1. Introduction

M.A.Z. Chowdhury and M.A. Oehlschlaeger

Department of Mechanical, Aerospace and Nuclear Engineering
Rensselaer Polytechnic Institute
Troy, New York

chowdm@rpi.edu
oehlsm@rpi.edu

"Data, data, data, I can't make bricks without clay."
- Sherlock Holmes, fictional data scientist

MANE 4962 and 6962

Course Content & Syllabus

Tentative Lecture Schedule

	Topic	Lectures[a]	
	Introduction to machine learning (ML)	2	
	Overview of ML methods	1	
	ML Project design	1	
	Mathematical background	1	
	Optimization and Gradient Descent	1	
	k-Nearest Neighbors	1	
	Linear and Logistic Regression	2	
	Neural Networks	5	
	Support Vector Machines	2	
	k-means clustering	1	
	Singular Value Decomposition and PCA	2	
	Decision Tree and Random Forest	2	
	Special Topics	1[b]	

[a] Each lecture is approximately 75 min. Actual lectures may not strictly follow this sequence.

[b] Depends on requirement and time remaining.

Course Promises

By the end of this course, you will

- ☞ understand fundamental ML methods
- ☞ identify, formulate, and solve engineering problems using ML
- ☞ build, scale, and manage ML engineering projects
- ☞ interpret ML models
- ☞ handle and wrangle signals, images, and descriptive data for model building
- ☞ be able to read and understand applied research papers on Machine learning
- ☞ have learned to program in Python and manage your programs using version control.

Marks Distribution

1. Homework (6 given, lowest dropped) - MANE 4962: 6%/each; 6962: 4%/each
2. Quizzes (6 given, lowest dropped) - MANE 4962: 4%/each; 6962: 2%/each
3. Exam 1 (Date TBD) - MANE 4962/6962: 15%
4. Project - MANE 4962/6962: Code 10%, report 10%, presentation 15%
5. Exam 2 (Finals Week or Take Home) - only MANE 6962: 20%

References

Not required to purchase, we will supply information.

1. **Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems** - *Aurélien Géron*,
   O'Reilly Media, Inc., 2019
2. **Deep Learning with Python** - *François Chollet*,
   Manning Publications Co., 2018
3. **Deep Learning** - *Ian Goodfellow, Yoshua Bengio, and Aaron Courville* ,
   The MIT Press, 2016

Python

- An object oriented interpreter-based programming language and widely used language for Machine Learning.
  - BDFL : Guido van Rossum (retired)
  - Data types: numbers, strings, lists, and dictionaries (hash-tables)
  - Control flow: While loops, For loops, conditionals and functions
  - Lists and hash-tables are references (like pointers in C)
  - All variables are passed by value
  - Very popular due to availability of useful library packages.
- Scientific computing : NumPy, SciPy, Matplotlib
- Visualization: Matplotlib, Seaborn
- Data wrangling libraries : Pandas, Pillow, scikit-image
- Machine learning and deep learning: scikit-learn, Tensorflow, PyTorch

Python is the glue that can hold all your ideas.

What is machine learning?

"the ability to learn without writing explicit programs”
– Arthur Samuel (1959)

What is machine learning?

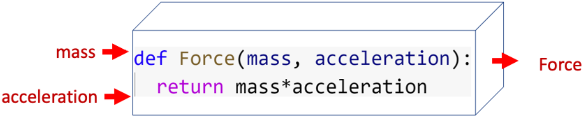
Learning : “knowledge or skill acquired by instruction or study”
– Webster’s dictionary

What is machine learning?

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."

– Tom Mitchell (1997)

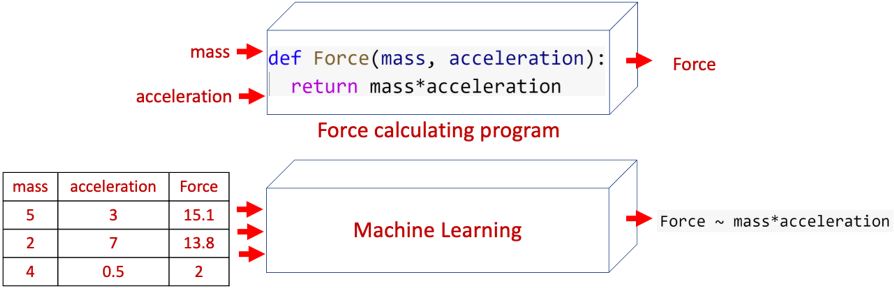
What is Machine Learning?



```python
def Force(mass, acceleration):
  return mass*acceleration
```

Force calculating program

What is Machine Learning?



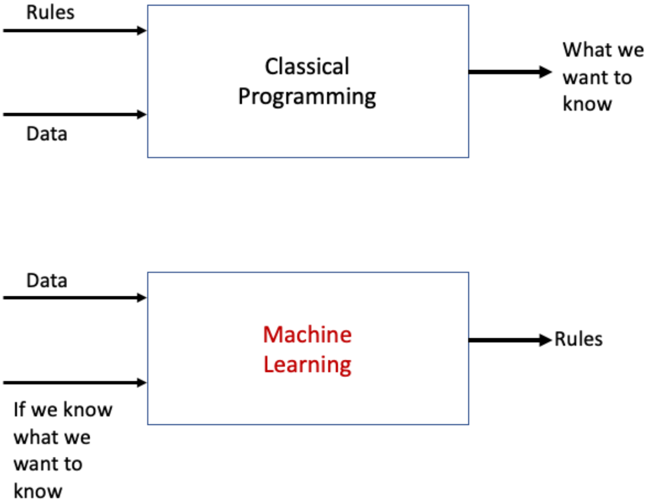
```python
def Force(mass, acceleration):
    return mass*acceleration
```

Force calculating program

mass	acceleration	Force
5	3	15.1
2	7	13.8
4	0.5	2

```
Force ~ mass*acceleration
```

What is Machine Learning?



Classify flowers

Let’s write our first machine learning program.

Classify flowers

But first we need to get some data.

The Iris dataset

Iris setosa

Iris virginica

Iris versicolor

Description

First reported by Sir R.A. Fisher. Possibly, the best known dataset to be found in the pattern recognition literature. The dataset contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.

The Iris dataset

```
from sklearn.datasets import load_iris
data = load_iris()
```

Follow demonstration using iris notebook where we will classify different iris flowers.

(In ML, this is called a model based supervised classification problem.)

What is a computer program?

- A program is a specific set of rules (definitions and commands) to achieve a specific objective. (such as addition of two numbers, finding roots of integers)
- Computers only know what we tell them.
- It may be virtually impossible to tell/write every rule needed to achieve some objectives.
- For example, identifying cats, flowers, cups etc. and predicting price of a house in NYC, number of COVID cases in Rensselaer county, predicting temperautre, ignition delay etc.

Classical programming
vs
machine learning

- A better approach is to write a computer program that is capable of “figuring out the rules” by itself by “looking at observations”.

- We will only specify the rules for “figuring out the rules”.

- This “figuring out the rules” is the “learning” part.

- That is how machine learning allows computers to learn without being explicitly programmed.

Pseudocode for a simple covid model

Predicting COVID cases in Rensselaer County. Population P=150000. New cases = W.

```
Baseline cases (B) = 0.01% of P = 15

if {10%<%mask wearing<50% of P}: W=B*2
elif {%mask wearing<10% of P}: W= B*5
elif {%mask wearing>50% of P}: W= B*0.8
elif {%mask wearing>80% of P}: W= B*0.3

if {Handwashing time of 80% or above of P<20 s}: W= W*10
elif {Handwashing time of 40%-80% of P<20 s}: W= W*6
elif {Handwashing time of 20%-40% of P<20 s}: W= W*2
elif {Handwashing time of 0%-20% of P<20 s}: W= W*1.25

if {Public Awareness == HIGH}: W= W*0.85
elif {Public Awareness == MEDIUM}: W= W*0.90
elif {Public Awareness == LOW}: W= W*1.10
```

and so on ... (way too many rules to specify)

Integer Square Root Finding Program

```
num = 8
num_sqrt = num**0.5
print("The square root of %0.3f is %0.3f", (num, num_sqrt))
```

Can you guess where we defined the rule?

Integer Square Root Finding Program

```
num = 8  #input integer
num_sqrt = num**0.5 #(!!! WE DEFINED THE RULE HERE !!!)
print("The square root of %0.3f is %0.3f", (num, num_sqrt))  # Program output
```

Nothing to learn here :(

Learning square roots from data ...

How about we give some data to a ML program to let it figure out the rules for finding a square root?

```
1,1
4,2
9,3
16,4
25,5
36,6
49,7,
64,8
81,9
100,10
```

A machine learning program for finding square root of an integer using a deep neural network.

```
import tensorflow as tf
from numpy import loadtxt
from keras.models import Sequential
from keras.layers import Dense
from keras import optimizers
from tensorflow.python.keras.optimizers import *
import numpy as np

#Load dataset
#define input (X) and output (y)
X = np.array([[1], [4], [9], [16], [25], [36], [49], [64], [81], [100]])
X = X*1.0
y = np.array([[1], [2], [3], [4], [5], [6], [7], [8], [9], [10]])
y = y*1.0
```

contd.

A machine learning program for finding square root of an integer using a deep neural network.

```
#define keras model (neural network)
model = Sequential()

model.add(Dense(6,input_dim=1,activation='relu'))
model.add(Dense(6,activation='relu'))
model.add(Dense(6,activation='relu'))
model.add(Dense(1))

#compile the keras model
opt = optimizers.Adam(learning_rate=0.001)
mse = tf.keras.losses.MeanSquaredError(reduction=tf.keras.losses.Reduction.SUM)
model.compile(loss=mse, optimizer=opt)
```

contd.

A machine learning program for finding square root of an integer using a deep neural network.

```
#fit the keras model on the dataset (CPU)
model.fit(X,y,epochs=2000,batch_size=10, verbose=0) # model is trying to learn here
model.summary()

#make class predictions with the model
predictions = model.predict(X)

#summarize the first 10 cases
for i in range(10):
    print('%s => %.2f (expected %.2f)' %(X[i].tolist(), predictions[i], y[i]))
```

☞ Nowhere in this program we defined the rule for calculating square root. Let's see how it does?

Let’s check the output. Your mileage may vary.

```
[1.0] => 2.18 (expected 1.00)
[4.0] => 2.47 (expected 2.00)
[9.0] => 2.96 (expected 3.00)
[16.0] => 3.56 (expected 4.00)
[25.0] => 4.34 (expected 5.00)
[36.0] => 5.29 (expected 6.00)
[49.0] => 6.41 (expected 7.00)
[64.0] => 7.70 (expected 8.00)
[81.0] => 9.17 (expected 9.00)
[100.0] => 10.81 (expected 10.00)
```

Let’s check the output. Your mileage may vary.

```
[1.0] => 2.18 (expected 1.00)
[4.0] => 2.47 (expected 2.00)
[9.0] => 2.96 (expected 3.00)
[16.0] => 3.56 (expected 4.00)
[25.0] => 4.34 (expected 5.00)
[36.0] => 5.29 (expected 6.00)
[49.0] => 6.41 (expected 7.00)
[64.0] => 7.70 (expected 8.00)
[81.0] => 9.17 (expected 9.00)
[100.0] => 10.81 (expected 10.00)
```

- Good, but not great!
- Why?
- Not enough data.
- Neural nets needs lot of data to learn something useful.
- The results may vary a little from different runs. Why?

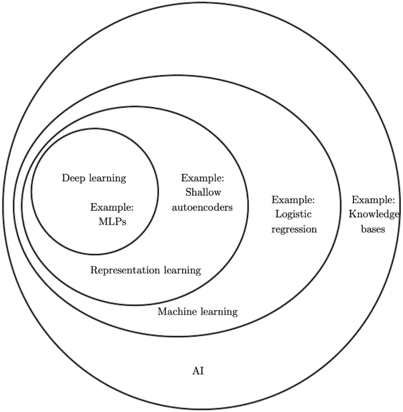
Let's take a breather ...

- ☞ A vast amount of time is spent in collecting and preparing "good" data.
- ☞ The goal is to have the "best" ML model to solve the problem you defined.

Definitions

Some common terms we hear all the time are:

- Computer Program: *Set of specific rules to achieve a set of objectives.*
- Artificial Intelligence: *Any method/algorithm that tries to replicate or mimic the results of some aspect of human cognition/expertise.*
- Machine Learning: *Programs that perform better with experience or learn from data to figure out rules.*
- Artificial Neural Networks: *A type of Machine Learning algorithm.*
- Deep Learning: *A type of ANN. Typically with more hidden layers.*
- Big Data: *Using data to discover useful unobvious patterns*



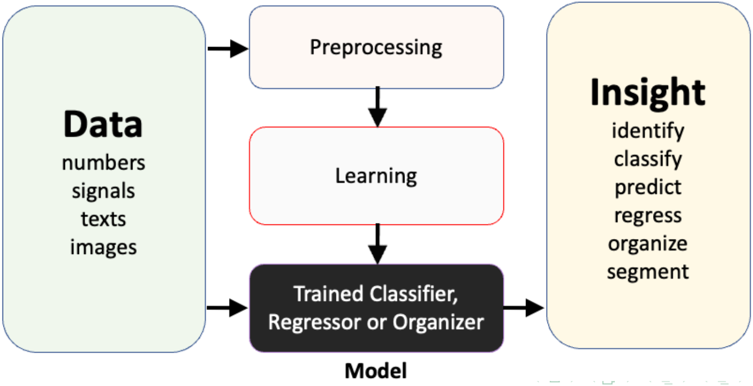
Why machine learning?

Tackle "unprogrammable" problems

Automate and optimize experiments/processes/products

Affordable computing power

Design interpretable predictive models



Types of Machine Learning Methods

Supervised

Data: (x, y)
x is data y is label/target

Goal: Learn function to map x to y

Unsupervised

Data: x
Just data no label

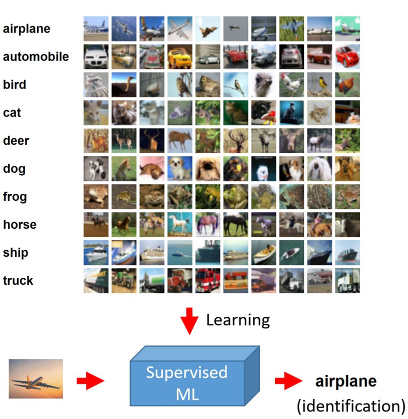
Goal: Learn underlying pattern

Reinforcement

Data: (s_t, a_t)
Agent interacting with environment

Goal: Maximize reward

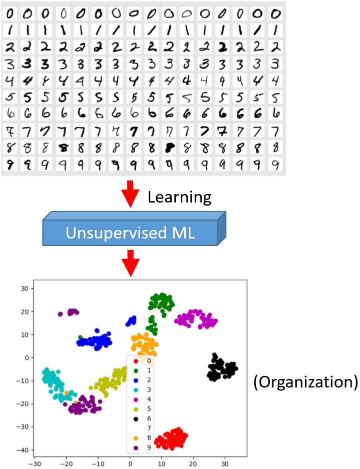
There are more types of machine learning, but lets focus on these for the moment.



Learning

Supervised ML

airplane
(identification)



Learning

Unsupervised ML

(Organization)

Dataset: CIFAR-10 and MNIST Digits

Features

- ☞ are n-dimensional vector. (Actually, tensor)
- ☞ quantify what we know about the problem we want to solve.
- ☞ measurable property or some characteristic of a problem or a phenomena.
- ☞ also called descriptor/predictor/attribute in different fields and applications.
- ☞ independent variable. Values are determined by data.
- ☞ are a representation of the data. You can represent the same data in many ways with different features.
- ☞ can be calculated/extracted from original features via process called feature extraction. Feature extraction derives new improved features.

1. In the Iris example, how many features were there?
2. In the square root example, how many features were there?

Targets

- ☞ are dependent variable.
- ☞ are the things we want to know.
- ☞ are discrete variables if we want to identify/categorize/classify something. (classification task)
- ☞ are continuous variables if we want to predict/regress/estimate something. (regression task)

1. In the Iris example, what was the target and how many targets were there?
2. In the square root example, what was the target and how many targets were there?