ACES: Intermediate Python Programming In JupyterLab

Accelerating Workflows on a Composable Cyberinfrastructure

Zhenhua He

07/14/2024

High Performance
Research Computing
DIVISION OF RESEARCH

ACES TechLab

Lab I. JupyterLab (30 mins)

We will load required modules and activate virtual environment and run JupyterLab on HPRC ACES portal.

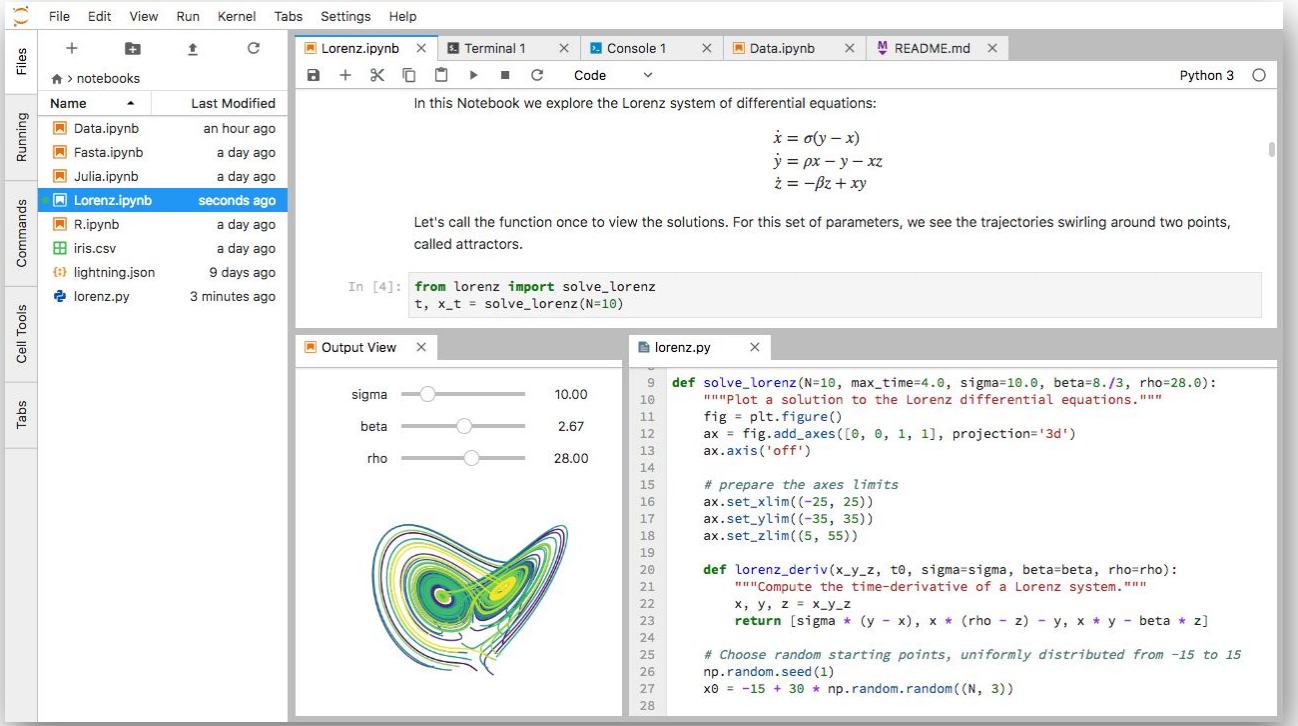
Lab II. Data Exploration with Python(30 mins)

We will go through some examples with a popular Python library Pandas for data exploration.

Lab III. Machine/Deep Learning (30 minutes)

We will learn how to use PyTorch to build and train a simple image classification model with deep neural network (DNN) on GPU.

Lab I. JupyterLab



L1 - Resources

- Texas A&M High Performance Research Computing (HPRC)
- HPRC Microcredentials and Courses
- ACES Quick Start Guide
- ACES Portal (ACCESS)
- ACCESS Documentation
- HPRC YouTube Channel
- help@hprc.tamu.edu

NSF ACES

Accelerating Computing for Emerging Sciences

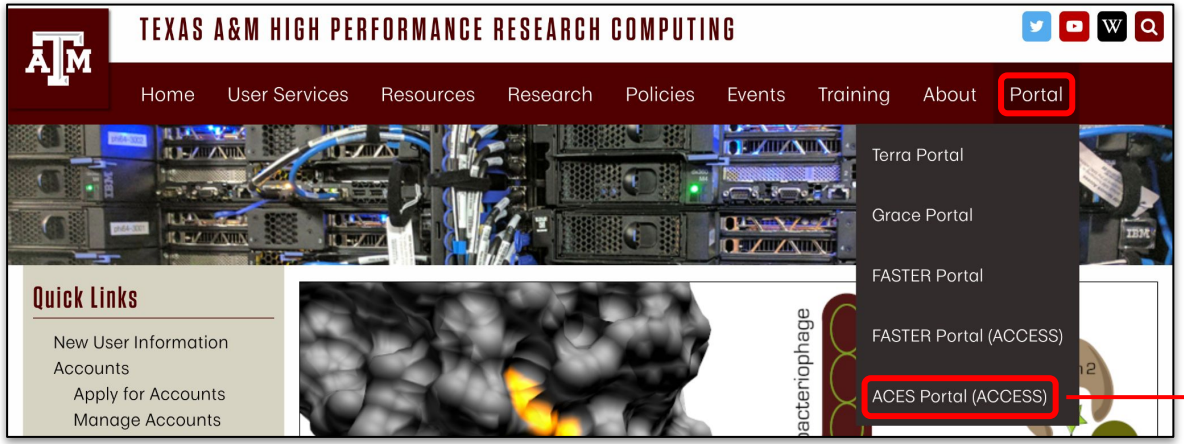
Our Mission:

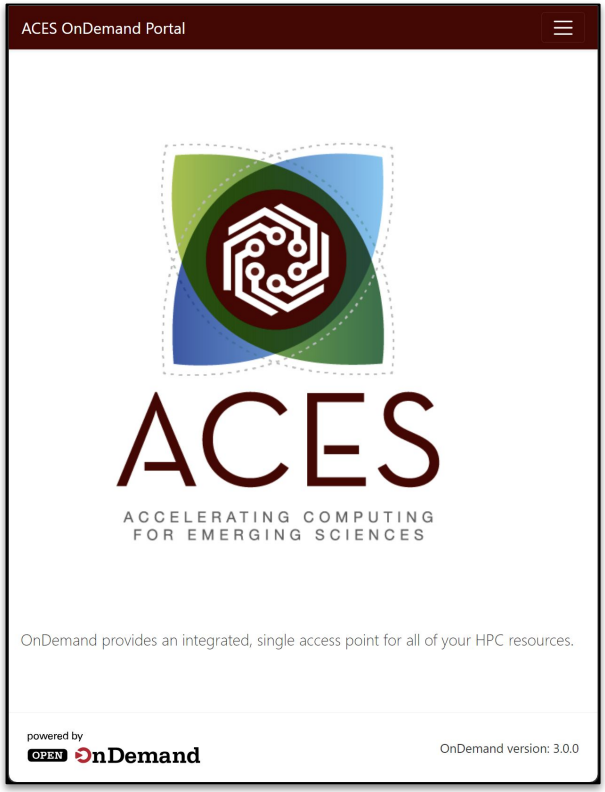
- Offer an accelerator testbed for numerical simulations and AI/ML workloads
- Provide consulting, technical guidance, and training to researchers
- Collaborate on computational and data-enabled research.

ACES Accelerators

Component	Quantity	Description
Graphcore IPU	32	16 Colossus GC200 IPUs, 16 Bow IPUs. Each IPU group hosted with a CPU server as a POD16 on a 100 GbE RoCE fabric
Intel PAC D5005 FPGA	2	Accelerator with Intel Stratix 10 GX FPGA and 32 GB DDR4
BittWare IA-840F FPGA	2	Accelerator with Agilex AGF027 FPGA and 64 GB of DDR4
NextSilicon Coprocessor	2	Reconfigurable accelerator with an optimizer continuously evaluating application behavior.
NEC Vector Engine	8	Vector computing card (8 cores and HBM2 memory)
Intel Optane SSD	48	18 TB of Intel Optane SSDs addressable as memory w/ MemVerge Memory Machine.
NVIDIA H100 + A30	30 + 4	NVIDIA GPUs for HPC, DL Training, AI Inference
Intel PVC	110	Software Development Platform for PVC

ACES Portal



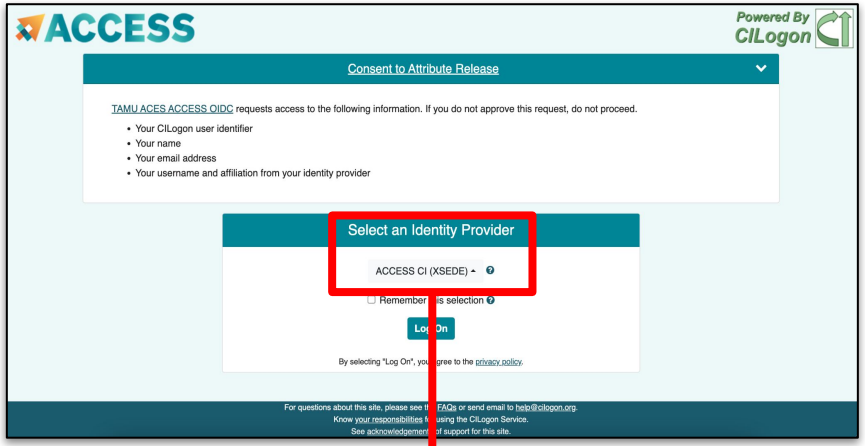


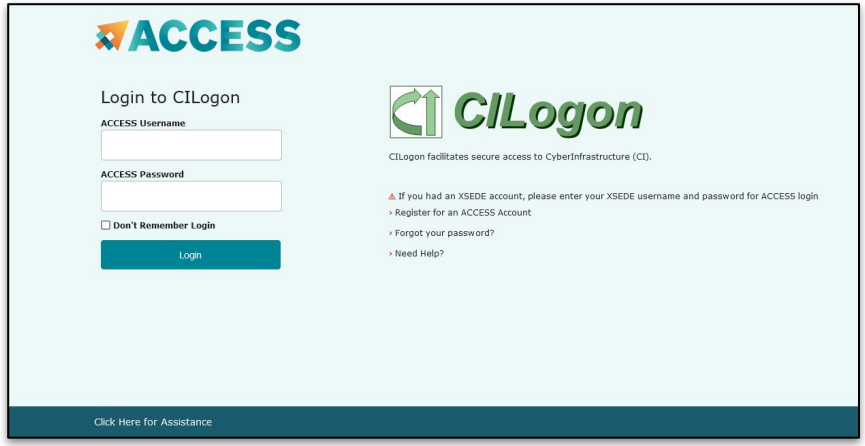
ACES Portal portal-aces.hprc.tamu.edu
is the web-based user interface for the ACES cluster

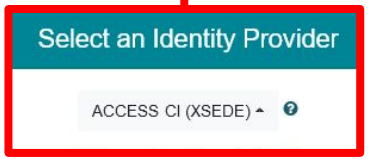
Open OnDemand (OOD) is an advanced web-based graphical interface framework for HPC users

Authentication via CILogon

Log-in using your ACCESS CI credentials.



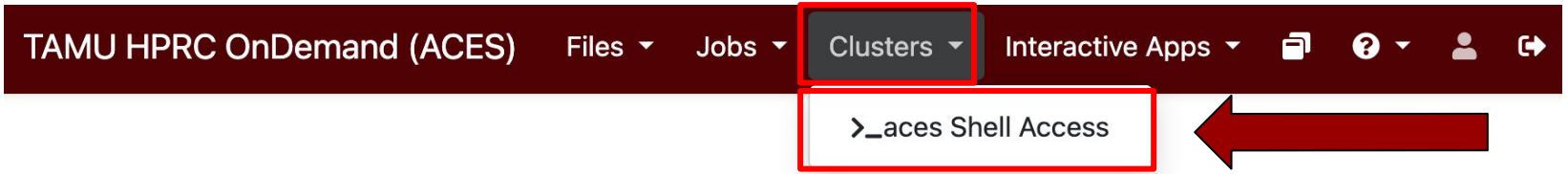




Select the Identity Provider appropriate for your account.

Get a Shell on ACES

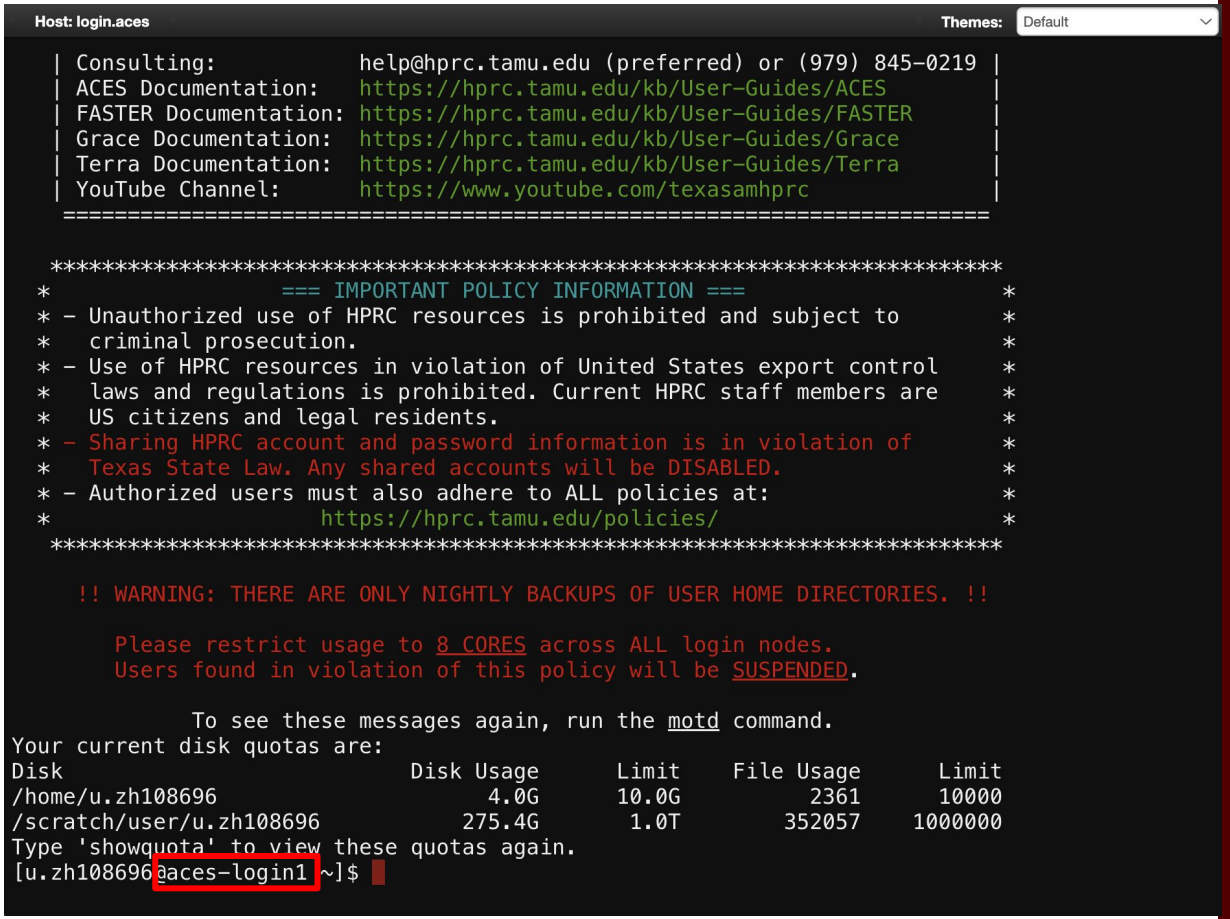
Click on “Clusters” menu → _aces Shell Access



Success!

Welcome to the ACES login node.

Check which login node you are on.



Commands to copy the materials

- Navigate to your personal scratch directory

  ```
  $ cd $SCRATCH
  ```

- Files for this course are located at

  `/scratch/training/nh_wkshop`

  Make a copy in your personal scratch directory

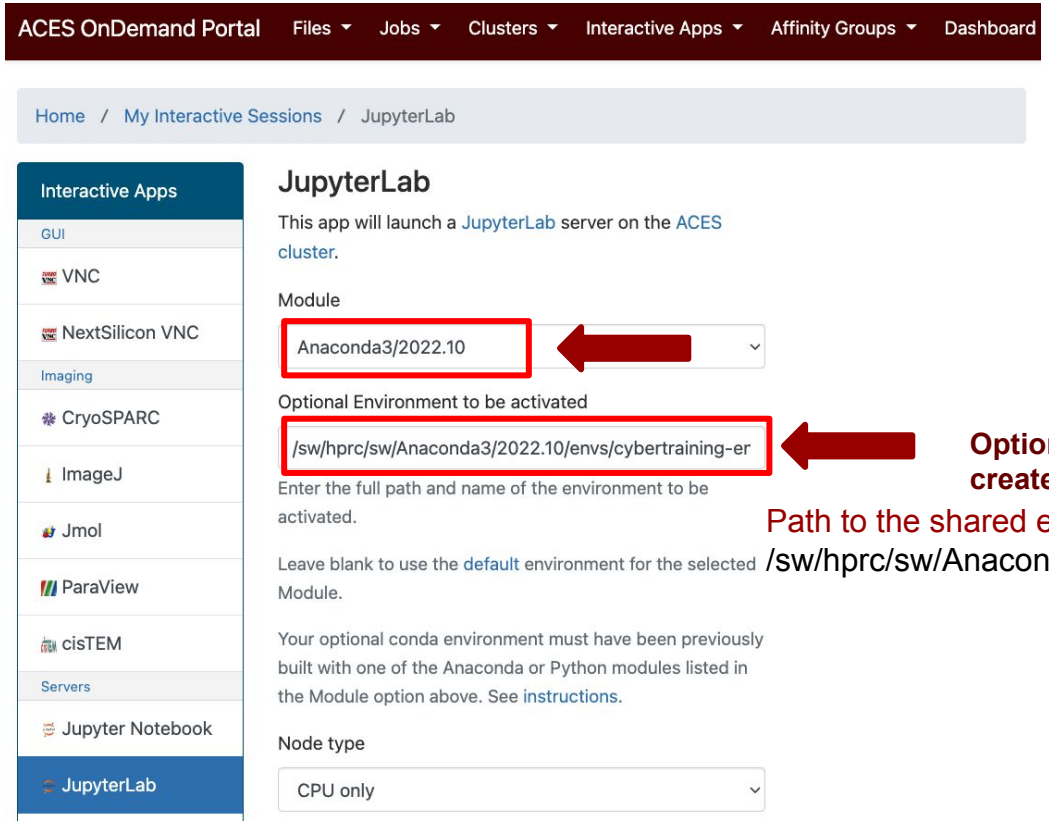
  ```
  $ cp -r /scratch/training/nh_wkshop $SCRATCH
  ```

- Enter this directory (your local copy)

  ```
  $ cd nh_wkshop
  ```

Go to JupyterLab Page

JupyterLab Page

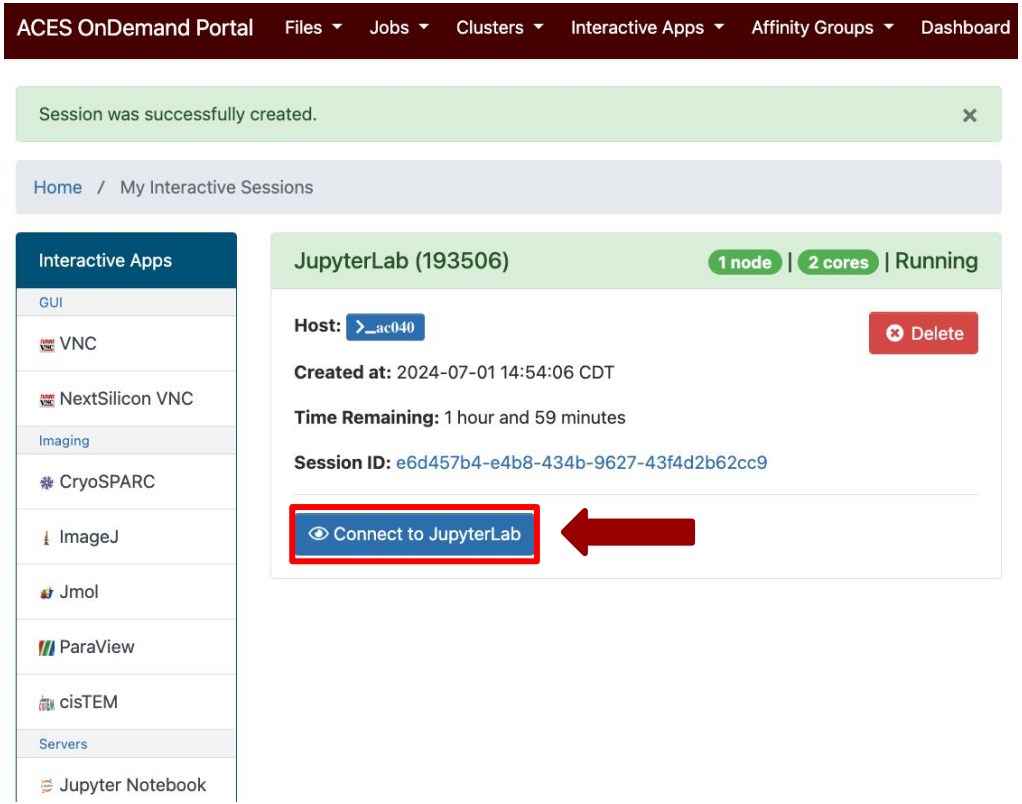


Option 1: Use a shared environment created by TAMU HPRC for this workshop

Path to the shared environment:
/sw/hprc/sw/Anaconda3/2022.10/envs/cybertraining-env

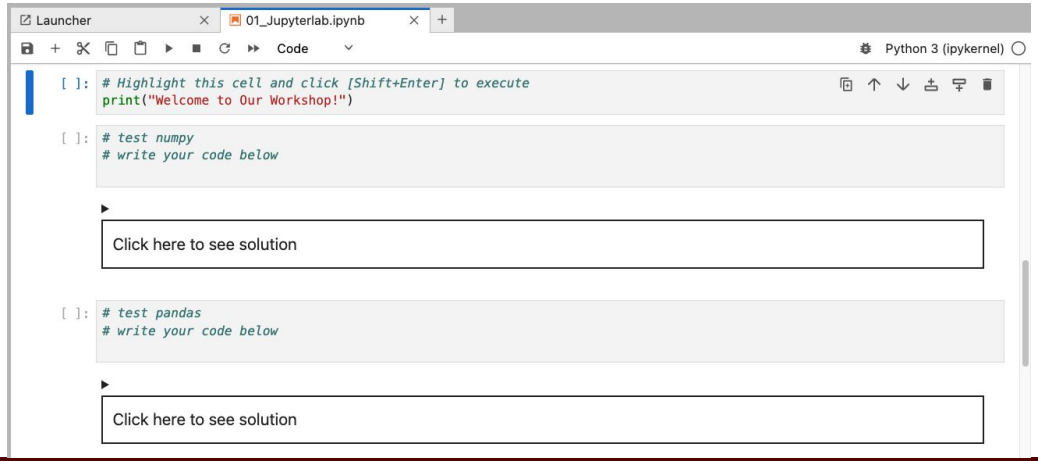
Other fields:
Node Type: CPU only
Number of hours: 2
Number of cores: 2
Total memory (GB): 5

Connect to JupyterLab

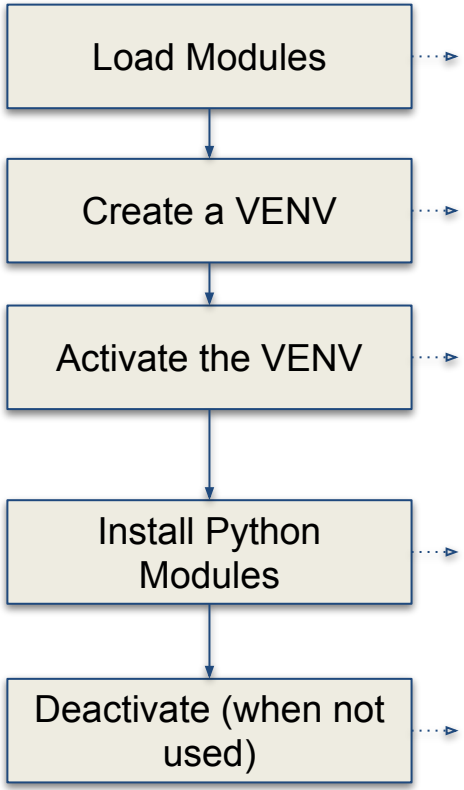


Review and Exercise

- Log into ACES through ACES Portal (ACCESS)
- Copy the training materials to your $SCRATCH directory
- Launch JupyterLab app
- In the notebook named *01_Jupyterlab.ipynb*, follow the instructions to import the required modules to make sure they have been loaded properly.

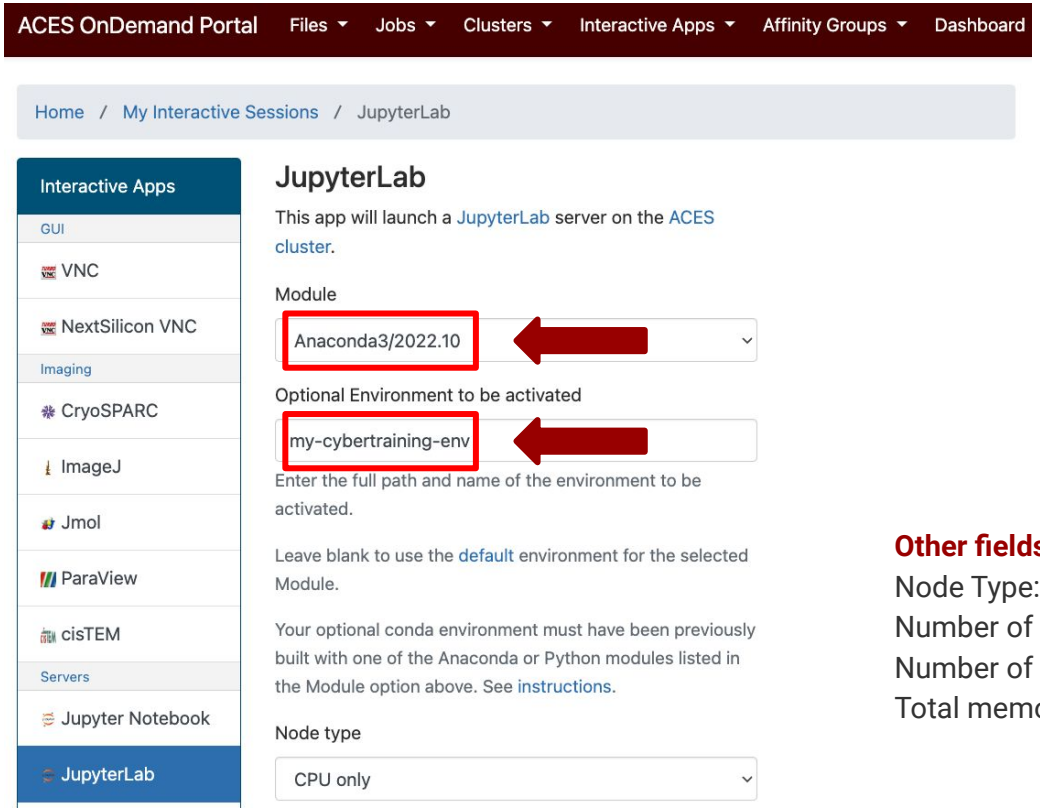


Option 2




```
# clean up and load Anaconda
cd $SCRATCH
module purge
module load Anaconda3/2022.10

# create a Python virtual environment
conda create -n my-cybertraining-env

# activate the virtual environment
source activate my-cybertraining-env

# install required package to be used in the portal
conda install -c conda-forge jupyterlab
conda install -c conda-forge numpy
conda install -c conda-forge pandas
conda install -c conda-forge xarray geopandas folium
conda install -c conda-forge netcdf4
(install other packages as well ...)

# deactivate the virtual environment
# source deactivate
```

JupyterLab Page



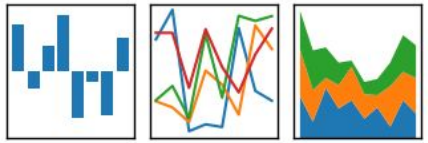
Other fields:

Node Type: CPU only
Number of hours: 2
Number of cores: 2
Total memory (GB): 5

Lab II. Data Exploration

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



matplotlib

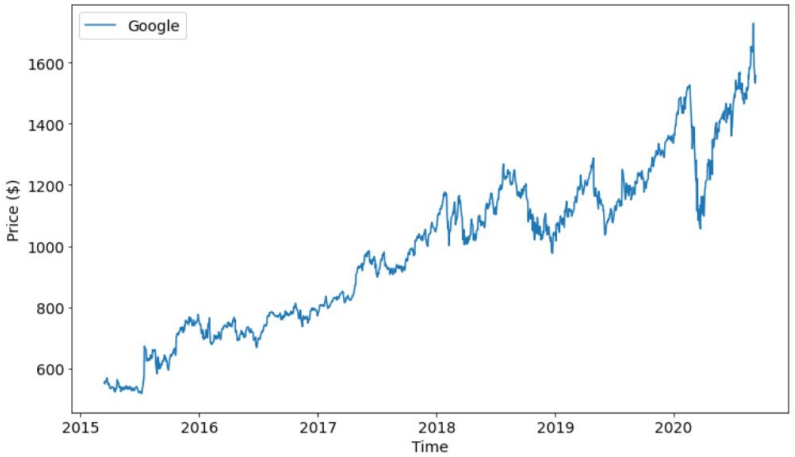
Data Structures

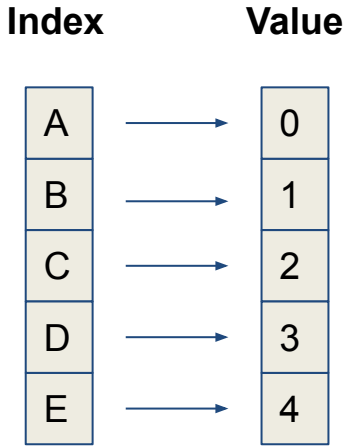
Pandas has two data structures that are descriptive and optimized for data with different dimensions.

- **Series:** 1D labeled array
- **DataFrame:** General 2D labeled, size-mutable tabular structure with potentially heterogeneously-typed columns

Series in pandas

- One-dimensional labeled array
- Capable of holding any data type (integers, strings, floating point numbers, etc.)
- Example: time-series stock price data

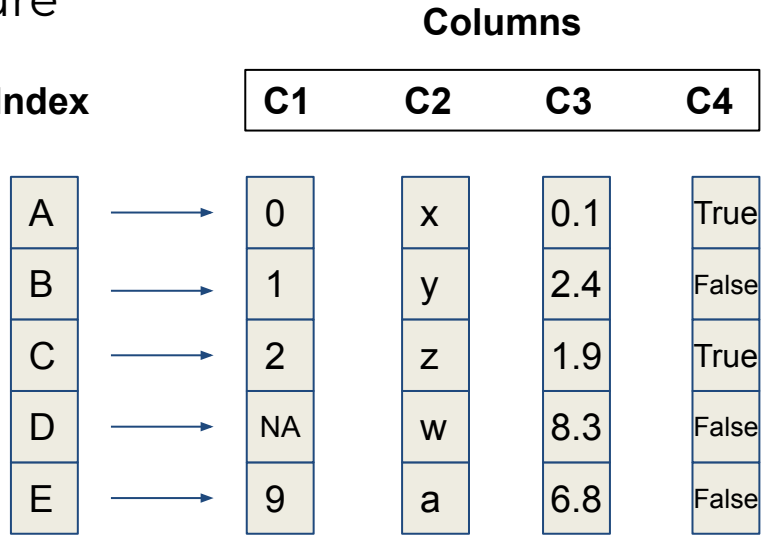




DataFrame in pandas

- Primary Pandas data structure
- A dict-like container for Series objects
- Two-dimensional size-mutable
- Heterogeneous tabular data structure

DATE and TIME (UTC)	LAT	LON	MAG	DEPTH km
01-JUL-2024 17:10:59	13.97	52.28	4.6	10
01-JUL-2024 15:22:40	-5.51	147.18	4.5	192
01-JUL-2024 13:04:10	-32.63	-69.34	4.3	10
01-JUL-2024 11:54:40	-61.97	155.22	4.6	10
01-JUL-2024 11:37:41	-23.41	-68.10	4.5	126
01-JUL-2024 10:20:59	37.86	73.24	4.6	10
01-JUL-2024 10:16:23	-5.56	153.59	5.0	42
01-JUL-2024 05:53:52	10.79	125.37	4.8	95
01-JUL-2024 05:22:23	10.74	125.39	4.8	68
01-JUL-2024 05:12:03	14.74	147.06	5.0	10
01-JUL-2024 02:38:03	-62.68	165.56	5.1	10



(source: https://ds.iris.edu/seismon/eventlist/index.phtml)

Pandas Learning Objectives

After this section, you will learn:

- DataFrame building
- DataFrame operations
  - Relabeling
  - Data grouping
- Data handling
  - Handle missing data
  - Handle duplicate data
  - Merege DataFrames

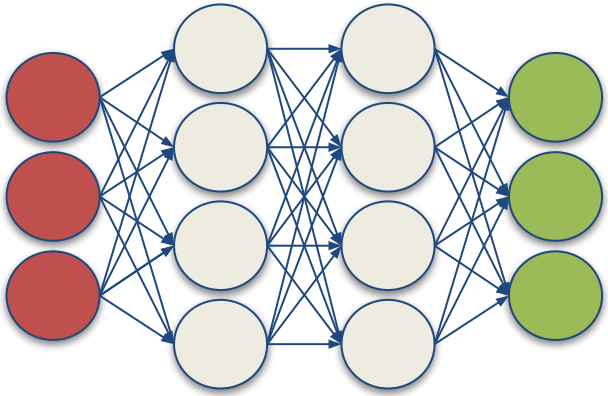
JupyterLab Exercises

Lab III. Machine/Deep Learning

Deep Learning
by Ian Goodfellow, Yoshua Bengio, and Aaron Courville
http://www.deeplearningbook.org/

Animation of Neutron Networks
by Grant Sanderson
https://www.3blue1brown.com/

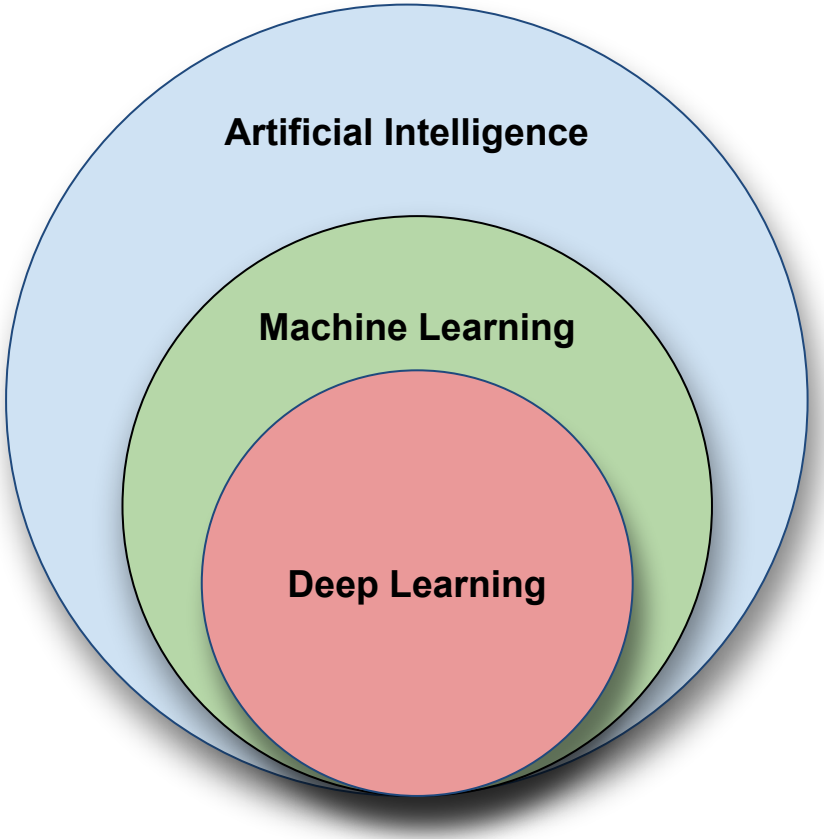
Visualization of CNN
by Adam Harley
https://adamharley.com/nn_vis/cnn/3d.html



TensorFlow PyTorch

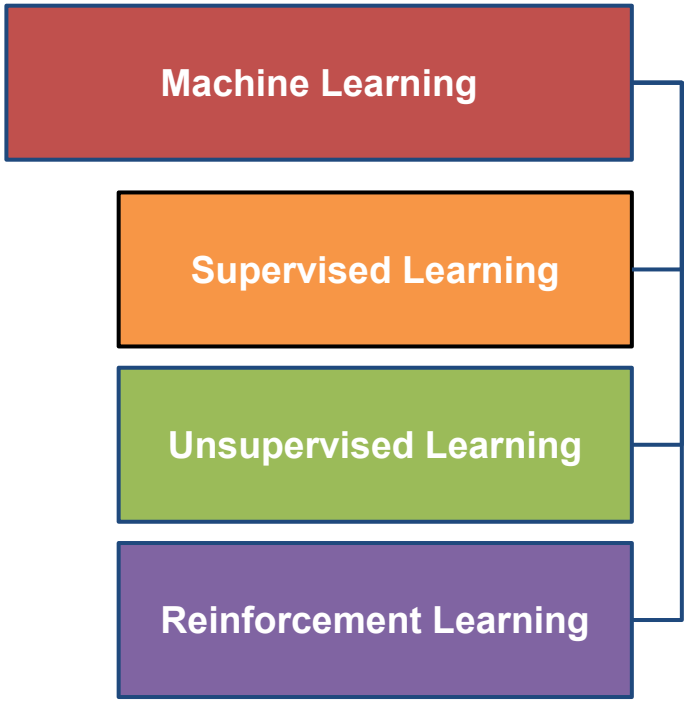
Relationship of AI, ML, and DL

- **Artificial Intelligence (AI)** is anything about man-made intelligence exhibited by machines.
- **Machine Learning (ML)** is an approach to achieve **AI**.
- **Deep Learning (DL)** is one technique to implement **ML**.

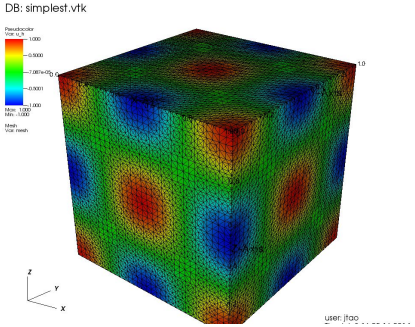


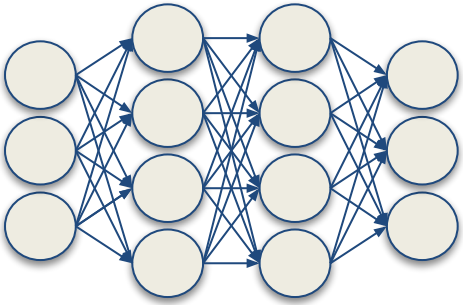
Types of ML Algorithms

- **Supervised Learning**
  - trained with labeled data; including regression and classification problems
- **Unsupervised Learning**
  - trained with unlabeled data; clustering and association rule learning problems.
- **Reinforcement Learning**
  - no training data; stochastic Markov decision process; robotics and business strategy planning.

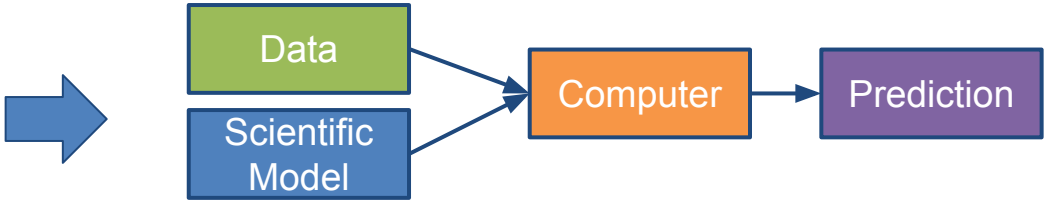


Machine Learning

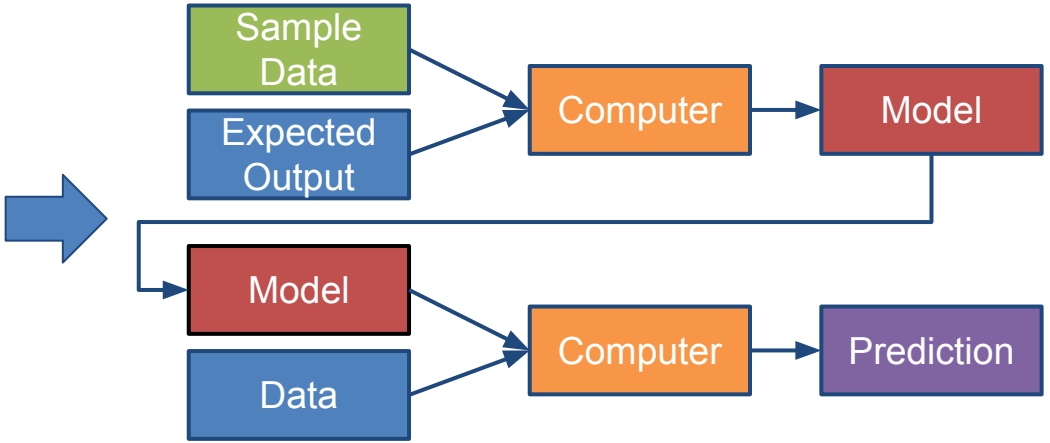




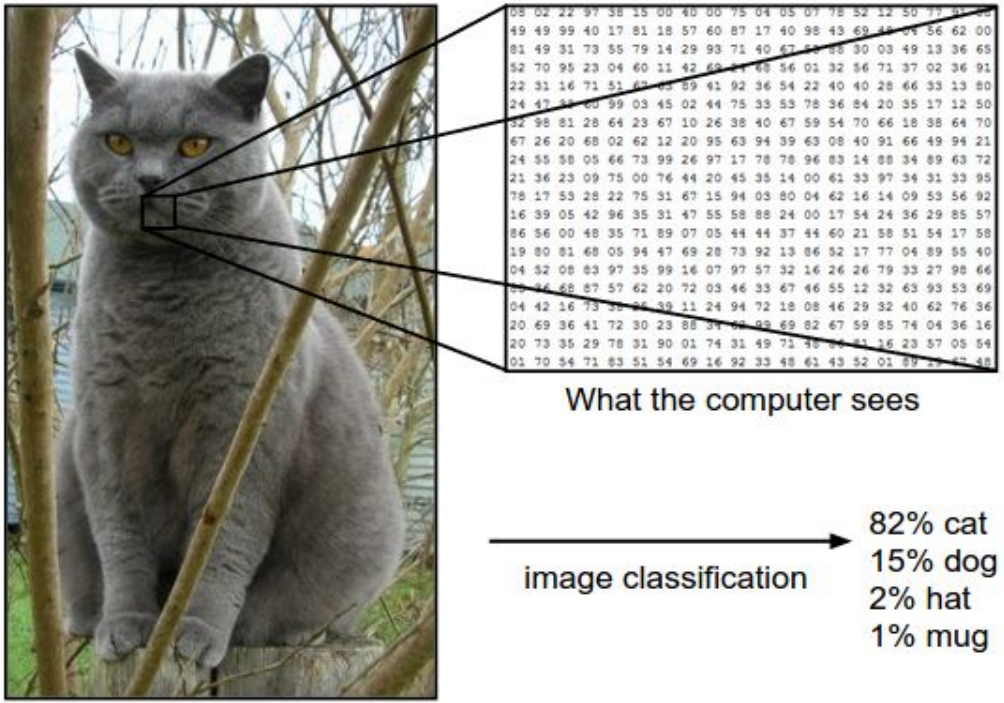
Traditional Modeling



Machine Learning (Supervised Learning)



Inputs and Outputs



What the computer sees

image classification

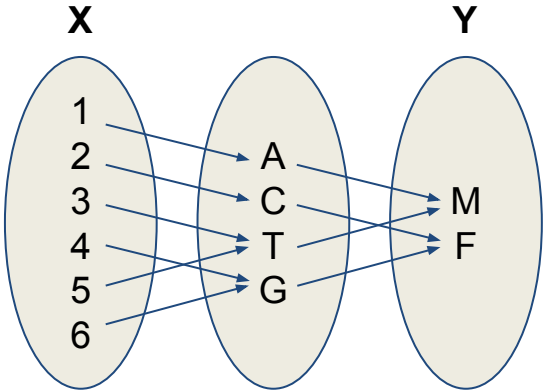
82% cat
15% dog
2% hat
1% mug

Image from the Stanford CS231 Course

256 X 256 Matrix

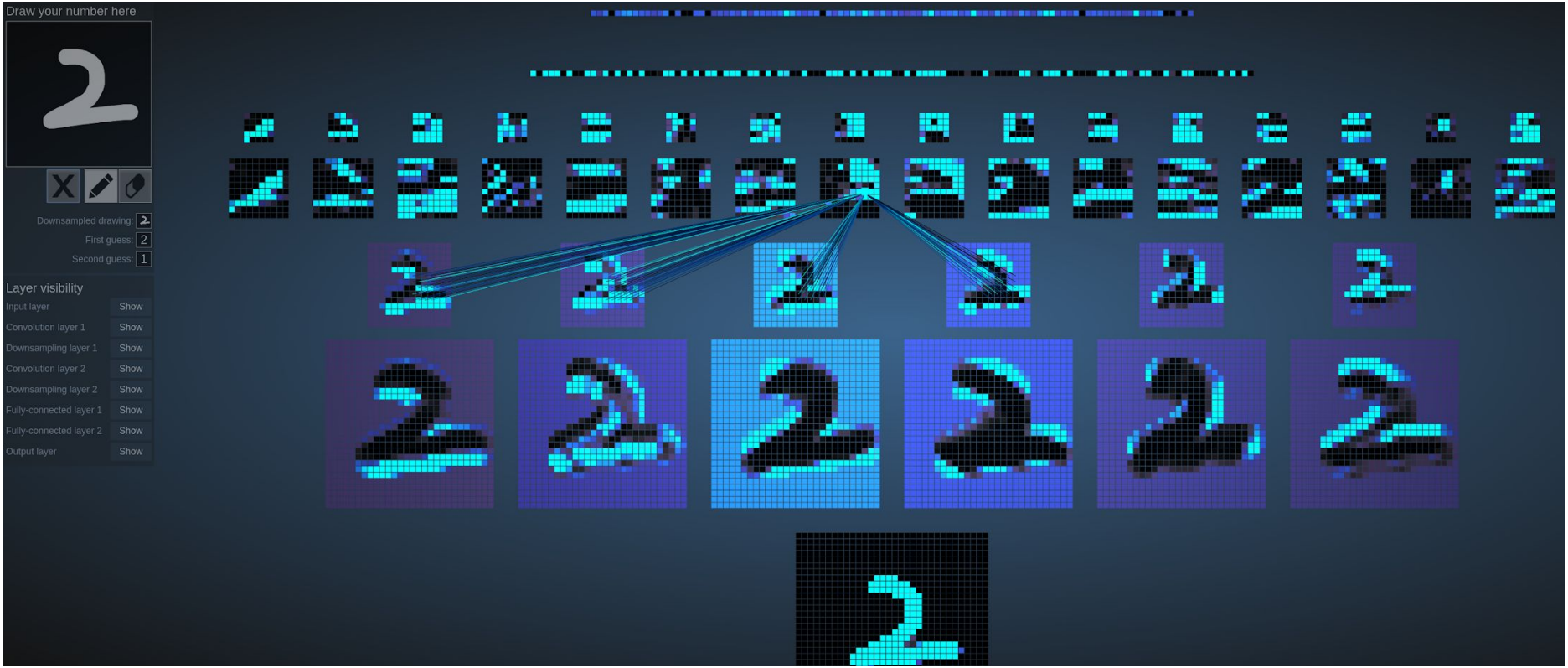
DL model

4-Element Vector



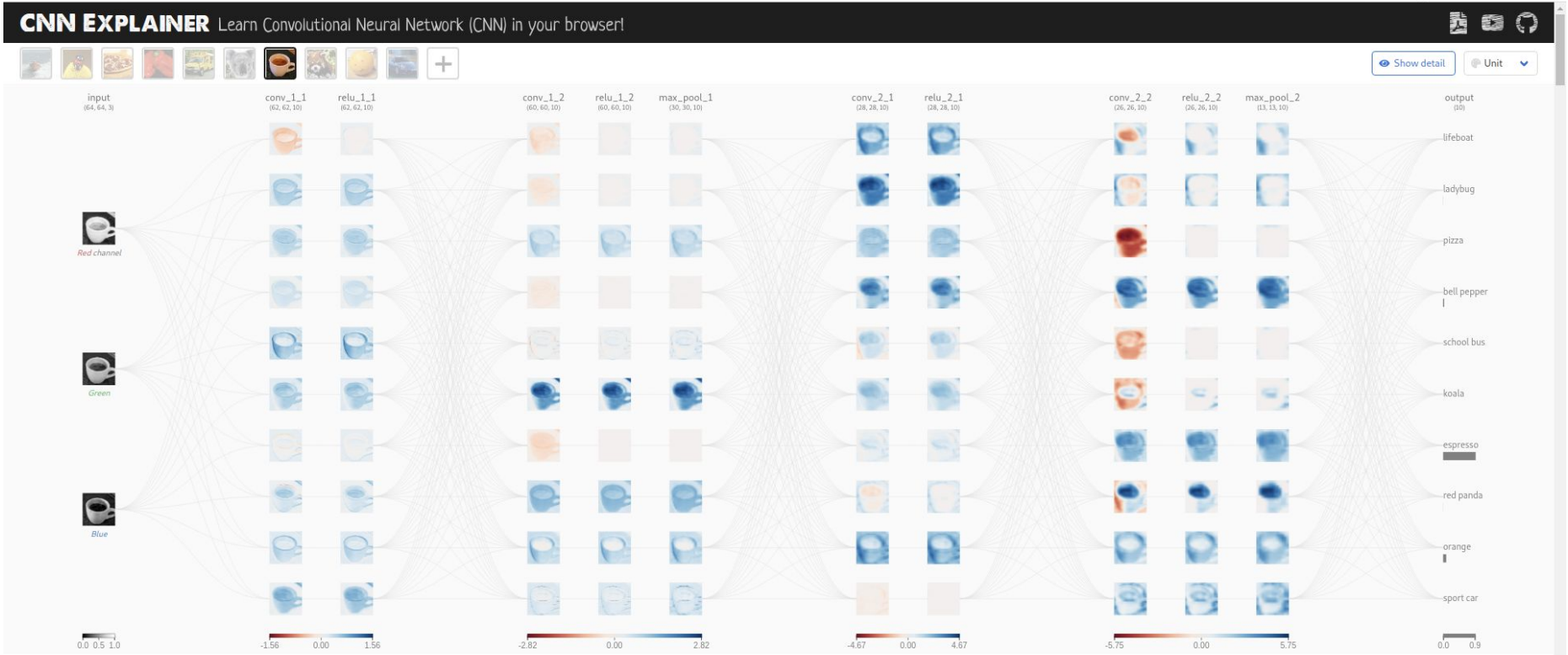
With deep learning, we are searching for a **surjective** (or **onto**) function **f** from a set **X** to a set **Y**.

MNIST - CNN Visualization



(Image Credit: https://adamharley.com/nn_vis/cnn/3d.html)

CNN Explainer



(Image Credit: https://poloclub.github.io/cnn-explainer/)

JupyterLab Exercises

High Performance Research Computing

DIVISION OF RESEARCH

https://hprc.tamu.edu

HPRC Helpdesk:

help@hprc.tamu.edu

Phone: 979-845-0219

Help us help you. Please include details in your request for support, such as, Cluster (ACES, FASTER, Grace, Terra, ViDaL), NetID (UserID), Job information (Job id(s), Location of your jobfile, input/output files, Application, Module(s) loaded, Error messages, etc), and Steps you have taken, so we can reproduce the problem.