

Technology Lab: Using AI Frameworks in Jupyter Notebook

Zhenhua He

02/21/2023



High Performance
Research Computing
DIVISION OF RESEARCH

AI Tech Labs

Lab I. JupyterLab (30 mins)

We will load required modules with Jupyter Lmode extension and run JupyterLab on HPRC 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 Keras 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.

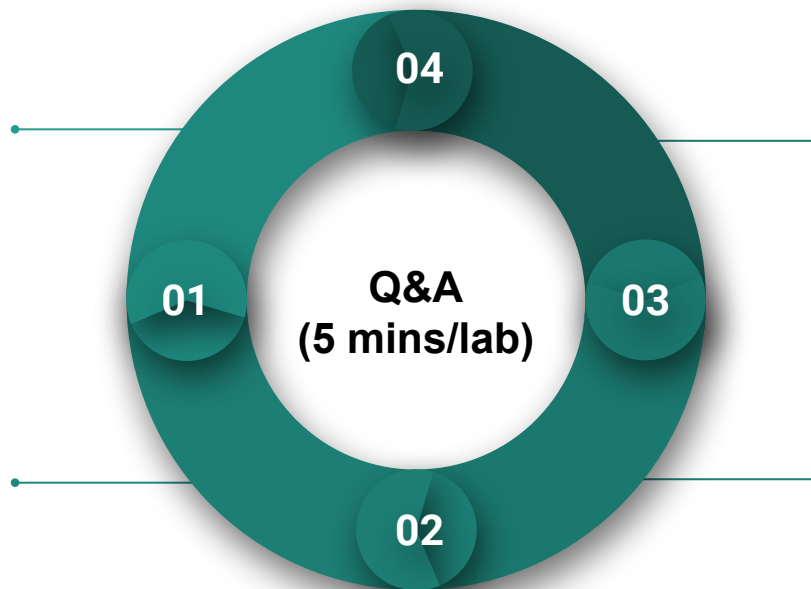


Figure 1. Structure of the AI Technology Labs.

Lab I. JupyterLab



File Edit View Run Kernel Tabs Settings Help

Files

- notebooks
- Data.ipynb (an hour ago)
- Fasta.ipynb (a day ago)
- Julia.ipynb (a day ago)
- Lorenz.ipynb (seconds ago)**
- R.ipynb (a day ago)
- iris.csv (a day ago)
- lightning.json (9 days ago)
- lorenz.py (3 minutes ago)

Running

Commands

Cell Tools

Output View

lorenz.pydef solve_lorenz(N=10, max_time=4.0, sigma=10.0, beta=8./3, rho=28.0):
 """Plot a solution to the Lorenz differential equations."""
 fig = plt.figure()
 ax = fig.add_axes([0, 0, 1, 1], projection='3d')
 ax.axis('off')

 # prepare the axes limits
 ax.set_xlim((-25, 25))
 ax.set_ylim((-35, 35))
 ax.set_zlim((5, 55))

 def lorenz_deriv(x_y_z, t0, sigma=sigma, beta=beta, rho=rho):
 """Compute the time-derivative of a Lorenz system."""
 x, y, z = x_y_z
 return [sigma * (y - x), x * (rho - z) - y, x * y - beta * z]

 # Choose random starting points, uniformly distributed from -15 to 15
 np.random.seed(1)
 x0 = -15 + 30 * np.random.random((N, 3))

Python 3

In this Notebook we explore the Lorenz system of differential equations:

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy \end{aligned}$$

Let's call the function once to view the solutions. For this set of parameters, we see the trajectories swirling around two points, called attractors.

In [4]: `from lorenz import solve_lorenz`
`t, x_t = solve_lorenz(N=10)`

sigma 10.00
beta 2.67
rho 28.00

A 3D plot of the Lorenz attractor, showing a complex, swirling trajectory in a three-dimensional space. The plot is rendered with a green-to-yellow color gradient and is set against a dark background. The axes are labeled x, y, and z, with the x-axis pointing to the right, the y-axis pointing upwards, and the z-axis pointing out of the page.

L1 - Resources

- [Texas A&M High Performance Research Computing \(HPRC\)](#)
- [FASTER Quick Start Guide](#)
- [ACES Phase I Guide](#)
- [ACCESS Documentation](#)
- [FASTER Portal \(TAMU\)](#)
- [FASTER Portal \(ACCESS\)](#)
- [HPRC YouTube Channel](#)
- help@hprc.tamu.edu

Getting Started with FASTER and ACES

FASTER Cluster

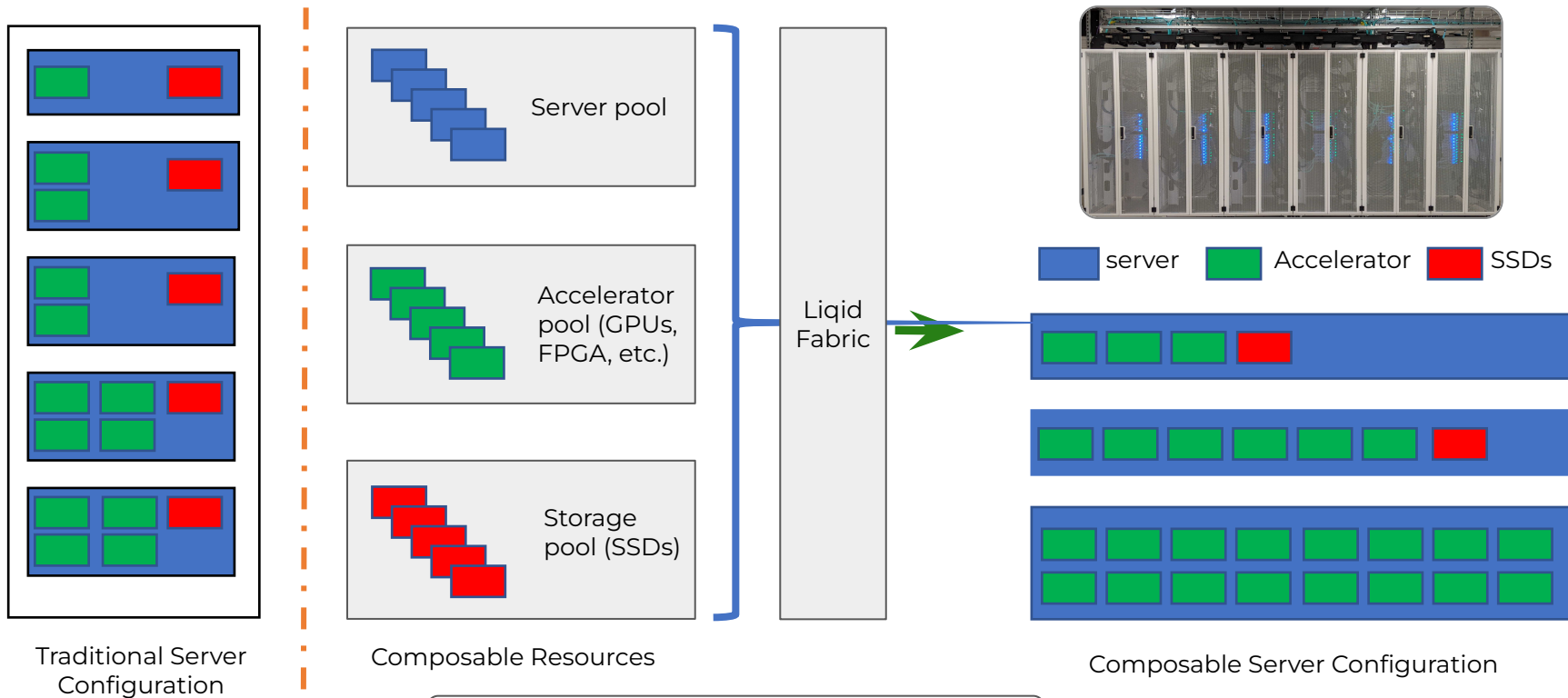
hprc.tamu.edu/wiki/FASTER:Intro

Resources	Quantity
64-core login nodes	4 (3 for TAMU, 1 for ACCESS)
64-core compute nodes (256GB RAM each)	180 (11,520 cores)
Composable GPUs	200 T4 16GB 40 A100 40GB 10 A10 24GB 4 A30 24GB 8 A40 48GB
Interconnect	Mellanox HDR100 InfiniBand (MPI and storage) Liquid PCIe Gen4 (GPU composability)
Global Disk	5PB DDN Lustre appliances



FASTER (Fostering Accelerated Sciences Transformation Education and Research) is a 180-node Intel cluster from Dell featuring the Intel Ice Lake processor.

Composability at the Hardware Level



hprc.tamu.edu/wiki/FASTER:Intro

ACES - Accelerating Computing for Emerging Sciences (Phase I)

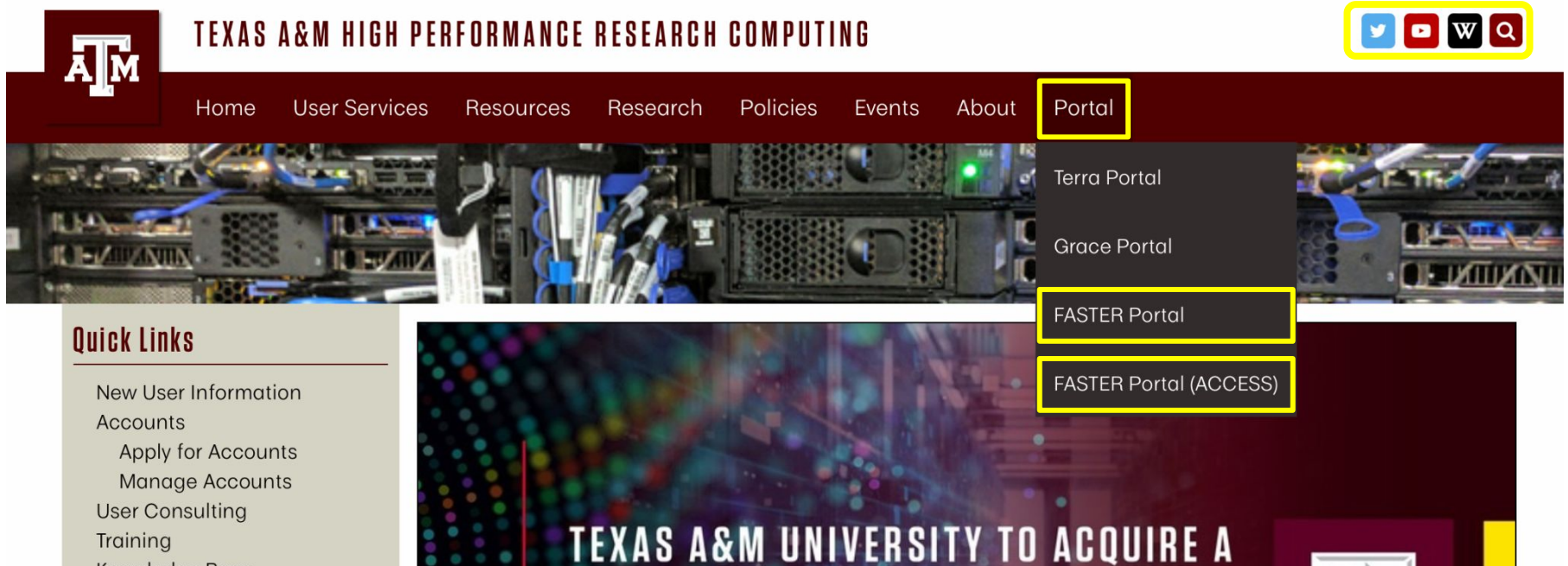


Component	Quantity	Description
Graphcore IPU	16	16 Colossus GC200 IPUs and dual AMD Rome CPU server on a 100 GbE RoCE fabric
Intel FPGA PAC D5005	2	FPGA SOC with Intel Stratix 10 SX FPGAs, 64 bit quad-core Arm Cortex-A53 processors, and 32GB DDR4
Intel Optane SSDs	8	3 TB of Intel Optane SSDs addressable as memory using MemVerge Memory Machine.

ACES Phase I components are available through [FASTER](#)

Accessing the HPRC Portal

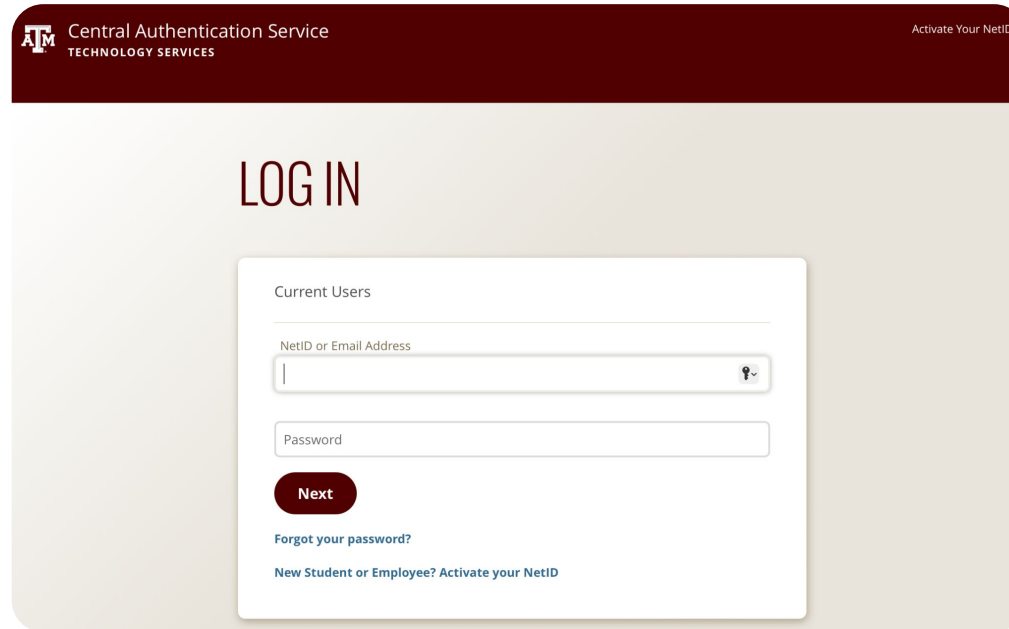
- HPRC webpage: hprc.tamu.edu, Portal dropdown menu



The screenshot displays the HPRC website interface. At the top left is the Texas A&M logo. The main header reads "TEXAS A&M HIGH PERFORMANCE RESEARCH COMPUTING". To the right of the header are social media icons for Twitter, YouTube, and LinkedIn, along with a search icon. Below the header is a navigation bar with the following links: Home, User Services, Resources, Research, Policies, Events, About, and Portal. The "Portal" link is highlighted with a yellow box. A dropdown menu is open from the "Portal" link, listing four options: Terra Portal, Grace Portal, FASTER Portal, and FASTER Portal (ACCESS). The "FASTER Portal" and "FASTER Portal (ACCESS)" options are highlighted with yellow boxes. Below the navigation bar is a banner image of server racks. On the left side, there is a "Quick Links" section with a list of links: New User Information, Accounts, Apply for Accounts, Manage Accounts, User Consulting, Training, and Knowledge Base. At the bottom, a banner features the text "TEXAS A&M UNIVERSITY TO ACQUIRE A" over a background of colorful bokeh lights.

Accessing FASTER via the HPRC Portal (TAMU)

Log-in using your TAMU NetID credentials.



The screenshot shows the login interface for the Central Authentication Service. At the top, there is a dark red header with the TAMU logo, the text "Central Authentication Service TECHNOLOGY SERVICES", and a link "Activate Your NetID". The main content area is light beige and features the heading "LOG IN" in large, dark letters. Below the heading is a white login form with the following elements:

- A label "Current Users" above a horizontal line.
- A label "NetID or Email Address" above a text input field with a small user icon on the right.
- A label "Password" above a text input field.
- A dark red "Next" button.
- A blue link "Forgot your password?".
- A blue link "New Student or Employee? Activate your NetID".

Accessing FASTER via the HPRC Portal (ACCESS)

Log-in using your ACCESS credentials.

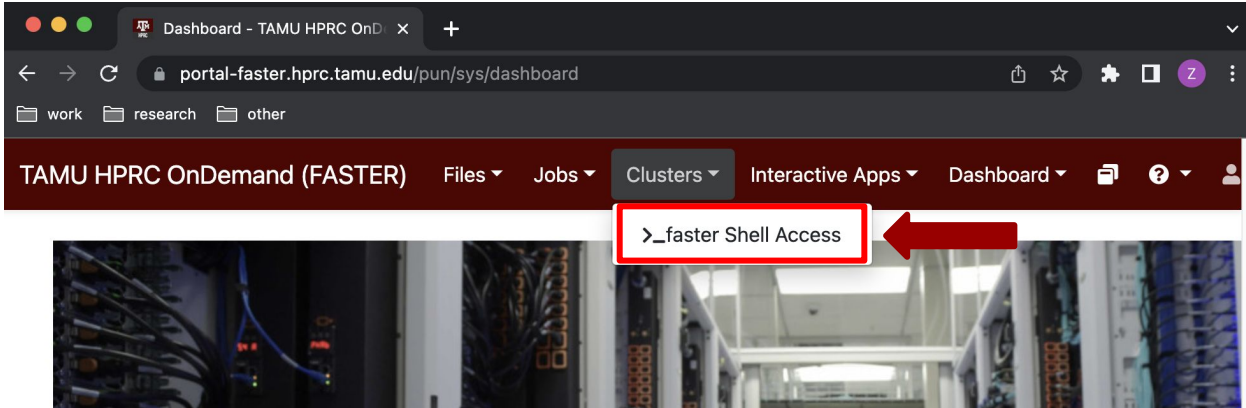
The screenshot shows the ACCESS portal interface. At the top left is the ACCESS logo, and at the top right is the 'Powered By CILogon' logo. Below the logo is a 'Consent to Attribute Release' section with a dropdown arrow. The consent text reads: 'TAMU FASTER ACCESS_OOD requests access to the following information. If you do not approve this request, do not proceed.' Below this are three bullet points: 'Your CILogon user identifier', 'Your name', 'Your email address', and 'Your username and affiliation from your identity provider'. Below the consent section is a 'Select an Identity Provider' section with a dropdown menu showing 'ACCESS CI (XSEDE)'. Below the dropdown is a 'Remember this selection' checkbox and a 'Log On' button. At the bottom of this section is a note: 'By selecting "Log On", you agree to the [privacy policy](#).' At the very bottom of the page is a footer with small text: 'For questions about this site, please see [FAQs](#) or send email to help@cilogon.org. Know your responsibilities using the CILogon Service. See [acknow/isp/privac](#) for support for this site.'

The screenshot shows the ACCESS portal login page. At the top left is the ACCESS logo, and at the top right is the CILogon logo. Below the logo is the text 'Login to CILogon'. Below this are two input fields: 'ACCESS Username' and 'ACCESS Password'. Below the password field is a checkbox for 'Don't Remember Login' and a 'Login' button. To the right of the login form is the CILogon logo and the text 'CILogon facilitates secure access to CyberInfrastructure (CI)'. Below this are four links: 'If you had an XSEDE account, please enter your XSEDE username and password for ACCESS login', 'Register for an ACCESS Account', 'Forgot your password?', and 'Need Help?'. At the bottom of the page is a link: 'Click Here for Assistance'.

This is a close-up of the 'Select an Identity Provider' dropdown menu. The dropdown is open, showing the selected option 'ACCESS CI (XSEDE)' with a question mark icon to its right.

Select the Identity Provider appropriate for your account.

Shell Access - I



The screenshot shows a web browser window with the URL `portal-faster.hprc.tamu.edu/pun/sys/dashboard`. The navigation bar includes links for "TAMU HPRC OnDemand (FASTER)", "Files", "Jobs", "Clusters", "Interactive Apps", and "Dashboard". A red box highlights the link ">_faster Shell Access" in the Clusters dropdown menu, with a red arrow pointing to it. Below the navigation bar is a banner image of server racks.

OnDemand provides an integrated, single access point for all of your HPC resources.

Message of the Day

IMPORTANT POLICY INFORMATION

- **Unauthorized use of HPRC resources is prohibited and subject to criminal prosecution.**
- **Use of HPRC resources in violation of United States export control laws and regulations is prohibited. Current HPRC staff members are US citizens and legal residents.**
- **Sharing HPRC account and password information is in violation of State Law. Any shared accounts will be DISABLED.**
- **Authorized users must also adhere to ALL policies at: <https://hprc.tamu.edu/policies>**

!! WARNING: THERE ARE ONLY NIGHTLY BACKUPS OF USER HOME DIRECTORIES. !!

Shell Access - II

```
Dashboard - TAMU HPRC OnD x happidence1@login2:~
portal-faster.hprc.tamu.edu/pun/sys/shell/ssh/faster.hprc.tamu.edu
work research other
Host: faster.hprc.tamu.edu Themes: Default
| Website: https://hprc.tamu.edu
| Consulting: help@hprc.tamu.edu (preferred) or (979) 845-0219
| FASTER Documentation: https://hprc.tamu.edu/wiki/FASTER
| Grace Documentation: https://hprc.tamu.edu/wiki/Grace
| YouTube Channel: https://www.youtube.com/texasamhprc
|=====
*****
== IMPORTANT POLICY INFORMATION ==
* - Unauthorized use of HPRC resources is prohibited and subject to
* criminal prosecution.
* - Use of HPRC resources in violation of United States export control
* laws and regulations is prohibited. Current HPRC staff members are
* US citizens and legal residents.
* - Sharing HPRC account and password information is in violation of
* Texas State Law. Any shared accounts will be DISABLED.
* - Authorized users must also adhere to ALL policies at:
* https://hprc.tamu.edu/policies/
*****

!! WARNING: THERE ARE ONLY NIGHTLY BACKUPS OF USER HOME DIRECTORIES. !!

Please restrict usage to 8_CORES across ALL login nodes.
Users found in violation of this policy will be SUSPENDED.

To see these messages again, run the motd command.
Your current disk quotas are:
Disk          Disk Usage  Limit  File Usage  Limit
/home/happidence1      56K      10.0G      26      10000
/scratch/user/happidence1  631.0G   2.0T     450644   1000000
* Quota increase for /scratch/user/happidence1 will expire on May 21, 2023
/scratch/group/benchmark_prj  325.1G   5.0T     1333878  5000000
/scratch/group/hprc          3.9T     10.0T     615489   1000000
* Quota increase for /scratch/group/hprc will expire on Dec 31, 2026
Type 'showquota' to view these quotas again.
(base) [happidence1@faster2 ~]$
```

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

The screenshot shows a web browser window displaying the TAMU HPRC OnDemand (FASTER) dashboard. The browser's address bar shows the URL `portal-faster.hprc.tamu.edu/pun/sys/dashboard`. The dashboard header includes navigation links for Files, Jobs, Clusters, Interactive Apps, and Dashboard. The 'Interactive Apps' menu is open, listing various applications such as BIO, Beauti, IGV, Mauve, Structure, GUI, ANSYS Workbench, Abaqus/CAE, MATLAB, VNC, Imaging, ChimeraX, Diffusion Toolkit & TrackVis, FSL, and Fiji. At the bottom of this menu, 'Jupyter Notebook' and 'JupyterLab' are listed. A red box highlights 'JupyterLab', and a red arrow points to it from the right. The dashboard content includes a 'Message of the Day' section, 'IMPORTANT POLICY INFORMATION' with a list of rules, and a red warning: '!! WARNING: THERE ARE ONLY NIGHTLY BACKUPS OF US'. The footer shows 'powered by OPEN OnDemand' and 'OnDemand version: 2.0.28'.

Dashboard - TAMU HPRC OnD x happidence1@login2:~ x +

portal-faster.hprc.tamu.edu/pun/sys/dashboard

work research other

TAMU HPRC OnDemand (FASTER) Files Jobs Clusters Interactive Apps Dashboard

BIO

Beauti

IGV

Mauve

Structure

GUI

ANSYS Workbench

Abaqus/CAE

MATLAB

VNC

Imaging

ChimeraX

Diffusion Toolkit & TrackVis

FSL

Fiji

Servers

Jupyter Notebook

JupyterLab

OnDemand provides an integrated, single access

Message of the Day

IMPORTANT POLICY INFORMATION

- Unauthorized use of HPRC resources is prohibited
- Use of HPRC resources in violation of United States
- HPRC staff members are US citizens and legal res
- Sharing HPRC account and password information
- DISABLED.
- Authorized users must also adhere to ALL policies

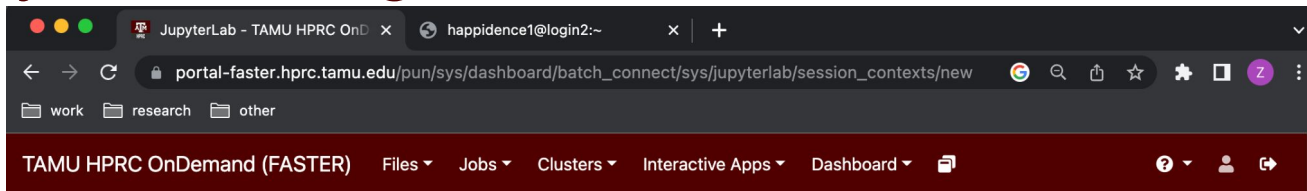
!! WARNING: THERE ARE ONLY NIGHTLY BACKUPS OF US

powered by OPEN OnDemand

OnDemand version: 2.0.28

<https://portal-faster.hprc.tamu.edu/pun/sys/dashboard#>

JupyterLab Page



Home / My Interactive Sessions / JupyterLab

Interactive Apps

- BIO
- Beauti
- IGV
- Mauve
- Structure
- GUI
- ANSYS Workbench
- Abaqus/CAE
- MATLAB
- VNC
- imaging
- ChimeraX

JupyterLab

This app will launch a [JupyterLab](#) server on the **FASTER** cluster.

Module

Python/3.8.2

Anaconda3 uses Python3

Optional Environment to be activated

Enter the name of the environment to be activated. (Optional)

The default virtualenvs for Anaconda3/2021.11 and Python/3.8.2 have jupyterlmod which enables loading lmod modules.

Leave blank to use the [default](#) environment for the selected Module.

Your optional conda environment must have been previously built with one of the Anaconda or Python modules listed in the Module option above. See [instructions](#).

Number of hours: 3
Number of cores: 1
Total memory (GB): 3
Node type: ANY

Connect to JupyterLab

The screenshot shows a web browser window with the URL `portal-faster.hprc.tamu.edu/pun/sys/dashboard/batch_connect/sessions`. The page header includes "TAMU HPRC OnDemand (FASTER)" and navigation menus for "Files", "Jobs", "Clusters", "Interactive Apps", and "Dashboard". A green notification bar at the top states "Session was successfully deleted." Below this is a breadcrumb trail: "Home / My Interactive Sessions".

The main content area is divided into two sections. On the left is a sidebar titled "Interactive Apps" with a list of application icons: BIO, Beauti, IGV, Mauve, Structure, GUI, ANSYS Workbench, Abaqus/CAE, MATLAB, VNC, and Imaging. On the right is a card for a "JupyterLab (76023)" session. The card has a green header with "1 node | 1 core | Running". Below the header, it displays: "Host: >_fc008" with a red "Delete" button; "Created at: 2022-11-28 14:28:18 CST"; "Time Remaining: 2 hours and 58 minutes"; and "Session ID: efdf374d-2e87-49e8-899d-f84a2cd42cd3". At the bottom of the card, a blue button labeled "Connect to JupyterLab" is highlighted with a red box, and a red arrow points to it from the right.

JupyterLab Lmod Extension

The screenshot displays a web browser window with the URL `portal-faster.hprc.tamu.edu/node/fc008/61153/lab/workspaces/auto-9/tree/AI_Labs/01_jupyterlab.ipynb`. The JupyterLab interface is visible, featuring a sidebar on the left with a file browser and a main work area on the right. A red arrow points to a blue hexagonal icon with a white gear, representing the Lmod extension, located in the sidebar. The main work area shows a JupyterLab notebook titled "01_jupyterlab.ipynb" with the following content:

JupyterLab

Originally created by Dr. [Jian Tao](#), Texas A&M University

Nov 29, 2022

The [JupyterLab Interface](#) is an interactive development environment that provides access to iPython notebooks, as well as the folder structure of our environment and a terminal window into the Ubuntu operating system. The first view you'll see includes a **menu bar** at the top, a **file browser** in the **left sidebar**, and a **main work area** that is initially open to the "Launcher" page.

```
In [4]: from lorenz import solve_lorenz
        t, x_t = solve_lorenz(N=18)
```

The notebook also includes a code cell with the following Python code:

```
def solve_lorenz(N=18, max_time=4.0, sigma=10.0, beta=8./3, rho=28.0):
    """Plot a solution to the Lorenz differential equations."""
    fig = plt.figure()
    ax = fig.add_subplot(0, 0, 1, 1, projection='3d')
    ax.axis('off')

    # prepare the axes limits
    ax.set_xlim([-25, 25])
    ax.set_ylim([-30, 30])
    ax.set_zlim([0, 55])

    def lorenz_deriv(x,y,z, sigma=sigma, beta=beta, rho=rho):
        """Compute the time-derivative of a Lorenz system."""
        x_dot = x*(rho - z)
        return (sigma*(y - x), x*(rho - z) - y, x*y - beta*z)

    # Choose random starting points, uniformly distributed from -15 to 15
```

The notebook also features an interactive output view with sliders for `sigma` (10.00), `beta` (2.67), and `rho` (28.00). Below the sliders is a 3D plot showing the Lorenz attractor.

JupyterLab Lmod Extension

portal-faster.hpc.tamu.edu/node/fc008/61153/lab/workspaces/auto-9/tree/AllLabs/01_jupyterlab.ipynb

work research other

File Edit View Run Kernel Tabs Settings Help

gcc

LOADED MODULES

- GCCcore/9.3.0
- GMP/6.2.0
- Python/3.8.2
- SQLite/3.31.1
- Tcl/8.6.10
- WebProxy/0000
- XZ/5.2.5
- binutils/2.34
- bzip2/1.0.8
- libffi/3.3
- libreadline/8.0
- ncurses/6.2
- zlib/1.2.11

AVAILABLE MODULES

- GCC/9.3.0 **Load**
- GCC/10.2.0
- GCC/10.3.0
- GCC/7.3.0-2.30
- GCC/11.2.0
- GCC/11.3.0
- GCC/12.1.0
- GCCcore/8.3.0
- GCCcore/10.2.0
- GCCcore/10.3.0

Launcher

JupyterLab

Originally created by Dr. [Jian Tao](#), Texas A&M University

Nov 29, 2022

The [JupyterLab Interface](#) is an interactive development environment that provides access to iPython notebooks, as well as the folder structure of our environment and a terminal window into the Ubuntu operating system. The first view you'll see includes a **menu bar** at the top, a **file browser** in the **left sidebar**, and a **main work area** that is initially open to the "Launcher" page.

Files

- notebooks
- Running
- Terminal 1
- Console 1
- Data.ipynb
- README.md

```
In [4]: from lorenz import solve_lorenz
        t, x_t = solve_lorenz(N=10)
```

Output View

sigma: 10.00
beta: 2.67
rho: 28.00

```
9 def solve_lorenz(N=10, max_time=4.0, sigma=10.0, beta=0.7, rho=28.0):
10     """Plot a solution to the Lorenz differential equations."""
11     fig = plt.figure()
12     ax = fig.add_axes([0, 0, 1, 1], projection='3d')
13     ax.axis('off')
14
15     # prepare the axis labels
16     ax.set_xlabel(-25, 25)
17     ax.set_ylabel(-30, 30)
18     ax.set_zlabel(5, 55)
19
20     # Compute the time-derivative of a Lorenz system.
21     def lorenz_deriv(x,y,z, t0, sigma=sigma, beta=beta, rho=rho):
22         """Compute the time-derivative of a Lorenz system."""
23         x_dot, y_dot, z_dot = x,y,z
24         return (sigma * (y - x), x * (rho - z) - y, x * y - beta * z)
25     # Choose random starting points, uniformly distributed from -15 to 15
```

Exercise: Load Required Modules

- GCC/9.3.0
- OpenMPI/4.0.3
- scikit-learn/0.23.1-Python-3.8.2
- TensorFlow/2.3.1-Python-3.8.2

Note: numpy and matplotlib have already been in the Scipy-bundle/2020.03-Python-3.8.2 module.

Test loaded modules

The screenshot shows a JupyterLab interface with a file browser on the left and a code editor on the right. The file browser shows a list of files and folders, with '01_jupyterlab.ipynb' selected. The code editor contains three code cells with instructions and code for testing numpy, pandas, and matplotlib.

code cell, press **Shift+Enter** or the "Run" button in the menu bar above, while a cell is highlighted. Sometimes, a content cell will get switched to editing mode. Pressing **Shift+Enter** will switch it back to a readable form.

Try executing the simple print statement in the cell below.

```
[ ]: # Highlight this cell and click [Shift+Enter] to execute
print("Welcome to AI Tech Labs!")
```

Click here to see solution

```
[ ]: # test numpy
# write your code below
```

Click here to see solution

```
[ ]: # test pandas
# write your code below
```

Click here to see solution

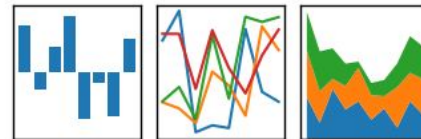
```
[ ]: # test matplotlib
# write your code below
```

Lab II. Data Exploration

matplotlib

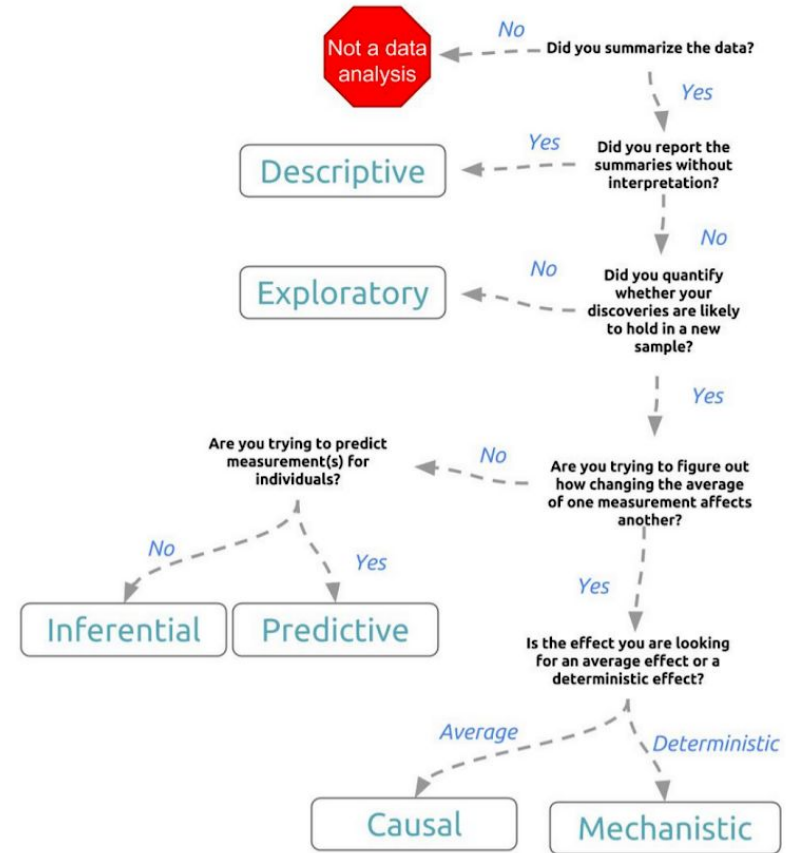
pandas

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



Types of Data Science Problems

- **Descriptive** (summaries, e.g., census)
- **Exploratory** (search for unknowns, e.g., four-planet solar system)
- **Inferential** (find correlations, e.g., many social studies)
- **Predictive** (make predictions, e.g., Face ID, Echo, Siri)
- **Causal** (explore causation, e.g., smoking versus lung cancer)
- **Mechanistic** (determine governing principles, e.g., experimental science)



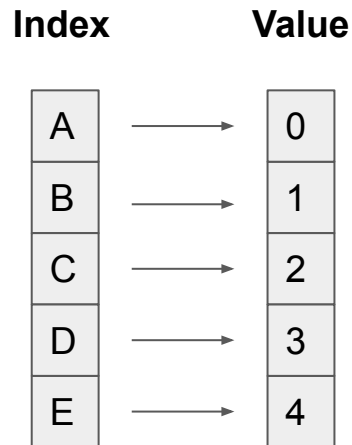
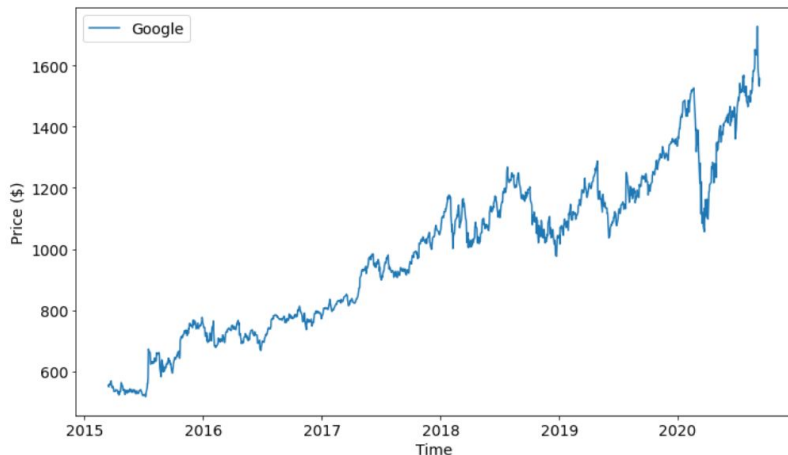
Data Structures

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

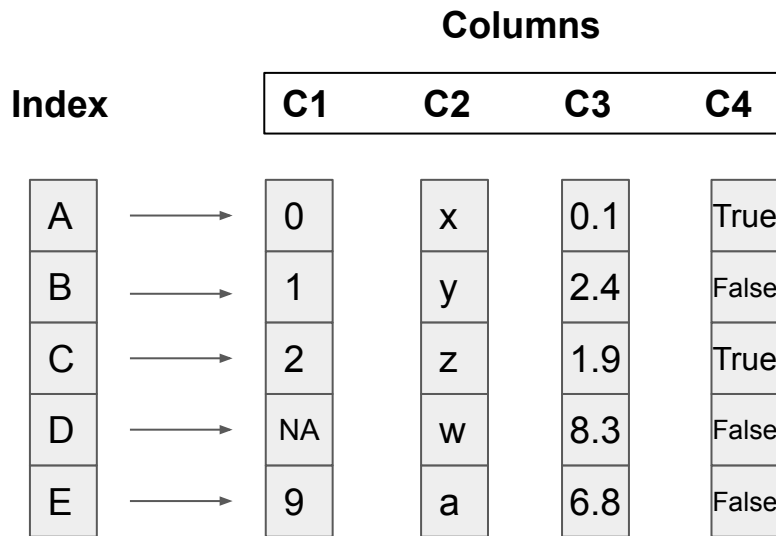
- One-dimensional labeled array
- Capable of holding any data type (integers, strings, floating point numbers, etc.)
- Example: time-series stock price data



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	20141013T00	221900	3	1	1180	5650	1
6414100192	20141209T00	538000	3	2.25	2570	7242	2
5631500400	20150225T00	180000	2	1	770	10000	1
2487200875	20141209T00	604000	4	3	1960	5000	1
1954400510	20150218T00	510000	3	2	1680	8080	1
7237550310	20140512T00	1.23E+06	4	4.5	5420	101930	1
1321400060	20140627T00	257500	3	2.25	1715	6819	2
2008000270	20150115T00	291850	3	1.5	1060	9711	1
2414600126	20150415T00	229500	3	1	1780	7470	1



Pandas Learning Objectives

After this lesson, you will know how to:

- Create a DataFrame
- Drop Entries
- Index, Select, and Filter data
- Sort data
- Input and Output



[JupyterLab Exercises](#)

Pandas Cheat Sheet

Data Wrangling
with pandas
Cheat Sheet
<http://pandas.pydata.org>

Syntax – Creating DataFrames

	a	b	c
1	4	7	10
2	5	8	11
3	6	9	12

```
df = pd.DataFrame(
    {"a": [4, 5, 6],
     "b": [7, 8, 9],
     "c": [10, 11, 12]},
    index = [1, 2, 3])
```

Specify values for each column.

```
df = pd.DataFrame(
    [[4, 7, 10],
     [5, 8, 11],
     [6, 9, 12]],
    index=[1, 2, 3],
    columns=['a', 'b', 'c'])
```

Specify values for each row.

	a	b	c
1	4	7	10
2	5	8	11
3	6	9	12

```
df = pd.DataFrame(
    {"a": [4, 5, 6],
     "b": [7, 8, 9],
     "c": [10, 11, 12]},
    index = pd.MultiIndex.from_tuples(
        [(1, 'a'), (2, 'b'), (3, 'c')],
        names=['n', 'v'])
```

Create DataFrame with a MultiIndex

Method Chaining

Most pandas methods return a DataFrame so that another pandas method can be applied to the result. This improves readability of code.

```
df = (pd.melt(df)
     .rename(columns={
         'variable': 'var',
         'value': 'val'})
     .query('val > 200'))
```

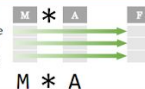
Tidy Data – A foundation for wrangling in pandas

In a tidy data set:

Each variable is saved in its own column

Each observation is saved in its own row

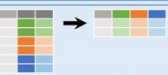
Tidy data complements pandas's **vectorized operations**. pandas will automatically preserve observations as you manipulate variables. No other format works as intuitively with pandas.



Reshaping Data – Change the layout of a data set



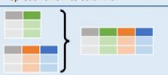
pd.melt(df)
Gather columns into rows.



df.pivot(columns='var', values='val')
Spread rows into columns.



pd.concat([df1, df2])
Append rows of DataFrames



pd.concat([df1, df2], axis=1)
Append columns of DataFrames

df.sort_values('mpg')
Order rows by values of a column (low to high).

df.sort_values('mpg', ascending=False)
Order rows by values of a column (high to low).

df.rename(columns = {'y': 'year'})
Rename the columns of a DataFrame

df.sort_index()
Sort the index of a DataFrame

df.reset_index()
Reset index of DataFrame to row numbers, moving index to columns.

df.drop(columns=['Length', 'Height'])
Drop columns from DataFrame

Subset Observations (Rows)



df[df.Length > 7]
Extract rows that meet logical criteria.

df.drop_duplicates()
Remove duplicate rows (only considers columns).

df.head(n)
Select first n rows.

df.tail(n)
Select last n rows.

df.sample(frac=0.5)
Randomly select fraction of rows.

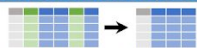
df.sample(n=10)
Randomly select n rows.

df.iloc[10:20]
Select rows by position.

df.nlargest(n, 'value')
Select and order top n entries.

df.nsmallest(n, 'value')
Select and order bottom n entries.

Subset Variables (Columns)



df[['width', 'length', 'species']]
Select multiple columns with specific names.

df['width'] or df.width
Select single column with specific name.

df.filter(regex='regex')
Select columns whose name matches regular expression regex.

df.loc[:, 'x2': 'x4']
Select all columns between x2 and x4 (inclusive).

df.iloc[:, 1:2, 5]
Select columns in positions 1, 2 and 5 (first column is 0).

df.loc[df['a'] > 10, ['a', 'c']]
Select rows meeting logical condition, and only the specific columns.

regex (Regular Expressions) Examples

regex	Matches
'\.'	Matches strings containing a period '.'
'Length\$'	Matches strings ending with word 'Length'
'^Sepal'	Matches strings beginning with the word 'Sepal'
'*[1-5]\$'	Matches strings beginning with 'X' and ending with 1,2,3,4,5
'^(?!Species)\$'	Matches strings except the string 'Species'

Logic in Python (and pandas)			
<	Less than	!=	Not equal to
>	Greater than	df.column.isin(values)	Group membership
==	Equals	pd.isnull(obj)	Is NaN
<=	Less than or equals	pd.notnull(obj)	Is not NaN
>=	Greater than or equals	df[['a', 'b']].all()	Logical and, or, not, xor, any, all

Summarize Data

```
df['w'].value_counts()
# Count number of rows with each unique value of variable
len(df)
```

Handling Missing Data

```
df.dropna()
# Drop rows with any column having NA/null data.
df.fillna(value)
```

Combine Data Sets

```
adf
x1 x2
A 1
+
bdf
x1 x2
A 1
=
```

Key Plotting Concepts in Matplotlib

- **Matplotlib: Figure**

Figure is the object that keeps the whole image output.

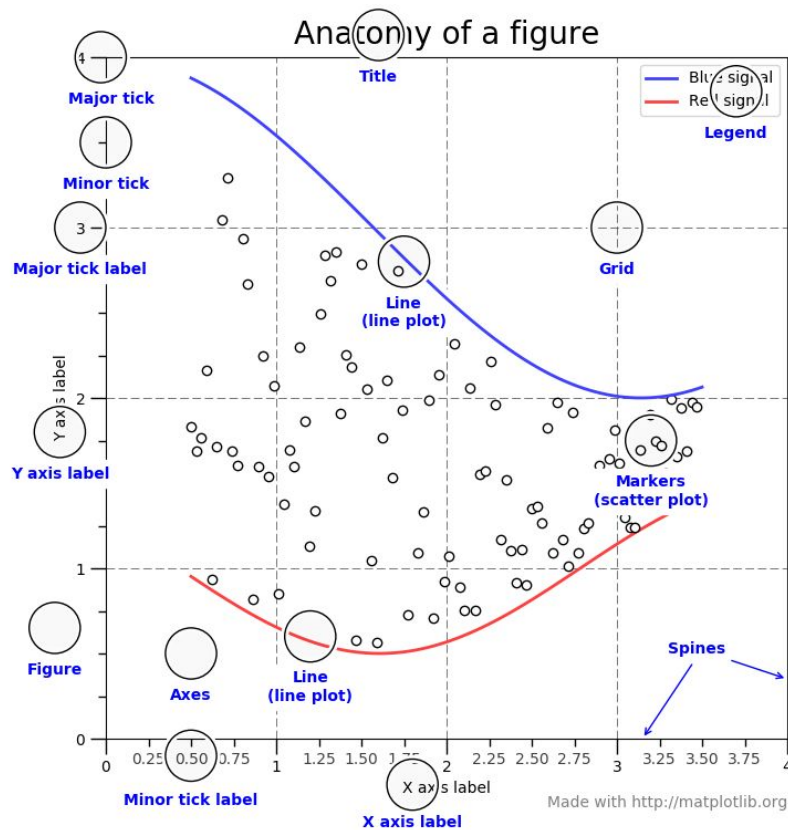
Adjustable parameters include:

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](#)

Matplotlib Cheat Sheet

Python For Data Science Cheat Sheet

Matplotlib

Learn Python interactively at [www.DataCamp.com](https://www.datacamp.com)

Matplotlib is a Python 2D plotting library which produces publication-quality figures in a variety of hardcopy formats and interactive environments across platforms.

1 Prepare The Data

Also see Lists & NumPy

1D Data

```
>>> import numpy as np
>>> x = np.linspace(0, 10, 100)
>>> y = np.cos(x)
>>> z = np.sin(x)
```

2D Data or Images

```
>>> data = 2 * np.random.random([10, 10])
>>> data2 = 3 * np.random.random([10, 10])
>>> T, X = np.meshgrid(np.linspace(0, 1, 10),
>>>                    np.linspace(0, 1, 10))
>>> U = 1 + X**2 + Y
>>> V = 1 + X + Y**2
>>> from matplotlib.cbook import get_sample_data
>>> img = np.load(get_sample_data('axvlines_demo.mat'))
```

2 Create Plot

```
>>> import matplotlib.pyplot as plt
```

Figure

```
>>> fig = plt.figure()
>>> fig2 = plt.figure(figsize=plt.figaspect(2.0))
```

Axes

All plotting is done with respect to an Axes. In most cases, a subplot will fit your needs. A subplot is an axes on a grid system.

```
>>> fig, ax = plt.subplots()
>>> ax1 = fig.add_subplot(221) # row=col=2
>>> ax2 = fig.add_subplot(222)
>>> fig3, axes = plt.subplots(nrows=2, ncols=2)
>>> fig4, axes2 = plt.subplots(ncols=3)
```

3 Plotting Routines

1D Data

```
>>> lines = ax.plot(x, y)
>>> ax.scatter(x, y)
>>> axes[0,0].bar([1, 2, 3], [3, 4, 5])
>>> axes[1,0].barh([0.5, 1, 2], [1, 0.5, 1])
>>> axes[1,1].axhline(0.45)
>>> axes[0,1].axvline(0.45)
>>> ax.fill(x, y, label='blue')
>>> ax.fill_between(x, y, color='yellow')
```

Draw points with lines or markers connecting them
Draw unconnected points, scaled or colored
Plot vertical rectangles (constant width)
Plot horizontal rectangles (constant height)
Draw a horizontal line across axes
Draw a vertical line across axes
Draw filled polygons
Fill between y-values and x

2D Data or Images

```
>>> fig, ax = plt.subplots()
>>> im = ax.imshow(img, cmap=plt.cm.winter,
>>>                interpolation='nearest',
>>>                vmin=2,
>>>                vmax=2)
```

Colormapped or RGB arrays

Vector Fields

```
>>> axes[0,1].arrow(0, 0, 0.5, 0.5)
>>> axes[1,1].quiver(y, z)
>>> axes[0,1].streamplot(X, Y, U, V)
```

Add an arrow to the axes
Plot a 2D field of arrows
Plot 2D vector fields

Data Distributions

```
>>> ax1.hist(y)
>>> ax1.boxplot(y)
>>> ax3.violinplot(z)
```

Plot a histogram
Make a box and whisker plot
Make a violin plot

Pseudocolor

```
>>> axes[0,1].pcolor(data2)
>>> axes[0,1].pcolormesh(data)
>>> C0 = plt.contour(x, z)
>>> axes[2,1].contour(data1)
>>> axes[2,1].contourf(data1)
>>> axes[2,1].ax.contour(C0)
```

Pseudocolor plot of 2D array
Pseudocolor plot of 2D array
Plot contours
Plot filled contours
Label a contour plot

Plot Anatomy & Workflow

Plot Anatomy

Workflow

The basic steps to creating plots with matplotlib are:

- 1 Prepare data
- 2 Create plot
- 3 Plot
- 4 Customize plot
- 5 Save plot
- 6 Show plot

```
>>> import matplotlib.pyplot as plt
>>> x = [1, 2, 4]
>>> y = [10, 20, 25, 30]
>>> fig = plt.figure()
>>> ax = fig.add_subplot(111)
>>> ax.plot(x, y, color='lightblue', linewidth=3)
>>> ax.scatter([2, 4], [15, 25],
>>>           color='darkgreen',
>>>           marker='*')
>>> ax.set_xlim(1, 4.5)
>>> plt.savefig('foo.png')
>>> plt.show()
```

4 Customize Plot

Colors, Color Bars & Color Maps

```
>>> plt.plot(x, y, x**2, x, x**3)
>>> ax.plot(x, y, alpha=0.4)
>>> ax.plot(x, y, c='k')
>>> fig.colorbar(orientation='horizontal')
>>> im = ax.imshow(img, cmap='seismic')
```

Markers

```
>>> fig, ax = plt.subplots()
>>> ax.scatter(x, y, marker='*')
>>> ax.plot(x, y, marker='*')
```

Linestyles

```
>>> plt.plot(x, y, linewidth=4.0)
>>> plt.plot(x, y, ls='solid')
>>> plt.plot(x, y, ls='-', x**2, y**2, '-.')
>>> plt.setp(lines, color='r', linewidth=4.0)
```

Text & Annotations

```
>>> ax.text(-2, 1,
>>>         'Example Graph',
>>>         style='italic')
>>> ax.annotate('Data',
>>>            xy=(8, 0),
>>>            xycoords='data',
>>>            textcoords='axesfraction',
>>>            xybox=(10, 0.5, 0.1),
>>>            textcoords='data',
>>>            arrowprops=dict(arrowstyle="->",
>>>                            connectionstyle="arc3"))
```

Mattext

```
>>> plt.title('Sigma_i=100', fontsize=20)
```

Limits, Legends & Layouts

Limits & Autoscaling

```
>>> ax.margins(x=0.5, y=0.1)
>>> ax.axis('tight')
>>> ax.set_xlim(0, 10.5), ylim=(-1.5, 1.5)
>>> ax.set_xlim(0, 10.5)
```

Add padding to a plot
Set the aspect ratio of the plot to 1
Set limits for x and y-axis
Set limits for x-axis

Legends

```
>>> ax.set(title='An Example Axes',
>>>        ylabel='Y-Axis',
>>>        xlabel='X-Axis')
>>> ax.legend(loc='best')
```

Set a title and x and y-axis labels
No overlapping plot elements
Manually set ticks
Make y-ticks longer and go in and out

Ticks

```
>>> ax.xaxis.set(ticks=range(1, 5),
>>>               ticklabels=[3, 100, -12, 'foo'])
>>> ax.ticks_params()['axis'] = 'y',
>>>                  direction='inout',
>>>                  length=10)
```

Adjust the spacing between subplots

Subplot Spacing

```
>>> fig.subplots_adjust(wspace=0.5,
>>>                     hspace=0.3,
>>>                     left=0.125,
>>>                     right=0.9,
>>>                     top=0.9,
>>>                     bottom=0.1)
```

Adjust the spacing between subplots

Axes Spines

```
>>> ax.spines['top'].set_visible(False)
>>> ax.spines['bottom'].set_position(('outward', 10))
```

Make the top axis line for a plot invisible
Move the bottom axis line outward

5 Save Plot

```
>>> plt.savefig('foo.png')
>>> plt.savefig('foo.png', transparent=True)
```

Save figures
Save transparent figures

6 Show Plot

```
>>> plt.show()
```

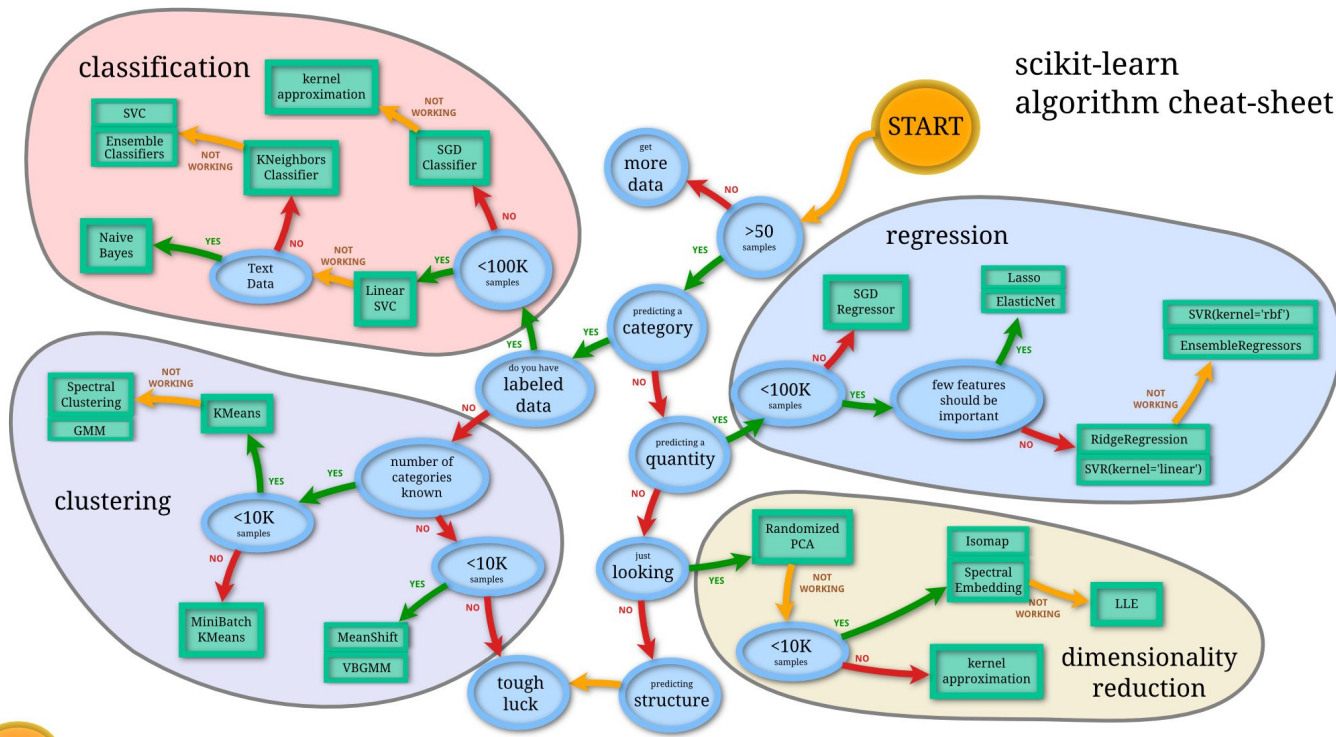
Close & Clear

```
>>> plt.clf()
>>> plt.cla()
>>> plt.close()
```

Clear an axis
Clear the entire figure
Close a window

DataCamp
Learn Python for Data Science interactively

Lab III. Machine Learning



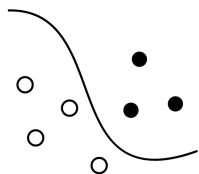
Main Features of scikit-learn



Classification

Identifying category of an object

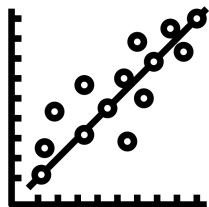
Applications: Spam detection, image recognition.
Algorithms: SVM, nearest neighbors, random forest, and more...



Regression

Predicting a attribute for an object

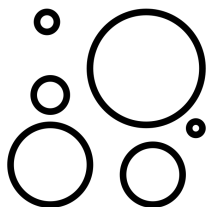
Applications: Drug response, Stock prices.
Algorithms: SVR, nearest neighbors, random forest, and more...



Clustering

Grouping similar objects into sets

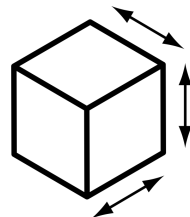
Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, and more...



Dimension Reduction

Reducing the number of dimensions

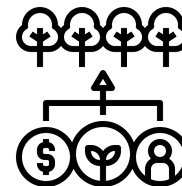
Applications: Visualization, Increased efficiency
Algorithms: k-Means, feature selection, non-negative matrix factorization, and more...



Model Selection

Selecting models with parameter search

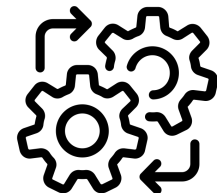
Applications: Improved accuracy via parameter tuning
Algorithms: grid search, cross validation, metrics, and more...



Preprocessing

Preprocessing data to prepare for modeling

Applications: Transforming input data such as text for use with machine learning algorithms.
Algorithms: preprocessing, feature extraction, and more...



Lab IV. Deep Learning

Deep Learning

by Ian Goodfellow, Yoshua Bengio, and Aaron Courville

<http://www.deeplearningbook.org/>

Animation of Neutron Networks

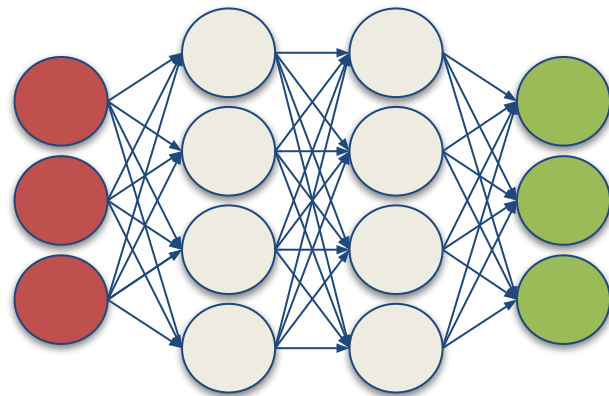
by Grant Sanderson

<https://www.3blue1brown.com/>

Visualization of CNN

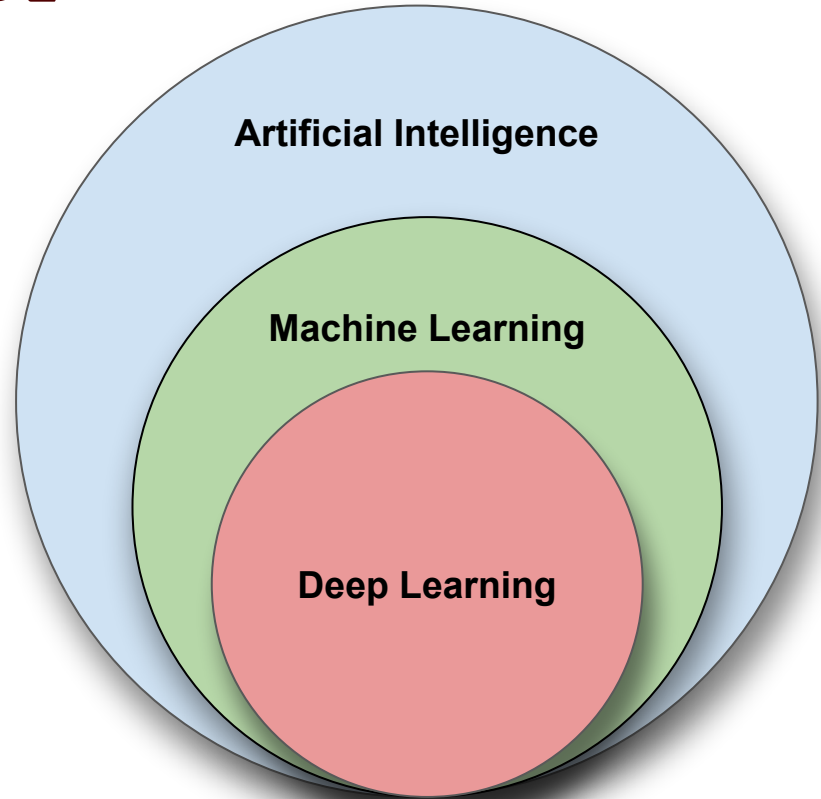
by Adam Harley

https://adamharley.com/nn_vis/cnn/3d.html



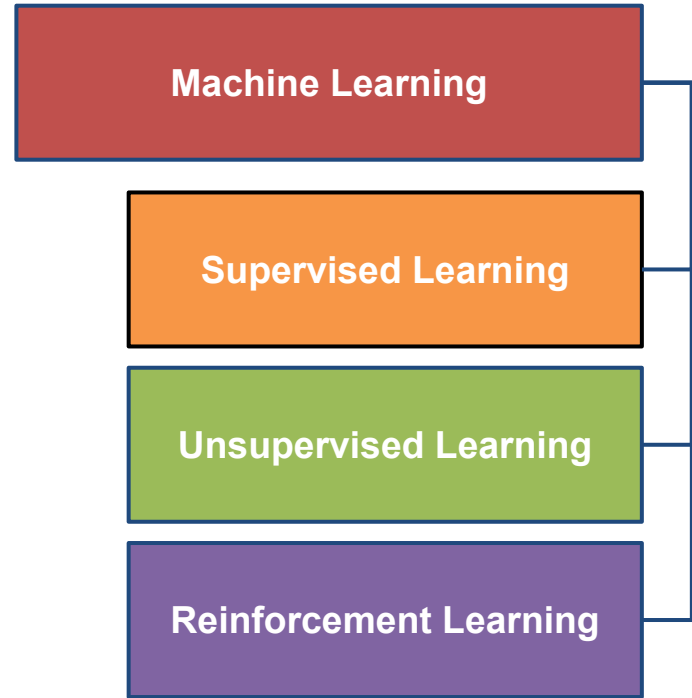
Relationship of AI, ML, and DL

- **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**.



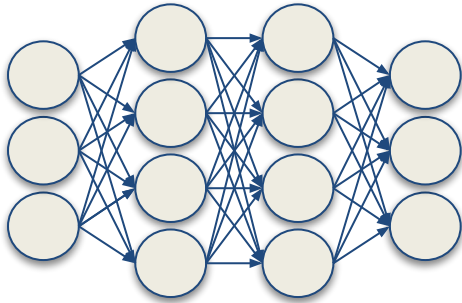
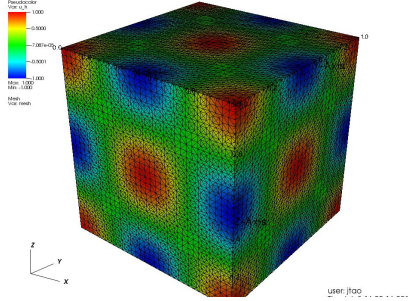
Types of ML Algorithms

- **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.

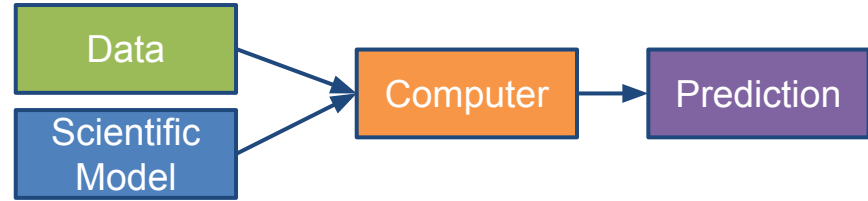


Machine Learning

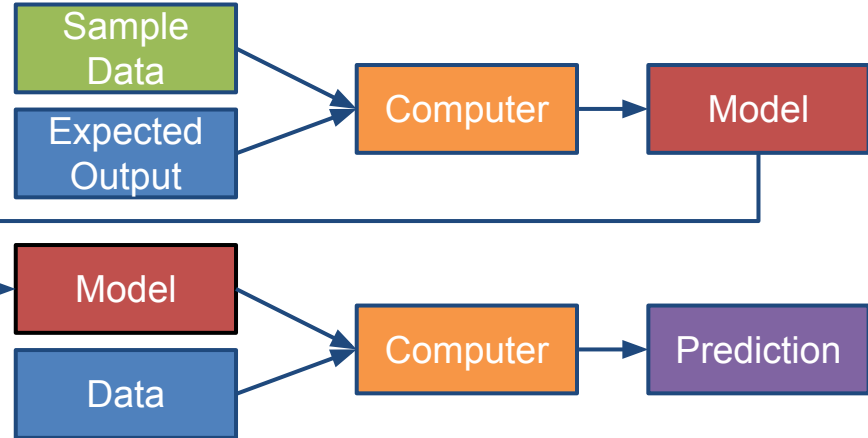
DB: simplest.vtk



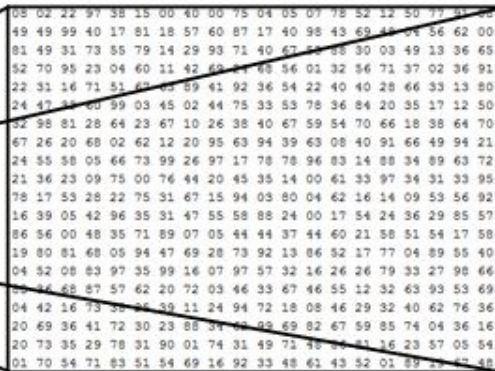
Traditional Modeling



Machine Learning (Supervised Learning)



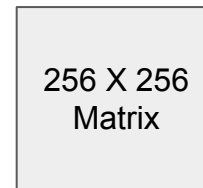
Inputs and Outputs



What the computer sees

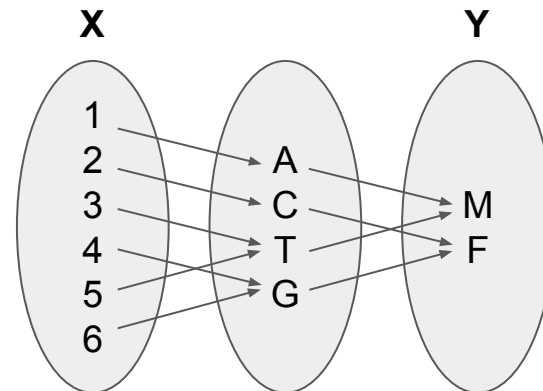
image classification → 82% cat
15% dog
2% hat
1% mug

Image from the [Stanford CS231 Course](#)



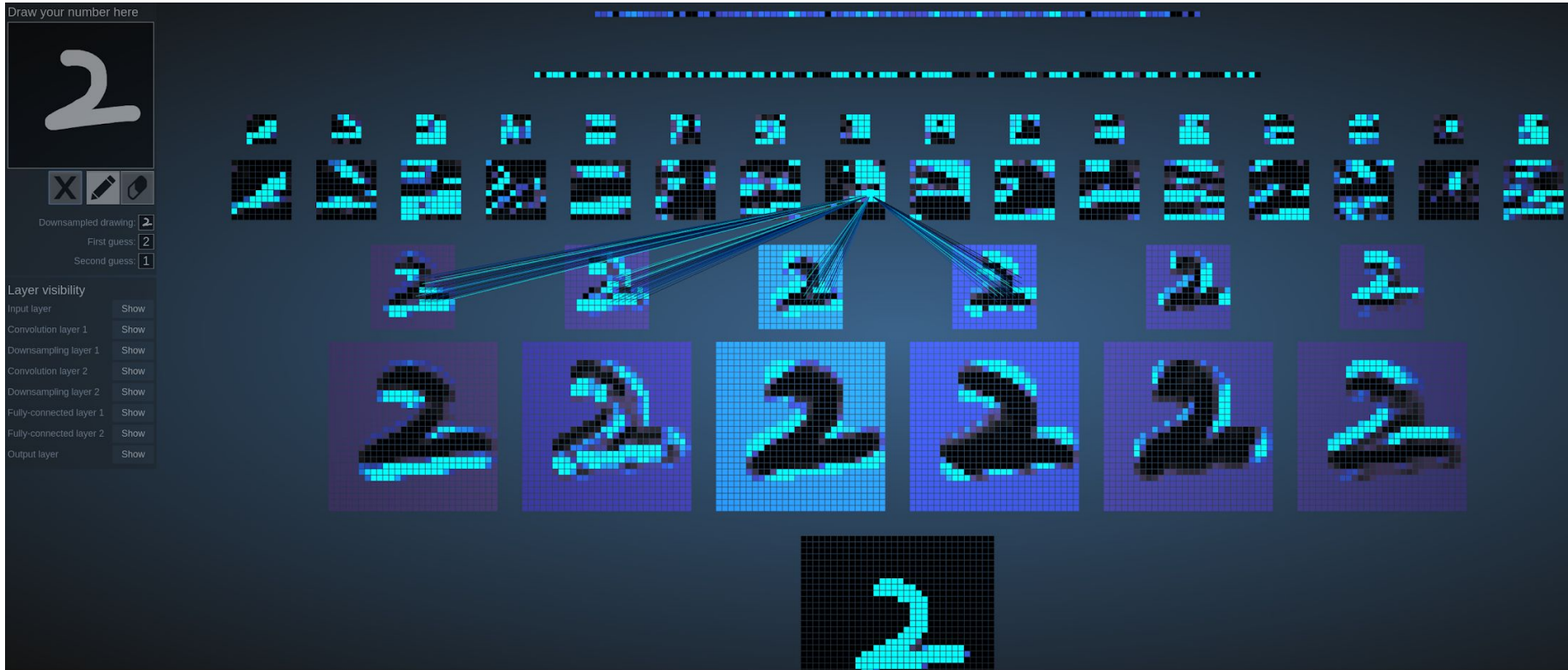
DL model

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



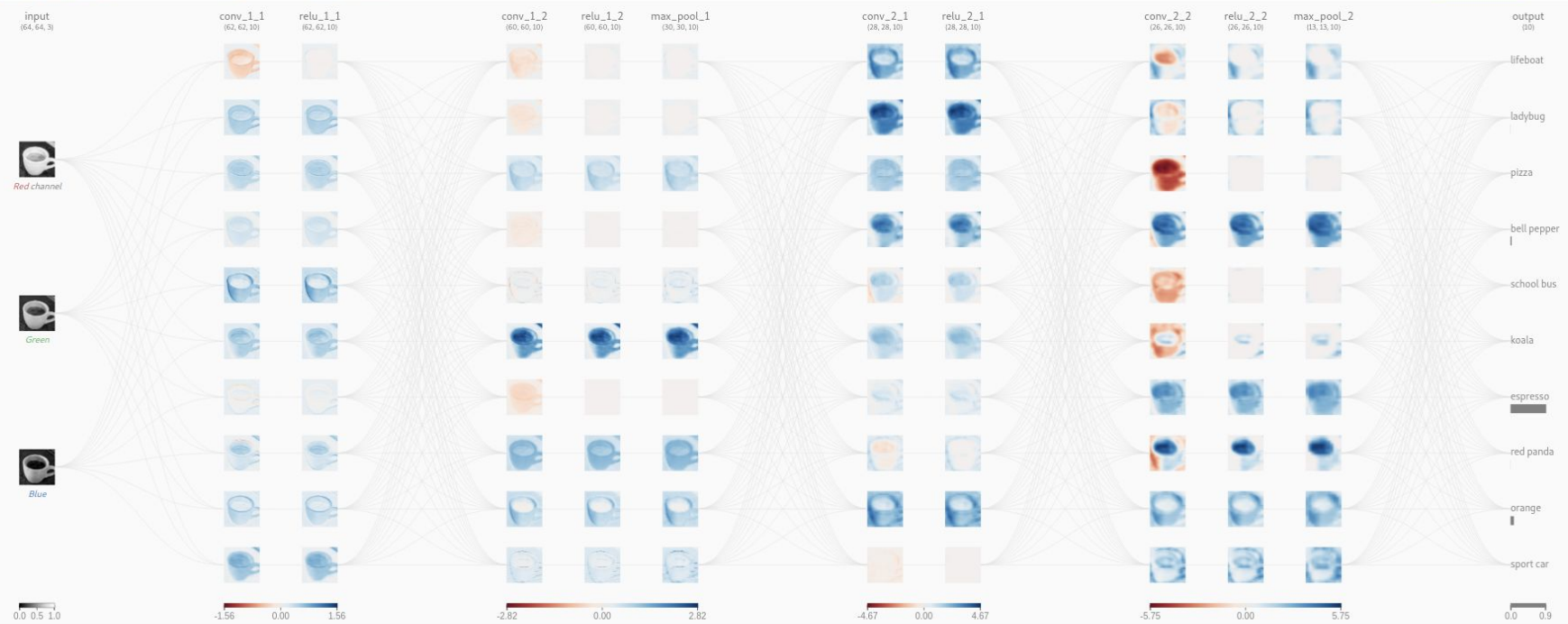
(Image Credit: https://adamharley.com/nn_vis/cnn/3d.html)

CNN Explainer

CNN EXPLAINER Learn Convolutional Neural Network (CNN) in your browser!

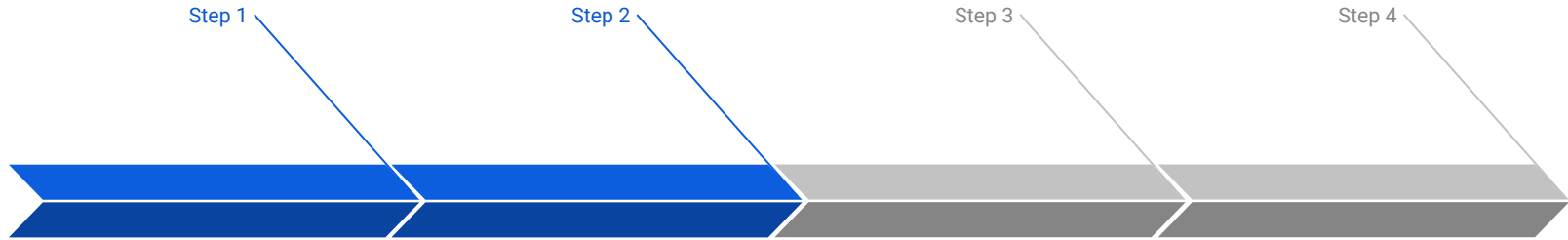


Show detail Unit



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

Machine Learning Workflow with Keras



Prepare Train Data

The preprocessed data set needs to be shuffled and splitted into training and testing data.

Define Model

A model could be defined with Keras Sequential model for a linear stack of layers or Keras functional API for complex network.

Training Configuration

The configuration of the training process requires the specification of an optimizer, a loss function, and a list of metrics.

Train Model

The training begins by calling the fit function. The number of epochs and batch size need to be set. The measurement metrics need to be evaluated.

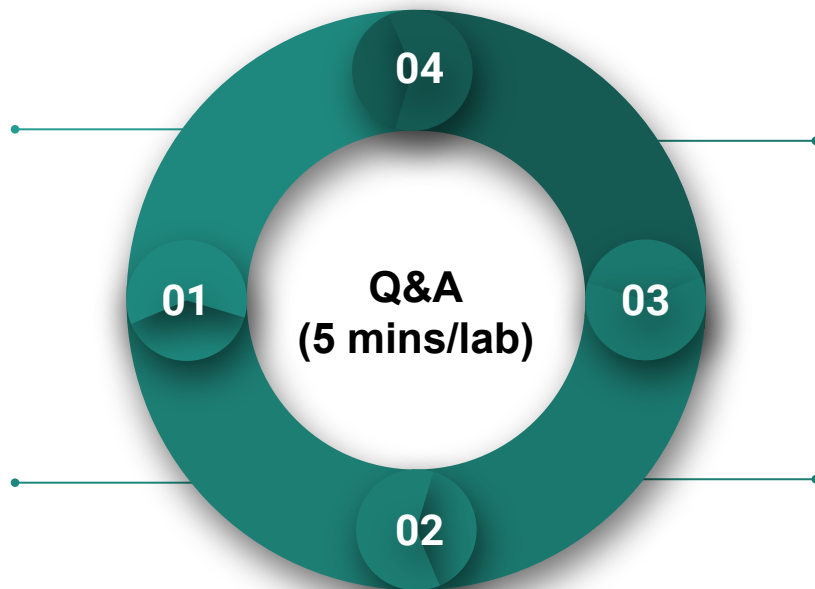
AI Tech Labs

Lab I. JupyterLab (30 mins)

We will load required modules with Jupyter Lmode extension and run JupyterLab on HPRC 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 Keras 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 Technology Labs.