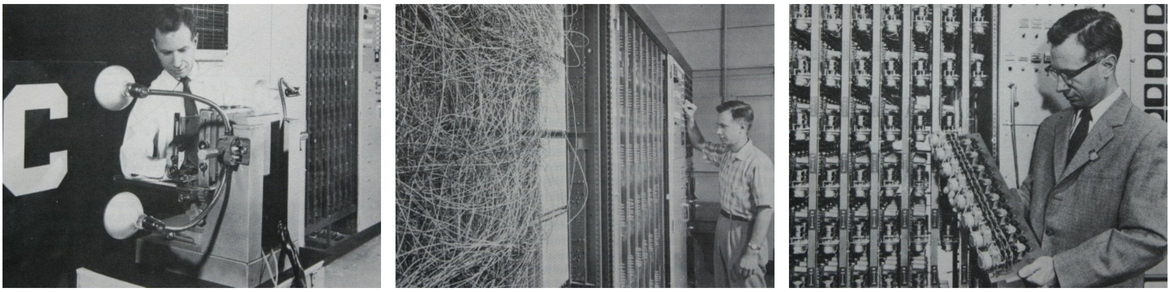

Figure 1.14 Illustration of the Mark 1 perceptron hardware. The photograph on the left shows how the inputs were obtained using a simple camera system in which an input scene, in this case a printed character, was illuminated by powerful lights, and an image focused onto a 20×20 array of cadmium sulphide photocells, giving a primitive 400-pixel image. The perceptron also had a patch board, shown in the middle photograph, which allowed different configurations of input features to be tried. Often these were wired up at random to demonstrate the ability of the perceptron to learn without the need for precise wiring, in contrast to a modern digital computer. The photograph on the right shows one of the racks of learnable weights. Each weight was implemented using a rotary variable resistor, also called a potentiometer, driven by an electric motor thereby allowing the value of the weight to be adjusted automatically by the learning algorithm.

implementation, as shown in Figure 1.14. A typical perceptron configuration had multiple layers of processing, but only one of those layers was learnable from data, and so the perceptron is considered to be a 'single-layer' neural network.

At first, the ability of perceptrons to learn from data in a brain-like way was considered remarkable. However, it became apparent that the model also has major limitations. The properties of perceptrons were analysed by Minsky and Papert (1969), who gave formal proofs of the limited capabilities of single-layer networks. Unfortunately, they also speculated that similar limitations would extend to networks having multiple layers of learnable parameters. Although this latter conjecture proved to be wildly incorrect, the effect was to dampen enthusiasm for neural network models, and this contributed to the lack of interest, and funding, for neural networks during the 1970s and early 1980s. Furthermore, researchers were unable to explore the properties of multilayered networks due to the lack of an effective algorithm for training them, since techniques such as the perceptron algorithm were specific to single-layer models. Note that although perceptrons have long disappeared from practical machine learning, the name lives on because a modern neural network is also sometimes called a *multilayer perceptron* or *MLP*.

1.3.2 Backpropagation

The solution to the problem of training neural networks having more than one layer of learnable parameters came from the use of differential calculus and the application of gradient-based optimization methods. An important change was to replace the step function (1.7) with continuous differentiable activation functions having a non-zero gradient. Another key modification was to introduce differentiable error functions that define how well a given choice of parameter values predicts the target variables in the training set. We saw an example of such an error function when we