

**UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA CIVIL
SECCIÓN DE POSGRADO**



**MODELO NEURONAL DE DEMANDA DE TRANSPORTE EN
REDES VIALES URBANAS**

TESIS

**PARA OPTAR EL GRADO DE MAESTRO EN CIENCIAS
CON MENCIÓN EN INGENIERÍA DE TRANSPORTE**

ELABORADO POR

RICARDO LENIN BECERRA GUEVARA

ASESOR

MSc. LEONARDO FLORES GONZÁLEZ

LIMA-PERÚ

2014

MODELO NEURONAL DE DEMANDA DE TRANSPORTE EN REDES
VIALES URBANAS

ING° RICARDO LENIN BECERRA GUEVARA

Presentado a la Sección de Posgrado de la Facultad de Ingeniería Civil en cumplimiento
parcial de los requerimientos para el grado de:

MAESTRO EN CIENCIAS CON MENCIÓN EN INGENIERÍA DE TRANSPORTE
DE LA
UNIVERSIDAD NACIONAL DE INGENIERÍA

2014

Autor : Ing. Ricardo Lenin Becerra Guevara

Recomendado : Msc. Ing. Leonardo Flores González
Asesor de la Tesis

Aceptado por : PhD Jorge Elias Alva Hurtado
Jefe de la Sección de Posgrado

RESUMEN

Esta tesis explora las posibilidades de uso de las redes neuronales de retropropagación en el problema de la estimación de demanda de transporte centrándose en aspectos de diseño metodológico y de arquitectura/estructura de las redes. Estudia también el uso de mapas auto-organizados de Kohonen como herramienta para el análisis de datos y para la construcción de conjuntos de entrenamiento y testeo de redes de retropropagación.

Mediante una serie de experimentos se encuentran varios elementos que inciden en el proceso de aprendizaje de las redes de retropropagación y que no se toman actualmente en consideración en modelación de demanda de transporte, dentro de los que resaltan el tipo de función de activación utilizada; la composición del conjunto de entrenamiento; la secuencia de presentación de pares de entrenamiento. Estos factores pueden llevar a encontrar mejores soluciones y/o a tener un mayor control del tipo de solución alcanzada y por lo tanto se sugiere prestarles mayor atención. Se propuso una metodología que incluyó estos elementos combinados con los tradicionales de una manera bastante específica, de la cual se obtuvo buenos resultados.

Las redes de Kohonen mostraron en los experimentos realizados ser una excelente herramienta, permitiendo a) Mayor facilidad para encontrar registros de datos anómalos; b) Revisar la hipótesis de que las variables independientes elegidas pueden explicar la variable dependiente; construir segmentaciones de datos que permitan el uso de más de una especificación de modelo predictivo; reducir el número de pares de entrenamiento necesarios en una red de retropropagación.

ABSTRACT

This thesis explores the possibilities of using backpropagation neural networks on the problem of estimating transport demand focusing on methodological aspects of design and architecture / structure of the networks. He also studied the use of Kohonen self-organizing as a tool for data analysis and for the construction of training and testing sets backpropagation network maps.

Through a series of experiments are several elements that affect the process of learning and backpropagation networks that are not currently taken into account in modeling transport demand within that highlight the type of activation function used; the composition of the training set; presentation sequence of training pairs. These factors can lead to better solutions and / or have greater control on the type of solution reached and therefore suggests further attention. A methodology that included these traditional elements combined with a fairly specific way, which was obtained good results was proposed.

Kohonen networks showed in experiments be an excellent tool, allowing a) easier to find records of anomalous data; b) review the hypothesis that the independent variables chosen may explain the dependent variable; Slicers build enabling the use of more than one specification of predictive model; pairs to reduce the number of necessary training a backpropagation network.

DEDICATORIA

A mis padres Ricardo y Nilda, por todo el apoyo y confianza depositada durante toda esta etapa de preparación; motivo de superación y los principales autores en mi formación personal, a mis hermanas Paola y Stephanie. A mis familiares y amigos que siempre me han ofrecido su apoyo y cariño

AGRADECIMIENTOS

Queda grabado por siempre en estas líneas, mi más sincero agradecimiento al Msc. Ing. Leonardo Flores González por la disposición, la paciencia y el asesoramiento en el desarrollo de la presente tesis, a mis profesores de la Maestría en Ciencias con mención en la Ingeniería de Transportes, quienes con su empuje me han permitido aprender y a mis colegas, quienes me brindaron su apoyo durante el desarrollo de la presente tesis.

Le doy las gracias al Dr. Pedro C. Espinoza H., quien aportó las capacidades y conocimientos que en su momento necesité.

DESCRIPTORES TEMÁTICOS

- Ingeniería de transporte.
- Transporte vial.
- Redes Neuronales Artificiales
- Modelamiento de la Demanda de Transporte

INTRODUCCIÓN

El uso de redes neuronales en transporte ya viene siendo estudiado de hace dos décadas, concentrándose principalmente en la última década. La motivación detrás de dicha experimentación se basa en los logros que las redes han mostrado en otros campos, siendo herramientas ya reconocidas por su capacidad para modelar fenómenos complejos, lo que motiva un esfuerzo por intentar encontrar en transporte espacios donde explotar esa capacidad.

Aplicar redes neuronales a la modelación de la demanda de transporte es un tema aún más específico, lo que implica que existe una literatura reducida en esta materia. Los escasos trabajos han tomado lo que podría considerarse la ruta natural, es decir, intentar comparar un modelo conocido y probado con una especificación intuitiva de modelo neuronal a fin de detectar si las redes tienen mayor capacidad de predicción. En principio los resultados son promisorios, pues sin mucha sofisticación algunos de estos modelos neuronales han sido capaces de al menos igualar las capacidades predictivas de los métodos tradicionales, por lo menos en problemas sencillos.

Como consecuencia de la forma aún no completamente comprendida en que las redes construyen una solución para un mapa de datos dado, es usual que en estas experiencias de modelación de demanda se plantee un modelo neuronal sin analizar sus propiedades, potenciales y limitaciones o que se lo calibre sin describir las características de dicho proceso (lo cual indica que no se le da mucha importancia al proceso mismo). Esta situación coexiste con el uso de un vocabulario heredado del campo de la inteligencia artificial, cuna de las redes neuronales, que muchas veces tiende a confundir por sus referencias a conceptos y mecanismos biológicos que con poca rigurosidad se aplican al grafo neuronal, implícitamente asumiéndose que el modelo posee ciertas capacidades que en realidad puede no tener.

Como Daugherty (1995) señala, tales situaciones evidencian que probablemente no hay una cabal comprensión de la herramienta, lo que llama a estudiar con mayor detención el planteamiento de modelos neuronales y su uso. Es justamente ésta la motivación de la presente tesis.

Teniendo en mente los elementos de diseño y sus consecuencias, es posible analizar críticamente los modelos neuronales existentes y buscar nuevas formas de abordar la estimación de la demanda de transporte. Más específicamente, se desea estudiar la posibilidad de diseñar un modelo predictivo de generación de viajes de la demanda de transporte, entendido de que un modelo de generación de viajes con redes neuronales podría representar mejor el fenómeno de los viajes.

A diferencia de otros trabajos existentes que exploran el uso de las redes neuronales en demanda de transporte, esta tesis no tiene como objetivo comparar el desempeño de éstas con el de métodos tradicionales de estimación de demanda, sino que se centra en

estudiar cómo diseñar redes neuronales capaces de reproducir adecuadamente el fenómeno de los viajes y en los elementos metodológicos que llevan a una buena calibración y correcta aplicación de un modelo neuronal.

El trabajo se divide en Seis partes. En la primera hace mención a las Generalidades de la presente tesis. En el capítulo II, llamado "Estado del Arte", se describen una breve historia de las redes neuronales al pasar de los años y hace mención de las redes neuronales relacionadas en la predicción. Luego, en "III. Marco Teórico", se presentan los diferentes Modelos y clasificaciones de las redes Neuronales para tener una visión general de cuál podría ser de utilidad para la creación del modelo neuronal, además menciona algunos modelos de demanda de transporte y para terminar el capítulo se hace mención de redes neuronales en el are del transporte. En capítulo IV, "Modelo Neuronal de Generación de Viajes", el cual habla la metodología y el tipo de red neuronal a utilizar, en el capítulo V, "Implementación del Modelo", se describe como se implementó la Red Som y la Red backpropagation en la herramienta computacional Matlab simulando datos reales de la ciudad de Tarapoto, para luego obtener resultados, para finalmente entregar conclusiones y recomendaciones en el capítulo VI.

TABLA DE CONTENIDOS

RESUMEN	i
ABSTRACT	i
DEDICATORIA	ii
AGRADECIMIENTOS	iii
DESCRIPTORES TEMÁTICOS	iv
INTRODUCCIÓN	v
TABLA DE CONTENIDOS	vii
INDICE DE FIGURAS	xi
ÍNDICE DE TABLAS	xii
CAPÍTULO I	1
1. Introducción	2
1.1. Antecedentes	2
1.2. Formulación del Problema	2
1.3. Objetivos de la investigación	3
1.3.1. Objetivo General	3
1.3.2. Objetivos Específicos	3
1.4. Hipótesis	3
1.5. Justificación	3
1.6. Alcance	4
1.7. Limitaciones	4
1.8. Tipo de investigación	4
1.9. Diseño de la investigación	4
CAPÍTULO II	5
2. Estado del Arte	6
2.1. Breve Historia de las Redes Neuronales	6
2.2. Redes Neuronales en el tiempo.	7
CAPÍTULO III	11
3. Marco Teórico	12
3.1. Introducción	12
3.2. Modelos y Clasificación de Redes Neuronales Artificiales	12

3.2.1.	Modelos de Redes Neuronales Artificiales _____	12
3.2.2.	Clasificación de la Redes Neuronales Artificiales _____	14
3.3.	Redes Neuronales _____	15
3.3.1.	Descripción de Conceptos previos. _____	15
3.3.1.1.	Neurona artificial _____	15
3.3.1.2.	Modelo matemático de una capa de neuronas artificiales _____	16
3.3.1.3.	Funciones de transferencias _____	17
3.3.1.4.	Redes neuronales de varias capas o redes multicapa _____	18
3.3.2.	Clasificación de las Redes Neuronales Artificiales (RNA) en supervisadas y no supervisadas _____	20
3.3.2.1.	Las Redes Neuronales Artificiales de aprendizaje supervisado _____	20
3.3.2.2.	Las Redes Neuronales Artificiales de aprendizaje no supervisado _____	20
3.3.3.	Redes Neuronales Backpropagation _____	21
3.3.3.1.	Introducción. _____	21
3.3.3.2.	Estructura y Aprendizaje de la Red Backpropagation _____	22
3.3.3.3.	Conjuntos de Entrenamiento y Testeo _____	22
3.3.3.4.	Funciones de Error _____	22
3.3.3.5.	Épocas _____	23
3.3.3.6.	Saturación _____	23
3.3.3.7.	Sobre-ajuste _____	24
3.3.3.8.	Aplicaciones de la Red Backpropagation _____	24
3.3.4.	Redes Neuronales Self Organizing Map (SOM). _____	25
3.3.4.1.	Introducción. _____	25
3.3.4.2.	Ideas intuitivas sobre el algoritmo del SOM _____	26
3.3.4.3.	Fundamentos biológicos _____	26
3.3.4.4.	Topologías de Capas de Neuronas _____	27
3.3.4.5.	Las métricas o nociones de distancia entre neuronas _____	27
3.3.4.6.	La noción de vecindad _____	28
3.3.4.7.	Arquitectura de las Redes Neuronales SOM. _____	29
3.4.	Modelación de la Demanda de Transporte _____	30
3.4.1.	Introducción _____	30
3.4.1.1.	El Modelo de las 4 Etapas _____	31

3.4.1.2.	Tiempo y Espacio _____	33
3.4.1.3.	Propósito y Tipología de Usuario _____	34
3.4.1.4.	Visión Agregada v/s Desagregada _____	34
3.4.2.	Modelos de Generación y Atracción _____	34
3.4.3.	Modelos de Distribución _____	36
3.4.4.	Modelos de Partición Modal _____	40
3.4.5.	Modelos de Demanda Integrados _____	43
3.4.6.	Modelos de Demanda Directa _____	44
3.4.7.	Otros Aspectos de la Modelación de la Demanda de Transporte _____	44
3.5.	Redes Neuronales y Transporte _____	44
3.6.	Estudios en Demanda de Transporte _____	48
3.6.1.	Metodología y Prácticas _____	49
3.6.2.	Experimentos de Distribución _____	49
CAPÍTULO IV _____		52
4.	Modelo Predictivo de Generación de Viajes _____	53
4.1.	Fuentes de Información _____	53
4.2.	Metodología para Calibrar y Aplicar el Modelo _____	56
4.2.1.	Etapa I: Mapa de Kohonen y Formación de Conjuntos de Entrenamiento, Validación y Testeo _____	56
4.2.2.	Etapa II: Diseño y Entrenamiento, Validación y Testeo de la red de retropropagación (modelo) _____	56
4.2.3.	Etapa III: Creación y Entrenamiento de la Red _____	57
4.2.4.	Etapa IV: Simulación de la Red _____	57
4.3.	Aprendizaje de las Redes Neuronales Artificiales backpropagation y la regla de la cadena _____	59
4.3.1.	Ejemplo Simple de Aprendizaje supervisado en RNA Backpropagation _____	59
4.3.2.	Formulación matemática de las capas _____	59
4.3.3.	Método del gradiente heurístico en el aprendizaje de la RNA _____	61
CAPÍTULO V _____		67
5.1.	Sintaxis en MATLAB de la RNA Retropropagación o Backpropagation (newff) _____	68
5.2.	Sintaxis en MATLAB de la RNA KOHONEN O SOM (con newsom) _____	69
5.3.	Implementación del Modelo Predictivo _____	69

5.3.1.	Implementación de Red SOM en Matlab	69
5.3.2.	Implementación de Red de Retropropagación (Backpropagation) en Matlab	80
5.4.	Aplicación de la Red de Retropropagacion (Backpropagation)	81
CAPÍTULO VI		92
6.1.-	Conclusiones	93
6.2.-	Recomendaciones	94
BIBLIOGRAFÍA		95

INDICE DE FIGURAS

- Figura N°01.- Clasificación de los modelos de redes neuronales artificiales
- Figura N°02.- Esquema de una neuronal Artificial
- Figura N°03.- Esquema de Modelo Matemático de una capa de neuronas Artificiales
- Figura N°04. Función de transferencia sigmoideal
- Figura N°05.- Función de transferencia tangente sigmoideal hiperbólica
- Figura N°06.- Esquema de redes neuronales de varias capas o redes multicapas
- Figura N°07. Estructura de un mapa autoorganizado (SOM)
- Figura N°08.- Resolución secuencial del modelo de 4 etapas
- Figura N°09: Estructura de la red utilizada para distribución. d_{ij} .
- Figura N°10.- Arquitectura de una Red Neuronal tradicional
- Figura N°11.- Interface de una Red SOM
- Figura N°12. - SOM Topology.
- Figura N°13.- SOM Neighbor Connections
- Figura N°14.- SOM Neighbor Weight Distances
- Figura N°15.- Hits
- Figura N°16. - Weight from Input 1
- Figura N°17.- Weight from Input 2
- Figura N°18.- Weight from Input 3
- Figura N°19. - Weight from Input 4
- Figura N°20.- Weight from Input 5
- Figura N°21. - SOM Weig Pos itionst
- Figura N°22.- Curvas de Entrenamiento, Validación y Test de la Red
- Figura N°23.- Interface de la Neural Network training
- Figura N°24.- Curvas de Entrenamiento, Validación y Test respecto al comportamiento de la norma del gradiente
- Figura N°25.- Gradientes
- Figura N°26.- Gráficos de Regresión entre Output y el valor esperado de la red
- Figura N°27.- Recta de regresión por mínimo cuadrados

ÍNDICE DE TABLAS

Tabla N° 01.- Clasificación de Redes neuronales Artificiales

Tabla N° 02.- Distribución de Trabajos con Redes Neuronales en Transporte

Tabla N° 03.- Modelos Usados

Tabla N° 04.- Resultados de EOD-Tarapoto por Zonas de tránsito.

Tabla N° 05.- Variables independientes y dependientes que tienen poca correlación.

Tabla N° 06.- Resultados de la Red SOM.

Tabla N° 07.- Resultados de la Red SOM y sus respectivas Zonas de tránsito

Tabla N° 08.- Resultados ordenados de menor a mayor por número cluster

Tabla N° 09.- Conjuntos de entrenamiento, validación y testeo.

Tabla N° 10.- Conjuntos de entrenamiento, validación y testeo preparado para la Red de Retropropagación.

Tabla N° 11.- Pesos y Sesgos (Ganancias) de la primera capa

Tabla N° 12.- Comparación de Valor esperado (Viajes Generados) y Output (Viajes generados por la Red)

CAPÍTULO I
GENERALIDADES

1. Introducción

En este capítulo se presenta el planteamiento del problema, objetivos de la investigación, justificación y el tipo de investigación que permitirá guiar sistemáticamente el proceso de investigación. El desarrollo enfatiza en el uso de la metodología de la investigación científica y el enfoque sistémico.

1.1. Antecedentes

En el área de transportes, uno de los primeros estudios de aplicación de Redes Neuronales Artificiales habría sido el de Nakatsuji y Kaku (1989), buscando resolver problemas relacionados con Ingeniería de Tráfico. En seguida, a lo largo de los años 90, el campo de estudios de transporte observó una verdadera explosión en el uso de Redes Neuronales (Dougherty, 1995). La habilidad para trabajar con datos incompletos hace a las Redes Neuronales especialmente atractivas para planificación en países en desarrollo, en los cuales algunos estudios están siendo desarrollados¹.

En el área de planificación y operación de transportes, algunos estudios fueron realizados utilizando Redes Neuronales Artificiales abordando temas que van, desde el comportamiento de los conductores delante de la visualización de la luz amarilla en la aproximación al grupo focal semafórico en intersecciones, hasta el análisis de probabilidad de ocurrencia de accidentes de tránsito a partir de parámetros como el flujo, la concentración y velocidades de los vehículos en las vías (Ivan, 1997). Costa y Markellos (1997) usaron Redes Neuronales Artificiales para desarrollar mecanismos para medición del desempeño del transporte público basados en el concepto de eficiencia de la productividad, usando datos sobre el metro de Londres. De particular interés para el presente estudio, no en tanto, es el trabajo desarrollado por Raia Jr. (2000), que utilizó Redes Neuronales Artificiales para estimar un Índice de Potencial de Viajes a partir de variables de movilidad y accesibilidad, con vistas a la planificación estratégica de transportes. Este modelo fue utilizado como base para el desarrollo de los procedimientos alternativos aquí explorados².

1.2. Formulación del Problema

Los escasos trabajos que han intentado comparar un modelo conocido y probado con un modelo neuronal a fin de detectar si las redes tienen mayor capacidad predictiva han obtenido buenos resultados, pues sin mucha sofisticación algunos modelos neuronales han sido capaces al menos igualar las capacidades predictivas de los métodos tradicionales, estas experiencias de modelación de demanda si bien han obtenido buenos resultados plantearon un modelo neuronal sin analizar sus propiedades, potenciales y limitaciones o que se lo calibre sin describir las características de dicho proceso lo cual se entiende que no se le ha dado mucha importancia al proceso mismo, esto hace parecer que los modelos tienen ciertas capacidades que en realidad no las poseen.

¹ NAKATSUJI, T. y T. KAKU (1989) Application of neural network models to traffic engineering problems. Proceeding of Infrastructure Planning 12, pp.297-304.

² RAIA JR., A.A. (2000) Acessibilidade e mobilidade na estimativa de um índice de potencial de viagens utilizando redes neurais artificiais. São Carlos. 202 p. Tese (Doutorado) – Escola de Engenharia de São Carlos, Universidad de São Paulo.

Con esto se pretende responder a la siguiente pregunta: ¿Cómo contar con un modelo de demanda de transporte en redes viales urbanas aplicando Redes Neuronales y cuál es la metodología a seguir cuando se desea postular un modelo Neuronal de Demanda?

1.3. Objetivos de la investigación

1.3.1. Objetivo General

Desarrollar un modelo de demanda de transporte en redes viales urbanas aplicando Redes Neuronales.

1.3.2. Objetivos Específicos

- Comprender de manera más precisa qué implicancias tienen distintas arquitecturas de red, cuáles son las capacidades de un modelo particular y dónde están sus limitaciones.
- Estudiar las variables de diseño que determinan un modelo y los elementos metodológicos relacionados con el tratamiento de los datos, la calibración y aplicación de las redes.
- Estimación del total de los viajes generados a partir de variables agregadas de los distritos de Tarapoto, La Banda de Shilcayo y Morales.
- Proponer una metodología de entrenamiento de la red neuronal para el caso estudiado.
- Diseñar la Red Neuronal para la solución del problema planteado.

1.4. Hipótesis

Utilizando las redes neuronales de Mapas auto-organizados de Kohonen y una red de retropropagación se desarrolla un Modelo Neuronal de la demanda de Transporte.

1.5. Justificación

Según TAO Yang, se han utilizado redes neuronales recurrentes para la predicción a corto plazo del tráfico en una carretera, a fin de prevenir congestiones y tener un control del acceso a la autopista. Para esto se utilizan datos estimados de otros días con propiedades similares; los mejores resultados se obtuvieron con una red multi - recurrente, y se pudo comprobar que las redes neuronales resolvieron este tipo de predicción y obtuvieron mejores resultados que los métodos estadísticos convencionales³.

La ventaja que tienen las Redes Neuronales Artificiales es que no necesitan de conocimientos de especialistas para tomar decisiones; ellas se basan únicamente en los ejemplos históricos que les son dados: no es necesario informar porqué tal situación resultó en tal decisión en el pasado, o porqué tal decisión resultó en tal consecuencia. Dependiendo del tipo de problema al cual son

³ Un interesante modelo para la predicción del tráfico. TAO, Yang, A Neural Network Model for Traffic Prediction in the Presence of Incidents via Data Fusion, Universidad de Wisconsin, Julio de 2004, http://homepages.cae.wisc.edu/~yang/incident%20impact_final.pdf

sometidas, las Redes Neuronales Artificiales han presentado un desempeño considerado superior a los métodos estadísticos utilizados para el mismo fin, por este motivo se desea realizar un modelamiento con redes neuronales para determinar la demanda de tráfico existente y proyectar una demanda a futura, la cual nos ayude a tomar decisiones a corto plazo.

1.6. Alcance

El alcance de la presente investigación abarca las entidades Regionales y locales en el sector Transporte.

1.7. Limitaciones

El presente trabajo de investigación tiene alcance de aplicación a instituciones públicas y privadas que requiera predecir la demanda de transporte para la toma de decisiones en los proyectos del sector transporte.

1.8. Tipo de investigación

El tipo de investigación es aplicada. El objetivo es obtener un modelo que pronostique la demanda de tráfico vehicular para facilitar la toma de decisiones en la red vial de tráfico estudiada.

El presente trabajo de investigación empleará los siguientes métodos científicos

- **Descriptiva:** Se considera un estudio de caso y por otra parte considerarla como una investigación de campo.
- **Inductivo - deductivo:** Estos métodos se aplicarán con el propósito de establecer las conclusiones y generalizar los resultados de la investigación.

1.9. Diseño de la investigación

Será de tipo experimental, descriptivo y correlacional - causal.

CAPÍTULO II
ESTADO DEL ARTE

2. Estado del Arte

2.1. Breve Historia de las Redes Neuronales

La estructura del cerebro, formada por millones de neuronas interconectadas, es una arquitectura que uno podría intentar emular directamente. En principio, si el cerebro puede realizar cosas tan complejas y maravillosas basadas sólo en diminutos elementos de tanta simpleza, es razonable pensar que las habilidades del cerebro están dadas por su configuración de red. Esta hipótesis es la que en un comienzo impulsó a investigadores a desarrollar bajo esta filosofía la idea de inteligencia artificial. En 1943 el neurobiólogo Warren McCulloch y el estadístico Walter Pitts publicaron su trabajo que con este enfoque motivaron el nacimiento de esta área del conocimiento. En ese entonces la idea de neurona y de las conexiones entre neuronas se manejaba con circuitos eléctricos muy simples.

En 1956 se celebra la primera conferencia internacional sobre inteligencia artificial y temas afines, en donde el concepto de red neuronal alcanza su legitimidad y el tema se hace ampliamente conocido en el ámbito científico. Hasta mediados de los 60's hubo gran financiamiento y esfuerzo dedicado a las redes neuronales. Algunos de los investigadores de esa época más renombrados fueron Rosenblatt y su perceptrón en 1957 y Widrow y su Adaline en 1958. Pero en 1969 Minsky y Papert publican su libro "Perceptrón", lo que fue un fuerte golpe a las redes neuronales de su tiempo. Este libro fue un trabajo crítico sobre los perceptrones y su principal conclusión es que estos elementos no eran de interés, pues eran incapaces de resolver ciertos problemas sumamente simples. Luego de esto, el estudio de las redes neuronales decayó fuertemente y sólo unos pocos investigadores pudieron continuar sus estudios en esta área. No fue sino hasta 1982 que las redes vuelven a aparecer con fuerza, cuando Hopfield enseña sus trabajos mostrando nuevamente el potencial de estas estructuras. Ciertos cambios estructurales a las redes acompañados de demostraciones matemáticas permitían confiar en que se podía superar las dificultades encontradas por Minsky y Papert, creando un renovado entusiasmo.

Las redes neuronales a partir de entonces han retomado su propio lugar en el ámbito académico y en la ingeniería y ya no están necesariamente ligadas a la búsqueda de inteligencia artificial. Diversas aplicaciones han sido desarrolladas en muy diversos ámbitos, como el reconocimiento de imágenes o la predicción de mercados bursátiles. En particular el transporte se ha beneficiado también de esta herramienta, aunque en cuanto a modelación de demanda está aún en una fase muy experimental.

2.2. Redes Neuronales en el tiempo.

En los años 80, ocurrieron dos acontecimientos cruciales en la evolución de los estudios sobre series temporales. Por un lado, el incremento en la potencia de los computadores personales, permitió el estudio de series temporales mucho más largas, la aplicación de algoritmos más complejos, y la visualización interactiva tanto de los datos como de los resultados. El segundo hecho fue el desarrollo de las técnicas de aprendizaje automático (inteligencia artificial) y concretamente de las Redes Neuronales Artificiales.

Las Redes Neuronales Artificiales (RNAs) son modelos matemáticos inspirados en la organización y el funcionamiento de las neuronas biológicas. Existen numerosas variantes de RNAs que están relacionadas con la naturaleza de la tarea que se ha asignado. De la misma manera, también existen distintas variaciones sobre cómo modelar la neurona; en algunos casos se asemejan mucho a las neuronas biológicas mientras que en otros, los modelos son muy diferentes.

Desde el punto de vista matemático, las RNAs se pueden considerar como aproximadores universales de funciones. Esto significa que pueden automáticamente aproximar la función que mejor se ajuste a los datos, permitiendo de esta manera extraer relaciones cuando las funciones son muy complejas. Además las RNAs son intrínsecamente no lineales [Rumelhart-1986], lo cual implica no sólo que pueden estimar correctamente funciones no lineales, sino que también pueden extraer elementos no lineales de los datos, una vez extraídos los términos lineales.

Algunos métodos estadísticos de series temporales tienen limitaciones debidas a la forma en que los modelos son estimados, por esta razón, la estimación de muchos tipos de modelos de series temporales requieren la intervención y supervisión humana. Además, la mayoría de modelos estadísticos deben ser reestimados periódicamente cuando se dispone de nuevos datos. Por el contrario, la estimación con RNAs puede ser automatizada [Hoptroff-1993], y no es necesario revisar los modelos puesto que las redes aprenden de forma incremental.

En cuanto a la predicción de demanda, las RNAs se presentan como una variante de gran aplicabilidad a la solución de problemas de predicción, debido a su capacidad intrínseca como aproximadora de funciones matemáticamente desconocidas y clasificadoras de patrones. Adicionalmente, su alta inmunidad al ruido, capacidad de auto-adaptación y tolerancia a fallos, las convierten en una herramienta con ventajas frente a los métodos convencionales de predicción [Lippmann-1987], [Hammerstrom-1993], [Haykin-1999].

Por lo general, al hablar de inteligencia artificial aplicada a predicción de demanda, se piensa en Redes Neuronales dadas las ventajas presentadas al emplearlas dentro de esquemas de predicción. No obstante, otros esquemas de inteligencia artificial tales como sistemas expertos [Ho-1990], [Sharaf-1993], o esquemas híbridos pueden ser encontrados en la bibliografía referente al tema. Entre estos esquemas híbridos se pueden encontrar combinaciones de Redes neuronales con otras técnicas, como por ejemplo, sistemas de lógica difusa [Srinivasan- 1996],[Shan-1997] o algoritmos genéticos [Heny-1998].

Algunos textos de consulta como [Granger-1989] y [Hanke-1996], para predicción de demanda a nivel general, así como [Freeman-1993] y [Haykin-1999] para redes neuronales, son una buena base de estudio para el trabajo de conceptualización inicial que requiere el desarrollo de esta investigación.

En cuanto a la forma de abordar el problema, diversas metodologías son encontradas en la bibliografía del tema de predicción de demanda con redes neuronales. La heurística presentada en el problema, debido a que la mayoría de implementaciones con redes neuronales surgen de procesos empíricos apoyados lógicamente en una teoría básica, hace que sean diversas las formas de enfrentar el problema de predicción; ejemplos de esto lo constituyen artículos como [Lee- 1991], [AlFuhaid-1997], [Choueiki-1997] o [Rewagad-1998]. Algunos aspectos claves, que pueden servir como base para el desarrollo de una metodología para la predicción de demanda en el corto plazo, en lo concerniente al manejo de variables climatológicas, se presentan en [Xiao-1995], [Wang-1998] y [Kandil- 1999]; otras alternativas interesantes, entre otras, como la de [Park-1993] consistente en la predicción de demanda a partir de la variación del consumo con respecto a un perfil de demanda patrón, o el empleo de redes neuronales recurrentes como en [Srivastava-1997] o el entrenamiento dinámico de la red (durante su etapa de producción) como en [Morioka-1993], amplían el espectro de posibilidades investigativas en el tema.

En los últimos años las redes neuronales han demostrado ser un método muy eficaz de modelado matemático. En particular, son muy adecuadas para el reconocimiento de patrones, análisis de series de tiempo y control. Más recientemente se ha prestado atención en su aplicación a la predicción de series temporales. Predicción de las condiciones de tráfico pueden ser considerados como parte de estas aplicaciones de predicción desafiantes.

Desde el punto de vista de gestión del tráfico urbano, es de particular interés ya que las redes neuronales pueden ser integradas en los sistemas de control de tráfico urbano. A través de los años diferentes técnicas de predicción se han utilizado para proporcionar predicciones como un soporte para el control de decisiones. La primera generación del sistema UTC, incluyendo Transit (Robertson, 1989) se basa únicamente en los datos históricos del pasado ya que es un sistema fuera de línea . Con el fin de hacer el sistema más adaptable en la segunda generación tales como el sistema Scoot (Hunt et al . , 1981) , utiliza medidas de tráfico actuales para corregir la desviación del tráfico desde el patrón histórico promedio. Sin embargo todavía se basa en datos históricos y media de las condiciones de Tráfico. La tercera generación, incluyendo Opac (Gartner et al., 1991), ProDyn (Henry et al., 1983) y Cronos (Boillot et al. , 1992), no utiliza los datos históricos, más bien predice condiciones basadas únicamente en las mediciones de tráfico actuales⁴.

El diseño de la red de transporte es un problema complejo no convexo. (Newell, 1979; Baaj and Mahmassani, 1991). Esto es usualmente formulado como un problema de optimización no lineal con ambas variables discretas y continuas y limitaciones. Los mejores y más eficientes métodos de solución están basados en procedimientos heurísticos pero sus aplicaciones se limitan principalmente a casos de prueba o redes de la vida real de tamaño pequeño. En años recientes, la evolución de la investigación operacional y el creciente

⁴ AN URBAN TRAFFIC FLOW MODEL INTEGRATING NEURAL NETWORKS - CORINNE LEDOUX- I.N.R.E.T.S, 2, Avenue du Général Malleret Joinville, 94114 Arcueil Cedex, France- Received 10 September 1996; in revised form 23 July 1997.

poder de una maquina computacional han producido un gran y renovado interés por este problema. Una revisión global sobre diseño de rutas, ajustes de frecuencia, horarios de línea de tránsito y sus combinaciones es propuesto por Desaulniers and Hickman (2007), Guihaire and Hao (2008), and Kepaptsoglou and Karlaftis (2009).

Entre las obras más notables en esta materia, hay que destacar Ceder and Israeli (1993), Baaj and Mahmassani (1995), Carrese and Gori (2002) and Lee and Vuchic (2005). Ceder and Israeli (1993) propone un procedimiento de diseño de red de tránsito basado en un enfoque de programación matemática. El primer paso genera un conjunto muy largo de rutas factibles conectando cada nodo a todos los otros. Luego, el sistema crea el mínimo de subconjunto de rutas resolviendo un problema conjunto cubierto y selecciona el subconjunto más adecuado para aplicar un análisis multiobjetivos. Baaj and Mahmassani (1995) uso un algoritmo heurístico de inteligencia artificial para la generación de rutas. Este algoritmo selecciona un número dado de pares de nodo de demanda alta y construye un esqueleto inicial de la red mediante la conexión de estos pares de nodos a través de los caminos más cortos. El esqueleto es entonces progresivamente ampliado a las rutas de acuerdo a una estrategia de selección de nodo que refleja las diferentes soluciones de compromiso entre el rendimiento, medidas y los costos de usuarios y operadores. Carrese and Gori (2002) propuso un modelo de diseño de red de buses para el desarrollo de un sistema de tránsito jerárquico. El procedimiento de diseño es subdividido en 2 fases, niveles de la red de autobuses integrados jerárquicamente articulada en líneas expresas, principales y secundarias.

Lee and Vuchic (2005) propuso un procedimiento interactivo, empezando de un conjunto inicial de rutas compuestas por el camino más corto para todos los pares origen - destino, trata de mejorarlo mediante la realineación de las rutas y la eliminación de las menos eficientes, teniendo en cuenta explícitamente el cambio de distribución modal⁵.

En las décadas pasadas, la investigación substancial ha sido hecha para desarrollar sistemas de control de tráfico urbano de respuesta (Sims, 1979; Hunt et al., 1981; Gartner et al., 1991; Boillot and Papageorgiou, 1992; Gartner et al., 1995). Uno de los principales elementos en estos sistemas es un modelo de predicción de flujo, la cual pronostica los flujos de tráfico en el futuro basado en las series temporales de datos de tráfico medidos de la red. Para la predicción a mediano plazo, Cremer y Keller emplean el método estimación de mínimos cuadrados recursivos para la predicción de los flujos de tráfico. Este método fue modificado y usado en el sistema SCATS (Luk, 1989)⁶.

El pronóstico o predicción de parámetros de tráfico o medidas de desempeño, tiene que ver con un horizonte a corto plazo de predicción (ventana de tiempo en el que las condiciones del tráfico se prevén en el futuro) en general, debido a su naturaleza cambiante dinámica. Ha habido varios estudios de investigación y modelos para la predicción de tiempo de viaje

⁵ Transit network design: A procedure and an application to a large urban area- Ernesto Cipriani , Stefano Gori, Marco Petrelli- University of Roma TRE, Civil Engineering Department, 62, Via Vito Volterra, 00146 Rome, Italy.

⁶ Urban traffic flow prediction using a fuzzy-neural approach - Hongbin Yin, S.C. Wong b, Jianmin Xu , C.L. Wong - College of Traffic and Communication, South China University of Technology, Guangzhou, PR China Department of Civil Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong, PR China.

para autopistas. Investigación y modelos de arterias urbanas similares todavía se está quedando atrás. Vlahogianni et al. (2004) da una discusión crítica sobre las técnicas de previsión de tráfico a corto plazo, tanto para las autopistas y arterias urbanas de diferentes tipos de implementaciones como ATIS y cajeros automáticos. La discusión de las metodologías para la previsión del tráfico identifica dos grandes categorías: las técnicas paramétricas y no paramétricas. Entre otras conclusiones, esta revisión se afirma que las técnicas de modelado no paramétricas como Redes Neuronales Artificiales (ANN) son de las más prometedoras para los problemas de previsión de tráfico y dan modelos robustos y precisos. Zwet y Rice (2004) propusieron un esquema de predicción mediante regresión lineal con coeficientes variables en el tiempo. Los coeficientes se someten a la hora del día y el tiempo hasta el inicio de un vehículo en una sección. Los coeficientes variables muestran el comportamiento dinámico del tráfico que es capturado por otras técnicas paramétricas y no paramétricas avanzadas. Ishak y Al- Deek (2002) propusieron un enfoque de series de tiempo no lineal para hacer predicciones de corto plazo de la velocidad con el perfil más reciente de velocidad en cada estación de detector de lazo y luego matemáticamente encontrado tiempo de viaje de Velocidad. Este estudio se basa en datos de autopistas y también presenta un análisis estadístico para determinar parámetros como el índice de congestión, horizonte rodante, horizonte de predicción y sus términos de interacción con el índice de congestión. Stathopoulos y Karlaftis (2003) propusieron un modelo de espacio de estados de series temporales multivariantes para la predicción del flujo de tráfico en las avenidas y en comparación con otras técnicas de series como ARIMA. Señalaron que las diferentes especificaciones del modelo son apropiadas para diferentes periodos de tiempo del día. Wu et al. (2004) utilizaron Apoyo Vector de regresión para predecir el tiempo de viaje para carreteras en Taiwán. Chien y Kuchipudi (2003) aplicaron Kalman Filtering para predecir enlace y tiempo de viaje basado en las rutas en base a datos de series temporales recogidos a través de las tecnologías inalámbricas en las autopistas. Skabardonis y Geroliminis (2005) presentan los modelos basados en la teoría de la onda cinemática para estimar los tiempos de viaje en las calles arteriales en tiempo real basado en los datos comúnmente proporcionadas por los detectores de lazo sistema y ajustes de la señal . Los modelos proporcionan predicciones precisas, pero no lo suficientemente genérica; se requieren datos para la ubicación específica de estudio⁷.

Hay estudios significativos relacionados con la aplicación de diferentes tipos de RNAs. Mark y Sadek (2004) propusieron un modelo de RNAs para autopistas para predecir el tiempo de viaje experiencial en condiciones transitorias de tráfico incluyendo incidentes. Ellos encontraron que la velocidad parece ser la variable de entrada más influyente para la predicción de tiempo de viaje. Un tipo especial de red neuronal recurrente (RNN) denominado Espacio de Estado Red Neural (SSNN) fue utilizado por Lint et al. (2002) para autopistas n la predicción de tiempo de viaje utilizando el flujo y la velocidad. El estudio mostró que el análisis de los estados internos y configuraciones de peso de SSNN podría desarrollar un modelo interno que está estrechamente relacionado con el proceso de tráfico subyacente⁸.

⁷ Modeling Arterial Travel Time with Limited Traffic Variables using Conditional Independence Graphs & State-Space Neural Networks- Ghassan Abu-Lebdeh, Ajay K. Sing- Department of Civil Engineering, American University of Sharjah, Sharjah, UAE Jacobs Engineering Group, Huston, TX, USA

⁸ Modeling Arterial Travel Time with Limited Traffic Variables using Conditional Independence Graphs & State-Space Neural Networks- Ghassan Abu-Lebdeh, Ajay K. Sing- Department of Civil Engineering, American University of Sharjah, Sharjah, UAE Jacobs Engineering Group, Huston, TX, USA

CAPÍTULO III
MARCO TEÓRICO

3. Marco Teórico

3.1. Introducción

Dado que el tema de esta tesis es la aplicación de las redes neuronales a la modelación de la demanda de transporte, la revisión bibliográfica se separa en Cuatro partes. En la primera se describirán los modelos de Redes neuronales Existentes y su Clasificación, en la segunda sección se revisará el material disponible referente a las redes neuronales, su funcionamiento y fundamentos. De la amplia variedad de tipos de red neuronal existente, sólo las dos que se utilizan en el resto de la tesis (redes de retropropagación y mapas auto-organizados de Kohonen) son incorporadas en esta revisión, en la tercera parte se reportan técnicas tradicionales de modelación de la demanda, las que pueden ser vistas en mayor profundidad en Juan de Dios Ortúzar (2000). Por último, en la cuarta parte se muestran aplicaciones previas de esta herramienta matemática al transporte y en particular al problema de la demanda de transporte.

3.2. Modelos y Clasificación de Redes Neuronales Artificiales

3.2.1. Modelos de Redes Neuronales Artificiales

Existen multitud de modelos de Redes Neuronales Artificiales. De ellos, unos cincuenta son medianamente conocidos y una quincena bastante populares. Como es fácil comprender, se precisa de una clasificación de los modelos para establecer algo de orden en su estudio. En la Fig. 6 se muestra una posible clasificación atendiendo al tipo de aprendizaje (supervisado, no supervisado, híbrido y reforzado) y a la topología de la red (redes realimentadas y redes hacia-adelante). El libro de Wasserman (1989) constituye una correcta y sencilla introducción a estos modelos. La clase de modelos más importante la constituyen las redes hacia-adelante y de aprendizaje supervisado. A ella pertenecen el perceptrón, la adalina y el perceptrón multicapa. El perceptrón multicapa es el modelo neuronal más conocido y empleado en la práctica. Suele emplearse junto al aprendizaje de retropropagación (backpropagation), por lo que habitualmente se denomina simplemente red de retropropagación o BP (Rumelhart, 86). Su importancia se debe a su potencia y generalidad, pues se ha demostrado que constituye un aproximador universal de funciones (Hornik, 1989), lo que hace de él uno de los modelos más útiles en la práctica. La operación de este sistema se relaciona con la regresión no lineal.

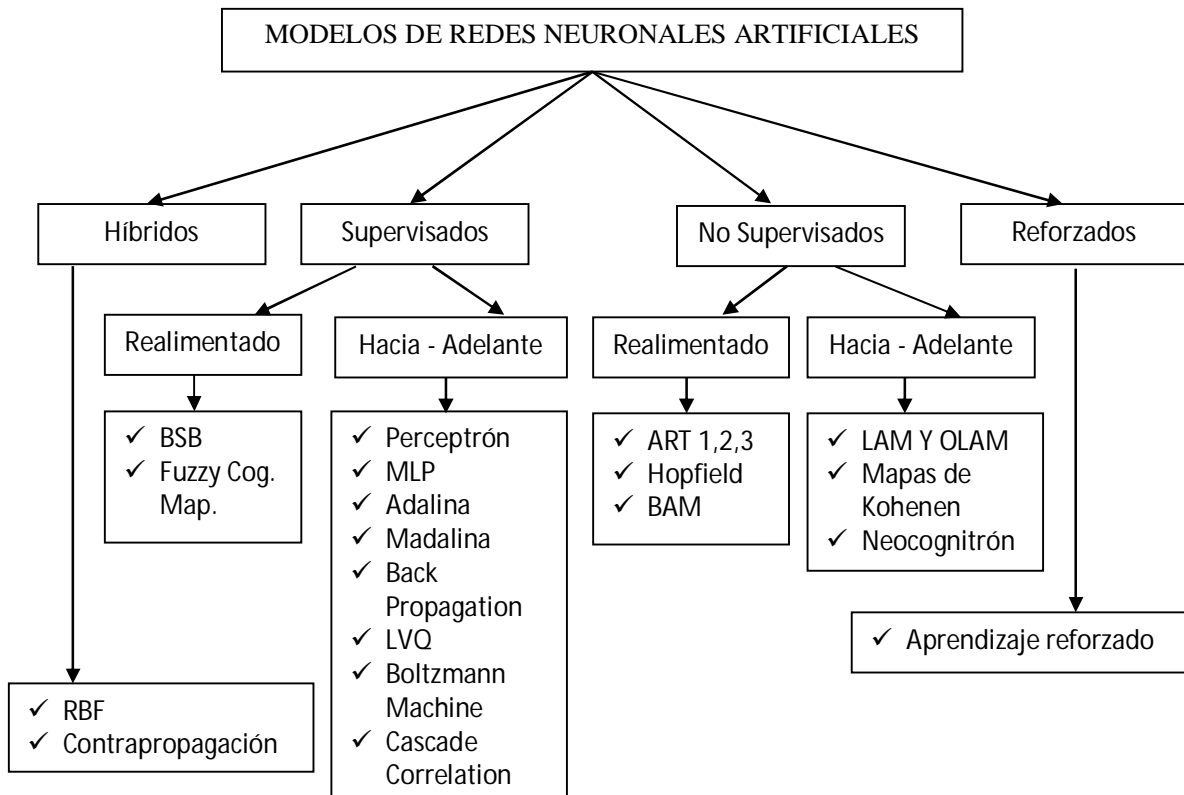


Figura N°01.- Clasificación de los modelos de redes neuronales artificiales

De entre todos los modelos no supervisados se pueden distinguir el ART, neocognitrón y las redes PCA (Wasserman, 1989). No obstante, los mapas autoorganizados o SOFM (Self-Organizing Feature Maps) (Kohonen, 1990) son posiblemente el más popular de todos. Este sistema neuronal reproduce el tipo de mapas sensoriales existentes en el córtex cerebral. Los SOFM, en esencia, realizan la proyección no lineal de un espacio multidimensional de entrada sobre las dos dimensiones del mapa, preservando los rasgos originales más importantes. Por este motivo se emplea con frecuencia en la visualización de complejos conjuntos de datos.

Uno de los modelos realimentados más conocidos y estudiados es el modelo de Hopfield (Hopfield, 1982). Consiste en una red completamente interconectada, de tipo autoasociativo y recurrente, que en su versión discreta se utiliza como memoria asociativa, y en su versión continua se aplica a la resolución de problemas de optimización. Se ha establecido su equivalencia con los modelos de spinglass procedentes de la mecánica estadística, por lo que sobre él se está realizando un gran trabajo teórico (Domany, 1991).

Finalmente, uno de los modelos híbridos más destacables, es el de funciones de base radial o RBF (Radial Basis Functions). Se estructura en tres capas, con la intermedia de tipo autoorganizado de nodos gaussianos, y la segunda supervisada. Un excelente introducción al RBF puede encontrarse en (Hush, 1993).

Estos modelos, junto a muchos otros, se encuentran ampliamente explicados en (Hertz, 1991; Wasserman, 1989; Hecht-Nielsen, 1991; Hush, 1993), referencias a las que remitimos al lector interesado en ampliar los detalles.

3.2.2. Clasificación de la Redes Neuronales Artificiales

La clasificación de las redes neuronales artificiales que se presenta en este capítulo es una simple descripción de las diferentes Redes Neuronales Artificiales más comunes y frecuentes en la mayoría de los simuladores software de sistemas de computación neuronal. La selección de una red se realiza en función de las características del problema a resolver. La mayoría de éstos se pueden clasificar en aplicaciones de Predicción, Clasificación, Asociación, Conceptualización, Filtrado y Optimización. Los tres primeros tipos de aplicaciones requieren un entrenamiento supervisado.

Tabla N° 01.- Clasificación de Redes neuronales Artificiales

Modelo de RNA	Diseñador	Características	Año	Tipo
Adaline y Madaline	Bernard Widrow	Técnicas de Adaptación para el Reconocimiento de Patrones	1960	Predicción
Adaptive Resonance Theory Networks (ART)	Carpenter, Grossberg	Reconocimiento de Patrones y Modelo del Sistema Neuronal. Concepto de Resonancia Adaptativa	1960-1986	Conceptualización
Back-Propagation	Rumelhart y Parker	Solución a las limitaciones de su red predecesora el Perceptron	1985	Clasificación
Bi-Directional Associative Memory (BAM) Networks	Bart Kosko	Inspirada en la red ART.	1987	Asociación
The Boltzmann Machine	Ackley, Hinton y Sejnowski	Similar a la red Hopfield	1985	Asociación
Brain-State-in a Box	James Anderson	Red Asociativa Lineal	1970-1986	Asociación
Cascade-Correlation-Networks	Fahhman y Lebiere	Adición de nuevas capas ocultas en cascada	1990	Asociación
Counter-Propagation	Hecht-Nielsen	Clasificación Adaptativa de Patrones	1987	Clasificación
Delta-Bar-Delta (DBD) Networks	Jacobb	Métodos Heurísticos para Acelerar la Convergencia	1988	Clasificación
Digital Neural Network Architecture (DNNA) Networks	Neural Semiconductor Inc.	Implementación Hardware de la función Sigmoid	1990	Predicción
Directed Random Search (DRS) Networks	Maytas y Solis	Técnica de valores Random en el mecanismo de Ajuste de Pesos	1965-1981	Clasificación
Functional-link Networks (FLN)	Pao	Versión mejorada de la red Backpropagation	1989	Clasificación
Hamming Networks	Lippman	Clasificador de vectores binarios utilizando la Distancia Hamming	1987	Asociación
Hopfield Networks	Hopfield	Concepto de la red en términos de energía	1982	Optimización
Learning Vector Quantization (LVQ) Networks	Kohonen	Red Clasificadora.	1988	Clasificación
Perceptron Networks	Rosenblatt	Primer modelo de sistema Neuronal Artificial.	1950	Predicción
Probabilistic Neural Network (PNN)	Spetcht	Clasificación de Patrones utilizando métodos estadísticos	1988	Asociación
Recirculation Networks	Hinton y McClelland	Alternativa a la red Backpropagation	1988	Filtrado
Self-Organizing Maps (SOM)	Kohonen	Aprendizaje sin supervisión	1979-1982	Conceptualización
Spatio-Temporal-Pattern Recognition (SPR)	Grossberg	Red clasificadora Invariante en el espacio y tiempo.	1960-1970	Asociación

3.3. Redes Neuronales

3.3.1. Descripción de Conceptos previos.

3.3.1.1. Neurona artificial

Las neuronas biológicas tienen esencialmente tres componentes: el cuerpo de la célula, las dendritas que actúan como canales de recepción de señales que vienen de otras neuronas y el axón es el canal de emisión de señales de una neurona. El punto de unión de una dendrita de una célula con el axón de otra, se denomina sinapsis. En este punto de unión, que es entre membranas celulares, se produce intercambios de sustancias químicas y por ende reacciones químicas e impulsos eléctricos, es decir todo un proceso complejo de transferencia de información⁹.

Modelo matemático de una neurona artificial

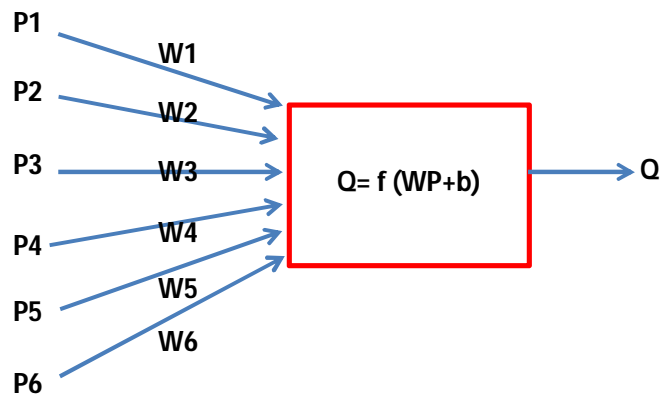


Figura N°02.- Esquema de una neuronal Artificial

El vector $p = [p_1, p_2, \dots, p_n]$ representa a un conjunto de n señales p_k , donde cada una de ellas ingresa a la neurona por la correspondiente dendrita; a esta última le está asociado un número w_k denominado peso. Al conjunto de las n dendritas le corresponde el vector de pesos $W = [w_1, w_2, \dots, w_n]$. Cuando un vector de señales p que llega a la neurona el proceso que sigue dentro del cuerpo de la célula, se representa mediante la suma de los productos de cada señal por su peso, más un valor b denominado ganancia o sesgo de la neurona. Todo esto se representa por la transformación afín:

$$W_p + b = w_1 p_1 + w_2 p_2 + \dots + w_n p_n + b$$

Interpretación del producto $W_p + b$

Esta expresión matemática trata de representar lo que ocurre en una neurona biológica. Por ejemplo una persona que conduce un vehículo, tiene el sentido de la visión en estado de mayor alerta en comparación a los del oído o del gusto. La vista recibe información relativa a las características de la pista, que el cerebro lo procesa y emite una respuesta, que se traduce en reflejos orientados a los músculos de los brazos para

⁹ Universidad Nacional de Ingeniería. FIIS-FIC. Curso de Capacitación: "Redes Neuronales y Simulación". Proyecto FINCYT: Redes Neuronales y Simulación de Monte Carlo para el estudio del concreto de alta resistencia. Código PIAP-1-P-676-13. Pedro C. Espinoza H.

controlar el timón y al de los pies para el control de la caja de cambios y el freno. En este proceso unos datos son acrecentados frente a otros que son disminuidos o anulados. Si $p = [p_1, p_2, \dots, p_n]$ representa todos los datos que recibe el conductor, los pesos $W = [w_1, w_2, \dots, w_n]$, representan la intensidad con que son tomados en cuenta cada uno de los ellos. La ganancia o sesgo b , podría interpretarse como una reacción de la neurona para incentivar, disminuir o anular el valor final del producto W_p .

Finalmente, el valor de $W_p + b$ es transformado mediante una función de transferencia $f(s)$ de la neurona, dando lugar a la respuesta final de la misma: $q = f(W_p + b)$.

3.3.1.2. Modelo matemático de una capa de neuronas artificiales

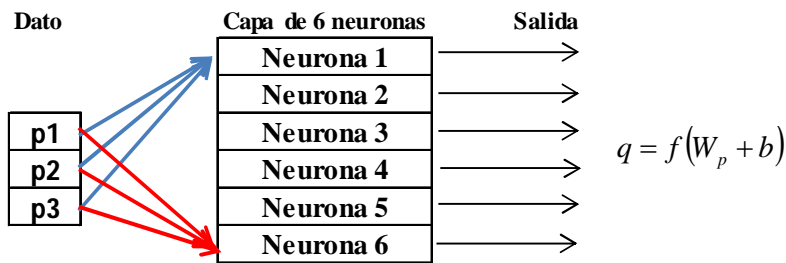


Figura N° 03.- Esquema de Modelo Matemático de una capa de neuronas Artificiales

Una capa de neuronas está formada por m neuronas, dispuestas en paralelo, esto es no hay comunicación entre ellas, operan independientemente.

A cada neurona j , de la capa le está asociada una matriz fila $W_j = [w_{j,1} + w_{j,2} + \dots + w_{j,n}]$ que es la matriz de pesos de dicha neurona. El proceso, dentro de la neurona j , se representa mediante la suma de los productos de cada señal por su peso, más un sesgo b_j , en otras palabras por: $W_j p + b_j = [w_{j,1} p_1 + w_{j,2} p_2 + \dots + w_{j,n} p_n + b_j]$. Este valor es transformado por la función de transferencia $f_j(s)$ de la misma neurona j , dando lugar a la respuesta final: $q_j = f_j(W_j p + b_j)$. Este proceso se da en cada neurona en forma independiente. Entonces la matriz de pesos de una capa de m neuronas será una matriz de orden $m \times n$:

$$W = [w_{j,k}] = \begin{bmatrix} w_{1,1}, w_{1,2}, \dots, w_{1,n} \\ w_{2,1}, w_{2,2}, \dots, w_{2,n} \\ \dots \\ w_{m,1}, w_{m,2}, \dots, w_{m,n} \end{bmatrix} \text{ y la ganancia o sesgo de la capa será la matriz}$$

$$\text{Columna } b = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{bmatrix} \text{ de los sesgos de cada neurona de la capa.}$$

Donde la primera fila de las dos matrices corresponden a la primera neurona, la segunda a la segunda neurona de la capa y así sucesivamente.

Para un vector $p = [p_1, p_2, \dots, p_n]$ que llega a la capa de neuronas, el proceso se representa mediante el producto de la matriz de pesos W por el vector p , más el vector b de ganancias de la capa de neuronas. Luego cada componente de este vector es transformado por la función de transferencia de cada neurona. Las funciones de transferencia en cada capa se representan mediante una función vectorial con valores vectoriales: $F: R^m \rightarrow R^m$, donde para cada vector $s = [s_1, s_2, \dots, s_m] \in R^m$, la función F está definida como: $F(s) = [f_1(s_1), f_2(s_2), \dots, f_m(s_m)]$.

En consecuencia a la respuesta de la capa será el vector $q = f(W_p + b)$, donde $F(W_p + b) = [f_1(w_1 p + b_1), f_2(w_2 p + b_2), \dots, f_m(w_m p + b_m)]$, que es la señal procesada y transformada por la capa de neuronas. Entonces es el vector de salida de la capa de neuronas es $q = T(p) = F(W_p + b)$

3.3.1.3. Funciones de transferencias

Existen muchas funciones de transferencia empleadas en el diseño de redes neuronales, las más comunes son:

a) Limitador fuerte (Hardlim):

$$f(s) = \begin{cases} 0 & \text{si } s < 0 \\ 1 & \text{si } s \geq 0 \end{cases}$$

b) Limitador fuerte simétrico (Hardlims):

$$f(s) = \begin{cases} -1 & \text{si } s < 0 \\ 1 & \text{si } s \geq 0 \end{cases}$$

c) Función de transferencia lineal positiva (poslin)

$$f(s) = \begin{cases} 0 & \text{si } s < 0 \\ s & \text{si } s \geq 0 \end{cases}$$

d) Función de transferencia lineal (purelin):

$$f(s) = s$$

e) Función de transferencia lineal saturada (satlin)

$$f(s) = \begin{cases} 0 & \text{si } s < 0 \\ s & \text{si } 0 \leq s \leq 1 \\ 1 & \text{si } 1 < s \end{cases}$$

f) Función de transferencia lineal saturada Simétrico (satlins)

$$f(s) = \begin{cases} -1 & \text{si } s < -1 \\ s & \text{si } -1 \leq s \leq 1 \\ 1 & \text{si } 1 < s \end{cases}$$

g) Función de transferencia sigmoideal (logsig):

$$f(s) = \frac{1}{1 + e^{-s}}$$

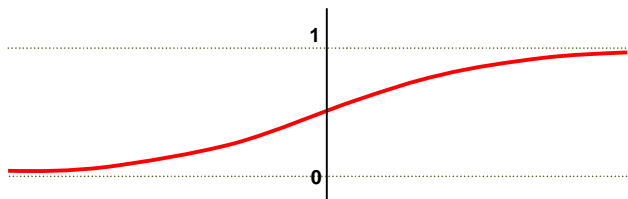


Figura N° 04. Función de transferencia sigmoideal

h) Función de transferencia tangente sigmoideal hiperbólica (tansig):

$$f(s) = \tanh(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}}$$

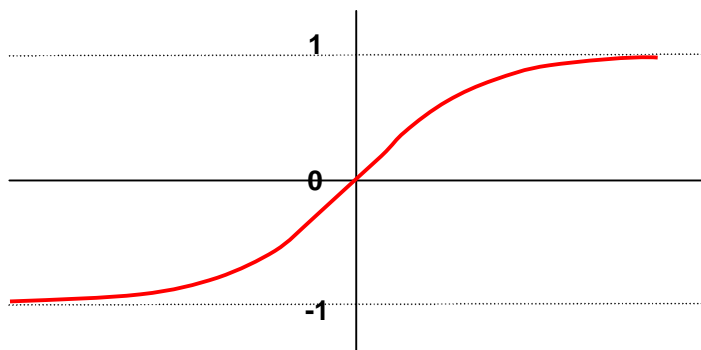


Figura N° 05.- Función de transferencia tangente sigmoideal hiperbólica

i) Función de transferencia competitiva (compet)

$$f(s) = \begin{cases} 1 & \text{neurona con } s \text{ máximo} \\ 0 & \text{en los demas casos} \end{cases}$$

3.3.1.4. Redes neuronales de varias capas o redes multicapa

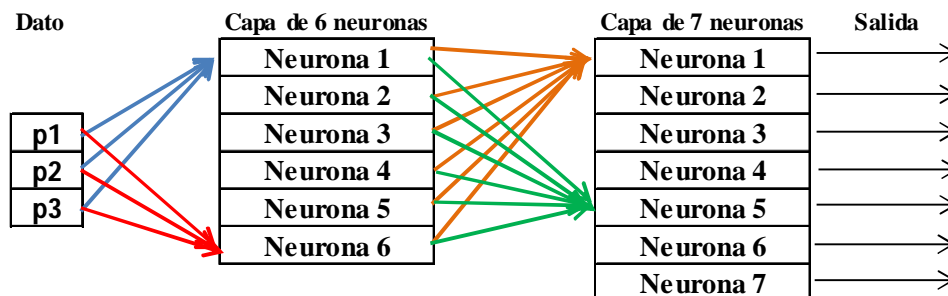


Figura N° 06.- Esquema de redes neuronales de varias capas o redes multicapas

Dentro de una red neuronal, los elementos de procesamiento se encuentran agrupados por capas, una capa es una colección de neuronas; de acuerdo a la ubicación de la capa en la RNA, estas reciben diferentes nombres.

Capa de entrada: Es la primera capa de neuronas. Es la que recibe los datos de entrada a la red.

Capas ocultas: Estas son las capas que siguen a la capa de entrada. Tampoco emiten las señales finales.

Capa de salida: Es la última capa que sigue a las capas anteriores y es la que envía la respuesta final de la red.

Formulación matemática de una red neuronal multicapa

Supongamos que se tiene una RNA de N capas. Entonces:

- a) Para la primera capa se tienen una matriz de pesos $W^1 = [w_{j,k}^1]$ de $m1 \times n$, un vector de ganancias $b^1 = [b_k^1]$ y una función vectorial de funciones de transferencia $F^1(S) = [f_1^1(s_1), f_2^1(s_2), \dots, f_{m1}^1(s_{m1})]$. La Salida es $q^1 = F^1(W^1 p + b^1)$.
- b) Para la segunda capa se tendrá una matriz de pesos $W^2 = [w_{j,k}^2]$ de $m2 \times m1$, un vector de ganancias $b^2 = [b_k^2]$ y una función vectorial de funciones de transferencia $F^2(S) = [f_1^2(s_1), f_2^2(s_2), \dots, f_{m2}^2(s_{m2})]$. La señal emitida será $q^2 = F^2(W^2 Q^1 + b^2)$.
- c) Para la N -ésima capa se tendrá una matriz de pesos $W^N = [w_{j,k}^N]$ de $mN \times m(N-1)$, un vector de ganancias $b^N = [b_k^N]$ y una función vectorial de funciones de transferencia $F^N(S) = [f_1^N(s_1), f_2^N(s_2), \dots, f_{mN}^N(s_{mN})]$. La señal emitida será $Q^N = F^N(W^N Q^{m(N-1)} + b^N)$. De esta manera se observa que la salida final de la red neuronal de N capas será:

$$p \rightarrow q^1 = F^1(W^1 p + b^1) \rightarrow q^2 = F^2(W^2 p + b^2) \dots \rightarrow q^N = F^N(W^N q^{m(N-1)} + b^N)$$

Que es la composición de N transformaciones o N funciones vectoriales no lineales a valores vectoriales. En síntesis es la composición de N transformaciones, que se concatenan o componen:

$$R^n \xrightarrow{T_1} R^{m1} \xrightarrow{T_2} R^{m2} \xrightarrow{T_3} \dots \xrightarrow{T_N} R^{mN}$$

3.3.2. Clasificación de las Redes Neuronales Artificiales (RNA) en supervisadas y no supervisadas

Existen otras clasificaciones de las redes neuronales. Una por ejemplo, es por el tipo de variables que se emplean, denominándose RNA de variables binarias o variables continuas.

Otra clasificación es por el tipo de aprendizaje. En este aspecto, existen dos tipos de redes, unas que tienen un aprendizaje supervisado y otras no.

3.3.2.1. Las Redes Neuronales Artificiales de aprendizaje supervisado

Se caracteriza por que está acompañada vectores $t = [t_1, t_2, \dots, t_M]$, $t_k \in R^m$ (o sea t es una matriz de m filas por M columnas) denominada: Valor esperado de la red.

Dada una matriz de vectores $p = [p_1, p_2, \dots, p_N]$, $p_k \in R^n$ (de n filas por N columnas), que son los datos de entrada a la red, se tiene una respuesta de la red: $q = [q_1, q_2, \dots, q_M]$, $q_k \in R^m$, que es una matriz del mismo orden t. Como es sabido, la respuesta de la RNA depende de los pesos y sesgos de todas las neuronas involucradas en ellas. El conjunto de estos pesos y sesgos se pueden representar con una variable vectorial: x.

Entonces la respuesta $q = q(x)$ depende de esta variable.

El aprendizaje de la red consiste en minimizar el error de la media cuadrática entre los vectores t y la respuesta q de la RNA a la que ha ingresado una secuencia de datos

$$p = [p_1, p_2, \dots, p_N], p_k \in R^n.$$

Entre otros términos el aprendizaje de la red consiste en minimizar la función de error

$$\text{medio cuadrático: } E(x) = \frac{1}{2} \sum_{k=1}^M \|q_k(x) - t_k\|^2, \text{ donde como se dijo antes, x es un vector}$$

que representa los pesos y sesgos de todas las neuronas que conforman la red. Lo que se espera es que la respuesta de la red este lo más cerca posible la valor esperado t, es decir que E(x) encuentre un mínimo local. Cuando se ha conseguido esto se dice que la RNA ha sido entrenada.

3.3.2.2. Las Redes Neuronales Artificiales de aprendizaje no supervisado

Son las que no requieren del valor esperado t, solo de una matriz de vectores $p = [p_1, p_2, \dots, p_N]$, $p_k \in R^n$. Estas redes tienen una sola capa, cuyo número de neuronas lo elige el usuario. Su función es agrupar los vectores en grupos o clusters, empleando parte de las normas o métricas (Minkowski). Además MATLAB ofrece visualizaciones gráficas de los clusters en dos dimensiones, bajo ciertas formas o estructuras geométricas, denominadas Topologías. Existen muchas Topologías que elige el usuario.

3.3.3. Redes Neuronales Backpropagation

3.3.3.1.Introducción.

Durante muchos años no se obtuvo ningún tipo de éxito en el diseño de algoritmos de entrenamiento de redes multicapa. A partir de la comprobación de la severa limitación de los sistemas de una capa, el mundo de la computación neuronal entró en un obscurecimiento y abandono casi general durante dos décadas.

La invención del algoritmo Backpropagation ha desempeñado un papel vital en el resurgimiento del interés de las redes neuronales artificiales. Backpropagation es un método de entrenamiento de redes multicapa. Su potencia reside en su capacidad de entrenar capas ocultas y de este modo supera las posibilidades restringidas de las redes de una única capa.

El concepto básico de Backpropagation fue presentado en 1974 por Paul Werbos e independientemente reinventado por David Parker en 1982, y también presentado en 1986 por Rumelhart, Hinton y Willians. La duplicidad de esfuerzos y trabajos es frecuente en cualquier disciplina, y más en el mundo de las Redes Neuronales Artificiales debido a su naturaleza interdisciplinaria.

El método backpropagation (propagación del error hacia atrás), basado en la generalización de la regla delta, a pesar de sus limitaciones, ha ampliado de forma considerable el rango de aplicaciones de las redes neuronales. El funcionamiento de la red backpropagation (BPN) consiste en el aprendizaje de un conjunto predefinido de pares de entradas-salidas dados como ejemplo: primero se aplica un patrón de entrada como estímulo para la primera capa de las neuronas de la red, se va propagando a través de todas las capas superiores hasta generar una salida, se compara el resultado en las neuronas de salida con la salida que se desea obtener y se calcula un valor de error para cada neurona de salida. A continuación, estos errores se transmiten hacia atrás, partiendo de la capa de salida hacia todas las neuronas de la capa intermedia que contribuyan directamente a la salida. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido un error que describa su aportación relativa al error total. Basándose en el valor del error recibido, se reajustan los pesos de conexión de cada neurona, de manera que en la siguiente vez que se presente el mismo patrón, la salida esté más cercana a la deseada.

La importancia de la red backpropagation consiste en su capacidad de autoadaptar los pesos de las neuronas de las capas intermedias para aprender la relación que existe entre un conjunto de patrones de entrada y sus salidas correspondientes. Es importante la capacidad de generalización, facilidad de dar salidas satisfactorias a entradas que el sistema no ha visto nunca en su fase de entrenamiento. La red debe encontrar una representación interna que le permita generar las salidas deseadas cuando se le dan entradas de entrenamiento, y que pueda aplicar, además, a entradas no presentadas durante la etapa de aprendizaje para clasificarlas

3.3.3.2. Estructura y Aprendizaje de la Red Backpropagation

En una red Backpropagation existe una capa de entrada con n neuronas y una capa de salida con m neuronas y al menos una capa oculta de neuronas internas. Cada neurona de una capa (excepto las de entrada) recibe entradas de todas las neuronas de la capa anterior y envía su salida a todas las neuronas de la capa posterior (excepto las de salida). No hay conexiones hacia atrás feedback ni laterales entre las neuronas de la misma capa.

La aplicación del algoritmo tiene dos fases, una hacia delante y otra hacia atrás. Durante la primera fase el patrón de entrada es presentado a la red y propagado a través de las capas hasta llegar a la capa de salida. Obtenidos los valores de salida de la red, se inicia la segunda fase, comparándose éstos valores con la salida esperada para así obtener el error. Se ajustan los pesos de la última capa proporcionalmente al error. Se pasa a la capa anterior con una retropropagación del error, ajustando los pesos y continuando con este proceso hasta llegar a la primera capa. De esta manera se han modificado los pesos de las conexiones de la red para cada patrón de aprendizaje del problema, del que conocíamos su valor de entrada y la salida deseada que debería generar la red ante dicho patrón.

La técnica Backpropagation requiere el uso de neuronas cuya función de activación sea continua, y por lo tanto, diferenciable. Generalmente, la función utilizada será del tipo tangente sigmooidal Hiperbólica (tansig).

3.3.3.3. Conjuntos de Entrenamiento y Testeo

El valor de una red Backpropagation entrenada se mide por su capacidad de generalizar. Esto es, de dar buenas predicciones para datos no utilizados en el entrenamiento. Esto implica el uso de por lo menos tres conjuntos de datos, uno para entrenar, otro para validar y otro para testear la red entrenada. Usualmente esta división de los datos originales que posee el modelador se hace utilizando al azar y normalmente tal división se efectúa de manera tal que el 25 % de los datos se deja para testear, el 25% de los datos se deja para validar y el resto para entrenar.

3.3.3.4. Funciones de Error

La función de error medio cuadrático: $E(x) = \frac{1}{2} \sum_{k=1}^M \|q_k(x) - t_k\|^2$, donde como se dijo antes, x es un vector que representa los pesos y sesgos de todas las neuronas que conforman la red. Lo que se espera es que la respuesta de la red este lo más cerca posible la valor esperado t, es decir que E(x) encuentre un mínimo local. Cuando se ha conseguido esto se dice que la RNA ha sido entrenada.

3.3.3.5.Épocas

No necesariamente los pesos deben ser cambiados cada vez que se aplica un par de entrenamiento. Podría ser deseable aplicar varios pares de entrenamiento registrando las variaciones de pesos que implica cada uno (guardándolas en memoria) y luego cambiar los pesos con la modificación neta que representan las modificaciones calculadas para cada par. Quedan definidos entonces grupos de pares de entrenamiento, cada uno de los cuales responsable de una modificación a los pesos. Estos grupos son llamados épocas¹⁰. La motivación para usar una metodología como esta es que si se modifican los pesos en cada par de entrenamiento se está buscando una mejora que es local a ese par, pero alterar los pesos en función de un neto de un grupo o el total de los pares minimiza el error del conjunto, buscando un conjunto de pesos que es el óptimo (o más cercano al óptimo) para el conjunto de datos presentados a la red en vez de a una observación en particular. Este tipo de entrenamiento es llamado Actualización Acumulativa o Retropropagación Acumulativa.

Cuando el valor de la época coincide con el número total de pares de entrenamiento (máximo valor que la época puede tomar), el algoritmo obtiene la mejor estimación posible del gradiente (por lo que en ese caso se habla de obtener el "gradiente verdadero"), mientras que actualizando los pesos con cada observación la estimación del gradiente contiene más ruido (razón por la que a veces es llamado "aprendizaje estocástico"). Aunque el uso de épocas mayores a 1 suena razonable, la experiencia muestra que una retropropagación acumulativa no necesariamente lleva a mejores resultados. Se ha encontrado que el aprendizaje con época 1 es más rápido y alcanza mejores soluciones, en parte porque el ruido asociado permite al algoritmo escapar de mínimos locales (LeCun et al., 1998).

3.3.3.6.Saturación

Cuando se usan funciones de activación tipo sigmoide, un problema que suele suceder es la saturación de las neuronas. Si la suma ponderada que una neurona recibe como entrada es muy grande o muy pequeña, la derivada de f se hace muy cercana a cero. Como el error calculado en una neurona es proporcional a esta derivada, resulta finalmente que el error local en dicha neurona es cero y por lo tanto la neurona deja de aprender (los pesos asociados a ella dejan de cambiar). Este fenómeno es conocido como "parálisis". No hay acuerdo en cómo debe ser resuelto el problema y varias formas de solucionarlo se han propuesto. Una de ellas, por ejemplo, recomienda cambiar los pesos de las conexiones que llegan a neuronas paralizadas a valores al azar, resolviendo el problema de la parálisis, pero dañando considerablemente el aprendizaje de dichos pesos hasta ese momento. Otra opción menos destructiva es aplicar un desfase (del orden de 0.1) a la derivada de f antes de que ésta escale el error, evitando así errores nulos.

¹⁰ A veces la palabra "época" es utilizada para indicar que se ha realizado una "pasada" al conjunto de entrenamiento (todos los pares del conjunto de entrenamiento fueron presentados a la red). Aquí se la utilizará en el sentido descrito en el cuerpo del texto.

3.3.3.7. Sobre-ajuste

En algunas aplicaciones se ha encontrado que la red ajusta muy bien los datos de entrenamiento, pero notoriamente peor los de testeo. Cuando ello ocurre, se dice que la red está "sobre-ajustada". El problema provendría de una alta capacidad de ajuste de la red en comparación con la cantidad y variabilidad de los datos, por lo que se admite que el sobre-ajuste puede resolverse mediante la disminución de las conexiones en la red y/o el aumento de datos en el conjunto de entrenamiento. Una técnica más elaborada para evitarlo es la de la "validación cruzada". La estrategia implica particionar el conjunto de datos en tres. Un conjunto es para entrenar, otro para testear y el tercero es el llamado "conjunto de validación". El aprendizaje se realiza con el conjunto de entrenamiento, pero se detiene cada ciertas iteraciones para aplicar la red sobre el conjunto de validación. La idea es detectar cuándo la red comienza a tener un ajuste muy alto con los datos de entrenamiento, pero en desmedro de datos fuera de dicho conjunto, comienza a perder la capacidad de generalizar. Detectado ese punto, la mejor configuración de la red corresponde justamente a la que posee la red en ese instante (o una iteración atrás, si se quiere). Finalmente se corre el verdadero test. La estrategia es costosa en tiempo y computación necesaria, pero puede ser una opción necesaria cuando existe el problema del sobre-ajuste. En la presente tesis se está utilizando la técnica de validación cruzada para evitar el Sobre ajuste.

3.3.3.8. Aplicaciones de la Red Backpropagation

Las redes Backpropagation han demostrado su capacidad de trabajar con éxito en un amplio rango de aplicaciones incluyendo clasificación de imágenes, síntesis de voz, clasificación de ecos de sonar, sistemas de base de conocimiento, codificación de información y muchos otros problemas de clasificación y problemas de percepción.

Algunos ejemplos y estudios de aplicaciones de la Backpropagation son los siguientes:

Sejnowski y Rosenberg (1987) lograron un gran éxito con el sistema llamado NetTalk, un sistema que convierte texto escrito en inglés a voz de alta inteligibilidad. La voz obtenida en la sesión de entrenamiento recuerda los sonidos de un niño en sus diferentes estados del aprendizaje del hablar.

En Japón NEC ha anunciado la utilización de una red backpropagation en un sistema de reconocimiento óptico de caracteres, obteniendo una exactitud superior al 99%. Esta mejora ha sido conseguida mediante la combinación de algoritmos convencionales y una backpropagation que provee una verificación adicional.

Otra aplicación de la red backpropagation es el reconocimiento de formas de dos dimensiones. Este tipo de sistemas es muy útil en aplicaciones de identificación de números escritos a mano, lectura de caracteres escritos a mano, ordenamiento de partes en una producción industrial, inspección automática de defectos y procesado de imágenes médicas (Dayhoff 1988).

Cotrell, Munro y Zipper (1987) han realizado una aplicación de compresión de imagen en la que las imágenes se representan con un bit por pixel, obteniendo una reducción de sobre los datos de entrada.

La aplicación de Waibel (1988) consiste en un estudio de clasificación de patrones que son presentados fuera de un período de tiempo. La red de Waibel es una red neuronal constituida con elementos de retardo en el tiempo que ha sido entrenada para reconocer sílabas habladas.

Un ejemplo clásico de la utilización de red Backpropagation es la función OR exclusivo. La red Perceptron no es capaz de resolver este problema porque el problema no es linealmente separable y su solución requiere dos capas de pesos ajustables. Sin embargo, la red Backpropagation dispone de un método de entrenamiento que ajusta los pesos de todas las capas y resuelve este problema linealmente no separable. No obstante uno de los problemas comunes en resolver la función X-or con la red Backpropagation es la presencia de mínimos locales y por consiguiente la falta de convergencia de las respuestas correctas para todos los patrones del conjunto de entrenamiento.

3.3.4. Redes Neuronales Self Organizing Map (SOM).

3.3.4.1. Introducción.

En 1982 T. Kohonen presentó un modelo de red denominado mapas auto-organizados o SOM (Self-Organizing Maps), basado en ciertas evidencias descubiertas a nivel cerebral. Este tipo de red posee un aprendizaje no supervisado competitivo.

Un mapa auto-organizado (SOM por sus siglas en inglés) es un tipo de red neuronal artificial que es entrenada usando aprendizaje no supervisado para producir una representación discreta del espacio de las muestras de entrada, llamado mapa. Los mapas auto-organizados son diferentes de otras redes neurales artificiales, en el sentido que estos usan una función de vecindad para preservar las propiedades topológicas del espacio de entrada.

SOM son útiles para visualizar vistas de baja dimensión de datos de alta dimensión, semejante a un escalado multidimensional. El modelo fue descrito por primera vez como una red neural artificial por el profesor finlandés Teuvo Kohonen, debido a lo cual en ocasiones son llamadas redes o mapas de Kohonen¹¹.

No existe ningún maestro externo que indique si la red neuronal está operando correcta o incorrectamente porque no se dispone de ninguna salida objetivo hacia la cual la red neuronal deba tender.

La red auto-organizada debe descubrir rasgos comunes, regularidades, correlaciones o categorías en los datos de entrada, e incorporarlos a su estructura interna de conexiones. Se dice, por tanto, que las neuronas deben auto-organizarse en función de los estímulos (datos) procedentes del exterior.

¹¹ TASOM: A New Time Adaptive Self-Organizing Map». IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics 33 (2): pp. 271–282. April 2003.

En el aprendizaje competitivo las neuronas compiten unas con otras con el fin de llevar a cabo una tarea dada. Se pretende que cuando se presente a la red un patrón de entrada, sólo una de las neuronas de salida (o un grupo de vecinas) se active.

Por tanto, las neuronas compiten por activarse, quedando finalmente una como neurona vencedora y anulada el resto, que son forzadas a sus valores de respuesta mínimos.

El objetivo de este aprendizaje es categorizar los datos que se introducen en la red. Se clasifican valores similares en la misma categoría y, por tanto, deben activar la misma neurona de salida

Las clases o categorías deben ser creadas por la propia red, puesto que se trata de un aprendizaje no supervisado, a través de las correlaciones entre los datos de entrada.

3.3.4.2. Ideas intuitivas sobre el algoritmo del SOM

- ✓ El SOM es, en realidad, un tipo de algoritmo para clasificar observaciones.
- ✓ Se elige un gran número de clusters y se colocan en forma de una red bidimensional. La idea es que los representantes de cada grupo (o pesos, según la notación de Kohonen) estén correlacionados espacialmente, de modo que los puntos más próximos en la rejilla sean más parecidos entre sí que los que estén muy separados.
- ✓ Este proceso es conceptualmente similar al MDS que transforma observaciones similares en puntos cercanos del espacio bidimensional.
- ✓ Si se discretiza el espacio bidimensional dividiéndolo, por ejemplo, en una rejilla de componentes rectangulares se puede definir una aplicación desde el espacio de alta dimensiones original sobre dicho espacio bidimensional.
- ✓ Además, se puede tomar la media de los elementos que se encuentran en cada elemento de la rejilla para definir representantes de las clases de la rejilla. Los representantes que están en clases próximas se parecen entre sí. La idea básica del SOM es, así, proporcionar una versión discreta del MDS.

3.3.4.3. Fundamentos biológicos

Se ha observado que en el córtex de los animales superiores aparecen zonas donde las neuronas detectoras de rasgos se encuentran topológicamente ordenadas; de forma que las informaciones captadas del entorno a través de los órganos sensoriales, se representan internamente en forma de mapas bidimensionales.

Aunque en gran medida esta organización neuronal está predeterminada genéticamente, es probable que parte de ella se origine mediante el aprendizaje. Esto sugiere, por tanto, que el cerebro podría poseer la capacidad inherente de formar mapas topológicos a partir de las informaciones recibidas del exterior.

También se ha observado que la influencia que una neurona ejerce sobre las demás es función de la distancia entre ellas, siendo muy pequeña cuando están muy alejadas.

El modelo de red auto-organizado presentado por Kohonen pretende mimetizar de forma simplificada la capacidad del cerebro de formar mapas topológicos a partir de las señales recibidas del exterior.

3.3.4.4. Topologías de Capas de Neuronas

Las neuronas en la capa de un SOM pueden ser representadas por puntos o posiciones de un plano bidimensional o plano coordenado XY, formando los vértices de un grafo, donde los vértices son las posiciones de las neuronas y las aristas o arcos son los enlaces entre estos. Los grafos adoptan diversas formas o geometrías, denominadas Topologías. En MATLAB existen funciones que ayudan a crear estas Topologías: gridtop, hextop o randtop, estas tres funciones presentan a las neuronas en forma de una malla rectangular, hexagonal, o en forma de un grafo aleatoria, respectivamente¹².

a) La topología gridtop

Crea una sucesión de puntos del plano XY, que son las posiciones de las neuronas y que son los vértices de una malla o rejilla rectangular.

b) La topología hextop

Con esta función se crean las posiciones de las neuronas, que son los vértices de un grafo, pero que tienen un patrón hexagonal.

c) La topología randtop

Crea grafos para una capa de neuronas con un patrón aleatorio, es decir los vértices del grafo ocupan posiciones aleatorias del plano XY, las aristas en forma aleatoria. El código siguiente genera un patrón aleatorio de estas neuronas.

3.3.4.5. Las métricas o nociones de distancia entre neuronas

La noción de distancia más común entre dos puntos de un espacio es la denominada distancia “euclidiana” que es la longitud del segmento de recta que une estos puntos. La distancia de la Plaza de Armas de Lima a la Iglesia de la Nazarenas se mide en cuadras, no tendría sentido hacerlo con la noción de distancia euclidiana. Desde el punto de vista matemático, existen infinitas nociones de distancia, que se definen a continuación. Pero estas nociones de distancia se basan en la noción de longitud de un vector, pues dos puntos definen un vector¹³.

¹² Universidad Nacional de Ingeniería. FIIS-FIC. Curso de Capacitación: “Redes Neuronales y Simulación”. Proyecto FINCYT: Redes Neuronales y Simulación de Monte Carlo para el estudio del concreto de alta resistencia. Código PIAP-1-P-676-13. Pedro C. Espinoza H.

¹³ Universidad Nacional de Ingeniería. FIIS-FIC. Curso de Capacitación: “Redes Neuronales y Simulación”. Proyecto FINCYT: Redes Neuronales y Simulación de Monte Carlo para el estudio del concreto de alta resistencia. Cap. 1 Sección 2.5. Código PIAP-1-P-676-13. Pedro C. Espinoza H.

Hay cuatro funciones de distancia que emplea MATLAB: dist, boxdist, linkdist y mandist, que son las distancias: euclideana, del máximo, del enlace y de Mahattan respectivamente.

Como hay tres topologías: gridtop, hextop o randtop podemos hacer combinaciones de estas con las cuatro distancias

3.3.4.6. La noción de vecindad

Una topología junto con una distancia, forman lo que se llama en el campo de la matemática, un espacio métrico.

Noción de una vecindad o esfera $B(q, r)$ en R^n , de centro q y radio $r > 0$ y que se definió como el conjunto de puntos p de R^n cuya distancia (euclideana) al centro q es menor que r . La noción de vecindad en los espacios métricos que se ha estudiado en la sección anterior, es similar, con la diferencia de que en estos casos se cuenta con otras nociones de distancia diferente de la euclideana. Los puntos son de R^2 o del plano XY que representan las posiciones de un conjunto de neuronas. Las esferas o vecindades son cerradas y tendrán diferentes formas dependiendo de la métrica.

a) Definición de Vecindad de una neurona

Dada una topología T y una distancia d , se tiene un espacio métrico. Entonces la vecindad o esfera de radio r , de una neurona i^* , se denota $N(i^*, r)$ y se define como el conjunto de todas la neuronas i que están en T a una distancia menor o igual que r , brevemente: $N(i^*, r) = \{i \text{ en } T, \text{ tal que } d(i, i^*) \leq r\}$.

b) Neurona ganadora y el algoritmo de Kohonen

En la caja de herramientas de MATLAB, hay tres maneras de crear topologías y cuatro formas de calcular distancias. Con ellas se definen diversos espacios métricos y en cada uno de estos se define lo que es una vecindad de neuronas.

En un mapa de auto-organización, lo que se hace es identificar una neurona ganadora i^* y actualizar los pesos (no tienen ganancias o sesgos) de todas las neuronas dentro de una determinada vecindad $N(i^*, r)$ de la neurona ganadora i^* .

Si $w_i(k) = [w_{i,1}(k), w_{i,2}(k), \dots, w_{i,N}(k)]$ es la matriz de pesos de una de las neuronas "i" en la iteración k . Entonces la actualización de estos valores para cada neurona de la vecindad $N(i^*, r) = \{i \in T : d(i, i^*) \leq r\}$ se realiza mediante el siguiente algoritmo:

$$w_{i,i}(k+1) = w_{i,j}(k) + \alpha(p_j - w_{i,j}(k))$$

$$w_{i,i}(k+1) = (1 - \alpha)w_{i,j}(k) + \alpha p_j$$

Aquí la vecindad $N(i^*, r)$ contiene todas las neuronas que se encuentran a una distancia menor o igual que r , de la neurona ganadora i^* .

Así, cuando un vector p ingresa a la red, los pesos de la neurona ganadora y de sus vecinos más próximos avanzan hacia p . En consecuencia, después de muchas iteraciones, las neuronas de una vecindad $N(i^*, r)$ tendrán pesos similares entre sí.

3.3.4.7. Arquitectura de las Redes Neuronales SOM.

Esta arquitectura no emplea sesgos, lo demás es como la de una red cualquiera, con la salvedad de que sus funciones de transferencia son como las de las RNA competitivas.

Se puede elegir entre varias topologías para desarrollar los gráficos de las capas de neuronas y también las diversas distancias para calcular las neuronas que están cerca de la neurona ganadora.

Un modelo SOM está compuesto por dos capas de neuronas. La capa de entrada (formada por N neuronas, una por cada variable de entrada) se encarga de recibir y transmitir a la capa de salida la información procedente del exterior. La capa de salida (formada por M neuronas) es la encargada de procesar la información y formar el mapa de rasgos.

Normalmente, las neuronas de la capa de salida se organizan en forma de mapa bidimensional como se muestra en la figura:

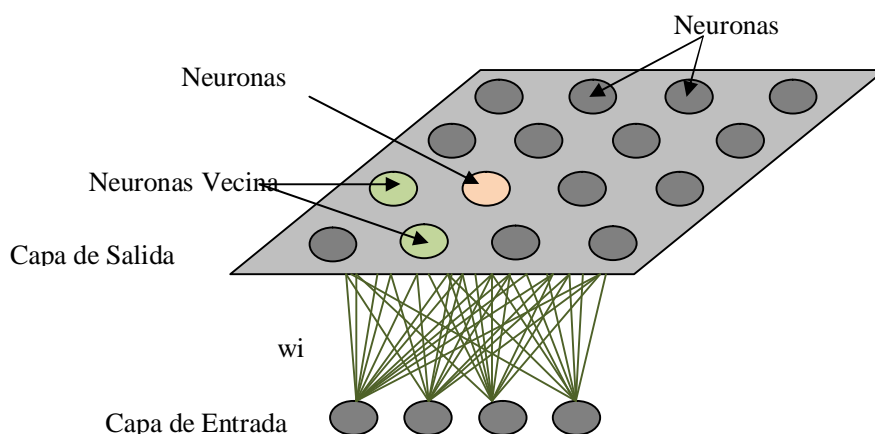


Figura N° 07. Estructura de un mapa autoorganizado (SOM)

Las conexiones entre las dos capas que forman la red son siempre hacia delante, es decir, la información se propaga desde la capa de entrada hacia la capa de salida. Cada neurona de entrada i está conectada con cada una de las neuronas de salida j mediante un peso w_{ji} . De esta forma, las neuronas de salida tienen asociado un vector de pesos W_j llamado vector de referencia (o codebook), debido a que constituye el vector prototipo (o promedio) de la categoría representada por la neurona de salida j . Así, el SOM define

una proyección desde un espacio de datos en alta dimensión a un mapa bidimensional de neuronas.

Entre las neuronas de la capa de salida, puede decirse que existen conexiones laterales de excitación e inhibición implícitas, pues aunque no estén conectadas, cada una de estas neuronas va a tener cierta influencia sobre sus vecinas. Esto se consigue a través de un proceso de competición entre las neuronas y de la aplicación de una función denominada de vecindad, que produce la topología o estructura del mapa. Las topologías más frecuentes son la rectangular y la hexagonal.

Las neuronas adyacentes pertenecen a una vecindad N_j de la neurona j . La topología y el número de neuronas permanecen fijos desde el principio. El número de neuronas determina la suavidad de la proyección, lo cual influye en el ajuste y capacidad de generalización del SOM.

Durante la fase de entrenamiento, el SOM forma una red elástica que se pliega dentro de la nube de datos originales. El algoritmo controla la red de modo que tiende a aproximar la densidad de los datos.

3.4. Modelación de la Demanda de Transporte

3.4.1. Introducción

Lo que se desea determinar en el modelamiento de la demanda de Transporte es la magnitud y estructura de los viajes en una determinada área de estudio en base a ciertas variables de ésta. Domencich y McFadden (1975) describen en forma clara y desde una perspectiva amplia el ámbito de la modelación de la demanda de transporte. Explican cómo ésta tiene componentes y características que la hacen muy compleja e interesante a la vez, como es la componente espacial, la componente temporal, la incidencia de las costumbres, actitudes y preferencias de los usuarios, así como las percepciones que éstos tienen de su entorno. Discuten también que el comportamiento de viajes que se desea modelar es el cúmulo de un proceso decisional de los individuos o incluso de grupos de individuos (como en el caso familiar) que incluyen desde la decisión básica de viajar o no, hasta las elecciones de destino, de modo (simple o mixto), de rutas, líneas de transporte público, hora de realización del viaje, etc.

Respecto a la parte técnica que tiene que ver con la formulación de modelos, estos autores indican que dichas formulaciones deben tener ciertas características básicas para que puedan ser realmente aprovechados. Entre éstas destacan la incorporación de variables de política y la sensibilidad de los modelos respecto de las variables del sistema de transporte tales como tiempos de viaje y tarifas de transporte público. La capacidad para incorporar estos elementos permitirá a los modelos ser reales herramientas de toma de decisión, puesto que en efecto podrán mostrar al modelador las consecuencias de los diferentes cambios que pueden preverse en el área de estudio.

Cabe mencionar que un modelo de demanda no siempre se construye como vía para obtener predicciones de situaciones futuras, sino también como forma de entender

mejor el fenómeno de la realización de viajes u obtener medidas colaterales relevantes como la valoración del tiempo de los individuos (Ortúzar y Willumsen, 1990).

Cabe indicar que posteriormente se presentaron varios estilos y métodos de modelación, lo cuales solo cubren una pequeña fracción del problema. Tal motivo hace presentar una breve discusión respecto del "modelo clásico del transporte", en donde en teoría se intenta recoger el fenómeno en conjunto.

3.4.1.1.El Modelo de las 4 Etapas

En el mercado del transporte el equilibrio general se puede resolver bajo un enfoque conocido como modelo de las cuatro etapas o modelo clásico del transporte (Ortúzar y Willumsen, 1990), en el cual el problema general ha sido dividido en 4 problemas de menor envergadura. Las cuatro etapas son las siguientes:

1. Generación/Atracción (decisión de viajar o no).
2. Distribución (dónde viajar).
3. Partición Modal (qué modo de transporte utilizar).
4. Asignación (qué ruta utilizar).

Estas etapas se usan en relación a la palabra demanda, pero no se precisa una definición concreta. Se podría referir a una de estas etapas o la combinación de estas. En esta tesis, distribución, Partición Modal y asignación no será incluida dentro del concepto de demanda.

Cada una de estas etapas utiliza un modelo matemático y la meta es encontrar una solución para cada modelo en que a la vez exista consistencia entre las variables comunes de los modelos, lo que se denomina "consistencia interna de los modelos" (Safwat y Magnanti, 1988). La coexistencia de varios modelos bajo una restricción de consistencia es un problema en sí. Como Williams (1977) dice, por un lado está la separabilidad, que dice relación con la factibilidad de separar el proceso decisional de los individuos en varios modelos¹⁴ y por otra parte se tiene la secuencia, que se refiere a la estructura de resolución de las etapas (o si es resuelto en forma simultánea). En el caso de modelos separados, el problema más claro en cuanto a interconectar modelos aparece en la variable costo de transporte, en donde diversas formas de agregación o construcción de costos compuestos han mostrado tener dificultades propias. De hecho, Williams analiza varias formas de composición de costo utilizadas en diversos trabajos y aplicaciones y encuentra que ninguna cumple condiciones básicas.

En cuanto a resolución secuencial o simultánea, matemáticamente lo apropiado es resolver los submodelos simultáneamente, de modo de obtener resultados consistentes entre sí (consistencia interna). Sin embargo, las dificultades matemáticas y computacionales de lograr esto han propiciado el enfoque secuencial. En este enfoque

¹⁴ Factibilidad en el sentido de considerar aceptable el supuesto de que la toma de decisión puede modelarse separadamente sin introducir sesgos.

los resultados de un modelo son entradas para el siguiente (siguiendo el orden de las 4 etapas) hasta obtener costos del submodelo de asignación que vuelven como entradas a la primera¹⁵ etapa, produciendo iteraciones que lleven eventualmente a una convergencia de las variables entre los submodelos (véase figura 8). El primer problema que asoma en este método es el orden de las etapas. Como Williams expone, desde el comienzo hubo dudas sobre la posición de la partición modal respecto de la distribución. En forma general, un orden dado implica supuestos sobre el proceso de decisión de los individuos. Además, Safwat y Magnanti explican que esta metodología de resolución (secuencial) no lleva necesariamente a resultados internamente consistentes, lo que ha generado interés académico en generar formas de resolver el problema simultáneamente. Esta simultaneidad no sólo es robusta matemáticamente, sino que además reconoce la inherente interrelación entre las etapas del modelo, con lo cual el proceso real de decisión de los individuos estaría más fielmente modelado. Tres estrategias conocidas para resolver el problema simultáneo son la Optimización Equivalente, las Desigualdades Variacionales y el Equilibrio Estocástico (Safwat y Magnanti, 1988). Para una descripción de ellas véase respectivamente a Florian (1986), Hammond (1984) y Sheffi y Daganzo (1980).

Es claro que el problema de la interrelación de variables entre modelos así como el problema de la secuencialidad/simultaneidad podría verse notoriamente aliviados en la medida que el número de submodelos disminuye. Esta idea es la que ha incentivado la creación de modelos integrados, generándose modelos que, por ejemplo, resuelven distribución y partición modal conjuntamente. Si aceptamos que separación v/s integración y secuencialidad v/s simultaneidad son estrategias nacidas a partir de la limitada capacidad que tenemos para resolver un problema de gran complejidad como lo es el comportamiento humano en la esfera particular de las decisiones de transporte, entonces un modelo único (mono-etapa) permanecería como el ideal al que aspirar.

¹⁵ Los algoritmos que resuelven el problema de las 4 etapas no recalculan normalmente generación porque por simplicidad se ha preferido modelos de generación y atracción inelásticos respecto de los costos de viaje. Teóricamente, sin embargo, el número de viajes que se realicen en el área de estudio ciertamente dependerán de los costos de viaje del sistema, especialmente para viajes no obligados. En la figura 1 se ha usado una línea discontinua para mostrar la ausencia de una relación en teoría existente.

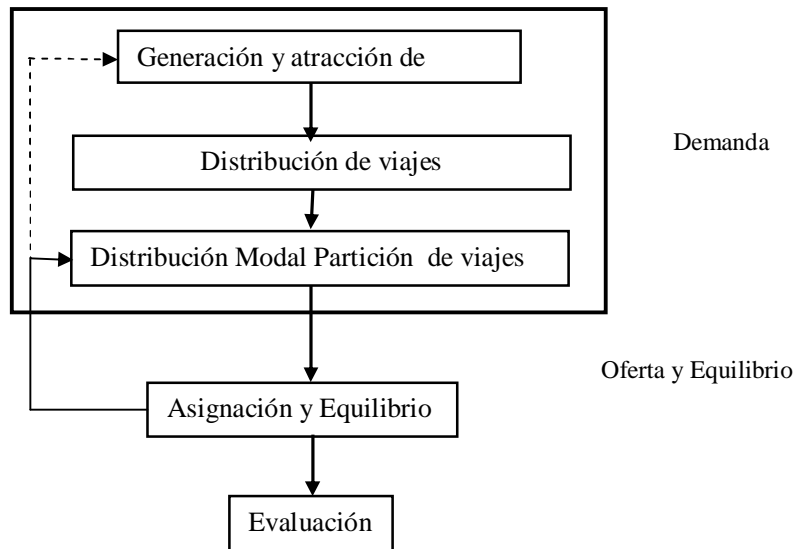


Figura N° 08.- Resolución secuencial del modelo de 4 etapas

De todas formas, ya sea en forma secuencial o simultánea, los submodelos (en distinto grado de integración) aparecen como piezas que pueden describirse y analizarse en forma individual, no sólo porque así ha sido la construcción histórica de ellos, sino porque en la mayoría de las aplicaciones no se desea resolver el problema global del equilibrio del mercado del transporte, sino que sólo interesa estudiar una parte del proceso decisional. Dicha descripción para cada submodelo es la meta de las secciones posteriores, previo a lo cual se comentan otras características generales de la modelación del transporte.

3.4.1.2. Tiempo y Espacio

En la modelación no sólo existe una división del proceso decisional, sino también del tiempo y el espacio. El tiempo se divide en períodos y el espacio en zonas (Ortúzar y Willumsen, 1990). Nuevamente es la necesidad de simplificación del fenómeno la que origina esta división. A lo largo del día, de la semana y del año, el comportamiento de los viajeros cambia y modelar este efecto directamente en el modelo no es fácil. En cuanto al espacio, sería imposible alimentar al modelo con información detallada de lo que existe en cada posición espacial. Esto lleva a la agregación espacial, creando zonas.

Estas simplificaciones adicionales pueden generar sus propios problemas. En el caso espacial, la división, aún bajo criterios razonables, puede limitar la capacidad de los modelos de reproducir el fenómeno en estudio, ya que se están artificialmente creando unidades independientes (zonas) a partir de un elemento intrínsecamente continuo (el suelo). Martínez y Schwarz (1997) estudiaron la autocorrelación espacial de las zonas para MUSSA (Modelo de Uso de Suelo de Santiago), modelo basado en la teoría Bid-Choice (Martínez, 1992). Encontraron que dicha correlación existe. Esta evidencia indicaría que la localización de los agentes en la ciudad no sólo depende de variables de la zona en cuestión que se analiza, sino también de las características de las demás zonas. Para el caso de los viajes podría plantearse como hipótesis, por ejemplo, que el número de viajes atraídos a una zona depende en parte de lo que existe en zonas aledañas.

3.4.1.3. Propósito y Tipología de Usuario

Otra división usual es la de separar la modelación según el motivo o propósito que define el viaje. Es así como se calibran modelos con distintos parámetros para cada uno de ellos, bajo el supuesto de que el comportamiento asociado a un determinado propósito es suficientemente distinto del asociado a otro. La modelación de los viajes realizados "al trabajo" por mucho acapara el interés en relación a otros propósitos, pues representan la mayoría de los viajes realizados en un área urbana.

También basado en diferencias de comportamiento, a veces se prefiere distinguir categorías de usuario en la población, preferentemente en base a ingreso, pero usualmente también utilizando la posesión de automóvil o la ocupación de las personas (Ortúzar y Willumsen, 1990).

3.4.1.4. Visión Agregada v/s Desagregada

Una división importante de los modelos de demanda nace a partir del nivel de información con que se analizan los viajeros. Ciertos modelos trabajan con observaciones individuales de personas tomando decisiones de transporte, mientras otros se centran en información macro, un agregado de las decisiones individuales (Ortúzar y Willumsen, 1990). El enfoque desagregado descansa sobre teorías microeconómicas y psicológicas que explican las decisiones de cada agente, mientras que el agregado opera bajo la noción de flujos de personas que pueden ser explicados por variables macro (zonales) y en donde el modelo no está construido basado en reglas de comportamiento, sino que toma cualquier forma funcional que razonablemente pueda ajustar los datos de trabajo.

Los enfoques microeconómico (desagregado) y estadístico (agregado) han competido entre sí históricamente, dando origen a lo que se pueden considerar escuelas de modelación diferentes. Sin embargo, esta dicotomía no necesariamente es tal, como lo muestra el trabajo de Anas (1983), en donde se demuestra formalmente que los dos modelos clásicos de cada área son equivalentes.

3.4.2. Modelos de Generación y Atracción

Una de las falencias más notorias de los modelos de este tipo es que no se suele incorporar la elasticidad frente a los costos de transporte. Este efecto está muy ligado al concepto de la accesibilidad, una palabra muy usada en la literatura, pero pocas veces especificada matemáticamente.

Que se modelen los viajes atraídos (modelos de atracción) en cada zona no deja de ser extraño. En principio bastaría sólo conocer los viajes generados y luego distribuirlos con un modelo de distribución hacia los destinos disponibles. Además, teniendo un modelo para generación y otro para atracción, difícilmente los totales generados y

atraídos (a nivel de área de estudio) coincidirán¹⁶. Los modelos de atracción nacen en realidad de modelos de distribución que requieren esta información, modelos de distribución que "cuadran" matrices origen-destino en vez de distribuir viajes generados.

Los modelos de generación/atracción suelen ser econométricos y lineales, es decir, toman la forma:

$$O_i = \alpha_0 + \sum_i \alpha_i x_i$$

$$D_j = \beta_0 + \sum_i \beta_i y_i$$

en que x_i son variables de generación de la zona como población, nivel de ingreso medio de la zona, etc. e y_i son variables de atracción de la zona como número de empleos disponibles, número de estacionamientos, superficie construida de comercio, etc. El uso de modelos econométricos permite estimar con bastante facilidad las variables generación y atracción.

La variable a predecir puede ser viajes totales (atraídos o generados) de una zona o una tasa de generación (o atracción) por hogar. Este último enfoque tiene algunas ventajas estadísticas que podrían permitir una mejor estimación (Douglas y Louis, 1970).

Otra opción es modelar una tasa de generación de viajes por hogar, enfoque conocido como Análisis por Categorías o por Sección Cruzada (Wootton y Pick, 1967). Se plantea que:

$$t(h) = \frac{T(h)}{H(h)}$$

en que $t(h)$ es la tasa de viajes que efectúa un hogar tipo h , $T(h)$ es el total de viajes observados que efectúa un hogar tipo h y $H(h)$ es el total de hogares observados tipo h . La idea subyacente es que cada hogar produce una cierta cantidad de viajes y por lo tanto el total de viajes que una zona generará será:

$$O_i = \sum_h a(h)_i t(h)$$

en que $a(h)_i$ es el total de hogares tipo h en la zona i . Las categorías de hogar h son secciones cruzadas de variables más desagregadas. Por ejemplo $h=1$ podría ser hogar socioeconómico rico, con 2 o más autos, habitado por más de 4 personas.

La ventaja es la evidente simpleza del modelo, que define que un cierto tipo de hogar genera y generará siempre el mismo número de viajes, por lo que sólo basta conocer el número de hogares de cada tipo en el área de estudio. Otra ventaja es que este método es independiente de la zonificación utilizada. La desventaja es que no considera la

¹⁶ En la práctica se los fuerza a coincidir, ajustando proporcionalmente el vector de viajes atraídos predichos al total de viajes originados predichos. (Ortúzar y Willumsen, 1990)

localización espacial del hogar y es por lo tanto insensible a variables como el costo de viaje.

Se han desarrollado también modelos a nivel de individuos con esta metodología (Supernak et al., 1983). Modelos de atracción con esta metodología pueden ser concebidos de similar manera, calculando la tasa de atracción de cada tipo de inmueble en una zona.

Una extensión de este tipo de modelación y que supera algunas de sus limitaciones (relacionadas con el número de datos necesarios) recibe el nombre de MCA (Multiple Classification Analysis, Análisis de Clasificación Múltiple) y ha sido presentada por Stopher y McDonald (1983).

Por otra parte, la econometría y la sección cruzada pueden ser utilizadas conjuntamente (Hall et al., 1987). Se trata de dividir los hogares en categorías usando un grupo de variables y luego estimar económicamente los viajes para cada categoría usando otro grupo de variables.

Finalmente, otra estrategia es predecir el aumento de la demanda en vez de su magnitud misma. Estos son los modelos de Factores de Crecimiento y que existen como consecuencia de la necesidad de proyectar vectores de viajes conocidos. Se plantea:

$$T_i = F(x_i)t_i$$

en que t_i es la demanda actual, T_i la demanda futura y F es el factor de crecimiento y es la función que se calibra. x_i corresponde a variables como el ingreso, la población y la posesión de automóvil y puede ser planteada como (Ortúzar y Willumsen, 1990):

$$F_i = \frac{f(x_i^d)}{f(x_i^a)}$$

en que f podría ser una función multiplicativa de los x_i y los índices d y a denotan la situación de diseño (predicción) y actual respectivamente.

3.4.3. Modelos de Distribución

Esta modelación asume la existencia de vectores O_i que describen los viajes totales generados en cada zona del área de estudio. Basándose en información que describa los elementos que potencialmente atraen viajes a distintas zonas (típicamente usos de suelo), el modelo debe predecir cómo se distribuyen los viajes generados en el área de estudio. Sin embargo, este es más bien el ideal, pues en la práctica el modelo más usado es el llamado de máxima entropía o gravitacional doblemente acotado, que no maneja directamente la información de uso de suelo sino que requiere como entrada los viajes totales atraídos D_j para cada zona. Así, este modelo debe encontrar con esa información como base, una matriz T_{ij} de los viajes realizados entre cada par de zonas. De aquí nace la necesidad de modelar viajes atraídos, como se comentó anteriormente.

El modelo entrópico doblemente acotado resulta de la solución de un problema de maximización de la entropía (Wilson, 1974) (se busca la matriz más probable que cumple restricciones de viajes atraídos y generados a nivel zonal) y tiene la forma:

$$T_{ij} = A_i B_j O_i D_j \exp(-\beta c_{ij})$$

$$A_i = \frac{1}{\sum_j B_j D_j \exp(-\beta c_{ij})} \quad B_j = \frac{1}{\sum_i A_i O_i \exp(-\beta c_{ij})}$$

Con c_{ij} es el costo de viajar entre i y j , β es interpretado como un parámetro de fricción al costo y A_i y B_j son llamados factores de balance y permiten que se cumplan las restricciones de origen y destino. Estos factores, que dependen los unos de los otros, verifican tener un punto fijo, con lo cual la convergencia y la existencia de la solución (única) están garantizadas.

Debe notarse que al calibrar este modelo lo único que se está determinando es el valor del parámetro β , ya que en predicción todas las demás variables cambian (inclusive los factores de balance). Parece extraño y simplista que un modelo de distribución de viajes tenga un único parámetro. Y es que en el fondo el modelo se fía ampliamente de los vectores de viajes atraídos para realizar su predicción, concentrándose únicamente en el efecto del costo de viaje. Todas las demás variables de decisión de los usuarios asociadas con el "dónde viajar" no son tratadas explícitamente, asumiendo que dicha información es manejada correctamente en el modelo de atracción que crea los vectores D_j e importada exógenamente. Esto implica que se tienen en realidad dos modelos interconectados de una determinada forma que por supuesto implícitamente asume ciertas independencias entre variables.

Estos supuestos pueden tener algunos efectos cuestionables. Por ejemplo, supóngase que se quiere analizar las consecuencias de un alza del costo de viajar de una zona i a otra j . El modelo acertadamente predecirá una baja de los viajes en ese par, aumentando los viajes desde i hacia otros destinos a fin de cumplir la restricción de viajes generados en i . Pero a la vez, para poder cumplir la restricción de viajes atraídos a j , deberá incrementar los viajes desde otro(s) origen(es) hacia j . Este aumento de viajes desde otras zonas no parece una consecuencia natural y razonable del aumento de costos en el par i - j . El problema radica en que los viajes atraídos a un cierto lugar son predichos con independencia de los costos de llegar a él.

En la medida que la generación y atracción (especialmente esta última) puedan ser incorporadas al modelo directamente en vez de venir exógenamente de modelos independientes, es esperable que no ocurran situaciones como la descrita. Ese sería el caso de los modelos "simplemente acotados" o "no acotados". El modelo recién expuesto es llamado "doblemente acotado", pues existen restricciones de origen y destino. La falta de una o ambas restricciones dan lugar a los otros recién nombrados. El modelo simplemente acotado en el origen, por ejemplo, toma la forma:

$$T_{ij} = A_i W_j O_i \exp(-\beta c_{ij}) \quad A_i = \frac{1}{\sum_j W_j \exp(-\beta c_{ij})}$$

en que W_j son conocidos como "factores de atracción" ("factores de generación" si fuera acotado en el destino) y es una función que depende de las variables de atracción. Los parámetros de los W_j serían estimados conjuntamente con el modelo entrópico, pasando las variables de atracción a ser endógenas en el problema de distribución. Sin embargo, esta alternativa de modelación ha sido poco aplicada. Menos aún la forma no acotada.

Otra forma de modelar distribución es usando datos de conteos de flujo. El objetivo expreso de este tipo de modelación es reducir notoriamente el costo de recolección de datos respecto del modelo gravitacional. El realizar conteos de flujos en vez de encuestas origen-destino es más simple y barato. Hay toda una línea de desarrollo con este esquema. En trabajos chilenos puede verse, por ejemplo, a Munizaga (1995). Willumsen (1981) ha escrito un resumen de este tipo de metodología.

Se plantea:

$$\begin{aligned} & \underset{t}{\text{Min}} d(t, T) \\ \text{s.a.} \quad & \sum_j p_{ij} t_j = V_i \quad \forall i \end{aligned}$$

en que t_j es el total de viajes en el par origen-destino j (variable buscada), p_{ij} es la proporción de viajes del par j que usa el arco i (dato exógeno) y V_i es el flujo vehicular observado en el arco i . t es la matriz formada por los t_j y T es una matriz objetivo a la que se desea la matriz t se parezca¹⁷. d es una función de distancia, usualmente definida en forma entrópica. Los p_{ij} son obtenidos de algún método de asignación y permanecen fijos durante la estimación de los t_j . Esto implica que se está en una situación donde la congestión no es un problema, ya que de otra forma los p_{ij} dependerían de los t_j .

Diferentes formas de definir la distancia definen los distintos modelos. Van Zuylen y Willumsen (1980) muestran el uso de distancias entrópicas y basadas en principios de minimización de la información; Spiess (1987) utiliza máxima verosimilitud.

Willumsen (1984) propone una formulación más general en que no se pide exacta igualdad con las mediciones de flujo y en que el modelador puede elegir su grado de confianza relativa en dichas mediciones frente a la matriz objetivo T y en donde de paso se requieren estimar los flujos en los arcos. Breninger-Göthe et al. (1989) refinan aún más lo anterior usando la función de distancia a nivel de arco.

La estimación matricial usando restricciones de totales generados y/o atraídos, usando conteos de flujo y otras restricciones que puedan imaginarse, pueden ser planteadas en una forma más general y puede demostrarse, como lo hacen McNeil y Hendrickson

¹⁷ Esta matriz objetivo puede ser una matriz de viajes antigua proyectada al futuro o cualquier matriz a priori elegida bajo algún criterio. Es necesaria pues de lo contrario el problema queda subespecificado.

(1985), que tales métodos son equivalentes a una regresión multivariada cuando se utilizan distancias euclidianas. Esta equivalencia es muy útil porque permite utilizar los resultados conocidos referentes a las regresiones para obtener mayor conocimiento de la solución alcanzada. Estas ventajas serían:

- Es posible acomodar todas las restricciones que se desee a partir de los datos que se posee.
- Permite el análisis de los errores en la estimación, creando, por ejemplo, intervalos de confianza para cada celda de la matriz.
- Dado que los resultados de la estimación son provenientes de una regresión multivariada, la matriz obtenida es la mejor estimación lineal insesgada para la matriz buscada.
- Por último, la implementación de este método es relativamente simple, ya que requiere de manipulaciones básicas de matrices que pueden ser encontradas en varios paquetes computacionales disponibles actualmente.

Sin embargo, todo este enfoque de estimación matricial tiene un alcance limitado, por cuanto asume inexistencia de congestión¹⁸ y no considera explícitamente variables de política o descriptivas del sistema (como usos de suelo) que permitan usar confiadamente el modelo en predicción a futuro.

Un acercamiento radicalmente distinto al problema y que no ha recibido mucha atención es el de Intervening Opportunities, introducido por Stouffer (1940) y generalizado por Schneider (1959). La presentación que se hace en el siguiente párrafo corresponde a la de Wilson (1974).

La idea es que a partir de una zona origen i un viajero evalúa las zonas destino a través de dos formas: El costo de viajar hasta ella y las oportunidades que en ella hay. "Oportunidades" son variables atractivas de la zona. Para hacer esto, el viajero hace un ranking u ordenación de las zonas con el primer criterio (costo de viaje). Luego una a una y en el orden dado por el ranking, se evalúa la probabilidad de quedar satisfecho en cada zona o la probabilidad de seguir a la siguiente.

Es claro que utilizar este modelo implica aceptar que el fenómeno de decisión sigue una mecánica bastante particular. Probablemente es ésta la razón para que este modelo no haya sido más difundido.

La modelación microeconómica también ha jugado un papel en la distribución de viajes. Modelos como el Logit han sido usados para predecir la elección de destino, como es el caso, por ejemplo, del modelo de Safwat y Magnanti (1988). Las bases del Logit y la modelación a nivel microeconómica en general serán presentados en la sección de partición modal donde han jugado un rol mucho más activo históricamente.

Por último y tal como en generación, también existen modelos de factores de crecimiento para distribución, en que matrices de viaje base son expandidas vía

¹⁸ Existen otros modelos de esta familia que integran asignación y distribución para evitar la limitación del supuesto de la congestión como Nguyen (1983) y Fisk y Boyce (1983).

multiplicación por un escalar. Por supuesto el resultado en general será limitado por cuanto es dudoso que los viajes crezcan igualmente en todos los pares origen-destino. Un enfoque un poco más general es el de disponer de un factor de crecimiento para los viajes generados y otro para los viajes atraídos. Más general aún, pueden tenerse factores de crecimiento x_i para cada zona origen i y factores de crecimiento y_j para cada zona destino j . En este caso se han desarrollado diversas metodologías para encontrar la matriz futura. El más conocido es el de Furness (1965) que usa un método iterativo con factores de balance. Wilson (1974) planteó una forma equivalente a este método.

El problema con la modelación con factores de crecimiento es que asumen un crecimiento vegetativo que mayormente no altera la estructura de los viajes y que es insensible a cambios profundos en el sistema como cambios en los costos de viaje o el uso del suelo.

3.4.4. Modelos de Partición Modal

Estos modelos deben distribuir los viajes en los distintos modos de transporte disponibles. En los últimos años el enfoque microeconómico se ha convertido en la estrategia dominante para resolver este problema, por lo que aquí sólo se revisará este paradigma.

La Teoría de la Utilidad Aleatoria nace en la psicología a partir de Thurston (1927). Para una exposición en profundidad, véase Ben Akiva y Lerman (1985). El enfoque asume que un individuo n que se ve enfrentado a una decisión de modo, elige aquel que le reporta mayor beneficio. Hay tres elementos:

1. Existe un conjunto de modos alternativos que el individuo considera para su elección.
2. Existe una función de utilidad U_{in} de carácter ordinal que describe la utilidad que le reporta al individuo n la elección del modo i . Esta función no es observable por el modelador y sólo es una representación que hace éste de la forma como el individuo considera sus opciones.
3. La elección es discreta. En la economía clásica los bienes son tratados de manera que siempre el individuo puede elegir cantidades continuas de cada bien. Aquí él debe elegir una y sólo una alternativa y consume una sola unidad del bien.

Se plantea que la utilidad de cada modo de transporte está dada por sus características. Esto implica una utilidad de la forma $U_{in} = U_{in}(X_1 \dots X_k)$, en que los x_l son atributos observables de los modos. La palabra observable es importante. El modelador asume que hay ciertos efectos que inciden en la utilidad y que él no es capaz de observar o de medir en forma correcta. Así, se plantea entonces que la utilidad tiene un término determinístico observable V_{in} y otro aleatorio e_{in} , es decir, $U_{in} = V_{in} + e_{in}$.

V_{in} usualmente se define como una combinación lineal de los atributos de los modos. La partición modal estará dada, por lo tanto, por la probabilidad de elección de los distintos modos. La probabilidad con que el individuo n elige el modo i es:

$$P_{ni} = P(U_{ni} \geq U_{nj} \quad \forall j \neq i) = P(U_{ni} \geq \max_{j \neq i} \{U_{nj}\}) = P(V_{ni} + e_{ni} \geq \max_{j \neq i} \{V_{nj} + e_{nj}\}) \quad [\text{III-1}]$$

Basta con definir la distribución de probabilidad de los errores para obtener las probabilidades buscadas. Los distintos modelos que existen nacen a partir de diferentes supuestos sobre dichas distribuciones de probabilidad. Se detalla en lo que sigue algunas de las formulaciones más usadas, pero antes vale la pena hacer algunos comentarios válidos para todos estos modelos:

1. Se asume una regla de decisión por parte de los individuos que pondera simultáneamente las diferentes características de un modo para definir sus preferencias. Este enfoque puede no representar adecuadamente la psiquis de los individuos. Bien podría ser que la forma en que las personas hacen su elección corresponda a un proceso iterativo de descarte a través de comparar cada atributo con un mínimo aceptable. Tales reglas han sido diseñadas y estudiadas y representan otras formas de elección, como la "eliminación por aspectos" (Tversky, 1972).
2. Dado que este tipo de modelación es desagregada, existen problemas cuando se desea agregar los resultados, lo que es necesario para hacer predicciones sobre el comportamiento de los viajes a nivel de zona. Los errores que nacen en este proceso de agregación son conocidos como "sesgo de agregación" y han sido estudiados, por ejemplo, por Koppelman (1975).
3. La racionalidad económica en que se basa el modelo ha sido criticada a la luz de diversas experiencias que evidencian que las personas no se rigen necesariamente por un sistema de beneficio máximo. McFadden (1999) revisa este tema en profundidad y sugiere una mayor atención a otras reglas de decisión.

El Logit Multinomial (MNL), ampliamente difundido, asume que los errores son IID-Gumbel (independientes e idénticamente distribuidos Gumbel). Se puede demostrar (por ejemplo Ben-Akiva y Lerman, 1985) que la ecuación [III-1] con estos supuestos resulta en:

$$P_{ni} = \frac{e^{\mu U_{ni}}}{\sum_j e^{\mu U_{nj}}}$$

en que μ es un factor de escala de la función distribución Gumbel.

La estimación del MNL se realiza maximizando la verosimilitud. McFadden (1974) mostró que si la solución existe, es única y que los parámetros así obtenidos son consistentes, asintóticamente normales y asintóticamente eficientes.

El logit multinomial ha sido ampliamente usado en la modelación de partición modal, pero es necesario tener presentes los supuestos sobre los cuales está construido para

evitar resultados incorrectos. Un ejemplo de esto es el error que se comete al olvidar que las alternativas no debe estar correlacionado. La "paradoja de los buses de colores" ejemplifica este tipo de error. En él se describe cómo la probabilidad de elegir el bus aumenta sólo con separar los buses por distinto color. La contradicción viene del hecho que los buses con distinto color son alternativas correlacionadas, pues comparten todas las características no observadas. Este problema, que en forma más general proviene de una estructura muy exigente para las correlaciones, ha generado la necesidad de buscar otros modelos que puedan manejar situaciones más complejas, en especial si se piensa que los modos efectivamente disponibles en la vida real comparten ciertas características típicamente no modeladas explícitamente tal como la comodidad en las opciones del transporte público.

La búsqueda por formas tipo Logit menos restrictivas que el multinomial ha dado origen a modelos como el Logit Jerárquico, de bastante uso. Este modelo admite ciertas correlaciones entre alternativas, aunque mantiene el supuesto de la homoscedasticidad, es decir, que todas las alternativas tienen igual varianza. Uno de los más recientes, por otra parte, es el a veces llamado Logit Mixto, que permite aún más flexibilidad en la definición de las correlaciones a través de la definición de más términos de error. (Ben-Akiva y Bolduc, 1996; Brownstone y Train, 1999; Álvarez, 2001)

El Probit nace de suponer que los errores tienen una distribución normal. Ningún otro supuesto adicional es necesario, lo que permite la más amplia flexibilidad en cuanto a correlación de alternativas. En este caso, la probabilidad de la ecuación [III-1] se transforma en

$$P_i = \int_{e_1=-\infty}^{V_i-V_1+e_i} \dots \int_{e_{i-1}=-\infty}^{V_i-V_{i-1}+e_i} \int_{e_i=-\infty}^{\infty} \int_{e_{i+1}=-\infty}^{V_i-V_{i+1}+e_i} \dots \int_{e_K=-\infty}^{V_i-V_K+e_i} N(e_1, \dots, e_K) de_1 \dots de_K$$

en que N es la función densidad normal conjunta multivariada de los errores. Desafortunadamente esta integral no puede ser resuelta analíticamente, lo que lo hace difícil de estimar. Con las capacidades computacionales actuales sigue siendo un modelo costoso de utilizar y con dificultades para recoger los parámetros relevantes (Álvarez, 2001).

Los anteriores son los más usados y populares en modelación desagregada, aunque hay otras propuestas como la incorporación de la variación de gustos. Este efecto se obtiene definiendo, además de un término de error aleatorio, los coeficientes que acompañan a las variables observables (en la utilidad) aleatorias también. Este tipo de modelos se conocen como modelos con coeficientes aleatorios.

Para una revisión de modelos de utilidad aleatoria, su estimación y las consecuencias de asumir distintas distribuciones de probabilidad para los errores, véase a Álvarez (2001).

3.4.5. Modelos de Demanda Integrados

Como se discutió al inicio de la revisión bibliográfica, integrar modelos tiene una lógica natural, ya que se está capturando en un solo modelo varios o todos los efectos de interacción entre las decisiones de los individuos. La solución así encontrada representaría un equilibrio u óptimo global de las decisiones que es más factible que sea el que realmente los usuarios están alcanzando.

A partir del modelo de distribución presentado anteriormente, fácilmente puede incorporarse partición modal usando costos a nivel de modo. El resultado obtenido es (Wilson, 1974):

$$T_{ijk} = A_i B_j O_i D_j \exp(-\beta c_{ijk})$$

$$A_i = \frac{1}{\sum_{j,k} B_j D_j \exp(-\beta c_{ijk})} \quad B_j = \frac{1}{\sum_{i,k} A_i O_i \exp(-\beta c_{ijk})}$$

Los comentarios y críticas a este modelo enunciadas con anterioridad para el caso de distribución pura son válidos aquí también.

Los modelos totalmente integrados son quizás los más escasos. La forma propuesta por Manheim (1973) es probablemente la más común adoptada en estudios urbanos y puede ser escrita en la forma:

$$T_{nijkr} = G_{in} M_{ink} M_{ijnk} M_{nijkr}$$

en que G_{in} es el total de viajes generados en la zona i por el tipo de hogar (o individuo) n , M_{ink} es la proporción que elige el modo k , M_{ijnk} es la proporción que elige el destino j y finalmente r denota elección de ruta. Estas proporciones pueden por supuesto ser vistas también como probabilidades de elección. T_{nijkr} es el total de viajes para la tipología n,j,k,i,r .

Durante los 70's existieron algunas aplicaciones de esta formulación y de formulaciones similares. Es claro que en este modelo cada proporción y los G_{in} son submodelos, de modo que se trata de una modelación separada y cuya resolución debe ser determinada a ser secuencial o simultánea, en donde los comentarios hechos a principios del capítulo de revisión bibliográfica son válidos. El modelo de Manheim es principalmente una abstracción teórica más que un modelo concreto.

Safwat y Magnanti (1988) proponen un modelo conjunto de generación, distribución, partición modal y asignación en que también hay una modelación separada para cada etapa y el aporte importante está en el proceso de resolución simultánea en donde se aprovechan varios resultados anteriores para asegurar consistencia interna y existencia de solución.

3.4.6. Modelos de Demanda Directa

Aunque la manera rigurosa de estimar demanda para un determinado destino, modo, ruta u otro elemento particular debe considerar la interacción de éste con los demás elementos del mercado del transporte, a veces es conveniente simplificar el problema a fin de poder tener de manera simple una estimación de los viajes. Ese es típicamente el caso para una empresa de transporte, por ejemplo. Es por ello que existen los llamados modelos de demanda directa o estructural que entregan exclusivamente la demanda para alguno de los elementos del sistema de transporte haciendo poca o nula referencia al resto del sistema.

Uno de los ejemplos de este tipo de modelación corresponde al realizado en Chile para el Metro de Santiago. En el trabajo de Jara-Díaz y Paredes (1993) se propone una función de demanda por metro de la forma:

$$Y_M = f(\text{Emp}, \text{PP}, \text{Ne}, \text{Au}, \text{Y}_t, \text{Y}_e, \text{P}_m, \text{P}_b)$$

en que Y_M es la demanda por metro, Emp es el número de empleos en la ciudad, PP es cantidad de agua caída (se incorpora un efecto climático en la demanda), Ne es el número de estaciones del metro, Au es una medida de acceso al automóvil por parte de la población, Y_t e Y_e don efectos tendenciales y estacionales de la demanda, P_m es la tarifa del metro y P_b la tarifa de los buses.

3.4.7. Otros Aspectos de la Modelación de la Demanda de Transporte

Paralelamente a lo que es la modelación dividida del proceso de decisión de viajes dentro de la cual se han mostrado hasta aquí diferentes modelos, existen otras consideraciones que se han incorporado al análisis. Tal es el caso de la modelación en serie de tiempo, que permite ver la evolución de la costumbre de los usuarios o los procesos de encadenamiento de viajes que permiten tener una visión más compleja de las decisiones al evitar aislar cada viaje (Ortúzar y Willumsen, 1990). Por otra parte, el análisis de sistemas de viaje más complejos como las elecciones de modos mixtos también han sido exploradas, donde conceptos como el park-and-ride y kiss-and-ride han sido introducidos. Según Ortúzar (1980) el problema de los modos mixtos no ha podido ser apropiadamente abordado. Analizando alrededor de 30 estudios realizados en los 70's, concluye que en ninguno este aspecto fue tratado adecuadamente.

3.5. Redes Neuronales y Transporte

Las redes neuronales no son la única herramienta relacionada con inteligencia artificial que se ha usado en el ámbito del transporte. Algoritmos genéticos y lógica difusa son otros paradigmas presentes. En este aspecto, puede verse a Teodorovic (1999) en su revisión de lógica difusa aplicada a transporte.

En cuanto a redes neuronales, Dougherty (1995) realizó una revisión de los estudios llevados a cabo en transporte usando esta herramienta. Redes Neuronales en Transporte

El interés por las redes neuronales en el ámbito del transporte es muy nuevo. Dougherty (1995) revisa 52 trabajos, 40 publicados en los 90's y admite que encontrar publicaciones en la década anterior es difícil. El mencionado autor nos provee de una tabla (citada aquí como Tabla 3.5.1) donde es posible observar las áreas del transporte donde las redes han sido utilizadas y el número de trabajos en cada área, dándonos una idea del balance de intereses de los investigadores.

La tabla no es del todo representativa, ya que mucho del trabajo en las áreas marítima y aérea se ha mantenido en secreto, sea por razones militares o comerciales.

Algunos ejemplos de las aplicaciones a que hace referencia la tabla:

1. En el área Comportamiento del Conductor, se pregunta a pilotos que elijan ruta basados en una serie de variables y luego se utiliza esta misma información (variables y decisión) para entrenar una red que al final del proceso pueda hacer una elección de ruta.
2. En la Estimación de Parámetros, redes pueden ser entrenadas para que produzcan una estimación del tiempo de viaje en un arco dada información de flujo y características de la vía.
3. En Detección/Clasificación de Vehículos una red neuronal puede ser entrenada para clasificar un vehículo según variables que tienen que ver con sonido detectado al paso de él.
4. En Políticas de Transporte y Economía, se entrenó una red para que, dados datos de una vivienda como tipo de construcción y superficie construida, en conjunto con datos de contaminación local, entregue una estimación de su precio, permitiendo recuperar el efecto de la polución en el precio.

La tabla 3.5.2 muestra los Modelos utilizados en los trabajos estudiados por Dougherty (la suma vertical es mayor a 52, pues algunos trabajos utilizan más de un paradigma, en cuyo caso aparecen en ambas categorías).

Tabla N° 02.- Distribución de Trabajos con Redes Neuronales en Transporte

AREA	NUMERO DE TRABAJOS
Comportamiento del Conductor/Vehículos Autónomos	12
Estimación de Parámetros	7
Mantenimiento de Pavimentos	6
Detección/Clasificación de Vehículos	5
Análisis de Patrones de Tráfico	5
Operación de Flotas de Carga	4
Predicción de Tráfico	4
Políticas de Transporte y Economía	2
Transporte Aéreo	2
Transporte Marítimo	2
Vehículos Submarinos	1
Operación de Metro	1
Control de Tráfico	1
TOTAL	52

Fuente: Dougherty (1995)

Tabla N° 03.- Modelos Usados

MODELOS	NUMERO
Retropropagación	36
Learning Vector Quantisation	7
Teoría de la Resonancia Adaptiva	4
Mapas Auto-Organizados	2
ADALINE	1
Redes de Hopfield	1
Funciones Basales	1
Contrapropagación	1
Máquina de Boltzmann	1

Fuente: Dougherty (1995)

Las críticas que hace Dougherty sobre el trabajo que se ha estado realizando con redes neuronales pueden resumirse así:

1. La idea de "caja negra" en las redes neuronales persiste fuertemente e incluso el concepto de aprendizaje de la red se mantiene en una forma muy abstracta, lo que probablemente refleja un conocimiento muy poco acabado de la herramienta que se está utilizando. En este sentido, poco y nada se hace a nivel de arquitectura de la red, utilizándose formas estándares. A su vez, poco esfuerzo hay en la interpretación de una red ya entrenada, siendo pobres los análisis del modelo.
2. En la mayoría de los trabajos, las redes juegan un rol como sub-sistemas de un modelo más grande o un algoritmo mayor. El rol de las redes sería de filtros de datos o preprocesamiento de datos. Esto sería así por una falta de determinación a usar las redes como el modelo en sí, posiblemente explicado por un conocimiento aún magro de ellas.
3. Falta mucho aún por decir en el contexto de la comparación con métodos tradicionales. La ausencia de medidas de ajuste (entre lo observado y lo predicho) que puedan utilizarse tanto en las redes como en los modelos tradicionales hace que las conclusiones sobre si las redes son o no mejores que tales métodos sean imprecisas e incluso a veces contradictorias.
4. Relacionado con los puntos 1 y 3, no basta en muchos casos que las redes tengan un mejor desempeño en predicción que un método tradicional, ya que lo que también se desea es un entendimiento mayor del fenómeno en estudio. La naturaleza de caja negra de las redes podría no competir con la riqueza de otros modelos donde cada parámetro tiene una interpretación, como utilidades, costos marginales y otros.

5. El paradigma de las redes neuronales estaría llegando a un punto donde hay bastante experimentación y poca implementación real. Este último paso sería clave para realmente poder generar juicios sobre ellas.
6. Señala que se debe tener muy presente que la cantidad de datos requerida para entrenamiento y testeo (que normalmente se admite como grande) puede volver poco atractivas las redes.
7. Dougherty observa que en muchos investigadores tiende a primar un entusiasmo exagerado en el paradigma de las redes más que criterios analíticos.
8. Por último, espera que el uso de los distintos paradigmas se diversifique más (dada la alta concentración en las redes de retropropagación) y ello ayude a poder definir qué tipo de paradigma es mejor para los distintos tipos de problemas.

Tras haber revisado para esta tesis los trabajos de redes aplicados a demanda es posible coincidir con Daugherty en varias de las críticas que realiza. En particular, se encuentra en dichos trabajos que un análisis de los modelos neuronales que se plantean no existe, utilizándose las redes principalmente como máquinas que toman ciertos datos y arrojan otros, confiando ampliamente en que las características teóricas de las redes podrán manejar cualquier complejidad que la demanda de viajes pueda tener (es decir, la idea de caja negra). La falta de medidas de ajuste que puedan ser utilizadas en redes como en métodos tradicionales también es un punto de coincidencia. Finalmente, también se está de acuerdo en cuanto a que los autores muestran más entusiasmo por las capacidades potenciales de las redes que disposición a criticarlas.

Por otra parte, no se coincide con Daugherty en lo que se refiere a la cantidad de datos necesaria para alimentar redes en comparación con métodos tradicionales. En los trabajos revisados este aspecto no tuvo relevancia alguna.

3.6. Estudios en Demanda de Transporte

En esta sección se revisarán los escasos trabajos asociados específicamente a demanda de transporte usando redes neuronales. Pese a estar compuesta la demanda por las ya mencionadas generación, distribución y partición modal, los esfuerzos de los investigadores se han centrado preferentemente en la última de ellas, la partición modal, y en especial con el interés de comparar el desempeño de las redes con el popular Logit. Ese es el caso de Nijkamp et al. (1996), Nijkamp et al. (1997), Abdelwahab y Sayed (1998), De Carvalho et al. (1998) y Hensher y Ton (2000). Mozolin, Thill y Usery (1999) comparan por otra parte las redes neuronales con un modelo gravitacional para la distribución de viajes. Fischer y Gopal (1994) realizaron un trabajo con redes para predecir flujos de telecomunicaciones (llamadas telefónicas y faxes) y por su gran similitud con la distribución de viajes será analizado también.

3.6.1. Metodología y Prácticas

En Nijkamp et al. (1996) se define una metodología de trabajo para redes neuronales de retropropagación de una capa escondida. En esencia, esta metodología define tres fases: a) Definición de arquitectura; b) Entrenamiento; c) Validación d) Test. La primera fase dice relación con la definición del número de neuronas de la red. El número de nodos de entrada y salida queda dado por el número de variables independientes y dependientes del problema respectivamente, mientras que el número de neuronas en la capa escondida se obtendría principalmente a través de varias pruebas experimentales previas con distintas cantidades. En forma más general, en realidad se procede con esta metodología varias veces, cada una con un número distinto de neuronas en la capa escondida.

3.6.2. Experimentos de Distribución

Mozolin, Thill y Usery (1999) hicieron experimentos comparativos entre redes neuronales de retropropagación y el modelo entrópico doblemente acotado para la distribución de viajes. Usando información de viajes al trabajo del área metropolitana de Atlanta, calibran el modelo gravitacional y varias redes neuronales con datos de 1980 e intentan predecir otro conjunto de datos de 1990. Al igual que los trabajos anteriores, las redes utilizadas tienen una sola capa escondida (con 5, 20 y 50 nodos) y más allá de esto la variable arquitectura no juega ningún rol. Los datos de entrada son los viajes generados O_i , atraídos D_j y la distancia d_{ij} y la red debe predecir el correspondiente número de viajes T_{ij} . La arquitectura de red utilizada, válida también para el trabajo de Fischer y Gopal (1994) que se comenta más abajo, se presenta en la figura 6.

El trabajo menciona dos fuentes de datos, una más detallada que la otra. En el caso más agregado la red neuronal es algo inferior al modelo tradicional, mientras que en el caso desagregado la red neuronal se desempeña notoriamente peor que el modelo entrópico. Los autores se centran más en este último caso y concluyen que su trabajo es una evidencia rotunda de la superioridad de la metodología tradicional respecto de las redes neuronales ("overwhelming evidence", según el artículo original). En busca de posibles causas para esto, los autores centran su atención en un posible efecto de sobre-ajuste de las redes, el cual parecen encontrar producto de un mapeo tridimensional que muestra que la red tiene un comportamiento con ciertas ondulaciones que logra ajustar bien a los datos específicos con los que fue entrenada, pero que no le permitirían realizar una buena predicción en las regiones entre datos. Suponen que el problema a su vez se origina por el excesivo número de neuronas (20 y 50) ya que en el caso de 5 neuronas la red no presenta estos comportamientos. Haciendo pruebas con redes con menos neuronas evitan la formación de las ondulaciones (lo que interpretan como una solución al sobre-ajuste), pero que va en desmedro de la capacidad predictiva de la red. Concluyen así que las redes presentan un arma de doble filo, en que al crecer el número de neuronas se aumenta su capacidad para ajustar datos (los de entrenamiento), pero eso finalmente no sería útil porque representaría un sobre-ajuste que en el caso de predecir a futuro la convertiría en un mal modelo. Este mismo tipo de efecto, proponen, sería el que hace que la red tenga un buen desempeño en el año base y sería la razón para que en el resto de la literatura las redes neuronales se consideren buenas herramientas, cuando

en realidad lo que se desea es hacer predicciones a futuro y en ese caso realmente fallarían. Esta es, para los autores, la principal conclusión del trabajo, mostrando que la diferencia temporal entre entrenamiento y predicción no puede ser manejada adecuadamente por la red neuronal.

Aseveran también, utilizando los gráficos tridimensionales, que las soluciones de la red neuronal tienden a reproducir la función sigmoide que está dentro de las neuronas de la red. De ahí sugieren que una red neuronal sólo podría manejar fenómenos que sigan la forma sigmoide. Con esto concluyen que la red neuronal no es una buena herramienta para análisis espacial por cuanto la forma de "S" de la sigmoide no es útil para un fenómeno que es más propio de funciones como la exponencial como es el efecto de la distancia.

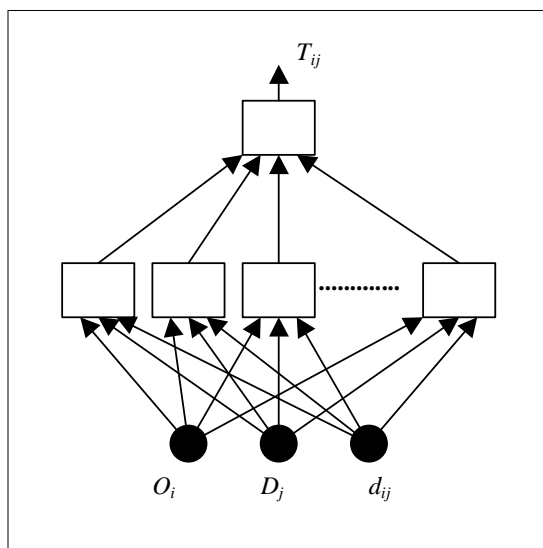


Figura N° 09: Estructura de la red utilizada para distribución. d_{ij} es la distancia entre i y j , O_i son los viajes generados en i , D_j son los viajes atraídos en j y T_{ij} son los viajes entre i y j . En el caso de Fischer y Gopal (1994) O_i es una medida del potencial para generar llamadas telefónicas, D_j un potencial de atraerlas y T_{ij} el número de llamados entre i y j .

Parece bastante apresurada la conclusión acerca de que las redes sólo podrían mapear fenómenos que sigan la forma funcional de la función de activación usada en las neuronas. Con experimentos bastante simples se puede rápidamente desmentir tal conclusión, como se podrá ver en el capítulo siguiente de esta tesis.

La "evidencia rotunda" de la superioridad del método tradicional, pese a que en esta tesis no se considera que la evidencia lleva una conclusión tan taxativa, podría ser explicada por una característica clave de la forma de modelación neuronal propuesta por estos autores. Ella se comentará en algunos párrafos más adelante tras describir el otro experimento de distribución que posee también la misma característica.

Fischer y Gopal (1994) realizaron un experimento con redes de retropropagación para predecir flujos de telecomunicaciones entre zonas de Austria y compararon sus resultados con un modelo del tipo gravitacional. Por su similitud con la distribución de viajes, vale la pena describirlo. Los autores utilizaron a manera de punto de comparación el modelo:

$$F_{ij} = KA_i^a B_j^b D_{ij}^d$$

en que A_i es una variable que mide el potencial de generar llamadas telefónicas en i y B_j es un potencial de atracción de llamadas telefónicas en j . D_{ij} es la distancia entre las zonas i y j . K , a , b y d son constantes. La red neuronal es una de retropropagación con una capa escondida de 30 neuronas que fue la elegida tras probar distintas cantidades. La red recibe como entradas A_i , B_j y D_{ij} y predice F_{ij} . Repitiéndose lo que encontraron los investigadores en el caso de la comparación con Logit, Fisher y Gopal tuvieron contradicciones en los indicadores que escogieron (R^2 y error cuadrático medio) y no consideraron la posibilidad de incidir mayormente en el diseño de la red. En cuanto a resultados, éstos mostraron una sutil mejor predicción de la red frente al modelo gravitacional.

En los dos casos mencionados de distribución de viajes, los autores no parecen detectar un problema de fondo que tiene su formulación de redes neuronales. Y es que al predecir el flujo en el par $i-j$, la red sólo recibe como inputs datos de las zonas i y j , lo cual implica que las demás zonas son irrelevantes para la predicción en ese par. Este es claramente un supuesto de independencia muy fuerte. A manera de ejemplo, si una zona k experimentara un fuerte crecimiento comercial, lo esperado sería que los viajes en el par $i-j$ disminuyeran en favor de un incremento en el par $i-k$. Esta *competencia* entre destinos es totalmente ignorada en la modelación, ya que la red no es capaz de "ver" la situación de k al predecir para $i-j$.

El ejemplo anterior muestra que el tipo de formulación elegida para las redes neuronales presenta una importante desventaja en términos de predicción para situaciones cambiantes del área de estudio. Es probablemente por esto que Mozolin, Thill y Usery (1999) encuentran un bastante peor desempeño de las redes frente al modelo gravitacional, ya que en su caso buscan predecir a futuro en donde, claramente, la situación de la ciudad ha cambiado notoriamente.

Tan importante como lo anterior, es muy probable que durante el entrenamiento la red se haya visto expuesta a información incoherente, lo cual mermó el aprendizaje y por lo tanto la capacidad predictiva final de la red. En efecto, supóngase que existen en el conjunto de entrenamiento dos pares de entrenamiento $p1$ y $p2$ asociados a los pares origen-destino $i-j$ y $a-b$ respectivamente. Supóngase ahora que las variables de entrada en los pares $p1$ y $p2$ son similares¹⁹. Es perfectamente factible que los viajes generados de i a j sean mucho mayores que los de a a b , simplemente porque llegar desde i a otros destinos distintos de j es caro en comparación con los costos que desde a existen a otros destinos distintos de b (la zona i está peor posicionada que a en términos del sistema de transporte). En tal caso se estaría pidiendo a la red, al presentarle ambos pares durante el entrenamiento, que genere salidas totalmente diferentes para dos inputs que son prácticamente el mismo. Es obvio que ante tal situación la red no podrá aprender adecuadamente, con la consiguiente pérdida en capacidad predictiva.

¹⁹ Por ejemplo, las zonas i y a pueden estar en lugares muy diferentes de la ciudad, pero ser similares en su tamaño y composición socioeconómica y las zonas j y b pueden ser la misma (por ejemplo el centro de la ciudad), cumpliéndose que llegar desde i al centro tiene un costo similar que llegar de a al centro.

CAPÍTULO IV

MODELO NEURONAL DE GENERACION DE VIAJES

4. Modelo Predictivo de Generación de Viajes

Para realizar un modelo de generación, se utilizaron las categorías de Kohonen para construir conjuntos de entrenamiento, validación y testeo para entrenar una red de retropropagación a modo de modelo de generación.

La idea es entrenar una red de retropropagación para que a partir de variables agregadas de una zona de Tarapoto, diera una estimación del total de viajes generados por ella. Se entrena y valida una red con ciertas zonas y se testea con otras.

Las variables que se poseen para este experimento son la población (desagregada en "trabajadores", "estudiantes" y "otros"), el número total de vehículos, el número total de hogares, el ingreso medio zonal de los hogares, Consumo Eléctrico y el total de viajes generados. Por supuesto, las 7 primeras serán usadas como **variables independientes (variables de entrada a la red)** y la última como variable dependiente (**variable que se desea que la red reproduzca en la salida**). Se poseen, provenientes de la EOD2014 (Encuesta Origen Destino de Tarapoto 2014), 25 zonas con esta información. No se dará un análisis descriptivo de estos datos, ya que para efecto de los experimentos que se deseaban realizar, era irrelevante el dato mismo o las medidas que se utilizaran para caracterizar una zona, ya que lo que se quiere es probar características de la redes.

4.1. Fuentes de Información

La principal fuente de información fue la Encuesta Origen Destino de Tarapoto 2014 (EOD2014) realizada por los alumnos de la Maestría en Ciencias con mención en Ingeniería de Transporte que representa una muestra de los hogares de Tarapoto y los viajes que sus integrantes realizan. Adicionalmente, con ayuda del plano de uso de Suelo de la ciudad de Tarapoto fue posible obtener información respecto de utilización del suelo de Tarapoto por las distintas actividades (industria, comercio, habitación, etc). Se creó zonas de tránsito teniendo en cuenta el uso de suelos y la red de arterias urbanas de la ciudad.

A continuación se presenta un resumen por zona de las variables independientes y dependientes fuentes de datos:

Tabla N° 04.- Resultados de EOD-Tarapoto por zonas de tránsito.

Registro # (ZONA)	Vehículos Totales	Ingreso medio de los hogares de la zona (en escala 1-7*)	Población de Trabajadores	Población de Estudiantes	Población " Otros "	Consumo Electrico	Total de hogares	Viajes Generados
112	80.00	2.98	305	70	8	63.57	103	880
113	40.00	2.27	127	67	9	32.18	64	196
114	97.00	1.79	289	99	0	52.55	120	753
115	65.00	1.93	189	56	0	51.08	89	360
116	27.00	3.14	57	16	2	44.73	21	395
117	52.00	2.45	146	47	3	54.75	58	433
118	41.00	1.51	155	89	9	28.34	78	308
119	124.00	2.97	380	126	0	65.15	145	967
120	87.00	2.06	247	86	0	47.5	125	636
121	22.00	2.00	75	25	0	46.57	38	1779
122	3.00	1.40	27	14	0	38.33	15	254
123	152.00	2.38	396	113	7	55.34	156	767
124	23.00	1.76	111	41	0	30.04	45	180
125	118.00	2.08	332	156	3	39.68	135	514
126	3.00	1.50	5	2	0	49.5	4	97
127	7.00	2.82	24	11	0	26.46	11	66
128	118.00	3.26	291	80	27	66.77	118	709
129	90.00	3.10	148	48	0	36.57	86	682
130	54.00	1.73	143	54	12	36.28	60	159
131	68.00	1.82	204	76	0	39.77	94	337
132	82.00	1.70	231	80	7	43.44	110	1684
133	111.00	1.81	296	97	0	40.68	125	353
134	126.00	3.19	253	91	10	63.5	94	1171
135	67.00	2.01	182	92	5	41.13	87	534
136	61.00	2.03	262	81	4	51.11	113	784

*Se trata de las categorías de ingreso de la EOD2014- Tarapoto: 1: menos de S/.600, 2: S/.601 - 1000, 3: S/.1001 - 2000, 4: S/.2001 - 3000, 5: S/.3001 - 4000, 6: S/.4000 - 7000, 7: Mas de S/.7000

Cabe indicar que antes de definir el conjunto de entrenamiento para la Red Kohonen y la Red de Retropropagación, se estableció para cada variable entre si un análisis de regresión lineal, esto se debió a que podría suceder que el Coeficiente de correlación múltiple entre las variables sea cercano al valor uno, esto indicaría que aquellas variables que tengan un Coeficiente de correlación múltiple alto, se interpretara que las variables tienen correlación entre si y esto generara que la red no modele correctamente la demanda de transporte, de esta manera se utilizaran solo las variables con poca correlación del cuadro N°01, después del análisis resultado lo siguiente:

Tabla N° 05.- Variables independientes y dependientes que tienen poca correlación.

Registro # (ZONA)	Vehículos Totales	Ingreso medio de los hogares de la zona (en escala 1- 7*)	Población de Estudiantes	Población " Otros "	Consumo Electrico	Total de hogares	Viajes Generados
112	80.00	2.98	70	8	63.57	103	880
113	40.00	2.27	67	9	32.18	64	196
114	97.00	1.79	99	0	52.55	120	753
115	65.00	1.93	56	0	51.08	89	360
116	27.00	3.14	16	2	44.73	21	395
117	52.00	2.45	47	3	54.75	58	433
118	41.00	1.51	89	9	28.34	78	308
119	124.00	2.97	126	0	65.15	145	967
120	87.00	2.06	86	0	47.5	125	636
121	22.00	2.00	25	0	46.57	38	1779
122	3.00	1.40	14	0	38.33	15	254
123	152.00	2.38	113	7	55.34	156	767
124	23.00	1.76	41	0	30.04	45	180
125	118.00	2.08	156	3	39.68	135	514
126	3.00	1.50	2	0	49.5	4	97
127	7.00	2.82	11	0	26.46	11	66
128	118.00	3.26	80	27	66.77	118	709
129	90.00	3.10	48	0	36.57	86	682
130	54.00	1.73	54	12	36.28	60	159
131	68.00	1.82	76	0	39.77	94	337
132	82.00	1.70	80	7	43.44	110	1684
133	111.00	1.81	97	0	40.68	125	353
134	126.00	3.19	91	10	63.5	94	1171
135	67.00	2.01	92	5	41.13	87	534
136	61.00	2.03	81	4	51.11	113	784

*Se trata de las categorías de ingreso de la EOD2014- Tarapoto: 1: menos de S/.600, 2: S/.601 - 1000, 3: S/.1001 - 2000, 4: S/.2001 - 3000, 5: S/.3001 - 4000, 6: S/.4000 - 7000, 7: Mas de S/.7000

4.2. Metodología para Calibrar y Aplicar el Modelo

La metodología usual de trabajo con redes de retropropagación tiene importantes dosis de pruebas y error, por ejemplo con el número de neuronas de la capa escondida. Si a esto uno suma las nuevas alternativas, tales como probar con diferentes funciones de activación, entonces el espacio de posibilidades de exploración es bastante amplio. En principio bastaría sólo con enumerar todas las herramientas que el modelador posee para buscar la mejor red, aceptando que la metodología de trabajo será por ende una exploración de estas posibilidades. Pero aquí se quiere ir un poco más lejos y plantear una metodología más definida. Ella puede plantearse en forma gruesa como:

- I. Entrenamiento de un mapa de Kohonen y formación, a partir de éste, de conjuntos de entrenamiento, Validación y testeo.
- II. Diseño y Entrenamiento, Validación y Testeo de la red de retropropagación (modelo).
- III. Creación y Entrenamiento de la Red
- IV. Simulación de la Red

4.2.1. Etapa I: Mapa de Kohonen y Formación de Conjuntos de Entrenamiento, Validación y Testeo

- I.1. Entrenamiento de un mapa de Kohonen.
- I.2. Análisis de Datos. Aprovechando el mapa de Kohonen es posible analizar la información en busca de outliers y otras características que se considere relevante.
- I.3. A nivel de cada categoría de Kohonen, se dividen los datos para formar conjuntos de entrenamiento, validación y testeo.
- I.4. Se analiza el conjunto de entrenamiento, validación y testeo para ser entregado a la red de Retropropagación.

4.2.2. Etapa II: Diseño y Entrenamiento, Validación y Testeo de la red de retropropagación (modelo)

- II.1. Matrices de Datos de entrada de la Red
- II.2. Normalización y depuración de los datos
- II.3. Datos para el entrenamiento, validación y test de la Red Neuronal Artificial

La calidad del modelo depende en forma relevante de los siguientes factores:

- a) Función de Activación usada en las neuronas.

- b) Prueba con el número de neuronas de la capa de entrada y la capa oculta.
- c) Secuencia de presentación de pares de entrenamiento a la red.

4.2.3. Etapa III: Creación y Entrenamiento de la Red

III.1. Creación de los Datos

III.2. Entrenamiento de la Red Neuronal Artificial

4.2.4. Etapa IV: Simulación de la Red

Ante una observación para la cual se desea hacer una predicción, se debe:

IV.1. Simulación de la Red Neuronal Artificial

IV.2. Conversión de la respuesta de la simulación a valores originales

IV.3. Comparación del resultado entregado por la red y el valor esperado

La metodología completa se reproduce a continuación en forma esquemática:

Etapa I: Mapa de Kohonen y Formación de Conjuntos de Entrenamiento, Validación y Testeo

I.1. Entrenamiento de un mapa de Kohonen.

I.2. Análisis de Datos. Aprovechando el mapa de Kohonen es posible analizar la información en busca de outliers y otras características que se considere relevante.

I.3. A nivel de cada categoría de Kohonen, se dividen los datos para formar conjuntos de entrenamiento, validación y testeo.

I.4. Se analiza el conjunto de entrenamiento, validación y testeo para ser entregado a la red de Retropropagación

Etapa II: Diseño y Entrenamiento, Validación y Testeo de la red de retropropagación (modelo)

II.1. Matrices de Datos de entrada de la Red

II.2. Normalización y depuración de los datos

II.3. Datos para el entrenamiento, validación y test de la Red Neuronal Artificial

La calidad del modelo depende en forma relevante de los siguientes factores:

- d) Función de Activación usada en las neuronas.
- e) Prueba con el número de neuronas de la capa de entrada y la capa oculta.
- f) Secuencia de presentación de pares de entrenamiento a la red.

Etapa III: Creación y Entrenamiento de la Red

III.1. Creación de los Datos

III.2. Entrenamiento de la Red Neuronal Artificial

Etapa IV: Simulación de la Red

IV.1. Simulación de la Red Neuronal Artificial

IV.2. Conversión de la respuesta de la simulación a valores originales

IV.3. Comparación del resultado entregado por la red y el valor esperado

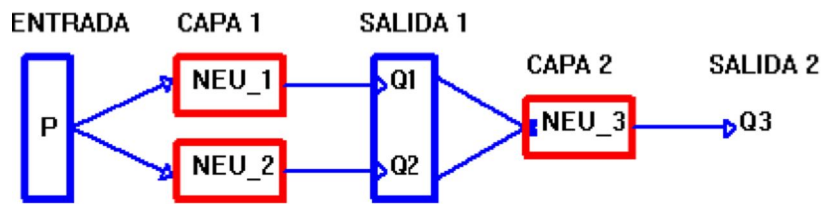
4.3. Aprendizaje de las Redes Neuronales Artificiales backpropagation y la regla de la Cadena en la derivación de funciones compuestas.

Dada la importancia de este concepto, se desarrollan estas ideas mediante un ejemplo, pero donde se muestra parte de los aspectos matemáticos que explican la implementación del método del gradiente y la regla de la cadena para encontrar un mínimo local del error medio cuadrático, descrito en el párrafo 3.3.3.4 de la sección 3.

4.3.1. Ejemplo Simple de Aprendizaje supervisado en RNA Backpropagation

Consideremos una Red Neuronal Artificial de dos capas, la primera con dos neuronas, donde cada neurona tiene solo un canal de entrada (una dentrita) y la segunda una neurona con dos canales de entrada (dos dentritas).

Figura N° 10.- Arquitectura de una Red Neuronal tradicional



Supuesto que para cada dato p_k , se busca una respuesta R_k de la red, que se aproxime a un valor T_k fijado. Este valor se llama respuesta o valor esperado de la red. En general si la respuesta esperada de la red es $T = (T_1, T_2, \dots, T_n) \in \mathcal{R}^n$, para un conjunto de datos que se le suministra $P = (p_1, p_2, \dots, p_n)$. Entonces la red cambiara sus pesos y ganancias para entregar una respuesta $R = (R_1, R_2, \dots, R_n)$, que se aproxima a $T = (T_1, T_2, \dots, T_n)$, de modo que minimice la media cuadrático

$$\frac{1}{2} \sum_{k=1}^n (T_k - R_k)^2 .$$

Como en este ejemplo cada neurona solo tiene una dentrita, entonces ingresaran las componentes de $P = (p_1, p_2, \dots, p_n)$, una a una.

4.3.2. Formulación matemática de las capas

a) Primera Capa

Matriz de pesos $W^1 = \begin{bmatrix} w_{1,1}^1 \\ w_{2,1}^1 \end{bmatrix}$ es una matriz de 2x1

Vector de ganancias $b^1 = \begin{bmatrix} b_1^1 \\ b_2^1 \end{bmatrix}$ es una matriz de 2x1

Dentro de la primera capa (antes de ser transformada) se tiene

$$V^1 = W^1 P + b^1 = \begin{bmatrix} w_{1,1}^1 p + b_1^1 \\ w_{2,1}^1 p + b_2^1 \end{bmatrix}$$

Funciones de transferencia de la primera capa

$$F^1(S) = \begin{bmatrix} f_1^1(s_1) \\ f_2^1(s_2) \end{bmatrix} = \begin{bmatrix} \tanh(s_1) \\ \tanh(s_2) \end{bmatrix}, \text{ donde}$$

$$f_1^1(s) = f_2^1(s) = \tanh(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}} \text{ (tansing)}$$

Salida de la primera capa

$$Q^1 = \begin{bmatrix} Q_1^1 \\ Q_2^1 \end{bmatrix} = F^1(V^1) = \begin{bmatrix} f_1^1(V_1^1) \\ f_2^1(V_2^1) \end{bmatrix} = \begin{bmatrix} f_1^1(w_{1,1}^1 p + b_1^1) \\ f_2^1(w_{2,1}^1 p + b_2^1) \end{bmatrix}, \text{ es una matriz de 2x1}$$

b) Segunda Capa

Matriz de pesos $W^2 = [w_{1,1}^2, w_{1,2}^2]$ es una matriz de 1x2

Vector de ganancias $b^2 = [b_1^2]$ es una matriz de 1x1

Dentro de la segunda capa (antes de ser transformada) se tiene

$$V^2 = W^2 Q^1 + b^2 = w_{1,1}^2 Q_1^1 + w_{1,2}^2 Q_2^1 + b_1^2$$

Funciones de transferencia de la segunda capa

$$F^2(S) = [f_1^2(s)] = f_1^2(s), \text{ matriz de 1x1, donde } f_1^2(s) = s \text{ (identidad=purelin)}$$

Salida de la segunda capa

$$Q^2 = F^2(V^2) = f_1^2(w_{1,1}^2 Q_1^1 + w_{1,2}^2 Q_2^1 + b_1^2) = w_{1,1}^2 Q_1^1 + w_{1,2}^2 Q_2^1 + b_1^2$$

4.3.3. Método del gradiente heurístico en el aprendizaje de la RNA

El vector $x = (w_{1,1}^1, w_{2,1}^1, b_1^1, b_2^1, w_{1,1}^2, w_{1,2}^2, b_1^2)$ es el vector de pesos y ganancias de la RNA del ejemplo que se viene implementando. El error en media cuadrática será la función

$F(x) = \frac{1}{2} \sum_{k=1}^n (T_k - R_k)^2$ de 7 variables. Basta minimizar cada sumando

$E(x) = \frac{1}{2} \sum_{k=1}^n (T_k - R_k)^2$, empleando el método del gradiente heurístico.

Se denotara con Q^2 el valor de R_k . Entonces se debe minimizar la función

$$E(x) = \frac{1}{2} (T_k - Q^2)^2.$$

El proceso de entrenamiento comienza dándole un valor inicial aleatorio al vector de pesos y ganancias $x = (w_{1,1}^1, w_{2,1}^1, b_1^1, b_2^1, w_{1,1}^2, w_{1,2}^2, b_1^2)$.

El algoritmo del gradiente heurístico indica que para minimizar la función $E(x)$ deben actualizarse los pesos y las ganancias de las dos capas con un factor de pequeño $0 < s < 1$.

Aplicando la regla de cadena, para la derivación de funciones compuestas, se comienza derivando las funciones de la segunda capa, de aquí viene el nombre de la red: Backpropagation.

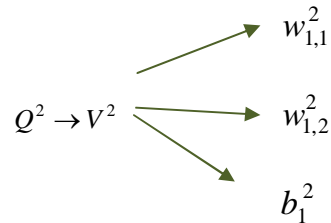
$$(a) \begin{cases} w_{1,1}^2 = w_{1,1}^2 - s \frac{\partial E}{\partial w_{1,1}^2}(x) \\ w_{1,2}^2 = w_{1,2}^2 - s \frac{\partial E}{\partial w_{1,2}^2}(x) \\ b_1^2 = b_1^2 - s \frac{\partial E}{\partial b_1^2}(x) \end{cases}$$

Luego se actualizan los pesos y ganancias de la primera capa

$$(b) \begin{cases} w_{1,1}^1 = w_{1,1}^1 - s \frac{\partial E}{\partial w_{1,1}^1}(x) \\ w_{2,1}^1 = w_{2,1}^1 - s \frac{\partial E}{\partial w_{2,1}^1}(x) \\ b_1^1 = b_1^1 - s \frac{\partial E}{\partial b_1^1}(x) \\ b_2^1 = b_2^1 - s \frac{\partial E}{\partial b_2^1}(x) \end{cases}$$

Derivadas parciales de $E(x)$

Para seguir desarrollando, se necesitan las derivadas parciales de $E(x) = \frac{1}{2}(T_k - Q^2)^2$ respecto de los pesos y ganancias. Para esto se aplica la regla de la cadena, en la derivación de funciones compuestas de varias variables. En el caso del ejemplo debe tomarse en cuenta las siguientes relaciones de dependencia, establecidas para la primera y segunda capa, por ejemplo:



Respecto de los pesos y ganancias de la segunda capa

a) La derivada parcial de $E(x)$ respecto del peso $w_{1,1}^2$

$$Q^2 = F^2(V^2) = f_1^2(V^2); \quad V^2 = w_{1,1}^2 Q_1^1 + w_{1,2}^2 Q_2^1 + b_1^2$$

$$-\frac{\partial E}{\partial w_{1,1}^2}(x) = (T_k - Q^2) \times \frac{dQ^2}{dV^2} \times \frac{\partial V^2}{\partial w_{1,1}^2}, \text{ Pero}$$

$$\frac{dQ^2}{dV^2} = (f_1^2)'(V^2) \text{ y } \frac{\partial V^2}{\partial w_{1,1}^2} = Q_1^1 \text{ entonces}$$

$$-\frac{\partial E}{\partial w_{1,1}^2}(x) = (T_k - Q^2) \times (f_1^2)' \times (V^2) \times Q_1^1$$

b) La derivada parcial de $E(x)$ respecto del peso $w_{1,2}^2$

$$Q^2 = F^2(V^2) = f_1^2(V^2); \quad V^2 = w_{1,1}^2 Q_1^1 + w_{1,2}^2 Q_2^1 + b_1^2$$

$$-\frac{\partial E}{\partial w_{1,2}^2}(x) = (T_k - Q^2) \times \frac{dQ^2}{dV^2} \times \frac{\partial V^2}{\partial w_{1,2}^2} \text{ Pero}$$

$$\frac{dQ^2}{dV^2} = (f_1^2)'(V^2) \text{ y } \frac{\partial V^2}{\partial w_{1,2}^2} = Q_2^1 \text{ entonces}$$

$$-\frac{\partial E}{\partial w_{1,2}^2}(x) = (T_k - Q^2) \times (f_1^2)' \times (V^2) \times Q_2^1$$

c) La derivada parcial de $E(x)$ respecto a la ganancia b_1^2

$$Q^2 = F^2(V^2) = f_1^2(V^2); \quad V^2 = w_{1,1}^2 Q_1^1 + w_{1,2}^2 Q_2^1 + b_1^2$$

$$-\frac{\partial E}{\partial w_{1,2}^2}(x) = (T_k - Q^2) \times \frac{dQ^2}{dV^2} \times \frac{\partial V^2}{\partial w_{1,2}^2} \quad \text{Pero}$$

$$\frac{dQ^2}{dV^2} = (f_1^2)'(V^2); \quad \frac{\partial V^2}{\partial b_1^2} = 1 \quad \text{Entonces}$$

$$-\frac{\partial E}{\partial b_1^2}(x) = (T_k - Q^2) \times (f_1^2)'(V^2)$$

Respecto de los pesos y ganancias de la primera capa

a) Derivada parcial de $E(x)$ respecto del peso $w_{1,1}^1$

$$Q^2 = F^2(V^2) = f_1^2(V^2);$$

$$V^2 = w_{1,1}^2 Q_1^1 + w_{1,2}^2 Q_2^1 + b_1^2; \quad Q_1^1 = f_1^1(V_1^1); \quad V_1^1 = w_{1,1}^1 p + b_1^1$$

$$-\frac{\partial E}{\partial w_{1,1}^1}(x) = (T_k - Q^2) \times \frac{\partial Q^2}{\partial V^2} \times \frac{\partial V^2}{\partial Q_1^1} \times \frac{\partial Q_1^1}{\partial V_1^1} \times \frac{\partial V_1^1}{\partial w_{1,1}^1} \quad \text{Pero}$$

$$\frac{dQ^2}{dV^2} = (f_1^2)'(V^2) \quad \text{y} \quad \frac{\partial V^2}{\partial Q_1^1} = w_{1,1}^2; \quad \frac{\partial Q_1^1}{\partial V_1^1} = (f_1^1)'(V_1^1);$$

$$\frac{\partial V_1^1}{\partial w_{1,1}^1} = p, \quad \text{entonces}$$

$$-\frac{\partial E}{\partial w_{1,1}^1}(x) = (T_k - Q^2) \times (f_1^2)'(V^2) \times w_{1,1}^2 \times (f_1^1)'(V_1^1) \times p$$

Denotando $\delta_1^1 = (T_k - Q^2) \times (f_1^2)'(V^2) \times w_{1,1}^2 \times (f_1^1)'(V_1^1)$, Sale

$$-\frac{\partial E}{\partial w_{1,1}^1}(x) = \delta_1^1 p$$

b) Derivada parcial de $E(x)$ respecto del peso $w_{2,1}^1$

$$Q^2 = F^2(V^2) = f_1^2(V^2);$$

$$V^2 = w_{1,1}^2 Q_1^1 + w_{1,2}^2 Q_2^1 + b_1^2; Q_2^1 = f_2^1(V_2^1); V_2^1 = w_{2,1}^1 p + b_2^1$$

$$-\frac{\partial E}{\partial w_{2,1}^1}(x) = (T_k - Q^2) \times \frac{\partial Q^2}{\partial V^2} \times \frac{\partial V^2}{\partial Q_2^1} \times \frac{\partial Q_2^1}{\partial V_2^1} \times \frac{\partial V_2^1}{\partial w_{2,1}^1} \quad \text{pero}$$

$$\frac{dQ^2}{dV^2} = (f_1^2)'(V^2) \quad \text{y} \quad \frac{\partial V^2}{\partial Q_2^1} = w_{1,2}^2; \quad \frac{\partial Q_2^1}{\partial V_2^1} = (f_2^1)'(V_2^1);$$

$$\frac{\partial V_2^1}{\partial w_{2,1}^1} = p, \quad \text{entonces}$$

$$-\frac{\partial E}{\partial w_{2,1}^1}(x) = (T_k - Q^2) \times (f_2^2)'(V^2) \times w_{1,2}^2 \times (f_2^1)'(V_2^1) \times p$$

$$\text{Llamando } \delta_2^1 = (T_k - Q^2) \times (f_2^2)'(V^2) \times w_{1,2}^2 \times (f_2^1)'(V_2^1),$$

$$-\frac{\partial E}{\partial w_{2,1}^1}(x) = \delta_2^1 p$$

c) Derivada parcial de $E(x)$ respecto a la ganancia b_1^1

$$Q^2 = f_1^2(V^2);$$

$$V^2 = w_{1,1}^2 Q_1^1 + w_{1,2}^2 Q_2^1 + b_1^2; Q_1^1 = f_1^1(V_1^1); V_1^1 = w_{1,1}^1 p + b_1^1$$

$$-\frac{\partial E}{\partial b_1^1}(x) = (T_k - Q^2) \times \frac{\partial Q^2}{\partial V^2} \times \frac{\partial V^2}{\partial Q_1^1} \times \frac{\partial Q_1^1}{\partial V_1^1} \times \frac{\partial V_1^1}{\partial b_1^1} \quad \text{pero}$$

$$\frac{dQ^2}{dV^2} = (f_1^2)'(V^2); \quad \frac{\partial V^2}{\partial Q_1^1} = w_{1,1}^2; \quad \frac{\partial Q_1^1}{\partial V_1^1} = (f_1^1)'(V_1^1);$$

$$\frac{\partial V_1^1}{\partial b_1^1} = 1, \quad \text{entonces}$$

$$-\frac{\partial E}{\partial b_1^1}(x) = (T_k - Q^2) \times (f_1^2)'(V^2) \times w_{1,1}^2 \times (f_1^1)'(V_1^1)$$

Como $\delta_1^1 = (T_k - Q^2) \times (f_1^2)'(V^2) \times w_{1,1}^2 \times (f_1^1)'(V_1^1)$,

$$-\frac{\partial E}{\partial b_1^1}(x) = \delta_1^1$$

d) Derivada parcial de $E(x)$ respecto a la ganancia b_2^1

$$Q^2 = f_1^2(V^2);$$

$$V^2 = w_{1,1}^2 Q_1^1 + w_{1,2}^2 Q_2^1 + b_1^2; Q_2^1 = f_2^1(V_2^1); V_2^1 = w_{2,1}^1 p + b_2^1$$

$$-\frac{\partial E}{\partial b_2^1}(x) = (T_k - Q^2) x \frac{\partial Q^2}{\partial V^2} \times \frac{\partial V^2}{\partial Q_2^1} \times \frac{\partial Q_2^1}{\partial V_2^1} \times \frac{\partial V_2^1}{\partial b_2^1} \text{ pero}$$

$$\frac{dQ^2}{dV^2} = (f_1^2)'(V^2); \frac{\partial V^2}{\partial Q_2^1} = w_{1,2}^2; \frac{\partial Q_2^1}{\partial V_2^1} = (f_2^1)'(V_2^1);$$

$$\frac{\partial V_2^1}{\partial b_2^1} = 1, \text{ entonces}$$

$$-\frac{\partial E}{\partial b_2^1}(x) = (T_k - Q^2) \times (f_1^2)'(V^2) \times w_{1,2}^2 \times (f_2^1)'(V_2^1)$$

Como $\delta_2^1 = (T_k - Q^2) \times (f_2^2)'(V^2) \times w_{1,2}^2 \times (f_2^1)'(V_2^1)$

$$-\frac{\partial E}{\partial b_2^1}(x) = \delta_2^1$$

Reemplazando las derivadas de las funciones de transferencia

a) Para la primera capa

$$f_1^1(s) = f_2^1(s) = \tanh(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}} \text{ entonces } (f_k^1)'(s) = 1 - [f_k^1(s)]^2$$

b) Para la segunda capa

$$f_1^2(s) = s \text{ entonces } (f_1^2)'(s) = 1$$

Se tienen las siguientes derivadas parciales:

Respecto de los pesos y ganancias de la segunda capa

$$\text{a) } -\frac{\partial E}{\partial w_{1,1}^2}(x) = (T_k - Q^2) \times Q_1^1$$

$$\text{b) } -\frac{\partial E}{\partial w_{1,2}^2}(x) = (T_k - Q^2) \times Q_2^1$$

$$\text{c) } -\frac{\partial E}{\partial b_1^2}(x) = (T_k - Q^2)$$

Respecto de los pesos y ganancias de la primera capa

Como $(f_1^1)'(V_1^1) = 1 - [f_1^1(V_1^1)]^2$ y $(f_2^1)'(V_2^1) = 1 - [f_2^1(V_2^1)]^2$, se tiene

$$\text{a) } -\frac{\partial E}{\partial w_{1,1}^2}(x) = p(T_k - Q^2) \times w_{1,1}^2 \times (1 - [f_1^1(V_1^1)]^2)$$

$$\text{b) } -\frac{\partial E}{\partial w_{2,1}^2}(x) = p(T_k - Q^2) \times w_{1,2}^2 \times (1 - [f_2^1(V_2^1)]^2)$$

$$\text{c) } -\frac{\partial E}{\partial b_1^1}(x) = (T_k - Q^2) \times w_{1,1}^2 \times (1 - [f_1^1(V_1^1)]^2)$$

$$\text{d) } -\frac{\partial E}{\partial b_2^1}(x) = (T_k - Q^2) \times w_{1,2}^2 \times (1 - [f_2^1(V_2^1)]^2)$$

Actualización de los pesos y ganancias en su forma vectorial

Segunda capa

$$\text{a) } w_{1,1}^2 = w_{1,1}^2 + s(T_k - Q^2) \times Q_1^1$$

$$\text{b) } w_{1,2}^2 = w_{1,2}^2 + s(T_k - Q^2) \times Q_2^1$$

$$\text{c) } b_1^2 = b_1^2 + s(T_k - Q^2)$$

CAPÍTULO V

IMPLEMENTACION DEL MODELO

5. Implementación del modelo

En esta sección se muestra la descripción general de la entrada de código de computadora junto con los requisitos y la estructura de las redes, así para el desarrollo del modelo en esta tesis se implementa el algoritmo en Matlab.

Las herramientas para la elaboración del modelo predictivo son diversas, en el presente trabajo la codificación está elaborada con funciones de redes neuronales las cuales el Matlab las tiene incorporadas. La implementación computacional de los algoritmos representa la culminación de los procesos de estudio y análisis realizados para la elaboración del modelo, y la ejecución de este modelo permite visualizar los resultados buscados.

Las funciones elaboradas son scripts desarrollados en Matlab cuyos archivos se guarda con extensión ".m". En el proceso de ejecución estos scripts devuelven valores, las variables cambian con su ejecución.

5.1. Sintaxis en MATLAB de la RNA Retropropagación o Backpropagation (**newff**)

Estas Redes Neuronales Artificiales son supervisadas. También se denominan Redes neuronales de propagación inversa. Su estructura o arquitectura es la misma que la de una red general, descrita en el capítulo III - sección 3.3. A continuación se describirá brevemente la sintaxis en MATLAB de esta clase de Redes Neuronales Artificiales y luego desarrollar, las etapas de ingreso de datos, depuración y transformación de la data, creación de la red neuronal Backpropagation, su entrenamiento y finalmente la simulación.

Para crear una red de propagación inversa se tiene la función:

net = newff(Rango,[C1 C2...Ck],{TF1 TF2...TFk},BTF,BLF,PF)

Dónde:

Rango= matrix de **nx2**, formado por los mínimo y máximo de cada fila de la matriz de datos de entrada, que tiene **n** filas. Si los datos de entrada fuera solo una fila, entonces

Rango será una matriz de **1x2**

Ci= Número de neuronas de la **capa- i**. Sucesión de capas [**C1 C2...Ck**] de la red.

TFi= Función de transferencia **capa- i** (por defecto = 'tansig')

BTF= Función de entrenamiento de la red Backpropagation (por defecto = 'traingdx')

BLF=Función de aprendizaje peso/sesgo de la red Backpropagation (por defecto = 'learnngdm')

PF= Función que mide la Performance (por defecto = 'mse'), y retorna una red neuronal Backpropagation de **k capas**.

La función de transferencia **TFi** puede ser cualquiera de las funciones de transferencia diferenciable tales como: **tansig**, **logsig**, o **purelin**.

La función de entrenamiento **BTF** puede ser cualquiera de las funciones de entrenamiento Backpropagation tales como: **trainlm**, **trainbfg**, **trainrp**, **traingd**, etc.

Para el entrenamiento se elige **trainlm** que según MATLAB, es un procedimiento basado en el método de Levenberg - Marquardt, que son algoritmos para resolver clases especiales de sistemas de ecuaciones, desarrolladas a partir de las publicaciones: [1] Marquardt D. y [2] Levenberg K.

5.2. Sintaxis en MATLAB de la RNA KOHONEN O SOM (con **newsom**)

Se puede crear una red SOM con la función: **newsom**. Esta función define los rangos de la matriz de pasos y el número de neuronas de la capa.

Sintaxis

net = newsom(PR,[D1,D2,...],TFCN,DFCN,STEPS,IN)

Donde:

PR= matrix de **Rx2**, formado por los mínimo y máximo de cada columna de la matriz de datos de entrada, que tiene **R** columnas.

Di=Dimensión de la capa **i**. La cual será [2 4]

TFCN=Función que describe la topología. Por defecto toma la topología “**hextop**”

DFCN=Función que describe la distancia. Por defecto toma la distancia “**linkdist**”

STEPS=Pasos para reducir el tamaño de la vecindad a 1. Por defecto=100

5.3. Implementación del Modelo Predictivo

El objetivo de la aplicación práctica es doble. Por un lado se desea chequear si modelo predictivo de generación es capaz de reproducir los viajes en un problema de generación de viajes y, por otro, chequear si la Metodología Propuesta para el Modelo Predictivo de generación de viajes usando un Red SOM y una Red de retropropagación tiene buenos resultados.

5.3.1. Implementación de Red SOM en Matlab

Etapas I: Mapa de Kohonen y Formación de Conjuntos de Entrenamiento, Validación y Testeo

I.1. Entrenamiento de un mapa de Kohonen.

Se define la matriz de entrada “**P**” la cual representara la matriz de datos de entrada para realizar el entrenamiento de un mapa de kohonen tal y como se muestra:

P=

80.00	305	70	63.57	103
40.00	127	67	32.18	64
97.00	289	99	52.55	120
65.00	189	56	51.08	89
27.00	57	16	44.73	21
52.00	146	47	54.75	58
41.00	155	89	28.34	78
124.00	380	126	65.15	145
87.00	247	86	47.5	125
22.00	75	25	46.57	38
3.00	27	14	38.33	15
152.00	396	113	55.34	156
23.00	111	41	30.04	45
118.00	332	156	39.68	135
3.00	5	2	49.5	4
7.00	24	11	26.46	11
118.00	291	80	66.77	118
90.00	148	48	36.57	86
54.00	143	54	36.28	60
68.00	204	76	39.77	94
82.00	231	80	43.44	110
111.00	296	97	40.68	125
126.00	253	91	63.5	94
67.00	182	92	41.13	87
61.00	262	81	51.11	113

Se puede crear una red SOM con la función: **newsom**. Esta función define los rangos de la matriz de pesos y el número de neuronas de la capa.

Como los datos de entrada a la red SOM son sucesiones de columnas, se toma la transpuesta de Q, que será una matriz de 5x25.

Se crea una red neuronal SOM de 2x4=8 neuronas, con Rango=minmax(Q); para sus pesos. Esta red se entrenará inicialmente con 100 iteraciones.

```
red = newsom(Rango, [2 4]);
red.trainParam.epochs = 100;
red = train(red,Q);
```

Una vez creada la red SOM mediante los scripts desarrollados en Matlab tal y como se muestra a continuación:

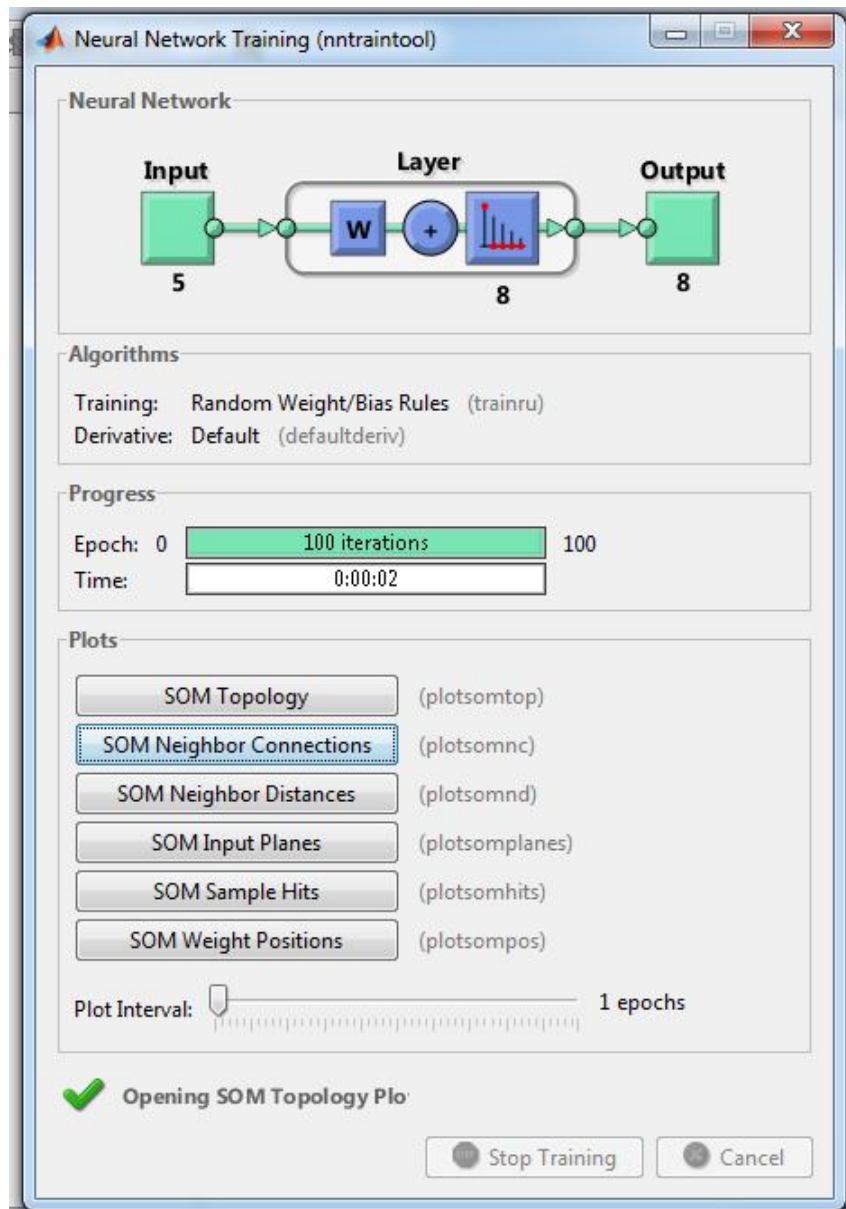
```

P=[ 80.00    305 70  63.57   103
    40.00    127 67  32.18    64
    97.00    289 99  52.55   120
    65.00    189 56  51.08    89
    27.00     57 16  44.73    21
    52.00    146 47  54.75    58
    41.00    155 89  28.34    78
    124.00   380 126 65.15   145
    87.00    247 86  47.5    125
    22.00     75 25  46.57    38
     3.00     27 14  38.33    15
    152.00   396 113 55.34   156
    23.00    111 41  30.04    45
    118.00   332 156 39.68   135
     3.00      5  2  49.5     4
     7.00     24 11  26.46    11
    118.00   291 80  66.77   118
    90.00    148 48  36.57    86
    54.00    143 54  36.28    60
    68.00    204 76  39.77    94
    82.00    231 80  43.44   110
    111.00   296 97  40.68   125
    126.00   253 91  63.5     94
    67.00    182 92  41.13    87
    61.00    262 81  51.11   113
]; Q=P';
Rango=minmax(Q);
red = newsom(Rango, [2 4]);
red.trainParam.epochs = 100;
red = train(red,Q);
W=red.iw{1,1};
distances=dist(Q',red.IW{1}');
[d,cndx]=min(distances,[],2);
cndx;
TABLA = [P cndx]

```

Luego de correr la Red Neuronal Artificial Kohonen (SOM) , sale una interface con 6 botones tal y como se muestra en el siguiente gráfico:

Figura N° 11.- Interface de una Red SOM



Activando cada botón que se puede visualizar sale:

Figura N° 12
SOM Topology

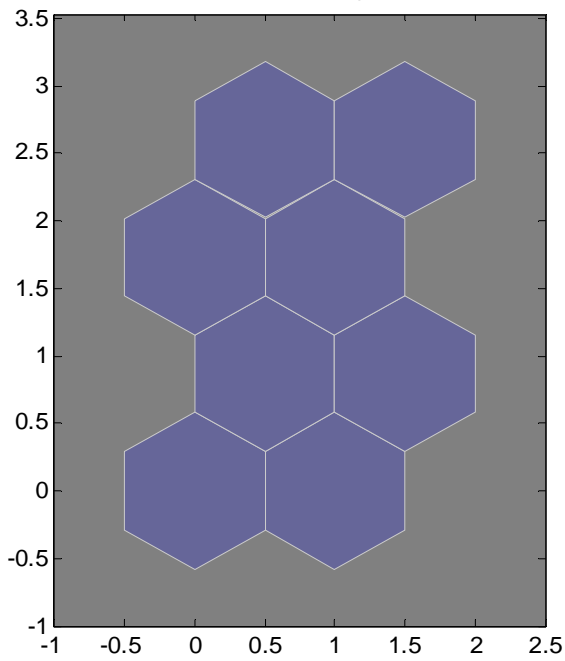


Figura N°13
SOM Neighbor Connections

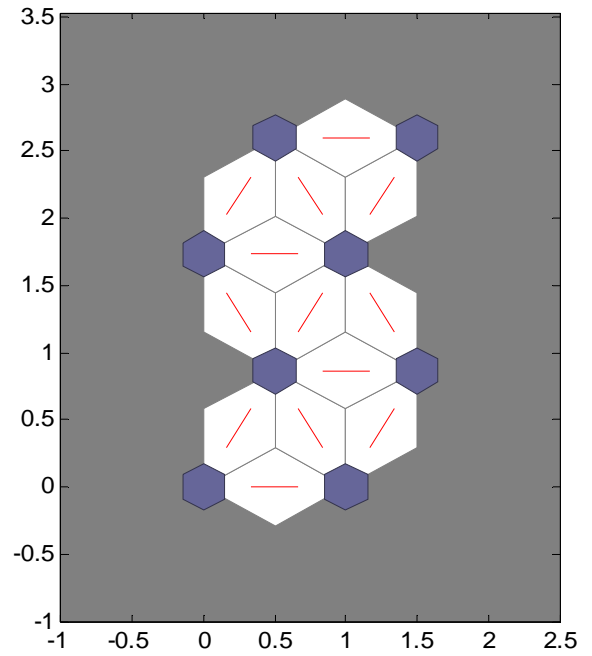


Figura N°14
SOM Neighbor Weight Distances

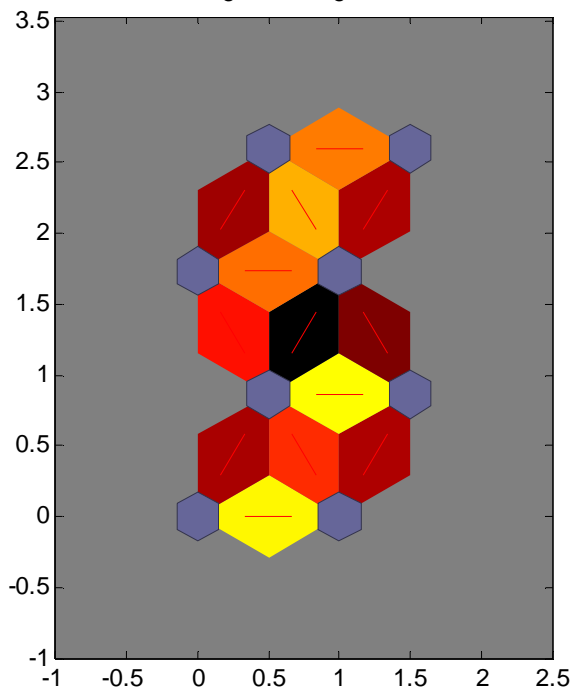


Figura N°15
Hits

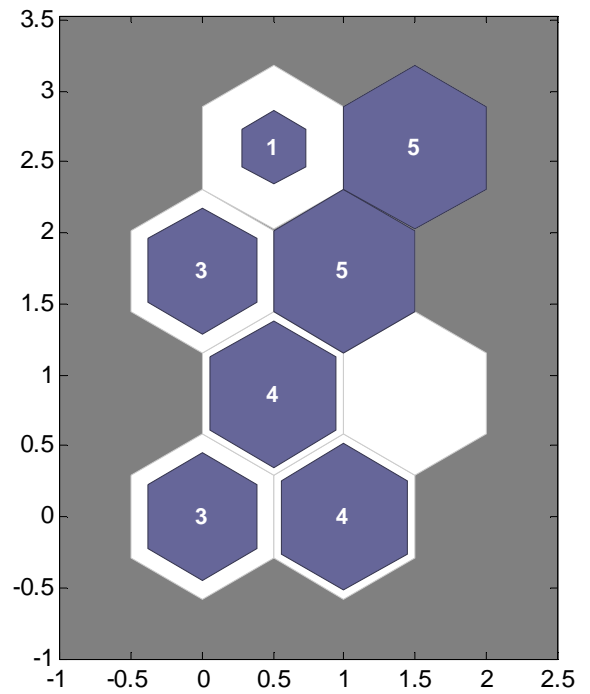


Figura N°16

Weights from Input 1

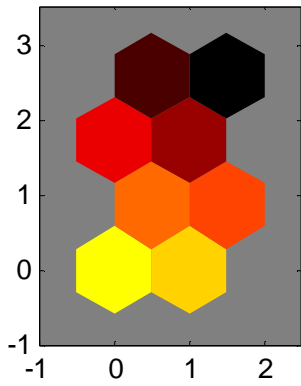


Figura N°17

Weights from Input 2

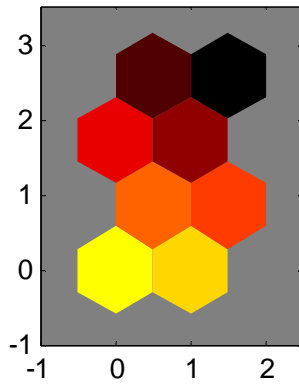


Figura N°18

Weights from Input 3

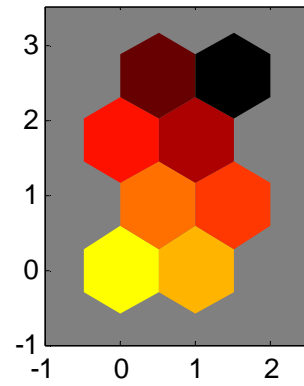


Figura N°19

Weights from Input 4

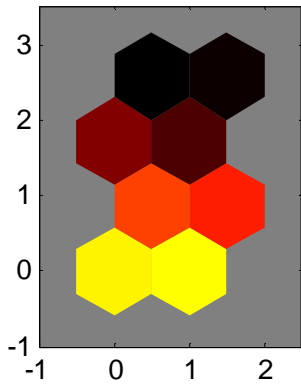


Figura N°20

Weights from Input 5

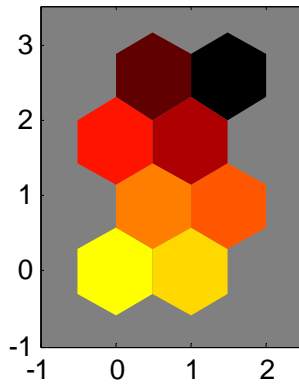
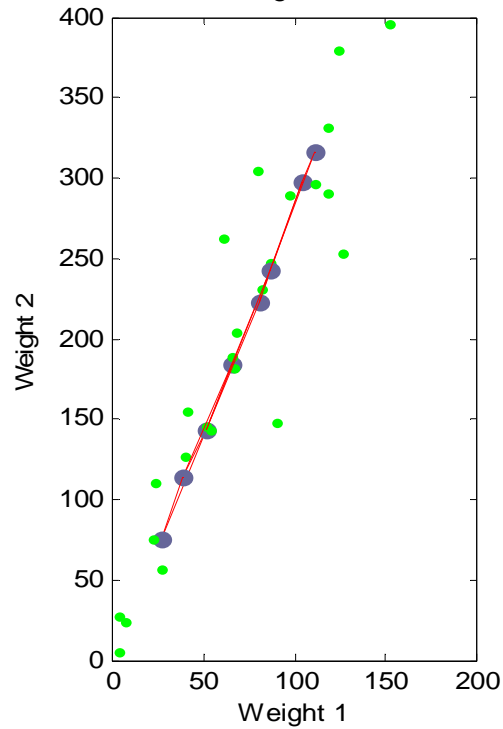


Figura N°21

SOM Weight Positions



Las gráficas que salen son:

SOM Topology: Muestra la topología hexagonal de la red (por defecto).

SOM Neighbord Connections: Representa las conexiones de las vecindades.

SOM Neighbord Distances: Los colores oscuros representan distancias largas, las claras distancias cortas.

Hits: Esta figura muestra las clases o clusters que se han formado y el número de miembros de P que agrupa cada uno de ellos.

SOM Weight Positions: Muestra la posición de los pesos del SOM (SOM Weight Positions) y de los puntos de P alrededor de ellas.

I.2. Análisis de Datos. Aprovechando el mapa de Kohonen es posible analizar la información en busca de outliers y otras características que se considere relevante.

De correr la red SOM se obtuvo una matriz de 25x5 que se pega a la tabla de datos en Excel con la etiqueta: Cluster

Tabla N° 06.- Resultados de la Red SOM.

Vehículos Totales	Ingreso medio de los hogares de la zona (en escala 1-7*)	Población de Estudiantes	Consumo Electrico	Total de hogares	CLUSTER
80	305	70	63.57	103	7
40	127	67	32.18	64	3
97	289	99	52.55	120	7
65	189	56	51.08	89	4
27	57	16	44.73	21	1
52	146	47	54.75	58	3
41	155	89	28.34	78	3
124	380	126	65.15	145	8
87	247	86	47.5	125	6
22	75	25	46.57	38	1
3	27	14	38.33	15	1
152	396	113	55.34	156	8
23	111	41	30.04	45	2
118	332	156	39.68	135	8
3	5	2	49.5	4	1
7	24	11	26.46	11	1
118	291	80	66.77	118	7
90	148	48	36.57	86	3
54	143	54	36.28	60	3
68	204	76	39.77	94	5
82	231	80	43.44	110	6
111	296	97	40.68	125	7
126	253	91	63.5	94	6
67	182	92	41.13	87	4
61	262	81	51.11	113	6

Seguidamente se coloca el registro de zonas de tránsito y los viajes generados tal y como se mostraron en la Tabla N°05:

Tabla N° 07.- Resultados de la Red SOM y sus respectivas Zonas de tránsito

Registro # (ZONA)	Vehículos Totales	Ingreso medio de los hogares de la zona (en escala 1-7*)	Población de Estudiantes	Consumo Eléctrico	Total de hogares	CLUSTER	Viajes Generados
112	80	305	70	63.57	103	7	880
113	40	127	67	32.18	64	3	196
114	97	289	99	52.55	120	7	753
115	65	189	56	51.08	89	4	360
116	27	57	16	44.73	21	1	395
117	52	146	47	54.75	58	3	433
118	41	155	89	28.34	78	3	308
119	124	380	126	65.15	145	8	967
120	87	247	86	47.5	125	6	636
121	22	75	25	46.57	38	1	1779
122	3	27	14	38.33	15	1	254
123	152	396	113	55.34	156	8	767
124	23	111	41	30.04	45	2	180
125	118	332	156	39.68	135	8	514
126	3	5	2	49.5	4	1	97
127	7	24	11	26.46	11	1	66
128	118	291	80	66.77	118	7	709
129	90	148	48	36.57	86	3	682
130	54	143	54	36.28	60	3	159
131	68	204	76	39.77	94	5	337
132	82	231	80	43.44	110	6	1684
133	111	296	97	40.68	125	7	353
134	126	253	91	63.5	94	6	1171
135	67	182	92	41.13	87	4	534
136	61	262	81	51.11	113	6	784

Luego se ordena de menor a mayor la columna del Cluster resulta la siguiente tabla:

Tabla N° 08.- Resultados ordenados de menor a mayor por numero cluster

Registro # (ZONA)	Vehículos Totales	Ingreso medio de los hogares de la zona (en escala 1-7*)	Población de Estudiantes	Consumo Electrico	Total de hogares	CLUSTER	Viajes Generados
119	124	380	126	65.15	145	1	967
123	152	396	113	55.34	156	1	767
125	118	332	156	39.68	135	1	514
112	80	305	70	63.57	103	2	880
114	97	289	99	52.55	120	2	753
128	118	291	80	66.77	118	2	709
133	111	296	97	40.68	125	2	353
120	87	247	86	47.5	125	3	636
132	82	231	80	43.44	110	3	1684
134	126	253	91	63.5	94	3	1171
136	61	262	81	51.11	113	3	784
115	65	189	56	51.08	89	5	360
131	68	204	76	39.77	94	5	337
135	67	182	92	41.13	87	5	534
113	40	127	67	32.18	64	6	196
117	52	146	47	54.75	58	6	433
118	41	155	89	28.34	78	6	308
129	90	148	48	36.57	86	6	682
130	54	143	54	36.28	60	6	159
124	23	111	41	30.04	45	7	180
116	27	57	16	44.73	21	8	395
121	22	75	25	46.57	38	8	1779
122	3	27	14	38.33	15	8	254
126	3	5	2	49.5	4	8	97
127	7	24	11	26.46	11	8	66

I.3. A nivel de cada categoría de Kohonen, se dividen los datos para formar conjuntos de entrenamiento, validación y testeo.

Se observa en la tabla N°08 que las variables independientes de la zona 121 no explican el resultado de la agrupación de la variable dependiente Generación de viajes, debido a que se refleja que para 38 hogares resultara 1779 viajes generados, por lo que no se tomara esta zona para ser entregada a la red de Retropropagación, quedando dividido los datos para los conjuntos de entrenamiento, validación y testeo, como se muestran en la siguiente tabla:

Tabla N° 09.- Conjuntos de entrenamiento, validación y testeo.

Registro # (ZONA)	Vehiculos Totales	Ingreso medio de los hogares de la zona (en escala 1- 7*)	Población de Estudiantes	Población "Otros"	Consumo Electrico	Total de hogares	CLUSTER	Viajes Generados
119	124	2.97	126	0	65.15	145	1	967
123	152	2.38	113	7	55.34	156	1	767
125	118	2.08	156	3	39.68	135	1	514
129	90	3.1	48	0	36.57	86	6	682
112	80	2.98	70	8	63.57	103	2	880
114	97	1.79	99	0	52.55	120	2	753
128	118	3.26	80	27	66.77	118	2	709
133	111	1.81	97	0	40.68	125	2	353
120	87	2.06	86	0	47.5	125	3	636
132	82	1.7	80	7	43.44	110	3	1684
134	126	3.19	91	10	63.5	94	3	1171
136	61	2.03	81	4	51.11	113	3	784
115	65	1.93	56	0	51.08	89	5	360
131	68	1.82	76	0	39.77	94	5	337
135	67	2.01	92	5	41.13	87	5	534
124	23	1.76	41	0	30.04	45	7	180
113	40	2.27	67	9	32.18	64	6	196
117	52	2.45	47	3	54.75	58	6	433
118	41	1.51	89	9	28.34	78	6	308
130	54	1.73	54	12	36.28	60	6	159
116	27	3.14	16	2	44.73	21	8	395
122	3	1.4	14	0	38.33	15	8	254
126	3	1.5	2	0	49.5	4	8	97
127	7	2.82	11	0	26.46	11	8	66

I.4. Se analiza el conjunto de entrenamiento, validación y testeo para ser entregado a la red de Retropropagación.

Se indica que se ha realizado múltiples pruebas en busca de la mejor formación de los conjuntos de entrenamiento, validación y testeo, posicionando de diferente manera los clusters hallados en por la Red kohonen y de esta manera obtener los mejores resultados con la Red de Retropropagación.

Tabla N° 10.- Conjuntos de entrenamiento, validación y testeo preparado para la Red de Retropropagación

Registro # (ZONA)	113	117	118	130	116	122	126	127	119	123	125	129	112	114	128	133	120	132	134	136	115	131	135	124
Vehiculos Totales	40	52	41	54	27	3	3	7	124	152	118	90	80	97	118	111	87	82	126	61	65	68	67	23
Ingreso medio de los hogares de la zona (en escala 1-7*)	2.27	2.45	1.51	1.73	3.14	1.4	1.5	2.82	2.97	2.38	2.08	3.1	2.98	1.79	3.26	1.81	2.06	1.7	3.19	2.03	1.93	1.82	2.01	1.76
Población de Estudiantes	67	47	89	54	16	14	2	11	126	113	156	48	70	99	80	97	86	80	91	81	56	76	92	41
Consumo Electrico	32.18	54.75	28.34	36.28	44.7	38.3	49.5	26.46	65.15	55.34	39.68	36.57	63.57	52.55	66.77	40.68	47.5	43.44	63.5	51.11	51.08	39.77	41.1	30.04
Total de hogares	64	58	78	60	21	15	4	11	145	156	135	86	103	120	118	125	125	110	94	113	89	94	87	45
CLUSTER	6	6	6	6	8	8	8	8	1	1	1	6	2	2	2	2	3	3	3	3	5	5	5	7
Viajes Generados	196	433	308	159	395	254	97	66	967	767	514	682	880	753	709	353	636	1684	1171	784	360	337	534	180

5.3.2. Implementación de Red de Retropropagación (Backpropagation) en Matlab

Se menciona que para la implementación de la red de Retropropagación se utilizara la Matriz de Datos ordenada tal y como se muestra en la tabla N°10.

A continuación se muestra el desarrollo del Modelo Neuronal de Demanda de Transporte en redes viales urbanas:

Previamente se corre el programa completo que trae MATLAB:

```
p=[40 52 41 54 27 3 3 7 124 152 118 90 80 97 118
111 87 82 126 61 65 68 67 23
2.27 2.45 1.51 1.73 3.14 1.4 1.5 2.82 2.97
2.38 2.08 3.1 2.98 1.79 3.26 1.81 2.06 1.7
3.19 2.03 1.93 1.82 2.01 1.76
67 47 89 54 16 14 2 11 126 113 156 48 70 99 80 97
86 80 91 81 56 76 92 41
32.18 54.75 28.34 36.28 44.73 38.33 49.5 26.46
65.15 55.34 39.68 36.57 63.57 52.55 66.77 40.68
47.5 43.44 63.5 51.11 51.08 39.77 41.13 30.04
64 58 78 60 21 15 4 11 145 156 135 86 103 120 118 125
125 110 94 113 89 94 87 45
];
t=[196 433 308 159 395 254 97 66 967 767 514 682 880 753 709
353 636 1684 1171 784 360 337 534 180
];
[pn,ps1] = mapstd(p);
[tn,ts] = mapstd(t);
[ptrans,ps2] = processpca(pn,0.001);
[T,Q] = size(ptrans);
Clas13 = [1:4:Q 2:4:Q];
Clas4 = 3:4:Q;
Clas2 = 4:4:Q;
Enp = ptrans(:,Clas13);
VEt = tn(:,Clas13);
val.P = ptrans(:,Clas4);
val.T = tn(:,Clas4);
test.P = ptrans(:,Clas2);
test.T = tn(:,Clas2);
Rango=minmax(ptrans);
net = newff(Rango,[5 5 1],{'tansig' 'tansig'
'purelin'},'trainlm');
[net,tr]=train(net,Enp,VEt,[],[],val,test);
plot(tr.epoch, tr.perf, tr.epoch, tr.vperf, tr.epoch, tr.tperf)
legend('ENTRENAMIENTO','VALIDACION','TEST',-1);
ylabel('ERROR CUADRÁTICO');
xlabel('ITERACIÓN')
title('CURVAS DE:')

an = sim(net,ptrans);size(an);
a = mapstd('reverse',an,ts);size(a)
for i=1:l
figure(i)
[m(i),b(i),r(i)]=postreg(a(i,:),t(i,:));
end
%=====Gráficos=====
aa=a';aaa=sort(aa);%Red de simulación
tt=t';ttt=sort(tt);%Target
```

```

nn=length(tt)
nnn=[1:nn]';
for i=1:1
    figure(4);
    subplot(3,1,i);
    plot(nnn,aa(:,i));hold on
    plot(nnn,tt(:,i),'r');
end
ss=[aa tt];
size(ss)

for i=1:1
    figure(5);
    subplot(3,1,i);
    plot(nnn,aaa(:,i));hold on
    plot(nnn,ttt(:,i),'r');
end
for i=1:1
    for j=1:nn-1
        for k=j+1:nn
            if (tt(j,i)>tt(k,i))
                c=tt(j,i);tt(j,i)=tt(k,i);tt(k,i)=c;
                c=aa(j,i);aa(j,i)=aa(k,i);aa(k,i)=c;
            end
        end
    end
end
for i=1:1
    figure(6);
    subplot(3,1,i);
    plot(nnn,aa(:,i));hold on
    plot(nnn,tt(:,i),'r');
end

```

El resultado es la salida gráfica de las curvas de entrenamiento, validación y prueba (test) seguido de graficas de interpretación de la respuesta de la red, entre ellas la regresión lineal, que compara los resultados que arroja la red con el valor esperado, que está en la matriz t.

5.4. Aplicación de la Red de Retropropagacion (Backpropagation)

Etapas II: Diseño y Entrenamiento, Validación y Testeo de la red de retropropagación (modelo)

II.1. Matrices de Datos de entrada de la Red

Las dos matrices p y t que son de orden 5x24 y 1x24 respectivamente, tal y como se muestran a continuación:

p=	40	52	41	54	27	3	3	7	124	152	118	90	80	97	118	111	87	82	126	61	65	68	67	23
	2.27	2.45	1.51	1.73	3.14	1.4	1.5	2.82	2.97	2.38	2.08	3.1	2.98	1.79	3.26	1.81	2.06	1.7	3.19	2.03	1.93	1.82	2.01	1.76
	67	47	89	54	16	14	2	11	126	113	156	48	70	99	80	97	86	80	91	81	56	76	92	41
	32.18	54.75	28.34	36.28	44.73	38.33	49.5	26.46	65.15	55.34	39.68	36.57	63.57	52.55	66.77	40.68	47.5	43.44	63.5	51.11	51.08	39.77	41.13	30.04
	64	58	78	60	21	15	4	11	145	156	135	86	103	120	118	125	125	110	94	113	89	94	87	45
	6	6	6	6	8	8	8	8	1	1	1	6	2	2	2	2	3	3	3	3	3	5	5	5

t=	196	433	348	348	395	254	258	50	967	767	771	779	880	753	934	675	636	1683	839	841	348	349	541	281
----	-----	-----	-----	-----	-----	-----	-----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	-----	-----	-----	-----	-----	-----

II.2. Normalización y depuración de los datos

a) Normalización

Para evitar el ingreso de datos muy grandes o muy pequeños a las RNA, se aplica la función `mapstd` a las matrices `p` y `t`.

Esta función transforma los elementos de cada fila de la matriz en otro, empleando la

función $H(x) = \frac{x - mc}{dc}$, donde `mc` y `dc` son respectivamente la media y la desviación

estándar de cada fila. El proceso biunívoco e independiente en cada fila y cada matriz.

De este modo resultan las matrices: `pn` y `tn` del mismo orden a las anteriores.

```
>> [pn,ps1] = mapstd(p);  
      [tn,ts] = mapstd(t);
```

b) Depuración de los datos mediante el Análisis de Componentes Principales

MATLAB tiene la función `processpca` que realiza el análisis de componentes principales de una matriz. Tiene como objetivo depurar de las 5 filas de la matriz de entrada `pn`, aquellas que se correlacionen entre sí, ordenar las componentes ortogonales resultantes, comenzando por aquellos que tienen mayor variación y eliminando las filas que menos contribuyen a la variación en el conjunto de datos, estando por debajo de una tolerancia (`0.001`). El resultado lo guarda en la matriz `ptrans`.

```
>>[ptrans,ps2] = processpca (pn,0.001);
```

Para ver cómo ha variado el número de filas hacemos:

```
>> size(ptrans)  
ans = 5    24  
>> ptrans(:,1)
```

Es decir la data se ha mantenido a una matriz de 4 filas y 24 columnas, concluyendo que la correlación que se inició en un inicio entre las variables ha sido correcta.

II.3. Datos para el entrenamiento, validación y test de la Red Neuronal Artificial

- a) Como la RNA tiene que entrenarse se debe separar la data en un grupo de columnas para el `entrenamiento`, en este caso se toma un 50%, es decir 12 columnas. Para la `validación` se toma un 25% y un 25% para el `test`.

Es mejor que esta selección sea aleatoria, pero MATLAB lo hace mediante subconjuntos de las clases de equivalencia módulo 4. Es decir selecciona las columnas de la matriz `ptrans` de acuerdo a estas clases de números:

```
>> Q=24;  
Clas13 = [1:4 :Q 2:4:Q]; % tiene 12 elementos  
Clas4   = 3:4 :Q;       % tiene 6 elementos  
Clas2   = 4:4 :Q;       % tiene 6 elementos
```

- b) Con estas clases se hace la elección de las columnas de **ptrans**, para el entrenamiento, la validación y el test, tomando las clases **Clas13**, **Clas4**, **Clas2** respectivamente.

```
>>Enp = ptrans(:,Clas13); %matriz de 5x12 : Entrenamiento
VEt = tn(:,Clas13); %matriz de 1x12
val.P = ptrans(:,Clas4); %matriz de 5x6 : Validación
val.T = tn(:,Clas4); %matriz de 1x6 :
test.P = ptrans(:,Clas2); %matriz de 5x6 : Test
test.T = tn(:,Clas2); %matriz de 1x6
```

La calidad del modelo depende en forma relevante de los siguientes factores:

- Función de Activación usada en las neuronas.
- Prueba con el número de neuronas de la capa de entrada y la capa oculta.
- Secuencia de presentación de pares de entrenamiento a la red.

Etapa III: Creación y Entrenamiento de la Red

III.1. Creación de los Datos

Para crear una red Backpropagation con MATLAB, como ya se indicó en 2.1, se llama a la función **newff**:

Como los datos de entrada están en la matriz **ptrans** de aquí sale el rango de la red con el código:

```
>>Rango=minmax(ptrans); %matriz de 5x2
```

El número de filas de **Rango** indica que cada neurona de la **primera capa** de la red tendrá 5 dendritas.

La primera capa de la red tiene 5 neuronas y 5 dendritas cada una.

La segunda capa (Capa escondida) de la red tiene 5 neuronas y 5 dendritas cada una (pues la capa anterior tiene 5 neuronas).

La tercera capa tiene 1 neurona y 5 dendritas (pues solo hay una sola variable de salida)

Las funciones de transferencia elegidas son **tansig**, **tansig** y **purelin**, respectivamente, para la primera, segunda capa.

Para el entrenamiento se elige **trainlm** que según MATLAB, es un procedimiento basado en el método de Levenberg-Marquardt, que son algoritmos para resolver clases especiales de sistemas de ecuaciones, desarrolladas en las publicaciones:

Marquardt, D., "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," SIAM J. Appl. Math. Vol. 11, pp 431-441, 1963.

Levenberg, K., "A Method for the Solution of Certain Problems in Least Squares," Quart. Appl. Math. Vol. 2, pp 164-168, 1944.

La red creada es entonces:

```
>>net = newff(Rango,[5 5 1],{'tansig' 'tansig' 'purelin'},'trainlm')
```

III.2. Entrenamiento de la Red Neuronal Artificial

Este proceso se realiza con los datos de la matriz **Enp** y con la parte correspondiente del valor esperado, que se encuentra en la matriz **VEt** que fueron seleccionados en el paso3. La validación y el test lo realizarán las otras matrices definidas en el paso indicado. La función de entrenamiento es **train** y su sintaxis es como sigue:

```
>>[net, tr]=train(net, Enp, VEt, [], [], val, test);
```

Nuevas matrices de peso y ganancias

Luego del entrenamiento aparecen nuevas matrices de peso y ganancias o sesgos, producto de los procesos de minimización de la media cuadrática que se realiza durante el entrenamiento de la red y que son:

Tabla N° 11.- Pesos y Sesgos (Ganancias) de la primera capa

Pesos					Ganancias
net.IW{1,1}					net.b{1,1}
-1.157	3.7227	3.7837	-0.3913	0.71	4.9594
4.1878	3.3685	2.6311	-0.5399	-1.9041	0.721
6.1226	-2.315	0.3137	1.1296	-0.3215	2.0744
0.1118	5.1751	3.2945	0.8666	3.7801	1.1286
-2.09	0.6389	1.2011	1.2151	1.669	3.0952

Luego de correr, se muestra una interface en la cual se aprecia que la salida indica que en 21 iteraciones se logra el entrenamiento y muestra las curvas de entrenamiento:

Figura N°22.- Curvas de Entrenamiento, Validación y Test de la Red

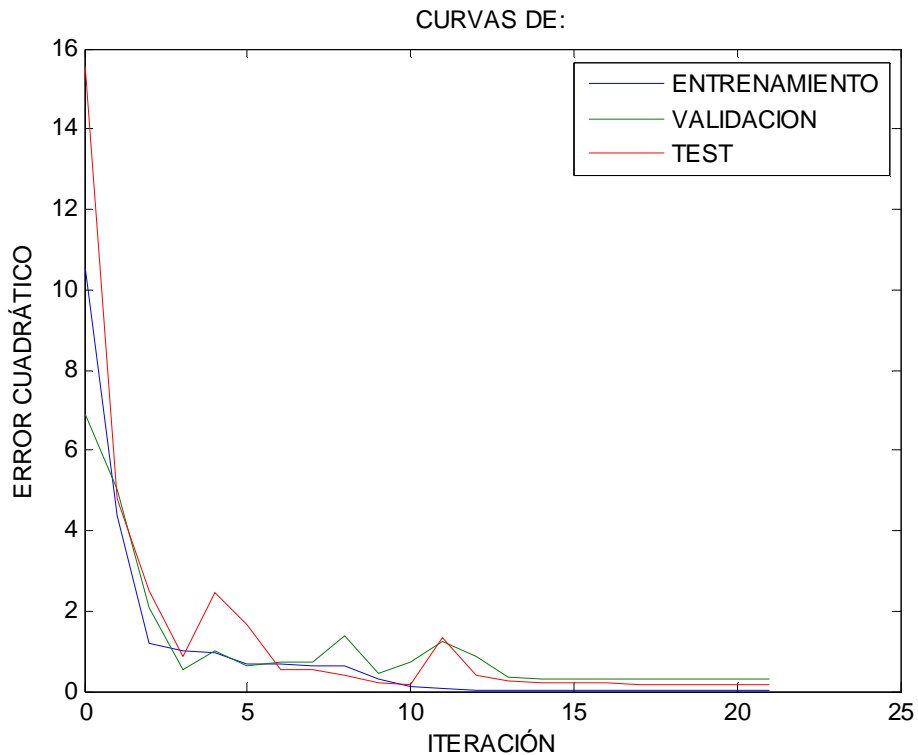
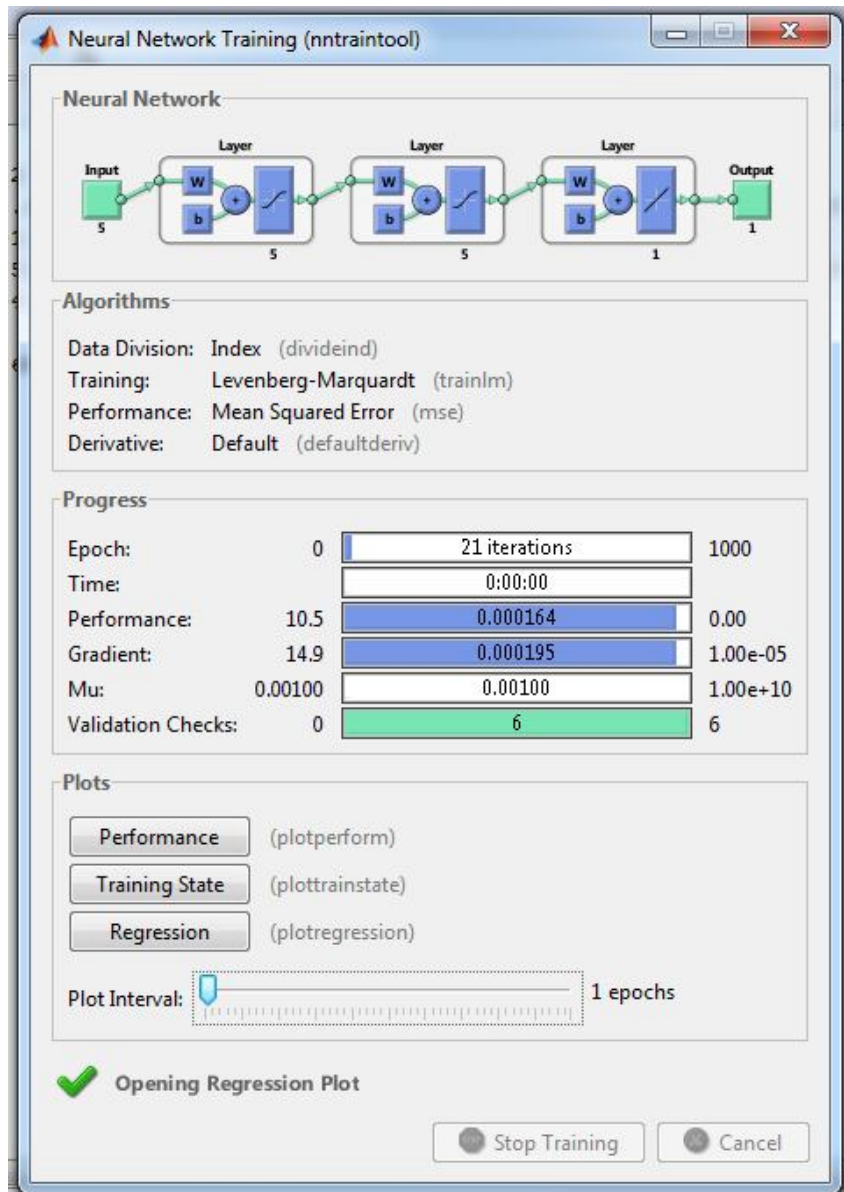


Figura N°23.- Interface de la Neural Network training



Ejecutando los botones:

a) Performance: Se ven las curvas del entrenamiento, de la validación y del test. Estas curvas muestran el comportamiento de la norma del gradiente.

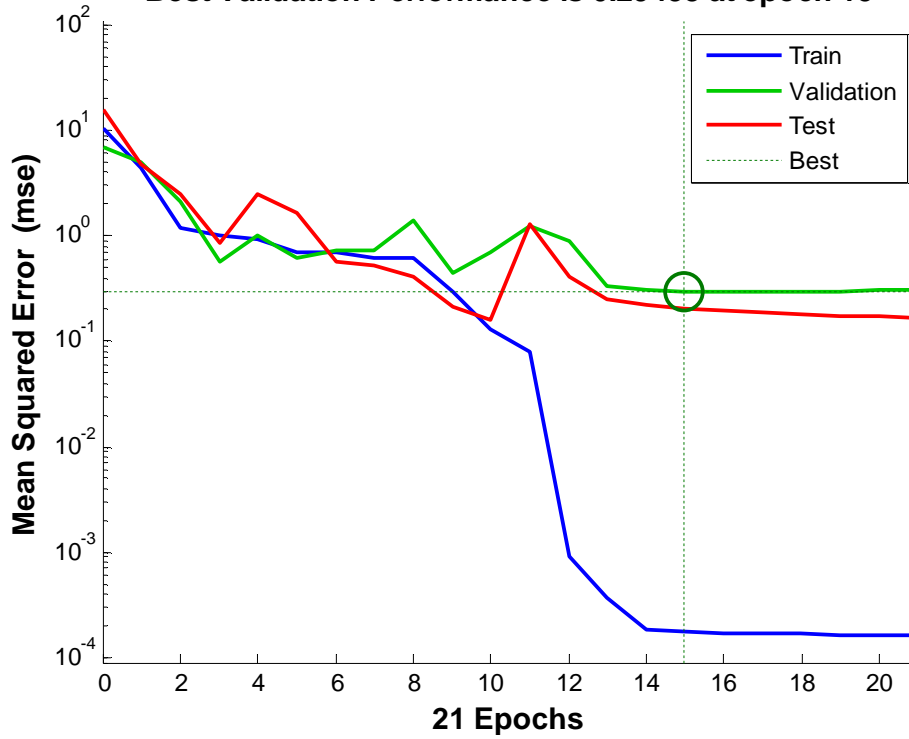
b) Training State: Muestra el comportamiento del gradiente en cada iteración, del mu y los errores del test.

c) Regresión: Presenta el R y la gráfica de la correlación entre el Output y el valor esperado de la red, en cada una de las tres etapas: **entrenamiento, validación, test y la correlación total.**

a) Performance

Figura N°24.- Curvas de Entrenamiento, Validación y Test respecto al comportamiento de la norma del gradiente

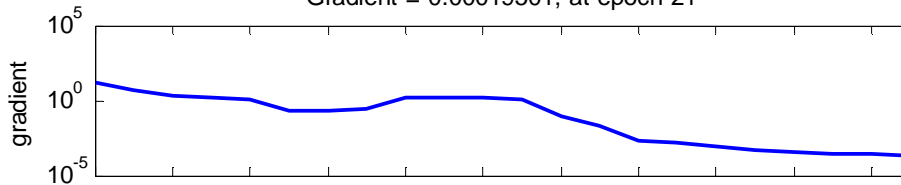
Best Validation Performance is 0.29438 at epoch 15



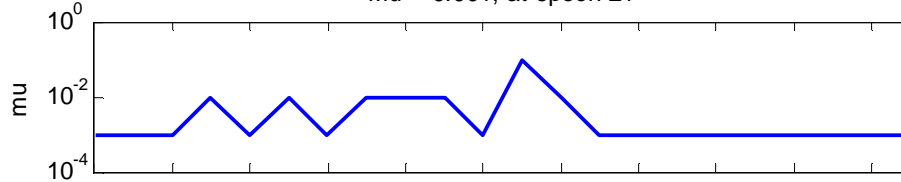
b) Training State

Figura N°25.- Gradientes

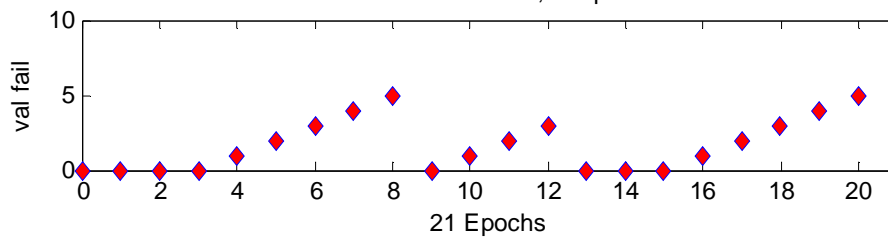
Gradient = 0.00019501, at epoch 21



Mu = 0.001, at epoch 21

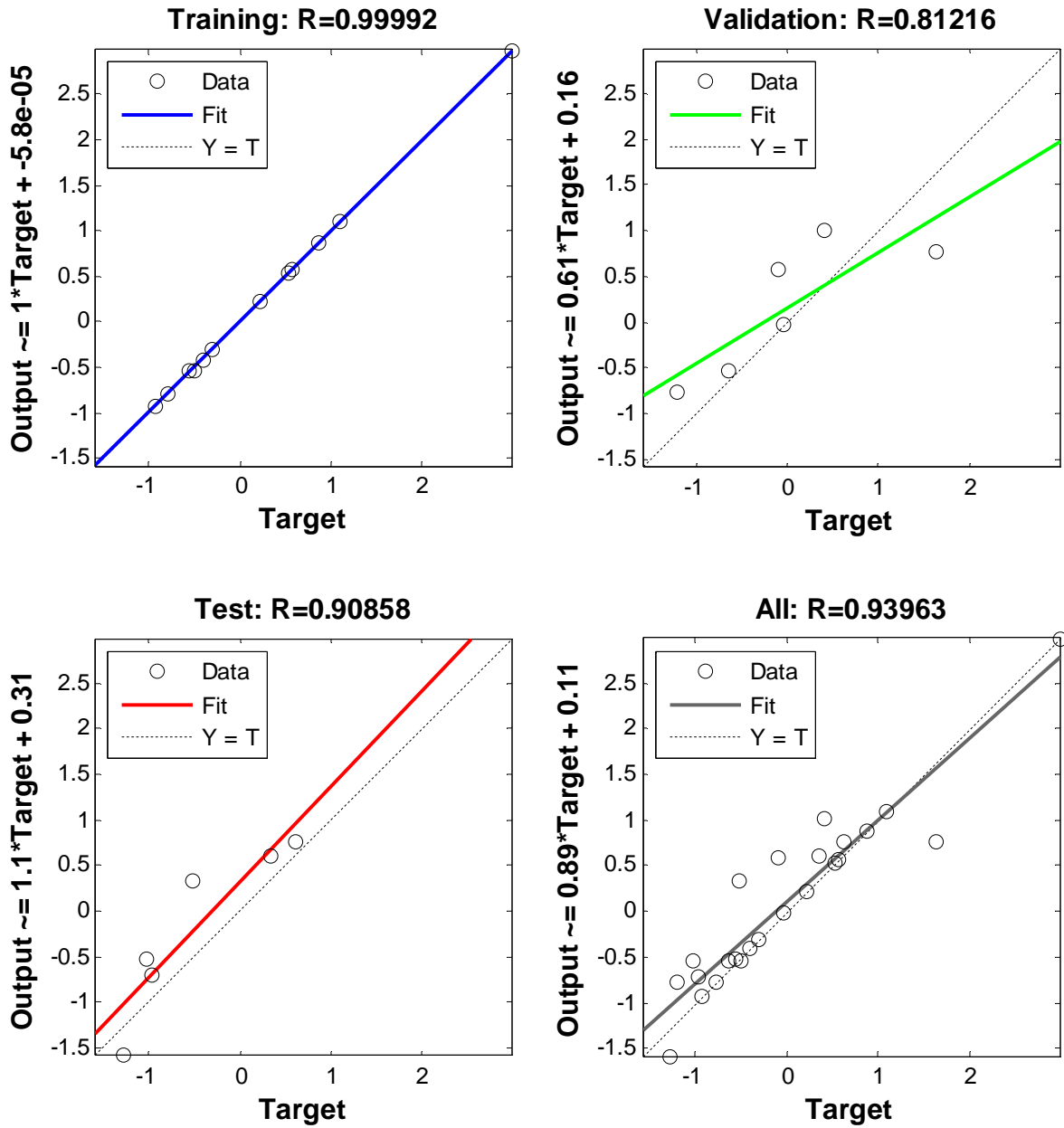


Validation Checks = 6, at epoch 21



b) Regresión

Figura N°26.- Gráficos de Regresión entre Output y el valor esperado de la red



Como se observa en las tres etapas entrenamiento, validación, test y la correlación total, existe una buen coeficiente de correlación, el cual es un indicador que habrá un buen resultado entre el Output y el valor esperado de la red

Etapa IV: Simulación de la Red

IV.1. Simulación de la Red Neuronal Artificial

Para ejecutar la simulación se llama a la función **sim** tal como aparece en el siguiente Commando

```
>>an = sim(net,ptrans);
```

La simulación consiste en ingresar por la red toda la data que contiene la matriz **ptrans** la red devuelve como respuesta la matriz **an**. Primero vemos de qué orden es **an** con:
ans = 1 24.

IV.2. Conversión de la respuesta de la simulación an a valores originales

En el paso II.2.- se normalizó la data y todo el proceso de entrenamiento y simulación de la red se ha realizado con valores normalizados. Para ver los resultados en el nivel y las características de la data original, se tiene que aplicar el proceso inverso de la normalización, llamando a la función:

```
>> a = mapstd('reverse',an,ts);  
>>size(a)  
ans = 1 24
```

IV.3. Comparación del resultado entregado por la red y el valor esperado

El siguiente bucle muestra la regresión lineal entre el valor entregado por la red: a y el valor esperado: t. Este proceso se realiza mediante la función postreg

```
for i=1:1  
figure(i)  
[m(i),b(i),r(i)]=postreg(a(i,:),t(i,:));  
end
```

La salida está formada por una gráfica, donde la recta de color rojo es la recta de regresión por mínimo cuadrados, con sus respectivos coeficientes de correlación R.

Figura N°27.- Recta de regresión por mínimo cuadrados

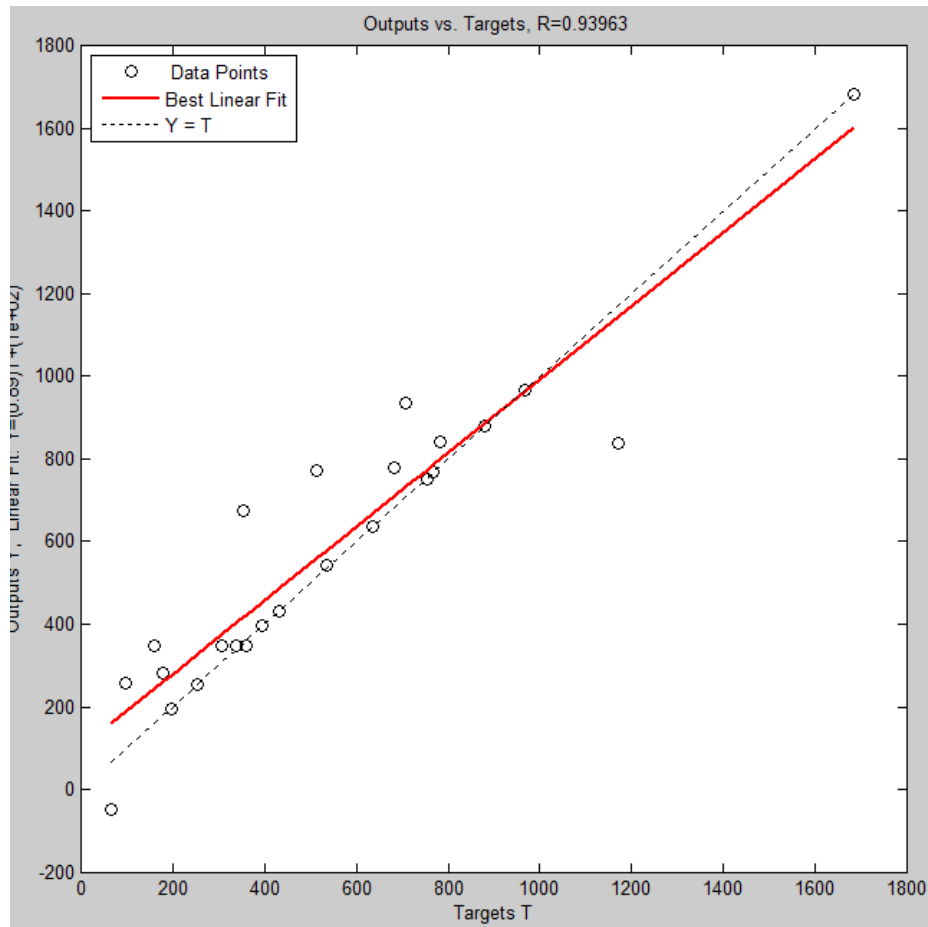


Tabla N° 12.- Comparación de Valor esperado (Viajes Generados) y Output (Viajes generados por la Red)

Registro # (ZONA)	Viajes Generados	Viajes generados por la red
113	196	196
117	433	433
118	308	348
130	159	348
116	395	395
122	254	254
126	97	258
127	66	50
119	967	967
123	767	767
125	514	771
129	682	779
112	880	880
114	753	753
128	709	934
133	353	675
120	636	636
132	1684	1683
134	1171	839
136	784	841
115	360	348
131	337	349
135	534	541
124	180	281

Se puede observar que en las zonas 113, 117, 122, 119, 123, 112, 114, 120,132 los resultados son exitosos ya que la red ha podido explicar con las variables independientes a la variable dependiente (Viajes generados).

Se indica si bien en en las zonas no mencionadas los datos no han logrado obtener resultados exactos si se observa los resultados de las zonas no mencionadas anteriormente los datos son datos cercanos a los esperados.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1.- Conclusiones

- Las redes de Kohonen permiten construir los conjuntos de entrenamiento, validación y testeo de una manera que provee de mayor control al modelador en cuanto a cobertura del espacio y sesgos que es posible evitar o introducir en el entrenamiento.
- Las redes de Kohonen pueden proveer información respecto de posible falta de variables explicativas y pueden servir **para encontrar información de dudosa calidad o outliers**.
- Las redes de Kohonen pueden ser útiles cuando se quiere crear varios modelos (especificaciones) para distintos rangos de datos.
- El filtro previo con Kohonen puede permitir utilizar menos datos en el entrenamiento.
- El filtro previo con Kohonen podría evitar el sobre-ajuste.
- Los mapas de Kohonen permiten definir rangos de validez para las redes de retropropagación.
- Se posee un algoritmo de entrenamiento para Kohonen.
- Se posee una forma más precisa de definir pertenencia a una categoría de Kohonen.
- Cambiar las funciones de activación de una red de retropropagación puede mejorar la estimación. Usar un tipo de función en la capa de entrada, escondida y otro en la de salida es una alternativa con buenos resultados.

6.2.- Recomendaciones

- El presente trabajo de investigación puede ampliarse a un modelamiento más integro de la modelación de la Demanda de transporte, utilizando la misma metodología propuesta en la presente tesis.
- El Ministerio de Transporte y Comunicaciones se le recomienda crear un Sistema de Información de Transporte Accesible por Internet, que permita acceder por vía Internet a los datos y resultados más relevantes de las Encuestas de Movilidad y los Censos de Flujo de Tráfico.
- Para implementar dicho sistema se crearía alguna entidad la cual se encargue de las encuestas de movilidad y censos de flujos de tráfico, Tal y como lo muestra la Secretaria de planificación de Transporte (SECTRA) en las principales ciudades de Chile durante los últimos años.

BIBLIOGRAFÍA

- [1] Curso: Programación Matemática. Pedro C. Espinoza H. Universidad Nacional de Ingeniería Facultad de Ingeniería Industria y de Sistemas Sección de Post Grado. Doctorado en Ingeniería Industrial.
- [2] Marquardt, D., "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," SIAM J. Appl. Math. Vol. 11, pp 431-441, 1963.
- [3] Levenberg, K., "A Method for the Solution of Certain Problems in Least Squares," Quart. Appl. Math. Vol. 2, pp 164-168, 1944.
- [4] Modelación de la Demanda de Transporte usando Redes Neuronales.- Rodrigo Quijada Plubins.- Universidad de Chile.-Facultad de Ciencias Físicas y Matemáticas Departamento de Ingeniería Civil Departamento de Ingeniería Industrial.
- [5] NAKATSUJI, T. y T. KAKU (1989) Application of neural network models to traffic engineering problems. Proceeding of Infrastructure Planning 12, pp.297-304.
- [6] RAIJA JR., A.A. (2000) Acessibilidade e mobilidade na estimativa de um índice de potencial de viagens utilizando redes neurais artificiais. São Carlos. 202 p. Tese (Doutorado) – Escola de Engenharia de São Carlos, Universidad de São Paulo.
- [7] Modeling Arterial Travel Time with Limited Traffic Variables using Conditional Independence Graphs & State-Space Neural Networks- Ghassan Abu-Lebdeh, Ajay K. Sing- Department of Civil Engineering, American University of Sharjah, Sharjah, UAE Jacobs Engineering Group, Huston, TX, USA
- [8] Modeling Arterial Travel Time with Limited Traffic Variables using Conditional Independence Graphs & State-Space Neural Networks- Ghassan Abu-Lebdeh, Ajay K. Sing- Department of Civil Engineering, American University of Sharjah, Sharjah, UAE Jacobs Engineering Group, Huston, TX, USA
- [9] G. Box, G. Jenkins, Time series analysis: forecasting and control, San Francisco: Holden Day, 1976.
- [10] M. Clements, P. Franses, N. Swanson, Forecasting economic and financial time-series with non-linear models, International Journal of Forecasting 20 (2) (2004) 169–183.
- [11] J. De Gooijer, R. Hyndman, 25 years of time series forecasting, International Journal of Forecasting 22 (3) (2006) 443–473.
- [12] B. Samanta, Prediction of chaotic time series using computational intelligence, Expert Systems with Applications 38 (9) (2011) 11406–11411.