Neural Architecture Search

Kale-ab Tessera, Dr. Benjamin Rosman

12 August 2019

R A I L
L A B

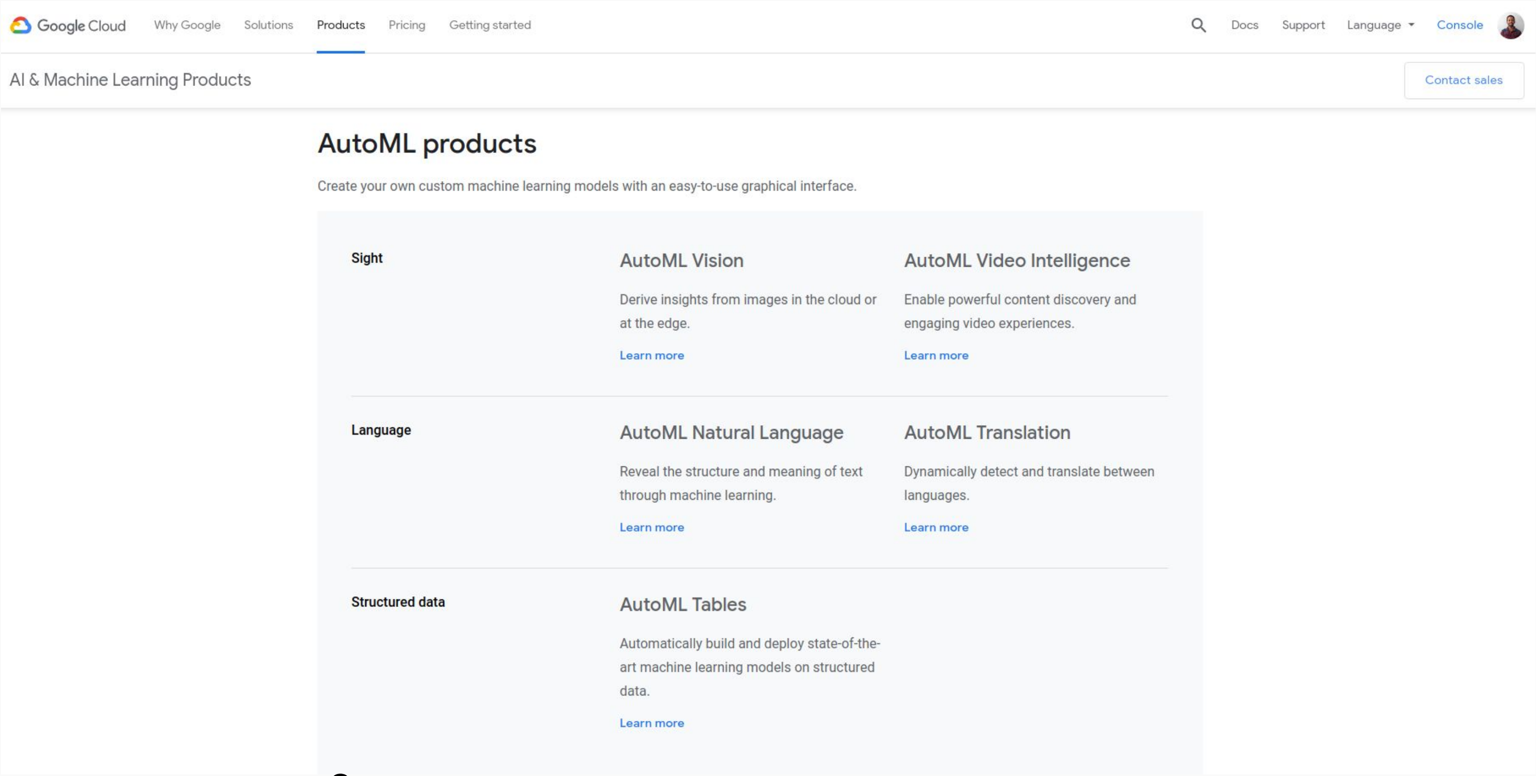
WITS
UNIVERSITY

Automated Machine Learning (AutoML) - What?

AutoMl is a field that aims to automate all parts of applying machine learning algorithms.
Examples of what we want to automate:

- Feature extraction.
- Architectures.
- Regularization methods.
- Neural network hyperparameters.
- Deployment.

Automated Machine Learning (AutoML) - What?



Google Cloud Why Google Solutions Products Pricing Getting started Docs Support Language Console

AI & Machine Learning Products Contact sales

AutoML products

Create your own custom machine learning models with an easy-to-use graphical interface.

Sight

AutoML Vision

Derive insights from images in the cloud or at the edge.

Learn more

AutoML Video Intelligence

Enable powerful content discovery and engaging video experiences.

Learn more

Language

AutoML Natural Language

Reveal the structure and meaning of text through machine learning.

Learn more

AutoML Translation

Dynamically detect and translate between languages.

Learn more

Structured data

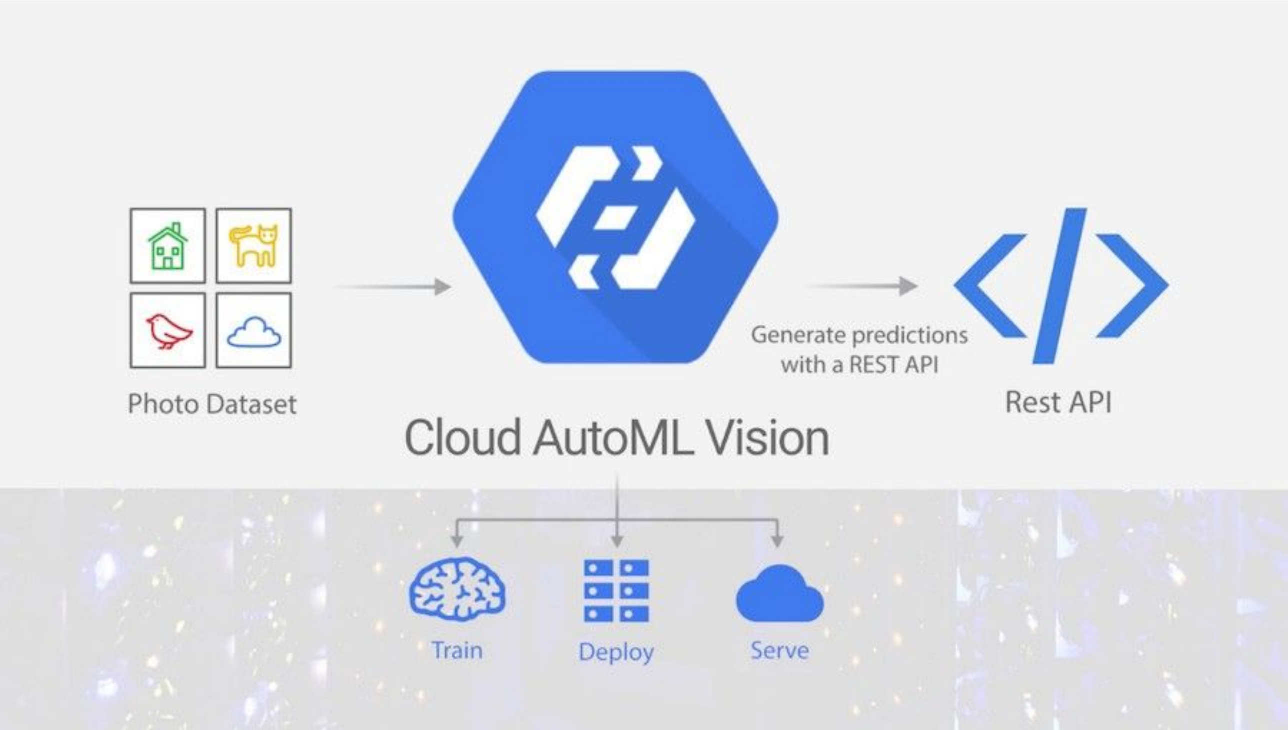
AutoML Tables

Automatically build and deploy state-of-the-art machine learning models on structured data.

Learn more

Source: https://cloud.google.com/automl/ .

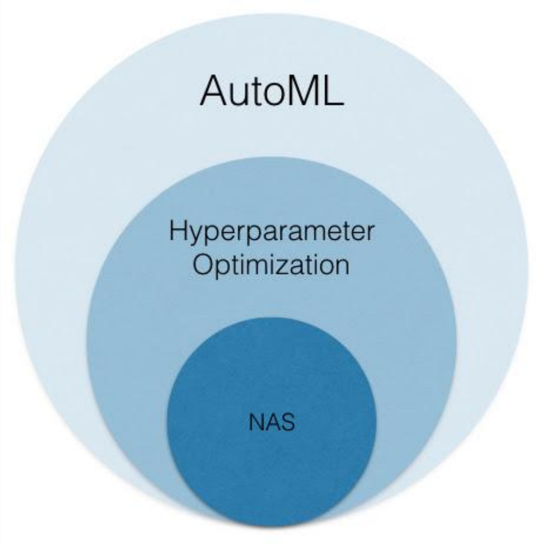
Automated Machine Learning (AutoML) - What?



Source: https://hackernoon.com/google-stop-trying-to-sell-us-automl-bc5eb53aca08 .

Neural Architecture Search (NAS) - What?

NAS - automatically learning neural architectures from the data.



Source: https://determined.ai/blog/neural-architecture-search/

Neural Architecture Search (NAS) - Why?

1. Our methods are primitive and limited by our biases

How we currently choose architectures (“experts”) :

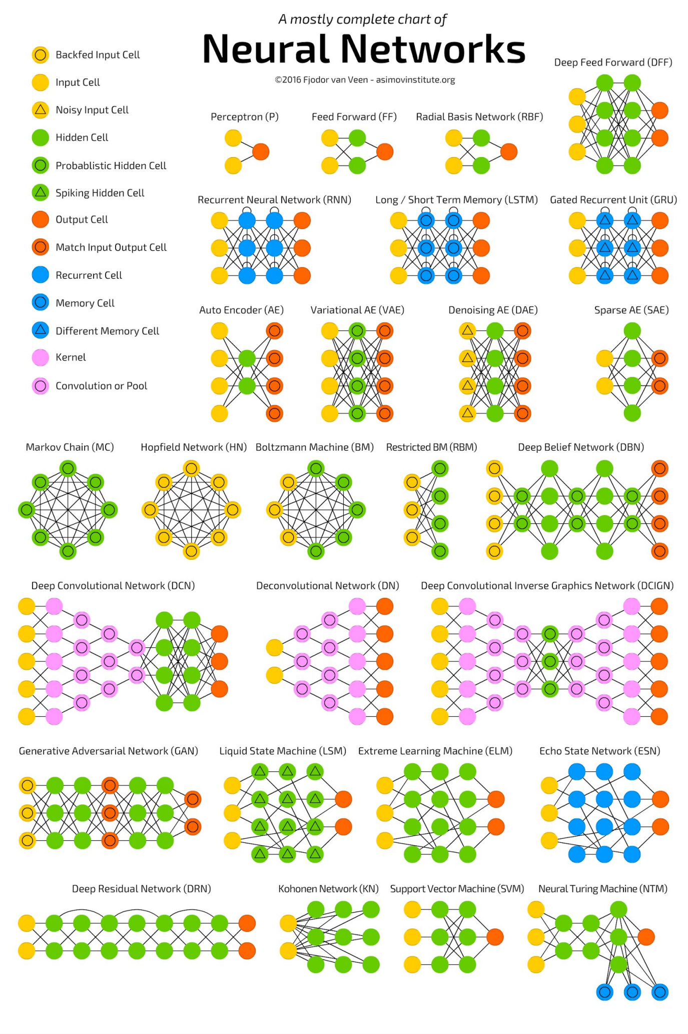
- Trial and Error on architectures we are most comfortable with.
- Copy architecture from a paper.
- Use methods such as grid search, random search or bayesian hyperparameter optimization that we find in a machine learning framework or library.

Decision are heavily influenced by our bias!

Neural Architecture Search (NAS) - Why?

1. Our methods are primitive and limited by our biases

Decision are heavily influenced by our knowledge!



Source: https://www.asimovinstitute.org/author/fjodorvanveen/

Neural Architecture Search (NAS) - Why?
2. Focusing on optimizing weights, as opposed to choosing the correct f.

Deep Learning Formulation (Goodfellow et al. [2016]):

$$h_\theta(x) = f(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{w}) = \phi(\boldsymbol{x}; \boldsymbol{\theta})^\top \boldsymbol{w}$$

Goal: Learn weights (θ and w).

Problem: Our goal is optimizing weights as opposed to choosing the correct function f.

*Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep learning. MIT press.

Neural Architecture Search (NAS) - Why?
3. Architectures are getting extremely complicated.

How would you come up with this?

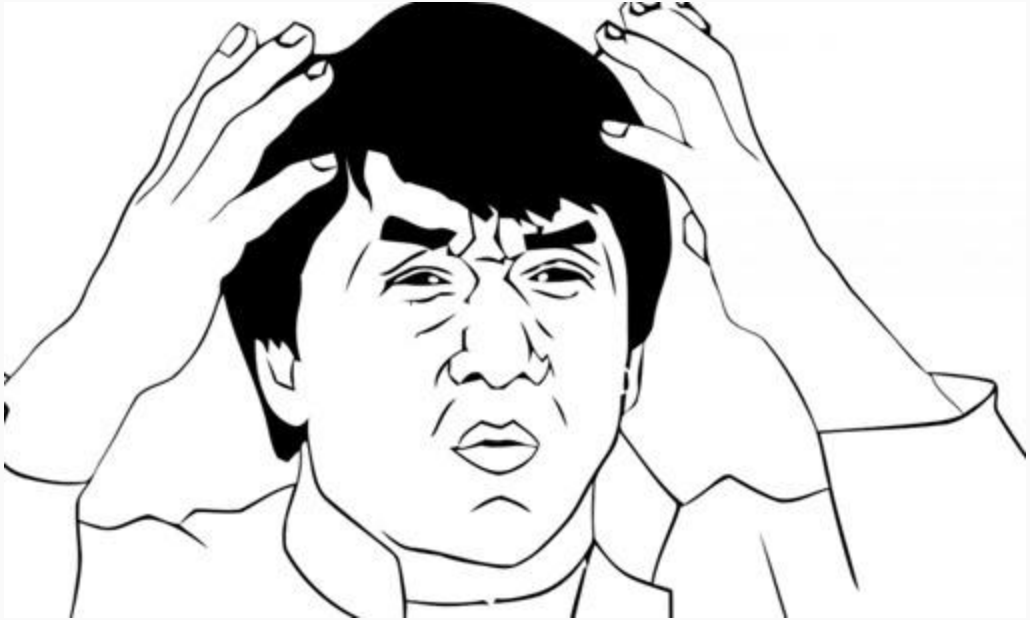
Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112×112	7×7 conv, stride 2			
Pooling	56×56	3×3 max pool, stride 2			
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv			
	28×28	2×2 average pool, stride 2			
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv			
	14×14	2×2 average pool, stride 2			
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14×14	1×1 conv			
	7×7	2×2 average pool, stride 2			
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1×1	7×7 global average pool			
		1000D fully-connected, softmax			

Table 1: DenseNet architectures for ImageNet. The growth rate for all the networks is $k = 32$. Note that each "conv" layer shown in the table corresponds the sequence BN-ReLU-Conv.

*Iandola, Forrest, et al. "Densenet: Implementing efficient convnet descriptor pyramids." *arXiv preprint arXiv:1404.1869* (2014).

Neural Architecture Search (NAS) - Why?

There must be a better way!

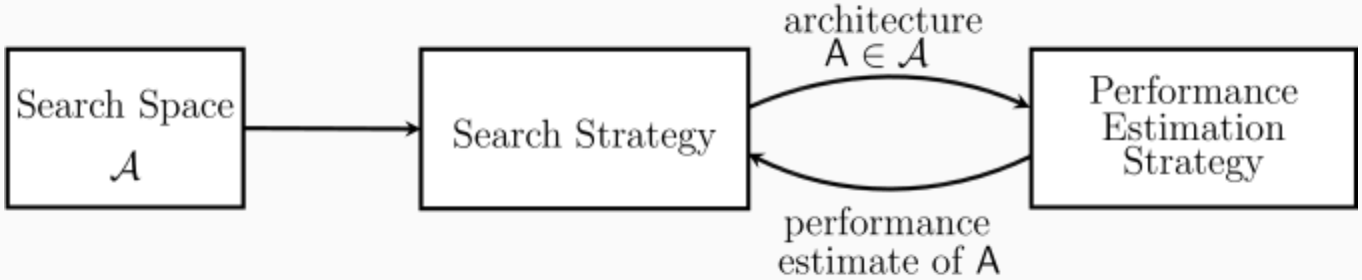


Neural Architecture Search (NAS) - Components

Components of NAS:

- **Search Space** - which possible architectures can be represented.
- **Search Strategy** - which search methods can be used such as random search or Bayesian hyperparameter optimization.
- **Performance Estimation** - techniques which estimate the performance of different architectures, preferably without having to fully train and evaluate each architecture.

Neural Architecture Search (NAS) - Components Interactions



Source: (Elsken et al. [2018b]):

*Elsken, T., Metzen, J. H., and Hutter, F. (2018b). Neural architecture search: A survey. arXiv preprint arXiv:1808.05377

Neural Architecture Search (NAS) - Search Space Formulation

Formulation:

Value of my current node.

Operations e.g. convolutions, pooling or concatenations.

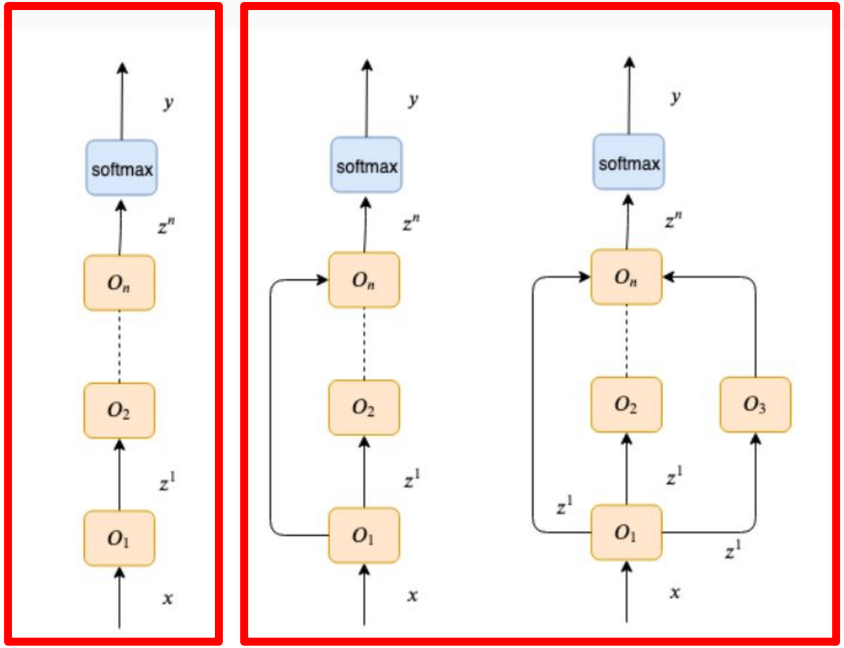
Value of my prev node.

$$z^k = o^k(I^k)$$

*Wistuba, M., Rawat, A., and Pedapati, T. (2019). A survey on neural architecture search. CoRR, abs/1905.01392.

Neural Architecture Search (NAS) - Search Space
Chain-structured neural networks

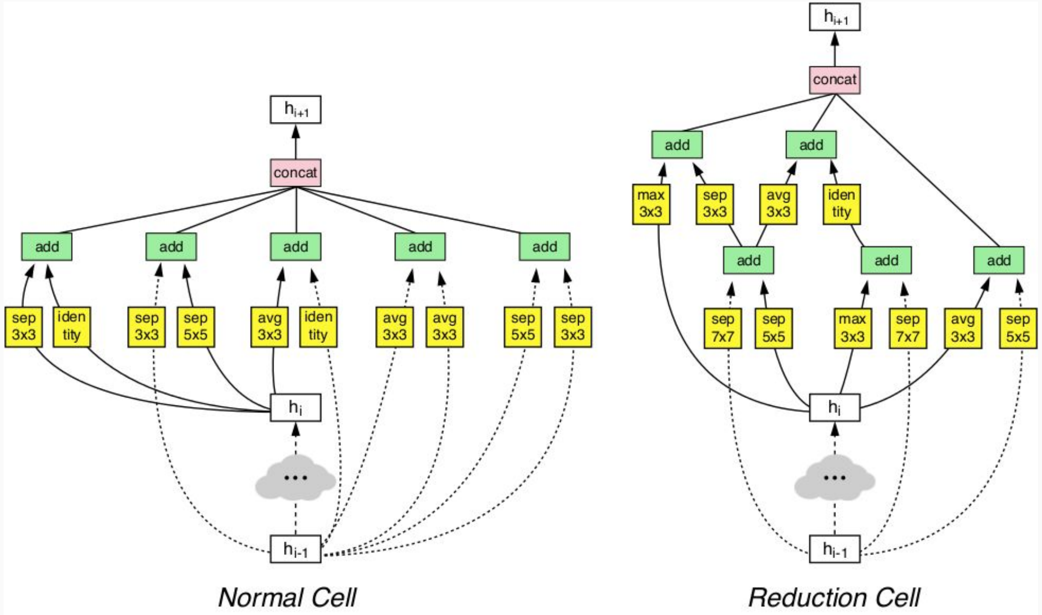
$$z^k = o^k(z^{k-1})$$



$$\boldsymbol{z}^{(k)} = o^{(k)}\left(\left\{\boldsymbol{z}^{(k-1)}\right\} \cup \left\{\boldsymbol{z}^{(i)} | \alpha_{i,k} = 1, i < k-1\right\}\right)$$

Neural Architecture Search (NAS) - Search Space
Cell-Based Architectures

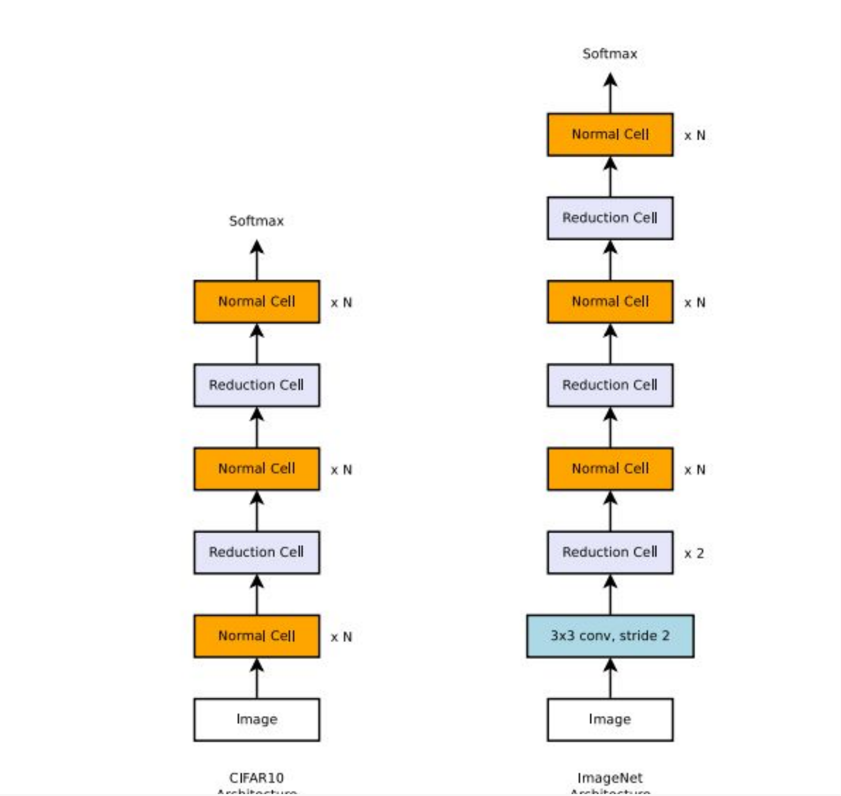
Zoph et al. [2018] noticed that many learned architecture were **composed of many repeating structures** and proposed rather learning **cells** (best convolutional layers), rather than learning whole architectures.



*Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 8697–8710.

Neural Architecture Search (NAS) - Search Space
Cell-Based Architectures

Once a cell had been learned on CIFAR-10, this cell was **transferred** to ImageNet (Zoph et al. [2018]).



*Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 8697–8710.

Neural Architecture Search (NAS) - Search Strategy

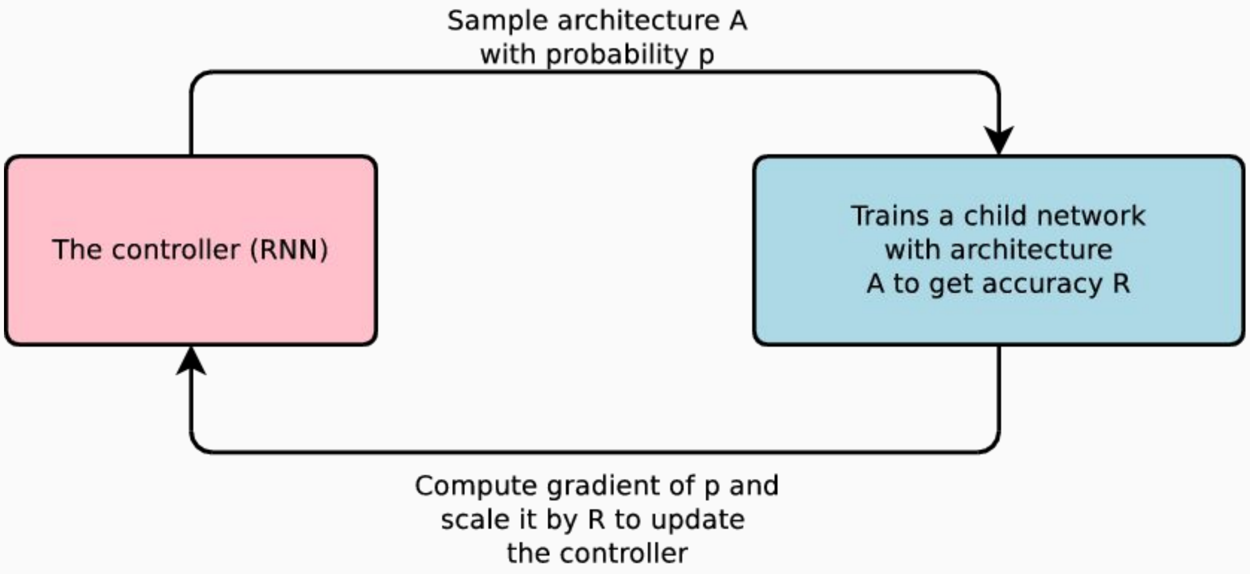
Optimization/Search Strategies used in NAS:

- **Reinforcement Learning** (Zoph and Le [2016], Baker et al. [2016])
- Random search (Sciuto et al. [2019], Li and Talwalkar [2019])
- **Bayesian optimization** (Mendoza et al. [2019])
- **Evolutionary methods** (Real et al. [2017], Elsken et al. [2017])
- Gradient-based methods (Liu et al. [2018b])

Neural Architecture Search (NAS) - Search Strategy
Reinforcement Learning

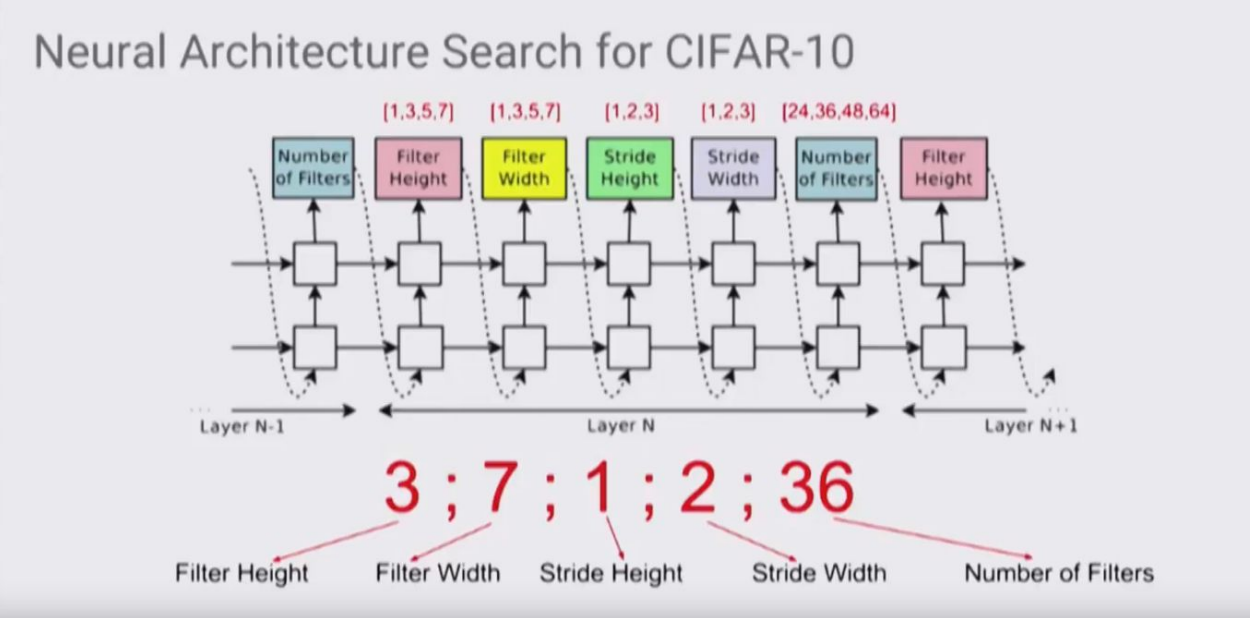
MDP:

- A (action space) refers to the search space in NAS,
- Taking an action - choosing an architecture.
- The reward signal R is the accuracy the architecture achieves on a holdout dataset and
- The policy is represented by a Recurrent Neural Network (RNN).



Zoph, B. and Le, Q. V. (2016). Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578.

Neural Architecture Search (NAS) - Search Strategy
Reinforcement Learning - Example



- ➢ State-of-the-art results for CIFAR-10 and Penn Treebank (Zoph and Le [2016]).
- ➢ Large computational demands
  - 800 GPUs for 3-4 weeks, 12 800 architectures evaluated.

Zoph, B. and Le, Q. V. (2016). Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578.

Neural Architecture Search (NAS) - Search Strategy
Evolutionary methods

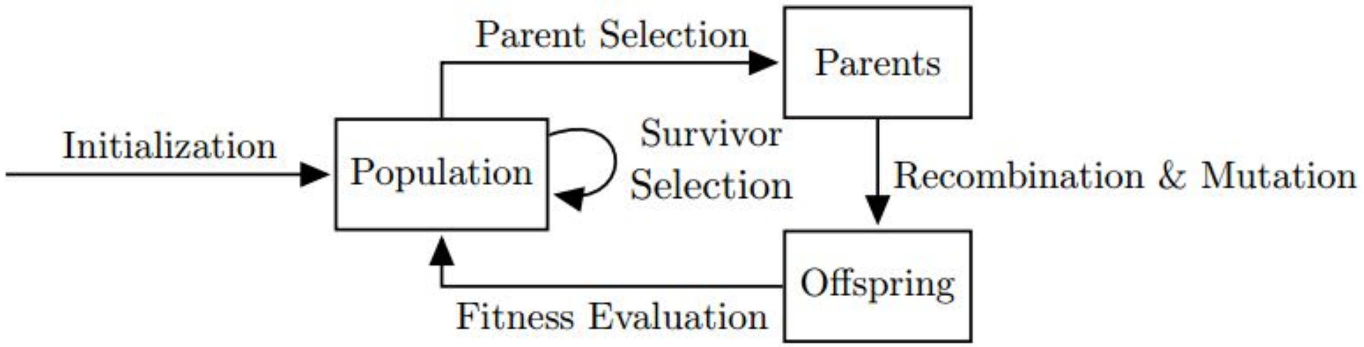


Figure 11: A general framework for evolutionary algorithms.

Source: Wistuba, M., Rawat, A., and Pedapati, T. (2019). A survey on neural architecture search. CoRR, abs/1905.01392.

Neural Architecture Search (NAS) - Search Strategy
Bayesian optimization

Gaussian Process based bayesian optimization:

- Kandasamy et al. [2018] derived a new kernel function κ for computing similarity between two network architectures.
- Assumption if κ (x; x0) is large, then f(x) and f (x0) are highly correlated.
- In this work, Kandasamy et al. [2018] managed to efficiently learn well-performing MLP and CNN architectures.

Kandasamy, K., Neiswanger, W., Schneider, J., Poczos, B., and Xing, E. P. (2018). Neural architecture search with bayesian optimisation and optimal transport. In Advances in Neural Information Processing Systems, pages 2016–2025.

Neural Architecture Search (NAS) - Performance Estimation

<u>Simple Method</u>
Train an architecture on the training data and fully evaluate it on test set.
Problem: Computationally expensive.

Neural Architecture Search (NAS) - Performance Estimation

Proxy Networks

Reduce computational cost is to use proxy networks which can be used to estimate the performance of the other architectures.

Examples:

- Trained for shorter training times (Zoph et al. [2018], Real et al. [2018])
- Trained on different types of input data, for example on less input data (Klein et al. [2016]) or lower quality images (Chrabaszcz et al. [2017]).
- A problem with these methods is that they highly rely on these proxy networks on being good estimates on performance on the desired architecture.

Neural Architecture Search (NAS) - Performance Estimation

Transfer Learning

Transfer learning, which is using information learned from previous task and applying it to a new, somewhat similar domain (West et al. [2007]).

Examples:

- CIFAR-10 and transferring it to ImageNet (Zoph et al. [2018]) .
- Use weights from previous architectures to initialize weights of a new, yet similar architecture (Wei et al. [2016]).

Neural Architecture Search (NAS) - Performance Estimation Other

Other

- Curve extrapolation - to terminate unpromising architectures early based on their performance (Klein et al. [2016],Baker et al. [2016]).
- Surrogate models - which predict the performance of architectures based on properties of their hyperparameters have also been proposed (Liu et al. [2018a]).

Weight sharing, one-shot methods..

Neural Architecture Search (NAS) - Summary of Results of Mentions Methods on CIFAR-10

	Reference	Error (%)	Params (Millions)	GPU Days
RL	Baker et al. (2017)	6.92	11.18	100
	Zoph and Le (2017)	3.65	37.4	22,400
	Cai et al. (2018a)	4.23	23.4	10
	Zoph et al. (2018)	3.41	3.3	2,000
	Zoph et al. (2018) + Cutout	2.65	3.3	2,000
	Zhong et al. (2018)	3.54	39.8	96
	Cai et al. (2018b)	2.99	5.7	200
	Cai et al. (2018b) + Cutout	2.49	5.7	200
EA	Real et al. (2017)	5.40	5.4	2,600
	Xie and Yuille (2017)	5.39	N/A	17
	Suganuma et al. (2017)	5.98	1.7	14.9
	Liu et al. (2018b)	3.75	15.7	300
	Real et al. (2019)	3.34	3.2	3,150
	Elsken et al. (2018)	5.2	19.7	1
	Wistuba (2018a) + Cutout	3.57	5.8	0.5
SMBO	Kandasamy et al. (2018)	8.69	N/A	1.7
	Liu et al. (2018a)	3.41	3.2	225
	Luo et al. (2018)	3.18	10.6	200
One-Shot	Pham et al. (2018)	3.54	4.6	0.5
	Pham et al. (2018) + Cutout	2.89	4.6	0.5
	Bender et al. (2018)	4.00	5.0	N/A
	Casale et al. (2019) + Cutout	2.81	3.7	1
	Liu et al. (2019b) + Cutout	2.76	3.3	4
	Xie et al. (2019b) + Cutout	2.85	2.8	1.5
	Cai et al. (2019) + Cutout	2.08	5.7	8.33
	Brock et al. (2018)	4.03	16.0	3
	Zhang et al. (2019)	2.84	5.7	0.84
Random	Liu et al. (2018b)	3.91	N/A	300
	Luo et al. (2018)	3.92	3.9	0.3
	Liu et al. (2019b) + Cutout	3.29	3.2	4
	Li and Talwalkar (2019) + Cutout	2.85	4.3	2.7
Human	Zagoruyko and Komodakis (2016)	3.87	36.2	-
	Gastaldi (2017) (26 2x32d)	3.55	2.9	-
	Gastaldi (2017) (26 2x96d)	2.86	26.2	-
	Gastaldi (2017) (26 2x112d)	2.82	35.6	-
	Yamada et al. (2016) + ShakeDrop	2.67	26.2	-

Source:
Wistuba, M., Rawat, A., and Pedapati, T. (2019). A survey on neural architecture search. CoRR, abs/1905.01392.

Neural Architecture Search (NAS) - Problems with previous methods

Issues with the current approaches:

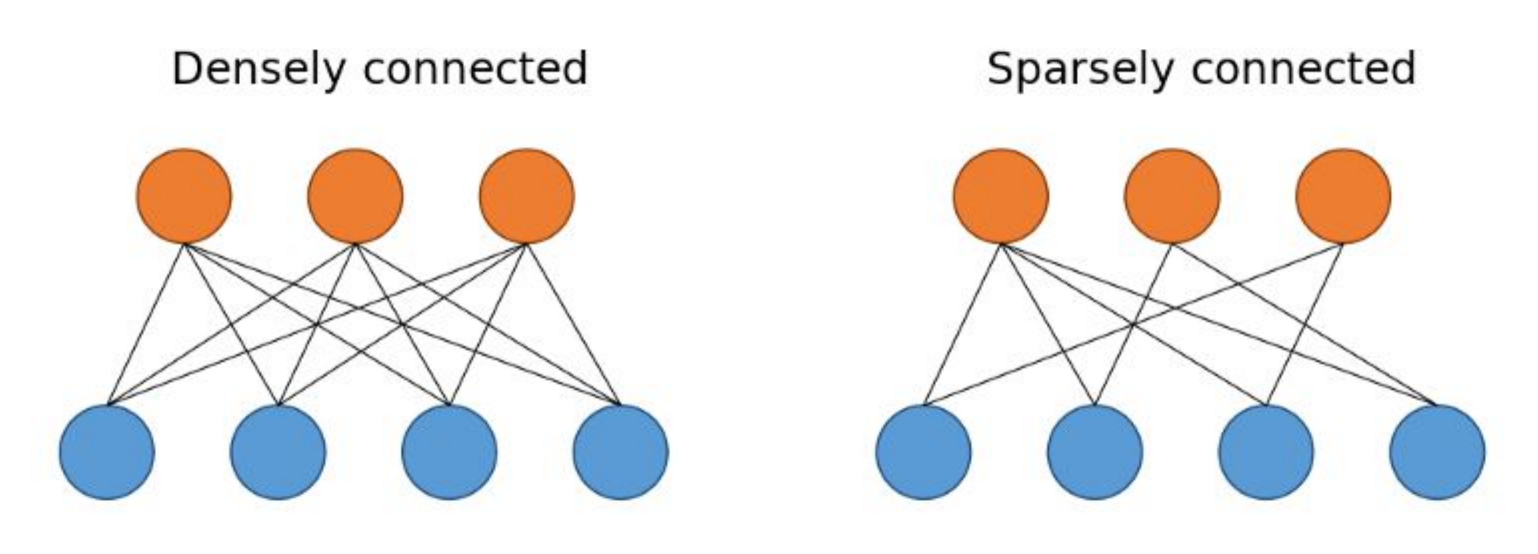
1. These methods are **computationally intensive,** with some of these methods taking thousands of GPU days (Zoph and Le [2016], Zoph et al. [2018], Real et al. [2017, 2019]).
2. A lot of these methods still require **domain engineering** (Zoph and Le [2016], Zoph et al. [2018]).
3. The search space of architectures used are usually restricted to using **convolutional layers**, which are a standard in computer vision and machine translation problems. However, these aren't easily transferable to other domains (Zoph et al. [2018], Real et al. [2019], Baker et al. [2016], Suganuma et al. [2017]).

Learning Compact, General Purpose Neural Network Architectures

Compact - smaller architectures. We will allow specification of a maximum depth and width of a neuron network and learn an efficient architecture within those bounds.

General Purpose - Only use standard component of a Dense layer in a Neural Network for the architecture (number of weight, number of nodes and number of hidden layers) and no domain specific parts of Neural Networks such as convolutions.

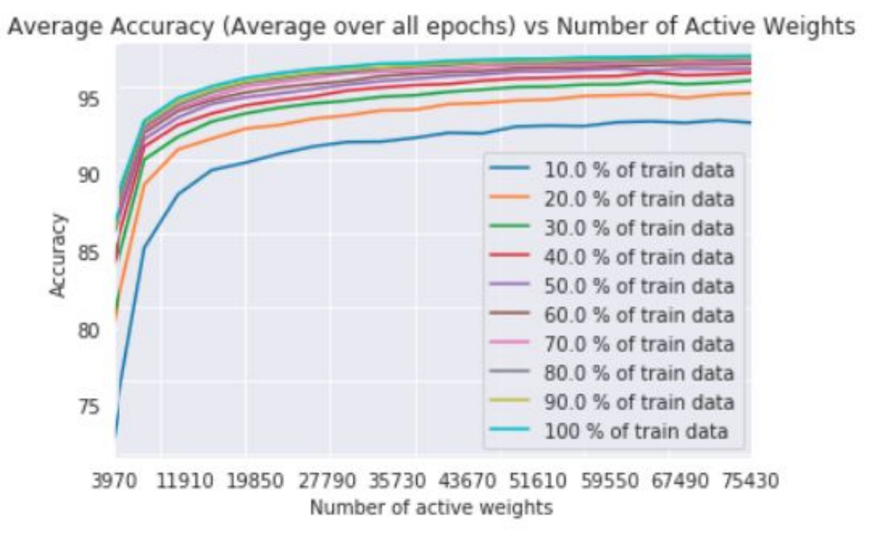
Learning Compact, General Purpose Neural Network Architectures
Sparse vs Dense Layers



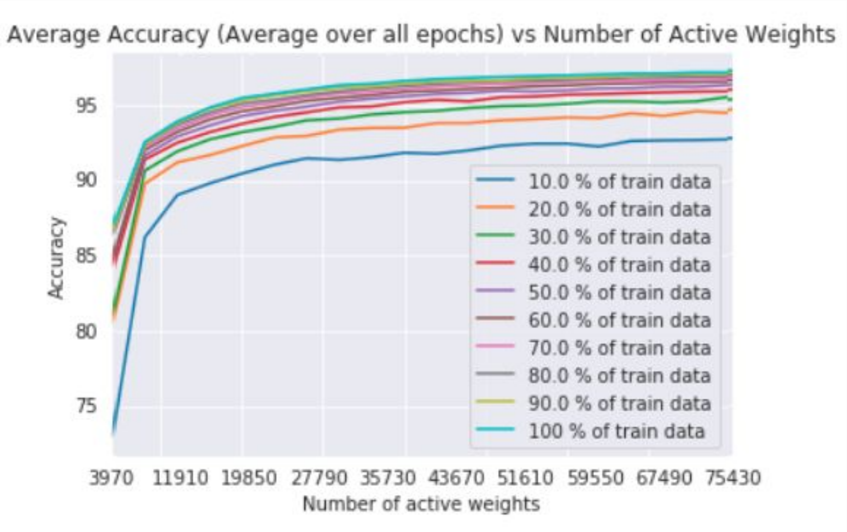
Source: https://amiralavi.net/blog/2018/07/29/vnn-implementation

Learning Compact, General Purpose Neural Network Architectures
Sparse vs Dense Layers - Experimental Results

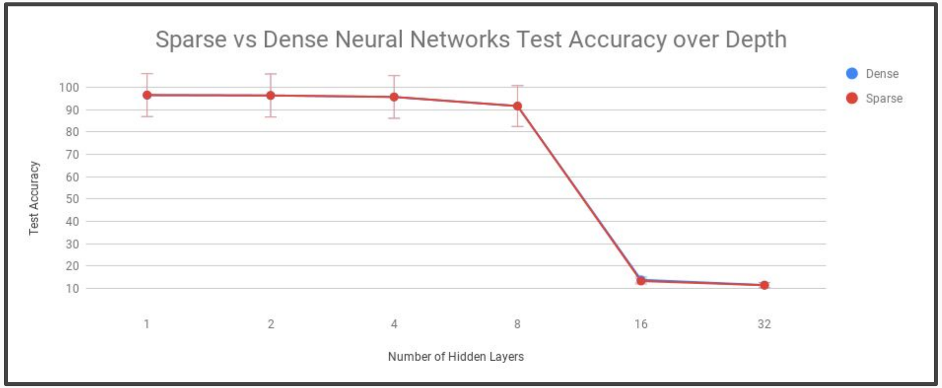
Sparse

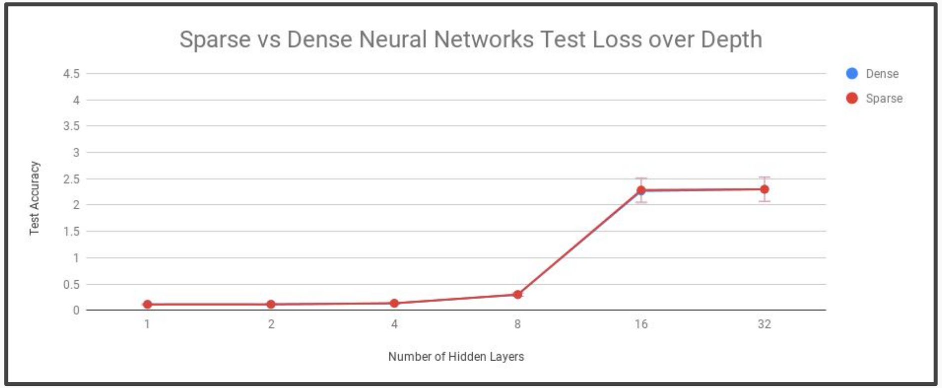


Dense



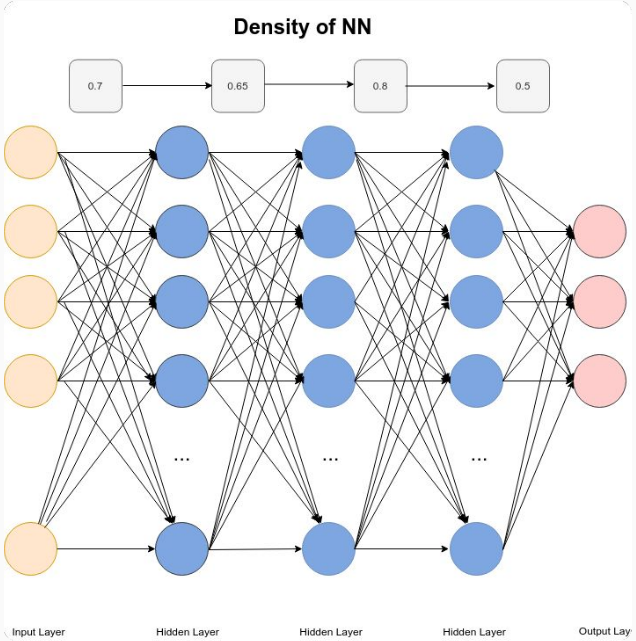
Learning Compact, General Purpose Neural Network Architectures
Shallow vs Deep Networks - Sparse Experimental Results



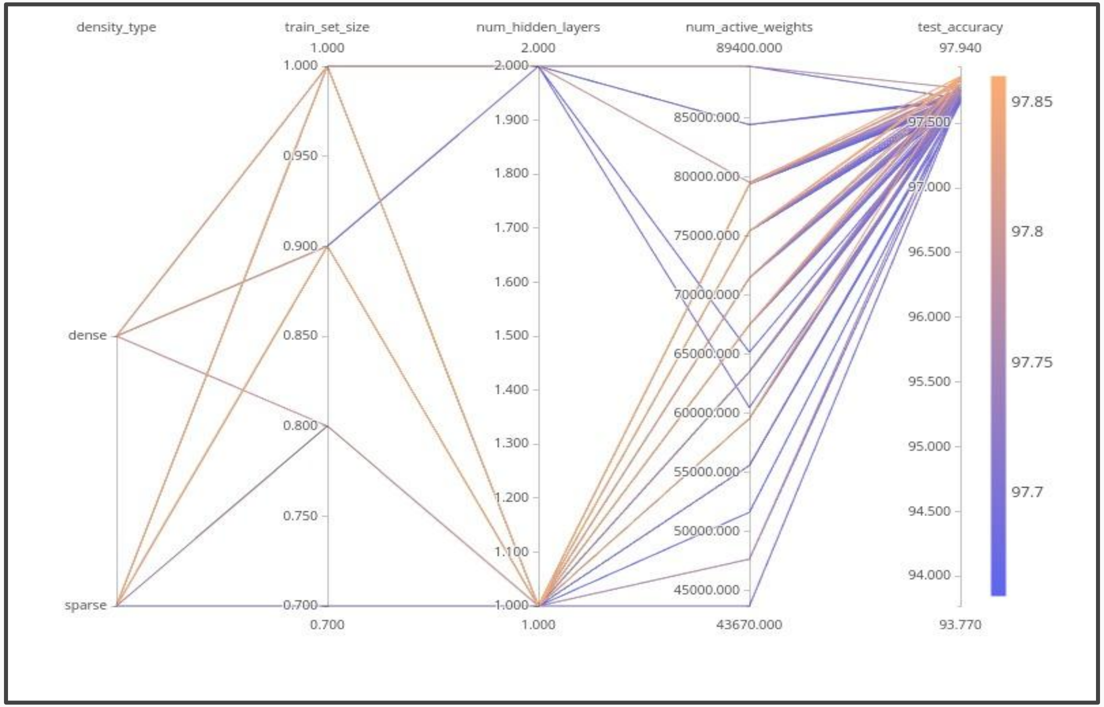


Learning Compact, General Purpose Neural Network Architectures
Density

Define a concept called **Density** - percentage of active weights in a layer.



Learning Compact, General Purpose Neural Network Architectures
Grid Search - Best Performing Architectures - Mnist



Conclusions

- AutoML and NAS will allow humans to become more productive -don’t waste time on trial and error on hyperparameters.
- Make ML accessible to non-experts.
- Interesting field - beyond tuning hyperparameters - a lot overlap with other research fields - RL, Deep Learning, Hyperparameter optimization.

Materials for NAS

Great Survey Papers:

- Neural Architecture Search: A Survey - Thomas Elsken, Jan Hendrik Metzen, Frank Hutter.
- A Survey on Neural Architecture Search - Martin Wistuba, Ambrish Rawat, Tejaswini Pedapati.

AutoML website - https://www.automl.org/

References

- Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 8697–8710.
- Real, E., Aggarwal, A., Huang, Y., and Le, Q. V. (2018). Regularized evolution for image classifier architecture search. arXiv preprint arXiv:1802.01548.
- Chrabaszcz, P., Loshchilov, I., and Hutter, F. (2017). A downsampled variant of imagenet as an alternative to the cifar datasets. arXiv preprint arXiv:1707.08819.
- Mendoza, H., Klein, A., Feurer, M., Springenberg, J. T., Urban, M., Burkart, M., Dippel, M.,Lindauer, M., and Hutter, F. (2019). Towards automatically-tuned deep neural networks. In Automated Machine Learning, pages 135–149. Springer.
- Wei, T., Wang, C., Rui, Y., and Chen, C. W. (2016). Network morphism. In International Conference on Machine Learning, pages 564–572.
- Baker, B., Gupta, O., Naik, N., and Raskar, R. (2016). Designing neural network architectures using reinforcement learning. CoRR, abs/1611.02167.
- Suganuma, M., Shirakawa, S., and Nagao, T. (2017). A genetic programming approach to designing convolutional neural network architectures. In Proceedings of the Genetic and Evolutionary Computation Conference, pages 497–504. ACM.

Questions

???