



NOMBRE DE LA ASIGNATURA: **Inteligencia Artificial**

Línea de Trabajo: **Procesamiento de Señales**

DOC – TIS – TPS - Horas Totales - Créditos

48 – 60 – 0 – 108 – 6

1.- Historial de la asignatura

Lugar y fecha de elaboración o revisión	Participantes	Observaciones (cambios y justificación)
Instituto Tecnológico de Morelia, Morelia, Michoacán, Noviembre del 2006	Dr. José Antonio Gutiérrez Gneccchi. Profesor de la Academia de Ingeniería Electrónica, ITM	Análisis y conformación de la asignatura
Instituto Tecnológico de Morelia, Morelia, Michoacán, Diciembre del 2006	Profesores de la maestría en Ing. Electrónica	Definición de la asignatura
I.T. Morelia, noviembre de 2014	Dr. José Antonio Gutiérrez Gneccchi M.C. Miguelangel Fraga Aguilar	Análisis y actualización de programa de estudios

2.- Pre-requisitos y correquisitos

Asignatura optativa a cursarse en el segundo semestre.

3.- Objetivo de la asignatura

Esta asignatura le presenta al estudiante las bases de inteligencia artificial, en específico *redes neuronales* y *lógica difusa*, para la solución de diversos problemas de interpolación, estimación de parámetros. El rango de tipo de redes utilizadas va desde redes para clasificación, hasta redes auto-organizables. Se incluyen varias aplicaciones en instrumentación y control.



4.- Aportacion al perfil del graduado

Presentarle a los alumnos técnicas de mapeo no-lineal, estimación de parámetros, control e identificación de sistemas, para solucionar problemas de diseño y optimización.

5.- Contenido temático por temas y subtemas

UNIDAD	TEMAS	SUBTEMAS
1	<p>INTRODUCCION AL USO DE LAS REDES NEURONALES</p> <p>TIEMPO: 6 HORAS</p>	<ol style="list-style-type: none"> 1. Reseña histórica de las redes neuronales (1 hora) 2. Principios y modelos matemáticos de neuronas (1 hora) 3. Definiciones (1 hora) 4. Elementos de una red neuronal (1 hora) 5. Clasificación de las redes neuronales (1 hora) 6. Casos de estudio: a) Aplicaciones en Instrumentación, b) Aplicaciones en control (1 hora)
2	<p>REGLAS DE APRENDIZAJE</p> <p>TIEMPO: 8 HORAS</p>	<ol style="list-style-type: none"> 1. Clasificadores: Perceptron (2 horas) 2. Redes con funciones lineales y sigmoides: Widrow-Hoff, y retropropagacion (2 horas) 3. Redes discretas: Hopfield, Kohonen y N-ésimas (2 horas) 4. Potencial de aplicación y consideraciones para implementación físicas y consideraciones para Implementación usando microprocesadores/microcontroladores/DSP's (2 horas)
3	<p>REDES DIGITALES</p> <p>TIEMPO: 6 HORAS</p>	<ol style="list-style-type: none"> 1. Funciones de nodos Booléanos (1 hora) 2. Modelos Booleanos de McCulloch and Pitts (2 hora) 3. Regla de entrenamiento de retropropagación de Rumelhart (1 hora) 4. LLNs (Large logic neurons) (1 hora)



		5. NODOS LÓGICOS PROBABILÍSTICOS (PLNs) (1 hora)
4	<p>APLICACIONES DE REDES NEURONALES PARA ESTIMACIÓN DE PARAMETROS E IDENTIFICACION DE SISTEMAS</p> <p>TIEMPO: 6 HORAS</p>	<ol style="list-style-type: none"> 1. Conceptos básicos y formulación del problema (1 hora) 2. Algoritmos de aprendizaje para redes neuronales de múltiples capas (1 horas) 3. Redes Neuronales basadas en funciones radiales (2 horas) 4. Estimación (mapeo no-lineal) (1 horas) 5. Aplicación a identificación de sistemas (1 horas)
5	<p>ASPECTOS COMPARATIVOS DE REDES NEURONALES VS LOGICA DIFUSA</p> <p>TIEMPO: 6 HORAS</p>	<ol style="list-style-type: none"> 1. Introducción a la lógica difusa (1 hora) 2. Terminología y definiciones (1 hora) 3. Aspectos comparativos de lógica difusa y redes neuronales (1 hora) 4.- Funciones de membresía y algoritmos de operación (1 horas) 5. Control inteligente con lógica difusa auto-organizable (SOFLIC) (1 horas) 5. Neurocontroladores (1 hora)

6.- Metodología de desarrollo del curso

La metodología se basa en clases presenciales, tutoriales, sesiones de laboratorio y tareas. Se sugiere llevar a cabo un grupo de prácticas que llevan a la producción de un proyecto final, que involucre la implementación de un sistema de instrumentación virtual. Se sugiere que el docente le provea al alumno de material escrito como guía del contenido del curso y propicie el que el alumno busque información acerca de los últimos avances en desarrollo y aplicaciones de la redes.

7.- Sugerencias de evaluacion

Exámenes teóricos, tareas, participaciones. Realizar ejercicios guiados para ayudarle al alumno a comprender como se llevan a cabo los procesos de entrenamiento de las redes y a analizar los indicadores, como gráficas de error, para determinar el correcto entrenamiento de la red. Utilizar datos reales y simulados para compara el desempeño de las redes en presencia de ruido. Realizar exámenes escritos así como prácticos.



8.- Bibliografía y software de apoyo

SOFTWARE DE APOYO

Software de análisis matemático:

MATLAB ver 5.3 en adelante incluyendo *SIMULINK, NEURAL NETWORK TOOLBOX, FUZZY LOGIC TOOLBOX, SYSTEM IDENTIFICATION TOOLBOX, DIGITAL SIGNAL PROCESSING TOOLBOX, IMAGE PROCESSING TOOLBOX.*

MATHCAD ver 2001 en adelante

MAPLE ver 6 en adelante

SCIENTIFIC WORKPLACE ver 4.0 en adelante

NeuroSolutions incluyendo generación de código en C++ y el plugin para Excel

Software para programación:

Compilador C/C++

Compilador Visual basic

Compilador Visual Java

Compilador para microprocesador/microcontrolador/DSP

(Code Composer (Texas Instruments), Code Warrior (Motorola),

BIBLIOGRAFÍA

Russel, J. Stuart, Norvig, Peter. Artificial Intelligence: A Modern Approach (2nd Edition). Prentice Hall. ISBN: 0137903952. (2002)

Haykin, Simon. Neural Network: A Comprehensive Foundation (2nd Edition). Prentice Hall. ISBN: 0132733501 (1998).

Fausett, Laurene V. Fundamentals of Neural Networks. Prentice Hall. ISBN: 0133341860 (1994)

Reed, Russel D., Marks, Robert J. Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks. Bradford Books. ISBN: 0262181908 (1999)

Hagan, Martin T., Demuth, Howard B., Beale Mark H. Neural Network Design (Electrical Engineering). Brooks Cole, 1st edition (1995). ISBN: 0534943322

Bishop, Christopher M. Neural Networks for Pattern Recognition. Oxford University Press. (1995). ISBN: 0198538642

Rogers, Joey. Object Oriented Neural Networks in C++. Morgan Kaufmann. Book and disk (1st Edition) (1996). ISBN: 0125931158

Masters, Timothy. Practical Neural Network Recipes in C++. Morgan Kaufmann. Book and Disk (1st Edition) 1993. ISBN: 0124790402



Ross, Timothy J. Fuzzy Logic with Engineering Applications. John Wiley and Sons, 2nd Edition (2004)
ISBN: 0470860758

Tanaka, Kazuo and Nimura, Tak. An introduction for Fuzzy Logic for Practical Applications. Springer-Verlag (1996) ISBN: 0387948074

Kartalopoulos, Stamatios V. Understanding Neural Networks and Fuzzy Logic: Basic Concepts and Applications (IEEE Press Understanding Science and Technology Series).

Wiley-IEEE Press (1995). ISBN 0780311280

Lecturas Adicionales

Aleksander, I. (ed.). Neural Computing Architectures. North Oxford Academic Publishers. 1991

Alexander, I. Introduction to Neural Nets, Chapter 9 in Applied Artificial Intelligence. K. Warwick (ed.) Peter Peregrinus Ltd. 1991.

Benes, J. On Neural Networks. Kybernetice, 26, No. 3 pp. 232-247 1990

Hebb, D.E. Organization of Behaviour, Wiley 1949

Hopfield, J.T. Neural Networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. USA 79, pp. 2554-2558

Kohonen, T. Self-Organization and associative memory. Springer-Verlag 1984

Minsky, M. and Papert, S. Perceptrons. MIT Press. 1969

Rosenblatt, F. Principles of Neurodynamics. Spartan. 1962

Rumelhart, D.E. and McClelland, J.L. Parallel Distributed Processing. MIT Press, Vols 1. y 2. 1986

9.- Prácticas propuestas

1.- Regla de entrenamiento PERCEPTRON (solución de problemas de clasificación). Clasificación con una capa, clasificación con múltiples capas; clasificación y efectos de valores fuera de rango; clasificación y efectos de entrenamiento con vectores de entrada linealmente separables e inseparables. Utilización de software de simulación para facilitar el procedimiento de implementación del método de entrenamiento. Prueba con datos simulados y datos reales. Implementación de programa usando compiladores como se proponen en el apartado 9.

2.- Regla de entrenamiento Widrow-Hoff . Uso de la red ADALINE (ADaptive LINear Element). Ejemplos de asociación de patrones, ejemplos de entrenamiento con pares de vectores entrada-salida sobredeterminados (mayor número de ecuaciones de restricción que salidas); ejemplos de entrenamiento con pares de vectores de entrada-salida sub-determinados (menor número de ecuaciones de restricción que variables); ejemplos de asociaciones de patrones con múltiples neuronas; efectos de entrenamiento usando vectores de entrada linealmente dependientes, efectos de la variación de valores de la razón de aprendizaje. Utilización de software de simulación para facilitar el procedimiento de implementación del



método de entrenamiento. Prueba con datos simulados y datos reales. Implementación de programa usando compiladores como se proponen en el apartado 9.

3.- Regla de entrenamiento de retro-propagación. Efectos en el entrenamiento dependiendo de una superficie de error no-lineal, efectos de una razón de aprendizaje muy grande, efectos de los errores locales y globales en el desempeño de la red; Aproximación de funciones usando redes de retro-propagación. Efectos de usar un número insuficiente de neuronas. Mejorando el método de entrenamiento basado en retro-propagación: Ejemplos de entrenamiento mediante el método del momentum; efectos de escoger mejores condiciones iniciales de entrenamiento; método de razón de aprendizaje adaptativa. Utilización de software de simulación para facilitar el procedimiento de implementación del método de entrenamiento. Prueba con datos simulados y datos reales. Implementación de programa usando compiladores como se proponen en el apartado 9.

4.- Redes Hopfield, Kohonen y funciones de nodos booleanos. Efectos de la agrupación de nodos booleanos (LLN: Large Logic Neurons); ejemplos de aplicación en reconocimiento de patrones. Utilización de software de simulación para facilitar el procedimiento de implementación del método de entrenamiento. Prueba con datos simulados y datos reales. Implementación de programa usando compiladores como se proponen en el apartado 9.

5.- Mapeo no-lineal usando redes de funciones radiales. Estudio del desempeño de redes basadas en funciones radiales para estimación y/o mapeo no-lineal. Efecto de ajuste del ancho, número y posición de las funciones radiales. Utilización de software de simulación para facilitar el procedimiento de implementación del método de entrenamiento. Prueba con datos simulados y datos reales. Implementación de programa usando compiladores como se proponen en el apartado 9.

6.- Uso y aplicaciones de lógica difusa para selección de clases. Agrupación de datos y selección de clases en 2D con agrupamiento c-means; mapeo (aproximación) de funciones no-lineales usando lógica difusa. Funciones de membresía, Utilización de software de simulación para facilitar el procedimiento de implementación del método de entrenamiento. Prueba con datos simulados y datos reales. Implementación de programa usando compiladores como se proponen en el apartado 9.

7.- Ejemplo de control no-lineal usando redes neuronales y/o lógica difusa. Diseño de la red, entrenamiento simulación y prueba de desempeño con respecto de otros controladores (por ejemplo, con respecto de la utilización de un controlador lineal aplicado a un sistema no-lineal, y con respecto de un modelo de cancelación perfecta).

10. Nombre y Firma del Catedrático Responsable:

M.C. Miguelangel Fraga Aguilar