

## 7.9 Parameter Tying and Parameter Sharing

Thus far, in this chapter, when we have discussed adding constraints or penalties to the parameters, we have always done so with respect to a fixed region or point. For example,  $L^2$  regularization (or weight decay) penalizes model parameters for deviating from the fixed value of zero. However, sometimes we may need other ways to express our prior knowledge about suitable values of the model parameters. Sometimes we might not know precisely what values the parameters should take but we know, from knowledge of the domain and model architecture, that there should be some dependencies between the model parameters.

A common type of dependency that we often want to express is that certain parameters should be close to one another. Consider the following scenario: we have two models performing the same classification task (with the same set of classes) but with somewhat different input distributions. Formally, we have model  $A$  with parameters  $\mathbf{w}^{(A)}$  and model  $B$  with parameters  $\mathbf{w}^{(B)}$ . The two models map the input to two different, but related outputs:  $\hat{y}^{(A)} = f(\mathbf{w}^{(A)}, \mathbf{x})$  and  $\hat{y}^{(B)} = g(\mathbf{w}^{(B)}, \mathbf{x})$ .

Let us imagine that the tasks are similar enough (perhaps with similar input and output distributions) that we believe the model parameters should be close to each other:  $\forall i, w_i^{(A)}$  should be close to  $w_i^{(B)}$ . We can leverage this information through regularization. Specifically, we can use a parameter norm penalty of the form:  $\Omega(\mathbf{w}^{(A)}, \mathbf{w}^{(B)}) = \|\mathbf{w}^{(A)} - \mathbf{w}^{(B)}\|_2^2$ . Here we used an  $L^2$  penalty, but other choices are also possible.

This kind of approach was proposed by [Lasserre et al. \(2006\)](#), who regularized the parameters of one model, trained as a classifier in a supervised paradigm, to be close to the parameters of another model, trained in an unsupervised paradigm (to capture the distribution of the observed input data). The architectures were constructed such that many of the parameters in the classifier model could be paired to corresponding parameters in the unsupervised model.

While a parameter norm penalty is one way to regularize parameters to be close to one another, the more popular way is to use constraints: *to force sets of parameters to be equal*. This method of regularization is often referred to as **parameter sharing**, because we interpret the various models or model components as sharing a unique set of parameters. A significant advantage of parameter sharing over regularizing the parameters to be close (via a norm penalty) is that only a subset of the parameters (the unique set) need to be stored in memory. In certain models—such as the convolutional neural network—this can lead to significant reduction in the memory footprint of the model.