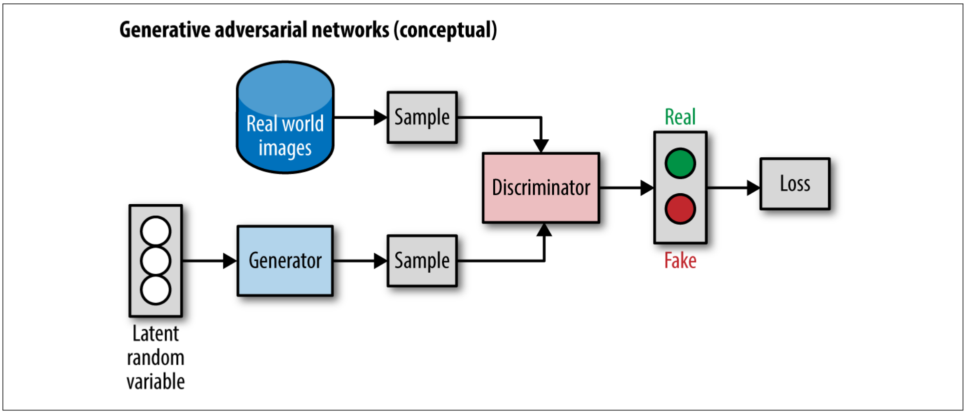

Figure 1-13. A conceptual depiction of a generative adversarial network (GAN).

GANs have proven capable of generating very realistic images, and will likely power the next generation of computer graphics tools. Samples from such systems are now approaching photorealism. However, many theoretical and practical caveats still remain to be worked out with these systems and much research is still needed.

Neural Turing Machines

Most of the deep learning systems presented so far have learned complex functions with limited domains of applicability; for example, object detection, image captioning, machine translation, or Go game-play. But, could we perhaps have deep architectures that learn general algorithmic concepts such as sorting, addition, or multiplication?

The Neural Turing machine (NTM) is a first attempt at making a deep learning architecture capable of learning arbitrary algorithms. This architecture adds an external memory bank to an LSTM-like system, to allow the deep architecture to make use of scratch space to compute more sophisticated functions. At the moment, NTM-like architectures are still quite limited, and only capable of learning simple algorithms. Nevertheless, NTM methods remain an active area of research and future advances may transform these early demonstrations into practical learning tools. The NTM architecture is conceptually illustrated in Figure 1-14.

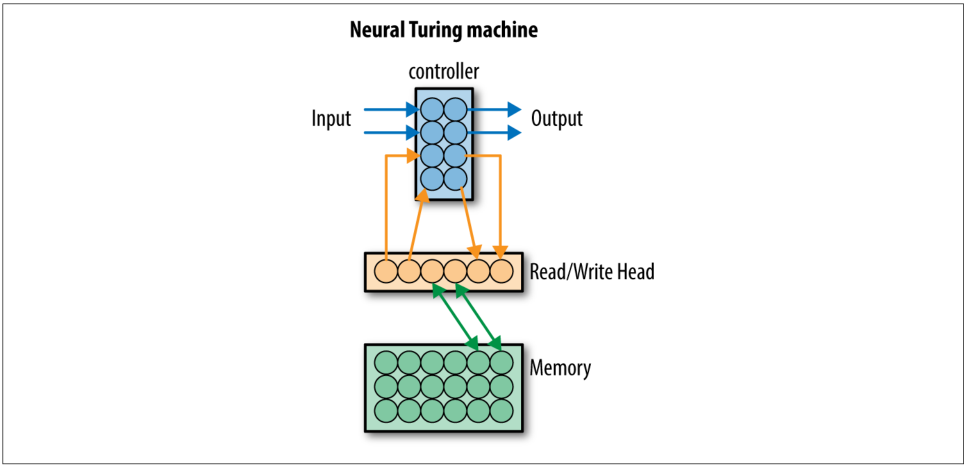

Figure 1-14. A conceptual depiction of a Neural Turing machine. It adds an external memory bank to which the deep architecture reads and writes.

Deep Learning Frameworks

Researchers have been implementing software packages to facilitate the construction of neural network (deep learning) architectures for decades. Until the last few years, these systems were mostly special purpose and only used within an academic group. This lack of standardized, industrial-strength software made it difficult for non-experts to use neural networks extensively.

This situation has changed dramatically over the last few years. Google implemented the DistBelief system in 2012 and made use of it to construct and deploy many simpler deep learning architectures. The advent of DistBelief, and similar packages such as Caffe, Theano, Torch, Keras, MxNet, and so on have widely spurred industry adoption.

TensorFlow draws upon this rich intellectual history, and builds upon some of these packages (Theano in particular) for design principles. TensorFlow (and Theano) in particular use the concept of tensors as the fundamental underlying primitive powering deep learning systems. This focus on tensors distinguishes these packages from systems such as DistBelief or Caffe, which don't allow the same flexibility for building sophisticated models.

While the rest of this book will focus on TensorFlow, understanding the underlying principles should enable you to take the lessons learned and apply them with little difficulty to alternative deep learning frameworks.