition suppression mechanism. The least active units are pushed away from the stimuli, becoming less significant as learning proceeds. When their relevance factor falls behind a predefined threshold, they are pruned from the network. Conversely, the most significant units are retained and positioned as to cover the four clusters (Fig. 2.c).

The CoRe algorithm was run starting from a pool of 50 neurons and converged to a stable state after only 30 training epochs, reducing the number of prototypes to 5. It is worth noting how 4 units positioned approximately on the four clusters mean, while 1 neuron was allocated to cover the outliers of the 2 bottommost clusters. The same dataset has been used to run the RPCL algorithm (Fig. 2.d): starting from an initial population of 50 neurons, RPCL evolved to a stable state with 36 active neurons, i.e. neurons winning at least one competition. The results of the test show the robustness of CoRe with respect to the choice of the initial units allocation: CoRe is capable of converging to a compact prototype allocation even though it is run starting from an oversized neural population.



(a) Initial State

(b) Repetition Suppression and Pruning

(c) Final State

(d) RPCL Learning

Fig. 2. Clustering test on four gaussian clusters: (a) CoRe starts with a random positioning of the gaussian centers; (b) the least active units are pushed away from the stimuli and network pruning eliminates non-relevant units; (c) the CoRe algorithm converges generating 5 units positioned on the four clusters and on their outliers; (d) prototype positioning by the RPCL algorithm after 200 learning epochs

5 Conclusion

This paper introduces CoRe learning, a soft-competitive learning scheme inspired by a cortical mechanism of implicit visual memory, i.e. repetition suppression. CoRe learning allows unsupervised training of feature detector units (e.g. RBF) without resorting to any explicit information concerning the input pattern distribution (e.g. number of classes), but only on the basis of the stimuli repetition.

We derived the CoRe learning rules for training the gaussian units of a radial basis function network and we tested the effectiveness of the proposed approach on classification and clustering tasks.

The CoRe learning model is part of a work aiming at the development of an articulated model of perceptual learning for machine vision applications. In particular, we believe that the repetition suppression mechanism, and therefore the CoRe learning scheme, may constitute an important tool for generating compact and sparse neural representations of the visual stimuli.

References

1. Nowlan, S.: Soft Competitive Adaptation: Neural Network Learning Algorithms based on Fitting Statistical Mixtures. Phd thesis, Carnegie-Mellon University, Pittsburg (1991)
2. DeSieno, D.: Adding conscience to competitive learning. In: IEEE Annu. Int. Conf. Neural Networks, IEEE Computer Society (1988) 1117–1124
3. Desimone, R.: Neural mechanisms for visual memory and their role in attention. In: Proceedings of Natl. Acad. Sci. USA. Volume 93. (1996) 13494–13499
4. Poggio, T., Girosi, F.: Networks for approximation and learning. In: Proceedings of the IEEE. Volume 78. (1990) 1481–1497
5. Tdodyks, M., Gilbert, C.: Neural networks and perceptual priming. Nature **7010**(431) (2004) 775–781
6. Mozer, M.C., Mytkowicz, T., Zemel, R.S.: Achieving robust neural representations:an account of repetition suppression. Technical report, Computer Science Deparment, University of Colorado, Boulder (2004)
7. French, R.M.: Semi-distributed representations and catastrophic forgetting in connectionist networks. Connection Science **4** (1992) 365 – 377
8. Rumelhart, D., Zipser, D.: Competitive learning. Cognitive Science **9** (1985) 75 – 112
9. Xu, L., Krzyzak, A., Oja, E.: Rival penalized competitive learning for clustering analysis, rbf net, and curve detection. IEEE Transactions on Neural Networks **4**(4) (1993) 636 – 649
10. Banerjee, A., Ghosh, J.: Frequency-sensitive competitive learning for scalable balanced clustering on high-dimensional hyperspheres. IEEE Transactions on Neural Networks **15** (2004) 702 – 719
11. Bienenstock, E.L., Cooper, L.N., Munro, P.W.: Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. Neurocomputing: foundations of research (1988) 437–455
12. Karayiannis, N.: Growing radial basis neural networks: merging supervised and unsupervised learning with network growth techniques. IEEE Transactions on Neural Networks **8**(6) (1997) 1492–1506
13. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Ann. Eugenics **7**(2) (1936) 179 – 188
14. de Castro, L.N., Hruschka, E.R., Campello, R.J.G.B.: An evolutionary clustering technique with local search to design RBF neural network classifiers. In: Proceedings of the 2004 IEEE International Joint Conference on Neural Networks. Volume 3. (2004) 2083–2088
15. Paetz, J.: Feature selection for RBF networks. In: Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'O2). Volume 2. (2002) 986–990