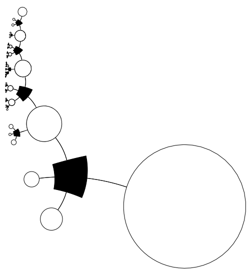
Part V

Neural networks



38

Introduction to Neural Networks

In the field of neural networks, we study the properties of networks of idealized 'neurons'.

Three motivations underlie work in this broad and interdisciplinary field.

Biology. The task of understanding how the brain works is one of the outstanding unsolved problems in science. Some neural network models are intended to shed light on the way in which computation and memory are performed by brains.

Engineering. Many researchers would like to create machines that can 'learn', perform 'pattern recognition' or 'discover patterns in data'.

Complex systems. A third motivation for being interested in neural networks is that they are complex adaptive systems whose properties are interesting in their own right.

I should emphasize several points at the outset.

- This book gives only a taste of this field. There are many interesting neural network models which we will not have time to touch on.
- The models that we discuss are not intended to be faithful models of biological systems. If they are at all relevant to biology, their relevance is on an abstract level.
- I will describe some neural network methods that are widely used in nonlinear data modelling, but I will not be able to give a full description of the state of the art. If you wish to solve real problems with neural networks, please read the relevant papers.

▶ 38.1 Memories

In the next few chapters we will meet several neural network models which come with simple learning algorithms which make them function as *memories*. Perhaps we should dwell for a moment on the conventional idea of memory in digital computation. A memory (a string of 5000 bits describing the name of a person and an image of their face, say) is stored in a digital computer at an *address*. To retrieve the memory you need to know the address. The address has nothing to do with the memory itself. Notice the properties that this scheme does *not* have:

1. Address-based memory is *not* associative. Imagine you know half of a memory, say someone's face, and you would like to recall the rest of the

memory – their name. If your memory is address-based then you can't get at a memory without knowing the address. [Computer scientists have devoted effort to wrapping traditional address-based memories inside cunning software to produce content-addressable memories, but content-addressability does not come naturally. It has to be added on.]

2. Address-based memory is *not* robust or fault-tolerant. If a one-bit mistake is made in specifying the *address* then a completely different memory will be retrieved. If one bit of a *memory* is flipped then whenever that memory is retrieved the error will be present. Of course, in all modern computers, error-correcting codes are used in the memory, so that small numbers of errors can be detected and corrected. But this error-tolerance is not an intrinsic property of the memory system. If minor damage occurs to certain hardware that implements memory retrieval, it is likely that all functionality will be catastrophically lost.

3. Address-based memory is not distributed. In a serial computer that is accessing a particular memory, only a tiny fraction of the devices participate in the memory recall: the CPU and the circuits that are storing the required byte. All the other millions of devices in the machine are sitting idle.

   Are there models of truly parallel computation, in which multiple devices participate in all computations? [Present-day parallel computers scarcely differ from serial computers from this point of view. Memory retrieval works in just the same way, and control of the computation process resides in CPUs. There are simply a few more CPUs. Most of the devices sit idle most of the time.]

Biological memory systems are completely different.

1. Biological memory is associative. Memory recall is *content-addressable*. Given a person's name, we can often recall their face; and *vice versa*. Memories are apparently recalled spontaneously, not just at the request of some CPU.

2. Biological memory recall is error-tolerant and robust.

   - Errors in the cues for memory recall can be corrected. An example asks you to recall 'An American politician who was very intelligent and whose politician father did not like broccoli'. Many people think of president Bush – even though one of the cues contains an error.
   - Hardware faults can also be tolerated. Our brains are noisy lumps of meat that are in a continual state of change, with cells being damaged by natural processes, alcohol, and boxing. While the cells in our brains and the proteins in our cells are continually changing, many of our memories persist unaffected.

3. Biological memory is parallel and distributed – not *completely* distributed throughout the whole brain: there does appear to be some functional specialization – but in the parts of the brain where memories are stored, it seems that many neurons participate in the storage of multiple memories.

These properties of biological memory systems motivate the study of 'artificial neural networks' – parallel distributed computational systems consisting of many interacting simple elements. The hope is that these model systems might give some hints as to how neural computation is achieved in real biological neural networks.

38.2 Terminology

Each time we describe a neural network algorithm we will typically specify three things. [If any of this terminology is hard to understand, it's probably best to dive straight into the next chapter.]

Architecture. The architecture specifies what variables are involved in the network and their topological relationships – for example, the variables involved in a neural net might be the *weights* of the connections between the neurons, along with the *activities* of the neurons.

Activity rule. Most neural network models have short time-scale dynamics: local rules define how the *activities* of the neurons change in response to each other. Typically the activity rule depends on the *weights* (the parameters) in the network.

Learning rule. The learning rule specifies the way in which the neural network's *weights* change with time. This learning is usually viewed as taking place on a longer time scale than the time scale of the dynamics under the activity rule. Usually the learning rule will depend on the *activities* of the neurons. It may also depend on the values of *target* values supplied by a *teacher* and on the current value of the weights.

Where do these rules come from? Often, activity rules and learning rules are invented by imaginative researchers. Alternatively, activity rules and learning rules may be *derived* from carefully chosen *objective functions*.

Neural network algorithms can be roughly divided into two classes.

Supervised neural networks are given data in the form of *inputs* and *targets*, the targets being a *teacher*'s specification of what the neural network's response to the input should be.

Unsupervised neural networks are given data in an undivided form – simply a set of examples $\{\mathbf{x}\}$. Some learning algorithms are intended simply to memorize these data in such a way that the examples can be recalled in the future. Other algorithms are intended to 'generalize', to discover 'patterns' in the data, or extract the underlying 'features' from them.

Some unsupervised algorithms are able to make predictions – for example, some algorithms can 'fill in' missing variables in an example $\mathbf{x}$ – and so can also be viewed as supervised networks.