Supplementary Materials for "Tilting the playing field: Dynamical loss functions for machine learning"

Miguel Ruiz-García,[1,2] Ge Zhang,[1] Samuel S. Schoenholz,[3] and Andrea J. Liu[1]

[1]*Department of Physics and Astronomy, University of Pennsylvania, Philadelphia, PA 19104, USA*

[2]*Department of Applied Mathematics, ETSII, Universidad Politécnica de Madrid, Madrid, Spain*

[3]*Google Research: Brain Team*

(Dated: June 11, 2021)

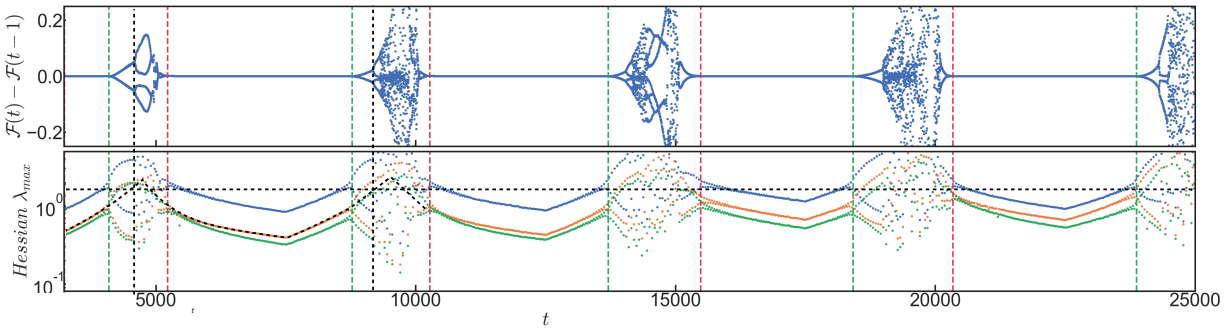

FIG. 1. Tracking the three largest eigenvalues of the Hessian. As in figure 4 of the main text, we use a neural network whose hidden layer has 100 units, $T = 5000$ and $A = 70$. Upper panel shows $\mathcal{F}(t) - \mathcal{F}(t-1)$ to display the instabilities more clearly. Bottom panel shows the three largest eigenvalues of the Hessian of the loss function computed using the Lanczos algorithm as described in [1] (we have used an implementation in Google-JAX [2]). Vertical green and red dashed lines mark the times at which Hessian $\lambda_{max}(t) - \lambda_{max}(t-1) \sim 0.1$ corresponding to the start and finish of the instabilities. Averaging Hessian λ_{max} at these times we get the horizontal dashed line in the bottom panel, the threshold above which instabilities occur. We have included a new dashed line approximately following the second largest eigenvalue (orange line). Subsequent bifurcations seem to occur when smaller eigenvalues cross the same threshold.

I. SECOND AND THIRD LARGEST EIGENVALUES OF THE HESSIAN TRIGGER SUBSEQUENT BIFURCATIONS.

We replot here two panels of Fig. 4 of the main text, see Fig. 1. It shows the behavior of the system as it descends in the dynamical loss function landscape. We use the spiral dataset for a case with a rather high period of $T = 5000$ minimization steps and amplitude of $A = 70$, chosen for ease of visualization. In Fig. 1 we include the three largest eigenvalues of the Hessian instead of only one. Indeed, second and third bifurcations seem to correspond to the second and third largest eigenvalues crossing the same threshold. Since all eigenvalues are also affected by the bifurcations, we have included new dashed lines to guide the eye.

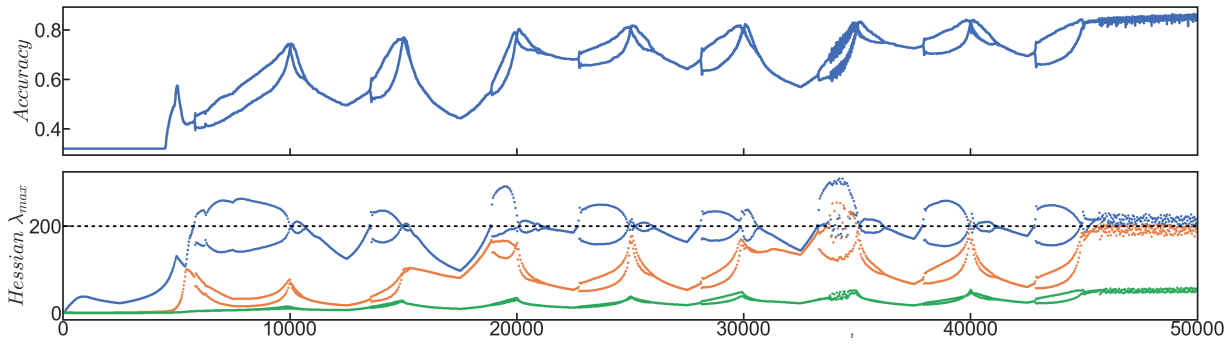

FIG. 2. Bifurcations correspond to eigenvalues of the Hessian crossing a threshold when using Myrtle5 to classify a subset of CIFAR10. For simplicity we use a subset of CIFAR10 with 3000 samples of the first three classes. Bifurcations are clearly present in the Accuracy and they occur when the largest eigenvalue of the Hessian crosses a threshold marked with a dashed line.

II. HESSIAN EIGENVALUES ALSO TRIGGER BIFURCATIONS USING MYRTLE5 AND CIFAR10

In the main text we have studied in detail the dynamics of learning with dynamical loss functions using the spiral dataset because the size of the model and the dataset made it computationally much cheaper, however, bifurcations are a general effect that is also present when using other models and datasets. To show this, we include here an example using Myrtle5 to classify CIFAR10 with a dynamical loss function as the one described in the main text. To make the training dataset more tractable we use a subset of 3000 samples of the first 3 classes of CIFAR10. Fig. 2 shows that the dynamics follow the same phenomenology described in detail with the spiral dataset in the main text. The accuracy presents bifurcations analogous to Fig. 4 in the main text, and they occur when the largest eigenvalues of the Hessian cross a threshold.

III. STOPPING THE OSCILLATIONS AT THE END OF TRAINING

Fig. 1 of the main text shows two phase diagrams for the dynamical loss function applied to Myrtle5 and CIFAR10. This neural network was adapted from [3]. We used 64 channels, Nesterov optimizer with momentum = 0.9, minibatch size 512, a linear learning rate schedule starting at 0, reaching 0.02 in the epoch 300 and decreasing to 0.002 in the final epoch (700).

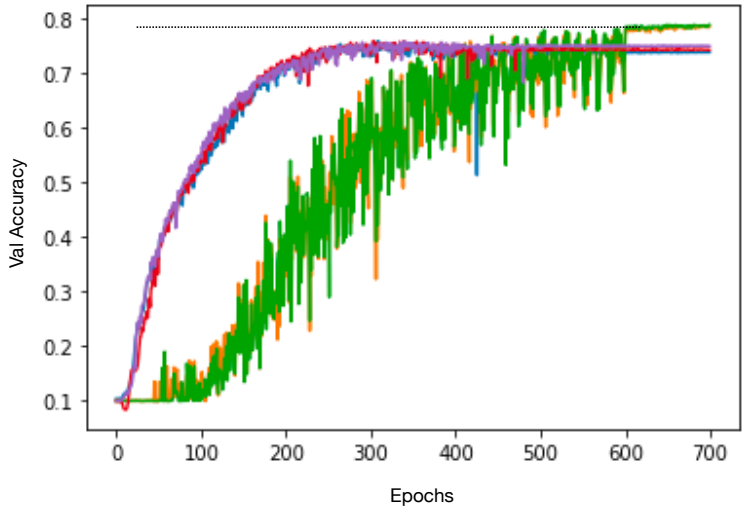

FIG. 3. Validation accuracy during the training of Myrtle5 on CIFAR10, as shown in the phase diagrams of Fig. 1 of the main text. The lines that rise very fast followed by a moderate over-fitting correspond to learning without oscillations. The other group of lines correspond to learning in the region $A \sim 50$ and $T \sim 100$. We stopped the oscillations after epoch 600. We have included a black dotted line to guide the eye, the final accuracy of the simulations with the oscillations is higher than any point belonging to the simulations without oscillations.

Fig. 1 of the main text showed that using the standard cross entropy loss function without the oscillations ($\Gamma_i = 1$, $A = 1$ line in both panels) the system was able to fit all the training data (training accuracy $\sim$ 1) and achieved a $\sim$ 0.73 validation accuracy. However, the validation accuracy improved up to 6% thanks to the oscillations for $A \sim 50$ and $T \sim 100$. For all A and T the oscillations stopped at epoch 600 (we changed $A = 1$ at that point for all the simulations). Even when the learning rate is decreasing in the second part of each simulation, the oscillations did not reduce appreciably in size at the end of training. In the last 100 epochs of training, the learning rate is already close to 0.002, but making $A = 1$ (removing the oscillations) helped the system to stabilize and increase the validation accuracy. We have included here Fig. 4 where we plot two groups of simulations. One group corresponds to learning without oscillations and the other one corresponds to the point in the (T, A) region where validation accuracy improved the most. Learning with oscillations is slower in this case but it reaches a higher validation accuracy. We have included a black

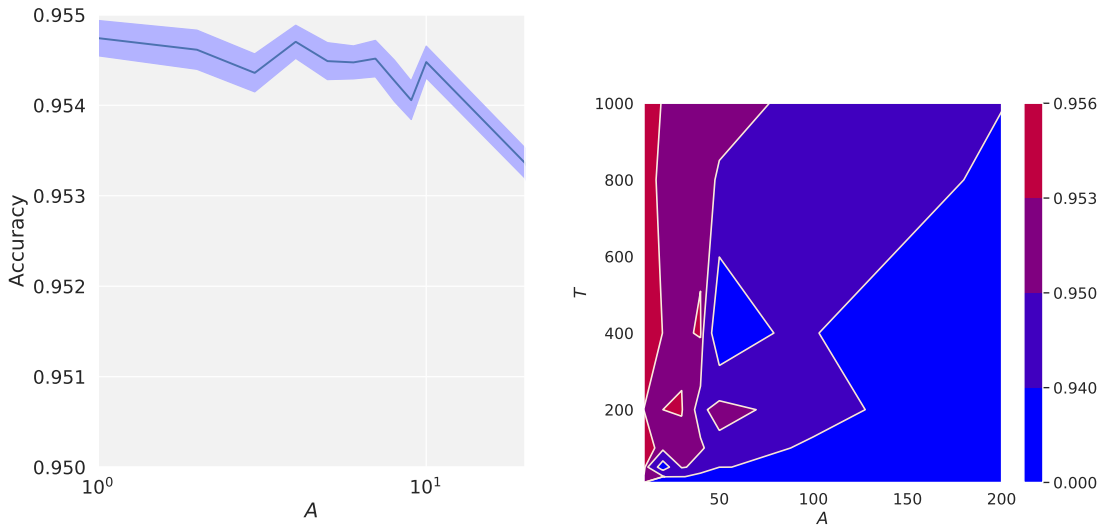

FIG. 4. Test accuracy of a Wide Residual Network. Left: The accuracy as a function of oscillation amplitude over a one-dimensional slice with $T = 200$ averaged over 50 seeds. The shaded region shows one standard deviation of the average test accuracy after training. Right: A two dimensional phase plot showing test accuracy as a function of both period and amplitude.

dotted line to guide the eye. We leave a systematic study for future work, but at least in this case, stopping the oscillations in the last part of learning had a positive effect, helping the model to achieve a higher validation accuracy.

IV. WIDE RESIDUAL NETWORKS

In addition to the Myrtle5 architecture, we also ran a number of experiments on Wide Residual Networks [4]. We used a standard 28-10 residual network with batch normalization and ReLU activation functions. We trained for 200 epochs using a batch size of 1024 running on a 2x2 TPUv2 with a batch size of 128 per chip. We used cosine learning rate decay with an initial learning rate of 0.1 along with the momentum optimizer. Finally we used augmented the data using random flips and crops. This model gets slightly lower accuracy than the version in the literature (95.5% vs 95.8%) due to the larger batch size employed here. For this architecture we do not observe a statistically significant improvement to the test performance by using an oscillatory loss.

[1] B. Ghorbani, S. Krishnan, and Y. Xiao, arXiv preprint arXiv:1901.10159 (2019).

[2] J. Gilmer, "Large scale spectral density estimation for deep neural networks," https://github.com/google/spectral-density (2020).

[3] V. Shankar, A. Fang, W. Guo, S. Fridovich-Keil, J. Ragan-Kelley, L. Schmidt, and B. Recht, in *International Conference on Machine Learning* (PMLR, 2020) pp. 8614–8623.

[4] S. Zagoruyko and N. Komodakis, CoRR **abs/1605.07146** (2016), arXiv:1605.07146.