HARVARD
School of Engineering
and Applied Sciences

Neural Architecture Search

Harvard Data Science Capstone 2019

Team
Michael S. Emanuel
Julien Laasri
Dylan Randle
Jiawei Zhuang

Scope of Work and Collaboration Infrastructure

Scope of Work

- Run DARTS on standard ML and scientific datasets
- Develop experimentation framework for rapid testing and discovery
  - Agnostic to model (operations), dataset, and compute infrastructure (local, Google cloud)
- Compare results achieved by DARTS/NAS with:
  - Best human designed network for problem class
  - Random search
- Determine whether state of the art CNN architectures perform well on scientific datasets
  - Formulate recommendations for researchers interested in applying ML to a new scientific data set: apply modern CNN architectures, or run NAS directly?
- Maintain a blog of our progress

Team and Collaboration Infrastructure

- Our team has a dedicated Slack channel for this project
  - This is our primary daily communications channel
- All source code is on a team GitHub repository at https://github.com/capstone2019-neuralsearch/AC297r_2019_NAS
- Our team plans on 100% attendance at all Tuesday lectures, meetings with TF Javier, and with Instructor Pavlos
- We are still working to develop procedures to collaborate by accessing shared compute resources
  - We plan to use Google Cloud and / or Jupyter Hub instances
- Once our shared infrastructure is working, different team members can explore NAS on different datasets in parallel

Problem Statement:
Neural Architecture Search

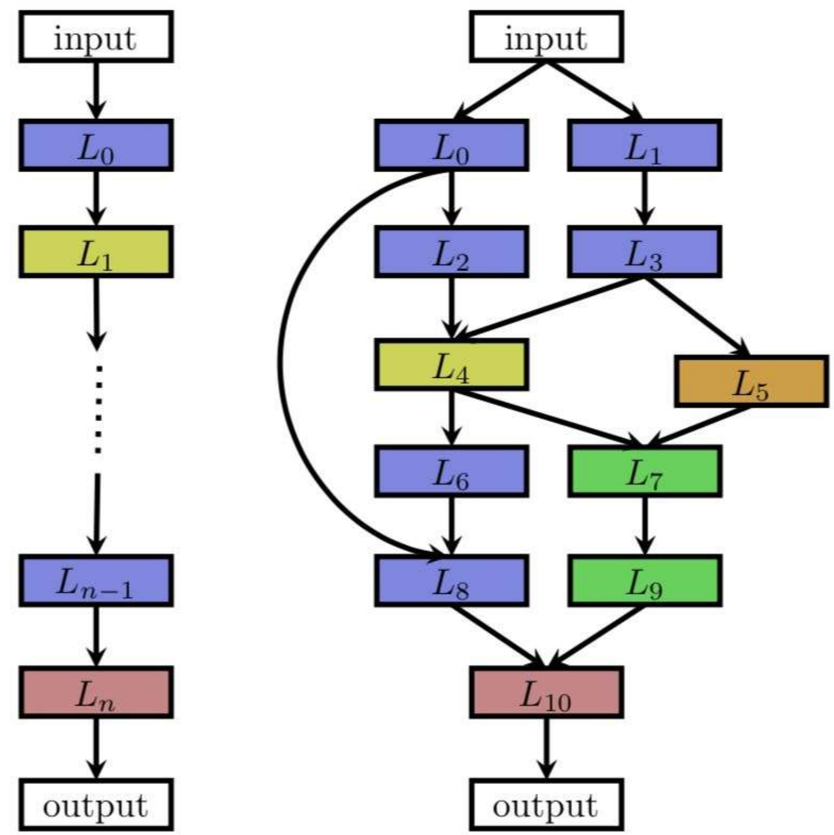
What is Neural Architecture Search? Why Now?

- Building a neural network can be separated into two phases: selecting a network ***architecture***, and ***training*** the parameters (weights and biases)
- Most neural networks today have architectures designed by experts
  - This is a laborious and error prone process, and a significant pain point among users
- **Neural Architecture Search** (NAS) is a technique for systematically searching a space of candidate network architectures to identify a good one
- Interest in NAS is increasing rapidly: there is now far more demand for neural network models than available experts who can design model architectures
- The dream of NAS is to reach a point where a user can input a data set and receive a high performing trained model
  - Google is trying to commercially realize this with its AutoML product

Neural Architecture Search (NAS)

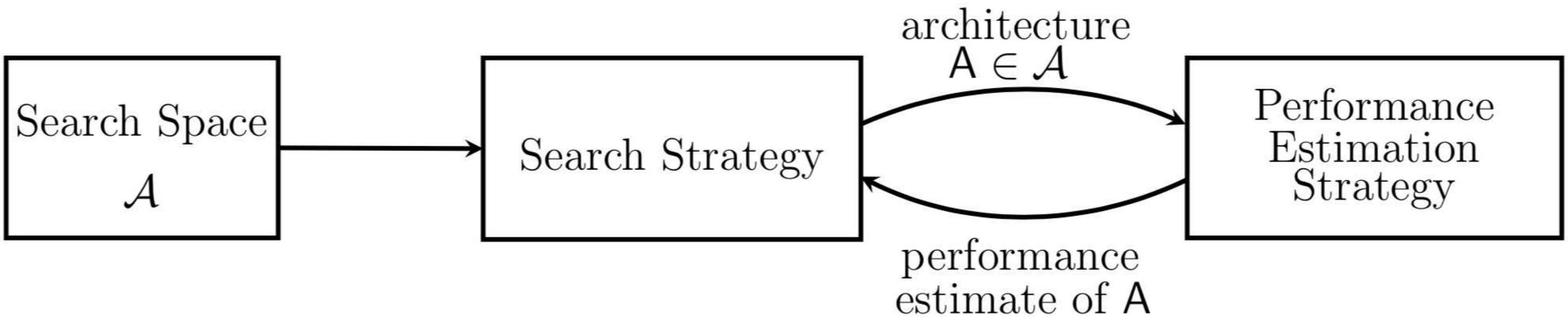
Search Space:

- Number of layers (unbounded)
- Type of layer (e.g. convolution, pooling)
- Hyperparameters (e.g. # filters, kernel size)
- Connectivity of layers (e.g. Chain, Residual, Dense)
    - Functions that combine previous layer outputs



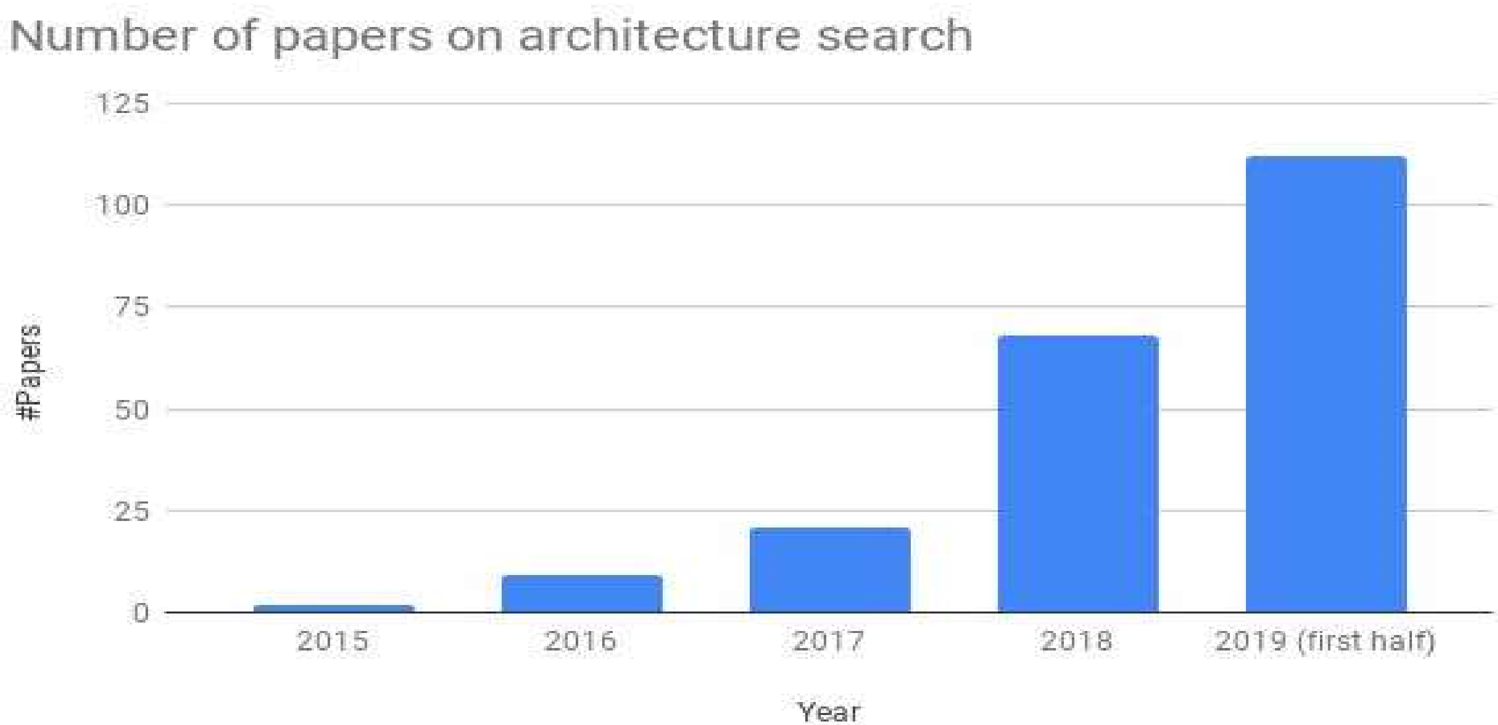
Credit: Elskin et. al, 2019

Neural Architecture Search Workflow



Credit: Elskin et. al, 2019

Academic Interest in NAS is Surging



NAS papers per year based on the literature list on automl.org. The number for 2019 only considers the first half of 2019.
(Lindauer and Hutter, 2019)

Learning Goals

- Literature review on NAS
  - All team members have read cited papers in project description
- Run DARTS on a set of selected datasets-from ML and scientific literature
  - ML datasets: CIFAR-10, ImageNet
  - Scientific datasets: LAMMPS Molecular Dynamics (graphene), Qure25k (head CT scans), PLAsTiCC (astronomical object classification), RFS Weather
- Understand and compare state of the art architectures: VGG, GoogLeNet, ResNet, DenseNets, Highway Networks
  - Read additional literature to gain understanding of these architectures
  - Possibly run DARTS using these high level architectures, but with an optimized cell
- Gain insights on Computational and Performance Limitations of NAS/DARTS
  - We will learn by doing and report on our experiences

Relevant Knowledge
And Literature Review

Search Strategy

$$\bar{o}^{(i,j)}(x) = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})} o(x)$$

"Softmax over operations"

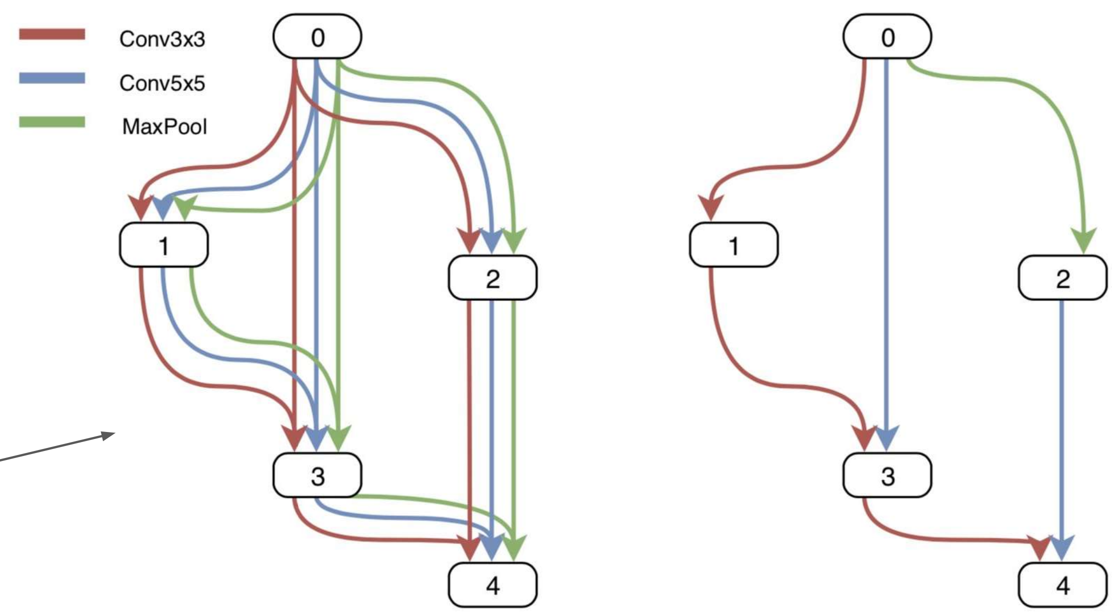
Credit: Liu et. al, 2019

- Random search
- Bayesian optimization
- Evolutionary algorithms
- Reinforcement learning
- **Gradient-based methods:** continuous relaxation of discrete search space
  - Convex combination of operations
  - Parameterize architecture of network
  - Alternate gradient updates of parameters (training set) and architecture (validation set)

Improvements over random search have, so far, been modest.

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

DARTS

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

Credit: Wikipedia

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

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

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

Liu et. al 2019

Project Ideas

Plan for Next Four Weeks

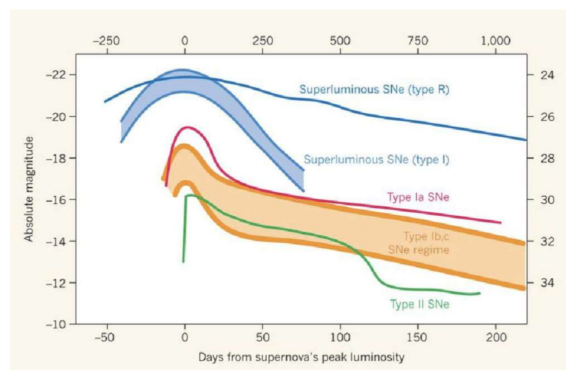
- Get reference DARTS implementation up and running
  - Use it train Cifar-10; replicate results of the DARTS paper
- Experiment with DARTS on two scientific datasets: graphene and astronomy
  - Start by using similar network types (normal / reduction cells) as in DARTS
  - Experiment with different cell types and network structures
- Compare and contrast with:
  - Best Human-Designed: ResNet / DenseNet / HighwayNet
  - Random Search: randomly sampling in architecture space
- Develop experimentation framework for rapid testing
  - Design codebase / abstractions before we write code!
  - Agnostic to model (operations), dataset (scientific, non-scientific, etc), and compute infrastructure (local machine, Google Compute, etc.)
  - This can happen throughout the process of discovery

Exploratory Data Analysis

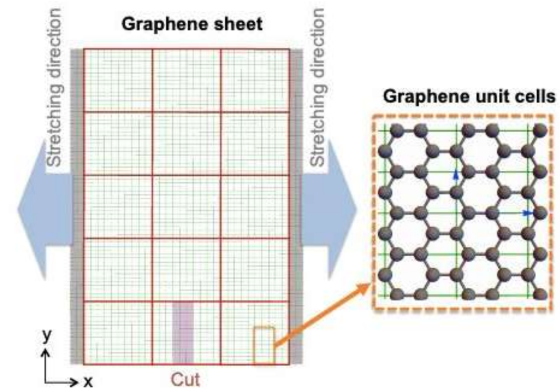
Datasets: Overview

- Image-based scientific datasets
  - Molecular dynamics simulator (graphene stretching)
  - Head CT scans
  - Classification of astronomical objects
  - Weather classification

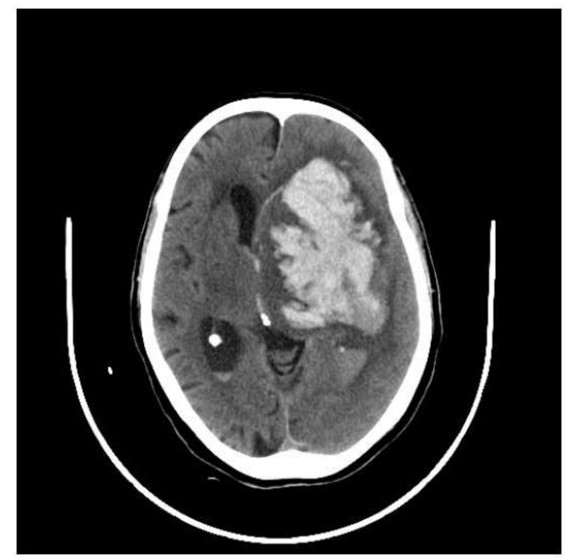
Credit: Guerra et. al 2018



Credit: Narayan, Kaggle.com



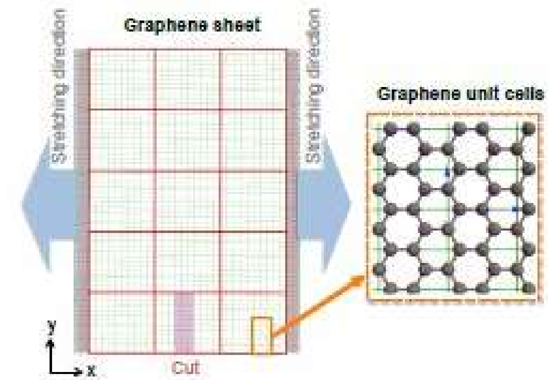
Credit: Hanakata et. al 2018

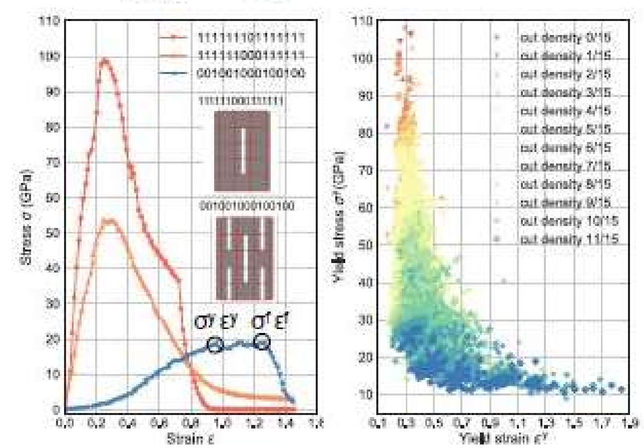


Credit: Chilamkurthy et. al 2018

Dataset: LAMMPS -Stretchable Graphene Kirigami

- This dataset was generated with the Sandia labs LAMMPS molecular simulator
- It contains the stress/strain plots for 29,791 kirigami configurations, costing ~120,000 CPU hours
- The published version includes source code only, but not this computationally expensive data set
- Google has promised to share the data with us, but has not yet done so
- Due to the high energy cost (~15,600 kWh assuming 130 W per CPU) we are waiting for Google
    - The average US household uses 867 kWh / month, so this data set is equivalent to 18 months of household energy!





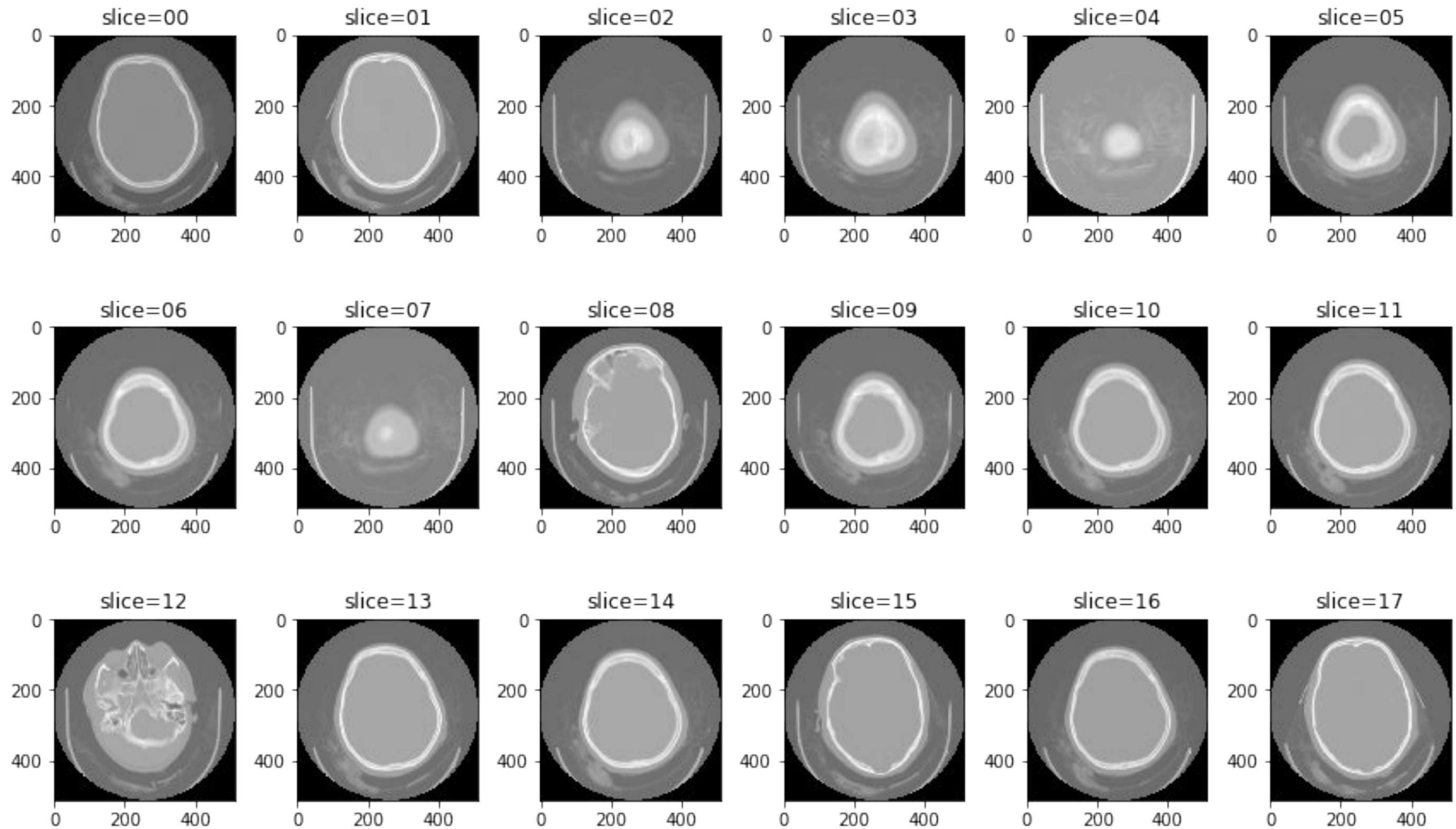
Dataset: Qure25k - Head CT Scans

- 491 scans, represented by 193,317 slices
- Annotated by three senior radiologists

Finding	Reader 1	Reader 2	Reader 3
Intracranial hemorrhage	✔	✔	✔
Intraparenchymal	✔	✔	✔
Intraventricular	✔	✔	✔
Subdural			
Extradural			
Subarachnoid	✔		✔
Location	Left, Right	Left, Right	Left, Right
Chronic			
Fracture			
Calvarial fracture			
Other fracture			
Midline Shift		✔	
Mass Effect	✔	✔	✔

Credit: http://headctstudy.qure.ai/#dataset

Head CT Scan Slices



In [4]:
```
reads.shape
```
Out[4]:
```
(491, 44)
```

In [3]:
```
reads.head()
```
Out[3]:

	name	Category	R1:ICH	R1:IPH	R1:IVH	R1:SDH	R1:EDH	R1:SAH	R1:BleedLocation-Left	R1:BleedLocation-Right	...	R3:EDH	R3:SAH	R3:BleedLocation-Left	R3:Bl
0	CQ500-CT-427	B2	1	1	0	0	0	0	0	1	...	0	0	1	
1	CQ500-CT-181	B2	1	1	0	1	0	1	1	0	...	1	1	1	
2	CQ500-CT-99	B1	0	0	0	0	0	0	0	0	...	0	0	0	
3	CQ500-CT-47	B1	0	0	0	0	0	0	0	0	...	0	0	0	
4	CQ500-CT-195	B1	0	0	0	0	0	0	0	0	...	0	0	0	

5 rows × 44 columns

In [5]:
```
reads.columns
```
Out[5]:
```
Index(['name', 'Category', 'R1:ICH', 'R1:IPH', 'R1:IVH', 'R1:SDH', 'R1:EDH',
       'R1:SAH', 'R1:BleedLocation-Left', 'R1:BleedLocation-Right',
       'R1:ChronicBleed', 'R1:Fracture', 'R1:CalvarialFracture',
       'R1:OtherFracture', 'R1:MassEffect', 'R1:MidlineShift', 'R2:ICH',
       'R2:IPH', 'R2:IVH', 'R2:SDH', 'R2:EDH', 'R2:SAH',
       'R2:BleedLocation-Left', 'R2:BleedLocation-Right', 'R2:ChronicBleed',
       'R2:Fracture', 'R2:CalvarialFracture', 'R2:OtherFracture',
       'R2:MassEffect', 'R2:MidlineShift', 'R3:ICH', 'R3:IPH', 'R3:IVH',
       'R3:SDH', 'R3:EDH', 'R3:SAH', 'R3:BleedLocation-Left',
       'R3:BleedLocation-Right', 'R3:ChronicBleed', 'R3:Fracture',
       'R3:CalvarialFracture', 'R3:OtherFracture', 'R3:MassEffect',
       'R3:MidlineShift'],
      dtype='object')
```

```
R1 agree with R2 (%): 0.5417515274949084
R2 agree with R3 (%): 0.5906313645621182
R1 agree with R3 (%): 0.5356415478615071
```

Dataset: PLAsTiCC Astronomical Classification

- This dataset contains simulated Time Series data of 7848 astronomical objects.
- We have access to the object's brightness as a function of time by measuring the photon flux in six different astronomical filters
- Total of 1.4M data points
- ~30 points per band per object
- Use these light curves to classify the variable sources into 15 classes