Inicio:

//Pedimos los valores necesarios para el MLP al usuario

archivoEntradas = entradaUsuario;

archivoTargets = entradaUsuario;

archivoArquitectura = entradaUsuario;

archivoFunciones = entradaUsuario;

epochMax = entradaUsuario;

errorEpoca = entradaUsuario;

alpha = entradaUsuario;

numval = entradaUsuario;

epocaEvaluacion = entradaUsuario;

Para cada capa en archivoArquitectura:

//Inicializamos los pesos y bias

//R en la primera capa es el valor de la entrada y posteriormente es el número de neuronas de la capa anterior

Pesos(capa) = NuevaMatriz( valor x R );

bias(capa) = NuevaMatriz(valor x 1);

//Separamos el DataSet en tres conjuntos

[conjuntoEntrenamiento, targetEntrenamiento] = separaDatos(archivoEntradas, archivoTargets);

[conjuntoValidacion, targetEntrenamiento] = separaDatos(archivoEntradas, archivoTargets);

[conjuntoPrueba, targetPrueba] = separaDatos(archivoEntradas, archivoTargets);

//iniciamos con el aprendizaje

contador = 0;

Mientras contador<epochMax:

Si ( contador%epocaEvalucacion==0 && contador!=0):

Evaluacion();

Sino:

error =0;

//Propagamos hacia adelante los datos del conjunto de entrenamiento

Para cada dato en conjuntoEntrenamiento:

Para cada capa en 1:|archivoArquitectura|

if funcion(capa)==1:

//salidas{capa}=purelin(pesos(capa)\*entradas{capa} + bias{capa})

elseif funcion(capa)==2:

//salidas{capa}=logsig(pesos(capa)\*entradas{capa} + bias{capa})

else

//salidas{capa}=tansig(pesos(capa)\*entradas{capa} + bias{capa})

error+=| (target(dato)-salidas(ultimo) |;

backPropagation();

guardarDatos(pesos,bias);

Si error< errorEpoca:

termina;

contador+=1;