The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks

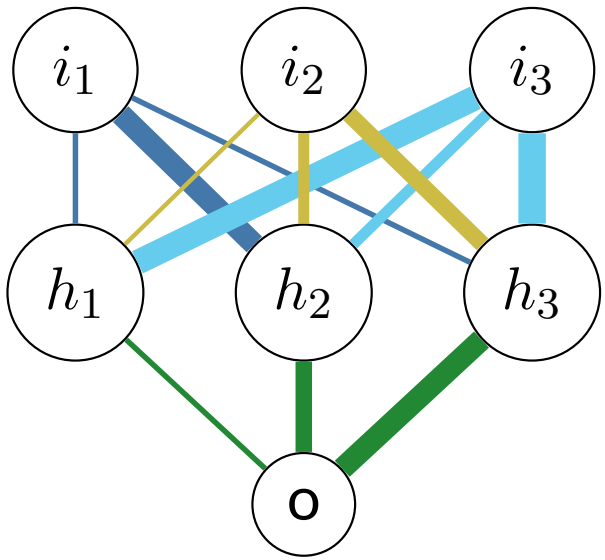
Benno Fünfstück

ChatGPT Paper Reading Group

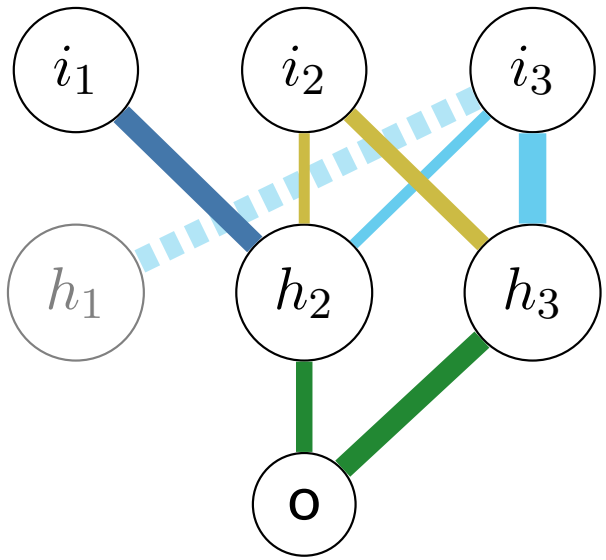
Do we need all those weights?

Oftentimes, $<$ **10%** of weights needed for same accuracy
(for models used in 2019 for computer-vision tasks)

trained network



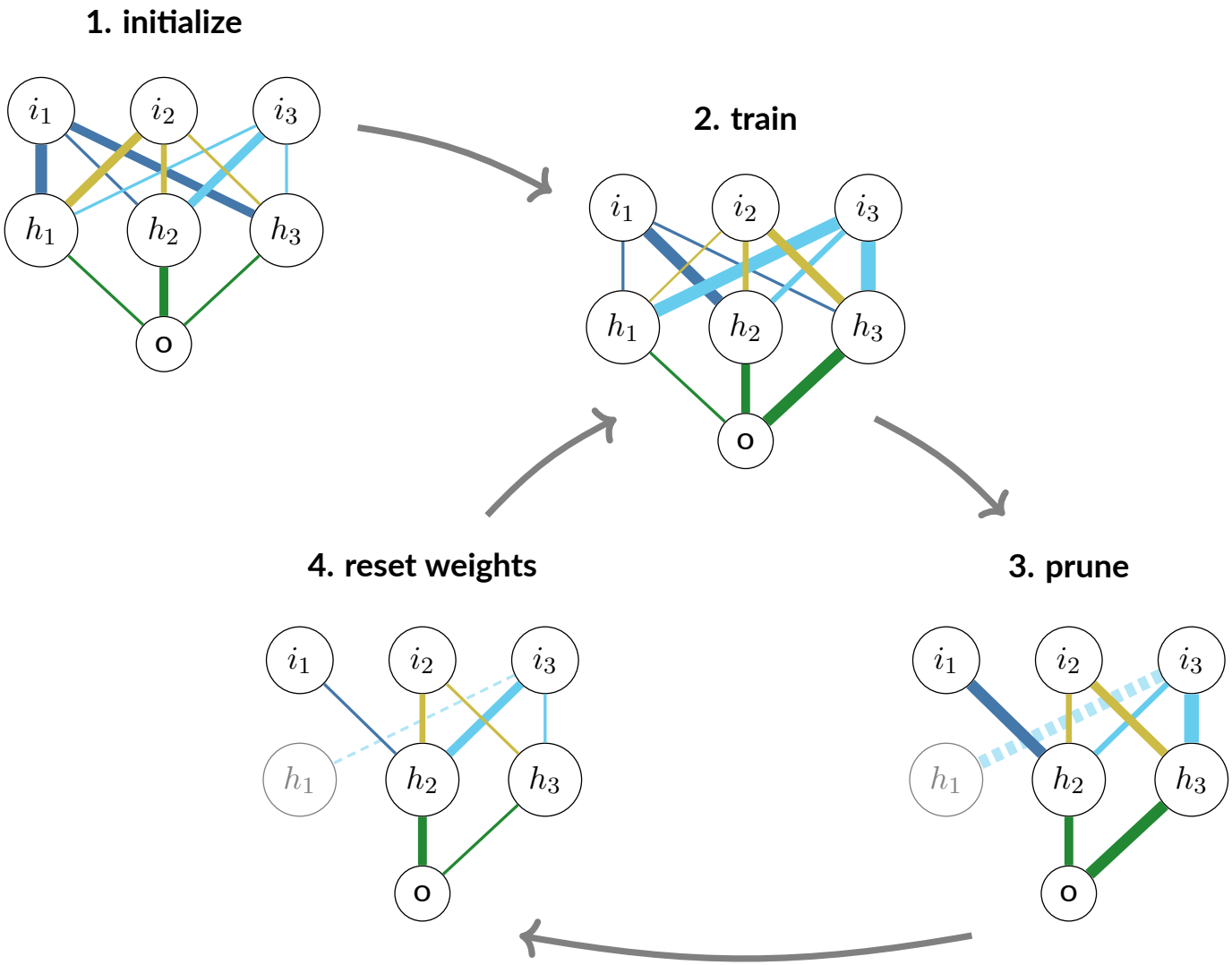
pruned network



The lottery ticket hypothesis

``*A randomly-initialized, dense neural network contains a* ***subnetwork*** *that is* ***initialized*** *such that—when* ***trained in isolation****—it can* ***match the test accuracy*** *of the original network after training for at most the same number of iterations.*''

How to find lottery tickets



Empirical validation

task: classifying 28x28 pixel images into digits (0-9) [MNIST]
model: feedforward with dimensions 300-100-10 ("LeNet")

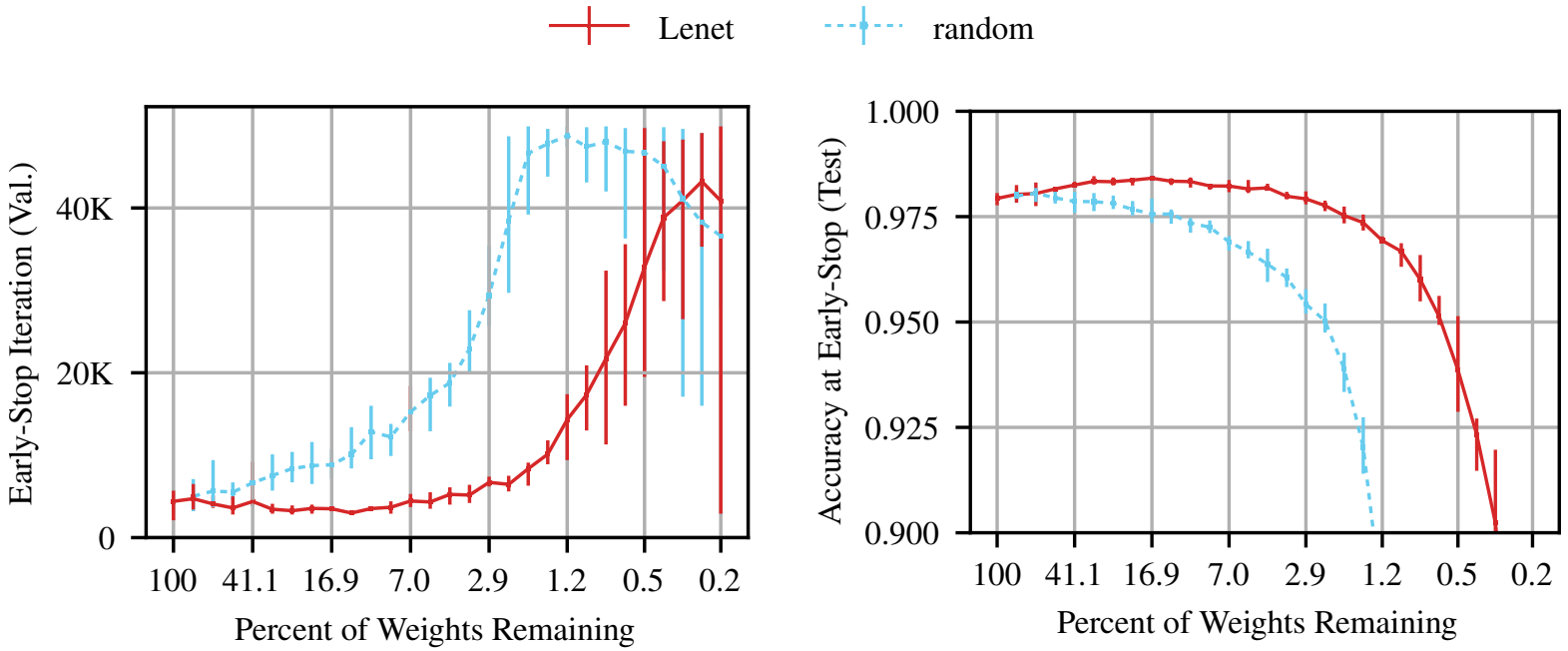


Figure 1 (page 2): average of 5 trials for winning ticket, 10 trials for random

Empirical validation

task: classifying 28x28 pixel images into digits (0-9) [MNIST]
model: feedforward with dimensions 300-100-10 ("LeNet")

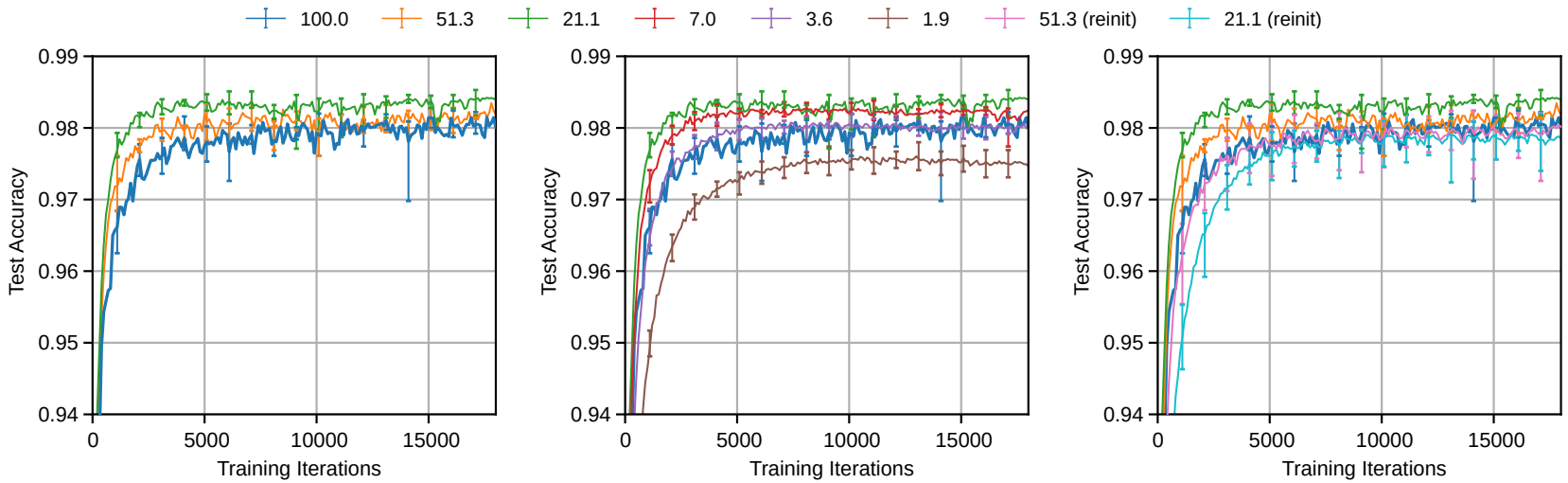


Figure 3 (page 4): test accuracy, each curve is average of 5 trials

The MNIST dataset

60k training samples
10k test samples

each sample is 28x28
8bit grayscale image
of a digit

The MNIST dataset

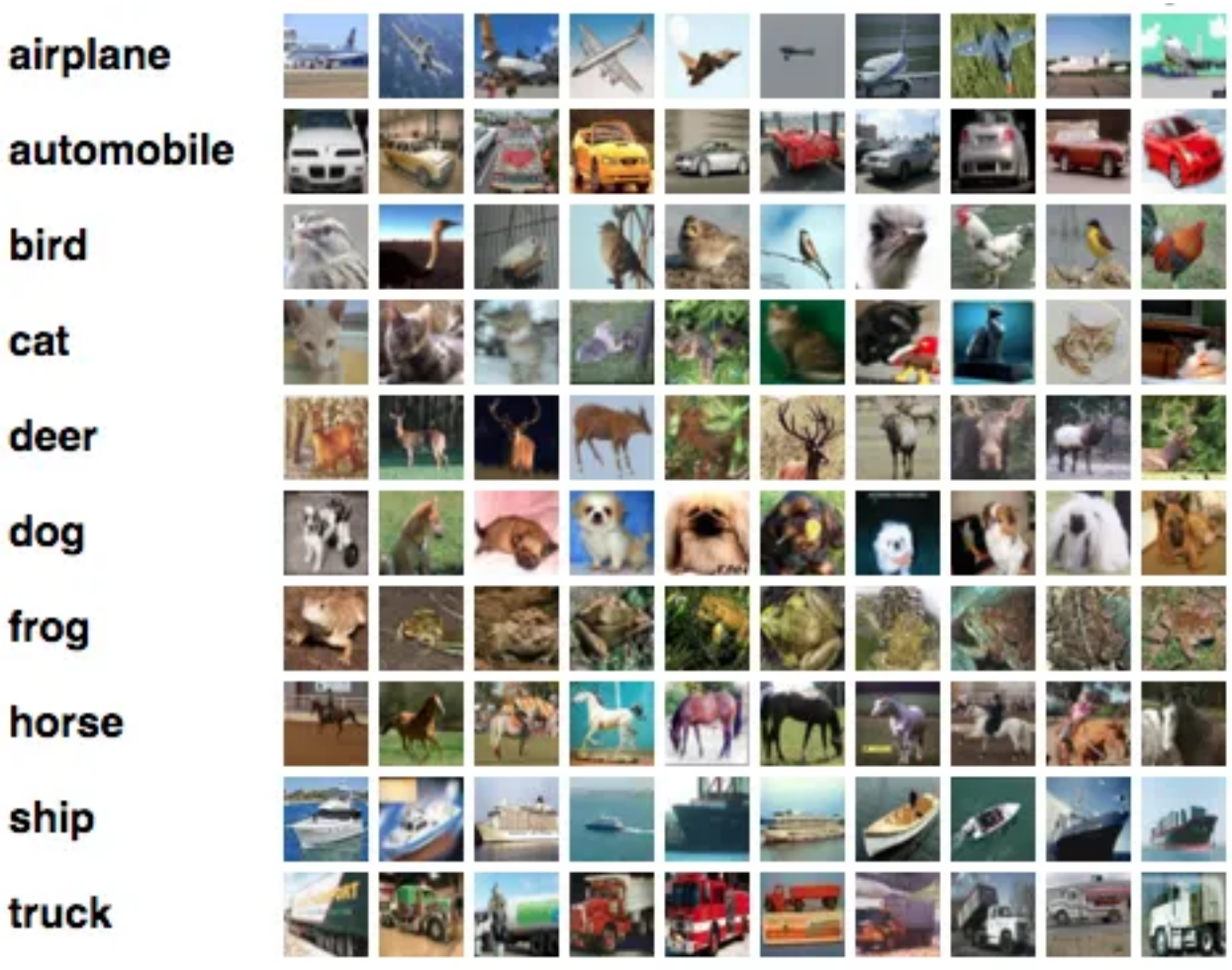
60k training samples
10k test samples

each sample is 28x28
8bit grayscale image
of a digit

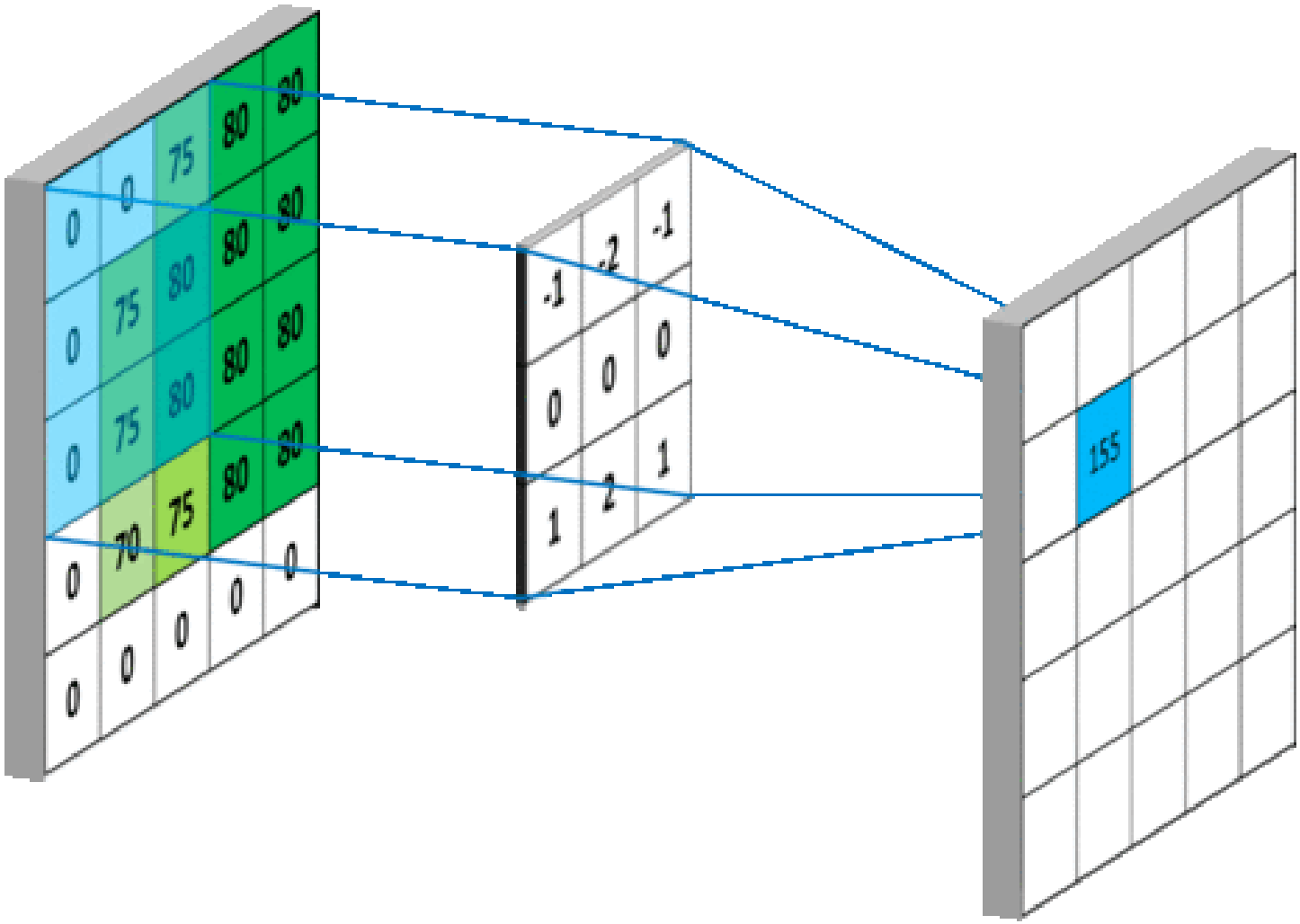
simple linear classifier
reaches 91% accuracy

The CIFAR10 dataset

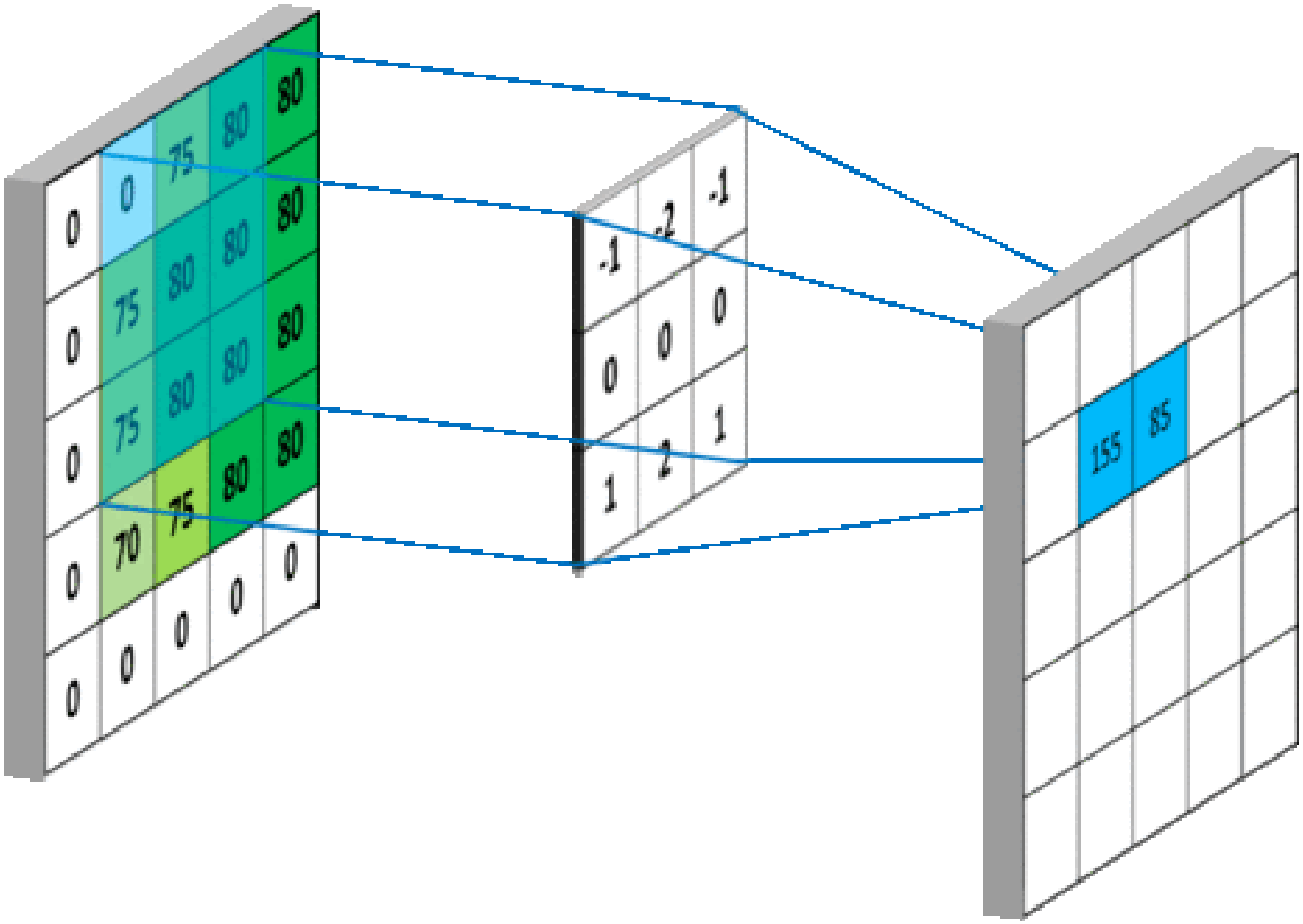
60k training samples, 10k test samples, 32x32 colour images



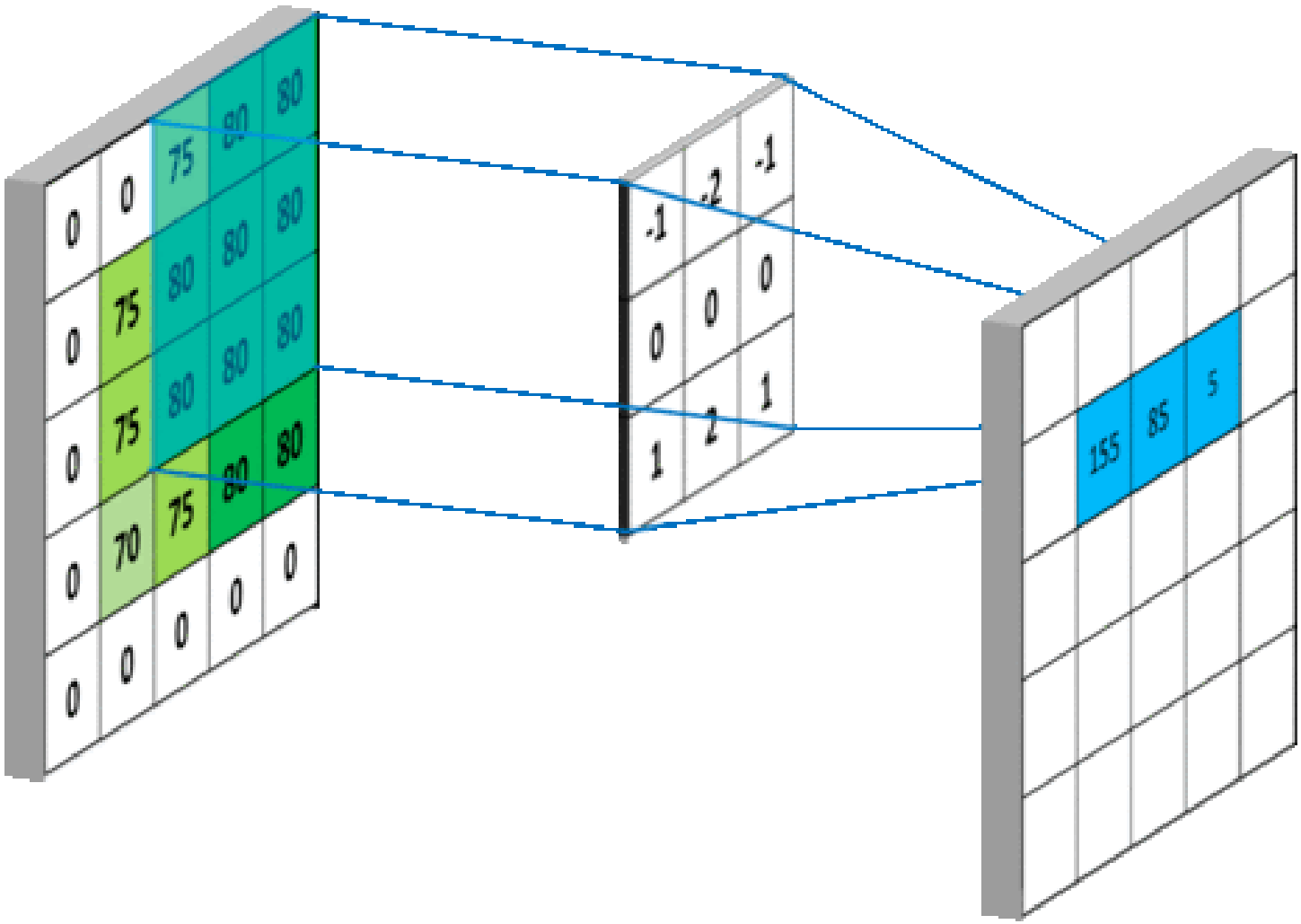
Convolutions



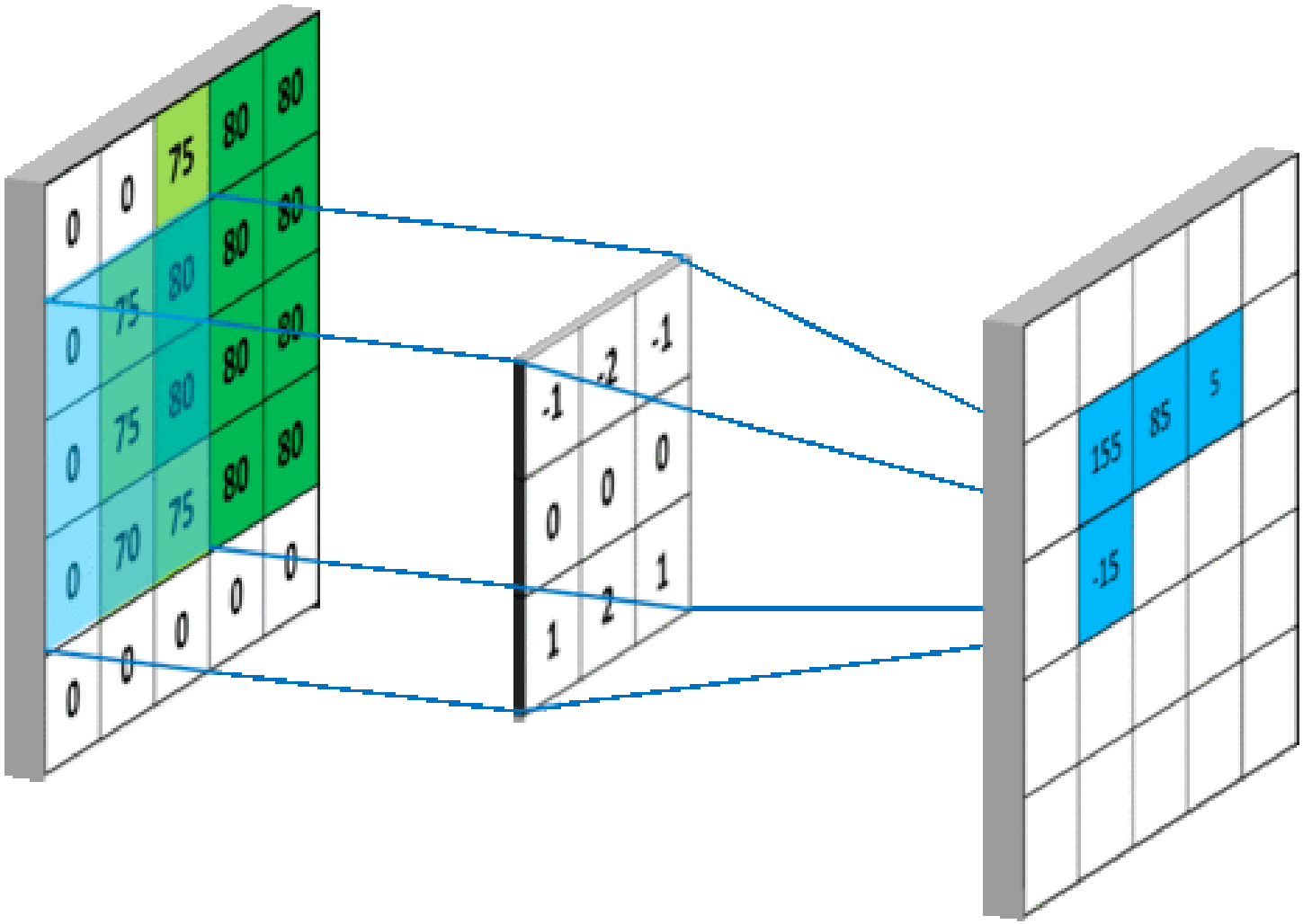
Convolutions



Convolutions



Convolutions



Convolutional models

conv-2

image (32x32x3)

conv (32x32x64)

conv (32x32x64)

maxpool (16x16x64)

ff (16*16*64 $\rightarrow$ 256)

ff (256 $\rightarrow$ 256)

ff (256 $\rightarrow$ 10)

conv-4

image (32x32x3)

conv (32x32x64)

conv (32x32x64)

maxpool (16x16x64)

conv (16x16x128)

conv (16x16x128)

maxpool (8x8x128)

ff (8*8*128 $\rightarrow$ 256)

ff (256 $\rightarrow$ 256)

ff (256 $\rightarrow$ 10)

Results on CIFAR10 for simple CNNs

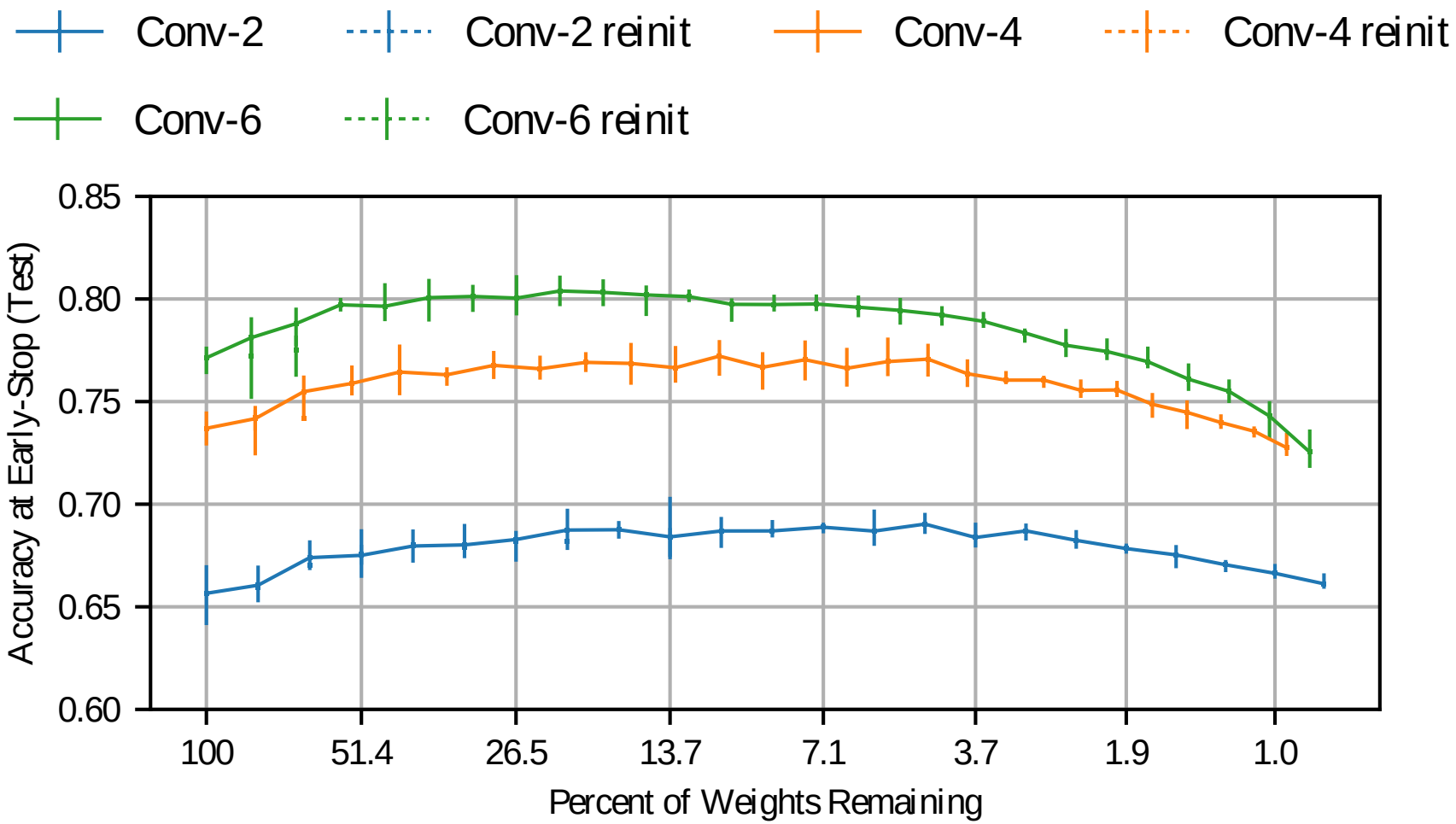


Figure 5 (page 6)

Results on CIFAR10 for simple CNNs

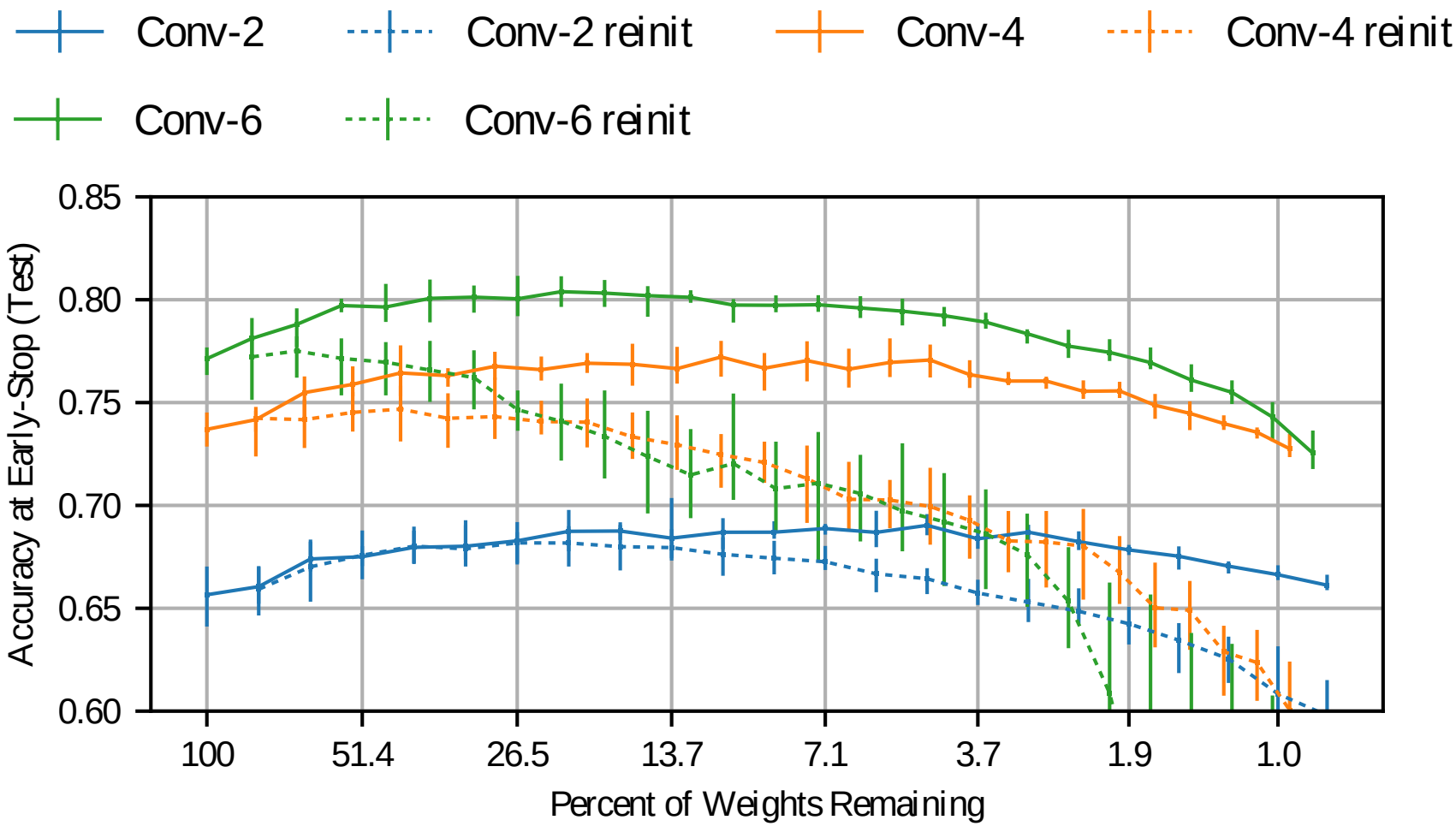


Figure 5 (page 6)

Results on CIFAR10 for realistic models

model: VGG-19 (20M parameters)

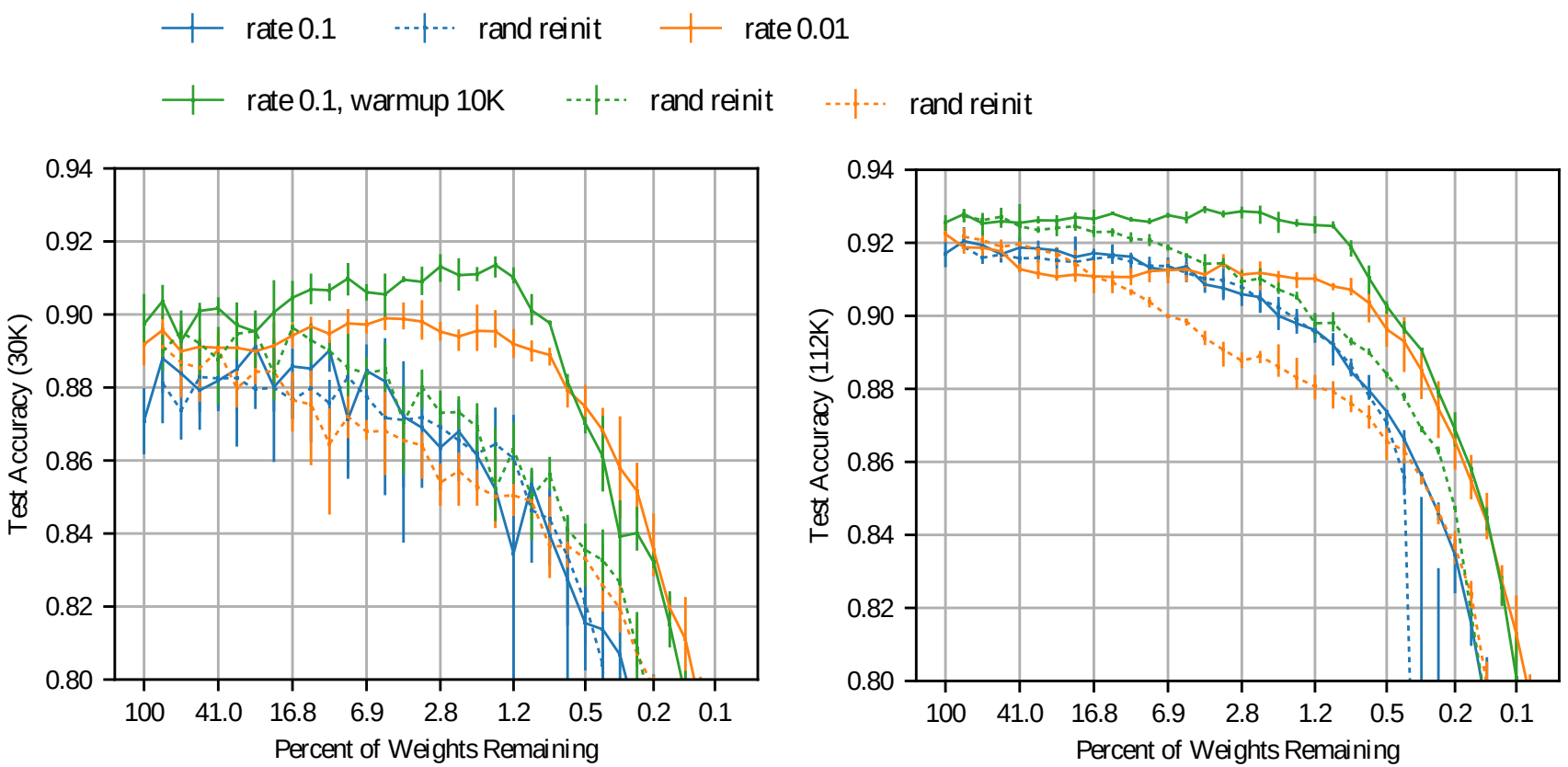


Figure 7 (page 7)

Results on CIFAR10 for realistic models

model: ResNet-18 (274K parameters)

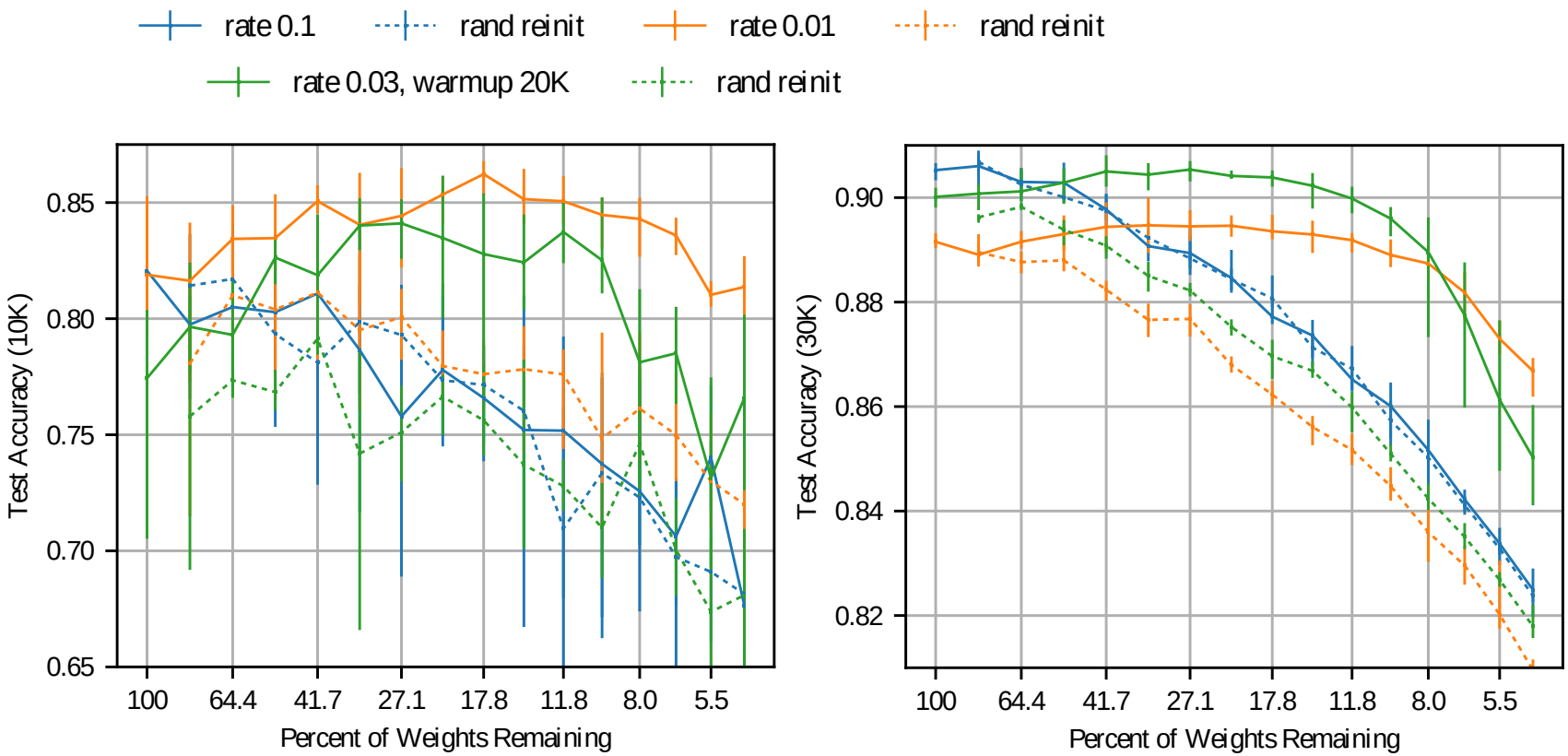


Figure 8 (page 8)

What about transformers/large language models?

? are transformers sparse like feed forward networks/CNNs

? do lottery tickets also work for NLP

? do lottery tickets scale to large networks

results for VGG-19/ResNet-18 suggest that scaling may not work 😕

model compression for transformers mainly uses reduced precision rather than pruning 😕

$\Longrightarrow$ perhaps need better pruning for transformers first?