



Certificación ISO 9001:2008 ‡

DESARROLLO DE UNA RED NEURONAL ARTIFICIAL PARA ANALIZAR LA RESPUESTA DEL PUERTO DE MANZANILLO BAJO EL EFECTO DE TSUNAMIS HIPOTÉTICOS QUE SE PUDIERAN OCASIONAR EN LA BRECHA SÍSMICA DE GUERRERO

José Miguel Montoya Rodríguez
Juan Pedro Vásquez López

**Publicación Técnica No. 445
Sanfandila, Qro. 2015**

SECRETARÍA DE COMUNICACIONES Y TRANSPORTES
INSTITUTO MEXICANO DEL TRANSPORTE

**Desarrollo de una red neuronal artificial para
analizar la respuesta del Puerto de Manzanillo bajo
el efecto de tsunamis hipotéticos que se pudieran
ocasionar en la brecha sísmica de Guerrero**

Publicación Técnica No.445
Sanfandila, Qro. 2015

Esta investigación fue realizada en la Coordinación de Ingeniería Portuaria Y Sistemas Geoespaciales del Instituto Mexicano del Transporte, por el Jefe de la División de Ingeniería de Puertos y Costas M. en C. José Miguel Montoya Rodríguez y el investigador titular M. en C. Juan Pedro Vásquez López.

Contenido

| | | |
|-------------------|-------------------|-----|
| Resumen | | iii |
| Abstract | | iv |
| Resumen ejecutivo | | vi |
| Capítulo 1. | Introducción | 1 |
| Capítulo 2. | Metodología | 5 |
| Capítulo 3 | Resultados | 9 |
| Capítulo 3. | Conclusiones | 13 |
| Bibliografía | | 15 |
| Anexo | Rutinas de código | 17 |

Resumen

El presente estudio tiene como objetivo diseñar, codificar y optimizar dos redes neuronales artificiales tipo feedforward, como estimadores de patrones de rapidez de corrientes en el interior del Puerto de Manzanillo en Colima, sujeto a la acción de tsunamis hipotéticos, generados en la Brecha Sísmica de Guerrero; lográndose el aprendizaje efectivo a partir de tsunamis sintéticos locales, el patrón de rapidez máxima y el patrón de rapidez media, con un error porcentual absoluto medio de 20% en ambos casos.

Abstract

The objective of this study is to design, code and optimize two feedforward-type artificial neural networks, as estimators of current speed patterns inside the Port of Manzanillo in Colima, with action of hypothetical tsunamis, generated in the Guerrero Seismic Gap; achieving effective learning from local synthetic tsunamis, the maximum speed pattern and the average speed pattern, with a mean absolute percentage error of 20% in both cases.

Resumen ejecutivo

Existe un segmento de la Costa Grande al noroeste de Acapulco, conocido como “*la Brecha de Guerrero*”, que tiene el mayor déficit de liberación de energía sísmica de la zona. El terremoto más reciente data de 1911 y, al día de hoy, más de 5m de convergencia tectónica, ha generado sólo unos cuantos eventos de magnitud $M_w=6$ en las orillas del segmento. La estimación del potencial sismogénico asume que el área de trabazón por fricción puede abruptamente liberarse en un solo evento, lo que podría resultar en un sismo de la misma magnitud que el evento de 1985 ($M_w=8.1$) y si la ruptura se propagara al sur de Acapulco, la magnitud podría ser hasta de $M_w=8.4$.

Un terremoto de tal envergadura puede también generar un tsunami. Ya que los tsunamis se propagan sin mucha pérdida energética en grandes distancias, su afectación es casi global.

Teniendo como objetivo de aprendizaje tres tsunamis hipotéticos generados en la brecha sísmica de Guerrero que arriban al Puerto de Manzanillo en Colima, el presente trabajo logró el diseño y programación de un par de redes neuronales artificiales tipo *feedforward*, para los patrones de rapidez máxima y media, a partir de seis tsunamis sintéticos locales.

La mayor parte del código está desarrollado en la plataforma estadística R, y haciendo uso de técnicas de cómputo paralelo para acortar el tiempo de cálculo de las miles de corridas involucradas y almacenando todos los datos en una base de datos relacional.

Se obtuvieron así, un par de redes neuronales artificiales que aprendieron efectivamente los patrones antes mencionados con una precisión del orden del 20% de error porcentual absoluto medio.

1 Introducción

Los terremotos generados en las zonas de ruptura en la interfaz entre placas tectónicas en subducción, apoyan la hipótesis geofísica de que la energía de tensión sísmica se acumula en partes de la interfaz trabadas por fricción y ésta se libera mediante sismos.

Se tiene que un área de la costa del Pacífico, sobre la Placa Americana Media, en el estado mexicano de Guerrero, es uno de los sitios de mayor peligro sísmico, justo donde se da la subducción de la Placa de Cocos bajo el sur de México. Allí se han originado más de veinte sismos de magnitud mayor a $M_w=7$, sólo en el siglo pasado. Las convergencias rápidas de 5 a 7 cm por año producen grandes terremotos en intervalos de 30 a 100 años [1].

Actualmente, existe un segmento de dicha área (Costa Grande al noroeste de Acapulco) conocido como “*la brecha de Guerrero*”, que aparentemente tiene un potencial sísmico mayor, es decir, que tiene el mayor déficit de liberación de energía sísmica, pues el terremoto más reciente data de 1911 y, al día de hoy, más de 5m de convergencia tectónica ha generado sólo unos cuantos eventos de magnitud $M_w=6$ en las orillas del segmento.

Además, la estimación del potencial sismogénico simplifica el comportamiento de la falla, asumiendo que el área de trabazón por fricción puede abruptamente liberarse en un solo evento, lo que resultaría excepcionalmente peligroso, pues esta brecha sísmica es también la más cercana a la Ciudad de México (con una población de casi 20 millones).

Estudios han establecido que esta particular zona generadora de sismos en Guerrero es de longitud similar a la ruptura de 1985 en Michoacán. En consecuencia, la ruptura de la brecha noroeste de Guerrero podría resultar en un sismo de la misma magnitud que el evento de 1985 ($M_w=8.1$) y si la ruptura se propaga al sur de Acapulco, la magnitud podría ser hasta de 8.4 [2,3].

Por otro lado, un terremoto de tal envergadura puede también generar un tsunami. Los tsunamis son series de ondas oceánicas largas, que se caracterizan por tener una longitud de onda mucho mayor que su altura, y que se forman al ocurrir alguna alteración, como la ruptura de la brecha lo haría, de corta duración pero de gran extensión en su superficie libre. Ya que los tsunamis se propagan sin mucha pérdida energética en grandes distancias, su afectación es casi global.

Se reconoce que, si bien existe una amplia literatura que abarca los temas tanto de la brecha sísmica de Guerrero, como de la dinámica de los tsunamis, no son muchas las fuentes que aborden el caso de los posibles tsunamis originados por dicha zona sísmogénica y sus efectos en el puerto mexicano de Manzanillo, Colima, y menos, que empleen para ello novedosas técnicas de modelación numérica basadas en inteligencia artificial, como es el caso de las redes neuronales tipo *feedforward*.

Particularmente, hay el antecedente de una investigación referente al tema, y que servirá de base fundamental al presente trabajo. Llevada a cabo por personal de la División de Ingeniería de Puertos y Costas del IMT, detalla el diseño, programación y optimización de un par de redes neuronales artificiales tipo *feedforward* (RNAf) que reproducen los patrones de rapidez máxima y media de las corrientes respectivamente, que se dan en el interior del Puerto de Manzanillo bajo la acción de tsunamis (Proyecto VI-17/13) [4].

En dicho trabajo se consideraron 50 puntos geográficos de modelación sobre la entrada del Puerto de Manzanillo, Figura 1.1, además de 21 casos de elevación inicial del tsunami sobre la línea batimétrica 34 metros del puerto, como se muestra en la Tabla 1.1.

Es importante mencionar que los datos de velocidades de corrientes, proceden a su vez de un estudio anterior (Proyecto VE-12/13) [5] donde se empleó un modelo hidráulico a escala 1:150, mostrado en la Figura 1.2.

Tabla 1.1 Casos de elevación inicial en la línea batimétrica 34m del Puerto de Manzanillo, Col. (Proyecto de investigación VI-17/13)

| Caso Núm. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|--------------|------|------|------|------|------|------|------|------|------|
| η_0 (m) | 1.07 | 1.08 | 1.24 | 1.42 | 1.46 | 1.60 | 1.78 | 2.49 | 2.58 |

| Caso Núm. | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|--------------|------|------|------|------|------|------|------|------|------|
| η_0 (m) | 2.66 | 2.84 | 3.02 | 3.20 | 3.37 | 3.44 | 3.55 | 4.30 | 6.02 |

| Caso Núm. | 19 | 20 | 21 |
|--------------|------|-------|-------|
| η_0 (m) | 8.60 | 12.05 | 15.50 |



Figura 1.1 Puntos de modelación del Puerto de Manzanillo para rapidez máxima y media de corrientes en el *Proyecto de investigación VI-17/13*



Figura 1.2 Modelo hidráulico a escala del Puerto de Manzanillo, utilizado en el *Proyecto de investigación VE-12/13* que aportó datos para el *Proyecto de investigación VI-17/13*

Las RNAf optimizadas se entrenaron con los primeros 20 casos de elevación inicial. Teniendo como variables de entrada:

- el número del punto de modelación
- la elevación inicial
- la latitud, la longitud y la profundidad del punto de modelación

Con una sola variable de salida: la rapidez máxima o media, según el caso.

Mostraron un aprendizaje efectivo del patrón de corrientes, obteniendo menos de 20% de error al contrastar las predicciones hechas con la RNAf contra el último caso de elevación inicial. Resultados satisfactorios considerando la baja cantidad total de casos de entrenamiento disponibles.

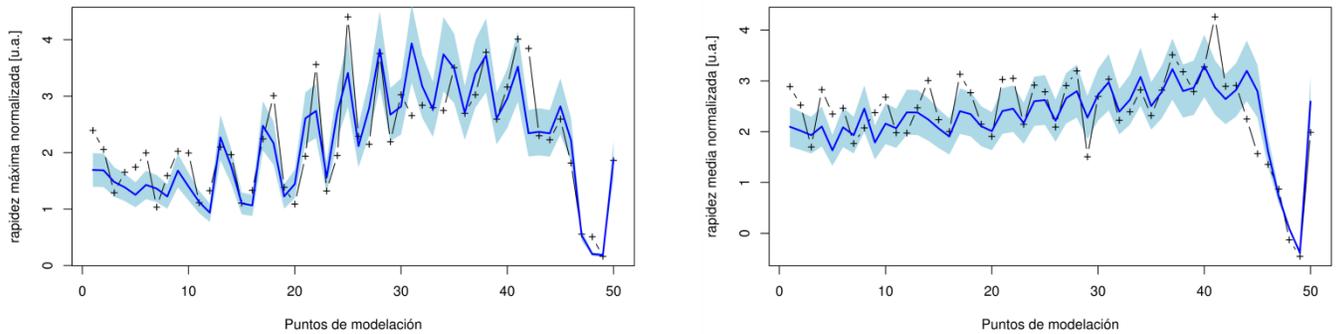


Figura 1.3 Comparación de patrones de rapidez máxima (izquierda) y rapidez media (derecha) de corrientes en el interior del Puerto de Manzanillo del *Proyecto de investigación VI-17/13*. Los puntos señalados con ‘+’ corresponden a los datos del modelo físico, la línea azul continua es la predicción de la RNAf y la sombra azul claro es el error porcentual absoluto medio entre ambos.

Ante tal panorama, como una necesaria continuación de los trabajos en este tema, el presente estudio tiene por objetivo:

Explorar la posibilidad de hacer un análisis de la respuesta del Puerto de Manzanillo, Colima, bajo el efecto de tsunamis hipotéticos que se pudieran ocasionar en la brecha sísmica de Guerrero, mediante una red neuronal artificial feedforward.

2 Metodología

Se utilizaron los datos sísmicos y de velocidad de corrientes aportados por el Proyecto de Investigación VE-12/13 [7], correspondientes a tres tsunamis hipotéticos que arriban al Puerto de Manzanillo, Colima, como se listan en la Tabla 2.1.

Tabla 2.1 Tsunamis históricos y la elevación inicial en la línea batimétrica 34m del Puerto de Manzanillo, Col. (*Proyecto de investigación VE-12/13*)

| Caso Núm. | 1 | 2 | 3 |
|----------------------|------|------|------|
| η_0 (m) | 1.24 | 1.78 | 3.37 |
| Momento sísmico (Mw) | 8.5 | 8.6 | 8.8 |

Con el fin de buscar un par de RNAf que aprendan efectivamente los patrones de rapidez máxima y media de corrientes generadas en el interior del puerto durante dichos tsunamis históricos, se consideraron como casos de entrenamiento para las redes, seis tsunamis sintéticos locales aportados por el Proyecto de Investigación VI-17/13 [6], listados en la Tabla 2.2.

Tabla 2.2 Tsunamis sintéticos locales y la elevación inicial en la línea batimétrica 34m del Puerto de Manzanillo, Col. (*Proyecto de investigación VI-17/13*)

| Caso Núm. | 1 | 2 | 3 | 4 | 5 | 6 |
|--------------|------|------|------|------|-------|-------|
| η_0 (m) | 2.58 | 4.30 | 6.02 | 8.60 | 12.05 | 15.50 |

Para hallar las RNAf óptimas, se definieron los siguientes conjuntos dominio, resumidos en la Tabla 2.3, que incluyen las posibles variantes de los parámetros de las redes, siendo la mejor combinación la que resulte en un menor error porcentual absoluto medio (MAPE) entre el patrón de rapidez del tsunami histórico y la respuesta de la red.

Tabla 2.3 Conjuntos dominio de parámetros de las RNAf

| Variables de entrada | Variables de salida | Tasa de aprendizaje | Momentum global | Ciclos de entrenamiento | Factor para el número de neuronas en capa oculta |
|---|--|---------------------|--------------------|-------------------------|--|
| | 0. Rapidez | 0.02 | 0.05 | 1×10^3 | 2 |
| | 0. Rapidez 1. Número de punto | 0.002 | 0.005 | 9×10^3 | 2.25 |
| 1. Número de punto 2. Elevación inicial 3. Longitud 4. Latitud 5. Profundidad | 0. Rapidez 1. Número de punto 2. Elevación inicial | 2×10^{-4} | 5×10^{-4} | 9×10^4 | 2.5 |
| | 0. Rapidez 1. Número de punto 5. Profundidad | 2×10^{-5} | 5×10^{-5} | | 2.75 |
| | 0. Rapidez 2. Elevación inicial 5. Profundidad | | | | 3 |
| 1. Número de punto 2. Elevación inicial | 0. Rapidez 1. Número de punto 2. Elevación inicial 5. Profundidad | | | | 3.25 |
| 1. Número de punto 2. Elevación inicial 5. Profundidad | | | | | |

De donde claramente se reconocen del orden de 5×10^3 corridas para cada uno de los tipos de rapidez de corriente. Por lo que, para acortar los tiempos de cálculo, es necesario implementar estrategias de cómputo paralelo, y así aprovechar toda la capacidad de procesamiento de los varios núcleos con los que cuentan los CPUs actuales, como con el que cuenta la computadora donde se llevaron a cabo las corridas. El código está desarrollado principalmente en 3 rutinas R: la primera almacena todas las combinaciones de los conjuntos dominio de parámetros de las RNAf en una base de datos relacional construida en MySQL®, la segunda hace propiamente el entrenamiento y simulación de la RNAf, almacenando sus resultados en la misma base de datos y, una última, grafica los resultados, como se describe en la Figura 2.1.

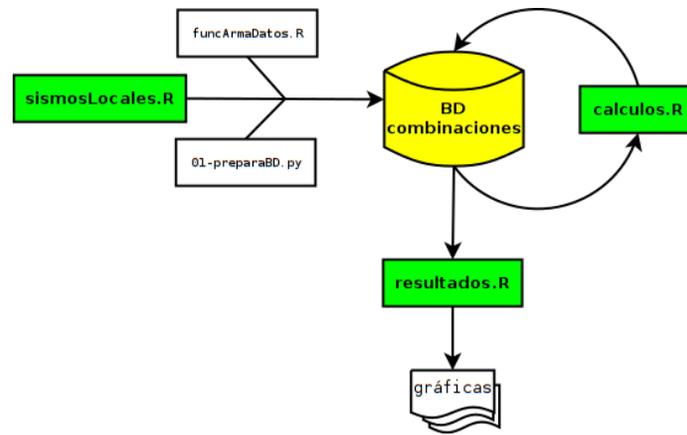


Figura 2.1 Flujo de trabajo de las rutinas que buscan las RNAf óptimas para los patrones de rapidez máxima y media de corrientes

3 Resultados

Las RNAf óptimas que aprendieron los patrones de rapidez máxima y media de corrientes al interior del Puerto de Manzanillo, sobre los 50 puntos de modelación mostrados en la Figura 1.1, bajo la acción de los tsunamis hipotéticos de la Tabla 2.1, tienen las siguientes características.

RNAf estimador del patrón de rapidez máxima

Para el patrón de rapidez máxima, el aprendizaje fue efectivo, con un error MAPE del 22.79%, como se muestra en las gráficas de linealidad y normalidad de la Figura 3.1.

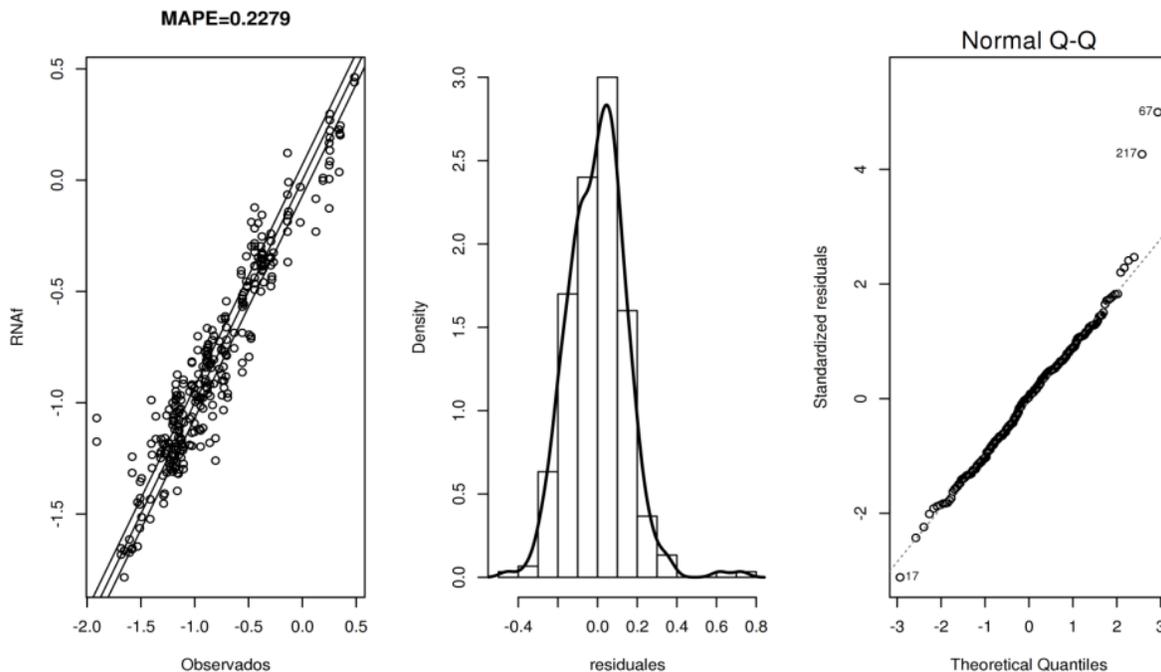


Figura 3.1 Gráficas de linealidad y normalidad de la respuesta de la RNAf del patrón de rapidez máxima

Su topología es de 5 variables de entrada: 1) número de punto de modelación, 2) elevación inicial (η_0), 3) latitud, 4) longitud y 5) profundidad z del punto de

modelación; y dos variables de salida: 1) la rapidez máxima y 2) el número de punto de modelación, con 14 neuronas en la capa intermedia y 9×10^4 ciclos de entrenamiento, como muestra el diagrama de la Figura 3.2.

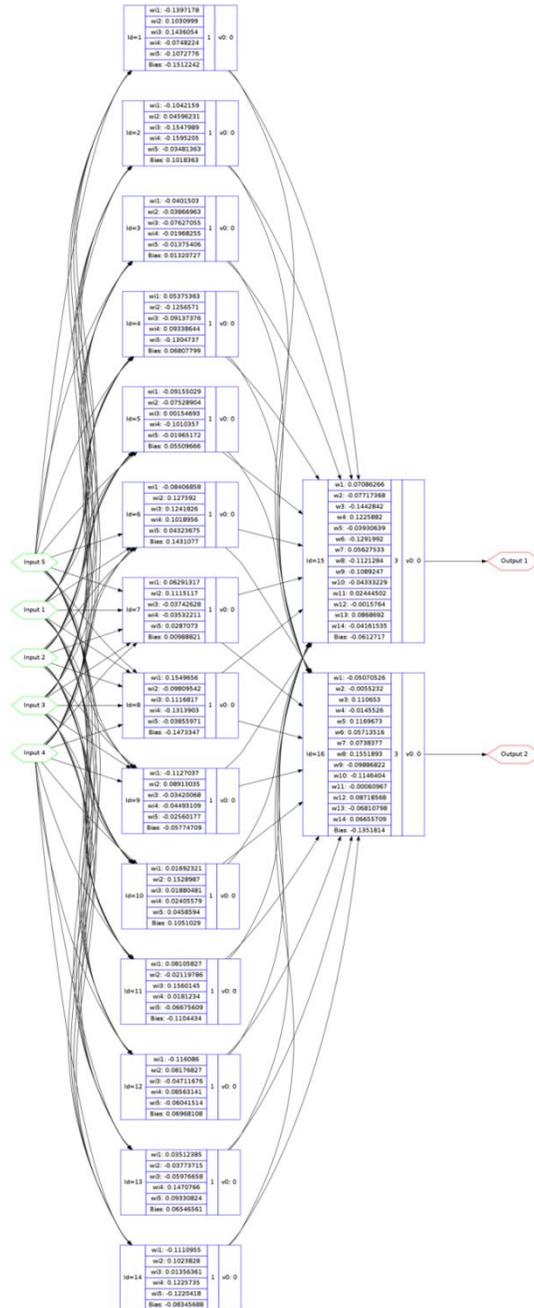


Figura 3.2 Topología de la RNAf del patrón de rapidez máxima

RNAf estimador del patrón de rapidez media

En el caso de rapidez media, el menor MAPE alcanzado fue de 21.10% como se muestra en la Figura 3.3.

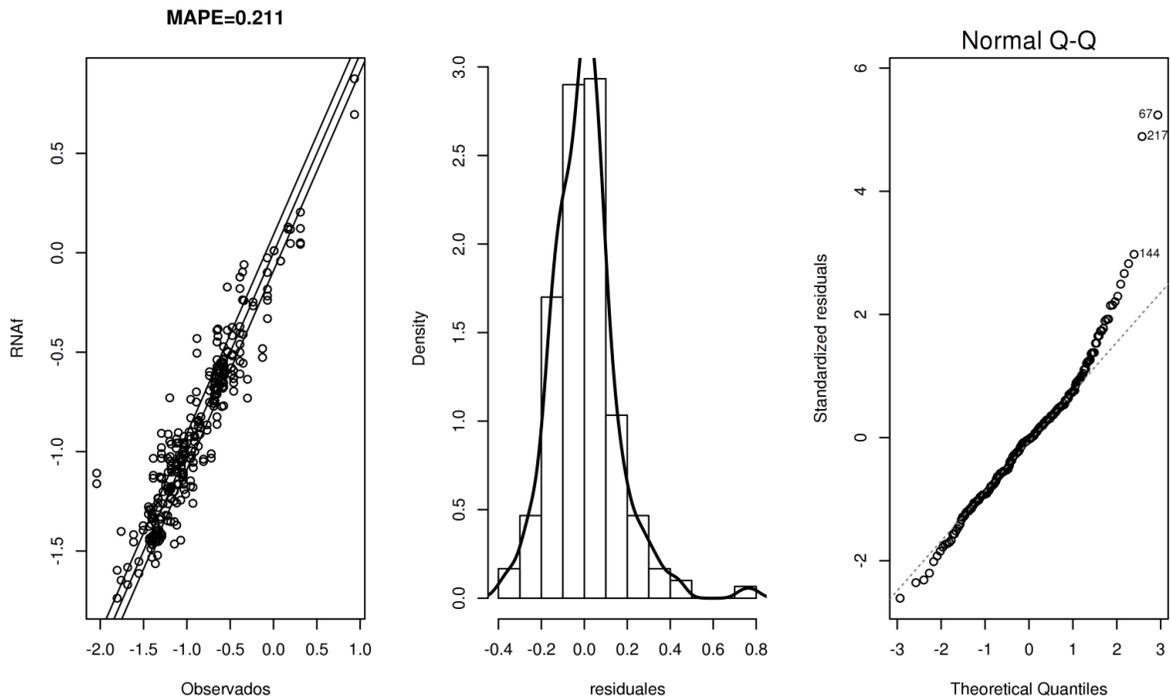


Figura 3.3 Gráficas de linealidad y normalidad de la respuesta de la RNAf del patrón de rapidez media

En este caso, la topología es de 5 variables de entrada: 1) número de punto de modelación, 2) elevación inicial (η_0), 3) latitud, 4) longitud y 5) profundidad z del punto de modelación; y una variables de salida: 1) la rapidez media, con 17 neuronas en la capa intermedia y 9×10^4 ciclos de entrenamiento, como muestra el diagrama de la Figura 3.4.

Desarrollo de una red neuronal artificial para analizar la respuesta del Puerto de Manzanillo bajo el efecto de tsunamis hipotéticos que se pudieran ocasionar en la brecha sísmica de Guerrero

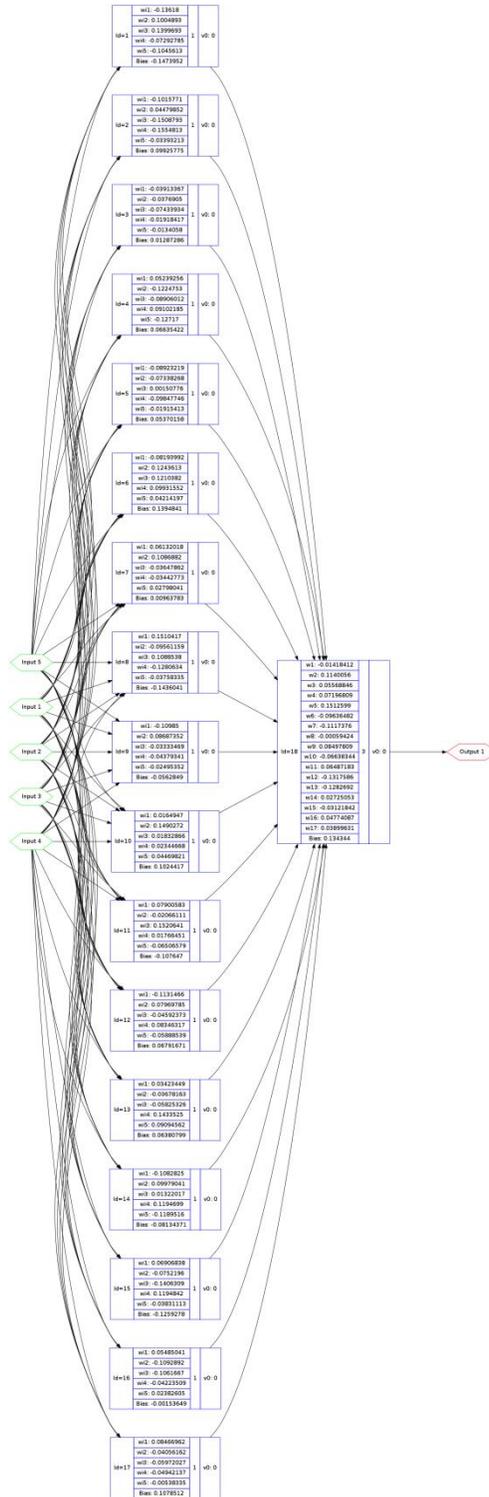


Figura 3.4 Topología de la RNAf del patrón de rapidez media

4 Conclusiones

Se logró entonces construir un par de RNAf que aprendieron los patrones de rapidez máxima y media de corrientes en el interior del Puerto de Manzanillo, Colima, bajo la acción de tres tsunamis hipotéticos, a partir de tsunamis sintéticos locales, con una precisión del orden del 20% de error porcentual absoluto medio, como evidencia, se muestran los resultados en la Figura 4.1.

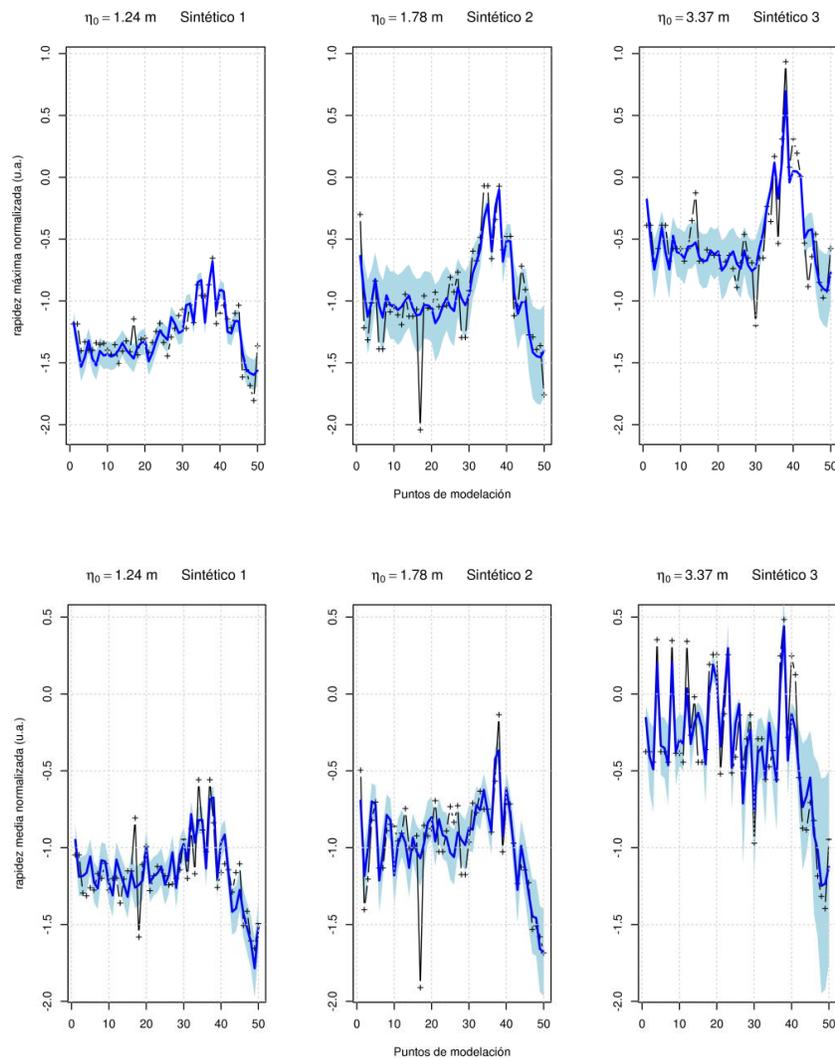


Figura 4.1 Patrones de rapidez máxima (superior) y media (inferior). En símbolos '+' negros el tsunami local sintético y la línea azul continua es la respuesta de la RNAf. El intervalo de confianza en sombra azul claro

5 Bibliografía

- [1] LOWRY, Anthony. R., LARSON, Kristine M., KOSTOGLODOV, Vladimir, et. al. Transient fault slip in Guerrero, southern Mexico. *Geophysical Research Letters*. Oct 2001. Vol 28, núm. 19 pp. 3753-3756.
- [2] KOSTOGLODOV Vladimir, et. al. Gravity and seismicity over the Guerrero seismic gap, Mexico. *Geophysical Research Letters*. Nov. 1996. Vol 23, núm. 23, pp. 3385-3388.
- [3] SUÁREZ, G., MONFRET, T., WITTLINGER, G., y DAVID, C. Geometry of subduction and depth of the seismogenic zone in the Guerrero gap, Mexico. *Nature*. Mayo 1990. Vol. 345, pp. 336-338.
- [4] VÁSQUEZ LÓPEZ Juan P. Red neuronal feedforward como estimador de patrones de corrientes en el interior del Puerto de Manzanillo sujeto a la acción de tsunamis. Proyecto de investigación de iniciativa interna IV-17/13. Publicación técnica 406. Instituto Mexicano del Transporte. 2014.
- [5] MONTOYA, JOSÉ MIGUEL, FLORES ÁLVAREZ, JUAN ESTÉBAN, CASAS VALENCIA, CINDY, et al. Estudio en modelo hidráulico para determinar el patrón de velocidades originado por tsunamis locales en el Puerto de Manzanillo, Col. Reporte Técnico VE-12/13. Instituto Mexicano del Transporte. 2013.

6 Anexo. Rutinas de código

sismosLocales.R

```
# LIBRERIAS Y VALORES INICIALES -----
rm(list = ls())

library(RMySQL) #para conectarse con la BD
library(doMC) #para cómputo paralelo
library(AMORE) #para RNAF
subdirTrabajo <- "~/Dropbox/2014/2014-07-21-VI-28-14/rutinasRv01/"
nCP <- 6 #número de casos de entrenamiento
colobj <- 7 #6 rapMax o 7 para rapMedia
registerDoMC(cores = 4) #4 bueno para Core-i5
system("python 01-preparaBD.py root imt123")

setwd(subdirTrabajo)
source("funcArmaDatos.R")
locales <- armaDatos("datos/locales/")
locales.sinteticos <- armaDatos("datos/sinteticosLocales/")
todo <- rbind(locales, locales.sinteticos)
todo.normalizado <- sapply(todo, function(original) normalizado <- (original
-
  mean(original))/sd(original))
locales.normalizado <- head(todo.normalizado, dim(locales)[1])
locales.sinteticos.normalizado <- tail(todo.normalizado,
dim(locales.sinteticos)[1])
rm(todo, todo.normalizado)

# CONJUNTOS DOMINIO -----
colsp <- rev(list(c(1, 2, 3, 4, 5), c(1, 2), c(1, 2, 5))) #columnas de
entrada
colsq <- rev(list(c(colobj), c(colobj, 1), c(colobj, 1, 2), c(colobj,
1, 5), c(colobj, 2, 5), c(colobj, 1, 2, 5))) #columnas de salida
LR <- c(0.02, 0.002, 2e-04, 2e-05) #tasas de aprendizaje
MG <- c(0.05, 0.005, 5e-04, 5e-05) #momenta
SS <- c(1000, 9000, 90000) #pasos de entrenamiento
FF <- c(2, 2.25, 2.5, 2.75, 3, 3.25) #factores p/núm. neuronas capa oculta
picked <- 0 #bandera: combinación en proceso
done <- 0 #bandera: combinación ya procesada

# PREPARACIÓN DE BASE DE DATOS -----
base <- "corridas"
usuario <- "userRNAF"
passw <- "imt123"
bd <- dbConnect(MySQL(), user = usuario, password = passw, dbname = base,
```

```
host = "localhost")
dbSendQuery(bd, "TRUNCATE combinaciones") #borra registros pasados
dbDisconnect(bd)

# CICLOS PARALELIZADOS PARA LLENAR COMBINACIONES EN BD -----
for (cP in colsP) {
  for (cQ in colsQ) {
    for (factor in FF) {
      foreach(k = 1:length(LR)) %dopar% {
        for (j in 1:length(MG)) {
          foreach(i = 1:length(SS)) %dopar% {
            bd <- dbConnect(MySQL(), user = usuario,
                             password = passw, dbname = base, host =
                               "localhost")

            lr = LR[k]
            mg = MG[j]
            ss = SS[i]
            orden <- paste("INSERT INTO combinaciones
                            (nCP,colsP,colsQ,hidden,lr,mg,ss,picked,done)
                            VALUES nCP, ',' , paste(as.character(cP),
                            collapse = '-'),
                            ',' , ',' , paste(as.character(cQ), collapse =
                            '-'),
                            ',' , ',' , (floor(length(cP) * factor + 1)),
                            ',' , lr, ',' , mg, ',' , ss, ',' , picked,
                            ',' , done, ')", sep = ",")
            dbSendQuery(bd, orden)
            dbDisconnect(bd) #desconecta la base de datos
          }
        }
      }
    }
  }
}
```

funcArmaDatos.R

```
armaDatos <- function(ruta, fraccPtosModelo = 1:50) {
  datos <- data.frame()
  x <- list.files(ruta, full.names = TRUE)
  sapply(x, function(y) {
    puntos.numPuntos <- fraccPtosModelo
    y.eta0 <- strsplit(y, "/")[1][2]
    y.eta0 <- as.numeric(substr(y.eta0, start = 6, stop = 9))/100
    puntos.eta0 <- rep(y.eta0, length(fraccPtosModelo))
    puntos.lonLatZ <- read.table("datos/TranspuestasXYZ.dat")
    puntos.lonLatZ <- puntos.lonLatZ[fraccPtosModelo, ]
    d <- read.table(y)
```

```

d <- d[, fraccPtosModelo]
vx <- d[1:59, ]
vy <- d[60:118, ]
rapidez <- sqrt(vx * vx + vy * vy)
rapidez.Maxima <- sapply(rapidez, max)
rapidez.Media <- sapply(rapidez, mean)
datos.parciales <- cbind(puntos.numPuntos, puntos.eta0,
puntos.lonLatZ, rapidez.Maxima, rapidez.Media)
names(datos.parciales) <- c("numPunto", "eta0", "lon",
"lat", "z", "rapidez.Maxima", "rapidez.Media")
datos <-<- rbind(datos, datos.parciales)
})
datos
}

```

01-preparaBD.py

```

import MySQLdb as db
import sys
con = db.connect('localhost',sys.argv[1], sys.argv[2])
cur = con.cursor()
cur.execute('SHOW DATABASES;')
todas = cur.fetchall()
bandera=0

for base in todas:
    if base[0] == 'corridas':
        bandera=1

if bandera == 1:
    cur.execute('DROP DATABASE corridas;')
    cur.execute("DROP USER 'userRNAf'@'localhost';")

cur.execute('CREATE DATABASE corridas;')
cur.execute('USE corridas;')
cur.execute('CREATE TABLE combinaciones (nCP INT, colSP CHAR(30), colSQ
CHAR(30), hidden INT, lr DOUBLE, mg DOUBLE, ss DOUBLE, picked INT, done INT,
mape DOUBLE, sigma DOUBLE, lmIs DOUBLE, tf DOUBLE, tff DOUBLE, colObj
INT);')
cur.execute("CREATE USER 'userRNAf'@'localhost' IDENTIFIED BY 'imt123';")
cur.execute("GRANT ALL ON corridas.* TO 'userRNAf'@'localhost';")

if con:
    con.close()

```

cálculos.R

```
# LIBRERIAS Y VALORES INICIALES -----
library(RMySQL) #para conectarse con la BD
library(doMC) #para cómputo paralelo
library(AMORE) #para RNAf
rm(list = ls())
subdirTrabajo <- "~/Dropbox/2014/2014-07-21-VI-28-14/rutinasRv01/"
colObj <- 7 #6 rapMax o 7 para rapMedia
registerDoMC(cores = 4) #4 bueno para Core-i5
set.seed(12)
setwd(subdirTrabajo)
source("funcArmaDatos.R")
locales <- armaDatos("datos/locales/")
locales.sinteticos <- armaDatos("datos/sinteticosLocales/")
todo <- rbind(locales, locales.sinteticos)
todo.normalizado <- sapply(todo, function(original) normalizado <- (original
-
  mean(original))/sd(original))
locales.normalizado <- head(todo.normalizado, dim(locales)[1])
locales.sinteticos.normalizado <- tail(todo.normalizado,
dim(locales.sinteticos)[1])
rm(todo, todo.normalizado)

# PREPARACIÓN DE BASE DE DATOS -----
base <- "corridas"
usuario <- "userRNAf"
passw <- "imt123"

# CICLO PARALELIZADO QUE EJECUTA TODAS LAS COMBINACIONES
# REGISTRAD ----- ARREGLA Y CUENTA CICLOS RESTANTES
t0 <- proc.time() #inicia cronómetro
bd <- dbConnect(MySQL(), user = usuario, password = passw, dbname = base,
  host = "localhost")
Q0 <- "UPDATE combinaciones SET picked=0 WHERE picked=1 AND done=0"
dbGetQuery(bd, Q0) #repara interrupción
Q1 <- "SELECT * FROM combinaciones WHERE picked=0 AND done=0"
combinacs <- dbGetQuery(bd, Q1) #faltan por procesar
dbDisconnect(bd)

# EJECUTA LOS RESTANTES CICLOS
foreach(k = 1:dim(combinacs)[1], .inorder = FALSE, .combine = rbind) %dopar%
{
  set.seed(12) #homologa semilla aleatoria
  caso <- combinacs[k, ] #cada una de las combinaciones
  bd <- dbConnect(MySQL(), user = usuario, password = passw,
    dbname = base, host = "localhost")
  marcaPicked <- paste("UPDATE combinaciones SET picked = 1 WHERE nCP=",
    caso$nCP, " AND colsP='", caso$colsP, "' AND colsQ='",
    caso$colsQ, "' AND hidden=", caso$hidden, " AND lr=",
```

```

    caso$lr, " AND mg=", caso$mg, " AND ss=", caso$ss,
    sep = "")
dbGetQuery(bd, marcaPicked) #asegura para procesar
t1 <- proc.time() #inicia cronómetro local del caso

# LEE LAS COLUMNAS DE ENTRADA
cP <- eval(parse(text = as.character(paste("c(",
    paste(strsplit(as.character(caso$colsP),
    "-")[[1]], collapse = ","), ")"))))

# LEE LAS COLUMNAS DE SALIDA
cQ <- eval(parse(text = as.character(paste("c(",
    paste(strsplit(as.character(caso$colsQ),
    "-")[[1]], collapse = ","), ")"))))
P <- as.matrix(locales.sinteticos.normalizado[, cP]) #matriz de entrada a
la RNAf
Q <- as.matrix(rbind(as.matrix(locales.normalizado[,
    cQ]), as.matrix(locales.normalizado[, cQ]))) #patrón de
entrenamiento

# ARMADO DE LA RNAf
set.seed(12)
net.start <- newff(n.neurons = c(dim(P)[2], caso$hidden,
    length(cQ)), learning.rate.global = caso$lr, momentum.global =
caso$mg,
    error.criterium = "LMLS", stao = NA, hidden.layer = "tansig",
    output.layer = "purelin", method = "ADAPTgdwm")

# ENTRENAMIENTO
set.seed(12)
result <- train(net.start, P, Q, error.criterium = "LMLS",
    report = TRUE, show.step = caso$ss, n.shows = 3)

# SIMULACIÓN
sim <- sim.MLPnet(result$net, P)
# valores observados (resto de datos)
obs <- Q
# modelo lineal simulados vs observados
fit <- lm(sim[, 1] ~ obs[, 1])
# cálculo del MAPE
mape <- mean(abs((sim[, 1] - obs[, 1])/obs[, 1]))
tf <- round((proc.time() - t1)[[3]], 4) #tiempo de cómputo RNAf
tff <- round((proc.time() - t0)[[3]], 4) #tiempo total del programa

# SALIDA DE RESULTADO A BD
marcaDone = paste("UPDATE combinaciones SET done = 1, mape=",
    mape, ", sigma=", round(summary(fit)$sigma, 4), ", lmls=",
    round(result$merror[3], 4), ", tf=", tf, ", tff=",
    tff, ", colobj=", colobj, " WHERE nCP=", caso$nCP,
    " AND colsP='", caso$colsP, "' AND colsQ='", caso$colsQ,
    "' AND hidden=", caso$hidden, " AND lr=", caso$lr,

```

```
" AND mg=", caso$mg, " AND ss=", caso$ss, " AND picked=",
1, sep = "")
dbGetQuery(bd, marcaDone) #marca la combinación de 'procesada'

# DESCONECTA DE LA BD
dbDisconnect(bd)
}
```

resultados.R

```
rm(list = ls())
library(RMySQL) #para conectarse con la BD
library(AMORE) #para RNAf
subdirTrabajo <- "~/Dropbox/2014/2014-07-21-VI-28-14/rutinasRv01/"
colobj <- 7 #6 rapMax o 7 para rapMedia
nombreFig1 <- "RNAf-rapMedia.dot"
nombreFig2 <- "lmRapMedia.png"
nombreFig3 <- "RapMedia.png"
yLabelFig3 <- "rapidez media normalizada (u.a.)"
source("funcArmaDatos.R")
locales <- armaDatos("datos/sinteticosBrecha/")
locales.sinteticos <- armaDatos("datos/sinteticosLocales/")
todo <- rbind(locales, locales.sinteticos)
todo.normalizado <- sapply(todo, function(original) normalizado <- (original
-
mean(original))/sd(original))
locales.normalizado <- head(todo.normalizado, dim(locales)[1])
locales.sinteticos.normalizado <- tail(todo.normalizado,
dim(locales.sinteticos)[1])
rm(todo, todo.normalizado)

# PREPARACIÓN DE BASE DE DATOS -----
base <- "corridas"
usuario <- "userRNAf"
passw <- "imt123"

# RECUPERA LA RNAf OPTIMA -----
bd <- dbConnect(MySQL(), user = usuario, password = passw, dbname = base, host
= "localhost")
q0 <- "SELECT * FROM combinaciones WHERE mape IN (SELECT MIN(mape) FROM
combinaciones)"
pmin <- dbGetQuery(bd, q0) #recupera parámetros de la RNAf optima
dbDisconnect(bd)
cP <- eval(parse(text = as.character(paste("c(",
paste(strsplit(as.character(pmin$colSP),
"-")[1]), collapse = ","), ")"))))
cQ <- eval(parse(text = as.character(paste("c(",
paste(strsplit(as.character(pmin$colSQ),
"-")[1]), collapse = ","), ")"))))
P <- as.matrix(locales.sinteticos.normalizado[, cP]) #matriz de entrada a la
RNAf
```

```

Q <- as.matrix(rbind(as.matrix(locales.normalizado[, cQ]),
as.matrix(locales.normalizado[,
  cQ]))) #patrón de entrenamiento

# armado de la RNAf
net.start <- newff(n.neurons = c(dim(P)[2], pmin$hidden, length(cQ)),
learning.rate.global = pmin$lr, momentum.global = pmin$mg,
error.criterion = "LMLS", Stao = NA, hidden.layer = "tansig",
output.layer = "purelin", method = "ADAPTgdwm")

# entrenamiento
set.seed(12)
result <- train(net.start, P, Q, error.criterion = "LMLS", report = TRUE,
  show.step = pmin$ss, n.shows = 3)

# simulación
set.seed(12)
sim <- sim.MLPnet(result$net, P)
obs <- Q
fit <- lm(sim[, 1] ~ obs[, 1]) # modelo lineal simulados vs observados
mape <- mean(abs((sim[, 1] - obs[, 1])/obs[, 1])) #MAPE
rmse <- round(sqrt(mean(resid(fit)^2)), 4) #RMSE

# GRAFICAS DE LA RNAf OPTIMA Y CALIDAD DE PREDICC. -----
graphviz.MLPnet(net = net.start, filename = nombreFig1)

png(filename = nombreFig2, width = 8, height = 4.944, units = "in",res = 600)
  layout(matrix(c(1:3), 1, 3, byrow = TRUE))
  plot(obs[, 1], sim[, 1], main = paste("MAPE=", round(mape, 4),
    sep = ""), ylab = "RNAf", xlab = "Observados")
  deltaOrdenada <- summary(fit)$coefficients[1] +
summary(fit)$coefficients[3]
  abline(deltaOrdenada, 1)
  abline(-deltaOrdenada, 1)
  abline(0, 1)
  hist(fit$residuals, probability = T, main = "", xlab = "residuales")
  lines(density(fit$residuals), lwd = 2)
  plot(fit, which = 2)
dev.off()

png(filename = nombreFig3, width = 8, height = 4.944, units = "in",res = 600)
  layout(matrix(c(1:3), 1, 3, byrow = TRUE))
  graficaParcial <- function(tramo, leyendaY, leyendaX, etaVal,
  leyendaMain) {
    mape <- mean(abs((sim[tramo, 1] - obs[tramo, 1])/obs[tramo,1]))
#MAPE
    low <- sim[tramo, 1] * (1 - mape)
    up <- sim[tramo, 1] * (1 + mape)
    plot(obs[tramo, 1], type = "n", ylab = leyendaY, xlab = leyendaX,
      ylim = c(min(min(obs[, 1], min(sim[, 1]))), max(max(obs[, 1],
        max(sim[, 1])))), main = bquote(eta[0] == .(paste(etaVal,
        "m", "\t", leyendaMain)))
    polygon(c((1:50), rev(1:50)), c(low, rev(up)), col = "lightblue",

```

```
border = FALSE)
  points(obs[tramo, 1], lwd = 1, col = "black", type = "b", pch = "+")
  lines(sim[tramo, 1], lwd = 2, col = "blue")
  grid()
}
graficaParcial(1:50, yLabelFig3, "", unique(locales$eta0)[1], "Sintético
1")
graficaParcial(51:100, "", "Puntos de modelación",
  unique(locales$eta0)[2], "Sintético 2")
graficaParcial(101:150, "", "", unique(locales$eta0)[3], "Sintético 3")
dev.off()
rm(list = ls())
```



Carretera Querétaro-Galindo km 12+000
CP 76700, Sanfandila
Pedro Escobedo, Querétaro, México
Tel +52 (442) 216 9777 ext. 2610
Fax +52 (442) 216 9671

publicaciones@imt.mx

<http://www.imt.mx/>