Generalization for Deep Learning

Philipp Grohs

universität wien

Faculty of Mathematics

Bedlewo, Nov 2019

Short Reading List

1. Mikhail Belkin, Daniel Hsu, Siyuan Ma, Soumik Mandal: Reconciling modern machine learning practice and the bias-variance trade-off; arXiv:1812.11118
2. Noah Golowich, Alexander Rakhlin, Ohad Shamir: Size-Independent Sample Complexity of Neural Networks; arXiv:1712.06541
3. Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, Oriol Vinyals: Understanding deep learning requires rethinking generalization; arXiv:1611.03530

Syllabus

1. Rademacher Complexity
2. Rademacher Complexity and Deep Learning?
3. Linear Regression Revisited (Jupyter and Blackboard)

Rademacher Complexity

Motivation

Setting: learning problem with loss function $l(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$, data distribution (X, Y) and, given training data $z = (x_i, y_i)_{i=1}^m$ i.i.d. according to $\mathbb{P}_{(X,Y)}$ and hypothesis class $\mathcal{H}$, solve the empirical risk minimization (ERM) problem

$$\hat{h}_{\mathcal{H},z} \in \operatorname*{argmin}_{h \in \mathcal{H}} \mathcal{E}_z(h),$$

where

$$\mathcal{E}_z(h) := \hat{\mathbb{E}}_z(l(h(X), Y)) := \frac{1}{m} \sum_{i=1}^m l(h(x_i), y_i).$$

Motivation

Setting: learning problem with loss function $l(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$, data distribution (X, Y) and, given training data $z = (x_i, y_i)_{i=1}^m$ i.i.d. according to $\mathbb{P}_{(X,Y)}$ and hypothesis class $\mathcal{H}$, solve the empirical risk minimization (ERM) problem

$$\hat{h}_{\mathcal{H},z} \in \operatorname*{argmin}_{h \in \mathcal{H}} \mathcal{E}_z(h),$$

where

$$\mathcal{E}_z(h) := \hat{\mathbb{E}}_z(l(h(X), Y)) := \frac{1}{m} \sum_{i=1}^m l(h(x_i), y_i).$$

Results: $m \gtrsim \frac{\ln(\mathcal{N}(\mathcal{H}, c\epsilon))}{\epsilon^2}$ samples suffice for generalization error $\leq \epsilon$ with high probability.

Motivation

Setting: learning problem with loss function $l(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$, data distribution (X, Y) and, given training data $z = (x_i, y_i)_{i=1}^m$ i.i.d. according to $\mathbb{P}_{(X,Y)}$ and hypothesis class $\mathcal{H}$, solve the empirical risk minimization (ERM) problem

$$\hat{h}_{\mathcal{H},z} \in \operatorname*{argmin}_{h \in \mathcal{H}} \mathcal{E}_z(h),$$

where

$$\mathcal{E}_z(h) := \hat{\mathbb{E}}_z(l(h(X), Y)) := \frac{1}{m} \sum_{i=1}^m l(h(x_i), y_i).$$

Results: $m \gtrsim \frac{\ln(\mathcal{N}(\mathcal{H}, c\epsilon))}{\epsilon^2}$ samples suffice for generalization error $\leq \epsilon$ with high probability.

Problem: The complexity measure $\ln(\mathcal{N}(\mathcal{H}, c\epsilon))$ is independent of the data distribution and independent of the sample, and hence very pessimistic.

From Cross-Validation to Rademacher Complexity

Recall that generalization error requires uniform bounds

$$\sup_{h\in\mathcal{H}} \left| \mathbb{E}[l(h(X),Y)] - \hat{\mathbb{E}}_z[l(h(X),Y)] \right|.$$

From Cross-Validation to Rademacher Complexity

Recall that generalization error requires uniform bounds

$$\sup_{h \in \mathcal{H}} \left| \mathbb{E}[l(h(X), Y)] - \hat{\mathbb{E}}_z[l(h(X), Y)] \right| .$$

In cross-validation we split the sample set z into two (independent) subsets z', z'' (of the same size) and estimate the generalization error as

$$\hat{\mathbb{E}}_{z'}[l(h(X), Y)] - \hat{\mathbb{E}}_{z''}[l(h(X), Y)]$$

From Cross-Validation to Rademacher Complexity

Recall that generalization error requires uniform bounds

$$\sup_{h \in \mathcal{H}} \left| \mathbb{E}[l(h(X), Y)] - \hat{\mathbb{E}}_z[l(h(X), Y)] \right| .$$

In cross-validation we split the sample set z into two (independent) subsets z', z'' (of the same size) and estimate the generalization error as

$$\hat{\mathbb{E}}_{z'}[l(h(X), Y)] - \hat{\mathbb{E}}_{z''}[l(h(X), Y)]$$

$$= \frac{2}{m} \left(\sum_{(x,y) \in z'} l(h(x), y) - \sum_{(x,y) \in z''} l(h(x), y) \right)$$

From Cross-Validation to Rademacher Complexity

Recall that generalization error requires uniform bounds

$$\sup_{h\in\mathcal{H}}\left|\mathbb{E}[l(h(X),Y)]-\hat{\mathbb{E}}_z[l(h(X),Y)]\right|.$$

In cross-validation we split the sample set z into two (independent) subsets z', z'' (of the same size) and estimate the generalization error as

$$\hat{\mathbb{E}}_{z'}[l(h(X),Y)]-\hat{\mathbb{E}}_{z''}[l(h(X),Y)]$$

$$=\frac{2}{m}\left(\sum_{(x,y)\in z'} l(h(x),y)-\sum_{(x,y)\in z''} l(h(x),y)\right)$$

$$=\frac{2}{m}\sum_{i=1}^{m}\sigma_i l(h(x_i),y_i),$$

where

$$\sigma_i=\begin{cases}1 & (x_i,y_i)\in z'\\ -1 & (x_i,y_i)\in z''\end{cases}$$

Hinge Loss

Suppose that $\mathcal{Y} = \{-1, 1\}$ and $l(h(x), y) = \frac{1-h(x)y}{2}$ (hinge loss).

Hinge Loss

Suppose that $\mathcal{Y} = \{-1, 1\}$ and $l(h(x), y) = \frac{1-h(x)y}{2}$ (hinge loss). Then

$$\frac{2}{m}\sum_{i=1}^{m}\sigma_i l(h(x_i), y_i) = \frac{1}{m}\sum_{i=1}^{m}\sigma_i(1 - y_i h(x_i)).$$

Hinge Loss

Suppose that $\mathcal{Y} = \{-1, 1\}$ and $l(h(x), y) = \frac{1-h(x)y}{2}$ (hinge loss). Then

$$\frac{2}{m}\sum_{i=1}^{m} \sigma_i l(h(x_i), y_i) = \frac{1}{m}\sum_{i=1}^{m} \sigma_i (1 - y_i h(x_i)).$$

If σ_i has random signs then also $-\sigma_i y_i$ has random signs.

Hinge Loss

Suppose that $\mathcal{Y} = \{-1, 1\}$ and $l(h(x), y) = \frac{1-h(x)y}{2}$ (hinge loss). Then

$$\frac{2}{m}\sum_{i=1}^{m}\sigma_i l(h(x_i), y_i) = \frac{1}{m}\sum_{i=1}^{m}\sigma_i(1 - y_i h(x_i)).$$

If σ_i has random signs then also $-\sigma_i y_i$ has random signs. Hence,

$$\mathbb{E}_\sigma\left[\sup_{h\in\mathcal{H}}\left|\frac{1}{m}\sum_{i=1}^{m}\sigma_i(1 - y_i h(x_i))\right|\right] = \mathcal{O}\left(\frac{1}{\sqrt{n}}\right) + \mathbb{E}_\sigma\left[\sup_{h\in\mathcal{H}}\left|\frac{1}{m}\sum_{i=1}^{m}\sigma_i h(x_i)\right|\right],$$

where $\sigma_i \in \{-1, 1\}$ uniformly at random.

Hinge Loss

Suppose that $\mathcal{Y} = \{-1, 1\}$ and $l(h(x), y) = \frac{1-h(x)y}{2}$ (hinge loss). Then

$$\frac{2}{m}\sum_{i=1}^{m}\sigma_i l(h(x_i), y_i) = \frac{1}{m}\sum_{i=1}^{m}\sigma_i(1 - y_i h(x_i)).$$

If σ_i has random signs then also $-\sigma_i y_i$ has random signs. Hence,

$$\mathbb{E}_\sigma\left[\sup_{h\in\mathcal{H}}\left|\frac{1}{m}\sum_{i=1}^{m}\sigma_i(1 - y_i h(x_i))\right|\right] = \mathcal{O}\left(\frac{1}{\sqrt{n}}\right) + \mathbb{E}_\sigma\left[\sup_{h\in\mathcal{H}}\left|\frac{1}{m}\sum_{i=1}^{m}\sigma_i h(x_i)\right|\right],$$

where $\sigma_i \in \{-1, 1\}$ uniformly at random.
In a sense, the quantity $\mathbb{E}_\sigma\left[\sup_{h\in\mathcal{H}}\left|\frac{1}{m}\sum_{i=1}^{m}\sigma_i h(x_i)\right|\right]$ measures how well the hypothesis class is able to fit random labels to the samples.

Rademacher Complexity

Definition

Rademacher complexity of function class $\mathcal{F}$ with respect to the sample $z = (x_i)_{i=1}^m$ is defined as

$$\mathcal{R}_z(\mathcal{F}) := \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \left| \frac{1}{m} \sum_{i=1}^m \sigma_i f(x_i) \right| \right],$$

where the expectation is taken over all Rademacher r.v.'s $\sigma = (\sigma_1, \ldots, \sigma_m) \in \{-1, 1\}^m$ with signs chosen uniformly at random. We also let

$$\mathcal{R}_m(\mathcal{F}) := \mathbb{E}_z \mathcal{R}_z(\mathcal{F}).$$

Rademacher Complexity

Definition

Rademacher complexity of function class $\mathcal{F}$ with respect to the sample $z = (x_i)_{i=1}^m$ is defined as

$$\mathcal{R}_z(\mathcal{F}) := \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \left| \frac{1}{m} \sum_{i=1}^m \sigma_i f(x_i) \right| \right],$$

where the expectation is taken over all Rademacher r.v.'s $\sigma = (\sigma_1, \ldots, \sigma_m) \in \{-1, 1\}^m$ with signs chosen uniformly at random. We also let

$$\mathcal{R}_m(\mathcal{F}) := \mathbb{E}_z \mathcal{R}_z(\mathcal{F}).$$

In the statistical learning setting we would take $\mathcal{F} = l \circ \mathcal{H}$.

Rademacher Complexity

Definition

Rademacher complexity of function class $\mathcal{F}$ with respect to the sample $z = (x_i)_{i=1}^m$ is defined as

$$\mathcal{R}_z(\mathcal{F}) := \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \left| \frac{1}{m} \sum_{i=1}^m \sigma_i f(x_i) \right| \right],$$

where the expectation is taken over all Rademacher r.v.'s $\sigma = (\sigma_1, \ldots, \sigma_m) \in \{-1, 1\}^m$ with signs chosen uniformly at random. We also let

$$\mathcal{R}_m(\mathcal{F}) := \mathbb{E}_z \mathcal{R}_z(\mathcal{F}).$$

In the statistical learning setting we would take $\mathcal{F} = l \circ \mathcal{H}$.
If l is 1-Lipschitz in x and $|l(x, y)| \leq 1$ it holds that $\mathcal{R}_z(l \circ \mathcal{F}) \leq \mathcal{R}_z(\mathcal{F}) + \mathcal{O}(1/\sqrt{m})$.

Generalization Bounds

Theorem

Suppose that $\mathcal{F} \subset \{f : \mathcal{X} \to [0,1]\}$ is a function class and X a r.v. on $\mathcal{X}$ and let $z = (x_1, \ldots, x_m)$ be i.i.d. drawn according to the law of X. Then with probability $\geq 1 - \delta$ for all $f \in \mathcal{F}$ it holds that

$$\sup_{f \in \mathcal{F}} \left| \mathbb{E}[f] - \hat{\mathbb{E}}_z[f] \right| \leq 2\mathcal{R}_m(\mathcal{F}) + \mathcal{O}\left(\sqrt{\frac{\ln(1/\delta)}{m}} \right).$$

Generalization Bounds

Theorem

Suppose that $\mathcal{F} \subset \{f : \mathcal{X} \to [0,1]\}$ is a function class and X a r.v. on $\mathcal{X}$ and let $z = (x_1, \ldots, x_m)$ be i.i.d. drawn according to the law of X. Then with probability $\geq 1 - \delta$ for all $f \in \mathcal{F}$ it holds that

$$\sup_{f \in \mathcal{F}} \left| \mathbb{E}[f] - \hat{\mathbb{E}}_z[f] \right| \leq 2\mathcal{R}_m(\mathcal{F}) + \mathcal{O}\left(\sqrt{\frac{\ln(1/\delta)}{m}} \right).$$

Same inequality holds true with $\mathcal{R}_m(\mathcal{F})$ replaced by $\mathcal{R}_z(\mathcal{F})$.

Composition Lemma

Lemma

Suppose that g is 1-Lipschitz. Then

$$\mathcal{R}_z(g \circ \mathcal{F}) \leq \mathcal{R}_z(\mathcal{F}).$$

Linear Classifiers

$$\mathcal{H} = \{\{y : \|y\|_2 \leq a\} \ni x \mapsto v^T x : \|v\|_2 \leq B\}.$$

Linear Classifiers

$$\mathcal{H} = \{\{y : \|y\|_2 \leq a\} \ni x \mapsto v^T x : \|v\|_2 \leq B\}.$$

$$\mathcal{R}_z(\mathcal{H}) = \mathbb{E}_\sigma \left[\sup_{\|v\|_2 \leq B} \left| \frac{1}{m} v^T \sum_{i=1}^{m} \sigma_i x_i \right| \right]$$

Linear Classifiers

$$\mathcal{H} = \{\{y : \|y\|_2 \leq a\} \ni x \mapsto v^T x : \|v\|_2 \leq B\}.$$

$$\begin{aligned} \mathcal{R}_z(\mathcal{H}) &= \mathbb{E}_\sigma \left[\sup_{\|v\|_2 \leq B} \left| \frac{1}{m} v^T \sum_{i=1}^m \sigma_i x_i \right| \right] \\ &= \frac{B}{m} \mathbb{E}_\sigma \left[\left\| \sum_{i=1}^m \sigma_i x_i \right\|_2 \right] \end{aligned}$$

Linear Classifiers

$$\mathcal{H} = \{\{y : \|y\|_2 \leq a\} \ni x \mapsto v^T x : \|v\|_2 \leq B\}.$$

$$\begin{aligned}
\mathcal{R}_z(\mathcal{H}) &= \mathbb{E}_\sigma \left[\sup_{\|v\|_2 \leq B} \left| \frac{1}{m} v^T \sum_{i=1}^{m} \sigma_i x_i \right| \right] \\
&= \frac{B}{m} \mathbb{E}_\sigma \left[\left\| \sum_{i=1}^{m} \sigma_i x_i \right\|_2 \right] \\
&\leq \frac{B}{m} \sqrt{\mathbb{E}_\sigma \left[\sum_{i=1}^{m} \|x_i\|_2^2 \right]}
\end{aligned}$$

Linear Classifiers

$$\mathcal{H} = \{\{y : \|y\|_2 \leq a\} \ni x \mapsto v^T x : \|v\|_2 \leq B\}.$$

$$\begin{aligned}
\mathcal{R}_z(\mathcal{H}) &= \mathbb{E}_\sigma \left[\sup_{\|v\|_2 \leq B} \left| \frac{1}{m} v^T \sum_{i=1}^m \sigma_i x_i \right| \right] \\
&= \frac{B}{m} \mathbb{E}_\sigma \left[\left\| \sum_{i=1}^m \sigma_i x_i \right\|_2 \right] \\
&\leq \frac{B}{m} \sqrt{\mathbb{E}_\sigma \left[\sum_{i=1}^m \|x_i\|_2^2 \right]} \\
&\leq \frac{Ba}{\sqrt{m}}
\end{aligned}$$

ReLU Networks with bounded Weights

$\mathcal{H}_{L,B1,\ldots,B_L} = \{\{y : \|y\|_2 \leq 1\} \ni x \mapsto W_L g(W_{L-1} \ldots g(W_1 x) \ldots) :$
$\|W_i\|_F \leq B_i,\ i = 1, \ldots, L\}$ and $g(t) = \max\{t, 0\}$.

ReLU Networks with bounded Weights

$\mathcal{H}_{L,B1,\ldots,B_L} = \{\{y : \|y\|_2 \leq 1\} \ni x \mapsto W_L g(W_{L-1} \ldots g(W_1 x) \ldots) :$
$\|W_i\|_F \leq B_i,\ i = 1, \ldots, L\}$ and $g(t) = \max\{t, 0\}$.

Theorem [Golowich-Rakhlin-Shamir (2019)]

$$\mathcal{R}_z\left(\mathcal{H}_{L,B}\right) \leq \frac{B(\sqrt{2\log(2)d} + 1)\prod_{i=1}^{L} B_i}{\sqrt{m}}$$

ReLU Networks with bounded Weights

$\mathcal{H}_{L,B1,\ldots,B_L} = \{\{y : \|y\|_2 \leq 1\} \ni x \mapsto W_L g(W_{L-1} \ldots g(W_1 x) \ldots) :$
$\|W_i\|_F \leq B_i,\ i = 1, \ldots, L\}$ and $g(t) = \max\{t, 0\}$.

Theorem [Golowich-Rakhlin-Shamir (2019)]

$$\mathcal{R}_z\left(\mathcal{H}_{L,B}\right) \leq \frac{B(\sqrt{2\log(2)d} + 1)\prod_{i=1}^{L} B_i}{\sqrt{m}}$$

Proof uses homogeneity of the ReLU.

Rademacher Complexity and Deep Learning?

Understanding Deep Learning Requires Rethinking Generalization

Table 1: The training and test accuracy (in percentage) of various models on the CIFAR10 dataset. Performance with and without data augmentation and weight decay are compared. The results of fitting random labels are also included.

model	# params	random crop	weight decay	train accuracy	test accuracy
Inception	1,649,402	yes	yes	100.0	89.05
		yes	no	100.0	89.31
		no	yes	100.0	86.03
		no	no	100.0	85.75
(fitting random labels)		no	no	100.0	9.78
Inception w/o BatchNorm	1,649,402	no	yes	100.0	83.00
		no	no	100.0	82.00
(fitting random labels)		no	no	100.0	10.12
Alexnet	1,387,786	yes	yes	99.90	81.22
		yes	no	99.82	79.66
		no	yes	100.0	77.36
		no	no	100.0	76.07
(fitting random labels)		no	no	99.82	9.86
MLP 3x512	1,735,178	no	yes	100.0	53.35
		no	no	100.0	52.39
(fitting random labels)		no	no	100.0	10.48
MLP 1x512	1,209,866	no	yes	99.80	50.39
		no	no	100.0	50.51
(fitting random labels)		no	no	99.34	10.61

[Zhang etal] State of the art networks can fit random labels.

Understanding Deep Learning Requires Rethinking Generalization

Table 1: The training and test accuracy (in percentage) of various models on the CIFAR10 dataset. Performance with and without data augmentation and weight decay are compared. The results of fitting random labels are also included.

model	# params	random crop	weight decay	train accuracy	test accuracy
Inception	1,649,402	yes	yes	100.0	89.05
		yes	no	100.0	89.31
		no	yes	100.0	86.03
		no	no	100.0	85.75
(fitting random labels)		no	no	100.0	9.78
Inception w/o BatchNorm	1,649,402	no	yes	100.0	83.00
		no	no	100.0	82.00
(fitting random labels)		no	no	100.0	10.12
Alexnet	1,387,786	yes	yes	99.90	81.22
		yes	no	99.82	79.66
		no	yes	100.0	77.36
		no	no	100.0	76.07
(fitting random labels)		no	no	99.82	9.86
MLP 3x512	1,735,178	no	yes	100.0	53.35
		no	no	100.0	52.39
(fitting random labels)		no	no	100.0	10.48
MLP 1x512	1,209,866	no	yes	99.80	50.39
		no	no	100.0	50.51
(fitting random labels)		no	no	99.34	10.61

[Zhang etal] State of the art networks can fit random labels. $\rightsquigarrow$ Rademacher complexity ~ 1.

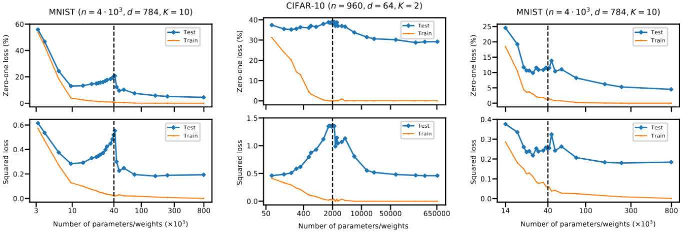
Understanding Deep Learning Requires Rethinking Generalization

Table 1: The training and test accuracy (in percentage) of various models on the CIFAR10 dataset. Performance with and without data augmentation and weight decay are compared. The results of fitting random labels are also included.

model	# params	random crop	weight decay	train accuracy	test accuracy
Inception	1,649,402	yes	yes	100.0	89.05
		yes	no	100.0	89.31
		no	yes	100.0	86.03
		no	no	100.0	85.75
(fitting random labels)		no	no	100.0	9.78
Inception w/o BatchNorm	1,649,402	no	yes	100.0	83.00
		no	no	100.0	82.00
(fitting random labels)		no	no	100.0	10.12
Alexnet	1,387,786	yes	yes	99.90	81.22
		yes	no	99.82	79.66
		no	yes	100.0	77.36
		no	no	100.0	76.07
(fitting random labels)		no	no	99.82	9.86
MLP 3x512	1,735,178	no	yes	100.0	53.35
		no	no	100.0	52.39
(fitting random labels)		no	no	100.0	10.48
MLP 1x512	1,209,866	no	yes	99.80	50.39
		no	no	100.0	50.51
(fitting random labels)		no	no	99.34	10.61

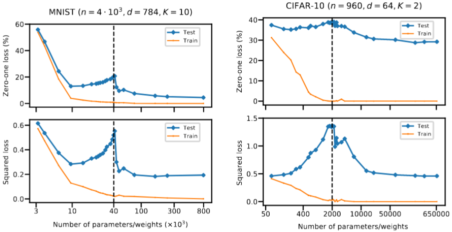
[Zhang etal] State of the art networks can fit random labels. $\rightsquigarrow$ Rademacher complexity ~ 1. $\rightsquigarrow$ classical theory does not explain generalization!

Double Descent Curve



(a) (b) (c)

Double Descent Curve



(a) (b) (c)

[Belkin etal (2019)] Sometimes a "double descent curve" is observed