

# A Simple Predicting Machine

Building a simple model to predict a missing value based on some past observations.

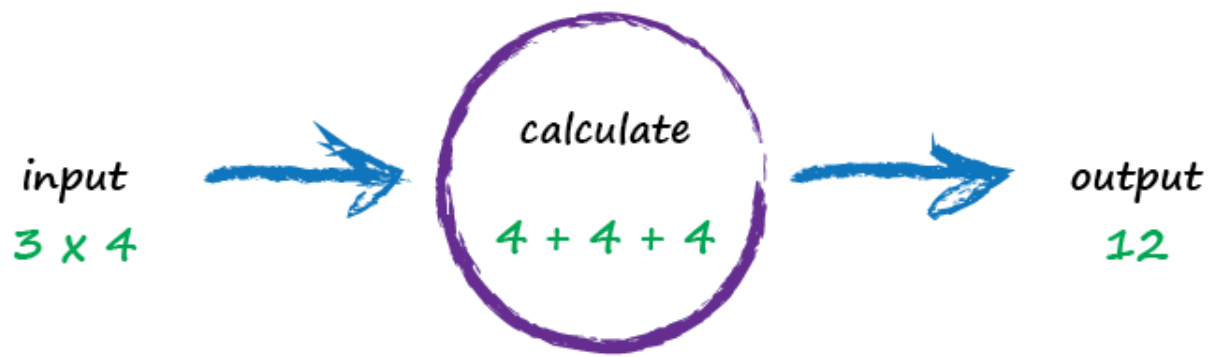
Let's start super simple and build up from there. Imagine a basic machine that takes a question, does some "thinking" and pushes out an answer. Just like the example above with ourselves taking input through our eyes, using our brains to analyse the scene, and coming to the conclusion about what objects are in that scene. Here's what this looks like:



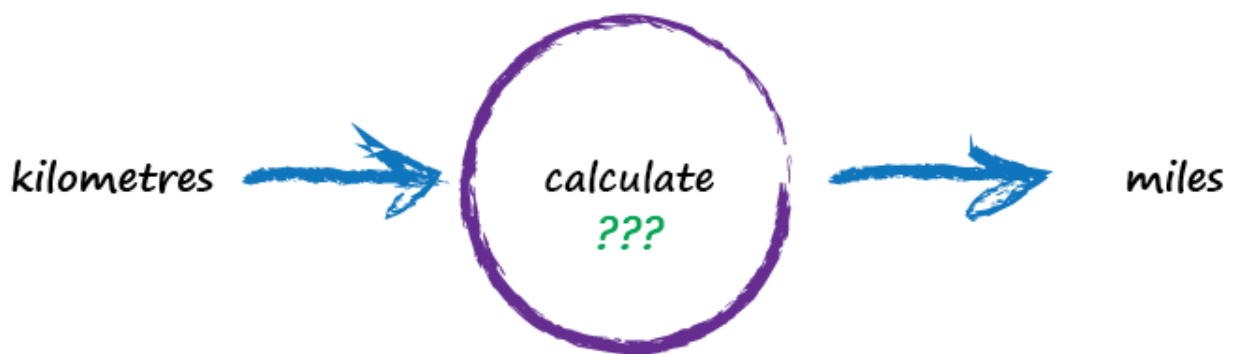
Computers don't really think, they are just glorified calculators remember, so let's use more appropriate words to describe what's going on:



A computer takes some input, does some calculation and pops out an output. The following illustrates this. An input of "3 x 4" is processed, perhaps by turning multiplication into an easier set of additions, and the output answer 12 pops out.



“That’s not so impressive!” you might be thinking. That’s ok. We’re using simple and familiar examples here to set out concepts which will apply to the more interesting neural networks we look at later. Let’s ramp up the complexity just a tiny notch. Imagine a machine that converts kilometers to miles, like the following:



Now imagine we don’t know the formula for converting between kilometers and miles. All we know is the relationship between the two is *linear*. That means if we double the number of miles, the same distance in kilometers is also doubled. That makes intuitive sense. The universe would be a strange place if that weren’t true!

This linear relationship between kilometers and miles gives us a clue about that mysterious calculation - it needs to be of the form  $miles = kilometers * c$ , where  $c$  is a constant. We don’t know what this constant  $c$  is yet.

The only other clues we have are some examples pairing kilometers with the correct value for miles. These are like real-world observations used to test scientific theories - they’re examples of real-world truth.

Truth Example	Kilometres	Miles
1	0	0
2	100	62.137

So how do we estimate the missing constant “c” by using these two examples that are given in the table? Let’s find it out in the next lesson!