ECE826 Lecture 3:

Concentration of the Empirical Risk

Part 2: Fancy parameter counts/ complexity bounds

From This

To this

Contents

- Parameter count bounds for ERM
- VC dim and Rademacher Complexity generalization bounds
- Do these bounds explain generalization in modern ML?
- What are we missing?

Some Definitions

- Our goal is to find a hypothesis (classifier) h_S with small expected risk

$$R[h_S] = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\ell(h_S(x); y)\right]$$

- The loss measures the disagreement between predictions and reality

Some Definitions

- Our goal is to find a hypothesis (classifier) h_S with small expected risk

$$R[h_S] = \mathbb{E}_{(x,y)\sim\mathscr{D}}\left[\ell(h_S(x); y)\right]$$

- The loss measures the disagreement between predictions and reality

- Since we can't directly measure $R[h_S]$ (our true cost function), we can consider optimizing its sample-average proxy, i.e., the empirical risk

$$\hat{R}[h_S] = \frac{1}{n}\sum_{i=1}^{n} \ell(h_S(x_i); y_i)$$

- Our hope is that $\hat{R}[h_S]$ is close to $R[h_S]$

The generalization gap

- The gap of the true cost function from the one we have access to

$$\epsilon_{gen} = |R[h_S] - \hat{R}[h_S]|$$

- Question: When is it possible to bound ϵ_{gen} by a small constant?

The generalization gap

- The gap of the true cost function from the one we have access to

$$\epsilon_{gen} = |R[h_S] - \hat{R}[h_S]|$$

- Question: When is it possible to bound ϵ_{gen} by a small constant?

- The answer must depend on:
  1) n, the sample size
  2) $\mathscr{H}$, the hypothesis class (and its geometry)
  3) $\mathscr{D}$, the data distribution

  [4) the optimization algorithm that outputs our classifier]

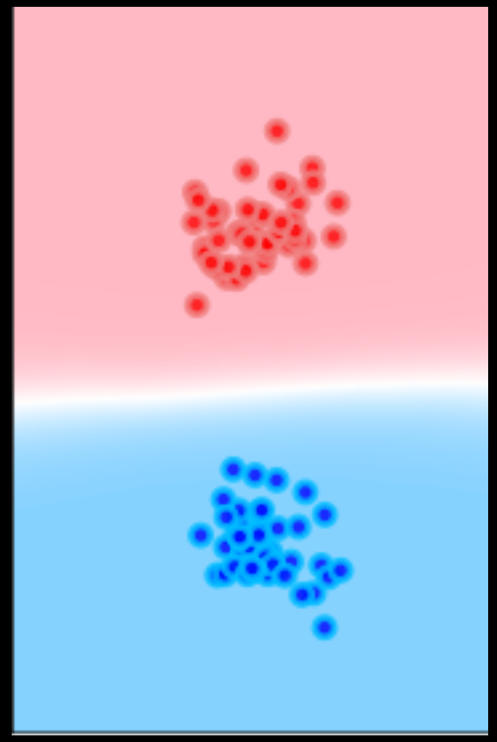
Previously: parameter count bounds

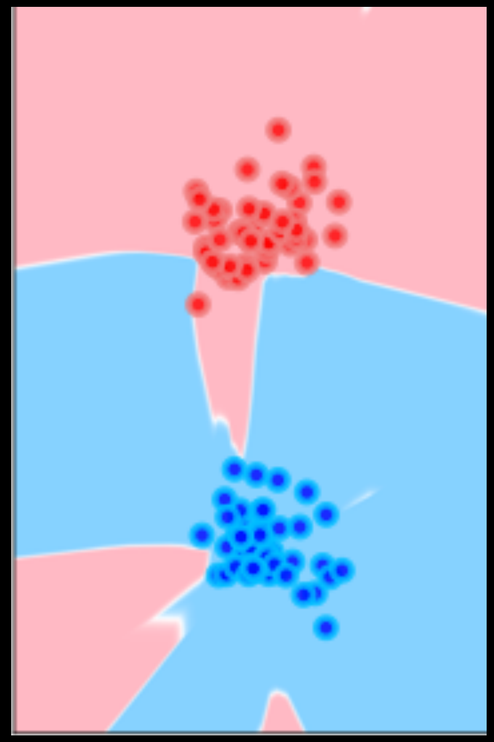
- If Floats+parametric model => n >> #params for good generalization (H.I.+Union bound over all classifiers)

Previously: parameter count bounds

- If Floats+parametric model => n >> #params for good generalization (H.I.+Union bound over all classifiers)
- Traditional theory for generalization bounds tries to handle infinite classes.
- VC-dimension, fat-shattering dimension, rademacher complexity, etc
- Can these more elaborate approaches result in interesting gen bounds for real models/data?

Measuring Complexity



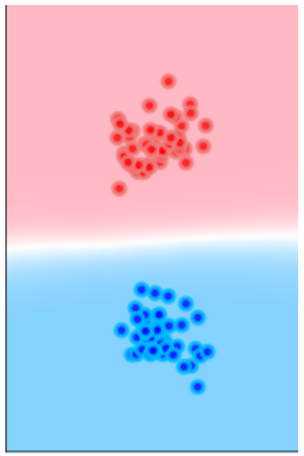


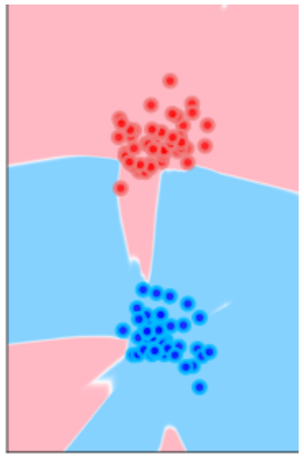
Bounding generalization via complexity measure

- General idea:

Bounding the expressiveness of a model => bounding the number of bits needed to describe it
=> bounding the generalization gap.

In other words, the less expressive/complex a class, the less surprises we'll have at test time.



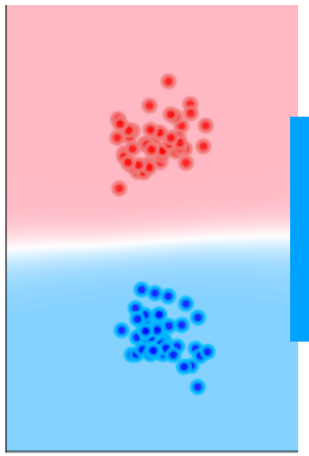


Bounding generalization via complexity measure

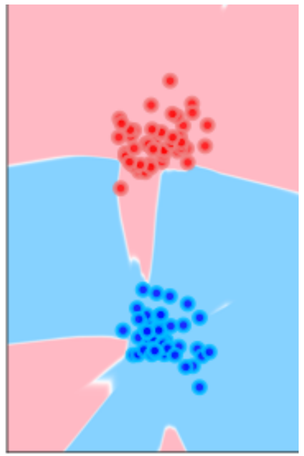
- General idea:

Bounding the expressiveness of a model => bounding the number of bits needed to describe it
=> bounding the generalization gap.

In other words, the less expressive/complex a class, the less surprises we'll have at test time.



a line that passes through points A and B

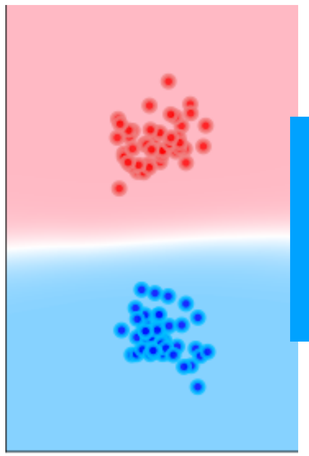


Bounding generalization via complexity measure

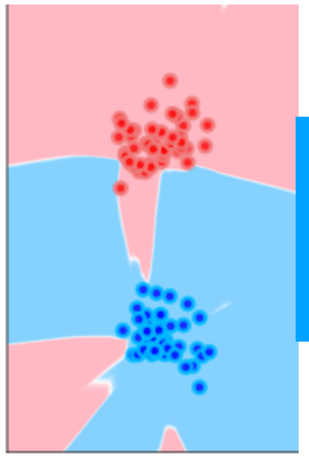
- General idea:

Bounding the expressiveness of a model => bounding the number of bits needed to describe it
=> bounding the generalization gap.

In other words, the less expressive/complex a class, the less surprises we'll have at test time.



a line that passes through points A and B



a curve that would take many more words to describe

Bounding generalization via complexity measure

- General idea:

Bounding the expressiveness of a model => bounding the number of bits needed to describe it
=> bounding the generalization gap.

In other words, the less expressive/complex a class, the less surprises we'll have at test time.

- Standard techniques: VC dimension and Rademacher Complexity

- Q: How do they work, what types of bounds do they imply?

VC dimension

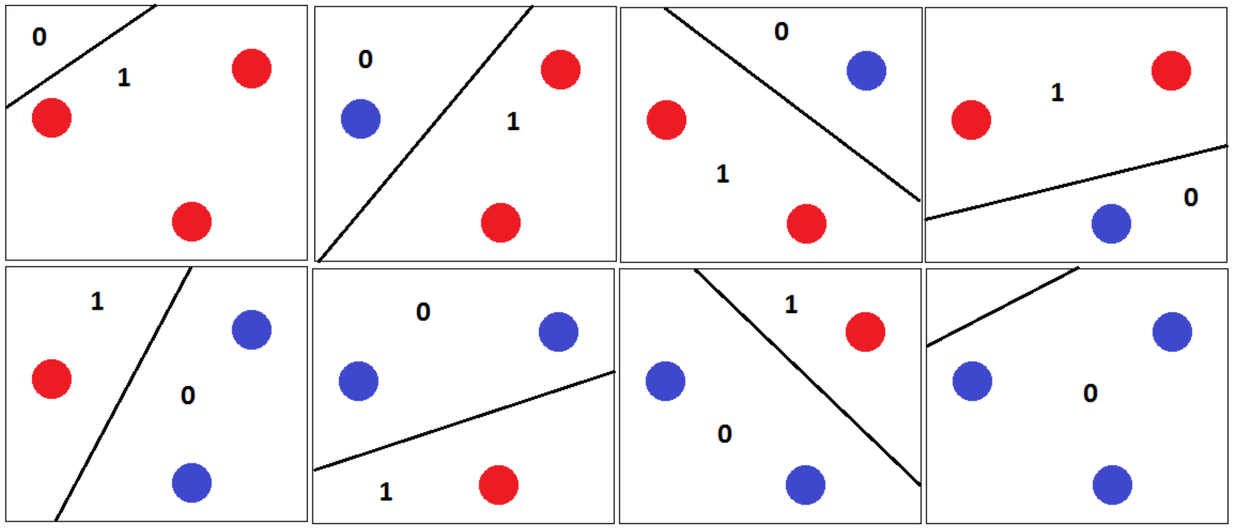
- VC dimension = measures expressiveness of a hypothesis class

Definition:
The VC-dimension of $\mathcal{H}$ is the largest number d such that there exist a set S of d samples that is shattered by a classifier $h \in \mathcal{H}$, i.e., for any labels $y_1, \ldots, y_n$ of S, $h(x_i) = y_i$ for all $x_i \in S$

VC dimension

- VC dimension = measures expressiveness of a hypothesis class

Definition:
The VC-dimension of $\mathcal{H}$ is the largest number d such that there exist a set S of d samples that is shattered by a classifier $h \in \mathcal{H}$, i.e., for any labels $y_1, \ldots, y_n$ of S, $h(x_i) = y_i$ for all $x_i \in S$



VC dimension

- VC dimension = measures expressiveness of a hypothesis class

Definition:
The VC-dimension of $\mathcal{H}$ is the largest number d such that there exist a set S of d samples that is shattered by a classifier $h \in \mathcal{H}$, i.e., for any labels $y_1, \ldots, y_n$ of S, $h(x_i) = y_i$ for all $x_i \in S$

- E.g., largest set of images that a classifier can give any set of labels.

VC dimension

- VC dimension = measures expressiveness of a hypothesis class

Definition:
The VC-dimension of $\mathcal{H}$ is the largest number d such that there exist a set S of d samples that is shattered by a classifier $h \in \mathcal{H}$, i.e., for any labels $y_1, \ldots, y_n$ of S, $h(x_i) = y_i$ for all $x_i \in S$

- E.g., largest set of images that a classifier can give any set of labels.

- Similar to memorization capacity, but not quite.

VC dimension

- VC dimension = measures expressiveness of a hypothesis class

Definition:
The VC-dimension of $\mathcal{H}$ is the largest number d such that there exist a set S of d samples that is shattered by a classifier $h \in \mathcal{H}$, i.e., for any labels $y_1, \ldots, y_n$ of S, $h(x_i) = y_i$ for all $x_i \in S$

- E.g., largest set of images that a classifier can give any set of labels.
- Similar to memorization capacity, but not quite.
- Q: how does VC connect with generalization error?

VC dimension

- VC dimension can handle infinite classes

Theorem:

For any $\epsilon, \delta > 0$, suppose that $VCdim(\mathcal{H}) = d$, and we draw a sample S of size

$$n \geq \frac{C}{\epsilon^2}\left(d\log(1/\epsilon) + \log(1/\delta)\right)$$

then with probability at least $1-\delta$, we have that $\max_{h \in \mathcal{H}} \epsilon_{gen}[h] \leq \epsilon$

VC dimension

- VC dimension can handle infinite classes

Theorem:

For any $\epsilon, \delta > 0$, suppose that $VCdim(\mathcal{H}) = d$, and we draw a sample S of size

$$n \geq \frac{C}{\epsilon^2}\left(d\log(1/\epsilon) + \log(1/\delta)\right)$$

then with probability at least $1-\delta$, we have that $\max_{h \in \mathcal{H}} \epsilon_{gen}[h] \leq \epsilon$

In fact there is a very famous theorem that says that a class cannot be "learned" in smaller than the above number of samples (in general).

VC dimension

- VC dimension can handle infinite classes

Theorem:

For any $\epsilon, \delta > 0$, suppose that $VCdim(\mathcal{H}) = d$, and we draw a sample S of size

$$n \geq \frac{C}{\epsilon^2}\left(d\log(1/\epsilon) + \log(1/\delta)\right)$$

then with probability at least $1-\delta$, we have that $\max_{h \in \mathcal{H}} \epsilon_{gen}[h] \leq \epsilon$

We need again $n > VC(\mathcal{H})$, for good generalization

Q: does this lead to non-vacuous bounds in practice?

VC dimension

- VC dimension = measure of expressiveness of a hypothesis class

Definition:
The VC-dimension of $\mathcal{H}$ is the largest number d such that there exist a set S of d samples that is shattered by a classifier $h \in \mathcal{H}$, i.e., if $y_1, \ldots, y_n$ are the labels of S, then $h(x_i) = y_i$ for all $(x_i, y_i) \in S$

VC dimension

- VC dimension = measure of expressiveness of a hypothesis class

Definition:
The VC-dimension of $\mathcal{H}$ is the largest number d such that there exist a set S of d samples that is shattered by a classifier $h \in \mathcal{H}$, i.e., if $y_1, \ldots, y_n$ are the labels of S, then $h(x_i) = y_i$ for all $(x_i, y_i) \in S$

Examples:

- $\mathcal{H} = \{h \mid h(x) = sign(w^T x - b)\}$, $VC(\mathcal{H}) = d + 1$
- $\mathcal{H}$ = neural nets with thresholds and d parameters, $VC(\mathcal{H}) = O(d \log d)$
- $\mathcal{H}$ = ReLU NNs with d parameters and depth D $VC(\mathcal{H}) = O(dD \log d)$

VC dimension

- VC dimension = measure of expressiveness of a hypothesis class

Definition:
The VC-dimension of $\mathcal{H}$ is the largest number d such that there exist a set S of d samples that is shattered by a classifier $h \in \mathcal{H}$, i.e., if $y_1, \ldots, y_n$ are the labels of S, then $h(x_i) = y_i$ for all $(x_i, y_i) \in S$

Examples:

- $\mathcal{H} = \left\{h \,|\, h(x) = sign(w^T x - b)\right\}, VC(\mathcal{H}) = d + 1$
- $\mathcal{H}$= neural nets with thresholds and d parameters, $VC(\mathcal{H}) = O(d \log d)$
- $\mathcal{H}$ = ReLU NNs with d parameters and depth D $VC(\mathcal{H}) = O(dD \log d)$

- For NNs it seems that VC dimension > #params.. Worse generalization than parameter count on FP networks…

VC dimension

- VC dimension = measure of expressiveness of a hypothesis class

Definition:
The VC-dimension of $\mathcal{H}$ is the largest number d such that there exist a set S of d samples that is shattered by a classifier $h \in \mathcal{H}$, i.e., if $y_1, \ldots, y_n$ are the labels of S, then $h(x_i) = y_i$ for all $(x_i, y_i) \in S$

Examples:

- $\mathcal{H} = \{h \mid h(x) = sign(w^T x - b)\}, VC(\mathcal{H}) = d + 1$
- $\mathcal{H}$ = neural nets with thresholds and d parameters, $VC(\mathcal{H}) = O(d \log d)$
- For finite NNs with d parameters and finite D, $VC(\mathcal{H}) = O(d D \log d)$

For finite Precision param count, VC doesn't lead to anything better than the simple UB technique from earlier…

- For NNs it seems that VC dimension > #params.. Worse generalization than parameter count on FP networks…

VC dimension

- VC dimension = measure of expressiveness of a hypothesis class

Definition:
The VC-dimension of $\mathcal{H}$ is the largest number d such that there exist a set S of d samples that is shattered by a classifier $h \in \mathcal{H}$, i.e., if $y_1, \ldots, y_n$ are the labels of S, then $h(x_i) = y_i$ for all $(x_i, y_i) \in S$

Examples:

- $\mathcal{H} = \{h \mid h(x) = sign(w^T x - b)\}, VC(\mathcal{H}) = d + 1$
- $\mathcal{H}$ = neural nets with thresholds and d parameters, $VC(\mathcal{H}) = O(d \log d)$
- $\mathcal{H}$ = ReLU NNs with d parameters and depth D $VC(\mathcal{H}) = O(dD \log d)$

Downsides of VC: talks about the worst possible set of data points, rather than a typical one. Also looks at the most expressive classifier in our set.

- For NNs it seems that VC dimension > #params.. Worse generalization than parameter count on FP networks…

VC dimension

- VC dimension = measure of expressiveness of a hypothesis class

Definition:
The VC-dimension of $\mathcal{H}$ is the largest number d such that there exist a set S of d samples that is shattered by a classifier $h \in \mathcal{H}$, i.e., if $y_1, \ldots, y_n$ are the labels of S, then $h(x_i) = y_i$ for all $(x_i, y_i) \in S$

Examples:

- $\mathcal{H} = \{h \mid h(x) = sign(w^T x - b)\}, VC(\mathcal{H}) = d + 1$
- $\mathcal{H}$ = neural nets with thresholds and d parameters, $VC(\mathcal{H}) = O(d \log d)$
- $\mathcal{H}$ = ReLU NNs with d parameters and depth D $VC(\mathcal{H}) = O(dD \log d)$

Can we improve by incorporating compression arguments?

- For NNs it seems that VC dimension > #params.. Worse generalization than parameter count on FP networks…

Refining parameter counts by a compression argument

Getting more out of param. counts

- Let's assume that our bag of classifiers is "compressible"

Assumption (hypothetical):

Assume that every model in $\mathscr{H}$ (infinite class) can be mapped to a model in $\mathscr{H}_\delta$, which can be described by p fixed precision parameters, at the cost of δ in overall loss.

Getting more out of param. counts

- Let's assume that our bag of classifiers is "compressible"

Assumption (hypothetical):

Assume that every model in $\mathscr{H}$ (infinite class) can be mapped to a model in $\mathscr{H}_\delta$, which can be described by p fixed precision parameters, at the cost of δ in overall loss.

Let's sketch this picture:

Getting more out of param. counts

- Let's assume that our bag of classifiers is "compressible"

Assumption (hypothetical):

Assume that every model in $\mathscr{H}$ (infinite class) can be mapped to a model in $\mathscr{H}_\delta$, which can be described by p fixed precision parameters, at the cost of δ in overall loss.

Lemma:

Let b be an upper bound on the abs. value of the params used to represent a model in $\mathscr{H}$. Then,

$$\max_{h \in \mathscr{H}} \epsilon_{gen}[h] \leq \sqrt{\frac{p \cdot \log b}{n}} + \delta$$

Getting more out of param. counts

- Let's assume that our bag of classifiers is "compressible"

Assumption (hypothetical):

Assume that every model in $\mathscr{H}$ (infinite class) can be mapped to a model in $\mathscr{H}_\delta$, which can be described by p fixed precision parameters, at the cost of δ in overall loss.

Lemma:

Let b be an upper bound on the abs. value of the params used to represent a model in $\mathscr{H}$. Then,

$$\max_{h \in \mathscr{H}} \epsilon_{gen}[h] \leq \sqrt{\frac{p \cdot \log b}{n}} + \delta$$

- Proof: Hoeffding's Inequality + Union bound over all possible $O(C^p)$ models ($C = o(1)$)

Getting more out of param. counts

- Let's assume that our bag of classifiers is "compressible"

Assumption (hypothetical):

Assume that every model in $\mathscr{H}$ (infinite class) can be mapped to a model in $\mathscr{H}_\delta$, which can be described by p fixed precision parameters, at the cost of δ in overall loss.

Lemma:

Let b be an upper bound on the abs. value of the params used to represent a model in $\mathscr{H}$. Then,

$$\max_{h \in \mathscr{H}} \epsilon_{gen}[h] \leq \sqrt{\frac{p \cdot \log b}{n}} + \delta$$

- Proof: Hoeffding's Inequality + Union bound over all possible $O(C^p)$ models ($C = o(1)$)
- Why is this useful?

Getting more out of param. counts

- Let's assume that our bag of classifiers is "compressible"

Assumption (hypothetical):

Assume that every model in $\mathscr{H}$ (infinite class) can be mapped to a model in $\mathscr{H}_\delta$, which can be described by p fixed precision parameters, at the cost of δ in overall loss.

Lemma:

Let b be an upper bound on the abs. value of the params used to represent a model in $\mathscr{H}$. Then,

$$\max_{h \in \mathscr{H}} \epsilon_{gen}[h] \leq \sqrt{\frac{p \cdot \log b}{n}} + \delta$$

- Proof: Hoeffding's Inequality + Union bound over all possible $O(C^p)$ models ($C = o(1)$)
- Why is this useful?

Nothing insightful so far

NNs may be very compressible!

- Let's assume that we are working with a FC network of ReLUs
- W is the width, and D the depth, $A_i \in \mathbb{R}^{d_i \times d_{i+1}}$ the weight matrix of layer i

NNs may be very compressible!

- Let's assume that we are working with a FC network of ReLUs
- W is the width, and D the depth, $A_i \in \mathbb{R}^{d_i \times d_{i+1}}$ the weight matrix of layer i

Lemma:
[Braverman et al, COLT21 http://proceedings.mlr.press/v134/braverman21b/braverman21b.pdf]

Every weight matrix matrix $A_i \in \mathbb{R}^{d_i \times d_{i+1}}$ can be approximated by a sparse matrix $\hat{A}_i$ such that $\|A_i - \hat{A}_i\| \leq \epsilon \|A_i\|$ with expected sparsity

NNs may be very compressible!

- Let's assume that we are working with a FC network of ReLUs
- W is the width, and D the depth, $A_i \in \mathbb{R}^{d_i \times d_{i+1}}$ the weight matrix of layer i

Lemma:
[Braverman et al, COLT21 http://proceedings.mlr.press/v134/braverman21b/braverman21b.pdf]

Every weight matrix matrix $A_i \in \mathbb{R}^{d_i \times d_{i+1}}$ can be approximated by a sparse matrix $\hat{A}_i$ such that $\|A_i - \hat{A}_i\| \leq \epsilon \|A_i\|$ with expected sparsity

$$\tilde{O}\left(\frac{\text{ns}(A_i) \cdot \text{sr}(A_i)}{\epsilon^2} + \frac{\sqrt{d_{i+1} \cdot \text{ns}(A_i) \cdot \text{sr}(A_i)}}{\epsilon}\right)$$

where $\text{ns}(A) = \|A\|_1^2 / \|A\|_F^2$ (numerical sparsity) and $\text{sr}(A) = \|A\|_F^2 / \|A\|_2^2$ (stable rank)

NNs may be very compressible!

TL;DR: if matrix is approximately sparse/low-rank, we can through away many elements.

Corollary:
[Braverman et al, COLT21 http://proceedings.mlr.press/v134/braverman21b/braverman21b.pdf]

Total number of effective parameters

$$p_{small}(\epsilon) = \tilde{O}\left(\sum_{i=1}^{D}\left(\text{ns}(A_i)\cdot\text{sr}(A_i)/\epsilon^2 + \sqrt{d_{i+1}\cdot\text{ns}(A_i)\cdot\text{sr}(A_i)}/\epsilon\right)\right)$$

where $\text{ns}(A) = \|A\|_1^2/\|A\|_F^2$ (numerical sparsity) and $\text{sr}(A) = \|A\|_F^2/\|A\|_2^2$ (stable rank)

NNs may be very compressible!

TL;DR: if matrix is approximately sparse/low-rank, we can through away many elements.

Corollary:
[Braverman et al, COLT21 http://proceedings.mlr.press/v134/braverman21b/braverman21b.pdf]

Total number of effective parameters

$$p_{small}(\epsilon) = \tilde{O}\left(\sum_{i=1}^{D}\left(\text{ns}(A_i)\cdot\text{sr}(A_i)/\epsilon^2 + \sqrt{d_{i+1}\cdot\text{ns}(A_i)\cdot\text{sr}(A_i)}/\epsilon\right)\right)$$

where $\text{ns}(A) = \|A\|_1^2/\|A\|_F^2$ (numerical sparsity) and $\text{sr}(A) = \|A\|_F^2/\|A\|_2^2$ (stable rank)

This may be much smaller than $D \cdot W^2$

What does that lemma mean for $\mathscr{H}$

Lemma:
Assume that all spectral norms for all weight matrices is less than 1. Then, any model in $\mathscr{H}$ can be replaced by one that has $p_{small}(\epsilon)$ parameters, and leads to output error

$$\sup_{\|x\|_2 \leq 1} \|f(x) - \hat{f}(x)\|_2 \leq O(D\epsilon)$$

where $f(x) = W_D\sigma(W_{D-1}\sigma(\ldots W_2\sigma(W_1 x)))$

- Proof sketch (2-layers):

What does that lemma mean for $\mathscr{H}$

Lemma:
Assume that all spectral norms for all weight matrices is less than 1. Then, any model in $\mathscr{H}$ can be replaced by one that has $p_{small}(\epsilon)$ parameters, and leads to output error

$$\sup_{\|x\|_2 \leq 1} \|f(x) - \hat{f}(x)\|_2 \leq O(D\epsilon)$$

where $f(x) = W_D\sigma(W_{D-1}\sigma(\ldots W_2\sigma(W_1 x)))$

- Hence, we'd have $p_{small}(\epsilon/D)$ for an error of $O(\epsilon)$

Ok but that doesn't lead to finite H?

- Next step: Replace each of the parameters of the compressed model with a quantized version.

Proposition:
When the spectral norm is bounded by a constant c, all elements of the sparsified weight matrices will also be bounded by a constant. Every weight can then be replaced by a quantized version of itself such that $\|A_{sparsified} - A^{q}_{sparsified}\| \leq \epsilon$, as long as we use $O(log(W/\epsilon))$ bits of precision

Ok but that doesn't lead to finite H?

- Next step: Replace each of the parameters of the compressed model with a quantized version.

Proposition:
When the spectral norm is bounded by a constant c, all elements of the sparsified weight matrices will also be bounded by a constant. Every weight can then be replaced by a quantized version of itself such that $\|A_{sparsified} - A^{q}_{sparsified}\| \leq \epsilon$, as long as we use $O(log(W/\epsilon))$ bits of precision

- Q: But how do we represent each of these networks:
  A: total parameters = sum of all sparsities*precision + positions of elements*precision

$$\tilde{O}\left(\sum_{i=1}^{D}\left(\text{ns}(A_i)\cdot\text{sr}(A_i)/\epsilon^2 + \sqrt{d_{i+1}\cdot\text{ns}(A_i)\cdot\text{sr}(A_i)}/\epsilon\right)\right)$$

Ok but that doesn't lead to finite H?

- Next step: Replace each of the parameters of the compressed model with a quantized version.

Proposition:
When the spectral norm is bounded by a constant c, all elements of the sparsified weight matrices will also be bounded by a constant. Every weight can then be replaced by a quantized version of itself such that $\|A_{sparsified} - A^{q}_{sparsified}\| \leq \epsilon$, as long as we use $O(log(W/\epsilon))$ bits of precision

- Q: But how do we represent each of these networks:
  A: total parameters = sum of all sparsities*precision + positions of elements*precision

$$\tilde{O}\left(\sum_{i=1}^{D}\left(\text{ns}(A_i)\cdot\text{sr}(A_i)/\epsilon^2 + \sqrt{d_{i+1}\cdot\text{ns}(A_i)\cdot\text{sr}(A_i)}/\epsilon\right)\right)$$

- Remember though we need ϵ/D in the bound above. Assume that all widths are the same (keep th
  second term for simplicity), we obtain $\tilde{O}\left(\frac{D}{\epsilon}\sum_{i=1}^{D}\sqrt{W\cdot\text{ns}(A_i)\cdot\text{sr}(A_i)}\right)$

Final step, choose ε

- It should be relatively clear at this point that any classifier in $\mathscr{H}$ can be mapped to one in $\mathscr{H}_\delta$, which represents all quantized+sparsified models. Therefore the generalization gap should be

Lemma:

Let p_δ be an upper bound on the value of the parameters required to represent a model in $\mathscr{H}_\delta$. Then,

$$\max_{h \in \mathscr{H}} \epsilon_{gen}[h] \leq \sqrt{\frac{p_\delta \cdot \log b}{n}} + \delta$$

Final step, choose ε

- It should be relatively clear at this point that any classifier in $\mathscr{H}$ can be mapped to one in $\mathscr{H}_\delta$, which represents all quantized+sparsified models. Therefore the generalization gap should be

Lemma:

Let p_δ be an upper bound on the value of the parameters required to represent a model in $\mathscr{H}_\delta$. Then,

$$\max_{h \in \mathscr{H}} \epsilon_{gen}[h] \leq \sqrt{\frac{p_\delta \cdot \log b}{n}} + \delta$$

The above requires us to use $p_\delta = \tilde{O}\left(\frac{D}{\delta} \sum_{i=1}^{D} \sqrt{W \cdot \text{ns}(A_i) \cdot \text{sr}(A_i)}\right)$ parameters. If δ is decaying then this leads to "suboptimal" rates, yet still gives non trivial bounds.

Final step, choose ε

- It should be relatively clear at this point that any classifier in $\mathscr{H}$ can be mapped to one in $\mathscr{H}_\delta$, which represents all quantized+sparsified models. Therefore the generalization gap should be

Lemma:

Let p_δ be an upper bound on the value of the parameters required to represent a model in $\mathscr{H}_\delta$. Then,

$$\max_{h \in \mathscr{H}} \epsilon_{gen}[h] \leq \sqrt{\frac{p_\delta \cdot \log b}{n}} + \delta$$

The above requires us to use $p_\delta = \tilde{O}\left(\frac{D}{\delta} \sum_{i=1}^{D} \sqrt{W \cdot \text{ns}(A_i) \cdot \text{sr}(A_i)}\right)$ parameters. If δ is decaying then this leads to "suboptimal" rates, yet still gives non trivial bounds.

we don't need n>> #params anymore

Very similar in spirit to

Stronger Generalization Bounds for Deep Nets via a Compression Approach

Sanjeev Arora [1] **Rong Ge** [2] **Behnam Neyshabur** [3] **Yi Zhang** [1]

Abstract

Deep nets generalize well despite having more parameters than the number of training samples. Recent works try to give an explanation using PAC-Bayes and Margin-based analyses, but do not as yet result in sample complexity bounds better than naive parameter counting. The current paper shows generalization bounds that are orders of magnitude better in practice. These rely upon new succinct reparametrizations of the trained net — a compression that is explicit and efficient. These yield generalization bounds via a simple compression-based framework introduced here. Our results also provide some theoretical justification for widespread empirical success in compressing deep nets. Analysis of correctness of our compression relies upon some newly identified "noise stability"properties of trained deep nets, which are also experimentally verified. The study of these properties and resulting generalization bounds are also extended to convolutional nets, which had eluded earlier attempts on proving generalization.

fueled research in this area by showing experimentally that standard architectures using SGD and regularization can still reach low training error on randomly labeled examples (which clearly won't generalize).

Clearly, deep nets trained on real-life data have some properties that reduce effective capacity, but identifying them has proved difficult —at least in a *quantitative* way that yields sample size upper bounds similar to classical analyses in simpler models such as SVMs (Bartlett and Mendelson, 2002; Evgeniou et al., 2000; Smola et al., 1998) or matrix factorization (Fazel et al., 2001; Srebro et al., 2005).

Qualitatively (Hochreiter and Schmidhuber, 1997; Hinton and Van Camp, 1993) suggested that nets that generalize well are *flat minima* in the optimization landscape of the training loss. Recently Keskar et al. (2016) show using experiments with different batch-sizes that sharp minima do correlate with higher generalization error. A quantitative version of "flatness" was suggested in (Langford and Caruana, 2001): the net's output is stable to *noise* added to the net's trainable parameters. Using PAC-Bayes bound (McAllester, 1998; 1999) this noise stability yielded generalization bounds for fully connected nets of depth 2. The theory has been extended to multilayer fully connected nets (Neyshabur et al., 2017b), although thus far yields sam-

Another way of thinking about this

- If among the classifiers in $\mathscr{H}$ there is "large correlation", we should not pay for it

- Equivalent to the idea that each event in the union bound can be very dependent to other events (e.g., h_1 is bad may imply h_2 is very bad!)

Open: Ways to improve?

- What if training data are compressible? (some very old papers on this)
- Compression beyond sparsity/rank (info theoretic approaches?)
- How far can we go with this?

Do the above explain generalization?

Next time: generalization though an algorithmic lens

Next time

- The gap of the true cost function from the one we have access to

$$\epsilon_{gen} = |R[h_S] - \hat{R}[h_S]|$$

- Question: When is it possible to bound ϵ_{gen} by a small constant?

- The answer must depend on:
  1) n, the sample size
  2) $\mathscr{H}$, the hypothesis class (and its geometry)
  3) $\mathscr{D}$, the data distribution

  [4) the optimization algorithm that outputs our classifier]

Conclusion

- Algorithm/Data agnostic generalization bounds are… tricky
- Can they explain the good performance of large models?
- Next: Generalization beyond "parameter counts" & complexity