

```

19 phi = lambda u: 1/(1+np.exp(-u))
20 dphi = lambda u: np.exp(-u)/(1+np.exp(-u))**2
21 soft = lambda u: np.exp(u)/np.sum(np.exp(u),axis=1).reshape(-1,1)
22
23 L = 7
24 np.random.seed(1)
25 b1, W1 = 2*np.random.rand(L)-1, 2*np.random.rand(2,L)-1
26 b2, W2 = 2*np.random.rand(3)-1, 2*np.random.rand(L,3)-1
27
28 MaxEpoch = 1000
29 alpha = 0.01
30 rho1, rho2, delta = 0.9, 0.999, 1e-7
31 ns = MaxEpoch/10
32
33 sb1, sW1 = np.zeros(L), np.zeros((2,L))
34 sb2, sW2 = np.zeros(3), np.zeros((L,3))
35
36 rb1, rW1 = np.zeros(L), np.zeros((2,L))
37 rb2, rW2 = np.zeros(3), np.zeros((L,3))
38
39 print('n={:4d}'.format(0))
40 for n in range(MaxEpoch):
41     uh = b1 + xx@W1
42     zz = phi(uh)
43     uu = b2 + zz@W2
44     yy = soft(uu)
45
46     temp = np.reshape(np.sum((yy-tt)*yy,axis=1),(-1,1))
47     dEdu2 = (yy-tt)*yy - temp*yy
48     gradb2 = 1/N*np.sum(dEdu2,axis=0)
49     gradW2 = 1/N*zz.T@dEdu2
50
51     dEdu1 = (dEdu2@W2.T)*dphi(uh)
52     gradb1 = 1/N*np.sum(dEdu1,axis=0)
53     gradW1 = 1/N*xx.T@dEdu1
54
55     sb1, sW1 = rho1*sb1+(1-rho1)*gradb1, rho1*sW1+(1-rho1)*gradW1
56     sb2, sW2 = rho1*sb2+(1-rho1)*gradb2, rho1*sW2+(1-rho1)*gradW2
57
58     rb1, rW1 = rho2*rb1+(1-rho2)*gradb1**2, rho2*rW1+(1-rho2)*gradW1**2
59     rb2, rW2 = rho2*rb2+(1-rho2)*gradb2**2, rho2*rW2+(1-rho2)*gradW2**2
60
61     hsb1, hsW1 = sb1/(1-rho1**(n+1)), sW1/(1-rho1**(n+1))
62     hsb2, hsW2 = sb2/(1-rho1**(n+1)), sW2/(1-rho1**(n+1))
63
64     hrb1, hrW1 = rb1/(1-rho2**(n+1)), rW1/(1-rho2**(n+1))
65     hrb2, hrW2 = rb2/(1-rho2**(n+1)), rW2/(1-rho2**(n+1))
66
67     b1 = b1 - alpha/(delta+np.sqrt(hrb1))*hsb1
68     W1 = W1 - alpha/(delta+np.sqrt(hrW1))*hsW1
69     b2 = b2 - alpha/(delta+np.sqrt(hrb2))*hsb2

```

```

70 W2 = W2 - alpha/(delta+np.sqrt(hrW2))*hsw2
71
72 uh = b1 + xx@W1
73 uu = b2 + phi(uh)@W2
74 yy = soft(uu)
75 Eval = 0.5/N*np.sum((yy-tt)**2)
76
77 acc = np.sum(np.argmax(yy,axis=1)==np.argmax(tt,axis=1))/N*100
78
79 if (n+1)%ns == 0:
80     print('n={:4d}, Eval={:.4f}, Acc={:5.1f}%'.format(n+1,Eval,acc))

```

```

n= 0
n= 100, Eval=0.2085, Acc= 63.0%
n= 200, Eval=0.1882, Acc= 72.0%
n= 300, Eval=0.1523, Acc= 78.0%
n= 400, Eval=0.0870, Acc= 92.0%
n= 500, Eval=0.0604, Acc= 94.0%
n= 600, Eval=0.0450, Acc= 97.0%
n= 700, Eval=0.0324, Acc=100.0%
n= 800, Eval=0.0235, Acc=100.0%
n= 900, Eval=0.0177, Acc=100.0%
n=1000, Eval=0.0141, Acc=100.0%

```

```

81 xx1, xx2 = np.meshgrid(np.linspace(-1,1), np.linspace(-1,1))
82 XX = np.column_stack((xx1.reshape(-1),xx2.reshape(-1)))
83 UH = b1 + XX@W1
84 UU = b2 + phi(UH)@W2
85 YY = soft(UU)
86 yy = [YY[:,j].reshape(xx1.shape) for j in range(3)]
87
88 fig = plt.figure()
89 ax = plt.axes()
90 gcmap = plt.colormaps.get_cmap('brg')
91 ax.scatter(xx[:,0], xx[:,1], c=labels, cmap=gcmap)
92 for j in range(3):
93     ax.contour(xx1, xx2, yy[j], [0.5])
94     ax.contourf(xx1, xx2, yy[j], [0.5,1.0], colors=[gcmap(j/2)], alpha=0.3)
95 ax.set_aspect('equal')
96 plt.show()

```

