1. Abstract

We give a covering number bound for deep learning networks that is independent of the size of the network. The key for the simple analysis is that for linear classifiers, rotating the data doesn’t affect the covering number. Thus, we can ignore the rotation part of each layer’s linear transformation, and get the covering number bound by concentrating on the scaling part.

2. Introduction

The generalization performance of deep learning networks is puzzling because conventional wisdom suggests that models with large number of parameters should not generalize well. However, the neural network models are known to explore only limited part of parameter space by keeping the spectral norm of linear transformations under control [2]. With our analysis we show that if the spectral norm of the layers is tight, then the layers of the network do not increase the model complexity and are likely to actually reduce it. This explains the observed behavior that deeper networks often performing better than shallower ones.

3. Covering Number

Definition [3]: Let $B$ be a metric space with metric $\psi$. Given observations $X^n = [x_1, \ldots, x_n]$, and vectors $f(\alpha, X^n) = [f(\alpha, x_1), \ldots, f(\alpha, x_n)] \in B^n$ parametrized by $\alpha$, the covering number in $p$-norm, denoted as $\mathcal{N}_p(f, \epsilon, X^n)$, is the minimum number $m$ of a collection of vectors $v_1, v_2, \ldots, v_m \in B^n$, such that $\forall \alpha, \exists v_j:

$$
\|\psi(f(\alpha, X^n), v_j)\|_p = \left[ \sum_{i=1}^{n} \psi(f(\alpha, x_i), v_j^i)^p \right]^{1/p} \leq n^{1/p} \epsilon.
$$

Note that for $p = \infty$, this corresponds to margin bounds.

4. A Simple Network

Consider a deep neural network defined as in [2] i.e., $K = w(\sigma_1(A_1 \sigma_{l-1}(A_{l-1} \cdot \cdot \cdot \sigma_1(A_1(x)) \cdot \cdot \cdot )))$, but where the $A_i$’s are constrained to be orthonormal i.e., $A_i^T A_i = I$, and each non-linearity $\sigma_i$ is ReLU, and $w$ is the final projection for binary classification. Also, $x \in X$ and $\|x\| \leq b$. 

In \cite{3}, the infinity norm covering number $N_\infty$ for linear functions for $w(x) = w \cdot x$ was bounded by,

\begin{equation}
\log(N_\infty(L(w(\cdot),x),\epsilon,n)) \leq 36\frac{a^2b^2}{\epsilon} \log(8abn/\epsilon + 4n + 1),
\end{equation}

where $\|w\| \leq a$, $L$ is the set of all the outputs that can be obtained by operations $w \cdot x$, and $n$ is the number of input examples.

Now, suppose we transform the input $x$ before taking the dot-product by using ReLUs, i.e., consider the covering number of $N_\infty(L(w(\sigma(\cdot)),x),\epsilon,n)$. As shown in \cite{2}, $\sigma$ is 1-Lipschitz. From \cite{1}, we know that if $F$ is $f$-Lipschitz,

\begin{equation}
N(F(x),\epsilon,n) \leq N(x,\epsilon/f,n).
\end{equation}

Hence,

\begin{equation}
N_\infty(L(w(\sigma(\cdot)),x),\epsilon,n) \leq N_\infty(L(w(\cdot),x),\epsilon,n)
\end{equation}

In other words, since $\sigma(X) \subseteq X$, applying ReLUs cannot increase the covering number.

As for transforming $x$ by $A$, since $Ax$ is simply a rotation operation transforming $x$ by $A$, it will keep the the covering number the same because $\|w \cdot (Ax - Ax')\| = \|w' \cdot (x - x')\|$ for $w' = wA$.

Since both ReLUs and transforming by $A_i$ doesn’t affect the covering number,

\begin{equation}
\log(N_\infty(K(\cdot,x),\epsilon,n)) \leq 36\frac{a^2b^2}{\epsilon^2} \log(8abn/\epsilon + 4n + 1)
\end{equation}

This bound also becomes clear when we realize that each layer of the network is simply applying ReLUs along different directions, which is nothing but projecting data repeatedly down to a smaller subspace.

5. General Networks

For general networks, $A_i$ can be any matrix of appropriate size but we can however build on earlier analysis by decomposing $A_i$ as $U_i \Lambda_i V_i^T$, where $U_i$ and $V_i$ are orthonormal matrices and $\Lambda_i$ is diagonal. The only missing piece in computing the covering number is how does scaling data along difference axis independently affects the covering number. Let $\rho_i = \max(\Lambda_i)$ be the spectral norm of the matrix $A_i$. $\Lambda_i x$ is then at most $\rho_i$-Lipschitz along every coordinate, and hence from \cite{2} $\Lambda_i x$ is $\rho_i$-Lipschitz overall. Hence, from \cite{1},

\begin{equation}
N_\infty((\Lambda_i(\cdot),x),\epsilon,n) \leq N_\infty(x,\epsilon/\rho_i,n)
\end{equation}

Hence, if we let $r = \rho_1 \rho_2 \cdots \rho_l$

\begin{equation}
\log(N_\infty(K(\cdot,x),\epsilon,n)) \leq 36\frac{a^2b^2r^2}{\epsilon^2} \log(8abr'n/\epsilon + 4n + 1).
\end{equation}

This can be further tightened by replacing $r$ by $r' = \|A_l \cdots A_2 A_1\|_\rho$ which is the spectral norm of the combined transformation. Then,

\begin{equation}
\log(N_\infty(K(\cdot,x),\epsilon,n)) \leq 36\frac{a^2b^2r'^2}{\epsilon^2} \log(8abr'n/\epsilon + 4n + 1).
\end{equation}
This is because $A(\sigma(x)) \subseteq A(x)$ and $\sigma(A(x)) \subseteq A(x)$ and hence $\sigma_l(A_l\sigma_{l-1}(A_{l-1} \cdots \sigma_1(A_1(x)) \cdots)) \subseteq (A_l(A_{l-1} \cdots A_1(x) \cdots))$.

6. Discussion and Conclusion

The bounds presented here are tighter than earlier bounds [2] because the earlier bounds depended upon $L_1$ norms of $A_i$ and as a result were not independent of the network size.

The bounds are as tight as margin bounds are for linear classifiers. We believe the bound can be further tightened by bounding the spectral norm of $\|A_l \cdots A_2 A_1(X)\|_\rho$ i.e., if the eigen-vectors of data are not aligned with the eigen-vectors of transformation then the bound will be tighter.

The bounds explains why deep networks generalize well, but does not explain why they generalize better when compared to linear classifier. An important reason for the difference could be the convolutional structure of the network. The constraint imposed by convolutional structure that that the same operations are applied independent of location clearly limits the parameter space available. Determining exactly how the constraint affects the bound could give insights on how to build the networks.

We hope that the simplicity of the bound and its proof will provide insights to practitioners to develop faster and more accurate networks.

References

[1] Peter Bartlett. http://www.stat.berkeley.edu/bartlett/courses/2013spring-stat210b/notes/12notes.pdf. page 10.

[2] Matus Telgarsky Peter Bartlett, Dylan Foster. Spectrally-normalized margin bounds for neural networks. NIPS, 2017.

[3] Tong Zhang. Covering number bounds of certain regularized linear function classes. Journal of Machine Learning Research, 2, 2002.