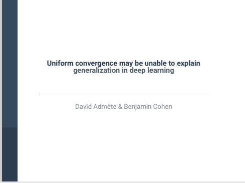
Uniform convergence may be unable to explain generalization in deep learning

David Admète & Benjamin Cohen

2022-03-23

Uniform convergence may be unable to explain **generalization in deep learning**



Uniform convergence may be unable to explain
generalization in deep learning

David Admète & Benjamin Cohen

Hello and thank you for attending this presentation. We are presenting the article "Uniform convergence may be unable to explain generalization in deep neural networks", from Nagarajan and Kolter, accepted in the NeurIPS 2019. We will first present the article in general and then give 5 minutes at the end of the talk to discuss the exercise we have created.

First of all, let us make a short introduction before announcing the plan. As we are going to go through a lot of presentations, and as the course is already called "Generalisation properties of algorithms in Machine Learning", I propose not to go back to the basics of ML, and to start directly from the last sessions, on deep learning.

Generalization properties in deep neural networks

Generalization of overparametrized networks without regularization ?

- Implicit Bias of GD
- Noise
- Initialization...

Find an upper bound for :

$$\left|\mathscr{L}_{\mathscr{D}}(h) - \hat{\mathscr{L}}_S(h)\right|$$

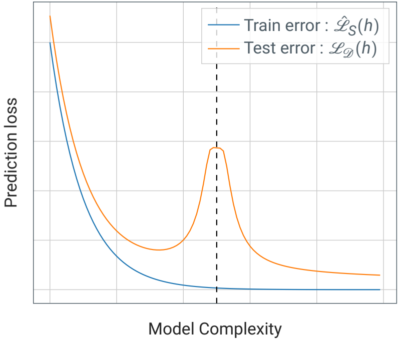
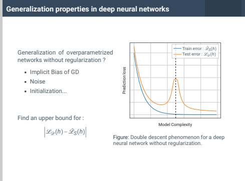


Figure: Double descent phenomenon for a deep neural network without regularization.



Generalization properties in deep neural networks

Generalization of overparametrized networks without regularization ?

- Implicit Bias of GD
- Noise
- Initialization...

Find an upper bound for :

$$\left|\mathscr{L}_{\mathscr{D}}(h) - \hat{\mathscr{L}}_S(h)\right|$$

Train error : $\hat{\mathscr{L}}_S(h)$
Test error : $\mathscr{L}_{\mathscr{D}}(h)$
Prediction loss
Model Complexity

Figure: Double descent phenomenon for a deep neural network without regularization.

As we have seen in the last slides, deep neural networks can have an unexpected behaviour when we increase the complexity of the models. Initially, when we increase the number of parameters, the train loss and test loss decrease, until the overfit, and the test loss goes up. However, even without explicit regularisation, the test loss can go down again, proof of a good generalisation properties of the network. We have discussed several possibilities of implicit regularisation, but the explain of the generalization properties are of course still not completely understood.

Here, we want to quantify these generalisation properties, so we want to find an upper bound of the difference between the training loss (which is empirical) and the test loss (which is an expected value). This error depends on many variables: the hypothesis h, the number of samples, the dataset S... There are therefore several ways to deal with it.

From generalization to uniform convergence

Generalization error :

$$\mathscr{L}_{\mathscr{D}}(h_S) - \hat{\mathscr{L}}_S(h_S) \underset{S}{\overset{1-\delta}{\leqslant}} \epsilon_{\text{gen}}(m, \delta).$$

Uniform convergence bound :

$$\sup_h \left| \mathscr{L}_{\mathscr{D}}(h) - \hat{\mathscr{L}}_S(h) \right| \underset{S}{\overset{1-\delta}{\leqslant}} \epsilon_{\text{unif}}(m, \delta).$$

What is a good u-c bound ?

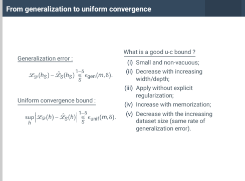
(i) Small and non-vacuous;

(ii) Decrease with increasing width/depth;

(iii) Apply without explicit regularization;

(iv) Increase with memorization;

(v) Decrease with the increasing dataset size (same rate of generalization error).



From generalization to uniform convergence

Generalization error :

$$\mathscr{L}_{\mathscr{D}}(h_S) - \hat{\mathscr{L}}_S(h_S) \underset{S}{\overset{1-\delta}{\leqslant}} \epsilon_{\text{gen}}(m,\delta).$$

Uniform convergence bound :

$$\sup_h \left| \mathscr{L}_{\mathscr{D}}(h) - \hat{\mathscr{L}}_S(h) \right| \underset{S}{\overset{1-\delta}{\leqslant}} \epsilon_{\text{unif}}(m,\delta).$$

What is a good u-c bound ?

(i) Small and non-vacuous;
(ii) Decrease with increasing width/depth;
(iii) Apply without explicit regularization;
(iv) Increase with memorization;
(v) Decrease with the increasing dataset size (same rate of generalization error).

For the moment, we will focus on two tools in particular: the generalization error and the uniform convergence bound. Without going into the details of the formula for the moment, we can immediately notice that the uniform convergence is a stronger notion, since it implies the supremum of the loss difference on all hypotheses, whereas the generalization error is an evaluation in a particular classifier. We can make an analogy with the study of sequences and series of functions : we use uniform convergence to find properties. It's good to know that the notion of uniform convergence for classifiers already existed when the notion of learnability arrived (quite old so). But what do we expect from a such bound ?

Basically that it behaves as the generalisation error. So we would like it to be non vacuous (meaning lesser than 1), decreasing with the number of parameters of the network. We want bounds concerning the network directly trained by SGD, without any explicit regularization. Increasing when we randomly flip the labels, and most of all, which behaves as the generalization error when the dataset size increase. This last point is of paramount importance for the understanding of the article. In summary it is quite logical, we just want it to behave like the generalization error. But, Remember the point 5. Now that we have an idea of what we expect from the uniform convergence, we can move on the position of the article.

Table of contents

1. **Introducing the main bounds**
   - Notations & Definitions
   - Generalization error
   - Uniform convergence bound
2. **About weights and uniform convergence bounds**
   - A ReLU network
   - Pruning the hypothesis class
   - Bounds growing with m
3. **Three examples of overparametrized models**
4. **Conclusion & References**
5. **About the exercise**

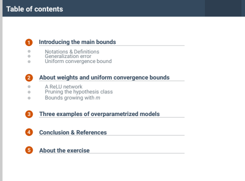


Table of contents

1. Introducing the main bounds
   - Notations & Definitions
   - Generalization error
   - Uniform convergence bound
2. About weights and uniform convergence bounds
   - A ReLU network
   - Pruning the hypothesis class
   - Bounds growing with *m*
3. Three examples of overparametrized models
4. Conclusion & References
5. About the exercise

After having reserved a part to the presentation of the various concepts, we will speak about the contributions of the article. It discuss many hypothesis, so it is not easy to identify a simple structure. But, two major parts can be distinguished.

First, the paper focuses on the last point of what is expected from a uniform convergence bound: that its dependence on the dataset size reflects the one of the generalization error. In particular, using the example of a deep relu network, it highlights the need to define a tighter notion of uniform convergence. And it shows that even these new bounds increase with the training set size, contrary to the generalization error.

Then, three examples of setups are studied. We will spend little time on this during the presentation because one of the setups, the linear classifier, is discussed in detail in our exercise.

1. **Introducing the main bounds**
   - Notations & Definitions
   - Generalization error
   - Uniform convergence bound
2. About weights and uniform convergence bounds
3. Three examples of overparametrized models
4. Conclusion & References
5. About the exercise

Notations

Main notations

$(X,y) \sim \mathscr{D}$	sample
$y \in \{-1,+1\}$	
m	dataset size
$S = \left\{(X^{(i)}, y^{(i)})\right\}$	dataset
$h \in \mathscr{H}$	hypothesis
h_S	hyp. trained on S

Expected loss :

$$\mathscr{L}_{\mathscr{D}}(h) := \mathbb{E}_{(x,y)\sim\mathscr{D}}[\mathscr{L}(h(x),y)]$$

Empirical loss :

$$\hat{\mathscr{L}}_S(h) := \frac{1}{m}\sum_{i=1}^{m}\mathscr{L}(h(x^{(i)}),y^{(i)})$$

The γ-loss $\mathscr{L} : \mathbb{R} \times \{-1,1\} \to [0,1]$:

$$\mathscr{L}(y',y) = \begin{cases} 1 & \text{if } yy' \leqslant 0 \\ 1 - \dfrac{yy'}{\gamma} & \text{if } yy' \leqslant (0,\gamma) \\ 0 & \text{if } yy' \geqslant 0. \end{cases}$$

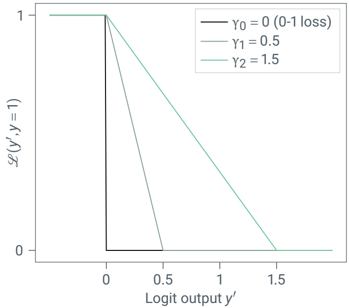
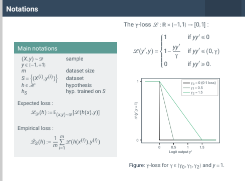


Figure: γ-loss for $\gamma \in \{\gamma_0, \gamma_1, \gamma_2\}$ and $y = 1$.



Notations

Main notations

$(X,y) \sim \mathscr{D}$	sample
$y \in \{-1,+1\}$	
m	dataset size
$S = \left\{(X^{(i)},y^{(i)})\right\}$	dataset
$h \in \mathscr{H}$	hypothesis
h_S	hyp. trained on S

Expected loss :

$$\mathscr{L}_{\mathscr{D}}(h) := \mathrm{E}_{(x,y)\sim\mathscr{D}}[\mathscr{L}(h(x),y)]$$

Empirical loss :

$$\hat{\mathscr{L}}_S(h) := \frac{1}{m}\sum_{i=1}^{m}\mathscr{L}(h(x^{(i)}),y^{(i)})$$

The γ-loss $\mathscr{L} : \mathbb{R} \times \{-1,1\} \to [0,1]$:

$$\mathscr{L}(y',y) = \begin{cases} 1 & \text{if } yy' \leqslant 0 \\ 1 - \frac{yy'}{\gamma} & \text{if } yy' \leqslant (0,\gamma) \\ 0 & \text{if } yy' \geqslant 0. \end{cases}$$

Figure: γ-loss for $\gamma \in \{\gamma_0, \gamma_1, \gamma_2\}$ and $y = 1$.

Let us introduce some notations, which we will use during all the presentation. We will consider samples (X,y) which follows the random distribution D. The label y equals to -1 or 1. m denotes the size of the training sets, and a training set S is also a random variable of indepent samples following D. We note h_S the classifier learned on S, which takes its values in R. We define the exepected loss (in practice it's the test loss) by the expected value of the loss between the predicted label and the ground truth. The empirical loss is the mean of the losses.

As you can see on the right side, we use the gamma loss, for gamma greater than 0. Basically, the GD on this gamma loss encourages the predicted value to be of the same sign than the true label. Depending on gamma, it also encourages the absolute value to be large. For gamma=0, it's just the 0-1 loss.

Generalization error

Definition : Generalization error

$$\Pr_{S \sim \mathscr{D}^m}\left[\mathscr{L}_{\mathscr{D}}(h_S) - \hat{\mathscr{L}}_S(h_S) \leqslant \epsilon_{\text{gen}}(m,\delta)\right] \geqslant 1-\delta.$$

Reminder

$(X,y) \sim \mathscr{D}$
$S = \left\{(X^{(i)},y^{(i)})\right\} \sim \mathscr{D}^m$

$\mathscr{L}_{\mathscr{D}}(h) :=$
$\mathbb{E}_{(x,y)\sim\mathscr{D}}[\mathscr{L}(h(x),y)]$

$\hat{\mathscr{L}}_S(h) :=$
$\frac{1}{m}\sum_{i=1}^{m}\mathscr{L}(h(x^{(i)}),y^{(i)})$

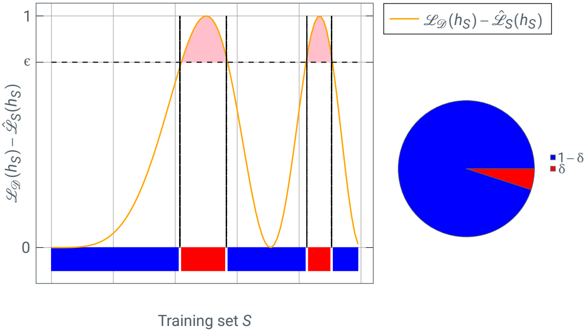
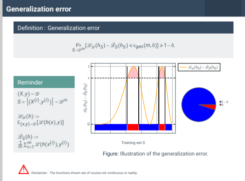


Figure: Illustration of the generalization error.

Disclaimer : The functions shown are of course not continuous in reality.



Generalization error

Definition : Generalization error

$$\Pr_{S\sim\mathscr{D}^m}[\mathscr{L}_{\mathscr{D}}(h_S)-\hat{\mathscr{L}}_S(h_S)\leqslant\epsilon_{\text{gen}}(m,\delta)]\geqslant 1-\delta.$$

Reminder

$(X,y)\sim\mathscr{D}$
$S=\left\{(X^{(i)},y^{(i)})\right\}\sim\mathscr{D}^m$

$\mathscr{L}_{\mathscr{D}}(h):=$
$\mathrm{E}_{(x,y)\sim\mathscr{D}}[\mathscr{L}(h(x),y)]$

$\hat{\mathscr{L}}_S(h):=$
$\frac{1}{m}\sum_{i=1}^{m}\mathscr{L}(h(x^{(i)}),y^{(i)})$

1
ϵ
0
$\mathscr{L}_{\mathscr{D}}(h_S)-\hat{\mathscr{L}}_S(h_S)$
Training set S
$\mathscr{L}_{\mathscr{D}}(h_S)-\hat{\mathscr{L}}_S(h_S)$
1 − δ
δ

Figure: Illustration of the generalization error.

⚠ Disclaimer : The functions shown are of course not continuous in reality.

We now define precisely the generalization error. Suppose that we have samples following the distribution D. We draw independently a training set S of m samples. So S follows D power m. for each S, we look at the expected loss of hS, minus the empirical loss of hS evaluated on S. Basically, we look at the test error minus the train error. It's represented on the plot : the various training sets S are on the x axis, and the orange curve represents the difference between the losses. Of course, there is no notion of continuity, it's just for representation purpose. The generalization error is the smallest epsilon, such that the probability, over the draw of S, that the difference is lesser than epsilon, is greater than 1-detla. These regions are colored in pink in the plot, and so we want that the part of the corresponding regions in red represent less than delta among the x axis, as indicated by the pie chart. So the generalization error depends on delta, and also on m.

Uniform convergence bound

Definition : Uniform convergence bound

$$\Pr_{S\sim\mathscr{D}^m}\left[\sup_{h\in\mathscr{H}}\left|\mathscr{L}_{\mathscr{D}}(h)-\hat{\mathscr{L}}_S(h)\right|\leqslant\epsilon_{\text{unif}}(m,\delta)\right]\geqslant 1-\delta.$$

Reminder

$(X,y)\sim\mathscr{D}$

$S=\left\{(X^{(i)},y^{(i)})\right\}\sim\mathscr{D}^m$

$\mathscr{L}_{\mathscr{D}}(h) := \mathbb{E}_{(x,y)\sim\mathscr{D}}[\mathscr{L}(h(x),y)]$

$\hat{\mathscr{L}}_S(h) := \frac{1}{m}\sum_{i=1}^m \mathscr{L}(h(x^{(i)}),y^{(i)})$

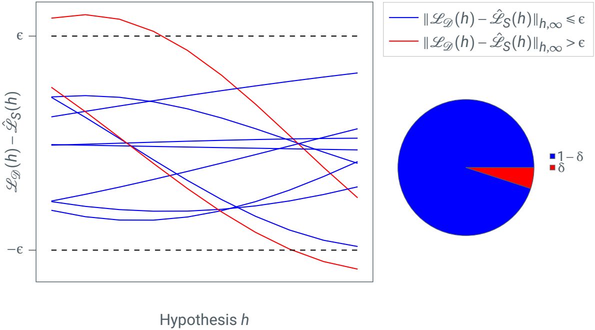
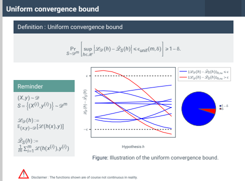


Figure: Illustration of the uniform convergence bound.

⚠ Disclaimer : The functions shown are of course not continuous in reality.



Uniform convergence bound

Definition : Uniform convergence bound

$$\Pr_{S\sim\mathscr{D}^m}\left[\sup_{h\in\mathscr{H}}\left|\mathscr{L}_{\mathscr{D}}(h)-\hat{\mathscr{L}}_S(h)\right|\leqslant\epsilon_{\text{unif}}(m,\delta)\right]\geqslant 1-\delta.$$

Reminder

$(X,y)\sim\mathscr{D}$
$S=\left\{(X^{(i)},y^{(i)})\right\}\sim\mathscr{D}^m$

$\mathscr{L}_{\mathscr{D}}(h):=$
$\mathrm{E}_{(x,y)\sim\mathscr{D}}[\mathscr{L}(h(x),y)]$

$\hat{\mathscr{L}}_S(h):=$
$\frac{1}{m}\sum_{i=1}^{m}\mathscr{L}(h(x^{(i)}),y^{(i)})$

ϵ
$-\epsilon$
$\mathscr{L}_{\mathscr{D}}(h)-\hat{\mathscr{L}}_S(h)$
Hypothesis h
$\|\mathscr{L}_{\mathscr{D}}(h)-\hat{\mathscr{L}}_S(h)\|_{h,\infty}\leqslant\epsilon$
$\|\mathscr{L}_{\mathscr{D}}(h)-\hat{\mathscr{L}}_S(h)\|_{h,\infty}>\epsilon$
$1-\delta$
δ

Figure: Illustration of the uniform convergence bound.

⚠ Disclaimer : The functions shown are of course not continuous in reality.

Then, we define the uniform convergence bound. We now watch the absolute value of the difference (it's a two sided bound). The main difference is that we have to evaluate the loss difference over all hypothesis h. So the x axis of the plot now represents the hypothesis, and each curve represents a draw of S. The difference between the loss depends on S because the empirical loss is evaluated on S. We watch the probability over S, of having the supremum of the differences between the losses, lesser than epsilon delta. It correponds in the plot to the curves in blue, which does not exceed epsilon in absolute value. We want this proportion greater than 1-delta, as indicated by the pie chart. As we have said it, it easy to see that this notion is stronger than generalization error.

Now that we have introduced these important notions, let's approach the first contribution of the article.

1 Introducing the main bounds

2 About weights and uniform convergence bounds

- A ReLU network
- Pruning the hypothesis class
- Bounds growing with *m*

3 Three examples of overparametrized models

4 Conclusion & References

5 About the exercise

A ReLU network

Parameters

Fully connected network with :

Inputs	MNIST
Depth	$d = 5$
Width	$h = 1024$
Optimizer	SGD with rate 0.1
Activations	ReLU
Loss	Crossentropy
Batch size	1

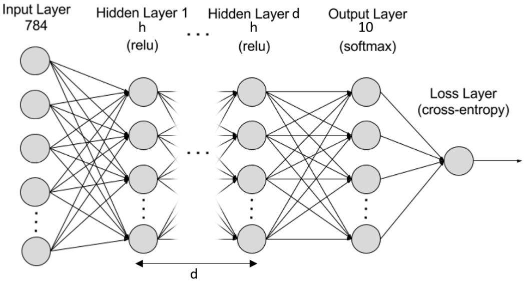
Stop criterion :

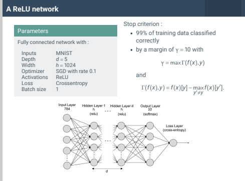
- 99% of training data classified correctly
- by a margin of $\gamma = 10$ with

$$\gamma = \max \Gamma(f(x), y)$$

and

$$\Gamma(f(x), y) = f(x)[y] - \max_{y' \neq y} f(x)[y'].$$





A ReLU network

Parameters

Fully connected network with :

Inputs	MNIST
Depth	$d = 5$
Width	$h = 1024$
Optimizer	SGD with rate 0.1
Activations	ReLU
Loss	Crossentropy
Batch size	1

Stop criterion :

- 99% of training data classified correctly
- by a margin of $\gamma = 10$ with

$$\gamma = \max \Gamma(f(x), y)$$

and

$$\Gamma(f(x), y) = f(x)[y] - \max_{y' \neq y} f(x)[y'].$$

At the beginning of the presentation, we mentioned that an important point was the behavious of the bounds regarding the evolution of the dataset size. The article study the case of a fully connected network on MNIST to approach the problem. It's a 5 layers of width 1024 neurons with ReLU activations, a crossentropy loss, trained by SGD with learning rate 0.1. They use a special stop criterion. There's of course a notion of accuracy (99%), but must of all a margin on the logits outputs, which represent how much the classifier is confident on the well classified samples. This margin is responsible (at least in part) for the generalization properties of the classifier.

First issue : need to prune the hypothesis class

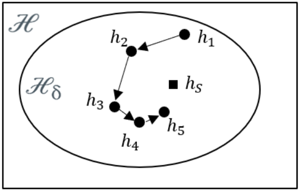
Remark

Uniform convergence bound is inadequate when $\mathscr{H}$ is large.

e.g. : Our fully connected network $|\mathscr{H}| = d \times h$.

Solution : Pruning $\mathscr{H}$.

$$\sup_{h \in \mathscr{H}} \xrightarrow[\text{implicit bias of SGD}]{\text{Taking into account}} \sup_{S \in S_\delta}$$

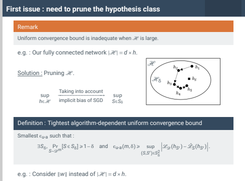


Definition : Tightest algorithm-dependent uniform convergence bound

Smallest $\epsilon_{\text{u-a}}$ such that :

$$\exists S_\delta, \Pr_{S \sim \mathscr{D}^m}[S \in S_\delta] \geq 1 - \delta \quad \text{and} \quad \epsilon_{\text{u-a}}(m, \delta) \geq \sup_{(S,S') \in S_\delta^2} \left| \mathscr{L}_{\mathscr{D}}(h_{S'}) - \hat{\mathscr{L}}_S(h_{S'}) \right|.$$

e.g. : Consider $\|w\|$ instead of $|\mathscr{H}| = d \times h$.



First issue : need to prune the hypothesis class

Remark

Uniform convergence bound is inadequate when $\mathscr{H}$ is large.

e.g. : Our fully connected network $|\mathscr{H}| = d \times h$.

Solution : Pruning $\mathscr{H}$.

$$\sup_{h \in \mathscr{H}} \xrightarrow[\text{implicit bias of SGD}]{\text{Taking into account}} \sup_{S \in S_\delta}$$

Definition : Tightest algorithm-dependent uniform convergence bound

Smallest $\epsilon_{\text{u-a}}$ such that :

$$\exists S_\delta, \Pr_{S \sim \mathscr{D}^m}[S \in S_\delta] \geq 1 - \delta \quad \text{and} \quad \epsilon_{\text{u-a}}(m, \delta) \geq \sup_{(S,S') \in S_\delta^2} \left| \mathscr{L}_{\mathscr{D}}(h_{S'}) - \hat{\mathscr{L}}_S(h_{S'}) \right|.$$

e.g. : Consider $\|w\|$ instead of $|\mathscr{H}| = d \times h$.

After the creation of the network, a first observation is made: there are too many parameters for the uniform convergence to make sense. On the one hand, most of the hypotheses will never be reached after learning. We have also seen that the implicit bias of SGD, associated to the margin, tends to favor solutions with good generalization properties. On the other hand, some training sets have very little chance of being drawn (for example a training set containing only m copies of the same sample). We therefore propose to define a new bound, by pruning the hypothesis class. The only difference is that instead of taking the supremum on all the classifiers, we take it only on the classifiers trained on the subset S_δ of the training sets. For S following D^m, we must have $S \in S_\delta$ with probability at least $1 - \delta$. This is an algorithm dependent bound, as the pruning depend on the algorithm.

Our study of uc bounds will then be continued using this tightest bound, to address this issue of large hypothesis class. In particular, this explains that we have an interest on the weights : the size of the pruned hypothesis class is measured by the potential weight norms of the network.

Bounds growing with *m*

Weights and bounds

Stop criterion implies :

$$\epsilon_{\text{u-a}} = \mathcal{O}\left(\frac{Bd\sqrt{h}}{\gamma\sqrt{m}} \prod_{k=1}^{d} \|W_k\|_2 \times \text{dist}_i \right)$$

with

$$\left[\begin{array}{l} \text{dist}_F = \sqrt{\sum_{k=1}^{d} \frac{\|W_k - Z_k\|_F^2}{\|W_k\|_2^2}} \\ \text{dist}_{2,1} = \frac{1}{d\sqrt{h}} \left(\sum_{k=1}^{d} \left(\frac{\|W_k - Z_k\|_{2,1}}{\|W_k\|_2} \right)^{2/3} \right)^{3/2} \end{array}\right. .$$

Reminder : for $A \in \mathcal{M}_{mn}$,

$$\|A\|_{2,1} = \sum_{i=1}^{m} \sqrt{\sum_{j=1}^{n} a_{ij}^2}.$$

Practical bound

In practice, $\epsilon_{\text{u-a}} = \Omega\left(m^{0.68}\right)$.

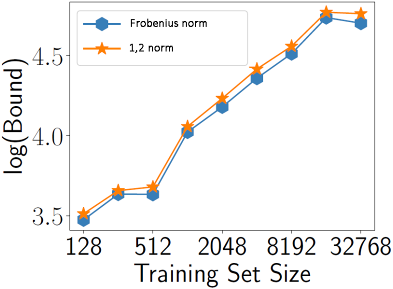
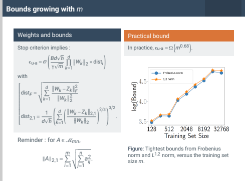


Figure: Tightest bounds from Frobenius norm and $L^{1,2}$ norm, versus the training set size *m*.



Bounds growing with m

Weights and bounds

Stop criterion implies :

$$\epsilon_{\text{u-a}} = \mathcal{O}\left(\frac{Bd\sqrt{h}}{\gamma\sqrt{m}} \prod_{k=1}^{d} \|W_k\|_2 \times \text{dist}_i\right)$$

with

$$\begin{cases} \text{dist}_F = \sqrt{\sum_{k=1}^{d} \frac{\|W_k - Z_k\|_F^2}{\|W_k\|_2^2}} \\ \text{dist}_{2,1} = \frac{1}{d\sqrt{h}} \left(\sum_{k=1}^{d} \left(\frac{\|W_k - Z_k\|_{2,1}}{\|W_k\|_2}\right)^{2/3}\right)^{3/2}. \end{cases}$$

Reminder : for $A \in \mathcal{M}_{mn}$,

$$\|A\|_{2,1} = \sum_{i=1}^{m} \sqrt{\sum_{j=1}^{n} a_{ij}^2}.$$

Practical bound

In practice, $\epsilon_{\text{u-a}} = \Omega\left(m^{0.68}\right)$.

Figure: Tightest bounds from Frobenius norm and $L^{1,2}$ norm, versus the training set size m.

Thus, from bounds on weights norms established by other articles, the authors find an estimation for the tightest uniform convergence bound, depending on parameters of the network. This estimation is obtained based on the condition satisfied by the margin. B denotes an upper bound of the inputs norms, Z_k the initialization weights. Several upper bounds are provided, depending on the chosen norm (Frobenius norm, 2-1 norm...), so depending on the way we prune the hypothesis class. Regardless of the norm, we find a practical estimation of the tightest bound of m^0.68, which means that the tightest uc bound increase with the training set size, which is the exact opposite of what we expect from a generalization bound. Then, pruning does not solve the problem of uc bounds, which have not the same behaviour than the generalization error. The authors also studied in practice the variations of the bounds with the evolution of the various parameters such as width or depth. These results are available in the appendix.

1 Introducing the main bounds

⋮

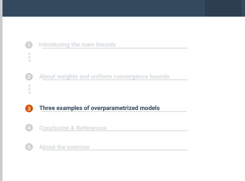
2 About weights and uniform convergence bounds

⋮

3 Three examples of overparametrized models

4 Conclusion & References

5 About the exercise

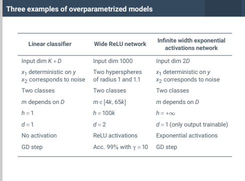


1. Introducing the main bounds
2. About weights and uniform convergence bounds
3. **Three examples of overparametrized models**
4. Conclusion & References
5. About the exercise

We will now quickly address the second part of the contribution, which concerns the detailed theoretical study of the three setups.

Three examples of overparametrized models

Linear classifier	Wide ReLU network	Infinite width exponential activations network
Input dim $K + D$	Input dim 1000	Input dim $2D$
x_1 deterministic on y x_2 corresponds to noise	Two hyperspheres of radius 1 and 1.1	x_1 deterministic on y x_2 corresponds to noise
Two classes	Two classes	Two classes
m depends on D	$m \in [4k, 65k]$	m depends on D
$h = 1$	$h = 100k$	$h = +\infty$
$d = 1$	$d = 2$	$d = 1$ (only output trainable)
No activation	ReLU activations	Exponential activations
GD step	Acc. 99% with $\gamma = 10$	GD step

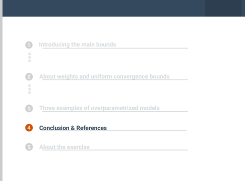


Three examples of overparametrized models

Linear classifier	Wide ReLU network	Infinite width exponential activations network
Input dim $K+D$	Input dim 1000	Input dim $2D$
x_1 deterministic on y x_2 corresponds to noise	Two hyperspheres of radius 1 and 1.1	x_1 deterministic on y x_2 corresponds to noise
Two classes	Two classes	Two classes
m depends on D	$m \in [4k, 65k]$	m depends on D
$h=1$	$h=100k$	$h=+\infty$
$d=1$	$d=2$	$d=1$ (only output trainable)
No activation	ReLU activations	Exponential activations
GD step	Acc. 99% with $\gamma=10$	GD step

The three models are binary classifiers, on various dimensional spaces. We will not go into details right away, because their study follows the same general outline, and the linear classifier is treated in the exercise. The linear setup is close to the infinite width classifier, although the latter has exponential activations, and is of infinite width. The ReLU classifier aims to separate two hyperspheres.

1. Introducing the main bounds
2. About weights and uniform convergence bounds
3. Three examples of overparametrized models
4. **Conclusion & References**
5. About the exercise



1 Introducing the main bounds

2 About weights and uniform convergence bounds

3 Three examples of overparametrized models

4 **Conclusion & References**

5 About the exercise

Let us conclude on the contributions of the article.

Conclusion

Overview

1. Dependence on *m* of the studied bounds,
2. Three setups where tightest UC bounds become nearly vacuous.

About the reviews

→ Four reviewers highlighted a thorough work.

→ Theoretical results supported by numerical results.

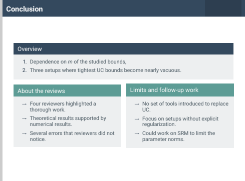
→ Several errors that reviewers did not notice.

Limits and follow-up work

→ No set of tools introduced to replace UC.

→ Focus on setups without explicit regularization.

→ Could work on SRM to limit the parameter norms.



Conclusion

Overview

1. Dependence on *m* of the studied bounds,
2. Three setups where tightest UC bounds become nearly vacuous.

About the reviews

→ Four reviewers highlighted a thorough work.
→ Theoretical results supported by numerical results.
→ Several errors that reviewers did not notice.

Limits and follow-up work

→ No set of tools introduced to replace UC.
→ Focus on setups without explicit regularization.
→ Could work on SRM to limit the parameter norms.

Thus, the article brings two main contributions. First, the necessity of studying a uniform convergence bound is highlighted, before concluding that it is insufficient because of its polynomial dependence on the size of the training set. Then, through several examples in which the absence of uniform convergence is mathematically proved, the authors bring an additional proof of the insufficiency of uniform convergence to explain generalization.

As the reviewers pointed out, this article provides a great deal of rigorous evidence, both theoretical and practical. The amount of results presented makes it sometimes difficult to identify the structure, so we hope that the presentation has been clear. We have modified a few results to correct errors not highlighted by the reviewers, particularly for the exercise. Although the article is very complete, it does not propose a more relevant tool than the uniform convergence. Also, it does not deal with explicit regularization mechanisms which, although often discarded for the study of generalization properties, allows to deal with more realistic cases. The method of Structural Risk Minimization refers to tuning the capacity of the classifier to the available amount of training data. The capacity of an algorithm is influenced by:
(1) Properties of the input space.
(2) Nature and structure of the classifier.
(3) The learning algorithm.

to improve generalization. A follow up work could then be the structural risk minimization, with explicit regularization.

Conclusion

Links and differences with the article

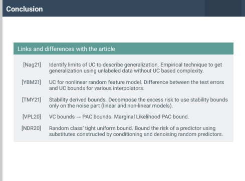
[Nag21] Identify limits of UC to describe generalization. Empirical technique to get generalization using unlabeled data without UC based complexity.

[YBM21] UC for nonlinear random feature model. Difference between the test errors and UC bounds for various interpolators.

[TMY21] Stability derived bounds. Decompose the excess risk to use stability bounds only on the noise part (linear and non-linear models).

[VPL20] VC bounds → PAC bounds. Marginal Likelihood PAC bound.

[NDR20] Random class' tight uniform bound. Bound the risk of a predictor using substitutes constructed by conditioning and denoising random predictors.



Conclusion

Links and differences with the article

[Nag21] Identify limits of UC to describe generalization. Empirical technique to get generalization using unlabeled data without UC based complexity.

[YBM21] UC for nonlinear random feature model. Difference between the test errors and UC bounds for various interpolators.

[TMY21] Stability derived bounds. Decompose the excess risk to use stability bounds only on the noise part (linear and non-linear models).

[VPL20] VC bounds → PAC bounds. Marginal Likelihood PAC bound.

[NDR20] Random class' tight uniform bound. Bound the risk of a predictor using substitutes constructed by conditioning and denoising random predictors.

Finally, we have identified the main influences that the article may have had in its field.

Overall, we can say that SGD on overparametrized networks can lead to decision boundaries that are very complex implying the failure of UC bounds to explain generalization while the generalization error is correct.

The article sets up a context of discovery for the readers. They do not give us a new box of generalization tools but point at the us the need for new ones relying on other techniques as stability. Indeed, the main point are that the usual techniques relying on UC are simply not effective to explain generalization in most cases. In order to get better results, and non-vacuous bounds, instead of working on VC bounds other articles suggest to focus on PAC bounds; try to randomize our predictors, predictors that are partially rerandomized (conditioning and denoising to get deterministic predictors) and use stability methods based on the observation that neural networks converge slowly fitting noise. So we apply stability on the noise part of the decomposed framework to explain generalization that work both for linear and non-linear models.

Several sources developing these points are listed in our bibliography.

References I

[BEHW89] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth, *Learnability and the vapnik-chervonenkis dimension*, Journal of the ACM (JACM) **36** (1989), no. 4, 929–965.

[BFT17] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky, *Spectrally-normalized margin bounds for neural networks*, Advances in neural information processing systems **30** (2017).

[Nag21] Vaishnavh Nagarajan, *Explaining generalization in deep learning: progress and fundamental limits*, arXiv preprint arXiv:2110.08922 (2021).

[NBMS17] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro, *Exploring generalization in deep learning*, Advances in neural information processing systems **30** (2017).

[NBS18] Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro, *A pac-bayesian approach to spectrally-normalized margin bounds for neural networks*, 2018.

References II

[NDR20] Jeffrey Negrea, Gintare Karolina Dziugaite, and Daniel Roy, *In defense of uniform convergence: Generalization via derandomization with an application to interpolating predictors*, International Conference on Machine Learning, PMLR, 2020, pp. 7263–7272.

[NK19] Vaishnavh Nagarajan and J Zico Kolter, *Generalization in deep networks: The role of distance from initialization*, arXiv preprint arXiv:1901.01672 (2019).

[SSSSS10] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan, *Learnability, stability and uniform convergence*, The Journal of Machine Learning Research **11** (2010), 2635–2670.

[TMY21] Jiaye Teng, Jianhao Ma, and Yang Yuan, *Towards understanding generalization via decomposing excess risk dynamics*, 2021.

[VPL20] Guillermo Valle-Pérez and Ard A Louis, *Generalization bounds for deep learning*, arXiv preprint arXiv:2012.04115 (2020).

[YBM21] Zitong Yang, Yu Bai, and Song Mei, *Exact gap between generalization error and uniform convergence in random feature models*, 2021.

Exercise : Uniform convergence and generalization bounds

Setup overview

- Dataset :

$$S = \left(x^{(i)}, y^{(i)}\right)_{i \in [\![1,m]\!]}$$

$$\in \mathbb{R}^{K+D} \times \{-1,1\}.$$

with

$$\left| \begin{array}{l} x = (x_1, x_2) \\ x_1 = 2y \cdot u \\ x_2 \sim \mathcal{N}(0, 16/D \cdot I_D). \end{array} \right.$$

- Hypothesis :

$$h(x) = w_1 \cdot x_1 + w_2 \cdot x_2$$

- Learning algorithm :

$$\max_{w} \left(h_S(x) \cdot y\right)$$

by GD.

Aim

For $\epsilon > 0$, $\delta > 0$, and D large enough,

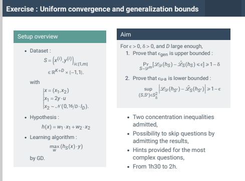
1. Prove that ϵ_{gen} is upper bounded :

$$\Pr_{S \sim \mathcal{D}^m} \left[\mathcal{L}_{\mathcal{D}}(h_S) - \hat{\mathcal{L}}_S(h_S) \leqslant \epsilon\right] \geqslant 1 - \delta$$

2. Prove that $\epsilon_{\text{u-a}}$ is lower bounded :

$$\sup_{(S,S') \in S_\delta^2} \left| \mathcal{L}_{\mathcal{D}}(h_{S'}) - \hat{\mathcal{L}}_S(h_{S'}) \right| \geqslant 1 - \epsilon$$

- Two concentration inequalities admitted,
- Possibility to skip questions by admitting the results,
- Hints provided for the most complex questions,
- From 1h30 to 2h.



Exercise : Uniform convergence and generalization bounds

Setup overview

- Dataset :
$$S=\left(x^{(i)},y^{(i)}\right)_{i\in[1,m]} \in \mathbb{R}^{K+D}\times\{-1,1\}.$$
with
$$\begin{vmatrix} x=(x_1,x_2) \\ x_1=2y\cdot u \\ x_2\sim\mathcal{N}(0,16/D\cdot I_D). \end{vmatrix}$$
- Hypothesis :
$$h(x)=w_1\cdot x_1+w_2\cdot x_2$$
- Learning algorithm :
$$\max_w (h_S(x)\cdot y)$$
by GD.

Aim

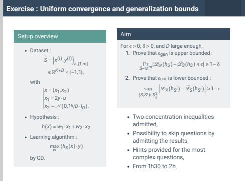
For $\epsilon>0$, $\delta>0$, and D large enough,

1. Prove that ϵ_{gen} is upper bounded :
$$\Pr_{S\sim\mathscr{D}^m}[\mathscr{L}_{\mathscr{D}}(h_S)-\hat{\mathscr{L}}_S(h_S)\leqslant\epsilon]\geqslant 1-\delta$$
2. Prove that $\epsilon_{\text{u-a}}$ is lower bounded :
$$\sup_{(S,S')\in S_\delta^2}\left|\mathscr{L}_{\mathscr{D}}(h_{S'})-\hat{\mathscr{L}}_S(h_{S'})\right|\geqslant 1-\epsilon$$

- Two concentration inequalities admitted,
- Possibility to skip questions by admitting the results,
- Hints provided for the most complex questions,
- From 1h30 to 2h.

The exercise we have chosen to design is based on a lighter version of the proof of Theorem 3.1 presented above. We present a case in which the generalisation error is arbitrarily small when we increase the number of features, while simultaneously the uniform convergence bound is arbitrarily close to 1.

The idea of the setup is as follows: we have a linear classifier on $\mathbb{R}^{(}K+D)$. Each vector x is decomposed into two vectors (x_1,x_2). x_1 is the vector that will be concretely used for classification, since the sign of the scalar product between x_1 and u directly determines the class of x. The vector x_2 follows a normal distribution, which corresponds to noise. The statement has been slightly modified at this point to correct an error in the proof, so the 32 in the article is replaced by a 16 in the standard deviation. It should be kept in mind that D is high dimensional compared to K. A linear classifier is then applied on x, which makes the scalar product between a weight w and x. Finally, the learning algorithm consists in maximising the 0-1 loss between the predicted data and our ground truth.



Exercise : Uniform convergence and generalization bounds

Setup overview

- Dataset :
$$S = \left(x^{(i)}, y^{(i)}\right)_{i\in[1,m]} \in \mathbb{R}^{K+D} \times \{-1,1\}.$$
with
$$\begin{cases} x = (x_1, x_2) \\ x_1 = 2y \cdot u \\ x_2 \sim \mathcal{N}(0, 16/D \cdot I_D). \end{cases}$$
- Hypothesis :
$$h(x) = w_1 \cdot x_1 + w_2 \cdot x_2$$
- Learning algorithm :
$$\max_w (h_S(x) \cdot y)$$
by GD.

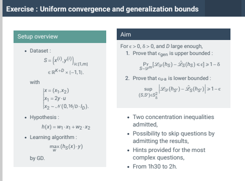
Aim

For $\epsilon > 0$, $\delta > 0$, and D large enough,

1. Prove that ϵ_{gen} is upper bounded :
$$\Pr_{S\sim\mathscr{D}^m}[\mathscr{L}_{\mathscr{D}}(h_S) - \hat{\mathscr{L}}_S(h_S) \leqslant \epsilon] \geqslant 1-\delta$$
2. Prove that $\epsilon_{\text{u-a}}$ is lower bounded :
$$\sup_{(S,S')\in S_\delta^2} \left|\mathscr{L}_{\mathscr{D}}(h_{S'}) - \hat{\mathscr{L}}_S(h_{S'})\right| \geqslant 1-\epsilon$$

- Two concentration inequalities admitted,
- Possibility to skip questions by admitting the results,
- Hints provided for the most complex questions,
- From 1h30 to 2h.

As you can see on the right block, the first part of the exercise focuses on the generalization error. So we want to show that with high probability over the draw of a training set S, the expected loss and the empirical loss, both for the classifier trained on S, are close. In the second part, we show that the tightest algorithm-dependant uniform convergence bound is greater than 1-epsilon. To do this, we only need to exhibit a pair of datasets (S,S'), such that the difference between expected loss and the empirical loss evaluated on S, both applied to the classifier train on S', are greater than 1-epsilon. The intuition of the exercise is that by adding noise via x_2, the classifier manages to succeed macroscopically, i.e. when averaging its results to obtain the generalization error. Because the generalization error is a convergence in probabilty. However, microscopic variations in the noise disturb the uniform convergence, as the supremum is more sensitive to its variations. As we show in the second part of the exercise, the probability of falling on a pair of datasets (S,S') on which the error will be too large is strictly positive, which explain the high uniform convergence bound.
Most of the questions can be done even if the others have not been passed by admitting their results, except the question 2.1 which uses the same reasoning of questions in part 1. Regarding the content of the exercise, it revolves unsurprisingly around demonstrations of inequalities on probabilities. Important concentration inequalities are presented in lemmas, and are demonstrated in the corrections. Only the knowledge of the Multivariate normal distribution, and the understanding of conditional probabilities is necessary to do the exercise.



Exercise : Uniform convergence and generalization bounds

Setup overview

- Dataset :

$$S = \left(x^{(i)}, y^{(i)}\right)_{i \in [1,m]} \in \mathbb{R}^{K+D} \times \{-1,1\}.$$

with

$$\left| \begin{array}{l} x = (x_1, x_2) \\ x_1 = 2y \cdot u \\ x_2 \sim \mathcal{N}(0, 16/D \cdot I_D). \end{array} \right.$$

- Hypothesis :

$$h(x) = w_1 \cdot x_1 + w_2 \cdot x_2$$

- Learning algorithm :

$$\max_w \left(h_S(x) \cdot y\right)$$

by GD.

Aim

For $\epsilon > 0$, $\delta > 0$, and D large enough,

1. Prove that ϵ_{gen} is upper bounded :

$$\Pr_{S \sim \mathscr{D}^m}\left[\mathscr{L}_{\mathscr{D}}(h_S) - \hat{\mathscr{L}}_S(h_S) \leqslant \epsilon\right] \geqslant 1 - \delta$$

2. Prove that $\epsilon_{\text{u-a}}$ is lower bounded :

$$\sup_{(S,S') \in S_\delta^2} \left|\mathscr{L}_{\mathscr{D}}(h_{S'}) - \hat{\mathscr{L}}_S(h_{S'})\right| \geqslant 1 - \epsilon$$

- Two concentration inequalities admitted,
- Possibility to skip questions by admitting the results,
- Hints provided for the most complex questions,
- From 1h30 to 2h.

As I said before, the theorem is slightly simplified for the proof, in two main respects. First, we do not look for exact conditions on the D-dimension for the theorem to hold, unlike the version in the paper. Secondly, we are only interested in the 0-1 loss, whereas the paper studies the gamma loss presented earlier. These simplifications make the calculations considerably lighter, but do not change the nature of the reasoning. However, the exercise is significantly more formal than the article, which changes some of the reasoning. I have also written a corrected and more formal version of the article's proof, without the simplifications of the exercise. It should be available on moodle, otherwise I can send it to you if you are interested of course.

In terms of difficulty, the questions are quite simple, and hints have been added for more complex questions. The exercise has been tested by students, and takes between 1.5 and 2 hours.