Preface

Learning to do simulations is like learning many skills. You learn best by doing. Reading about simulations and watching others do them can be useful. But when it comes down to it, there is no substitute for first-hand experience.

Connectionists are certainly not the first to develop models, nor the first to run simulations of them. But simulations play a particularly important role in connectionism. Why is this?

Connectionist models are often deceptively simple. After all, they are composed of simple neuron-like processing elements (usually called "nodes"). In fact, the nodes of most connectionist models are vastly simpler than real neurons. There appear to be no elaborate rules nor grammars (which has led some to argue—erroneously in our opinion—that connectionist models "have no rules"). The nodes are simply hooked together in rather straightforward arrangements, presented with some stimulus, and then allowed to interact freely. Periodically, some nodes may output values.

Put this way, one might wonder why go to the bother of writing a computer program to actually simulate a process whose outcome ought to be entirely predictable in advance. The surprise is that very often the outcome is *not* what one might have predicted. The dynamics of these simple systems turn out to be quite complex. For various reasons, including the fact that they are nonlinear (see Chapter 4 in the companion volume, *Rethinking Innateness*), it is frequently only possible to know how they will perform by running careful experiments and observing their behavior under controlled conditions. (In this respect, they are not unlike humans and other unpredictable biological systems.) Simulations are therefore crucial, and modelers' expectations are often betrayed.

The experimental aspect of running simulations is something we want to emphasize in this book. For many of our readers, particularly those who have been trained as developmentalists, the experimental methodology will be familiar. Rather than thinking of doing simulations as a very new and different way of doing research, we encourage our experimentalist readers to regard this as simply doing experiments with a different sort of subject pool!

There is also an important conceptual role which is played by simulations, and we wish to emphasize this as well. Although it may be satisfying to build a model which successfully learns some behavior, especially if that behavior is complex and not obviously learnable, we feel the real value of simulating models is that we can then ask, "How did the model solve the problem?" When we run experiments with human subjects, the most we can usually do is to construct theories which are consistent with our data. This is an inferential process. At best, we may have multiple sources of evidence which converge on the same theory. But rarely do we have the opportunity to look inside our subjects' heads and verify that the mechanism responsible for the behavior we have observed is indeed at work.

Simulated models, on the other hand, *can* be opened up for inspection and analysis. What we often find is that the mechanism which is actually responsible for the behavior we observed from the outside looks very different from what we guessed it might be. We must be careful, of course, not to assume that the model's solution is necessarily the same as the one discovered by species or children. The model can be a rich source of new hypotheses, however. Discovering new solutions to old problems is very liberating and exciting.

So the second thing we will emphasize in this handbook is understanding what makes the models tick. We will present several techniques for network analysis, and try to extract the basic principles which are responsible for the networks' behavior. Network analysis is not easy, and the tools we suggest here only scratch the surface of what one would like to do. The problem is not unlike the one we would encounter if we actually did get to open up our subjects' heads: It's pretty complicated in there. The difference, of course, is that our models are vastly more tolerant of being poked and probed and lesioned than are our human subjects. And they recover faster!

Do you need to know how to program in order to do simulations? No. Ultimately it's helpful, but a great deal can be done without actually having to do any programming, and no programming at all is required to use this handbook. The handbook is organized around a set of simulations which can be run using the software which is provided on the enclosed floppy disks. This software includes a neural network simulator called **tlearn**. Having a simulator allows you to experiment with different network architectures and training regimes without having to actually program the networks yourself (e.g., by writing software in C or Fortran). So if you do not know how to program, don't worry. The simulator simply needs to be told what the network's architecture is, what the training data are, and a few other parameters such as learning rate, etc. The simulator creates the network for you, does the training, and then reports the results. Other tools are included which allow you to examine the network's solutions. Thus, while you might find it eventually useful to be able to program yourself (for example, to create new stimuli), you will not need to do any programming to run the simulations described in this book. You will need to be able to use a simple text editor; whichever one you use is up to you.

The software itself exists in several versions, and will run on Macintoshes, PCs, Unix workstations, and most other machines which have ANSI standard C compilers. Precompiled binaries exist for several common platforms, but sources are provided in case you wish to modify and recompile the simulator yourself. Chapter 3 provides an overview of the simulator; installation procedures and reference manuals are supplied in Appendices A and B. When we designed our first few exercises, we kept in mind that most readers would need explicit help in using the simulator. These exercises should help ease the reader into the process of running a simulation. We do not guarantee that the software provided with this book is free of bugs. In fact, we guarantee that if you try hard enough you will find some situations where **tlearn** will break. Please tell us about any software problems you experience by emailing us at `innate@crl.ucsd.edu`. We may not respond to your queries immediately but we will fix the bugs as time permits. You can obtain the latest versions of **tlearn** via anonymous ftp at `crl.ucsd.edu` in the `pub` directory or on the world wide web at `http://crl.ucsd.edu/innate` or via anonymous ftp at `ftp.psych.ox.ac.uk` also in the `pub` directory.

A very major source of support—intellectual as well as monetary—for this project came from the MacArthur Foundation through their funding of the Neural Modeling Training Program for Developmentalists which was run at the Center for Research in Language at

UCSD from 1988 to 1993. We believed, and MacArthur was willing to gamble, that the connectionist paradigm has great potential for developmentalists. Through the training program we were able to bring to CRL more than a dozen developmentalists, junior researchers as well as senior people, for periods ranging from a week to several months. The training program worked in two directions. Trainees learned how to model; we got to pick their brains. It has been a stimulating and rewarding experience and we are enormously grateful to our trainees and to MacArthur for the willingness to embark on this venture with us. The far-sighted approach of Bob Emde, Mark Appelbaum, Kathryn Barnard, Marshall Haith, Jerry Kagan, Marion Radke-Yarrow, and Arnold Sameroff was critical in this effort. Without their support—both tangible as well as intellectual—we could not have written this book.

We also thank our trainees: Dick Aslin, Alain Content, Judith Goodman, Marshall Haith, Roy Higginson, Claes von Hofsten, Jean Mandler, Michael Maratsos, Bruce Pennington, Elena Pizzuto, Rob Roberts, Jim Russell, Richard Schwartz, Joan Stiles, David Swinney, and Richard Wagner. Neither this nor the companion volume (*Rethinking Innateness*) could have been written without their excitement, inspiration, and practical advice. And we are very grateful to those who served as trainers in this program: Arshavir Blackwell, Mary Hare, Cathy Harris, and Virginia Marchman.

In addition, the material contained in the book has been used by participants in several of the Oxford Connectionist Modeling Summer Schools sponsored by the Oxford McDonnell-Pew Centre for Cognitive Neuroscience. We should like to thank the Summer School participants for their feedback on earlier versions of the exercises and the McDonnell-Pew Foundation for their generous support of this annual meeting. The demonstrators on the summer school—Todd Bailey, Neil Forrester, Patrick Juola, Denis Mareschal, Ramin Nakisa, Graham Schafer and Michael Thomas—helped us spot many errors in earlier drafts and helped us to realize that what seemed obvious was often a mystery.

Last, and certainly not least, we should like to express our gratitude to John Kendall and Steven Young for their heroic efforts in responding to our requests to continually update the **`tlearn`** interface, add new utilities and then change everything back again. Without their patience, determination and fine programming skills, this book would simply not exist.

How to use this book

This book is designed as a companion volume to *Rethinking Innateness: A connectionist perspective on development.* That book is our attempt to sketch out what we hope may serve as a framework for a theory of development. The book necessarily omits some details, particularly regarding finer details of simulations. One purpose of this book is to provide a way for the interested reader to pursue in depth some of the issues which we raise in the first volume.

Our larger ambition is that we will have persuaded some readers that this is a useful methodology which can serve them in their own research. We hope that this volume will therefore provide those readers with tools which they can use on their own. There are other simulators, of course, and architectures which may require more special-purpose software. This simulator, however, has served us well and is used by us not only for pedagogical purposes, but in our own research. We hope it will serve you as well!

KP & JLE

Oxford
La Jolla
1997