Towards robots that give each other navigational directions: Learning symbols for perceptual categories.

Jason Fleischer and Ulrich Nehmzow
Department of Computer Science
The University of Manchester
Manchester, M13 9PL, U.K.
jfleischer,ulrich@cs.man.ac.uk

5.4.2001

Abstract

We are interested in how robots can learn to give navigational directions to each other. As a first step towards this goal, this paper presents a neural network that enables two robots to agree on the meaning of a set of symbols describing perceptual categories. The network has two layers of nodes, one that automatically classifies sensor readings into n perceptual categories (unique to each robot), and another which links the resulting categories to the same number of symbols (agreed on by both robots). Symbols are assigned such that each perceptual category has only one symbol but one symbol may represent any number of categories. The network was tested in an experiment where two robots learned symbols to represent eight locations in a small T-shaped environment. Symbol agreement between the two robots was successfully learnt for a wide range of learning parameters.

1 Introduction

1.1 Motivation: Effect of communication on a scouting task

Communication is an important part of any multi-robot system. The transfer of information from one robot to another enables the group to act more efficiently then it would otherwise. However, it is often difficult to quantify the exact effect that communication has upon task execution, unless the task can be performed both with and without communication.

One task that can be performed in the presence or absence of communication is finding a goal location in an unknown environment. It can be accomplished by random searching if there is no better strategy, and does not require any communication with other robots. However, if another robot has already found the goal location it could try to give navigational directions to the robot attempting the task now. Any statistically significant change in the time to find the goal between robots searching randomly and robots that can give each other directions is the effect of communication on task performance. We call this experimental scenario the *scouting task*. While there are many ways of performing the scouting task, some principles guide our choice of methods.

What to communicate

Two kinds of information are useful for communicating navigational directions, metric and topological. *Metric information* consists of measurements specifying the exact position of an object in relation to some global reference point, e.g. '38° N 15° W'. *Topological information* consists of information about the adjacency and orientation of locations without exact distances, e.g. 'east of the T-intersection'. To communicate about metric information requires a reliable way of measuring position against the previously agreed on global reference point. To communicate topological information requires a symbol system, a way of representing arbitrary reference points and their relationships to one another. Note

Proc. TIMR 01 - Towards Intelligent Mobile Robots, Manchester 2001.
Technical Report Series, Department of Computer Science, Manchester University, ISSN 1361 - 6161.
Report number UMCS-01-4-1. http://www.cs.man.ac.uk/cstechrep/titles01.html

that the term symbol is meant in the simple sense of an arbitrary entity that is part of a formal symbol system. For the sake of clarity, we ignore linguistic and philosophical aspects of what makes a symbol different from any other form of representation

We believe that robots should communicate about navigation using a mutually learnt symbol system rather than transferring metric information. One reason for this belief is that metric information can be difficult to use in navigation if it is inaccurate. Unfortunately, most mobile robots cannot generate accurate metric information in large scale environments because they use wheel odometry, which suffers from cumulative error. We prefer a symbol system for communicating about navigation because it can transfer topological information, which we know is sufficient for mobile robot navigation [5].

Symbol grounding

The symbol system should be learnt by each robot instead of being included in the robot's algorithm design as *a priori* information. Any symbol system where the meanings of the symbols are not intrinsic to the symbol system itself is said to suffer from the symbol grounding problem [6]. Symbol grounding is the process by which the meanings of symbols are linked to (grounded in) the actual perceptions of an agent. Since a robot is a physical system that interacts with the real world, any attempt to assign symbol meanings *a priori* would mean that the symbols were not grounded in the real world. This is problematical, because according to the physical symbol system hypothesis [3], if a robot's representations are not grounded in the real world will not be able to act in an intelligent fashion.

Unsupervised learning

We would also like our robots to be as autonomous as possible. This means that we would prefer to remove all human input from the symbol learning process. For instance, if a human has to label exemplars of locations in a large environment it could be very time consuming. Therefore, we want a method that allows the robots to create their own symbol system in the absence of an external teaching signal.

1.2 Problem: Learning coordinated symbols for locations

Given the above arguments, it can be seen that the first step towards robots that communicate directions to each other is a method for allowing a group of robots to learn symbols. In this paper we address the problem of learning symbols representing locations in the environment.

Unfortunately, most robots cannot learn symbols that directly represent locations. It is impossible to perceive location unless the robot can sense absolute position beacons, as in GPS[1]. Since most robots do not possess such sensors or operate indoors where GPS signals are inaccessible, they must learn symbols that represent sensory perceptions of locations rather than absolute locations themselves. The crucial difference is that there is not a one-to-one mapping between sensory perceptions and locations. A robot may generate several different sensory perceptions in the same location or the same sensory perception may occur in more then one location. Therefore, the problem of learning symbols representing locations is only approximated by the problem of learning symbols that represent robot perceptions of locations.

Adding additional robots only makes symbol grounding more complicated. When multiple robots are learning the same symbol system, each robot must not just learn to use one symbol for each perceptual category consistently, but it must also do it in a way which is consistent with the use of the symbol by other robots. But even two robots of the same model are unlikely to have the same perceptual categories, due to small differences in the robots' mechanics, electronics, and software. So learning to use symbols in a way that is consistent with other robots is a non-trivial problem. This leads to the full definition of the problem: each robot must assign symbols to perceptual categories, which are approximations of locations, so that all robots generate the same symbol for any specific location, regardless of differences in perception. When consistent use of symbols occurs among the robots we call the resulting agreement *symbol coordination*.

Notice that the full problem definition above says that the robots should generate the same symbol for any specific location. This does not reconcile well with the notions presented earlier, that the robots cannot directly sense their locations and that we want to remove human input from the learning process.

[1] Global Positioning System, i.e. a method to determine latitude, longitude, and altitude from satellite radio signals.

Without either human input or a location sensor how can the robots sense whether they are communicating about the same location? To overcome this problem we postulate the existence of an attention mechanism, a component of the robot's controller that is separate from the communication system that can be used to provide context for the messages exchanged.

Other communication learning systems have also used attention mechanisms to solve similar problems. Typically, this is done by placing both robots in the same situation simultaneously, thus assuring that they are communicating about the same context. Our robots, on the other hand are unable to differentiate between sensor returns generated by another robot and those that are generated by the rest of the environment. For our system, this problem remains unsolved at the moment. The current implementation uses human input to show that the rest of the symbol coordination system works. A discussion of the effects of substituting external information for the attention mechanism is given in section 4.

1.3 Similar problems addressed in the literature

This section presents other research on robots learning grounded symbols for communication. Much more work has been done in simulation, especially in the area of Artificial Life, but this is beyond the scope of the current review. The terminology of the original authors has been used in this review, so it should be noted that following terms are equivalent: the meaning of a set of signals, a vocabulary, a lexicon, a set of labels, and the mapping between symbols and perceptual categories.

Yanco and Stein [11] were the first to demonstrate the feasibility of using a simple learning system to assign the meaning of a fixed number of symbols using real robots. More specifically, they demonstrated that two robots using a reinforcement learning algorithm can coordinate a set of symbols describing motions, such as 'spin' or 'move forward'. The robots received positive reinforcement if they performed the same action, and negative reinforcement if they did not.

Researchers at the Vrije Universiteit Brussel have been working on several different aspects of self-organised communication. Their approach is for robots to play 'naming games' that allow them to self-organise a lexicon that is maintained separately by each robot. Each robot takes turns as both a speaker and a listener in the playing the naming games until each robot's lexicon agrees with the others. The naming games require that each robot have a similar and useful method of categorising the environment; this is provided by having each robot play a 'discrimination game' with itself where sets of feature trees compete to best categorise the environment [4, 9]. Using these two games, both robotic and simulated agents can successfully create coordinated lexicons using language games [8, 9].

Rather than self-organising a language from scratch, it is also possible for one robot to learn the language from another which already posses the full vocabulary. This imitative learning method has been successfully used to coordinate a vocabulary between a human teacher and a single robot learner, a robot teacher and a single robot learner, and a simulated robot teacher and a group of nine simulated robot learners [1, 2]. This method is based on a recurrent neural network that learns associations between communication signals and perceptions in spite of possible temporal delays between the arrival of the communication signal and the perception of the communication topic.

2 Generating symbol coordination

2.1 The neural network

The basic idea behind our approach is for each robot to learn what symbol is appropriate for describing a given perception. This is done in two separate steps: (1) classification of current robot perceptions into one of n perceptual categories, and (2) association of the perceptual category with one of n symbols, where there are the same number of symbols as categories. These two steps are performed in separate layers of the network and trained at different times. The network is represented schematically in figure 1.

First layer - Category creation

Categorisations of robot perceptions are formed using Kohonen's Self-Organizing Map (SOM) [7]. A SOM is a single layer neural network that performs categorisation by comparing the weight vectors of every node to the input vector. The node whose weights are closest by Euclidean norm to the inputs is designated as the category of the input. At each time step the weights are updated so that the weight

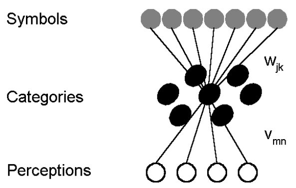

Figure 1: A schematic representation of a neural network for generating symbol coordination. The first layer is a Self-Organising Map that generates n categorisations of the perceptions. The second layer performs a winner-take-all assignment of each of the n symbols to zero or more categories.

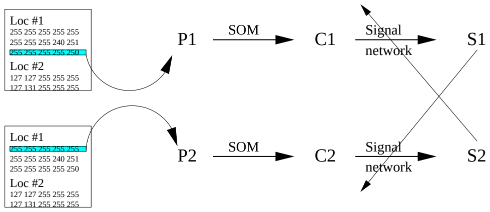

Figure 2: The symbol coordination training process. Each robot randomly selects a perception(**P**) in its sensor history from the topic location and uses the SOM to create an environmental categorisation (**C1** and **C2**). The symbol assignment layer determines the symbol (**S1** and **S2**) for that categorisation and transmits it to the other robot. Each robot uses the symbol transmitted by the other robot as a target value for training its own symbol assignment layer.

vectors of all nodes in a variable-size neighbourhood around the winner move closer to the input just presented, according to equation 1. Let $u_m(t)$ be the value of input node m at time t and v_{mn} be the the weight between input node m and output node n. The weights for the winning node and its neighbours in the map field are updated using:

$$v_{mn}(t+1) = v_{mn}(t) + \eta(t)\left(u_m(t) - v_{mn}(t)\right), \tag{1}$$

where m is the index of a node in the neighbourhood, and η is the learning rate ($0 \le \eta(t) \le 1$). For further details see [7, 10].

Second layer - Symbol assignment

The symbol meanings are assigned by a single layer winner-take-all network which links each of the n output nodes of the SOM to a like number of symbol nodes. The number of symbols is the same as the number of categories since there could never be a requirement to communicate about more locations then the robots could perceive, but it might occasionally be the case that there would be as many distinguishable locations to communicate about as there are categories.

The network increases symbol coordination by adjusting the weights of the symbol meaning network in the following manner. Let $w_{jk}(t)$ be the weight connecting the jth of n inputs to the kth of n outputs at time t. If j' is the SOM node representing the current perception and k' is the symbol generated by the other robot communicating about the same location, then after every communication episode the weights are updated by the equations below using the learning rate $\epsilon \in [0,1]$:

$$\begin{aligned} w_{j'k'}(t+1) &= w_{j'k'}(t) + \epsilon\ , & (2)\\ w_{j'k}(t+1) &= \frac{w_{j'k}(t) - \min\limits_k w_{j'k}(t)}{\max\limits_k w_{j'k}(t) - \min\limits_k w_{j'k}(t)}\ ,\ \ \forall k. & (3) \end{aligned}$$

In other words, at each communication episode the robot adjusts its beliefs closer to the symbol it received from the other robot and then performs a linear normalisation of all weights from the winning input node.

Training the network

Training of the system occurs in two stages. In the exploration phase the robots take sonar scans of their training environment. These scans are used to train the SOM to produce perceptual categorisations, after which the first layer weights are frozen.

The second phase is communication training, where the robots generate symbols to represent locations in the environment. The location is selected using an attention mechanism[2] that ensures each robot is

[2] In this paper, the attention mechanism is replaced by the experimenter grouping together sensor scans that represent the same location.

communicating about the same location. A perception from the location highlighted by the attention mechanism is used by each robot to generate a symbol, after which each robot trains its network with the symbol generated by the other robot as the target value. This process is illustrated in figure 2. In the event that a symbol has not previously been assigned to a perceptual category, a previously unassigned symbol is assigned to the category with its weight having the value of the learning rate, ϵ.

2.2 Group-supervised learning

The neural network described above uses supervised learning to produce communication in an unsupervised fashion. When training its symbol assignments, each robot uses the symbols output by other robots as target values for training its own network. In this way, the individual robots use supervised learning to produce unsupervised learning of communication at the group level. This process works because each robot will generate different symbol assignments due to the initial random selection of network weights. When the robots' different initial symbol assignments come into conflict, the learning system works to coordinate the symbol assignments without any teaching signal that is external to the robot group. The idea is similar to self-supervised learning in an auto-associative network, but in this case two separate networks each provide the training signal for the other. We call this type of learning *group-supervised*.

The attention mechanism plays an important role in group-supervised learning of symbols to represent locations. If the attention mechanism cannot successfully tell the robots when they are communicating about the same location, then the group-supervised paradigm will not work. The robots will end up developing arbitrary symbol assignments rather then ones which increase symbol coordination. Ideally, the attention mechanism replaces a group-external teaching signal with a mechanism that is group-internal.

3 The experiment

The experiment task requires two robots to agree on a set of symbols for eight locations in a small T-shaped environment. This experiment serves two purposes. First, it is intended to demonstrate that the neural network described previously can successfully learn symbol coordination on two real robots. As such, it is an early test using a very simplified scouting task and substituting human knowledge for an internal attention mechanism. Secondly, this experiment was intended to explore the effects of varying network parameters on the quality and speed of symbol coordination. The parameters network size, n, and learning rate, ϵ, were compared using the same robot data in each case. The network's performance was evaluated for $n = 4, 8, 15, 32$, and 65 nodes and for $\epsilon = 0.2, 0.4$, and 1.

3.1 Experiment description

Hardware and software

The experiment was run on two Nomad Scout robots. The Scouts are circular-plan differential drive robots, approximately 40cm in diameter and 35cm tall, with 16 Polaroid sonar sensors arranged in a ring around the robot. Network training and testing took place off-line using data collected by the Scouts in the T-shaped environment.

The SOMs had two-dimensional hexagonal topology, with the size ratios of the two dimensions being determined by the ratio of the first two principle components of the training data. Nodes were first linearly initialised along the two largest principal components of the training data. The SOM was batch trained in two phases; first a rough training phase with large initial η (learning rate) that linearly decreases each epoch for a small number of epochs, and then a fine tuning phase with a smaller initial η for a larger number of epochs. The exact values of η and the number of epochs are functions of network size and the amount of data in the training set (see [10] for details). After training the SOM weights were then frozen for training the symbol assignment layer.

The symbol assignment layer was trained on the same data as the SOM layer. As mentioned previously, as a substitute for an attention mechanism the experimenter grouped sonar scans together that came from the same location. A complete training run was composed of 800 communication episodes, where each episode consisted of the random selection of a location and each robot's subsequent random selection

	$\epsilon = 0.2$	$\epsilon = 0.4$	$\epsilon = 1.0$
$n = 4$	87.18% (.14%)	95.65% (.08%)	81.21% (.16%)
$n = 8$	96.57% (.08%)	97.58% (.06%)	81.30% (.16%)
$n = 15$	98.15% (.06%)	98.65% (.05%)	92.12% (.11%)
$n = 32$	97.09% (.07%)	96.16% (.08%)	84.37% (.15%)
$n = 65$	99.20% (.04%)	99.43% (.03%)	82.07% (.16%)

Table 1: Test set performance versus learning rate, ϵ and network size, n. The performance for each parameter setting is the total percentage of correct symbol coordinations using the test set over all 25 separate iterations of symbol training. The number in the parenthesis is the standard deviation for that parameter setting, calculated by assuming the results are generated by a binomial distribution.

of one of its sonar scans from that location. The robots could determine what location a sonar scan belonged to by using the labels generated by the human experimenter.

Setup

Each robot collected sensor data on five different runs through the environment for use in off-line training and testing. For every visit to one of the eight locations the robots took ten consecutive scans without moving, giving each robot $8 \times 5 \times 10 = 400$ sensor scans of the environment. The locations themselves were designated by the human experimenter and the robots were manually driven to approximately the same locations on each data collection run. Therefore, the robot locations and orientations have fairly large variations, on the order of perhaps 25 centimetres displacement and 15 degrees misalignment from run to run.

The 50 sonar scans each robot had collected at each location were randomly split into separate training (2/3) and testing (1/3) data sets. The training set was used to train first the SOM and then the symbol assignment layers. After training the test data was used to evaluate the generalisation of the learned symbol system, since there is a danger that the network would learn to fit specific features that exist in only the training data. The data-split/train/test cycle was repeated 25 times for each set of n and ϵ that were investigated.

3.2 Results and discussion

Symbol coordination

Figure 3 shows the symbol coordination values during training and testing for networks ($n = 4, 8, 15, 65$; $\epsilon = 0.2$). Each training communication episode is assigned a score of 1 for a successful symbol match between the robots, or a 0 otherwise. The graphs show these scores averaged over the 25 runs for each training episode. The figure also shows the test performance as the raw number of communication episodes in the test set where the two robots agreed on the same symbol, out of a possible 57800.

From the information in figure 3, it can be seen that the neural network succeeds in producing symbol coordination in this experiment. After training, the system has learnt a set of symbols that agree for almost every perception in both the training and testing data sets. Further evidence of this comes from table 1, which shows the test set performance of every parameter value. Even for the worst performing parameter values, the test performance never drops below 80%.

Learning to increase symbol coordination means that each robot will assign the same symbol to any set of locations that are indistinguishable in the presence of noise. Even better, if any robot in the group cannot distinguish two locations all robots in the group will assign those locations the same symbol. This effect can be seen in figure 4, which shows examples of coordinated symbol systems for ($n = 4, 65$; $\epsilon = 0.2$). In the four node network the locations *left*, *edge*, *right*, *bot1*, and *bot2* are categorised in such a way that they are not distinguishable by these two SOMs. Therefore, all of these locations are collectively labelled with the symbol 'ab'. The remaining three locations are labelled with the symbol 'ac' since they share a single SOM node on both robots. Since only two sets of locations are distinguishable between the two robots only two symbols are assigned. However, in the 65 node networks all locations are always distinguishable, and thus eight symbols are always used. The networks of intermediate sizes tend to produce a number of symbols in between two and eight.

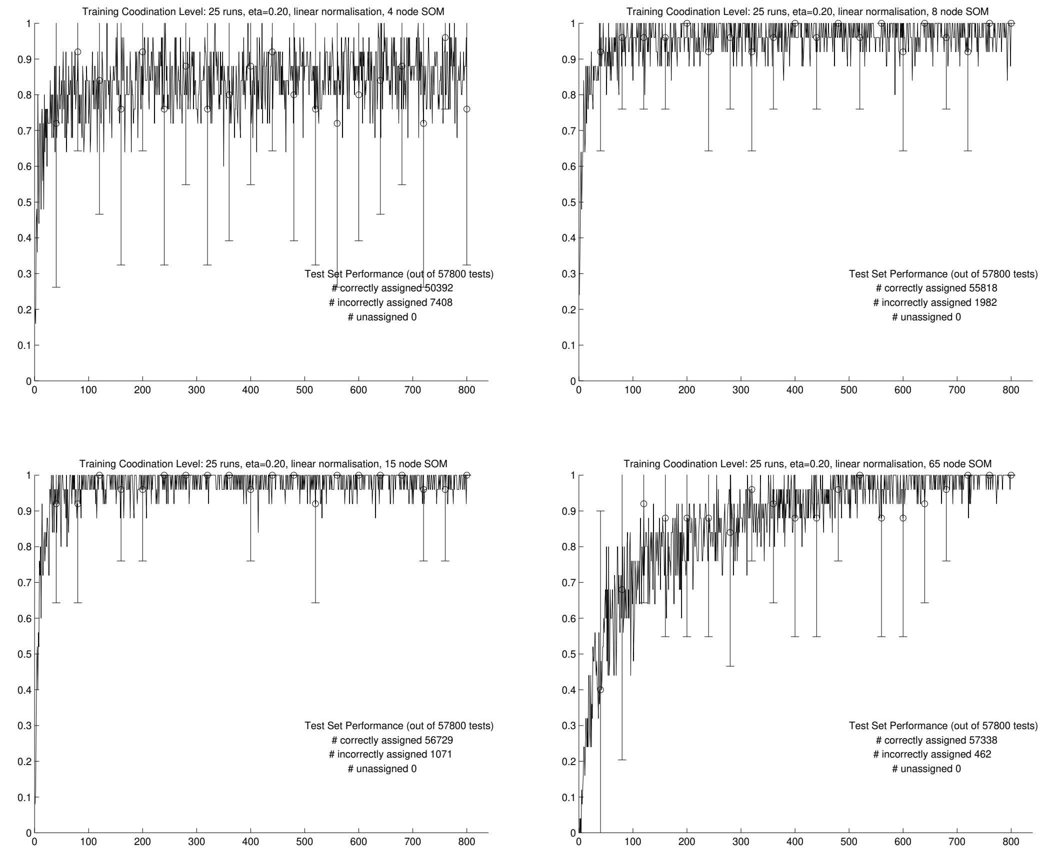

Figure 3: Symbol coordination values during training averaged over 25 runs of the learning system ($\epsilon = 0.2$). From top left to bottom right are 4, 8, 15, and 65 node networks. Each communication episode is assigned a score of 1 for a successful symbol match between the robots and a 0 otherwise. The values are averaged for each episode over the 25 runs to produce the values on the vertical axis. The horizontal axis indicates episode number. Error bars indicate one standard deviation every 40th episode. Each graph is labelled with the test set performance for that set of parameters. After each training iteration the robot's symbol systems were tested for every possible combination of perceptions in each location. The values on the graph represent the total number of successful tests (robots agree on a symbol) over all 25 runs.

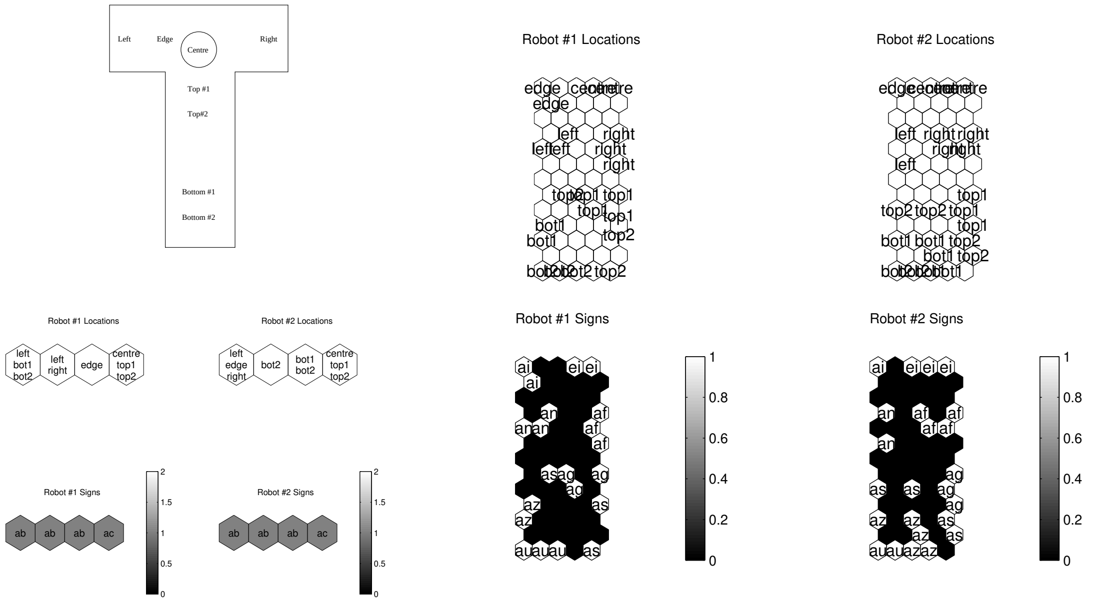

Figure 4: Top left: The T-shaped environment and the eight locations learnt in the experiment. The circle indicates the approximate size of the robot. Bottom left and Right: Randomly picked examples of the symbol labelling learned by the two robots after 800 communication iterations ($n = 4, 65$; $\epsilon = 0.2$). The hexagonal grid represents the output nodes of the SOM. In the top half of each sub-figure (*Locations*) the maps are labelled by the physical locations which activate each node. In the bottom half of each sub-figure (*Signs*) the maps are labelled by the symbol assigned to each node. If a category is never activated by the set of sonar inputs then it is not assigned a symbol. Symbols are designated as two-letter combinations for convenience in referring to them.

Why did the 65 node networks perform so well? It appears that if the network is large enough there will be no two locations which were indistinguishable by both robots. A 65 node network is an acceptable size for computing in our case, but recall that sonar scans were only collected at eight distinct locations. It is unclear how the number of nodes required to ensure good symbol coordination scales with the number of locations for which the robots have taken sensor scans. If this scaling is say, exponential, it could make it very difficult to use this network in a realistic robot application.

Time to learn

Figure 3 shows that the speed of convergence in symbol coordination depends on the value of ϵ chosen and the size of the SOM selected. Typically, for ($n = 15$; $\epsilon = 0.2, 0.4$) the two robots approach total symbol coordination in less than 200 communication episodes. Using the same values of ϵ with $n = 65$ produces near total coordination in less then 500 episodes. In general smaller numbers of nodes produce quicker convergence. Unsurprisingly, quicker convergence is also produced by higher values of learning rate; $\epsilon = 0.4$ produces roughly twice the speed of convergence as $\epsilon = 0.2$ does.

What is much more interesting, is that in our case we did not see the exponential relationship between number of symbols and time to convergence that is predicted for the symbol coordination problem in other work [11]. Networks with $n = 65$ took about five times as long to produce convergence in the robots as networks with $n = 4$. However, this effect could be due to the fact that even though the number of available symbols increased, there was no increase in the complexity of the sensor data to be learned. That is, the number of symbol categories that the robots could collectively distinguish remained the same. It would be interesting to determine if the convergence would be exponential with the number of (very distinct) locations the robots visited.

Best parameters

Although it is not necessarily possible to extrapolate the performance of the neural network from this experiment to a general case, we are still interested in finding out which parameter values perform the best as a guideline for future research. To evaluate the performance of various parameters, the number of successfully coordinated communication episodes in the 25 run test set is assumed to have been generated by a binomial distribution, such that the probability of observing r successes in n trials is $P(r) = \frac{n!}{r!(n-r)!}p^r(1-p)^{n-r}$ where p is the probability of success of any single trial. So if the percentage of successes is the binomial variable X, then the obtained experimental value is the expected or mean value of the distribution, given by $E[X] = np$; this allows us to calculate p. Subsequently, we can determine the standard deviation of the distribution by $\sigma[X] = \sqrt{np(1-p)}$. Therefore, the performance of one set of parameters, α can be said to be better then another set, β with 95% confidence if $E_\alpha[X] > E_\beta[X] + 3(\sigma_\alpha[X] + \sigma_\beta[X])$. The test performance of the network for each set of parameter values, along with the standard deviations of the binomial distributions describing them, are shown in table 1.

The set of parameter values with the best test performance, ($n = 65$, $\epsilon = 0.4$), had significantly more symbol agreements in the test set, over 99%, then any other set of parameter values, when modelled as a binomial distribution and testing for a 95% confidence interval. However, the separation between the 3σ error bars of the best parameter set and the second best one ($n = 65$, $\epsilon = 0.2$) was only 0.02%. The performance of these two 65 node networks were separated from the performance of all other sets of parameters by a clear margin. By looking at table 1 the general trend appears to be that the larger networks perform better then the smaller ones.

If training time is an important factor, the parameter values ($n = 15;$, $\epsilon = 0.4$) look very good. These parameter values produce over 98% performance but appear to be fully trained in less then 200 episodes. Although the 65 node networks had better test performance, they did not reach high levels of training performance until nearly 800 episodes of training.

Test set performance was also linked to the value of ϵ. The test performances of every number of nodes was averaged over each value of ϵ. A Student's t-test (95% confidence) on the resulting means showed that $\epsilon = 1.0$ performed significantly worse then the other two ϵ values. However, the t-test did not distinguish between $\epsilon = 0.2$ and $\epsilon = 0.4$. This shows a successful communication system requires gradual convergence rather than one-step learning.

4 Limitations and future work

Although it is capable of coordinating symbols, this neural network is unsuitable for the scouting task without an attention mechanism that enables the robots to autonomously coordinate the topic of their communication. Previous research has placed two robots in the same physical location as a way to ensure that they are communicating about the same topic. This option is not open to us because our robots use sonar sensors; each robot's ultrasonic emissions would interfere with the other's sonar readings. The physical nature of the robots restricts our approach to one where the robots must communicate about topics which are not perceivable at the time of communication. This very interesting problem is currently under investigation.

There are two forms of attention mechanism that will be the topic of future work:

- Transferring the SOM weights of the topic perceptual category from one robot to another. The robot receiving the message finds the SOM node in its own map that is closest in Euclidean norm to the perceptual category that is the topic of the speaker. This node is assumed to be the category representing the topic location. Although this methods will obviously only work with robots of near-identical perceptual abilities, preliminary experiments have shown promising results with our two Nomad Scouts. However, space precludes a full discussion of this work.

- Bootstrapping the meaning of a new symbol by using a previously learnt location symbol as a reference in learning a new one. A human-oriented example would be 'location X is just past the T-intersection'. This method has not yet been tried in an experimental situation.

5 Conclusion

We have presented a simple two-layer neural network that learns to coordinate symbols describing locations in the environment for two robots. The network classifies the environment into n perceptual categories and assigns a like number of symbols to those categories such that each category only has one symbol but each symbol may be assigned to zero or more categories. The assignment is performed in a way to increase the number of locations in the environment that the robots describe with the same symbol. After training for 800 episodes no parameter values in the experiment generated agreement in less then 80% of the test set perceptions.

Future work will concentrate on developing the missing attention mechanism that will enable this network to learn symbols without human input and to apply the network to the scouting task.

References

[1] Aude Billard and Kerstin Dautenhahn. Experiments in social robotics: Grounding and use of communication in autonomous agents. *Adaptive Behavior*, 7(3/4), 2000.

[2] Aude Billard and Gillian Hayes. Grounding communication in autonomous robots: An experimental study. *Robotics and Autonomous Systems*, 24:71–79, 1998.

[3] Rodney Brooks. Elephants don't play chess. *Robotics and Autonomous Systems*, 6, 1990.

[4] Edwin De Jong and Paul Vogt. How should a robot discriminate between objects? A comparison between two methods. In R. Pfeifer, B. Blumberg, J.-A. Meyer, and S. Wilson, editors, *From Animals to Animats 5*, pages 86–91. MIT Press, 1998.

[5] Tom Duckett and Ulrich Nehmzow. Exploration of unknown environments using a compass, topological map and neural network. In *Proceedings of 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation,*, Monterey, CA, 1999.

[6] Stevan Harnad. The symbol grounding problem. *Physica D*, 42:225–346, 1990.

[7] Teuvo Kohonen. Self-organised formation of topologically correct feature maps. *Biological Cybernetics*, 43: 59–69, 1982.

[8] Luc Steels. Emergent adaptive lexicons. In *From Animals to Animats 4*, pages 562 – 567. MIT Press, 1996.

[9] Luc Steels and Paul Vogt. Grounding adaptive language games in robotic agents. In I. Harvey and P. Husbands, editors, *Proceedings of the 4th European Conference on Artificial Life*, 1997.

[10] Juha Vesanto, Johan Himberg, Esa Alhoniemi, and Juha Parhankangas. SOM Toolbox for Matlab 5. Report A57, Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland, April 2000.

[11] Holly Yanco and Lynn Andrea Stein. An adaptive communication protocol for cooperating mobile robots. In *From Animals to Animats 2*, pages 478–485, 1993.