

Inteligencia Artificial

Trabajo Práctico 6 y 7

Perceptrón Multicapa

Imágenes de dos personas como entradas de la red

Expandir el código para tomar las imágenes como "entradas" del perceptrón multicapa. Así, en lugar de tener sólo 2 entradas e_1 y e_2 , con valores 0 o 1 habrán $80 \times 96 = 7680$ entradas (e_1 a e_{7680} , con valores de 0 a 255). Cada uno de estos valores de entrada será un pixel de la imagen.

Recomendación: investigar cómo se pueden leer imágenes en Python (o el lenguaje que estén usando) para obtener un valor de 0 a 255 por cada pixel.

Al ser una imagen con niveles de gris, los valores R, G y B devueltos son iguales para cada pixel.

En cuanto a los valores de las "salidas deseadas", se toma "0" para la persona A y "1" para la persona B.

La tabla de verdad se construye alternando una imagen perteneciente a A y otra perteneciente a B.

Tomar 5 ejemplos de cada una, con los gestos de 1 a 5.

Presentar gráfica de errores. Como son 10 filas de datos, deben aparecer 10 errores.

Recomendación: en lugar de generar los pesos aleatorios iniciales con valores de -1 a 1 utilicen -0.01 a +0.01.

La cantidad inicial de neuronas que utilizaremos para la capa oculta será de 100.

En este caso, para agilizar la lectura de las fotos, lo que se realizó fue convertir las imágenes de entrada a blanco y negro utilizando las funciones de opencv. Una vez leídas las imágenes se agrega a cada fila el valor de salida deseado, para armar la tabla de la verdad correspondiente al sistema.

Luego de correr el sistema con 200 neuronas en la capa oculta y cumplir 100 iteraciones, los resultados obtenidos para cada entrada fueron los siguientes:

Salida Deseada: 0.000000 Perceptrón salida: 0.079678

Salida Deseada: 1.000000 Perceptrón salida: 0.984618

Salida Deseada: 0.000000 Perceptrón salida: 0.021050

Salida Deseada: 1.000000 Perceptrón salida: 0.990624

Salida Deseada: 0.000000 Perceptrón salida: 0.021992

Salida Deseada: 1.000000 Perceptrón salida: 0.939488

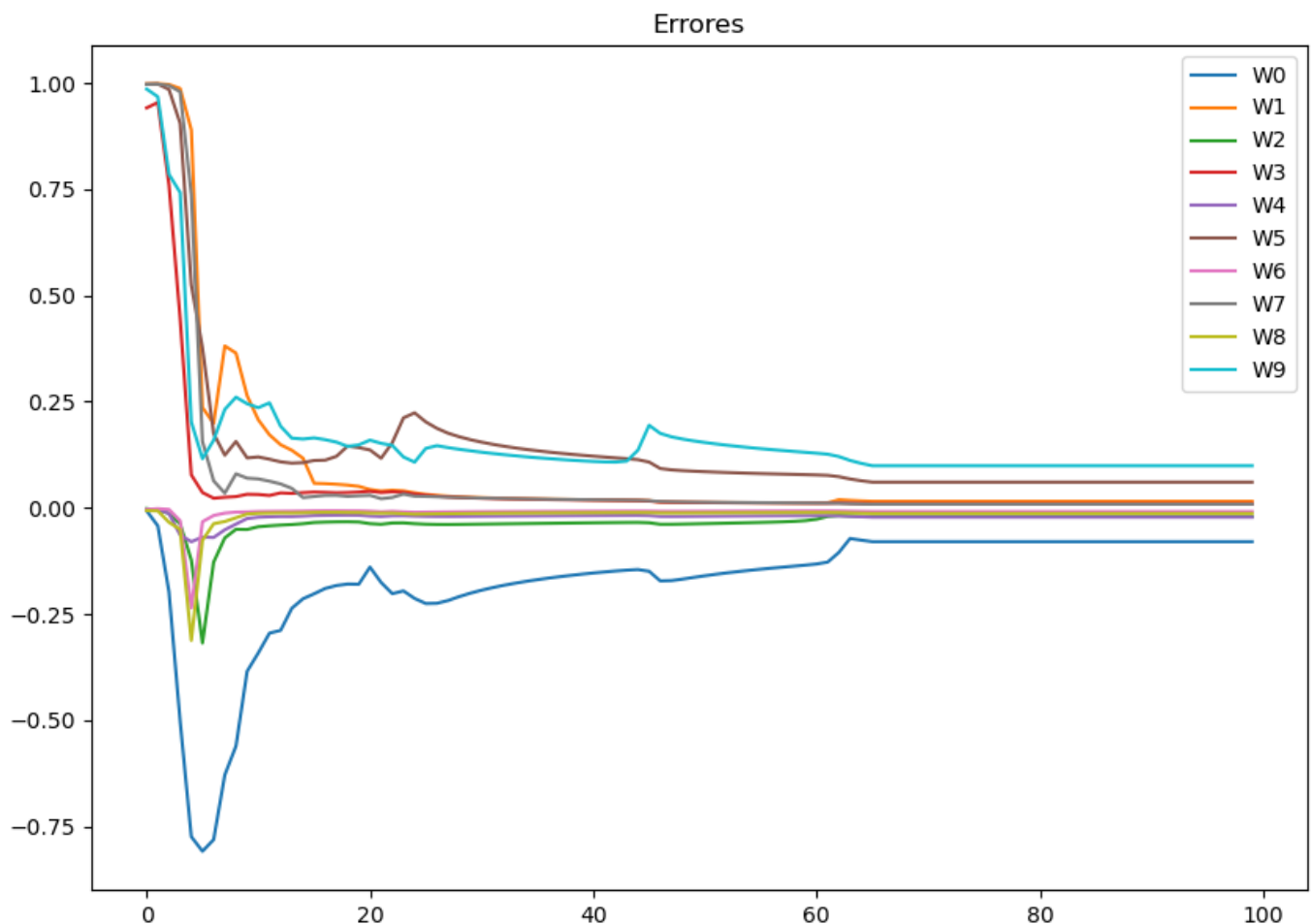
Salida Deseada: 0.000000 Perceptrón salida: 0.008755

Salida Deseada: 1.000000 Perceptrón salida: 0.991533

Salida Deseada: 0.000000 Perceptrón salida: 0.013299

Salida Deseada: 1.000000 Perceptrón salida: 0.900838

A continuación tenemos el siguiente gráfico de error:



Trabajo Práctico N°7

Para el caso que ya se tienen los pesos del sistema luego de “aprender”, al leer las imágenes restantes, se obtienen muy buenos resultados en principio, y con muy pocas iteraciones ya el sistema aprende rápidamente. Se adjuntan los resultados de la primera y la segunda iteración, y el gráfico de error, el cual se observa que rápidamente deja de variar debido a que es muy pequeño el error.

Cantidad de iteraciones: 1

Salida Deseada: 0.000000 Perceptron salida: 0.012462

Salida Deseada: 1.000000 Perceptron salida: 0.999600

Salida Deseada: 0.000000 Perceptron salida: 0.015948

Salida Deseada: 1.000000 Perceptron salida: 0.980043

Salida Deseada: 0.000000 Perceptron salida: 0.001766

Salida Deseada: 1.000000 Perceptron salida: 0.771351

Cantidad de iteraciones: 2

Salida Deseada: 0.000000 Perceptron salida: 0.031465

Salida Deseada: 1.000000 Perceptron salida: 0.999602

Salida Deseada: 0.000000 Perceptron salida: 0.020824

Salida Deseada: 1.000000 Perceptron salida: 0.992621

Salida Deseada: 0.000000 Perceptron salida: 0.003648

Salida Deseada: 1.000000 Perceptron salida: 0.900880

