INTRODUCTION TO MACHINE LEARNING USING HPC

Research Technologies Department
University of Arizona

https://ua-researchcomputing-hpc.github.io/Intro-to-Machine-Learning/
Introduction to Artificial Intelligence (AI) and Machine Learning (ML)

Zhenhua He | Ridham Patoliya
Learning objectives

Terminology of Machine Learning

The different types of machine learning techniques

Applications of machine learning techniques
Python Data Exploration and Analysis

- **Pandas**: Data manipulation/analysis library
- **Matplotlib**: Data visualization library
- **Scikit-Learn**: Machine learning library
ENTERPRISE DATA vs MACHINE LEARNING DATA
What is Artificial Intelligence

• **Wikipedia**: intelligence demonstrated by machines as opposed to natural intelligence displayed by animals including humans.

• **Oxford**: the theory and development of computer systems able to perform tasks that normally require human intelligence.

• **IBM**: leverages computers and machines to mimic the problem-solving and decision-making capabilities of the human mind.
ML Terminology is used inconsistently

- Artificial Intelligence
- Machine Learning
- "Traditional" ML
- Neural Networks (Deep Learning)
Formats of data

NUMBERS

TEXTS

IMAGES
Image (as seen by computers)
Types of data:

- Labelled
- Unlabelled
Labelled data

Dog

Cat

Dog

Dog

Cat

Cat

Cat

Dog

Cat
Unlabelled Data
General ML Process

Gather Data

Define Question

Choose ML Model/Method

Divide Data into Training/Testing Sets

Train or Fit Model

Test Model

Use Model for Inference
What is training?

The process used to create our ML model. Find a set of weights and biases that give accurate predictions, by minimizing a cost function.
What is testing?

Measuring the predictions of our model against known outcomes, using a subset of our data.
What is inference?

Running our model on live data to produce actionable output.
Common types of Learning

Supervised learning
We have labelled data, and we want to make some prediction
- Regression
- Classification

Unsupervised learning
We have unlabeled data, and we want to make some prediction
- Clustering
Supervised learning
Regression
Regression

Y-axis: Housing Price
X-axis - area in sq. feet

- $0.1 million
- 1000 square feet

Texas A&M University HPRC Summer Computing Academy 2021
Regression

X-axis – area in sq. feet

Y-axis Housing Price

$0.3 million

$0.1 million

1000 square feet

3500 square feet

1000 square feet

3500 square feet

Texas A&M University HPRC Summer Computing Academy 2021
Quiz

• Which of the following CANNOT be an example of regression?
  • A) Using past data of weather in college station to predict future’s weather.
  • B) Predicting prices of stocks using previous month’s price data
  • C) Determining if an email is spam or not
  • D) Determining network traffic for today using previous month’s data
Classification
Which of the following CANNOT be an example of classification?

- A) Using blood pressure and weight data to determine if a patient is diabetic or not
- B) Estimating amount of annual rain from previous year’s data
- C) Classifying Pokémon in different types (e.g., fire, ice, poison, electric)
- D) Determining if an email is spam or not
Unsupervised learning
clustering
Quiz

• Which of the following CANNOT be an example of clustering?
  • A) Sorting and making groups of research papers having similar content
  • B) Determining whether a news article is about politics or sports
  • C) Identifying clusters of stars having similar characteristics
  • D) Sorting through subjects of emails and grouping them accordingly
Quiz

Which of the following CANNOT be an example of machine learning? Select all that apply.

- A) Manually trying out different passwords on your amazon account to check if it works
- B) Your virtual assistant starts recognizing your voice after first few tries
- C) Fire alarm goes off when smoke level is more than a specific level
- D) Sorting through subjects of emails and grouping them accordingly
Neural Networks and Biology

Key components of a neuron:
- **Cell Body**: Contains the nucleus and is the site of cell processes and structure.
- **Nucleus**: Controls the cell’s activities.
- **Dendrites**: Receive signals from other neurons.
- **Axon**: Transmits signals to other neurons.
- **Synapses**: Sites of connection between neurons, where neurotransmitters are released to communicate.

**Figure**: Structure of a typical neuron
Jupyter Notebooks on HPC

ood.hpc.arizona.edu
ocelote / 2 hours / 1 core / 6 mem / standard queue / chrisreidy

Accessing files for the exercises
ssh netid@hpc.arizona.edu
shell
ocelote
https://ua-researchcomputing-hpc.github.io/Intro-to-Machine-Learning

Then Accessing Workshop Files and cut / paste the section starting “wget”
(old method: cp /xdisk/chrisreidy/workshops/* .)
Choice #1: Cut and paste commands into Jupyter from .txt file
Choice #2: Run the Notebook .ipynb file
Choice #3: Type in the commands. Syntax is very important
Train a linear regression model
Jupyter Notebooks on UArizona HPC with Python.

ood.hpc.arizona.edu
Jupyter Notebooks on UArizona HPC with Python
Jupyter Notebooks on UArizona HPC with Python

<table>
<thead>
<tr>
<th>Jupyter Notebook</th>
<th>Interactive Apps</th>
<th>Desktops</th>
<th>Interactive Desktop</th>
<th>GUIs</th>
<th>ABAQUS GUI</th>
<th>ANSYS Workbench GUI</th>
<th>MATLAB GUI</th>
<th>Mathematica GUI</th>
<th>Servers</th>
<th>Jupyter Notebook</th>
<th>RStudio Server</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Jupyter Notebook</th>
<th>This app will launch a Jupyter server using Python on a UAz cluster.</th>
</tr>
</thead>
</table>

**Cluster**

Ocelote Cluster

**Run Time**

1

Enter maximum number of wall clock hours the job is allowed to run.

**Core count on a single node**

1

Enter the number of cores on a single node that the job is allowed to use.

**Memory per core**

6

Enter the number of Gigabytes of RAM needed per core.

**Special Options**

Enter node specific requirements, if any.

**PI Group**

chrisreidy

Enter an HPC PI group to be charged for time used.
Jupyter Notebooks on UArizona HPC with Python
Jupyter Notebooks on UArizona HPC with Python
Jupyter Notebooks on UArizona HPC with Python
Train a linear regression model

- Import libraries
  ```python
  # Import libraries
  import pandas as pd
  import numpy as np
  import matplotlib.pyplot as plt

  from sklearn.linear_model import LinearRegression
  from sklearn.model_selection import train_test_split
  ```

- Use Pandas to load the data and view the first 5 rows
  ```python
  # load data and view the first 5 rows
  data = pd.read_excel("king_county_house_data.xlsx")
  
data.head(5)
  ```
<table>
<thead>
<tr>
<th>id</th>
<th>date</th>
<th>price</th>
<th>bedrooms</th>
<th>bathrooms</th>
<th>sqft_living</th>
<th>sqft_lot</th>
<th>floors</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>20141013</td>
<td>221900</td>
<td>3</td>
<td>1.00</td>
<td>1180</td>
<td>5650</td>
<td>1.0</td>
</tr>
<tr>
<td>1</td>
<td>20141209</td>
<td>538000</td>
<td>3</td>
<td>2.25</td>
<td>2570</td>
<td>7242</td>
<td>2.0</td>
</tr>
<tr>
<td>2</td>
<td>20150225</td>
<td>180000</td>
<td>2</td>
<td>1.00</td>
<td>770</td>
<td>10000</td>
<td>1.0</td>
</tr>
<tr>
<td>3</td>
<td>20141209</td>
<td>604000</td>
<td>4</td>
<td>3.00</td>
<td>1960</td>
<td>5000</td>
<td>1.0</td>
</tr>
<tr>
<td>4</td>
<td>20150218</td>
<td>510000</td>
<td>3</td>
<td>2.00</td>
<td>1680</td>
<td>8080</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Train a linear regression model

- Choose the columns from the data
- Split the data into train and test sets
- Visualize the train set

```python
space = data['sqft_living']
price = data['price']

# Change X into 2D array
X = np.array(space).reshape(-1, 1)
Y = np.array(price)

# Split data into train sets and test sets
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=1/3, random_state=0)

# Visualize training set
plt.scatter(X_train, Y_train, color="red", label="Living Area")
plt.title("Housing Prices in King County, WA")
plt.xlabel("Area (sq-ft)")
plt.ylabel("Price (USD)")
plt.legend()
plt.show()
```
Train a linear regression model

- Train the model with train set
- Predict on test set
- Visualize the train data and the best fit line

```python
# Train
regressor = LinearRegression()
regressor.fit(X_train, Y_train)

# Prediction
y_pred = regressor.predict(X_test)
```

```python
# Visualize the data and the best fit line
plt.scatter(X_train, Y_train, color="red", label="Living Area")
plt.title("Housing Prices in King County, WA")
plt.plot(X_train, regressor.predict(X_train), color="blue", label="Price")
plt.xlabel("Area (sq-ft)")
plt.ylabel("Price (USD)"")
plt.legend()
plt.show()
```
Train a linear regression model

- Predict the price of a house with a certain area
- Visualize the test data

```python
area = 5000
price = regressor.predict([[area]])
print('House of %d sq-ft costs about $%d' % (area, price))
```

```
# Visualize test set
plt.scatter(X_test,Y_test,color='red',label="Living Area")
plt.plot(X_test,regressor.predict(X_test),color="blue",label="Price")
plt.xlabel("Area (sq-ft)"
plt.ylabel("Price (USD)"
plt.legend()
plt.show()
```
Build a clustering model for Iris Dataset
Build a clustering model – Iris dataset

- Import libraries

```python
# import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
#matplotlib inline
from sklearn.cluster import KMeans
from sklearn.datasets import load_iris
```
Build a clustering model – Iris dataset

- Load the data
Build a clustering model – Iris dataset

- Visualize the data
Build a clustering model – Iris dataset

- Estimate k with elbow method - first try k = 5

```python
# Let's first try k = 5
x = iris.data
kmeans5 = KMeans(n_clusters=5, init = 'k-means++', random_state = 0)
y = kmeans5.fit_predict(x)
print(y)

array([[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
        4 4 4 4 4 4 4 4 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
        1 1 4 1 1 1 2 2 4 1 4 2 4 1 2 4 1 2 2 1 4 4 2 1 1 4 1 1 4 1 1 4 1
        1 4]]

kmeans5.cluster_centers_

array([[5.006 , 3.428 , 1.462 , 0.246  ],
       [6.52916667, 3.05833333, 5.50833333, 2.1625 ],
       [7.475 , 3.125 , 6.3  , 2.05   ],
       [5.508 , 2.6  , 3.908 , 1.204  ],
       [6.20769231, 2.85384615, 4.74615385, 1.56410256]])
```
Build a clustering model – Iris dataset

- Estimate k with elbow method

```python
[7] plt.scatter(x[y == 0,0], x[y==0,1], s = 15, c='red', label = 'Cluster_1')
plt.scatter(x[y == 1,0], x[y==1,1], s = 15, c='blue', label = 'Cluster_2')
plt.scatter(x[y == 2,0], x[y==2,1], s = 15, c='green', label = 'Cluster_3')
plt.scatter(x[y == 3,0], x[y==3,1], s = 15, c='cyan', label = 'Cluster_4')
plt.scatter(x[y == 4,0], x[y==4,1], s = 15, c='magenta', label = 'Cluster_5')
plt.scatter(kmeans5.cluster_centers_[;0], kmeans5.cluster_centers_[;1], s = 25, c='yellow', label = 'Centroids')
plt.legend()
plt.show()
```
Build a clustering model – Iris dataset

- Estimate $k$ with elbow method

```python
[8] Error =
    for i in range(1, 11):
        kmeans1 = KMeans(n_clusters = i, init = 'k-means++', max_iter = 300, n_init = 10, random_state = 0).fit(x)
        kmeans1.fit(x)
        Error.append(kmeans1.inertia_)
import matplotlib.pyplot as plt
plt.plot(range(1, 11), Error)
plt.title('Elbow-Method with k=1-11') #within cluster sum of squares
plt.xlabel('Number of clusters')
plt.ylabel('Error')
plt.show()
```

$K = 3$
Get the optimal $k = 3$ from the elbow method. Cluster centers

```python
[9] kmeans3 = KMeans(n_clusters=3, random_state=21)
y = kmeans3.fit_predict(x)
kmeans3.cluster_centers_

array([[5.9016129 , 2.7483871, 4.39354839, 1.43387097],
       [5.006 , 3.428 , 1.462 , 0.246 ],
       [6.85 , 3.07368421, 5.74210526, 2.07105263]])
```

```python
[11] plt.scatter(x[y == 0,], x[y==0,1], s = 15, c= 'red', label = 'Cluster_1')
plt.scatter(x[y == 1,], x[y==1,1], s = 15, c= 'blue', label = 'Cluster_2')
plt.scatter(x[y == 2,], x[y==2,1], s = 15, c= 'green', label = 'Cluster_3')
plt.scatter(kmeans3.cluster_centers_[:,0], kmeans3.cluster_centers_[:,1], s = 25, c = 'yellow', label = 'Centroids')
plt.legend()
plt.show()
```
Build a clustering model – Iris dataset

- Compared the actual and predicted clusters
Getting help

• HPC documentation docs.hpc.arizona.edu

• Support ticket
  https://uarizona.service-now.com/sp?id=sc_cat_item&sys_id=2983102adbd23c109627d90d689619c6&sysparm_category=84d3d1acdbc8f4109627d90d6896191f

• Office Hours – Wednesday 2-4 PM
  https://gather.town/app/dVsAprPNBVmI9NpL/hpc-office-hours

• HPC consulting
  hpc-consult@list.arizona.edu

• Visualization consulting
  vislab-consult@list.arizona.edu

• Statistics consulting
  stat-consult@list.arizona.edu