Question 4

May 1, 2019

In [1]:
```
import numpy as np
import os
```

In [2]:
```
x0 = 1
x1 = 0.05
x2 = 0.1
y = 0.9

a_k = 0.1 #learning rate
```

In [3]:
```
params = np.random.uniform(-1,1,9)
```

In [4]:
```
w0 = params[0]
w1 = params[1]
w2 = params[2]
a0 = params[3]
a1 = params[4]
a2 = params[5]
b0 = params[6]
b1 = params[7]
b2 = params[8]
```

In [5]:
```
def sigmoid_fn(x1):
    return(np.exp(x1)/(np.exp(x1)+1))
```

1 question (a) Start with initial values for ai's, bi's and wi's. Then calculate hi's, o and E.

```
In [6]: h0 = 1
        h1 = sigmoid_fn(a0+a1*x1+a2*x2)
        h2 = sigmoid_fn(b0+b1*x1+b2*x2)
        o  = sigmoid_fn(w0+w1*h1+w2*h2)

In [7]: E = ((y-o)**2)/2

In [8]: print("a0 = %f"%(a0))
        print("a1 = %f"%(a1))
        print("a2 = %f\n"%(a2))

        print("b0 = %f"%(b0))
        print("b1 = %f"%(b1))
        print("b2 = %f\n"%(b2))

        print("w0 = %f"%(w0))
        print("w1 = %f"%(w1))
        print("w2 = %f\n"%(w2))

        print("h1 = %f"%(h1))
        print("h2 = %f"%(h2))
        print("o = %f"%(o))
        print("E = %f"%(E))
```

```
a0 = -0.837356
a1 = -0.075478
a2 = 0.310341

b0 = 0.408669
b1 = 0.580519
b2 = 0.816329

w0 = -0.006751
w1 = -0.270897
w2 = -0.497940

h1 = 0.307870
h2 = 0.626990
o = 0.400747
E = 0.124627
```

2 question (b) Then update wi's, and o and E.

```
In [9]: dE_dw0 = -(y-o)*o*(1-o)*h0
        dE_dw1 = -(y-o)*o*(1-o)*h1
        dE_dw2 = -(y-o)*o*(1-o)*h2

        dE_da0 = -(y-o)*o*(1-o)*w1*h1*(1-h1)*x0
        dE_da1 = -(y-o)*o*(1-o)*w1*h1*(1-h1)*x1
        dE_da2 = -(y-o)*o*(1-o)*w1*h1*(1-h1)*x2

        dE_db0 = -(y-o)*o*(1-o)*w2*h2*(1-h2)*x0
        dE_db1 = -(y-o)*o*(1-o)*w2*h2*(1-h2)*x1
        dE_db2 = -(y-o)*o*(1-o)*w2*h2*(1-h2)*x2

In [10]: # update params wi's
         w0 -= a_k*dE_dw0
         w1 -= a_k*dE_dw1
         w2 -= a_k*dE_dw2

In [11]: h1 = sigmoid_fn(a0+a1*x1+a2*x2)
         h2 = sigmoid_fn(b0+b1*x1+b2*x2)
         o  = sigmoid_fn(w0+w1*h1+w2*h2)
         E = ((y-o)**2)/2

In [12]: print("o = %f\n"%(o))
         print("E = %f\n"%(E))

o = 0.405038

E = 0.122494
```

3 question (c) Then update ai's and bi's, and o and E.

In [13]:
```
# update params ai's, bi's
a0 -= a_k*dE_da0
a1 -= a_k*dE_da1
a2 -= a_k*dE_da2

b0 -= a_k*dE_db0
b1 -= a_k*dE_db1
b2 -= a_k*dE_db2
```

In [14]:
```
h1 = sigmoid_fn(a0+a1*x1+a2*x2)
h2 = sigmoid_fn(b0+b1*x1+b2*x2)
o  = sigmoid_fn(w0+w1*h1+w2*h2)
E = ((y-o)**2)/2
```

In [15]:
```
print("o = %f\n"%(o))
print("E = %f\n"%(E))
```

```
o = 0.405087

E = 0.122470
```

4 question (d) Repeat 2), 3)

In [16]:
```
step = 0
while True:
    # get derivate of parmas
    dE_dw0 = -(y-o)*o*(1-o)*h0
    dE_dw1 = -(y-o)*o*(1-o)*h1
    dE_dw2 = -(y-o)*o*(1-o)*h2

    dE_da0 = -(y-o)*o*(1-o)*w1*h1*(1-h1)*x0
    dE_da1 = -(y-o)*o*(1-o)*w1*h1*(1-h1)*x1
    dE_da2 = -(y-o)*o*(1-o)*w1*h1*(1-h1)*x2

    dE_db0 = -(y-o)*o*(1-o)*w2*h2*(1-h2)*x0
    dE_db1 = -(y-o)*o*(1-o)*w2*h2*(1-h2)*x1
    dE_db2 = -(y-o)*o*(1-o)*w2*h2*(1-h2)*x2

    # update params wi's
    w0 -= a_k*dE_dw0
    w1 -= a_k*dE_dw1
    w2 -= a_k*dE_dw2

    # update params ai's, bi's
    a0 -= a_k*dE_da0
    a1 -= a_k*dE_da1
    a2 -= a_k*dE_da2

    b0 -= a_k*dE_db0
    b1 -= a_k*dE_db1
    b2 -= a_k*dE_db2

    # hidden/output layer
    h1 = sigmoid_fn(a0+a1*x1+a2*x2)
    h2 = sigmoid_fn(b0+b1*x1+b2*x2)
    o  = sigmoid_fn(w0+w1*h1+w2*h2)
    E = ((y-o)**2)/2

    step += 1
    if step%100==0:
        print("step = %d, o = %f, E=%f"%(step,o,E))
    else:
        pass

    if E<0.00000001: # when Error < 0.00000001, finish training.
        print("training finished")
        break
    else:
        continue
```

```
            if step>10000: # prevent infinite loop
                break
            else:
                continue
```

```
step = 100, o = 0.687336, E=0.022613
step = 200, o = 0.776952, E=0.007570
step = 300, o = 0.816117, E=0.003518
step = 400, o = 0.837972, E=0.001924
step = 500, o = 0.851924, E=0.001156
step = 600, o = 0.861592, E=0.000738
step = 700, o = 0.868665, E=0.000491
step = 800, o = 0.874043, E=0.000337
step = 900, o = 0.878248, E=0.000237
step = 1000, o = 0.881609, E=0.000169
step = 1100, o = 0.884339, E=0.000123
step = 1200, o = 0.886587, E=0.000090
step = 1300, o = 0.888458, E=0.000067
step = 1400, o = 0.890030, E=0.000050
step = 1500, o = 0.891359, E=0.000037
step = 1600, o = 0.892491, E=0.000028
step = 1700, o = 0.893459, E=0.000021
step = 1800, o = 0.894291, E=0.000016
step = 1900, o = 0.895009, E=0.000012
step = 2000, o = 0.895630, E=0.000010
step = 2100, o = 0.896169, E=0.000007
step = 2200, o = 0.896637, E=0.000006
step = 2300, o = 0.897046, E=0.000004
step = 2400, o = 0.897403, E=0.000003
step = 2500, o = 0.897715, E=0.000003
step = 2600, o = 0.897988, E=0.000002
step = 2700, o = 0.898228, E=0.000002
step = 2800, o = 0.898438, E=0.000001
step = 2900, o = 0.898623, E=0.000001
step = 3000, o = 0.898786, E=0.000001
step = 3100, o = 0.898929, E=0.000001
step = 3200, o = 0.899054, E=0.000000
step = 3300, o = 0.899165, E=0.000000
step = 3400, o = 0.899263, E=0.000000
step = 3500, o = 0.899349, E=0.000000
step = 3600, o = 0.899425, E=0.000000
step = 3700, o = 0.899492, E=0.000000
step = 3800, o = 0.899551, E=0.000000
step = 3900, o = 0.899603, E=0.000000
step = 4000, o = 0.899650, E=0.000000
step = 4100, o = 0.899690, E=0.000000
step = 4200, o = 0.899726, E=0.000000
```

```
step = 4300, o = 0.899758, E=0.000000
step = 4400, o = 0.899786, E=0.000000
step = 4500, o = 0.899811, E=0.000000
step = 4600, o = 0.899833, E=0.000000
step = 4700, o = 0.899852, E=0.000000
training finished
```