Chapter 12

Loss landscape analysis

In Chapter 10, we saw how the weights of neural networks get adapted during training, using, e.g., variants of gradient descent. For certain cases, including the wide networks considered in Chapter 11, the corresponding iterative scheme converges to a global minimizer. In general, this is not guaranteed, and gradient descent can for instance get stuck in non-global minima or saddle points.

To get a better understanding of these situations, in this chapter we discuss the so-called loss landscape. This term refers to the graph of the empirical risk as a function of the weights. We give a more rigorous definition below, and first introduce notation for neural networks and their realizations for a fixed architecture.

Definition 12.1. Let $\mathcal{A} = (d_0, d_1, \dots, d_{L+1}) \in \mathbb{N}^{L+2}$, let $\sigma \colon \mathbb{R} \to \mathbb{R}$ be an activation function, and let $B > 0$. We denote the set of neural networks Φ with L layers, layer widths $d_0, d_1, \dots, d_{L+1}$, all weights bounded in modulus by B, and using the activation function σ by $\mathcal{N}(\sigma; \mathcal{A}, B)$. Additionally, we define

$$\mathcal{PN}(\mathcal{A}, B) \coloneqq \bigtimes_{\ell=0}^{L} \left([-B, B]^{d_{\ell+1} \times d_\ell} \times [-B, B]^{d_{\ell+1}} \right),$$

and the **realization map**

$$\begin{aligned} R_\sigma \colon \mathcal{PN}(\mathcal{A}, B) &\to \mathcal{N}(\sigma; \mathcal{A}, B) \\ (\boldsymbol{W}^{(\ell)}, \boldsymbol{b}^{(\ell)})_{\ell=0}^{L} &\mapsto \Phi, \end{aligned} \tag{12.0.1}$$

where Φ is the neural network with weights and biases given by $(\boldsymbol{W}^{(\ell)}, \boldsymbol{b}^{(\ell)})_{\ell=0}^{L}$.

Throughout, we will identify $\mathcal{PN}(\mathcal{A}, B)$ with the cube $[-B, B]^{n_\mathcal{A}}$, where $n_\mathcal{A} \coloneqq \sum_{\ell=0}^{L} d_{\ell+1}(d_\ell + 1)$. Now we can introduce the loss landscape of a neural network architecture.

Definition 12.2. Let $\mathcal{A} = (d_0, d_1, \dots, d_{L+1}) \in \mathbb{N}^{L+2}$, let $\sigma \colon \mathbb{R} \to \mathbb{R}$. Let $m \in \mathbb{N}$, and $S = (\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^{m} \in (\mathbb{R}^{d_0} \times \mathbb{R}^{d_{L+1}})^m$ be a sample and let $\mathcal{L}$ be a loss function. Then, the **loss landscape**

is the graph of the function $\Lambda_{\mathcal{A},\sigma,S,\mathcal{L}}$ defined as

$$\begin{aligned}\Lambda_{\mathcal{A},\sigma,S,\mathcal{L}} : \mathcal{PN}(\mathcal{A};\infty) &\to \mathbb{R} \\ \theta &\mapsto \widehat{\mathcal{R}}_S(R_\sigma(\theta)).\end{aligned}$$

with $\widehat{\mathcal{R}}_S$ in (1.2.3) and R_σ in (12.0.1).

Identifying $\mathcal{PN}(\mathcal{A},\infty)$ with $\mathbb{R}^{n_\mathcal{A}}$, we can consider $\Lambda_{\mathcal{A},\sigma,S,\mathcal{L}}$ as a map on $\mathbb{R}^{n_\mathcal{A}}$ and the loss landscape is a subset of $\mathbb{R}^{n_\mathcal{A}} \times \mathbb{R}$. The loss landscape is a high-dimensional surface, with hills and valleys. For visualization a two-dimensional section of a loss landscape is shown in Figure 12.1.

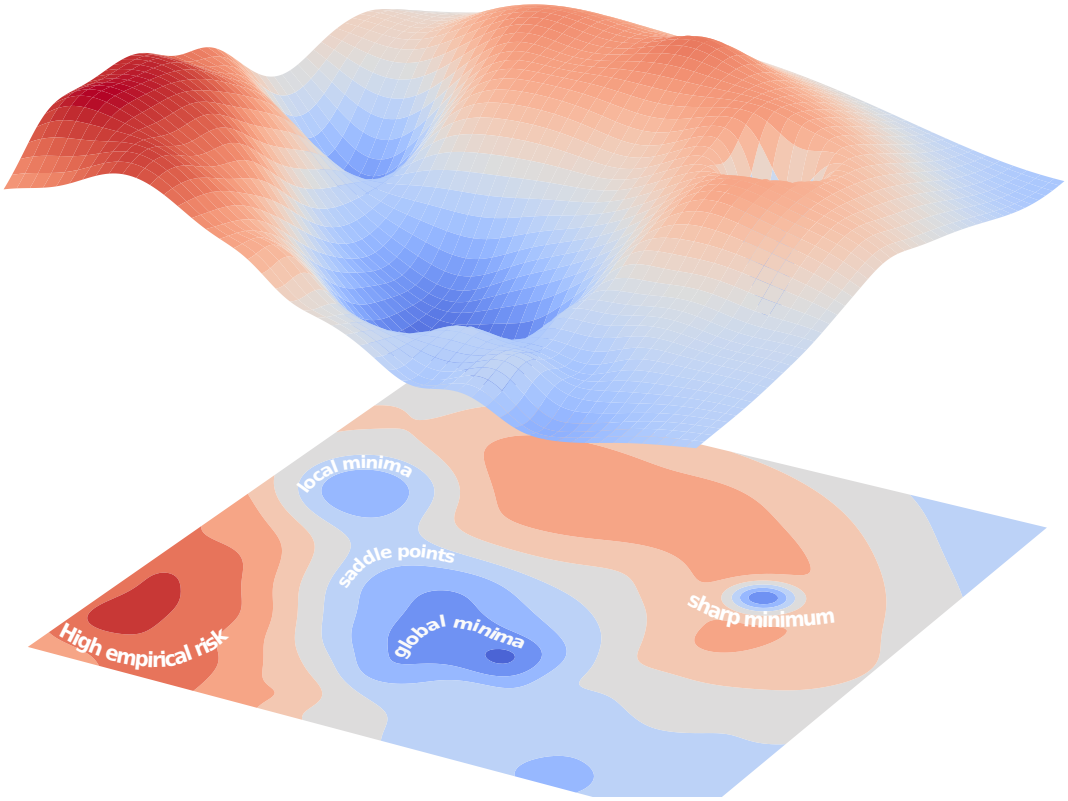

Figure 12.1: Two-dimensional section of a loss landscape. The loss landscape shows a spurious valley with local minima, global minima, as well as a region where saddle points appear. Moreover, a sharp minimum is shown.

Questions of interest regarding the loss landscape include for example: How likely is it that we find local instead of global minima? Are these local minima typically sharp, having small volume, or are they part of large flat valleys that are difficult to escape? How bad is it to end up in a local minimum? Are most local minima as deep as the global minimum, or can they be significantly higher? How rough is the surface generally, and how do these characteristics depend on the network architecture? While providing complete answers to these questions is hard in general, in the rest of this chapter we give some intuition and mathematical insights for specific cases.

12.1 Visualization of loss landscapes

Visualizing loss landscapes can provide valuable insights into the effects of neural network depth, width, and activation functions. However, we can only visualize an at most two-dimensional surface embedded into three-dimensional space, whereas the loss landscape is a very high-dimensional object (unless the neural networks have only very few weights and biases).

To make the loss landscape accessible, we need to reduce its dimensionality. This can be achieved by evaluating the function $\Lambda_{\mathcal{A},\sigma,S,\mathcal{L}}$ on a two-dimensional subspace of $\mathcal{PN}(\mathcal{A},\infty)$. Specifically, we choose three-parameters μ, θ_1, θ_2 and examine the function

$$\mathbb{R}^2 \ni (\alpha_1, \alpha_2) \mapsto \Lambda_{\mathcal{A},\sigma,S,\mathcal{L}}(\mu + \alpha_1\theta_1 + \alpha_2\theta_2). \tag{12.1.1}$$

There are various natural choices for μ, θ_1, θ_2:

- *Random directions:* This was, for example used in [73, 101]. Here θ_1, θ_2 are chosen randomly, while μ is either a minimum of $\Lambda_{\mathcal{A},\sigma,S,\mathcal{L}}$ or also chosen randomly. This simple approach can offer a quick insight into how rough the surface can be. However, as was pointed out in [132], random directions will very likely be orthogonal to the trajectory of the optimization procedure. Hence, they will likely miss the most relevant features.

- *Principal components of learning trajectory:* To address the shortcomings of random directions, another possibility is to determine μ, θ_1, θ_2, which best capture some given learning trajectory; For example, if $\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(N)}$ are the parameters resulting from the training by SGD, we may determine μ, θ_1, θ_2 such that the hyperplane $\{\mu + \alpha_1\theta_1 + \alpha_2\theta_2 \,|\, \alpha_1, \alpha_2 \in \mathbb{R}\}$ minimizes the mean squared distance to the $\theta^{(j)}$ for $j \in \{1, \dots, N\}$. This is the approach of [132], and can be achieved by a principal component analysis.

- *Based on critical points:* For a more global perspective, μ, θ_1, θ_2 can be chosen to ensure the observation of multiple critical points. One way to achieve this is by running the optimization procedure three times with final parameters $\theta^{(1)}$, $\theta^{(2)}$, $\theta^{(3)}$. If the procedures have converged, then each of these parameters is close to a critical point of $\Lambda_{\mathcal{A},\sigma,S,\mathcal{L}}$. We can now set $\mu = \theta^{(1)}$, $\theta_1 = \theta^{(2)} - \mu$, $\theta_2 = \theta^{(3)} - \mu$. This then guarantees that (12.1.1) passes through or at least comes very close to three critical points (at $(\alpha_1, \alpha_2) = (0,0), (0,1), (1,0)$). We present six visualizations of this form in Figure 12.2.

Figure 12.2 gives some interesting insight into the effect of depth and width on the shape of the loss landscape. For very wide and shallow neural networks, we have the widest minima, which, in the case of the tanh activation function also seem to belong to the same valley. With increasing depth and smaller width the minima get steeper and more disconnected.

12.2 Spurious minima

From the perspective of optimization, the ideal loss landscape has one global minimum in the center of a large valley, so that gradient descent converges towards the minimum irrespective of the chosen initialization.

This situation is not realistic for deep neural networks. Indeed, for a simple shallow neural network

$$\mathbb{R}^d \ni \boldsymbol{x} \mapsto \Phi(\boldsymbol{x}) = \boldsymbol{W}^{(1)}\sigma(\boldsymbol{W}^{(0)}\boldsymbol{x} + \boldsymbol{b}^{(0)}) + \boldsymbol{b}^{(1)},$$

it is clear that for every permutation matrix $\boldsymbol{P}$

$$\Phi(\boldsymbol{x}) = \boldsymbol{W}^{(1)}\boldsymbol{P}^T\sigma(\boldsymbol{P}\boldsymbol{W}^{(0)}\boldsymbol{x} + \boldsymbol{P}\boldsymbol{b}^{(0)}) + \boldsymbol{b}^{(1)} \qquad \text{for all } \boldsymbol{x} \in \mathbb{R}^d.$$

Hence, in general there exist multiple parameterizations realizing the same output function. Moreover, if least one global minimum with non-permutation-invariant weights exists, then there are more than one global minima of the loss landscape.

This is not problematic; in fact, having many global minima is beneficial. The larger issue is the existence of non-global minima. Following [234], we start by generalizing the notion of non-global minima to *spurious valleys*.

Definition 12.3. Let $\mathcal{A} = (d_0, d_1, \dots, d_{L+1}) \in \mathbb{N}^{L+2}$ and $\sigma\colon \mathbb{R} \to \mathbb{R}$. Let $m \in \mathbb{N}$, and $S = (\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^m \in (\mathbb{R}^{d_0} \times \mathbb{R}^{d_{L+1}})^m$ be a sample and let $\mathcal{L}$ be a loss function. For $c \in \mathbb{R}$, we define the sub-level set of $\Lambda_{\mathcal{A},\sigma,S,\mathcal{L}}$ as

$$\Omega_\Lambda(c) \coloneqq \{\theta \in \mathcal{PN}(\mathcal{A}, \infty) \,|\, \Lambda_{\mathcal{A},\sigma,S,\mathcal{L}}(\theta) \leq c\}.$$

A path-connected component of $\Omega_\Lambda(c)$, which does not contain a global minimum of $\Lambda_{\mathcal{A},\sigma,S,\mathcal{L}}$ is called a **spurious valley**.

The next proposition shows that spurious local minima do not exist for shallow overparameterized neural networks, i.e., for neural networks that have at least as many parameters in the hidden layer as there are training samples.

Proposition 12.4. *Let $\mathcal{A} = (d_0, d_1, 1) \in \mathbb{N}^3$ and let $S = (\boldsymbol{x}_i, y_i)_{i=1}^m \in (\mathbb{R}^{d_0} \times \mathbb{R})^m$ be a sample such that $m \leq d_1$. Furthermore, let $\sigma \in \mathcal{M}$ be not a polynomial, and let $\mathcal{L}$ be a convex loss function. Further assume that $\Lambda_{\mathcal{A},\sigma,S,\mathcal{L}}$ has at least one global minimum. Then, $\Lambda_{\mathcal{A},\sigma,S,\mathcal{L}}$, has no spurious valleys.*

Proof. Let $\theta_a, \theta_b \in \mathcal{PN}(\mathcal{A}, \infty)$ with $\Lambda_{\mathcal{A},\sigma,S,\mathcal{L}}(\theta_a) > \Lambda_{\mathcal{A},\sigma,S,\mathcal{L}}(\theta_b)$. Then we will show below that there is another parameter θ_c such that

- $\Lambda_{\mathcal{A},\sigma,S,\mathcal{L}}(\theta_b) = \Lambda_{\mathcal{A},\sigma,S,\mathcal{L}}(\theta_c)$
- there is a continuous path $\alpha : [0,1] \to \mathcal{PN}(\mathcal{A}, \infty)$ such that $\alpha(0) = \theta_a$, $\alpha(1) = \theta_c$, and $\Lambda_{\mathcal{A},\sigma,S,\mathcal{L}}(\alpha)$ is monotonically decreasing.

By Exercise 12.7, the construction above rules out the existence of spurious valleys by choosing θ_a an element of a spurious valley and θ_b a global minimum.

Next, we present the construction: Let us denote

$$\theta_o = \left(\left(\boldsymbol{W}_o^{(\ell)}, \boldsymbol{b}_o^{(\ell)}\right)_{\ell=0}^1\right) \qquad \text{for } o \in \{a, b, c\}.$$

Moreover, for $j = 1, \dots, d_1$, we introduce $\boldsymbol{v}_o^j \in \mathbb{R}^m$ defined as

$$(\boldsymbol{v}_o^j)_i = \left(\sigma\left(\boldsymbol{W}_o^{(0)}\boldsymbol{x}_i + \boldsymbol{b}_o^{(0)}\right)\right)_j \qquad \text{for } i = 1, \dots, m.$$

Notice that, if we set $\boldsymbol{V}_o = ((\boldsymbol{v}_o^j)^\top)_{j=1}^{d_1}$, then

$$\boldsymbol{W}_o^{(1)}\boldsymbol{V}_o = \left(R_\sigma(\theta_o)(\boldsymbol{x}_i) - \boldsymbol{b}_o^{(1)}\right)_{i=1}^m, \tag{12.2.1}$$

where the right-hand side is considered a row-vector.

We will now distinguish between two cases. For the first the result is trivial and the second can be transformed into the first one.

Case 1: Assume that $\boldsymbol{V}_a$ has rank m. In this case, it is obvious from (12.2.1), that there exists $\widetilde{\boldsymbol{W}}$ such that

$$\widetilde{\boldsymbol{W}}\boldsymbol{V}_a = \left(R_\sigma(\theta_b)(\boldsymbol{x}_i) - \boldsymbol{b}_a^{(1)}\right)_{i=1}^m.$$

We can thus set $\alpha(t) = ((\boldsymbol{W}_a^{(0)}, \boldsymbol{b}_a^{(0)}), ((1-t)\boldsymbol{W}_a^{(1)} + t\widetilde{\boldsymbol{W}}, \boldsymbol{b}_a^{(1)})$.

Note that by construction $\alpha(0) = \theta_a$ and $\Lambda_{\mathcal{A},\sigma,S,\mathcal{L}}(\alpha(1)) = \Lambda_{\mathcal{A},\sigma,S,\mathcal{L}}(\theta_b)$. Moreover, $t \mapsto (R_\sigma(\alpha(t))(\boldsymbol{x}_i))_{i=1}^m$ describes a straight path in $\mathbb{R}^m$ and hence, by the convexity of $\mathcal{L}$ it is clear that $t \mapsto \Lambda_{\mathcal{A},\sigma,S,\mathcal{L}}(\alpha(t))$ is monotonically decreasing.

Case 2: Assume that V_a has rank less than m. In this case, we show that we find a continuous path from θ_a to another neural network parameter with higher rank. The path will be such that $\Lambda_{\mathcal{A},\sigma,S,\mathcal{L}}$ is monotonically decreasing.

Under the assumptions, we have that one $\boldsymbol{v}_a^j$ can be written as a linear combination of the remaining $\boldsymbol{v}_a^i$, $i \neq j$. Without loss of generality, we assume $j = 1$. Then, there exist $(\alpha_i)_{i=2}^m$ such that

$$\boldsymbol{v}_a^1 = \sum_{i=2}^m \alpha_i \boldsymbol{v}_a^i. \tag{12.2.2}$$

Next, we observe that here exists $\boldsymbol{v}^* \in \mathbb{R}^m$ which is linearly independent from all $(\boldsymbol{v}_a^j)_{i=1}^m$ and can be written as $(\boldsymbol{v}^*)_i = \sigma((\boldsymbol{w}^*)^\top \boldsymbol{x}_i + b^*)$ for some $\boldsymbol{w}^* \in \mathbb{R}^{d_0}, b^* \in \mathbb{R}$. Indeed, if we assume that such $\boldsymbol{v}^*$ does not exist, then it follows that $\operatorname{span}\{(\sigma(\boldsymbol{w}^\top \boldsymbol{x}_i + b))_{i=1}^m \mid \boldsymbol{w} \in \mathbb{R}^{d_0}, b \in \mathbb{R}\}$ is an $m-1$ dimensional subspace of $\mathbb{R}^m$ which yields a contradiction to Theorem 9.3.

Now, we define two paths: First,

$$\alpha_1(t) = ((\boldsymbol{W}_a^{(0)}, \boldsymbol{b}_a^{(0)}), (\boldsymbol{W}_a^{(1)}(t), \boldsymbol{b}_a^{(1)})), \qquad \text{for } t \in [0, 1/2]$$

where

$$(\boldsymbol{W}_a^{(1)}(t))_1 = (1-2t)(\boldsymbol{W}_a^{(1)})_1 \qquad \text{and} \qquad (\boldsymbol{W}_a^{(1)}(t))_i = (\boldsymbol{W}_a^{(1)})_i + 2t\alpha_i(\boldsymbol{W}_a^{(1)})_1$$

for $i = 2, \dots, d_1$, for $t \in [0, 1/2]$. Second,

$$\alpha_2(t) = ((\boldsymbol{W}_a^{(0)}(t), \boldsymbol{b}_a^{(0)}(t)), (\boldsymbol{W}_a^{(1)}(1/2), \boldsymbol{b}_a^{(1)})), \text{ for } t \in (1/2, 1],$$

where

$$(\boldsymbol{W}_a^{(0)}(t))_1 = 2(t-1/2)(\boldsymbol{W}_a^{(0)})_1 + (2t-1)\boldsymbol{w}^* \qquad \text{and} \qquad (\boldsymbol{W}_a^{(0)}(t))_i = (\boldsymbol{W}_a^{(0)})_i$$

for $i = 2, \dots, d_1$, $(\boldsymbol{b}_a^{(0)}(t))_1 = 2(t-1/2)(\boldsymbol{b}_a^{(0)})_1 + (2t-1)b^*$, and $(\boldsymbol{b}_a^{(0)}(t))_i = (\boldsymbol{b}_a^{(0)})_i$ for $i = 2, \dots, d_1$.

It is clear by (12.2.2) that $(R_\sigma(\alpha_1)(\boldsymbol{x}_i))_{i=1}^m$ is constant. Moreover, $R_\sigma(\alpha_2)(\boldsymbol{x})$ is constant for all $\boldsymbol{x} \in \mathbb{R}^{d_0}$. In addition, by construction for

$$\bar{\boldsymbol{v}}^j \coloneqq \left(\left(\sigma \left(\boldsymbol{W}_a^{(0)}(1)\boldsymbol{x}_i + \boldsymbol{b}_a^{(0)}(1) \right) \right)_j \right)_{i=1}^m$$

it holds that $((\bar{\boldsymbol{v}}^j)^\top)_{j=1}^{d_1}$ has rank larger than that of $\boldsymbol{V}_a$. Concatenating α_1 and α_2 now yields a continuous path from θ_a to another neural network parameter with higher associated rank such that $\Lambda_{\mathcal{A},\sigma,S,\mathcal{L}}$ is monotonically decreasing along the path. Iterating this construction, we can find a path to a neural network parameter where the associated matrix has full rank. This reduces the problem to Case 1. □

12.3 Saddle points

Saddle points are critical points of the loss landscape at which the loss decreases in one direction. In this sense, saddle points are not as problematic as local minima or spurious valleys if the updates in the learning iteration have some stochasticity. Eventually, a random step in the right direction could be taken and the saddle point can be escaped.

If most of the critical points are saddle points, then, even though the loss landscape is challenging for optimization, one still has a good chance of eventually reaching the global minimum. Saddle points of the loss landscape were studied in [44, 170] and we will review some of the findings in a simplified way below. The main observation in [170] is that, under some quite strong assumptions, it holds that *critical points in the loss landscape associated to a large loss are typically saddle points, whereas those associated to small loss correspond to minima.* This situation is encouraging for the prospects of optimization in deep learning, since, even if we get stuck in a local minimum, it will very likely be such that the loss is close to optimal.

The results of [170] use random matrix theory, which we do not recall here. Moreover, it is hard to gauge if the assumptions made are satisfied for a specific problem. Nonetheless, we recall the main idea, which provides some intuition to support the above claim.

Let $\mathcal{A} = (d_0, d_1, 1) \in \mathbb{N}^3$. Then, for a neural network parameter $\theta \in \mathcal{PN}(\mathcal{A}, \infty)$ and activation function σ, we set $\Phi_\theta \coloneqq R_\sigma(\theta)$ and define for a sample $S = (\boldsymbol{x}_i, y_i)_{i=1}^m$ the errors

$$e_i = \Phi_\theta(\boldsymbol{x}_i) - y_i \qquad \text{for } i = 1, \dots, m.$$

If we use the square loss, then

$$\widehat{\mathcal{R}_S}(\Phi_\theta) = \frac{1}{m} \sum_{i=1}^m e_i^2. \tag{12.3.1}$$

Next, we study the Hessian of $\widehat{\mathcal{R}}_S(\Phi_\theta)$.

Proposition 12.5. *Let $\mathcal{A} = (d_0, d_1, 1)$ and $\sigma : \mathbb{R} \to \mathbb{R}$. Then, for every $\theta \in \mathcal{PN}(\mathcal{A}, \infty)$ where $\widehat{\mathcal{R}}_S(\Phi_\theta)$ in (12.3.1) is twice continuously differentiable with respect to the weights, it holds that*

$$\boldsymbol{H}(\theta) = \boldsymbol{H}_0(\theta) + \boldsymbol{H}_1(\theta),$$

where $\boldsymbol{H}(\theta)$ is the Hessian of $\widehat{\mathcal{R}}_S(\Phi_\theta)$ at θ, $\boldsymbol{H}_0(\theta)$ is a positive semi-definite matrix which is independent from $(y_i)_{i=1}^m$, and $\boldsymbol{H}_1(\theta)$ is a symmetric matrix that for fixed θ and $(\boldsymbol{x}_i)_{i=1}^m$ depends linearly on $(e_i)_{i=1}^m$.

Proof. Using the identification introduced after Definition 12.2, we can consider θ a vector in $\mathbb{R}^{n_\mathcal{A}}$. For $k = 1, \dots, n_\mathcal{A}$, we have that

$$\frac{\partial \widehat{\mathcal{R}}_S(\Phi_\theta)}{\partial \theta_k} = \frac{2}{m} \sum_{i=1}^m e_i \frac{\partial \Phi_\theta(\boldsymbol{x}_i)}{\partial \theta_k}.$$

Therefore, for $j = 1, \dots, n_\mathcal{A}$, we have, by the Leibniz rule, that

$$\begin{aligned} \frac{\partial^2 \widehat{\mathcal{R}}_S(\Phi_\theta)}{\partial \theta_j \partial \theta_k} &= \frac{2}{m} \sum_{i=1}^m \left(\frac{\partial \Phi_\theta(\boldsymbol{x}_i)}{\partial \theta_j} \frac{\partial \Phi_\theta(\boldsymbol{x}_i)}{\partial \theta_k} \right) + \frac{2}{m} \left(\sum_{i=1}^m e_i \frac{\partial^2 \Phi_\theta(\boldsymbol{x}_i)}{\partial \theta_j \partial \theta_k} \right) \\ &=: \boldsymbol{H}_0(\theta) + \boldsymbol{H}_1(\theta). \end{aligned} \tag{12.3.2}$$

It remains to show that $\boldsymbol{H}_0(\theta)$ and $\boldsymbol{H}_1(\theta)$ have the asserted properties. Note that, setting

$$J_{i,\theta} = \begin{pmatrix} \frac{\partial \Phi_\theta(\boldsymbol{x}_i)}{\partial \theta_1} \\ \vdots \\ \frac{\partial \Phi_\theta(\boldsymbol{x}_i)}{\partial \theta_{n_\mathcal{A}}} \end{pmatrix} \in \mathbb{R}^{n_\mathcal{A}},$$

we have that $\boldsymbol{H}_0(\theta) = \frac{2}{m} \sum_{i=1}^m J_{i,\theta} J_{i,\theta}^\top$ and hence $\boldsymbol{H}_0(\theta)$ is a sum of positive semi-definite matrices, which shows that $\boldsymbol{H}_0(\theta)$ is positive semi-definite.

The symmetry of $\boldsymbol{H}_1(\theta)$ follows directly from the symmetry of second derivatives which holds since we assumed twice continuous differentiability at θ. The linearity of $\boldsymbol{H}_1(\theta)$ in $(e_i)_{i=1}^m$ is clear from (12.3.2). □

How does Proposition 12.5 imply the claimed relationship between the size of the loss and the prevalence of saddle points?

Let θ correspond to a critical point. If $\boldsymbol{H}(\theta)$ has at least one negative eigenvalue, then θ cannot be a minimum, but instead must be either a saddle point or a maximum. While we do not know anything about $\boldsymbol{H}_1(\theta)$ other than that it is symmetric, it is not unreasonable to assume that it has a negative eigenvalue especially if $n_\mathcal{A}$ is very large. With this consideration, let us consider the following model:

Fix a parameter θ. Let $S^0 = (\boldsymbol{x}_i, y_i^0)_{i=1}^m$ be a sample and $(e_i^0)_{i=1}^m$ be the associated errors. Further let $\boldsymbol{H}^0(\theta), \boldsymbol{H}_0^0(\theta), \boldsymbol{H}_1^0(\theta)$ be the matrices according to Proposition 12.5.

Further let for $\lambda > 0$, $S^\lambda = (\boldsymbol{x}_i, y_i^\lambda)_{i=1}^m$ be such that the associated errors are $(e_i)_{i=1}^m = \lambda (e_i^0)_{i=1}^m$. The Hessian of $\widehat{\mathcal{R}}_{S^\lambda}(\Phi_\theta)$ at θ is then $\boldsymbol{H}^\lambda(\theta)$ satisfying

$$\boldsymbol{H}^\lambda(\theta) = \boldsymbol{H}_0^0(\theta) + \lambda \boldsymbol{H}_1^0(\theta).$$

Hence, if λ is large, then $\boldsymbol{H}^\lambda(\theta)$ is perturbation of an amplified version of $\boldsymbol{H}_1^0(\theta)$. Clearly, if $\boldsymbol{v}$ is an eigenvector of $\boldsymbol{H}_1(\theta)$ with negative eigenvalue $-\mu$, then

$$\boldsymbol{v}^\top \boldsymbol{H}^\lambda(\theta) \boldsymbol{v} \leq (\|\boldsymbol{H}_0^0(\theta)\| - \lambda\mu) \|\boldsymbol{v}\|^2,$$

which we can expect to be negative for large λ. Thus, $\boldsymbol{H}^{\lambda}(\theta)$ has a negative eigenvalue for large λ.

On the other hand, if λ is small, then $\boldsymbol{H}^{\lambda}(\theta)$ is merely a perturbation of $\boldsymbol{H}_0^0(\theta)$ and we can expect its spectrum to resemble that of $\boldsymbol{H}_0^0$ more and more.

What we see is that, the same parameter, is more likely to be a saddle point for a sample that produces a high empirical risk than for a sample with small risk. Note that, since $\boldsymbol{H}_0^0(\theta)$ was only shown to be *semi*-definite the argument above does not rule out saddle points even for very small λ. But it does show that for small λ, every negative eigenvalue would be very small.

A more refined analysis where we compare different parameters but for the same sample and quantify the likelihood of local minima versus saddle points requires the introduction of a probability distribution on the weights. We refer to [170] for the details.

Bibliography and further reading

The results on visualization of the loss landscape are inspired by [132, 73, 101]. Results on the non-existence of spurious valleys can be found in [234] with similar results in [182]. In [38] the loss landscape was studied by linking it to so-called spin-glass models. There it was found that under strong assumptions critical points associated to lower losses are more likely to be minima than saddle points. In [170], random matrix theory is used to provide similar results, that go beyond those established in Section 12.3. On the topic of saddle points, [44] identifies the existence of saddle points as more problematic than that of local minima, and an alternative saddle-point aware optimization algorithm is introduced.

Two essential topics associated to the loss landscape that have not been discussed in this chapter are mode connectivity and the sharpness of minima. Mode connectivity, roughly speaking describes the phenomenon, that local minima found by SGD over deep neural networks are often connected by simple curves of equally low loss [63, 53]. Moreover, the sharpness of minima has been analyzed and linked to generalization capabilities of neural networks, with the idea being that wide neural networks are easier to find and also yield robust neural networks [91, 33, 246]. However, this does not appear to exclude sharp minima from generalizing well [52].

Exercises

Exercise 12.6. In view of Definition 12.3, show that a local minimum of a differentiable function is contained in a spurious valley.

Exercise 12.7. Show that if there exists a continuous path α between a parameter θ_1 and a global minimum θ_2 such that $\Lambda_{\mathcal{A},\sigma,S,\mathcal{L}}(\alpha)$ is monotonically decreasing, then θ_1 cannot be an element of a spurious valley.

Exercise 12.8. Find an example of a spurious valley for a simple architecture. (*Hint:* Use a single neuron ReLU neural network and observe that, for two networks one with positive and one with negative slope, every continuous path in parameter space that connects the two has to pass through a parameter corresponding to a constant function.)

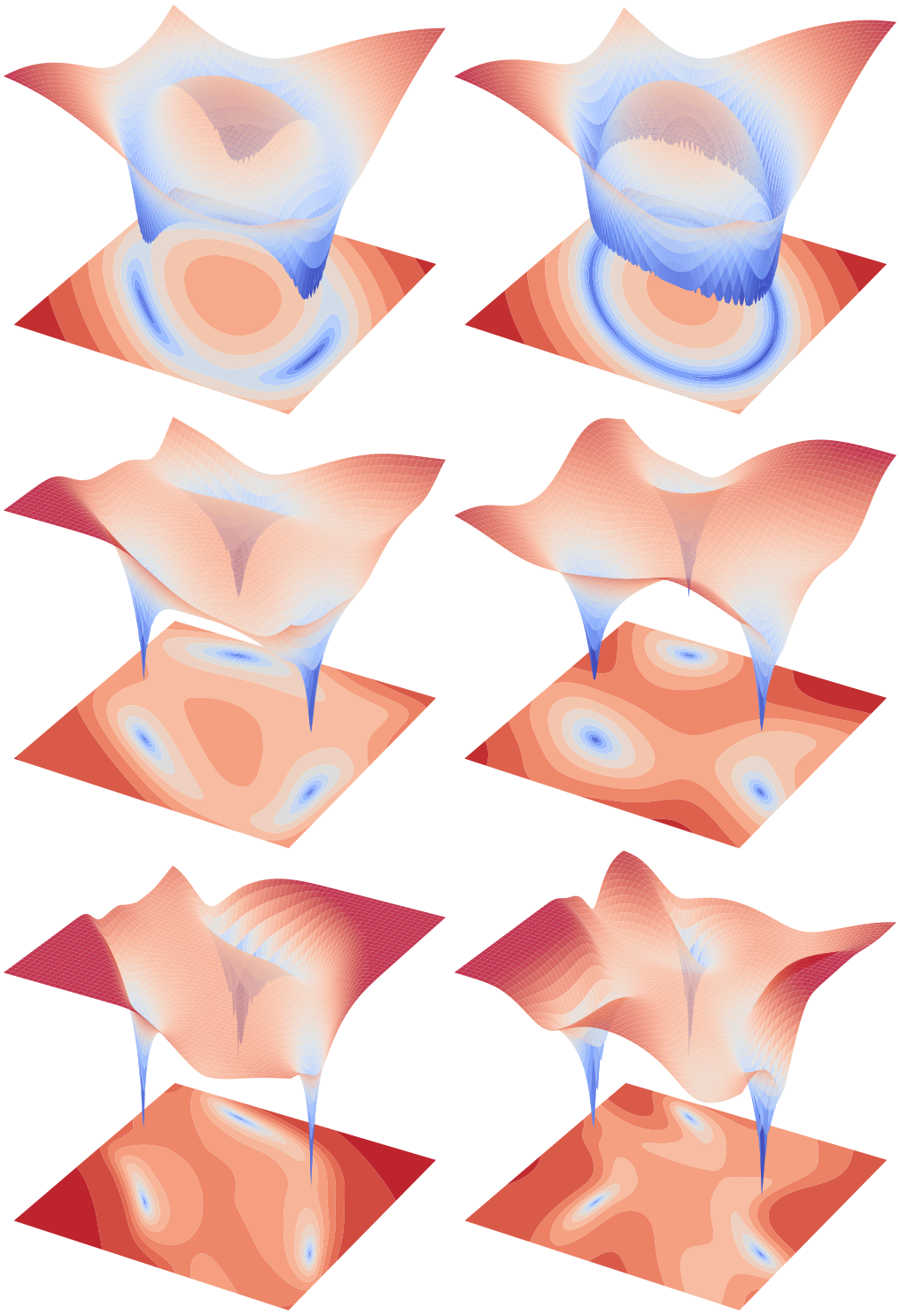

Figure 12.2: A collection of loss landscapes. In the left column are neural networks with ReLU activation function, the right column shows loss landscapes of neural networks with the hyperbolic tangent activation function. All neural networks have five dimensional input, and one dimensional output. Moreover, from top to bottom the hidden layers have sizes 1000, 20, 10, and the number of layers are 1, 4, 7.