Optimizing Neural Networks with Gradient Lexicase Selection

Li Ding[1] (liding@umass.edu), **Lee Spector**[2,1] (lspector@amherst.edu)

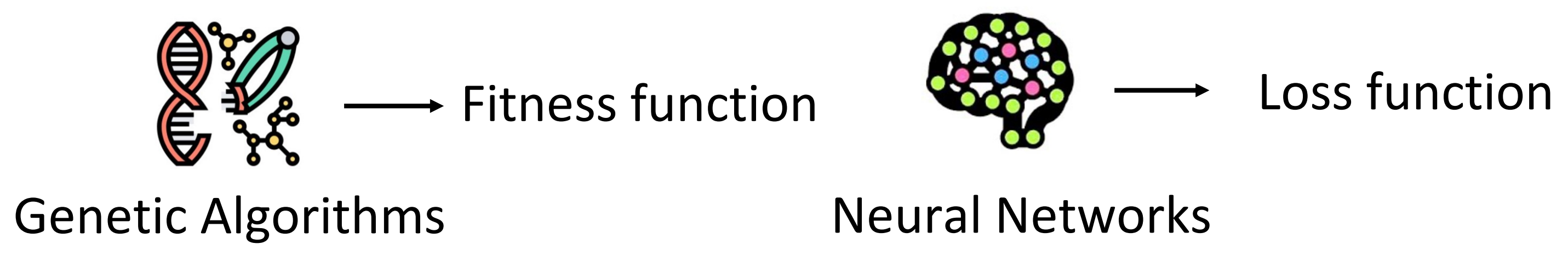
[1] Manning College of Information & Computer Sciences, University of Massachusetts Amherst

[2] Department of Computer Science, Amherst College

Aggregated Performance Measure

The potential drawback of seeking compromises.

Modern data-driven learning algorithms are usually optimized by computing the aggregate performance on the training data.



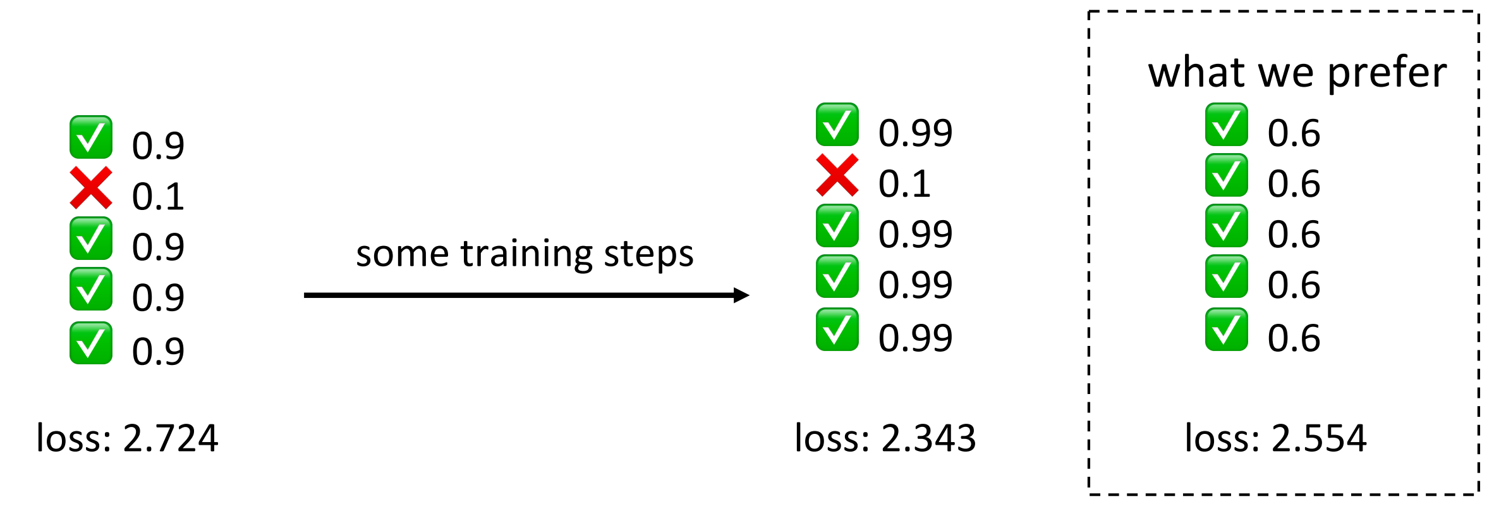
Fitness function

Genetic Algorithms

Loss function

Neural Networks

One potential drawback for aggregated performance measurement is that the model may learn to seek "compromises" and getting stuck at local optima.



✓ 0.9
✗ 0.1
✓ 0.9
✓ 0.9
✓ 0.9

loss: 2.724

some training steps

✓ 0.99
✗ 0.1
✓ 0.99
✓ 0.99
✓ 0.99

loss: 2.343

what we prefer

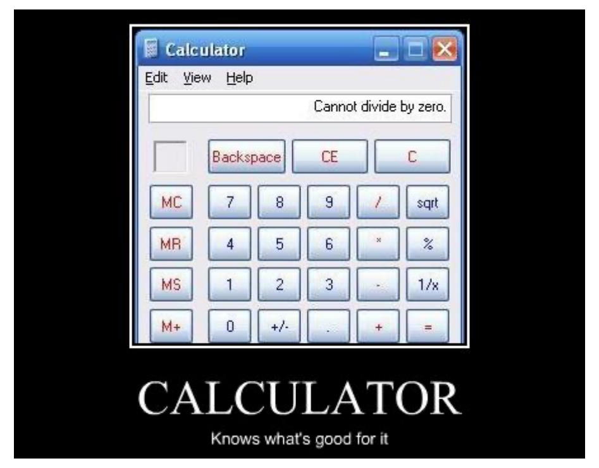
✓ 0.6
✓ 0.6
✓ 0.6
✓ 0.6
✓ 0.6

loss: 2.554

Lexicase Selection[1]

A method for uncompromising problems.

Uncompromising problems are problems for which it is not acceptable for a solution to perform sub-optimally on any of the cases in exchange for better performance on others.



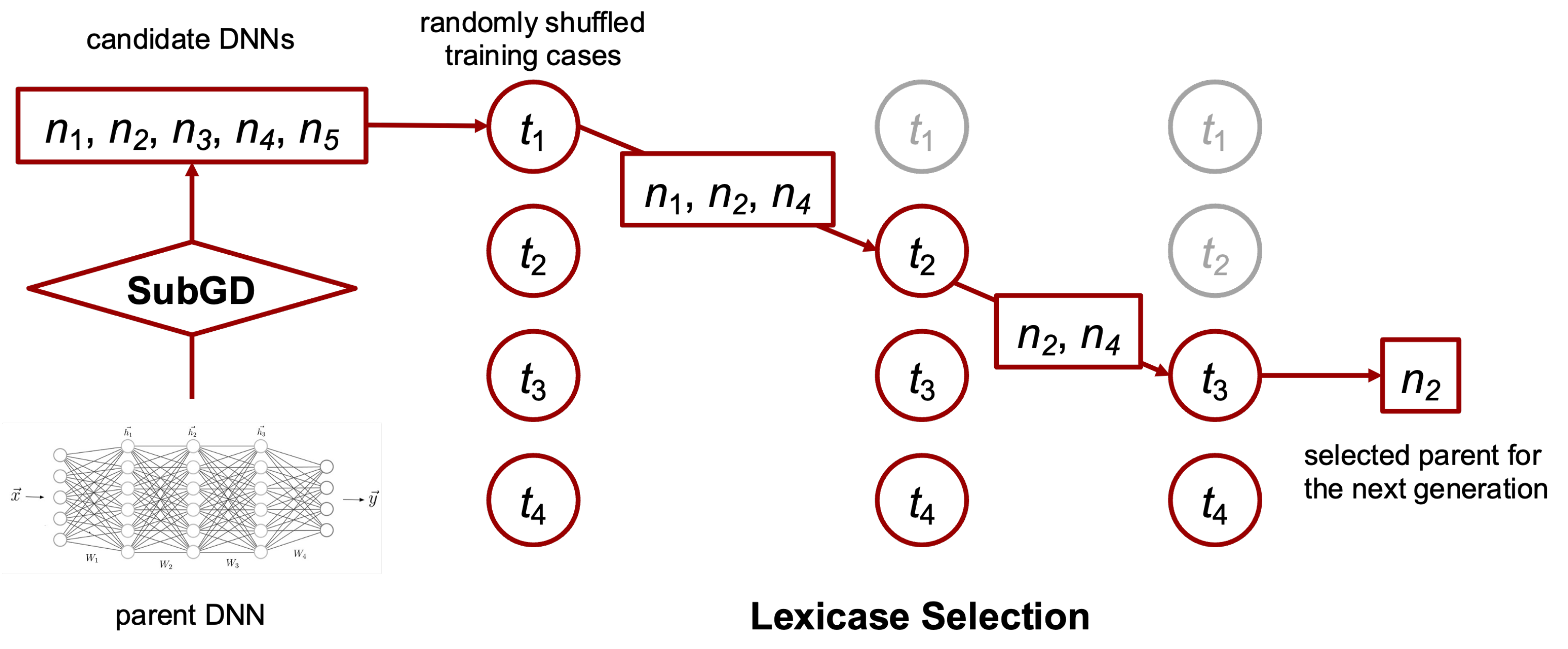
Such problems have been recently explored in genetic programming (GP) and genetic algorithms (GAs) for tasks such as program synthesis.

Instead of using aggregated fitness functions, **lexicase selection** gradually eliminates candidates by evaluating on each individual training case.

Lexicase selection has also been used in rule-based learning, symbolic regression, constraint satisfaction problems, machine learning, and evolutionary robotics to improve model generalization.

This Work: Gradient Lexicase Selection

Our method has two main components: subset gradient descent (SubGD) and lexicase selection.



candidate DNNs

randomly shuffled training cases

n_1, n_2, n_3, n_4, n_5

SubGD

parent DNN

t_1 t_2 t_3 t_4

n_1, n_2, n_4

t_1 t_2 t_3 t_4

n_2, n_4

t_1 t_2 t_3 t_4

n_2

selected parent for the next generation

Lexicase Selection

Overview of the proposed Gradient Lexicase Selection method

Mutation by Subset Gradient Descent

We propose a gradient-based mutation method: the training set is randomly divided into subsets. Each model instance is then trained on one of the subsets using stochastic gradient descent.

There are several advantages:

- All the candidates are trained with different non-overlapping training samples, so they are more likely to evolve diversely.
- Each candidate is trained using gradient descent for efficiency.
- Candidates can be trained in parallel to further reduce computation time.

Lexicase Selection for DNNs

After mutation, the offspring become candidates and we use lexicase selection to select a parent from them for the next generation.

First, a randomly shuffled sequence of training data points is used for selection. Starting from the first data point, we evaluate all the candidates on each case individually and remove the candidate from the selection pool if it does not make the correct prediction.

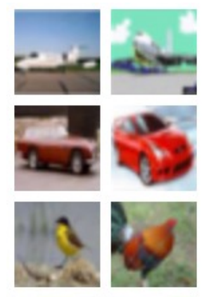
This process is repeated until if:

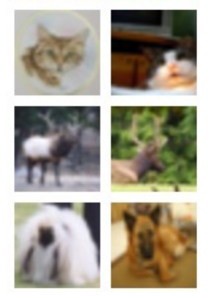
1) there is only one candidate left, which will be selected as the parent
2) all the training samples are exhausted, in which case we randomly pick a candidate from the remaining pool.

Experimental Results

We test our method with six popular DNN architectures on three image classification benchmarks. Our method outperforms SGD consistently under most of the settings.

Table 1: Image classification results. We report the mean percentage accuracy (*acc.*) with standard deviation (*std.*) obtained by running the same experiment with three different random seeds. The last column (*acc.* ↑) calculates the difference of accuracy by using our method compared to baseline, where positive numbers indicate improvement.

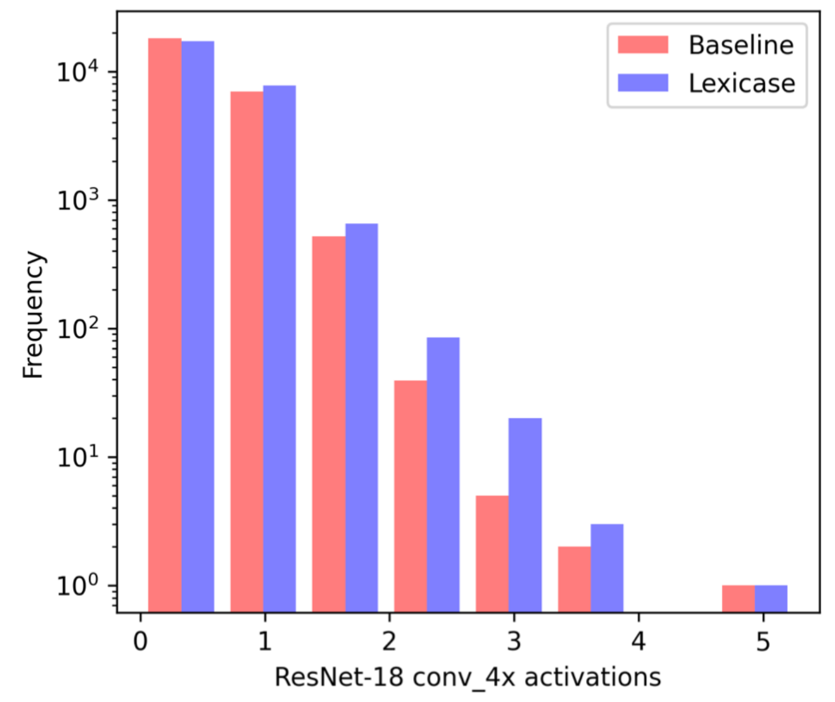


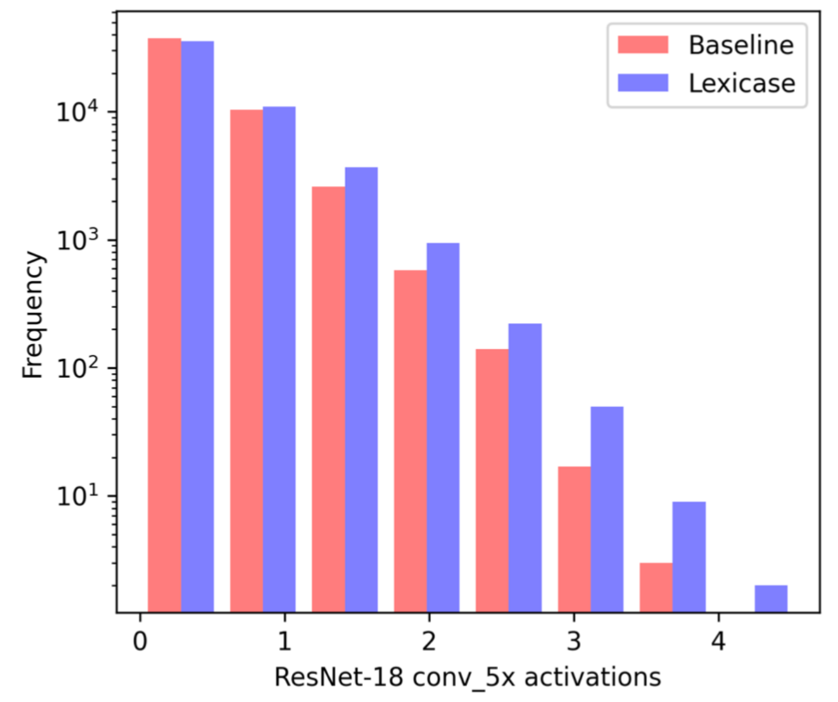


Dataset	Architecture	Baseline		Lexicase		
		acc.	*std.*	*acc.*	*std.*	*acc.* ↑
CIFAR-10	VGG16	92.85	0.10	93.40	0.13	**0.55**
	ResNet18	94.82	0.10	95.35	0.06	**0.53**
	ResNet50	94.63	0.46	94.98	0.18	**0.34**
	DenseNet121	95.06	0.31	95.38	0.04	**0.32**
	MobileNetV2	94.37	0.19	93.97	0.12	-0.39
	SENet18	94.69	0.14	95.37	0.23	**0.68**
	EfficientNetB0	92.60	0.18	93.00	0.22	**0.40**
CIFAR-100	VGG16	72.09	0.52	72.53	0.20	**0.44**
	ResNet18	76.33	0.29	76.68	0.40	**0.35**
	ResNet50	76.82	0.96	77.44	0.25	**0.63**
	DenseNet121	78.72	0.82	79.08	0.26	**0.36**
	MobileNetV2	75.87	0.28	75.57	0.30	-0.30
	SENet18	76.97	0.06	77.22	0.29	**0.25**
	EfficientNetB0	71.03	0.86	71.36	0.87	**0.33**
SVHN	VGG16	96.27	0.06	96.29	0.08	**0.02**
	ResNet18	96.43	0.14	96.62	0.08	**0.19**
	ResNet50	96.69	0.21	96.74	0.07	**0.04**
	DenseNet121	96.82	0.16	96.87	0.03	**0.05**
	MobileNetV2	96.23	0.13	96.26	0.07	**0.03**
	SENet18	96.62	0.19	96.59	0.11	-0.03
	EfficientNetB0	96.14	0.12	95.94	0.10	-0.20

Qualitative Analysis

We visualize the feature activations in ResNet-18 trained using normal SGD and the proposed gradient lexicase selection. Our method produce more diverse and normalized representations.





References

[1] Thomas Helmuth, Lee Spector, and James Matheson. "Solving uncompromising problems with lexicase selection". IEEE Transactions on Evolutionary Computation, 19(5):630–643, 2014.

This material is based upon work supported by the National Science Foundation under Grant No. 1617087. This work was performed using high performance computing equipment obtained under a grant from the Collaborative R&D Fund managed by the Massachusetts Technology Collaborative.

Link to the paper: