set5

October 26, 2017

In [2]:
```
import numpy as np
import random as ran
```

In [3]:
```
w = [1,1]
r = 0.1

E = lambda u,v: (u*np.e**v-2*v*np.e**(-u))**2
dE_u = lambda u,v: 2*(np.e**v+2*v*np.e**(-u))*(u*np.e**v-2*v*np.e**(-u))
dE_v = lambda u,v: 2*(u*np.e**v-2*v*np.e**(-u))*(u*np.e**v-2*np.e**(-u))
```

In [4]:
```
iters = 0
e = E(w[0],w[1])
while e > 10**(-14):
    w0 = w[0] - r*dE_u(w[0],w[1])
    w1 = w[1] - r*dE_v(w[0],w[1])

    w[0] = w0
    w[1] = w1


    e = E(w[0],w[1])
    iters += 1
```

In [5]:
```
print ('it took : ' + str(iters) + ' iterations')
```

```
it took : 10 iterations
```

In [6]:
```
(w[0],w[1])
```

Out[6]:
```
(0.04473629039778207, 0.023958714099141746)
```

In [7]:
```
w = [1,1]
e = E(w[0],w[1])


for i in range(15):
    # move only in the u direction
    w0 = w[0] - r*dE_u(w[0],w[1])
```

```
            w[0] = w0

            # move only in the v direction
            w1 = w[1] - r*dE_v(w[0],w[1])
            w[1] = w1


            e = E(w[0],w[1])
```

In [8]:
```
print ('error after 15 iterations is: ' + str(e))
```

```
error after 15 iterations is: 0.13981379199615324
```

In [9]:
```
def getLine():
            (x1, y1) = (ran.uniform(-1,1), ran.uniform(-1,1))
            (x2, y2) = (ran.uniform(-1,1), ran.uniform(-1,1))
            line = lambda x:(y2-y1)/(x2-x1)*(x-x1)
            return line
```

In [10]:
```
def labelPts(N, line):
             pts = []

             for i in range(N):
                 (x1,x2) = ran.uniform(-1,1), ran.uniform(-1,1)

                 pts.append([np.array([1,x1,x2,]),np.sign(x2-line(x1))])

             return pts
```

In [11]:
```
def ptGradient(w, pt):
             exp = -pt[1]*w.dot(pt[0])
             return -(pt[1]*pt[0]/(1+np.e**(exp)))

         def ptCrossEntropyError(w, pt):
             exp = -pt[1]*w.dot(pt[0])
             return np.log(1+ np.e**(-exp))
```

In [12]:
```
r = 0.01
```

```
         all_epoc = []
         all_E_in = []
         all_E_out = []
         for i in range(100):
             line = getLine()
             pts = labelPts(100, line)
```

```
        epoc_end_w = np.array([0,0,0])
        epoc = 0
        w = np.array([0,0,0])
        while True:
            options = list(range(len(pts)))

            while len(options)>0:
                choice = ran.choice(options)
                options.remove(choice)

                pt = pts[choice]

                grad = ptGradient(w,pt)
                w = w + r * grad
            diff = epoc_end_w - w
            mag = np.sqrt(diff.dot(diff))
            if mag < 0.01:
                break
            epoc_end_w = w

            epoc += 1

        all_epoc.append(epoc)
        # get E_in
        E_in=0
        for pt in pts:
            E_in += ptCrossEntropyError(w,pt)

        E_in /= float(len(pts))
        all_E_in.append(E_in)


        # approx E_out
        test_pts = labelPts(10000, line)
        E_out = 0.0
        for pt in test_pts:
            E_out += ptCrossEntropyError(w,pt)

        E_out /= float(len(test_pts))
        all_E_out.append(E_out)
```

```
In [13]: print("average approx. E_out is: " + str(np.mean(all_E_out)))
```

```
average approx. E_out is: 0.10063694652
```

```
In [14]: print("average epocs: " + str(np.mean(all_epoc)))
```

```
average epocs: 333.21
```

In [ ]: