Generalization Ideas in Deep Learning

Marcel Bruckner

Seminar: Optimization and Generalization in Deep Learning

Abstract

Trying to understand the generalization abilities of deep neural networks several capacity measures are experimentally explored. Following the thoughts of Neyshabur et al. [7] different norm-based capacity measures on the network weights and sharpness as the robustness to perturbations on the parameter space are investigated. The measures are used in different experiments and the results are plotted against each other.

1 Introduction

In recent years we have seen the huge empirical success of deep neural networks. Even though they solve complex, highly non-convex optimization problems simple stochastic gradient descent methods achieve good generalization behavior in practice. We see that even in the highly over parameterized regime in which deep neural networks are located many of the local minima achieve good generalization on unseen data. Nonetheless, it remains mostly unclear why they exhibit the desired generalization behavior.
Neyshabur et al. [7] provided us with a methodology to investigate into measures of the generalization ability. Using this framework we tried to reproduce their results and to understand how different measures on the parameter space can explain certain empirical phenomena that are directly connected to generalization abilities of deep neural networks.

2 Capacity control

Neyshabur et al. [7] examined complexity measures that have recently been suggested, or could be considered, in explaining generalization in deep learning. They evaluated the measures based on their ability to theoretically guarantee generalization, and their empirical ability to explain the phenomena discussed in Section 3.

2.1 Norm-based capacity control

For deep neural networks a multitude of norm-based regularization techniques on the network weights have been established. Different approaches, e.g. the $l1/l2$ norms, their respective path norms [8] or the spectral and nuclear norm are used to control the capacity of networks independent of the number of parameters.

Margin To meaningful compare norms of deep neural networks we have to explicitly take into account the scaling of the outputs of the network. As we drive the training error to zero we have to push the cross entropy loss to zero and thus the outputs of the network must go to infinity. This means that minimizing the cross entropy loss will drive the norms towards infinity.
In practice, we will stop the training after some finite time and the norm of the network will be large but also finite. The resulting norm value will mostly be an indicator on how far the training has progressed.
To overcome this scaling issues [7] suggested using the margin over the whole training set, where the margin for a single data point is the difference between the score of the correct label and the maximum score of other labels $f_w(x)[y_{true}] - max_{x \neq y_{true}} f_w(x)[y]$. As taking the minimum margin over the whole training set is really sensitive to outliers [7] defined the γ_{margin} as the lowest value of γ such that $\lceil \epsilon m \rceil$ data points have margin lower than γ, where m is the size of the training set and $\epsilon > 0$ is small.

Norms The measures investigated by [7] and the corresponding capacity bounds are as follows:

- $l2$ norm with capacity proportional to $\frac{1}{\gamma^2_{margin}} \prod_{i=1}^{d} 4 \left\| W_i \right\|_F^2$ (1)
- $l1$-path norm with capacity proportional to $\frac{1}{\gamma^2_{margin}} \left| \sum_{j \in \prod_{k=0}^{d}[h_k]} \left| \prod_{i=1}^{d} 2W_i[j_i, j_{i-1}] \right| \right|^2$ (2)
- $l2$-path norm with capacity proportional to $\frac{1}{\gamma^2_{margin}} \sum_{j \in \prod_{k=0}^{d}[h_k]} \prod_{i=1}^{d} 4h_i W_i^2[j_i, j_{i-1}]$ (3)
- spectral norm with capacity proportional to $\frac{1}{\gamma^2_{margin}} \prod_{i=1}^{d} h_i \left\| W_i \right\|_2^2$ (4)

where $\prod_{i=1}^{d} \dots$ is the product over all network layers, h_i is the number of hidden units in layer i and $\prod_{k=0}^{d}[h_k] \dots$ is the Cartesian product over sets $[h_k]$ and displays the enumeration of all paths through the network from input to output nodes.

2.2 Sharpness

Keshkar er al. [4] recently suggested sharpness as a generalization measure that corresponds to robustness to adversarial perturbations on the parameter space:

$$\zeta_\alpha(w) = \frac{\max_{|\nu_i| \leq \alpha(|w_i|+1)} \hat{L}(f_{w+\nu}) - \hat{L}(f_w)}{1 + \hat{L}(f_w)} \cong \max_{|\nu_i| \leq \alpha(|w_i|+1)} \hat{L}(f_{w+\nu}) - \hat{L}(f_w) \tag{5}$$

where $\hat{L}(f_w)$ is the empirical loss over the whole training set and generally very small, so we can drop it from the denominator.

3 Empirical study

In this section we discuss the experimental results using the proposed measures and their implications on different phenomena seen in practice regarding the generalization behavior.

3.1 Difference between true and random labeled data

It is possible to obtain zero training error on random labels using the same architecture for which training with real labels leads to good generalization. We would expect the networks learned using real labels (and which generalizes well) to have much lower complexity, under the suggested measure, than those learned using random labels (and which obviously do not generalize well). [7]

Figure 1 shows the results of [7]. We see that training the network on true labels (blue) only results in small increases of the norm-based capacity measures in comparison to the increases when training on random labels (magenta). As the network learns the true functional dependence between the input and output using true labels it only requires small increases in complexity as the subset size increases, whereas it requires more capacity for every newly seen data point when using random labels as it needs to memorize the data point.
We also see that the norm-based capacity measures are all bigger for random labeled data as the network has to memorize the data whereas the dependence between the input and output can be learned with lower capacity on true labeled data.

Looking at the sharpness as a capacity measure in Figure 1 we do not see a direct correlation between its value and the capacity of the network. We expect the model to become less sharp when increasing the network size which can be seen looking at the true labeled data points (blue) but clearly gets violated by the random labeled data points (magenta). Furthermore, we see a lower sharpness for random labels for the two smallest subset sizes which contradicts the intuition that the sharpness should be always lower for true labels.

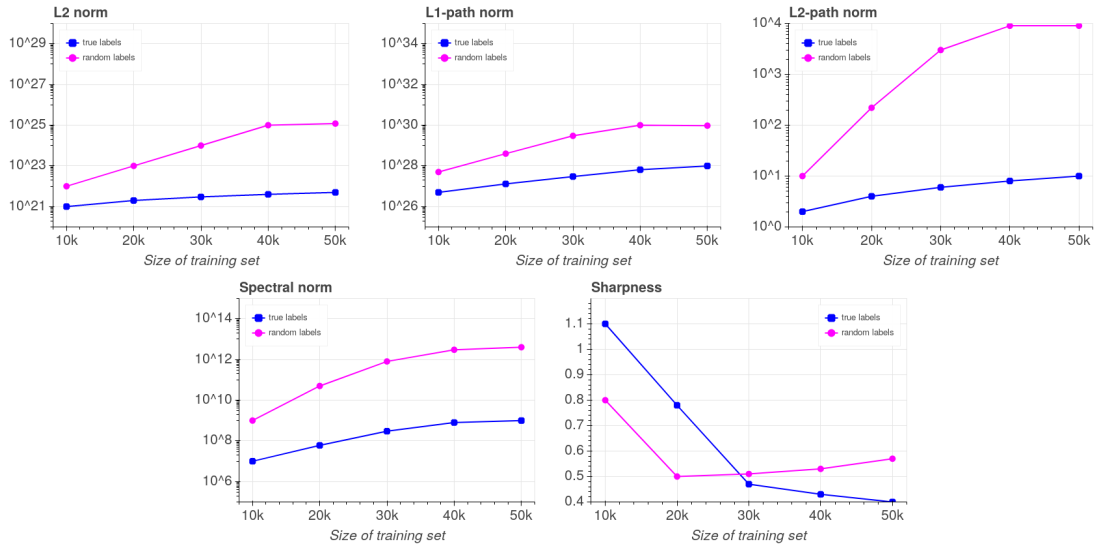

Figure 1: The proposed capacity measures calculated on the VGG model [10] after training on random (magenta) and true (blue) labeled subsets of CIFAR-10. We see only a small increase of the measures on true labels but a huge increase using random labels and the measures on random labels to be bigger for every subset. Reproduced from [7].

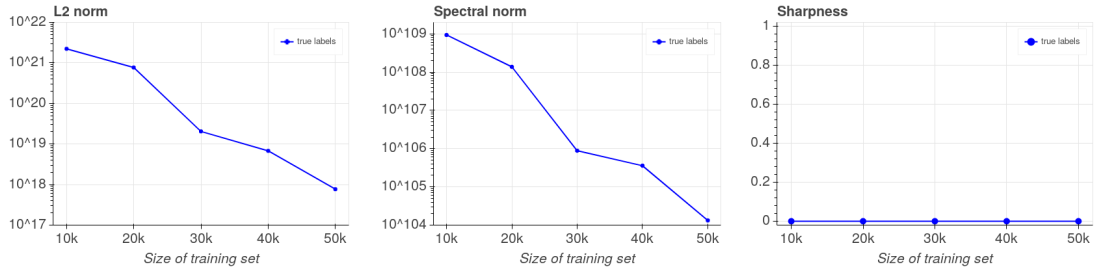

Figure 2: Our own results for the $l2$ and spectral norm as well as the sharpness calculated on the VGG-16 model after training on subsets of CIFAR-10. The results show no reproducibility of the results presented by [7].

3.1.1 Problems reproducing results

Following the vague instructions of [7] the VGG-16 network with batch normalization (Fig: 7) is trained on different subsets of the CIFAR-10 (Fig: 7) data set. For every subset a copy with the labels replaced by random labels has been created and the model is trained on both the true and random labeled subset.
The huge scope for interpretation of the approach used in [7] caused the non reproducibility of the results which are shown in Figure 2. The problems we face are:

Reproduction of network training Neyshabur et al. [7] only stated that their results were produced using VGG. As VGG is a class of CNN architectures with varying depths and optional batch normalization we performed a grid search over the implementations. Considering the capacity vs. training time trade-of we achieved the best results using VGG-16 with batch normalization.
Furthermore, [7] didn't provide information on the used solving strategy. Therefore, we tried simple stochastic gradient descent and Adam but stuck with the later as of the faster convergence.

Calculation of norms The calculation of path norms is trivial for the fully connected case but not for convolutional layers. As we need to enumerate every path through the network we need a way to associate the layer neurons of succeeding layers when applying a convolution. In the short

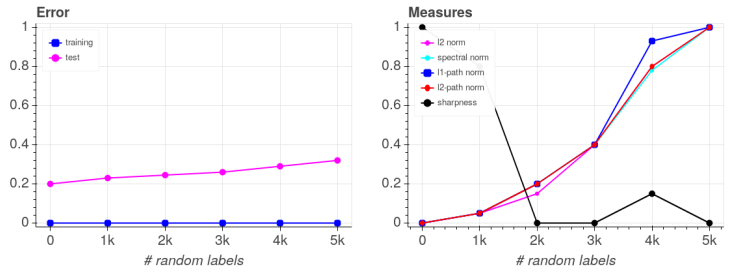

Figure 3: The test and training error on the subset of CIFAR-10 and varying confusion set sizes (left) and the corresponding proposed capacity measures (right). Reproduced from [7].

time of this paper we couldn't come up with a feasible implementation which made it impossible to reproduce the path norms.

Training on random labeled data Following the missing details of the network implementation and solving strategy a training on random labeled data was not possible. The network didn't overcome a 1/10 test accuracy on random labels which represents a random choice of the result by the network out of the 10 classes in the CIFAR-10 data set.

Training time The short scope of the seminar enclosing this paper combined with training times ranging from roughly five hours on the smallest subset ranging up to over one day for the biggest one resulted in only small search intervals for the grid search used on the hyper parameters as well for the algorithmic and architectural choices.

Sharpness Sharpness as the maximum over the adversarial perturbations is a maximization problem where the difference between the loss of the network with the added perturbation and the original network gets maximized. This maximization problem is constrained on the values of the perturbations and thus requires a constrained optimization strategy. We couldn't come up with such a strategy in the seminars time and thus tried a random search strategy. Due to the high duration of forward passes over the whole training set this random search didn't provide results in a feasible amount of time.

3.2 Difference between local minima

When training the same architecture, with the same training set, using two different optimization methods (or different algorithmic or parameter choices), one method results in better generalization even though both lead to zero training error. We would expect to see a correlation between the complexity measure and generalization ability among zero-training error models. [7]

To drive the network towards different parameter choices [7] proposed to train the VGG network [10](Fig: 7) on a subset of 10000 true labeled data points of the CIFAR-10 (Fig: 7) and to add confusion sets with varying sizes. A confusion set is a subset of the same data set where the labels are replaced with random labels. When optimizing the network over the union of the true and random labeled sets the resulting optimum has to be optimal for both of the sets.
We expect to see an increase of the test error with increasing confusion set size as the network not only has to learn the functional dependence between the input and output data but also has to memorize an increasing number of random labeled data points. This memorization prevents the network from generalizing well and can be seen in Figure 3.
Following the increase in test error and the resulting worse generalization behavior we expect the capacity measure to increase with the test error. In Figure 3 we see this behavior for all the norm-based capacity measures which hints towards them being a valid measure for generalization behavior. But again we see the sharpness to have no direct correlation to the test error which hints that it cannot predict generalization.

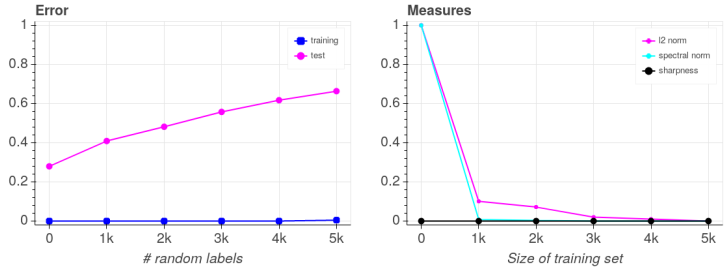

Figure 4: Our own results for the test and training error on the subset of CIFAR-10 and varying confusion set sizes (left) and the corresponding proposed capacity measures (right).

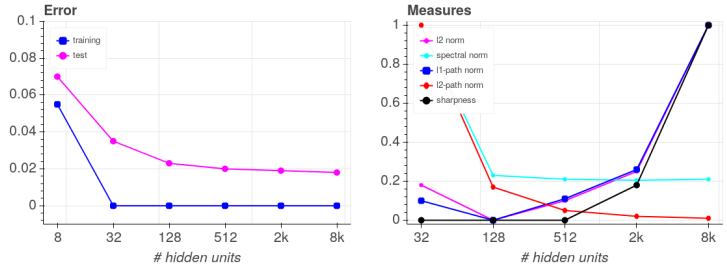

Figure 5: The test and training error on the MNIST handwritten digit data set and varying hidden unit sizes (left) and the corresponding proposed capacity measures (right). Reproduced from [7].

3.2.1 Problems reproducing results

We faced most of the problems already described in 3.1.1. Especially the missing model and solving strategy details 3.1.1 as well as the calculation of the norms 3.1.1 prevented us from reproducing meaningful results. Nonetheless, the results are displayed in Figure 4 where we can see an increase of the test error as expected when increasing the subset size. The calculated measures do not follow the expected behavior to increase with the worse generalization ability and sharpness wasn't calculable at all.

Training to zero training error not possible For the biggest confusion set size training to zero training error was not possible. We tried different optimization strategies and hyper parameters but couldn't drive the model to zero training error.

3.3 Implications of different hidden unit sizes

Increasing the number of hidden units, thereby increasing the number of parameters, can lead to a decrease in generalization error even when the training error does not decrease. We would expect to see the complexity measure decrease as we increase the number of hidden units. [7]

To investigate into this phenomenon Neyshabur et al. [7] trained a two layer perceptron (Fig: 8) on the MNIST handwritten digit data set (Fig: 8) using different numbers of hidden units. We see in 5 that the networks achieve zero training error for 32 and more hidden units and even though there is no decrease in the training error afterwards we see a decrease in the test error with every increase of the hidden units.

Looking at the capacity measures in Figure 5 we see that the $l2$-path and spectral norm follow the expected behavior and decrease with the increase of hidden units whereas the $l1$-path norm, the $l2$ norm and sharpness do not behave as proposed. Reproducing the results of [7] put out the values displayed in Figure 6 where we can see that we could achieve a decrease of the $l2$ and $l1$-path norm with the increase of hidden units but see the opposing behavior for the other norms and again no direct correlation between sharpness and generalization behavior.

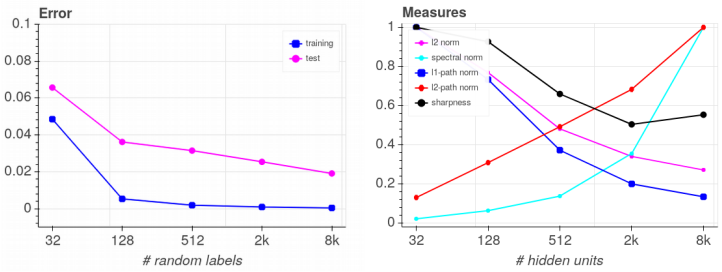

Figure 6: Our own results for the test and training error on the MNIST handwritten digit data set and varying hidden unit sizes (left) and the corresponding proposed capacity measures (right).

4 Conclusion

We saw that the norm-based capacity controls provide a good direction into understanding the empirical phenomena that correlate with the generalization ability of deep neural networks, but also that they alone are not sufficient to explain every phenomenon and further research is necessary. For sharpness, we have seen that it does not provide a valid capacity measure on its own and that it is not sufficient to ensure generalization. Neyshabur et al. [7] provide some further investigation of expected sharpness in the PAC-Bayesian framework and how they can be combined to provide a capacity measure.

5 Appendix

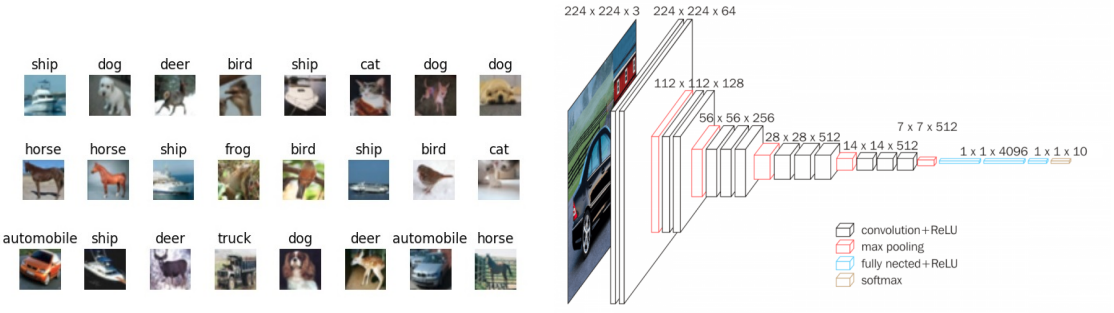

Figure 7: Left: A subset of the CIFAR-10 small image data set with ten classes. Right: The used implementation of the VGG-16 network with batch normalization. [12, 6]

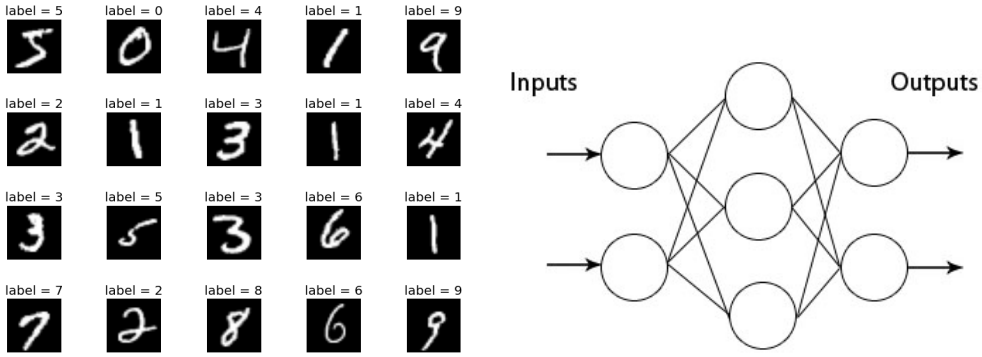

Figure 8: Left: A subset of the MNIST handwritten digit data set. Right: An exemplary two layer perceptron with one input layer, one hidden layer of hidden unit size three and one output layer. [11, 1]

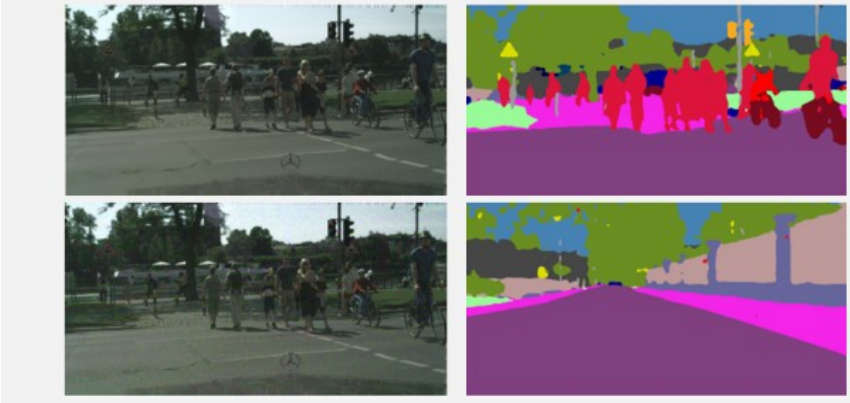

Figure 9: Upper row: The original image (left) and the correct segmentation mask (right). Lower row: The image with some additive adversarial perturbation barely visible to the human eye (left) and the corresponding wrong segmentation mask (right). [5]

References

[1] . http://neuroph.sourceforge.net/tutorials/MultiLayerPerceptron.html. [Online; accessed 12-10-2019].

[2] Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. In David Helmbold and Bob Williamson, editors, *Computational Learning Theory*, pages 224–240, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[3] Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. Generalization in deep learning, 2017.

[4] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *CoRR*, abs/1609.04836, 2016.

[5] Jan Hendrik Metzen, Mummadi Chaithanya Kumar, Thomas Brox, and Volker Fischer. Universal adversarial perturbations against semantic image segmentation, 2017.

[6] Neurohive. VGG16 - Convolutional Network for Classification and Detection. https://neurohive.io/en/popular-networks/vgg16/, 2018. [Online; accessed 12-10-2019].

[7] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 5947–5956, 2017.

[8] Behnam Neyshabur, Ruslan Salakhutdinov, and Nathan Srebro. Path-sgd: Path-normalized optimization in deep neural networks. *CoRR*, abs/1506.02617, 2015.

[9] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. *CoRR*, abs/1503.00036, 2015.

[10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[11] David Yang. Tutorial 1: MNIST, the Hello World of Deep Learning. https://mc.ai/tutorial-1-mnist-the-hello-world-of-deep-learning/, 2016. [Online; accessed 12-10-2019].

[12] David Yang. Tutorial 1: Cifar10 with Google Colab's free GPU - 92.5. https://mc.ai/tutorial-1-cifar10-with-google-colabs-free-gpu%E2%80%8A-%E2%80%8A92-5/, 2019. [Online; accessed 12-10-2019].

[13] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *CoRR*, abs/1611.03530, 2016.