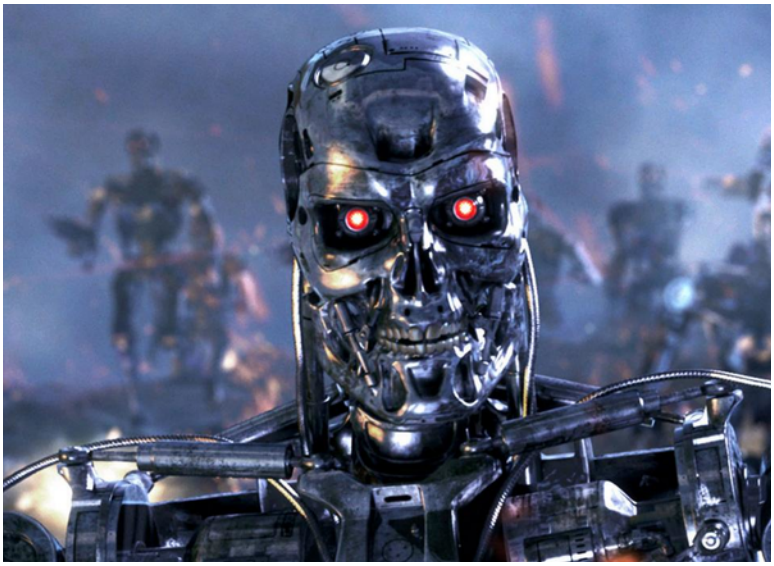

Figure 4-4. Unfortunately (or perhaps fortunately), this book won't teach you to build a Terminator!

AI Winters

Artificial intelligence has gone through multiple rounds of boom-and-bust development. This cyclical development is characteristic of the field. Each new advance in learning spawns a wave of optimism in which prophets claim that human-level (or superhuman) intelligences are incipient. After a few years, no such intelligences manifest, and disappointed funders pull out. The resulting period is called an AI winter.

There have been multiple AI winters so far. As a thought exercise, we encourage you to consider when the next AI winter will happen. The current wave of deep learning progress has solved many more practical problems than any previous wave of advances. Is it possible AI has finally taken off and exited the boom-and-bust cycle or do you think we're in for the "Great Depression" of AI soon?

Learning Fully Connected Networks with Backpropagation

The first version of a fully connected neural network was the Perceptron, (Figure 4-5), created by Frank Rosenblatt in the 1950s. These perceptrons are identical to the "neurons" we introduced in the previous equations.

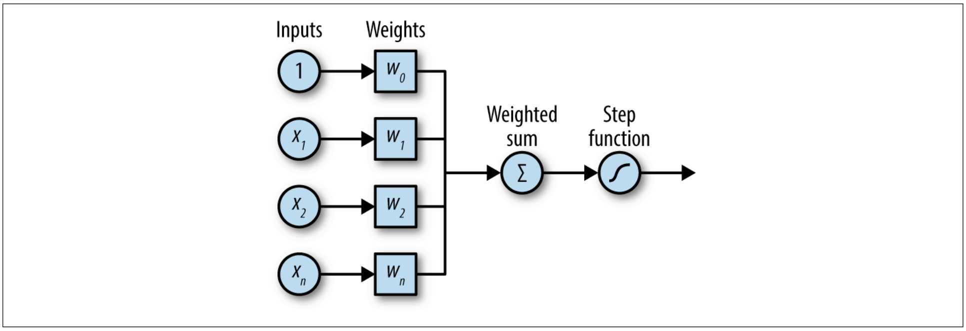

Figure 4-5. A diagrammatic representation of the perceptron.

Perceptrons were trained by a custom "perceptron" rule. While they were moderately useful solving simple problems, perceptrons were fundamentally limited. The book *Perceptrons* by Marvin Minsky and Seymour Papert from the end of the 1960s proved that simple perceptrons were incapable of learning the XOR function. Figure 4-6 illustrates the proof of this statement.

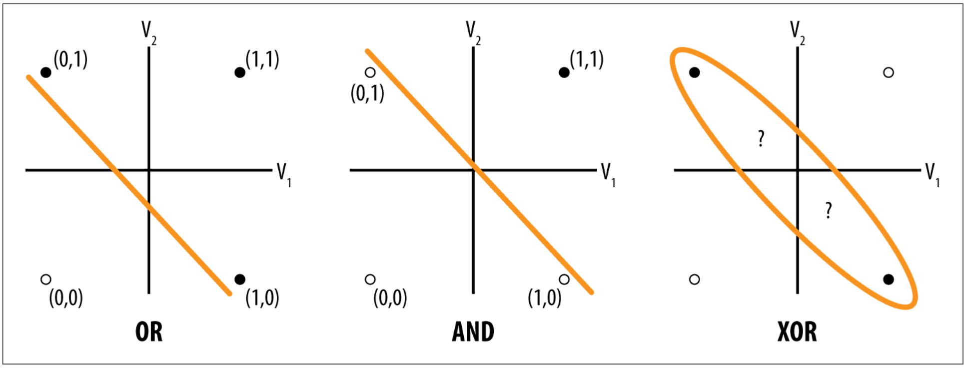

Figure 4-6. The perceptron's linear rule can't learn the perceptron.

This problem was overcome with the invention of the multilayer perceptron (another name for a deep fully connected network). This invention was a formidable achievement, since earlier simple learning algorithms couldn't learn deep networks effectively. The "credit assignment" problem stumped them; how does an algorithm decide which neuron learns what?

The full solution to this problem requires backpropagation. Backpropagation is a generalized rule for learning the weights of neural networks. Unfortunately, complicated explanations of backpropagation are epidemic in the literature. This situation is unfortunate since backpropagation is simply another word for automatic differentiation.

Let's suppose that $f(\theta, x)$ is a function that represents a deep fully connected network. Here x is the inputs to the fully connected network and θ is the learnable weights. Then the backpropagation algorithm simply computes $\frac{\partial f}{\partial \theta}$. The practical complexities arise in implementing backpropagation for all possible functions f that arise in practice. Luckily for us, TensorFlow takes care of this already!

Universal Convergence Theorem

The preceding discussion has touched on the ideas that deep fully connected networks are powerful approximations. McCulloch and Pitts showed that logical networks can code (almost) any Boolean function. Rosenblatt's Perceptron was the continuous analog of McCulloch and Pitt's logical functions, but was shown to be fundamentally limited by Minsky and Papert. Multilayer perceptrons looked to solve the limitations of simple perceptrons and empirically seemed capable of learning complex functions. However, it wasn't theoretically clear whether this empirical ability had undiscovered limitations. In 1989, George Cybenko demonstrated that multilayer perceptrons were capable of representing arbitrary functions. This demonstration provided a considerable boost to the claims of generality for fully connected networks as a learning architecture, partially explaining their continued popularity.

However, if both backpropagation and fully connected network theory were understood in the late 1980s, why didn't "deep" learning become more popular earlier? A large part of this failure was due to computational limitations; learning fully connected networks took an exorbitant amount of computing power. In addition, deep networks were very difficult to train due to lack of understanding about good hyperparameters. As a result, alternative learning algorithms such as SVMs that had lower computational requirements became more popular. The recent surge in popularity in deep learning is partly due to the increased availability of better computing hardware that enables faster computing, and partly due to increased understanding of good training regimens that enable stable learning.

Is Universal Approximation That Surprising?

Universal approximation properties are more common in mathematics than one might expect. For example, the Stone-Weierstrass theorem proves that any continuous function on a closed interval can be a suitable polynomial function. Loosening our criteria further, Taylor series and Fourier series themselves offer some universal approximation capabilities (within their domains of convergence). The fact that universal convergence is fairly common in mathematics provides partial justification for the empirical observation that there are many slight variants of fully connected networks that seem to share a universal approximation property.