Simulating Language
10: From evolution to learning

Simon Kirby

simon@ling.ed.ac.uk

The story so far...

- Looked at the evolution of innate optimal signalling
- Treat communication system as a pair of matrices
  - Production matrix: rows are meanings, columns are signals
  - Reception matrix: rows are signals, columns are meanings
- Values in matrix represent strength of association
- Innately coded (i.e. given by genes)
- Evolution by natural selection can lead to *adaptation* of these genes
- Research question: under what condition will genes giving optimal signalling evolve?

Simplifying a bit: **one matrix** for production and reception

(Some of you may have already coded this up)

- Production: Look along rows and pick highest
- Reception: Look down columns and pick highest

	s1	s2	s3
m1	1	2	0
m2	0	1	1
m3	0	3	4

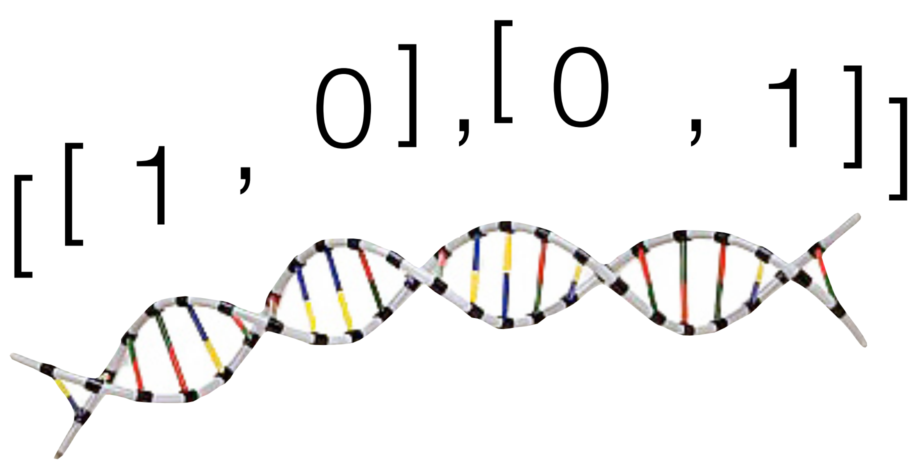
- **Does this necessarily mean that an agent's production and reception behaviours will be well-aligned (i.e. it will understand itself)?**

  A: Yes

  B: No

A few important questions

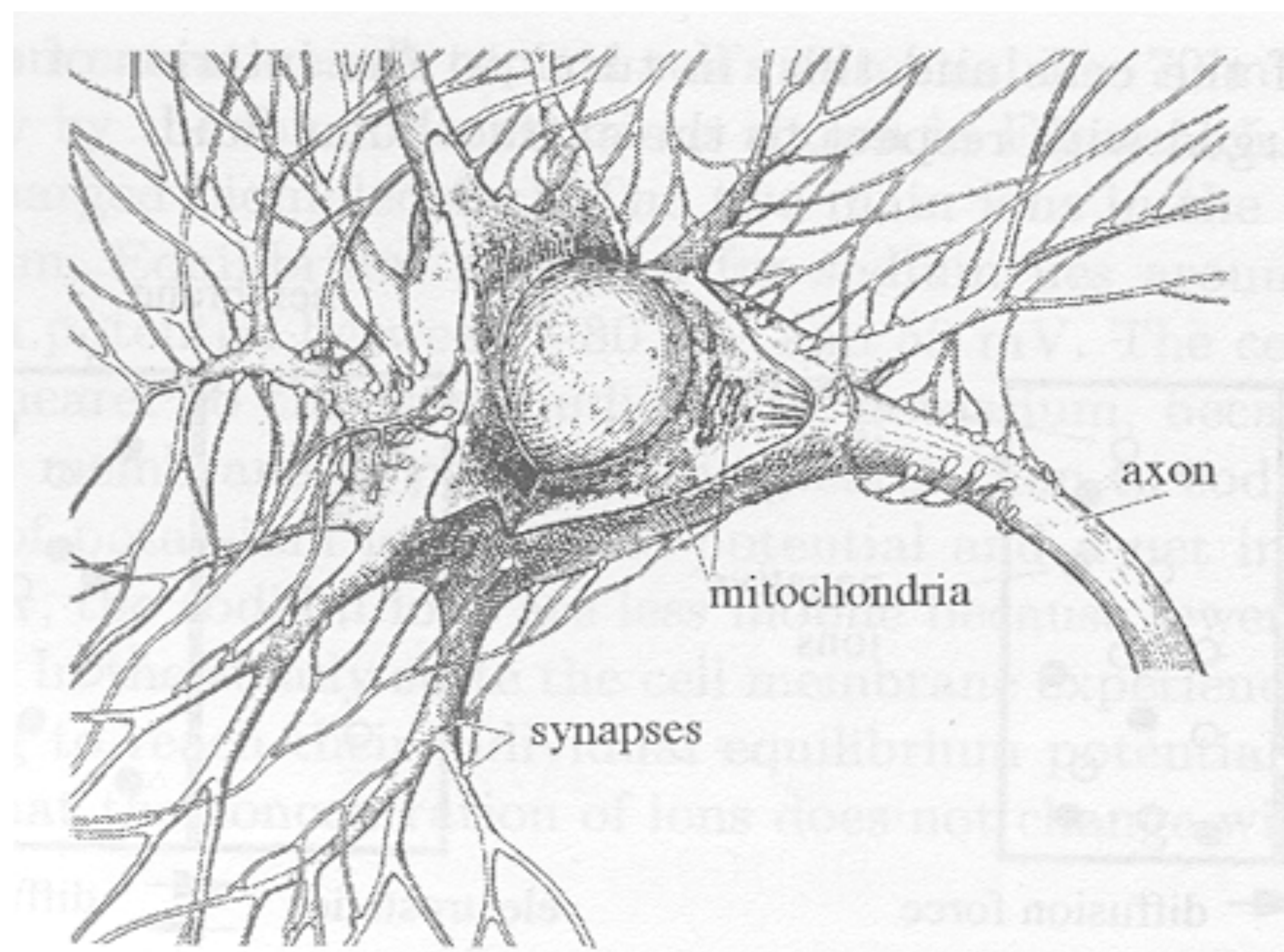
- What do these numbers in the matrix actually correspond to in reality?
- What relation does a model like this have to human language?
- Are there ways of getting “good numbers” other than by natural selection?
- We have equated genes and phenotype, but is this justified? If not, then what’s missing?



	s1	s2
m1	1	0
m2	0	1

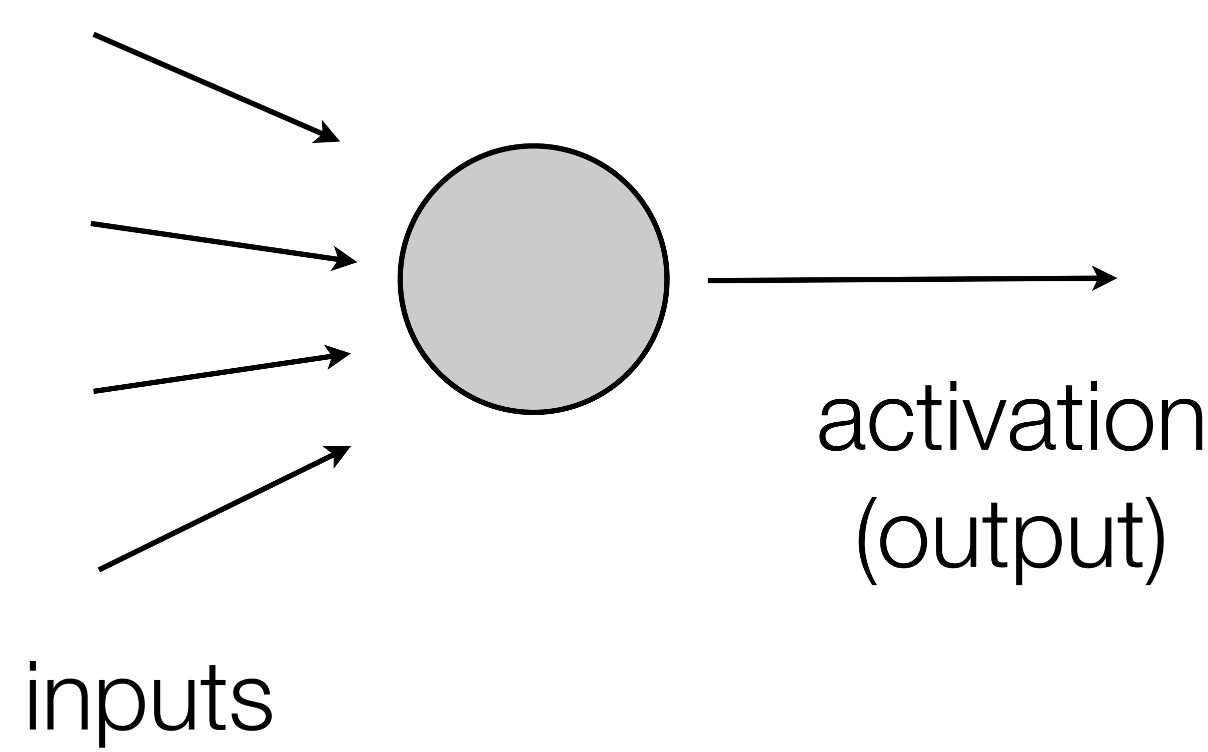
Neural networks

- A prominent approach to modelling cognition is called *connectionism*
- Principal tool is *artificial neural networks*: a (very) abstract model loosely based on how the brain works
- A neuron is a computational unit that sums up inputs and uses them to decide whether to produce an output



Neural networks

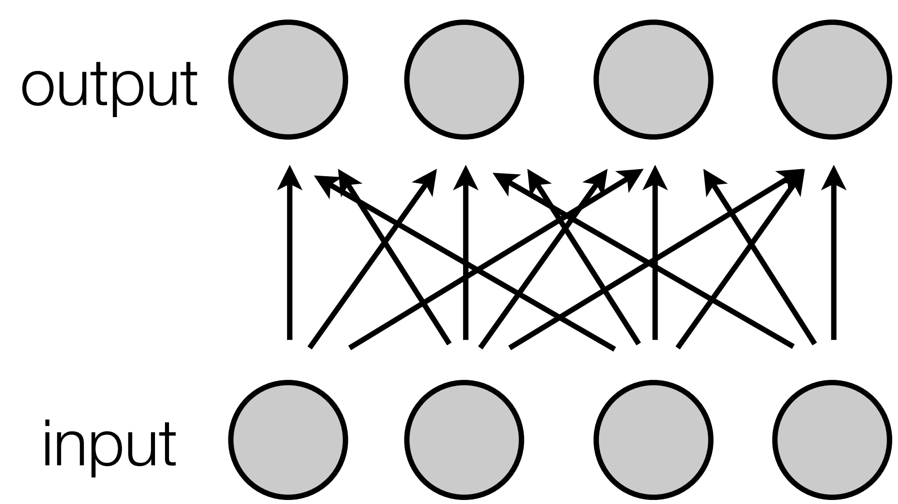
- A prominent approach to modelling cognition is called *connectionism*
- Principal tool is *artificial neural networks*: a (very) abstract model loosely based on how the brain works
- A neuron is a computational unit that sums up inputs and uses them to decide whether to produce an output



Neural networks

- Typically these “neurons” are nodes in a network (just like in the brain...)

- Many neural network models have nodes arranged in layers, with some layers interfacing with “input” and/or “output”



- Connections are *weighted*. In other words, they modify the signals passing along them. Think of this as representing the knowledge encoded by the network

What has this got to do with our model???

- We’ve actually been evolving a simple kind of neural network

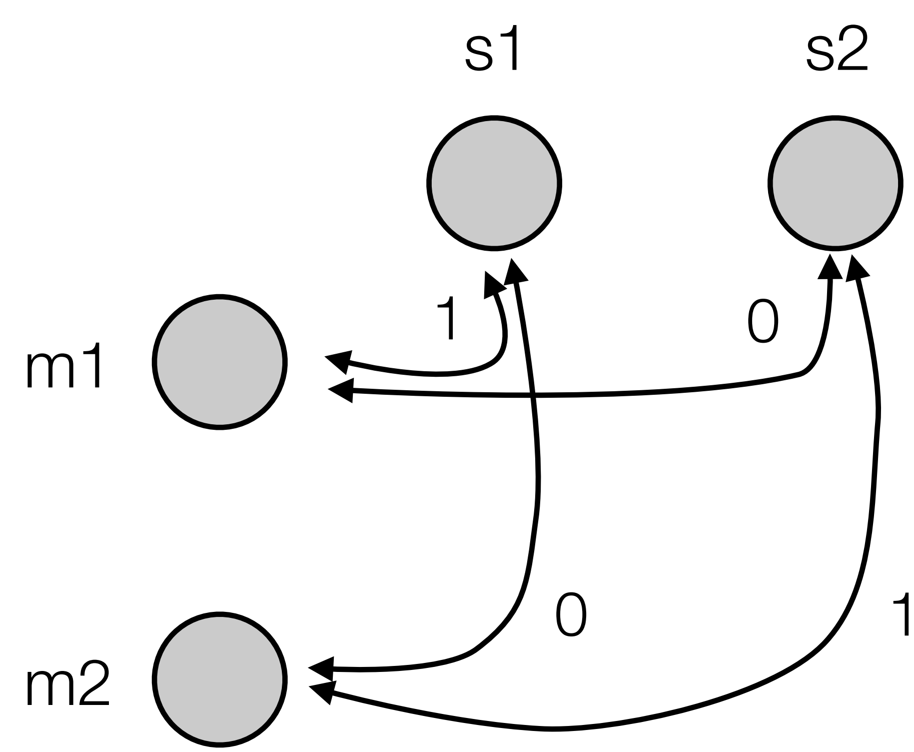
- It represents (very **very** abstractly) the brain of the organism we’re modelling

	s1	s2
m1	1	0
m2	0	1

What has this got to do with our model???

- We've actually been evolving a simple kind of neural network
- It represents (very **very** abstractly) the brain of the organism we're modelling

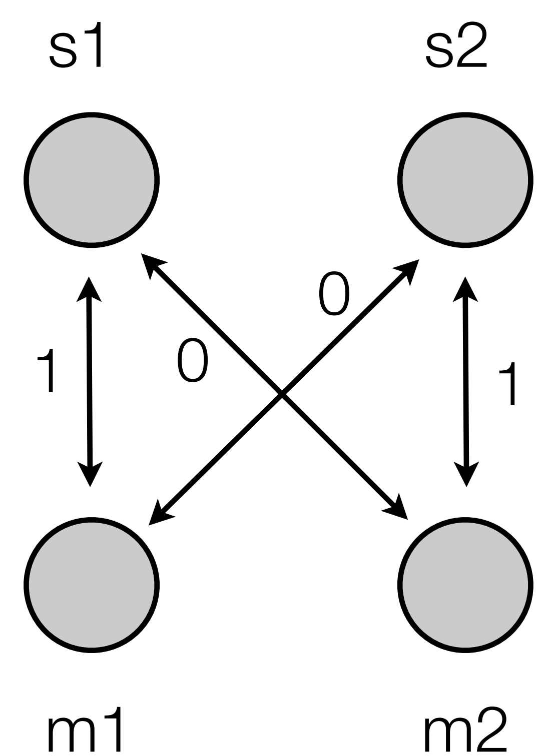
	s1	s2
m1	1	0
m2	0	1



What has this got to do with our model???

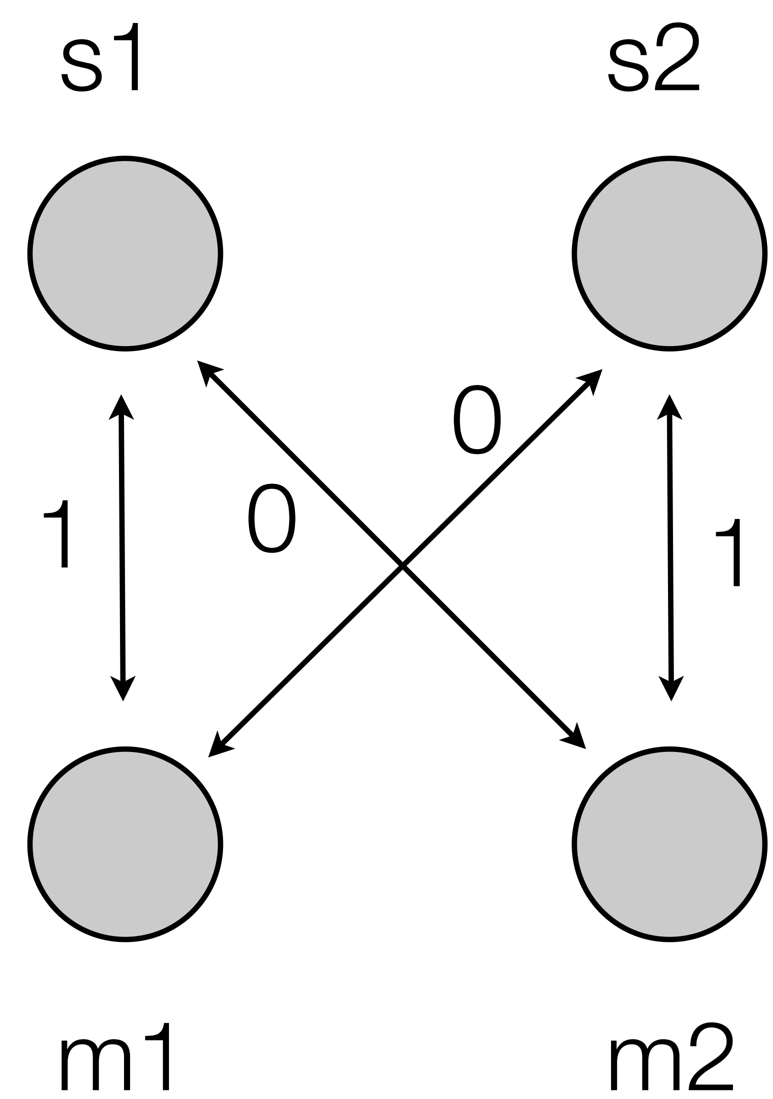
- We've actually been evolving a simple kind of neural network
- It represents (very **very** abstractly) the brain of the organism we're modelling

	s1	s2
m1	1	0
m2	0	1



How the network works

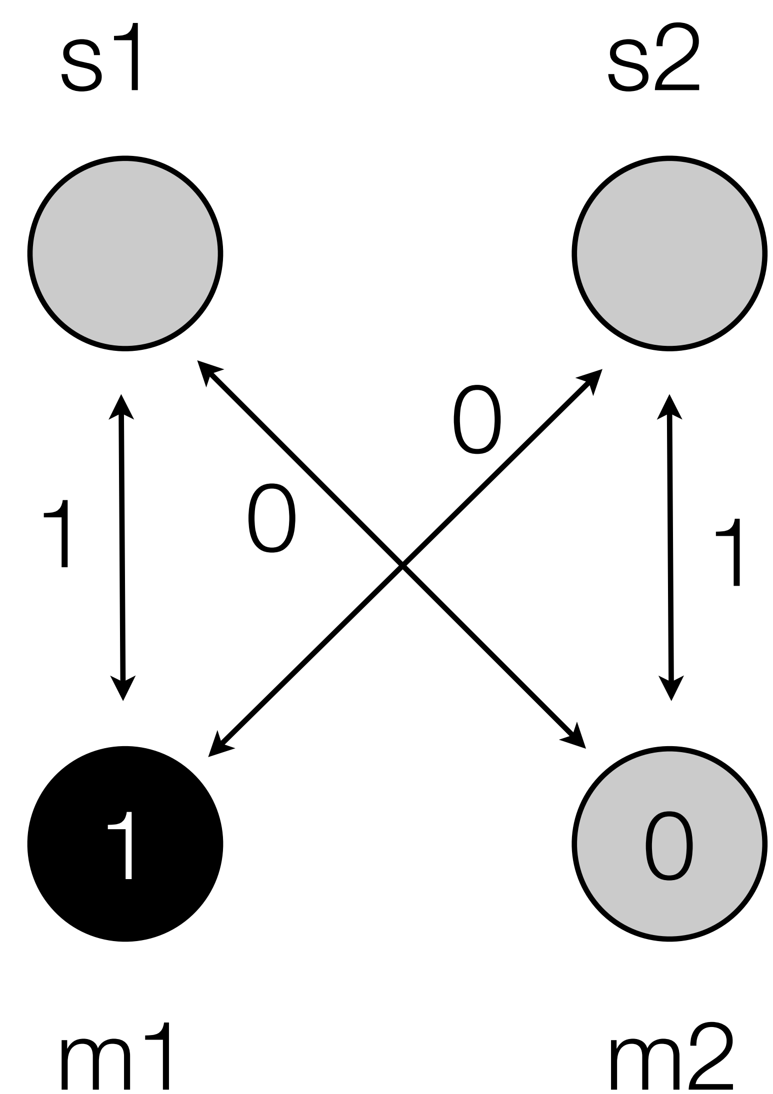
- Input nodes are "activated", and activation flows through the connections, modified by the weights and is summed up at the output nodes



How the network works

- Input nodes are “activated”, and activation flows through the connections, modified by the weights and is summed up at the output nodes

- Meaning 1 activated as input

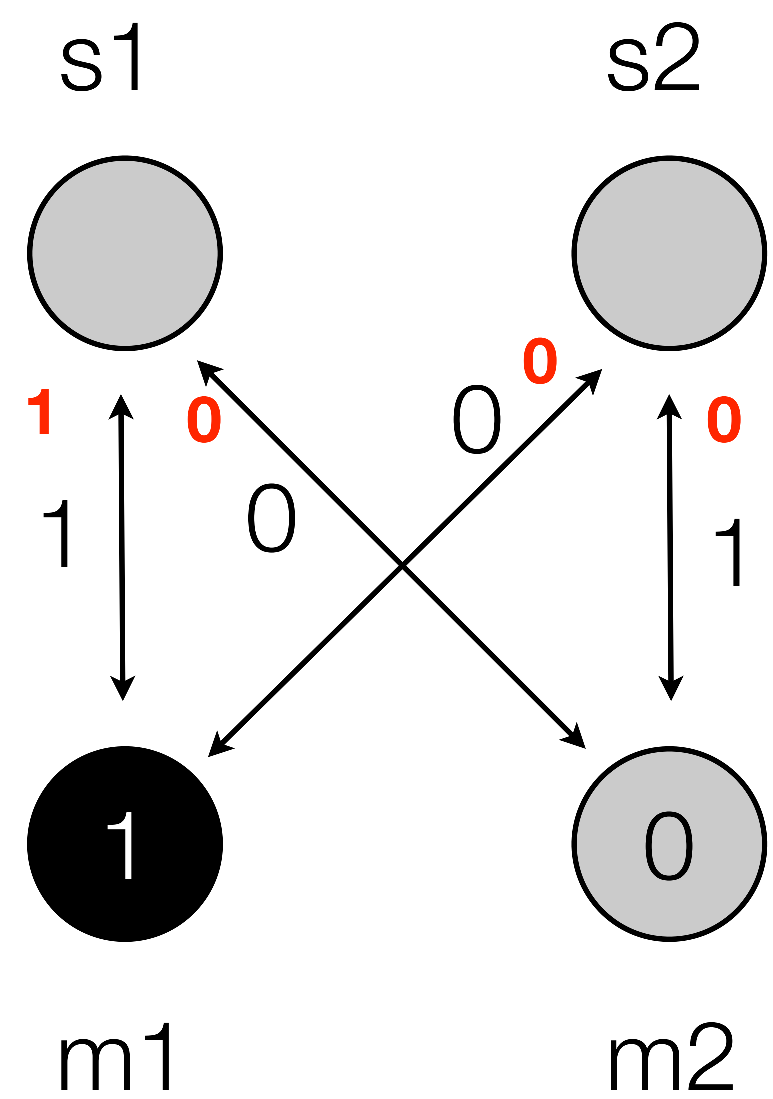


How the network works

- Input nodes are “activated”, and activation flows through the connections, modified by the weights and is summed up at the output nodes

- Meaning 1 activated as input

- Activation multiplied by weights as it passes down connections



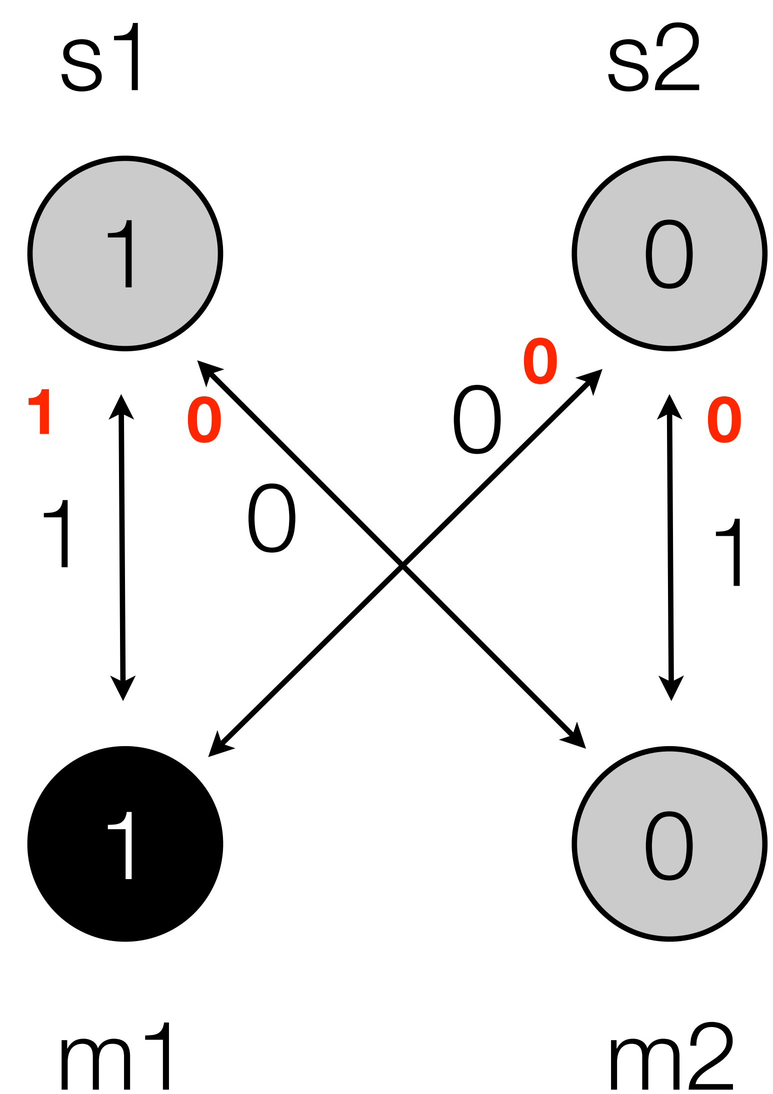
How the network works

- Input nodes are "activated", and activation flows through the connections, modified by the weights and is summed up at the output nodes

- Meaning 1 activated as input

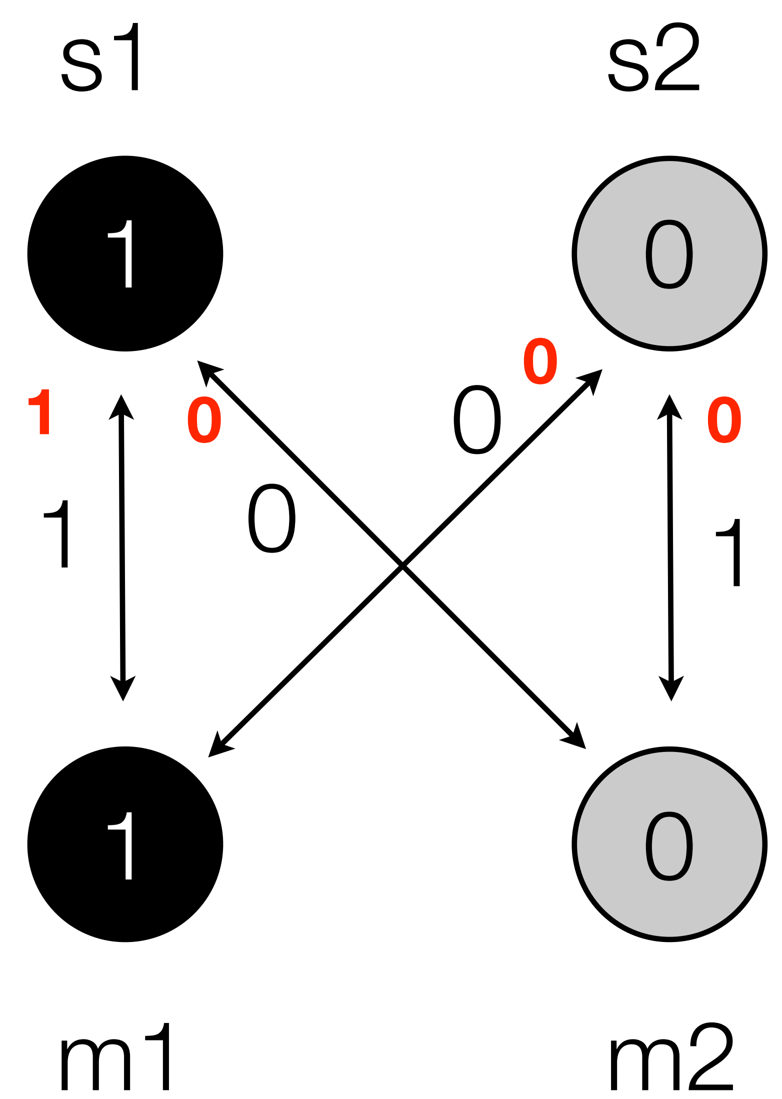
- Activation multiplied by weights as it passes down connections

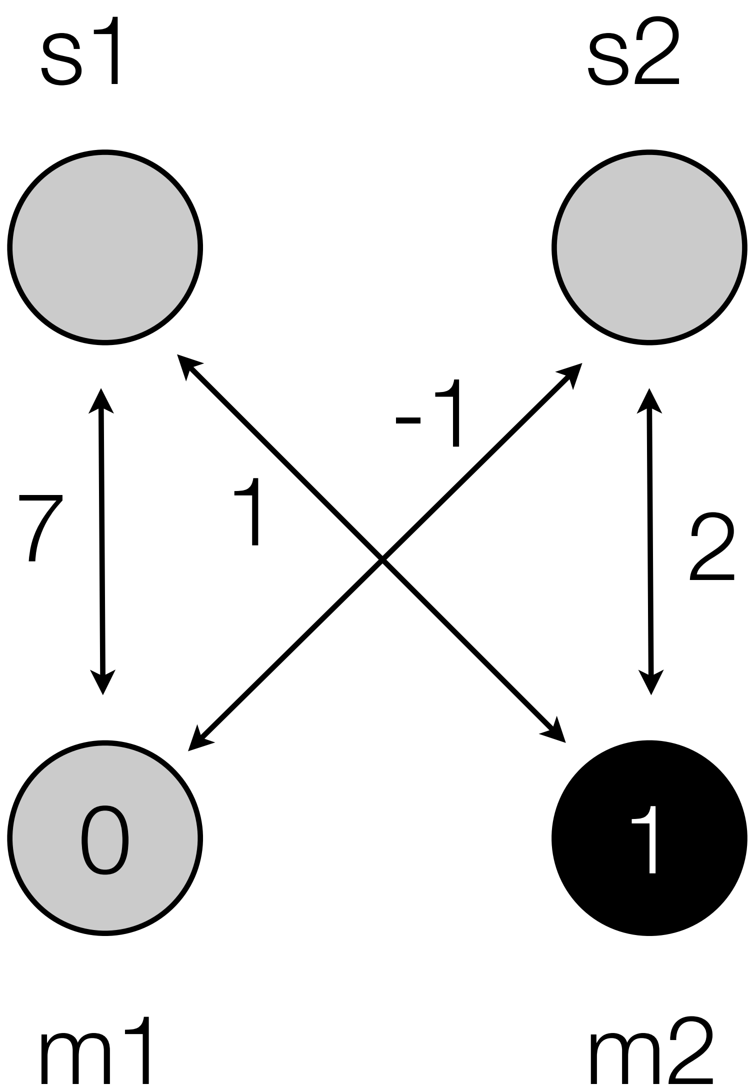
- Added up to give signal activations



How the network works

- Input nodes are “activated”, and activation flows through the connections, modified by the weights and is summed up at the output nodes
- Meaning 1 activated as input
- Activation multiplied by weights as it passes down connections
- Added up to give signal activations
- Signal activations converted into an actual signal output (e.g. pick the most activated node)

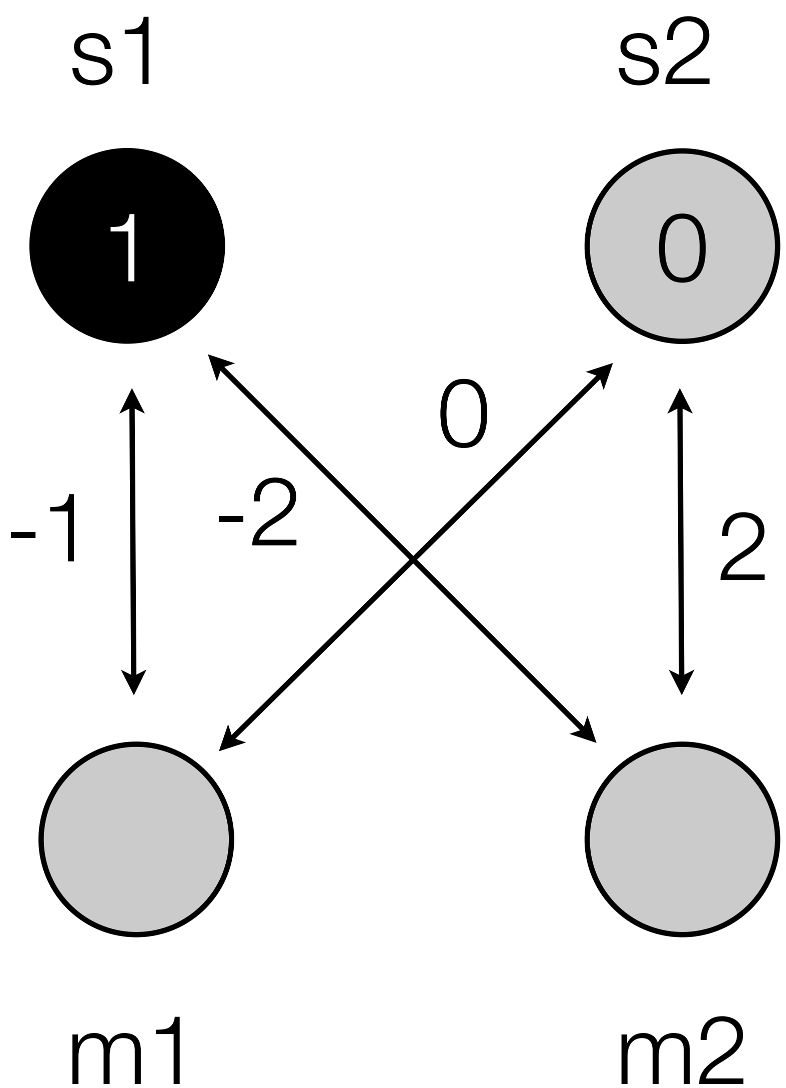




- **Which signal node will become active?**

  A: s1

  B: s2



- **Which meaning node will become active?**

  A: m1

  B: m2

Learning

- We've assumed agents are born with their knowledge set in these connection weights, and we've allowed biological evolution to decide what the weights should be

- But, there's another way... *learning*

- What is learning?

  - One view: learning occurs when an organism changes its internal state on the basis of experience

- Neural networks (and brains!) are designed so that connections change with experience. Learning breaks the simple connection between genes and phenotype.

A **very** simple model of learning

- Assume that, at least some of the time, agents observe meanings and signals appearing together in the environment

- In other words, their meaning and signal nodes can both be activated together as input

- Hebbian approach to learning:
  "any two cells or systems of cells that are repeatedly active at the same time will tend to become 'associated', so that activity in one facilitates activity in the other." (Hebb 1949)

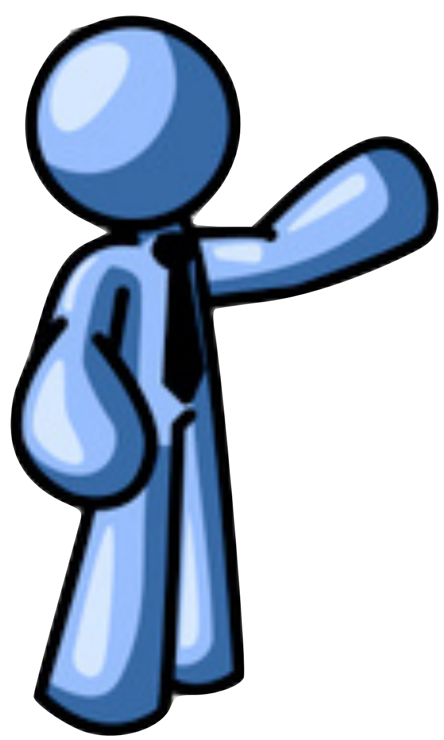
- Simple approach:
  Start with all weights zero, and increase connection weight whenever two nodes fire together

Example

	s1	s2
m1	0	0
m2	0	0

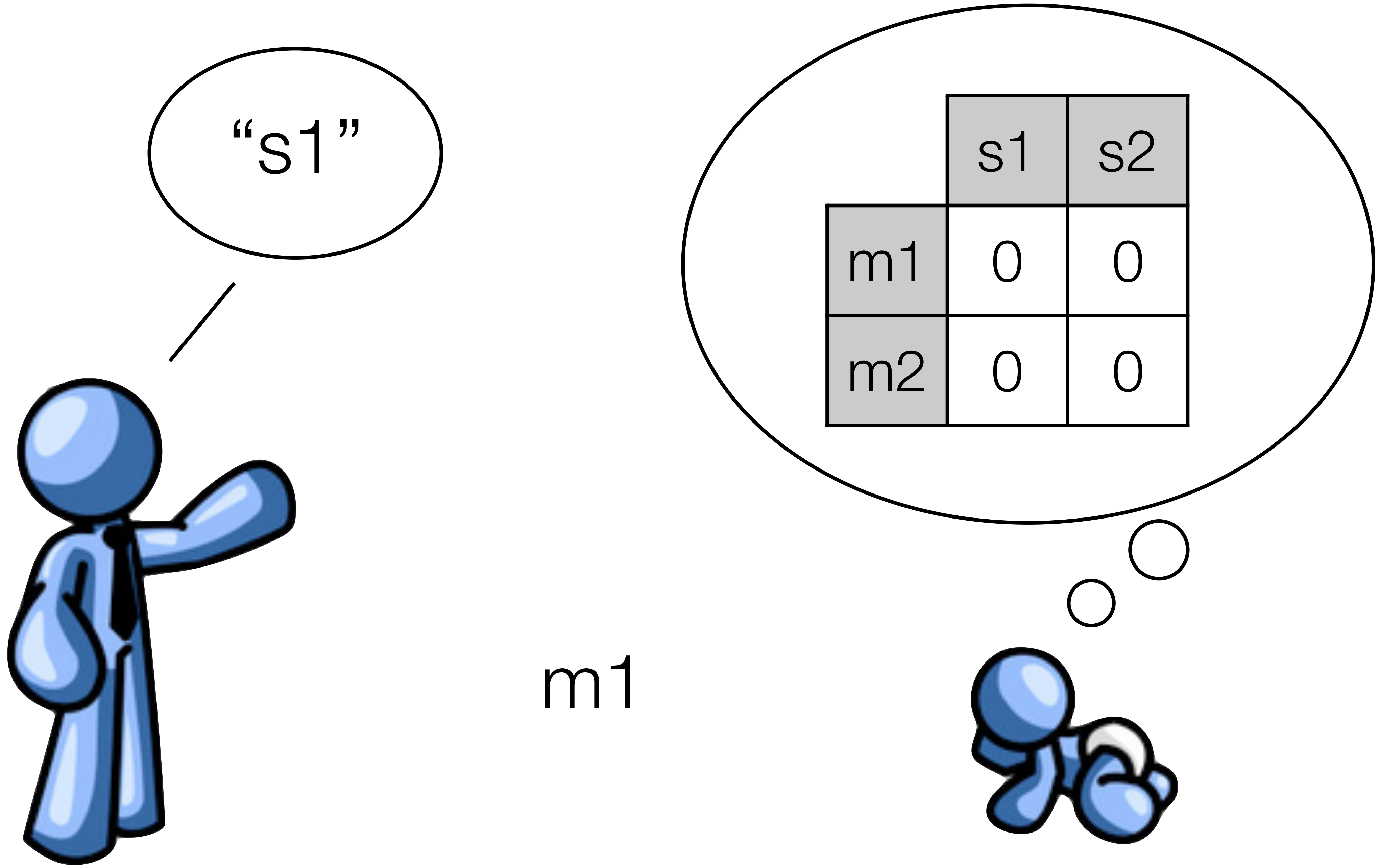
Example

	s1	s2
m1	0	0
m2	0	0



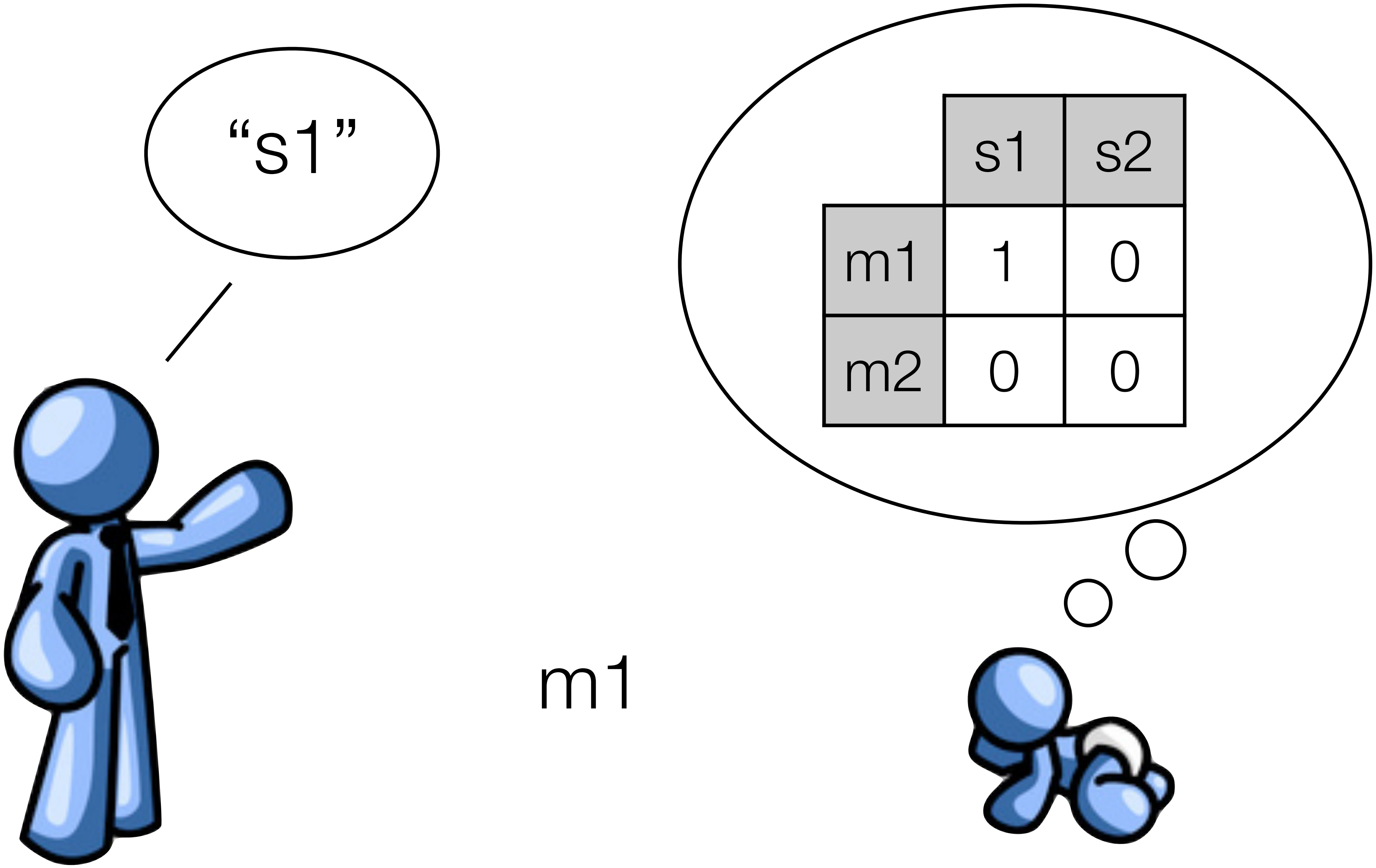
m1

Example



	s1	s2
m1	0	0
m2	0	0

Example



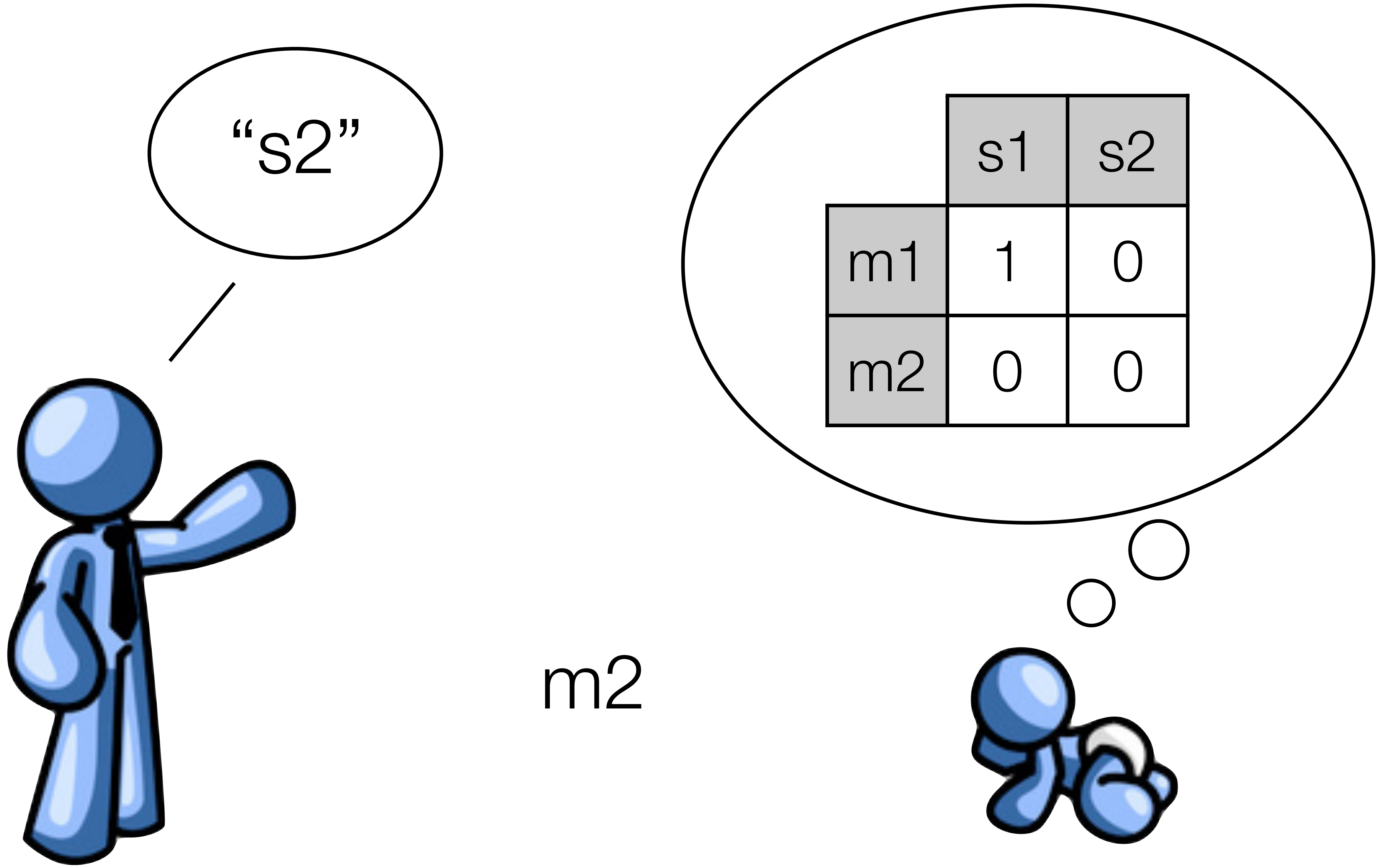
	s1	s2
m1	1	0
m2	0	0

Example

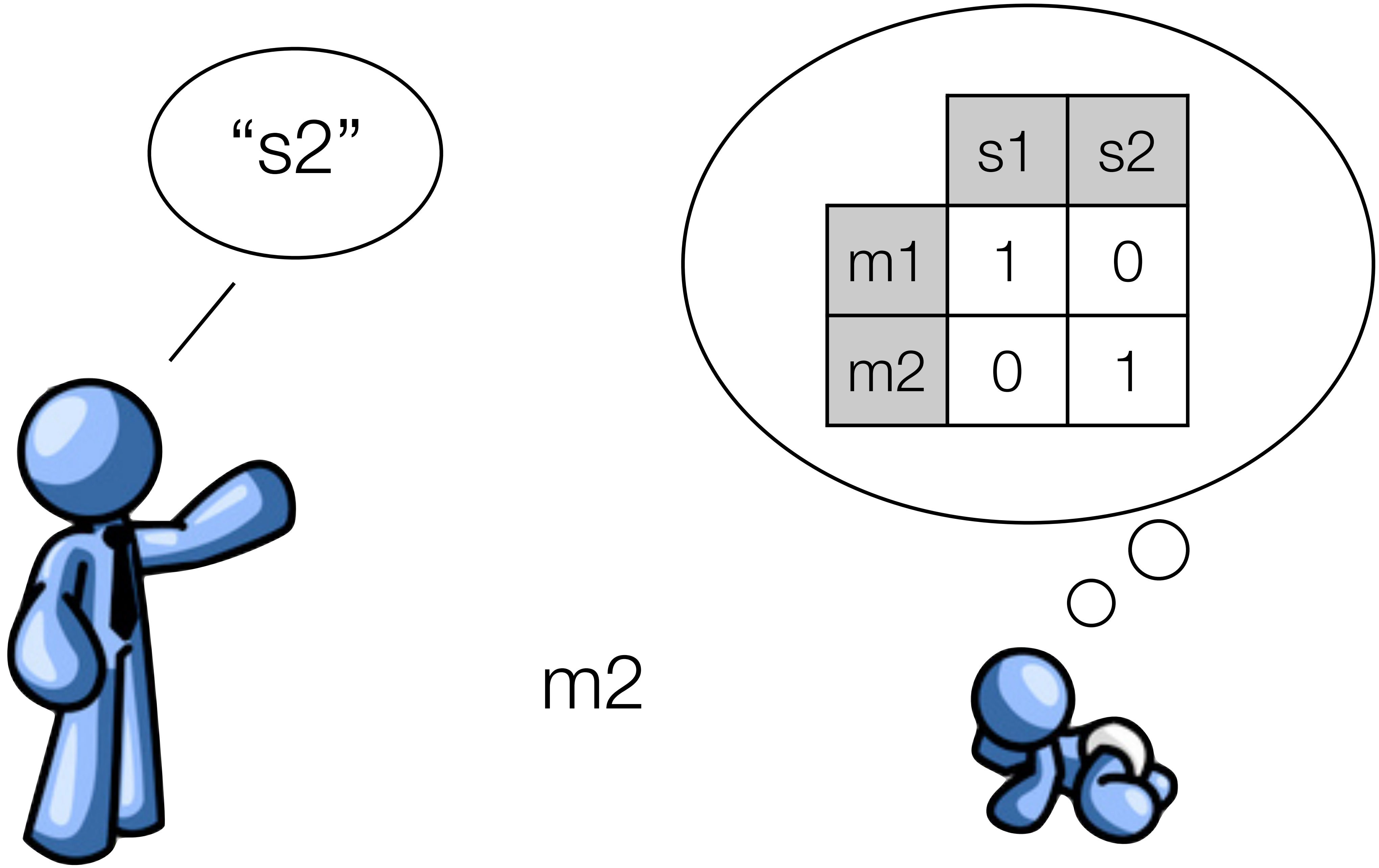
	s1	s2
m1	1	0
m2	0	0

m2

Example



Example

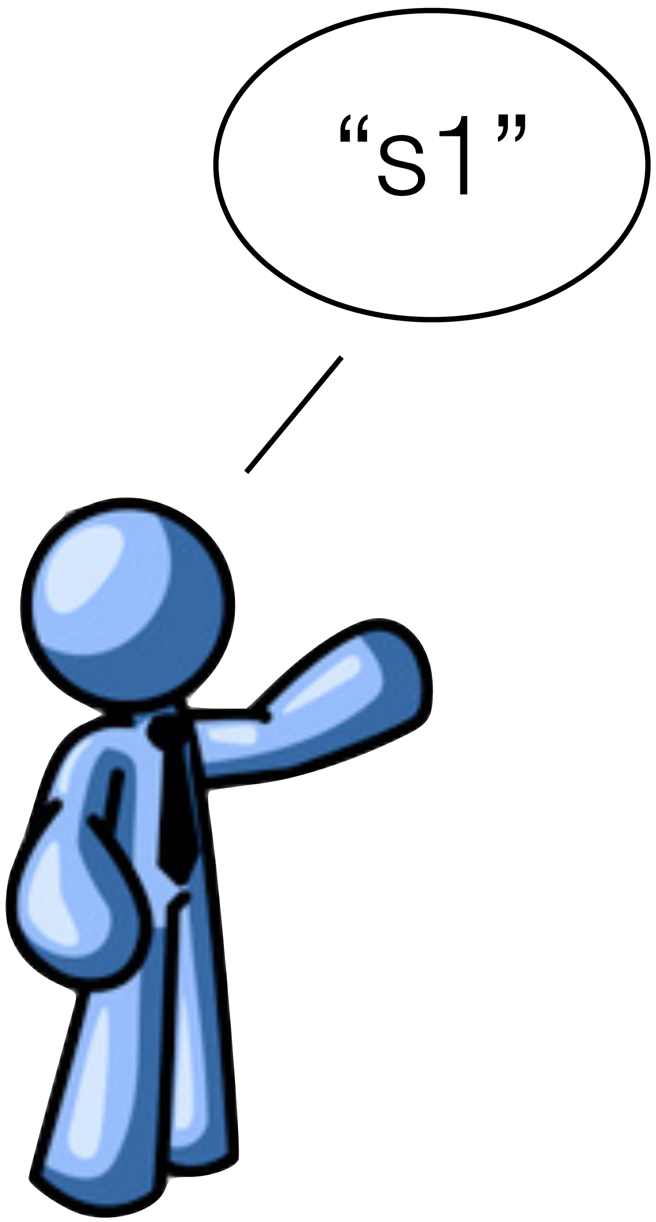


Example

	s1	s2
m1	1	0
m2	0	1

m1

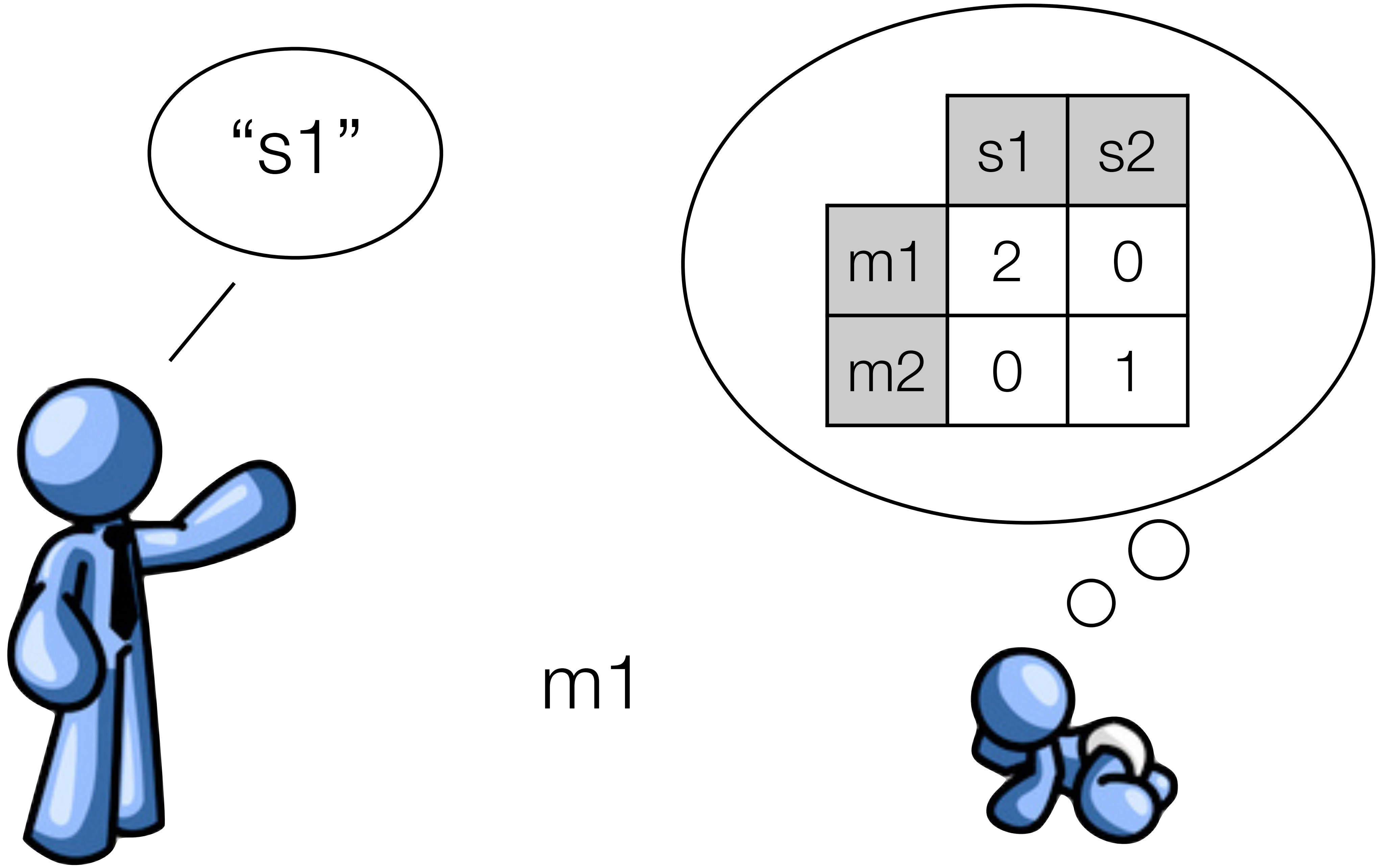
Example



m1

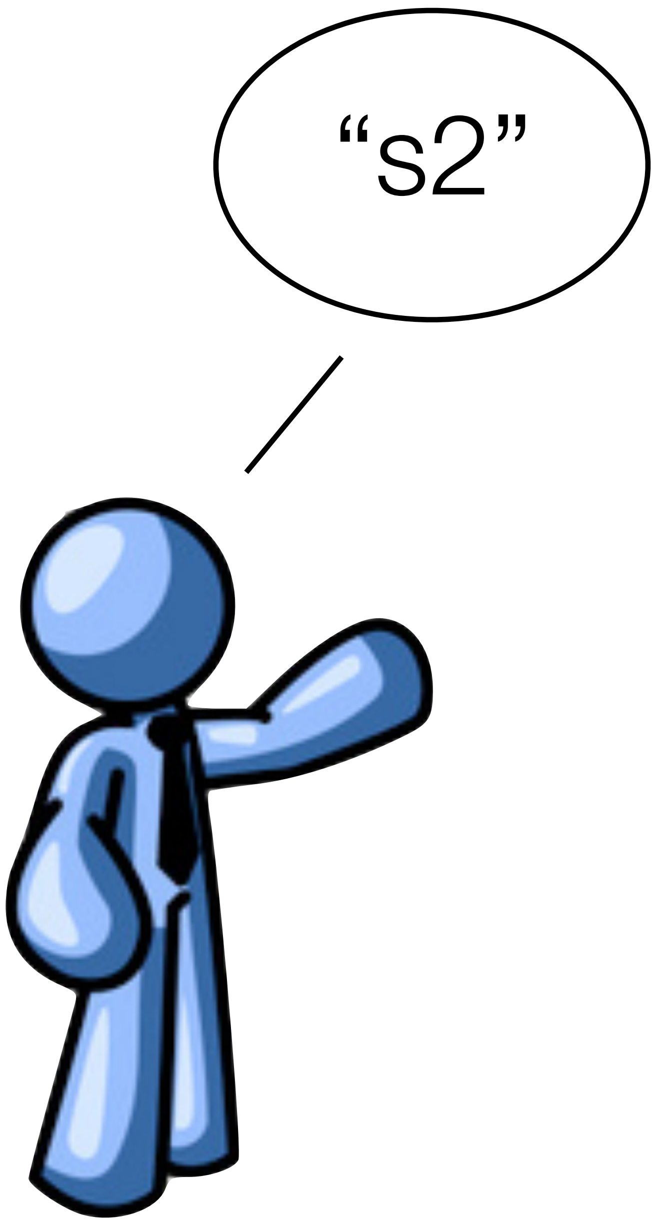
	s1	s2
m1	1	0
m2	0	1

Example



	s1	s2
m1	2	0
m2	0	1

Example



	s1	s2
m1	2	1
m2	0	1

m1

A **very** simple model of learning

- We can think of this in terms of a Hebbian-like associationist neural network learning procedure
- ... but it's essentially equivalent to keeping a frequency count of all pairings of meanings and signals.
- It's the simplest model of learning we could think of
- We can implement it by adding two lines of python to the code for our very first model
    - Plus a little bit of other stuff to go to a single-matrix model of production/reception
- Key questions: is this model of learning sufficient? What can an agent with this learning algorithm actually acquire? Does it give another route to explaining where optimal signalling comes from?

Questions for discussion (if we have time)

- What do you think of the assumption that learners see signals **and** meanings?

- How else might you model it?

- We are increasing weights between co-active units. Is there anything else we could or should be doing?