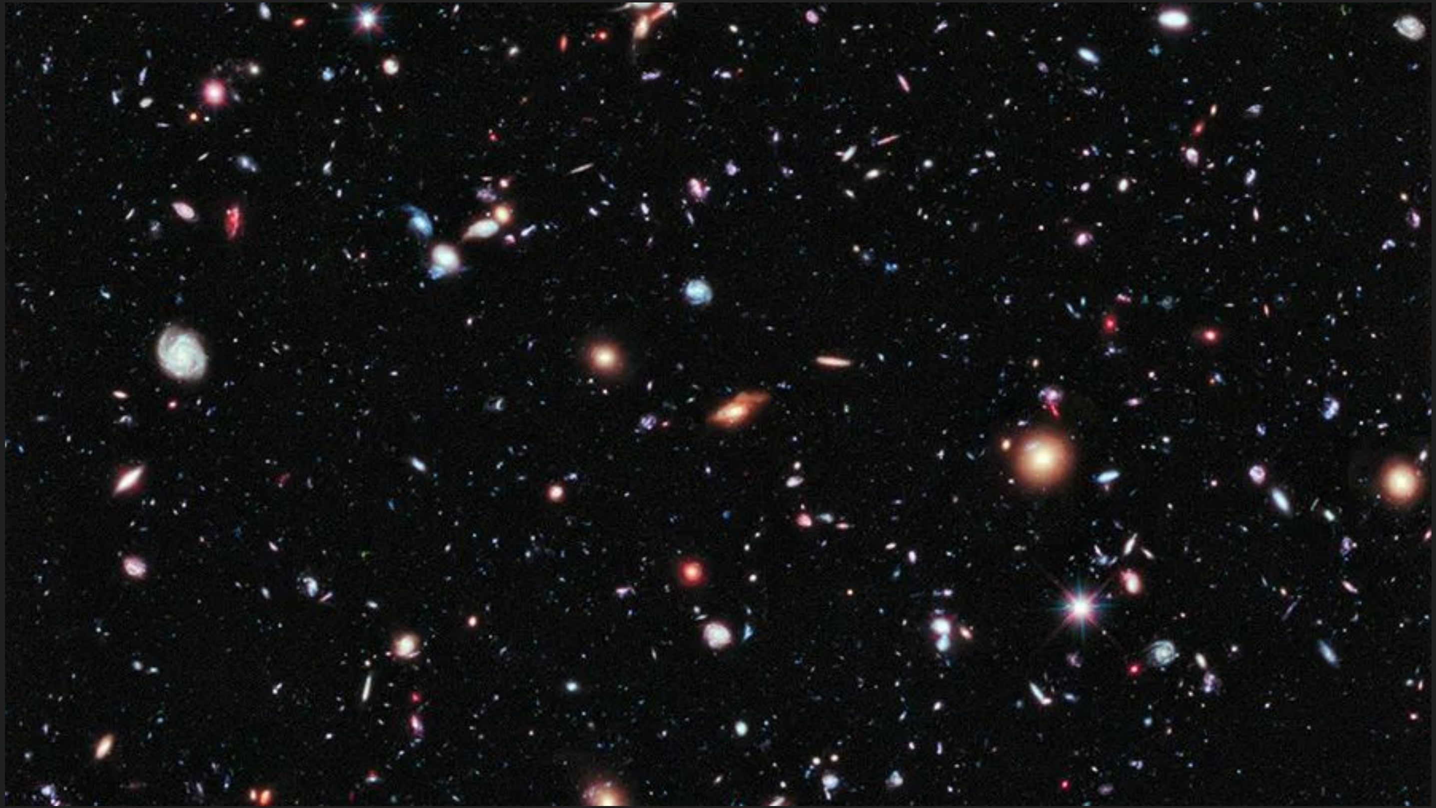
AutoML

Prepared for:
Winter 2022 ENEL 645
Data Mining & Machine Learning

Prepared by:
Mike Lasby
Dr. Roberto Souza

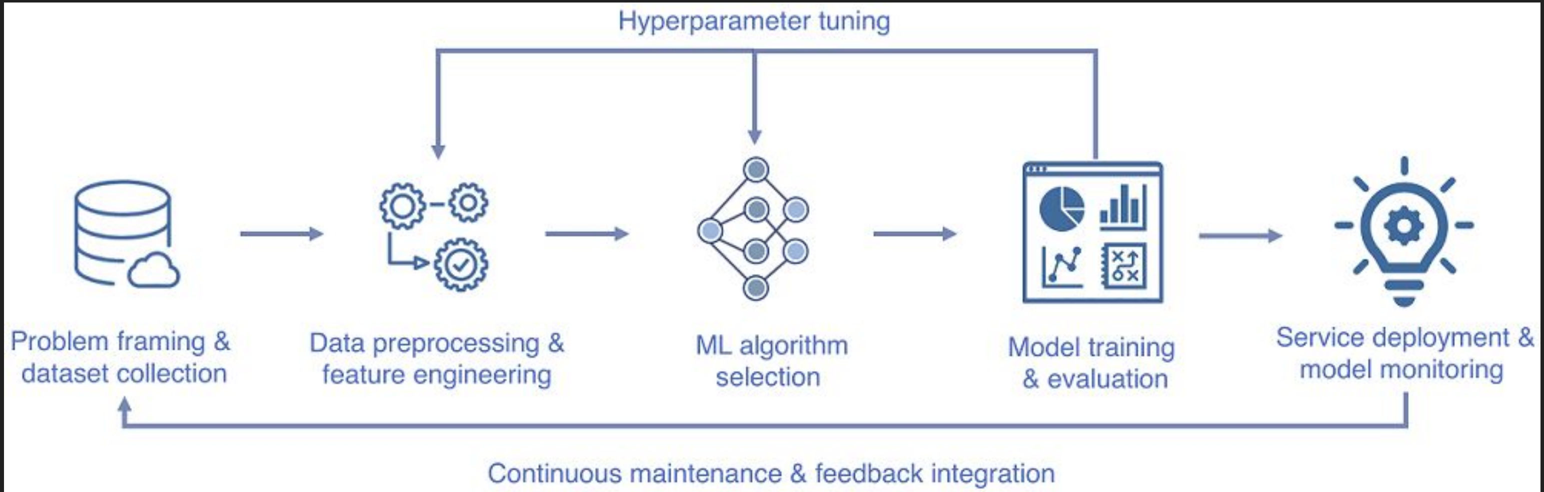
What is Automated Machine Learning (AutoML)?

UNIVERSITY OF CALGARY



What is Automated Machine Learning (AutoML)?

- Automation of the machine learning model development process
- End-to-end optimization
- Grid search on steroids
- Mimic plasticity in human brain

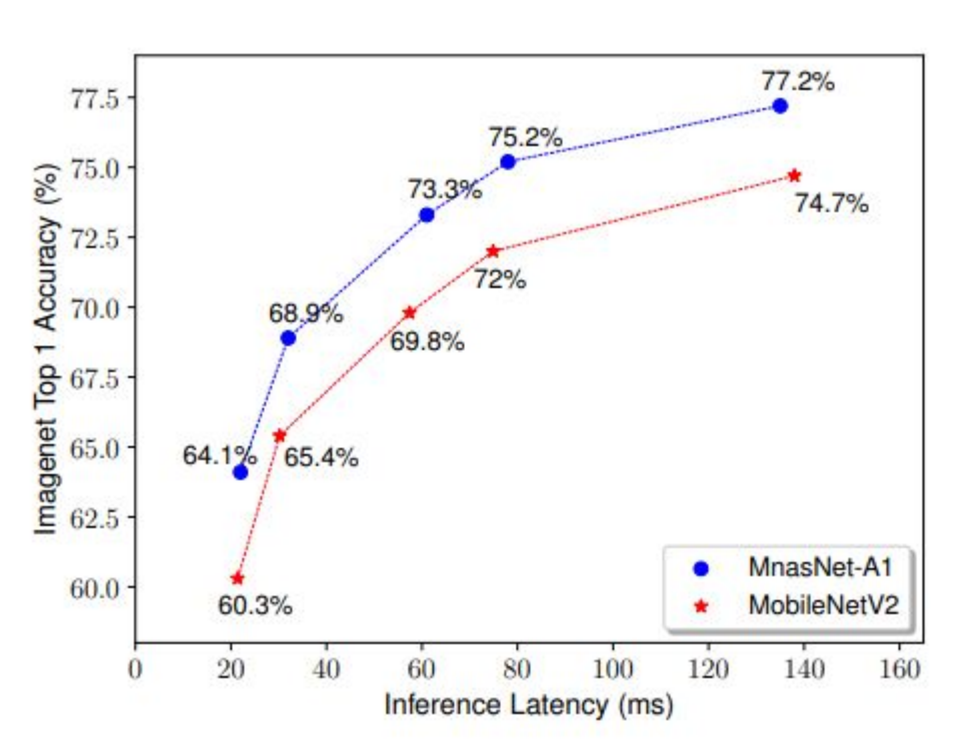


Song, Q., Jin, H., and Hu, X., *Automated Machine Learning in Action*, MEAP Version 7, Manning Publications, 2021.

Why AutoML?

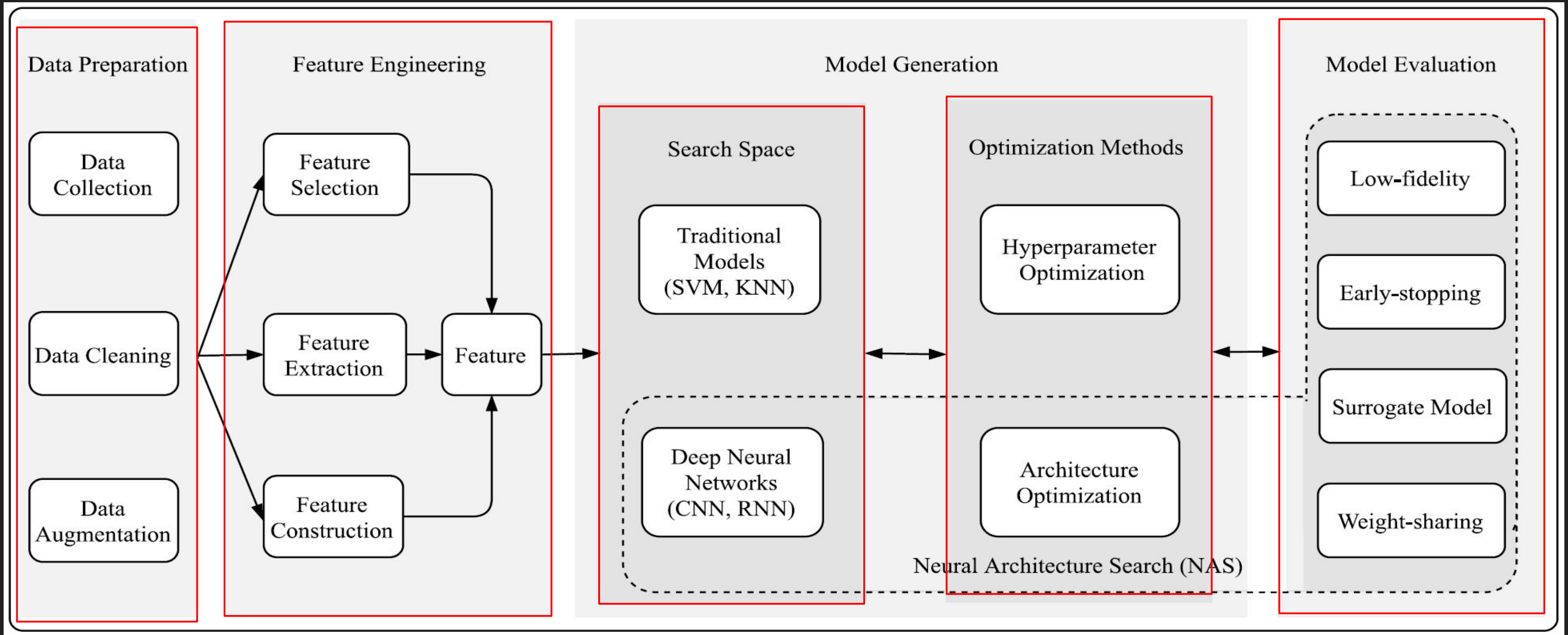
UNIVERSITY OF CALGARY

- Democratization of machine learning
- Put ML in the hands of domain experts
- Facilitate development of low / no code solutions
- Develop novel machine learning algorithms & architectures
- Achieve SOTA on existing benchmark



M. Tan *et al.*, “MnasNet: Platform-Aware Neural Architecture Search for Mobile,” *arXiv:1807.11626 [cs]*, May 2019, Accessed: Nov. 23, 2021. [Online]. Available: http://arxiv.org/abs/1807.11626

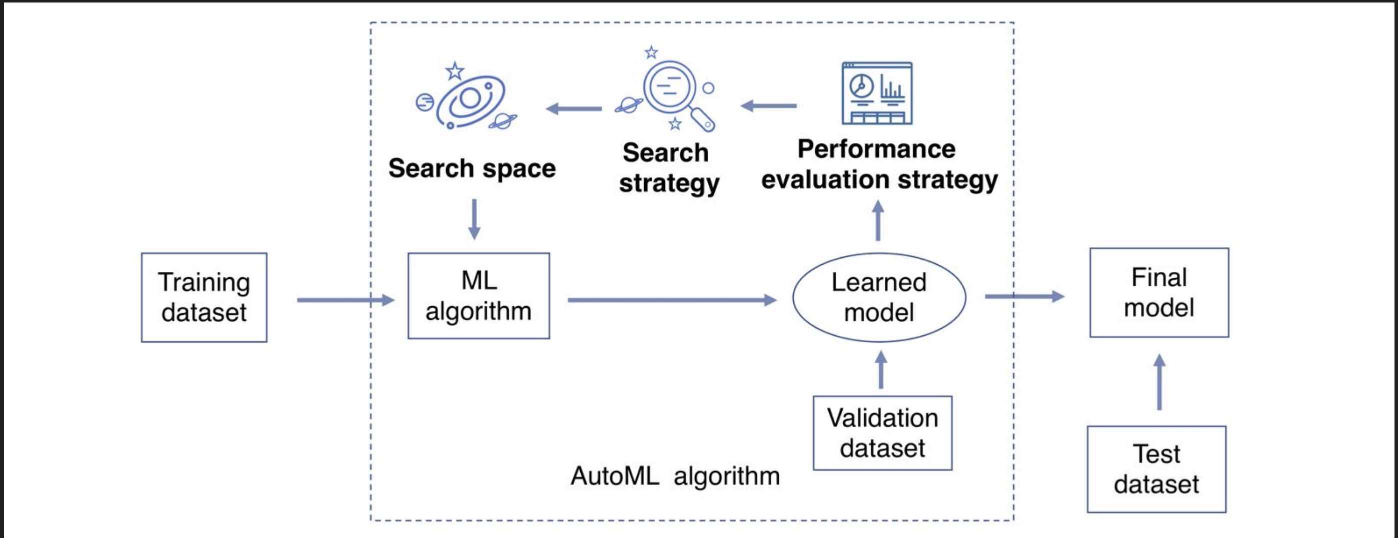
AutoML Components



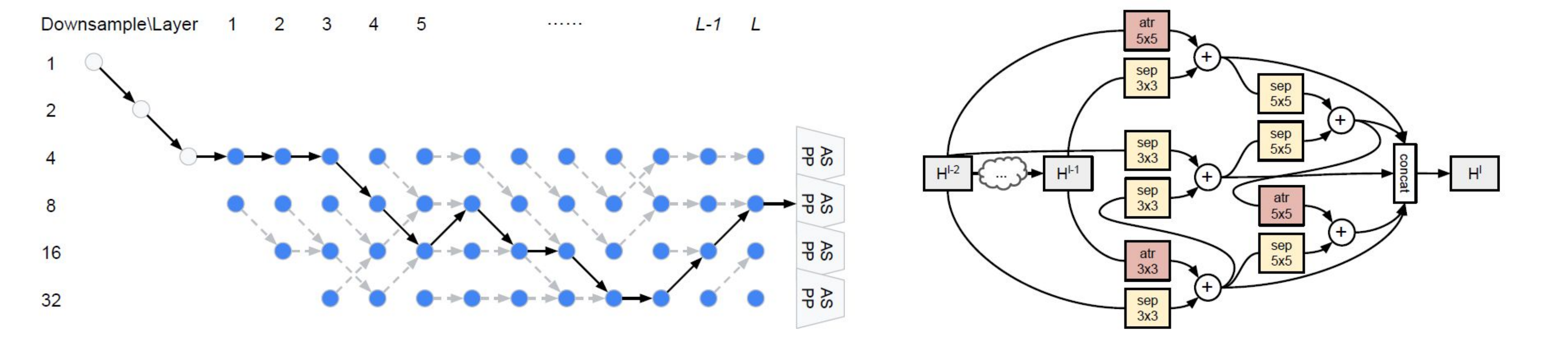
X. He, K. Zhao, X. Chu., “AutoML: A survey of the state-of-the-art,” Knowledge-Based Systems, vol. 212, 2021.

Neural Architecture Search (NAS)

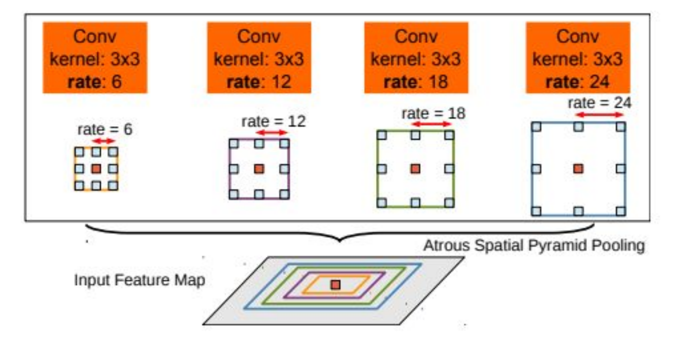
- Neural architecture search (NAS) is the primary challenge in deep learning based AutoML



NAS Search Space



C. Liu et al., “Auto-DeepLab: Hierarchical Neural Architecture Search for Semantic Image Segmentation,” in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, Jun. 2019, pp. 82–92. doi: 10.1109/CVPR.2019.00017.

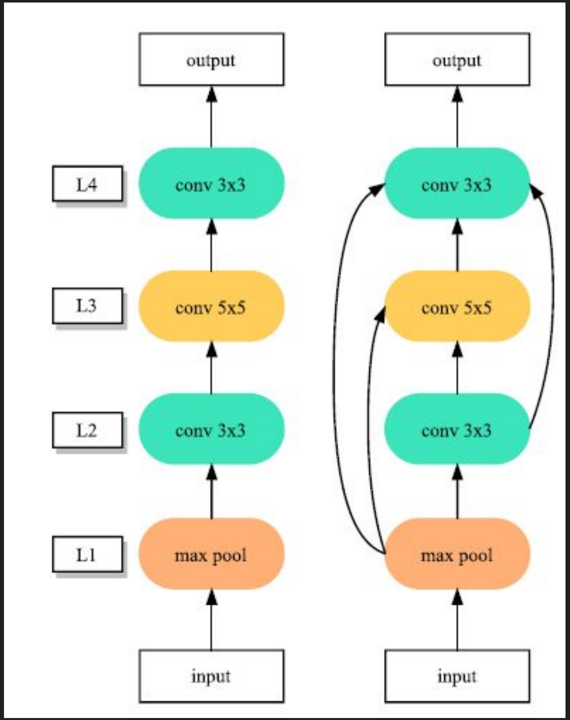


L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs,” arXiv:1606.00915 [cs], May 2017, Accessed: Nov. 25, 2021. [Online]. Available: http://arxiv.org/abs/1606.00915

NAS Search Space - Entire-structured

UNIVERSITY OF CALGARY

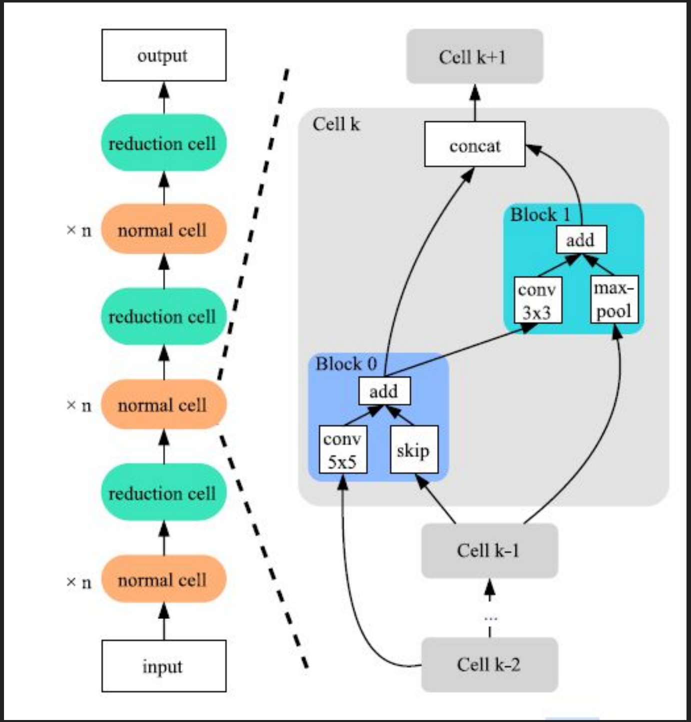
- Define search space:
    - **Entire-structured - Inefficient, least generalizable**
    - Cell-based - Fixed number of nodes, suitable for scaling, no network-level search
    - Hierarchical - Arbitrary number of cell nodes, arbitrary number of cell types, network level search
    - Morphism-based - Performs depth, width, and kernel size-morphing, efficient



X. He, K. Zhao, X. Chu., "AutoML: A survey of the state-of-the-art," Knowledge-Based Systems, vol. 212, 2021.

NAS Search Space - Cell-based

- Define search space:
    - Entire-structured - Inefficient, least generalizable
    - **Cell-based - Fixed number of nodes, suitable for scaling, no network-level search**
    - Hierarchical - Arbitrary number of cell nodes, arbitrary number of cell types, network level search
    - Morphism-based - Performs depth, width, and kernel size-morphing, efficient

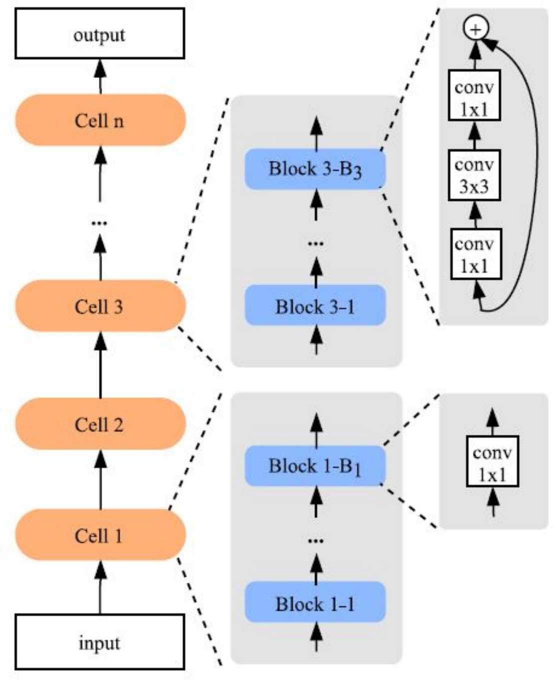


X. He, K. Zhao, X. Chu., "AutoML: A survey of the state-of-the-art," Knowledge-Based Systems, vol. 212, 2021.

NAS Search Space - Hierarchical

UNIVERSITY OF CALGARY

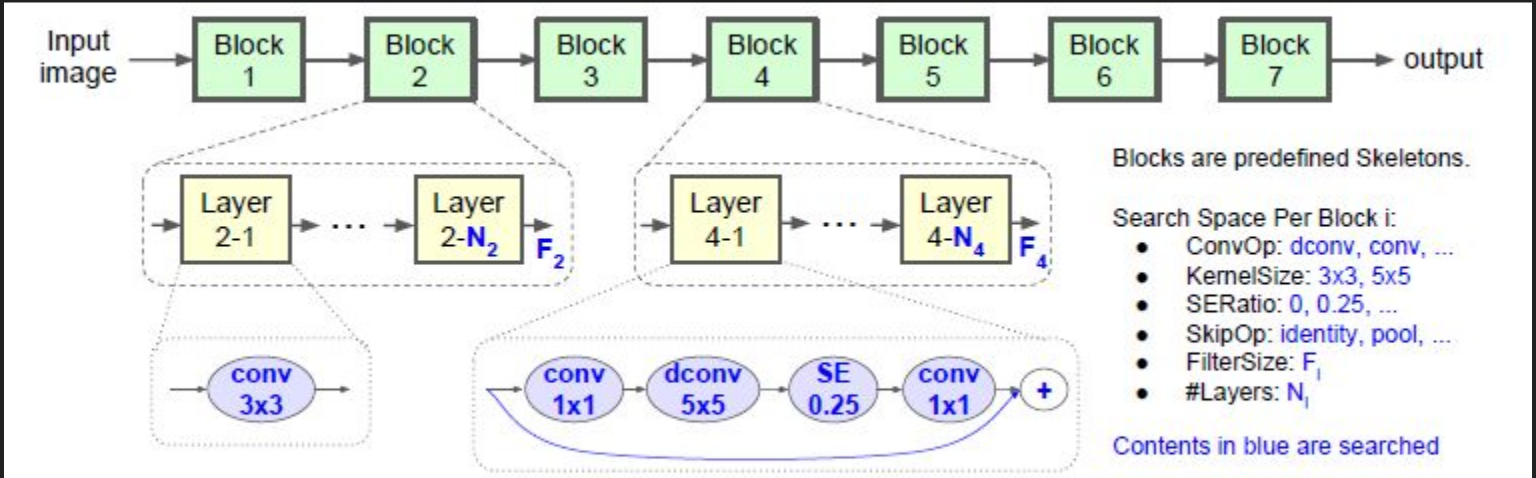
- Define search space:
  - Entire-structured - Inefficient, least generalizable
  - Cell-based - Fixed number of nodes, suitable for scaling, no network-level search
  - **Hierarchical - Arbitrary number of cell nodes, arbitrary number of cell types, network level search**
  - Morphism-based - Performs depth, width, and kernel size-morphing, efficient



X. He, K. Zhao, X. Chu., "AutoML: A survey of the state-of-the-art," Knowledge-Based Systems, vol. 212, 2021.

NAS Search Space - Hierarchical

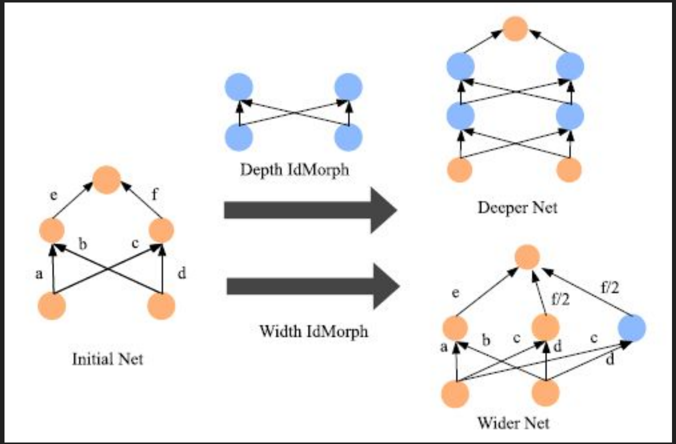
- Hierarchical Search Space For MnasNet:



M. Tan et al., “MnasNet: Platform-Aware Neural Architecture Search for Mobile,” arXiv:1807.11626 [cs], May 2019, Accessed: Nov. 23, 2021. [Online]. Available: http://arxiv.org/abs/1807.11626

NAS Search Space

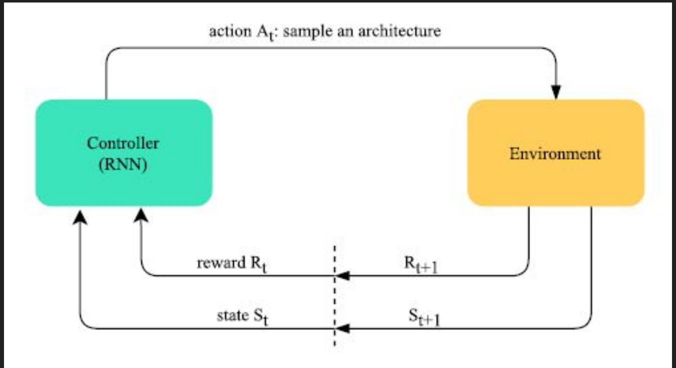
- Define search space:
  - Entire-structured - Inefficient, least generalizable
  - Cell-based - Fixed number of nodes, suitable for scaling, no network-level search
  - Hierarchical - Arbitrary number of cell nodes, arbitrary number of cell types, network level search
  - **Morphism-based - Performs depth, width, and kernel size-morphing, efficient**



X. He, K. Zhao, X. Chu., "AutoML: A survey of the state-of-the-art," Knowledge-Based Systems, vol. 212, 2021.

Architecture Optimization - RL

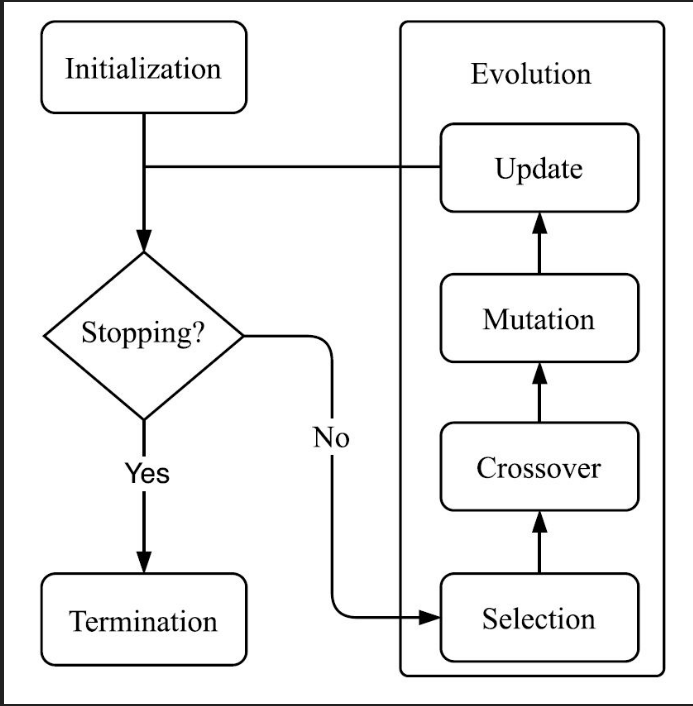
- Methods
  - Reinforcement Learning
    - Controller samples an architecture from the search space
    - Controller updates sampling strategy based on reward returned
    - Very computationally expensive
    - Can share weights to improve cost



X. He, K. Zhao, X. Chu., "AutoML: A survey of the state-of-the-art," Knowledge-Based Systems, vol. 212, 2021.

Architecture Optimization - EA

- Methods
  - Evolutionary Algorithms
    - Select networks and evaluate on train / val sets
    - Best networks are paired up to create next generation which includes a mutation
    - Very computationally expensive

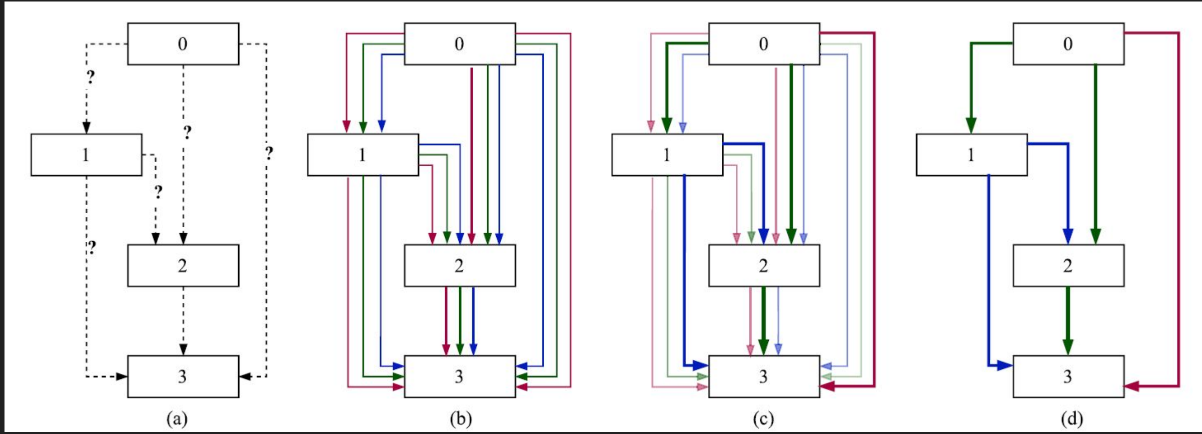


X. He, K. Zhao, X. Chu., “AutoML: A survey of the state-of-the-art,” Knowledge-Based Systems, vol. 212, 2021.

Architecture Optimization - DAS

UNIVERSITY OF CALGARY

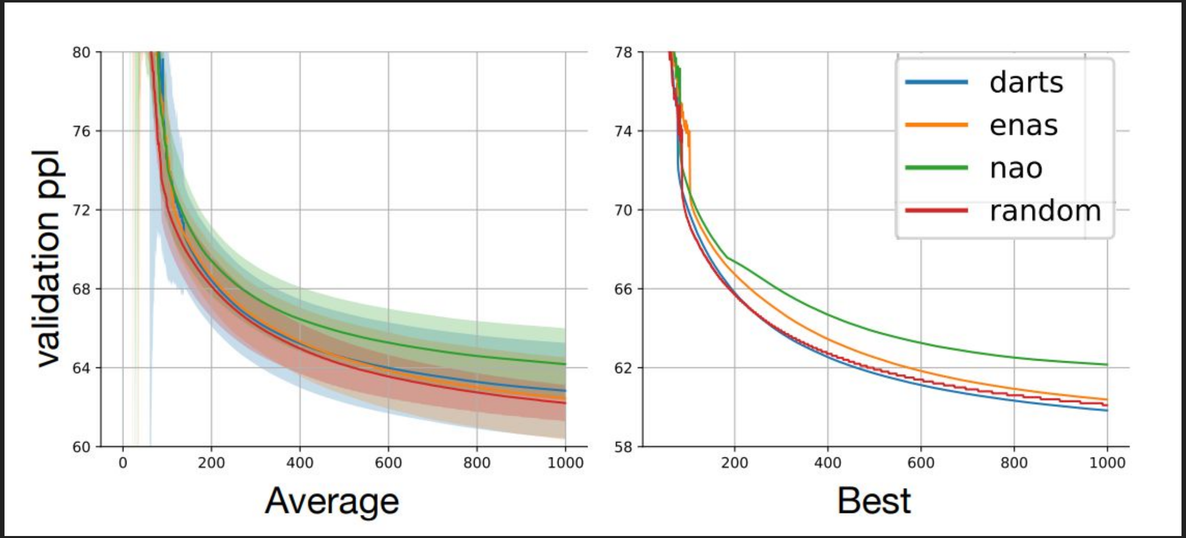
- Methods
  - Differentiable Architecture Search
    - Uses gradient descent to search architectures
    - Creates a “supernet” of all possible child networks
    - Introduces several problems
    - Typically not possible to use on large datasets due to memory overhead



X. He, K. Zhao, X. Chu., “AutoML: A survey of the state-of-the-art,” Knowledge-Based Systems, vol. 212, 2021.

Architecture Optimization - Random search

- Methods
  - Random search
    - A competitive benchmark for NAS
    - Performance heavily depends on search space size



K. Yu, C. Sciuto, M. Jaggi, C. Musat, and M. Salzmann, “Evaluating the Search Phase of Neural Architecture Search,” arXiv:1902.08142

Architecture Evaluation

- Methods
    - Low Fidelity
        - Decrease size of data
        - Decrease image resolution
        - Decrease number of images
        - Eg., ImageNet 64x64
        - Reduce model size
        - Model performance strongly correlated between short and long training durations

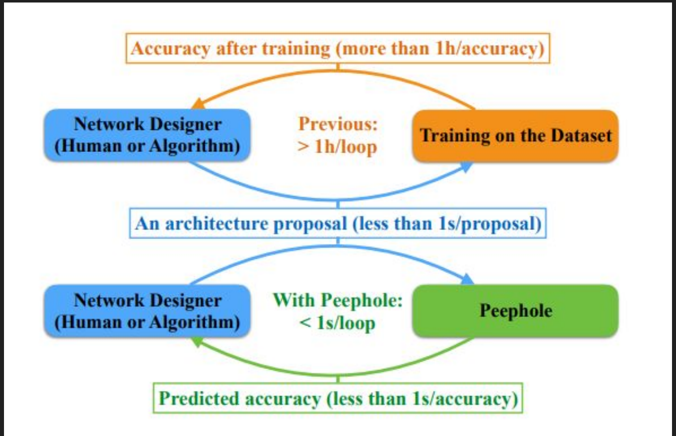
Spearman rank correlation coefficients for various training periods

	1200s	1h	3h
400s	0.87	0.31	0.05
1200s		0.88	0.64
1h			0.86

A. Zela, A. Klein, S. Falkner, and F. Hutter, “Towards Automated Deep Learning: Efficient Joint Neural Architecture and Hyperparameter Search,” arXiv:1807.06906 [cs, stat], Jul. 2018, Available: http://arxiv.org/abs/1807.06906

Architecture Evaluation - Other methods

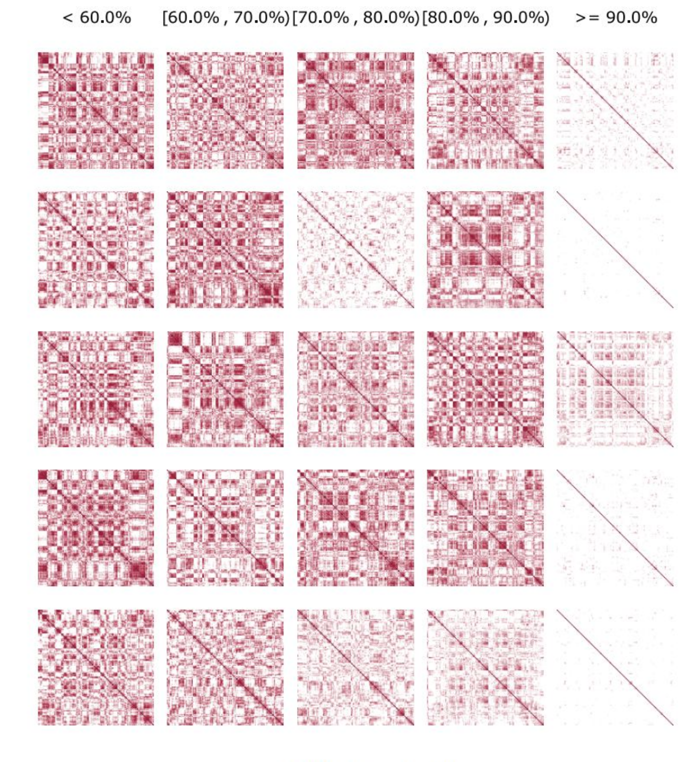
- Methods
  - Weight sharing
    - Share weights among child networks
    - 1000x faster network evaluation
  - Surrogate
    - Find a model that predicts performance of neural networks
    - Possible by using NAS benchmarks, need large amounts of labelled architectures
    - Holy grail?
  - Early stopping
    - Stop evaluations of models that appear to be training poorly based on learning curve



B. Deng, J. Yan, and D. Lin, “Peephole: Predicting Network Performance Before Training,” arXiv:1712.03351 [cs, stat], Dec. 2017, Accessed: Nov. 25, 2021. [Online]. Available: http://arxiv.org/abs/1712.03351

Architecture Evaluation - Training free

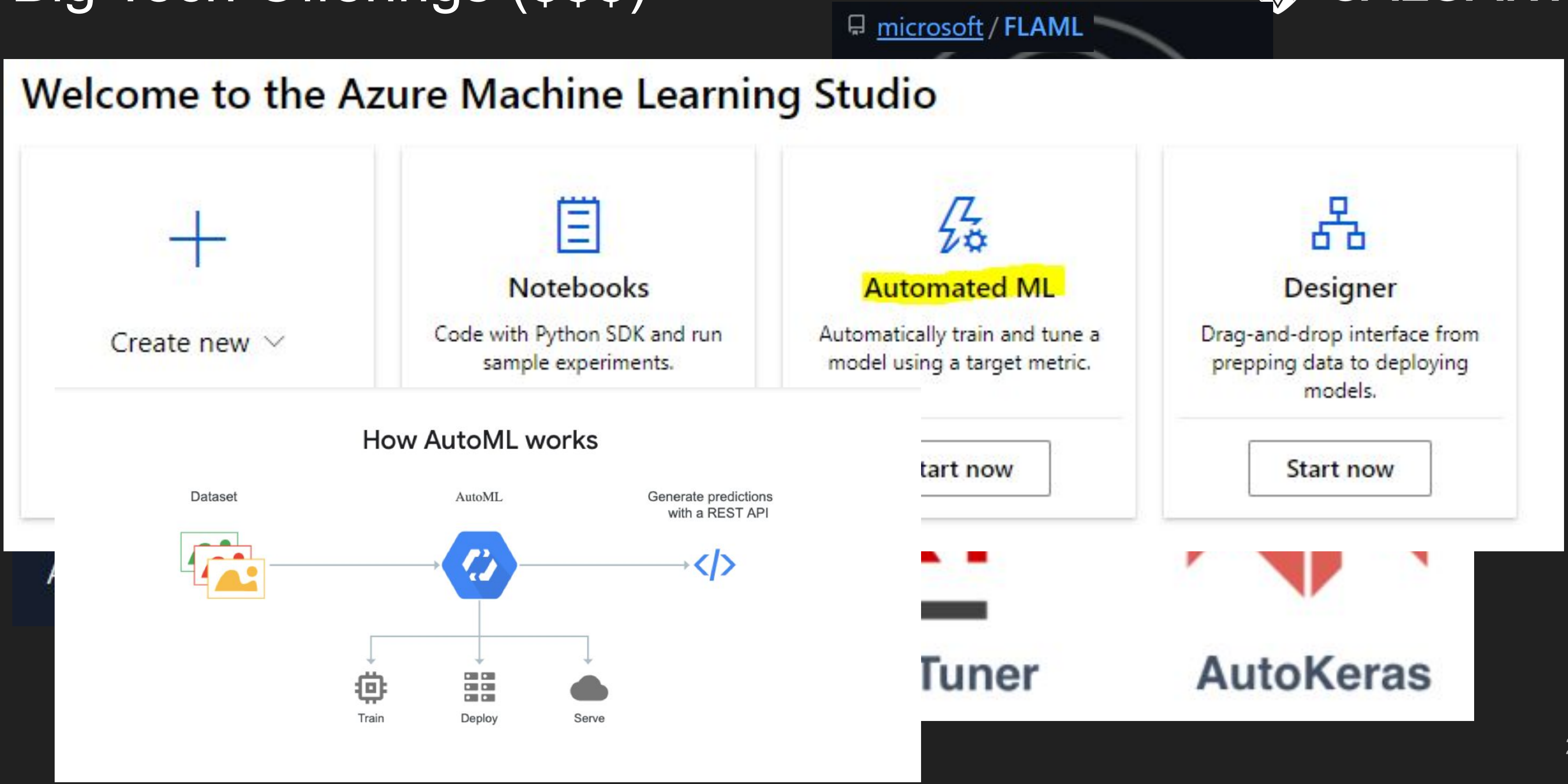
- Methods
  - Training free NAS?
    - Examine overlaps between ReLU activations between data points in untrained networks
    - Extremely low computational overhead
    - Unproven outside of NAS-benchmark datasets



(a) NAS-Bench-201

J. Mellor, J. Turner, A. Storkey, E.J. Crowley, “Neural Architecture Search without Training,” Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021

Big Tech Offerings ($$$)



Open Source Frameworks

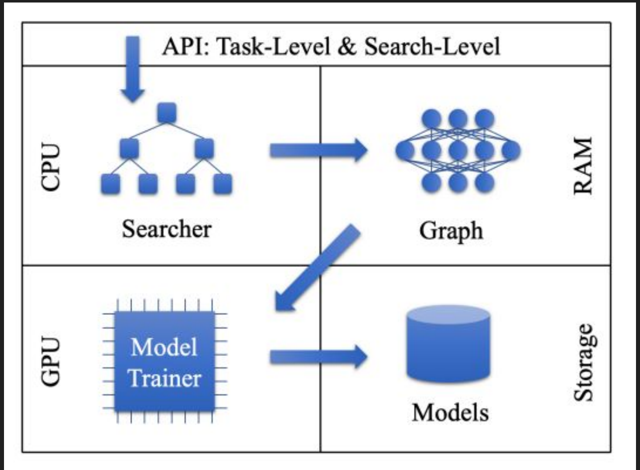
UNIVERSITY OF CALGARY

Traditional Machine Learning

- Auto-sklearn
- TPOT
- PyCaret
- H20 AutoML
- Hyperopt
- Many more

Deep Learning

- AutoKeras
- Keras Tuner
- Auto-PyTorch



H. Jin, Q. Song, and X. Hu, “Auto-Keras: An Efficient Neural Architecture Search System,” arXiv:1806.10282 [cs, stat], Mar. 2019, Accessed: Nov. 25, 2021. [Online]. Available: http://arxiv.org/abs/1806.10282

```
from tpot import TPOTClassifier
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split

digits = load_digits()
X_train, X_test, y_train, y_test = train_test_split(digits.data, digits.target,
                                                    train_size=0.75, test_size=0.25)

tpot = TPOTClassifier(generations=5, population_size=50, verbosity=2, n_jobs=-1)
tpot.fit(X_train, y_train)
```

```
Optimization Progress:   0%|          | 0/300 [00:00<?, ?pipeline/s]
```

```
print(tpot.score(X_test, y_test))
```

SCHULICH
School of Engineering

UNIVERSITY OF CALGARY

Questions?