lecture0-welcome

September 15, 2020

1 Lecture 0: Welcome and Course Information

1.0.1 Applied Machine Learning

Volodymyr Kuleshov Cornell Tech

2 Welcome to Lecture 0 of AML

Today, we will cover the following: * Introduction to machine learning. Why is ML interesting? * Overview of the course. Content, format, materials. * Course logistics. Requirements, lecture times, assignments, etc. * Meet the teaching team.

3 ML in Everyday Life: Search Engines

You use machine learninng every day when you run a search engine query.

4 ML in Everyday Life: Personal Assitants

Machine learning also powers the speech recognition, question answering and other intelligent capabilities of smartphone assistants like Apple Siri.

5 ML in Everyday Life: Spam/Fraud Detection

Machine learning is used in every spam filter, such as in Gmail.

ML systems are also used by credit card companies and banks to automatically detect fraudulent behavior.

6 ML in Everyday Life: Self-Driving Cars

One of the most exciting and cutting-edge uses of machine learning algorithms are in autonomous vehicles.

7 A Definition of Machine Learning

In 1959, Arthur Samuel defined machine learning as follows.

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

What does "learn" and "explicitly programmed" mean here? Let's look at an example.

8 An Example: Self Driving Cars

A self-driving car system uses dozens of components that include detection of cars, pedestrians, and other objects.

9 Self Driving Cars: A Rule-Based Algorithm

One way to build a detection system is to write down rules.

[2]:

```
# pseudocode example for a rule-based classification system
object = camera.get_object()
if object.has_wheels(): # does the object have wheels?
    if len(object.wheels) == 4: return "Car" # four wheels => car
    elif len(object.wheels) == 2:,
        if object.seen_from_back():
            return "Car" # viewed from back, car has 2 wheels
        else:
            return "Bicycle" # normally, 2 wheels => bicycle
return "Unknown" # no wheels? we don't know what it is
```

In practice, it's almost impossible for a human to specify all the edge cases.

10 Self Driving Cars: An ML Approach

The machine learning approach is to teach a computer how to do detection by showing it many examples of different objects.

No manual programming is needed: the computer learns what defines a pedestrian or a car on its own!

11 Revisiting Our Definition of ML

> Machine learning is a field of study that gives computers the ability to learn without being explicitly programmed. (Arthur Samuel, 1959.)

This principle can be applied to countless domains: medical diagnosis, factory automation, machine translation, and many more!

12 Why Machine Learning?

Why is this approach to building software interesting?

- It allows building practical systems for real-world applications that couldn't be solved otherwise.
- Learning is wildly regarded as a key approach towards building general-purpose artificial intelligence systems.
- The science and engineering of machine learning offers insights into human intelligence.

# Part 2: About the Course

Next, let's look at the machine learning topics that we will cover.

13 Teaching Approach

The focus of this course is on applied machine learning. * We will cover a broad toolset of core algorithms from many different subfields of ML. * We will emphasize applications and show how to implement and apply algorithms via examples and exercises.

14 What You Will Learn

- What are the core algorithms of ML and how to define them in mathematical language.
- How to implement algorithms from scratch as well as using ML libraries and apply them to problems in computer vision, language processing, medical analysis, and more.
- Why machine learning algorithms work and how to use that knowledge to debug and improve them.

15 Course Content

The course spans about 25 lectures approximately divided up into a set of blocks: 1. Supervised and unsupervised algorithms. 2. Foundations of machine learning. 4. Applying machine learning in practice. 5. Advanced topics and guest lectures.

16 Executable Course Materials

The core materials for this course (including the slides!) are created using Jupyter notebooks. * We are going to embed an execute code directly in the slides and use that to demonstrate algorithms. * These slides can be downloaded locally and all the code can be reproduced.

[29]:

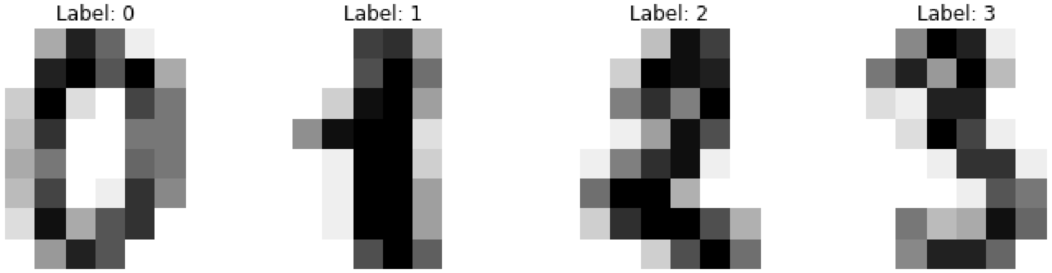
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets, neural_network
plt.rcParams['figure.figsize'] = [12, 4]
```

We can use these libraries to load a simple datasets of handwritten digits.

[7]:

```
# https://scikit-learn.org/stable/auto_examples/classification/
 ↪plot_digits_classification.html
# load the digits dataset
digits = datasets.load_digits()

# The data that we are interested in is made of 8x8 images of digits, let's
# have a look at the first 4 images.
_, axes = plt.subplots(1, 4)
images_and_labels = list(zip(digits.images, digits.target))
for ax, (image, label) in zip(axes, images_and_labels[:4]):
    ax.set_axis_off()
    ax.imshow(image, cmap=plt.cm.gray_r, interpolation='nearest')
    ax.set_title('Label: %i' % label)
```



We can now load and train this algorithm inside the slides.

[30]:

```
np.random.seed(0)
# To apply a classifier on this data, we need to flatten the image, to
# turn the data in a (samples, feature) matrix:
data = digits.images.reshape((len(digits.images), -1))

# create a small neural network classifier
from sklearn.neural_network import MLPClassifier
classifier = MLPClassifier(alpha=1e-3)

# Split data into train and test subsets
X_train, X_test, y_train, y_test = sk.model_selection.train_test_split(data,␣
 ↪digits.target, test_size=0.5, shuffle=False)
```

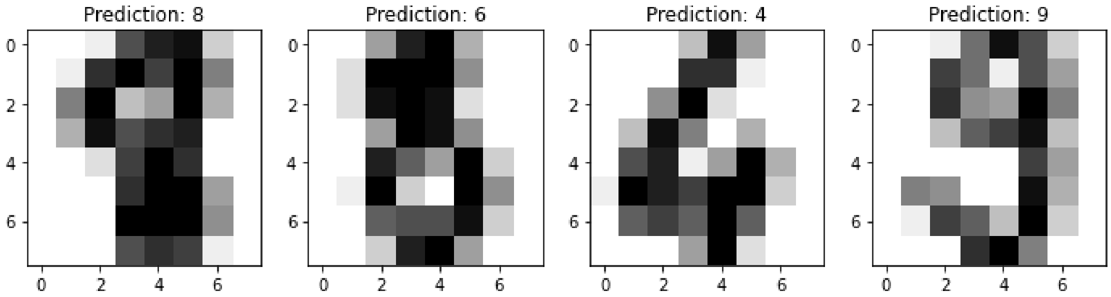
```
# We learn the digits on the first half of the digits
classifier.fit(X_train, y_train)

# Now predict the value of the digit on the second half:
predicted = classifier.predict(X_test)
```

We can now visualize the results.

[31]:

```
_, axes = plt.subplots(1, 4)
images_and_predictions = list(zip(digits.images[n_samples // 2:], predicted))
for ax, (image, prediction) in zip(axes, images_and_predictions[:4]):
    ax.imshow(image, cmap=plt.cm.gray_r, interpolation='nearest')
    ax.set_title('Prediction: %i' % prediction)
```



# Part 3: Logistics and Other Information

We will go over some practical bits of information.

17 Teaching Staff

The Zoom URL for office hours are available in Canvas. * Volodymyr Kuleshov (Instructor). Office Hours: Tue 10pm-11pm ET; Thu 1:50pm-2:45pm ET. * Jin Sun (Course Coordinator). * Andrew Bennett (Teaching Assistant). Office Hours: Wed and Fri 3pm-4pm ET. * Kai Zhang (Teaching Assistant). Office Hours: Tue and Fri 8am-9am ET. * Shachi Deshpande (Teaching Assistant). Office Hours: Mon and Fri 11pm-12am ET.

18 Course Format: Online Lectures

Instruction will be fully remote, and the lectures will be given via Zoom. * The class will be held twice a week, Tuesdays and Thursdays. * Each class will be given twice: 1-1:50pm and 11-11:50pm. Material in each session is the **same**. Choose the session that works best for you. * Zoom links are available in Canvas.

19 Course Format: Reverse Classrom

The format of this course will be that of the “reverse classroom”. * Pre-recorded lecture videos will be made available online ahead of time. You should watch them ahead of each weekly lecture. * In-class discussions will focus on answering student questions, going over homework problems, doing tutorials. * First few weeks will consist of tutorials (programming, probability, linear algebra).

20 Logistics

- We will use Canvas for announcements, discussions, hosting materials, etc. (https://canvas.cornell.edu/courses/19987/)
- We will use Gradescope for grading and submitting assignments.
- There is no required textbook, but we recommend Elements of Statistical Learning by Hastie, Tibshirani, and Friedman.

21 Prerequisites. Is This Course For You?

This course is designed to aimed at a very general technical audience. Main requirements are: * Programming experience (at least 1 year), preferably in Python. * College-level linear algebra. Matrix operations, the SVD decomposition, etc. * College-level probability. Probability distributions, random variables, Bayes’ rule, etc.

22 Grading

Grading will be based on: 1. Four assignments covering the material seen in the class (15% of the grade each). 2. A class project, due at the end of the semester (40% of the grade).

There will be no prelim or final.

23 Assignments

Assignments will be released every two weeks or so via Canvas/Gradescope. * They will include a mix of programming and theory exercises. * Will be due two weeks after release. * Assignment 0 is released today; it will not count towards grade, but you should do it.

24 Project

The course project will involve applying ML to a real-world problem of your choice. * Final result is 5 page writeup, like a research paper. * Three components: proposal (5%), milestone (10%) and final (25%). * Should be done in teams of up to three. * Teachinng staff will suggest projects. Stay tuned in Canvas.

25 Late Submissions

- 6 Late days in total.
- 20%/day penalty after that.
- No more than 3 days per submission.
- No late days for final projects, because we need to grade them quickly.
- When submitting as a team, you must each use a late day.

[ ]: