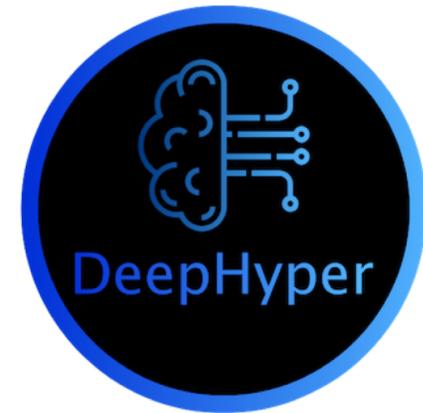

OCT 1, 2019

Argonne 
NATIONAL LABORATORY

DeepHyper: AutoML package for deep neural networks



Prasanna Balaprakash and Romit Maulik

Romain Egele, Misha Salim, Stefan Wild, Venkat Vishwanath

Mathematics and Computer Science Division & Leadership Computing Facility

Argonne National Laboratory

Team



Prasanna
Balaprakash,
MCS & ALCF



Romain Egele,
MCS



Misha Salim,
ALCF



Stefan Wild,
MCS



Venkat
Vishwanath,
ALCF

Contributors



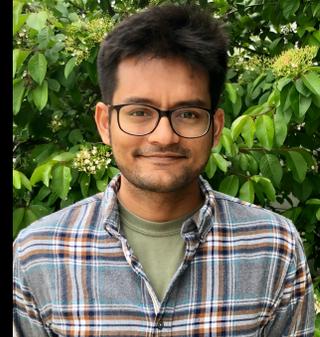
Taylor Childers,
ALCF



Tom Uram,
ALCF



Elise Jennings,
ALCF



Romit Maulick,
ALCF



Bethany Lusch,
ALCF



Epoch
001,644

Learning rate
0.03

Activation
ReLU

Regularization
None

Regularization rate
0

Problem type
Classification

DATA

Which dataset do you want to use?



Ratio of training to test data: 50%

Noise: 0

Batch size: 10

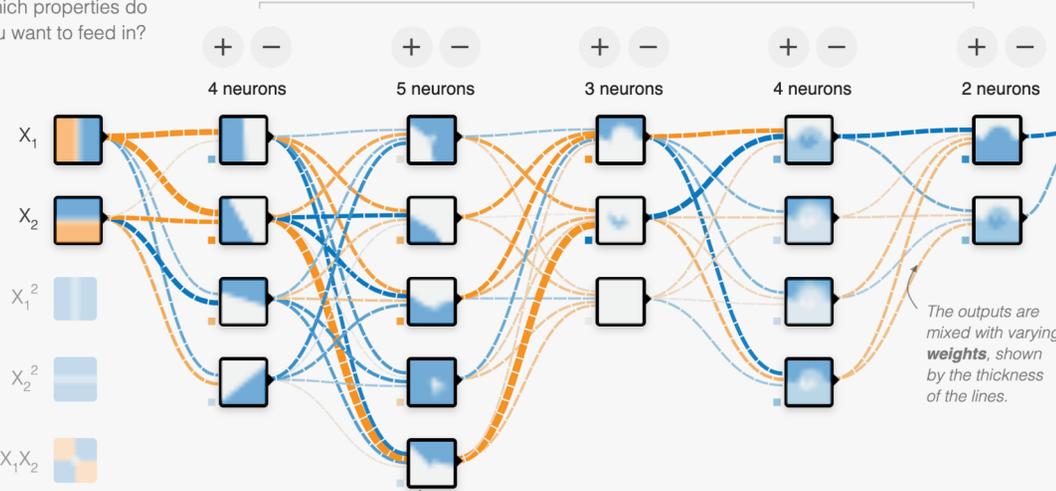
REGENERATE

FEATURES

Which properties do you want to feed in?

- X_1
- X_2
- X_1^2
- X_2^2
- $X_1 X_2$
- $\sin(X_1)$
- $\sin(X_2)$

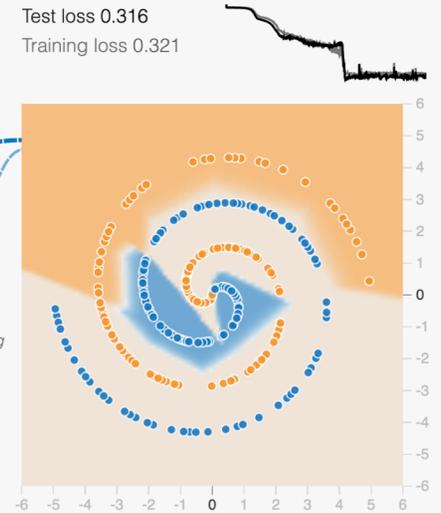
5 HIDDEN LAYERS



This is the output from one neuron. Hover to see it larger.

OUTPUT

Test loss 0.316
Training loss 0.321



The outputs are mixed with varying weights, shown by the thickness of the lines.

Colors shows data, neuron and weight values.

Show test data Discretize output



Epoch
001,142

Learning rate
0.03

Activation
ReLU

Regularization
None

Regularization rate
0

Problem type
Classification

DATA

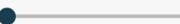
Which dataset do you want to use?



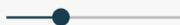
Ratio of training to test data: 50%



Noise: 0



Batch size: 10



REGENERATE

FEATURES

Which properties do you want to feed in?

X_1



X_2



X_1^2



X_2^2



$X_1 X_2$



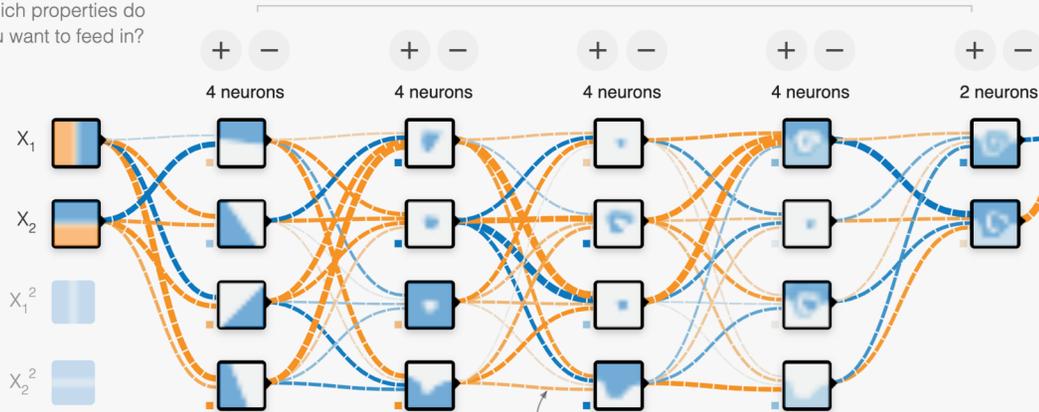
$\sin(X_1)$



$\sin(X_2)$



5 HIDDEN LAYERS

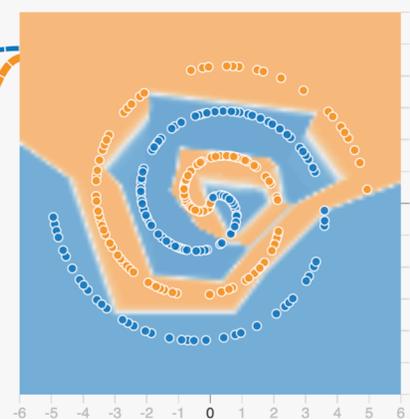
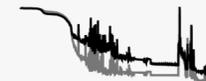


This is the output from one **neuron**. Hover to see it larger.

The outputs are mixed with varying **weights**, shown by the thickness of the lines.

OUTPUT

Test loss 0.063
Training loss 0.015



Colors shows data, neuron and weight values.

Show test data Discretize output



Epoch
001,442

Learning rate
0.03

Activation
ReLU

Regularization
None

Regularization rate
0

Problem type
Classification

DATA

Which dataset do you want to use?



Ratio of training to test data: 50%

Noise: 0

Batch size: 10

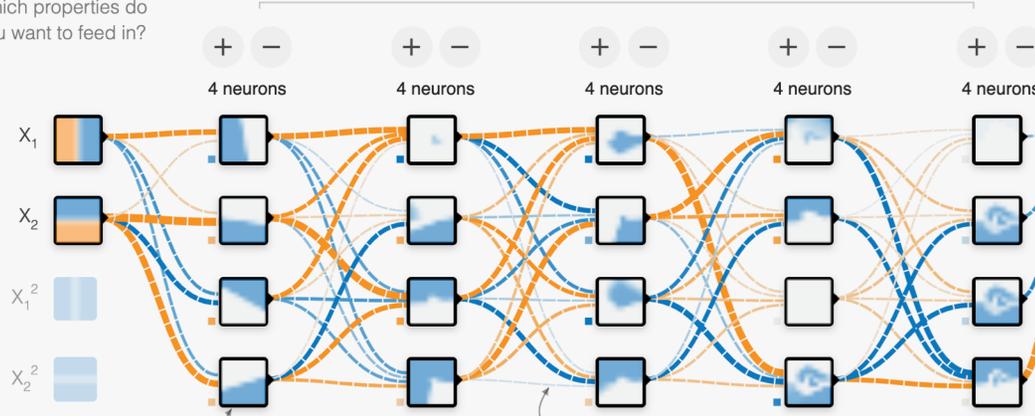
REGENERATE

FEATURES

Which properties do you want to feed in?

- X_1
- X_2
- X_1^2
- X_2^2
- $X_1 X_2$
- $\sin(X_1)$
- $\sin(X_2)$

5 HIDDEN LAYERS



This is the output from one **neuron**. Hover to see it larger.

The outputs are mixed with varying **weights**, shown by the thickness of the lines.

OUTPUT

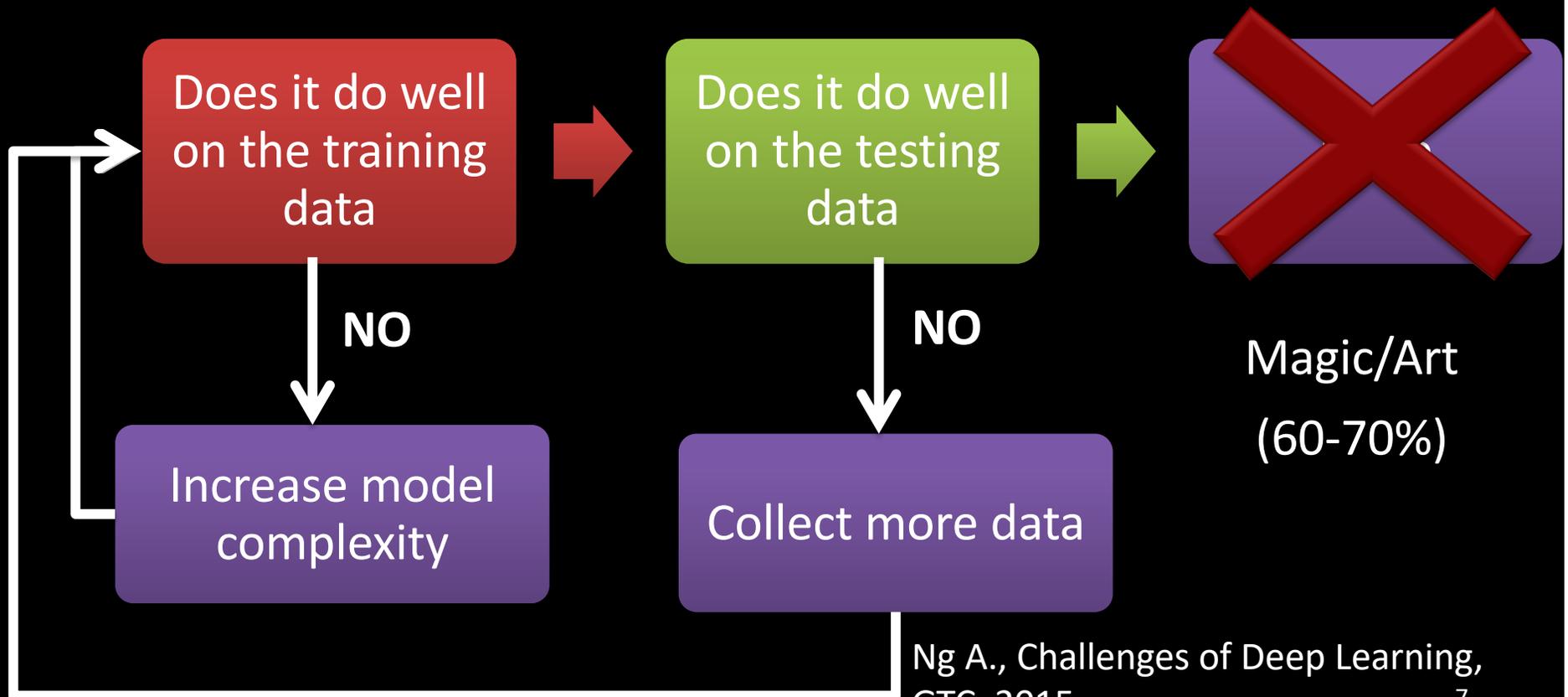
Test loss 0.239
Training loss 0.146



Colors shows data, neuron and weight values.

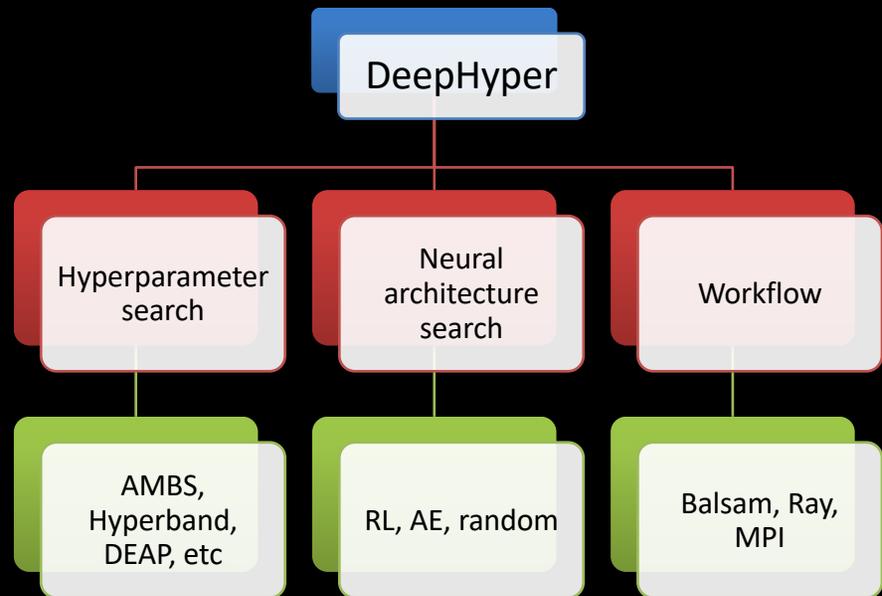
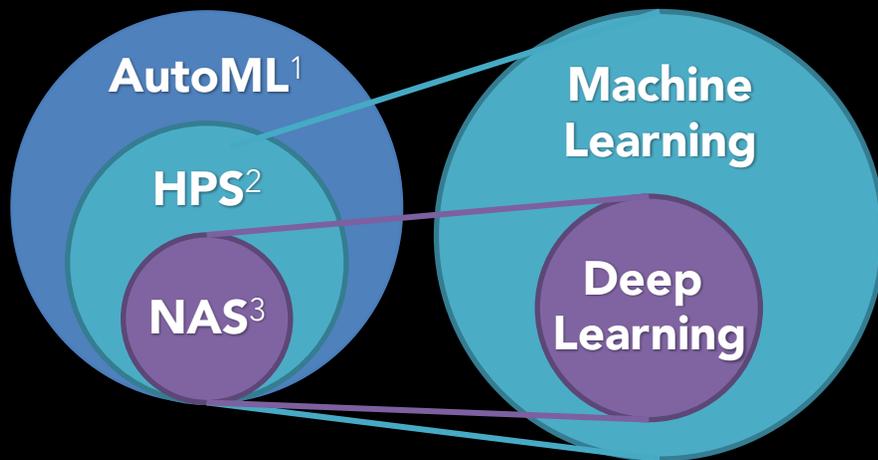
Show test data Discretize output

Deep neural networks





**A scalable automated machine learning (AutoML) package
for developing deep neural networks**



¹Automated Machine Learning
²Hyperparameter Search
³Neural Architecture Search

AutoML for deep learning

Lower-level problem: Training data

$$\text{solve} \quad \underset{w}{\text{minimize}} \quad \text{err}_T([\mathcal{X}_A, \mathcal{X}_P]; \mathcal{T}; w)$$

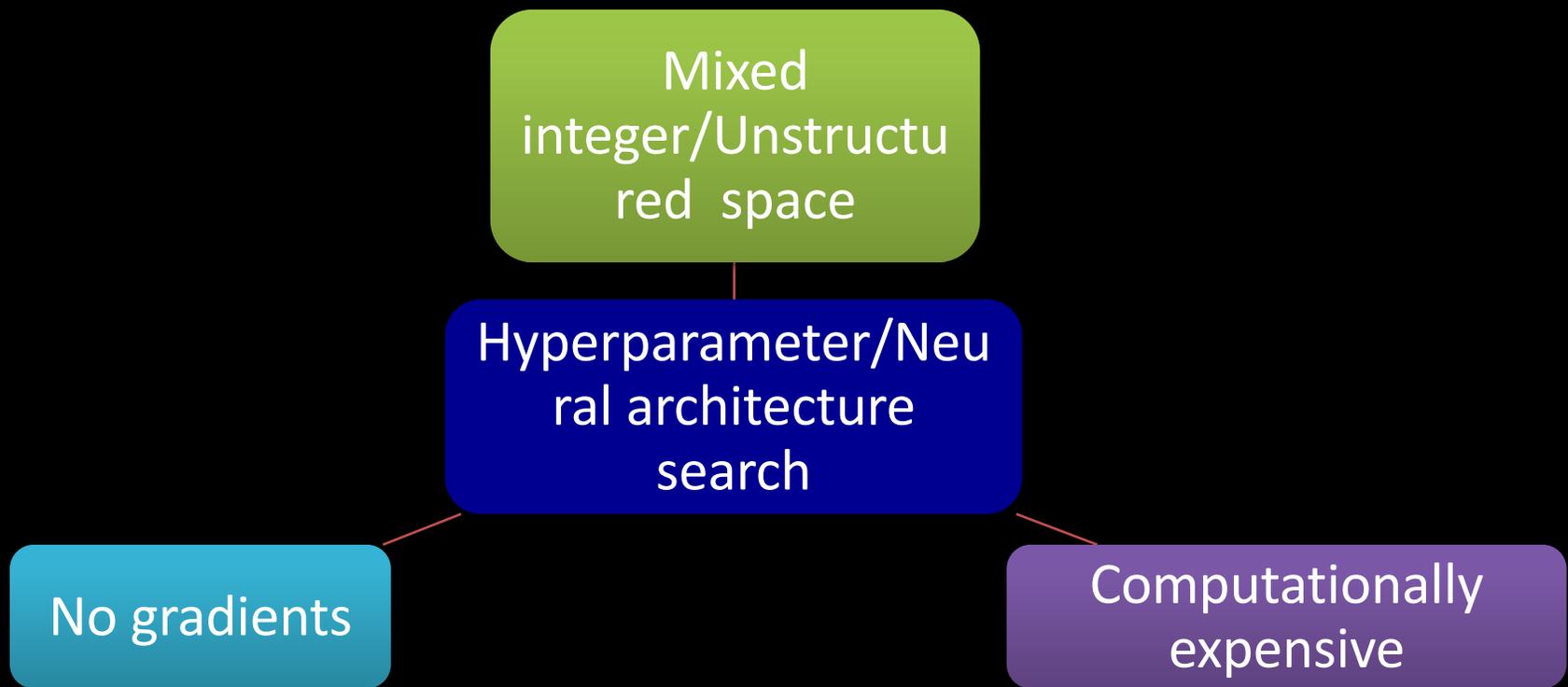
Upper-level problem: Validation data

$$\text{solve} \quad \underset{\mathcal{X}_A, \mathcal{X}_P}{\text{minimize}} \quad \text{err}_V([\mathcal{X}_A, \mathcal{X}_P]; \mathcal{V}; w^*[\mathcal{X}_A, \mathcal{X}_P])$$

Architecture space Hyperparameter space



Challenges



Grid search and random search are not the way to go!

Hyperparameter search

Example hyperparameters

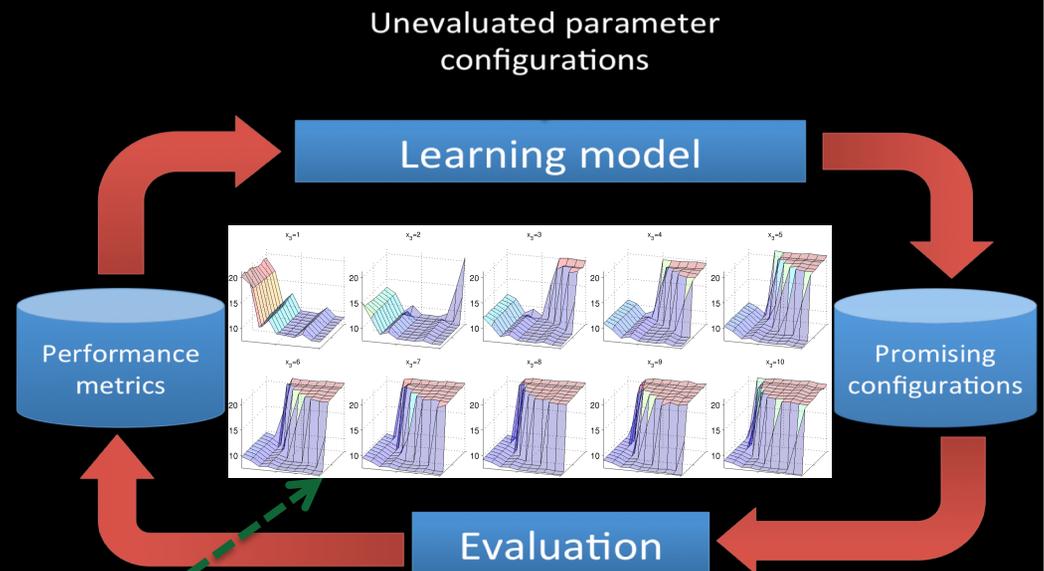
```
# Optimizer common hyperparameters
learning_rate = param.continuous("learning_rate", 1e-06, 1, prior=prior.LOGUNIFORM)
clipnorm = param.continuous('clipnorm', 1e-04, 1)
clipvalue = param.continuous('clipvalue', 1e-04, 1)
epsilon = param.continuous("epsilon", 1e-20, 1, prior=prior.LOGUNIFORM)
decay = param.continuous("decay", 0, 1)
beta_1 = param.continuous("beta_1", 0, 1 - 1e-06)
beta_2 = param.continuous("beta_2", 0, 1 - 1e-08)
rho_momentum = param.continuous("rho_momentum", 0, 1)
```

```
space = [
    param.discrete('batch_size', 8, 512, step.GEOMETRIC, 2),
    param.discrete('epochs', 5, 60, step.ARITHMETIC, 1),
    param.discrete('nunits', 256, 1024, step.ARITHMETIC, 1),
    activation,
    dropout,
    optimizer,
    set_clipnorm,
    set_clipvalue
]
```

AMBS: Asynchronous model-based search

– Framework:

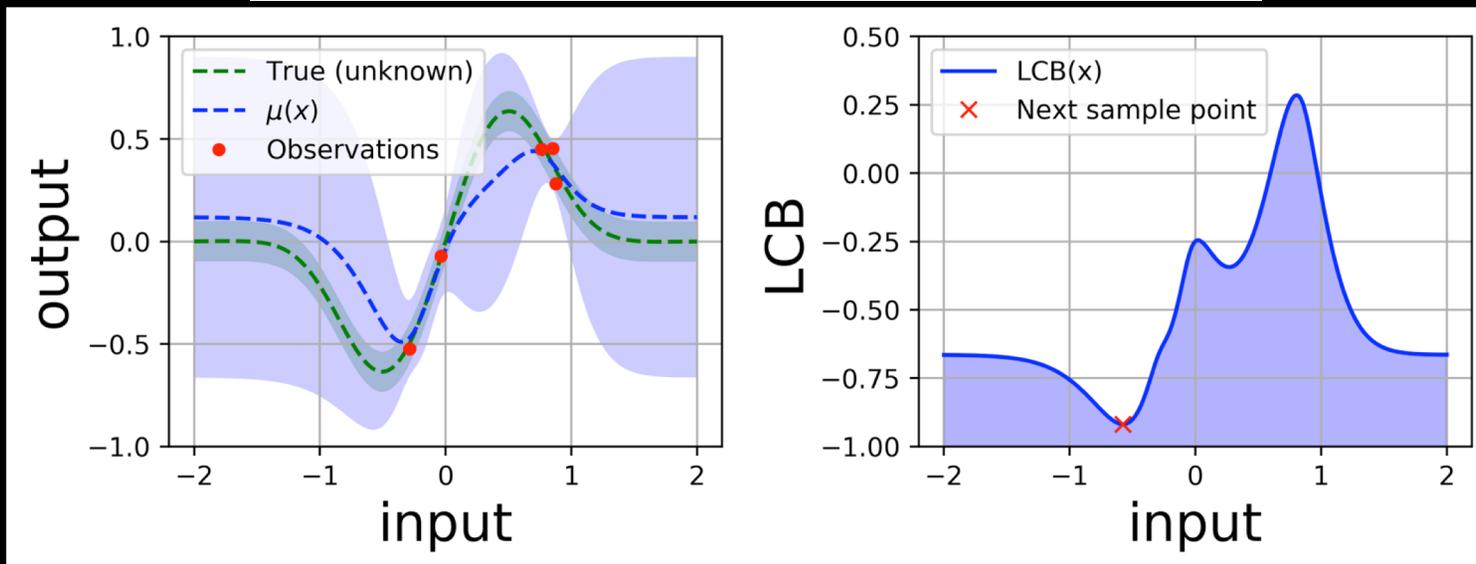
- Initialization phase
 - Random or Latin hypercube sampling
- Iterative phase
 - Fit model
 - Sample using the model



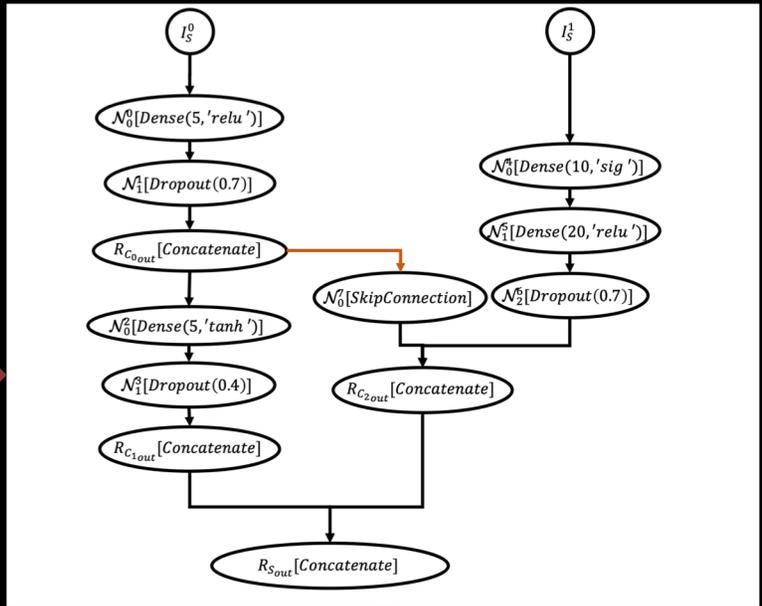
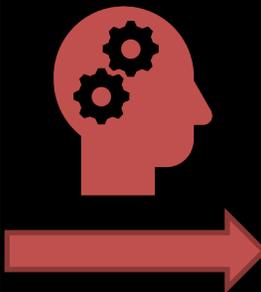
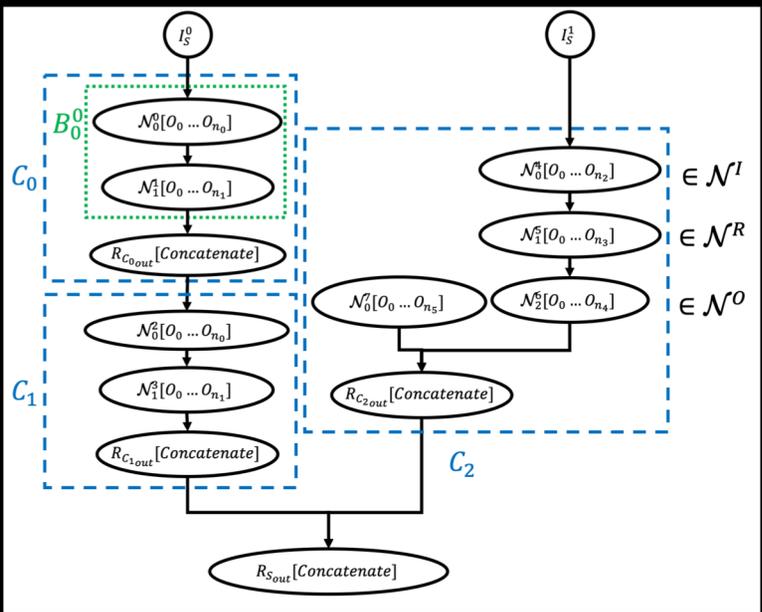
*Example Surrogate Model Fitted to Sampled Performance
(iterative refinement improves the learning model)*

Bayesian optimization

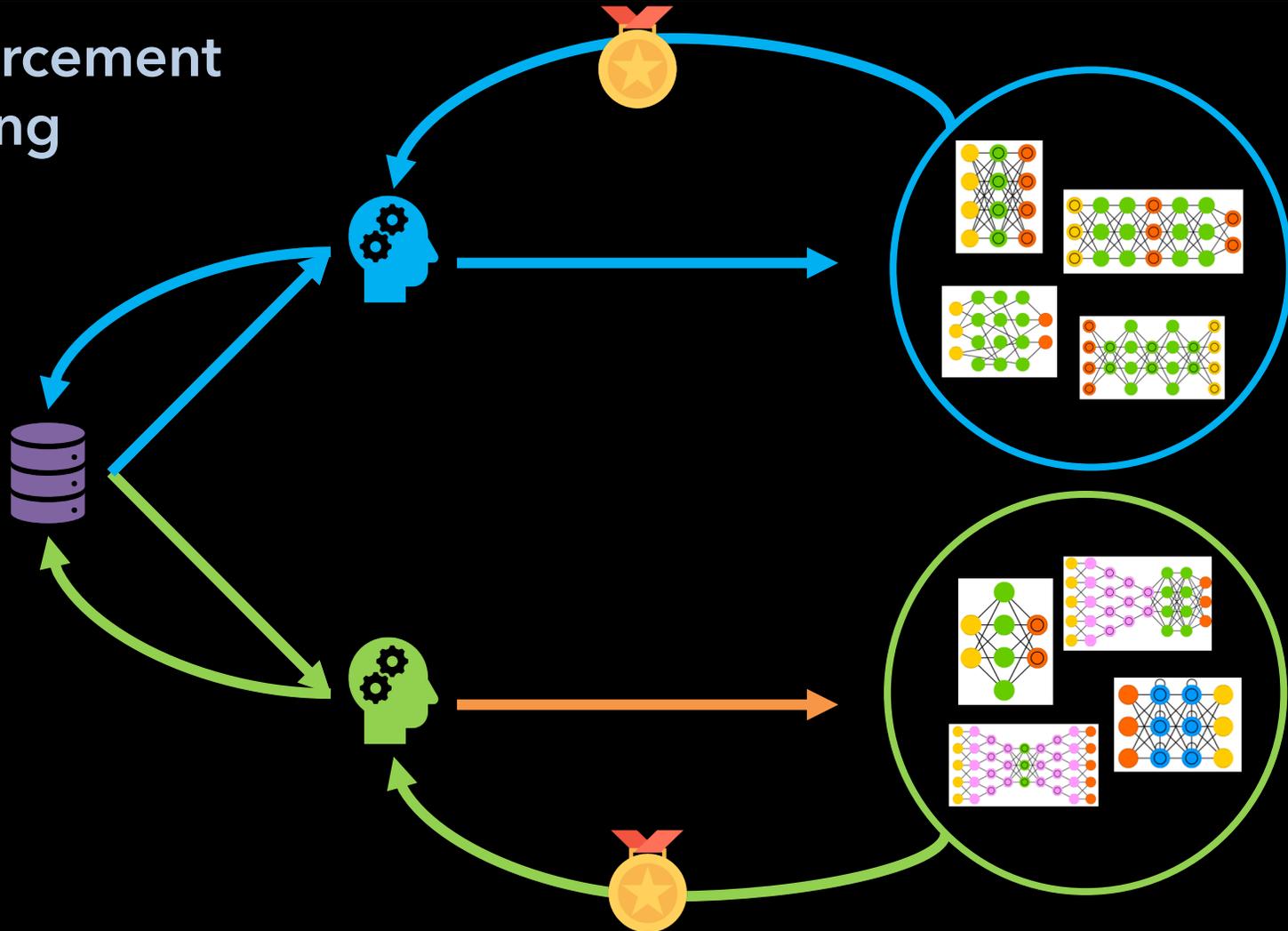
$$LCB(x, \beta) = \mu(x) - \beta \times \sigma(x)$$



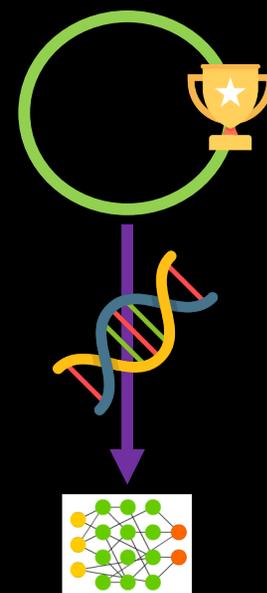
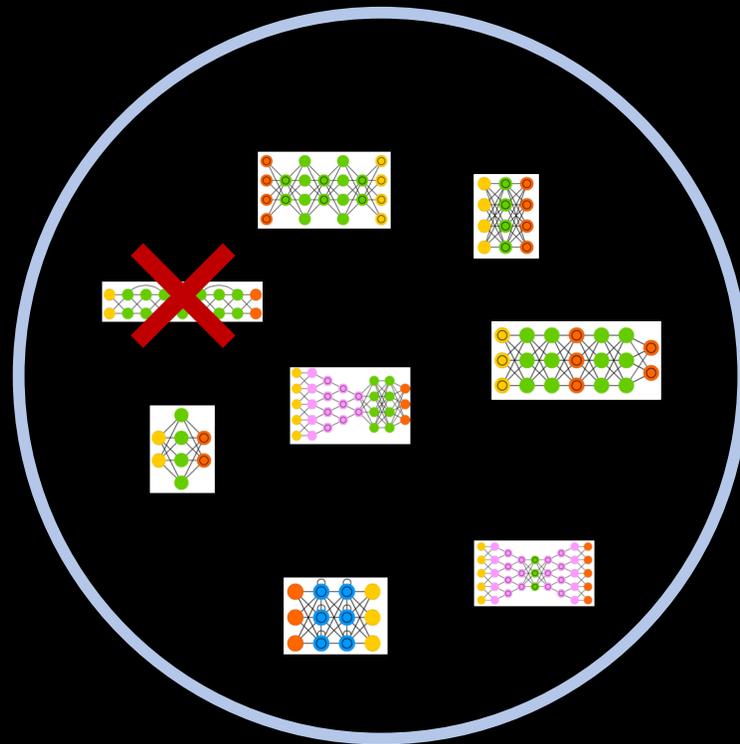
Neural architecture search



Reinforcement Learning



Aging Evolution



AutoML for deep learning

Lower-level problem: Training data

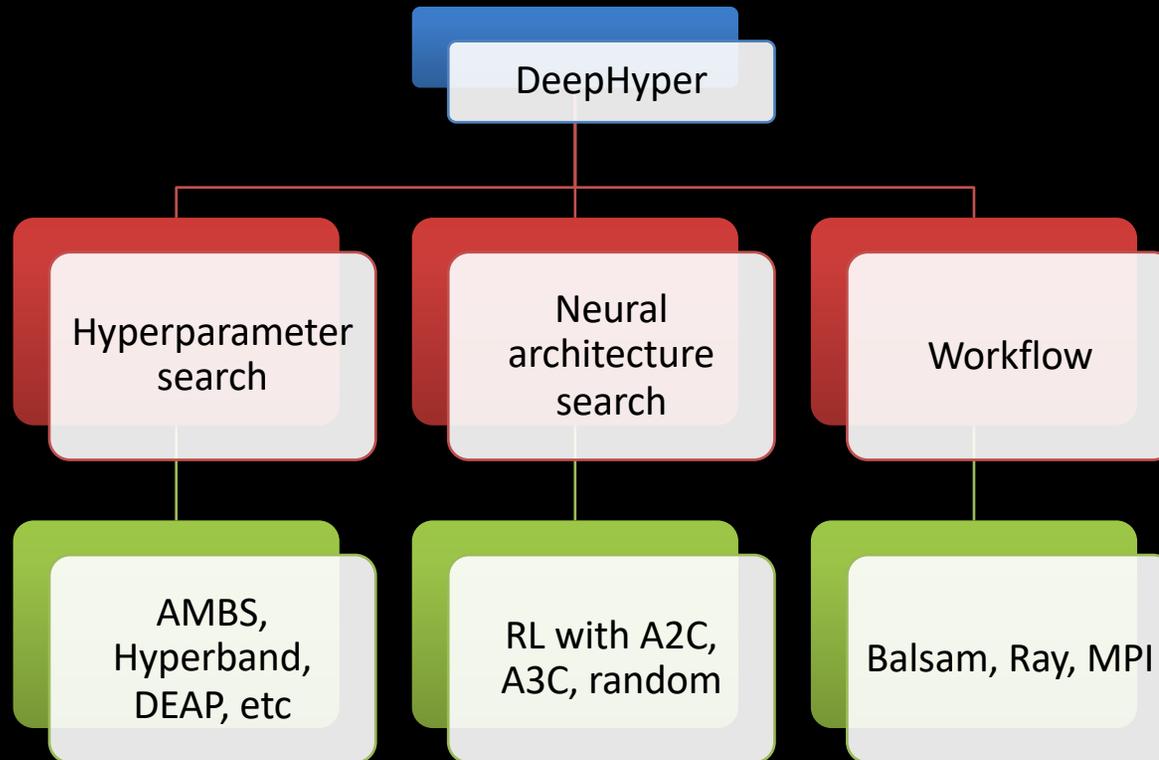
$$\text{solve} \quad \underset{w}{\text{minimize}} \quad \text{err}_T([\mathcal{X}_A, \mathcal{X}_P]; \mathcal{T}; w)$$

Upper-level problem: Validation data

$$\text{solve} \quad \underset{\mathcal{X}_A, \mathcal{X}_P}{\text{minimize}} \quad \text{err}_V([\mathcal{X}_A, \mathcal{X}_P]; \mathcal{V}; w^*[\mathcal{X}_A, \mathcal{X}_P])$$

Architecture space Hyperparameter space

DeepHyper: Scalable AutoML package



<https://github.com/deephyper/deephyper>

Acknowledgements



U.S. DEPARTMENT OF
ENERGY

DOE Early Career Research Program, ASCR

Argonne Leadership Computing Facility



Laboratory Directed Research and Development (LDRD)