Introduction: Modelling perception with artificial neural networks

Colin R. Tosh and Graeme D. Ruxton

This book represents a substantial update of a theme issue of the *Philosophical Transactions of the Royal Society B Journal*, 'The use of artificial neural networks to study perception in animals' (*Phil Trans R Soc B* 2007 March 29; 362(1479)). Most of the 14 papers in that theme issue have been significantly updated and we include a further five entirely new chapters, reflecting emerging themes in neural network research. Our reasons for undertaking the theme issue and this book were not entirely altruistic. Having a young but growing interest in the use of artificial neural networks, we hoped that the publications would be an excuse for us to learn about areas in neural network research that seemed interesting to us and of potential application to our research. The people who will get most from the book are, therefore, ecologists and evolutionary biologists, perhaps with a notion of using neural network models of perception, but with little experience of their use. That said, the content of this book is extremely broad and we are confident that there is something in it for any scientist with an interest in animal (including human) perception and behaviour.

We organise the book into four fairly loose categories. The chapters by Kevin Gurney and Steve Phelps are broad reviews and introduce the two main themes of the book: neural networks as tools to explore the nature of perceptual processes, and neural networks as models of perception in ecology and evolutionary biology. Kevin Gurney's chapter is an excellent general introduction to the theory and use of neural networks and tackles the question: what can simple neural network models tell us about real neural circuits and the brain? Steve Phelps's chapter is a 'where it's at and where it's going' of artificial neural network models used to explore perceptual allocation and bias, and the models and ideas in it can be applied to many other areas of ecology and evolutionary biology. Like most of the articles in the book, both of these chapters can be appreciated by those with little or no mathematical expertise.

The next six chapters are research or focused review articles on neural network models and their use in elucidating the nature of perceptual processes in animals. Axel Borst's chapter describes and compares the properties of different neural models of motion

Modelling Perception with Artificial Neural Networks, ed. C. R. Tosh and G. D. Ruxton. Published by Cambridge University Press. © Cambridge University Press 2010.

detection: specifically Reichardt and gradient detectors. We (the editors) are excited about the potential of applying such models to issues in predator–prey interactions, to address how predator-targeting accuracy is affected by the speed and number of moving prey items. Robert White and Larry Snyder use a recurrent neural network model to investigate how accurate internal representations of visual space are formed in primates. Francesco Manella *et al.* use a novel computational model which is strongly rooted in the anatomy and physiology of the mammalian brain to investigate the role of the amygdala in the phenomenon of devaluation in an instrumental conditioning task. Raffaele Calabretta explores the concept of 'genetic interference': a phenomenon that can reduce the evolvability of both modular and nonmodular visual neural networks but can be alleviated by 'sexual reproduction' in neural networks. The last two chapters of this section represent a distinct sub-theme: the relationship between connective architecture of neural networks and their functioning. Hiraku Oshima and Takashi Odagaki investigate the influence of regular, small world, random network structures on the storage capacity and retrieval time of Hopfield networks. Linda Douw *et al.* consider whether the neural and behavioural consequences of brain tumours are due to disruption of the small world properties of whole brain networks. The issue of the relationship between network structure and functioning in a burgeoning theme in wider network theory (e.g. social and communication networks) should be of interest to anyone interested in how animal behaviour evolves in response to the environment.

The next five chapters are by ecologists and evolutionary biologists and apply neural networks to classic questions in these disciplines. Karin Pfennig and Michael Ryan apply Elman networks to study the evolution of character displacement and mate choice using the calls of tundra frogs as network input. Sami Merilaita reviews recent work on the anti-predator benefits of prey colouration that uses simple neural network models. Noél Holmgren *et al.* review recent work on the use of neural networks to study ecological specialisation and sympatric speciation: an interesting approach that offers a potentially powerful alternative to traditional mathematical simulation models in these areas. All of these papers additionally use or discuss genetic algorithms, an optimisation framework also applicable to models other than neural networks that tune model parameters through a selective process analogous to natural selection. This powerful 'organic' selection method can be applied to a variety of systems. David Krakauer *et al.* use analytical mathematics with simple feedforward neural networks to show that multimodal signals (animal signals that exploit multiple sensory organs) can increase the robustness of signals through multiple channels (e.g. frequencies in vocalisation). Finally, Richard Peters investigates the difficulties involved in signal recognition by a species of lizard using a saliency map and a winner-take-all neural network of leaky integrate-and-fire neurons. This model is based on some of the known properties of visual processing in primates and will appeal to ecologists who want to explore what the most salient object is in a particular visual scene, but are discouraged by the abstraction of simple connectionist approaches.

The next five chapters are generally on methodological issues in the use simple feedforward networks. Chapter 14 by Stephano Ghirlanda and Magnus Enquist and

Chapter 15 by Colin Tosh and Graeme Ruxton (the editors) are on the phenomenon coined 'path dependence' by the former authors. This is the tendency of certain neural networks with commonly used architectures and training methods to vary in predictive properties, depending on the order of presentation of training inputs, or with stochastic variation in the starting properties of networks. This effect could have important biological as well as methodological implications. In their chapter, Dan Franks and Graeme Ruxton argue that training methods such as back propagation that researchers have used with feedforward nets to study learning in animals are inappropriate as normally applied because learning is too slow. They offer a modified protocol for the application of training procedures that better replicates the tempo of learning in real animals. Dara Curran and Colm O'Riordan review methods used to effect adaptive evolution in both the weights and architecture of artificial neural networks. We also place the chapter by Julian Olden in this section. This chapter, as well as being an interesting research paper on the relationship between landscape properties and animal movements, applies methods that allow one to dissect the functioning of neural networks. These methods should help to dispel the common myth that neural networks are 'black boxes' that produce interesting results but whose functioning and action cannot be analysed. Finally, the chapter by Roddy Williamson and Abdul Chrachri does not fit into any of the aforementioned categories and describes a real neural network: the cephalopod vestibular system. This chapter emphasises the fact that real neural networks are considerably more complex than most of the simple artificial ones described in the book, and in some (perhaps many) neural systems this complexity must be embraced in order to fully understand the system.

One of our loftier objectives in putting together this book was to attract readers from a broad and disparate range of disciplines and so foster cross-fertilisation of ideas. Papers in the book should interest readers from psychology, neurobiology, mathematics, ethology, ecology and evolutionary biology. It is hoped that readers from each of these disciplines might find something from another discipline that interests them and gives them new ideas for their own research. For example, many psychologists and neurobiologists could undoubtedly benefit from an increased appreciation of the evolutionary context of their study system, while many ecologists and evolutionary biologists could benefit from a greater appreciation of the neural mechanisms underlying phenomena at the level of the whole organism. We also hope that greater use of artificial neural networks might reduce the need for invasive animal experimentation. The study of nervous systems, using artificial models or otherwise, will always be founded on experiments with real nervous systems, but models can reduce the need for experimentation at particular stages of a research programme. A reliable model can simulate multiple scenarios and inform researchers on which areas of endeavour are likely to be most rewarding, thereby reducing the need for experimentation in areas that could lead up 'blind alleys'.

We keep this introduction short and leave the job of covering broad scientific themes in the use of neural network models to study animal perception to the first two chapters of the book.

Cambridge University Press
978-0-521-76395-0 - Modelling Perception with Artificial Neural Networks
Colin R. Tosh and Graeme D. Ruxton
Excerpt
More information

Part I

General themes

1

Neural networks for perceptual processing: from simulation tools to theories

Kevin Gurney

1.1 Introduction

This paper has two main aims. First, to give an introduction to some of the construction techniques – the ‘nuts-and-bolts’ as it were – of neural networks deployed by the authors in this book. Our intention is to emphasise conceptual principles and their associated terminology, and to do this wherever possible without recourse to detailed mathematical descriptions. However, the term ‘neural network’ has taken on a multitude of meanings over the last couple of decades, depending on its methodological and scientific context. A second aim, therefore, given that the application of the techniques described in this book may appear rather diverse, is to supply some meta-theoretical landmarks to help understand the significance of the ensuing results.

In general terms, neural networks are tools for building models of systems that are characterised by data sets which are often (but not always) derived by sampling a system input-output behaviour. While a neural network model is of some utility if it mimics the behaviour of the target system, it is far more useful if key mechanisms underlying the model functionality can be unearthed, and identified with those of the underlying system. That is, the modeller can ‘break into’ the model, viewed initially as an input-output ‘black box’, and find internal representations, variable relationships, and structures which may correspond with the underlying target system. This target system may be entirely non-biological (e.g. stock market prices), or be of biological origin, but have nothing to do with brains (e.g. ecologically driven patterns of population dynamics). In these instances, we can ask whether the internal network machinations are informative of specific relationships between system inputs and outputs, and any internal variables. However, the mechanistic elements of a network have names which are evocative of processing in the animal brain; there is talk of ‘artificial neurons’, their interconnection strengths and ‘learning’. If, therefore, a neural network is a model of part of the brain, the problem of interpretation of internal mechanisms is particularly acute. For, if these mechanisms are based on those in the brain, is it the case that they reflect genuine, biological neural mechanisms? These and related questions are explored in the second half of the chapter.

Modelling Perception with Artificial Neural Networks, ed. C. R. Tosh and G. D. Ruxton. Published by Cambridge University Press.

1.2 Neural network principles

This section gives a high-level view of some of the principles and techniques used in this book. A more comprehensive treatment at this level can be found in Gurney (1997) while the books by Haykin (1999) and Bishop (1996) take a more mathematical approach.

We start with a pragmatic, working definition of a neural network: A neural network is an interconnected assembly of simple processing elements, *units* or *nodes* whose functionality is loosely based on the animal neuron. The processing ability of the network is stored in the inter-unit connection strengths, or *weights*, obtained by a process of adaptation to, or *learning* from, a set of *training patterns*. The rest of this section is devoted to unwrapping these terms with special emphasis on those networks that appear in subsequent chapters in this book.

1.2.1 Artificial neurons

Figure 1.1 is a graphical description of a typical neural network node.

Input signals $x_1, x_2, \ldots x_n$ are combined to form an output y via an *activation* variable a. The latter is formed by taking a weighted sum of inputs. That is,

$$a = \sum_i w_i x_i \tag{1}$$

The weights w_i may be positive or negative. The activation is then usually transformed by some kind of squashing function which limits the output y to a specified range (usually the interval [0,1]) and introduces a nonlinearity; this latter feature proves to be crucial in endowing neural nets with their powerful functionality (see next section). In the figure, the squashing function has been chosen to be the logistic sigmoid

$$y = \frac{1}{1 + \exp(-(a - \theta))} \tag{2}$$

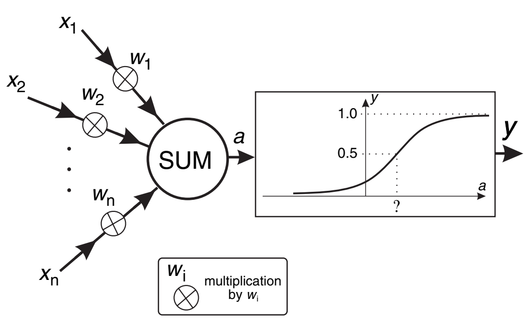


Figure 1.1. Simple model neuron.

although other, similar functions are occasionally used. The constant θ defines the point at which y takes its mid-point value. Moreover, it is the point where the function is changing most rapidly and is therefore the value of the activation at which the node is most sensitive to small changes in the inputs. The negative of θ is therefore sometimes referred to as the *bias*. Notice that y approaches 0 and 1 asymptotically as the activation decreases and increases respectively (so, y is never equal to 0 or 1, but may be made as close to these as we please).

The basic node described above has a long lineage. The first artificial neural node was the Threshold Logic Unit (TLU) introduced by McCulloch & Pitts (1943). This was also a two-stage device with the first stage given by (1) but with the output nonlinearity defined by a discontinuous step function, rather than the smooth ramp described by (2). Thus, the output of the TLU had only two values, 0 or 1, depending on whether the activation was less than or greater than the threshold θ, respectively. A more complex node – the *Perceptron* – was introduced by Rosenblatt (1958) which retained the Boolean (0,1) output of the TLU, but allowed pre-processing of Boolean input variables with arbitrary functions (so-called 'association units') whose outputs then formed the variables x_i in (1). The TLU is therefore a special case of the Perceptron when 'association units' just pass a single input through to each weight.

The neurobiological inspiration for the structure of Figure 1.1 is as follows. The input x_i corresponds to the presynaptic input on afferent i, while the weight w_i encapsulates the corresponding synaptic strength. The product $w_i x_i$ is akin to the post-synaptic potential (PSP) which is inhibitory/excitatory according to whether w_i is negative/positive. The integration of PSPs over the dendritic arbour and soma is represented by simple arithmetic addition, and the quantity a corresponds to the somatic membrane potential. This is then transformed by the squashing function to give a firing rate y. Clearly some of these correspondences are, at best, merely qualitative analogues. The issue of realism is revisited in the second half of the chapter.

1.2.2 Feedforward networks and classification

A ubiquitous problem in perception is that of classification or pattern recognition. As an example, consider the problem of identifying letters of the alphabet. Humans are able to recognise letters in many sizes, orientations and fonts (including handwritten variations) with ease. Any individual person can never see all possible letter variants, but, instead, will learn idealised letter shapes from a very small set of possibilities (usually a plain font in children's reading books). This latter point demonstrates that *generalisation* is a key component in the classification process. That is, the ability to generalise knowledge of specific pattern exemplars to a wide variety of related cases.

Based on this example we now formalise the general problem of classification as follows. Given an arbitrary sensory input pattern drawn from some universal set of patterns, is it possible to place that pattern in its appropriate class or category, where there are generally many fewer classes than the patterns themselves? Further, we suppose that we do not have an exhaustive list of the entire universe of patterns; rather, we only have immediate access to some subset of patterns P, and knowledge of the category that each member of P belongs to. By way of terminology P is the *training set* referred to in the motivational definition of a neural network at the start of this section. The problem is to construct an input-output model, based on this limited knowledge, which will generalise so that, if it is presented with a pattern not in P, it will elicit the correct classification for that pattern. Notice that a 'model' which simply classifies P but does not generalise is easy to construct but of no real interest – it is just a lookup table of pattern-class pairings. We will return to the relationship between neural processing and generalisation later. In the meantime we will look at how the classification problem may be solved in principle by a neural network.

Figure 1.2 shows a *feedforward network* which consists of a layered structure with information flowing from the inputs, at the bottom of the diagram, to the outputs at the top.

The inputs have no functionality as such, but are simply points which receive pattern information and distribute this information to the first layer of neural nodes per se (of the type described above). In the example, there are four inputs, and so all patterns for classification would have to be defined by a list of four numbers. In more formal analyses, these lists of numbers are properly referred to as *vectors* with numeric *components*, and we sometimes speak of *pattern vectors*. This first layer of functional nodes is sometimes referred to as a *hidden layer* since we are not supposed to inspect or control the output values on these nodes (y in Eq. 2) during the process of setting the network weights; that is during *training* or *learning*. The outputs of the hidden layer are subsequently processed by an *output layer* which is used to read out the category in which the input pattern is placed. There are several ways of doing this depending on the way information is represented in the network. We will refer to a network of the kind shown in Figure 1.2 as a two-layer network since it contains two layers of processing nodes. Some authors include the input layer in the layer count so that the network in Figure 1.2 would constitute a 3-layer net. The final point to make here is that networks of the kind depicted

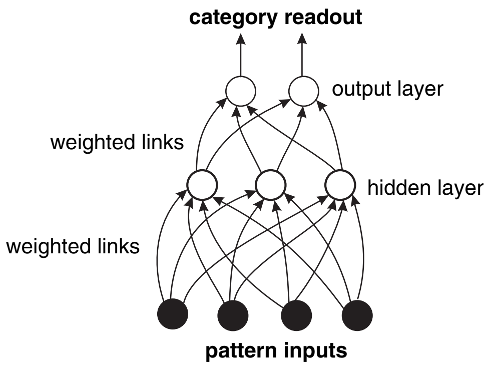


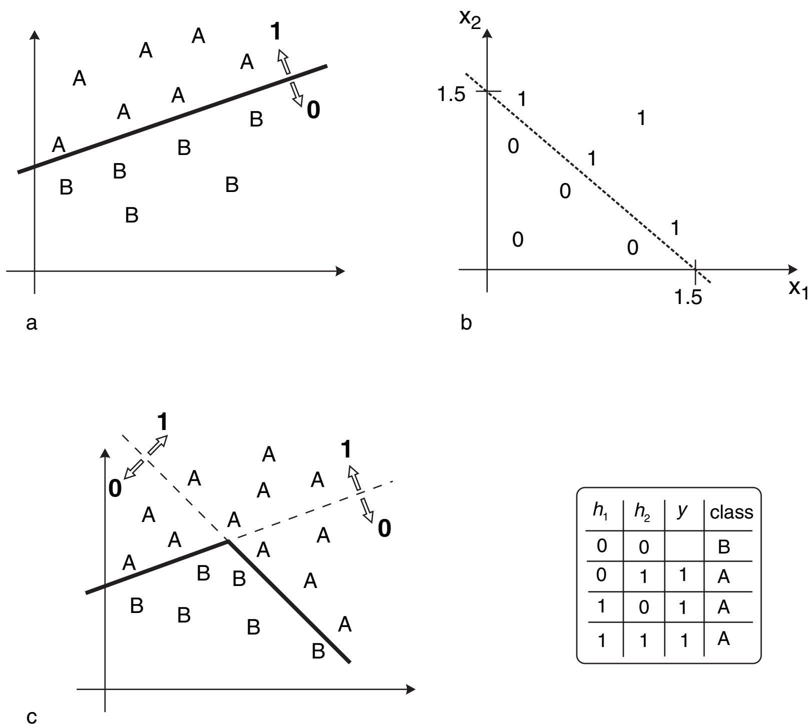
Figure 1.2. Simple two-layer feedforward neural network.

in Figure 1.2 are sometimes called *multi-layer perceptrons* or *MLPs*, in deference to the important role played by Rosenblatt's original perceptron in shaping the theory of neural network learning (Minsky & Papert, 1969).

Notice that processing by any particular node can be performed independently of that in any other. Thus, processing could, in principle, be performed in parallel if we had the necessary hardware resources to assign to each node. In spite of this, most networks find their implementation in software simulation in a conventional computer in which each node has to be visited serially to compute its output.

There is a mathematical framework which is particularly useful for describing quantitatively the process of classification in networks. It is based on the notion that patterns reside in some *pattern space* and is evocative of geometric analogies that enable the problem to be visualised. Suppose, for example, we have patterns belonging to two classes, *A* and *B*. If each pattern was defined by only two numerical components, then it could be represented quantitatively as a point in Cartesian axes as shown in Figure 1.3a.

If, in fact, each pattern is a vector with $n > 2$ components, Figure 1.3 is just a cartoon schematic which is simply illustrative of the case in *n*-dimensions. In Figure 1.3a, the patterns are shown as being separated by a straight line. In 3-D this situation implies a



h_1	h_2	y	class
0	0		B
0	1	1	A
1	0	1	A
1	1	1	A

Figure 1.3. Geometric view of pattern classification.

Cambridge University Press
978-0-521-76395-0 - Modelling Perception with Artificial Neural Networks
Colin R. Tosh and Graeme D. Ruxton
Excerpt
More information

plane, and in n-D ($n > 3$) a hyperplane. In all these cases we say that the patterns are *linearly separable*, and the straight line is schematically indicative of this.

Suppose we have a single artificial neuron with n-inputs, then it could attempt to solve the classification problem in Figure 1.3a by indicating output values of 1,0 for classes A,B respectively. This could occur exactly if the node was a TLU, and approximately using a node of the form shown in Figure 1.1 (since, in this case, the output approaches 0 and 1 asymptotically). It may be shown that linearly separable problems can indeed be solved by a single artificial neuron; a result which follows from the linearity of signal combination in Eq. 1. To see this, consider, the two-input case and assume, for simplicity, a TLU. The critical condition that defines classification occurs when the activation a equals the threshold θ, since small changes in a around this value cause the node to switch its output between 0 and 1. Putting $a = \theta$, gives $w_1x_1 + w_2x_2 = \theta$. This may be solved for x_2 in terms of x_1 to give

$$x_2 = -\left(\frac{w_1}{w_2}\right)x_1 + \left(\frac{\theta}{w_2}\right)$$

which is a straight line with slope $-w_1 / w_2$ and intercept θ / w_2. Now put, for example, $w_1 = w_2 = 1$, and $\theta = 1.5$. This defines a line $x_2 = -x_1 + 1.5$ as shown in Figure 1.3b. Here, pairs of values (x_1,x_2) defining points on the same side of the line as the origin give TLU outputs of 0, while values defining points on the other side of this line give TLU outputs of 1. In particular, the Boolean inputs (1,1) give an output of 1, while the other three Boolean input pairs give an output of 0 (in this case the TLU is acting as a classical logic AND gate).

Figure 1.3c shows a harder problem in pattern space which may only be solved by a *decision line* (in n-D, a *decision surface*) consisting of two straight, but non-colinear segments (shown by the solid line in the figure). The dotted lines show the extension of the line segments which make each of them a continuous straight line throughout pattern space (similar to the line in Figure 1.3a). Each extended straight line then defines a linearly separable problem which may be solved by nodes with outputs *h1* and *h2*. While each of these separate classifications mixes patterns A and B together, the table in the figure shows how the original classification problem may now be solved by taking suitable combinations of *h1* and *h2*; that is, class B is signalled if and only if both *h1* and *h2* are zero. This 2-component classification problem is linearly separable and may be solved with a single 2-input neural node. The original *A/B* classification problem has therefore been decomposed into two stages which may be solved by a two-layer net with two hidden nodes (yielding *h1* and *h2*) and a single output node.

As the classification becomes more complex, we may now ask the following question: is it possible to solve an arbitrary classification problem with a two-layer net – or do we need to resort to more complex structures? That is, in an analogous way to the example above, can we describe the decision surface of the problem in a piecewise linear way, solve the resulting decomposition using hidden units, and then combine their outputs in a