Phase transitions and structure formation in learning local rules

Bojan Žunkovič[*]
Faculty of computer and information science
University of ljubljana
Večna pot 113, Ljubljana, Slovenia
bojan.zunkovic@fri.uni-lj.si

Enej Ilievski
Faculty of mathematics and physics
University of ljubljana
Jadranska ulica 19, Ljubljana, Slovenia
enej.ilievski@fmf.uni-lj.si

Abstract

We study a teacher-student rule learning scenario, where the teacher is determined by a local rule and the student model is a uniform tensor-network attention model. The student model also implements a map from variable-size binary inputs to the latent space $\mathcal{V} = \mathbb{R}^d$, where d is the bond dimension of the student model. Using gradient descent learning we find a second-order phase transition in the test error. At the transition we observe a sudden drop in the effective dimension of the mapped training data. We also find that small-effective dimension corresponds to structure formation in the latent space $\mathcal{V}$.

1 Introduction

We can improve generalisation properties of deep neural networks by increasing the number of parameters (the double descend phenomenon [1, 2, 3, 4]) and by training past zero train error [5, 6]. Recently two empirical observations have been made in the terminal phase of training, i.e. when training past zero train error. Namely, neural collapse [7] and grokking (generalisation beyond over-fitting) [8].

Neural collapse refers to the collapse of the $N-$dimensional, last-layer features (input to the last/classification layer) [7] to a $(C-1)$-dimensional equiangular tight frame (ETF) structure, where C is the number of classes. We can partially understand neural collapse within the unconstrained features and local elasticity models [9]. However, its role in generalisation, relation to grokking, and appearance of different latent space structures are still not completely understood.

Grokking also occurs in the terminal phase of training and refers to a sudden decrease of the test error from approximately one to zero [8]. This transition has been discussed within an effective theory approach [10], where an empirical connection between representation/structure formation and generalisation has been made. A related empirical study [11] established a relation between grokking and training loss spikes and weight norm increase. However, it is unclear how to reconcile grokking with the standard generalisation theory based on statistical methods [12]. Finally, it is not yet clear what is the minimal framework within which we can understand these phenomena. This paper tries to fill this gap by proposing a novel teacher-student learning setting that features grokking and structure formation [2].

[*] https://qmltn.ai/

[2] https://github.com/qml-tn/grokking/

Machine Learning and the Physical Sciences workshop, NeurIPS 2022.

2 Learning a local rule

In the standard statistical-learning scenario, we determine the the rule by a simple one-layer teacher model

$$f(x) = \mathrm{sgn}(x \cdot w + b), \tag{1}$$

where x is the input vector, w is the weight vector, and b is the bias– $x, w \in \mathbb{R}^N$, and $b \in \mathbb{R}$. To facilitate analytical calculations generalisation properties of the model are discussed in the thermodynamics limit, i.e. $N \to \infty$, with appropriate scaling of the input and weight vectors with N. In this case, all values of the input contribute to the final result, which leads to a mean-field like behaviour, i.e. the value of the input at any particular position has only infinitesimal influence on the result/rule.

We will study the opposite, local scenario $x \to y$, where $x, y \in \{-1, 1\}^M$. The i-th component of the output vector, i.e. y_i, will depend only on a K-neighborhood of the input x at position i,

$$y_i = \mathrm{rule}(x_{i-K}, \ldots, x_i, \ldots x_{i+K}). \tag{2}$$

We call such model a $K-$local model. The Eq. 2 describes a well-known cellular automata computational paradigm. Cellular automata are a universal discrete space-time dynamical systems with a finite set of possible states at each position [13, 14]. We define a cellular automaton by a set of rules which transform one configuration of states into another configuration. We will consider the rule 30 one-dimensional automaton ($K = 1$) [13, 14], which exhibits chaotic behaviour and is defined by the rule $y_i = \mathrm{rule30}(x_{i-1}, x_i, x_{i+1})$. The next state of the cell i, i.e. y_i, is determined by the current configuration at cells $i-1$, i, and $i+1$, i.e. x_{i-1}, x_i, x_{i+1}, as follows

$\mathbf{x_{i-1}, x_i, x_{i+1}}$	-1,-1,-1	-1,-1,1	-1,1,-1	-1,1,1	1,-1,-1	1,-1,1	1,1,-1	1,1,1
$\mathbf{y_i} = \mathrm{rule30}(\mathbf{x_{i-1}, x_i, x_{i+1}})$	-1	1	1	1	1	-1	-1	-1

. (3)

The rule-30 automaton has already been discussed in the context of sequence-to-sequence prediction with tensor networks [15, 16, 17], however, no grokking phenomena have been reported. To study the effect of the neighbourhood size K, we shall consider a rule defined by K consecutive applications of rule 30. We will refer to such rule as a K–local rules.

In summary, we modify the standard perceptron teacher-student setup by restricting the teacher model to local instead of global rules. The teacher will be modelled by a local map transforming a sequence x into the sequence y. The task will be to approximate the chosen map by training on a finite set of input samples of length M. For a finite M we will choose open boundary conditions with $x_0 = x_{M+1} = -1$. The test set will include all possible inputs with sizes $M_{\mathrm{test}} = 3, 4, \ldots, M_{\max} = 1000$. We will determine the error as the ratio of incorrectly predicted values y_i.

3 Tensor-network attention model

In this section, we will introduce a simplified version of the tensor network proposed in [17], which will serve as a student model. The tenor-network model has two parts: an embedding layer and a tensor-network attention layer. We define the embedding layer with a local embedding function $\phi(x_i) : \{-1, 1\} \to \mathbb{R}^2$ as

$$\phi(-1) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \phi(1) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{4}$$

After the embedding, we apply the tensor-network attention determined by an attention tensor A and a classification tensor B. First, we construct matrices $\mathcal{A}(i)$ by contracting the attention tensor A with the local embedding vectors $\phi(x_i)$

$$\mathcal{A}_{\mu,\nu}(i) = \sum_{j=1}^{2} A_{\mu,\nu,j} \phi(x_i)_j. \tag{5}$$

Then, we use the matrices $\mathcal{A}(i)$ to construct the left and right context matrices $H^{\mathrm{L,R}}(i)$,

$$H^{\mathrm{L}}(1) = \mathbb{1}_d, \qquad H^{\mathrm{L}}(i) = H^{\mathrm{L}}(i-1)\mathcal{A}(i-1), \tag{6}$$

$$H^{\mathrm{R}}(M) = G, \qquad H^{\mathrm{R}}(i) = \mathcal{A}(i+1)H^{\mathrm{R}}(i+1). \tag{7}$$

The matrix G determines the boundary conditions and is set to $G = v^{\mathrm{L}} \otimes v^{\mathrm{R}}$. The boundary vectors $v^{\mathrm{L,R}} \in \mathbb{R}^d$ are determined as left and right eigenvectors of the matrix A_0 corresponding to the largest eigenvalue. We obtain

the final local weight vector $w(i)$ by contracting the tensor B with the normalised left and right context matrices $H_{\mathrm{N}}^{\mathrm{L,R}} = H^{\mathrm{L,R}}/||H^{\mathrm{L,R}}||_2$

$$w(i)_j = \mathrm{Tr}\left(H_{\mathrm{N}}^{\mathrm{L}}(i) B_j H_{\mathrm{N}}^{\mathrm{R}}(i)\right), \quad j = 1, 2, \quad i = 1, \ldots, M, \tag{8}$$

where B_j denotes the matrix with elements $[B_j]_{\mu,\nu} = B_{\mu,\nu,j}$. We calculate the attention layer output at position i as

$$\hat{y}_i = w(i) \cdot \phi(x_i). \tag{9}$$

We calculate the final model output by using the sign nonlinearity $f(x) = \mathrm{sgn}(\hat{y})$. The described tensor-network layer is a generalisation of the linear-dot attention mechanism (see [17]). Therefore, we refer to it as a tensor-network attention.

Tensor network map The described tensor-network attention model also implements a map from inputs of variable length M to vectors of length $z_i(x) \in \mathcal{V} = \mathbb{R}^{2d^2}$, where

$$z_i(x) = H_{\mathrm{N}}^{\mathrm{R}}(i) H_{\mathrm{N}}^{\mathrm{L}}(i) \otimes \phi(x_i). \tag{10}$$

By considering $z_i(x)$ as inputs we interpret the model defined by Eq. 9 as a perceptron, namely

$$\hat{y} = z_i(x) \cdot \vec{B}, \tag{11}$$

where $\vec{B}$ denotes the vectorised classification tensor B. Interestingly, for any $K-$local rule, we can find 4^K-dimensional matrices A for which the transformed problem is solvable by a simple perceptron model and exhibits the grokking phenomena. Therefore, the standard $1/\alpha$ dependence on the training set size (see [12]) seems to be a consequence of the infinite-range rule. For any local rule, we will observe grokking.

4 Results

In all experiments we initialise the model with a random initial condition, where all the entries of the tensors A, B are uncorrelated and sampled according to a normal distribution with zero mean and unit variance. We train the model with the Adam optimiser (with standard parameter setting) and learning rate 0.005. We use the mean squared error loss with $L_{1,2}$ regularisation strength $\lambda_{1,2} \in [0, 0.001]$, which is the same for the attention tensor A and the classifier tensor B. We also use the sigmoid non-linearity instead of the sign non-linearity to improve the training stability and reduce the training time. We perform tests in three situations, namely, without regularisation (`Example 1`: $\lambda_{1,2} = 0$), with L_2 regularisation (`Example 2`: $\lambda_1 = 0$, $\lambda_2 = 0.0001$), and with L_1 regularisation (`Example 3`: $\lambda_1 = 0$, $\lambda_2 = 0.001$). We chose the regularisation strengths $\lambda_{1,2}$ to be the largest regularisation strengths with only few spikes in the terminal phase of training. Unless specified otherwise we use bond dimension $d = 40$.

Average test error and average effective dimension First, we investigate the dynamics of the average test error and calculate the critical exponent ν (see Fig. 1). The test error drops to zero at the grokking transition, i.e. at time t_ϵ. Following the grokking transition, the test error is non-zero and experiences fluctuations. These fluctuations can be detected as sharp increases in the training loss and are more common in models with large regularisation (see also Fig. 2). Therefore, the $L_{1,2}$ regularised models have larger average test error after the grokking transition. We also observe that the critical exponent decreases upon increasing regularisation. Larger regularisation leads to a sharper transition to zero test error. We also observe the effective

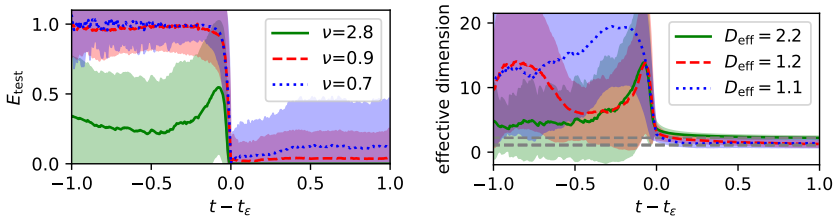

Figure 1: The average test error at the phase transition. We align the first point where the test error becomes zero (i.e. the time t_ϵ) and take the average over 1000 initialisations of the model parameters. The colors correspond to `Example 1` (green), `Example 2` (red), and `Example 3` (blue). Shaded regions show the standard deviation. Critical exponents and D_{eff} are reported in the legends of the figures.

dimension at the grokking transition. We calculate the effective dimension D_{eff} as the exponent of the entropy $S = -\sum_k \sigma_k \log \sigma_k$, where σ_k denote the fraction of the variance explained by the kth principal component of the training dataset features $z_i(x)$. As shown in Fig. 1, the average effective dimension drops significantly just before the grokking transition. We observe that regularisation decreases the effective dimension of the mapped vectors $z_i(x)$.

Structure formation and grokking We show in Fig. 2 that small effective dimension signals an emergent latent space structure which, however, can be different in each example. Hence, we argue that the sharp decrease in the effective dimension is a consequence of structure formation. Grokking and structure formation are therefore related on average as shown in Fig. 1 but not for every trained model individually (as argued in [10]). We disentangle model-wise structure formation from model-wise grokking by observing specific training examples. In Fig. 2 we show the structures appearing in the features $v^{\text{L}} H^{\text{L}}_{\text{n}}(i)$ with bond dimension $d = 3$. We observe that the structure of the latent space data changes also during a single run. This can be detected as a spike in the training loss or as a step-like jump in the effective dimension. The structures can change from lower to higher dimensional and vice versa. Finally, we also show that we can have a small generalisation/test error with complex or non apparent latent space structures (the structure marked by $\times$ in Fig. 2). These empirical observations suggest that grokking and structure formation are not related model wise.

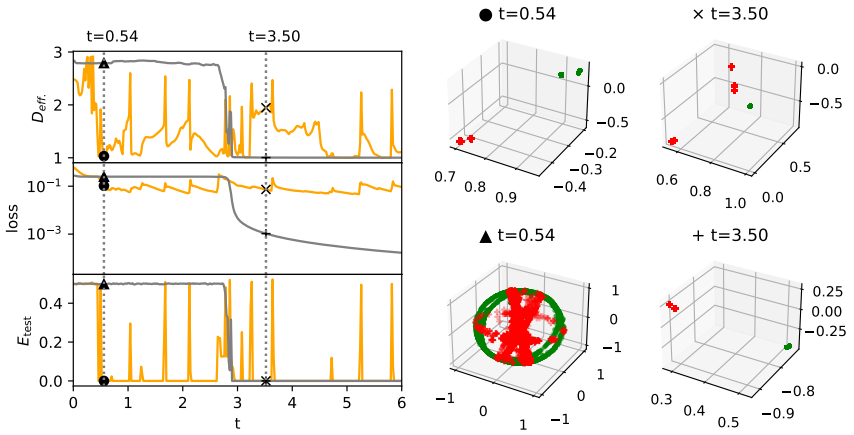

Figure 2: Several emergent structures in the latent space $\mathcal{V}$. The left plots show the effective dimension D_{eff} (top), train loss (middle), and test error (bottom). The gray line corresponds to training without regularisation and the orange line to training with $\lambda_1 = 0.01$. The black markers show the value of the plotted quantities at specific times marked by vertical dotted lines. The right panels show the structure of the features at the marked times: top row shows the L_1 regularised case, bottom row shows the non-regularised case.

Grokking time We define the *grokking time* as the difference between t_ϵ (zero-test-error time) and the time at which the training error becomes zero. We observe the grokking time in the `Examples 1-3` discussed above (see Fig. 3 left panel). Taking the non-regularised case (`Example 1`) as the baseline, we find that regularisation (`Example 3`) decreases the average grokking time $\overline{t_{\text{G}}}$ significantly more than L_2 regularisation (`Example 2`). Since the grokking time is measured relative to the time at which the zero train error is achieved, we estimate also the density of t_ϵ (see Fig. 3 right panel). We find that both L_1 and L_2 reduce t_ϵ. Therefore, both, the L_1 and the L_2 regularisation decrease the number of steps required for good generalisation. In addition, the L_1 generalisation seems to be more efficient, in the sense, that there is a shorter time interval with a large difference between train and test error.

5 Discussion and summary

While various concepts and tools from physics such as entropy, replica trick, tensor networks, effective theory, have proven useful in machine learning generalisation theory, the notion of locality has, to our knowledge, not been discussed thus far. By using a novel tensor-network attention model (likewise inspired by physics) we show that locality plays an important role when learning a rule. We show that local rules lead in general to second-order phase transitions. We numerically calculate the critical exponent and show that the phase transition is also related to structure formation in latent space.

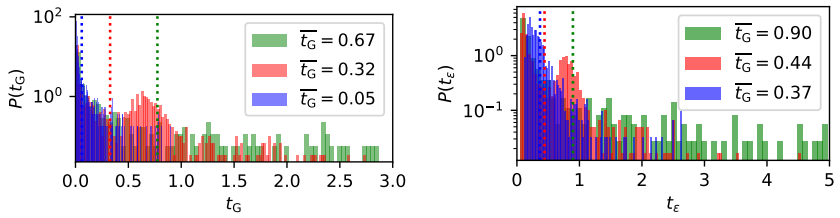

Figure 3: The estimated grokking-time density and t_ϵ density. The colors correspond to `Example 1` (green), `Example 3` (red), and `Example 3` (blue). The vertical lines correspond to the averages reported in the legends of the panels. In the legends we show the critical times for the corresponding examples.

6 Broader impact

The developed tensor-network map offers a new tool for studying generalisation properties of local rules (local teacher-student models), which could lead to more complex learning dynamics (compared to the standard infinite-range rules).

Our results provide further evidence about the benefits of the terminal phase of training and can be relevant also for deep learning training practice. We conjecture that good generalisation is more probable in models with latent space data distributions with small effective dimension. We also find numerical evidence that L_1 regularisation improves generalisation properties of models compared to L_2 regularisation (especially in the last classification layer). Further, we show that spikes in the loss (which often occur during training of deep neural networks) correspond to latent space structural changes that can be beneficial or detrimental for generalisation. Assuming this is the case also in deep networks, the latent space effective dimension can be used to decide whether to revert the model to a state before the spike or to continue training with the current model. We also find that train loss spikes are more common in models with regularisation. While larger regularisation significantly decreases the zero test error time t_ϵ, smaller regularisation (or lack thereof) in the terminal phase of training leads to simpler structures and better generalisation. These finding may also be relevant in more complex models where grokking and structure formation are observed.

Acknowledgments and Disclosure of Funding

The authors received support from Sloveinan research agency (ARRS) project J1-2480. Computational resources were provided by SLING – Slovenian national supercomputing network.

References

[1] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

[2] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

[3] Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

[4] Mohammad Pezeshki, Amartya Mitra, Yoshua Bengio, and Guillaume Lajoie. Multi-scale feature learning dynamics: Insights for double descent. *arXiv preprint arXiv:2112.03215*, 2021.

[5] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

[6] Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. *arXiv preprint arXiv:1906.05890*, 2019.

[7] Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

[8] Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

[9] Vignesh Kothapalli, Ebrahim Rasromani, and Vasudev Awatramani. Neural collapse: A review on modelling principles and generalization. *arXiv preprint arXiv:2206.04041*, 2022.

[10] Ziming Liu, Ouail Kitouni, Niklas Nolte, Eric J Michaud, Max Tegmark, and Mike Williams. Towards understanding grokking: An effective theory of representation learning. *arXiv preprint arXiv:2205.10343*, 2022.

[11] Vimal Thilak, Etai Littwin, Shuangfei Zhai, Omid Saremi, Roni Paiss, and Joshua Susskind. The slingshot mechanism: An empirical study of adaptive optimizers and the grokking phenomenon. *arXiv preprint arXiv:2206.04817*, 2022.

[12] Andreas Engel and Christian Van den Broeck. *Statistical mechanics of learning*. Cambridge University Press, 2001.

[13] Stephen Wolfram. Statistical mechanics of cellular automata. *Reviews of modern physics*, 55(3):601, 1983.

[14] Stephen Wolfram et al. *A new kind of science*, volume 5. Wolfram media Champaign, 2002.

[15] Chu Guo, Zhanming Jie, Wei Lu, and Dario Poletti. Matrix product operators for sequence-to-sequence learning. *Physical Review E*, 98(4):042114, 2018.

[16] Stavros Efthymiou, Jack Hidary, and Stefan Leichenauer. Tensornetwork for machine learning. *arXiv preprint arXiv:1906.06329*, 2019.

[17] Bojan Žunkovič. Deep tensor networks with matrix product operators. *Quantum Machine Intelligence volume*, 4(21), 2022.

A Compute Resources

All our experiments are performed on a compute cluster managed by Slurm Workload Manager. Each node has access to four NVidia A100 with 40 GB HBMI2. Estimated total compute time for presented experiments is 110 days.

Checklist

1. For all authors...

(a) Do the main claims made in the abstract and introduction accurately reflect the paper's contribu- tions and scope? [Yes]

(b) Did you describe the limitations of your work? [Yes] The limitations should be clear from the setup described in Section 2 and Section 3.

(c) Did you discuss any potential negative societal impacts of your work? [N/A]

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes] in Section 3 and Section 3

(b) Did you include complete proofs of all theoretical results? [Yes] Although the work is primarily numerical the setup is novel and described in detail in Section 3 and Section 3.

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] At the end of the introduction.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] At the beginning of Section 4

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report the standard deviation where applicable, see Fig. 1

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix A

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [N/A]

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're us- ing/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable informa- tion or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]