Not, And, and Or

Rushi Shah

19 May 2016

1 Not

$$y = \frac{1}{1 + e^{100x-36}}$$

2 And

$$y = \frac{1}{1 + e^{-66(x_1+x_2)+100}}$$

3 Or

$$y = \frac{1}{1 + e^{-100(x_1+x_2)+63}}$$

4 Code

```
#Expected to use Python 3!!

from math import e

def f(w, b, x):
        return 1 / (1 + (e ** (-w * x + b)))

def g(w,b,x,y):
        return 1 / (1 + (e ** (-w * (x+y) + b)))

# not
# [0,1]
# [1,0]
def singleValueHillClimbing(xs, ys):
        bestW = 0
        bestB = 0
```

```
        bestError = 100000000
        w = 100
        b = 100
        while w > -100:
                while b > -100:
                        # print(w,b)
                        currError = 0
                        for i in range(len(xs)):
                                currError += (ys[i] - f(w, b, xs[i])) ** 2
                        # print(w,b,currError)
                        if(currError < bestError):
                                bestError = currError
                                bestW = w
                                bestB = b
                                print(bestW, bestB, currError)
                        b -= .1
                b = 100
                w -= .1

# # AND
# # 0 -> 0 -> 0
# # 0 -> 1 -> 0
# # 1 -> 0 -> 0
# # 1 -> 1 -> 1
# [0,0,1,1]
# [0,1,0,1]
# [0,0,0,1]

# # OR
# # 0 -> 0 -> 0
# # 0 -> 1 -> 1
# # 1 -> 0 -> 1
# # 1 -> 1 -> 1
# [0,0,1,1]
# [0,1,0,1]
# [0,1,1,1]
def doubleValueHillClimbing(xs, ys, zs):
        bestW = 0
        bestB = 0
        bestError = 100000000
        w = 100
        b = 100
        while w > -100:
                while b > -100:
                        # print(w,b)
                        currError = 0
```

```
        for i in range(len(xs)):
                currError += (zs[i] - g(w, b, xs[i], ys[i])) ** 2
        # print(w,b,currError)
        if(currError < bestError):
                bestError = currError
                bestW = w
                bestB = b
                print(bestW, bestB, currError)
        b -= .1
b = 100
w -= .1
```