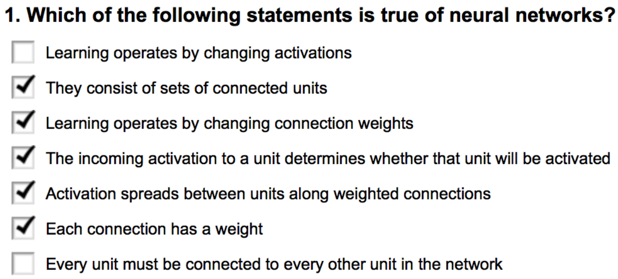
Comments on pre-reading quiz 5



1. Which of the following statements is true of neural networks?

☐ Learning operates by changing activations

☑ They consist of sets of connected units

☑ Learning operates by changing connection weights

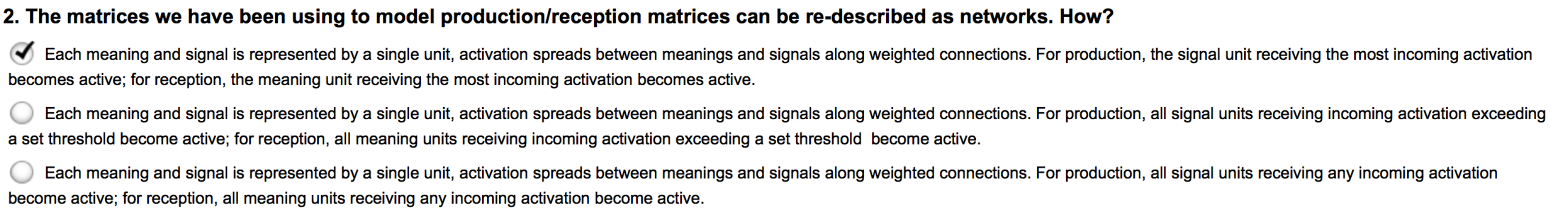
☑ The incoming activation to a unit determines whether that unit will be activated

☑ Activation spreads between units along weighted connections

☑ Each connection has a weight

☐ Every unit must be connected to every other unit in the network

Bear in mind that the order of these options might have been different for you. I haven't selected the first option here because learning is about changing **weights**, not activations. I haven't selected the last because you can have partially-connected networks - for instance, in multi-layer networks, units are only connected to units in the next layer; in our simple little networks, all the meanings are connected to all the signals, but not to all the other meanings. So partially-connected networks are pretty standard.



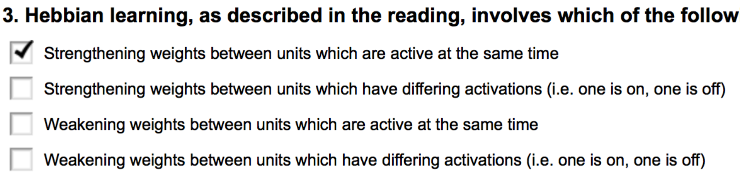
2. The matrices we have been using to model production/reception matrices can be re-described as networks. How?

● Each meaning and signal is represented by a single unit, activation spreads between meanings and signals along weighted connections. For production, the signal unit receiving the most incoming activation becomes active; for reception, the meaning unit receiving the most incoming activation becomes active.

○ Each meaning and signal is represented by a single unit, activation spreads between meanings and signals along weighted connections. For production, all signal units receiving incoming activation exceeding a set threshold become active; for reception, all meaning units receiving incoming activation exceeding a set threshold become active.

○ Each meaning and signal is represented by a single unit, activation spreads between meanings and signals along weighted connections. For production, all signal units receiving any incoming activation become active; for reception, all meaning units receiving any incoming activation become active.

The difference between these three possibilities lies in how we select which unit to activate, based on the incoming activations. We are using a very simple rule: the unit with the highest incoming activation is switched on, and all others stay off. The other options listed in this question (a set threshold, or a set threshold of 0, i.e. any activation will do) are used in other networks, but we are using this winner-take-all style of rule.



3. Hebbian learning, as described in the reading, involves which of the following operations?

☑ Strengthening weights between units which are active at the same time

☐ Strengthening weights between units which have differing activations (i.e. one is on, one is off)

☐ Weakening weights between units which are active at the same time

☐ Weakening weights between units which have differing activations (i.e. one is on, one is off)

Hopefully an easy one - all the other options are things you might **consider** doing (and decreasing connection weights between units that have mismatched activations is a good idea), but classic Hebbian learning just involves strengthening connection weights between co-active units.

4. [Some of you have trouble with images in these surveys, so I am writing this question without them! I am going to write the weights out as a grid: the first two shows the weights from unit m1 to signals s1 and s2, the second row shows the weights from m2 to signals s1 and s2]
Imagine a network representing an agent with two meanings and two signals. Initially, all connection weights are 0, i.e. it looks like this:

0 0
0 0

This learner makes four observations of the signalling behaviour of some other individual: meaning 1 is conveyed using signal 2; meaning 2 is conveyed using signal 1; meaning 1 is conveyed using signal 1; meaning 1 is conveyed using signal 2. Assuming the learner is applying Hebbian learning, what will the connection weights in the network be after learning?

○
2 1
0 1

○
0 0
0 0

○
-2 1
0 3

☑
1 2
1 0

If you found this at all confusing, the best way to work this out is with a pencil and paper: draw out the network, indicate the active units, apply Hebb’s rule, update the weights, and repeat. If you are looking for a shortcut: Hebbian learning involves strengthening association strengths between co-active units, meanings and signals are co-active when they are observed as a pair by the learner, so with simple Hebbian learning the network ends up being a simple co-occurence count of observed meaning-signal pairs. The learner say m1-s1 once, m1-s2 twice, m2-s1 once, m2-s2 zero times, so those are the appropriate weights.

These are, in order, Simulation 1, 2, 3 and 4 - the only condition where communication fails to evolve is in Simulation 2, which is receiver-only payoff and no additions (reciprocity, kin selection) to compensate.

The problem with the receiver-only condition is that there is no selection acting on the population's send behaviour, so it fluctuates randomly.

Oliphant encodes an individual's send and receive behaviours as entirely separate (which is what we are also doing at the moment, with our separate send and receive matrices), so you could have an individual who sends signal a for meaning 1 but interprets signal a as conveying meaning 2, for instance. In Oliphant's Simulation 2, the only thing which influences an individual's fitness (reproductive chances) is their reception behaviour - production behaviour is irrelevant for determining who reproduces and who doesn't. Consequently, production behaviour doesn't evolve by natural selection - there is heritable variation in production behaviour, but it doesn't impact on fitness at all. The population's production behaviour therefore just changes randomly - one production system will increase in numbers for a while, since it happens to live in individuals who are good receivers, but then mutation will introduce different variants and the numbers of the various possible production systems will fluctuate unpredictably. As a result, communication is never stable - the population is always evolving to be able to understand the current most common production system, but that production system is always changing (and probably changing fast, given how high Oliphant sets his mutation rate).

We will spend some time discussing this result in the lecture, so if this explanation doesn't make sense, come equipped with questions!