Day 1: Introduction to Machine Learning

Summer STEM: Machine Learning

Department of Electrical and Computer Engineering
NYU Tandon School of Engineering
Brooklyn, New York

June 27, 2020

Outline

Naman

Huaijiang

Armand

Virinchi

Tell the class about yourself

- Write down the following information:
  - Name
  - Grade
  - Where are you from?
  - What do you want to get out of this class?
  - What is your favourite movie?
  - What is the IMDB score of this movie!
  - What is the category of this movie? (thriller/drama/action, etc)
  - Rate your coding experience from 1 (no experience) to 5 (plenty of experience)!
- Share your answers with the class!
- We'll visualize this dataset using Python later today
  - Link to excel sheet here

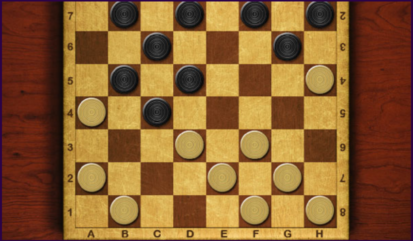
Outline

Machine Learning

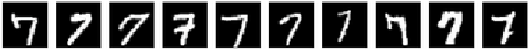
- Most recent exciting technology
- We use these algorithms dozens of times a day
  - Search Engine
  - Recommendations
- Machine Learning is an important component to achieve the big AI dream
- Practice is the key to learn machine learning

Definition

- Machine Learning is a field of study that gives computers the ability to learn without being explicitly programmed.



Example: Digit Recognition



- Challenges with expert approach
  - Simple expert rule breaks down in practice
  - Difficult to translate our knowledge into code
- Machine Learning approach
  - Learned systems do very well on image recognition problems

```python
def classify(image):
    ...
    nv = count_vert_lines(image)
    nh = count_horiz_lines(image)
    ...

    if (nv == 1) and (nh == 1):
        digit = 7
    ...

    return digit
```

Machine Learning Problem Pipeline

1. Formulate the problem: regression, classification, or others?
2. Gather and visualize the data
3. Design the model and the loss function
4. Train your model
   - (a) Perform feature engineering
   - (b) Construct the design matrix
   - (c) Choose regularization techniques
   - (d) Tune hyper-parameters using a validation set
   - (e) If the performance is not satisfactory, go back to step (a)
5. Evaluate the model on a test set

Outline

Course Outline

- Day 1: Introduction to ML
- Day 2: Linear Regression
- Day 3: Overfitting and Generalization
- Day 4: Classification and Logistic Regression
- Day 5: Mini Project
- Day 6: Neural Networks
- Day 7: Convolutional Neural Networks
- Day 8: Deep Generative Models
- Day 9: Final Project
- Day 10: Social Impacts of ML and Final Project Presentations

Course Format, Website, Resources

- Course Website: `github.com/huaijiangzhu/SummerML`
  - Github: share collections of documents, repositories of code
  - Contains lecture slides, code notebooks, and datasets
  - Slides and demo code posted before lecture, solutions to the lab posted after
- After-class discussion: Piazza
- We strongly encourage programming in Python via Google Colab
  - No installation required
- We'll give additional resources at the end of each day based on student interest

Outline

Setting Up Python

- Google Colab
  - Interactive programming online
  - No installation
  - Free GPU for 12 hours
- Your task:
  Register a Google account and set up Google Colab
  Run `print('hello world!')`
  Save the notebook on your computer

Modules/Libraries/Packages

- NumPy: math, vectors and matrices
- MatPlotLib: plotting graphs, visualizing data
- Pandas: convenient for storing and retrieving data
- Sklearn: convenient wrapper for simple ML problems
- PyTorch: deep learning
- Your task:
  Run `import numpy as np`
  Play with `np.add()`
  Open `demo_python_basics.ipynb` from Github

Outline

Python Basics

- Program
  - We write operations to be executed on variables
- Variables
  - Referencing and interacting with items in the program
- If-Statements
  - Conditionally execute lines of code
- Functions
  - Reuse lines of code at any time

Outline

Vectors

- A **vector** is an ordered list of numbers or symbols
  - Represents quantities with magnitude and direction.
  - Ex: Planar force: $\mathbf{v} = (1, 2)$

Vectors

- Vectors of the same size may be added together, element-wise
  - Ex: $\mathbf{u} = (3, -1)$
    $\mathbf{v} + \mathbf{u} = (1 + 3, 2 + (-1)) = (4, 1)$
- Vectors may be scaled by a number, element-wise
  - Ex: $3\mathbf{v} = (3 \times 1, 3 \times 2) = (3, 6)$

Vectors

- Norm of a vector (L2 Norm)
  - Ex: $\|\mathbf{v}\| = \sqrt{1^2 + 2^2} = \sqrt{5}$
- Inner product: sum of element-wise products of two vectors
  - Ex: $\mathbf{u} \cdot \mathbf{v} = 1 \times 3 + 2 \times (-1) = 3 - 2 = 1$
  - Gives the angle between two vectors $\cos\theta = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}$

NYU TANDON SCHOOL OF ENGINEERING

Exercise: Vectors

Let $\mathbf{p} = (3, 2, 9, 4)$ and $\mathbf{q} = (1, 9, 0, 3)$, calculate

- $3\mathbf{q} + 2\mathbf{p}$
- $\mathbf{q} \cdot \mathbf{q}$ and $\|\mathbf{q}\|^2$
- $\mathbf{p} \cdot \mathbf{q}$ and $\|\mathbf{p}\| \, \|\mathbf{q}\|$
- $\left\| \frac{\mathbf{p}}{\|\mathbf{p}\|} \right\|$

Matrices

- A **matrix** is a rectangular array of numbers or symbols arranged in rows and columns. We can conceptualize it as a collection of vectors.
  - Ex: 2 by 2 matrix, $M = \begin{bmatrix} 1 & 3 \\ 2 & -1 \end{bmatrix}$
- Matrices of the same shape may be added together, element-wise
  - Ex: $A = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 & 8 \\ 7 & 11 \end{bmatrix}, A + B = \begin{bmatrix} 1 & 9 \\ 9 & 12 \end{bmatrix}$
- Matrices may be scaled, element-wise
  - Ex: $aB = \begin{bmatrix} 0 & 8a \\ 7a & 11a \end{bmatrix}$, where a is a scalar

Exercise: Matrices

- $\begin{bmatrix} 1 & 3 \\ 2 & -1 \\ 4 & 7 \end{bmatrix} + \begin{bmatrix} 9 & 2 \\ -7 & 6 \\ 3 & 1 \end{bmatrix} = ?$
- $2 \begin{bmatrix} 1 & 9 \\ 3 & -2 \end{bmatrix} = ?$
- $2\left(\begin{bmatrix} 5 & 1 \\ 1 & -3 \end{bmatrix} + \begin{bmatrix} 3 & 4 \\ 2 & -8 \end{bmatrix} \right) + 3 \begin{bmatrix} -2 & 1 \\ 4 & -7 \end{bmatrix} = ?$

NYU TANDON SCHOOL OF ENGINEERING

Vectors and Matrices

- We may consider a vector as a matrix
  - **Row Vector**: shape (1 x N)
    Ex: $\mathbf{v} = \begin{bmatrix} 1 & 2 \end{bmatrix}$
  - **Column Vector**: shape (N x 1)
    Ex: $\mathbf{v} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$
  - We'll consider vectors as column vectors by default

NYU TANDON SCHOOL OF ENGINEERING

Outline

Python Basics

- Lists
  - Store an ordered collection of data
- Loops
  - Conditionally re-execute code
- Strings
  - Words and sentences are treated as lists of characters
- Classes (advanced)
  - Making your own data-type. Functions and variables made to be associated with it too.

NYU TANDON SCHOOL OF ENGINEERING

Lab: Python Basics

- Write a function to find the second largest number in a list (Hint: use sort())
- Define a class which has at least two methods:
  - getString: to get a string from console input (Hint: use input())
  - printString: to print the string in upper case (Hint: use upper())
- Write a test function to test the class methods.

Outline

More About Matrices

- Matrices may be multiplied together provided their shapes meet the criteria
  - $(AB)_{ij} = \sum_{k=1}^{N} A_{ik} B_{kj}$
  - Inner product of the i-th row of A and the j-th column of B
  - Criteria: for AB to be valid, # cols of A must equal the # rows of B
  - Result is a matrix with shape (# rows A, # cols B)
  - $\begin{bmatrix} 1 & 3 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = ?$, $\begin{bmatrix} 1 & 3 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} 2 & 4 \\ 3 & 2 \end{bmatrix} = ?$
  - $\begin{bmatrix} 2 & 4 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} 1 & 3 \\ 2 & -1 \end{bmatrix} = ?$
- In general, $AB \neq BA$

More About Matrices

- **Transpose**: A^T swaps the rows and columns of matrix A
- Ex: $\begin{bmatrix} 2 \\ 3 \end{bmatrix}^T = \begin{bmatrix} 2 & 3 \end{bmatrix}$ and $\begin{bmatrix} 2 & 4 \\ 3 & 2 \end{bmatrix}^T = \begin{bmatrix} 2 & 3 \\ 4 & 2 \end{bmatrix}$
- $\begin{bmatrix} 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & -1 \end{bmatrix} = ?$
- $(AB)^T = B^T A^T$
- **Inverse**: A^{-1} satisfies the equation $AA^{-1} = A^{-1}A = \mathbb{I}$
  - Square matrices only!

Exercise: Playing with NumPy

- NumPy: Python package for linear algebra
- Open `demo_vectors_matrices.ipynb`

Outline

Demo: Plotting Functions

- Generate and plot the following functions in Python:
  - Scatter plot of points: (0,1), (2,3), (5,2), (4,1)
  - Straight Line: $y = mx + b$
  - Sine-wave $y = sin(x)$
  - Polynomial e.g. $y = x^3 + 2$
  - Exponential e.g. $y = e^-2x$
  - Gaussian (Use $\sigma = 0.5$)
  - Choose a function of your own
- Create separate plots for each of the functions, Compute the mean and variance of each function
- Use Wikipedia and Numpy Documentation to search for mathematical formulas and python functions

Outline

Looking at our ice-breaker data in spreadsheets

- Columns have labels in the first row
- Collected data (numbers, words) follow below
- Let's export it to a Comma-Separated Values (CSV) file and open it

---

These slides have been modified from the original slides provided through the courtesy of Nikola, Akshaj, Aishwarya, and Jack.