Chapter 8

Nonconvex Optimization

In the previous chapter, we outlined conceptual topics in deep learning theory and how the situation was different from classical machine learning theory. In particular, we described *approximation theory*, *statistical generalization* and *optimization*. In this chapter, we will focus on optimization theory in deep learning. We will introduce some basics about optimization (Section 8.2), discuss how we can make the notion "all local minima are global minima" rigorous, and walk through two examples where this is the case (Section 8.3). Finally, we introduce the neural tangent kernel approach which allows us to characterize of the loss of general neural networks near a specific initialization (or under specific parameterization).

8.1 Optimization landscape

The big question that we have in mind is the following: many existing optimizers are designed for optimizing convex functions. **Why do they still work well empirically for non-convex functions?** We note that it is not true that these optimizers always work well with non-convex functions: there are still some very hard cases that give trouble (e.g. very deep feed-forward networks are still hard to fit because of issues like vanishing and exploding gradients). One possible reason is that the non-convex functions that we are minimizing in deep learning usually have some nice properties: see Figure 8.1 for an illustration.

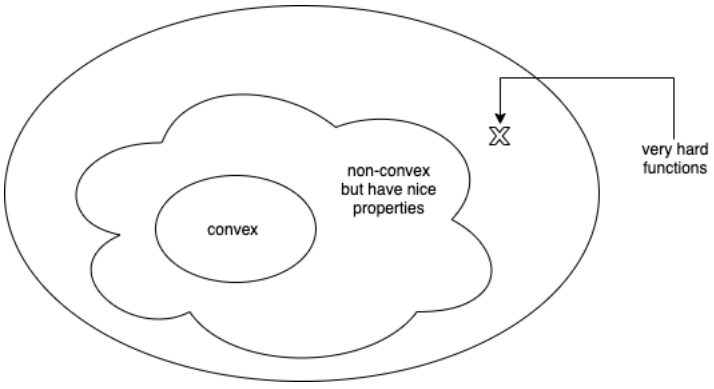

Figure 8.1: Classification of different functions for optimization. The functions we optimize in deep learning seem to fall mostly within the middle cloud.

Before diving into details, we first highlight some observations that will be important to keep in mind when discussing optimization in deep learning. Suppose $g(\theta)$ is the loss function. Recall that the *gradient descent (GD)* algorithm would do the following:

1. $\theta_0 \triangleq$ initialization

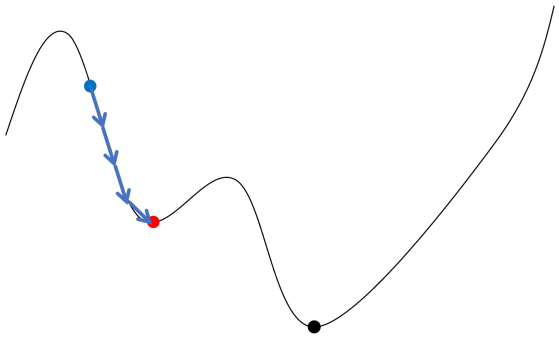

Figure 8.2: Illustration of how gradient descent does not always find the global minimum. In the picture, gradient descent initialized at the blue point only makes it to the local minimum at the red point: it does not find the global minimum at the black point.

2. $\theta_{t+1} = \theta_t - \eta \nabla g(\theta_t)$, where η is the step size.

Here are some observations to :

Observation 1: Gradient descent can find a global minimum for convex functions[1] but cannot always find the global minimum for any general continuous functions (see Figure 8.2 for an illustration).

Observation 2: Finding the global minimum of general non-convex functions is NP-hard.

Observation 3: The objective function in deep learning is non-convex., but empirically gradient descent/stochastic gradient descent typically finds an approximate global minimum of loss function in deep learning.

These observations motivate the following two-step plan:

1. Identify a large set of functions that stochastic gradient descent/gradient descent can solve.
2. Prove that some of the loss functions in machine learning problems belong to this set. (Most of the effort will be spent here.)

Basic idea: Gradient descent can find local minimum + all local minima of f are also global $\Rightarrow$ Gradient descent can find global minima.

8.2 Efficient convergence to (approximate) local minima

Let f be a twice-differentiable function. We start with the following definition:

Definition 8.1 (Local minimum of a function)**.** We say that x is a *local minimum* of a function f if there exists an open neighborhood N around x such that in N, the function values are at least $f(x)$.

Note that if x is a local minimum of f, then $\nabla f(x) = 0$ and $\nabla^2 f(x) \succeq 0$. However, as the next example shows, the reverse is not true. When $\nabla f(x) = 0$ and $\nabla^2 f(x)$ vanishes in some direction (i.e. merely positive semi-definite instead of being strictly positive definite), higher-order derivatives start to matter.

Example 8.2. Consider the function $f(x_1, x_2) = x_1^2 + x_2^3$. $(x_1, x_2) = (0, 0)$ satisfies $\nabla f(x) = 0$ and $\nabla^2 f(x)|_{(x_1,x_2)=(0,0)} = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} \succeq 0$. However, if we move in the negative direction of x_2, we can decrease the function value. Hence, this example shows why $\nabla f(x) = 0$ and $\nabla^2 f(x) \succeq 0$ does not imply that x is a local minimum.

[1] A more precise version of this claim is that gradient descent can find a point that has function value arbitrary close to the global minimal value.

It is generally not easy to verify if a point is a local minimum. In fact, we have the following theorem regarding the computational tractability:

Theorem 8.3. *It is NP-hard to check whether a point is a local minimum or not [Murty and Kabadi, 1987]. In addition, Hillar and Lim [Hillar and Lim, 2013] show that a degree four polynomial is NP-hard to optimize.*

8.2.1 Strict-saddle condition

Theorem 8.3 forces us to consider more specific types of functions to be able to obtain computational tractability. To this end, we define the following *strict-saddle condition*:

Definition 8.4 (Strict-saddle condition [Lee et al., 2016])**.** For positive α, β, γ, we say that $f : \mathbb{R}^d \mapsto \mathbb{R}$ is (α, β, γ)-*strict-saddle* if every $x \in \mathbb{R}^d$ satisfies one of the following:

1. $\|\nabla f(x)\|_2 \geq \alpha$.
2. $\lambda_{\min}(\nabla^2 f(x)) \leq -\beta$.
3. x is γ-close to a local minimum x^* in Euclidean distance, i.e. $\|x - x^*\|_2 \leq \gamma$.

Intuitively speaking, this definition is saying if a point has zero gradient and positive semi-definite Hessian, it must be close to a local minimum, i.e. there is no pathological case like Example 8.2.

We have the following theorem for functions that satisfy strict-saddle condition:

Theorem 8.5 (Informally stated)**.** *If f is (α, β, γ)-strict-saddle for some positive α, β, γ, then many optimizers (e.g. gradient descent, stochastic gradient descent, cubic regularization) can converge to a local minimum with ϵ-error in Euclidean distance in time poly $\left(d, \frac{1}{\alpha}, \frac{1}{\beta}, \frac{1}{\gamma}, \frac{1}{\epsilon}\right)$.*

Therefore, if all local minima are global minima and the function satisfies the strict-saddle condition, then optimizers can converge to a global minimum with ϵ-error in polynomial time. (See Figure 8.3 for an example of a function whose local minima are all global minima.) The next theorem expresses this concretely by being explicit about the strict-saddle condition:

Theorem 8.6. *Suppose f is a function that satisfies the following condition: $\exists\ \epsilon_0, \tau_0, c > 0$ such that if $x \in \mathbb{R}^d$ satisfies $\|\nabla f(x)\|_2 \leq \epsilon < \epsilon_0$ and $\nabla^2 f(x) \succeq -\tau_0 I$, then x is ϵ^c-close to a global minimum of f. Then many optimizers can converge to a global minimum of f up to δ-error in Euclidean distance in time poly $\left(\frac{1}{\delta}, \frac{1}{\tau_0}, d\right)$.*

8.3 All local minima are global minima: two examples

So far, we have focused on general results. Next, we give two concrete examples that have the property that all local minima are global minima: (i) principal components analysis (PCA)/matrix factorization/linearized neural nets, and (ii) matrix completion.

8.3.1 Principal components analysis (PCA)

Let matrix $M \in \mathbb{R}^{d \times d}$ be symmetric and positive semi-definite. Consider the problem of finding the best rank-1 approximation of the matrix M. The objective function here is non-convex:

$$\min_{x \in \mathbb{R}^d} g(x) \triangleq \frac{1}{2}\|M - xx^\top\|_F^2. \tag{8.1}$$

Theorem 8.7. *All local minima of g are global minima (even though g is non-convex).*

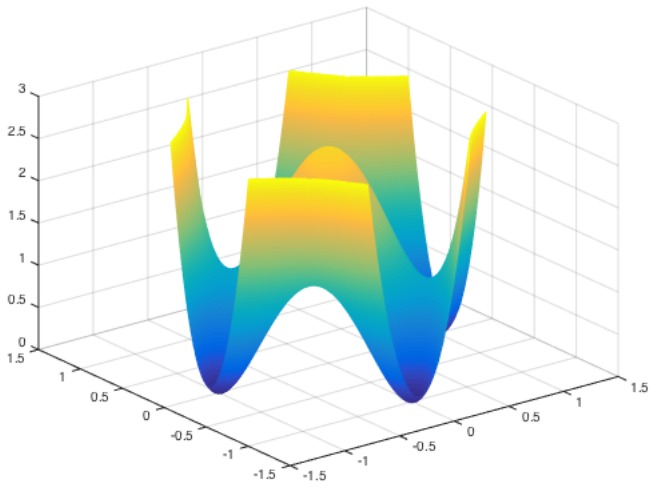

Figure 8.3: A two-dimensional function with the property that all local minima are global minima. It also satisfies the strict-saddle condition because all the saddle points have a strictly negative curvature in some direction.

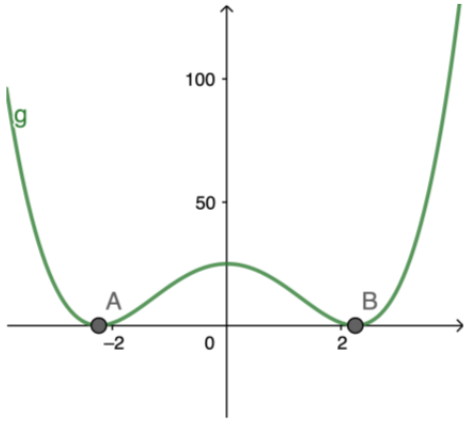

Figure 8.4: Objective function for principal components analysis (PCA) when $d = 1$.

Remark 8.8. For $d = 1$, $g(x) = \frac{1}{2}(m - x^2)^2$ for some constant m. Figure 8.4 below shows such an example. We can see that all local minima are indeed global minima.

Proof. Step 1: Show that all stationary points must be eigenvectors. From HW0, we know that $\nabla g(x) = -(M - xx^\top)x$, hence

$$\nabla g(x) = 0 \implies Mx = \|x\|_2^2 \cdot x, \tag{8.2}$$

which implies that x is an eigenvector of M with eigenvalue $\|x\|_2^2$. From the Eckart–Young–Mirsky theorem we know the global minimum (i.e. the best rank-1 approximation) is the eigenvector with the largest eigenvalue.

Step 2: Show that all local minima must be eigenvectors of the largest eigenvalue. We use the second order condition for this. For x to be a local minimum we need $\nabla^2 g(x) \succeq 0$, which means for any $v \in \mathbb{R}^d$,

$$\langle v, \nabla^2 g(x) v \rangle \geq 0. \tag{8.3}$$

To compute $\langle v, \nabla^2 g(x) v \rangle$, we use the following trick: expand $g(x+v)$ into $g(x)$ + linear term in v + quadratic term in v, then the quadratic term will be $\frac{1}{2}\langle v, \nabla^2 g(x) v \rangle$ (see HW0 Problem 2d for an example). Using this trick, we get

$$\begin{aligned} g(x+v) &= \frac{1}{2}\|M - (x+v)(x+v)^\top\|_F^2 && (8.4)\\ &= \frac{1}{2}\|M - xx^\top\|_F^2 - \langle M - xx^\top, xv^\top + vx^\top \rangle + \frac{1}{2}\langle xv^\top + vx^\top, xv^\top + vx^\top \rangle \\ &\quad - \langle M - xx^\top, vv^\top \rangle + \text{higher order terms in } v. && (8.5) \end{aligned}$$

Hence, we have

$$\begin{aligned} \frac{1}{2}\langle v, \nabla^2 g(x) v \rangle &= \frac{1}{2}\langle xv^\top + vx^\top, xv^\top + vx^\top \rangle - \langle M - xx^\top, vv^\top \rangle && (8.6)\\ &= \langle x, v \rangle^2 + \|x\|_2^2 \|v\|_2^2 - v^M v + \langle x, v \rangle^2 && (8.7)\\ &= 2\langle x, v \rangle^2 + \|x\|_2^2 \|v\|_2^2 - v^\top M v. && (8.8) \end{aligned}$$

Picking $v = v_1$, the unit eigenvector with the largest eigenvalue (denoted λ_1), for x to be a local minimum it must satisfy

$$\langle v_1, \nabla^2 g(x) v_1 \rangle = 2\langle x, v_1 \rangle^2 - v_1^\top M v_1 + \|x\|_2^2 \geq 0. \tag{8.9}$$

Note that by (8.2), all our candidates for local minima are eigenvectors of M so naturally we have two cases:

- *Case 1: x has eigenvalue λ_1.* Then x is the global minimum (by the Eckart–Young–Mirsky theorem).
- *Case 2: x has eigenvalue $\lambda < \lambda_1$.* Then we know x and v_1 are orthogonal (eigenvectors with different eigenvalues are always orthogonal), hence

  $$2\langle x, v_1 \rangle^2 - v_1^\top M v_1 + \|x\|_2^2 = 0 - \lambda_1 + \lambda \geq 0, \tag{8.10}$$

  which implies $\lambda \geq \lambda_1$, a contradiction.

In summary, if x is a stationary point and x is not a global minimum, then moving in the direction of v_1 would lead to second-order improvement and x cannot be a local minimum. □

8.3.2 Matrix Completion [Ge et al., 2016]

We consider rank-1 matrix completion for simplicity. Let $M = zz^\top$ be a rank-1 symmetric and positive semi-definite matrix for some $z \in \mathbb{R}^d$. Given random entries of M, our goal is to recover the rest of entries. Formally, we have the following definitions:

Definition 8.9. Suppose $M \in \mathbb{R}^{d \times d}$ and $\Omega \subseteq [d] \times [d]$, we define $P_\Omega(M)$ to be the matrix obtained by zeroing out every entry outside Ω.

Definition 8.10 (Matrix Completion)**.** Suppose $M \in \mathbb{R}^{d \times d}$ and every entry of M is included in Ω with probability p. The *matrix completion task* is to recover M (with respect to some loss functions) given the observation $P_\Omega(M)$.

A nice real world example of matrix completion is when we have a matrix describing the user ratings for each item. We only observe a small portion of the entries as each customer only buys a small subset of the items. A good matrix completion algorithm is indispensable for a recommendation engine.

Remark 8.11. We need d parameters to describe a rank-1 matrix M and the number of observations is roughly pd^2. Thus, for identifiability we need to work in the regime where $pd^2 > d$, i.e. $p \gg \frac{1}{d}$.

We define our non-convex loss functions to be

$$\min_{x \in \mathbb{R}^d} f(x) \triangleq \frac{1}{2} \sum_{(i,j) \in \Omega} (M_{ij} - x_i x_j)^2 \tag{8.11}$$

$$= \frac{1}{2} \|P_\Omega(M - xx^\top)\|_F^2. \tag{8.12}$$

To really solve our problem we need some regularity condition on the ground truth vector z (recall $M = zz^\top$). *Incoherence* is one such condition:

Definition 8.12 (Incoherence)**.** Without loss of generality, assume the ground truth vector $z \in \mathbb{R}^d$ satisfies $\|z\|_2 = 1$. z satisfies the *incoherence condition* if $\|z\|_\infty \leq \frac{\mu}{\sqrt{d}}$, where μ is considered to be a constant or log in dimension d.

Remark 8.13. A nice counterexample to think about why such condition is necessary is when $z = e_1$ and $M = e_1 e_1^\top$. All entries of M are 0 except for a 1 in the top-left corner. There is no way to recover M without observing the top-left corner.

The goal is to prove that local minima of this objective function are close to a global minimum:

Theorem 8.14. *Assume $p = \dfrac{poly(\mu, \log d)}{d\epsilon^2}$ for some sufficient small constant ϵ and assume z is incoherent. Then with high probability, all local minima of f are $O(\sqrt{\epsilon})$-close to $+z$ or $-z$ (the global minima of f).*

Before presenting the proof, we make some observations that will guide the proof strategy.

Remark 8.15. $f(x)$ can be viewed as a sampled version of the PCA loss function $g(x) = \frac{1}{2}\|M - xx^\top\|_F^2 = \frac{1}{2}\sum_{(i,j)\in[d]\times[d]}(M_{ij} - x_i x_j)^2$, in which we only observe a subset of the matrix entries. Thus, we would like to claim that $f(x) \approx g(x)$. However, matching the values of f and g is not sufficient to prove the theorem: even a small margin of error between f and g could lead to creation of many spurious local minima (see Figure 8.5 for an illustration). In order to ensure that the local minima of f look like the local minima of g, we will need further conditions like $\nabla f(x) \approx \nabla g(x)$ and $\nabla^2 f(x) \approx \nabla^2 g(x)$.

Remark 8.16. Key idea: concentration for scalars is easy. We can approximate a sum of scalars via a sample:

$$\sum_{(i,j)\in\Omega} T_{ij} \approx p \sum_{(i,j)\in[d]\times[d]} T_{ij}, \tag{8.13}$$

where we use $\approx$ to mean that

$$\Big| \sum_{(i,j)\in\Omega} T_{ij} - p \sum_{(i,j)\in[d]\times[d]} T_{ij} \Big| < \epsilon \tag{8.14}$$

with high probability. This suggests the strategy of casting the estimation of our desired quantities in the form of estimating a scalar sum via a sample. In particular, we note that for any matrices A and B,

$$\langle A, P_\Omega(B) \rangle = \sum_{(i,j)\in\Omega} A_{ij} B_{ij} \approx p \langle A, B \rangle. \tag{8.15}$$

To make use of this observation to understand the quantities of interest ($\nabla f(x)$ and $\nabla^2 f(x)$), we compute the bilinear and quadratic forms for $\nabla f(x)$ and $\nabla^2 f(x)$ respectively:

$$\langle v, \nabla f(x) \rangle = \langle v, P_\Omega(M - xx^\top)x \rangle = \langle vx^\top, P_\Omega(M - xx^\top) \rangle, \tag{8.16}$$

where we have used the fact that $\langle A, BC \rangle = \langle AC^\top, B \rangle$. Also note that $vx^\top$ is a rank-1 matrix and $M - xx^\top$ is a rank-2 matrix.

$$\langle v, \nabla^2 f(x) v \rangle = \|P_\Omega(vx^\top + xv^\top)\|_F^2 - 2\langle P_\Omega(M - xx^\top), vv^\top \rangle \tag{8.17}$$

$$= \langle P_\Omega(vx^\top + xv^\top), vx^\top + xv^\top \rangle - 2\langle P_\Omega(M - xx^\top), vv^\top \rangle, \tag{8.18}$$

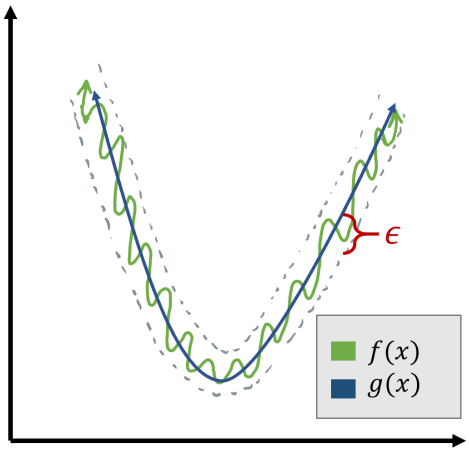

Figure 8.5: Even if $f(x)$ and $g(x)$ are no more than ϵ apart at any given x, without any additional knowledge, the local minima of f may possibly look dramatically different from the local minima of g. However, the proofs in this section show that the landscape of f (the matrix completion objective) and g (the PCA objective) are have similar properties by proving more advanced concentration inequalities.

where we have used the fact that $\|P_\Omega(A)\|_F^2 = \langle P_\Omega(A), P_\Omega(A)\rangle = \langle P(\Omega(A), A\rangle$.

The key lemma that applies the scalar concentration to these matrix quantities is as follows:

Lemma 8.17. *Let $\epsilon > 0$, $p = \frac{poly(\mu, \log d)}{d\epsilon^2}$. Given that $A = uu^\top, B = vv^\top$ for some u, v satisfying $\|u\|_2 \le 1$, $\|v\|_2 \le 1$, $\|u\|_\infty \le \mu/\sqrt{d}$, $\|v\|_\infty \le \mu/\sqrt{d}$, we have $|\langle P_\Omega(A), B\rangle/p - \langle A, B\rangle| \le \epsilon$ w.h.p.*

If we can show that g has no bad local minima via a proof that only uses g via terms of the form $\langle v, \nabla g(x)\rangle$ and $\langle v, \nabla^2 g(x) v\rangle$, then by Lemma 8.17 this proof will automatically generalize to f by concentration.

Next, we prove some facts about g and show the analogous proofs for f that we will use in the proof of Theorem 8.14.

Lemma 8.18 (Connecting inner product and norm for g)**.** *If x satisfies $\nabla g(x) = 0$, then $\langle x, z\rangle^2 = \|x\|_2^4$.*

Proof.

$$\nabla g(x) = 0 \implies \langle x, \nabla g(x)\rangle = 0 \tag{8.19}$$
$$\implies \langle x, (zz^\top - xx^\top)x\rangle = 0 \qquad \text{(because } \nabla g(x) = (M - xx^\top)x\text{)} \tag{8.20}$$
$$\implies \langle x, z\rangle^2 = \|x\|_2^4. \tag{8.21}$$

□

Lemma 8.19 (Connecting inner product and norm for f)**.** *Suppose $\|x\|_\infty \le 2\mu/\sqrt{d}$. If x satisfies $\nabla f(x) = 0$, then $\langle x, z\rangle^2 \ge \|x\|_2^4 - \epsilon$ with high probability.*

Proof.

$$\nabla f(x) = 0 \implies \langle x, \nabla f(x)\rangle = 0 \tag{8.22}$$
$$\implies \langle x, \nabla g(x)\rangle \approx \langle x, \nabla f(x)\rangle/p \pm \epsilon \qquad \text{(by Lemma 8.17)} \tag{8.23}$$
$$\implies |\langle x, (zz^\top - xx^\top)x\rangle| \le \epsilon \qquad \text{w.h.p.} \tag{8.24}$$
$$\implies \langle x, z\rangle^2 \ge \|x\|_2^4 - \epsilon \qquad \text{w.h.p.} \tag{8.25}$$

□

Lemma 8.20 (Bound norm for g)**.** *If $\nabla^2 g(x) \succeq 0$, then $\|x\|_2^2 \geq 1/3$.*

Proof.

$$\begin{aligned}
\nabla^2 g(x) \succeq 0 &\implies \langle z, \nabla^2 g(x) z\rangle \geq 0 && (8.26)\\
&\implies \|zx^\top + xz^\top\|_F^2 - 2z^\top(zz^\top - xx^\top)z \geq 0 && (8.27)\\
&\implies 2\|x\|_2^2 + 2\langle x, z\rangle^2 - 2 + 2\langle x, z\rangle^2 \geq 0 \qquad \text{(cyclic trace prop.)} && (8.28)\\
&\implies 3\|x\|_2^2 = \|x\|_2^2 + 2\|x\|_2^2 \geq \|x\|_2^2 + 2\langle x, z\rangle^2 \geq 1 \qquad \text{(by Cauchy-Schwarz)} && (8.29)\\
&\implies \|x\|_2^2 \geq 1/3. && (8.30)
\end{aligned}$$

□

Lemma 8.21 (Bound norm for f)**.** *Suppose $\|x\|_\infty \leq \mu/\sqrt{d}$. If $\nabla^2 f(x) \succeq 0$, then $\|x\|_2^2 \geq 1/3 - \epsilon/3$ with high probability.*

Proof.

$$\begin{aligned}
\nabla^2 f(x) \succeq 0 &\implies \langle z, \nabla^2 f(x) z\rangle \geq 0 && (8.31)\\
&\implies \langle z, \nabla^2 g(x) z\rangle \geq -\epsilon \qquad \text{w.h.p. (by Lemma 8.17)} && (8.32)\\
&\implies 3\|x\|_2^2 \geq 1 - \epsilon \qquad \text{w.h.p.} && (8.33)\\
&\implies \|x\|_2^2 \geq 1/3 - \epsilon/3 \qquad \text{w.h.p.} && (8.34)
\end{aligned}$$

□

Lemma 8.22 (g has no bad local minimum)**.** *All local minima of g are global minima.*

Proof.

$$\begin{aligned}
\nabla g(x) = 0 &\implies \langle z, \nabla g(x)\rangle = 0 && (8.35)\\
&\implies \langle z, (zz^\top - xx^\top)x\rangle = 0 && (8.36)\\
&\implies \langle x, z\rangle(1 - \|x\|_2^2) = 0. && (8.37)
\end{aligned}$$

Since $|\langle x, z\rangle| \geq 1/3 \neq 0$ (by Lemma 8.20), we must have $\|x\|_2^2 = 1$. But then Lemma 8.18 implies $\langle x, z\rangle^2 = \|x\|_2^4 = 1$, so $x = \pm z$ by Cauchy-Schwarz. □

We now prove Theorem 8.14, restated for convenience:

Theorem 8.23 (f has no bad local minimum)**.** *Assume $p = \dfrac{poly(\mu, \log d)}{d\epsilon^2}$. Then with high probability, all local minima of f are $O(\sqrt{\epsilon})$-close to $+z$ or $-z$.*

Proof. Observe that $\|x - z\|_2^2 = \|x\|_2^2 + \|z\|_2^2 - 2\langle x, z\rangle \leq \|x\|_2^2 + 1 - 2\langle x, z\rangle$. Our goal is to show that this quantity is small with high probability, hence we need to bound $\|x\|_2^2$ and $\langle x, z\rangle$ w.h.p. Note that the following bounds in this proof are understood to hold w.h.p.

Let x be such that $\nabla f(x) = 0$. For $\epsilon \leq 1/16$,

$$\begin{aligned}
\langle x, z\rangle^2 &\geq \|x\|_2^4 - \epsilon \qquad \text{(by Lemma 8.19)} && (8.38)\\
&\geq (1/3 - \epsilon/3)^2 - \epsilon \qquad \text{(by Lemma 8.21)} && (8.39)\\
&\geq 1/32. && (8.40)
\end{aligned}$$

With this, we can get a bound on $\|x\|_2^2$:

$$\begin{aligned}
\nabla f(x) = 0 &\implies \langle x, \nabla f(x)\rangle = 0 && (8.41)\\
&\implies |\langle z, \nabla g(x)\rangle| \le \epsilon && \text{(by Lemma 8.17)} \quad (8.42)\\
&\implies |\langle x, z\rangle| \cdot |1 - \|x\|_2^2| \le \epsilon && \text{(by dfn of } g) \quad (8.43)\\
&\implies |1 - \|x\|_2^2| \le 32\epsilon = O(\epsilon) && \text{(by (8.40))} \quad (8.44)\\
&\implies \|x\|_2^2 = 1 \pm O(\epsilon). && (8.45)
\end{aligned}$$

Next, we bound $\langle x, z\rangle$:

$$\begin{aligned}
\langle x, z\rangle^2 &\ge \|x\|_2^4 - \epsilon && \text{(by Lemma 8.19)} \quad (8.46)\\
&\ge (1 - O(\epsilon))^2 - \epsilon && \text{(by (8.45))} \quad (8.47)\\
&= 1 - O(\epsilon). && (8.48)
\end{aligned}$$

Finally, we put these quantities together to bound $\|x - z\|_2^2$. We have two cases:
Case 1: $\langle x, z\rangle \ge 1 - O(\epsilon)$. Then

$$\begin{aligned}
\|x - z\|_2^2 &= \|x\|_2^2 + \|z\|_2^2 - 2\langle x, z\rangle && (8.49)\\
&\le \|x\|_2^2 + 1 - 2\langle x, z\rangle && (8.50)\\
&\le 1 + O(\epsilon) + 1 - 2(1 - O(\epsilon)) && (8.51)\\
&\le O(\epsilon). && (8.52)
\end{aligned}$$

Hence we conclude x is $O(\sqrt{\epsilon})$-close to z.
Case 2: $\langle x, z\rangle \le -(1 - O(\epsilon))$. Then by an analogous argument, x is $O(\sqrt{\epsilon})$-close to $-z$. □

We have shown above that matrix completion of a rank-1 matrix has no spurious local minima. This proof strategy can be extended to handle higher-rank matrices and noisy matrices [Ge et al., 2016]. The proof also demonstrates a generally useful proof strategy: often, reducing a hard problem to an easy problem results in solutions that do not give much insight into the original problem, because the proof techniques do not generalize. It can often be fruitful to seek a proof in the simplified problem that makes use of a restricted set of tools that could generalize to the harder problem. Here we limited ourselves to only using $\langle v, \nabla g(x)\rangle$ and $\langle v, \nabla^2 g(x) v\rangle$ in the easy case; these quantities could then be easily converted to analogous quantities in f via the concentration lemma (Lemma 8.17).

8.3.3 Other problems where all local minima are global minima

We have now demonstrated that two classes of machine learning problems, rank-1 PCA and rank-1 matrix completion, have no spurious local minima and are thus amenable to being solvable by gradient descent methods. We now outline some major classes of problems for which it is known that there are no spurious local minima.

- Principal component analysis (covered in previous lecture).
- Matrix completion (and other matrix factorization problems). On a related note, it has also been shown that linearized neural networks of the form $y = W_1 W_2 x$, where W_1 and W_2 are optimized separately, have no spurious local minima [Baldi and Hornik, 1989]. It should be noted that linearized neural networks are not very useful in practice since the advantage of optimizing W_1 and W_2 separately versus optimizing a single $W = W_1 W_2$ is not clear.
- Tensor decomposition. The problem is as follows:

$$\text{maximize} \quad \sum_{i=1}^{d}\sum_{j=1}^{d}\sum_{k=1}^{d}\sum_{l=1}^{d} T_{ijkl} x_i x_j x_k x_l \quad \text{such that} \quad \|x\|_2 = 1. \tag{8.53}$$

Additionally, constraints are imposed on the tensor T to make the problem tractable. For example, one condition is that T must be a low-rank tensor with orthonormal components [Ge et al., 2015].

8.4 The Neural Tangent Kernel (NTK) Approach

In the previous sections, we studied non-convex optimization problems in which all local minima are global. Selecting the parameters of a deep neural network is another commonly encountered non-convex optimization problem, but it is unrealistic to expect that all local minima will also be global minima in this setting. Here we consider a particular objective for which we can identify particular regions of the input space in which all local minima are also global minima. We can show that this objective corresponds to certain types of deep neural networks, but this analysis remains limited. For further reading about this approach to studying neural network optimization, see [Liang et al., 2018] and [Du and Hu, 2019].

To be more formal, we take an appropriate parameter initialization θ^0 such that in a neighborhood around it, which we denote by $B(\theta^0)$, the loss function is convex and its global minimum is attained. Figure 8.6 depicts a function and region for which this condition holds.

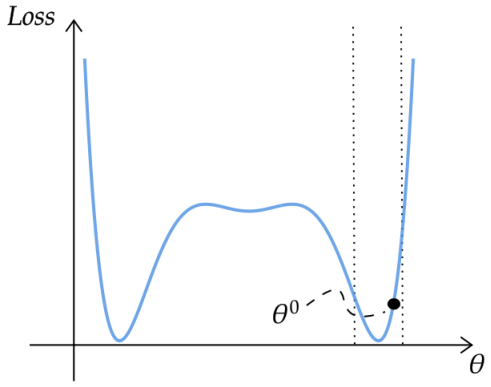

Figure 8.6: Training loss around an initialized θ^0. The dotted lines indicate $B(\theta^0)$, a region where the loss is convex, and where a global minimum exists.

Given a nonlinear $f_\theta(x)$, we examine the Taylor expansion at θ^0:

$$f_\theta(x) = \underbrace{f_{\theta^0}(x) + \langle \nabla_\theta f_{\theta^0}(x), \theta - \theta^0 \rangle}_{\triangleq g_\theta(x)} + \text{higher order terms} \tag{8.54}$$

Note that $g_\theta(x)$ is an affine function in θ, as $f_{\theta^0}(x)$ is a constant for fixed x, θ^0. Similarly, defining $\Delta\theta = \theta - \theta^0$, we can say that $g_\theta(x)$ is linear in $\Delta\theta$. For convenience, we will sometimes choose θ^0 such that $f_{\theta^0}(x) = 0$ for all x. It is easy to see why such an initialization exists. Consider splitting a two-layer neural network $f_\theta(x)$ with width $2m$ into two halves, each with m neurons; the outputs of these two networks are then given by $\sum_{i=1}^m a_i \sigma(w_i^\top x)$ and $\sum_{i=1}^m -a_i \sigma(w_i^\top x)$, respectively. Here, w_i can be randomly chosen so long as W_i is the same in both halves, and a_i can be randomly chosen as long as the other half is initialized with $-a_i$. Summing these two networks together yields $f_{\theta^0}(x) \equiv 0$ for all x.

When $f_{\theta^0}(x) \equiv 0$, we have that

$$g_\theta(x) = \langle \nabla_\theta f_{\theta^0}(x), \Delta\theta \rangle \,, \tag{8.55}$$

we observe that $\Delta\theta$ depends upon the parameter we evaluate the network at, while $\nabla_\theta f_{\theta^0}(x)$ can be thought of as a feature map since it is a fixed function of x (given the architecture and θ^0) that does not depend on θ whatsoever. We thus let $\phi(x) \triangleq \nabla_\theta f_{\theta^0}(x)$, which motivates the following definition:

Definition 8.24 (Neural Tangent Kernel)**.** For simplicity, we assume $f_{\theta^0}(x) = 0$ so that $y = y'$. The *neural tangent kernel* K is given by

$$K(x, x') = \langle \phi(x), \phi(x') \rangle \tag{8.56}$$
$$= \langle \nabla_\theta f_{\theta^0}(x), \nabla_\theta f_{\theta^0}(x') \rangle . \tag{8.57}$$

Here, the feature $\nabla_\theta f_{\theta^0}(x)$ is precisely the gradient of the neural network. This is where the "tangent" in Neural Tangent Kernel comes from.

Instead of $f_\theta(x)$, suppose we use the approximation $g_\theta(x)$, which we recall is linear in θ. The kernel method gives a linear model on top of features. When $\theta \approx \theta^0$, given a convex loss function ℓ, we have

$$\underbrace{\ell(f_\theta(x), y)}_{\substack{\text{not} \\ \text{necessarily} \\ \text{convex}}} \approx \underbrace{\ell(g_\theta(x), y)}_{\text{convex}} . \tag{8.58}$$

Convexity of the RHS follows from the fact that a convex function, ℓ, composed with a linear function, g_θ, is still convex.

A natural question to ask is: how valid is this approximation? We devote the rest of this chapter to answering this question. First, we define the empirical loss:

$$\hat{L}(f_\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell \left(f_\theta\big(x^{(i)}\big), y^{(i)} \right) \tag{8.59}$$

$$\hat{L}(g_\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell \left(g_\theta\big(x^{(i)}\big), y^{(i)} \right) . \tag{8.60}$$

The key idea is that the Taylor approximation works for certain cases. We defer a more complete enumeration of these cases to a later section of this monograph. Here we outline the high-level approach we take to validate and use this Taylor expansion. Namely, we will show that there exists a neighborhood around θ^0 called $B(\theta^0)$, such that we have the following:

1. Accurate approximation: $f_\theta(x) \approx g_\theta(x)$, and $\hat{L}(f_\theta) \approx \hat{L}(g_\theta)$ for all $\theta \in B(\theta^0)$.

2. It suffices to optimize in $B(\theta^0)$: There exists an approximate global minimum $\hat{\theta} \in B(\theta^0)$, so $\hat{L}(g_{\hat{\theta}}) \approx 0$. This is the lowest possible loss (because the loss is nonnegative), which implies we are close to the global minimum. Because of 1, this implies that $\hat{L}(f_{\hat{\theta}}) \approx 0$ as well. See Figure 8.7 for an illustration.

3. Optimizing $\hat{L}(f_\theta)$ is similar to optimizing $\hat{L}(g_\theta)$ and does not leave $B(\theta^0)$, i.e. everything is confined to this region. Intuitively, this last point to some extent is "implied" by (1) and (2), but this claim still requires a formal proof.

Note (1), (2), and (3) can all be true in various settings. In particular, to attain all three, we will require:

(a) Overparametrization and/or a particular scaling of the initialized θ^0.

(b) Small (or even zero) stochasticity, so θ never leaves $B(\theta^0)$. This condition is guaranteed by a small learning rate or full-batch gradient descent.

Despite the limitations of the requirements of (a) and (b), the existence of such a region is still surprising. Given the loss landscape which could potentially be highly non-convex, it is striking to find a neighborhood where the loss function is convex (e.g. quadratic) with a global minimum. This suggests there is some flexibility in the loss landscape.

To begin our formal discussion, we start by providing tools for proving (1) and (2). Let

$$\phi^{(i)} = \phi(x^{(i)}) = \nabla_\theta f_{\theta^0}(x^{(i)}) \in \mathbb{R}^p \tag{8.61}$$

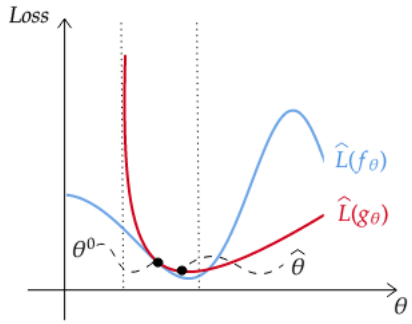

Figure 8.7: Here, $\hat{L}(g_\theta)$ and $\hat{L}(f_\theta)$ are both plotted. At $\hat{\theta}$, we have reached the approximate global minimum where $\hat{L}(g_{\hat{\theta}}) \approx 0$, in turn implying also that $\hat{L}(f_{\hat{\theta}}) \approx 0$.

and

$$\Phi = \begin{bmatrix} \phi^{(1)\top} \\ \vdots \\ \phi^{(n)\top} \end{bmatrix} \in \mathbb{R}^{n \times p} \tag{8.62}$$

where p is the number of parameters. Taking the quadratic loss, we have

$$\hat{L}(g_\theta) = \frac{1}{n} \sum_{i=1}^{n} \left(y^{(i)} - \phi\left(x^{(i)}\right)^\top \Delta\theta \right)^2 = \frac{1}{n} \|\vec{y} - \Phi \cdot \Delta\theta\|_2^2 \tag{8.63}$$

where $\vec{y} = \left[y^{(1)}, \cdots, y^{(n)}\right]^\top \in \mathbb{R}^n$. Note that this looks a lot like linear regression, where Φ and $\Delta\theta$ are the analogues of the design matrix and parameter, respectively. We further assume that $y^{(i)} = O(1)$ and $\|y\|_2 = O(\sqrt{n})$. Now, we can prove a lemma that addresses the second of the three conditions we described above, i.e. that it is sufficient to optimize in some small ball around θ^0.

Lemma 8.25 (for (2))**.** *Suppose we are in the setting where $p \geq n$, $\operatorname{rank}(\Phi) = n$, and $\sigma_{\min}(\Phi) = \sigma > 0$. Then, letting $\Delta\hat{\theta}$ denote the minimum norm solution, i.e. the nearest global minimum, of $\vec{y} = \Phi\Delta\theta$, we have*

$$\|\Delta\hat{\theta}\|_2 \leq O(\sqrt{n}/\sigma) \tag{8.64}$$

Remark 8.26. The meaning of the bound on $\Delta\hat{\theta}$ becomes clear if we consider the ball given by

$$B_{\theta^0} = \{\theta = \theta^0 + \Delta\theta : \|\Delta\theta\|_2 \leq O(\sqrt{n}/\sigma)\}. \tag{8.65}$$

In particular, notice that B_{θ^0} contains a global minimum, so this lemma characterizes how large the ball must be to contain a global minimum.

Remark 8.27. We also note that the condition $\operatorname{rank}(\Phi) = n$ and $\sigma > 0$ can be thought of as a "finite-sample expressivity" condition, saying that the features Φ are expressive enough so that there exists a linear model on top of these features that perfectly fit the data. The condition $\operatorname{rank}(\Phi) = n$ requires $p \geq n$—so we need some amount of over-parameterization to apply these analysis.

Proof. Letting Φ^+ denote the Moore-Penrose pseudoinverse of Φ, note that $\Delta\hat{\theta} = \Phi^+\boldsymbol{y}$, and $\|\Phi^+\|_{\text{op}} = \frac{1}{\sigma_{\min}(\Phi)} = \frac{1}{\sigma}$. A simple argument shows

$$\|\Delta\hat{\theta}\|_2 \leq \|\Phi^+\|_{\text{op}} \cdot \|\vec{y}\|_2 \tag{8.66}$$

$$\leq O\left(\frac{1}{\sigma} \cdot \sqrt{n}\right), \tag{8.67}$$

where the last inequality follows from the assumption that $\|\vec{y}\|_2 \leq O(\sqrt{n})$. □

Next, we prove a lemma that addresses the first of the three steps we described above.

Lemma 8.28 (for (1))**.** *Suppose $\nabla_\theta f_\theta(x)$ is β-Lipschitz in θ, i.e. for every x, and θ, θ', we have*

$$\|\nabla_\theta f_\theta(x) - \nabla_\theta f_{\theta'}(x)\|_2 \leq \beta \cdot \|\theta - \theta'\|_2. \tag{8.68}$$

Then,

$$|f_\theta(x) - g_\theta(x)| \leq O\left(\beta \|\Delta\theta\|_2^2\right). \tag{8.69}$$

If we further restrict our choice of θ using B_{θ^0} as defined in Remark 8.26, we obtain that

$$|f_\theta(x) - g_\theta(x)| \leq O\left(\frac{\beta n}{\sigma^2}\right), \quad \forall \theta \in B_{\theta^0}. \tag{8.70}$$

Proof. The proof comes from the following fact: if $h(\theta)$ is such that $\nabla h(\theta)$ is β-Lipschitz (which if differentiable is equivalent to $\|\nabla^2 h(\theta)\|_{\text{op}} \leq \beta$), then

$$\left| \underbrace{h(\theta)}_{f_\theta(x)} \underbrace{- h(\theta^0) - \left\langle \nabla h(\theta^0), \theta - \theta^0 \right\rangle}_{-g_\theta(x)} \right| \leq O\left(\beta \|\theta - \theta^0\|_2^2\right). \tag{8.71}$$

As shown above, the proof is as simple as plugging in $f_\theta(x) = h(\theta)$ and $g_\theta(x) = h(\theta^0) + \left\langle \nabla h(\theta^0), \Delta\theta \right\rangle$. □

Remark 8.29. The lemma above bounds the approximation error. Intuitively, as you move farther away from θ^0, the Taylor approximation gets worse; the approximation error is bounded above by a second order $\Delta\theta$ term.

Remark 8.30. Note that if f_θ involves a relu function, then ∇f_θ is not continuous everywhere. This requires a technical fix outside the scope of our discussion.[2]

8.4.1 Two examples of the NTK regime

By (8.70), we have now established a bound on our approximation error, but we have yet to analyze how good it is, as $\beta n/\sigma^2$ is neither obviously either big nor small. An important fact to notice is that β/σ^2 is not scaling invariant, so we can play with the scaling in order to drive this term to 0. In particular, there are two notable cases (with specific parameterization, initialization, etc) where $\beta/\sigma^2 \to 0$. In the literature, such situation is often referred to as the NTK regime or the lazy training regime [Chizat and Bach, 2018].

1. **Reparameterize with a scalar** [Chizat and Bach, 2018]. Let $f_\theta(x) = \alpha \cdot \bar{f}_\theta(x)$ where $\bar{f}_\theta(x)$ is an arbitrary neural net with fixed width and depth. We only vary α, i.e. the scaling, and we see how the crucial quantity β/σ^2 changes accordingly. Fix an initial θ^0, and let

$$\bar{\sigma} = \sigma_{\min}\left(\begin{bmatrix} \nabla_\theta \bar{f}_{\theta^0}\left(x^{(1)}\right)^\top \\ \vdots \\ \nabla_\theta \bar{f}_{\theta^0}\left(x^{(n)}\right)^\top \end{bmatrix}\right). \tag{8.72}$$

   Furthermore, let $\bar{\beta}$ be the Lipschitz parameter of $\nabla_\theta \bar{f}_\theta(x)$ in θ. A simple chain-rule gradient argument shows that scaling $\bar{f}_\theta$ by α also scales σ and β accordingly, i.e. $\sigma = \alpha\bar{\sigma}$, and $\beta = \alpha\bar{\beta}$. Some straightforward algebra yields

$$\frac{\beta}{\sigma^2} = \frac{\bar{\beta}}{\bar{\sigma}^2} \cdot \frac{1}{\alpha} \to 0 \quad \text{as} \quad \alpha \to \infty. \tag{8.73}$$

   Once α becomes big enough, then by Lemma 8.28, the approximation $|f_\theta(x) - g_\theta(x)| \leq O\left(\beta n/\sigma^2\right)$ becomes very good.

[2] A relu function is continuous almost everywhere, so we can make some minor fixes and still use some modified notion of Lipschitzness to derive an upper bound.

Remark 8.31. A priori, such a phenomenon may appear to be too good to be true. To understand it better, we first note that this re-parameterizaton does not change the scale of the loss, but rather change the shape of the loss function. Intuitively, as α becomes larger, the function f_θ becomes sharper and more non-smooth (leading to higher approximation error). However, on the other hand, we note that we only need to travel a little bit away from θ^0 to find a global minimum given that there is a global minimum within radius $O(\sqrt{n}/\sigma)$. It turns out that the radius needed shrinks faster than the smoothness grows.

To visualize this effect, we can consider the following example with only 1 data point with 1-dimensional input $(x, y) = (1, 1)$ and the quadratic model $\bar{f}_\theta(x) = x(\theta + \beta\theta^2) = \theta + \beta\theta^2$. Using the squared loss, we have

$$\widehat{L}(\bar{f}_\theta) = (1 - (\theta + \beta\theta^2))^2 \tag{8.74}$$

Let $\theta^0 = 0$. Taylor expanding at θ^0 gives the linear approximation $\bar{g}_\theta(x) = \theta x = \theta$, and the resulting loss function that is quadratic

$$\widehat{L}(\bar{g}_\theta) = (1 - \theta)^2 \tag{8.75}$$

In this case, $\nabla f_{\theta^0}(x) = 2\beta\theta x = 2\beta\theta$ is 2β-Lipschitz, and $\sigma = 1$.

Now we vary α and get

$$\widehat{L}(\alpha\bar{f}_\theta) = (1 - \alpha(\theta + \beta\theta^2))^2 \tag{8.76}$$

and

$$\widehat{L}(\alpha\bar{g}_\theta) = (1 - \alpha\theta)^2 \tag{8.77}$$

Note that the minimizer of $\widehat{L}(\alpha\bar{g}_\theta)$ is $1/\alpha$, which is closer to θ^0 as $\alpha \to \infty$. We zoom into the region $[0, 1/\alpha]$ and find out the difference between $\alpha\bar{f}_\theta$ and $\alpha\bar{g}_\theta$ is $\alpha\beta\theta^2 \le \beta/\alpha$, which is much smaller than the value of $\alpha\bar{g}_\theta \approx O(1)$.

We visualize the these functions in Figure 8.8. We observe that $\widehat{L}(\alpha\bar{g}_\theta)$ becomes a better approximation of $\widehat{L}(\alpha\bar{f}_\theta)$ in the region $[0, 1/\alpha]$ as $\alpha \to \infty$ (though $\widehat{L}(\alpha\bar{g}_\theta)$ is a worse approximation of $\widehat{L}(\alpha\bar{f}_\theta)$ globally.)

2. **Overparametrization (with specific initialization)**. Early papers on the NTK take this approach (e.g., [Li and Liang, 2018, Du and Hu, 2019]). Consider a two-layer network with m neurons.

$$\hat{y} = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} a_i \sigma(w_i^\top x) \tag{8.78}$$

The scaling $1/\sqrt{m}$ is to ensure that a random initialization with constant scale will have output on the right order, as we see momentarily. We make the following assumptions regarding the network and its inputs.

$$W = \begin{bmatrix} w_1^\top \\ \vdots \\ w_m^\top \end{bmatrix} \in \mathbb{R}^{m \times d} \tag{8.79}$$

$$\sigma \text{ is 1-Lipschitz and twice-differentiable} \tag{8.80}$$

$$a_i \sim \{\pm 1\} \qquad \text{(not optimized)} \tag{8.81}$$

$$w_i^0 \sim \mathcal{N}(0, I_d) \tag{8.82}$$

$$\|x\|_2 = \Theta(1) \tag{8.83}$$

$$\theta = \text{vec}(W) \in \mathbb{R}^{dm} \qquad \text{(vectorized } W\text{)} \tag{8.84}$$

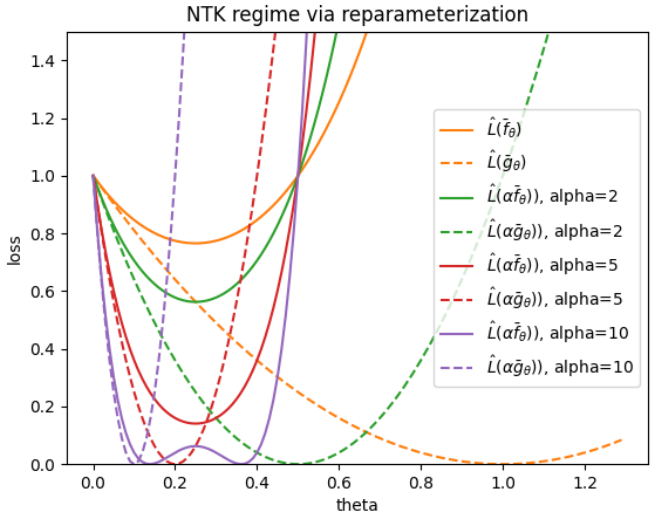

Figure 8.8: The approximation $\widehat{L}(\alpha\bar{g}_\theta)$ becomes a better approximation of $\widehat{L}(\alpha\bar{f}_\theta)$ in the region $[0, 1/\alpha]$ as $\alpha \to \infty$ (though $\widehat{L}(\alpha\bar{g}_\theta)$ is a worse approximation of $\widehat{L}(\alpha\bar{f}_\theta)$ globally).

We will assume $m \to \infty$ polynomially in n and d. In particular, for fixed n, d, we have $m = \mathsf{poly}(n, d)$. Why do we use the $1/\sqrt{m}$ scaling? Note that $\sigma\left(w_i^{0\top} x\right) \approx 1$ because $\|x\|_2 = \Theta(1)$ and w_i^0 is drawn from a spherical Gaussian. Thus, as some a_i are positive and others are negative, $\left|\sum_{i=1}^m a_i \sigma\left(w_i^{0\top} x\right)\right| = \Theta\left(\sqrt{m}\right)$, and finally $f_{\theta^0}(x) = \Theta(1)$.

Now we analyze σ and β. We let

$$\sigma = \sigma_{\min}(\Phi) = \sqrt{\sigma_{\min}\left(\Phi\Phi^\top\right)} \tag{8.85}$$

where

$$\left(\Phi\Phi^\top\right)_{ij} = \left\langle \nabla_\theta f_{\theta^0}\left(x^{(i)}\right), \nabla_\theta f_{\theta^0}\left(x^{(j)}\right) \right\rangle \tag{8.86}$$

Note that the gradient with respect to w_i is given by

$$\frac{\partial f_\theta(x)}{\partial w_i} = \frac{1}{\sqrt{m}} \sigma'(w_i^\top x) \cdot x \tag{8.87}$$

Now observe that

$$\|\nabla f_\theta(x)\|_2^2 = \frac{1}{m} \sum_{i=1}^m \|\sigma'\left(w_i{}^\top x\right) \cdot x\|_2^2 \tag{8.88}$$

$$= \frac{1}{m} \|x\|_2^2 \cdot \sum_{i=1}^m \left(\sigma'\left(w_i{}^\top x\right)\right)^2 \tag{8.89}$$

$$\to \mathop{\mathbb{E}}_{w \sim \mathcal{N}(0, I_d)} \left[\sigma'\left(w^\top x\right)^2\right] \cdot \|x\|_2^2 \quad \text{as} \quad m \to \infty \tag{8.90}$$

$$= O(1) \qquad \text{(not depending on } m\text{)} \tag{8.91}$$

where the penultimate line follows from the law of large numbers, as $\frac{1}{m}\sum_{i=1}^m \left(\sigma'(w_i^\top x)\right)^2$ can be interpreted as a mean.

Note that the scale of $\|\nabla_\theta f_{\theta^0}(x)\|_2$ does not depend on m, so the inner product in (8.86) also does not depend on m either. As above, we can show

$$\langle \nabla_\theta f_{\theta^0}(x), \nabla_\theta f_{\theta^0}(x') \rangle = \frac{1}{m} \langle x, x' \rangle \sum_{i=1}^{m} \sigma'(w^\top x)\sigma'(w^\top x') \tag{8.92}$$

$$\to \mathop{\mathbb{E}}_{w \sim \mathcal{N}(0, I_d)} \left[\sigma'(w^\top x)\sigma'(w^\top x') \right] \langle x, x' \rangle \tag{8.93}$$

(8.93) implies that as $m \to \infty$, $\Phi\Phi^\top$ converges to a constant matrix denoted by

$$K^\infty = \lim_{m \to \infty} \Phi\Phi^\top \tag{8.94}$$

This is precisely the NTK with $m = \infty$. Though we omit the proof of this claim, it can be shown that K^∞ is full rank. Then, let

$$\sigma_{\min} \triangleq \sigma_{\min}(K^\infty) > 0. \tag{8.95}$$

We can show that

$$\sigma = \sigma_{\min}\left(\Phi\Phi^\top\right) > \frac{1}{2}\sigma_{\min} \tag{8.96}$$

Intuitively, $\Phi\Phi^\top \to K^\infty$, so the spectrum of the matrix should also converge. Thus, in some sense, we have shown that σ is constant in the limit.

Now what about β? If we can show $\beta \to 0$ as $m \to \infty$, we are done. We begin by analyzing this key expression:

$$\nabla_\theta f_\theta(x) - \nabla_\theta f_{\theta'}(x) = \left[\frac{1}{\sqrt{m}} \left(\sigma'\left(w_i^\top x\right) - \sigma'\left(w_i'^\top x\right) \right) \cdot x \right]_{i=1}^{m} \tag{8.97}$$

Note that (8.97) above consists of matrices, as θ is a vectorized matrix. Then,

$$\|\nabla_\theta f_\theta(x) - \nabla_\theta f_{\theta'}(x)\|_2^2 = \frac{1}{m} \sum_{i=1}^{m} \|x\|_2^2 \left(\sigma'\left(w_i^\top x\right) - \sigma'\left(w_i'^\top x\right) \right)^2 \tag{8.98}$$

$$\leq O\left(\frac{1}{m} \sum_{i=1}^{m} \|x\|_2^2 \left(w_i^\top x - w_i'^\top x\right)^2 \right) \tag{8.99}$$

$$= O\left(\frac{1}{m} \sum_{i=1}^{m} \|w_i - w_i'\|_2^2 \right) \tag{8.100}$$

$$= O\left(\frac{1}{m} \|\theta - \theta'\|_2^2 \right) \tag{8.101}$$

The first line follows from the fact that $\frac{1}{\sqrt{m}} \left(\sigma'\left(w_i^\top x\right) - \sigma'\left(w_i'^\top x\right) \right)$ is a scalar. The second line uses the assumption that σ' is $O(1)$-Lipschitz. The third line uses Cauchy-Schwarz and the fact that $\|x\|_2^2 \approx 1$. Taking the square root, we have that

$$\|\nabla_\theta f_\theta(x) - \nabla_\theta f_{\theta'}(x)\|_2 \lesssim \frac{1}{\sqrt{m}} \|\theta - \theta'\|_2 \tag{8.102}$$

Thus, the Lipschitz parameter is $\beta = O(1/\sqrt{m})$. Thus, our key quantity β/σ^2 goes to 0 as m grows. Namely,

$$\frac{\beta}{\sigma^2} \approx \frac{1}{\sqrt{m}} \cdot \frac{1}{\sigma_{\min}^2} \to 0 \quad \text{as} \quad m \to \infty. \tag{8.103}$$

Recall here that $\sigma_{\min}$ does not depend on m. Concretely, this result tells us that our function becomes more smooth (the gradient has a smaller Lipschitz constant) as we add more neurons.

8.4.2 Optimizing $\hat{L}(g_\theta)$ vs. $\hat{L}(f_\theta)$

We now discuss how to establish the last of the three conditions under which we claimed a Taylor approximation is reasonable. We need to show that optimizing $\hat{L}(f_\theta)$ is similar to optimizing $\hat{L}(g_\theta)$. To do so, we require two steps:

(A) Analyze optimization of $\hat{L}(g_\theta)$.

(B) Analyze optimization of $\hat{L}(f_\theta)$ by re-using or modifying the proofs in (A).

There are two approaches in the literature for (A), which implies that there exist two approaches for (B) as well.

(i) We leverage the strong convexity of $\hat{L}(g_\theta)$, and then show an exponential convergence rate.[3]

(ii) Instead of strong convexity, we rely on the smoothness of f_θ (i.e. bounded second derivative).

We will only discuss the first of these two methods in the sequel.

Remark 8.32. In both either approach (i) or (ii), we will implicitly or explicitly use the following simple fact. Suppose at any θ^t, we take the Taylor expansion of f_θ at θ^t:

$$g_\theta^t(x) = f_{\theta^t}(x) + \left\langle \nabla f_{\theta^t}(x), \theta - \theta^t \right\rangle \tag{8.105}$$

Consider the gradient we are interested in taking: $\nabla \hat{L}(f_{\theta^t})$. Notice that:

$$\nabla \hat{L}(f_{\theta^t}) = \nabla \hat{L}(g_{\theta^t}^t) \tag{8.106}$$

This is really saying that f_θ and g_θ^t agree up to first-order at θ^t. This implies that $L(f_\theta)$ and $L(g_\theta^t)$ also agree to first-order at θ^t. This also means that T steps of gradient descent on $\hat{L}(f_\theta)$ is the same as performing online gradient descent[4] on a sequence of changing objectives $L(g_\theta^0), \ldots, L(g_\theta^T)$, and this online learning perspective is useful in the approach (ii).

We will now show that under the strong convexity regime, optimizing a neural network f_θ is equivalent to optimizing a linear model g_θ. We will also observe that this regime is not particularly practically relevant, but this analysis is nevertheless of interest to us for two reasons. First, the approach used in the subsequent exposition is of technical interest and second, it remains quite interesting that optimizing f_θ and optimization g_θ yields the same results under *any* regime.

Optimizing g_θ

We relate the optimization of g_θ to performing linear regression. Recall that we can think of $\nabla f_{\theta^0}(x)$ as a feature map. Then, the problem of choosing $\Delta\theta$ to get $g_\theta(x)$ to be close to $\vec{y}$ is a linear regression. In particular, we use gradient descent to minimize

$$\|\vec{y} - \Phi\Delta\theta\|_2^2, \tag{8.107}$$

where

$$\Phi = \begin{bmatrix} \nabla f_{\theta^0}(x^{(1)})^\top \\ \vdots \\ \nabla f_{\theta^0}(x^{(n)})^\top \end{bmatrix} \in \mathbb{R}^{n \times p}. \qquad \vec{y} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix} \in \mathbb{R}^n \tag{8.108}$$

[3]Recall that a differentiable function f is μ-strongly convex if

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{\mu}{2}\|y - x\|_2^2 \tag{8.104}$$

for some $\mu > 0$ and all x, y.

[4]Online gradient descent is the algorithm that takes one gradient descent step upon receiving a new objective function. See Chapter 11 for more discussions about online learning.

For learning rate η, the gradient descent update rule is

$$\Delta\theta^{t+1} = \Delta\theta^t - \eta\Phi^\top(\Phi\Delta\theta^t - \vec{y}). \tag{8.109}$$

This analysis considers changes in the output space. Define $\hat{y}^t = \Phi\Delta\theta^t$. Then, we're interested in changes in

$$\begin{aligned}
\hat{y}^{t+1} - \vec{y} &= \Phi\Delta\theta^{t+1} - \vec{y} && (8.110)\\
&= \Phi\left(\Delta\theta^t - \eta\Phi^\top(\Phi\Delta\theta^t - \vec{y})\right) - \vec{y} && \text{(by (8.109))} \quad (8.111)\\
&= \left(\Phi - \eta\Phi\Phi^\top\Phi\right)\Delta\theta^t - (I - \eta\Phi\Phi^\top)\vec{y} && (8.112)\\
&= (I - \eta\Phi\Phi^\top)\Phi\Delta\theta^t - (I - \eta\Phi\Phi^\top)\vec{y} && (8.113)\\
&= (I - \eta\Phi\Phi^\top)(\Phi\Delta\theta^t - \vec{y}) && (8.114)\\
&= (I - \eta\Phi\Phi^\top)(\hat{y}^t - \vec{y}). && (8.115)
\end{aligned}$$

From this decomposition, we see that the residuals, $\hat{y}^t - \vec{y}$, are monotonically shrinking since $\eta\Phi\Phi^\top$, i.e. the term we are subtracting from I in (8.115), is positive semidefinite. Next, we quantify how quickly we are shrinking the residuals. Define

$$\begin{aligned}
\tau^2 &= \sigma_{\max}(\Phi\Phi^\top) && (8.116)\\
\sigma &= \sigma_{\min}(\Phi) = \sqrt{\sigma_{\min}(\Phi\Phi^\top)}. && (8.117)
\end{aligned}$$

Then, we claim that when $\eta \le \frac{1}{\tau^2}$,

$$\|I - \eta\Phi\Phi^\top\|_{\text{op}} \le 1 - \eta\sigma^2. \tag{8.118}$$

Why? Let the eigenvalues of $\Phi\Phi^\top$ be (in descending order) $\tau_1^2, \ldots, \tau_n^2$. By definition, $\tau_1^2 = \tau^2$ and $\tau_n^2 = \sigma^2$. Now, given the singular value decomposition, $\Phi = U\Sigma V^\top$, we obtain the eigendecomposition:

$$\begin{aligned}
I - \eta\Phi\Phi^\top &= I - \eta U\Sigma^2 U^\top && (8.119)\\
&= UU^\top - \eta U\Sigma^2 U^\top && (8.120)\\
&= U(I - \eta\Sigma^2)U^\top. && (8.121)
\end{aligned}$$

(8.121) is the eigendecomposition of $I - \eta\Phi\Phi^\top$, so $I - \eta\Phi\Phi^\top$ has eigenvalues $1 - \eta\tau_1^2, \ldots, 1 - \eta\tau_n^2$. Note that assuming $\eta \le \frac{1}{\tau^2}$ ensures that all eigenvalues of $I - \eta\Phi\Phi^\top$ are non-negative. Thus,

$$\begin{aligned}
\|I - \eta\Phi\Phi^\top\|_{\text{op}} &\le \max_j |1 - \eta\tau_j^2| && (8.122)\\
&= 1 - \eta\tau_n^2 && (8.123)\\
&= 1 - \eta\sigma^2, && (8.124)
\end{aligned}$$

where the non-negativity of $1 - \eta\tau_j^2$ for all j implies (8.123).

Using this result, we obtain our desired result. Namely, assuming $\eta \le \frac{1}{\tau^2}$,

$$\begin{aligned}
\|\hat{y}^{t+1} - \vec{y}\|_2 &= \|I - \eta\Phi\Phi^\top\|_{\text{op}} \cdot \|\hat{y}^t - \vec{y}\|_2 && (8.125)\\
&\le (1 - \eta\sigma^2)\|\hat{y}^t - \vec{y}\|_2 && (8.126)\\
&\le (1 - \eta\sigma^2)^{t+1}\|\hat{y}^0 - \vec{y}\|_2. && (8.127)
\end{aligned}$$

This yields the desired exponential decay in the error. Thus, after $T = O\left(\frac{\log 1/\epsilon}{\eta\sigma^2}\right)$ iterations,

$$\|\hat{y}^T - \vec{y}\|_2 \le \epsilon\|\hat{y}^0 - \vec{y}\|_2. \tag{8.128}$$

Optimizing f_θ

We now transition to an analysis of the optimization of f_θ. Our key result is Theorem 8.33. If we compare it against what we have in (8.128), we see the claimed similarity between f_θ and g_θ in error decay under optimization.

Theorem 8.33. *There exists a constant $c_0 \in (0,1)$ such that for $\frac{\beta}{\sigma^2} \leq \frac{c_0}{n}$ and sufficiently small η (which could depend on β, σ, or p), $\hat{L}\left(f_{\theta^T}\right) \leq \epsilon$ after $T = O\left(\frac{\log 1/\epsilon}{\eta\sigma^2}\right)$ steps.*

Proof. (This is actually a proof sketch that elides a few technical details for the sake of a simpler exposition.) Our approach is to follow the preceding analysis of g_θ, making changes where necessary.

Let

$$\Phi^t = \begin{bmatrix} \nabla f_{\theta^t}(x^{(1)})^\top \\ \vdots \\ \nabla f_{\theta^t}(x^{(n)})^\top \end{bmatrix} \in \mathbb{R}^{n \times p}. \tag{8.129}$$

To obtain our gradient descent update rule, we find, using the chain rule,

$$\nabla \hat{L}\left(f_{\theta^t}\right) = \sum_{i=1}^{n} \left(f_{\theta^t}\left(x^{(i)}\right) - y^{(i)}\right) \nabla f_{\theta^t}\left(x^{(i)}\right) \tag{8.130}$$

$$= \sum_{i=1}^{n} \left(\hat{y}^{(i),t} - y^{(i)}\right) \nabla f_{\theta^t}\left(x^{(i)}\right) \tag{8.131}$$

$$= \left(\Phi^t\right)^\top \left(\hat{y}^t - \vec{y}\right). \tag{8.132}$$

This results in the policy

$$\theta^{t+1} = \theta^t - \eta \nabla \hat{L}\left(f_{\theta^t}\right) \tag{8.133}$$

$$= \theta^t - \eta (\Phi^t)^\top \left(\hat{y}^t - \vec{y}\right) \tag{8.134}$$

$$= \theta^t - \eta b^t, \tag{8.135}$$

where we have let $b^t = (\Phi^t)^\top (\hat{y}^t - \vec{y})$. Following our treatment of g_θ, we want to express $\hat{y}^{t+1}$ as a function of $\hat{y}^t$. The challenge now is that f is nonlinear. To deal with this, we Taylor expand f_θ at θ_t:

$$f_{\theta^{t+1}}(x^{(i)}) = f_{\theta^t}(x^{(i)}) + \left\langle \nabla f_{\theta^t}(x^{(i)}), \theta^{t+1} - \theta^t \right\rangle + \text{high order terms} \tag{8.136}$$

$$= f_{\theta^t}(x^{(i)}) + \left\langle \nabla f_{\theta^t}(x^{(i)}), -\eta b^t \right\rangle + O\left(\|\theta^{t+1} - \theta^t\|_2^2\right). \tag{8.137}$$

Since $O\left(\|\theta^{t+1} - \theta^t\|_2^2\right)$ is $O\left(\eta^2\right)$, we can ignore this term as $\eta \to 0$. Vectorizing (8.137) without $O\left(\|\theta^{t+1} - \theta^t\|_2^2\right)$,

$$\hat{y}^{t+1} = \hat{y}^t - \eta \Phi^t b^t \tag{8.138}$$

$$= \hat{y}^t + \eta \Phi^t \left(\Phi^t\right)^\top (\vec{y} - \hat{y}^t). \tag{8.139}$$

Subtracting $\vec{y}$ and re-arranging,

$$\hat{y}^{t+1} - \vec{y} = \hat{y}^t - \vec{y} + \eta \Phi^t \left(\Phi^t\right)^\top (\vec{y} - \hat{y}^t) \tag{8.140}$$

$$= \left(I - \eta \Phi^t \left(\Phi^t\right)^\top\right) \left(\hat{y}^t - \vec{y}\right). \tag{8.141}$$

Comparing (8.141) with (8.115), we see one difference: in (8.141), our convergence depends on $\eta\Phi^t\left(\Phi^t\right)^\top$, which is a matrix that changes as we iterate, whereas in (8.115), convergence is controlled by a matrix that is fixed as we iterate.

To understand the convergence implications of (8.141), we examine the eigenvalues of $I-\eta\Phi^t\left(\Phi^t\right)^\top$. For now, suppose

$$\|\theta^t-\theta^0\|_2 \leq \sigma/(4\sqrt{n}\beta) \tag{8.142}$$

at time t. This implies that $\|\Phi^t-\Phi\|_F \leq \frac{\sigma}{4}$ by the Lipschitzness of $\nabla f_\theta(x)$ in θ. Then, we claim that

$$\sigma_{\min}(\Phi^t) \geq 3\sigma/4. \tag{8.143}$$

Why does (8.143) hold? Observe that

$$\begin{aligned}\sigma_{\min}(\Phi^t) &= \min_{\|x\|_2=1} x^\top\Phi^t x &(8.144)\\ &\geq \min_{\|x\|_2=1} x^\top(\Phi^t-\Phi)x + \min_{\|x\|_2=1} x^\top\Phi x. &(8.145)\end{aligned}$$

We can lower bound the first term of (8.145) as follows:

$$\begin{aligned}x^\top(\Phi^t-\Phi)x &\geq -|\left\langle x,(\Phi^t-\Phi)x\right\rangle| &(8.146)\\ &\geq -\|x\|_2\|(\Phi^t-\Phi)x\|_2 \qquad \text{(Cauchy-Schwarz)} &(8.147)\\ &\geq -\|\Phi^t-\Phi\|_2 \qquad (\|x\|_2=1) &(8.148)\\ &\geq -\sigma/4 \qquad \text{(Lipschitzness of } \Phi\text{)}. &(8.149)\end{aligned}$$

Next, we note that the second term of (8.145) is lower bounded by σ by simplifying and applying the definition of σ given in (8.117). Combining this observation with (8.149), we conclude that (8.143) must hold.

Applying this lower bound on the eigenvalues of Φ^t, we can use the same argument we used to establish (8.118) to conclude that

$$\|I-\eta\Phi^t\left(\Phi^t\right)^\top\|_{\text{op}} \leq 1-3\eta\sigma/4, \tag{8.150}$$

and

$$\|\hat{y}^{t+1}-\vec{y}\|_2 \leq (1-3\eta\sigma/4)^{t+1}\|\hat{y}^0-\vec{y}\|_2. \tag{8.151}$$

So, as desired, we see exponential decay in the error at each iteration and after $T=O\left(\frac{\log 1/\epsilon}{n\sigma^2}\right)$ iterations,

$$\hat{L}(f_{\theta^T}) \leq \epsilon. \tag{8.152}$$

To complete our proof, observe that this argument is predicated upon the assumption that $\|\theta^t-\theta^0\|_2 \leq \sigma/(4\sqrt{n}\beta)$. This assumption is reasonable, however, given what we have already proven. Recall that in Lemma 8.25, we proved that

$$\|\Delta\hat{\theta}\|_2 = \|\hat{\theta}-\theta^0\|_2 \lesssim \sqrt{n}/\sigma. \tag{8.153}$$

Thus, when $\beta/\sigma^2 \to 0$, eventually, $\sqrt{n}/\sigma \ll \sigma/(4\sqrt{n}\beta)$. To extend this to $\|\hat{\theta}-\theta^t\|_2$ for arbitrary t, we heuristically argue that since the empirical minimizer is within $\sigma/(4\sqrt{n}\beta)$ of θ^0, we would not expect to have traveled more than $\sigma/(4\sqrt{n}\beta)$ from θ^0 at *any* iteration.

More formally, we claim that for all $t\in\mathbb{N}$,

$$\|\hat{y}^t-\vec{y}\|_2 \leq \mathcal{O}(\sqrt{n}). \tag{8.154}$$

We proceed via induction. For $t = 0$, because each element of $\hat{y}$ is of order 1, we know that:

$$\frac{1}{\sqrt{n}}\|\hat{y}^0 - \vec{y}\|_2 \leq O(1). \tag{8.155}$$

Now, suppose that (8.154) holds for some t. Then, because the errors are monotonically decreasing, (cf. (8.141) and (8.150)),

$$\frac{1}{\sqrt{n}}\|\hat{y}^{t+1} - \vec{y}\|_2 \leq \frac{1}{\sqrt{n}}\|\hat{y}^t - \vec{y}\|_2 \leq O(1). \tag{8.156}$$

Thus, (8.154) holds for all $t \in \mathbb{N}$.

Next, applying Lemma 8.28 with $\theta = \theta^t$ and our assumption that $\frac{\beta}{\sigma^2} \lesssim \frac{1}{n}$, we conclude that:

$$\frac{1}{\sqrt{n}}\|\Phi\theta^t - \hat{y}^t\|_2 \leq O(1) \tag{8.157}$$

Using this result and (8.154), we can show that $\frac{1}{\sqrt{n}}\|\Phi(\theta^t - \hat{\theta})\|_2$ is $O(1)$.

$$\begin{aligned}
\frac{1}{\sqrt{n}}\|\Phi(\theta^t - \hat{\theta})\|_2 &= \frac{1}{\sqrt{n}}\|\Phi\theta^t - \vec{y}\|_2 && (\vec{y} = \Phi\hat{\theta}) && (8.158)\\
&= \frac{1}{\sqrt{n}}\|\Phi\theta^t - \hat{y}^t + \hat{y}^t - \vec{y}\|_2 && && (8.159)\\
&\leq \frac{1}{\sqrt{n}}\|\Phi\theta^t - \hat{y}^t\|_2 + \frac{1}{\sqrt{n}}\|\hat{y}^t - \vec{y}\|_2 && \text{(triangle ineq.)} && (8.160)\\
&\leq O(1). && && (8.161)
\end{aligned}$$

Then, leveraging the definition of σ given in (8.117) and rearranging, we obtain (nearly) the desired result:

$$\|\theta^t - \hat{\theta}\|_2 \leq \frac{1}{\sigma}\|\Phi(\theta^t - \hat{\theta})\|_2 \leq O(\sqrt{n}/\sigma). \tag{8.162}$$

Recall that in Lemma 8.25, we proved that

$$\|\hat{\theta} - \theta^0\|_2 \leq O(\sqrt{n}/\sigma). \tag{8.163}$$

If $\beta/\sigma^2 \ll 1/n$, we conclude that

$$\begin{aligned}
\|\theta^t - \theta^0\|_2 &\leq \|\hat{\theta} - \theta^0\|_2 + \|\theta^t - \hat{\theta}\|_2 && \text{(triangle ineq.)} && (8.164)\\
&\leq O\left(\frac{\sqrt{n}}{\sigma}\right) \leq \frac{\sigma}{4\sqrt{n}\beta}. && && (8.165)
\end{aligned}$$

□

8.4.3 Limitations of NTK

The NTK approach has its limitations.

- Empirically, optimizing $g_\theta(x)$ as described in the theory does not work as well as state-of-the-art (or even standard) deep learning methods. For example, using the NTK approach (i.e., taking the Taylor expansion and optimizing $g_\theta(x)$) with a ResNet generally does not perform as well as ResNet with best-tuned hyperparameters.
- The NTK approach requires a specific initialization scheme and learning rate which may not coincide with what is commonly used in practice.

- The analysis above was for gradient descent, while stochastic gradient descent is used in practice, introducing noise in the procedure. This means that NTK with stochastic gradient descent requires a small learning rate to stay in the initialization neighborhood. Deviating from the requirements can lead to leaving the initialization neighborhood.

One possible explanation for the gap between theory and practice is because NTK effectively requires a fixed kernel, so there is no incentive to select the right features. Furthermore, the minimum ℓ_2-norm solution is typically dense. This is similar to the difference between sparse and dense combinations of features observed in the ℓ_1-SVM/two-layer network versus the standard kernel method SVM (or ℓ_2-SVM) analyzed previously.

To make these ideas more concrete, consider the following example [Wei et al., 2020].

Example 8.34. Let $x \in \mathbb{R}^d$ and $y \in \{-1, 1\}$. Assume that each component of x satisfies $x_i \in \{-1, 1\}$. Define the output $y = x_1 x_2$, that is, y is only a function of the first two components of x.

This output function can be described exactly by a neural network consisting of a sparse combination of the features (4 neurons to be exact):

$$\hat{y} = \frac{1}{2}\left[\phi_{\text{relu}}(x_1 + x_2) + \phi_{\text{relu}}(-x_1 - x_2) - \phi_{\text{relu}}(x_1 - x_2) - \phi_{\text{relu}}(x_2 - x_1)\right] \tag{8.166}$$

$$= \frac{1}{2}\left(|x_1 + x_2| - |x_1 - x_2|\right) \tag{8.167}$$

$$= x_1 x_2. \tag{8.168}$$

(8.167) follows from the fact that $\phi_{\text{relu}}(t) + \phi_{\text{relu}}(-t) = |t|$ for all t, while (8.168) follows from evaluating the 4 possible values of (x_1, x_2). Thus, we can solve this problem exactly with a very sparse combination of features.

However, if we were to use the NTK approach (kernel method), the network's output will always involve $\sigma(w^\top x)$ where w is random so it includes all components of x (i.e. a dense combination of features), and cannot isolate just the relevant features x_1 and x_2. This is illustrated in the following informal theorem:

Theorem 8.35. *The kernel method with NTK requires $n = \Omega(d^2)$ samples to learn Example 8.34 well. In contrast, the neural network regularized by $\sum_{j=1}^m |u_j| \|w_j\|_2$ only requires $n = O(d)$ samples.*