Many artificial intelligence tasks can be solved by designing the right set of features to extract for that task, then providing these features to a simple machine learning algorithm. For example, ... a strong clue as to whether the speaker is a man, woman, or child.

However, for many tasks, it is difficult to know what features should be extracted. For example, ... Unfortunately, it is difficult to describe exactly what a wheel looks like in terms of pixel values. ... or an object in the foreground obscuring part of the wheel, and so on.

Suppose we want to separate two categories of data by drawing a line between them in a scatterplot.

One solution to this problem is to use machine learning to discover not only the mapping from representation to output but also the representation itself. This approach is known as representation learning. Learned representations often result in much better performance than can be obtained with hand-designed representations. They also allow AI systems to rapidly adapt to new tasks, with minimal human intervention. A representation learning algorithm can discover a good set of features for a simple task in minutes, or a complex task in hours to months. Manually designing features for a complex task requires a great deal of human time and effort; it can take decades for an entire community of researchers.

The quintessential example of a representation learning algorithm is ... Different kinds of autoencoders aim to achieve different kinds of properties.

Deep learning addresses this central problem in representation learning by introducing representations that are expressed in terms of simpler, more fundamental representations. Deep learning enables the computer to construct complex concepts from simpler ones. Figure 1.2 illustrates how a deep learning system can represent the concept of an image of a person by combining simpler concepts, such as corners and contours, which are defined in terms of edges.

It is challenging for a computer to comprehend the significance of unprocessed sensory input data, such as an image depicted as a set of pixel values. The function that maps a collection of pixels to an object identity is highly intricate. Attempting to learn or assess this mapping directly appears to be an overwhelming task. Deep learning addresses this challenge by deconstructing the complex desired mapping into a sequence of simpler nested mappings, each represented by a distinct layer of the model. The input is introduced at the visible layer, aptly named because it encompasses the variables that we can observe. Subsequently, a succession of hidden layers extracts progressively abstract features from the image. These layers are termed 'hidden' as their values are not provided in the data; rather, the model must ascertain which concepts are pertinent for elucidating the relationships within the observed data. The images presented here are visual representations of the types of features associated with each hidden unit. Given the pixel data, the first layer can readily identify edges by comparing the brightness of adjacent pixels. Utilizing the first hidden layer's depiction of edges, the second hidden layer can efficiently search for corners and extended contours, which are identifiable as groups of edges. Based on the second hidden layer's interpretation of the image in terms of corners and contours, the third hidden layer can recognize entire segments of specific objects by locating particular collections of contours and corners. Ultimately, this characterization of the image in relation to the object parts it encompasses can facilitate the recognition of the objects present within the image.

Depth is defined as the longest path's length from input to output. The computation depicted in these graphs is the output of a logistic regression model, $\sigma(w^T x)$, where σ is the logistic sigmoid function. If we use addition, multiplication and logistic sigmoids as the elements of our computer language, then this model has depth three. If we view logistic regression as an element itself, then this model has depth one.

Two main ways of measuring the depth of a model are
1) using the number of sequential instructions.
2) depth of the graph saying how concepts are linked to each other
Because it is not always clear which of these two views—the depth of the computational graph, or the depth of the probabilistic modeling graph—is most relevant, and because different people choose different sets of smallest elements from which to construct their graphs, there is no single correct value for the depth of an architecture, just as there is no single correct value for the length of a computer program. Nor is there a consensus about how much depth a model requires to qualify as “deep.” However, deep learning can safely be regarded as the study of models that either involve a greater amount of composition of learned functions or learned concepts than traditional machine learning does.

1.1 Who Should Read This Book?

This book can be useful for a variety of readers, but we wrote it with two main target audiences in mind. One of these target audiences is university students (undergraduate or graduate) learning about machine learning, including those who are beginning a career in deep learning and artificial intelligence research. The other target audience is software engineers who do not have a machine learning
or statistics background, but want to rapidly acquire one and begin using deep learning in their product or platform.

The training algorithm used to adapt the weights of the ADALINE was a special case of ...?

In the 1980s, the second wave of neural network research emerged in great part via a movement called ...?

... DeepMind demonstrated that a reinforcement learning system ...