

While you are waiting...

- **socrative.com**, room number **SIMLANG2016**

Simulating Language

Lecture 6: Learning bias

Simon Kirby

simon@ling.ed.ac.uk



Summary - from evolution to learning

- A big difference between animal signalling and human language
 - Animals typically are born with the relationship between meanings and signals given innately in their genes (as a first approximation)
 - Humans *acquire* this relationship during development
- In our model, the relationship between meanings and signals is represented by connection weights in a network
 - Our animal model has these fixed in each agent, with the possibility of biological evolution
 - Our human model is born with all weights set to zero, with the possibility of changing them in response to hearing utterances (i.e. learning)

How good is our model at learning?

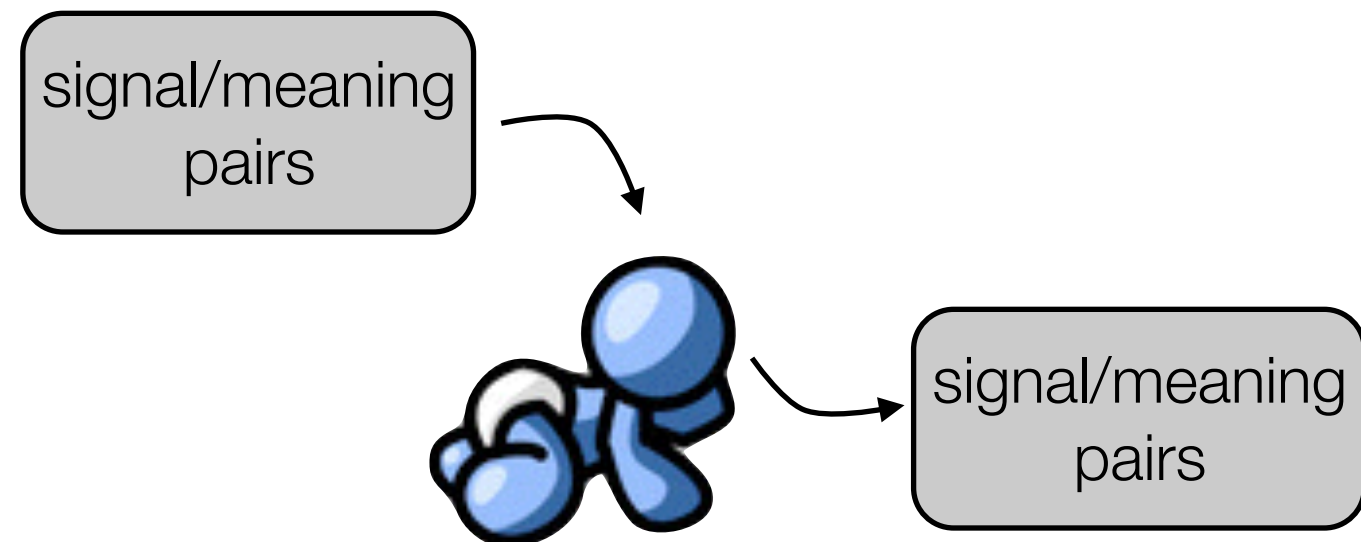
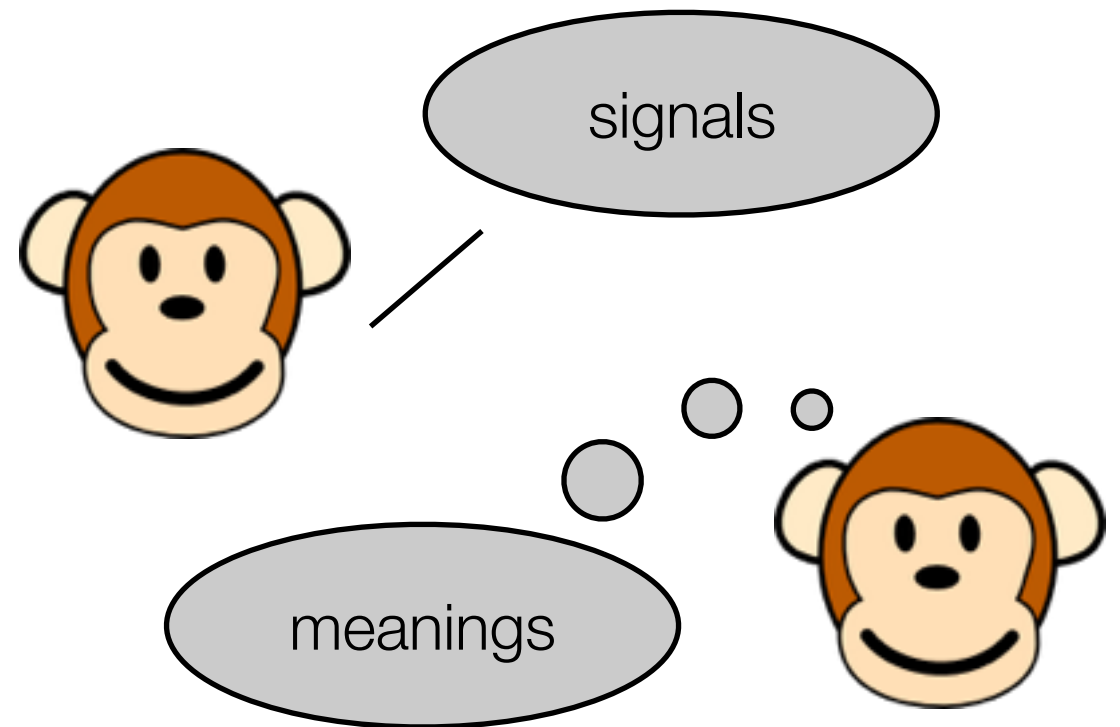
- What does it mean for something to be ‘good’ at learning?
 - One answer: will two agents given the same data be able to **communicate**? Will a learner be able to communicate with its teacher?
 - Another answer: given some training data, can it **recall** that data?
 - A third answer: given some training data, can it **generalise** correctly to unseen data?
- **Which of these do you think is the most important sense of ‘good at learning’ for human language?**
- **A:** Communication
- **B:** Recall
- **C:** Generalisation

How good is our model at learning?

- What does it mean for something to be ‘good’ at learning?
 - One answer: will two agents given the same data be able to **communicate**? Will a learner be able to communicate with its teacher?
 - Another answer: given some training data, can it **recall** that data?
 - A third answer: given some training data, can it **generalise** correctly to unseen data?
- Our training data is meaning-signal pairs, so an obvious test is simply whether meanings correctly map to signals (and vice versa) after learning
- So, some kinds of learner will be good at learning, and others will be bad, right?
- Not as simple as that... **it will depend on what is being learned**

A new kind of question

- Previously, we were interested in how good two innate signalling systems were for communication
- Now, we want to know what kinds of errors a particular learner makes with a particular language



An aside: how to do this with our code

- Use **train** to train a particular network with a set of data. e.g.:

```
>>> net = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]
>>> train(net, [[0, 0], [1, 1], [2, 1]])
>>> net
[[1, 0, 0], [0, 1, 0], [0, 1, 0]]
```

- Then you can test what the resulting network's reception/production behaviour is using **wta** in combination with **production_weights** and **reception_weights**. e.g.:

```
>>> wta(production_weights(net, 0))
0
>>> wta(production_weights(net, 2))
1
>>> wta(reception_weights(net, 2))
0
```

What about our learner?

- How well does it learn?
- Given an optimal language, it learns well:

TRAINING

$m1 \rightarrow s1$

$m2 \rightarrow s2$

$m3 \rightarrow s3$

	s1	s2	s3
m1	0	0	0
m2	0	0	0
m3	0	0	0

What about our learner?

- How well does it learn?
- Given an optimal language, it learns well:

TRAINING

$m1 \rightarrow s1$

$m2 \rightarrow s2$

$m3 \rightarrow s3$

	s1	s2	s3
m1	1	0	0
m2	0	1	0
m3	0	0	1

RESULT

$m1 \rightarrow s1$

$m2 \rightarrow s2$

$m3 \rightarrow s3$

What about our learner?

- How well does it learn?
- Given a language with synonymy?

A: s1 only

B: s2 only

C: s1 and s2, in a 1:2 ratio

D: s1 and s2, with equal frequency

TRAINING

$m1 \rightarrow s1$

$m1 \rightarrow s2$

$m1 \rightarrow s2$

	s1	s2	s3
m1	0	0	0
m2	0	0	0
m3	0	0	0

What about our learner?

- How well does it learn?
- Given a language with synonymy, production behaviour depends on frequency of items in training:

TRAINING

$m1 \rightarrow s1$

$m1 \rightarrow s2$

$m1 \rightarrow s2$

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

RESULT

$m1 \rightarrow s2$ only

What about our learner?

- How well does it **generalise**?
- Unable to correctly generalise an optimal language:

TRAINING

$m1 \rightarrow s1$

$m2 \rightarrow s2$

~~$m3 \rightarrow s3$~~

	s1	s2	s3
m1	1	0	0
m2	0	1	0
m3	0	0	0

RESULT

What about our learner?

- How well does it **generalise**?
- Unable to correctly generalise an optimal language:

TRAINING

$m1 \rightarrow s1$

$m2 \rightarrow s2$

~~$m3 \rightarrow s3$~~

	s1	s2	s3
m1	1	0	0
m2	0	1	0
m3	0	0	0

RESULT

$m1 \rightarrow s1$

$m2 \rightarrow s2$

$m3 \rightarrow s1, s2, s3$

What about our learner?

- How well does it **generalise**?
- Unable to correctly generalise to a maximally ambiguous language:

TRAINING

$m1 \rightarrow s1$

$m2 \rightarrow s1$

~~$m3 \rightarrow s1$~~

	s1	s2	s3
m1	1	0	0
m2	1	0	0
m3	0	0	0

RESULT

$m1 \rightarrow s1$

$m2 \rightarrow s1$

$m3 \rightarrow s1, s2, s3$

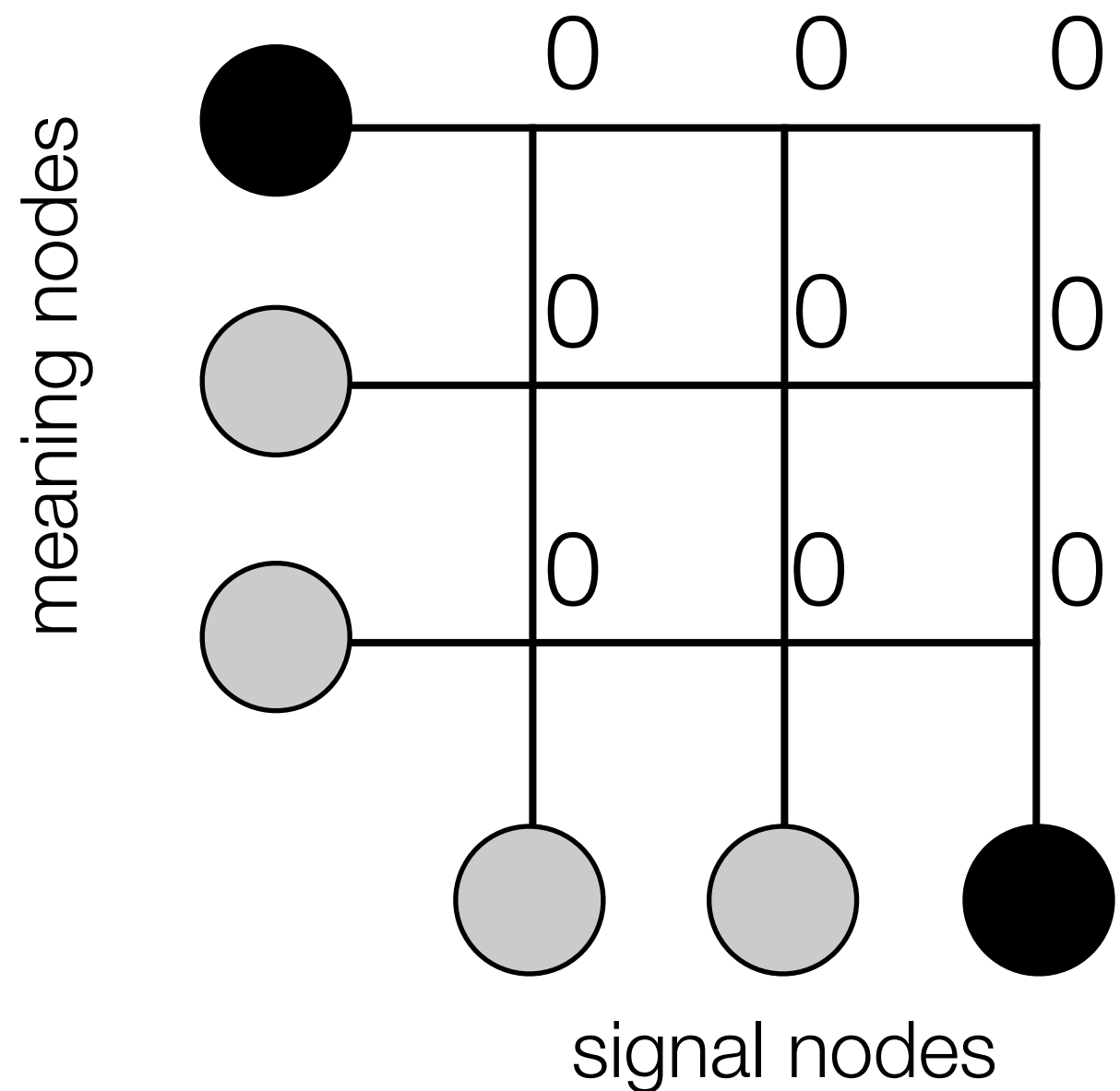
Bias

- Our learner is not a completely “blank slate”. It responds differently to different training sets
 - In this case: it struggles with synonyms, but is otherwise faithful to its data (to the extent that it misses ‘obvious’ generalisations)
- Where does this behaviour come from?
- Features of the architecture of the model create an inherent *learning bias* which may favour some languages over others
 - Cf. Christiansen & Devlin (1997): a very different kind of neural network making the same point: learning bias means some languages are more learnable than others
- What features could we modify to manipulate bias?
- One possibility: the way we update the weights...

Our weight-update rule

- If signal node and meaning node are active, increase connection weight by one

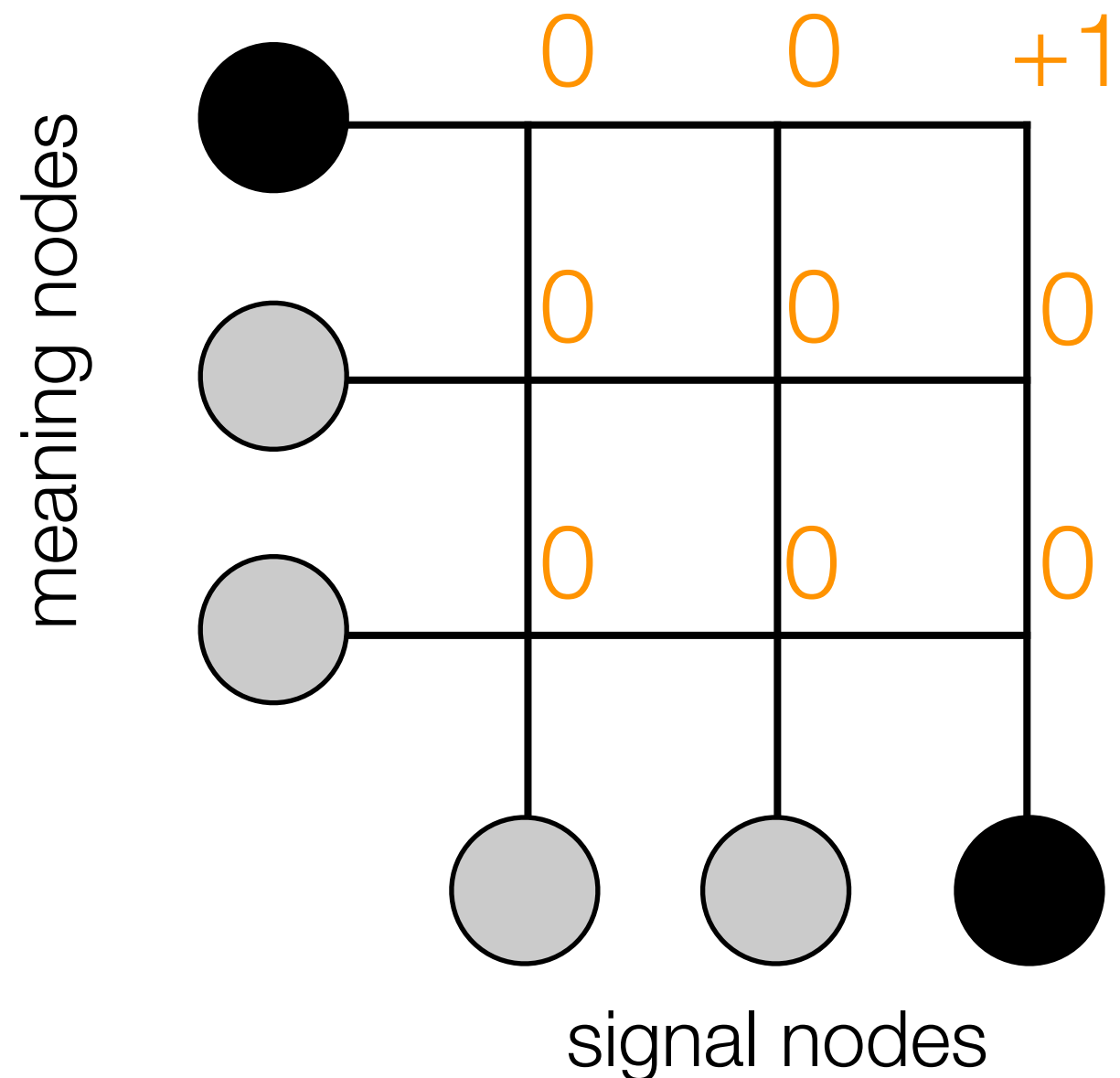
Observation:
 $m1 \leftrightarrow s3$



Our weight-update rule

- If signal node and meaning node are active, increase connection weight by one

Observation:
 $m1 \leftrightarrow s3$

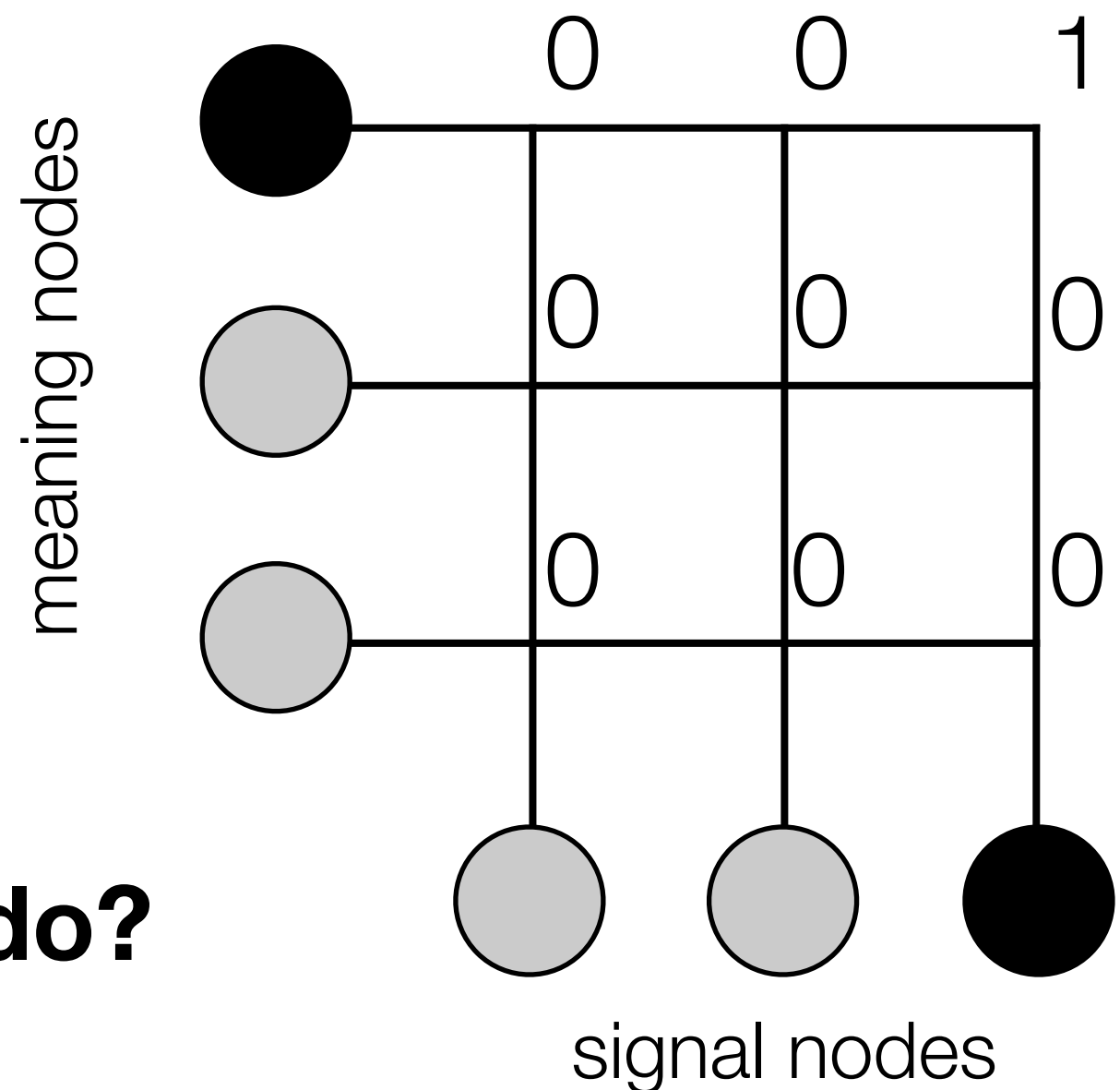


Our weight-update rule

- If signal node and meaning node are active, increase connection weight by one

Observation:
 $m1 \leftrightarrow s3$

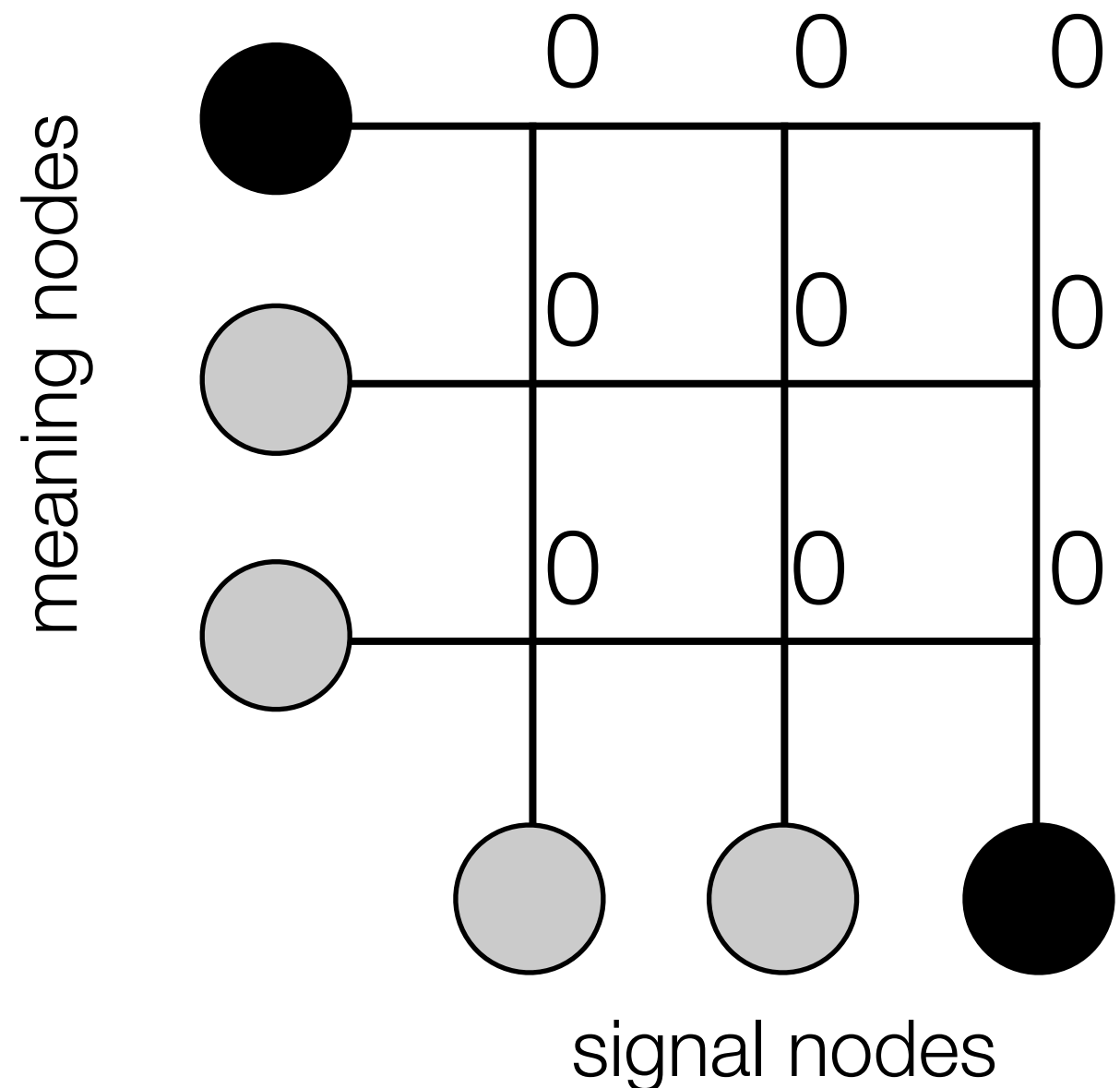
What else could we do?



There are other possibilities

- Some of you wondered if it was possible to *reduce* connection weights between nodes that were ‘competing’ for the same meaning or signal

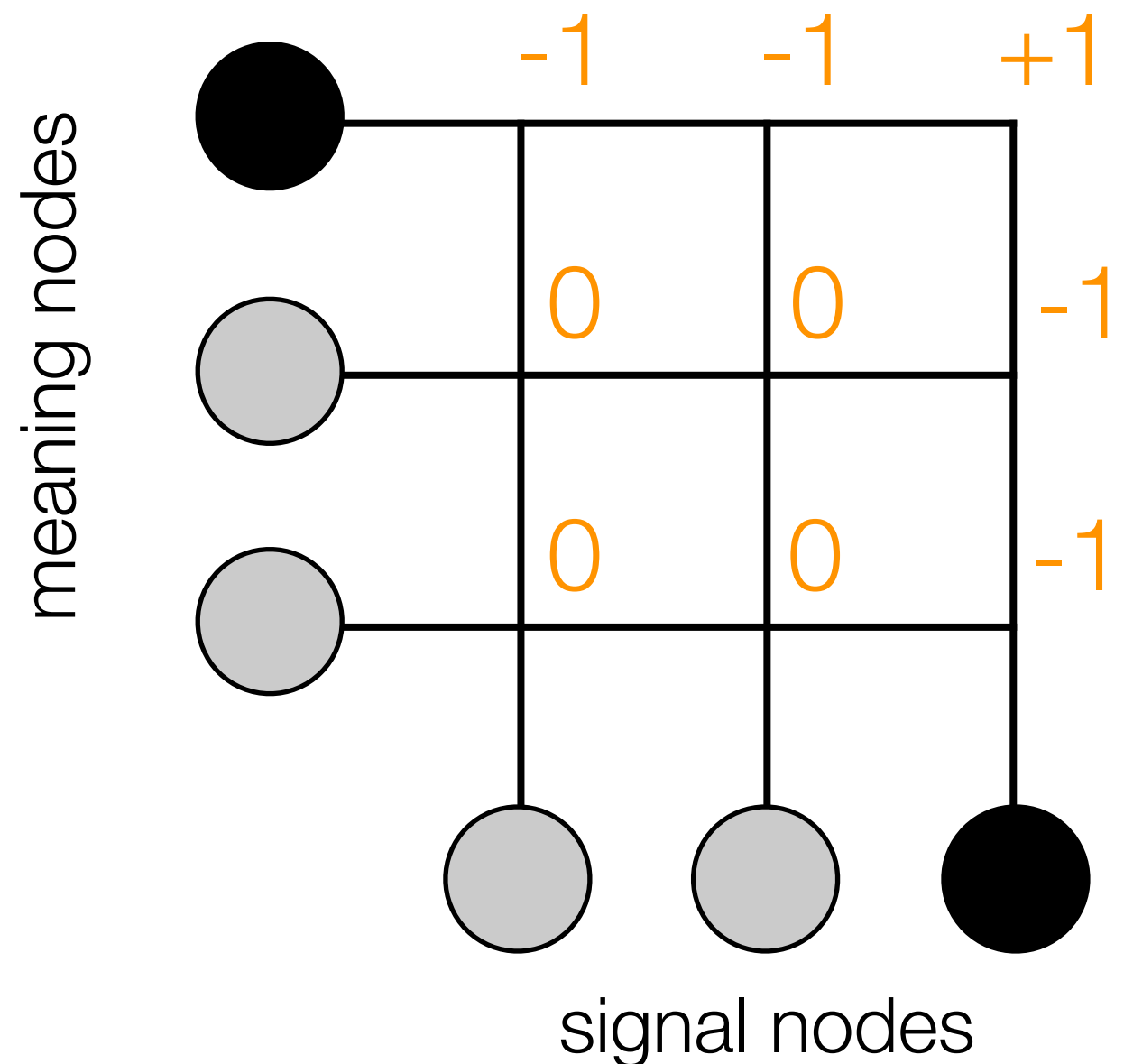
Observation:
 $m1 \leftrightarrow s3$



There are other possibilities

- Some of you wondered if it was possible to *reduce* connection weights between nodes that were ‘competing’ for the same meaning or signal

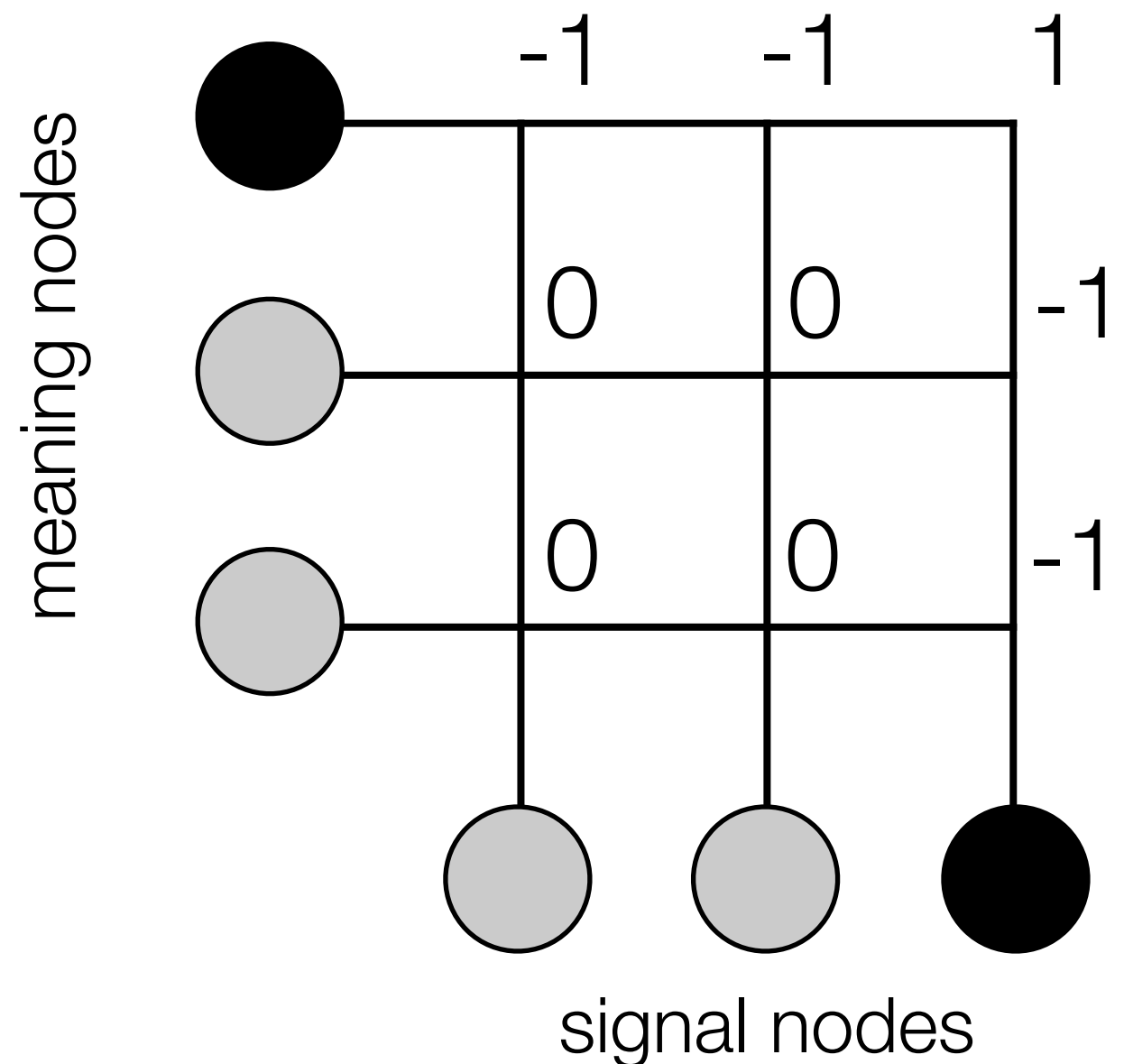
Observation:
 $m1 \leftrightarrow s3$



There are other possibilities

- Some of you wondered if it was possible to *reduce* connection weights between nodes that were 'competing' for the same meaning or signal

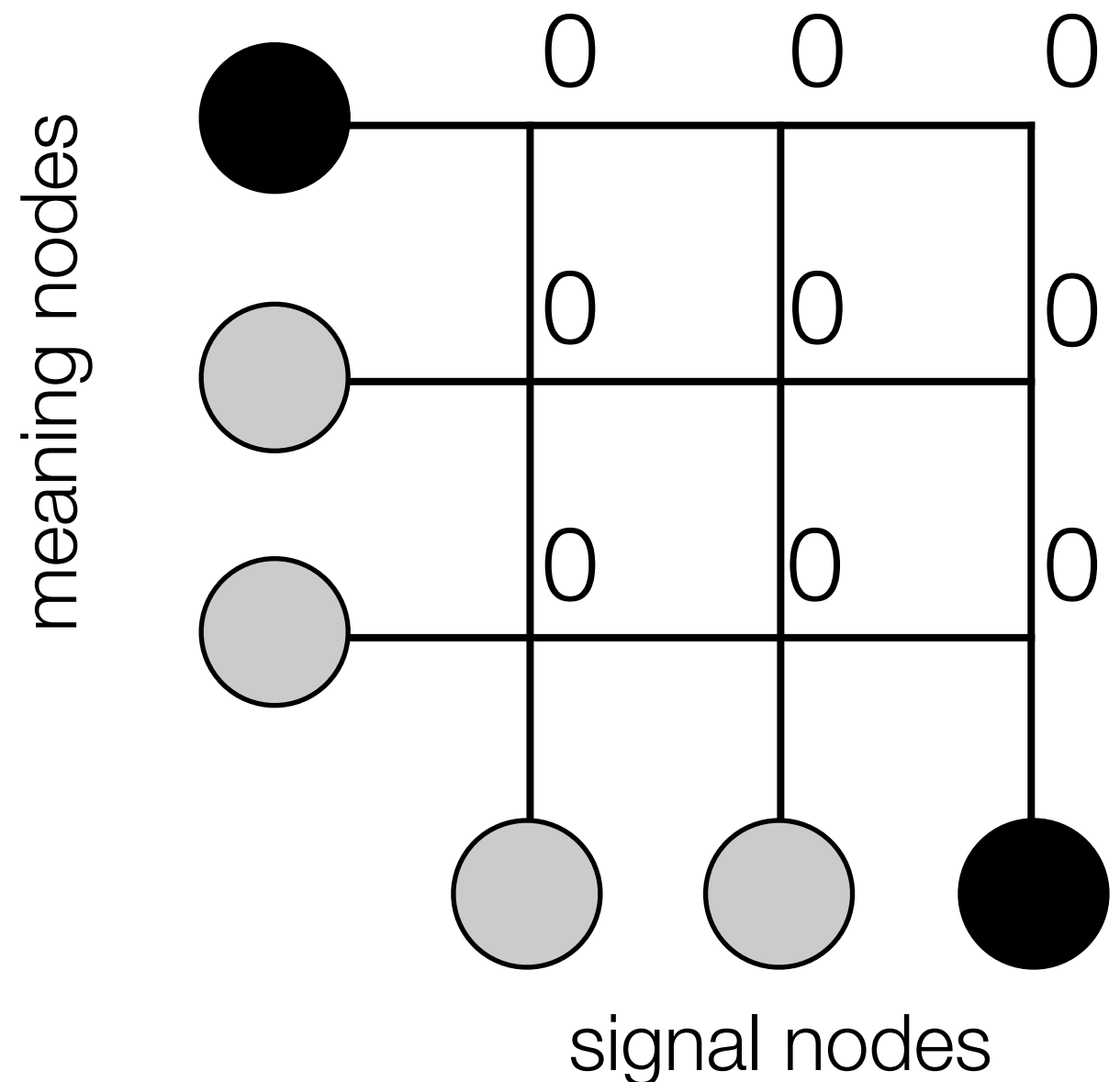
Observation:
 $m1 \leftrightarrow s3$



There are other possibilities

- Maybe we should reduce connection weights between nodes that were simultaneously inactive

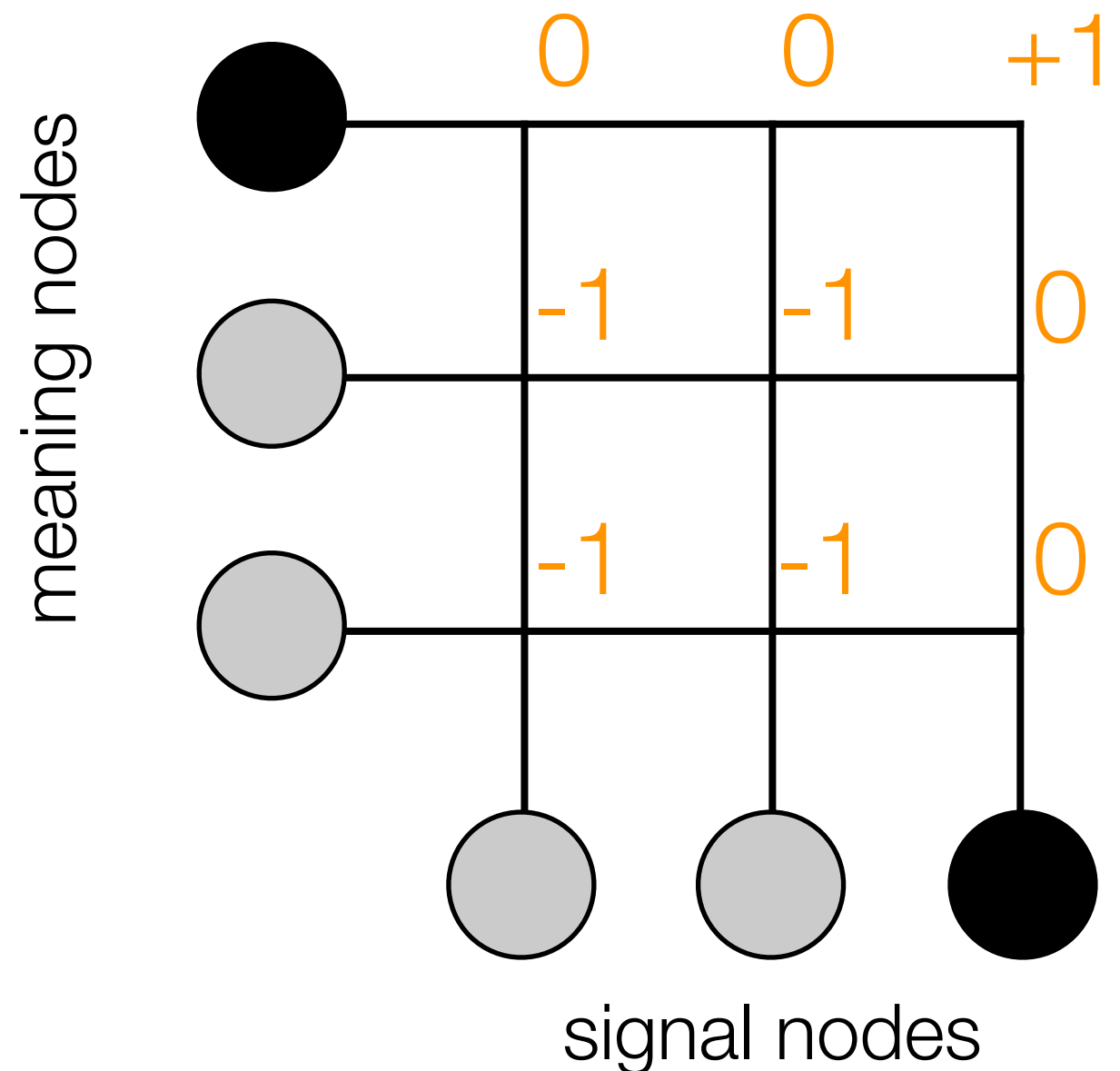
Observation:
 $m1 \leftrightarrow s3$



There are other possibilities

- Maybe we should reduce connection weights between nodes that were simultaneously inactive

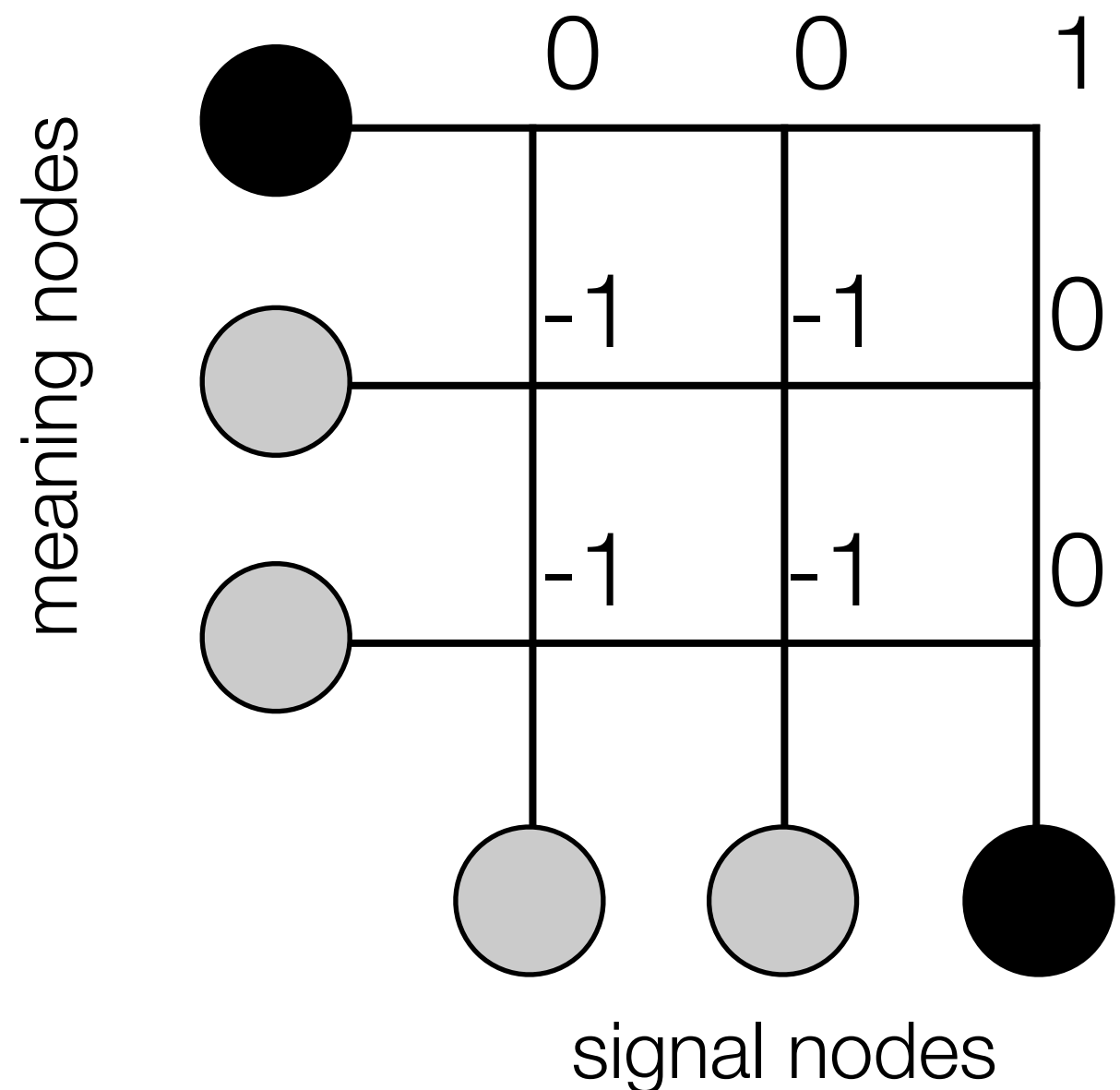
Observation:
 $m1 \leftrightarrow s3$



There are other possibilities

- Maybe we should reduce connection weights between nodes that were simultaneously inactive

Observation:
 $m1 \leftrightarrow s3$



A set of possible *weight update rules*

- We need to specify what will happen to a weight in four different situations:

$\Delta w_{m_i s_j} = ?$ if both m_i and s_j are active

$\Delta w_{m_i s_j} = ?$ if m_i is active and s_j is inactive

$\Delta w_{m_i s_j} = ?$ if m_i is inactive and s_j is active

$\Delta w_{m_i s_j} = ?$ if both m_i and s_j are inactive

A set of possible *weight update rules*

- We need to specify what will happen to a weight in four different situations:

$\Delta w_{m_i s_j} = +1$ if both m_i and s_j are active
 $\Delta w_{m_i s_j} = 0$ if m_i is active and s_j is inactive
 $\Delta w_{m_i s_j} = 0$ if m_i is inactive and s_j is active
 $\Delta w_{m_i s_j} = 0$ if both m_i and s_j are inactive

Our rule

A set of possible *weight update rules*

- We need to specify what will happen to a weight in four different situations:

$\Delta w_{m_i s_j} = +1$ if both m_i and s_j are active

$\Delta w_{m_i s_j} = -1$ if m_i is active and s_j is inactive

$\Delta w_{m_i s_j} = -1$ if m_i is inactive and s_j is active

$\Delta w_{m_i s_j} = 0$ if both m_i and s_j are inactive

Another rule

A set of possible *weight update rules*

- We need to specify what will happen to a weight in four different situations:

$\Delta w_{m_i s_j} = +1$ if both m_i and s_j are active
 $\Delta w_{m_i s_j} = 0$ if m_i is active and s_j is inactive
 $\Delta w_{m_i s_j} = 0$ if m_i is inactive and s_j is active
 $\Delta w_{m_i s_j} = -1$ if both m_i and s_j are inactive

Yet another rule

A set of possible *weight update rules*

- We need to specify what will happen to a weight in four different situations:

$\Delta w_{m_i s_j} = \alpha$ if both m_i and s_j are active

$\Delta w_{m_i s_j} = \beta$ if m_i is active and s_j is inactive

$\Delta w_{m_i s_j} = \gamma$ if m_i is inactive and s_j is active

$\Delta w_{m_i s_j} = \delta$ if both m_i and s_j are inactive

General specification of rules: $[\alpha, \beta, \gamma, \delta]$

Investigation into weight update rules

- If we limit ourselves to +1, 0 or 1 for each weight update, then there are 81 different possible rules
- For each of these weight update rules (or at least some of them!) we want to ask:
 - How well does it recreate the training data for certain important types of language (e.g. the optimal language, or a maximally ambiguous language)?
 - How well does it generalise to unseen data for each of these languages?
 - How well will a pair of agents with the rule communicate after being trained on these languages?

Bias and innateness

- Each of these 81 rules may model a different *learning bias*
- What do they correspond to in reality?
 - They are a feature that an agent is born with that changes the learnability of different kinds of languages. A different kind of innateness.
- What are the consequences for language of this kind of innateness?
 - For the animal model, there's a simple relationship between genes and behaviour (i.e. signalling)
 - For the learning model, the relationship between genes and behaviour (i.e. language) is much more complex