Towards Understanding the Mystery behind Grokking Phenomenon via Model and Training Efficiency

Abstract

This project delves into the intriguing phenomenon of grokking in modular addition, examining its manifestations across different network architectures and varied training configurations. Our investigation reveals that grokking is a universal occurrence observed in various models, with a notable impact from the fraction of training data used. Moreover, our findings suggest that incorporating weight decay and introducing controlled noise during training can enhance the generalization ability of the models. To provide deeper insights, we endeavor to elucidate the grokking phenomenon and conduct a series of experiments across diverse models to validate our explanations.

1 Introduction

The phenomenon of grokking, characterized by the model's ability to generalize long after overfitting the training set, has been observed in algorithmic datasets [7]. This intriguing phenomenon highlights the delayed generalization exhibited by neural networks on specific learning problems. While grokking has garnered attention, only a limited number of studies have delved into its analysis. Notable works exploring this phenomenon include the method of mechanistic explanations through reverse engineering [5], the investigation into how Representation Learning generalization stems from structured representations [3], and the exploration of Modular Arithmetic to further minimize the right features [1].

In this project, we delve into the phenomenon of grokking in modular addition across diverse network architectures, such as Transformer, LSTM, and MLP. Surprisingly, we observe that grokking is a universal occurrence present in various models. An intriguing revelation surfaces as we investigate the impact of the training data fraction on this phenomenon. Notably, as the training data fraction increases, a more pronounced alignment between the training and validation curves emerges, resulting in a diminished or negligible grokking effect. Moreover, our exploration extends to the influence of different optimizers and regularization methods on data efficiency. Notably, we discover that factors like weight decay and the introduction of controlled noise play pivotal roles in enhancing the generalization ability of the models. These findings shed light on the intricate interplay between model architecture, data fraction, and optimization strategies in the context of grokking phenomenon.

Furthermore, our investigation draws inspiration from the insightful work [9], where they provide an elucidation of the grokking phenomenon. Our exploration commences with an analysis inspired by their approach, focusing on the efficiency of memorization and generalization models. Specifically, we distinguish between models achieving full training accuracy yet limited validation accuracy (memorization models) and those achieving full accuracy on both training and validation sets (generalization models). This analytical lens reveals three crucial properties that contribute significantly to the grokking phenomenon: the existence and accessibility of memorization and generalization, superior model efficiency for generalization models, and enhanced training efficiency for memorization models. Building upon the foundations laid by [9], which primarily concentrated on Transformers, our experiments extend across various network architectures, encompassing Transformer, LSTM, and

Final project for Mathematical Introduction to Machine Learning course in Peking University (2023 fall).

MLP, all applied to the modular addition task. The empirical evidence we gather strongly supports and extends the explanatory framework proposed in [9].

2 Preliminaries

Task Setup. Our task setup aligns with the framework proposed in [7], adopting binary operation tables of the form $a \circ b = c$, where a, b, and c are discrete tokens lacking structural information, and $\circ$ denotes a binary operation. For the sake of simplicity, our focus in this project is specifically on the modular addition task, represented as $(x, y) \to (x + y) \bmod p$ for $x, y \in \mathbb{Z}_p$. Due to the finite nature of input tokens and the absence of structural details, we approach the task as a classification problem. In this classification setup, the classes correspond to all possible tokens within the defined domain.

Classification Problems and Logits. In our study of classification problems, we consider two common loss functions: cross-entropy loss and mean squared error (MSE) loss. Given a set of inputs X, labels Y, and a training set $\mathcal{D} = \{(x_1, y_1^*), \cdots, (x_D, y_D^*)\}$, the objective is to train a classifier with optimal performance on both training and validation data. In the context of the modular addition task, where X and Y represent all possible tokens, any classifier $h : X \times Y \to \mathbb{R}$ generates the class **logit** for a specific class, denoted as $o_h^y(x) := h(x, y)$. We sometimes simplify this notation to o_h^y when the input x is clear from context. Additionally, the vector of logits for all classes corresponding to a given input is denoted as $\vec{o}_h(x)$ or simply $\vec{o}_h$. The softmax function is applied to this vector to obtain the probabilities of all classes. Subsequently, the cross-entropy loss and MSE loss are defined as follows:

$$\mathcal{L}_{\text{ent}}(h) = -\frac{1}{D} \sum_{(x,y^*)\in\mathcal{D}} \log \frac{\exp\left(h(x, y^*)\right)}{\sum_{y'\in Y} \exp\left(h(x, y')\right)}$$

$$\mathcal{L}_{\text{mse}}(h) = -\frac{1}{D} \sum_{(x,y^*)\in\mathcal{D}} \left[\left(\frac{\exp\left(h(x, y^*)\right)}{\sum_{y'\in Y} \exp\left(h(x, y')\right)} - 1 \right)^2 + \sum_{\hat{y}\in Y, \hat{y}\neq y^*} \left(\frac{\exp\left(h(x, \hat{y})\right)}{\sum_{y'\in Y} \exp\left(h(x, y')\right)} \right)^2 \right]$$

Weight decay. Parametric classifiers, exemplified by neural networks, are characterized by a parameter vector θ that defines the classifier h_θ. In practical settings, it is common to introduce weight decay regularization as an additional term to the loss function, denoted as $\mathcal{L}_{\text{wd}}(h) = \frac{1}{2}\|\theta\|^2$. Consequently, the overall loss function is expressed as:

$$\mathcal{L}(h) = \mathcal{L}_{\text{data}}(h) + \alpha \mathcal{L}_{\text{wd}}(h)$$

Here, $\mathcal{L}_{\text{data}}$ represents the loss on training data, such as $\mathcal{L}_{\text{ent}}$ or $\mathcal{L}_{\text{mse}}$, and α serves as a control parameter governing the influence of weight decay. The inclusion of weight decay aims to prevent overfitting and encourage the learning of simpler models by penalizing large values in the parameter vector θ.

Memorization Model and Generalization Model. Within the context of a defined model space and datasets for training and validation, it's common to observe the existence of two distinctive models: the memorization model (C_{mem}) and the generalization model (C_{gen}). The memorization model achieves full accuracy on the training dataset but exhibits limited accuracy on the validation dataset. In contrast, the generalization model achieves full accuracy on both the training and validation datasets. This duality in performance highlights the trade-off between memorization and generalization capabilities within the model space.

3 Exploring Grokking Phenomenon on Modular Addition

Our initial focus lies on understanding the grokking phenomenon within the context of modular addition, specifically represented as $(x, y) \to (x + y) \bmod p$ for $x, y \in \mathbb{Z}_p$. Remarkably, we observe the manifestation of the grokking phenomenon across diverse models, encompassing Transformer [10], LSTM [2], and MLP. This cross-model presence underscores the universality of the grokking phenomenon. Furthermore, our exploration extends beyond model types to investigate how various training configurations impact the grokking phenomenon. We delve into the influence of factors such as training data fraction, different optimizers, and various regularization methods. This comprehensive examination aims to provide insights into the nuanced dynamics of grokking across different learning scenarios.

3.1 Grokking Appears on Various Models

We conducted training sessions on a two-layer Transformer, two-layer LSTM, and three-layer MLP using the AdamW optimizer for the modular addition task, utilizing 50% of the training data. The total non-embedding parameters in these models range from approximately 3×10^5 to 4×10^5. The training and validation accuracy trajectories for each model are depicted in Figure 1.

Notably, training the Transformer on modular addition reveals a distinctive pattern where both training and validation accuracy rise concurrently until achieving perfect training accuracy. Subsequently, the validation accuracy plateaus at intermediate levels before surging to full validation accuracy. In contrast, training LSTM or MLP on modular addition exhibits a different dynamic: the validation accuracy remains close to random guessing until a point post-perfect training accuracy, after which it accelerates to reach perfect prediction. Despite nuanced differences in dynamics, a commonality emerges as the validation accuracy remains stagnant for an extended period before its upward trajectory continues.

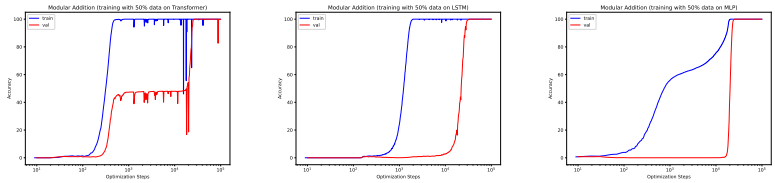

Figure 1: Grokking on Transformer **(left)**, LSTM **(middle)**, and MLP **(right)**.

3.2 Training Data Fraction Impacts Grokking Significantly

To investigate the relationship between training data fraction and generalization, we varied the training data fraction while training models. Figures 2 and 3 illustrate the training and validation curves for Transformer and LSTM models at 30%, 50%, and 70% training data fractions.

As anticipated, an increase in training data fraction correlates with enhanced generalization to validation data, aligning with the expectation of less or no significant grokking. Notably, when training the Transformer, the middling point of validation accuracy approaches full accuracy more closely, and the validation accuracy exhibits a swifter ascent from this midpoint to full accuracy as the training data fraction increases. When training the LSTM, though the number of training steps required to achieve full training accuracy sees a slight increase with higher training data fractions, there is a substantial reduction in the training steps needed to attain full validation accuracy. These observations reinforce the notion that a higher training data fraction facilitates more effective generalization and diminishes the grokking phenomenon.

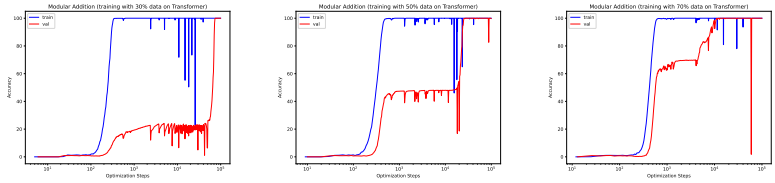

Figure 2: Grokking on Transformer with 30% **(left)**, 50% **(middle)**, and 70% **(right)** training data.

3.3 Data Efficiency for Different Optimizers and Regularization Methods

In our exploration of the impact of different optimizers and regularization methods on the generalization ability of models in the modular addition task, we considered a range of interventions, including diverse batch sizes, learning rates, residual dropout [8], weight decay [4], and gradient noise [6]. Figure 4 illustrates the data efficiency curves for the Transformer model under these interventions.

Our findings reveal that incorporating weight decay towards zero significantly enhances data efficiency, with weight decay towards the initialization of the network also proving effective albeit

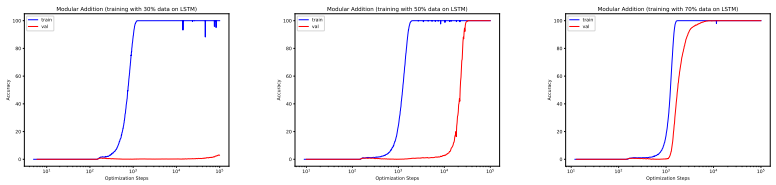

Figure 3: Grokking on LSTM with 30% **(left)**, 50% **(middle)**, and 70% **(right)** training data.

to a lesser extent. The introduction of noise to the optimization process, whether through update noise, weight noise, or dropout, consistently demonstrates a positive impact on data efficiency, aligning with the notion that noise aids in generalization. Additionally, we observe that learning rate demands careful tuning to ensure both good data efficiency and stable training. These insights emphasize the importance of thoughtful optimization and regularization strategies in achieving optimal generalization performance.

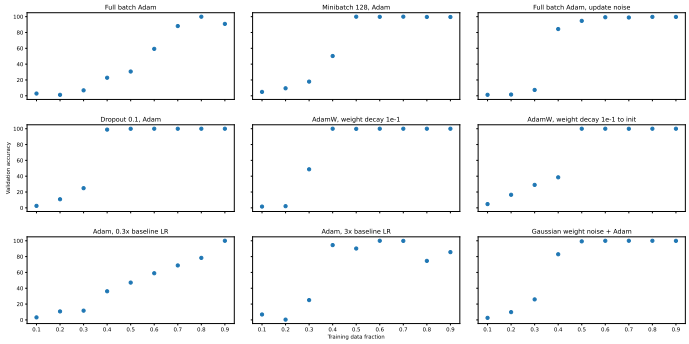

Figure 4: Data efficiency curves on modular addition task using Transformer for a variety of configs: **(first row)** full batch Adam, Adam with batchsize 128, full batch Adam with update noise, **(second row)** Adam with drop 0.1, AdamW with weight decay 0.1 to zero, AdamW with weight decay 0.1 to init, **(third row)** Adam with learning rate 1e-4, Adam with learning rate 1e-3, Adam with Gaussian weight noise.

4 Explanation of Grokking Phenomenon

In this section, our objective is to provide a comprehensive explanation for the grokking phenomenon, drawing insights from both model efficiency and training efficiency. We begin by illustrating how the disparities in model efficiency and training efficiency between the memorization model (C_{mem}) and the generalization model (C_{gen}) contribute to the emergence of grokking.

The differences observed in the efficiency aspects of these models set the stage for understanding grokking, paving the way for a more nuanced comprehension of the relationship between model efficiency and dataset size. Through these insights, we can predict two related phenomena, shedding light on the underlying mechanisms that drive the grokking phenomenon.

4.1 Model Efficiency and Training Efficiency

Our exploration into the grokking phenomenon commences with a focus on model efficiency, quantified by the model's ability to produce large logits for the correct class using a minimal set of parameters. As the training process unfolds, the loss on training data, such as $\mathcal{L}_{\text{ent}}$ or $\mathcal{L}_{\text{mse}}$, exerts pressure on the model, pushing it towards larger parameters to improve accuracy. Simultaneously, regularization methods, represented by $\mathcal{L}_{\text{wd}}$, counteract this trend by encouraging smaller parameter values. In parallel, the concept of training efficiency comes into play, representing the optimization steps required to attain specific model parameters. This dual consideration of both model and training efficiency provides a nuanced understanding of how the interplay between loss functions and

regularization methods during training shapes the model parameters and influences the grokking phenomenon.

The grokking phenomenon unfolds as a dynamic interplay between model efficiency and training efficiency, particularly in the contrasting behaviors of memorization models (C_{mem}) and generalization models (C_{gen}). During the initial stages of training, C_{mem} rapidly learns, achieving full training accuracy but limited validation accuracy. Subsequently, as training progresses, C_{gen} emerges, guided by regularization methods. The model parameters gradually shift from those of C_{mem} to C_{gen}, culminating in the attainment of perfect training and validation accuracy.

This two-phase training process gives rise to the grokking phenomenon observed in learning curves. The explanation relies on three key conditions:

- **Existence of memorization models and generalization models:** Both C_{mem} and C_{gen} coexist within the model space and are easily attainable through major optimization methods. It's worth noting that this condition may not hold when the training data fraction is large, as perfect training and validation accuracy may be achieved simultaneously.
- **Model efficiency:** C_{gen} exhibits significantly greater efficiency than C_{mem}, producing equivalent logits on the training set with a smaller parameter norm.
- **Training efficiency:** The learning rate of C_{gen} is slower compared to C_{mem}, resulting in the dominance of C_{mem} during the early stages of training.

In summary, the grokking phenomenon is a consequence of the distinct characteristics of memorization and generalization models, their efficiency disparities, and the differential rates at which they learn during the training process.

4.2 Relationship of Model Efficiency with dataset size

To grasp the impact of training data fraction on grokking, we delve into the relationship between model efficiency and dataset size. Consider a dataset $\mathcal{D}$ of size D and another dataset $\mathcal{D}'$ with an additional data point, i.e., $\mathcal{D}' = \mathcal{D} \cup (x, y^*)$. Let $h_{\mathcal{D}}$ and $h_{\mathcal{D}'}$ represent two classifiers trained on $\mathcal{D}$ and $\mathcal{D}'$ with weight decay, respectively. Intuitively, the efficiency of $h_{\mathcal{D}'}$ cannot surpass that of $h_{\mathcal{D}}$. If it did, $h_{\mathcal{D}'}$ would outperform $h_{\mathcal{D}}$ on the training dataset $\mathcal{D}$ due to similar loss on training data but better weight decay. Consequently, the model efficiency tends to be non-increasing as the dataset size increases.

We demonstrate that the model efficiency of C_{mem} tends to decrease as the dataset size increases, while the model efficiency of C_{gen} remains unchanged with increasing dataset size. Consider $h_{\mathcal{D}}$, a classifier predicting the new data point (x, y^*) correctly. In this scenario, transitioning from dataset $\mathcal{D}$ to $\mathcal{D}'$ results in nearly unchanged loss on training data and regularization loss. However, if $h_{\mathcal{D}}$ fails to predict the new data point (x, y^*) correctly, the loss on training data substantially increases. To mitigate this increase in loss, additional regularization loss is incurred, affecting the model efficiency. Consequently, $h_{\mathcal{D}'}$ becomes less efficient than $h_{\mathcal{D}}$. Given that C_{mem} consistently struggles to generalize to new data points, its model efficiency experiences a decline with increasing dataset size. In contrast, C_{gen} consistently generalizes well to new data points. This characteristic ensures that the model efficiency of C_{gen} remains unchanged despite variations in dataset size.

For extremely small datasets, where the model can effortlessly memorize all data points, C_{mem} initially exhibits higher efficiency than C_{gen}. However, as indicated earlier, the efficiency of C_{mem} diminishes as the training dataset size increases. We anticipate the existence of a critical dataset size, denoted as D_{crit}, where C_{mem} and C_{gen} converge to similar model efficiency. Beyond D_{crit}, as the dataset size significantly exceeds this critical point ($D \gg D_{\text{crit}}$), the training process undergoes a shift from favoring C_{mem} to embracing C_{gen}. This transition leads to the observed grokking phenomenon, where C_{gen} starts dominating the learning process. Conversely, when the dataset size is substantially smaller than D_{crit} ($D \ll D_{\text{crit}}$), the training process consistently favors C_{mem}, and grokking fails to materialize. The critical dataset size thus serves as a pivotal threshold, dictating the dynamics of the learning process and the manifestation of the grokking phenomenon.

4.3 Predictions of two phenomena: ungrokking and semi-grokking

Based on the analysis of model efficiency and the relationship with the critical dataset size D_{crit}, we can further predict two relative phenomena as follows:

- **Ungrokking:** If a classifier is initially trained on a large dataset, achieving full training and validation accuracy, and subsequently continues training on a significantly smaller dataset ($D \ll D_{\text{crit}}$), C_{mem} becomes more efficient than C_{gen}. Consequently, the training process favors C_{mem}, leading to a decline in validation accuracy. This phenomenon, termed "ungrokking", highlights the impact of dataset size on the dominance of memorization models.
- **Semi-grokking:** When training a classifier on a dataset with a size approximately equal to the critical dataset size ($D \approx D_{\text{crit}}$), where C_{mem} and C_{gen} exhibit similar model efficiency, the training process can yield C_{mem}, C_{gen}, or a combination of both. This results in varying degrees of validation accuracy, ranging from zero (full dominance of C_{mem}) to full accuracy (full dominance of C_{gen}) or an intermediate accuracy level. This nuanced scenario is termed "Semi-grokking," capturing the interplay between memorization and generalization models.

5 Experimental Evidence

In this section, we systematically investigate and validate the above explanation of the grokking phenomenon. Section 5.1 underscores the crucial role of differences in training efficiency in the manifestation of grokking. Moving forward, Section 5.2 delves into the assessment of model efficiency for both C_{mem} and C_{gen}, elucidating how these efficiencies are influenced by variations in training dataset size. Finally, Section 5.3 is dedicated to verifying the existence of the predicted phenomena, specifically ungrokking and semi-grokking.

5.1 Differences in Training efficiency is Necessary

We employ a mixed MLP, denoted as C_{mix}, characterized by the combination $w_{\text{mem}}C_{\text{mem}} + w_{\text{gen}}C_{\text{gen}}$, where C_{mem} and C_{gen} represent pre-trained MLPs. Notably, C_{mem} achieves full training accuracy and modest validation accuracy, while C_{gen} attains full validation accuracy. During the training of the mixture MLP, our focus is solely on updating the weights w_{mem} and w_{gen}, with the architectures of C_{mem} and C_{gen} remaining unchanged.

To effectively control the training efficiency of C_{mem} and C_{gen}, we use a parameterization strategy: $w_{\text{mem}} = w_{m_1}w_{m_2}$ and $w_{\text{gen}} = w_{g_1}w_{g_2}$. This formulation ensures that the value of w_{m_2} influences the gradient of w_{m_1}, providing a mechanism to control the training efficiency. Proper initialization of w_{m_1}, w_{m_2}, w_{g_1}, and w_{g_2} allows for fine-tuned control over the training efficiency of w_{mem} and w_{gen}, contributing to a nuanced exploration of the impact of training efficiency on the grokking phenomenon.

In our experiments, we initialize $w_{m_1} = w_{g_1} = 0$ uniformly across all settings. We then configure two scenarios for the initialization of w_{m_2} and w_{g_2} : first, with $w_{m_2} = 10$ and $w_{g_2} = 1$, indicating that C_{gen} learns more slowly than C_{mem}; and second, with $w_{m_2} = w_{g_2} = 1$, reflecting equal learning speeds for C_{mem} and C_{gen}. Figure 5 presents the accuracy and loss curves corresponding to these different settings. Notably, when $w_{m_2} = 10$ and $w_{g_2} = 1$, significant grokking is observed. This is evident in the validation loss increasing while the training loss decreases, followed by a subsequent decrease in validation loss after the training loss approaches zero. Conversely, in the scenario where $w_{m_2} = w_{g_2} = 1$, denoting equal learning speeds for C_{mem} and C_{gen}, the training and validation loss curves decrease in tandem, and no significant grokking manifests. While acknowledging the inherent complexities in the experiment, the observed patterns highlight the necessity of a difference in training efficiency between C_{mem} and C_{gen} for the emergence of grokking.

5.2 Relationship of Model efficiency with dataset size

In Figure 6, we analyze the parameter norm and the predictions of correct logits on select training data for both C_{mem} and C_{gen} during training on an MLP with varying training data fractions. Notably, our observations reveal distinct patterns in model efficiency between C_{mem} and C_{gen}. In the case of

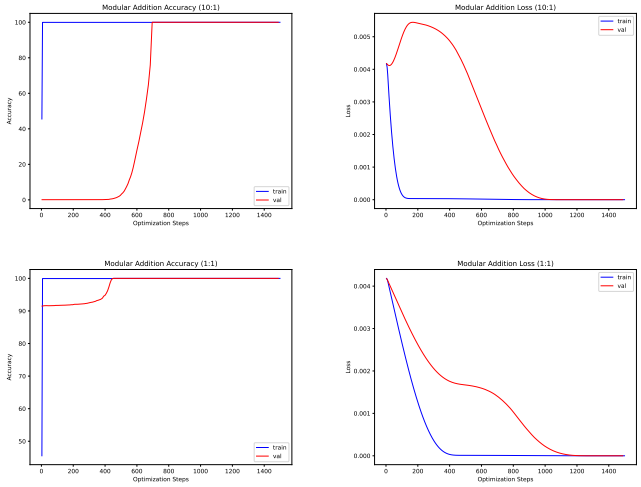

Figure 5: Accuracy and loss curves of mixture MLP with different initializations. **(first row)** Accuracy and loss curves for $w_{m_2} = 10, w_{g_2} = 1$ with significant grokking. The validation loss goes down only after training loss gets close to zero. **(second row)** Accuracy and loss curves for $w_{m_2} = w_{g_2} = 1$ with no significant grokking. The training and validation loss goes down together.

C_{mem}, we note that a higher parameter norm is required as the dataset size increases to generate same logit. This trend suggests diminished model efficiency for C_{mem} when faced with larger datasets. The necessity for increased parameter norm implies that C_{mem} exhibits less model efficiency in leveraging its parameters effectively as the dataset size expands. However, the parameter norm of C_{gen} remains relatively stable when dataset size increases. This consistency in parameter norm underscores the similar model efficiency of C_{gen} across varying dataset sizes.

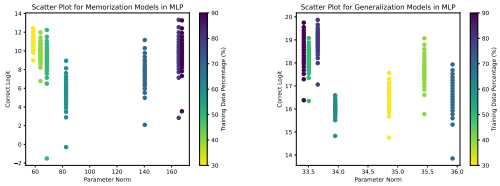

Figure 6: Parameter norm and prediction of correct logits on some training data of C_{mem} and C_{gen} obtained by training on MLP. The model efficiency of C_{mem} decreases while the model efficiency of C_{gen} remains unchanged as the dataset size increases.

5.3 Experimental Evidence for Ungrokking and Semi-grokking

This section aims at providing experimental evidence for ungrokking and semi-grokking phenomenon.

- **Ungrokking:** Figure 7 presents experimental results confirming the ungrokking phenomenon across diverse architectures, namely Transformer, LSTM, and MLP. The key observation is that when a pre-trained model, initially exhibiting perfect training and validation accuracy, is subsequently trained on a sufficiently small dataset, the validation accuracy gradually decreases to zero. Furthermore, the exploration extends to MLP models with varying weight decay, as showcased in Figure 8. The consistent occurrence of ungrokking across different weight decay values underscores its independence from the specific choice of weight decay.
- **Semi-grokking:** Experimental evidence supporting the semi-grokking phenomenon is illustrated in Figure 9 using a Transformer model. As the training data fraction surpasses 27%, the observed grokking phenomenon manifests after 6e4 steps, indicative of a transition from C_{mem} to C_{gen}. Notably, when the training data fraction ranges from approximately 23%

to 27%, middling validation accuracy is observed. This intermediate accuracy level signifies a mixture of C_{mem} and C_{gen} during the training process.

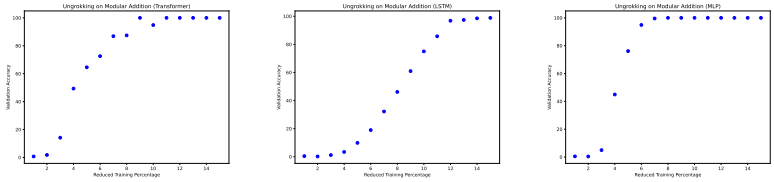

Figure 7: Ungrokking phenomenon on Transformer **(left)**, LSTM **(middle)**, and MLP **(right)**.

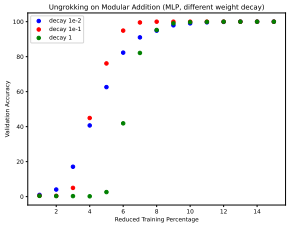

Figure 8: Ungrokking phenomenon on MLP with different weight decay. This shows that the appearance of ungrokking phenomenon is independent of the value of weight decay.

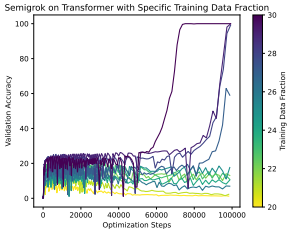

Figure 9: Semi-grokking on Transformer. Runs with $D \approx D_{\text{crit}}$ may obtain a mixture of C_{mem} and C_{gen} and get middling validation performance.

References

[1] California 94025 Department of Physics Condensed Matter Theory Center University of Maryland College Park Maryland 20740 Andrey Gromov Meta AI Meta Platforms, Inc. Menlo Park. Grokking modular arithmetic. arXiv:2301.02679, 2023.

[2] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[3] Ziming Liu, Ouail Kitouni, Niklas Nolte, Eric J. Michaud, Max Tegmark, and Mike Williams. Towards understanding grokking: An effective theory of representation learning. *Department of Physics, Institute for AI and Fundamental Interactions, MIT*, arXiv.2205.10343, 2022.

[4] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[5] Neel Nanda, Lawrence Chan, Tom Lieberum† Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. arXiv.2301.05217, 2023.

[6] Arvind Neelakantan, Luke Vilnis, Quoc V. Le, Ilya Sutskever, Lukasz Kaiser, Karol Kurach, and James Martens. Adding gradient noise improves learning for very deep networks, 2015.

[7] Alethea Power, Yuri Burda, Harrison Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *CoRR*, abs/2201.02177, 2022.

[8] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[9] Vikrant Varma, Rohin Shah, Zachary Kenton, János Kramár, and Ramana Kumar. Explaining grokking through circuit efficiency, 2023.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. 30, 2017.