gradient_descent

September 21, 2022

1 Gradient descent

- By Tao Huang
- Date: Sep 21, 2022

[61]:

```
# Dependencies

!pip install numpy
!pip install matplotlib
```

```
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Requirement already satisfied: numpy in
/Users/taou/opt/anaconda3/envs/spinningup/lib/python3.6/site-packages (1.18.5)
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Requirement already satisfied: matplotlib in
/Users/taou/opt/anaconda3/envs/spinningup/lib/python3.6/site-packages (3.1.1)
Requirement already satisfied: numpy>=1.11 in
/Users/taou/opt/anaconda3/envs/spinningup/lib/python3.6/site-packages (from
matplotlib) (1.18.5)
Requirement already satisfied: cycler>=0.10 in
/Users/taou/opt/anaconda3/envs/spinningup/lib/python3.6/site-packages (from
matplotlib) (0.10.0)
Requirement already satisfied: python-dateutil>=2.1 in
/Users/taou/opt/anaconda3/envs/spinningup/lib/python3.6/site-packages (from
matplotlib) (2.8.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
/Users/taou/opt/anaconda3/envs/spinningup/lib/python3.6/site-packages (from
matplotlib) (2.4.7)
Requirement already satisfied: kiwisolver>=1.0.1 in
/Users/taou/opt/anaconda3/envs/spinningup/lib/python3.6/site-packages (from
matplotlib) (1.2.0)
Requirement already satisfied: six in
/Users/taou/opt/anaconda3/envs/spinningup/lib/python3.6/site-packages (from
cycler>=0.10->matplotlib) (1.15.0)
```

[62]:

```
import numpy as np
import time
```

1.1 Basic implementation

Objective function $f(x) = \|x\|_2^2$ is a mapping: $\mathbb{R}^n \to \mathbb{R}$

Gradient $\nabla f(x) = 2x$ is a mapping: $\mathbb{R}^n \to \mathbb{R}^n$

Convergence criterion: distance between the values of the function in two consecutive iterations is less than a threshold, i.e, $\|f(\theta_{t+1}) - f(\theta_t)\|_2 < \epsilon$.

[63]:

```
# Suppose x is a 5-dimensional real vector
n = 5
# Objective function
f = lambda x: np.sum(x * x)
# Gradient
df = lambda x: 2 * x
```

[64]:

```
# Convergence threshold
epsilon = 1e-5
# Learning rate
alpha = 0.1
```

[65]:

```
# Initialization
x0 = np.random.randn(n)
x = x0
x_last = np.inf * np.ones(n)
print('Initialize: x =', x)
print('')

# Start gradient descent
iters = 0

while abs(f(x_last) - f(x)) >= epsilon:
    x_last = x
    x = x - alpha * df(x)

    iters += 1

    print('==> Iter %s, x=%s \n f(x)=%s, |f(x) - f(x_last)|=%s' % (iters, x,␣
 ↪f(x), abs(f(x_last) - f(x))))
    time.sleep(0.3)

print('Converged')
```

```
Initialize: x = [1.27900756 0.16849124 0.03427397 0.47934266 0.73817261]

==> Iter 1, x=[1.02320605 0.13479299 0.02741918 0.38347413 0.59053809]
 f(x)=1.5616592175447348, |f(x) - f(x_last)|=0.8784333098689134
==> Iter 2, x=[0.81856484 0.1078344  0.02193534 0.3067793  0.47243047]
```

```
 f(x)=0.9994618992286304, |f(x) - f(x_last)|=0.5621973183161044
==> Iter 3, x=[0.65485187 0.08626752 0.01754827 0.24542344 0.37794438]
 f(x)=0.6396556155063234, |f(x) - f(x_last)|=0.35980628372230694
==> Iter 4, x=[0.5238815  0.06901401 0.01403862 0.19633875 0.3023555 ]
 f(x)=0.40937959392404705, |f(x) - f(x_last)|=0.2302760215822764
==> Iter 5, x=[0.4191052  0.05521121 0.0112309  0.157071   0.2418844 ]
 f(x)=0.26200294011139014, |f(x) - f(x_last)|=0.1473766538126569
==> Iter 6, x=[0.33528416 0.04416897 0.00898472 0.1256568  0.19350752]
 f(x)=0.1676818816712897, |f(x) - f(x_last)|=0.09432105844010044
==> Iter 7, x=[0.26822733 0.03533517 0.00718777 0.10052544 0.15480602]
 f(x)=0.10731640426962538, |f(x) - f(x_last)|=0.060365477401664314
==> Iter 8, x=[0.21458186 0.02826814 0.00575022 0.08042035 0.12384481]
 f(x)=0.06868249873256026, |f(x) - f(x_last)|=0.038633905537065125
==> Iter 9, x=[0.17166549 0.02261451 0.00460017 0.06433628 0.09907585]
 f(x)=0.04395679918883856, |f(x) - f(x_last)|=0.024725699543721696
==> Iter 10, x=[0.13733239 0.01809161 0.00368014 0.05146903 0.07926068]
 f(x)=0.028132351480856684, |f(x) - f(x_last)|=0.015824447707981876
==> Iter 11, x=[0.10986591 0.01447329 0.00294411 0.04117522 0.06340854]
 f(x)=0.018004704947748276, |f(x) - f(x_last)|=0.010127646533108409
==> Iter 12, x=[0.08789273 0.01157863 0.00235529 0.03294018 0.05072684]
 f(x)=0.011523011166558897, |f(x) - f(x_last)|=0.006481693781189379
==> Iter 13, x=[0.07031418 0.0092629  0.00188423 0.02635214 0.04058147]
 f(x)=0.007374727146597693, |f(x) - f(x_last)|=0.004148284019961203
==> Iter 14, x=[0.05625135 0.00741032 0.00150739 0.02108171 0.03246517]
 f(x)=0.004719825373822546, |f(x) - f(x_last)|=0.002654901772775169
==> Iter 15, x=[0.04500108 0.00592826 0.00120591 0.01686537 0.02597214]
 f(x)=0.0030206882392464158, |f(x) - f(x_last)|=0.0016991371345761088
==> Iter 16, x=[0.03600086 0.00474261 0.00096473 0.0134923  0.02077771]
 f(x)=0.0019332404731177064, |f(x) - f(x_last)|=0.0010874477661287094
==> Iter 17, x=[0.02880069 0.00379409 0.00077178 0.01079384 0.01662217]
 f(x)=0.001237273902795332, |f(x) - f(x_last)|=0.0006959665703223743
==> Iter 18, x=[0.02304055 0.00303527 0.00061743 0.00863507 0.01329774]
 f(x)=0.0007918552977890125, |f(x) - f(x_last)|=0.0004454186050063196
==> Iter 19, x=[0.01843244 0.00242821 0.00049394 0.00690806 0.01063819]
 f(x)=0.000506787390584968, |f(x) - f(x_last)|=0.0002850679072040445
==> Iter 20, x=[0.01474595 0.00194257 0.00039515 0.00552644 0.00851055]
 f(x)=0.0003243439299743795, |f(x) - f(x_last)|=0.0001824346061058848
==> Iter 21, x=[0.01179676 0.00155406 0.00031612 0.00442116 0.00680844]
 f(x)=0.00020758011518360288, |f(x) - f(x_last)|=0.00011676381479077663
==> Iter 22, x=[0.00943741 0.00124325 0.0002529  0.00353692 0.00544675]
 f(x)=0.00013285127371750585, |f(x) - f(x_last)|=7.472884146609703e-05
==> Iter 23, x=[0.00754993 0.0009946  0.00020232 0.00282954 0.0043574 ]
 f(x)=8.502481517920374e-05, |f(x) - f(x_last)|=4.782645853830211e-05
==> Iter 24, x=[0.00603994 0.00079568 0.00016185 0.00226363 0.00348592]
 f(x)=5.441588171469039e-05, |f(x) - f(x_last)|=3.060893346451335e-05
==> Iter 25, x=[0.00483195 0.00063654 0.00012948 0.00181091 0.00278874]
 f(x)=3.482616429740185e-05, |f(x) - f(x_last)|=1.9589717417288538e-05
==> Iter 26, x=[0.00386556 0.00050923 0.00010359 0.00144872 0.00223099]
```

```
 f(x)=2.2288745150337183e-05, |f(x) - f(x_last)|=1.2537419147064667e-05
==> Iter 27, x=[3.09245052e-03 4.07386829e-04 8.28693819e-05 1.15897944e-03
 1.78479186e-03]
 f(x)=1.4264796896215799e-05, |f(x) - f(x_last)|=8.023948254121384e-06
Converged
```

[66]:

```
# Optimal solution
x
```

[66]:

```
array([3.09245052e-03, 4.07386829e-04, 8.28693819e-05, 1.15897944e-03,
       1.78479186e-03])
```

[14]:

```
# Minimum
f(x)
```

[14]:

```
1.3847683027448377e-05
```

1.2 Encapsulation of gradient descent

[67]:

```
def grad_descent(obj_fn, grad_fn, alpha, is_converge, init_x, max_iters = 1000,␣
↪logging=False):
    x = init_x
    x_last = np.inf * np.ones_like(init_x)

    iters = 0

    while not is_converge(obj_fn(x_last), obj_fn(x)):
        x_last = x
        x = x - alpha * grad_fn(x)

        iters += 1

        if logging:
            print('==> Iter %s, x=%s \n f(x)=%s, is converged=%s' % (iters, x,␣
↪f(x), is_converge(obj_fn(x_last), obj_fn(x))))

        if iters >= max_iters:
            print('Not converged in %s steps' % iters)
            return x

    print('Converged in %s steps' % iters)

    # Return optimal solution
    return x
```

[68]:

```
converge_criterion = lambda last, curr: abs(last - curr) < epsilon
```

```
grad_descent(f, df, 0.1, converge_criterion, x0)
```

```
Converged in 27 steps
```

```
[68]: array([3.09245052e-03, 4.07386829e-04, 8.28693819e-05, 1.15897944e-03,
              1.78479186e-03])
```

```
[ ]:
```

1.3 Verify the example

```
[69]: converge_criterion2 = lambda last, curr: abs(curr) < 0.01

      f_quiz = lambda theta: theta ** 2
      df_quiz = lambda theta: 2 * theta

      grad_descent(f_quiz, df_quiz, 0.01, converge_criterion2, np.array([10]),␣
       ↪logging=True, max_iters=1000)
      # Try: alpha = 1.0, 0.1, 0.01
```

```
==> Iter 1, x=[9.8]
 f(x)=96.04000000000002, is converged=[False]
==> Iter 2, x=[9.604]
 f(x)=92.23681600000002, is converged=[False]
==> Iter 3, x=[9.41192]
 f(x)=88.5842380864, is converged=[False]
==> Iter 4, x=[9.2236816]
 f(x)=85.07630225817857, is converged=[False]
==> Iter 5, x=[9.03920797]
 f(x)=81.70728068875471, is converged=[False]
==> Iter 6, x=[8.85842381]
 f(x)=78.47167237348003, is converged=[False]
==> Iter 7, x=[8.68125533]
 f(x)=75.36419414749021, is converged=[False]
==> Iter 8, x=[8.50763023]
 f(x)=72.37977205924958, is converged=[False]
==> Iter 9, x=[8.33747762]
 f(x)=69.5135330857033, is converged=[False]
==> Iter 10, x=[8.17072807]
 f(x)=66.76079717550945, is converged=[False]
==> Iter 11, x=[8.00731351]
 f(x)=64.11706960735928, is converged=[False]
==> Iter 12, x=[7.84716724]
 f(x)=61.578033650907855, is converged=[False]
==> Iter 13, x=[7.69022389]
 f(x)=59.139543518331905, is converged=[False]
==> Iter 14, x=[7.53641941]
```

```
 f(x)=56.79761759500595, is converged=[False]
==> Iter 15, x=[7.38569103]
 f(x)=54.54843193824372, is converged=[False]
==> Iter 16, x=[7.23797721]
 f(x)=52.38831403348927, is converged=[False]
==> Iter 17, x=[7.09321766]
 f(x)=50.3137367977631, is converged=[False]
==> Iter 18, x=[6.95135331]
 f(x)=48.32131282057168, is converged=[False]
==> Iter 19, x=[6.81232624]
 f(x)=46.40778883287704, is converged=[False]
==> Iter 20, x=[6.67607972]
 f(x)=44.57004039509511, is converged=[False]
==> Iter 21, x=[6.54255812]
 f(x)=42.80506679544934, is converged=[False]
==> Iter 22, x=[6.41170696]
 f(x)=41.10998615034955, is converged=[False]
==> Iter 23, x=[6.28347282]
 f(x)=39.482030698795704, is converged=[False]
==> Iter 24, x=[6.15780337]
 f(x)=37.918542283123394, is converged=[False]
==> Iter 25, x=[6.0346473]
 f(x)=36.416968008711706, is converged=[False]
==> Iter 26, x=[5.91395435]
 f(x)=34.97485607556672, is converged=[False]
==> Iter 27, x=[5.79567526]
 f(x)=33.589851774974285, is converged=[False]
==> Iter 28, x=[5.67976176]
 f(x)=32.259693644685306, is converged=[False]
==> Iter 29, x=[5.56616652]
 f(x)=30.98220977635577, is converged=[False]
==> Iter 30, x=[5.45484319]
 f(x)=29.755314269212086, is converged=[False]
==> Iter 31, x=[5.34574633]
 f(x)=28.577003824151284, is converged=[False]
==> Iter 32, x=[5.2388314]
 f(x)=27.445354472714897, is converged=[False]
==> Iter 33, x=[5.13405478]
 f(x)=26.35851843559539, is converged=[False]
==> Iter 34, x=[5.03137368]
 f(x)=25.314721105545807, is converged=[False]
==> Iter 35, x=[4.93074621]
 f(x)=24.31225814976619, is converged=[False]
==> Iter 36, x=[4.83213128]
 f(x)=23.349492727035454, is converged=[False]
==> Iter 37, x=[4.73548866]
 f(x)=22.424852815044847, is converged=[False]
==> Iter 38, x=[4.64077888]
```

```
 f(x)=21.536828643569073, is converged=[False]
==> Iter 39, x=[4.54796331]
 f(x)=20.683970229283737, is converged=[False]
==> Iter 40, x=[4.45700404]
 f(x)=19.864885008204098, is converged=[False]
==> Iter 41, x=[4.36786396]
 f(x)=19.078235561879215, is converged=[False]
==> Iter 42, x=[4.28050668]
 f(x)=18.322737433628795, is converged=[False]
==> Iter 43, x=[4.19489655]
 f(x)=17.597157031257098, is converged=[False]
==> Iter 44, x=[4.11099862]
 f(x)=16.900309612819317, is converged=[False]
==> Iter 45, x=[4.02877864]
 f(x)=16.231057352151673, is converged=[False]
==> Iter 46, x=[3.94820307]
 f(x)=15.588307481006469, is converged=[False]
==> Iter 47, x=[3.86923901]
 f(x)=14.971010504758612, is converged=[False]
==> Iter 48, x=[3.79185423]
 f(x)=14.378158488770172, is converged=[False]
==> Iter 49, x=[3.71601714]
 f(x)=13.808783412614872, is converged=[False]
==> Iter 50, x=[3.6416968]
 f(x)=13.261955589475322, is converged=[False]
==> Iter 51, x=[3.56886286]
 f(x)=12.7367821481321, is converged=[False]
==> Iter 52, x=[3.49748561]
 f(x)=12.232405575066071, is converged=[False]
==> Iter 53, x=[3.4275359]
 f(x)=11.748002314293455, is converged=[False]
==> Iter 54, x=[3.35898518]
 f(x)=11.282781422647433, is converged=[False]
==> Iter 55, x=[3.29180547]
 f(x)=10.835983278310596, is converged=[False]
==> Iter 56, x=[3.22596936]
 f(x)=10.406878340489499, is converged=[False]
==> Iter 57, x=[3.16144998]
 f(x)=9.994765958206115, is converged=[False]
==> Iter 58, x=[3.09822098]
 f(x)=9.598973226261153, is converged=[False]
==> Iter 59, x=[3.03625656]
 f(x)=9.218853886501211, is converged=[False]
==> Iter 60, x=[2.97553143]
 f(x)=8.853787272595765, is converged=[False]
==> Iter 61, x=[2.9160208]
 f(x)=8.50317729660097, is converged=[False]
==> Iter 62, x=[2.85770038]
```

```
 f(x)=8.166451475655574, is converged=[False]
==> Iter 63, x=[2.80054637]
 f(x)=7.843059997219613, is converged=[False]
==> Iter 64, x=[2.74453545]
 f(x)=7.532474821329715, is converged=[False]
==> Iter 65, x=[2.68964474]
 f(x)=7.234188818405059, is converged=[False]
==> Iter 66, x=[2.63585184]
 f(x)=6.947714941196217, is converged=[False]
==> Iter 67, x=[2.58313481]
 f(x)=6.672585429524847, is converged=[False]
==> Iter 68, x=[2.53147211]
 f(x)=6.408351046515662, is converged=[False]
==> Iter 69, x=[2.48084267]
 f(x)=6.154580345073643, is converged=[False]
==> Iter 70, x=[2.43122581]
 f(x)=5.910858963408725, is converged=[False]
==> Iter 71, x=[2.3826013]
 f(x)=5.676788948457741, is converged=[False]
==> Iter 72, x=[2.33494927]
 f(x)=5.451988106098814, is converged=[False]
==> Iter 73, x=[2.28825029]
 f(x)=5.236089377097301, is converged=[False]
==> Iter 74, x=[2.24248528]
 f(x)=5.0287402377642465, is converged=[False]
==> Iter 75, x=[2.19763558]
 f(x)=4.829602124348783, is converged=[False]
==> Iter 76, x=[2.15368286]
 f(x)=4.63834988022457, is converged=[False]
==> Iter 77, x=[2.11060921]
 f(x)=4.454671224967678, is converged=[False]
==> Iter 78, x=[2.06839702]
 f(x)=4.278266244458958, is converged=[False]
==> Iter 79, x=[2.02702908]
 f(x)=4.108846901178383, is converged=[False]
==> Iter 80, x=[1.9864885]
 f(x)=3.9461365638917187, is converged=[False]
==> Iter 81, x=[1.94675873]
 f(x)=3.789869555961607, is converged=[False]
==> Iter 82, x=[1.90782356]
 f(x)=3.6397907215455274, is converged=[False]
==> Iter 83, x=[1.86966709]
 f(x)=3.4956550089723244, is converged=[False]
==> Iter 84, x=[1.83227374]
 f(x)=3.35722707061702, is converged=[False]
==> Iter 85, x=[1.79562827]
 f(x)=3.2242808786205863, is converged=[False]
==> Iter 86, x=[1.7597157]
```

```
 f(x)=3.096599355827211, is converged=[False]
==> Iter 87, x=[1.72452139]
 f(x)=2.9739740213364536, is converged=[False]
==> Iter 88, x=[1.69003096]
 f(x)=2.8562046500915304, is converged=[False]
==> Iter 89, x=[1.65623034]
 f(x)=2.743098945947906, is converged=[False]
==> Iter 90, x=[1.62310574]
 f(x)=2.6344722276883683, is converged=[False]
==> Iter 91, x=[1.59064362]
 f(x)=2.530147127471909, is converged=[False]
==> Iter 92, x=[1.55883075]
 f(x)=2.429953301224021, is converged=[False]
==> Iter 93, x=[1.52765413]
 f(x)=2.33372715049555, is converged=[False]
==> Iter 94, x=[1.49710105]
 f(x)=2.241311555335926, is converged=[False]
==> Iter 95, x=[1.46715903]
 f(x)=2.1525556177446235, is converged=[False]
==> Iter 96, x=[1.43781585]
 f(x)=2.067314415281936, is converged=[False]
==> Iter 97, x=[1.40905953]
 f(x)=1.9854487644367713, is converged=[False]
==> Iter 98, x=[1.38087834]
 f(x)=1.906824993365075, is converged=[False]
==> Iter 99, x=[1.35326077]
 f(x)=1.8313147236278182, is converged=[False]
==> Iter 100, x=[1.32619556]
 f(x)=1.7587946605721563, is converged=[False]
==> Iter 101, x=[1.29967165]
 f(x)=1.689146392013499, is converged=[False]
==> Iter 102, x=[1.27367821]
 f(x)=1.6222561948897642, is converged=[False]
==> Iter 103, x=[1.24820465]
 f(x)=1.5580148495721295, is converged=[False]
==> Iter 104, x=[1.22324056]
 f(x)=1.496317461529073, is converged=[False]
==> Iter 105, x=[1.19877575]
 f(x)=1.4370632900525215, is converged=[False]
==> Iter 106, x=[1.17480023]
 f(x)=1.3801555837664417, is converged=[False]
==> Iter 107, x=[1.15130423]
 f(x)=1.325501422649291, is converged=[False]
==> Iter 108, x=[1.12827814]
 f(x)=1.273011566312379, is converged=[False]
==> Iter 109, x=[1.10571258]
 f(x)=1.2226003082864092, is converged=[False]
==> Iter 110, x=[1.08359833]
```

```
 f(x)=1.1741853360782672, is converged=[False]
==> Iter 111, x=[1.06192636]
 f(x)=1.1276875967695679, is converged=[False]
==> Iter 112, x=[1.04068783]
 f(x)=1.083031167937493, is converged=[False]
==> Iter 113, x=[1.01987408]
 f(x)=1.0401431336871685, is converged=[False]
==> Iter 114, x=[0.9994766]
 f(x)=0.9989534655931566, is converged=[False]
==> Iter 115, x=[0.97948706]
 f(x)=0.9593949083556677, is converged=[False]
==> Iter 116, x=[0.95989732]
 f(x)=0.9214028699847832, is converged=[False]
==> Iter 117, x=[0.94069938]
 f(x)=0.8849153163333858, is converged=[False]
==> Iter 118, x=[0.92188539]
 f(x)=0.8498726698065837, is converged=[False]
==> Iter 119, x=[0.90344768]
 f(x)=0.816217712082243, is converged=[False]
==> Iter 120, x=[0.88537873]
 f(x)=0.7838954906837863, is converged=[False]
==> Iter 121, x=[0.86767115]
 f(x)=0.7528532292527083, is converged=[False]
==> Iter 122, x=[0.85031773]
 f(x)=0.723040241374301, is converged=[False]
==> Iter 123, x=[0.83331138]
 f(x)=0.6944078478158788, is converged=[False]
==> Iter 124, x=[0.81664515]
 f(x)=0.6669092970423699, is converged=[False]
==> Iter 125, x=[0.80031224]
 f(x)=0.640499688879492, is converged=[False]
==> Iter 126, x=[0.784306]
 f(x)=0.6151359011998642, is converged=[False]
==> Iter 127, x=[0.76861988]
 f(x)=0.5907765195123496, is converged=[False]
==> Iter 128, x=[0.75324748]
 f(x)=0.5673817693396604, is converged=[False]
==> Iter 129, x=[0.73818253]
 f(x)=0.5449134512738099, is converged=[False]
==> Iter 130, x=[0.72341888]
 f(x)=0.5233348786033669, is converged=[False]
==> Iter 131, x=[0.7089505]
 f(x)=0.5026108174106737, is converged=[False]
==> Iter 132, x=[0.69477149]
 f(x)=0.48270742904121106, is converged=[False]
==> Iter 133, x=[0.68087606]
 f(x)=0.4635922148511791, is converged=[False]
==> Iter 134, x=[0.66725854]
```

```
 f(x)=0.44523396314307234, is converged=[False]
==> Iter 135, x=[0.65391337]
 f(x)=0.42760269820260666, is converged=[False]
==> Iter 136, x=[0.6408351]
 f(x)=0.4106696313537834, is converged=[False]
==> Iter 137, x=[0.6280184]
 f(x)=0.3944071139521736, is converged=[False]
==> Iter 138, x=[0.61545803]
 f(x)=0.37878859223966754, is converged=[False]
==> Iter 139, x=[0.60314887]
 f(x)=0.3637885639869767, is converged=[False]
==> Iter 140, x=[0.5910859]
 f(x)=0.3493825368530924, is converged=[False]
==> Iter 141, x=[0.57926418]
 f(x)=0.33554698839371, is converged=[False]
==> Iter 142, x=[0.56767889]
 f(x)=0.322259327653319, is converged=[False]
==> Iter 143, x=[0.55632532]
 f(x)=0.30949785827824755, is converged=[False]
==> Iter 144, x=[0.54519881]
 f(x)=0.29724174309042894, is converged=[False]
==> Iter 145, x=[0.53429483]
 f(x)=0.285470970064048, is converged=[False]
==> Iter 146, x=[0.52360894]
 f(x)=0.2741663196495116, is converged=[False]
==> Iter 147, x=[0.51313676]
 f(x)=0.26330933339139095, is converged=[False]
==> Iter 148, x=[0.50287402]
 f(x)=0.2528822837890919, is converged=[False]
==> Iter 149, x=[0.49281654]
 f(x)=0.2428681453510438, is converged=[False]
==> Iter 150, x=[0.48296021]
 f(x)=0.2332505667951425, is converged=[False]
==> Iter 151, x=[0.47330101]
 f(x)=0.2240138443500549, is converged=[False]
==> Iter 152, x=[0.46383499]
 f(x)=0.2151428961137927, is converged=[False]
==> Iter 153, x=[0.45455829]
 f(x)=0.2066232374276865, is converged=[False]
==> Iter 154, x=[0.44546712]
 f(x)=0.1984409572255501, is converged=[False]
==> Iter 155, x=[0.43655778]
 f(x)=0.19058269531941832, is converged=[False]
==> Iter 156, x=[0.42782662]
 f(x)=0.18303562058476938, is converged=[False]
==> Iter 157, x=[0.41927009]
 f(x)=0.1757874100096125, is converged=[False]
==> Iter 158, x=[0.41088469]
```

```
 f(x)=0.16882622857323187, is converged=[False]
==> Iter 159, x=[0.402667]
 f(x)=0.16214070992173188, is converged=[False]
==> Iter 160, x=[0.39461366]
 f(x)=0.15571993780883128, is converged=[False]
==> Iter 161, x=[0.38672138]
 f(x)=0.14955342827160156, is converged=[False]
==> Iter 162, x=[0.37898696]
 f(x)=0.14363111251204616, is converged=[False]
==> Iter 163, x=[0.37140722]
 f(x)=0.13794332045656912, is converged=[False]
==> Iter 164, x=[0.36397907]
 f(x)=0.13248076496648897, is converged=[False]
==> Iter 165, x=[0.35669949]
 f(x)=0.12723452667381602, is converged=[False]
==> Iter 166, x=[0.3495655]
 f(x)=0.12219603941753289, is converged=[False]
==> Iter 167, x=[0.34257419]
 f(x)=0.11735707625659861, is converged=[False]
==> Iter 168, x=[0.33572271]
 f(x)=0.11270973603683729, is converged=[False]
==> Iter 169, x=[0.32900825]
 f(x)=0.10824643048977851, is converged=[False]
==> Iter 170, x=[0.32242809]
 f(x)=0.10395987184238328, is converged=[False]
==> Iter 171, x=[0.31597953]
 f(x)=0.0998430609174249, is converged=[False]
==> Iter 172, x=[0.30965994]
 f(x)=0.09588927570509487, is converged=[False]
==> Iter 173, x=[0.30346674]
 f(x)=0.09209206038717312, is converged=[False]
==> Iter 174, x=[0.2973974]
 f(x)=0.08844521479584107, is converged=[False]
==> Iter 175, x=[0.29144945]
 f(x)=0.08494278428992576, is converged=[False]
==> Iter 176, x=[0.28562047]
 f(x)=0.0815790500320447, is converged=[False]
==> Iter 177, x=[0.27990806]
 f(x)=0.07834851965077572, is converged=[False]
==> Iter 178, x=[0.27430989]
 f(x)=0.075245918272605, is converged=[False]
==> Iter 179, x=[0.2688237]
 f(x)=0.07226617990900985, is converged=[False]
==> Iter 180, x=[0.26344722]
 f(x)=0.06940443918461307, is converged=[False]
==> Iter 181, x=[0.25817828]
 f(x)=0.06665602339290239, is converged=[False]
==> Iter 182, x=[0.25301471]
```

```
 f(x)=0.06401644486654347, is converged=[False]
==> Iter 183, x=[0.24795442]
 f(x)=0.061481393649828346, is converged=[False]
==> Iter 184, x=[0.24299533]
 f(x)=0.05904673046129515, is converged=[False]
==> Iter 185, x=[0.23813542]
 f(x)=0.05670847993502787, is converged=[False]
==> Iter 186, x=[0.23337272]
 f(x)=0.05446282412960076, is converged=[False]
==> Iter 187, x=[0.22870526]
 f(x)=0.052306096294068564, is converged=[False]
==> Iter 188, x=[0.22413116]
 f(x)=0.050234774880823445, is converged=[False]
==> Iter 189, x=[0.21964853]
 f(x)=0.048245477795542835, is converged=[False]
==> Iter 190, x=[0.21525556]
 f(x)=0.046334956874839336, is converged=[False]
==> Iter 191, x=[0.21095045]
 f(x)=0.0445000925825957, is converged=[False]
==> Iter 192, x=[0.20673144]
 f(x)=0.04273788891632491, is converged=[False]
==> Iter 193, x=[0.20259681]
 f(x)=0.04104546851523845, is converged=[False]
==> Iter 194, x=[0.19854488]
 f(x)=0.03942006796203501, is converged=[False]
==> Iter 195, x=[0.19457398]
 f(x)=0.037859033270738425, is converged=[False]
==> Iter 196, x=[0.1906825]
 f(x)=0.036359815553217184, is converged=[False]
==> Iter 197, x=[0.18686885]
 f(x)=0.03491996685730978, is converged=[False]
==> Iter 198, x=[0.18313147]
 f(x)=0.03353713616976031, is converged=[False]
==> Iter 199, x=[0.17946884]
 f(x)=0.03220906557743781, is converged=[False]
==> Iter 200, x=[0.17587947]
 f(x)=0.030933586580571274, is converged=[False]
==> Iter 201, x=[0.17236188]
 f(x)=0.02970861655198065, is converged=[False]
==> Iter 202, x=[0.16891464]
 f(x)=0.028532155336522216, is converged=[False]
==> Iter 203, x=[0.16553635]
 f(x)=0.027402281985195943, is converged=[False]
==> Iter 204, x=[0.16222562]
 f(x)=0.02631715161858218, is converged=[False]
==> Iter 205, x=[0.15898111]
 f(x)=0.025274992414486327, is converged=[False]
==> Iter 206, x=[0.15580148]
```

```
 f(x)=0.024274102714872667, is converged=[False]
==> Iter 207, x=[0.15268546]
 f(x)=0.023312848247363707, is converged=[False]
==> Iter 208, x=[0.14963175]
 f(x)=0.0223896594567681l, is converged=[False]
==> Iter 209, x=[0.14663911]
 f(x)=0.021503028942280088, is converged=[False]
==> Iter 210, x=[0.14370633]
 f(x)=0.0206515089961658, is converged=[False]
==> Iter 211, x=[0.1408322]
 f(x)=0.019833709239917632, is converged=[False]
==> Iter 212, x=[0.13801556]
 f(x)=0.0190482943540169, is converged=[False]
==> Iter 213, x=[0.13525525]
 f(x)=0.018293981897597827, is converged=[False]
==> Iter 214, x=[0.13255014]
 f(x)=0.017569540214452957, is converged=[False]
==> Iter 215, x=[0.12989914]
 f(x)=0.016873786421960614, is converged=[False]
==> Iter 216, x=[0.12730116]
 f(x)=0.016205584479650977, is converged=[False]
==> Iter 217, x=[0.12475513]
 f(x)=0.0155638433342568, is converged=[False]
==> Iter 218, x=[0.12226003]
 f(x)=0.014947515138220234, is converged=[False]
==> Iter 219, x=[0.11981483]
 f(x)=0.014355593538746712, is converged=[False]
==> Iter 220, x=[0.11741853]
 f(x)=0.013787112034612343, is converged=[False]
==> Iter 221, x=[0.11507016]
 f(x)=0.013241142398041694, is converged=[False]
==> Iter 222, x=[0.11276876]
 f(x)=0.012716793159079242, is converged=[False]
==> Iter 223, x=[0.11051338]
 f(x)=0.012213208149979705, is converged=[False]
==> Iter 224, x=[0.10830312]
 f(x)=0.011729565107240509, is converged=[False]
==> Iter 225, x=[0.10613705]
 f(x)=0.011265074328993784, is converged=[False]
==> Iter 226, x=[0.10401431]
 f(x)=0.01081897738556563, is converged=[False]
==> Iter 227, x=[0.10193403]
 f(x)=0.010390545881097233, is converged=[False]
==> Iter 228, x=[0.09989535]
 f(x)=0.009979080264205782, is converged=[ True]
Converged in 228 steps
```

```
[69]: array([0.09989535])
```

1.4 Gradient descent for logistic regression

1.4.1 Data generation

```
[70]: # Feature 1
      raw_x1 = [0.959, 0.750, 0.395, 0.823, 0.761, 0.844]
      # Feature 2
      raw_x2 = [0.382, 0.306, 0.760, 0.764, 0.874, 0.435]
      # Observed labels
      raw_labels = [0, 0, 0, 1, 1, 1]
```

```
[71]: data_X = np.stack([raw_x1, raw_x2], axis=1)

      # Rows: samples, columns: features
      data_X
```

```
[71]: array([[0.959, 0.382],
             [0.75 , 0.306],
             [0.395, 0.76 ],
             [0.823, 0.764],
             [0.761, 0.874],
             [0.844, 0.435]])
```

```
[72]: gt_Y = np.array(raw_labels, dtype=np.float32)
      print(gt_Y)

      # Make Y a column vector
      gt_Y = gt_Y[:, np.newaxis]
      gt_Y
```

```
[0. 0. 0. 1. 1. 1.]
```

```
[72]: array([[0.],
             [0.],
             [0.],
             [1.],
             [1.],
             [1.]], dtype=float32)
```

1.4.2 Data visualization

```
[73]: import matplotlib.pyplot as plt
```

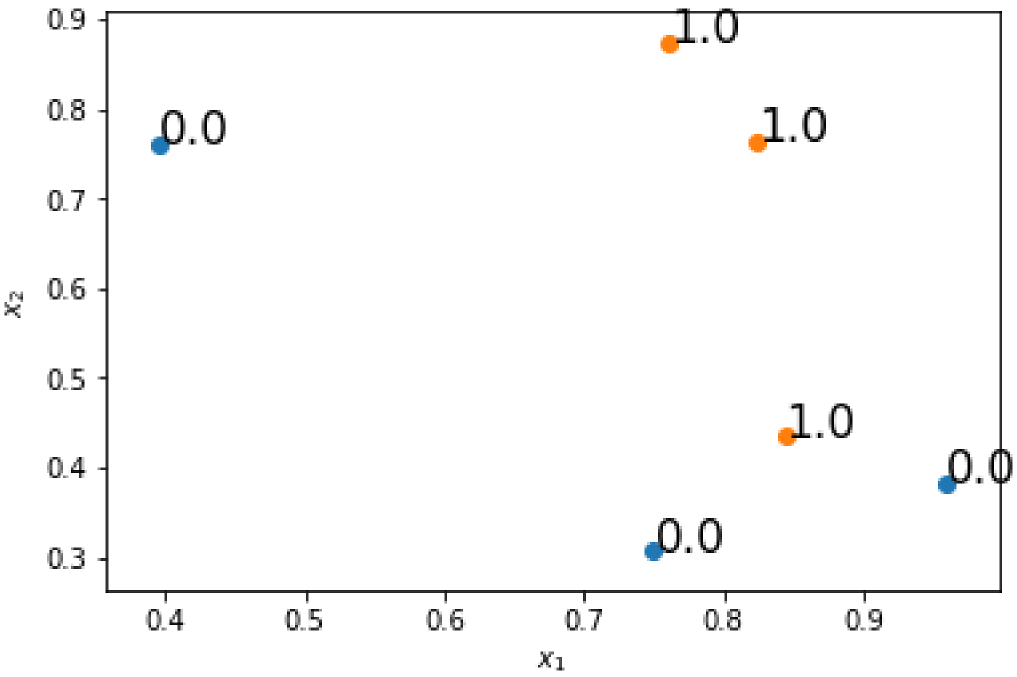
```
[75]: # Plot data points with label = 0
      neg_labels_idx = gt_Y[:,0] == 0
```

```
plt.scatter(data_X[neg_labels_idx, 0], data_X[neg_labels_idx, 1])

# Plot data points with label = 1
pos_labels_idx = gt_Y[:,0] == 1
plt.scatter(data_X[pos_labels_idx, 0], data_X[pos_labels_idx, 1])

# Annotate each point with its ground truth label
for i, txt in enumerate(gt_Y[:, 0]):
    plt.annotate(txt, data_X[i,0:2], fontsize=17)

plt.xlabel('$x_1$')
plt.ylabel('$x_2$')
plt.show()
```



1.4.3 Logistic regression

[50]:
```
# Sigmoid function
sigmoid = lambda z: 1.0 / (1 + np.exp(-z))

# Probability P(Y=1|X), parameterized by Theta
P = lambda X, Theta: np.clip(sigmoid(X @ Theta), 1e-3, 1 - 1e-3)

# Cross-entropy loss
```

```
ce_loss = lambda X, Theta, Y_true: np.sum(-Y_true * np.log(P(X, Theta)) - (1.0␣
↪- Y_true) * np.log(1 - P(X, Theta)))
d_ce_loss = lambda X, Theta, Y_true: np.sum((P(X, Theta) - Y_true) * X, axis=0,␣
↪keepdims=True).T  # Output: column vector

# Probability to category
def prob2category(prob_ls, threshold = 0.5):
    prob_ls = prob_ls.copy()
    prob_ls[prob_ls >= 0.5] = 1
    prob_ls[prob_ls < 0.5] = 0
    return prob_ls
```

[76]:

```
def logistic_regress(X, Y, learning_rate = 0.01):
    # Note that Theta should be in 2-dimensional real space
    Theta_0 = np.random.randn(X.shape[1], 1)

    converge_criterion3 = lambda last, curr: abs(last - curr) < 1e-5
    obj_f = lambda Theta: ce_loss(X, Theta, Y)
    d_obj_f = lambda Theta: d_ce_loss(X, Theta, Y)

    # Use gradient descent to find the best model
    Theta_optim = grad_descent(obj_f, d_obj_f, learning_rate,␣
↪converge_criterion3, Theta_0, max_iters=10000)

    return Theta_optim
```

[55]:

```
Theta_optim_1 = logistic_regress(data_X, gt_Y)
Theta_optim_1
```

```
Converged in 1995 steps
```

[55]:

```
array([[-0.64745928],
       [ 1.2060521 ]])
```

[77]:

```
prob_ls = P(data_X, Theta_optim_1)
pred_Y = prob2category(prob_ls)
pred_Y
```

[77]:

```
array([[0.],
       [0.],
       [1.],
       [1.],
       [1.],
       [0.]])
```

[78]:

```
# Plot data points with label = 0
neg_labels_idx = pred_Y[:,0] == 0
```

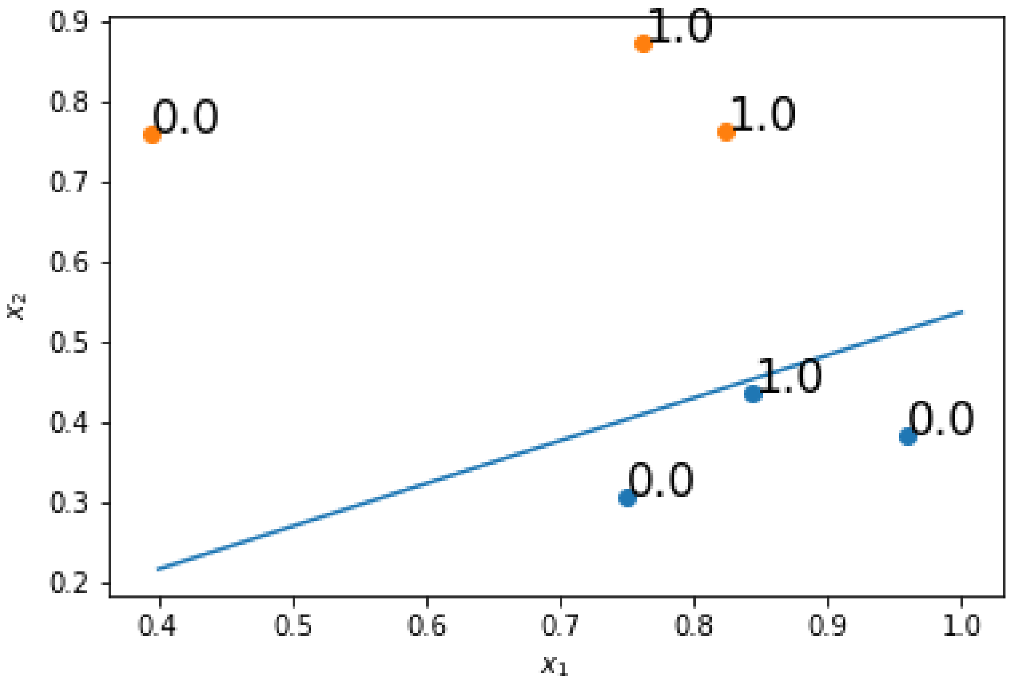
```
plt.scatter(data_X[neg_labels_idx, 0], data_X[neg_labels_idx, 1])

# Plot data points with label = 1
pos_labels_idx = pred_Y[:,0] == 1
plt.scatter(data_X[pos_labels_idx, 0], data_X[pos_labels_idx, 1])

# Annotate each point with its ground truth label
for i, txt in enumerate(gt_Y[:, 0]):
    plt.annotate(txt, data_X[i,0:2], fontsize=17)

# Draw decision boundary
range_x1 = np.linspace(0.4, 1, 100)
lin_eq = lambda x1: -Theta_optim_1[0] * x1  / Theta_optim_1[1]
plt.plot(range_x1, lin_eq(range_x1))


plt.xlabel('$x_1$')
plt.ylabel('$x_2$')
plt.show()
```



Add a bias term θ_0

[79]:
```
data_X2 = np.concatenate([data_X, np.ones((data_X.shape[0], 1))], axis=1)
data_X2
```

```
[79]: array([[0.959, 0.382, 1.   ],
              [0.75 , 0.306, 1.   ],
              [0.395, 0.76 , 1.   ],
              [0.823, 0.764, 1.   ],
              [0.761, 0.874, 1.   ],
              [0.844, 0.435, 1.   ]])
```

```
[80]: Theta_optim_2 = logistic_regress(data_X2, gt_Y, 0.1)
      Theta_optim_2
```

```
      Converged in 4836 steps
```

```
[80]: array([[ 11.01826859],
              [ 11.73760933],
              [-14.97535489]])
```

```
[81]: prob_ls = P(data_X2, Theta_optim_2)
      pred_Y2 = prob2category(prob_ls)
      pred_Y2
```

```
[81]: array([[1.],
              [0.],
              [0.],
              [1.],
              [1.],
              [0.]])
```

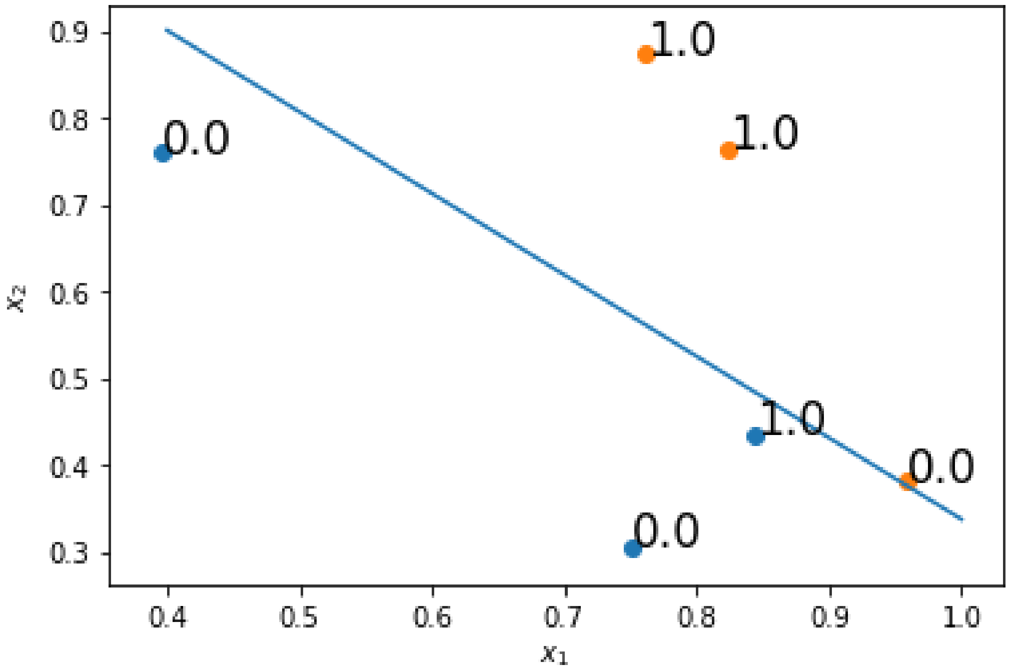
```
[82]: # Plot data points with label = 0
      neg_labels_idx = pred_Y2[:,0] == 0
      plt.scatter(data_X[neg_labels_idx, 0], data_X[neg_labels_idx, 1])

      # Plot data points with label = 1
      pos_labels_idx = pred_Y2[:,0] == 1
      plt.scatter(data_X[pos_labels_idx, 0], data_X[pos_labels_idx, 1])

      range_x1 = np.linspace(0.4, 1, 100)
      lin_eq = lambda x1: (-Theta_optim_2[0] * x1 - Theta_optim_2[2]) /␣
       ↪Theta_optim_2[1]
      plt.plot(range_x1, lin_eq(range_x1))

      for i, txt in enumerate(gt_Y[:, 0]):
          plt.annotate(txt, data_X[i,0:2], fontsize=17)

      plt.xlabel('$x_1$')
      plt.ylabel('$x_2$')
      plt.show()
```


0.9
0.8
0.7
0.6
0.5
0.4
0.3
x_2
0.4
0.5
0.6
0.7
0.8
0.9
1.0
x_1
1.0
1.0
0.0
1.0
0.0
0.0

```
[ ]:
```