Small Data, Big Decisions: Model Selection in the Small-Data Regime

Jörg Bornschein [1] **Francesco Visin** [1] **Simon Osindero** [1]

Abstract

Highly overparametrized neural networks can display curiously strong generalization performance – a phenomenon that has recently garnered a wealth of theoretical and empirical research in order to better understand it. In contrast to most previous work, which typically considers the performance as a function of the model size, in this paper we empirically study the generalization performance as the size of the training set varies over multiple orders of magnitude. These systematic experiments lead to some interesting and potentially very useful observations; perhaps most notably that training on smaller subsets of the data can lead to more reliable model selection decisions whilst simultaneously enjoying smaller computational overheads. Our experiments furthermore allow us to estimate Minimum Description Lengths for common datasets given modern neural network architectures, thereby paving the way for principled model selection taking into account Occams-razor.

1. Introduction

According to classical statistical learning theory, achieving an optimal generalisation loss requires selecting a model capacity that strikes the best balance between underfitting and overfitting, i.e., between not having enough capacity to model the training data accurately and having too much, and thus prone to adapt too closely to the training data at the expense of generalisation. Under this theory, the final generalisation loss plotted against model capacity should behave as a U-shaped curve – initially decreasing as the capacity increases (underfitting) to reach a minimum (optimal model size) and then finally increase again (overfitting).

Contrary to these results, it has long been observed that neural networks show a curiously good generalization performance (in terms of error) when applied to classification problems, even though the generalization cross-entropy exhibits all the characteristics of overfitting. There has recently been renewed interest in studying this phenomenon, both theoretically and empirically (Advani & Saxe, 2017; Spigler et al., 2018). Belkin et al. (2019) for example argue that beside the classical underfitting and overfitting regimes, a third one for massively overparameterised models exists. The transition into this regime is called the *interpolation threshold* and is characterized by a peak in generalization error, followed by a phase of further decrease. The peculiar shape of the generalization-error over model-size curve lends the term *double-descent* to this phenomenon. Work by Nakkiran et al. (2019) has sharpened this picture and shown that this behaviour can be observed when training modern neural network architectures on established, challenging datasets. Most of the work studying generalization of neural networks focuses on the relationship between generalization performance and model size. Instead we present an empirical study that investigates the generalization performance as a function of the training set size.

In the rest of the Introduction we outline the main contributions of our work.

1.1. Performance analysis

One of our key contributions is to gather performance curves as a function of the training set size for a wide range of architectures on ImageNet, CIFAR10, MNIST and EMNIST. We also perform an extensive sweep over a wide range of architectures, model sizes, optimizers and, crucially, we vary the size of the training sets over multiple orders of magnitude, starting from the full dataset down to only few examples per class. Our study covers even extreme cases – for example the training of oversized ResNet architectures with 10 or less examples per class. This is, to the best of our knowledge, the most extensive empirical analysis conducted on generalization for massively overparameterized models, and strengthens the emerging understanding of training regimes for modern deep learning.

1.2. A ranking-hypothesis

One salient observation we have made has not been described in the literature: overparameterized model archi-

---

[1] DeepMind, London, United Kingdom. Correspondence to: Jörg Bornschein <bornschein@google.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

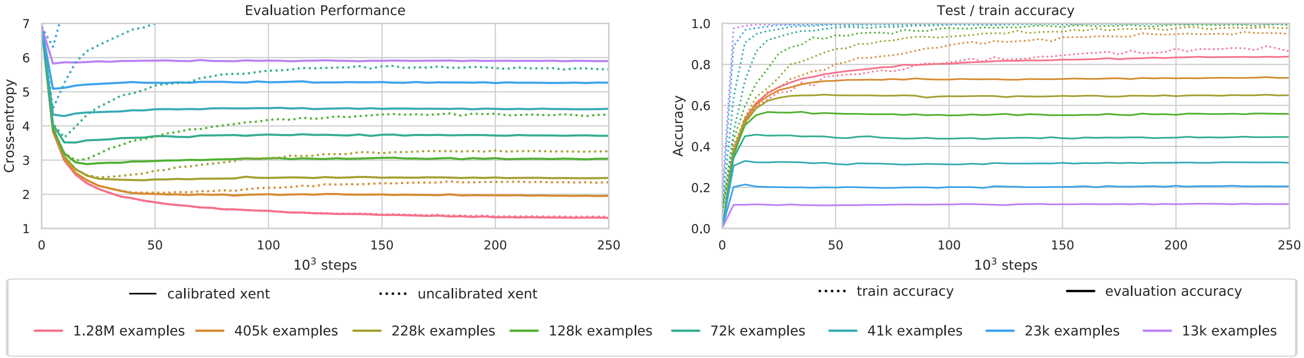

Figure 1. Learning curves for ResNet-101 on subsets of the ImageNet dataset using a RMSProp with a cosine learning-rate schedule. Left: the uncalibrated generalization cross-entropy shows a strong overfitting signature (dashed line), the calibrated cross-entropy does not.

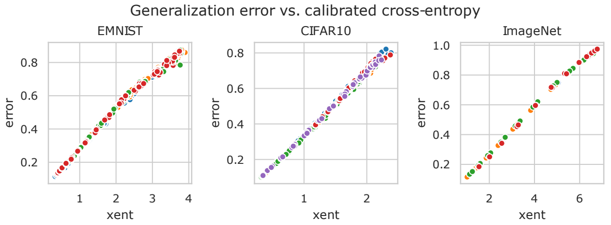

Figure 2. Post-convergence generalization error vs. generalization cross-entropy without early-stopping for a range of model architectures and training set sizes. The colours represent the different model architectures. We observe that the calibrated cross entropy is strongly correlated with the generalization error rate.

tectures seem to maintain their relative ranking in terms of generalization performance, when trained on arbitrarily small subsets of the training set.

This observation prompts us to hypothesize that: when (i) two sufficiently large neural network architectures A and B are trained with a well tuned optimizer on datasets of size N; and (ii) we observe that, in expectation, A performs better than B; then (iii) A will also perform better than B in expectation for all differently sized datasets drawn from the same underlying distribution, as long as we remain well beyond the interpolation threshold.

Unfortunately this is only an hypothesis, but if this conjecture is true it would have profound practical implications. Namely, it would mean that, for sufficiently large models, it would be possible to perform model selection or architecture-search using small subsets of the data, and expect that the decision regarding which model to prefer remains valid when applied to much larger datasets. Indeed, our experiments show that training on small or medium sized subsets of the training data leads in many cases not only to faster convergence, but also to a more robust signal for model selection than training on big datasets, and is therefore often preferable.

1.3. Calibration & minimum description length

Independently of the model-selection hypothesis, we also show that it is possible to avoid some negative effects of overfitting by simply choosing an optimal softmax-temperature on a small held-out dataset; i.e., by calibrating the neural network (Guo et al., 2017). After calibration, the generalization cross-entropy becomes a stable and well-behaved quantity even when model sizes and training set sizes vary considerably. Being able to compute well-behaved generalization cross-entropy on small training sets allow us to compute reliable Minimum Description Length estimates, a quantity that is of interest for principled model selection taking into account Occam's-razor. We will discuss this more in depth in Section 5.

2. Related Work

The literature on generalization performance for learned predictors is vast and spans seminal work on the classical bias-variance-tradeoff (Geman et al., 1992; Domingos, 2000) all the way to theoretical and empirical work investigating the still poorly understood, but often strong generalization performance of neural networks. The latter has been approached from a number of different directions: For example pointing out that neural networks seem to perform implicit capacity control (Zhang et al., 2016), investigating the learning dynamics and properties of the loss landscape (Spigler et al., 2018) and interpreting stochastic gradient descent as an approximation of probabilistic inference (Mandt et al., 2017). Finally, two lines of work recently contributed to the understanding of generalization, the first studying infinitely wide neural networks (Jacot et al., 2018; Allen-Zhu et al., 2019) and the second focusing on the double-decent phenomenon (Belkin et al., 2019; Nakkiran et al., 2019).

The vast majority of these previous studies focus on the model-size dependency of the generalization performance. Notable exceptions are the work by Nakkiran et al. (2019), who investigates the neighbourhood of the interpolation threshold, and the work by Hestness et al. (2017), who first evaluated the generalization error-rate for language models and ResNets as a function of the training set size. They point out that the generalization error-rate can be well predicted by assuming a power law and interpolating from smaller training sets to bigger ones. In a similar fashion, our work shows that the relative model ranking trained on small datasets is maintained when trained on bigger ones, a result of practical importance for model selection and architecture search.

3. Methods

3.1. Temperature calibration

Overparameterized models are typically trained by minimizing either a regression loss or a categorical cross-entropy loss, but in contrast are then evaluated on their generalization performance measured on the error-rate. This is because the generalization cross-entropy can overfit severely for models that are sufficiently big.

We show here that it is possible to avoid the negative effects of overfitting when using the categorical cross-entropy by simply choosing an optimal softmax-temperature on a small heldout dataset; i.e., by calibrating the neural network in the way proposed by Guo et al. (2017). In practice we implement temperature calibration by performing gradient descent on the calibrated cross-entropy loss w.r.t. a single scalar temperature parameter. We interleave the regular model training steps with steps of gradient descent on the calibration loss on some held-out data, that we refer to as the *calibration dataset*. This allows us to track the generalization cross-entropy online during learning.

This calibration procedure can prevent the generalization cross-entropy from overfitting, even when the model size is order of magnitudes larger than the training set size, and without relying on early-stopping (see Figure 1 and Figure 2). Note that, just as with the post-convergence calibration proposed by Guo et al. (2017), the calibration procedure does not modify the other parameters, directly or indirectly.

Being able to compute well-behaved generalization cross-entropies is desirable because it is the loss we optimize for and because it allows us to compute MDL estimates, as explained in Section 5. That being said, we could express all other results in this paper in terms of error-rates instead of cross-entropies, and the observations and conclusions would still hold (see Supplementary material).

3.2. Datasets and models

Throughout this study we present a large number of experiments on subsets of different sizes of popular benchmarking datasets. We call the total set of datapoints available for a particular training run the *available dataset*, which is split into a *training set* and a *calibration set*. The former is used to optimize the connection weights, the batch-norm parameters and all other parameters that are considered part of the neural networks, while the latter to optimise the calibration temperature, as explained in Section 3.1, as well as to determine the optimal learning-rate and, potentially, to perform early-stopping. If not mentioned otherwise, we will use a 90%/10% training/calibration split of the *available dataset*. To assess a model's ability to generalise beyond the available-dataset, we then successively evaluate them on a separate held-out or *evaluation dataset*. We experimented with balanced subset sampling, i.e. ensuring that all subsets always contain an equal number of examples per class. But we did not observe any reliable improvements from doing so and therefore reverted to a simple i.i.d sampling strategy. We always pay particular attention to not to inadvertently leak data, i.e. use datapoints that have not been properly accounted for to select the hyperparameters.

We conduct experiments on the following datasets:
MNIST consists of 60k training and 10k test examples from 10 classes (LeCun, 1998). We train MLPs of various depth and width, with and without dropout, as well as standard ConvNets on this dataset. Unless otherwise noted, we use ReLU as the nonlinearity.
EMNIST provides 112,800 training and 18,800 test datapoints from 47 classes in its balanced subset (Cohen et al., 2017). We train the same family of model architectures we also train on MNIST.
CIFAR10 consists of 50k training and 10k test examples from 10 classes (Krizhevsky et al., 2009). We train a wide range of models on this dataset, including simple architectures like MLPs and ConvNets; architectures that have been carefully optimized for image classification: ResNet-20 (He et al., 2016) and Wide ResNets (Zagoruyko & Komodakis, 2016) as well as a selection of architectures from the NASBench-101 (Ying et al., 2019) search space. The latter were chosen by disregarding the worst 10% and then picking 5 equally-spaced from the remaining ones. The Supplementary material contains the description of these architectures. The rationale is that we want to evaluate a range on non-optimal architectures and confirm their relative ranking is preserved when using smaller datasets too.
ImageNet contains 1.28M training and 50k validation examples from 1000 classes (Russakovsky et al., 2015). We train a selection of widely known standard models like VGG-16 (Simonyan & Zisserman, 2014), ResNets (He et al., 2016) and Inception (Szegedy et al., 2016). Additionally we consider S3TA (Zoran et al., 2019), a sequential and attention based model that takes multiple glimpses at various spacial locations in the image before emitting a prediction. It is composed of a ResNet-style feature extractor with a reduced

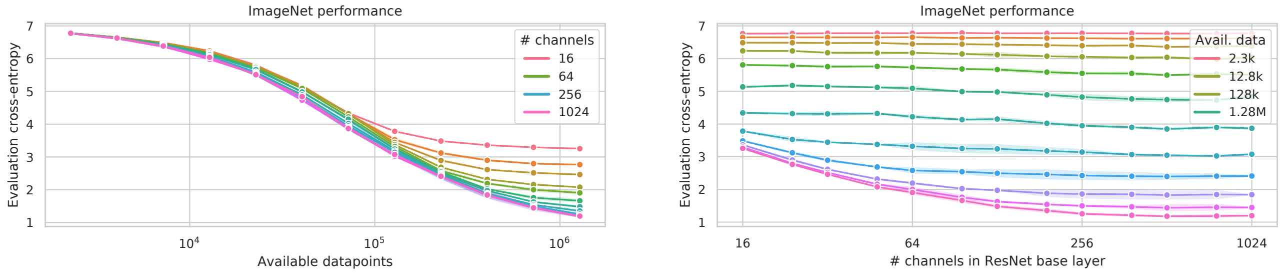

Figure 3. Cross-entropy performance profiles for the ResNet-101 architecture on ImageNet when trained with RMSProp. We use 90% of the available data for training, 10% for calibration and report the generalization performance on the unseen validation set. Note that even when training with as little as ≈ 2.3 images per class, there is no harm in using a ResNet model with 4 × more channels (16× more parameters) than the standard ResNet.

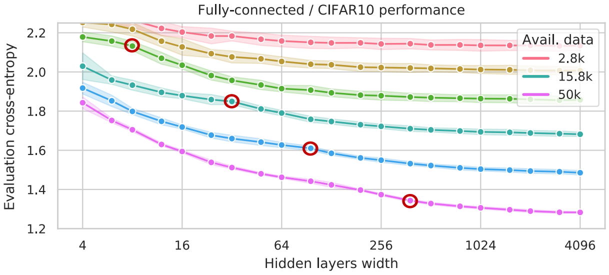

Figure 4. Performance profiles for a fully connected MLP with 3 hidden layers on CIFAR10 as a function of the hidden layer size. Red points mark the smallest models that approach a close to zero training error-rate.

number of strides, to keep the spatial resolution higher, and an LSTM equipped with a spatial attention mechanism to ingest features. The rationale for including it in our exploration is that it supposedly has a different inductive bias than pure conventional models, which could increase the chance of falsifying our consistent ranking hypothesis.

3.3. Training

To ensure our observations are not specific to a particular optimization method, we run experiments with different variants of gradient-descent. For each experiment we sweep over a fixed set of possible learning rates and pick the best one according to the calibration loss, independently for each model architecture and training set size.

Throughout this study we use the following optimizers:
Adam (Kingma & Ba, 2014) with fixed learning rates $\{10^{-4}, 3 \cdot 10^{-4}, 10^{-3}\}$ and 50 epochs.
Momentum SGD with initial learning rates $\{10^{-4}, 3 \cdot 10^{-4}, \cdots, 10^{-1}\}$ cosine-decaying over 50 epochs down to 0 (0.9 momentum and $\epsilon = 10^{-4}$).
RMSProp + cosine schedule (Tieleman & Hinton, 2012) with initial learning rates of $\{0.03, 0.1, 0.3\}$ and cosine-decaying to 0 over 50 epochs. We choose the same hyperparameters used by (Ying et al., 2019) for their NASBench-101 experiments, with the exception of the number of epochs, which we reduced from 108 to 50 (momentum=0.9, $\epsilon = 1$).

For all experiments we use a batch size of 256 examples. The term epoch always refers to the number of gradient steps required to go through the full-sized dataset once; i.e., on ImageNet an epoch is always $1.28M/256 = 5000$ gradient steps, regardless of the size of the actual training set used.

We evaluated many more combinations than those presented in this paper, a selection of which is contained in the Supplementary material. Throughout all these control runs our qualitative results were confirmed, which suggests that these results do not just emerge from the interaction of specific models and optimizers; or from a particularly sensible hyperparameter choice.

4. Experiments

4.1. Properties of calibration

Figure 1 visualizes typical learning curves when training a ResNet-101 model with cosine-decayed RMSProp on subsets of ImageNet. Depending on the training set size the model can memorise the training set and reach a zero training error-rate within a few thousand gradient steps. The uncalibrated cross-entropy on held-out data shows severe symptoms of overfitting in these cases but, as reported before, the error computed on the same set tends to instead flatline and not to degrade significantly.

This suggests that the models are not unfavourably adjusting their decision boundaries as training progresses into the overfitting regime, but are first and foremostly becoming unduly confident in their predictions. When using calibration we rectify this over-confidence and obtain stable cross-entropies that show the same kind of robust behaviour as the evaluation error-rate.

Early-stopping can still have a positive effect though: We regularly observe a small degradation in generalization performance just around the point where the rapid decrease in

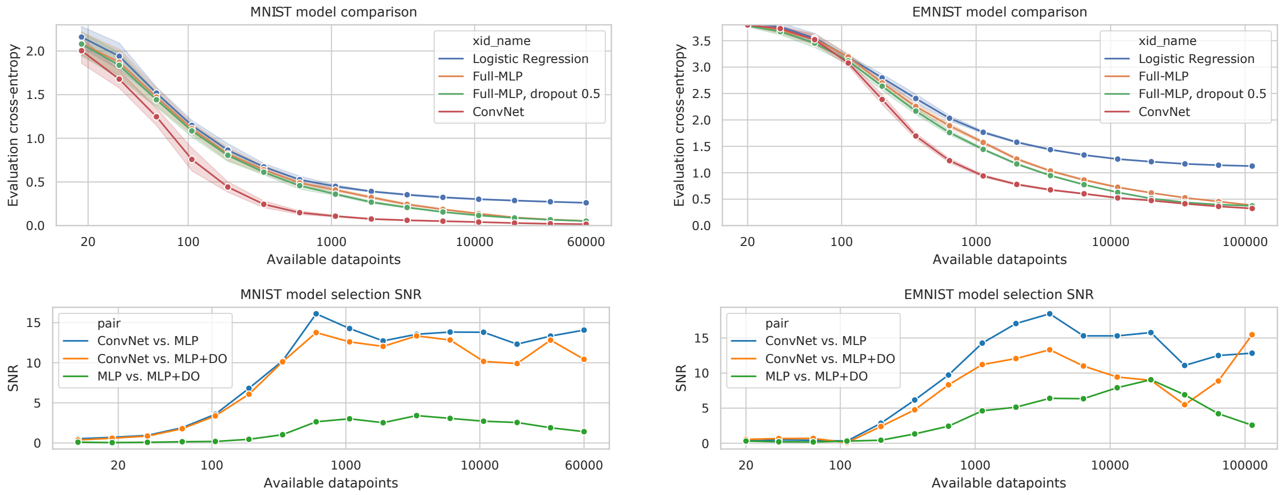

Figure 5. **Top row**: Generalization performance for various model architectures trained with Adam on MNIST and EMNIST as a function of training set size. Uncertainty bands represent standard-deviation after training with 30 different seeds. **Bottom row**: SNR for the performance difference between models; estimated using 1000 bootstrap samples.

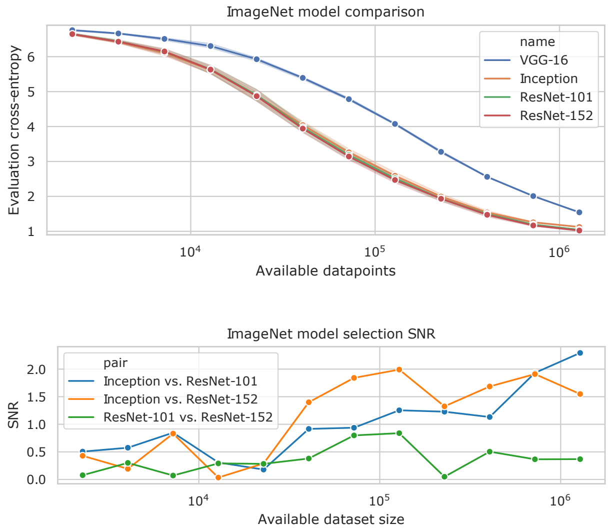

Figure 6. **Top**: Generalization performance for models trained on ImageNet; Uncertainty bands from training 5 different seeds. **Bottom**: SNR for the performance difference between models; estimated using 1000 bootstrap samples.

training error tapers off into a slow decrease towards zero. We can usually minimize this effect by making the models bigger and tuning the learning rate more accurately to reach optimal performance. These observations are not specific to this experiment, but generalize to all optimizers and models considered in the paper.

Figure 2 depicts the very strong correlation between the generalization error and calibrated crossentropy for EMNIST models trained with Momentum-SGD, CIFAR10 models with Adam and ImageNet models with annealed RMSProp.

4.2. Interaction of model size and training set size

We designed a set of experiments to directy connect to previous work and to confirm that the recent insight regarding model size and generalization performance holds in the small data regime. Just like previous work we vary the model size by proportionally scaling the number of channels in all the convolutional layers or, for MLPs, the width of all the fully connected layers throughout the models. In contrast to previous work, here we additionally sweep over the size of available data used for training. As expected, it is evident from Figure 3b) and Figure 4 that scaling up models does generally improve the generalization performance, but comes with a price in terms of memory and computational effort, and is subject to diminishing marginal improvements. Selecting a model-size for a problem therefore requires finding the best trade-off between computational resources and the missed generalization performance one is willing to accept. An interesting observation from these experiments is that the model size to achieve close-to-optimal performance seems to be independent of the size of the training set. For ResNet-101 on ImageNet for example (Figure 3 b), it takes $\approx$ 384 channels in the first ResNet block to achieve close to optimal performance when training on either the 10k or the 1.2M samples training sets. Similarly on CIFAR10 (Figure 4), between 1024 and 2048 units in the hidden layers achieves close-to optimal performance on all training sizes. The Supplementary material contains additional plots suggesting the same constant relationship over a number of experiments.

According to the double-decent perspective (Belkin et al., 2019) we would expect the generalization error (and potentially the generalization cross-entropy) over model-size to

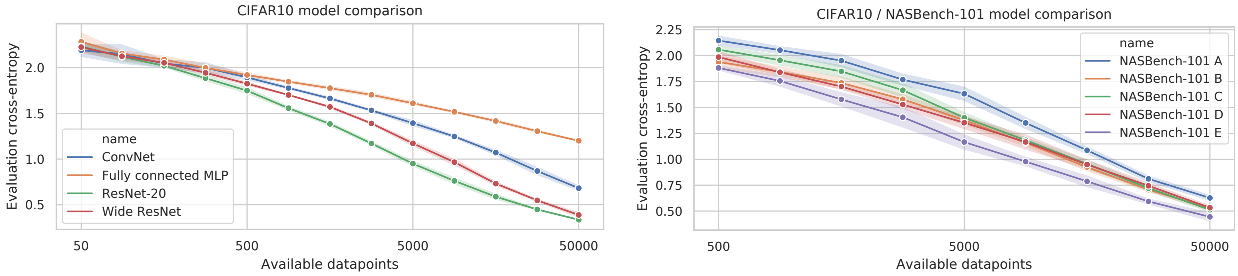

Figure 7. **Left**: Selected models trained on CIFAR10; uncertainty bands from 30 different seeds. The ConvNet has 4 hidden layers with 3x3 kernels and 256 channels; every second layer uses stride 2 width a single 2048 unit wide fully connected hidden layer on top. **Right**: Five models equidistantly selected from the NASBench-101 architecture search space; 50% of the available datapoints were used for training, 50% for calibration.

spike around the point where a model barely reaches approximate zero training error. We never quite see a spike, but we do observe recognizable artifacts in the curves; most pronounced on CIFAR10 when using a fully connected MLP (see Figure 4). For all other models and datasets this effect was much less pronounced and usually not recognizable.

4.3. Consistent Model Ranking

Having established that we get reasonable and consistent results when training big neural networks on tiny datasets down to only few examples per class, and that it is neither necessary nor desirable to downscale the model size when the training set is small, we can now focus on comparing different model architectures on a given dataset. For all these experiments we choose sufficiently big models: models that are beyond their interpolation threshold for the full datasets. More precisely, we consider a model "big enough" if doubling its size (in terms of number of parameters) does not result in a noticeable generalization improvement.

Figure 5 (top-row) shows the results for MNIST and EMNIST. For both datasets we compare fully connected MLPs with 3 hidden layers, 2048 units each, either with or without dropout and a simple convolutional network with 4 layers, 5x5 spatial kernel, stride 1 and 256 channels. For comparison we also added logistic regression, even though we do not consider it part of our hypothesis because it can not be scaled to be a universal approximater to approach the irreducible Bayes-error.

Overall, we tried many different variations and architectures: Replacing ReLUs with tanh non-linearities, adding batch-norm or layer-norm, changing the number of hidden layers or convolution parameters. Many of these changes have an effect on the performance over dataset-size curves, but we could not find a pair of architectures with performance curves that crossed outside of their uncertainty bands, i.e., whose relative ranking was not maintained across all the datasets sizes we trained on. We therefore suspect that, in expectation, whenever a model A achieves better generalization performance than an architecture B given a certain sized training set, it will also have a better performance for all other training set sizes.

Figure 6 (top-row) shows the corresponding results for ImageNet, with uncertainty estimates computed over 5 seeds. All models here are slightly oversized versions of their literature counterparts, for example the ResNets use 384 channels in their lowest ResNet block instead of 256 like the ones described by He et al. (2016).

Not only the standard image recognition models maintain their relative ranking as the training set size changes, but even the S3TA model follows the same pattern. Indeed we notice that the generalization performance-curve is dominated by the ResNet-style feature extractor, i.e, that training a S3TA model with a ResNet-50 style feature extractor moves the curve close to the standard ResNet-50 performance curve. In the interest of clarity, to better appreciate the difference between the curves, we plot them in a separate figure without uncertainty estimates (see Supplementary material).

Figure 7 (top-row) shows the results for models on CIFAR10, with uncertainty bands obtained from training 30 different seeds. Even though the uncertainty is higher and the models are generally closer together, we still observe that the model architectures maintain their relative ranking, within their respective uncertainty bands. For all these models the generalization error seems to be far away from converging to the underlying (irreducible) Bayes error even when training on the full dataset.

4.4. Uncertainty of model selection

If the ranking is preserved across dataset sizes we should be able to perform model selection using training sets of arbitrary size and assume that our decision remains optimal for larger datasets created from the same underlying distribution.

Visually inspecting Figure 5, 6 or 7 reveals indeed some of the features we expect in this case: In the limit of very small datasets, all models start on average with a uniform prediction, achieving a generalization loss of $\approx \log(\#\text{classes})$ for balanced datasets. In the limit of very large datasets, and assuming sufficiently large models and effective training procedures, they all approach a common performance: presumably the irreducible Bayes error of the dataset. For intermediate sized training sets though, different models exhibit vastly more varied generalization performance. In fact, we see that the performance gap between any two models is visibly larger for some intermediate sized training sets than for the full sized datasets.

While this is encouraging, optimization of neural networks is a stochastic process, with noise from the selection of training and calibration examples, from parameter initialization, the mini-batching process and potentially others sources. To ensure a reliable model selection process one can not consider the absolute performance gap alone, but should also take into account its uncertainty. To quantify the reliability of the model selection signal, i.e. the performance gap, we estimate the signal to noise ratio: $SNR = \sqrt{\frac{\Delta^2}{Var[\Delta]}}$ with $\Delta = \mathcal{L}_A - \mathcal{L}_B$, where $\mathcal{L}_X$ is the final generalization cross-entropy after training model X. The bottom rows of Figures 5 and 6 show the SNR estimates computed using 1000 bootstrap samples.

4.5. Computational savings from smaller datasets

We used fixed learning rates, or fixed learning rate schedules, for all our experiments without any early stopping. This simplified the experimental setup and reduced the number of additional hyperparameters we needed to tune independently per dataset size, which could have skewed the results if not done fairly. But that also means that, by choice, we did not enjoy any computational savings from using smaller datasets, even though Figure 1 suggests that it takes fewer training steps to converge on smaller datasets and thus that such savings should be possible.

To get a lower bound on the potential computational savings, we implemented a simple automatic annealing scheme for ImageNet: Every $1.28M/256 = 5000$ training steps we performed a full-batch evaluation of the calibrated cross-entropy on the calibration set. We lowered the learning rate by a factor of 10 when the performance did not improve for three consecutive iterations. We terminated training when the learning rate reached $^1/_{1000}$-th of it's initial value. Using this schedule for ResNet models, we use 128k training examples (which provides a good SNR, as suggested by Figure 6) and observe that training terminates after 4-7x fewer gradient steps than training on the full 1.28M examples.

5. MDL and Bayes Factor Estimation

We now show that we can use calibrated generalization cross-entropy estimates to perform model selection following *Occam's Razor*, the principle according to which we should prefer simpler models if possible.

Within the field of deep learning model selection is typically done with a simple cross validation strategy: Split the available data into a training and a validation set and choose the model that performs best on the validation subset. This approach works well when a lot of data is available and when overfitting is not a major concern.

Other approaches to model selection like for example Bayesian model selection or the Minimum Description Length (MDL) principle have been studied in detail and can often be understood in a context of a general theory of inductive inference. For these it is known that their definition gives raise to a notion of the models *complexity*. And we know that they prefer to select simpler models (according to their respective complexity measure) if they have similar predictive performance.

Unfortunately, neither of these approaches are naively applicable to deep neural networks: To perform Bayesian model selection between two model architectures $\mathcal{H}_1$ and $\mathcal{H}_2$ for example, we need to compute $\frac{p(\mathcal{H}_1|\mathcal{D})}{p(\mathcal{H}_2|\mathcal{D})} = \frac{p(\mathcal{D}|\mathcal{H}_1)p(\mathcal{H}_1)}{p(\mathcal{D}|\mathcal{H}_2)p(\mathcal{H}_2)}$, where $p(\mathcal{D}|\mathcal{H}_i) = \int_\Theta p(\mathcal{D}|\Theta)p(\Theta)d\Theta$, which is in most cases intractable. Indeed, estimating this quantity for deep neural networks proved to to be a very challenging problem, and usually requires changes to the parameterization and training procedure, often leading to models with significantly worse predictive performance (Mac Kay, 2003).

Minimum Description Length (MDL) (Rissanen, 1978; 1989; Grünwald, 2007) provides a closely related alternative principle that also considers $p(\mathcal{D}|\mathcal{H}_i)$ the quantity of interest, but does not force us to consider the prior $p(\Theta)$ and posterior $p(\Theta|\mathcal{D})$ explicitly. Instead, we are only concerned with the shortest possible compression of the dataset $\mathcal{D}$: The length of a message that transmits the content of the dataset (the labels) uniquely from a sender to a receiver assuming some model and coding-procedure $\mathcal{H}$. We will refer to the length of the message as $\text{DL}(\mathcal{D}|\mathcal{H})$. By virtue of the Kraft-McMillan theorem, we know that any such procedure corresponds to a probability distribution over all possible datasets: since $\sum_\mathcal{D} \exp(-\text{DL}(\mathcal{D}|\mathcal{H})) \leq 1$, we can identify $p(\mathcal{D}|\mathcal{H}) \propto \exp(-\text{DL}(\mathcal{D}|\mathcal{H}))$.

Blier & Ollivier (2018) described an approach to constructing a *prequential* (Dawid, 1984; Grünwald, 2007) code that works well together with deep neural networks: The basic idea is to transmit one datapoint at a time; always coding it with a code derived from a model that was trained on the previously transmitted datapoints. When the sender and receiver use a deterministic training method (i.e. fixed seeds

$\mathrm{DL}(\mathcal{D}\|\mathcal{H}_1) - \mathrm{DL}(\mathcal{D}\|\mathcal{H}_2)$	LR	MLP	MLP+do
Log. regression			
MLP	11982 ± 142		
MLP+dropout	12237 ± 154	254 ± 60	
ConvNet	15293 ± 126	3310 ± 64	3057 ± 87

Table 1. MDL log-evidence estimates for various model pairs on MNIST in nats. For example, the evidence in favour of an MLP with dropout vs an MLP without is $\exp(254) \approx 10^{110}$ to 1.

$\mathrm{DL}(\mathcal{D}\|\mathcal{H}_1) - \mathrm{DL}(\mathcal{D}\|\mathcal{H}_2)$	LR	MLP	MLP+do
MLP	11982 ± 142		
MLP+dropout	12237 ± 154	254 ± 60	
ConvNet	15293 ± 126	3310 ± 64	3057 ± 87

Table 2. MDL log-evidence estimates for selected model pairs on CIFAR10 in nats.

for all randomness; well-defined stopping criteria etc.) both sides learn exactly the same model and therefore can construct a common code from its predictions. The total length of the message to transmit a dataset $\mathcal{D} = \{y_i\}_{i=1}^N$ of size N is therefore $\mathrm{DL}(\mathcal{D}|\mathcal{H}) \leq \sum_{i=1}^N \log p(y_i|\{y_j\}_{j<i}, \mathcal{H})$. Blier & Ollivier (2018) showed that they obtain significantly shorter description lengths than with other approaches to estimate them for deep learning models. However, they did not have a principled way to regularize the models to be trainable on small datasets. Instead they proposed to switch between different model architectures with different capacities; therefore not obtaining description lengths for a single model, but rather for the entire procedure consisting of multiple, increasingly more elaborate architectures and their switching pattern.

Using calibration we can instead construct a coding method for a single, fixed model and training procedure: Both sender and receiver simply derive a training and calibration split from the previously transmitted data and use the calibrated cross-entropy prediction to construct the code for the next datapoint. The remainder of the prequential approach remains unchanged.

In practice this means that the description length of a dataset for a given $\mathcal{H}$ can be approximated by the *area under the curve* of plots like Figure 5, 6 and 7, if we just construct the *available* and *evaluation* sets in a specific way: Instead of randomly sampling them from the underlying distribution, we have to ensure that the available dataset for a subset of size $N > M$ is a superset of the datapoints used when training the model for the subset of size M; and the evaluation examples have to be exactly those examples that were added.

A description length based model selection approach thus considers the area between two plots in Figures 5 and 6 and Figure 7 (top-row) because it corresponds to the log-evidence in favour of the model with the smaller area under the curve: $\log \mathrm{DL}(\mathcal{D}|\mathcal{H}_1) - \log \mathrm{DL}(\mathcal{D}|\mathcal{H}_2)$.

$\mathrm{DL}(\mathcal{D}\|\mathcal{H}_1) - \mathrm{DL}(\mathcal{D}\|\mathcal{H}_2)$	VGG16	RN-50	RN-101
ResNet-50	$1.14M$		
ResNet-101	$1.23M$	$87k$	
ResNet-152	$1.27M$	$125k$	$37k$

Table 3. MDL log-evidence estimates for selected model pairs on ImageNet in nats. The uncertainty for all models is less then 20k nats for all pairings.

We implement the suggested splitting procedure and estimate the area between the performance curves using a simple trapezoid integral approximation for various model pairs. We obtain uncertainty estimates on this measure by running 3 different seeds and by varying the number points for the interval estimate (Tables 1 to 3).

6. Conclusions

In this paper we present a comprehensive empirical study on how overparameterized neural networks generalize as a function of the training set size. Our results confirm and extend the recently growing body of literature that analyzes and seeks to understand the curiously good generalization performance of big neural networks; and takes an in-depth analysis of the dependency between the training set size and the generalization performance.

From our experimental result we derive the hypothesis that sufficiently big neural networks, those that operate far beyond their interpolation threshold, maintain their expected relative generalization performance ranking irrespective of the size of the training set they are trained on. While we are not aware of a theoretical argument that could support this conjecture, we provide a strong empirical verification of this claim conducted on several architectures, model sizes, datasets and optimization methods. The results of our experiments have a wide range of practical implications for model selection and architecture search, even more so because our experiments show that training on smaller subsets of the data can not only save computational resources, but also improve the model selection signal at the same time. Both these aspects are of particular importance for neural architecture search approaches and can find immediate practical application.

Finally, in this work we also show that simple temperature calibration is sufficient to obtain reliable and well behaved generalization performance in terms of cross-entropy. Taking a prequential perspective, this allows us to compute Minimum Description Length estimates for deep-learning models, a quantity that was previously inaccessible, and thereby paving the way for a principled model selection procedure that takes into account Occam's-razor.

Acknowledgements

We would like to thank Andriy Mnih, Markus Wulfmeier, Sam Smith and Yee Whye Teh for helpful discussions and their insightful comments while working on this paper.

References

Advani, M. S. and Saxe, A. M. High-dimensional dynamics of generalization error in neural networks. *arXiv preprint arXiv:1710.03667*, 2017.

Allen-Zhu, Z., Li, Y., and Liang, Y. Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in neural information processing systems*, pp. 6155–6166, 2019.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Blier, L. and Ollivier, Y. The description length of deep learning models. In *Advances in Neural Information Processing Systems*, pp. 2216–2226, 2018.

Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

Dawid, A. P. Present position and potential developments: Some personal views statistical theory the prequential approach. *Journal of the Royal Statistical Society: Series A (General)*, 147(2):278–290, 1984.

Domingos, P. A unified bias-variance decomposition. In *Proceedings of 17th International Conference on Machine Learning*, pp. 231–238, 2000.

Geman, S., Bienenstock, E., and Doursat, R. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.

Grünwald, P. D. *The Minimum Description Length Principle*. The MIT Press, 2007. ISBN 0262072815.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1321–1330. JMLR. org, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hestness, J., Narang, S., Ardalani, N., Diamos, G., Jun, H., Kianinejad, H., Patwary, M., Ali, M., Yang, Y., and Zhou, Y. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 8571–8580. Curran Associates, Inc., 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

LeCun, Y. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Mac Kay, D. J. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

Mandt, S., Hoffman, M. D., and Blei, D. M. Stochastic gradient descent as approximate bayesian inference. *The Journal of Machine Learning Research*, 18(1):4873–4907, 2017.

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. Deep double descent: Where bigger models and more data hurt. *arXiv preprint arXiv:1912.02292*, 2019.

Rissanen, J. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

Rissanen, J. *Stochastic complexity in statistical inquiry*. World Scientific, 1989.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Spigler, S., Geiger, M., d'Ascoli, S., Sagun, L., Biroli, G., and Wyart, M. A jamming transition from under-to over-parametrization affects loss landscape and generalization. *arXiv preprint arXiv:1810.09665*, 2018.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Tieleman, T. and Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4 (2):26–31, 2012.

Ying, C., Klein, A., Real, E., Christiansen, E., Murphy, K., and Hutter, F. Nas-bench-101: Towards reproducible neural architecture search. *arXiv preprint arXiv:1902.09635*, 2019.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Zoran, D., Chrzanowski, M., Huang, P.-S., Gowal, S., Mott, A., and Kohl, P. Towards robust image classification using sequential attention models. *arXiv preprint arXiv:1912.02184*, 2019.