Universidad Nacional Mayor de San Marcos
Universidad del Perú. Decana de América

Dirección General de Estudios de Posgrado
Facultad de Ciencias Matemáticas
Unidad de Posgrado

Zonificación dinámica para el recojo de basura usando redes de Kohonen: caso distrito de Bellavista-Callao

TESIS

Para optar el Grado Académico de Magíster en Investigación de Operaciones y Sistemas con mención en Optimización de Sistemas de Mercadeo y Producción

AUTOR

Alicia Cirila RIOJAS CAÑARI

ASESOR

Dra. María Del Pilar ÁLVAREZ RIVAS

Lima, Perú

2023



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

https://creativecommons.org/licenses/by-nc-sa/4.0/

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Riojas, A. (2023). *Zonificación dinámica para el recojo de basura usando redes de Kohonen: caso distrito de Bellavista-Callao*. [Tesis de maestría, Universidad Nacional Mayor de San Marcos, Facultad de Ciencias Matemáticas, Unidad de Posgrado]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Alicia Cirila Riojas Cañari
Tipo de documento de identidad	DNI
Número de documento de identidad	06018994
URL de ORCID	https://orcid.org/0000-0002-1110-7255
Datos de asesor	
Nombres y apellidos	María Del Pilar Álvarez Rivas
Tipo de documento de identidad	DNI
Número de documento de identidad	06445832
URL de ORCID	https://orcid.org/0000-0002-9580-0347
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Ricardo López Guevara
Tipo de documento	DNI
Número de documento de identidad	07593458
Miembro del jurado 1	
Nombres y apellidos	Sonia Esther Castro Ynfantes
Tipo de documento	DNI
Número de documento de identidad	06736632
Miembro del jurado 2	
Nombres y apellidos	Luis Gonzalo Acosta Espejo
Tipo de documento	DNI
Número de documento de identidad	09437003
Datos de investigación	

Línea de investigación	A.3.3.1. Optimización matemática
Grupo de investigación	Investigación interdisciplinaria en epistemología, ciencias y humanidades - EPISTEME
Agencia de financiamiento	Sin financiamiento.
Ubicación geográfica de la investigación	Universidad Nacional Mayor de San Marcos Facultad de Ciencias matemáticas. Latitud: -12.060353528963441 Longitud: -77.08203435888929
Año o rango de años en que se realizó la investigación	Marzo 2014 – enero 2020
URL de disciplinas OCDE	Ciencias de la computación https://purl.org/pe-repo/ocde/ford#1.02.01 Matemáticas aplicadas https://purl.org/pe-repo/ocde/ford#1.01.02

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
Universidad del Perú, DECANA DE AMÉRICA

FACULTAD DE CIENCIAS MATEMÁTICAS
VICEDECANATO DE INVESTIGACIÓN Y POSGRADO
UNIDAD DE POSGRADO

ACTA DE SUSTENTACIÓN DE TESIS PARA OPTAR EL GRADO ACADÉMICO DE MAGÍSTER

Siendo las 15:55 horas del día miércoles trece de setiembre de dos mil veintitrés, en el Laboratorio UPG de la Facultad de Ciencias Matemáticas, se reunió el Jurado Evaluador de tesis, presidido por el Dr. Ricardo López Guevara e integrado por los siguientes miembros: Mg. Sonia Esther Castro Ynfantes (Jurado Informante) y la Dra. María Del Pilar Álvarez Rivas como Asesora; y con la participación del Dr. Luis Gonzalo Acosta Espejo (Jurado Externo) en la sala virtual (meet.google.com/jma-diam-kej), para la sustentación de la tesis titulada: **«ZONIFICACIÓN DINÁMICA PARA EL RECOJO DE BASURA USANDO REDES DE KOHONEN: CASO DISTRITO DE BELLAVISTA-CALLAO»** presentada por la Señora Bachiller **ALICIA CIRILA RIOJAS CAÑARI**, para optar el **Grado Académico de Magíster en Investigación de Operaciones y Sistemas con Mención en Optimización de Sistemas de Mercadeo y Producción**.

Concluida la exposición, los miembros del Jurado de Tesis procedieron a formular sus preguntas que fueron absueltas por la graduanda; acto seguido se procedió a la evaluación correspondiente, según tabla adjunta, habiendo obtenido a la Bachiller **ALICIA CIRILA RIOJAS CAÑARI**, el calificativo de 19 (EXCELENTE).

Habiendo sido aprobada la sustentación de la Tesis, el Jurado Evaluador recomienda que el Consejo de Facultad apruebe el otorgamiento del **Grado Académico de Magíster en Investigación de Operaciones y Sistemas con Mención en Optimización de Sistemas de Mercadeo y Producción** a la Bachiller **ALICIA CIRILA RIOJAS CAÑARI.**

Siendo las 15:58 horas, se levantó la sesión, firmando para constancia la presente Acta:

Dr. Ricardo López Guevara
PRESIDENTE

Mg. Sonia Esther Castro Ynfantes
MIEMBRO

Dr. Luis Gonzalo Acosta Espejo
MIEMBRO EXTERNO

Dra. María Del Pilar Álvarez Rivas
ASESORA

CERTIFICAD0 DE SIMILITUD

Yo María del Pilar Álvarez Rivas. en mi condición de asesor acreditado con la Resolución Decanal N°001372-2023-D-FCM/UNMSM de la tesis, cuyo título es "ZONIFICACIÓN DINÁMICA PARA EL RECOJO DE BASURA USANDO REDES DE KOHONEN: CASO DISTRITO DE BELLAVISTA-CALLAO", presentado por la bachiller Alicia Cirila Riojas Cañari para optar el Grado Académico de Magíster en Investigación de Operaciones y Sistemas con mención en Optimización de Sistemas de Mercadeo y Producción INFORMO que se ha cumplido con lo establecido en la Directiva de Originalidad y de Similitud de Trabajos Académicos, de Investigación y Producción Intelectual al someter la mencionada tesis a la revisión, análisis y evaluación mediante el software de similitud textual TURNITIN, determinando que el documento evaluado cuenta con el porcentaje de 3.% de similitud, nivel **PERMITIDO** para continuar con los trámites correspondientes y para su **publicación en el repositorio institucional.** Emito la presente certificación, en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para la obtención del grado correspondiente.

MARÍA DEL PILAR ALVAREZ RIVAS
DNI: 06445832



Huella

digital

INDICE GENERAL

INTRODUCCIÓN _____ *1*

CAPITULO 1: PLANTEAMIENTO DEL PROBLEMA _____ *3*

1.1. Situación Problemática _____ **3**

1.2. Descripción del sistema _____ **3**

1.3. Formulación del Problema _____ **7**

1.4. Justificación de la investigación _____ **8**

1.5. Objetivos _____ **9**

1.6. Hipótesis de investigación _____ **9**

CAPITULO 2: MARCO DE REFERENCIA _____ *10*

2.1. Antecedentes del Problema _____ **10**

2.2. Marco epistemológico de la investigación _____ **12**

2.3. Marco histórico _____ **13**

2.4. Marco teórico _____ **14**

2.4.1. Residuos Sólidos _____ 15

2.4.2. Diagrama de Voronoi _____ 17

2.4.3. Redes neuronales _____ 24

2.4.4. Redes de Kohonen _____ 39

2.4.5. Mapas autoorganizados en Matlab _____ 54

CAPITULO 3: MÉTODO _____ *67*

3.1 Tipo y Diseño de Investigación _____ **67**

3.2 Unidad de análisis _____ **67**

3.3 Población de estudio _____ **67**

3.4 Tamaño de muestra _____ **67**

3.5. Técnicas de recolección de datos _____ **67**

3.6 Análisis e interpretación de la información _____ **71**

3.7 Pasos realizados para la clústerización _____ **71**

CAPITULO 4: RESULTADOS Y DISCUSIÓN _____ *74*

4.1. Análisis, interpretación y discusión de resultados _____ **74**

4.1.1 Escenario 1: Cinco camiones en una red 1x5 _____ 74

4.1.2 Escenario 2: Seis camiones _____ 81

4.1.3 Escenario 3: Nueve camiones _____ 85

4.2. Prueba de la hipótesis de investigación 86

4.3. Presentación de resultados 86

CAPITULO 5: IMPACTOS *88*

5.1. Propuesta para la solución del problema 88

5.2. Costos de implementación de la propuesta 88

5.3. Beneficios que aporta la propuesta 88

CONCLUSIONES *90*

RECOMENDACIONES PARA FUTURAS INVESTIGACIONES *91*

BIBLIOGRAFÍA *92*

ANEXOS *95*

ANEXO N°1: Distrito de Bellavista por manzanas 96

ANEXO N°2 Ejemplo de red supervisada 98

ANEXO N°3 Ejemplo de una red de Hopfield 99

ANEXO N°4.1: Script de clusterizacion mediante Kohonen REDSOM 102

ANEXO N°4.2: Script para evaluar la estructura de la red de Kohonen EVALUARED 104

ANEXO N°5: Distancias máximas para diversos epoch. Caso 3x2 106

ANEXO N°6 Coordenadas UTM del clúster 1 Con una arquitectura (1X6) 107

ANEXO N°7 Matriz de consistência 108

Lista de Tablas

Tabla 1 Cantidad de residuos en el área metropolitana de Barcelona *11*
Tabla 2 Cantidad de residuos en el distrito federal *12*
Tabla 3 Puntajes obtenidos por los 30 postulantes del año anterior *34*
Tabla 4 Puntajes obtenidos por los 5 postulantes de prueba *34*
Tabla 5 Pesos finales *37*
Tabla 6 Evaluación de los 5 postulantes de prueba *38*
Tabla 7 Coordenadas de las neuronas (postas médicas) mi en epoch 1 ..... *46*
Tabla 8 Distancia de los 20 pueblos a cada mi en la epoch 1 ..... *47*
Tabla 9 Coordenadas de los pueblos asignados a las postas médicas en epoch 1 ..... *47*
Tabla 10 Distancia de cada neurona (pueblo) a su neurona ganadora en epoch 1 ..... *47*
Tabla 11 Coordenadas de las neuronas (postas médicas) mj luego de la actualización en el epoch 1 ..... *49*
Tabla 12 Distancia de cada neurona (pueblo) a su neurona ganadora en epoch 1 después de la actualización *50*
Tabla 13 Diferencias de distancia de los pueblos a la posta luego de epoch1 ..... *50*
Tabla 14 Coordenadas de las neuronas (postas médicas) mj luego de la actualización en el epoch 1000 ..... *50*
Tabla 15 Distancia de cada neurona (pueblo) a su neurona ganadora en epoch 1000 ..... *51*
Tabla 16 Diferencias de distancia de los pueblos a la posta luego de epoch1000 ..... *51*
Tabla 17 Pesos de las neuronas ganadoras (UTM) después de 200 epochs *59*
Tabla 18 Latitud y longitud de la región 1 ..... *69*
Tabla 19 Intersecciones (nodos) de la región 1 con sus nodos adyacentes *70*
Tabla 20 Distancia entre los nodos adyacentes de la región 1 ..... *70*
Tabla 21 Distancia entre los nodos adyacentes de la región 1 ..... *71*
Tabla 22 Esquinas de cada clúster escenario 1 ..... *75*
Tabla 23 Coordenadas de las neuronas ganadoras en UTM escenario 1 ..... *76*
Tabla 24 Distancias promedio en metros dentro de cada clúster escenario 1 ..... *76*
Tabla 25 Distancias máximas en metros dentro de cada clúster escenario 1 ..... *77*
Tabla 26 Distancia de todas las esquinas a cada neurona ganadora. escenario 1 ..... *77*
Tabla 27 Estadísticos descriptivos de las máximas distancias (dimensión 1x5) *79*
Tabla 28 Asimetría y curtosis de las distancias máximas *79*
Tabla 29 Pruebas de normalidad *81*
Tabla 30 Número de epochs sugeridas para minimizar la distancia máxima (caso seis camiones) *82*
Tabla 31 Esquinas de cada clúster en el escenario 2 ..... *82*
Tabla 32 Coordenadas de las neuronas ganadoras escenario 2 ..... *83*
Tabla 33 Distancias promedio en metros dentro de cada clúster escenario 2 ..... *84*
Tabla 34 Distancias máximas en metros dentro de cada clúster escenario 2 ..... *84*
Tabla 35 Esquinas del clúster 1 ..... *84*
Tabla 36 Nodos adyacentes y distancias *85*
Tabla 37 Número de epochs sugeridas para minimizar la distancia máxima C *86*

Lista de Figuras

Figura 1 Mapa de Bellavista-Callao 4
Figura 2 Conjunto S de puntos (lugares) y casco convexo del conjunto S 18
Figura 3Triangulación del conjunto S 18
Figura 4 Recta bisectriz de (a,b) 19
Figura 5 Semi plano H(a,b) Semi plano H(b,a) 20
Figura 6 Diagrama de Voronoi para 12 puntos pi 22
Figura 7 Región V_2: todos los puntos que están más cerca de p_2 22
Figura 8 Triángulo que une p_2, p3 y p_4 23
Figura 9 Triangulación de Delaunay asociada con el diagrama de Voronoi 23
Figura 10 Triangulación de Delaunay 24
Figura 11 Analogía entre una neurona artificial y una biológica 26
Figura 12 Relación entre las entradas y salida de una RNA 31
Figura 13 Ejemplo de una topología de RNA 31
Figura 14 Puntajes de los 30 postulantes del año anterior 34
Figura 15 Topología de la red 35
Figura 16 Hiperplano que separa a los aptos de los no aptos 37
Figura 17 Ubicación de los cinco postulantes de prueba 38
Figura 18 Reubicación de las neuronas ganadoras 42
Figura 19 comparación del primer vector con las cuatro m_i iniciales. 43
Figura 20 Datos de entrada 44
Figura 21 Datos en R^3 agrupados en cinco clústeres mapeados a R^2 45
Figura 22 Los 20 pueblos y las postas médicas iniciales 46
Figura 23 Cálculo de la variación parcial (kernel del vecindario) de cada neurona ganadora 48
Figura 24 Cálculo de la variación de cada neurona ganadora en epoch 1 49
Figura 25 Asignación final de los pueblos a las postas médicas epoch 1000 51
Figura 26 Unir neuronas ganadoras 52
Figura 27 Bisectriz entre las postas 3 y 4 52
Figura 28 Bisectriz entre las postas 3 y 1 53
Figura 29 Diagrama de Voronoi 53
Figura 30 (a) SOM [1 6] (b) SOM [2 3] /c) SOM [3 2] 55
Figura 31 Topólogías (a) aleatoria (b) triangular (c) Cuadricula (d) Hexagonal 56
Figura 32 IW para epoch = 1 58
Figura 33 IW para epoch = 3 58
Figura 34 IW para epoch = 5 59
Figura 35 IW para epoch = 200 59
Figura 36 Resultado del comando sim 60

Figura 37 Resultado de la aplicación de la red entrenada 61
Figura 38 Resultado de clasificar los datos 62
Figura 39 Resultado de emparejar los datos con sus clúster correspondientes 62
Figura 40 Resultado de gscatter(x(:,1),x(:,2),group, 'brgkm','o',5) 64
Figura 41 Estructura de la red SOM 65
Figura 42 Resultado de plotsompos de la red 65
Figura 43 Cantidad de elementos de cada clúster 66
Figura 44 Mapas parciales del distrito de Bellavista - Callao 68
Figura 45 Intersecciones del distrito de Bellavista - Callao 68
Figura 46 Cálculo de la distancia entre dos nodos 69
Figura 47 Intersecciones (nodos) de una región del distrito en estudi0 70
Figura 48 Estructura de la red neuronal auto organizada escenario 1 74
Figura 49 plotsomhits(Red) para una dimensión de 1x5 con 1000 epoch 75
Figura 50 Las esquinas de Bellavista en cinco clúster dimensión (1x5) 76
Figura 51 Distancias máximas para epochs desde 901 a 1200 78
Figura 52 Diagrama de cajas de la distancia máxima (300 epochs diferentes) 79
Figura 53 Histograma de las distancias máximas 80
Figura 54 Gráfico Q-Q normal de las distancias máximas 80
Figura 55 Configuraciones posibles para 6 camiones 81
Figura 56 Estructura de la red neuronal auto organizada escenario 2 82
Figura 57 plotsomhits(Red) para una dimensión de 1x6 con 1000 epoch 83
Figura 58 Las esquinas de Bellavista en seis clúster dim(1x6) 83
Figura 59 Las esquinas de Bellavista pertenecientes al Clúster 1 85

Resumen

“ZONIFICACIÓN DINÁMICA PARA EL RECOJO DE BASURA USANDO REDES DE KOHONEN: CASO DISTRITO DE BELLAVISTA-CALLAO

La municipalidad de Bellavista Callao cuenta con camiones recolectores para el recojo puerta a puerta de los residuos sólidos domiciliarios (RSD), pero estos no siempre están disponibles, ya sea por desperfectos o por mantenimiento, y se tiene que reasignar las zonas en función del número de camiones disponibles. El propósito de la investigación fue zonificar (clusterizar) el distrito para un número variable de camiones usando una red autoorganizada de Kohonen, para lo cual se realizaron dos programas computacionales: uno que determina las dimensiones de la red neuronal y el número de epochs para entrenamiento con la finalidad de obtener una distancia máxima menor al interior de cada clúster y otro que realiza la clusterización. Se analizaron dos escenarios: uno para cinco camiones y otro para seis. En el escenario de cinco camiones se consideró una dimensión 1x5; en el escenario con seis camiones, se evalúo primero una red con dimensión 1x6, otra de 2x3; y la tercera, de dimensión 3x2, se encontró que la configuración que proporciona una menor distancia máxima al interior de cada clúster es 1x6 con 1000 epochs. El método puede ser utilizado en cualquier distrito, para cualquier número de camiones, solo se necesitan las coordenadas de cada esquina del distrito.

Palabras claves: Redes auto organizadas, Kohonen, Voronoi, residuos sólidos domiciliarios, zonificación

Abstract

“DYNAMIC ZONING USING KOHONEN NETWORKS TO PICK UP DOMICILIARY WASTE IN BELLAVISTA-CALLAO DISTRICT”

The municipality of Bellavista Callao has collection trucks for door-to-door collection of household solid waste (SW), but these are not always available, either because of damage or maintenance, and the zones have to be reallocated according to the number of trucks available. The purpose of the research was to cluster the district for a variable number of trucks using a self-organizing Kohonen network, for which two computer programs were developed: one module that determines the dimensions of the neural network and the number of epochs for training in order to obtain a smaller maximum distance within each cluster, and another module that performs the clustering. Two scenarios were analyzed: one for five trucks and one for six trucks. In the scenario with five trucks, a 1x5 dimension was considered; in the scenario with six trucks, a network with dimension 1x6 was evaluated first, another with 2x3; and the third, with dimension 3x2, it was found that the configuration that provides a smaller maximum distance to the interior of each cluster is 1x6 with 1000 epochs. The method can be used in any district, for any number of trucks, only the coordinates of each corner of the district are needed.

Keywords: Self Organized Networks, Kohonen, Voronoi, Solid household waste, Zoning

INTRODUCCIÓN

Las estadísticas muestran que la población mundial superó los siete mil millones en el año 2015 y está creciendo a un ritmo anual de 1,2 por ciento, con un 46% de esta población en las zonas urbanas. Casi la mitad de la humanidad vive en ciudades y la población urbana aumentó desde aproximadamente 750 millones en 1950 hasta 3.600 millones en 2011. Se estima que hacia 2030, casi un 60% de la población mundial residirá en zonas urbanas, (Suocheng, 2001) lo cual implica una generación de residuos sólidos proporcional.

A nivel mundial, los residuos sólidos urbanos generados en 1997 fueron de alrededor de 490 millones de toneladas, con una tasa estimada de crecimiento anual de 3.2% a 4.5% en los países desarrollados, y del 2% a 3% en las naciones en vías de desarrollo; se estima que esta aumentará a 3400 millones de toneladas en 2050. (Banco_mundial, 2018) lo cual muestra que el problema de los residuos sólidos es de alta preocupación para los gobiernos tanto globales como locales.

A nivel nacional, la cantidad promedio diaria de residuos sólidos recolectados, en el 2015 era 22 390 372 toneladas; , en el 2016, solo en la provincia de Lima se producía diariamente 8 670 toneladas de residuos sólidos generando un estimado per cápita (Kg/hab/día) de 1,1. (INEI, 2017), dichos residuos terminan en el mejor de los casos, en los rellenos sanitarios, aunque muchos terminan en botaderos informales.

“La mala gestión de los desechos está perjudicando la salud humana y los entornos locales, agravando al mismo tiempo los desafíos que plantea el cambio climático”, afirmó Laura Tuck, vicepresidenta de Desarrollo Sostenible del Banco Mundial. (Banco_mundial, 2018)

Los responsables de la gestión de residuos son las municipalidades distritales, lo cual les representa una alta inversión económica y una gran presión social por parte de los ciudadanos.

Una parte del proceso de gestión es la recolección de residuos domiciliarios (Tchobanoglous, 1994) para lo cual las municipalidades disponen de un número limitado de camiones que están sujetos a fallas que afectan el normal proceso de recolección. La Municipalidad tiene rutas de recojo de residuos sólidos, las mismas que se han establecido sobre la base de sus recursos disponibles y la experiencia, pero una pregunta que se puede presentar es: ¿cuál debe ser la ruta de recolección el día que se dispone de solo n-camiones?

Un problema para tratar es el de la asignación de rutas óptimas para lo cual, la Investigación Operativa proporciona diversos métodos, pero un tópico que se puede considerar previo es el de segmentar el distrito en función del número de camiones disponibles, para recién hacer rutas en cada segmento.

El objetivo del presente estudio es encontrar una segmentación (zonificación) variable del distrito de Bellavista en función del número de camiones recolectores disponibles tal que se minimice el tiempo de recorrido de los mismos, para lo cual se sectorizó las esquinas utilizando una red neuronal de Kohonen.

La tesis se organiza de la siguiente manera: En el capítulo 1 se plantea el problema de la recolección de residuos en Bellavista Callao. En el capítulo 2 se presenta el marco de referencia para describir investigaciones relacionadas, para dar las bases históricas y las bases conceptuales y así clarificar los conceptos. En el capítulo 3 se describe el método empleado. En el capítulo 4, se presentan los resultados obtenidos, se analizan y sistematizan para finalmente presentar las conclusiones y recomendaciones.

CAPITULO 1: PLANTEAMIENTO DEL PROBLEMA

1.1. Situación Problemática

El distrito de Bellavista Callao es una zona residencial cuya población mayoritariamente pertenece a los niveles socioeconómicos B (65%) y C (34%). (Municipalidad de Bellavista, Plan de gestión y manejo de residuos solidos del distrito de Bellavista 2016-202., 2016) Tiene zonas comerciales y fabricas que generan rentas para la Municipalidad, pero también generan residuos sólidos y ésta no se da abasto para realizar una gestión de los residuos sólidos satisfactoria.

1.2. Descripción del sistema

De acuerdo con la evaluación de la gestión de residuos sólidos en el Perú, la generación per cápita de residuos sólidos domiciliarios, se incrementa proporcionalmente al crecimiento de la economía. (Ministerio de Ambiente, 2012)

Según el plan de Gestión y manejo de residuos sólidos de Bellavista de 2016-2021 (Municipalidad de Bellavista, Plan de gestión y manejo de residuos solidos del distrito de Bellavista , 2016), la proyección poblacional para el 2021 es de 78 382.15 habitantes, la generación de residuos sólidos por habitante por día es de 0,55kg/hab/día y la densidad de residuos es de 218,26 kg/metro cuadrado.

El crecimiento demográfico de los últimos años ha generado el aumento de los residuos sólidos y como consecuencia de ello, se ha generado las siguientes situaciones no deseadas:

- Acumulación de residuos sólidos frente a las viviendas, lo cual atrae a animales que rompen las bolsas buscando restos de comida.
- Generación de enfermedades relacionadas a este mal manejo de los residuos sólidos.
- Incremento de puntos críticos donde las familias acumulan los residuos sólidos.
- Fomento de una inadecuada cultura de manejo de residuos sólidos.

Caracterización del distrito

El distrito de Bellavista es uno de los siete distritos que conforman la Provincia Constitucional del Callao del Perú, es uno de los pocos distritos en poseer una forma regular, pues se presenta como un rectángulo cuya base está alineada con el eje de la Av. Venezuela

- Limita con el distrito de La Perla- Callao y el distrito limeño de San Miguel por el Sur.
- Con el distrito del Callao por el norte, siendo el limite la Av. Óscar R. Benavides (antes Av. Colonial)
- Con el Hospital Naval “Santiago Távara” (límite con la Universidad Nacional Mayor de San Marcos – Lima) por el Este.
- Con la Jr. Santiago Vigil (Callao) por el Oeste.

En la Figura 1 se muestra el mapa del distrito de Bellavista



Figura 1 Mapa de Bellavista-Callao

El servicio de recolección se realiza diariamente de lunes a domingo en tres turnos. A la fecha la Municipalidad cuenta con cinco unidades compactadores de carga trasera (cuatro con capacidad de 15m3 y uno con 21m3) y cinco camiones para los contenedores soterrados solamente. En el turno de 09:00 pm a 05:00 am se recoge residuos en todo el distrito excepto en las instituciones educativas, de 05:00 a 01:00pm y de 01:00pm a 09:00 pm se repasa por las avenidas principales y se recoge de las instituciones educativas (Municipalidad de Bellavista, Plan de gestión y manejo de residuos solidos del distrito de Bellavista , 2016). En el presente estudio sólo se considera el servicio de recolección de residuos domiciliarios, no el de contenedores.

Si bien es cierto que la Municipalidad cuenta con cinco camiones recolectores, estos no siempre están disponibles por desperfectos o por mantenimiento preventivo. Por lo que es necesario en algún momento hacer zonificaciones de una manera dinámica para reasignar los camiones.

Problemática del recojo de residuos sólidos

En el Informe (Municipalidad de Bellavista, Plan operativo anual, 2018) (Página 16) se menciona la identificación de los siguientes problemas:

- Carencia de herramientas y maquinarias
- Demora en la entrega de materiales de seguridad para el personal.
- Deficiencia en la infraestructura del local, donde labora el personal operativo
- **Demora en solución de problemas con las unidades vehiculares que se encuentran en mal estado.**
- **Demora en la atención de los mantenimientos correctivos y preventivos a las unidades vehiculares**.
- Existencia de puntos críticos de desmonte
- Existencia de maleza cerca a los contenedores
- Falta de adaptación de los vecinos al uso de contenedores

Los tres primeros son aspectos logísticos y solo requieren de la voluntad de las autoridades para ser resueltos.

Los dos siguientes pertenecen al ámbito de la Investigación Operativa y le competen al área de mantenimiento, se requiere de un estudio para hacer un diagnóstico y plantear soluciones, siendo su principal efecto que el número de camiones recolectores no siempre es el mismo, por lo tanto, el tener un ruteo óptimo fijo no es de mucha utilidad, es por ello que con el presente trabajo de investigación se pretende hacer una zonificación dinámica en función del número de camiones disponibles.

Los tres últimos son muy puntuales y se refieren al medio ambiente que afecta y es afectado por la gestión municipal de residuos sólidos, y se pueden resolver o disminuir con campañas de sensibilización y/o capacitación de los vecinos.

Desde la óptica de la Investigación Operativa, en la gestión de RSD se identifican diversos problemas como:

- Asignación de rutas más cortas de los camiones recolectores
- Optimización de la capacidad de los camiones recolectores
- Localización con cubrimiento de contenedores
- Localización de rellenos sanitarios o puntos de acopio para mantener armonía con el entorno y que no ofrezcan riesgos a la salud de la población ni de contaminación del medio ambiente
- **Asignación de las manzanas** en grupos uniformes más pequeños con características y necesidades semejantes y formación de clústeres para facilitar el recorrido de los camiones recolectores o de barrenderos.
- Planificación del mantenimiento y reparación de los camiones recolectores

Estos dos últimos problemas están muy relacionados entre sí, pues si en un día dado hay un número variable de camiones en reparación o mantenimiento, se debe hacer una reasignación de rutas para poder cubrir las zonas de recojo de los camiones no disponibles.

Si el mantenimiento es preventivo, se puede hacer con anticipación las reasignaciones respectivas basadas en la experiencia del despachador, pero una falla inesperada ocasionaría que un sector se quede sin recibir el servicio o que otros camiones se sobrecarguen y/o aumenten sus viajes al vertedero con las consiguientes molestias, retrasos y sobretiempos.

Una mejor asignación de los recursos disponibles, es decir, de los camiones recolectores, podría disminuir los volúmenes de residuos en la vía pública, no es el único factor que influye en el problema, pues hay una dimensión relacionada con la cultura ambiental de los ciudadanos que va desde disminuir el uso de desechables, hasta separar en fuente los residuos para propiciar la reutilización y reciclaje, pero ese no es tema de este estudio, sino, el tema central es la asignación de manzanas a los camiones recolectores

1.3. Formulación del Problema

Todo lo mencionado anteriormente, lleva al siguiente cuestionamiento:

1.3.1 Problema General

¿Cómo sectorizar dinámicamente las esquinas de un distrito para minimizar la distancia recorrida en función del número de camiones recolectores disponibles aplicando una red de Kohonen?

1.3.2 Problemas Específicos:

1. ¿Cómo construir una base de datos geográfica que permita generar un grafo y una lista de adyacencia que represente las calles distrito de Bellavista?
2. ¿Cómo construir una red de Kohonen para sectorizar las esquinas de un distrito aprovechando al máximo la disponibilidad de los camiones recolectores?
3. ¿Cómo implementar eficiente y amigablemente para un usuario final una red de Kohonen en un medio computacional?

1.4. Justificación de la investigación

Justificación práctica

La gestión de residuos sólidos es un problema multi dimensional, el informe (ONU, 2018) presenta una clara relación entre las metas de los ODS y la gestión de residuos. Lo cual incluye a los ODS 3: una vida sana para todos; 6: agua limpia y saneamiento; 11: ciudades seguras; 12: producción y consumo responsables; 14: vida submarina; 15: ecosistemas terrestres. Es un problema tanto por su exceso como por las dificultades que existen para su eliminación, siendo la recolección el inicio del proceso, pasando por actividades de clasificación, reciclaje, eliminación de contaminantes etc. Por lo tanto, un sistema de recolección eficiente, es decir en menor tiempo en la municipalidad de Bellavista, beneficiaría a toda la población del distrito, pues se minimizarían los residuos en las vías públicas y con ello disminuiría el riesgo a contraer enfermedades, mejorando la calidad de vida de los pobladores.

Los resultados del presente estudio podrían dar lugar a otras investigaciones desde una mirada social, como son los usos y costumbres en la generación y reaprovechamiento de los residuos sólidos, o desde una mirada técnica, como es la localización de contenedores públicos y centros de disposición final entre otros. Además, que la metodología empleada podría ser usada para estudios similares en otros distritos.

Justificación teórica

El uso de Redes neuronales, en este caso redes de Kohonen, propicia el uso de herramientas tecnológicas para resolver problemas clásicos de la investigación operativa aplicando medios computacionales, que si bien es cierto sólo proporcionan soluciones aproximadas, permiten resolver problemas complejos a los cuales se les tendría que hacer muchas simplificaciones para modelarlos según los métodos de optimización tradicionales. La red neuronal de Kohonen al tener un aprendizaje no supervisado, no necesita información recolectada especialmente, sino solamente se necesita la ubicación geográfica de las esquinas del distrito en estudio.

1.5. Objetivos

1.5.1. Objetivo General

Sectorizar dinámicamente las esquinas del distrito de Bellavista para minimizar la distancia recorrida en función del número de camiones recolectores disponibles aplicando una red de Kohonen

1.5.2. Objetivos específicos

1. Construir una base de datos geográfica para generar un grafo y una lista de adyacencia que representen las calles del distrito de Bellavista.
2. Construir una red de Kohonen para sectorizar las esquinas de un distrito aprovechando al máximo la disponibilidad de los camiones recolectores.
3. Implementar una red de Kohonen, en un medio computacional, eficiente y amigable para el usuario final.

1.6. Hipótesis de investigación

1.6.1. Hipótesis General

La sectorización de esquinas usando una red de Kohonen al agrupar las esquinas más cercanas en función del número de camiones recolectores, minimiza la distancia recorrida

1.6.2. Hipótesis específicas

1. Una base de datos geográfica se puede utilizar para generar un grafo y una lista de adyacencia que representen las calles del distrito de Bellavista.
2. Utilizando una red de Kohonen para sectorizar las esquinas de las manzanas del distrito, se maximiza el aprovechamiento de los camiones recolectores disponibles.
3. El programa computacional que implementa la red de Kohonen que genera una sectorización dinámica es eficiente y amigable para el usuario final.

CAPITULO 2: MARCO DE REFERENCIA

El objetivo del presente capítulo es presentar el contexto del estudio, primeramente, se presenta investigaciones relacionadas, las bases epistemológicas de la investigación, luego las bases históricas de la problemática de la gestión de residuos sólidos, y finalmente las bases teóricas necesarias para la mejor comprensión del estudio.

2.1. Antecedentes del Problema

(Komolafe, 2012) presenta una propuesta de manejo de residuos sólidos para la ciudad de en Nigeria, mostrando cuatro escenarios: dos en países desarrollados como son Suecia y Dinamarca, y dos en países en vías de desarrollo como son Malasia y Sudáfrica. Sustenta su propuesta en una cultura ciudadana de manejo eficiente de la basura y en una legislación fuerte por parte de los gobiernos locales.

Las realidades que presentan son culturalmente muy diferentes a la del Perú, pero servirán como marco para establecer las características de un sistema de gestión de residuos sólidos eficiente.

(Fatin, S., 2011) expone diversos factores de éxito o fracaso en la gestión de residuos, como son: factores administrativos, la falta de habilidades, conocimientos y equipamiento de las municipalidades, la carencia de leyes promulgadas específicamente para la gestión de los residuos sólidos o inadecuada aplicación de la ley por conocimiento limitado, incluyendo la debilidad de las políticas de acción del gobierno. Presenta un análisis de los siguientes factores: falta de participación de la comunidad: los hábitos y actitudes de las personas al tirar la basura, la falta de conciencia, de apoyo y de interés de las comunidades en la

gestión de residuos, además de otras contingencias como son aumento de la población, el rápido desarrollo económico, la escasez de tierras para la eliminación de residuos, la creciente migración a las grandes ciudades entre otras. Dichos factores sirven para focalizar el problema de la gestión de RSD

(Luna, 2003) estudia el comportamiento de los ciudadanos en relación al ciclo de generación de basura (compra, manejo y desecho) para identificar los factores que facilitan las conductas ecológicamente responsables, además compara conductas reportadas en Barcelona y México. Si bien es cierto que su enfoque es de comportamiento social, presenta un capítulo de “Basurología” donde desarrolla conceptos útiles para el presente estudio. Además, presenta los volúmenes estándar de basura diaria por persona en las ciudades de Barcelona, ver Tabla 1 y México DF, ver Tabla 2 dos ciudades con diferentes niveles socioeconómicos, culturales y formas de administrar el servicio de recolección de basura, con lo cual se puede hacer una aproximación a la realidad del distrito en estudio.

Encuentra volúmenes de producción de basura por persona muy parecidos en Barcelona, 1.34 kg por persona por día y México DF 1.38 kg respectivamente.

Tabla 1 Cantidad de residuos en el área metropolitana de Barcelona

Año	Toneladas	Kg/hab/día
1996	1.290.417	1,22
1997	1.374.711	1,3
1998	1.373.796	1,3
1999	1.449.719	1,37
2000	1.407.768	1,33
2001	1.428.817	1,34

Fuente Agencia Metropolitana de residuos, 2002

Tabla 2 Cantidad de residuos en el distrito federal

Año	Toneladas	Kg/hab/día
1997	4,121.767	1,31
1998	4,220.944	1,26
1999	4,150.601	1,27
2000	4,350.691	1,38
2001	4,350.691	1,38

Fuente: Secretaría de desarrollo social, subsecretaría de desarrollo urbano y ordenación territorial, 2002

Sus conclusiones se limitan a describir comportamientos de la población en relación al problema de residuos sólidos.

(Negreiros & Palhano, 2006) hacen un estudio del sistema recolector de basura con el fin de establecer sectores (clústeres) que minimicen el tiempo de recorrido. Este trabajo sirvió de base y de motivación para hacer una sectorización previa al ruteo de camiones.

En (Bonomo, Duran, Larumbe, & Marenco, 2006) se presenta una aplicación de la Optimización Combinatoria para optimizar el ruteo de camiones en la Ciudad de Buenos Aires. Dada una zona con distintos puntos por los que el vehículo debe pasar exactamente una vez, el programa desarrollado devuelve un orden de los puntos para que la distancia sea mínima. Si bien es cierto que la tesis trata de camiones recolectores que recogen residuos de contenedores, El trabajo presenta un método para la construcción del grafo sobre el cual se busca la solución.

2.2. Marco epistemológico de la investigación

Una cuestión importante en toda investigación es la búsqueda de la verdad como idea reguladora. Importa determinar cómo se adquiere el conocimiento, cómo se organiza y cómo puede este ser generalizado.

Para el mejor entendimiento de estos aspectos, se exponen a continuación los diferentes enfoques metodológicos que sustentan el presente estudio.

La verificabilidad no se basa en la objetividad de la investigadora sino en la presentación de las teorías a la crítica abierta del mundo científico,

siendo el principal componente para evaluar los resultados, un criterio de economía y de eficacia.

La adquisición de información y la representación de la realidad, se basa en el enfoque sistémico, el cual se caracteriza por la construcción de sistemas teóricos o "modelos" abstraídos de la realidad. Es decir, los modelos creados guardan una relación de correspondencia con la realidad. El enfoque sistémico también se caracteriza por los conceptos de “entidades” y "procesos", lo que equivale a decir, que un sistema es una dinámica entre entidades (ontología) que tienen relaciones, las cuales son creaciones analíticas de la inteligencia (gnoseología), cuya correspondencia con la realidad debe ser probada.

El principal problema metodológico del enfoque sistémico es justamente la construcción de "sistemas abstraídos", ya que éste depende del objetivo para el cual fue creado y de la capacidad y creatividad del investigador. En el presente estudio, en el caso de las variables de entrada, se usan variables numéricas, cuyos valores son obtenidos de google maps y para hallar las variables de salida se utilizan métodos cuantitativos sustentados en teorías probadas y en técnicas computacionales cuyos algoritmos se presentan en un anexo para permitir la reproducibilidad.

El enfoque sistémico es metodológicamente dinámico, puesto que se orienta a describir y explicar dos momentos: la situación “actual” y la “deseada” buscando en cada caso acortar la brecha entre ambas con la solución propuesta. Pues, parafraseando a Aristóteles, de poco sirve la teoría si no se concreta en la realidad

2.3. Marco histórico

Las actividades humanas generalmente producen residuos. “Los más voluminosos son los residuos agrícolas, después los producidos por las actividades mineras, los derivados de la industria, los residuos urbanos y en último lugar los derivados de la producción de energía.” (UNED, 2003)

Los residuos urbanos constituyen una pequeña fracción del total, sin embargo, generan un problema para la calidad de vida en las zonas urbanas, por lo que cada gobierno, ya sea del país, regional o local establece una normatividad ad hoc para enfrentarlo, las normas más antiguas de la gestión ambiental son el acta de sanidad urbana aprobada en Inglaterra (1888) y “Rivers and harbors act” en los Estados Unidos (1899). (Tchobanoglous, 1994)

Luego de la segunda guerra mundial, “el gerenciamiento de residuos sólidos se focalizó según consideraciones económicas y con las consecuencias ambientales de disposición de residuos en Rellenos Sanitarios. A principio de la década del 1970, la preocupación pública internacional respecto al potencial y la reducción de los recursos naturales causó un cambio en el foco del gerenciamiento de los residuos sólidos.” (Theisen, 2000)

Con el tiempo los países establecieron mecanismos regulatorios y se adoptó el concepto de jerarquía de residuos (OECD, 2008) lo que llevó a adoptar el principio de inversión jerarquizada para cambiar de un enfoque de inversión pública/privada a la financiación de recursos hacia la prevención de residuos y la sostenibilidad ambiental.

2.4. Marco teórico

En el presente acápite se presenta las bases teóricas que permiten entender los procesos, para luego desarrollar los conceptos involucrados. Los temas tratados son:

- Residuos sólidos
- Redes neuronales
- Redes auto organizadas o de Kohonen
- Diagramas de Voronoi

2.4.1. Residuos Sólidos

Las definiciones de Residuos Sólidos varían en la literatura, las que van desde basura hasta desechos, siendo el término residuo el aceptado actualmente.

Según la secretaria de ambiente y desarrollo sustentable de la nación Argentina "**Residuo** es todo elemento, material, objeto o sustancia que, como consecuencia de los procesos de consumo y desarrollo de actividades humanas, es desechado o abandonado" .https://www.argentina.gob.ar/ambiente/control/rsu

Los orígenes de los Residuos Sólidos (RS) en una comunidad están, en general, relacionados con el uso de suelo y su localización. Una clasificación es la siguiente:

- Institucional.
- Construcción y demolición.
- Servicios municipales (barrido de calles, arreglo de parques, entre otros
- Industrial
- Doméstico y
- Comercial.

El presente estudio se centra en estos dos últimos

Sistemas de gestión de residuos sólidos

La gestión de residuos sólidos comprende toda actividad técnica administrativa de planificación, coordinación, concertación, diseño, aplicación y evaluación de políticas, estrategias, planes y programas de acción de manejo de los residuos sólidos de ámbito nacional, regional y local.

En el presente estudio se usó las definiciones del Banco interamericano de desarrollo (Terraza, 2009)

> *"El servicio de aseo urbano comprende las siguientes actividades relacionadas al manejo de los residuos sólidos municipales: almacenamiento; presentación; recolección;*

transporte; transferencia; tratamiento; disposición sanitaria; barrido y limpieza de vías y áreas públicas; recuperación y reciclaje. (Terraza, 2009)*"*

Los sistemas de gestión deben brindar los siguientes servicios:

- Monitoreo de la producción y de la composición de los residuos sólidos urbanos
- Distribución y mantenimiento de la dotación de cubos y contenedores para residuos sólidos urbanos
- **Recogida y transporte de los residuos** hasta los centros de tratamiento
- Tratamiento de los residuos con recuperación y aprovechamiento de las fracciones valorizables, y eliminación de las no valorizables

Estos servicios se deben complementar con las funciones de sensibilización de la población acerca de la minimización de la producción de residuos y su separación en origen. (Terraza, 2009)

Previa a la etapa de recojo y transporte de residuos sólidos suele hacerse una sectorización, el objetivo del presente estudio es precisamente sectorizar una zona geográfica en función del número de camiones recolectores disponibles.

El problema de ruteo de vehículos tanto el de visitar un numero finito de lugares (VRP), como el de recorrer las calles es un problema complejo que está clasificado como np-completo (Garey, M y Johnson, D, 1979) pp 211-212 por lo cual puede ser poco fructífero buscar soluciones exactas, las redes neuronales como un enfoque computacional brindan buenas soluciones en tiempos computacionales razonables.

2.4.2. Diagrama de Voronoi

El diagrama de Voronoi de un conjunto de puntos es una partición del plano en regiones, una para cada punto, de modo que la cada región contiene todos los puntos del plano que están más cerca, de un punto p, que de los otros. El problema al calcular diagramas de Voronoi rápidamente es que su estructura depende de la noción de distancia y el tipo de puntos. (Klein, 1989)

Tales particiones son de gran importancia al diseñar rutas pues al reunir los puntos más cercanos en un clúster facilitará después el ruteo de vehículos al interior del mismo, pues los puntos (esquinas) estarán agrupados en función de la distancia entre ellos.

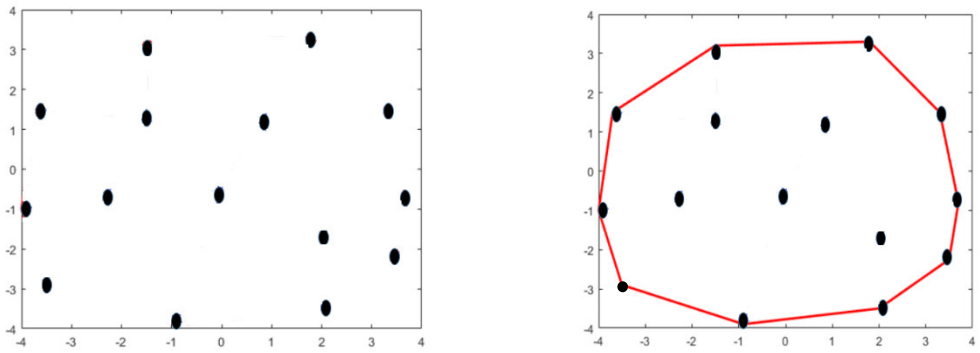
Para la mayor comprensión de los conjuntos de Voronoi se introduce los conceptos de espacio afín, casco convexo, triangulación y teselación.

La noción de **espacio afín** procede del descubrimiento de geometrías diferentes de la Geometría Euclidiana, un espacio afín es una estructura geométrica que generaliza algunas de las propiedades de los espacios euclidianos de tal manera que sus elementos son independientes de los conceptos de distancia y medida de ángulos, manteniendo solo las propiedades relacionadas con el paralelismo, por lo tanto, un espacio euclidiano es afín, pero no lo inverso. En un espacio afín, no hay un punto de origen. Por lo tanto, ningún vector tiene un origen fijo y ningún vector puede asociarse únicamente a un punto. (Klein, 1989) En el presente estudio, los puntos son lugares georreferenciados, es decir, se indica su latitud y longitud y si bien es cierto que forman un espacio afín serán tratados como un espacio euclidiano.

El **casco convexo** (convex hull) o envoltura convexa de una forma geométrica es el conjunto convexo más pequeño que la contiene.

En la Figura 2 se muestra un conjunto de puntos (lugares) y el casco convexo del mismo.

Figura 2 Conjunto S de puntos (lugares) y casco convexo del conjunto S

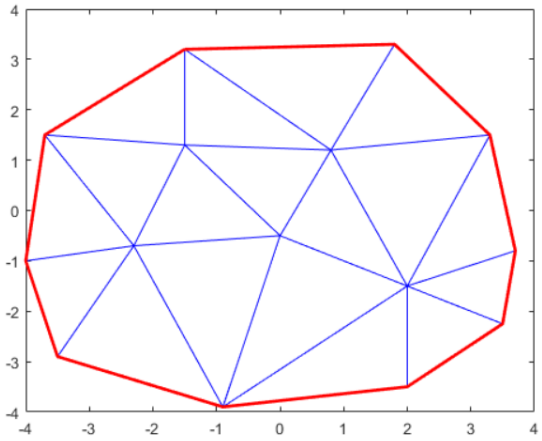


Fuente Gallier, J. & Quaintance, J.,(2017)

En geometría, una triangulación es una subdivisión de un objeto planar en triángulos, En la mayoría de los casos, se requiere que los triángulos de una triangulación sean vecinos entre sí borde-a-borde y vértice-a-vértice. (Gallier, J. & Quaintance, J., 2017)

En la Figura 3 se muestra la triangulación del conjunto S

Figura 3Triangulación del conjunto S



Fuente Gallier, J. & Quaintance, J.,(2017)

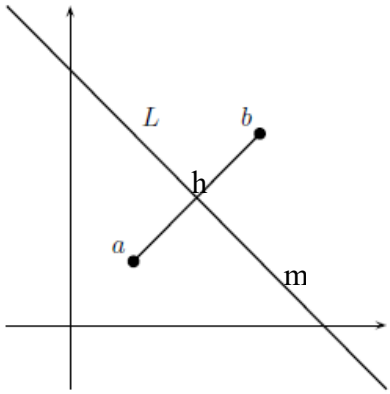
En general, si se da un conjunto finito S de objetos p_i en un espacio M, calcular el diagrama de Voronoi de S significa dividir el espacio en regiones, R (p_i, S), de tal manera que R (p_i, S) contiene todos los puntos de M que están más cerca de p_i que de cualquier otro objeto p_j en S. Para lo cual se debe definir qué tipo de objetos ("puntos") se consideran y cuál

es la medida de distancia subyacente en el espacio M. Además, se debe especificar qué hacer con los puntos que tienen más de un vecino más cercano en S. (Klein, 1989)

Es útil encontrar una partición del espacio en regiones que contengan cada una un solo punto de S y tengan algunas propiedades específicas. También suele ser útil encontrar triangulaciones del casco convexo de S que tengan también algunas propiedades.

Los dos problemas están relacionados. Para resolver el primer problema, suponer primero que es un plano, es decir, tiene dimensión 2, dados dos puntos distintos a y b en S, la línea ortogonal al segmento de línea (a,b) y que pasa por el punto medio de este segmento es el lugar geométrico de todos los puntos que tienen la misma distancia tanto a “a” como a “b”. Se llama la línea bisectriz de (a,b). La línea bisectriz L de dos puntos se ilustra en la Figura 4

Figura 4 Recta bisectriz de (a,b)



Donde h= ½ a + ½ b es el punto medio del segmento de línea (a,b)

m un punto arbitrario en la línea bisectriz.

La ecuación de tal recta es tal que hm es ortogonal a ab. Sea m = (x, y), a = (a1, a2), b = (b1, b2), la ecuación de esta recta es:

$$(b1 - a1)\left(x - \frac{a1 + b1}{2}\right) + (b2 - a2)\left(x - \frac{a2 + b2}{2}\right) = 0$$

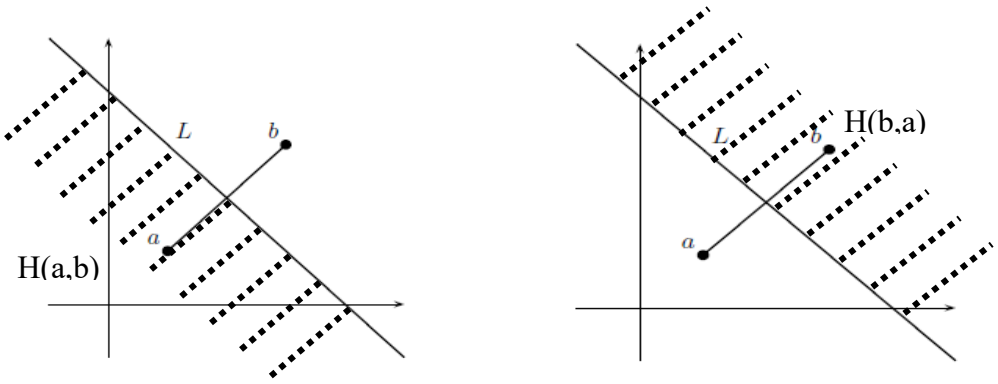
que también se puede escribir como:

$$(b1 - a1)x + (b2 - a2)y = \frac{b_1^2 + b_2^2}{2} - \frac{a_1^2 + a_2^2}{2}$$

El semiplano cerrado H (a, b) que se muestra en la Figura 5 contiene el punto a y con límite en la línea bisectriz es el lugar geométrico de todos los puntos de manera que:

$$(b1 - a1)x + (b2 - a2)y \leq \frac{b_1^2 + b_2^2}{2} - \frac{a_1^2 + a_2^2}{2}$$

Figura 5 Semi plano H(a,b) Semi plano H(b,a)



Análogamente el semiplano cerrado H (b, a) contiene el punto b y con el límite en la línea bisectriz es el lugar geométrico de todos los puntos de manera que

$$(b1 - a1)x + (b2 - a2)y \geq \frac{b_1^2 + b_2^2}{2} - \frac{a_1^2 + a_2^2}{2}$$

En consecuencia, el semiplano cerrado H (a, b) es el conjunto de todos los puntos cuya distancia a “a” es menor o igual que la distancia a “b”, y viceversa para H (b, a).

Graham y Yao citados por (Gallier, J. & Quaintance, J., 2017) consideran a modo de ejemplo el problema llamado el problema de las oficinas de correo. Dado cualquier conjunto P = {p_1, p_2….p_n} de n puntos en el plano (oficinas de correos), para cualquier punto arbitrario “x” encontrar cual oficina de correo está más cerca de “x”. Como x es arbitrario, es deseable calcular previamente los conjuntos V (p_i) –

vecindarios de p_i - que consiste de todos los puntos que están más cerca de p_i que de cualquier otro punto p_j diferente de p_i.

Si se conocieran los conjuntos V (p_i) para todo i, la respuesta sería *la oficina de correo p_i tal que x pertenece a V (pi)*. Por lo tanto, queda por calcular los conjuntos V (pi). Para esto, si x está más cerca de p_i que de cualquier otro punto p_j diferente de p_i, entonces x está en el mismo lado que p_i con respecto a la línea bisectriz de p_i y p_j para cada j diferente de i y así V (p_i) es la intersección de todos los semiplanos H(p_i; p_j)

$$V(pi) = \bigcap_{j \neq i} H(pi, pj)$$

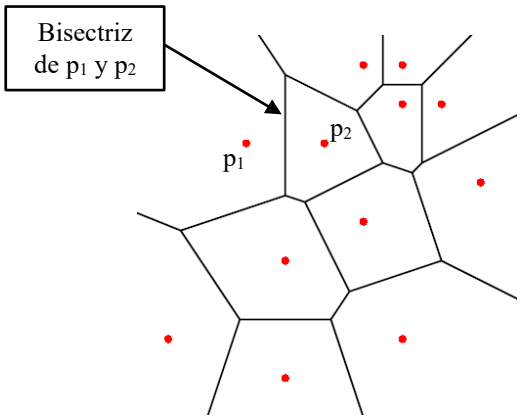
Sea E un espacio euclidiano de dimensión m≥1 Dado cualquier conjunto P = {p1, p2…pn} de n puntos en E, el diagrama de **Dirichlet Voronoi** denotado por Vor(P) de P = {p1, p2…pn} es la familia de subconjuntos de E que consiste en los conjuntos Vi = ⋂(p_i, p_j) para j≠i y de todas sus intersecciones

Los diagramas de Dirichlet Voronoi también se denominan diagramas de Voronoi, teselaciones de Voronoi o polígonos de Thiessen (Gallier, J. & Quaintance, J., 2017). Como intersecciones de conjuntos convexos, ya que son semiplanos cerrados, las regiones de Voronoi V_i son conjuntos convexos.

Que una región V_i esté acotada o no depende de la ubicación de p_i. Si p_i pertenece al borde del casco convexo del conjunto P, entonces V_i es no acotada de lo contrario lo es. En E^2 el casco convexo es un polígono convexo mientras que en dimensión tres, el casco convexo es un poliedro convexo.

En la Figura 6 se muestran los conjuntos de Voronoi para cada punto p_i.

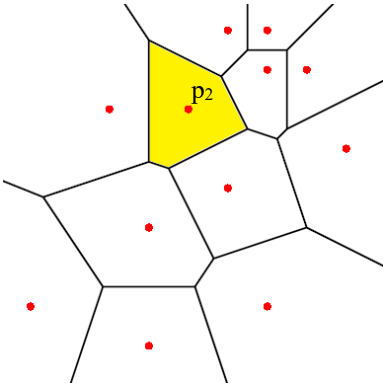
Figura 6 Diagrama de Voronoi para 12 puntos pi



Fuente (Gallier, 2017)

En la Figura 7 se muestra que todos los puntos que pertenecen a la región sombreada están más cerca de p_2 que de cualquier otro p_j para todo $j \neq 2$

Figura 7 Región V_2: todos los puntos que están más cerca de p_2

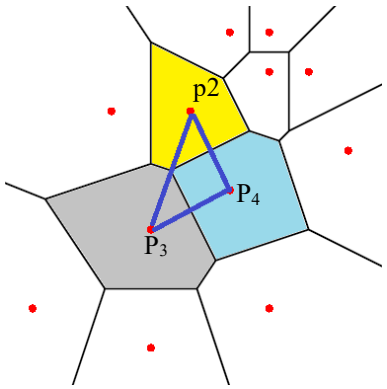


Fuente (Gallier, 2017)

Existe una relación entre los cascos convexos y los diagramas de Voronoi (no dirigido). Del diagrama de Voronoi se obtiene un grafo no dirigido de la siguiente manera: los vértices del grafo son los puntos p_i (cada uno correspondiente a una región única de Vor (P)), y hay una arista entre p_i y p_j si y solo si las regiones V_i y V_j comparten un borde. El grafo resultante se llama triangulación de Delaunay del casco convexo de P, en honor a Delaunay, quien inventó este concepto en

1934 (Gallier, J. & Quaintance, J., 2017). En la Figura 8 se muestra parte de la triangulación de Delaunay del conjunto de 12 puntos.

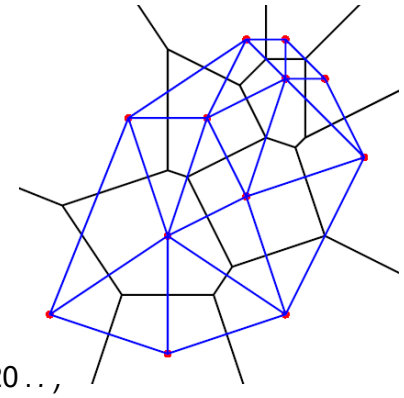
Figura 8 Triángulo que une p_2, p3 y p_4



Adaptado de (Gallier, 2017)

La triangulación completa se muestra en la Figura 9

Figura 9 Triangulación de Delaunay asociada con el diagrama de Voronoi

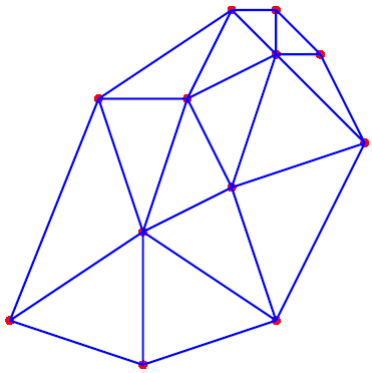


Fuente (Gallier, 20 . . ,

Como se observa, la triangulación de Delaunay y el diagrama de Voronoi son grafos duales, por lo cual la construcción de uno se hace en función del otro.

Eliminando el diagrama de Voronoi queda solo la triangulación que se muestra en la Figura 10 donde cada nodo es uno de los elementos del conjunto P={p_1,p_2, ….p_n}

Figura 10 Triangulación de Delaunay



Fuente (Gallier, 2017)

Una vez determinados los puntos $P=\{p_1, p_2 \ldots p_n\}$ se construye primero el grafo de Delaunay y tomando las bisectrices se construye el diagrama de Voronoi el cual define conjuntos de lugares que están más cerca de un p_i que de los otros.

La determinación del conjunto P es precisamente lo que hace la red de Kohonen, son precisamente los "centroides" que en este contexto se denominan "neuronas ganadoras", los conjunto de Voronoi están conformados por los nodos pertenecientes al clúster de cada neurona ganadora, a continuación se presentan las bases teóricas sobre las redes neuronales en general y las redes de Kohonen en particular.

2.4.3. Redes neuronales

Muchos investigadores utilizan métodos sin conocer la teoría que subyace, un ejemplo trivial de ello es el caminar, cuando se camina no se piensa en calcular o siquiera considerar las leyes de la gravedad o la fricción, lo mismo ocurre con las redes neuronales artificiales (RNA), sin embargo, en un ambiente académico es necesario establecer bases teóricas que permitan una mejor comprensión de los resultados encontrados.

La definición formal dada por (Kohonen T. , 1988) es *"Las redes neuronales artificiales son redes interconectadas masivamente paralelas de elementos simples (generalmente adaptativos) y sus organizaciones jerárquicas que tienen la intención de interactuar con los objetos del*

mundo real de la misma manera que lo hacen los sistemas nerviosos biológicos." Las RNA, no pretenden emular el cerebro humano, solo pueden describir un pedazo del tejido cerebral, por ejemplo, no más de unos pocos milímetros de diámetro, el cerebro es muchísimo más complejo no solo en cantidad de neuronas, sino en la diversidad de conexiones y tipos de neuronas (Kohonen, Hynninen, Kangas, & Laaksonen, 1995). Aun como herramienta, solo son una pequeña parte de la Inteligencia Artificial, las RNA realizan tareas muy pequeñas y específicas. Las redes neuronales operan sincrónicamente, esto es, una red neural solo dará una salida cuando se le presente una entrada. No es como el cerebro humano, que no funciona exactamente de forma síncrona. El cerebro humano responde a la entrada, pero producirá resultados en cualquier momento independientemente de si hubo una entrada o no. (Heaton, 2011)

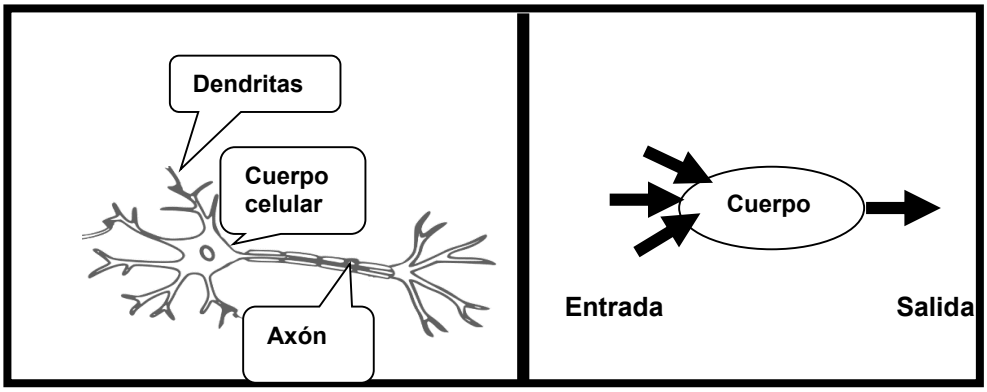
El propósito de cualquier sistema nervioso biológico es garantizar la supervivencia, se busca pues el control centralizado de las funciones vitales como son las de movimiento, circulación, respiración, metabolismo, reacción ante emergencias etc. Los animales superiores (incluido el ser humano) poseen memoria e imaginación y la capacidad de planificar para propiciar su supervivencia, utilizando sus dos sistemas de control en el cerebro: el de las funciones cognitivas y el de las reacciones emocionales (Kohonen, Hynninen, Kangas, & Laaksonen, 1995), las RNA pretenden emular las funciones cognitivas no así las emocionales que son altamente complejas y poco estudiadas desde una mirada de la neurociencia.

"Muchos modelos de redes neuronales, no reflejan la fisiología neuronal de bajo nivel, como la percepción del color, el ojo humano tiene millones de células para diferenciar colores y matices de colores, mientras que las llamadas RNAs son más bien modelos conductuales, o modelos de aprendizaje las cuales son capaces de predecir nuevas observaciones sobre la base de una data existente. Una RNA se asemeja al cerebro en dos aspectos: uno de ellos es que el conocimiento es adquirido a partir del ambiente, a través de un proceso de aprendizaje y otro es que las conexiones entre neuronas, conocidas como pesos (weights) sinápticos,

son utilizados para almacenar y transportar el conocimiento adquirido." (Hilera, 1995)

La unidad básica de una RNA es la neurona, cuyo modelo tiene unas entradas (dendritas), un cuerpo (soma) y una estructura de salida (axón). Las neuronas se conectan entre si vía la sinapsis, la eficiencia sináptica determina si la neurona conectada se activa o no. En la figura 11 se muestra la analogía muy simplificada entre una neurona biológica y una artificial.

Figura 11 *Analogía entre una neurona artificial y una biológica*



El proceso en una neurona artificial es muy sencillo: cada neurona recibe un número de entradas (ya sea datos originales o las salidas de otras neuronas en la red) cada entrada tiene un peso, con los cuales la neurona ejecuta un cálculo ponderando sus entradas, se activa si éste sobrepasa un valor denominado umbral y dispara una señal de salida la cual puede ser la salida de la red o ser la entrada de otra neurona. Este es el principio expresado por McCulloch y Pitts en 1943 quienes concibieron una célula neural como un elemento de un circuito que se dispara cuando la suma de sus entradas excede un valor de umbral. (Kohonen, Hynninen, Kangas, & Laaksonen, 1995)

Para que sea útil usar una red neuronal se deben cumplir algunas condiciones, como son:

El problema debe tener un cierto grado de dificultad, pues, por ejemplo, los programas que se escriben fácilmente como diagramas de flujo no son

aplicaciones ideales para ser tratados como redes neuronales. Otro criterio a considerar es la probabilidad de que cambie la lógica del programa, pues una de las características principales de las redes neuronales es su capacidad de aprender, es decir, si el algoritmo utilizado para resolver el problema es una regla invariable, no hay razón para utilizar una red neuronal. Finalmente, las redes neuronales a menudo no son adecuadas para problemas que requieren una solución rastreable, esto es, si es necesario explicar por qué la red dio tal respuesta. "*Una red neuronal puede ser muy útil para resolver el problema para el que fue entrenada, pero no puede explicar su razonamiento"* (Heaton, 2011)

Según (Kohonen, Hynninen, Kangas, & Laaksonen, 1995), las principales funciones que se implementan a través de una RNA son:

- funciones sensoriales (percepción artificial)
- funciones motoras (manipulación, locomoción).
- toma de decisiones (razonamiento, estimación, resolución de problemas)
- Comportamiento verbal (comprensión y producción del habla, lectura y escritura, respuesta a preguntas).

La implementación de esas funciones se usa para reconocimiento de patrones, los cuales pueden ser características, imágenes, textos, voz; o para clasificación, pronósticos, regresión, optimización, reconstrucción de imágenes y en general para minería de datos.

Mecanismo de aprendizaje

Una función importante de un sistema nervioso es definir las acciones que son parte del comportamiento y control del estado del organismo ante las circunstancias del medio ambiente. Para lo cual se apela al principio de recompensa y castigo, es por ello que se necesita cierta retro-propagación (*backpropagation*) de información y el significado y la calidad de las acciones están condicionados por los criterios de rendimiento, los cuales a su vez están condicionados por los resultados esperados. (Kohonen T. , 1988)

Para el logro de los propósitos para los que se crean las RNA, se requiere que esta sea capaz de adaptarse a los cambios, es decir, debe ser posible entrenarlas por medio de algoritmos. Este es el problema fundamental de la teoría de las redes neuronales, entrenar o enseñar a la red. Se considera que una red neuronal aprende cuando modifica sus pesos en respuesta a una información de entrada. El aprendizaje puede ser supervisado o no supervisado. (Matich, 2001)

Es supervisado cuando uno de los datos de cada entrada es la respuesta real de la misma, por ejemplo en el caso de morosidades de pago: una entrada i-ésima puede constar de n+1 campos, los primeros n representan las características del iésimo cliente, como edad, sexo, nivel educativo, monto de préstamo, lugar de residencia, historial crediticio entre otros y el campo n+1 puede ser su nivel de morosidad, una variable discreta que puede ser 0 en el caso no moroso hasta 5 que representaría un crédito fallido (irrecuperable). La red neuronal aprende de los primeros n campos y el campo n+1 le permite corroborar si su aprendizaje fue correcto o si tiene que introducir medidas correctivas (en los pesos)

Es no supervisado cuando no se dispone de una variable respuesta sino lo que se busca es que la red neuronal encuentre relaciones entre las entradas que pueden ser por distancias o similitudes, por ejemplo en el caso de lavado de activos: una entrada i-ésima puede constar de n campos que representan las características del iésimo cliente, como edad, sexo, nivel educativo, saldo de la cuenta, saldo promedio de los últimos seis meses, lugar de residencia, número de transacciones por mes, numero de agencias utilizadas por mes, monto promedio y desviación estándar de las transacciones por mes, entre otros. La red neuronal *aprende* de todos los datos y busca auto organizarse, es decir, busca agrupar por similitudes a los clientes para encontrar comportamientos atípicos. En general las redes neuronales no supervisadas se usan para clasificar, organizar o visualizar grandes conjuntos de datos, los cuales pueden estar muy bien delimitados o ser borrosos y carecer de límites claramente definidos.

Topología y mapas topológicos

Una topología es la estructura de la red neuronal, es un mapa que se ha simplificado de manera que sólo contiene la información esencial y todos los detalles innecesarios se han eliminado. Estos mapas carecen de escala, y la distancia y la dirección están sujetas a cambio y variación, pero la relación entre los puntos se mantiene, el nombre de mapa topológico se deriva de la topología, la rama de las matemáticas que estudia las propiedades de los objetos que no cambian cuando el objeto es deformado.

Componentes de una RNA

a) A nivel de la neurona (estructura)

Para caracterizar la unidad básica de procesamiento, es decir la neurona artificial, se debe establecer con precisión y sin ambigüedades lo siguiente : (Hilera, 1995)

- Estado de activación E_{it}= estado de la neurona i en el tiempo t, se debe definir E(i)= vector de N estados posibles de la neurona i.
- Umbral(i) = valor a partir del cual la neurona i se activa, es decir, si el valor obtenido es menor que el umbral, la neurona no se activa.
- Patrón de conectividad entre neuronas

  Determina la eficiencia de las sinapsis, es decir, los pesos.

  Vector de pesos = W_{it} = $(w_1,w_2,.....w_n)$ pesos de las entradas en el tiempo t
- Regla o función de entrada

  Determina cómo se combinan las entradas con los pesos

  Vector de entrada = X_t = $(x_1,x_2,.....x_n)$ valores de las entradas en el tiempo t

Función de entrada = $F(X_{it}, Wit)$ es una regla de comportamiento entre entradas y pesos, puede ser sumatoria, producto, máximo, mínimo etc.

- Regla o función de activación

  Determina el estado de la neurona i en el tiempo t+1, cuyo rango normalmente va de (0 a 1) donde cero significa inactiva y uno activa al máximo o también el rango puede ser de (–1 a 1).

  Función de activación = $G(F(x_t, W_t), E_t, Umbral_i)$ es una función que determina si la neurona se activa (E=1) o no (E=0) en función de la combinación de x, w y del umbral asignado a la neurona. Puede ser una función lineal, sigmoidea, tangente hiperbólica etc.

  Error = e_t es el error en relación a la variable respuesta (en el caso de aprendizaje supervisado), e = Z-E, es decir, el valor obtenido por el algoritmo versus la respuesta, si son iguales el error es cero.

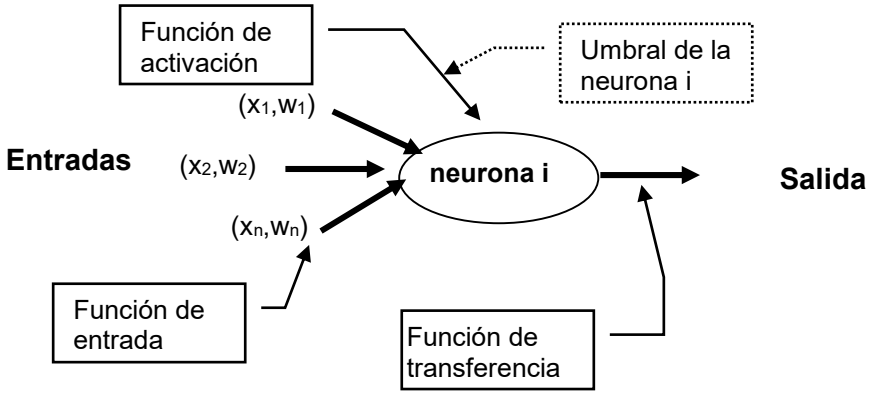
- Regla o función de transferencia

  Determina el valor que se transfiere a las neuronas vinculadas

  Función de salida o de transferencia = H(G)

La relación entre las funciones y la neurona artificial se presenta gráficamente, en la Figura 12

Figura 12 Relación entre las entradas y salida de una RNA



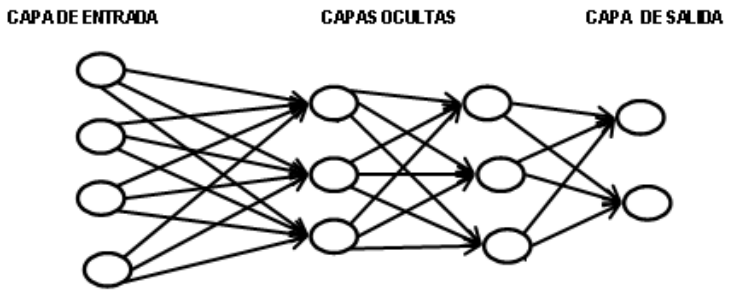
b) A nivel de la red (Topología)

La estructura esencial (topología) de una red neuronal consta de lo siguiente:

- Número de capas (capa de entrada, capa(s) oculta(s), capa de salida)
- Número de neuronas por capa
- Patrones de conexión (propagación hacia a delante, propagación hacia atrás, sistemas recurrentes)

En la figura 13 se muestra una RNA cuya topología consta de 4 capas de cuatro, tres, tres y dos neuronas cada una y con una conectividad hacia adelante.

Figura 13 Ejemplo de una topología de RNA



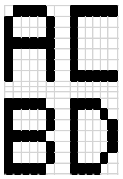
Tipos de redes neuronales

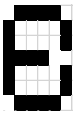
(Kohonen, Hynninen, Kangas, & Laaksonen, 1995) dice que "… *tradicionalmente, solo las siguientes tres categorías de modelos se consideran RNAs "puras": redes de transferencia de señal adaptativa, redes de transferencia de estado y redes de aprendizaje competitivo o autoorganizadas*."

- *Redes de transferencia de señal* en las cuales los valores de la señal de salida dependen únicamente de las señales de entrada. Se procesan los datos por cálculo algebraico. Los representantes típicos de este tipo son las redes de avance en capas, como el Perceptron, Madaline, la red backpropagation entre otras. Más adelante se presenta un ejemplo de Perceptron con entrada en R^2.
- *Redes de transferencia de estado*. Es un caso en el que en la fase de entrenamiento se presentan diversos estados y se entrena con retroalimentación, luego se presenta a la red un estado diferente, puede ser un estado similar a los usados para entrenamiento, pero distorsionado o incompleto, para que la red lo asocie con el estado más parecido, es decir con un *atractor*. Los representantes típicos son la red Hopfield y la máquina Boltzmann.

  Una red de Hopfield funciona como una memoria asociativa, primero se entrena con un conjunto de casos y luego cuando se presenta un dato nuevo, la red busca el patrón más parecido Es un proceso iterativo hasta que no haya cambios en la salida, en cuyo caso se dice que la red se ha estabilizado y ha encontrado el elemento del conjunto más parecido al dato nuevo.

  Por ejemplo, se entrena a la red con los patrones.

  Y se presenta un nuevo patrón:  para que lo "asocie con su conocimiento"

  Este tipo de red no es el tema principal del presente estudio por lo que no se detalla, pero su desarrollo en este ejemplo puede verse en el anexo 3

- *Redes de aprendizaje competitivo o autoorganizadas.* En las redes de aprendizaje competitivo o autoorganizadas, las neuronas de salida reciben información de entrada idéntica y mediante interacciones laterales compiten en sus actividades, las diversas neuronas se convierten en decodificadores o detectores específicos de sus respectivos dominios de señal en el espacio de entrada. Estos decodificadores forman una red en un orden con sentido (Kohonen, Hynninen, Kangas, & Laaksonen, 1995) Cada neurona se conecta con otras de su entorno de manera que produce una excitación en las más próximas y una inhibición en las más alejadas, formando clústeres tales que las entradas más semejantes “se agrupan” y la neurona que las agrupa se le denomina la *ganadora*.

Ejemplo del uso de una red neuronal supervisada

Se presenta un ejemplo de red de transferencia de señal, en este caso un Perceptrón con una capa de entrada y una de salida.

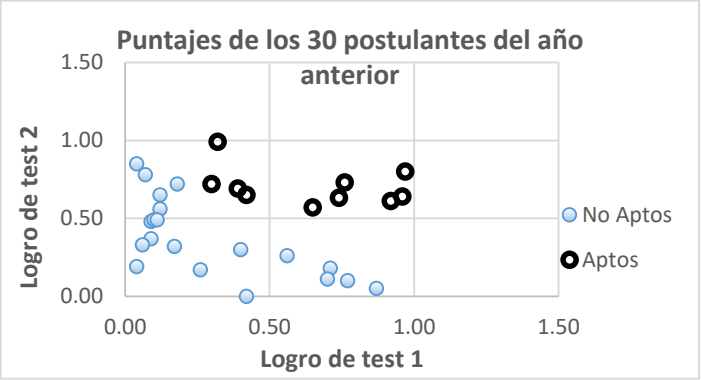
El problema a resolver es el siguiente: El año anterior se hizo una evaluación de postulantes y se les sometió a ocho test, pero los especialistas dicen que con solo los dos primeros se puede determinar si el postulante es apto o no para el trabajo. Se tiene un conjunto de datos compuesto por 3 variables (ver Tabla 3), dos que representan los porcentajes de logro en los dos primeros test y una variable respuesta que representa el diagnóstico luego de todo el proceso de evaluación: apto (1) o no apto (0).

Tabla 3 Puntajes obtenidos por los 30 postulantes del año anterior

t	Logro Test 1	Logro Test 2	Resultado real
1	39%	69%	1
2	71%	18%	0
3	96%	64%	1
4	42%	65%	1
5	76%	73%	1
6	92%	61%	1
7	17%	32%	0
8	12%	65%	0
9	9%	48%	0
10	56%	26%	0
11	30%	72%	1
12	77%	10%	0
13	40%	30%	0
14	42%	0%	0
15	26%	17%	0
16	65%	57%	1
17	18%	72%	0
18	74%	63%	1
19	32%	99%	1
20	9%	37%	0
21	4%	85%	0
22	87%	5%	0
23	97%	80%	1
24	89%	30%	1
25	70%	11%	0
26	19%	92%	1
27	68%	40%	1
28	73%	86%	1
29	12%	56%	0
30	10%	49%	0

En la figura 14 se muestran los puntajes en un plano

Figura 14 Puntajes de los 30 postulantes del año anterior



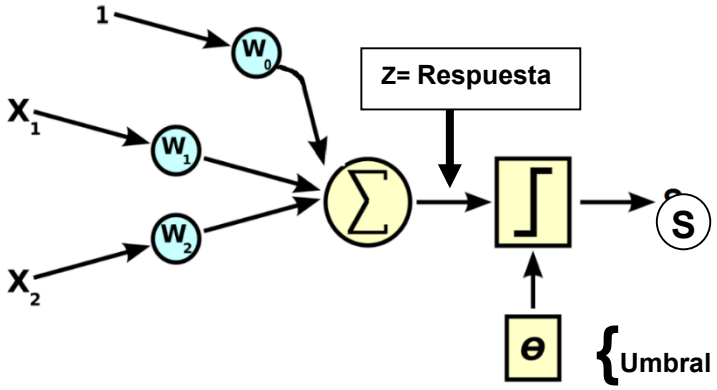
Para validar el rendimiento de la red neuronal se ha separado cinco postulantes, tal que se conoce sus resultados, pero no se van a usar para entrenar a la red, y luego se determinará si la red puede clasificarlos correctamente como aptos o no aptos. Los valores obtenidos por los cinco postulantes de prueba se muestran en la tabla 4

Tabla 4 Puntajes obtenidos por los 5 postulantes de prueba

Logro Test 1	Logro Test 2	Final
20%	49%	0
7%	70%	0
58%	61%	1
14%	19%	0
10%	33%	0

La topología de la red utilizada se muestra en la figura 15

Figura 15 Topología de la red



Donde:

X1 = % de logro en el test 1

X2 = % de logro en el test 2

La tercera neurona de entrada es una constante igual a “1” para todos los postulantes para que represente la intersección con el eje vertical.

Wi = peso de la i-ésima entrada (inicialmente cero)

Z= variable respuesta, es decir, Apto o No apto.

θ = umbral

S = neurona de salida: 1 si es apto; 0 si no lo es.

A continuación, se describen las funciones de la RNA

Función de entrada:

- Vector de entrada para t =1 X_t = (x0, x1, x2) = (1, 39%, 69%)
- Vector de pesos para t =1 Wt = (w_1,w_2,.....w_n) = (0,0,0)
- Variable respuesta Z_t en este caso para t =1 z = 1, es decir, “Apto”

- Función de entrada = $F(X_{it}, W_{it})$ en este caso es una sumatoria, para t= 1:

$$\sum x_i w_i = x_0 w_0 + x_1 w_1 + x_2 w_2 \quad = 1*0 + 0.39*0 + 0.69*0 = 0$$

Función de activación:

- La función utilizada es la función escalón: si F es menor que el umbral, no se activa, es decir S=0 en caso contrario S=1
- Umbral(i) = 0.4, como $F(x_{11},x_{21}) = 0<0.4$ la neurona no se activa
- Valor de salida en el tiempo t = $S_1 =0$
- Error en t=1 es e= Z-E = 1-0 = 1
- Constante de corrección o tasa de aprendizaje = 0.1
- Función de corrección = $C_1 = e_1$ * corrección = 1*0.1=0.1

Función de transferencia

- Los pesos para t+1 se calculan en base al error y a la función de corrección $w_{t+1} = w_{it} + x_{it}*C_t$

  $W_{02} =0 +1*0.1 = 0.1$ peso para la neurona de entrada cero en la iteración 2

  $W_{12} =0 +0.39*0.1 = 0.039$

  $W_{22} =0 +0.69*0.1 = 0.069$

- En la iteración 2 los pesos son (0.1, 0.039, 0.069) y se toma las variables de entrada de la fila 2 es decir x2=(1, 0.71, 0.18) y la variable respuesta es cero, es decir, ese postulante con ese puntaje no es apto.
- Se repite el proceso hasta completar los datos o que los pesos se repitan (los pesos se repiten si no hay errores)

- En este caso, los pesos se repiten a partir de la iteración 28, (ver Anexo 2), en la Tabla 5 se muestran los pesos finales

Tabla 5 Pesos finales

W0	W1	W2
0.2	0.172	0.216

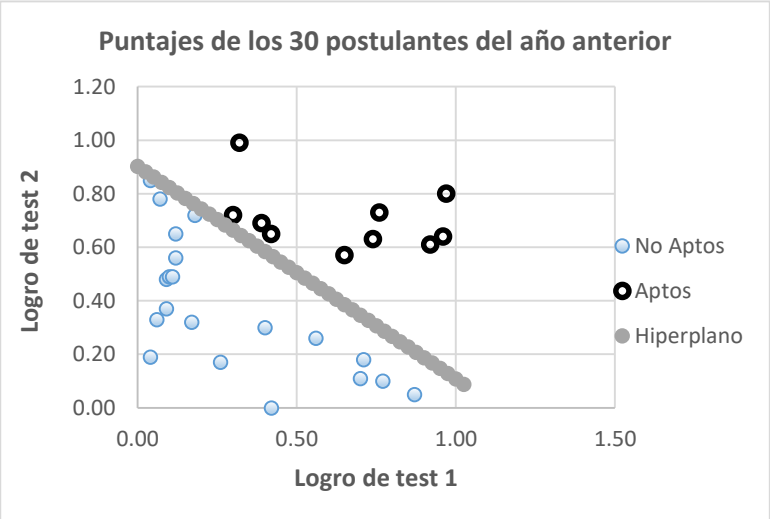
Los pesos representan los parámetros del hiperplano que separa el conjunto de los aptos y los no aptos, como en este caso se considera solo dos variables se ha obtenido tres parámetros (tres pesos)

$w_1x_1+w_2x_2 \geq w_0 \rightarrow 0.172x_1+0.216x_2 \geq 0.200$

La ecuación del hiperplano es: $X_2 = \frac{-W_1}{W_2} X_1 + \frac{-W_0}{W_2}$

En la figura 16 se observan los treinta datos y el hiperplano que separa a los aptos de los no aptos

Figura 16 Hiperplano que separa a los aptos de los no aptos



Para validar el proceso se evalúa a los cinco postulantes de prueba, dado que se conoce si fueron considerados aptos o no. Se determina a cuál lado del hiperplano pertenecen, al de los aptos o al de los no aptos

$H= 0.172x_1+0.216x_2 \geq 0.200$

Donde

x_1 = Logro en el test 1 y

x_2 = Logro en el test 2

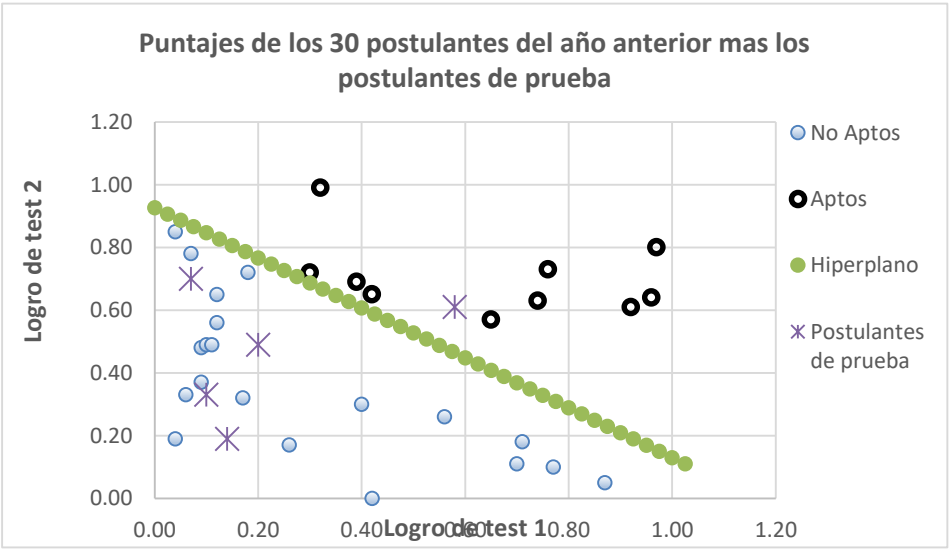
En la Tabla 6 se muestran los cálculos

Tabla 6 Evaluación de los 5 postulantes de prueba

Logro Test 1	Logro Test 2	0.172x1+0.216x2	≥ 0.200?	Resultado real
20%	49%	0.1402	No Apto	0
7%	70%	0.1632	No Apto	0
58%	61%	0.2315	Apto	1
14%	19%	0.0651	No Apto	0
10%	33%	0.0885	No Apto	0

La figura 17 muestra su ubicación en el plano

Figura 17 Ubicación de los cinco postulantes de prueba



Interpretación

En los cinco casos la red neuronal acertó en la adjudicación de la condición de apto o no apto. Estos aciertos son circunstanciales a este caso, podría tenerse otras situaciones en las que se producen errores.

En el próximo concurso de selección se puede evaluar a los nuevos postulantes solo en el test 1 y el test 2 y aplicando los pesos finales obtenidos por la red neuronal se podrá determinar si son aptos o no para el trabajo.

No hay un modelo explicativo de porqué esos postulantes se consideran aptos o no aptos.

Desventajas de las redes neuronales en el contexto de optimización

Las redes neuronales generalmente tienen un componente aleatorio, y al ser métodos computacionales tienen un cierto número de iteraciones a diferencia de los métodos exactos, no hay un "punto de convergencia" ni una garantía de óptimo, se podría pensar que con más iteraciones se aproxima más a la solución real, lo cual no es cierto pues al igual que en los procesos mentales humanos un exceso de información no garantiza soluciones óptimas.

Como son métodos computacionales para resolver problemas de casi cualquier área, se requiere de un equipo multidisciplinario conformado por especialistas que conocen el tema y también por especialistas en ciencias de la computación, pero no es un proceso tan sencillo como el de un sastre que hace un terno a la medida, sino que ambos grupos deben conocer el trabajo del otro, lo que implica que la formación profesional de cada grupo debe tener un nivel de rigurosidad importante en lo que se refiere a matemáticas, computación y análisis de datos para poder tomar decisiones.

Una tercera desventaja es que las redes neuronales artificiales no generan un modelo ni descriptivo ni explicativo de la situación en estudio. Solo es la respuesta al entrenamiento con un conjunto de datos de entrada, pero, parafraseando el dicho atribuido a Galileo "eppur si muove".

A continuación, se presenta una red neuronal que emula los procesos mentales humanos de autoorganización, es decir ni siquiera hay un supervisor que orienta hacia un aprendizaje "correcto" sino que es una red no supervisada, una red de Kohonen, la cual es el tema principal del presente estudio.

2.4.4. *Redes de Kohonen*

El desarrollo de las redes de Kohonen como modelo neural fue motivado por una característica del cerebro humano, el cual tiene la capacidad de formar mapas de características (mapas topológicos) de las informaciones recibidas desde el exterior, organizando las informaciones en muchas regiones, de tal manera que diferentes entradas sensoriales pueden llevar a una misma región

(como por ejemplo el condicionamiento del marketing: una persona compra en un supermercado un conjunto de productos frecuentemente con un patrón de compra similar al de otras personas, por lo tanto el supermercado lo asigna a un grupo “x” y así le ofrece paquetes que contienen los productos que compra además de otros que compran las otras personas del grupo “x”).

Los mapas autoorganizados (SOM por las siglas en inglés self organizing map) se utilizan tanto para agrupar datos (en clústeres) como para reducir la dimensionalidad de los datos. Se inspiran en las asignaciones sensoriales y motoras en el cerebro de los mamíferos, que también parecen organizar automáticamente la información topológicamente. (https://www.mathworks.com/help/deeplearning/ref/selforgmap.html)

Una red de Kohonen es una red de dos capas, una de entrada y la otra de salida, las conexiones entre la capa de entrada y la de salida son de tipo hacia adelante (feedforward) y existe conexiones laterales entre las neuronas de salida.

Una red de Kohonen es una red competitiva con la habilidad de realizar mapeos que preservan la topología entre los espacios de entrada y de salida. Su objetivo es descubrir patrones significativos o característicos de los datos de entrada.

Tiene dos variantes, denominadas LVQ (Learning Vector Quantization) y TPM (Topology-Preserving Map – llamada también SOM - self organizating map). Ambas forman mapas topológicos para establecer características comunes entre las informaciones de entrada.

Existen muchas versiones del SOM, pero su esencia es muy simple y efectiva, el SOM define una "red elástica" de puntos (parámetros, referencias o vectores de códigos (codebook vectors)) que se ajustan al espacio de la señal de entrada para aproximar su función de densidad de una manera ordenada. Las principales aplicaciones del SOM se encuentran, por lo tanto, en la visualización de datos complejos en visualización bidimensional y en la creación de abstracciones como en muchas técnicas de agrupación (clústerización).

Se define una asignación desde el espacio de datos de entrada R^n a un arreglo de nodos bidimensional (la utilidad de que sea bidimensional es por razones gráficas) a cada nodo i-ésimo se le asocia un vector m_i perteneciente a R^n. Entonces, a un vector de entrada X perteneciente a R^n se le compara con cada m_i, (i= 1….k) y la mejor coincidencia m_g se define como "respuesta ganadora", como se muestra en la ecuación (1)

$$||X-m_g|| = \min_i \{||X-m_i||\} \quad (1)$$

Por lo que dicha entrada X se asigna a m_g y en consecuencia al nodo de salida al que está asociada. En otras palabras, el SOM es una "proyección no lineal" de los datos de entrada de dimensión n en una gráfica bidimensional (Kohonen, Hynninen, Kangas, & Laaksonen, 1995)

De todas las neuronas de salida solo una es la ganadora, Durante el aprendizaje, los nodos que están topológicamente cerca del arreglo de salida hasta una cierta distancia se activarán entre sí para aprender de la misma entrada. Se puede considerar que los valores de m_i se pueden encontrar como límites de convergencia del proceso de aprendizaje siguiente, es decir el de (t+1), por lo que los valores iniciales de m_i (0) pueden ser arbitrarios o aleatorios, pues en las siguientes iteraciones se recalcularan. (Kohonen, Hynninen, Kangas, & Laaksonen, 1995)

Como se observa en la ecuación (2) los m_i asociados a la capa de salida se recalculan

$$m_i(t+1) = m_i(t) + h_{gi}(t)\,[x(t) - m_i(t)] \quad (2)$$

Donde t es un número entero (la coordenada tiempo) y h_{gi} (t) es el llamado *kernel del vecindario*, la cual es una función definida sobre los puntos de la red. Por lo general, $h_{gi} = h(||r_g - r_i||, t)$ donde r_g y r_i, ambos en R^2, son los vectores radiales de los nodos g e I, respectivamente, en la capa de salida. De tal manera que cuando se incrementa $||r_g - r_i||$, es decir cuando, r_i está “lejos” de r_g, entonces h_{gi} tiende a cero. En palabras sería “si el punto r_i está lejos de la *ganadora* r_g, la probabilidad de que pertenezca al vecindario de esta es muy pequeña” El ancho y la forma promedio de h_{gi} define la "rigidez" de la "superficie elástica" que debe ajustarse a los puntos de datos. (Kohonen,

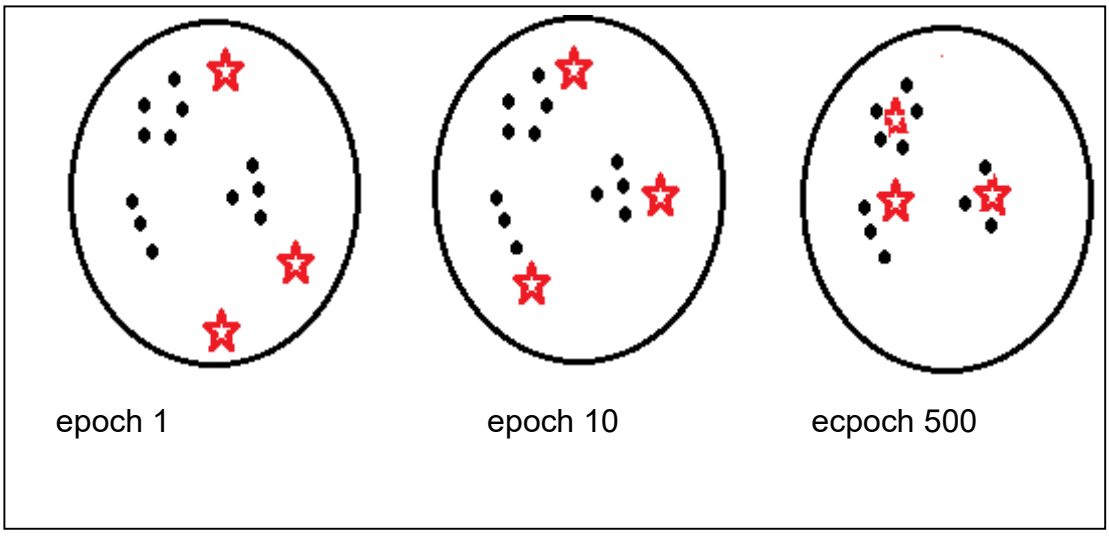
Hynninen, Kangas, & Laaksonen, 1995) El procedimiento se repite hasta que se cumpla la condición de parada.

Hay dos conceptos de iteración que hay que precisar:

- Iteración del lote de datos. (batch). - el índice j varía de 1 a N, donde N es el cardinal del conjunto de entrada, es decir se procesa cada elemento del conjunto de entrada.
- Iteración de corridas (Epoch). - una iteración es el proceso desde que se lee un vector de entrada hasta que se actualizan los pesos de las neuronas y se prepara para un nuevo input. El índice i varía de 1 a un número arbitrario que se puede fijar inicialmente y/o hacerlo variar en función de la precisión que se quiera del modelo.

En síntesis, el algoritmo es un proceso iterativo tal que en cada iteración j (con cada elemento del conjunto de entrada) se produce una "ganadora" la misma que se actualiza en función de la data existente hasta agrupar a todos los elementos del conjunto de entrada en "vecindarios" alrededor de las m_i. Una descripción gráfica en R^2 se presenta en la Figura 18

Figura 18 Reubicación de las neuronas ganadoras



Fuente Kohonen et al (1995)

Algoritmo SOM

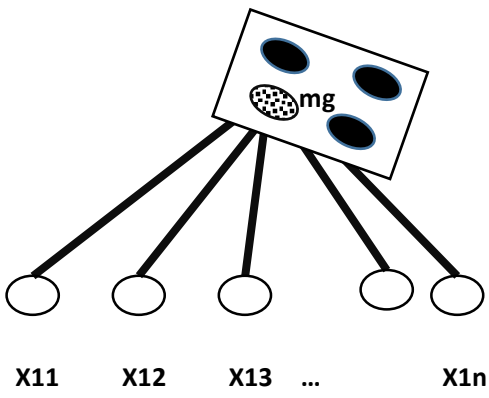
1. Inicializar las m_i (puede ser aleatoriamente)
2. Seleccionar un vector de entrada X
3. Comparar X con las mi usando una métrica cualquiera en la ecuación (1)
4. Determinar la mi ganadora denominada mg tal que da la mínima distancia
5. Actualizar la ganadora tal que se acerque más a los vectores que pertenecen a su vecindario, usando la ecuación (2)
6. Repetir de 2 a 5 hasta completar todos los X que pertenecen al conjunto de entrada.
7. Ajustar el parámetro de la tasa de aprendizaje, la misma que es una función decreciente.
8. Repetir hasta que se cumpla la condición de parada (epoch fijada)

Sea X1 = (x11, x12, …x1n) en R^n un vector de datos de entrada, la capa de salida consta de un arreglo de 4 nodos bidimensionales, a X1 se le asocia un m_i (i = 1,2..4) los cuales al inicio pueden ser generados aleatoriamente. Se compara X1 con todas la m_i en cualquier métrica; en aplicaciones prácticas se usa la de las distancias euclidianas $||X1 - m_i||$ entonces se considera que la “ganadora" m_g es la que cumple:

$$||X1-m_g|| = \min_i \{||X1-m_i||\} \quad (1)$$

En la figura 19 se presenta la iteración 1

Figura 19 comparación del primer vector con las cuatro m_i iniciales.



Cuatro elementos m_i cada uno de dimensión n

Primer elemento del conjunto de entrada de dimensión n

En este caso g=1, m1 es el elemento que está más próximo a X1, entonces se dice que X1 pertenece al vecindario de m1.

Se actualiza a la ganadora mg de tal manera que mg se acerque más a X1, junto con los demás vecinos de la ganadora, es decir se reubica la posición de mg y se modifica el radio del vecindario por medio de la ecuación (2)

Se repite para todos los datos de entrada, Esto constituye una corrida (*epoch*), se realizan tantas *epochs* como se considere necesario.

Si se supone que los vectores de entrada son de dimensión 3, por ejemplo,

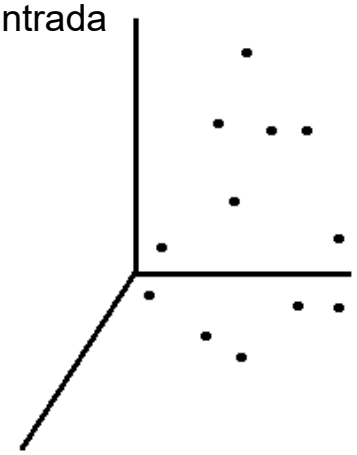
Eje 1 = rango de edades

Eje 2 = nivel de sueldo

Eje 3 = número de hijos

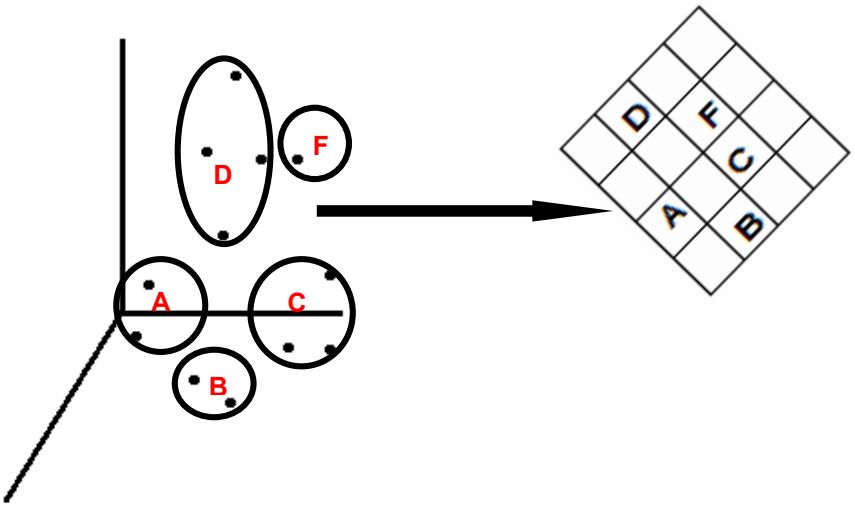
En la figura 20 se muestra los datos y en la figura 21 los clústeres formados según sus características

Figura 20 Datos de entrada



El algoritmo SOM permite reducir el espacio de características a dos dimensiones. Ver Figura 21

Figura 21 Datos en R^3 agrupados en cinco clústeres mapeados a R^2



Tal vez al pasar de R3 a R2 no se aprecie la diferencia visual, pero, si los datos de entrada constan de más de 3 dimensiones la reducción de características es importante.

Ejemplo del uso de una red SOM

Se presenta un ejemplo sencillo con 20 nodos para mostrar el algoritmo, pues en el caso desarrollado en el presente estudio se tiene más de doscientos nodos.

Por ejemplo, en una región se tiene 20 pueblos en zonas rurales, sus coordenadas geográficas, y existe la capacidad para instalar cuatro postas médicas, Se quiere ubicar las postas de modo que se atienda a los 20 pueblos tal que los pobladores se desplacen mínimamente para llegar a la Posta médica asignada.

Aplicación del algoritmo SOM con Tasa de aprendizaje = taprend = 0.7 y Número de corridas (epoch) = 1000

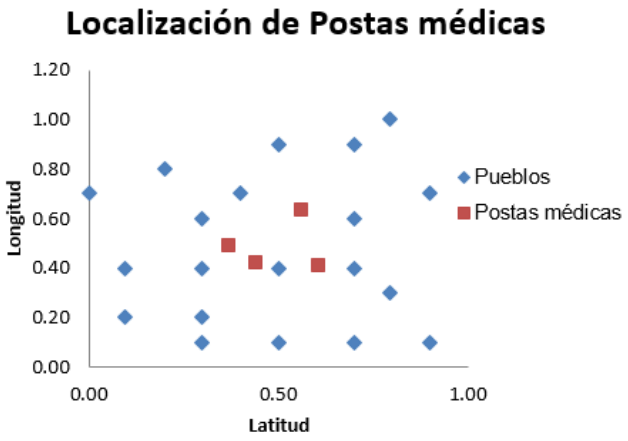
1. **Inicializar las m_i (puede ser aleatoriamente)**

   Se inicializa aleatoriamente la ubicación de las cuatro postas médicas (mj), lo cual se muestra en la Tabla 7 y se representa en la Figura 22.

Tabla 7 Coordenadas de las neuronas (postas médicas) mi en epoch 1

Neurona	x	y
m1	0.608	0.410
m2	0.443	0.422
m3	0.562	0.632
m4	0.370	0.487

Figura 22 Los 20 pueblos y las postas médicas iniciales



2. **Seleccionar un vector de entrada X (la ubicación de cada pueblo)**

   Se lee secuencialmente la data, es decir las coordenadas de los veinte pueblos.

3. **Comparar X con las m_i usando una métrica cualquiera en la ecuación (1)**

   En este caso se usa la distancia euclidiana

   Para el pueblo 1 = X1 = [0.5 0.1]

   La distancia a m1 es $||X1-m_1|| = ((0.5 - 0.608)^2 + (0.1 – 0.401)^2)^{1/2} = 0.328$

   Análogamente calcular para m2, m3 y m4, los resultados se muestran en la Tabla 8.

Tabla 8 Distancia de los 20 pueblos a cada mi en la epoch 1

Pueblo	x	y	distancia. a m1	distancia. a m2	distancia. a m3	distancia. a m4	min	Neurona ganadora
1	0.5	0.1	0.328	**0.327**	0.54	0.41	**0.327**	m2
2	0.7	0.4	0.09	0.26	0.27	0.34	0.09	m1
3	0.7	0.9	0.50	0.54	0.30	0.53	0.30	m3
..	…							…
20	0.3	0.4	0.31	0.14	0.35	0.11	0.11	m4

4. Determinar la m_j ganadora denominada m_g tal que proporciona la mínima distancia

En la Tabla 9 se observa que para X1=(0.5 0.1) es decir el pueblo 1, la ganadora es m_2 En la tabla 9 se presentan los clústeres de las neuronas ganadoras en epoch 1

Tabla 9 Coordenadas de los pueblos asignados a las postas médicas en epoch 1

Pueblos que pertenecen al clúster de la Neurona1		Pueblos que pertenecen al clúster de la Neurona2		Pueblos que pertenecen al clúster de la Neurona3		Pueblos que pertenecen al clúster de la Neurona4	
0.7	0.4	0.5	0.1	0.7	0.9	0	0.7
0.8	0.3	0.3	0.2	0.8	1	0.3	0.6
0.9	0.1	0.5	0.4	0.9	0.7	0.1	0.4
0.7	0.1	0.3	0.1	0.7	0.6	0.1	0.2
				0.4	0.7	0.2	0.8
				0.5	0.9	0.3	0.4

Las distancias de cada pueblo a su respectiva posta médica se presentan en la Tabla 10.

Tabla 10 Distancia de cada neurona (pueblo) a su neurona ganadora en epoch 1

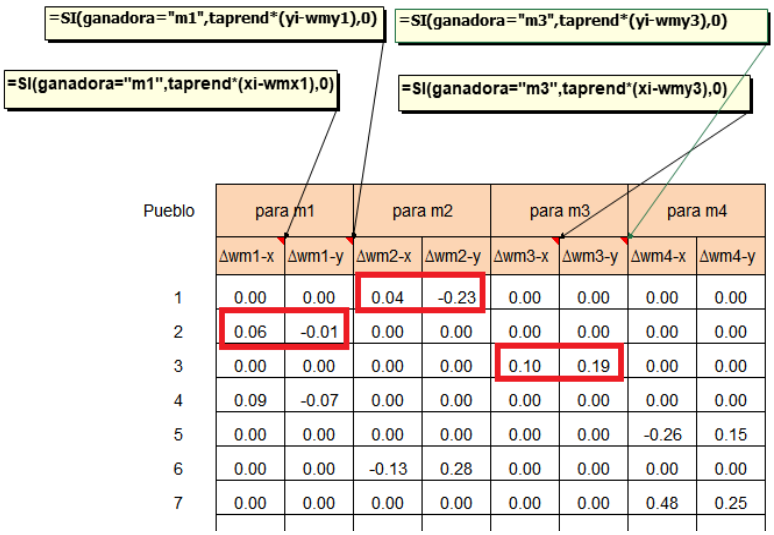
	Posta 1	Posta 2	Posta 3	Posta 4
	0.092	0.327	0.301	0.427
	0.221	0.264	0.438	0.133
	0.426	0.061	0.345	0.284
	0.323	0.353	0.142	0.394
			0.175	0.356
			0.275	0.111
Promedio	0.265	0.251	0.279	0.284

5. **Actualizar la ganadora tal que se acerque más a los vectores que pertenecen a su vecindario, usando la ecuación (2)**

   Se toma la primera neurona de entrada (pueblo1) y se calcula su kernel del vecindario, es decir se calcula la distancia de cada coordenada por la tasa de aprendizaje y luego se modifica cada coordenada de las postas médicas. Repetir de 2 a 5 hasta completar todos los X que pertenecen al conjunto de entrada.

   En la figura 23 se presentan los cálculos para actualizar los *kernel del vecindario* de las coordenadas de las postas médicas

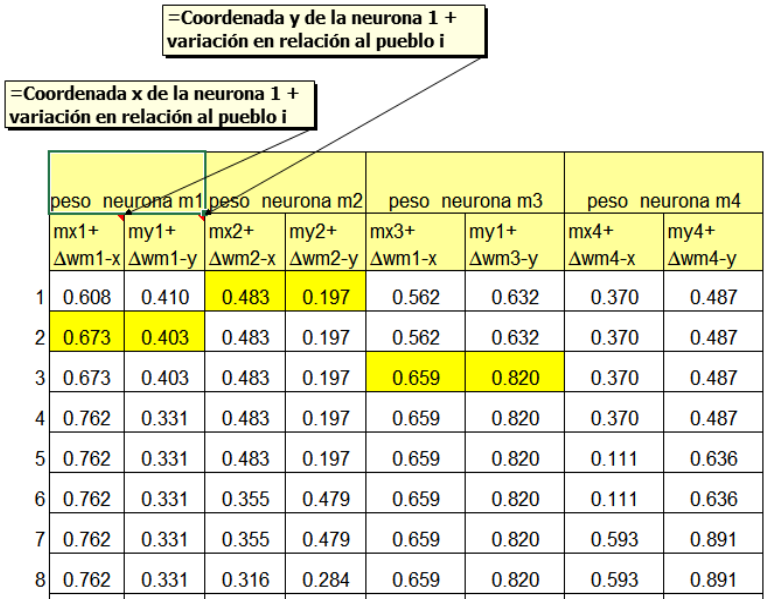
Figura 23 Cálculo de la variación parcial (kernel del vecindario) de cada neurona ganadora



Pueblo	para m1		para m2		para m3		para m4	
	Δwm1-x	Δwm1-y	Δwm2-x	Δwm2-y	Δwm3-x	Δwm3-y	Δwm4-x	Δwm4-y
1	0.00	0.00	0.04	-0.23	0.00	0.00	0.00	0.00
2	0.06	-0.01	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.10	0.19	0.00	0.00
4	0.09	-0.07	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	-0.26	0.15
6	0.00	0.00	-0.13	0.28	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.48	0.25

Como se observa en la Tabla 9, el pueblo 1 pertenece al clúster de la posta 2, por lo tanto, en su fila correspondiente solo se modifica la variación de la neurona 2 (ver recuadro en la figura) análogamente el pueblo 2 pertenece a la posta 1 y así sucesivamente. Con la variación calculada se aplica la ecuación (2), lo cual se muestra en la Figura 24

Figura 24 Cálculo de la variación de cada neurona ganadora en epoch 1



	peso neurona m1		peso neurona m2		peso neurona m3		peso neurona m4	
	mx1+ Δwm1-x	my1+ Δwm1-y	mx2+ Δwm2-x	my2+ Δwm2-y	mx3+ Δwm1-x	my1+ Δwm3-y	mx4+ Δwm4-x	my4+ Δwm4-y
1	0.608	0.410	0.483	0.197	0.562	0.632	0.370	0.487
2	0.673	0.403	0.483	0.197	0.562	0.632	0.370	0.487
3	0.673	0.403	0.483	0.197	0.659	0.820	0.370	0.487
4	0.762	0.331	0.483	0.197	0.659	0.820	0.370	0.487
5	0.762	0.331	0.483	0.197	0.659	0.820	0.111	0.636
6	0.762	0.331	0.355	0.479	0.659	0.820	0.111	0.636
7	0.762	0.331	0.355	0.479	0.659	0.820	0.593	0.891
8	0.762	0.331	0.316	0.284	0.659	0.820	0.593	0.891

Se observa que:

Como el pueblo 1 no pertenece al clúster de la posta 1, en la primera iteración esta no se ha modificado sigue siendo (0.608 0.410), la ubicación de la neurona 2 si se ha modificado , inicialmente era (0.443 0.422) luego de la actualización es (0.483 0.197).

Cada pueblo modifica la ubicación de su neurona ganadora acercándola. Luego de iterar para los 20 pueblos, las coordenadas de las neuronas postas se muestran en la tabla 11

Tabla 11 Coordenadas de las neuronas (postas médicas) mj luego de la actualización en el epoch 1

Neurona	x	y
m1	0.719	0.169
m2	0.309	0.336
m3	0.500	0.835
m4	0.234	0.632

Y las distancias de los pueblos correspondientes a sus clústeres se presentan en la Tabla 12.

Tabla 12 Distancia de cada neurona (pueblo) a su neurona ganadora en epoch 1 después de la actualización

	Posta 1	Posta 2	Posta 3	Posta 4
	0.231	0.304	0.210	0.243
	0.154	0.136	0.342	0.074
	0.194	0.201	0.422	0.268
	0.072	0.236	0.308	0.453
			0.168	0.171
			0.065	0.242
Promedio	0.163	0.219	0.253	0.242

Comparando con las distancias promedio de la Tabla 4 se obtiene una reducción, ver tabla 13

Tabla 13 Diferencias de distancia de los pueblos a la posta luego de epoch1

	Posta 1	Posta 2	Posta 3	Posta 4
Promedio antes	0.265	0.251	0.279	0.284
Promedio después	0.163	0.219	0.253	0.242
% de variación	38%	13%	9%	15%

Como se observa, en la primera iteración se ha logrado disminuir la distancia de los pueblos a sus respectivas postas.

6. **Ajustar el parámetro de la tasa de aprendizaje, la misma que es una función decreciente.**

7. **Repetir hasta que se cumpla la condición de parada (epoch =1000)**

Luego de 1000 iteraciones se consiguió una ubicación para las postas la cual se muestra en la tabla 14

Tabla 14 Coordenadas de las neuronas (postas médicas) mj luego de la actualización en el epoch 1000

Neurona	x	y
m1	0.291	0.188
m2	0.716	0.243
m3	0.216	0.625
m4	0.713	0.813

Y las distancias de los pueblos correspondientes a sus clústeres se presentan en la Tabla 15.

Tabla 15 Distancia de cada neurona (pueblo) a su neurona ganadora en epoch 1000

	Posta 1	Posta 2	Posta 3	Posta 4
	0.227	0.158	0.229	0.088
	0.014	0.102	0.087	0.206
	0.192	0.267	0.199	0.219
	0.089	0.233	0.253	0.214
	0.212	0.144	0.176	0.230
Promedio	0.147	0.181	0.189	0.191

Comparando con las distancias promedio de la Tabla 10 se obtiene una reducción, (ver tabla 16)

Tabla 16 Diferencias de distancia de los pueblos a la posta luego de epoch1000

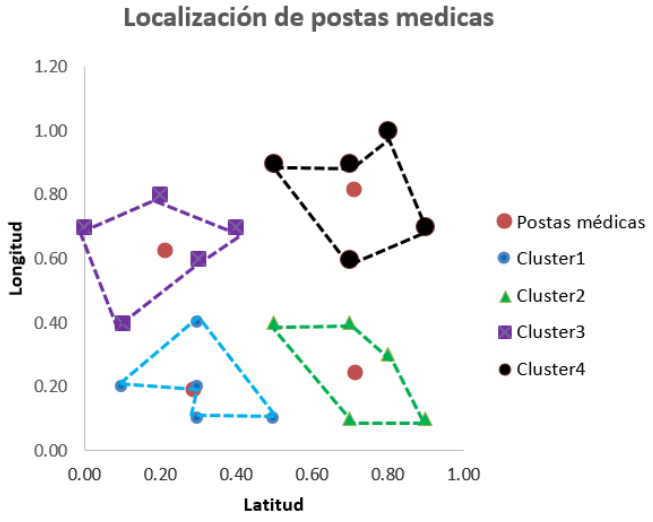
	Posta 1	Posta 2	Posta 3	Posta 4
Promedio inicial	0.265	0.251	0.279	0.284
Promedio final	0.147	0.181	0.189	0.191
% de variación	45%	28%	32%	33%

Como se observa,en promedio la distancia entre los pueblos y las postas se ha reducido considerablemente.

Una característica de los métodos computacionales es que no se puede garantizar óptimos. Podría haberse encontrado distancias menores con más iteraciones o con puntos iniciales diferentes.

La asignación final se muestra en la Figura 25

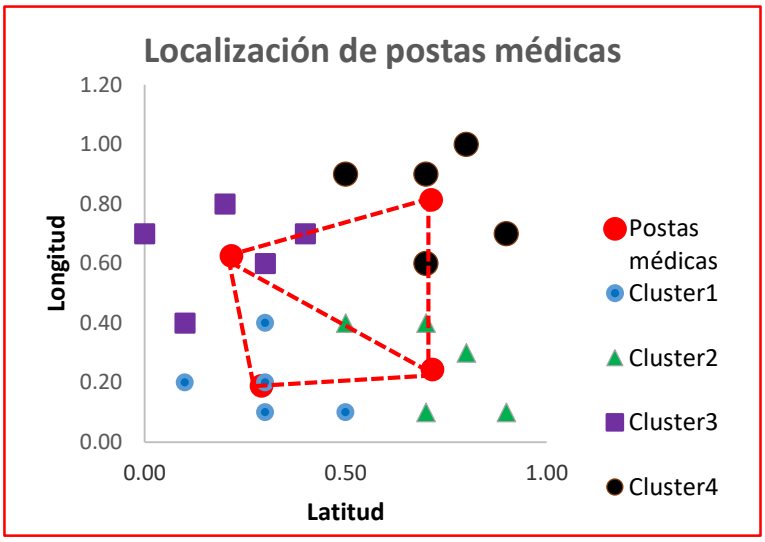
Figura 25 Asignación final de los pueblos a las postas médicas epoch 1000



8. **construir los conjuntos de Voronoi:**

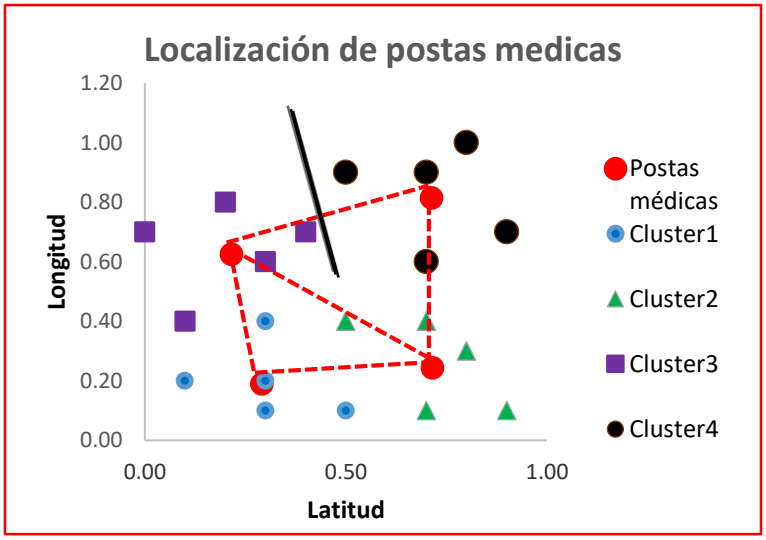
Primero se unen las neuronas resultantes, en este caso las postas médicas, formando un grafo de cuatro nodos y cinco aristas como se muestra en la Figura 26.

Figura 26 Unir neuronas ganadoras



En segundo lugar, se toma el punto medio de cada arista y se traza su recta bisectriz, ver figura 27 el trazo de una bisectriz

Figura 27 Bisectriz entre las postas 3 y 4



Luego se trazan las otras bisectrices como se observa en la Figura 28 hasta completar las particiones (Figura 29).

Figura 28 Bisectriz entre las postas 3 y 1

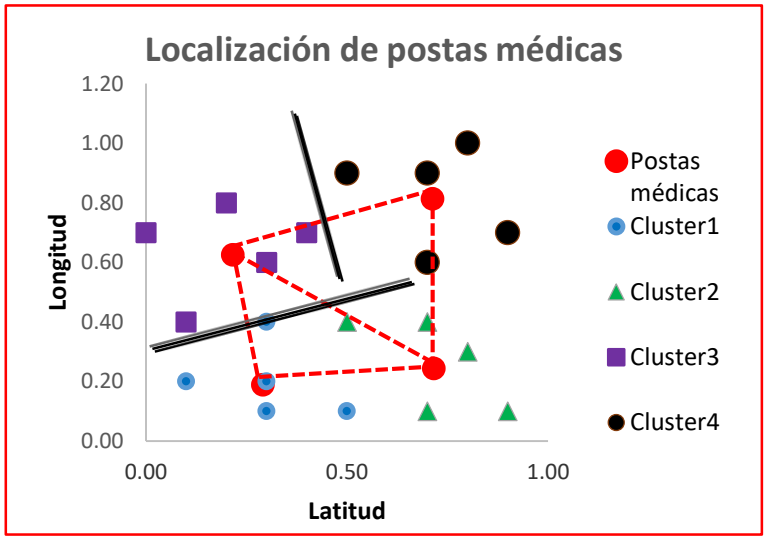
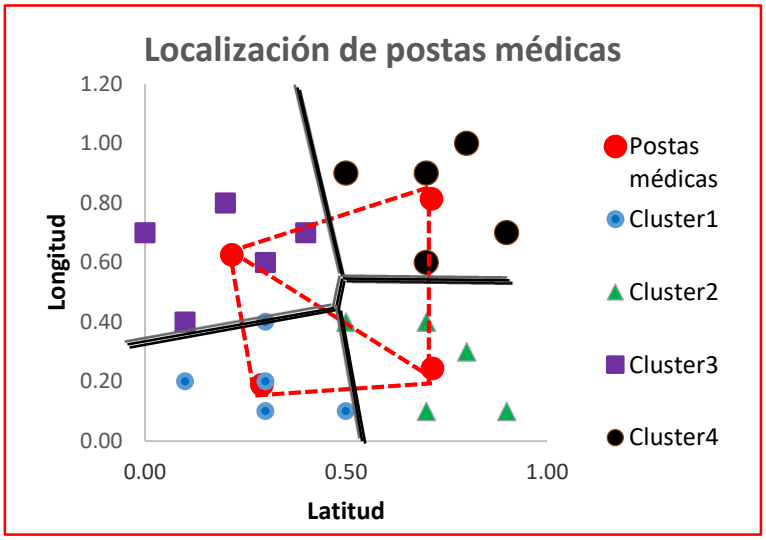


Figura 29 Diagrama de Voronoi



Como se observa en la Figura 29 cada conjunto de Voronoi (Vi) contiene a los elementos del clúster correspondiente; al incluir el casco convexo se incluye a todos los puntos y se consigue una partición.

En el presente estudio se hace el mismo tratamiento a las esquinas del distrito de Bellavista que debido a la cantidad de puntos pi no se puede realizar manualmente sino mediante un programa computacional.

En este ejemplo se ha considerado cuatro clústeres, pero este número puede ser variable indicándolo como un parámetro de entrada al programa computacional.

2.4.5. Mapas autoorganizados en Matlab

Los comandos usados para crear y usar mapas autoorganizados en MatLab son:

- selforgmap.- Crea una red de kohonen (Self-Organizing Map)
- train.- Entrena a una red simple (no para Deep learning)
- trainParam.epochs.- indica el número de iteraciones (epochs)
- net.IW.- guarda los pesos (weights) de las neuronas ganadoras de la red (net)
- sim(p).- Simula la red neuronal para identificar a cuál clúster pertenece el punto p
- find.- Proporciona los índices de las neuronas a cuyo vecindario pertence cada punto de entrada
-

Para graficar

- gscatter.- grafica los clústeres
- plot
- plotsompos(net) grafica las posiciones de los centroides de la red

Crear Mapa autoorganizado: self-org-map

La sintaxis de selforgmap es:

Red = selforgmap (dimensiones, coverSteps, initNeighbor, topologyFcn, distanceFcn)

Donde:

Dimensiones

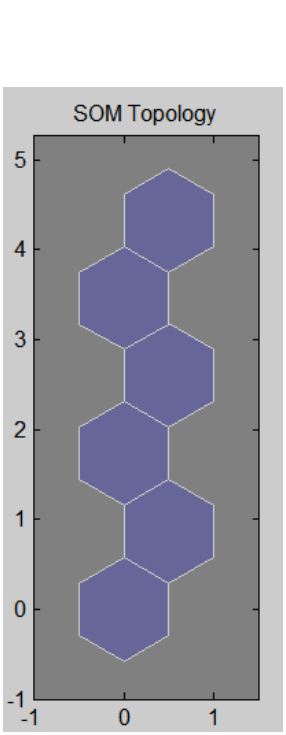
Es un vector fila que representa las dimensiones de la salida.

El producto de la dimension1 x dimension2 es igual al número de clústeres

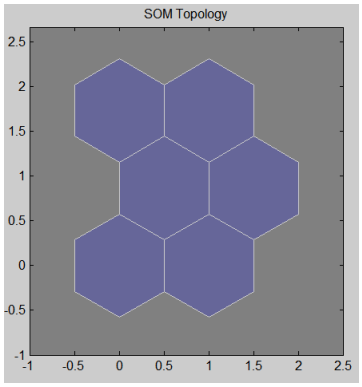
Por ejemplo si se tiene 6 camiones, las dimensiones pueden ser 1x6 o 2x3 o 3x2

En la Figura 30 se presenta un ejemplo de los tres casos.

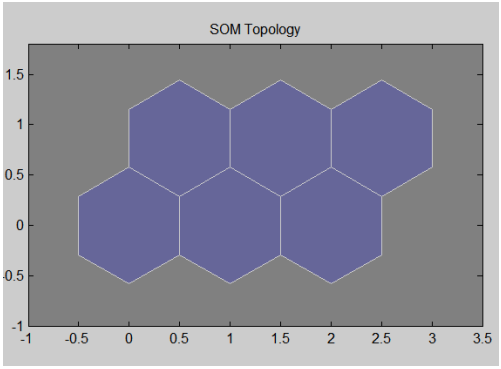
Figura 30 (a) SOM [1 6] (b) SOM [2 3] /c) SOM [3 2]



(a)



(b)



(c)

coverSteps

Número de pasos de entrenamiento para la cobertura inicial del espacio de entrada (predeterminado = 100)

initNeighbor

Tamaño del vecindario inicial (predeterminado = 3). El vecindario va a ir aumentando de acuerdo a las distancias.

topologyFcn

Función de topología de la capa (predeterminada = 'hextop') (Hexagonal)-

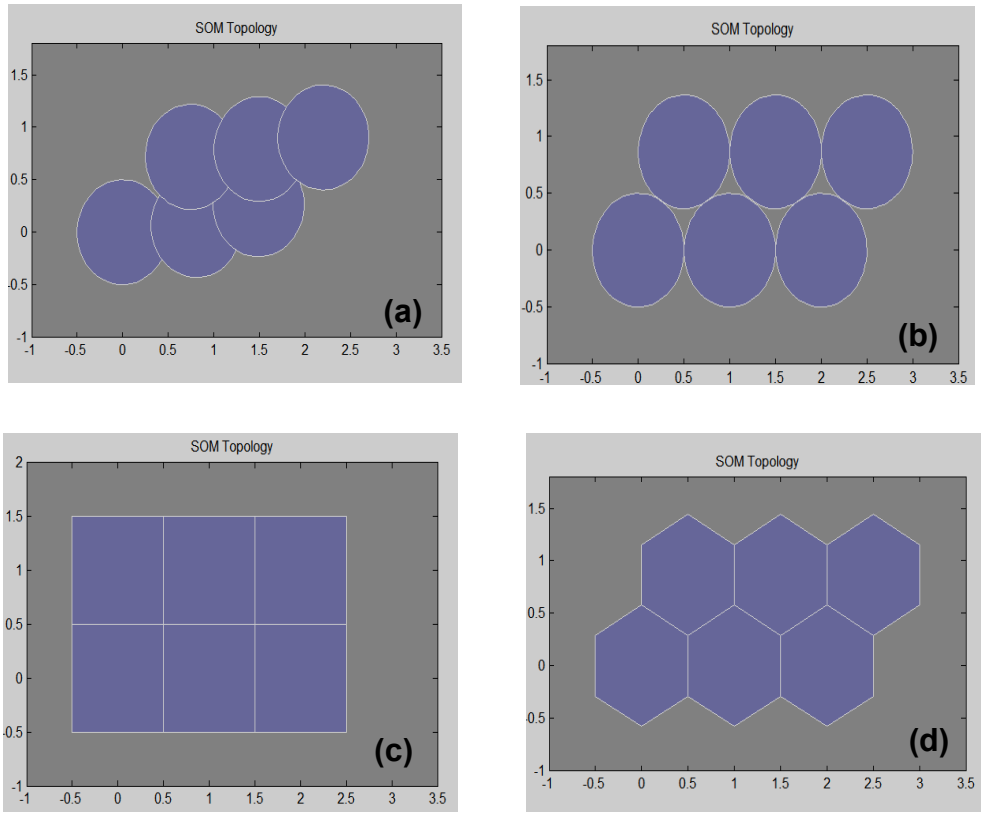
Puede ser tirangular= tritop

Aleatoria = randtop

Cuadricula = Gridtop

En la Figura 31 se presentan ejemplos de los tipos de topología. Aunque en este estudio es más conveniente la topología hexagonal por la densidad de los puntos (esquinas)

Figura 31 Topólogías (a) aleatoria (b) triangular (c) Cuadricula (d) Hexagonal



distanceFcn

Función de distancia neuronal (predeterminado = 'linkdist')

La distancia tipo *linkdist* de una neurona es el número de enlaces, o pasos, que se deben tomar para llegar de una neurona a otra, mientras que la función *dist* calcula las distancias euclidianas, la cual se usar en este estudio.

Entrenar a una red (train)

Se realiza el proceso de utilizar la data de entrada para autoorganizarse. Se utiliza el algoritmo de Kohonen descrito en el acápite 2.4.4.

La sintaxis para entrenar a la red es:

Red_entrenada = train(Red,X)

Donde:

Red. - es la red creada con el comando selforgmap (El nombre de la red antes del entrenamiento y después del entrenamiento puede ser el mismo, es decir, es válido: Red = train(Red,X)

"X".- es el conjunto de datos de entrada, observar que los datos se guardan en una matriz tal que la dimensión se representa en las filas y la cantidad de datos en las columnas., si se tiene un conjunto de 200 datos en R^2, la matriz X es de dimensión 2x200.

Número de iteraciones (epoch)

epoch es la cantidad de veces que todos los vectores de entrenamiento se usan para actualizar los pesos. Para el entrenamiento todos los datos de entrenamiento pasan por algoritmo de aprendizaje simultáneamente en una época antes de que se actualicen los pesos. Comunmente se entiende como una iteración, pero en cada iteración se usan todos los datos de entrenamiento, La sintaxis para indicar las epochs es:

<nombre de la red>.trainParam.epochs =Iter indica el número de iteraciones (epochs)

Donde:

El nombre de la red es la *Red* creada con el comando selforgmap

Iter es el número de veces que se desea entrenar a la red.

Pesos de las neuronas ganadoras

Guarda los pesos (weights) de las neuronas ganadoras de la red.

Sintaxis:

w = <nombre de la red>. IW {1};

Donde

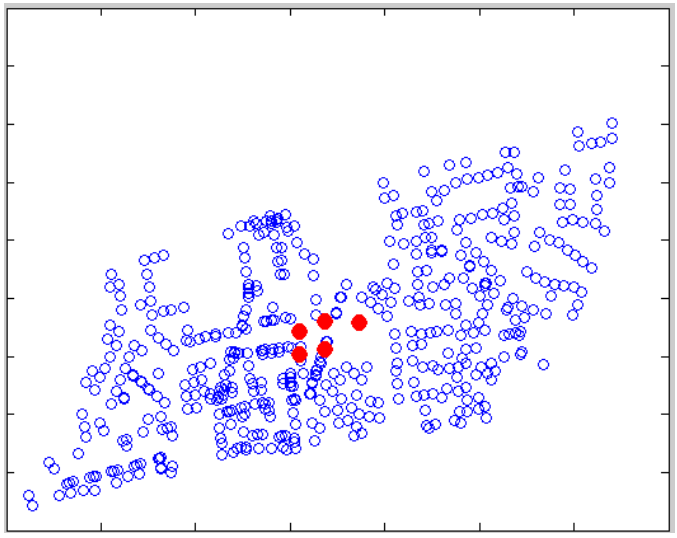
El nombre de la red creada con el comando selforgmap

Vector *IW* son los Input *weights* (pesos que se combinan con las entradas para modificarse)

En este caso de las redes autoorganizadas la dimensiones del IW es de (dimensión de la entrada) X (número de clústeres)

Se inicializan aleatoriamente y en cada epoch se van actualizando. En la figura 32 se muestra las coordenadas (*weights*) de las neuronas ganadoras (centroides) para *epoch* = 1. Los puntos azules son las esquinas del distrito de Bellavista y los puntos rojos las neuronas ganadoras

Figura 32 IW para epoch = 1



En la Figura 33 se observa que las neuronas ganadoras se han actualizado

Figura 33 IW para epoch = 3

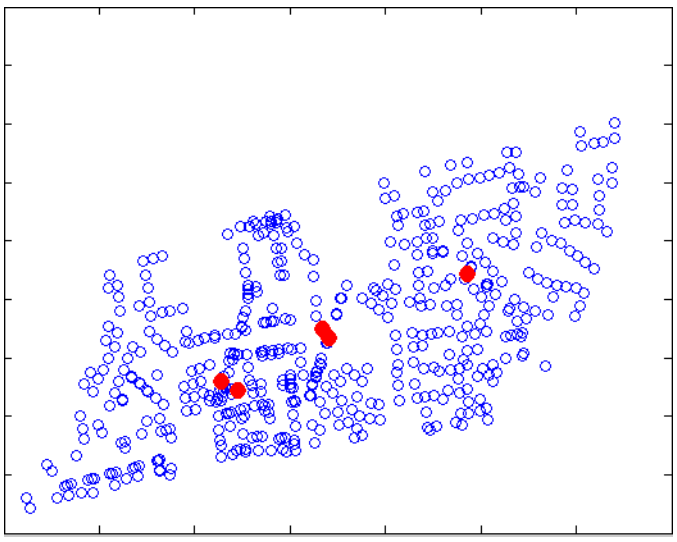


Figura 34 IW para epoch = 5

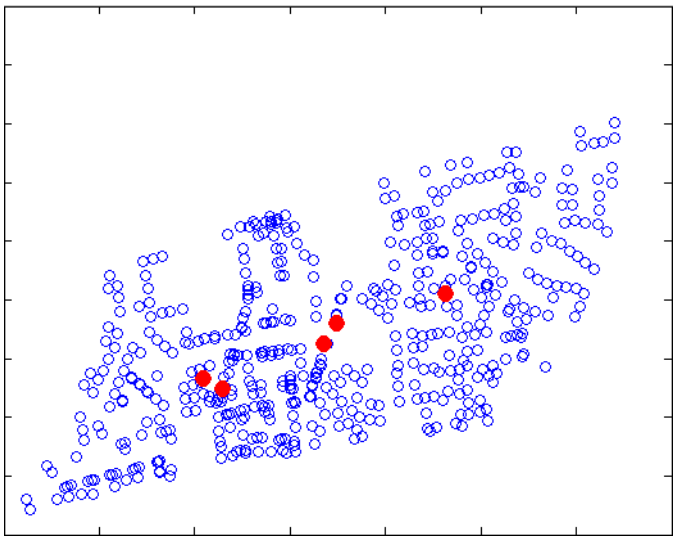
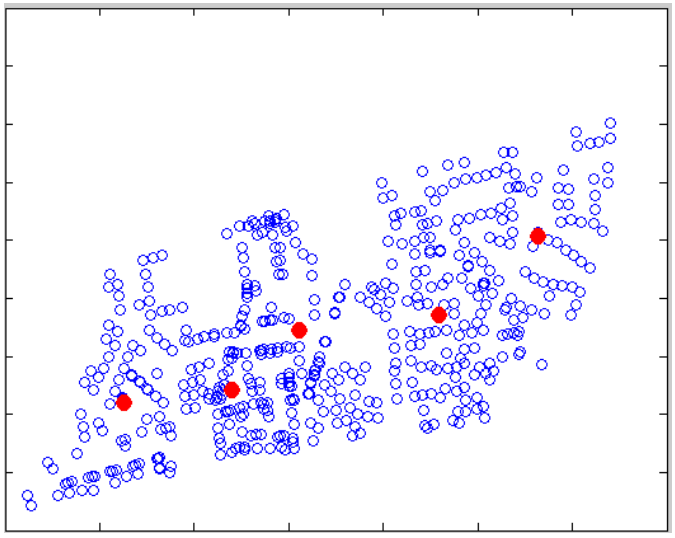


Figura 35 IW para epoch = 200



Luego de 200 epochs los pesos de las cinco neuronas ganadoras se muestran en la tabla 17.

Tabla 17 Pesos de las neuronas ganadoras (UTM) después de 200 epochs

Neurona	Este (X)	Norte (Y)
1	269 904.275	866 5740.704
2	270 530.716	866 5801.453
3	270 912.003	866 6021.929
4	271 711.208	866 6101.435
5	272 295.181	866 6414.860

Identificar a cuál clúster pertenece un dato de entrada

Sintaxis

Cl = sim(Red,caso)

Donde:

Red es el nombre de la red creada con el comando selforgmap

Cl es el clúster al que pertenece el punto caso

Por ejemplo:

Si se quiere saber a cuál clúster pertenece la esquina (-12.0536; -77.0979)

Los comandos serían los siguientes

caso= [-12.0536; -77.0979]

Cl = sim(Red,caso)

En la figura 36 se muestra el resultado

Figura 36 Resultado del comando sim

Cl <5x1 double>

	1	2
1	0	
2	0	
3	0	
4	0	
5	1	

El vector Cl indica que la esquina "*caso*" pertenece al clúster 5

Usar la red entrenada para clústerizar los datos de entrada

Sintaxis:

Grupos = Red(*Datos*);%clústerización de los datos de entrada

Donde:

Red es el nombre de la red creada con el comando selforgmap

Datos son los datos de entrada

Con este comando se ha creado la matriz Grupos la cual tiene tantas filas como clúster se hayan determinado y tantas columnas como los datos de entrada.

En la Figura 37 se muestra la clasificación de los datos de entrada, la esquina 18 pertenece al clúster 4 al igual que la 19, 20, 26 y 27, mientras que las esquinas21 a 25 pertenecen al clúster 5.

Figura 37 Resultado de la aplicación de la red entrenada

Grupos <5x523 double>

	18	19	20	21	22	23	24	25	26	27
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	1	1	1	0	0	0	0	0	1	1
5	0	0	0	1	1	1	1	1	0	0
6										

Observar que en este caso la matriz Grupos tiene 5 filas y 523 columnas. Cinco filas pues se ha creado cinco clústeres y 523 columnas pues se cuenta con 523 datos de entrada (coordenada de las esquinas)

Al sumar cada fila se obtiene la cantidad de elementos que tiene cada clúster

Emparejar los datos de entrada con sus respectivos clústeres

Se usa el comando find (encontrar) el cual proporciona los índices de las neuronas a cuyo vecindario pertenece cada punto de entrada.

Sintaxis:

[x,y]=find(Grupos)

Donde:

> *"x"* es el clúster al que pertenece el punto *"y"*, según la clasificación en la matriz Grupos

En la Figura 38 se muestran los vectores *"x"* y *"y"*

Se observa que la esquina 18 pertenece al clúster 4 al igual que la 19, 20, 26 y 27, mientras que las esquinas 21 a 25 pertenecen al clúster 5.

Figura 38 Resultado de clasificar los datos

x <523x1 double>

	1
15	5
16	5
17	4
18	4
19	4
20	4
21	5
22	5
23	5
24	5
25	5
26	4
27	4
28	5

y <523x1 double>

	1
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28

Con los vectores [x y] se empareja cada dato con su correspondiente clúster, en la figura 39 se muestra la matriz emparejada, donde las dos primeras columnas son las coordenadas de los datos y la tercera columna es el clúster al que pertenece

Figura 39 Resultado de emparejar los datos con sus clúster correspondientes

matriz_para_graficar <523x3 double>

	1	2	3
15	-77.0957	-12.0533	5
16	-77.0956	-12.0539	5
17	-77.0954	-12.0552	4
18	-77.0957	-12.0561	4
19	-77.0955	-12.0569	4
20	-77.0952	-12.0564	4
21	-77.0948	-12.0551	5
22	-77.0950	-12.0538	5
23	-77.0946	-12.0538	5

Este emparejamiento sirve para graficar los puntos de cada clúster de un color diferente., pero previamente hay que ordenar los clústeres, para lo cual se usa el comando sortrows (ordena filas)

matriz_ordenada=sortrows(matriz_para_graficar,3);

ordena la matriz en función de la columna 3

Gráficos

Para graficar se usan los comandos

- gscatter.- grafica los clústeres
- plot.- grafica puntos
- plotsompos(net) grafica las posiciones de las neuronas ganadoras de la red
- plotsomhits(net) grafica la cantidad de elementos de cada clúster luego del entrenamiento.

gscatter

Crea un diagrama de dispersión de x e y, agrupado por grupo.

Sintaxis

gscatter (x, y, grupo, color, marcas, tamaño)

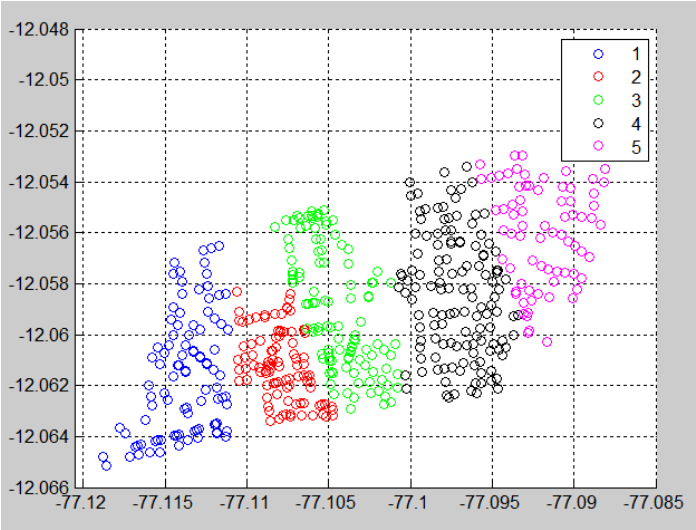
Donde:

- x e y son vectores del mismo tamaño, en este caso coordeadas.
- grupo es una variable de agrupación, una variable categórica, en este caso el número de cada clúster.

Puntos en el mismo grupo aparecen en el gráfico con el mismo marcador y color

Por ejemplo para un conjunto en el que hay cinco grupos, gscatter(x(:,1),x(:,2),group, 'brgkm','o',5) indica que se graficará el diagrama de dispersión de x1 vs x2 tal que el primer grupo se representará de color b (blue) para el grupo 2 el color es r (red) , g(green) para el tercero, k (black) para el cuarto grupo y m(magenta) el último.negro (k) y las marcas son “o” de tamaño 5, como se observa en la Figura 40.

Figura 40 Resultado de gscatter(x(:,1),x(:,2),group, 'brgkm','o',5)



plot

Otro comando utilizado es el graficador de puntos para mostrar en el plano las neuronas ganadoras, esto es más útil cuando lo que se busca es localizar una *facilidad* ,

Sintaxis:

plot(x1,x2, marca, color, tamaño, color del relleno)

Donde

x1 x2 son las variables a graficar en el gráfico de dispersión.

Marca es la forma de símbolo, puede ser cualquier carácter ASCCI

Color son los colores son los standard de Matlab

Tamaño de la marca, se especifica con 'MarkerSize'

Color de relleno, se especifica con 'MarkerFaceColor'

Con la instrucción:

plot(w(:,1),w(:,2),'o' 'k','MarkerSize',10,'MarkerFaceColor','k' ; se plotea los pesos de las neuronas ganadoras, sobre el gráfico de los clústeres:

View(net)

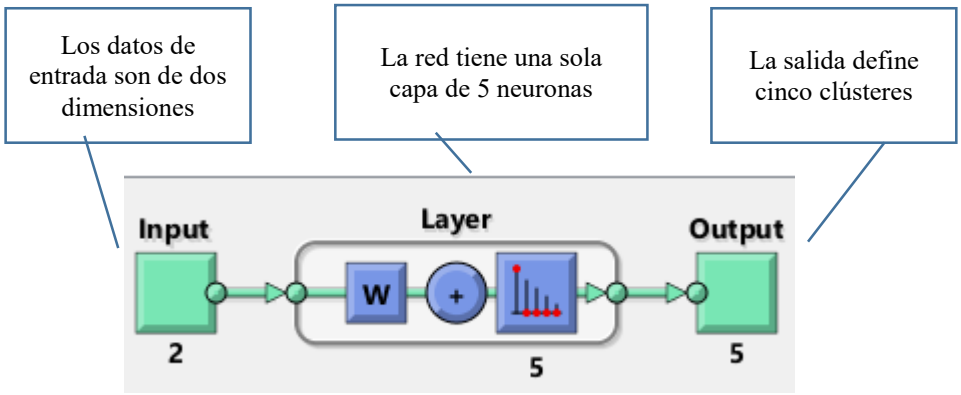
Grafica la estructura de la red (net)

Sintaxis

View(Red) % Red es el nombre de la red

En la Figura 41 se muestra la topología (estructura) de la red

Figura 41 Estructura de la red SOM



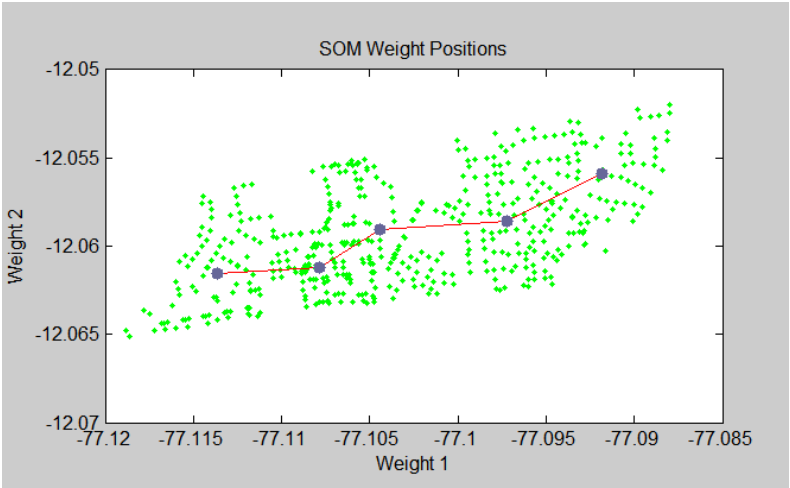
plotsompos

Grafica (*plot*) las posiciones (*pos*) de las neuronas ganadoras de la red autoorganizada (*som*).

Sintaxis:

plotsompos(Red) grafica la red creada en *selforgmap,* luego del entrenamiento durante las epoch establecidas. Lo cual se muestra en la Figura 42.

Figura 42 Resultado de plotsompos de la red



plotsomhits

plotsomhits(net, Datos)

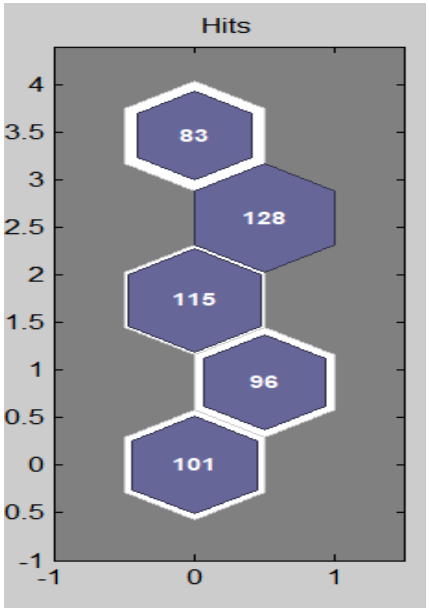
Donde:

Net es la red creada

Datos es el conjunto de datos de entrada

Grafica una capa SOM de la red tal que cada neurona muestra el número de vectores de entrada que pertenecen a su vecindario. El número relativo de vectores para cada neurona se muestra a través del tamaño del hexágono interno.

Figura 43 Cantidad de elementos de cada clúster



Como se observa en la Figura 43

La dimensión de las neuronas de salida es (1, 5)

El clúster 1 tiene 101 esquinas, el clúster 2 tiene 96 y así sucesivamente.

El clúster que tiene menos elementos en su vecindario muestra el hexágono interno de menor tamaño.

CAPITULO 3: MÉTODO

3.1 Tipo y Diseño de Investigación

El presente estudio es de tipo aplicativo, no experimental, se recoge información de la ubicación geográfica de las esquinas del distrito de Bellavista, se organiza en un grafo no orientado y sobre él se construye una red de Kohonen.

3.2 Unidad de análisis

El objeto de estudio son las esquinas del distrito de Bellavista y las relaciones de conectividad entre ellas, la información que se recoge es su latitud y longitud, se convierten a UTM y se aplica la distancia euclidiana para medir distancias.

3.3 Población de estudio

La población o universo en el estudio es el conjunto de esquinas del mencionado distrito puesto que es una aplicación concreta para los mismos.

3.4 Tamaño de muestra

No se consideró una muestra.

3.5. Técnicas de recolección de datos

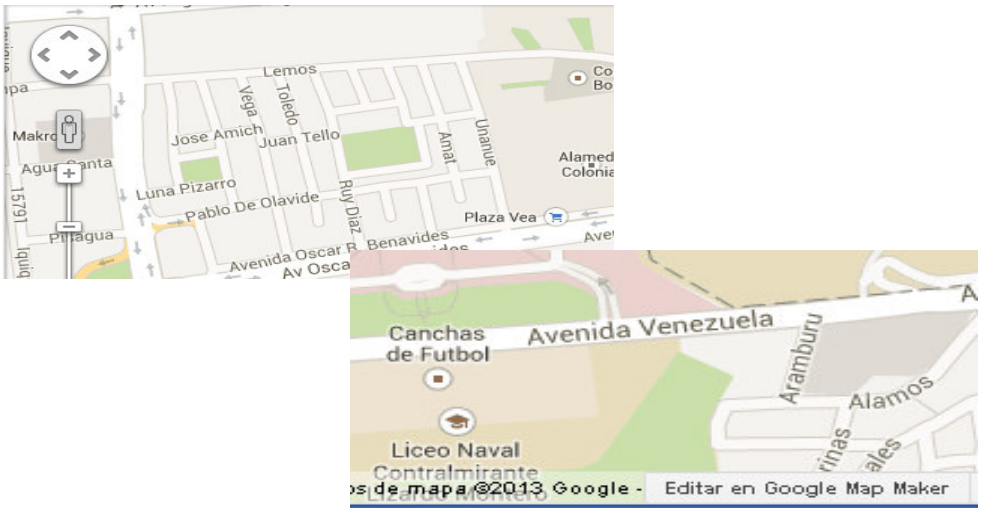
El procedimiento para obtener la latitud y longitud de las unidades de análisis fue observacional, a través de Google map se recogió información de todas las esquinas del distrito.

Se dividió el mapa en 6 regiones arbitrarias y se procedió a construir primero un grafo y luego mediante google maps se registró su latitud y longitud en una hoja de cálculo

Las instrucciones para poder replicar el proceso fueron las siguientes:

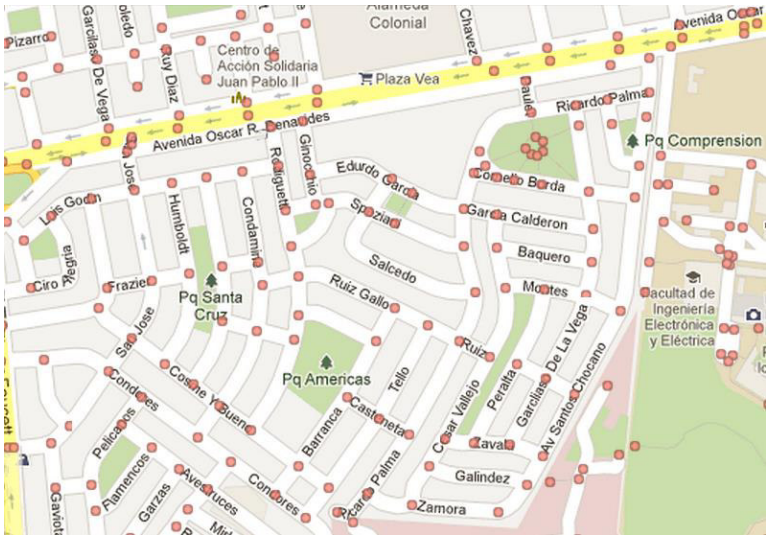
Ingresar a google maps y buscar el distrito en cuestión. Aparece un mapa que se muestra en la Figura 44.

Figura 44 Mapas parciales del distrito de Bellavista - Callao



En google maps seleccionar Editar -Características de línea – Interseccion. En la figura 45 se muestran las intersecciones

Figura 45 Intersecciones del distrito de Bellavista - Callao



A continuación, se extrajo la latitud y longitud de cada intersección lo cual se registró en la tabla 18.

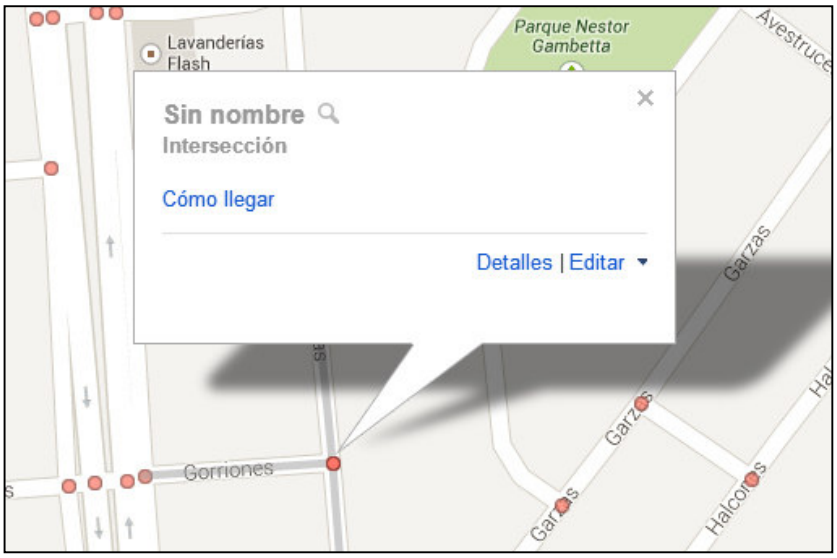
Tabla 18 Latitud y longitud de la región 1

Región	orden	NODOS	Coordenadas geo referenciadas	
			Latitud	Longitud
1	1	S1N1	-12.053611	-77.097906
1	2	S1N2	-12.055422	-77.097633
1	3	S1N3	-12.055897	-77.097528
1	4	S1N4	-12.056411	-77.097517
1	5	S1N5	-12.056375	-77.096975
1	6	S1N6	-12.055317	-77.097181
1	7	S1N7	-12.054625	-77.097253
1	8	S1N8	-12.054347	-77.097542
…				

Donde SiNj representa al nodo j de la región i.

Luego se calculó las distancias entre intersecciones usando la opción “Cómo llegar de A a B” como se ilustra en la Figura 46.

Figura 46 Cálculo de la distancia entre dos nodos



El producto obtenido es un mapa con nodos y aristas parte del cual se muestra en la Figura 47 donde el número que figura es el *nombre* del nodo

Figura 47 Intersecciones (nodos) de una región del distrito en estudi0



El grafo tiene su correspondiente lista de adyacencia, la misma que se muestra en las Tablas19 y 20 solo para los primeros 8 nodos

Tabla 19 Intersecciones (nodos) de la región 1 con sus nodos adyacentes

Región	orden	NODOS	Nodos adyacentes			
			1	2	3	4
1	1	S1N1	S1N2	S1N8		
1	2	S1N2	S1N1	S1N6	S1N3	
1	3	S1N3	S1N2	S1N4	S1N12	
1	4	S1N4	S1N3	S1N5		
1	5	S1N5	S1N4	S1N11		
1	6	S1N6	S1N2	S1N7	S1N10	
1	7	S1N7	S1N6	S1N8	S1N10	
…						

Se considera hasta cuatro adyacentes, dado que es el máximo caso observado en el distrito. Además, se registró las distancias en metros, la cual se muestra en la tabla 20.

Tabla 20 Distancia entre los nodos adyacentes de la región 1

Región	orden	NODOS	Distancia correspondiente en metros			
			1	2	3	4
1	1	S1N1	95	32		
1	2	S1N2	95	52	52	
1	3	S1N3	52	59	160	
1	4	S1N4	59	61		
1	5	S1N5	61	110		
1	6	S1N6	52	81	180	
1	7	S1N7	81	40	59	
1	8	S1N8	32	40	240	
…						

Por ejemplo, en la Tabla 19 se consigna que el nodo S1N1 tiene dos nodos adyacentes S1N2 y S1N8, en la Tabla 20 se registra que la distancia entre los nodos S1N1 y S1N2 es 95 metros y la distancia entre los nodos S1N1 y S1N8 es 32 metros

Así mismo se registró las calles que hacen intersección por si se encontraba errores, las cuales se muestran en la tabla 21.

Tabla 21 Distancia entre los nodos adyacentes de la región 1

Región	orden	NODOS	Distancia correspondiente en metros	
			Calle 1	Calle 2
1	1	S1N1	av. Oscar R. Benavides	av. Elmer Faucett
1	2	S1N2	Ciro Alegria	av. Elmer Faucett
1	3	S1N3	Hipolito Hunanue	av. Elmer Faucett
1	4	S1N4	Condores	av. Elmer Faucett
1	5	S1N5	Gaviotas	Condores
1	6	S1N6	Ciro Alegria	Hurtado Romo
1	7	S1N7	Luis Godin	Hurtado Romo
1	8	S1N8	av. Oscar R. Benavides	Hurtado Romo
…				

3.6 Análisis e interpretación de la información

Como se mencionó, para simplificar la toma de datos, se dividió el distrito en seis regiones se construyó el grafo y se codificaron los nodos.

Desafortunadamente no se contó con información del número de familias y/o personas para poder calcular los volúmenes de residuos que generan diariamente, por lo tanto, no se considera capacidad de los camiones recolectores, solo el número de estos.

Las coordenadas de cada esquina fueron los datos de entrada de la red neuronal creada y por medio de un programa computacional se encontró el conjunto de neuronas ganadoras que definen los clúster, es decir, las particiones del grafo.

3.7 Pasos realizados para la clústerización

3.7.1. Se construyó el grafo del distrito de Bellavista Callao como se detalla en el acápite 3.5, el grafo contiene 523 nodos (esquinas)

Los nodos representados por coordenadas georreferenciadas se convirtieron al sistema UTM (**sistema de coordenadas universal transversal de Mercator**) expresado en metros para poder realizar operaciones y hallar distancias.

3.7.2 Se elaboró un programa en Matlab llamado REDSOM con las siguientes etapas

i. Primera etapa: inicialización del mapa. Para lo cual se ingresa al software los datos, las dimensiones, el número máximo de iteraciones (epoch) y se construye el mapa SOM.

ii. Segunda etapa: entrenamiento del mapa, por cada epoch se presenta todo el conjunto de datos para reajustar los pesos de las neuronas ganadoras, se actualiza la pertenencia de una esquina a un clúster o a otro hasta realizar el número total de epochs ingresada por el usuario en la etapa anterior.

iii. Tercera etapa: visualización del mapa de puntos con un color diferente por cada clúster y el número de elementos por clúster

iv. Cuarta etapa: Verificación de que la distancia de cada esquina a la neurona ganadora correspondiente a su clúster es la mínima.

v. Quinta etapa: Exportación de los resultados a una hoja de cálculo para el informe usando las siguientes instrucciones:

```
%************************************************************************
%Exporta a excel las coordenadas de las esquinas, el clúster al que pertenece y las distancia a cada neurona ganadora
%************************************************************************
xlswrite('<nombre de archivo.xls>',w); % se exporta los pesos de las neuronas ganadoras
xlswrite('<nombre de archivo.xls>",matrix_de_distancias); % las distancias a cada neurona ganadora
```

El script completo del software REDSOM se muestra en el Anexo 4.1

3.7.3. Paralelamente se hizo otro software denominado EVALUARED el cual es una una modificación del REDSOM para que evalúe en varias epochs diferentes pero, que no presente gráficos ni grabe archivos de resultados, sino que solo evalúe diversos valores de

epochs para seleccionar los parámetros que proporcionen una menor distancia máxima al interior de cada clúster. Para lo cual se agregó lo siguiente:

for Iter = 901:1200

….

"*Programa original sin gráficos ni archivos de salida*"

Calcula la distancia máxima al interior de cada clúster y la guarda en un arreglo

….

end

El script de EVALUARED se presenta en el Anexo4.2

Si bien es cierto que 901 a 1200 puede parecer arbitrario y para descartar que con más iteraciones se encuentre mejores soluciones, se analiza todas las distancias máximas comprobando que se distribuyen de acuerdo a una normal, presentando los siguientes resultados:

- Una media cercana a la mediana lo cual indica simetría en la distribución de probabilidad
- Un valor de asimetría cercana a cero, lo cual indica simetría.
- Un valor de curtósis cercano a cero, lo cual indica una campana mesocurtica, como en el caso de la normal.
- Un valor de la prueba Kolmogorov –smirnov que no rechaza la hipótesis de normalidad.

  La hipótesis nula de esta prueba es que la población está normalmente distribuida. Por lo tanto, si el p-valor es mayor que el nivel alfa elegido (en este caso 0.05), entonces no se rechaza la hipótesis nula de que los datos se distribuyen normalmente.
- El grafico Q-Q de Normalidad el cual debe mostrar pocos valores fuera de la recta normal.

CAPITULO 4: RESULTADOS Y DISCUSIÓN

4.1. Análisis, interpretación y discusión de resultados

Se realizaron dos escenarios

Escenario 1: Cinco camiones en una red 1x5

Escenario 2: Seis camiones con configuraciones 1x6, 2x3 y 3x2

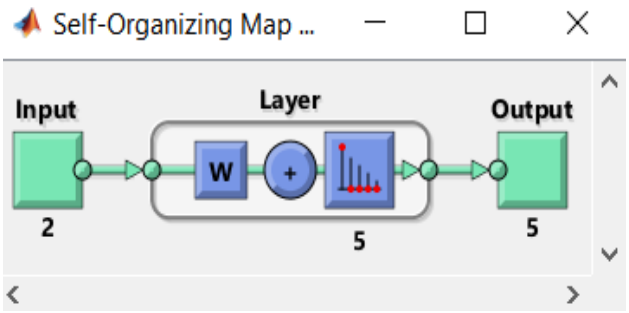
4.1.1 Escenario 1: Cinco camiones en una red 1x5

Se ejecuta el software y se ingresan los parámetros:

Ingrese el nombre del archivo: <*nombre_del_archivo.xlsx*>
clúster = dimension1 x dimension2
Ingrese la dimensión 1: 1
Ingrese la dimensión 2: 5
Ingrese el número de iteraciones (epochs): 1000

En la Figura 48 se muestra la estructura de la red neuronal

Figura 48 Estructura de la red neuronal auto organizada escenario 1



Donde:

- La entrada (Input) es de dimensión dos, pues la data contiene la latitud y la longitud (dos filas)

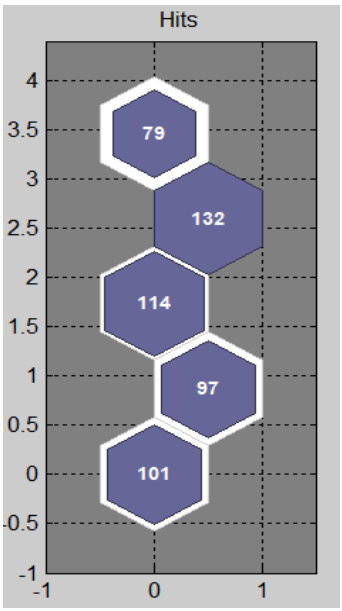
- La única capa tiene cinco neuronas, pues se formaran cinco clústeres.
- La Salida (output) representa a las cinco neuronas (centroides) de cada clúster.

El número de esquinas asociadas a cada clúster se puede calcular contando el número de elementos de cada clúster lo cual se muestra en la Tabla 22 usando el comando *plotsomhits* (Red) cuyo resultado se muestra en la figura 49.

Tabla 22 Esquinas de cada clúster escenario 1

Clúster	Cantidad de esquinas
1	101
2	97
3	114
4	132
5	79
Total	523

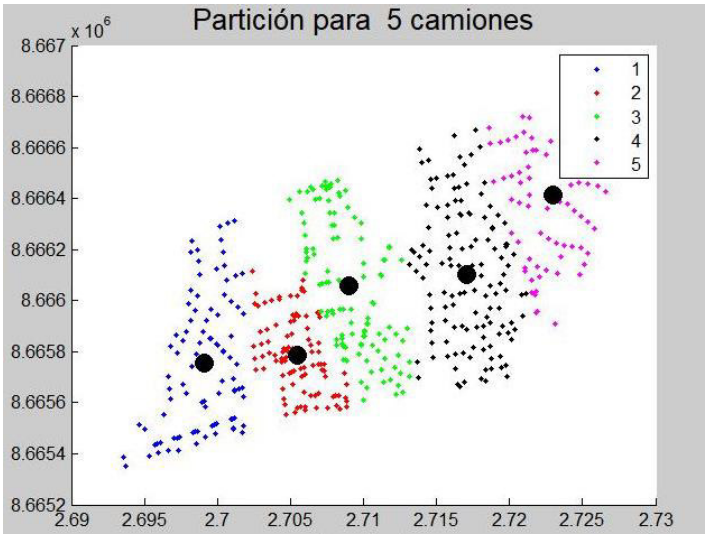
Figura 49 *plotsomhits*(Red) para una dimensión de 1x5 con 1000 epoch



Se seleccionó la topología hexagonal.

El gráfico obtenido se presenta en la Figura 50.

Figura 50 Las esquinas de Bellavista en cinco clúster dimensión (1x5)



Las coordenadas UTM de las neuronas ganadoras se muestran en la Tabla 23.

Tabla 23 Coordenadas de las neuronas ganadoras en UTM escenario 1

Neurona	Este (X)	Norte (Y)
w1	269 907.31	8 665 754.04
w2	270 548.12	8 665 784.35
w3	270 900.55	8 666 058.78
w4	271 708.74	8 666 101.85
w5	272 295.05	8 666 415.49

La distancia promedio de todas las esquinas a su correspondiente neurona ganadora se muestra en la Tabla 24.

Tabla 24 Distancias promedio en metros dentro de cada clúster escenario 1

Clúster	Distancia promedio
1	282.87
2	198.85
3	303.37
4	306.17
5	299.55

Y la distancia máxima dentro de cada clúster, es decir, la distancia entre la esquina más alejada y la neurona ganadora de cada clúster se muestran en la Tabla 25.

Tabla 25 Distancias máximas en metros dentro de cada clúster escenario 1

Clúster	Distancia máxima
1	666.34
2	452.14
3	558.92
4	588.78
5	584.88

Dentro de cada clúster, la distancia máxima de una esquina a la neurona ganadora en el peor de los casos es de 666.34, por lo tanto, la máxima distancia entre dos esquinas del clúster 1 es menor de (666.34*2=1332.68) un aproximado de 13 cuadras.

Prueba 1: la distancia a la neurona ganadora es la mínima

El software calcula las distancias de cada una de esquinas a las cinco neuronas ganadoras comprobándose que la distancia es menor cuando la esquina en cuestión corresponde al clúster de cada neurona ganadora, y se exporta la matriz final la que se muestra en la Tabla 26, donde se observa que si la esquina pertenece al clúster i entonces la distancia mínima se da a la neurona ganadora i-ésima.

Tabla 26 Distancia de todas las esquinas a cada neurona ganadora. escenario 1

Este (X)	Norte (Y)	Clúster	Distancia a la neurona ganadora:					Distancia mínima
			1	2	3	4	5	
269,683.16	8,665,822.74	1	234.45	865.81	1240.07	2044.72	2678.31	234.45
270,314.87	8,665,732.45	2	408.13	238.95	670.46	1441.99	2094.67	238.95
271,314.65	8,665,704.28	3	1408.22	770.70	545.11	559.80	1211.20	545.11

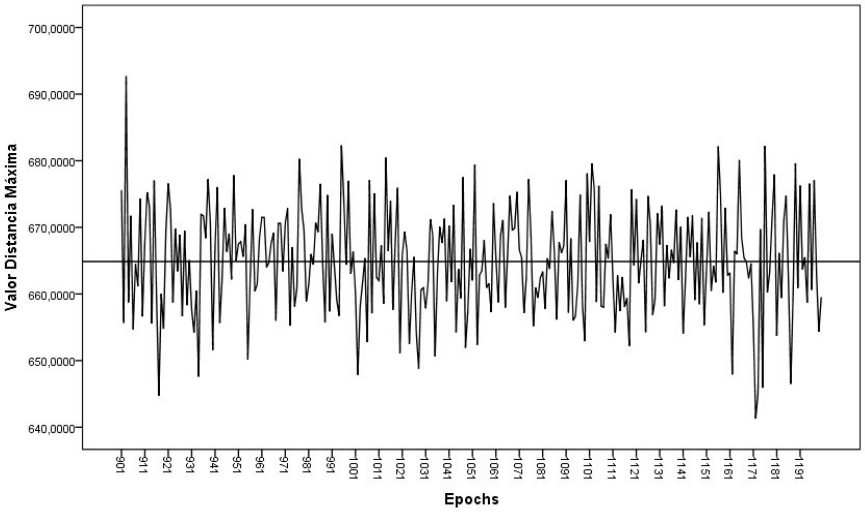
Como se puede observar cuando la esquina pertenece al clúster 1 la distancia de dicha esquina a la neurona ganadora correspondiente al clúster 1 es menor que a cualquier otra.

El chequeo de que la distancia mínima se da dentro de su clúster correspondiente para cada punto de entrada lo realiza el software, en este caso se presenta solo como comprobación.

Pero la pregunta que queda latente es “¿Cómo saber si con un número de epochs diferente de 1000, se consigue una distancia máxima menor?”, para lo cual se utiliza el software mencionado en 3.7.3 y se encontró que si se ejecuta el software con epoch igual a 1172 se obtiene una distancia máxima de 641.31 metros.

Para descartar que al aumentar epochs se mejore la distancia máxima, se aplicó pruebas a los 300 valores máximos, primeramente, se graficaron los datos, los cuales se muestran en la figura 51.

Figura 51 Distancias máximas para epochs desde 901 a 1200



Como se observa no hay una tendencia ni al crecimiento ni al decrecimiento, se aplicaron luego pruebas de normalidad para confirmar la estacionariedad de los valores.

Esto no es ninguna garantía de que con menos de 901 o con más de 1200 se consiga una configuración que proporcione menores distancias máximas, pero como se mencionó, en las soluciones computacionales en algún momento se tiene que hacer un corte y aproximar la solución.

Prueba 2: Prueba de Normalidad de las distancias máximas para diversos epochs

Se ejecutó el programa y los estadísticos descriptivos de los 300 valores encontrados para 5 camiones, con una dimensión de 1x5 se presentan en la tabla 27.

Tabla 27 Estadísticos descriptivos de las máximas distancias (dimensión 1x5)

Estadístico descriptivo	Valor
Media	664.87
Media recortada al 5%	664.93
Mediana	664.70
Desviación estándar	8.00
Mínimo	**641.31**
Máximo	692.68

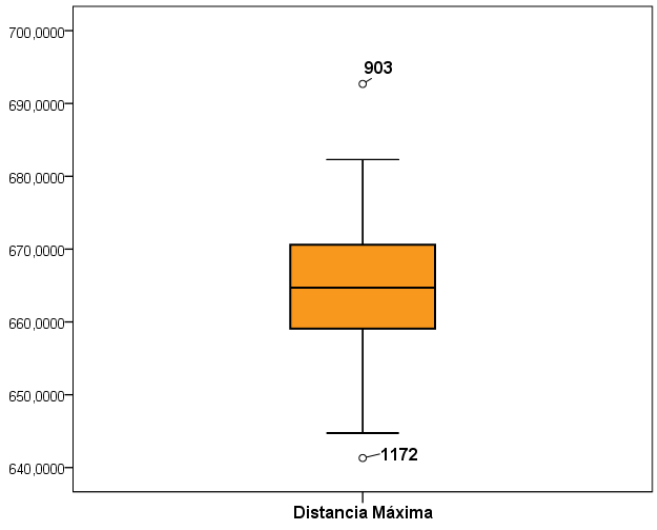
Además, en la tabla 28 se presenta la asimetría y la curtosis lo que podría ser un indicador de que los 300 valores se distribuyen según la distribución normal.

Tabla 28 Asimetría y curtosis de las distancias máximas

Asimetría	Curtosis
0.0032	0.0014

El diagrama de cajas se presenta en la Figura 52

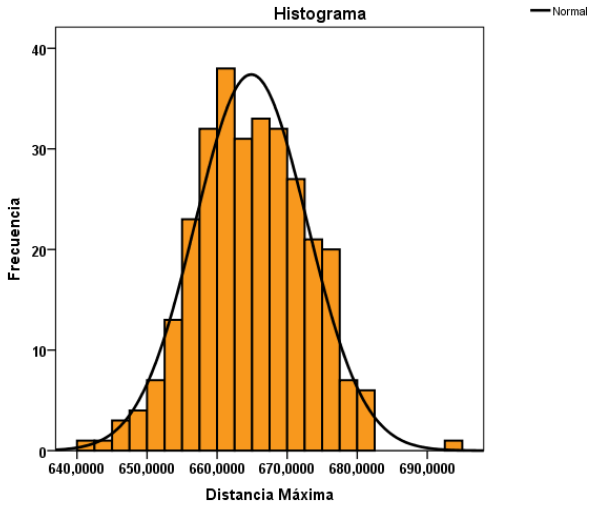
Figura 52 Diagrama de cajas de la distancia máxima (300 epochs diferentes)



Como se observa el valor mínimo es el correspondiente a epoch igual a 1172, lo cual indica que si se construye la red SOM para una dimensión 1x5 con 1172 epoch la distancia máxima que se obtenga será igual al mínimo de la Tabla 27.

La Figura 53 muestra el histograma de los 300 valores de distancias máximas obtenidos.

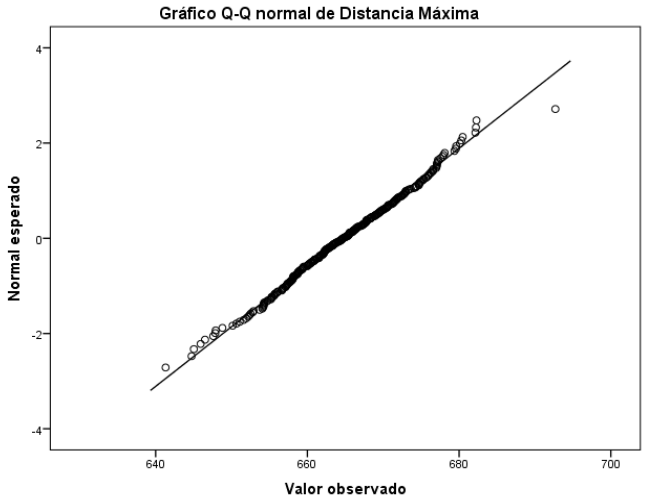
Figura 53 Histograma de las distancias máximas



Se observa

La Figura 54 muestra el diagrama Q-Q de las distancias máximas.

Figura 54 Gráfico Q-Q normal de las distancias máximas



Como se observa hay pocos valores observados, especialmente los extremos, que no coinciden con los valores esperados en el caso de que fueran normales.

Además, se aplicó la prueba Kolmogorov – Smirnov y se obtuvo la Tabla 29

Tabla 29 Pruebas de normalidad

	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
Distancia Máxima	Estadístico	gl	Sig.	Estadístico	gl	Sig.
	.027	300	,200*	.996	300	.659

*. Esto es un límite inferior de la significación verdadera.
a. Corrección de significación de Lilliefors

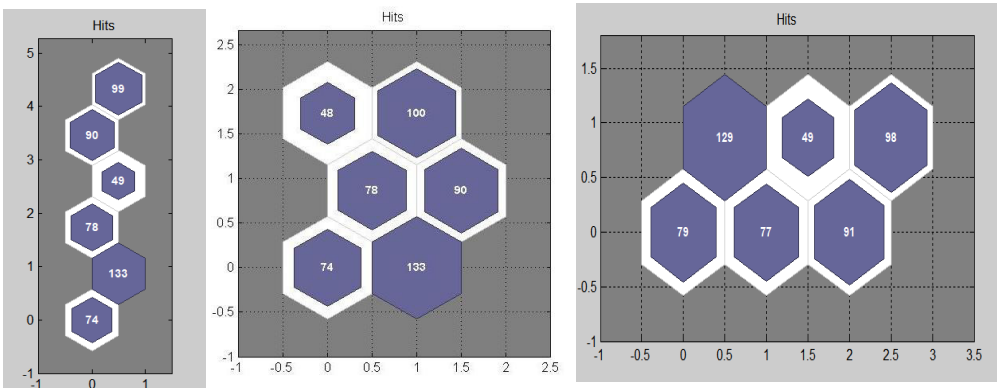
De la tabla se observa que no hay evidencia para rechazar que los datos se distribuyen normalmente. Por lo que se puede suponer que aumentar el número de epochs no necesariamente mejora la distancia máxima.

4.1.2 Escenario 2: Seis camiones

A continuación, se presenta el escenario para 6 camiones, los cuales pueden configurarse de tres maneras: i) dimensión 1x6, ii) dimensión 2x3 y iii) dimensión 3x2.

Las configuraciones se muestran en la figura 55.

Figura 55 Configuraciones posibles para 6 camiones



i) 1x6 ii) 2x3 iii) 3x2.

Se aplicó el módulo EVALUARED del software creado que determina el número de epochs y se obtuvo la Tabla 30.

Tabla 30 Número de epochs sugeridas para minimizar la distancia máxima (caso seis camiones)

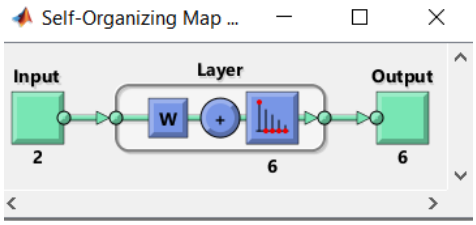
Dimensiones	Epochs	Distancia máxima
1x6	1000	641.454
2x3	1156	642.644
3x2	902	642.350

Se observa que con la configuración 1x6 con epoch igual a 1000, se obtiene la menor distancia máxima, por lo tanto, en el módulo REDSOM del software creado, se ingresó:

```
Ingrese el nombre del archivo: <nombre_del_archivo.xlsx>
# clúster = dimension1 x dimension2
Ingrese la dimensión 1: 1
Ingrese la dimensión 2: 6
Ingrese el Nro de iteraciones: 1000
```

En la Figura 56 se muestra la estructura de la red neuronal

Figura 56 Estructura de la red neuronal auto organizada escenario 2



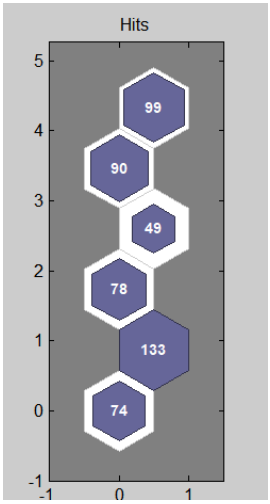
El número de esquinas asociadas a cada clúster se muestra en la Tabla 31.

Tabla 31 Esquinas de cada clúster en el escenario 2

Clúster	Cantidad de esquinas
1	74
2	133
3	78
4	49
5	90
6	99
Total	523

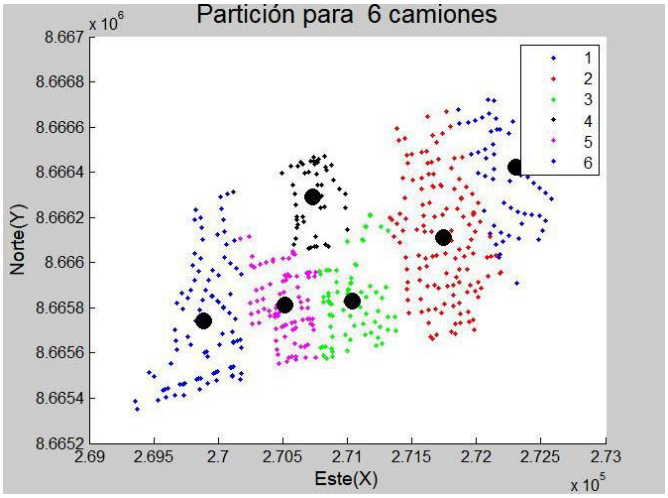
En la Figura 57 se presenta la cantidad de elementos por clúster

Figura 57 *plotsomhits*(Red) para una dimensión de 1x6 con 1000 epoch



El gráfico obtenido se presenta en la Figura 58.

Figura 58 Las esquinas de Bellavista en seis clúster dim(1x6)



Las coordenadas de las neuronas ganadoras se muestran en la Tabla 32.

Tabla 32 Coordenadas de las neuronas ganadoras escenario 2

Neurona	Este (X)	Norte (Y)
w1	272 307.13	8 666 421.91
w2	271 742.08	8 666 112.35
w3	271 035.25	8 665 829.83
w4	270 730.07	8 666 289.40
w5	270 518.86	8 665 813.35
w6	269 886.23	8 665 740.67

La distancia promedio de todas las esquinas a su correspondiente neurona ganadora se muestra en la Tabla 33.

Tabla 33 Distancias promedio en metros dentro de cada clúster escenario 2

Clúster	Distancia promedio
1	290.66
2	304.95
3	205.56
4	180.62
5	183.22
6	279.80

Y la distancia máxima dentro de cada clúster, es decir, la distancia entre la esquina más alejada y la neurona ganadora de cada clúster se muestran en la Tabla 34.

Tabla 34 Distancias máximas en metros dentro de cada clúster escenario 2

Clúster	Distancia máxima
1	571.87
2	599.10
3	420.40
4	609.58
5	455.14
6	**641.45**

Como se observa en la Tabla 34 la distancia máxima, en el peor de los casos es aproximadamente (641.45 *2) metros

Finalmente, cuando se dispone de los clústeres se debe identificar cuales esquinas pertenecen a cuál clúster y programar los ruteos correspondientes, el clúster 1 está conformado por 74 nodos, parte de los cuales se presentan en la tabla 35

Tabla 35 Esquinas del clúster 1

#	Nodo	Calle1	Calle2	Latitud	Longitud	Este (X)	Norte (Y)
1	15	San José	av. Oscar R. Benavides	-12.05334	-77.09573	271,858.88	8,666,676.67
2	16	Luis Godin	San José	-12.05389	-77.09565	271,868.12	8,666,615.27
3	21	Frazier	Humboldt	-12.05512	-77.09482	271,959.61	8,666,480.73
4	22	Humboldt	Luis Godin	-12.05384	-77.09504	271,934.02	8,666,622.23
…	…	…	…	…	…		
74	117	Calle Condores	Calle Barranca	-12.058	-77.094	272,102.67	8,666,128.03

Además, en la Tabla 36 se presentan sus nodos adyacentes y sus distancias correspondientes

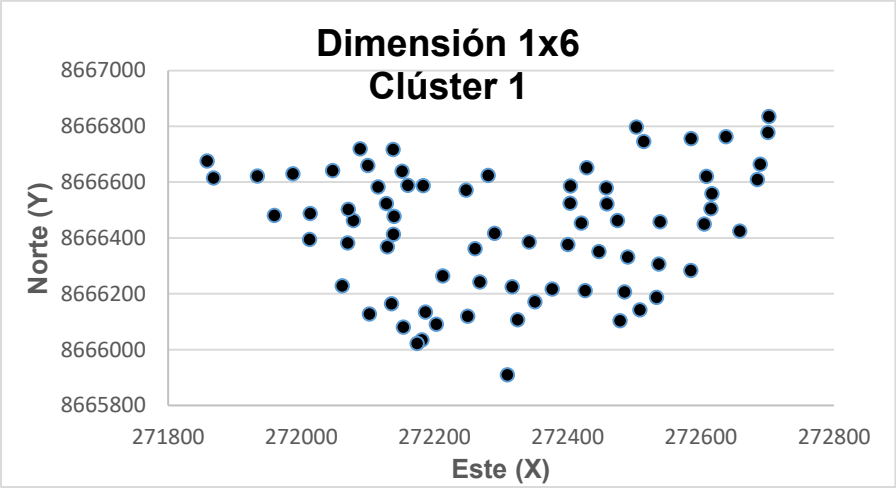
Tabla 36 Nodos adyacentes y distancias

sector	Orden	Nodo	Nodos adyacentes				Distancia correspondiente en metros			
			1	2	3	4	1	2	3	4
1	15	S1N15	S1N8	S1N16	S1N33		240	64	240	
1	16	S1N16	S1N14	S1N15	S1N17	S1N22	71	64	150	61
1	21	S1N21	S1N17	S1N20	S1N22		62	160	150	
1	22	S1N22	S1N16	S1N21	S1N23		61	150	56	
..	..	..	..	..	..	..	..	..	..	..
2	117	S2N24	S2N7	S2N25			62.9	90.13		

Las 74 esquinas del clúster se presentan en el Anexo 6

En la Figura 59 se presenta el grafo del clúster 1

Figura 59 Las esquinas de Bellavista pertenecientes al Clúster 1



Con la información contenida en las tablas 35 y 36 se puede hacer el ruteo correspondiente, lo cual no es tema del presente estudio.

4.1.3 Escenario 3: Nueve camiones

Para nueve camiones, se realiza un proceso análogo, primero se determina las posibles configuraciones, en este caso solo son dos: i) dimensión 1x9 y ii) dimensión 3x3

Entonces para cada caso se debe hallar el número de epochs que hace mínima la mayor distancia y construir la red SOM correspondiente. Para lo cual se construye la Tabla 37

Tabla 37 Número de epochs sugeridas para minimizar la distancia máxima C

Dimensiones	Epochs	Distancia máxima
1x9		
3x3		

Y así se puede hallar los clústeres para cualquier número de camiones disponibles.

4.2. Prueba de la hipótesis de investigación

La hipótesis que orientó la investigación fue “La sectorización de esquinas usando una red de Kohonen al agrupar las esquinas más cercanas en función del número de camiones recolectores, minimiza la distancia recorrida de los camiones recolectores de residuos sólidos”

Como el número de camiones recolectores puede ser variable, se construye una partición de las esquinas del distrito de Bellavista en función del número de camiones, de tal manera que la máxima distancia entre cualquier esquina y su “centroide”, en este contexto “la neurona ganadora”, sea la mínima. Al ser menor la distancia máxima cada camión recorre menores distancias que si recorriera otras esquinas que no pertenecen a su clúster asignado.

Con los dos softwares creados: REDSOM y EVALUARED el proceso de particionar las esquinas del distrito y asignar a cada camión un cierto número de esquinas- las pertenecientes al clúster correspondiente- es una tarea sencilla.

4.3. Presentación de resultados

El presente estudio no pretende resolver el problema para un número dado de camiones recolectores de residuos sólidos en el distrito de Bellavista sino para cualquier número de camiones, por lo tanto, el resultado es más un protocolo que una solución puntual, el cual consiste en:

1. Construir el grafo del distrito, donde los nodos son las esquinas, latitud y longitud, y los arcos las calles que los unen para lo cual se puede usar la plataforma google maps. Dichas coordenadas son transformadas a UTM para poder calcular distancias en metros.
2. Identificar el número de camiones disponibles, teniendo en cuenta, en el caso de Bellavista, que se puede disponer de tres turnos. El uso de los camiones en uno o más turnos es competencia del despachador del área.
3. Determinar las posibles configuraciones, es decir, las formas como se puede descomponer en factores el número de camiones.
4. Evaluar todos los posibles casos usando el software EVALUARED y eligir la configuración que proporciona menor distancia máxima.
5. Aplicar el software REDSOM con los parámetros hallados en el paso anterior y hallar los clústeres.

CAPITULO 5: IMPACTOS

5.1. Propuesta para la solución del problema

El presente estudio particiona un conjunto de esquinas del distrito de Bellavista en clústeres de configuración variable, usando una red de Kohonen, tal que cada clúster determina un conjunto de Voronoi con lo cual se logra agrupaciones de menor distancia; Pero si se sigue el protocolo desde el inicio se puede hacer para cualquier distrito.

5.2. Costos de implementación de la propuesta

El único costo está representado por el recurso humano necesario para construir el grafo.

El INEI tiene información de población por nivel socioeconómico y por manzana que podría integrarse para estimar volúmenes de residuos sólidos por clúster y así asignar los camiones en función de su capacidad. Pero dicha información no está disponible en la página del INEI lo que hace suponer que tiene un costo.

5.3. Beneficios que aporta la propuesta

El principal beneficio es la facilidad de recalcular los clústeres en diversas situaciones de disponibilidad de camiones. Pero, para que la implementación de la presente investigación resulte beneficiosa, la gerencia de gestión de residuos de la Municipalidad debería comprometerse a utilizar los softwares para determinar las nuevas zonas.

Por ejemplo si el Despachador de camiones sabe que el día siguiente va a disponer de tres camiones, que dos pueden hacer tres turnos y uno solo dos

turnos, sabe que debe particionar el distrito en 8 sectores, usando el software EVALUARED, determina si es más eficiente una configuración 8x1, 2x4 o 4x2 y con ello usando el software REDSOM halla los sectores correspondientes.

CONCLUSIONES

La investigación presenta un método para sectorizar las esquinas del distrito de Bellavista en función del número de camiones recolectores disponibles, para lo cual

Se generó una base de datos geográfica del distrito de Bellavista mediante la plataforma google maps, herramienta que es de libre acceso y permite replicar el proceso tanto para el distrito de Bellavista como para cualquier otro distrito.

Dado que en cualquier distrito se dispone de un número finito de camiones recolectores de residuos sólidos domiciliarios y además, dado que no siempre están disponibles todos ellos simultáneamente ya sea por mantenimiento preventivo o por reparaciones, la construcción de una red de Kohonen para sectorizarlas esquinas no puede ser estática, debe ser posible determinar un numero de sectores en función de los camiones disponibles

La red de Kohonen construye conjuntos de Voronoi con lo cual se forman sectores que contienen a las esquinas más cercanas entre sí.

Hacer una sectorización dinámica implicó hacer herramientas computacionales que facilitan los cálculos, con lo cual, dada una situación se puede sectorizar el distrito en un número arbitrario de sectores muy rápidamente.

RECOMENDACIONES PARA FUTURAS INVESTIGACIONES

Estimando volúmenes de población por conjunto de esquinas se puede hacer una estimación de volúmenes de residuos sólidos para asignar los camiones a los sectores, pues cada camión tiene una capacidad dada.

Una vez que se tiene definido cada sector se puede hacer un ruteo mediante el algoritmo del cartero chino, pues el recojo de residuos domicilio por domicilio requiere de caminos eulerianos.

BIBLIOGRAFÍA

Banco_mundial. (2018). *What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050.* Recuperado el 18 de 12 de 2018, de https://www.bancomundial.org/es/news/press-

Bonomo, F., Duran, G., Larumbe, F., & Marenco, J. (2006). *Revista Ingeniería de Sistemas*. Recuperado el 9 de 9 de 2013, de www.dii.uchile.cl/~ris/RISXXIII/Bonomo71.pdf

Fatin, S. (2011). A review on the success factors for community participation in solid waste management. *International Conference on Management (ICM 2011) Proceeding 2011-070-260, Conference Master Resources.*, (pág. 0962_0976). Recuperado el 04 de 02 de 2018, de http://www.internationalconference.com.my/proceeding/icm2011_proceeding/070_260_ICM2011_PG0962_0976_COMMUNITY_PARTICIPATION.pdf

Gallier, J. & Quaintance, J. (2017). *Aspects of Convex Geometry, Polyhedra, Linear Programming,Shellings, Voronoi Diagrams, Delaunay Triangulations.* Philadelphia, PA 19104, USA: University of Pennsylvania.

Garey, M y Johnson, D. (1979). *Computers and Intractability_ A Guide to the Theory of NP-Completeness.* San Francisco- California: Series of Books in the Mathematical Sciences.

Heaton, J. (2011). *Programming Neural Networks with Encog3 in Java.* Publisher: Heaton Research Inc .

Hilera, J. M. (1995). *Redes Neuronales Artificiales. Fundamentos, modelos y aplicaciones. .* Madrid.: Editorial RA-MA.

INEI. (2017). *Informe anual INEI.* Recuperado el 01 de 08 de 2018, de INEI: http://www.inei.gob.pe

Klein, R. (1989). *Concrete and Abstract Voronoi Diagrams.* Berlin: Springer-Verlag Berlin Heidelberg.

Kohonen, T. (1988). An introduction to neural computing. . *Neural Networks , 1*(1), Pages 3-16. Obtenido de

https://www.sciencedirect.com/science/article/pii/0893608088900202

Kohonen, T., Hynninen, J., Kangas, J., & Laaksonen, J. (1995). *SOM PAK: The self-organizing map program package.* Helsinki University of Technology. Helsinki: Laboratory of Computer and Information Science. Recuperado el 5 de 10 de 2020, de https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.455.8698&rep=rep1&type=pdf

Komolafe, S. (2012). *Integrated Solid Waste Management: A Possible Solution to Environmental and Sanitation Problems in the Ancient City of Ibadan.* Tesis de maestría, Linköpings Universitet, Oyo State, Nigeria. Recuperado el 12 de 08 de 2018, de http://liu.diva-portal.org/smash/get/diva2:603041/FULLTEXT01.pdf

Luna, G. (2003). *Factores involucrados en el manejo de la basura doméstica por parte del ciudadano.* Tesis doctoral, Tesis Doctorales en Red . Recuperado el 25 de 08 de 2017, de http://www.tdx.cat/bitstream/handle/10803/2668/GLL_TESIS.PDF

Matich, D. (2001). *Redes Neuronales: Conceptos Básicos y Aplicaciones.* Universidad Tecnológica Nacional – Facultad Regional. Rosario Argentina: Universidad Tecnológica Nacional – Facultad Regional. Recuperado el 12 de 1 de 2018, de https://www.frro.utn.edu.ar/repositorio/catedras/quimica/5_anio/orientadora1/monograias/matich-redesneuronales.pdf

Ministerio de Ambiente, P. (2012). *Ministerio de Ambiente.* Recuperado el 12 de 01 de 2018, de Ministerio de Ambiente: https://www.gob.pe/institucion/minam/informes-publicaciones

Municipalidad de Bellavista, C. (2016). *Plan de gestión y manejo de residuos solidos del distrito de Bellavista .* Recuperado el 02 de 09 de 2019, de Plan de gestión y manejo de residuos solidos del distrito de Bellavista: ,(http://www.munibellavista.gob.pe c/wp-content/uploads/2017/07/3.PGyMRS2016-2021con-plan-de-accion-y-presup..pdf

Municipalidad de Bellavista, C. (2016). *Plan de gestión y manejo de residuos solidos del distrito de Bellavista 2016-202.* Obtenido de Munibella:

http://munibellavista.gob.pe

Municipalidad de Bellavista, C. (2018). *Plan operativo anual.* Obtenido de Plan operativo anual: http://www.munibellavista.gob.pe/wp-content/uploads/2018/10/POI-2018.pdf

Negreiros , M., & Palhano, A. (2006). The capacitated centred clustering problem. *Computers & Operations Research*, 1639–1663.

OECD. (2008). *Environmental outlook to 2030.* (OECD, Ed.) Recuperado el 25 de 08 de 2016, de www.oecd.org/bookshop/

ONU. (2018). *Perspectiva de la gestión de residuos en América Latina y el Caribe.* Ciudad de Panama: Programa de las Naciones Unidas para el Medio Ambiente. Recuperado el 4 de 12 de 2018, de https://news.un.org/es/story/2018/10/1443562

Suocheng, D. T. (2001). Municipal solid waste management in China: using commercial. *Utilities Policy*, 7–11.

Tchobanoglous, G. (1994). *Gestión Integral De Residuos Sólidos. .* Madrid: McGRAW-HILL.

Terraza, H. (2009). *Lineamientos estratégicos del Banco Interamericano de Desarrollo para el sector de residuos sólidos (2009 – 2013) .* Obtenido de Banco Inter-Americano de Desarrollo: http://idbdocs.iadb.org/wsdocs/getdocument.aspx?docnum=2197909

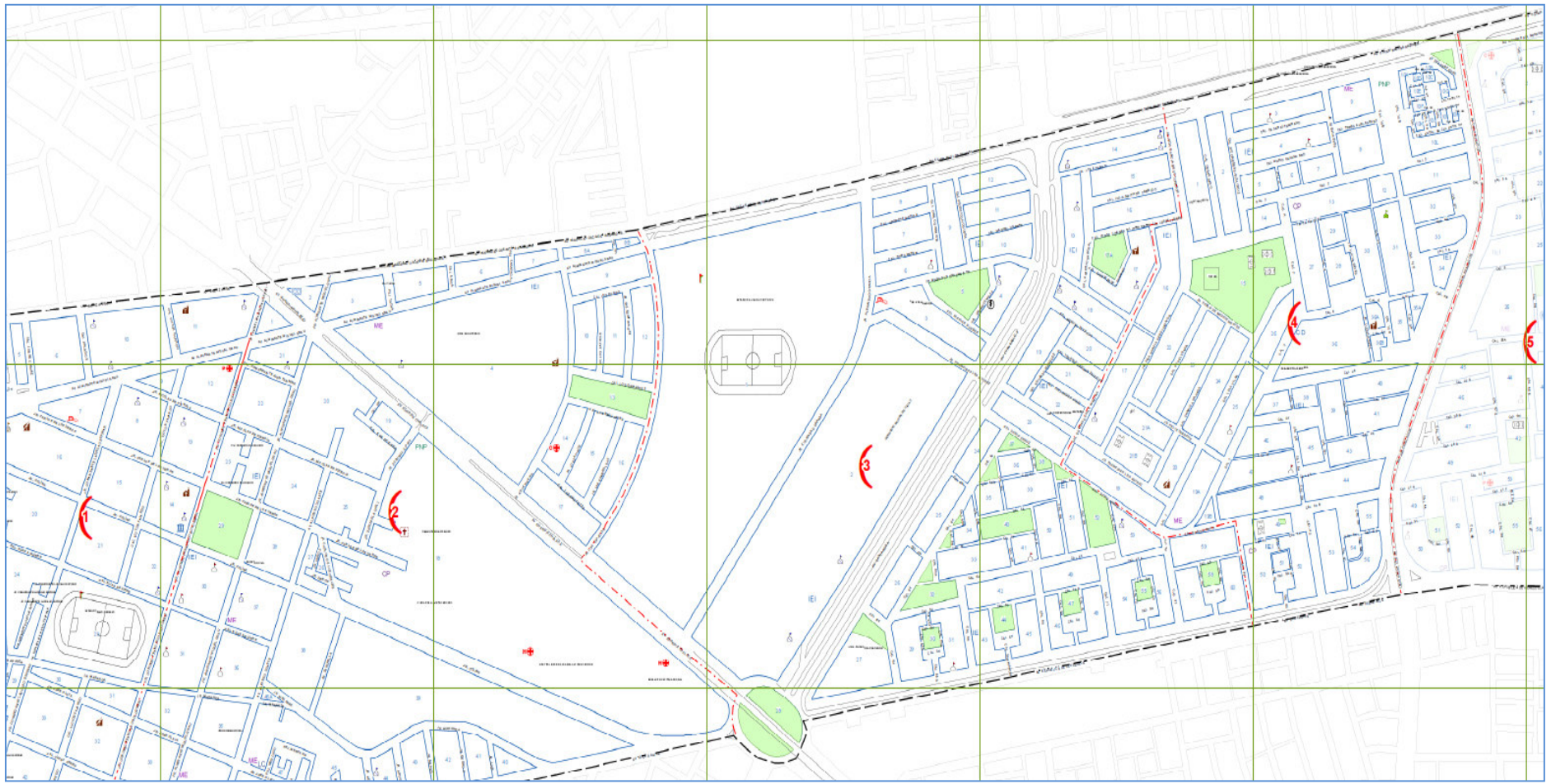
Theisen, H. (2000). *Gestión Integral de Residuos Sólidos.* Buenos Aires: ARS.

UNED. (2003). *Los residuos urbanos y su problemática.* Recuperado el 09 de 09 de 2018, de http://www.uned.es/biblioteca/rsu/pagina1.htm

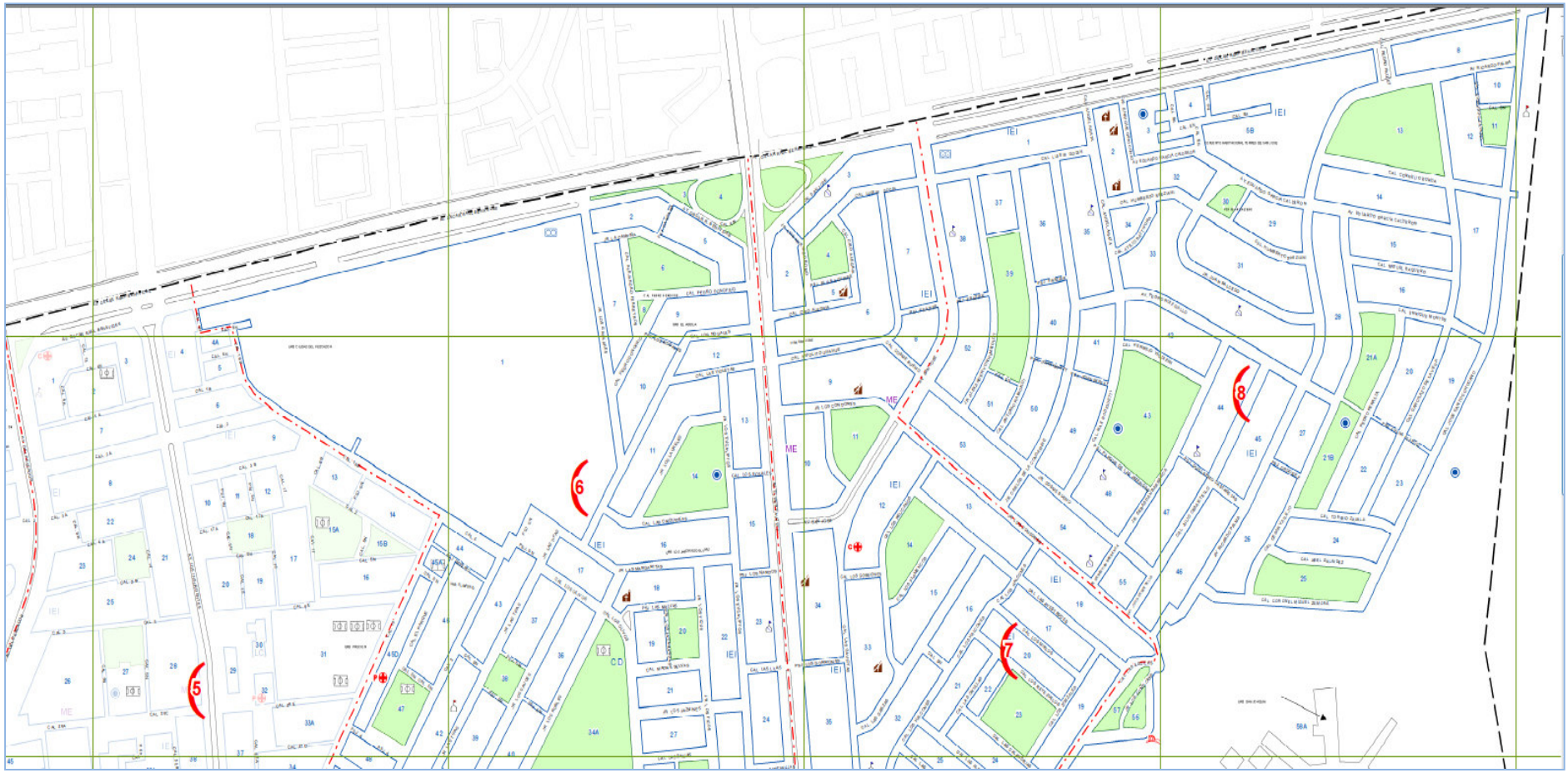
ANEXOS

ANEXO N°1: Distrito de Bellavista por manzanas

Zonas 1 a 5



Zonas 5 a 8



ANEXO N°2 Ejemplo de red supervisada

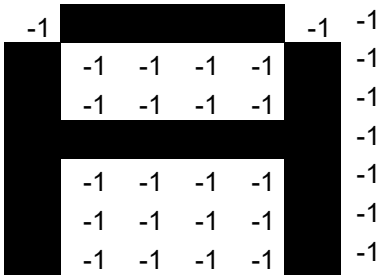
	Entrada				Pesos iniciales			Función de entrada				Función de activación ns = neurona de salida		Función de transferencia			
	Valores de entrada			Resultado real										Corrección	Pesos finales (corregidos)		
	x0	x1	x2	z=1 Apto z=0 no apto	w0	w1	w2	x0w0	x1w1	x2w2	s = Sxiwi	if s > umbral then ns = 1, else ns = 0	e = z-ns	Tasa de aprendizaje*error	w0(t+1)	w1(t+1)	w2(t+1)
1	1	0.39	0.69	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0	1	0.1	0.1	0.039	0.069
2	1	0.71	0.18	0	0.10	0.04	0.07	0.10	0.03	0.01	0.14	0	0	0	0.1	0.039	0.069
3	1	0.96	0.64	1	0.10	0.04	0.07	0.10	0.04	0.04	0.18	0	1	0.1	0.2	0.135	0.133
4	1	0.42	0.65	1	0.20	0.14	0.13	0.20	0.06	0.09	0.34	0	1	0.1	0.3	0.177	0.198
5	1	0.76	0.73	1	0.30	0.18	0.20	0.30	0.13	0.14	0.58	1	0	0	0.3	0.177	0.198
6	1	0.92	0.61	1	0.30	0.18	0.20	0.30	0.16	0.12	0.58	1	0	0	0.3	0.177	0.198
7	1	0.17	0.32	0	0.30	0.18	0.20	0.30	0.03	0.06	0.39	0	0	0	0.3	0.177	0.198
8	1	0.12	0.65	0	0.30	0.18	0.20	0.30	0.02	0.13	0.45	1	-1	-0.1	0.2	0.165	0.133
9	1	0.09	0.48	0	0.20	0.17	0.13	0.20	0.01	0.06	0.28	0	0	0	0.2	0.165	0.133
10	1	0.56	0.26	0	0.20	0.17	0.13	0.20	0.09	0.03	0.33	0	0	0	0.2	0.165	0.133
11	1	0.3	0.72	1	0.20	0.17	0.13	0.20	0.05	0.10	0.35	0	1	0.1	0.3	0.195	0.205
12	1	0.77	0.1	0	0.30	0.20	0.21	0.30	0.15	0.02	0.47	1	-1	-0.1	0.2	0.118	0.195
13	1	0.4	0.3	0	0.20	0.12	0.20	0.20	0.05	0.06	0.31	0	0	0	0.2	0.118	0.195
14	1	0.42	0	0	0.20	0.12	0.20	0.20	0.05	0.00	0.25	0	0	0	0.2	0.118	0.195
15	1	0.26	0.17	0	0.20	0.12	0.20	0.20	0.03	0.03	0.26	0	0	0	0.2	0.118	0.195
16	1	0.65	0.57	1	0.20	0.12	0.20	0.20	0.08	0.11	0.39	0	1	0.1	0.300	0.183	0.252
17																	
18																	
27	1	0.68	0.4	1	0.30	0.18	0.27	0.30	0.13	0.11	0.53	1	0	0	0.300	0.184	0.272
28	1	0.73	0.86	1	0.30	0.18	0.27	0.30	0.13	0.23	0.67	1	0	0	0.300	0.184	0.272
29	1	0.12	0.56	0	0.30	0.18	0.27	0.30	0.02	0.15	0.47	1	-1	-0.1	0.200	0.172	0.216
30	1	0.1	0.49	0	0.20	0.17	0.22	0.20	0.02	0.11	0.32	0	0	0	**0.200**	**0.172**	**0.216**

ANEXO N°3 Ejemplo de una red de Hopfield

Cada letra se representa en una matriz de 7 filas por 6 columnas, para facilidad de los cálculos la matriz debería ser cuadrada, por lo que se agrega una columna en blanco al final.

Los espacios marcados en negro se codifican como “1” y los espacios en blanco como “-1”

El estado del patrón A consta de 49 caracteres que pueden ser 1 o -1,



A se almacena como un vector de 49 dimensiones

1	-1
2	1
3	1
4	1
5	1
6	-1
7	-1
8	1
9	-1
10	-1
11	-1
12	-1
	…
	…
49	1

Se calcula $W_{ij}\begin{cases}\sum x\,x^{t} & i \neq j \\ 0 & i = j\end{cases}$ análogamente para para cada patrón

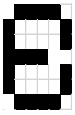
En Matlab se calculó de la siguiente manera

Fase de entrenamiento

1. AA = A*A’; BB=B*B’ ……..DD=D*D’

2. I = eye(49) la matriz identidad de dimensión 49
3. WA = AA-I la matriz de pesos asociada a la entrada “A”, análogamente WB, WC y WD.
4. W = WA+WB+WC+WD La matriz de pesos de la red

Fase de aplicación

5. Se desea reconstruir la entrada 
6. Se traduce a un vector de 49 posiciones llamado Nuevo
7. Se calcula el estado E_1 = Nuevo*W
8. Se calcula la salida de la red neuronal en la iteración k, S_k

$$s_{ik}\begin{cases} 1 & \text{si } e_{ik} \geq 0 \\ -1 & \text{si } e_{ik} < 0 \end{cases}$$

9. El estado $E_{(k+1)} = S_k$*W
10. Se repiten los pasos 8 y 9 hasta que la diferencia entre S_k y $S_{(k+1)}$ sea cero, es decir cuando el proceso ha sido atraído por un patrón.
11. Se compara la última salida Sk con los patrones que sirvieron para el entrenamiento y se encuentra a cuál se aproxima más.

Los resultados numéricos se presentan en las tablas ____ y _____

Tabla __

Estados y salidas de la entrada a reconstruir

Neurona	1	2	3	4	5	6	7	8	9	…	47	48	49
E1	22	52	52	60	38	-34	-52	52	-52	…	0	4	-52
S1	1	1	1	1	1	-1	-1	1	-1	…	1	1	-1
										…			
E2	54	136	136	136	78	-82	-136	136	-136	…	4	-8	-136
S2	1	1	1	1	1	-1	-1	1	-1	…	1	-1	-1
										…			
E3	66	136	136	136	82	-78	-136	136	-136	…	12	-8	-136
S3	1	1	1	1	1	-1	-1	1	-1	…	1	-1	-1

Tabla

Representación de los datos de entrada y la salida de la red

#	A	B	C	D	S3
1	-1	1	1	1	1
2	1	1	1	1	1
3	1	1	1	1	1
4	1	1	1	1	1
5	1	1	1	-1	1
6	-1	-1	1	-1	-1
7	-1	-1	-1	-1	-1
8	1	1	1	1	1
9	-1	-1	-1	-1	-1
10	-1	-1	-1	-1	-1
	…				…
48	1	-1	1	-1	-1
49	-1	-1	-1	-1	-1

Como se puede observar las mayores semejanzas de S3 son con la entrada “B”, por lo tanto se concluye que el dato a evaluar se corresponde con el patrón “B”

ANEXO N°4.1: Script de clusterizacion mediante Kohonen REDSOM

```
%************************************************************************
% Descripción de las variables:
%************************************************************************
%   archivo: es un archivo en excel nombre_Archivo.xlsx con las esquinas a clusterizar
%   Datos : son los datos de entrada
%   cluster: Numero de cluster
%   Iter: máximo número de iteraciones (epochs)
%   Red: red SOM
%   w: pesos de las neuronas ganadoras (en este caso coordenadas)
%   x: da los indices de las neuronas ganadoras a la que pertence cada punto
%   y: númeración de los nodos
%   cluster_index: registra el cluster de cada dato de entrada, matriz de 1xcant_datos
% Lee datos
archivo=input('Ingrese el nombre del archivo: ','s');
Datos=xlsread(archivo);
fprintf('# cluster = dimension1 x dimension2\n');
dimension1 = input('Ingrese la dimensión 1: ');
dimension2=input('Ingrese la dimensión 2: ');
clusters = dimension1*dimension2;
Iter=input('Ingrese el Nro de iteraciones: ');

%************************************************************************
% Crea una red de kohonen(Self-Organizing Map)
%************************************************************************
Red = selforgmap([dimension1 dimension2], 10,3, 'hextop', 'dist');
% selforgmap(dimensions,coverSteps,initNeighbor,topologyFcn,distanceFcn)

%************************************************************************
% Entrena a la red
%************************************************************************
Red.trainParam.epochs =Iter;
Red= train(Red,Datos);
w = Red.IW{1};

%************************************************************************
%clasifica los Datos, organizandolos por cluster
%************************************************************************
Forma_Grupos = Red(Datos);% implementa la red SOM en los datos de entrada
[x,y]=find(Forma_Grupos);
cluster_index = vec2ind(Forma_Grupos);% igual que x pero en una fila
matriz_para_graficar=[Datos' x]; % nueva matriz con los puntos clasificados
matriz_ordenada=sortrows(matriz_para_graficar,3);% ordena la matriz  en funcion de la columna 3

%************************************************************************
% Grafica
%************************************************************************
gscatter(matriz_ordenada(:,1),matriz_ordenada(:,2),matriz_ordenada(:,3),'brgkm','.',10); % gscatter grafica
los clusters de diferente color
Grupos=matriz_ordenada(:,3);%solo contiene el cluster al que pertenece cada punto
hold on;
plot(w(:,1),w(:,2),'ok','MarkerSize',10, 'MarkerFaceColor','k');
% plot grafica las neuronas ganadoras en negro y más grandes
title(['Partición para  ',num2str(dimension1*dimension2),' camiones'],'FontSize',16)
ylabel('Latitud','FontSize',12)
xlabel('Longitud','FontSize',12)
```

```
hold on;
% Guarda la figura de los clusters en archivo jpg
I = getframe(gcf);
imwrite(I.cdata,'C:\Users\user\Documents\2020\pa tesis IO\clusters.jpg');
%presenta los gráficos estándar de una red som
plotsompos(Red)
plotsomhits(Red, Datos)
view(Red);% muestra la estructura de la red
hold off
grid on

%************************************************************************
% evalua una esquina de prueba para determinar a cual cluster pertenece
%************************************************************************
%esquina= [-12.05;   -77.097]
%Cl = sim(Red,esquina)% indica a cuál cluster pertenece el punto "esquina"

%************************************************************************
%Verificación de errorres
%************************************************************************
% evalua que las distancias de cada punto son menores dentro de su propio
% cluster
for g = 1:clusters
  for n = 1:length(Datos)
    cant(g)=numel(Grupos(Grupos==g));
    % cuenta cuantos elementos hay en cada cluster
  end
end
coordenadas = matriz_ordenada(:,1:2)% extrae las primeras dos columnas
for g = 1:clusters
  for n = 1:length(Datos)
    distancias(n,g) = sqrt((coordenadas(n,1)-w(g,1))^2+ (coordenadas(n,2)-w(g,2))^2);
    %calcula distancias de cada punto de entrada a las g neuronas ganadoras
  end
end
for n = 1:length(Datos)
    Minimos = min(distancias,[],2)% halla el minimo de cada fila de la matriz distancias
    % 2 significa minimo de fila si fuese min de columna debe ser 1
end
matrix_de_distancias =[coordenadas distancias Minimos Grupos]%junta cuatro arreglos
for n = 1:length(Datos)
   aciertos(n)=distancias(n,Grupos(n))-Minimos(n)%si el punto pertenece al grupo n la distancia minima es a
la neurona (n)
end
error=sum(aciertos);
fprintf('El error es %i\n', error);

%************************************************************************
%Exporta a excel las coordenadas, el cluster al que pertenece, las distancia a cada neurona ganadora
%************************************************************************
xlswrite('C:\Users\user\Documents\2020\w.xls',w);
xlswrite('C:\Users\user\Documents\2020\matrix_de_distancias.xls',matrix_de_distancias);
```

ANEXO N°4.2: Script para evaluar la estructura de la red de Kohonen EVALUARED

```
%************************************************************************
% Descripción de las variables:
%************************************************************************
%   archivo: es un archivo en excel nombre_Archivo.xlsx con las esquinas a clusterizar
%   Datos : son los datos de entrada
%   cluster: Numero de cluster
%   Iter: máximo número de iteraciones (epochs)
%   Red: red SOM
%   w: pesos de las neuronas ganadoras (en este caso coordenadas)
%   x: da los indices de las neuronas ganadoras a la que pertence cada punto
%   y: númeración de los nodos
%   cluster_index: registra el cluster de cada dato de entrada, matriz de 1xcant_datos
% Lee datos
archivo=input('Ingrese el nombre del archivo: ','s');
Datos=xlsread(archivo);
fprintf('# cluster = dimension1 x dimension2\n');
dimension1 = input('Ingrese la dimensión 1: ');
dimension2=input('Ingrese la dimensión 2: ');
clusters = dimension1*dimension2;
for Iter = 901:1200
    rng('default')% restaura el generador de numeros aleatorios, para la ubicaciíon inicial de las neuronas
ganadoras
%************************************************************************
% Crea una red de kohonen(Self-Organizing Map)
%************************************************************************
Red = selforgmap([dimension1 dimension2], 10,3, 'hextop', 'dist');
%************************************************************************
% Entrena a la red
%************************************************************************
Red.trainParam.epochs =Iter;
Red= train(Red,Datos);
w = Red.IW{1};

%************************************************************************
%clasifica los Datos, organizandolos por cluster
%************************************************************************
Forma_Grupos = Red(Datos);% implementa la red SOM en los datos de entrada
[x,y]=find(Forma_Grupos);
cluster_index = vec2ind(Forma_Grupos);% igual que x pero en una fila
matriz_para_graficar=[Datos' x]; % nueva matriz con los puntos clasificados
matriz_ordenada=sortrows(matriz_para_graficar,3);% ordena la matriz  en funcion de la columna 3

%************************************************************************
%Verificación de errorres
%************************************************************************
% evalua que las distancias de cada punto son menores dentro de su propio
% cluster
for g = 1:clusters
  for n = 1:length(Datos)
     cant(g)=numel(Grupos(Grupos==g));
     % cuenta cuantos elementos hay en cada cluster
  end
end
coordenadas = matriz_ordenada(:,1:2)% extrae las primeras dos columnas
for g = 1:clusters
```

```
  for n = 1:length(Datos)
    distancias(n,g) = sqrt((coordenadas(n,1)-w(g,1))^2+ (coordenadas(n,2)-w(g,2))^2);
    %calcula distancias de cada punto de entrada a las g neuronas ganadoras
  end
end
for n = 1:length(Datos)
    Minimos = min(distancias,[],2)% halla el minimo de cada fila de la matriz distancias
    % 2 significa minimo de fila si fuese min de columna debe ser 1
end
Maximo(Iter-900)=max(Minimos);
matrix_de_distancias =[coordenadas distancias Minimos Grupos]%junta cuatro arreglos
for n = 1:length(Datos)
   aciertos(n)=distancias(n,Grupos(n))-Minimos(n)%si el punto pertenece al grupo n la distancia minima es a
la neurona (n)
end
error=sum(aciertos);
fprintf('El error es %i\n', error);
```

ANEXO N°5: Distancias máximas para diversos epoch. Caso 3x2

Evaluación de distancias máximas en metros para diversos valores de epoch
Dimensiones 3x2

epoch	distancia maxima	epoch	distancia maxima	epoch	distancia maxima	epoch	distancia maxima	epoch	distancia maxima
901	646.12	931	655.43	961	656.99	991	659.89	1021	662.36
902	642.35	932	652.44	962	668.68	992	663.30	1022	668.62
903	665.04	933	663.62	963	677.74	993	657.22	1023	663.38
904	661.25	934	649.55	964	668.62	994	651.95	1024	671.27
905	665.07	935	667.63	965	648.56	995	654.66	1025	657.82
906	661.39	936	664.67	966	646.64	996	653.76	1026	672.04
907	675.19	937	656.88	967	653.44	997	651.50	1027	667.00
908	662.90	938	658.90	968	649.01	998	675.77	1028	659.00
909	658.46	939	666.94	969	668.45	999	650.42	1029	656.74
910	671.51	940	652.17	970	656.50	1000	647.44	1030	668.73
911	655.79	941	663.83	971	677.01	1001	671.19	1031	662.26
912	654.93	942	667.70	972	668.13	1002	656.04	1032	668.01
913	658.68	943	661.52	973	667.50	1003	663.15	1033	664.96
914	668.47	944	660.85	974	663.52	1004	660.55	1034	665.50
915	659.71	945	655.06	975	665.94	1005	650.80	1035	677.04
916	666.04	946	678.03	976	666.62	1006	664.31	1036	666.08
917	671.07	947	667.54	977	663.09	1007	659.09	1037	645.15
918	666.69	948	656.20	978	662.18	1008	671.70	1038	667.89
919	667.60	949	658.56	979	664.71	1009	659.70	1039	657.24
920	676.41	950	671.96	980	668.21	1010	657.53	1040	659.50
921	652.19	951	673.52	981	669.13	1011	658.68	1041	672.59
922	660.90	952	668.23	982	680.47	1012	667.51	1042	655.45
923	652.12	953	681.82	983	668.31	1013	645.20	1043	659.84
924	678.76	954	666.66	984	654.16	1014	657.74	1044	648.04
925	655.98	955	673.48	985	665.66	1015	657.26	1045	665.61
926	651.89	956	656.27	986	661.88	1016	653.04	1046	662.22
927	673.48	957	653.12	987	660.85	1017	668.26	1047	657.89
928	669.46	958	655.56	988	663.05	1018	662.29	1048	667.19
929	663.83	959	662.08	989	651.53	1019	668.45	1049	670.16
930	685.74	960	665.92	990	660.92	1020	669.82	1050	657.14

La mínima distancia máxima (642.35) ocurre cuando epoch = 902

ANEXO N°6 Coordenadas UTM del clúster 1 Con una arquitectura (1X6)

#	Este (X)	Norte(Y)
1	271,858.88	8,666,676.67
2	271,868.12	8,666,615.27
3	271,959.61	8,666,480.73
4	271,934.02	8,666,622.23
5	271,987.50	8,666,630.63
6	272,013.70	8,666,487.90
7	272,012.59	8,666,395.68
8	272,069.87	8,666,381.98
9	272,078.64	8,666,462.58
10	272,070.76	8,666,503.40
11	272,047.02	8,666,641.23
12	272,088.47	8,666,719.92
13	272,099.82	8,666,660.07
14	272,115.23	8,666,582.73
15	272,127.78	8,666,523.81
16	272,139.33	8,666,477.80
17	272,138.31	8,666,413.86
18	272,129.57	8,666,369.22
19	272,061.66	8,666,229.16
20	272,135.67	8,666,164.56
21	272,212.35	8,666,264.73
22	272,261.22	8,666,362.23
23	272,290.45	8,666,416.86
24	272,160.27	8,666,588.61
25	272,151.41	8,666,638.95

#	Este (X)	Norte(Y)
26	272,138.11	8,666,717.23
27	272,182.96	8,666,587.55
28	272,247.83	8,666,571.76
29	272,281.01	8,666,623.96
30	272,342.12	8,666,385.90
31	272,268.49	8,666,242.72
32	272,186.42	8,666,134.82
33	272,153.25	8,666,080.78
34	272,203.09	8,666,091.61
35	272,250.06	8,666,120.25
36	272,316.72	8,666,225.88
37	272,400.57	8,666,377.13
38	272,420.86	8,666,453.82
39	272,403.69	8,666,524.38
40	272,404.42	8,666,587.09
41	272,429.03	8,666,652.44
42	272,503.55	8,666,798.09
43	272,514.23	8,666,746.53
44	272,458.33	8,666,579.51
45	272,459.36	8,666,522.65
46	272,474.94	8,666,463.14
47	272,447.06	8,666,351.05
48	272,376.99	8,666,217.12
49	272,351.01	8,666,172.04
50	272,324.89	8,666,106.99

#	Este (X)	Norte(Y)
51	272,426.34	8,666,211.96
52	272,490.16	8,666,332.01
53	272,538.82	8,666,457.79
54	272,585.55	8,666,756.91
55	272,608.67	8,666,620.61
56	272,617.00	8,666,559.51
57	272,615.30	8,666,505.40
58	272,605.43	8,666,450.31
59	272,537.24	8,666,307.17
60	272,485.67	8,666,207.19
61	272,478.89	8,666,103.86
62	272,508.85	8,666,143.12
63	272,533.62	8,666,187.57
64	272,584.92	8,666,283.86
65	272,658.57	8,666,424.89
66	272,685.29	8,666,609.82
67	272,689.42	8,666,664.26
68	272,637.83	8,666,763.76
69	272,700.95	8,666,778.07
70	272,702.33	8,666,835.25
71	272,181.13	8,666,035.20
72	272,174.27	8,666,021.93
73	272,309.44	8,665,910.46
74	272,102.67	8,666,128.04

ANEXO N°7 Matriz de consistencia

Problema	Objetivos	Hipótesis	Actividades para el logro de los objetivos
Prob. general ¿Cómo sectorizar dinámicamente las esquinas de un distrito para minimizar la distancia recorrida en función del número de camiones recolectores disponibles aplicando una red de Kohonen?	**Objetivo general** Sectorizar dinámicamente las esquinas del distrito de Bellavista para minimizar la distancia recorrida en función del número de camiones recolectores disponibles aplicando una red de Kohonen	**Hipótesis General** La sectorización de esquinas usando una red de Kohonen al agrupar las esquinas más cercanas en función del número de camiones recolectores, minimiza la distancia recorrida	
Prob específicos 1. ¿Cómo construir una base de datos geográfica que permita generar un grafo y una lista de adyacencia que represente las calles distrito de Bellavista?	**Objetivos específicos** 1. Construir una base de datos geográfica para generar un grafo y una lista de adyacencia que representen las calles del distrito de Bellavista.	Hipótesis específicas 1 Una base de datos geográfica se puede utilizar para generar un grafo y una lista de adyacencia que representen las calles del distrito de Bellavista.	Extracción de coordenadas de las esquinas de Bellavista de google maps Conversión de coordenadas gps a UTM Cálculo de distancias entere esquinas Construcción del grafo: lista de distancias
2. ¿Cómo construir una red de Kohonen para sectorizar las esquinas de un distrito aprovechando al máximo la disponibilidad de los camiones recolectores?	2. Construir una red de Kohonen para sectorizar las esquinas de un distrito aprovechando al máximo la disponibilidad de los camiones recolectores.	2 Se maximiza el aprovechamiento de los camiones recolectores disponibles utilizando una red de Kohonen para sectorizar las esquinas de las manzanas del distrito.	Adaptar el algoritmo SOM de Kohonen al distrito de Bellavista Construcción del software REDSOM y del software EVALUARED
3. Cómo implementar eficiente y amigablemente para un usuario final una red de Kohonen en un medio computacional	3. Implementar una red de Kohonen, en un medio computacional, eficiente y amigable para el usuario final	3 El programa computacional que implementa la red de Kohonen que genera una sectorización dinámica es eficiente y amigable para el usuario final.	Ejecución del software EVALUARED para identificar la mejor configuración de la red. Ejecución de REDSOM con la configuración hallada en EVALUARED