

# Neural Architecture Search (NAS) and Uncertainty Quantification (UQ) with DeepHyper

Romain Egele, Prasanna Balaprakash

Simulation, Data and Learning Workshop (October 7<sup>th</sup> 2021)



# The DeepHyper Project

"Automated development of machine learning algorithms to support scientific applications"



**Prasanna  
Balaprakash**



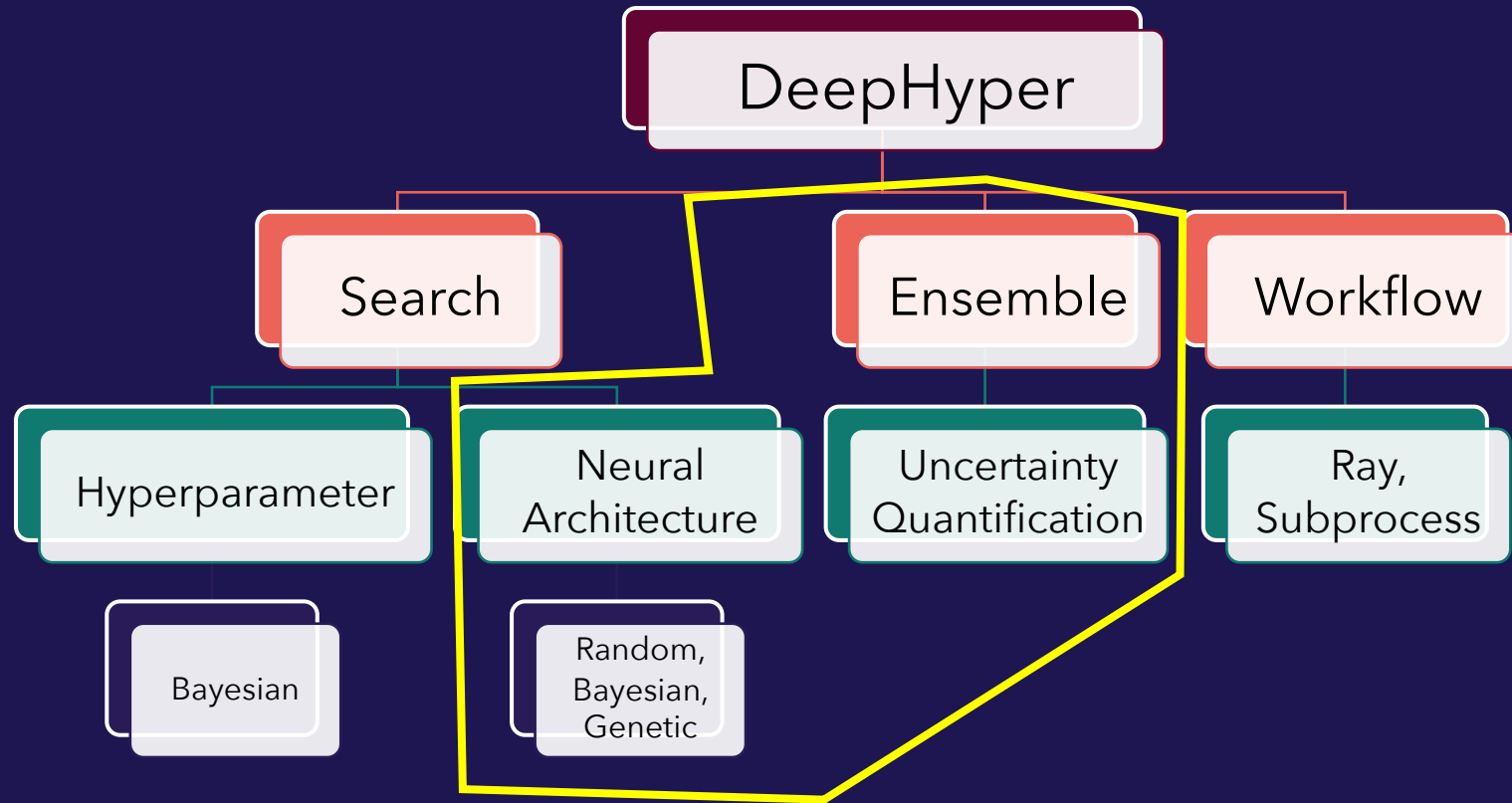
**Romain  
Egele**



Open-Source

<https://deephyper.readthedocs.io/>

# DeepHyper Overview



DeepHyper documentation: <http://deephyper.readthedocs.io>

# Installed on ALCF systems

- **Theta**

```
$ module load conda/2021-09-22
```

- **ThetaGPU**

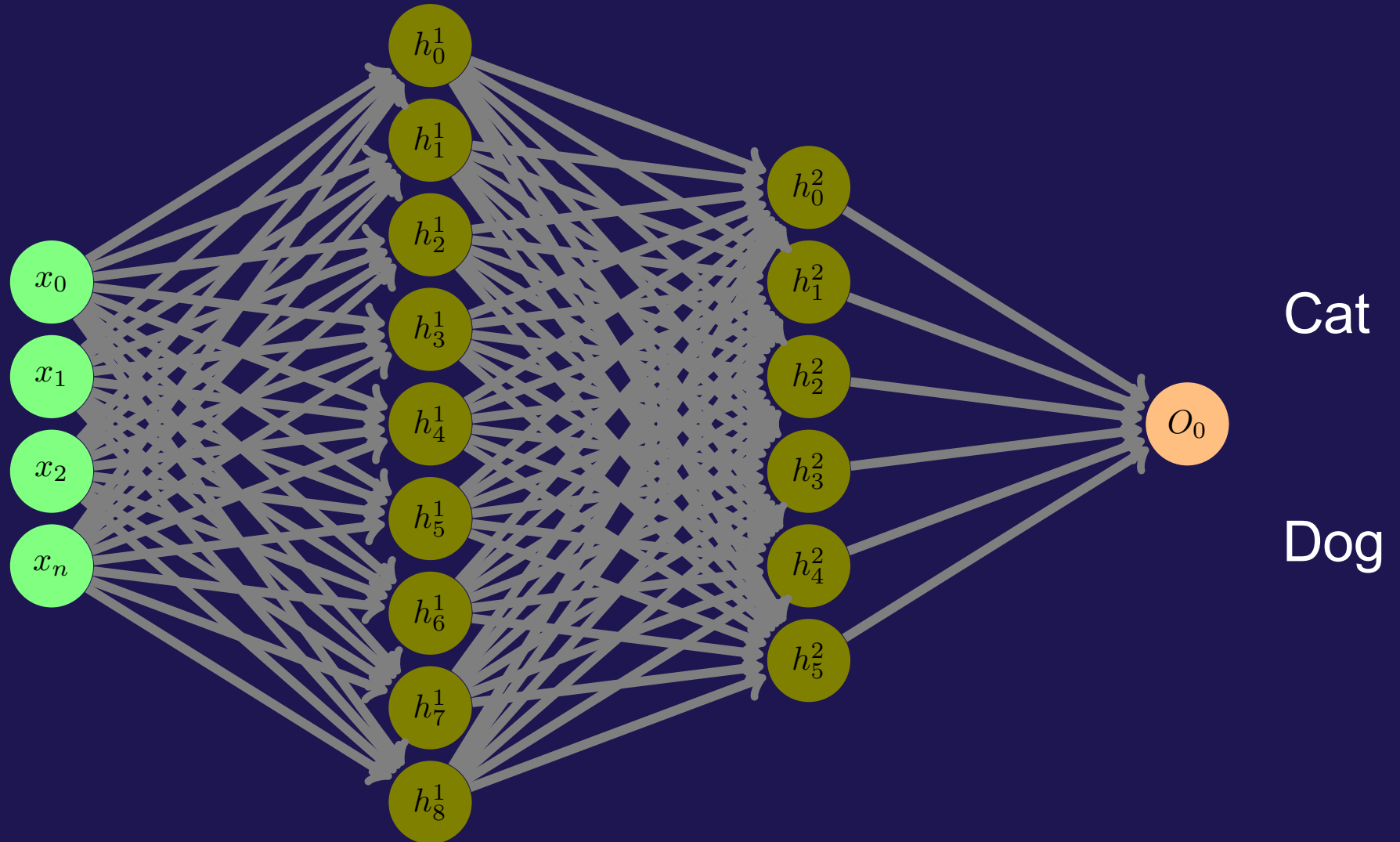
```
$ module load conda/2021-09-22
```

**Warning:** After loading the module, don't forget to run `$ conda activate base`



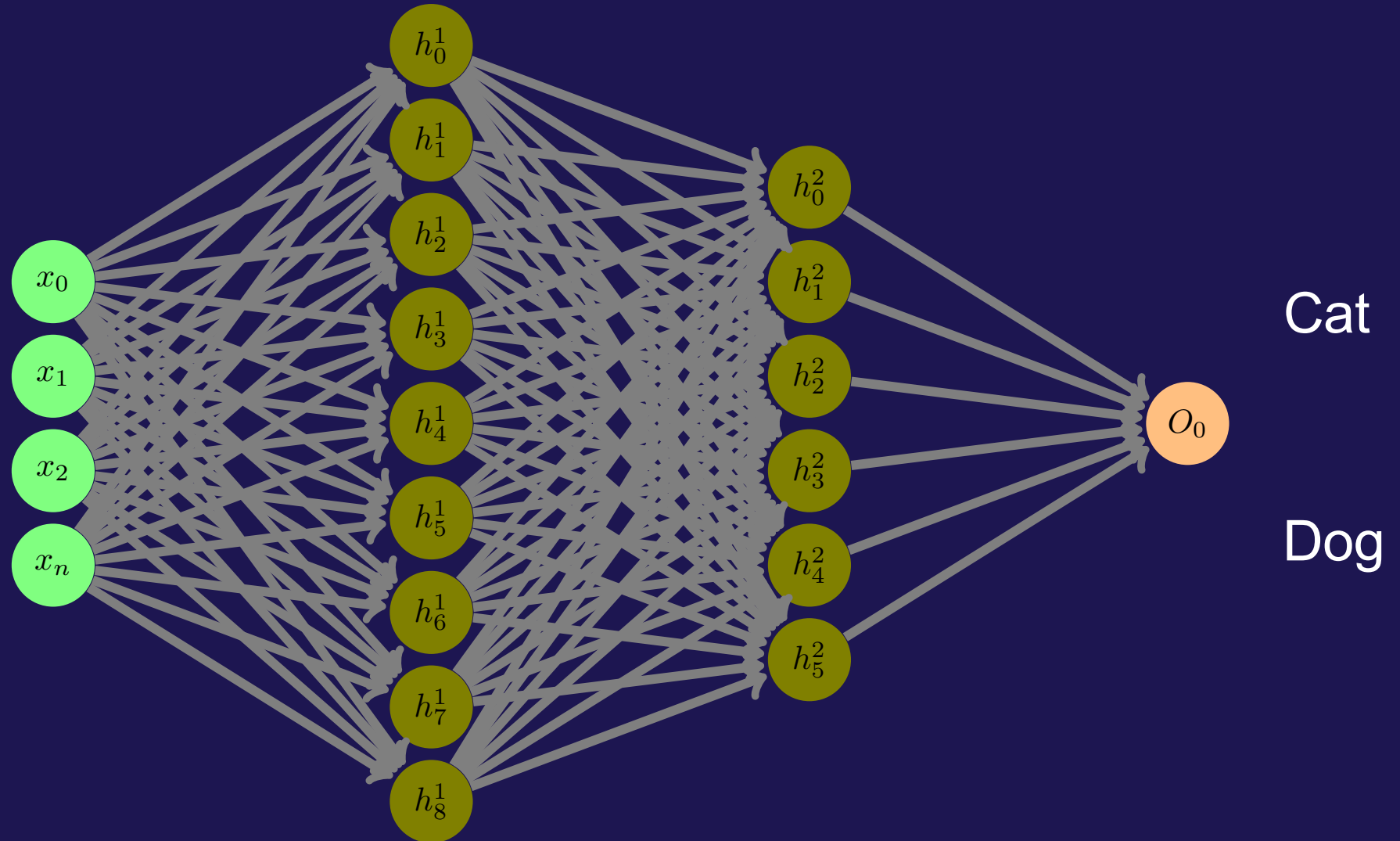
# Artificial neural networks

## Training



# Artificial neural networks

## Inference

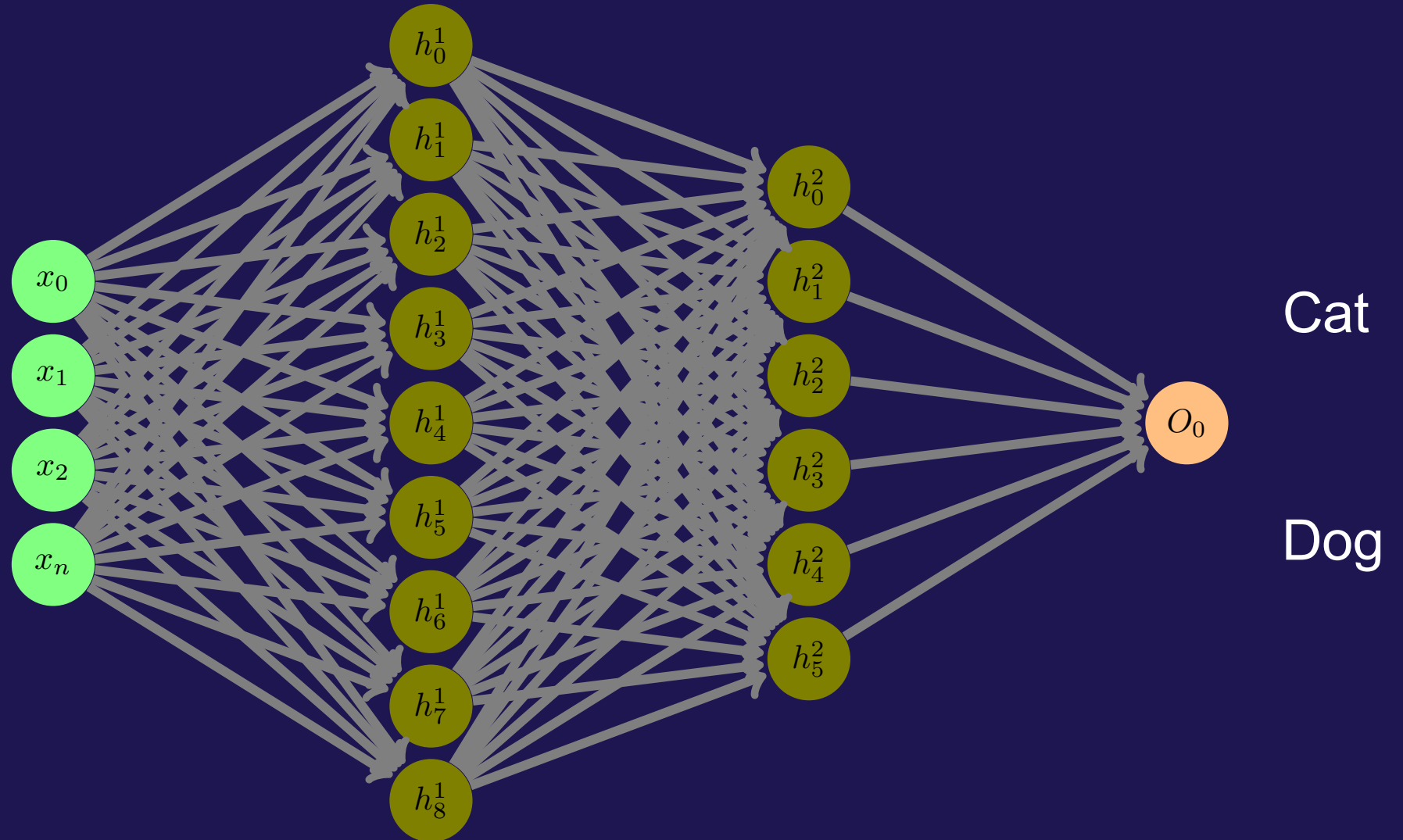


# Artificial neural networks

## Inference

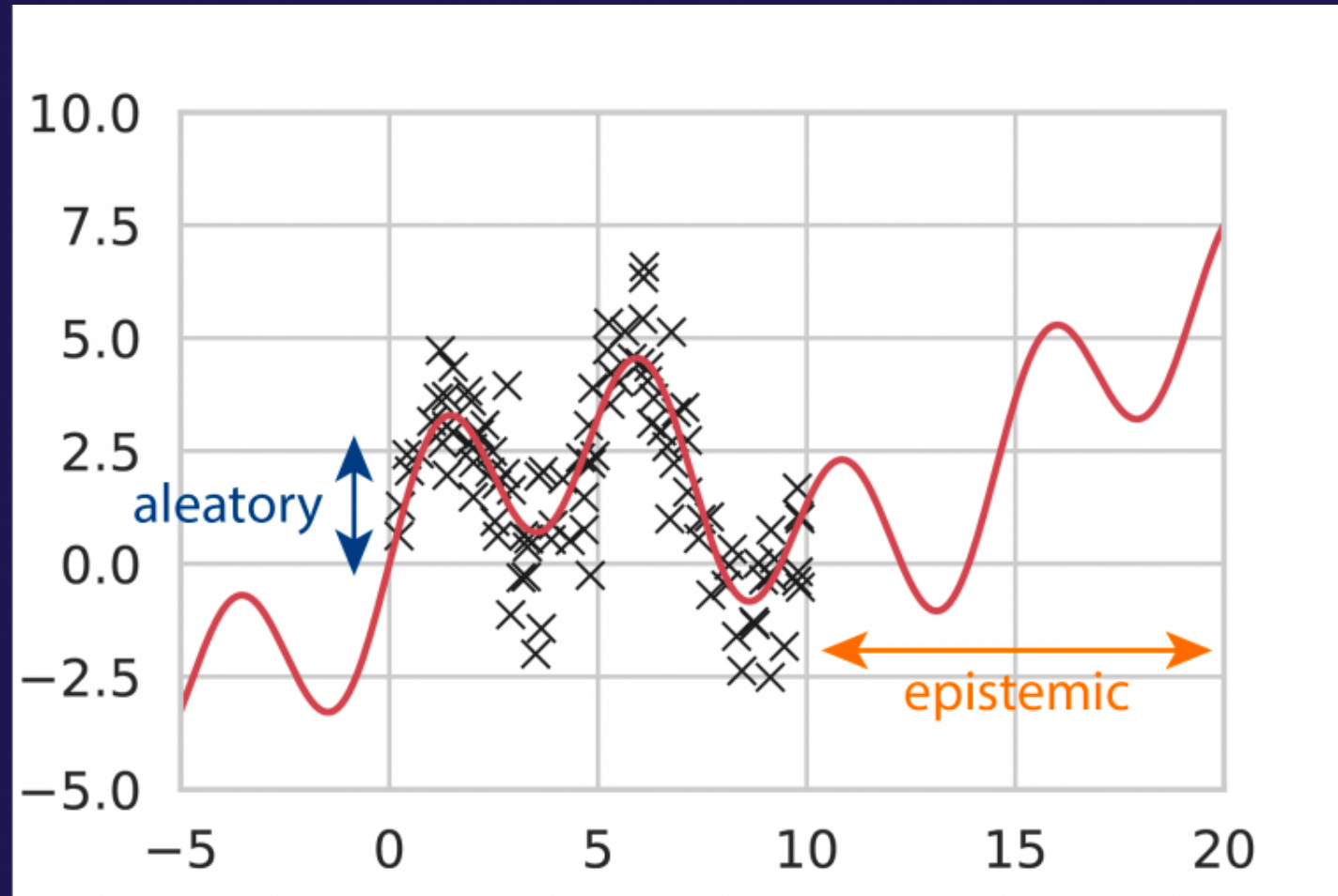


Out of distribution



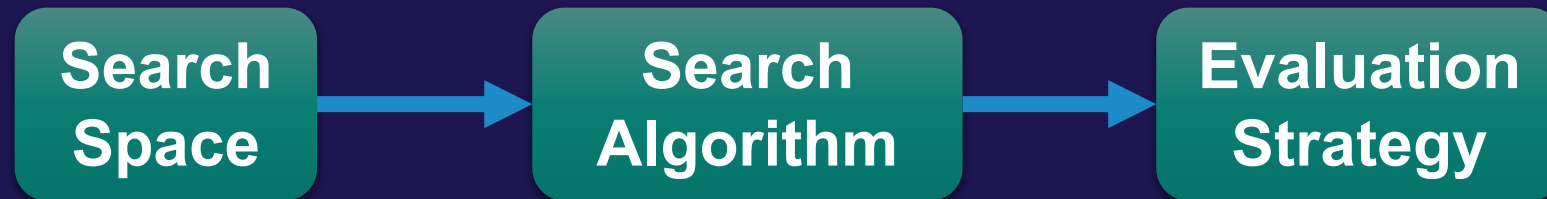


# Two major forms of uncertainty

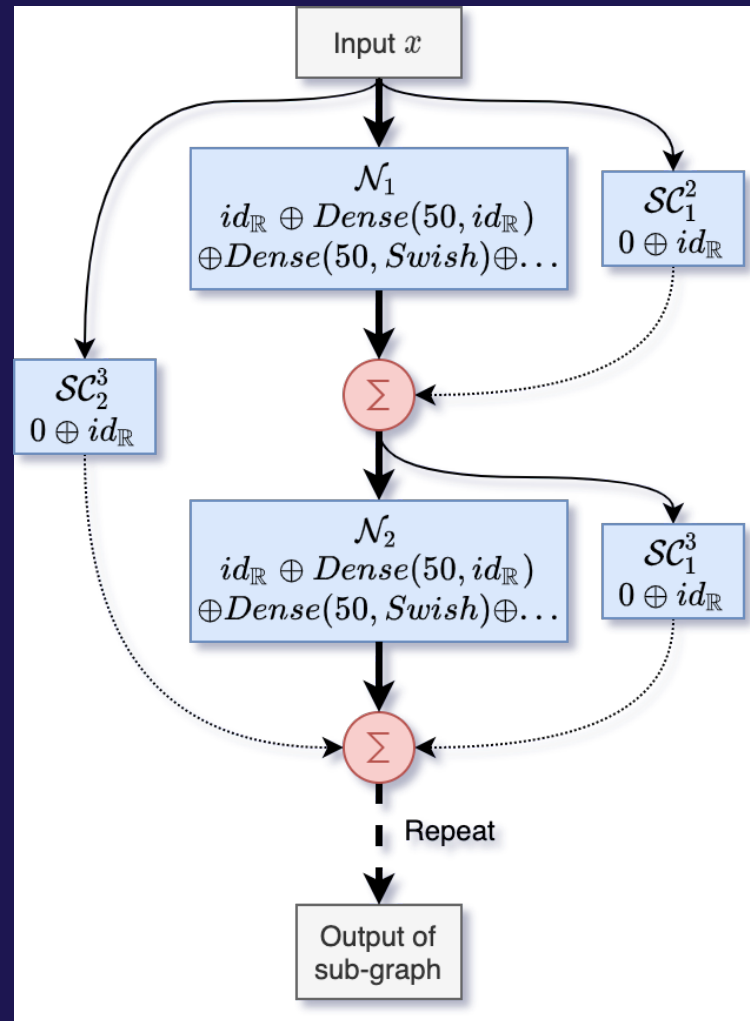


<https://www.inovex.de/de/blog/uncertainty-quantification-deep-learning/>

# Neural Architecture Search



# Neural Architecture Search Space

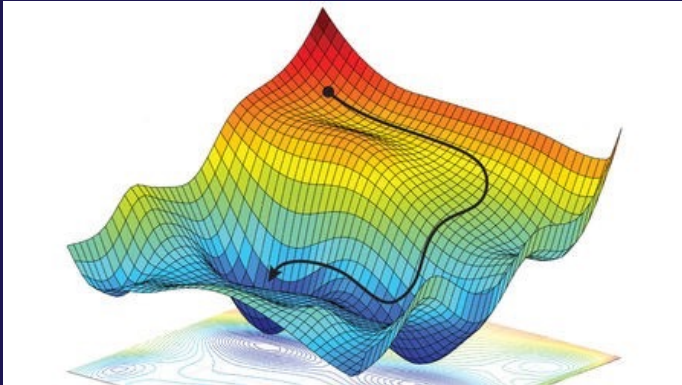


**WARNING:** Hyperparameters are kept constant in general



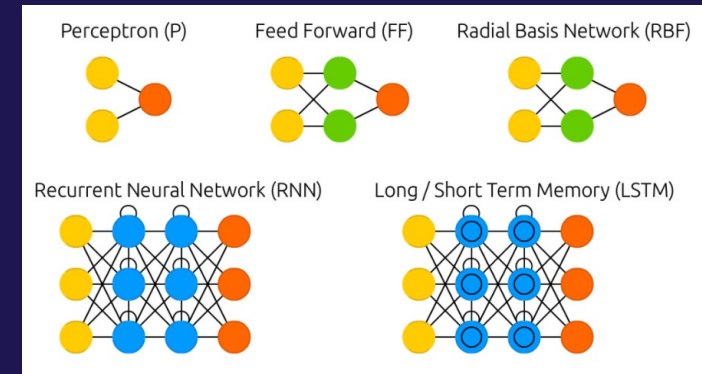
# Hyperparameters of Neural Networks

## Algorithm Hyperparameters



Optimizer: SGD, RMSprop, Adam...  
Learning rate  
Mini-batch size  
Learning rate scheduler  
Adaptative batch size  
...

## Model Hyperparameters



Number of layers  
Type of the layer: Fully Connected, Convolution, Recursive...  
Activation function  
Dropout rate  
Skip connection  
...

# Joint Hyperparameter and Architecture Search Problem

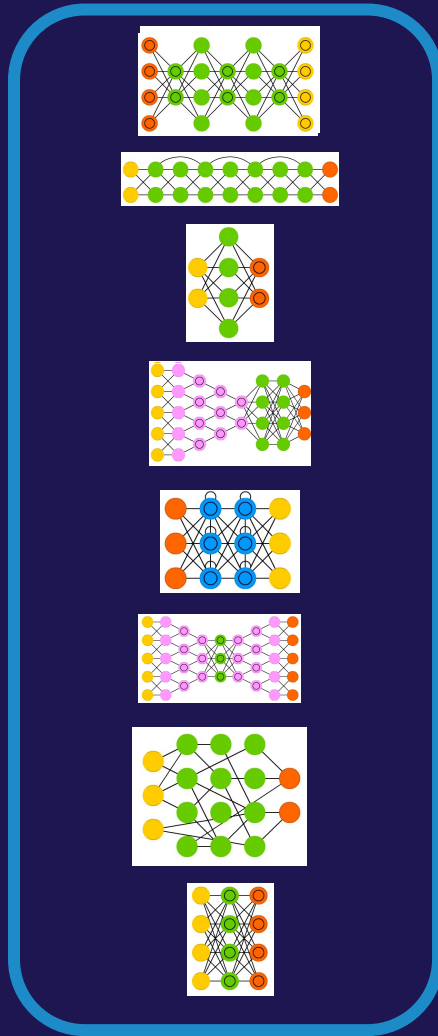
Lower-level problem: *Training data “T”*

$$\min_w \text{err}_T(a, h; T; w)$$

Upper-level problem: *Validation data “V”*

$$\min_{a, h} \text{err}_V(a, h; V; w^*)$$

# Selection of K models for Deep Ensembles with UQ



Catalogue



Ensemble

**Each neural network predicts a  
Probability distribution**

**The ensemble reflects a new mixture  
of distribution.**




# Decomposition of Ensemble Uncertainty

$$\sigma_{\mathcal{E}}^2(\mathbf{x}) = \overset{\text{aleatoric}}{E_{p(\theta)}[\sigma_{\theta}^2(\mathbf{x})]} + \overset{\text{epistemic}}{V_{p(\theta)}[\mu_{\theta}(\mathbf{x})]}$$

Learned and predicted by the neural networks

# Learn more about DeepHyper



Search the docs ...

**GET STARTED**

- Installations
- Tutorials**
- Notebooks
- Argonne LCF
- Research & Publications

**API REFERENCE**

- Core
- Ensemble
- Evaluator
- NAS
- Problem
- Search
- Sklearn

**DEVELOPER GUIDES**

←

## Tutorials

- **Notebooks**
  - 1. Hyperparameter Search for Machine Learning (Basic)
  - 2. Hyperparameter Search for Machine Learning (Advanced)
  - 3. Hyperparameter Search for Deep Learning (Basic)
  - 4. Neural Architecture Search (Basic)
  - 5. Automated Machine Learning with Scikit-Learn
- **Argonne LCF**
  - 1. Execution on the Theta supercomputer
  - 2. Execution on the ThetaGPU supercomputer

← previous  
**Analytics**

---

By Argonne  
© Copyright 2018-2021, Argonne.

<https://deephper.readthedocs.io>

# A Tutorial for NAS, Joint HPS + NAS, Deep Ensemble



## From Neural Architecture Search to Automated Deep Ensemble with Uncertainty Quantification

**GITHUB: [sdl\\_ai\\_workshop/03\\_distributedHyperOpt/02\\_NAS\\_and\\_more/](https://github.com/sdl-ai-workshop/03_distributedHyperOpt/02_NAS_and_more/)**

### Table of Contents

1. [Imports and GPU Detection](#)
2. [Start Ray](#)
3. [A Synthetic Dataset](#)
4. [Scaling the Data](#)
5. [Baseline Neural Network](#)
6. [Define the Neural Architecture Search Space](#)
7. [Define the Neural Architecture Search Optimization Problem](#)
8. [Define the Evaluator Object](#)
9. [Define and Run the Neural Architecture Search](#)
10. [Adding Uncertainty Quantification to the Baseline Neural Network](#)
11. [Ensemble of Neural Networks With Random Initialization](#)
12. [AutoDEUQ: Automated Deep Ensemble with Uncertainty Quantification](#)