Deep Learning in Practice

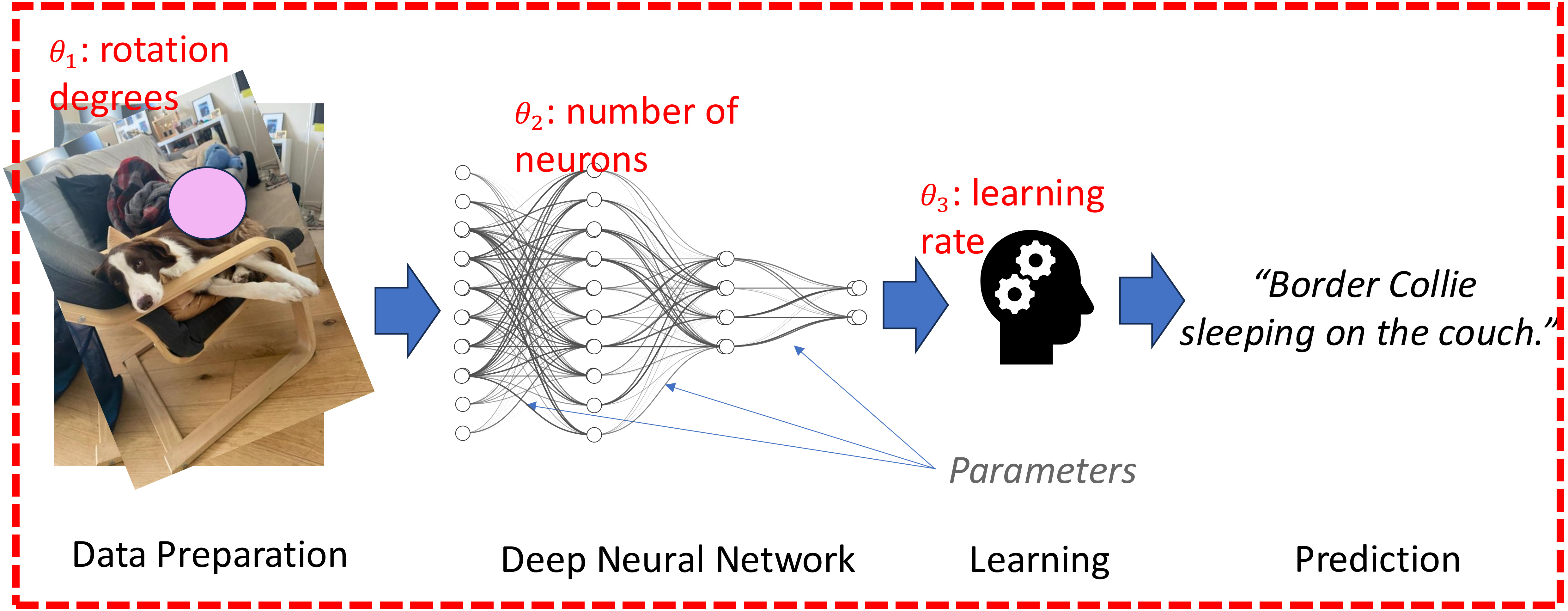
Optimization of Learning Workflows

Hyperparameter Optimization, Neural Architecture Search, Ensemble

Romain Egele

Oak Ridge National Laboratory

An Example of Learning Workflow



$$c = f(\theta)$$

Parametrization of Learning Worfklows

Data Preparation θ_1

Model Design θ_2

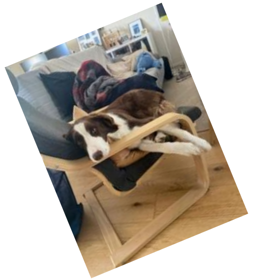
Learning θ_3

Predictor

$$c = f(\theta)$$

Categorical, Discrete, Real

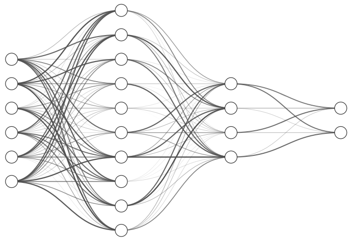
Data Preparation



θ_1
- Rotation Degrees
- Augmentation Type
- Noise Rate
- …

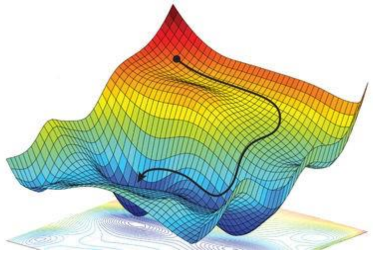
[Egelé and Jacques Junior, 2024]

Model Design



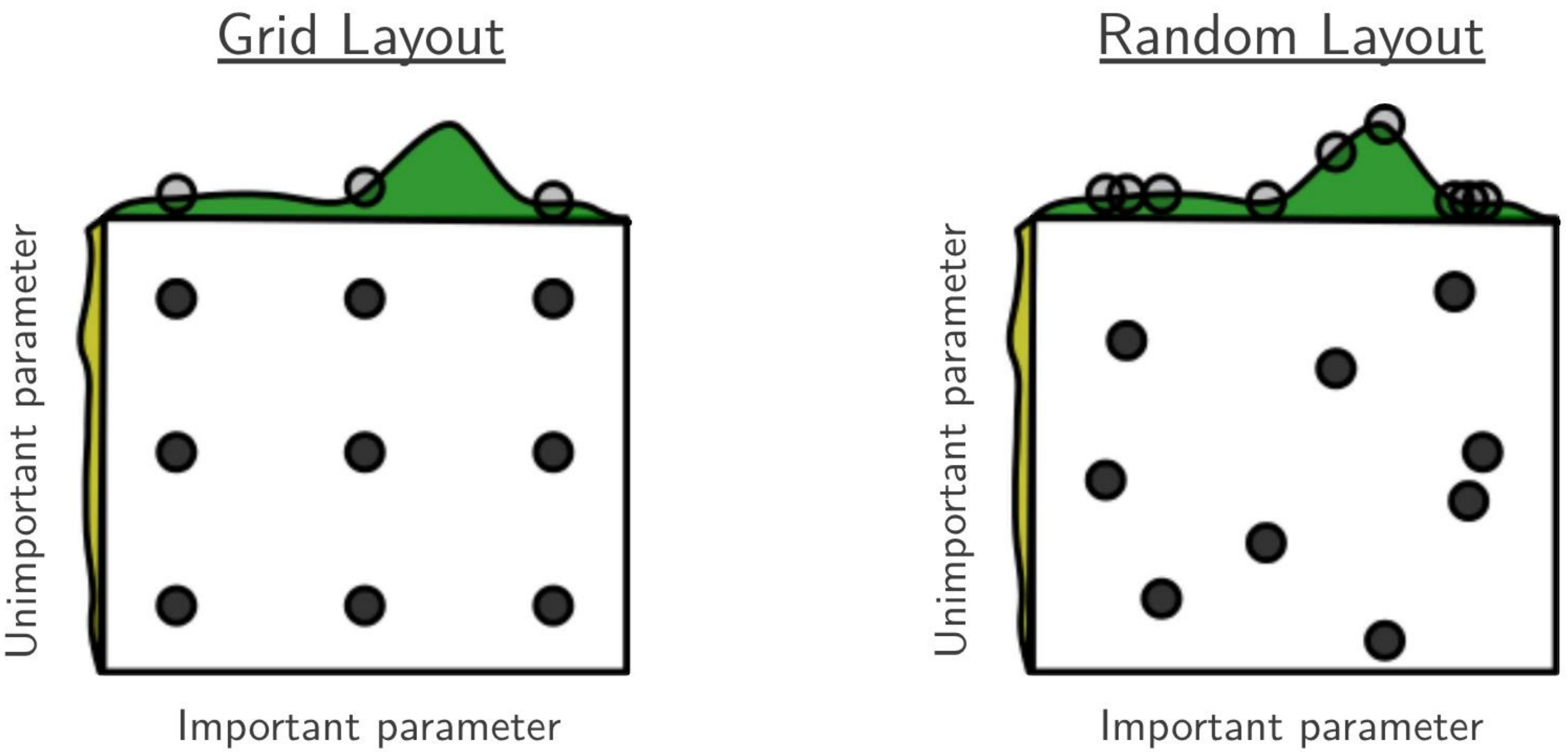
θ_2
- Number of layers
- Layer Type
- Dropout rate
- …

Learning



θ_3
- Optimizer
- Learning rate
- Batch size
- …

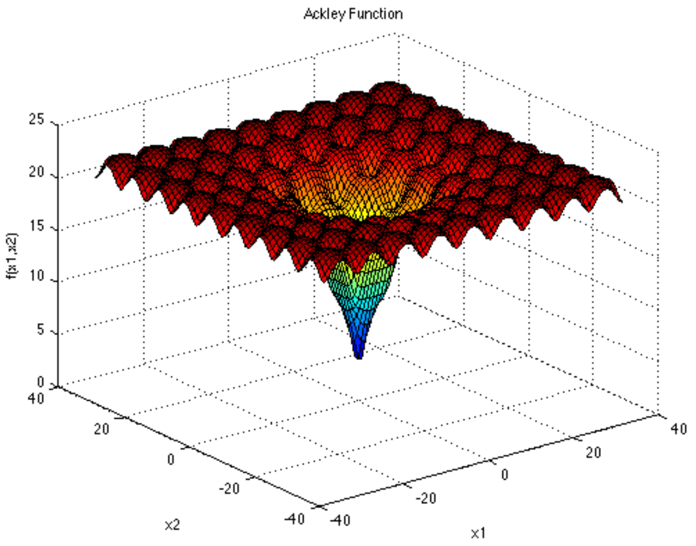
Grid Search

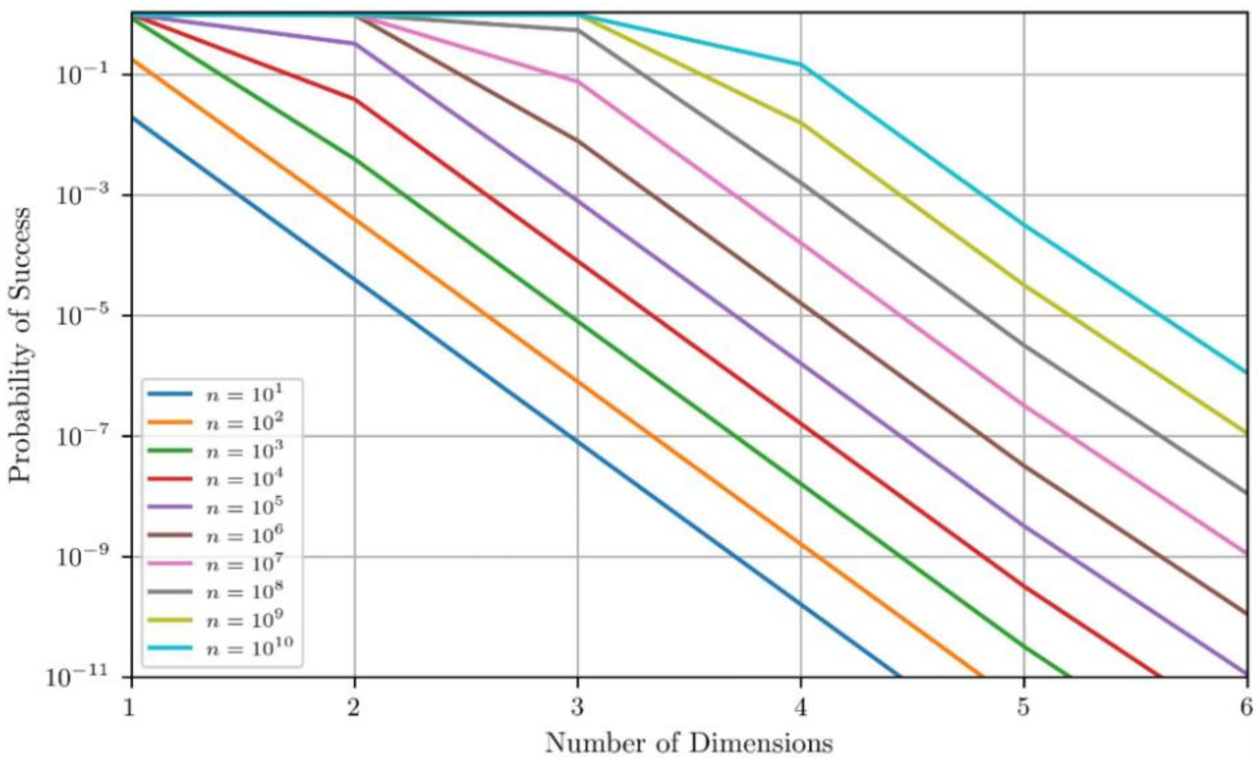


(Bergstra et Bengio, 2012)

Random Search

$$p_\epsilon := \mathrm{P}\left(|\theta - \theta^*| \leq \epsilon\right)$$





(b) Increasing the Number of Samples

Example with DeepHyper

http://tinyurl.com/deephyper-grid

Black-Box Optimization

$$\Theta^* = \arg\min_{\theta \in \Theta} f(\theta)$$

In General

The **input space** can be a mixed of Real, Discrete, Categorical parameters.

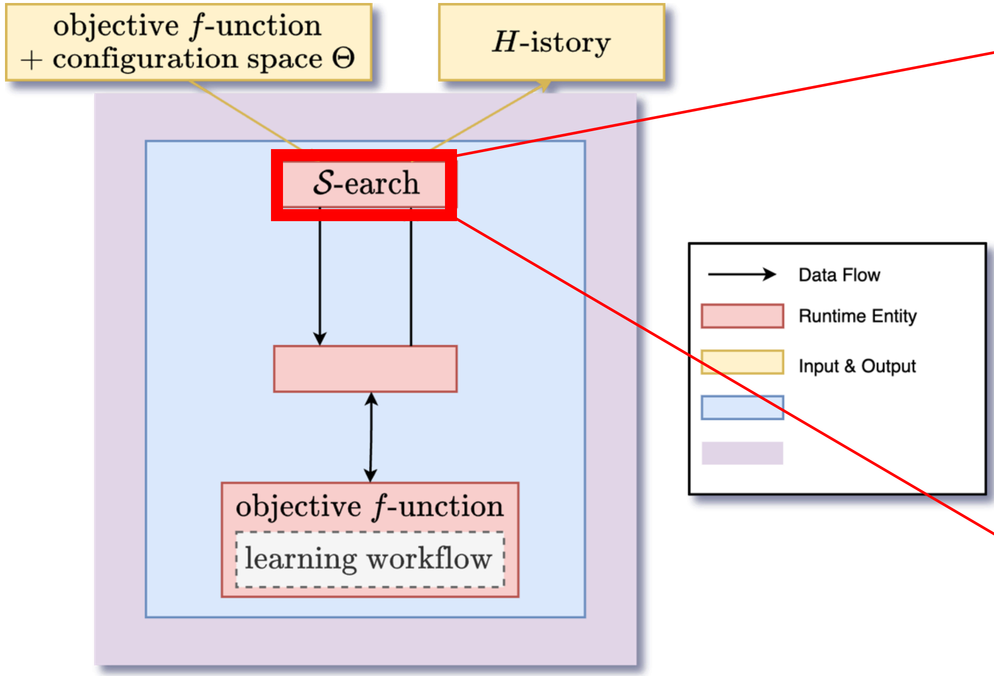
The **output space** is often mapped to real.

The **function** can be continuous, noisy, derivable.

Bayesian Optimization Framework

Find hyperparameters θ to improve cost c = $f(\theta)$
of the learning workflow

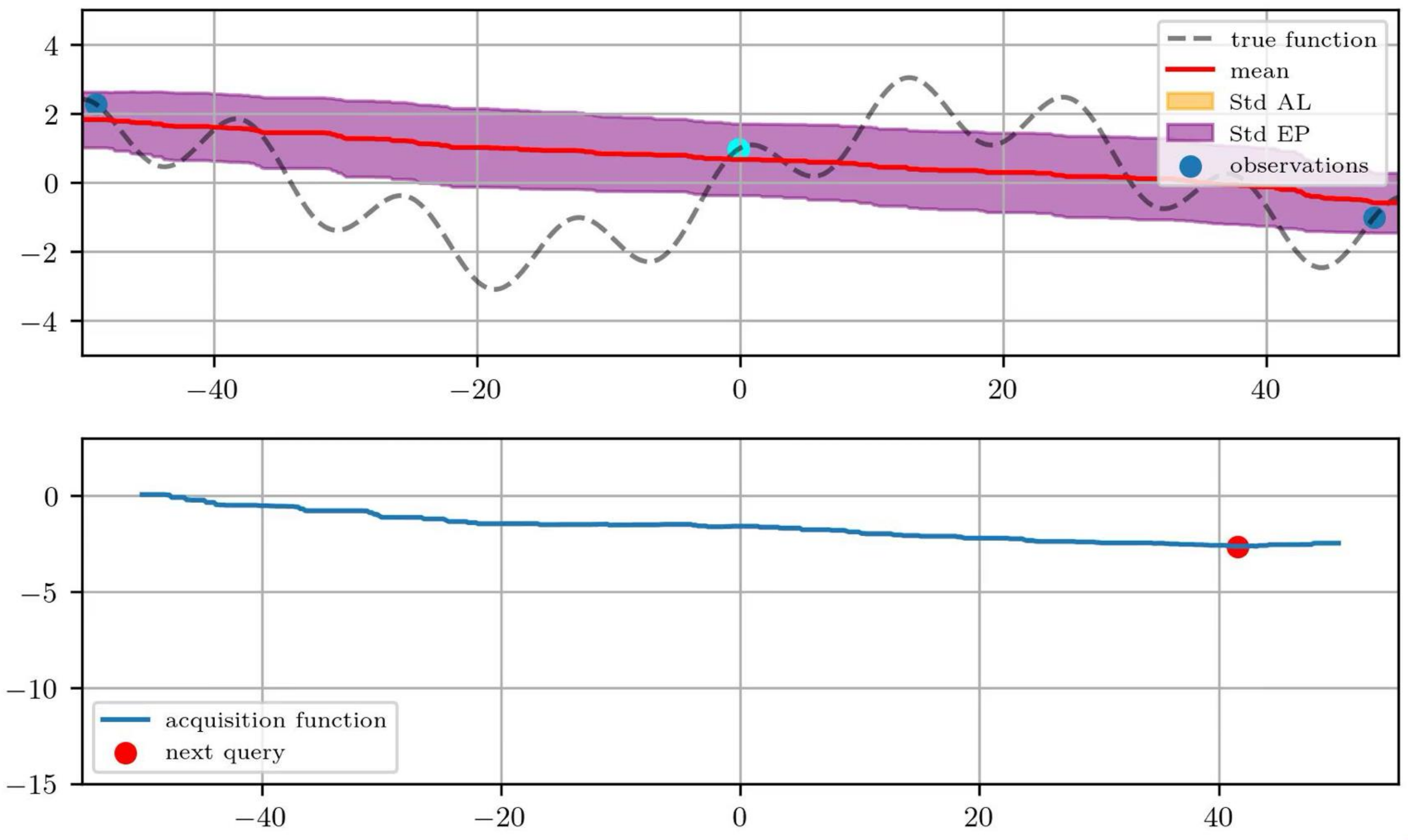
$$c^* = \min_{\theta} f(\theta)$$



1. <u>Suggest</u> *hyperparameters* θ
2. <u>Evaluate</u> cost c = $f(\theta)$
3. <u>Update</u> g surrogate of c = $f(\theta)$

Repeat

An Example of Bayesian Optimization



Example with DeepHyper

http://tinyurl.com/deephyper-bbo

Sequential Bayesian Optimization Algorithm

Algorithm 1: Bayesian Optimization (*a.k.a.*, Efficient Global Optimization (EGO))

```
Inputs :  thetaSpace: a configuration space
          nInitial: the number of initial hyperparameter configurations
          f: a function that returns the cost of the learning workflow
Output:   thetaStar the recommended hyperparameter configuration.
thetaArray, costArray ← New empty arrays of hyperparameter configurations and costs ;
model ← New surrogate model ;
while stopping criteria not valid do
    if Length of thetaArray < nInitial then
        theta ← Sample hyperparameter configuration from thetaSpace ;
    else
        Update model with thetaArray, costArray ;
        theta ← Returns theta in thetaSpace that minimizes the acquisition function for current model ;
    end
    cost ← Returns the cost of learning workflow f(theta) ;
    thetaArray, costArray ← Concatenate thetaArray with [theta] and costArray with [cost];
    thetaStar ← Update recommendation ;
end
```

Acquisition functions

- Lower Confidence Bound

$$a_{\mathsf{LCB}}(\theta;\kappa) := \mu(\theta) - \kappa \cdot \sigma(\theta)$$

- Probability of Improvement

$$\begin{aligned} I(\theta;\xi) &:= \max(f(\theta^{\circledast}) - f(\theta) - \xi, 0) \\ &= \max(f(\theta^{\circledast}) - \mu(\theta) - z \cdot \sigma(\theta) - \xi, 0) \quad \text{with } z \sim \mathcal{N}(0,1) \end{aligned}$$

$$a_{\mathsf{PI}}(\theta;\xi) := \mathrm{P}\left(0 < I(\theta;\xi)\right)$$

- Expected Improvement

$$a_{\mathsf{EI}}(\theta;\xi) := \mathbb{E}\left[I(\theta;\xi)\right]$$

Lower Confidence Bound

Acquisition function is based on uncertainty.

Surrogate g

$$a_{LCB}(\theta;\kappa) = \mu(\theta) - \kappa \cdot \sigma(\theta)$$

Lower-Confidence Bound

[Cox, 1992]

"Most optimistic outcome"

Simple, cheap, less flat

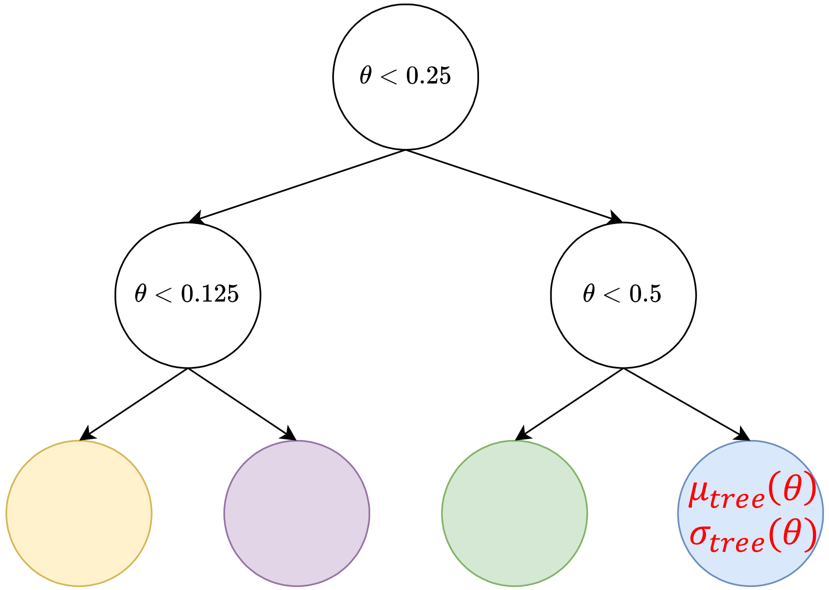
Next hyperparameters suggested have minimum LCB score

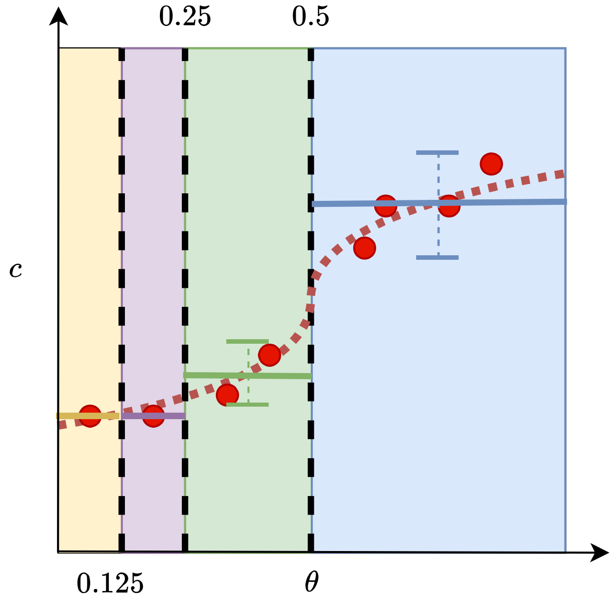
$$\theta^* = \min_{\theta} a_{LCB}\,(\theta;\kappa)$$

Randomized forests as model

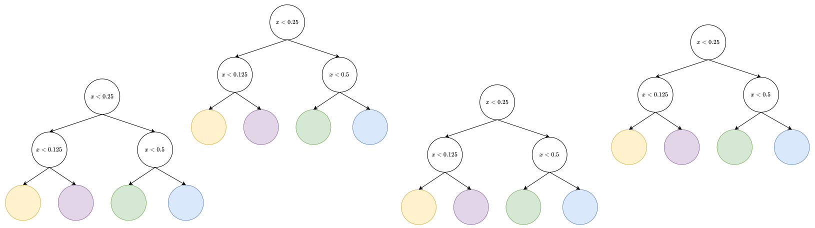
Surrogate g is an ensemble of randomized decision trees.

Randomized Decision Tree





Random-Forest
Surrogate g



Other Seminal Surrogates

Gaussian Process [Eriksson, 2019],
Bayesian NN [Springenberg, 2016],
Random-Forest [Hutter, 2011] (temporal complexity and parallelizable)
TPE [Bergstra, 2013]...

Randomized forests and uncertainty

$$a_{LCB}(\theta;\kappa) = \overbrace{\mu(\theta) - \kappa \cdot \sigma(\theta)}^{\text{Surrogate } g}$$

$$\sigma^2(\theta) = \underbrace{E_{\text{tree}}[\sigma^2_{\text{tree}}(\theta)]}_{\text{aleatoric}} + \underbrace{V_{\text{tree}}[\mu_{\text{tree}}(\theta)]}_{\text{epistemic}}$$

Randomization Effects

Algorithms

	Bootstrapping	Feature	Split
Tree Bagging (TB)	x		
Random Space (RS)		x	
Random Forest (RF)	x	x	
Extremely Randomized Trees (ET)			x
Mondrian Forest (MF)		x	x

Randomized Splits

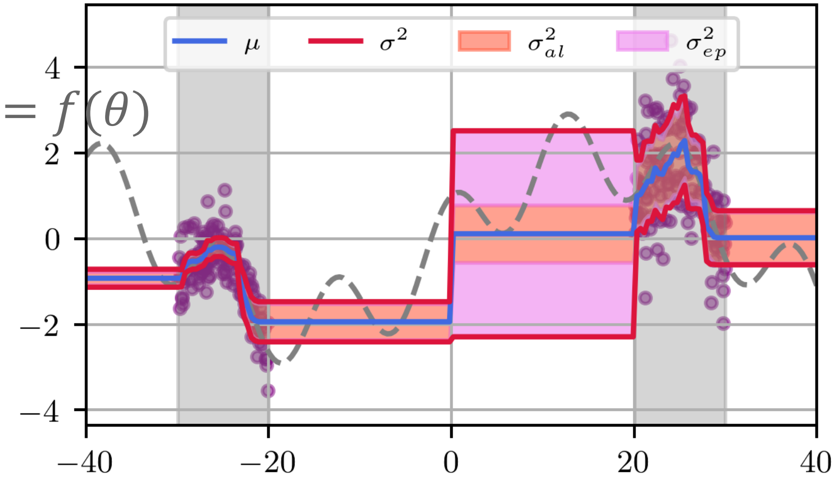
Surrogate g

$$a_{LCB}(\theta;\kappa) = \mu(\theta) - \kappa \cdot \sigma(\theta)$$

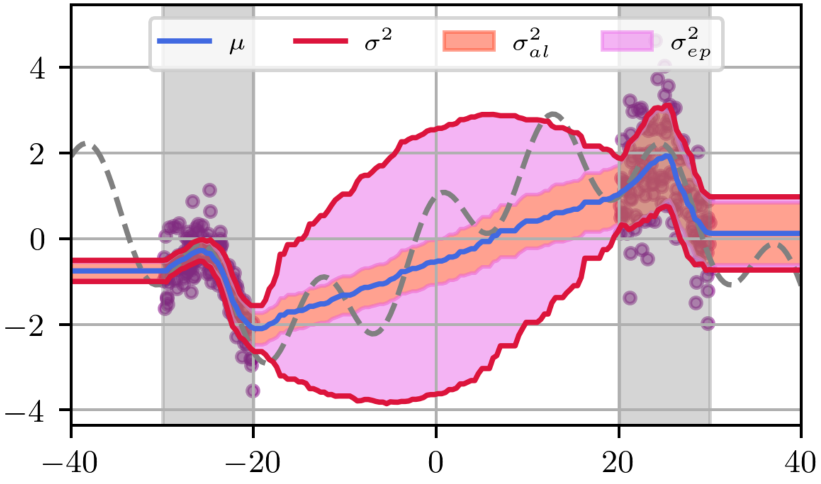
$$\sigma^2(\theta) = \underbrace{E_{\text{tree}}[\sigma^2_{\text{tree}}(\theta)]}_{\text{aleatoric}} + \underbrace{V_{\text{tree}}[\mu_{\text{tree}}(\theta)]}_{\text{epistemic}}$$

Best Split

$c = f(\theta)$



Random Split



[1, Sec. 4.3.2] Hutter, Frank, et al. "Algorithm runtime prediction: Methods & evaluation." *Artificial Intelligence* 206 (2014): 79-111.

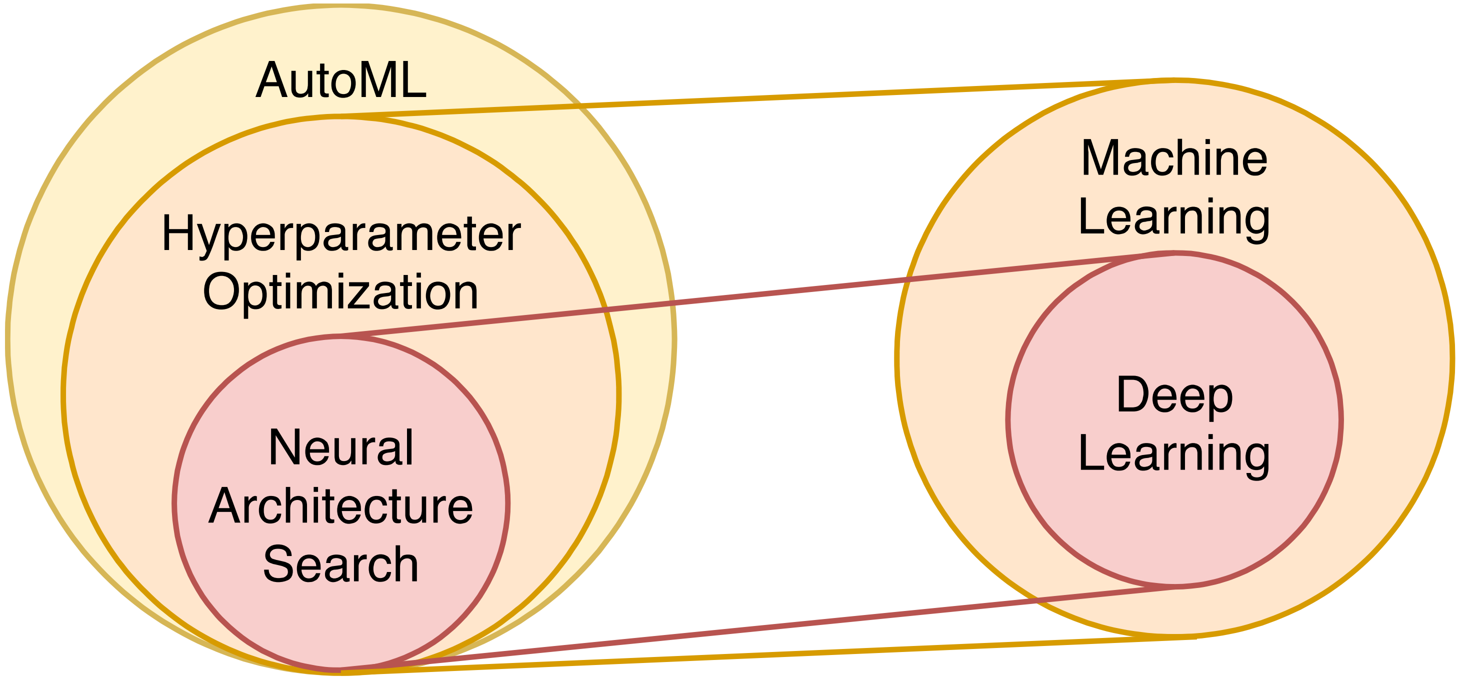
Optimization of the Acquisition Function

- Depends on the surrogate model (e.g., can we have a gradient)?
- Zero, First, Second order…

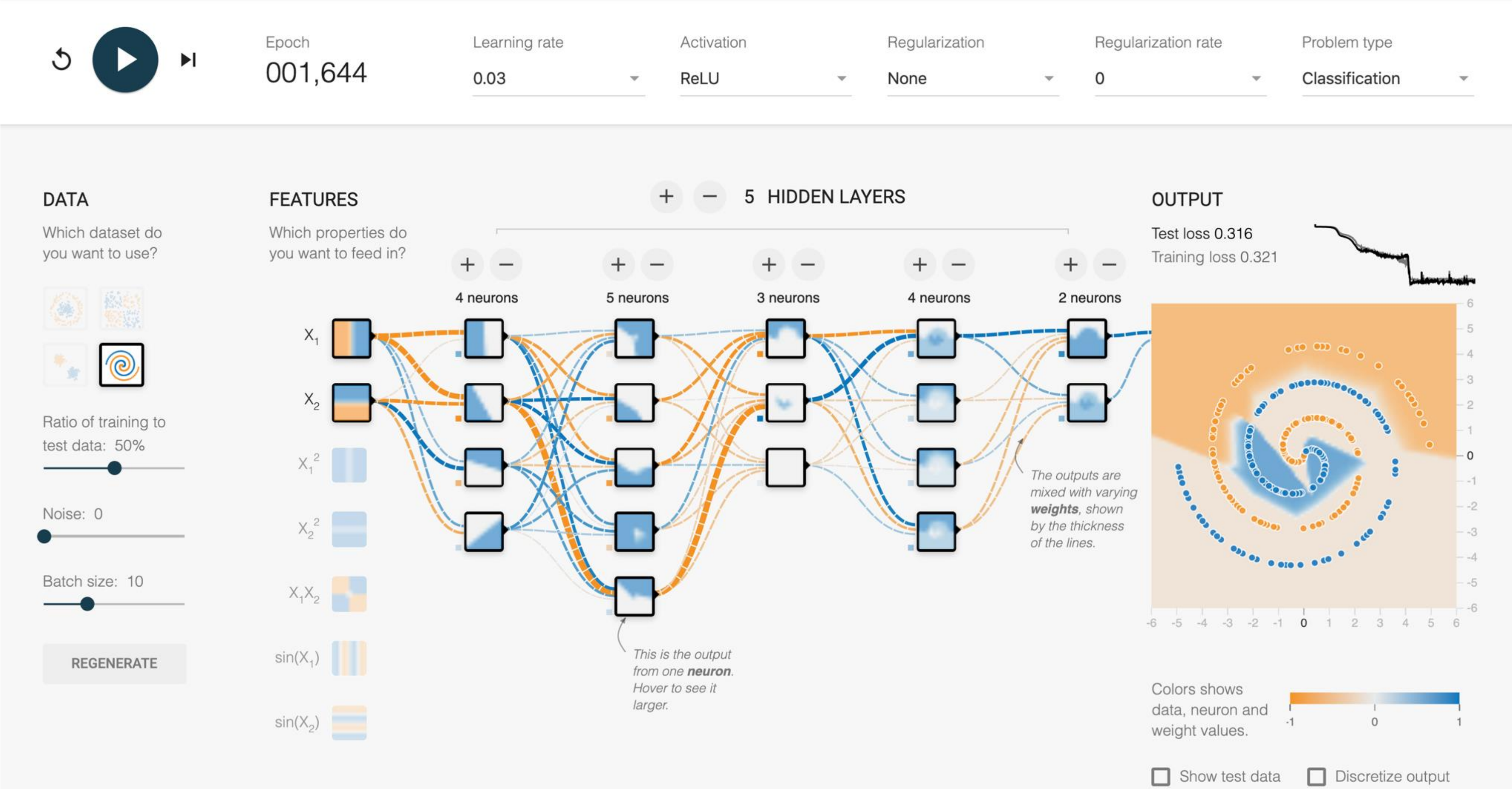
Examples:

- SGD
- LBFGS
- Genetic Algorithm
- CMAES (Covariance Matrix Adaptation Evolutionary Search)

Neural Architecture Search

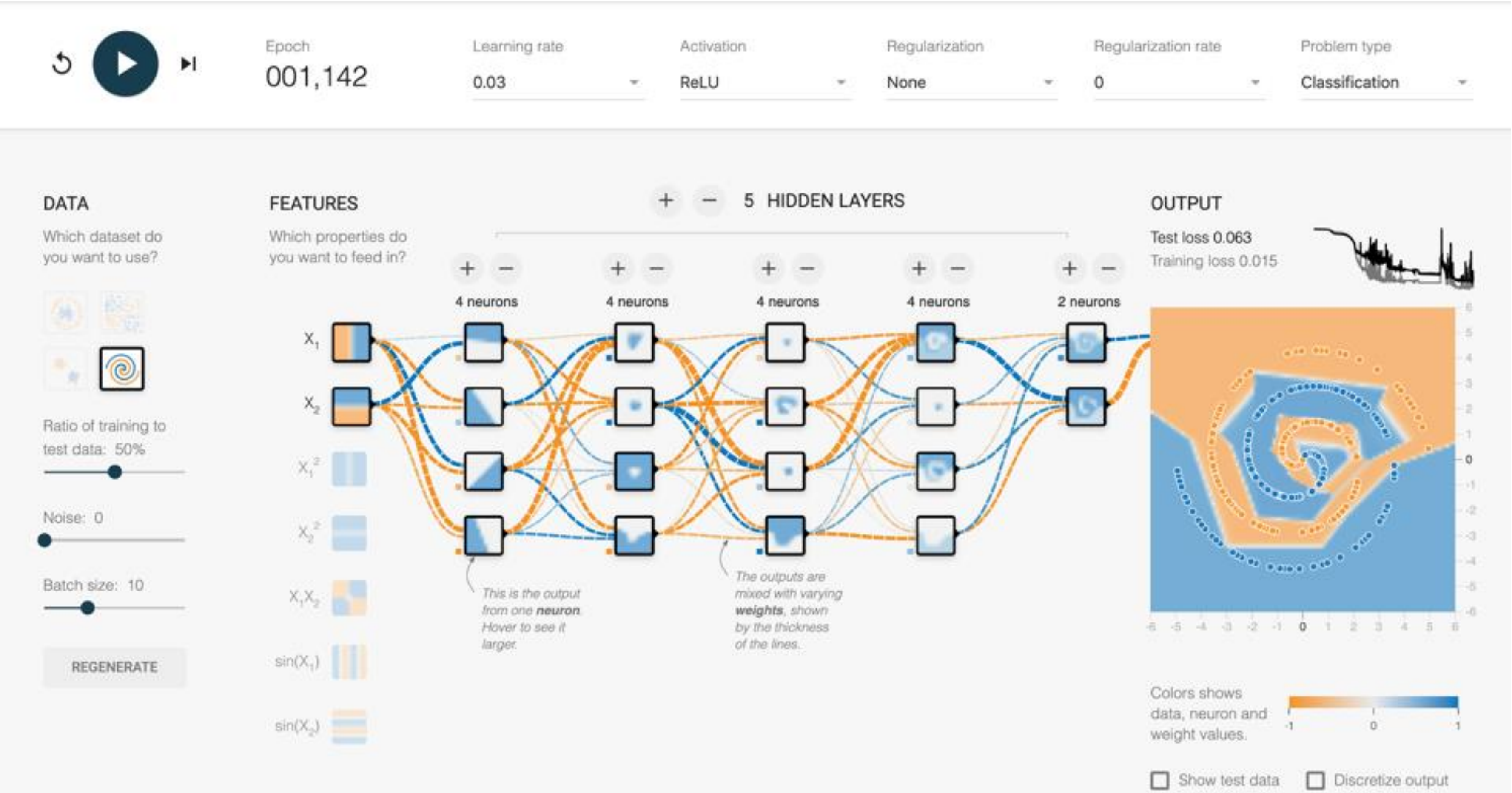


Example of sensitive neural architecture



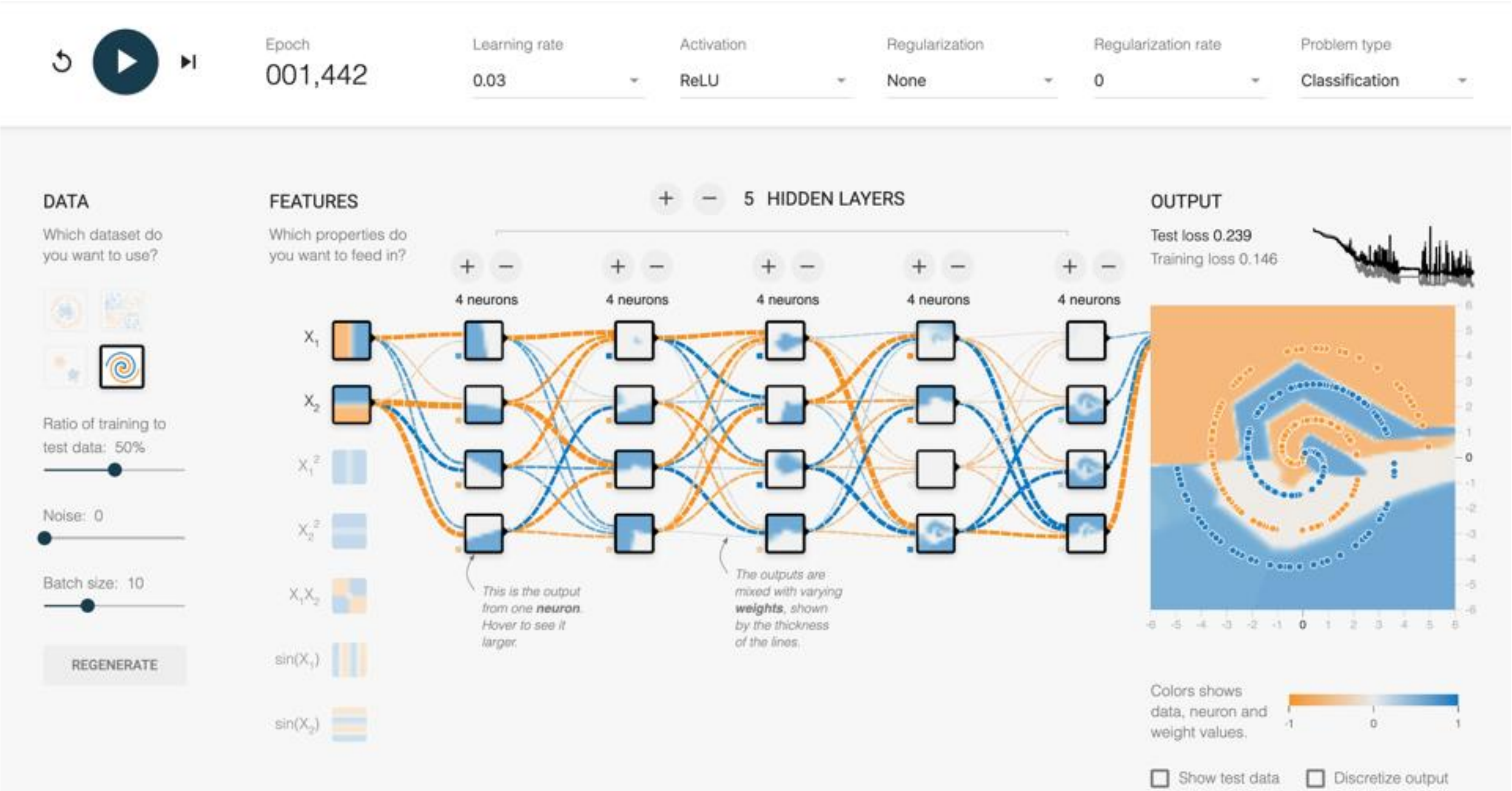
https://playground.tensorflow.org/

Example of sensitive neural architecture



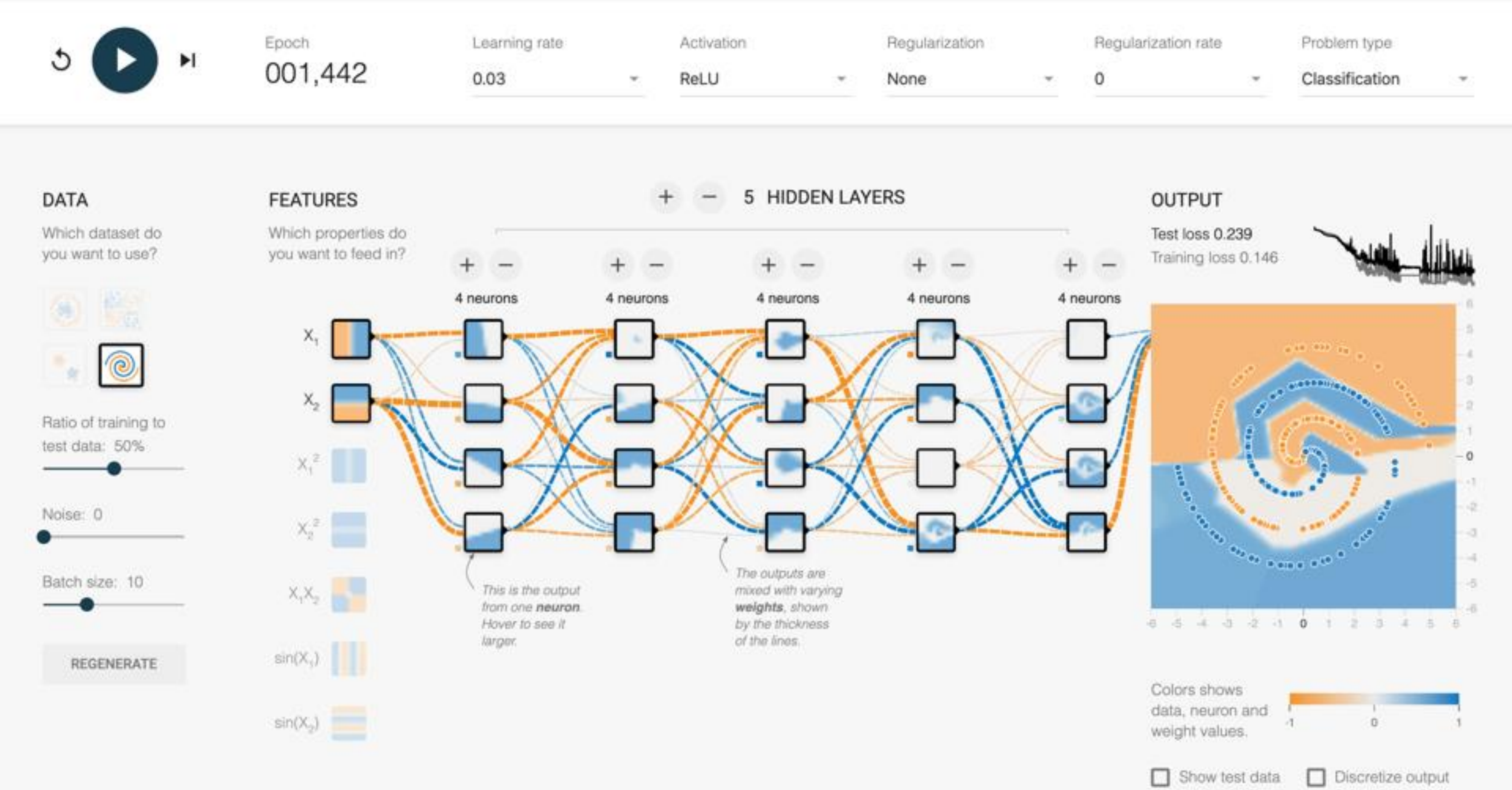
https://playground.tensorflow.org/

Example of sensitive neural architecture



https://playground.tensorflow.org/

Example of sensitive neural architecture



https://playground.tensorflow.org/

The performance of neural networks can be very sensitive with respect to the "architecture" (i.e. hyperparameters) of the neural network.

Hyperparameter Optimization with Constraints

- A layer is active if the number of layer is large enough.
- The parameters of a layer change depending on its type (dense, conv, batchnorm, dropout).
- Residual/skip connections can be created if input/output layers exists.

Example with DeepHyper

http://tinyurl.com/deephyper-autodeuq-reg

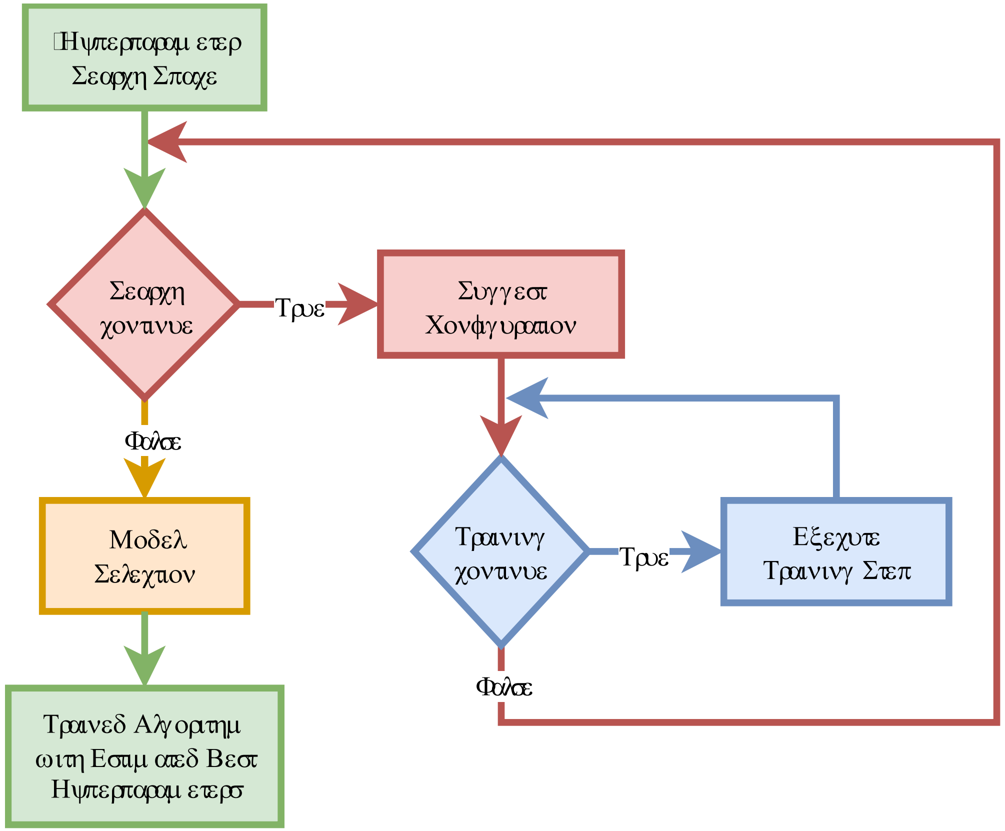
Overfitting in Hyperparameter Optimization

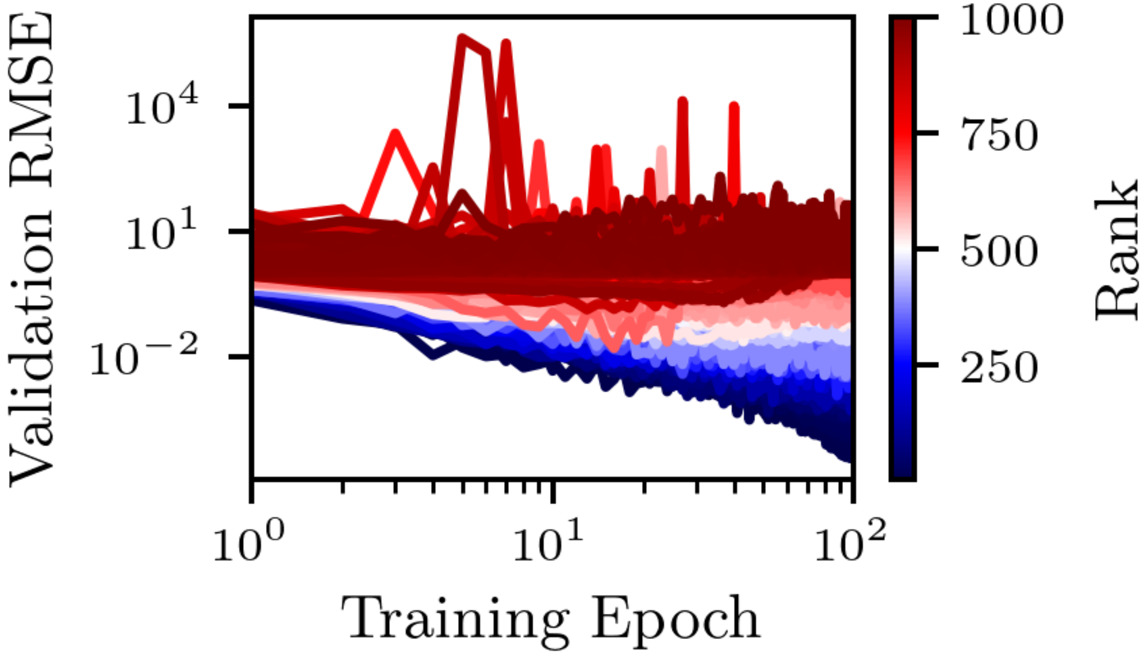
- 3-way split of the data: Training, Validation, Test
  - Training: for the weights of a neural network
  - Validation: for the hyperparameters
  - Test: generalization performance

Development Data

- Overfitting in HPO would mean that validation score improves when test score worsen.
  - Generaly not observed…
  - Similar to the problem of overfitting the test set for Cifar10/Imagenet
  - Similar to the problem of development phase and final phase in machine learning competitions (Kaggle)

Early discarding





Interested to dive in?

https://github.com/deephyper/deephyper