Regresión Lineal

```
import numpy as num

N = 6
M = 2
P = 1000

m = num.random.uniform( -1, 1, (N+1,M))

x = num.random.uniform( -9, 9, (P,N+1))
x[:,-1] = 1

z = num.dot(x,m)

w = num.random.uniform( -0.1, 0.1, (N+1,M))

lr = 1e-5
E, t = 1., 0
while E>0.01 and t<900:
    y = num.dot( x, w)
    d = z-y
    dw = lr*num.dot( x.T, d)
    w += dw
    E = num.mean( num.square( d))
    t += 1
    if t%10==0:
        print(t, E)

xp = num.random.uniform( -25, 25, (100,N+1))
xp[:,-1] = 1
zp = num.dot(xp,m)

yp = num.dot( xp, w)
ep = num.mean( num.square( zp-yp))
print( "Error de testeo:", ep)

print( m)
print( w)
```

Perceptrón Simple

```
import numpy as num

N = 6
M = 2
P = 1000

m = num.random.uniform( -1, 1, (N+1,M))

x = num.random.uniform( -9, 9, (P,N+1))
x[:,-1] = 1
z = num.sign( num.dot(x,m))

w = num.random.uniform( -0.1, 0.1, (N+1,M))

lr = 0.1/P
E, t = 1., 0
while E>0.01 and t<9000:
    y = num.tanh( num.dot( x, w))
    d = z-y
    dw = lr*num.dot( x.T, d)
    w += dw
    E = num.mean( num.square( d))
    t += 1
    if t%100==0:
        print(t, E)

xp = num.random.uniform( -25, 25, (100,N+1))
xp[:,-1] = 1
zp = num.sign( num.dot(xp,m))

yp = num.tanh( num.dot( xp, w))
ep = num.mean( num.square( zp-yp))
print( ep)

# c = w[0,0]/m[0,0]
# print( m*c)
# print( w)

# M = 1
# z = num.sign( num.prod( x, axis=1).reshape((P,1)))
# zp = num.sign( num.prod( xp, axis=1))
```