HARVARD
School of Engineering and Applied Sciences

Neural Architecture Search

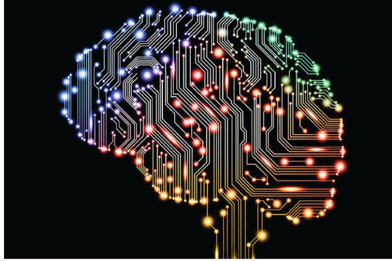
Harvard Data Science Capstone 2019
Midterm Presentation

Team
Michael S. Emanuel
Julien Laasri
Dylan Randle
Jiawei Zhuang

IACS HARVARD School of Engineering and Applied Sciences

Contents

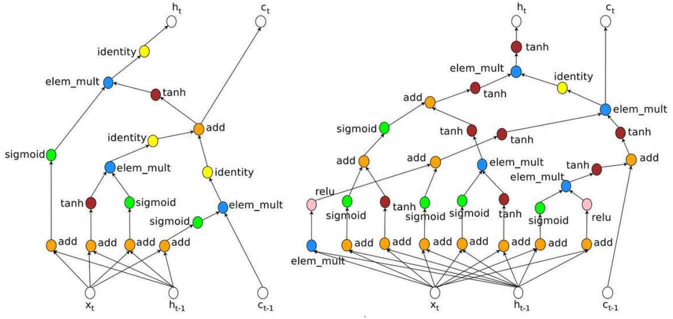
Background: Google Brain



- Google, Inc
  - Market cap of $842 bn; $137bn revenues 2018; ~103k employees
  - Google is a tech company, a household name, and a verb
- Google Brain started as research collaboration between Google and academics interested in deep learning
  - Andrew Ng, Geoff Hinton
  - Google Brain has been considered very successful
- Google Translate: Most visible success of Google Brain
  - Now powered by a RNNs developed at G.B.; big increase in quality
- Research Areas: Machine Learning Algos, Computer Systems for ML, Robotics, People & AI Research, Natural Language Understanding, ML for Perception

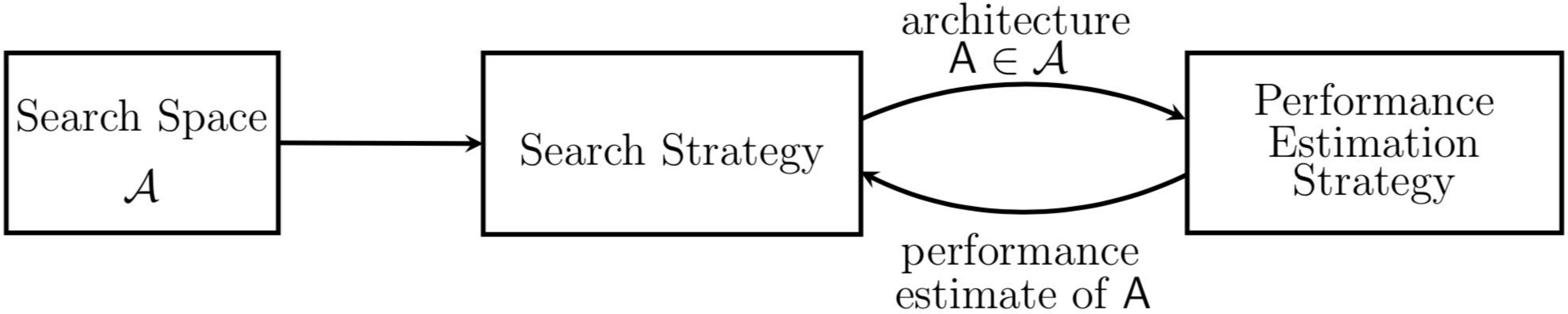
What is Neural Architecture Search?

- Training a neural network can be separated into two phases: selecting an *architecture*, and *fitting* the weights
- Today: designed by experts
  - Labor-intensive
  - Error-prone
- Tomorrow: **Neural Architecture Search (NAS)**
  - Systematically searching for best model architecture



- Interest in NAS is increasing rapidly: there is now far more demand for neural network models than available experts who can design model architectures
- Google already offering “AutoML” as a product

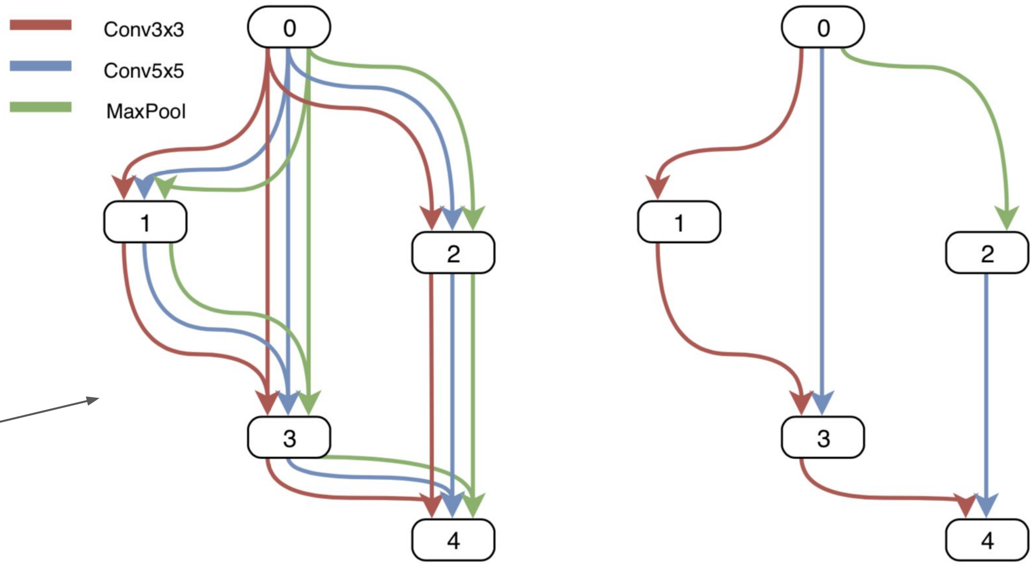
Neural Architecture Search Workflow



Credit: Elskin et. al, 2019

Performance Estimation Strategy

- Full model evaluation (costly)
- Lower fidelity estimates (biased)
- Learning curve extrapolation (difficult / unreliable)
- Weight inheritance (warmstart, less training)
- **One-shot:** architectures are subgraphs of supergraph



Credit: Elskin et. al, 2019

Overview of Approaches

- Most successful and accessible technique so far: DARTS
  - Uses a convex relaxation of discrete search space; allows for use of gradients
- ENAS (Efficient Neural Architecture Search)
  - Has also achieved good results; more complicated; uses an RNN controller network
- Random Search
  - Dirty secret of NAS: Random search almost as good as the best methods!
  - Shows the importance of picking a sensible search space, and the power of NAS
- Other Approaches: Evolutionary Algorithms, Bayesian Optimization, RL
  - Thus far these approaches have not shown as much promise as DARTS or ENAS
  - Some of them have achieved reasonable end results, but at the cost of huge energy use

DARTS (Differentiable ARchiTecture Search)

- Combines:
  a. **gradient-based search** with
  b. **one-shot performance evaluation**
- Uses much less compute resources compared to many other NAS methods

$$\min_{\alpha} \quad \mathcal{L}_{val}(w^*(\alpha), \alpha)$$
$$\text{s.t.} \quad w^*(\alpha) = \text{argmin}_w \; \mathcal{L}_{train}(w, \alpha)$$

Approximate architecture gradient: use a single inner (w) gradient step (bilevel optimization)

$$\nabla_\alpha \mathcal{L}_{val}(w^*(\alpha), \alpha)$$
$$\approx \nabla_\alpha \mathcal{L}_{val}(w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha), \alpha)$$

Liu et. al 2019

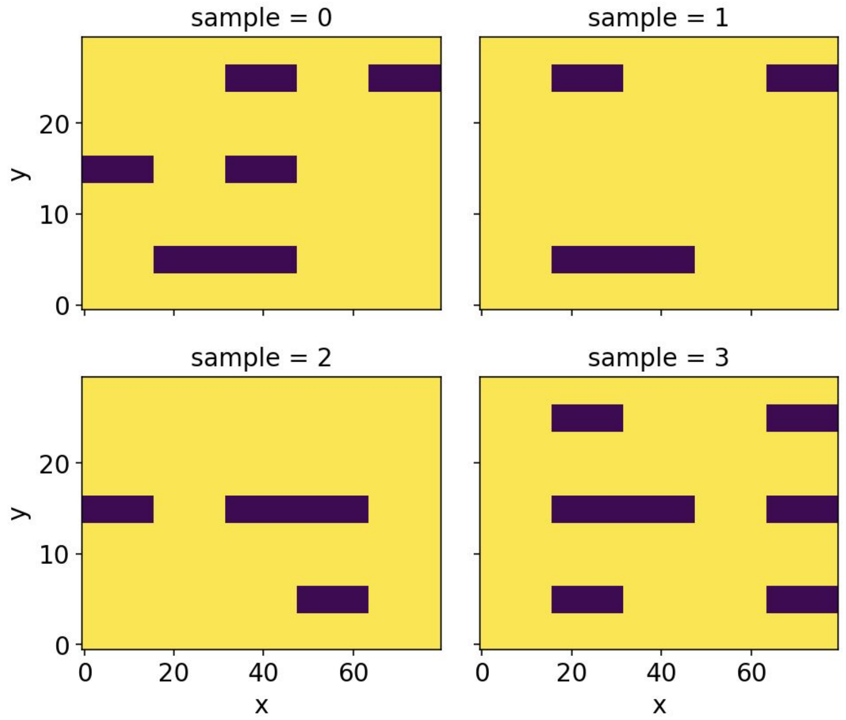
Hanakata et al. 2018, PRL

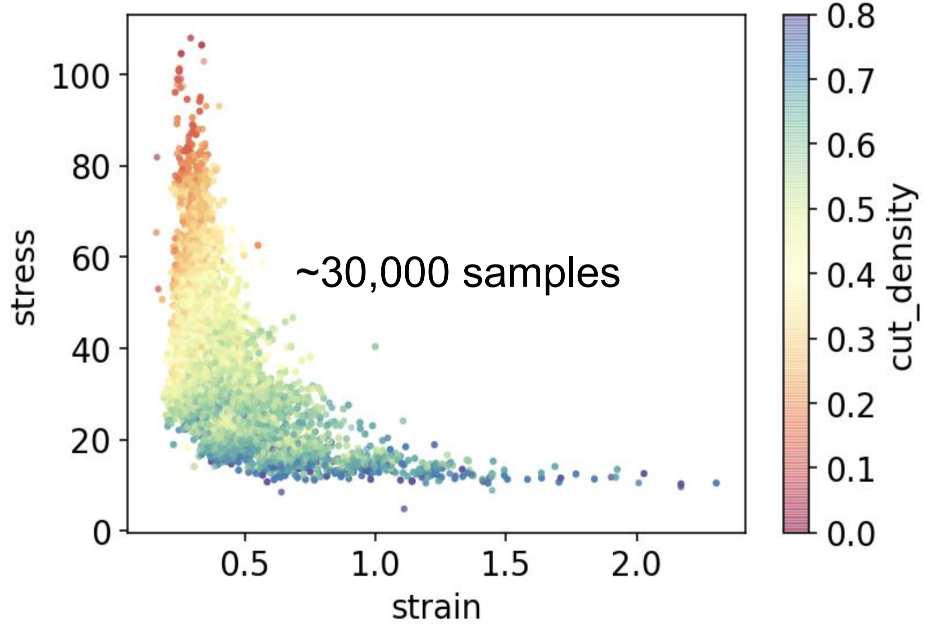
Graphene Kirigami Dataset

predict

cut configuration

stretch property





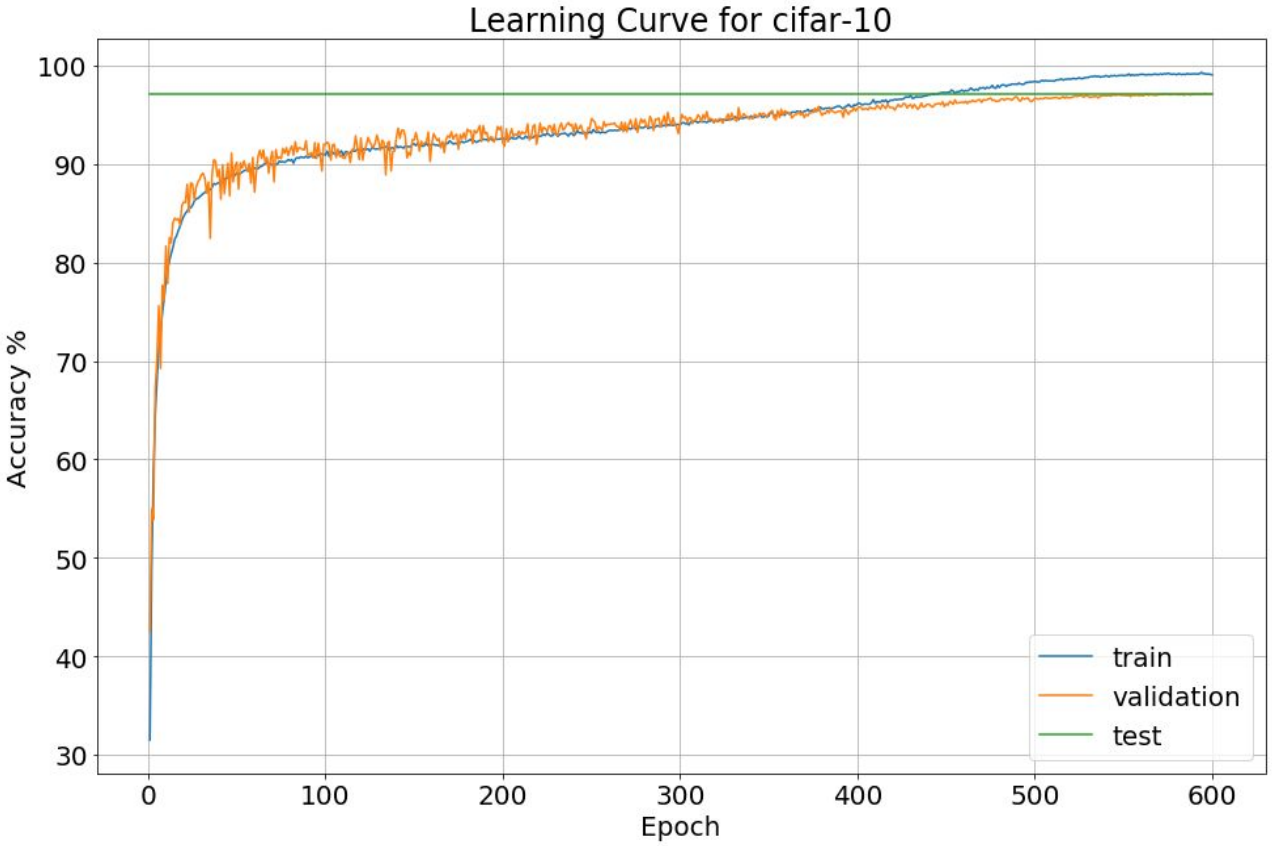
Results: Graphene Kirigami

Method / model	No. Parameters	Training time* (minutes)	Test R^2
DARTS	195,236	1042	0.92
ResNet-18	11,168,193	11	0.92
"Tiny ResNet" (10 layers, 8x less filters)	77,273	4	0.92

1. Time reported to train for 30 epochs on a single GPU.

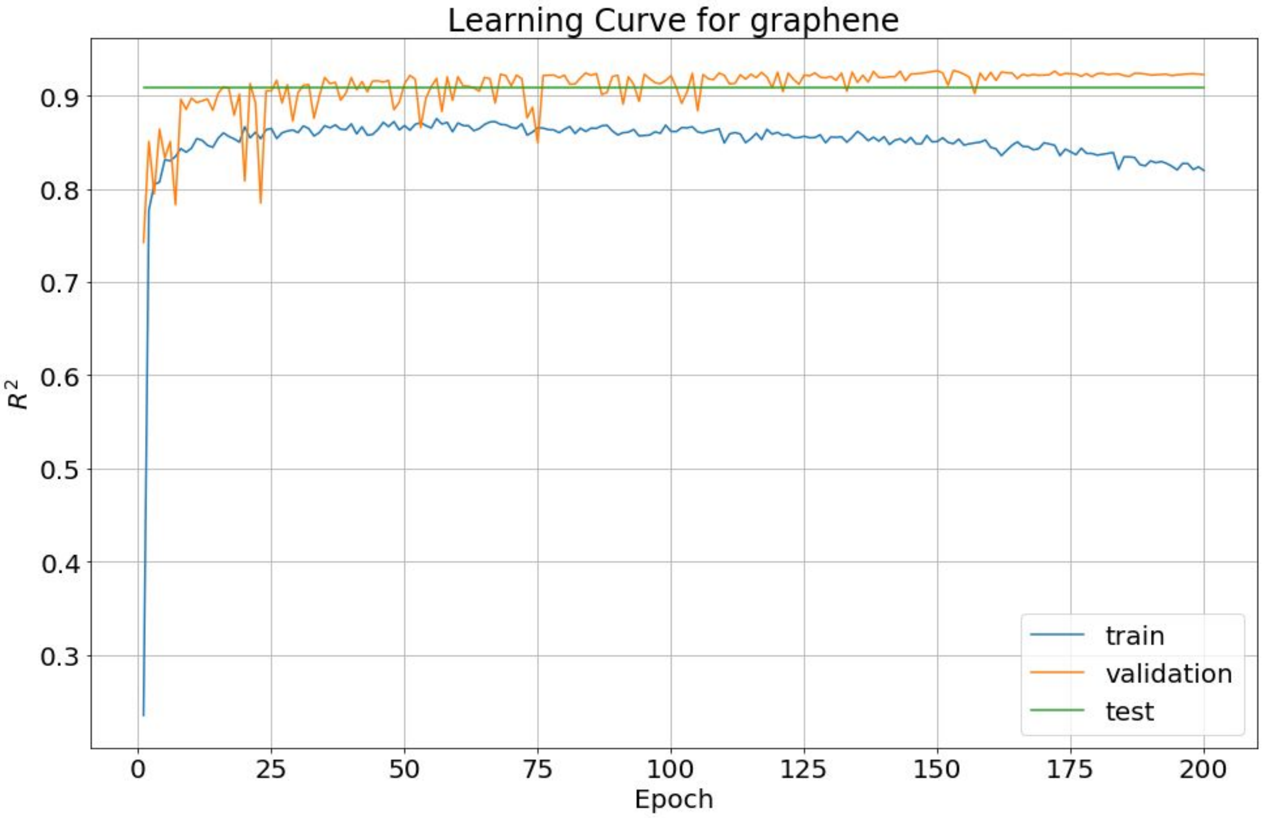
Conclusion: Graphene Kirigami dataset is too simple for DARTS to be useful.

Learning Curve: CIFAR-10



- Expected relationship between train/validation performance
- Some overfitting after epoch 400

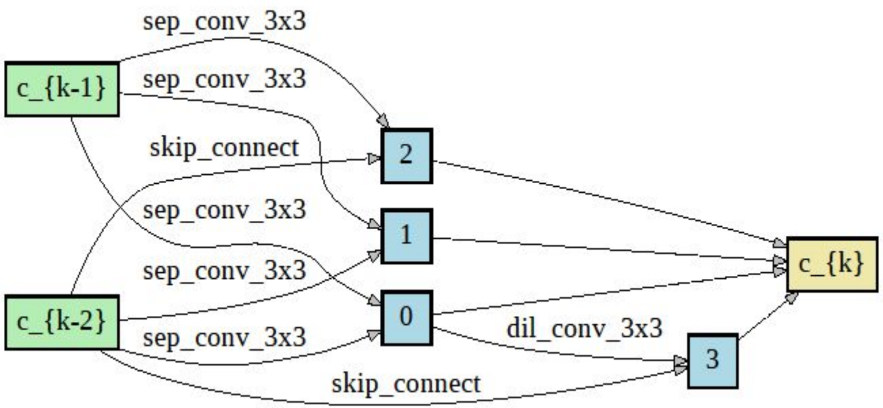
Learning Curve: Graphene



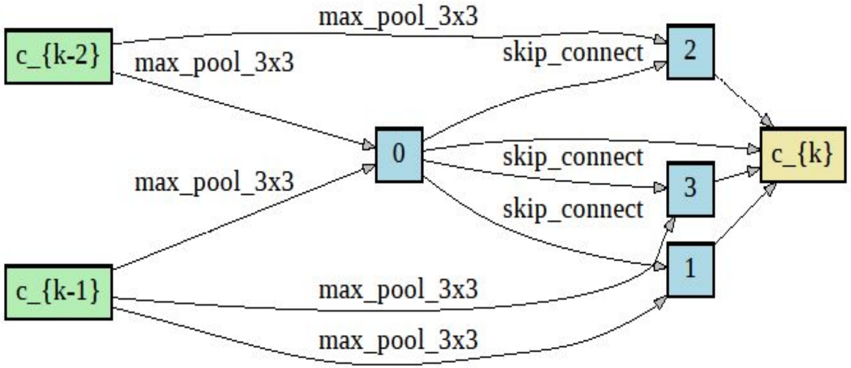
- Training diverges slightly after epoch ~100
- Additional tuning may be required to improve performance / avoid divergences

Learned Cells for CIFAR-10

Normal cell (stride-1 operations, keep image size the same)



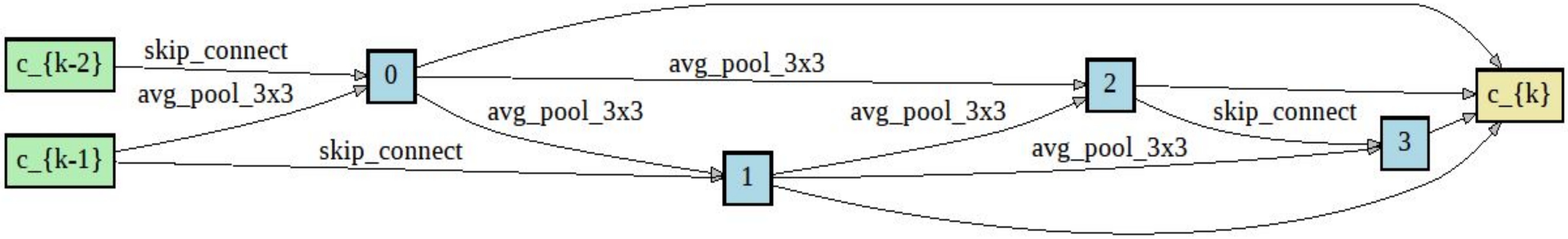
Reduction cell (stride-2 operations, down-sample image by factor of 2)



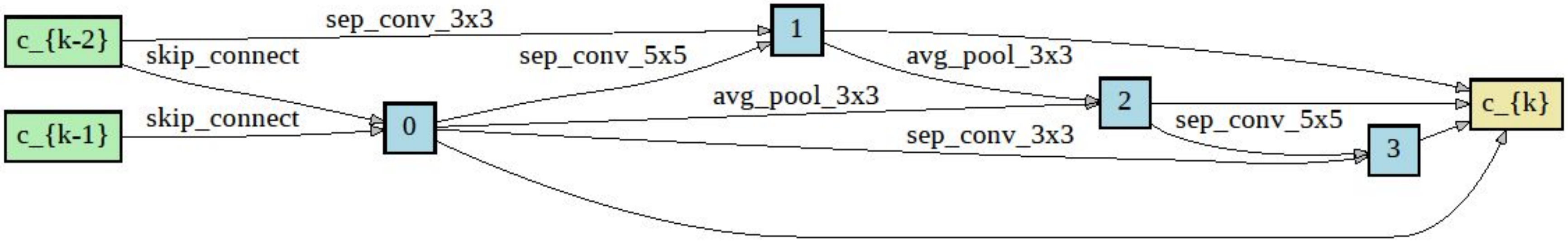
Learned Cells for Graphene

Mostly contain average pooling or skip connections. Those operations have no tunable parameters, consistent with the fact that the graphene data is too easy to fit well.

Normal cell (stride-1 operations, keep image size the same)

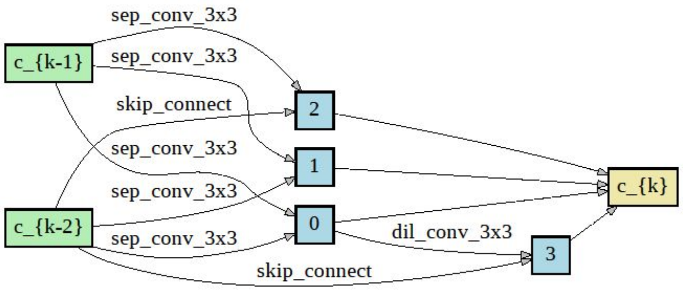


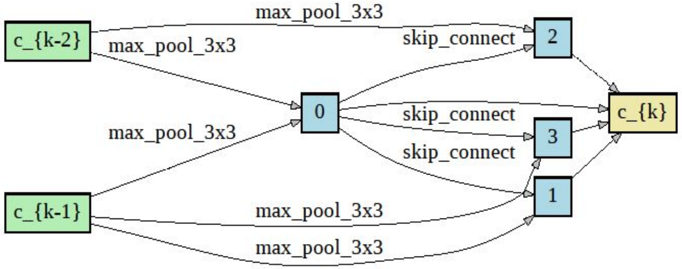
Reduction cell (stride-2 operations, down-sample image by factor of 2)



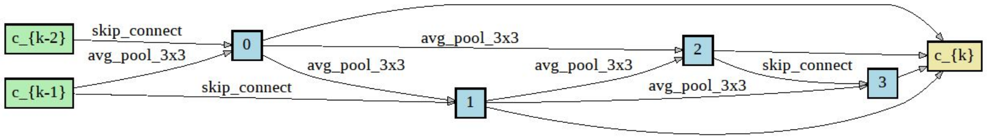
Learned Cells for CIFAR-10 and Graphene

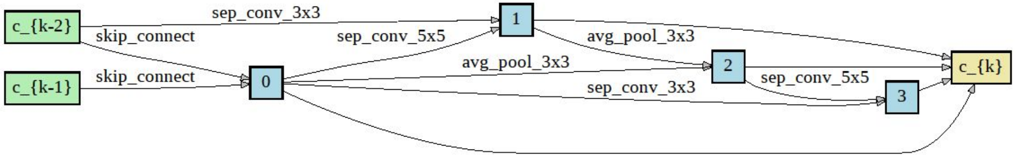
CIFAR-10: Normal & Reduction Cells





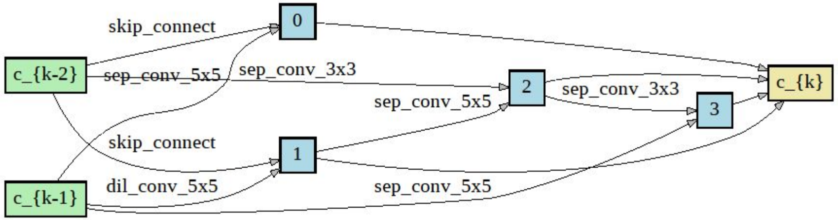
Graphene: Normal & Reduction Cells

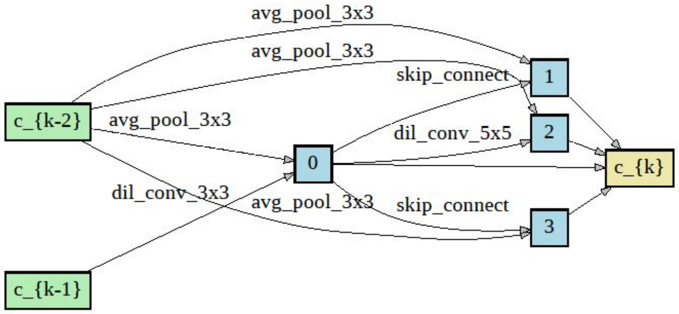




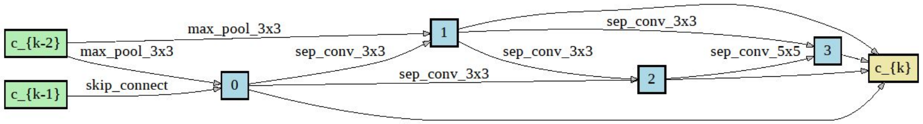
Learned Cells for FashionMNIST and MNIST

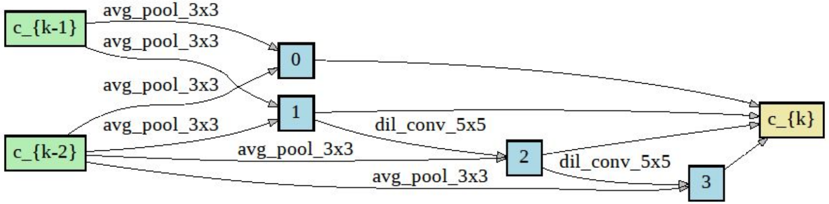
FashionMNIST: Normal & Reduction Cells



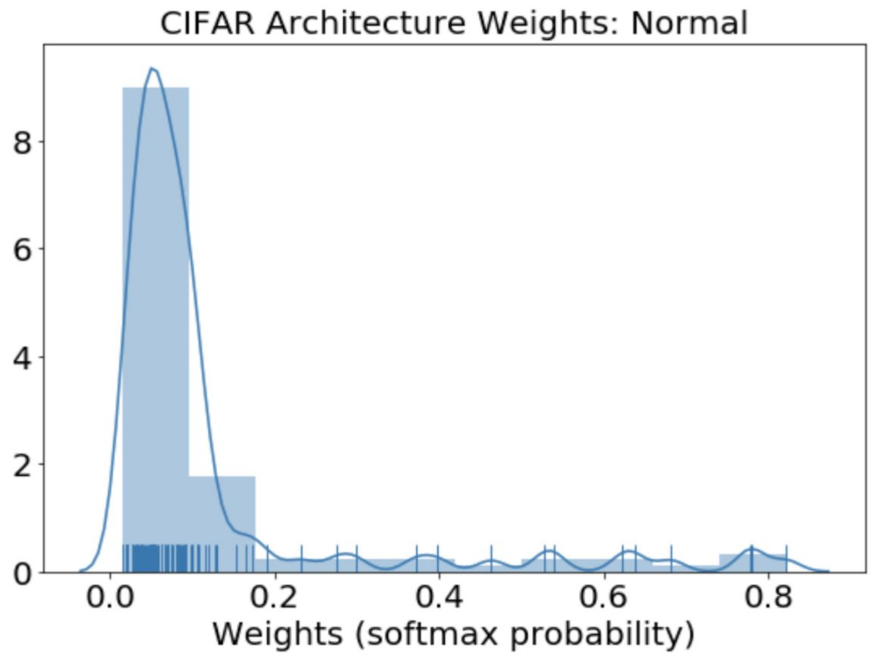


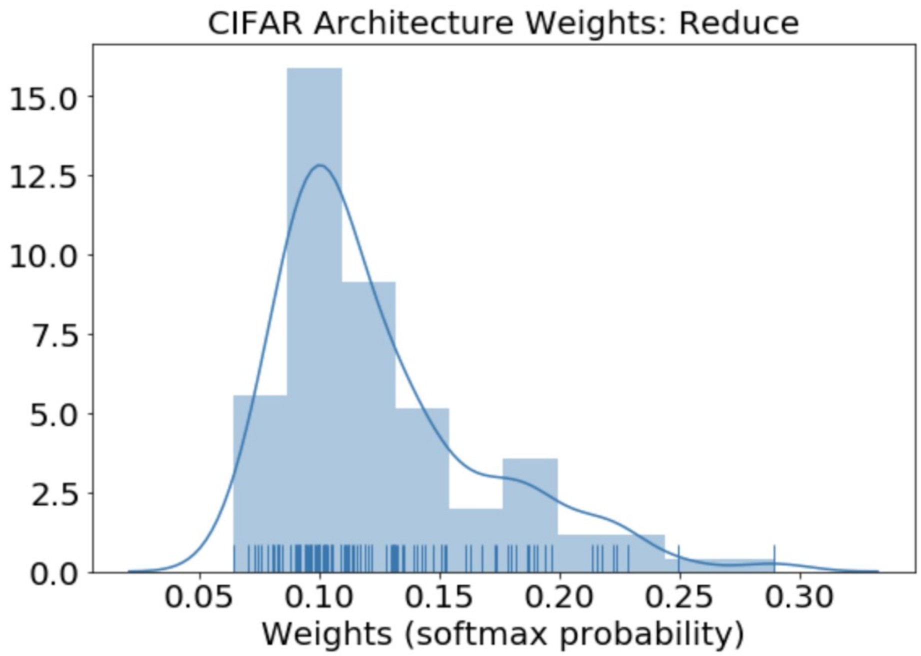
MNIST: Normal & Reduction Cells



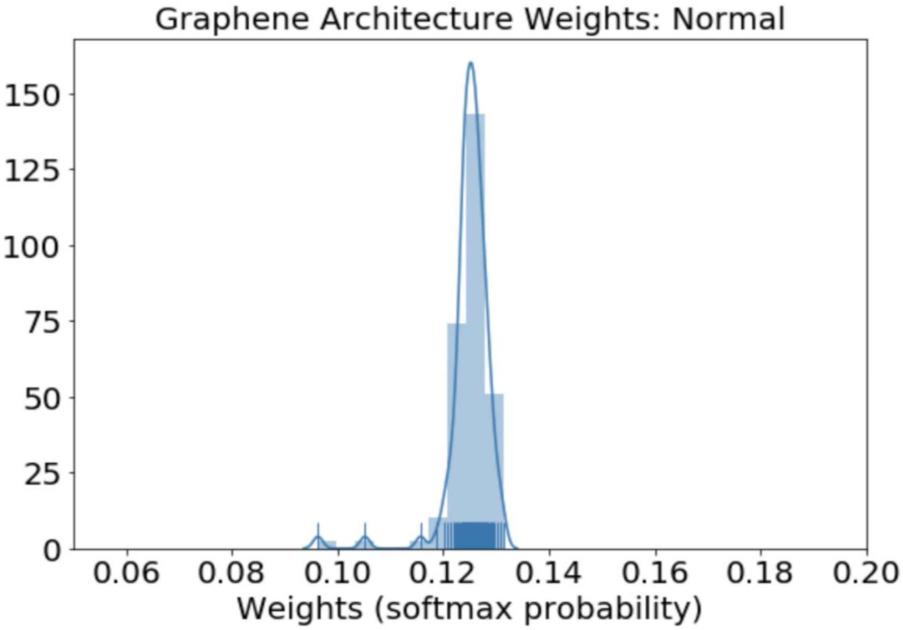


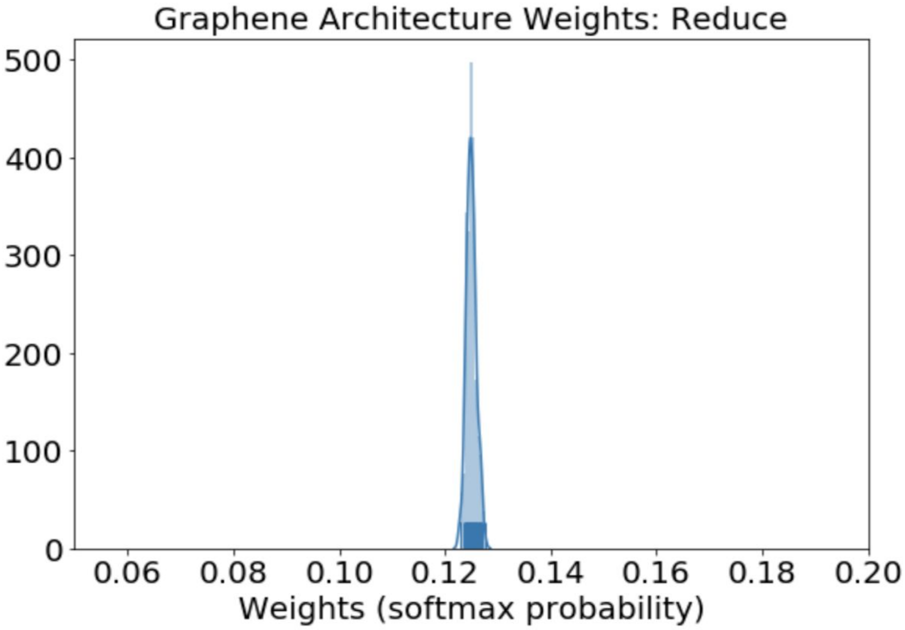
Learned Architecture Weights: CIFAR





Learned Architecture Weights: Graphene





Random Search Algorithm: A Baseline Approach

Table 1: Comparison with state-of-the-art image classifiers on CIFAR-10 (lower error rate is better). Note the search cost for DARTS does not include the selection cost (1 GPU day) or the final evaluation cost by training the selected architecture from scratch (1.5 GPU days).

Architecture	Test Error (%)	Params (M)	Search Cost (GPU days)	#ops	Search Method
DenseNet-BC (Huang et al., 2017)	3.46	25.6	–	–	manual
NASNet-A + cutout (Zoph et al., 2018)	2.65	3.3	2000	13	RL
NASNet-A + cutout (Zoph et al., 2018)†	2.83	3.1	2000	13	RL
BlockQNN (Zhong et al., 2018)	3.54	39.8	96	8	RL
AmoebaNet-A (Real et al., 2018)	3.34 ± 0.06	3.2	3150	19	evolution
AmoebaNet-A + cutout (Real et al., 2018)†	3.12	3.1	3150	19	evolution
AmoebaNet-B + cutout (Real et al., 2018)	2.55 ± 0.05	2.8	3150	19	evolution
Hierarchical evolution (Liu et al., 2018b)	3.75 ± 0.12	15.7	300	6	evolution
PNAS (Liu et al., 2018a)	3.41 ± 0.09	3.2	225	8	SMBO
ENAS + cutout (Pham et al., 2018b)	2.89	4.6	0.5	6	RL
ENAS + cutout (Pham et al., 2018b)*	2.91	4.2	4	6	RL
$\boxed{\text{Random search baseline}^\ddagger \text{ + cutout}}$	3.29 ± 0.15	3.2	4	7	random
DARTS (first order) + cutout	3.00 ± 0.14	3.3	1.5	7	gradient-based
DARTS (second order) + cutout	2.76 ± 0.09	3.3	4	7	gradient-based

Planned deliverables

- An **improved DARTS implementation** that works on all kinds of datasets
- A reproducible training & evaluation pipeline with Nvidia-Docker containers
- Evaluation on **more datasets**
- Random search and hand-designed nets for comparison
- Observations of patterns in discovered cells (including **sparsity, distributions of edge weights**) and network depths that perform well
- Advice for practitioners who wish to use DARTS on novel scientific datasets
- Study the **properties of learned architecture transfer** to other datasets

Potential Datasets

- Graphene Kirigami (https://arxiv.org/abs/1808.06111)
- PLAsTiCC Astronomical Classification (https://www.kaggle.com/c/PLAsTiCC-2018)
- Galaxy Zoo (https://data.galaxyzoo.org/)
- Music Genre Classification (https://github.com/mlachmish/MusicGenreClassification)
- Structural Optimization (https://arxiv.org/pdf/1909.04240.pdf)
- Potentially some radiology/healthcare imaging datasets

Timeline

Date	Milestone
10/15 - 10/22	Collect new datasets
10/23 - 11/01	Run experiments (DARTS, hand-designed, random search)
11/02 - 11/10	Visualize and analyze results
11/10 - 11/30	Evaluate learned architecture transferability, compile results into recommendations, use findings to suggest future direction