Lecture 2: Data Labelling, Augmentation and Visualization

Wei Qi Yan

Auckland University of Technology

March 2, 2023

Table of Contents

What Is Deep Learning?

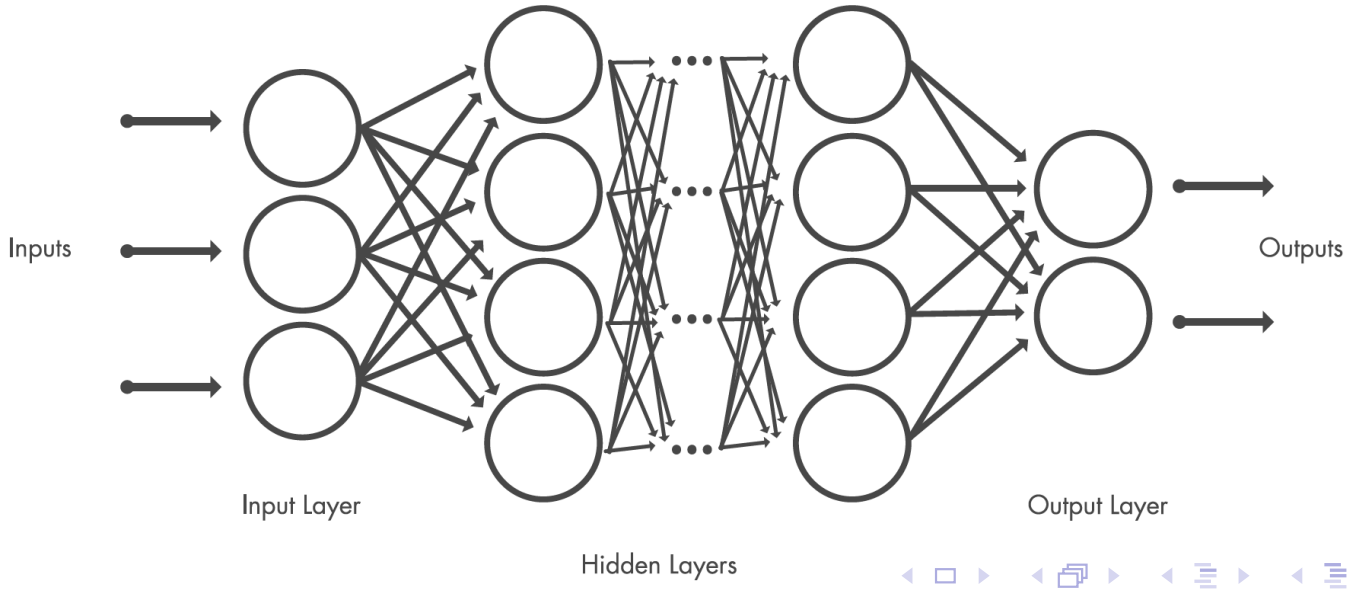
Deep Learning

- Deep learning is a branch of machine learning that trains computers to do what comes naturally to humans: Learn from experience.
- Machine learning algorithms use computational methods to "learn" information directly from data without relying on a predetermined equation as a model.
- Deep learning is especially suitable for computer vision, which is important for solving problems such as face recognition, motion detection, and many advanced driver assistance technologies.

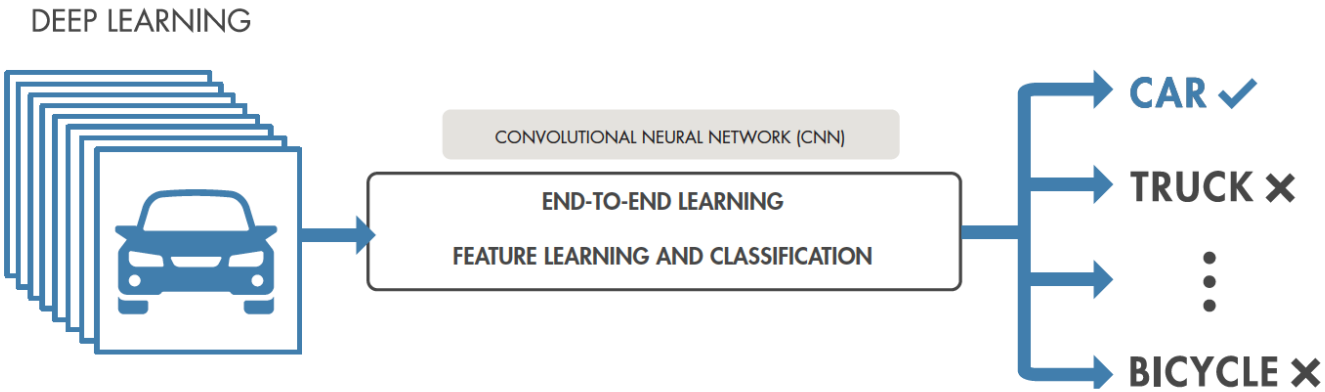
Web: Mark Hudson Beale Martin T. Hagan Howard B. Demuth. MATLAB R2019b Deep Learning Toolbox User's Guide.

Deep Neural Networks(DNNs)

- A deep neural network combines multiple nonlinear processing layers, using simple element operating in parallel and inspired by biological nervous systems.
- A deep neural network consists of an input layer, several hidden layers, and an output layer.
- The layers are interconnected via nodes, or neurons, with each hidden layer using the output of previous layer as its input.



Machine Learning(ML) *vs* Deep Learning(DL)

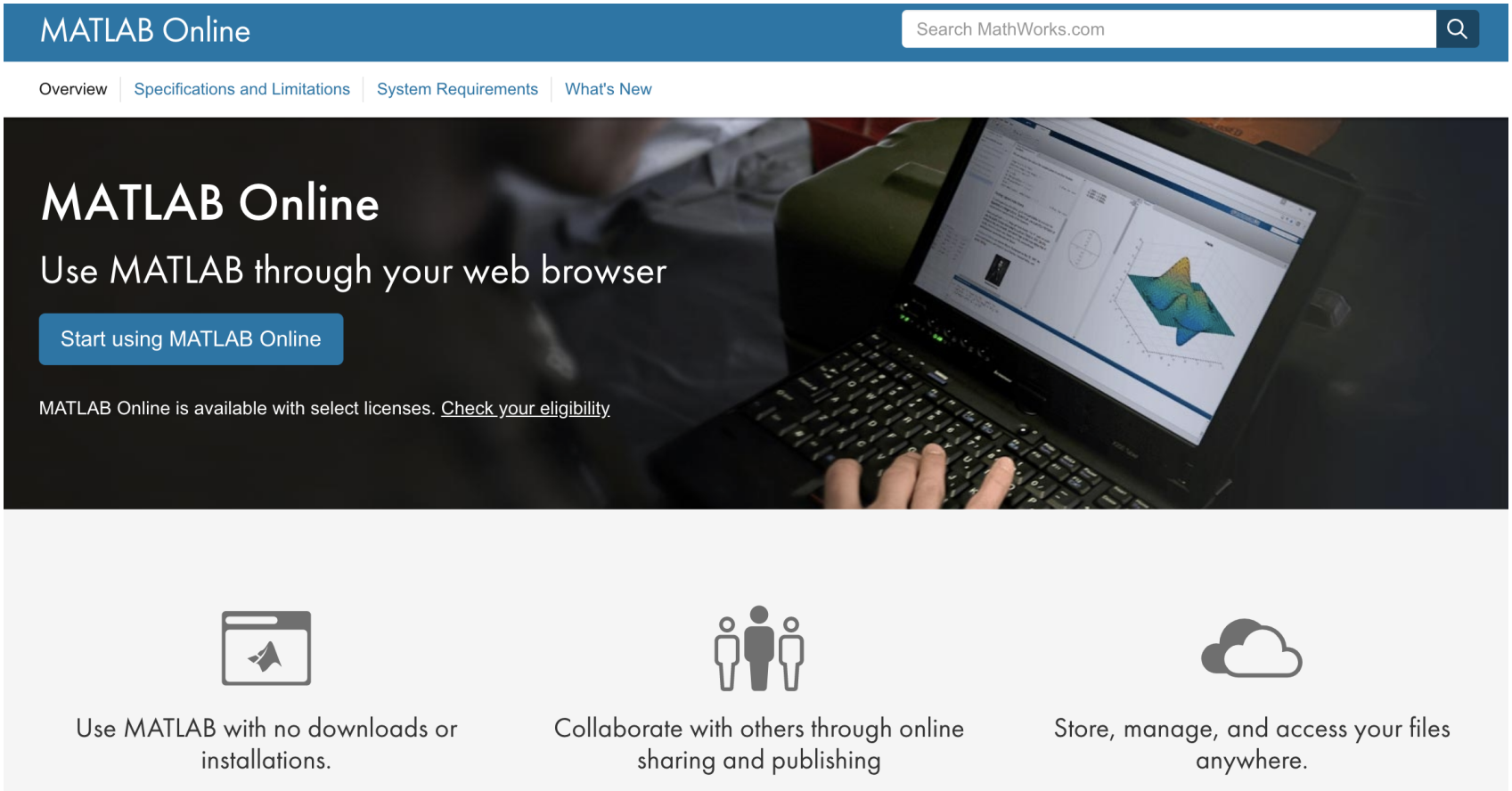


Machine Learning	Deep Learning
+ Good results with small data sets	— Requires very large data sets
+ Quick to train a model	— Computationally intensive
— Need to try different features and classifiers to achieve best results	+ Learns features and classifiers automatically
— Accuracy plateaus	+ Accuracy is unlimited

Questions?

Web: Mark Hudson Beale Martin T. Hagan Howard B. Demuth. MATLAB R2019b Deep Learning Toolbox User's Guide.

MATLAB Online



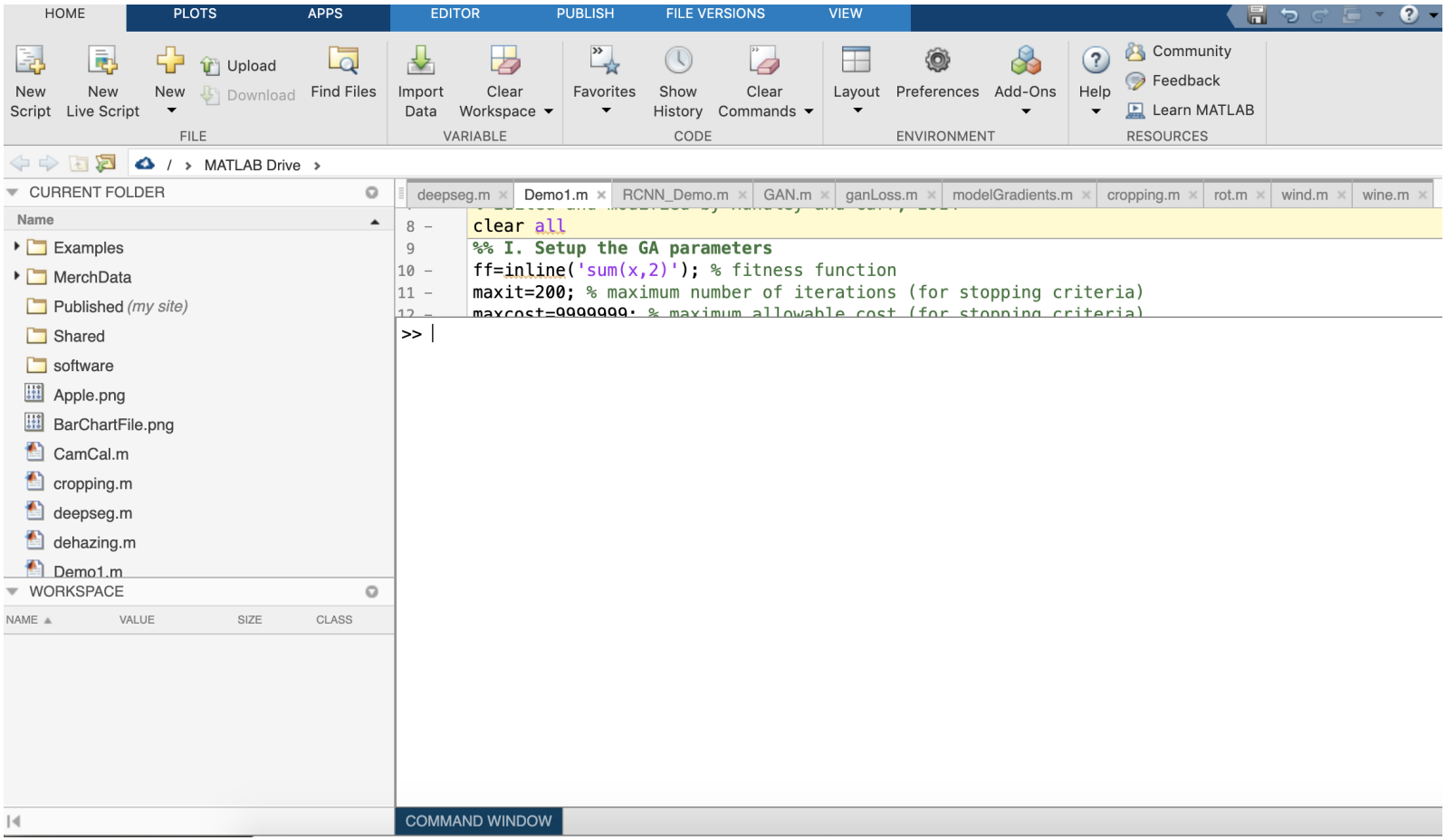
Web: https://www.mathworks.com/products/matlab-online.html

Getting Started

- MATLAB Online provides access to MATLAB from any standard web browsers.
- It is ideal for teaching and learning as well as convenient and lightweight access.
- With MATLAB Online, scripts, live scripts, and other MATLAB files are shared directly.
- MATLAB Online automatically updates to the latest version.
- There is 5GB store to access and manage files from anywhere.
- Use MATLAB Drive Connector to sync files, eliminating the need for manual upload or download.

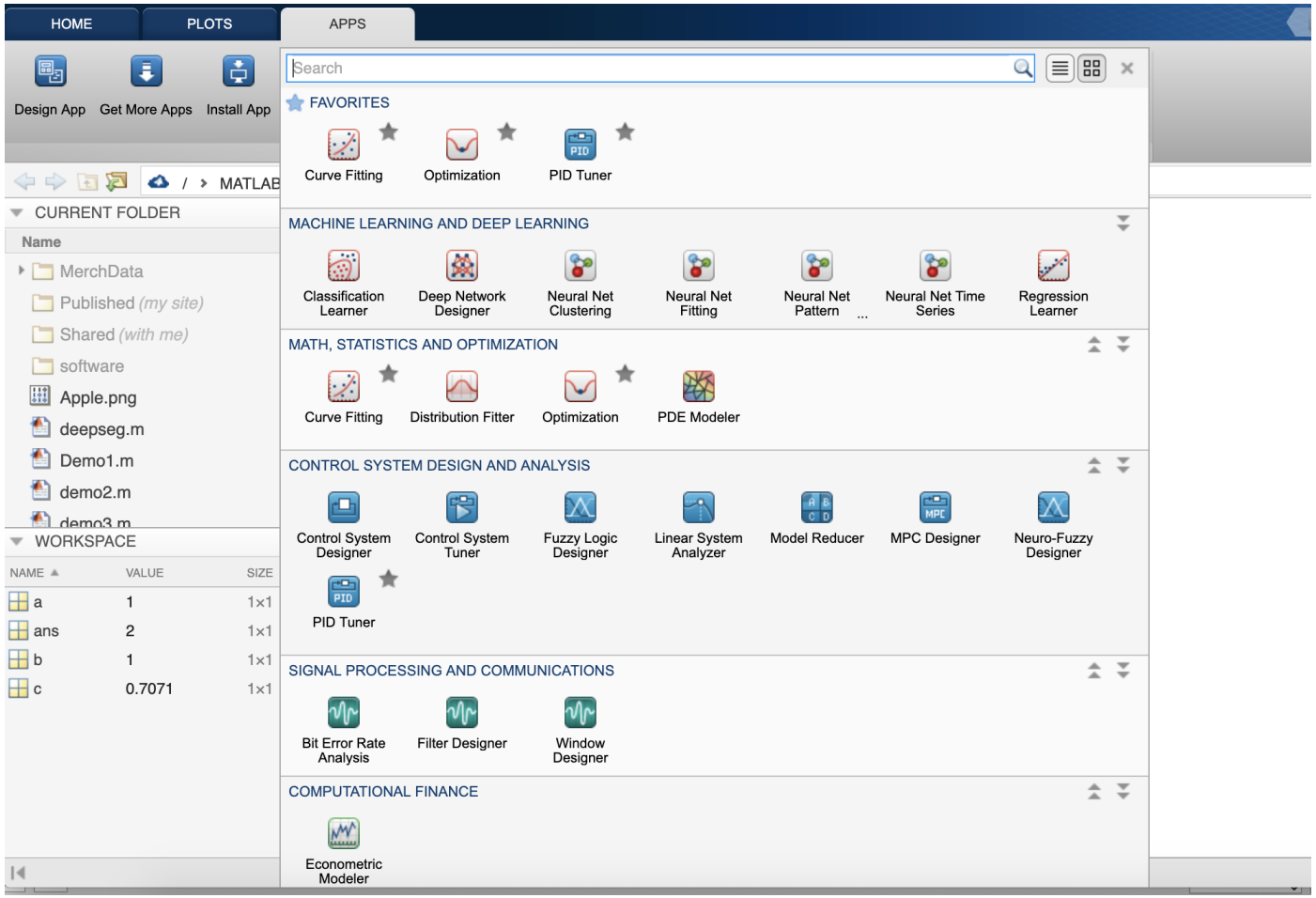
Web: https://www.mathworks.com/products/matlab-online.html

MATLAB Interface



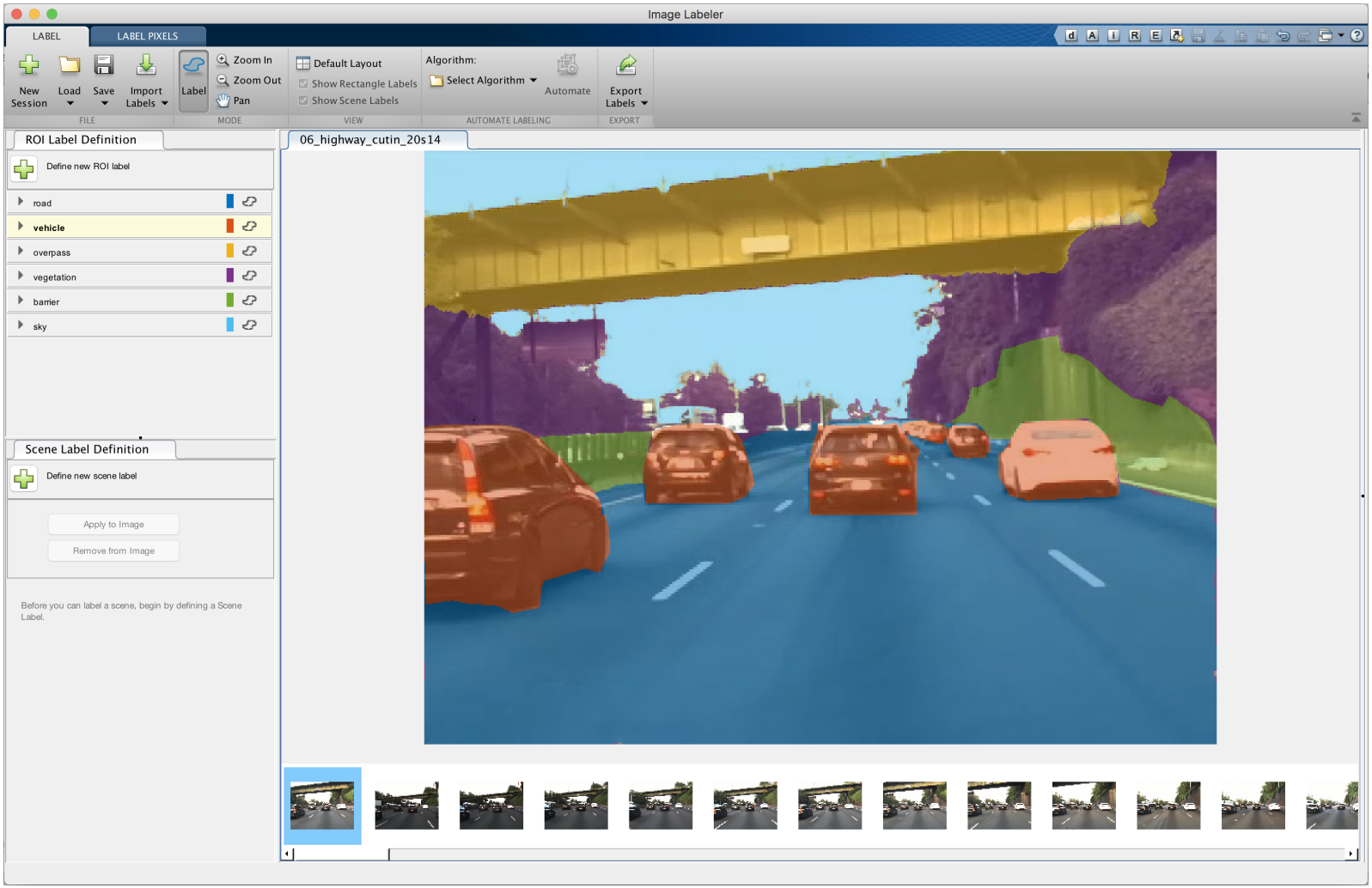
Web: https://www.mathworks.com/products/matlab-online.html

Getting Started



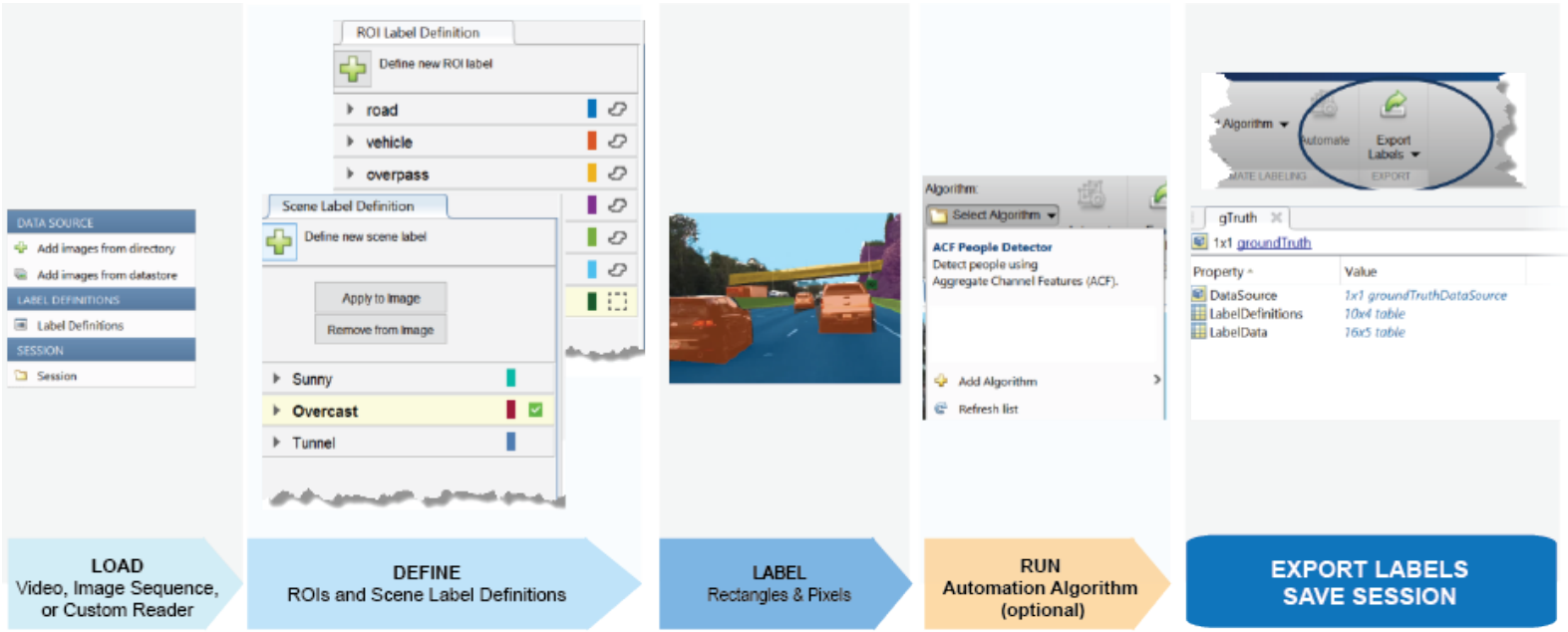
Data Labelling and Augmentation

MATLAB Image Labeler



Web: https://au.mathworks.com/help/vision/ref/imagelabeler-app.html

MATLAB Labeling Workflow



Web: https://au.mathworks.com/help/vision/ref/imagelabeler-app.html

Data Labelling and Augmentation

MATLAB Datastore for Deep Learning

- Because deep learning often requires large amounts of data, datastores are an important part of deep learning.
- Datastores can be used directly as input for network training, validation, and inference.
- The data read from these datastores must be preprocessed into a format required by a deep learning network.
- Deep learning frequently requires the data to be preprocessed and augmented before the data is in an appropriate form to input to a network.

Data Labelling and Augmentation

MATLAB Image Augmentation

- Resize images using rescaling and cropping.
- Multiply the height and width of images by using a scaling factor.
- Extract a subregion of the image and preserve the spatial extent of each pixel.
- Augment images with a random combination of resizing, rotation, reflection, shearing, and translation transformations.
- Combine datastores with image data.
- Associate image and label data.
- Transform volumetric data in image datastore.

Web: https://au.mathworks.com/help/deeplearning/ug/preprocess-images-for-deep-learning.html

MATLAB Image Augmentation: Image Resizing

>>I = imread('ngc6543a.jpg');
>>J = imresize(I, 0.5);% resize half
>> figure, imshow(I), figure, imshow(J)

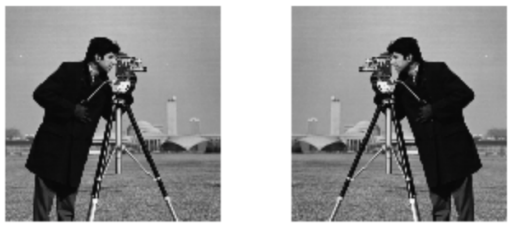
Web: https://au.mathworks.com/help/images/ref/imresize.html

MATLAB Image Augmentation: Image Reflecting

MATLAB Drive

gradients.m | ganLoss.m | GAN.m | cropping.m | reflection.m | Figure 1

```
1  function R = reflection(I,L)
2  %Author : Jeny Rajan
3  %I - Image to be Reflected
4  %L - Line position
5  %R - Reflected image
6  %eg : R = reflection(I,128);
7  % L should be between 1 and number of columns
8  [x y z]=size(I);
9  R=zeros(x,y,z);
10 R(:,L+1:y,:)=I(:,1:y-L,:);
11 R(:,1:L,:)=I(:,L:-1:1,:);
12 R=uint8(R);
13 end
```



Web: https://au.mathworks.com/matlabcentral/fileexchange/13914-image-reflection

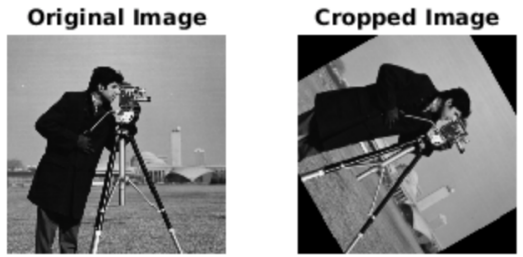
MATLAB Image Augmentation: Image Rotating

MATLAB Drive

modelGradients.m | ganLoss.m | GAN.m | cropping.m | rot.m

```
1 -  I = imread('cameraman.tif');
2
3    %I2 = imcrop(I,[75 68 130 112]);
4
5 -  I2 = imrotate(I,-60,'bilinear','crop');
6
7 -  subplot(1,2,1)
8 -  imshow(I)
9 -  title('Original Image')
10 - subplot(1,2,2)
11 - imshow(I2)
12 - title('Cropped Image')
13
14
```

Figure 1

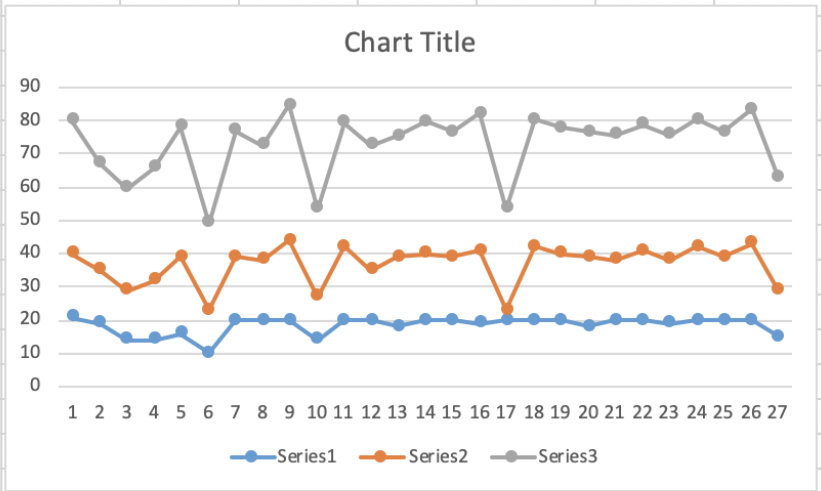


Web: https://au.mathworks.com/help/images/ref/imrotate.html

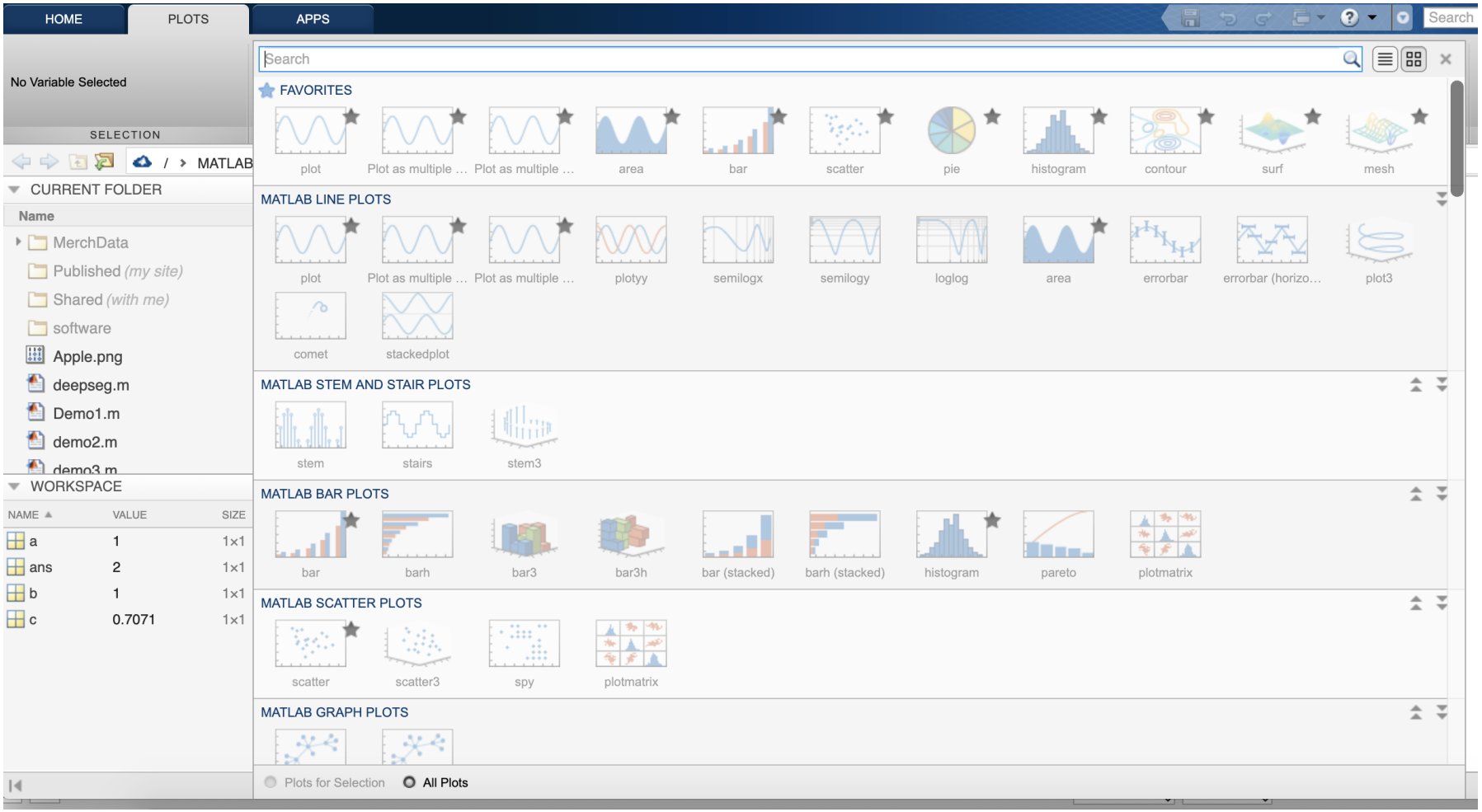
Questions?

Microsoft Office: Excel

21	19	40
19	16	32
14	15	30.75
14	18	33.75
16	23	39
10	13	26.25
20	19	38
20	18	34.75
20	24	40.75
14	13	26.5
20	22	37.25
20	15	37.5
18	21	36.25
20	20	39.75
20	19	37.75
19	22	41.25
20	3	30.5
20	22	38.25
20	20	37.75
18	21	37.75
20	18	37.75
20	21	37.75
19	19	37.75
20	22	38.25
20	19	37.75
20	23	40.25
15	14	33.75

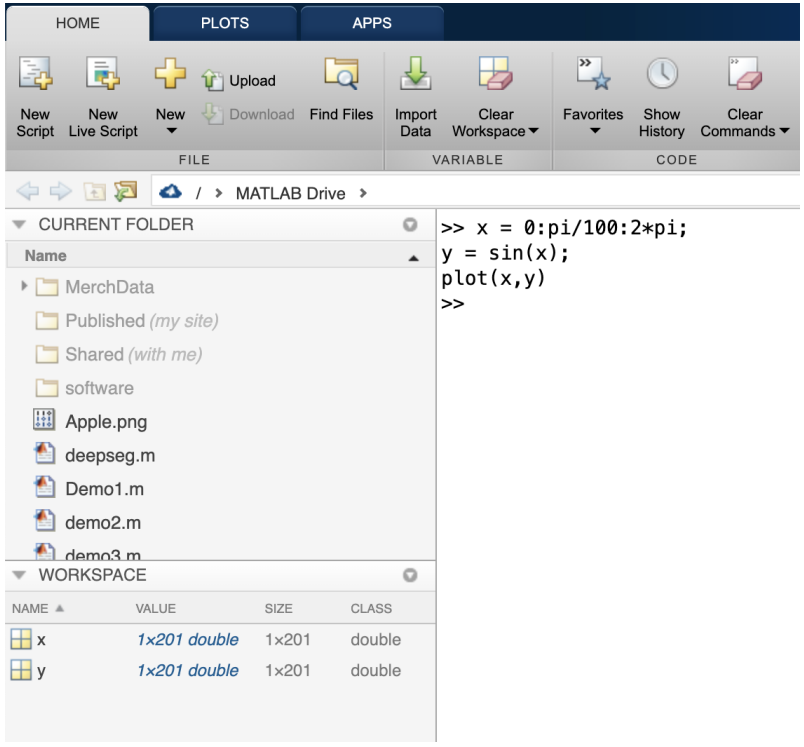


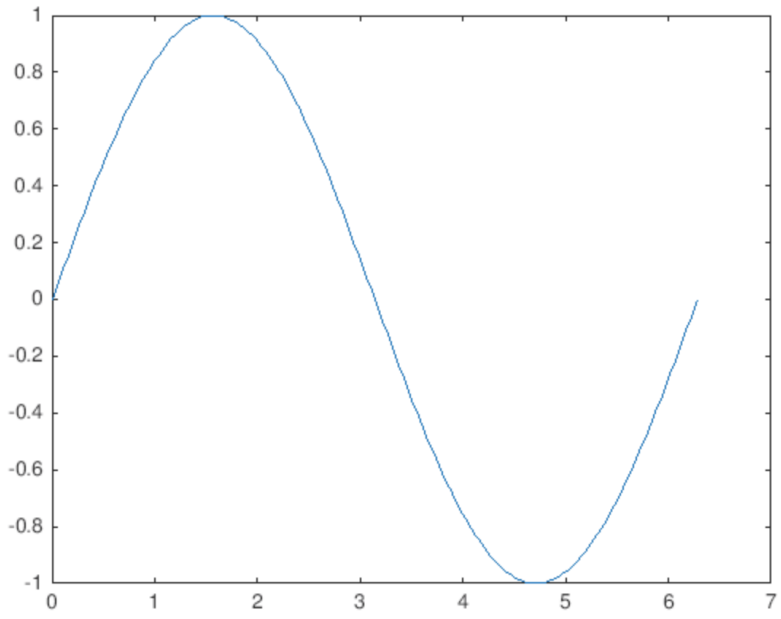
MATLAB: Data Visualization



Web: https://www.mathworks.com/products/matlab-online.html

Basics: 2D Plot

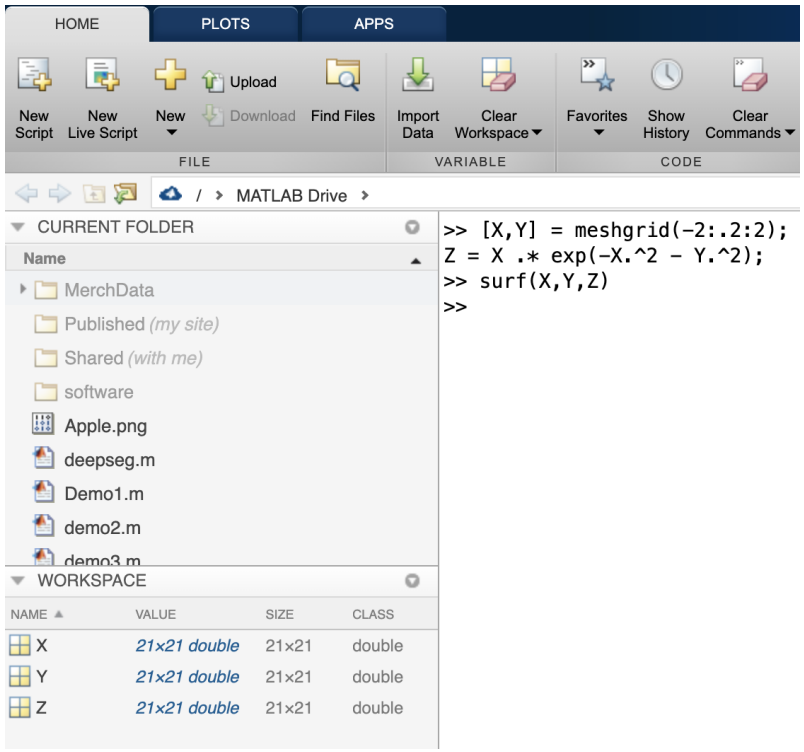


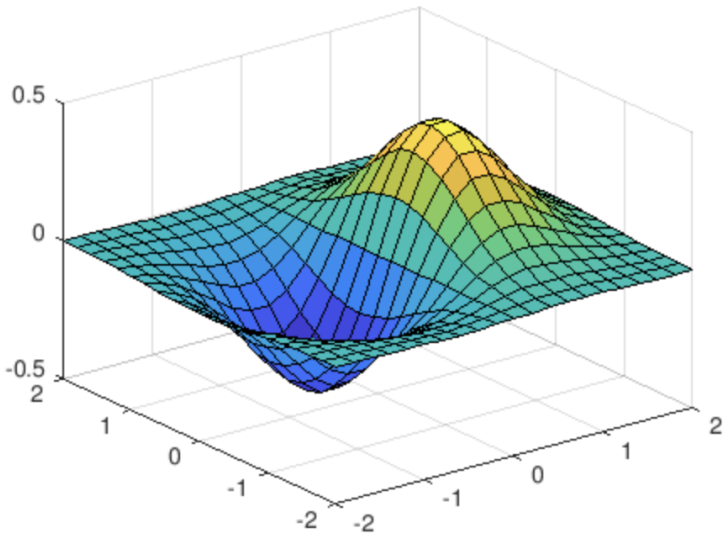


Web:
https://au.mathworks.com/help/matlab/examples.html?category=graphics&s_tid=CRUX_lftnav_examplplots

MATLAB: Data Visualization

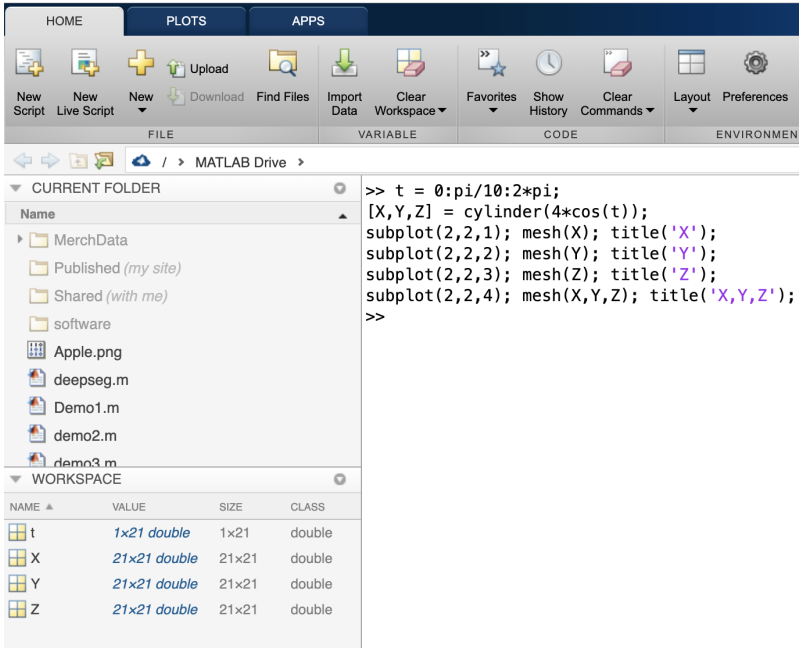
Basics: 3D Plot

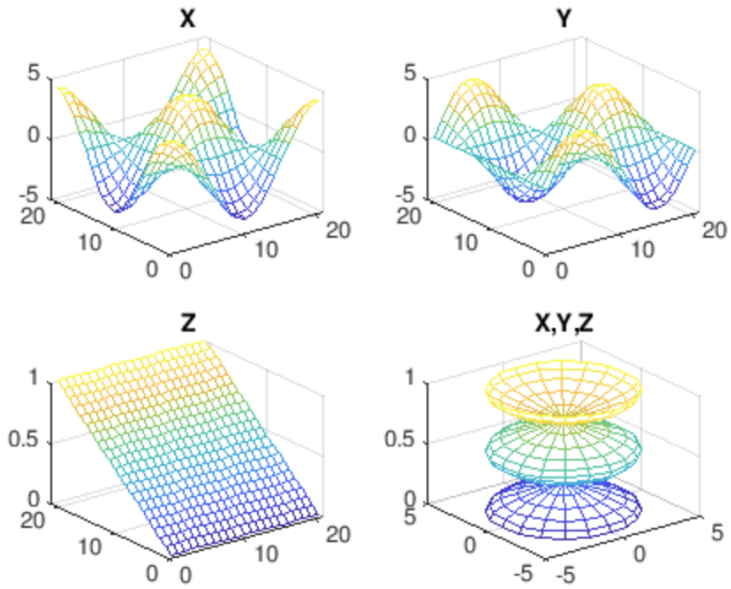




Web:
https://au.mathworks.com/help/matlab/examples.html?category=graphics&s_tid=CRUX_lftnav_examplplots

Basics: Subplot





Web:
https://au.mathworks.com/help/matlab/examples.html?category=graphics&s_tid=CRUX_lftnav_examplplots

Matplotlib: Data Visualization

matplotlib
Version 3.1.3

Installation Documentation Examples Tutorials Contributing

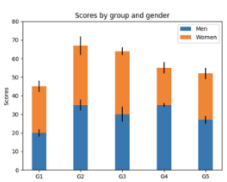
home | contents »

Gallery

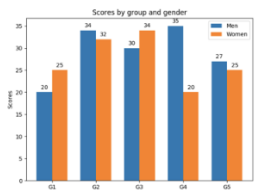
This gallery contains examples of the many things you can do with Matplotlib. Click on any image to see the full image and sou

For longer tutorials, see our tutorials page. You can also find external resources and a FAQ in our user guide.

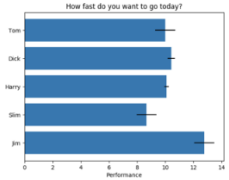
Lines, bars and markers



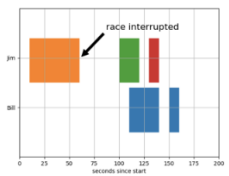
Stacked Bar Graph



Grouped bar chart with labels



Horizontal bar chart



Broken Barh

https:// matplotlib.org/gallery/mplot3d/wire3d.html♯sphx-glr-gallery-mplot3d-wire3d-py

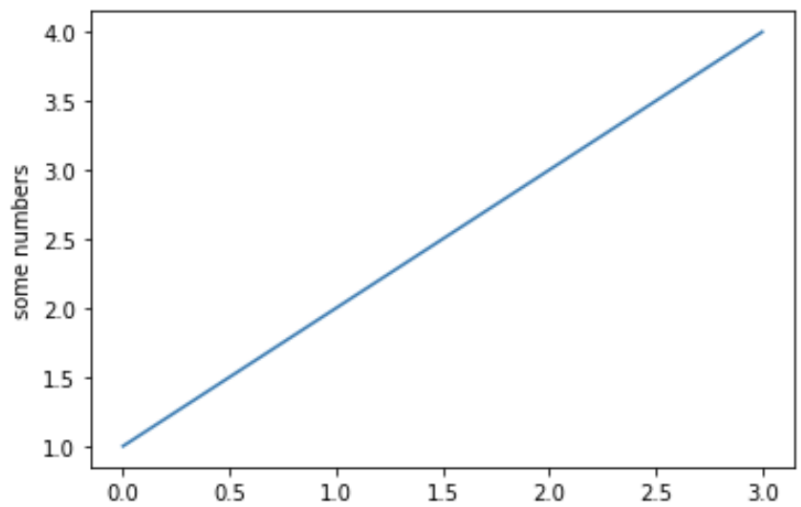
CoLaboratory: Data Visualization

Untitled0.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

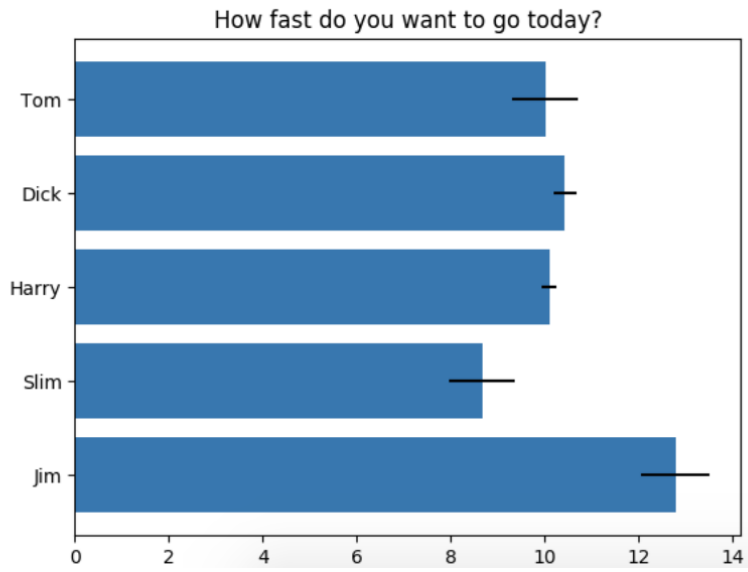
```
import matplotlib.pyplot as plt
plt.plot([1, 2, 3, 4])
plt.ylabel('some numbers')
plt.show()
```



https:// matplotlib.org/gallery/mplot3d/wire3d.html♯sphx-glr-gallery-mplot3d-wire3d-py

Matplotlib: Data Visualization

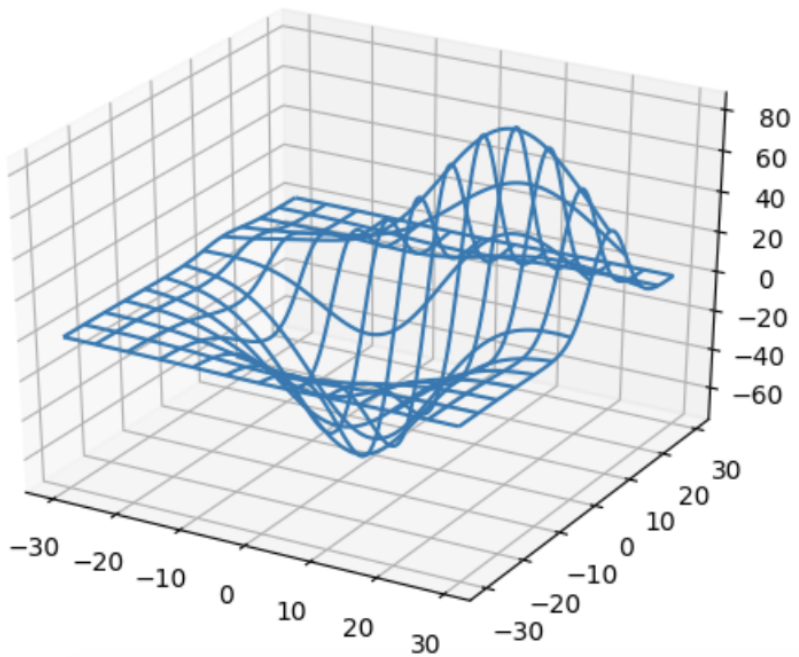
```
[1] import matplotlib.pyplot as plt
    import numpy as np

    # Fixing random state for reproducibility
    np.random.seed(19680801)


    plt.rcdefaults()
    fig, ax = plt.subplots()

    # Example data
    people = ('Tom', 'Dick', 'Harry', 'Slim', 'Jim')
    y_pos = np.arange(len(people))
    performance = 3 + 10 * np.random.rand(len(people))
    error = np.random.rand(len(people))

    ax.barh(y_pos, performance, xerr=error, align='center')
    ax.set_yticks(y_pos)
    ax.set_yticklabels(people)
    ax.invert_yaxis()  # labels read top-to-bottom
    ax.set_xlabel('Performance')
    ax.set_title('How fast do you want to go today?')

    plt.show()
```



https:// matplotlib.org/gallery/mplot3d/wire3d.html♯sphx-glr-gallery-mplot3d-wire3d-py

Matplotlib: Data Visualization

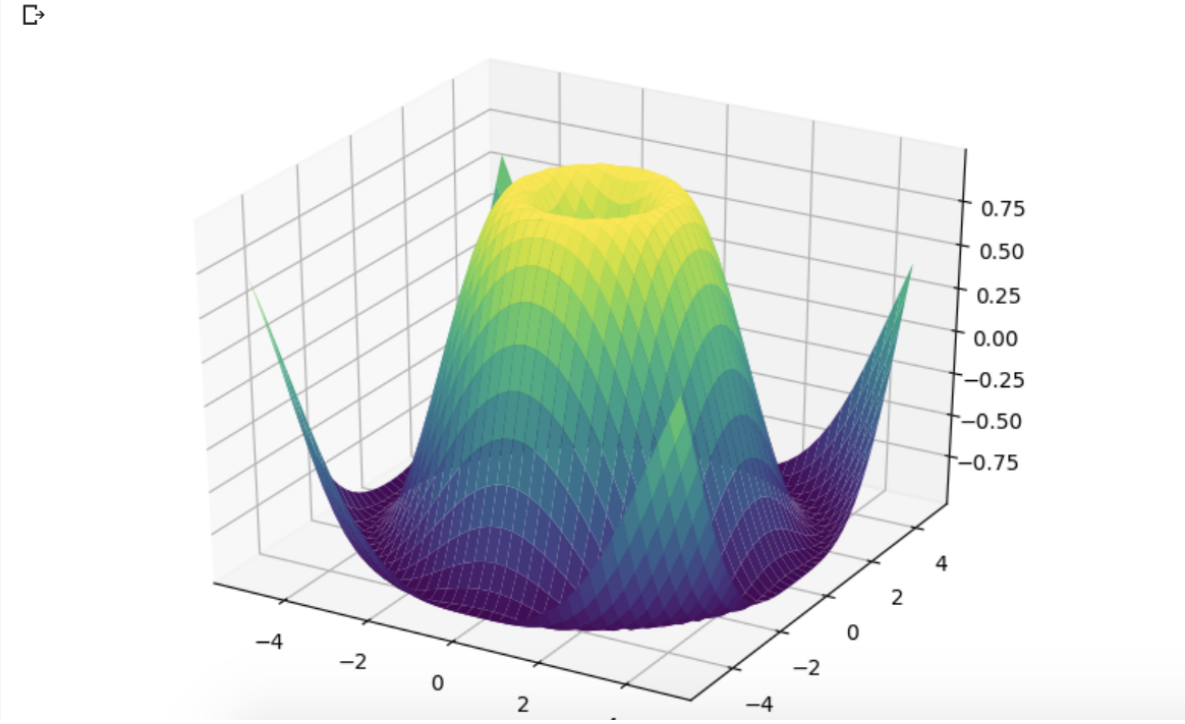
```
from mpl_toolkits.mplot3d import axes3d
import matplotlib.pyplot as plt


fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Grab some test data.
X, Y, Z = axes3d.get_test_data(0.05)

# Plot a basic wireframe.
ax.plot_wireframe(X, Y, Z, rstride=10, cstride=10)

plt.show()
```



https:// matplotlib.org/gallery/mplot3d/wire3d.html#sphx-glr-gallery-mplot3d-wire3d-py

Matplotlib: Data Visualization

```
import matplotlib.pyplot as plt
from matplotlib import cm
from mpl_toolkits.mplot3d import Axes3D

X = np.arange(-5, 5, 0.25)
Y = np.arange(-5, 5, 0.25)
X, Y = np.meshgrid(X, Y)
R = np.sqrt(X**2 + Y**2)
Z = np.sin(R)

fig = plt.figure()
ax = Axes3D(fig)
ax.plot_surface(X, Y, Z, rstride=1, cstride=1, cmap=cm.viridis)

plt.show()
```



https:// matplotlib.org/gallery/mplot3d/wire3d.html♯sphx-glr-gallery-mplot3d-wire3d-py

TensorBoard: Data Visualization

TensorFlow Graph

- TensorFlow graph: Creating the computational graph.
- The nodes (Operations) and edges (Tensors): Indicating how individual operations are composed together.
- TensorFlow collections: Storing metadata.

TensorBoard

- TensorBoard is to visualize a computation graph.
- TensorBoard needs to save the computation graph to a summary file (tf.summary.FileWriter).
- TensorBoard renders the structure of a graph visually in a browser.
- TensorBoard is launched by using:

  c:> tensorboard - -logdir="..\tensorflow\graph"

Website: https://www.tensorflow.org/

TensorBoard: Data Visualization

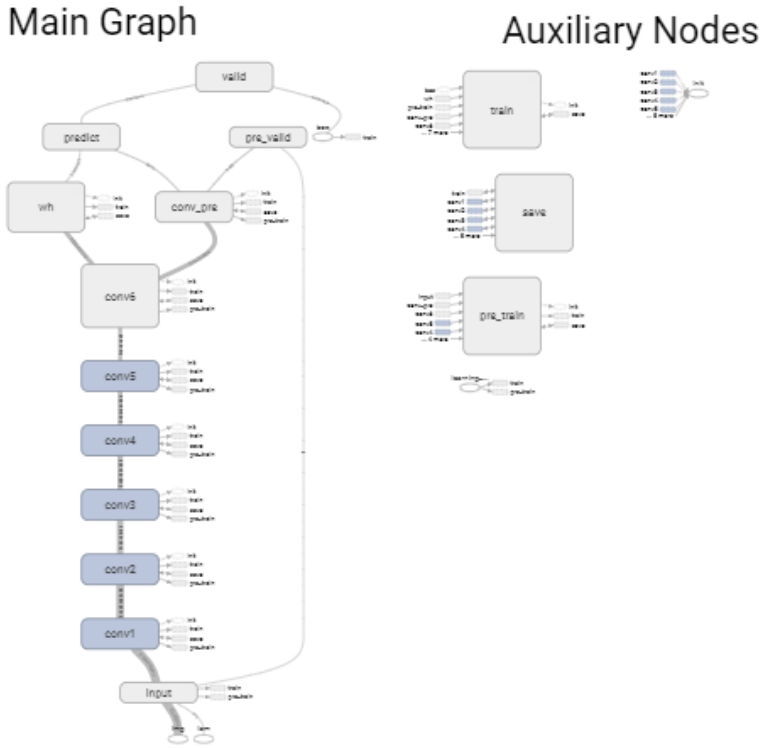
YOLO-Based Flame Detection

```
import tensorflow as tf
import numpy as np
import BuildGraph as md
import DataProvider as dp
import time
import os
import shutil

dataPath = 'pydata'
savePath = 'model/'

sess = tf.Session()
img, ism, box, lr, pre_train, pre_loss, pre_acc, train, loss, acc,
sess.run(tf.global_variables_initializer())

trainData = dp.trainData(dataPath)

if os.path.exists(savePath):
    shutil.rmtree(savePath)
os.mkdir(savePath)

sm = tf.summary.merge_all()
ckpSaver = tf.train.Saver()
ckpSaver.export_meta_graph(savePath + 'model.meta')
smWriter = tf.summary.FileWriter(savePath, sess.graph)
smWriter.add_graph(sess.graph, 0)
```

```
======== RESTART: C:\wyan\research\deep learning\SDQ_SRC\DFL\tr
ain.py ========
1 : loss = 1269.335693 , acc = 0.300000 , time = 7.203/24778.756
2 : loss = 873.849243 , acc = 0.700000 , time = 13.844/23811.254
3 : loss = 401.209290 , acc = 0.500000 , time = 20.422/23417.107
4 : loss = 248.261627 , acc = 0.600000 , time = 26.953/23179.691
5 : loss = 1315.869873 , acc = 0.200000 , time = 33.391/22972.752
6 : loss = 400.726990 , acc = 0.300000 , time = 39.719/22772.087
7 : loss = 809.821289 , acc = 0.400000 , time = 46.094/22651.789
8 : loss = 163.144196 , acc = 0.500000 , time = 52.391/22527.972
9 : loss = 305.354431 , acc = 0.400000 , time = 58.594/22395.836
10 : loss = 247.942047 , acc = 0.700000 , time = 64.859/22311.628
11 : loss = 209.520798 , acc = 0.300000 , time = 71.219/22272.049
12 : loss = 97.307373 , acc = 0.300000 , time = 77.641/22256.982
13 : loss = 192.668365 , acc = 0.300000 , time = 84.500/22360.003
14 : loss = 47.122841 , acc = 0.700000 , time = 91.156/22398.396
15 : loss = 88.590630 , acc = 0.600000 , time = 97.813/22431.670
16 : loss = 126.988731 , acc = 0.500000 , time = 104.500/22467.504
17 : loss = 29.922207 , acc = 0.700000 , time = 111.125/22486.474
18 : loss = 20.898193 , acc = 0.900000 , time = 117.484/22452.573
19 : loss = 191.536560 , acc = 0.300000 , time = 123.688/22393.951
20 : loss = 69.449844 , acc = 0.400000 , time = 129.797/22325.066
21 : loss = 81.624252 , acc = 0.600000 , time = 135.984/22275.539
22 : loss = 104.009354 , acc = 0.400000 , time = 142.469/22276.936
23 : loss = 65.351791 , acc = 0.700000 , time = 149.563/22369.351
24 : loss = 36.031826 , acc = 0.500000 , time = 157.031/22507.816
```

TensorBoard: Data Visualization

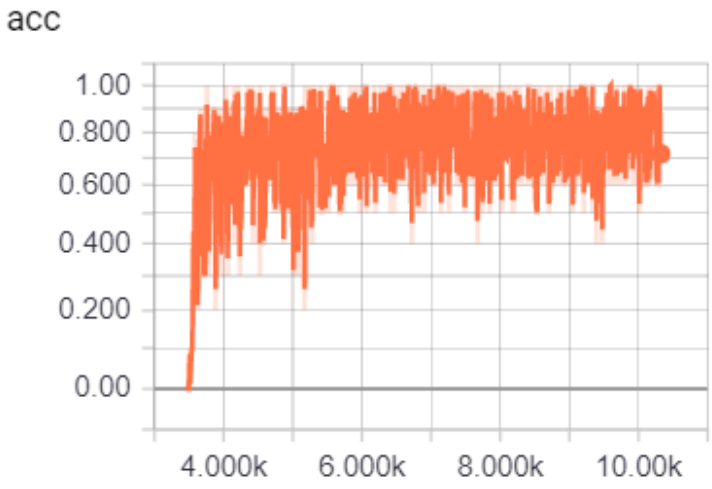
YOLO-Based Flame Detection

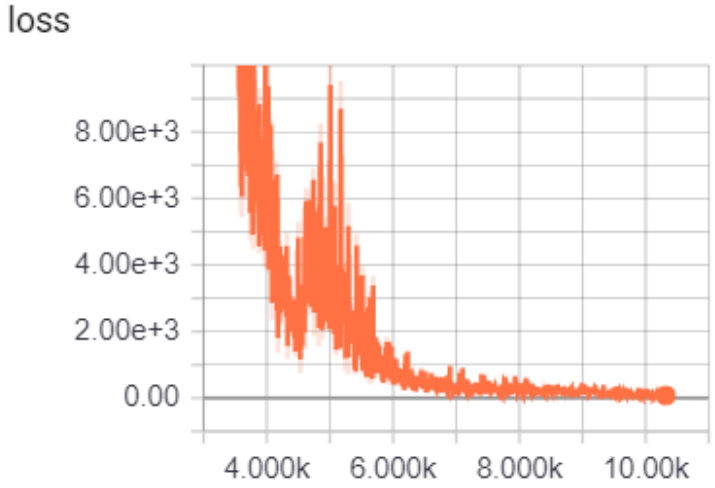


Website: http://localhost:6006/♯ graphs

TensorBoard: Data Visualization

YOLO-Based Flame Detection





Website: http://localhost:6006/# graphs

Questions?

How to Use the MATLAB Toolbox

MATLAB uses four graphical tools for training neural networks to solve problems in function fitting, pattern recognition, clustering, and time series. There are four ways to use the MATLAB Toolbox:

- Through the four graphical user interfaces (GUIs).
- Through basic command-line operations.
- Through customization.
- Through any of the functions contained in the toolbox.

Implementation of Machine Learning

MATLAB Toolboxes

- Function fitting (>> nftool)
- Pattern recognition (>>nprtool)
- Clustering (>> nctool)
- Time series analysis (>>ntstool)

Steps of Using the MATLAB Toolbox

1. Collect data;
2. Create a network;
3. Configure the network;
4. Initialize the weights and biases;
5. Train the network;
6. Validate the network;
7. Use the network.

Implementation of Machine Learning

e.g., Fitting a Function

1. Load dataset, e.g., $[inputs; targets] = house\_dataset;$
2. Create a network, e.g.,
   $hiddenLayerSize = 10$
   $net = \texttt{fitnet}(hiddenLayerSize);$
3. Set up the division of data, e.g.,
   $net.divideParam.trainRatio = 70/100;$
   $net.divideParam.valRatio = 15/100;$
   $net.divideParam.testRatio = 15/100;$
4. Train the network, e.g., $[net; tr] = train(net; inputs; targets);$
5. Test the network, e.g.,
   $outputs = net(inputs);$
   $errors = gsubtract(outputs, targets);$
   $performance = perform(net, targets, outputs)$
6. View the network, e.g., $view(net)$.

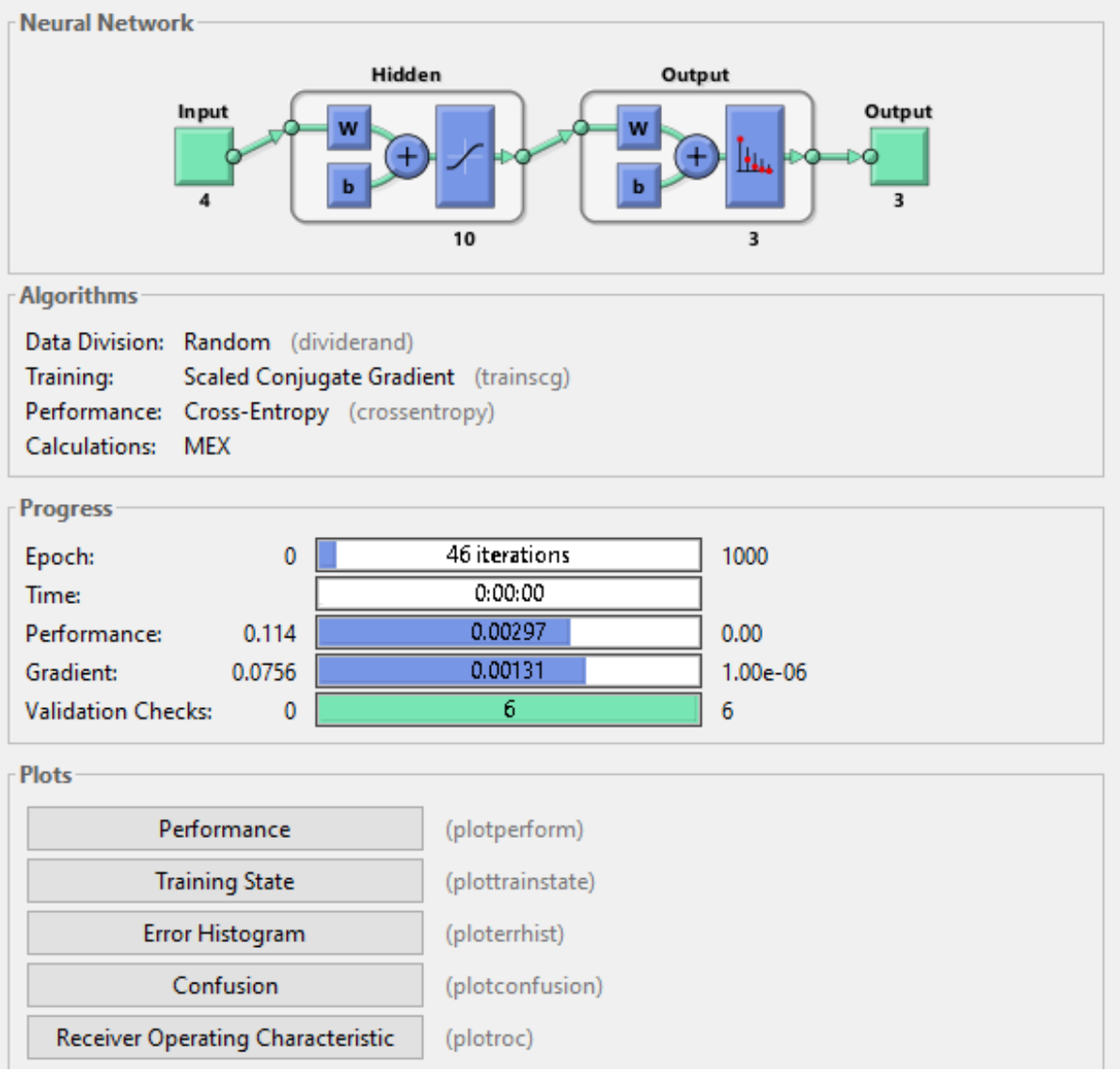
Implementation of Machine Learning

e.g., Pattern Classification

1. Load dataset, e.g.,
   $inputs = cancerInputs;$
   $targets = cancerTargets;$
2. Create a network, e.g.,
   $hiddenLayerSize = 10$
   $net = \texttt{patternnet}(hiddenLayerSize);$
3. Set up the division of data, e.g.,
   $net.divideParam.trainRatio = 70/100;$
   $net.divideParam.valRatio = 15/100;$
   $net.divideParam.testRatio = 15/100;$
4. Train the network, e.g., $[net; tr] = train(net; inputs; targets);$
5. Test the network, e.g.,
   $outputs = net(inputs);$
   $errors = gsubtract(outputs, targets);$
   $performance = perform(net, targets, outputs)$
6. View the network, e.g., $view(net)$.

Implementation of Machine Learning

e.g., Pattern Classification

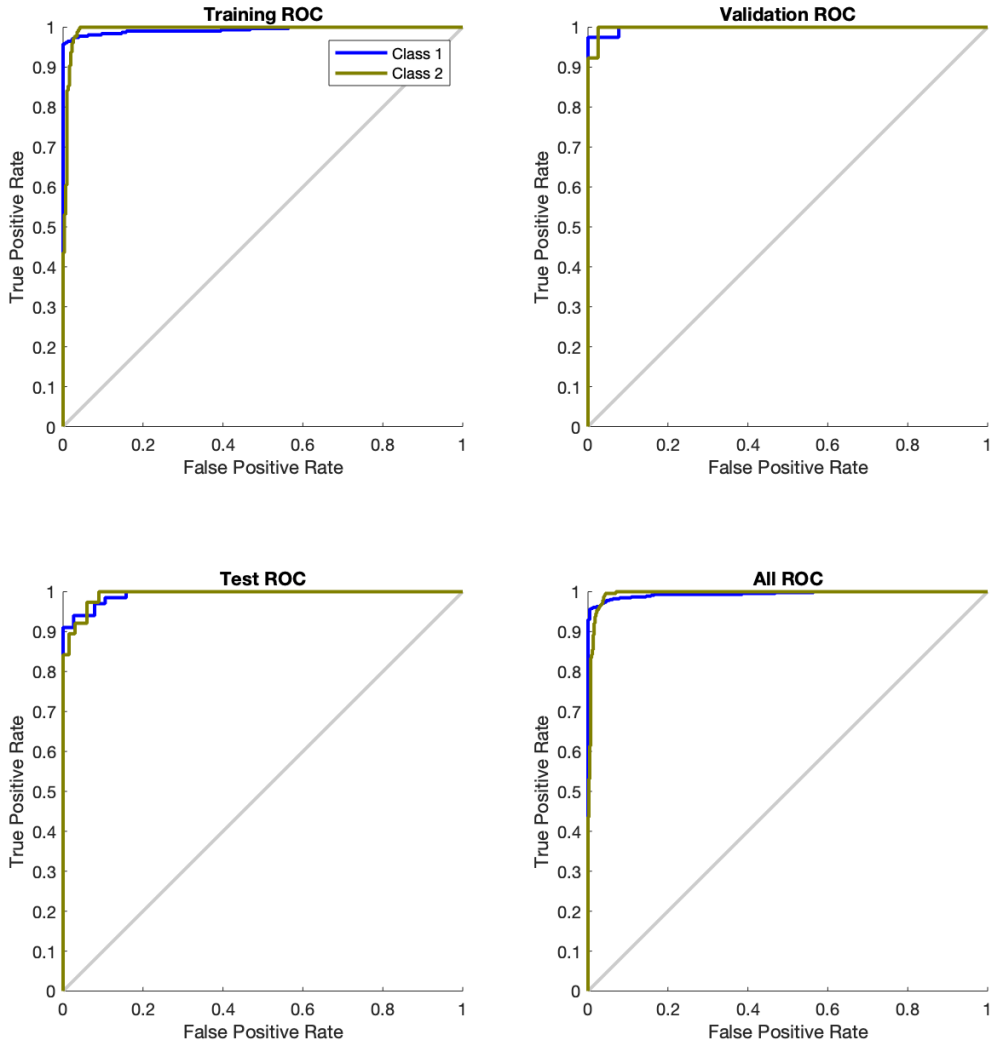


Implementation of Machine Learning

e.g., Pattern Classification

Implementation of Machine Learning

e.g., Pattern Classification



Implementation of Machine Learning

e.g., Clustering

1. Load dataset, e.g.,
   $inputs = simpleclusterInputs$;
2. Create a Self-Organizing Map (SoM), e.g.,
   $dimension1 = 10$;
   $dimension2 = 10$;
   $net =$ `selforgmap`$([dimension1 dimension2])$;
3. Train the network, e.g., $[net, tr] = train(net, inputs)$;
4. Test the network, e.g.,
   $outputs = net(inputs)$;
5. View the network, e.g., $view(net)$.

Implementation of Machine Learning

e.g., Time-Series Prediction and Modeling

1. Load dataset, e.g.,
   $inputSeries = phInputs$; $targetSeries = phTargets$;
2. Create a nonlinear autoregressive network with external input, e.g., $inputDelays = 1:4$; $feedbackDelays = 1:4$;
   $hiddenLayerSize = 10$; $net =$
   `narxnet`$(inputDelays, feedbackDelays, hiddenLayerSize)$;
3. Set up division of data for training, validating, testing, e.g.,
   $net.divideParam.trainRatio = 70/100$;
   $net.divideParam.valRatio = 15/100$;
   $net.divideParam.testRatio = 15/100$;
4. Train the network, e.g.,
   $[net, tr] = train(net, inputs, targets, inputStates, layerStates)$;
5. Test the network, e.g.,
   $outputs = net(inputs, inputStates, layerStates)$;
   $errors = gsubtract(targets, outputs)$;
   $performance = perform(net, targets, outputs)$;
6. ...

Implementation of Machine Learning

e.g., Time-Series Prediction and Modeling

- ...
- View the network, e.g., $view(net)$
- Plot the performance training record to check for potential overfitting, e.g., $figure, plotperform(tr)$;
- Close the loop on the NARX network, e.g., $netc = closeloop(net)$;
- Remove a delay from the network, e.g., $nets = removedelay(net)$;

Questions?

Learning Objectives

- To justify the differences between machine learning and deep learning.
- To critically compare and evaluate the major tools of data labelling and augmentation.
- To critically compare and evaluate the major algorithms of machine learning and pattern recognition.