Developing a Community Language

Colin Fyfe and Daniel Livingstone,
Department of Computing and Information Systems,
The University of Paisley.
email: fyfe0ci,livi-ci0@paisley.ac.uk

Abstract

We describe simulations of a community of agents who live in an environment which has some structure that the agents can learn to identify and subsequently about which they learn to communicate. Each agent has two entirely separate artificial neural networks which learn to perform the two tasks: identification of the structure in the environment and communication to others about the environment's structure. We show that a community of agents with similar representation capabilities is most successful in generating a common language and further that a community with different representation capabilities will evolve so that all members have the same representation capabilities.

1 Introduction

This paper will investigate issues involved in the origin of a new language within a community of users of the language. We describe simulations involving agents situated within a common environment. These agents each have two entirely separate artificial neural networks:

1. The first encodes the environment by finding those sources which are creating the environment. This is done in an unsupervised way which leads to each agent having an unique internal representation of the environment. We shall see that it is beneficial to the community to have all agents within the environment sharing a common capacity for representation though not necessarily sharing a common representation.

2. The second takes the output of the first neural network and encodes this in a way that is agreed by the community of agents. This second encoding is the shared communication in the community of agents.

No agent can investigate the internal representation of the environment held by any other agent. Each agent however exists within the shared environment and can perceive the output/communication of the other agents. Because of this we can view

- the external environment as objective
- the internal representation as subjective
- the language as an inter-subjectively agreed entity existing within the common social environment of the community of agents.

The remainder of this paper is organised as follows: section 2 identifies in more detail the prerequisites for a language community; section 3 discusses representation of the environment; section 4 deals with aspects of learning a language and we then have 3 sections of experimental work before completing with a discussion of what has been learned.

2 A Language Community

There has been much recent research into the evolution of communication using simulations. Most simulations (e.g. [7]) are predicated on the assumption that there is a need for the behavior of the receiver of the communication to be changed in some way by the communication. We, on the other hand, perform simple simulations to investigate the development of a joint communication language which is formed when the sender alone is forced to adapt to conform to the society in which it finds itself. We do however agree with [7] that such simulation should be grounded in the environment in which each agent finds itself. Therefore each agent uses a simple artificial neural network to extract information about the environment. We deliberately use a set of parameters for this network (particularly a fast learning rate) such that this information extraction is not 100% accurate and investigate the knock-on effect this inexactness has for the development of a common language.

It has been stated [6] that "learning is more a transfer of skills than a discovery". While this may be true about the acquisition of language it is certainly not true about the acquisition of concepts about the environment. Thus, since we insist on grounding our agents' languages in environmental experiences, we must maintain a strict segregation between the acquisition of information about the environment and the task of learning a language to describe the environment.

Therefore we believe that there are several distinct prerequisites which must be met before the development of

a language is possible:

1. There must be some regularity in the environment to make communication worthwhile. If the environment were to be totally random - such as a maximum entropy environment - the only useful language would have to describe each and every event separately. There would be no advantage to be gained in describing sets of events. This constraint is specific to the environment and is external to any agent or group of agents.

2. Each agents must be able to identify separately individual features in the environment. Notice that this does not necessitate similar internal representations of the environment. This constraint is on the agent acting as an individual existing in an external (to the agent) environment.

3. A language must be shared by two or more agents in an environment: this does require that the agents share common external representations. This constrains the joint behaviour of the agents.

Thus we maintain *separate* constraints on the environment, on the representation capabilities of the agents and on the language which the community of agents use to communicate.

We describe a set of experiments in which a group of agents exist in an environment which has some redundancy. Each agent is able to learn, in an unsupervised manner, to identify and represent the independent features of the environment. Each agent's internal representations of the common global environment are unique to itself. However each agent also lives in a social community of agents which is learning to communicate with each other. Thus there exists in the environment an intersubjective agreement on the use of language - the convergence to a common language is supervised by the language itself.

We show that

- Such a community can learn a common language regardless of how many different internal representations of the environment exist.

- The accuracy of the coding between the features of the environment and the internal representations of the environment determines the accuracy of coding between the representation of the environment and the communication scheme.

- There exists a regularity in the process of transforming the internal representations to an externally agreed language which determines the language.

- A population with initially differing representation capabilities can evolve to a population of similar representation capabilities (though not necessarily with similar representations) when the impetus for convergence is at the communication level *not* the representation level.

3 Coding the Environment

Barlow [1] has developed a theory of learning based on the neuron as a "suspicious coincidence detector": if input A is regularly met in conjunction with input B this represents a suspicious coincidence; there is something in the neuron's environment which is worth investigating. A crude example might be the coincidence of mother's face and food and warmth for a young animal.

The types of codes which are required are sometimes known as "factorial codes": we have lots of different symbols representing the different parts of the environment and the occurrence of a particular input is simply the product of probabilities of the individual code symbols. So if neuron 1 says that it has identified a sheep and neuron 2 states that it has identified blackness, then presenting a black sheep to the network will cause neurons 1 and 2 to both fire. Also such a coding should be invertible: if we know the code we should be able to go to the environment and identify precisely the input which caused the code reaction from the network. So when we see neurons 1 and 2 firing we know that it was due to a black sheep.

We will maintain a close connection with psychological principles which requires that we are using a biologically plausible rule such as the Hebb rule (see e.g. [5]).

The benchmark experiment for this problem - due to Földiák [2]- is shown in Figure 1. The top line shows sample input data which consists of a square grid of input values where $x_i = 1$ if the i^{th} square is black and 0 otherwise. However the patterns are not random patterns: each input consists of a number of randomly chosen horizontal or vertical lines. The important thing to note is that each line is an independent source of blackening a pixel on the grid: it may be that a particular pixel will be twice blackened by both a horizontal and a vertical line at the same time but we wish our agents to identify both of these sources.

We will use a 4*4 grid on which each of the 8 possible lines (4 horizontal and 4 vertical) may be drawn with a fixed probability of $\frac{1}{4}$ independently from each of the others. The data set then is highly redundant in that there exists 2^{16} possible patterns and we are only using at most 2^8 of these. We will initially have 8 output neurons whose aim is to identify (or respond optimally to) one of the input lines. Thus from any pattern composed of some of the set of 8 lines, we can identify exactly which of the 8 lines were used to create the pattern. Note the factorial nature of the coding we are looking for: for example, neurons a, b and c will fire if and only if the input is composed of a pattern from sources 1, 3 and 8. Note also the code's reversibility: given that neurons a,

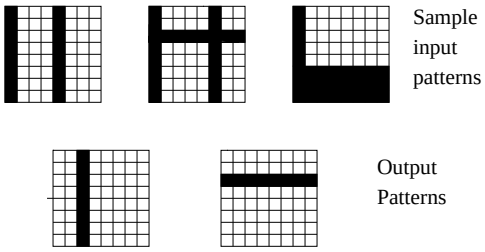

Figure 1: The top line shows sample data (input values) presented to the network. The second layer shows the independent sources which we hope will also be the network's response. If this is so, it has clearly identified the suspicious coincidences of lines.

b and c are firing we can recreate the input data exactly- it has come from sources 1, 3 and 8.

Note also that the factorial nature of the coding is unaffected by which neuron responds to which source: we do not care whether it is neuron 1 or neuron 5 which has identified the first horizontal line. We only care that it is identified accurately.

3.1 The Neural Network

It has been shown [3] that a layer of neurons whose weights are modified using simple Hebbian learning can be used to extract maximum information from a set of environmental data (they perform a Principal Component Analysis(PCA) of the input data) provided the network uses a negative feedback of activation.

Let the inputs to the network be denoted by the vector $\mathbf{x}$, the outputs by $\mathbf{y}$ and the weights between inputs and outputs by the matrix W. The operation of the network(Figure 2) is given by

$$\mathbf{y} = W^T\mathbf{x} \quad (1)$$

$$\mathbf{x} \leftarrow \mathbf{x} - W\mathbf{y} \quad (2)$$

$$\Delta W = \eta \mathbf{x}\mathbf{y}^T \quad (3)$$

where η is a learning rate which must satisfy the usual conditions for convergence for stochastic approximation algorithms. There is no explicit weight decay, normalisation or clipping of weights in the model. If we write the i^{th} input at time t as $x_i(t)$ etc. then we have

$$y_i = \sum_j w_{ij}x_j \quad (4)$$

$$x_j(t+1) \leftarrow x_j(t) - \sum_k w_{kj}y_k \quad (5)$$

Therefore,

$$\Delta w_{ij} = \eta y_i x_j(t+1)$$

$$= \eta(x_j(t) - \sum_k w_{kj}y_k)y_i \quad (6)$$

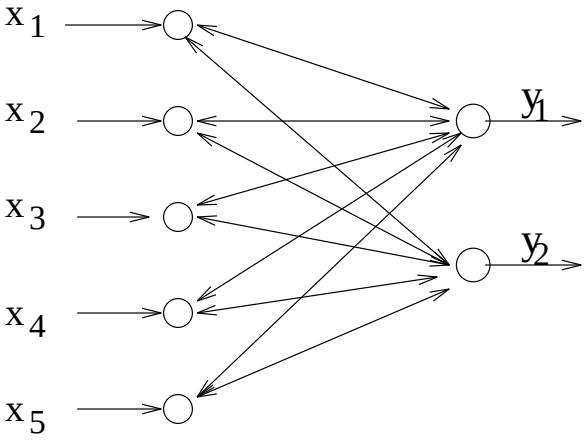

Figure 2: Activation is fed forward from inputs $\mathbf{x}$ through weights W to outputs $\mathbf{y}$. It is then returned as inhibition through the same weights before simple Hebbian learning changes the weights.

This last formulation of the learning rule is exactly the learning rule for Oja's Subspace Algorithm [8]: the weights then will not converge to the actual Principal Components but will converge to a basis of the subspace spanned by these components. Experimental results on both real and artificial data have previously been given to substantiate this (e.g. [3]).

3.2 Finding Independent Codes

If we use such a principal component net on the bar data described in the last section, our first principal component will be a small magnitude uniform vector over all 16 positions. i.e. we get a global smearing of the patterns which does not reveal how each pattern came to be formed. Therefore a PCA network will not be useful in identifying the individual sources.

The Subspace Algorithm will find a rotation of the principal components and will be similarly unable to identify any of the individual sources. However we make one simple change to the learning rule in the last section: we do not allow the weights to change from positive to negative. If a weight should be made to change sign because of the learning rule, we simply set its value to 0. It can later move from this value because of future learning but we will never allow any weight to become 0.

We have previously shown [4] that operating this simple Non-negative Weight Algorithm on Földiák's data causes the weights of the network to converge to find the independent sources. The weights into each output neuron will be large in those sections of the input space corresponding to a single source and each output neuron will be optimally placed to respond to one and only one of the independent sources.

However we cannot specify in advance which output neuron will capture which source. This is the agent's

internal representation of the environment. So as not to bias the internal representation in any way, the initial values of the weights are all set to small random values and each agent learns its internal representation on a different data set (though each data set is drawn from the same distribution of input data).

In terms of equation (4), the representation layer is a set of floating point numbers which in some way identify the particular set of sources which were used to create the current environment. The vector of outputs from this layer, $\mathbf{y}$, codes as floating point numbers the agents belief as to which sources have been firing to create the current environment. The weights from the environment to the representation layer have been adjusted so that the representation neurons can code the environment in such a way that the individual independent sources are identifiable by means of which neurons are firing to their maximum values.

In the following, we will use the notation $\mathbf{y}^p = \{y_1^p, y_2^p, ..., y_m^p\}$ to refer to the vector of representations or meanings of agent p on the current environment.

4 Learning a Language

We now however ask all agents to learn to communicate with each other: at this stage the whole population of agents exists in a shared environment and must learn to find a common language to describe this environment. We use a language comprising floating point numbers (as in [10]) rather than a discrete alphabet of terms since it is more easily extensible than a discrete representation would be.

Each agent has a (perhaps different) representation of the environment. It also has the means to express to the others in the community its view of the environment. Its communication network has output $\mathbf{s}$ where

$$\mathbf{s} = V\mathbf{y} \tag{7}$$

or for a single word s,

$$s = \mathbf{v}^T\mathbf{y} = \sum_i v_i y_i \tag{8}$$

where $\mathbf{y}$ is the agent's internal representation of the current environment and $\mathbf{v}$ is its vector of weights joining the representation vector to the communication word. We have omitted the superscript p for agent identification in order to render the equations more comprehensible.

We require the internal representation to be communicable to the others in the community. It would be possible to model a community of language users who, as they meet each other, interact in such a way as to bring each user's language closer to the other. For example when agent A_p meets agent A_k we could change the v-weights of each by

$$\Delta v_{ij}^p = \eta(s_i^k - s_i^p)y_j^p$$

$$\Delta v_{ij}^k = \eta(s_i^p - s_i^k)y_j^k$$

where v_{ij}^p is the weight between agent p's representation layer output y_j^p and its communication output s_i^p and similarly with agent k. η is a small learning rate which remains constant for the course of the simulation.

However, we have preferred to make a type of mean field approximation in that, rather than model each individual interaction, we have assumed that each agent is equally likely to interact with every other agent in the shared environment and so we may change each agent's language to make its language closer to the average language being used in the community.

This is performed by supervised learning where such supervision is performed by the language itself: the language at all times exists as the mean (floating-point) word describing the current environment where such mean is taken over all the agents currently existing within the environment.

Therefore if an agent has independent state $\mathbf{y}$ due to extraction and coding of the independent sources of the currently-viewed external environment E, we adapt the weights of the language module

$$\Delta v_{ij}^p = \eta(E_{l_i} - s_i^p)y_j^p \tag{9}$$

where E_{l_i} is the mean word in the environment describing the current environment. We also perform explicit normalisation after the learning step to ensure that the length of the vector into any one language neuron is always 1. This ensures that no single agent has an overbearing influence on the language.

We see that equation 9 will move the weights of an individual's language units so that its output (its spoken word based on the current environmental input) is more likely to be closer to the mean of its peers' spoken output based on the current environmental input. Clearly learning in (9) only stops when the agent is responding to the environment with a word which is exactly equal to the mean of all agents' responses to the current environment.

Notice that the language, while it exists externally to any one agent, is itself a function of the language (the output) of each agent. Thus the language itself is not reified; the language is responsive to the community which creates the language and is thus indirectly driven by the environment in which it is to be used. Therefore the language is potentially a moving target driven by both the environment and the interpretation of the environment by the community which uses the language.

The learning rule then is the simple LMS algorithm which will have converged when the output of the language output module equals the expected value of the language modules of all agents. This provides for the convergence of the whole community to a common language.

Input	A1/6	A2/7	A3/8	A4/9	A5/10
h1 h2 h3 v1 v2	3.10	3.06	3.08	3.06	3.15
	3.12	3.05	3.11	3.12	3.03
h0 h3	1.68	1.66	1.68	1.66	1.65
	1.76	1.69	1.67	1.67	1.62
h0 h2 h3 v3	2.77	2.73	2.74	2.77	2.70
	3.07	2.72	2.74	2.77	2.68
h1 h2 v1	1.99	1.95	1.96	1.96	1.95
	2.05	1.94	1.97	2.03	1.98
h1 v0 v2	2.55	2.53	2.55	2.55	2.58
	2.31	2.55	2.58	2.57	2.54
h2	0.94	0.91	0.90	0.93	0.88
	1.01	0.89	0.90	0.93	0.91
h1	0.81	0.81	0.83	0.81	0.85
	0.85	0.83	0.83	0.86	0.84
h0 h2 v1	2.00	1.97	1.99	1.97	1.94
	2.06	2.02	1.97	2.02	1.98
h3	0.84	0.82	0.82	0.83	0.81
	0.87	0.78	0.84	0.82	0.77
v0	0.68	0.69	0.69	0.67	0.66
	0.77	0.66	0.67	0.68	0.65

Table 1: Examples of common floating point communication about the environment where lines are labelled h0-h3 and v0-v3 and agents A1-A10. For compactness we have agents A1-A5 in one line and A6-A10 in the next.

5 Experiment 1 - Learning a language

We use a population of 10 agents, each of which learns an internal representation of the environment using 8 output neurons. If we have a fairly leasurely time in which each agent can learn its own internal representation, such representation can be learned so accurately that we could as easily simulate the situation with a look-up table. We have therefore used a simulation for the learning of the input representation which is so fast that it introduces inaccuracies into the agent's learning of the environment and their learned common language. We see the results in Table 1 where the left hand column shows typical sets of input data: h1,h2,h3,v1,v2 represents three horizontal and two vertical lines. The agents do clearly learn to share a common language - their individual responses to the input h1,h2,h3,v1,v2 are clustered round 3.1 while their responses to h0,h3 are clustered round 1.7 etc.

However it appears that Agent 6 is a little out of step with its community. Investigation of the details of its learning (its weights) shows that the problem occurs at the first stage i.e. with its encoding of the environment. We have deliberately used a high learning rate which in some cases (dependent on initial weights and the order in which environments were seen) causes convergence to non-optimal solutions. Agent 6 has in fact not identified all sources uniquely: the weights into its representation layer show that one neuron is identifying a particular pixel rather than the 4 pixels which comprise a source and has thus disrupted the learning which surrounds two sources. This has caused the language output corresponding to some environments to be higher than it should and that for others to be lower.

Source	Lower Bound	Upper Bound
h0	0.88	0.92
h1	0.95	1.00
h2	0.96	1.00
h3	0.86	0.92
v0	1.28	1.36
v1	0.58	0.61
v2	1.24	1.28
v3	0.56	0.59

Table 2: The agent language has synonyms for h0 and h3, h1 and h2 and, to a lesser extent, v1 and v3.

Also we note that, while there must inevitably be different weight vectors in the representation-to-language layer in different agents, there is a commonality between them: we find that e.g. if the weight to the neuron representing source 6 is emphasised (has larger absolute value) in one agent, all agents will emphasise the same source even though each has different representations of source 6. It does not matter to the agents that one source has more emphasis since they can still communicate about the source.

Thirdly, we report that our agent community is capable of puns: it has a set of synonyms as shown in Table 2: we see that the common output for h0 and h1 has a great deal of overlap[1] as do h1 and h2 and, to a lesser extent, v1 and v2. It should be emphasised that these results were achieved by a community *which was not trained on the independent sources*: environments were chosen randomly with each source having equal probability of being in any single environment. So typically any presentation of an environment to the community of agents would contain more than one single source yet the community still manages to create the language to describe the individual sources.

Finally, we stated that our language/output could be a vector of floating point numbers rather than a single floating point number. An example of the language weights of an individual agent with three language outputs are shown in Table 3. We see that two outputs will be identical since two sets of weights have converged to identical values. This is a somewhat surprising result since the randomness of the initial weights would suggest that three separate languages would be created. In this simulation, we find that every agent had two identical languages and a third different. Simulations have shown that e.g. with 7-dimensional language we often get two languages - one of which gives a positive output for each environment while the other gives a negative. Occasion-

[1] The results if we ignore the outlying neurons are even closer.

Language	v_1	v_2	v_3	v_4
1	0.794652	0.296743	0.247789	0.0501685
2	0.355686	-0.365324	-0.341216	0.307368
3	0.355686	-0.365324	-0.341216	0.307368
Language	v_5	v_6	v_7	v_8
1	0.259856	0.16591	0.137537	0.320294
2	-0.377908	0.347575	0.371135	-0.357434
3	-0.377908	0.347575	0.371135	-0.357434

Table 3: The three set of weights connecting the representation layer of an agent with its communication layer. Two languages (2 and 3) are equivalent.

Language	v_1	v_2	v_3	v_4
1	0.284492	0.0477801	0.270426	0.330045
2	0.355767	-0.34443	0.333977	0.363784
3	-0.355767	0.34443	-0.333977	-0.363784
Language	v_5	v_6	v_7	v_8
1	0.135112	0.272097	0.166876	0.783953
2	-0.381219	0.328808	-0.371822	-0.345289
3	0.381219	-0.328808	0.371822	0.345289

Table 4: The three sets of weights connecting the representation layer of an agent with its communication layer. One language is redundant in that the second is the negative of the third.

ally we see one set of weights converging to the negative of the other (Table 4). Both of these results suggest a limited number of possible languages, perhaps based on a deep structure of our artificial language. We believe that this last point is an extension of that discussed in [9] and is worthy of future investigation.

6 Experiment 2 - Different Forms of Life

The above set of experiments assumed that all the agents were equally capable of extracting the same independent sources from the environment: each agent had the same number of representation neurons and so had equivalent capacity to find the independent sources. However, we noted that some agents appeared out of step with their community because they had erred in their representation of the environment.

We continue this investigation by creating a community which have differing capabilities at the representation layer i.e. which have different number of neurons at that layer. We report on the results with different numbers of neurons in Tables 5 and 6. The number of neurons available to each agent in the population is drawn randomly from a uniform distribution on {7,8,9} in Table 5 and from a uniform distribution on {5,6,7,8} for the population in Table 6. These tables suggest that, when the community has differing capabilities, the development of a common language is difficult though not impossible since there is a general agreement about the value of the floating point number which best describes

Agent	h0	h1	h2	h3	v0	v1	v2	v3
$A_1(9)$	1.46	0.69	0.89	0.73	1.28	0.91	0.82	0.97
$A_2(8)$	1.27	0.77	0.96	0.87	1.34	0.94	0.82	0.76
$A_3(9)$	1.31	0.74	0.89	0.81	1.32	0.89	0.82	0.73
$A_4(8)$	1.31	0.80	0.97	0.88	1.36	1.02	0.82	0.76
$A_5(8)$	1.27	0.77	0.97	0.84	1.28	0.96	0.85	0.77
$A_6(9)$	1.24	0.80	0.95	0.88	1.32	0.88	0.91	0.77
$A_7(7)$	1.39	0.69	0.92	0.81	1.33	0.92	0.85	0.71
$A_8(9)$	1.37	0.71	0.78	0.91	1.29	0.93	0.88	0.69
$A_9(9)$	1.33	0.75	0.91	0.84	1.35	0.94	0.87	0.68
$A_{10}(7)$	1.36	0.73	0.87	0.84	1.31	0.90	0.85	0.76

Table 5: The outputs of a community of 10 agents labelled $A_1 - A_{10}$ on the independent sources in their environment. The figure in brackets after each agent refers to the number of neurons in its representation layer.

Agent	h0	h1	h2	h3	v0	v1	v2	v3
$A_1(6)$	0.92	0.76	0.93	1.31	0.87	1.30	0.85	0.91
$A_2(7)$	1.29	0.69	0.68	1.24	0.88	1.23	0.88	0.91
$A_3(7)$	1.23	0.93	0.71	1.02	0.80	1.27	0.89	0.92
$A_4(6)$	1.26	0.64	0.63	1.24	0.90	1.22	0.85	0.81
$A_5(7)$	1.25	0.76	0.76	1.18	0.91	1.20	0.93	0.91
$A_6(7)$	1.27	0.71	0.77	1.05	0.81	1.17	0.89	0.93
$A_7(5)$	1.30	0.93	0.86	0.78	0.83	1.36	0.86	0.83
$A_8(8)$	1.25	0.74	0.76	1.19	0.89	1.23	0.92	0.90
$A_9(5)$	1.29	0.65	0.61	1.26	0.95	0.88	0.92	1.05
$A_{10}(5)$	1.28	0.83	0.98	0.79	0.81	1.37	0.84	0.86

Table 6: The outputs of a community of 10 agents labelled $A_1 - A_{10}$ on the independent sources in their environment. The figure in brackets after each agent refers to the number of neurons in its representation layer.

the environmental input. Note that when we compare Table 5 with Table 6, we see that in the latter the spread of outputs is greater i.e. there is less agreement on the shared language. This suggests that the wider the range of representational capabilities, the harder it is to build that common language.

This leads to the investigation in Experiment 3: we now posit that, in a community whose agents' fitness is dependent on the ability to communicate, those agents who have similar representational capabilities as their peers are at an evolutionary advantage compared to those who have differing representational capabilities *even when such agents have superior capabilities.* Thus we investigate whether there may be an evolutionary drive towards common capacities if the ability to communicate with one's community is a measure of one's fitness in one's community.

7 Experiment 3 - Evolving a population with common language capabilities

Finally we wish to experiment with a population to investigate whether aptitude for a common language will in fact favour evolution to a common representation capability. We use a simple genetic algorithm to determine the number of representation neurons which we define in a four bit vector. Therefore the number of representation neurons may vary between 0 and 15. We use one point crossover and no mutation which with this very small population (10 agents) is liable to lead to very fast convergence. Our fitness function, though, is not directly a function of the representation layer but is measured

in terms of how close the individual's language is to the population's language on average.

Therefore the fitness of agent A_i is given by

$$f(A_i) = \frac{1}{a + error_i}$$
$$\text{where } error_i = \sum_{s \in S} |E_{l_i} - s_i|$$

where a is a constant, set to 0.5 in the simulation reported here and as before we are using the mean field approximation E_{l_i} as the average representation of the current environment in the i^{th} symbol.

Beginning with that population whose responses to the various independent components of their common environment are shown in Table 6, we find that after one generation our population has 5,5,5,13,7,5,5,9,5 and 5 representation neurons respectively while after 2 or more generations, the entire population is composed of agents whose representation layer is composed of 5 neurons. This community has a common ability to represent the environment, though of course they do so in very different ways depending on the actual environments which they met while the representation layer was learning and the initial conditions of their weights.

Repeat experiments with larger populations of 50, 100 and 1000 agents similarly converge within a small number of generations to populations of agents with uniform numbers of neurons in the representation layer.

In the experiment detailed above, and in some of the repeat experiments, we note that the representation capacity is not optimal - we actually require 8 representation neurons to identify the 8 sources. However, the populations were evolved according to the view that the fitness of an individual was inversely proportional to his inability to communicate with his community. A more sophisticated fitness function would take account of both the representational capability of each agent as well as its capacity to communicate.

8 Conclusion

We have reported simulations which have investigated the creation of a language in a population of simple agents existing in an environment which has some statistical regularity. In our simulations, we have enabled the agents to learn about the environment from samples drawn from the environment. Therefore each agent learns a different representation of the environment depending on both the initial values of the weights of the representational layer and the actual samples of the environment met while this representational layer was learning. We have shown that when such agents share a common ability to represent the environment, they can quickly learn to communicate about the environment, even though they do not have identical representations about the environment.

We have further shown that when there is a difference between the capabilities of the agents, they can communicate about the environment but the communication becomes vaguer/more diffuse. We have used this fact to allow a population of agents to evolve under conditions where the fitness of the agent is determined by the agent's ability to agree a common communication code with its peers. Such populations evolve to share a common representation capability though not a common representation.

We believe that there are a number of potentially fruitful extensions to the work described herein. Future work will include

- We have described the development of a small number of possible languages for any given community in a given environment. It is of great interest that there seems to be a small number of possible convergence points for the language.

- Our simulation of the evolution of a population converging to a population sharing a common representation capability could be enhanced by adding an additional constraint that the population should have as strong a representational capacity as possible.

- We have required each agent to have a transmission capability without considering a corresponding reception capability. i.e. we have viewed the emission of a communication signal as the essential requirement thereby implicitly suggesting that the receipt of such a signal is a separate issue. It would clearly be simple to create a separate (third) decoding network to make the intersubjective communication to internal representation mapping. However we believe that we could more fruitfully amend our language creating network by using a backpropagating type algorithm which autoassociates with an objective function which both minimises the mean square reconstruction error and also has a Lagrange multiplier for the criterion of optimising representation.

Acknowledgement We would like to gratefully acknowledge long and fruitful discussions with M. Bronte-Stewart.

References

[1] H. B. Barlow. Unsupervised learning. *Neural Computation*, 1:295–311, 1989.

[2] P. Földiák. *Models of Sensory Coding.* PhD thesis, University of Cambridge, 1992.

[3] C. Fyfe. Pca properties of interneurons. In *From Neurobiology to Real World Computing, ICANN 93*, pages 183–188, 1993.

[4] C. Fyfe. A neural net for pca and beyond. *Neural Processing Letters*, 6(1):1–9, 1997.

[5] John Hertz, Anders Krogh, and Richard G. Palmer. *Introduction to the Theory of Neural Computation.* Addison-Wesley Publishing, 1992.

[6] A. Moukas and G. Hayes. Synthetic robotic language acquisition by observation. In *From Animals to Animats 4 - Proceedings of the Fourth International Conference on Adaptive Behaviour*, 1996.

[7] J. Noble and D. Cliff. On simulating the evolution of communication. In *From Animals to Animats 4 - Proceedings of the Fourth International Conference on Adaptive Behaviour*, 1996.

[8] E. Oja. Neural networks, principal components and subspaces. *International Journal of Neural Systems*, 1:61–68, 1989.

[9] M. Oliphant. The dilemma of saussurean communication. *BioSystems*, 37:31–38, 1996.

[10] G. M. Saunders and J. B. Pollack. The evolution of communication schemes over continuous channesl. In *From Animals to Animats 4 - Proceedings of the Fourth International Conference on Adaptive Behaviour*, 1996.