



Aprendizaje computacional

Examen:

Estimación de la radiación solar en Sonora usando redes
neuronales

Leonardo Coronado Arvayo

2 de diciembre de 2020

Índice

1. Introducción	2
2. Datos	2
3. Metodología	3
3.1. ANN para predecir la radiación solar	3
3.2. Métricas de medición	4
4. Resultados	4
4.1. Métricas de medición de las diferentes ANNs	6
5. Conclusiones y discusión	6

Resumen

En el presente trabajo se usan Redes Neuronales Artificiales (ANN por sus siglas en inglés) para estimar la radiación solar geográfica en Sonora, usando datos de 63 estaciones meteorológicas. Los tipos de ANNs utilizados son el algoritmo de memoria Limitada Broyden–Fletcher–Goldfarb–Shanno (L-BFGS), Estimación de momentos adaptativa acelerada de Nesterov (Nadam, por sus siglas en inglés), Descenso de Gradiente (DG), así como el modelo de Levenberg–Marquardt (LM). En general, el optimizador LM tuvo los mejores resultados y los más consistente, seguido de cerca por L-BFGS.

1. Introducción

La radiación o irradiación solar en la superficie (RSS) afecta significativamente los procesos de la superficie, característicamente tiene una longitud de onda de 0.3 a 0.4 μm y sus unidades son W/m^2 [6]. Un ejemplo de la importancia de la RSS, es para la generación de energía solar en vista de que a mayor energía solar recibida, mayor es la electricidad generada [5].

Los tipos de métodos de aprendizaje computacional que más se usan para la estimación de la RSS, con sus diferentes variantes, son el de red neuronal artificial, máquinas de soporte vectorial o de soporte, redes Bayesianas, aprendizaje basado en árboles de decisión, entre otros [6, 7, 8, 9, 10].

En el presente trabajo se usan los algoritmos de optimización cuasi newtonianos L-BFGS, Nadam y Levenberg–Marquardt (de las librerías [11], [12] y [4], respectivamente). Este último algoritmo se agrega debido a que [3] encuentra que mejores resultados con este método.

El trabajo se divide primero en una sección para describir brevemente los datos, específicamente las estaciones meteorológicas y su ubicación geográfica; seguido se revisa brevemente la metodología de funcionamiento de las ANNs, así como las ANNs comúnmente usadas para predecir la radiación solar y la métrica de optimización usada para elegir la ANN que se adapta mejor

a los datos; luego se describen los resultados en términos de estas métricas de optimización y se eligen los mejores modelos; por último, se presentan las conclusiones.

2. Datos

Los datos de se obtienen gracias a las estaciones de las Redes de Estaciones Meteorológicas Automáticas de Sonora (REMAS) que son parte del CESAVE – SIAFESON [1].

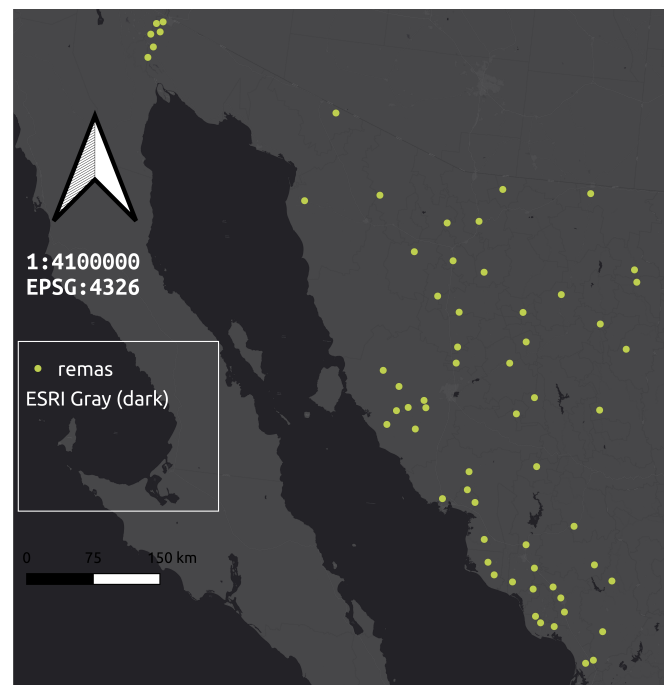


Figura 1: Mapa de las estaciones meteorológicas de [1]

En este trabajo se usaron 63 estaciones repartidas en diferentes partes de Sonora, que se pueden ver en la siguiente figura, aunque REMAS cuenta con más de 100 estaciones repartidas en Sonora, en este caso se usan 63 debido a que estas cuentan con mayor cantidad de mediciones temporales. En este trabajo se contemplaron 68 meses para cada estación que van desde enero de 2015 hasta agosto de 2020.

En total se usaron 4,284 observaciones para entrenar y probar las ANNs, las formas de división se mencionan en las siguientes secciones, pero buscan tener una relación de 80-20.

3. Metodología

Las ANNs buscan seguir o imitar el proceso de aprendizaje del cerebro humano, que sirve para resolver problemas complejos, no lineales [2]. A diferencia de los enfoques más convencionales, la base de las ANNs son los datos dados a la red para estimar la variable objetivo (o clases en un problema de clasificación).

Una ANN tiene la estructura que se presenta en la siguiente figura que consiste en una cierta cantidad de inputs o variables que se alimenta a la red, cierta cantidad de capas escondidas que se forman por n cantidad de neuronas (cada capa puede tener una diferente cantidad de neuronas).

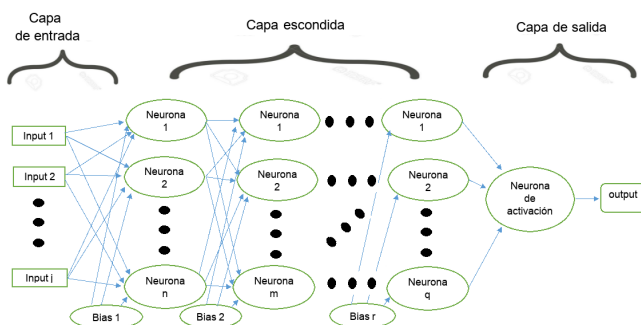


Figura 2: Mapa mental de una ANN

Además a cada capa escondida se le agrega una bias que es una constante que alimenta a todas las neuronas de dicha capa, una capa de salida que consiste en este caso en una neurona y toma de los inputs a los outputs de la última capa escondida y, por último, se comparan los datos con la variable objetivo.

Cabe mencionar que el mapa mental de la ANN mostrado en la figura anterior, corresponde al caso en el que se busca predecir una variable objetivo, si existen más de una o es un problema de calificación, la forma de la ANN cambia en la capa de salida.

Las neuronas por su parte funcionan recibiendo una cantidad n de inputs, cada input se multiplica por un peso y se suman los valores, a esta sumatoria se le agrega una constante que se denomina bias que tiene el papel similar al de la constante en la regresión lineal, seguido se le activa una función de activación a la sumatoria que es el output de la neurona. En la siguiente figura se ve el esquema de una neurona.

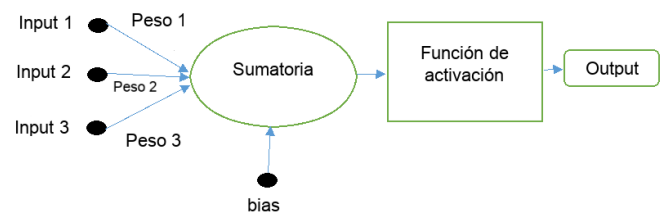


Figura 3: Mapa mental de una neurona

3.1. ANN para predecir la radiación solar

De acuerdo con [2, 3] las ANNs para predecir la radiación solar consisten en diferentes variables de entrada, una capa escondida y una capa de salida con solo una neurona. En el presente trabajo se usa una ANN de la forma presentada en la Figura 4.

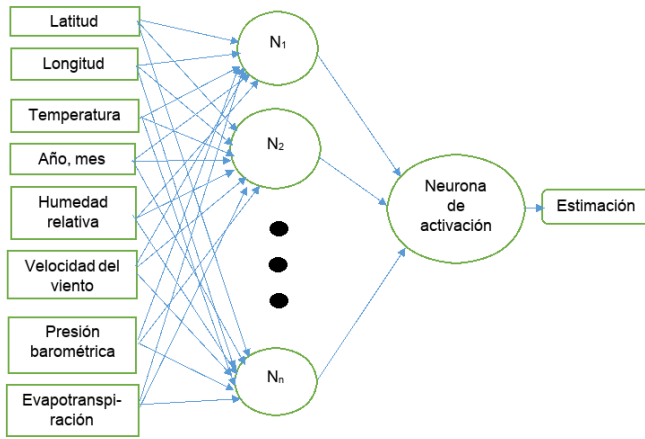


Figura 4: ANN para estimar la radiación solar

La función de activación que se aplica en las neuronas es la logística, que tiene la forma:

$$\text{Logística}(x) = \frac{1}{1 + \exp[-x]}$$

El rango de la función logística para predecir probabilidades de una variable objetivo es de 0 a 1, por lo que es necesario normalizar las variables en este rango, que se logra con la función [2]:

$$X_N = 0.8 \left(\frac{X - X_{\min}}{X_{\max} - X_{\min}} \right)$$

Donde X_N es la variable normalizada en el rango de 0.1 a 0.9, X es la variable original, X_{\min} es el mínimo de X y X_{\max} el máximo.

3.2. Métricas de medición

Para encontrar los modelos mejores para predecir la radiación solar, se usó el error cuadrático promedio (ECP) que se define como:

$$\text{ECP} = \frac{1}{N} \sum_n (y_n - f_n)^2$$

También el error absoluto promedio (EAP) que tiene la siguiente forma:

$$\text{EAP} = \frac{1}{N} \sum_n |y_n - f_n|$$

Y la bondad de ajuste (R^2):

$$R^2 = \frac{\sum_n (y_n - f_n)^2}{\sum_n (y_n - \bar{y})^2}$$

Donde y_n es la variable objetivo, f_n es el estimado de la variable objetivo y \bar{y} es el promedio de y .

4. Resultados

El entrenamiento y prueba se realizó separando aleatoriamente 50 estaciones (79.37 %) para entrenar y 13 (20.63 %) para realizar la prueba del modelo. Un aspecto que vale la pena mencionar, es que se probaron diferentes algoritmos de optimización con distintas librerías y el que se encontró mejor fue el de LM con 12 neuronas.

Otros aspectos a denotar es que la selección de la muestra de prueba se realizó manualmente debido a que existen zonas con mayor densidad de estaciones y se considero importante mantener las zonas con pocas estaciones en el proceso de entrenamiento.

También cabe mencionar que la validación no se pudo agregar debido a que la mayoría de las librerías no tienen dicha opción para los algoritmos seleccionados (con la excepción de Keras [12]).

Para comparar el rendimiento del optimizador de LM, se usaron diferentes algoritmos con el mismo número de neuronas. Las características computables de los diferentes algoritmos de optimización de las ANNs se describen en las siguientes tablas.

Algunas de las características que pudieron ser especificadas en el algoritmo LM de [4] se

presentan en la siguiente tabla. El valor inicial de μ se usó de acuerdo con el valor reportado por [3] pero los valores de crecimiento/decrecimiento no pudieron ser modificados acordemente por restricciones de la librería, tampoco fue posible ajustar el valor mínimo de eficiencia del gradiente.

Característica	Parámetro
μ	0.001
Tasa de crecimiento de μ	1
Tasa de decrecimiento de μ	1
Tipo de error a minimizar	Error cuadrático
Máximo número de iteraciones	NE
Tolerancia mínima	NE
Shuffle data	True

Tabla 1: Características del modelo LM

Nota: NE es no especificado

Como se mencionó anteriormente L-BFGS es un optimizador cuasi newtoniano que puede converger más rápido y tener un mejor rendimiento para bases de datos “pequeñas” [11]. Cabe mencionar que [11] es una de las librerías que permite menos alteraciones/modificaciones, específicamente para este algoritmo. Prácticamente solo permite cambiar la tasa de aprendizaje. Las características se pueden ver en la siguiente tabla.

Característica	Parámetro
Tasa de aprendizaje	0.001
Tipo de error a minimizar	Error cuadrático
Máximo número de iteraciones	500
Tolerancia mínima	No aplica
Shuffle data	No aplica

Tabla 2: Características del modelo L-BFGS

En el caso del método Nadam, [12] permite modificar la tasa de aprendizaje, la tasa de decrecimiento exponencial del primer momento estimado β_1 y de la norma ponderada infinita β_2 , una constante pequeña de estabilidad ϵ , así como el número de iteraciones. Las características se muestran en la siguiente tabla.

Característica	Parámetro
Tasa de aprendizaje	0.01
β_1	0.9
β_2	0.999
ϵ	1×10^{-8}
Tipo de error a minimizar	Error cuadrático
Máximo número de iteraciones	100
Tolerancia mínima	No aplica
Shuffle data	No aplica

Tabla 3: Características del modelo Nadam

En el caso del DG, la eficiencia mínima del gradiente no se consideró debido a que al utilizar una de magnitud 10^{-5} , es un principio fue debido a problemáticas con el código, pero ya que se aplicó se sospechó que la R^2 era menor que al no ponerse, por lo que se decidió no agregarla.

Característica	Parámetro
Máximo número de iteraciones	1,000
Tasa de aprendizaje	0.01
Tiempo maximo de entrenamiento	inf
Eficiencia mínima del gradiente	NE
Condición de validación	$R^2 \leq 1$

Tabla 4: Características del modelo DG

Como se puede ver en la tabla anterior, se

condiciono el código a que el R^2 sea menor o igual a 1. Lo anterior para evitar máximos locales o global, en los cuales este coeficiente resulta mayor que 1 y, por ende, no son modelos adecuados para modelar la radiación solar.

4.1. Métricas de medición de las diferentes ANNs

En esta sección se describen los mejores modelos que se encontraron usando las tres métricas de medición mencionadas anteriormente. Un primer aspecto a denotar es que el modelo con resultados óptimos se eligió usando como referencia su capacidad predictiva, es decir, sus resultados en los datos de prueba.

La siguiente tabla enseña los resultados para los algoritmos de optimización en los datos de entrenamiento. El modelo con mejores resultados en este caso es el LM, seguido por el de L-BFGS.

ANN	ECP	EAP	R^2
LM	0.0003	0.0147	0.9789
L-BFGS	0.0018	0.0250	0.8951
Nadam	0.0026	0.0374	0.8453
DG	0.0026	0.0386	0.8448

Tabla 5: Diferentes métodos de optimización para las ANNs de 12 nodos: entrenamiento

En el caso de los datos de prueba, se obtienen resultados muy similares. En el sentido de que el optimizador de LM es el que tiene las mejores métricas de medición en los tres casos, seguido por el de L-BFGS.

Un primer aspecto a denotar es que, en general, se obtuvieron mejores métricas de medición para los datos de prueba, aunque esto varía un poco con cada algoritmo.

ANN	ECP	EAP	R^2
LM	0.0027	0.0414	0.9432
L-BFGS	0.0029	0.0435	0.9384
Nadam	0.0035	0.0458	0.9268
DG	0.0047	0.0566	0.9031

Tabla 6: Diferentes métodos de optimización para las ANNs de 12 nodos: prueba

5. Conclusiones y discusión

En los métodos cuasi newtonianos (L-BFGS, Nadam y LM), así como en el de descenso de gradiente, se encontró consistencia para los ECP y EAP. En el sentido de que son menores para los datos de entrenamiento y ligeramente mayores para los de prueba. El optimizador LM parece el más consistente en estas métricas, especialmente en la ECP.

Por otro lado, la bondad de ajuste resultó tener una lógica contradictoria en la predicción de datos de entrenamiento y prueba, ya que fueron mayores para estos últimos. Con la excepción del algoritmo LM.

Lo anterior indica que las métricas de optimización del error cuadrado parecen tener resultados óptimos en la minimización de la ECP y EAP, pero no necesariamente se maximiza la bondad de ajuste. Aunque en algunos casos se obtuvieron bondades de ajuste más altas para los datos de prueba, lo que podría indicar una buena capacidad de predicción de los modelos.

Cabe destacar que los resultados varían dependiendo de la muestra elegida para la prueba de los algoritmos de optimización, pero mantienen características similares con los mostrados en este trabajo.

Referencias

- [1] CESAVE - SIAFESON (2020). Red de Estaciones Meteorológicas Automáticas de Sonora [online] www.siafeson.com/remas
- [2] Solmaz, O. & Ozgoren, M. (2012). Prediction of Hourly Solar Radiation in Six Provinces in Turkey by Artificial Neural Networks. *Journal of Energy Engineering*. 138, 194-204.
- [3] Premalatha, N. & Arasu, A.V. (2016). Prediction of solar radiation for solar systems by using ANN models with different back propagation algorithms. *Journal of applied research and technology*. 14(3).
- [4] Neupy (2019). `neupy.algorithms.LevenbergMarquardt`. Neupy. [online] <http://neupy.com/modules/generated/neupy.algorithms.LevenbergMarquardt.html>
- [5] Amini, S. & Amin Asoodar, M. (2016). Selecting the most appropriate tractor using Analytic Hierarchy Process – An Iranian case study. *Information Processing in Agriculture*. 3, 223–234.
- [6] Huang, G., Li, Z., Li, X., Liang, S., Yang, K., Wang, D., & Zhang, Y. (2019). Estimating surface solar irradiance from satellites: Past, present, and future perspectives. *Remote Sensing of Environment*, 233, 111371.
- [7] Aguiar, L.M., Pereira, B., David, M., Diaz, F. & Lauret, P. (2015). Use of satellite data to improve solar radiation forecasting with Bayesian Artificial Neural Networks. *Sol. Energy* 122, 1309–1324.
- [8] Linares-Rodriguez, A., Ruiz-Arias, J.A., Pozo-Vazquez, D. & Tovar-Pescador, J. (2013). An artificial neural network ensemble model for estimating global solar radiation from Meteosat satellite images. *Energy* 61, 636–645.
- [9] Voyant, C., Notton, G., Kalogirou, S., Nivet, M.L., Paoli, C., Motte, F. & Fouilloy, A. (2017). Machine learning methods for solar radiation forecasting: a review. *Renew. Energy* 105, 569–582.
- [10] Senkal, O. (2010). Modeling of solar radiation using remote sensing and artificial networks in Turkey. *Energy*. 35, 4795-4801.
- [11] Scikit-learn developers (2020). `sklearn.neural_network.MLPRegressor`. Scikit-learn developers. [online] https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html
- [12] Keras (s.f.). Nadam. Keras. [online] <https://keras.io/api/optimizers/Nadam/>