There are three types of **feedback** that can accompany the inputs, and that determine the three main types of learning:

Feedback

- In **supervised learning** the agent observes input-output pairs and learns a function that maps from input to output. For example, the inputs could be camera images, each one accompanied by an output saying "bus" or "pedestrian," etc. An output like this is called a **label**. The agent learns a function that, when given a new image, predicts the appropriate label. In the case of braking actions (component 1 above), an input is the current state (speed and direction of the car, road condition), and an output is the distance it took to stop. In this case a set of output values can be obtained by the agent from its own percepts (after the fact); the environment is the teacher, and the agent learns a function that maps states to stopping distance.

Supervised learning

Label

- In **unsupervised learning** the agent learns patterns in the input without any explicit feedback. The most common unsupervised learning task is **clustering**: detecting potentially useful clusters of input examples. For example, when shown millions of images taken from the Internet, a computer vision system can identify a large cluster of similar images which an English speaker would call "cats."

Unsupervised learning

- In **reinforcement learning** the agent learns from a series of reinforcements: rewards and punishments. For example, at the end of a chess game the agent is told that it has won (a reward) or lost (a punishment). It is up to the agent to decide which of the actions prior to the reinforcement were most responsible for it, and to alter its actions to aim towards more rewards in the future.

Reinforcement learning

19.2 Supervised Learning

More formally, the task of supervised learning is this:

Given a **training set** of N example input–output pairs

Training set

$$(x_1, y_1), (x_2, y_2), \ldots (x_N, y_N),$$

where each pair was generated by an unknown function $y = f(x)$,
discover a function h that approximates the true function f.

The function h is called a **hypothesis** about the world. It is drawn from a **hypothesis space** $\mathcal{H}$ of possible functions. For example, the hypothesis space might be the set of polynomials of degree 3; or the set of Javascript functions; or the set of 3-SAT Boolean logic formulas.

Hypothesis space

With alternative vocabulary, we can say that h is a **model** of the data, drawn from a **model class** $\mathcal{H}$, or we can say a **function** drawn from a **function class**. We call the output y_i the **ground truth**—the true answer we are asking our model to predict.

Model class

Ground truth

How do we choose a hypothesis space? We might have some prior knowledge about the process that generated the data. If not, we can perform **exploratory data analysis**: examining the data with statistical tests and visualizations—histograms, scatter plots, box plots—to get a feel for the data, and some insight into what hypothesis space might be appropriate. Or we can just try multiple hypothesis spaces and evaluate which one works best.

Exploratory data analysis

How do we choose a good hypothesis from within the hypothesis space? We could hope for a **consistent hypothesis**: an h such that each x_i in the training set has $h(x_i) = y_i$. With continuous-valued outputs we can't expect an exact match to the ground truth; instead we

Consistent hypothesis