
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

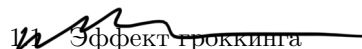
Ветров Д.П.
кандидат ф.-м. наук
Профессор НИУ ВШЭ
Исследовательская группа Байесовских методов

Abstract

Глубокие нейронные сети часто обучаются с использованием слоёв нормализации [6, 4], что позволяет стабилизировать процесс обучения и повысить точность предсказания модели. Однако иногда использование слоёв нормализации в совокупности с применением методов регуляризации может привести к возникновению необычных эффектов при обучении нейросетей. Например, в статье [3] описываются результаты использования слоёв нормализации **BatchNorm** вместе с регуляризатором «weight decay» и возникающее в процессе обучения нейросети периодическое поведение. В данной работе будут приведены результаты исследования периодического поведения при обучении нейронных сетей на примере эффекта «гроккинга» - явления, связанного с переобучением нейросетевых, в частности, трансформерных моделей и описанного в одноимённой статье [1] исследователями из OpenAI. Будет показано, что периодического поведения для эффекта гроккинга можно добиться путём использования масштабно-инвариантных архитектур. В частности, для проведения экспериментов в качестве модели будет использоваться нейросетевая архитектура, построенная на основании статей [9, 8].

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

1 Введение

 Эффект гроккинга

Обобщение перепараметризованных нейронных сетей уже давно является источником интереса для сообщества машинного обучения, поскольку оно бросает вызов интуиции, вытекающей из классической теории обучения [5, 7]. В выше упомянутой статье [1] показывается, 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. 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.

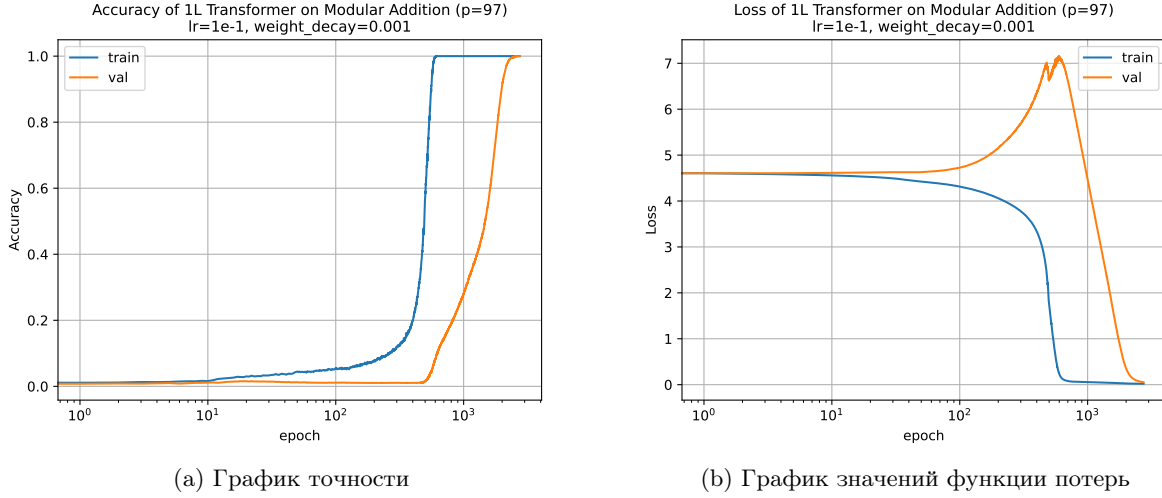


Рис. 1: Графики точности и значения функции потерь, типичные для эффекта гроккинга

В качестве модели авторами оригинальной статьи [1] 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 $(x + y) \bmod p$ или $(x * y) \bmod p$ (где x и y - целые неотрицательные числа, не превосходящие $p - 1$). 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 [1, 9].

1.2 Масштабная инвариантность

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 зависимости от расстояния до начала координат:

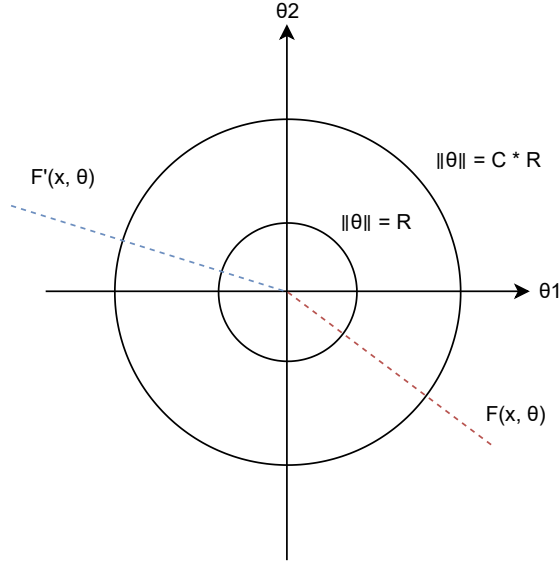


Рис. 2: Визуализация масштабной инвариантности модели для случая двумерного пространства весов. F и F' - функции, которые реализует модель вдоль соответствующих лучей. R - радиус малой окружности, C - вещественная положительная константа.

2 Постановка задачи

нужна мат по статистике и матем.

Целью данной работы является достижение периодического поведения при обучении глубокой нейронной сети на примере эффекта гроккинга посредством использования масштабно-инвариантных слоёв, а также последующий анализ статистик обучения с целью выявления причин возникновения данного шаблона поведения.

2.1 Описание данных

→ это в эксперименте.

Все эксперименты в данной работе были проведены на алгоритмически сгенерированном датасете равенств вида « $a \circ b = c$ », описанном в пункте 1.1, где в качестве бинарной операции было выбрано сложение по модулю 97: $(x + y) \bmod 97$. Единственное отличие между данными, использовавшимися в оригинальной статье, и датасетом, использовавшимся в этой работе, заключается в том, что при токенизации равенств « $a \circ b = c$ » не использовались токены для самой бинарной операции « \circ », 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.

2.2 Модель

→ эксперимент.

В качестве модели для проведения экспериментов была выбрана трансформерная архитектура, описанная в статье [9].

Расшифруем обозначения, использованные на рис.3:

- **LayerNorm** - слой нормализации
- **Attention Layer** - слой внимания (является основой трансформерной архитектуры [2])
- **MLP** - два полносвязных линейных слоя с ReLu в качестве функции активации после первого из них
- Кружок со знаком «+» внутри означает Skip Connection

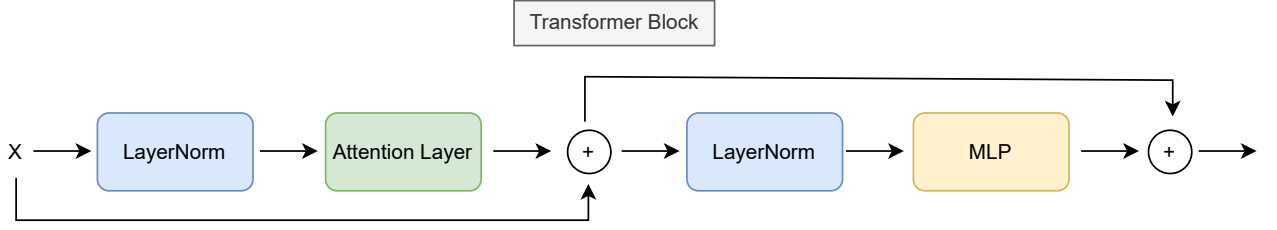


Рис. 3: Блок трансформера используемой нейросетевой архитектуры

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 3

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`).

2.3 Способ обучения

→ Эксперимент

В рамках воспроизведения эффекта гроккинга решается задача многоклассовой классификации, в которой таргетом является токен «с» (результат выполнения бинарной операции $(a + b) \bmod 97$ - 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) с постоянным темпом обучения с целью упрощения интерпретируемости 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 Эксперименты

3.1 Использование нормализации для достижения периодического поведения

В статье [3] 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. Воспользуемся данной идеей для нашей задачи и попробуем добиться периодического поведения за счёт 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, задействованным 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:

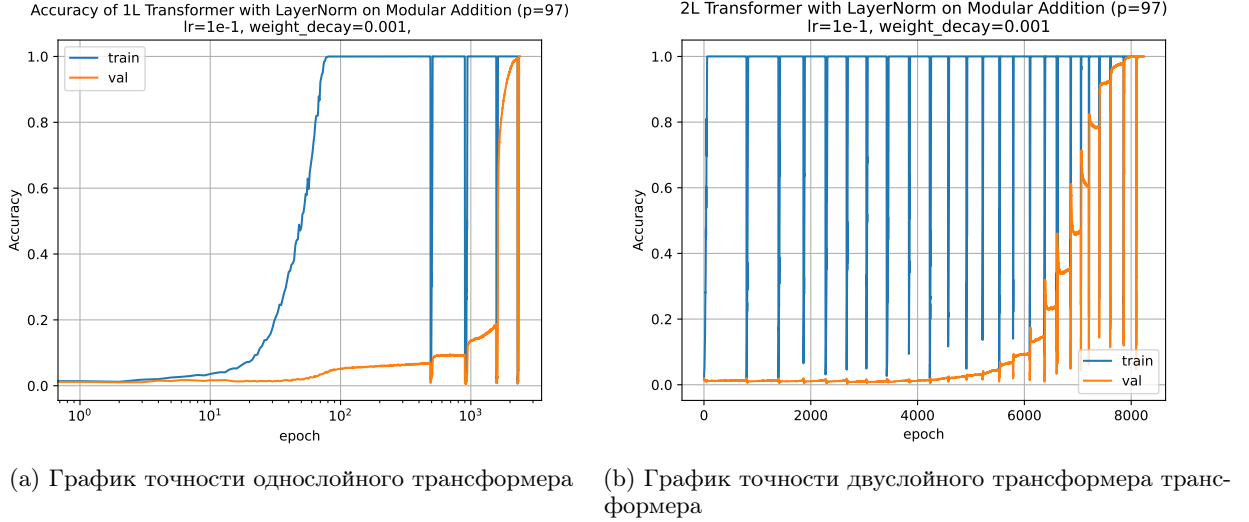


Рис. 4: Графики точности моделей трансформеров с использованием нормализации

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

3.2 Анализ статистик обучения

Для дальнейшего анализа статистик обучения необходимо ввести некоторые дополнительные понятия, которые бы учитывали масштаб-инвариантность нейросети. 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.

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

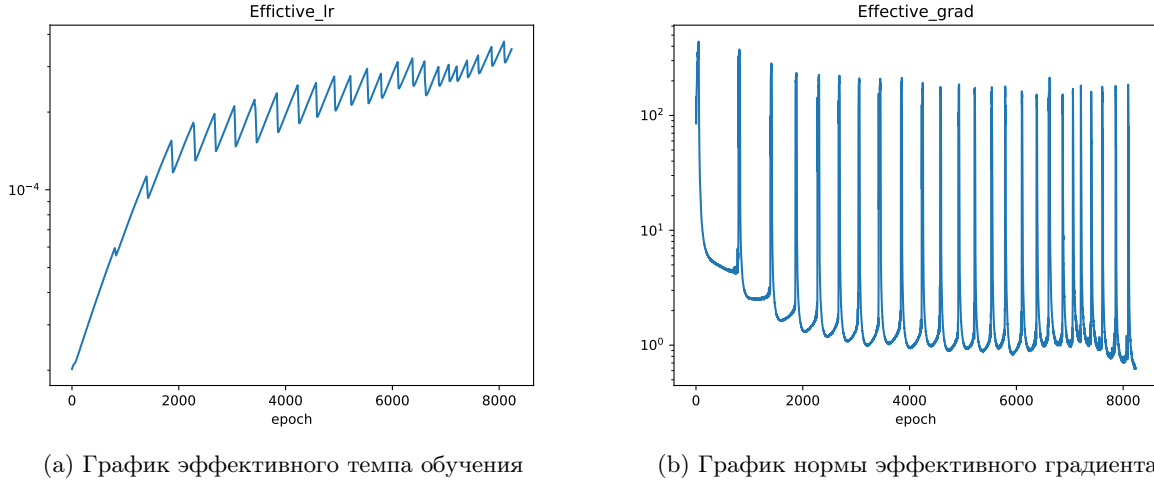


Рис. 5: Графики эффективных статистик двуслойного трансформера с использованием LayerNorm

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).

+ Babushkin.
Список литературы

- [1] Harri Edwards Igor Babuschkin Vedant Misra Alethea Power, Yuri Burda. Grokking: Generalization beyond overfitting on small algorithmic datasets. arXiv:2201.02177, 2022.
- [2] Niki Parmar Jakob Uszkoreit Llion Jones Aidan N. Gomez Lukasz Kaiser Illia Polosukhin Ashish Vaswani, Noam Shazeer. Attention is all you need. arXiv:1706.03762, 2017.
- [3] Nadezhda Chirkova Andrey Malinin Dmitry Vetrov Ekaterina Lobacheva, Maxim Kodryan. On the periodic behavior of neural network training with batch normalization and weight decay. arXiv:2106.15739, 2021.
- [4] Geoffrey E. Hinton Jimmy Lei Ba, Jamie Ryan Kiros. Layer normalization. arXiv:1607.06450, 2016.
- [5] Yamini Bansal Tristan Yang Boaz Barak Ilya Sutskever Preetum Nakkiran, Gal Kaplun. Deep double descent: Where bigger models and more data hurt. arXiv:1912.02292, 2019.
- [6] Christian Szegedy Sergey Ioffe. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167, 2015.
- [7] Enzo Tartaglione Victor Quétu. Can we avoid double descent in deep neural networks? arXiv:2302.13259, 2023.
- [8] Manzil Zaheer Sashank J. Reddi Sanjiv Kumar Zhiyuan Li, Srinadh Bhojanapalli. Robust training of neural networks using scale invariant architectures. arXiv:2202.00980, 2023.
- [9] Max Tegmark Ziming Liu, Eric J. Michaud. Omnigrok: Grokking beyond algorithmic data. arXiv:2210.01117, 2022.

Маг
220