ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 315-320

Inductive learning in animat-based neural networks

Mark Wexler
Laboratoire de la Physiologie de la Perception et de l'Action
Collège de France/C.N.R.S., 11 pl. Marcelin Berthelot
75231 Paris Cedex 05, France
wexler@lpthe.jussieu.fr

Abstract. Inductive learning is hard because any interesting hypothesis space is very large, and formal biases either underconstrain or overconstrain the problem: the bias/variance dilemma. I suggest a new form of inductive bias, in which a simple embodied agent—an animat—is constrained to represent the intermediate states of an abstract problem through its movement. This is illustrated with a network-controlled animat that learns n-parity. Unlike other general learning devices, such as disembodied networks, it learns from very few examples and generalizes correctly to previously unseen cases. The relationship to recent trends, such as embodied robotics, symbol grounding, and external representation is briefly discussed.

1. Induction

In artificial inductive learning, the machine is trained on only part of the set of possible input-output pairs; once it reaches some criterion of success on this training set, the machine is tested for "correct" generalization on the remaining test set. The number of generalizations consistent with the training set is usually very large. The trouble is: no purely formal, syntactic criterion can systematically pick out what we consider as the "correct" generalization out of this large field of possibilities. From a strictly unbiased, objective point of view any generalization that is consistent with the training set is as good as any other. Nevertheless, if all generalizations are equal, some generalizations are more equal than others: they strike *us* as more clever or perspicacious, and it is *these* generalizations that we want our learning machine to prefer.

The usual solution is to endow the learning system with systematic inductive bias. A typical form of bias is a norm, "badness", on the hypothesis language—the size of the classification tree or LISP expression, for example. Induction is then just a form of optimization or search, an attempt to minimize the badness of the hypothesis while maximizing consistency with the training set.

The practical problem with this picture of biased learning is the so-called bias/variance dilemma. Not enough bias underconstrains induction, while too much bias makes the system ad hoc or overly specialized, and deprives it of

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 315-320

plasticity. It is very difficult to find robust forms of bias that avoid these two extremes inside particular problem domains, and crossing domain boundaries is even more difficult. The result is that learning systems must be restricted to specialized domains, and the inductive bias, to be put in by the programmer—or by nature—carries a heavy load of specific information. Such a solution to the problem of induction begs the cognitive question: if bias is to be put in by hand, whose hand is it? (If your answer is natural selection, keep in mind that phylogenetic learning is at least as difficult as the ontogenetic kind.) Moreover, both nervous system development and learning behavior are more flexible than such a picture would suggest.

The problem of induction has recently been discussed from the point of view of intermediate representations (Kirsh 1992, Clark and Thornton 1997). In the inductive learning of patterns, categories or rules, the distinction is made between low-order regularities that are present directly in the training data (such as conditional probabilities between individual input and output bits that are close to 0 or 1), and more subtle higher-order regularities that can only be expressed in a richer language and in terms of the lower-order regularities (such as relational properties, etc.). The lower-order regularities can certainly be picked up by formal, fairly unbiased techniques. As for the higher-order regularities, if the learner is to succeed it must first re-represent the raw input in order to make the more subtle pattern manifest. The problem of perspicacious re-representation is thus seen as the critical component of higher-order learning.

Multi-layer neural networks have become popular mainly due to the assumption that the intermediate layers will somehow carry out this re-representation, and the magic is supposed to be that this is induced by means of a completely mindless procedure, hill-climbing. Clark and Thornton (1997) have shown that this is not so for a number of difficult learning problems; the one I will discuss is n-parity, as I also use this problem in my toy model (see below). This mapping is hard to learn from examples because they provide no raw statistical information: all the conditional probabilities between input and output bits are 0.5. It is well known that multi-layer feed-forward networks can "learn" parity by means of hill-climbing—but this is only when they are trained on *all* the 2^n input-output pairs. Reproducing the training set is necessary but not sufficient for having a concept or a rule; simple memorization will lead to the same result, and we would not then say that the memorizer has learned a rule, because (among other reasons) no re-representation of the input has taken place. A sharper test for having learned a rule is of course correct generalization to previously unseen cases.

Clark and Thornton have found that *no* training algorithm on *any* network architecture leads to networks that generalize correctly to previously unseen cases of parity; in fact, even in the best cases it suffices to withhold a very small fraction of the problems from the training set for generalization to fail completely. (Other, non-network general algorithms fail just as badly.) Practically, it shows how bad general hill-climbing is at generalization: many interesting problems can be expected to contain embedded parity or more complicated

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 315-320

higher-order structures. More fundamentally, it shows that, at least in this case, that when a network learns to reproduce input-output pairs, what it has actually learned is entirely unlike the rule that *we* have used to arrive at the training set. Indeed, what reason do we have to suppose that a neural network, trained by hill-climbing, would generalize beyond the training set in the way *we* would?

2. Bias through action

I would like to suggest a novel form of inductive bias that may overcome some of the difficulties of inductive learning (Wexler 1996). Namely, the system is embedded in an physical-like environment (real or simulated), and allowed to act on this environment and to perceive its surroundings, including the results of its own actions. The system's action, besides serving purely pragmatic ends, may also ground what are generally considered "higher" cognitive functions. The system may re-represent the states of an abstract learning problem through, for example, its movements. Such re-representation may channel learning and generalization in what we consider as the "correct" direction. This somewhat paradoxical notion can be fleshed out by the following toy model.

3. Induction in an animat

The goal of the toy system to be presented is not to serve as a model of anything, but as the simplest illustration of the role that action and embodiment may play in inductive learning.

We begin with a highly simplified—but not altogether unrealistic—embodied creature, an animat. This particular animat lives on a 2-dimensional plane; its external "world" state is simply its position on the plane and its heading. Time is taken as discrete. At each tick the animat's muscles receive a motor command (the output of a neural network), giving the distance to travel forward and the angle by which to turn. The sensory input (read by a second network) consists of the distance to a fixed landmark, and the angle between the animat's heading and the landmark (given as a sine and cosine, to avoid discontinuities).

The goal is for the system to learn n-parity. To this end there are two neural networks that connect the animat to its task. The animat always starts out in the same position and orientation (at (1,0) where the landmark is at the origin, and facing north). The parity problem is fed bit-by-bit to an input-motor network (a one-layer perceptron), whose job is not to give the answer to the problem but to issue motor commands to the animat. The sensory signals from the animat are fed to a second, sensory-output network (also a one-layer perceptron), which at the end of the presentation of the problem is supposed to output the answer—the parity of the input string. Having no other representational methods at its disposal (such internal recurrent connections), the system is obliged to represent the problem, and to keep track of the intermediate results, by means of its action.

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 315-320

The networks are trained by means of a genetic algorithm. Each experiment proceeds as follows. A fraction f of the 2^n problems are assigned to the test set, and are never used in training (the training set always has an equal number of even and odd cases). In each generation of the GA each member of the (initially random) population is evaluated on the $2^n(1-f)$ training problems. (The weights and thresholds of the networks are coded as 10-bit strings in the genome.) The score on each problem is the absolute value between the output of the sensory-output network and the true answer (since logical 0 is represented by -1.0 and 1 by $+1.0$, the best score is 0, the worst is 2, and 1 is chance level); the score on the entire training set is the mean of the scores on each problem. The population (size 50, 10 bits/weight) is evolved by both 2-point crossover and mutation, with rank-based fitness. The experiment was stopped as soon as a member of the population reached criterion on the training set, a score of 0.001 or below. (Occasionally experiments ran over 200 generations without reaching criterion; these were discarded.) The best member of the population is then tested on the $2^n f$ problems in the test set, which, I stress, the population had never seen during its training. The average score on the best population member on the generalization test is the score for the experiment. I ran 100 experiments for each value of f and averaged the scores, where new training and test sets were chosen for each experiment. (Further details available on request. All techniques used were very generic. The results were not sensitive to small variations in the parameters.)

For the control, I wanted to make things as hard as possible by comparing the generalization performance of the embodied systems with that of the *best generalizing* disembodied networks. As shown by Clark and Thornton (1997), feed-forward networks for the non-temporal version of the problem are miserable at generalization. For the temporal version feed-forward networks won't do, as they do not preserve state, and therefore at least some recurrent connections are required. After experimenting with a number of architectures, I found that simple recurrent ("Elman") networks generalize best. Within this class, 1-a^*-b-1 architectures are best (* denotes a recurrent context layer), and as long as b is not too large the performance depends essentially on a; $b = a$ seems a good choice. The three best architectures are 1-2*-2-1, 1-3*-3-1, and 1-4*-4-1. These disembodied networks were trained by exactly the same method as the embodied systems. It should be noted that the disembodied systems got stuck in local minima much more often than the embodied.

The results for 4-parity are presented in Fig. 1, where the mean generalization error is plotted against f, the fraction of the 2^4 problems that were withheld from the training set. Error of 1 corresponds to chance level. (There is a good but uninteresting explanation for why the disembodied networks actually perform worse than chance for large values of f.) The embodied systems generalize almost perfectly up to $f = 0.75$. As for the disembodied networks, with the marginal exception of the 1-2*-2-1 architecture (which is almost pre-engineered to become a parity-calculating flip-flop), they generalize very poorly (as do Clark and Thornton's models): omitting just two problems gives very high error, and

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 315-320

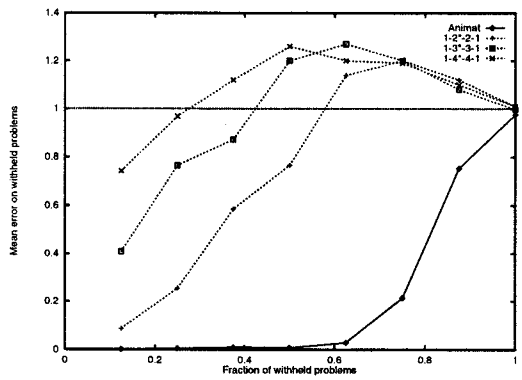

Figure 1: Generalization error (chance=1) for embodied systems and disembodied controls

at four problems they are close to chance level. Even the 1-2*-2-1 architecture has errors that are 50-100 times greater than those of the embodied systems for f below 0.75. The problem length can be increased without substantially changing the results: keeping f fixed I got similar generalization performance for 5-, 6-, and 7-parity.

A final word concerning technique. The crucial aspect of this model is the sensorimotor embodiment, not the details of the two (input-motor, sensory-output) controllers. Although I used as controllers neural networks evolved by genetic algorithms—mainly because I am familiar with these techniques—other general purpose learning systems would probably do just as well, unless they are especially maladapted to the task. The point is that given a proper embodiment, the task of the controllers becomes very simple.

4. External representation

The interesting question is how the embodied systems managed to generalize so well. As I have already discussed, these systems had no *internal* means to represent the problem, therefore they had to perform all "computations" externally, i.e., by means of their movement. All the successfully generalizing systems adopted variations on the following strategy in the input-motor network: do not move forward, do nothing on 0, turn by 180° on 1. To calculate parity, the sensory-output network just has to give 0 if the animat is facing north (the landmark is to its left), and 1 if it is facing south (landmark to the right).

The representation of the parity problem spontaneously evolved by the embodied systems is closely akin to "epistemic action" recently investigated by Kirsh in the context of human problem solving (Kirsh 1995, Clark and Chalmers 1996). The idea is that for reasons having to do with human cognitive limitations, people choose to offload certain mental operations onto the external world; i.e., instead of performing an operation "in the head" one performs a much sim-

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 315-320

pler physical action, lets the world evolve, and then reads off the answer. For instance, instead of mentally rotating the falling pieces in the computer game Tetris in order to see where they would best fit, people offload the rotation by physically rotating the pieces on the screen. On-line planning is another example where the perception-action cycles spares the system difficult internal calculations. In these examples and in many others, the cognitive operation is spread beyond the confines of the cognitive system into its external world. This kind of offloading, I suggest, may play an important role in grounding and channeling inductive learning. Even when overt physical action is absent (the usual case in human reasoning), such action and its benefits may be present in the covert, internalized form of mental imagery (visual and kinesthetic) and its transformations, often reported in reasoning (and, interestingly enough, by humans learning parity from examples).

The conclusion we can draw is as follows. The dynamics hill-climbing does not lead neural networks to generalize "correctly" (i.e., in a way that makes sense to us) to previously unseen cases of difficult problems such as parity—indeed, why should it? If we trade this internal dynamics for an external, sensorimotor dynamics of a simple embodied creature, the generalization becomes correct. This suggests that induction is less a matter of formal selection criteria on hypotheses,[1] but rather a procedure grounded in one's sensorimotor relations with the external world. Indeed, we choose the hypotheses that we do, even in abstract contexts, partly because they have a certain *meaning* for us—for a human being parity is simply more meaningful than one of the "erroneous" hypotheses generated by the disembodied networks—and meaning is not present in formal, ungrounded, disembodied systems (Harnad 1990). The problem of induction is to discover what is it about certain hypotheses that makes them more meaningful than others in certain contexts. Perhaps a relation to sensorimotor schemas is part of the answer.

5. References

Clark, A. and Chalmers, D. 1996. The external mind. Manuscript, Dept. of Philosophy, Washington Univ.

Clark, A. and Thornton, C. 1997. Trading spaces: Computation, representation and the limits of uninformed learning. *Beh. Brain Sci.* 20(1).

Harnad, S. 1990. The symbol grounding problem. *Physica* D42:335–46.

Kirsh, D. 1992. From connectionist theory to practice. In M. Davis, ed., *Connectionism: Theory and practice.* New York: Oxford Univ. Press.

Kirsh, D. 1995. The intelligent use of space. *Art. Intell.* 73:31–68.

Sejnowski, T. and Quartz, S. 1997. The constructivist manifesto. *Beh. Brain Sci.*, to appear.

Wexler, M. 1996. Embodied induction: Learning external representations. Paper given at AAAI 1996 Fall Workshop, Cambridge, Mass.

[1]Inductive learning in neural networks is no less formal than in classical AI; the formal computations simply occur on a lower and less explicit ("subsymbolic") level.