Large Neural Networks Learning from Scratch with Very Few Data and without Explicit Regularization

Christoph Linse
Thomas Martinetz

{linse,martinetz}@inb.uni-luebeck.de
Institute for Neuro- and Bioinformatics, University of Lübeck
Lübeck, SH, Germany

ABSTRACT
Recent findings have shown that highly over-parameterized Neural Networks generalize without pretraining or explicit regularization. It is achieved with zero training error, i.e., complete over-fitting by memorizing the training data. This is surprising, since it is completely against traditional machine learning wisdom. In our empirical study we fortify these findings in the domain of fine-grained image classification. We show that very large Convolutional Neural Networks with millions of weights do learn with only a handful of training samples and without image augmentation, explicit regularization or pretraining. We train the architectures ResNet018, ResNet101 and VGG19 on subsets of the difficult benchmark datasets Caltech101, CUB_200_2011, FGVC_Aircraft, Flowers102 and StanfordCars with 100 classes and more, perform a comprehensive comparative study and draw implications for the practical application of CNNs. Finally, we show that a randomly initialized VGG19 with 140 million weights learns to distinguish airplanes and motorbikes with up to 95% accuracy using only 20 training samples per class.

CCS CONCEPTS
• Theory of computation → Sample complexity and generalization bounds; • Computing methodologies → Neural networks.

KEYWORDS
convolutional neural networks, fine-grained image classification, generalization, image recognition, over-parameterized, small data sets

1 INTRODUCTION
Large Deep Convolutional Neural Networks (CNNs) show excellent generalization capabilities in a variety of tasks, particularly in image recognition. Often, their performance can be boosted by increasing the depth or the width of the architectures, which is bound to an increase in trainable parameters. The good generalization capabilities are often attributed to a suitable bias within modern CNN architectures and effective regularization strategies like weight decay or dropout.

However, generalization is also present in highly over-parameterized regimes and without explicit regularization, where even random label assignments or random image data can be memorized [16]. Belkin called it the "modern" interpolation regime" [3]. The generalization capabilities of large neural networks do not seem to be harmed by the millions of parameters in today’s popular architectures. Due to their large, inherent capacity the training error is zero after training by Empirical Risk Minimization (ERM), i.e. the training data is memorized, but, nevertheless, the test error can be unexpectedly low, as shown in [16]. This is in contrast to standard machine learning wisdom, where one would not expect any learning in these highly over-parameterized regimes. The data is highly over-fitted. Statistical learning theory based on uniform convergence requires a high probability that there is no bad solution with a large deviation between its empirical and its true risk, i.e. the training and the generalization error. This can only be expected if the number of training samples is (much) larger than the underlying VC-dimension of the classifier. However, this is not the case in over-parameterized regimes. In over-parameterized regimes, bad solutions with zero training but large generalization error do exist. Interestingly, it seems that they hardly occur in general.

In this paper, we empirically study the generalization of large, highly over-parameterized networks with a focus on fine-grained image classification. Usually, fine-grained classification is tackled using various techniques to boost the generalization abilities, such as image augmentation, regularization, novel loss functions, semi-supervised learning, or pretrained networks. In this paper, we want to study the extreme case of very large networks with 10-140 million weights and just a few training samples of 10-67 per class, with no augmentation, no pretraining, and no explicit regularization. We train models with random weight initialization and significantly more classes than samples per class and choose complex, fine-grained classification problems. The generalization is measured for the popular, large models ResNet018, ResNet101, and VGG19 with increasing numbers of weights. We provide an extensive range of experiments and train hundreds of models on subsets.
of the Caltech101, CUB_200_2011, FGVCAircraft, Flowers102, and StanfordCars datasets.

This work is not related to transfer learning, which harnesses the well-known power of pretrained networks to boost generalization capabilities [1]. Such models are pretrained on large-scale datasets like ImageNet [4] and are subsequently fine-tuned on a smaller dataset to solve a specific problem. However, during pretraining the total number of train images is similar to the number of trainable parameters, which is far from the highly over-parameterized regime that we study in this paper. Here, we use completely randomly initialized networks.

The paper is structured as follows. Section 2 provides all details about our experimental setup including the fine-grained classification datasets and the architectures. The results are presented and discussed in Section 3. The paper completes with our conclusions in Section 4.

2 MATERIALS AND METHODS

2.1 Datasets

The fine-grained classification datasets are summarized in Table 1, where the number of classes, the size of the datasets, and statistics about the number of images per class are presented. Figure 1 provides example images from every dataset showing 2 images for 4 classes, respectively.

Caltech101. [8] was published in 2004. It contains 101 categories and a background class, which we ignore in our experiments. The categories offer a wide degree of diversity including objects like laptops, smartphones and motorbikes, but also animal species and plants. 8677 images were collected manually from the internet and scaled to have a width of about 300 pixels. Sometimes, the objects are shown in a cluttered environment and sometimes they are seen in front of a white background.

Caltech2. is a two-class subset of Caltech101 with 798 images showing motorbikes and 798 images showing airplanes.

CUB_200_2011. is the Caltech-UCSD Birds-200-2011 dataset [13] from 2011, which extends the Caltech-UCSD Birds 200 dataset [14]. It contains a total of 5994 train and 5794 test images with a mean width of about 470 pixels. There are 200 different bird categories and each category is represented with about 30 training images. Variations in the color of the plumage, lighting, perspectives, and poses introduce a large intra-class variance while the differences between bird categories can be subtle, which makes this data set particularly challenging.

FGVCAircraft. or the Fine-Grained Visual Classification of Aircraft dataset [9] was published in 2013 and contains 6667 training and 3333 test images showing 100 different airplane models. The images typically have wide aspect ratios and a width of about 1100 pixels. The data is split into an official training and test set with about 67 and 33 images per class, respectively. Airplanes are rigid objects and therefore do not pose the challenge of deformation. However, there is a notable degree of intra-class variance in the visual appearance due to advertisement, airlines and perspective.

Flowers102. was published in 2008 as the 102 Category Flower dataset [10]. On average the images are 630 pixels wide and show one or several blossom instances. In order to be consistent with the other datasets, which have training and test sets only, we train with the combination of the training and validation set of Flowers102, leading to 20 training images for each flower type and 2040 images in total. For testing, there are 6149 images. While flowers can have small inter-class variations, visual properties like scale, pose, light, and also flower subtypes (e.g. color variants) can lead to strong intra-class variation. Therefore, Flowers102 is considered to be a challenging dataset.

StanfordCars. [7] from Stanford University was presented in 2013 and is composed of 196 different car models. The 8144 training and 8041 test images have an average width of 700 pixels. Typically, only one car is shown at once. The imaging conditions and the locations of the cars are not restricted to a single environment. Sometimes, a car is on a road, other images have been taken indoors and with a stylized background. Also, the perspective is not constrained, such that the image recognition system has to learn features from the front, side, and rear of a car model.

Table 1: Detailed structure of the benchmark datasets.

| Dataset        | Mode  | Classes | Number of images in/of | Number of images in/of | avg. per class |
|----------------|-------|---------|------------------------|------------------------|---------------|
|                |       |         | dataset                | smallest class         | largest class |
| Caltech101     | -     | 101     | 8677                   | 31                     | 800           | 85.91         |
| Caltech2       | -     | 2       | 1306                   | 788                    | 798           | 798.66        |
| CUB_200_2011   | train | 200     | 5794                   | 29                     | 30            | 29.97         |
|                | test  |         |                        |                        |               |               |
| FGVCAircraft   | train | 100     | 6667                   | 66                     | 67            | 66.67         |
|                | test  |         |                        |                        |               |               |
| Flowers102     | train | 102     | 2400                   | 20                     | 20            | 20.00         |
|                | test  |         |                        |                        |               |               |
| StanfordCars   | train | 196     | 8041                   | 28                     | 48            | 41.03         |

Figure 1: Fine-grained image classification datasets. Two examples of four different classes are shown, respectively. For Caltech2 with its two classes, four examples per class are depicted.
Table 2: Details on the architectures. The memory consumption during the forward pass was measured using a batch size of 32 and an image size of 3 × 224 × 224 pixels.

| Model     | Parameters(10⁶) | Multi-Adds(10⁹) | Size(MB) | GPU Memory Forward Pass(MB) |
|-----------|-----------------|-----------------|----------|----------------------------|
| ResNet018 | 11.2            | 1.8             | 43.9     | 1825                       |
| ResNet101 | 42.7            | 7.8             | 167.5    | 5119                       |
| VGG19     | 140.0           | 19.6            | 546.9    | 6009                       |

2.2 Network Architectures

ResNet018 [6] is employed as a baseline model. It is moderately sized with 11.2 million weights, but still in the over-parameterized regime with regard to the small-scale datasets in the study. Then, we use ResNet101 as a much deeper model with 101 layers and almost four times as many weights as ResNet018. Third, we use the VGG19 [12] architecture with a total of 140.0 million weights. A trained VGG19 model needs more than half a gigabyte when saved on the disk. The VGG19 consists of a set of 16 convolutional layers and 3 fully connected ones. The majority of the weights are located at the fully connected part of the network. More detailed information about the architectures is provided in Table 2, where the number of parameters, the number of numerical operations, the model size, and the memory consumption on the GPU when performing a forward pass are summarized.

2.3 Training Set-up

The network weights are initialized using Kaiming initialization [5]. Training minimizes the cross-entropy loss for 300 epochs. At the end of the training procedure, the classification error on the training set is always zero. The Lamb optimizer [15] is used to get a stable training process with smooth learning curves. The automatic differentiation framework of Pytorch is used [11]. The initial learning rate of 0.0015 is decreased until 1.5 × 10⁻⁶ in a step-wise manner. The batch size is 32. No image augmentation is performed. Also, there is no explicit regularization like weight decay or dropout. In our data pre-processing pipeline the input images are reshaped to have 224 × 224 pixels. In a normalization step, the mean RGB value of the entire dataset is subtracted from each image and we divide it by the standard deviation. 5 models are trained in each setting to enable a statistical validation of their performance metrics. Depending on the dataset size, it takes between 30 minutes and a few hours to train a single model on a single NVIDIA GTX 2080 Ti graphics card.

When dealing with very small datasets (between 40 and 8144 train samples) statistical fluctuations have to be taken into account. We address this issue by presenting results on 5 different datasets to enable the comparison between different data distributions and to draw meaningful conclusions. For every single experiment, we train and test 5 networks with different seeds. Please note, that we also choose different data subsets for the 5 networks when training on the Caltech dataset.

3 RESULTS AND DISCUSSION

The models are trained on different subsets with an increasing number of training samples per class. Table 3 and Table 4 show the obtained accuracies on the test sets. For the Caltech101 dataset, we have a separate table, since there is no official split in the training and test set and we experimented with different splits.

Table 3: Test accuracy of different classifiers on the benchmark datasets with various numbers of training samples per class. For each configuration, 5 models were trained with different seeds and initializations. The results show the mean test performance and its standard deviation.

| Dataset | Classes | Model     | Samples p-class | Accuracy | Std |
|---------|---------|-----------|-----------------|----------|-----|
| ResNet018 | 20 | ResNet018 | 15.64 | 0.00 |
| ResNet101 | 20 | ResNet101 | 22.45 | 0.58 |
| VGG19 | 20 | VGG19 | 16.99 | 0.80 |
| ResNet018 | 30 (all) | ResNet018 | 22.66 | 0.75 |
| ResNet101 | 30 (all) | ResNet101 | 29.96 | 1.03 |
| VGG19 | 30 (all) | VGG19 | 24.99 | 0.97 |
| SVM (HOG + RGB) [2] | 30 (all) | SVM (HOG + RGB) | 5.00 | - |
| CUB_200_2011 | 200 | ResNet018 | 16.03 | 1.09 |
| ResNet101 | 20 | ResNet101 | 25.01 | 0.46 |
| VGG19 | 20 | VGG19 | 36.63 | 0.85 |
| ResNet018 | 67 (all) | ResNet018 | 49.88 | 0.59 |
| ResNet101 | 67 (all) | ResNet101 | 58.91 | 0.24 |
| VGG19 | 67 (all) | VGG19 | 68.78 | 0.26 |
| SVM, Spatial Pyramid [9] | 67 (all) | SVM, Spatial Pyramid | 48.69 | - |
| Flowers102 | 102 | ResNet018 | 37.21 | 0.82 |
| ResNet101 | 10 | ResNet101 | 40.45 | 2.07 |
| VGG19 | 10 | VGG19 | 36.99 | 1.06 |
| ResNet018 | 20 (all) | ResNet018 | 54.66 | 1.52 |
| ResNet101 | 20 (all) | ResNet101 | 58.87 | 3.40 |
| VGG19 | 20 (all) | VGG19 | 53.05 | 0.76 |
| SVM (HSV) [10] | 10 | SVM (HSV) | 43.0 | - |
| StanfordCars | 196 | ResNet018 | 20.33 | 0.66 |
| ResNet101 | 20 (all) | ResNet101 | 24.34 | 1.27 |
| VGG19 | 20 (all) | VGG19 | 24.6 | 4.75 |
| SVM, Spatial Pyramid [7] | 20 (all) | SVM, Spatial Pyramid | 69.5 | - |

Table 4: Performance of ResNet018, ResNet101 and VGG19 on the Caltech101 dataset for different splits into training and test set. The results show the mean performance and standard deviation of 5 trained models. DAGSVM shows the mean recognition rates per class.

| Model     | Train samples | Test samples | Accuracy | Std |
|-----------|---------------|--------------|----------|-----|
| ResNet018 | 10 per class  | 20 per class  | 27.24    | 1.12 |
| ResNet101 | 10 per class  | 20 per class  | 32.79    | 1.28 |
| VGG19     | 10 per class  | 20 per class  | 39.41    | 0.57 |
| ResNet018 | 20 per class  | 10 per class  | 42.30    | 0.98 |
| ResNet101 | 20 per class  | 10 per class  | 43.96    | 1.73 |
| VGG19     | 10 per class  | 20 per class  | 54.38    | 1.01 |
| ResNet018 | 80.00% (25-640 p.c.l.) | 20.00% (6-160 p.c.l.) | 65.90    | 0.52 |
| ResNet101 | 80.00% (25-640 p.c.l.) | 20.00% (6-160 p.c.l.) | 65.27    | 1.53 |
| VGG19     | 80.00% (25-640 p.c.l.) | 20.00% (6-160 p.c.l.) | 71.11    | 0.83 |
| ResNet018 | rest (21-790 p.c.l.) | 10 per class  | 66.08    | 1.24 |
| ResNet101 | rest (21-790 p.c.l.) | 10 per class  | 66.32    | 1.87 |
| VGG19     | rest (21-790 p.c.l.) | 10 per class  | 71.52    | 0.59 |
| DAGSVM [17] | 15 per class  | rest (17-785 p.c.l.) | 59.05    | 0.56 |
| DAGSVM [17] | 30 per class  | rest (1-770 p.c.l.) | 66.23    | 0.48 |

In Figure 2 the results are illustrated more clearly. All networks with their 10-140 million parameters do learn with a handful of training data per class. Before training, their accuracy is 0.01-0.005 (random guessing), and after training the accuracy increased by a factor of 10-70. It is astonishing, that a classifier with 140 million parameters (VGG19) learns to discriminate 100 different aircraft.
models with only 67 example images per class with up to 70% accuracy, which is an increase by a factor of 70. It is correct in more than 2 of 3 cases in a task which is definitely difficult also for humans. This is in great contrast to what traditional machine learning wisdom would have expected in these highly over-parameterized regimes. The Flowers102 dataset provides similar results. With only 20 example images per class, the classifier is able to learn to discriminate 102 flowers up to an accuracy of more than 50%. Instead of being correct in every 100th case (before training), after training it is correct in every second case. The increase in accuracy is by a factor of 50. For the 200 birds, ResNet101 achieves the best results with an increase in accuracy by a factor of 60 by training with 30 images per class. The variance is always below 5%. For these large networks, there exist bad zero training error solutions, even solutions with zero test accuracy. However, these bad solutions did not occur. The accuracies for different seeds hardly vary.

On the datasets we studied, the most successful model is ResNet101, which is also the deepest one in this study. It is the best generalizing model in 6 of 8 cases. VGG19, the model with the highest number of weights, is the best generalizing model in 2 of 8 cases. The model with the fewest parameters, ResNet018, has the worst results on the fine-grained image classification tasks, even though it should generalize better when trained on small datasets according to traditional machine learning wisdom.

Of course, these results are not state-of-the-art but can be further improved by standard tricks like data augmentation, regularization, or pretrained weights. Instead, we demonstrate that learning takes place easily far outside the orbits of traditional machine learning wisdom, often just as good or even better than it was state-of-the-art a few years ago within the orbits of traditional thinking. For this purpose, we have added some state-of-the-art results from then in the Tables 3 and 4, usually achieved with Support-Vector Machines on elaborated handcrafted features.

Another interesting observation in Figure 2 is that the accuracy curves of different architectures plotted against the number of training samples per class have no intersections. This means that the performance ranking of the architectures is consistent over the number of training samples. For CUB_200_2011, Flowers102, and StanfordCars the best generalizing model is ResNet101, disregarding the size of the training set. For FGVCAircraft, the VGG19 net consistently provides the best results. Thus, the experiments suggest that the selection of the best model depends on the dataset and not on the number of training images. In other words, the data distribution chooses the architecture, not the size of the training dataset. Of course, it is not clear to what extent this relationship can be generalized.

3.1 Further Experiments on the Caltech101 Dataset

In order to substantiate the findings further experiments are performed on the Caltech101 dataset. Table 4 presents the accuracy of ResNet018, ResNet101, and VGG19 using different splits of the Caltech101 dataset, which lacks an official split. Interestingly, the network with the highest number of parameters, VGG19, has always the highest generalization ability with a maximum of 71.52% (59%) test accuracy. Again, the relative order of the models in terms of generalization does not depend on the choice of the training split.

This finding is studied more closely by ablating the Caltech101 dataset. We built a subset of Caltech101 by selecting 798 motorbike and 798 aircraft images as they are the classes with the largest number of samples. We trained the three architectures on this binary classification problem using different training and test splits and kept the previous hyperparameters. Figure 3 shows the performances of 5 models trained on each configuration. The test accuracy is plotted against the number of training images. Again, the models always have zero training error.

Just as in the case of the complete Caltech101 dataset, VGG is the best generalizing model. It achieves a test accuracy of about 95% with only 20 training samples per class, which is highly unexpected in such a highly over-parameterized regime. The curves of ResNet018 and ResNet101 are very close and cannot be distinguished from each other. Only for 20 training images per class,
We studied the generalization abilities of deep, high capacity Convolutional Neural Networks in the context of fine-grained image classification with very small training datasets. We trained the frequently used CNNs ResNet018, ResNet101, and VGG19 without augmentation, explicit regularization, or pretrained weights in order to measure their genuine generalization ability in highly over-parameterized regimes on the challenging benchmark datasets Caltech101, CUB_200-2011, FGVCAircraft, Flowers102, and StanfordCars. We were able to show that the generalization error is remarkably low already with very little training data and that the two larger CNNs performed consistently better than the smallest one. For a binary classification problem, we could show that a network with 140 million weights (VGG19) reaches 95% accuracy with only 20 training samples per class. This is against the narrative that Deep Neural Networks require large training data sets. We observed, that certain architectures work better on certain datasets, no matter the size of the training split. The relative generalization abilities between architectures do not seem to depend on the number of training images. This supports the idea that problems prefer certain models, but models do not prefer a specific training set size more than other models. We conclude, that very large Neural Networks can learn with very little data, also without any data augmentation, explicit regularization, or pretraining. Performance even can benefit from an increased number of network weights in these highly over-parametrized regimes. Training up to zero training error, i.e., complete over-fitting by memorizing the training data, does not harm generalization. All this is against traditional machine learning wisdom. At least in the case of binary image classification, explicit regularization and again the accuracy curves of single architectures with respect to the number of training samples do not intersect, which is in line with our previous observations. Already for 200 training samples, all three models converge to an almost perfect classification accuracy of 99%. The variance is below 1%. Again, bad solutions, which exist, do not occur.

4 CONCLUSION AND OUTLOOK

We studied the generalization abilities of deep, high capacity Convolutional Neural Networks in the context of fine-grained image classification with very small training datasets. We trained the frequently used CNNs ResNet018, ResNet101, and VGG19 without augmentation, explicit regularization, or pretrained weights in order to measure their genuine generalization ability in highly over-parameterized regimes on the challenging benchmark datasets Caltech101, CUB_200-2011, FGVCAircraft, Flowers102, and StanfordCars. We were able to show that the generalization error is remarkably low already with very little training data and that the two larger CNNs performed consistently better than the smallest one. For a binary classification problem, we could show that a network with 140 million weights (VGG19) reaches 95% accuracy with only 20 training samples per class. This is against the narrative that Deep Neural Networks require large training data sets. We observed, that certain architectures work better on certain datasets, no matter the size of the training split. The relative generalization abilities between architectures do not seem to depend on the number of training images. This supports the idea that problems prefer certain models, but models do not prefer a specific training set size more than other models. We conclude, that very large Neural Networks can learn with very little data, also without any data augmentation, explicit regularization, or pretraining. Performance even can benefit from an increased number of network weights in these highly over-parametrized regimes. Training up to zero training error, i.e., complete over-fitting by memorizing the training data, does not harm generalization. All this is against traditional machine learning wisdom. At least in the case of binary image classification, explicit regularization and again the accuracy curves of single architectures with respect to the number of training samples do not intersect, which is in line with our previous observations. Already for 200 training samples, all three models converge to an almost perfect classification accuracy of 99%. The variance is below 1%. Again, bad solutions, which exist, do not occur.

“over-fitting” and “the larger the network the more data is needed” do not apply in this realm.

ACKNOWLEDGMENTS

The work of Christoph Linse was supported by the Bundesministerium für Wirtschaft und Klimaschutz through the Mittelstand-Digital Zentrum Schleswig-Holstein Project.

REFERENCES

[1] Hamman Alshahazi, Christoph Linse, Erhardt Barth, and Thomas Martinetz. 2019. Ensembles of Deep Learning Models and Transfer Learning for Ear Recognition. sensors 19, 19 (2019), 4139. https://www.mdpi.com/1424-8220/19/19/4139
[2] Anne L. Alter and Karen M Wang. 2017. An Exploration of Computer Vision Techniques for Bird Species Classification.
[3] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. 2019. Reconciling modern machine-learning practice and the classical bias–variance trade-off. Proceedings of the National Academy of Sciences 116, 32 (Aug. 2019), 15849–15854. https://doi.org/10.1073/pnas.1903701116
[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition. Ieee, 248–255.
[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In 2015 IEEE International Conference on Computer Vision (ICCV). IEEE, Santiago, Chile, 1026–1034. https://doi.org/10.1109/ICCV.2015.123
[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Ieee, Las Vegas, NV, USA, 770–778. https://doi.org/10.1109/CVPR.2016.90
[7] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 2013. 3D Object Representations for Fine-Grained Categorization. In 2013 IEEE International Conference on Computer Vision Workshops. Ieee, Sydney, Australia, 554–561. https://doi.org/10.1109/ICCVW.2013.77
[8] Li Fei-Fei, R. Fergus, and P. Perona. 2004. Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. In 2004 Conference on Computer Vision and Pattern Recognition Workshop. Ieee, Washington, DC, USA, 178–178. https://doi.org/10.1109/CVPR.2004.383
[9] Shubhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. 2013. Fine-Grained Visual Classification of Aircraft. arXiv:1306.5151 [cs] (June 2013). http://arxiv.org/abs/1306.5151 arXiv: 1306.5151
[10] Maria-Elena Nilsback and Andrew Zisserman. 2008. Automated Flower Classification over a Large Number of Classes. In 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing. Ieee, Bhubaneswar, India, 722–729. https://doi.org/10.1109/ICVGIP.2008.47
[11] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In Proceedings of the 31st Conference on Neural Information Processing Systems. 1–4.
[12] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556 [cs] (April 2015). http://arxiv.org/abs/1409.1556 arXiv: 1409.1556
[13] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. 2011. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001. California Institute of Technology.
[14] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. 2010. Caltech-UCSD Birds-200. Technical Report CNS-TR-2010-001. California Institute of Technology.
[15] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaoqian Song, James Demmel, and Cho-Jui Hsieh. 2020. Large batch optimization for deep learning. Training BERT in 76 minutes. In International Conference on Learning Representations.
[16] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. arXiv:1611.03530 [cs] (Feb. 2017). http://arxiv.org/abs/1611.03530 arXiv: 1611.03530
[17] Hao Zhang, A.C. Berg, M. Maire, and J. Malik. 2006. SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (CVPR’06), Vol. 2. 2126–2136. https://doi.org/10.1109/CVPR.2006.301