The Lottery Ticket Hypothesis: Finding Sparce, Trainable Neural Networks

Maxim Kobelev, 161
NRU HSE Research Seminar

План

- Введение

- Описание и формулировка Lottery Hypothesis
- Важные моменты, discussion авторов
- Мотивация и дальнейшие ожидания

- Работа с полносвязными сетями (LeNet 300-100-10 on MNIST)
- Работа со свёрточными сетями (Conv-2, Conv-4, Conv-6 on CIFAR10)
- Работа с большими сетями (VGG, ResNet on CIFAR10)

- Заключение

Проблемы современности

- Имеет место тенденция увеличения числа параметров сети для достижения наилучшего качества

- Вместе с увеличением числа параметров сети растет и время её обучения

- Текущие подходы позволяют урезать такие нейронные сети вплоть до 90%, уменьшая их размер и ускоряя «прямой проход» по ним

- Опыт показывает, что обучение с нуля урезанных сетей происходит сложнее, достигается меньшее качество, чем в оригинальной сети.

Почему мы не обучаем эти уменьшенные архитектуры, ведь это быстрее?

Lottery Ticket Hypothesis

Полносвязная случайно инициализированная сеть содержит в себе подсеть меньшего размера (winning ticket), которая переобучаясь заново с той же инициализацией, достигает качества, сравнимого с оригинальным, за то же число шагов.

Более формально

$$f(x;\theta) \qquad \theta = \theta_0 \sim \mathcal{D}_\theta$$

with SGD optimisation f reaches minimum validation loss ***L*** on ***j***-th iteration with test set accuracy ***a***

$$f(x; m \odot \theta) \qquad m \in \{0,1\}^{|\theta|}$$

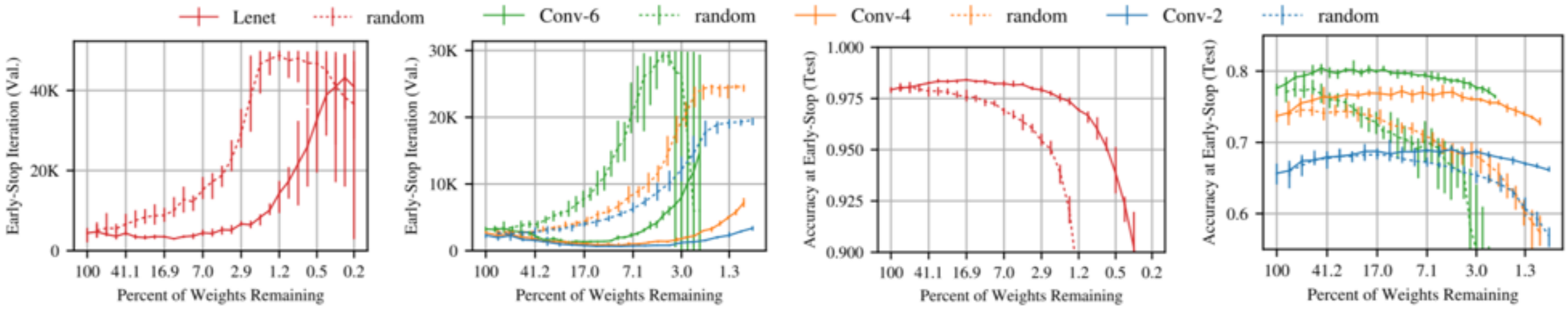
with SGD optimisation f reaches minimum validation loss ***L'*** on ***j'***-th iteration with test set accuracy ***a'***

Lottery Ticket Hypothesis:

$$\exists\, m : \quad \begin{aligned} j' &\leq j \\ a' &\geq a \\ \|m\|_0 &\ll |\theta| \end{aligned}$$

Постановка экспериментов

- Случайная инициализация весов нейронной сети $f(x;\theta_0)$
- Обучаем ***j*** итераций и приходим к θ_j
- Урезаем ***p**% параметров в* θ_j , создавая маску ***m***
- Возвращаем веса к исходным инициализированным из θ_0, получая на выходе winning ticket $f(x;m\odot\theta_0)$



One-Shot и Iterative Pruning

- Такой подход, описанный выше подходит под определение **One-Shot**. Урезается ***p%*** весов, оставшиеся - сбрасываются к начальным.

- **Iterative Pruning** - повторяем итерации обучения и урезания сети на протяжении ***n*** раундов, таким образом в каждый следующий раунд передаются $p^{\frac{1}{n}}\%$ весов.

- Было показано, что **Iterative Pruning** техника позволяет искать *winning tickets* гораздо меньших размеров, имеющих сравнительное с оригиналом качество, чем **One-Shot**.

Назад к мотивации

- Получилось показать, что урезание сетей находит подсети меньшего размера, достигающие качество сравнимое с оригиналом, за сравнимое с его числом итераций временем обучения.

- Обещающая способность таких *winning ticket-подсетей* выше, как и точность на валидации, как и скорость обучения, по сравнению с оригинальной сетью.

- Предлагается использовать гипотезу как метод поиска базовых моделей для их применения в ансамблях.

Ожидания

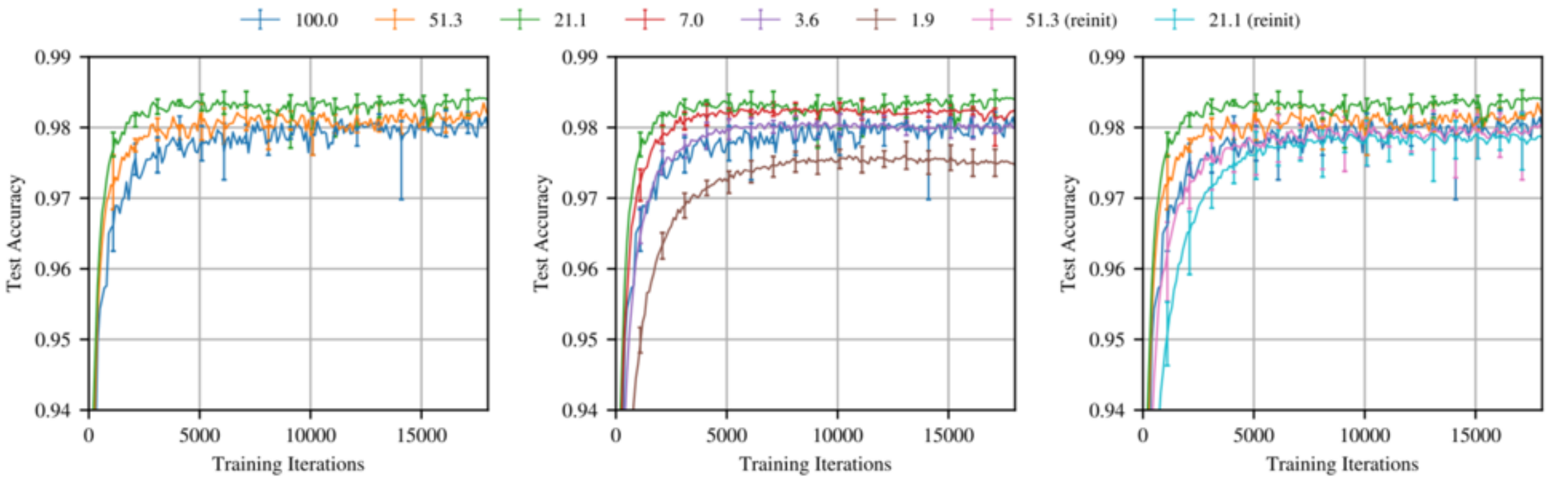
- Из-за необходимости переобучения winning-ticket’s с начала, необходимо разработать метод поиска этих подсетей на как можно более ранних этапах.

- Дизайн новых архитектур и схем инициализации весов, с использованием опыта построения таких разреженных подсетей

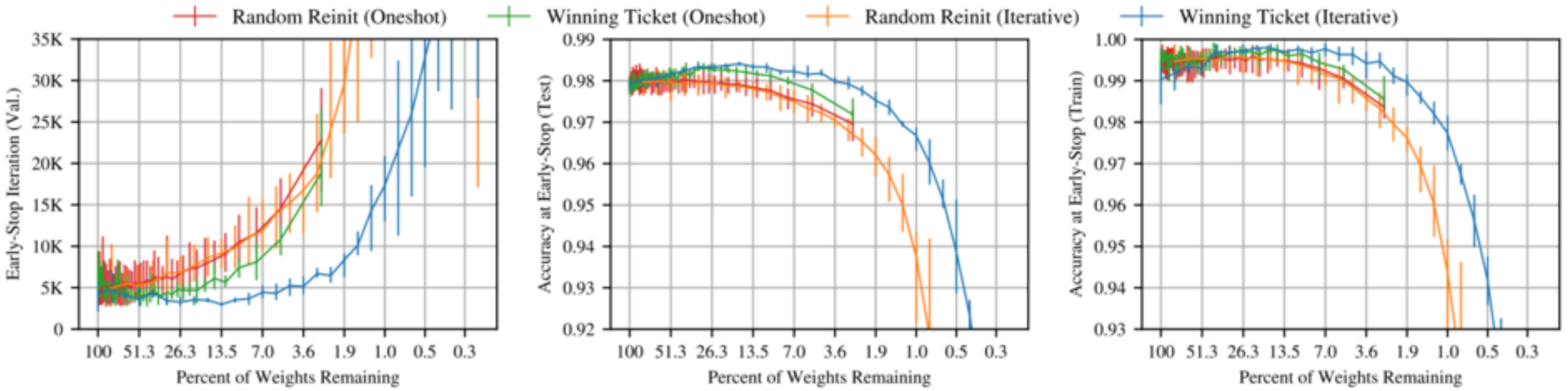
- Улучшение наших теоретических знаний и понимения нейросетей.

Эксперименты.

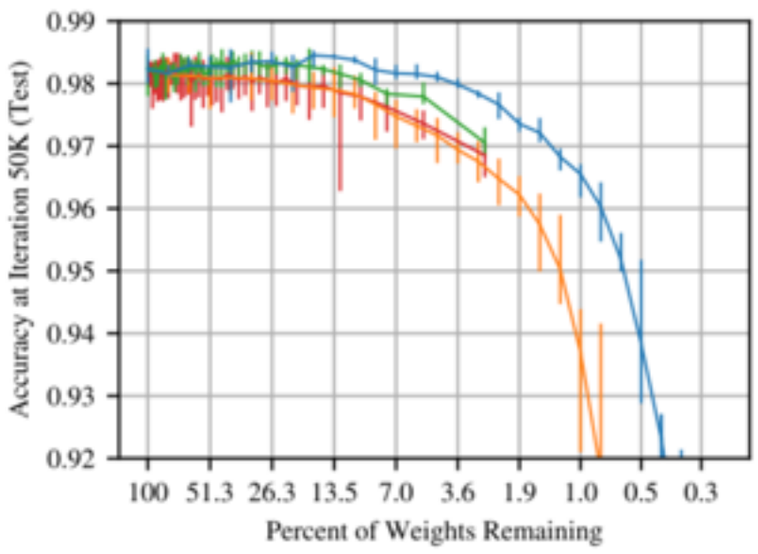
Network	Lenet	Conv-2	Conv-4	Conv-6	Resnet-18	VGG-19
Convolutions		64, 64, pool	64, 64, pool 128, 128, pool	64, 64, pool 128, 128, pool 256, 256, pool	16, 3x[16, 16] 3x[32, 32] 3x[64, 64]	2x64 pool 2x128 pool, 4x256, pool 4x512, pool, 4x512
FC Layers	300, 100, 10	256, 256, 10	256, 256, 10	256, 256, 10	avg-pool, 10	avg-pool, 10
All/Conv Weights	266K	4.3M / 38K	2.4M / 260K	1.7M / 1.1M	274K / 270K	20.0M
Iterations/Batch	50K / 60	20K / 60	25K / 60	30K / 60	30K / 128	112K / 64
Optimizer	Adam 1.2e-3	Adam 2e-4	Adam 3e-4	Adam 3e-4	← SGD 0.1-0.01-0.001 Momentum 0.9 →	
Pruning Rate	fc20%	conv10% fc20%	conv10% fc20%	conv15% fc20%	conv20% fc0%	conv20% fc0%



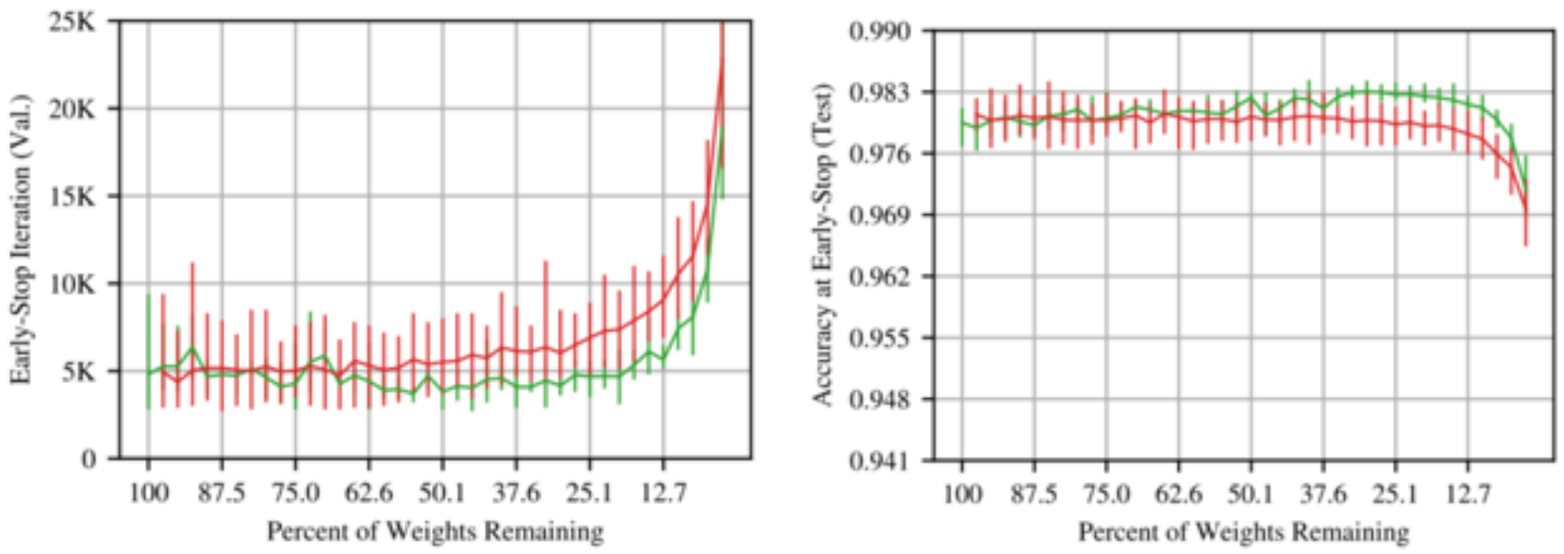
LeNet 300-100-10 on MNIST



(a) Early-stopping iteration and accuracy for all pruning methods.

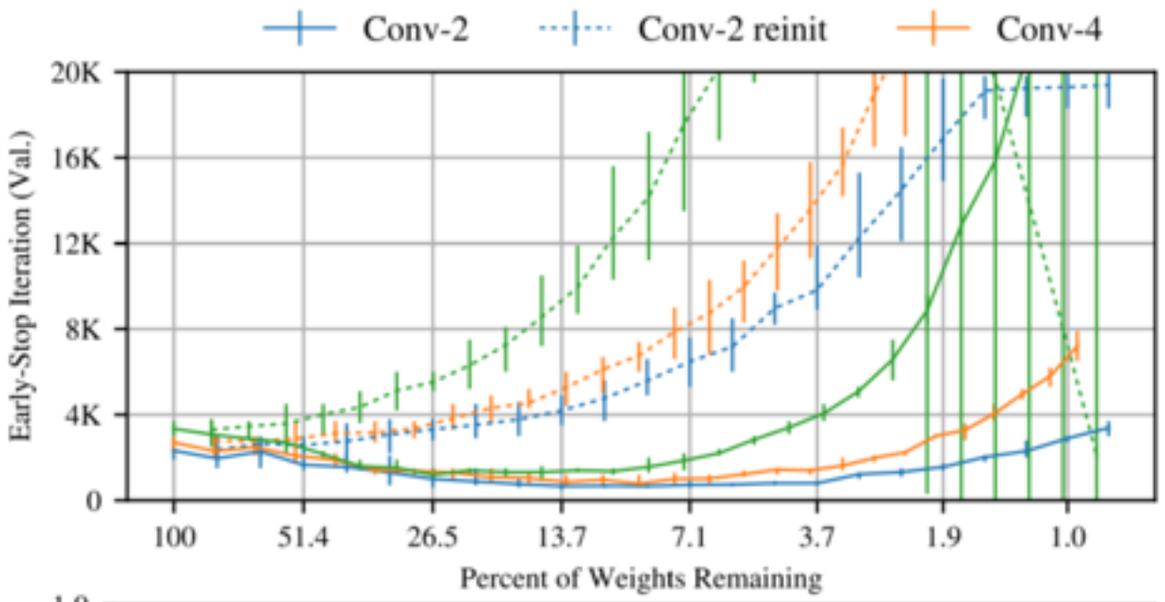


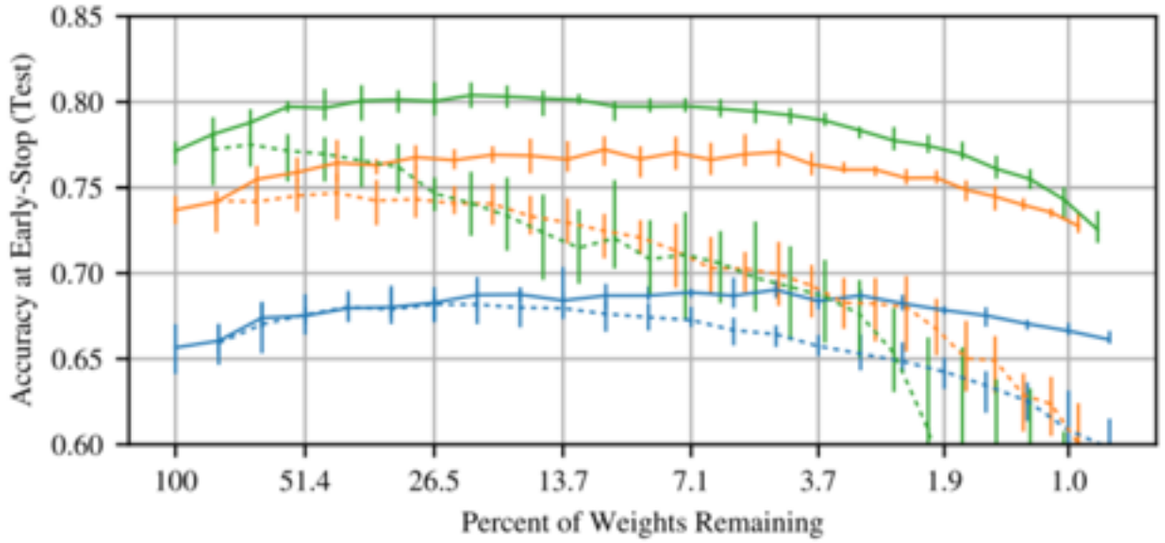
(b) Accuracy at end of training.

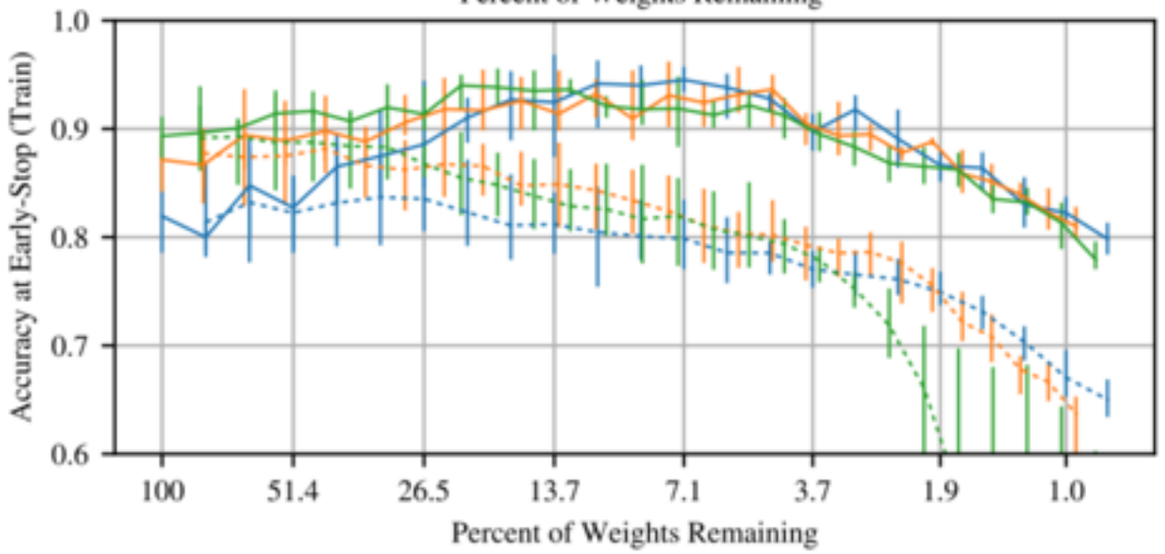


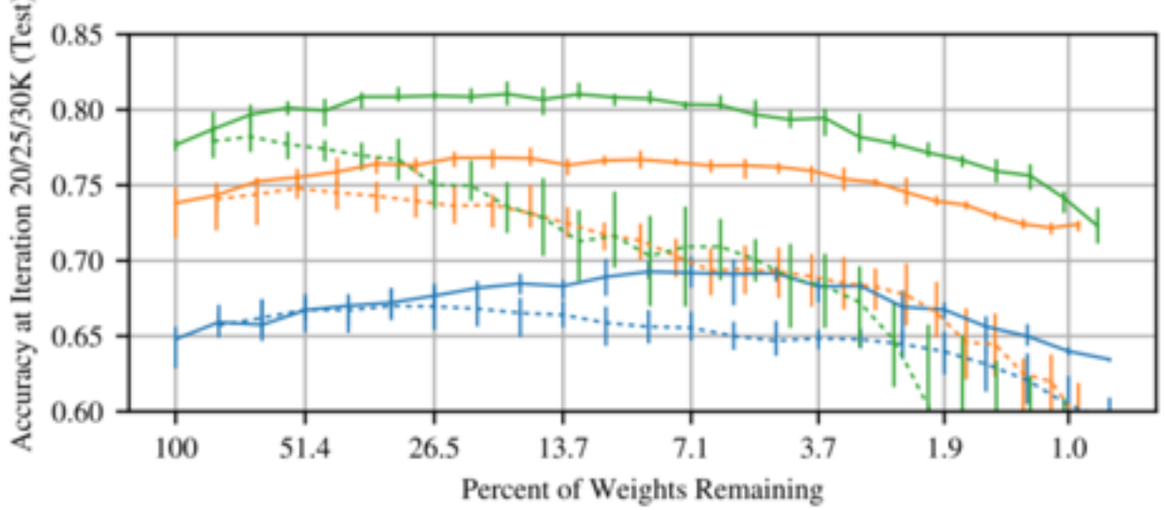
(c) Early-stopping iteration and accuracy for one-shot pruning.

One-Shot vs Iterative
Random ReInit vs Winning Ticket









Conv-2/4/6 with iteratively pruned and random reinitialised.

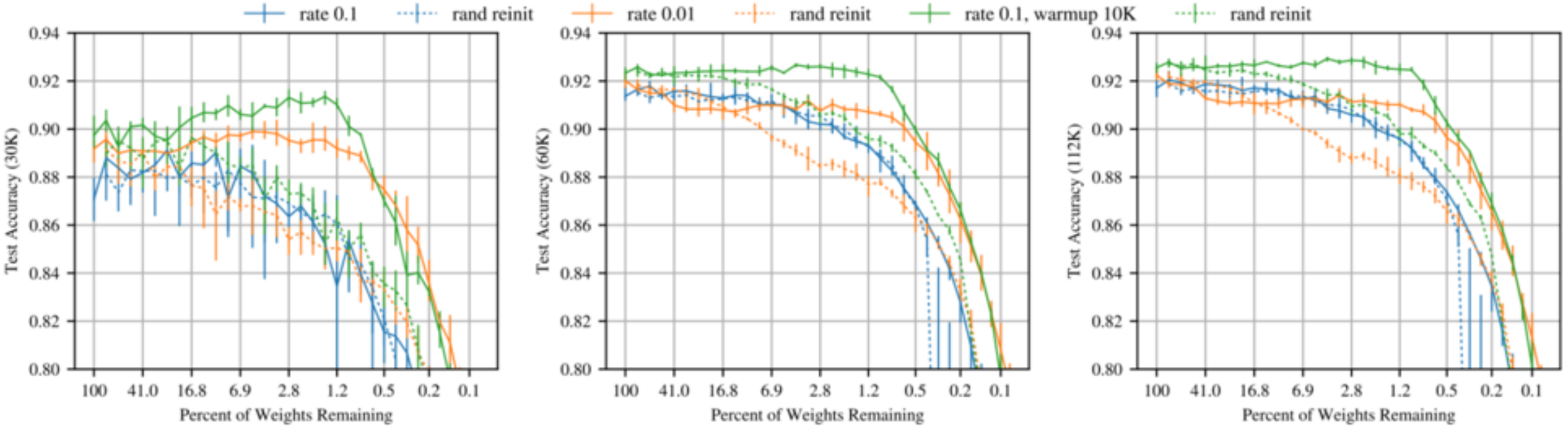


Figure 7: Test accuracy (at 30K, 60K, and 112K iterations) of VGG-19 when iteratively pruned.

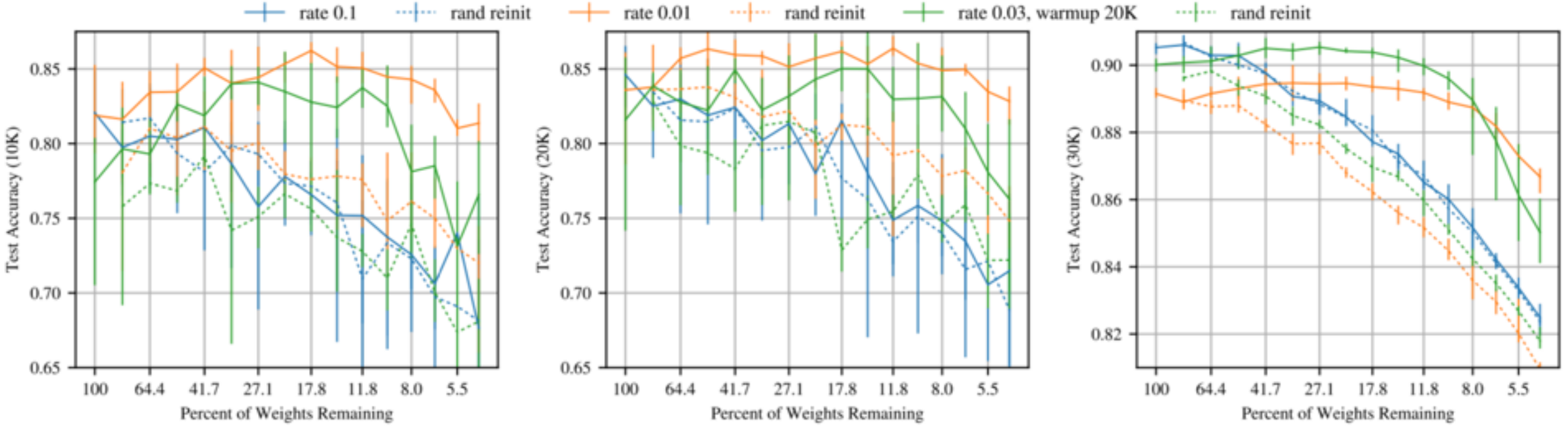


Figure 8: Test accuracy (at 10K, 20K, and 30K iterations) of Resnet-18 when iteratively pruned.

Выводы

- Текущие знания показывают что функции приближаемые нейронными сетями могут быть представлены с гораздо более меньшим числом параметров.

- Найденная гипотеза описывает механизм выделения таких подмножеств параметров.

- Описанный метод урезания нейронных сетей работает одинаково хорошо как на полносвязных, так и на сверточных сетях.

- При необходимости иметь сильно разреженную по весам модель *Iterative Pruning >> One-Shot Pruning.*

Вопросы:

- Что такое *winning-ticket* в контексте *lottery ticket hypothesis*
- Сформулируйте *lottery ticket hypothesis*
- Опишите алгоритм нахождения *winning-ticket-a.*
- К каким последствиям приводит *random re-initialization* весов.