

Estuardo Teleguario 1554809  
Kevin Martínez 1552809

## Entrenamiento con Neuroph

Los datos que se usaran son las siguientes:

Datos de entrada: estas se obtienen a partir de la transformada de Fourier; nos proporciona 6 datos como entrada para la red neuronal.

Tabla 1. Datos que se usaran como entradas a la red neuronal.

Abrir Word	5.03765443	-4.91747495	1.16557777	2.54495674	-5.87675429	2.55821221
Abrir CMD (terminal)	0.80438469	-1.70752047	-0.11393426	1.31285914	-1.86314206	0.02671551
Abrir Solitario	4.69720506	-1.77711417	-1.20990075	0.71086019	-0.64044225	4.48441995
Abrir Excel	2.04424812	-4.69246883	3.29483815	3.26663668	-2.38758877	1.87432863

En la primera columna están las palabras dichas, y la fila completa representa la transformación del audio a datos, en este caso números, que pueda ser procesada por la red neuronal.

Datos de salida: por la naturaleza del proyecto, se pensó en asignar a cada salida de audio un arreglo de bits como: 1000 como condicionantes, para proceder a abrir cierto programa.

Tabla 2. Datos de salida de la red neuronal

Abrir Word	1	0	0	0
Abrir terminal	0	1	0	0
Abrir solitario	0	0	1	0
Abrir Excel	0	0	0	1

## Herramienta para entrenamiento

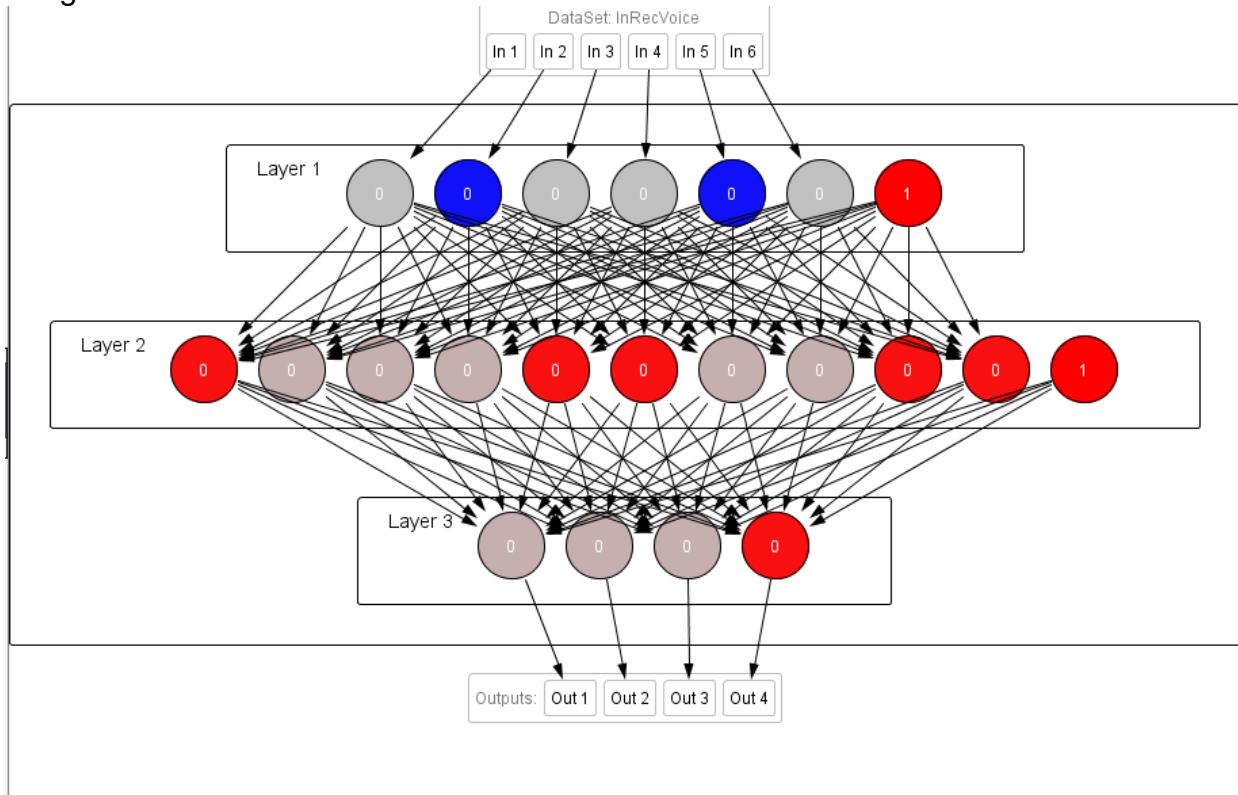
Se utilizó Neuroph Studio para realizar los entrenamientos de las cuatro muestras de audio, quedando estructurado de la siguiente manera:

Backpropagation (Perceptron Multilayer): como algoritmo de aprendizaje supervisado.

Capas y capas ocultas: en total son 3 capas, 1 capa de entrada con 6 neuronas, 1 capa oculta de 10 neuronas, y una capa de salida con 4 neuronas.

Las capas ocultas fueron de 10, para que el resultado de las salidas no sean tan exactos, de esta forma si existiera una segunda voz, se puedan realizar rangos de aceptación de una voz o de otras.

Imagen 1. Estructura de la red neuronal.



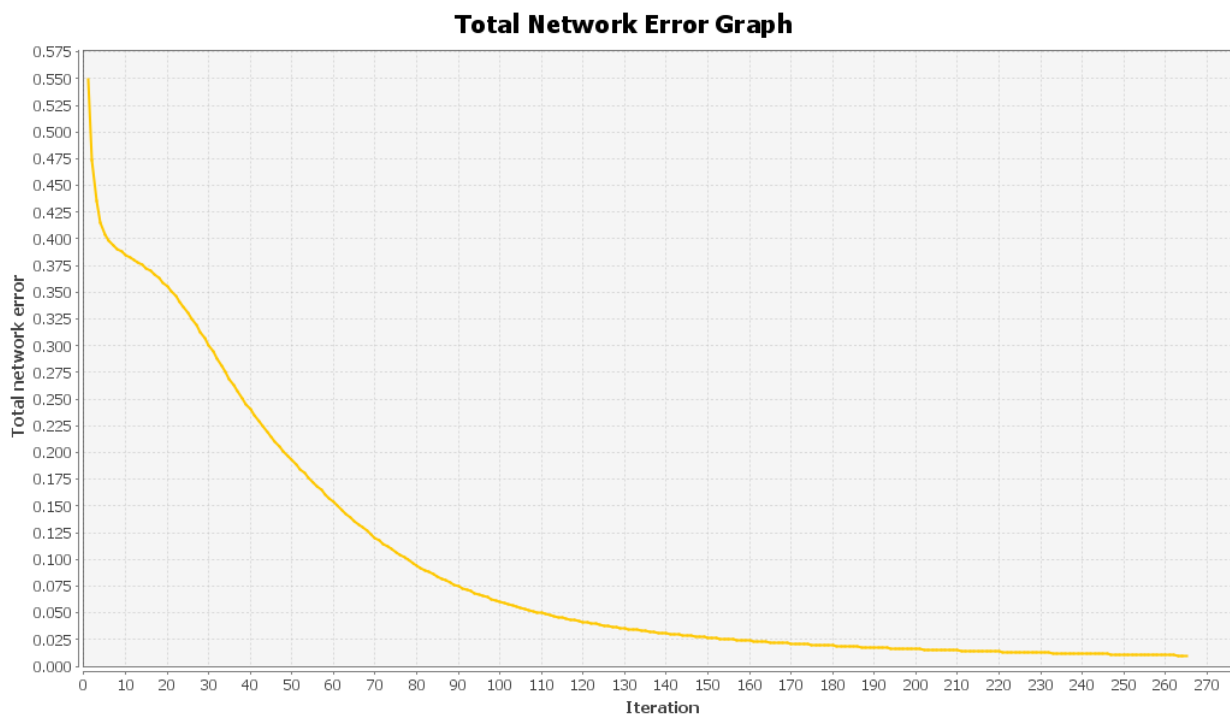
Entrenamiento: para el entrenamiento se utilizó un Error máximo de 0.01 y una constante de aprendizaje de 0.2. Dando un buen resultado de entrenamiento:

### Imagen 2. Resultado del entrenamiento.

Input: 5.0377; -4.9175; 1.1656; 2.545; -5.8768; 2.5582; Output: 0.8569; 0.0877; 0.0171; 0.0354; Desired output: 1; 0; 0; 0; Error: -0.1431; 0.0877; 0.0171; 0.0354;  
 Input: 0.8044; -1.7075; -0.1139; 1.3129; -1.8631; 0.0267; Output: 0.1119; 0.8968; 0.0128; 0.053; Desired output: 0; 1; 0; 0; Error: 0.1119; -0.1032; 0.0128; 0.053;  
 Input: 4.6972; -1.7771; -1.2099; 0.7109; -0.6404; 4.4844; Output: 0.0398; 0.0127; 0.9424; 0.0368; Desired output: 0; 0; 1; 0; Error: 0.0398; 0.0127; -0.0576; 0.0368;  
 Input: 2.0442; -4.6925; 3.2948; 3.2666; -2.3876; 1.8743; Output: 0.0592; 0.0221; 0.0506; 0.9151; Desired output: 0; 0; 0; 1; Error: 0.0592; 0.0221; 0.0506; -0.0849;  
 Total Mean Square Error: 0.004751953159676957

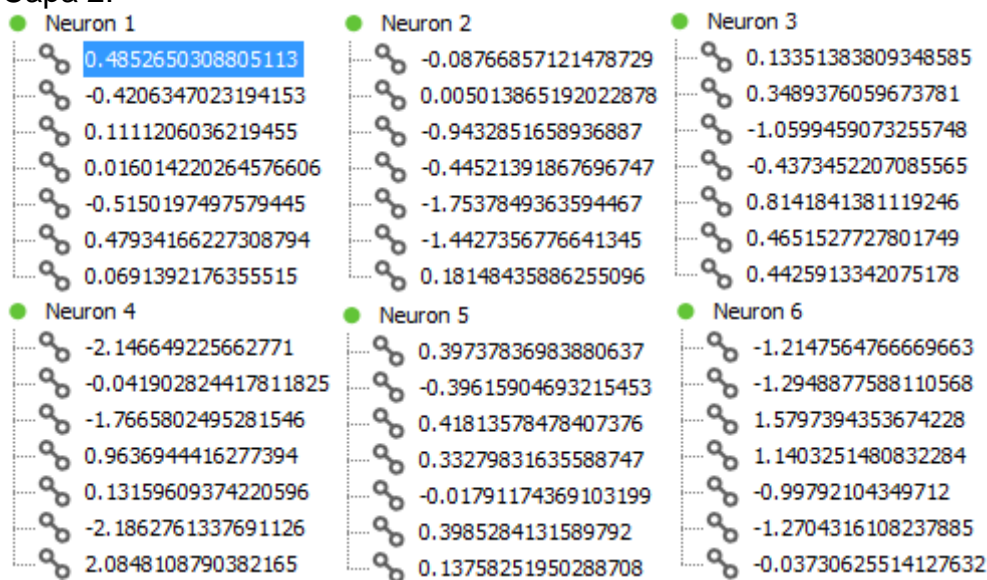
En la imagen 2 se puede observar que las salidas están en un rango aceptable, arriba de 0.8, tomando en cuenta que solo se entrenó una sola voz diciendo varias palabras "comandos", para poder abrir un programa.

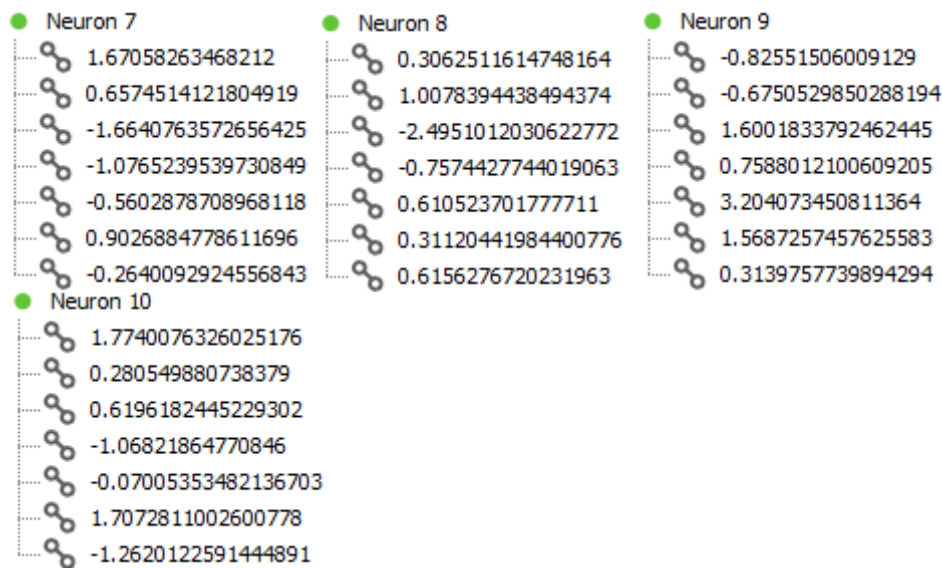
La gráfica del entrenamiento se comportó de la siguiente manera:



Pesos de las neuronas:

Capa 2:





### Capa 3:

