Training Fully Connected Neural Networks

As we mentioned previously, the theory of fully connected networks falls short of practice. In this section, we will introduce you to a number of empirical observations about fully connected networks that aid practitioners. We strongly encourage you to use our code (introduced later in the chapter) to check our claims for yourself.

Learnable Representations

One way of thinking about fully connected networks is that each fully connected layer effects a transformation of the feature space in which the problem resides. The idea of transforming the representation of a problem to render it more malleable is a very old one in engineering and physics. It follows that deep learning methods are sometimes called "representation learning." (An interesting factoid is that one of the major conferences for deep learning is called the "International Conference on Learning Representations.")

Generations of analysts have used Fourier transforms, Legendre transforms, Laplace transforms, and so on in order to simplify complicated equations and functions to forms more suitable for handwritten analysis. One way of thinking about deep learning networks is that they effect a data-driven transform suited to the problem at hand.

The ability to perform problem-specific transformations can be immensely powerful. Standard transformation techniques couldn't solve problems of image or speech analysis, while deep networks are capable of solving these problems with relative ease due to the inherent flexibility of the learned representations. This flexibility comes with a price: the transformations learned by deep architectures tend to be much less general than mathematical transforms such as the Fourier transform. Nonetheless, having deep transforms in an analytic toolkit can be a powerful problem-solving tool.

There's a reasonable argument that deep learning is simply the first representation learning method that works. In the future, there may well be alternative representation learning methods that supplant deep learning methods.

Activations

We previously introduced the nonlinear function σ as the sigmoidal function. While the sigmoidal is the classical nonlinearity in fully connected networks, in recent years researchers have found that other activations, notably the rectified linear activation (commonly abbreviated ReLU or relu) $\sigma(x) = \max(x, 0)$ work better than the sigmoidal unit. This empirical observation may be due to the *vanishing gradient* problem in deep networks. For the sigmoidal function, the slope is zero for almost all values of its input. As a result, for deeper networks, the gradient would tend to zero. For the ReLU function, the slope is nonzero for a much greater part of input space, allowing non-

zero gradients to propagate. Figure 4-7 illustrates sigmoidal and ReLU activations side by side.

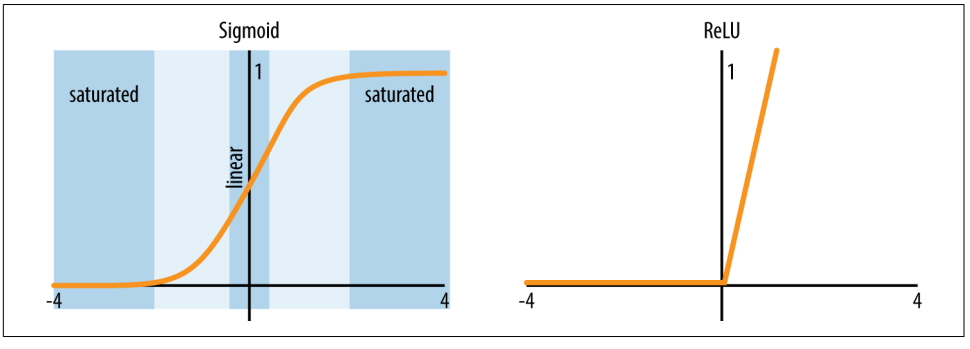

Figure 4-7. Sigmoidal and ReLU activation functions.

Fully Connected Networks Memorize

One of the striking aspects about fully connected networks is that they tend to memorize training data entirely given enough time. As a result, training a fully connected network to "convergence" isn't really a meaningful metric. The network will keep training and learning as long as the user is willing to wait.

For large enough networks, it is quite common for training loss to trend all the way to zero. This empirical observation is one the most practical demonstrations of the universal approximation capabilities of fully connected networks. Note however, that training loss trending to zero does not mean that the network has learned a more powerful model. It's rather likely that the model has started to memorize peculiarities of the training set that aren't applicable to any other datapoints.

It's worth digging into what we mean by peculiarities here. One of the interesting properties of high-dimensional statistics is that given a large enough dataset, there will be plenty of spurious correlations and patterns available for the picking. In practice, fully connected networks are entirely capable of finding and utilizing these spurious correlations. Controlling networks and preventing them from misbehaving in this fashion is critical for modeling success.

Regularization

Regularization is the general statistical term for a mathematical operation that limits memorization while promoting generalizable learning. There are many different types of regularization available, which we will cover in the next few sections.