

# THE LOTTERY TICKET HYPOTHESIS: FINDING SPARSE, TRAINABLE NEURAL NETWORKS

# Постановка задачи

Современные методы «стрижки» нейросетей позволяют уменьшить количество параметров более чем на 90%.

Однако в большинстве случаев получившаяся в итоге нейронная сеть не покрывает все компоненты исходной, необходимые для быстрого обучения.

Необходимо попытаться найти в исходной сети такую подсеть, что за одинаковое количество итераций добивалась бы точности сопоставимой с исходной.

# The Lottery Ticket Hypothesis

- Пусть у нас есть сеть  $f(x; \theta)$  с изначальными параметрами  $\theta = \theta_0 \sim D_\theta$ . Определим подсеть следующим образом:  $f(x; m \odot \theta_0)$ , где  $m$  - маска:  $m \in \{0,1\}^\theta$
- Гипотеза:  $\exists m : j' \leq j, \alpha' \leq \alpha, ||m||_0 \ll |\theta|$
- где  $j$  - количество итераций,  $\alpha$  - точность.

# Identifying winning tickets (one-shot)

1. Случайным образом инициализируем сеть  $f(x; \theta_0)$
2. Обучим на  $j$  итерациях, достигнув параметра  $\theta_j$
3. Исключим  $p\%$  параметров из  $\theta_j$  (жадно), создав маску  $m$ .
4. Заменим оставшиеся параметры на  $\theta_0$  и создадим выигрышный билет  $f(x; m \odot \theta_0)$

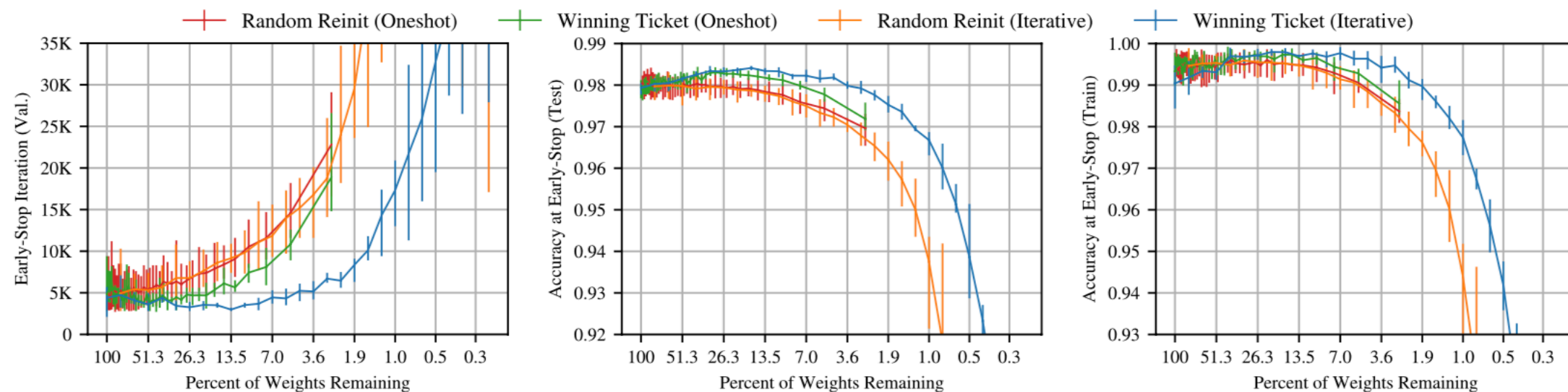
# Iterative pruning

- Будем повторять one-shot pruning  $n$  раз, каждый раз обнуляя  $p^{1/n} \%$  весов, оставшихся после предыдущего этапа.
- Экспериментально показано, что итеративный подход позволяет находить выигрышные билеты меньшего размера, чем one-shot.

# Параметры применения выигрышных билетов в полносвязных сетях.

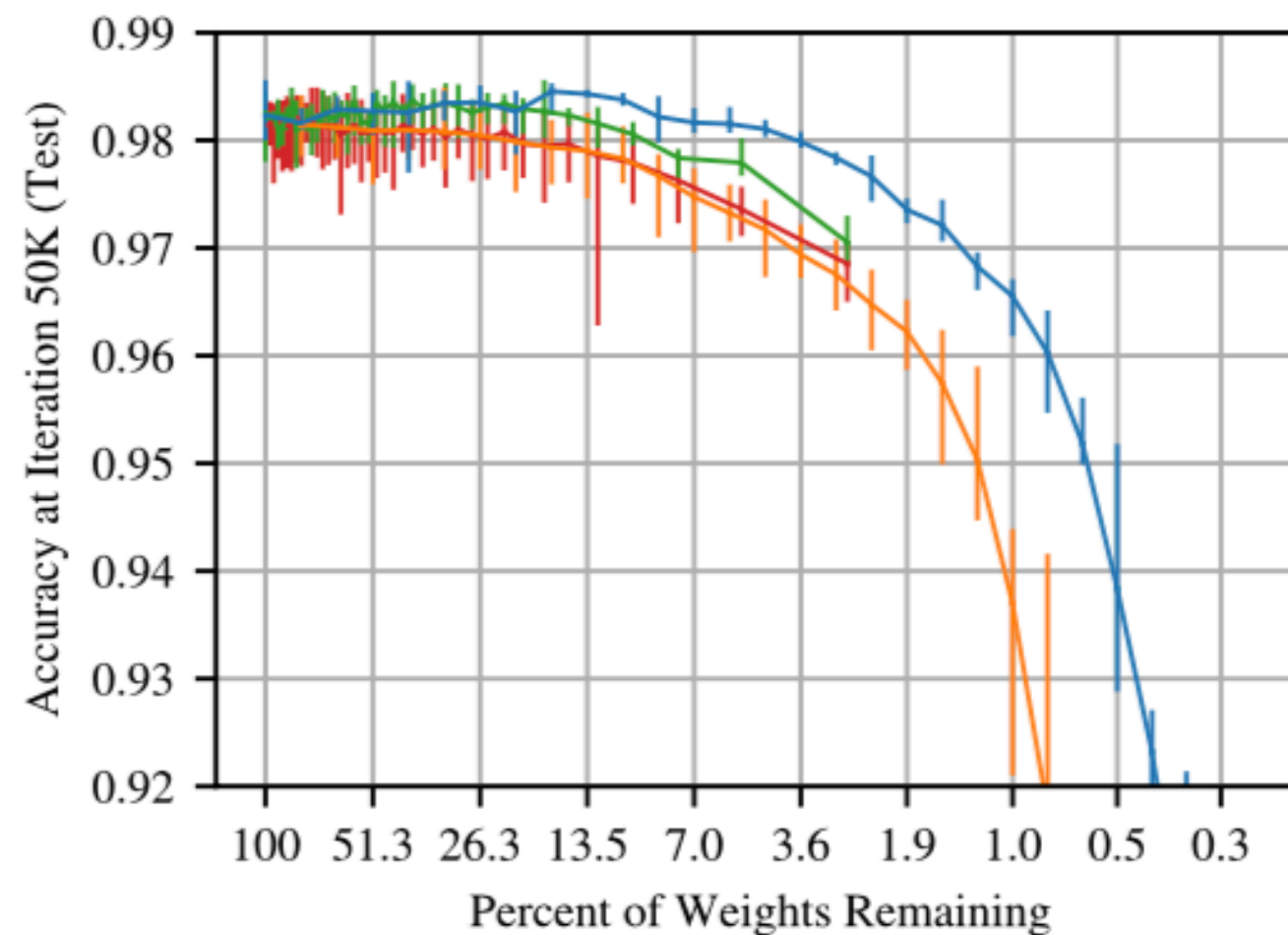
- LeNet 300-100 (2 скрытых слоя размеров 300 и 100 соответственно, на выходе 10 - так как MNIST)
- Соединения с выходным слоем обрезаются вполовину в сравнении с остальными
- Learning rate 0,0012
- Pruning rate: 0.2
- Optimization: SGD

# Выигрышные билеты в полносвязных нейросетях



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

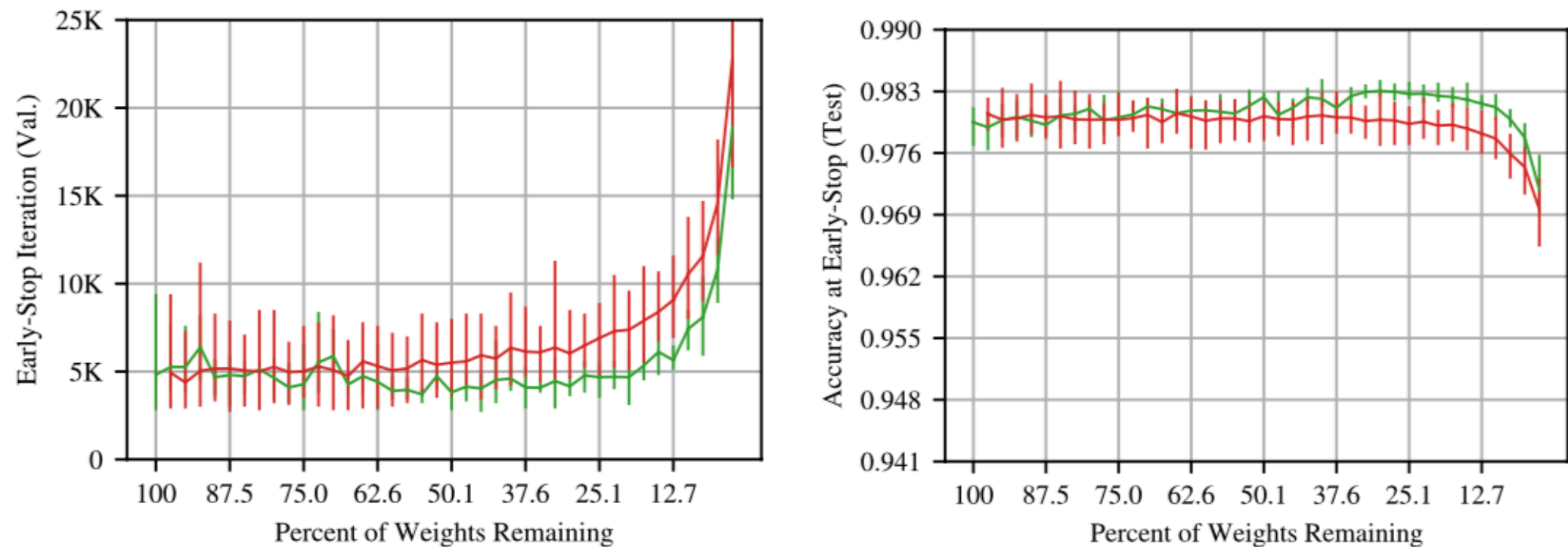
# Выигрышные билеты в полносвязных нейросетях



(b) Accuracy at end of training.



# Выигрышные билеты в полносвязных нейросетях



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

# CNN архитектура

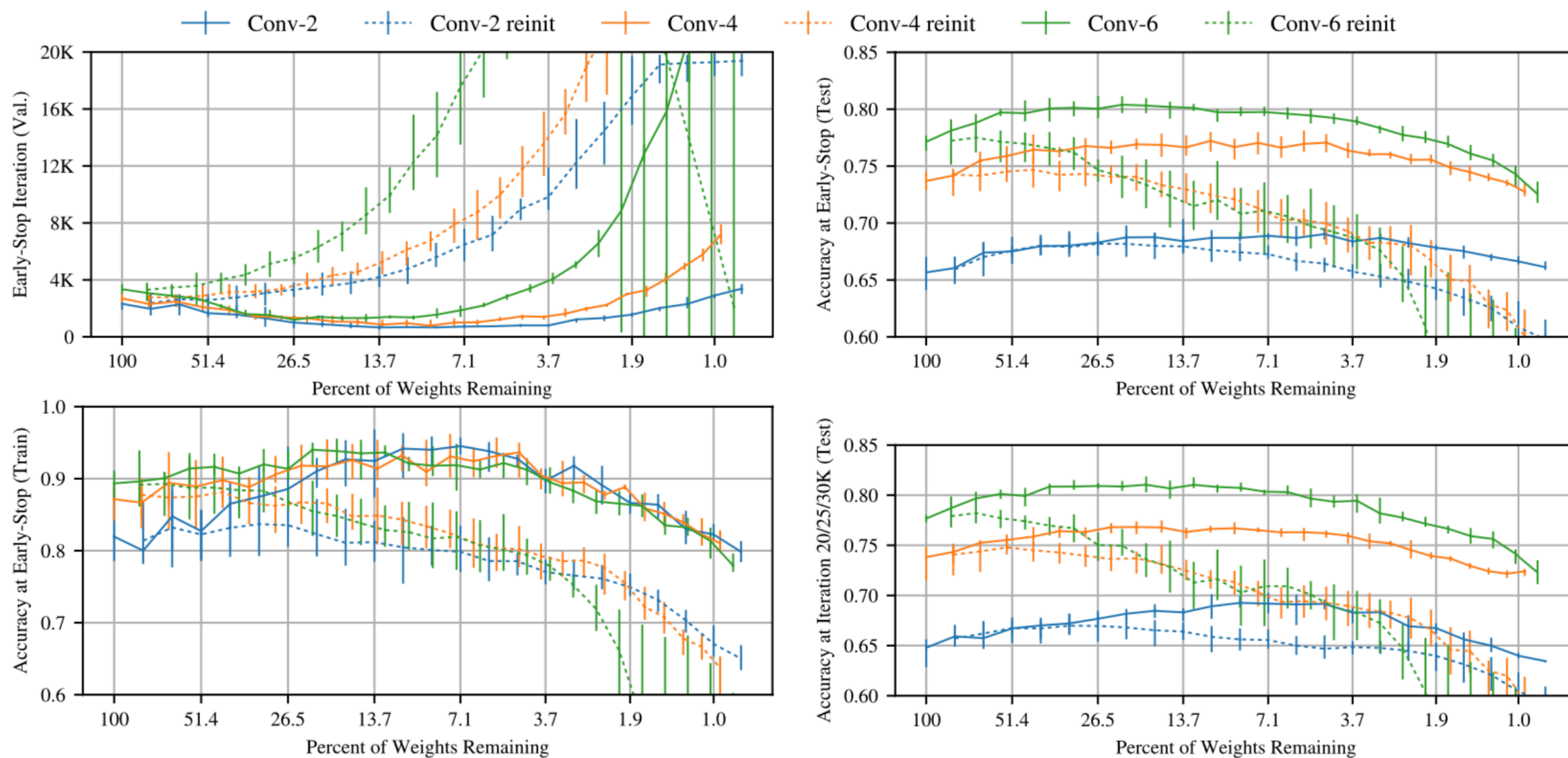
- Используются свертки с 2,4, 6 сверточными слоями
- После каждых 2 сверточных слоев применяется max-pooling
- Перед выходом нейросети применяются 2 полносвязных слоя

.

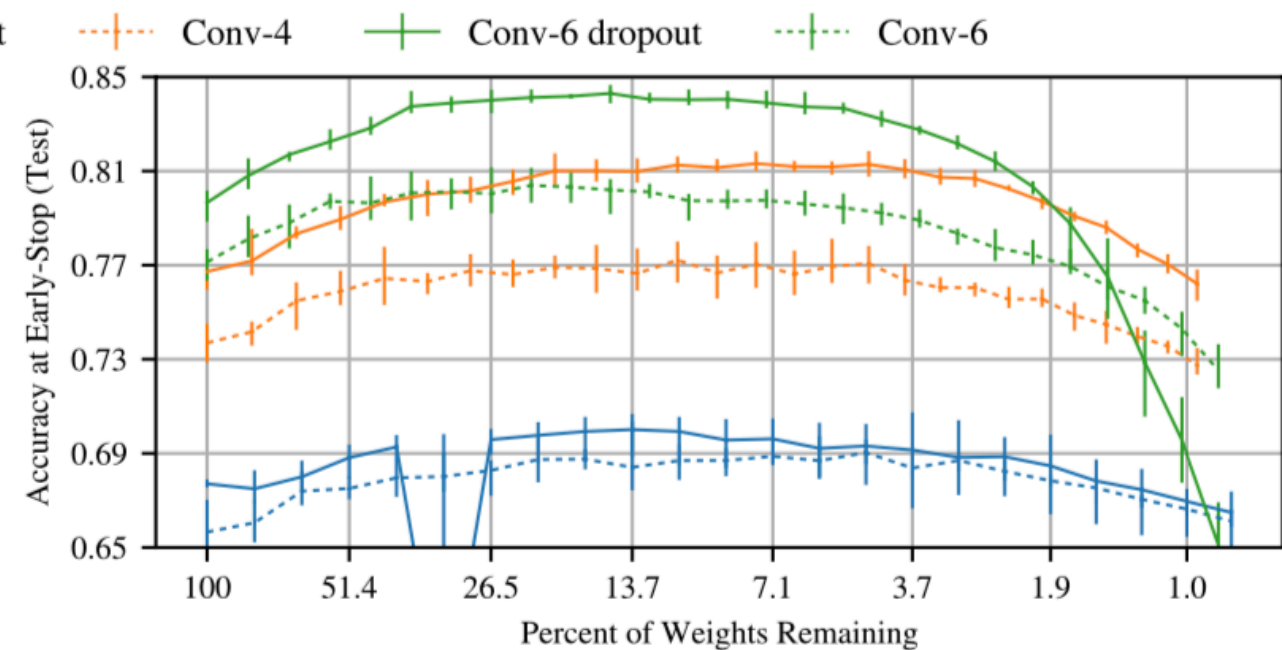
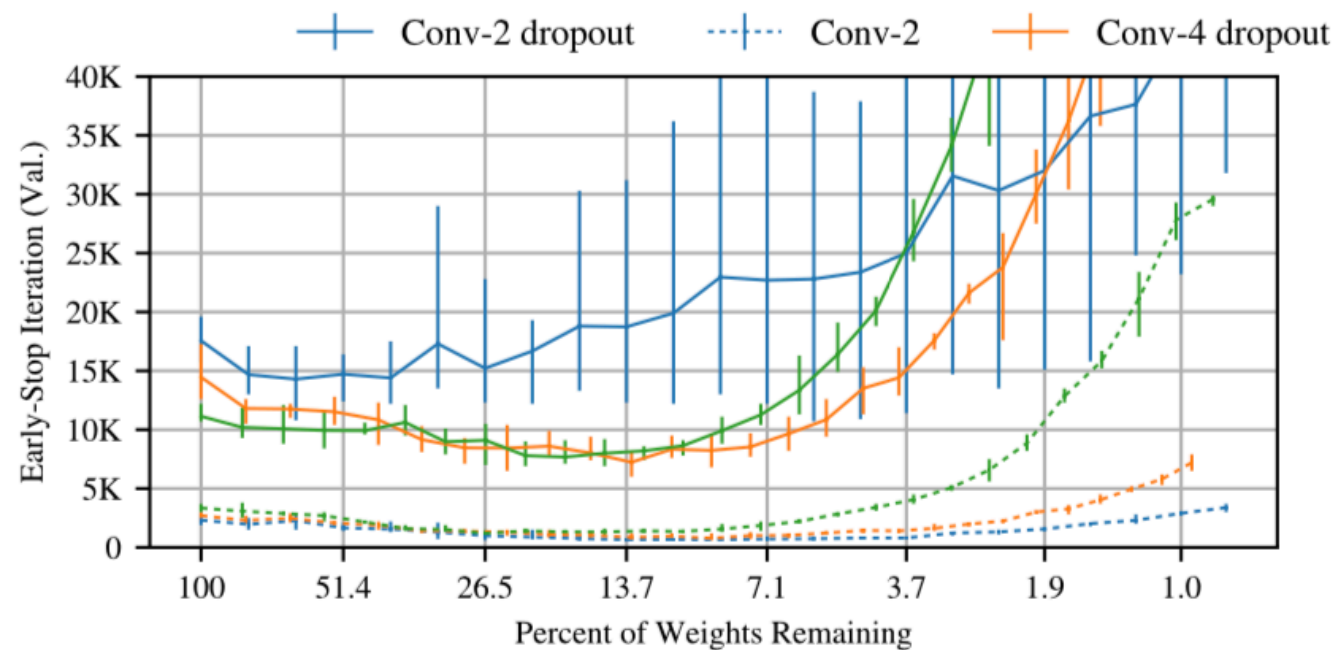
# CNN параметры

- Learning rate: CONV2 - 0.0002, CONV4&CONV6 - 0.0003
- Pruning Rate: CONV2 & CONV4 - 10%, CONV6 - 15%
- Learning Rate for dropout: 0.0003
- Эксперименты проводились на CIFAR10 (50,000 32\*32\*3 - обучающая выборка, 10000 - test)

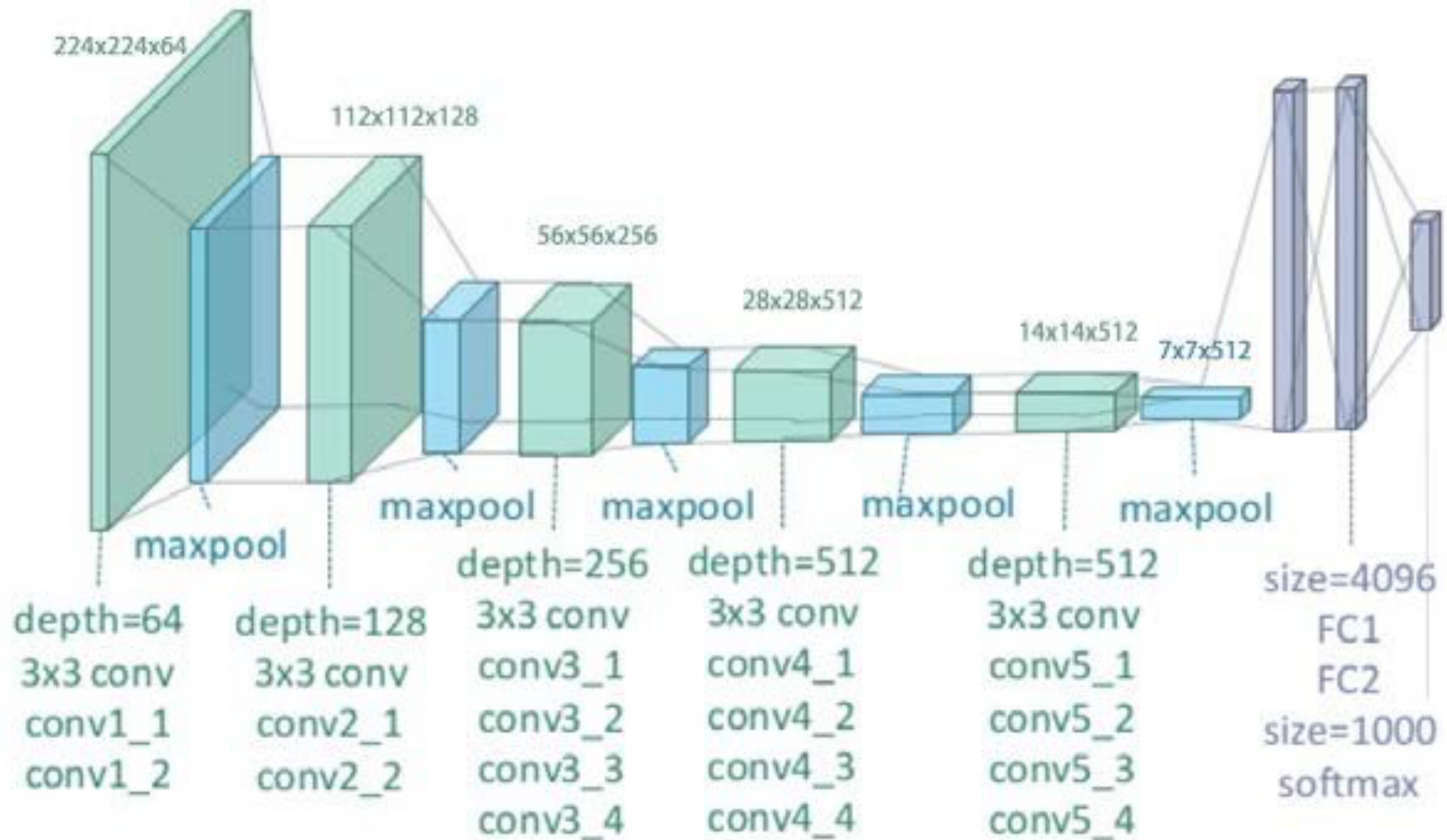
# CNN



# CNN with dropout



# VGG (напоминание)



# VGG результаты

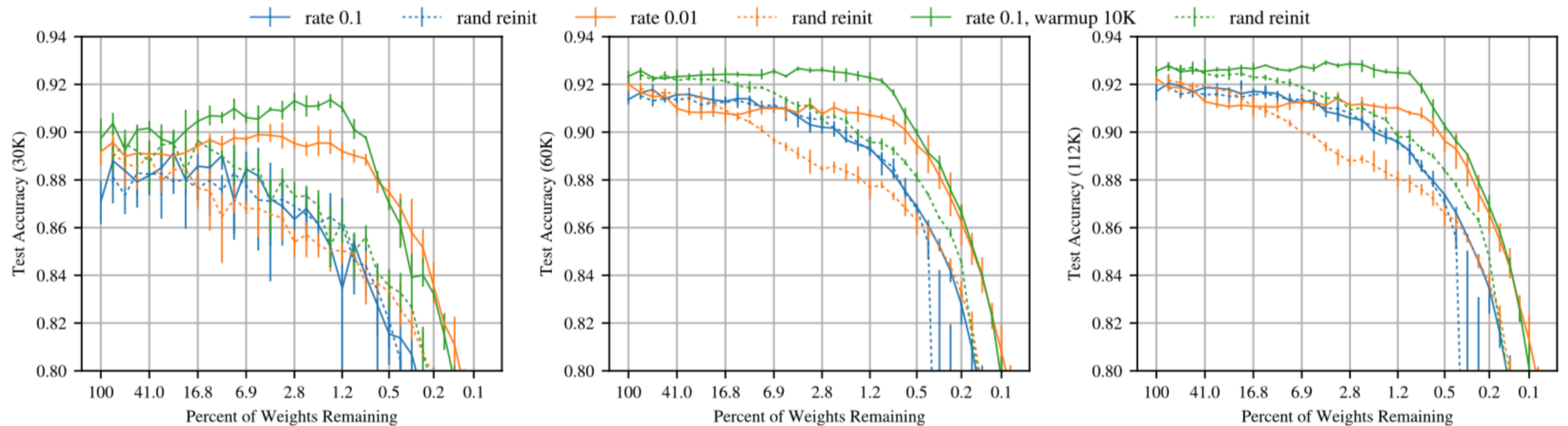


Figure 7: Test accuracy (at 30K, 60K, and 112K iterations) of VGG-19 when iteratively pruned.

# Важные свойства выигрышных билетов

- Важность корректной инициализации
- Важность структуры
- Улучшенная обобщающая способность
- Применение для нейросетевой оптимизации



# Будущие направления исследований

- Разработка менее ресурсоемкого алгоритма (итерационный подход вычислительно сложный)
- Попробовать другие способы «стрижки» более оптимизированные под современные библиотеки.
- Выяснить, почему при обучении в глубоких сетях (Resnet-18, VGG-19) не удастся найти выигрышный билет без проведения «разогрева» (warmup)

# Общие результаты

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

# ИСТОЧНИКИ

- [https://www.researchgate.net/figure/Illustration-of-the-network-architecture-of-VGG-19-model-conv-means-convolution-FC-means\\_fig2\\_325137356](https://www.researchgate.net/figure/Illustration-of-the-network-architecture-of-VGG-19-model-conv-means-convolution-FC-means_fig2_325137356)
- <https://arxiv.org/pdf/1803.03635.pdf>