Tutorial for Assignment 1

COMP3314
Machine Learning

Xi CHEN

Tutorial for Assignment 1

- Basics of Python
- Requirements of Assignment 1
- Guidelines for Assignment 1

Basics of Python

Basics of Python

- Python is an interpreted, high-level and general-purpose programming language [1]. The usage of Python is like that of other scripting languages, such as R and MATLAB.
- Currently, Python is one of the most popular programming languages [2].

[1] https://en.wikipedia.org/wiki/Python_(programming_language)
[2] https://spectrum.ieee.org/top-programming-languages-2022/

Basics of Python

- Python provides many useful packages (also know as libraries or wheels) to process data, build machine learning models, do scientific computing, etc.
- Python is the first choice in various deep learning frameworks, such as PyTorch and TensorFlow.

  - NumPy
  - SciPy library
  - scikit-learn
  - Matplotlib
  - Pandas
  - Tensorflow
  - Jupyter Notebook

Basics of Python

- AlexNet, the first deep CNN published in 2012, is written in tens of thousands lines of C++ .
- Nowadays, we could use 5-10 lines of code to build and train a CNN model with the help of some highly integrated python packages like fast.ai.

```
path = untar_data(URLs.PETS)/'images'

def is_cat(x): return x[0].isupper()
dls = ImageDataLoaders.from_name_func(
    path, get_image_files(path), valid_pct=0.2, seed=42,
    label_func=is_cat, item_tfms=Resize(224))

learn = vision_learner(dls, resnet34, metrics=error_rate)
learn.fine_tune(1)
```

Basics of Python

- Packages that might be used in Assignment 1
  - scikit-learn
    - Use some implemented classifiers to train and predict on your data.
    - Tutorial: Documentation
  - pandas
    - Load and process your dataset .
    - Tutorial: 10 minutes to pandas
  - Matplotlib
    - Draw figures for visualize your results.
    - Tutorial: Quick start for Matplotlib
  - NumPy
    - Manipulate and do computations for matrix.
    - Tutorial: Quick start for NumPy

Basics of Python

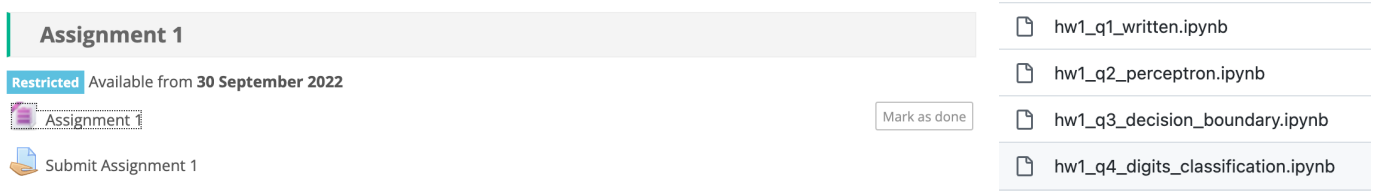
- How to edit and run python programs?
  - Recommendation:
    - Google Colab https://colab.research.google.com
    - Advantage: easy to use, most of the packages are already installed.

  - For those who are familiar with Python:
    - Use Anaconda/Miniconda to manage your environments.
    - Use Jupyter notebook/ PyCharm/ VSCode to edit and run your code.
    - Advantage: manage the project on your own computer.

Requirements of Assignment 1

Requirements of Assignment 1

- Where to download the assignment?
  - Course Moodle Assignment 1 (4 files with questions and some template code).



- What to submit?
  - Completed python notebook with executed outputs.
  - Name the file as using your uid, xxxx.zip. For example: 3009666.zip.
- Where to submit?
  - Course Moodle Assignment 1.
- When to submit?
  - Before 16 Oct (11:59 PM).
- Do not copy! Both the student who copies and the student who offers his/her work for copying shall be penalized.

Requirements of Assignment 1

- Quiz (20 %) - 2 equally weighted written quizzes
  - Q1: 18 Oct, 12:30 pm - 1:20 pm, written
  - Q2: 29 Nov, 12:30 pm - 1:20 pm, written
- Assignments (30 %) - 3 equally weighted programming assignments
  - A1: 30 Sep - 16 Oct (11:59 PM)
  - A2: 28 Oct - 13 Nov
  - A3: 25 Nov - 11 Dec
  - Late submission policy:
    - 20% deduction within 24 hours, 50% deduction within 48 hours
    - no accept beyond 48 hours, unless extreme emergency
- Final examination (50 %)
  - Written, 120 minutes exam
  - Candidates are permitted to refer to the following electronic/printed materials in the examination: textbook, lecture slides, assignment handout and sample solutions, and self-made notes
  - Internet searching is NOT allowed

Guidelines for Assignment 1

Guidelines for Assignment 1

- Demos of using Google Colab for Assignment 1
  - Step1: download the files (4 python notebooks) from Moodle.
  - Step2: visit https://colab.research.google.com/
  - Step3: choose “Upload” to upload hw1_q1_written.ipynb
  - Step4: answer the questions in this file.
  - Step5: execute all the code blocks to print the results.
  - Step6: save and download this finished .ipynb file
  - Step7: repeat Step 3-6 for all assignment files.
  - Step8: put the finished 4 files in one .zip, and name it using your uid, like 3009666.zip
  - Step9: submit the .zip on Moodle.

Guidelines for Assignment 1

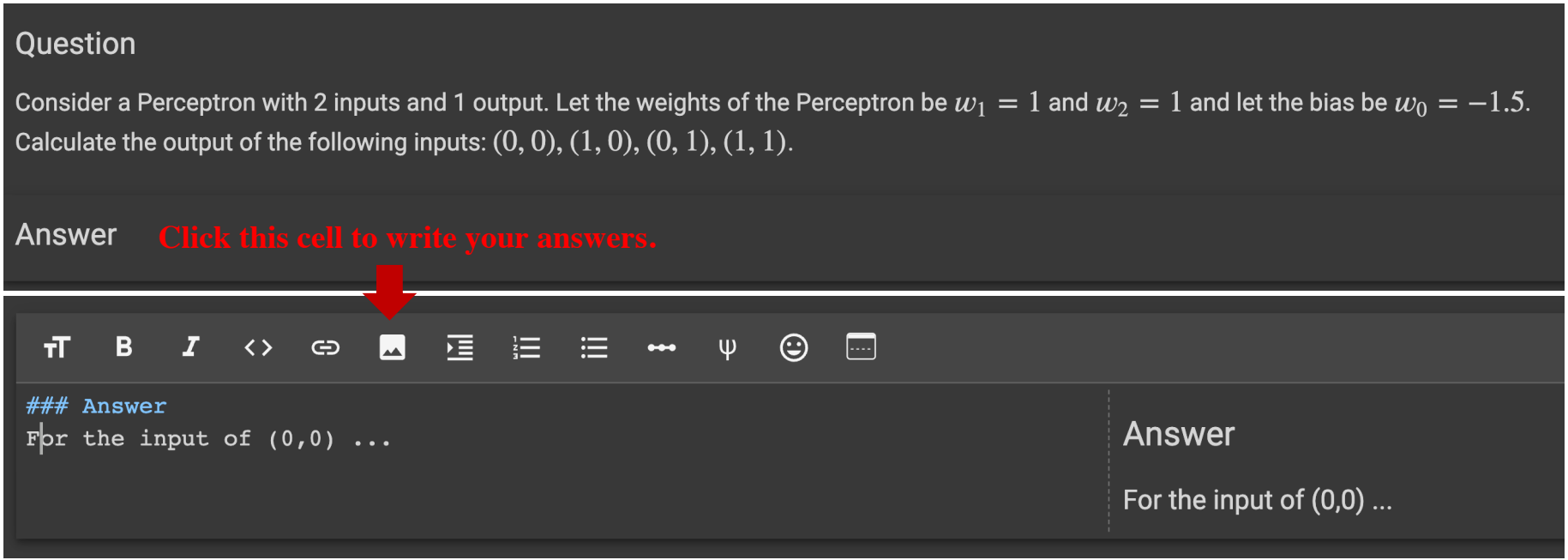
- Assignment 1: Overview
  - HW1-Q1: Written Questions (50 points)
  - HW1-Q2: Perceptron Boolean Operators (10 points)
  - HW1-Q3: Decision Boundary (20 points)
  - HW1-Q4: Hand-written Digits Classification (20 points)

Guidelines for Assignment 1

- HW1-Q1: Written Questions(50 points)
  - Q1-1: Perceptron Basics (5 points)
  - Q1-2: Boolean Operators (5 points)
  - Q1-3: Parity Check (5 points)
  - Q1-4 Support Vectors (5 points)
  - Q1-5: Entropy (5 points)
  - Q1-6: Decision Tree (25 points)

Guidelines for Assignment 1

- HW1-Q1-1: Perceptron Basics (5 points)
  - Write your answer in the given cell using using Markdown grammar and Latex math equations



Guidelines for Assignment 1

- HW1-Q1-2: Boolean Operators (5 points)
  - Filling the tables with right output value for different Boolean operators.

Answer

`NOT`: (example table)

x_1	output
0	1
1	0

`AND`:

x_1	x_2	output
0	0	
0	1	
1	0	
1	1	

`OR`:

x_1	x_2	output
0	0	
0	1	
1	0	
1	1	

`NAND`:

x_1	x_2	output
0	0	
0	1	
1	0	
1	1	

`NOR`:

x_1	x_2	output
0	0	
0	1	
1	0	
1	1	

Guidelines for Assignment 1

- HW1-Q1-3: Parity Check (5 points)
  - Explanation: the input could only be 0/1. For example, if the 3 inputs are (0,1,1) ,the number of 1 equals to 2. As 2 is even, the output is expected to be 1.
  - You are expected to calculate the weights for the perceptron that can do parity check, or you should prove why the perceptron can not.

Question

The parity problem returns 1 if the number of inputs that are 1 is even, and 0 otherwise.

Can a perceptron learn this problem for 3 inputs?

Answer

Guidelines for Assignment 1

- HW1-Q1-4: Support Vectors (5 points)
  - First, run the given code(show in right) to draw the separating line.
  - Then, answer the questions after observing the data points.

Question

Suppose that the following are a set of point in two classes:

- Class1: $(1, 1), (1, 2), (2, 1)$
- Class2: $(0, 0), (1, 0), (0, 1)$

1. Plot them and find the optimal separating line. What are the support vectors?
2. What is the meaning of support vectors?

```
[ ] %matplotlib inline
    import matplotlib.pyplot as plt
    import numpy as np

    # select setosa and versicolor
    c1 = np.array([[1, 1], [1, 2], [2, 1]])
    c2 = np.array([[0, 0], [1, 0], [0, 1]])
    # plot data
    plt.figure(figsize=(5, 5))
    plt.scatter(c1[:, 0], c1[:, 1], color='red', marker='o', label='class1')
    plt.scatter(c2[:, 0], c2[:, 1], color='blue', marker='x', label='class2')

    x = np.linspace(0, 1., 5)
    plt.plot(x, 1 - x, 'b--')
    x = np.linspace(0, 2., 5)
    plt.plot(x, 2 - x, 'r--')
    x = np.linspace(0, 1.5, 5)
    plt.plot(x, 1.5 - x, 'y-')

    plt.xlim(-0.2, 2.5)
    plt.ylim(-0.2, 2.5)
    plt.grid(True)
    plt.tight_layout()
    plt.show()
```

Guidelines for Assignment 1

- HW1-Q1-5: Entropy (5 points)
  - Calculate the entropy according to formulations and explain the meaning.

Question

Suppose that the probability of five events:

- $P(\text{first}) = 0.5$
- $P(\text{second}) = 0.125$
- $P(\text{third}) = 0.125$
- $P(\text{fourth}) = 0.125$
- $P(\text{fifth}) = 0.125$

Calculate the entropy and write down in words what this means.

Answer

Guidelines for Assignment 1

- HW1-Q1-6: Decision Tree (25 points)
  - Given several data samples, compute the Gini impurity and the information gain.
  - Know how to select features to make decision.

Question

The new energy vehicle (NEV) is the growing trend in the automotive industry to replace traditional gas-powered vehicles.

Consider a dataset of customer preference for vehicles. Here are the possible values for each feature:

- Engine: {Gas, NEV}
- Style: {Sedan, SUV}
- Price: {Regular, Luxury}

Note that samples with the same features can have different labels. If the leaf node of a decision tree is not pure, the majority vote is used to determine the output label.

Now, you want to build a decision tree to predict the preference of a customer. In particular, you want to know which feature (among Engine, Style, and Price) is the most important feature to predict the preference. In other words, which feature should be the root node of the decision tree to maximize the information gain?

Your tasks:

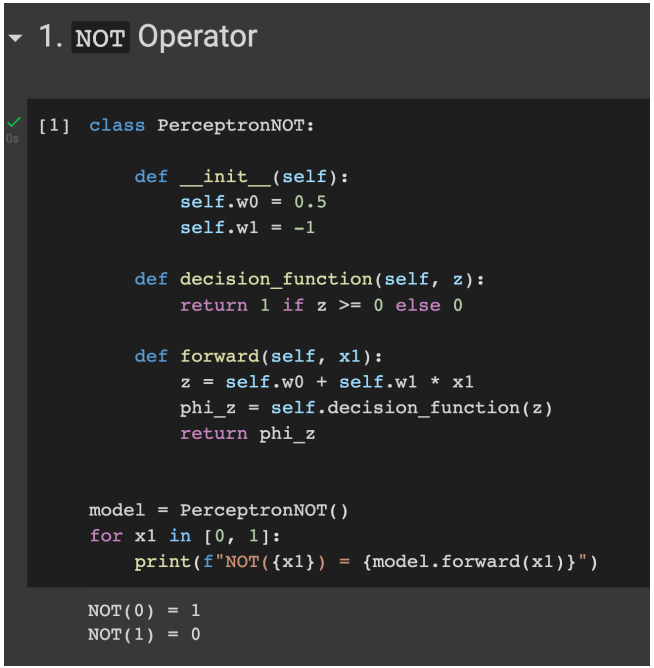
1. Compute Gini impurity at the root node. (5 points)
2. For the 3 features (Engine, Style, and Price), compute the information gain if that feature is used split the root node. (15 points)
3. Conclude which feature is the most important feature to predict the preference as it maximizes the information gain. (5 points)

Here is the dataset:

Engine	Style	Price	Preference?
Gas	Sedan	Regular	Yes
Gas	Sedan	Regular	No
Gas	Sedan	Luxury	No
Gas	Sedan	Luxury	No
Gas	SUV	Regular	Yes
Gas	SUV	Luxury	Yes
Gas	SUV	Luxury	No
NEV	Sedan	Regular	Yes
NEV	Sedan	Regular	Yes
NEV	Sedan	Luxury	Yes
NEV	Sedan	Luxury	Yes
NEV	SUV	Regular	No
NEV	SUV	Regular	Yes
NEV	SUV	Regular	Yes
NEV	SUV	Luxury	Yes
NEV	SUV	Luxury	No

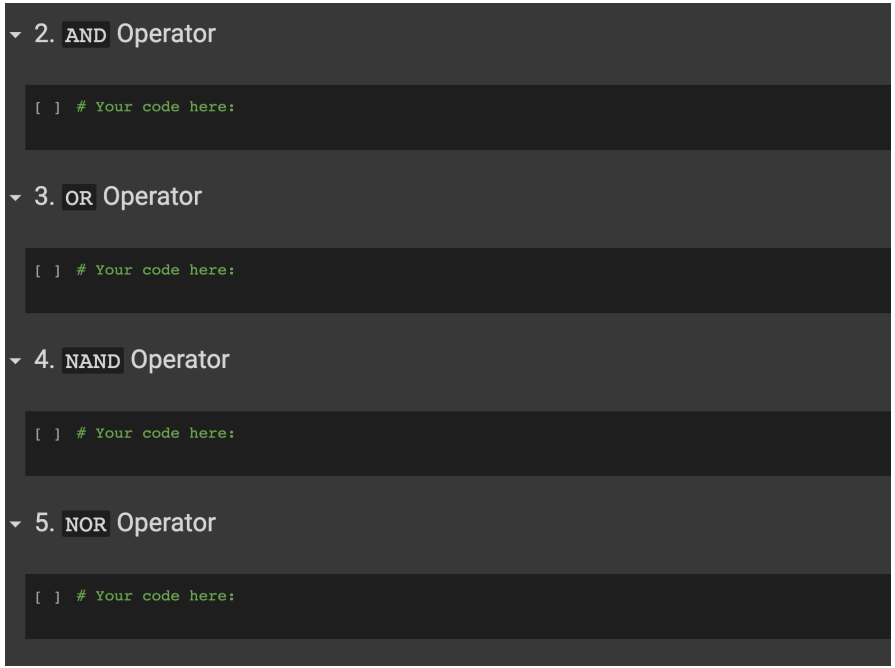
Guidelines for Assignment 1

- HW1-Q2: Perceptron Boolean Operators (10 points)
  - The demo code for “NOT Operator” is provided, do not change it
  - Implement other operators referring to the demo code.



1. NOT Operator

```
[1] class PerceptronNOT:

        def __init__(self):
            self.w0 = 0.5
            self.w1 = -1

        def decision_function(self, z):
            return 1 if z >= 0 else 0

        def forward(self, x1):
            z = self.w0 + self.w1 * x1
            phi_z = self.decision_function(z)
            return phi_z


    model = PerceptronNOT()
    for x1 in [0, 1]:
        print(f"NOT({x1}) = {model.forward(x1)}")

NOT(0) = 1
NOT(1) = 0
```



2. AND Operator

```
[ ] # Your code here:
```

3. OR Operator

```
[ ] # Your code here:
```

4. NAND Operator

```
[ ] # Your code here:
```

5. NOR Operator

```
[ ] # Your code here:
```

Guidelines for Assignment 1

- HW1-Q3: Decision Boundary (20 points)
  - A lot of supporting codes are provided, run the cells one by one. Do not change them !

```
[ ] import matplotlib.pyplot as plt
    from matplotlib.colors import ListedColormap
    import numpy as np
    import pandas as pd
```

```
"""
Note: Do not change the code in this cell.
"""


class Perceptron(object):

    def __init__(self, eta=0.01, n_iter=10):
        """
        Args:
            eta (float, optional): Learning rate. Defaults to 0.01.
            n_iter (int, optional): Number of iterations. Defaults to 10.
        """
        self.eta = eta
        self.n_iter = n_iter

    def fit(self, xs, ys):
        """
        Fit training data.

        Args:
            xs (array-like): Training vectors, shape = (n_samples, n_features).
            ys (array-like): Target values, shape = (n_samples,).
```

```
"""
Note: Do not change the code in this cell.
"""


def fetch_dataset():
    """
    Download and get a subset the UCI Iris dataset.

    Returns:
        (xs, ys), where xs has shape (100, 2) and ys has shape (100,).
    """
    # Download dataset
    url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
    df = pd.read_csv(url, header=None)
    df.tail()

    # Select setosa and versicolor
    num_samples = 100
    ys = df.iloc[0:num_samples, 4].values
    ys = np.where(ys == "Iris-setosa", -1, 1)

    # Extract sepal length and petal length
    xs = df.iloc[0:num_samples, [0, 2]].values

    return xs, ys
```

Guidelines for Assignment 1

- HW1-Q3: Decision Boundary (20 points)
  - The supporting codes would guide you to:
    - Train a perceptron on given dataset.
    - Visualize the dataset and decision regions.
    - Draw a random decision boundary
  - What you need to do:
    - Write code in the last cell to compute the actual decision boundary for the trained perceptron.

Guidelines for Assignment 1

- HW1-Q4: Hand-written Digits Classification (20 points)
  - We provide the template code for perceptron.
  - You are required to use several other classifiers like LR, SVM, KNN for digits classification, and analyze the results.

Classifier #1 Perceptron

```
# Example code, including training and testing, to observe the accuracies.
from sklearn.linear_model import Perceptron

# Tune the eta0 hyperparameter.
eat0_list = [0.0001, 0.001, 0.01, 0.1, 1]

# Your code here.
accuracies = []
for eta0 in eat0_list:
    model = Perceptron(max_iter=100, tol=1e-3, eta0=eta0)
    model.fit(xs_train, ys_train)
    ys_pred = model.predict(xs_test)
    accuracy = get_accuracy(ys_test, ys_pred)
    accuracies.append(accuracy)

for eta0, accuracy in zip(eat0_list, accuracies):
    print(f"eta0 = {eta0:.4f}, accuracy = {accuracy:.4f}")
```

```
eta0 = 0.0001, accuracy = 0.9500
eta0 = 0.0010, accuracy = 0.9167
eta0 = 0.0100, accuracy = 0.9389
eta0 = 0.1000, accuracy = 0.9333
eta0 = 1.0000, accuracy = 0.9333
```

Classifier #2 Logistic Regression

```
[ ] # Your code, including training and testing, to observe the accuracies.

    from sklearn.linear_model import LogisticRegression

    # Tune the C hyperparameter.
    C_list = [1e-3, 0.001, 0.01, 1, 10, 100]

    # Your code here.
```

...

Classifier #6 KNN

```
[ ] # Your code, including training and testing, to observe the accuracies.

    from sklearn.neighbors import KNeighborsClassifier

    # Tune the n_neighbors hyperparameter.
    n_neighbors_list = [2, 3, 5, 10, 20]

    # Your code here.
```

Further Questions

- Mr. Xi Chen (email: xchen2@cs.hku.hk)
- Office: HW-RSC or zoom (https://hku.zoom.us/my/xavier.xichen)
- Consultation hours : Wednesday, 10:00 am - 12:00 pm

- Mr. Yixing Lao (email: laoyx@connect.hku.hk)
- Office: HW-RSC or zoom (https://hku.zoom.us/my/laoyixing)
- Consultation hours: Tuesday, 2:00 pm - 4:00 pm
- Please send an email before our meeting

Q & A