Efficient Activation Function Optimization through Surrogate Modeling

Garrett Bingham and Risto Miikkulainen
garrett@gjb.ai

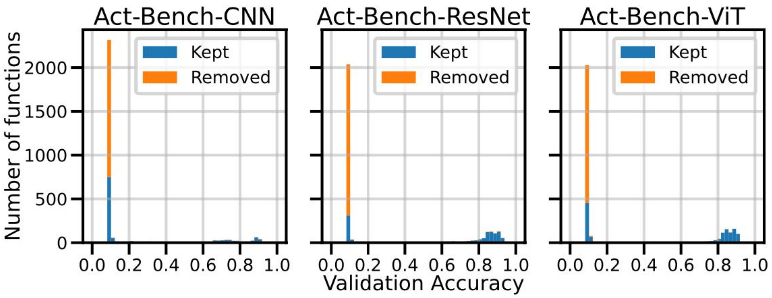
Google DeepMind
TEXAS The University of Texas at Austin
cognizant

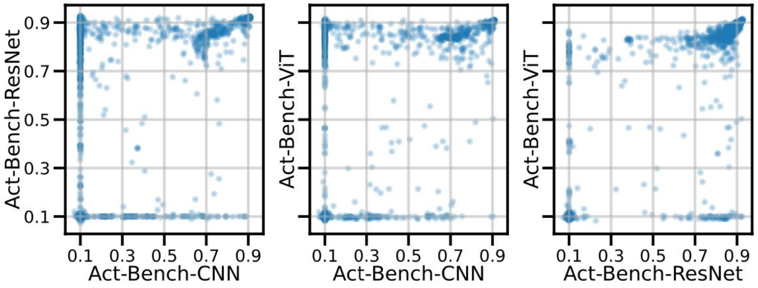
Components of AQuaSurF

- Benchmark Datasets (similar to NAS-Bench-101, etc.)
  - Precomputed results
  - Easy to experiment with different features and search algorithms
- Representation Learning
  - What features predict activation function performance?
- Surrogate and Search Algorithm Design
  - How can better activation functions be found efficiently?
- Improving Performance on Real-World Tasks
  - New datasets, architectures, and search spaces.

Benchmark Datasets

- 2,913 unique activation functions evaluated on three tasks
  - All-CNN-C on CIFAR-10 (Act-Bench-CNN)
  - ResNet-56 on CIFAR-10 (Act-Bench-ResNet)
  - MobileViTv2-0.5 on Imagenette (Act-Bench-ViT)





Features and Distance Metrics

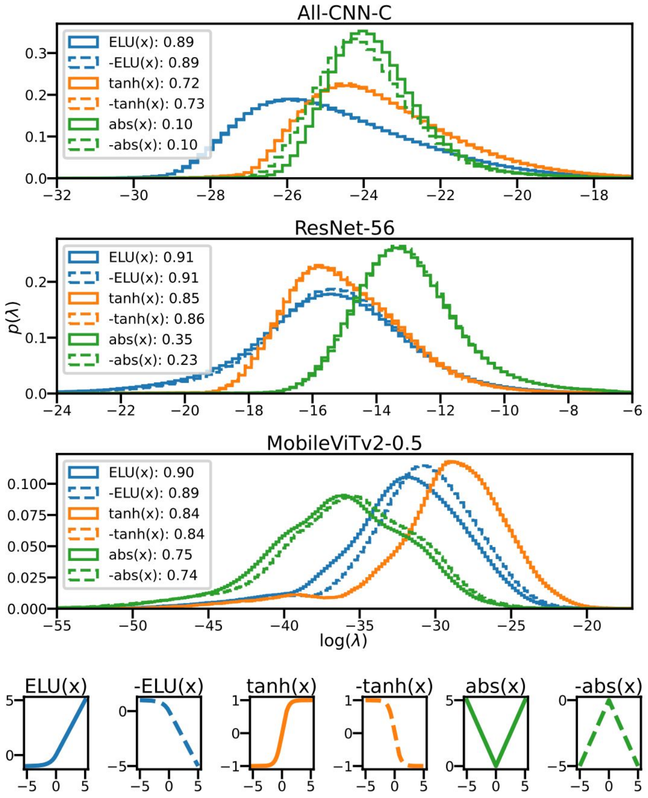
- Fisher information matrix (FIM) eigenvalues

$$\mathbf{F} = \mathop{\mathrm{E}}_{\substack{\mathbf{x}\sim Q_{\mathbf{x}} \\ \mathbf{y}\sim R_{\mathbf{y}|f(\mathbf{x};\boldsymbol{\theta})}}}\left[\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{y}, f(\mathbf{x};\boldsymbol{\theta}))\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{y}, f(\mathbf{x};\boldsymbol{\theta}))^\top\right]$$

$$d(f_\phi, f_\psi) = \sum_{l=1}^{L} \frac{W_1(\mu_l, \nu_l)}{w_l}$$

- Activation function outputs

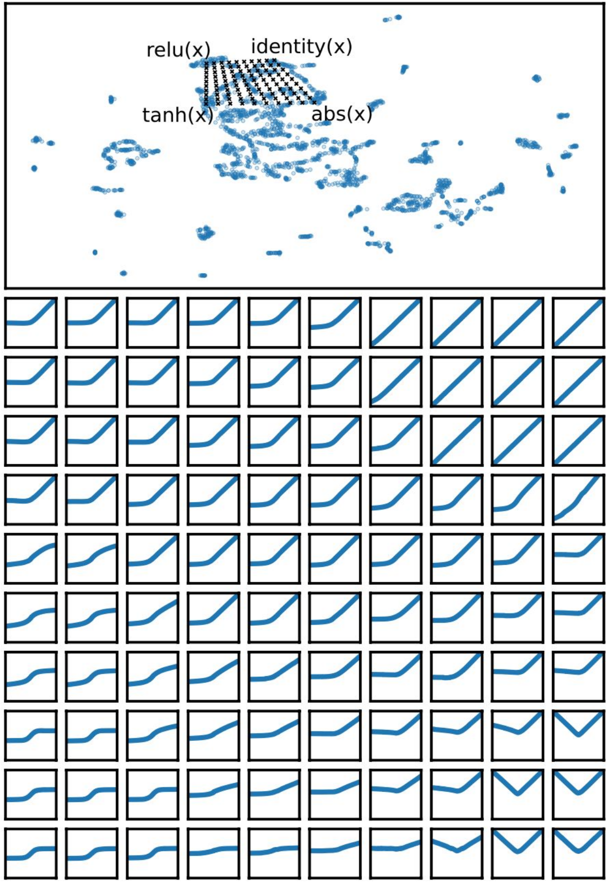
$$d(f_\phi, f_\psi) = \sqrt{\frac{\sum_{i=1}^{n}(\phi(x_i) - \psi(x_i))^2}{n}}, \quad x \sim \mathcal{N}(0,1).$$



Low-Dimensional Embedding

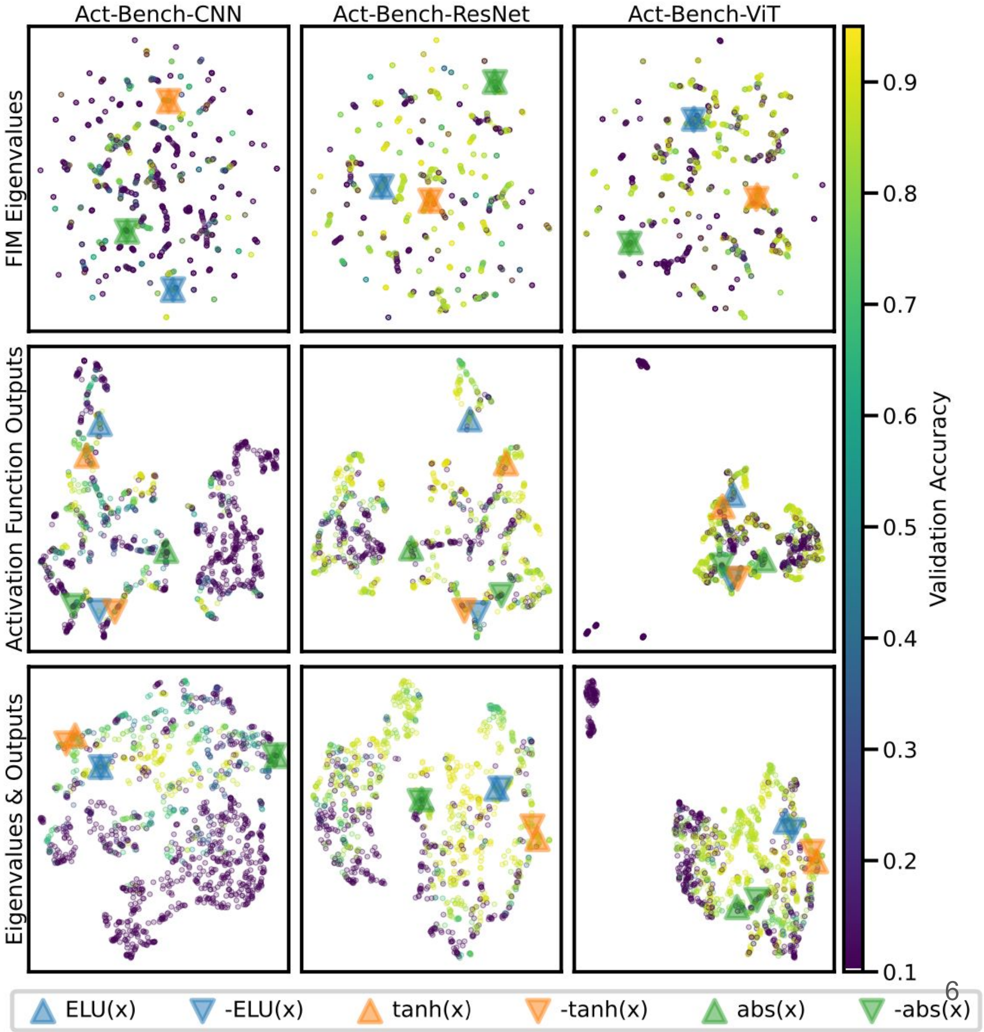
- The features and distance metrics are used to map activation functions to a low-dimensional embedding space

- UMAP (similar to t-SNE) learns an informative embedding and smoothly interpolates between activation functions



Learning a Surrogate

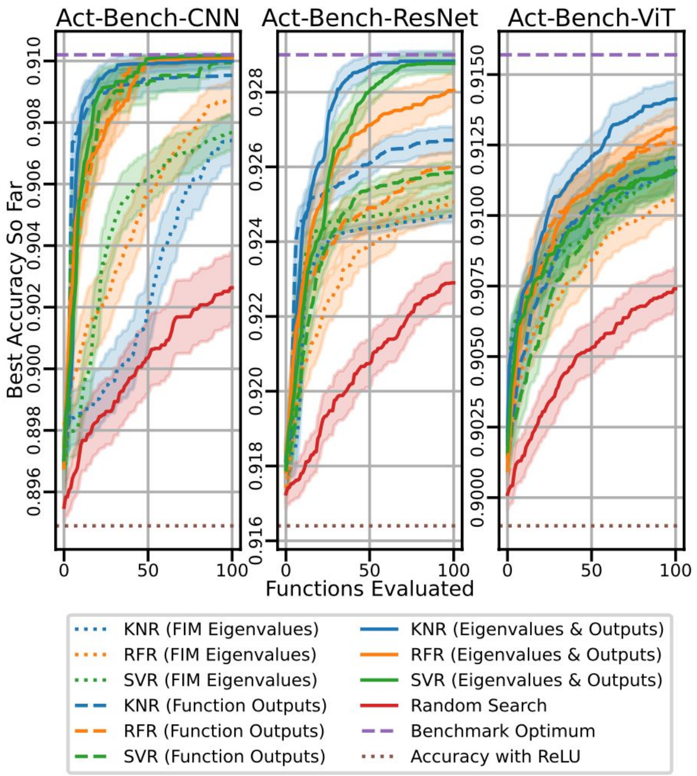
- Unsupervised UMAP embeddings show the predictive power of the features.
- Combining the two features (eigenvalues & outputs) provides the most powerful embedding



Searching on the Benchmark Tasks

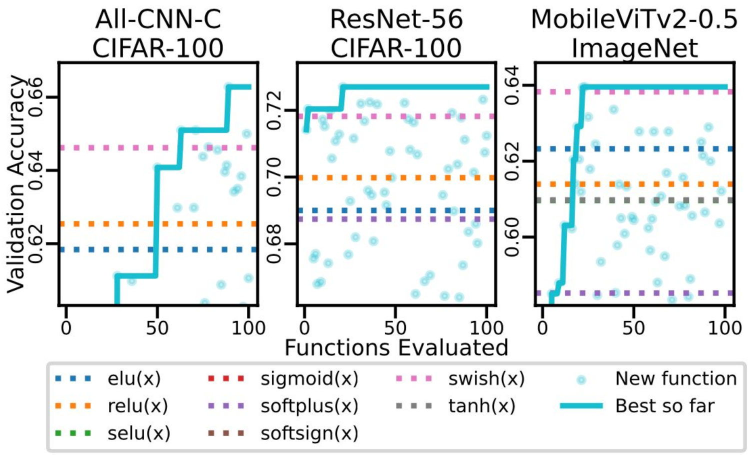
- Three algorithms were evaluated
  - Weighted k-nearest regression with k = 3 (KNR)
  - Random forest regression (RFR)
  - Support vector regression (SVR)

- The algorithms utilized three kinds of features
  - FIM eigenvalue
  - Activation function outputs
  - Both FIM eigenvalues and function outputs



Searching on New Tasks

- The best search algorithm (KNR) scales to more challenging problems
  - CIFAR-100 and ImageNet datasets
  - A larger search space with 425,896 unique functions



All-CNN-C on CIFAR-100		ResNet-56 on CIFAR-100		MobileViTv2-0.5 on ImageNet	
$\text{HardSigmoid}(\text{HardSigmoid}(x)) \cdot \text{ELU}(x)$	**0.6990**	$\text{Swish}(-2x)$	**0.7469**	$-x \cdot \sigma(x) \cdot \text{HardSigmoid}(x)$	**0.6396**
$\sigma(\text{Softsign}(x)) \cdot \text{ELU}(x)$	0.6950	$\text{SELU}(\sinh(e^{\arctan(x)} - 1))$	0.7458	$\text{ELU}(\text{Swish}(-x))$	0.6394
$\text{Swish}(x)/\text{SELU}(1)$	0.6931	$x \cdot \text{erfc}(\text{ELU}(x))$	0.7419	$\text{Swish}(x) \cdot \text{erfc}(\text{bessel\_i0e}(x))$	0.6336
ELU	0.6312	ELU	0.7411	ELU	0.6233
ReLU	0.6897	ReLU	0.7348	ReLU	0.6139
SELU	0.0100	SELU	0.6967	SELU	0.6096
sigmoid	0.0100	sigmoid	0.5766	sigmoid	0.5032
Softplus	0.6563	Softplus	0.7397	Softplus	0.5853
Softsign	0.2570	Softsign	0.6624	Softsign	0.5710
Swish	0.6913	Swish	0.7401	Swish	0.6383
tanh	0.3757	tanh	0.6754	tanh	0.6098

Transfer Across Tasks

- The best activation functions discovered in the three searches improve performance in a new task.

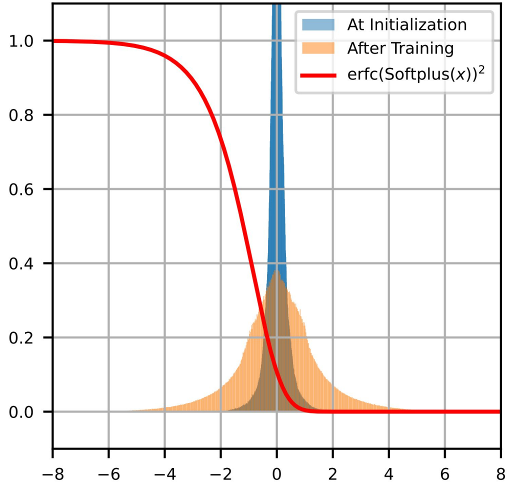
- ResNet-50 top-1 accuracy on ImageNet, median of three runs.

- Eight of the nine functions outperform ReLU.

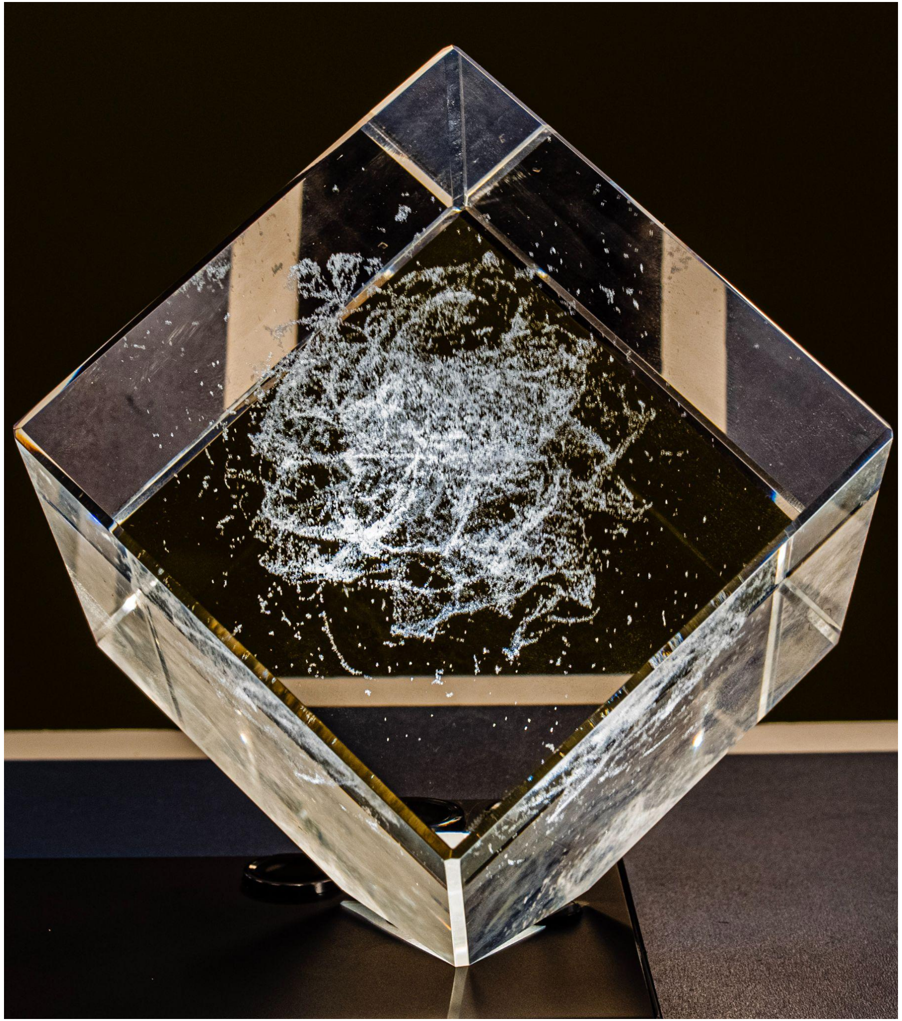
$-x \cdot \sigma(x) \cdot \text{HardSigmoid}(x)$	**0.7776**
$\text{Swish}(x)/\text{SELU}(1)$	0.7771
$\text{Swish}(x) \cdot \text{erfc}(\text{bessel\_i0e}(x))$	0.7755
$\sigma(\text{Softsign}(x)) \cdot \text{ELU}(x)$	0.7734
$\text{SELU}(\sinh(e^{\arctan(x)} - 1))$	0.7719
$\text{HardSigmoid}(\text{HardSigmoid}(x)) \cdot \text{ELU}(x)$	0.7718
$\text{ELU}(\text{Swish}(-x))$	0.7679
$\text{Swish}(-2x)$	0.7664
$x \cdot \text{erfc}(\text{ELU}(x))$	0.7635
$\text{ReLU}(x)$	0.7660

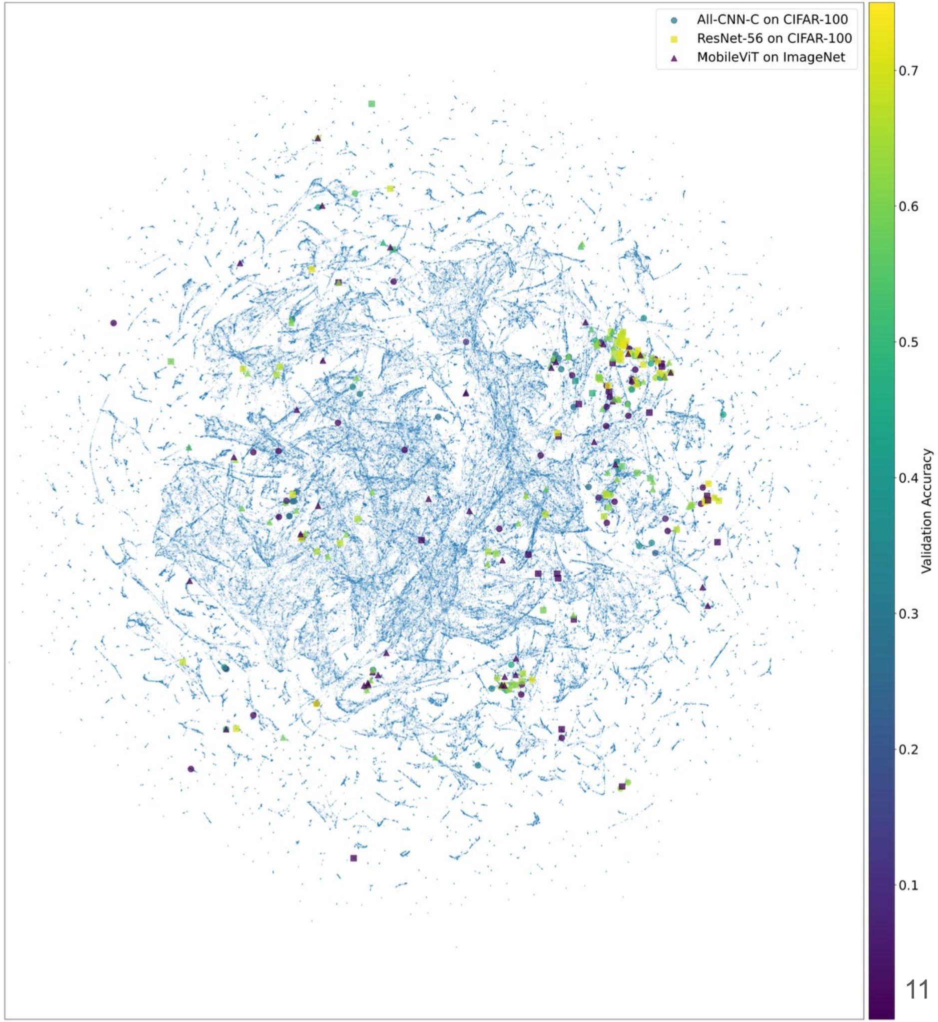
A Surprising Discovery for CoAtNet

- A sigmoidal design that outperformed all other activation functions was discovered.
- The network uses the function like a rectifier at initialization and like a sigmoidal function after training.
- The discovery challenges the status quo of always using rectifier nonlinearities in deep learning.



$\text{erfc}(\text{Softplus}(x))^2$	**0.8907**
$\min\{\text{Softplus}(x)^2, -x\}$	0.8861
$\text{arcsinh}(\text{ELU}(\text{Swish}(x)))$	0.8828
ELiSH	0.1000
ELU	0.8629
GELU	0.8841
HardSigmoid	0.8487
Leaky ReLU	0.8815
Mish	0.8762
ReLU	0.8772
SELU	0.8194
sigmoid	0.8586
Softplus	0.8678
Softsign	0.8530
Swish	0.8736
tanh	0.8415




All-CNN-C on CIFAR-100
ResNet-56 on CIFAR-100
MobileViT on ImageNet
Validation Accuracy
0.7
0.6
0.5
0.4
0.3
0.2
0.1