Assignment 0

Hao Qi

Rice University

COMP 576 - An Introduction to Deep Learning

hq15@rice.edu

Due Date: September 16

Contents

1 Task 1 **3**

2 Task 2 **4**

3 Task 3 **26**

4 Task 4 **27**

5 Task 5 **28**

6 Task 6 **28**

1 Task 1

```
(deep_learning) qihao@qihaodeMacBook-Pro ~ % conda info

     active environment : deep_learning
    active env location : /opt/anaconda3/envs/deep_learning
            shell level : 2
       user config file : /Users/qihao/.condarc
 populated config files : /Users/qihao/.condarc
          conda version : 24.11.2
    conda-build version : 24.5.1
         python version : 3.12.4.final.0
                 solver : libmamba (default)
       virtual packages : __archspec=1=m1
                          __conda=24.11.2=0
                          __osx=14.4=0
                          __unix=0=0
       base environment : /opt/anaconda3  (writable)
      conda av data dir : /opt/anaconda3/etc/conda
  conda av metadata url : None
           channel URLs : https://repo.anaconda.com/pkgs/main/osx-arm64
                          https://repo.anaconda.com/pkgs/main/noarch
                          https://repo.anaconda.com/pkgs/r/osx-arm64
                          https://repo.anaconda.com/pkgs/r/noarch
          package cache : /opt/anaconda3/pkgs
                          /Users/qihao/.conda/pkgs
       envs directories : /opt/anaconda3/envs
                          /Users/qihao/.conda/envs
               platform : osx-arm64
             user-agent : conda/24.11.2 requests/2.32.2 CPython/3.12.4 Darwin
         /23.4.0 OSX/14.4 solver/libmamba conda-libmamba-solver/24.1.0
         libmambapy/1.5.8 aau/0.4.4 c/6PMS291KB2ESac0vsCdKdg s/
         IpdopgJn9qh4SzdbjGs2Kg e/z8zV5Oz0-TqBYW9WZuYTzg
                UID:GID : 501:20
             netrc file : None
           offline mode : False
```

2 Task 2

```
a0.shape / b0.shape / v.shape
(21, 9) (21, 9) (9,)

ndims(a0)
2

numel(a0)
189

size(a0)
[21 9]

size(a0,2)
9

np.array([[1,2,3],[4,5,6]])
[[1. 2. 3.]
 [4. 5. 6.]]

np.block([[I,1],[2,3I]])
[[1. 0. 1. 1.]
 [0. 1. 1. 1.]
 [2. 2. 3. 0.]
 [2. 2. 0. 3.]]

a[-1]
[0.81 0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89]
```

```
a[1,4]
0.14

a[1,:]
[0.1 0.11 0.12 0.13 0.14 0.15 0.16 0.17 0.18]

a[:5,:]
[[0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09]
 [0.1 0.11 0.12 0.13 0.14 0.15 0.16 0.17 0.18]
 [0.19 0.2 0.21 0.22 0.23 0.24 0.25 0.26 0.27]
 [0.28 0.29 0.3 0.31 0.32 0.33 0.34 0.35 0.36]
 [0.37 0.38 0.39 0.4 0.41 0.42 0.43 0.44 0.45]]

a[-5:,:]
[[0.45 0.46 0.47 0.48 0.49 0.5 0.51 0.52 0.53]
 [0.54 0.55 0.56 0.57 0.58 0.59 0.6 0.61 0.62]
 [0.63 0.64 0.65 0.66 0.67 0.68 0.69 0.7 0.71]
 [0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.8 ]
 [0.81 0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89]]

a[0:3,4:9]
[[0.05 0.06 0.07 0.08 0.09]
 [0.14 0.15 0.16 0.17 0.18]
 [0.23 0.24 0.25 0.26 0.27]]

a[np.ix_([1,3,4],[0,2])]
[[0.1 0.12]
 [0.28 0.3 ]
 [0.37 0.39]]

a[2:21:2,:]
[[0.19 0.2 0.21 0.22 0.23 0.24 0.25 0.26 0.27]
 [0.37 0.38 0.39 0.4 0.41 0.42 0.43 0.44 0.45]
 [0.55 0.56 0.57 0.58 0.59 0.6 0.61 0.62 0.63]
 [0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.8 0.81]
 [0.91 0.92 0.93 0.94 0.95 0.96 0.97 0.98 0.99]
 [0.09 0.1 0.11 0.12 0.13 0.14 0.15 0.16 0.17]
 [0.27 0.28 0.29 0.3 0.31 0.32 0.33 0.34 0.35]
 [0.45 0.46 0.47 0.48 0.49 0.5 0.51 0.52 0.53]
 [0.63 0.64 0.65 0.66 0.67 0.68 0.69 0.7 0.71]
 [0.81 0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89]]

a[::2,:]
[[0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09]
 [0.19 0.2 0.21 0.22 0.23 0.24 0.25 0.26 0.27]
 [0.37 0.38 0.39 0.4 0.41 0.42 0.43 0.44 0.45]
 [0.55 0.56 0.57 0.58 0.59 0.6 0.61 0.62 0.63]
 [0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.8 0.81]
 [0.91 0.92 0.93 0.94 0.95 0.96 0.97 0.98 0.99]
 [0.09 0.1 0.11 0.12 0.13 0.14 0.15 0.16 0.17]
 [0.27 0.28 0.29 0.3 0.31 0.32 0.33 0.34 0.35]
 [0.45 0.46 0.47 0.48 0.49 0.5 0.51 0.52 0.53]
 [0.63 0.64 0.65 0.66 0.67 0.68 0.69 0.7 0.71]
 [0.81 0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89]]

a[::-1,:]
[[0.81 0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89]
 [0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.8 ]
 [0.63 0.64 0.65 0.66 0.67 0.68 0.69 0.7 0.71]
 [0.54 0.55 0.56 0.57 0.58 0.59 0.6 0.61 0.62]
 [0.45 0.46 0.47 0.48 0.49 0.5 0.51 0.52 0.53]
 [0.36 0.37 0.38 0.39 0.4 0.41 0.42 0.43 0.44]
 [0.27 0.28 0.29 0.3 0.31 0.32 0.33 0.34 0.35]
 [0.18 0.19 0.2 0.21 0.22 0.23 0.24 0.25 0.26]
 [0.09 0.1 0.11 0.12 0.13 0.14 0.15 0.16 0.17]
 [0. 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08]
 [0.91 0.92 0.93 0.94 0.95 0.96 0.97 0.98 0.99]
 [0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89 0.9 ]
 [0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.8 0.81]
 [0.64 0.65 0.66 0.67 0.68 0.69 0.7 0.71 0.72]
 [0.55 0.56 0.57 0.58 0.59 0.6 0.61 0.62 0.63]
 [0.46 0.47 0.48 0.49 0.5 0.51 0.52 0.53 0.54]
 [0.37 0.38 0.39 0.4 0.41 0.42 0.43 0.44 0.45]
 [0.28 0.29 0.3 0.31 0.32 0.33 0.34 0.35 0.36]
 [0.19 0.2 0.21 0.22 0.23 0.24 0.25 0.26 0.27]
 [0.1 0.11 0.12 0.13 0.14 0.15 0.16 0.17 0.18]
 [0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09]]


a[np.r_[:len(a),0],:]
[[0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09]
 [0.1 0.11 0.12 0.13 0.14 0.15 0.16 0.17 0.18]
 [0.19 0.2 0.21 0.22 0.23 0.24 0.25 0.26 0.27]
 [0.28 0.29 0.3 0.31 0.32 0.33 0.34 0.35 0.36]
 [0.37 0.38 0.39 0.4 0.41 0.42 0.43 0.44 0.45]
 [0.46 0.47 0.48 0.49 0.5 0.51 0.52 0.53 0.54]
 [0.55 0.56 0.57 0.58 0.59 0.6 0.61 0.62 0.63]
 [0.64 0.65 0.66 0.67 0.68 0.69 0.7 0.71 0.72]
 [0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.8 0.81]
 [0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89 0.9 ]
 [0.91 0.92 0.93 0.94 0.95 0.96 0.97 0.98 0.99]
 [0. 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08]
 [0.09 0.1 0.11 0.12 0.13 0.14 0.15 0.16 0.17]
 [0.18 0.19 0.2 0.21 0.22 0.23 0.24 0.25 0.26]
 [0.27 0.28 0.29 0.3 0.31 0.32 0.33 0.34 0.35]
 [0.36 0.37 0.38 0.39 0.4 0.41 0.42 0.43 0.44]
 [0.45 0.46 0.47 0.48 0.49 0.5 0.51 0.52 0.53]
 [0.54 0.55 0.56 0.57 0.58 0.59 0.6 0.61 0.62]
 [0.63 0.64 0.65 0.66 0.67 0.68 0.69 0.7 0.71]
 [0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.8 ]
 [0.81 0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89]
 [0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09]]
```

```
a.T
[[0.01 0.1 0.19 0.28 0.37 0.46 0.55 0.64 0.73 0.82 0.91 0. 0.09 0.18 0.27 0.36
    0.45 0.54 0.63 0.72 0.81]
 [0.02 0.11 0.2 0.29 0.38 0.47 0.56 0.65 0.74 0.83 0.92 0.01 0.1 0.19 0.28
    0.37 0.46 0.55 0.64 0.73 0.82]
 [0.03 0.12 0.21 0.3 0.39 0.48 0.57 0.66 0.75 0.84 0.93 0.02 0.11 0.2 0.29
    0.38 0.47 0.56 0.65 0.74 0.83]
 [0.04 0.13 0.22 0.31 0.4 0.49 0.58 0.67 0.76 0.85 0.94 0.03 0.12 0.21 0.3
    0.39 0.48 0.57 0.66 0.75 0.84]
 [0.05 0.14 0.23 0.32 0.41 0.5 0.59 0.68 0.77 0.86 0.95 0.04 0.13 0.22 0.31
    0.4 0.49 0.58 0.67 0.76 0.85]
 [0.06 0.15 0.24 0.33 0.42 0.51 0.6 0.69 0.78 0.87 0.96 0.05 0.14 0.23 0.32
    0.41 0.5 0.59 0.68 0.77 0.86]
 [0.07 0.16 0.25 0.34 0.43 0.52 0.61 0.7 0.79 0.88 0.97 0.06 0.15 0.24 0.33
    0.42 0.51 0.6 0.69 0.78 0.87]
 [0.08 0.17 0.26 0.35 0.44 0.53 0.62 0.71 0.8 0.89 0.98 0.07 0.16 0.25 0.34
    0.43 0.52 0.61 0.7 0.79 0.88]
 [0.09 0.18 0.27 0.36 0.45 0.54 0.63 0.72 0.81 0.9 0.99 0.08 0.17 0.26 0.35
    0.44 0.53 0.62 0.71 0.8 0.89]]

ac.conj().T
[[ 1.-2.j 2.-0.j 3.-3.j]
 [ 3.+1.j -1.-4.j -2.-1.j]
 [ 0.-1.j 5.+2.j 4.-0.j]]

A @ B
[[ 6. 9. 12.]
 [16. 20. 24.]
 [18. 21. 24.]]

A * B
[[ 2. 2. 0.]
 [ 4. 10. 6.]
 [ 0. 8. 18.]]


A / B
[[2. 0.5 0. ]
 [0.25 0.4 0.1667]
 [0. 0.125 0.2222]]


A**3
[[8. 1. 0.]
 [1. 8. 1.]
 [0. 1. 8.]]


(a0 > 0.5)
[[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 1 1 1 1]
 [1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1]
 [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 1 1 1]
 [1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1]]
```

```
np.nonzero(a0>0.5)
[[ 5 5 5 5 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8 8 9 9 9 9 9 9 9
    9 9 10 10 10 10 10 10 10 10 10 16 16 16 17 17 17 17 17 17 17 17 17 18 18
   18 18 18 18 18 18 18 19 19 19 19 19 19 19 19 19 20 20 20 20 20 20 20 20 20]
 [ 5 6 7 8 0 1 2 3 4 5 6 7 8 0 1 2 3 4 5 6 7 8 0 1 2 3 4 5 6 7 8 0 1 2 3 4 5 6
     7 8 0 1 2 3 4 5 6 7 8 6 7 8 0 1 2 3 4 5 6 7 8 0 1 2 3 4 5 6 7 8 0 1 2 3 4
     5 6 7 8 0 1 2 3 4 5 6 7 8]]

a0[:, np.nonzero(v>0.5)[0]]
[[0.06 0.07 0.08 0.09]
 [0.15 0.16 0.17 0.18]
 [0.24 0.25 0.26 0.27]
 [0.33 0.34 0.35 0.36]
 [0.42 0.43 0.44 0.45]
 [0.51 0.52 0.53 0.54]
 [0.6 0.61 0.62 0.63]
 [0.69 0.7 0.71 0.72]
 [0.78 0.79 0.8 0.81]
 [0.87 0.88 0.89 0.9 ]
 [0.96 0.97 0.98 0.99]
 [0.05 0.06 0.07 0.08]
 [0.14 0.15 0.16 0.17]
 [0.23 0.24 0.25 0.26]
 [0.32 0.33 0.34 0.35]
 [0.41 0.42 0.43 0.44]
 [0.5 0.51 0.52 0.53]
 [0.59 0.6 0.61 0.62]
 [0.68 0.69 0.7 0.71]
 [0.77 0.78 0.79 0.8 ]
 [0.86 0.87 0.88 0.89]]

a0[:, v.T > 0.5]
[[0.06 0.07 0.08 0.09]
```

```
 [0.15 0.16 0.17 0.18]
 [0.24 0.25 0.26 0.27]
 [0.33 0.34 0.35 0.36]
 [0.42 0.43 0.44 0.45]
 [0.51 0.52 0.53 0.54]
 [0.6 0.61 0.62 0.63]
 [0.69 0.7 0.71 0.72]
 [0.78 0.79 0.8 0.81]
 [0.87 0.88 0.89 0.9 ]
 [0.96 0.97 0.98 0.99]
 [0.05 0.06 0.07 0.08]
 [0.14 0.15 0.16 0.17]
 [0.23 0.24 0.25 0.26]
 [0.32 0.33 0.34 0.35]
 [0.41 0.42 0.43 0.44]
 [0.5 0.51 0.52 0.53]
 [0.59 0.6 0.61 0.62]
 [0.68 0.69 0.7 0.71]
 [0.77 0.78 0.79 0.8 ]
 [0.86 0.87 0.88 0.89]]


a[a<0.5]=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. 0.5 0.51 0.52 0.53 0.54]
 [0.55 0.56 0.57 0.58 0.59 0.6 0.61 0.62 0.63]
 [0.64 0.65 0.66 0.67 0.68 0.69 0.7 0.71 0.72]
 [0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.8 0.81]
 [0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89 0.9 ]
 [0.91 0.92 0.93 0.94 0.95 0.96 0.97 0.98 0.99]
 [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.  0.5 0.51 0.52 0.53]
 [0.54 0.55 0.56 0.57 0.58 0.59 0.6 0.61 0.62]
 [0.63 0.64 0.65 0.66 0.67 0.68 0.69 0.7 0.71]
 [0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.8 ]
 [0.81 0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89]]


a0 * (a0>0.5)
[[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.  0.51 0.52 0.53 0.54]
 [0.55 0.56 0.57 0.58 0.59 0.6 0.61 0.62 0.63]
 [0.64 0.65 0.66 0.67 0.68 0.69 0.7 0.71 0.72]
 [0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.8 0.81]
 [0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89 0.9 ]
 [0.91 0.92 0.93 0.94 0.95 0.96 0.97 0.98 0.99]
 [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.  0.51 0.52 0.53]
 [0.54 0.55 0.56 0.57 0.58 0.59 0.6 0.61 0.62]
 [0.63 0.64 0.65 0.66 0.67 0.68 0.69 0.7 0.71]
 [0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.8 ]
 [0.81 0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89]]


a[:]=3
```

```
[[3. 3. 3. 3. 3. 3. 3. 3. 3.]
 [3. 3. 3. 3. 3. 3. 3. 3. 3.]
 [3. 3. 3. 3. 3. 3. 3. 3. 3.]
 [3. 3. 3. 3. 3. 3. 3. 3. 3.]
 [3. 3. 3. 3. 3. 3. 3. 3. 3.]
 [3. 3. 3. 3. 3. 3. 3. 3. 3.]
 [3. 3. 3. 3. 3. 3. 3. 3. 3.]
 [3. 3. 3. 3. 3. 3. 3. 3. 3.]
 [3. 3. 3. 3. 3. 3. 3. 3. 3.]
 [3. 3. 3. 3. 3. 3. 3. 3. 3.]
 [3. 3. 3. 3. 3. 3. 3. 3. 3.]
 [3. 3. 3. 3. 3. 3. 3. 3. 3.]
 [3. 3. 3. 3. 3. 3. 3. 3. 3.]
 [3. 3. 3. 3. 3. 3. 3. 3. 3.]
 [3. 3. 3. 3. 3. 3. 3. 3. 3.]
 [3. 3. 3. 3. 3. 3. 3. 3. 3.]
 [3. 3. 3. 3. 3. 3. 3. 3. 3.]
 [3. 3. 3. 3. 3. 3. 3. 3. 3.]
 [3. 3. 3. 3. 3. 3. 3. 3. 3.]
 [3. 3. 3. 3. 3. 3. 3. 3. 3.]
 [3. 3. 3. 3. 3. 3. 3. 3. 3.]]

y_view[0] / y_ref[0]
[ 0.1 123.456]

x.flatten()
[ 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 123.456 0.11 0.12 0.13 0.14
   0.15 0.16 0.17 0.18 0.19 0.2 0.21 0.22 0.23 0.24 0.25 0.26 0.27 0.28 0.29
   0.3 0.31 0.32 0.33 0.34 0.35 0.36 0.37 0.38 0.39 0.4 0.41 0.42 0.43 0.44
   0.45 0.46 0.47 0.48 0.49 0.5 0.51 0.52 0.53 0.54 0.55 0.56 0.57 0.58 0.59
   0.6 0.61 0.62 0.63 0.64 0.65 0.66 0.67 0.68 0.69 0.7 0.71 0.72 0.73 0.74
   0.75 0.76 0.77 0.78 0.79 0.8 0.81 0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89
   0.9 0.91 0.92 0.93 0.94 0.95 0.96 0.97 0.98 0.99 0. 0.01 0.02 0.03 0.04
   0.05 0.06 0.07 0.08 0.09 0.1 0.11 0.12 0.13 0.14 0.15 0.16 0.17 0.18 0.19
```

```
    0.2 0.21 0.22 0.23 0.24 0.25 0.26 0.27 0.28 0.29 0.3 0.31 0.32 0.33 0.34
    0.35 0.36 0.37 0.38 0.39 0.4 0.41 0.42 0.43 0.44 0.45 0.46 0.47 0.48 0.49
    0.5 0.51 0.52 0.53 0.54 0.55 0.56 0.57 0.58 0.59 0.6 0.61 0.62 0.63 0.64
    0.65 0.66 0.67 0.68 0.69 0.7 0.71 0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79
    0.8 0.81 0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89 ]

x.flatten('F')
[ 0.01 123.456 0.19 0.28 0.37 0.46 0.55 0.64 0.73 0.82 0.91 0. 0.09 0.18 0.27
    0.36 0.45 0.54 0.63 0.72 0.81 0.02 0.11 0.2 0.29 0.38 0.47 0.56 0.65 0.74
    0.83 0.92 0.01 0.1 0.19 0.28 0.37 0.46 0.55 0.64 0.73 0.82 0.03 0.12 0.21
    0.3 0.39 0.48 0.57 0.66 0.75 0.84 0.93 0.02 0.11 0.2 0.29 0.38 0.47 0.56
    0.65 0.74 0.83 0.04 0.13 0.22 0.31 0.4 0.49 0.58 0.67 0.76 0.85 0.94 0.03
    0.12 0.21 0.3 0.39 0.48 0.57 0.66 0.75 0.84 0.05 0.14 0.23 0.32 0.41 0.5
    0.59 0.68 0.77 0.86 0.95 0.04 0.13 0.22 0.31 0.4 0.49 0.58 0.67 0.76 0.85
    0.06 0.15 0.24 0.33 0.42 0.51 0.6 0.69 0.78 0.87 0.96 0.05 0.14 0.23 0.32
    0.41 0.5 0.59 0.68 0.77 0.86 0.07 0.16 0.25 0.34 0.43 0.52 0.61 0.7 0.79
    0.88 0.97 0.06 0.15 0.24 0.33 0.42 0.51 0.6 0.69 0.78 0.87 0.08 0.17 0.26
    0.35 0.44 0.53 0.62 0.71 0.8 0.89 0.98 0.07 0.16 0.25 0.34 0.43 0.52 0.61
    0.7 0.79 0.88 0.09 0.18 0.27 0.36 0.45 0.54 0.63 0.72 0.81 0.9 0.99 0.08
    0.17 0.26 0.35 0.44 0.53 0.62 0.71 0.8 0.89 ]

1:10
[ 1. 2. 3. 4. 5. 6. 7. 8. 9. 10.]

0:9
[0. 1. 2. 3. 4. 5. 6. 7. 8. 9.]

[1:10]'
[[␣1.]
␣[␣2.]
␣[␣3.]
␣[␣4.]
␣[␣5.]
␣[␣6.]
```

```
␣[␣7.]
␣[␣8.]
␣[␣9.]
␣[10.]]

np.r_[1:10:10j]
[␣1.␣␣2.␣␣3.␣␣4.␣␣5.␣␣6.␣␣7.␣␣8.␣␣9.␣10.]

zeros(3,4)␣/␣ones(3,4)␣/␣eye(3)
[[0.␣0.␣0.␣0.]
␣[0.␣0.␣0.␣0.]
␣[0.␣0.␣0.␣0.]]
[[1.␣1.␣1.␣1.]
␣[1.␣1.␣1.␣1.]
␣[1.␣1.␣1.␣1.]]
[[1.␣0.␣0.]
␣[0.␣1.␣0.]
␣[0.␣0.␣1.]]

zeros(3,4,5)
[[[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.]]]
```

```
diag(a)␣/␣diag(v,0)
[1.␣5.␣9.]
[[10.␣␣0.␣␣0.]
␣[␣0.␣20.␣␣0.]
␣[␣0.␣␣0.␣30.]]

rng(42)␣;␣rand(3,4)
[[0.774␣␣0.4389␣0.8586␣0.6974]
␣[0.0942␣0.9756␣0.7611␣0.7861]
␣[0.1281␣0.4504␣0.3708␣0.9268]]

linspace(1,3,4)
[1.␣␣␣␣␣1.6667␣2.3333␣3.␣␣␣␣]

meshgrid␣lists
X
␣[[1␣2␣4]
␣[1␣2␣4]
␣[1␣2␣4]]
Y
␣[[2␣2␣2]
␣[4␣4␣4]
␣[5␣5␣5]]

mgrid
[[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.]]

ogrid␣shapes
(9,␣1)␣(1,␣6)

ix_␣shapes
(9,␣1)␣(1,␣6)

repmat␣->␣tile
[[1.␣2.␣1.␣2.␣1.␣2.]
␣[3.␣4.␣3.␣4.␣3.␣4.]
␣[1.␣2.␣1.␣2.␣1.␣2.]
␣[3.␣4.␣3.␣4.␣3.␣4.]]

[a␣b]␣/␣[a;␣b]␣/␣column_stack␣/␣c_␣/␣r_
[[1.␣2.␣5.␣6.]
␣[3.␣4.␣7.␣8.]]
[[1.␣2.]
␣[3.␣4.]
␣[5.␣6.]
␣[7.␣8.]]
[[1␣4]
␣[2␣5]
␣[3␣6]]
[[1␣4]
```

```
␣[2␣5]
␣[3␣6]]
[[1␣2]
␣[3␣4]]

np.concatenate␣row-wise
[[1.␣2.␣3.]
␣[4.␣5.␣6.]]

max/max/nanmax/column/row/maximum
a.max()/nanmax:␣nan␣6.0
max␣columns:␣[␣4.␣nan␣␣6.]
max␣rows:␣[nan␣␣6.]
maximum:␣[2␣5␣3]

norm(v)
13.0

logical_and␣/␣logical_or
[False␣False␣␣True]
[␣True␣False␣␣True]

bitand/␣bitor␣via␣&␣and␣|
[0␣3␣1]
[3␣3␣5]

solve(A,b)
[0.5␣0.␣␣1.5]

inv(A)
[[␣0.75␣-0.5␣␣␣0.25]
␣[-0.5␣␣␣1.␣␣␣-0.5␣]
␣[␣0.25␣-0.5␣␣␣0.75]]
```

```
pinv(A)
[[␣0.75␣-0.5␣␣␣0.25]
␣[-0.5␣␣␣1.␣␣␣-0.5␣]
␣[␣0.25␣-0.5␣␣␣0.75]]


det(A)
4.0


trace(A)
6.0


rank(A)
3


norms␣[2,␣fro,␣1,␣inf]
[3.4142␣4.␣␣␣␣␣4.␣␣␣␣␣4.␣␣␣␣]


eig(A):␣w
[3.4142+0.j␣2.␣␣␣␣+0.j␣0.5858+0.j]


eig(A):␣V
[[-0.5␣␣␣␣␣0.7071␣␣0.5␣␣␣]
␣[-0.7071␣␣0.␣␣␣␣␣-0.7071]
␣[-0.5␣␣␣␣-0.7071␣␣0.5␣␣␣]]


generalized␣eig(A,B)
eigvals:␣[0.3694+0.j␣0.6667+0.j␣0.7735+0.j]
eigvecs:
␣[[-0.5␣␣␣␣␣0.7071␣-0.5␣␣␣]
␣[␣0.7071␣-0.␣␣␣␣␣-0.7071]
␣[-0.5␣␣␣␣-0.7071␣-0.5␣␣␣]]


eigs␣(sparse,␣k=1)
eigs␣vals:␣[3.4142+0.j]
```

```
eigs␣vecs:
␣[[0.5␣␣␣+0.j]
␣[0.7071+0.j]
␣[0.5␣␣␣+0.j]]

svd
s
␣[3.4142␣2.␣␣␣␣␣0.5858]
U
␣[[-0.5␣␣␣␣␣0.7071␣␣0.5␣␣␣]
␣[-0.7071␣-0.␣␣␣␣␣-0.7071]
␣[-0.5␣␣␣␣-0.7071␣␣0.5␣␣␣]]
Vh
␣[[-0.5␣␣␣␣-0.7071␣-0.5␣␣␣]
␣[␣0.7071␣␣0.␣␣␣␣␣-0.7071]
␣[␣0.5␣␣␣␣-0.7071␣␣0.5␣␣␣]]

qr␣R
[[-2.2361␣-1.7889␣-0.4472]
␣[␣0.␣␣␣␣␣-1.6733␣-1.9124]
␣[␣0.␣␣␣␣␣␣0.␣␣␣␣␣␣1.069␣]]

chol(A^T␣A␣+␣I)
[[2.4495␣1.633␣␣0.4082]
␣[0.␣␣␣␣␣2.0817␣1.6013]
␣[0.␣␣␣␣␣0.␣␣␣␣␣1.8081]]

LU:␣P,L,U␣and␣check
P=
␣[[0.␣1.␣0.]
␣[0.␣0.␣1.]
␣[1.␣0.␣0.]]
L=
␣[[1.␣␣␣␣␣0.␣␣␣␣␣0.␣␣␣␣]
```

```
␣[0.1429␣1.␣␣␣␣␣0.␣␣␣␣]
␣[0.5714␣0.5␣␣␣␣1.␣␣␣␣]]
U=
␣[[␣7.␣␣␣␣␣␣8.␣␣␣␣␣␣9.␣␣␣␣]
␣[␣0.␣␣␣␣␣␣0.8571␣␣1.7143]
␣[␣0.␣␣␣␣␣␣0.␣␣␣␣␣-0.␣␣␣␣]]
P@L@U=
␣[[1.␣2.␣3.]
␣[4.␣5.␣6.]
␣[7.␣8.␣9.]]

a\b␣vs␣b/a
x␣from␣a\b:␣[0.5␣0.␣␣1.5]
X␣from␣b/a␣(solve␣A.T␣X.T␣=␣C.T):
␣[[␣1.25␣-1.5␣␣␣1.75]
␣[␣1.5␣␣␣0.␣␣␣␣2.5␣]]
Check␣X␣@␣A:
␣[[␣1.␣-0.␣␣2.]
␣[␣3.␣␣4.␣␣5.]]

cg
cg␣solution:␣[0.0909␣0.6364]␣info:␣0

fft␣/␣ifft
[10.+0.j␣-2.+2.j␣-2.+0.j␣-2.-2.j]
[1.+0.j␣2.+0.j␣3.+0.j␣4.+0.j]

sort␣columns␣/␣sort␣rows␣/␣sortrows␣by␣first␣column
sort␣columns:
␣[[3.␣2.␣1.]
␣[6.␣5.␣4.]
␣[9.␣8.␣7.]]
sort␣rows:
␣[[1.␣2.␣3.]
```

```
␣[4.␣5.␣6.]
␣[7.␣8.␣9.]]
rows␣sorted␣by␣col0:
␣[[3.␣2.␣1.]
␣[6.␣5.␣4.]
␣[9.␣8.␣7.]]

in-place␣sort␣a.sort(axis=0)
[[3.␣2.␣1.]
␣[6.␣5.␣4.]
␣[9.␣8.␣7.]]

lstsq␣(x␣=␣Z\y)
[3.5␣1.4]

decimate␣~␣resample
[␣6.5␣␣␣␣␣1.2116␣␣6.7143␣␣7.7534␣12.2466␣13.2857␣18.7884]

unique␣/␣squeeze
[1␣2␣3␣4]
before␣squeeze:␣(1,␣1,␣3)␣after:␣(3,)
```

code

```
np.set_printoptions(threshold=np.inf, linewidth=10**6, precision=4, suppress=True)

print(RANGES 1:10 / 0:9 / column vector [1:10]')
print(np.arange(1., 11.)); print(np.arange(10.)); print(np.arange(1., 11.)[:, np.
                                        newaxis]); print()

print(zeros / ones / eye)
print(np.zeros((3,4))); print(np.ones((3,4))); print(np.eye(3)); print()

print(diag(a) and diag(v,0))
```

```
v_demo =np.array([10., 20., 30.])
A_demo =np.array([[1.,2.,3.],[4.,5.,6.],[7.,8.,9.]])
print(np.diag(A_demo)); print(np.diag(v_demo, 0)); print()

print(rng(42) ; rand(3,4) (NumPy default_rng))
from numpy.random import default_rng
rng =default_rng(42)
print(rng.random((3,4))); print()

print(linspace(1,3,4))
print(np.linspace(1,3,4)); print()

print(meshgrid / mgrid / ogrid / ix_)
X, Y =np.meshgrid([1,2,4],[2,4,5])
print(meshgrid lists -> X\n, X, \nY\n, Y); print()
print(mgrid)
print(np.mgrid[0:9., 0:6.][0]); print(np.mgrid[0:9., 0:6.][1]); print()
print(ogrid best for eval)
ogx, ogy =np.ogrid[0:9., 0:6.]
print(ogx.shape, ogy.shape); print()
print(ix_ best for eval on vectors)
print(np.ix_(np.r_[0:9.], np.r_[0:6.])[0].shape, np.ix_(np.r_[0:9.], np.r_[0:6.])[1].
                                            shape); print()

print(repmat(a,m,n) -> tile)
print(np.tile(np.array([[1.,2.],[3.,4.]]), (2,3))); print()

print(column/row concatenation: hstack / vstack / column_stack / c_ / r_)
left =np.array([[1.,2.],[3.,4.]])
right =np.array([[5.,6.],[7.,8.]])
print(np.hstack((left, right)))
print(np.vstack((left, right)))
print(np.column_stack((np.array([1,2,3]), np.array([4,5,6]))))
print(np.c_[np.array([1,2,3]), np.array([4,5,6])])
print(np.r_[np.array([[1,2]]), np.array([[3,4]])]); print()

print(max(max(a)) / max(a) by column / max(a,[],2) by row / maximum(a,b))
M =np.array([[1., np.nan, 3.],[4.,5.,6.]])
print(a.max() / nanmax:, M.max(), np.nanmax(M))
print(max by columns:, M.max(0))
print(max by rows:, M.max(1))
print(element-wise maximum with B:, np.maximum(np.array([1,5,2]), np.array([2,4,3])));
                                            print()

print(logical_and / logical_or (elementwise))
p =np.array([True, False, True])
q =np.array([False, False, True])
print(np.logical_and(p,q)); print(np.logical_or(p,q)); print()

print(bitwise AND/OR (&, |) on integers)
x_bits =np.array([1,3,5], dtype=int)
y_bits =np.array([2,3,1], dtype=int)
print(x_bits & y_bits); print(x_bits | y_bits); print()

print(lstsq (x = Z\\y))
Z =np.array([[1.,1.],[1.,2.],[1.,3.],[1.,4.]])
y =np.array([6.,5.,7.,10.])
coef, *_ =la.lstsq(Z, y)
print(coef); print()

print(generalized eig (D,V) = eig(A,B))
B_spd =np.array([[3.,1.,0.],[1.,3.,1.],[0.,1.,3.]])
D_gen, V_gen =la.eig(A, B_spd)
print(eigvals:, D_gen); print(eigvecs:\n, V_gen); print()

print(eigs (sparse, k=3 largest) -- may return complex dtype)

S_sparse =csr_matrix(np.array([[2.,1.,0.],[1.,2.,1.],[0.,1.,2.]]))
vals, vecs =eigs(S_sparse, k=1)
print(eigs vals:, vals); print(eigs vecs:\n, vecs); print()

print(cg (conjugate gradients) on SPD)
A_spd =np.array([[4.,1.],[1.,3.]])
b_spd =np.array([1.,2.])
x_cg, info =cg(csr_matrix(A_spd), b_spd, maxiter=1000, tol=1e-12)
print(cg solution:, x_cg, info:, info); print()

print(FFT / IFFT)
```

```
sig =np.array([1., 2., 3., 4.])
F =np.fft.fft(sig)
fi =np.fft.ifft(F)
print(F); print(fi); print()

print(sort each column / sort each row / sortrows by first column)
S =np.array([[3.,2.,1.],[6.,5.,4.],[9.,8.,7.]])
print(sort columns:\n, np.sort(S, axis=0))
print(sort rows:\n, np.sort(S, axis=1))
I =np.argsort(S[:,0])
print(rows sorted by col0:\n, S[I,:]); print()

print(unique / squeeze)
u =np.array([1,2,2,3,3,3,4])
print(np.unique(u))
sq =np.array([[[[1,2,3]]]])
print(before squeeze:, sq.shape, after:, sq.squeeze().shape); print()

print(decimate(x,q) ~ signal.resample(x, ceil(len(x)/q)))
xsig =np.arange(20.)
q =3
down =signal.resample(xsig, int(np.ceil(len(xsig)/q)))
print(down); print()

print(a(:, find(v>0.5)) and a(:, v.T > 0.5) forms)
v_col =v.reshape(-1,1) # column vector (9,1)
print(a0[:, np.nonzero(v >0.5)[0]])
print(a0[:, (v_col.T >0.5).ravel()]); print()

print(LU factorization: P,L,U (A == P@L@U))
P,L,U =la.lu(A_demo)
print(P=\n, P, \nL=\n, L, \nU=\n, U)
print(Check P@L@U:\n, P@L@U); print()

print(zeros(3,4,5) -> 3D zeros)
print(np.zeros((3, 4, 5))); print()

print(Right division b/a)
C =np.array([[1., 0., 2.],
             [3., 4., 5.]])
X_right =la.solve(A.T, C.T).T
print(X =\n, X_right)
print(Check X @ A =\n, X_right @ A); print()

print(Vector norm)
v_demo2 =np.array([3., 4., 12.])
print(np.linalg.norm(v_demo2)); print()

print(Row concatenation with np.concatenate((a,b)))
a_row =np.array([[1., 2., 3.]])
b_row =np.array([[4., 5., 6.]])
print(np.concatenate((a_row, b_row))); print()

print(In-place sort a.sort(axis=0))
S2 =np.array([[3.,2.,1.],[6.,5.,4.],[9.,8.,7.]])
S2.sort(axis=0)
print(S2); print()

print(Range via np.r_[1:10:10j])
print(np.r_[1:10:10j]); print()
```

3 Task 3

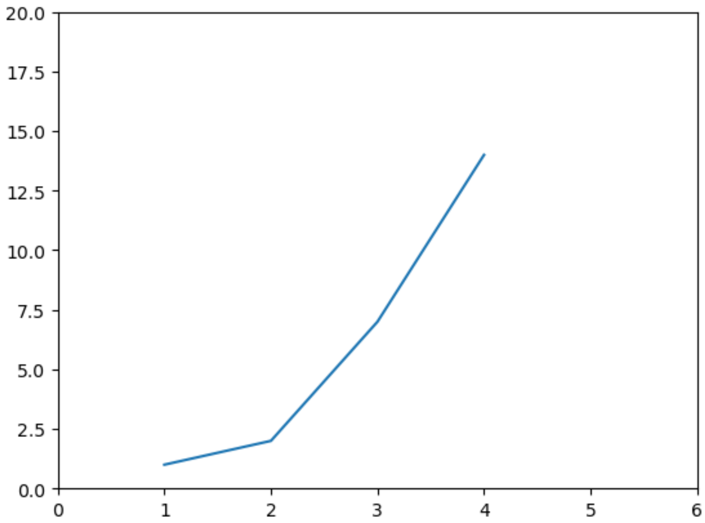

Figure 1: Line plot

code

```
import matplotlib.pyplot as plt

plt.plot([1,2,3,4], [1,2,7,14])
plt.axis([0, 6, 0, 20])
plt.show()
```

4 Task 4

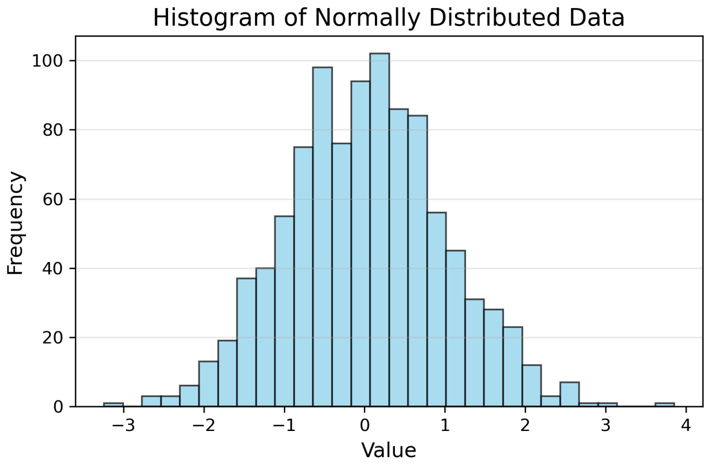

Figure 2: Line plot

```
import numpy as np
import matplotlib.pyplot as plt

np.random.seed(42)
data =np.random.randn(1000)

plt.figure(figsize=(6,4), dpi=300)
plt.hist(data, bins=30, color=skyblue, edgecolor=black, alpha=0.7)
plt.title(Histogram of Normally Distributed Data, fontsize=14)
plt.xlabel(Value, fontsize=12)
plt.ylabel(Frequency, fontsize=12)
plt.grid(axis=y, alpha=0.3)
plt.tight_layout()
plt.savefig(task4_histogram.png, dpi=300)
plt.show()
```

5 Task 5

Github: qih33333

Link: https://github.com/qih33333

6 Task 6

Link: https://github.com/qih33333/An-Introduction-to-Deep-Learning