From Perceptrons to Neural Networks

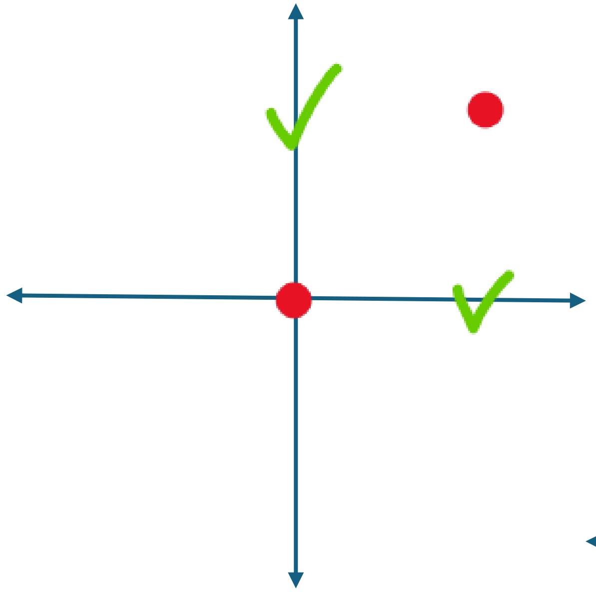
INF1-CG Week 3

A single perceptron is powerful!

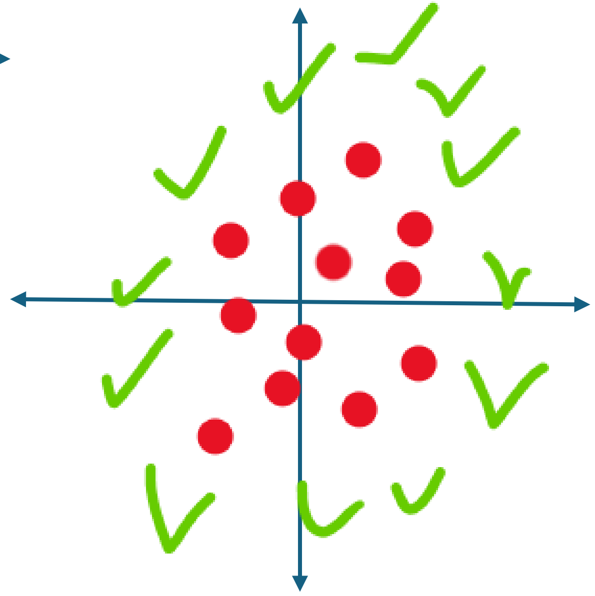
- Can represent <u>any</u> linear decision boundary in n-dimensions
- Using the perceptron learning rule, weights can be <u>automatically</u> determined to match <u>any</u> dataset that is linearly separable
  - In other words, if it is possible to divide the data, the perceptron learning rule is guaranteed to find a solution!
- A perceptron can learn to represent the basic logic gates: AND, OR, NAND, NOR

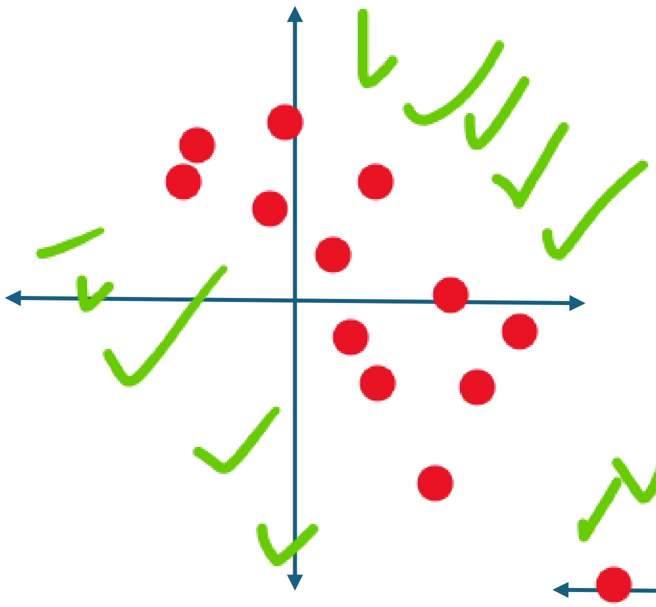
…but not powerful enough

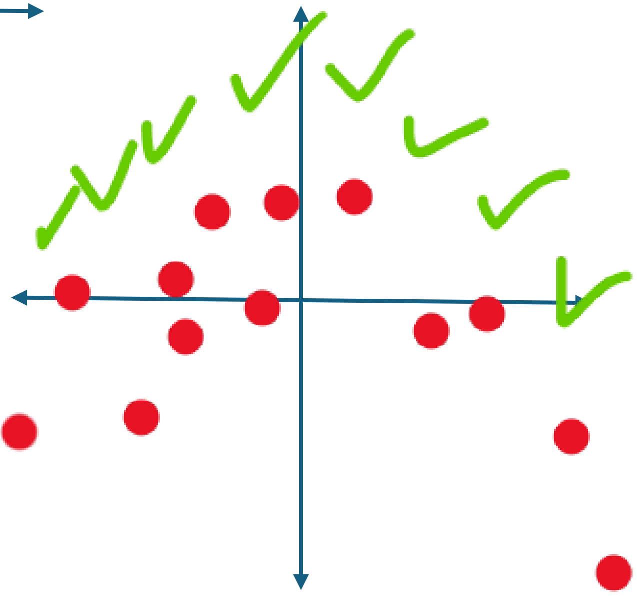
- Lots of datasets are <u>not</u> linearly separable



Famous counter-example: XOR function

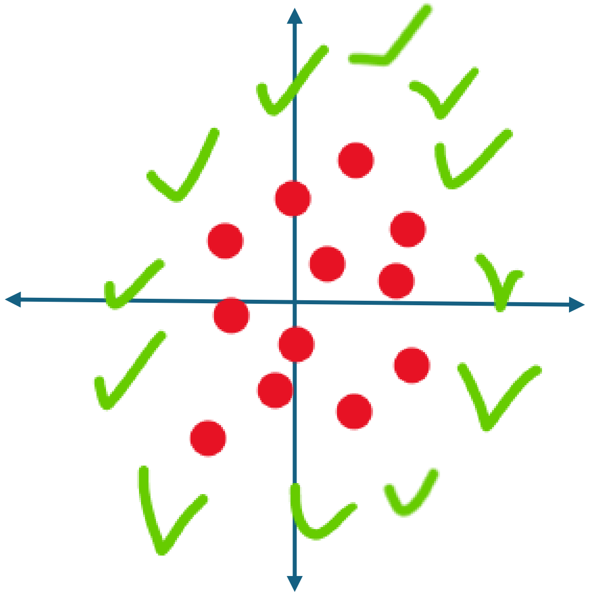


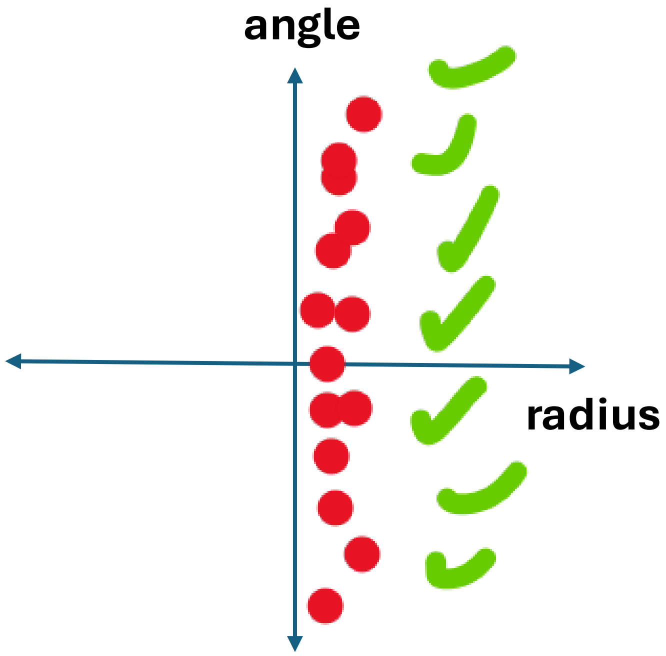




So what to do???

- One idea: Can we change input space to make the data linearly separable?
  - For example, what if we plot this using polar coordinates?





- This is cool, and a lot of machine learning does this!
- But it's also tricky, and often data-dependent. We want a more general-purpose solution

Different solution…

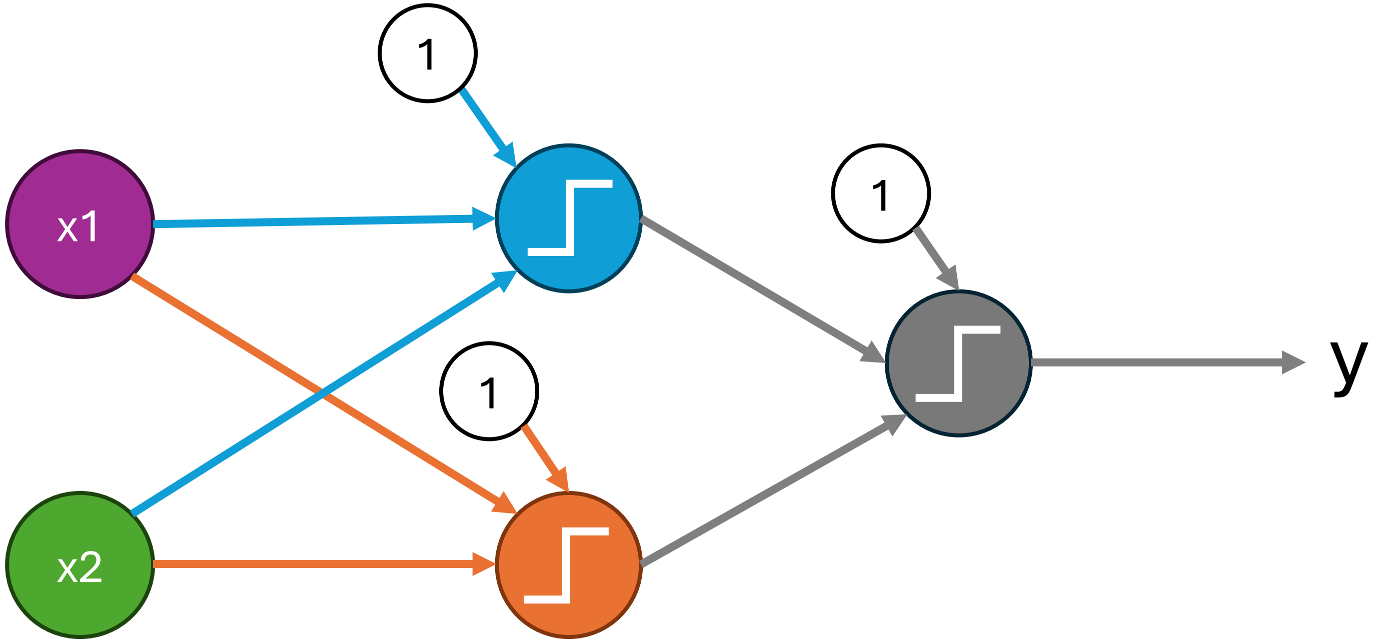
- Use <u>multiple</u> perceptrons!

- A neural network is just multiple ”neurons” put together into a giant collection

- Each “neuron” is very similar to a perceptron, but with some added features that we will talk about next week!

Small puzzle problem for you

- Remember, we cannot represent XOR with a single perceptron
- But… we can with this set of three perceptrons! Can you find (by hand) a set of weights that works?



- Hint: there are 9 weights
- Hint #2: Try breaking the problem down into small pieces…