**Trabajo Práctico 7**

**Probando red neuronal con gestos distintos a los de entrenamiento**

**Probamos la red neuronal a medida que la vamos entrenando.  
Ingresamos los 6 gestos restantes 3 de cada persona para ver si logra reconocerlos y aprende a generalizar gestos.**

**from math import e**

**from random import random, uniform**

**import matplotlib.pyplot as plt**

**from PIL import Image**

**import numpy as np**

**def generalizar(imagenes,n,w\_oculta,w\_salida,salidas\_gen=[[],[],[],[],[],[]]):**

**for pix in range(len(imagenes)):**

**x = 0**

**solucion=(0,1,0,1,0,1)**

**# Salida de las neuronas de la capa oculta**

**salida = [1]**

**solucion\_deseada = int(solucion[pix])**

**x\_neu\_salida = 0**

**for i in range(n):**

**x = 0**

**for j in range(len(imagenes[pix])):**

**# x es sumatoria de cada peso por su entrada**

**x = x + imagenes[pix][j]\*w\_oculta[i][j]**

**# salida de las neuronas capa oculta**

**salida.append(1/(1+e\*\*(-x)))**

**#print(salida)**

**for i in range(n+1):**

**# salida de la ultima neurona**

**x\_neu\_salida += salida[i]\*w\_salida[i]**

**salida\_real = 1/(1+e\*\*(-x\_neu\_salida))**

**#error = solucion\_deseada-salida\_real**

**salidas\_gen[pix].append(salida\_real)**

**return salidas\_gen**

**def entrenar\_perceptron(imagenes, lr, iteraciones, n,solucion,imagenes\_generalizar):**

**errores= [[],[],[],[],[],[],[],[],[],[]]**

**# Pesos de la Capa oculta**

**w\_oculta = []**

**# Pesos ultima Neurona**

**w\_salida = []**

**for i in range(n):**

**w\_oculta.append([])**

**for \_ in range(7681):**

**w\_oculta[i].append(round(uniform(-0.01, 0.01), 2))**

**# Pesos de la Ultima Neurona**

**for i in range(n+1):**

**w\_salida.append(round(uniform(-0.01, 0.01), 2))**

**#print("Pesos capa oculta:", w\_oculta)**

**#print("Pesos ultima neurona:", w\_salida)**

**for iteraciones in range(iteraciones):**

**print("Iteracion: ", iteraciones+1)**

**for pix in range(len(imagenes)):**

**x = 0**

**# Salida de las neuronas de la capa oculta**

**salida = [1]**

**# Delta de los pesos de la neurona de salida**

**dw\_n\_salida = []**

**# Delta de las neuronas capa oculta**

**delta\_co = []**

**# Delta de los pesos de las neuronas capa oculta**

**dw\_co = []**

**solucion\_deseada = int(solucion[pix])**

**#print(solucion\_deseada)**

**x\_neu\_salida = 0**

**#print(imagenes[pix][0])**

**#print(len(imagenes[pix]))**

**for i in range(n):**

**x = 0**

**for j in range(len(imagenes[pix])):**

**# x es sumatoria de cada peso por su entrada**

**x = x + imagenes[pix][j]\*w\_oculta[i][j]**

**# salida de las neuronas capa oculta**

**salida.append(1/(1+e\*\*(-x)))**

**#print(salida)**

**for i in range(n+1):**

**# salida de la ultima neurona**

**x\_neu\_salida += salida[i]\*w\_salida[i]**

**salida\_real = 1/(1+e\*\*(-x\_neu\_salida))**

**error = solucion\_deseada-salida\_real**

**deltaf = salida\_real\*(1-salida\_real)\*error**

**# Delta de los pesos de la neurona de salida**

**for i in range(n+1):**

**dw\_n\_salida.append(lr\*salida[i]\*deltaf)**

**# Nuevos pesos ultima Neurona**

**w\_salida[i] = w\_salida[i]+dw\_n\_salida[i]**

**salida.pop(0)**

**for i in range(n):**

**# Deltas de las neuronas capa oculta**

**delta\_co.append(salida[i]\*(1-salida[i])\*deltaf)**

**for en in imagenes[pix]:**

**# Delta de los pesos de la capa oculta**

**dw\_co.append(lr\*en\*delta\_co[i])**

**dw\_co.append(lr\*en\*delta\_co[i])**

**dw\_co.append(lr\*en\*delta\_co[i])**

**count = 0**

**for i in range(n):**

**for j in range(3):**

**# Nuevos pesos de la capa oculta**

**w\_oculta[i][j] = w\_oculta[i][j]+dw\_co[count]**

**count += 1**

**#print("Deltas de los pesos ultima Neurona:",dw\_n\_salida)**

**#print("Pesos ultima neurona:",w\_salida)**

**#print("Nuevos pesos:",w\_oculta)**

**#print("Salida real:", salida\_real)**

**#print("Salida deseada:", solucion\_deseada)**

**#print("error:", error)**

**#print("-------------------------")**

**#errores[pix].append(error)**

**salidas\_gen=generalizar(imagenes\_generalizar,n,w\_oculta,w\_salida)**

**plt.plot(salidas\_gen[0])**

**plt.plot(salidas\_gen[1])**

**plt.plot(salidas\_gen[2])**

**plt.plot(salidas\_gen[3])**

**plt.plot(salidas\_gen[4])**

**plt.plot(salidas\_gen[5])**

**#plt.plot(errores\_gen[6])**

**#plt.plot(errores\_gen[7])**

**#plt.plot(errores\_gen[8])**

**#plt.plot(errores\_gen[9])**

**plt.ylabel('Salida real')**

**plt.xlabel('Iteraciones')**

**plt.show()**

**def leer\_imagenes():**

**ruta='/home/valentin/Escritorio/5to/InteligenciaArtificial/tp6/fotos/'**

**lista=[]**

**imagenes=[]**

**for gesto in range(1,6):**

**for persona in 'AB':**

**nombre=str(gesto)+persona+'58018.jpg'**

**#print(nombre)**

**img = Image.open(ruta+nombre)**

**w,h=img.size**

**#print("entradas:",w\*h)**

**#print(h)**

**pixel=img.load()**

**for i in range(w):**

**for j in range(h):**

**lista.append(pixel[i,j][0])**

**lista.insert(0,1)**

**imagenes.append(lista)**

**lista=[]**

**#print(len(imagenes))**

**return(imagenes)**

**def leer\_imagenes2():**

**ruta='/home/valentin/Escritorio/5to/InteligenciaArtificial/tp6/fotos/'**

**lista=[]**

**imagenes=[]**

**for gesto in range(6,9):**

**#print(gesto)**

**for persona in 'AB':**

**nombre=str(gesto)+persona+'58018.jpg'**

**print(nombre)**

**img = Image.open(ruta+nombre)**

**w,h=img.size**

**#print("entradas:",w\*h)**

**#print(h)**

**pixel=img.load()**

**for i in range(w):**

**for j in range(h):**

**lista.append(pixel[i,j][0])**

**lista.insert(0,1)**

**imagenes.append(lista)**

**lista=[]**

**#print(len(imagenes))**

**return(imagenes)**

**if \_\_name\_\_ == "\_\_main\_\_":**

**#Datos de las imagenes que vamos a usar para entrenar la red**

**imagenes\_entrenar=leer\_imagenes()**

**#Datos de las imagenes que vamos a usar para ver si aprendio a reconocer cualquier tipo de gesto de la persona A y B**

**imagenes\_generalizar=leer\_imagenes2()**

**lr = 0.5**

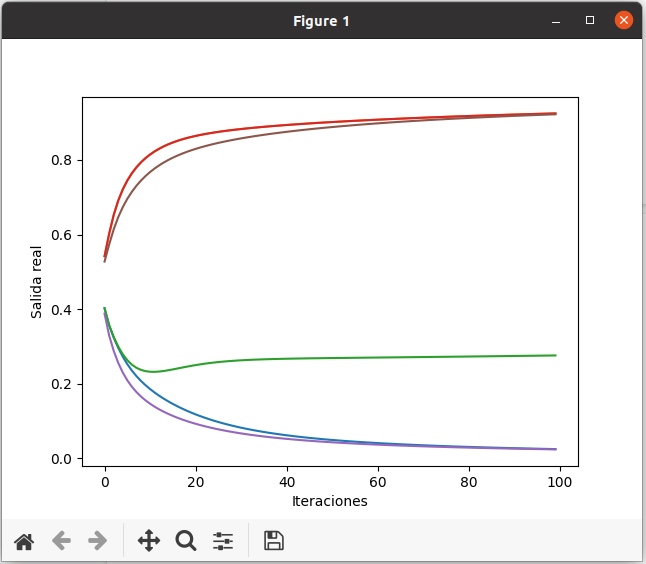
**iteraciones = 100**

**neuronas = 10**

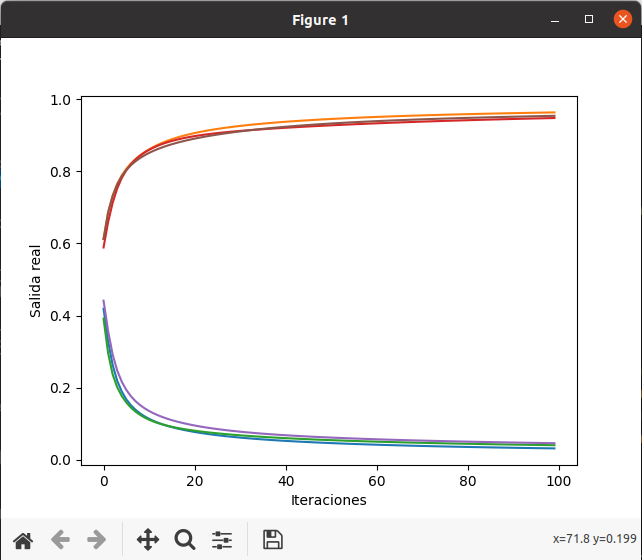
**solucion=(0,1,0,1,0,1,0,1,0,1)**

**entrenar\_perceptron(imagenes\_entrenar, lr, iteraciones, neuronas,solucion, imagenes\_generalizar)**

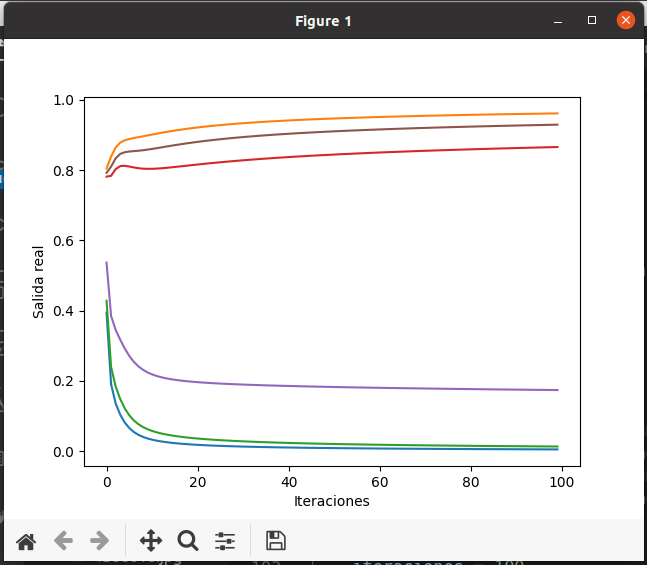
**10 NEURONAS**



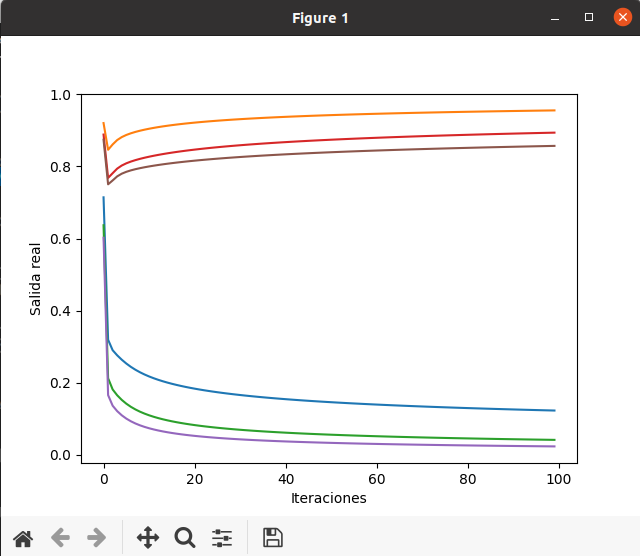
**20 NEURONAS**

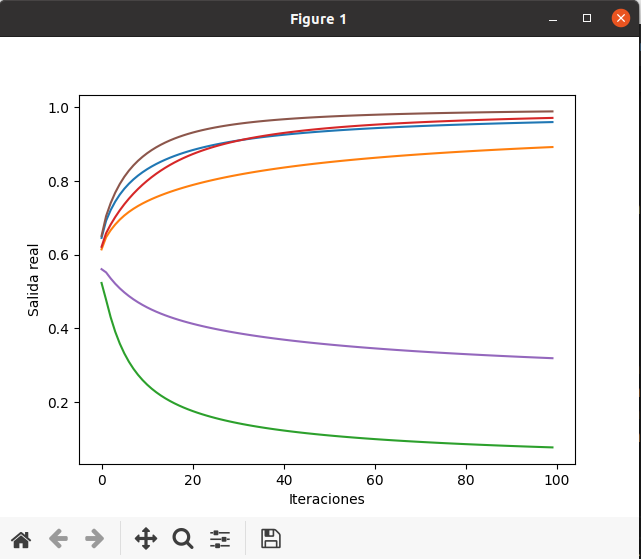


**50 NEURONAS**



**100 NEURONAS**





**Podemos afirmar que en todos los casos nuestra red pudo generalizar y diferenciar a las dos personas con gestos distintos a los entrenados !!!!!!!**