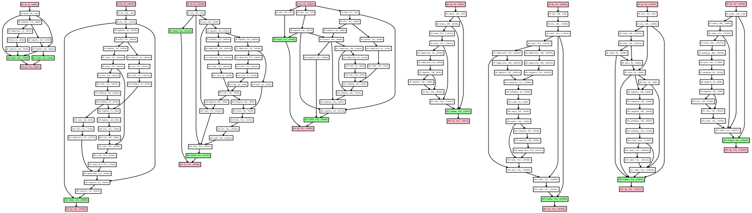
Neural Architecture Search with Bayesian Optimisation and Optimal Transport



Kirthevasan Kandasamy

Willie Neiswanger, Jeff Schneider, Barnabás Póczos, Eric Xing

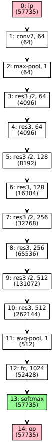
Carnegie Mellon University

Auton Lab

NeurIPS 2018

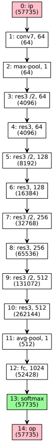
Montreal, Canada

Neural Architecture Search



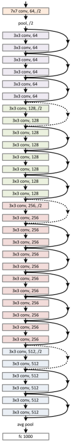
Feedforward network

Neural Architecture Search

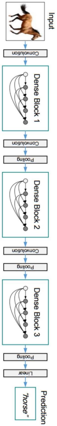


Feedforward network

GoogLeNet (Szegedy et al. 2015)

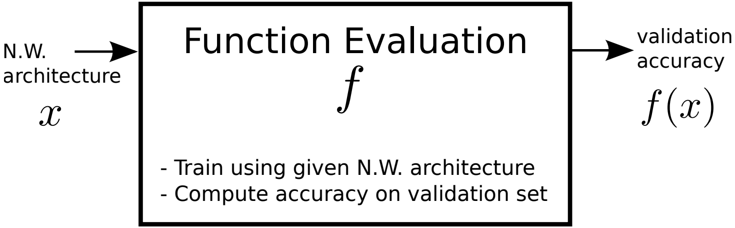


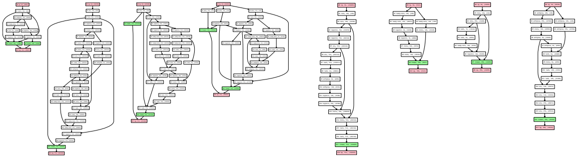
ResNet (He et al. 2016)



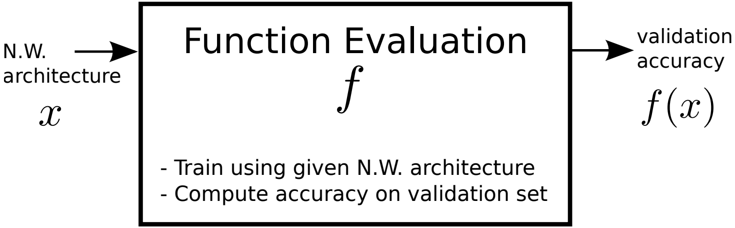
DenseNet (Huang et al. 2017)

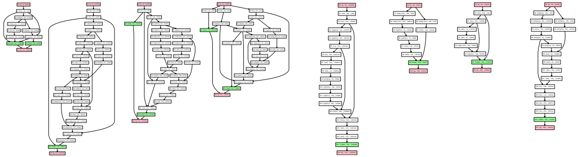
Neural architecture search is a zeroth order optimisation problem where each function evaluation is expensive.





Neural architecture search is a zeroth order optimisation problem where each function evaluation is expensive.





Bayesian Optimisation methods are well suited for optimising expensive blackbox functions.

Prior Work in Neural Architecture Search

Based on Reinforcement Learning:

(Baker et al. 2016, Zhong et al. 2017, Zoph & Le 2017, Zoph et al. 2017)

RL is more difficult than optimisation (Jiang et al. 2016).

Based on Evolutionary Algorithms:

(Kitano 1990, Stanley & Miikkulainen 2002, Floreano et al. 2008, Liu et al. 2017, Miikkulainen et al. 2017, Real et al. 2017, Xie & Yuille 2017)

EA works well for optimising cheap functions, but not when function evaluations are expensive.

Other:

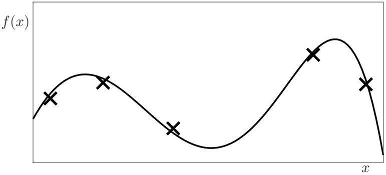
(Swersky et al. 2014, Mendoza et al. 2016, Negrinho & Gordon 2017, Jenatton et al. 2017)

Mostly search among feed-forward structures.

And a few more in the last two years ...

Bayesian Optimisation

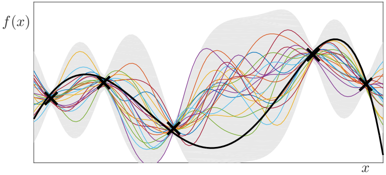
At each time step



Bayesian Optimisation

At each time step

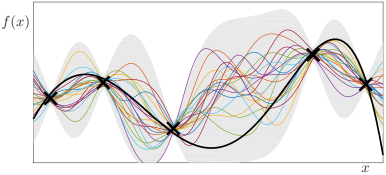
Compute posterior $\mathcal{GP}$



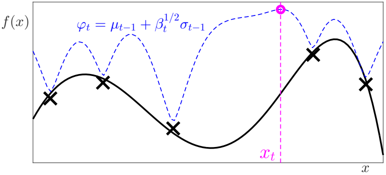
Bayesian Optimisation

At each time step

Compute posterior $\mathcal{GP}$



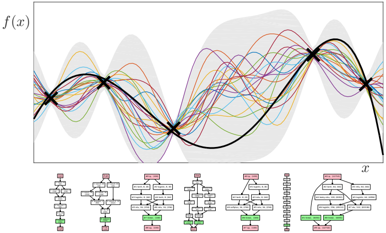
Maximise acquisition



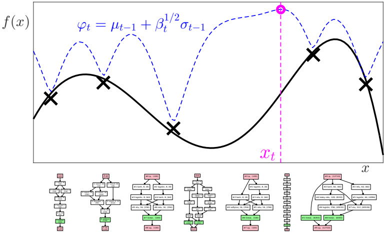
Bayesian Optimisation

At each time step

Compute posterior $\mathcal{GP}$



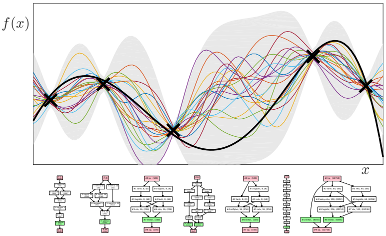
Maximise acquisition



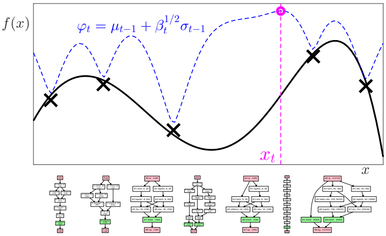
Bayesian Optimisation

At each time step

Compute posterior $\mathcal{GP}$



Maximise acquisition

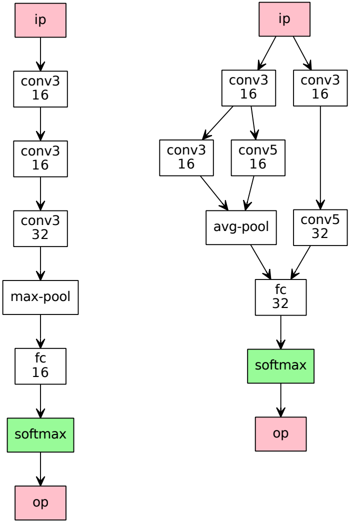


Bayesian optimisation for Neural Architecture Search

- Define a kernel between neural network architectures.
- Optimise acquisition in the space of neural networks.

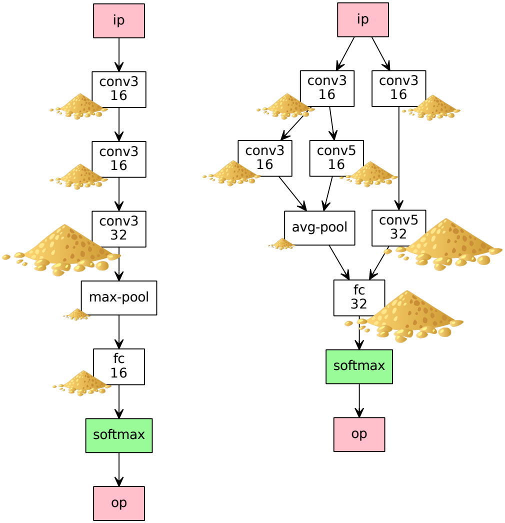
OTMANN: A optimal transport based distance for neural architectures.

Given this distance d, we use $e^{-\beta d}$ as the kernel.



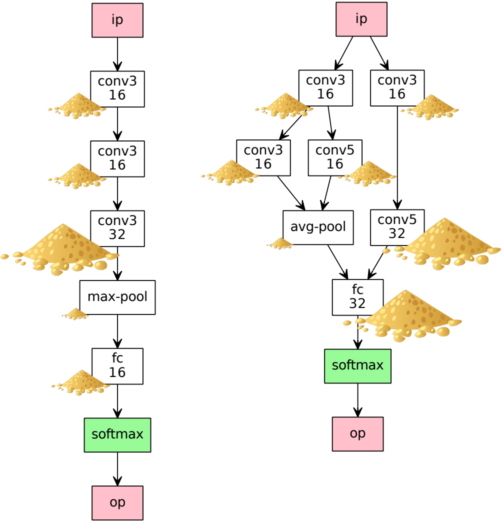
OTMANN: A optimal transport based distance for neural architectures.

Given this distance d, we use $e^{-\beta d}$ as the kernel.



OTMANN: A optimal transport based distance for neural architectures.

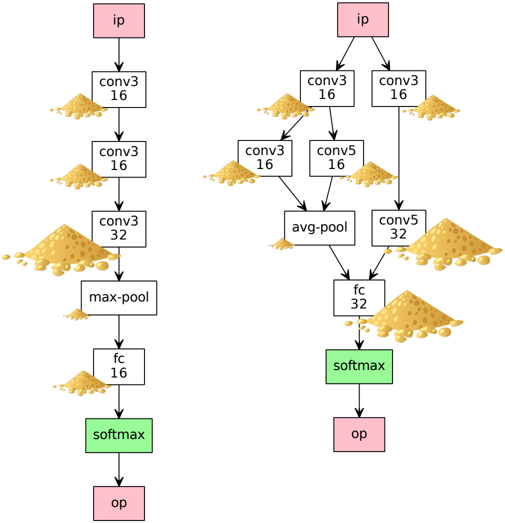
Given this distance d, we use $e^{-\beta d}$ as the kernel.



Penalty function:
- type of operation.
- structural position.

OTMANN: A optimal transport based distance for neural architectures.

Given this distance d, we use $e^{-\beta d}$ as the kernel.

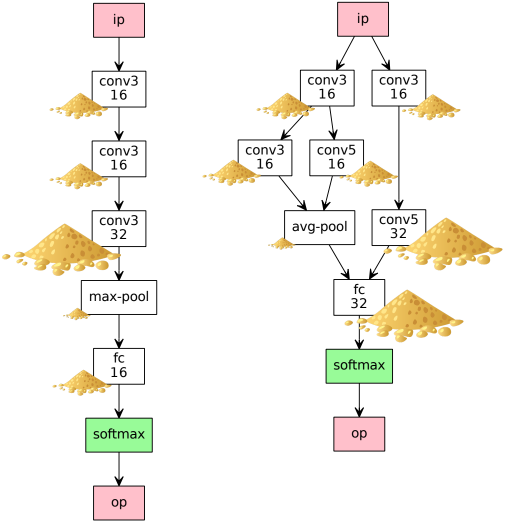


Penalty function:
- type of operation.
- structural position.

Can be computed via an optimal transport scheme.

OTMANN: A optimal transport based distance for neural architectures.

Given this distance d, we use $e^{-\beta d}$ as the kernel.

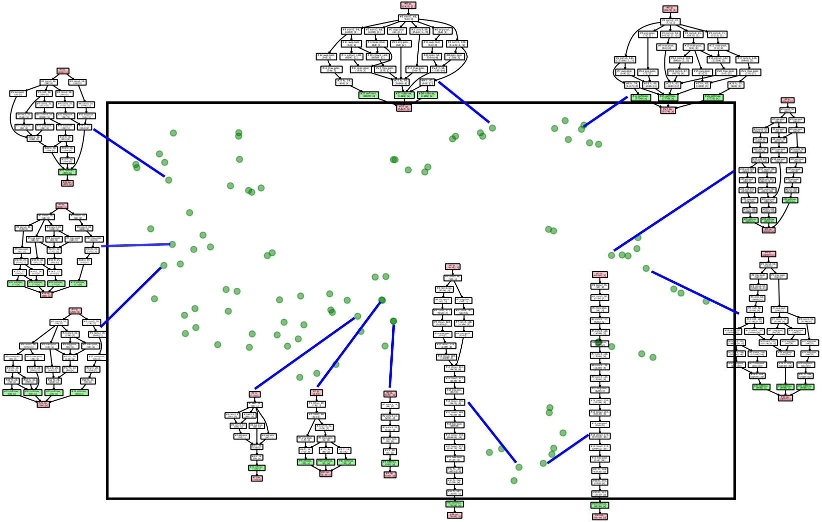


Penalty function:
- type of operation.
- structural position.

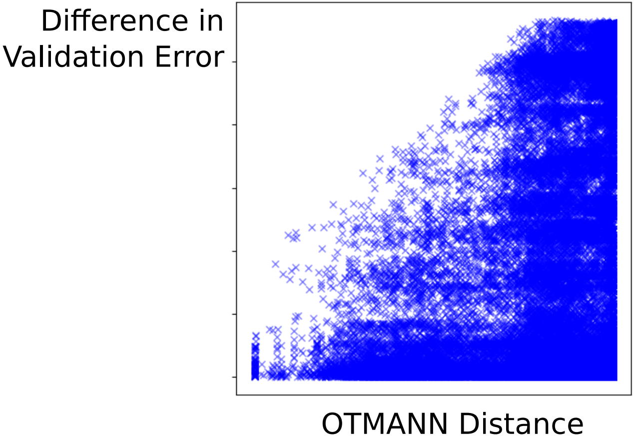
Can be computed via an optimal transport scheme.

Theorem: OTMANN is a pseudo-distance.

OTMANN: Illustration with tSNE Embeddings



OTMANN correlates with cross validation performance



Optimising the acquisition

Modifiers to navigate search space:
inc_single, dec_single, inc_en_masse, dec_en_masse, remove_layer, wedge_layer, swap_layer, dup_path, skip_path.

Apply an evolutionary algorithm using these modifiers.

Optimising the acquisition

Modifiers to navigate search space:
inc_single, dec_single, inc_en_masse, dec_en_masse, remove_layer, wedge_layer, swap_layer, dup_path, skip_path.

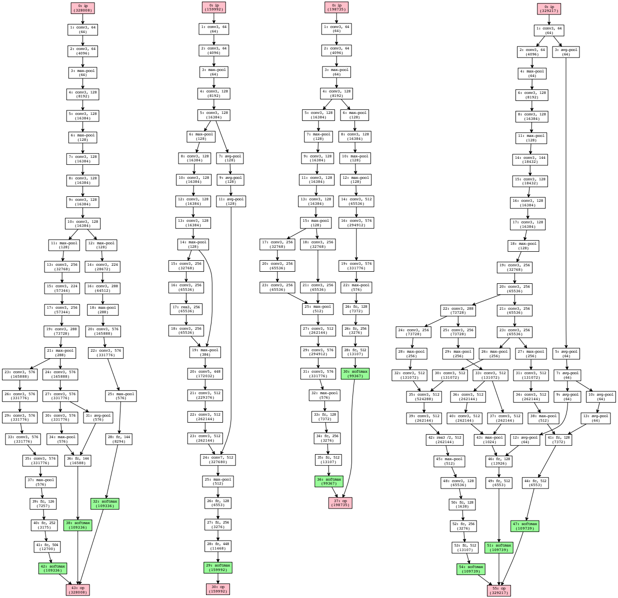
Apply an evolutionary algorithm using these modifiers.

Resulting procedure: NASBOT
Neural Architecture Search with Bayesian Optimisation and Optimal Transport (Kandasamy et al. NeurIPS 2018)

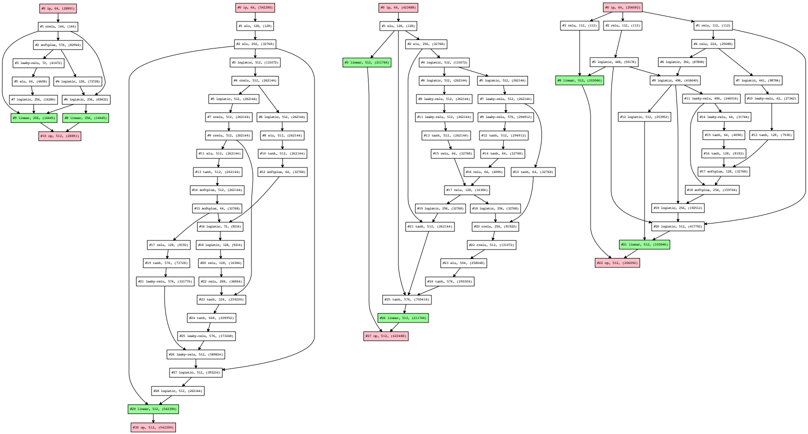
Test Error on 7 Datasets

Method	Blog $(60K, 281)$	Indoor $(21K, 529)$	Slice $(54K, 385)$	Naval $(12K, 17)$	Protein $(46K, 9)$	News $(40K, 61)$	Cifar10 $(60K, 1K)$	Cifar10 $150K$ iters
RAND	0.780 $\pm$ 0.034	**0.115 $\pm$0.023**	0.758 $\pm$ 0.041	0.0103 $\pm$ 0.002	0.948 $\pm$ 0.024	**0.762 $\pm$0.013**	0.1342 $\pm$ 0.002	0.0914 $\pm$ 0.008
EA	0.806 $\pm$ 0.040	0.147 $\pm$ 0.010	0.733 $\pm$ 0.041	**0.0079 $\pm$0.004**	1.010 $\pm$ 0.038	**0.758 $\pm$0.038**	0.1411 $\pm$ 0.002	0.0915 $\pm$ 0.010
TreeBO	0.928 $\pm$ 0.053	0.168 $\pm$ 0.023	0.759 $\pm$ 0.079	0.0102 $\pm$ 0.002	0.998 $\pm$ 0.007	0.866 $\pm$ 0.085	0.1533 $\pm$ 0.004	0.1121 $\pm$ 0.004
NASBOT	**0.731 $\pm$0.029**	**0.117 $\pm$0.008**	**0.615 $\pm$0.044**	**0.0075 $\pm$0.002**	**0.902 $\pm$0.033**	**0.752 $\pm$0.024**	**0.1209 $\pm$0.003**	**0.0869 $\pm$0.004**

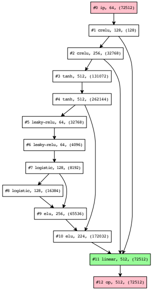
Architectures found on Cifar10

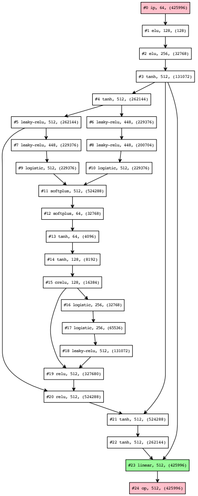


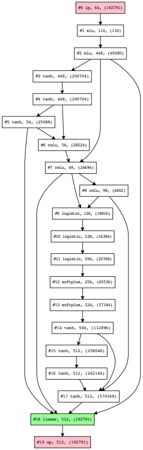
Architectures found on Indoor Location

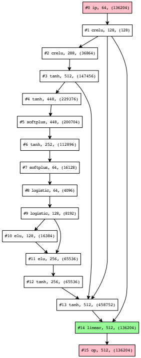


Architectures found on Slice Localisation









Willie
Neiswanger

Jeff
Schneider

Barnabás
Póczos

Eric
Xing

Carnegie Mellon University

Auton Lab

Code: `github.com/kirthevasank/nasbot`

Poster: AB #166