ResNet strikes back: An improved training procedure in timm

Abstract

competitive training settings and pre-trained models

a vanilla ResNet-50 reaches 80.4% top-1 accuracy at resolution 224×224 on ImageNet-val without extra data or distillation

Introduction

$$\text{accuracy (model)} = f(\mathcal{A}, \mathcal{T}, \mathcal{N}),$$

A is the architecture design, T is the training setting along with its hyperparameters, and N is the measurement noise

Ideally, i.e., without resource and time constraints, one would optimally adopt the best possible training procedure for each architecture but realistically this is not possible

$$\mathcal{T}^{\star}(\mathcal{A}) = \max_{\mathcal{T}} f(\mathcal{A}, \mathcal{T}, \mathcal{N}),$$

When comparing architectures, most papers compare their results to other reported in older publications, but for which architectures were trained with potentially weaker recipes

이 논문에서 제시하는 것들

three training procedures intended to be strong baselines for a vanilla ResNet-50 used at inference resolution 224 × 224 - different numbers of epochs (100, 300 and 600)

we depart from the usual cross-entropy loss - binary cross entropy (Mixup / Cutmix)

stability of the accuracy over a large number of runs with different seeds, and discuss the overfitting issue

train popular architectures and re-evaluate their performance

discuss the necessity to optimize jointly the architecture and the training procedure

Related Work

Image Classification

The timm library

Pre-trained weights

implementations of many data augmentations, regularization techniques, optimizers, and learning rate schedulers

ResNet

some papers have also focused on ResNet-50 training, but they have either modified the architecture or changed the resolution, which does not allow for a direct comparison to the original ResNet-50 at resolution 224×224

Training ingredients & recipes

Training Procedures

Training Procedure	Number of epochs	Training resolution	Training time	Peak memory by GPU (MB)	Numbers of GPU	Top-1 accuracy val	real	v2
A1	600	224 × 224	110h	22,095	4	80.4	85.7	68.7
A2	300	224 × 224	55h	22,095	4	79.8	85.4	67.9
A3	100	160 × 160	15h	11,390	4	78.1	84.5	66.1

Table 1: Training resources used for our three training procedures on V100 GPUs and corre

Table 1: Training resources used for our three training procedures on V100 GPUs and corresponding accuracies at resolution 224×224 on ImageNet1k-val, -V2 and -Real. Note, the top-1 val acc. of pytorch-zoo [1] is 76.1%.

Procedure A1 aims at providing the best performance for ResNet-50. It is therefore the longest in terms of epochs (600) and training time (4.6 days on one node with 4 V100 32GB GPUs).

Procedure A2 is a 300 epochs schedule that is comparable to several modern procedures like DeiT, except with a larger batch size of 2048 and other choices introduced for all our recipes.

Procedure A3 aims at outperforming the original ResNet-50 procedure with a short schedule of 100 epochs and a batch size 2048. It can be trained in 15h on 4 V100 16GB GPUs and could be a good setting for exploratory research or studies.

Loss: multi-label classification objective

mixup / cutmix

binary cross-entropy (BCE)

Data-Augmentation

Regularization

weight decay + label smoothing, RepeatedAugmentation(RA) and stochastic-Depth

more regularization for longer training schedules.

RA와 stochastic depth는 초기단계에서 느리고 짧은 스케쥴에는 좋지 않다

putting more of RandAugment, mixup and stochastic depth regularization on top of A2 recipe

Optimization

larger batch (2048)

with repeated augmentation / the binary cross entropy loss, LAMB is good

LAMB w/ cosine schedule

Details of our ingredients and comparison to existing training procedures.

	Previous approaches					Ours		
Procedure → Reference	ResNet [13]	PyTorch [1]	FixRes [48]	DeiT [45]	FAMS (×4) [10]	A1	A2	A3
Train Res	224	224	224	224	224	224	224	160
Test Res	224	224	224	224	224	224	224	224
Epochs	90	90	120	300	400	600	300	100
# of forward pass	450k	450k	300k	375k	500k	375k	188k	63k
Batch size	256	256	512	1024	1024	2048	2048	2048
Optimizer	SGD-M	SGD-M	SGD-M	AdamW	SGD-M	LAMB	LAMB	LAMB
LR	0.1	0.1	0.2	1×10^{-3}	2.0	5×10^{-3}	5×10^{-3}	8×10^{-3}
LR decay	step	step	step	cosine	step	cosine	cosine	cosine
decay rate	0.1	0.1	0.1	-	$0.02^{t/400}$	-	-	-
decay epochs	30	30	30	-	1	-	-	-
Weight decay	10^{-4}	10^{-4}	10^{-4}	0.05	10^{-4}	0.01	0.02	0.02
Warmup epochs	✗	✗	✗	5	5	5	5	5
Label smoothing ε	✗	✗	✗	0.1	0.1	0.1	✗	✗
Dropout	✗	✗	✗	✗	✗	✗	✗	✗

Dropout	✗	✗	✗	✗	✗	✗	✗	✗
Stoch. Depth	✗	✗	✗	0.1	✗	0.05	0.05	✗
Repeated Aug	✗	✗	✓	✓	✗	✓	✓	✗
Gradient Clip.	✗	✗	✗	✗	✗	✗	✗	✗
H. flip	✓	✓	✓	✓	✓	✓	✓	✓
RRC	✗	✓	✓	✓	✓	✓	✓	✓
Rand Augment	✗	✗	✗	9/0.5	✗	7/0.5	7/0.5	6/0.5
Auto Augment	✗	✗	✗	✗	✓	✗	✗	✗
Mixup alpha	✗	✗	✗	0.8	0.2	0.2	0.1	0.1
Cutmix alpha	✗	✗	✗	1.0	✗	1.0	1.0	1.0
Erasing prob.	✗	✗	✗	0.25	✗	✗	✗	✗
ColorJitter	✗	✓	✓	✗	✗	✗	✗	✗
PCA lighting	✓	✗	✗	✗	✗	✗	✗	✗
SWA	✗	✗	✗	✗	✓	✗	✗	✗
EMA	✗	✗	✗	✗	✗	✗	✗	✗
Test crop ratio	0.875	0.875	0.875	0.875	0.875	0.95	0.95	0.95
CE loss	✓	✓	✓	✓	✓	✗	✗	✗
BCE loss	✗	✗	✗	✗	✗	✓	✓	✓
Mixed precision	✗	✗	✗	✓	✓	✓	✓	✓
Top-1 acc.	75.3%	76.1%	77.0%	78.4%	79.5%	80.4%	79.8%	78.1%

aim

1 quantifying the sensitivity of the performance to random factors

2 evaluating the overfitting by measuring on a different test set

Comparison of training procedures for ResNet-50

Performance comparison with other architectures.

Table 3: Comparison on ImageNet classification between other architectures trained with our ResNet-50 optimized training procedure **without any hyper-parameters adaptation**. In particular, our procedure must be adapted for deeper/larger models, which benefit from more regularization. For the training cost we report the training time (time) in hours, the number of GPU used (#GPU) and the peak memory by GPU (Pmem) in GB. For A1 and A2, we adopt the same training and test resolution as in the original publication introducing the architecture. For A3 we use a smaller training resolution to reduce the compute-time. †: torchvision [1] results. *: DeiT [45] results.

	A1-A2-org.		A3		Cost							ImageNet-1k-val			
	train	test	train	test	A1	A2	A1-A2		A3			A1	A2	A3	org.
↓ Architecture	res.	res.	res.	res.	time (hour)		# GPU	Pmem	time	# GPU	Pmem	Accuracy(%)			
ResNet-18 [13]†	224	224	160	224	186	93	2	12.5	28	2	6.5	71.5	70.6	68.2	69.8
ResNet-34 [13]†	224	224	160	224	186	93	2	17.5	27	2	9.0	76.4	75.5	73.0	73.3
ResNet-50 [13]†	224	224	160	224	110	55	4	22.0	15	4	11.4	80.4	79.8	78.1	76.1
ResNet-101 [13]†	224	224	160	224	74	37	8	16.3	8	8	8.5	81.5	81.3	79.8	77.4
ResNet-152 [13]†	224	224	160	224	92	46	8	22.5	9	8	11.8	82.0	81.8	80.6	78.3
RegNetY-4GF [32]	224	224	160	224	130	65	4	27.1	15	4	13.9	81.5	81.3	79.0	79.4
RegNetY-8GF [32]	224	224	160	224	106	53	8	19.8	10	8	10.3	82.2	82.1	81.1	79.9
RegNetY-16GF [32]	224	224	160	224	150	75	8	25.6	13	8	13.4	82.0	82.2	81.7	80.4
RegNetY-32GF [32]	224	224	160	224	120	60	16	17.6	12	16	9.4	82.5	82.4	82.6	81.0
SE-ResNet-50 [20]	224	224	160	224	102	51	4	27.6	16	4	14.2	80.0	80.1	77.0	76.7
SENet-154 [20]	224	224	160	224	110	55	16	23.3	12	16	12.2	81.7	81.8	81.9	81.3
ResNet-50-D [14]	224	224	160	224	100	50	4	23.9	14	4	12.3	80.7	80.2	78.7	79.3
ResNeXt-50-32x4d [51]†	224	224	160	224	80	40	8	14.3	15	4	14.6	80.5	80.4	79.2	77.6
EfficientNet-B0 [41]	224	224	160	224	110	55	4	22.1	15	4	11.4	77.0	76.8	73.0	77.1
EfficientNet-B1 [41]	240	240	160	224	62	31	8	17.9	8	8	7.9	79.2	79.4	74.9	79.1
EfficientNet-B2 [41]	260	260	192	256	76	38	8	22.8	9	8	11.9	80.4	80.1	77.5	80.1

EfficientNet-B3 [41]	300	300	224	288	62	31	16	19.5	6	16	10.1	81.4	81.4	79.2	81.6
EfficientNet-B4 [41]	380	380	320	380	64	32	32	20.4	8	32	14.3	81.6	82.4	81.2	82.9
ViT-Ti [45]*	224	224	160	224	98	49	4	16.3	14	4	7.0	74.7	74.1	66.7	72.2
ViT-S [45]*	224	224	160	224	68	34	8	16.1	8	8	7.0	80.6	79.6	73.8	79.8
ViT-B [11]*	224	224	160	224	66	33	16	16.4	5	16	7.3	80.4	79.8	76.0	81.8
timm [50] specific architectures															
ECA-ResNet-50-T	224	224	160	224	112	56	4	29.3	15	4	15.0	81.3	80.9	79.6	-
EfficientNetV2-rw-S [42]	288	384	224	288	52	26	16	16.6	7	16	10.1	82.3	82.9	80.9	83.8
EfficientNetV2-rw-M [42]	320	384	256	352	64	32	32	18.5	9	32	12.1	80.6	81.9	82.3	84.8
ECA-ResNet-269-D	320	416	256	320	108	54	32	27.4	11	32	17.8	83.3	83.9	83.3	85.0

Table 4: **Performance of models trained with A1 training procedure.** We measure peak memory and throughput on one GPU V100 32GB with batch size 128, FP16 precision and test resolution from Table 3. Note that the throughput is indicative, since it depends on the GPU hardware, the software that runs the models, and other factors like the adjustment of batch size (we keep it fix in this table).

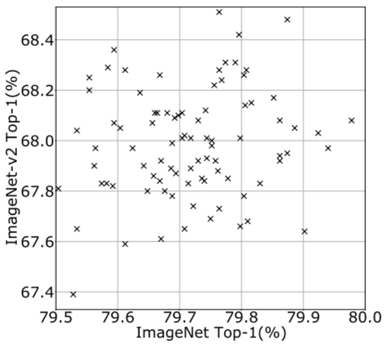
Architecture	# params $\times 10^6$	FLOPs $\times 10^9$	Throughput (im/s)	Peak mem (MB)	Top-1 Acc.	Real Acc.	V2 Acc.
ResNet-18 [13]	11.7	1.8	7960.5	588	71.5	79.4	59.4
ResNet-34 [13]	21.8	3.7	4862.6	642	76.4	83.4	65.1
ResNet-50 [13]	25.6	4.1	2536.6	1,155	80.4	85.7	68.7
ResNet-101 [13]	44.5	7.9	1547.9	1,264	81.5	86.3	70.3
ResNet-152 [13]	60.2	11.6	1094.0	1,355	82.0	86.4	70.6
RegNetY-4GF [32]	20.6	4.0	1690.6	1,585	81.5	86.7	70.7
RegNetY-8GF [32]	39.2	8.1	1122.3	2,139	82.2	86.7	71.1
RegNetY-16GF [32]	83.6	16.0	694.1	3,052	82.0	86.4	71.2
RegNetY-32GF [32]	145.0	32.4	431.5	3,366	82.5	86.6	71.7
SE-ResNet-50 [20]	28.1	4.1	2174.8	1,193	80.0	85.8	68.8
SENet-154 [20]	115.1	20.9	511.5	2,414	81.7	86.0	71.2
ResNet-50-D [14]	25.6	4.4	2418.8	1,205	80.7	85.9	68.9
ResNeXt-50-32x4d [51]	25.0	4.3	1727.5	1,247	80.5	85.5	68.4
EfficientNet-B0 [41]	5.3	0.4	3701.5	932	77.0	83.8	65.0
EfficientNet-B1 [41]	7.8	0.7	2365.2	1,077	79.2	85.3	67.7
EfficientNet-B2 [41]	9.2	1.0	1786.8	1,318	80.4	86.0	69.3
EfficientNet-B3 [41]	12.0	1.8	1082.4	2,447	81.4	86.7	70.4
EfficientNet-B4 [41]	19.0	4.2	561.3	5,058	81.6	85.9	70.8
ViT-Ti [45]	5.7	1.3	3497.7	346	74.7	82.1	62.4
ViT-S [45]	22.0	4.6	1762.3	682	80.6	85.6	69.4
ViT-B [11]	86.6	17.6	771.0	1,544	80.4	84.8	69.4
timm [50] specific architectures							
ECA-ResNet50-T	25.6	4.4	2139.7	1,155	81.3	86.1	69.9
EfficientNetV2-rw-S [42]	23.9	8.8	823.1	2,339	80.6	84.8	69.2
EfficientNetV2-rw-M [42]	53.2	18.5	456.8	2,916	82.3	87.1	71.7
ECA-Resnet269-D	102.1	70.6	168.1	4,134	83.3	86.9	71.9

Significance of measurements: seed experiments

weight initialization, but also for the optimization procedure - inherent random

we measure the distribution of performance when changing the random generator choices

exist of outliers significantly outperforming or underperforming the average outcome of a traing procedure



	Top-1 accuracy (%)				
dataset ↓	mean	std	max	min	seed 0
ImageNet-val	79.72	0.10	79.98	79.50	79.85
ImageNet-real	85.37	0.08	85.55	85.21	85.45
ImageNet-V2	67.99	0.23	68.69	67.39	67.90

Figure 1: *Top* ↑: Statistics for ResNet-50 trained with A2 and 100 different seeds. The column "seed 0" corresponds to the weights that we take as reference. Its performance is +0.13% above the average top-1 accuracy on Imagenet-val.

← *Left:* Point cloud plotting the ImageNet-val top-1 accuracy vs ImageNet-V2 for all seeds. Note that the outlying seed that achieves 68.5% top-1 accuracy on ImageNet-V2 has an average performance on ImageNet-val.

Peak performance and control of overfitting

However optimizing over a large number of choices typically leads to overfitting.

whether this model is intrinsically better than the average ones, or if it was just lucky on this particular measurement set.

we compute for all the seeds the couples

some significant measurement noise, which advocates to report systematically the performance on different datasets, more particularly one making a clear distinction between validation and test.

More on sensitivity analysis: variance along epochs.

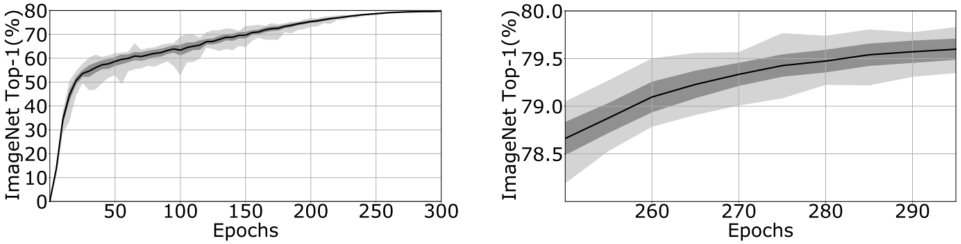


Figure 3: We show how the mean, standard deviation, minimum and maximum of the top-1 accuracy on ImageNet-val evolves during training with the A2 procedure (ResNet-50 architecture). **(Left)** For all 300 training epochs. **(Right)** Same but for the last epochs. We note that the variance in accuracy is high at the beginning, see for instance at epoch 100, where the difference

variance in accuracy is high at the beginning, see for instance at epoch 100, where the difference in performance can be as large as 10% in accuracy. Towards the end of the training, most of the networks converge to similar values and the range significantly decreases in the last 50 epochs. *Credit*: this figure and experiment was inspired by Picard [30].

Transfer Learning

Table 5: Performance comparison on transfer-learning tasks for different pre-training recipes.

Dataset	Train size	Test size	#classes	Pytorch [1]	A1	A2	A3
ImageNet-val [36]	1,281,167	50,000	1000	76.1	**80.4**	79.8	78.1
iNaturalist 2019 [18]	265,240	3,003	1,010	73.2	73.9	**75.0**	73.8
Flowers-102 [29]	2,040	6,149	102	**97.9**	**97.9**	**97.9**	97.5
Stanford Cars [24]	8,144	8,041	196	92.5	**92.7**	92.6	92.5
CIFAR-100 [25]	50,000	10,000	100	86.6	**86.9**	86.2	85.3
CIFAR-10 [25]	50,000	10,000	10	98.2	**98.3**	98.0	97.6

fine-tuning tend to smooth the difference of performance on certain datasets, such as CIFAR or Stanford Cars

Comparing architectures and training procedures: a show-case of contradictory conclusions

	test set →	ImageNet-val		ImageNet-v2	
↓ architecture	training →	A2	T2	A2	T2
ResNet-50		79.9	79.2	67.9	67.9
DeiT-S		79.6	80.4	68.1	69.2

parameter 수는 비슷한데 학습 방식에 따라 성능 비교가 달라지더라

Ablations

Main ingredients and hyper-parameters

loss	LR	WD	RA	A2
BCE	2×10^{-3}	0.02	✓	78.24
BCE	2×10^{-3}	0.03	✓	78.47
BCE	3×10^{-3}	0.02	✓	79.16
BCE	3×10^{-3}	0.03	✓	79.28
BCE	5×10^{-3}	0.01	✓	79.66
BCE	5×10^{-3}	0.02	✓	79.85
BCE	5×10^{-3}	0.03	✓	79.73
BCE	8×10^{-3}	0.02	✓	79.63
BCE	3×10^{-3}	0.02	✗	78.74
BCE	5×10^{-3}	0.02	✗	79.57
BCE	5×10^{-3}	0.03	✗	79.58
CE	2×10^{-3}	0.02	✓	77.37

CE	3×10^{-3}	0.02	✓	78.22
CE	5×10^{-3}	0.02	✓	79.18
CE	5×10^{-3}	0.03	✓	79.23
CE	5×10^{-3}	0.05	✓	79.31
CE	8×10^{-3}	0.03	✓	79.12
CE	3×10^{-3}	0.02	✗	77.71
CE	5×10^{-3}	0.01	✗	78.93
CE	5×10^{-3}	0.02	✗	79.00
CE	5×10^{-3}	0.03	✗	78.62
CE	8×10^{-3}	0.02	✗	78.72

Learning rate and Weight Decay.

Loss: Binary Cross Entropy versus Cross Entropy

mixup	Rep. aug.	RandA	label smooth.	stoch. depth	BCE target	top-1 acc.
0.1	✓	7	✗	0.05	✓	79.85
0.2	✗					79.62
0.2		6				79.61
0.05						79.57
					✗	79.57

Repeated augmentation

Stochastic Depth & Smoothing

drop-factor	A1	A2	A3
0	79.94	79.79	78.06
0.05	80.38	79.85	77.57
0.1	80.12	79.62	77.32
smoothing			
✗	80.22	79.85	78.06
✓	80.38	79.58	77.99

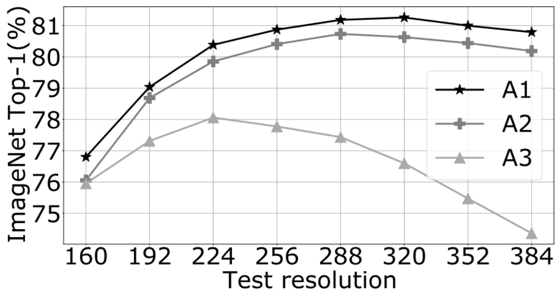
Augmentation

Crop-ratio

	A1			A2			A3		
crop-ratio	mean (*std*)	max – min	seed 0	mean (*std*)	max – min	seed 0	mean (*std*)	max – min	seed 0
0.875	80.18 (*0.14*)	80.45 – 79.90	80.14	79.67 (*0.08*)	79.91 – 79.59	79.91	77.69 (*0.10*)	77.85 – 77.48	77.69
0.9	80.22 (*0.15*)	**80.54** – 79.98	80.25	79.73 (*0.09*)	79.89 – 79.56	79.75	77.86 (*0.09*)	78.01 – 77.62	77.83

0.9	80.22 (*0.15*)	**80.54** – 79.98	80.25	79.73 (*0.09*)	79.89 – 79.56	79.75	77.86 (*0.09*)	78.01 – 77.62	77.83
0.95	80.24 (*0.14*)	80.49 – 79.91	80.38	79.68 (*0.09*)	79.85 – 79.57	79.85	78.00 (*0.09*)	78.09 – 77.83	78.06
1.0	80.15 (*0.11*)	80.15 – 79.66	80.19	79.58 (*0.13*)	79.88 – 79.32	79.88	78.02 (*0.10*)	78.16 – 77.83	77.93

Evaluation at other resolutions



Conclusion

new training procedures for a vanilla ResNet-50

we have established the new state of the art for training this gold-standard model

we do not claim that our procedures are universal, quite the opposite