Assignment0

September 3, 2018

0.0.1 Task 1: run conda info in terminal.

```
    active environment : None
       user config file : /home/wywang/.condarc
 populated config files :
          conda version : 4.5.4
    conda-build version : 3.10.5
         python version : 3.6.5.final.0
       base environment : /home/wywang/anaconda3  (writable)
           channel URLs : https://repo.anaconda.com/pkgs/main/linux-64
                          https://repo.anaconda.com/pkgs/main/noarch
                          https://repo.anaconda.com/pkgs/free/linux-64
                          https://repo.anaconda.com/pkgs/free/noarch
                          https://repo.anaconda.com/pkgs/r/linux-64
                          https://repo.anaconda.com/pkgs/r/noarch
                          https://repo.anaconda.com/pkgs/pro/linux-64
                          https://repo.anaconda.com/pkgs/pro/noarch
          package cache : /home/wywang/anaconda3/pkgs
                          /home/wywang/.conda/pkgs
       envs directories : /home/wywang/anaconda3/envs
                          /home/wywang/.conda/envs
                platform : linux-64
              user-agent : conda/4.5.4 requests/2.18.4 CPython/3.6.5 Linux/4.15.0-33-generic ubu
                 UID:GID : 1000:1000
              netrc file : None
            offline mode : False
```

0.0.2 Task 2: Run all of python commmands in the table "Linear Algebra Equivalents"

```
In [3]: import numpy as np
        import scipy.linalg

In [8]: a = np.arange(10)
        a.ndim

Out[8]: 1

In [10]: a.size
```

```
Out[10]: 10

In [11]: a.shape

Out[11]: (10,)

In [12]: a.shape[0]

Out[12]: 10

In [15]: np.array([[1,2,3],[5,6,7]])

Out[15]: array([[1, 2, 3],
                [5, 6, 7]])

In [23]: a = np.ones([2,2])
         b = 2 * np.ones([2,2])
         c = 3 * np.ones([2,2])
         d = 4 * np.ones([2,2])
         np.block([[a,b], [c,d]])

Out[23]: array([[1., 1., 2., 2.],
                [1., 1., 2., 2.],
                [3., 3., 4., 4.],
                [3., 3., 4., 4.]])

In [24]: a = np.arange(10)
         a[-1]

Out[24]: 9

In [25]: a[0:4]

Out[25]: array([0, 1, 2, 3])

In [30]: a = np.array([[1,2,3],[4,5,6],[7,8,9]])
         a[1]

Out[30]: array([4, 5, 6])

In [31]: a[-2:]

Out[31]: array([[4, 5, 6],
                [7, 8, 9]])

In [36]: a[0:2][:,-2:]

Out[36]: array([[2, 3],
                [5, 6]])

In [38]: a[np.ix_([0,2],[0,2])]
```

```
Out[38]: array([[1, 3],
                [7, 9]])

In [40]: a[0:3:2,:]

Out[40]: array([[1, 2, 3],
                [7, 8, 9]])

In [41]: a[::2,:]

Out[41]: array([[1, 2, 3],
                [7, 8, 9]])

In [42]: a[::-1,:]

Out[42]: array([[7, 8, 9],
                [4, 5, 6],
                [1, 2, 3]])

In [44]: a[np.r_[:len(a),0]]

Out[44]: array([[1, 2, 3],
                [4, 5, 6],
                [7, 8, 9],
                [1, 2, 3]])

In [45]: a.transpose()

Out[45]: array([[1, 4, 7],
                [2, 5, 8],
                [3, 6, 9]])

In [46]: a.conj().transpose()

Out[46]: array([[1, 4, 7],
                [2, 5, 8],
                [3, 6, 9]])

In [47]: a = np.array([[1,2],[3,4]])
         b = np.array([[5,6],[7,8]])
         a@b

Out[47]: array([[19, 22],
                [43, 50]])

In [48]: a*b

Out[48]: array([[ 5, 12],
                [21, 32]])

In [49]: a/b
```

```
Out[49]: array([[0.2       , 0.33333333],
                [0.42857143, 0.5       ]])

In [50]: a**2

Out[50]: array([[ 1,  4],
                [ 9, 16]])

In [61]: a = np.array([[1,2,3,4]])
         (a>2)

Out[61]: array([[False,  True,  True,  True]])

In [64]: np.nonzero(a>2)

Out[64]: (array([0, 0]), array([2, 3]))

In [65]: a = np.array([[1,2,3,4]])
         a[:,np.nonzero(a>2)[0]]

Out[65]: array([[1, 1]])

In [67]: v = np.array([1,2,3])
         a = np.array([[1,2,3],[4,5,6],[7,8,9]])
         a[:, v.T>=2]

Out[67]: array([[2, 3],
                [5, 6],
                [8, 9]])

In [69]: a[a<3]=0
         a

Out[69]: array([[0, 0, 3],
                [4, 5, 6],
                [7, 8, 9]])

In [70]: a = np.array([[1,2,3],[4,5,6],[7,8,9]])
         a*(a>2)

Out[70]: array([[0, 0, 3],
                [4, 5, 6],
                [7, 8, 9]])

In [72]: a[:]=3
         a

Out[72]: array([[3, 3, 3],
                [3, 3, 3],
                [3, 3, 3]])
```

```
In [74]: b = a.copy()
         b

Out[74]: array([[3, 3, 3],
                [3, 3, 3],
                [3, 3, 3]])

In [76]: b = a[1,:].copy()
         b

Out[76]: array([3, 3, 3])

In [78]: b = a.flatten()
         b

Out[78]: array([3, 3, 3, 3, 3, 3, 3, 3, 3])

In [80]: np.arange(1,11)

Out[80]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])

In [81]: np.arange(10)

Out[81]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

In [88]: np.arange(1, 11).transpose()

Out[88]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])

In [89]: np.zeros((3,4))

Out[89]: array([[0., 0., 0., 0.],
                [0., 0., 0., 0.],
                [0., 0., 0., 0.]])

In [90]: np.zeros((3,4,5))

Out[90]: 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.]]])
```

```
In [91]: np.ones((3,4))

Out[91]: array([[1., 1., 1., 1.],
                [1., 1., 1., 1.],
                [1., 1., 1., 1.]])

In [92]: np.eye(3)

Out[92]: array([[1., 0., 0.],
                [0., 1., 0.],
                [0., 0., 1.]])

In [97]: np.diag((1,2,3,4,5))

Out[97]: array([[1, 0, 0, 0, 0],
                [0, 2, 0, 0, 0],
                [0, 0, 3, 0, 0],
                [0, 0, 0, 4, 0],
                [0, 0, 0, 0, 5]])

In [101]: np.diag((5,5))

Out[101]: array([[5, 0],
                 [0, 5]])

In [102]: np.random.rand(3,4)

Out[102]: array([[0.27358405, 0.75375842, 0.11881394, 0.93192517],
                 [0.29265746, 0.73293208, 0.89438027, 0.44682016],
                 [0.67722277, 0.74351313, 0.73389366, 0.29747123]])

In [103]: np.linspace(1,3,4)

Out[103]: array([1.        , 1.66666667, 2.33333333, 3.        ])

In [104]: np.mgrid[0:9, 0:6]

Out[104]: array([[[0, 0, 0, 0, 0, 0],
                  [1, 1, 1, 1, 1, 1],
                  [2, 2, 2, 2, 2, 2],
                  [3, 3, 3, 3, 3, 3],
                  [4, 4, 4, 4, 4, 4],
                  [5, 5, 5, 5, 5, 5],
                  [6, 6, 6, 6, 6, 6],
                  [7, 7, 7, 7, 7, 7],
                  [8, 8, 8, 8, 8, 8]],

                 [[0, 1, 2, 3, 4, 5],
                  [0, 1, 2, 3, 4, 5],
                  [0, 1, 2, 3, 4, 5],
```

```
                [0, 1, 2, 3, 4, 5],
                [0, 1, 2, 3, 4, 5],
                [0, 1, 2, 3, 4, 5],
                [0, 1, 2, 3, 4, 5],
                [0, 1, 2, 3, 4, 5],
                [0, 1, 2, 3, 4, 5]]])
```

```
In [106]: np.ogrid[0:9, 0:6]
```

```
Out[106]: [array([[0],
                  [1],
                  [2],
                  [3],
                  [4],
                  [5],
                  [6],
                  [7],
                  [8]]), array([[0, 1, 2, 3, 4, 5]])]
```

```
In [107]: np.meshgrid([1,2,4],[2,4,5])
```

```
Out[107]: [array([[1, 2, 4],
                  [1, 2, 4],
                  [1, 2, 4]]), array([[2, 2, 2],
                  [4, 4, 4],
                  [5, 5, 5]])]
```

```
In [109]: np.ix_([1,2,4],[2,4,5])
```

```
Out[109]: (array([[1],
                  [2],
                  [4]]), array([[2, 4, 5]]))
```

```
In [111]: np.tile(1, (2, 3))
```

```
Out[111]: array([[1, 1, 1],
                 [1, 1, 1]])
```

```
In [113]: a = np.array([1,2])
          b = np.array([3,4])
          np.hstack((a,b))
```

```
Out[113]: array([1, 2, 3, 4])
```

```
In [114]: np.vstack((a, b))
```

```
Out[114]: array([[1, 2],
                 [3, 4]])
```

```
In [115]: a.max()
```

```
Out[115]: 2

In [117]: a = np.array([[1,2],[3,4]])
          a.max(0)

Out[117]: array([3, 4])

In [118]: a.max(1)

Out[118]: array([2, 4])

In [120]: a = np.array([[1,2],[3,4]])
          b = np.array([[-1,-2],[5,6]])
          np.maximum(a, b)

Out[120]: array([[1, 2],
                 [5, 6]])

In [122]: np.sqrt(a@a)

Out[122]: array([[2.64575131, 3.16227766],
                 [3.87298335, 4.69041576]])

In [123]: np.logical_and(a, b)

Out[123]: array([[ True,  True],
                 [ True,  True]])

In [126]: np.logical_or(a, b)

Out[126]: array([[ True,  True],
                 [ True,  True]])

In [124]: a & b

Out[124]: array([[1, 2],
                 [1, 4]])

In [125]: a | b

Out[125]: array([[-1, -2],
                 [ 7,  6]])

In [127]: np.linalg.inv(a)

Out[127]: array([[-2. ,  1. ],
                 [ 1.5, -0.5]])

In [128]: np.linalg.pinv(a)

Out[128]: array([[-2. ,  1. ],
                 [ 1.5, -0.5]])
```

```
In [129]: np.linalg.matrix_rank(a)

Out[129]: 2

In [130]: np.linalg.solve(a, b)

Out[130]: array([[ 7., 10.],
                 [-4., -6.]])

In [131]: np.linalg.solve(a.T, b.T)

Out[131]: array([[-1., -1.],
                 [ 0.,  2.]])

In [138]: U, S, Vh = np.linalg.svd(a)
          V = Vh.T
          print(U)
          print(S)
          print(V)

[[-0.40455358 -0.9145143 ]
 [-0.9145143   0.40455358]]
[5.4649857  0.36596619]
[[-0.57604844  0.81741556]
 [-0.81741556 -0.57604844]]


In [140]: a = np.array([[1,2],[0,3]])
          np.linalg.cholesky(a).T

Out[140]: array([[1.        , 0.        ],
                 [0.        , 1.73205081]])

In [143]: D, V = np.linalg.eig(a)
          print(D)
          print(V)

[1. 3.]
[[1.         0.70710678]
 [0.         0.70710678]]


In [147]: Q, R = scipy.linalg.qr(a)
          print(Q)
          print(R)

[[ 1.  0.]
 [-0.  1.]]
[[1. 2.]
 [0. 3.]]
```

```
In [154]: a = scipy.array([[1,2],[0,3]])
          P, L, U = scipy.linalg.lu(a)
          print(P)
          print(L)
          print(U)
```

```
[[1. 0.]
 [0. 1.]]
[[1. 0.]
 [0. 1.]]
[[1. 2.]
 [0. 3.]]
```

```
In [158]: from scipy.sparse import *
          a = scipy.array([[1,2],[0,3]])
          b = scipy.array([[1],[2]])
          scipy.sparse.linalg.cg(a, b)
```

```
Out[158]: (array([ 0.75743899, -1.65423765]), 20)
```

```
In [163]: a = np.arange(10)
          b = np.fft.fft(a)
          print(b)
```

```
[45. +0.j         -5.+15.38841769j -5. +6.8819096j  -5. +3.63271264j
 -5. +1.62459848j -5. +0.j         -5. -1.62459848j -5. -3.63271264j
 -5. -6.8819096j  -5.-15.38841769j]
```

```
In [164]: np.fft.ifft(b)
```

```
Out[164]: array([7.10542736e-16+0.j, 1.00000000e+00+0.j, 2.00000000e+00+0.j,
                 3.00000000e+00+0.j, 4.00000000e+00+0.j, 5.00000000e+00+0.j,
                 6.00000000e+00+0.j, 7.00000000e+00+0.j, 8.00000000e+00+0.j,
                 9.00000000e+00+0.j])
```

```
In [170]: a = np.random.rand(3,4)
          np.sort(a)
```

```
Out[170]: array([[0.13775675, 0.35762676, 0.78258167, 0.83157544],
                 [0.02734015, 0.05892903, 0.53870447, 0.58131199],
                 [0.18879844, 0.54436066, 0.61407269, 0.77115915]])
```

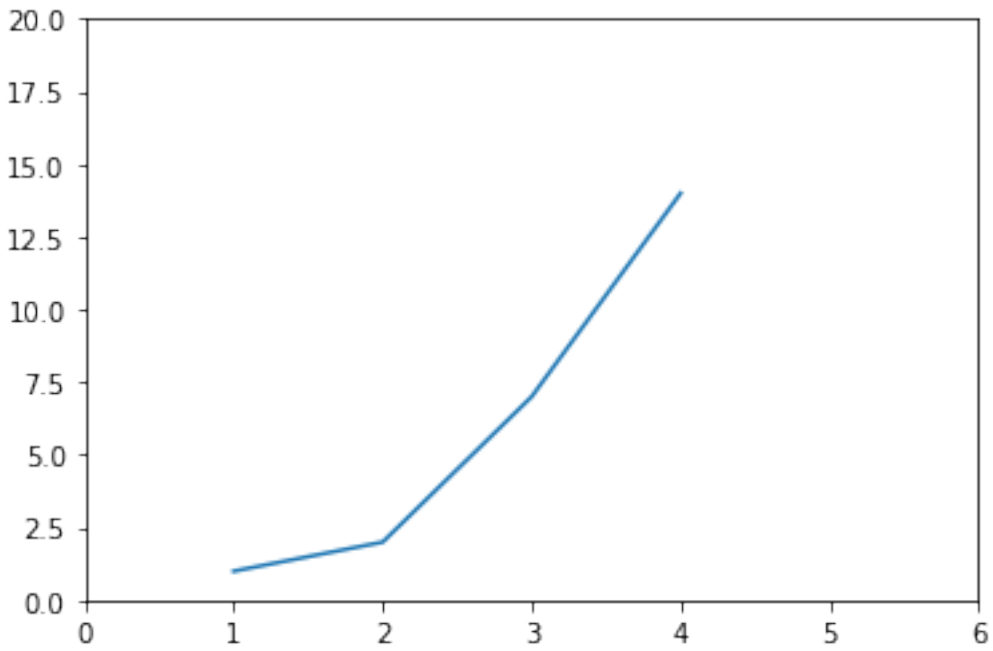
```
In [176]: a = np.random.rand(3,4)
          I = np.argsort(a[:,1])
          a[I, :]
```

```
Out[176]: array([[0.17371901, 0.0678271 , 0.11320244, 0.17468762],
                 [0.30527243, 0.45604222, 0.18982779, 0.87278614],
                 [0.64538564, 0.99044285, 0.18857263, 0.72564055]])
```

```
In [180]: a = np.array([[1,2],[3,4]])
          b = np.array([[5,6],[7,8]])
          np.linalg.lstsq(a, b, rcond=-1)

Out[180]: (array([[-3., -4.],
                  [ 4.,  5.]]),
           array([], dtype=float64),
           2,
           array([5.4649857 , 0.36596619]))

In [187]: from scipy.signal import *
          x = np.arange(100)
          q = 10
          scipy.signal.resample(x, len(x)//q)

Out[187]: array([44.5       ,  1.96583317, 23.64081788, 28.26920094, 40.5046528 ,
                 50.5       , 58.4953472 , 72.73079906, 75.35918212, 99.03416683])

In [191]: a = np.array([[1,1,2,2],[3,3,4,4]])
          np.unique(a)

Out[191]: array([1, 2, 3, 4])

In [192]: a.squeeze()

Out[192]: array([[1, 1, 2, 2],
                 [3, 3, 4, 4]])
```

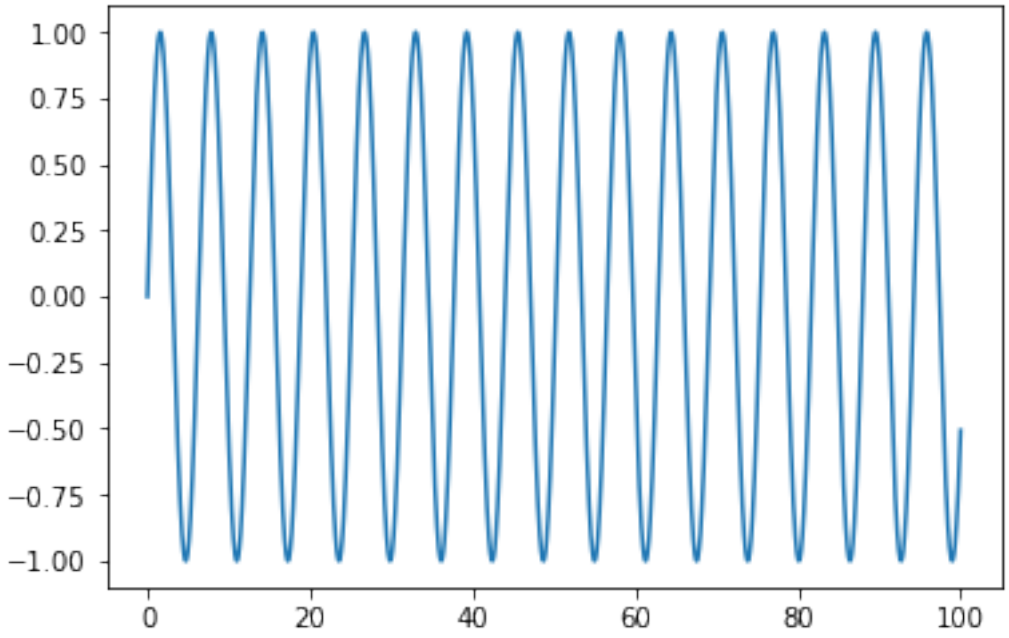
0.0.3 Task3: Make Figure

```
In [194]: import matplotlib.pyplot as plt
          plt.plot([1,2,3,4], [1,2,7,14])
          plt.axis([0, 6, 0, 20])
          plt.show()
```



0.0.4 Task4: Make Figure of my own choice

```
In [197]: import matplotlib.pyplot as plt
          xx = np.linspace(0,100,10000)
          yy = np.sin(xx)
          plt.plot(xx, yy)
          plt.show()
```



0.0.5 Task5: VCS Account

My github account is https://github.com/wywang21

0.0.6 Task6: Repository for this class

The url of my repository is: https://github.com/wywang21/deeplearning2018 I created a new directory assignment0/ in this repository and I also created a helloWorld project in this directory.