minds and machines (3)

neural networks
- another way to compute

"We [will] achieve the computational capacity of the human brain for about $1,000 by 2019 … about 100 billion neurons, times about 1,000 interneuronal connections per neuron, times 200 calculations per second per connection, or about 20 million billions calculations per second….

We are already well along the path of reverse engineering the brain to understand its principles of operation…by 2029, we will be able to combine the subtle powers of pattern recognition that the human brain excels in, with several attributes in which machine intelligence already exceeds human capabilities. These include speed, memory capacity, and the ability to instantly share knowledge.

Computers in 2029 possessing human levels of language understanding will be able to go out on the web and read and absorb all of the available literature and knowledge"

– Ray Kurzweil, text p.327ff.

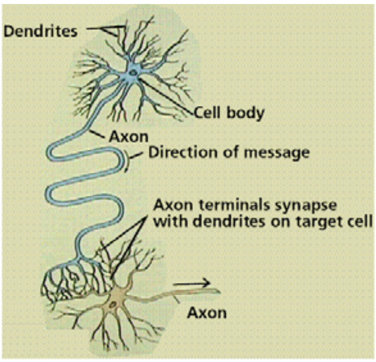
goals

- learn what artificial neural networks are
- understand why neural networks may provide information processing capabilities that current algorithms don't have

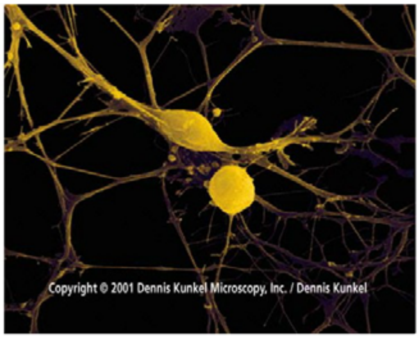
neural network models of the brain

- the model we will look at is an abstraction of nerve cells (neurons) in the brain
- it is very simplistic, but still useful to test theories of how brain works
  - rule out poor theories
  - hone in on important features

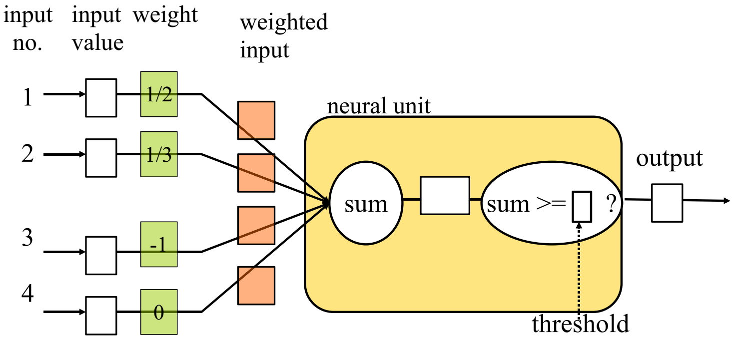
neuron



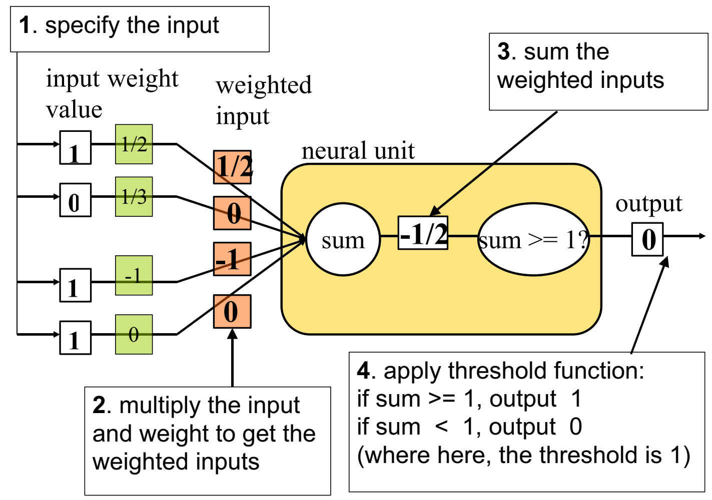
signals between neurons



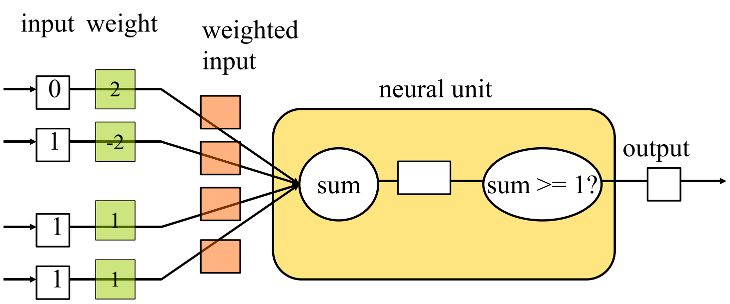
artificial neuron



computing the neuron output

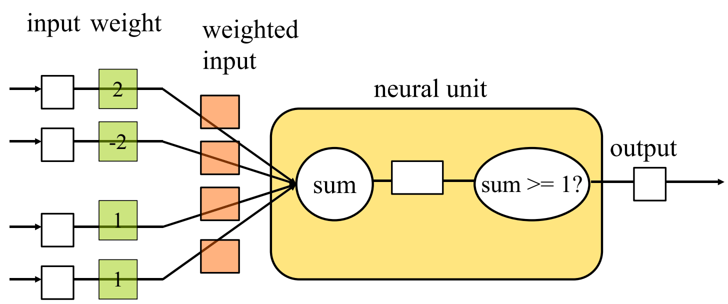


if input is 0111, what is output?



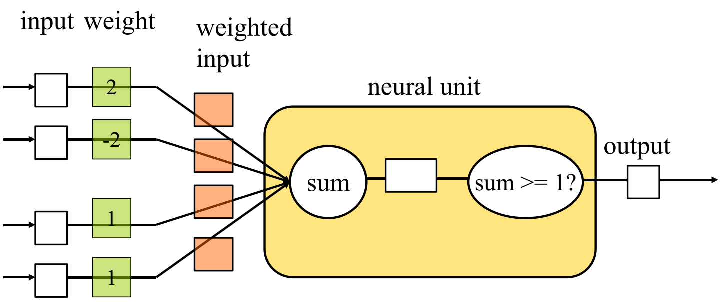
pattern recognition by neural units

- the set of inputs which cause the network to output "1" is the *pattern* it recognizes
- what patterns does this unit recognize?

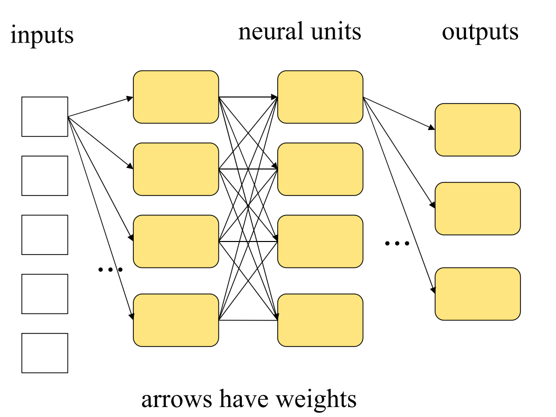


patterns recognised:

n	0000	n	0100	y	1000	n	1100
y	0001	n	0101	y	1001	y	1101
y	0010	n	0110	y	1010	y	1110
y	0011	n	0111	y	1011	y	1111



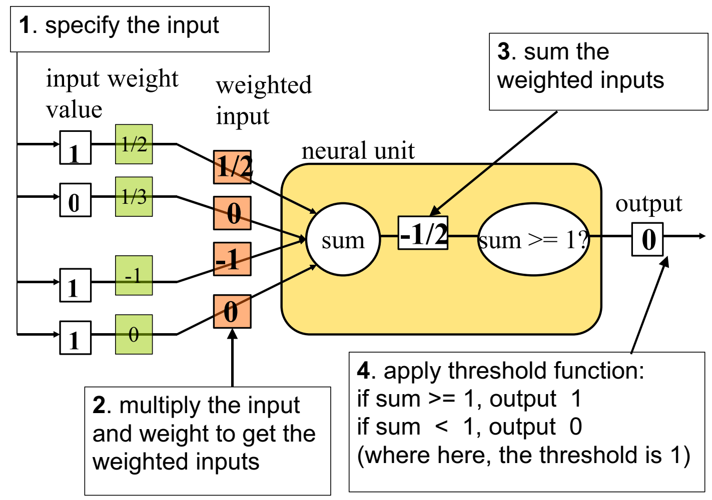
artificial neural network



pattern recognition by neural networks

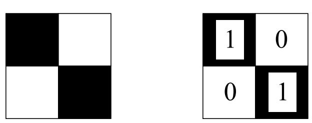
- just like neural units, neural networks, with one unit in the output layer, output “1” on some inputs and “0” on others
- the set of inputs which cause the network to output “1” is the pattern it recognizes

computing the neuron output



example: designing a neural network

- design a neural network to recognize a simple pattern: “checkered”



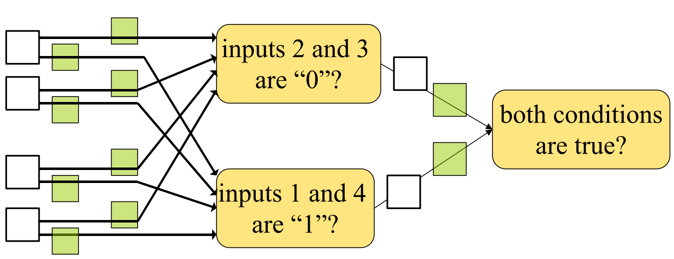
- algorithm input is 1001, representing the colour of each square, in the order: top left, top right, bottom left, bottom right
- input bits are numbered 1,2,3,4 from left to right
- algorithm output should be "1" on input 1001
- algorithm output should be "0" on all other inputs

pseudocode to recognise checkered pattern

```
if ((input 1 == 1)  and (input 2 == 0)  and
    (input 3 == 0)  and (input 4 == 1))
      {  output “yes”;  }
else
      {  output “no”;  }
```

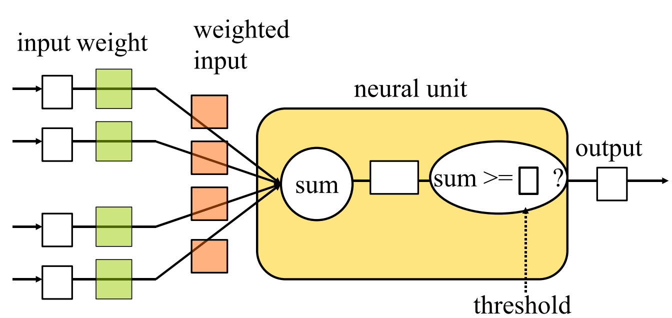
neural network to recognize checkered pattern

idea: use three neurons:

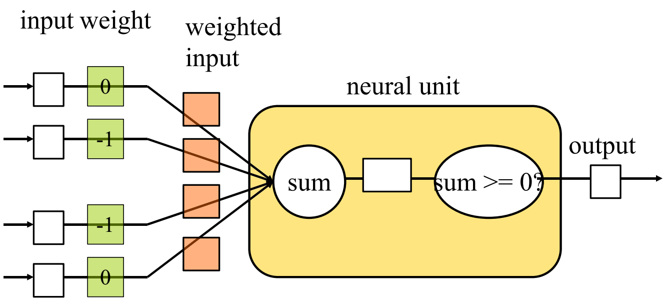


unit for "inputs 2 and 3 are 0?"

to design the neuron, we need to choose the weights and the threshold

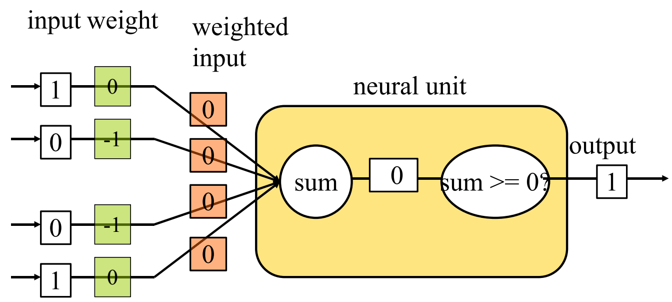


unit for "inputs 2 and 3 are 0?"

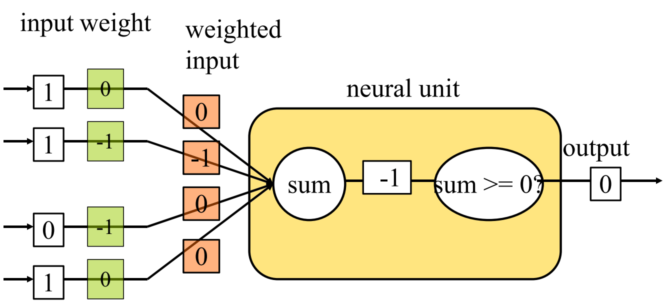


what happens on input 1001? 1101?

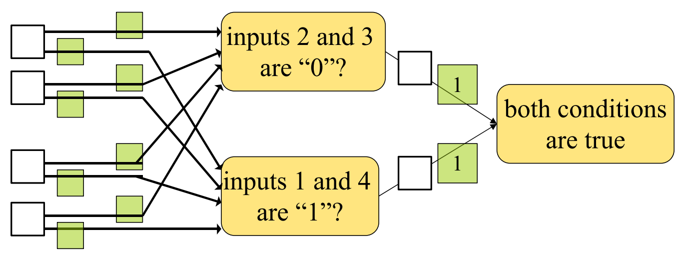
unit for "inputs 2 and 3 are 0?"



unit for "inputs 2 and 3 are 0?"

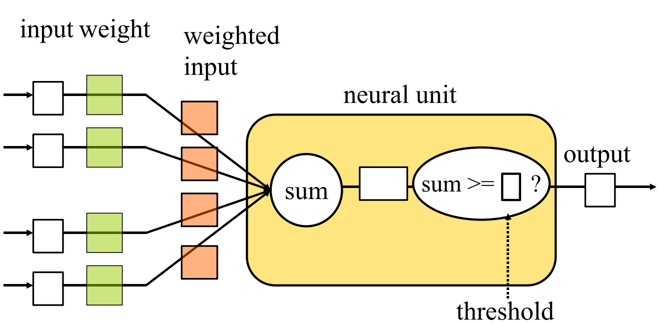


neural network to recognize checkered pattern

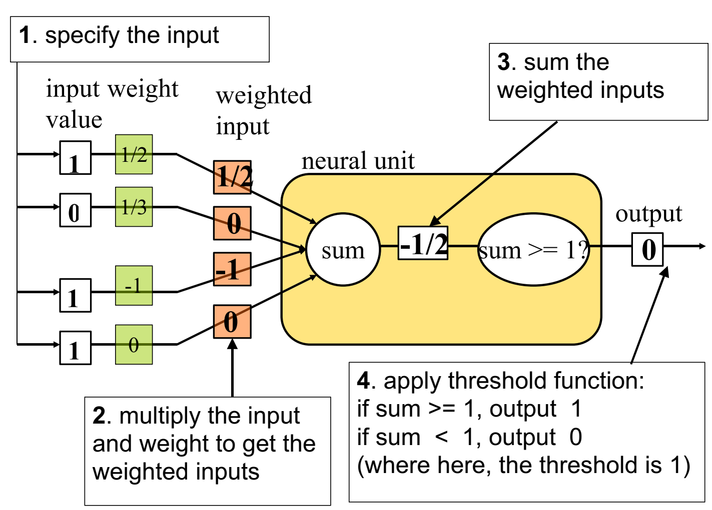


unit for "inputs 1 and 4 are 1"?

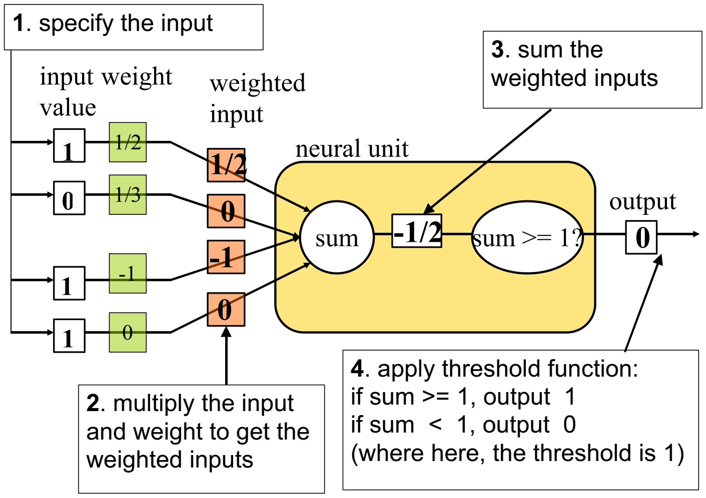
what weights and threshold would you use?



computing the neuron output

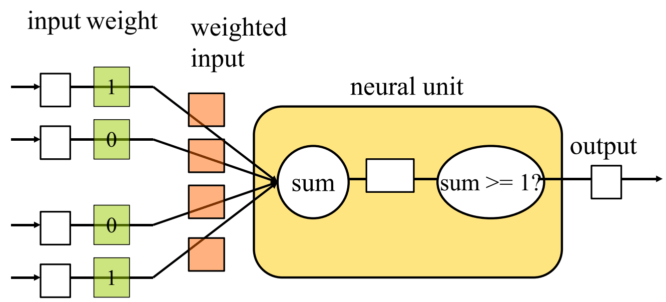


computing the neuron output



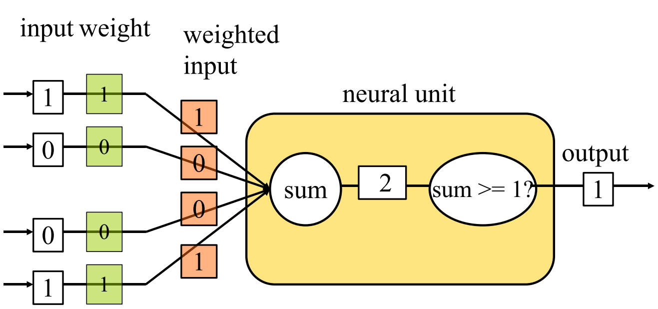
unit for “inputs 1 and 4 are 1?”

- let's try:
  - weights of 0 for the "irrelevant" inputs
  - weights of 1 for the "relevant" inputs 1 and 4
  - threshold of 1



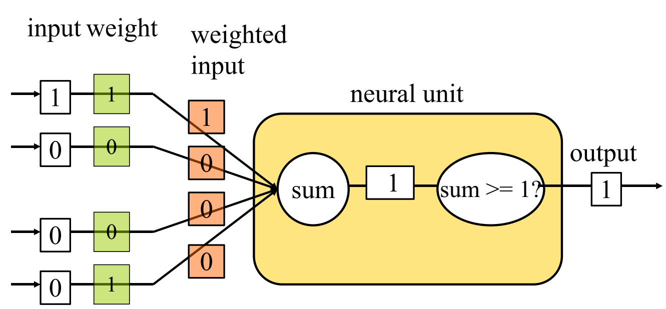
unit for “inputs 1 and 4 are 1?”

- unit does the right thing on input 1001!

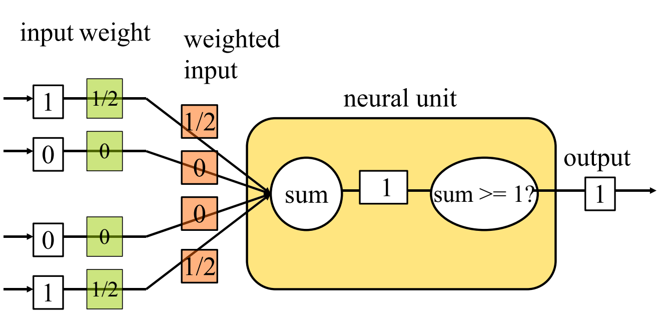


unit for “inputs 1 and 4 are 1?”

- … but the unit does the wrong thing on input 1000 :-(

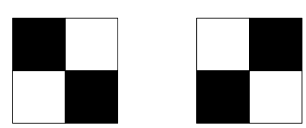


correct unit for “inputs 1 and 4 are 1?”



example to think about

- recognizing a more general checkerboard pattern:



- 1001 and 0110 are checkerboards; other inputs are not

where is the algorithm?

- *in a neural network, the algorithm is implicit in the weights and connections!*
- very different from a computer program, where the algorithm is a sequence of instructions

algorithm types

- the types of algorithms we have already seen are often referred to as *procedural* algorithms
  - generating geometric designs
  - binary search
  - calculating quiz score
- neural networks belong to the class of *connectionist* algorithms
- research question: what are relative merits of “connectionist” versus “procedural” algorithms?

learning

- in real-life, neural systems in the brain learn to perform computational tasks
- the algorithm (choice of weights) is not hard-wired into the brain
- how can learning occur?

a learning model for artificial neural networks

- learning occurs by changing the weights so that the influence of one neuron on another changes
- the network can be *trained,* using training examples, whereby weights are adjusted so that the network gives the correct output on the training examples
- in real neurons, learning can modulate the strength of the signal between neurons

applications of neural networks

- neural networks - simulated in traditional software - have been applied to solve problems in many application areas: weather prediction, cancer tissue diagnosis, speech recognition and more

summary: neural networks

- artificial neural networks offer many benefits in processing information:
  - *parallelism* (many neurons compute at once)
  - ability to *learn* how to solve a task from examples
  - ability to *adapt* over time
- neural networks, simulated by traditional software, are successfully used for some applications where training via examples is possible
- open question: in the future, will neural network machines be able accomplish tasks requiring intelligence?

summary: computer science and psychology

- psychologists and computer scientists work together to ...
  - design better interfaces to software
  - figure out how computers might perform complex tasks, e.g., recognising a scene, navigate unknown terrain
  - to understand how the brain processes information

what do you think?

- what potential do you see for more collaboration between psychologists and computer scientists?
- what are the pitfalls?