Research Proposal:
Does Grokking Happen in the Global Valley?

Matt Sárdi

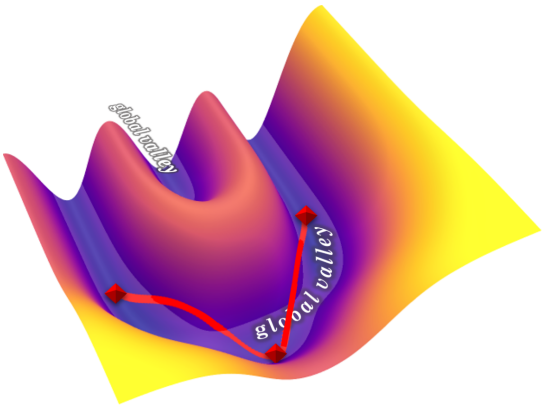
Abstract

I propose some empirical experiments about the theoretical properties of deep neural network loss surfaces, after some very recently discovered phenomena (grokking) have not yet been thoroughly studied in this context.

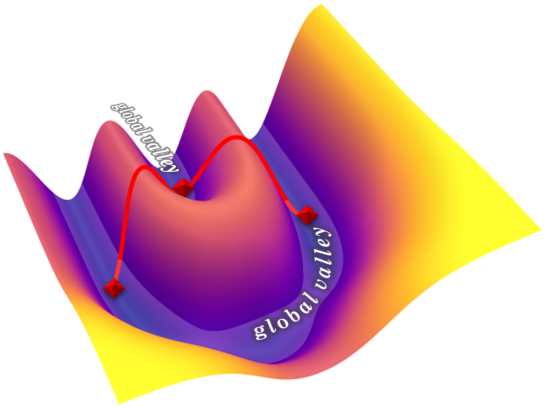
My hypothesis is that *grokking* (Power et al. 2021, unusually long training that suddenly improves generalization) happens during a walk along the *global valley* (Nguyen 2019, the continuous region of the loss surface that contains most local minima). The experiment (although this may be adjusted during research) is to periodically sample optimal points during the grokking phase of the training and check if there's *linear mode connectivity* between successive ones. If there is, the hypothesis is true. Another way to say that, if the resulting plot is wavy, that means the optimization process "jumps", or climbs up and down valleys between minima, but if it's smooth and stays small, then it "walks" along the bottom of the global valley.

1 Introduction

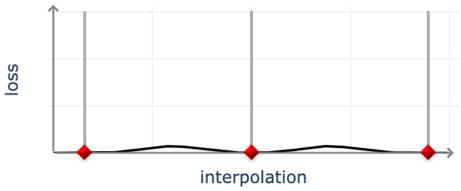
Studying the loss surfaces of deep neural networks seems to be a useful way to gain various insights into how these models work. Although the effectiveness of deep learning is empirically well



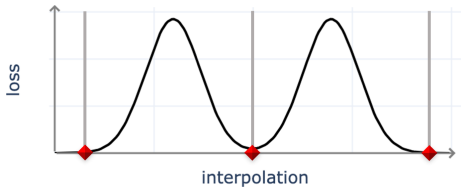
(a) Illustration where the hypothesis is true.



(b) Illustration where the hypothesis is false.



(c) Linear mode connectivity graph for 1a. There **is** linear mode connectivity, so the hypothesis is true.



(d) Linear mode connectivity graph for 1b. There is **no** linear mode connectivity.

Figure 1: Intuitive visualization of the proposed experiment. We sample successive points during grokking and check if there's linear mode connectivity between them. (We essentially plot linear slices of the landscape.) The example loss landscape is greatly simplified to illustrate the point.

validated, i.e. we know it works in practice, interestingly, a big part of our theoretical understanding of it is still missing as of 2021 (Ma et al., 2020).

Even though it's a commonplace that (stochastic) gradient descent can be intuitively visualized as traveling down on a mountainy landscape (the loss surface), always heading downwards, hoping to find the lowest point, regularly, newer and newer discoveries are being made (every 2 years or so), about real DNN loss surfaces behaving in weird and unexpected ways (Zhang et al. 2016, Frankle and Carbin 2018, Garipov et al. 2018, Power et al. 2021) that initially seem counter-intuitive for the landscape metaphor, probably in part because these surfaces live in million-or-so dimensional spaces, which behave differently than our well known 2 or 3 dimensions, and in part because of some stochastic properties of SGD. This research proposal aims to continue some of the most recent such findings of NN loss surfaces, (more or less picking up where they left off with the "future work" section), and proposes performing some simple, clear, and hopefully insightful experiments.

Grokking (Power et al., 2021) is a newfound phenomenon, wherein training DNNs for an unusually long time prompts them to improve generalization. Weight decay is assumed to take an important part in it.

Neural networks trained on complex data have a lot of local minima. Previously, it was thought not to be a problem because these minima are almost equally good. More recently, it was discovered (Garipov et al. 2018, Draxler et al. 2018) that these minima are not isolated, but connected, sort of forming a high-dimensional valley. My intuition of grokking is that during training, the loss goes directly to the closest minimum, straight down the wall of the valley, but grokking happens as a slow walk along the bottom of the valley.

The experiment is to measure linear mode connectivity between the points during grokking. Even though it may seem obvious this is the case, as we've seen, neural networks have a track record of defying what seems intuitive, so I'd argue whatever comes out of this experiment would be a valuable contribution to helping understanding deep learning for fellow researchers.

2 Background, related work

In this section, well-known concepts will be described with short definitions. More peculiar research will be explained longer.

DNNs or deep neural networks are a class of machine learning algorithms that are perhaps the most used in practice as of 2021. In this discussion, it's enough to consider them as black-box algorithms. They behave like functions, so for a given input, they provide an output (prediction), and they have a certain number of real-valued parameters (adjustable "knobs" on the box), called weights, which control the function somehow. In the supervised learning problem setting, we are provided with training examples, which are pairs of (presumedly correct) inputs (features) and corresponding outputs (labels) (e.g. a photo of a dog, and the label "dog"). I will use parameters and weights interchangeably. Some research emphasizes that these models are overparametrized, meaning they have much more parameters than the number of training examples, and they could "memorize" all training examples (however, in practice, these models generalize well).
Loss (or, using a more general term, objective function) is a single-value metric of how badly our model performs. It's usually some "distance" between the algorithm's output and the true label, averaged over the entire training set. The task of the learning problem is to "train" the model, in other words, to adjust the weights in a way that minimizes the loss (thus meaning the model's output is as close to the correct output as it can be for every example).
Error is similar to loss, but there can be multiple error metrics, and, in contrast to the loss, it's not necessarily related to the training process. Accuracy is an error metric for classification tasks, which measures the fraction of correct predictions.
Generalization is whether the algorithm works well for previously unseen examples, too. A common way to assess generalization is to split the training data, usually into 2 (or 3) sets, training, and test (or training, validation, and test) sets, then training on the training set and testing the performance on the test set. Generalization error is the difference between the training error and the test error (the errors measured on respective sets). Overfitting is when a model

learns the training examples perfectly, but fails to generalize.
Regularization is usually adding extra terms to the loss function to penalize undesirable solutions, and prevent overfitting. ℓ_2-loss, or weight decay is a regularization that penalizes large weights, and thus in a sense, prefers simple solutions.

Loss surface or loss landscape of a deep neural network is a high-dimensional plot. Each point corresponds to a configuration of the neural network weights, and the value (the altitude) is the training error produced with those weights. In a sense, (in the supervised learning setting), a loss surface encodes the whole training set, it outputs what the training error would be on the whole training set with the given parameters. It's also affected by the type of loss used, and a potential regularization. In illustrations, we plot them in 2 or 3 dimensions, but this simplification hides many properties of them, like the fact a point has many neighbors in different directions. Since it is "loss", the badness of mistakes the algorithm makes, the lower the better. Many counter-intuitive phenomena were discovered about loss landscapes, and more generally DNNs. Some examples are: DNNs can easily fit random data (Zhang et al., 2016), generalization gap and sharp minima (Keskar et al., 2017), sharp minima can generalize (Dinh et al., 2017). the lottery ticket hypothesis (Frankle and Carbin, 2018), double descent (Belkin et al., 2018), SGD can reach bad minima (Liu et al., 2019), kernel and rich regimes (Woodworth et al., 2020).

GD and SGD. DNNs are usually trained with some type of gradient descent (GD). The goal is to make small adjustments to the weights that each reduce the loss. After a number of adjustments, a low loss can be achieved. The process can be conceived as taking small steps downward on the loss surface. Our current "position" is the current configuration of the weights. Evaluating the loss for the full training set is computationally expensive, so most practical applications use stochastic gradient descent (SGD), which takes a small random so-called mini-batch of training data, and calculates the gradient on those. The SGD step can be thought of as a noisy estimate of the (full-batch) gradient descent (GD) step, and this noise can even provide some beneficial properties, like preferring flat minima over sharp ones (Keskar et al., 2017). Flat minima are usually found to generalize better (on the test set). Adaptive step-size algorithms, which can consider scale and momentum information (intuitively, taking steps vs. rolling a ball downhill), like Adam (Kingma and Ba, 2014) are also commonly used.

Mode connectivity can refer to both checking whether this property holds, or the bigger discovery that in real DNN loss surfaces this property in fact almost always holds among minima. Modes are optimal points of the loss surface. (In the usual statistical sense, modes are "bumps" on graphs, but here, since we're talking about loss, and lower is better, modes are actually "holes".) Two modes are mode connected if there exists a path between them along which the loss and error are almost constant (and since modes have low loss, it also stays low along the path). We essentially interpolate the weights along the path and check the training loss at each point along the path. Such mode connectivity plots between two points and along a given path can be plotted, with the value of the plot being the training error. If the plot is a flat line, there is mode connectivity. If it is a bump, there isn't. The plots in fig. 1 are actually multiple mode connectivity charts placed next to each other or "stitched together", where the ending point of the first one is the starting point of the next one. (Or equivalently, it can be thought of as a mode connectivity plot along a polygonal path, which contains multiple line segments.) Previously it was commonly thought that DNN loss landscapes have many isolated (almost equally good) minima. Garipov et al. (2018) and Draxler et al. (2018) simultaneously discovered that these modes are connected, and provided procedures to find such a path, similar to gradient descent. Garipov et al. (2018) used VGG (Simonyan and Zisserman, 2014) and ResNet (He et al., 2016) models trained on CIFAR-10, CIFAR-100, and ImageNet. They found that usually, they could find a path with just two line segments, so a path composed of two straight lines. They also took this idea further and, leveraging their findings, invented an optimal ensembling algorithm (Fast Geometric Ensembling, FGE). They train one network, then use a cyclic learning rate and really few steps to sample "adjacent" points. They average the optima along such a curve and achieve results better than traditional training.
Linear mode connectivity is a special case of mode connectivity where the path is a straight line. So it holds when two modes' weights can be linearly interpolated, without the error going up along this path. Linear mode connectivity is sometimes used to assess the similarity of minima (configuration), with mode-connected ones being "essentially the same", or "really similar" minima, like in Frankle et al. (2020). Since almost every two points are mode-connected (along some path), measuring general mode connectivity is not useful in terms of this research, and we will always measure *linear* mode connectivity.

Global valley was used in Nguyen (2019), where he studied mode connectivity. It is possible that in that paper it is given a slightly different definition, but I will just use it as a convenient simple term to refer to the large mode connected low-loss area in DNN loss surfaces that Garipov et al. (2018) and Draxler et al. (2018) discovered and Garipov et al. (2018) further suggested in that paper in sec. 7, referred to as simply "valleys of low loss". In fig. 1a and 1b, both valleys are labeled as global valley, because these are connected, there is a single continuous global valley.

Grokking (Power et al., 2021) is the phenomenon where unusually long training suddenly improves generalization. So train a network up to overfitting perfectly but keep training, for orders of magnitudes more epochs. As of 2021, it has only been demonstrated on small synthetic datasets of operation tables. Regularization is assumed to play an important role in it, they used ℓ_2 regularization.

3 Problem definition

As the title and the introduction already said, the question is whether grokking follows a path along the bottom of the global valley.

In the grokking process, the training loss minimizes quickly, but after a long training, the test loss (the generalization error) also drops. It seems apparent that the point travels, possibly a long distance in this long time. When this phenomenon is viewed through the lens of the landscape metaphor, it seems plausible that after initially finding a (local) minimum (rushing straight down the wall of the valley), the point of optimization moves slowly along the bottom of the global valley, or equivalently, along a path that yields mode connectivity, before it finds an even "better" optimum (probably better because this one also minimizes the ℓ_2 regularization loss).

The goal is to perform an experiment that can quantitatively assess if this is the case, if grokking follows this path. To the best of my knowledge, this experiment hasn't been performed and published yet. Learning the result could give us new insights into the geometry of loss surfaces. Confirming the hypothesis would be valuable, but disproving it would also surface some new phenomena. It could establish a link between grokking and mode connectivity. There are also many interesting small results that could be discovered during these experiments, just as some off-the-cuff examples, the number of such paths between two points could affect the experiment, and it could prompt follow-up questions to measure this property, or "noisiness" of SGD could affect the experiment, and it could prompt measuring how much the noisiness of SGD prevents it from following optimal paths on the landscape, and how learning rate or bach size effects it, etc.).

One way to measure certain properties of the path of optimization, or how "similar" two modes are is through measuring linear mode connectivity. Of course, there could be more than one way to measure properties of the path of optimization, and I may resort to those, but linear mode connectivity seems a sensible first guess. We start a training, wait for the grokking phase, sample points during grokking, and see if there's linear mode connectivity between successive ones, meaning if we directly linearly interpolate the points, the loss doesn't go up much along this path.

More details about performing the experiment can be found in section 4.3.2.

4 Research plan

4.1 Objectives, deliverables

Resources. A server with GPU, or Google Colab.

Objective. Set up and perform the experiment described in section 3.

Deliverables. Experiment results, paper.

Possible outcomes.
The following list describes a few possible outcomes of the experiment, divided into larger categories, and further divided into individual cases.

- Question is not applicable, need to change the original plan
  - Upon more literature research, it's revealed that the question is fundamentally ill-posed. As I did some literature research, I wouldn't consider it likely, but it's still a possibility.
  - Upon more research, it turns out the research question was trivially simple because of some theorem or previous result or is already answered in other research exactly.
  - Negative result, but it gets shown that SGD cannot follow mode-connected paths, grokking or otherwise, anytime.
  - Measuring linear mode connectivity turns out to be not the proper way to assess whether the training follows a path. In that case, a different measurement method needs to be found, most probably alternatives exist.
- Accept hypothesis
  - Experiment runs as expected, results show linear mode connectivity.
- Reject hypothesis
  - Experiment runs as expected, results show no linear mode connectivity.
- Borderline
  - The mode connected loss between sampled points is neither particularly high nor particularly low. So we cannot clearly tell whether there's linear mode connectivity. In this case, we would need to draw inspiration from other linear mode connectivity research about how they define the thresholds. Further examining the trajectory that SGD makes can also be helpful in this case as it directly explains the result.

4.2 Timeline

Task description	2022			2023										2024									
	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10
Preparations																									
Kickoff																									
Literature review																									
Planning																									
Reproduce the grokking result																									
Set up experiment framework																									
Sanity check of the framework																									
Experiments																									
Planning																									
Perform experiments																									
Draw and formulate conclusions																									
Address possible follow-up questions																									
Ablation studies																									
Write-up																									
Planning																									
Compose the main body of text																									
Edit figures																									
Review, proofread																									
ICLR submission period?																									

Table 1: Proposed timeline.

4.3 Tasks

4.3.1 Preparations

Kickoff
A meeting to clarify the goals and success criteria, and to outline high-level tasks.

Literature review
Of course, careful literature review could save us from some dead-ends and futile efforts. Ample time should be given to review the relevant literature. Search for everything on grokking, and try to find experiments with linear mode connectivity, both papers and source code. Litmaps (https://www.litmaps.co) is a recent tool that seems particularly handy for this task.

Reproduce the grokking result
Reproduce the main result of the grokking paper (Power et al., 2021). To my experience, reproducing a result from a paper can take up to 2 months, as a pessimistic estimate. Even more so if there's no code available. As of writing this in 2021, there exists a GitHub repo for the paper. This task should probably be timeboxed, if a better goal can't be achieved within the timeframe, we settle with a worse goal. It's not required to reproduce the exact result, only the generic phenomenon. In the worst case, if we're far from reproducing grokking, it's enough if we can argue for why our results would be relevant with "our version" of grokking.

Set up experiment framework
The idea is to create the "measurement tool" first and perform the experiment afterwards when it's done and can be trusted. Create a git repo, agree on coding conventions. Agree about whether to use testing. Agree in using an experiment tracker, like wandb (https://wandb.ai). The framework should be capable to do trainings, and draw (or otherwise output) mode connectivity graphs.

Sanity check of the framework
Run an experiment that is already known to have linear mode connectivity and one which is known not to. Check if the framework outputs the expected results.

4.3.2 Experiments

Perform experiments
Perform the experiment with the following 2 models: the original operation tables from the grokking paper, and a small real-world computer vision model (probably a VGG-16 or ResNet-18) trained on CIFAR-10. When writing this research proposal, a lot of details are not known about the feasibility of the second experiment (like whether VGG on CIFAR-10 would exhibit grokking in the first place, and what are the computational requirements for it), so only the first is mandatory, the second is optional. Start the training and wait for the grokking phase. In the grokking phase (after overfitting and before generalization), save the weights of the network as checkpoints, every n gradient descent step. A full epoch would be probably too long. (The process is described here for simplicity, but these are also features of the experiment framework and should be already implemented at this point.) If that would yield too many samples, it's enough samples for a shorter duration. After the training is done, check if grokking indeed happened, or stop the training as soon as grokking happens. Create mode connectivity plots for 10 or so subsequent checkpoints. Determine, based on the plot, if there is mode connectivity. Comparing the increase in loss to a previously agreed threshold. If we don't see mode connectivity, the experiment can be repeated with lower n. In theory, if everything goes perfectly, only this single experiment execution and this single result is needed, but of course, in practice, probably multiple tries and adjustments will be needed, doing changes as we go that we see fit.

Draw and formulate conclusions
Draw the first round of conclusions from the results. Write notes (we're not yet writing paper, but some drafts would be useful). At this point, of course, we don't know what we will see. It's possible that we discover new phenomena.

Address possible follow-up questions
As things will most likely not go exactly as planned (in terms of the results), we will most likely have follow-up questions. I leave ample time for these. Formulate follow-up questions, and perform follow-up experiments.

Ablation studies
Remove certain parts and components of the experiment to see what contributes to the results. In this experiment, I expect that both the model and the experiment can be ablated in some way. It could also take some time to address and perform every important ablation. This task is time-boxed, anything we don't have time for goes to "future work".

4.3.3 Write-up

Compose the main body of text
At this point, the findings should be already formulated. Compose the main text, aiming at about 8 pages (ICLR recommendation, max. is 10), with figures. Agree on how many appendices to use. Add tables with the result, if applicable.

Edit figures
Create 2-3 figures. The "Figure 1" should be a cartoony overview of the whole experiment, visualizing everything on a high level. In addition, probably a grokking plot, and a mode connectivity plot will be needed, times the different models we ended up testing (operation tables, VGG). Ablations, if any, probably also require basic plots.

Release code
Decide whether to release the code publicly. A script that reproduces the numbers from the paper (without the whole experimenting framework) should be sufficient. If so, finalize and release the code on GitHub.

4.4 Impact

In the short term, these kinds of experiments and research questions seem to be among those types which get the most attention at conferences, as neural network loss surfaces and optimization behavior seems somewhat mysterious, with a lot of unsolved problems and unexpected discoveries.

On the other hand, to my understanding, this is an experiment that someone, somewhen eventually will perform (or the experiment doesn't work, in which case probably no one will publish the negative results). It's a piece of the puzzle, and providing this result would contribute to the common understanding of deep neural networks and contribute to the machine learning community. If the outcome is not what is expected, that is even more exciting, because that would mean the discovery of yet another counter-intuitive phenomenon.

In the long term, this result (in particular depending on the outcome) can potentially affect practical applications, such as better optimization algorithms (faster or more memory-efficient), new ensembling methods, helping the study of generalization and systematic generalization, and improving loss landscape visualization. A general better understanding of deep learning could lead to many advances.

References

Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2018). Reconciling modern machine learning practice and the bias-variance trade-off. *arXiv preprint arXiv:1812.11118*.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. (2017). Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pages 1019–1028. PMLR.

Draxler, F., Veschgini, K., Salmhofer, M., and Hamprecht, F. (2018). Essentially no barriers in neural network energy landscape. In *International conference on machine learning*, pages 1309–1318. PMLR.

Frankle, J. and Carbin, M. (2018). The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*.

Frankle, J., Dziugaite, G. K., Roy, D., and Carbin, M. (2020). Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pages 3259–3269. PMLR.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D., and Wilson, A. G. (2018). Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 8803–8812.

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

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. (2017). On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*.

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

Liu, S., Papailiopoulos, D., and Achlioptas, D. (2019). Bad global minima exist and sgd can reach them. *arXiv preprint arXiv:1906.02613*.

Ma, C., Wojtowytsch, S., Wu, L., et al. (2020). Towards a mathematical understanding of neural network-based machine learning: what we know and what we don't. *arXiv preprint arXiv:2009.10713*.

Nguyen, Q. (2019). On connected sublevel sets in deep learning. In *International Conference on Machine Learning*, pages 4790–4799. PMLR.

Power, A., Burda, Y., Edwards, H., Babuschkin, I., and Misra, V. (2021). Grokking: Generalization beyond overfitting on small algorithmic datasets. In *ICLR MATH-AI Workshop*.

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

Woodworth, B., Gunasekar, S., Lee, J. D., Moroshko, E., Savarese, P., Golan, I., Soudry, D., and Srebro, N. (2020). Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, pages 3635–3673. PMLR.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. arxiv preprint (in iclr 2017). *arXiv preprint arXiv:1611.03530*.