Basics of Machine Learning in Python

Outline

www.github.com/ajfisch/mit_deeplearning_bootcamp

Python Review

- Data Types
- Control Flow
- File I/O
- Modules

Lists

- Store generic data in an array:
  - [0]
  - [1.2, 2.3]
  - [4, “hello”, “world”, 5.0]

- Indexing items:
  - students = [“Adam”, “Regina”, “Tommi”]
  - students[0] → “Adam”
  - students[-1] → “Tommi”

- Iterate:
  - for student in students:
    print(student) → “Adam”, “Regina”, ...

Dictionaries

- Dictionaries are lookup tables.
- They map from a “key” to a “value”.
  - symbol_to_name = {“MA”: “Massachusetts”,
    “NJ”: “New Jersey”,
    “NY”: “New York”}
- Indexing items:
  - symbol_to_name[“MA”] → “Massachusetts”
  - symbol_to_name.keys() → [“MA”, “NJ”, “NY”]
  - symbol_to_name.values() → [“Massachusetts”, “New Jersey”, “New York”]
- Keys can be any immutable value (numbers, strings, tuples, etc)

Control Flow

- Things that evaluate to “false”:
  - The keyword **False**
  - 0, 0.0
  - Empty containers ([], {}, …}
- Things that evaluate to “true”:
  - The keyword **True**
  - Anything not False
- Results of equality statements (==, >, <, …)
- When used as numbers, booleans act as 0 or 1.

If Statement

- if mode == “train”:
    (do something for training)
  else:
    (do something for testing)

- predictions = [1.0, 0.0, …]
  labels = [1.0, 1.0, …]
  num_correct = 0
  for i in range(len(predictions)):
    if predictions[i] == labels[i]:
      num_correct = num_correct + 1

List Comprehensions

- predictions = [1.0, 0.0, …]
  labels = [1.0, 1.0, …]
  scores = []
  for i in range(len(predictions)):
  scores.append(predictions[i] == labels[i])

- scores = [predictions[i] == labels[i] for i in range(len(predictions))]

- scores = [x == y for x, y in zip(predictions, labels)]

File Input and Output

- Reading…
  - f_in = open(“input.txt”, “r”)
  - first_line = f_in.readline()
  - remaining_lines = [line for line in f_in]
- Writing…
  - f_out = open(“output.txt”, “w”)
  - f_out.write(“Hello World\n”)
- Can also use pickle to write/read serialized objects.
  - object = pickle.load(open(“pickled_file.pkl”, “rb”))
  - pickle.dump(object, open(“pickled_file.pkl”, “wb”))

Modules

- When a program starts it only has access to basic functions and classes.
- Modules can be “imported” to give additional functionality.
- import math
  math.log(math.e)
- from math import log, e
  log(e)

SciKit Learn

- SciKit-Learn is a Python package that provides many machine learning algorithms and related utilities.
- Built on top of SciPy and Numpy.
- Open source and free to use!
- One of the best general purpose ML tools.

SciKit Learn API

- SciKit learn is centered around the concept of **Estimators.**

- class **Estimator**(object):

```
    def fit(self, X, y=None):
        """Fits estimator to data..."""
        ...

    def predict(self, X):
        """Returns prediction for new inputs."""
        ...
```