Supplementary Material: Nondeterminism and Instability in Neural Network Optimization

Cecilia Summers [1] **Michael J. Dinneen** [1]

A. Linear, 2-Layer, and ResNet-10 Results

In Table A1 we include results for linear networks, 2-layer networks (1 hidden layer), and a ResNet-10 on CIFAR-10, omitted from the main text due to space constraints. As noted in the main text, parameter initialization generally has less effect for linear models, and random bit changes in particular have nearly no effect on linear models, highlighting the stability of SGD in optimizing linear models. Also of note is the relative smaller effect of a single bit change for a 2-layer network where the hidden layer is convolutional — still much larger than for the linear model, but significantly smaller than for any other non-linear model. This suggests a similar effect to what was previously observed on MNIST (Table 6) in that degree of instability might be related to the interplay of model, dataset, and the degree of overfitting.

Otherwise, these results follow the results in the main text, wherein each source of nondeterminism has roughly the same effect as each other, and a majority of this is due to the instability of optimization, evidenced by the high variability in models with only random bit changes at initialization. Test-time augmentation also remains effective in reducing model variability as compared to "All Nondeterminism Sources", the setting TTA is applied to.

B. Impact of Random Bit Changes Over Time

In Fig. A1 we plot the effect of a random bit change for a linear and single hidden layer model on CIFAR-10, illustrating the effect of instability as described in Sec. 4. In the first few epochs of training, we observe that the standard deviation and range of cross-entropy for the model with one hidden layer quickly grows, only eventually decreasing much later in training as the model's parameters converges toward their final values. On the other hand, for linear models, the standard deviation consistently remains 5 or more orders of magnitude lower throughout training.

[1]Department of Computer Science, University of Auckland, Auckland, New Zealand. Correspondence to: Cecilia Summers <cecilia.summers.07@gmail.com>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

C. Impact of Network Width

Here we present results studying the interaction of instability and model width, training variants of the ResNet-14 on CIFAR-10 with width modifiers ranging from 0.125 to 8 in powers of two. Results are given in Table A2. Across model widths, we observe that instability still produces roughly as much variability as all other nondeterminism sources combined, as shown by the relative variability of models trained with "Random Bit Change" as a source of nondeterminism compared with those trained with all other sources ("All Sources"). The only minor outlier was the width multiplier of 0.125, for which random bit changes produced somewhat less variability, reminiscent of the experiments on MNIST (Table 6). We also observe that variability generally decreases as model width increases (except for "Pairwise Corr."), though it is difficult to say whether this is due to a unique property of model width or simply wider models having higher test set performance.

D. Test-Time Augmentation Details

CIFAR-10. On CIFAR-10, in addition to TTA with horizontal flipping (*i.e.* ensembling model predictions on the original image with its horizontally-flipped version), we also used a form of TTA with cropping. Our usage of crop-based TTA was based on the version of cropping used as data augmentation during model training, in which each image was zero-padded by four pixels along each side, after which a random 32×32 crop was drawn. In the main text, we experimented with 25- and 81-crop TTA variants, where the 81-crop variant uses all possible crops $[(4 \cdot 2 + 1)^2 = 81]$, and the 25-crop variant uses a stride of 2 when sampling the possible crops $[(2 \cdot 2 + 1)^2 = 25]$. When adding in horizontally-flipped versions, the number of TTA examples doubles, for a maximum of 162 augmented versions of the original image.

ImageNet. On ImageNet, the standard evaluation protocol we use for our experiments first resizes each image to have its smaller side be length 256, after which the central 224×224 crop is taken. For TTA, besides the horizontal image flipping used in CIFAR-10, we also experimented with crops ('Crop-TTA' and 'Flip-Crop-TTA') as follows:

Nondeterminism Source	Accuracy SD (%)	Cross-Entropy SD	Pairwise Disagree (%)	Pairwise Corr.	Ensemble Δ (%)
CIFAR-10: Linear model					
Parameter Initialization	$0.03 \pm$ 2e-3	$0.0002 \pm$ 1e-5	0.5	0.997	-4e-3
All Nondeterminism Sources	0.10 ± 0.01	$0.0007 \pm$ 4e-5	5.5	0.996	0.06
Random Bit Change	0.00 ± 0.00	1e-7 $\pm$ 1e-8	0.0	1.000	0.00
Single/Flip-Crop-TTA	$0.05 \pm$ 3e-3	$0.0001 \pm$ 1e-5	0.9	0.998	-3e-3
CIFAR-10: One hidden layer (fully-connected)					
Parameter Initialization	0.31 ± 0.02	0.0051 ± 0.0003	24.2	0.941	1.38
All Nondeterminism Sources	0.30 ± 0.02	0.0054 ± 0.0004	24.9	0.937	1.49
Random Bit Change	0.28 ± 0.02	0.0051 ± 0.0004	23.4	0.945	1.32
Single/Flip-Crop-TTA	0.15 ± 0.01	0.0017 ± 0.0001	7.1	0.993	0.15
CIFAR-10: One hidden layer (convolutional)					
Parameter Initialization	0.26 ± 0.02	0.0040 ± 0.0003	12.5	0.974	0.64
All Nondeterminism Sources	0.22 ± 0.01	0.0042 ± 0.0003	12.7	0.973	0.68
Random Bit Change	0.14 ± 0.01	0.0022 ± 0.0002	6.4	0.993	0.18
Single/Flip-Crop-TTA	0.19 ± 0.01	0.0033 ± 0.0002	7.5	0.989	0.24
CIFAR-10: ResNet-10					
Parameter Initialization	0.23 ± 0.01	0.0060 ± 0.0003	13.7	0.912	2.13
All Nondeterminism Sources	0.23 ± 0.01	0.0065 ± 0.0004	13.6	0.911	2.13
Random Bit Change	0.25 ± 0.02	0.0065 ± 0.0005	13.5	0.913	2.08
Single/Flip-Crop-TTA	0.24 ± 0.02	0.0047 ± 0.0003	9.5	0.943	1.22
Acc. Ens.	0.23 ± 0.01	0.0047 ± 0.0003	8.8	0.962	0.96
Acc. Ens./Flip-Crop-TTA	0.18 ± 0.01	0.0035 ± 0.0002	6.7	0.973	0.58

Table A1. Linear, 2-layer, and ResNet-10 experiments on CIFAR-10.

Setting	Accuracy SD (%)	Cross-Entropy SD	Pairwise Disagree (%)	Pairwise Corr.	Ensemble Δ (%)
0.125/All Sources (*N=30*)	0.86 ± 0.10	0.0183 ± 0.0020	40.8	0.887	2.04
0.125/Random Bit Change (*N=30*)	0.66 ± 0.10	0.0133 ± 0.0020	38.0	0.943	1.73
0.25/All Sources (*N=30*)	0.59 ± 0.06	0.0124 ± 0.0012	25.9	0.880	2.37
0.25/Random Bit Change (*N=30*)	0.71 ± 0.08	0.0145 ± 0.0019	25.2	0.908	2.15
0.5/All Sources (*N=30*)	0.33 ± 0.04	0.0077 ± 0.0010	15.8	0.890	2.21
0.5/Random Bit Change (*N=30*)	0.30 ± 0.04	0.0076 ± 0.0008	15.0	0.917	1.98
1.0/All Sources (*N=100*)	0.26 ± 0.02	0.0072 ± 0.0005	10.7	0.871	1.82
1.0/Random Bit Change (*N=100*)	0.21 ± 0.01	0.0068 ± 0.0004	10.6	0.874	1.82
2.0/All Sources (*N=30*)	0.13 ± 0.01	0.0066 ± 0.0008	7.1	0.790	1.19
2.0/Random Bit Change (*N=30*)	0.18 ± 0.02	0.0054 ± 0.0012	7.0	0.798	1.13
4.0/All Sources (*N=30*)	0.14 ± 0.02	0.0049 ± 0.0006	5.0	0.781	0.75
4.0/Random Bit Change (*N=30*)	0.15 ± 0.02	0.0058 ± 0.0007	4.9	0.785	0.74
8.0/All Sources (*N=30*)	0.13 ± 0.02	0.0038 ± 0.0005	3.8	0.807	0.53
8.0/Random Bit Change (*N=30*)	0.11 ± 0.01	0.0039 ± 0.0004	3.8	0.809	0.50

Table A2. Experiments varying model width for ResNet-14 on CIFAR-10. In each row, the experimental setting is abbreviated by [width multiplier]/[sources of nondeterminism] (*N*=[number of models trained]).

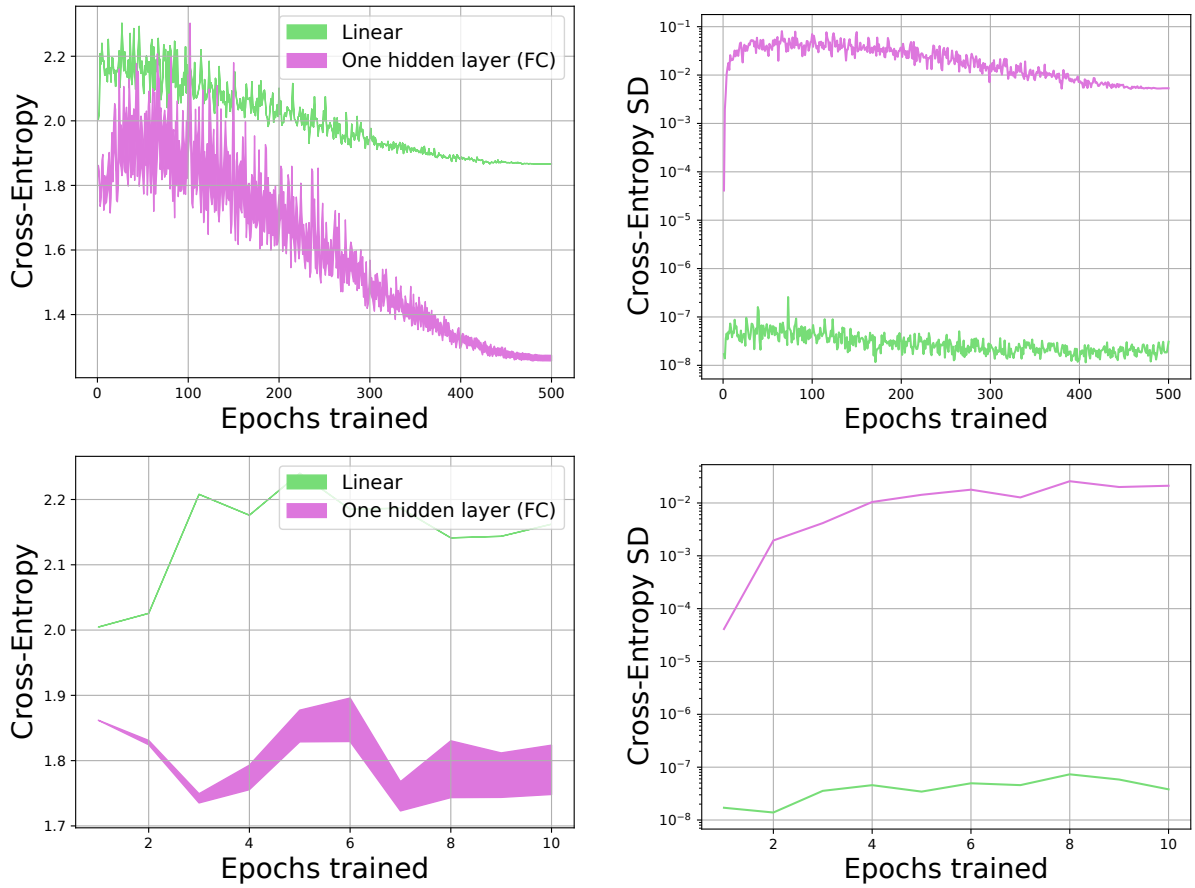

Figure A1. The impact of a random bit change during initialization for linear models vs 2-layer models with a single fully-connected hidden layer, where row 1 considers the full 500 epochs of training, and row 2 zooms in on the first 10 epochs. The left column of each row gives the range of cross-entropy values for 100 models in the middle 95th percentile of cross-entropy, plotted at each epoch. The right column of each row presents the standard deviation of these models, roughly corresponding to half the width of the corresponding range in the left-hand plots.

after each image is resized to have its smaller side length 256, a central 256×256 crop is taken, and then 9 crops of size 224×224 are taken in a 3×3 grid, starting from the top-left, and where the spacing between crops in the grid is 16 pixels.

E. Approaches That Don't Reduce Instability

In the process of finding an approach that reduces run-to-run variability of models (Sec. 5), we experimented with many approaches which all failed to make a dent in improving variability and stability. For the benefit of the field, here we provide our experiences with these approaches which did not succeed in improving stability, despite the intuitive arguments for why they might help.

Learning Rate and Duration of Training. Noticing that the effects of nondeterminism seemed to accumulate during the course of training (Fig. 2), it seemed reasonable that varying the learning rate or duration of training might have an effect. However, varying the duration of training from anywhere between 50 and $2{,}000$ epochs on CIFAR-10 all produced models with a similar variance in performance as the results in the rest of this work (which used 500 epochs), even though the absolute performance differed by up to $\sim 2\%$.

We show these results in Table A3. In general, increasing the number of epochs or changing the learning rate did not change the variability in performance (Accuracy SD; Cross-Entropy SD) much, with only a very slight increase in variability as the number of epochs grew to extremely large values (*i.e.* 2,000 epochs). There were slightly larger changes in pairwise representation-based metrics, where training longer again increased run-to-run variability. However, none of these attempts actually reduced variability while maintaining performance; they only served to potentially make it larger.

As part of these experiments, we also verified the effects of instability with only 200 epochs of training and the effectiveness of accelerated ensembling techniques ("Acc. Ens.") with this reduced training time, given in the last three rows of Table A3.

Choice of Optimizer. Since instability and nondeterminism are both a property of optimization, it is conceivable that use of a different optimizer might be able to lessen the degree of instability in model training. We experimented with SGD using various values of momentum, ranging from 0 for pure SGD to 0.999 for a momentum-driven optimizer, but none succeeded in reduce instability. In addition, we experimented with Adam (Kingma & Ba, 2014), picked as a representative of the class of adaptive learning rate algorithms, but this, too, had no effect on stability.

Aggressive Stochastic Weight Averaging. Inspired by the success found by Madhyastha & Jain (2019), we tried Aggressive Stochastic Weight Averaging (ASWA), a variant of SWA (Izmailov et al., 2018). However, we could not get the model to converge to a reasonable degree of performance with the original formulation due to update sizes that decreased too rapidly, and though we were able to modify it to converge successfully, the output variance remained as high as the other models.

Gradient Clipping. With the intuition that instability might be caused by spurious gradients of large magnitude, we experimented with clipping the norm of gradients (using `pytorch`'s implementation of `torch.nn.utils.clip_grad.clip_grad_norm_`. Like other approaches, though, this had no effect on model variability.

Weight Augmentation. A very experimental approach, to reduce instability we experimented with taking an averaged gradient around the current set of parameters at each step, approximated by sampling a random weight offset before doing a forward or backward pass through the model. Intuitively, this might encourage optimization to not be too sensitive to the current value of weights; however, in practice this simply didn't affect the variance or stability of the model.

F. Accelerated Ensembling in Language Modeling

Although the accelerated ensembling technique we employed in the main text, Snapshot Ensembles (Huang et al., 2017), was only designed for image classification, we have also experimented with its usage for language modeling (see Sec. 3.3 for problem setup). We present results in Table A4, where we additionally compare models trained for 500 and 1000 epochs. In both cases, accelerated ensembling resulted in lower model variability when considering pairwise metrics (reducing the fraction of tokens models disagreed on and reducing the PPL improvement from ensembling). However, the variability in PPL was more mixed, and in fact we note that the accelerated ensembles actually had higher average PPL than their counterparts (*e.g.* 75.0 for the accelerated ensemble vs 73.0 for the regular model), indicating that alternative accelerated ensembling techniques may be warranted for language modeling.

G. Subtleties in Evaluation

Choice of Seeds. While we have done our best to make our experimental protocol straightforward and easy to interpret, one subtlety related to seed selection arises when examining results. First, recall that in our example from Sec. 3.1,

Setting	Accuracy SD (%)	Cross-Entropy SD	Pairwise Disagree (%)	Pairwise Corr.	Ensemble Δ (%)
All Sources/.40/50 (*N=20*)	0.31 ± 0.04	0.0070 ± 0.0007	11.4	0.922	1.54
All Sources/.40/100 (*N=20*)	0.26 ± 0.04	0.0068 ± 0.0015	10.8	0.909	1.71
All Sources/.40/250 (*N=20*)	0.19 ± 0.03	0.0054 ± 0.0009	10.7	0.889	1.78
All Sources/.40/500 (*N=100*)	0.26 ± 0.02	0.0072 ± 0.0005	10.7	0.871	1.82
All Sources/.40/2000 (*N=20*)	0.24 ± 0.02	0.0096 ± 0.0013	11.2	0.828	2.08
Shuffle/.40/500 (*N=100*)	0.25 ± 0.02	0.0082 ± 0.0005	10.6	0.871	1.81
Shuffle/.20/500 (*N=100*)	0.23 ± 0.02	0.0071 ± 0.0005	11.0	0.858	1.95
Shuffle/.20/1000 (*N=100*)	0.21 ± 0.02	0.0088 ± 0.0005	11.1	0.837	2.02
Shuffle/.10/500 (*N=100*)	0.20 ± 0.01	0.0076 ± 0.0005	11.6	0.845	2.08
Shuffle/.10/2000 (*N=100*)	0.24 ± 0.02	0.0100 ± 0.0006	11.6	0.801	2.19
Param. Init/.40/500 (*N=100*)	0.23 ± 0.02	0.0074 ± 0.0005	10.7	0.872	1.82
Param. Init/.20/500 (*N=100*)	0.23 ± 0.02	0.0084 ± 0.0005	11.0	0.859	1.97
Param. Init/.20/1000 (*N=100*)	0.25 ± 0.02	0.0095 ± 0.0007	11.1	0.836	2.06
Param. Init/.10/500 (*N=100*)	0.26 ± 0.02	0.0083 ± 0.0005	11.7	0.844	2.13
Param. Init/.10/2000 (*N=100*)	0.22 ± 0.01	0.0093 ± 0.0008	11.6	0.800	2.18
All Sources/.40/200 (*N=100*)	0.23 ± 0.02	0.0076 ± 0.0004	10.6	0.895	1.75
Random Bit/.40/200 (*N=100*)	0.21 ± 0.01	0.0067 ± 0.0004	10.3	0.897	1.70
Acc. Ens./All Sources/.40/200 (*N=100*)	0.21 ± 0.01	0.0046 ± 0.0003	6.6	0.963	0.68

Table A3. Experiments varying the learning rate and number of epochs for ResNet-14 on CIFAR-10. In each row, the experimental setting is abbreviated by [sources of nondeterminism]/[maximum learning rate]/[number of epochs] (*N*=[number of models trained]), with the exception of the last row, which is a Snapshot ensemble but otherwise follows the same format.

Setting	PPL SD	Pairwise Disagree (%)	Ensemble PPL Δ
All Nondeterminism Sources/500	0.18 ± 0.01	17.4	-2.07
Acc. Ens./All Sources/500	0.21 ± 0.02	13.7	-1.33
All Nondeterminism Sources/1000	0.17 ± 0.01	17.6	-2.08
Acc. Ens./All Sources/1000	0.16 ± 0.01	14.1	-1.34

Table A4. The effects of accelerated model ensembling on Penn Treebank; 100 runs per row. "Acc. Ens." indicates accelerated ensembling, and the trailing number in each setting name is the number of epochs models are trained for.

testing the effects of random initialization corresponded to training models for $(S_1, S_2, S_3) \in \{(i, 1, 1)\}_{i=1}^{R}$, where S_1 was the seed for random initialization, S_2 was the seed for training data shuffling, and S_3 was set to 1 to indicate the deterministic mode for cuDNN. The subtlety arises in that the resulting distribution of $(S_1, S_2, S_3) \in \{(i, 1, 1)\}_{i=1}^{R}$ is not necessarily the same as the distribution where S_2 is set to a different arbitrary constant value, *e.g.* $S_2 = 2$. Due to this, there may be minor discrepancies when comparing the diversity in performance between two different sources of nondeterminism (though unlikely to change general conclusions unless the magnitude of the discrepancy is very large). Combined with the natural sampling variability implicit in only training a finite number of models, this can lead to paradoxical results such as the standard deviation for a particular metric being slightly higher for a random bit change as compared to an entirely different random parameter initialization. While we have separately validated that the general conclusions of our results hold when varying a few of these constant factors (*i.e.* running experiments where S_2 is set to 2 and 3, in this example), it is difficult to resolve the discrepancy entirely without models according to the full cross-product of random seeds, which is prohibitive due to the exponential amount of required computation.

Variability vs Model Performance. Another challenge when interpreting results is that model variability covaries with model performance, which impacts validating any approach that affects both model variability and performance. Though this is a difficult evaluation issue to solve in general, it is possible to show that the reductions in variability from Accelerated Ensembling and Test-Time Augmentation are not simply due to this: For example, "Acc. Ens./Flip-Crop81-TTA" for ResNet-14 (Table 5) has lower variability across all metrics than "All Nondeterminism Sources" for ResNet-18 (Table 6), despite the ResNet-18 having higher accuracy (92.0% vs 94.9%). Similar trends hold for ResNet-10 with Acc. Ens. and TTA (Table A1) vs ResNet-14 (88.7% vs 89.8%) and ResNet-6 vs ResNet-10 (77.9% vs 86.1%), albeit only visible on certain variability metrics in the latter case.

References

Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J. E., and Weinberger, K. Q. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*, 2017.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

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

Madhyastha, P. and Jain, R. On model stability as a function of random seed. In *Conference on Computational Natural Language Learning*, pp. 929–939, 2019.