CS229M/STATS214: Machine Learning Theory

Lecturer: Tengyu Ma — Lecture # 13
Scribe: Rohith Kuditipudi and Kefan Dong — Mar 1st, 2021

13.1 Review and overview

In the last lecture, we finished our discussion of the neural tangent kernel (NTK). Recall that in the NTK regime, we choose our random initialization scheme in a way that guarantees the existence of a global minimum in a small neighborhood of the initialization, and then show that (stochastic) gradient descent (SGD) converges to this minimum.

As discussed previously, the NTK framework does not fully explain the empirical success of deep learning. In particular, the NTK solution typically performs substantially worse (in terms of generalization error) than standard deep learning algorithms and architectures. Indeed, one of the miracles of modern deep learning is the phenomenon of *algorithmic regularization* (also known as *implicit regularization* or *implicit bias*): although the loss landscape may contain infinitely many global minimizers, many of which do not generalize well, in practice our optimizer (e.g. SGD) tends to recover solutions with good generalization properties.

The focus of today's lecture will be to illustrate algorithmic regularization in simple settings. In particular, we first show that gradient descent (with the right initialization) identifies the minimum norm interpolating solution in overparametrized linear regression. Next, we show that for a certain non-convex reparametrization of the linear regression task where the data is generated from a sparse ground-truth model, gradient descent (again, suitably initialized) approximately recovers a sparse solution with good generalization.

13.2 Algorithmic regularization in overparametrized linear regression

We prove that gradient descent initialized at the origin converges to the minimum norm interpolating solution (assuming such a solution exists). Let $X := \left[x^{(1)}, ..., x^{(n)}\right]^T \in \mathbb{R}^{n \times d}$ denote our data matrix and $y := \left[y^{(1)}, ..., y^{(n)}\right]^T \in \mathbb{R}^n$ denote our label vector, where $n < d$. Assume X is full rank. Our goal is to find a weight vector β that minimizes our empirical loss function $\widehat{L}(\beta) := \frac{1}{2}||y - X\beta||_2^2$.

13.2.1 Analysis of algorithmic regularization

As we are in the overparametrized setting with $n < d$ and X full rank, there exist infinitely many global minimizers that interpolate the data and hence achieve zero loss. In fact, the following lemma shows that the set of global minimizers forms a subspace.

Lemma 13.1. *Let $X^\dagger$ denote the pseudoinverse[1] of X. Then β is a global minimizer if and only if $\beta = X^\dagger y + \zeta$ for some ζ such that $\zeta \perp x_1, ..., x_n$.*

[1] Since X is full rank, XX^T is invertible and so we have $X^\dagger = X^T(XX^T)^{-1}$. Note that $XX^\dagger X = X$.

Proof. For any $\beta \in \mathbb{R}^d$, we can decompose it as $\beta = X^\dagger + \zeta$ for some $\zeta \in \mathbb{R}^d$. Since

$$X\beta = X(X^\dagger y + \zeta) = y + X\zeta, \tag{13.1}$$

β is a global minimizer if and only if $X\zeta = 0$, which happens if and only if $\zeta \perp x_1, ..., x_n$.

□

From Lemma 13.1, we can derive an explicit formula for the minimum norm interpolant $\beta^\star := \arg\min_{\beta:\widehat{L}(\beta)=0} ||\beta||_2$.

Corollary 13.2. $\beta^\star = X^\dagger y$.

Proof. Take any β such that $\widehat{L}(\beta) = 0$, and write $\beta = X^\dagger y + \zeta$. Then from the definition of $X^\dagger$ and the fact that $X\zeta = 0$ (see the proof of Lemma 13.1), we have

$$\begin{aligned}
||\beta||_2^2 &= ||X^\dagger y||_2^2 + ||\zeta||_2^2 + 2\langle X^\dagger y, \zeta\rangle && (13.2)\\
&= ||X^\dagger y||_2^2 + ||\zeta||_2^2 + 2\langle X^T(XX^T)^{-1}y, \zeta\rangle && (13.3)\\
&= ||X^\dagger y||_2^2 + ||\zeta||_2^2 + 2\langle (XX^T)^{-1}y, X\zeta\rangle && (13.4)\\
&= ||X^\dagger y||_2^2 + ||\zeta||_2^2 \qquad (\because X\zeta = 0) && (13.5)\\
&\geq ||X^\dagger y||_2^2, && (13.6)
\end{aligned}$$

with equality if and only if $\zeta = 0$.

□

Now, suppose we learn β using gradient descent with initialization β^0, where at iteration t we set $\beta^t = \beta^{t-1} - \eta\nabla\widehat{L}(\beta^{t-1})$ for some learning rate η. Since $\widehat{L}(\beta)$ is convex, we know from standard results in convex optimization that gradient descent will converge to a global minimizer for a suitably chosen learning rate η (in particular, taking η to be sufficiently small). Assuming $\beta^0 = 0$, we will in fact recover the minimum norm interpolating solution.

Theorem 13.3. *Suppose gradient descent on $\widehat{L}(\beta)$ with initialization $\beta^0 = 0$ converges to a solution $\hat{\beta}$ such that $L(\hat{\beta}) = 0$. Then $\hat{\beta} = \beta^\star$.*

Proof. We first show via induction that $\beta^t \in \text{span}\left\{x^{(1)}, \ldots, x^{(n)}\right\}$ for all t. Note that $\beta^0 = 0 \in \text{span}\left\{x^{(1)}, \ldots, x^{(n)}\right\}$. Now suppose $\beta^{t-1} \in \text{span}\left\{x^{(1)}, \ldots, x^{(n)}\right\}$. Recall that $\beta^t = \beta^{t-1} - \eta\nabla\widehat{L}(\beta^{t-1})$. Since left-multiplying any vector by X^T amounts to taking a linear combination of the rows of X, it follows that $\eta\nabla\widehat{L}(\beta^{t-1}) = \eta X^T(X\beta^{t-1} - y) \in \text{span}\left\{x^{(1)}, \ldots, x^{(n)}\right\}$, and so $\beta^t = \beta^{t-1} - \eta\nabla\widehat{L}(\beta^{t-1}) \in \text{span}\left\{x^{(1)}, \ldots, x^{(n)}\right\}$. This proves the induction step.

Next, we show that $\hat{\beta} \in \text{span}\left\{x^{(1)}, \ldots, x^{(n)}\right\}$ and $\widehat{L}(\hat{\beta}) = 0$ implies $\hat{\beta} = \beta^\star$. By definition, $\hat{\beta} \in \text{span}\left\{x^{(1)}, \ldots, x^{(n)}\right\}$ implies $\hat{\beta} = X^T v$ for some $v \in \mathbb{R}^n$. Since $\widehat{L}(\hat{\beta}) = 0$, we have $0 = X\hat{\beta} - y = XX^T v - y$. This implies $v = (XX^T)^{-1}y$, and so $\hat{\beta} = X^T v = X^T(XX^T)^{-1}y = X^\dagger y = \hat{\beta}^\star$. □

13.3 Algorithmic regularization in non-linear models

We give an example of algorithmic regularization in a non-convex version of the overparametrized linear regression task considered in the previous section.

Take X and y as defined in Section 13.2. This time, our goal is to find a weight vector that minimizes our empirical loss function

$$\widehat{L}(\beta) := \frac{1}{4n}\sum_{i=1}^{n}\left(y^{(i)} - f_\beta(x^{(i)})\right)^2, \tag{13.7}$$

where $f_\beta(x) := \langle \beta \odot \beta, x\rangle$. (The operation $\odot$ denotes the Hadamard product: for $u, v \in \mathbb{R}^d$, $u \odot v \in \mathbb{R}^d$ is defined by $(u \odot v)_i := u_i v_i$ for $i = 1, \ldots, d$.)

We assume $x^{(1)}, ..., x^{(n)} \overset{\text{iid}}{\sim} \mathcal{N}(0, I_{d\times d})$ and $y^{(i)} = f_{\beta^\star}(x^{(i)})$, where the ground truth vector $\beta^\star$ is r-sparse (i.e. $\|\beta^\star\|_0 = r$). For simplicity, we assume $\beta^\star_i = \mathbf{1}\{i \in S\}$ for some $S \subset [d]$ such that $|S| = r$. We again analyze the overparametrized setting, where this time $n \ll d$ but also $n \geq \widetilde{\Omega}(r^2)$.

13.3.1 Main results of algorithmic regularization

Note that while f_β is still linear over x, our loss is no longer convex over β. (To see this, suppose $\beta \neq 0$ is a global minimizer. Then we have $\widehat{L}(0) > \widehat{L}(\beta) = \widehat{L}(-\beta)$.) Thus, the effect of algorithmic regularization induced by gradient descent will be much different from the overparametrized linear regression setting.

In the previous setting of linear regression, solutions with low ℓ_2 norm are desirable as they tend to generalize well. In the present setting, we know our ground-truth parameter $\beta^\star$ is sparse. Thus, we want to learn a sparse solution $\hat{\beta}$, avoiding non-sparse solutions that may not generalize well. One approach to finding sparse solutions, called *lasso regression*, is to minimize the ℓ_1-regularized proxy loss

$$\sum_{i=1}^{n}\left(\langle \theta, x^{(i)}\rangle - y^{(i)}\right)^2 + \lambda\|\theta\|_1 \tag{13.8}$$

with respect to θ, where $\theta = \beta \odot \beta$. However, it turns out that we can equivalently learn a sparse solution by running gradient descent from a suitable initialization on the original *unregularized* loss.

To be specific, let $\beta^0 = \alpha\mathbf{1} \in \mathbb{R}^d$ be the initialization where α is a small positive number. The update rule of gradient descent algorithm is given by $\beta^{t+1} = \beta^t - \eta\nabla\widehat{L}(\beta^t)$. The next theorem shows that when $n = \widetilde{\Omega}(r^2)$, gradient descent on $\widehat{L}(\beta)$ converges to $\beta^\star$.

Theorem 13.4. *Let c be a sufficiently large universal constant. Suppose $n \geq cr^2\log^2(d)$ and $\alpha \leq 1/d^c$, then when $\dfrac{\log(d/\alpha)}{\eta} \lesssim T \lesssim \dfrac{1}{\eta\sqrt{d\alpha}}$, we have*

$$\left\|\beta^T \odot \beta^T - \beta^\star \odot \beta^\star\right\|_2^2 \leq O\left(\alpha\sqrt{d}\right). \tag{13.9}$$

(Here, T indexes the gradient descent steps.)

We make several remarks about Theorem 13.4 before presenting the proof.

Remark 13.5. In this problem we don't use $\beta^0 = 0$ as the initialization point because $\beta = 0$ is a critical point, that is, $\nabla\widehat{L}(0) = 0$. Note that the lower bound on T depends logarithimically on $1/\alpha$, so we can take α to be a small inverse polynomial on d and the lower bound won't change much. Also, the upper bound depends polynomially on $1/\alpha$ (which is considered very big when c is sufficiently large), so we do not need to use early stopping in a serious way.

Remark 13.6. Theorem 13.4 is a simplified version of Theorem 1.1 in [LMZ18].

Remark 13.7. $\widehat{L}(\beta)$ has many global minima. To see this, observe that the number of parameters is d and the number of constraints to fit all the examples is $O(n)$ because there are only n examples. Recall that for overparameterized model we have $d \gg n$; consequently, there exists many global minima of $\widehat{L}(\beta)$.

Remark 13.8. $\beta^\star$ is the min-norm solution in this case. That is,

$$\beta^\star = \operatorname{argmin} \|\beta\|_2^2 \qquad \text{s.t. } \widehat{L}(\beta) = 0. \tag{13.10}$$

Informally, this is because we can view $\beta \odot \beta$ as a vector $\theta \in \mathbb{R}^d$, which leads to $\|\beta\|_2^2 = \|\theta\|_1$. Then in the θ space (and with a little abuse of notation), the optimization problem (13.10) becomes

$$\theta^\star = \operatorname{argmin} \|\theta\|_1 \qquad \text{s.t. } \widehat{L}(\theta) = 0, \tag{13.11}$$

which is a lasso regression, whose solution is sparse.

Remark 13.9. In this non-linear case and the linear case before, gradient descent with small initialization converges to minimum ℓ_2-norm solution. Similarly, gradient descent with any initialization converges to the solution nearest to the initialization (for example, in the NTK analysis). However, we do not have a general theorem for this phenomenon.

13.3.2 Analysis of algorithmic regularization

In this section we prepare the ground work for the proof of Theorem 13.4, which will be presented next lecture.

We start by showing several basic properties about $\widehat{L}(\beta)$. Note that for any fixed vector $v \in \mathbb{R}^d$ and $x \in \mathbb{R}^d$, when x is drawn from $\mathcal{N}(0, I)$, we have

$$\mathbb{E}\left[\langle x, v\rangle^2\right] = \mathbb{E}\left[v^\top x x^\top v\right] = v^\top \mathbb{E}\left[xx^\top\right] v = \|v\|_2^2. \tag{13.12}$$

It follows that

$$\begin{aligned} L(\beta) &= \frac{1}{4}\mathbb{E}_{x\sim\mathcal{N}(0,I)}\left[(y - \langle \beta \odot \beta, x\rangle^2\right] && (13.13)\\ &= \frac{1}{4}\mathbb{E}_{x\sim\mathcal{N}(0,I)}\left[\langle \beta^\star \odot \beta^\star - \beta \odot \beta, x\rangle^2\right] && \text{(by dfn of } y) \quad (13.14)\\ &= \frac{1}{4}\left\|\beta^\star \odot \beta^\star - \beta \odot \beta\right\|_2^2. && \text{(by (13.12))} \quad (13.15) \end{aligned}$$

Note that (13.15) is the metric that we use to characterize how close β is to the ground-truch parameter $\beta^\star$ (see (13.9)).

In the following lemma we show that $\widehat{L}(\beta) \approx L(\beta)$ by uniform convergence. Generally speaking, uniform convergence of the loss function for all β requires $n \geq \Omega(d)$ samples, so in our setting (where $n \ll d$) $\widehat{L}(\beta) \approx L(\beta)$ does not always hold. However, since we assume $\beta^\star$ is sparse, the analysis only requires uniform convergence for sparse vectors.

Lemma 13.10. *Assume $n \geq \widetilde{\Omega}(r^2)$. With high probability over the randomness in $x^{(1)}, \cdots, x^{(n)}$, $\forall v$ such that $\|v\|_0 \leq r$ we have*

$$(1-\delta)\|v\|_2^2 \leq \frac{1}{n}\sum_{i=1}^{n}\langle v, x^{(i)}\rangle^2 \leq (1+\delta)\|v\|_2^2. \tag{13.16}$$

Lemma 13.10 is a special case of Lemma 2.2 in [LMZ18] so the proof is omitted here. We say the set $\left\{x^{(1)}, \cdots, x^{(n)}\right\}$ (or $X = [x^{(1)}, \cdots, x^{(n)}]$) satisfies (r, δ)-*RIP condition* (*restricted isometric property*) if Lemma 13.10 holds.

By algebraic manipulation, (13.16) is equivalent to

$$(1-\delta)\|v\|_2^2 \le v^\top \left(\frac{1}{n}\sum_{i=1}^n x^{(i)}(x^{(i)})^\top\right) v \le (1+\delta)\|v\|_2^2. \tag{13.17}$$

In other words, from the point of view of a sparse vector v we have $\sum_{i=1}^n x^{(i)}(x^{(i)})^\top \approx I$. (Note however that $\sum_{i=1}^n x^{(i)}(x^{(i)})^\top$ is not close to $I_{d\times d}$ in other notions of closeness. For example, $\sum_{i=1}^n x^{(i)}(x^{(i)})^\top$ is not close to $I_{d\times d}$ in spectral norm. Another way to see this is that $\sum_{i=1}^n x^{(i)}(x^{(i)})^\top$ is a $d \times d$ matrix but only has rank $n \ll d$.)

As a result, with the RIP condition we have $\widehat{L}(\beta) \approx L(\beta)$ if β is sparse. With more tools we can also get $\nabla\widehat{L}(\beta) \approx \nabla L(\beta)$. Let us define the set $S_r = \{\beta : \|\beta\|_0 \le O(r)\}$, the set where we have uniform convergence of $\widehat{L}$ and $\nabla\widehat{L}$. (Note, on the other hand, that there exists a dense $\beta \notin S_r$ such that $\widehat{L}(\beta) = 0$ but $L(\beta) \gg 0$.)

The main intuition for proving Theorem 13.4 is to leverage the uniform convergence when β belongs to the set S_r. Note that the initialization β^0 is not exactly r-sparse, but taking α to be sufficiently small, β^0 is approximately 0-sparse. The proof is decomposed into the following steps:

1. Gradient descent on $L(\beta)$ converge to $\beta^\star$ without leaving S_r, and
2. Gradient descent on $\widehat{L}(\beta)$ is similar to gradient descent on $L(\beta)$ inside S_r.

Combining the two steps we can show that gradient descent on $\widehat{L}(\beta)$ does not leave S_r and converges to $\beta^\star$.

As a warm up, in the following we prove the following theorem for gradient descent on $L(\beta)$.

Theorem 13.11. *For sufficiently small η, gradient descent on $L(\beta)$ converges to $\beta^\star$ in $\Theta\left(\frac{\log(1/(\epsilon\alpha))}{\eta}\right)$ iteration with ϵ-error in ℓ_2-distance.*

Proof (to be completed next lecture). Since

$$\nabla L(\beta) = (\beta \odot \beta - \beta^\star \odot \beta^\star) \odot \beta, \tag{13.18}$$

the gradient descent step is

$$\beta^{t+1} = \beta^t - \eta(\beta^t \odot \beta^t - \beta^\star \odot \beta^\star) \odot \beta^t. \tag{13.19}$$

Recall that $\beta^\star = \mathbf{1}\{i \in S\}$ and $\beta^0 = \alpha\mathbf{1}$, and the update rule above decouples across the coordinates of β^t. Thus, we only need to show that $|\beta_i^\star - \beta^t| \le \epsilon$ for the number of iterations stated in the Theorem. In this lecture we only consider the case where the coordinate $i \in S$. (We will deal with the $i \notin S$ case next lecture.) Assuming that $i \in S$, the update rule for coordinate i is

$$\beta_i^{t+1} = \beta_i^t - \eta(\beta_i^t \cdot \beta_i^t - 1 \cdot 1) \cdot \beta_i^t \tag{13.20}$$

$$= \beta_i^t - \eta\left[\left(\beta_i^t\right)^2 - 1\right]\beta_i^t. \tag{13.21}$$

Consider the following two cases:

- If $\beta_i^t \le 1/2$, we have

$$\beta_i^{t+1} = \beta_i^t \left[1 + \eta\left(1 - \left(\beta_i^t\right)^2\right)\right] \tag{13.22}$$

$$\ge \beta_i^t \left(1 + \frac{3}{4}\eta\right). \tag{13.23}$$

  Consequently, β_i^{t+1} grow exponentially, and it takes $\Theta\left(\frac{\log(1/\alpha)}{\eta}\right)$ iterations for β_i^t to grow from α to at least $1/2$.[2] This will bring us into the second case.

- if $\beta_i^t \ge 1/2$, we have

$$1 - \beta_i^{t+1} = 1 - \beta_i^t + \eta\left[\left(\beta_i^t\right)^2 - 1\right]\beta_i^t \tag{13.24}$$

$$= 1 - \beta_i^t - \eta\left(1 - \beta_i^t\right)\left(1 + \beta_i^t\right)\beta_i^t \tag{13.25}$$

$$\le 1 - \beta_i^t - \eta\left(1 - \beta_i^t\right)\beta_i^t \qquad (\because 1 + \beta_i^t \ge 1) \tag{13.26}$$

$$= \left(1 - \beta_i^t\right)\left(1 - \eta\beta_i^t\right) \tag{13.27}$$

$$\le \left(1 - \beta_i^t\right)(1 - \eta/2). \qquad (\because \beta_i^t \ge 1/2) \tag{13.28}$$

  Therefore it takes $\Theta\left(\frac{\log(1/\epsilon)}{\eta}\right)$ iterations to achieve $1 - \beta_i^t \le \epsilon$.

□

[2] This is because $(1+\eta)^{1/\eta} \approx e$, so $(1+\eta)^{c/\eta} \approx e^c$.

Bibliography

[LMZ18] Yuanzhi Li, Tengyu Ma, and Hongyang Zhang, *Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations*, Conference On Learning Theory, PMLR, 2018, pp. 2–47.