Exploring Generalization in Deep Learning: An experimental study on complexity measures and generalization behaviour

Christina Aigner

Seminar: Optimization and Generalization in Deep Learning

Abstract

Deep neural networks have good generalization behavior in practice, but it is still unclear how we can measure and explain such generalization in greater detail. This study explores different complexity measures and their potential to explain generalization. For this, a comparative experimental study was conducted by reproducing and comparing the results from the paper "Exploring Generalization in Deep Learning", Neyshabur et. al. (2017). This report discusses the intuition behind the concepts introduced in the paper, as well as, similarities and differences between the original and the reproduced results. Further, the effectiveness and the limitation of each measure is analyzed with respect to generalization.

1 Introduction

Deep neural networks have become a main tool in solving complex nonlinear problems. Especially in the field of computer vision, they represent a considerable improvement over traditional machine learning methods. The over-parameterization of such networks, meaning the number of parameters being much higher than the number of training examples, has made it possible to train zero-training-error-models without over-fitting heavily. However, it has been unclear for a long time, whether this limits the generalization ability of such models. Recent studies suggest that the increase of the "generalization gap", when the model starts to overfit, does not arise for networks in an over-parametrized regime [7] [9]. This means, we can achieve better generalization with increasing architecture size (number of parameters). Still, there are a number of other factors which influence generalization. A great amount of research has been done in this area, but it remained an open question how to measure such generalization ability. Thus, a measure needs to be found, which, on the one hand, can express generalization and, on the other hand, can explain why an over-parameterized setting is favorable for generalization. When looking for a suitable measure, we first need to take look at the different parameters which can affect generalization. The authors in [8] focus on generalization with respect to (1) the training data, (2) the model architecture and (3) the optimization algorithm. Training data is an important factor because good data pre-processing is a prerequisite for generalization and a vital task for data-driven learning in general. If the data set contains a lot of noise, we won't be able to learn a good classifier. Also, the data set has to be class-balanced and well distributed, so that the training data distribution is close to the real problem. But most of all, the size of the training set matters. The more well-distributed data the model has seen, the better it can generalize. Second, the architecture of the network has an impact on generalization. A more complex architecture can learn more complex things and we could reason that more knowledge means better generalization. But this is not true for all problems, which we will see in section 4. To measure the effect of architecture size on generalization, we need to look at the capacity of the model. Capacity describes the intensity of the model weights and can be measured with weight norms. The authors in the paper suggest that a model with low complexity (lower architecture size) can be sufficient for learning when the capacity of the weights is high. Simply put, we can solve a problem by either taking a large network and train only a few epochs or taking a smaller network which is trained for longer. This relationship can be observed in figure 6 in the right plot, which visualizes the number of epochs required to train a fully connected network to zero training error. Both scenarios can be sufficient for generalization, nevertheless, measuring the capacity of the models and putting it in relation to the network size and training set size, could give valuable insights on generalization. Third, we want to find out weather different global minima on the training set can have different generalization behaviour

and investigate the role of implicit vs. explicit regularization. Explicit regularization in the loss function facilitates generalization by biasing the weights towards lower values and thus, being less likely to overfit the data. Neyshabur claims in [7] that the implicit regularization of the optimizer also significantly effects the generalization ability of a model and should not be underestimated.

This report is structured as follows: Section 2 describes the experiments of the paper and which questions should be answered with them. Section 3 gives a short theoretical introduction to the measures implemented in the paper and provides an intuition on how to interpret those measures. The results of my experiments are presented in section 4, where I will discuss the differences and similarities to the results in the paper. Section 5 provides a conclusion, as well as, limitations of the paper and an outlook on future research.

2 Experiments

The authors tested generalization behavior of deep learning models with respect to four different experiment parameters:

1. The increase of the training set size. The authors trained a VGGNet with 10 different training set sizes ranging from 1k up to 50k examples and observed the differences in generalization ability.

2. The training data. To analyse the effect of weight capacity and generalization ability, they trained the above setting once for true and once for random labels, as this yields the maximal difference in weight capacity (learning "by heart" vs. reusing previously learned knowledge). This led to another 10 models trained on different training set sizes with random labels.

3. The increase in model complexity. We trained model sizes of a fully connected two-layer perceptron with hidden units ranging from 23 up to 213 (= 11 different model sizes) to analyze the correlation between model complexity and generalization.

4. The effect of different global minima on the generalization behavior. For this, the authors trained a VGG-Net with a union subset of CIFAR10 with size 10k plus a varying size of random labels from 0 to 5K, which were 6 different models to test.

Set-ups 1., 2. and 3. were tested in my own experiments, set-up 4. will only be discussed in this report.

Furthermore, the paper explored 6 different measures for each of the above mentioned experiment setups: L2-Norm, L1-Path-Norm, L2-Path-Norm, Spectral-Norm, Sharpness and PAC Bayes. All norms, as well as, Sharpness were implemented in my experiments. PAC Bayes will only discussed in this report.

3 Theory

In this paper, various potential complexity measures are tested and analysed to which extent they can explain generalization. As this is an experimental seminar report, I won't go into the details of the theoretical foundation of those measures and rather explain their interpretation, behavior and intuition based on the results of the experiments.

3.1 Norms

Neyshabur et. al. (2017) explore four different norms as complexity measure. The l2 norm (equation 1), l1-path norm (equation 2) and l2-path norm (equation 3) are based on the results of Neyshabur's work on norms in [10] and Bartlett and Mendelson's theory on using Rademacher complexities as risk bounds [2]. The spectral norm (equation 4) used in this paper is based on the spectrally-normalized margin bounds of of Bartlett, Foster and Telgarsky [1]. With d being the number of hidden units, h the hidden unit and W being the weight matrix.

$$\text{l2 norm:} \frac{1}{\gamma_{margin}} \prod_{i=1}^{d} 4||W_i||_F^2 \tag{1}$$

$$\text{l1-path norm:} \frac{1}{\gamma_{margin}} | \sum_{j \in \prod_{k=0}^{d}[h_k]} | \prod_{i=1}^{d} 2W_i[j_i, j_{i-1}]||^2 \tag{2}$$

$$\text{l2-path norm:} \frac{1}{\gamma_{margin}} \sum_{j \in \prod_{k=0}^{d}[h_k]} \prod_{i=1}^{d} 4h_i W_i^2[j_i, j_{i-1}] \tag{3}$$

$$\text{spectral norm:} \frac{1}{\gamma_{margin}} \prod_{i=0}^{d} h_i ||W_i||_2^2 \tag{4}$$

From the equations we can see, that the norms also depend on a margin γ. This margin was introduced to make norms comparable for different kinds of classification problems. In neural networks, the scale of classification probability and weights can vary from problem to problem. This effects the value of the norm and would make it hard to use these norms as a generic complexity measure, as the results cannot be compared. Therefore, the results are normalized by a margin γ which captures the difference between the classification value of the true label minus the maximal value of all other labels 5. We then choose the a minimal γ, called γ_{margin} for which N examples have a γ lower than this γ_{margin}. In other words, γ_{margin} is the bound which holds true N examples in the training set. With this we make sure that the useful bound for our training set and does not only apply to a single outlier.

$$f_w(x)[y_{true}] - \max_{y \neq y_{true}} f_w(x)[y] \tag{5}$$

3.2 Sharpness

Sharpness ζ is a measure which describes the sensitivity of the training loss to a perturbation v which is added to the model weights and was introduced in [4]. If the perturbation makes us deviate strongly from the previously found optimization minimum, we can argue, that we generalize less well, whereas low sharpness indicates that we generalize better. All possible perturbations for model parameter i are drawn from a fixed bound where $|v_i| < \alpha(|w_i| + 1)$, Where α is a parameter fixed to 5e-4. For each of those possible v_i's we test for the resulting training loss. So the sharpness of the model is then, the maximal difference between the original loss and the perturbed loss (equation 6). This is a maximization problem, which the authors solved with the L-BFGS algorithm in a separate training environment.

$$\zeta_\alpha(w) = \frac{\max_{|v_i| < \alpha(|w_i|+1)} \hat{L}(f_{w+v}) - \hat{L}(f_w)}{1 + \hat{L}(f_w)} \simeq \max_{|v_i| < \alpha(|w_i|+1)} \hat{L}(f_{w+v}) - \hat{L}(f_w) \tag{6}$$

3.3 PAC Bayes

The PAC-Bayes framework [6] provides the basis for this complexity measure proposed in [5] and [3]. The PAC-Bayes measure has two important components: the expected sharpness and the KL divergence to the model prior (equation 7). Where m is the number of training examples and 1 δ the probability over the draw of the training data. The Expected sharpness is an approximation to the maximization problem in equation 6 as it randomly draws a weight perturbation from a zero-mean Gaussian distribution with the bound $|v_i| < \alpha(|w_i| + 1)$ as the standard deviation. So we would expect the values being slightly lower for the expected sharpness than the max sharpness, which might be the case for the observations in figure 5. The PAC-Bayes measure can be understood by fixing one parameter (here we fix the expected sharpness) and observe the change in the other (KL divergence). The higher the KL divergence for a fixed expected sharpness, the higher the capacity of the model. However, it is still unclear if this can be properly linked to generalization behavior, which will be discussed in section 4.

$$E_{v\sim\mathcal{N}(0,\sigma)^n}\hat{L}(f_{w+v}) \leqslant \hat{L}(f_w) + E_{v\sim\mathcal{N}(0,\sigma)^n}\hat{L}(f_{w+v}) - \hat{L}(f_w) + 4\sqrt{\frac{1}{m}(\frac{||w||_2^2}{2\sigma^2} + \ln\frac{2m}{\delta})} \quad (7)$$

4 Results

In this section I will discuss the experiments introduced in section 2 and will compare my results with the results in the paper [8]

4.1 Data capacity - Experiments 1 and 2

When training a VGG Network with training set sizes from 1k to 50k we have a classical over-parameterized example in deep learning, as VGG Net has about 140 million parameters. Figure 1 shows that the generalization error decreases with increasing the training set size, which is an expected behavior because adding more knowledge to the model leads to better generalization (except when the training data distribution is different from the real problem). When training random labels, the knowledge generated from additional data does not promote learning and so we expect the capacity to increase almost linearly with the number of examples (because every new sample has to be learned "by heart"). Furthermore, the learned weights cannot be meaningfully reused for predictions on the test data. This leads to a constantly high test error which is independent from the training set size and hence, bad generalization ability. In other words, we expect that training random labels leads to an increasing capacity in the weights, whereas the generalization ability stays low. In contrast, when training true labels, we expect a (slight) increase in both, capacity and generalization ability. So the measure we are looking for must be able to express those properties in order to be a valid generalization measure.

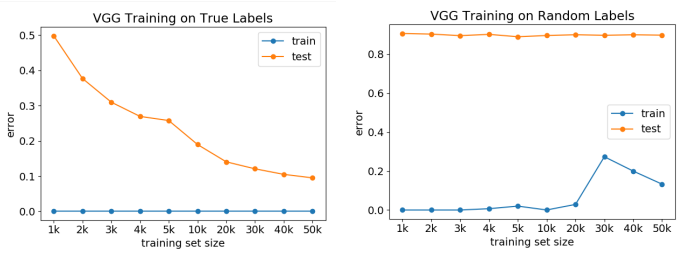

Figure 1: Training VGG on CIFAR10 Subsets

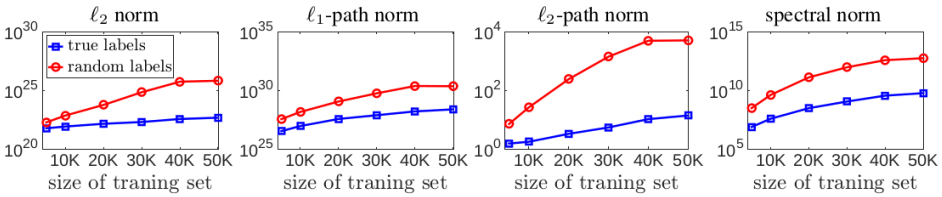

Figure 2: Norms for VGG trained on real and random labels - results from the paper

Figure 2 depicts the calculated norms for a VGGNet trained on true and random labels. We can see that the capacity (value of the norm) increases with increasing the size of the training set for true and random labels. This increase is due to learning of the network, the more data we see, the more expressive (or "smart") the weights become. This means weights trained with a larger training set absorb more knowledge and thus, have higher capacity. Important to notice here is, that the capacity for random labels is not only higher, but also increases with a higher rate than the capacity for true labels, as the model sees entirely new information with each new training

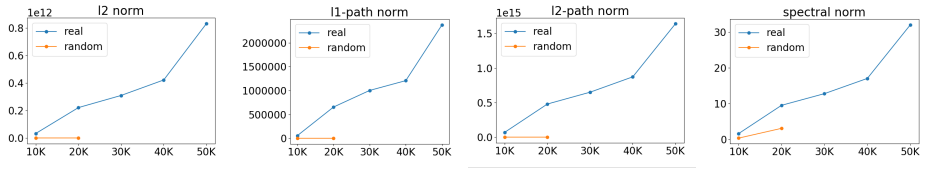

Figure 3: Norms for VGG trained on real and random labels - my results

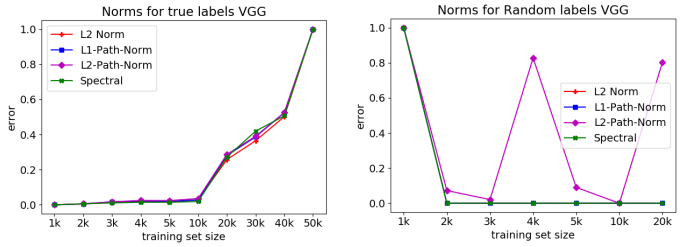

Figure 4: Norms for VGG trained on real and random labels (normalized values)

example and cannot reuse its previously learned knowledge efficiently. The l2 norm and l2-path norm most dominantly express this behavior by having the highest difference between the capacity of true and random labels on their own scale. Figure 3 shows the results of my calculation of the norms. For real labels (left plot), I was able to reproduce the results of the norms with a slight difference in value scales. Figure 4 displays all norms on a normalized scale from 0 to 1, to make the marginal increase better comparable. I added the results of sizes 1k to 5k to the plot because this shows that the norms only provide significant value increase for larger training set sizes, which the authors did not include in their plots. For larger sizes, however, we can see a constant increase as the training set size increases. Comparing the norms for true labels with the generalization error from figure 1 suggests that high capacity in an over-parameterized setting can be linked to good generalization ability. However, when we look at the results for random labels, it becomes clear that this norms are not a valid generalization measure. Capacity values for random labels (Figure 2) are higher than for real labels but generalize badly. Unfortunately, I was not able to reproduce the norms for models trained on random labels. First of all, I could only use training set sizes up to 20K for this comparison, as the other training set sizes did not converge zero training error, even after long training time (1500 epochs and roughly 10 hours for the 50k model). Second, Figure 4 and 3 show that even for models trained to zero training error on random labels, the norms did not provide any useful result. This result, however, cannot be due to a mistake in the implementation of the norms, because the norms for experiment 3 (see figure 7) are identical to the results in the paper. Therefore, I suspect that I pre-processed the data differently for random label training than the authors did. The paper did not provide any specific information on how they trained the random label models.

Sharpness-based measures (equation 6 and 7) can be interpreted as follows: Sharpness describes the how sensible the model reacts to perturbations in the weights, more specifically, how much the training loss differs from the original loss when perturbation is added. So if the perturbation makes us deviate strongly from the previously found optimization minimum, we can argue, that we generalize less well, whereas low sharpness indicates that we generalize better. For figure 5 this means, that the continuously decreasing sharpness for true labels can be connected to continuously increasing generalization (which would be a true statement for our experiment). I could reproduce this behavior for true labels by calculating the expected sharpness. For models trained with random labels the interpretation is not as straightforward. As we have a constant high test error in the training results, we know that random labels generalize badly. Sharpness for random labels shows a decreases for training set sizes up to 20k and increases for larger sizes, which would indicate an decreasing generalization error followed by an increase. As the paper does not provide the training results for VGG models, we do not know weather this is a valid statement for their setting. When comparing to my results, this is not a valid generalization measure.

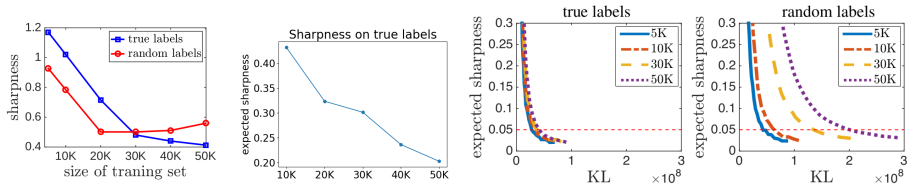

Figure 5: Maximized sharpness from the paper (left) and my results on the expected sharpness for VGG models trained on real labels (right).

4.2 Architectural capacity - Experiment 3

In this experiment we test how the introduced capacity measures behave in terms of varying network sizes. For this, I trained a fully connected network on the MNIST dataset. Figure 6 shows that I was possible to reproduce the trained models almost exactly. Like in the above setting with VGG, training the network until zero training error was a precondition for applying the margin-based bounds. So again, we can only use the model sizes 32 to 8k for the measurements (the authors observed the same issue and also analysed the same models subset). From figure 6 we can observe that networks with more hidden units generalize consistently better until the size of 256 hidden units. From that size onward, the generalization error continues to fall slightly or stays the same. In the paper, the authors state that the only possible explanation for this is the implicit regularization by the optimization algorithm, since the optimization was done without any explicit regularization. *This experiment also confirms the results of* [7] *as I did not observe an increase in generalization error for large networks.*

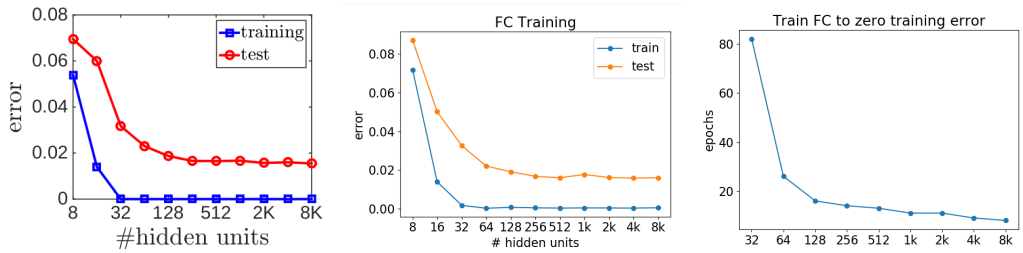

Figure 6: Training a fully connected feed-forward network network with varying number of hidden units on MNIST to zero training error. Results from the paper (left) and my results (right).

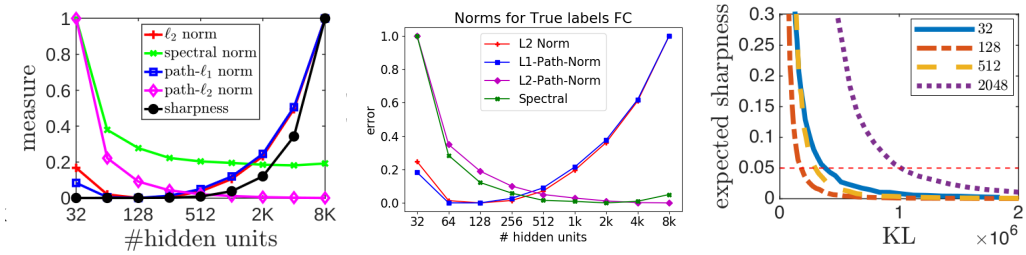

Figure 7: Complexity measures for a fully connected feed-forward network network with varying number of hidden units. Results for norms (normalized values) from the paper (left) and mine (middle). PAC-Bayes results on FC from the paper (right).

Figure 7 depicts the results of the different norms for this model subset normalized between 0 and 1 (like in the paper). I was able to reproduce the norms almost entirely, with the spectral norm being the only measure which showed slightly different values. Still, I could observe decreasing behavior for both, spectral and l2-path norm and an increasing behavior for l1-path norm and l2 norm. We see that the norms-based complexity measures decrease for networks up to 64-128

hidden units. For all larger networks, we observe that the l2 norm and l1-path norm increase with the size of the network. This is by no means in correlation with the marginal increase in generalization. The l2-path norm and the spectral norm have an opposite behavior as they depend on the number of hidden units, which also means, that they cannot be used as an independent generalization measure as the result will always be biased by the architecture. Also PAC-Bayes fails to explain the generalization behavior as we can see that the values of the KL divergence for fixed expected sharpness of 0.05 does not increase relative to the generalization error in figure 6 (the KL divergence for 32 hidden units is higher than for 128 and 512 hidden units and lower than for 2048 hidden units. But the generalization error of 32 hidden units is the higher than for all the others). From this we can reason that norms, margin and sharpness based complexity measures do not provide explanation for generalization, especially in an over-parametrized regime. Neyshabur et. al. further investigate this issue in their follow-up paper "Towards Understanding the Role of Over-Parameterization in Generalization of Neural Networks", where they introduce novel complexity measures based on unit-wise capacities. The proposed capacity bounds correlate with the behavior of test error with increasing network sizes and thus, a measure of great potential as this could explain the improvement in generalization with over-parametrization.

5 Conclusion

The results show, that a combination of expected sharpness and norms do seem to capture much of the generalization behavior, especially in terms of varying training set sizes. However, the measures largely fail to explain generalization when changing the size of the architecture. Summing up, the proposed measures in the paper provide valuable insights on generalization but are not a valid generic generalization measure that captures generalization with respect to training data, architecture and optimization. A topic, which is still left unresolved, is how the choice of the optimization algorithm biases such complexity to be low and the relationship between optimization and implicit regularization. This issue is addressed by follow-up papers by the author. Also, the practical usability of such capacity measures is rather limited, as margin-based measures can only be used for zero training error models, which usually is not the case in practice.

References

[1] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6240–6249, 2017.

[2] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[3] Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 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. *arXiv preprint arXiv:1609.04836*, 2016.

[5] David McAllester. Simplified pac-bayesian margin bounds. In *Learning theory and Kernel machines*, pages 203–215. Springer, 2003.

[6] David A McAllester. Some pac-bayesian theorems. *Machine Learning*, 37(3):355–363, 1999.

[7] Behnam Neyshabur. Implicit regularization in deep learning. *arXiv preprint arXiv:1709.01953*, 2017.

[8] 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.

[9] Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076*, 2018.

[10] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401, 2015.