

## 7.1 Parameter Norm Penalties

Regularization has been used for decades prior to the advent of deep learning. Linear models such as linear regression and logistic regression allow simple, straightforward, and effective regularization strategies.

Many regularization approaches are based on limiting the capacity of models, such as neural networks, linear regression, or logistic regression, by adding a parameter norm penalty  $\Omega(\boldsymbol{\theta})$  to the objective function  $J$ . We denote the regularized objective function by  $\tilde{J}$ :

$$\tilde{J}(\boldsymbol{\theta}; \mathbf{X}, \mathbf{y}) = J(\boldsymbol{\theta}; \mathbf{X}, \mathbf{y}) + \alpha\Omega(\boldsymbol{\theta}) \quad (7.1)$$

where  $\alpha \in [0, \infty)$  is a hyperparameter that weights the relative contribution of the norm penalty term,  $\Omega$ , relative to the standard objective function  $J$ . Setting  $\alpha$  to 0 results in no regularization. Larger values of  $\alpha$  correspond to more regularization.

When our training algorithm minimizes the regularized objective function  $\tilde{J}$  it will decrease both the original objective  $J$  on the training data and some measure of the size of the parameters  $\boldsymbol{\theta}$  (or some subset of the parameters). Different choices for the parameter norm  $\Omega$  can result in different solutions being preferred. In this section, we discuss the effects of the various norms when used as penalties on the model parameters.

Before delving into the regularization behavior of different norms, we note that for neural networks, we typically choose to use a parameter norm penalty  $\Omega$  that penalizes *only the weights* of the affine transformation at each layer and leaves the biases unregularized. The biases typically require less data to fit accurately than the weights. Each weight specifies how two variables interact. Fitting the weight well requires observing both variables in a variety of conditions. Each bias controls only a single variable. This means that we do not induce too much variance by leaving the biases unregularized. Also, regularizing the bias parameters can introduce a significant amount of underfitting. We therefore use the vector  $\mathbf{w}$  to indicate all of the weights that should be affected by a norm penalty, while the vector  $\boldsymbol{\theta}$  denotes all of the parameters, including both  $\mathbf{w}$  and the unregularized parameters.

In the context of neural networks, it is sometimes desirable to use a separate penalty with a different  $\alpha$  coefficient for each layer of the network. Because it can be expensive to search for the correct value of multiple hyperparameters, it is still reasonable to use the same weight decay at all layers just to reduce the size of search space.