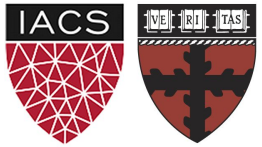


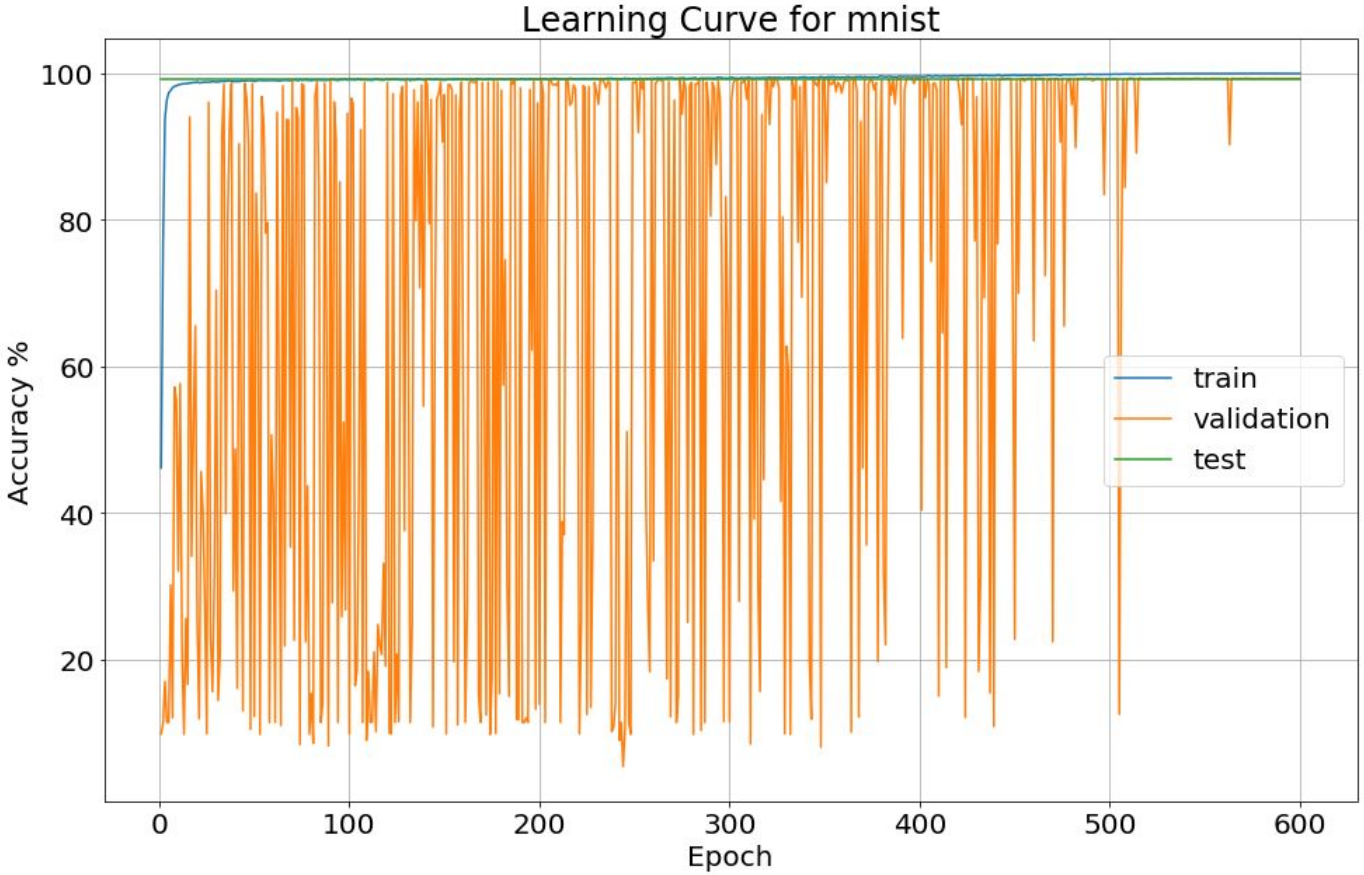
HARVARD
School of Engineering
and Applied Sciences

Neural Architecture Search

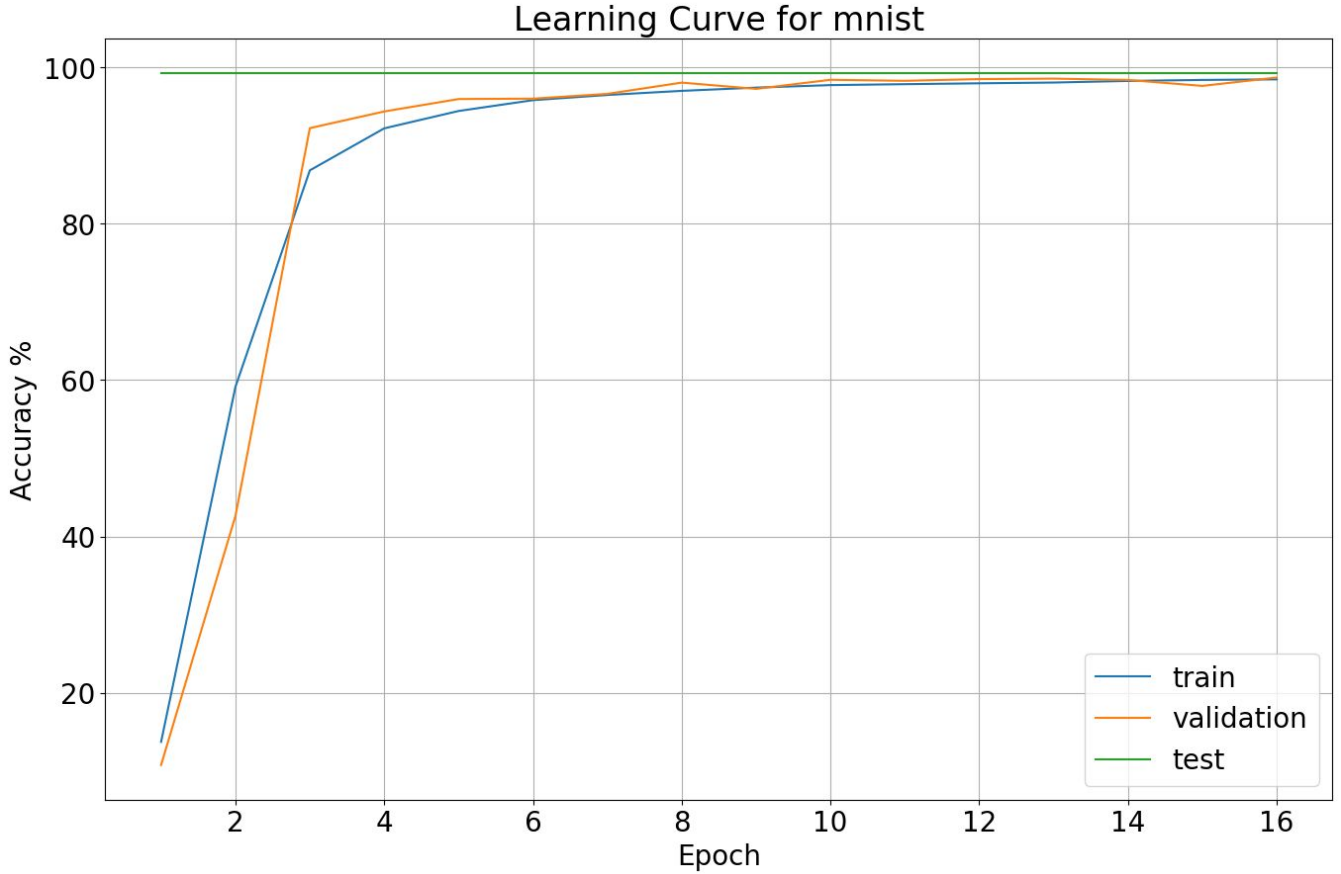
Harvard Data Science Capstone 2019
Milestone 3 Presentation

Team
Michael S. Emanuel
Julien Laasri
Dylan Randle
Jiawei Zhuang

Out-of-box DARTS crashes on MNIST



Tuning the learning rate is crucial



DARTS has a lot of traditional hyper-parameters

The parameters below are all fixed & chosen by human in the DARTS paper

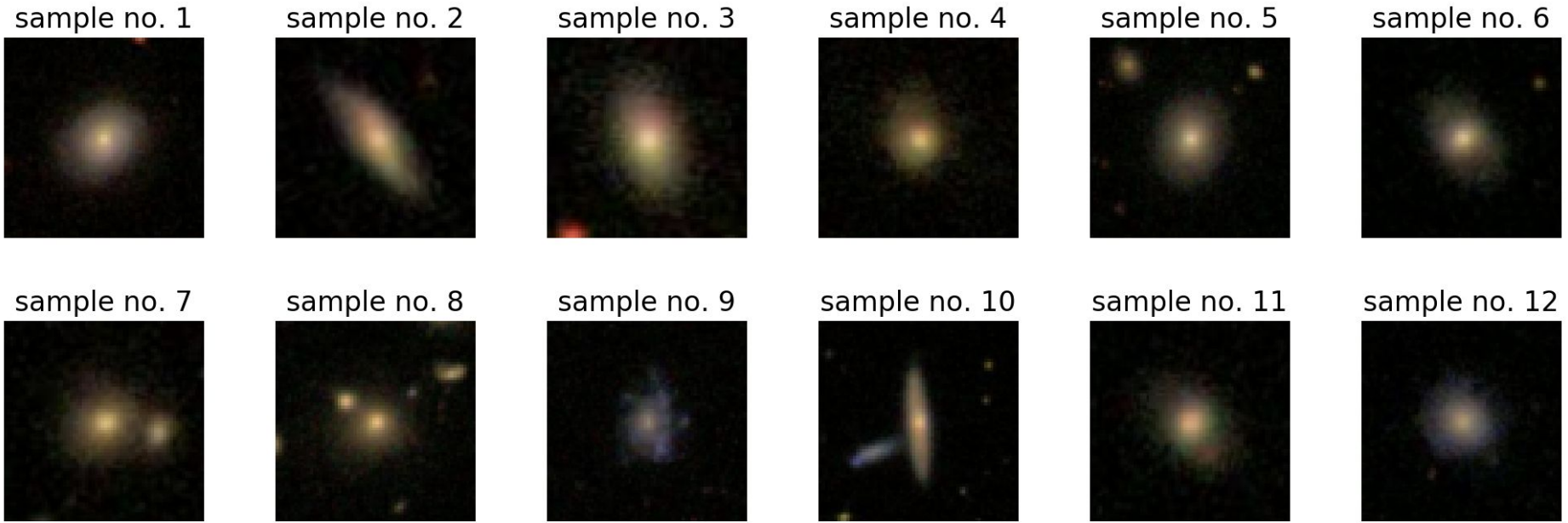
- Number of cells to stack together (8 for search and 20 for final model)
- Batch size for SGD (64 / 96)
- Number of channels (16 / 36 for first layer)
- Training epochs (50 / 600)
- Learning rates (0.025 for model weights w and 0.0003 for architecture weight α)

We find that learning rate has crucial impact when applying DARTS to other datasets.

This problem has been pointed out by many ICLR reviewers:
https://openreview.net/forum?id=S1eYHoC5FX&noteId=r1ekErZ53Q

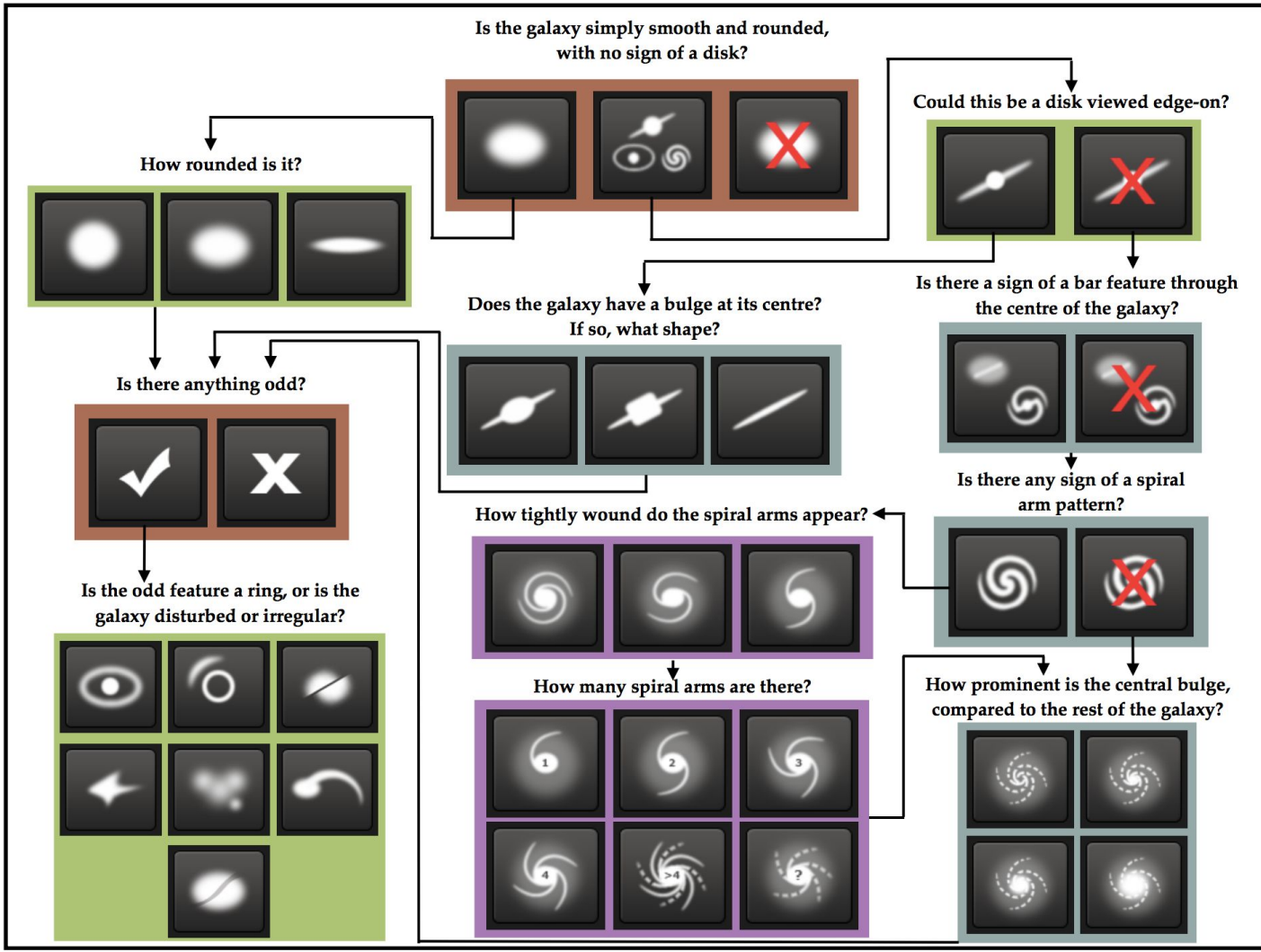
Galaxy Zoo data (2013 Kaggle competition)

- Input: RGB-channel image, 424 x 424 raw size, cropped and downsampled to 64 x 64
- Output label: a 37-element vector of probabilities, describing the galaxy type
- 61578 training samples, 79975 test samples



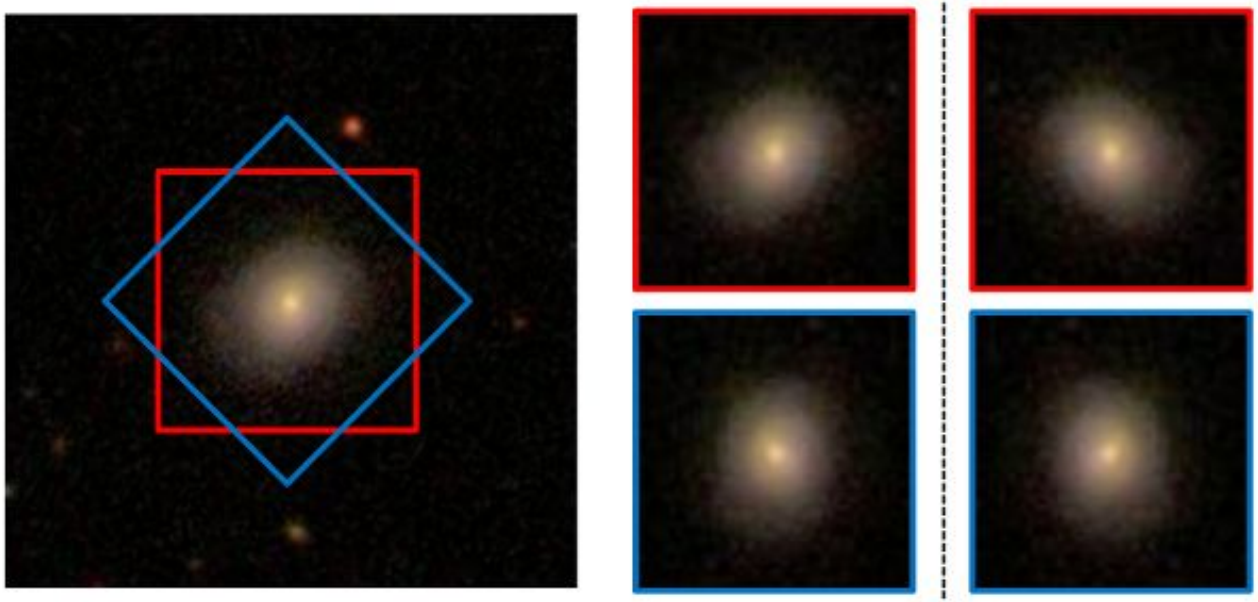
Galaxy Zoo "Decision Tree"

- 37 unique combinations of answers
- Data collected from crowdsourced questionnaire
- A single image was classified by multiple people, to get a probabilistic labeling instead of 0-1 labels



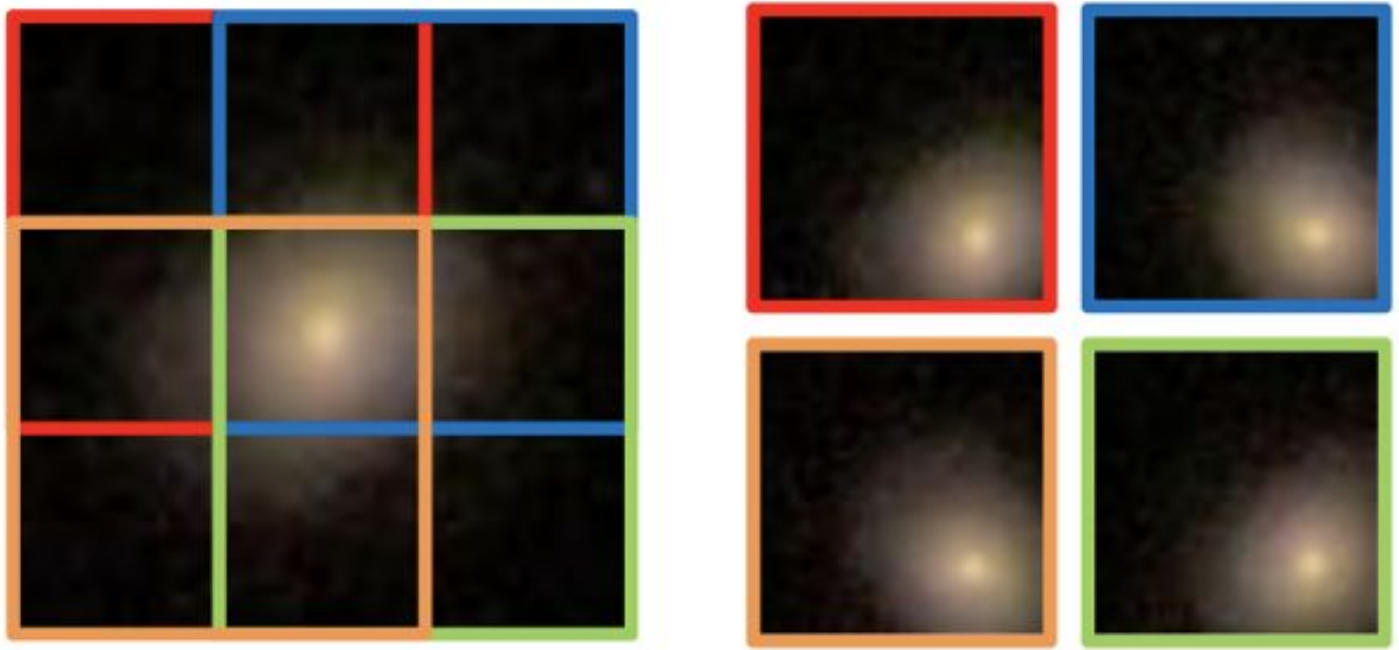
Data pre-processing and augmentation by Kaggle winner

- **Cropping:** Extract central part of the image, remove uninformative background
- **Rotation**: random with angle between 0° and 360° (uniform)
- **Translation**: random with shift between -4 and 4 pixels in x and y direction
- **Zoom**: random with scale factor between 1/1.3 and 1.3 (log-uniform)
- **Flip**: yes or no (bernoulli)



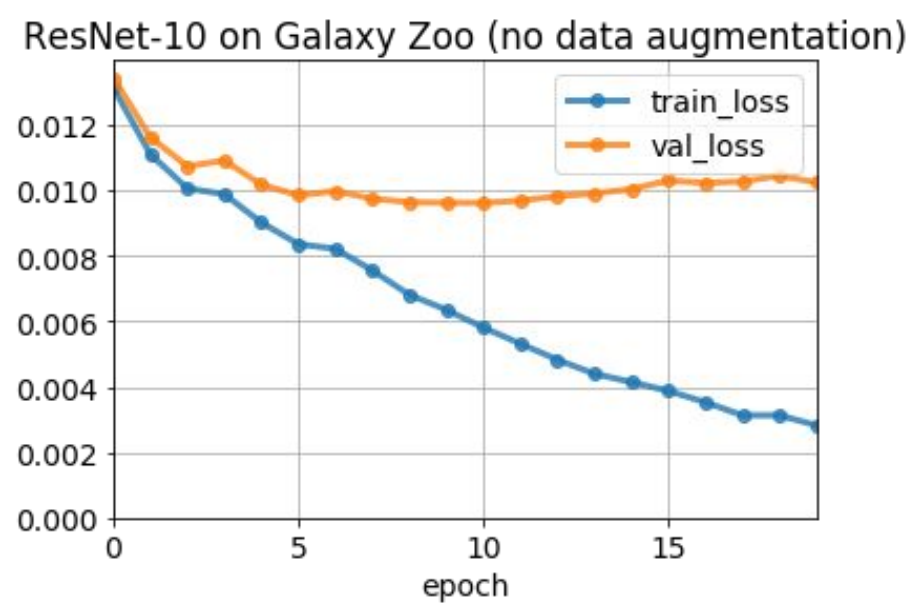
Further data augmentation by winning solution

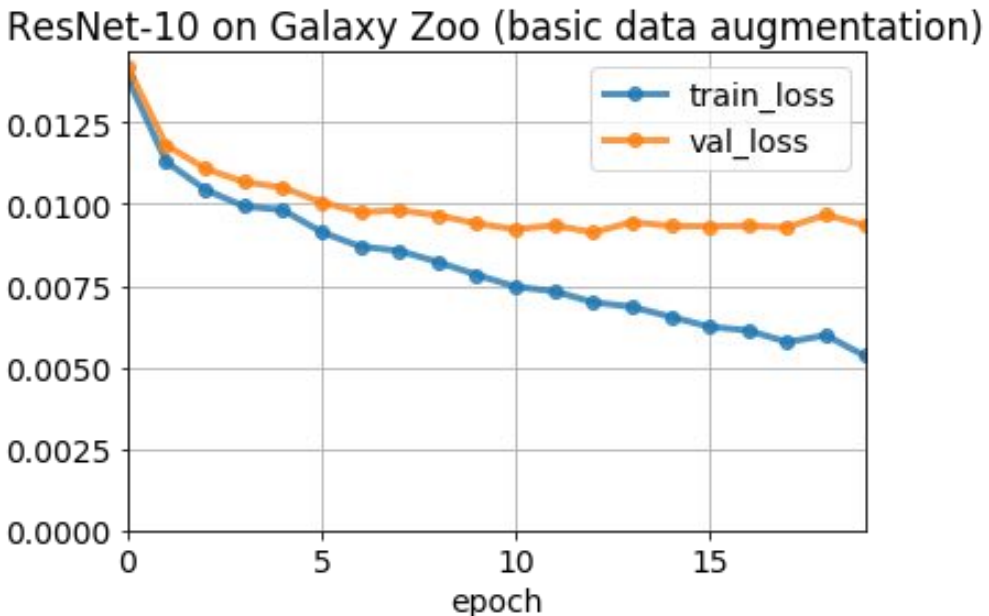
1. Extract four overlapping parts from the same image
2. Rotate each part so that the galaxy is in the bottom right corner.
3. Stack into the sample dimension of each mini-batch during training



Baseline ResNet model performance

- The out-of-box ResNet-18 severely overfits, so we halve the layers to 10
- Got validation RMSE of 0.101 (MSE loss ~ 0.0102) without any data augmentation
- Got validation RMSE of 0.096 (MSE loss ~ 0.0093) with basic data augmentation (random rotation and flipping)
- More data augmentation is needed to match Kaggle winner score (RMSE ~ 0.075)



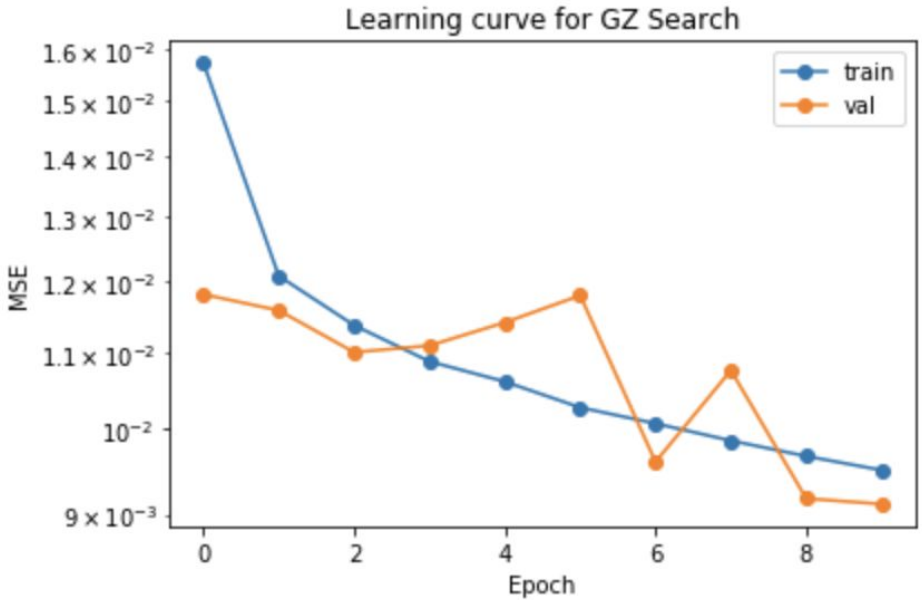


Galaxy Zoo Architecture with DARTS

- We began by reviewing the publicly posted code of the winner
- We expanded the list of primitive operations to include 2x2 max pooling
- We specialized the Network class to support the decision tree
  - Interestingly, it turned out that a naive regression treating all 37 outputs independently did as well or better; the winner used a tree with a hand rolled “divisive normalization” rather than softmax
- We added two fully connected layers at the end of the network
  - We searched with sizes of [1024, 1024]; the winner had 2 MaxOut layers of size 2048
- We started with simple data augmentation (random rotation and flip)
  - We later used a simplified version of the data augmentation tried by the winner
  - This turned out not to make much of a difference

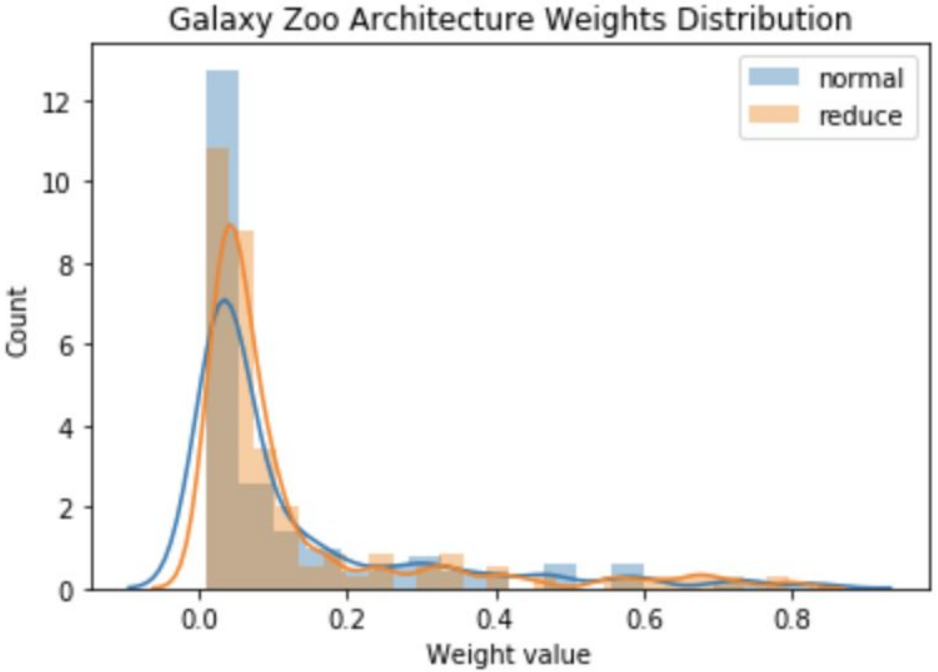
DARTS on Galaxy Zoo: Search

- Same basic data augmentation (no GZ tree)
- Performs comparable to ResNet

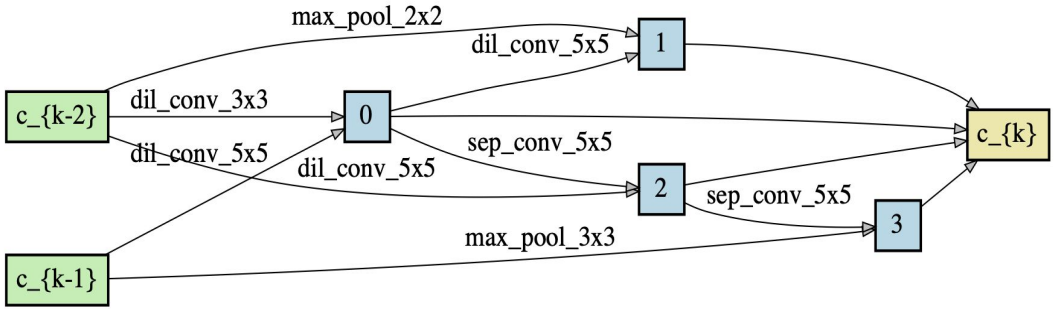


Learned Architecture is Sparse

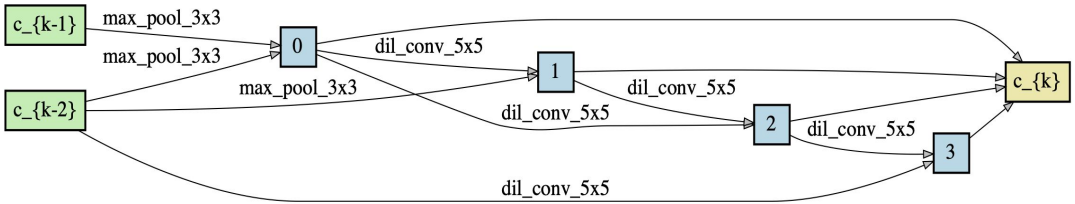
- Brings up **another** issue with DARTS: why do we take the maximal softmax score instead of using the full continuous relaxation cell?
- We significantly increased the architecture learning rate to 0.01 to encourage greater sparsity



Discretized Cells (Argmax of Softmax)

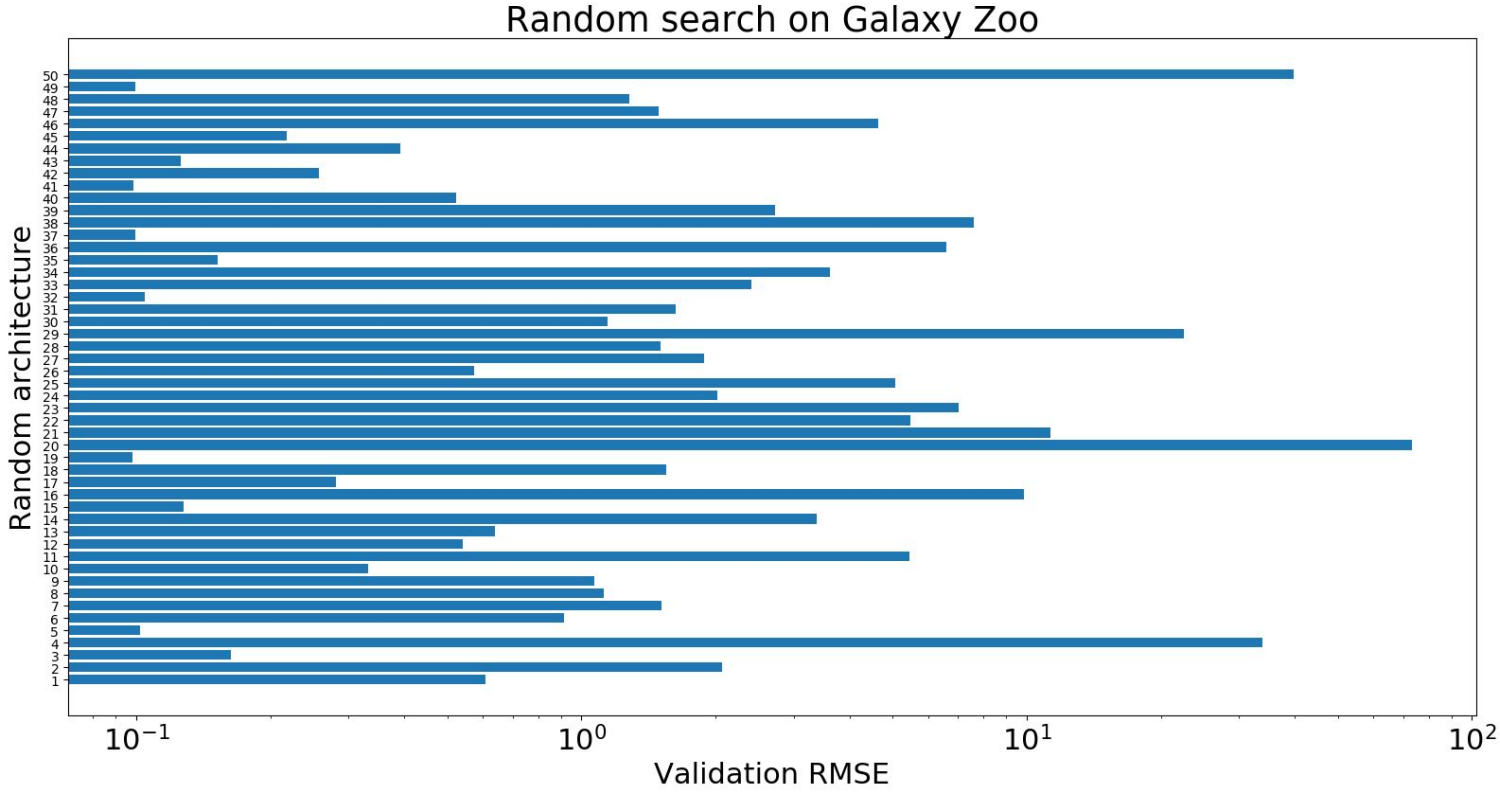


- Normal



- Reduction

Random search results



Mean RMSE: 5.388
Std RMSE: 12.428

DARTS on Galaxy Zoo: “Full” (discretized) Model

Models	RMSE
DARTS (Search)	0.094
DARTS (Discretized, Same)	0.114
DARTS (Discretized, Small)	0.101
Random Search	0.098
ResNet	0.095

- “Same” denotes that the model uses the same number of layers and channels as the DARTS Search model (non-discretized).
- “Small” denotes a model that uses half the number of layers and half the number of channels.

→ Poor performance of heuristic method for training “full” models

Galaxy Zoo Conclusion

These results highlight the problem with DARTS

1. There are too many hyperparameters to be “automated”
2. The notion of training a “search” model, discretizing the cell, then training a new “full” model
    - Does not work all the time (e.g. Galaxy Zoo)
    - Requires further tuning

Tentative Conclusion: A Tool for NAS, not “The Answer”

DARTS does not work out of the box on many problems

1. Traditional hyper-parameters (e.g. learning rate) must be chosen correctly —the case of MNIST
2. Dataset too simple and DARTS gives a model that is way bigger than needed—the case of Graphene
3. Problem limited by training data & generalizability instead of model capacity, where DARTS itself is not enough to get top score—the case of data augmentation for galaxy

Also, DARTS incorporates prior human knowledge learned from popular models like ResNet—the fact that identical cells are stacked sequentially, and the fact that each cell is a DAG with skip connections inside.

Possible Future Work

- Are there ways that DARTS could be extended to be a more general NAS? Here are a few ideas that might be tried in future work
- Automatic tuning of the learning rate during network search
  - We are already doing a “ping / pong” between training and validation to learn the architecture
  - This could be an ideal place in the algorithm to maintain statistics that could update the LR
- Opportunistic pruning rather than argmax followed by re-training?
  - We have no guarantees when or if the argmax architecture will work
  - Instead, we the search algorithm could try to eliminate primitives from the graph when their edge weights drop below a threshold
- L1 Regularization to encourage sparsity during architecture search