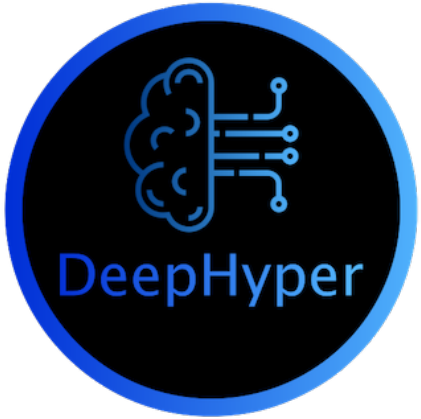
SEP 25, 2019

Argonne NATIONAL LABORATORY

DeepHyper: AutoML package for deep neural networks



Prasanna Balaprakash

Computer Scientist

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

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_1X_2

$\sin(X_1)$

$\sin(X_2)$

5 HIDDEN LAYERS

4 neurons | 5 neurons | 3 neurons | 4 neurons | 2 neurons

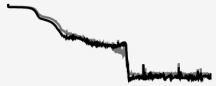
*This is the output from one **neuron**. Hover to see it larger.*

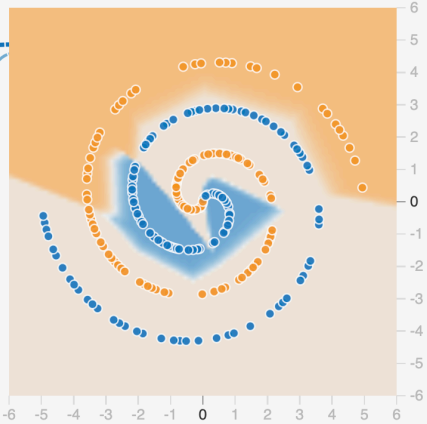
*The outputs are mixed with varying **weights**, shown by the thickness of the lines.*

OUTPUT

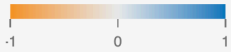
Test loss 0.316

Training loss 0.321





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_1X_2

$\sin(X_1)$

$\sin(X_2)$

5 HIDDEN LAYERS

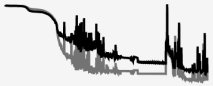
4 neurons | 4 neurons | 4 neurons | 4 neurons | 2 neurons

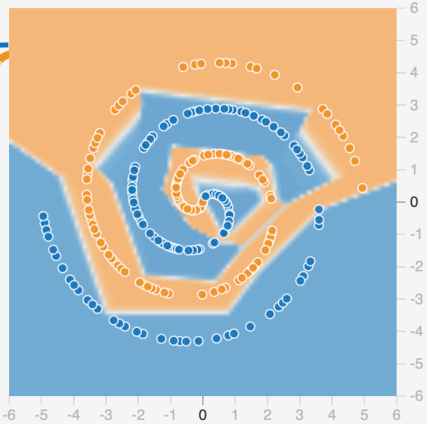
*This is the output from one **neuron**. Hover to see it larger.*

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

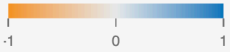
OUTPUT

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_1X_2

$\sin(X_1)$

$\sin(X_2)$

5 HIDDEN LAYERS

4 neurons | 4 neurons | 4 neurons | 4 neurons | 4 neurons

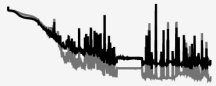
*This is the output from one **neuron**. Hover to see it larger.*

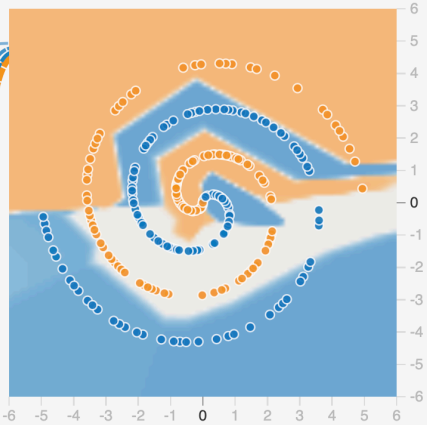
*The outputs are mixed with varying **weights**, shown by the thickness of the lines.*

OUTPUT

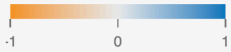
Test loss 0.239

Training loss 0.146





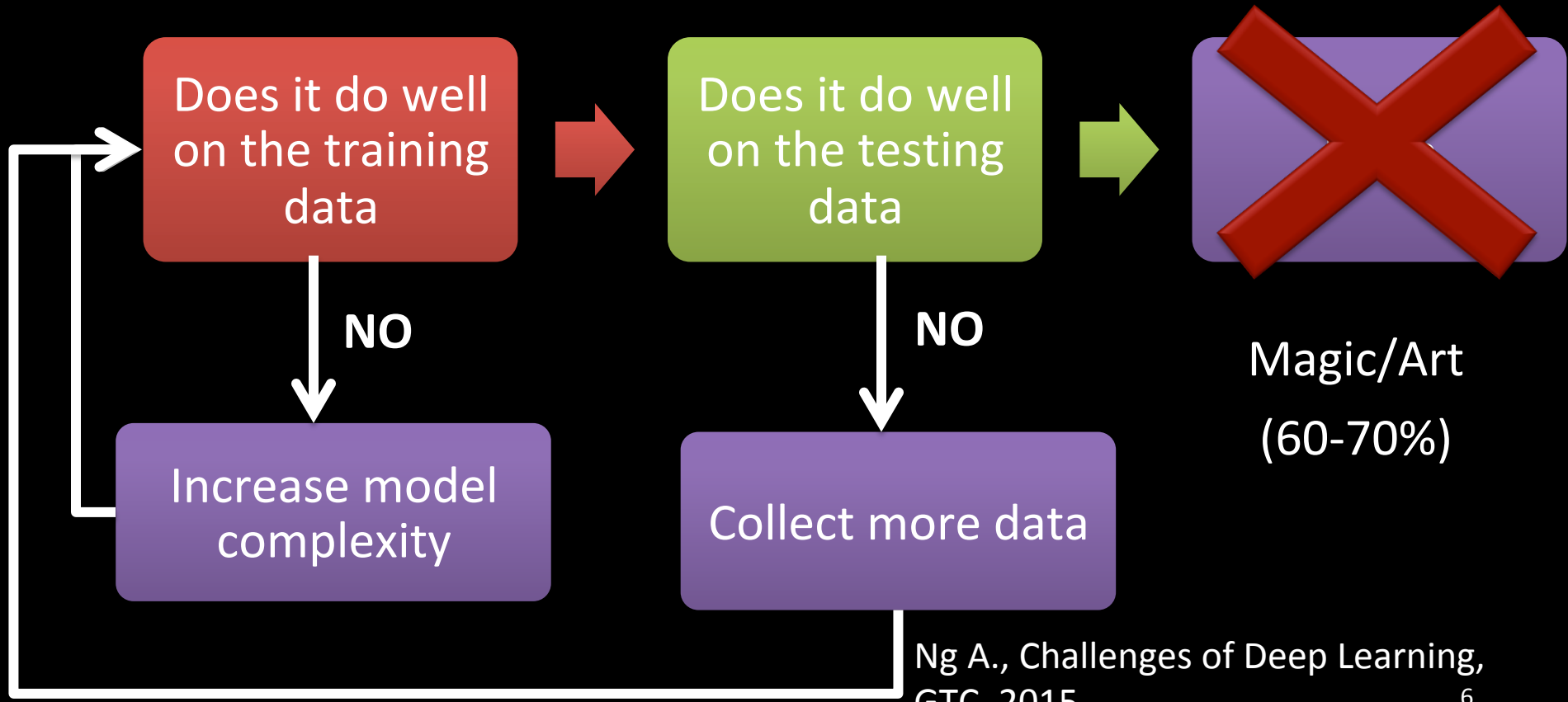
Colors shows data, neuron and weight values.



Show test data

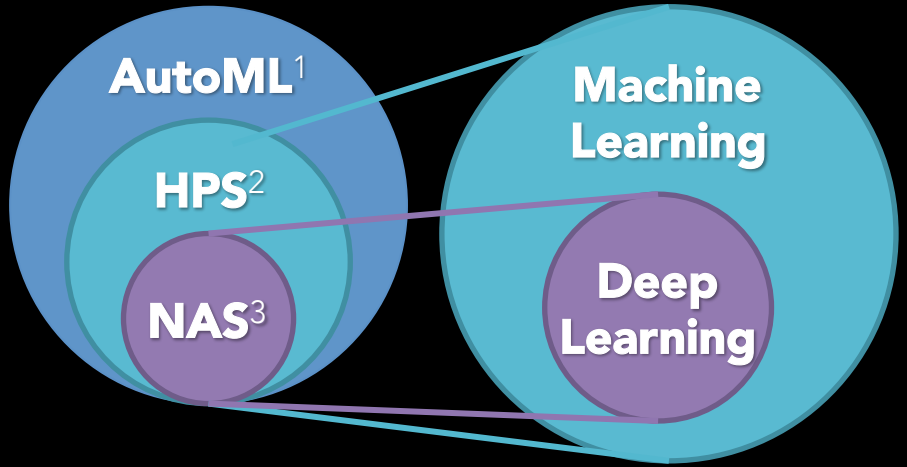
Discretize output

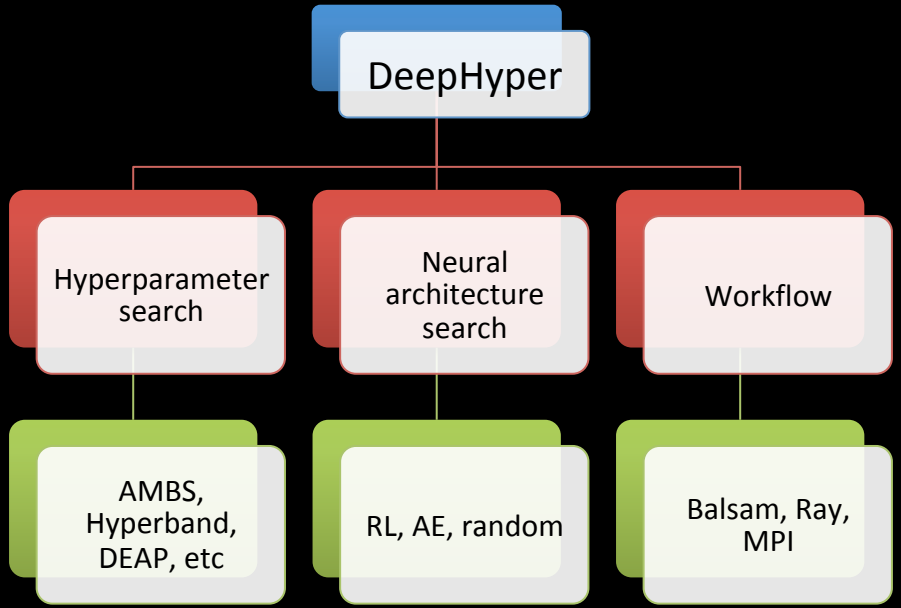
Deep neural networks



Ng A., Challenges of Deep Learning, GTC, 2015

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





[1]Automated Machine Learning
[2]Hyperparameter Search
[3]Neural Architecture Search

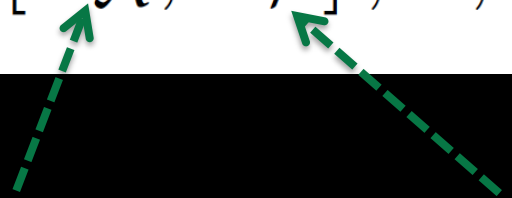
AutoML for deep learning

Lower-level problem:

$$\text{solve} \quad \underset{w}{\text{minimize}} \ \text{err}_T\left(\left[\mathcal{X}_\mathcal{A}, \mathcal{X}_\mathcal{P}\right]; \mathcal{T}; w\right)$$

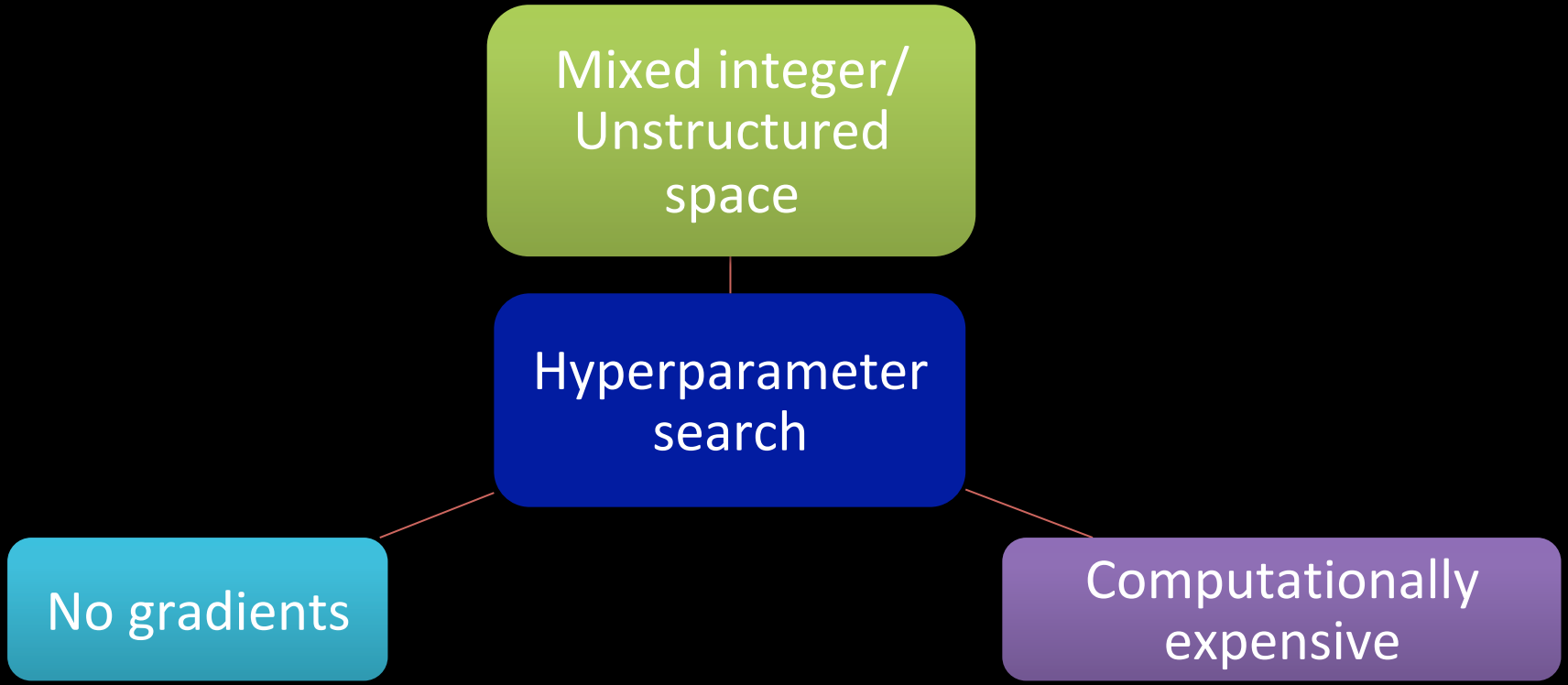
Upper-level problem:

$$\text{solve} \quad \underset{\mathcal{X}_\mathcal{A}, \mathcal{X}_\mathcal{P}}{\text{minimize}} \ \text{err}_V\left(\left[\mathcal{X}_\mathcal{A}, \mathcal{X}_\mathcal{P}\right]; \mathcal{V}; w^*\left[\mathcal{X}_\mathcal{A}, \mathcal{X}_\mathcal{P}\right]\right)$$



Architecture space Hyperparameter space

Challenges



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

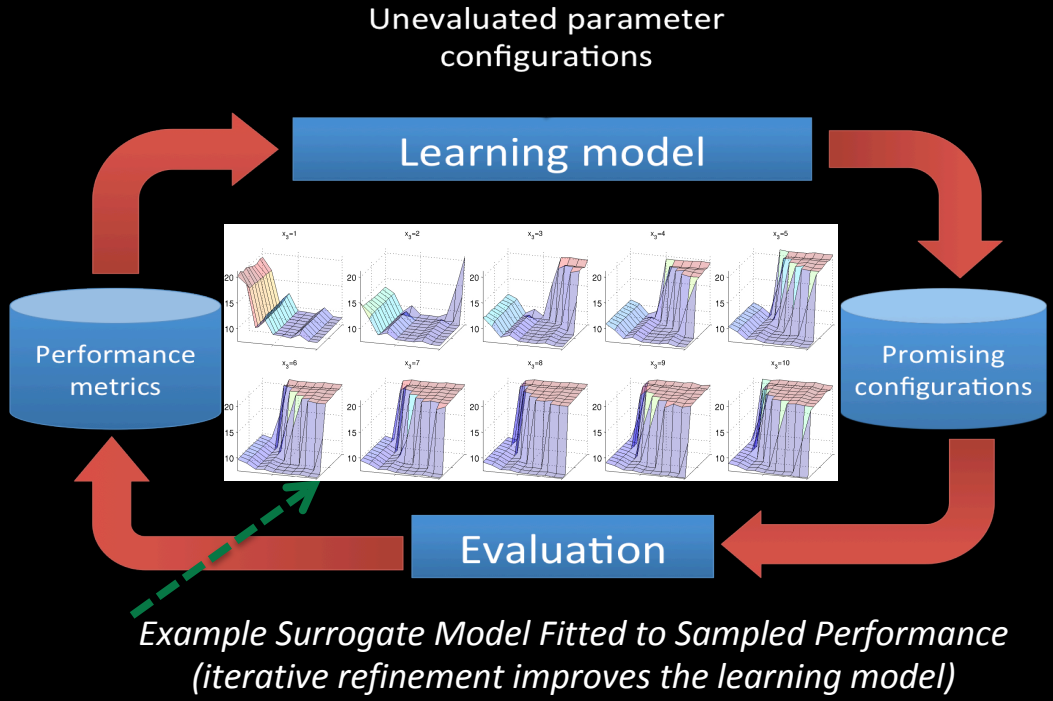
Example hyperparameters

```
# 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

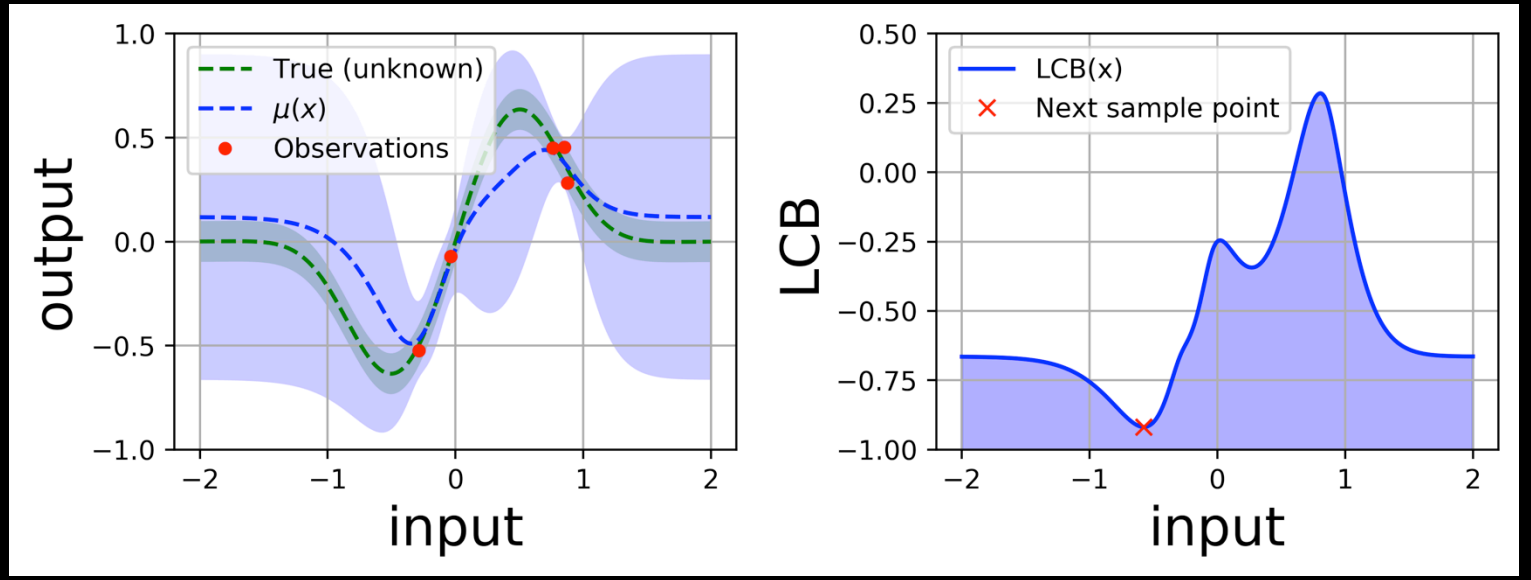
- Framework:
  - Initialization phase
    - Random or Latin hypercube sampling
  - Iterative phase
    - Fit model
    - Sample using the model

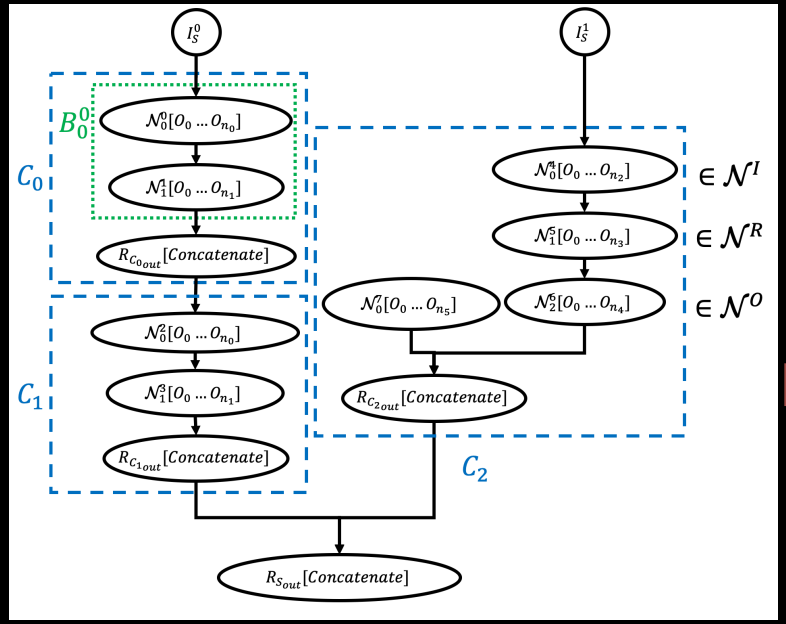


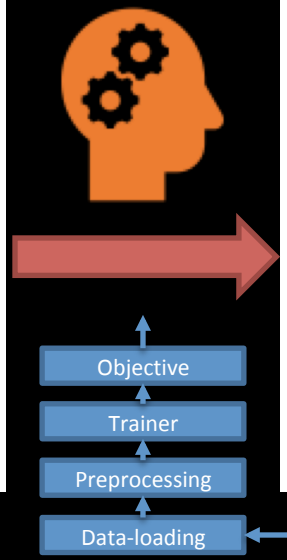
Example Surrogate Model Fitted to Sampled Performance (iterative refinement improves the learning model)

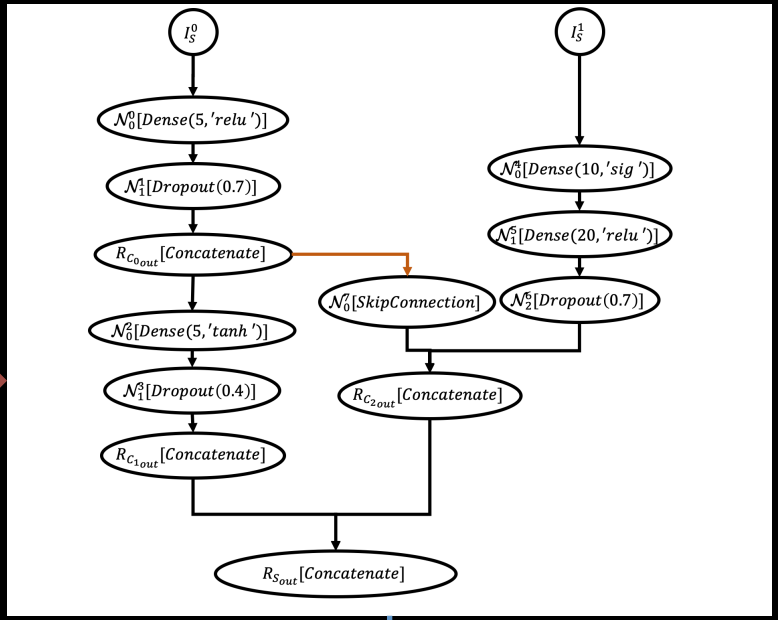
Bayesian optimization

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

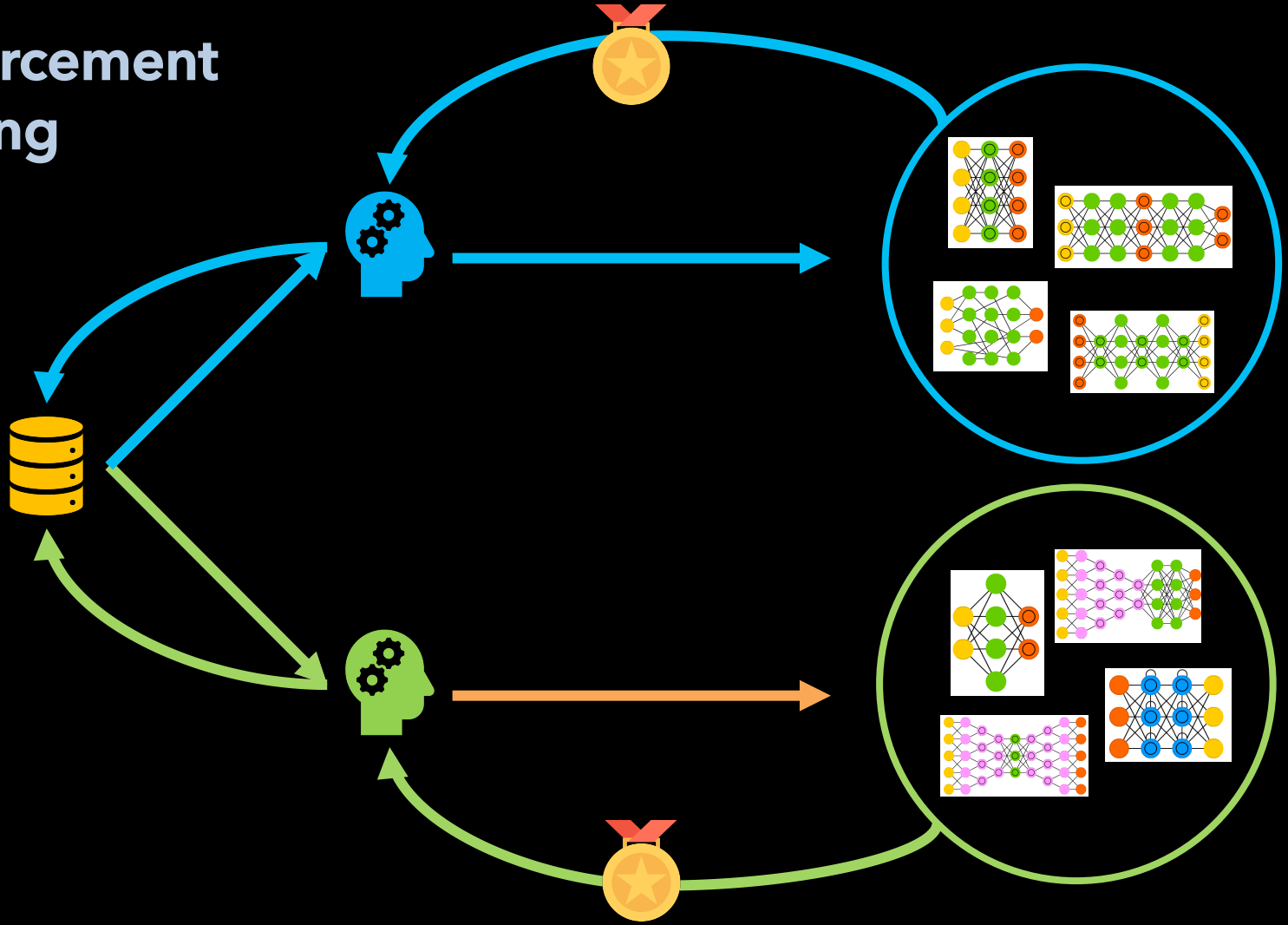



I_S^0
I_S^1
B_0^0
C_0
C_1
C_2
$\mathcal{N}_0^0[O_0 \dots O_{n_0}]$
$\mathcal{N}_1^0[O_0 \dots O_{n_1}]$
$R_{C_{0_{out}}}[Concatenate]$
$\mathcal{N}_0^1[O_0 \dots O_{n_0}]$
$\mathcal{N}_1^1[O_0 \dots O_{n_1}]$
$R_{C_{1_{out}}}[Concatenate]$
$\mathcal{N}_0^2[O_0 \dots O_{n_2}]$
$\mathcal{N}_1^2[O_0 \dots O_{n_3}]$
$\mathcal{N}_0^2[O_0 \dots O_{n_5}]$
$\mathcal{N}_2^2[O_0 \dots O_{n_4}]$
$R_{C_{2_{out}}}[Concatenate]$
$\in \mathcal{N}^I$
$\in \mathcal{N}^R$
$\in \mathcal{N}^O$
$R_{S_{out}}[Concatenate]$

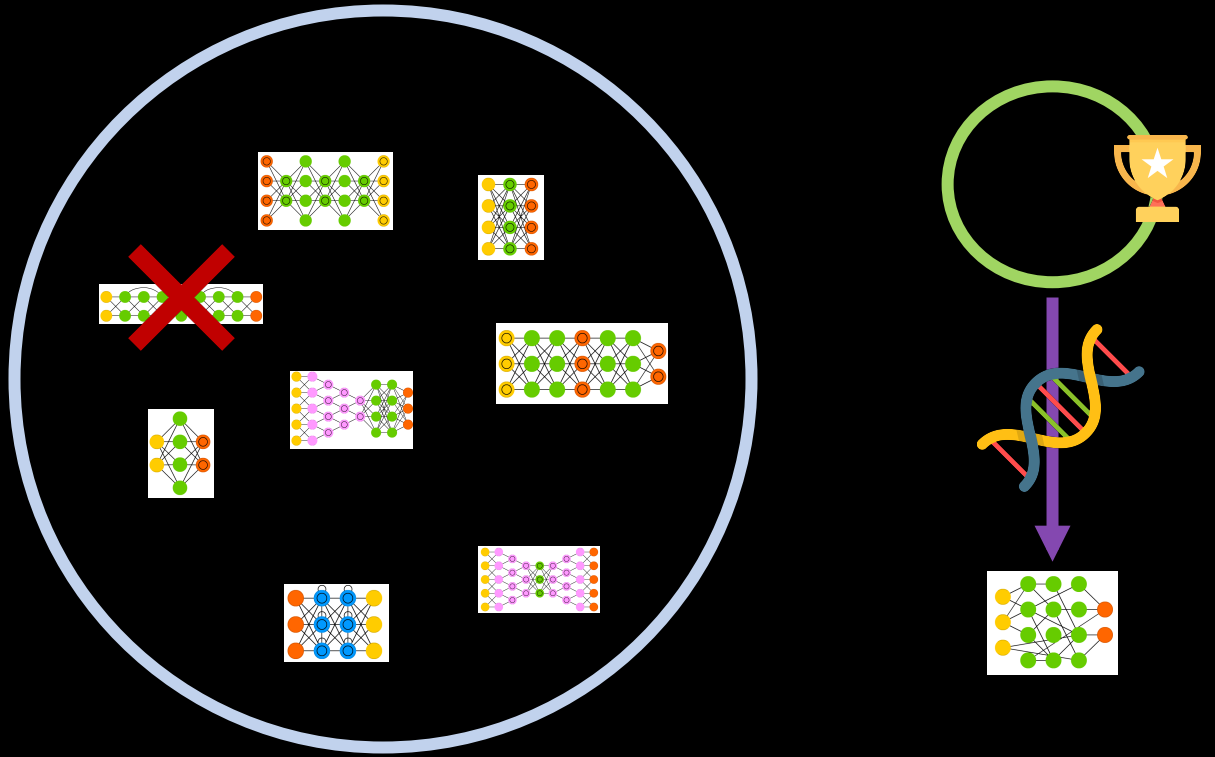

Objective
Trainer
Preprocessing
Data-loading


I_S^0
I_S^1
$\mathcal{N}_0^0[Dense(5,'relu')]$
$\mathcal{N}_1^0[Dropout(0.7)]$
$R_{C_{0_{out}}}[Concatenate]$
$\mathcal{N}_0^1[Dense(5,'tanh')]$
$\mathcal{N}_1^1[Dropout(0.4)]$
$R_{C_{1_{out}}}[Concatenate]$
$\mathcal{N}_0^2[Dense(10,'sig')]$
$\mathcal{N}_1^2[Dense(20,'relu')]$
$\mathcal{N}_0^2[SkipConnection]$
$\mathcal{N}_2^2[Dropout(0.7)]$
$R_{C_{2_{out}}}[Concatenate]$
$R_{S_{out}}[Concatenate]$

Reinforcement Learning



Aging Evolution



Automated machine learning for deep learning

Lower-level problem:

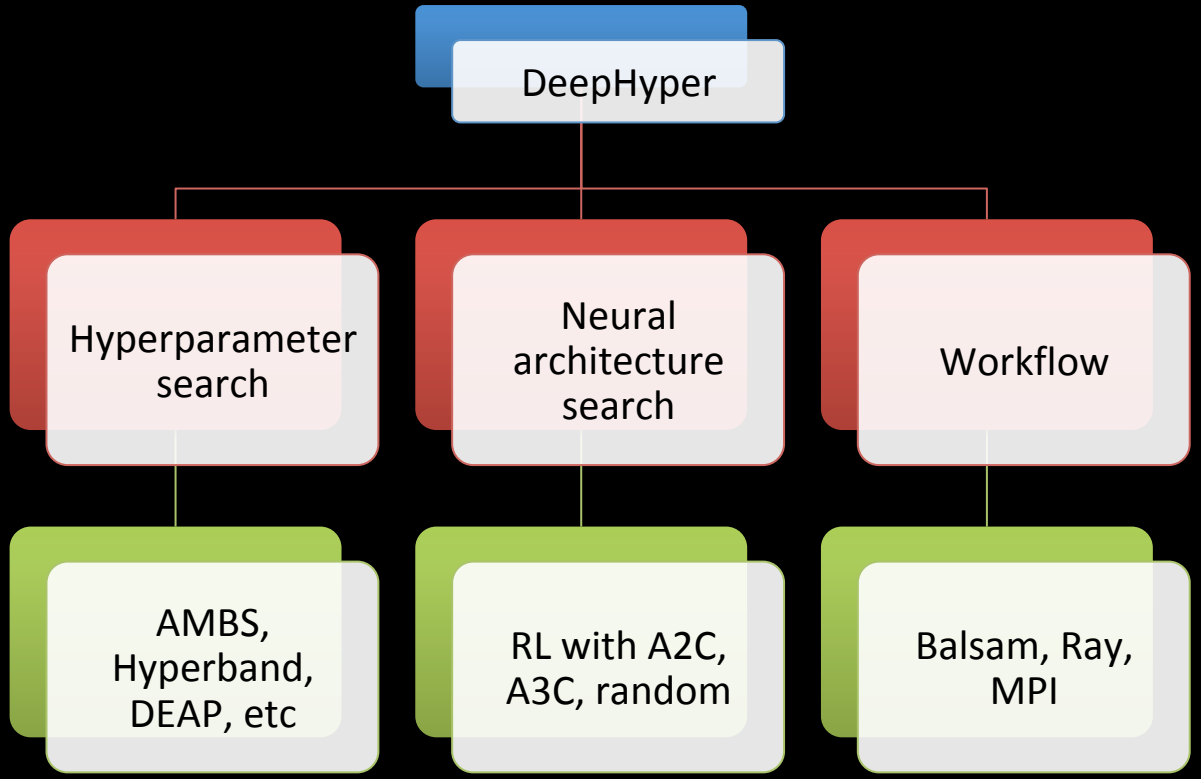
$$\text{solve} \quad \underset{w}{\text{minimize}} \ \text{err}_T\left(\left[\mathcal{X}_\mathcal{A}, \mathcal{X}_\mathcal{P}\right]; \mathcal{T}; w\right)$$

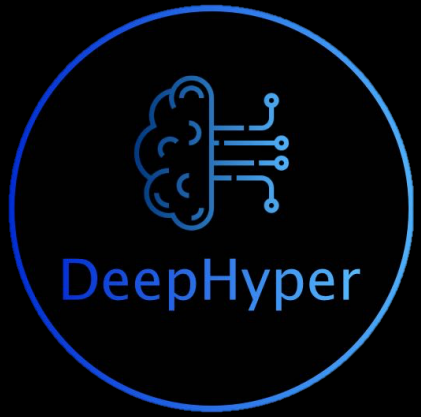
Upper-level problem:

$$\text{solve} \quad \underset{\mathcal{X}_\mathcal{A}, \mathcal{X}_\mathcal{P}}{\text{minimize}} \ \text{err}_V\left(\left[\mathcal{X}_\mathcal{A}, \mathcal{X}_\mathcal{P}\right]; \mathcal{V}; w^*\left[\mathcal{X}_\mathcal{A}, \mathcal{X}_\mathcal{P}\right]\right)$$

Architecture space Hyperparameter space

DeepHyper: Scalable AutoML package





https://github.com/deephyper/deephyper

Acknowledgements

DOE Early Career Research Program, ASCR

Argonne Leadership Computing Facility

Laboratory Directed Research and Development (LDRD)