

**IMPLEMENTACION DE SENSORES INTELIGENTES UTILIZANDO REDES
NEURONALES APLICADOS EN PROCESOS DE REFINACION DEL PETROLEO**

**HARRY JOSE PABA ARGOTE
EUDILSON NÚÑEZ COSSIO**

**Proyecto de grado para optar al título de
Maestría en Ciencias Computacionales**

**Director
FERNANDO OTERO
PhD Química**

**MAESTRIA EN CIENCIAS COMPUTACIONALES
UNIVERSIDAD AUTONOMA DE BUCARAMANGA
INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY (MÉXICO)
BUCARAMANGA, 2000**

Nota de aceptación

Presidente del jurado

Jurado

Jurado

Bucaramanga, 15 de Diciembre del 2000

Con todo mi amor:
A Dios,
A mi esposa Aleida,
Mi hijo Diego,
Mis padres: Saúl y Carmen
Y amigos que colaboraron
En este proyecto.
Gracias.

Harry

A Dios,
A mi esposa Azucena,
A mis padres Luisa y Manuel

Por estar junto a mi y
ayudarme con su amor a
alcanzar nuevas metas

Eudilson

TABLA DE CONTENIDO

INTRODUCCIÓN	xx
1 MARCO TEÓRICO	1
1.1 EL PROCESO ELEGIDO	1
1.1.1 Planteamiento del Problema.....	1
1.1.2 El Proceso de Extracción de Aceites Lubricantes con Fenol.	2
1.1.3 Variables Principales del Proceso	6
1.1.4 Variable de Salida	10
1.1.5 Análisis en Línea (El Refractómetro)	15
1.1.6 Análisis en Laboratorio	19
1.1.7 Diseño Funcional del Sistema de Control.....	24
1.2 ALTERNATIVAS PARA PREDECIR EL IV – IR	27
1.2.1 Sensores inteligentes	28
1.2.2 Redes neuronales con alimentación hacia adelante.....	29
1.2.3 La Red Neuronal Con Activación De Pesos Aleatorios (Rawn)	32
1.3 FUNDAMENTOS MATEMATICOS.....	51
1.3.1 Distribución.....	53
1.3.2 Regresión Múltiple.....	53
1.3.3 Funciones Autocorrelacionales.....	61
1.3.4 Dispersión	65
1.3.5 Medición de la predicción del error.....	65
1.4 CRITERIOS DE SELECCION DE VARIABLES DE ENTRADA	69
1.4.1 Por Regresión Múltiple	69
1.4.2 Por Correlación y Dispersión.....	72
1.4.3 Por el Experto.....	74

1.5	PREPARACION DE LOS DATOS DE ENTRADA	76
1.5.1	Colección de Datos	77
1.5.2	Ruido y calidad de los datos	80
1.5.3	Datos de entrenamiento y Datos de Prueba	81
1.5.4	Procesamiento de la señal (filtraje).....	82
1.5.5	Datos Falsos	84
1.5.6	Escalización	86
2	DISEÑO DE UN SENSOR INTELIGENTE PARA INFERIR EL IV EN LA PLANTA DE EXTRACCIÓN DE ACEITES LUBRICANTES CON FENOL.....	88
2.1	HERRAMIENTA DE PROGRAMACIÓN (MATLAB)	88
2.2	DIAGRAMA DE BLOQUES FUNCIONAL DEL SENSOR INTELIGENTE.....	90
2.2.1	Funciones de Importancia para el Sensor en Matlab	90
2.2.2	Código fuente en Matlab de las principales funciones	91
3	PRUEBAS.....	94
3.1	DESCRIPCION DE LAS PRUEBAS EFECTUADAS	94
3.2	RESULTADOS POR MODELOS PROPUESTOS	96
3.2.1	Modelo por Regresión	96
3.2.2	Modelo por Correlación y Dispersión.....	97
3.2.3	Modelo por el Experto	99
3.3	OPTIMIZACIÓN FINAL DE PARÁMETROS DE LA RED.	100
3.4	ANÁLISIS DE RESULTADOS	102
4	CONCLUSIONES Y RECOMENDACIONES	104
4.1	CONCLUSIONES	104
4.2	RECOMENDACIONES	107
5	BIBLIOGRAFIA.....	108
5.1	LIBROS Y ARTÍCULOS	108
5.2	DIRECCIONES DE INTERNET.....	115
5.2.1	Matemáticas.....	115
5.2.2	Sensórica	115
5.2.3	Redes Neuronales.....	116

LISTA DE TABLAS

Tabla 1.1 Conversión de IR a IV.....	14
Tabla 1.2 Valores del L y H para Viscosidad Cinemática en Sistemas 40-100 °C.....	20
Tabla 1.3 Las 10 Variables de Mayor Influencia por Regresión	72
Tabla 1.4 Las 10 Variables de Mayor Influencia por Correlación	74
Tabla 1.5 Variables Seleccionadas por el Experto.....	75
Tabla 2.1 Funciones en Matlab del Sensor Virtual.....	91
Tabla 3.1 Resultados de los Experimentos con el Modelo por Regresión.....	96
Tabla 3.2 Resultados de los Experimentos con el Modelo por Correlación y Dispersión.....	97
Tabla 3.3 Resultados de los Experimentos con el Modelo por el Experto	99
Tabla 3.4 Comparación de los Diferentes Modelos.....	101
Tabla 3.5 RMSE y MAXE para entrenamiento en la zona que cubre las muestras de la 4000 a la 5000.....	101
Tabla 3.6 Bias para las Neuronas de la Capa Oculta (B1) y la Neurona de Salida (B2).....	101
Tabla 3.7 Pesos de Entrada (*1E-04)	101
Tabla 3.8 Pesos de Salida.....	102

LISTA DE FIGURAS

Figura 1.1 Esquema simplificado de las Plantas Petroquímicas	2
Figura 1.2 Ubicación de la Planta de Extracción de Aceites Lubricantes con Fenol (Extracción con Fenol)	3
Figura 1.3 Relación de Cargas y Productos en la Planta de Fenol	3
Figura 1.4 Unidad de Extracción de Aceites Lubricantes con Fenol.....	4
Figura 1.5 Diagrama de flujo general unidad de extracción con fenol.....	6
Figura 1.6 Curva IR Vs. VI.....	15
Figura 1.7 Imágenes ópticas.....	16
Figura 1.8 Principio de Refracción.....	16
Figura 1.9 Sensor Real.....	17
Figura 1.10 Esquema Sensor Real.....	17
Figura 1.11 Tren de Pulsos.....	17
Figura 1.12 Indicador del Trasmisor (Monitor)	17
Figura 1.13 Diagrama BRIX e Imagen del algoritmo de Análisis.....	18
Figura 1.14 Imagen Óptica del Sensor Real	18
Figura 1.15 Instalación del Sensor.....	18
Figura 1.16 Sensor Real (Interno).....	18
Figura 1.17 Viscosímetro de Laboratorio (<100 C)	20
Figura 1.18 Viscosímetro de Laboratorio (>100 C)	22
Figura 1.19 Resumen de las estrategias de control avanzado propuestas	26
Figura 1.20 Ejemplo de una red neuronal alimentada hacia adelante. Los detalles muestran la también llamada "neurona de McCulloch-Pitts"	30
Figura 1.21 Resumen del algoritmo RAWN para entrenar una red neuronal alimentada hacia adelante con una capa oculta.....	38
Figura 1.22 Funciones de Activación Continua.....	40

Figura 1.23 Principio de Clustering	44
Figura 1.24 Línea de regresión de población hipotética con términos de error distribuidos normalmente.....	55
Figura 1.25 Resultado ordenado de las importancia de las variables de entrada para predecir la salida por regresión, eliminando las de menor importancia (una corrida de ejemplo).....	70
Figura 1.26 Resultado ordenado de las importancia de las variables de entrada para predecir la salida por correlación	73
Figura 1.27 Grafica de Dispersión de la variable 57. Una alta correlación con alta dispersión no debe ser tomada en cuenta para la predicción.....	73
Figura 1.28 Factory Link (Monitoreo y Control de la Planta)	77
Figura 1.29 Razón para evitar el rechazo automático de datos desviados.....	86
Figura 2.1Ejemplo 1 Gráficas Matlab.....	89
Figura 2.2 Ejemplo 2 Gráficas Matlab.....	89
Figura 2.3 Diagrama de Bloque y Principales Funciones del Sensor Virtual.....	90
Figura 3.1 RMSE por Número de Neuronas con Regresión.....	96
Figura 3.2 MAXE por Número de Neuronas con Regresión.....	97
Figura 3.3 RMSE por Número de Neuronas con Correlación y Dispersión.	98
Figura 3.4 MAXE por Número de Neuronas con Correlación y Dispersión.	98
Figura 3.5 RMSE por Número de Neuronas por el Modelo del Experto.	99
Figura 3.6 MAXE por Número de Neuronas por el modelo del Experto.	100
Figura 3.7 Comportamiento del Sensor Virtual Obtenido (Zona de Entrenamiento con 1000 muestras y Validación de las restantes).....	103

LISTA DE ANEXOS

ANEXO A. MANUAL DEL USUARIO

ANEXO B. GUIA DE INSTALACIÓN

ANEXO C. RELACION DE INFORMACIÓN ENTREGADA EN DISCO COMPACTO

ANEXO D. HOJA DE ESPECIFICACIONES DEL SENSOR REAL

ANEXO E. HOJA DE ESPECIFICACIONES DEL SENSOR VIRTUAL

GLOSARIO

ANÁLISIS DE COMPONENTES PRINCIPALES (PCA): Una técnica estadística que proyecta un conjunto de datos multi-dimensional hacia un nuevo sistema coordinado en el cual el eje principal cuenta para la mayoría de las variaciones en los datos. Posteriormente, los ejes ortogonales cuentan para pequeñas cantidades incrementales de variación en los datos originales. Permite la menor dimensión descriptiva a descartarse con un mínimo de pérdida en la información.

APRENDIZAJE: Aprender, en sentido estadístico, se refiere a cualquier proceso que requiere el uso de datos para sintonizar un conjunto de parámetros que describan un modelo estadístico para esos datos. No implica ninguna cualidad humana como son entendimiento, conciencia o inteligencia asociada con nuestras habilidades de aprendizaje.

ARQUITECTURA: Se refiere al diseño de una red neuronal: el número de unidades y su patrón de conexión. El término es usado a veces para referirse a los tipos de redes neuronales tales como Multi-layer Perceptróns (MLP's) o redes Kohonen, pero en este libro su significado está restringido al diseño específico de un MLP.

BIAS: Valor de Ajuste

CONEXIONES: Ver Pesos.

CONJUNTO DE ENTRENAMIENTO: Un conjunto de datos que no es utilizado durante el entrenamiento de tal forma que las habilidades de generalización de la red puedan ser probadas.

CONJUNTO DE VALIDACIÓN: Un conjunto de datos utilizado durante el entrenamiento, no para alterar pesos de la red, sino para revisar el progreso de las habilidades de generalización de la red. Utilizado para evitar el sobreajustamiento. Puede ser lo mismo que el conjunto de entrenamiento.

CONVOLUCIÓN: Aplicación de una función a una señal con el fin de suavizarla tomando un promedio de movimiento de pesos. Es equivalente a la multiplicación de la transformada de Fourier de la señal con aquella de la función de Convolución.

ENTRENAMIENTO: El proceso de permitir que una red neuronal aprenda. Ver aprendizaje.

ENTRENAMIENTO EN LÍNEA: Lo contrario de entrenamiento por bloques. Los pesos de las redes son actualizados luego de la presentación de cada patrón de entrenamiento en turno. Esto requiere que el orden de las presentaciones de entrenamiento sea mezclado pero es menos probable de que quede atrapado en un mínimo local.

EPOCAS: Una presentación del conjunto de entrenamiento completo a una red neuronal.

ETA (η): Un parámetro que toma valores entre cero y uno y que define la proporción de cada error que se utiliza para actualizar los pesos durante el aprendizaje de la propagación hacia atrás. En efecto balancea la velocidad de aprendizaje respecto a la estabilidad.

FUNCIÓN DE ACTIVACIÓN: Una función matemática que toma los valores de activación pesados que llegan a una unidad, los suma, y traslada el resultado a una posición a lo largo de una escala dada. La función de activación con frecuencia comprime el valor sumado dentro de un rango dado y en consecuencia es conocido también como función squashing (compresión) y función de compresión sigmoideal.

FUNCIÓN DE ACTIVACIÓN SIGMOIDAL: Una función de activación que comprime su entrada a un rango entre cero y uno. A medida que la entrada a la función crece en magnitud, es comprimida dentro de una parte exponencialmente más pequeña de los extremos del rango de salida.

FUNCIÓN DE ERROR: La función que calcula cuán cerca de la salida deseada se encuentra en efecto cualquier salida de la red actual. Puede ser la diferencia entre el destino deseado y los valores actuales, o puede estar basado en otros criterios.

FUNCIÓN DE TRANSFERENCIA: Ver Función de activación

MÍNIMO LOCAL: Una combinación de valores de peso que no conducen al error más bajo posible para un conjunto de datos, pero que, debido a la forma local del error, parece ser óptimo y en consecuencia que un algoritmo de gradiente descendiente proceda posteriormente.

MOMENTUM: Parámetro del Backpropagation para mejorar la velocidad de aprendizaje. Afecta los pesos por los pesos previos evitando el mínimo local.

MULTI LAYER PERCEPTRÓN (MPL): Perceptrón Multicapa. Tipo de red neuronal con capas de entrada, ocultas y de salida distintas.

OFF-LINE: Fuera de Línea.

OLS (Ordinary Least Squares): Mínimos cuadrados ordinarios.

ON-LINE: En línea.

PESOS: Conexiones de fuerzas variables que llevan la información de activación entre unidades de la red.

POST-PROCESAMIENTO: Cualquier procesamiento de datos que sea llevado a cabo sobre los valores de salida de una red neuronal. Un ejemplo típico es el de la conversión de etiquetas de categoría numérica en nombres de categoría.

PRE-PROCESAMIENTO: El proceso de preparar datos crudos para su presentación a una red neuronal.

PROPAGACIÓN HACIA ATRÁS DEL ERROR: El acto de calcular el error en una unidad oculta de red, basado en el error derivado de las unidades superiores y las fuerzas de los pesos conectados a ellas.

RAWN: (Random Activation Weight Network) Técnica de entrenamiento para redes neuronales consistente en la asignación de pesos aleatorios a las entradas y posterior cálculo de los pesos de salida por OLS.

REGULARIZACIÓN: Cualquier dispositivo que limite la complejidad, y en consecuencia el ajuste de un modelo estadístico tal como una red neuronal. Con frecuencia utilizado para limitar el crecimiento de pesos.

RMS(Root Mean Square): Valor eficaz

RMSE: (Root Mean Square Error) Error de Raíz Cuadrada Media o valor eficaz del error

SQUASHING: Tipo de función de compresión.

TAG: Nombre con que se identifica una variable de proceso

TAZA DE APRENDIZAJE: Ver Eta

TOPOLOGÍA: Ver Arquitectura

TRANSFORMADA RÁPIDA DE FOURIER (FFT): Una aproximación rápida a la transformada de Fourier que calcula las frecuencias constituyentes en una señal compleja. Requiere una señal de tamaño 2^n .

UNIDADES OCULTAS / CAPA OCULTA: Las unidades de una red que no reciben entradas ni generan salidas. Las unidades ocultas combinan y extraen características provenientes de los datos de entrada.

VALIDACIÓN DE DATOS: Aseguramiento de que los datos de entrenamiento contienen la información requerida para ejecutar cierta tarea.

VALOR DE ACTIVACIÓN: El valor producido por una función de activación de una unidad, también conocida como la salida de la unidad.

WINDOWING (VENTANEANDO): Un proceso para construir vectores de entrada estáticos a partir de una señal variable en el tiempo concatenando los contenidos de una ventana deslizante temporal.

RESUMEN

TITULO: IMPLEMENTACIÓN DE SENSORES INTELIGENTES UTILIZANDO REDES NEURONALES APLICADOS EN PROCESOS DE REFINACIÓN DEL PETRÓLEO.

AUTORES: HARRY JOSE PABA ARGOTE Y EUDILSON NÚÑEZ COSSIO

En este trabajo se desarrolló un sensor virtual para predecir el índice de viscosidad de un proceso de la industria petroquímica denominado Extracción de Aceites Lubricantes con Fenol. Para ello se utiliza una red neuronal entrenada por el método de Activación de Pesos Aleatorios, técnica que a diferencia de otras no es iterativa y por ello resulta ser mucho más rápida que otros métodos tradicionales, alcanzando objetivos de error suficientemente bajos para reemplazar el sensor real.

Como resultado, la herramienta de software elaborada para tal fin, puede ser utilizada en el entrenamiento de cualquier sistema (entradas-salida) que se pretenda resolver aplicando redes neuronales de este tipo. Cuenta con todos los pasos intermedios requeridos como la asistencia en la selección de variables por métodos estadísticos que utilizan la matemática requerida para el tratamiento de esta clase de procesos estocásticos, pruebas de posible linealidad, tratamiento de señal para filtrar ruido y eliminar datos falsos, entrenamiento-validación y pruebas por simulación fuera de línea y en línea con el proceso real. Se utiliza la medición del Error RMS y el Máximo Error encontrado, principalmente en la fase de validación para ser usado como el parámetro de comparación que permita evaluar el desempeño del modelo obtenido.

La técnica de entrenamiento utilizada es suficientemente rápida para implementar funciones de re-entrenamiento en línea en caso de ser requerido por el sistema lo cual se registra gráficamente para atender este requerimiento.

INTRODUCCION

En la actualidad existe una tendencia en las diferentes áreas de la ingeniería en la aplicación de nuevas metodologías de computación, que concentren y mejoren los métodos de resolución de problemas, en los cuales la computación tradicional no ha proporcionado los mejores resultados.

Esta nueva forma de computación se basa en la emulación inteligente del comportamiento biológico humano y es capaz de manejar las imprecisiones e incertidumbres que aparecen cuando se trata de darle solución a problemas relacionados con el mundo real (predicción de variables, control de procesos, inferencia de modelos, etc.), ofreciendo soluciones robustas de fácil implantación.

El objetivo de éste trabajo es desarrollar un sensor inteligente funcionando en línea con un proceso industrial, para demostrar la exitosa aplicabilidad de ésta rama de la computación, utilizando algoritmos optimizados de aprendizaje que en un tiempo mínimo aseguren la precisión requerida en el proceso.

El alcance es elaborar una herramienta que permita diseñar sensores virtuales con facilidades gráficas para el usuario, permitiéndole seleccionar y adecuar las variables de mayor incidencia sobre la medida a predecir, entrenar el modelo de red neuronal escogido para obtener los parámetros óptimos de respuesta, simular fuera de línea el comportamiento de la variable a predecir con datos históricos de las variables de entrada del proceso y entregar en línea la medida de la variable seleccionada para la predicción.

La importancia de contar con un sensor virtual en un proceso industrial es la de permitir una alternativa de respaldo a la medición tradicional, validando los resultados de un

instrumento real instalado en la línea de producción o fuera de ella (si se hace en laboratorio), lográndose inclusive reemplazarlos cuando ya se han entrenado suficientemente para responder a las condiciones extremas del proceso. Cubre un amplio rango de aplicaciones, que van desde los diferentes procesos petroquímicos (destilación, ruptura catalítica, viscorreducción, etc.) hasta el control ambiental, en donde para todos ellos se utilizan analizadores costosos que requieren frecuentes calibraciones y ajustes, con patrones de alta precisión y muchas veces de alto riesgo para su manejo. La utilización de este tipo instrumentos virtuales, elimina el problema común de los analizadores tradicionales como lo es el alto mantenimiento requerido, debido a que son instalados en medios altamente corrosivos, de altas vibraciones, temperaturas y presiones que los deterioran constantemente.

Los resultados obtenidos sobre la planta de Fenol demuestran que el instrumento real puede ser reemplazado por el virtual (ver detalles en el capítulo 3), posibilitando la utilización de éste nuevo analizador en un lazo de control cerrado (ver sección 1.1.7 sobre el Control del Proceso).

Para un proceso con tiempo muerto considerable como el tratado aquí, se constituye en otra ventaja la utilización de estos analizadores virtuales, ya que es la única forma de anticiparse a los resultados de calidad, tomando las condiciones actuales del proceso y predecir lo que pasará con él al cabo de un determinado tiempo. Para ello es necesario contar con una buena cantidad de datos históricos que permitan alimentar el modelo a entrenar con datos desfasados entre las entradas y la salida, según sea el retardo del proceso.

En el contenido de este documento se encuentran los detalles para la elaboración de un instrumento virtual aplicable a cualquier proceso de la Industria del Petróleo, para efectos ilustrativos se seleccionó la aplicación de un Sensor Virtual que reemplaza el instrumento de medición de Índice de Refracción de una planta de Aceites Lubricantes, utilizado para controlar la calidad en el Índice de Viscosidad del refinato producido en la unidad. Para ello se establece un modelo de red neuronal que permite inferir valores de variables de proceso de difícil medición en función de otras variables de medición fácil y continua, utilizando instrumentos sencillos y robustos tales como medidores de temperatura, presión, flujo y nivel.

Inicialmente se recopila en el Marco Teórico la información básica sobre el proceso elegido (ver sección 1.1) y las variables que allí se manejan, tanto las de entrada como la de salida. Ilustra los diferentes métodos de medición de la variable objetivo en línea con el proceso, como también aquellos que se utilizan para obtener los resultados de calidad en el laboratorio.

La sección 1.2 presenta las alternativas para la predicción de la variable de salida, describe qué son los sensores inteligentes, las redes neuronales y el método seleccionado para el entrenamiento de la red neuronal (RAWN).

Los Fundamentos Matemáticos necesarios (ver sección 1.3), dan un repaso concreto sobre las técnicas estadísticas utilizadas, teniendo en cuenta el tipo de procesos estocásticos tratados.

Una vez alcanzado todo este conocimiento se procede a seleccionar las variables de entrada que mejor puedan inferir la salida (ver sección 1.4). Para ello se utilizan métodos estadísticos tales como la regresión múltiple, correlación y dispersión, conjugados con la experiencia del operador y la información de diseño de la planta.

La captura y preparación de los datos se documenta en la sección 1.5, en donde se resuelven los principales problemas tales como ruido, datos falsos y escalización.

En el capítulo 2 se muestra cómo se construyó el sensor inteligente, utilizando una herramienta de programación de alto nivel (Matlab 5.2) que ofrece gran flexibilidad en cuanto a plataformas de ejecución se refiere (Unix y Windows), facilidades para elaborar una interfase gráfica de usuario de alta calidad, el manejo de matrices y toda la matemática requerida, ejecutadas con comandos y funciones sencillas.

Una vez construida la aplicación y habiendo colectado datos de las mejores variables que puedan inferir la salida, se procede a realizar pruebas (ver capítulo 3) para determinar las características óptimas de una red neuronal: cantidad de muestras, número de neuronas, pesos y ajustes (bias) obtenidos del entrenamiento y así implementarlos en cualquier sistema que ofrezca soluciones con redes neuronales, ya sea en hardware o en software, como lo ofrece la aplicación desarrollada.

1 MARCO TEÓRICO

1.1 EL PROCESO ELEGIDO

1.1.1 Planteamiento del Problema

Para el control adecuado de los procesos de la Industria del Petróleo, se requiere medir muchas variables para la toma de decisiones adecuadas, en cuanto a la producción en cantidad y calidad. Estas variables van desde simples mediciones de flujos, presiones, niveles y temperaturas, tomados con instrumentos estándar que funcionan por la relación de mediciones físicas en forma indirecta, hasta otras mucho más complejas que en ocasiones requieren análisis fuera de línea (en el laboratorio), pero que ocasionan retardos y muchas veces no son oportunos para tomar decisiones.

También existen analizadores instalados en línea con el proceso, pero presentan características que los hace costosos, tales como su instalación, mantenimiento, confiabilidad y disponibilidad. Por tal razón surge la necesidad de crear alternativas que entreguen esta información a tiempo, de manera confiable y continua para alcanzar los objetivos que se quieren en la planta. Cualquiera que sea el proceso seleccionado se encuentra en él todas las condiciones anteriormente mencionadas, así como una característica común a los procesos de la Industria del Petróleo, como lo es la no-linealidad que relaciona el comportamiento de las variables de entrada con la de una salida a inferir, lo cual los hace indicados para atacarlos con soluciones que involucren redes neuronales.

El proceso elegido para esta implementación es la Planta de Extracción de Aceites Lubricantes con Fenol, la cual reúne todas las características para ejemplificar la

aplicación del sensor virtual. Cuenta con un analizador en línea y toda la instrumentación asociada al control y monitoreo del proceso, centralizado en un equipo que sirve como Interfase de Operación.

El objetivo final es reemplazar el analizador real por el virtual (de software) dadas las fallas frecuentes que viene presentando el primero y así tener un control instantáneo en la calidad del producto que se está entregando. Trae además ganancias económicas para la planta, debido a que disminuye los costos de las materias primas y los servicios utilizados no requeridos para obtener una calidad de refinato por encima de lo necesario. Cambiando la situación actual, en la cual se opera para producir calidad en exceso y así evitar reclamaciones de los clientes.

1.1.2 El Proceso de Extracción de Aceites Lubricantes con Fenol.

En el Complejo Industrial de Barrancabermeja, la Unidad de Extracción de Aceites Lubricantes con Fenol hace parte de la Planta de Parafinas que a su vez pertenece al grupo de Plantas Petroquímicas, cuya actividad es producir compuestos químicos de alta pureza.



Figura 1.1 Esquema simplificado de las Plantas Petroquímicas

La unidad de extracción con fenol busca mejorar el índice de viscosidad, la resistencia a la oxidación y disminuir la tendencia de los aceites lubricantes a formar depósitos carbonosos. Esta unidad trabaja con tres tipos de carga alternadamente: destilado medio, destilado liviano y DAO.

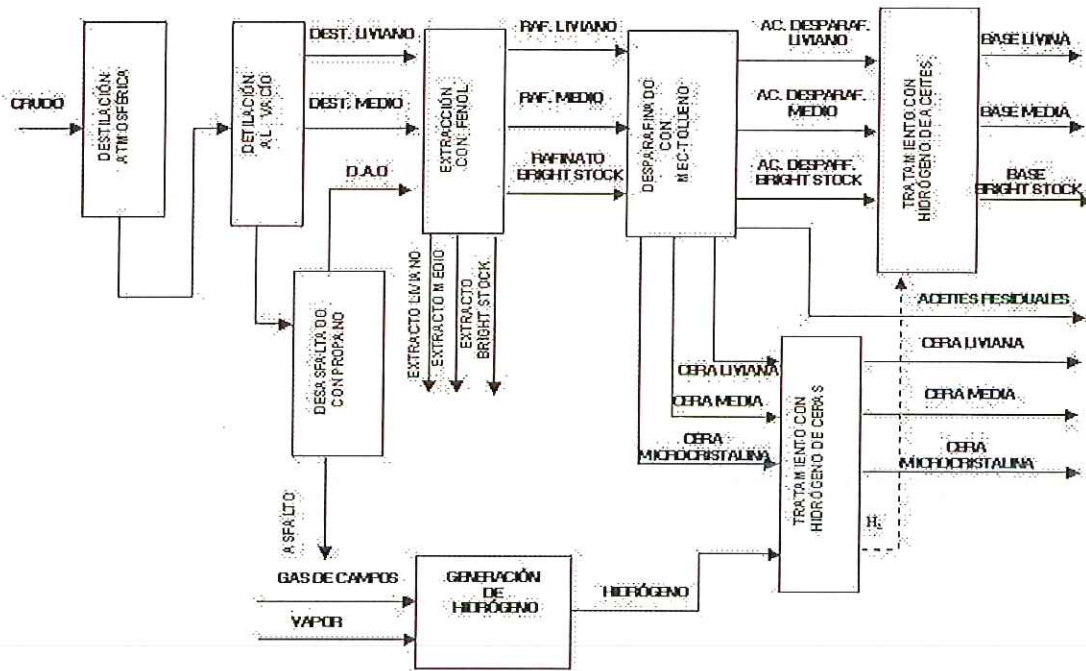


Figura 1.2 Ubicación de la Planta de Extracción de Aceites Lubricantes con Fenol (Extracción con Fenol)

La operación principal de esta unidad se basa en una extracción líquido-líquido, donde la utilización de un solvente polar de baja temperatura de ebullición como lo es el fenol, extrae el material aromático y nafténico del aceite de carga, de modo que en el refinado se retire un aceite tipo parafínico con el índice de viscosidad deseado. El fenol es el encargado de solubilizar la parte nafténica que sale en el extracto y las operaciones siguientes están encargadas de retirarle el solvente a cada una de las dos corrientes generadas en la etapa anterior, con el fin de recircularla y utilizarla nuevamente.

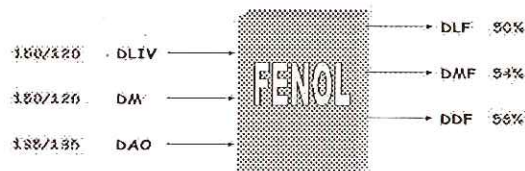


Figura 1.3 Relación de Cargas y Productos en la Planta de Fenol

La descripción del flujo para la unidad de extracción con fenol puede ser analizada separando el proceso en 3 secciones:

Sección de absorción
Tratamiento con fenol
Recuperación de fenol

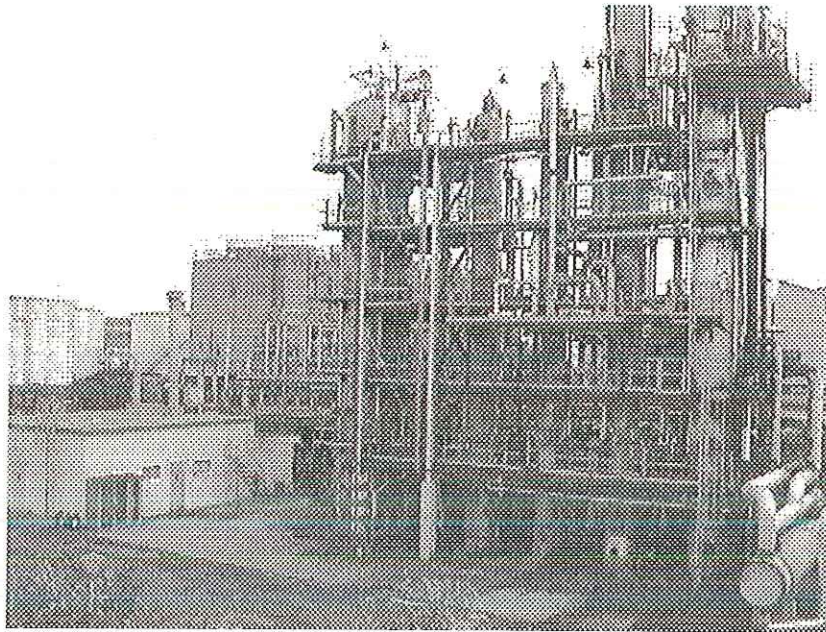


Figura 1.4 Unidad de Extracción de Aceites Lubricantes con Fenol

Sección de absorción: Ésta sección busca recuperar la mayor cantidad posible de fenol de la corriente de vapor fenólico que se inyecta a la torre, donde la corriente de carga pasa en contracorriente, permitiendo un contacto íntimo entre sus moléculas y haciendo posible la transferencia de masa.

Sección de tratamiento con fenol: El tratamiento hace que los compuestos aromáticos y nafténicos sean extraídos del aceite de carga debido a diferencia de solubilidades, en la torre de extracción el aceite fenolizado es cargado por el fondo, mientras que el solvente se carga por la cima y desciende por gravedad.

En la cima se mantiene un nivel de interfase entre fenol – aceite. El fenol y el agua fenólica se inyectan a la torre sobre la bandeja de cima pero debajo del nivel de la interfase, manteniendo la altura deseada por medio de la descarga de extracto.

Sección de recuperación de fenol: La sección de recuperación es la encargada de separar el solvente del extracto y el refinato. Para cada corriente se utiliza una torre de vaporización flash y una torre despojadora, en el despojo de fenol del extracto se utiliza una torre adicional (torre secadora).

Recuperación del fenol en el extracto: El flujo de extracto que sale por el fondo de la torre de extracción entra a una torre secadora, donde se retira el agua con fenol de la mezcla fenol-agua-aceite, el medio de calentamiento de la torre se suministra por intercambio de calor entre la corriente de cima de la torre de vaporización instantánea.

El flujo de vapor fenólico después de salir por la cima de la secadora, se divide en dos corrientes, la mayor parte se condensa y va al tambor de agua fenólica y la menor porción a la torre absorbadora ya mencionada. La mezcla de fenol y extracto secos, desciende a través de la sección secadora en contracorriente con los vapores de fenol que despojan el agua remanente de la mezcla y pasa al fondo de la torre a través del rehervidor. Una porción de fenol en el líquido que desciende se vaporiza al pasar por el rehervidor, mientras que el resto de la solución se acumula en el fondo de la torre, parte del vapor de fenol se condensa por la carga y se une con el chorro de fenol que viene de la sección instantánea.

La solución de extracto seco acumulado en el fondo de la torre, se envía a la cima de la torre instantánea de extracto, pasando en contracorriente con los vapores de fenol que suben y se acumula en una batea. El medio de calentamiento a esta sección es suministrado por el horno que toma líquido de la batea, el vapor de fenol que sale por la cima de la torre es el encargado de intercambiar calor en el rehervidor que proporciona calor al fondo de la torre secadora. El fenol se condensa y se lleva al tanque acumulador de fenol puro.

Recuperación del fenol en el refinato: El refinato solución (cima de la torre de tratamiento) pasa por un horno antes de ir a la torre de vaporización instantánea, de ésta torre se retira por la cima vapor fenólico, que va al tanque de almacenamiento de fenol puro y por el fondo, el aceite tratado pasa a la torre despojadora.

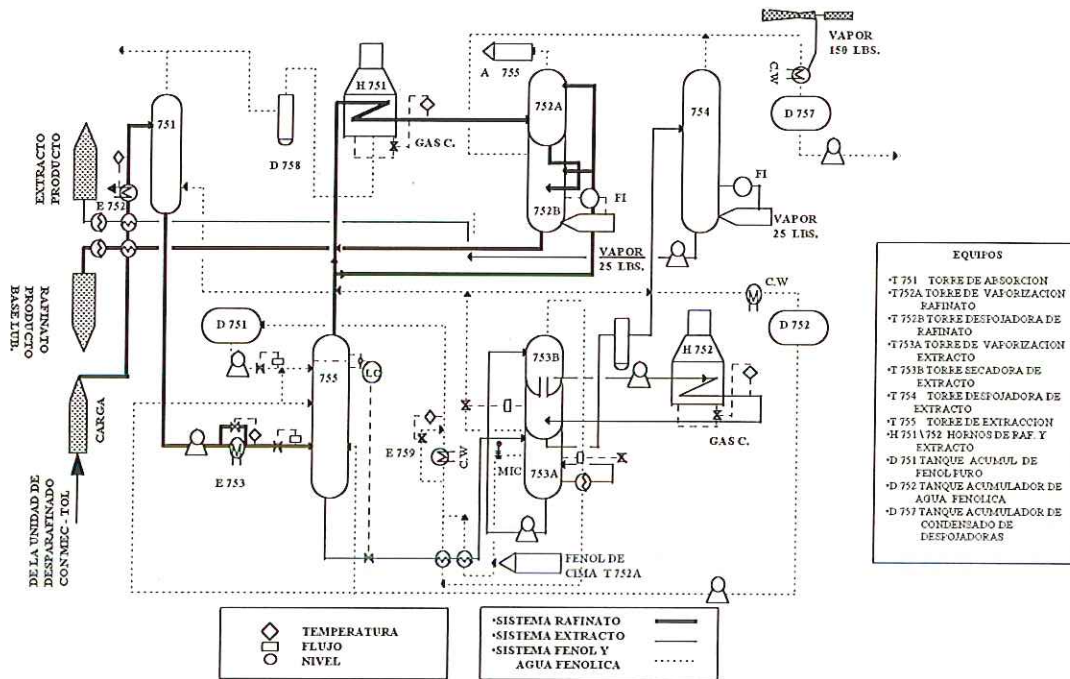


Figura 1.5 Diagrama de flujo general unidad de extracción con fenol

1.1.3 Variables Principales del Proceso

A continuación se describen las principales variables del proceso y la forma como deben controlarse para una buena operación de la planta (el Anexo A muestra los valores típicos para la operación).

Las variables controladas en el tratamiento con fenol son las siguientes:

- Relación Fenol-Aceite: Es la relación entre la rata de inyección de fenol y la carga de aceite multiplicada por 100, medidos ambos flujos a la entrada de la torre. Para una cantidad determinada de aceite de carga, un aumento en la inyección de fenol mejora la calidad del rafinado pero disminuye su rendimiento. Entre más pequeña sea la relación fenol-aceite, mayor es el efecto de las causas que tienden a trastornar la operación en la torre. La óptima relación para un aceite dado se determina empíricamente.
- Calidad de la carga: Cuanto más viscoso sea el aceite, menor es la capacidad de carga. En otras palabras, entre más viscosa sea la carga, mayor será la relación

fenol-aceite requerida. Es muy importante puesto que la calidad del producto depende de la calidad de la carga.

- (c) Temperatura de tratamiento: Cuanto más baja sea la temperatura de tratamiento, mejor es la selectividad del fenol, lo cual se traduce en un ligero aumento en el rendimiento de refinado del mismo índice de viscosidad. Pero un aumento en la selectividad del fenol está acompañado por una disminución en su poder solvente. Como resultado de esto, la relación de fenol - aceite debe aumentarse para obtener el mismo índice de viscosidad. También, cuanto más baja sea la temperatura de tratamiento, más viscosos serán el aceite y el solvente y menor la capacidad de tratamiento. Este es un factor primordial cuando se cargan aceites de alta viscosidad.

Los diferentes aceites de carga se mezclan en todas las proporciones con el fenol a temperaturas que por lo regular son cercanas a las especificaciones de tratamiento. Generalmente, para obtener un tratamiento eficiente, es ventajoso operar a temperaturas lo más cercanas posible de las de mezcla. La temperatura de mezcla cambia con las diferentes cargas y con el contenido de agua en la carga.

Cuanto más cerca esté la temperatura de tratamiento de la temperatura de mezcla, más fácilmente se trastornará la operación en la torre. La inundación o la falta de separación del refinado empieza en o debajo de la zona de carga del aceite. Una señal de esta condición es la acumulación del fenol en la torre de tratamiento, seguido por un aumento de la cantidad de fenol en la solución de refinado. También puede resultar, debido a esta condición, arrastre de aceite en la solución de extracto por el fondo de la torre.

- (d) Inyección de Agua: El agua y el fenol se mezclan completamente en todas las proporciones a temperaturas mayores de 155 °F. Por debajo de esta temperatura (155 °F), dependiendo de la concentración de fenol, una mezcla de fenol y agua puede separarse en dos o más capas. El fenol se concentra en la fase o capa inferior y el agua en la superior.

- (e) Añadiendo agua con fenol, se aumenta la selectividad del solvente. Un aumento en la selectividad del fenol está acompañada por una reducción en la solubilidad de aceite. Como resultado, la relación fenol-aceite debe ser aumentada para obtener la misma calidad en el refinado.

El agua se inyecta en la torre con la carga de fenol, para disminuir la solubilidad del aceite en la solución de extracto, reduciendo así el aceite de alta calidad disuelto en el extracto. El aceite fluye hacia la solución de refinado y sirve como reflujo. Como resultado, el aceite nafténico, el cual se disuelve más fácilmente en el fenol, sale como extracto. El uso de agua, en esta manera, aumenta el rendimiento de refinado, pero disminuye su calidad.

Los resultados preliminares de los cambios de condiciones en el tratamiento, pueden apreciarse en corto tiempo en la torre de tratamiento, pero se requieren varias horas para que la torre alcance el equilibrio y así apreciar los resultados de los cambios en el producto.

Se debe evitar por lo tanto, el hacer cambios frecuentes en las condiciones de tratamiento, sin esperar a evidenciar los resultados.

Se tienen instalados controles para ayudar al operador a establecer y mantener una buena operación, con los diferentes cargas a la torre secadora. Las funciones de esos controles, se describen a continuación:

FR-7009: registrador del reflujo a la cima de la torre secadora. Este reflujo debe ser aproximadamente la décima parte de la cantidad de vapor que sale por la cima de la torre.

TRC-7004: Controlador registrador de la temperatura de la carga a la torre secadora. Controla la temperatura mediante un pase en el lado fenol(casco) del precalentador de la carga, intercambiador E-755 A/B. Debe ajustarse para obtener la temperatura controlable más alta, sin exceder 330°F (o su equivalente, sí la presión de la torre no es 10 psig).

TR-7007: Controlador registrador de la temperatura de los vapores de cima de la torre secadora. Controla dicha temperatura ajustando el reflujo de agua fenólica a la cima de la torre.

PRC-7002: Controlador registrador de la presión de la torre secadora. Trabaja variando la apertura de la válvula de control que regula la salida de los vapores de cima de la torre secadora.

LC-7006: Controlador de nivel del fondo de la torre secadora, opera variando la válvula de control en la descarga de la bomba de solución de extracto seco, P-753 A/B.

MIC-7003: Controlador del chorro lateral, registrado por el FR-7007. En algunos de los platos se han instalado indicadores de temperatura y termo pozos con el fin de que el operador se dé cuenta, con suficiente anticipación, del aumento en el contenido de agua en la bandeja número 14, y así pueda prever la necesidad de hacer algún cambio en las condiciones de operación; sin embargo, el examen del sistema mostrará que puede pasar un tiempo considerable entre el momento en que se efectuó el cambio y la aparición del efecto producido por dicho cambio. Por esta razón, se aconseja que los ajustes que se hagan en las condiciones de operación sean hechos en incrementos pequeños y que se deje pasar suficiente tiempo entre dichos cambios para permitir que el sistema se equilibre.

Como un ejemplo del efecto producido por el tiempo de retardo, se puede considerar la serie de efectos que entran en juego cuando se disminuye la extracción lateral. En primer lugar, el calor cedido a la carga de la secadora disminuye, por lo que debe ser compensado mediante el aumento del precalentamiento de la misma en el intercambiador E-755 A/B, lo cual ocurre automáticamente. En segundo lugar, la cantidad de fenol en el fondo de la secadora aumenta. En tercer lugar, hay un aumento de la carga de fenol en el horno de solución de extracto, lo cual se compensa con aumento en el gas a los quemadores, para mantener constante la temperatura de salida. En cuarto lugar, los vapores de fenol adicionales provenientes del horno, salen por la cima de la torre de vaporización instantánea y suministran un calor adicional al rehervidor de la secadora, produciendo aumento de temperatura en las bandejas de la torre, haciéndose necesario abrir el MIC-7003, aumentando así la extracción lateral. El tiempo requerido para producir

todos estos efectos, puede ser de 5 a 10 minutos, de acuerdo con la retención del líquido del sistema, particularmente la del fondo de la torre secadora y la de la batea de la torre de vaporización instantánea.

Aunque se espera que los instrumentos de control hagan automáticamente los ajustes requeridos por las variaciones normales de la carga, cambios mayores y los efectuados durante el arranque, necesitan ajustes adicionales, que son realizados por el operador.

Cuando la torre secadora esté recibiendo demasiado calor, puede suceder el suministrado a través de la carga o a través del rehervidor en las bandejas bajas. El exceso de calor será indicado por la demanda excesiva de reflujo para mantener la temperatura de la cima bajo control. Esta situación puede ser causada por el precalentamiento excesivo de la carga, en cuyo caso el TRC- 7004 debe ser ajustado lo máximo que permita la capacidad del enfriador E-759 para absorber el calor que queda en el chorro de fenol.

1.1.4 Variable de Salida

Existen varias propiedades del lubricante que pueden ser usadas para indicar la calidad del producto. Una de ellas es el Índice de Viscosidad (IV) al cual se le hace un muestreo dos veces al día para analizar en laboratorio y llevar un control adecuado. El IV del lubricante es afectado por las propiedades de la carga y controlado por condiciones físicas en la columna de extracción, sin embargo está influenciado por condiciones dentro del desparafinado; por lo tanto, el muestreo se hace sobre el aceite despojado en el desparafinado para el control de la calidad. El tiempo de residencia combinado de la unidad de extracción y la de desparafinado es considerablemente más corto que el período de muestreo del IV, volviéndose muy difícil el control de éste. Adicionalmente, el flujo de refinato producto de la unidad de extracción varía con el almacenamiento de carga. Así, el sistema puede estar sujeto a retardos grandes y variables de tiempo del proceso, ocasionando un problema en el control.

Las estrategias actuales de operación utilizan un esquema de control de cascada paralela de lazo abierto para regular el Índice de Viscosidad. Un lazo interno regula una variable secundaria, el Índice de Refracción del refinato (IR).

Con una de esta dos variables se puede inferir la otra debido a su relación lineal. De esta forma en nuestro proceso se mide la calidad en línea del refinato utilizando un analizador de IR que posteriormente es convertido a IV como más adelante se explica.

1.1.4.1 Índice de Viscosidad.

La viscosidad de una sustancia es la resistencia de ruptura de una capa de un líquido que separa dos placas horizontales, una de las cuales esta siendo movida a través de la otra. La viscosidad absoluta de una sustancia es la fuerza en dinas que puede, mover un centímetro cuadrado de una superficie plana a una velocidad de un centímetro por segundo.

Relativo a otra superficie plana paralela de la cual esta separada por una capa de una sustancia de un centímetro de espesor. Esto puede ser expresado por la siguiente formula:

$$\text{viscosidad} = (\text{fuerza} \times \text{grosor de la película}) / (\text{área del plano} \times \text{velocidad})$$

El tubo de Cannon-Fenske mide la viscosidad cinemática en unidades de centistokes y es igual al tiempo de flujo multiplicado por la constante del tubo. Más exactamente los valores para la constante de un tubo dado, pueden ser determinados usando un líquido de viscosidad conocida, midiendo el tiempo en segundos y calculando "c" en la formula:

$$\text{viscosidad cinemática (centistokes o cSt)} = c \times \text{tiempo (s)}$$

La viscosidad es sensible a la temperatura y es normalmente probada en baños de temperatura constante.

Unidades de Viscosidad

Poise: La unidad de poise, el valor para la viscosidad absoluta, es igual a una dina por segundo sobre centímetro cuadrado. Esto es, cuando dos placas tienen un área de

ruptura de un centímetro cuadrado y el grosor de la capa es de un centímetro, si una fuerza de una dina se requiere para mantener la velocidad en un centímetro por segundo, del fluido se dice que tiene una viscosidad de 1 poise. Un poise también es igual a un stoke multiplicado por su la densidad (a una temperatura específica).

Centipoise: Debido a que el poise es una unidad muy grande, frecuentemente es conveniente usar el centipoise (1/100 poise).

Stoke: La viscosidad cinemática de una sustancia es la relación de la viscosidad para la densidad de la sustancia a la temperatura de medición. La unidad de viscosidad cinemática es el stoke y es igual a la que posee un fluido que tiene una viscosidad de un poise y una densidad de un gramo por centímetro cúbico, por ejemplo para un fluido con viscosidad de 9 poise y una densidad de 0.9090 es igual a 8.18 stokes (818 centistokes). La selección del tubo apropiado será el 46460-400 para un rango de 240-1200 centistokes.

Centistoke: Nuevamente, debido a que el stoke es una unidad tan grande, generalmente es conveniente usar el centistoke (1/100 stoke).

El índice de viscosidad es un número arbitrario usado para caracterizar la variación de la viscosidad cinemática de un producto derivado del petróleo con la temperatura.

Uno de los mayores requerimientos de los aceites para motores y otros aceites minerales, es que tengan características satisfactorias de Índice de Viscosidad, es decir, la variación de la viscosidad con la temperatura. Es necesario en los aceites, para que éstos no pierdan su fluidez y muestren un comportamiento igual, a través de las distintas temperaturas a que se someten cuando son usados. Cuanto más bruscos sean los posibles cambios de temperatura, más pequeños deben ser los cambios de viscosidad. Esto asegura una distribución igual del aceite por toda la máquina y evita la posibilidad de desgaste de las piezas, el cual ocurre durante el tiempo que demora la máquina en calentarse a la temperatura normal de trabajo. La viscosidad alta, a baja temperatura restringe el flujo de aceite cuando la máquina está fría, mientras que con viscosidad baja a la temperatura normal de operación de la máquina, puede no formar una película de aceite de suficiente espesor que evite el contacto directo de las piezas metálicas en

movimiento. Debido principalmente a la conveniencia de usar un número sencillo que exprese las características de viscosidad con relación a la temperatura de un aceite, el Índice de Viscosidad (VI) ideado hace varios años, continua usándose hoy en la operación de plantas de aceites lubricantes, primordialmente como una indicación conveniente, de la tendencia o nivel de extracción de una planta para una carga dada.

El índice de viscosidad de un aceite se presenta en un gráfico o tabla, determinando sus viscosidades a 40 °C y 100 °C.

En la operación de una planta, el mejoramiento del IV por lo regular va asociado con la disminución del rendimiento. Por lo tanto, los intentos de producir Índices de Viscosidad (VI) más altos que los normales o los especificados en el diseño, no son económicamente deseables.

1.1.4.2 Índice de Refracción

Se define como la relación de la velocidad de la luz (longitud de onda específica) en el aire, con relación a su velocidad en la sustancia bajo examen. También se puede definir como el seno del ángulo de incidencia dividido por el seno del ángulo de refracción. A medida que la luz pasa del aire hacia la sustancia, produce una deflexión que marca una diferencia de ángulo cuya relación se llama índice; éste es el índice de refracción relativo. Si se desea el índice de refracción absoluto (referido al vacío), éste valor se debe multiplicar por el factor 1.00027 (índice de refracción absoluto del aire). El valor numérico del índice de refracción de los líquidos, varía en forma inversa con la longitud de onda y la temperatura.

Dispersión refractiva.

Es la diferencia entre los índices de refracción de una sustancia para la luz y dos longitudes de onda diferentes, medidos ambos índices a la misma temperatura. Por conveniencia en los cálculos, el valor de la diferencia obtenida es usualmente multiplicada por 10,000.

Cálculo del índice de refracción

El índice de refracción se mide por el método de ángulo crítico con un refractómetro de precisión Bausch & Lomb usando una luz monocromática. El instrumento se calibra previamente por medio de un estándar de referencia sólido y se corrigen los valores observados, cuando sea necesario, por la calibración obtenida con los estándares líquidos certificados.

Significado y Uso

El índice de refracción y la dispersión refractiva son propiedades físicas fundamentales que se pueden usar junto con otras propiedades para caracterizar los hidrocarburos puros y sus mezclas.

1.1.4.3 Relación del Índice de Viscosidad con el Índice de Refracción

Periódicamente el Instituto Colombiano del Petróleo (ICP) actualiza las constantes m y b que relacionan en forma lineal el Índice de Viscosidad (IV) y el Índice de Refracción (IR). Estas variaciones son debidas a los cambios en las características de la carga tales como su densidad y composición.

En la [Tabla 1.1](#) se relacionan los valores típicos de las últimas pruebas entregadas por el ICP (Septiembre del 2000) y la [Figura 1.6](#) traza su comportamiento lineal inverso.

$$IV = m * IR + b$$

PRODUCTO	m	b
Destilado Medio	-1990.3	2994.9
Destilado Liviano	-1914.1	2865.0
DAO	-1678.2	2561.9

Tabla 1.1 Conversión de IR a IV

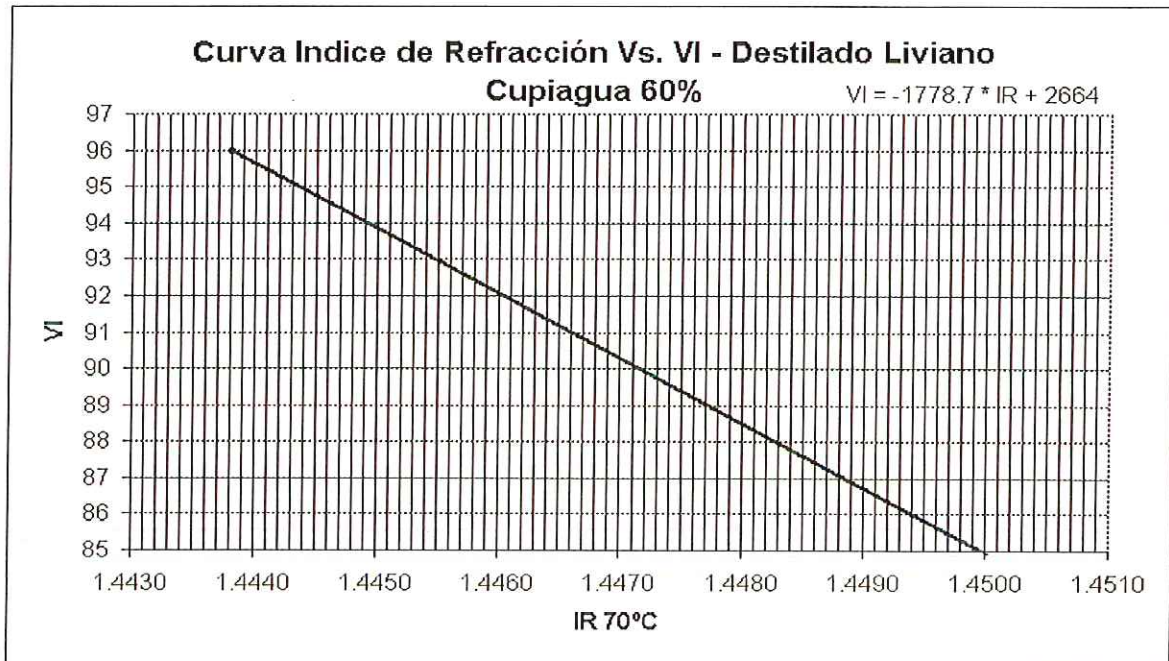


Figura 1.6 Curva IR Vs. VI

1.1.5 Análisis en Línea (El Refractómetro)

1.1.5.1 Principio de Funcionamiento

El refractómetro de proceso determina el índice de refracción (IR) de la solución del proceso midiendo el ángulo crítico de refracción. La luz proveniente de una fuente lumínica (L), (Figura 1.7) se dirige contra la interfase entre un prisma (P) y la solución de proceso (S). Los rayos de luz se encuentran con esta superficie a distintos ángulos. Los rayos reflejados forman una imagen (ACB), donde (C) es la posición del rayo de ángulo crítico. Los rayos en (A) se reflejan totalmente en la interfase, los rayos en (B) son parcialmente reflejados y parcialmente refractados dentro de la solución del proceso. De esta manera la imagen óptica se divide en una área iluminada (A) y una área oscura (B). La posición del límite (C) entre las áreas muestra el valor del ángulo crítico y por tanto del índice refractivo de la solución del proceso. El índice refractivo normalmente se incrementa al aumentar la concentración.

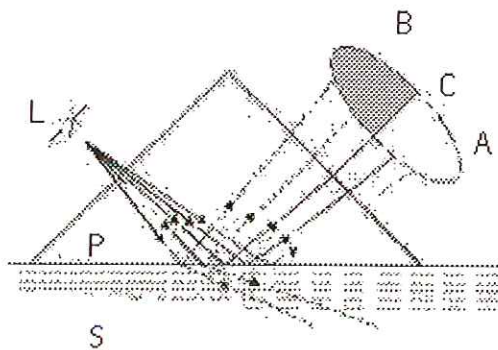


Figura 1.7 Imágenes ópticas

Concentración Baja Concentración Alta

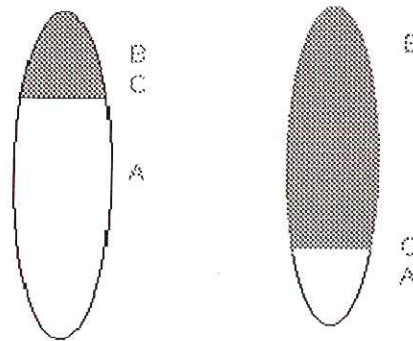


Figura 1.8 Principio de Refracción

De esto se puede concluir que la imagen óptica cambia con la concentración de la solución del proceso como se muestra en la [figura 1.8](#), la imagen óptica es convertida en señal eléctrica por un detector de imagen. Por este método es medida la concentración de la solución. El color de la solución, las bombas de gas o las partículas no disueltas no interfieren con el resultado.

1.1.5.2 Descripción del Sensor

En el sensor del refractómetro de proceso ([figura 1.10](#)), el prisma de medida (A) está montado sobre la superficie oblicua cerca de la punta. La fuente de luz (B) es un diodo emisor de luz.

El refractómetro de proceso utiliza un detector de imagen digital (C) el detector de imagen consiste de 256 foto-celdas en fila, integradas en un solo chip, la salida del detector de imagen es un tren de pulsos como se muestra en la [figura 1.11](#). Este número de pulsos altos corresponde a la posición del eje sombreado en la imagen óptica. El número de pulsos altos es una medida directa del ángulo crítico.

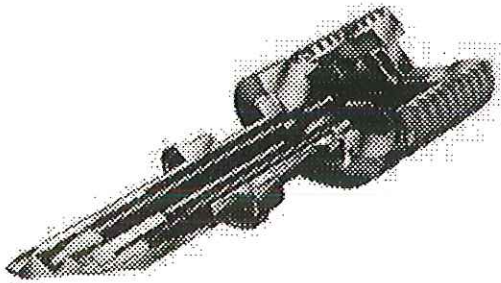


Figura 1.9 Sensor Real

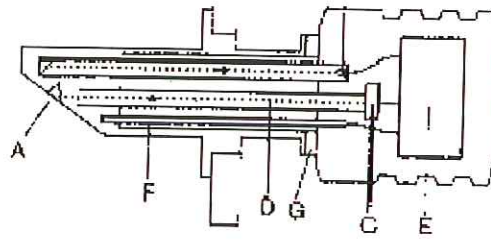


Figura 1.10 Esquema Sensor Real

El digitalizador de imagen (E) transforma este tren de pulsos a una señal digital serial. Esta señal serial transmite un paquete que contiene una descripción completa de la imagen óptica y datos de temperatura al transmisor indicador.

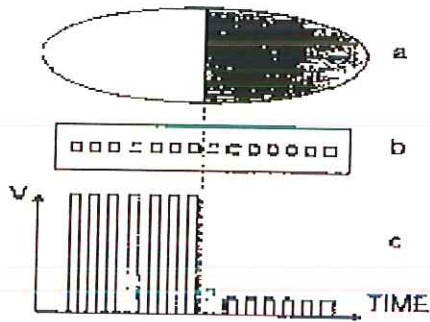


Figura 1.11 Tren de Pulsos

Para la compensación automática de temperatura, la punta del sensor contiene una sonda de temperatura del proceso (F). El sensor de imagen digital (C) se separa del calor del proceso por fibra óptica (D) y el aislador térmico (G) es montado en la cabeza del sensor enfriada por aire

1.1.5.3 El Monitor

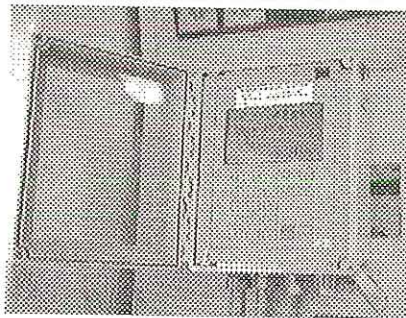


Figura 1.12 Indicador del Trasmisor (Monitor)

El monitor (figura 1.12) recibe una señal serial del sensor que describe la imagen óptica y además entrega la temperatura del proceso. El sistema de microprocesador construye internamente la imagen óptica (figura 1.14) e implementa un algoritmo de análisis de

imagen (figura 1.13), la cual identifica la posición exacta del eje sombreado mostrado en la figura 1.11.

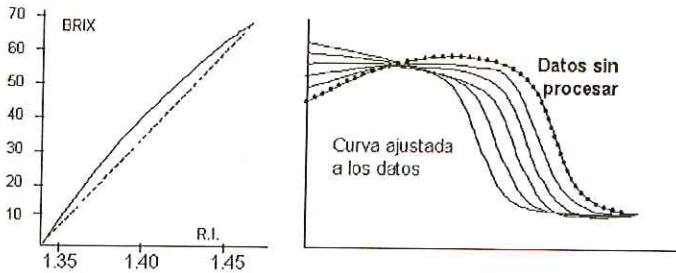


Figura 1.13 Diagrama BRUX e Imagen del algoritmo de Análisis

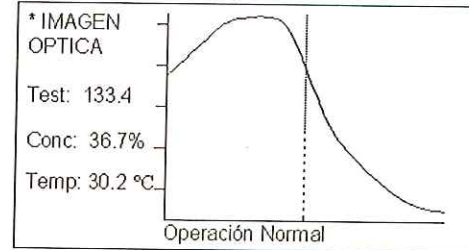


Figura 1.14 Imagen Óptica del Sensor Real

El transmisor indicador contiene un suministro de potencia, un sistema microprocesador y un panel frontal con una pantalla de cristal líquido (LCD) y un teclado. Las señales de salida son señales de concentración de 4 a 20 miliamperios y una señal de salida serial, RS 232 o RS 485 como alternativas. El transmisor indicador también acepta cuatro contactos de conmutación de entrada para el Hold de la señal o selección de escala. Un bus serial conecta el transmisor indicador a las unidades externas tal como la unidad de relevos o la unidad de corriente externa.

El sistema microprocesador alinea la lectura de concentración, como se muestra en el ejemplo de la figura 1.11, y realiza una compensación automática de temperatura.

1.1.5.4 El Sensor Real Instalado

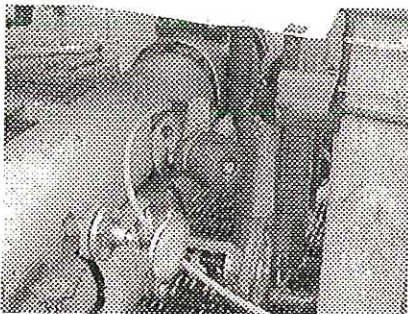


Figura 1.15 Instalación del Sensor

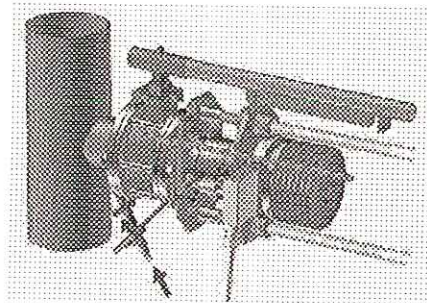


Figura 1.16 Sensor Real (Interno)

El sensor está diseñado para ser montado directamente sobre la línea de proceso, con el eje principal en forma horizontal, sin obstáculos para el flujo de aire alrededor de su cabeza, ni exposición a radiación de alta temperatura, normalmente debe tener suministro de enfriamiento de aire para su conservación, si la temperatura ambiente es mayor que 45 °C, el aire de enfriamiento debe ser mejorado aplicando aire presurizado contra la cubierta del sensor.

1.1.6 Análisis en Laboratorio

Esta práctica especifica los procedimientos para calcular el índice de viscosidad de productos derivados del petróleo, tales como aceites lubricantes y materiales relacionados a partir de sus viscosidades cinemáticas a 40 y 100 °C. Para ello existe dos procedimientos:

Procedimiento A. Para productos derivados del petróleo de índice de viscosidad menor o igual a 100.

Procedimiento B. - Para derivados del petróleo cuyo índice de viscosidad es mayor de 100

La tabla 1.2 dada en esta norma se aplica a los productos derivados del petróleo con viscosidades cinemáticas entre 2 y 70 mm²/s (cSt) a 100 °C.

Las ecuaciones están dadas para calcular el índice de viscosidad de productos derivados del petróleo que tengan viscosidades cinemáticas por encima de 70 mm²/s (cSt) a 100 °C.

Los valores de viscosidad cinemática se determinan tomando como referencia un valor de 1.0038 mm²/s (cSt) a 20.00 °C para el agua destilada. La determinación de la viscosidad cinemática de un producto derivado del petróleo se debe llevar a cabo de acuerdo con los métodos de prueba D 445, IP 71, ISO 3104 o ISO 2909.

Un índice de viscosidad alto indica una pequeña disminución en la viscosidad cinemática con el incremento de la temperatura de un lubricante

VC:cSt	L	H	VC:cSt	L	H	VC:cSt	L	H	VC:cSt	L	H	VC:cSt	L	H	VC:cSt	L	H
2.00	7.99	6.39	7.20	82.39	50.69	12.4	213.6	113.3	17.6	392.7	189.5	25.6	769.3	332.7	45.0	2152	780.9
2.10	8.64	6.89	7.30	84.53	51.78	12.5	216.6	114.7	17.7	396.7	191.3	25.8	779.7	336.7	45.5	2197	794.5
2.20	9.31	7.41	7.40	46.66	52.88	12.6	219.6	116.0	17.8	400.7	192.9	26.0	790.4	340.5	46.0	2243	808.2
2.30	10.00	7.94	7.50	88.85	53.98	12.7	222.6	117.4	18.9	404.6	194.6	26.2	801.6	344.4	46.5	2288	821.9
2.40	10.71	8.50	7.60	91.04	55.09	12.8	225.7	118.7	18.0	408.6	196.2	26.4	812.8	348.3	47.0	2333	835.5
2.50	11.45	9.06	7.70	93.20	56.20	12.9	228.8	120.1	18.1	412.6	197.8	26.6	824.1	352.3	47.5	2380	849.2
2.60	12.21	9.65	7.80	95.43	57.31	13.0	231.9	121.5	18.2	416.7	199.4	26.8	835.5	356.4	48.0	2426	863.0
2.70	13.00	10.25	7.90	97.72	58.45	13.1	235.0	122.9	18.3	420.7	201.0	27.0	847.0	360.5	48.5	2473	876.9
2.80	13.80	10.87	8.00	100.00	59.60	13.2	238.1	124.2	18.4	424.9	202.6	27.2	857.5	364.6	49.0	2521	890.9
2.90	14.63	11.50	8.10	102.30	60.74	13.3	241.2	125.6	18.5	429.0	204.3	27.4	869.0	368.3	49.5	2570	905.3
3.00	15.49	12.15	8.20	104.60	61.89	13.4	244.3	127.0	18.6	433.2	205.9	27.6	880.6	372.3	50.0	2618	919.6
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
6.90	75.72	47.51	12.1	204.8	109.4	17.3	381.0	184.9	25.0	736.5	320.9	43.5	2021	741.3	69.5	4839	1,541.0
7.00	78.00	48.57	12.2	207.8	110.7	17.4	384.9	186.5	25.2	747.2	324.9	44.0	2064	754.4	70.0	4905	1,558.0
7.10	80.25	49.61	12.3	210.7	112.0	17.5	388.9	188.9	25.4	758.2	328.8	44.5	2108	767.6			

Tabla 1.2 Valores del L y H para Viscosidad Cinemática en Sistemas 40-100 °C

1.1.6.1 Cálculo en laboratorio - Procedimiento A.

Se utiliza para aceites con Índice de viscosidad menor o igual a 100.

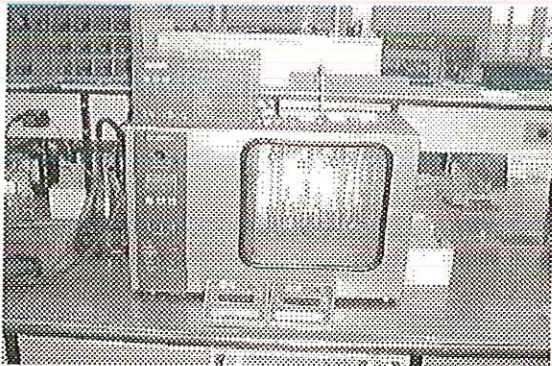


Figura 1.17 Viscosímetro de Laboratorio (<100 C)

Si la viscosidad cinemática de los aceites a 100 °C es menor o igual 70 cSt mm²/s (cSt), tome de la tabla 1.2 los valores correspondientes para L y H. Los valores medidos que no se encuentran listados pero se encuentran dentro del rango de la tabla 1.2, se pueden obtener por interpolación lineal.

El índice de viscosidad no está definido y no se puede reportar para aceites de viscosidades cinemáticas menores que 2.0 mm²/s (cSt) a 100 °C

Si la viscosidad cinemática se encuentra por encima de 70 mm²/s (cSt) a 100 °C, calcule los valores de L y H como sigue:

$$\text{Ecuación 1.1} \quad L = 0.8353 Y^2 + 14.67 Y - 216$$

Ecuación 1.2 $H = 0.1684 Y^2 + 11.85 Y - 97$

Donde:

L = Viscosidad cinemática, en mm²/s (cSt), a 40 °C de un aceite de índice de viscosidad 0 que tenga la misma viscosidad cinemática a 100 °C que el aceite al cual se le calcula el índice de viscosidad.

Y = Viscosidad cinemática, en mm²/s (cSt), a 100 °C del aceite al cual se debe calcular el índice de viscosidad.

H = Viscosidad cinemática, en mm²/s (cSt), a 40 °C de un aceite de índice de viscosidad 100 que tenga la misma viscosidad cinemática a 100 °C que el aceite al cual se le debe calcular el índice de viscosidad.

Calcule el índice de viscosidad, VI, del aceite como sigue:

Ecuación 1.3 $VI = [(L - U)/(L - H)] \times 100$

Donde:

U = Viscosidad cinemática, en mm²/s (cSt) a 40 °C de aceite cuyo índice de viscosidad debe ser calculado.

Ejemplo de cálculo:

Viscosidad cinemática medida a 40 °C del aceite cuyo índice de viscosidad debe ser calculado = 73.30 mm²/s (cSt).

Viscosidad cinemática a 100 °C del aceite cuyo índice de viscosidad debe ser calculado = .86mm²/s (cSt).

De la tabla 1.2 (por interpolación) L = 119.94.

De la tabla 1.2 (por interpolación) H = 69.48.

Sustituyendo en la ecuación 1.3 y redondeando lo más cerca el número total:

$$VI = [(119.94 - 73.30)/(119.94 - 69.48)] \times 100 = 92.43$$

$$VI = 92$$

1.1.6.2 Cálculo en el laboratorio - Procedimiento B.

Se utiliza para aceites con Índice de Viscosidad de 100 o superior.

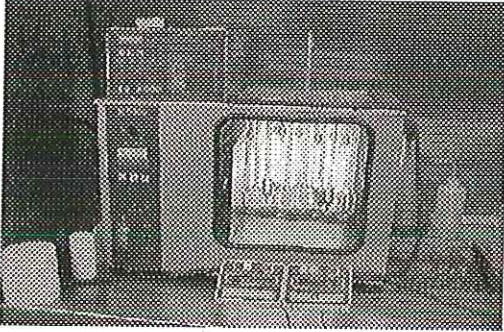


Figura 1.18 Viscosímetro de Laboratorio (>100 C)

Si la viscosidad cinemática del aceite a 100 °C es menor o igual a 70 mm²/s (cSt), tome el valor correspondiente para H de la tabla 1.2. Los valores medios que no se encuentran listados, pero que se encuentran dentro del rango de la tabla 1.2, pueden ser obtenidos por interpolación lineal.

El índice de viscosidad no está definido y no se puede reportar para aceites de viscosidad cinemática menor de 2.0 mm²/s (cSt) a 110 °C.

Si la viscosidad cinemática medida a 100 °C es mayor que 70 mm²/s (cSt), calcule el valor de H como sigue:

$$\text{Ecuación 1.4} \quad H = 0.1684 Y^2 + 11.85 Y - 97$$

Donde:

Y = Viscosidad cinemática, en mm²/s (cSt), a 100 °C del aceite al cual se le debe calcular el índice de viscosidad.

H = Viscosidad cinemática, en mm²/s (cSt), a 40 °C de un aceite de índice de viscosidad 100 que tiene la misma viscosidad cinemática a 100 °C que el aceite al cual se debe calcular el índice de viscosidad.

Calcule el índice de viscosidad, VI, del aceite como sigue:

$$\text{Ecuación 1.5} \quad VI = [(\text{antilog } N) - 1]/0.00715 + 100$$

Donde:

$$\text{Ecuación 1.6} \quad N = (\log H - \log U)/\log Y \quad \text{ó} \quad Y^N = H/U \quad \text{con:}$$

U = Viscosidad cinemática, en mm²/s (cSt), a 40 °C del aceite al cual se debe calcular el índice de viscosidad.

Ejemplo de cálculo:

Viscosidad cinemática medida a 40 °C del aceite al cual se le debe calcular el índice de viscosidad = 22.83 mm²/s (cSt).

Viscosidad cinemática a 100 °C del aceite al cual se le debe calcular el índice de viscosidad = 5.05 mm²/s (cSt).

De la tabla 1.2 por interpolación H = 28.97

Substituyendo por la ecuación 1.6 y por medio de logaritmos:

$$N = [(\log 28.97 - \log 22.83)/\log 5.05] = 0.14708$$

Substituyendo en la ecuación 1.5 y redondeando al número entero más cercano:

$$VI = [(\text{antilog } 0.14708) - 1]/0.00715] + 100$$

$$VI = [(1.40307 - 1)/0.00715] + 100 = 156$$

Viscosidad cinemática medida a 40 °C del aceite al cual se debe calcular el índice de viscosidad = 53.47 mm²/s (cSt).

Viscosidad cinemática a 100 °C el aceite al cual se debe calcular el índice de viscosidad = 7.80 mm²/s (cSt).

De la tabla 1.2, H = 57.31

Substituyendo en la ecuación 1.6 por medio de logaritmos:

$$N = [(\log 57.31 - \log 53.47)/\log 7.80] = 0.03376$$

Substituyendo en la ecuación 1.5 y redondeando al número entero más cercano:

$$VI = [(\text{antilog } 0.03376) - 1]/0.00715] + 100$$

$$VI = [(1.08084 - 1)/0.00715] + 100 = 111$$

1.1.7 Diseño Funcional del Sistema de Control

En la actualidad, la unidad de extracción de bases lubricantes con fenol posee 27 lazos de control regulatorios, utilizando bloques de control tipo PID (Proporcional Integral y Derivativo), con la acción derivativa ligeramente o poco usada por su alto efecto sobre las variables a controlar. Tres de ellos están en cascada con otro controlador (temperatura primaria con presión secundaria, presión primaria con temperatura secundaria y nivel primario con flujo secundario). Seis controladores son de temperatura (TIC), siete de flujo (FIC), seis de presión (PIC), seis de nivel (NIC) y dos válvulas de control manual (MIC). [Ordóñez] muestra la descripción en detalle de cada uno de ellos y allí mismo se describe las estrategias de control avanzado para la unidad.

El objetivo de la unidad de extracción es el de operar máximo rendimientos de producto valioso a un mínimo de calidad regalada, optimizando la relación de tratamiento y la sección de recobro a su vez en esta última el objetivo es minimizar los consumos de energía, optimizando el control del horno y las torres despojadoras. Mientras que en la sección de extracción es aumentar el rendimiento de base parafínica, la estabilidad de la unidad, su disponibilidad y transferencia de información en tiempo real.

Una combinación entre el control multivariable y su respectiva estrategia de optimización con alguna estrategia conocida de control avanzado, servirían para producir los máximos beneficios en la unidad, mientras se minimizan energía y pérdidas en rendimiento.

El desafío del control es el de eliminar las perturbaciones que se presentan en los cambios de carga tan pronto como sea posible sin bajar rendimientos, maximizar la producción de refinato aumentando del rendimiento y/o carga.

Una propuesta para llevar a cabo esta aplicación, es descrita por [Ordóñez], utilizando tecnología de control MIMO (múltiples entradas múltiples salidas), las cuales se apoyan en modelos de propiedades inferidas y modelos de identificación de procesos, incluyendo compensación para todas las perturbaciones de las variables, contando para esto con tecnología de modelos de control predictivo.

El objetivo de la principal estrategia de control avanzado a implementar para nuestro propósito con la utilización del sensor virtual, es la de mantener el índice de Viscosidad o Refractividad del refinado producto dentro de especificación, lo más cerca posible del límite. Para la torre de extracción, las variables manipuladas primarias son la temperatura del fenol, el nivel de interfase, la relación fenol-carga y la relación fenol-carga a la torre. Este control puede residir en el computador supervisorio.

El Índice de Viscosidad puede ser inferido a partir de:

- Temperatura de cima de la torre de extracción
- Diferencia de temperatura entre la cima y el fondo
- Carga a la torre extractora
- Rata de fenol
- Calidad del fenol
- Presión de la torre

Los datos reportados por el analizador en línea servirán para entrenar al sensor virtual y así tener una medida del Índice de refracción e Índice de viscosidad con mayor disponibilidad y confiabilidad, pudiéndose implementar el control requerido.

Diseño Funcional del Sistema de Control Planta de Extracción de Bases Lubricantes con Fenol

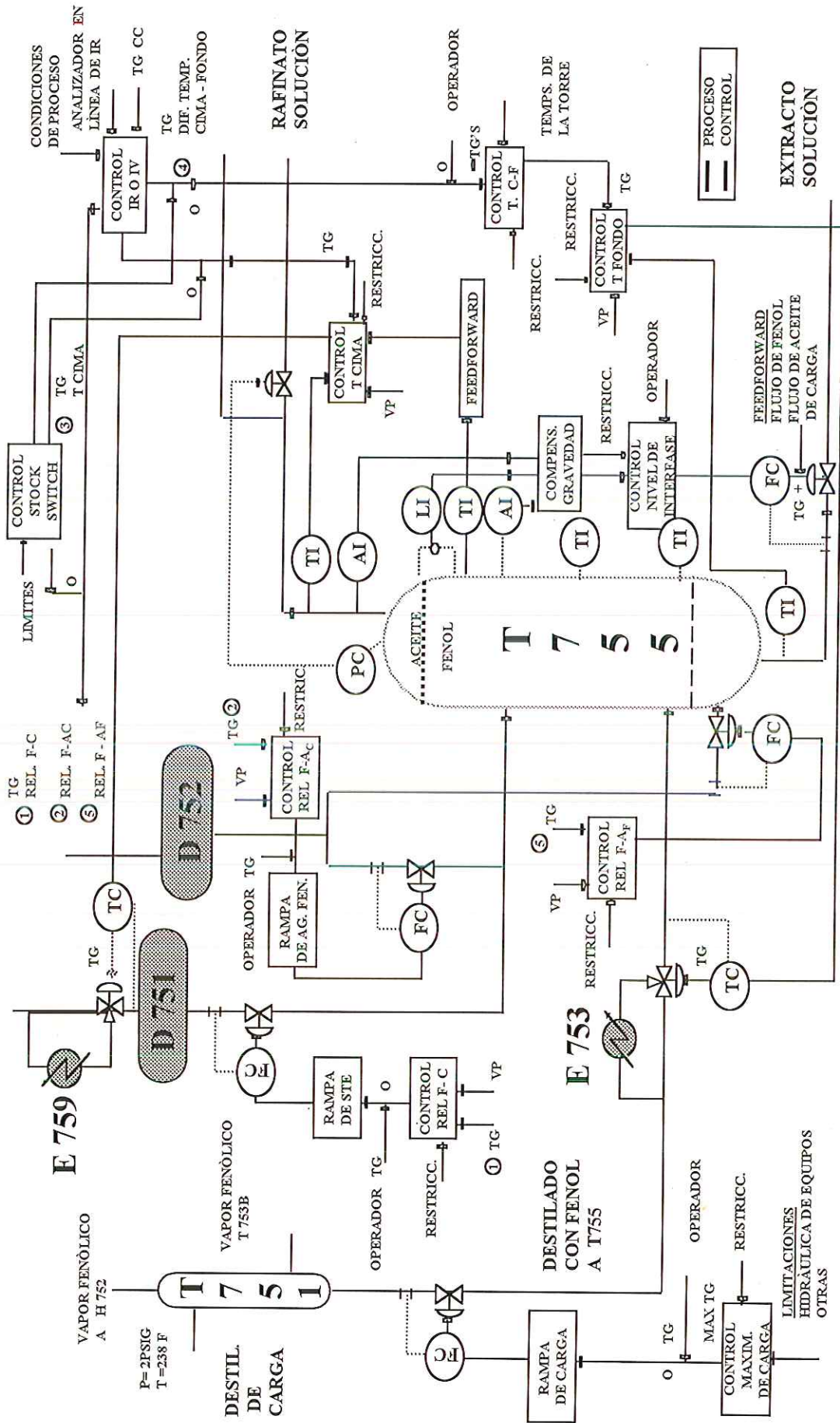


Figura 1.19 Resumen de las estrategias de control avanzado propuestas

1.2 ALTERNATIVAS PARA PREDECIR EL IV – IR

La medición de la calidad de los productos requiere control en línea y frecuentemente no está disponible o presenta grandes retrasos. Las técnicas de sensores de software o sensores inteligentes son métodos atractivos y eficientes para predecir dichas variables. Hoy en día hay tres clases de aproximaciones para construir modelos usados como sensores por software en los procesos: modelos mecánicos, modelos de regresión estadística y modelos de inteligencia artificial. Debido a la no linealidad, complejidad e incertidumbre de los procesos químicos en la práctica, no es fácil obtener modelos mecánicos. Los modelos basados en métodos estadísticos de manejo de datos o las redes neuronales artificiales proveen una alternativa útil en este campo.

La aproximación típica al modelo de manejo de datos ha sido el modelaje no lineal con un modelo simple. Para el modelo basado en redes neuronales artificiales, hay muchos métodos de selección de estructuras óptimas y así obtener el menor error predictivo. Estos modelos con frecuencia tienen baja robustez y problemas de sobredimensionamiento. Pero en principio, dado que no hay restricciones en el tamaño de la red y hay suficientes datos, se puede encontrar una red neuronal artificial para resolver el problema dado, aún cuando en la práctica y con frecuencia, los problemas industriales se pueden definir usando una cantidad limitada de datos altamente confiables más que por las capacidades de generalización de la red.

[Bates y Granger] proponen la idea de combinar modelos para mejorar la precisión en la predicción. Hoy es ampliamente aceptado que la predicción combinada sea una simple, pragmática y sensible forma de producir mejores predicciones. Arquitecturas de modelos múltiples combinadas con el algoritmo FCM (fuzzy c-means clustering) que separa un conjunto de datos entrenados en varios grupos (clusters) con diferentes centros, en donde cada subconjunto es entrenado por la función de red de base radial (radial basis RBFN) o por el algoritmo de mínimos cuadrados parciales (PLS); se ha encontrado que el uso del algoritmo PLS es indicado para el manejo de datos limitados y el desempeño del RBFN es indicado para modelar sistemas no lineales complejos con cantidad suficiente de muestras. De acuerdo con esta idea, los modelos basados en PLS son construidos por un subconjunto de pocas muestras, mientras que el modelo basado en RBFN está construido por un subconjunto de muchas muestras.

En este trabajo se presenta el modelo propuesto por [H.A.B. Te Braake, G. Van Straten] y otros, permitiendo combinar diferentes arquitecturas por el algoritmo de RAWN (Random Activation Weight Network, red de activación de pesos aleatorios).

Luego aplicamos el método a un problema práctico de modelar la obtención de la calidad del refinado con un sensor virtual de índice de refracción, por el cual se infiere el índice de viscosidad de una planta de extracción de aceites lubricantes con fenol. Todos los resultados demuestran la promesa de esta aproximación en el mejoramiento del modelaje de sensores no lineales por software.

1.2.1 Sensores inteligentes

Un uso cada vez más frecuente de las redes neuronales artificiales, en áreas relacionadas con el control de procesos, lo constituye el desarrollo de sensores virtuales o sensores inteligentes, basados en modelos no lineales. Este tipo de aplicaciones es muy útil para el control inferencial y el seguimiento de variables críticas en los procesos. Se fundamenta en establecer un modelo de red neuronal que permita inferir valores de variables de proceso de difícil medición en función de otras variables de fácil medición y continua, utilizando instrumentos sencillos y robustos como transmisores de presión, temperatura, nivel y flujo. En general las variables de difícil medición tales como el Índice de Viscosidad y el Índice de Refracción requieren analizadores fuera de línea, introduciendo retardos o costosos analizadores en línea que requieren frecuentes calibraciones.

Si la red neuronal está bien entrenada, estos sensores virtuales se pueden utilizar también para la estimación dinámica de variables de estado en controladores basados en modelos, como variable controlada en el control convencional usando PID y para monitorear el funcionamiento de los analizadores.

Muy frecuentemente en las torres de destilación, las composiciones de cima y fondo se infieren midiendo solamente temperaturas en platos cercanos a la cima y al fondo de la torre. Esto es aceptable, para sistemas binarios con un buen control de la presión, pues en estos casos existe una relación bien definida entre la temperatura y la composición.

Para el caso de multicomponentes, esta relación no está bien definida y por lo tanto la incorporación de otras variables de proceso, mediante el uso de un sensor virtual, mejora notablemente la inferencia continua de la composición u otra propiedad de interés y por lo tanto el comportamiento de los sistemas de control.

El comportamiento no lineal de estos procesos de la industria petroquímica, los convierte en ejemplos interesantes para el estudio y uso de las redes neuronales como sensores inteligentes.

1.2.2 Redes neuronales con alimentación hacia adelante.

La red neuronal a ser considerada, es una red neuronal con alimentación hacia adelante, con una capa oculta (ver [figura 1.20](#)), la cual puede ser expresada de la siguiente forma, la entrada para la función de activación no lineal está dada por:

$$\text{Ecuación 1.7 } z_j(k) = \sum_{i=1}^{N_i} W_{ij}^h x_i(k) + b_j^h$$

Donde W_{ij}^h es el peso de activación entre la entrada i y la neurona j . El índice k denota el k -ésimo evento. N_i es el número de entradas y N_h es el número de neuronas en la capa oculta. La salida de la neurona de la capa oculta está dada por:

$$\text{Ecuación 1.8 } v_j(k) = f(z_j(k)).$$

La salida de la red neuronal completa está dada por:

$$\text{Ecuación 1.9 } y_l(k) = \sum_{j=1}^{N_h} W_{jl}^o v_j(k) + b_l^o$$

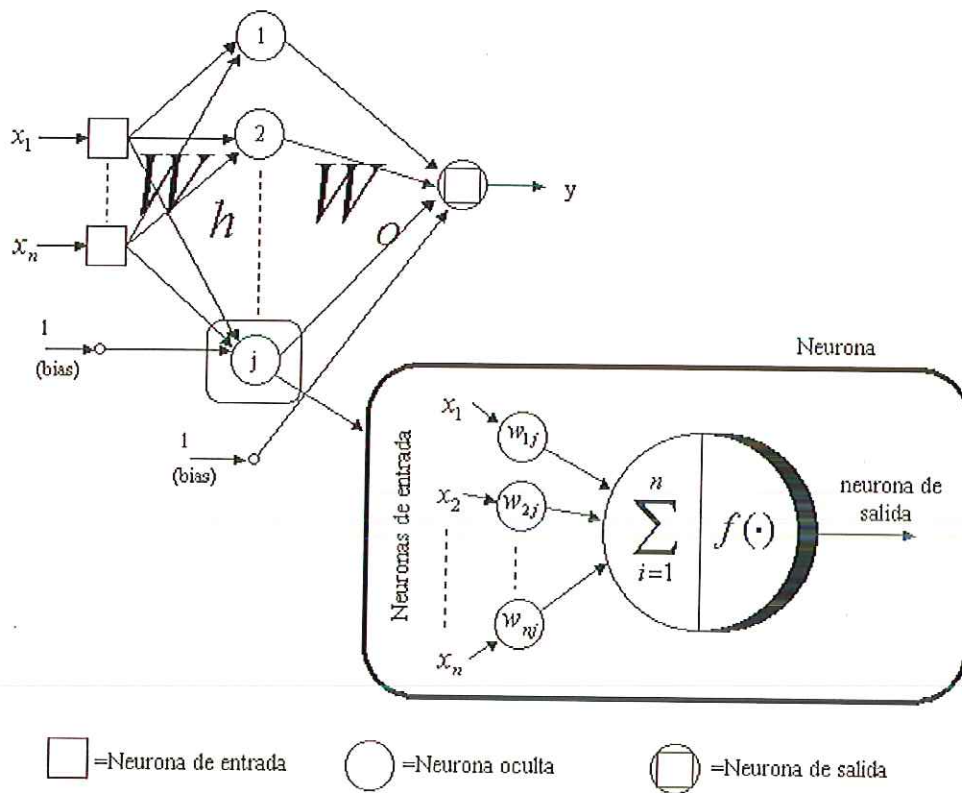


Figura 1.20 Ejemplo de una red neuronal alimentada hacia adelante. Los detalles muestran la también llamada “neurona de McCulloch-Pitts”

Los pesos W_{ij}^h son llamados los pesos de “activación” y los pesos W_{ji}^o los pesos de “salida”. El ajuste de activación es denotado como b^h y el ajuste de salida como b^o . La función $f(\cdot)$ es llamada la función de activación. Esta función puede ser cualquier función sigmoideal o Gaussiana. En este tratamiento se utiliza la función \tanh . Esta función tiene las siguientes propiedades:

$$\tanh(t) \rightarrow \begin{cases} 1, & \text{para } t \rightarrow +\infty \\ -1, & \text{para } t \rightarrow -\infty \end{cases}$$

La derivada de la $\tanh(x/e)$, $e > 0$, está dada por $(1/e)(1 - \tanh^2(x/e))$ y converge en la función de DIRAC a medida que $e \rightarrow 0$, o él $\lim_{e \rightarrow +0} (1/e)(1 - \tanh^2(x/e)) = \delta(x)$

Otras propiedades que no son constantes aumentan de manera continua y está limitada, la idea de las dos últimas propiedades el teorema de aproximación de [Funahashi (1989)], se podrá aplicar. Este teorema dice que para un $\delta > 0$ arbitrario existe una red neural de alimentación hacia delante con máscara oculta, con una no constante incrementándose continuamente y con una función de activación ilimitada que se mapea como una función de entrada y salida:

$$y = \tilde{f}(x_1, \dots, x_n) \text{ de tal forma que } \max_{x \in k} |f(x_1, \dots, x_n) - \tilde{f}(x_1, \dots, x_n)| < \varepsilon$$

(k es un subconjunto compacto de \mathfrak{R}^n y $f(x_1, \dots, x_n)$ es una función continua de valores reales en K). Note que este problema no especifica cuantas neuronas se requieren o cuales son los valores de los pesos que deben tener con el fin de alcanzar una barrera de aproximación pre-específica. Ahora definimos la siguiente matriz de entrada:

$$X = [x(1), x(2), \dots, x(k), \dots, x(N_e)]^r$$

Donde $x(k) = [x_1(k) \dots x_i(k) \dots x_{N_i}(k) 1]$ es el vector de entrada (incluyendo uno usado para los bias) y N_e es el número de los llamados eventos. La matriz de salida se da igualmente como:

$$Y = [y(1), y(2), \dots, y(k), \dots, y(N_e)]^r$$

Donde $y(k) = [y_1(k) \dots y_{N_o}(k)]$ es el vector de salida.

La red neural puede ser expresada concisamente por:

$$\begin{aligned} Z &= XW^h [N_e \times N_h] \\ \text{Ecuación 1.10 } V &= f(Z) [N_e \times N_h] \\ Y &= Vb^0 [N_e \times N_o] \end{aligned}$$

Las matrices Z y V contienen resultados intermedios. La matriz $V_b \in \mathcal{V}^{N_e \times (N_h+1)}$ es igual a $V \in \mathcal{V}^{N_e \times N_h}$, excepto que una columna con unos se adiciona para expresar el ajuste de salida b^0 . El conjunto V es un conjunto compacto real con valores entre -1 y 1. Los pesos de activación se agrupan juntos en una matriz $W^h \in \mathfrak{R}^{[N_i+1] \times N_h}$ y los pesos de salida en una matriz $W^0 \in \mathfrak{R}^{[N_h+1] \times N_o}$.

1.2.3 La Red Neuronal Con Activación De Pesos Aleatorios (Rawn)

La función de aproximación general puede ser obtenida por redes neuronales retroalimentadas consistentes de sólo una capa oculta y neuronas no lineales. La innovación descrita en este documento es que el entrenamiento de los pesos entre las entradas y la capa oculta no se requiere. Tomando estos pesos de activación como aleatorios, el problema resultante es lineal en parámetros y puede ser fácilmente resuelto por métodos ordinarios estándar de mínimos cuadrados. Aquí se demuestra que utilizando tales redes con pesos de activación aleatoria (RAWNs), se puede obtener un excelente mapeo, siempre y cuando los pesos de activación sean escogidos de tal forma de que la matriz de regresión no sea singular. Se pueden alcanzar mejoras más adelante por regularización y con la selección adecuada de la señal de excitación usada para el entrenamiento. Se halla que: (i) se obtienen errores mucho más pequeños comparado con las redes de backpropagation utilizando los mismos grados de libertad, (ii) el mapeo depende sólo un poco de los valores reales de los pesos ocultos, a condición de que la red tenga suficientes neuronas y (iii) debido a que no se necesitan iteraciones, la velocidad de computación es incomparablemente más rápida que el backpropagation. Estas propiedades hacen que la red RAWN sea particularmente adecuada para aplicaciones de control.

Las redes neuronales estáticas pueden ser vistas como un mapeo no lineal conveniente del espacio de las variables independientes al espacio de las variables dependientes. Un tipo de red neuronal de alimentación hacia adelante particularmente útil, consiste de una capa oculta con función de activación sigmoideal, no lineal y una capa de salida con una solución de activación lineal. La relación entrada-salida de este tipo de red neuronal puede ser descrita como:

$$\text{Ecuación 1.11 } y_p(k) = \sum_{j=1}^{N_h} W_{jp}^o f \left(\sum_{j=1}^{N_i+1} W_{ij}^h x_i(k) \right)$$

donde $x_l, l = 1, \dots, N_i$ es la entrada,

$x_{N_i+1} = 1$ es una entrada prefijada,

$y_p, p = 1, \dots, N_o$, es salida,

$j, j=1, \dots, N_h$ indica el número de neuronas en la capa oculta

W^h, W^o Son los pesos de la matriz en la capa oculta y la capa de salida, respectivamente.

El índice $k, k = 1, \dots, N_e$ representa el número de evento o índice de tiempo.

La función $f(\cdot)$ Es una función de transición sigmoideal.

Ha sido demostrado teóricamente por [Cybenko] que este tipo de red puede aproximar cualquier función continua a cualquier precisión siempre que existan suficientes neuronas. Los elementos de los pesos en las matrices W^o y W^h puede ser vistos como parámetros que deben ser encontrados para obtener un acercamiento aceptable a los datos disponibles. Este proceso es llamado "entrenamiento". El procedimiento usual para entrenar redes neuronales en adelante, es utilizando backpropagation. A pesar de las mejoras tales como la aceleración, rata de aprendizaje adaptativo [Drago], y redes auxiliares [Fortuna], el backpropagation es lento y tiene propiedades de convergencia pobres.

Recientemente, un número de documentos ha enfocado el entrenamiento de redes neuronales continuas de alimentación hacia delante, como un problema de estimación de parámetros [Singhal, Scalero]. Se han reportado mejoras significantes en velocidad y convergencia. También, están disponibles esquemas recursivos [Chen, Billings]. Cualquiera que sea el procedimiento, se necesita gran cantidad de datos fuera de línea para obtener errores residuales aceptables. Por ello, a pesar del progreso significativo, el entrenamiento de las redes neuronales sigue siendo el cuello de botella en las aplicaciones de control en tiempo real.

Una red neuronal del tipo descrito en la ecuación 1.7 con N^i entradas, N_h neuronas en la capa oculta y N^o salidas, tiene $(N_i + 1) * N_h + (N_h + 1) * N_o$ pesos. De este modo, para cualquier utilidad práctica, el número de parámetros llega a ser muy grande. Puesto que, además, la red es no lineal, la hiper-superficie del criterio de error como función de los parámetros puede tener una forma muy irregular y usualmente tiene muchos mínimos locales. A pesar del hecho de obtener un buen mapeo con cualquiera de las funciones de arriba, no hay garantía de encontrar el mínimo global. De hecho, es bastante probable que el mínimo obtenido sea un mínimo local; y así la red será usada solamente con mapeo en el rango de calibración, asegurándose que posea buenas propiedades de

Interpolación local. La forma clásica para probar el desempeño es revisando la salida contra el segundo conjunto de datos independientes.

La novedad básica descrita en este trabajo es obtener muy buenos mapeos tomando tan sólo los pesos aleatoriamente en la capa oculta. En tal caso, la estimación de los pesos en la capa de salida, se vuelve lineal en los parámetros y por lo tanto puede ser fácilmente resuelta por métodos ordinarios estándar de mínimos cuadrados (OLS). Una red construida en esta forma puede ser llamada como red con pesos de activación aleatorios (RAWN). Se demuestra que con algunas precauciones, las redes RAW pueden entrenarse por OLS con excelente desempeño sin la necesidad de iteraciones. Esta propiedad es particularmente útil para aplicaciones de control.

1.2.3.1 El entrenamiento por OLS

La ecuación 1.11 de la red neuronal, puede ser escrita de una manera más conveniente en forma de matriz. Agrupando los vectores de entrada en una matriz, con una fila por cada evento $X = (x(1)...x(k)...x(N_e))^T$, y la salida definida por:

$Y = (y(1)...y(k)...y(N_e))^T$, la red neuronal puede ser escrita como:

$$\begin{aligned} Z &= XW^h && [N_e * N_h] \\ \text{Ecuación 1.12 } U &= f(Z) && [N_e * N_h] \\ Y &= UW^o && [N_e * N_o] \end{aligned}$$

Sea Y_d los datos de salida en el conjunto de entrenamiento pertenecientes a la entrada X, Y_d puede ser considerada como general por el proceso

$$\text{Ecuación 1.13 } Y_d = UW^o + e$$

Donde e es un término de error. Suponga, además, que los pesos W^h fueron fijados; entonces el entrenamiento de la red equivale a encontrar los pesos W^o que minimizan la diferencia entre la salida de la red neuronal $Y = (UW^o)$ y la salida objetivo Y_d . Este es un problema ordinario de mínimos cuadrados. Dado que la matriz U es de rango completo (esto es, tiene rango N_h), la solución de mínimos cuadrados es:

$$\text{Ecuación 1.14 } W^o = (U^T U)^{-1} U^T Y^d$$

La matriz U tiene que ser una matriz con N^h columnas independientes lineales. Si no, la matriz $(U^T U)^{-1}$ es singular y no existe una solución para la ecuación 1.14. Si U fuera formada de Z por una función lineal, U tendría el rango de Z. puesto que XW^h es un producto externo, Z está construido con columnas linealmente independientes. Consecuentemente, U no será de rango completo. Sin embargo, en el caso de una función de activación no lineal, U será de rango completo debido a que varias columnas dependientes de Z son transformadas en una forma no lineal, dado que no hay dos columnas en Z que sean exactamente las mismas. Lo último es fácilmente alcanzable tomando los pesos de activación W^h aleatoriamente. Para mostrar esto, considere el caso donde los pesos son generados aleatoriamente con una distribución normal con $E(w) = \mu$ y $(w) = \sigma^2$, donde w representa un elemento de la matriz W^h . La probabilidad de que una w particular sea generada, denotada por \bar{w} , puede ser calculada usando la siguiente igualdad [Kreyszig]:

$$\begin{aligned} P[w = \bar{w}] &= \lim_{\delta \rightarrow 0} P[\bar{w} - \delta < w \leq \bar{w} + \delta] \\ \text{Ecuación 1.15} \quad &= \lim_{\delta \rightarrow 0} \int_{\bar{w}-\delta}^{\bar{w}+\delta} f(v) dv = 0 \end{aligned}$$

La función continua f es la probabilidad de la función densidad. Este resultado afirma que la probabilidad de obtener un peso exactamente igual a \bar{w} es igual a cero. Ahora asuma que las columnas de la matriz X son linealmente independientes y conocidas a prior. Entonces la probabilidad de obtener una cierta columna \bar{z} de la matriz Z es:

$$\text{Ecuación 1.16 } P[z = \bar{z}] = P \left[\bigcap_{k=1}^{N_e} \left[\sum_{l=1}^{N^l} x_{kl} w_{lj} \right] \right] \leq P \left[\bigcap_{k=1}^{N_e} \left[\bigcup_{l=1}^{N^l} x_{kl} w_{lj} \right] \right]$$

Aplicando las expresiones

$$P\left[\bigcap_{i=1}^{N_e} A_i\right] = [A_1]P[A_2|A_1]..$$

Ecuación 1.17 $P[A_n|A_1 \cap A_2 \cap \dots \cap A_{N_e-1}]$

$$P\left[\bigcap_{i=1}^{N_e} A_i\right] = \sum_{i=1}^{N_i} P[A_i]$$

a la entrada correspondiente N_i y N_e términos de eventos A_i de la ecuación 1.16, podemos concluir que $P[z = \bar{z}] = 0$, debido a que al menos un término de la parte baja del lado derecho de la expresión de la ecuación 1.16 llegará a ser 0 debido a la ecuación 1.15. Si la probabilidad de obtener una cierta columna \bar{z} es igual a cero, entonces la probabilidad de obtener dos columnas iguales es también igual a cero, debido a que $P[B|A] \leq P[A]$.

1.2.3.2 Procedimiento de Entrenamiento

Dada una red neuronal de alimentación hacia adelante con solo una capa oculta, los elementos de las matrices de pesos W^h y W^0 son parámetros que deben ser encontrados con el fin de obtener un ajuste aceptable de los datos disponibles, este proceso es llamado entrenamiento. En los casos de la redes neuronales de alimentación hacia delante, el entrenamiento no es nada más que encontrar el mínimo local o global de las siguientes funciones de criterio:

$$j(W^h, W^0) = \|e\|^2$$

Donde e es un vector con errores de modelamiento y $\|\cdot\|$ es el vector normalizado estándar Euclidiano.

Los métodos de entrenamiento más comúnmente usados, tales como el método de propagación hacia atrás o Levenberg–Marquardt, son métodos basados en gradientes, así estos métodos tratan de encontrar el mínimo descendiendo en la dirección del gradiente negativo [Hassoun]. Los métodos de entrenamiento basados en gradientes tienen dos desventajas principales, primero, la rutina de búsqueda puede quedarse estancada en el mínimo local y segundo, el entrenamiento basado en gradiente es un procedimiento iterativo posiblemente resultando en tiempos de entrenamientos largos antes de que se encuentre el mínimo. En esta sección el esquema RAWN alternativo se

desarrolla con el único objetivo de acelerar el proceso de entrenamiento, sin embargo, ambas aproximaciones pueden aun resultar en soluciones sub-óptimas. La aproximación al entrenamiento de la redes neuronales de pesos por activación reguladas se basa en la división del problema en dos subproblemas mutuamente independientes que pueden ser resueltos óptimamente [Braake 1996]. Sin embargo, no implica que el problema sea resuelto total y óptimamente aunque la aplicación del método propuesto en varios procesos reales haya mostrado que los resultados satisfactorios pueden ser obtenidos, el primer subproblema es la estimación de los pesos de activación W^h , el segundo subproblema es la estimación de los pesos de salida W^o . Como se mostrará de esta manera los subproblemas resultantes tienen parámetros lineales y la solución pueden ser encontrada por algoritmos no iterativos rápidos. Esquemáticamente el procedimiento puede ser dividido en los 2 siguientes pasos:

El cálculo de los pesos de activación W^h .

El cálculo de los pesos de la salida W^o .

Cada paso puede ser dividido en diferentes acciones y cálculos, en la figura 1.21 mostramos un resumen de la aproximación total del RAWN. La primera parte del algoritmo tiene que ver con la estimación de los pesos de activación, esto se basa en la construcción de modelos lineales locales, pero para construir estos modelos lineales, el conjunto de datos primero necesita ser particionado en diferentes subconjuntos, los cuales serán utilizados para la estimación de un modelo lineal local particular, para evitar la saturación de la función de activación, estos subconjuntos necesitan ser escalizados, este es el segundo paso del algoritmo, el tercer paso es, con los parámetros de estos modelos locales lineales que deben ser calculados por técnicas cuadráticas lineales. Los parámetros resultantes de los modelos lineales locales se utilizan como los pesos de activación de la red neuronal. Ahora que los pesos han sido calculados, el último paso del proceso de entrenamiento RAWN es la estimación de los pesos de salida por técnicas cuadráticas lineales estándar. Debe resaltarse en este punto que el algoritmo total no necesita iteraciones en principio a través del conjunto de datos. Cada uno de estos pasos mencionados de construcción deben ser ejecutados solo una vez.

1.2.3.3 Regulación de los Pesos de Activación

El primer paso es el cálculo de los pesos de activación. Se ha demostrado [Te Braake y Van Straten (1995)] que tomando los pesos aleatoriamente, es decir, la aproximación de redes de pesos por activación aleatoria ya se han obtenido resultados de entrenamiento satisfactorios. Una exclusión a esta aproximación se describe aquí, las aproximaciones lineales por partes se utilizarán para escoger los pesos de activación, de allí que se le llaman redes de pesos de activación reguladas. Precauciones como la escogencia de W^h deben ser tomadas debido a que la matriz Z en la ecuación 1.10 debe satisfacer las condiciones generales con el fin de garantizar que la salida cuadrática del problema de activación de pesos, es decir, el segundo subproblema pueda ser resuelto.

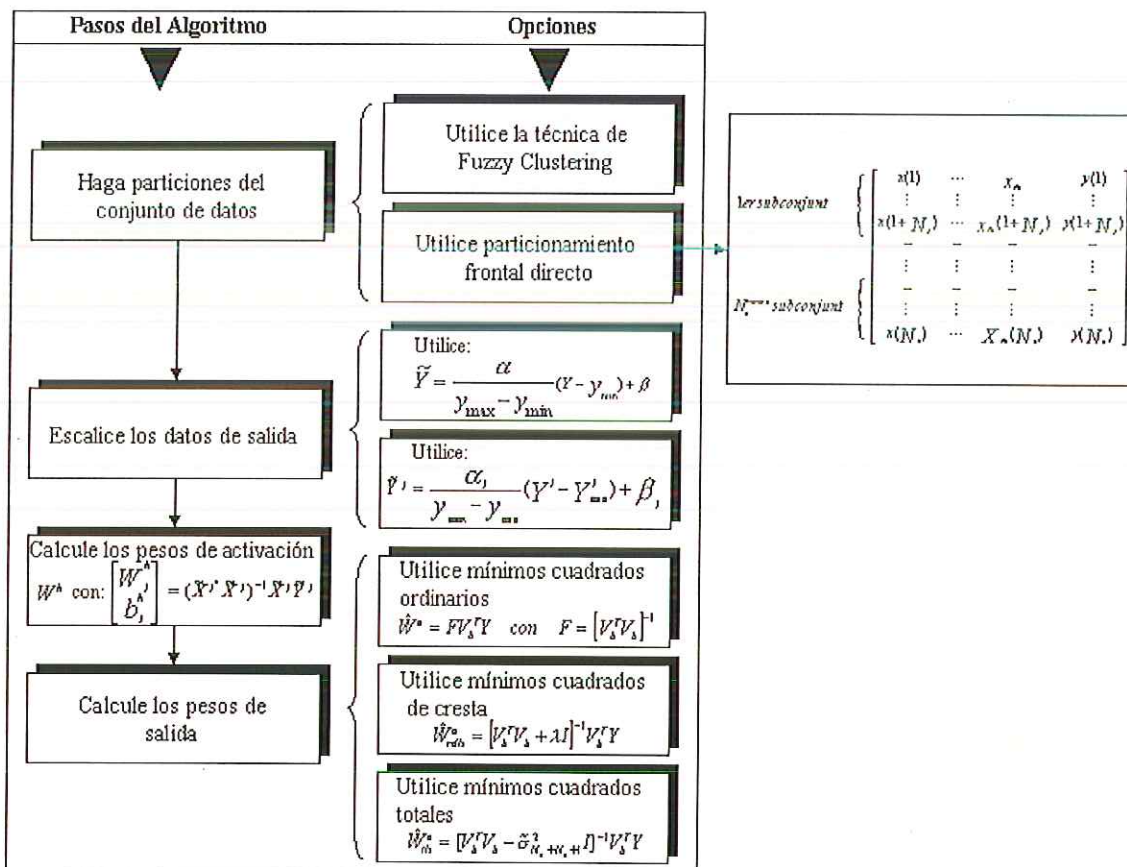


Figura 1.21 Resumen del algoritmo RAWN para entrenar una red neuronal alimentada hacia adelante con una capa oculta

Entonces Z^* permanece después de borrar todas las filas iguales de Z . Es decir, si existen columnas de Z que son iguales unas a otras, entonces estas columnas también serán iguales en la matriz V . En ese caso V no es de rango completo.

1.2.3.4 Funciones de Activación

Entre las unidades o neuronas que forman una red neuronal artificial existe un conjunto de conexiones que unen unas a otras. Cada unidad transmite señales a aquellas que están conectadas con su salida. Asociada con cada unidad U_i hay una función de salida, $f_i(a_i(t))$ que transforma el estado actual de activación $a_i(t)$ en una señal de salida $y_i(t)$, es decir:

$$y_i(t) = f_i(a_i(t))$$

El vector que contiene las salidas de todas las neuronas en un instante t es:

$$Y(t) = (f_1(a_1(t)), f_2(a_2(t)), \dots, f_i(a_i(t)), \dots, f_N(a_N(t)))$$

En algunos modelos esta salida es igual al nivel de activación de la unidad, en cuyo caso la función f_i es la función identidad, $f_i(a_i(t)) = a_i(t)$. A menudo, f_i es de tipo sigmoïdal, y suele ser la misma para todas las unidades.

Existen cuatro funciones de transferencia típicas que determinan distintos tipos de neuronas: Función escalón, Lineal y mixta, Sigmoïdal y Gaussiana.

La función escalón o umbral únicamente se utiliza cuando las salidas de la red son binarias (dos posibles valores). La salida de una neurona se activa solo cuando el estado de activación es mayor o igual que cierto valor umbral. La función lineal o identidad equivale a no aplicar función de salida. Se usa muy poco. Las funciones mixta y sigmoïdal son las más apropiadas cuando queremos como salida, información analógica.

Neuronas de función Sigmoïdal

Cualquier función definida simplemente en un intervalo de posibles valores de entrada, con un incremento monótonico y que tenga ambos límites superiores e inferiores (por

ejemplo, las funciones sigmoide o arco tangente), podrá realizar la función de activación o de transferencia de forma satisfactoria.

Con la función sigmoide, para la mayoría de los valores del estímulo de entrada (variable independiente), el valor dado por la función es cercano a uno de los valores asintóticos. Esto hace que en la mayoría de los casos, el valor de salida este comprendido en la zona alta o baja del sigmoide. De hecho, cuando la pendiente es elevada, esta función tiende a la función escalón. Sin embargo, la importancia de la función sigmoide (o cualquier otra función similar) es que su derivada es siempre positiva y cercana a cero para los valores grandes positivos o negativos; además, toma su valor máximo cuando x es 0. esto hace que se puedan utilizar las reglas de aprendizaje definidas para las funciones escalón, con la ventaja, respecto a esta función, de que la derivada esta definida en todo el intervalo. La función escalón no podía definir la derivada en el punto de transición, y esto no ayuda a los métodos de aprendizaje en los cuales se usan derivadas.

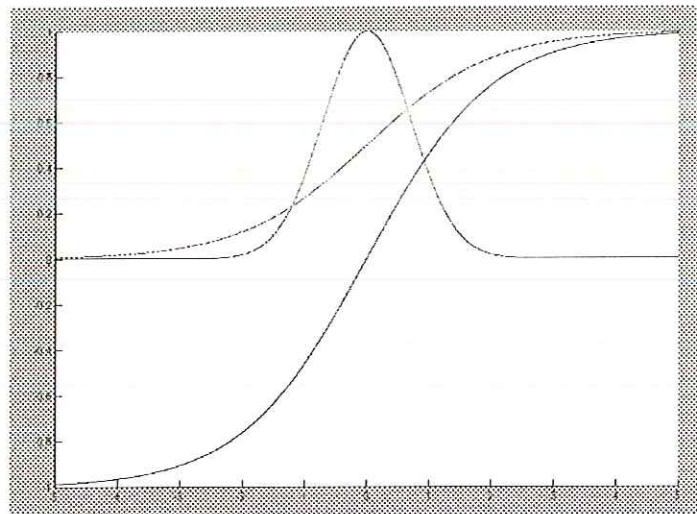


Figura 1.22 Funciones de Activación Continua

Sigmoide	$f(x) = \frac{1}{1 + e^{-ax}}$	(rojo)
----------	--------------------------------	--------

Tanh	$f(x) = \frac{1 - e^{-ax}}{1 + e^{-ax}}$	(negro)
------	--	---------

Gauss	$f(x) = e^{-ax^2}$	(azul)
-------	--------------------	--------

Función de transferencia Gaussiana

Los centros y anchura de estas funciones pueden ser adaptados, lo cual las hace mas adaptativas que las funciones sigmoidales. Mapeos que suelen requerir dos niveles ocultos, utilizando neuronas con funciones de transferencia sigmoidales; algunas veces se pueden realizar con un solo nivel en redes con neuronas de función gaussiana.

1.2.3.5 Estimación de los pesos de activación por aproximaciones lineales

Aquí se explicaran la estimación de los pesos de activación Wh . El acercamiento de la estimación se basa en asumir que una función no lineal puede ser aproximada por una secuencia de funciones lineales basada en estimaciones cuadráticas lineales de los parámetros de las partes lineales. Los parámetros en estas ecuaciones entonces podrán ser usados como los elementos de Wh . Así cada neurona se le asigna a un subconjunto específico del conjunto de datos completo y la red se construye asignando secuencialmente una neurona a cada uno de los subconjuntos. El número de neuronas es el mismo, como el número de los subconjuntos bajo consideración. La aproximación puede ser aplicada a ambos problemas SISO y MISO.

Suponga una red neural de alimentación hacia delante que debe cumplir una cierta función estática $y=f(x)$, $X \in \mathbb{R}^{N_i}$ y $y \in \mathbb{R}$. Luego un conjunto de datos Φ puede estar creado con pares de entrada y salida medidos o la matriz indicada:

$$\Phi \begin{bmatrix} x_1(1) & \cdots & x_{n_i}(1) & y(1) \\ \vdots & \vdots & \vdots & \vdots \\ x_1(k) & \cdots & x_{n_i}(k) & y(k) \\ \vdots & \vdots & \vdots & \vdots \\ x_1(N_e) & \cdots & x_{n_i}(N_e) & y(N_e) \end{bmatrix} = [XY]$$

El índice k denota el evento K th. Particionando la matriz Φ en N_h subconjuntos es posible construir un modelo lineal para cada subconjunto

$\Phi_j = [X_j, Y_j]$, con $\Phi_j \subset \Phi$ y $j \in \{1, \dots, 2, \dots, N_h\}$. El modelo lineal entonces estará dado

$$\text{por: } \bar{Y}_j = X^j W_{j+}^h b_j^h$$

Los pesos y $W_j^h (W_j^h \in R^{N_h \times 1})$ puede ser calculada con OLS (cuadráticos ordinarios):

$$\begin{bmatrix} W_j^h \\ b_j^h \end{bmatrix} = (\tilde{X}^{jT} \tilde{X}^j)^{-1} \tilde{X}^j \tilde{Y}^j$$

En esta ecuación \tilde{X}^j es la matriz X_j suministrada con una columna extra llenada con unos para los bias b_j^h . \tilde{Y}^j es una matriz escalar Y^j con las siguientes propiedades:

$$\max_k (|\tilde{Y}^j|) < C$$

con el valor de c donde $|df/dz| = \varepsilon$ (f es la función de activación y ε es un escalar real positivo pequeño).

Cada función lineal describe el mapeo entre la entrada y la salida para un subconjunto particular n . Finalmente después de calcular N_h modelos lineales, la matriz W^h se construye como:

$$W^h = \begin{bmatrix} W_1^h & W_2^h & \dots & W_j^h & \dots & W_{N_h}^h \\ b_1^h & b_2^h & \dots & b_j^h & \dots & b_{N_h}^h \end{bmatrix} \in \mathfrak{R}^{(N_h+1) \times N_h}$$

1.2.3.6 Particionamiento del conjunto de datos

Aquí describimos el particionamiento y la escalización de la matriz Φ . Se exponen dos métodos de particionamiento de esta matriz: la aproximación de particionamiento uniforme y la aproximación de Fuzzy Clustering.

El primer método no iterativo de particionamiento del conjunto de datos Φ simplemente es dividir el conjunto de datos en N_h intervalos:

$$\begin{array}{l}
\text{1er subconjunto} \\
\\
\\
\\
\\
\\
\\
\\
\\
\\
\text{N}_h\text{-ésimo subconjunto}
\end{array}
\left\{ \begin{array}{cccc}
x(1) & \cdots & x_{Ni} & y(1) \\
\vdots & \vdots & \vdots & \vdots \\
x(1+N_j) & \cdots & x_{Ni}(1+N_j) & y(1+N_j) \\
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
x(N_e) & \cdots & X_{Ni}(N_e) & y(N_e)
\end{array} \right.$$

Para simplificar N_j es igual a N_e/N_h , redondeado hacia el entero más cercano. Si N_j no es un entero, o cuando haya relativamente pocos pares de datos disponibles, entonces las particiones pueden superponerse. Esta aproximación se espera que sea razonable si la entrada y la salida tienen un cambio suave sobre los eventos, como se da en el caso de muchos problemas con no linealidades suaves, por ejemplo en diferenciales.

Si la entrada y la salida no tienen un cambio suave y uniforme en el conjunto de datos Φ o cuando existen saltos en el conjunto de datos, se pueden utilizar algoritmos de Fuzzy Clustering para encontrar particiones útiles del conjunto de datos **[Babuska y Verbruggen, 1994]**. Con respecto a la partición de los conjuntos de datos, el concepto de membresía por grados (membresía Fuzzy) puede ser utilizado para representar el grado al cual un objeto dado representado por un vector de características es similar a cierto prototipo, por ejemplo, punto, línea hiper plano, elipsoide. El grado de similitud entre un punto y el prototipo se puede calcular utilizando una medida de distancia escogida adecuadamente.

Basados en esta similitud los datos pueden ser particionados de tal forma que los datos dentro de un cluster son tan similares como sea posible o tan cercanos, y los datos de diferentes cluster sean lo más distintos posibles. Este resultado en un vector con prototipo de cluster y una matriz con grado de membresía de los puntos de datos con respecto a los diferentes cluster. En la figura 1.23 que se muestra a continuación se da un ejemplo de un conjunto de datos que debe ser particionado en dos partes, con los prototipos C1 y C2, que para este caso son simplemente puntos.

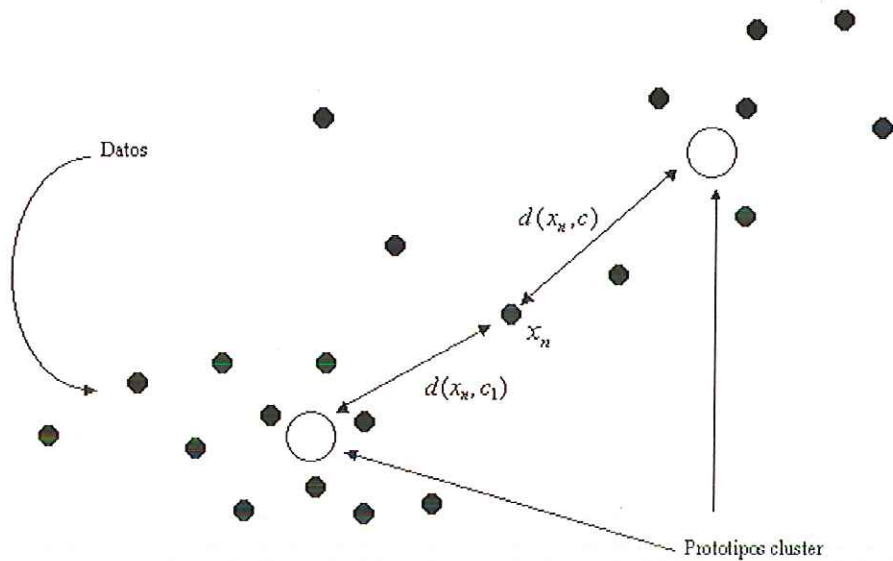


Figura 1.23 Principio de Clustering

El clustering resultante en una matriz de partición Fuzzy puede ser expresada como:

$$U = \begin{pmatrix} \mu_{11} & \mu_{12} \\ \vdots & \vdots \\ \mu_{k1} & \mu_{k2} \\ \vdots & \vdots \\ \mu_{Ne1} & \mu_{Ne2} \end{pmatrix}$$

Esta matriz le asigna a cada punto de datos X_k un valor μ_{kj} expresando el valor de la membresía de ese punto al cluster prototipo C_j .

Existen varias técnicas de clustering o particionamiento pero aquí usaremos la técnica de clustering Fuzzy de Gustafson-Kessel. El prototipo usado en el algoritmo G-K es un punto, y la norma que induce la medida de distancia es una variable la cual también debe ser optimizada. Por lo tanto se tiene la habilidad para adaptar la forma del cluster. Este detecta la estructura del cluster en vez de imponer una estructura predefinida, así es posible detectar una mezcla de forma de cluster tal como la esférica, elipsoide, lineal o plana [Dave 1992].

Para proceder permita que $C = \{C_j | j=1,2,\dots,N_h\}$ sea el prototipo de cluster N_h ésimo y los R_{Ni+1} , $C_j = (C_{j1}, C_{j2}, \dots, C_{j(N_i+1)})$. La matriz U es llamada matriz de partición Fuzzy, y sus elementos $[\mu_{kj}]$ representa un grado de membresía del punto de datos ϕ_k dentro del cluster C_j . Posteriormente, $U = [\mu_{kj}], K=1,2,\dots,N_e$ y $J=1,2,\dots,N_h$ con $\mu_{kj} \in [0,1]$. Φ_k no debe ser confundida con ϕ_j , el cual es un subconjunto de la matriz de datos Φ . El algoritmo G-K busca una partición óptima Fuzzy U y un vector prototipo C minimizando la función objetiva siguiente:

$$J(\Phi, C, U) = \sum_{j=1}^{N_h} \sum_{k=1}^{N_e} \mu_{kj}^m D_2(\phi_k, C_j)$$

donde $m > 1$ es un parámetro que controla el fuzziness del cluster, usualmente $m=2$. La función $D(\phi_k, C_j)$ es la distancia del punto de datos Φ_k al prototipo de cluster C_j , inducido por una matriz definida positiva M_j :

$$D_2(\Phi_k, C_j) = (\Phi_k - C_j) M_j (\Phi_k - C_j)^T$$

donde M_j se calcula como el inverso normalizado de la matriz covariante del cluster F_j :

$$M_j = \det(F_j)^{-1} / (N_j - 1) (F_j - 1)$$

La matriz covariante del cluster F_j :

$$F_j = \frac{\sum_{k=1}^{N_e} \mu_{kj}^m (\Phi_k - C_j)(\Phi_k - C_j)}{\sum_{k=1}^{N_e} \mu_{kj}^m}$$

La matriz con partición Fuzzy estará dada entonces por:

$$\mu_{kj} = \left\{ \sum_{j=1}^{N_h} \frac{D_{jk}}{(D_{kl})^{2/(m-1)}} \right\}^{-1}, 1 \leq j \leq N_h, 1 \leq k \leq N_e$$

y el prototipo de cluster se calcula con:

$$C_j = \frac{\sum_{k=1}^{N_c} \mu_{kj}^m \Phi_k}{\sum_{k=1}^{N_c} \mu_{kj}^m}, 1 \leq j \leq N_h$$

En [Babuska y Verbrugen (1995)] se da una descripción del algoritmo. Las particiones fuzzy obtenidas pueden ser utilizadas para estimar los pesos de activación, similar al caso de un particionamiento obtenido del conjunto de datos. Claro está que el algoritmo G-K necesitará más tiempo de calculo, pero si el conjunto de datos no está uniformemente distribuido y si los datos no cambian gradualmente y suavemente, este método es más útil para encontrar las particiones.

1.2.3.7 Escalización Interna

Antes de calcular los pesos de activación y los bias, es necesario escalizar Y_j , el cual es la salida particionada, o Y con el fin de evitar una matriz mal acondicionada ($V_b^r V_b$). Claro está que esto puede ser realizado escalizando el conjunto de datos de entrada / salida directamente, sin embargo esto no es necesario. Existen diversas maneras para evitar una matriz mal acondicionada ($V_b^r V_b$), una breve descripción de algunos métodos de escalización interna se estarán dando. El primero y más simple método es escalizar la matriz Y de acuerdo a:

$$\tilde{Y} = \frac{\alpha}{y_{\max} - y_{\min}} (Y - y_{\min}) + \beta$$

Con Y_{\max} y Y_{\min} siendo los máximos y los mínimos valores de Y , respectivamente. Y_{\min} es un vector del mismo tamaño que Y , y llenado con el valor Y_{\min} . Las variables α y β son parámetros para cambiar el resultado de escalización. Utilizando la ecuación anterior, α puede ser vista como la ganancia y β como el offset desde cero; típicamente ellos pueden ser escogidos como 1 y - 0.5 respectivamente. Y entonces

será escalizado (\tilde{Y}) entre -0.5 y $0.5+1$. Note sin embargo que esta escogencia depende de la aplicación en que vaya a ser modelada.

Otra aproximación está en escalizar la información por neurona:

$$\tilde{Y}^j = \frac{\alpha_j}{y_{\max} - y_{\min}} (Y^j - Y_{\min}^j) + \beta_j$$

Donde α_j y β_j pueden ser escogidas diferentes para cada neurona, y así para cada partición. Esta aproximación puede ser utilizada para realizar una sintonización fina de la capacidad de modelamiento. Ambas aproximaciones de escalización son sensibles a falsos ruidos. Si existe datos falsos, entonces la escalización basada en la variación y el significado de la señal de salida puede superponerse a este problema (estandarización).

El último método que describiremos actualiza α_j y β_j iterativamente. El procedimiento primero toma un α_j y β_j inicial, y luego calcula las columnas de W_h y W_0 de acuerdo a las ecuaciones mostradas. Luego, α_j y β_j son ajustadas en la dirección del gradiente negativo, de tal forma que la ecuación de la función de criterio completa disminuye, y W_h y W_0 se estiman nuevamente hasta que el criterio se cambia por menos del valor preespecifico. El método utilizado es el Quasi- Gauss-Newton.

Note que este método de escalización tiene una influencia directa en los pesos de activación, y por lo tanto no es necesario para el usuario escalizar el conjunto de datos de entrada y salida. En efecto, el usuario no ve los resultados de la escalización interna y así tampoco tiene que preocuparse sobre los parámetros de escalización.

1.2.3.8 Estimación de los Pesos de Salida

Como se ha mencionado anteriormente el propósito es de dividir el problema de entrenamiento en dos subproblemas que pueden cada uno ser resueltos óptimamente. El cálculo de W_0 será analizado a continuación. Los pesos W_h ya son conocidos, y por lo tanto V_b también es conocido. Entonces, suponga que la salida verdadera puede estar modelada por: $Y = V_b W_0 + e$

lo cual resulta en un problema de estimación de parámetros que contiene parámetros lineales. El vector e denota el error de modelamiento. Este error se asume de ser independiente del vector de entrada V_b , y tiene una varianza homogénea σ^2 y significa $\mu = 0$. El minimizar la suma de errores de modelamiento cuadrático pueden ser vistos como encontrar una proyección ortogonal del vector de error y del vector de entrada, o:

$$V_b^T [Y - V_b W^0] = 0$$

Este resultado es bien conocido en ecuaciones normales, y puede ser escrita de otra forma como:

$$\tilde{W}_0 = F V_b^T Y \quad \text{Con } F = [V_b^T V_b]^{-1}$$

La línea encima de W indica el factor de que la variable relacionada es la estimación de esta variable. La matriz simétrica $V_b^T V_b$ debe ser no singular, de otra manera la matriz $[V_b^T V_b]^{-1}$ no existirá. Esto implica que V_b debe tener un rango $N_h + 1$. Mas aun sobre el teorema de Gau β -Markov, se puede mostrar que, asumiendo una e , esta estimación es el mejor estimador lineal sin bias para W_0 . La varianza del error puede ser estimada únicamente por el siguiente estimador sin bias :

$$S^2 = \sigma^2 = \frac{1}{r} [Y - V_b W^0]^T [Y - V_b W^0]$$

donde r es un entero que expresa los grados de libertad del modelo, o el número de puntos de datos (N_e) menos el número de regresores ($N_h + 1$). Mas formalmente, $r = \dim(Y) - \dim(\langle V_b \rangle)$, donde $\langle V_b \rangle$ es el espacio de envergadura de los vectores de V_b . La varianza de la estimación de los pesos de salida particulares se pueden calcular con: $\text{cov}(W_0) = S^2 F$.

Como se ha mencionado anteriormente, la estimación cuadrática ordinaria, da un estimado de un bias y tiene la propiedad de varianza mínima, por ejemplo, no contiene la

varianza mínima de todos los estimadores lineales no bias. Sin embargo, no existe un límite superior en la varianza del estimador y la presencia de multicolinealidad puede producir largas varianzas, es decir, algunas columnas de regresores son casi linealmente dependientes, lo cual resulta en una matriz casi singular $V_b^T V_b$. Así para preservar la falta de un bias puede resultar una gran varianza. La presencia de multicolinealidad sugiere que la red debe contener menos neuronas de activación. No existe suficiente información en los datos del regresor para garantizar el modelamiento con neuronas N_h la solución es decrementar el número de neuronas de activación, en términos de, por ejemplo, análisis de principio de componentes. Otra solución es el uso de los llamados estimadores Ridge, resultante de la estimación de bias de los parámetros [Myers, 1986].

La estimación bias se utiliza para obtener una reducción substancial en la varianza, con un incremento acompañado en la estabilidad de los coeficientes de regresión. La base para tal problema de regresión por bias es la siguiente función de criterio, el cual también es llamado el problema de regresión cuadrática regularizada:

$$J(W_h, W_0) = \|e\|^2 + \lambda \|W_0\|^2$$

Así, no solo es pesado el modelo de error, pero también la magnitud de los pesos se obtienen. El estimador de regresión ridge resultante es [Myers, 1986]:

$$\hat{W}_{rdls}^o = [V_b^T V_b + \lambda I]^{-1} V_b^T Y$$

(el sufijo rdls denota: ridge least squares). Otra solución para el problema de la multicolinealidad es el uso del análisis de principio de componentes.

En vez de usar un criterio ordinario cuadrático, es posible utilizar un criterio que no solo tenga en cuenta los errores en las salidas, pero también para errores en las entradas. Un criterio útil para este propósito es el criterio total cuadrático [Golub y Van Loan, 1980; Van Huffel y Vandewalle, 1991]:

$$\hat{W}_{tls}^o = [V_b^T V_b - \tilde{\sigma}_{N_h + N_o + 1}^2 I]^{-1} V_b^T Y$$

El sufijo tls significa "total least squares". en esta ecuación $\sigma_{2N_h+N_0+1}$ es el menor valor singular de la matriz $[V_b; Y]$. Los valores singulares pueden ser calculados utilizando la descomposición de valor singular dada por:

$$[V_b; Y] = \Sigma \Pi^T$$

donde la matriz Σ contiene los vectores singulares de la izquierda y Π contiene los valores singulares de la derecha. La matriz Σ es una matriz diagonal donde los elementos diagonales σ_i son los valores singulares ordenados de la matriz $[V_b; Y]$. La relación entre el OLS y el TLS estimado está dado:

$$\hat{W}_{0tls} = [I - \sigma_{2N_h+N_0+1}^{-2} (V_b^T V_b)^{-1}]^{-1} \hat{W}_0$$

también se puede demostrar que se cumple lo siguiente:

$$\|\hat{W}_{0tls}\|_F \geq \|\hat{W}_0\|_F$$

donde $\|\cdot\|_F$ denota la norma Frobenius. De esto se puede concluir que para aplicaciones de redes neuronales el estimador TLS no se espera que sea muy importante. mas aun el estimador TLS es un estimador bias. Debido al factor de que un término extra de bias es adicionado a las neuronas, un problema mezclado OLS-TLS aún sigue remanente.

La red RAWN ha sido probada en varios problemas de modelamiento prácticos. Aunque la aproximación es más que todo una forma heurística de entrenar redes neuronales estas pruebas demuestran que se pueden obtener resultados razonables de una manera muy rápida. Estas pruebas también han demostrado en algunos casos que la aproximación TLS no se desempeñó con suficiencia. La aproximación de Ridge Least Squares dio buenos resultados pero la sintonización del λ no fue muy correcta. Para todos los problemas cuadráticos lineales el método OLS es muy utilizado para resolver este tipo de problemas.

00-02-01

En resumen, el algoritmo RAWN apunta a la minimización del tiempo de entrenamiento de redes neuronales de realimentación con una sola capa oculta. Esencialmente, el algoritmo está dividido en 2 partes: (1) estimación de los pesos de activación; (2) estimación de los pesos de salida. Ambos problemas de estimación pueden ser resueltos óptimamente aplicando algoritmos existentes, desarrollados para problemas lineales. Aunque no se haya probado, el principal objetivo es que la combinación de las dos soluciones resultará en una solución subóptima del problema de entrenamiento total. Haciendo esto, un enorme decremento en los tiempos resultantes de cómputo, con capacidades de modelamiento comparables a aquellas de las redes entrenada con métodos de optimización iterativa de tipo gradiente. Solo cuando 2 opciones especiales en la aproximación son utilizadas, es decir, la partición de datos por Fuzzy Clustering, descrita anteriormente, y la escalización iterativa, más de una iteración debe ser realizada, resultando en un proceso de entrenamiento más lento, todas las otras opciones son libres de iteración. Otra ventaja del algoritmo es que no es necesario escalizar los conjuntos de datos de entrada-salida. Las técnicas basadas en gradientes necesitan escalización de el conjunto de datos entrada-salida (esto se aplica para Backpropagation, por ejemplo), o están fuertemente influenciadas por la escalización (como en el caso del [entrenamiento con Levenberg Marquardt](#)).

Intuitivamente, sin embargo, se espera que existan situaciones donde la red RAWN necesitará más neuronas que una red entrenada con algoritmos iterativos. Esta idea también está soportada por las pruebas que han sido realizadas. Una solución para este tipo de problema puede ser el de la reoptimización de los pesos. Un candidato potencial para esta reoptimización es la combinación frontal RAWN y de las técnicas Levenberg-Marquardt. RAWN entonces será utilizada como un suministrador de las matrices de peso iniciales, que serán utilizadas en un esquema de entrenamiento iterativo posteriormente.

1.3 FUNDAMENTOS MATEMATICOS

Aplicar la técnica de análisis de datos multivariado ha sido aceptada y usada en todos los campos de investigación científica durante los últimos años. Hay muchas razones para que se dé este hecho, dos de las más importantes son:

Las relaciones simultaneas entre dos o más variables que en muchas investigaciones científicas son necesarias, tal como en nuestro caso se busca la relación de las variables de entrada con la variable de salida del proceso.

El advenimiento de la alta velocidad de los computadores con grandes capacidades de almacenamiento y la facilidad de desarrollo para implementar paquetes de software con análisis multivariado como el ofrecido por la herramienta Matlab.

El análisis multivariado puede ser simplemente definido como la aplicación de métodos que trata razonablemente cantidades de muestras grandes de determinadas variables hechas sobre cada muestra en uno o más instantes simultáneamente. En esta corta definición el punto importante es que el análisis multivariado tiene que ver con relaciones simultáneas entre variables. En otras palabras, son técnicas multivariadas diferenciadas para analizar univariable y bivARIABLES en la que hay directa relación para el análisis de la media y la varianza de una simple variable, o para la relación entre parejas de dos variables, el análisis de covarianza o correlación el cual refleja la extensión de la relación entre tres o más variables.

Por otro lado, encontramos que para esta aplicación el proceso tiene características estocásticas debido a que uno o más aspectos son desconocidos con anterioridad. De esta forma un proceso estocástico se puede mirar como una familia de funciones en el tiempo, cada una de las cuales se distingue por su comportamiento.

Un proceso estocástico se define sencillamente como una colección indexada de variables aleatorias $\{V_i\}$, donde el subíndice toma valores de un conjunto dado. Con frecuencia este conjunto se toma como el conjunto de enteros no negativos y V_i representa una característica de interés medible en el tiempo. Por ejemplo, el proceso estocástico, V_1, V_2, V_3, \dots , Puede representar la colección de variables del un proceso como los son, niveles, flujos, temperaturas y presiones o la calidad resultante en la producción de la planta.

Vamos a simbolizar tanto el proceso estocástico como su comportamiento concreto como $v(t)$. Si en el resultado de la observación se obtiene algún conjunto de comportamientos del proceso estocástico, entonces el proceso dado puede ser descrito por una serie de características no estocásticas como lo son la esperanza matemática, la dispersión y la

función autocorrelacional, la definición de ellas está basada en la medida del conjunto de su comportamiento.

En general, las características observadas en un proceso estocástico son funciones del tiempo. Sin embargo existe una clase importante de procesos estocásticos que se llaman estacionarios, para los cuales estas características son constantes. En muchos casos hay características de procesos estocásticos estacionarios, obtenidos a través de la medición del conjunto de realizaciones que coinciden con las características logradas como resultado de la medida de un comportamiento suficientemente continuo en el tiempo. Tales procesos se llaman "ERGODICOS".

Las principales características de los procesos estacionarios ergódicos se definen en los siguientes numerales.

1.3.1 Distribución

Numerosos fenómenos tales como las puntuaciones de los textos en un examen, los errores de las medidas y muchas otras variaciones se consideran como distribución normal, naturalmente, esto no significa que una distribución de frecuencias relativas realmente observada siga exactamente la distribución normal. La distribución normal es simplemente una teoría que sirve para explicar algunas variables aleatorias, la relación entre los intervalos de sus valores, y su correspondiente probabilidad

Es habitual, en estudios especializados de redes neuronales, utilizar distribuciones especiales de puntos según una determinada distribución aleatoria para observar la evolución de los valores de los pesos de la red ante distribuciones diferentes, en este trabajo se ha utilizado la distribución para verificar el comportamiento de cada variable aleatoriamente en el tiempo.

1.3.2 Regresión Múltiple

Esta es quizás comúnmente el método más conocido y usado método multivariado. La regresión múltiple se preocupa por el estudio de la dependencia de una variable, la variable dependiente, sobre un conjunto de otras variables, las variables predictoras, con

miras a estimar ó predecir el valor del error de la variable dependiente con base en los valores conocidos de las variables predictoras.

"Análisis de regresión" es quizá es la más extensa técnica aplicada al análisis de conjunto de datos usada para estimar relaciones entre variables. Específicamente, es usada para investigar las relaciones entre una variable dependiente Y y una o más variables independientes (predictoras). Para entender mejor primero debemos hacer un pequeño repaso de lo que es análisis de regresión.

Análisis de regresión es primeramente concebida como estimación y/o predicción de la población o valor de lado variable dependiente y sobre las bases de los valores conocidos de una o más variables explicativas X_i . el modelo popular de regresión dice que la condición de medida denotada por $E(Y | X_i)$, son funciones lineales de X_i y puede ser escrita

$$E(Y | X_i) = \beta_0 + \beta_1 X_i$$

Donde β_0 y β_1 son desconocidas pero con parámetros conocidos sobre coeficientes de regresión. Actualmente, referimos a β_0 como el intercepto el cual es dado $E(Y | X_i) = 0$ y β_1 sobre el cálculo de los coeficientes que permanecen cambiando en (Y), las unidades respectivamente.

El conocido modelo de regresión es irreal, ya que implica que los datos X y Y han sido coleccionados sobre todo los individuales, o familiares que definen la población de interés. Aún que esto es una situación altamente diferente, ello hace, sin embargo, permitir el desarrollo de modelo de regresión en muchas formas simples. Porque los datos de la población de entrada son asumidos como válido, y el condicional de probabilidad de Y es dado por X denotado por $P(Y | X_j)$ puede ser calculado tomando

$$P(Y = Y_i | X = X_j) = \frac{P(Y = Y_i, X = X_j)}{P(X = X_j)}$$

Siendo $P(X = X_j) > 0$ la condición de medida o expresión condicional $E(Y | X_j)$, es el valor medido de la probabilidad condicional de Y dando X. Ello puede ser calculado por

$$\sum Y_i P(Y = Y_i | X = X_j)$$

Considere la Figura 1.24, con un conjunto hipotético de puntos de datos y mostrando la medida condicional de la variable dependiente Y para los valores explicados de la variable X. la línea que conecta los valores condicionales es la línea de regresión. La figura muestra que para cada X_i hay una población de Y_i valores asumiendo aquí la distribución normal, y una correspondencia condicional de la medida. La línea de regresión pasa entre estas medidas condicionales.

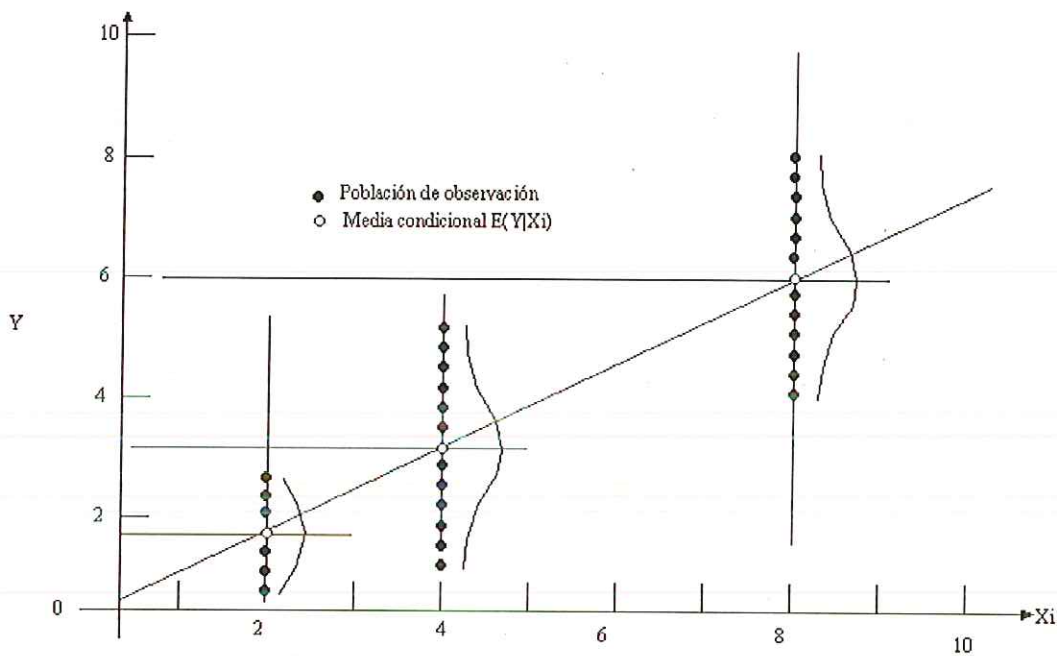


Figura 1.24 Línea de regresión de población hipotética con términos de error distribuidos normalmente.

Estas especificaciones del modelo de regresión lineal mostrada en la ecuación, muestra que ambos son lineales; los parámetros y las variables, como opuesto, es decir, el modelo $E(Y | X_i) = \beta_0 + \beta_1 X_i^2$, la cual es lineal en los parámetros pero no lineal en la variables X. Linealmente en los parámetros es relevante la condición impuesta para desarrollo de la teoría de regresión que será presentada. Así el término lineal en regresión es usada, si se refiere al modelo de regresión que es lineal en los parámetros, pero no necesariamente en las variables.

1.3.2.1 Regresión, Correlación, y la Casualidad de Casos (Procesos Estocásticos)

El análisis de regresión y el de correlación están estrechamente relacionadas, pero son conceptualmente muy diferentes. El análisis de regresión intenta estimar el valor del error de la variable dependiente sobre la base de valores conocidos de una o más variables predictoras. El análisis de correlación intenta medir el poder o grado lineal de asociación entre dos variables. En el análisis de regresión se hace una distinción entre la variable dependiente y las variables independientes; en otras palabras, hay asimetría de la forma de variables independientes y dependientes que son tratadas. Además la variable dependiente se asume que tiene una distribución de probabilidad, mientras que las variables independientes suponen tener valores fijos, es no estocástico (aunque anteriormente actuáramos como si el X fuera aleatorio) si se habla sobre la distribución de probabilidad del X constante, en el análisis de correlación no se hace ninguna distinción entre las variables X y Y; esto es, se hace cualquier par de variables simétrica, que promedie ambas variables aun cuando son vistas con inicio aleatorio y distribución de probabilidad.

La tentación de muchos campos de investigación es atribuir casualidad a los resultados de un análisis estático usando métodos de regresión o correlación.

1.3.2.2 El Caso de los X Fijos

Introduciendo análisis de regresión, se asume que la variable independiente es no estocástica; eso es, sus valores son de sentido fijos en una repetida prueba; en otras palabras los datos generados por un procedimiento de prueba en el cual para cada valor fijo de X_i un valor de Y será seleccionado aleatoriamente para la población. Dependiendo del área de investigación la natural suposición no estocástica de la variable independiente puede o no puede ser razonable. Por ejemplo, una escena de laboratorio donde, decimos, que del investigador está intentando predecir el efecto tóxico promedio que es presentado claramente en diferentes fracciones del organismo inducido por la succión de una cierta droga, es razonable asumir que el investigador tiene control sobre los niveles de dosificación administrado y que ellos pueden fijar en repetidas pruebas. En contraposición al estudio de investigación diseñada una escena de la cual el investigador tiene poco control sobre los valores que pueden tomar las variables, y así la variable independiente es

intrínsecamente estocástica. Veamos en una próxima sesión que sobre una cierta suposición, variables independientes estocásticas no causan dificultades adicionales con respecto al procedimiento de estimación OLS o sus propiedades estadísticas.

1.3.2.3 La Aproximación Ordinaria de Mínimos Cuadrados

A continuación se discute la aproximación ordinaria de mínimos cuadrados (OLS) para el análisis de regresión. Para la mayor parte, la discusión es en términos de p variables o múltiples modelos de regresión; sin embargo, en ciertas instancias se confía sobre las dos variables del modelo de regresión en orden para simplificar la exposición. En todo los casos donde las simples dos variables del modelo de regresión es usada, la generalización al modelo de regresión múltiple es aplicable.

1.3.2.4 La Variable P del Modelo de Regresión

La variable lineal p del modelo de regresión múltiple involucra la variable dependiente Y y la variable independiente p enumerada como $X_2, X_3, X_4, \dots, X_p$, que puede ser escrita como

$$\text{Ecuación 1.18 } Y_i = \beta_1 + \beta_2 X_{2i} + \beta_3 X_{3i} + \dots + \beta_p X_{pi} + \varepsilon_i, \quad i = 1, 2, 3, \dots, n$$

Donde β_1 denota el de intercepto, $\beta_2, \beta_3, \dots, \beta_n$ son la inclinación parcial de los coeficientes de regresión, y ε_i es el término residual asociado con la i -ésima observación.

El modelo regresión de todos los valores esperados de Y condicionados a los valores fijos de X_2, X_3, \dots, X_p más el componente de error. Actualmente la ecuación 1.18 corresponde a un sistema de ecuaciones de la forma:

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} = \begin{pmatrix} \beta_1 + \beta_2 X_{21} + \beta_3 X_{31} + \dots + \beta_p X_{p1} + \varepsilon_1 \\ \beta_1 + \beta_2 X_{22} + \beta_3 X_{32} + \dots + \beta_p X_{p2} + \varepsilon_2 \\ \vdots \\ \beta_1 + \beta_2 X_{2n} + \beta_3 X_{3n} + \dots + \beta_p X_{pn} + \varepsilon_n \end{pmatrix}$$

La matriz de anotación de este sistema de ecuaciones puede ser escrita como

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} = \begin{pmatrix} 1 + X_{21} + X_{31} + \dots + X_{p1} \\ 1 + X_{22} + X_{32} + \dots + X_{p2} \\ \vdots \\ 1 + X_{2n} + X_{3n} + \dots + X_{pn} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

$$\underset{(n \times 1)}{Y} = \underset{(n \times p)}{X} \underset{(p \times 1)}{\beta} + \underset{(n \times 1)}{\varepsilon}$$

Donde

$Y = n \times 1$ vector columna de observación sobre las variables dependiente Y

$X = n \times P$ matriz que resulta para n observaciones sobre $P - 1$ variables independientes

X_2, X_3, \dots, X_p , donde la primera columna de unos representa el término de intercepto β_1 ; que es $X_1 = -1$

$\beta = p \times 1$ vector columna de parámetros desconocidos $\beta_1, \beta_2, \dots, \beta_p$ donde β_1 es el término intercepto, y $\beta_2, \beta_3, \dots, \beta_p$ son llamados los coeficientes parciales o pesos de regresión

$\varepsilon = n \times 1$ vector columna de n error residuales ε_i

En el modelo de regresión

$$Y = X\beta + \varepsilon$$

Los $p \times 1$ vectores columna β contiene los $p - 1$ coeficientes parciales de regresión $\beta_2, \beta_3, \dots, \beta_p$, y el término intercepto β_1 . El término parcial es usado por que el j -ésimo coeficiente de regresión β_j , que mide el cambio en el valor del error de Y y por unidad de cambio en X_j , perteneciendo todas de las otras variables constantes independientes. En una tercera variable del modelo regresión, para el ejemplo, β_2 mide los cambios en el valor del error de Y , $E(Y_i | X_2, X_3)$, con respecto de X_2 la influencia pertenece a la constante X_3 debido a la variación concomitante, así como no es calculada doble, en la contribución de la evaluación de X_2 a un cambio en Y nosotros controlamos la influencia de X_3 , e igualmente influencia de la contribución de X_3 a el cambio en Y nosotros controlamos la

influencia de X_2 tendríamos mucho más es decir sobre los coeficientes de regresiones parciales de la siguiente sesión.

1.3.2.5 Estimador de Mínimos Cuadrados (LSE)

En el problema general de mínimos cuadrados, la salida de un modelo lineal y , esta dado por la expresión parametrizada linealmente:

$$\text{Ecuación 1.19 } y = \theta_1 f_1(u) + \theta_2 f_2(u) + \dots + \theta_n f_n(u),$$

donde $u = [u_1, \dots, u_p]^T$ es el modelo del vector de entrada, f_1, \dots, f_n son funciones conocidas de u y $\theta_1, \dots, \theta_n$ son parámetros desconocidos a ser estimados. En estadística, la tarea de ajustar datos usando un modelo lineal es referido como **regresión lineal**. Así la ecuación 1.19 es también llamada **función de regresión** y los θ_i son llamados coeficientes de la regresión.

Para identificar los parámetros desconocidos θ_i , usualmente tenemos que ejecutar experimentos para obtener un **conjunto de datos entrenados** compuestos de pares de datos; ellos representan la pareja de entrada-salida deseados del sistema objetivo a ser modelado. Substituyendo cada pareja de datos en la ecuación 1.19 producirá un conjunto de m ecuaciones lineales:

$$\begin{cases} f_1(u_1)\theta_1 + f_2(u_1) + \dots + f_n(u_1)\theta_n & = & y_1, \\ f_1(u_2)\theta_1 + f_2(u_2) + \dots + f_n(u_2)\theta_n & = & y_2, \\ & \vdots & \vdots \\ f_1(u_m)\theta_1 + f_2(u_m) + \dots + f_n(u_m)\theta_n & = & y_m. \end{cases}$$

Usando notación matricial, podemos describir las anteriores ecuaciones de una forma concisa:

$$\text{Ecuación 1.20 } A\theta = y,$$

donde A es una matriz de $m \times n$ (algunas veces llamada **matriz de diseño**):

$$A = \begin{bmatrix} f_1(u_1) & \cdots & f_n(u_1) \\ \vdots & \vdots & \vdots \\ f_1(u_m) & \cdots & f_n(u_m) \end{bmatrix},$$

θ es vector desconocido de parámetros de $n \times 1$:

$$\theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix},$$

e y es un vector de salida de $m \times 1$:

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}.$$

La i -ésima fila de la matriz unión de datos $[A:y]$, denotado por $[a_i^T : y_i]$, esta relacionado con la i -ésima entrada-salida de la pareja de datos $(u_i; y_i)$ por medio de

$$a_i^T = [f_1(u_i), \dots, f_n(u_i)].$$

Puesto que la mayoría de nuestros cálculos están basados en las matrices A e y , algunas veces nos referimos aproximadamente a $(a_i^T; y_i)$ como el i -ésimo par de datos del conjunto de datos de entrenamiento.

Para identificar el vector desconocido θ de manera única, es necesario que $m \geq n$. Si A es el cuadrada ($m=n$) y no singular, entonces podemos resolver x de la ecuación 1.20 por

$$\theta = A^{-1}y.$$

Sin embargo, usualmente m es mas grande que n , indicando que tenemos mas parejas de datos que parámetros de ajuste. En este caso, una solución exacta para satisfacer todas las m ecuaciones no siempre es posible, puesto que los datos estarán

contaminados por ruido o el modelo puede no ser apropiado para describir el sistema objetivo. De este modo la ecuación 1.20 debe ser modificada incorporando un vector de error e para contar el ruido aleatorio o el modelaje del error, como sigue:

$$A\theta + e = y.$$

Ahora en lugar de encontrar una solución exacta a la ecuación 1.20, deseamos encontrar un $\theta = \hat{\theta}$ que minimice la **sumatoria de errores cuadráticos** definido por

$$\text{Ecuación 1.21} \quad E(\theta) = \sum_{i=1}^m (y_i - \alpha_i^T \theta)^2 = e^T e = (y - A\theta)^T (y - A\theta),$$

donde $e = y - A\theta$ es el vector de error producido por la escogencia de un θ específico. Note que $E(\theta)$ esta en la forma cuadrática y tiene un mínimo único en $\theta = \hat{\theta}$. El siguiente teorema declara una condición necesaria que se satisface por el estimador de mínimos cuadrados $\hat{\theta}$.

Teorema Estimador de Mínimos Cuadrados

El error cuadrático en la ecuación 1.21 es minimizado cuando $\theta = \hat{\theta}$, llamado el **estimador de mínimos cuadrados** (conocido como LSE), que satisface la **ecuación normal**

$$A^T A \hat{\theta} = A^T y.$$

Si $A^T A$ no es singular, $\hat{\theta}$ es única y esta dada por

$$\hat{\theta} = (A^T A)^{-1} A^T y.$$

Una prueba de esto se puede ver en el ejemplo planteado por [Jang J.-S.R.] (pag. 106).

1.3.3 Funciones Autocorrelacionales

$$\text{Ecuación 1.22} \quad R_{v,v}(\tau) = M \{ [v(t) - \bar{v}] \cdot [v(t+\tau) - \bar{v}] \} = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T \{ [v(t) - \bar{v}] \cdot [v(t+\tau) - \bar{v}] \} dt$$

La función correlacional caracteriza el grado de conexión de los valores del proceso que se separan uno del otro en el tiempo τ .

De las ecuaciones 1.22 y 1.28 se deduce directamente que

$$\text{Ecuación 1.23 } R_{v,v}(0) = \sigma_v^2$$

Cuando se analizan sistemas de control automático, que se encuentran bajo la influencia de perturbaciones estocásticas, la densidad espectral juega un papel importante para el proceso estocástico $S_v(\omega)$, que se define como una transformada doble de furrier de la función autocorrelacional:

$$S_v(\omega) = \frac{1}{2T} \int_{-\infty}^{\infty} R_{v,v}(\tau) e^{-j\omega\tau} d\tau = \frac{1}{\pi} \int_0^{\infty} R_{v,v}(\tau) \cos \omega\tau d\tau$$

A su vez la función autocorrelacional puede ser definida como una transformación inversa de furrier de la densidad espectral:

$$R_{v,v}(\tau) = \int_{-\infty}^{\infty} S_v(\omega) e^{-j\omega\tau} d\omega = 2 \int_0^{\infty} S_v(\omega) \cos \omega\tau d\omega$$

Con esta dispersión, como se deduce de la ecuación 1.23, se define la expresión

$$\sigma_v^2 = R_{v,v}(0) = 2 \int_0^{\infty} S_v(\omega) d\omega$$

La representación espectral es muy conveniente cuando se estudian transformaciones de procesos estacionarios estocásticos con sistemas lineales. Es conocido que la densidad espectral $S_y(\omega)$ de una señal de salida de un sistema lineal es igual al producto de la densidad espectral $S_x(\omega)$ de una señal de entrada por el cuadrado del módulo de una función de frecuencia transmisora del sistema:

$$\text{Ecuación 1.24 } S_y(\omega) = S_x(\omega) |W(j\omega)|^2$$

Miremos un proceso estocástico con valor medio igual a cero, que tiene la propiedad tal que las magnitudes $v(t_1)$ y $v(t_2)$ son no correlacionales con valores tan pequeños como se quiera $\tau = t_2 - t_1$, es decir:

$$R_{v,v}(\tau) = 0 \text{ cuando } \tau \neq 0 .$$

La función autocorrelacional de tal proceso, en una forma ideal, se describe de la siguiente manera:

$$\text{Ecuación 1.25 } R_{v,v}(\tau) = V\delta(\tau), \quad V \geq 0$$

donde $\delta(\tau)$ - delta de la función; V - intensidad del proceso.

Tal proceso lleva el nombre de "ruido blanco". La densidad espectral del "ruido blanco" $S_v(\omega)$ puede ser definida con base en la función autocorrelacional (ecuación 1.25):

$$S_v(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} V\delta(\tau)e^{-j\omega\tau} d\tau = \frac{V}{2\pi} = \text{const}$$

En la práctica, la automatización de proyectos industriales, muestra que la función autocorrelacional de muchos procesos estocásticos reales se aproximan mucho a la expresión:

$$\text{Ecuación 1.26 } R_{d,d}(\tau) = \frac{VT_d}{2} e^{-\frac{|\tau|}{T_d}}$$

llamada correlación exponencial.

De esta forma, para la modelación de excitaciones de correlación exponencial puede ser utilizado un sistema lineal como:

$$\text{Ecuación 1.27 } \dot{d} = A_d d + v(t)$$

excitado por el ruido blanco.

Para los procesos estocásticos vectoriales, supongamos que $v_1(t), v_2(t), \dots, v_n(t)$ - n son procesos estocásticos escalares, posiblemente dependientes, entonces

$$\eta(t) = \text{col}[v_1(t), v_2(t), \dots, v_n(t)]$$

es un proceso estocástico vectorial.

Mirando el proceso estocástico vectorial $\eta(t)$. Entonces

$$\bar{\eta}(t) = M\{\eta(t)\} = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T \eta(t) dt - \text{el vector de las esperanzas matemáticas del proceso;}$$

$$R_{\eta, \eta}(t_1, t_2) = M\{[\eta(t_1) - \bar{\eta}(t_1)][\eta(t_2) - \bar{\eta}(t_2)]^T\} - \text{la matriz de covarianza autocorrelacional}$$

$$R_{\eta, \eta}(t, t) = Q_{\eta}(t) - \text{matriz de dispersión.}$$

Para los procesos estocásticos vectoriales las características vistas son constantes.

En conclusión daremos la definición de un proceso normal o de Gauss.

La probabilidad de que un proceso estocástico tome un valor en algún momento fijo de tiempo, que se encuentre en el intervalo $(v; v + dv)$, depende del valor v del momento de tiempo elegido y proporcional a dv , es decir que ella es igual a:

$$p_1(v, t_1) dv.$$

la probabilidad de que e proceso estocástico en el momento de tiempo t_1 , tomara el valor encontrado en el intervalo $(v_1; v_1 + dv_1)$ y en el momento de tiempo t_2 el valor encontrado en el intervalo $(v_2; v_2 + dv_2)$ depende de los valores v_1 y v_2 de los momentos t_1 y t_2 y es proporcional a dv_1, dv_2 , es decir, la probabilidad es igual a:

$$p_2(v_1, t_1; v_2, t_2) dv_1 dv_2.$$

Las funciones $p_1(v, t_1)$ y $p_2(v_1, t_1; v_2, t_2)$ representan correspondientemente la densidad única y doble de la distribución de un proceso estocástico. Análogamente, se pueden definir las densidades conjuntas de una distribución de ordenes superiores.

El proceso estocástico $v(t)$ se llama normal (de Gauss), si la densidad conjunta de una distribución para cualquier conjunto final de magnitudes $v_k = v(t_k), k = 1, n$ es normal.

Notemos que muchos procesos reales estocásticos tienen densidad de distribución cercanas a cero.

1.3.4 Dispersión

Con la Distribución se puede decir y localizar el centro de la misma, pero no se puede decir como se reparten o dispersan los datos a uno y otro lado del centro, esta ultima característica de una distribución se puede llamar Dispersión o Variación. Si la dispersión es poca, indica gran uniformidad de las partículas en la distribución. Por el contrario, una gran dispersión indica poca uniformidad. La ausencia de dispersión es señal de uniformidad perfecta, lo cual quiere decir que todos los datos de la distribución son idénticos.

Para este trabajo, la dispersión se calcula mediante la siguiente formula, la cual permite tener buenos resultados.

$$\text{Ecuación 1.28 } \sigma_v^2 = M \{ [v(t) - \bar{\gamma}]^2 \} = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T [v(t) - \bar{\gamma}]^2 dt$$

La dispersión caracteriza la distribución medida de los valores de un proceso estocástico con relación al promedio.

1.3.5 Medición de la predicción del error

Suponga que tenemos una red neuronal entrenada para realizar predicción en una serie del tiempo. Tenemos una serie de pruebas con la cual evaluar el desempeño de la red entrenada. Usando grupos de puntos conocidos de la serie en prueba, hacemos predicciones. Para cada predicción, sabemos el valor de la serie predecida por la red y el verdadero valor de la serie. El problema ahora es expresar el desempeño de una red de una forma útil.

La mayoría de los algoritmos de entrenamiento de redes neuronales trabajan minimizando el error cuadrático medio de la salida. Naturalmente, esto se hace durante el entrenamiento, en términos de los datos escalados que la red ve realmente. Para nuestro propósito, asumimos que la salida de la red no ha sido escalada, de tal forma que estamos trabajando con datos del problema real.

El error cuadrático medio es calculado encontrando la diferencia entre el objetivo deseado de la salida y la salida alcanzada, elevándolo al cuadrado y sumándolo a través de todas las muestras. La suma es dividida por el número de intentos para dar un valor medio. Si t_i es el valor objetivo (real) de la predicción para el intento de presentación i y o_i es el valor obtenido por la red, el error cuadrático medio para n ensayos de presentación es calculado como:

$$MSE = \frac{1}{n} \sum_{i=0}^{n-1} (t_i - o_i)^2$$

Por varias razones técnicas, es usualmente la medición de error más fácil de minimizar. Por lo tanto, a pesar del hecho de que el error cuadrático medio puede no siempre ser el más significativo, es importante entenderlo.

La operación de elevar al cuadrado en la fórmula anterior, aunque destaca todos los géneros de elegancia matemática, se hace de difícil para los humanos por la interpretación directa del error cuadrático medio. Tomando su raíz cuadrada, obtenemos el Error RMS (error de raíz cuadrada media). Esta medición de error es extremadamente popular. Los ingenieros eléctricos y de comunicaciones están familiarizados con la utilidad de expresar niveles de señal en valores RMSE. Es fácilmente interpretable porque es lineal. Duplicar todos los errores individuales, duplicaría el error RMS. De la misma forma, duplicando la operación, cuadruplica el error cuadrático medio, causando dificultad para comparar resultados.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (t_i - o_i)^2}$$

El error RMS es un número absoluto en el que no está directamente relacionado a la magnitud de la predicción de la serie. Un error RMS de 5 es muy pequeño si la serie tiene

valores típicos en el orden de los millones, pero es grande si la serie oscila alrededor de 10. Su escala esta atada a la unidad de medición de los datos. Así, si se utilizan pulgadas como unidad y luego se enfrenta el mismo problema utilizando milímetros, la comparación de los errores se complica por el hecho de que debemos compensar las diferentes unidades. Para evitar esto, debemos de alguna forma dividir el error por el valor de la serie predecida para obtener un número adimensional. Existen dos formas comunes de hacer esto.

Una forma es expresar el error como una cantidad relativa adimensional, es calcular el valor RMS del termino de la serie a ser predecida y usar su razón como medición del error:

$$RELMSERR = \sqrt{\frac{\sum_{i=0}^{n-1} (t_i - o_i)^2}{\sum_{i=0}^{n-1} t_i^2}}$$

La formula de arriba esta influenciada por una constante de compensación de cero. En otras palabras, sumando una constante a la serie, reducirá el error relativo debido a un incremento en el denominador. Algunas veces, la constante de compensación (offset) es significativa y esto es lo que deseamos. Si no, la media de la serie debe ser restada de cada punto de tal forma que trabajemos con la varianza verdadera.

$$RELMSERR = \sqrt{\frac{\sum_{i=0}^{n-1} (t_i - o_i)^2}{\sum_{i=0}^{n-1} (t_i - \bar{t})^2}}$$

Cual de estas formulas usar es un problema dependiente. El factor determinante es si la constante de compensación de la serie es o no considerada significativa. En la mayoría de los problemas prácticos, no lo es, así que es la ultima ecuación la mas apropiada.

Otra forma menos común de expresar el error como un número relativo adimensional, es dividiendo cada valor de error por el valor objetivo antes de sumarlo. En este caso, se usa el valor absoluto en lugar de elevar al cuadrado como usualmente se hace. Cada término en la suma es la fracción del valor verdadero de la serie representado por el error. La media de estas fracciones es usualmente llamada el error relativo medio (MRE).

$$MRE = \frac{1}{n} \sum_{i=0}^{n-1} \left| \frac{t_i - o_i}{t_i} \right|$$

Los programas para calcular el error de arriba deben tomar precaución para evitar división por cero. En casi todos los casos en que la fórmula de arriba es apropiada, la serie predecida tendrá cero probabilidad de ser igual a cero. No obstante, una programación cuidadosa debe fijar un valor base en el denominador. Un procedimiento común seguido cuando el denominador es menor que un mínimo predefinido es poner el radio igual al denominador, mejor que dividir el numerador por el valor base. Cual método usar es un problema dependiente, pero ambos deben ser considerados.

La diferencia entre el error RMS relativo (penúltima ecuación) y el error medio relativo (última ecuación) recae principalmente en el denominador. La decisión de cual usar depende de la importancia de la *variación* de las series comparado con el *valor actual* de la serie. Si estamos midiendo la posición de una partícula en un flujo turbulento, ya sea que la partícula este en un extremo u otro, o cerca al centro de su rango, es probable que sea de poco interés considerar el error. Probablemente la mayoría de las veces desearíamos comparar el grado de error con el grado de variación en conjunto. Por lo tanto, la penúltima ecuación será más apropiada. Por otro lado, podemos intentar predecir un precio de almacenamiento. En este caso, usando una variación total en el denominador no sería bueno. Sería preferible ser capaz de decir que nuestra predicción del error es una cierta fracción del precio actual. Así, la última ecuación será más apropiada.

La medida del error anteriormente mencionadas no significa una lista completa de posibilidades. Otra es la relación señal-ruido. La consideración más importante es el problema en sí mismo ¿Qué queremos medir? Si esto es relativo a cualquier cosa, ¿a qué es relativo?. ¿El propósito de la medición del error tiene sentido?. ¿Puede una persona no experimentada tener al menos una comprensión de su significado? Solamente siendo cuidadoso al responder estas preguntas sus respuestas pueden llegar a ser la mejor forma de describir la predicción del error.

1.4 CRITERIOS DE SELECCION DE VARIABLES DE ENTRADA

Cuando se consideran muchas variables, para fines de discriminación debemos resolver la duda de si realmente son necesarias todas para una predicción eficaz o determinar, cuales son las variables que dan una mejor predicción.

Se han propuesto procedimientos de selección de las variables que pueden dar lugar a cierta guía a los investigadores que estén deseando seleccionar un subconjunto de las variables medidas para usarlas con fines de discriminación. La mayoría de los procedimientos existentes de selección de variables son un tanto semejantes a los procedimientos correspondientes que se usan para los problemas de regresión múltiple. [Dallas E. J] describe el procedimiento de Selección Hacia delante, Eliminación Hacia Atrás y Por Pasos que combina los dos primeros.

Para nuestro caso utilizamos el segundo que es el que el mismo autor recomienda, pero trabajando con una función de Regresión Múltiple.

También utilizamos el Método de Correlación, combinado con la Dispersión por ser muy conocido y practico en el campo de los procesos químicos.

Finalmente se describen las variables recomendadas por la experiencia operativa y el diseño del fabricante.

1.4.1 Por Regresión Múltiple

Utilizamos la regresión aplicando el procedimiento de eliminación hacia atrás que se inicia con la inclusión de todas las variables candidatas posibles como discriminadoras, después de lo cual se elimina la variable aparentemente menos útil para la discriminación. La variable seleccionada a eliminarse se identifica al realizar una regresión múltiple para cada variable usando todas las demás variables como covariables. Si todas las variables son estadísticamente significativas en un nivel elegido de significación, el procedimiento se detiene; pero si algunas de las variables no son estadísticamente significativas, entonces el procedimiento elimina la variable que sea menos significativa.

Si se elimina una variable, entonces el procedimiento de eliminación hacia atrás busca una segunda variable para quitar. La variable que debe eliminarse a continuación se identifica al realizar una Regresión Múltiple con las variables restantes. Si todas las variables restantes son estadísticamente significativas, el procedimiento se detiene, si algunas de las variables restantes no lo son, entonces el procedimiento busca una tercera, etc.

En cada paso se usa la función regress del Matlab, utilizando todas las variables restantes como covariables, para seleccionar la siguiente variable que se va a eliminar. El procedimiento de eliminación hacia atrás se detiene cuando todas las variables restantes son estadísticamente significativas.

Utilizando la regresión se clasifica ordenadamente la influencia de las variables de entrada sobre la salida utilizando el procedimiento de Eliminación hacia Atrás, anteriormente descrito.

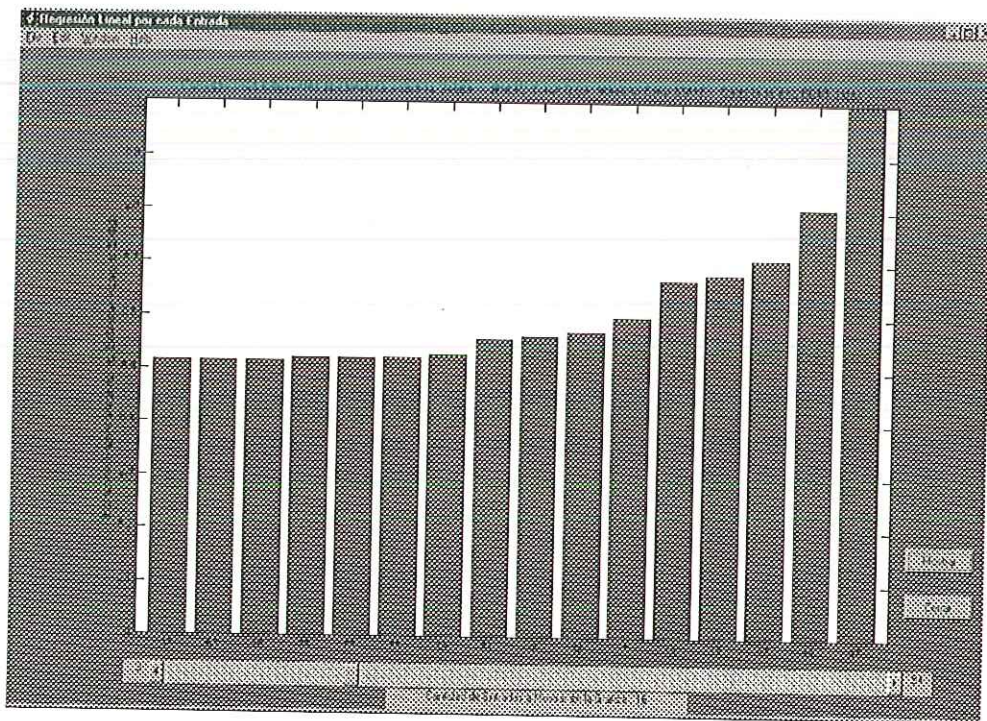


Figura 1.25 Resultado ordenado de las importancia de las variables de entrada para predecir la salida por regresión, eliminando las de menor importancia (una corrida de ejemplo).

Utilizando este método y realizando estadística de las ocurrencias para los diferentes casos analizados, se pueden encontrar las variables de mayor influencia sobre la salida. Se sugiere utilizar el siguiente programa realizado también en Matlab, para tales efectos (función Prioriza.m que se encuentra en la carpeta de la aplicación en el CD)

```

Cantidad=6; %Cantidad de Casos x1, x2, x3 ....
% El número de valores en x1, x2,...xn es igual a la cantidad de valores en pesos
% En la comparación utilizar un solo método Ej. solo Regress
x1=[57,34,48,40,54,9,43,46,15,36,10,18,47,17,52,64]; % Regress Sep 29 DAO
x2=[10,40,34,44,57,9,41,48,18,52,50,19,35,14,37,30]; % Regress Sep 28 DAO
x3=[14,47,33,56,39,15,10,57,34,36,43,31,64,51,24,29]; % Regress Ago 8 Med-Liv
x4=[45,36,6,41,37,49,30,52,56,13,19,9,27,23,35,53]; % Regress Ago 14 Med
x5=[12,56,15,18,14,6,40,50,19,53,29,45,51,60,26,52]; % Regress Ago 18 Liv
x6=[11,37,19,5,46,40,45,18,25,56,10,43,8,24,50,54]; % Regress Sep 26 Med-DAO
Peso=[16,15,14,13,12,11,10,9,8,7,6,5,4,3,2,1];
[mx npeso]=size(Peso);
[mx nx1]=size(x1);
Variable=x1;
Total=Peso;
Repeticion=x1;
Repeticion(1:nx1) = 1;
for k=2:Cantidad
    eval(strcat('x2=x',num2str(k),':'));
    [mx nx2]=size(x2);
    [mx nx1]=size(Variable);
    for i=1:nx2
        Indica=0;
        for j=1:nx1
            if x2(i)==Variable(j)
                Total(j)=Total(j)+(npeso-i+1);
                Repeticion(j)=Repeticion(j)+1;
                Indica=1;
            end
        end
    end
    if Indica==0
        Variable=[Variable x2(i)];
        Total=[Total (nx2-i+1)];
        Repeticion=[Repeticion 1];
    end
end
end
clc;
Orden=[Variable' Total' Repeticion'];
Orden=SORTROWS(Orden,[3 2])

```

Las variables seleccionadas por regresión fueron:

VARIABLE #	TAG
V63	R_AFC_F
V32	FRC7005
V56	FC
V33	FRC7006
V48	PIC7010
V55	RRC
V45	PI7003
V60	BALANCE
V26	FR7004
V43	PI7001

Tabla 1.3 Las 10 Variables de Mayor Influencia por Regresión
(case27-sep-2000.mat)

1.4.2 Por Correlación y Dispersión

El coeficiente de correlación proporciona una medida de la asociación lineal entre dos variables. Esto es muy importante para comprender por que parece que la palabra correlación se usa de manera equivocada en forma general; aparentemente decimos que las variables están correlacionadas cuando creemos que están relacionadas y que las cosas no están correlacionadas cuando creemos que no están relacionadas.

No estar correlacionadas es equivalente a ser independientes, cuando dos variables tienen una distribución normal conjunta, pero esto no se cumple en general. En la [Figura 1.27](#), se muestra una grafica de dispersión de una variable aleatoria proveniente de una población de dos variables (también conocida como bivariada). Se puede ver que cuando las parejas de variables no están correlacionadas, no se puede concluir con seguridad que las dos variables no están relacionadas.

Nunca se debe calcular un coeficiente de correlación entre dos variables sin también situar las dos variables, una contra la otra, en una gráfica. Si es importante determinar la correlación entre dos variables, entonces es de igual importancia trazar una gráfica de dispersión entre ellas. Por lo común, la gráfica de dispersión revelara, si en realidad, las dos variables están relacionadas entre si y cuándo están relacionadas entre si, la propia grafica mostrará a menudo como podrían estar relacionadas.

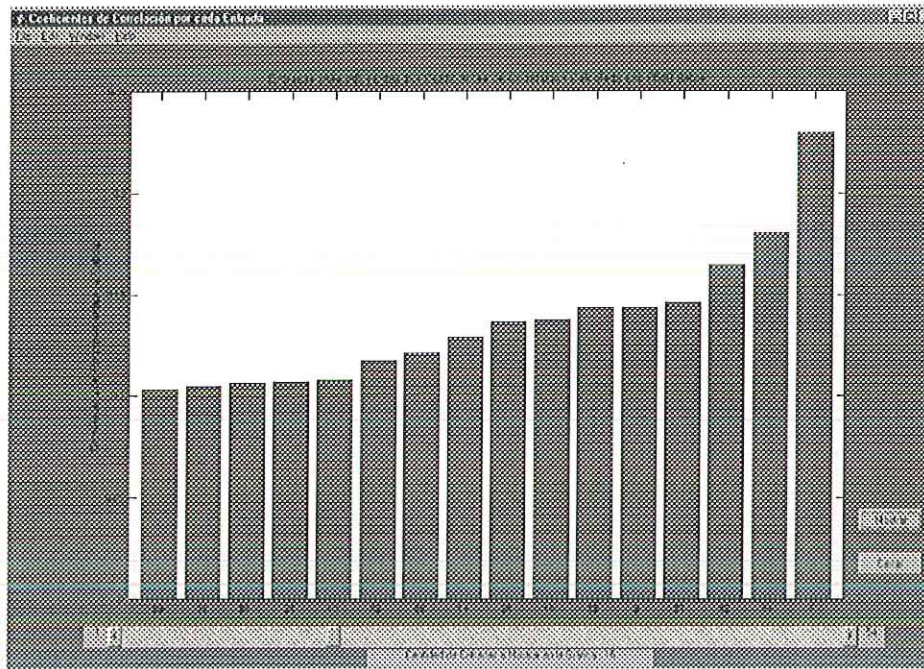


Figura 1.26 Resultado ordenado de las importancia de las variables de entrada para predecir la salida por correlación

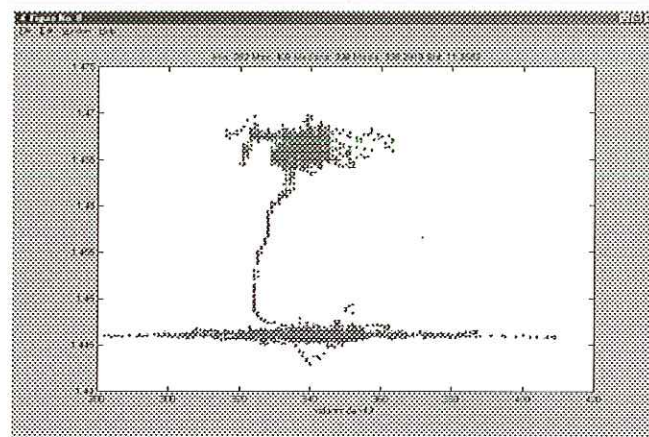


Figura 1.27 Grafica de Dispersión de la variable 57. Una alta correlación con alta dispersión no debe ser tomada en cuenta para la predicción.

Utilizando este método y realizando estadística de las ocurrencias para los diferentes casos analizados, se encontró que las variables que mayor influencia tienen por correlación y dispersión son:

VARIABLE #	TAG
V6	TI71B_36
V18	TI71A_01
V14	TI71B_34
V10	TI71A_06
V32	FRC7005
V63	R_AFC_F
V17	TI71B_37
V15	TI71B_35
V44	PI7002
V7	TI71A_24

Tabla 1.4 Las 10 Variables de Mayor Influencia por Correlación

Para llevar todo esto a la práctica se utilizó la función `corrcoef` que suministra el Matlab. Se clasifica ordenadamente la influencia que arroja este análisis para determinar cuales variables afectan más a la salida. Los resultados que entrega la función varían entre -1 y 1 , siendo estos resultados extremos los de mayor influencia para la salida.

1.4.3 Por el Experto

Variables seleccionadas por el experto según diseño original de la Planta de Fenol:

TAG	V	SERVICIO	CAPT.	MUESTRA
TI71B_37	V17	TEMPERATURA CIMA T753A	TR	20 s
FR7001	V24	FLUJO RAFINATO	TR	20 s
FR7002	V25	FLUJO RAFINATO SOLUCION H751	TR	20 s
FRC7003	V30	FLUJO EXTRACTO SOLUCION	TR	20 s
FRC7006	V33	FLUJO FENOL	TR	20 s
LR7002	V39	NIVEL FENOL D751	TR	20 s

LRC7005	V41	NIVEL T753B	TR	20 s
LR_7003		NIVEL D752	MAN	1 h
MIC7003	V42	% VÁLVULA CHORRO LATERAL	TR	20 s
PRC7002	V50	PRESION T753A	TR	20 s
PRC7003	V51	PRESION T753B	TR	20 s
DELTA_705	V59	DIFERENCIA TEMP CIMA-FONDO T755	TR	20 s
IR_7001	V61	CALIDAD DEL RAFINATO	TR	20 s
FRC7001	V28	FLUJO CARGA T755	TR	20 s
TI71B_36	V16	TEMPERATURA DE CIMA T755	TR	20 s
TI7001-30		TEMPERATURA CIMA T753B	MAN	1 h
FRC7005	V32	FLUJO AGUA FENÓLICA CIMA	TR	20 s
		PUNTO DE CHISPA DE LA CARGA	LAB	1 día
		PUNTO DE EBULLICIÓN INICIAL	LAB	1 día
		GRAVEDAD API DE LA CARGA	LAB	1 día
		% FENOL EN EL RAFINATO SOLUCION	LAB	1 día
		TIPO DE CARGA	MAN	Al Cambiar
		COLOR AGUA	MAN	1 h
		COLOR FENOL	MAN	1 h
		% ACEITE EN AGUA FENOLICA	MAN	6 por día
		% FENOL EN AGUA FENOLICA	MAN	
		% ACEITE EN FENOL	MAN	6 por día

Tabla 1.5 Variables Seleccionadas por el Experto

Algunas variables son tomadas del archivo histórico del sistema de monitoreo y control de la planta (TR) con una frecuencia de 20 segundos, otras deben ser tomadas manualmente (MAN) con periodos de 1 hora típicamente y otras requieren el resultado del laboratorio (LAB) que se ejecutan 2 ó 3 veces por día.

Como se puede observar esto requiere una alta disponibilidad de los diferentes equipos desde la instrumentación bien calibrada, comunicaciones, bases de datos, hasta la rutina operativa de tomar los datos manuales y las muestras para análisis en laboratorio. Esto hace de este modelo una tarea un poco ardua de llevar a cabo pero justificable para obtener mejores resultados.

1.5 PREPARACION DE LOS DATOS DE ENTRADA

Cuando se tiene conocimiento del formato en que se encuentran los datos, podemos empezar a recolectarlos y en nuestro caso contamos con una gran facilidad ya que se tiene una arquitectura de red de datos tipo ethernet que interconecta todos los sistemas de monitoreo y control a esta red y permite bajar datos en forma remota desde sitios que cuentan con la debida autorización para hacerlo.

Para la etapa de entrenamiento y validación no es estrictamente necesario contar con esta facilidad ya que el programa se desarrollo con las opciones para importar datos de la hoja de calculo Microsoft Excel y bien podría utilizarse la vía del diskette con los datos del proceso, pero para la puesta en marcha del proyecto obviamente lo practico del sensor virtual es poder contar con una interconexión entre la plataforma donde se encuentra el monitoreo y control del proceso y la maquina donde se ejecute el algoritmo del sensor virtual entrenado a menos que dicha aplicación resida en la misma maquina.

Para el caso, las muestras son tomadas vía red de datos entre maquinas de la misma familia de Microsoft , Windows NT server para el sistema de Monitoreo y Control llamado Factory Link donde reside la información en un archivo de historia (2 días con periodo de muestreo de 20 segundos, 8640 datos) en formato de base de datos (*.cdb) y Windows 95 instalado remotamente para la captura de la información de la planta y posterior alimentación a la red neuronal para el entrenamiento y validación fuera de línea. En esta ultima maquina se encuentra la aplicación del Sensor Virtual pero dadas las características del lenguaje en que se desarrollo, permite migrarlo a plataformas de Windows o Unix si se quiere tenerlo localmente.

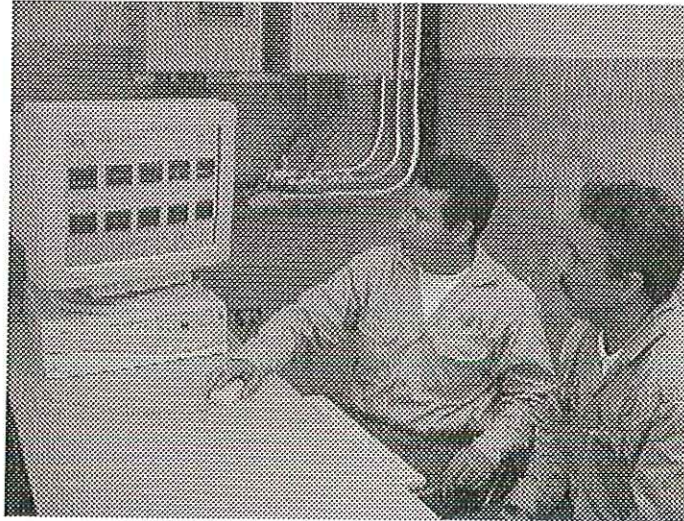


Figura 1.28 Factory Link (Monitoreo y Control de la Planta)

Mostraremos aquí algunos asuntos relacionados con el tamaño y alcance del entrenamiento y se discutirá sobre técnicas válidas para controlar la calidad de la colección de datos. La calidad de datos se define en términos de la conveniencia para el entrenamiento de una red neuronal sobre una tarea dada y la validación de datos, es el proceso para garantizar la conveniencia del entrenamiento de datos.

1.5.1 Colección de Datos

Los factores que determinan cuantos datos necesitan ser recolectados, son los relacionados con la razón que se necesita para optimizar el tamaño de la red o extender el conjunto de datos para mejorar la capacidad generalizada de la red. Muchos métodos son finalmente empleados dependiendo del costo y viabilidad de los datos, el tiempo de entrenamiento de la red, y el nivel de ruido. Se requiere habilidad para su generalización y simplicidad de la red, para cada aplicación en particular.

El tamaño de la red entrenada está un poco restringido por el tamaño del conjunto de entrenamiento. La regla más simple fue la de asegurar que el conjunto de entrenamiento sea más pequeño o igual a $(1 / \text{mínimo error de destino}) \times \text{número de pesos}$. Un error mínimo de 0.1 requiere un conjunto de entrenamiento 10 veces más grande que el número de pesos en la red neuronal. Las siguientes consideraciones son afectadas también por el tamaño de la red y las dos pueden ser negociadas una con respecto a la

otra hasta cierto punto. El intercambio es importante cuando la colección de datos es costosa o existen datos que están esparcidos.

Se conceptúa que el tamaño del conjunto de entrenamiento está determinado por la dimensionalidad intrínseca, la resolución requerida, y la probabilidad de distribución, la "complejidad" la cual está determinada por lo bien que el sistema empieza a ser modelado con la interpolación entre ejemplos y el nivel de ruido del sistema que será modelado.

La dimensión intrínseca de un sistema está determinada por un número de restricciones las cuales existen entre las variables de entrada. Las restricciones entre las entradas limitan el número de combinaciones del patrón de entrada posibles. Aunque el tamaño de la entrada o distancia crece espontáneamente con la suma de nuevas variables, las restricciones entre estas variables reduce el crecimiento del tamaño. Las variables continuas pueden ser restringidas por una y otras leyes de la naturaleza.

Si el número de restricciones en un sistema es desconocido, entonces la dimensionalidad intrínseca es calculada simplemente como el número de entradas por unidad menos el número de restricciones. Un riguroso análisis basado sobre la covarianza o mutua entropía cuenta entre las diferentes unidades de entradas dando una medida exacta del número, y grado, de las restricciones pero requieren una gran cantidad de trabajo que puede no proveer un beneficio suficiente sobre el que se obtendría al tener un razonable entendimiento de la tarea, si la dimensión intrínseca no se conoce, o si las entradas son independientes, entonces la máxima dimensionalidad es igual al número de entradas por unidad.

Resumiendo, la capa de entrada más que contener pocas variables debería tener la dimensión intrínseca de los datos constantes pero reducirá el número de pesos en la red y reduce el número requerido de entrenamiento en el ejemplo en relación con el tamaño de la red. Si es posible identificar una pequeña dimensión intrínseca en un común de variables, entonces sería posible luego reducir el número de variables usadas para entrenar la red. Esto es deseable al reducir la complejidad del modelo y así beneficia su habilidad de generalización. El requerimiento indicado de puntos al construir un modelo sobre un espacio n dimensional es alrededor de 2^n datos. A menudo es posible tener éxito con mucho menos, sin embargo, la solución suele recaer sobre un subespacio o

diversos espacios dentro de todo el espacio del sistema. La dimensionalidad intrínseca de una serie de tiempos esta determinada por la dimensión encajada.

La **resolución** de datos que serán modelados es decidida por el número de divisiones hechas a lo largo de cada dimensión. Por codificación binaria, la cuestión es trivial, como cada unidad solamente toma uno de dos valores. Para valores continuos, y valores probables calculados en lugar de clasificación binaria, la medida puede ser limitada solo por la exactitud del computador sobre el cual la red es implementada. Ciertas escalas naturalmente pueden caer sobre un número discreto de pasos: precipitado en centímetros, por ejemplo, si no estamos interesados en alguna gran ocurrencia, entonces podemos evitar la necesidad de grandes cantidades de datos pero limitando la escala apropiadamente y existe la necesidad de igualar la resolución entre las variables de entrada y de salida, así que no esperamos una mayor resolución de la que existe en la entrada.

Un sistema complejo es aquel que muestra gran variación entre puntos cercanos. Mientras la capacidad generalizada de una red neuronal esta basada en la construcción de una función sin problemas entre un conjunto de puntos de entrenamiento, algún comportamiento que caiga entre los puntos de datos lo suficientemente diferente desde una interpolación uniforme de estos puntos no será correctamente moldeado. En tales casos, el conjunto de datos puede ser probado a una escala lo suficientemente buena, esa interpolación uniforme es válida. Esto es análogo a la necesidad de probar la señal; para probar una señal a una frecuencia lo suficientemente alta para capturar la frecuencia requerida. La complejidad es absolutamente relativa a la resolución de pocos puntos continuos a lo largo de una escala que proporcione menos cambios para la conducta compleja entre puntos.

A mayor complejidad del sistema, mayor número de peso requerido para modelar el sistema. Teniendo una arquitectura de red igualada a la complejidad de los datos, el tamaño del conjunto de datos puede derivarse del tamaño de una sola red, la conversión también es verdadera.

Se necesita una **distribución uniforme** del conjunto de entrenamiento. Las consecuencias de estos requerimientos de colección de datos son dobles. En el primer

caso, hay muchas formas para controlar el número de ejemplos para la colección relacionada en el sistema con cada número de estados. En este caso, debemos tener cuidado de recoger un conjunto balanceado de ejemplos extendidos a través del rango operativo del sistema. Esto puede ser tan sencillo como escoger los ejemplos con cuidado, o tan complejo como dirigir una planta a través de cada aspecto de su conjunto normal de operaciones. Seguidamente si no somos capaces de controlar la distribución de los ejemplos de la colección, pero somos capaces de medir la distribución, podemos recolectar suficientes datos para asegurar completamente la covarianza del rango de sistema operativo. Si conocemos que la probabilidad del sistema ingresándole un estado dado es P , entonces debemos coleccionar al menos $1/P$ puntos de datos para estar razonablemente seguros de coleccionar un ejemplo sencillo de s . El último punto, aunque trabajable en una dimensión, es considerablemente confuso para cualquier práctica propuesta cuando más de dos o tres variables de entrada son usadas. Con un número grande de variables, cada una con desigual distribución, es mejor seguir el heurístico que menciona que:

A medida que la varianza en la distribución del conjunto de entrenamiento crece, así mismo crece el número de ejemplos de entrenamiento requeridos para estar seguros de cubrir el rango del sistema en operación.

1.5.2 Ruido y calidad de los datos

Tenemos visto que sumando ruido al conjunto entrenamiento incrementa la habilidad de la red para generalizarse al "embotar" el ruido presente en los datos de entrenamiento. La colección de datos extras tiene un efecto similar sobre el ruido a medida que los ejemplos adicionales reforzarán cualquier estructura en los datos e introducirá ruido que confundirá el efecto de cualquier ruido en el pequeño conjunto de entrenamiento. Ruido, en este caso, es definido como perturbaciones de los puntos de datos registrados para sus verdaderos valores debido a inexactitudes en los procesos de medida o de colección.

El ruido también puede estar presente en la forma de valores erróneos, desclasificaciones o en fuera de lugar. En este caso, puede verse que si el 20% de los puntos de datos coleccionados son ruido y no tienen efecto sobre la solución final de la red, entonces nuestro conjunto de entrenamiento es efectivamente el 20% más pequeño que la idea

original. En tal caso los datos extras pueden ser coleccionados permitiéndole ruido y, donde sea posible, descartar los ejemplos erróneos.

El número de pesos en una red neuronal puede ser conservado bajo los efectos mínimos del ruido. En tales casos, podría aparecer que son requeridos menos datos de entrenamiento a medida que deban ser usados menos pesos. Como hemos visto, éste no es el caso. Dada la incertidumbre que rodea la elección del tamaño de una red, es a menudo prudente elegir una red con más unidades ligeramente escondidas (y consecuentemente los pesos) de las necesarias e incrementar el tamaño del conjunto de entrenamiento de acuerdo a ello. En tales casos el intercambio puede ser decidido por la facilidad y el costo de la colección de datos. Si los datos están esparcidos y son costosos entonces deben ser obviamente preferidos los métodos basados en redes.

Estas consideraciones son resumidas diciendo que si un sistema de salida varía sobre una fina escala en un amplio rango, necesitamos muestrearlo bajo un gran número de puntos. Adicionando dimensiones incrementa el espacio del cual se muestrean los puntos exponencialmente, pero las restricciones entre dimensiones reducen esta explosión. Si los puntos mostrados no son esparcidos uniformemente por todo el espacio, necesitamos coleccionar más puntos para compensar, y si los puntos contienen ruido, nuestro número efectivo de puntos coleccionados caería.

1.5.3 Datos de entrenamiento y Datos de Prueba

Hecht Neilson correctamente sugiere que la primera cosa que debe hacerse una vez el conjunto de entrenamiento ha sido definido es construir el conjunto de prueba. Los resultados obtenidos del conjunto de la prueba deben definir si el proyecto ha sido un éxito o no, y también debe ser lo más exhaustivo y potencialmente destructivo posible. Todos los aspectos del funcionamiento normal deben ser tenidos en cuenta junto con las pruebas de respuesta de la red a entradas fuera del rango normal.

Sin embargo el objetivo definido en el plan de proyectos debe ser incluido por el conjunto de prueba y debe persistir a través del desarrollo del proyecto. Parecería prudente la construcción de un completo conjunto de entrenamientos covariando todos los aspectos de los sistemas para lo cual el conjunto de prueba será derivado.

Maren sugiere un método llamado "Shaping" el cual involucra un subconjunto tomado del conjunto total de entrenamiento y rápidamente entrenar una red a la figura general de éste subconjunto. Cuando la red está cerca a una solución, el resto de los datos de entrenamiento son usados para refinar el modelo. Este método aplica un pequeño conjunto de entrenamiento a una red diseñada para una grande y así, en lugar de construir un modelo simple rápido, fácilmente construiríamos un súper modelo el cual necesitaría ser cancelado por un conjunto total de entrenamiento. Como la mayoría de las fases de aprendizaje ocurren en los primeros ciclos de entrenamiento, este método nos beneficiaría poco.

1.5.4 Procesamiento de la señal (filtraje)

Si somos capaces de almacenar una ventana de valores previos de una fuente de entradas o una red de retardo en el tiempo entonces también podemos procesar datos en esa ventana para que la información sea explícita sobre el modelo en que está basado. Es un caso común que la información importante está contenida en los componentes de frecuencia de la señal. Hay dos métodos populares para extraer bandas de frecuencia de la señal de interés: utilizando un conjunto de filtros o usando transformada de Fourier.

Un filtro toma una señal y remueve un conjunto de frecuencias definidas. Los filtros pasa banda cortan todas las frecuencias que están fuera del rango dado, y los títulos corta banda cortan toda la frecuencias dentro del rango lado. Los filtros pocas veces tienen forma cuadrada; es decir que el punto de corte no es agudo, pero las pendientes caen rápidamente, alternando las frecuencias en el borde del área. Un banco de filtros es un conjunto de filtros pasa banda, cada uno sincronizado en una banda diferente y exacta.

La transformada de Fourier calcula la cantidad de potencia en cada ancho de banda de frecuencias en una señal. Ella ejecuta la función equivalente de un banco de filtros conteniendo un filtro sencillo para cada banda angosta de frecuencia en la señal. El ancho de banda describe la resolución espectral de una transformada. Una transformada de Fourier cuyas salidas consisten de valores de potencia para cada frecuencia entera tienen la resolución espectral de 1 Hz.

Una función de convolución ejecuta el peso promedio sobre un chorro de datos y también provee un método útil para extraer ciertas frecuencias. Los pesos de los promedios son tomados de una ventana moviéndose sobre la señal. La misma convulsión puede ser ejecutada tomando primero una transformada de Fourier de los datos en una ventana, luego multiplicando esta transformada por la transformada de la función de convulsión. Plataforma inversa del espectro resultante es igual a la señal de la convolución. La función de convolución usada en el dominio del tiempo es conocida como la función de respuesta al impulso y es equivalente a la función de multiplicación en el dominio de Fourier como la función de transferencia. La equivalencia entre la convulsión en un dominio y la multiplicación en otro es de particular uso cuando las dos necesitan ser combinadas.

Es claro que multiplicar las bandas de potencia en la transformada por una función de transferencia con 0 salidas fuera el rango pasa banda seleccionará solamente las bandas requeridas. Para seleccionar tal ancho de bandas de una señal en el dominio del tiempo debemos hacer la convolución usando la función de respuesta al impulso al igual que la transformada de Fourier de la función de transferencia. Obviamente, una vez más sabemos que la forma de la función de respuesta al impulso para los anchos de banda requeridos no necesitan llevar más ninguna transformación de Fourier.

Como la transformada de una onda cero pura es un pico sencillo (conocido como función delta), la función de respuesta al impulso requeridas para filtrar una señal y extraer una frecuencia es una onda cero de esa frecuencia. Haciendo la convolución de una señal con una onda cero de una frecuencia nada es consecuentemente equivalente a filtrar la señal a esa frecuencia. El resultado de tal operación de filtrar es una onda cero sencillo con frecuencia igual a aquella del filtro y una amplitud igual a aquella de la misma componente de frecuencias en la señal original. Una línea plana indica que de la señal no contiene componentes en la frecuencia de filtrar. La función de convolución deberá ser una onda cero que cubra exactamente el ciclo de cero a cero. Su tamaño dependerá de la rata de muestreo y la frecuencia. Llevar esta convulsión a una señal en que la potencia a la frecuencia la varilla con el tiempo produce una onda cero de frecuencia constante y amplitud variada (por ejemplo una señal de amplitud modulada).

1.5.5 Datos Falsos

En esta sección se trata el problema de que hacer con los valores aislados que bien pueden ser falsos. Los instrumentos que capturan datos ocasionalmente presentan saltos en la medida. El ruido se monta sobre la línea de comunicación. Los operadores cometen errores al entrar los datos. La respuesta a que tan serios son estos problemas, como podemos detectarlos y como podemos corregirlos, pueden ser encontradas en estadística apropiada y libros de análisis de datos. Aquí brevemente revisaremos aspectos del problema que es relevante para las redes neuronales.

Debemos tomar severas precauciones contra el rechazo de datos no adecuados a menos que tengamos una firme razón a priori para creer que nuestros datos pueden estar contaminados, debemos restringir nuestro deseo de hacer cosas severas. Hay dos principios para tener en cuenta cuando se encara el dilema del rechazo de datos.

Primero, recuerde que algunas anomalías son normales. Por definición de la tarea, estamos examinando los datos falsos de nuestra colección de datos. En cualquier muestra habrán datos falsos. Cerca del 90% de la distribución normal cae dentro del 1.5 de la desviación estándar de la media. Incluso de una colección ideal normalmente distribuida de 100 muestras se puede esperar que tengan al menos un punto con una desviación estándar de 2.5 alejado. En una grafica dispersa, este punto puede llamarnos la atención. Y esto no toma mucha partida de la normalidad para incrementar este efecto dramáticamente. De tal forma de que una regla del pulgar es que usualmente no se deben rechazar puntos a menos que realmente sean malos.

El otro principio importante de la eliminación de datos es que debemos muestrear completamente las posibilidades inherentes en la distribución de los datos. Si nuestro conjunto de entrenamiento no es representativo de los datos sobre los cuales la red será probada, estaremos perdiendo nuestro tiempo. Aun si una muestra sospechosa es tan mala que no hay posibilidad de esperar un dato falso de la misma distribución como del resto de los datos, considere la posibilidad de que esta pertenezca a otra distribución, perfectamente valida. Quizás esta pasando mas en el experimento de lo que nosotros nos damos cuenta. Para ejecutar una eliminación de datos sin intervención manual de los valores observados es necesario meternos de lleno en el problema. A menos que podamos satisfacer nuestras propias dudas sobre la observación de valores erróneos

debemos esforzarnos para encontrar una explicación de los valores, y quizás modificar nuestros procedimientos de recolección a tal punto de incluir sus particularidades.

Otro argumento a favor de mantener cuestionables los datos falsos es innato a la robustez a la mayoría de los modelos de redes neuronales. Los investigadores acostumbrados a técnicas lineales tradicionales toman con precaución siempre los datos falsos. Las redes que tienen capas con funciones de activación sigmoide tienen normalizadores de datos construidos dentro de ellas. Por ejemplo, si una neurona tiene una función de activación logística, teniendo una entrada neta de 2.5, su salida será 0.92 como valor razonable. Si un punto de datos brutos maneja la red de entrada con hasta un millón, su salida solo se incrementara ligeramente a menos de 1.0, habiendo absorbido el choque bien. Naturalmente, permanecemos enfrentando el problema de preguntarnos por aprender una parte incorrecta de la información, asumiendo, por supuesto, que la información es incorrecta. Si el conjunto de entrenamiento es razonablemente grande, esto no seria un problema para las redes que aprenden minimizando el error medio a través del conjunto de entrenamiento, los datos falsos tendrán un impacto comparativamente pequeño sobre los pesos encontrados, siendo primero tratados por la función de achatamiento (squashing), inundado por la mayoría de los datos correctos.

Algunas redes tienen la habilidad implícita de procesar datos brutos sin que afecten el aprendizaje de los datos normales, las redes neuronales probabilísticas caen dentro de esta categoría. A menos que el parámetro de suavización sea inusualmente grande, los puntos de datos únicos serán separados del resto de los datos que no tienen una contribución esencial a las funciones potenciales que determinan la respuesta de la red, es decir a menos que puntos de muestras inusuales sean significativas y se encuentren valores relativos cuando la red sea usada. En este caso, los puntos alejados harán la contribución más valorable. Si la meta es la clasificación y se esperan datos inusuales este es el tipo de red que frecuentemente se escoge. Algunos métodos para tomar la decisión en el momento de descartar los datos falsos utilizan la media y la desviación estándar de las muestras, asumiendo que están normalmente distribuidas y se decide basado en las marcas Z. Estos marcadores de los puntos de una muestra se encuentran sustrayendo la población media y dividiendo por la desviación estándar:

$$Z = \frac{x - \mu}{\sigma}$$

sin embargo en nuestra opinión la automatización del rechazo de posibles datos falsos no debe ser hecha. El método recomendado es graficar un histograma de la muestra y cuidadosamente revisar con criterio dado por la experiencia sobreponiéndose a las aplicaciones estadísticas. No es de sorprenderse encontrar muchas veces histogramas similares como el que se muestra en la siguiente figura:

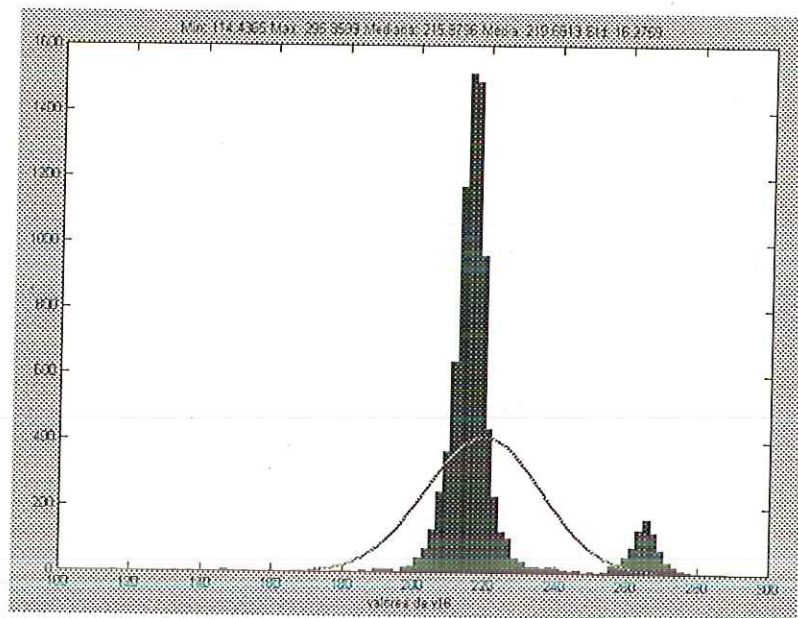


Figura 1.29 Razón para evitar el rechazo automático de datos desviados

Cuando vemos histogramas como este, inmediatamente debemos pensar en que se requiere un nuevo procedimiento de muestreo para salir de dudas

1.5.6 Escalización

No siempre es necesario utilizar la escalización pero es casi siempre recomendado. Primero, debemos distinguir entre cuando la escalización debe hacerse a diferencia de cuando estamos disponibles para elegirla. Si una variable esta siendo usada para entrenar neuronas de salida y las neuronas de salida tienen una función de activación con rangos limitados, ciertamente debemos restringir los objetivos de las activaciones a valores que puedan ser confortablemente aprendidos. Algunos modelos de redes, tales como las de Kohonen tienen limites estrictos en sus valores de entrada. En estos casos no hay nada que hacer. Debemos escalizar.

Algunas veces, sin embargo, tenemos la opción de trabajar con datos brutos. Es posible diseñar redes multicapa retroalimentadas con la función identidad como función de activación de las neuronas de salida (por supuesto, si las neuronas de la capa oculta son lineales, el problema se reduce a una regresión lineal). Tales redes pueden tener salidas sin límites, de tal forma que no se necesitan escalizaciones de las salidas teóricamente. Las redes retroalimentadas no tienen límites implícitos en sus entradas, así que no se necesitan escalizaciones para las variables de entrada. Sin embargo, la escalización para rangos uniformes es recomendada.

Una de las razones para la escalización uniforme es inicializar la importancia de las variables igualadas. Teóricamente es posible que una red aprenda la importancia relativa de las variables ajustando los pesos de entrada. Si una de las variables tiene un orden de magnitud de un 1.000.000, mientras que otra está en el orden de 0.000001, la red deberá ser capaz de aprender a usar pequeños pesos de entrada para la primera variable y grandes pesos para la segunda, pero esto sería muy incierto. Se ha consultado muchos algoritmos de aprendizaje para superar tales rangos. De hecho muchos algoritmos de aprendizaje imponen límites fijos sobre sus pesos para prevenir caer en obsolescencia. Estos algoritmos simplemente no serán capaces de aprender de tales datos extremos. Nosotros podemos hacer que la vida de la red sea un poco más fácil dándoles datos escalizados de tal forma de que todos los pesos puedan permanecer pequeños dentro de rangos predecibles.

Esto se torna más importante cuando los datos están siendo aprendidos por las neuronas de salida. La mayoría de los algoritmos de entrenamiento minimizan el error total de todas las salidas. Si las variables de la salida están desigualmente escalizadas, con grandes variaciones estas serán favorecidas y serán predominantes en la suma del error. Esto puede tener profundas consecuencias y es un problema que debe ser siempre considerado.

Otra razón para escalizar los datos es mejorar la interpretabilidad de los pesos de la red. Podemos desear ver los pesos encontrados por nuestra red, con la esperanza de que tengan algo de sentido sobre que usa la red para tomar sus decisiones. Bajo las mejores condiciones esto puede ser una tarea muy difícil.

2 DISEÑO DE UN SENSOR INTELIGENTE PARA INFERIR EL IV EN LA PLANTA DE EXTRACCIÓN DE ACEITES LUBRICANTES CON FENOL

2.1 HERRAMIENTA DE PROGRAMACIÓN (MATLAB)

Matlab ofrece operaciones con arreglos que nos permiten manipular conjuntos de datos en una amplia variedad de formas rápidamente. Además de su orientación hacia el manejo de matrices, Matlab ofrece características de programación similares a aquellas que se utilizan en otros lenguajes de programación de computadores. Finalmente, Matlab ofrece una interfase gráfica de usuario (GUI), herramienta que permite usar Matlab para el desarrollo de aplicaciones. Esta combinación de arreglos de estructuras de datos, características de programación y herramientas GUI hacen de Matlab una herramienta extremadamente poderosa para resolver problemas en muchos campos así como el de nuestra aplicación de sensores virtuales con redes neuronales, para el cual ofrece facilidades de funciones ya implementadas denominadas toolboxes. Para este caso se utilizaron algunas funciones adaptadas a nuestra necesidad, como también se crearon otras nuevas que permiten implementar el sensor virtual y todo ello se incorporó a una interfase gráfica para facilitar el manejo del sensor virtual ofrecido.

Por otro lado, esta herramienta también ofrece la posibilidad de generar código fuente en lenguaje C, lo cual sería altamente recomendable a fin de elaborar aplicaciones comerciales con mayor velocidad de ejecución y protección del código, permitiendo luego bajo cualquier plataforma realizar su compilación. Para fines didácticos y dado el sobre costo que conlleva el uso de esta herramienta de compilación, se dejó el código en su formato fuente original.

Para efectos de introducción sobre el manejo del Matlab, se dan algunos ejemplos, del poder que esta herramienta ofrece para realizar procedimientos altamente complejos, bajo otros lenguajes de programación. y que aquí con pocas líneas de código se obtienen.

Manejo de matrices:

REGRESS Regresión Lineal Múltiple Usando Mínimos Cuadrados.

$b = \text{REGRESS}(y,X)$ Retorna el Vector de Coeficientes de Regresión, B.

Dado un modelo lineal: $y = Xb$,

Espacio en tres dimensiones:

```
[X,Y,Z]=meshgrid(-10:10,-10:2:10,-10:1.5:10)
```

```
V=sqrt(X.^2+Y.^2+Z.^2); slice(X,Y,Z,V,0,0,0)
```

```
[A,B,C]=peaks;
```

```
slice(X,Y,Z,V,3*A,3*B,1.2*C)
```

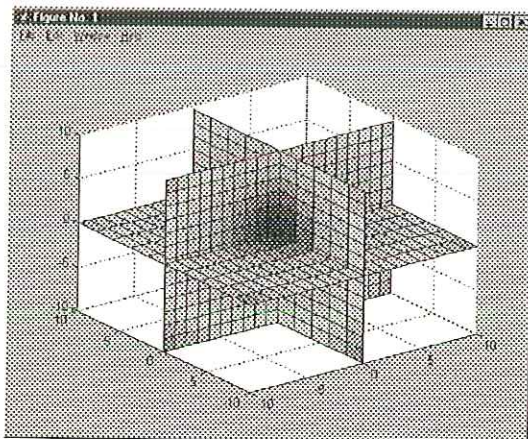


Figura 2.1 Ejemplo 1 Gráficas Matlab

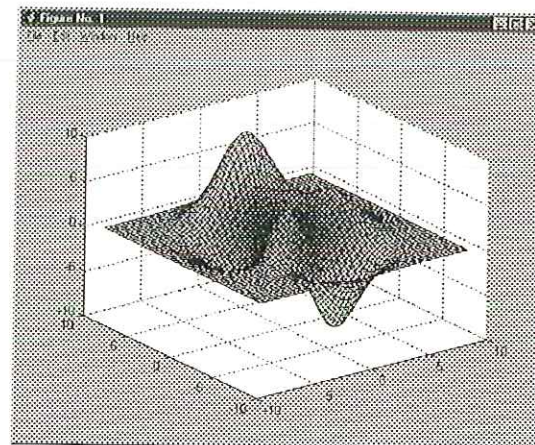


Figura 2.2 Ejemplo 2 Gráficas Matlab

2.2 DIAGRAMA DE BLOQUES FUNCIONAL DEL SENSOR INTELIGENTE

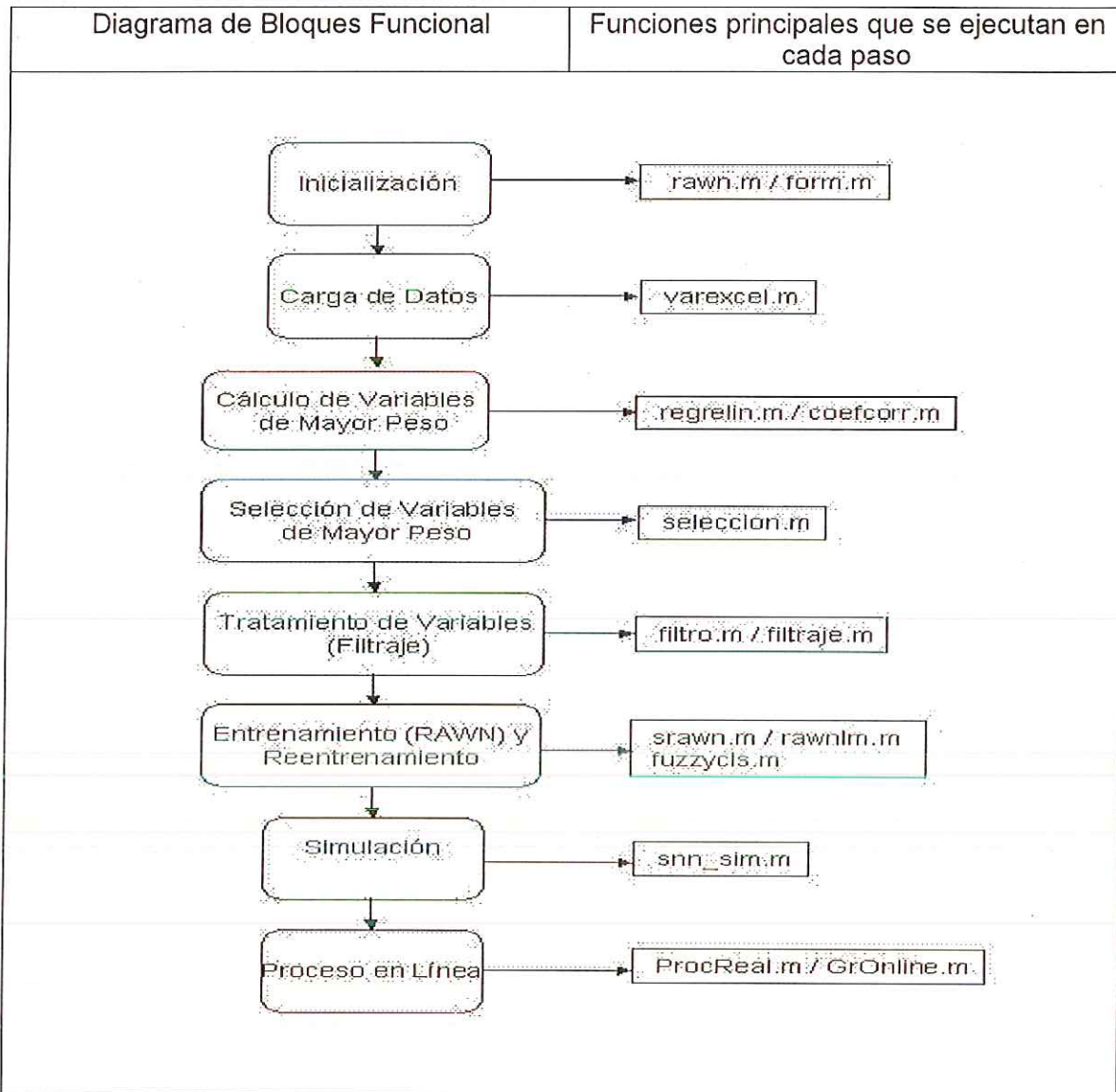


Figura 2.3 Diagrama de Bloque y Principales Funciones del Sensor Virtual

2.2.1 Funciones de Importancia para el Sensor en Matlab

A continuación se presenta una tabla con las más importantes funciones utilizadas en el sensor, cabe destacar que estas funciones se encuentran ordenadas en la tabla según su funcionalidad como lo indica cada una de los nombres de las columnas de la tabla

Funciones de Interfaz de Usuario	Funciones de Carga de Datos y Config.	Funciones de Tratamiento de Datos	Funciones de Trat. y Cálculo Previo	Funciones de Cálculo Medio	Funciones de Entrenamiento	Funciones de Simulación	Funciones de Proceso Real en línea
Rawn	VarExcel	filtro	Regrelin	Purlin	Srawn	Snn_sim	Proreal
Form	Arranque	Normal	Coefcorr	Lmtr	Fuzzycls	GrOfflin	GrOnline
Filtraje1	Selección	desnormal	calregre	Fretr	Rawnlm		
Plotgraf				Cubici2			
Plotgraf1				Cubici3			
Menaje				Constr1			
Errorbox				Tanhsim-gaus-Simoide			
graficar				sqpmimo			
Función1							

Tabla 2.1 Funciones en Matlab del Sensor Virtual

2.2.2 Código fuente en Matlab de las principales funciones

2.2.2.1 Cálculo de Pesos y Bias (función srawn)

```

%P "Matriz de variables de entrada
v = [P ones(Ne,1)];
v = v.^2;
[rijt colt]=size(v);
for col = 1 : colt
    gem(col) = mean(v(:,col));
end %for%
for col = 1 : colt
    fac(col) = sqrt(1.25 / (colt*gem(col)));
    Wh(col,:) = fac(col)*randss(1,Nh);
    if col == colt
        Wh(col,:) = (1/colt-1)*Wh(col,:);
    end %if%
    set(phiter1,'ydata',[0 col col 0]./colt);drawnow;
end %for%
B1 = Wh(Ni+1,:); B1 = B1';
W1 = Wh(1:Ni,:); W1 = W1';

```

```
%Cálculo los Pesos de Salida con el Método OLS
```

```
P = P'; T = T';
eval(['A1 = ',ActFunc,'(W1*P,B1);'])
Astar = [A1 ; ones(1,Ne)];
% TLS:
if tls
    AA=Astar'; BB=T';
    AA(:,Nh+1)=[];
    X1 = ones(length(AA),1) \ BB;
    Sigma = svd([AA BB]);
    X2=pinv(AA'*AA(Sigma(Nh+1)^2)*eye(Nh))*AA*(BB*X1*ones(length(AA),1));
    Wo1=[X2;X1];
    Wo1 = Astar' \ T';
% OLS:
else
    Wo1 = Astar' \ T';
end %if%
P = P'; T = T';
Wo = Wo1';
[a,b] = size(Wo);
W2 = Wo(:,1:b-1);
B2 = Wo(:,b);
clear Wo a b
```

```
%Presentación de los resultados del entrenamiento
```

```
Result = feval('purlin',W2*A1,B2); %Valores de la Salida virtual
TR = T - Result;
RMSE1 = sqrt(sum((TR).^2)/(length(Result)));
```

2.2.2.2 Sensor Virtual en Línea (Principales Líneas de Código)

```
%Construcción de la matriz de entradas y el vector de salida real
```

```
Yy(k:k,:)=ddereq(channel,strcat(Nomenclafilas,'2c',num2str(colsalida),':',Nomenclafilas,'2
c',num2str(colsalida)));
for i=1:length(Xtitulos)
    columnas =strcat(Nomenclafilas,'2c',num2str(Xtitulos(:,i)),':',
    ,Nomenclafilas,'2c',num2str(Xtitulos(:,i)));
    X1=ddereq(channel,columnas);
    XX=[XX X1];
end
[m n]= size(XX);
```

```
%Cálculo de el Máximo y el Mínimo
```

```
if max(max(u)) >= 0
    maxu = 1.15*max(max(XX));
else
    maxu = 0.85*max(max(XX));
end %if%
if min(min(u)) >= 0
    minu = 0.85*min(min(XX));
else
    minu = 1.15*min(min(XX));
end %if%
if max(max(y)) >= 0
    maxy = 1.15*max(max(Yy));
else
    maxy = 0.85*max(max(Yy));
end %if%
if min(min(y)) >= 0
    miny = 0.85*min(min(Yy));
else
    miny = 1.15*min(min(Yy));
end %if%
maxt = length(XX);
Ux(k:k,:)=XX;
```

```
%Calculo de la Salida virtual en Línea
```

```
evalstring3 = ['y1(k:k,:)=purlin(W2*ActFunc,(W1*Ux(k-1:k-1,:)),B1)
,B2)";]
```


3 PRUEBAS

3.1 DESCRIPCION DE LAS PRUEBAS EFECTUADAS

Las pruebas propuestas buscan encontrar los parámetros óptimos para la red neuronal que garanticen el mejor desempeño del sensor virtual a implementar en la Unidad de Fenol, dadas las condiciones operacionales a que está sometido el proceso, principalmente por los cambios de carga (Liviano, Medio y DAO).

Para ello se debe recopilar suficiente información que abarque en lo posible la mayor variación en las entradas y la salida, si las condiciones operacionales y de control del proceso lo permiten sin poner en riesgo el proceso y la calidad del producto a entregar (rafinato).

Esto puede tomar alrededor de 6 meses, recopilando archivos con tendencias históricas de las diferentes variables a utilizar en la que se incluyan, arrancadas y paradas de la planta, emergencias, condiciones normales de operación, puntos de máximo rendimiento y lógicamente los cambios de carga intrínsecos a esta operación en particular.

Posteriormente se deben reconciliar los datos capturados. Este proceso consiste en validar la consistencia de los datos que entrega la instrumentación, descartar e inclusive corregirlos para que la red neuronal aprenda adecuadamente de dicha información.

De esta forma se preparó un archivo de datos históricos con buena cantidad de información y buena calidad de la misma (case-27-sep-2000.mat) con 8640 datos, suficientes para entrenar y validar el modelo, que incluye un cambio de carga con su correspondiente variación en la salida (índice de refracción).

Para centrar más el problema en la solución de los aspectos que matemáticamente no están establecidos, se toman como constantes los parámetros definidos a continuación, de esta forma la complejidad del problema se reduce y permite hallar parámetros óptimos para el desempeño de la red neuronal para el propósito establecido.

Función de Activación: Sigmoide. Según las conclusiones de [Braake] y [Otero] en sus publicaciones (comprobadas de igual forma en nuestros experimentos).

Método de Entrenamiento: RAWN. Para obtener rápidamente los parámetros de la arquitectura la red neuronal, permitiendo el análisis de muchos casos en corto tiempo y decidir el mejor comportamiento variando los parámetros restantes. Posteriormente se re-entrena con un método completo, con los parámetros óptimos encontrados por RAWN, lográndose resultados dramáticamente superiores.

Así, los demás parámetros, tales como, el número de neuronas en la capa oculta, el número de muestras requeridas para el entrenamiento y el número de pasos hacia atrás, se convierten en el objetivo para determinar la red ideal.

Por cada entrenamiento se evalúa el RMSE (Error RMS) complementado con el MAXE

(Máximo Error: $MAXE = \max_{i=1}^m (|y_i - \tilde{y}_i|)$) con los datos de validación como lo sugiere

[Yu], para determinar cuál de los experimentos tiene el mejor desempeño y tomar así sus parámetros como óptimos.

El método descrito puede ser ampliado y mejorado en el futuro dado que la aplicación desarrollada permite experimentar libremente sobre nuevas teorías para obtener mejores resultados. El aporte computacional radica en la reducción del tiempo de ejecución necesario para el entrenamiento que requieren los algoritmos tradicionales en la obtención de los parámetros de la red neuronal (pesos y ajustes).

3.2 RESULTADOS POR MODELOS PROPUESTOS

3.2.1 Modelo por Regresión

Tomando como entradas las variables seleccionadas en la tabla 1.3 y entrenando por RAWN se obtuvieron los resultados que se muestran en la siguiente tabla:

# NEURONAS	RMSE	MAXE
1	0.0017439	0.0033676
3	0.0013600	0.0031367
4	0.0021229	0.0038839
5	0.0015836	0.0035062
6	0.0016774	0.007212
7	0.0019241	0.0061608
8	0.0017042	0.008346
10	0.0013473	0.0058143
20	0.0018437	0.011341
30	0.0020754	0.031088
40	0.0058415	0.12686

Tabla 3.1 Resultados de los Experimentos con el Modelo por Regresión

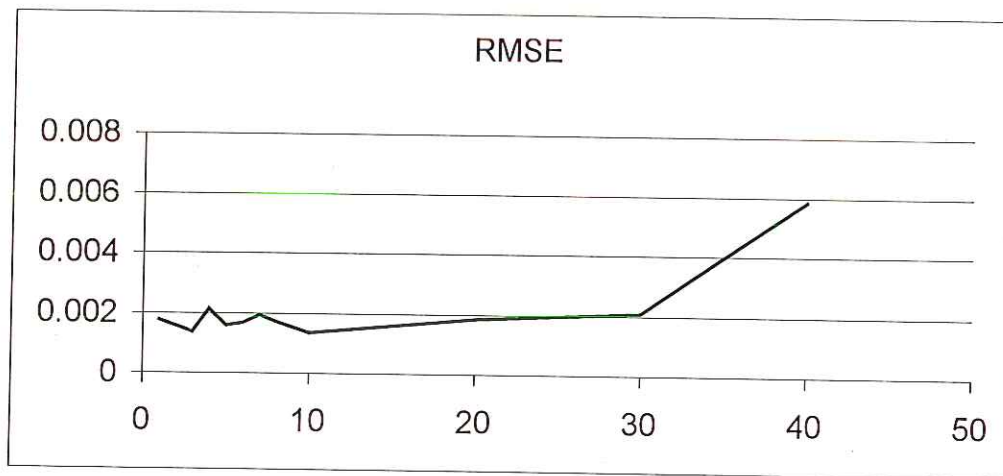


Figura 3.1 RMSE por Número de Neuronas con Regresión.

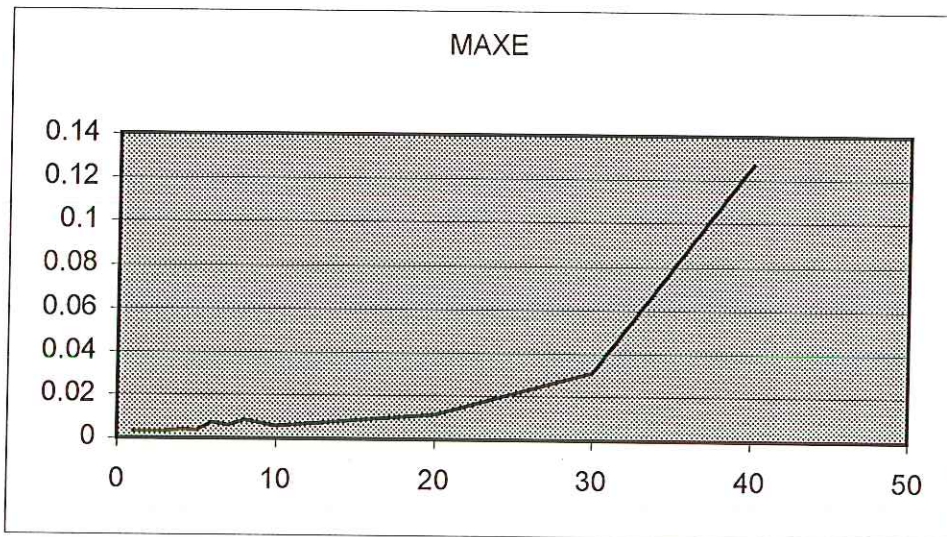


Figura 3.2 MAXE por Número de Neuronas con Regresión.

De las figuras anteriores se encuentra el caso óptimo en donde RMSE y MAXE tienen los picos más bajos simultáneamente es con 10 neuronas para la capa oculta.

3.2.2 Modelo por Correlación y Dispersión

Tomando como entradas las variables seleccionadas en la [tabla 1.4](#) y entrenando por RAWN se obtuvieron los resultados que se muestran en la siguiente tabla:

# NEURONAS	RMSE	MAXE
3	0.0010291	0.0027133
5	0.0011054	0.0024877
10	0.003029	0.012585
12	0.00096497	0.012741
15	0.0036674	0.0073696
20	0.00072544	0.0071936

Tabla 3.2 Resultados de los Experimentos con el Modelo por Correlación y Dispersión

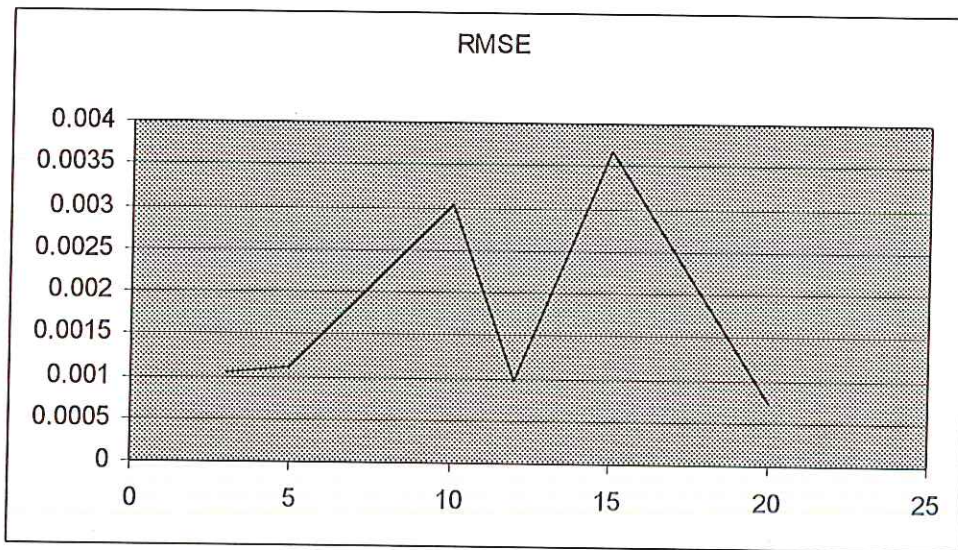


Figura 3.3 RMSE por Número de Neuronas con Correlación y Dispersión.

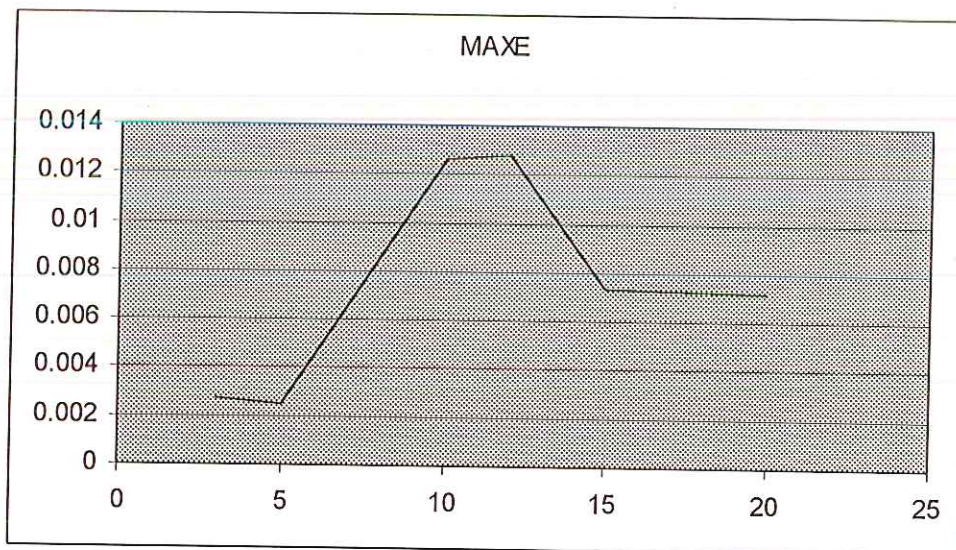


Figura 3.4 MAXE por Número de Neuronas con Correlación y Dispersión.

De las figuras anteriores se encuentra el caso óptimo en donde RMSE y MAXE tienen los picos más bajos simultáneamente es con 5 neuronas para la capa oculta.

3.2.3 Modelo por el Experto

Tomando como entradas las variables seleccionadas y disponibles en la [tabla 1.5](#) y entrenando por RAWN se obtuvieron los resultados que se muestran en la siguiente tabla:

# NEURONAS	RMSE	MAXE
5	0.0019238	0.003634
10	0.0017794	0.0037834
12	0.0016122	0.0035634
15	0.0021772	0.0071826
20	0.0035932	0.020569

Tabla 3.3 Resultados de los Experimentos con el Modelo por el Experto

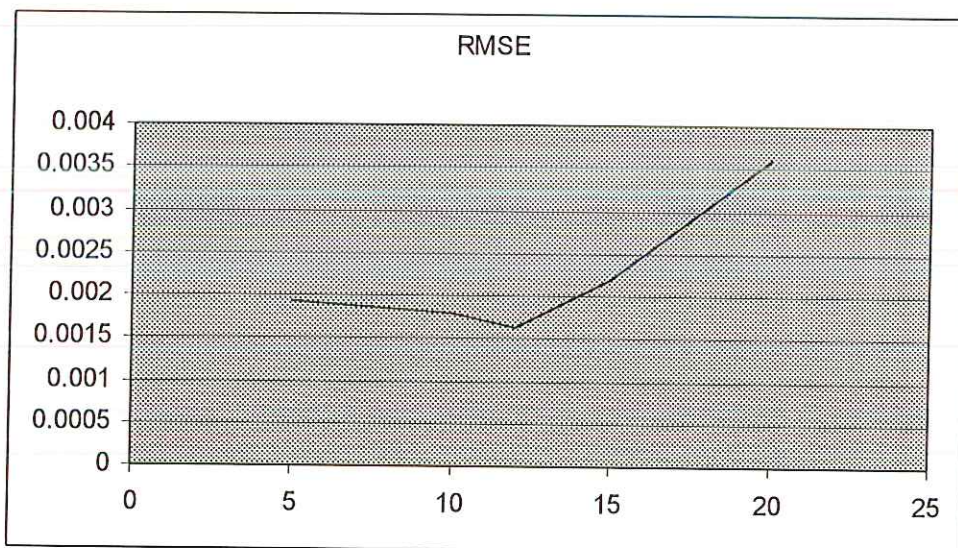


Figura 3.5 RMSE por Número de Neuronas por el Modelo del Experto.

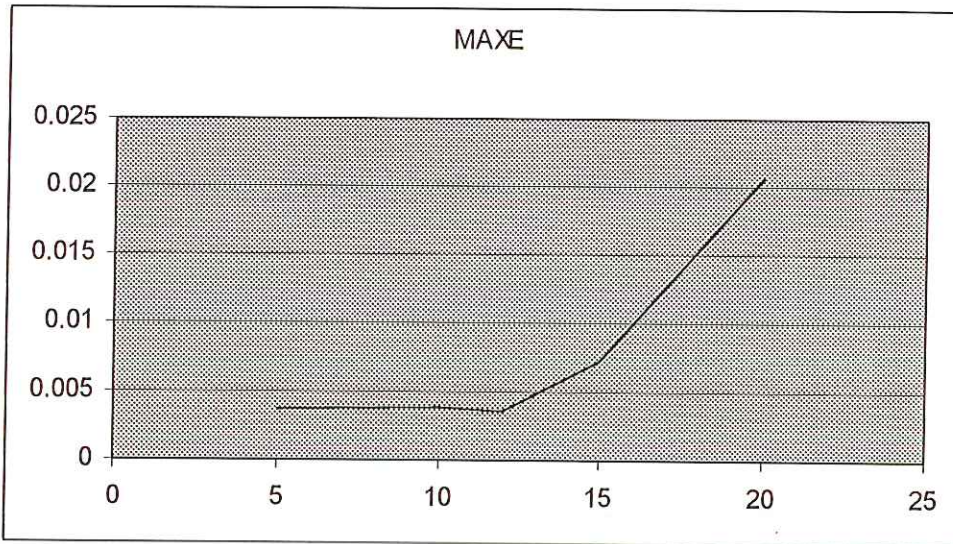


Figura 3.6 MAXE por Número de Neuronas por el modelo del Experto.

De las figuras anteriores se encuentra el caso óptimo en donde RMSE y MAXE tienen los picos más bajos simultáneamente es con **12 neuronas** para la capa oculta.

3.3 Optimización Final de Parámetros de la Red.

El mejor modelo fue entregado por la red alimentada con datos de entrada seleccionados por correlación y dispersión.

Sin embargo el modelo que mayor cantidad de neuronas en la capa oculta utiliza es el propuesto por el Experto, lo cual daría una mayor robustez con otros caso de prueba. Se esperaría que este último fuese el mejor pero la falta de variables sugeridas por el experto (faltan 3), variables fuera de rango e instrumentos en mal estado (v50: PRC7002 Presión de la Torre 753 A) de la planta, impiden que esto sea más evidente.

MODELO	# NEURONAS	RMSE	MAXE
REGRESIÓN	10	0.0013473	0.0058143
CORRELACION-DISPERSION	5	0.0011054	0.0024877
EXPERTO	12	0.0016122	0.0035634

Tabla 3.4 Comparación de los Diferentes Modelos.

MODELO	# NEURONAS	RMSE	MAXE
CORRELACION-DISPERSION	5	0.0010612	0.0034794

Tabla 3.5 RMSE y MAXE para entrenamiento en la zona que cubre las muestras de la 4000 a la 5000.

B1	B2
-0.0401	1.4030
-0.1741	
0.2608	
-0.0609	
0.2661	

Tabla 3.6 Bias para las Neuronas de la Capa Oculta (B1) y la Neurona de Salida (B2)

W1	V6	V7	V10	V14	V15	V17	V18	V32	V44	V63	
N#1	-0	6	4	3	4	12	-10	342	63	-1071	-1282
N#2	11	-6	-4	1	6	-6	-14	-457	-58	312	-529
N#3	9	-7	6	-1	0	-6	11	-65	12	1017	1248
N#4	4	-6	1	3	5	9	-8	-923	-126	117	797
N#5	9	-1	-2	2	-5	6	-6	1036	-51	549	171

Tabla 3.7 Pesos de Entrada (*1E-04)

W2	0.0330	-0.0532	0.0284	0.0168	0.0039
----	--------	---------	--------	--------	--------

Tabla 3.8 Pesos de Salida

3.4 ANALISIS DE RESULTADOS

La técnica utilizada para analizar el comportamiento de los diferentes casos es novedosa y sencilla. Combinada con el entrenamiento por RAWN permite poblar en pocas horas un buen número de pruebas para obtener muy buenos resultados para el sensor virtual.

La poca disponibilidad de datos buenos de la planta, limita en gran parte la obtención de mejores resultados.

Dentro de la información consignada en las tablas y figuras de este mismo capítulo se aprecia la evolución de estas pruebas y nos lleva a seleccionar el caso del modelo que utiliza entradas de variables obtenidas por correlación y dispersión, para el cual se muestra el comportamiento final del sensor virtual contra el real en la [figura 3.7](#).

Esta selección, a pesar de no ser la óptima bajo condiciones ideales de diseño, se constituye en la que mejores resultados ofrece dadas las limitaciones del proceso y los cambios que este ha sufrido a través de sus años de servicio.

El número de muestras necesarias para obtener buenos resultados no es función de la cantidad como si lo es de la calidad de los datos tanto de entrada como de salida y para ello además se deben escoger rangos de entrenamiento donde se cubra la mayor variabilidad siempre y cuando esta información verdadera para el proceso.

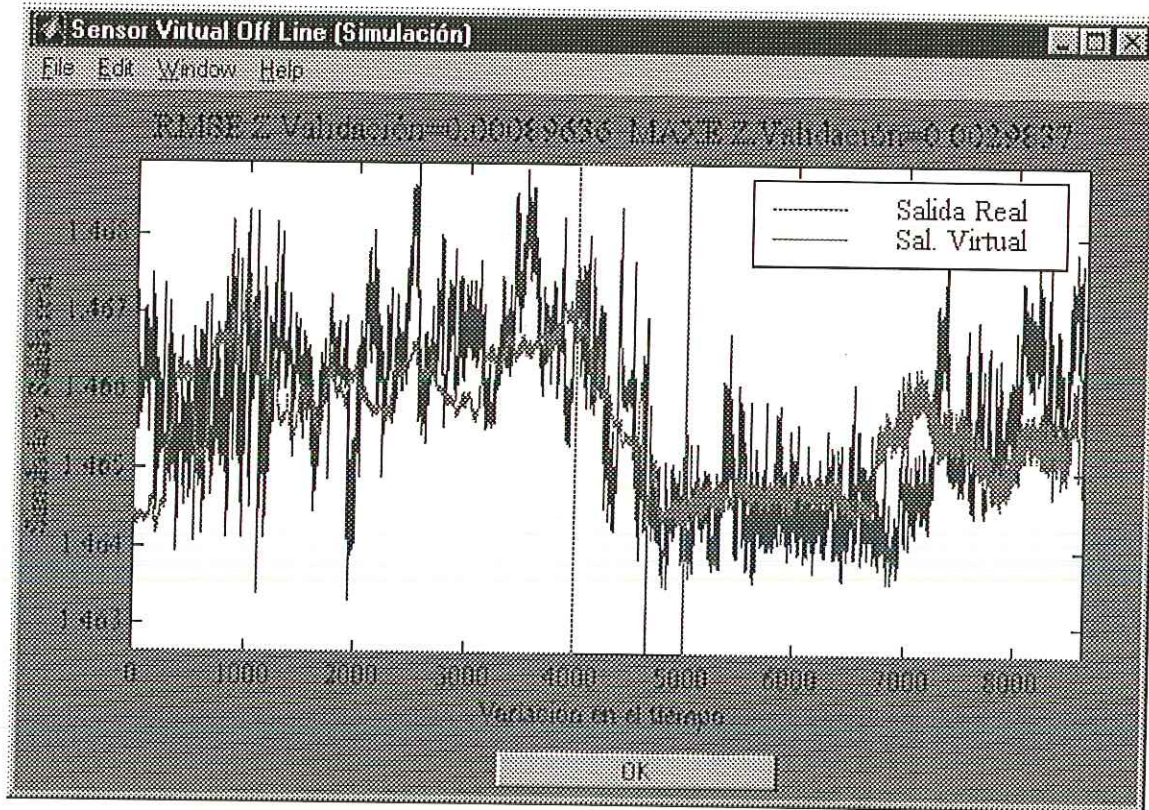


Figura 3.7 Comportamiento del Sensor Virtual Obtenido (Zona de Entrenamiento con 1000 muestras y Validación de las restantes)

4 CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

Las redes neuronales artificiales son capaces de manejar problemas complejos y no lineales, pueden procesar información muy rápidamente y reducen el esfuerzo computacional requerido en el desarrollo de modelos computacionales intensivos, encontrando formas funcionales para modelos empíricos como el de nuestro caso con el sensor virtual.

En una red neuronal artificial solo se necesitan datos de entrada y salida para que la red reconozca un patrón envuelto en el mapeo de las variables de entrada a la respuesta de la salida. Es verdad que las redes neuronales han sido descritas como una "caja negra" para solucionar problemas, pero la habilidad de la red neuronal para dar valores rápidos y precisos para el caso de los ingenieros de proceso hace de ellas una herramienta muy útil. Su habilidad para ejecutar fácilmente el problema inverso de intercambiar los vectores de entrada y la salida de la red, también se constituye en otra ventaja para el análisis y diagnóstico de un sistema dado.

Se puede obtener un excelente mapeo con redes neuronales en adelante con tan sólo una capa de neuronas no lineales tomando los pesos de activación aleatoriamente, seguido por un entrenamiento de los pesos de salida por mínimos cuadrados ordinarios. Ejemplos estáticos y dinámicos muestran la factibilidad de esta aproximación. Como en cualquier identificación no lineal, se debe tener cuidado para asegurarse de que la entrada de excitación usada para identificación, esté en el mismo rango de frecuencia y amplitud como en la aplicación. Posteriores mejoras se pueden obtener por regularización de los pesos de activación.

Habiendo observado las extraordinarias propiedades de la red RAW, podemos decir que la selección aleatoria de pesos puede quizás suponer la necesidad de un gran número de neuronas que son estrictamente requeridas cuando se ejecuta un entrenamiento completo.

A pesar de esto, el número de grados de libertad con una RAWN permanece más pequeño que con redes completas entrenadas por backpropagation.

La escogencia aleatoria de los pesos de activación no impide la solución de problemas lineales no separables. No se debe olvidar que el número de parámetros libres en una red en adelante de dos neuronas entrenada por backpropagation es cuatro, si no se utilizan entradas de ajuste (bias), de lo contrario serían seis, mientras que una red neuronal RAWN sólo tiene tres grados de libertad.

Una vez los pesos de activación se escojan, no se necesitan más iteraciones y siempre se encuentra un mínimo a diferencia del backpropagation que sufre de propiedades de convergencia serias. Un buen ejemplo de esto es el problema clásico de la "or exclusiva" (XOR), donde la salida es uno si cualquiera de las dos entradas es uno, pero no ambas. Este problema es no lineal separable. Si una red neuronal es entrenada con backpropagation, se necesitan dos neuronas para un mapeo apropiado de la XOR. Con una red neuronal RAWN se necesitan tres neuronas, pero este caso se obtiene un mejor mapeo.

Los resultados presentados aquí sugieren que la regularización puede aún incrementar el desempeño de RAWN. Ciertamente vale la pena investigar por esquemas, como el sugerido por **Drao** y **Ridella**. Resultados preliminares con una modificación de la regularización de entrada inteligente presentada aquí, sugieren que en algunos casos se pueden alcanzar mejoras adicionales.

El entrenamiento de la salida lineal lleva alguna similaridad con aproximación de función de base radial, pero en el caso de RAWN no se tienen que postular o estimar parámetros de función. El método de pesos de entrada aleatorios ha sido mencionado brevemente por **Pao** y otros, en conjunto con la aproximación de la red funcional encadenada. Sin embargo, los resultados reportados aquí indican que un buen desempeño se puede

obtener sin la manipulación de entrada funcional. Esto es, sólo se requiere un mínimo de información sobre el sistema en una aproximación RAWN, mientras se mantiene una estructura extremadamente simple.

Un problema de considerable interés práctico en aplicaciones dinámicas sin importar la clase de red neuronal o el método de entrenamiento, es determinar cuantos pasos hacia atrás de la salida y de las señales de entrada se necesitan para un buen mapeo. Esto es similar a la determinación del orden en el análisis de series de tiempo lineales. Para aplicaciones dinámicas no lineales se necesita desarrollar un criterio de balance similar al criterio en la teoría no lineal de Akaike. En la ausencia de una teoría apropiada, el entrenamiento rápido y no iterativo de RAWN nos permite recalcular rápidamente con varios órdenes y desplazamientos de entrada hacia atrás, facilitando esto la selección heurística de la estructura más apropiada.

Finalmente, se debe señalar que es bastante posible, y de hecho bastante fácil, desarrollar una versión recursiva del algoritmo y de este modo poder traer aplicaciones en línea incluso más exactas. Debido a que la aproximación RAWN con OLS no requiere iteraciones, el procedimiento es extremadamente rápido. Esto hace que el método sea particularmente adecuado para propósitos de control en línea.

Se obtuvo una herramienta para diseñar sensores virtuales, que utilizando la técnica de entrenamiento RAWN, consigue rápidamente los parámetros óptimos de la red, para posteriormente re-entrenarlo por métodos más complejos de afinamiento del modelo. Los parámetros de pesos de entrada y salida junto con los bias se pueden utilizar para implementar el sensor virtual bajo cualquier plataforma o lenguaje de programación conocida la ecuación de la red neuronal artificial y sus parámetros obtenidos.

Una desventaja de utilizar analizadores virtuales como el propuesto, es el requisito de mantener la instrumentación asociada a él en óptimas condiciones, lo cual en la práctica es una tarea que requiere mucho compromiso del personal de mantenimiento.

4.2 RECOMENDACIONES

Teniendo datos históricos o experimentales de una planta se puede poner en uso una red neuronal para su validación. La aplicación de las redes neuronales como herramienta de predicción y diagnóstico es de gran utilidad para los ingenieros de proceso y vale la pena seguir investigando aún más sobre ellas.

La simulación y el control es otro campo en donde las redes neuronales artificiales se pueden usar. Para la complejidad y no linealidad de los procesos, ellas pueden ofrecer un eficiente comportamiento que otras técnicas tradicionales de control no reúnen en cuanto a estabilidad y desempeño.

Trabajar con lenguajes de programación de alto nivel facilitan la implementación de aplicaciones completas, pero requieren de un alto recurso computacional. En el Anexo C se describen los requerimientos mínimos para trabajar el Matlab bajo plataforma Windows. Como recomendación final sugerimos trabajar bajo plataforma Unix, que realiza una administración más efectiva de estos recursos como ya ha sido probado a nivel de investigación en otras universidades.

Para la selección de variables se recomienda también utilizar el procedimiento de eliminación hacia atrás utilizando ANCOVA. Otro método indicado también sería el PCA (Análisis de Componentes Principales) y el ANOVA, cuando no se tenga información detallada del proceso estudiado.

5 BIBLIOGRAFIA

5.1 LIBROS Y ARTÍCULOS

BABUSCA, R. y Verbruggen, H. B., Applied Fuzzy Modeling. Applied Fuzzy Modeling, Proceedings of the IFAC Symposium on Artificial Intelligence in Real Time Control, 1994. 61-66p.

BABUSCA, R. y Verbruggen, H.B., Identification of composite linear models via fuzzy clustering, Proceedings of the ECC'95, 1995. 1207-1212p.

BALDI, P., Gradient descent learning algorithm overview: a general dynamical systems perspective, IEEE Transactions on Neural Networks, 1995. 6:182-195p.

BARHEN. J.. Toomarian. N. and Fijany, A., Learning without local minima, Proceedings of the IEEE International Conference on Neural Networks (ICNN), IEEE World Congress on Computational Intelligence, 1994. 7:4592-4596p.

BARNET, M. W., Adaptative Neural Networks for Intrelligent Operation of the Activaded Sludge Process, Gensym Corp., 2000. 9p.

BASHEER, A. I. And Najjar, Y.M., Predicting Dynamic Response of Adsorption Columns with Neural Nets, Journal of Computing in Civil Engineering, 1996. 10(1): 31-39p.

BARRON, A. R., Universal approximation bounds for superpositions of sigmoidal function. IEEE Transactions on Information Theory , 1993. 39, 930-945p.

- BARTO, R. S. Sutton, and C. J. C. H. Watkins, Learning and sequential decision making. Technical report, COINS. 1989.
- BATES, J. M. and C.W. J. Granger, The combination of forecast, Operations Research Quarterly, 1969. 20:319-325p.
- BATTIKHA, N.E., The Condensed Handbook of Measurement and Control. Instrument Society of America. 1997.
- BATTITI, R., Usual mutual information for selecting features in supervised neural network learning. IEEE Transactions on Neural Networks. 1995.
- BHAT, N. and McAvoy, T.J., Use of Neural Nets for Dynamic Modeling and Control of Chemical Process Systems, Computers Chem.Engng., 1990. 14(4/5): 573-583p.
- BILLIGS S. A., Jamaluddin H. B. and Chen S. A comparison of the backpropagation and recursive prediction error algorithms for training neural networks. Mech. Syst. Signal Process. 1991. 5, 233-255p.
- BRAAKE, H. A. B. and van Straten, G. Random Activation Weight neural net (RAWN) for fast non-iterative training. Engineering Applications of Artificial Intelligence, 1995. 8, 71-80p.
- BRAAKE, H. A. B., van Can, H. J. L., van Straten, G. and Verbruggen, H. B.. Regulated Activation Weights Neural Network (RAWN). Proceedings of the European Symposium on Artificial Neural Networks, ESANN'96, Brugge, 1996. 19-24p.
- BRAJESH, K.J., Tambe, S.S. and Kulkarni, B.D., Estimating Diffusion Coefficients of a Micellar System Using an Artificial Neural Network, J. Colloid & Interface Science, 1995. 170:392-398p.
- CHAO, L.L., Estadística para las Ciencias Administrativas. McGraw-Hill. 1975.

- CHERRE, D., Use of Artificial Neural Networks in Process Control, Master of Science Thesis, University of South Florida, Tampa. 1998.
- CHEN S., Cowan C. F. N., Billings S. S and Grant P. M. Parallel recursive prediction error algorithm for training layered neural networks. *Int. J. Control.* 1990. 51, 1215-1228p.
- CORRIPIO, A., Smith, C., Control Automatico de Procesos. Limusa. 1996.
- CYBENKO G. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst.* 1989.2, 303-314p.
- DABNEY, James. *Mastering Simulink 2*, Upper Saddle River, NJ: Prentice Hall, 1998. 345p.
- DALLAS, E. J. *Métodos Multivariados Aplicados al Analisis de Datos*. Mexico, 1998. 566p.
- DELGADO, J.A., *Elementos de Redes Neuronales y Algoritmos Genéticos*, Neuronales Ltda. 1992.
- DAVE, R. N. Boundary Detection through Fuzzy Clustering. *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, San Diego, 1992. 127-134p.
- DAYHOFF, J. *Neural Networks Architectures: An Introduction*, New York, NY: Van Nostrand Reinhold, 1990. 320p.
- DIEDERICH, J. *Artificial Neural Networks: Concept Learning*, Los Alamitos, CA: Computer Society Press, 1990. 160p
- DRAGO G. P. and Ridella S. Statistically controlled activation weight initialization (SCAWI). *IEEE Trans. Neural Networks* 1992. 3, 627-631p
- FORTUNA, L., Graciani, S., Lo Presti, M. and Muscato, G. Improving backpropagation learning using auxiliary neural networks. *International, Journal of Control*, 1992. 55, 793-907p.
- FUNAHASHI, K.I., On the approximate realization of continuous mapping by neural networks. *Neural Networks*, 1989. 2, 183-192p.

GOLUB, G. H. and van Loan, C. F. An Analysis of the Total Least Squares Problem. SIAM Journal on Numerical Analysis, 1980. 17, 883-893p.

HAGAN, M.T. and Menhaj, M., Training Feedforward Networks with the Marquadt Algorithm, IEEE Transactions on Neural Networks, 1994. 5(6):989-993p.

HANSELMAN, D., Littlefield, B., Mastering Matlab 5. Prentice Hall. 1998.

HASSOUN, M. H. Fundamentals of Artificial Neural Networks. The Mit Press, Cambridge, MA. 1995.

HAYKIN, S., Neural Networks. A Comprehensive Foundation, Macmillan College Publishing Company, N.Y. 1994

HILERA, José R. Redes Neuronales Artificiales: Fundamentos, Modelos y Aplicaciones, Madrid: Addison-Wesley Iberoamericana, 1995. 390 p.

HONEYWELL Hi-Spec Solutions, Lube Oil Extraction, Hydrocarbon Processing, Phoenix. 1999. 99:125p.

HUNT, K. J. Sbarbaro, D., Zbikowki, R and Gawthrop, P. J., Neural Network for control systems a survey. Automatica, 1992. 28:1083-1112p.

IGOR, Aleksander. Neural Computing Architectures, Cambridge, MA: MIT Press, 1989. 240p.

JANG, J. S. Neuro-Fuzzy and Soft Computing. Upper Saddle River, NJ: Prentice Hall, 1997. 614p.

JOHNSON, D.E., Metodos Multivariados Aplicados al Analisis de Datos. Thomson Editores. 2000.

K-PATENTS, Inc. Instruction Manual for PR-01-S (-EX/FM) Version 6.0. 1996.

KOSKO, B. Neural networks and fuzzy system; a dynamical systems approach to machine intelligence. Prentice Hall, Englewood Cliffs, Nj. 1992.

KREYSZIG E. Introductory Mathematical Statistics; Principles and Methods. Wiley, New York .1970.

LEVIN, Mijail. Automatización de Procesos Piro e Hidrometalúrgicos, Saint Petersburgo, Rusia: Editorial Ciencia y Técnica, 1989.

LIN, Chin-Teng. Neural Fuzzy Systems, Upper Saddle River, NJ: Prentice Hall, 1995. 797p.

LITTLEFIELD, Bruce. Mastering Matlab 5, Upper Saddle River, NJ: Prentice Hall, 1998. 639p.

MARE, Harston. Handbook of Neural Computing Applications, San Diego, CA: Academic Press, 1990. 245p.

MASTERS, Timothy. Practical Neural Network Recipes in C++. San Diego, CA: Academic Press. 1993. 493p.

MINSKY, M. L. Perceptrons (Expanded Edtition), Cambridge, MA: MIT Press, 1988. 230p.

MYERS, R. H., Classical and Modern Regression with Applications, Duxbury Press. 1986.

OGATA, K., Ingeniería de Control Moderna. Prentice Hall. 1980.

ORDOÑEZ, O., Aplicación de una metodología estadística para estimar beneficios de implantar sistema de control avanzado a la unidad de extracción de bases lubricantes con fenol en el Complejo Industrial de Barrancabermeja ECOPETROL. Universidad Insustrial de Santander. Escuela de Ingeniería Química. Bucaramanga. 1995. 252 p.

- OTERO, F., Use of Neural Networks in Process Engineering. CT&F, 1998. 49-64p.
- PAO Y-H., Phillips S. M. and Sobajic D. J. Neural-net computing and the intelligent control of systems. Int. J. Control . 1992. 56, 263-289p
- PEREZ, R., Artificial Neural Networks, Computer Science and Eng. Dept., University of South Florida. 1998.
- QUINTERO, A., Sistemas Unitarios de Control. Universidad Pontificia Bolivariana, Facultad de Ingenieria Electrica y Electrónica, Medellín. 1993.
- RAMESH, K. Tock, R. W. and Narayan, R.S., Prediction of solvent Activity in Polymer System with Neural Networks, Ind. Eng. Chem. Res., 1995. 34:3974-3980p.
- SHINSKEY, F.G., Process Control System. McGraw-Hill. 1988.
- SIMPSON, Patrick K. Artificial Neural Systems, New York, NY: Pergamon Press, 1990. 162p.
- SINGHAL S. and Wu L. Training feed-forward networks with the extended Dalman algorithm. IEEE Proc. 1989. 1187-1190p.
- SCALERO R. S. and Tependelenlioglu N. A fast new algorithm for training feedforward neural networks. IEEE Trans. Signal Process. 1992. 40, 202-210p.
- SOVKOVIC-Stevanovic,J., Neural Network for Process Analysis and Optimization: Modeling and Applications, Computers Chem. Engng., 1994. 18(11/12):1149-1155p.
- STEPHEN, J. Neural Network Design and The Complexity of Learning, Cambridge, MA, 1990. 305p.
- SWINGLER, Kevin. Applying Neural Networks: A practical Guide, San Diego, CA: Academic Press, 1996. 303p.

- TE BRAAKE, H.A.B., Van Can, H.J.L., Van Straten, G. and Verbrugger, H.B., Two-step Approach in Training of Regulated Activation Weight Networks (RAWN), Engng. Applic.Artif.Intell., 1997. 10(2):157-170p.
- TE BRAAKE, H.A.B., Van Straten, G. Random Activation Weight Neural Net (RAWN) for Fast Non-Interactive Training, Engng. Applic. Artif. Intell. 1995. 8(1):71-80p.
- THAM, M.T., A. Montague, et al., Soft-sensors for process estimation and inferential control, J. Proc. Cont., 1991. 1:3-14p.
- THOMPSON, W., How Neural Network Modeling Methods Complement Those of Physical Modeling, NPRA Computer Conference. 1996.
- VALENCIA, J.H., Sistemas Automáticos de Control, Universidad Pontificia Bolivariana, Facultad de Ingeniería Eléctrica y Electrónica, Medellín. 1991.
- VAN CAN, H. J. L., Braake Te, H.A.B., Hellings, C., Krijgsman, A, J., Verbruggen, H. B., Luyben, K, Ch. A. M. and Heijnen, J.J., Design and real time testing of a neural model predictive controller for a nonlinear system. Chemical Engineering Science , 1994. 50,2419-2430p.
- VEMURI, V. Artificial Neural Networks: Theoretical Concepts, Los Alamitos, CA: Computer Society Press, 1990. 125p.
- VILLALOBOS, E., Aplicación de Redes Neuronales al Desarrollo de Sensores Virtuales, Postgrado en Control de Procesos, Universidad del Zulia. 1999.
- WASSERMAN, P.D. Neural Computing, New York, NY: Van Nostrand-Reinhold, 1989. 202p.
- YU, J., Zhong, W., Improve nonlinear soft sensing modeling by combining multiple models, Hydrocarbon Processing, 2000. 108-112p.

5.2 DIRECCIONES DE INTERNET

5.2.1 Matemáticas

Procesos Estocásticos

<http://www.itlp.edu.mx/publica/tutoriales/investoper2/tema44.htm>

5.2.2 Sensórica

A Real-Time Neural Network-Based Intelligent Performance and Emissions Prediction System for on-board Diagnostics and Engine Control.

<http://www.cemr.wvu.edu/~virtsens/>

Example of flexible regression methods applied to the development of software sensors

<http://www.hds.utc.fr/WEB/heudiasyc/anglais/themes/diaq/bas.html>

Virtual Sensors

<http://www.chaotic.com/chaos/virtual.html>

Journal sobre Sensores

<http://www.mcb.co.uk/sr.htm>

MCB University Press. Centro Para Tecnología de Sensores

<http://www.ee.upenn.edu/center.cst.html>

<http://pender.ee.upenn.edu/~jan/>

Viscosidad

<http://www.kimble.com/html/Viscosity-254.html>

Viscosímetros de Laboratorio

Saybolt Viscosimeter Bath

http://www.geneq.com/catalog/en/asphalt/saybolt_viscosimeter_bath.html

Viscosímetros en Línea

<http://www.iqe.ethz.ch/pel/annrep95/p1.15.html>

5.2.3 Redes Neuronales

GENSYM CORP.

<http://www.gensym.com>

NEURODIMENSION INC.

<http://www.nd.com/>

Z-SOLUTIONS.

Neural Networks: Advanced Technology That Can Change The Way You Work

<http://www.zsolutions.com/index.htm>

An Introduction to Neural Networks

<http://www.cs.stir.ac.uk/~lss/NNIntro/InvSlides.html>

Introducción a redes neuronales

<http://www.cs.stir.ac.uk/~lss/NNIntro/InvSlides.html>

Elementos de una red neuronal

<http://www.ka.net/sac/neural.html>

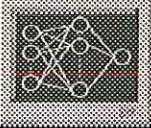
Publicaciones de Inteligencia Artificial del Departamento de Ciencias Experimentales e Ingeniería de la Universidad Rey Juan Carlos.

<http://www.escet.urjc.es/~ia/grupo/publicaciones.html>

Neural network based virtual sensor

<http://www.cemr.wvu.edu/~virtsens/>

ANEXO A

	MAESTRÍA EN CIENCIAS COMPUTACIONALES UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA	SVRN-01
	SENSOR VIRTUAL REDES NEURONALES Manual de Usuario	Pag 1 de 25

MAESTRÍA EN CIENCIAS COMPUTACIONALES

*Universidad Autónoma de Bucaramanga
Colombia*

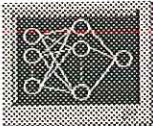
SENSOR VIRTUAL REDES NEURONALES

MANUAL USUARIO
Ver. 1.0

Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

Contenido

Contenido.....	2
INTRODUCCIÓN.....	3
1 GENERALIDADES.....	4
1.1 Descripción de la ventana.....	4
1.2 Entrada a la Aplicación	5
1.3 Configuración de la aplicación.....	6
1.4 Cierre de la aplicación.....	7
2 ARCHIVO DE DATOS.....	8
2.1 Carga de datos para entrenar.....	8
2.2 Guardar un proyecto	10
2.3 Carga de un proyecto.....	11
3 SELECCIÓN Y TRATAMIENTO DE VARIABLES.....	12
3.1 Análisis por Regresión Lineal.....	12
3.2 Análisis por Coeficientes de Correlación.....	13
3.3 Selección de Variables de Entrada	14
3.4 Tratamiento de Variables	15
4 ENTRENAMIENTO Y VALIDACIÓN.....	17
4.1 Entrenamiento por Regresión Lineal	17
4.2 Entrenamiento por Pesos de Activación Aleatoria.....	18
4.3 Entrenamiento por el Método de Proyección de Variables	19
5 PRUEBA DEL SENSOR.....	20
5.1 Salida Real y Virtual fuera de línea.....	20
5.2 Salida Real y Virtual en Línea.....	21
5.3 Monitoreo de Variables	22
6 AYUDA.....	23
6.1 Manual en Línea.....	23
6.2 Acerca de	23
6.3 Nombre de Variables	24
Archivo de Arranque del Sistema.....	25

	MAESTRÍA EN CIENCIAS COMPUTACIONALES UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA	SVRN-01
	SENSOR VIRTUAL REDES NEURONALES Manual de Usuario	Pag 3 de 25

INTRODUCCIÓN

Este manual tiene por objeto describir los diferentes procesos que se pueden realizar a través de la aplicación, detallando los pasos a seguir en cada uno de ellos, convirtiéndose en una sencilla guía para el manejo de la misma.

Para mejor manejo, ha sido distribuido en seis capítulos como son:

Generalidades: Describe la forma de entrar, configurar y salir de la aplicación.

Archivo de datos: dentro de este capítulo se describe la forma de cargar y guardar datos en la aplicación.

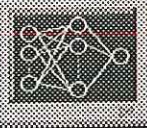
Selección y Tratamiento de Variables. Permite seleccionar las entradas que más impactan al proceso.

Entrenamiento y Validación. Realiza los cálculos necesarios para obtener la Salida Virtual del proceso utilizando las entradas seleccionadas anteriormente.

Prueba del Sensor. Permite visualizar la Salida Virtual con base en el entrenamiento realizado.

Ayuda. Permite orientar al usuario sobre el manejo del sensor y la interpretación de los diferentes resultados.

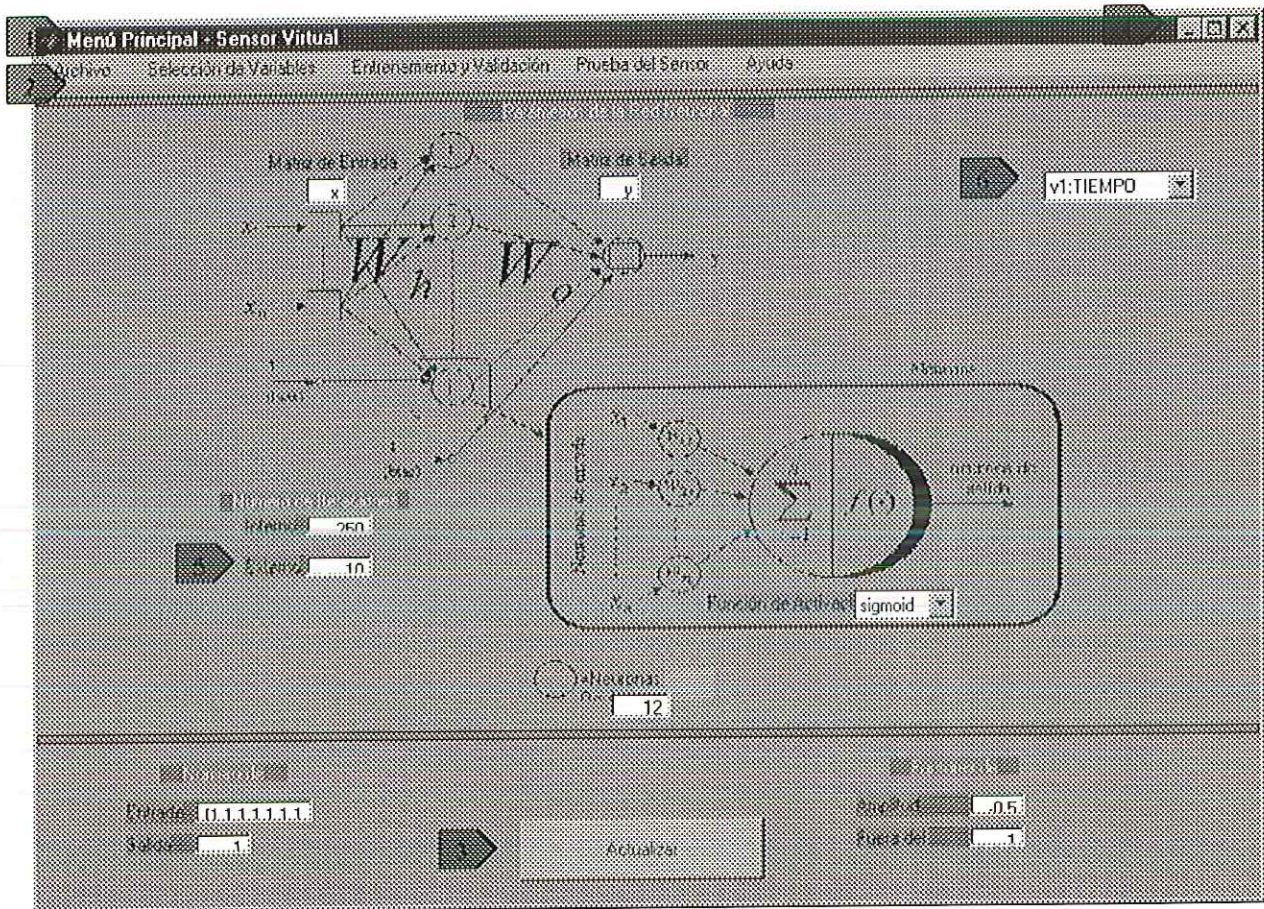
Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000



1 GENERALIDADES

1.1 Descripción de la ventana

Las ventanas que forman parte del módulo han sido creadas de acuerdo a los estándares seguidos por Windows, es decir, poseen la barra de título, barra de herramientas, caja de control y los botones para modificar el tamaño de la misma.



- 1 **Barra de Títulos:** Nombre que identifica la ventana.
- 2 **Barra de Menús:** Contiene los nombres de los menús o comandos a ejecutar.
- 3 **Botones de Función:** Permiten ejecutar rápidamente la función determinada.
- 4 **Botones Ventana:** Permiten Restaurar, Minimizar y Cerrar la ventana activa.
- 5 **Cajas de Dialogo:** Permite digitar información.
- 6 **Cajas de Selección:** Permite seleccionar información.

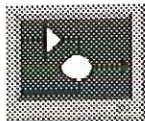
Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

1.2 Entrada a la Aplicación

Para acceder a la aplicación **SENSOR VIRTUAL** ubíquese en:

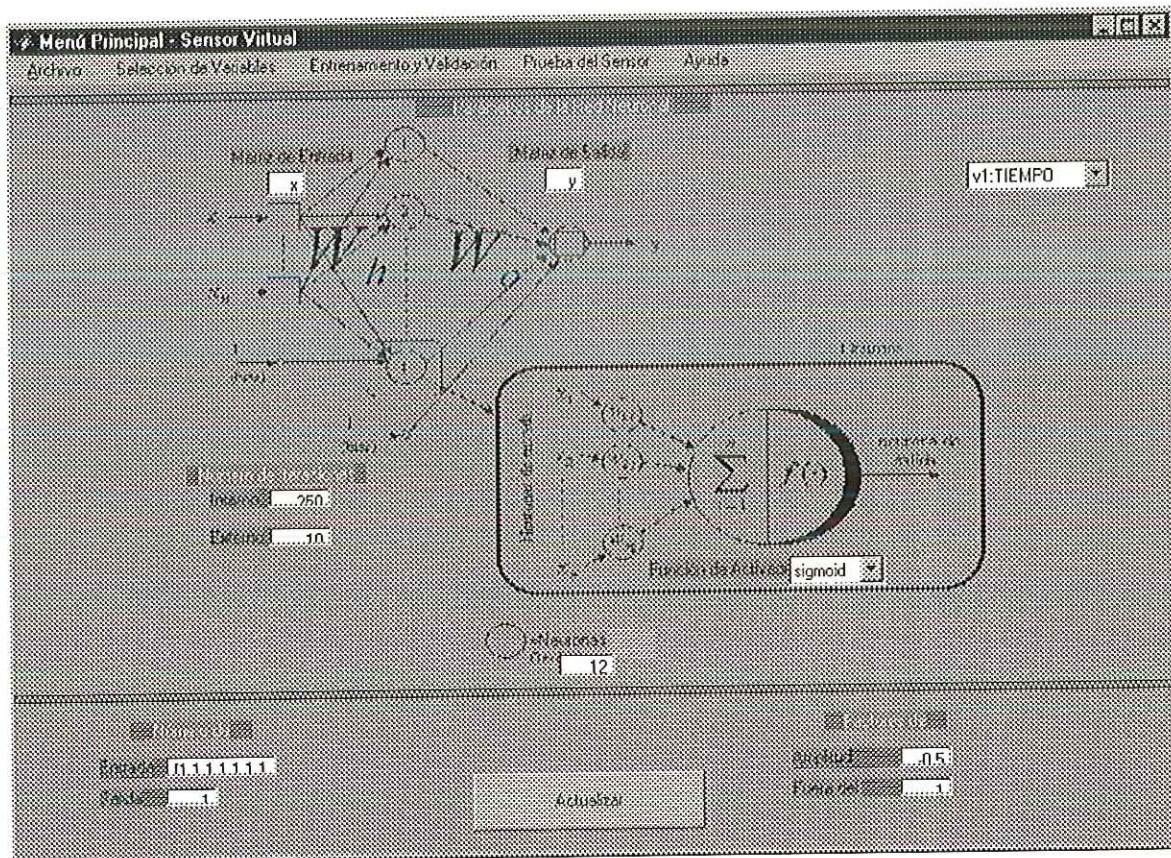
Inicio → **Programas** → **Sensor Virtual**.

O utilice el icono que aparece en su escritorio



Sensor Virtual

La aplicación se compone de los siguientes procesos:



Archivo: Permite cargar / guardar datos en la aplicación.

Selección y Tratamiento de Variables. Selecciona las entradas base para el sensor.

Entrenamiento y Validación. Obtiene los pesos y las bias, componente principal para el calculo de la Salida Virtual del proceso con base en las entradas seleccionas previamente

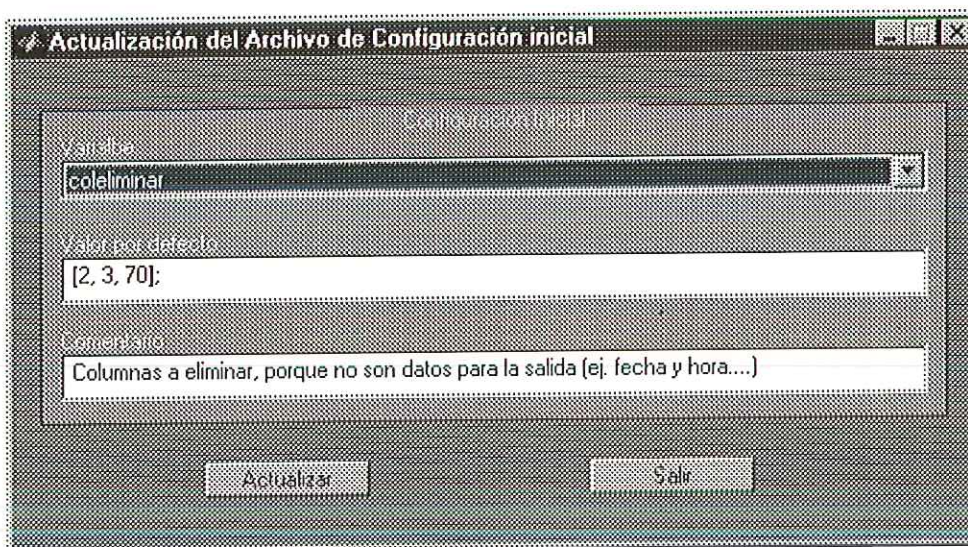
Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

Prueba del Sensor. Muestra el comportamiento de la variables cargadas en la aplicación

1.3 Configuración de la aplicación

1.3.1 Arranque del Sistema

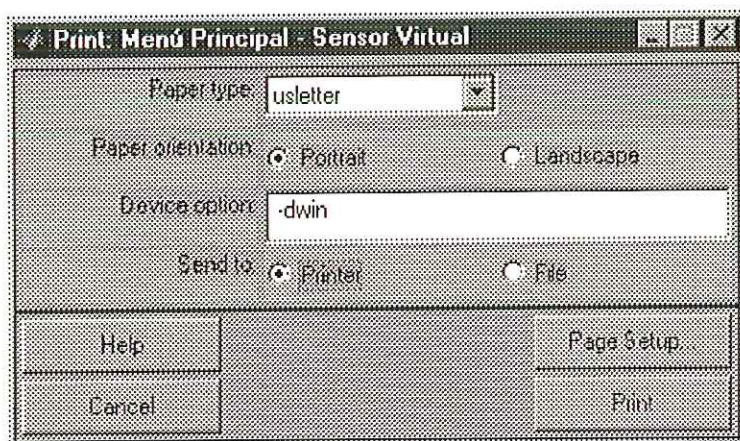
seleccione del menú Archivo → Configurar → Arranque del Sistema



Esta ventana permite definir entre otras: la columna donde viene la salida a inferir, las columnas que no se deben tener en cuenta en el proceso, si la interfaz en con Microsoft Excel en inglés o español y la línea de los títulos de las variables del proceso. **(Ver Anexo 1.):**

1.3.2 Impresora

seleccione del menú Archivo → Configurar → Impresora

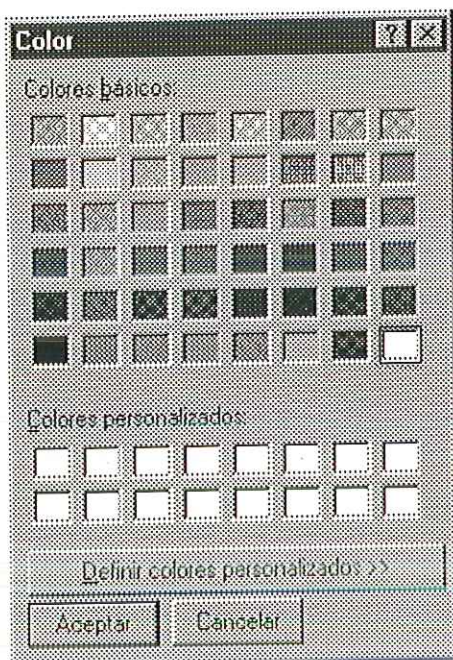


Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

Permite la configuración básica de la impresora

1.3.3 Color Fondo de Pantalla

seleccione del menú **Archivo** → **Configurar** → **Color Fondo de Pantalla**



Permite la configuración del color del fondo de la pantalla.

1.4 Cierre de la aplicación

Para salir del Sensor Virtual ubíquese en la ventana principal y con cualquiera de las siguientes opciones, podrá abandonar la aplicación

seleccione del menú **Archivo** → **Salir**.

En los botones de ventana **X**: **Cerrar**.

Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

2 ARCHIVO DE DATOS

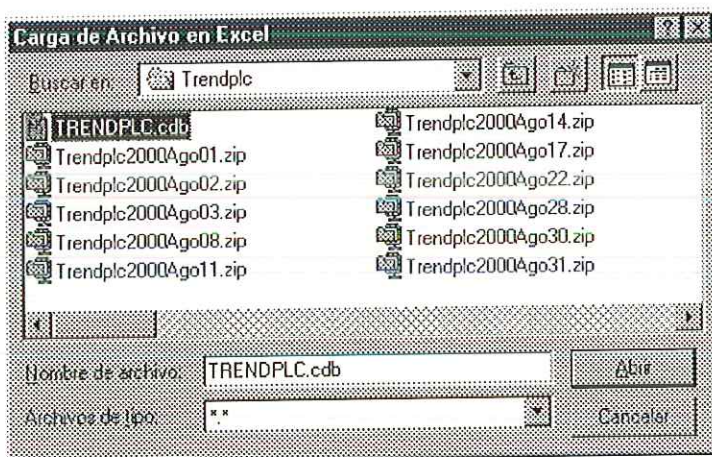
Aquí se muestra la forma de cargar datos originales del proceso o un proyecto ya entrenado por el Sensor Virtual, al igual que la forma de guardarlo.

2.1 Carga de datos para entrenar

2.1.1 De Microsoft Excel

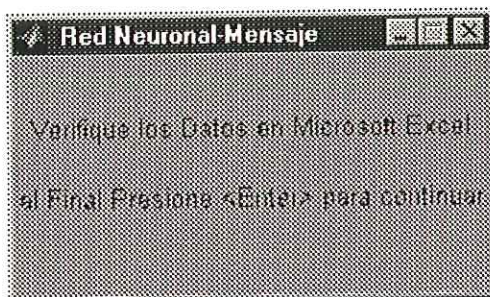
La siguiente opción permite cargar datos al proceso de cualquier tipo de archivo que pueda ser editado en Microsoft Excel, de la siguiente forma:

seleccione del menú **Archivo → Cargar Datos → de Microsoft Excel (.xls)**



Aquí se selecciona el archivo que contiene la información del proceso a entrenar.

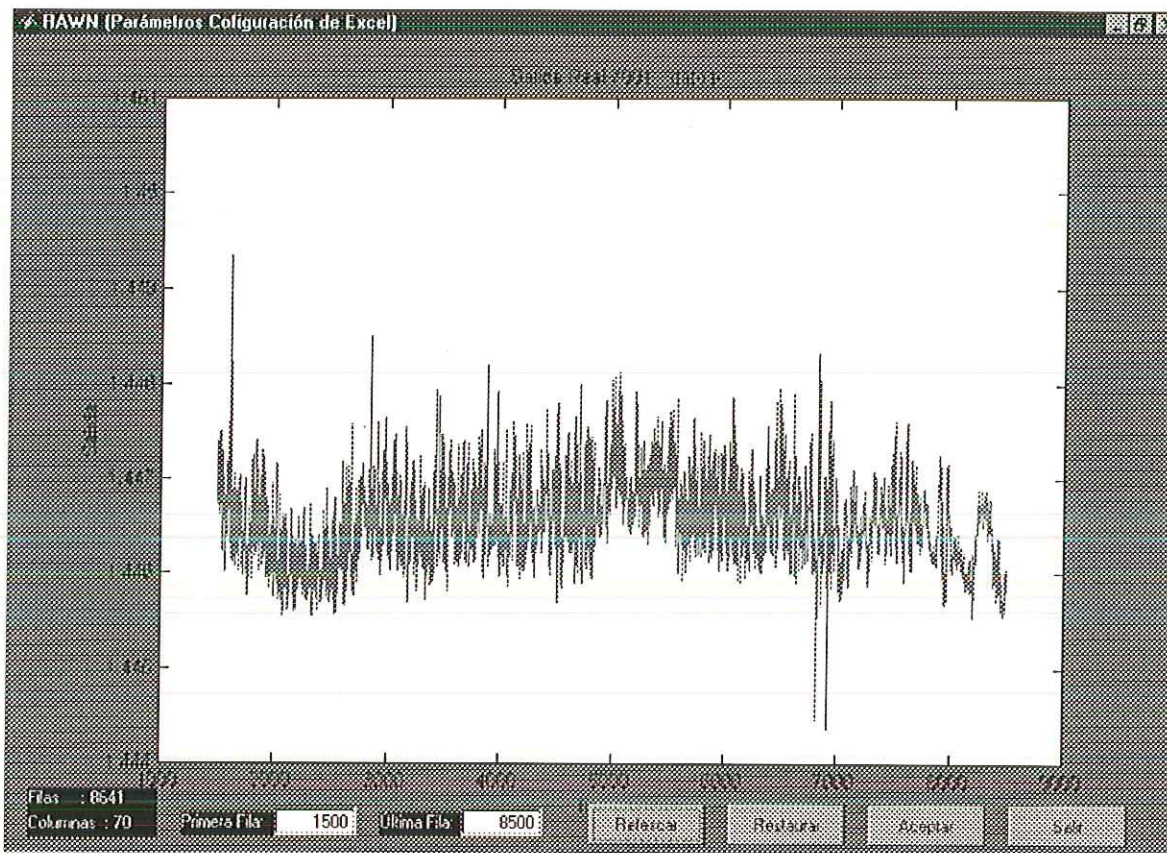
A continuación se abrirá Microsoft Excel con el archivo especificado y aparecerá en la aplicación el siguiente mensaje:



Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

Esto indica que valide si el archivo que abrió contiene los datos para el entrenamiento y para continuar deber pulsar ENTER ↵ en éste mensaje.

De acuerdo a la información del archivo de configuración, la aplicación graficará la salida indicada como se muestra en la siguiente pantalla:



Descripción de los Objetos en la pantalla

Las cajas de dialogo que aparecen en esta pantalla cumplen el siguiente objetivo:

Primera Fila: permite indicar el punto inicial para graficar y cargar los datos del archivo que está actualmente abierto en Microsoft Excel.

Última Fila: permite indicar el punto final para graficar y cargar los datos del archivo que está actualmente abierto en Microsoft Excel.

Los botones que aparecen en esta pantalla cumplen el siguiente objetivo:

Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

Refrescar: Permite graficar la salida con los datos que se encuentren dentro del rango especificado en **Primera Fila y Última Fila**

Restaurar: Permite graficar nuevamente la salida original, con todos los datos que tiene en el archivo especificado

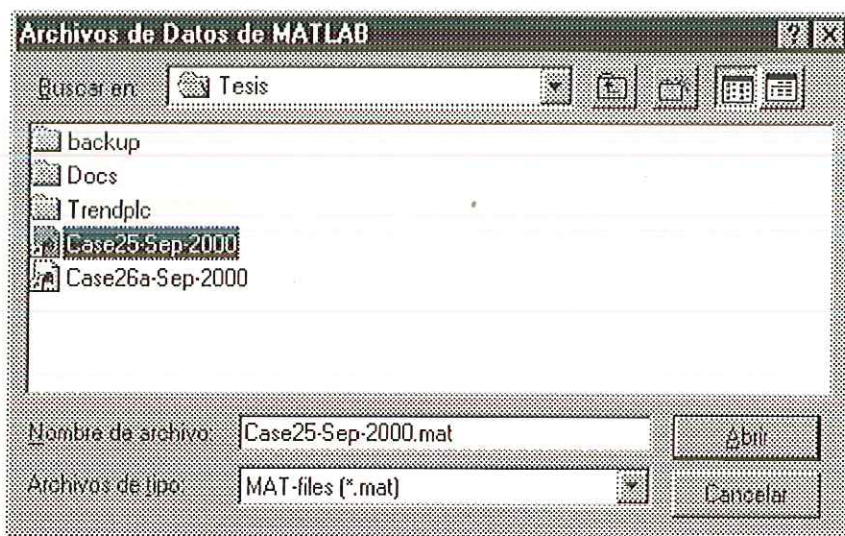
Aceptar: Carga toda la información de acuerdo al rango especificado en **Primera Fila y Última Fila**, eliminando aquellas variables constantes y las especificadas en el archivo de configuración como columnas a eliminar.

Salir: Cierra la ventana sin cargar información a la aplicación.

2.1.2 De Matlab

Si los datos se encuentran en un archivo de Matlab, lo puede cargar de la siguiente forma:

seleccione del menú **Archivo** → **Cargar Datos** → **de Matlab (.mat)**



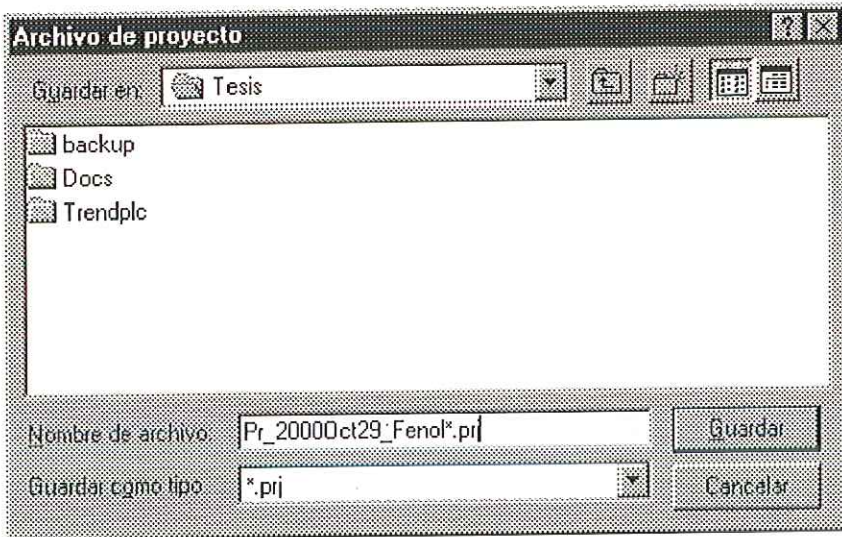
Con solo oprimir el botón abrir la aplicación carga los datos especificados.

2.2 Guardar un proyecto

En ocasiones es necesario guardar la información de un proyecto ya entrenado, sobre el cual se pudo obtener resultados importantes, esta opción del menú permite salvar los datos mas relevantes de ese entrenamiento.

seleccione del menú **Archivo** → **Guardar Proyecto**

Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000



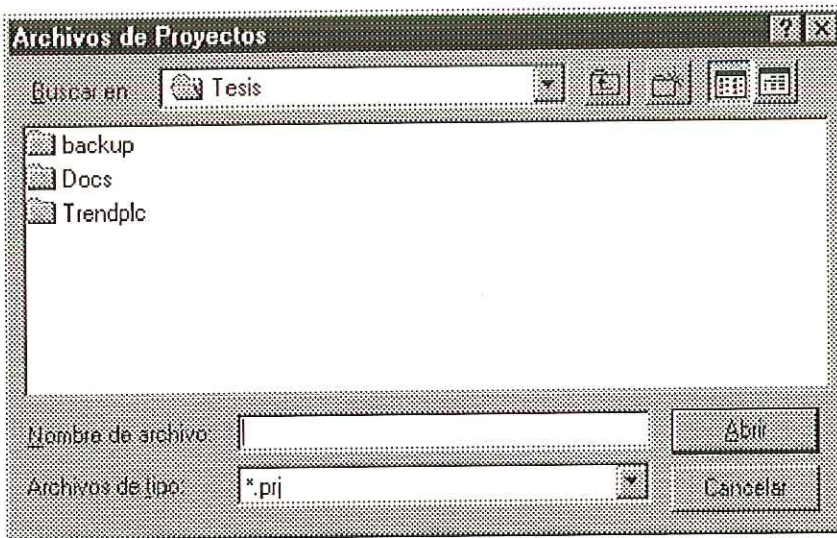
Digitar el nombre del .prj y oprimir el botón **Guardar**

2.3 Carga de un proyecto

La siguiente opción permite cargar datos que han sido salvados como un proyecto:

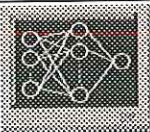
La información aquí recuperada está conformada por: pesos, bias, tratamiento a las variables y demás datos correspondientes a un entrenamiento previamente realizado.

Para utilizarla seleccione del menú **Archivo** → **Cargar Proyecto Entrenado**



Aquí debe seleccionar un archivo .prj y oprimir el botón **abrir**

Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000



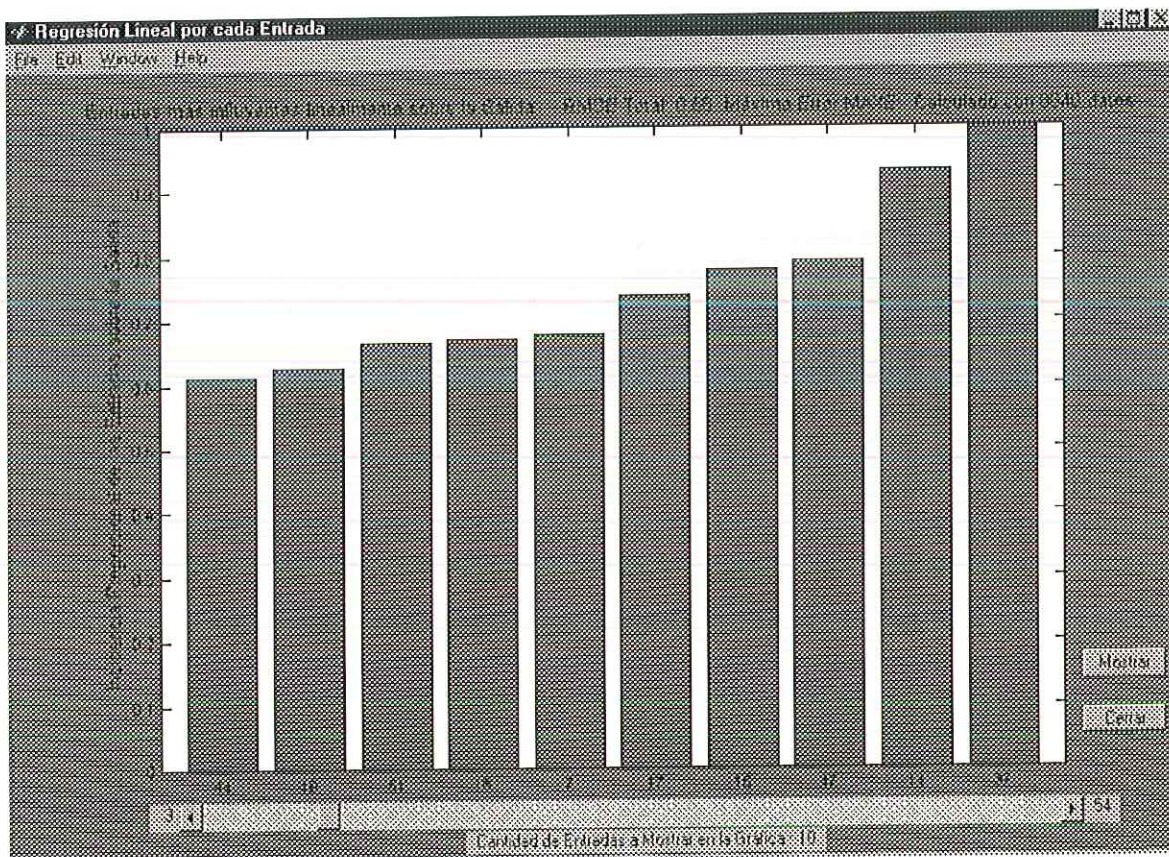
3 SELECCIÓN Y TRATAMIENTO DE VARIABLES

La aplicación está provista de dos herramientas estadísticas, que apoyarán la selección matemática de las variables que más influyen en la salida, al igual que permite seleccionarlas y realizarles un tratamiento básico de filtraje y corte a las mismas.

3.1 Análisis por Regresión Lineal

seleccione del menú **Selección de Variables** → **Análisis por Regresión Lineal**

Éste cálculo toma su tiempo dependiendo del número de variables y de la cantidad de datos que posea cada variable, al final suministra una gráfica de barras ordenada de acuerdo a la influencia lineal que cada entrada tiene sobre la salida; así:



La gráfica está provista de una barra de desplazamiento (slider) que permite visualizar el comportamiento de la cantidad de entradas que se desee hasta su máximo; solo debe desplazarse sobre la barra y oprimir **Mostar**

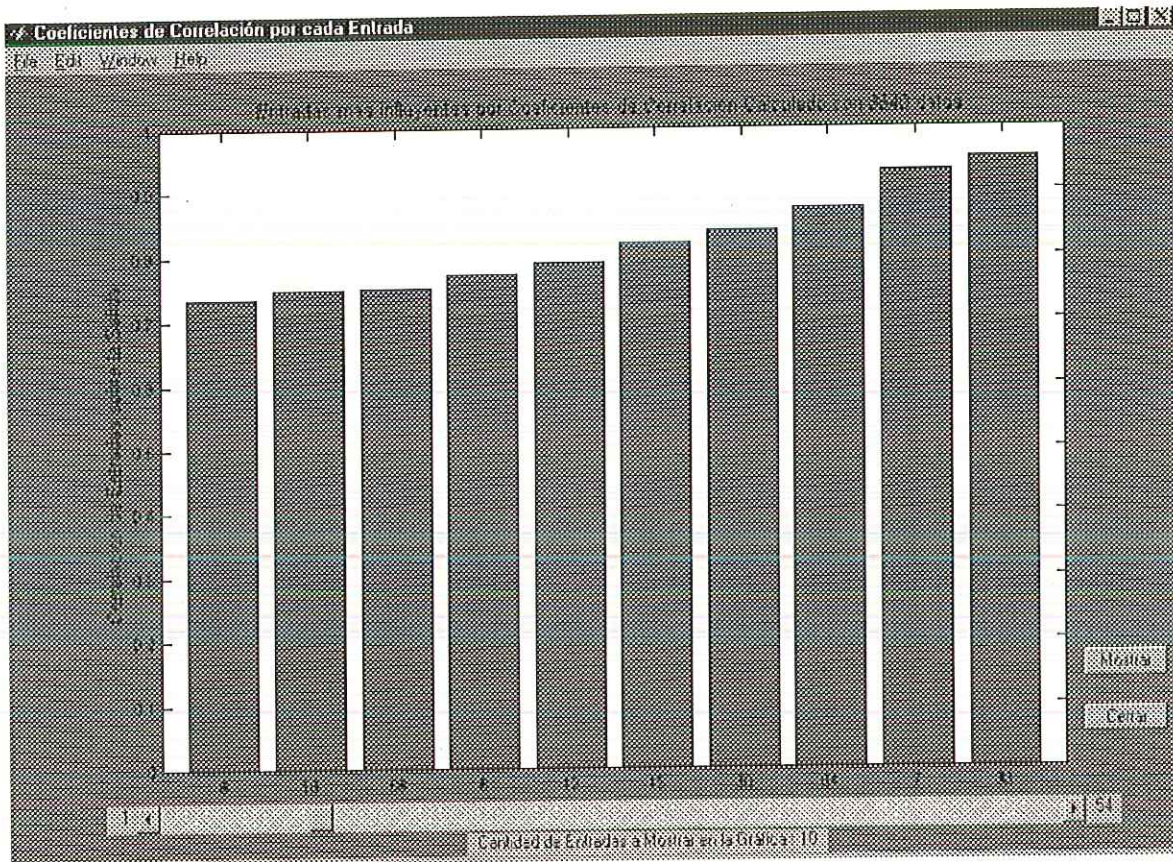
Con el botón **cerrar** abandona la ventana.

Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

3.2 Análisis por Coeficientes de Correlación

seleccione del menú **Selección de Variables** → **Análisis por Coeficientes de Correlación**

Suministra una gráfica de barras ordenada de acuerdo a la influencia por Coeficientes de Correlación que cada entrada tiene sobre la salida, así:



Al igual que la ventana anterior, la gráfica está provista de una barra de desplazamiento (slider) que permite visualizar el comportamiento de la cantidad de entradas que se desee hasta su máximo; solo debe desplazarse sobre la barra y oprimir **Mostar**

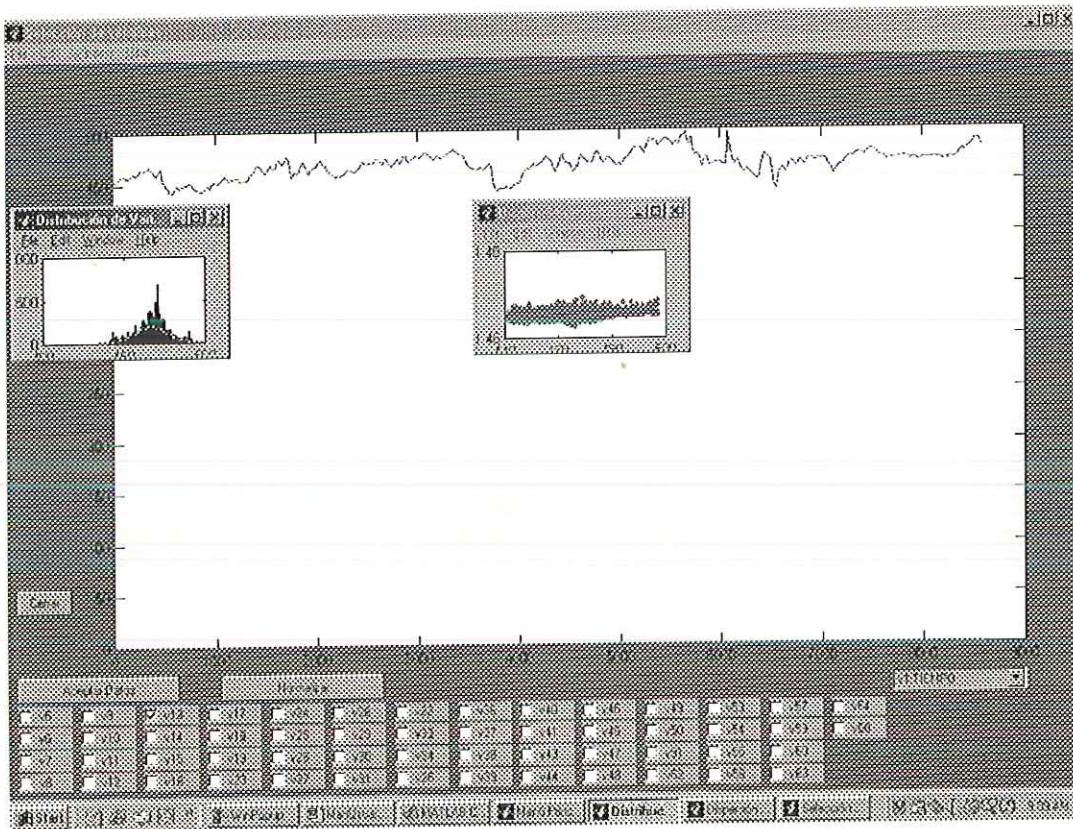
Con el botón **cerrar** abandona la ventana.

Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

3.3 Selección de Variables de Entrada

seleccione del menú **Selección de Variables** → **Selección de Variables de Entrada**

Esta opción abre una ventana en la cual se muestra siempre la grafica de la salida real, como también permite ver dos ventanas más con las gráficas de dispersión de la variable seleccionada contra la salida real, al igual que muestra otra ventana con la distribución de la ultima variable seleccionada.



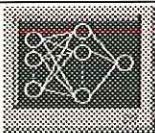
La ventana esta provista de tres botones cuya función se describe a continuación, un check box por cada variable que contenga el archivo de datos y una caja combobox (winpopup)

Descripción de botones:

Normalizar: Este botón permite normalizar (llevar a la misma escala) todas las variables registradas en la ventana, una vez realizada la normalización el botón cambia su nombre por Restaurar, permitiendo así la reversión de las variables a su estado normal.

Aceptar Datos: Este botón permite hacer una discriminación de las variables de menor importancia, de esta forma solo basta con seleccionar haciendo clic sobre cada una de aquellas variables que deseamos que hagan parte del proyecto.

Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000



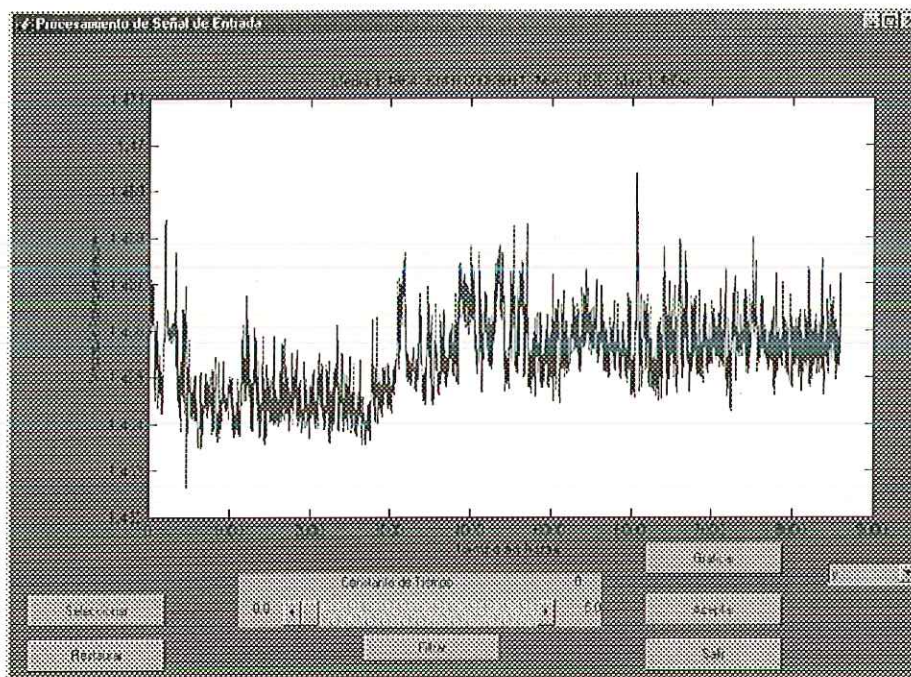
Combobox (Winpopup): Esta caja de persiana contiene una lista (nombre o tag) de todas las variables que contiene el archivo original.

Cerrar: Este botón cierra la ventana como también las ventanas auxiliares de dispersión y distribución

3.4 Tratamiento de Variables

Seleccione del menú **Selección de Variables** → **Selección de Variables de Entrada**

Esta opción permite mostrar una ventana con una barra de desplazamiento y seis botones permanentes los cuales sirven para hacer el tratamiento requerido a cada variable, la función de cada botón se describe a continuación:



Descripción de botones y orden de uso:

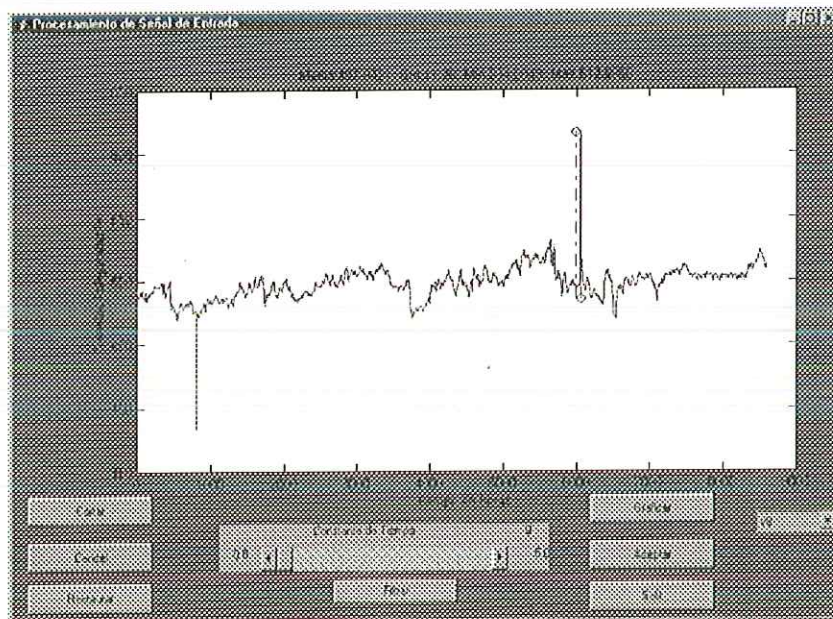
Winpopup: Esta persiana permite seleccionar la variable sobre la cual se debe hacer un filtro o un corte, luego de seleccionada debe hacer clic sobre el botón graficar para poder efectuar sobre ella el tratamiento.

Graficar: Este botón permite mostrar (graficar) la variable previamente seleccionada, como también asigna su contenido a una variable de tratamiento.

Slider: Esta barra de deslizamiento permite seleccionar hasta máximo cinco (5) unidades de filtro para ser tratados en la variable.

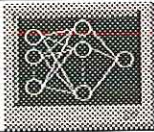
Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

- Filtrar:** Este botón efectúa la cantidad de filtro seleccionado sobre la variable en cuestión, solo tiene efecto por una vez, para que el filtro quede registrado en la variable se debe hacer clic sobre el botón aceptar.
- Aceptar:** El botón aceptar permite registrar cualquier cambio sobre la variable así este sea efectuado con corte o filtraje de la variable.
- Selección:** Esta opción tiene un efecto especial sobre el mouse dejándolo con facultades gráficas para hacer clic sobre un área de la gráfica sobre la cual se quiere hacer corte, aquí se debe hacer clic sobre las dos esquinas que formen un rectángulo sobre el que se abarca el área de corte. Inmediatamente sea formado el rectángulo de corte en la ventana, aparece activado el botón < **cortar** > es necesario hacer un corte o cancelar el área seleccionada para seguir manejando la aplicación.



- Cortar:** El botón cortar permite retirar de la grafica los puntos seleccionados efectuando una interpolación sobre los datos que quedan, es importante observar que el botón < **seleccionar** > cambia de etiqueta una vez este demarcada el área de corte, este botón cambia por el nombre de < **cancelar** >.
- Restaurar:** Este botón restaura a los valores originales, la variable en tratamiento en el momento presionarlo.
- Salir:** Este botón cierra la ventana dejando guardado el cambio hechos a las variables.

Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000



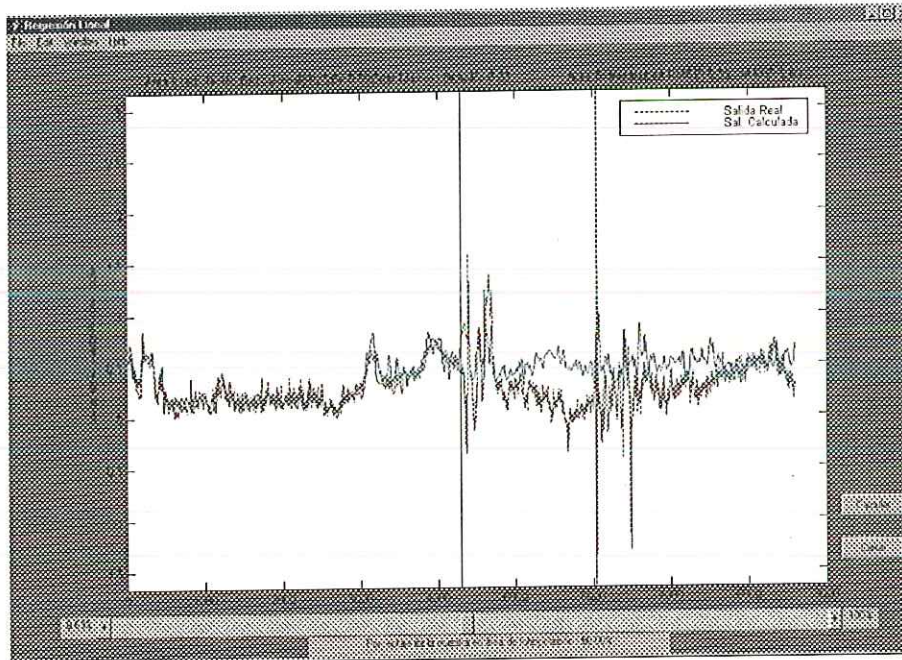
4 ENTRENAMIENTO Y VALIDACIÓN

La persiana de entrenamiento y validación posee tres opciones que permiten analizar y recolectar datos para la construcción de una salida virtual.

4.1 Entrenamiento por Regresión Lineal

seleccione del menú **Entrenamiento y Validación** → **Regresión Lineal**

Éste cálculo toma su tiempo dependiendo del numero de variables y de la cantidad de datos en cada variable.



Inicialmente se muestra un entrenamiento con el cincuenta por ciento (50%) de los datos y una validación con el restante (50%) pero el usuario puede hacer una variación de los mismos deslizando la barra (**slider**), aumentando o disminuyendo la cantidad por defecto, la ventana también posee un botón llamado **<calcular>** y permite efectuar los cambios que el usuario seleccionó.

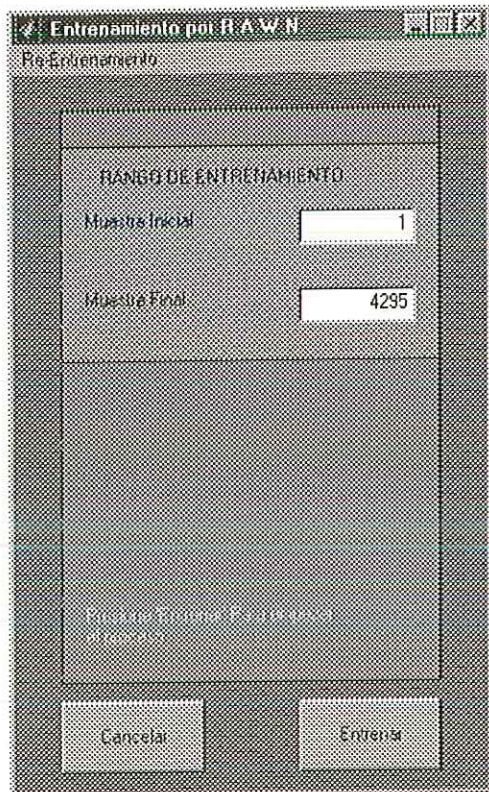
Este entrenamiento solo tiene efectos gráficos y no establece pesos para la red neuronal en construcción.

Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

4.2 Entrenamiento por Pesos de Activación Aleatoria

seleccione del menú Entrenamiento y Validación → Pesos de Activación Aleatoria

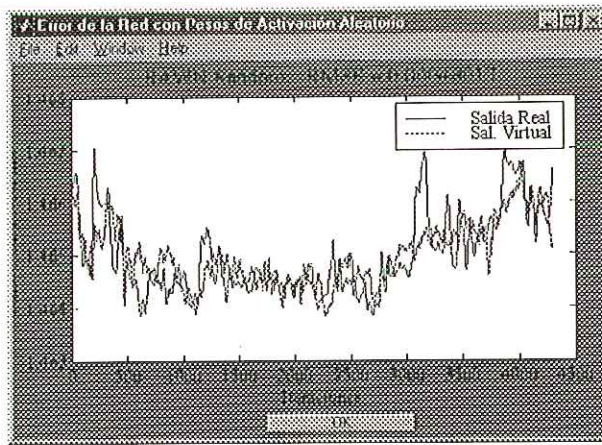
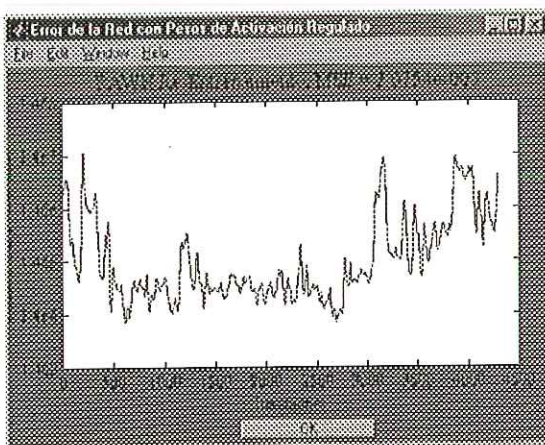
Esta ventana permite definir numéricamente el rango de entrenamiento sobre el conjunto de datos, por defecto aparece la primera mitad de los datos, no debe digitar un rango superior a la cantidad de datos totales de la muestra. Se recomienda que el rango de entrenamiento no sea superior al rango de validación



En la ventana también aparece dos botones y un menú de persiana con la opción de reentrenamiento

El botón **entrenar** permite hacer un entrenamiento básico de los datos seleccionados, si el resultado no es el esperado, (aceptable), se debe presionar el botón < O.K.> de la nueva ventana en la que aparece el resultado del entrenamiento como se muestra en la figura de la izquierda:

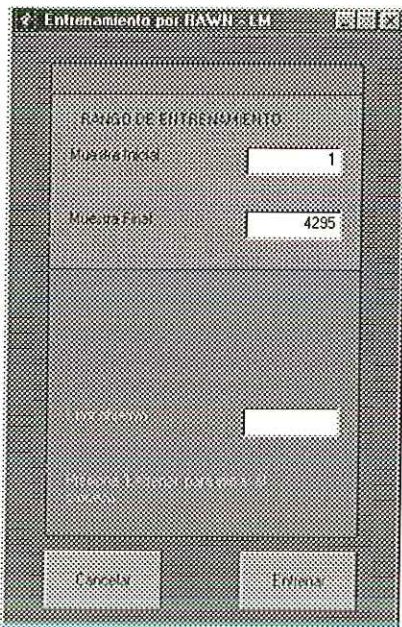
“Sirve recordar que este gráfico solo muestra resultado de entrenamiento con únicamente la cantidad de datos seleccionados” luego seleccione de la persiana de re-entrenamiento <optimización de escala completa> “este proceso es lento” pero obtiene buen resultado como se muestra en la figura de la derecha:



Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

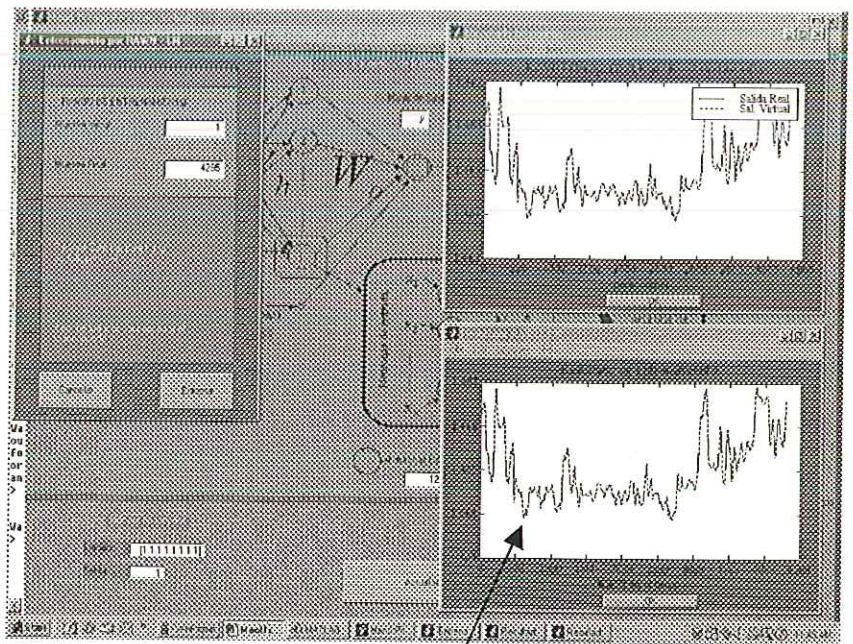
4.3 Entrenamiento por el Método de Proyección de Variables

seleccione del menú **Entrenamiento y Validación** → **Método de Proyección de Variables**



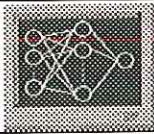
Esta opción tiene una ventana muy parecida a la anterior pero con las siguientes diferencias: no existe submenú de persiana o reentrenamiento, además tiene una nueva opción de caja de texto en el cual se le debe indicar a la red cuál es el **objetivo de error** que se desea alcanzar, si este valor no es especificado el programa coloca por defecto el valor de **< 0.02 >**.

El resultado de este entrenamiento se muestra en la siguiente ventana sobre la cual se puede ver una ventana con resultado intermedio del entrenamiento y otra ventana con el resultado general.



El entrenamiento termina cuando el sistema muestra una ventana desplegable con el mensaje **< Objetivo de error alcanzado >**, si este entrenamiento se realiza después de haber sido entrenada la red con otro método o con el mismo el sistema optimiza los pesos anteriores y como resultado solo muestra la ventana resultado del entrenamiento.

Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000



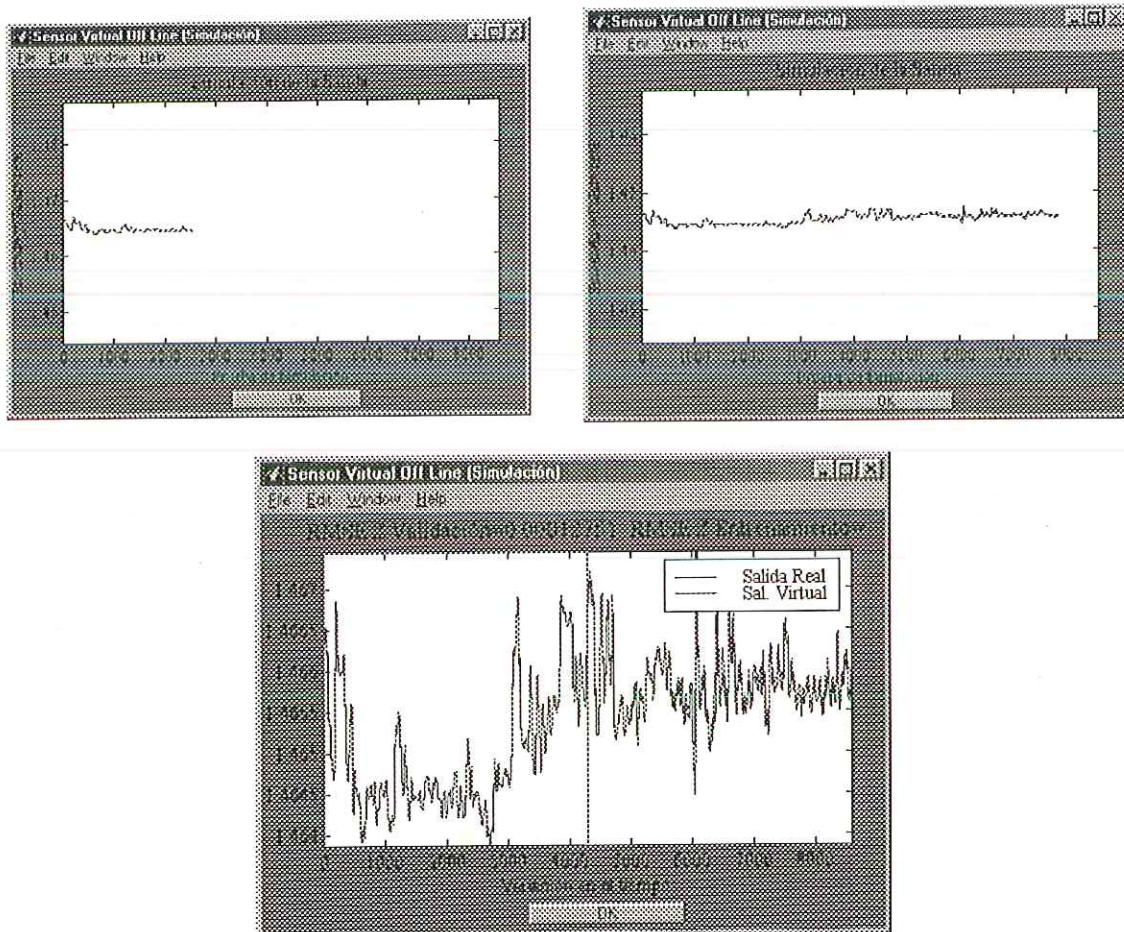
5 PRUEBA DEL SENSOR

Esta persiana del menú de barra contiene fundamentalmente dos opciones que permiten esquematizar el sensor de dos formas; una primera es *fuera de línea* y la otra es *en línea*, cabe resaltar que los resultados obtenidos son igualmente acertado aunque la filosofía de manejo es un poco diferente.

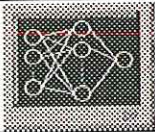
5.1 Salida Real y Virtual fuera de línea

seleccione del menú **Prueba del sensor** → **Fuera de Línea** → **Salida Real y Virtual**

En esta opción se abre una ventana sobre la cual se grafica punto a punto la salida real de color azul y la salida virtual de color verde

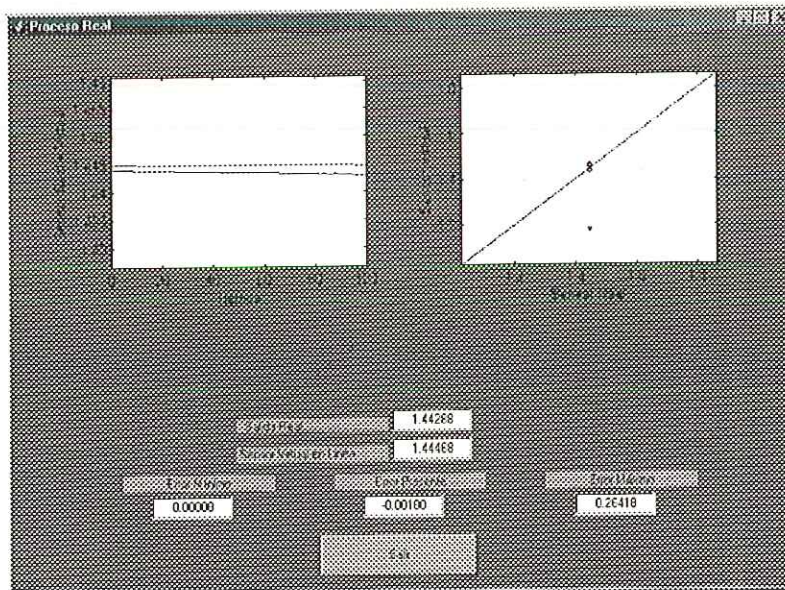
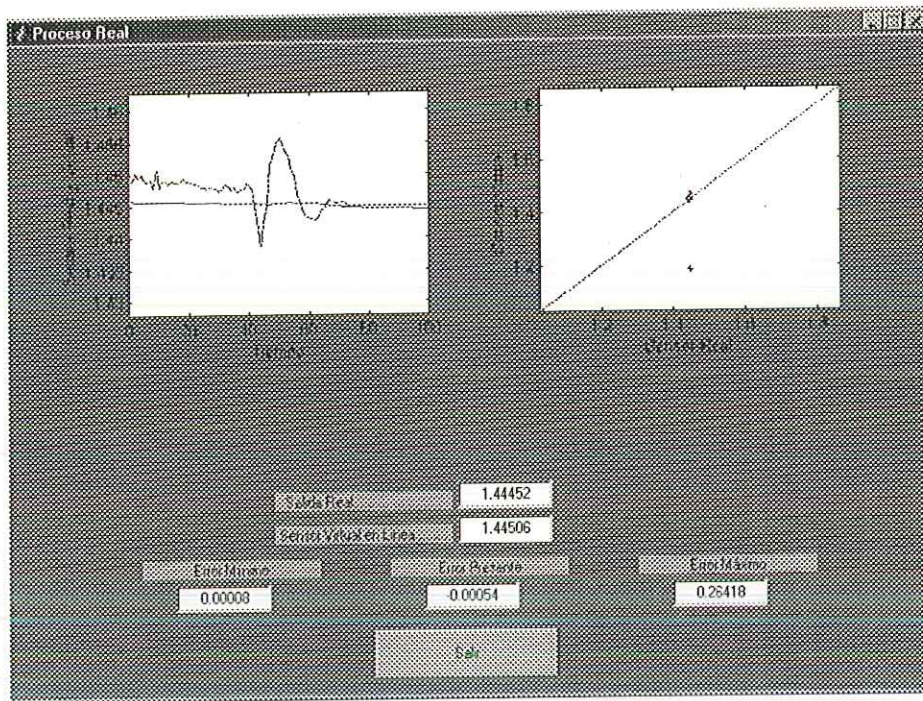


Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000



5.2 Salida Real y Virtual en Línea

seleccione del menú Prueba del sensor → en Línea → Salida Real y Virtual



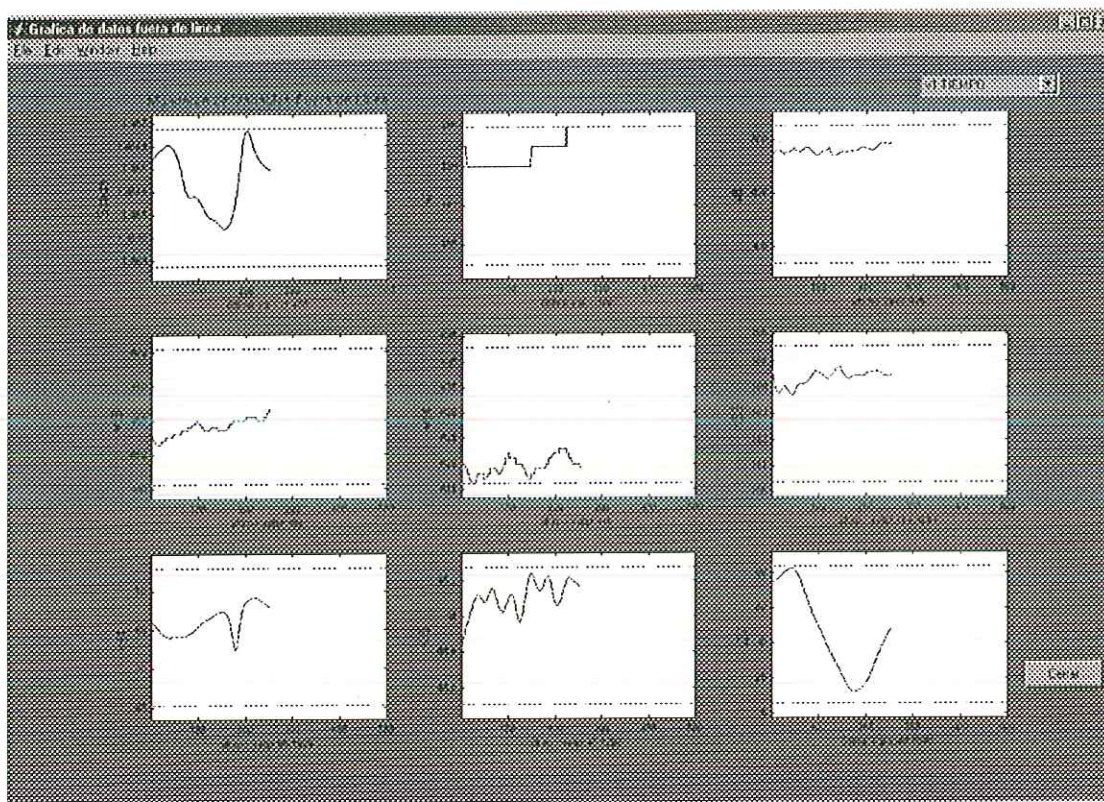
Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

5.3 Monitoreo de Variables

seleccione del menú Prueba del sensor → Fuera de Línea → Monitoreo de Variables

o seleccione: Prueba del sensor → En Línea → Monitoreo de Variables

Esta opción permite visualizar en la misma pantalla, el comportamiento de la salida, al igual que de las diferentes variables seleccionadas para el entrenamiento. Dependiendo de la opción seleccionada los datos graficados serán en línea o los que están actualmente cargados en la aplicación.



Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

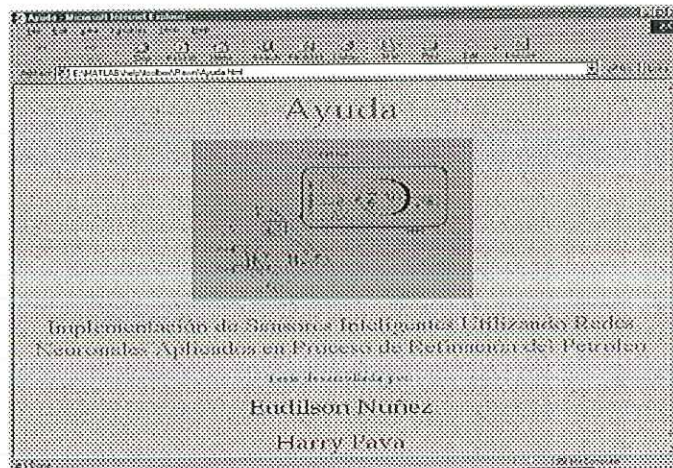
6 AYUDA

Esta opción del menú de barra permite mostrar de forma general asistencia en línea de cómo manejar este programa, como también encontramos este manual que de una u otra forma puede servir de ayuda para los que inician el manejo del mismo.

6.1 Manual en Línea

seleccione del menú **Ayuda** → **Manual en Línea**

Esta opción hace un enlace con el Internet Explorer de Microsoft Windows y permite cargar un archivo de tipo .html sobre el cual esta hecha la ayuda de este programa como se muestra en la figura siguiente:



6.2 Acerca de

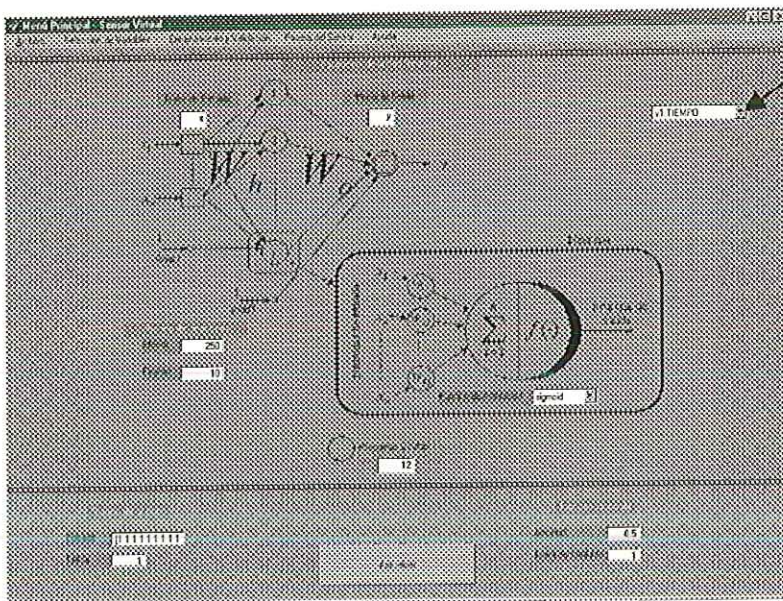
seleccione del menú **Ayuda** → **Acerca de**



Elaboró: E. Núñez – H. Pava	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

6.3 Nombre de Variables

seleccione del menú Ayuda → Nombre de Variables



Esta opción inserta en la ventana principal el **tag** (nombre original de las variables) permitiendo así poder ayudar a referenciar las variables de mayor influencia en la red.

Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

Archivo de Arranque del Sistema

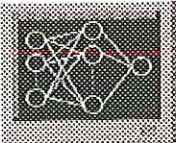
Archivo: tesrawn.ini

```

pathMicrosoft Excel ='c:\Archiv~1\Micros~2\Office\'; %Directorio donde se encuentre el Microsoft
Excel.exe %
NomenclraFilas ='l'; %Para Microsoft Excel en Español 'l':líneas, para Microsoft Excel
en Inglés: 'r' de row%
coltiempo =[1]; %Columna donde está el consecutivo de tiempo %
colsalida =[61]; %Columna donde está la salida para éste caso el la columna 61 %
coeliminar =[2, 3, 70]; %Columnas a eliminar, porque no son (ej. fecha y hora....) %
tiempoespera =100000; %Tiempo máximo de espera para cargue de información %
CantDatosGra =40; %Cantidad de datos de la gráfica de salida en línea %
LineaTitulos =1; %Línea donde se encuentran los títulos de las variables en el archivo %
  
```

Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

ANEXO B

	MAESTRÍA EN CIENCIAS COMPUTACIONALES UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA	SVRN-02
	SENSOR VIRTUAL REDES NEURONALES Manual de Instalación	Pag 1 de 8

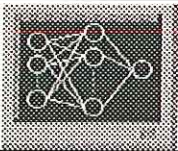
MAESTRÍA EN CIENCIAS COMPUTACIONALES

Universidad Autónoma de Bucaramanga
Colombia

SENSOR VIRTUAL REDES NEURONALES

Guía de Instalación
Ver. 1.0

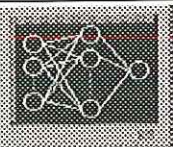
Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000



Contenido

Contenido	2
INTRODUCCIÓN	3
1 Software de Desarrollo	4
2 Requerimientos mínimos de Hardware y Software	5
2.1 Hardware	5
2.2 Software	5
3 Instalación	5
3.1 Copiado de archivos.....	6
3.2 Achivo de configuración	6
3.3 Prueba de funcionamiento.....	8

Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

	MAESTRÍA EN CIENCIAS COMPUTACIONALES UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA	SVRN-02
	SENSOR VIRTUAL REDES NEURONALES Manual de Instalación	Pag 3 de 8

INTRODUCCIÓN

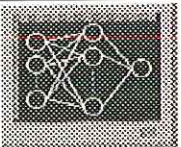
Este manual le ayuda a instalar paso a paso el software Sensor Virtual, Los sensores virtuales son una forma muy útil de darle a operadores e ingenieros más información sobre el desempeño actual de sus unidades. Además, un sensor inteligente es altamente útil para predecir el desempeño futuro.

Las pruebas de laboratorio usualmente involucran retardos. Los operadores y los ingenieros pueden obtener de ellos la información que necesitan, pero no la obtienen tan rápido como frecuentemente se necesita. Un Sensor Virtual puede mejorar la calidad, disminuir el tiempo y frecuencia de los resultados y disminuir los costos de laboratorio al mismo tiempo.

Es necesario que siga cuidadosamente Este manual donde se orienta la instalación de la aplicación para ser utilizada en el cuarto de control de la planta y así brindar información efectiva y oportuna que apoye la toma de decisiones en el ajuste del proceso.

La aplicación está desarrollada con una interfaz de ventanas (Windows) permitiendo así seguir el estandar de software, como también brindar al usuario una aplicación amigable, acorde con las necesidades presente y futura.

Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

	MAESTRÍA EN CIENCIAS COMPUTACIONALES UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA	SVRN-02
	SENSOR VIRTUAL REDES NEURONALES Manual de Instalación	Pag 4 de 8

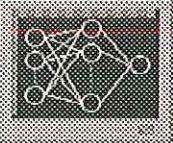
1 Software de Desarrollo

Para la implementación de esta aplicación se utilizó MatLab 5.2.0.3085 por ser una de las mejores herramientas para el manejo de los cálculos requeridos en la obtención del sensor virtual.

Dentro de sus principales características están:

- Sencillo manejo de operaciones y funciones matemáticas
- Manejo una buena cantidad de datos simultaneamente
- Permite el manejo de la programación orientada a eventos en el cual el programador genera (ya sea mediante un código o mediante una herramienta de diseño de interfaz gráfica de usuario "*guide*") las pantallas, menús, etc. Los asocia con procedimientos de validación e inicialización, luego cede el control al sistema operacional, el cual solo bajo ciertas acciones específicas (eventos) cede el control a los procedimientos creados. (Entre los eventos se cuentan: ingreso o salida de un campo, selección de una opción del menú, el presionar un botón o el mover el puntero del Mouse.)
- Permite crear aplicaciones que manejen la interfaz gráfica de usuario.
- Permite la compilación de su funciones para generar su propio archivo ejecutable.

Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

	MAESTRÍA EN CIENCIAS COMPUTACIONALES UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA	SVRN-02
	SENSOR VIRTUAL REDES NEURONALES Manual de Instalación	Pag 5 de 8

2 *Requerimientos mínimos de Hardware y Software*

2.1 *Hardware*

Procesador : Pentium II o Superior

RAM : 64 MB

Velocidad : 200 Mhz

Espacio en disco requerido: 60 MB

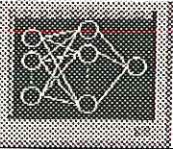
- Una impresora instalada
- Un Mouse
- Monitor con alta resolución de interfaz gráfica.

2.2 *Software*

El computador para instalar la aplicación debe tener preinstalado

- Un Sistema Operativo Windows 95 o superior
- Software Matlab ver. 5.2.0. o superior (es importante resaltar que en esta aplicación se necesita tener actualizado la versión del matlab para windows 95 y windows 98, con respecto a las librerías de mejora efectuado para el enlace dinámico de datos (DDE) con Microsoft excel, como también se requiere la actualización de matlab para windows 98 con respecto a las librerías que crean los checkbox de una ventana gráfica (ofrecidas en el cd de instalación).
- Software Microsoft Excel.
- Hoja con macro de datos fenol-pi3.xls.

Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

	MAESTRÍA EN CIENCIAS COMPUTACIONALES UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA	SVRN-02
	SENSOR VIRTUAL REDES NEURONALES Manual de Instalación	Pag 6 de 8

3 Instalación

- Se recomienda cerrar todas las aplicaciones antes de realizar la instalación.
- Antes de la instalación debe realizarse la configuración y verificación de los requisitos.

Inserte el CD etiquetado sensor virtual ver. 1.0 en su unidad de cd rom y verifique que el contenido coincida con las siguientes carpetas.

D:\Doc.....documento de word incluido este manual.

D:\Datos...datos de prueba y proceso.

D:\Aplicación....aplicación sensor virtual.

D:\Ayuda...manual del usuario en HTML.

D:\herramienta...archivos de herramientas.

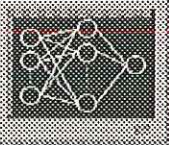
D:\patch...Correcciones para windows 95 y 98.

3.1 Copiado de archivos

Inserte el CD marcado "Sensor Virtual v. 1.0" en su unidad de CD-ROM.

En su unidad de disco debe existir una carpeta llamada Matlab y en el interior de la misma existen 2 carpetas de interes nuestro llamada help\ y otra llamada toolbox\ propias de matlab. Una vez identificadas las carpetas arrastre desde el CD el contenido de la carpeta ayuda y sueltelo dentro de la carpeta toolbox, que a la vez se encuentra dentro la carpeta Help, igualmente tome desde el CD el

Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

	MAESTRÍA EN CIENCIAS COMPUTACIONALES UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA	SVRN-02
	SENSOR VIRTUAL REDES NEURONALES Manual de Instalación	Pag 7 de 8

contenido de la carpeta aplicación y depositelo dentro de la carpeta toolbox (subcarpeta de matlab).

La aplicación busca por default los datos en la carpeta donde se encuentre cuando se inicia matlab, normalmente esta carpeta es llamada bin, si lo desea puede copiar el contenido de la carpeta Datos del CD, en el subdirectorio bin de matlab como tambien puede dejarlos y buscarlos con la aplicación en la unidad de CD.

Si en su equipo de trabajo usted tiene windows 95 o windows 98 copie los archivos de actualizacion desde la carpeta D:\patch bajo la siguiente explicación.

Problemas con DDE bajo windows 95 y 98:

Solucion:

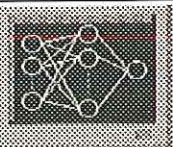
Reemplazar los archivos ddeinit.dll, ddeadv.dll, ddeunadv.dll que estan en el CD en la carpeta D:\PATCH\DDE\ en la ruta *:\matlab\toolbox\matlab\winfun\

Problemas con los checkbox bajo windows 98:

Solución:

Reemplazar los archivos que se encuentran en la carpeta D:\PATCH\CHECKBOX\ en la dirección *:\matlab\toolbox\matlab\winfun

Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

	MAESTRÍA EN CIENCIAS COMPUTACIONALES UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA	SVRN-02
	SENSOR VIRTUAL REDES NEURONALES Manual de Instalación	Pag 8 de 8

3.2 Archivo de configuración

Ingresa a la aplicación y modifique el archivo de configuración de acuerdo a sus parámetros. Es decir es necesario definir en el programa la ruta donde se encuentra el ejecutable de excel del PC donde instaló la aplicación. Ver Manual del Usuario numeral 1.3 Configuración de la aplicación, al igual que es necesario establecer en el archivo de configuración si el excel esta en español o en inglés para poder asignar "l" o "r".

3.3 Prueba de funcionamiento

Al instalar la aplicación copiando los archivos como son explicado e este manual, usted dispondrá de una herramienta poderosa que le ayudará a predecir el índice de viscosidad para una plata de parafinas en la industria de refinación de petróleos, como también es fácil hacer adaptaciones futuras a esta herramienta para poder predecir cualquier variable cuyo comportamiento dependa de un conjunto de entradas.

El funcionamiento de esta aplicación es bastante sencillo todo depende de como siga las instrucciones dadas en el manual del usuario, donde se dispone de forma clara como ejecutar la aplicación.

Elaboró: E. Núñez – H. Paba	Revisó:	Aprobó: UNAB
Fecha: oct. 1999 a oct. 2000	Fecha:	Fecha: Octubre del 2000

ANEXO C

RELACION DE INFORMACIÓN ENTREGADA EN DISCO COMPACTO

Junto con este documento se entrega toda la información del proyecto en un compact disk que contiene la siguiente información:

D:\Doc	Documento elaborado en Microsoft Word (este manual).
D:\Datos	Datos de prueba y proceso.
D:\Aplicación	Aplicación del Sensor Virtual en Matlab (código fuente).
D:\Ayuda	Manual del usuario en formato HTML.
D:\patch	Correcciones de Matlab 5.2 para Windows 95 y 98.

ANEXO D

ANEXO D. HOJA DE ESPECIFICACIONES DEL SENSOR REAL

Marca	K-PATENTS		
Process Refractometer Instrument	PR-01-S-57	Serial Number	97J05-2827
Approval	FM	Probe	HPY
Supply voltage	100-115 V /50-60 Hz		
Current output	4-20 mA		
Body	AISI 316L		
Prism	Spinel		
Packing	Kalrez 4079		
Connection	Flange ANSI 150 lbs		
Process			
Temperature norm	45 C max 70 C		
Curve	RI (25)		
Medium	Rafinato		
Range	1,4508 - 1,4731 RI		
Lense f	18 mm		
Parameters in EEPROM			
Zero	14,508	Span	4
Bias	1,464	Gain	0
Tnorm	45	Coeff	0
Ribias	45,703	Rigain	-0,0297
TMPCO	-2,25	TMPC1	0
D.P.	0,0223		
Squar	0,02881		
Tcadj	0,00035		
Risquar	28,776		
RI Cub	0		
Generales			
Refractive Index Range	Low range: RI 1.320 ... 1.460	Ambiente temperature	Max 45 C (113 F), min -20 C (-4 F)
Max span	High range: RI 1.380... 1.530	Process preasure	Flange connections up to 25 bar (350 psi)
Accuracy	RI 0.08	Sensor weinght	10.5 kg
Speed of rensponse	RI +- 0.0002	Current output	4-20 mA / 0-20 mA, max load 1000 Ohm
Damping time constant	0.8 s undamped	Serial output	RS485/RS232
Process temperature	Selectable up to 5 min	Indicating Transmitter	Enclosure IP66 Nema 4X
	Max 150 C (300 F)		

ANEXO E

**HOJA DE ESPECIFICACIONES DEL SENSOR VIRTUAL PARA LA MEDICIÓN DEL
INDICE DE REFRACCION EN LA PLANTA DE FENOL**

Variables de Entrada (correlación-dispersión)	V6,18,14,10,32,63,17,15,44,7
Neuronas de la Capa Oculta	5
Neuronas de Salida	1
No. Muestras para entrenamiento (típico)	1000
No. Iteraciones Interna	250 (entrenamiento)
No. Iteraciones Externa	10 (entrenamiento)
Factor de Escala - Amplitud	-0.05 (entrenamiento)
Factor de Escala – Fuera del Conjunto	1 (entrenamiento)
RMSE	0.00089 (de IR)
Pesos de Entrada	Ver Tabla 3.7
Pesos de Salida	Ver Tabla 3.8
Bias de Entrada y Salida	Ver Tabla 3.6
Ecuación del Sensor	$A_1 = \text{sigmoide}(W_1 * P, B_1)$ $Y_v = \text{Purlin}(W_2 * A_1, B_2)$ <p>Donde :</p> $P = \text{MatrizEntrada}$ $a = \text{Purlin}(n, b) = [nr, nc] = \text{size}(n)$ $a = n + b * \text{ones}(1, nc)$ $b = B_2, n = W_2 * A_1$