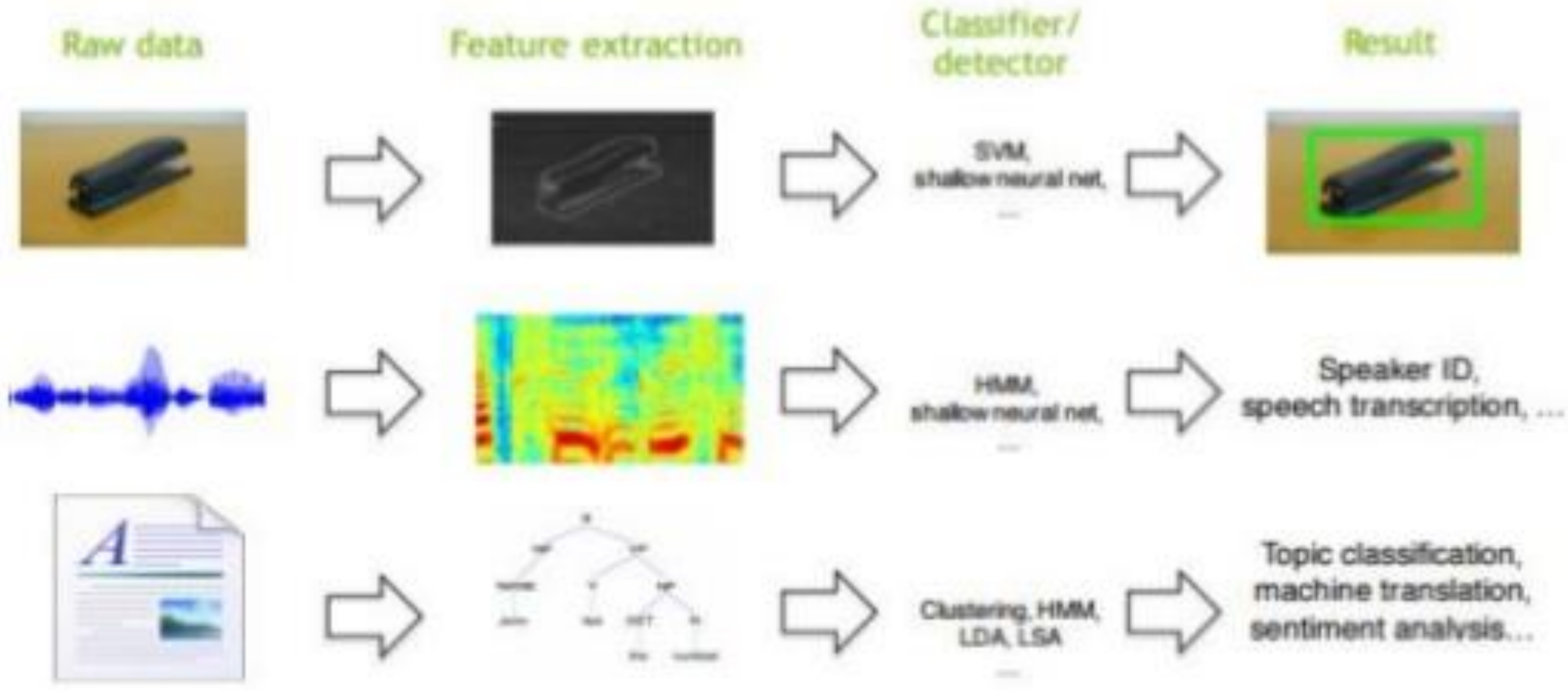
Neural Architecture Search with Reinforcement Learning

Jim Liu

ICLR2017

Traditional Machine Perception

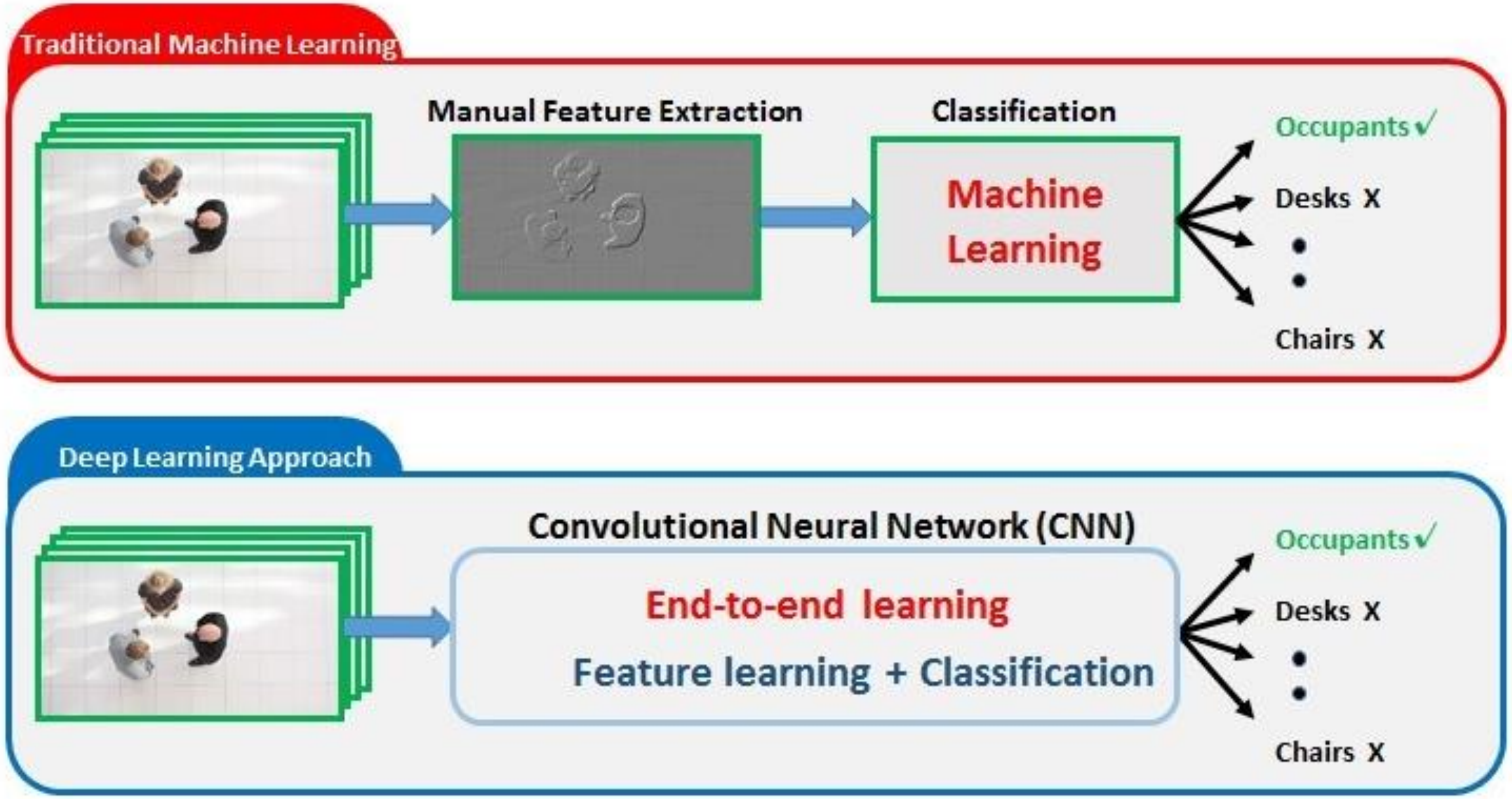
Hand crafted feature extractors



*https://www.slideshare.net/kuanhoong/big-data-malaysia-a-primer-on-deep-learning

Machine Learning V.S. Deep Learning

deep networks we can perform feature extraction and classification in one shot.

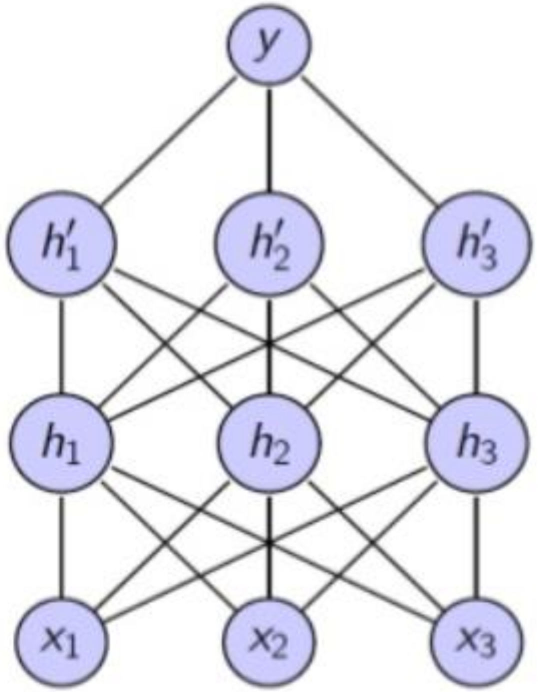


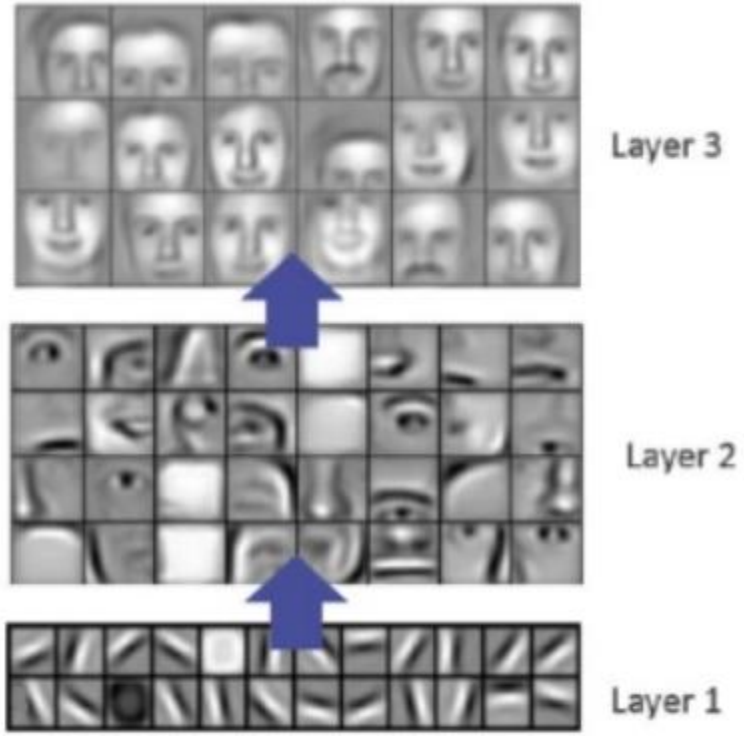
Traditional Machine Learning vs. Deep Learning

*https://www.wirelessdesignmag.com/article/2016/08/applying-deep-learning-edge-analytics-sensors-building-automation

Machine Learning V.S. Deep Learning

deep networks we can perform feature extraction and classification in one shot.

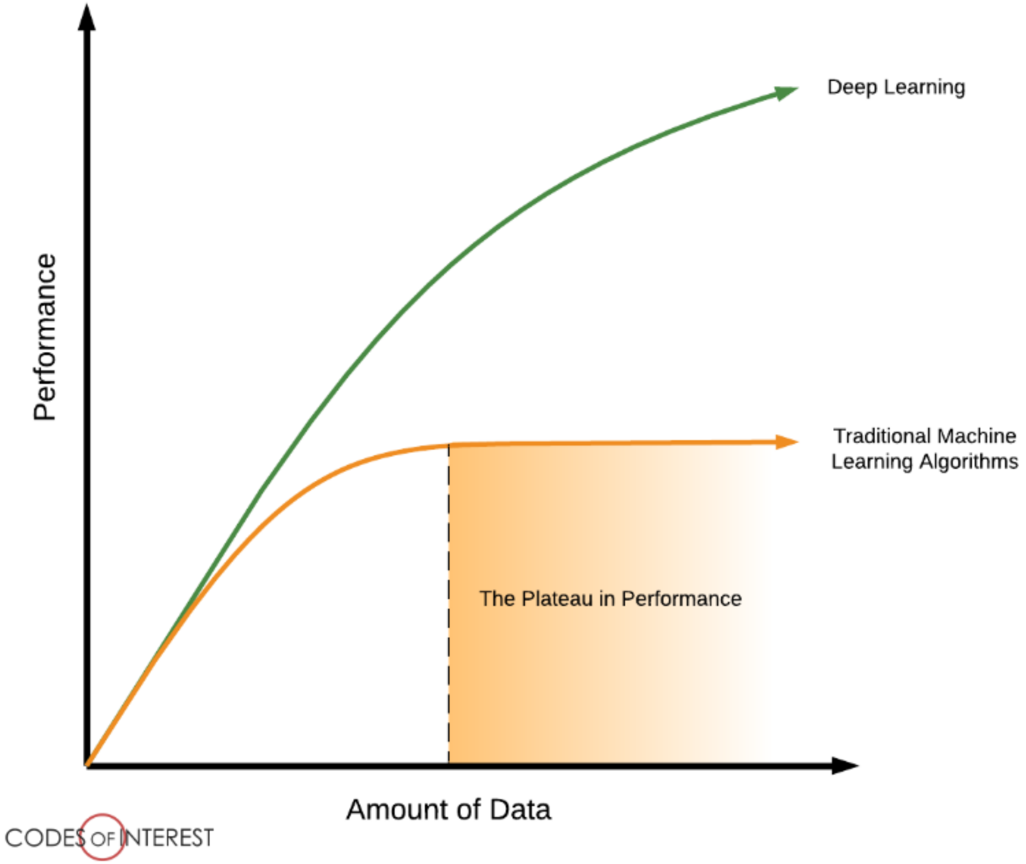




*https://www.google.com.tw/url?sa=i&source=images&cd=&ved=2ahUKEwiqnrCA2qrcAhXIkpQKHepaBBQQjhx6BAgBEAM&url=https%3A%2F%2Fwww.slideshare.net%2Fmravendi%2Fintro-deep-learning&psig=AOvVaw3l8bhj0cl_AahIythbq8PC&ust=1532073514111634

Machine Learning V.S. Deep Learning

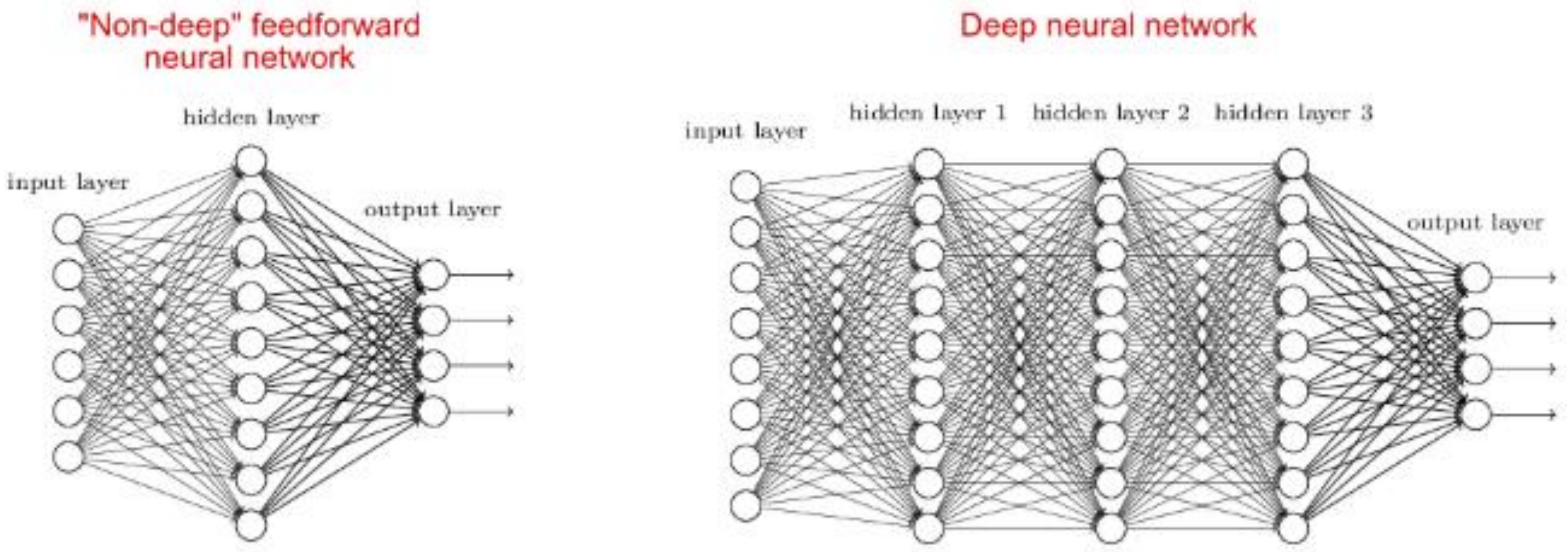
We saw that Deep Learning algorithms don't have a "plateau in performance" compared to traditional machine learning algorithms



*https://www.codesofinterest.com/2017/04/how-deep-should-deep-learning-be.html

Neural network V.S. Deep Learning

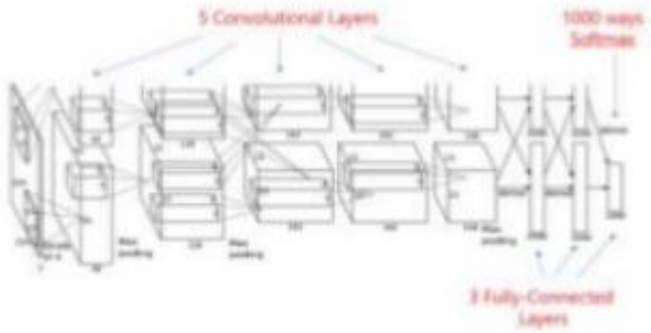
Deep neural network is simply a feedforward network with many hidden layers



*https://stats.stackexchange.com/questions/182734/what-is-the-difference-between-a-neural-network-and-a-deep-neural-network-and-w

Network Architectures: Which one is better?

AlexNet



VGG

GoogLeNet

ResNet

*https://www.google.com.tw/url?sa=i&source=images&cd=&ved=2ahUKEwiR0-fR6arcAhUGwbwKHR9WBSYQjhx6BAgBEAM&url=https%3A%2F%2Fwww.slideshare.net%2Fsamchoi7%2Fcnn-tutorial-66719728&psig=AOvVaw3jL88DQsXCj0XovPEIdMMv&ust=1532077977363109

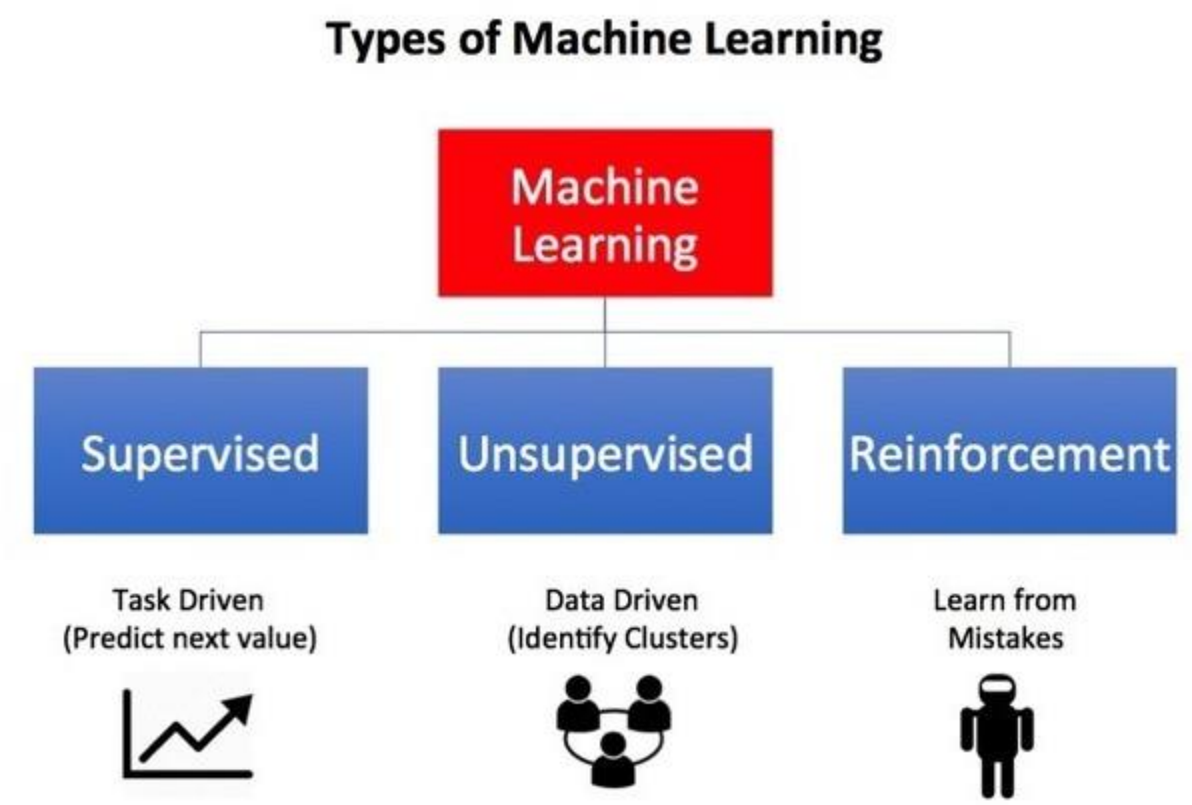
Put Problem Into Model Architecture Design

Handcraft Feature Extraction → **Model Architecture Design**

Neural architecture search

- RL-based architecture search
- Model architecture evolution
- learning to learn or meta-learning
- #mean field theory and dynamic isometry

Reinforcement Learning



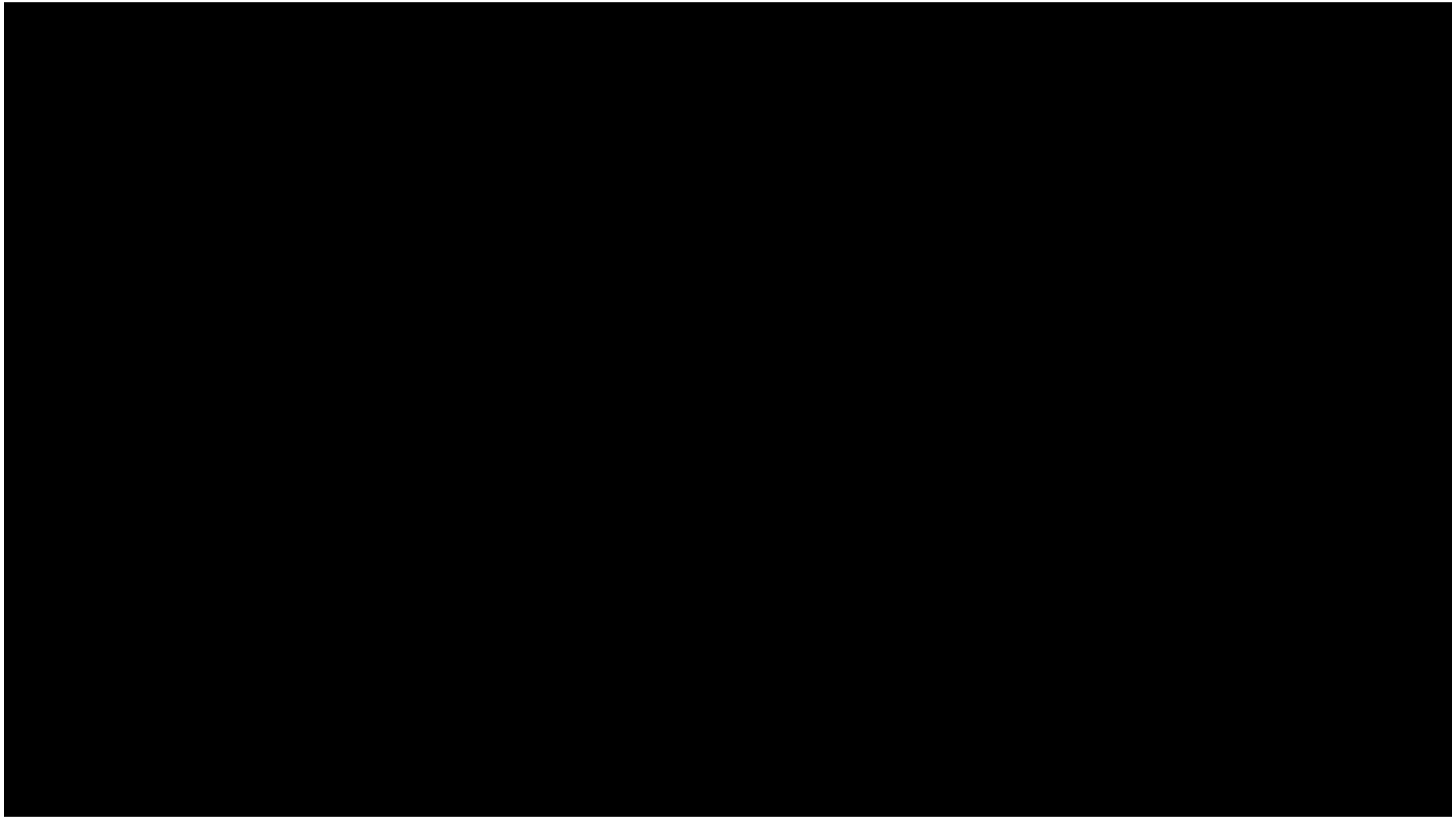
*https://medium.com/@shweta_bhatt/reinforcement-learning-101-e24b50e1d292

Reinforcement Learning

1. **Environment —** Physical world in which the agent operates
2. **State —** Current situation of the agent
3. **Reward —** Feedback from the environment
4. **Policy —** Method to map agent's state to actions
5. **Value —** Future reward that an agent would receive by taking an action in a particular state

*https://medium.com/@shweta_bhatt/reinforcement-learning-101-e24b50e1d292

Reinforcement Learning

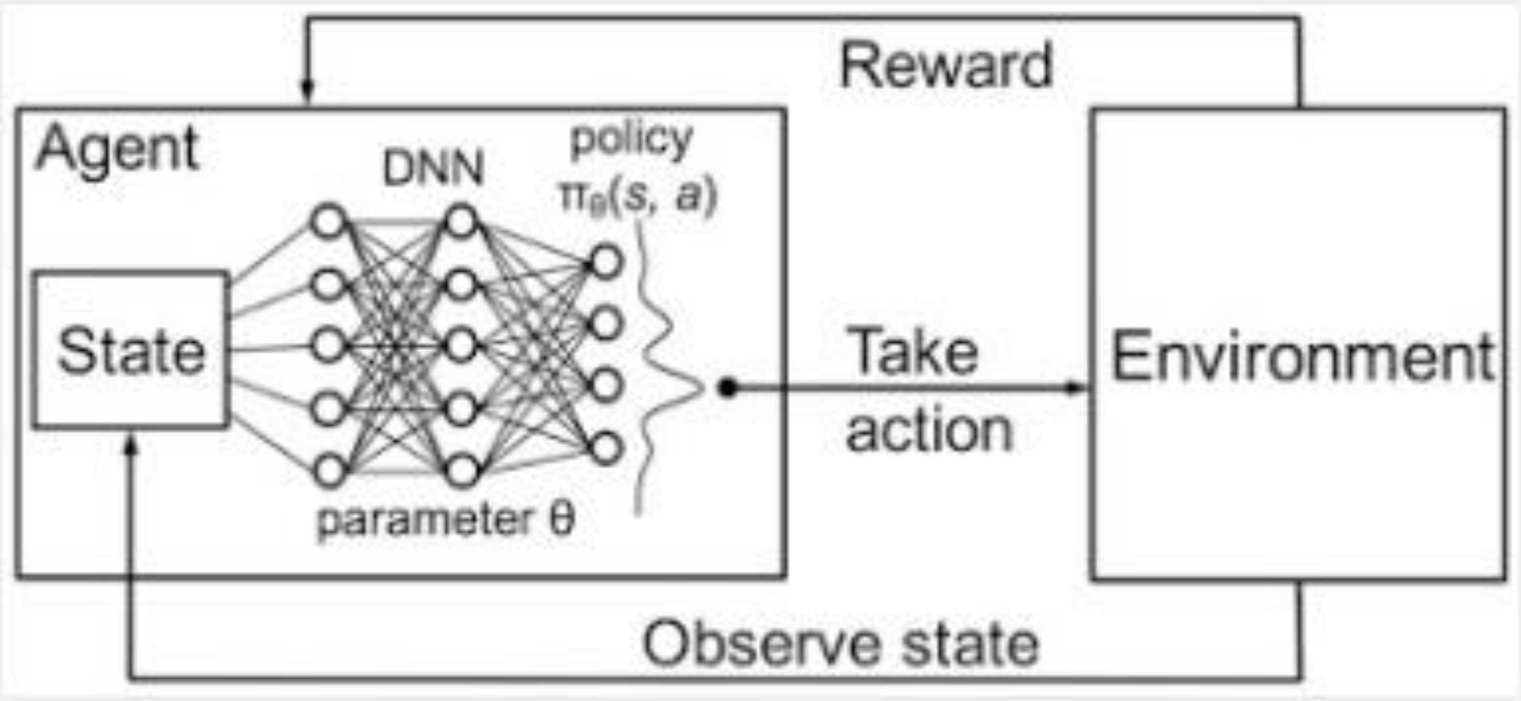


*https://medium.com/@shweta_bhatt/reinforcement-learning-101-e24b50e1d292

Reinforcement Learning

*https://www.google.com.tw/url?sa=i&source=images&cd=&ved=2ahUKEwjNwIidj7TcAhVDH5QKHb9wBqkQjhx6BAgBEAM&url=%2Furl%3Fsa%3Di%26source%3Dimages%26cd%3D%26ved%3D%26url%3Dhttps%253A%252F%252Fdzone.com%252Farticles%252Freinforcement-learning-for-the-enterprise%26psig%3DAOvVaw0LT5rvSl2X4qax8lFyK6kt%26ust%3D1532397357086253&psig=AOvVaw0LT5rvSl2X4qax8lFyK6kt&ust=1532397357086253

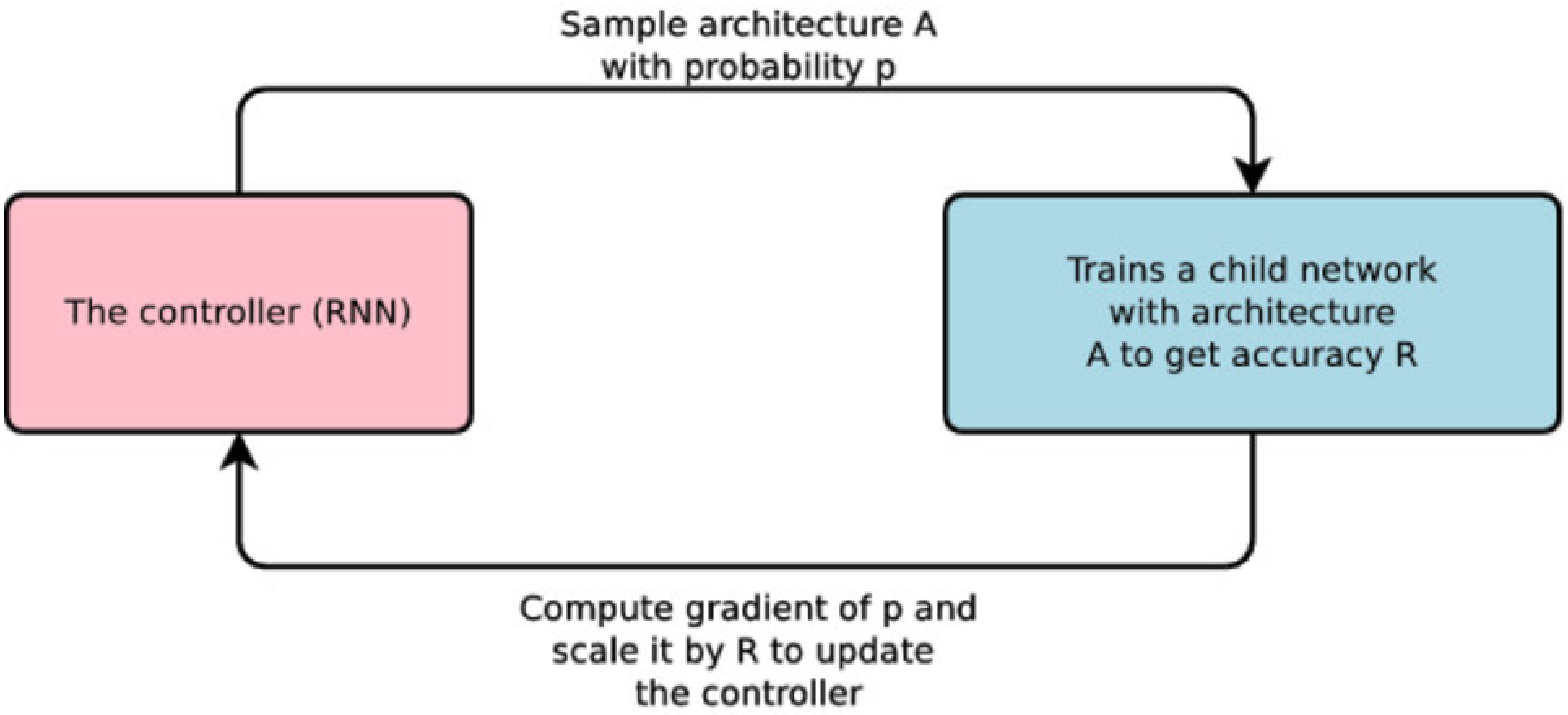
Reinforcement Learning



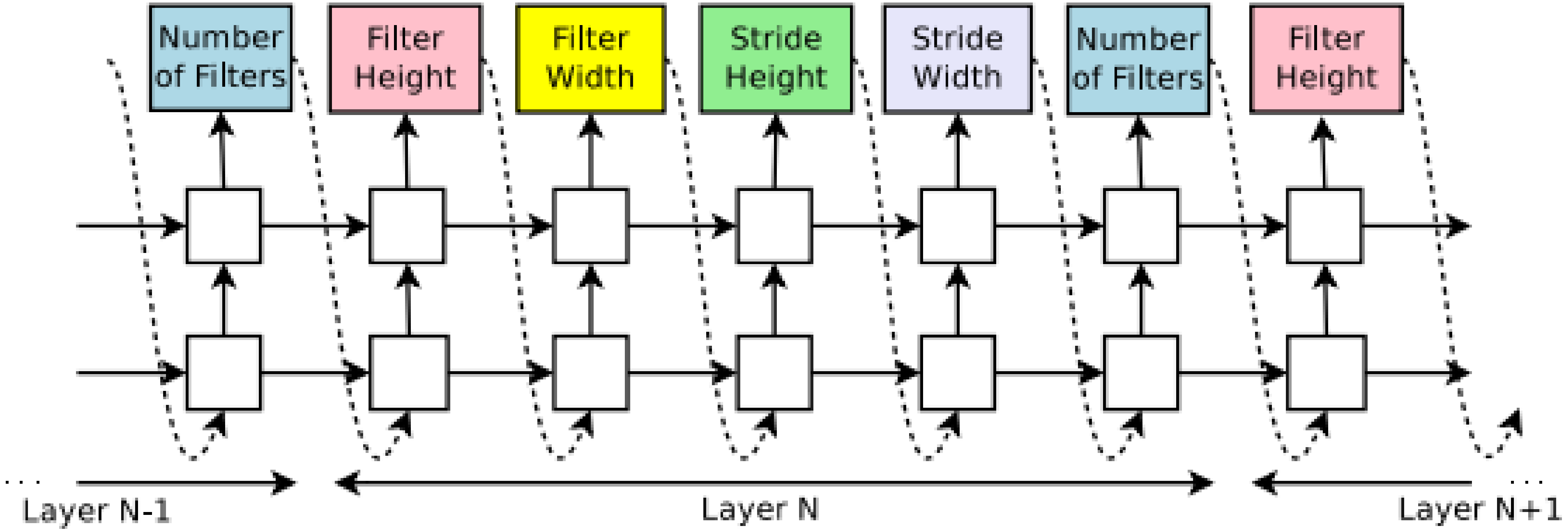
*http://people.csail.mit.edu/hongzi/content/publications/DeepRM-HotNets16.pdf

Paper solution

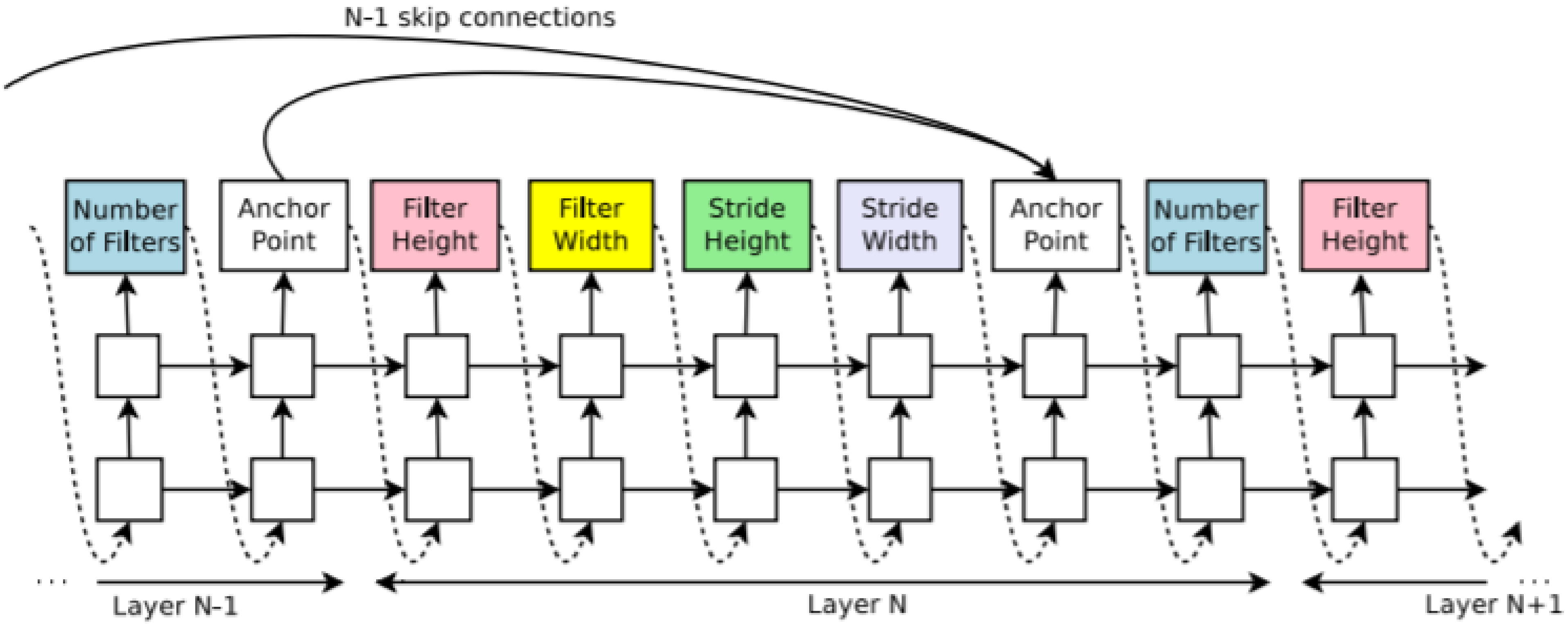
Neural Architecture Search with Reinforcement Learning



How our controller recurrent neural network samples a simple convolutional network

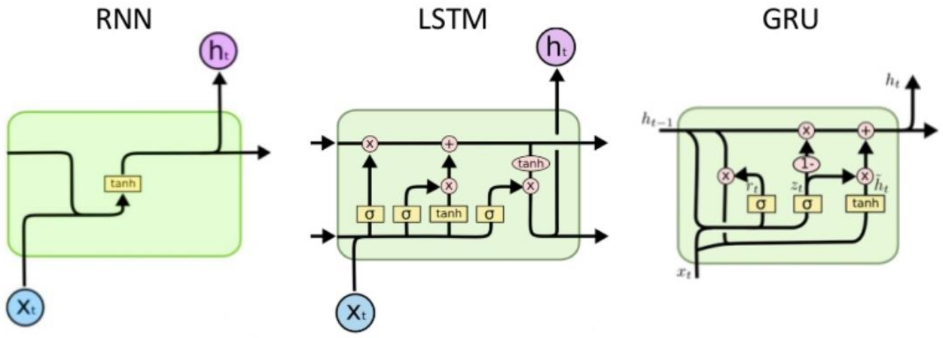


The controller uses anchor points, and set-selection attention to form skip connections.

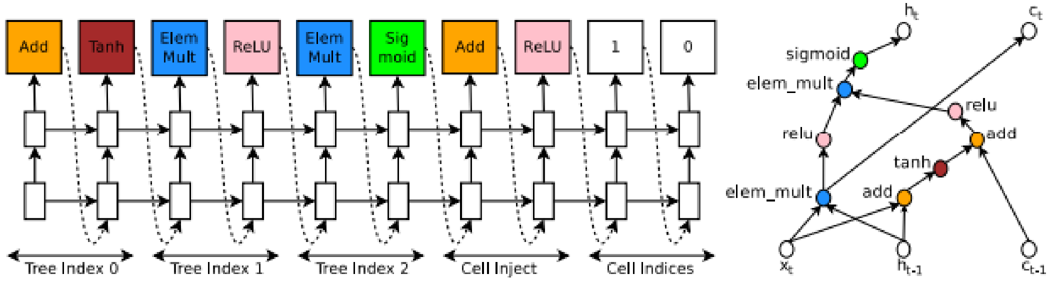


Generate Recurrent Cell Architectures

Classic



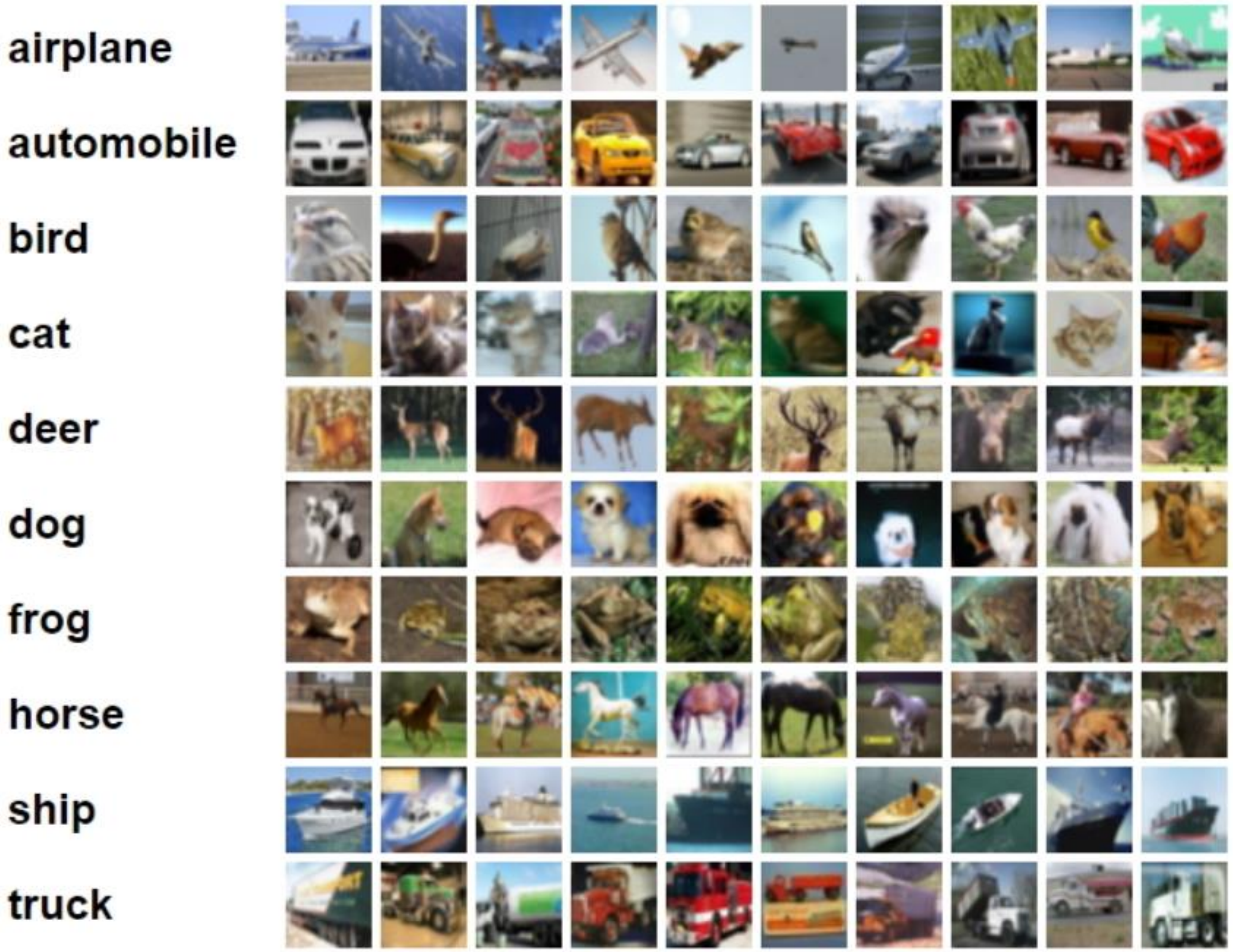
Reinforcement



*https://www.slideshare.net/YanKang/rnn-explore-71268007

Experiments and Results

LEARNING CONVOLUTIONAL ARCHITECTURES FOR CIFAR-10



Experiments and Results

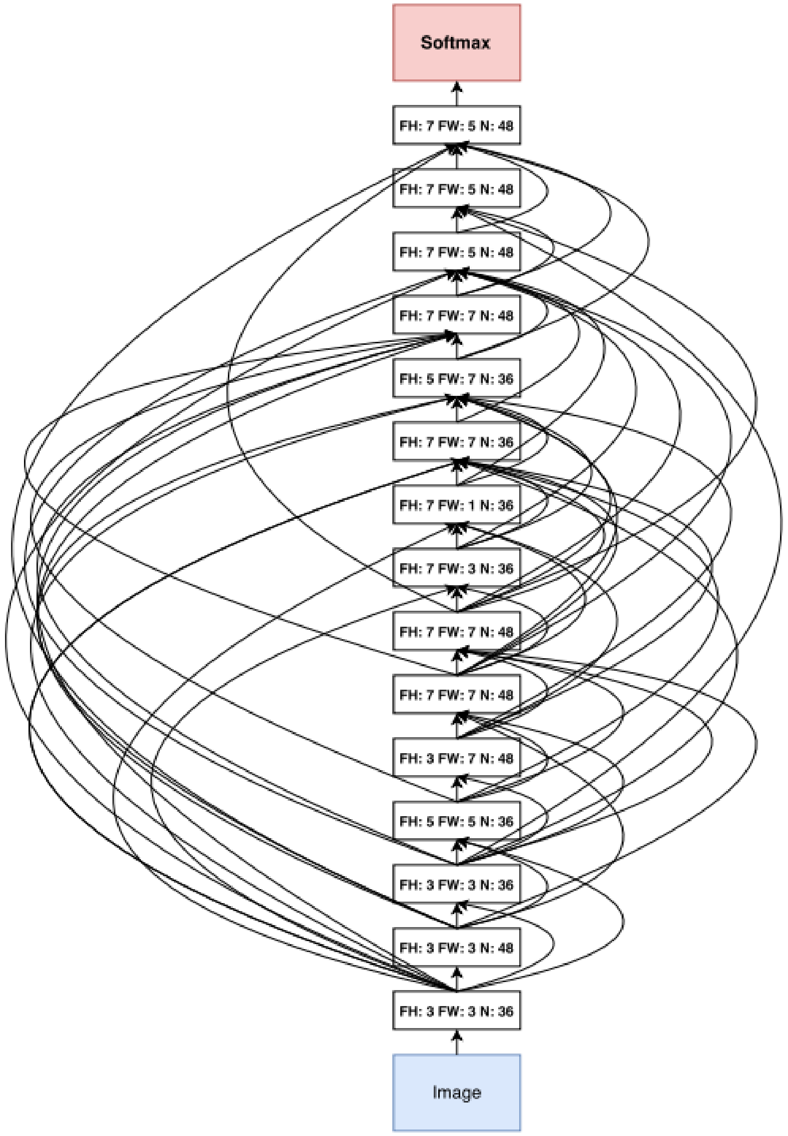
LEARNING CONVOLUTIONAL ARCHITECTURES FOR CIFAR-10

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

- The controller RNN is a two-layer LSTM with 35 hidden units on each layer. It is trained with the ADAM optimizer with a learning rate of 0.0006.
- After the controller trains 12,800 architectures, we find the architecture that achieves the best validation accuracy.
- The DenseNet model that achieves 3.46% error rate (Huang et al., 2016b) uses 1x1 convolutions to reduce its total number of parameters, which we did not do, so it is not an exact comparison.

Experiments an

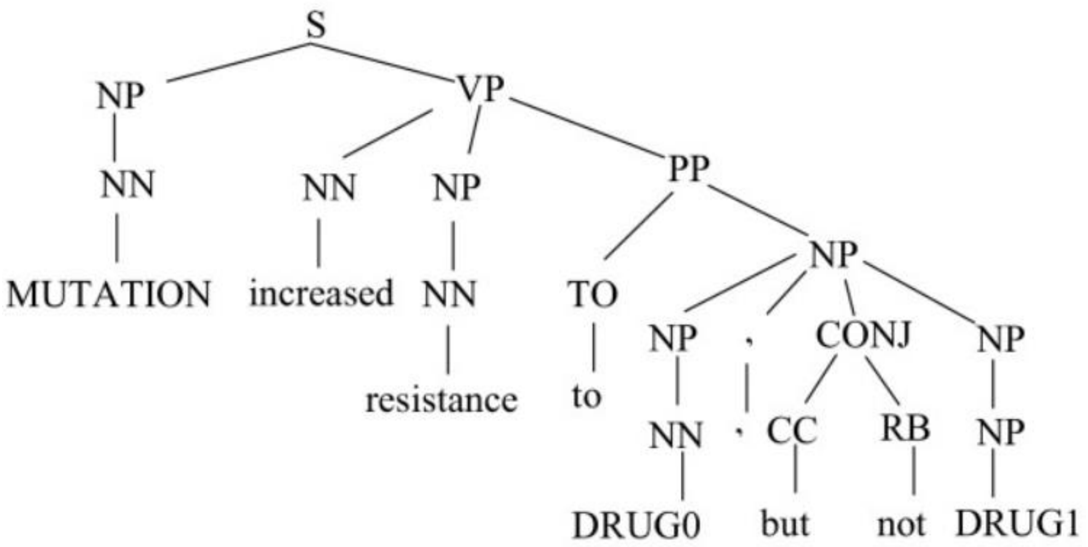
LEARNING CONVOLUTION



Experiments and Results

LEARNING RECURRENT CELLS FOR PENN TREEBANK

1.	CC	Coordinating conjunction	25.	TO	*to*
2.	CD	Cardinal number	26.	UH	Interjection
3.	DT	Determiner	27.	VB	Verb, base form
4.	EX	Existential *there*	28.	VBD	Verb, past tense
5.	FW	Foreign word	29.	VBG	Verb, gerund/present participle
6.	IN	Preposition/subordinating conjunction	30.	VBN	Verb, past participle
7.	JJ	Adjective	31.	VBP	Verb, non-3rd ps. sing. present
8.	JJR	Adjective, comparative	32.	VBZ	Verb, 3rd ps. sing. present
9.	JJS	Adjective, superlative	33.	WDT	*wh*-determiner
10.	LS	List item marker	34.	WP	*wh*-pronoun
11.	MD	Modal	35.	WP$	Possessive *wh*-pronoun
12.	NN	Noun, singular or mass	36.	WRB	*wh*-adverb
13.	NNS	Noun, plural	37.	#	Pound sign
14.	NNP	Proper noun, singular	38.	$	Dollar sign
15.	NNPS	Proper noun, plural	39.	.	Sentence-final punctuation
16.	PDT	Predeterminer	40.	,	Comma
17.	POS	Possessive ending	41.	:	Colon, semi-colon
18.	PRP	Personal pronoun	42.	(	Left bracket character
19.	PP$	Possessive pronoun	43.	)	Right bracket character
20.	RB	Adverb	44.	"	Straight double quote
21.	RBR	Adverb, comparative	45.	‘	Left open single quote
22.	RBS	Adverb, superlative	46.	“	Left open double quote
23.	RP	Particle	47.	’	Right close single quote
24.	SYM	Symbol (mathematical or scientific)	48.	”	Right close double quote



Experiments and Results

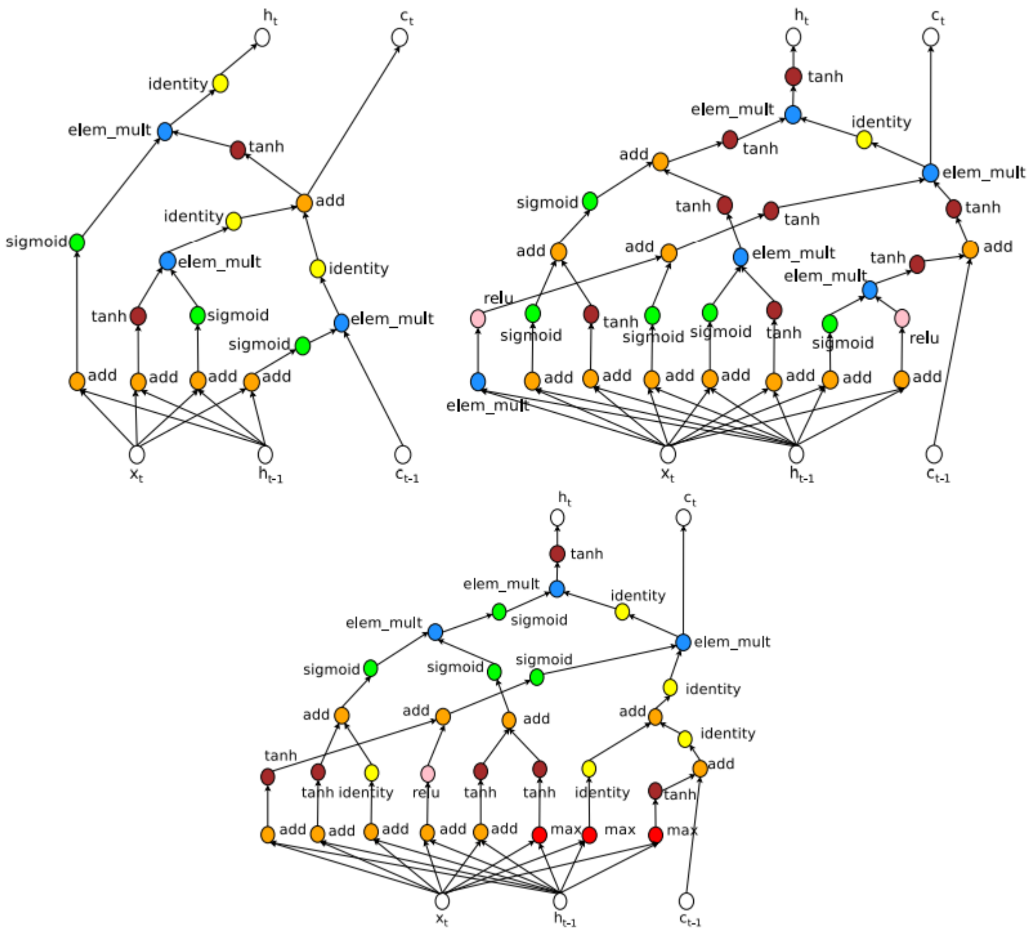
LEARNING RECURRENT CELLS FOR PENN TREEBANK

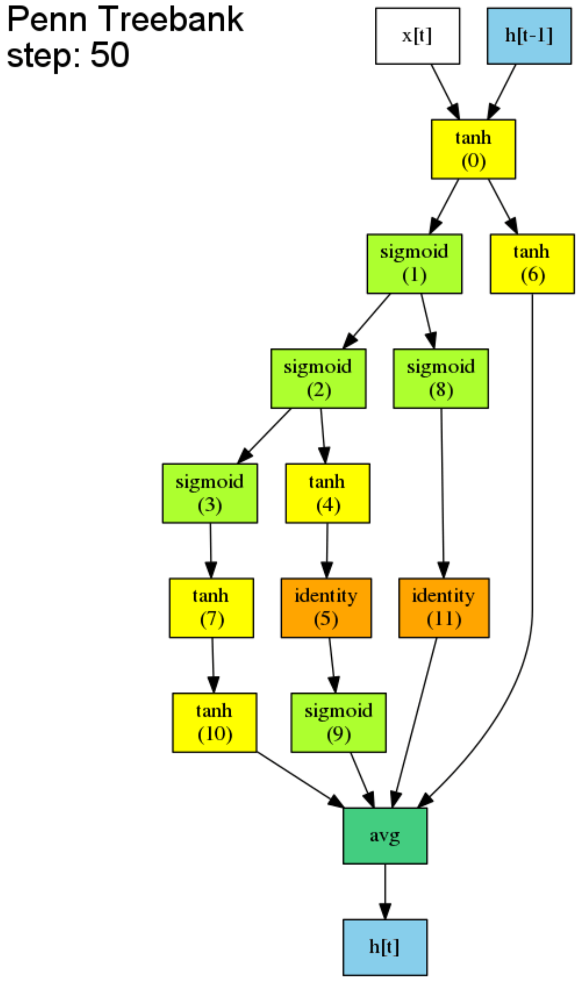
Model	Parameters	Test Perplexity
Mikolov & Zweig (2012) - KN-5	2M‡	141.2
Mikolov & Zweig (2012) - KN5 + cache	2M‡	125.7
Mikolov & Zweig (2012) - RNN	6M‡	124.7
Mikolov & Zweig (2012) - RNN-LDA	7M‡	113.7
Mikolov & Zweig (2012) - RNN-LDA + KN-5 + cache	9M‡	92.0
Pascanu et al. (2013) - Deep RNN	6M	107.5
Cheng et al. (2014) - Sum-Prod Net	5M‡	100.0
Zaremba et al. (2014) - LSTM (medium)	20M	82.7
Zaremba et al. (2014) - LSTM (large)	66M	78.4
Gal (2015) - Variational LSTM (medium, untied)	20M	79.7
Gal (2015) - Variational LSTM (medium, untied, MC)	20M	78.6
Gal (2015) - Variational LSTM (large, untied)	66M	75.2
Gal (2015) - Variational LSTM (large, untied, MC)	66M	73.4
Kim et al. (2015) - CharCNN	19M	78.9
Press & Wolf (2016) - Variational LSTM, shared embeddings	51M	73.2
Merity et al. (2016) - Zoneout + Variational LSTM (medium)	20M	80.6
Merity et al. (2016) - Pointer Sentinel-LSTM (medium)	21M	70.9
Inan et al. (2016) - VD-LSTM + REAL (large)	51M	68.5
Zilly et al. (2016) - Variational RHN, shared embeddings	24M	66.0
Neural Architecture Search with base 8	32M	67.9
Neural Architecture Search with base 8 and shared embeddings	25M	64.0
Neural Architecture Search with base 8 and shared embeddings	54M	62.4

- The controller and its training are almost identical to the CIFAR-10 experiments except for a few modifications: 1. the learning rate for the controller RNN is 0.0005.
- Not only is our cell is better, the model that achieves 64 perplexity is also more than two times faster because the previous best network requires running a cell 10 times per time step.

Experiments and Results

LEARNING RECURRENT CELLS FOR PENN TREEBANK





Automated Data Science & Machine Learning
Is coming…