

Maestría en Ciencia de Datos

Minería de Datos

Proyecto 01: Regresión multivariada no lineal

Profesor:

Dr. Ángel Fernando Kuri Morales

Alumna:

Gabriela Flores Bracamontes

Clave única:

160124

México, D.F. 27 de octubre de 2015.

Contenido

1. Antecedentes	3
2. Metas del problema asociado a resolver	3
3. Proyecto	3
3.1. Plataforma tecnológica	3
3.2. Descripción de los datos.....	3
3.3. Fases del proyecto.....	5
3.3.1. Descargar la base de datos.....	5
3.3.2. Análisis de los datos:	5
3.3.3. Preprocesamiento:	5
3.3.3.1. Creación de las bases de datos dbf	7
3.3.3.2. Categorizar	8
3.3.3.3. Escalar	9
3.3.3.4. Estabilizar	10
3.3.3.5. Correlacionar	10
3.3.3.6. Convertir los archivos en formato dbf a txt	12
3.3.4. Red Neuronal Perceptrón Multicapa con algoritmo Back Propagation:.....	12
3.3.4.1. Convertir los archivos en formato txt a dat	12
3.3.4.2. Determinamos las neuronas para la capa oculta	13
3.3.4.3. Crear dos redes neuronal de perceptrones multicapa	14
3.3.4.4. Configurar los parámetros para las 2 redes	15
4. Comparativo de resultados	18

1. Antecedentes

En un estudio, se ha identificado la localización de 1484 proteínas en levadura.

Se requiere predecir la localización celular de estas proteínas, por lo que se entrenará una red neuronal para que aprenda a validar la localización de las proteínas en base a las variables proporcionadas en la base de datos.

2. Metas del problema asociado a resolver

Obtener un modelo de clasificación no lineal aplicado al problema de la localización de las proteínas, utilizando el entrenamiento de una red neuronal de perceptrón multicapa mediante el algoritmo back-propagation.

3. Proyecto

3.1. Plataforma tecnológica

Para realizar el presente proyecto se instaló una máquina virtual con Sistema Operativo Windows XP a 32 bits, ya que las herramientas utilizadas solamente funcionan en Sistemas Operativos Windows a 32 bits:

- PREPROC versión 9.2.- Se utilizó para realizar el preprocesamiento de los datos.
- DATA ENGINE versión 2.10.012.- Se utilizó para crear y entrenar la red neuronal de perceptrones multicapa.

3.2. Descripción de los datos.

La base de datos Localización de los sitios de las proteínas, se descargó del repositorio de Aprendizaje Máquina de la UCI en la siguiente ruta:

- <https://archive.ics.uci.edu/ml/datasets/Yeast>

1. Title: Protein Localization Sites

2. Creator and Maintainer:

Kenta Nakai

Institute of Molecular and Cellular Biology

Osaka, University

1-3 Yamada-oka, Suita 565 Japan

nakai@imcb.osaka-u.ac.jp

<http://www.imcb.osaka-u.ac.jp/nakai/psort.html>

Donor: Paul Horton (paulh@cs.berkeley.edu)

Date: September, 1996

See also: ecoli database

3. Past Usage.

Reference: "A Probabilistic Classification System for Predicting the Cellular Localization Sites of Proteins", Paul Horton & Kenta Nakai, Intelligent Systems in Molecular Biology, 109-115. St. Louis, USA 1996.

Results: 55% for Yeast data with an ad hoc structured probability model. Also similar accuracy for Binary Decision Tree and Bayesian Classifier methods applied by the same authors in unpublished results.

Predicted Attribute: Localization site of protein. (non-numeric).

4. The references below describe a predecessor to this dataset and its development. They also give results (not cross-validated) for classification by a rule-based expert system with that version of the dataset.

Reference: "Expert System for Predicting Protein Localization Sites in Gram-Negative Bacteria", Kenta Nakai & Minoru Kanehisa, PROTEINS: Structure, Function, and Genetics 11:95-110, 1991.

Reference: "A Knowledge Base for Predicting Protein Localization Sites in Eukaryotic Cells", Kenta Nakai & Minoru Kanehisa, Genomics 14:897-911, 1992.

5. Number of Instances: 1484 for the Yeast dataset.

6. Number of Attributes.
for Yeast dataset: 9 (8 predictive, 1 name)

7. Attribute Information.

1. Sequence Name: Accession number for the SWISS-PROT database
2. mcg: McGeoch's method for signal sequence recognition.
3. gvh: von Heijne's method for signal sequence recognition.
4. alm: Score of the ALOM membrane spanning region prediction program.
5. mit: Score of discriminant analysis of the amino acid content of the N-terminal region (20 residues long) of mitochondrial and non-mitochondrial proteins.
6. erl: Presence of "HDEL" substring (thought to act as a signal for retention in the endoplasmic reticulum lumen). Binary attribute.
7. pox: Peroxisomal targeting signal in the C-terminus.
8. vac: Score of discriminant analysis of the amino acid content of vacuolar and extracellular proteins.

9. nuc: Score of discriminant analysis of nuclear localization signals of nuclear and non-nuclear proteins.

8. Missing Attribute Values: None.

9. Class Distribution. The class is the localization site. Please see Nakai & Kanehisa referenced above for more details.

CYT (cytosolic or cytoskeletal)	463
NUC (nuclear)	429
MIT (mitochondrial)	244
ME3 (membrane protein, no N-terminal signal)	163
ME2 (membrane protein, uncleaved signal)	51
ME1 (membrane protein, cleaved signal)	44
EXC (extracellular)	37
VAC (vacuolar)	30
POX (peroxisomal)	20
ERL (endoplasmic reticulum lumen)	5

3.3. Fases del proyecto

3.3.1. Descargar la base de datos

La base de datos se descargó de la página señalada en la sección “Descripción de los datos” y se guardó en un archivo con formato TXT.

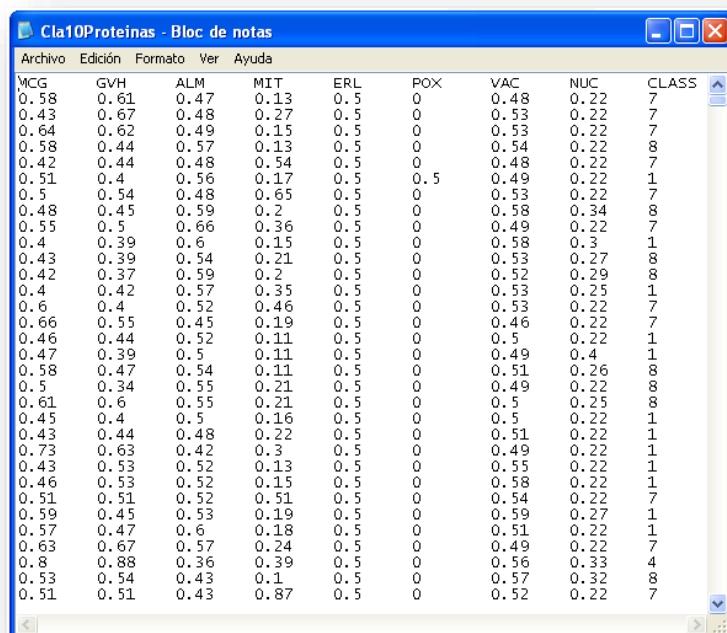
3.3.2. Análisis de los datos:

- Los datos venían separados por espacios y se remplazaron por tabuladores.
- El primer campo denominado **Nombre de Secuencia** es solamente una etiqueta que no aporta mayor valor a los datos por lo que procedí a eliminarla.
- La variable ERL se trató como variable categórica, ya que solamente utiliza 2 valores: 0.5 y 1
- El último campo denominado **Clase** es la variable dependiente y aunque es una variable categórica, se trató como una variable numérica.

3.3.3. Preprocesamiento:

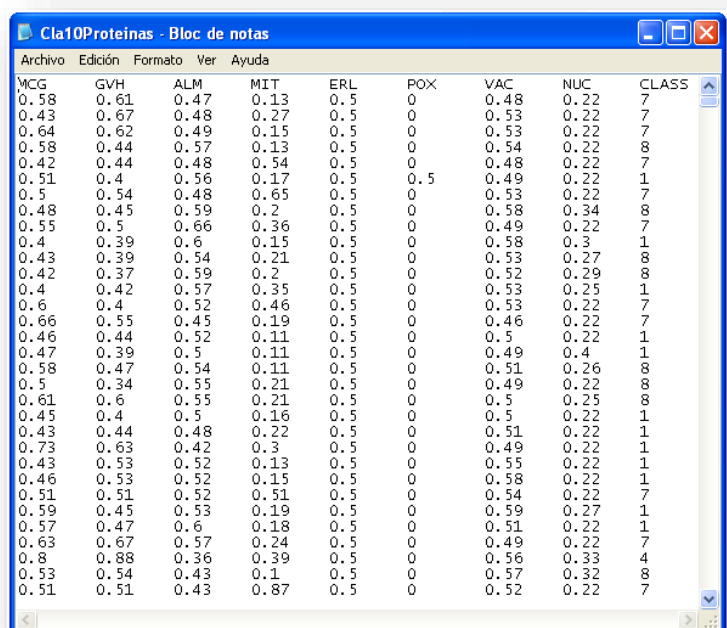
Para el Preprocesamiento de los datos se utilizaron 2 archivos:

- **Cla10Proteinas.txt** - La variable Clase utiliza valores entre 1-10



VCG	GVH	ALM	MIT	ERL	POX	VAC	NUC	CLASS
0.58	0.61	0.47	0.13	0.5	0	0.48	0.22	7
0.43	0.67	0.48	0.27	0.5	0	0.53	0.22	7
0.64	0.62	0.49	0.15	0.5	0	0.53	0.22	7
0.58	0.44	0.57	0.13	0.5	0	0.54	0.22	8
0.42	0.44	0.48	0.54	0.5	0	0.48	0.22	7
0.51	0.4	0.56	0.17	0.5	0.5	0.49	0.22	1
0.5	0.54	0.48	0.65	0.5	0	0.53	0.22	7
0.48	0.45	0.59	0.2	0.5	0	0.58	0.34	8
0.55	0.5	0.66	0.36	0.5	0	0.49	0.22	7
0.4	0.39	0.6	0.15	0.5	0	0.58	0.3	1
0.43	0.39	0.54	0.21	0.5	0	0.53	0.27	8
0.42	0.37	0.59	0.2	0.5	0	0.52	0.29	8
0.4	0.42	0.57	0.35	0.5	0	0.53	0.25	1
0.6	0.4	0.52	0.46	0.5	0	0.53	0.22	7
0.66	0.55	0.45	0.19	0.5	0	0.46	0.22	7
0.46	0.44	0.52	0.11	0.5	0	0.5	0.22	1
0.47	0.39	0.5	0.11	0.5	0	0.49	0.4	1
0.58	0.47	0.54	0.11	0.5	0	0.51	0.26	8
0.5	0.34	0.55	0.21	0.5	0	0.49	0.22	8
0.61	0.6	0.55	0.21	0.5	0	0.5	0.25	8
0.45	0.4	0.5	0.16	0.5	0	0.5	0.22	1
0.43	0.44	0.48	0.22	0.5	0	0.51	0.22	1
0.73	0.63	0.42	0.3	0.5	0	0.49	0.22	1
0.43	0.53	0.52	0.13	0.5	0	0.55	0.22	1
0.46	0.53	0.52	0.15	0.5	0	0.58	0.22	1
0.51	0.51	0.52	0.51	0.5	0	0.54	0.22	7
0.59	0.45	0.53	0.19	0.5	0	0.59	0.27	1
0.57	0.47	0.6	0.18	0.5	0	0.51	0.22	1
0.63	0.67	0.57	0.24	0.5	0	0.49	0.22	7
0.8	0.88	0.36	0.39	0.5	0	0.56	0.33	4
0.53	0.54	0.43	0.1	0.5	0	0.57	0.32	8
0.51	0.51	0.43	0.87	0.5	0	0.52	0.22	7

- **Cla05Proteinas.txt** - La variable Clase utiliza valores entre 1 – 5. Las variables menos representativas como son ME2, ME1, EXC, VAC, POX y ERL se agruparon en una sola categoría.



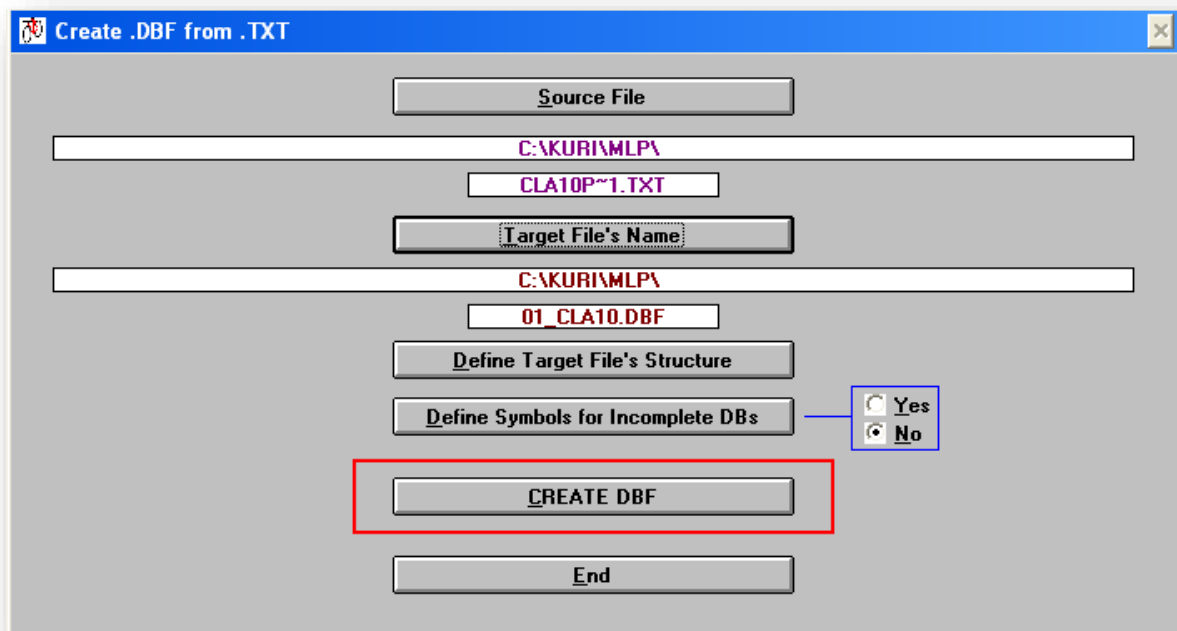
VCG	GVH	ALM	MIT	ERL	POX	VAC	NUC	CLASS
0.58	0.61	0.47	0.13	0.5	0	0.48	0.22	7
0.43	0.67	0.48	0.27	0.5	0	0.53	0.22	7
0.64	0.62	0.49	0.15	0.5	0	0.53	0.22	7
0.58	0.44	0.57	0.13	0.5	0	0.54	0.22	8
0.42	0.44	0.48	0.54	0.5	0	0.48	0.22	7
0.51	0.4	0.56	0.17	0.5	0.5	0.49	0.22	1
0.5	0.54	0.48	0.65	0.5	0	0.53	0.22	7
0.48	0.45	0.59	0.2	0.5	0	0.58	0.34	8
0.55	0.5	0.66	0.36	0.5	0	0.49	0.22	7
0.4	0.39	0.6	0.15	0.5	0	0.58	0.3	1
0.43	0.39	0.54	0.21	0.5	0	0.53	0.27	8
0.42	0.37	0.59	0.2	0.5	0	0.52	0.29	8
0.4	0.42	0.57	0.35	0.5	0	0.53	0.25	1
0.6	0.4	0.52	0.46	0.5	0	0.53	0.22	7
0.66	0.55	0.45	0.19	0.5	0	0.46	0.22	7
0.46	0.44	0.52	0.11	0.5	0	0.5	0.22	1
0.47	0.39	0.5	0.11	0.5	0	0.49	0.4	1
0.58	0.47	0.54	0.11	0.5	0	0.51	0.26	8
0.5	0.34	0.55	0.21	0.5	0	0.49	0.22	8
0.61	0.6	0.55	0.21	0.5	0	0.5	0.25	8
0.45	0.4	0.5	0.16	0.5	0	0.5	0.22	1
0.43	0.44	0.48	0.22	0.5	0	0.51	0.22	1
0.73	0.63	0.42	0.3	0.5	0	0.49	0.22	1
0.43	0.53	0.52	0.13	0.5	0	0.55	0.22	1
0.46	0.53	0.52	0.15	0.5	0	0.58	0.22	1
0.51	0.51	0.52	0.51	0.5	0	0.54	0.22	7
0.59	0.45	0.53	0.19	0.5	0	0.59	0.27	1
0.57	0.47	0.6	0.18	0.5	0	0.51	0.22	1
0.63	0.67	0.57	0.24	0.5	0	0.49	0.22	7
0.8	0.88	0.36	0.39	0.5	0	0.56	0.33	4
0.53	0.54	0.43	0.1	0.5	0	0.57	0.32	8
0.51	0.51	0.43	0.87	0.5	0	0.52	0.22	7

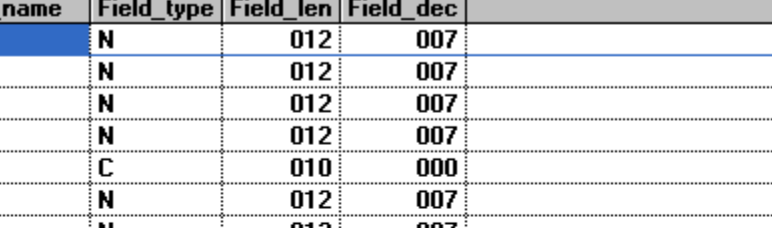
3.3.3.1. Creación de las bases de datos dbf

Utilizamos la opción **DBF_TXT** del programa PREPROC, para crear las bases de datos: 01_CLA10.DBF y 01_CLA05.DBF.



Seleccionamos el archivo fuente en este caso Cla10Proteinas.txt y el nombre de la base de datos, en este caso 01_CLA10.DBF. Se siguió el mismo procedimiento para la base de datos 01_CLA05.DBF que tiene las 5 categorías.



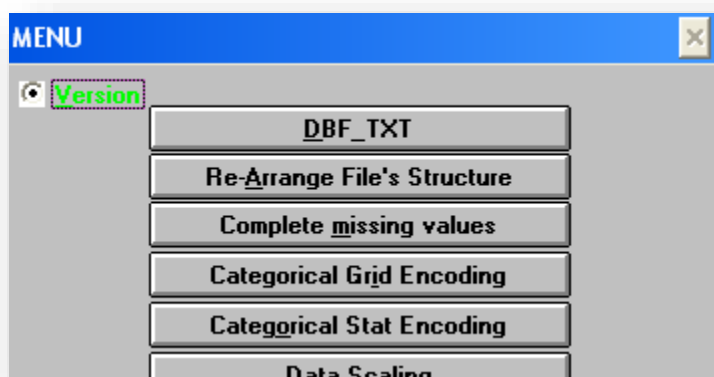


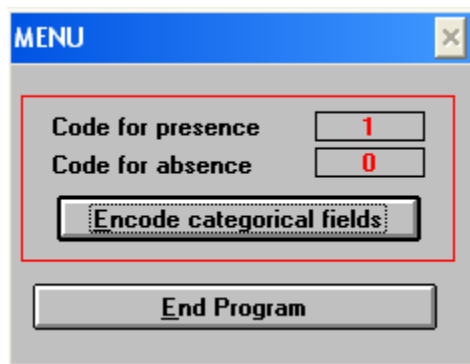
The screenshot shows a window titled "DBF Structure" with a table of field definitions. The table has four columns: Field_name, Field_type, Field_len, and Field_dec. The fields listed are MCG, GYH, ALM, MIT, ERL, POX, VAC, NUC, and CLASS. The first four fields (MCG, GYH, ALM, MIT) are numeric (N) with a length of 012 and 7 decimal places (007). The fifth field (ERL) is character (C) with a length of 010 and 0 decimal places (000). The last four fields (POX, VAC, NUC, CLASS) are numeric (N) with a length of 012 and 7 decimal places (007).

Field_name	Field_type	Field_len	Field_dec
MCG	N	012	007
GYH	N	012	007
ALM	N	012	007
MIT	N	012	007
ERL	C	010	000
POX	N	012	007
VAC	N	012	007
NUC	N	012	007
CLASS	N	012	007

3.3.3.2. Categorizar

El siguiente paso fue codificar la variable categórica ERL y convertirla en pseudo-binarias, utilizando la opción Categorical Grid Encoding, esto para las 2 bases de datos.



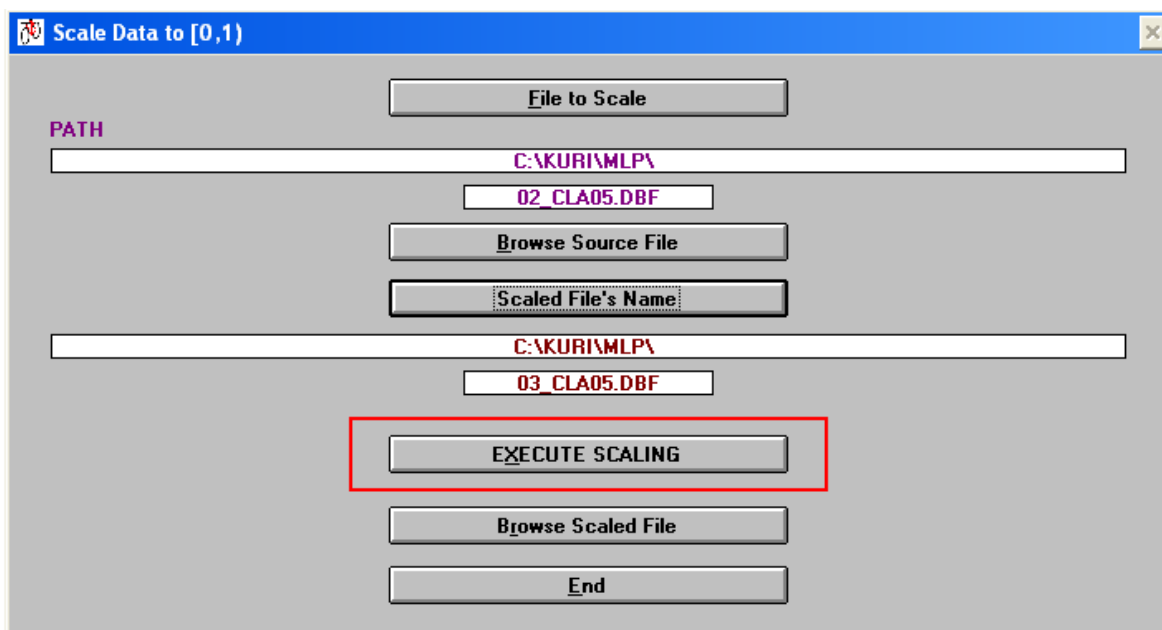


De este procedimiento se generan las siguientes bases de datos:

- 02_CLA10.DBF
- 02_CLA05.DBF

3.3.3.3. Escalar

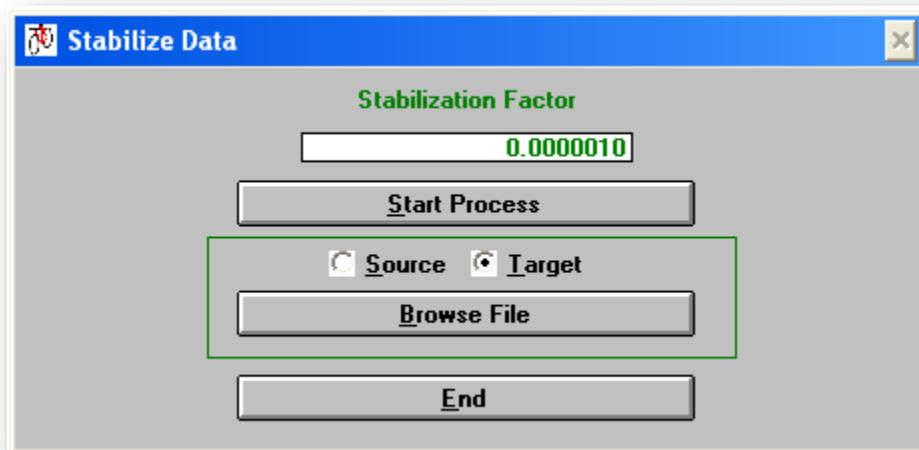
Los datos son escalados entre 0,1:



De este procedimiento se generan las siguientes bases de datos:

- 03_CLA10.DBF
- 03_CLA05.DBF

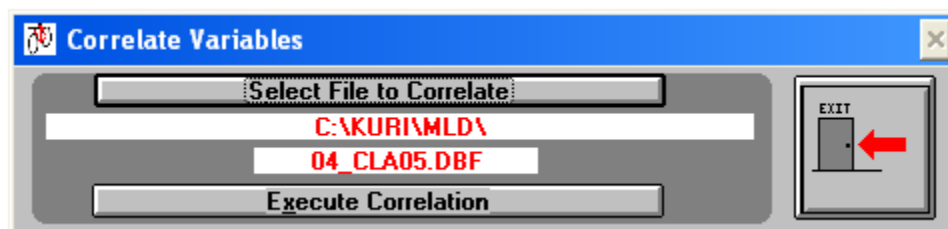
3.3.3.4. Estabilizar



De este procedimiento se generan las siguientes bases de datos:

- 04_CLA10.DBF
- 04_CLA05.DBF

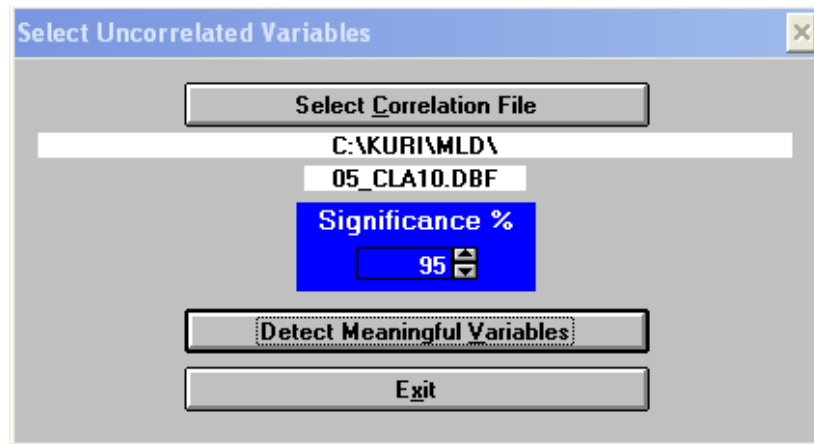
3.3.3.5. Correlacionar



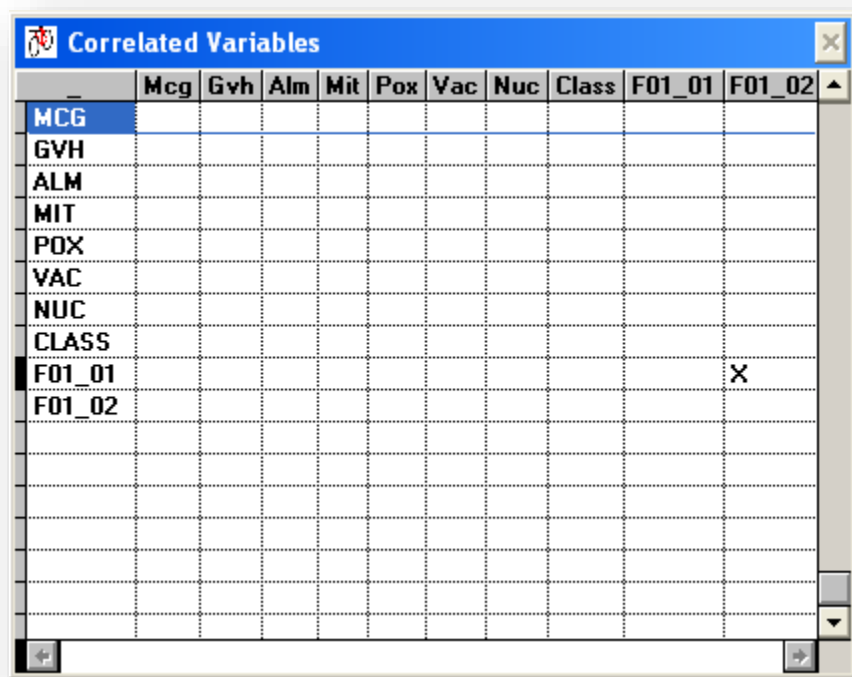
De este procedimiento se generan las siguientes bases de datos:

- 05_CLA10.DBF
- 05_CLA05.DBF

Ahora buscamos la variables correlacionadas al 95 %



Las variables pseudobinarias F01_01 y F01_02 se correlacionan al 95%, por lo que se elimina la variable F01_02, ya que con una sola representa el valor de las 2.



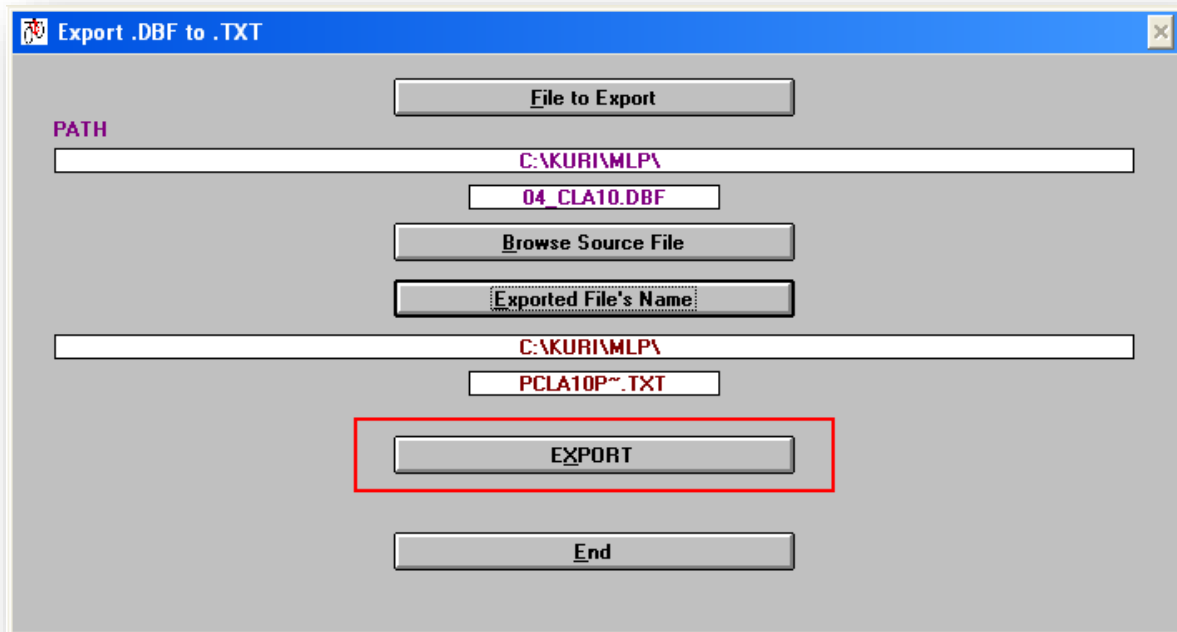
The "Correlated Variables" dialog box displays a table with the following variables and correlation status:

	Mcg	Gvh	Alm	Mit	Pox	Vac	Nuc	Class	F01_01	F01_02
McG										
GVH										
ALM										
MIT										
POX										
VAC										
NUC										
CLASS										
F01_01										X
F01_02										

Eliminamos la variable F01_02 y ponemos la variable dependiente **Clase** al final.

3.3.3.6. Convertir los archivos en formato dbf a txt

Por último exportamos la base de datos a un archivo en formato TXT.



De este procedimiento se generan los siguientes archivos:

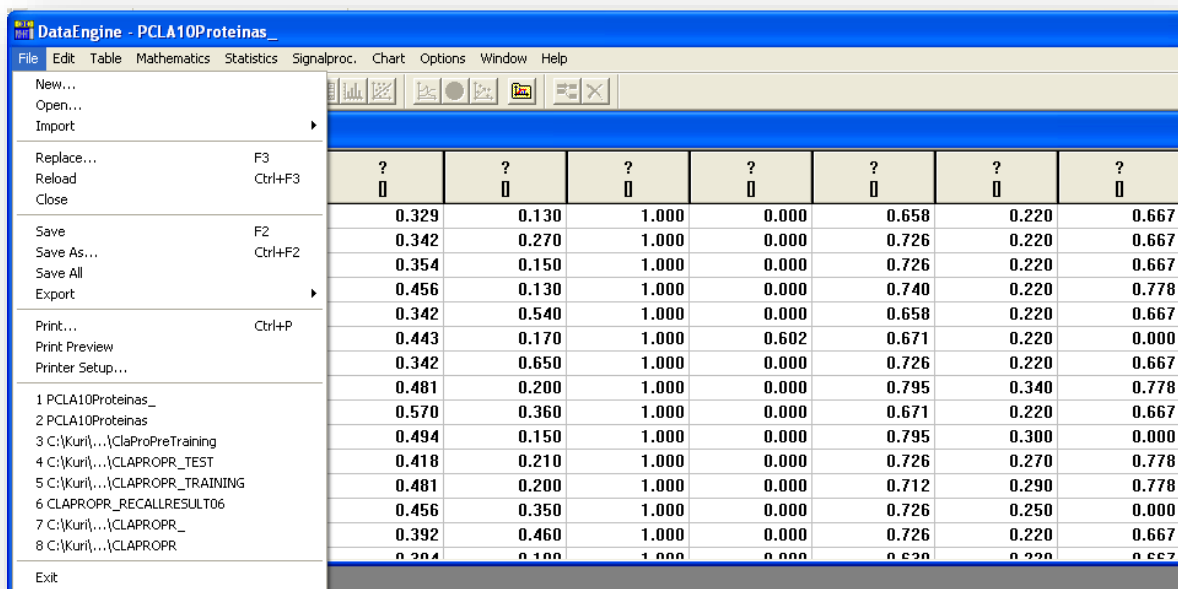
- PCL10Proteinas.txt
- PCL05Proteinas.txt

3.3.4. Red Neuronal Perceptrón Multicapa con algoritmo Back Propagation:

3.3.4.1. Convertir los archivos en formato txt a dat

El programa Data Engine utiliza archivos en formato .dat por lo que es necesario seleccionar la opción Import del Menú File y seleccionar la opción ASCII.

Una vez que el programa nos muestra el contenido del archivo, seleccionamos la opción Save Ass y guardamos el archivo con el nuevo formato.



The screenshot shows the 'DataEngine - PCLA10Proteinas' application. The 'File' menu is open, displaying options like 'New...', 'Open...', 'Import', 'Replace...', 'Reload', 'Close', 'Save', 'Save As...', 'Save All', 'Export', 'Print...', 'Print Preview', 'Printer Setup...', and a list of files. The main window displays a table with 8 columns and 14 rows of data.

	?	?	?	?	?	?	?
	0.329	0.130	1.000	0.000	0.658	0.220	0.667
	0.342	0.270	1.000	0.000	0.726	0.220	0.667
	0.354	0.150	1.000	0.000	0.726	0.220	0.667
	0.456	0.130	1.000	0.000	0.740	0.220	0.778
	0.342	0.540	1.000	0.000	0.658	0.220	0.667
	0.443	0.170	1.000	0.602	0.671	0.220	0.000
	0.342	0.650	1.000	0.000	0.726	0.220	0.667
	0.481	0.200	1.000	0.000	0.795	0.340	0.778
	0.570	0.360	1.000	0.000	0.671	0.220	0.667
	0.494	0.150	1.000	0.000	0.795	0.300	0.000
	0.418	0.210	1.000	0.000	0.726	0.270	0.778
	0.481	0.200	1.000	0.000	0.712	0.290	0.778
	0.456	0.350	1.000	0.000	0.726	0.250	0.000
	0.392	0.460	1.000	0.000	0.726	0.220	0.667
	0.304	0.100	1.000	0.000	0.620	0.220	0.667

De este procedimiento se generan los siguientes archivos:

- PCL10Proteinas.dat
- PCL05Proteinas.dat

3.3.4.2. Determinamos las neuronas para la capa oculta

Para determinar las neuronas de la capa oculta, tenemos que comprimir los datos y saber el radio de compresión, para lo cual utilizamos nuevamente el programa PREPROC en la opción de PPMZ2 (Regression). Los archivos utilizados son:

- PCL10Proteinas.dat
- PCL05Proteinas.dat

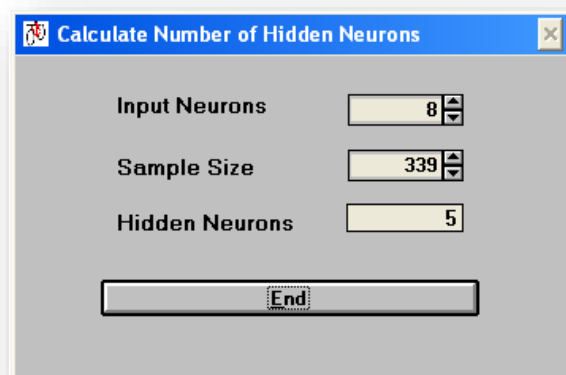
Target File Size (bytes)	40,017
Compression Ratio	4.38000

Tenemos 1,484 observaciones y 8 neuronas de entrada.

Si sabemos que nuestro radio de compresión del archivo es 4.38, obtenemos el siguiente tamaño de la muestra será:

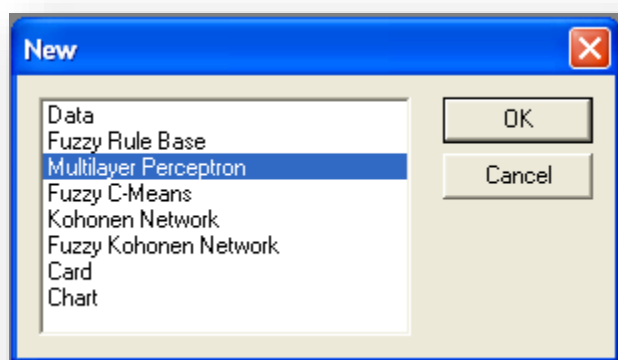
$$1,484 / 4.38 = 339$$

Utilizamos la opción NN Architecture del programa PREPROC y ponemos los valores obtenidos anteriormente.



3.3.4.3. Crear dos redes neuronal de perceptrones multicapa

Abrimos el programa Data Engine, seleccionamos del menú la opción New y seleccionamos Multilayer Perceptron, esto se hace para las dos redes.

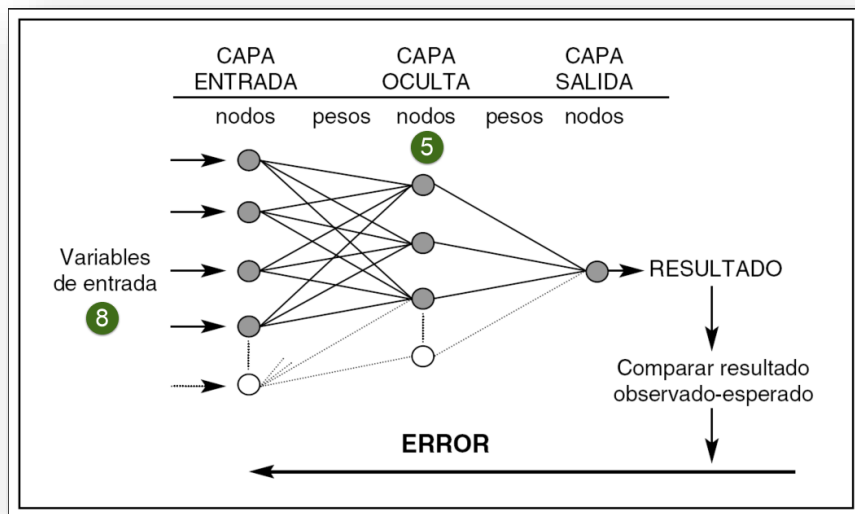


De este procedimiento se generan los siguientes archivos:

- PCL10Proteinas_RedNeuronal.mlp
- PCL05Proteinas_RedNeuronal.mlp

3.3.4.4. Configurar los parámetros para las 2 redes

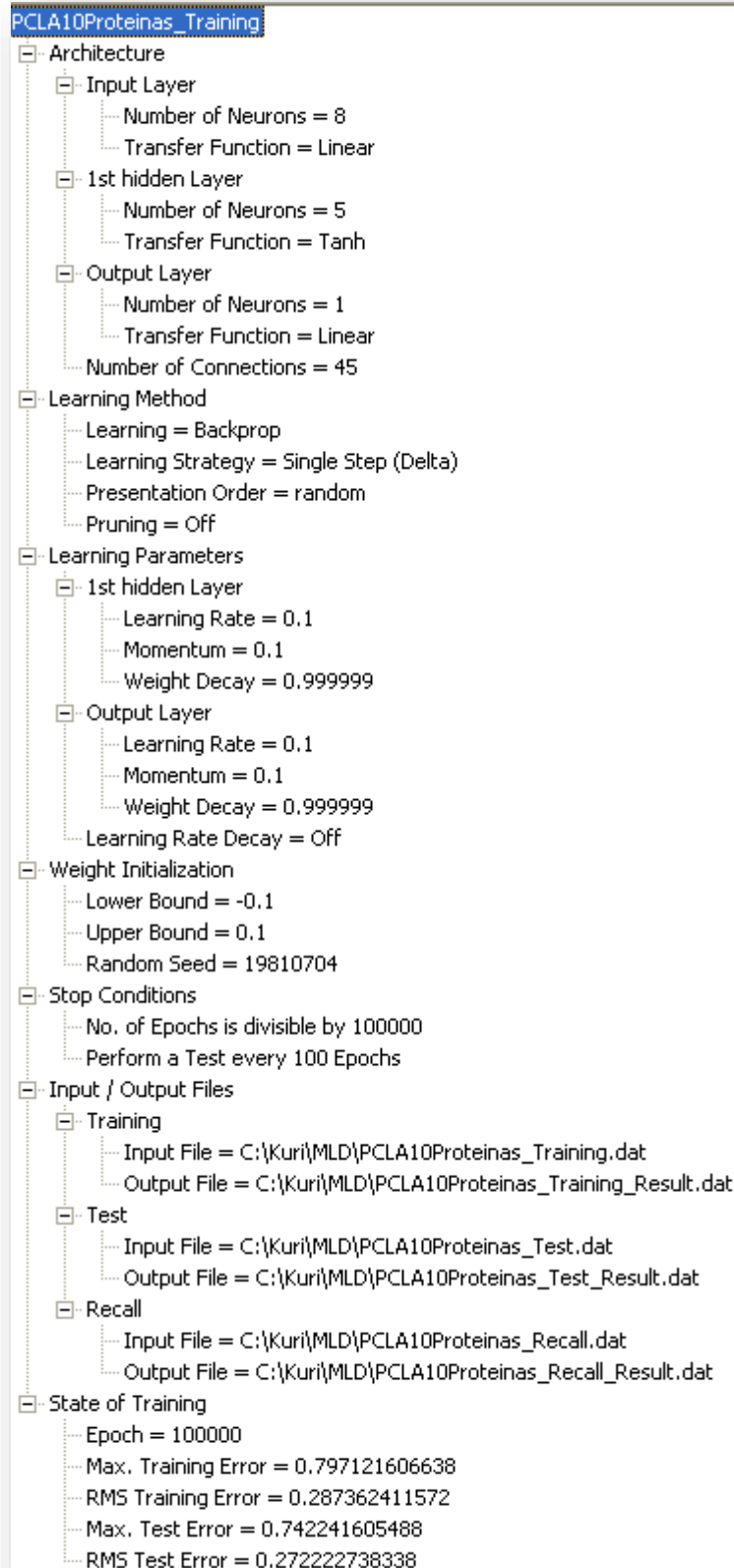
Utilizamos 8 neuronas de entrada, solamente una capa oculta con 5 neuronas, el algoritmo BackPropagation, para el entrenamiento usando 100,000 épocas y el entrenamiento lo detenemos cada 100 épocas para las pruebas.

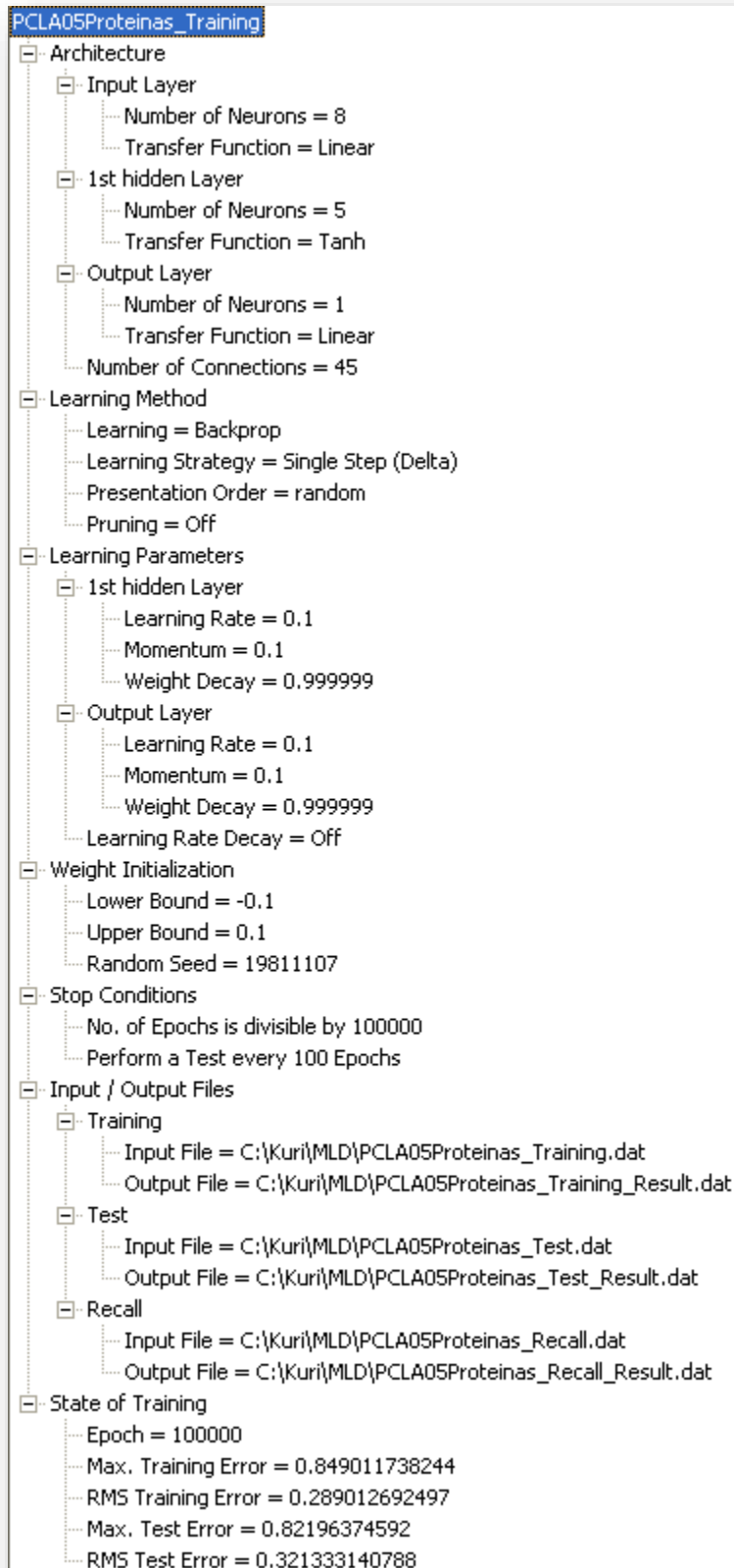


Del 100% de la muestra vamos a generar 3 nuevos archivos, cabe mencionar que se realizó el mismo procedimiento para la red que utiliza las 10 categorías de la variable clase y para la que solo utiliza 5.

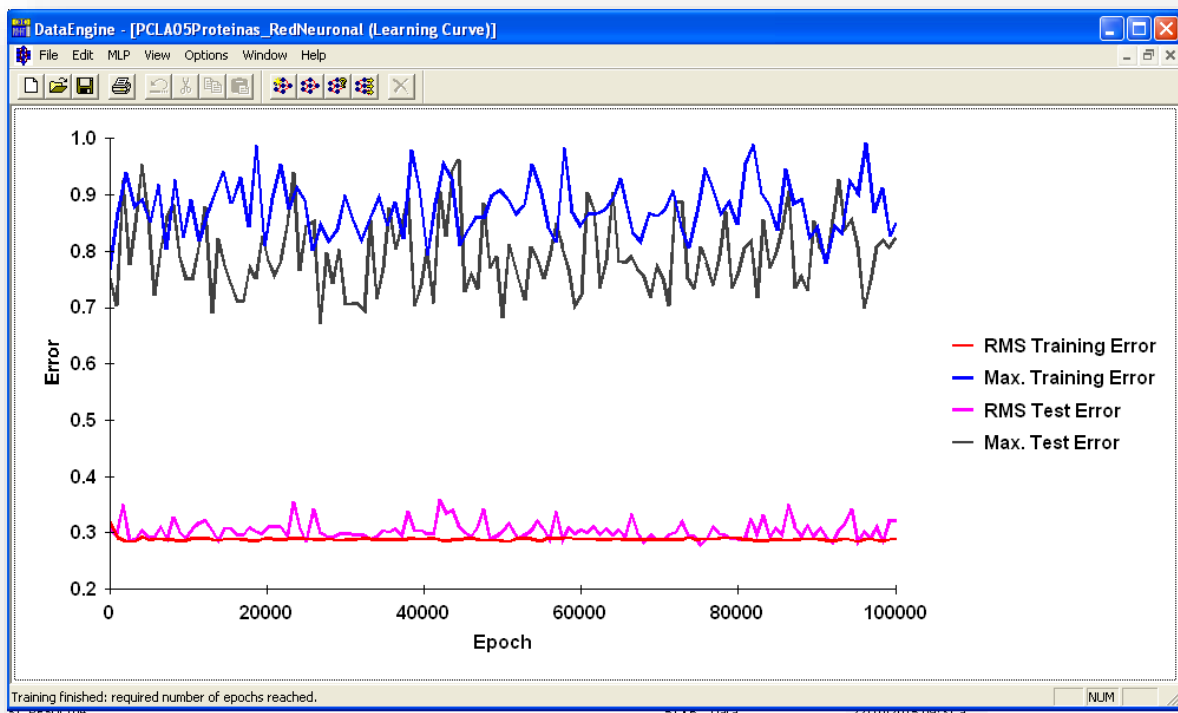
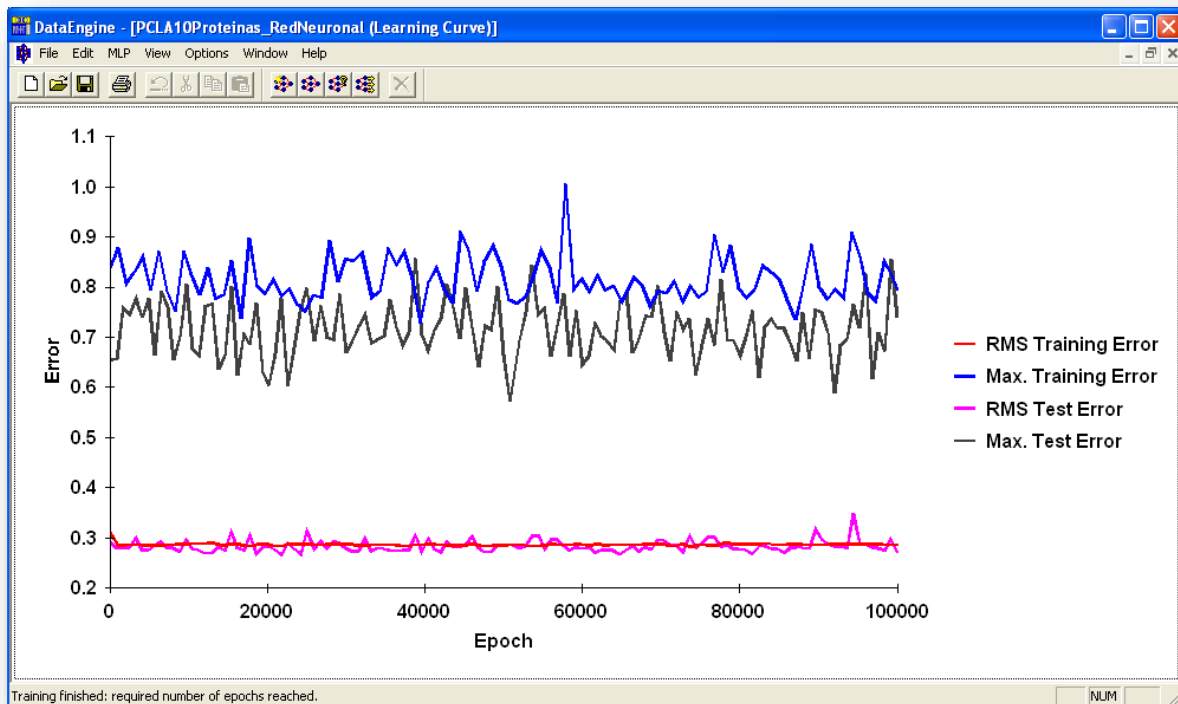
- Tomamos aleatoriamente el 80% y lo guardamos en un archivo que nos servirá para entrenamiento. Los archivos generados son:
 - PCLA10Proteinas_Training.dat
 - PCLA05Proteinas_Training.dat
- El 20 % restante lo guardamos en otro archivo y este nos servirá para realizar las pruebas. Los archivos generados son:
 - PCLA10Proteinas_Test.dat
 - PCLA05Proteinas_Test.dat
- Del mismo 20% restante, eliminamos la variable Clase y lo guardamos en un tercer archivo, que utilizaremos para verificar que tan bueno ha sido el entrenamiento. Los archivos generados son:
 - PCLA10Proteinas_Recall.dat
 - PCLA05Proteinas_Recall.dat

A continuación se muestra la configuración para cada una de las redes.





4. Comparativo de resultados



Abrimos los archivos que se generan con los resultados del entrenamiento, y obtenemos la siguiente información:

- **PCLA10Proteinas_Training_Result.dat.**- El error más pequeño lo encontramos en la época **42,000** y fue de **0.2553** para la red neuronal PCLA10Proteinas_RedNeuronal

	Epoch	RMS_Training_Error	Max_Training_Error	Epoch	RMS_Test_Error	Max_Test_Error	*
53	53,760.000	0.2620441891	1.004	42,000.000	0.2555374928	0.716	
54	54,784.000	0.2643987994	0.971	42,800.000	0.277604768	0.864	
55	55,808.000	0.2641623296	0.881	43,600.000	0.2642231058	0.747	
56	56,832.000	0.2661110382	0.926	44,400.000	0.2699630073	0.762	

- **PCLA05Proteinas_Training_Result.dat.**- El error más pequeño lo encontramos en la época **42,496** y fue de **0.2628** para la red neuronal PCLA05Proteinas_RedNeuronal.

	Epoch	RMS_Training_Error	Max_Training_Error	Epoch	RMS_Test_Error	Max_Test_Error	*
42	42,496.000	0.2628191884	0.873	33,200.000	0.2832192948	0.853	
43	43,520.000	0.2678293618	0.939	34,000.000	0.2773577969	0.834	
44	44,544.000	0.2659996234	0.766	34,800.000	0.2758407987	0.823	
45	45,568.000	0.26896141	0.875	35,600.000	0.2833103025	0.856	