**Pasos para crear una red neuronal de perceptrón de múltiples capas.**

* Se comienza creando un nuevo proyecto de Neuroph que sería: (*File -> New Project)* seleccionando la carpeta (*Neuroph*) y el Project (*Neuroph Project*), después se le agrega un nombre (*ejemplo: PrediccionFutbol*)
* Se debe crear un nuevo archivo en (*File -> New File*), (*Categories: Neuroph -> File Types: Neural Network*) selecciona la neuronal que es (*Multi Layer Perceptron*) y le agrega un nombre.
* En Setting multi layer perceptron’s parameters : Debe tener la cantidad de unidades de entrada y salida que se definen en un problema dado. Se debe tener en cuenta que al agregar muchas capas ocultas no se puede moldear los datos y si son pocas, el entrenamiento se hara excesivamente largo. Ademas se selecciona “**Usar bias neuron**” las neuronas de sesgo se agregan a las redes neuronales para ayudar a aprender patrones. En nuestro conjunto de datos, los valores están en el intervalo entre 0 y 1, por lo que utilizamos la función de transferencia **Sigmoid**. Como regla de aprendizaje, elija **Backpropagation With Momentum.**Backpropagation Con el algoritmo Momentum muestra una tasa de convergencia mucho más alta que el algoritmo Backpropagation. Al finalizar se puede ver la red neuronal como un grafico.
* Para crear un conjunto de datos se debe crear un nuevo archivo en (*File -> New File*), (*Categories: Neuroph -> File Types: Data set*). Se debe ingresar un nombre y tener en cuenta las entradas y salidas asignadas por el problema, también saber el tipo de entrenamiento que son supervisados ​​y no supervisados. En el aprendizaje supervisado, el usuario de la red recopila un conjunto de datos de entrenamiento. Los datos de entrenamiento contienen ejemplos de entradas junto con los resultados correspondientes, y la red aprende a inferir la relación entre los dos. En otras palabras, el aprendizaje supervisado se usa para la clasificación. Para una regla de aprendizaje no supervisada, el conjunto de entrenamiento consiste solo en patrones de entrenamiento de entrada. El aprendizaje no supervisado, por otro lado, se utiliza para agrupar. Al finalizar se dirige al **Data set** creado y crea una tabla de datos que se le pueda a las entradas y salidas de la red neuronal.
* Para ver la grafica de Error debemos arrastrar el archivo **Data.tset** a las entradas de la red neuronal al cargar los datos en la red, se pone en entrenamiento. En la parte superior izquierda aparece un icono que dice (*Train*) al dar click aparece el análisis y grafica de Total Network Error Graph.
* Lo siguiente es poner a prueba la red, en la parte superior izquierda aparece un icono que dice (*Test*). El resultado nos muestra que el error cuadrático, errores individuales con esta información se puede concluir que una red puede ser buena o no.