How a neural net grows symbols

Proceedings of the Seventh Australian Conference on Neural Networks, Canberra, 1996, pp. 91-6

James Franklin
School of Mathematics
University of New South Wales
Sydney 2052
J.Franklin@unsw.edu.au

April 8, 2005

Abstract

Brains, unlike artificial neural nets, use symbols to summarise and reason about perceptual input. But unlike symbolic AI, they "ground" the symbols in the data: the symbols have meaning in terms of data, not just meaning imposed by the outside user. If neural nets could be made to grow their own symbols in the way that brains do, there would be a good prospect of combining neural networks and symbolic AI, in such a way as to combine the good features of each.

It is argued that the secret of growing symbols in neural nets lies in cluster analysis. Algorithms for clustering, many of them naturally implementable in neural hardware, would produce clusters, which are discrete entities summarising data that have all the properties of symbols.

1 Introduction

The war between symbolic artificial intelligence and its neural net rival continues because each has strengths that the other lacks, and it has proved impossible to combine them successfully. It is agreed that symbolic systems work well on discretely structured problems, like chess, and give a transparent understanding of what they are doing, which allows their use in new situations through adding and deleting rules. But it is difficult to make them adaptive to data, especially in situations where there is only data to go on, and almost no understanding via rules, such as face recognition. Scaling up from toy to real problems is also hard. Neural nets, on the other hand, are strong where symbolic AI is weak, and vice versa. They adapt easily to data, but the black-box nature of their processing makes it very difficult to understand what they do, and hence to improve it, or adapt it to a different problem.

Naturally, one would like to combine the two approaches, to take advantage of the strengths of each. Unfortunately, current attempts, though not wholly unsuccessful, generally find themselves saddled also with the weaknesses of both approaches. Machine learning, for example, will certainly produce rules from data, but in all but the simplest problems, there are so many rules that the system is no more comprehensible, and no more adaptable to new situations, than neural nets are. And attempts

to extract rules from trained neural nets have also not proved very successful. [**?**] [**?**] They face a fundamental problem in tying the terms in the rules to data. The meaningfulness of a rule "If X then Y" depends on making the symbol "X" meaningful in terms of the data. Into how many "X"s should the space of possible data points or inputs be cut up? Should some "X"s cover more of the space than others? What happens at the joins? And so on. These "symbol-grounding" problems, concerning how to tie symbols to their meaning in data, have been too often ignored, and their neglect has vitiated attempts to combine symbolic and neural approaches to AI. (Fuzzy rule systems tuned by data are more hopeful, but they are close to the approach to be described [**?**]).

It is clear that the brain has solved these problems. It is clear also, at least in a very general way, how it has done it. Somehow, the brain does manipulate discrete symbols in a way that makes those symbols meaningful in terms of the flow of experience. The discrete symbols are grounded in continuous experience.

It is further clear that this will be a very difficult feat to imitate. But the promised payoff is large. This is a first attempt.

2 The Problem: Experience into Symbol

What is sought is an algorithm which takes experience (perceptual experience, or a list of vectors in feature space, or raw images) and outputs symbols – discrete entities which attach to items in the experience which "naturally" go together. All cat experiences should get one label, all dog experiences another. Then those labels can be used in rules. If such an algorithm could be found, it would solve three problems:

- The engineering problem of how to link neural nets with rule-based AI, in such a way that the symbols in the rules are correctly tied to the data.

- The "symbol grounding" problem in cognitive science, or the philosophy of language; that is, the problem of how the meaning of a word is connected to the relevant experience – what "cat" has to do with the experience of cats.

- The data reduction problem – that is, how to explain, and imitate, the brain's ability to operate with huge quantities of noisy (perceptual) data, by recognising in it small numbers of persistent items.

These problems are each hard, but have much in common: they all require for their solution a principled method of reducing a large continuous space of experience to a set of discrete items, while losing as little information as possible. The aim of this paper is to cast light on the first (engineering) problem, by analysing what is known about the other two. They are cognitive science problems, but ones which are amenable to a certain amount of purely abstract analysis in terms of data-processing algorithms. The problems are examined in turn.

3 The Symbol Grounding Problem

The symbol grounding problem, posed in Harnad's famous article, [**?**] concerns the old chestnut of how words get their meanings. How does a symbol, internal or external, like "cat" comes to refer to cats, not dogs? "How can the semantic interpretation of a formal

symbol system be made *intrinsic* to the system, rather than just parasitic on the meanings in our heads? How can the meanings of meaningless symbol tokens, manipulated solely on the basis of their (arbitrary) shapes, be grounded in anything but other meaningless symbols?" "Grounding" must be a process which takes experience, of cats or other things, and issues in some entity sufficiently discrete to be a representation of the class of cats, and to which a symbol can be attached. That is, it must perform some sort of *clustering* on (preprocessed) experiences, which recognises that cat-experiences have sufficient similarity, and sufficient dissimilarity to dog-experiences, to form a natural category, or cluster. Of course, there is also a grounding problem at a lower level. The world appears to be more or less uniquely divided into things, which themselves are more or less uniquely divided into kinds. The division of experience into persistent objects – should we think of it as the symbol grounding problem for proper names? – is the prior one. The grounding problem for common nouns, the one chiefly considered by Harnad, takes as given the persistent objects and their features, and aims to create discrete and natural clusters of the objects, considered as vectors in the multidimensional feature space.

Since clusters can themselves form clusters, cluster analysis gives the potentiality for automatic construction of the **is-a** hierarchies that are crucial to knowledge representation: "a cat is a (kind of) animal" simply because the cluster of cats is a subset of the cluster of animals.

And its being a cluster is explicitly represented, by the label that is given by the cluster algorithm to all and only those points in the cluster. Thus cluster analysis of this hierarchical kind is capable of extracting a discrete structure, a tree, from essentially continuous or inchoate input. The belief that this is somehow impossible seems to be behind some of the assertions that neural nets do not naturally form the structures found in, for example, language, because "Connectionist architecture recognizes no combinatorial structure in mental representations". [**?**], p. 49.

Suppose we ask how human cognition differs from the "thermometer model" of knowledge. [**?**] A correctly-working thermometer records or tracks the ambient temperature, in that its mercury reading goes through a time pattern of states which is literally identical to the time pattern of variations in the ambient temperature. The totality of readings – and by extension, an individual reading – can therefore be said to represent temperature. Counterfactuals are supported: if the temperature had been different, the reading would have been different. The model also makes sense of perceptual error, at least up to a point: if the readings are correct for all temperatures up to 40^oC, but wrong for the rare cases when the temperature is over 40°, then it can be said that the thermometer is generally reliable, but in error when the temperature is over 40°. The thermometer model, especially if we think of a thermostat attached, does provide an adequate model of knowledge for a sufficiently simple organism, which responds to its environment in a simple and pre-programmed way. It is still a reasonable model for knowledge in a trained artificial neural net, which also responds to inputs in the same automatic way as a thermometer responds to temperature (although the way it acquires the ability to do so is very different). Similar remarks could be made about any dynamical systems view of cognition. [**?**]

The feature of cognition in humans and the higher animals that is not captured by the temperature model is fiction. This means fiction in a wide sense, including expectation and anticipation (of, for example, what other cars on the road will do), planning (forseeing the results of one's own actions) and history (inferring the probable nationality of Homer's

mother, for example) as well as the construction of whole fictional narratives like novels. The fictive faculty must be able to represent scenarios not presently actual, including ones that may attain actuality in the past or future. A thermometer cannot do fiction. It is too mired in the present, the actual, to cope with the sphere of the possible. It is much the same with a trained artificial neural net, as the symbolic AI experts regularly point out. Although a net can, in a sense, have memory traces from its actual past, it is useless on any task other than the exact one on which it has been trained (or at least, it is impossible to tell on which tasks it will work). It does not allow the *transfer* of learned expertise to a new domain. [**?**] Now the possible, or at least the epistemically possible, consists of *recombinations* of items acquired from experience of the actual. In an older idiom, the "imagination", the faculty of fiction, is the organ of "recombining and dividing", [**?**] inhabited by such entities as winged horses and golden mountains. The thermometer cannot do anything like this, simply because it does not identify items at all. Similarly, a neural net trained on data cannot divide its ability into parts and reuse the appropriate parts in other circumstances.

The emphasis, then, in getting beyond merely causal tracking to real cognition, has to be on first dividing the flow of perception into meaningful "items" which are capable of recombination, cutting and pasting, omission and reincorporation. This brings us back to to symbol-grounding problem in Harnad's sense, if it is agreed that a useful "item" should have at least a measure of internal homogeneity, compared to its surrounds.

4 The Data Reduction Problem

Natural intelligence operates in a data rich environment. This is not the case for standard symbolic AI. The preference of symbolic AI for calculation over either remembering or learning has been widely recognised as explaining its success in structure-rich but data-poor situations like chess, but lack of success as a flexible model of, for example, common-sense reasoning.

The situation of natural intelligence is very different. It is presented with a massive flow of perceptual information (over 100 million receptors in each retina, responding many times per second, to take just the visual system). It must "drink from the firehose of data", [**?**] that is, achieve its goal in real time – in fact, in better than real time, since it must predict the state of the world ahead of time. The flux of information contains gross structures, the "affordances" like the continuity of the optic flow, [**?**] that contain information about the world and the organism's changing relationship to it. That information does not lie on the surface: a great deal of processing of some sort is needed to extract, for example, an object's 3D motion from the flow of projections on its retina. The immediate problem is where to start, or how to even begin to reduce the mass to some manageable and meaningful quantity. [**?**] Tasks one will need to perform, at some stage between registering perceptions at the retina or other sensory extremities and object recognition, include:

- Identifying an object across time, that is, across "frames", despite gradual changes in its and the organism's position, and despite noise and occlusions.

- Identifying regions; for example, recognising that foreground pixels should go to-

gether, and separating them from background.

- Distinguishing sudden from gradual changes of colour, texture, loudness and so on; with the definition of "sudden" being appropriately sensitive to context (for example, a leaf should stand out against a plain background, but not against a background of other leaves).

5 The Answer: Cluster Analysis

What is sought, then, are algorithms that perform data-reduction on huge data-sets in a fast and robust manner. Ideally, they should be parallelizable in a way that suits a neural network architecture. The kind of algorithm that will eventually be able to extract the correct structure from a huge data flow must be like present-day cluster analysis.

Figure 1: : Two clusters

The essential idea of cluster analysis, as normally understood, is simply described. [**?**] [**?**] It takes unlabelled points in a space. It clusters or "clumps" together those which lie close together, and are separated by empty space from other clusters. It outputs a labelling of all the data points, identifying which cluster they are in. Thus, the two-dimensional data in figure 1 fall naturally into two clusters.

The space in which the points lie may be a physical space. But cluster analysis is actually most used in "feature spaces", where the dimensions represent features of objects, and a point is the aggregate of features that a particular object has. Close points thus represent similar objects. Cluster analysis is used to find natural groupings for the classification of neuroses, shoplifters, markets and so on. The tendency of cluster analysis to concentrate on such ill-defined subject matters, where there is a suspicion that there are no clear clusters to be found, has led to cluster analysis having a low profile among statistical methods, and a generally poor reputation. But that is no reason to doubt its applicability to perception.

Since there are many algorithms that perform clustering, let us attempt a high-level specification of the task. Any algorithm that takes unlabelled data on which there is some measure of similarity or distance between points, and apportions them to groups such that the within-group similarity is high, compared to the between-group similarity, is a form of cluster analysis. Normally, one wants the algorithm to issue also in a division of the space, not just of the input points (since one wants to say of a new point which of the clusters it would be in).

It is important that cluster analysis is thus specified at a higher level than the algorithmic – at the level of "what the system does", not "how it does it". It is a level somewhat vaguely characterised in the cognitive science literature, under a variety of names such as

the "ecological" or "intentional" or "semantic" level, or the "computational" level [**?**], all of them intended to name a level in which the task of a system is described "in ways that are noncommittal on how the system does it". [**?**] There is a better understanding of this level in certain other disciplines, especially those involving the design of systems, since analysis of what the system has to do naturally precedes the detailed design. "Task analysis" in factories is one example. [**?**] Research has gone furthest in the computer scientists' "formal specification", which has the aim of specifying precisely the task that software is to perform, before any writing of it begins. [**?**] Working at this level makes it possible to discuss in more general terms the possible applications of cluster analysis, without needing to specify what clustering algorithms are being used. More importantly, it allows us to classify as cluster analysis various algorithms developed for quite other purposes.

It is argued, not merely that cluster analysis might be helpful for the problems of early perceptual grouping and of symbol grounding, but that the nature of the problems means that *any* solution to them must be *some* form of cluster analysis. They all involve forming a discrete object out of a cluster, that is, a mass of neighbouring data points that are all reasonably well separated from other data points.

There are many subtleties and matters to be considered in performing clustering, which will be mentioned briefly later. But let us assume the existence of algorithms meeting the specifications just outlined, and inquire what tasks in early perceptual organization could be performed by them.

Consider a large black spot seen against a white background. To oversimplify, a retina could report this situation to its brain by transmitting many 3D vectors (x, y, c), where x and y specify the location of a pixel and c specifies its colour value, white or black. The brain's task is to "stick together" the black ones and also the white ones, to construct two coherent objects, spot and background. It must, that is, join together, or cluster, the black dots, which form a tight cluster in the 3D space. That cluster is well separated from the cluster of white dots, as the value in the last (colour) coordinate is quite different. [**?**]

It is to be observed that this "region growing" approach [**?**] to identifying patches is quite different to one that relies wholly on detecting edges and joining edge portions. Clustering pixels is a method that generalizes automatically to more realistic cases where the colour value is more complicated (when it is grey scale, or red-green-blue colour, or even includes more complicated continuously varying quantities like shine and texture). Take for example a somewhat textured patch against a plain background, such as the moon. The moon pixels are all still close to each other, relative to the background pixels, and are easily identified as one object. A cluster method is in principle more robust to noise than edge detection methods, since fuzziness in the edges simply means the clusters are not so well separated; nevertheless there will be few points between the clusters, and a reasonable cluster algorithm should have plenty of information to recognise the clusters. The red spot and other swirls on Jupiter have enough internal homogeneity to be picked out as coherent objects, though most are not recognisable as any describable shape, and have no clear edges. A cluster analysis approach, that "grows" homogeneous regions, is also suggested by the many psychological experiments in which the visual system imposes contours even when they are not present in real luminance. [**?**]

To recognize objects across time, one simply adds a time coordinate to the above example, and performs clustering in the resulting 4D space. There is the opportunity to identify objects with temporal gaps, such as balls that

disappear behind chairs and later reappear, and also to join parts of objects that move parallel to each other – that have a "common fate", in Gestalt terminology. The pixels of the before and after temporal parts of the continuant differ in their time coordinate, but if agreement is sufficient in other coordinates, such as colour and speed (compared, as always, with the background), then there is some hope of identifying the parts, if they are not too distant in time. This also gives the opportunity to correct any noise in the individual frames by the pixel values in the frames immediately before and after. This is a general phenomenon, which makes work with large data sets essentially different from attempting to "scale-up" AI methods that work on toy problems: the algorithms used must be noise-loving, so as to take advantage of the noise-correction capabilities of having many modalities available simultaneously. As far as possible, one should identify everything simultaneously, not work with one data type at a time and wonder how to integrate the answers later. [**?**] Cluster analysis does this naturally; few other AI methods do.

One has to observe, also, that even before the symbol-grounding problem is posed, one has to identify the symbols themselves in the flow of perception. It is all very well to imagine that a child learns the meaning of "cat" by hearing the word said many times in the presence of cats, but there are two problems the child must solve first. What portions of the ambient world is it supposed to recognise as cats? And what portion of the sound stream is it supposed to recognise as "cat"? Any "associative learning" presupposes identified items in both the world and the sound stream. In view of the difficulty of making commercial speech recognition systems that can even segment continuous speech correctly, it is something of a mystery how children learn to do so. The word is a particularly difficult unit to identify, and very young children prefer the syllable and the longer continuous sound. They gradually construct the word from syllables that co-occur frequently, and use the resulting words to gradually learn to find the meaningful units in speech. [**?**] That is, blocks of similar sound profiles heard at different times are clustered to form a coherent entity, which can be recognised and extracted from the sound stream later.

6 The Cluster Analysis Smorgasbord

Having specified cluster analysis at such a high level, it is not surprising that there is an enormous diversity of actual clustering algorithms. Which does the brain use? Little is known. Which should be recommended to a neural net engineer? A great number of issues have to be considered in deciding between cluster algorithms. There is space here only to indicate briefly some of these.

Some of the algorithms are bottom-up (that is, they begin by joining close data points), while others are top-down (they begin by looking at ways to divide the whole data cloud). Some are fast, some are slow. Naturally, the fast ones lose something, in that they do not check for good clustering, or consider various alternatives. Some of the faster ones give results sensitive to the order of presentation of the data; "leader" clustering, for example, takes the first points seen as cluster centres, and compares later ones with them. Some algorithms allow fuzzy clusters, or overlapping ones, or allow a few "outliers" to belong to no clusters. Some have a preference for certain shapes of clusters, for example, elliptical or convex shapes (as do humans [**?**]). Some decide on the appropriate number of clusters, some have to be told. Here, just one issue will

be taken up: which cluster algorithms are well adapted to implementation in neural network style hardware?

There is some problem, admittedly, in characterising what is rightly called a "neural network". It is possible, for example, to implement clustering that fits a mixture of elliptical data clouds in a kind of neural network, if one is prepared to allow the neurons to have (trainable) Gaussian receptive fields (as opposed to the usual simple dot product of the incoming vector with the fan-in weight vector). [**?**] To some extent, the problem is simply one of deciding the appropriate definition of the "matching" of two vectors. In cluster analysis, one normally takes the Euclidean distance between them, but in the usual neural nets, one takes distance in projective space instead, so that one ignores differences of scale: vectors match if one is a multiple of the other. "Radial basis function" networks show that one can use the distributed-processing style of neural network architecture with the usual Euclidean definition of matching. It is true that one can therefore illustrate virtually any algorithm that involves comparison of vectors with a "neural" diagram, and call the algorithm a neural net in the hope of securing a larger grant. Still, it is not wholly dishonest to do so; the essence of neural networks really is massively parallel processing based on the results of matching of vectors.

Of special interest are the ART1 algorithms and their descendants. [**?**] [**?**] Originally cast in a neural network framework, they have been seen more accurately as implementations of adaptive leader clustering algorithms. [**?**] Combining the speed of leader clustering with a reasonable adaptivity to the data, these algorithms and slight modifications of them have had a number of successes in classifying large and awkward data sets, such as fingerprint images and Chinese characters. While the clustering performed by ART-type networks is not hierarchical, certain close relatives of them can produce hierarchies: by training a top-level net on all the data, and using its output to divide the training data into clusters which in turn are used as the training sets for several lower-level nets, one can obtain a partially parallelizable method for identifying multilevel statistical structure in novel data. [**?**]

Obviously, there is much more to be done in the area of neural net cluster algorithms, but there is already a solid beginning.

7 Conclusion

Cluster algorithms implementable in neural nets are available. They can identify recombinable items, or "symbols", in the flux of sensation. The next task is to use their outputs in rules, to imitate the range of tasks that the brain can perform well, but current AI cannot. To begin, one can learn relations between symbols by associative learning – now that there are items to associate.

Acknowledgments

I am grateful to Hugh Clapin for valuable discussions.

References

[1] H.M. Abbas & M.M. Fahmy, "Neural networks for maximum likelihood clustering", *Signal Processing* 36, pp. 111-26, 1994.

[2] S. Abe & M.-S. Lan, "A method for fuzzy rules extraction directly from numerical data and its application to pattern classification", *IEEE Transactions on Fuzzy Systems* 3, pp. 18-28, 1995.

[3] R. Adams & L. Bischof, "Seeded region growing", *IEEE Trans. on Pattern Analy-*

sis and Machine Intelligence 16, pp. 641-7, 1994.

[4] J. Ambros-Ingerson, R. Granger & G. Lynch, "Simulation of paleocortex performs hierarchical clustering", *Science* 247, pp. 1344-8, 1990.

[5] D.M. Armstrong, *Belief, Truth and Knowledge.* Cambridge, 1973, pp. 166-71.

[6] L.I. Burke, "Clustering characterization of adaptive resonance", *Neural Networks* 4, pp. 485-491, 1991.

[7] G.A. Carpenter & S. Grossberg, "ART2: Self organization of stable category recognition codes for analog patterns", *Applied Optics* 2, pp. 4919-4938, 1987.

[8] G.A. Carpenter & S. Grossberg, eds, *Neural networks for vision and image processing.* Cambridge, Mass, 1992.

[9] A. Clark, *Associative Engines.* Cambridge, Mass, 1993, ch. 4.

[10] P.A. Devijver & M.M. Dekesel, "Cluster analysis under Markovian dependence with application to image segmentation", in *Classification and Related Methods of Data Analysis,* ed. H.H. Bock. Amsterdam, 1988, pp. 203-217.

[11] B. Everitt, *Cluster Analysis.* 3rd ed, London, 1993.

[12] J.A. Fodor & Z.W. Pylyshyn, "Connectionism and cognitive architecture", *Cognition* 28, pp. 3-71, 1988.

[13] J.J. Gibson, *The Senses Considered as Perceptual Systems.* Boston, 1966.

[14] J.V. Goodsitt, J.L. Morgan & P.K. Kuhl, "Perceptual strategies in prelingual speech segmentation", *J. of Child Language* 20, pp. 229-252, 1993.

[15] W.B. Goh & G. Martin, "Model-based multiresolution motion estimation in noisy images", *CVGIP: Image Understanding* 59, pp. 307-319, 1994.

[16] S. Grossberg & E. Mingolla, "Neural dynamics of perceptual grouping: Textures, boundaries and emergent segmentations", *Perception and Psychophysics* 38, pp. 141-171, 1985.

[17] S. Harnad, "The symbol grounding problem", *Physica D* 42, pp. 335-346, 1990.

[18] D.C. Ince, *An Introduction to Discrete Mathematics and Formal System Specification.* Oxford, 1988, ch. 2.

[19] A.K. Jain & R.C. Dubes, *Algorithms for Clustering Data.* Englewood Cliffs, NJ, 1988.

[20] B. Kirwan & L.K. Ainsworth, eds, *A Guide to Task Analysis.* London, 1992.

[21] J.K. Kruschke, "ALCOVE: An exemplar-based connectionist model of category learning", *Psychological Review* 99, pp. 22-44, 1992.

[22] D. Marr, *Vision.* New York, 1982, pp. 24-5.

[23] J. Pomerantz, "Perceptual organization in information processing", in M. Kubovy & J. Pomerantz, eds, *Perceptual Organization* Hillsdale, 1981, pp. 141-180.

[24] R. Port & T. van Gelder, eds, *Mind as Motion: Explorations in the Dynamics of Cognition.* Cambridge, Mass, 1995.

[25] K. Sterelny, *The Representational Theory of Mind.* Oxford, 1990, pp. 44-6.

[26] R. Sun & L. Bookman, "How do symbols and networks fit together?", *AI Magazine* 14 (2), pp. 20-23, Summer 1993.

[27] Thomas Aquinas, *Summa theologiae* bk I q. 78 art. 4.

[28] G.G. Towell & J.W. Shavlik, "Knowledge-based artificial neural networks", *Artificial Intelligence* 70, pp. 119-65, 1994.

[29] M.M. Waldrop, "Learning to drink from a firehose", *Science* 248, pp. 674-5, 1990.