

can cause similar difficulties for learning if the loss function is not designed to compensate for it.

The argument \mathbf{z} to the softmax function can be produced in two different ways. The most common is simply to have an earlier layer of the neural network output every element of \mathbf{z} , as described above using the linear layer $\mathbf{z} = \mathbf{W}^\top \mathbf{h} + \mathbf{b}$. While straightforward, this approach actually overparametrizes the distribution. The constraint that the n outputs must sum to 1 means that only $n - 1$ parameters are necessary; the probability of the n -th value may be obtained by subtracting the first $n - 1$ probabilities from 1. We can thus impose a requirement that one element of \mathbf{z} be fixed. For example, we can require that $z_n = 0$. Indeed, this is exactly what the sigmoid unit does. Defining $P(y = 1 \mid \mathbf{x}) = \sigma(z)$ is equivalent to defining $P(y = 1 \mid \mathbf{x}) = \text{softmax}(\mathbf{z})_1$ with a two-dimensional \mathbf{z} and $z_1 = 0$. Both the $n - 1$ argument and the n argument approaches to the softmax can describe the same set of probability distributions, but have different learning dynamics. In practice, there is rarely much difference between using the overparametrized version or the restricted version, and it is simpler to implement the overparametrized version.

From a neuroscientific point of view, it is interesting to think of the softmax as a way to create a form of competition between the units that participate in it: the softmax outputs always sum to 1 so an increase in the value of one unit necessarily corresponds to a decrease in the value of others. This is analogous to the lateral inhibition that is believed to exist between nearby neurons in the cortex. At the extreme (when the difference between the maximal a_i and the others is large in magnitude) it becomes a form of **winner-take-all** (one of the outputs is nearly 1 and the others are nearly 0).

The name “softmax” can be somewhat confusing. The function is more closely related to the arg max function than the max function. The term “soft” derives from the fact that the softmax function is continuous and differentiable. The arg max function, with its result represented as a one-hot vector, is not continuous or differentiable. The softmax function thus provides a “softened” version of the arg max. The corresponding soft version of the maximum function is $\text{softmax}(\mathbf{z})^\top \mathbf{z}$. It would perhaps be better to call the softmax function “softargmax,” but the current name is an entrenched convention.

6.2.2.4 Other Output Types

The linear, sigmoid, and softmax output units described above are the most common. Neural networks can generalize to almost any kind of output layer that we wish. The principle of maximum likelihood provides a guide for how to design