ACES: AI/ML TechLab

Accelerating AI/ML Workflows on a Composable Cyberinfrastructure

Zhenhua He

10/03/2023

High Performance
Research Computing
DIVISION OF RESEARCH

Acknowledgements

This short course is sponsored in part by the CSSN Community Engagement Program Travel Award. We gratefully acknowledge their support in providing this community resource.

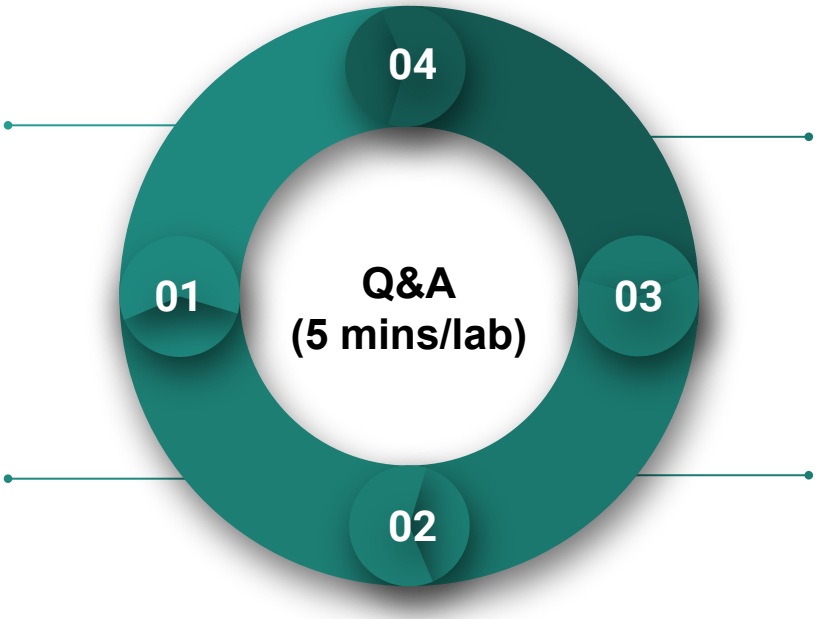
AI/ML TechLab

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 (30 mins)

We will go through some examples with two popular Python libraries: Pandas and Matplotlib for data exploration.



Lab IV. Deep Learning (30 minutes)

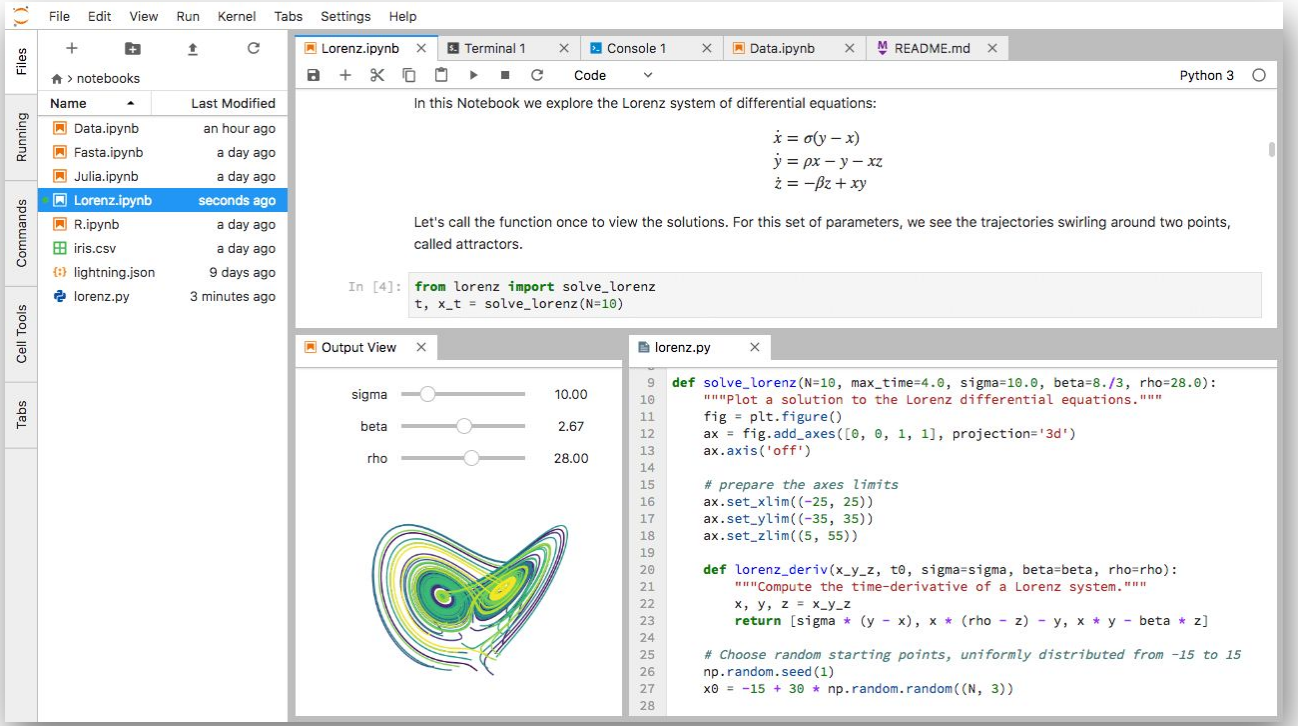
We will learn how to use PyTorch to build and train a simple image classification model with deep neural network (DNN).

Lab III Machine Learning (30 minutes)

We will learn to use scikit-learn library for linear regression and classification applications.

Figure 1. Structure of the AI/ML TechLab.

Lab I. JupyterLab



L1 - Resources

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

NSF ACES

Accelerating Computing for Emerging Sciences

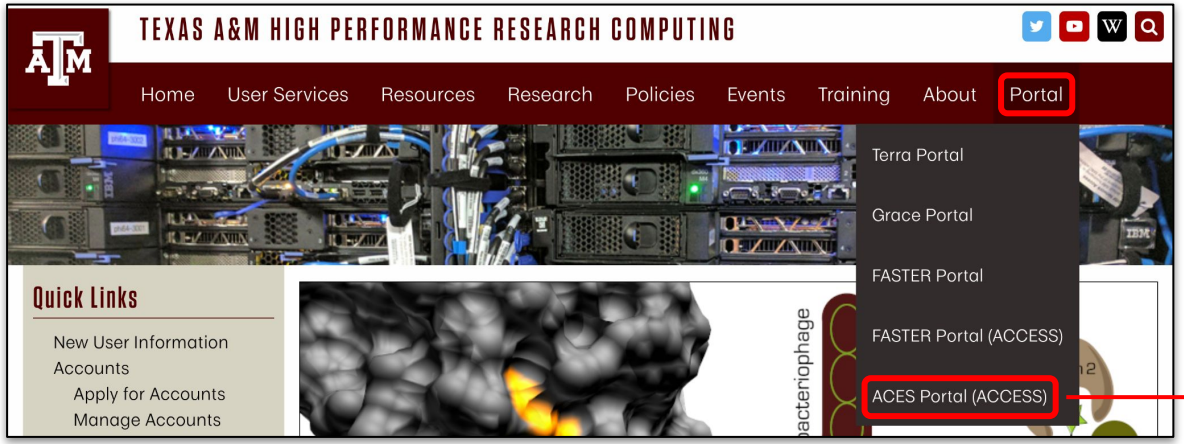
Our Mission:

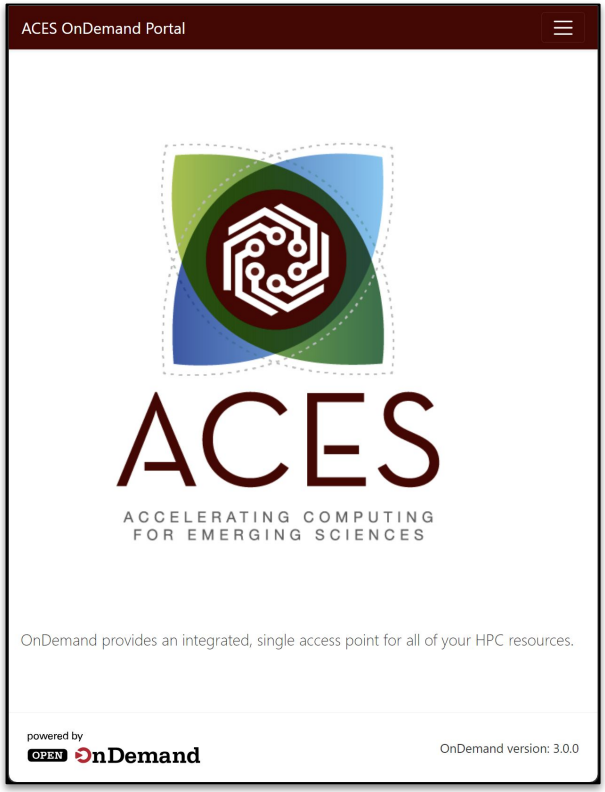
- Offer an accelerator testbed for numerical simulations and **AI/ML workloads**
- Provide consulting, technical guidance, and training to researchers
- Collaborate on computational and data-enabled research.

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 + ATS-P	12 + 22	Software Development Platform for PVC

ACES Portal



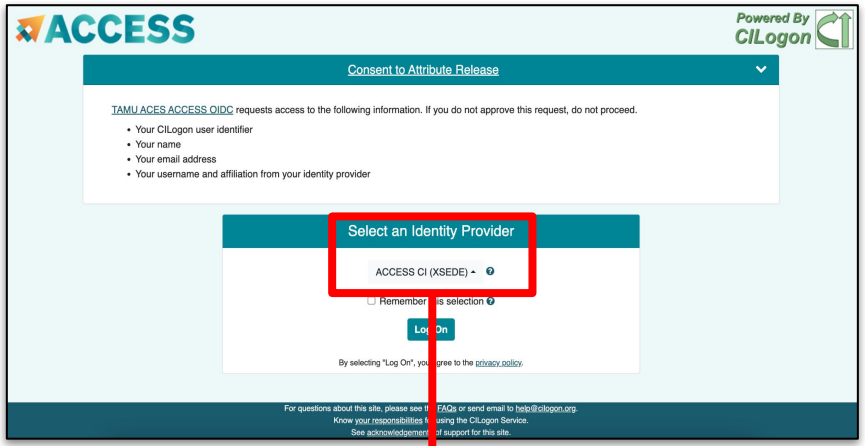


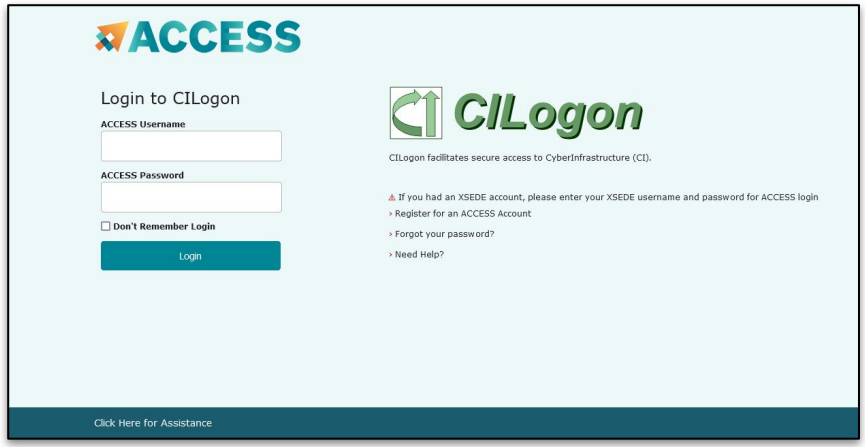
ACES Portal portal-aces.hprc.tamu.edu
is the web-based user interface for the ACES cluster

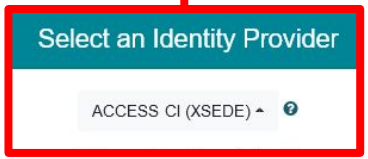
Open OnDemand (OOD) is an advanced web-based graphical interface framework for HPC users

Authentication via CILogon

Log-in using your ACCESS CI credentials.



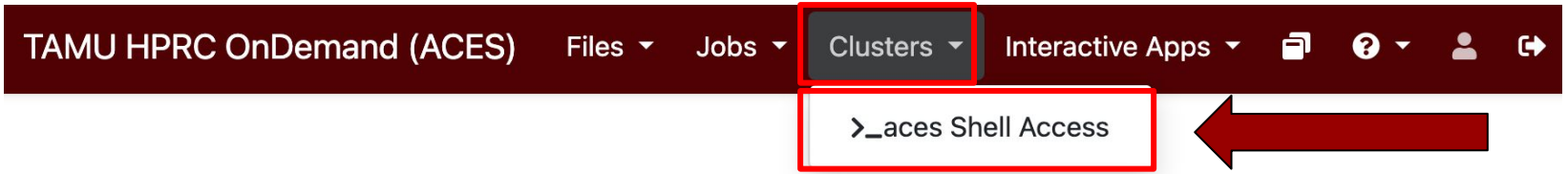




Select the Identity Provider appropriate for your account.

Get a Shell on ACES

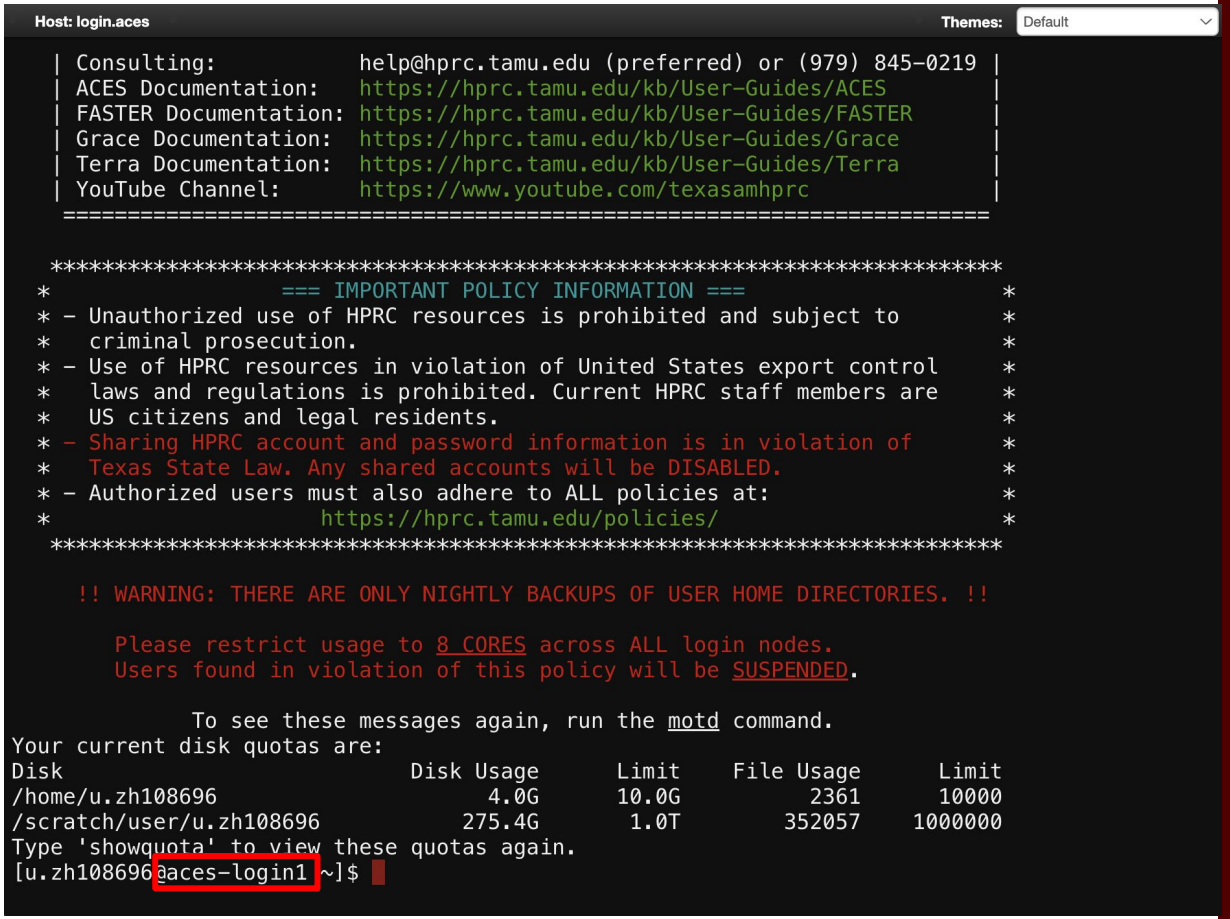
Click on “Clusters” menu → _aces Shell Access



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/ai_tech_labs
  ```

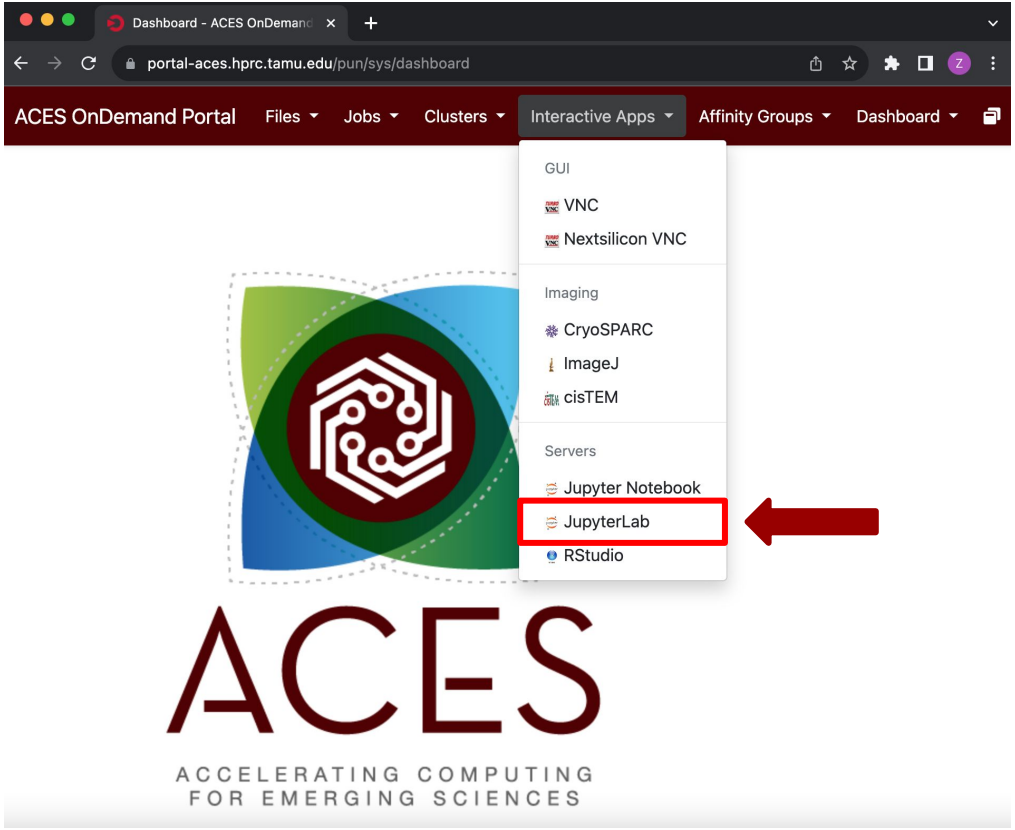
  Make a copy in your personal scratch directory

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

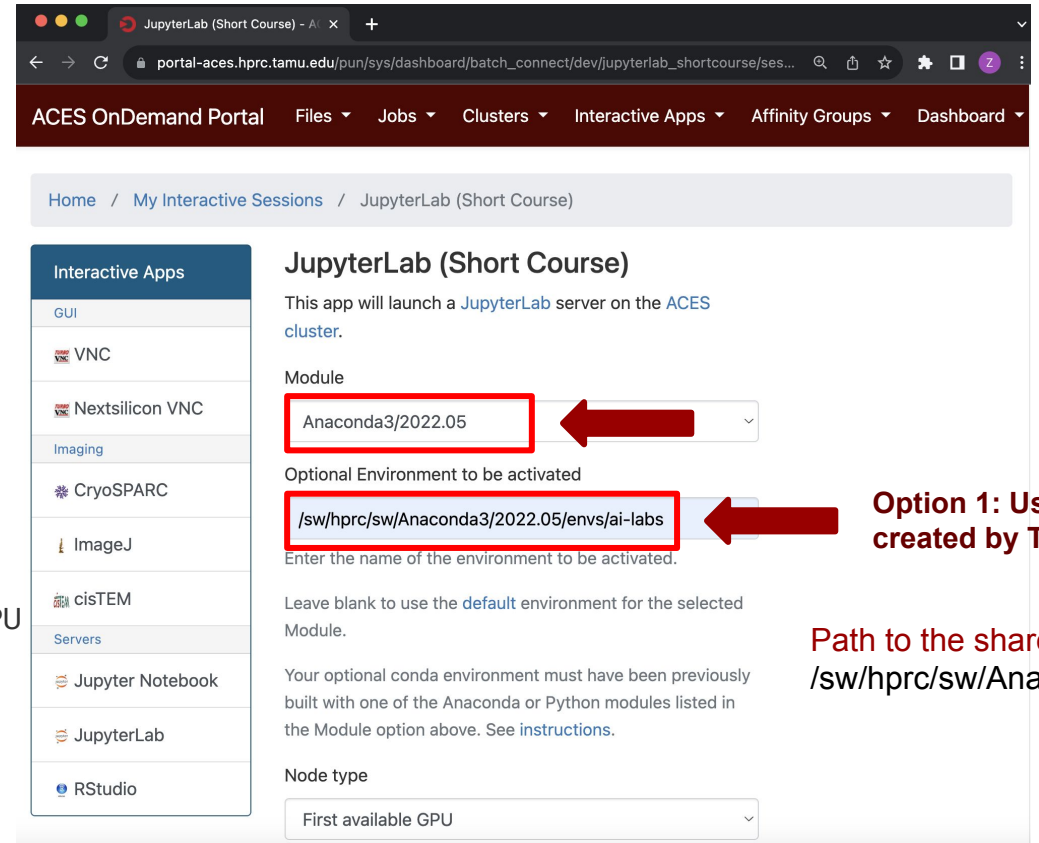
- Enter this directory (your local copy)

  ```
  $ cd ai_tech_labs
  ```

Go to JupyterLab Page



JupyterLab Page



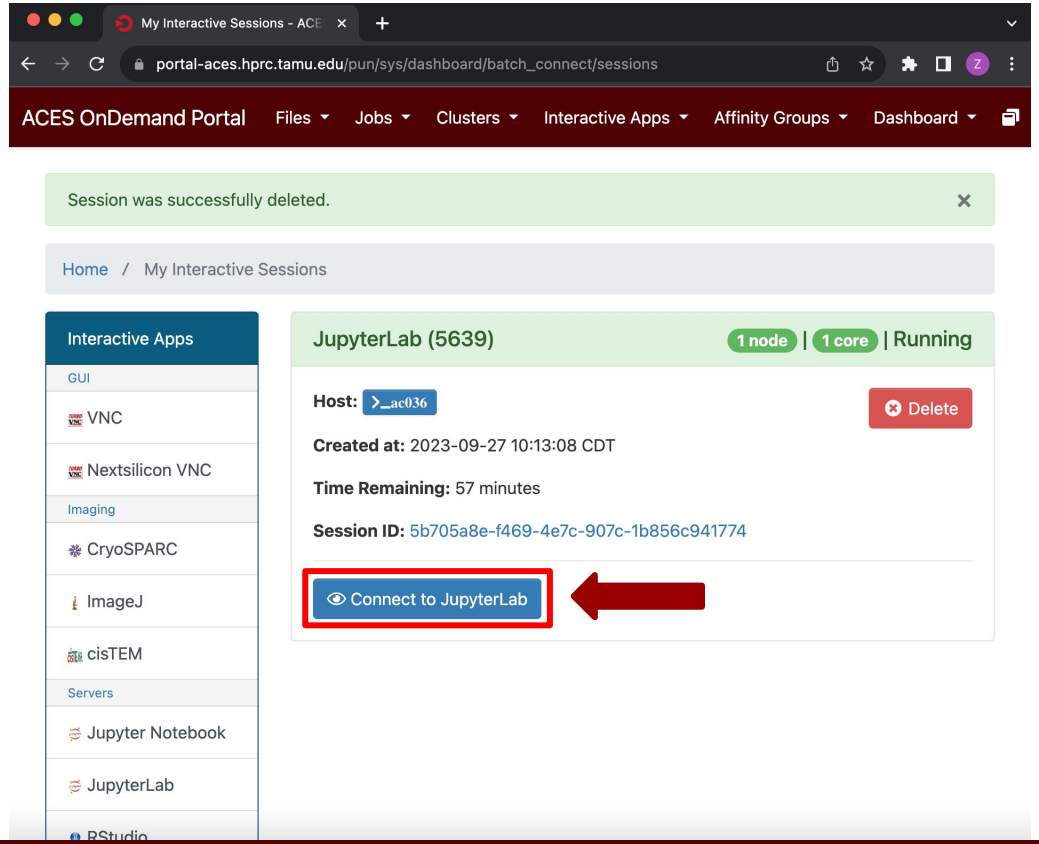
Other fields:

Node Type: First available GPU
Number of GPUs: 1
Number of hours: 3
Number of cores: 3
Total memory (GB): 5

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

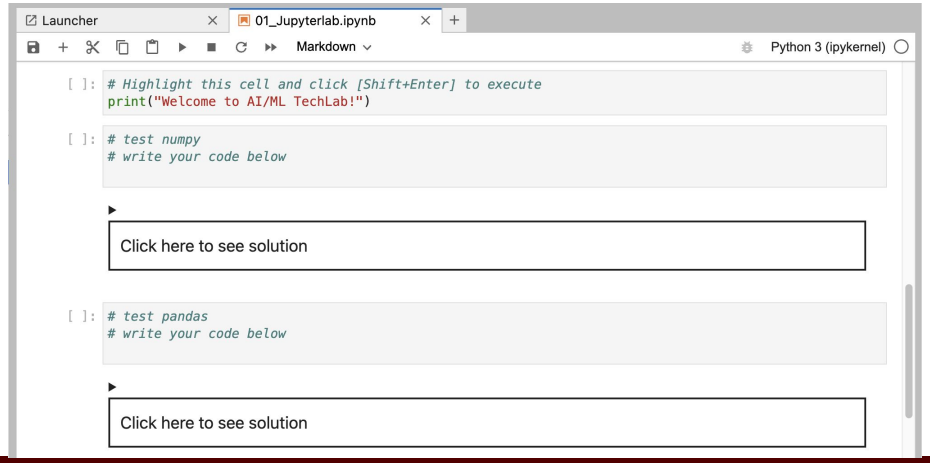
Path to the shared environment:
/sw/hprc/sw/Anaconda3/2022.05/envs/ai-labs

Connect to JupyterLab

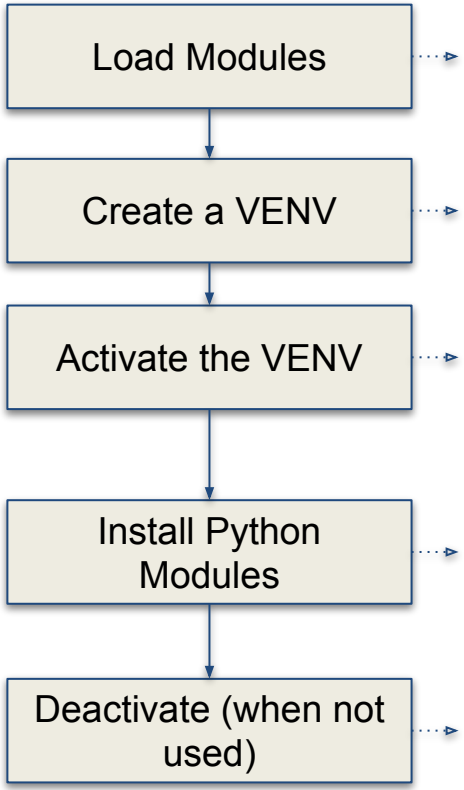


Review and Exercise

- 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.

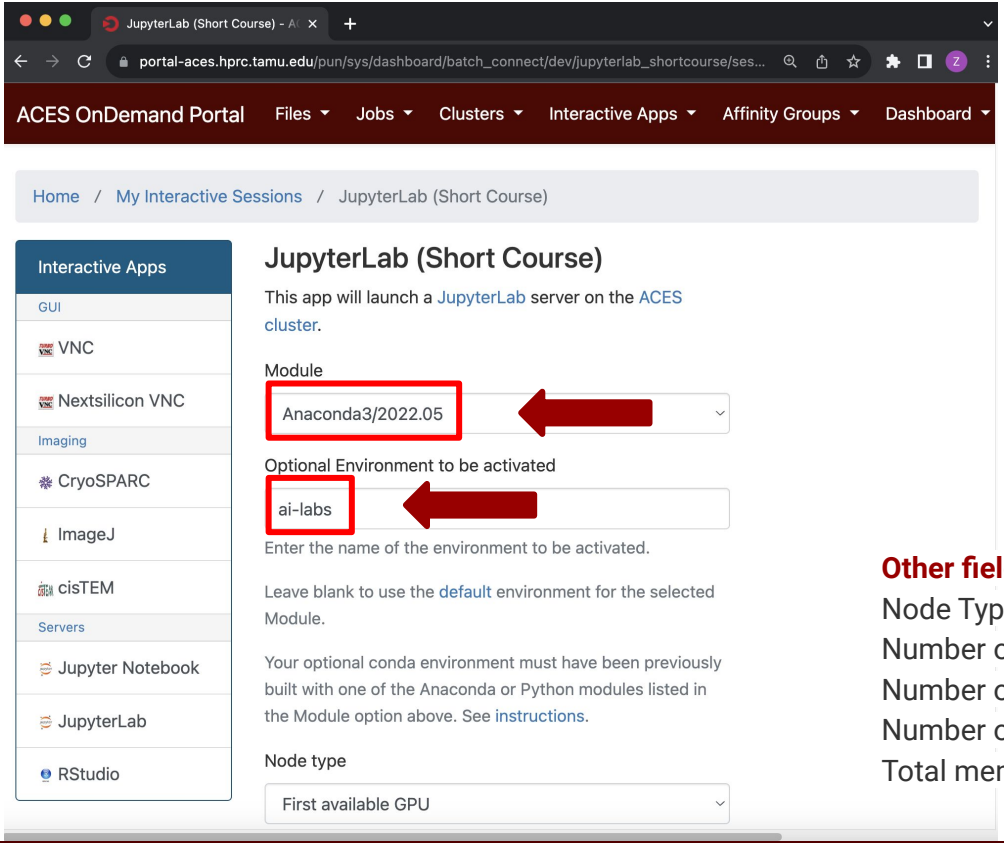


Option 2




```
# clean up and load Anaconda
cd $SCRATCH
module purge
module load Anaconda3/2022.05

# create a Python virtual environment
conda create -n ai-labs

# activate the virtual environment
source activate ai-labs

# install required package to be used in the portal
conda install -c anaconda jupyter
conda install -c anaconda pandas
conda install -c conda-forge matplotlib
conda install -c anaconda scikit-learn
conda install pytorch torchvision torchaudio
pytorch-cuda=11.8 -c pytorch -c nvidia

# deactivate the virtual environment
# source deactivate
```

JupyterLab Page



Other fields:

Node Type: First available GPU
Number of GPUs: 1
Number of hours: 3
Number of cores: 3
Total memory (GB): 5

Lab II. Data Exploration

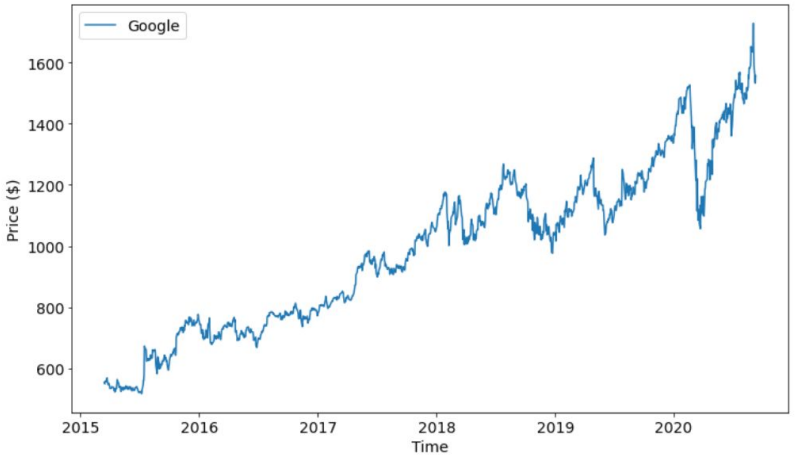
Data Structures

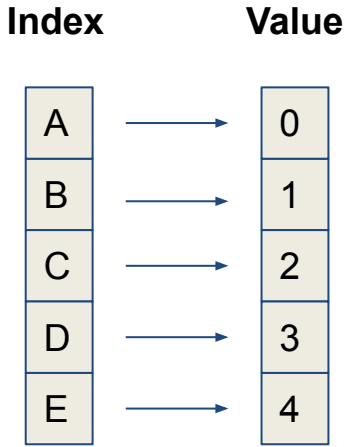
Pandas has two data structures that are descriptive and optimized for data with different dimensions.

- **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

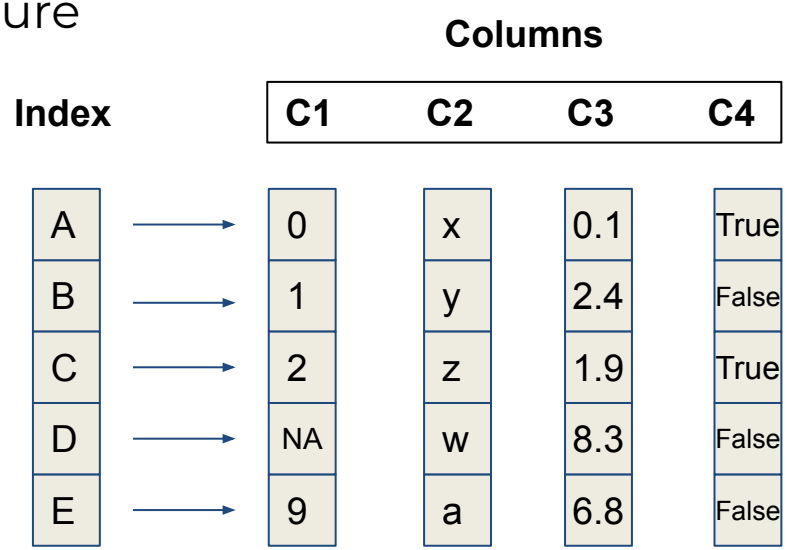




DataFrame in pandas

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

A	B	C	D	E	F	G	H
id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors
7129300520	20141013T0(	221900	3	1	1180	5650	1
6414100192	20141209T0(	538000	3	2.25	2570	7242	2
5631500400	20150225T0(	180000	2	1	770	10000	1
2487200875	20141209T0(	604000	4	3	1960	5000	1
1954400510	20150218T0(	510000	3	2	1680	8080	1
7237550310	20140512T0(	1.23E+06	4	4.5	5420	101930	1
1321400060	20140627T0(	257500	3	2.25	1715	6819	2
2008000270	20150115T0(	291850	3	1.5	1060	9711	1
2414600126	20150415T0(	229500	3	1	1780	7470	1



Pandas Learning Objectives

After this lesson, you will know how to:

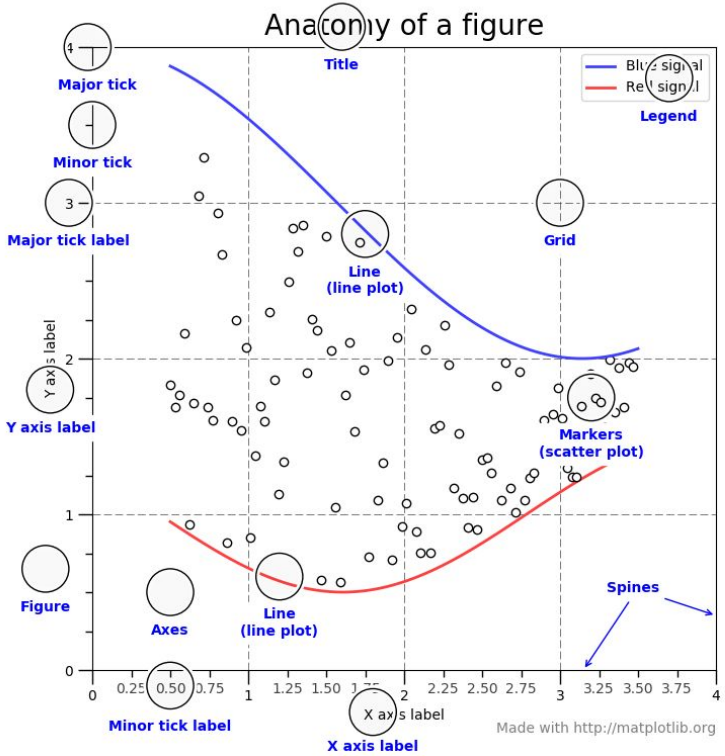
- Create a DataFrame
- Retrieve a Row or Column
- Drop Entries
- Index, Select, and Filter data
- Sort data
- Input and Output

JupyterLab Exercises

Key Plotting Concepts in Matplotlib

- **Matplotlib: Figure**
  Figure is the object that keeps the whole image output. Adjustable parameters include:
  1. Image size (set_size_inches())
  2. Whether to use tight_layout (set_tight_layout())

- **Matplotlib: Axes**
  Axes object represents the pair of axis that contain a single plot (x-axis and y-axis). The Axes object also has more adjustable parameters:
  1. The plot frame (set_frame_on() or set_frame_off())
  2. X-axis and Y-axis limits (set_xlim() and set_ylim())
  3. X-axis and Y-axis Labels (set_xlabel() and set_ylabel())
  4. The plot title (set_title())



(Credit: matplotlib.org)

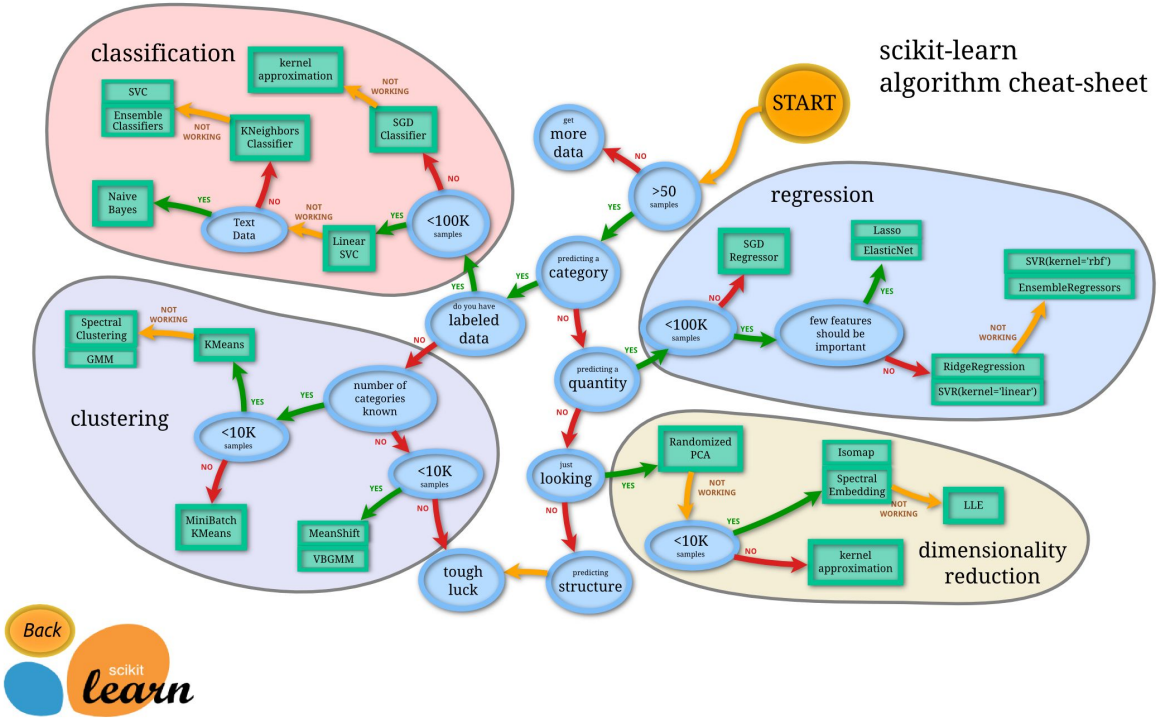
Matplotlib Learning Objectives

After this lesson, you will know how to:

- Scatter plot and Line plot
- Subplots
- Color map
- Contour figures
- 3D figures
  - Surface plots
  - Wire-frame plot
  - Contour plots with projections

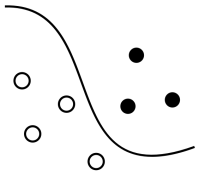
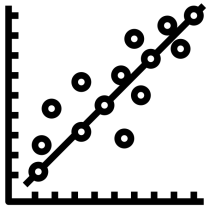
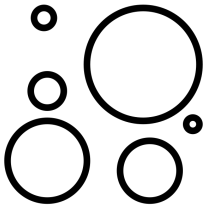
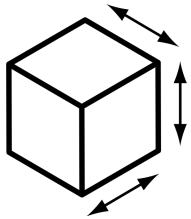
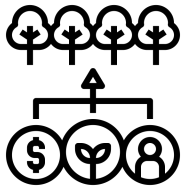
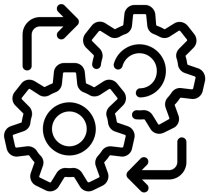
JupyterLab Exercises

Lab III. Machine Learning



Main Features of scikit-learn

Classification	Regression	Clustering	Dimension Reduction	Model Selection	Preprocessing
Identifying category of an object	**Predicting a attribute for an object**	**Grouping similar objects into sets**	**Reducing the number of dimensions**	**Selecting models with parameter search**	**Preprocessing data to prepare for modeling**
Applications: Spam detection, image recognition. **Algorithms**: SVM, nearest neighbors, random forest, and more...	**Applications**: Drug response, Stock prices. **Algorithms**: SVR, nearest neighbors, random forest, and more...	**Applications:** Customer segmentation, Grouping experiment outcomes **Algorithms:** k-Means, spectral clustering, mean-shift, and more...	**Applications:** Visualization, Increased efficiency **Algorithms:** k-Means, feature selection, non-negative matrix factorization, and more...	**Applications:** Improved accuracy via parameter tuning **Algorithms:** grid search, cross validation, metrics, and more...	**Applications:** Transforming input data such as text for use with machine learning algorithms. **Algorithms:** preprocessing, feature extraction, and more...

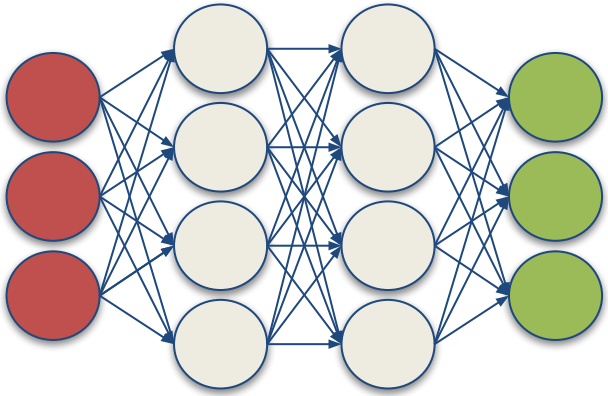
JupyterLab Exercises

Lab IV. 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/

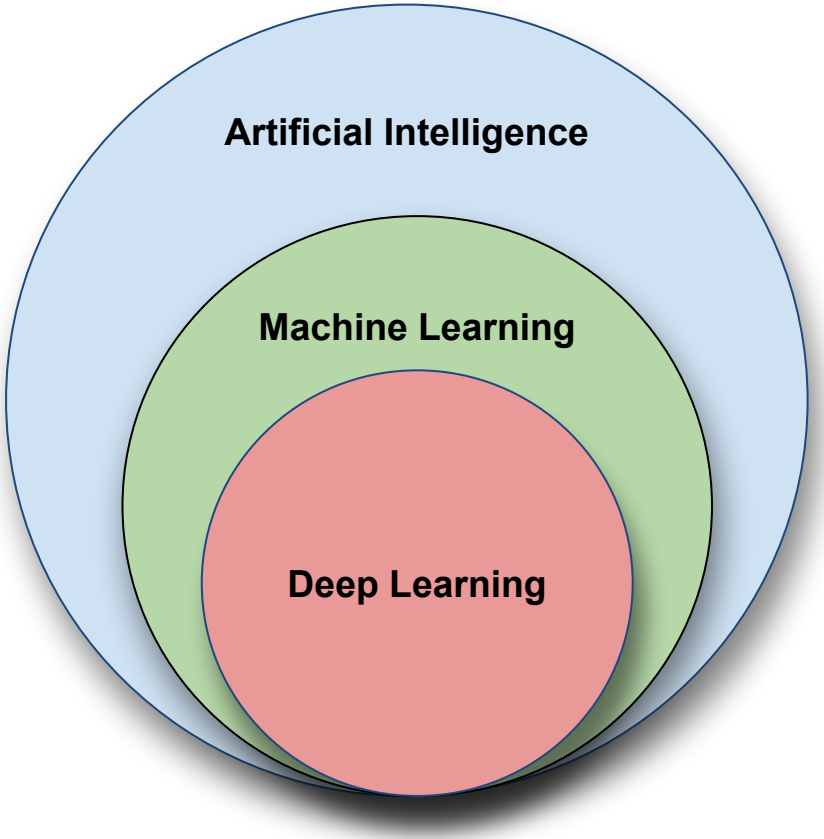
Visualization of CNN
by Adam Harley
https://adamharley.com/nn_vis/cnn/3d.html



TensorFlow PyTorch

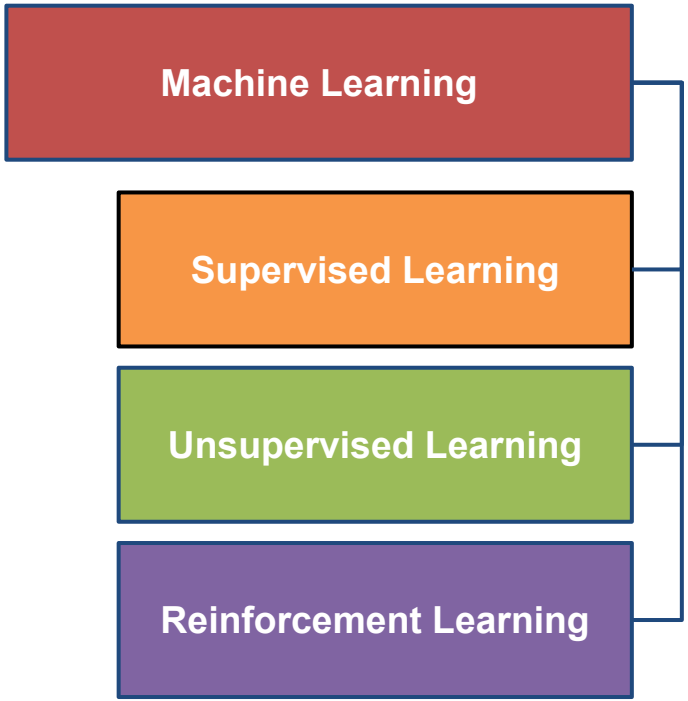
Relationship of AI, ML, and DL

- **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**.

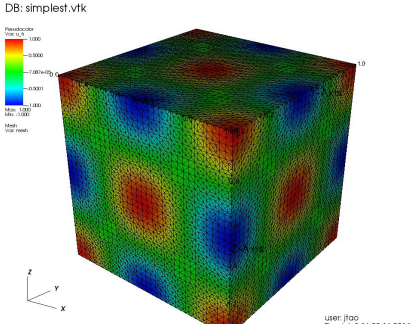


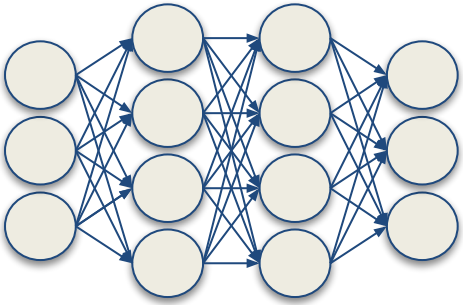
Types of ML Algorithms

- **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.

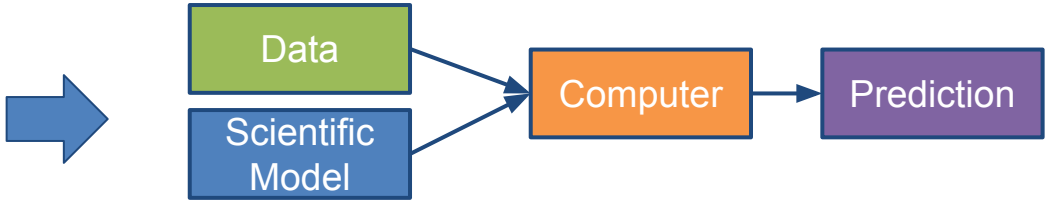


Machine Learning

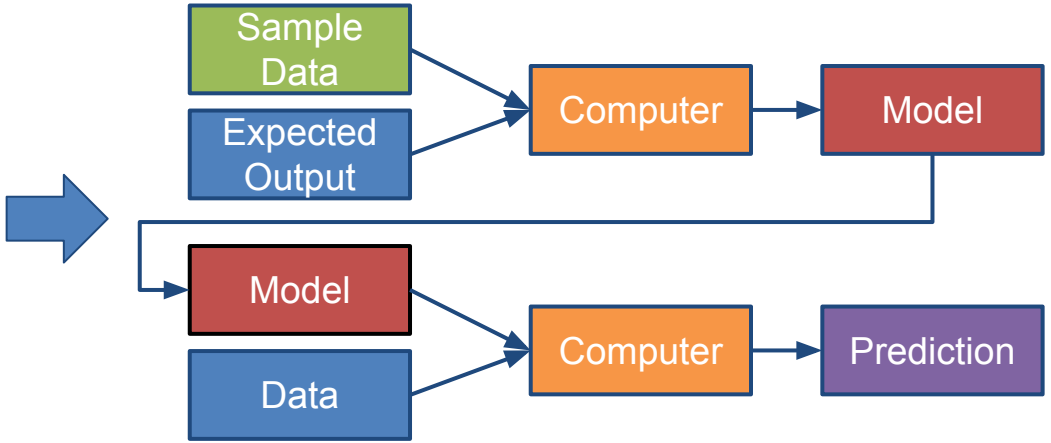




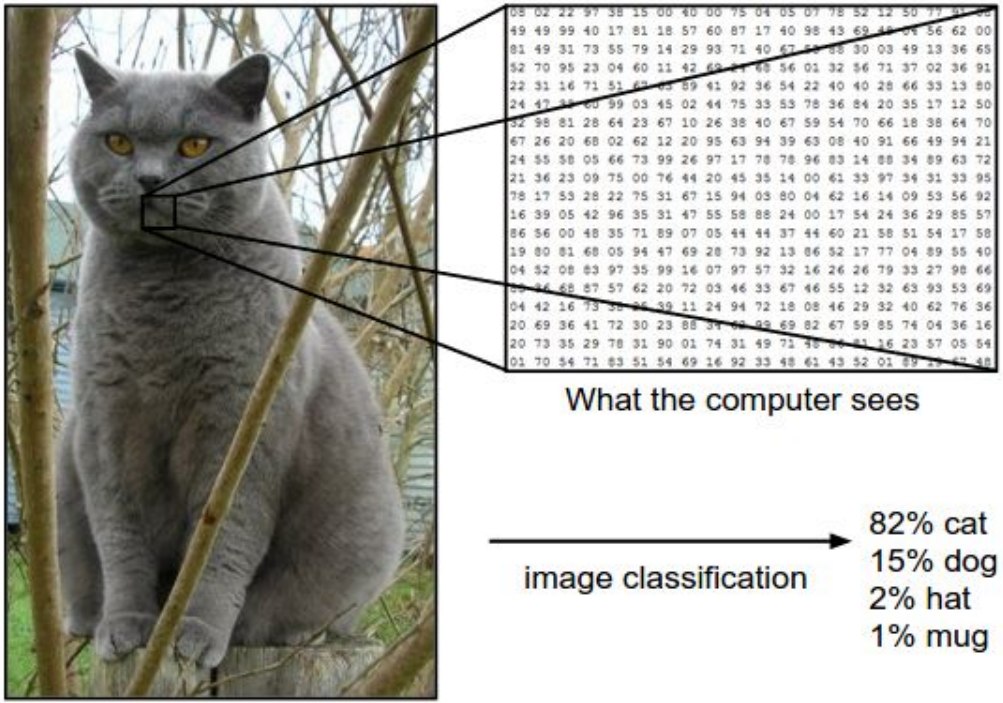
Traditional Modeling



Machine Learning (Supervised Learning)



Inputs and Outputs



What the computer sees

image classification

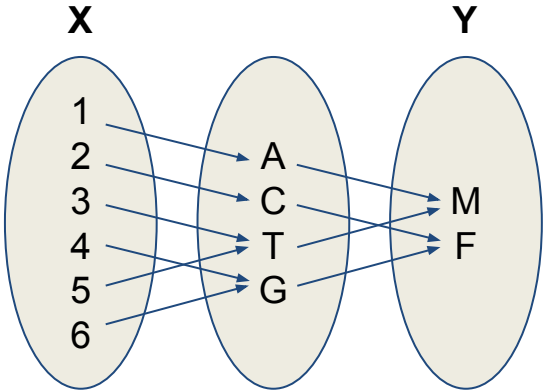
82% cat
15% dog
2% hat
1% mug

Image from the Stanford CS231 Course

256 X 256 Matrix

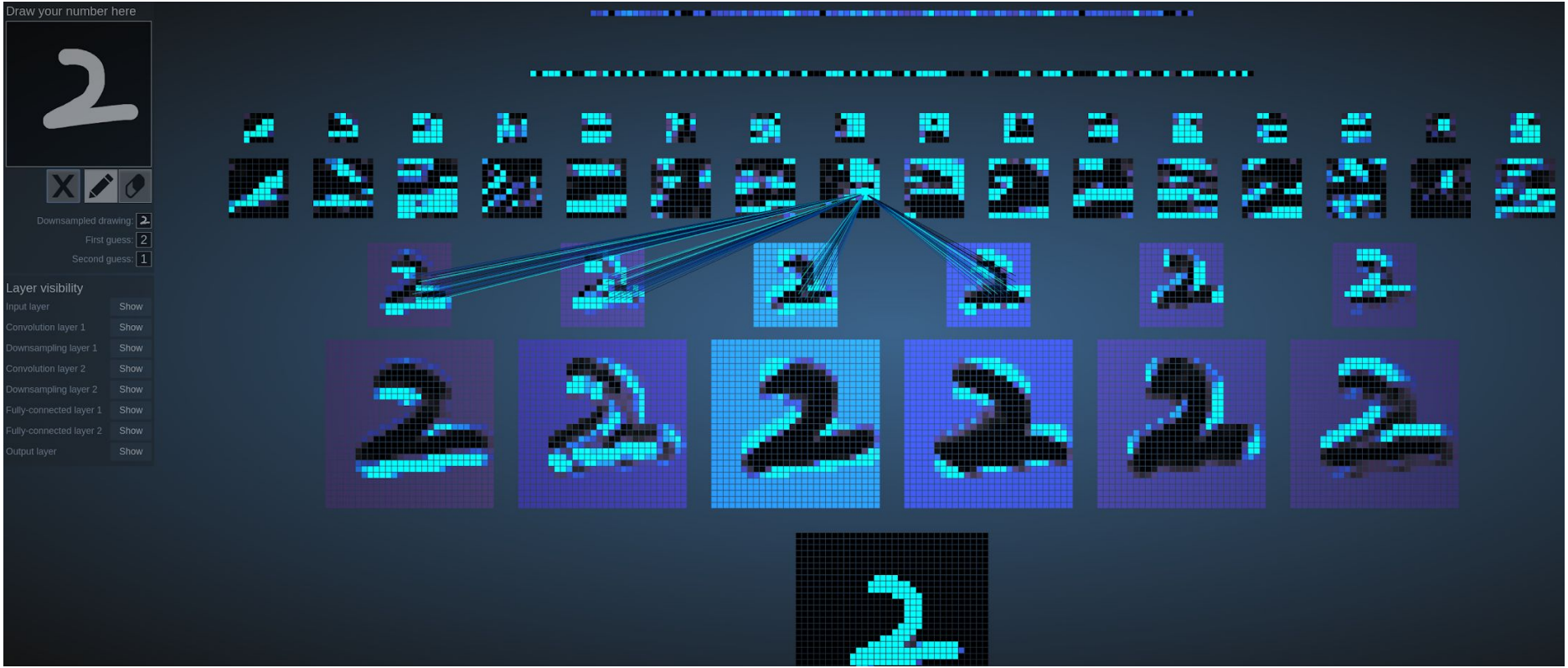
DL model

4-Element Vector



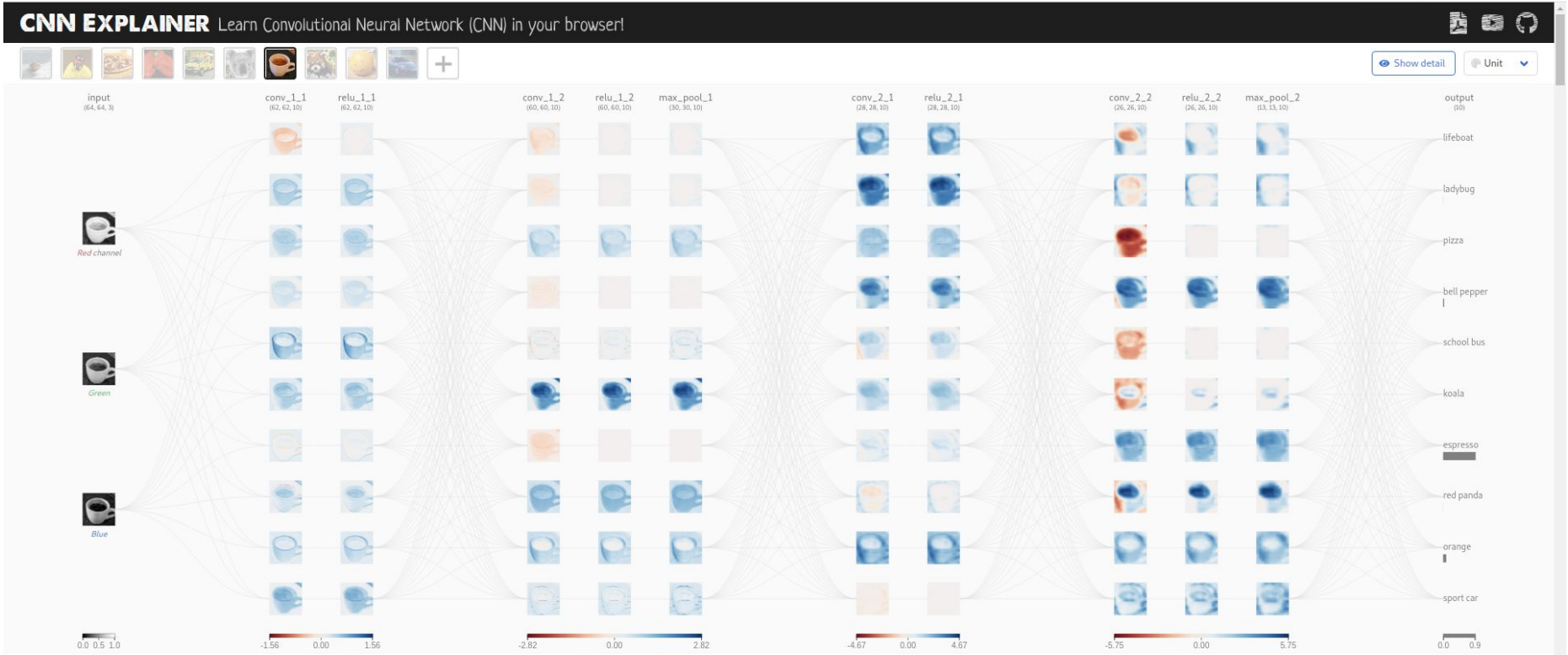
With deep learning, we are searching for a **surjective** (or **onto**) function **f** from a set **X** to a set **Y**.

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 (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.