Meta Learning, NASnet & AutoML

By Sam Witteveen
@sam_witteveen
sam@reddragon.ai

RED DRAGON AI

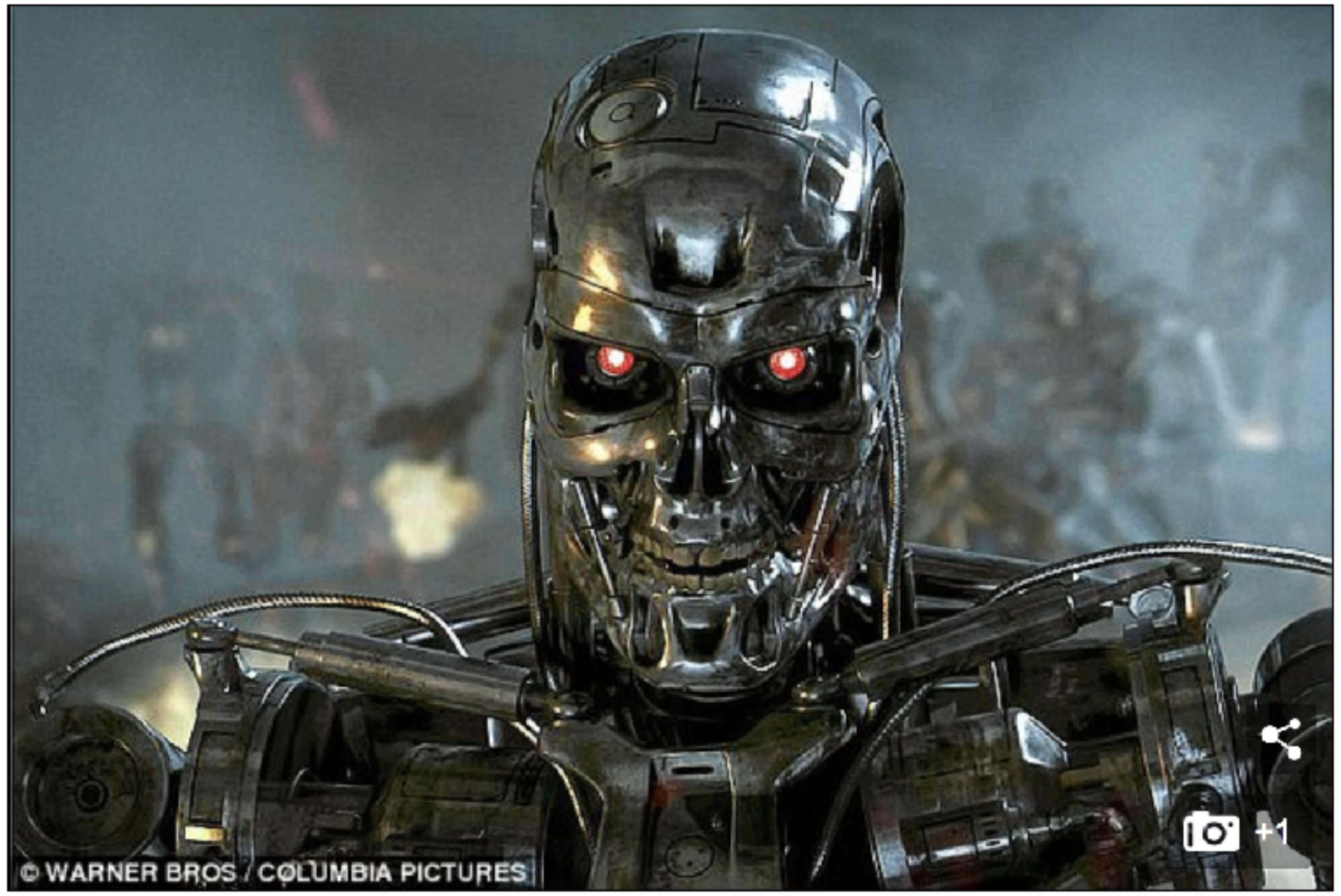
AI creates AI = AI Take Over

Google supercomputer creates its own 'AI child' that can outperform any machine made by humans

- The machine called NASNet learns through 'reinforcement learning'
- It reports back to its AI 'parent' and then leanst how it ca
- AI can recognise objects, such as people, cars, handbag
- The findings show automation could create more AI all b

What could possibly go wrong? Google's machine learning AI has been able to replicate itself for the first time

- Google's AutoML software is designed to help the firm create other AIs
- ul machine-learning software than humans have
- ep toward machines making complex AI without



Google's machine learning artificial intelligence (AI) software has learned to replicate itself for the first time. The breakthrough could one day lead to machines that can learn without human input, a long fascination of science fiction - including films like The Terminator (pictured)

RED DRAGON AI

What is Meta Learning

- The framework of using one learning system to modify or optimize certain aspects of another learning system
- AutoML and Learning to Learn are often used with Meta Learning interchangeably
- Learning of Architectures and learning of hyper parameters

RED DRAGON AI

How to architect models

- Decide on an architecture
- Start to assemble and train
- Mess with it - Hyper parameters
- Often art as much as science
- ….. This can be automated

Architecture Search Space

- Type of layers and cells
- Settings on those cells and layers
    - Eg. Filter width/height, #Filters, Non-linearities
- Related to grid search in traditional ML but much more advanced

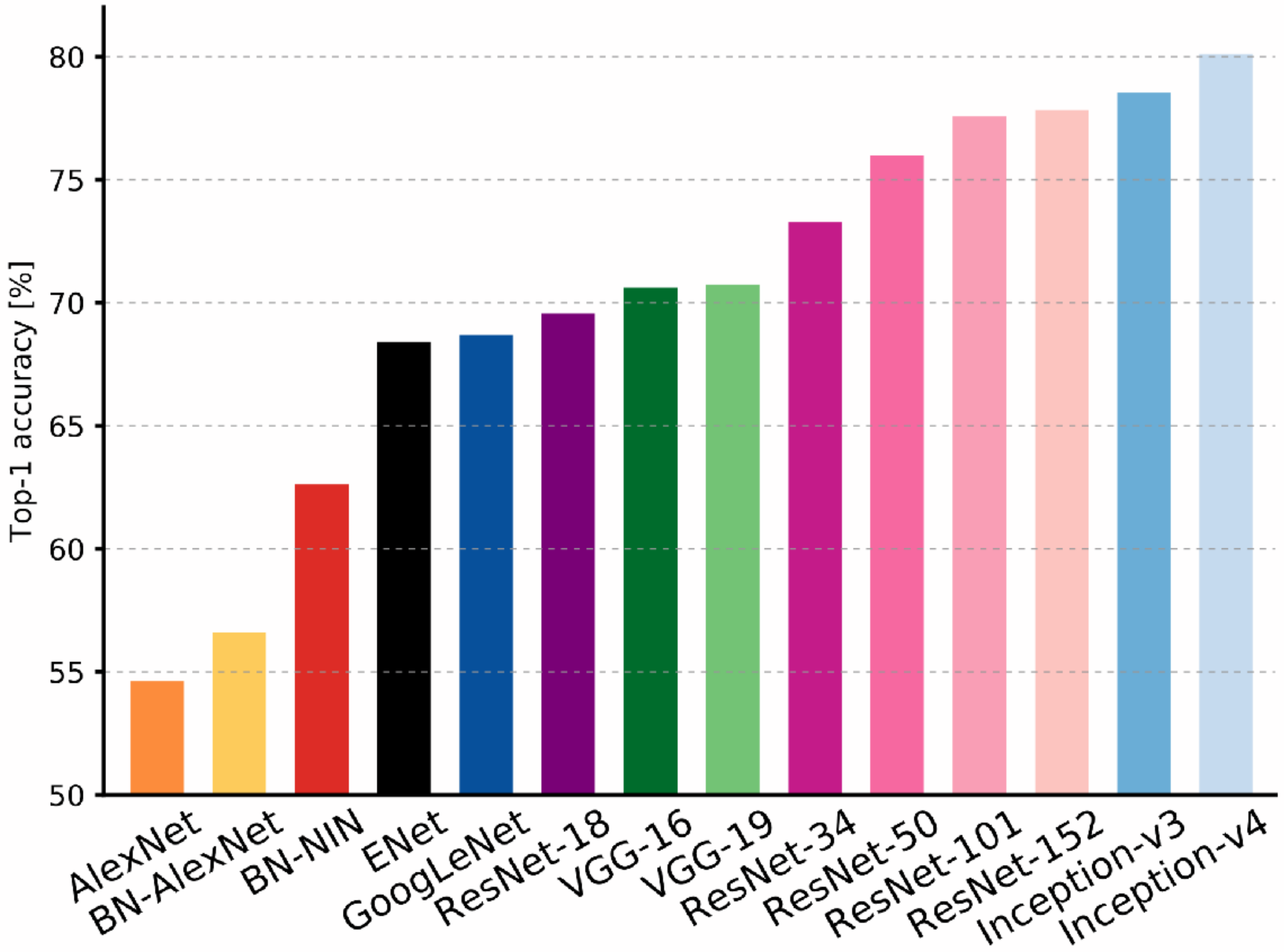
RED DRAGON AI

Why Meta Learning

- As frameworks get bigger and more complicated, architectures and hyper parameters get increasingly harder to tune and optimize

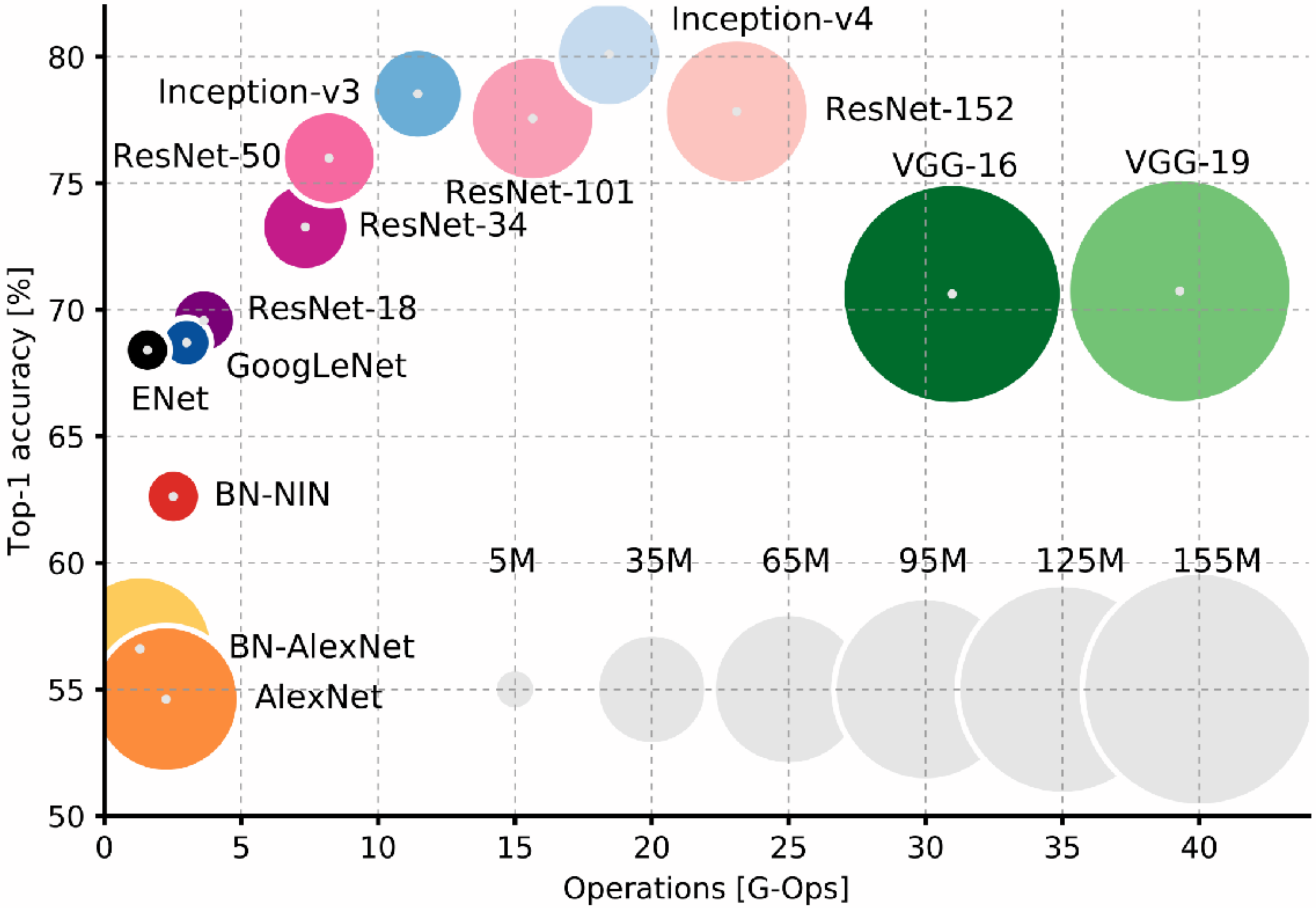
- What if there are architectures that do better than anything humans will think of?

RED DRAGON AI



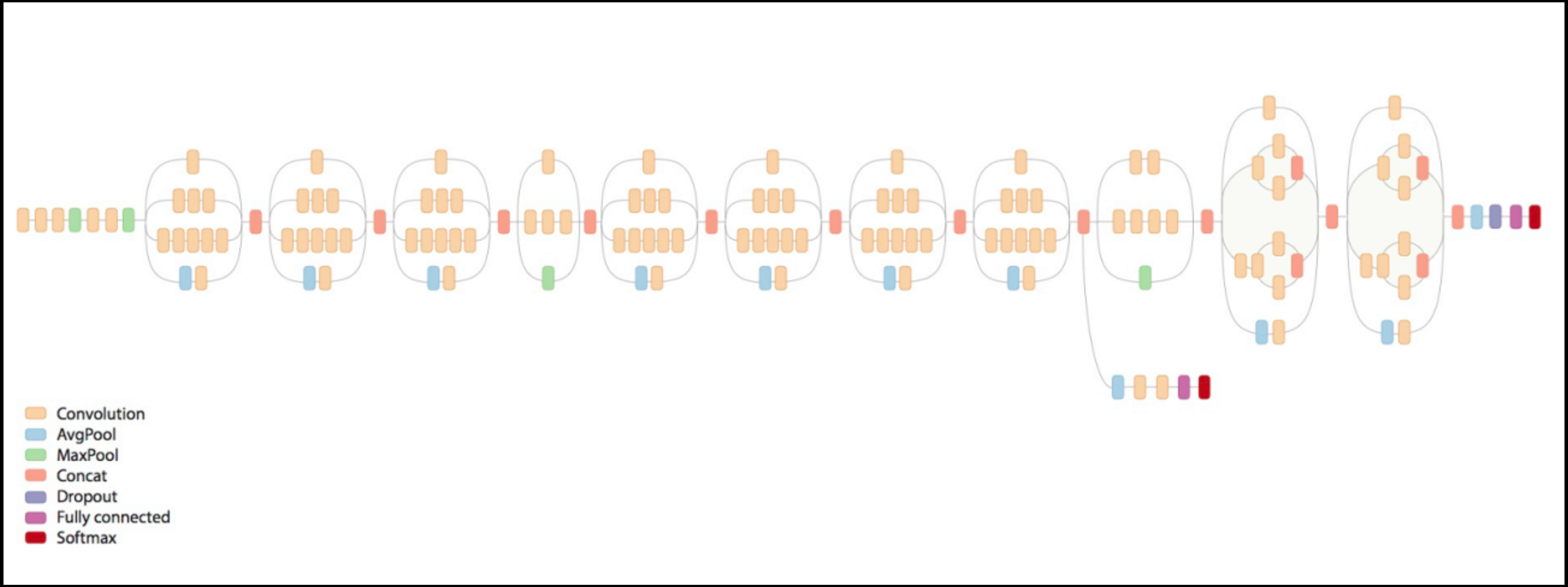
Canziani et al, 2017

RED DRAGON AI



RED DRAGON AI

InceptionV3



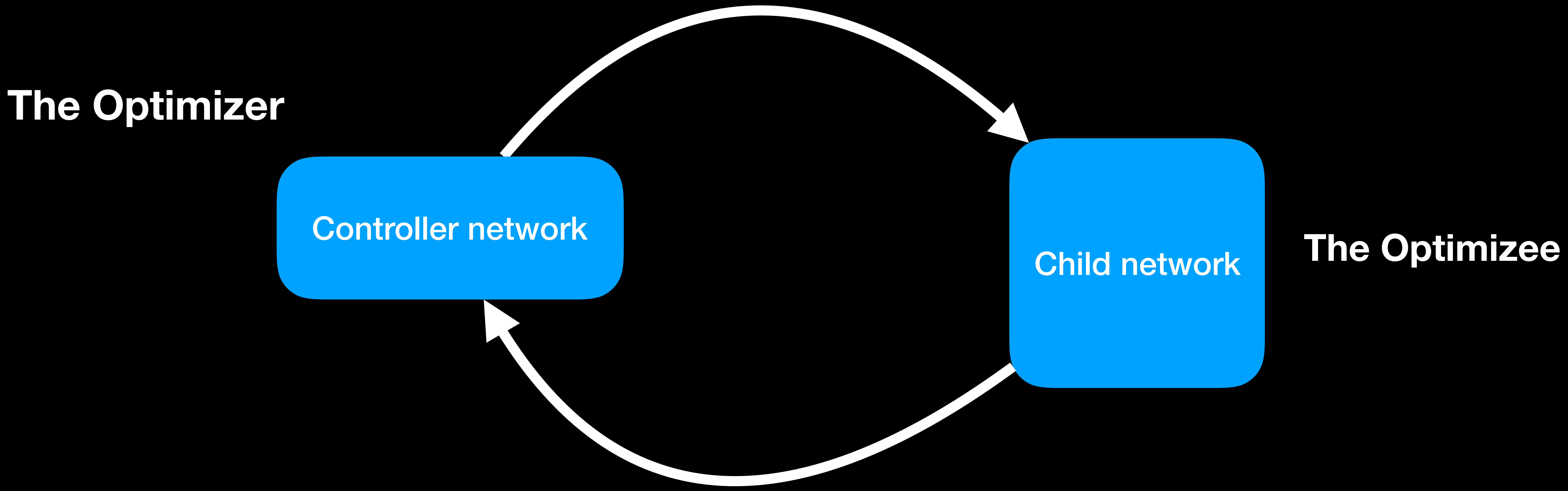
RED DRAGON AI

4 ways to do Meta Learning

- Evolutionary Optimization
- Bayesian Optimization
- Gradient Descent
- Reinforcement Learning

RED DRAGON AI

Parts of the Meta Learning Network

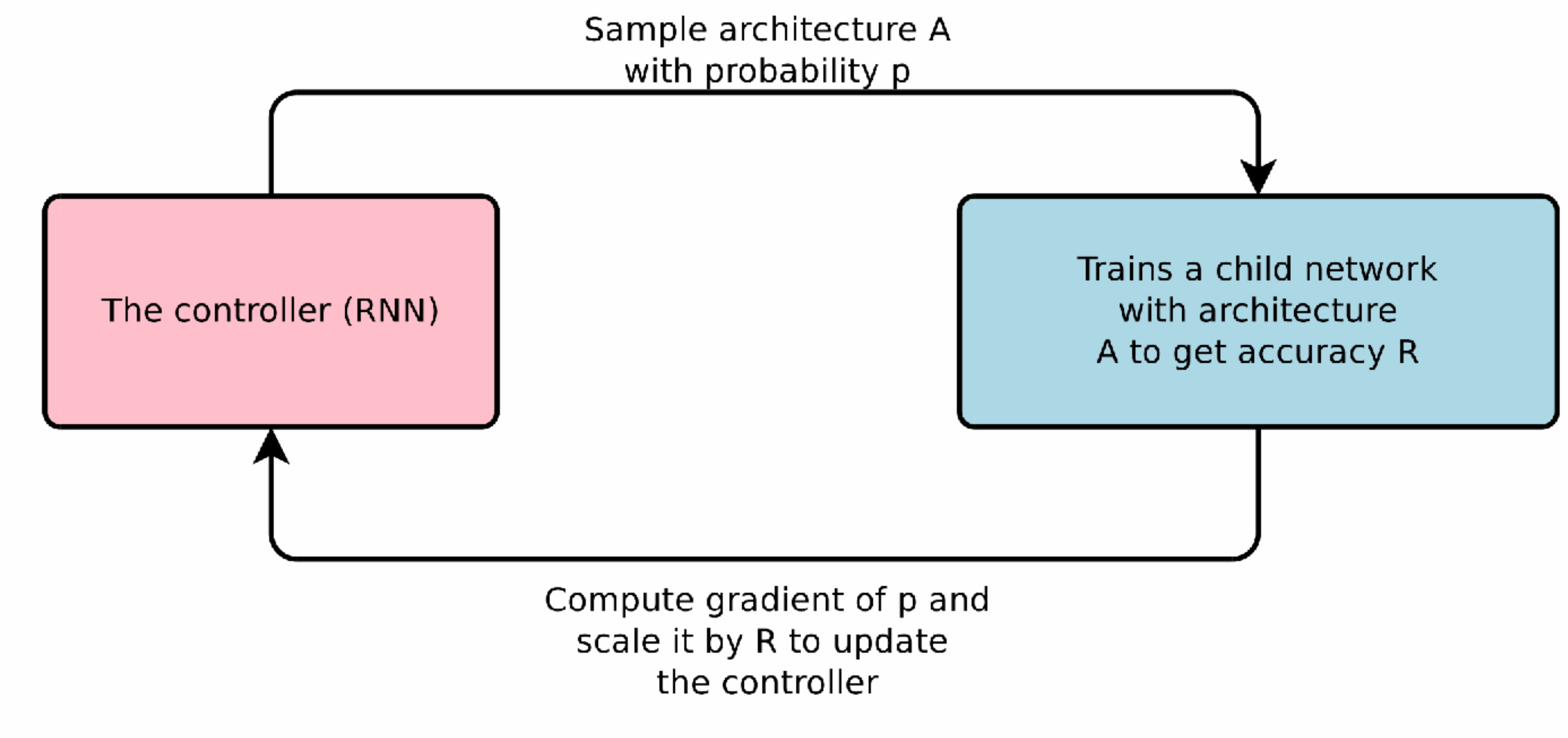


Neural Architecture Search

- Networks can be represented like a config string that specifies the structure and details of the network
- NAS uses a RNN “Controller” to generate these config strings that specify a “Child” network
- A parser then constructs the network based on this config string
- Train the child network for 50 epochs the check its accuracy on a validation set
- Then use Reinforcement Learning to update the “Controller” parameters based on the “Child” accuracy as a reward

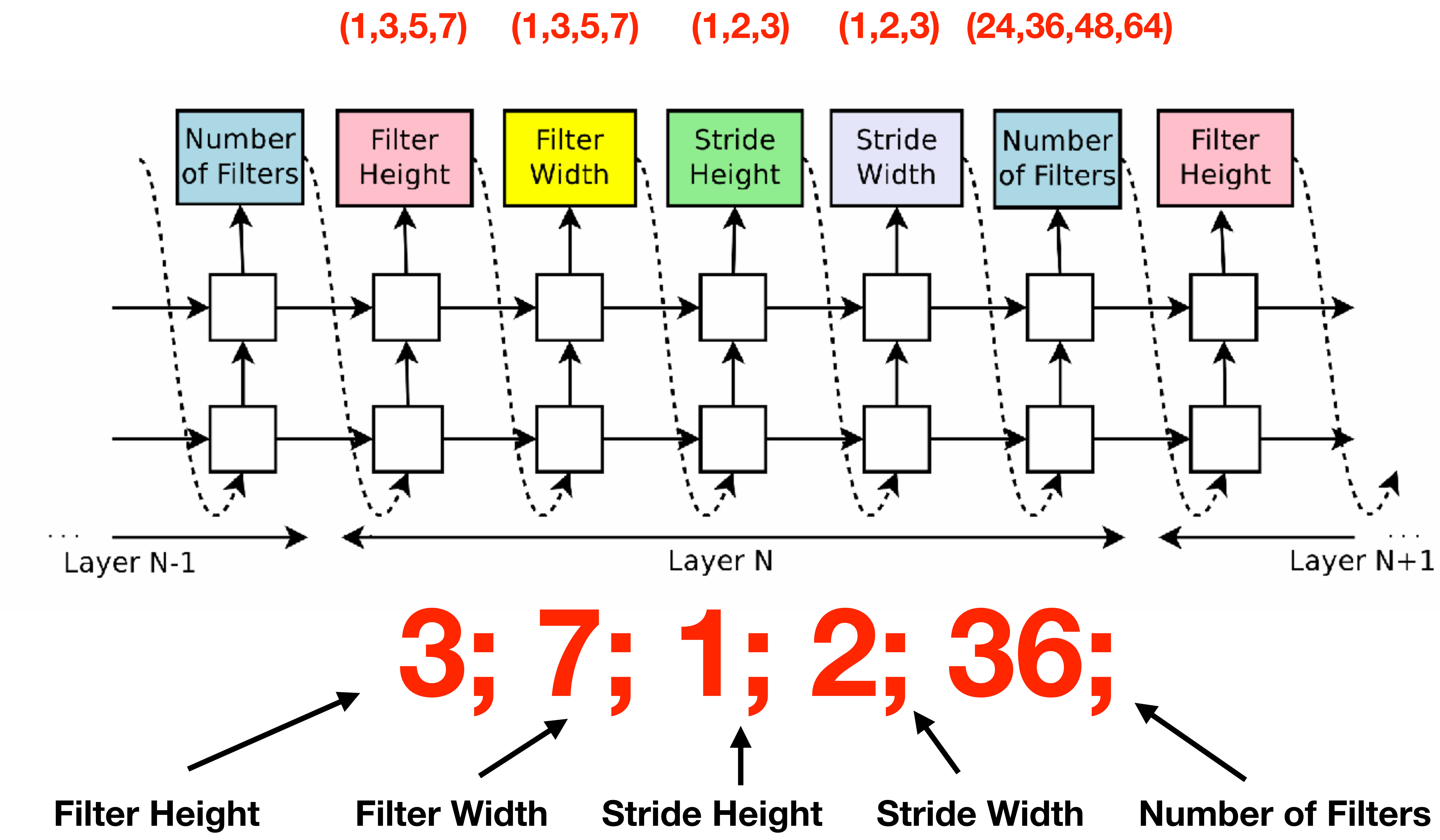
RED DRAGON AI

Neural Architecture Search



RED DRAGON AI

RNN Controller

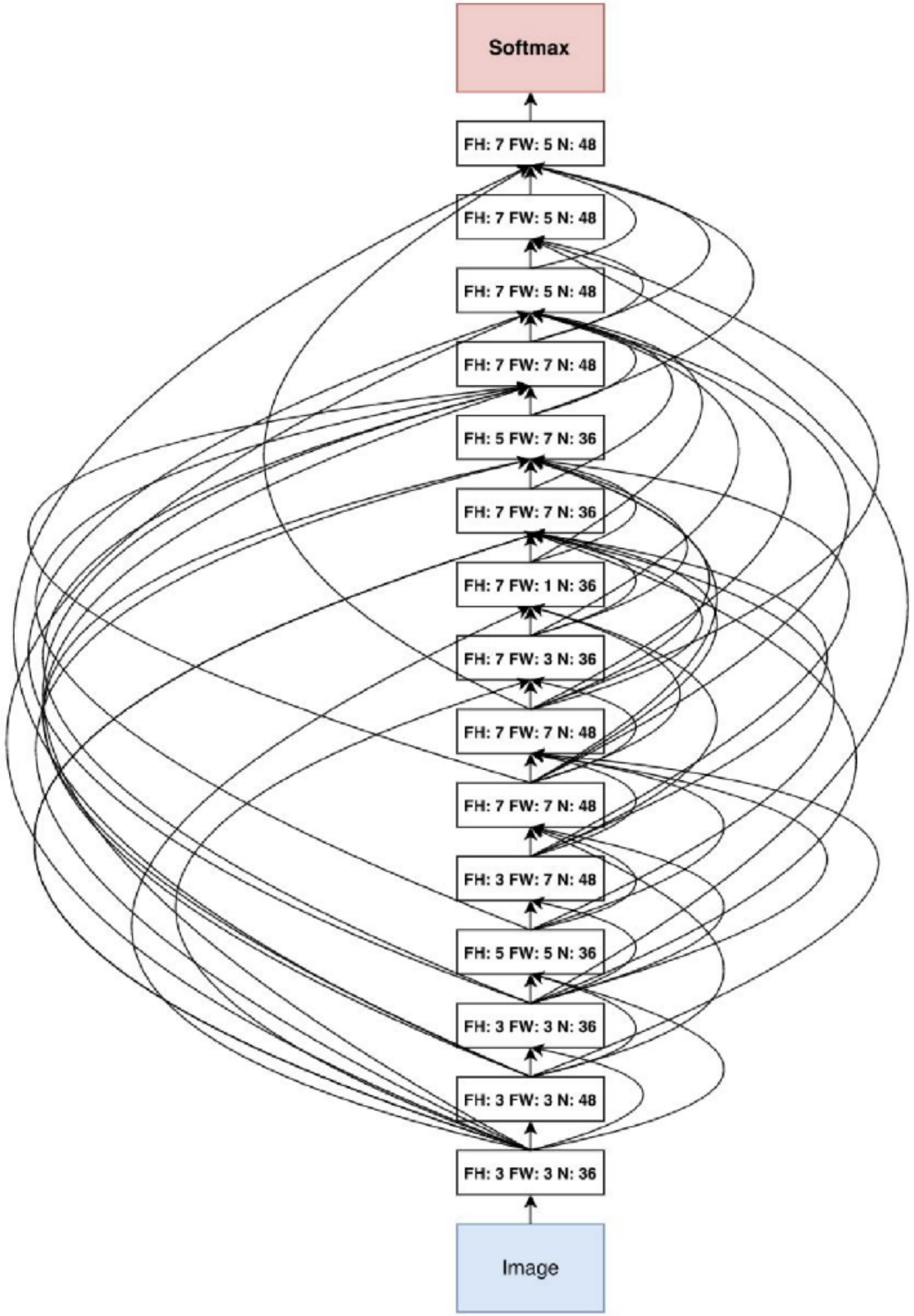


Quoc Le 2017

NAS for CiFAR 10

- 800 GPUs concurrently
- Parameter Server - ‘Controller’ Replicas & ‘Child’ Networks
- 50 Epochs for each model
- Controller is given the validation accuracy score as reward score
- Total of 12,800 models trained
- Gets SOTA results

RED DRAGON AI



The Final Child Network

CiFAR10 Results

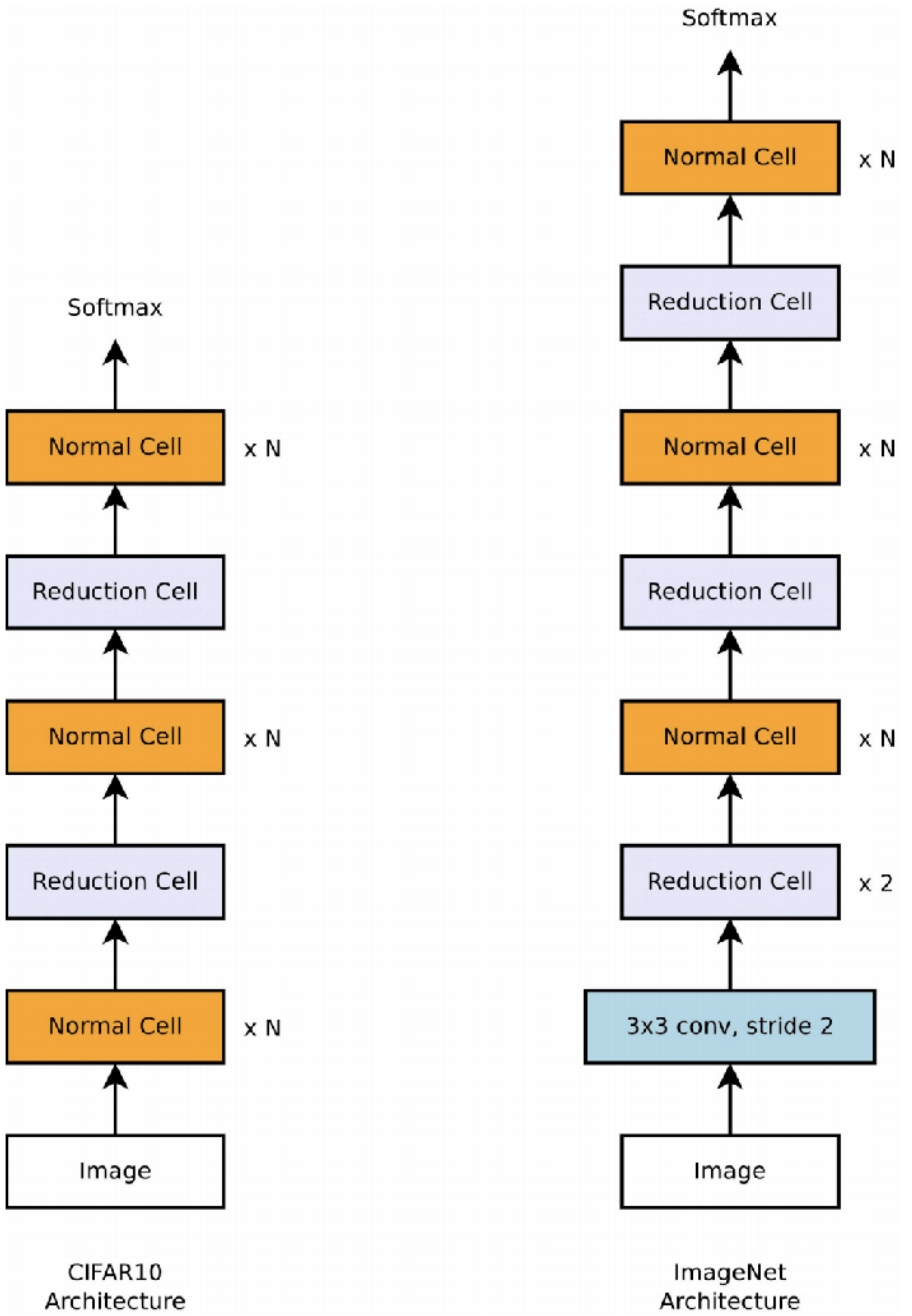
Model	Depth	Parameters	Error rate (%)
Network in Network (Lin et al., 2013)	-	-	8.81
All-CNN (Springenberg et al., 2014)	-	-	7.25
Deeply Supervised Net (Lee et al., 2015)	-	-	7.97
Highway Network (Srivastava et al., 2015)	-	-	7.72
Scalable Bayesian Optimization (Snoek et al., 2015)	-	-	6.37
FractalNet (Larsson et al., 2016)	21	38.6M	5.22
with Dropout/Drop-path	21	38.6M	4.60
ResNet (He et al., 2016a)	110	1.7M	6.61
ResNet (reported by Huang et al. (2016c))	110	1.7M	6.41
ResNet with Stochastic Depth (Huang et al., 2016c)	110	1.7M	5.23
	1202	10.2M	4.91
Wide ResNet (Zagoruyko & Komodakis, 2016)	16	11.0M	4.81
	28	36.5M	4.17
ResNet (pre-activation) (He et al., 2016b)	164	1.7M	5.46
	1001	10.2M	4.62
DenseNet $(L = 40, k = 12)$ Huang et al. (2016a)	40	1.0M	5.24
DenseNet$(L = 100, k = 12)$ Huang et al. (2016a)	100	7.0M	4.10
DenseNet $(L = 100, k = 24)$ Huang et al. (2016a)	100	27.2M	3.74
DenseNet-BC $(L = 100, k = 40)$ Huang et al. (2016b)	190	25.6M	3.46
Neural Architecture Search v1 no stride or pooling	15	4.2M	5.50
Neural Architecture Search v2 predicting strides	20	2.5M	6.01
Neural Architecture Search v3 max pooling	39	7.1M	4.47
Neural Architecture Search v3 max pooling + more filters	39	37.4M	3.65

arXiv:1611.01578

NAS for Imagenet

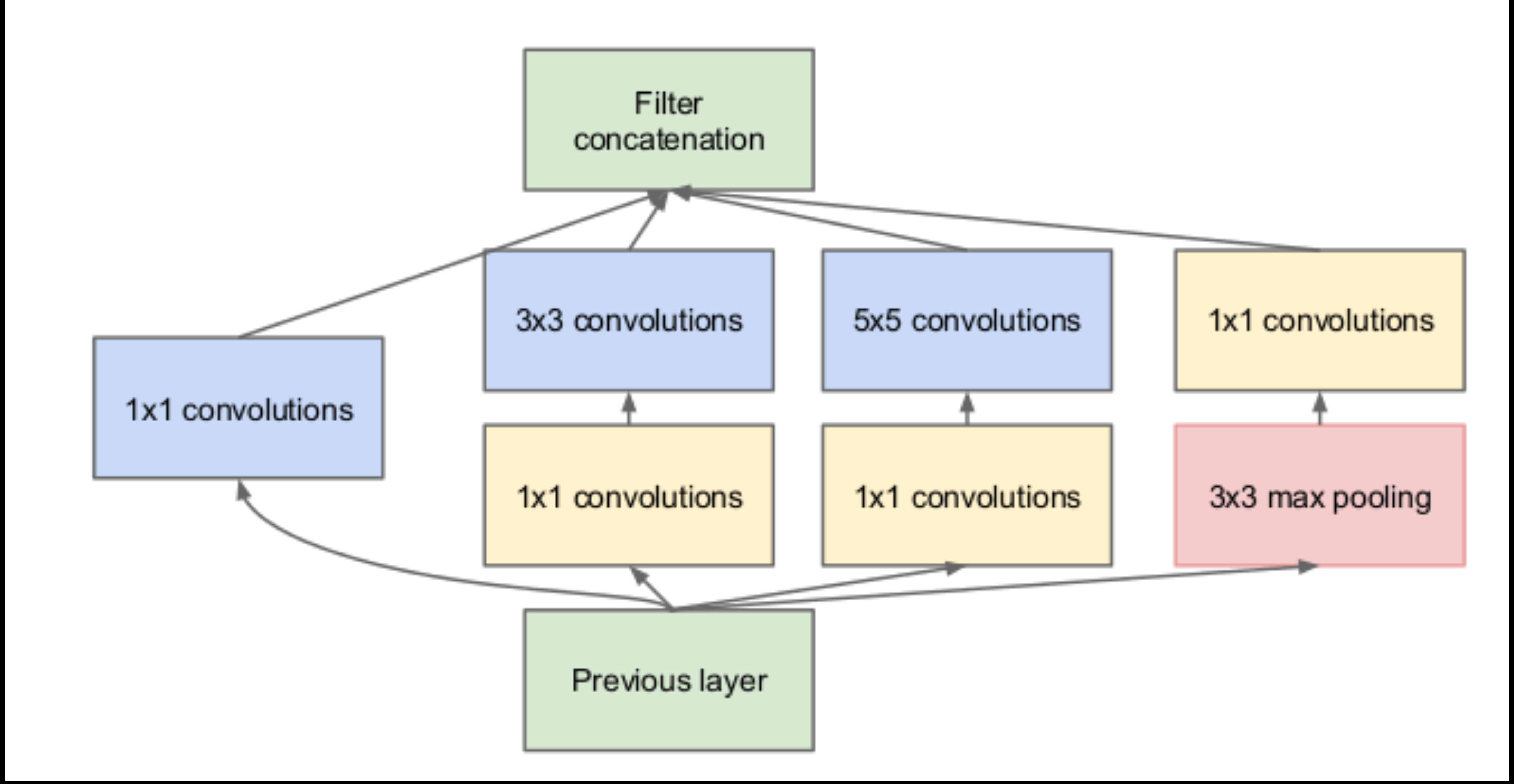
- A lot more GPUs would be needed
- Goal here is to find a cell (akin to an InceptionCell)
- Assemble a network (layer ordering) with these cells

RED DRAGON AI



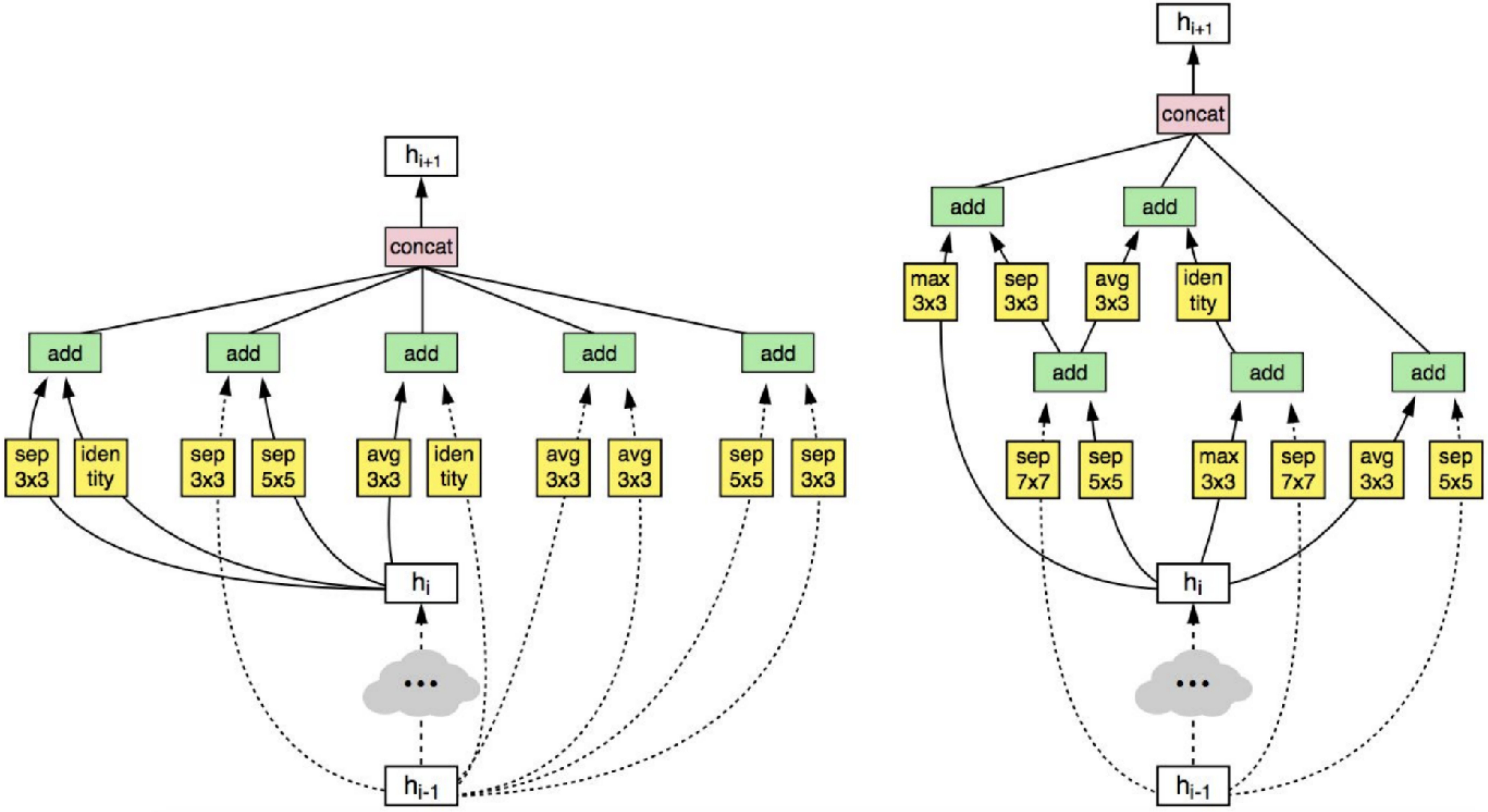
NASNet layers

Inception Cell



RED DRAGON AI

NASNet cells



Normal Cell

Reduction Cell

CiFAR10 Cell Results

model	depth	# params	error rate (%)
DenseNet ($L = 40, k = 12$) [26]	40	1.0M	5.24
DenseNet($L = 100, k = 12$) [26]	100	7.0M	4.10
DenseNet ($L = 100, k = 24$) [26]	100	27.2M	3.74
DenseNet-BC ($L = 100, k = 40$) [26]	190	25.6M	3.46
Shake-Shake 26 2x32d [18]	26	2.9M	3.55
Shake-Shake 26 2x96d [18]	26	26.2M	2.86
Shake-Shake 26 2x96d + cutout [12]	26	26.2M	2.56
NAS v3 [70]	39	7.1M	4.47
NAS v3 [70]	39	37.4M	3.65
NASNet-A (6 @ 768)	-	3.3M	3.41
NASNet-A (6 @ 768) + cutout	-	3.3M	2.65
NASNet-A (7 @ 2304)	-	27.6M	2.97
NASNet-A (7 @ 2304) + cutout	-	27.6M	2.40
NASNet-B (4 @ 1152)	-	2.6M	3.73
NASNet-C (4 @ 640)	-	3.1M	3.59

arXiv:1611.01578

NASNet ImageNet Results

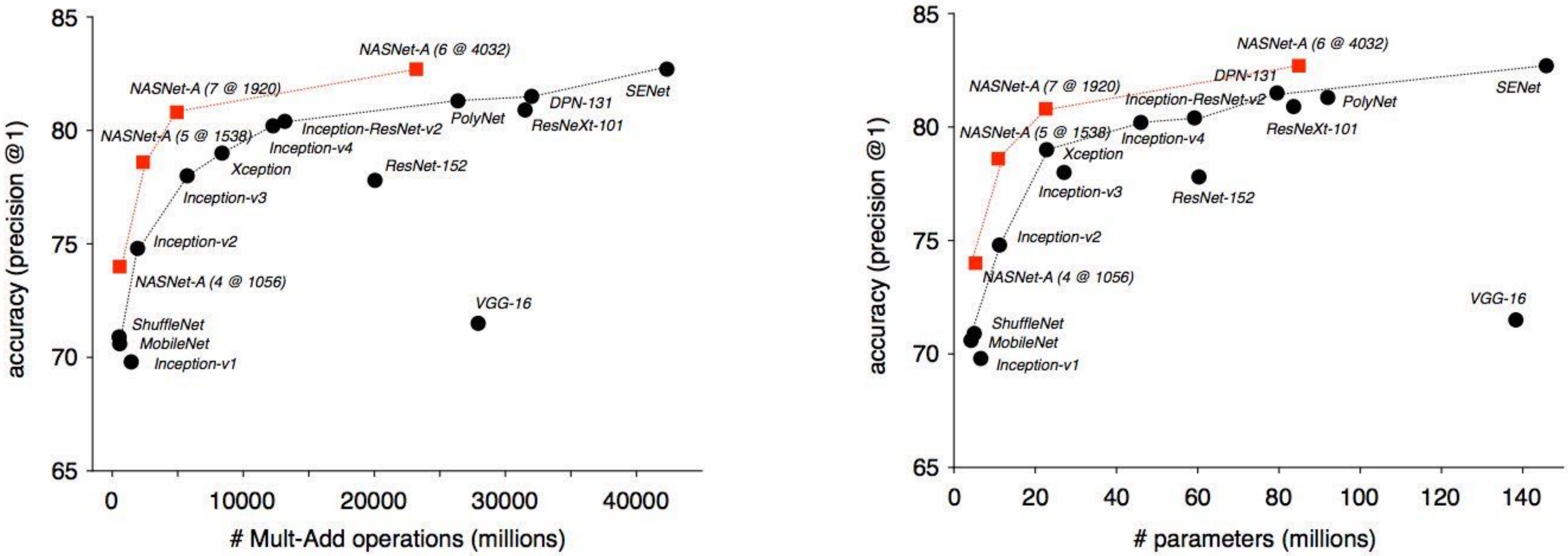


Figure 5. Accuracy versus computational demand (left) and number of parameters (right) across top performing published CNN architectures on ImageNet 2012 ILSVRC challenge prediction task. Computational demand is measured in the number of floating-point multiply-add operations to process a single image. Black circles indicate previously published work and red squares highlight our proposed models.

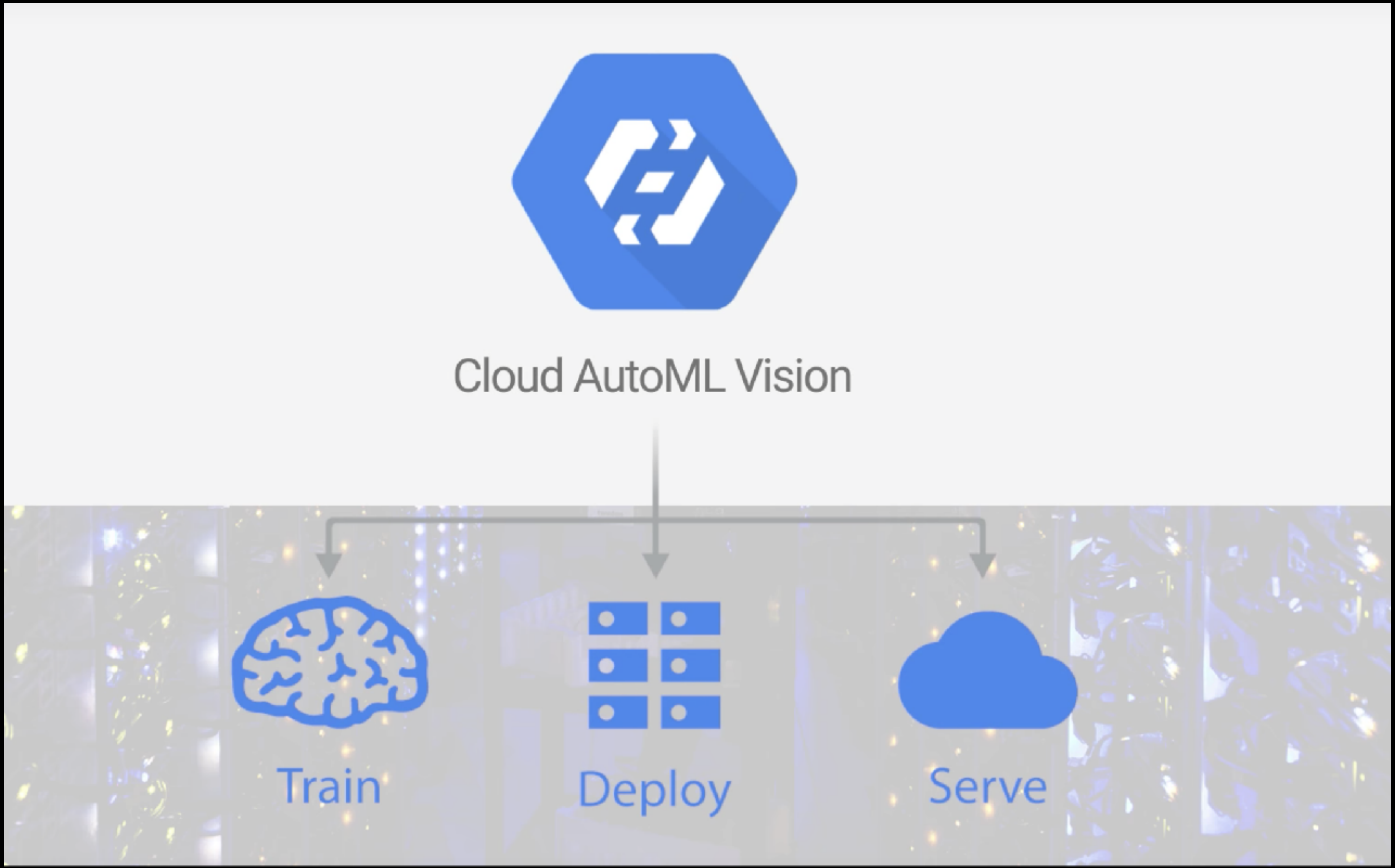
Updates

Efficient Neural Architecture Search

- Brand New for ICLR2018
- Reduces the need for GPUs by a factor of 1000
- Whole model trained overnight on a 1080Ti
- Big key difference is it doesn't train from scratch each time it shares weights from previous iterations
- It treats the search space as on big graph and samples from that

RED DRAGON AI

Google Cloud AutoML



RED DRAGON AI

Key Papers/Articles

- Neural Architecture Search - https://arxiv.org/abs/1611.01578
- AutoML Google Research - https://research.googleblog.com/2017/11/automl-for-large-scale-image.html
- Efficient Neural Architecture Search - https://arxiv.org/abs/1802.03268
- Learning to learn by gradient descent by gradient descent https://arxiv.org/abs/1606.04474

RED DRAGON AI

NASnet implementations

- TF-Slim - https://github.com/tensorflow/models/tree/master/research/slim/nets/nasnet
- Keras - https://github.com/titu1994/Keras-NASNet
- NAScell/RNN - https://www.tensorflow.org/api_docs/python/tf/contrib/rnn/NASCell

RED DRAGON AI

Up coming events

- TensorFlow & Deep Learning Back to Basics: CNNs - SGInnovate March 6th
- TensorFlow & Deep Learning: - TBA March 22th
- AI Day: Mid April

RED DRAGON AI