On the Periodic Behavior of DNNs training on the example of the Grokking effect

A Preprint

Мельник Ю. М.
Кафедра ММП, факультет ВМК
МГУ им. М.В. Ломоносова
melnik.um@yandex.ru

Южаков Т. А.
ФКН, НИУ ВШЭ
Исследовательская группа Байесовских методов

Ветров Д.П.
кандидат ф.-м. наук
Профессор НИУ ВШЭ
Исследовательская группа Байесовских методов

Abstract

Глубокие нейронные сети часто обучаются с использованием слоёв нормализации, что позволяет стабилизировать процесс обучения и повысить точность предсказания модели. Однако иногда использование подобных техник в совокупности с применением методов регуляризации может привести к возникновению необычных эффектов при обучении нейросетей. В данной работе приведены результаты исследования одного из таких эффектов, а именно `периодического поведения`, которое выражается в том, что в процессе обучения нейросети изменение значений лосс функции и основных метрик происходит по определённому повторяющемуся шаблону. На примере эффекта «`гроккинга`» - явления, связанного с переобучением нейросетевых, в частности, трансформерных моделей - было показано, что периодического поведения при обучении глубоких нейросетей можно добиться путём использования `масштабно-инвариантных` архитектур. Также была установлена связь между возникновением периодического поведения и изменением значений `эффективного темпа обучения` и средней нормы `эффективного градиента` в процессе обучения модели.

Keywords Grokking · Loss Landscape · Weight Decay · Scale-Invariance

1 Введение

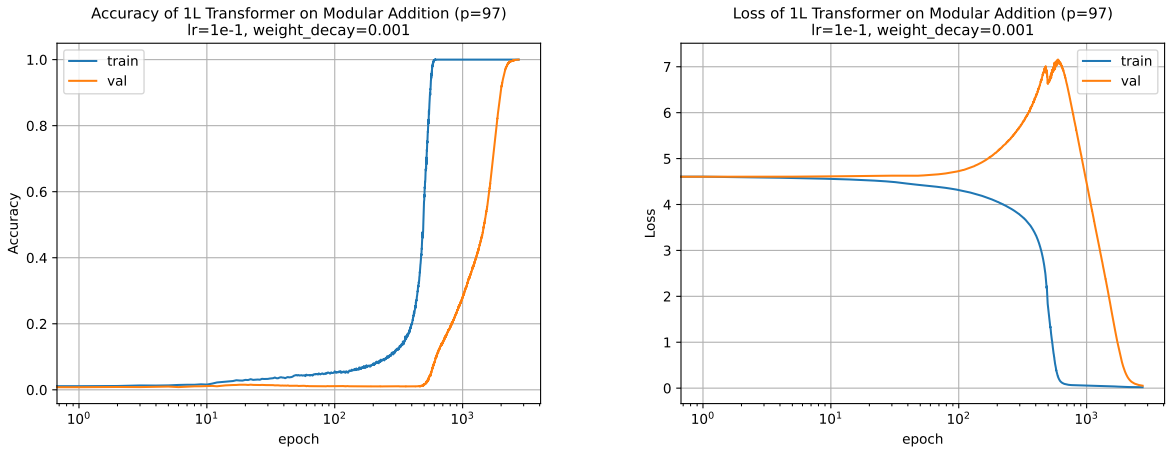
Обобщение перепараметризованных нейронных сетей уже давно является источником интереса для сообщества машинного обучения, поскольку оно бросает вызов интуиции, вытекающей из классической теории обучения [17, 20]. Так, в статье [14] описываются результаты использования слоёв нормализации `BatchNorm` [5] вместе с регуляризатором «weight decay» [11, 15, 6] и возникающее в процессе обучения нейросети периодическое поведение, которое заключается в том, что изменение значений лосс функции и основных метрик происходит по определённому повторяющемуся шаблону. Другим же необычным эффектом, возникающим при обучении перепараметризованных нейросетей, является эффект «гроккинга», впервые обнаруженный авторами статьи [19]. Исследователями из OpenAI было показано, что обучаемые на небольших алгоритмически сгенерированных наборах данных сети могут демонстрировать необычные шаблоны обобщения, явно не связанные с качеством на обучающей выборке: если продолжить обучать переобученную модель, то спустя некоторое достаточно большое время это приведёт к росту точности на отложенной выборке. В других работах по изучению эффекта гроккинга авторы делятся своей интуицией по поводу причин его возниконовения. Например, в статье [18] эффект гроккинга рассматривает через призму «фазовых переходов» от запоминания

выборки нейросетью до выучивания закономерностей в данных. В статье же [9] говорится, что эффект гроккинга не что иное, как переход от "ленивой"динамики обучения к стадии "быстрого"выучивания закономерностей в обучающей выборке. Авторы же другой статьи [16] рассматривают эффект гроккинга в контексте так называемого феномена «золотого билета». В данной же работе будут высказаны идеи, отличные от выше описанных, которые, однако, во многом опираются на результаты, полученные авторами статей [19, 13].

Итак, эффект гроккинга можно разбить на два основных этапа:

- точность на обучающей выборке равно 100%, при этом соответствующее значение для отложенной выборки близко к нулю (этап запоминания обучающей выборки, то есть переобучение)
- значение точности на обучающей и отложенной выборках равно 100% (этап выучивания закономерностей в данных, то есть генерализация)

Далее будем называть состояние модели, при котором достигается 100% точности на обучающей выборке, «точкой 1», а состояние, при котором точность на трейне и на валидации достигает 100% одновременно, «точкой 2» (точки пронумерованы в соответствие с хронологией обучения модели). На графиках ниже показано типичное поведение точности и значения функции потерь на обучающей и валидационной выборках при наблюдении эффекта гроккинга.



(a) График точности

(b) График значений функции потерь

Рис. 1: Графики точности и значения функции потерь, типичные для эффекта гроккинга

В качестве модели авторами оригинальной статьи [19] была выбрана небольшая трансформерная архитектура (2 слоя ширины 128 с 4 головками внимания), обучавшаяся с помощью оптимизатора AdamW со следующими параметрами: learning rate = 10^{-3}, weight decay = 1, $\beta_1 = 0.9$, $\beta_2 = 0.98$, linear learning rate warmup первые 10 шагов оптимизации, размер мини-батча 512.

Для обучения нейросетевой модели использовался алгоритмически сгенерированный датасет равенств вида «a o b = c», где «a», «b», «c» - целые неотрицательные числа, а «o» - некая бинарная операция. В качестве бинарной операции «o» авторы использовали бинарные операции по модулю простого числа p, например (x + y) mod p или (x * y) mod p (где x и y - целые неотрицательные числа, непревосходящие p - 1). Важно отметить, что для лучшей воспроизводимости эффекта гроккинга при генерации данных следует выбирать симметричные относительно своих аргументов операции.

Ещё одним важным гиперпараметром модели, влияющим на воспроизводимость гроккинга, является доля обучающей выборки. В зависимости от значения данного параметра точность на валидационной выборке может вести себя по-разному. Крайними ситуациями такого поведения являются нулевые показатели точности (модель переобучилась и генерализация не наступила) и одновременный рост точности на обучающей и валидационной выборках (то есть отсутствие эффекта гроккинга). Стоит отметить, что необходимым условием возникновения эффекта гроккинга является использование при обучении нейросетевой модели регуляризатора весов. Данный факт подтверждается результатами

экспериментов по анализу поведения нормы весов модели в процессе обучения и согласуется с выводами авторов статей [19, 13].

Для понимания причин возникновения периодического поведения в процессе обучения нейросети необходимо определить понятие масштабной инвариантности. Пусть $p(y|x, \theta)$ - предсказание нейросетевой модели. Тогда модель называется масштаб-инвариантной по весам, если выполняется следующее соотношение:

$$log\ p(y|x, \theta) = log\ p(y|x, C\theta), \quad \forall C > 0$$

В сущности это понятие означает, что модель реализует единственную функцию вдоль каждого вектора (в пространстве весов), берущего начало в нуле, вне завивисмости от расстояния до начала координат:

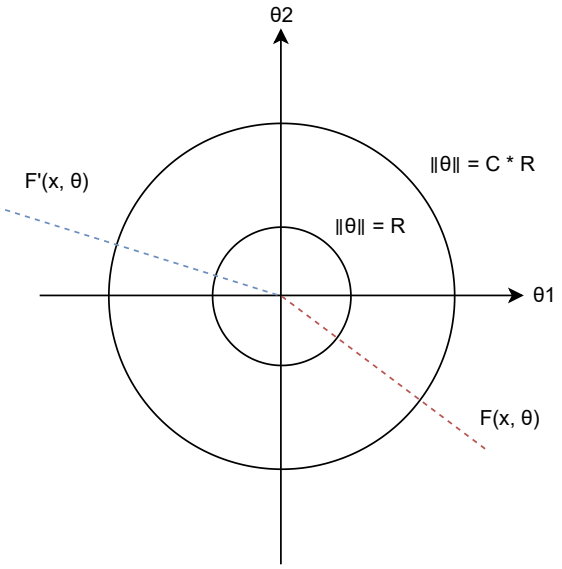


Рис. 2: Визуализация масштабной инвариантности модели для случая двумерного пространства весов. F и F' - функции, которые реализует модель вдоль соответствующих лучей. R - радиус малой окружности, C - вещественная положительная константа.

2 Постановка задачи

В рамках воспроизведения эффекта гроккинга решается задача многоклассовой классификации с числом классов $K = 97$. В качестве лосс функции была выбрана кросс-энтропийная функция потерь. Пусть $X = \{x_i\}_{i=1}^N$, $x_i \in R^2$ - обучающая выборка, $Y = \{y_i\}_{i=1}^N$, $y_i \in \{0, 1, ..., K\}$ $f_\theta(x) : R^2 \to R^{97}$ - функция, реализующаяся нейросетью с параметрами θ, λ - коэффициент регуляризации. Тогда в рамках решения задачи многоклассовой классификации решается следующая оптимизационная задача:

$$\mathcal{L}(X, Y, \theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{K} y_{ik} log \frac{\exp f_\theta(x_i)_k}{\sum_{j=0}^{K} \exp f_\theta(x_i)_j} + \frac{\lambda}{2}\|\theta\|^2 \to \min_\theta$$

Данная оптимизационная задача решается с помощью метода Стохастического градиентного спуска.

3 Эксперименты

3.1 Описание данных

Все эксперименты в данной работе были проведены на алгоритмически сгенерированном датасете равенств вида «a o b = c», описанном в пункте 1.1, где в качестве бинарной операции было выбрано сложение по модулю 97: (x + y) mod 97. Единственное отличие между данными, использовавшимися в оригинальной статье, и датасетом, использовавшемся в этой работе, заключается в том, что при токенизации равенств «a o b = c» не использовались токены для самой бинарной операции «o», а также знака «=» (то есть каждый объект состоит только из трёх токенов, соответствующих операндам и результату). Выбор такого способа токенизации продиктован меньшей зашумлённостью процесса обучения, по сравнению с вариантом, предложенным авторами оригинальной статьи.

3.2 Модель

В качестве модели для проведения экспериментов была выбрана трансформерная архитектура, описанная в статье [13].

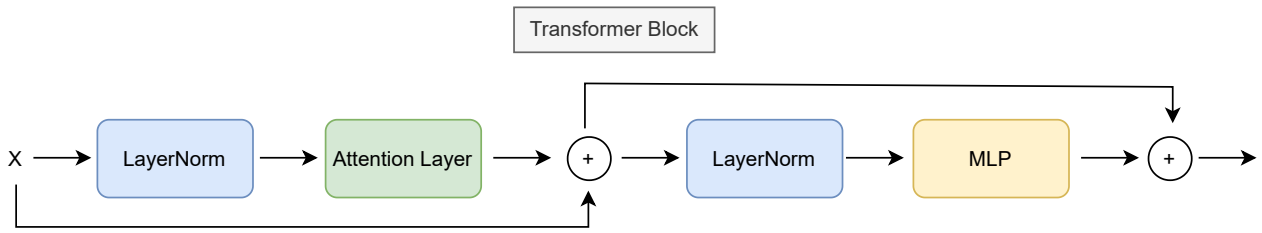


Рис. 3: Блок трансформера используемой нейросетевой архитектуры

Расшифруем обозначения, использованные на рис.3:

- `LayerNorm` - слой нормализации [1]
- `Attention Layer` - слой внимания (является основой трансформерной архитектуры [21])
- `MLP` - два полносвязных линейных слоя с `ReLu` в качестве функции активации после первого из них
- Кружок со знаком «+» внутри означает `Skip Connection`

Для всех последующих экспериментов зафиксируем параметры модели:

- Размер словаря (параметр `d_vocab`) равен 97
- Размерность скрытого пространства модели (параметр `d_model`) равна 128
- Размерность линейного слоя (параметр `d_mpl`) равна 512
- Число голов внимания (параметр `num_heads`) равно 4
- Размерность скрытого пространства механизма внимания (параметр `d_head`) равна 32
- Длина контекста (параметр `n_ctx`) равна 2

Если же говорить про значения параметра `num_layers`, который отвечает за число слоёв трансформерной архитектуры, то в данном исследовании будут рассмотрены однослойная и двуслойная модели (`num_layers` = 1 и `num_layers` = 2).

3.3 Способ обучения

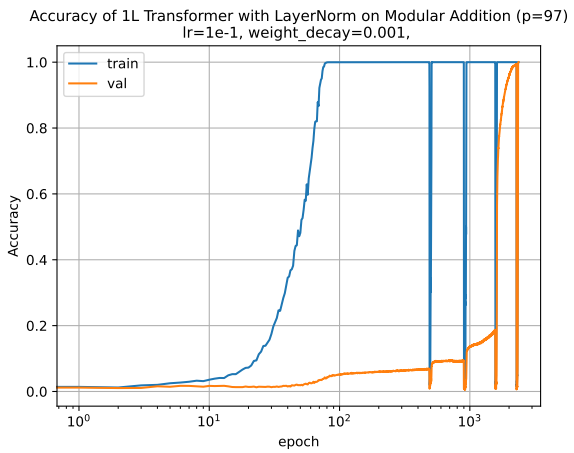
В рамках воспроизведения эффекта гроккинга решается задача многоклассовой классификации, в которой таргетом является токен «c» (результат выполнения бинарной операции (a + b) mode 97 - один из 97 классов), а исходными признаками - токены «a» и «b> (операнды данной бинарной операции). В качестве функционала ошибки для решения данной задачи была выбрана кросс-энтропийная функция потерь. В отличие от оригинальной статьи в данной работе в качестве оптимизатора во всех экспериментах использовался `SGD` (Stochastic Gradient Descent) [2] с постоянным темпом обучения с целью упрощения интерпретабельность результатов. В качестве значений параметров оптимизатора были приняты величины: `lr = 0.1` (темп обучения), `weight_decay = 0.001` (значение константы λ перед регуляризационным слагаемым). Значения гиперпараметров модели, а именно доли обучающей выборки и размера батча, выберем равными 0.4 и 512 соответственно.

3.4 Использование нормализации для достижения периодического поведения

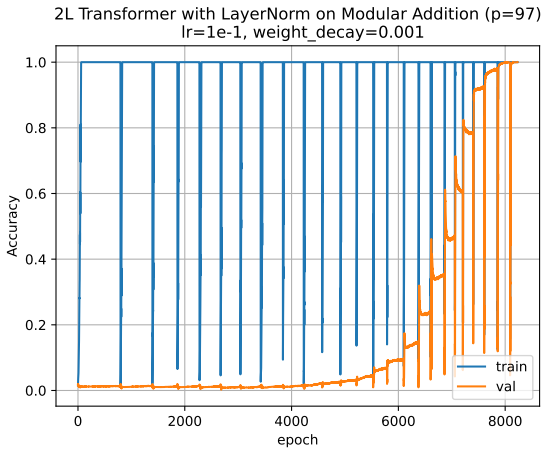
В статье [14] было показано, что использование слоёв нормализации `BatchNorm` после каждого свёрточного слоя сети приводит к возникновению периодического поведения при обучении модели за счёт приобретения ею свойства масштабной инвариантности [12, 8]. Воспользуемся данной идеей для нашей задачи и попробуем добиться периодического поведения засчёт использования слоёв `LayerNorm` в соответствии со схемой, изображённой на рис.3. Помимо блока трансформера, слои нормализации также добавляются и внутрь блока `MLP`, а также перед последним линейным слоем модели, отвечающим за перевод данных из векторного представления (эмбедингов) в токены исходного словаря.

Прежде чем переходить к анализу результатов данного эксперимента стоит сделать важное замечание. Данная модель не является полностью масштаб-инвариантной из-за наличия в ней слоя внимания (`Attention Layer`), поэтому при дальнейшем анализе статистик (раздел 3.2), характерных именно для масштаб-инвариантных сетей, будут сделаны некоторые допущения, которые не умаляют логики рассуждений, а частичную масштаб-инвариантность, свойственную моделям, задействованных в экспериментах, будем называть просто масштаб-инвариантностью для краткости.

Глядя на графики точности на обучающей и валидационной выборках (рис. 4), можно заметить что эффекта гроккинга удалось добиться как для однослойной, так и для двуслойной модели. Однако вместе с тем обучение обеих моделей приобрело желаемый периодический характер: на графиках видны просадки значений точности на обучающей выборке, которым соответствуют восходящие скачки точности на валидации:



(a) График точности однослойного трансформера



(b) График точности двуслойного трансформера трансформера

Рис. 4: Графики точности моделей траснформеров с использованием нормализации

Данный эффект можно объяснить приобретением свойства масштаб-инвариантности большинством весов нейронной сети за счёт использования нормализации между слоями модели.

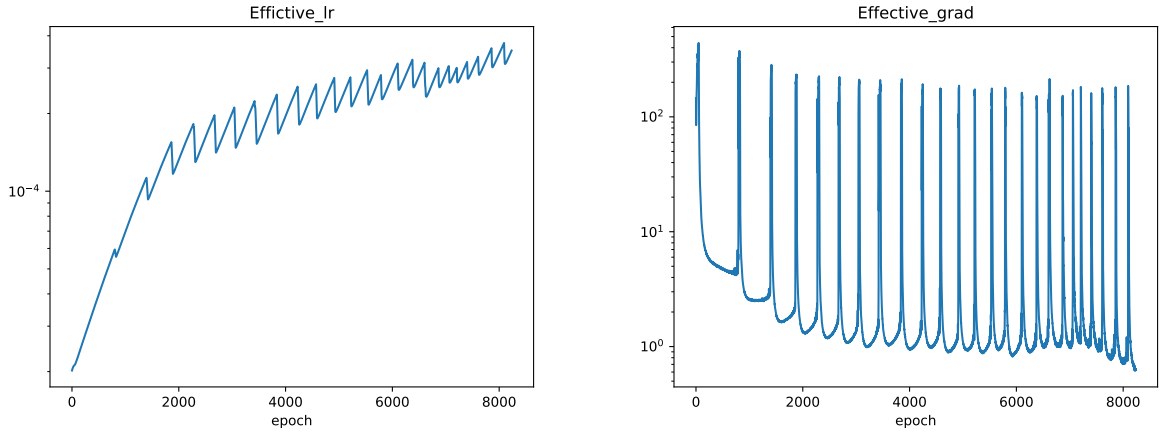
3.5 Анализ статистик обучения

Для дальнейшего анализа статистик обучения необходимо ввести некоторые дополнительные понятия, которые бы учитывали масштаб-инвариантность нейросети [3, 10]. Пусть η - темп обучения, θ - веса модели, g - градиент функции потерь по весам. Тогда определим эффективный градиент и эффективный темпа обучения как:

$$g_{eff} = g\|\theta\|$$
$$\eta_{eff} = \frac{\eta}{\|\theta\|^2}$$

В силу масштаб-инвариантности модели процессу оптимизации во всём пространстве весов можно взаимно однозначно сопоставить процесс оптимизации на единичной гиперсфере. Тогда понятия эффективных градиента и темпа обучения означают градиент и темп обучения при оптимизации на единичной гиперсфере.

Теперь установим причины периодического поведения в процессе обучения на примере модели двуслойного трансформера , график точности которого представлен на рисунке 4b. Для этого проанализируем поведение эффективного темпа обучения и нормы эффективного градиента в процессе обучения.



(a) График эффективного темпа обучения

(b) График нормы эффективного градиента

Рис. 5: Графики эффективных статистик двуслойного трансформера с использованием LayerNorm

Сопоставляя графики на рис. 5 и график точности на рис. 4b, можно заметить, что моменты падений точности на обучающей выборке соответствуют моментам «скачков» эффективных темпа обучения и нормы градиента. При этом график эффективного темпа обучения имеет «пиловидный» характер, что можно объяснить резкими изменениями значения нормы весов модели: «скачки» градиента заставляют совершать перемещение по гиперсферам разного радиуса в пространстве весов, а за счёт наличия регуляризатора weight decay норма весов уменьшается в процессе обучения, что эквивалентно увеличению эффективного темпа обучения. Также стоит отметить тенденцию к уменьшению нормы эффективного градиента по мере перемещения из точки 1 (точность на обучении равна 100%) в точку 2 (точность на обучении и на валидации равна 100%), что является подтверждением перемещения по мере обучения в минимум, обладающий лучшей генерализацией (норма стохастического градиента является метрикой, по которой довольно грубо можно оценить «качество» минимума: чем меньше норма, тем лучшей генерализацией он обладает [7, 4]).

4 Выводы

На основании поставленных в процессе исследования экспериментов можно сделать следующие выводы о возникновении периодического поведения в процессе обучения глубоких нейронных сетей:

1. Периодическое поведение возникает в процессе обучения масштаб-инвариантных нейросетей с использованием регуляризатора, в частности weight decay.

2. Масштаб-инвариантности нейросети (частичной масштаб-инвариантности) можно добиться путём использования слоёв нормализации (в частности `LayerNorm`) между каждыми двумя слоями модели.
3. Периодическое поведение возникает за счёт противодействия в процессе обучения двух движущих сил: weight decay, стремящегося уменьшить норму весов, и градиента, за счёт которого происходит перемещение по гиперсферам в пространстве весов модели.
4. Для корректного оценивания статистик обучения масштаб-инвариантной модели следует использовать эффективные градиент и темп обучения.

Список литературы

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

[2] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Yao Liu, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and Quoc V. Le. Symbolic discovery of optimization algorithms, 2023.

[3] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization, 2021.

[4] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns, 2018.

[5] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

[6] Ryo Karakida, Tomoumi Takase, Tomohiro Hayase, and Kazuki Osawa. Understanding gradient regularization in deep learning: Efficient finite-difference computation and implicit bias, 2023.

[7] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima, 2017.

[8] Maxim Kodryan, Ekaterina Lobacheva, Maksim Nakhodnov, and Dmitry Vetrov. Training scale-invariant neural networks on the sphere can happen in three regimes, 2023.

[9] Tanishq Kumar, Blake Bordelon, Samuel J. Gershman, and Cengiz Pehlevan. Grokking as the transition from lazy to rich training dynamics, 2023.

[10] Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks, 2021.

[11] Aitor Lewkowycz and Guy Gur-Ari. On the training dynamics of deep networks with l_2 regularization, 2020.

[12] Zhiyuan Li, Srinadh Bhojanapalli, Manzil Zaheer, Sashank J. Reddi, and Sanjiv Kumar. Robust training of neural networks using scale invariant architectures, 2022.

[13] Ziming Liu, Eric J. Michaud, and Max Tegmark. Omnigrok: Grokking beyond algorithmic data, 2023.

[14] Ekaterina Lobacheva, Maxim Kodryan, Nadezhda Chirkova, Andrey Malinin, and Dmitry Vetrov. On the periodic behavior of neural network training with batch normalization and weight decay, 2021.

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

[16] Gouki Minegishi, Yusuke Iwasawa, and Yutaka Matsuo. Grokking tickets: Lottery tickets accelerate grokking, 2023.

[17] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt, 2019.

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

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

[20] Victor Quétu and Enzo Tartaglione. Can we avoid double descent in deep neural networks?, 2023.

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