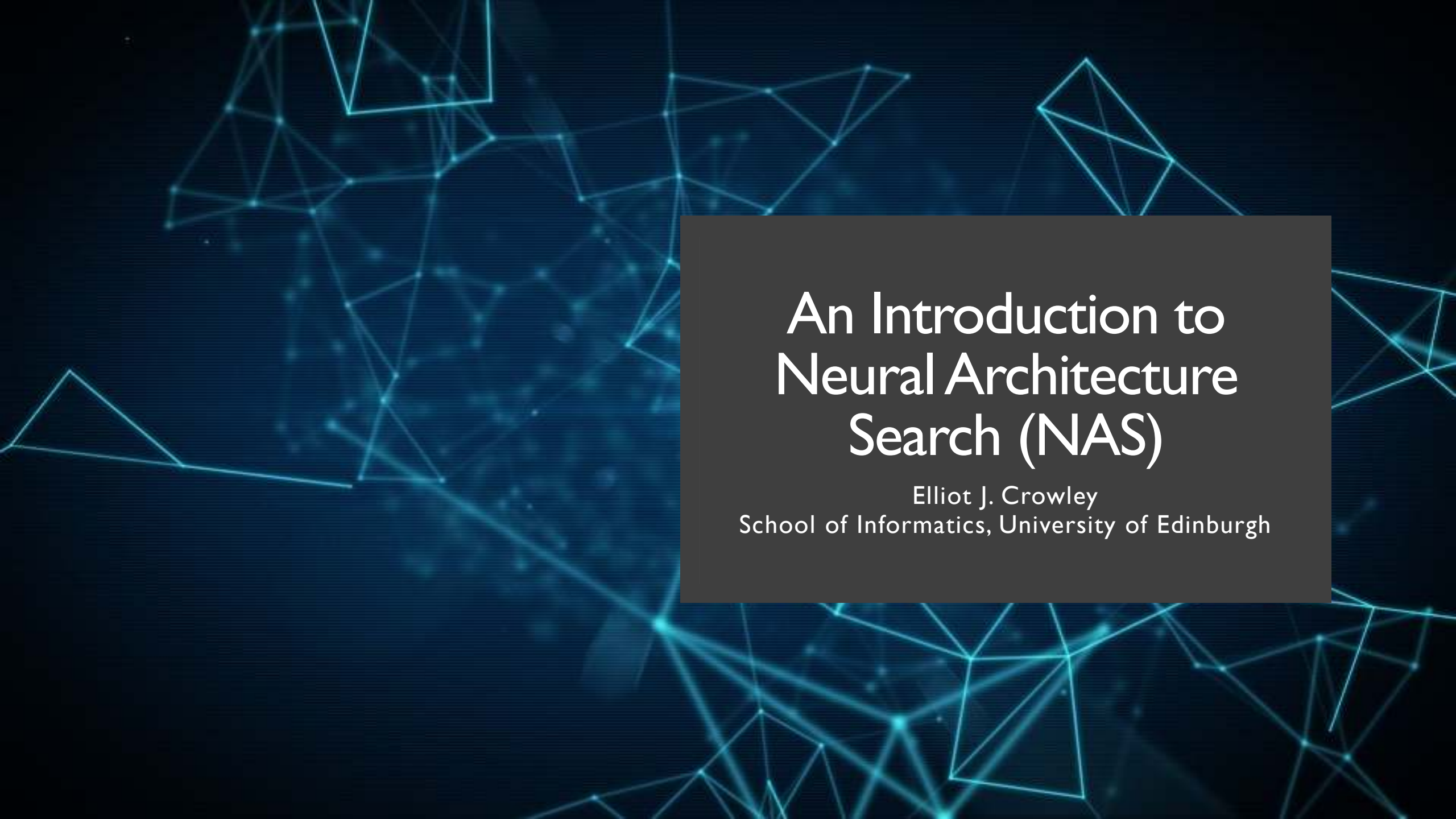


An Introduction to Neural Architecture Search (NAS)

Elliot J. Crowley
School of Informatics, University of Edinburgh

Why do we care?

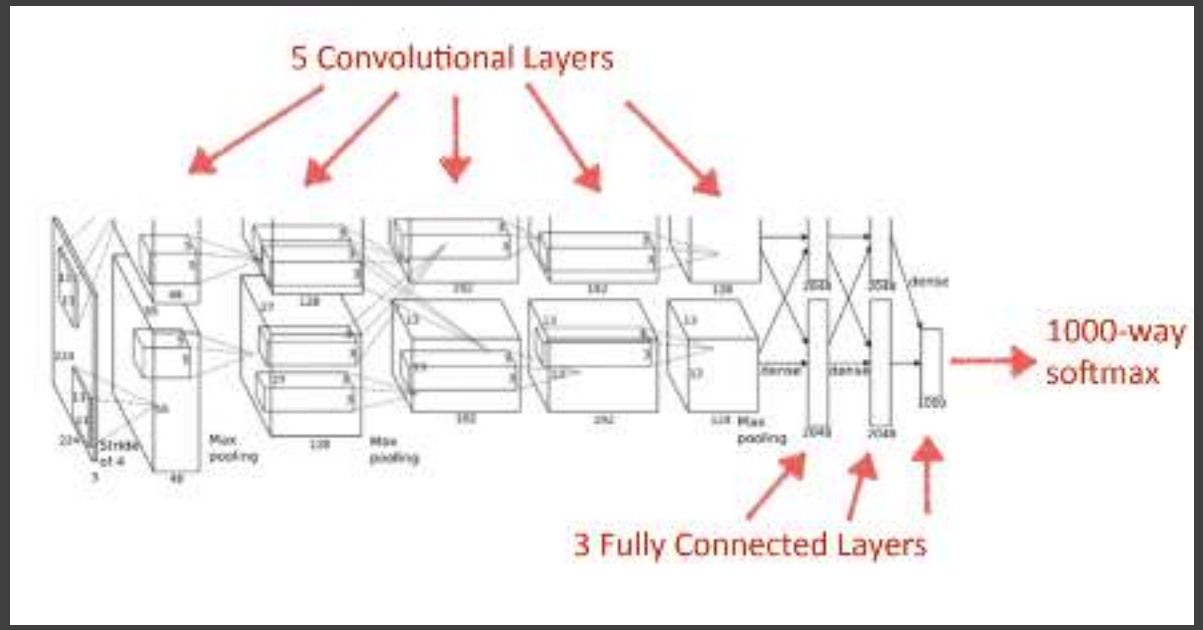
We want smaller, faster networks without compromising on accuracy

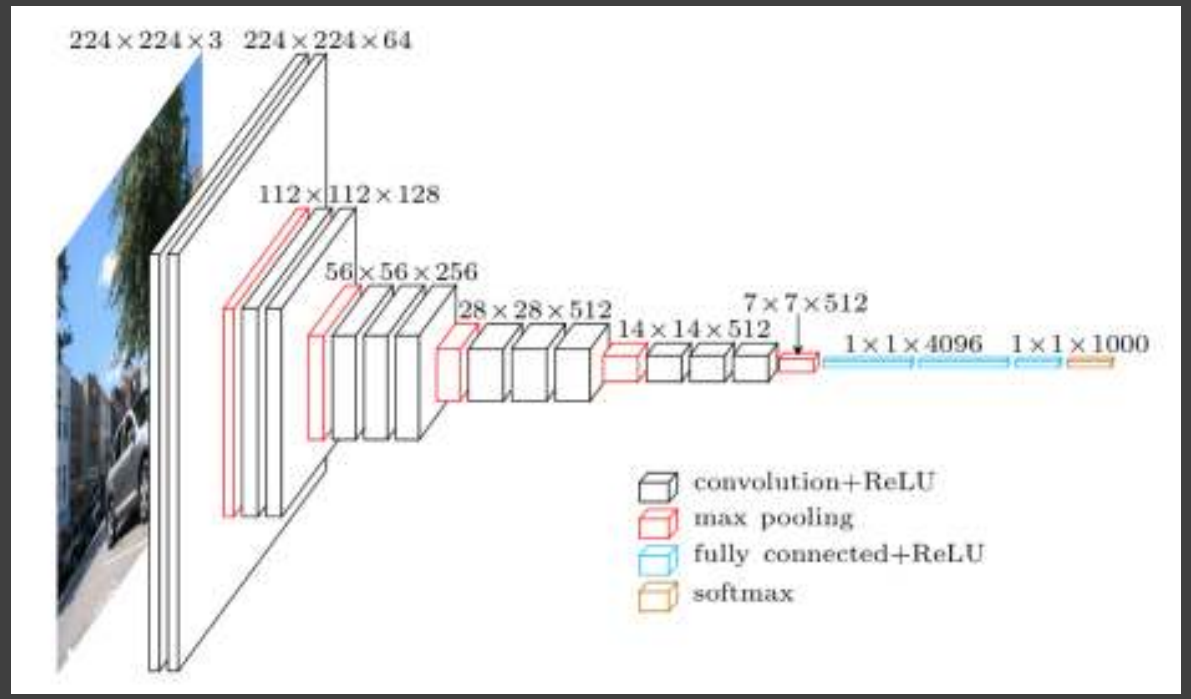
Designing neural networks is expensive (takes human expertise)

We want the best network for a particular task

Two paradigms for NAS

- **Bottom up: Design blocks and stack**
- Top down: Start with a big network and remove redundancies



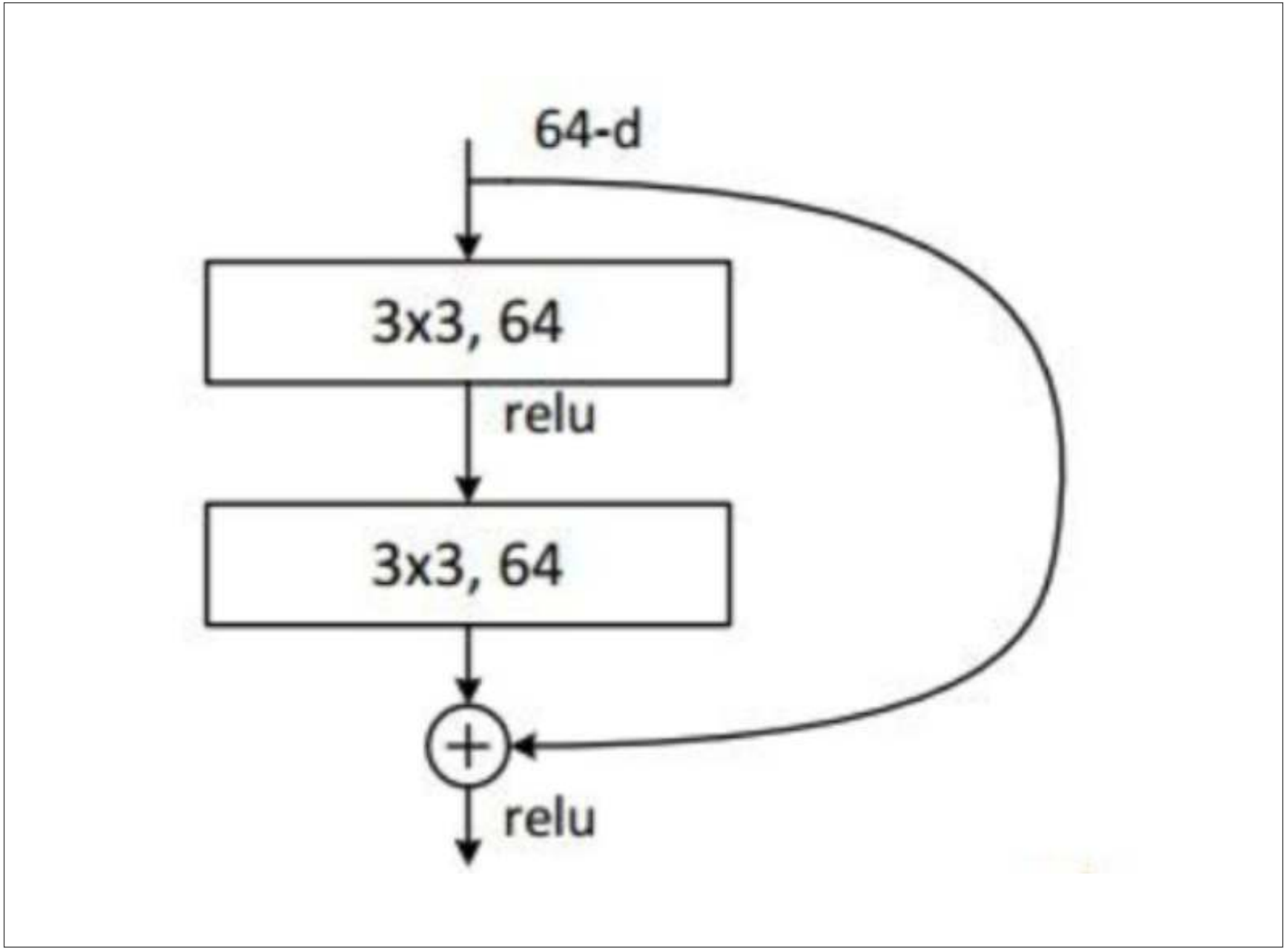


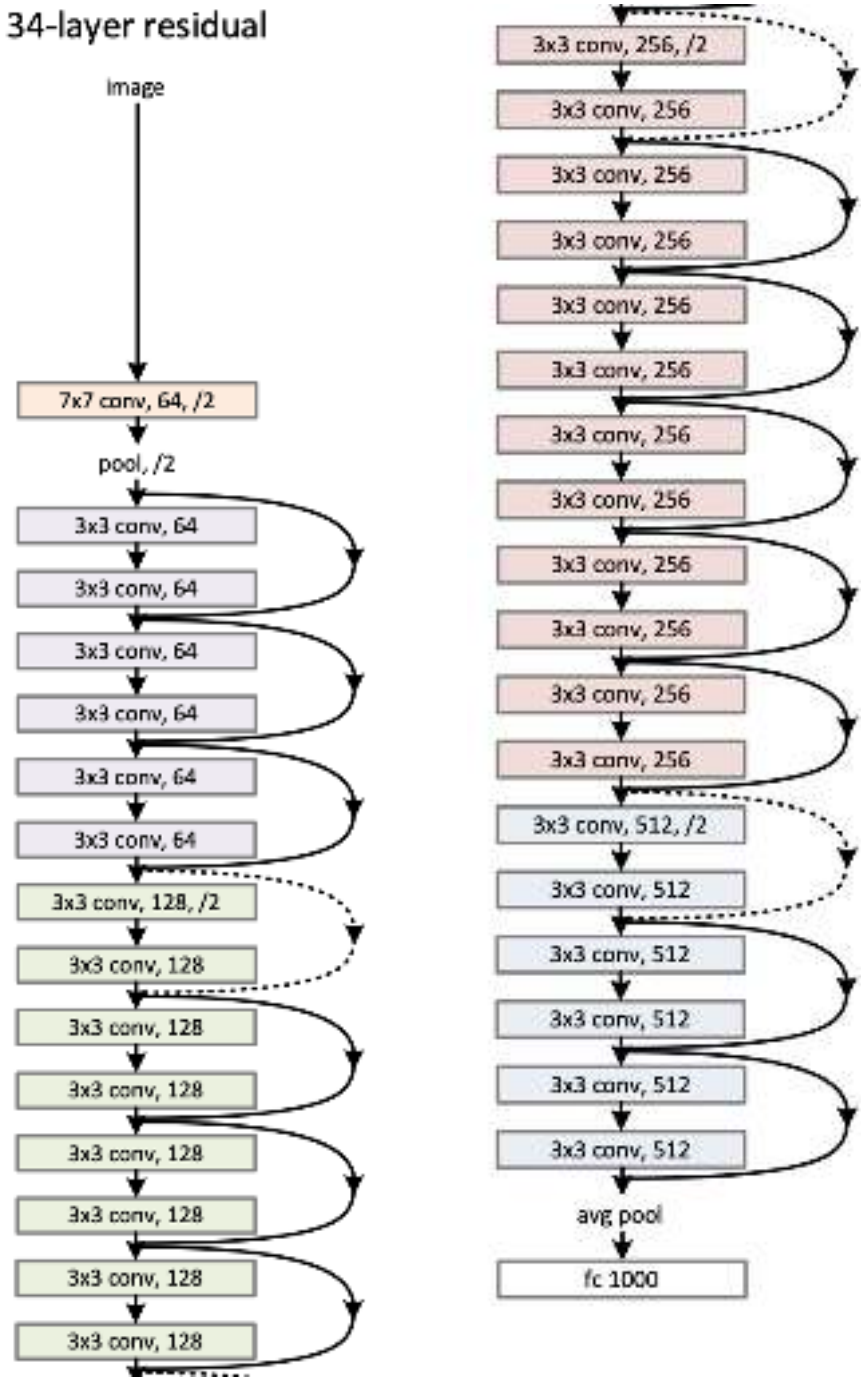
NETS OF OLD

Convolutional neural network designs before 2015 tended to be rather ad hoc

The repeating block

- ResNets popularized the idea of having **repeating blocks** to make up a network

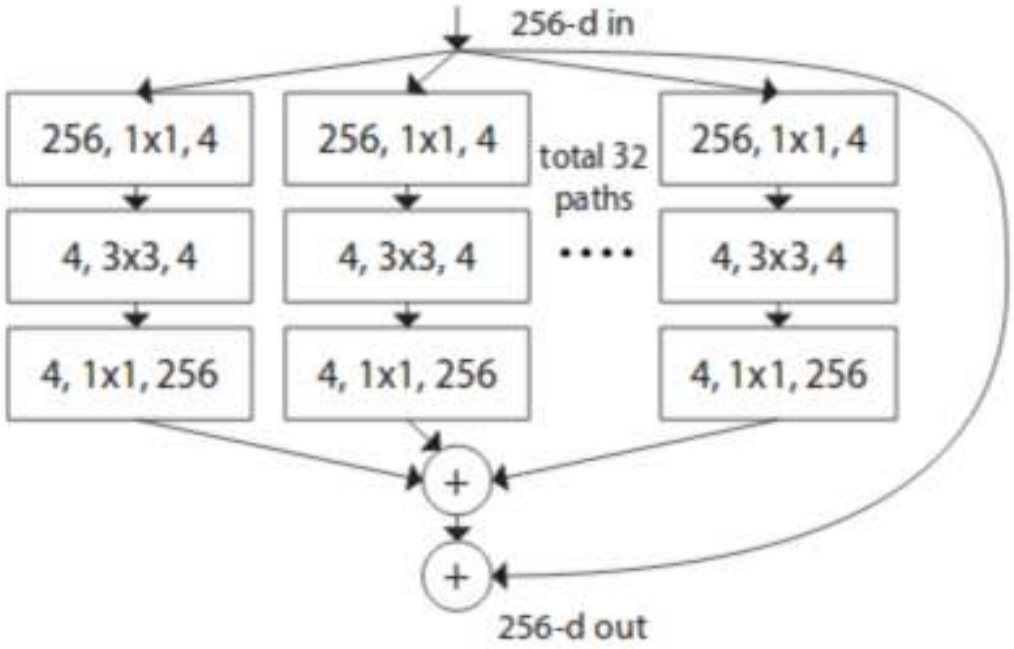


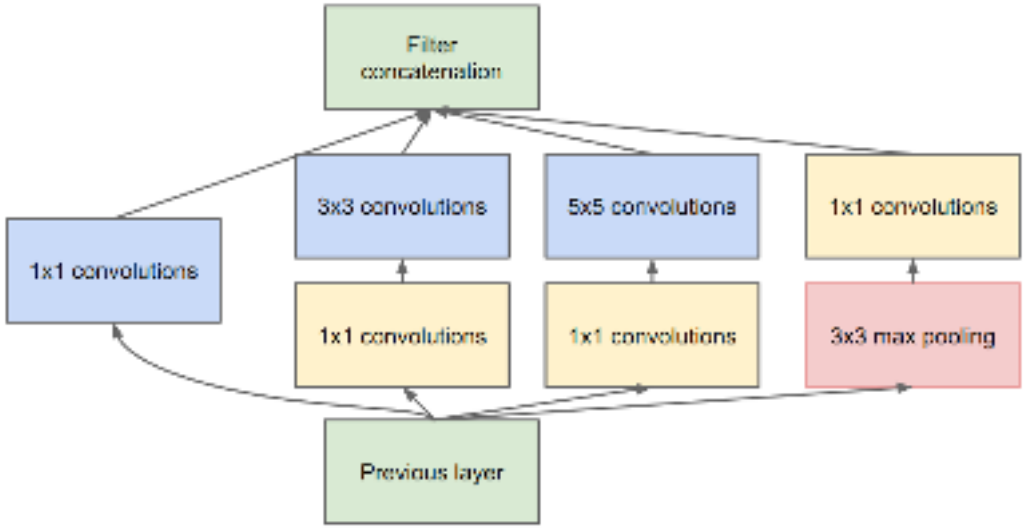


ResNet34

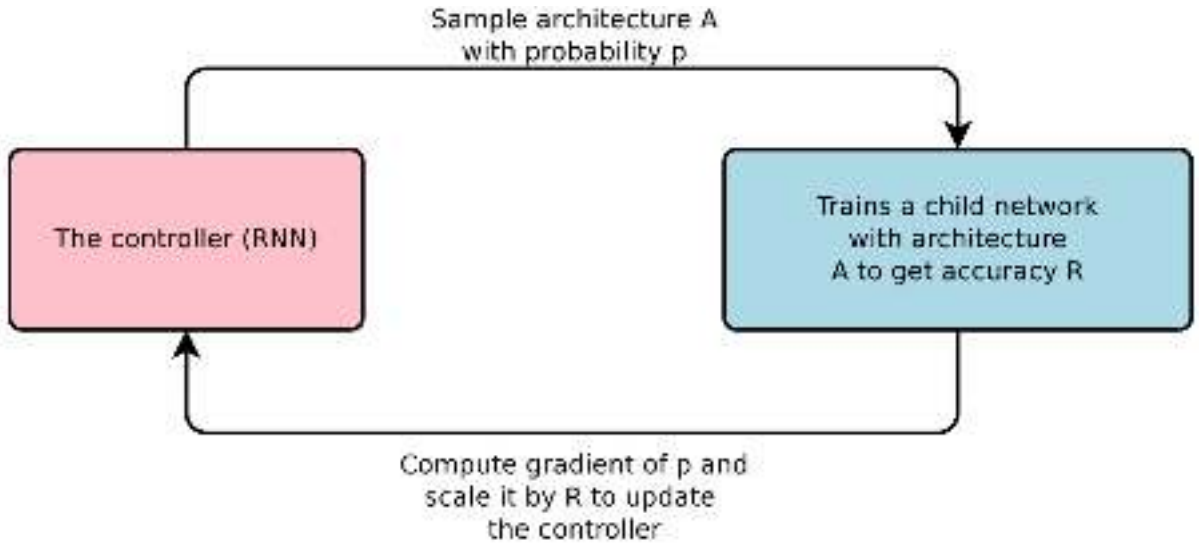
Blocks = [3 4 6 3]

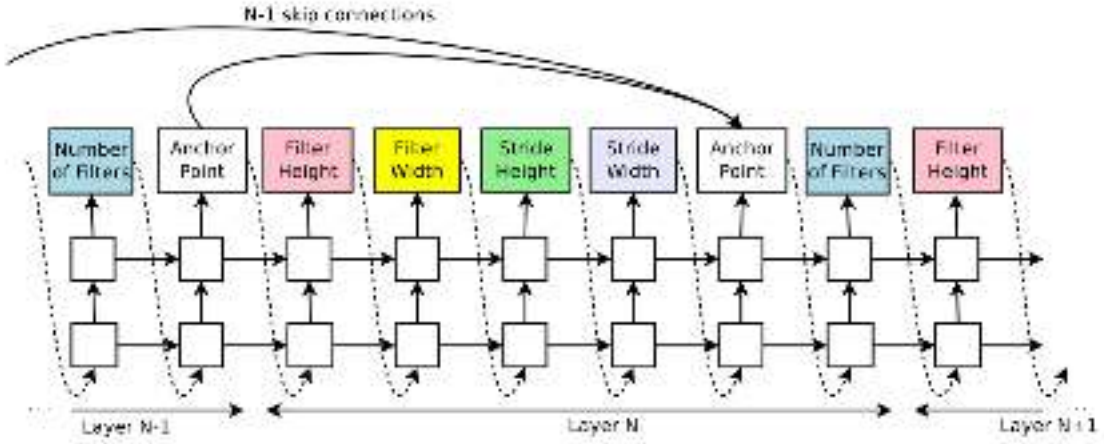
Channels = [64 128 256 512]





THINGS GET SILLY QUITE QUICKLY

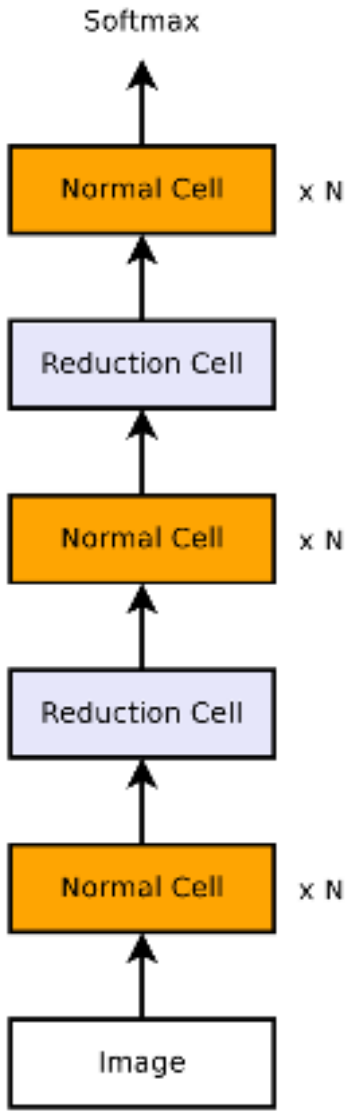




NEURAL ARCHITECTURE SEARCH WITH RL (ZOPH & LE, ICLR 2017)

LEARNING THE WHOLE NETWORK IS EXTREMELY EXPENSIVE AND PAINFUL!

800 GPUS FOR A MONTH :|

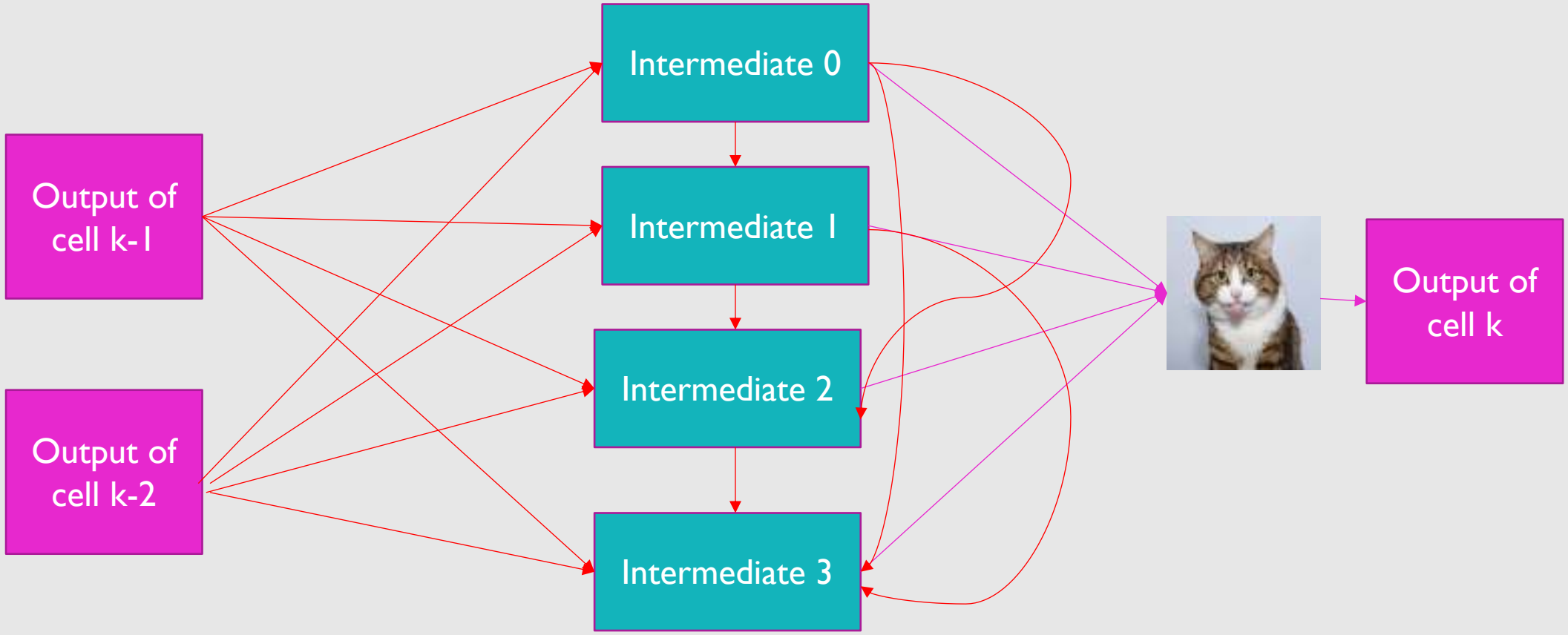


AN ALL-PURPOSE ARCHITECTURE

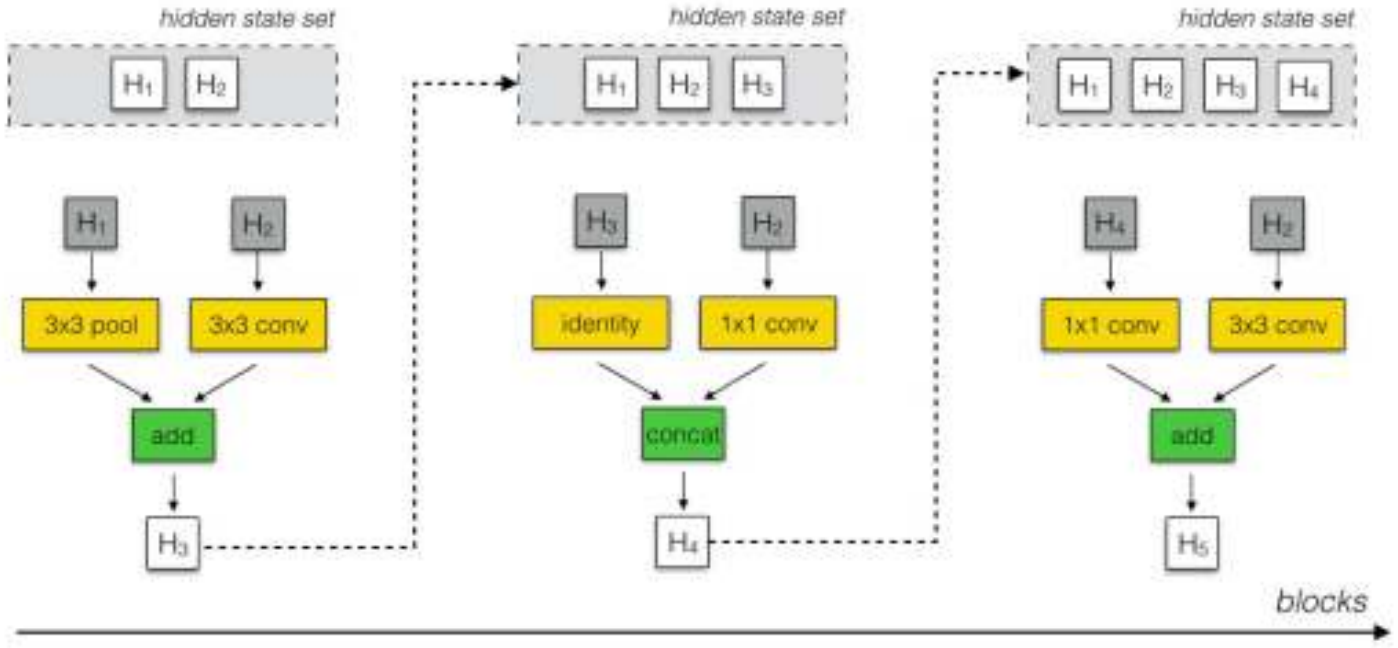
LEARN A CELL RATHER THAN A WHOLE NETWORK (CHEAPER)

N DEPENDING ON BUDGET

The building block in DARTS



LEARNING TRANSFERABLE ARCHITECTURES FOR SCALABLE IMAGE RECOGNITION (ZOPH ET AL. CVPR 2018)



- identity
- 1x7 then 7x1 convolution
- 3x3 average pooling
- 5x5 max pooling
- 1x1 convolution
- 3x3 depthwise-separable conv
- 7x7 depthwise-separable conv
- 1x3 then 3x1 convolution
- 3x3 dilated convolution
- 3x3 max pooling
- 7x7 max pooling
- 3x3 convolution
- 5x5 depthwise-seperable conv

450 GPUs for 3 days

Weight sharing to the rescue*

Fixed weight for each connection e.g. between intermediate 0 and intermediate 1 we have W01

Don't have to train from scratch every time

Only 16 hours on 1 GPU

*weight sharing ruins everything

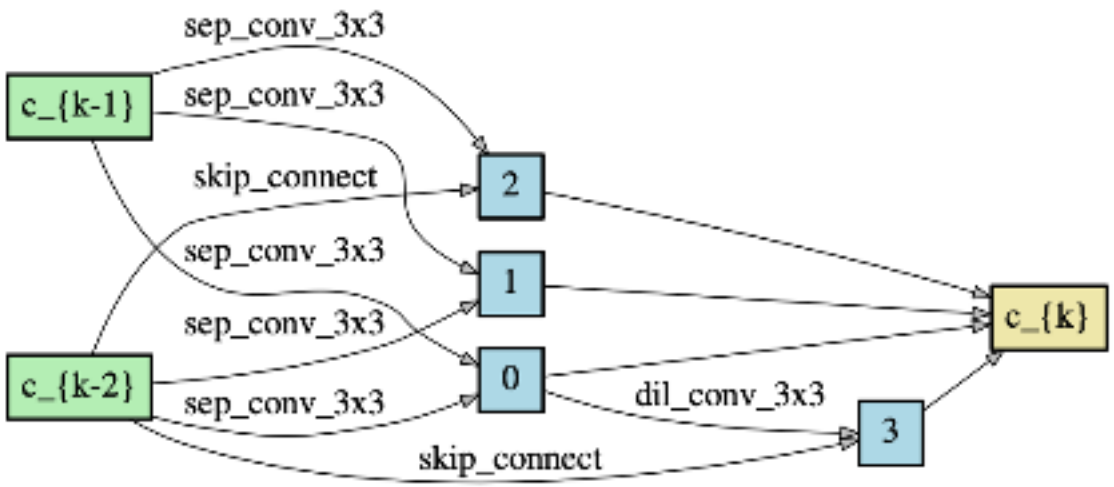
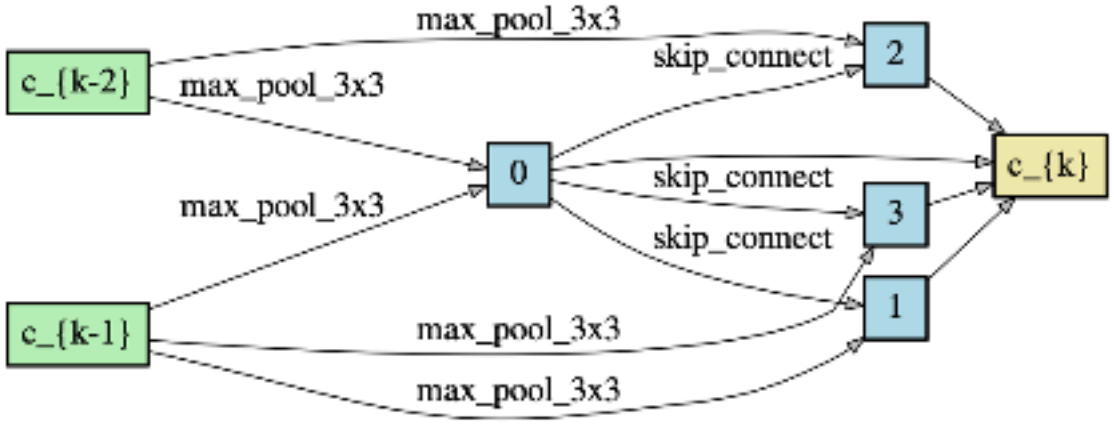
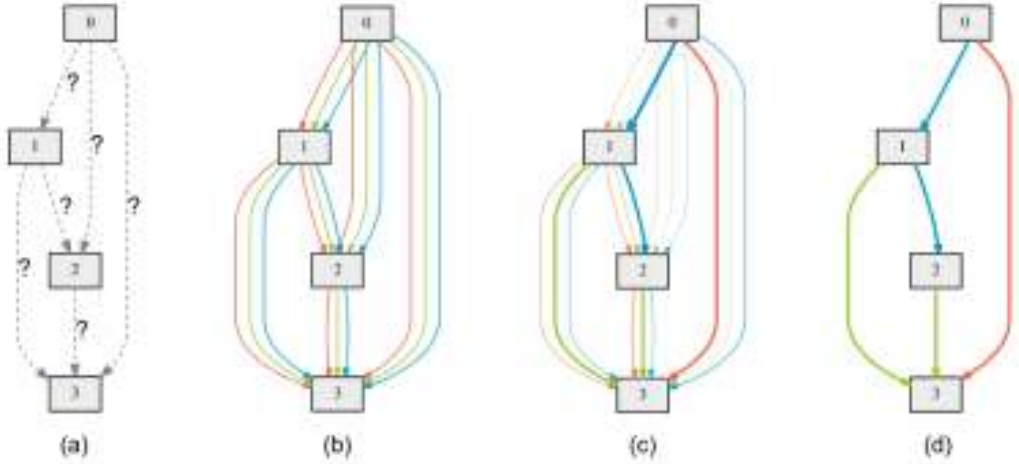


Figure 4: Normal cell learned on CIFAR-10.

DARTS (LIU ET AL. ICLR 2019)

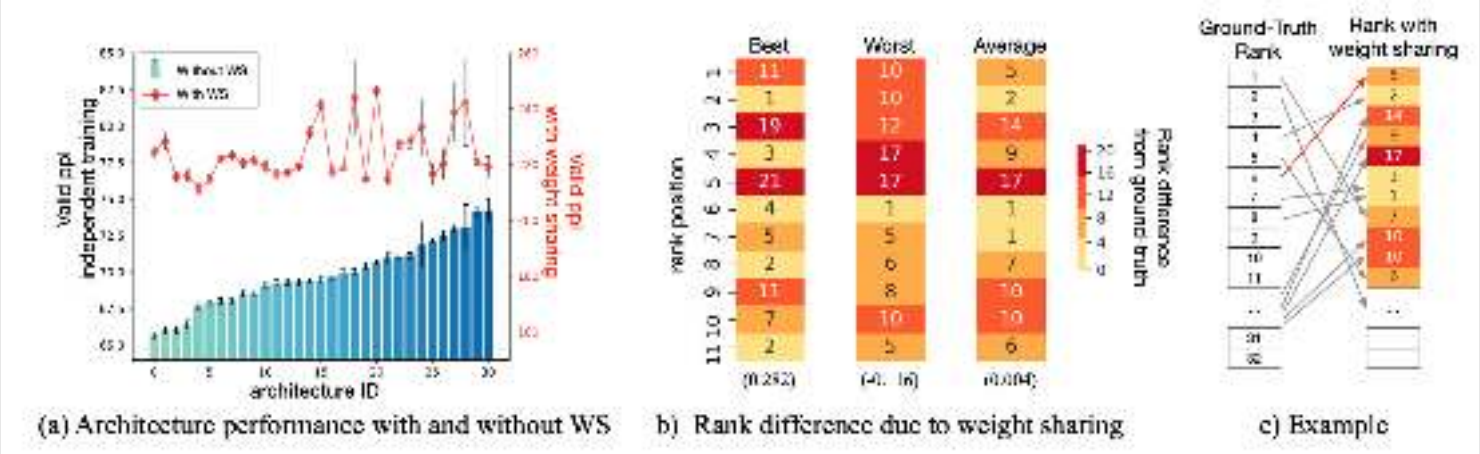




Architecture	Test Error (%)	Params (M)	Search Cost (GPU days)	#ops	Search Method
DenseNet-BC (Huang et al., 2017)	3.46	25.6	–	–	manual
NASNet-A + cutout (Zoph et al., 2018)	2.65	3.3	2000	13	RL
NASNet-A + cutout (Zoph et al., 2018)†	2.83	3.1	2000	13	RL
BlockQNN (Zhong et al., 2018)	3.54	39.8	96	8	RL
AmoebaNet-A (Real et al., 2018)	3.34 ± 0.06	3.2	3150	19	evolution
AmoebaNet-A + cutout (Real et al., 2018)†	3.12	3.1	3150	19	evolution
AmoebaNet-B + cutout (Real et al., 2018)	2.55 ± 0.05	2.8	3150	19	evolution
Hierarchical evolution (Liu et al., 2018b)	3.75 ± 0.12	15.7	300	6	evolution
PNAS (Liu et al., 2018a)	3.41 ± 0.09	3.2	225	8	SMBO
ENAS + cutout (Pham et al., 2018b)	2.89	4.6	0.5	6	RL
ENAS + cutout (Pham et al., 2018b)*	2.91	4.2	4	6	RL
Random search baseline‡ + cutout	3.29 ± 0.15	3.2	4	7	random
DARTS (first order) + cutout	3.00 ± 0.14	3.3	1.5	7	gradient-based
DARTS (second order) + cutout	2.76 ± 0.09	3.3	4	7	gradient-based

Evaluating the Search Phase of NAS (Yu et al. ICLR, 2020)

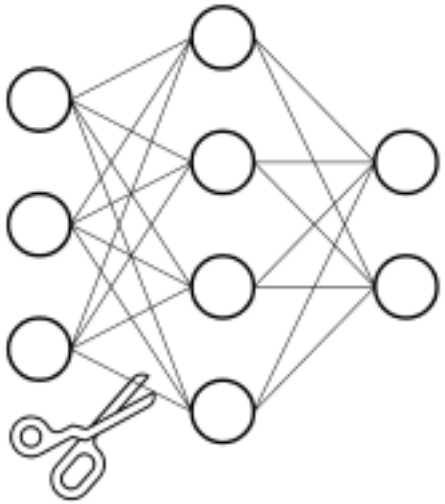
- Random is similar to NAS!
- Constrained search space is very good
- Weight sharing ruins rank



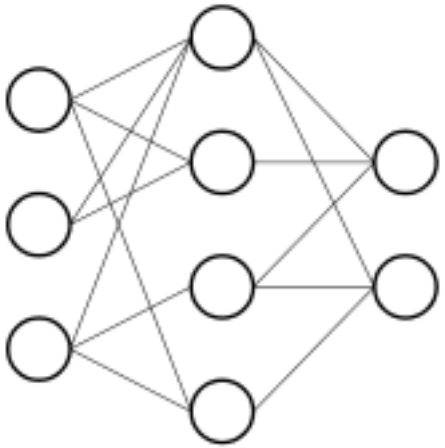
	PTB (PPL)	t-test	CIFAR-10 (acc.)	t-test
ENAS	59.88 ± 1.92	0.73	96.79 ± 0.11	0.01
DARTS	60.61 ± 2.54	0.62	96.62 ± 0.23	0.20
NAO	61.99 ± 1.95	0.02	96.86 ± 0.17	0.00
Random	60.13 ± 0.65	-	96.44 ± 0.19	-

Two paradigms for NAS

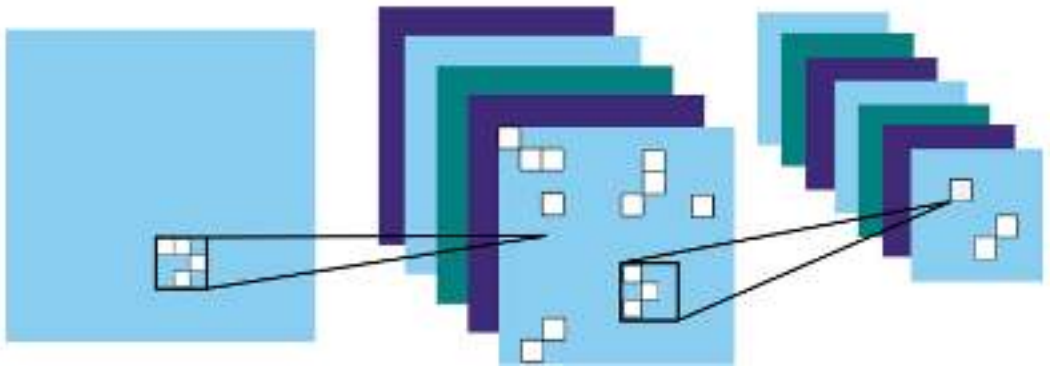
- Bottom up: Design blocks and stack
- **Top down: Start with a big network and remove redundancies**



Before pruning



After pruning



WEIGHT PRUNING

Classic Approach to Weight Pruning (Based on Han et al. ICLR 2016)

TAKE A LARGE TRAINED NETWORK

RANK CONNECTIONS (E.G. BY MAGNITUDE OF EACH WEIGHT)

KILL WEAKEST CONNECTIONS

FINE-TUNE

The Lottery Ticket Hypothesis (Frankle and Carbin, ICLR 2019)

They postulate that within a network there exists a sparse subnetwork that was fortuitously initialized (a lottery ticket)

This is found through weight pruning

SNIP (Lee et al., ICLR 2019)

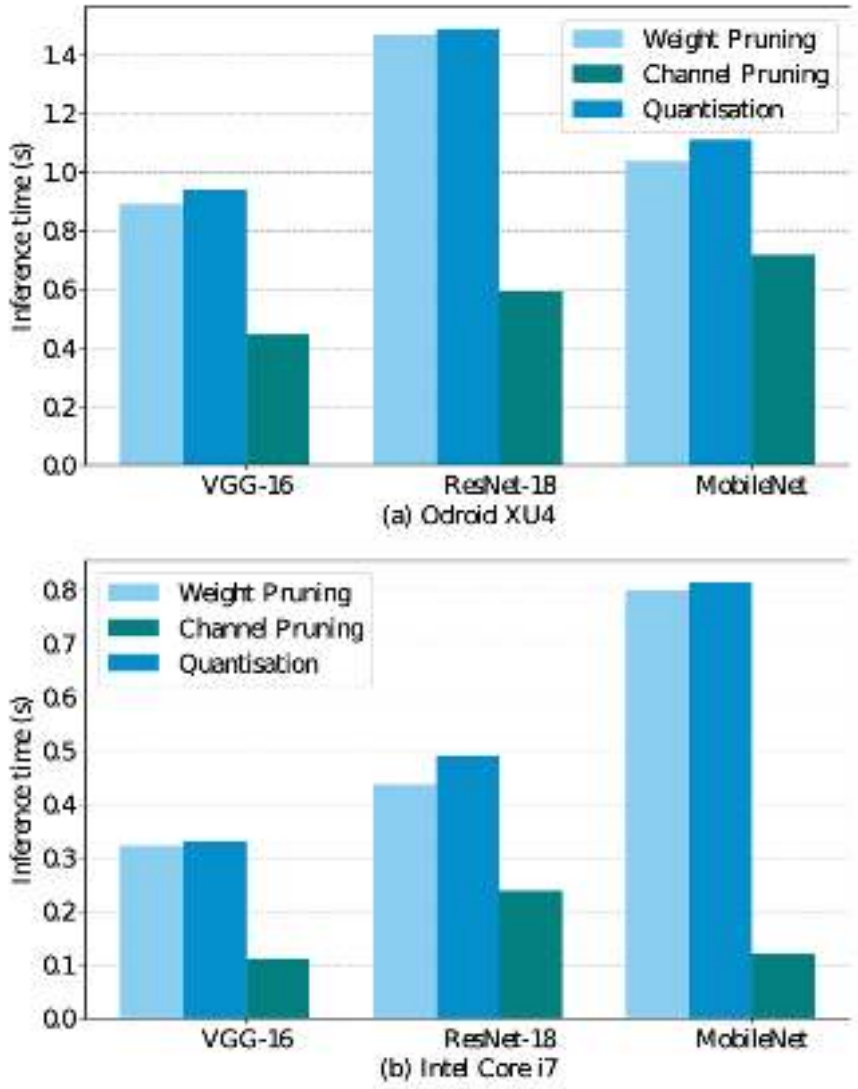
Take a large **untrained** network

Push a single minibatch through

Look at the connection sensitivity

Remove weakest connections

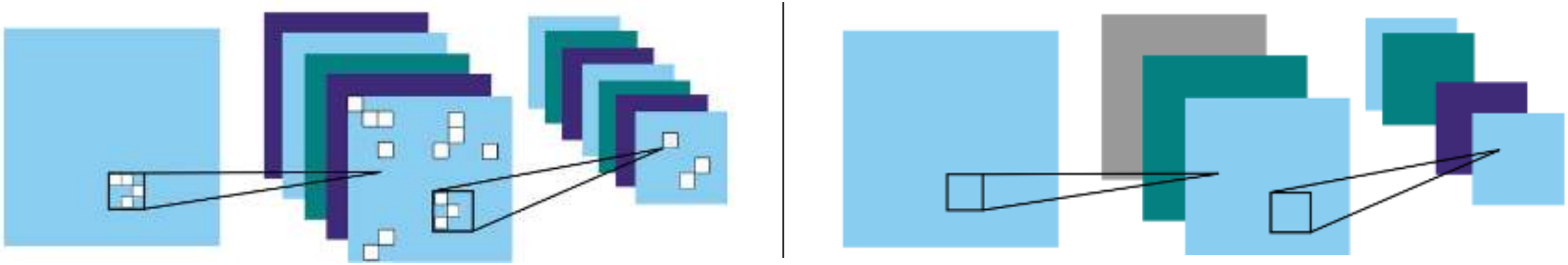
Train from scratch



The problem with sparse networks

- They are not hardware-friendly ☹

Note that there is work on making sparse networks fast (e.g. Fast Sparse ConvNets by Elsen et al. 2019) but results are limited to a single-core CPU



CHANNEL PRUNING (RIGHT)

STILL NOT AS FAST

				Core i7 (CPU)		1080Ti (GPU)	
Network	Params	MACs	Error	Speed	MACs/ps	Speed	MACs/ps
ResNet-18	11.6M	1.81G	30.24	**0.060s**	3.03	**0.002s**	101.3
ResNet-34-A	12.5M	2.42G	**28.14**	0.085s	2.83	0.004s	72.0
ResNet-34-B	7.5M	1.78G	30.77	0.066s	2.71	0.003s	49.6
ResNet-9	5.4M	0.89G	37.04	**0.035s**	2.52	**0.001s**	79.0
ResNet-34-C	4.9M	1.22G	**33.49**	0.054s	2.28	0.003s	35.4
ResNet-34-D	2.5M	0.66G	38.88	0.042s	1.16	0.003s	18.0

Channel pruning relies on reducing channel width which is hardware-unfriendly

Turns out training a smaller version (e.g. lower depth, width) of the original large network is faster and as good!
Paper worthy?

Nope! ICML 2019 Reviews (Reject) ☹

"Unfortunately, the authors do not seem to understand two primary goals of pruning: 1) reducing the number of weights for storage/bandwidth efficiency and 2) **use in (not yet existing) hardware** with sparse arithmetic support."

"This paper did not propose any new method and only reported some simple pruning experiments. The novelty is limited. "

"The paper is well-written and performs an interesting set of experiments. My main concern is that there is little novelty in this work which reduces the significance of the contributions. "

But sometimes…

Paper Decision

ICLR 2020 Conference Program Chairs

19 Dec 2019 (modified: 20 Dec 2019) ICLR 2020 Conference Paper1983 Decision Readers: Everyone

Decision: Accept (Poster)

Comment: This paper is far more borderline than the review scores indicate. The authors certainly did themselves no favours by posting a response so close to the end of the discussion period, but there was sufficient time to consider the responses after this, and it is somewhat disappointing that the reviewers did not engage.

Reviewer 2 states that their only reason for not recommending acceptance is the lack of experiments on more than one KG. The authors point out they have experiments on more than one KG in the paper. From my reading, this is the case. I will consider R2 in favour of the paper in the absence of a response.

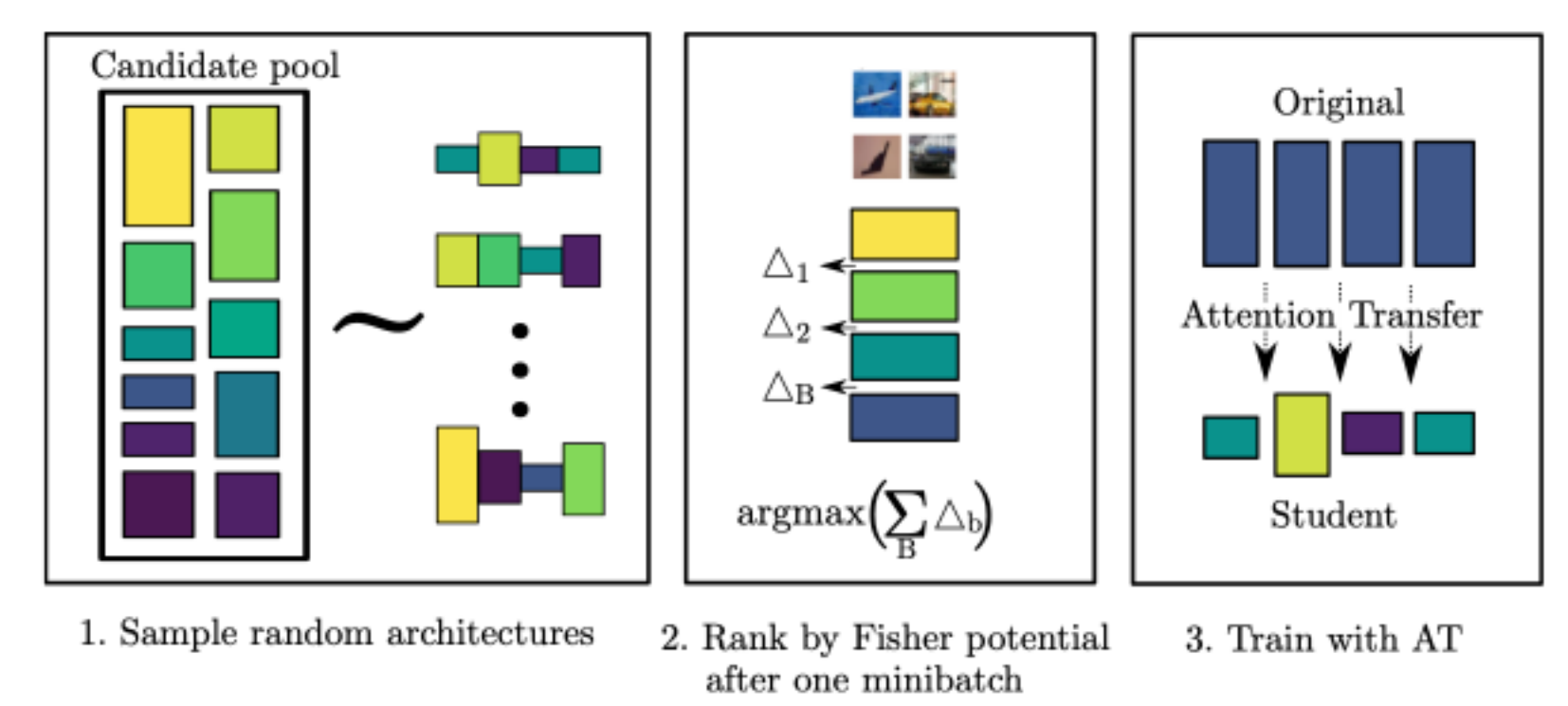
Reviewer 3 gives a fairly clear initial review which states the main reasons they do not recommend acceptance. While not an expert on the topic of GNNs, I have enough of a technical understanding to deem that the detailed response from the authors to each of the points does address these concerns. In the absence of a response from the reviewer, it is difficult to ascertain whether they would agree, but I will lean towards assuming they are satisfied.

Reviewer 1 gives a positive sounding review, with as main criticism "Overall, the work of this paper seems technically sound but I don't find the contributions particularly surprising or novel. Along with plogicnet, there have been many extensions and applications of Gnns, and I didn't find that the paper expands this perspective in any surprising way." This statement is simply re-asserted after the author response. I find this style of review entirely inappropriate and unfair: it is not a the role of a good scientific publication to "surprise". If it is technically sound, and in an area that the reviewer admits generates interest from reviewers, vague weasel words do not a reason for rejection make.

I recommend acceptance.

WARNING: SHAMELESS SELF-PROMOTION TO FOLLOW

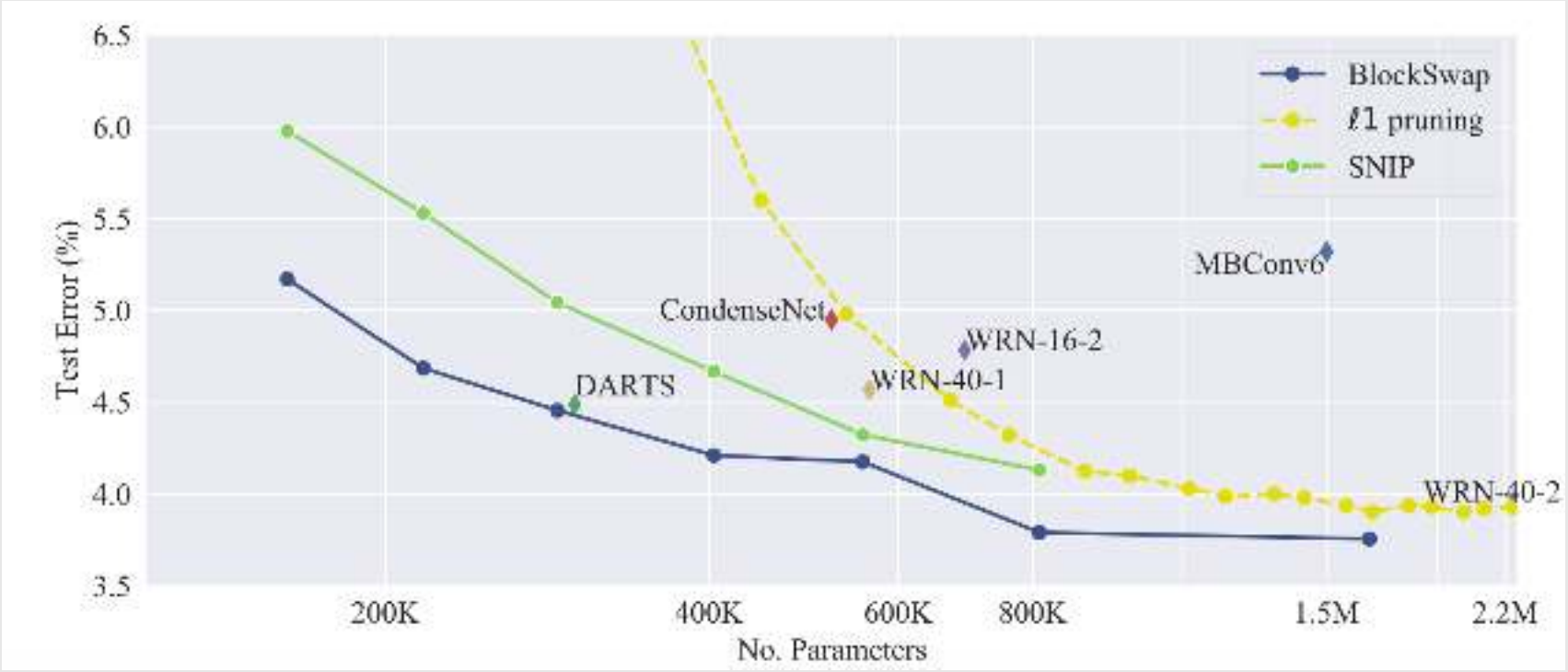
BlockSwap (Turner et al. ICLR 2020)



Takes 5 minutes on 1 GPU

We use the very simple blocks from Moonshine (Crowley et al., NeurIPS 2018)

Block	S	$G(g)$	$B(b)$	$BG(b,g)$
Structure	Conv Conv	GConv (g) Conv1x1 GConv (g) Conv1x1	Conv1x1($N \rightarrow \frac{N}{b}$) Conv Conv1x1($\frac{N}{b} \rightarrow N$)	Conv1x1($N \rightarrow \frac{N}{b}$) GConv(g) Conv1x1($\frac{N}{b} \rightarrow N$)
Conv Params	$2N^2k^2$	$2N^2(\frac{k^2}{g}+1)$	$N^2(\frac{k^2}{b^2}+\frac{2}{b})$	$N^2(\frac{k^2}{gb^2}+\frac{2}{b})$
BN Params	$4N$	$8N$	$N(2+\frac{4}{b})$	$N(2+\frac{4}{b})$



- Works well (similar to DARTS despite the search being 300x faster)
- And works better than random!

Thank you!

- Email elliot.j.crowley@ed.ac.uk
- Or visit bayeswatch.com for our group work
-