Machine Learning Lab II

- PS1
- NUMPY

PS1 Q&A

Numpy Tutorial

Overview

- Numpy: basic objects, methods, functions
- Numpy: linear algebra
- Numpy: random
- Matplotlib: 2D plots
- Matplotlib: 3D plots
- Scipy vs Numpy
- Discuss assignment 1

Numpy

- Fundamental package for working with N-dimensional array objects (vector, matrix, tensor, …)
- Numpy arrays are a fundamental data type for some other packages to use
- Numpy has many specialized modules and functions:
  - numpy.linalg (Linear algebra)
  - numpy.random (Random sampling)
  - numpy.fft (Discrete Fourier transform)
  - sorting/searching/counting
  - math functions
  - numpy.testing (unit test support)

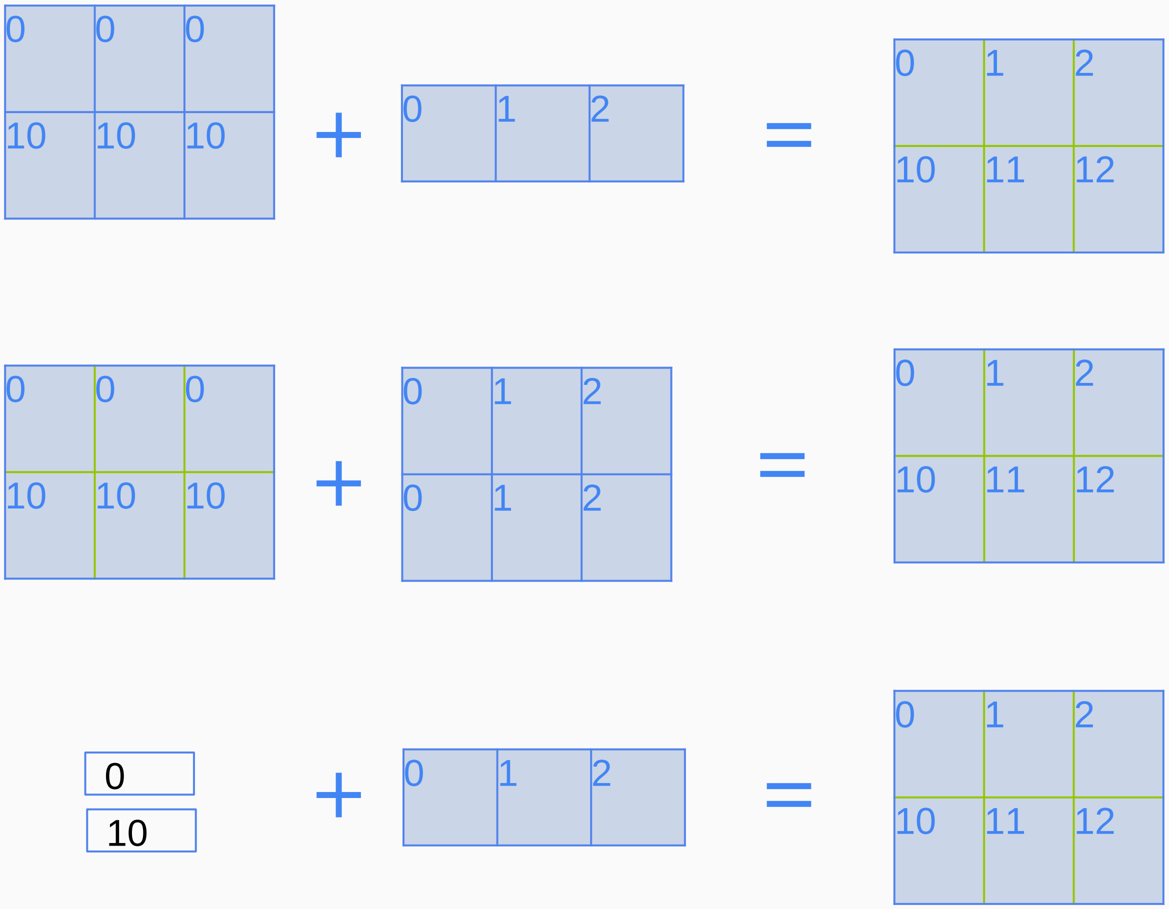
Declaring a Numpy array

- Each Numpy array has some attributes:
    - shape (a tuple of the size in each dimension)
    - dtype (data type of entries)
    - size (total # of entries)
    - ndim (# of dimensions)
    - T (transpose)

What can you do?

- Add two arrays
- Add all entries in one array
- Multiply two arrays (1D, 2D)
- Take the exponential of each element in an array
- Multiply an array by a scalar
- Get the minimum element of an array
- Print a few elements of an array
- Print a single column or row of an array
- Multiply two arrays via matrix multiplication

Array broadcasting



$$\begin{bmatrix}0&0&0\\10&10&10\end{bmatrix} + \begin{bmatrix}0&1&2\end{bmatrix} = \begin{bmatrix}0&1&2\\10&11&12\end{bmatrix}$$

$$\begin{bmatrix}0&0&0\\10&10&10\end{bmatrix} + \begin{bmatrix}0&1&2\\0&1&2\end{bmatrix} = \begin{bmatrix}0&1&2\\10&11&12\end{bmatrix}$$

$$\begin{bmatrix}0\\10\end{bmatrix} + \begin{bmatrix}0&1&2\end{bmatrix} = \begin{bmatrix}0&1&2\\10&11&12\end{bmatrix}$$

Iterating over an array

- Iteration over all elements of array:
    - for element in A.flat

- Iteration over multidimensional arrays is done on slices in the first dimension:
    - for row in A

- Alternatively, could access entries through indices:
    - for i in range(A.shape[0]):
        - for j in range(A.shape[1]):

Reshaping an array

- Use reshape to modify the dimensions of an array while leaving the total number of elements the same
  - A = np.arange(8)
  - A.reshape(2,4)
  - gives [[0,1,2,3],[4,5,6,7]]
- Use resize to remove elements or append 0's in place
  - (size can change under some circumstances*)
  - resize(2,3)
- Use resize to return a copy with removed elements or repeated copies
  - b = resize(a,(2,4))

Numpy: Linear Algebra

name	explanation
dot(a,b)	dot product of two arrays
kron(a,b)	Kronecker product
linalg.norm(x)	matrix or vector norm
linalg.cond(x)	condition number
linalg.solve(A,b)	solve linear system Ax=b
linalg.inv(A)	inverse of A
linalg.pinv(A)	pseudo-inverse of A
linalg.eig(A)	eigenvalues/vectors of square A
linalg.eigvals(A)	eigenvalues of general A
trace(A)	trace (diagonal sum)
linalg.svd(A)	singular value decomposition

http://docs.scipy.org/doc/numpy/reference/routines.linalg.html

Numpy: Random

- x = np.random.randn(50)
- y = 3.5*x+2+np.random.randn(50)*0.3

- If you run this, you'll get different numbers each time, so you might want to use np.random.seed(*) to reproduce a random experiment

Numpy: Random

name	explanation
rand(n0,n1,…)	ndarray of random values from uniform [0,1]
randn(n0,n1,…)	random standard normal
randint(lo, [hi, size])	random integers [lo, hi)
shuffle(seq)	shuffle sequence randomly
choice(seq,[size,replace,p])	sample k items from a 1D array with or without replacement
chisquare(df,[size])	sample from Chi-squared distribution with df degrees of freedom
exponential([scale,size])	sample from exponential distribution

http://docs.scipy.org/doc/numpy/reference/routines.random.html

Matplotlib: 2D plots

- Matplotlib is the 2D Python plotting library
- We'll mostly use matplotlib.pyplot
- There are tons of options, so consult the documentation:
  - http://matplotlib.org/users/beginner.html
- matplotlib.pyplot can do many types of visualizations including:
  - Histograms, bar charts (using hist)
  - Error bars on plots, box plots (using boxplot, errorbar)
  - Scatterplots (using scatter)
  - Line plots (using plot)
  - Contour maps (using contour or tricontour)
  - Images (matrix to image) (using imshow)
  - Stream plots which show derivatives at many locations (streamplot)
  - Pie charts, polar charts (using pie, polar)

Matplotlib: 2D plots

- How do we show two curves on the same plot?
  - import numpy as np
  - import matplotlib.pyplot as plt

  - x = np.linspace(0,np.pi,100)
  - y = np.sin(x)
  - plt.plot(x,y)
  - plt.show()
- More examples: http://matplotlib.org/gallery.html
- Documentation: http://matplotlib.org/api/pyplot_api.html

Scipy vs. Numpy

- Scipy is a library that can work with Numpy arrays, but can achieve better performance and has some more specialized libraries
  - linear algebra (scipy.linalg uses BLAS/LAPACK)
  - statistics (scipy.stats has hypothesis tests, correlation analysis)
  - optimization (scipy.optimize has multiple solvers, gradient checks, simulated annealing)
  - sparse matrices (scipy.sparse supports sparse linear algebra, graph analysis, multiple sparse matrix formats)
  - signal processing (scipy.signal has convolutions, wavelets, splines, filters)

http://docs.scipy.org/doc/scipy/reference/

References

- Standford CME 193 By Eileen Martin
- https://docs.scipy.org/doc/numpy/index.html