~~It's not too hard to imagine what the model might look like for a checkers program~~. There might be a range of checkers strategies encoded, and some kind of search mechanism, and then the weights could vary how strategies are selected, what parts of the board are focused on during a search, and so forth.

- ~~But it's not at all obvious what the model might look like for an image recognition program.~~

What we need is some kind of function that is so flexible, that it could be used to solve any given problem, just by varying its weights. Amazingly enough, this function actually exists! It’s called the neural network. A mathematical proof called the universal approximation theorem shows that this function can solve any problem to any level of accuracy.

~~In addition, there is a completely general way to update the weights of a neural network, to make it improve at any given task.~~ This is called stochastic gradient descent (SGD). We'll see how neural networks and SGD work in detail later in this book, as well as explaining the universal approximation theorem. For now, however, we will instead use Samuel's own words: We need not go into the details of such a procedure to see that it could be made entirely automatic and to see that a machine so programed would "learn" from its experience.