Robustez y flexibilidad en los mapas autoorganizativos para ambientes no estacionarios.

Sebastián Moreno A.
Universidad Técnica Federico Santa María, Departamento de Informática,
Valparaíso, Chile, 239-0123.
smoreno@inf.utfsm.cl

Resumen

Los datos son una fuente de información de gran valor y han sido utilizados por el hombre a lo largo de su historia. Los modelos de redes neuronales artificiales son una herramienta útil para poder encontrar, clasificar y reconocer patrones dentro del conjunto de datos, lamentablemente estos modelos nos pueden llevar a una pobre generalización del espacio de entrada debido a la naturaleza cambiante del fenómeno y la existencia de datos atípicos denominados outliers.

Los outliers corresponden a ciertos puntos del conjunto de entrada que contienen características disimilares a la mayoría de la base de datos, los cuales afectan el aprendizaje de las redes neuronales. Otro problema es el de interferencia catastrófica, que permite que la red neuronal olvide lo aprendido anteriormente cuando se presenta un nuevo conjunto de datos de entrenamiento.

Esta tesis propone un modelo híbrido basado en redes SOM que aborde los problemas descritos. Esta propuesta busca la sinergia de algoritmos ya existentes, logrando un modelo robusto, flexible y dinámico, capaz de aprender la topología y particiones existentes en los datos sin ser afectado de manera significativa por la presencia de outliers y que aprenda a través del tiempo sin olvidar en forma catastrófica los datos aprendidos.

Palabras clave: Reconocimiento de patrones, Redes Neuronales Artificiales, Interferencia Catastrófica, Robustez, SOM.

1. Introducción

En la actualidad, se genera una gran cantidad de datos que son almacenados diariamente en grandes bases de datos. Lamentablemente, la mayoría de estos datos poseen características no deseadas para su análisis, como cambios importantes del fenómeno investigado en el tiempo o datos que poseen comportamiento totalmente distinto al resto de los datos, los denominados datos aberrantes u outliers.

La principal idea de almacenar estos grandes volúmenes de datos, es poder extraer la información relevante de manera tal que sirva para obtener conocimiento del fenómeno o entidad bajo estudio. Para ello se han desarrollado diversos modelos siendo una de ellos las Redes Neuronales Artificiales.

Las redes neuronales artificiales (RNA) son máquinas de aprendizaje que poseen la capacidad de aprender y generalizar a partir de un conjunto de datos, lo que nos permite reconocer y clasificar patrones, como también aproximar y predecir. Lamentablemente existen problemas que no han sido abordados en forma definitiva, como la presencia de datos aberrantes (outliers) y la naturaleza cambiante del espacio de entrada. Lo anterior impide modelar en forma precisa el espacio de entrada y, además, los RNA olvidan catastróficamente los patrones que fueron aprendidos con anterioridad [10].

En esta Tesis se propone un nuevo modelo robusto de arquitectura flexible y dinámica para las redes SOM. El objetivo es lograr que esta red sea capaz de aprender la topología y particiones existentes en los datos sin ser afectados por la presencia de datos aberrantes, y que aprenda a través del tiempo sin olvidar en forma catastrófica los datos aprendidos anteriormente, pero olvidando patrones que ya no aportan conocimiento.

El principal aporte al realizar esta Tesis es la creación de modelos con aprendizaje no supervisados basados en los mapas autoorganizativos que lograrán aprender la topología del espacio de entrada cuando

éstos sean no estacionarios y tengan presencia de datos aberrantes. Además el modelo propuesto será validado empíricamente con conjunto de datos sintéticos y reales, mediante una comparación entre las distintas variantes de los mapas autoorganizativos

El próximo capítulo trata sobre el estado del arte, que comienza explicando las redes SOM y, posteriormente, se detalla los problemas de interferencia catastrófica y robustez en las RNA. En el capítulo tres se detalla la propuesta de esta Tesis, explicando cuatro nuevos modelos para abordar los problemas comentados. En el capítulo cuatro se realizan las simulaciones pertinentes, utilizando diversos experimentos que muestran las diversas capacidades de los nuevos modelos. Finalmente el último capítulo se presentan las conclusiones pertinentes y trabajos futuros.

2. Estado del Arte

2.1. Self Organizing Maps

A principios de los años ochenta Teuvo Kohonen crea una red no supervisada, los mapas autoorganizativos [17], la cual preserva la topología de los datos proyectándolos a una malla de dimensión más baja, siendo éste uno de los motivos de su popularidad.

La arquitectura de la red SOM consiste en dos capas. La capa de entrada que es unidimensional de tamaño N, debido a la dimensión del vector de datos ($\underline{x} = [x_1, x_2, \cdots x_N]^T$), la cual traspasa la información obtenida a todas las neuronas de la siguiente capa. La segunda capa es donde se realiza el procesamiento, esta capa contiene k neuronas conectadas de cierta manera formando una malla que usualmente tiene una estructura bi-dimensional, pero se puede dar el caso que sea una malla tridimensional o unidimensional.

Una vez determinado el número de neuronas de la malla principal (k), el cual es un problema abierto, se procede a la inicialización de los pesos de las neuronas, las cuales se pueden realizar en forma aleatoria en torno a cierto punto.

El entrenamiento de una red SOM consiste en una serie de pasos que comienza cuando se presenta un vector de entrada a la red y se procede a obtener la neurona mas cercana a este denominado Best Matching Unit (BMU) (ver ecuación 1). El resultado obtenido por la red depende de la distancia que se seleccione para encontrar la BMU, comúnmente la medida más utilizada es la distancia euclidiana.

$$BMU = \underline{w}_{bmu} = \arg \min_{i=1\ldots k} d(\underline{x}, \underline{w}_i) \tag{1}$$

Una vez que la neurona ganadora es seleccionada se procede a la actualización de los pesos de las neuronas mediante la fórmula de adaptación:

$$\underline{w}_i(t+1) = \underline{w}_i(t) + \alpha(t) h_{bmu}(\underline{w}_i, t)[\underline{x}(t) - \underline{w}_i(t)] \quad i = 1 \ldots K \tag{2}$$

donde t corresponde a la iteración en el tiempo, $x(t)$ el dato presentado en la iteración t, $w_i(t)$ el peso del prototipo v_i en la iteración t y finalmente $\alpha(t)$ y $h_{bmu}(\underline{w}_i, t)$ son la tasa de aprendizaje y función de vecindad, respectivamente, las cuales se explicarán más adelante.

La estimación del vector de pesos $\underline{w}$ se llama aprendizaje competitivo suave debido a que cuando se presenta una instancia de la muestra, se actualiza el peso de la neurona ganadora y sus vecinas, las cuales son definidas por la malla y la función $h_{bmu}(\underline{w}_i, t)$. La actualización del peso de una neurona corresponde al peso anterior más la diferencia existente entre el dato de entrada y el peso actual, todo esto multiplicado por los factores $\alpha(t)$ y $h_{bmu}(\underline{w}_i, t)$. Estos dos factores evitan que todas las neuronas aprendan de igual manera el punto, ya que el máximo valor que toma la multiplicación de estos factores es 1 y sólo ocurre para la BMU, mientras que el factor es menor para el resto de las neuronas.

La tasa de aprendizaje $\alpha(t)$ es una función monótonamente decreciente con el tiempo, cuyo valor varía entre 0 y 1. Es importante que esta función decrezca con el tiempo ya que con ello asegura la convergencia del algoritmo de entrenamiento de las SOM. Algunas funciones típicas de la tasa aprendizaje son Exponencial: $\alpha(t) = \alpha_0 \left(\frac{\alpha_f}{\alpha_0}\right)^{t/t_\alpha}$ y Lineal: $\alpha(t) = \alpha_0 + (\alpha_f - \alpha_0)\frac{t}{t_\alpha}$. El valor inicial de la tasa de aprendizaje en la fase de sintonización es cercano a uno para que las neuronas se adapten rápidamente a los patrones. A medida que avanzan las iteraciones es necesario decrecer la tasa de aprendizaje para que las neuronas no ganadoras no se acerquen tanto a la instancia presentada, permitiendo con esto la dispersión de las neuronas en los datos [17]. Se recomienda que α_0 sea menor que 1, mientras que α_f sea aproximadamente 0,01.

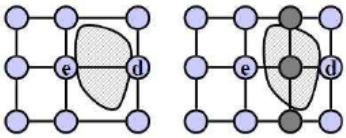


Figura 1: **Proceso de crecimiento de la red GSOM**

La función de vecindad de las redes SOM determina cuáles son las neuronas vecinas que deben ser actualizadas cada vez que se presenta un dato en el entrenamiento. Esta función está definida en función de la neurona ganadora ($BMU = \underline{w}_{bmu}$), la neurona que se está actualizando ($\underline{w}_i$) y el número de iteraciones t. El máximo valor que puede tomar la función de vecindad es 1 y se da cuando $\underline{w}_i = \underline{w}_{bmu}$, mientras que el mínimo valor podría ser -1, dependiendo de la función que se escoja. El valor negativo implica que en lugar de acercar $\underline{w}_i$ al dato presentado éste se aleja, logrando la dispersión de ciertas neuronas con respecto a la neurona ganadora. Una función de vecindad comúnmente utilizada es la gaussiana $h_{bmu}(\underline{w}_j, t) = \exp\left(-\frac{\|\underline{w}_j - \underline{w}_{bmu}\|^2}{2R^2(t)}\right)$

2.1.1. Variantes de Arquitecturas Flexibles de redes SOM

Una de las desventajas de las redes SOM es su arquitectura no flexible, es decir, una vez seleccionada la cantidad de neuronas y la forma de la malla es imposible realizar algún cambio. Es por esta razón que se han creado variantes más flexibles que se basan en este tipo de redes, uno de ellos es el *Growing Self Organizing Maps* (GSOM) [4], que consiste en una malla con la capacidad de crecer agregando nuevas neuronas si el modelo lo estima conveniente, logrando con ello un mejor modelamiento del espacio de entrada. La red GSOM comienza con 4 neuronas formando una malla cuadradra, que una vez entrenada procede a calcular el error de cuantización para cada neurona, dado por:

$$qe_j = \sum_{\underline{x}_i \in PV_j} \|\underline{x}_i - \underline{w}_j\| \quad \forall j \in 1, \ldots, k \tag{3}$$

donde PV_j corresponde al polígono de Voronoi definido por la neurona j:

$$PV_j = \{\underline{x} \in \mathbb{R}^n | s(\underline{x}) = v_j\} \tag{4}$$

Luego se compara el error de cuantización de cada neurona con qe_0, que corresponde al error de cuantización de una neurona raíz v_0 localizada en el centro de los datos. Si $qe_j > \gamma \cdot qe_0$ para alguna neurona j, donde γ es un valor entre 0 y 1, entonces la representación de los datos no es óptima por lo cual es necesario que el mapa crezca y se entrene de nuevo; para ello se selecciona la neurona con mayor error de cuantización (e) y su neurona vecina más disimilar (d), tanto e como d se calculan como:

$$e = arg \max_{j=1..k} \left\{ \sum_{\underline{x}_i \in PV_j} \|\underline{x}_i - \underline{w}_j\| \right\} \qquad d = arg \max_{j \in \mathcal{N}_e} \left(\|\underline{w}_e - \underline{w}_j\| \right) \tag{5}$$

donde $\mathcal{N}_e$ es el conjunto de neuronas vecinas de v_e.

Una fila o una columna es insertada entre e y d y sus neuronas son inicializadas como el promedio de sus respectivos vecinos como se aprecia en la figura 1. Luego se procede a entrenar el mapa. Este proceso se repite hasta que $qe_j < \gamma \cdot qe_0 \ \forall j \in 1, \ldots, k$.

Otra desventaja de la rigidez de la arquitectura de la red SOM es la imposibilidad de modelar datos jerarquizados, lo cual fue solucionado por A. Rauber, D. Merkl y M. Dittenbach en el paper *The Growing Hierarchical Self-Organizing Map: Exploratory Analysis of High-Dimensional Data* [7]. Esta red consiste un conjunto de mallas GSOM las cuales están conectadas en forma jerárquica en una estructura de árbol. En un principio se crea una neurona superior (v_0) y debajo de ella un mapa GSOM, la cual se entrena tal como se explicó en la sección anterior, pero el mapa deja de crecer si $MQE_m < \gamma \cdot qe_u$ donde qe_u es el error de cuantización de la neurona en el nivel superior que se calcula con la ecuación 3 y MQE_m corresponde al promedio de los errores de cuantización en el nivel m dado por

$$MQE_m = \frac{1}{n_\mu} \sum_{i \in \mu} qe_i, \qquad n_\mu = |\mu| \tag{6}$$

donde μ corresponde al conjunto de neuronas hijas de qe_u y $|\mu|$ es el número de neuronas hijas de qe_u.

Por ejemplo si se analiza el primer mapa éste dejará de crecer si $MQE_1 < \gamma \cdot qe_0$.

Una vez finalizada la etapa de crecimiento se procede a verificar qué neuronas no están representando bien los datos para crear una nueva malla bajo de ellas, para ello se utiliza el criterio $qe_i < \tau \cdot qe_0$, donde cada neurona que no cumpla con la relación crea un nuevo mapa GSOM debajo de v_i con los datos que modela y se procede a entrenar tal como se explicá anteriormente.

El entrenamiento finaliza una vez que todos los mapas son entrenados y no es necesario crear más jerarquías.

2.2. Dilema de Estabilidad-Plasticidad e Interferencia Catastrófica

El problema de interferencia se ha estudiado en el ser humano y no es tan drástico como para olvidar completamente lo aprendido anteriormente, aunque si existe un olvido gradual cuando se le enseñan nuevas cosas. Barnes y Underwood en [3] determinaron que existía una interferencia gradual y no catastrófica en el ser humano. Esto no sucede en las redes neuronales donde la interferencia es catastrófica debido a que la red neuronal puede olvidar completamente los datos aprendidos anteriormente cuando se le presenta un nuevo conjunto de datos, tal como lo muestra Ratcliff [21].

Existen diversas formas de solucionar el problema descrito anteriormente. Una de ellas corresponde a modificar los datos de entrada, creando un nuevo conjunto que contengan parte o la totalidad de los datos presentados anteriormente cada vez que se presenta datos de entrada (ver [9]).

Otra forma de enfrentar éste problema es tratar de reducir el traslape de los nuevos datos con los antiguos. Uno de los primeros algoritmos de este tipo fue creado por Kortge [1], otro algoritmo de este tipo son las redes ART [6].

Otros modelos denominados convolución-correlación pueden aprender la información de manera secuencial y a la vez generalizar cuando se presentan nuevos datos. Un ejemplo de este tipo de modelo es CHARM [18].

2.2.1. Dilema de Estabilidad-Plasticidad

El dilema de estabilidad-plasticidad concierne a cómo las representaciones internas se deben mantener estable ante la presencia de efectos erosivos que provengan de fluctuaciones de conductas irrelevantes y aún así adaptarse rápidamente a las fluctuaciones del ambiente que son claves para la supervivencia [11]. Si lo vemos desde el punto de vista de redes neuronales no supervisadas, significa que un modelo debe ser estable ante la presencia de datos atípicos y a la vez debe tener la suficiente plasticidad para adaptarse a la presencia de nuevos patrones en los datos.

Muchos analistas de redes han ignorado este dilema, considerando que el comportamiento de los datos no cambian a través del tiempo [12]. En particular, la comunidad de redes SOM también ha ignorado este dilema, debido a la estructura no plástica de este tipo de red y que, además, ante la presencia de nuevos datos sufre el problema de interferencia catastrófica. Aunque existen otros modelos que tienen cierta plasticidad, como GSOM y GHSOM, los cuales fueron explicados anteriormente, éstos tampoco están acorde al dilema de estabilidad-plasticidad debido a que en presencia de datos novedosos pueden modificar su estructura para ajustarse a ellos, pero, aun así, ésto puede conllevar a un olvido catastrófico de lo aprendido anteriormente.

2.3. Métodos Robustos

La estadística robusta es una disciplina que se preocupa en estudiar la estabilidad de los procedimientos estadísticos. Esta teoría investiga el efecto que tienen dichos procedimientos, cuando ocurren desviaciones en los supuestos del modelo afectando, por ejemplo, las estimaciones de los parámetros. Además se preocupa en desarrollar nuevos métodos de estimación que mitiguen el impacto de dichas desviaciones.

En la estadística paramétrica se asumen 3 supuestos: los datos provienen de alguna distribución de probabilidad; cada dato proviene de una distribución independiente y las distribuciones de donde provienen los datos son idénticas. Cada uno de estos tres supuestos puede tener efectos no deseados en la estimación de los parámetros; por el contrario, si se cumpliera estos tres supuestos es de esperar que los métodos

se comporten relativamente bien bajo pequeñas variaciones de los datos, pero desafortunadamente estas pequeñas desviaciones en los supuestos pueden provocar efectos desastrosos en los procedimientos estadísticos paramétricos [24].

No existe una definición formal para los outliers, pero Hampel [14] los definió como datos que se alejan fuertemente del modelo estocástico subyacente sugerido por la mayoría de los datos. Uno de los problemas de los outliers es que puede tener una gran influencia en algunos métodos de estimación de parámetros, lo que arruinaría completamente el método de estimación a utilizar. Lamentablemente no es posible filtrar este tipo de datos si se detectaran debido a que se desconoce el origen del por qué caen en esta categoría.

2.3.1. Funcional estadístico

Sea $X_1, ..., X_n$ una muestra aleatoria de una población con distribución de probabilidad P_θ y función de distribución F, donde $P_\theta \in \mathbb{P} = \{P_\theta : \theta \in \Theta \subseteq \mathbb{R}^p\}$. Entonces θ puede ser tratado como un funcional $\theta = T(P)$ definido sobre $\mathbb{P}$. Intuitivamente un estimador de θ basado en las observaciones $X_1, ..., X_n$ es $T(P_n)$, donde P_n es la distribución de probabilidad empírica de $X_1, ..., X_n$ dada por $P_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i} \Rightarrow P_n(A) = \frac{1}{n}\sum_{i=1}^{n} I[X_i \in A]$ donde δ_{x_i} es una distribución que asigna probabilidad 1 en x_i.

La distribución de función empírica (F_n) perteneciente a P_n corresponde a $F_n(x) = P_n((-\infty, x]) = \frac{1}{n}\sum_{i=1}^{n} I[X_i \leq x], \qquad x \in \mathbb{R}$

Una de las propiedades que debe cumplir un estimador es que sea Fisher consistente, introducido por R. A. Fisher en el año 1922. Se dice que un estimador es Fisher consistente si $T(P_\theta) = \theta$ para todo $\theta \in \Theta$, es decir, que el parámetro estimado por el funcional $T(\cdot)$ sea el correcto para la función de distribución de probabilidad P_θ.

2.3.2. Función de Influencia y Sensibilidad al error crítico

La función de influencia es una medida de robustez local, la cual describe el efecto de una contaminación infinitesimal en el punto x sobre la estimación dada por $T(\cdot)$.

La derivada de Gâteaux del funcional T en la distribución P en la dirección de la distribución de Dirac $\delta_x,\ x \in \chi$ es llamada la función de influencia de T en P, es decir,

$$IF(x; T, P) = \lim_{t \to 0} \frac{T((1-t)P + t\delta_x) - T(P)}{t} = \frac{d}{dt} T((1-t)P + t\delta_x)|_{t=0} \tag{7}$$

La función de influencia es una medida local, es decir, ve la influencia infinitesimal de x sobre el funcional T.

La *sensibilidad al error crítico* (gross error sensitivity) es una medida global de robustez, introducido por Hampel en el año 1968 [13] y que está basada en la función de influencia. La sensibilidad del error crítico de un funcional T en la distribución P es

$$\gamma_\mu(T, P) = \sup_{x \in \chi} \|IF(x; T, P)\| \tag{8}$$

donde $\|\cdot\|$ es la norma euclidiana.

Esta medida permite determinar la mayor influencia que puede ejercer un dato x en el funcional T, por lo tanto es de esperar que esta influencia no sea infinita, si es así, se dirá que el funcional T es *B-robusto.*

Un ejemplo de un estimador no B-robusto es la función promedio, debido a que sun función de influencia esta dada por $IF(x; T, P) = -T(P) + X = X - E[X]$ y la sensibilidad del error crítico está dado por $\gamma_\mu(T, P) = \sup_x (X - E[X]) = \infty$.

2.3.3. M-Estimadores

La clase de M-estimadores fue propuesto por Huber en el año 1964 para determinar los parámetros de localización de una distribución y sus propiedades han sido estudiados en [16]. Un M-estimador T_n^M se define

como la solución del problema de minimización de

$$T_n^M = \arg\min_{\theta\in\Theta} \sum_{i=1}^{n} \rho(X_i, \theta) \tag{9}$$

donde $\rho : \chi \times \Theta \to \mathbb{R}_0^+$. Si ρ es diferenciable en θ obteniendo una derivada continua definida como $\psi(\cdot, \theta) = \frac{d}{d\theta}\rho(\cdot, \theta)$, entonces $T_n^M \in \Theta$ corresponde a una de las raíces de la ecuación $\sum_{i=1}^{n} \psi(X_i, \theta) = 0,\ \theta \in \Theta$. Además, se puede observar que el M-funcional, es decir, el funcional estadístico correspondiente a la expresión anterior es definido como la solución de la siguiente ecuación $\int_\chi \psi(x, T(P))dP(x) = 0, \qquad T(P) \in \Theta$ donde χ es el espacio de los datos.

Un ejemplo de estimador M es el de máxima verosimilitud, donde $\rho(x, \theta) = -lnf(x, \theta)$ y $\psi(x, \theta) = s(x, \theta)$ es la función score.

Para poder estudiar la influencia infinitesimal de un dato x en un estimador se utiliza la función de influencia, que en el caso de los M-estimadores, la función de influencia viene dada por [14]:

$$IF(x; T, P) = \frac{\psi(x, T(P))}{-\int_\chi \psi'(y, T(P))dP(y)} \tag{10}$$

donde $\psi'(\cdot, T(P)) = [\frac{d}{d\theta}\psi(\cdot, \theta)]_{\theta=T(P)}$.

3. Propuesta

3.1. Motivación

En una gran cantidad de aplicaciones reales, los ambientes de donde la información es extraída son complejos y no-estacionarios. Lamentablemente diversos modelos no supervisados no logran una correcta generalización del espacio de entrada cuando se presentan ambientes con estas características siendo afectados por la presencia de datos aberrantes y olvidando en forma catastrófica lo aprendido en épocas anteriores. Estas falencias motivan a crear nuevos modelos que no sean afectados tan drásticamente por este tipo de ambientes.

3.2. No robustez de la red SOM

En esta sección se mostrará empíricamente que la regla de aprendizaje del algoritmo SOM no es B-robusta. Lamentablemente al no existir una función de energía que minimizar [8], tampoco existe un funcional asociado, lo que dificulta la demostración.

3.2.1. Influencia de un dato en el aprendizaje de la red SOM

Al presentar un vector ($\underline{x}$) cuyo comportamiento es diferente a la mayoría del conjunto de datos, éste se considerará como outlier o dato aberrante, si la distancia de la neurona ganadora v_{bmu} a $\underline{x}$ es de gran magnitud. La gran magnitud de las distancia del dato al BMU afectará el proceso de aprendizaje, debido a que la neurona ganadora y sus vecinas tratarán de modelar el outlier moviendo tanto al prototipo ganador como al grupo neuronas vecinas hacia $\underline{x}$, apartándose del resto de las observaciones.

El impacto de un dato en la regla de aprendizaje de la red SOM, será medido como *el supremo del salto de aprendizaje* dado por:

$$\sup_{\underline{x}\in\mathbb{R}^n} (\alpha(t)h_{bmu}(\underline{w}_j, t)[\underline{x}(t) - \underline{w}_j(t)]) \qquad j = 1..K \tag{11}$$

En el caso de la red SOM el supremo del salto de aprendizaje es no acotado, lo que implica que el aprendizaje de la red SOM no es B-robusto, puesto que es afectado por la presencia de datos aberrantes.

3.3. Modelo Robust Self Organizing Map

En esta sección, se propone un nuevo modelo denominado Robust Self Organizing Map (RSOM) [2], el cual incluye una regla de aprendizaje robusta que disminuye la influencia de los datos outliers en el aprendizaje.

3.3.1. Aprendizaje de la topología

Para disminuir la influencia que ejercen los datos aberrantes en la regla de aprendizaje de la red SOM dado por la ecuación 2, es necesario utilizar una función robusta $\psi : \chi \times \mathbb{W} \to \mathbb{R}, \underline{w}_i \in \mathbb{W}$ en el aprendizaje, para acotar el impacto que podría generar vectores aberrantes, dada por:

$$\underline{w}_j(t+1) = \underline{w}_j(t) + \alpha(t) h_{bmu}(\underline{w}_j, t) \psi_c(\underline{x}(t) - \underline{w}_j(t)) \quad j = 1 \dots K \tag{12}$$

cuya constante de forma de la función $\psi_c(\cdot)$ es $c = r \cdot s_j(t)$, donde r es una constante y $s_j(t)$ es una estimación robusta de la varianza de los datos modelados por el prototipo v_j, que se puede calcular usando una variante de la función MEDA dada por

$$s_j(t) = \frac{1}{\phi^{-1}(3/4)} \; median_j\{|\alpha(t) h_{bmu}(\underline{w}_j, t)[\underline{x}(t) - \underline{w}_j(t)] - median_j(\alpha(t) h_{bmu}(\underline{w}_j, t)[\underline{x}(t) - \underline{w}_j(t)])|\} \tag{13}$$

donde $\phi^{-1}(p)$ es la inversa de la función de distribución acumulativa de una Gaussiana estándar en la probabilidad p. La constante $\frac{1}{\phi^{-1}(3/4)} \approx 1{,}483$ es necesaria para que la estimación de $s_j(t)$ sea Fisher consistente cuando la data tiene un comportamiento de una distribución Gaussiana.

3.4. Modelo Robust Growing Hierarchical Self Organizing Map

El modelo GHSOM, explicado en la sección 2.1.1, es un modelo de arquitectura flexible, que utiliza en cada mapa la misma regla de aprendizaje que la red SOM, por lo tanto, el modelo GHSOM no es B-robusto debido a que la influencia de un dato en el entrenamiento del prototipo puede ser arbitrariamente grande.

Si se presentara un dato outlier en el entrenamiento de la red GHSOM, el error MQE_m (ver ecuación 6) será de gran magnitud, lo que llevara a aumentar el número de neuronas del mapa y posteriormente puede suceder que se cree un mapa de 4 neuronas para modelar el outlier.

Por este motivo es necesario robustecer la red GHSOM, lo que genera una nueva red denominada robust growing hierarchical self organizing map (RGHSOM) [20], que consiste en una arquitectura tipo árbol jerárquico donde cada nodo consiste en una red RSOM descrita en la sección 3.3.

3.4.1. Posicionamiento de v_0 y creación del modelo base

En la red GHSOM es necesario establecer una neurona v_0 en el medio de los datos a modelar, por lo cual la localización de v_0 debe estar dada por la minimización de un funcional robusto dado por

$$RL(\underline{w}_0, s_0) = \sum_{\underline{x}_i \in \mathcal{I}} \rho\left(\frac{\underline{x}_i - \underline{w}_0}{s_0}\right) \quad \mathcal{I} \neq \phi$$

donde $\mathcal{I}$ corresponde al conjunto de datos a modelar. En esta tesis se utiliza la mediana para la localización de v_0.

Una vez situada v_0 se procede a construir un mapa de dimensión 2×2 donde cada neurona puede ser inicializada en forma aleatoria o determinista con media w_0 y varianza s_0^2.

3.4.2. Entrenamiento del mapa

El entrenamiento de los mapas de esta red se realizan en forma robusta tal como se explico en la sección 3.3.

3.4.3. Medida de calidad

Para determinar si una neurona modela en forma correcta los datos pertenecientes a su polígono de Voronoi, se utiliza una versión robusta del error de cuantización que se utiliza en el modelo GHSOM (ver ecuación 3), denominado error de cuantización robusto (rqe), dado para la neurona v_j por:

$$rqe_j = \sum_{\underline{x}_i \in \mathcal{C}_{v_j}} \rho\left(\frac{\underline{x}_i - \underline{w}_j}{s_j}\right), \quad \mathcal{C}_{v_j} \neq \phi \tag{14}$$

donde $\mathcal{C}_{v_j}$ corresponden a todos los datos que pertenecen al polígono de Voronoi determinado por la neurona v_j perteneciente al mapa $\mathcal{W}$.

3.4.4. Crecimiento del mapa

La etapa de crecimiento consiste en agregar una fila o columna de neuronas al mapa que se está entrenando en caso que se determine que no modela en forma correcta los datos que le corresponden.

Para determinar si el mapa $\mathcal{W}$ no modela en forma correcta su conjunto de datos, se calcula el error promedio de cuantización robusta (RMQE) determinado por

$$RMQE = \frac{1}{W} \sum_{\underline{w}_j \in \mathcal{W}} rqe_j \tag{15}$$

donde $W = |\mathcal{W}|$ es el número de neuronas que existen en el mapa $\mathcal{W}$ y rqe_j es determinado por la formula 14.

Sea γ un factor entre 0 y 1 y rqe_k el error de cuantización robusto de la neurona padre del mapa $\mathcal{W}$. Si $RMQE > \gamma \cdot rqe_k$, entonces el mapa no está modelando en forma correcta los datos, por lo tanto es necesario agregar nuevas neuronas, para lo cual se calcula v_e que corresponde a la neurona con mayor rqe del mapa $\mathcal{W}$ y v_d que es el vecino más lejano de v_e determinadas por

$$v_e = arg \max_{v_j \in \mathcal{W}} (rqe_j \in \mathcal{W}) \quad v_d = arg \max_{v_j \in \mathcal{N}_e} \left(\|\underline{w}_e - \underline{w}_j\| \right) \tag{16}$$

donde $\mathcal{N}_e$ es el conjunto de neuronas vecinas de v_e y $\underline{w}_e = w(v_e)$, además se define $\underline{w}_d = w(v_d)$. Una vez determinadas v_e y v_d se agrega una columna o fila de neuronas entre v_e y v_d donde sus pesos iniciales corresponden al promedio de las neuronas vecinas, tal como se explicó en la sección 2.1.1.

3.4.5. Jerarquización del modelo

Si un mapa está modelando en forma correcta su conjunto de datos, pero existe alguna neurona que necesite una mayor representación del espacio que modela, es necesario crear un nuevo mapa que se dedique exclusivamente a modelar los datos que pertenecen al polígono de Voronoi de la neurona.

Sea rqe_0 el error de cuantización robusto de la neurona raíz. Las neuronas v_j que no cumplan con la condición

$$rqe_j < \tau \cdot rqe_0 \tag{17}$$

no están modelando en forma correcta su conjunto de datos, por lo cual crearán un nuevo mapa de dimensión 2×2, abajo de ellas cuyos pesos iniciales se determinan en forma aleatoria o determinista utilizando como centro a v_j.

3.5. Modelo Flexible Architecture of Self Organizing Map

El modelo Flexible Architecture of Self Organizing Map (FASOM) [22] corresponde a K mapas GSOM que se adaptan al espacio de entrada. Este modelo tiene la capacidad de aprender nuevos datos a través del tiempo sin ser afectado por el problema de interferencia catastrófica. Para ello cuando se encuentra en presencia de nuevos conjuntos de datos que no han sido modelados, crea diversos mapas para modelar estos nuevos conjuntos, además tiene la capacidad de ir olvidando gradualmente los datos que ya no se utilizan.

El entrenamiento de este nuevo modelo se separa en dos fases. En la primera se crean los primeros K mapas GSOM y se modela el espacio de entrada que se presenta. La segunda fase ocurre cuando un nuevo conjunto de datos es presentado y requiere ser modelado.

3.5.1. Construcción del modelo base

En esta etapa se buscan clusters en el espacio de entrada, utilizando una variación del algoritmo *K-means* [15] y posteriormente se unirán los clusters más cercanos utilizando el algoritmo *Single-linkage* [5].

Una vez determinado los K cluster y sus respectivos centros, se crean K mapas de dimensión 2×2 sobre los centroides $\overline{z}_r$ $r = 1..K$.

Creados los mapas se procede al entrenamiento, en este proceso cuando se presenta un dato $\underline{x}$ se procede a encontrar el *Best Matching Map* (BMM), que corresponde al mapa que contiene la neurona más cercana al dato presentado. Sea $\mathcal{M}_k$ el conjunto de todas las neuronas que pertenecen al mapa $\mathcal{W}_k$. La neurona ganadora

$\underline{w}_{\mathcal{W}_k}$, $k = 1, ..., K$, correspondiente a $\underline{x}$ para cada uno de los K mapas que se utilizan son detectadas, luego el mapa que contiene a la neurona ganadora más cercana a $\underline{x}$ es el BMM cuyo índice se determina mediante

$$\zeta = \arg \min_{k=1..K} \left\{ \|\underline{x} - \underline{w}_{\mathcal{W}_k}\|, \underline{w}_{\mathcal{W}_k} \in \mathcal{M}_k \right\} \tag{18}$$

Una vez detectado el BMM, se actualizan los pesos de todas las neuronas pertenecientes al mapa $\mathcal{W}_\zeta$ mediante una adaptación de la regla de aprendizaje de las redes SOM dada por:

$$\underline{w}_j(t+1) = \underline{w}_j(t) + \alpha(t) h_{bmu}(\underline{w}_j, t)[\underline{x}(t) - \underline{w}_j(t)] \quad \forall \underline{w}_j \in \mathcal{W}_\zeta \tag{19}$$

donde $\underline{w}_{bmu}$ es la neurona ganadora del mapa ζ cuando se presenta el dato $\underline{x}$.

Para medir la calidad de representación de los datos de la neurona j, perteneciente al mapa k, se utiliza una variante del error de cuantización dado por:

$$qe_j^{[k]} = \sum_{\underline{x} \in \mathcal{C}(\underline{w}_j)} \|\underline{x} - \underline{w}_j\| \quad \underline{w}_j \in \mathcal{W}_k \tag{20}$$

donde $\mathcal{C}(\underline{w}_j)$ corresponde al conjunto de datos que modela la neurona v_j.

Si se define $qe_0^{[k]}$ como el error de cuantización del cluster k, donde la neurona $v_0^{[k]}$ corresponde a una neurona situada en el centroide del cluster, entonces todas las neuronas que no satisfagan $qe_j^k < \gamma \cdot qe_0^k$ necesitan una mejor representación por lo cual es necesario aumentar el número de neuronas tal como se realiza en el algoritmo GSOM de la sección 2.1.1.

3.5.2. Aprendizaje del ambiente cambiante

Al ser un modelo flexible, el FASOM tiene la ventaja de poder modelar nuevos datos creando nuevos mapas e inclusive disminuir los mapas ya existentes hasta eliminarlo si éste ya no modela datos.

Cuando se presenta un nuevo conjunto de datos a la red se verifica si su topología puede ser modelada utilizando el modelo actual $\mathcal{H}_{T-1}$, para ello se calcula la influencia de las muestras $\underline{x}$ en el modelo $\mathcal{H}_{T-1}$, calculando $\delta(\underline{x}, \mathcal{H}_{T-1}) = \|\underline{x} - \underline{w}_{\mathcal{W}_\zeta}(t)\|$, donde $\underline{w}_{\mathcal{W}_\zeta}(t)$ corresponde a la BMU del BMM ζ para el dato $\underline{x}$. Utilizando la medida definida se procede a calcular $\mathcal{I}_T^{[new]}$, que corresponde al conjunto de todos los datos en el tiempo T cuya influencia en el modelo $\mathcal{H}_{T-1}$ es mayor a cierto umbral ϵ, es decir, $\delta(\underline{x}, \mathcal{H}_{T-1}) > \epsilon$.

Si $|\mathcal{I}_T^{[new]}| > N_{min}$ donde $|\mathcal{I}_T^{[new]}|$ es la cantidad de datos novedosos y N_{min} es el número mínimo de datos requeridos para crear un mapa, entonces se está en presencia de un conjunto de datos novedosos los que deberán ser modelados a través de un nuevo modelo $\mathcal{H}_T^{[new]}$ que se creará tal como se explicó en la primera fase. Este nuevo conjunto de mapas se unirá al ya existente, obteniendo $\mathcal{H}_T = \mathcal{H}_{T-1} \cup \mathcal{H}_T^{[new]}$.

El aprendizaje del modelo $\mathcal{H}_T$ se realiza de la misma manera que se entrenó el modelo $\mathcal{H}_1$.

Debido a que el ambiente del espacio de entrada no es estacionario, es decir, los clusters pueden ir modificando su comportamiento cabiendo la posibilidad de desaparecer completamente, es necesario introducir una técnica de olvido al modelo. La estrategia consiste en olvidar gradualmente los datos modelados por mapas que no hayan sido modificados en su arquitectura durante el entrenamiento. Este olvido se realiza contrayendo las neuronas del mapa hacia su respectivo centroide, y en caso de que la malla sea pequeña, está se contrae podando una fila o columna de neuronas, este proceso se puede repetir hasta que el mapa sea una pura neurona. Una vez finalizada esta etapa se reentrenarán los mapas que olvidaron parte de sus datos. Éste entrenamiento es realizado por si algún mapa esta modelando en forma correcta sus datos, con lo cual no modificará su estructura.

El centroide del mapa $\overline{w}_k$ esta dado por el promedio de todas las neuronas pertenecientes al mapa k, es decir, $\overline{w}_k = \frac{1}{|\mathcal{W}_k|} \sum_{j=1}^{|W_k|} \underline{w}_j, \quad \underline{w}_j \in \mathcal{W}_k$, donde $|\mathcal{W}_k|$ es el número de neuronas del mapa k.

El mapa k olvidará gradualmente los datos aplicando la siguiente regla de olvido

$$\underline{w}_j(T) = \underline{w}_j(T-1) + \lambda[\underline{w}_j(T-1) - \overline{w}_k] \quad \forall \underline{w}_j \in \mathcal{W}_k \tag{21}$$

donde $\lambda \in [0, 1]$ corresponde a la tasa de olvido. $\underline{w}_j(T-1)$ y $\underline{w}_j(T)$ son los pesos de las neuronas al final de la etapa $T-1$ y al comienzo de la etapa T respectivamente.

Una vez realizado el olvido de las neuronas se procede a verificar si las neuronas de algún mapa k se encuentran muy cercanas entre sí para proceder a podarlas. Si la malla de neuronas tiene una forma

rectangular, se procede a buscar las dos filas o columnas que estén más cercanas entre sí calculando ν, dada por

$$\nu = \underset{e=1..N_r-1;\ d=1..N_c-1}{\arg\min} \left(\frac{1}{N_c} \sum_{j=1}^{N_c} \|\underline{w}_{(e,j)} - \underline{w}_{(e+1,j)}\|, \frac{1}{N_r} \sum_{i=1}^{N_r} \|\underline{w}_{(i,d)} - \underline{w}_{(i,d+1)}\| \right) \tag{22}$$

donde $\underline{w}_{(i,j)}$ es el peso de la unidad en la posición (i, j) en el mapa k y N_r y N_c son el número de filas y columnas respectivamente. Una vez calculado ν se procede a calcular la distancia existente entre ν y $\nu + 1$ que corresponde a la distancia entre las dos filas o columnas más cercanas determinado por

$$\varrho = \frac{1}{N_\nu} \sum_{j=1}^{N_\nu} \|\underline{w}_{\nu \cdot j} - \underline{w}_{\nu+1 \cdot j}\| \tag{23}$$

donde N_ν corresponde al número de neuronas de ν y $\underline{w}_{\nu \cdot j}$ es el peso de $\underline{w}_{\nu,j}$ o $\underline{w}_{j,\nu}$ según sea el caso.

Si el criterio $\varrho < \beta$, donde β es un factor de olvido, es satisfecho, entonces una columna o fila de neuronas es insertada entre ν y $\nu + 1$ donde sus pesos son inicializados como el promedio de sus respectivos vecinos; luego las columnas o filas ν y $\nu + 1$ se eliminan. El mapa se contrae en forma iterativa hasta que ninguna fila o columna satisfaga el criterio o el mapa sea eliminado.

3.6. Modelo RoFlex-HSOM

Esta red denominada Robust and Flexible model of Hierarchical Self Organizing Map, es la propuesta más importante de esta tesis, la cual puede ser considerada como una mezcla de los modelos RGHSOM y FASOM. El RoFlex-HSOM [23] consiste en una arquitectura jerárquica en forma de árbol donde cada mapa es un modelo GSOM robusto que se adapta automáticamente a los datos presentados, con la capacidad de modelar datos cambiantes en el tiempo.

Este modelo tiene la capacidad de detectar nuevos conjuntos de datos cuando se les presentan y crear nuevos mapas para modelarlos sin olvidar en forma catastrófica el espacio aprendido anteriormente. Otra capacidad importante de este modelo es que permite olvidar gradualmente los datos aprendidos en épocas anteriores achicando sus mallas y podando sus neuronas. Por último, se incorporan estrategias robustas, para realizar un aprendizaje que no se vea influenciado por la presencia de outliers.

El aprendizaje se separa en dos partes, la primera corresponde a la primera vez que se presentan los datos, donde se realiza la construcción del modelo base y se procede a su entrenamiento. La segunda parte corresponde a dos fases, el entrenamiento y adaptación de los mapas cuando se presentan datos novedosos y la fase de olvido de los mapas. La descripción detallada de este algoritmo se realizará en las siguientes secciones.

Al ser un modelo que mezcla diversas partes de modelos explicados anteriormente, no se referenciará ni duplicará la información.

3.6.1. Construcción del modelo base

La construcción del modelo base comienza con el posicionamiento de la neurona v_0, la cual al ser un modelo jerárquico y robusto es necesario establecerla en el medio del conjunto de datos, cuya posición está dada por la minimización de un funcional robusto dado por $RL(\underline{w}_0, s_0) = \sum_{\underline{x}_i \in I} \rho\left(\frac{\underline{x}_i - \underline{w}_0}{s_0}\right) \quad I \neq \phi$

Una vez situada v_0 se procede a construir un mapa de dimensión 2x2 donde cada neurona puede ser inicializada en forma aleatoria o determinista con media $\underline{w}_0$ y varianza s_0^2.

Una vez posicionada la neurona v_0 se procede al aprendizaje de los datos el que se realiza en forma robusta, tal como se explicó en el modelo RSOM.

Para decidir si un mapa necesita crecer o introducir un nuevo nivel de descripción al modelo, es necesario cuantificar la calidad de adaptación del modelo al espacio de entrada. La calidad de adaptación de la neurona $v_r \in \mathcal{H}_T$ se mide utilizando el error robusto de cuantización (rqe_r), el cual depende si $v_r \in \mathcal{N}_H$ o $v_r \in \mathcal{N}_I$, donde $\mathcal{N}_H$ es el conjunto de las neuronas hojas del árbol jerárquico y $\mathcal{N}_I$ es el conjunto de las neuronas internas, calculando rqe_r a través de:

$$rqe_r = \begin{cases} \sum_{\underline{x}_i \in \mathcal{C}(\underline{w}_r)} \rho(\underline{x}_i - \underline{w}_r) & \text{Si } \underline{w}_r \in \mathcal{N}_H \\ \frac{1}{|\mathcal{W}_u|} \sum_{\underline{w}_i \in \mathcal{W}_u} rqe_i & \text{Si } \underline{w}_r \in \mathcal{N}_I \end{cases} \tag{24}$$

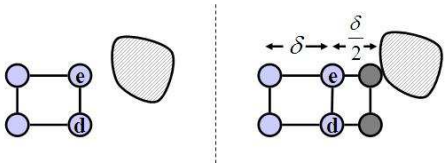


Figura 2: **Crecimiento externo de la malla:** Una columna de neuronas es agregada al costado de las neuronas v_e y v_d.

donde si $v_r \in \mathcal{N}_I$ entonces $\mathcal{W}_u$ es el mapa hijo. Cabe destacar que el error de cuantización robusto de la neurona v_0 es calculado sobre todo el conjunto de datos $\mathcal{I}_T$

La etapa de crecimiento es bastante similar a la explicada en el modelo RGHSOM, pero además existe la posibilidad de agregar neuronas en la parte exterior de la malla. Esta comienza cuantificando la calidad de adaptación de cada neurona $v_r \in \mathcal{W}$ utilizando el error de cuantización robusto (rqe) (ecuación 24), luego se calcula la neurona con mayor error v_e, que corresponde a la neurona con mayor error de cuantización robusto, es decir, $v_e = \arg\max\{rqe_r \in \mathcal{W}\}$.

Para decidir si el mapa necesita crecer en forma interna se calcula el error de cuantización robusto del mapa determinado por:

$$MRQE = \sum_{\underline{x}_i \in \mathcal{C}(\mathcal{W})} \rho\left(\underline{x}_i - \overline{w}\right) \tag{25}$$

donde $\overline{w}$ es el centroide del mapa $\mathcal{W}$ dado por

$$\overline{w} = \frac{1}{|\mathcal{W}|} \sum_{\underline{w}_j \in \mathcal{W}} \underline{w}_j \tag{26}$$

Si el mapa cumple con el criterio $\frac{1}{W}\sum_{\underline{w}\in\mathcal{W}} rqe_r \geq \gamma \cdot MRQE$ *donde* $0 < \gamma << 1$ entonces el mapa no está modelando en forma correcta los datos, por lo tanto, es necesario agregar nuevas neuronas en forma interna al mapa, para lo cual se calcula v_d que corresponde a la neurona vecina más lejana de v_e dado por $v_d = \arg\max\{\|\underline{w}_r - \underline{w}_e\| \underline{w}_r \in \mathcal{N}_e\}$, donde $\mathcal{N}_e$ es el conjunto de neuronas vecinas de v_e. Una vez determinada v_e y v_d se agrega una columna o fila de neuronas entre v_e y v_d donde sus pesos iniciales corresponden al promedio de las neuronas vecinas.

Para decidir si el mapa crecerá en forma externa, es necesario comparar el error de cuantización robusto de la neurona v_e respecto a las otras neuronas del mapa, donde si rqe_e es mayor que el factor θ multiplicado por el máximo error de cuantización del resto de las neuronas, entonces el mapa crecerá en forma externa, es decir,

$$rqe_e \geq \theta \max_{\underline{w}_r \in \mathcal{W}, \underline{w} \neq \underline{w}_e} rqe_r \quad con\ \theta >> 1 \tag{27}$$

Si se cumple la condición 27 se calcula v_d que corresponde a la neurona vecina de v_e cuyo error de cuantización robusto es mayor, y se procede a agregar una columna o fila de neuronas en el costado de las neuronas v_e y v_d cuyo vector de pesos es inicializado como la mitad de la distancia de la columna o fila vecina, pero en dirección opuesta. En la figura 2 se aprecia el crecimiento externo de la malla.

Para decidir si una neurona necesita crecer en forma jerárquica, debido a que necesita una representación más detallada de los datos que modela, se calcula $rqe_r,\ \forall v_r \in \mathcal{N}_H$ las cuales son comparadas con rqe_0. Cada neurona que satisfaga el criterio de crecimiento jerárquico dado por

$$rqe_r \geq \tau \cdot rqe_0 \quad y \quad |\mathcal{C}(\underline{w}_r)| \geq N_{min} \quad 0 < \tau << 1 \tag{28}$$

donde N_{min} es el número mínimo de datos requeridos para crear un mapa, entonces creará un mapa de tamaño inicial 2×2, cuyos pesos iniciales se determinaran en forma aleatoria o determinista alrededor de $\underline{w}_r$.

3.6.2. Aprendizaje del ambiente cambiante

En esta parte del proceso es necesario implementar nuevas capacidades al modelo para que pueda adaptarse a los ambientes cambiantes sin olvidar catastróficamente lo aprendido anteriormente. Estas nuevas capacidades incluyen la detección de datos novedosos, el congelamiento de neuronas y el olvido gradual de los datos aprendidos.

El modelo, al tener la capacidad de aprender de los datos a través del tiempo, tiene la necesidad de congelar las neuronas padres para que solamente aprendan las neuronas hojas, es decir, las neuronas que pertenecen a $\mathcal{N}_I$ del modelo $\mathcal{H}_T$, son congeladas para no verse afectadas durante el proceso de entrenamiento. Finalizado el entrenamiento de las neuronas pertenecientes a $\mathcal{N}_H$ se procede a la modificación de pesos de las neuronas internas, donde el peso de $v_r \in \mathcal{N}_I$ es modificado como el promedio de las neuronas de su mapa hijo utilizando la siguiente ecuación.

$$\underline{w}_r = \overline{w}_r \text{ tal que } \overline{w}_r = \frac{1}{W_r} \sum_{\underline{w}_i \in \mathcal{W}_r} \underline{w}_i, \quad \mathcal{W}_r \text{ mapa hijo de } \underline{w}_r \tag{29}$$

Para obtener una mejor generalización y mantener el modelo estable, es necesario una estrategia de olvido. La estrategia consiste en olvidar gradualmente los datos, contrayendo las neuronas hacia su respectivo centroide, y en caso de que la malla sea pequeña, esta se contraiga eliminando una fila o columna de neuronas.

Una vez calculado el centroide $\overline{w}$ de las neuronas con la ecuación 26, el mapa $\mathcal{W}$ procederá a olvidar los datos aplicando la siguiente regla de olvido:

$$\underline{w}_r(T) = \underline{w}_r(T-1) + \lambda[\underline{w}_r(T-1) - \overline{w}] \quad \underline{w}_r \in \mathcal{W} \tag{30}$$

donde λ es una taza de olvido, $\underline{w}_r(T-1)$ y $\underline{w}_r(T)$ son los pesos de las neuronas al final de la etapa $T-1$ y al comienzo de la etapa T respectivamente.

Posteriormente se realiza el olvido gradual de los datos aprendidos de la misma forma que se realiza el olvido en el modelo FASOM.

3.7. Evaluación de los algoritmos

Para evaluar la calidad de adaptación a los datos de los modelos a utilizar en esta tesis, se implementarán diversas medidas comunes para todos ellos.

La primera medida está basada sobre el promedio de error de cuantización al cuadrado dada por:

$$MSQE = \frac{1}{|\mathcal{I}|} \sum_{\underline{x}_i \in \mathcal{I}} \|\underline{x}_i - \underline{w}_{\eta(\underline{x})}\|^2 \tag{31}$$

donde $|\mathcal{I}|$ es el número total de datos que pertenecen al conjunto de entrada $\mathcal{I}$, y $\underline{w}_{\eta(\underline{x})}$ es la neurona ganadora (BMU) correspondiente al dato $\underline{x}$ del modelo $\mathcal{H}_T$, definido en la ecuación 1. En caso de ser un modelo no jerarquizado se utilizará la neurona más cercana al dato.

Otra medida a utilizar corresponde a la complejidad del modelo, que en este caso va a estar dado por el número total de neuronas hojas (N_h) que utiliza el modelo; es decir, una neurona que sea padre de un mapa en un nivel inferior no se contabiliza, pero si los nodos que son hojas.

4. Estudio Experimental de los Modelos

En esta sección se desarrollarán pruebas con datos sintéticos y reales, para comparar el desempeño, la capacidad de generalización, el ajuste topológico y el olvido gradual/catastrófico de los modelos propuestos en esta Tesis. Previo a cada experimento se realizaron diversas pruebas con distintos parámetros, para determinar la mejor configuración de parámetros posibles para cada modelo bajo estudio, el detalle del valor especificado para los parámetros se especifican en el trabajo [19].

Para los experimentos se evaluaron seis modelos distintos. La red Self Organizing Map (SOM) (ver sección 2.1) y el modelo Growing Hierarchical Self Organizing Map (GHSOM) (ver sección 2.1.1) correspondiente a redes ya existentes y los cuatro modelos propuestos en esta Tesis Robust Self Organizing Map (RSOM) (ver sección 3.3), Robust Growing Hierarchical Self Organizing Map (RGHSOM) (ver sección 3.4) Flexible

Architecture Self Organizing Map (FASOM) (ver sección 3.5) y Robust and Flexible model of Hierarchical Self Organizing Map (RoFlex-HSOM) (ver sección 3.6).

En este trabajo los ambientes dinámicos serán tratados en etapas que consisten en intervalos de tiempo, por ejemplo, una base de datos con registros en los años 1990, 1991 y 1992 tendría 3 etapas.

En los experimentos el espacio de entrada fue separado en conjunto de datos de entrenamiento y conjunto de datos de test para cada etapa. En cada etapa se utilizó los datos de test para obtener el $MSQE$ (ver ecuación 31). En los experimentos se trabaja con ambientes cambiantes por lo cual existen varias etapas, y para evaluar la capacidad de los algoritmos frente al problema de interferencia catastrófica se calcularon las mismas mediciones para una etapa dada T y la etapa anterior a ella $T-1$, es decir, se utilizó el conjunto de datos de test de la época actual y la del conjunto de datos de test de la época anterior.

4.1. Experimento Sintético

En esta sección se diseñará y ejecutará un experimento con un conjunto de datos dinámico y con presencia de outliers. Los resultados mostrados en esta sección corresponden al promedio de 20 pruebas con conjuntos de datos aleatorios generados con las mismas características.

4.2. Ambiente dinámico con datos aberrantes

En este experimento se evalúan los modelos frente a un conjunto de datos dinámico en el tiempo y con presencia de datos aberrantes. El conjunto de datos corresponde a cuatro etapas, compuesta con un total de 6 clusters a los cuales se les agregó $\varpi = 0\,\%,\ 1\,\%,\ 5\,\%,\ y\ 10\,\%$ de datos aberrantes en forma aditiva. Las características de cada cluster se describen a continuación:

- **Cluster 1:** Este cluster está compuesto por tres subcluster gaussianos equidistantes al centroide [0,9; 0,01] y con varianza $0{,}0167^2 \cdot I_2$. Este cluster aparece en la primera etapa con un total de 500 datos entre entrenamiento y test, luego desaparece hasta la cuarta etapa donde vuelve a aparecer con un total de 50 datos.

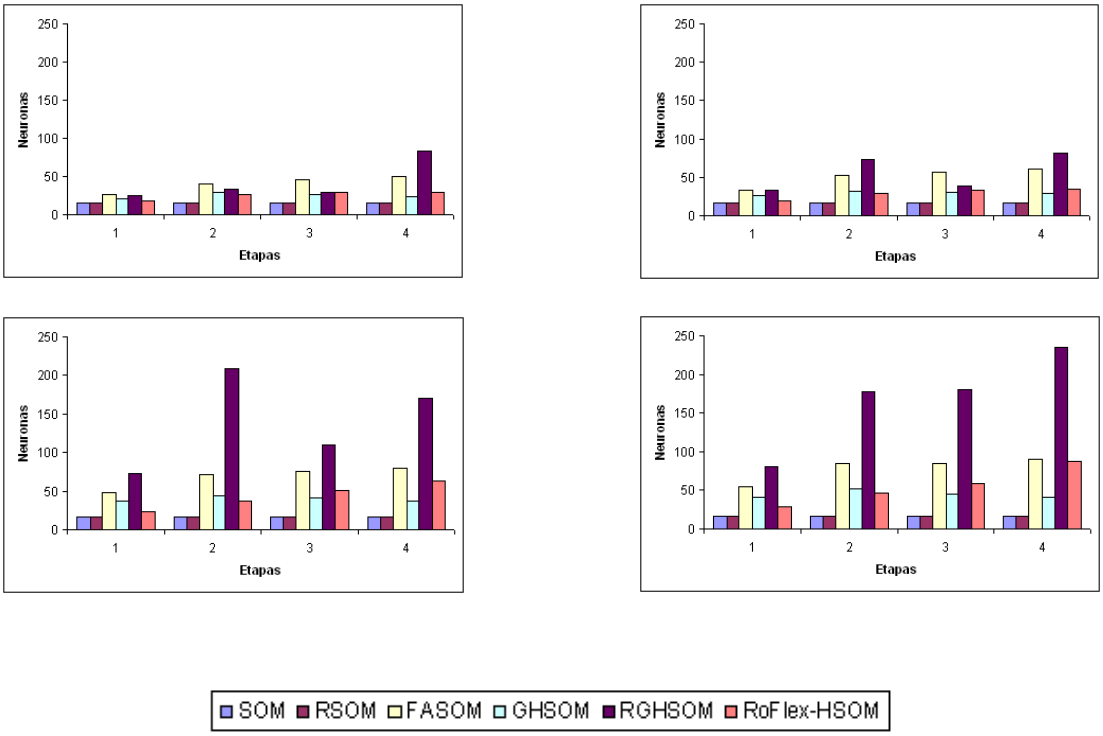
- **Cluster 2:** El cluster dos cuyo centroide es [0,8; 0,5] esta compuesto por tres subcluster gaussianos con varianza $0{,}0167^2 \cdot I_2$. Existen 500 datos entre entrenamiento y test que conforman el cluster en la primera y segunda etapa del experimento, pero, en la tercera etapa este número es reducido a 333 para finalizar con 166 datos en la última etapa.

- **Cluster 3:** A diferencia de los conglomerados anteriores, este cluster, cuyo centroide es [0,6; 0,8], aparece en la segunda etapa del experimento con un total de 500 datos, siendo formado por tres subclusters con varianza $0{,}0167^2 \cdot I_2$, en la tercera etapa el cluster cambia su estructura y lo conforman solamente dos subclusters con varianza $0{,}0250^2 \cdot I_2$ y un total de 500 datos, estas características se mantienen para la última etapa.

- **Cluster 4:** Este conglomerado es el único clusters que se mueve a través de las épocas, siendo en la primera etapa conformado por 500 datos y tres subcluster equidistantes al centroide [0,1; 0,8] y con $\Sigma = 0{,}0167^2 \cdot I_2$, ya en la segunda etapa se mantiene el número de datos, pero la estructura cambia formándose un solo cluster con 500 datos y $\Sigma = 0{,}05^2 \cdot I_2$ con $\mu = [0{,}125;\ 0{,}775]$, luego en la tercera etapa y cuarta etapa se mantiene el número de datos y la varianza constante pero la media es $\mu = [0{,}150;\ 0{,}750]$ y $\mu = [0{,}175;\ 0{,}725]$ respectivamente.

- **Cluster 5:** En la primera etapa este cluster esta conformado por 500 datos correspondiente a una distribución gaussiana con $\mu = [0{,}01;\ 0{,}5]$ y $\Sigma = 0{,}05^2 \cdot I_2$, luego en la segunda etapa, el número de muestras se mantiene pero la varianza aumenta a $\Sigma = 0{,}054^2 \cdot I_2$, en la tercera y cuarta etapa la estructura del cluster cambia y pasa a estar conformado por tres subclusters equidistantes a la media antigua pero con $\Sigma = 0{,}0194^2 \cdot I_2$ y $\Sigma = 0{,}0210^2 \cdot I_2$ respectivamente.

- **Cluster 6:** Este cluster esta conformado por 1500 datos repartidos en tres subclusters equidistantes al centroide [0,5; 0,5] y con $\Sigma = 0{,}0167^2 \cdot I_2$, pero solamente aparece en la tercera y cuarta etapa.

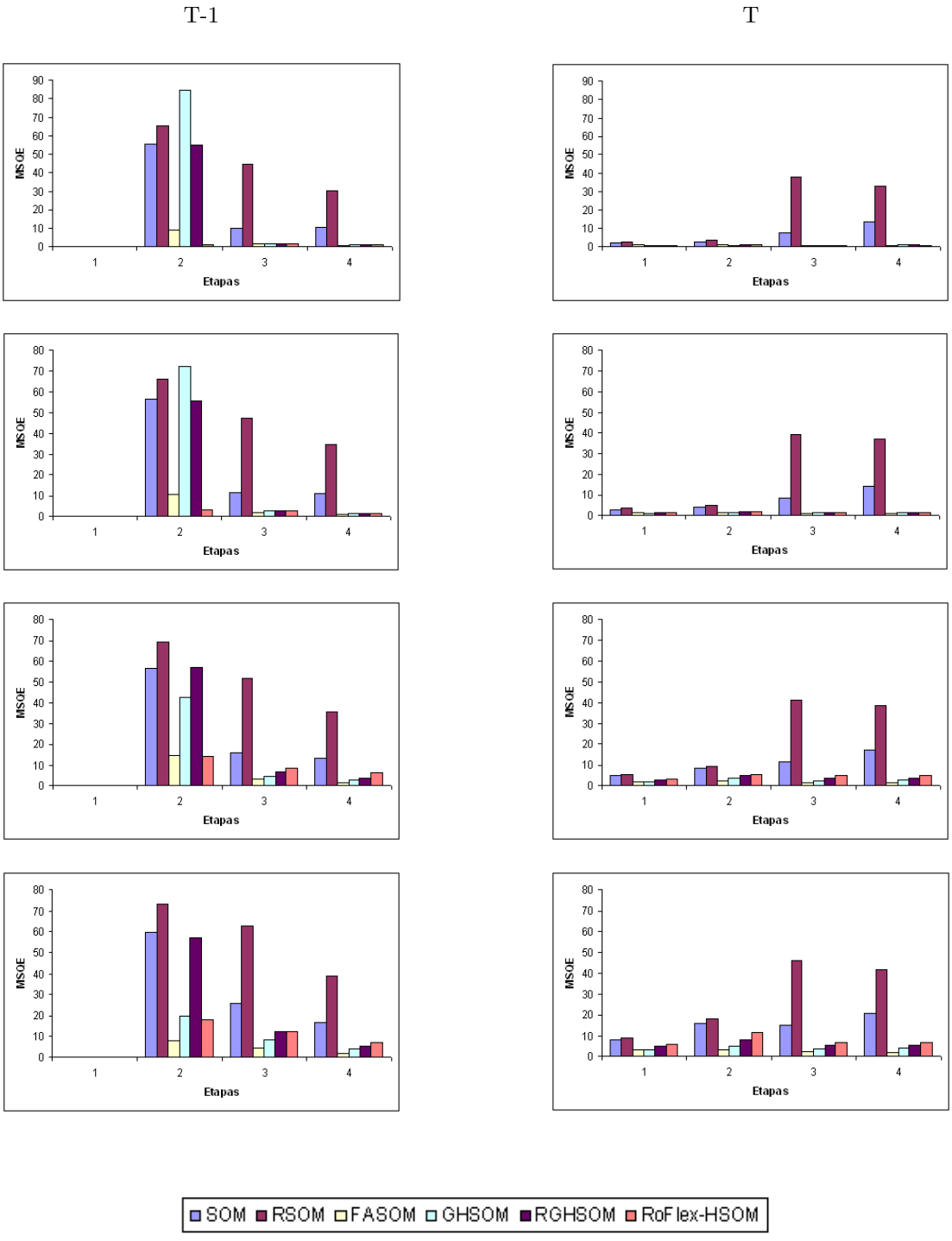


Cuadro 1: **Experimento #3:** Número de neuronas. Los gráficos muestran el número de neuronas promedio que utiliza cada modelo en las cuatro etapas correspondientes al conjunto de dato con $\varpi = 0\,\%$ (superior izquierda), $\varpi = 1\,\%$ (superior derecha), $\varpi = 5\,\%$ (inferior izquierda), $\varpi = 10\,\%$ (inferior derecha).

En el cuadro 1 se muestran el número de neuronas que ocupa cada modelo para los datos con $\varpi = 0\,\%, 1\,\%, 5\,\%$ y $10\,\%$ de datos aberrantes. Es posible observar que el modelo RGHSOM, aumenta considerablemente su tamaño cuando $\varpi = 5\,\%$ y $10\,\%$; una posible explicación sería que podría existir un problema entre el crecimiento de la malla y el aprendizaje robusto. El modelo FASOM y GHSOM muestran un aumento en el número de neuronas proporcional al aumento de porcentaje de datos aberrantes debido al modelamiento de éstos.

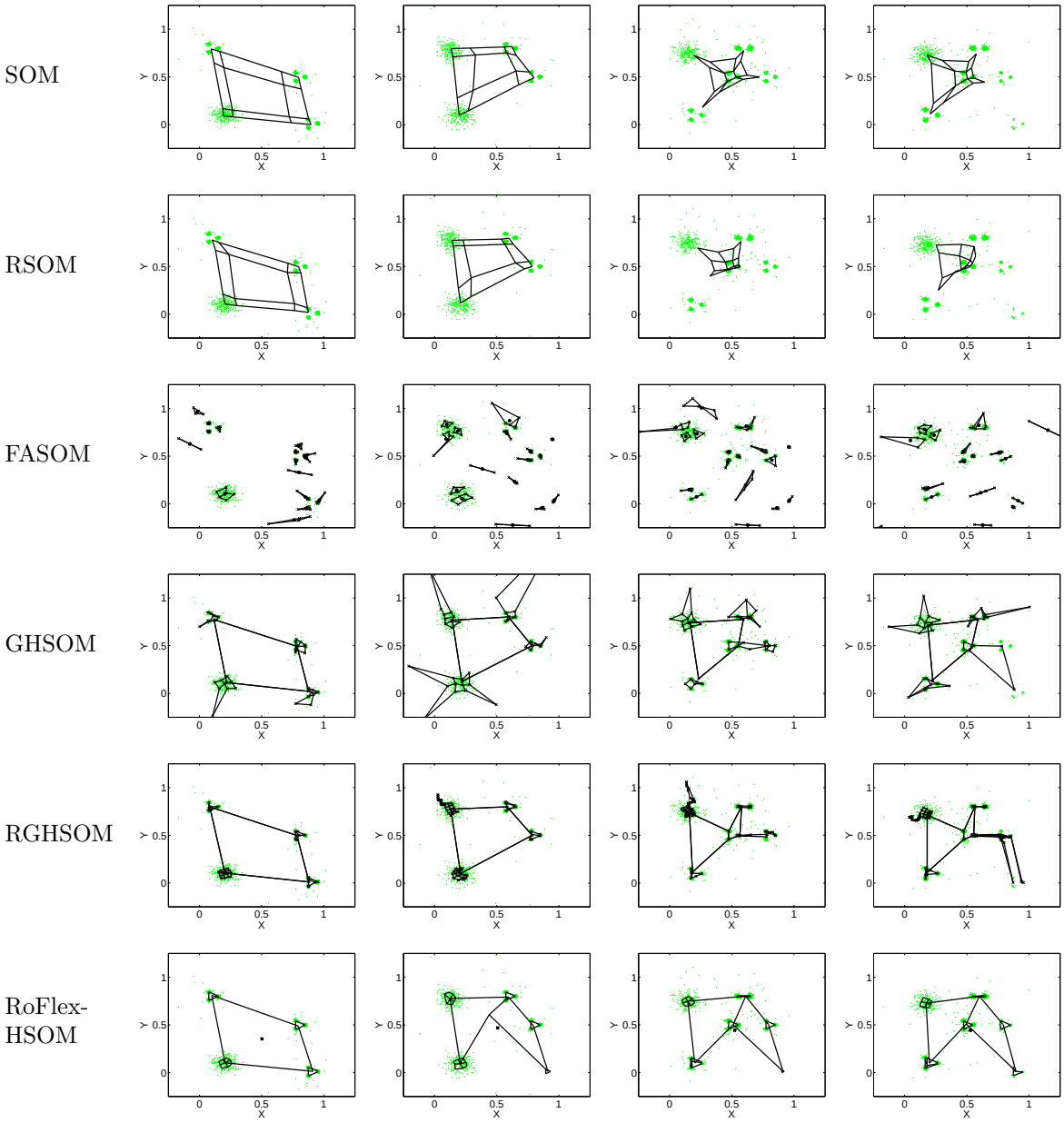
El cuadro 2 muestra los resultados del $MSQE$ para el espacio de entrada con $0\,\%$, (primera fila) $1\,\%$, (segunda fila) $5\,\%$ (tercera fila) y $10\,\%$ (cuarta fila) de outliers. Los gráficos de la izquierda del cuadro 2 representan el $MSQE$ utilizando el conjunto de los datos de test de la etapa anterior, mientras que los gráficos de la derecha muestra el $MSQE$ de los datos de test de la época actual. Como puede ser apreciado en los gráficos de la izquierda, los modelos no flexibles tiene una gran error cuadrático de cuantización en la segunda etapa, lo que implica un olvido catastrófico de los datos aprendidos en la primera etapa, esto no ocurre en las siguientes etapas debido a que no existe un cluster que desaparezca. Como es de esperar el error de los modelos robustos es mayor que los demás debido a que no están modelando los datos con outliers y ello influye en el error cuadrático de cuantización. Esto se aprecia claramente debido a que a medida que aumenta el porcentaje de datos aberrantes incrementa el error de los modelos robustos.

En el cuadro 3 se muestra los resultados gráficos de cada modelo para las distintas etapas con $5\,\%$ de datos aberrantes. En la primera fila se encuentra el modelo SOM, en la segunda fila el modelo RSOM, en la tercera FASOM, cuarta GHSOM, quinta RGHSOM y finalmente RoFlexHSOM. Notar que los modelos SOM, RSOM, GHSOM y RGHSOM sufren de olvido catastrófico entre la segunda y tercera etapa. Al observar la primera columna que corresponde a la primera etapa, se puede observar como los modelos no robustos FASOM y GHSOM se ven afectados por la presencia de datos aberrantes produciendo, en el primer modelo, mallas innecesarias, mientras que en el segundo los mapas se abren para modelar datos aberrantes. En la segunda etapa los modelos mencionados anteriormente siguen siendo afectados por la presencia de outliers, mientras que el modelo RGHSOM crea un nuevo mapa con un gran número de neuronas en forma innecesaria para modelar ciertos datos aberrantes. En la tercera etapa los modelos SOM y RSOM se concentran en la



Cuadro 2: **Experimento #3:** Error cuadrático de cuantización. Los gráficos muestran el $MSQE$ para todos los modelos utilizando un conjunto de datos con 0 % (primera fila), 1 % (segunda fila), 5 % (tercera fila) y 10 %(cuarta fila) de datos aberrantes. A la izquierda se presenta el $MSQE$ para la etapa T-1, mientras que a la derecha los resultados para la etapa T.

gran cantidad de datos que aparecen en esta etapa (1500 datos), lo que explica la forma de los mapas correspondientes. En la cuarta etapa los modelos RSOM y SOM no modelan el pequeño conjunto de datos presentados en la esquina inferior izquierda, mientras que los modelos FASOM, GHSOM y RGHSOM se ven afectados por los datos aberrantes. Es necesario observar el correcto comportamiento del modelo RoFlex-HSOM el cual modela los datos sin verse afectado por los datos outliers y que además presenta un olvido gradual de los datos cuando estos desaparecen, tal como se aprecia en el mapa que modela el cluster 1 entre las etapas 1 y 3. En general, cuando se está en presencia de outliers, el modelo FASOM encontró una cantidad de clusters muy superior a la que corresponde y no logra una buena representación topológica. Por último, los modelos jerárquicos (GHSOM, RGHSOM y RoFlex-HSOM) tienen una mayor estabilidad y por ende se obtiene una mejor representación topológica del espacio de entrada.



Cuadro 3: **Experimento #3:** Resultados gráficos. Cada columna, de izquierda a derecha, corresponde a la primera, segunda, tercera y cuarta etapa del conjunto de datos con 5 % de datos aberrantes.

4.3. Experimento con conjunto de datos reales

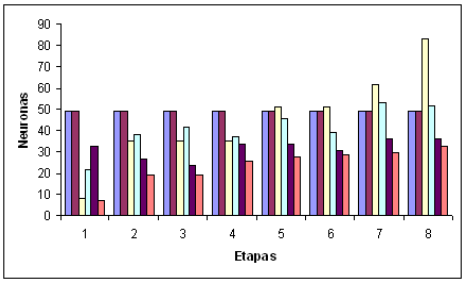
Se realizará un experimento utilizando la base de datos reales **El Niño**, los resultados mostrados corresponden a un promedio de 20 pruebas utilizando los mismos datos y sin modificar los parámetros de cada modelo.

4.3.1. Experimento: Corriente del Niño

El banco de datos **El niño** se encuentra en el sitio *http://kdd.ics.uci.edu*. Esta base de datos fue obtenida por el Pacific Marine Environment Laboratory National Oceanic and Atmospheric Administration, la cual contiene mediciones oceánicas y climatológicas de la superficie obtenidas a través de boyas posicionadas en el océano Pacífico, con el fin de entender y predecir los ciclos de El Niño Southern Oscillation.

El conjunto de datos está compuesto por las variables: Fecha, latitud, longitud, vientos zonales (Oeste<0, Este>0), vientos meridionales (Sur<0, Norte>0), humedad relativa, temperatura del aire, temperatura de la superficie del mar y temperatura del agua a 500 metros de profundidad. Los registros más antiguos son del año 1980.

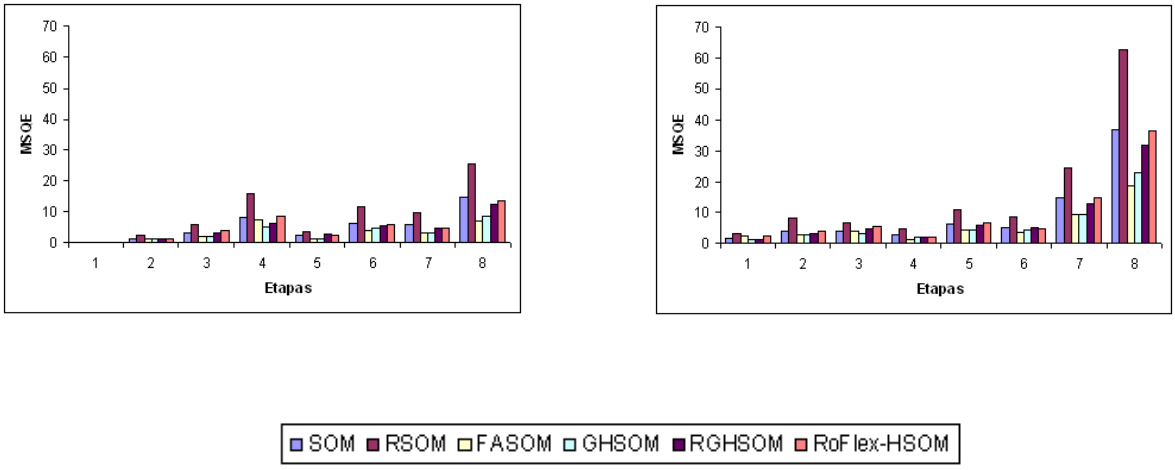
El conjunto de datos fue modificado borrando los datos que contuvieran valores perdidos. Para el experimento se utilizó los datos correspondientes a los años 1980 hasta el 1987, contabilizando un total de 6311 datos con 4 dimensiones (vientos meridionales, humedad relativa, temperatura de la superficie del agua y temperatura del agua a 500 metros de profundidad). De este total de muestras, 4415 se utilizan para entrenamiento y 1896 datos para test. El conjunto de datos fue divido en 8 etapas, una para cada año, compuesta de datos de aprendizaje y de test. La cantidad de datos de entrenamiento utilizados para cada etapa son: 110, 226, 293, 160, 411, 337 837 y 2041, mientras que el número de datos de test son: 48, 97, 126, 69, 177, 145, 359 y 875. El incremento en el número de datos en los últimos dos años puede deberse a un aumento en el número de boyas que adquieren los datos. Para poder ejecutar las simulaciones se estandarizó los datos del conjunto de entrenamiento de la primera etapa entre 0 y 1, las siguientes etapas se estandarizaron utilizando la misma escala de la primera etapa. La estandarización de los datos se realiza para que todas las dimensiones tengan la misma importancia en los modelos.



Cuadro 4: **Experimento corriente del Niño:** Número de neuronas. El gráfico muestra el número de neuronas promedio que utiliza cada modelo en las ocho etapas correspondientes al conjunto de datos.

Un comportamiento general de los modelos es aumentar el número de neuronas para modelar el espacio de entrada a medida que avanzan las épocas, tal como se muestra en el cuadro 4, este aumento de neuronas se puede dar debido al incremento de la cantidad de datos en cada etapa o en el caso de los modelos FASOM y ROFLEX-HSOM, porque necesitan nuevas neuronas para modelar nuevos conjuntos de datos. Cabe destacar que el modelo RoFlex-HSOM es uno de los modelos menos complejos en relación a la cantidad de neuronas que utiliza.

En el cuadro 5 se muestra los resultados obtenidos para el error cuadrático de cuantización para cada modelo en cada una de las etapas. En la imagen izquierda se muestran los resultados utilizando el conjunto de datos correspondiente a la etapa T-1, mientras que en la imagen derecha se utilizó el conjunto de datos



Cuadro 5: **Experimento corriente del Niño:** Error cuadrático de cuantización. Los gráficos muestran el $MSQE$ para todos los modelos. En la parte izquierda se presenta el $MSQE$ para la etapa T-1, mientras que en la parte derecha los resultados para la etapa T.

correspondiente a la etapa T. Al analizar los resultados obtenidos en la imagen superior se aprecia que en la cuarta etapa, el segundo conjunto de entrenamiento más pequeño, los modelos tienden a olvidar lo aprendido en la tercera etapa, por lo cual debe existir un comportamiento extraño de los datos en la cuarta etapa. A pesar de este comportamiento extraño que se vuelve a repetir en la octava etapa, la cual se puede deber al aumento en el número de datos, no existe modelo que olvide en forma catastrófica los datos, por lo cual es posible pensar que no existe un cambio abrupto en el tiempo, es decir, no debe existir un patrón que desaparezca en alguna época.

Al analizar los valores del error cuadrático de cuantización el modelo FASOM, obtiene el menor MSQE en la mayoría de las épocas lo que puede deberse a la gran cantidad de neuronas que utiliza a medida que avanzan las etapas al modelar posibles datos aberrantes. Es posible apreciar que los modelos que tienen una estructura más compleja (RoFlex-HSOM, GHSOM y RGHSOM) logran modelar de mejor manera el espacio de entrada al obtener un MSQE más bajo que los modelos SOM y RSOM. Finalmente se puede observar que a pesar de la simplicidad en relación al número de neuronas del modelo RoFlex-HSOM, éste obtiene uno de los valores más bajos de $MSQE$ en la mayoría de las etapas.

5. Conclusiones y Trabajos Futuros

5.1. Conclusiones

En esta Tesis se modificó el modelo SOM para entregar a los nuevos modelos (RGHSOM, RoFlex-HSOM) la capacidades de aprender de jerarquías existentes en los datos. Además se proponen modelos robustos frente a datos aberrantes (RSOM, RGHSOM, RoFlex-HSOM) y flexibles para adaptarse a cambios en el ambiente (FASOM y RoFlex-HSOM). Los modelos RSOM y RGHSOM son modelos robustos que no se ven mayormente afectado por la presencia de datos aberrantes, mientras que el modelo FASOM no olvida en forma catastrófica los datos aprendidos en épocas anteriores, finalmente el modelo RoFlexHSOM es un modelo robusto y flexible con la capacidad de aprender la topología de los datos a través del tiempo sin olvidar en forma catastrófica los datos aprendidos anteriormente.

Mediante el estudio de la función de influencia se muestra que el aprendizaje de la red SOM no es robusto en el sentido de Huber, lo cual fue solucionado con el modelo RSOM, además se robusteció el modelo GHSOM, el cual, al utilizar la misma regla de aprendizaje que el modelo SOM tampoco es robusto en el sentido distribucional.

Se demostró empíricamente que los modelos robustos tienen mejor desempeño en MSQE y además son más simples al tener una menor cantidad de neuronas. Además los modelos robustos logran una mejor

distribución de los datos en las neuronas y un mejor factor de uso de las neuronas.

Se estudió y presentó el dilema de plasticidad y flexibilidad, y además, el problema de Interferencia Catastrófica, y se comprobó que los mapas autoorganizativos presentan esta dificultad. Además se demostró empíricamente que los modelos Flexibles no sufren de Interferencia catastrófica, sino que de un olvido gradual cuando el ambiente es cambiante.

Al analizar los resultados del experimento sintético, queda demostrado que los modelos SOM, GHSOM y FASOM son afectados por la presencia de datos aberrantes. Los modelos RSOM, RGHSOM y RoFlex-HSOM no son afectados mayormente por la presencia de outliers, entre estos modelos, el RSOM no modela en forma correcta la topología de los datos al tener una estructura rígida, los modelos RGHSOM y RoFlex-HSOM son los modelos mejor evaluados frente a un espacio de entrada con datos aberrantes.

Se comprobó, a través del experimento sintético, que los modelos SOM, RSOM, GHSOM y RGHSOM olvidan en forma catastrófica la topología aprendida en épocas anteriores. Los modelos FASOM y RoFlex-HSOM tienen la capacidad de olvidar en forma gradual, y no catastrófica, los patrones aprendidos en etapas anteriores.

En los datos reales, a pesar de que el modelo RoFlex-HSOM obtuvo un mayor MSQE que los modelos FASOM, GHSOM, RGHSOM, podemos pensar que, debido a los resultados obtenidos en los datos sintéticos, obtuvo una mejor representación topológica.

Al observar el MSQE y la cantidad de neuronas en el experimento real, es posible que todos los modelos excepto el RoFlex-HSOM, hayan modelado los outliers disminuyendo así el MSQE, pero aumentado la cantidad de neuronas.

Se demostró que el modelo RoFlex-HSOM, utilizando una menor cantidad de neuronas, logró una mejor representación topológica que los demás modelos cuando el ambiente es no-estacionario y con presencia de datos aberrantes.

Lamentablemente el MSQE es un indicador global y no es una buena herramienta para evaluar la calidad de los modelos en ambientes dinámicos con presencia de datos aberrantes.

En esta Tesis se logró crear el modelo RoFlex-HSOM, que es un algoritmo flexible y robusto, que obtiene una buena representación topológica de los datos obtenidos de ambientes no estacionarios y con datos aberrante, y olvida gradualmente patrones aprendidos con anterioridad.

5.2. Trabajos Futuros

Las ventajas de los modelos presentados en esta Tesis son muchas, puesto que permiten modelar ambientes dinámicos y con datos aberrantes, pero existen varios puntos en los que queda bastante trabajo por hacer, como se explica a continuación.

Al robustecer la función de aprendizaje de la red SOM, se utilizó la función de Huber, se propone realizar un estudio teórico y empírico que compare el uso de otros estimadores robustos en la función de aprendizaje de los modelos para seleccionar el de mejor comportamiento.

El principal modelo propuesto en esta Tesis Roflex-HSOM es un modelo robusto que puede modelar ambientes cambiantes en el tiempo, por esta razón será un modelo bastante útil para el modelamiento de problemas reales, por lo que se propone utilizarlo en bases de datos reales como contaminación del aire, calentamiento global, etc.

Otro problema a tratar es incorporar robustez y flexibilidad a otros modelos para analizar su comportamiento y tratar de modelar datos en ambientes dinámicos con presencia de datos aberrantes.

Referencias

[1] *Episodic memory on connectionist networks* (1990), Proceedings of the Twelth Conference of the Cognitive Science Society.

[2] Allende, H., Moreno, S., Rogel, C., and Salas, R. Robust self organizing maps. *LNCS 3287* (Nov 2004), 179–186.

[3] Barnes, J., and Underwood, B. "fate.ºf first-learned associations in transfer theory. *Journal of experimental Psychology 58* (1959), 97–105.

[4] Bauer, H.-U., and Villmann, T. Growing a hypercubical output space in a self-organizing feature map. *IEEE Transactions on Neural Networks 8*, 2 (March 1997), 218–226.

[5] Berkhin, P. Survey of clustering data mining techniques. Tech. rep., Accrue Software, San Jose, CA, 2002.

[6] Carpenter, G. A. A distributed outstar network for spatial pattern learning. *Neural Netw. 7*, 1 (1994), 159–168.

[7] Dittenbach, M., Merkl, D., and Rauber, A. The Growing Hierarchical Self-Organizing Map. In *Proc of the International Joint Conference on Neural Networks (IJCNN 2000)* (Como, Italy, July 24. – 27. 2000), S. Amari, C. L. Giles, M. Gori, and V. Puri, Eds., vol. VI, IEEE Computer Society, pp. 15 – 19.

[8] Erwin, E., Obermayer, K., and Shulten, K. Self-organizing maps: ordering, convergence properties and energy functions. *Biol. Cyb. 67* (1992), 47–55.

[9] Frean, M., and Robins, A. Catastrophic forgetting in simple networks: an analysis of the pseudorehearsal solution. *Computation in Neural Systems 10* (1999), 227–236.

[10] French, R. M. Catastrophic interference in connectionist networks: Can it be predicted, can it be prevented? In *Advances in Neural Information Processing Systems* (1994), J. D. Cowan, G. Tesauro, and J. Alspector, Eds., vol. 6, Morgan Kaufmann Publishers, Inc., pp. 1176–1177.

[11] Grossberg, S. Processing of expected and unexpected events during conditioning and attention: A psychophysiological theory. *Psychological Review 89* (1982), 529–572.

[12] Grossberg, S. Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science 11* (1987), 23–63.

[13] Hampel, F. *Contributions to the theory of robust estimation.* PhD thesis, University of California, Berkeley, 1968.

[14] Hampel, F., Ronchetti, E., Rousseeuw, P., and Stahel, W. *Robust Statistics.* Wiley Series in Probability and Mathematical Statistics, 1986.

[15] Hartigan, J. A., and Wong, M. A. A K-means clustering algorithm. *Applied Statistics 28* (1979), 100–108.

[16] Huber, P. J. Robust estimation of a location parameter. *Ann. Math. Statist.*, 35 (1964), 73–1001.

[17] Kohonen, T. The self-organizing map. In *Proceedings of the IEEE* (1990), vol. 78, pp. 1464–1480.

[18] Metcalfe, J. A composite holographic associative recall model. *Psychological Review 89* (1982), 627–661.

[19] Moreno, S. Robustez y flexibilidad en los mapas autoorganizativos para ambientes no estacionarios. Master's thesis, Universidad Técnica Federico Santa María, 2007.

[20] Moreno, S., Allende, H., Rogel, C., and Salas, R. Robust growing hierarchical self organizing maps. *LNCS LNCS:3512* (Jun 2005), 341–348.

[21] Ratcliff, R. Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions. *Psychological Review 97* (1990), 285–308.

[22] Salas, R., Allende, H., Moreno, S., and Saavedra, C. Flexible architecture of self organizing maps in changing environments. *LNCS 3773* (Nov 2005), 642–653.

[23] Salas, R., Moreno, S., Allende, H., and Moraga, C. A robust and flexible model of hierarchical self organizing maps for nonstationary environments. *To appear Neurocomputing* (2007).

[24] Tukey, J. A survey of sampling from contaminated distributions. *In Contributions to Probability and Statistics* (1960), 448–485.