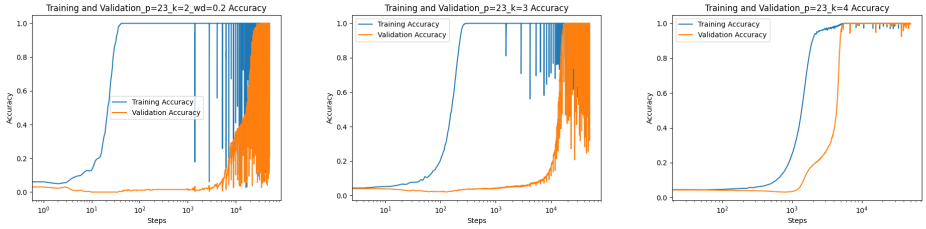
Modular Addition with K Numbers

The dataset size for modular addition with K numbers scales as p^K, growing exponentially with K. As K increases, the dataset becomes significantly larger, and the problem grows more complex. To investigate the variations in the Grokking phenomenon as KKK increases, we conduct experiments with p=23 and K=2,3,4. The corresponding batch sizes are set to 1024, 1024, and 2048, respectively. The training data fractions are 0.5, 0.3, and 0.2, while the weight decay values are 0.2, 0.1, and 0.003, respectively.

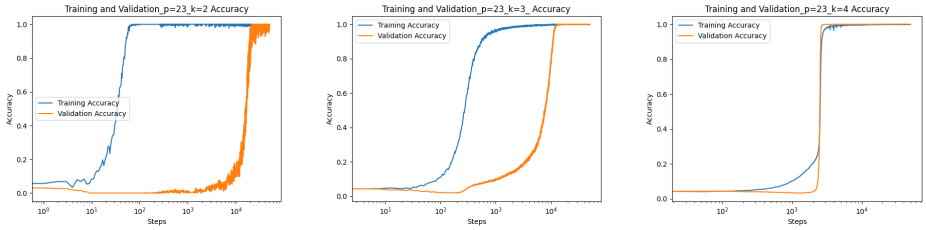
Under these settings, we performed experiments with dropout values of 0 and 0.1. The results, as shown in the figures, indicate that Grokking occurs earlier as K increases. Additionally, the conditions for Grokking to emerge become more stringent, even as we progressively decrease the training data fraction and weight decay. During training, we observed that overly weak regularization leads to slow improvements in validation accuracy, whereas overly strong regularization causes training accuracy and validation accuracy to rise in tandem. This effect becomes more pronounced with larger K. For example, with K=4 and dropout = 0.1, strong regularization resulted in both training accuracy and validation accuracy rising almost simultaneously to 100%.

A possible explanation for this behavior is as follows: while the model framework remains unchanged, the number of parameters grows linearly with KKK, whereas the dataset size grows exponentially. This discrepancy implies that the increase in problem complexity outpaces the growth in model capacity, reducing the degree of overparameterization and narrowing the difficulty gap between memorization and generalization. Consequently, Grokking tends to occur earlier.

accuracy，dropout=0：



accuracy，dropout=0.1



Explanation of the Grokking Phenomenon

1. **Phases of Grokking**

In previous studies [1], Grokking is divided into three phases: Memorization, Circuit Formation, and Cleanup. However, based on our experiments on modular addition, we find it more suitable to classify Grokking into two phases: Memorization and Generalization.

- **Memorization:** As illustrated in the loss-steps curve, during the initial training phase, the model focuses on fitting the training set. Training loss decreases, but performance on the test set is worse than random predictions, resulting in increasing validation loss.
- **Generalization:** After a certain point, the training loss plateaus. Under the influence of regularization, the model attempts to generalize, leading to a sharp decrease in test loss.

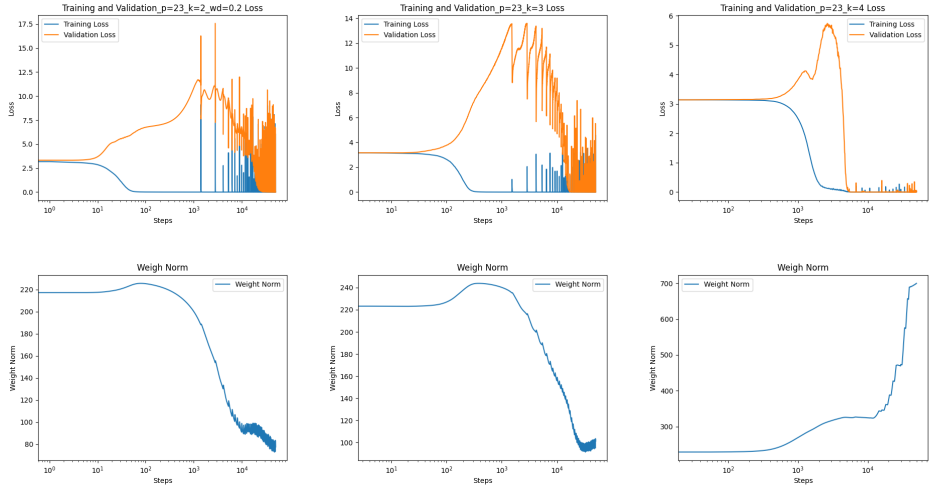
This simplified categorization arises from our observation that regularization techniques such as weight decay and dropout not only facilitate generalization and the emergence of Grokking but also play roles beyond merely reducing the weight norm and simplifying the model.

2.**Weight Norm**

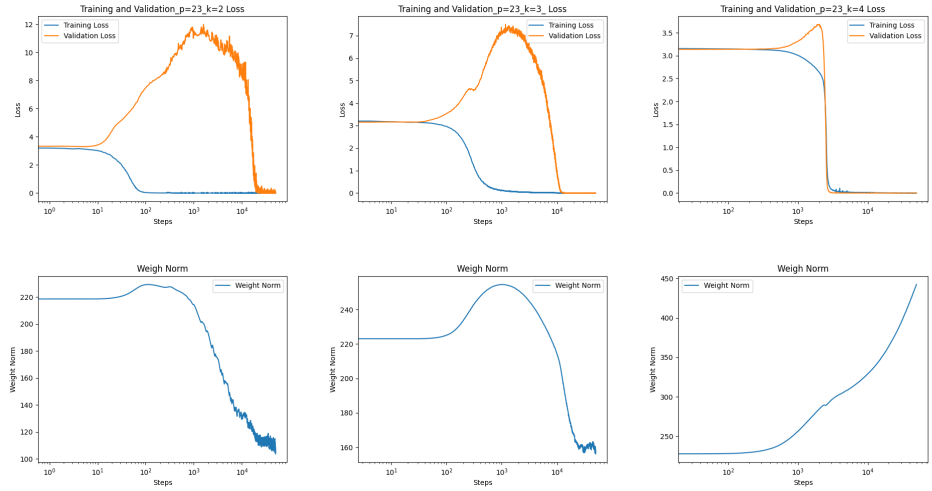
In Subtask 3, we demonstrated that appropriate regularization effectively enhances the model's generalization ability and promotes Grokking. In Subtask 4, we monitored the changes in weight norm over steps for K=2,3,4. For K=2,3, the weight norm behavior aligned with our expectations: it remained high initially, indicating the model's strong capacity to memorize the training set. Subsequently, the weight norm decreased gradually, accompanied by a rapid decline in validation loss. This suggests that the model was attempting to find simpler solutions to better generalize.

However, for K=4, an unexpected trend was observed: the weight norm continued to increase throughout training. Surprisingly, this growth phase (10^3 - 10^4) coincided with the phase where validation loss decreased. This observation implies that for K=4, the complexity of the problem exceeds the model's ability to express it in a "simpler" manner.

loss & weigh norm ， dropout=0



loss & weigh norm， dropout=0.1



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