Периодическое поведение при обучении глубоких нейронных сетей на примере эффекта гроккинга

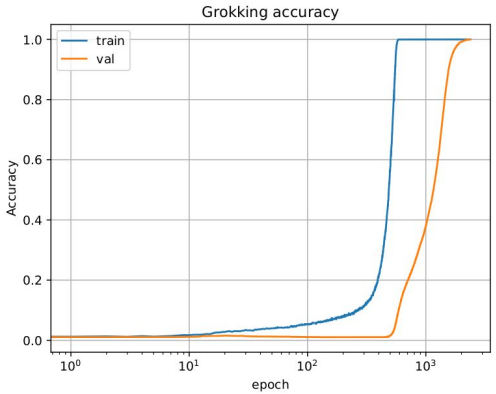
Мельник Юрий Максимович

Кафедра ММП, факультет ВМК
МГУ им. Ломоносова

Научный руководитель: кандидат ф.-м. наук Ветров Д.П
Научный консультант: Южаков Т.А.

Эффект гроккинга

- Продолжение обучения переобученной модели может привестит к росту точности на тестовой выборке
- При обучении необходимо использовать регуляризатор, в частности weight decay
- Модель обучается на алгоритмически сгенерированный датасет равенств вида «a o b = c», где «a», «b», «c» - целые неотрицательные числа, а «o» - бинарная операция (x + y) mod 97



★	a	b	c	d	e
a	a	d	?	c	d
b	c	d	d	a	c
c	?	e	d	b	d
d	a	?	?	b	c
e	b	b	c	?	a

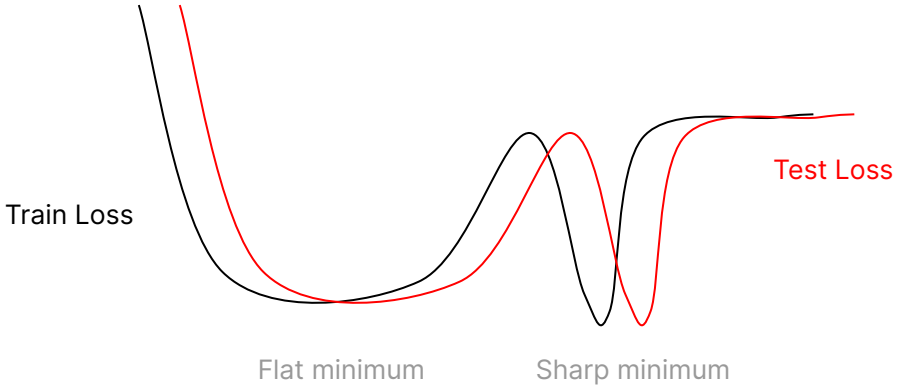
График точности характерный для эффекта “гроккинга” (левый рисунок)
Синтетические данные для задачи “гроккинга” (правый рисунок)

Кривизна оптимумов функции потерь и её связь с возникновением гроккинга

- У функции потерь различаются “широкие” и “узкие” минимумы, обладающие разным уровнем генерализации
- В процессе обучения стремимся находить более “широкие” оптимумы
- В качестве метрики “ширины” минимума можно использовать среднюю норму стохастического градиента

Гипотеза:

- Зазор между точностью на обучающей и тестовых выборках в гроккинге возникает из-за того, что вначале мы попадаем в “узкий” минимум, но в процессе дальнейшего обучения всё же находим более “широкий”

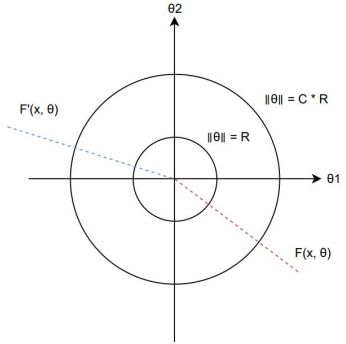


Периодическое поведение и его связь со свойством масштабной инвариантности

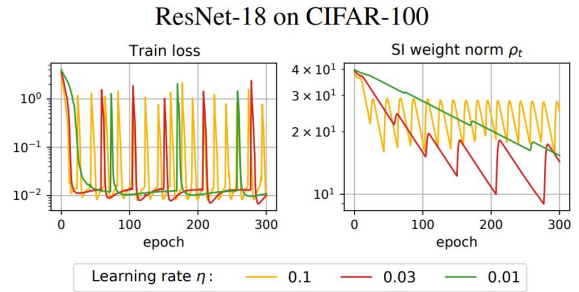
- Пусть log p(y|x, θ) - предсказание нейросетевой модели. Тогда модель называется масштабно-инвариантной по весам, если выполняется следующее соотношение:

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

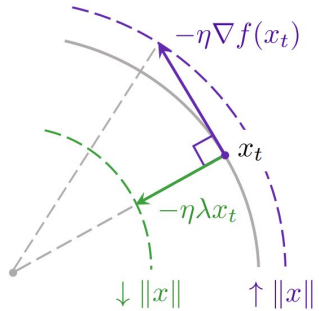
- Периодическое поведение возникает за счёт противодействия в процессе обучения двух движущих сил - weight decay и градиента



Визуализация свойства масштабной инвариантности



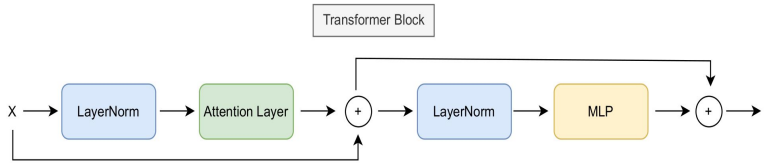
Пример периодического поведения при обучения нейросети из статьи “On the Periodic Behavior …”



Взаимное расположение weight decay и градиента для масштабно-инвариантной сети

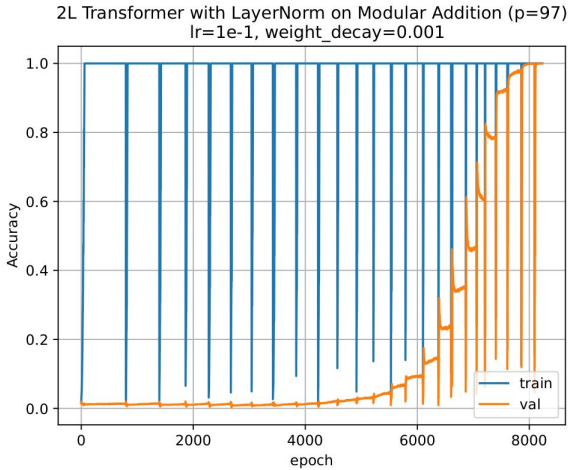
Базовый эксперимент

- Была обучена масштабно-инвариантная трансформерная модель
- Удалось добиться ожидаемого периодического поведения
- На основании графика нормы эффективного градиента можно сказать, что в процессе обучения, действительно, скатились в более “широкий” минимум из более “узкого”

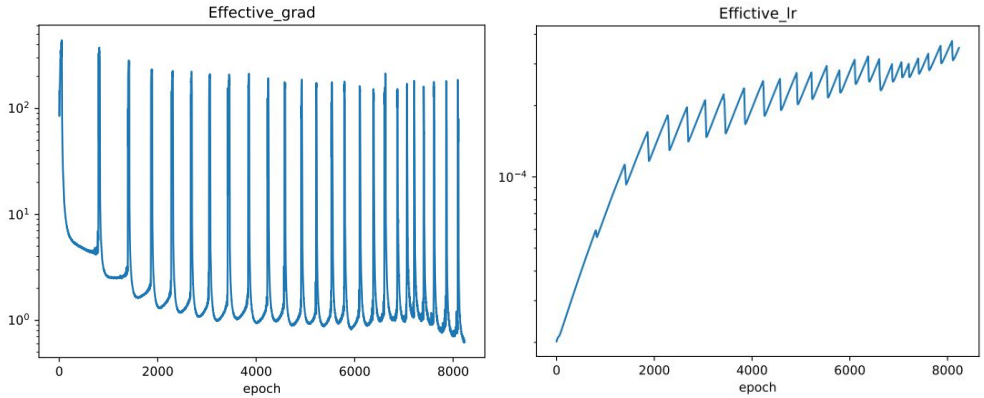


Трансформерная архитектура, использовавшаяся в экспериментах

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



Периодическое поведение при обучении модели в задаче “гроккинга”



Графики эффективных градиента (нормы) и лёрнинг рейта

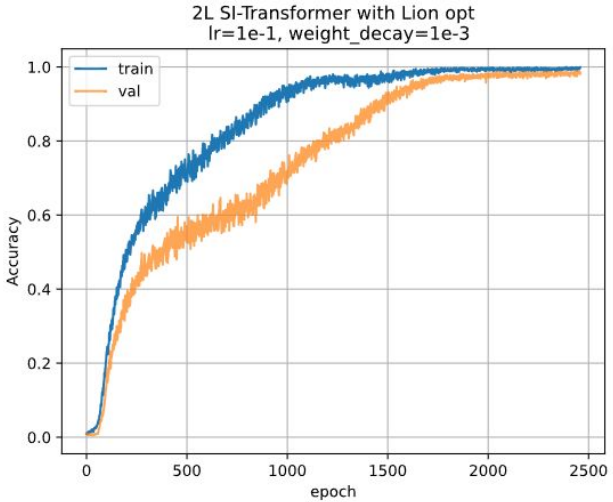
Эксперимент с оптимизатором Lion

- Модель была обучена с помощью оптимизатора Lion
- Эффект гроккинга перестал наблюдаться
- В процессе обучения Lion “нащупал” нужный learning rate, засчёт чего удалось избежал попадания в узкий минимум

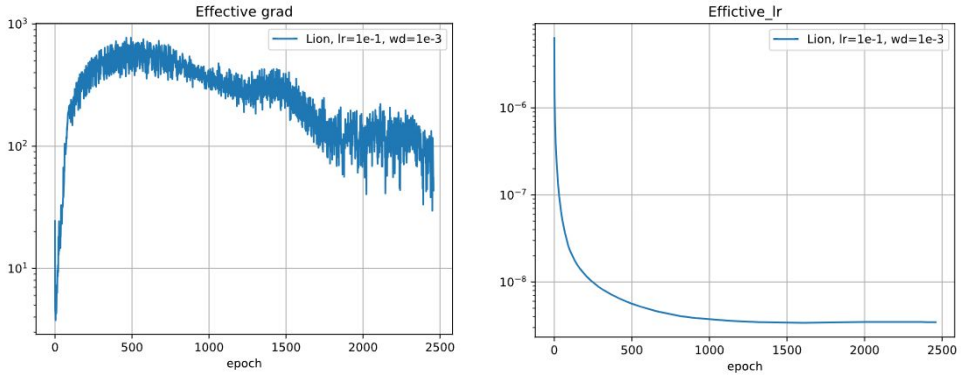
Algorithm 2 Lion Optimizer (ours)

given $\beta_1, \beta_2, \lambda, \eta, f$
initialize $\theta_0, m_0 \leftarrow 0$
while θ_t not converged **do**
  $g_t \leftarrow \nabla_\theta f(\theta_{t-1})$
  update model parameters
  $c_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$
  $\theta_t \leftarrow \theta_{t-1} - \eta_t(\text{sign}(c_t) + \lambda\theta_{t-1})$
  update EMA of g_t
  $m_t \leftarrow \beta_2 m_{t-1} + (1-\beta_2) g_t$
end while
return θ_t

Алгоритм оптимизации Lion



Отсутствие эффекта гроккинга при использовании Lion



Графики эффективных градиента (нормы) и лёрнинг рейта

Выводы и дальнейшие планы

1. Периодическое поведение возникает засчёт 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.

2. Подбор learning rate влияет на разрешающую способность модели - способность “видеть” широкие и узкие минимумы

3. Оптимизатор Lion как бы подбирает значение learning rate для каждого веса в отдельности (засчёт процедуры оптимизации). Идея: подбирать learning rate не для каждого веса, а для групп параметров.

4. Сконструировать на основе идеи Lion оптимизатор, который бы подбирал значение learning rate для каждой группы параметров.