Connection Science

ISSN: 0954-0091 (Print) 1360-0494 (Online) Journal homepage: https://www.tandfonline.com/loi/ccos20

Competitive Learning and its Application in Adaptive Vision for Autonomous Mobile Robots

Dean K McNeill & Howard C Card

To cite this article: Dean K McNeill & Howard C Card (1999) Competitive Learning and its Application in Adaptive Vision for Autonomous Mobile Robots, Connection Science, 11:3-4, 345-357, DOI: 10.1080/095400999116296

To link to this article: https://doi.org/10.1080/095400999116296

Published online: 01 Jul 2010.

Connection Science, Vol. 11, No. 3&4, 1999, 345–357

Competitive Learning and its Application in Adaptive Vision for Autonomous Mobile Robots

DEAN K. MCNEILL & HOWARD C. CARD

The task of providing robust vision for autonomous mobile robots is a complex signal processing problem which cannot be solved using traditional deterministic computing techniques. In this article we investigate four unsupervised neural learning algorithms, known collectively as competitive learning, in order to assess both their theoretical operation and their ability to learn to represent a basic robotic vision task. This task involves the ability of a modest robotic system to identify the components of basic motion and to generalize upon that learned knowledge to classify correctly novel visual experiences. This investigation shows that standard competitive learning and the DeSieno version of frequency-sensitive competitive learning (FSCL) are unsuitable for solving this problem. Soft competitive learning, while capable of producing an appropriate solution, is too computationally expensive in its present form to be used under the constraints of this application. However, the Krishnamurthy version of FSCL is found to be both computationally efficient and capable of reliably learning a suitable solution to the motion identification problem both in simulated tests and in actual hardware-based experiments.

KEYWORDS: Neural computation, visual representation, adaptive vision, autonomous mobile robots.

1. Introduction

The goal of developing effective and robust sensory processing systems for robots is, by its very nature, a complex issue. Yet this capability is essential if robotic systems are to operate effectively in real-world environments. This situation is further exacerbated if these robots are also required to function autonomously, as this places additional constraints on the hardware and software systems which may be used. For such systems it is clearly unrealistic to expect to predict in advance all potential sensory experiences which may be encountered and to produce deterministically the necessary responses to those stimuli. However, it is possible to provide a sensory framework that will permit the system to respond in a reasonable way to novel experiences and to adapt to those experiences to improve future responses. Artificial neural networks have demonstrated their ability to provide this type of adaptive signal processing when confronted with related

D. K. McNeill, Department of Electronics and Electrical Engineering, The University of Edinburgh, The King's Buildings, Edinburgh EH9 3JL, UK. Tel: +44 131 650 5661/+44 797 720 7161; Fax: +44 131 650 6554; E-mail: Dean.McNeill@ee.ed.ac.uk. H. C. Card, Department of Electrical and Computer Engineering, University of Manitoba, 15 Gillson St, Winnipeg, MB R3T 5V6 Canada. E-mail: hcard@ee.umanitoba.ca.

ISSN 0954-0091 (print)/ISSN 1360-0494 (online)/99/030345-13

problems in other areas (Gorman & Sejnowski, 1988; Gozani *et al.*, 1989; Le Cun *et al.*, 1989; Pomerleau, 1996). It is the capabilities of a class of these connectionist algorithms which we wish to explore in the context of a robotic visual system.

Traditionally, evaluation of different neural algorithms involves the comparison of metrics such as time to convergence or relative incremental improvements in minimum achievable error. These studies frequently involve the simulation of algorithms on high speed workstations possessing generous memory resources and practically infinite energy budgets. However, such measures are unrealistic when one is considering the implementation of neural algorithms in an autonomous mobile robotic system. The historically poor energy-to-weight ratio of existing battery technology severely constrains the type of hardware and algorithms that such a robot can support. It would be inappropriate to use something such as a Pentium™ processor in a small robot, as it would very quickly consume all available battery power. Instead, the use of much more modest processors is necessary, and this will have a direct impact on the complexity of the algorithms which may be used and the types of problems which may be addressed. As a result, one is forced to make trade-offs between algorithmic complexity and the resulting benefit in overall system performance which would result from their use. A marginal improvement in performance, such as lower mean squared error, would not be sufficient to justify the use of a significantly more computationally intensive technique.

In the remainder of this article we shall examine the properties of four closely related neural learning algorithms which form part of the class of unsupervised techniques known as competitive learning (CL). These will be evaluated based on their effectiveness in processing a variety of both artificially generated data distributions and actual experimental results obtained using a mobile robotic system.

2. The Algorithms

The four algorithms investigated are standard or hard competitive learning (HCL), soft competitive learning (SCL) and two variations of frequency-sensitive competitive learning (FSCL). These are all unsupervised techniques intended to extract underlying structure from a collection of unlabelled data vectors.

2.1. *Standard or Hard Competitive Learning*

The simplest competitive learning algorithm is standard or hard competitive learning (Rumelhart & Zipser, 1985). This is a winner-take-all (WTA) approach, with the winning unit ($i^\star$) being determined through the calculation of a simple distance measure between a unit's weight vector ($\mathbf{w}_i$) and the current data vector ($\mathbf{x}$).

$$h_i = \| \mathbf{w}_i - \mathbf{x} \| \tag{1}$$

The unit having the shortest distance, and therefore which most closely matches the input, is selected as the winner and updates it weights according to equation (2):

$$\Delta w_{i^\star j} = \varepsilon (x_j - w_{i^\star j}) \tag{2}$$

In the context of the results reported in this article a learning rate value of $\varepsilon = 0.001$ was used in the calculation of the weight adjustments.

While the HCL algorithm has the advantage of being computationally efficient, it unfortunately suffers from a critical shortcoming arising from the WTA nature of the weight updates. In the situation where a neuron has its weight vector

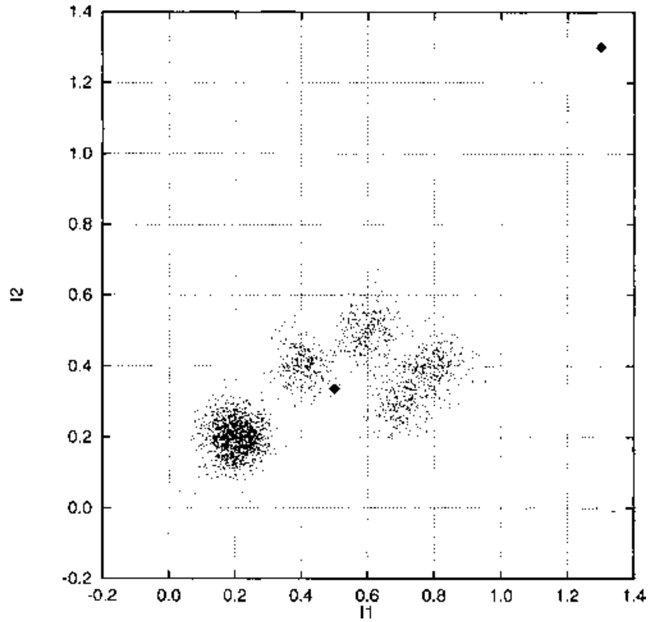


Figure 1. Hypothetical data distribution showing the orphaned unit problem prevalent in HCL.

initialized in a region remote from the data, it is extremely unlikely that that unit will succeed in winning a competition for any of the data points and will therefore not have the opportunity to update its weights. This will result in the unit being abandoned or orphaned, meaning that it will not contribute in a meaningful way to the development of a solution. This clearly results in ineffective use of available network resources and will produce very poor solutions, even in straightforward clustering problems such as that depicted in Figure 1.

2.2. Frequency-sensitive Competitive Learning

In an attempt to overcome the orphaned unit problem, DeSieno (1988) proposed the introduction of a frequency-based penalty to the learning process known as the conscience mechanism. The objective of this approach is to ensure that units which fail to win based solely on their distance from the data are, none the less, given an opportunity to update their weights occasionally as a consequence of their low winning frequency. Or conversely, those units which win often are penalized for this domination of the learning and thereby permit other units to perform weight updates part of the time. Under this scheme, each of the N units within a network layer should ultimately win $1/N$th of the competitions at the conclusion of training.

In terms of the calculation of unit activations (y_i), FSCL$_D$ operates identically to HCL. However, unlike HCL, the winning unit is not necessarily the unit which will perform a weight adjustment. That decision is made by computing an adjusted distance value (g_i) which incorporates a penalty term based on the proportion (p_i) of preceding competitions each unit has succeeded in winning.

$$g_i = \| \mathbf{w}_i - \mathbf{x} \|^2 - C\left(\frac{1}{N} - p_i\right) \tag{3}$$

The coefficient C in equation (3) is the bias factor and controls the strength of the conscience penalty. The winning proportion for each unit is adjusted following each competition as dictated by equation (4):

$$\Delta p_i = B(y_i - p_i) \tag{4}$$

Here the term B controls the sensitivity of the conscience to the winning of a single competition. DeSieno recommends that B be kept small ($1 \gg B > 0$) to ensure that the conscience responds to the global statistics of the dataset.

While the frequency-based approach is conceptually appropriate, the DeSieno technique ($\mathrm{FSCL_D}$) significantly increases the amount of computation required during learning. An alternative and simpler frequency-based approach was proposed by Krishnamurthy *et al.* (1990). This technique differs from $\mathrm{FSCL_D}$ in that the frequency dependence is introduced as a multiplicative rather than an additive component of the distance calculation. This is achieved through the use of a fairness function $F(\cdot)$, which is a monotonically increasing function of the number of competitions (u_i) each unit has suceeded in winning.

$$h_i = F(u_i)\,\| \mathbf{w}_i - \mathbf{x} \| \tag{5}$$

A typical fairness function is simply $F(u_i) = u_i$. Again, a single winning unit is determined based on the distance h_i, with the weight adjustments being performed according to equation (2). In terms of computation, the $\mathrm{FSCL_K}$ approach is clearly much less demanding than $\mathrm{FSCL_D}$.

2.3. Soft Competitive Learning

The final algorithm to be considered is SCL (Hertz *et al.*, 1991; Moody & Darken, 1989). This algorithm differs from the previous three in that the unit activations are computed using Gaussian radial basis functions (RBFs) and thereby produce analogue activities instead of binary WTA values. In the context of this article, only symmetric Gaussian units are considered.

$$y_i = \frac{e^{-\|\mathbf{w}_i - \mathbf{x}\|/(2\sigma^2)}}{\sum_k e^{-\|\mathbf{w}_k - \mathbf{x}\|/(2\sigma^2)}} \tag{6}$$

In addition to the use of analogue activations, SCL has the further advantage that each unit also performs a weight adjustment in response to each training pattern and in proportion to its activation.

$$\Delta w_{ij} = \varepsilon y_i (x_i - w_{ij}) \tag{7}$$

This has the effect of avoiding orphaned units by ensuring, even in the situation of poorly initial weights, that every unit will make small advances towards the data and will eventually contribute in a meaningful way to a solution. However, this guaranteed participation is achieved at the expense of long training times in the case of very poor initialization. Furthermore, it is clear that SCL achieves its results at the expense of a significant amount of computation, including the need to evaluate an exponential function.

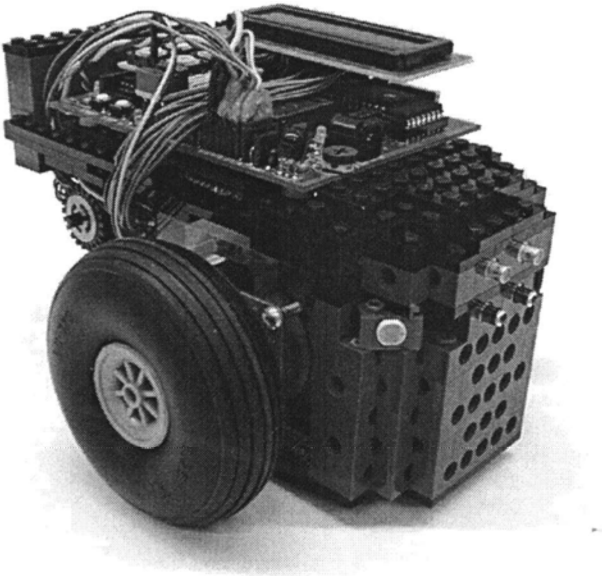


Figure 2. Autonomous mobile robot.

3. A Simple Robotic System

We now wish to examine the relative performance of the four CL algorithms and have selected a basic robotic vision task with which to perform this analysis. As has already been indicated, evaluation of the four approaches will be based on their ability to solve the given task while keeping in mind the associated computational requirements of each technique. The task selected is the unsupervised clustering of motion within the visual environment of a modest autonomous robot.

The robot developed and used for these experiments is shown in Figure 2. For convenience and durability the physical structure of the system was constructed using plastic LEGO® building bricks. Movement of the robot is achieved through a differential wheel drive powered by two independently controlled stepper motors. Control of the system is provided by a Motorola® MC68HC11-based microcontroller board developed at the Massachusetts Institute of Technology and adapted for this application. The Motorola HC11 processor is a widely used industrial 8-bit integer microcontroller with on-board analogue-to-digital conversion and operates at a conservative clock frequency of 1 MHz in this implementation. In addition to the modest computational power of this system, the total memory available for the storage of both the neural algorithm and associated data values is limited to only 32k.

Given the constraints imposed by the computational resources of the microcontroller, it is necessary that the visual apparatus also be of a modest nature to allow for real-time processing of the sensory input. While the resulting visual capabilities will be much less sophisticated than those of mammals, nature has none the less demonstrated that relatively 'simple' creatures are able to make very effective use of basic visual apparatus. The selection and arrangement of optical sensors used on the robot was inspired by the visual system of jumping spiders (salticidae) (Foelix, 1996). While all spiders receive information about the world from a rich variety of sensory apparatus, vision is particularly important to the survival of hunting spiders. In fact, in the case of jumping spiders, vision is absolutely critical to their ability to capture prey.

As with the jumping spider, the vision system of the robot is made up of six sensors of varying types. These are arranged in three pairs along the front and side of the robot as seen in Figure 2. The four forward-facing sensors are phototransistors, while the side-facing sensors are cadmium sulphide photoresistors. The lower pair of phototransistors (OP805) possesses a narrow field of view of approximately $\pm 15°$ from the centre and corresponds to the spider's main (anterior-medial) eyes. The upper pair (L14C1) possesses a moderate field of view of approximately $\pm 40°$ from the centre, corresponding to the spider's secondary (anterior-lateral) eyes. The photoresistors, which are oriented at a 45° angle to the forward surface, have a field of view of approximately $\pm 90°$ and correspond to the arachnid's secondary (posterior-lateral) eyes. It is known from biological experiments that the anterior-lateral and posterior-lateral eyes are used primarily for motion detection, while the anterior-medial eyes provide fine resolution over a narrow field of view. Figure 3 shows the measured angular response of the three pairs of robotic sensors as a function of distance from a single source.

4. Empirical Investigations of Algorithm Performance

Based on the robotic system just described, we wish to evaluate the relative performance of the four learning algorithms. This analysis was carried out initially through simulation using modelled properties of the robot's actual sensory hardware, and a subset of those results is later verified using the actual robot. To assist in the analysis, a simplified sensor geometry consisting of an array of five symmetrically arranged detectors with uniform characteristics was also investigated in simulation. Figure 4 shows a frontal view of the robotic and simplified cross-shaped sensors arrangements. The five detectors in the latter set-up each possess a $\pm 25°$ field of view.

4.1. Dataset Characteristics

A series of training and test datasets were produced to exercise and evaluate the ability of the algorithms to cluster the motion or position of a single optical source. The first of these consisted of the modelled response of the symmetric sensor structure (Figure 4(b)) when excited by a single active source from a plane of five sources also arranged in an identical cross format. Each entry in the resulting dataset thus consists of a five-dimensional input vector corresponding to these calculated analogue values from the modelled sensors. In order to more closely replicate the behavior of real sensors, a small amount of Gaussian noise was added to the calculated intensities to produce the final dataset consisting of 1000 training patterns. The corresponding test dataset was composed of the perceived noise-free response of these same sensors to a full three-by-three grid of nine sources.

In addition to the static position dataset, two motion-based test datasets were prepared corresponding to the perceived response of the robotic and cross-based sensor arrangements, respectively. In order to encode the temporal aspect of the environment the input vectors in these datasets were expanded to include both the current and a single time-delayed value from the sensor arrays. This results in a 12-dimensional input vector in the case of the robotic sensor structure and a 10-

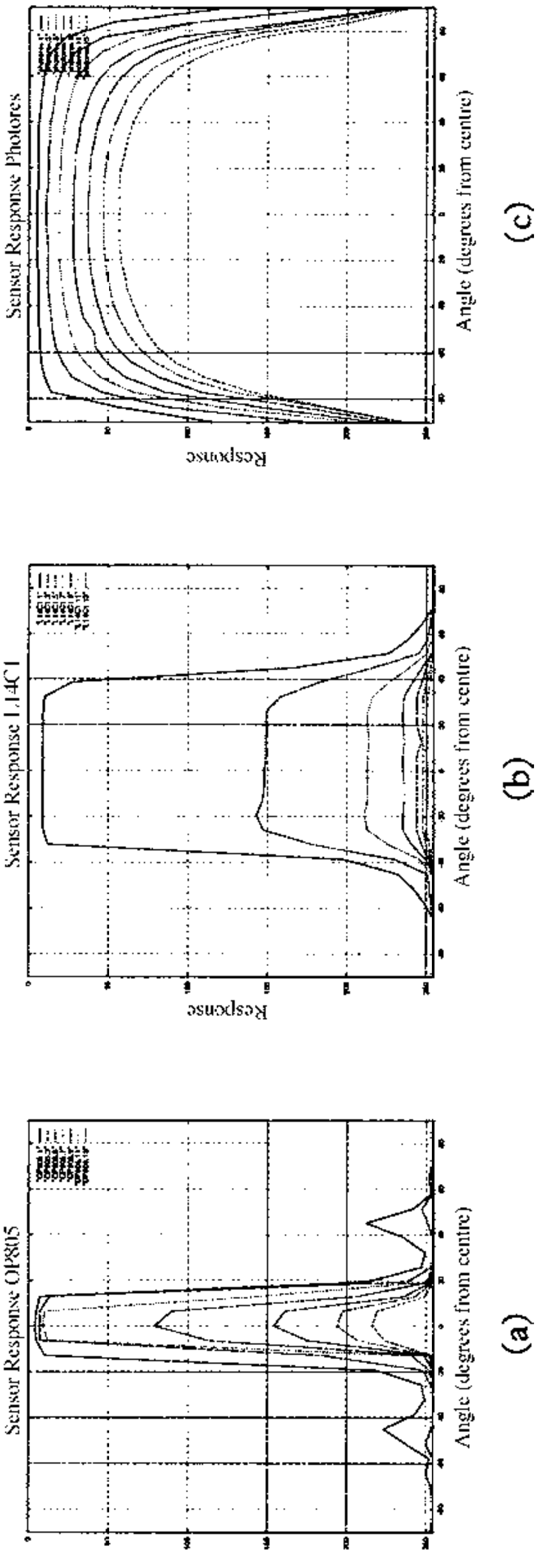
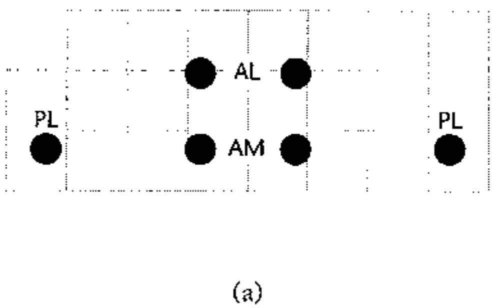


Figure 3. Measured angular sensor response of the: (a) OP805 phototransistor; (b) L14C1 phototransistor; and (c) CdS photoresistor.



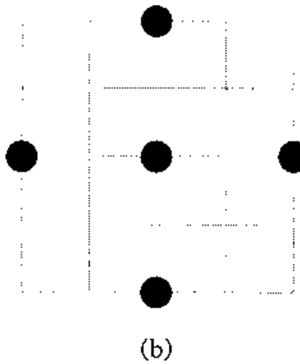


Figure 4. Front view of: (a) robotic sensor arrangement; and (b) simplified uniform sensor arrangement.

dimensional vector for the cross-shaped arrangement. Again, only a single source is permitted to be illuminated at any one time, allowing for a total of eight possible motion transitions as illustrated in Figure 5(a). As with the static excitations described earlier, a small amount of Gaussian noise was added to the modelled sensor values to produce 1000 vectors.

Further to the training sets, motion-based test datasets were also produced for both sensor geometrics. These consisted of the noise-free excitations corresponding to Manhattan and diagonal transitions on a full three-by-three grid of sources (Figure 5). Finally, a third test set of stationary excitations was prepared which was made up of pairs of identical sensor values for the current and single time-delayed portions of the input vectors.

4.2. *Analysis of Algorithm Clustering Performance*

Each of the four algorithms was initially trained on the position and motion datasets corresponding to the cross-shaped sensor geometry. The resulting learned weight vectors were compared with the known ideal solution (corresponding to the noise-free excitation observed by the five sensors). Using fully connected single layer networks with five and eight outputs, respectively, one would expect the networks to tune each of those outputs to recognize one of the five unique fixed source positions or eight unique source transitions as appropriate.

Simulation of the HCL algorithm on the static and motion datasets confirmed that the technique is highly susceptible to the orphaning of units. It was observed that the algorithm consistently failed to discover the optimal representation of the data vectors. Instead, several of the available units remained unused, with the precise number of unused units being dependent upon the initial values of the network weights. In the worst case, when the weights were deliberately initialized in a region remote from the data, only a single unit was found to win for all the data vectors. These empirical results confirm the pathological example previously depicted (in two dimensions) in Figure 1. As a consequence of these observations, it is clear that HCL would be unsuitable for use in any practical unsupervised clustering application.

Similar experiments involving the $FSCL_D$ approach showed that it is capable of learning to cluster optimally the training data in situations where the weights are

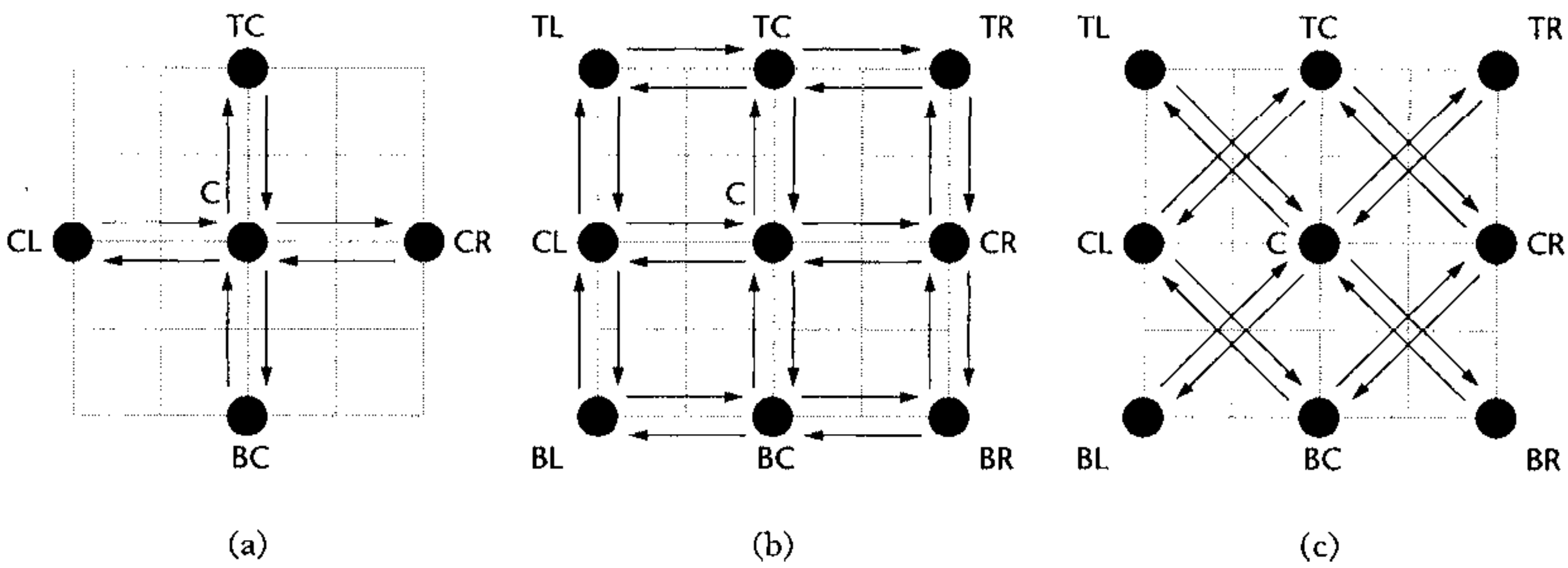


Figure 5. Source transition diagram showing: (a) the eight basic transitions used in network training; (b) the Manhattan test transitions; and (c) the diagonal test transitions.

initialized in the general vicinity of that data. Empirically, it was determined that a bias factor of $C = 2$ produced the most reliable clustering under these conditions. However, when the weights were again deliberately initialized in a region remote from the data it was found that $FSCL_D$ also failed to use all units effectively. This problem was attributed to the magnitude of the bias factor being insufficient to compensate for the correspondingly larger distance between weights and inputs. The conscience penalty can be strengthened by selecting a larger value for the bias factor in order to compensate for this poor weight initialization. However, while this will achieve the desired goal of bringing the peripheral units into the region of the data, it was found to introduce an unwanted secondary effect. Once the units have moved to the vicinity of the data, a large bias will inappropriately dominate the distance calculation of equation (3) and thus cause all units to acquire identical weight vectors. This clearly produces an unacceptable solution.

The underlying problem with $FSCL_D$ exposed by this investigation is the need to balance carefully the influence of the distance metric versus that of the conscience penalty. It may be possible to achieve this by beginning the training process with a large initial bias factor and slowly decaying this value as training proceeds. However, even if such a modification were shown to be effective, one is still left with the problem of determining a reasonable choice for both the initial bias factor and its rate of decay. Since unsupervised techniques are intended to be used to identify unknown structure within high dimensional distributions of unlabelled data, it would be difficult to confidently select appropriate initial bias parameters in order to apply reliably the approach to complex distributions.

In contrast to the problems encountered with HCL and $FSCL_D$, investigation of the Krishnamurthy implementation of FSCL found that the algorithm provides consistent and reliable clustering of the static and motion-based training data corresponding to the cross-shaped sensor array. Even when beginning with poorly initialized weights, this approach encountered no difficulties in accurately identifying the existence of the unique input states. This result is particularly encouraging given the simplicity of the $FSCL_K$ approach.

One problem that did arise, however, was not a specific shortcomings of the learning, but rather a consequence of the WTA activations. When evaluating the network's generalization abilities on the test datasets it was observed that the WTA nature of the outputs forces the system to classify novel input vectors as members of only one of the previously learned classes. However, this does not accurately represent the true nature of the new inputs, which would be more appropriately represented not by one unit, but by a mixture of the previously learned states. For example, in the case of one of the corner sources being illuminated in the static position tests, it would be preferable to represent the new input as a combination of both the corresponding horizontal and vertical components of its position. Results of this type can be achieved by replacing the WTA activation function with a soft activation, such as that of equation (8):

$$y_i = \frac{\left(\sum_j (x_j - w_{ij})^2\right)^{-1}}{\sum_k \left(\sum_j (x_j - w_{kj})^2\right)^{-1}} \tag{8}$$

The benefits of this modification can be seen clearly from the Hinton diagrams of

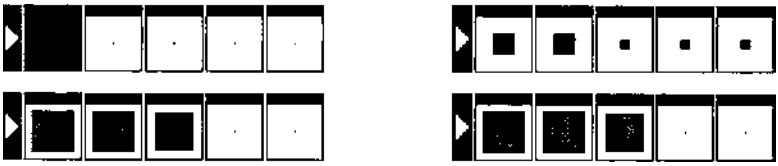


Figure 6. Comparison of output classifications resulting from the introduction of a modified network activation function.

Figure 6. Here, two networks possessing identical weights were excited from the same test vector (bottom row of activations). The left network employed the original WTA activations, while the network to the right used the analogue activations.

Further simulated and actual hardware tests of $FSCL_K$ using the true robotic sensor geometry showed that the algorithm was capable of learning to identify the horizontal component of the motion task. However, the properties of the selected sensors did not provide sufficient information for the network to discriminate the vertical component of the motion. It should be noted that this is not a specific criticism of the clustering technique, but rather a consequence of the sensor geometry employed. Also, it was necessary to modify the activation function used during the hardware-based experiments to accommodate the integer-only mathematics of the microcontroller. The actual activation function used in those tests is given in equation (9):

$$y_i = \frac{255 \times \sum_j (x_j - w_{ij})^2}{\sum_k \left(\sum_j (x_j - w_{kj})^2 \right)} \tag{9}$$

Finally, the performance of the SCL algorithm was evaluated on the vision tasks. When using a variance value of 0.1 the network was found to cluster easily the static and motion-based data corresponding to the simplified sensor geometry (Figure 4(b)). However, in situations of poorly initialized weights, SCL exhibited some sensitivity to the selection of network variance. When a small variance is used the corresponding activations become so small that the network makes extremely minute weight changes. This results in very long training times. It is possible to overcome this problem by commencing the training with a large variance and decaying this value as learning progresses. However, as was the case with the proposed modifications to $FSCL_D$, it is not obvious how best to select the initial values of these parameters in order to help ensure robust learning of complex distributions. Also, while SCL performed well on the test problems, it requires significantly more computation than $FSCL_K$, making it less desirable for implementation in constrained hardware systems such as the mobile robot presented here. For this reason it was not possible to test SCL in the physical robot, though simulations using the robotic sensor geometry yielded solutions which were consistent with those achieved previously with $FSCL_K$. The histogram of Figure 7 summarizes the relative performance of FSCL and SCL on the simulated motion-based tests.

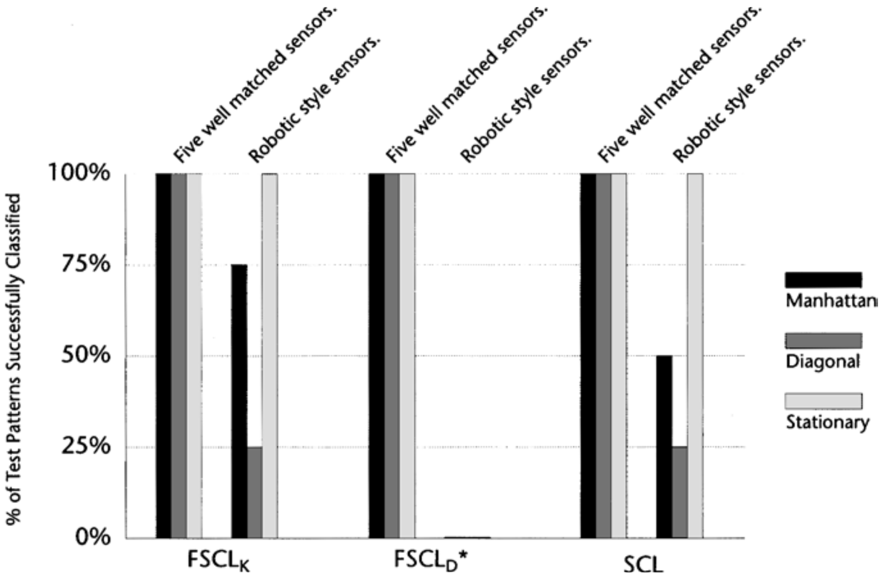


Figure 7. Generalization properties determined as per cent of novel patterns correctly represented. (*Best possible algorithm settings.)

5. Conclusions

This article has examined the relative performance of four neural network learning algorithms based on their theoretical and practical abilities to solve vision-based learning tasks. Specifically, the task of identifying motion of a point source was investigated in the context of a small autonomous robotic system. It has been shown that standard competitive learning is highly susceptible to the orphaning of network units and was therefore found to be unsuitable for general use. The $FSCL_D$ algorithm was able to learn correctly a subset of the problems investigated, but demonstrated sensitivity to the network bias factors owing mainly to the additive nature of this bias component. In contrast, $FSCL_K$ was found to operate extremely reliably and did not suffer from the shortcoming of $FSCL_D$. Finally, SCL performed fairly well on the tasks considered, but has the disadvantage that it requires significantly more computation than the $FSCL_K$ approach, which performed as well, or better. Through the course of this investigation we have also introduced the notion of an analogue unit activation to the FSCL algorithms, which significantly improved its generalization performance. Finally, tests on an actual physical robot demonstrate that the $FSCL_K$ algorithm can be implemented effectively in such a modest system and that it is able to provide a robust preprocessing system for motion detection.

Acknowledgements

The authors wish to acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

DeSieno, D. (1988) Adding a conscience to competitive learning. *IEEE International Conference on Neural Networks*. San Diego.

Foelix, R.F. (1996) *Biology of Spiders*. New York: Oxford University Press.

Gorman, R.P. & Sejnowski, T.J. (1988) Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks,* **1**, 75–89.

Gozani, T., Ryge, P., Shea, P., Seher, C. & Morgado, R.E. (1989) Explosive detection system based on thermal neutron activation. *IEEE Aerospace and Electronic Systems Magazine,* **4**, 17–20.

Hertz, J., Krogh, A. & Palmer, R. (1991) *Introduction to the Theory of Neural Computation*. Redwood City, CA: Addison-Wesley.

Krishnamurthy, A.K., Ahalt, S.C., Melton, D.E. & Chen, P. (1990) Neural networks for vector quantization of speech and images. *IEEE Journal on Selected Areas in Communications,* **8**, 1449–1457.

Le Cun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W. & Jackel, L.D. (1989) Backpropagation applied to handwritten zip code recognition. *Neural Computation*, **1**, 541–551.

Moody, J. & Darken, C.J. (1989) Fast learning in networks of locally-tuned processing units. *Neural Computation,* **1**, 281–294.

Pomerleau, D. (1996) Rapid adapting machine vision for automated vehicle steering. *IEEE Expert,* **11**, 19–27.

Rumelhart, D.E. & Zipser, D. (1985) Feature discovery by competitive learning. *Cognitive Science,* **9**, 75–112.