Quick Recap

Up to this point you have learned the Machine Learning paradigm, and how it works to allow a computer to figure out the parameters that fit a function. When that function gives you the rules to map data to answers, you've changed from a programming paradigm where *you* have to figure out the rules that act on the data to give you answers:

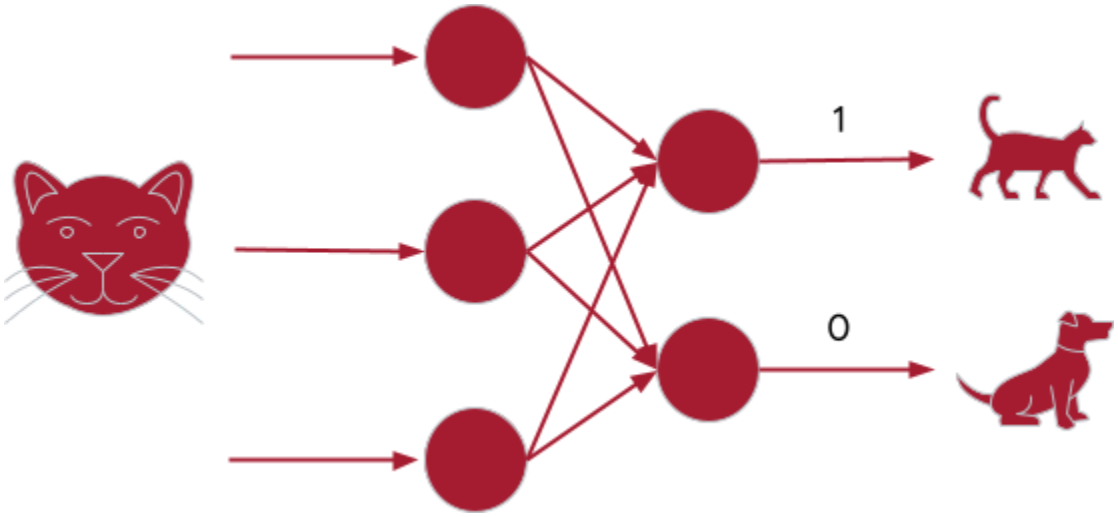
To one that looks like this, where you provide the answers with the data, and the computer figures out the rules that maps them to each other:

To achieve this, of course, you still need to write code, but instead of writing the code for the rules, you write the code that figures out what the rules are. The algorithm to do this looks like the following:

As an example, you used two sets of numbers, for X and Y and wrote code that used the above algorithm to figure out the relationship between them where Y = wX+b, and the above algorithm could figure out the values of w and b that would fit.

This was the basis of a mathematical neuron, where the neuron could store the values of w and b, and again, using the above algorithm, those values could be figured out to give the desired Y for an input X.

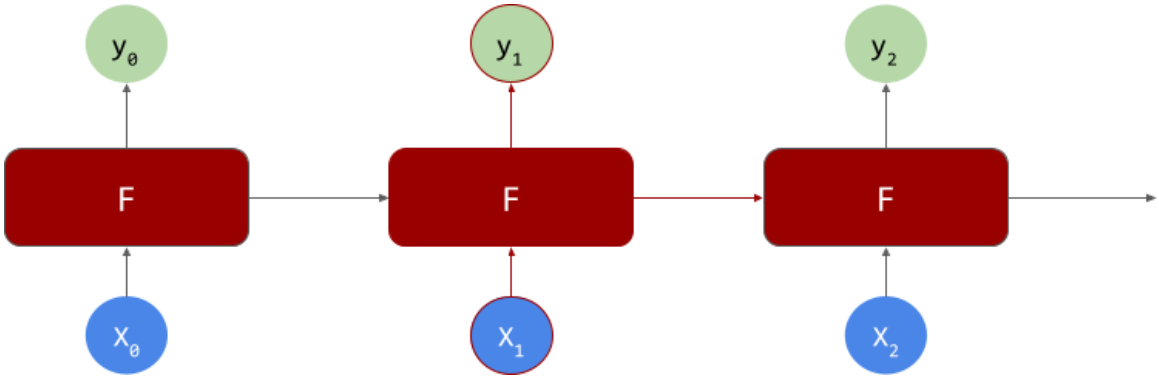
When these neurons are used together, in layers, more complex relationships could be figured out, and you saw how having multiple output neurons could take you into the territory of *classification*, where, each neuron could have a role in the final output, which of course is perfect for computer vision. Below is a simplified neural network diagram illustrating this point where input data can be classified -- i.e. does the network 'see' a dog or a cat.



Basic neurons, stacked together like this form what is called a *Dense* layer, based on the fact that they're densely connected to each other. When there are multiple layers between the input and the output, you have a *Deep* neural network, which gives us the term *Deep Learning.* Additionally, the layers between the input and the output are often called *hidden* layers.

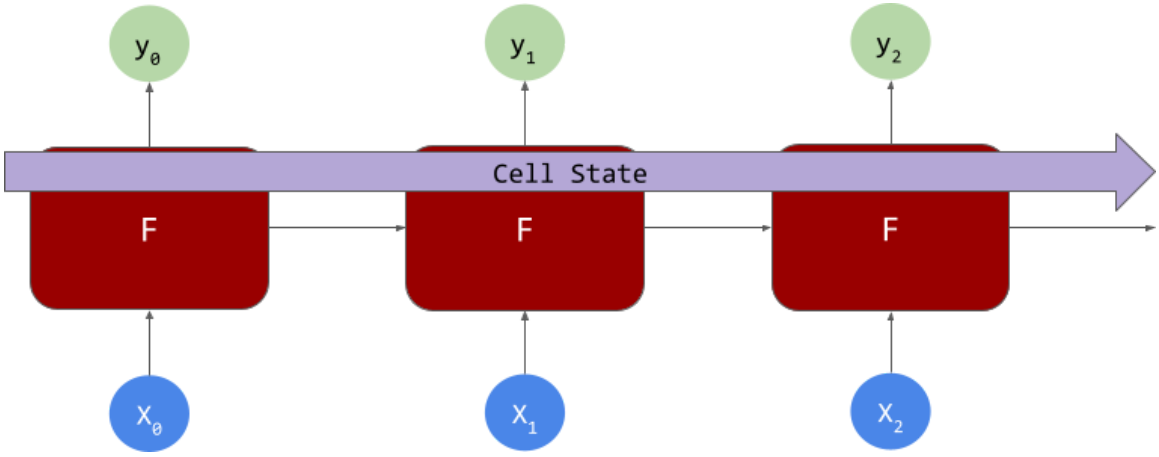
As you continue your Machine Learning journey, you'll discover that there are many other types of layer that learn different parameters to just the w and b that a neuron learns.

For example, there are *Recurrent* layers, that are beyond the scope of this course, but they are able to learn values in a sequence where each neuron learns values *and* passes those values to another neuron in the same layer. The basic idea can be represented like this:



Where, if we have a sequence of values, X0, X1, X2 that we want to learn a corresponding sequence Y0, Y1, Y2 for, then the idea is the the neurons (F) can not only try to match X0->Y0, but also pass values along the sequence, indicated by the right pointing arrows, so the input to the second neuron is the output from the first *and* X1, from which it can learn the parameters for Y1 and so on.

There are many variations on this idea, but a very powerful one is called *Long Short Term Memory* where not only are values passed from neuron to neuron, so that X0 can impact Y1, X1 can impact Y2, but values from further back can also have an impact -- so that X0 could impact Y99 for example. This is achieved using a data structure called a *Cell State* where context can be preserved across multiple neurons.



Another type of neural network layer is called a *Convolution*, where a filter that can transform data, particularly image data, can be learned. You'll explore those next!

I hope this helps you understand that Machine Learning goes beyond simple neurons that learn a w and b in the Y=wX+b scenario.