d5_breast_cancer_detector

October 21, 2022

Table of Contents

1 Breast Cancer Wisconsin (Diagnostic) Data Set

https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(diagnostic)

1.1 Attribute Information:

1. ID number
2. Diagnosis (M = malignant:-1, B = benign:1) M: B:
3. 3-32

Ten real-valued features are computed for each cell nucleus:

- radius (mean of distances from center to points on the perimeter)
- texture (standard deviation of gray-scale values)
- perimeter
- area
- smoothness (local variation in radius lengths)
- compactness (perimeter^2 / area - 1.0)
- concavity (severity of concave portions of the contour)
- concave points (number of concave portions of the contour)
- symmetry
- fractal dimension ("coastline approximation" - 1)

mean, stderr, worst

http://people.idsia.ch/~juergen/deeplearningwinsMICCAIgrandchallenge.html

1.2

(training) $\boldsymbol{A}$ $\boldsymbol{b}$ (-1) (1)

$\boldsymbol{y}$ $C(\boldsymbol{y})$

1.3

( ) $\mathbb{R}^D$ $\mathbb{R}$ $h(\cdot)$

$$C(\boldsymbol{y}) = \begin{cases} +1 & \text{when} \quad h(\boldsymbol{y}) \geq 0 \\ -1 & \text{when} \quad h(\boldsymbol{y}) < 0 \end{cases}$$

$h : \mathbb{R}^D \to \mathbb{R}$ D $\boldsymbol{w}$

$$h(\boldsymbol{y}) = \boldsymbol{w} \cdot \boldsymbol{y}$$

D $\boldsymbol{w}$ $\boldsymbol{w}$ h $\boldsymbol{w}$

-1,1 $\boldsymbol{w}$

2

$$L(w) = \sum_{i=1}^{n} (A_i \cdot \boldsymbol{w} - b_i)^2$$

$\boldsymbol{w}$ j

$$\frac{\partial L}{\partial w_j} = \sum_{i=1}^{n} \frac{\partial}{\partial w_j} (A_i \cdot w - b_i)^2$$

$$= \sum_{i=1}^{n} 2(A_i \cdot w - b_i) A_{ij}$$

A_{ij} A_i j $\frac{\partial L}{\partial w_j}$ $\boldsymbol{w}_j$ (slope) $L(w)$ (local minimum)

(steepest descent method)

2.1 print_w

$\boldsymbol{w}$ j

[ ]:
```
def print_w(w):
    params = ["radius", "texture","perimeter","area",
              "smoothness","compactness","concavity","concave points",
              "symmetry","fractal dimension"]
    print("    (params)      :    ",end="")
    print("   (mean)     (stderr)      (worst)")
    for i, param in enumerate(params):
```

```
        print("%18s:" %param, end="")
        for j in range(3):
            print("%13.9f" % w[i*3+j], end="")
        print()
```

2.2

[ ]:
```
import numpy as np
tmp = np.fromfile('./train_A.data', np.float64, -1, " ")
A = tmp.reshape(300,30)
tmp = np.fromfile('./train_b.data', np.float64, -1, " ")
b = tmp.reshape(300,1)
w = np.zeros(30).reshape(30,1)
for i in range(30):
    w[i] = 0
```

[ ]:
```
A[0]
```

[ ]:
```
array([1.799e+01, 1.038e+01, 1.228e+02, 1.001e+03, 1.184e-01, 2.776e-01,
       3.001e-01, 1.471e-01, 2.419e-01, 7.871e-02, 1.095e+00, 9.053e-01,
       8.589e+00, 1.534e+02, 6.399e-03, 4.904e-02, 5.373e-02, 1.587e-02,
       3.003e-02, 6.193e-03, 2.538e+01, 1.733e+01, 1.846e+02, 2.019e+03,
       1.622e-01, 6.656e-01, 7.119e-01, 2.654e-01, 4.601e-01, 1.189e-01])
```

[ ]:
```
b[0]
```

[ ]:
```
array([-1.])
```

2.3 w (steepest descent)

[ ]:
```
loop, sigma = 300, 3.0*10**(-9)
for i in range(loop):
  dLw = A.dot(w)-b
  w = w - (dLw.transpose().dot(A)).transpose()*sigma

print_w(w)
```

```
    (params)      :       (mean)     (stderr)      (worst)
            radius:  0.000426997  0.000741817  0.002548876
           texture:  0.001687946  0.000004707  0.000000127
         perimeter: -0.000003968 -0.000002078  0.000008954
              area:  0.000003595  0.000002569  0.000070324
        smoothness:  0.000001139 -0.000881778  0.000000430
       compactness:  0.000000441  0.000000723  0.000000267
         concavity:  0.000001200  0.000000191  0.000411499
    concave points:  0.000921972  0.002395138 -0.001932789
```

```
         symmetry:  0.000005930 -0.000003750 -0.000008147
fractal dimension: -0.000002341  0.000011565  0.000003523
```

```
   (params)      :       (mean)     (stderr)      (worst)
           radius:  0.000426997  0.000741817  0.002548876
          texture:  0.001687946  0.000004707  0.000000127
        perimeter: -0.000003968 -0.000002078  0.000008954
             area:  0.000003595  0.000002569  0.000070324
       smoothness:  0.000001139 -0.000881778  0.000000430
      compactness:  0.000000441  0.000000723  0.000000267
        concavity:  0.000001200  0.000000191  0.000411499
   concave points:  0.000921972  0.002395138 -0.001932789
         symmetry:  0.000005930 -0.000003750 -0.000008147
fractal dimension: -0.000002341  0.000011565  0.000003523
```

3

[ ]:
```
def show_accuracy(mA, vb, vw):
    # M: (-1) B: (1)

    correct,safe_error,critical_error=0,0,0
    predict = mA.dot(vw)
    n = vb.size
    for i in range(n):
        if predict[i]*vb[i]>0:
            correct += 1
        elif (predict[i]<0 and vb[i]>0): #
            safe_error += 1
        elif (predict[i]>0 and vb[i]<0): #
            critical_error += 1
    print("       correct: %4d/%4d" % (correct,n))
    print("    safe error: %4d" % safe_error)
    print("critical error: %4d" % critical_error)
```

[ ]:
```
show_accuracy(A, b, w)
```

```
       correct:  274/ 300
    safe error:    5
critical error:   21
```

[ ]:
```
tmp = np.fromfile('./validate_A.data', np.float64, -1, " ")
A = tmp.reshape(260,30)
tmp = np.fromfile('./validate_b.data', np.float64, -1, " ")
b = tmp.reshape(260,1)

show_accuracy(A, b, w)
```

```
       correct:  240/ 260
```

```
    safe error:   10
critical error:   10
```

4 QR decomposition

QR A $||A.w - b||^2$ w

QR $n \times m$

$$A = QR$$

Q $n \times m$ R $m \times m$

$||Aw - b||$ QR

$$Q.R.w = bR.w = Q^t.bR^{-1}.R.w = R^{-1}.Q^t.b$$

[ ]:
```
import numpy as np

tmp = np.fromfile('./train_A.data', np.float64, -1, " ")
A = tmp.reshape(300,30)
tmp = np.fromfile('./train_b.data', np.float64, -1, " ")
b = tmp.reshape(300,1)

q, r = np.linalg.qr(A)
```

[ ]:
```
ww = np.linalg.inv(r).dot(np.transpose(q).dot(b))
```

[ ]:
```
q.shape
```

[ ]:
```
(300, 30)
```

[ ]:
```
print(r[0,0:5])
```

```
[-2.57579883e+02 -3.32324268e+02 -1.68607899e+03 -1.29450676e+04
 -1.65446346e+00]
```

[ ]:
```
show_accuracy(A, b, ww)
```

```
      correct:  286/ 300
    safe error:    1
critical error:   13
```

[ ]:
```
print_w(ww)
```

```
    (params)      :       (mean)     (stderr)      (worst)
            radius:  0.869921844 -0.024313948 -0.062679561
           texture: -0.003274619 -8.790300861  1.747147500
         perimeter: -0.202849407 -6.506451098  5.061760446
              area: 49.167541566 -0.956591421 -0.082052658
```

```
         smoothness: -0.007943157  0.004976908-27.841944367
        compactness:  3.301527110  4.985959134-16.318886295
          concavity: 10.316289081-21.332232171 -0.408605816
     concave points: -0.003345722 -0.000677873  0.002510735
           symmetry:  4.531369718  0.590110016 -0.719368704
  fractal dimension: -2.158965299 -3.803467225-12.298417038
```

[ ]:
```
tmp = np.fromfile('./validate_A.data', np.float64, -1, " ")
A = tmp.reshape(260,30)
tmp = np.fromfile('./validate_b.data', np.float64, -1, " ")
b = tmp.reshape(260,1)

show_accuracy(A, b, ww)
```

```
       correct:  252/ 260
    safe error:    6
critical error:    2
```