



[Home](#)
[Archive catalogue](#)
[Bio of Turing](#)
[More about Turing](#)
[Codebreaking](#)
[Artificial Intelligence](#)
[Computer history](#)
[Photo gallery](#)
[Books on Turing](#)
[Cambridge archive](#)
[Links](#)
[Copyright](#)



Turing's Neural Networks of 1948

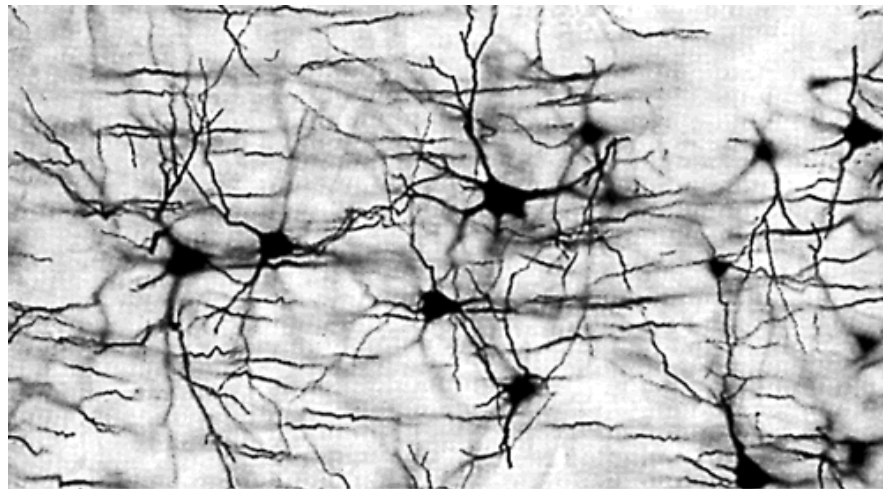
By Jack Copeland and Diane Proudfoot

© Copyright B.J. Copeland, D. Proudfoot September 2000

- Modern connectionism
- Turing's B-type neural networks
- Training a B-type network
- How Turing's model neurons work
- Two examples of B-type networks
- Making and breaking connections
- B-types and the brain

Modern Connectionism

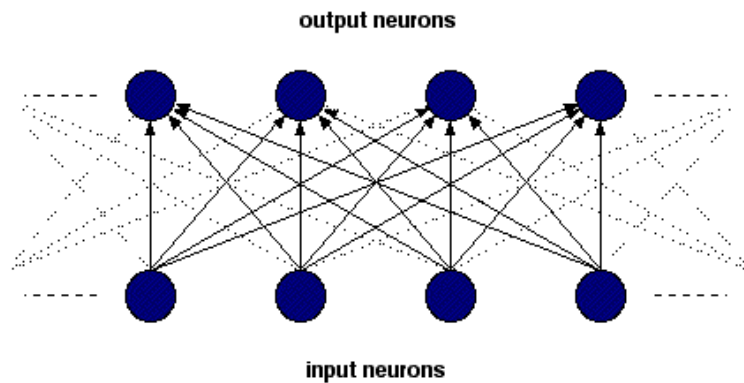
Connectionism is the emerging science of computing with networks of artificial neurons.



A natural neural network. The Golgi method of staining brain tissue renders the neurons and their interconnecting fibres visible in silhouette.

At the present stage of development of connectionism, reserachers simulate neurons and their interconnections using an ordinary digital computer (just as an engineer may use a computer to simulate an aircraft wing or a weather analyst to simulate a storm system). A training algorithm that runs on the computer adjusts the connections between the neurons, honing the network into a special-purpose machine dedicated to performing some particular task.

In a vivid demonstration of connectionism's potential, James McClelland and David Rumelhart have trained a network of 920 neurons to form the past tenses of English verbs.



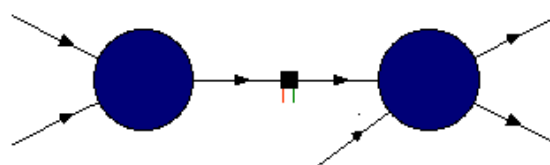
Each of the 460 neurons in the input layer is connected to each of the 460 neurons in the output layer

Root forms of verbs--such as come, look and sleep--were presented to a layer of input neurons (in a suitably encoded form). The supervisory system running on the computer observed the difference between the actual response of the layer of output neurons and the desired response--came, say--and mechanically adjusted the connections throughout the network in a way that gave the network a slight push in the direction of the correct response. About 400 different verbs were presented one by one to the network and the connections were adjusted after each presentation. If this whole procedure is repeated often enough using the same verbs--and it took almost 200 repetitions--the connections come to accommodate the differing needs of all the verbs in the training set. Once this stage is reached, the network will correctly form the past tense of unfamiliar verbs as well as of the original verbs. For example, when presented for the first time with *guard* it responded *guarded*, with *weep* *wept*, with *cling* *clung*, and with *drip* *dripped* (notice the double 'p'). (Sometimes, though, the peculiarities of English were too much for the network and it formed *squawked* from *squat*, *shipped* from *shape*, and *membled* from *mail*.)

Modern connectionists look back on [Frank Rosenblatt](#) as the founder of their approach. Rosenblatt published the first of many papers on connectionism in 1957. Few realize that Alan Turing wrote a blueprint for much of the connectionist project as early as 1948, in a little-known paper entitled '[Intelligent Machinery](#)'. Written while Turing was working for the National Physical Laboratory in London, the paper did not meet with his employers' approval. Sir Charles Darwin, the rather headmasterly director of the Laboratory, called it a 'schoolboy essay' and wrote to Turing complaining about its 'smudgy' appearance. In reality this far-sighted paper was the first manifesto of Artificial Intelligence, but sadly Turing never published it. In it he not only set out the fundamentals of connectionism but also brilliantly introduced many of the concepts that were later to become central in AI, in some cases after re-invention by others.

Turing's B-Type Neural Networks

Turing introduced a type of neural network that he called a 'B-type unorganised machine', consisting of artificial neurons, depicted below as circles, and connection-modifiers, depicted as boxes. A B-type machine may contain any number of neurons connected together in any pattern, but subject always to the restriction that each neuron-to-neuron connection passes through a connection-modifier. (What Turing called an 'A-type unorganised machine' is simply a B-type without the connection-modifiers. Without the ingenious connection-modifier, A-type machines cannot be trained.)



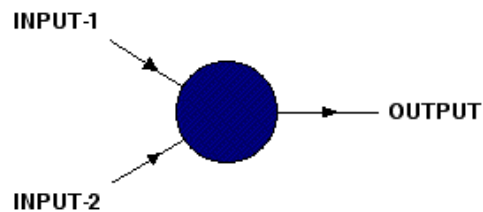
Two neurons from a B-type network. The red and green fibres on

the connection-modifier enable training by an external agent.

Training a B-Type Network

A connection-modifier has two training fibres (coloured green and red in the diagram). Applying a pulse to the green training fibre sets the box to pass its input--either 0 or 1--straight out again. This is pass mode. In pass mode, the box's output is identical to its input. The effect of a pulse on the red fibre is to place the modifier in interrupt mode. In this mode, the output of the box is always 1, no matter what its input. While it is in interrupt mode, the modifier destroys all information attempting to pass along the connection to which it is attached. Once set, a connection-modifier will maintain its function unless it receives a pulse on the other training fibre. The presence of these modifiers enables a B-type unorganised machine to be trained, by means of what Turing called 'appropriate interference, mimicking education'. Turing theorized that 'the cortex of an infant is an unorganised machine, which can be organised by suitable interfering training'.

How Turing's Model Neurons Work



Each neuron has two input fibres, and the output of a neuron is a simple logical function of its two inputs. Every neuron in the network performs the same logical operation, called 'nand'.

Nand is defined by the following table:

INPUT-1	INPUT-2	OUTPUT
1	1	0
1	0	1
0	1	1
0	0	1

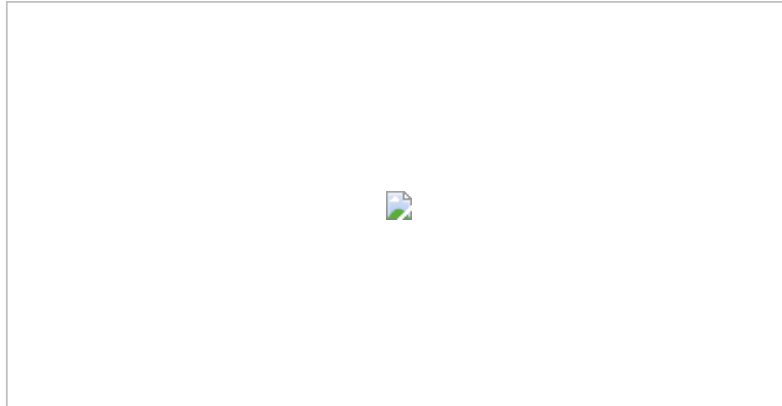
The output of a connection-modifier in interrupt mode is always 1. So if one of the neuron's input connections passes via a modifier in interrupt mode, the neuron's output is simply the opposite (or 'boolean negation') of whatever comes in along the second input fibre. For example, the first two lines of the table show what happens if INPUT-1 is connected to a modifier in interrupt mode. In this case the output from the neuron is the opposite of INPUT-2.

Turing chose nand as the basic operation of his model neurons because every other logical (or boolean) operation can be carried out by groups of nand-neurons. Turing showed that even the connection-modifier itself can be built out of nand-neurons. So each B-type network consists of nothing more than nand-neurons and their connecting fibres. This is about the simplest possible model of the cortex.

Turing wished to investigate more complex models of the cortex as well. He longed to do what modern connectionists are able to do: simulate a neural network and its training regimen using an ordinary digital computer. He would, he said, 'allow the whole system to run for an appreciable period, and then break in as a kind of "inspector of schools" and see what progress had been made'. But his own research on neural networks was carried out shortly before the first general-purpose electronic computers were up and running and he used only paper and pencil. Thereafter he turned his attention to related research in what is now called **Artificial Life**. It was not until 1954, the year of Turing's death, that **B.G. Farley and W.A. Clark** succeeded in running the first computer simulation of a

small neural network, at MIT.

Two Examples of B-Type Networks

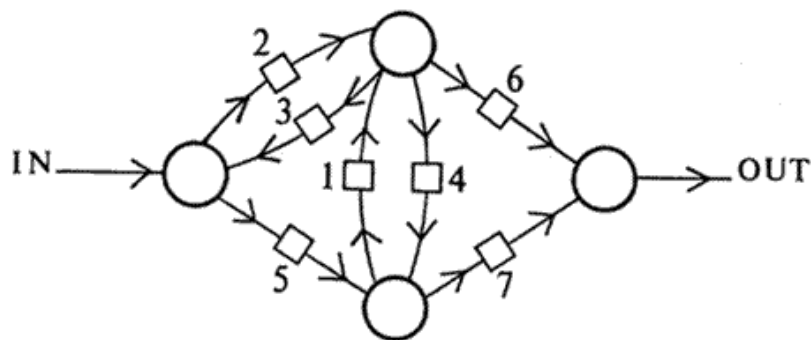


When both the connection-modifiers in this tiny B-type network are in pass mode, the network behaves as shown in the table on the left. That is to say, the network computes what logicians call the *inclusive disjunction* of the inputs. However, if the lower connection-modifier is switched to interrupt mode, the network behaves as shown in the table on the right. In this case, the output takes the same value as **A**, no matter what the value of **B**. If both modifiers are switched to interrupt mode, the network's output is always 0.

A	B	OUTPUT
1	1	1
1	0	1
0	1	1
0	0	0

A	B	OUTPUT
1	1	1
1	0	1
0	1	0
0	0	0

The second example of a small B-type network is Turing's own and is from page 11 of his 1948 paper '[Intelligent Machinery](#)'. Turing describes the example as 'chosen at random'. Can you work out how this network behaves? (You may like to refer to the discussion on pages 9-11 of '[On Alan Turing's Anticipation of Connectionism](#)' by Jack Copeland and Diane Proudfoot (from *Synthese*, vol. 108 (1996) pp. 361-377).)



An example of a larger B-type network is given later in this article.

Making and Breaking Connections

In 1958, Rosenblatt defined connectionism as the theory that 'stored information takes the form of new connections, or transmission channels in the nervous system, or the creation of conditions which are functionally equivalent to new connections'. The destruction of existing connections can be functionally equivalent to the creation of new connections. A network for performing a specific

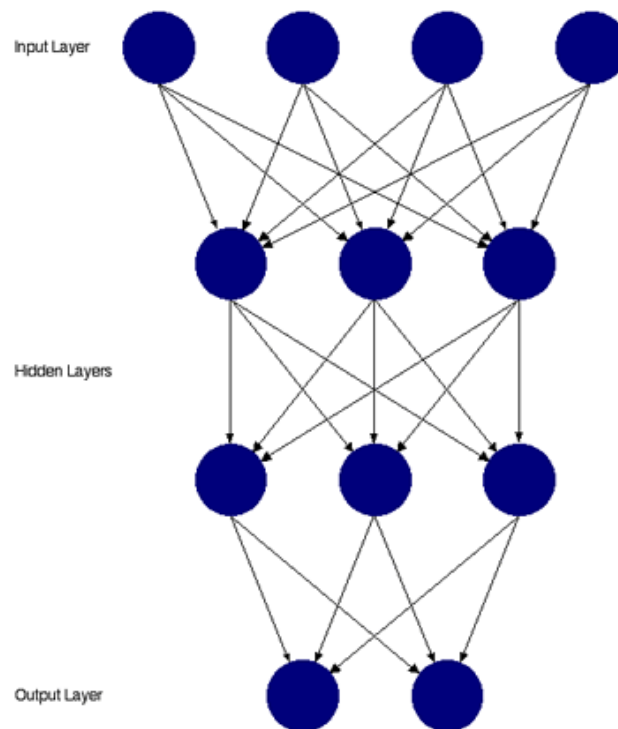
task may be produced by taking a network with more connections than are needed and selectively destroying some of them. Both processes, destruction and creation, are employed in the training of a B-type. In its initial state, the network that is to be trained has a large number of random inter-neural connections, and the modifiers on these connections are also set randomly, some in pass mode and some in interrupt mode. Unwanted connections are destroyed by switching their attached modifiers to interrupt mode. The output of the neuron immediately upstream of the modifier then no longer finds its way along the connection to the neuron on the downstream end. Conversely, changing the setting of the modifier on an initially interrupted connection to pass mode is in effect to create a new connection. This selective culling and enlivening of connections hones the initially random network into one organised for a given task.

Turing discovered that a large enough B-type neural network can be configured (via its connection-modifiers) in such a way that it itself becomes a general-purpose computer.

B-Types and the Brain

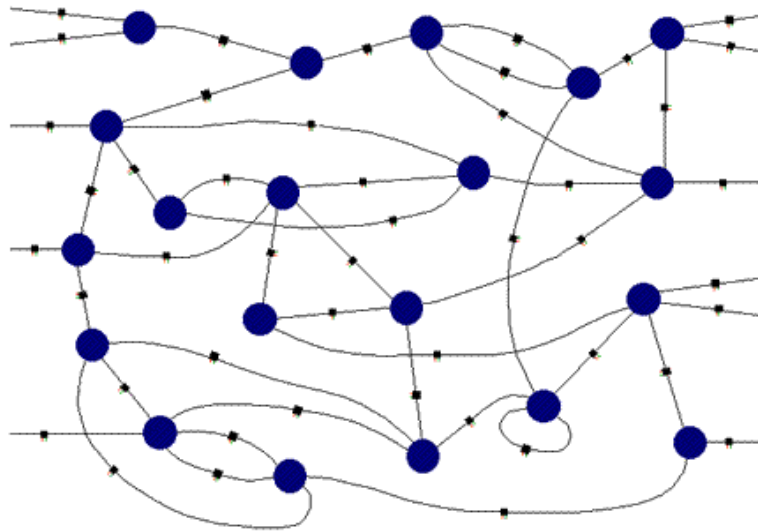
A large number of the output fibres of a neuron in the brain may be connected to the neuron's own input fibres, either directly or via some intervening chain of neurons. Neuroscientists have long stressed the importance and ubiquity of feedback within the brain. For example, the brain uses feedback to help us focus our attention on certain perceptions to the exclusion of others. Stefan Treue and John Maunsell have recently shown that when a monkey has its attention directed to one of several independently moving dots on a computer screen, feedback returns from neurons in the higher cortex to neurons in the lower cortical areas where motion is identified. This feedback serves to inhibit the activity of neurons that are firing in response to the motions of non-attended dots. However, despite its importance in the brain, feedback is seldom employed in modern connectionist networks. In contrast, the neurons in a B-type network interconnect very freely and, like a brain, a large network will typically be awash with feedback.

Below is a network of the sort typically studied by modern connectionists. Notice the regular, layered structure and the absence of any feedback. Information moves unidirectionally through the net from layer to layer.



A conventional neural network

In contrast, the neurons in a B-type neural network interconnect freely and a large B-type may be awash with feedback:



Part of a large initially random B-type network

Perhaps Turing's B-types contain lessons for modern connectionists.

Graphics by Robert Rozee

Google™ Custom Search

Search ×

Site created and maintained by Jack Copeland
Copyright © 1999 - 2015, Jack Copeland
All rights reserved