

Configuración de red neuronal para el análisis de polen

Leyra Martínez, Pedro Arguijo, Antonio Hiram Vázquez López,
Roberto Ángel Meléndez Armenta

Tecnológico Nacional de México/Instituto Tecnológico Superior de Misantla,
Misantla, Ver., México
leyramartinez.lm@gmail.com, pedro_arguijo@excite.com,
jahvazquez@itsm.edu.mx, ramelendeza@itsm.edu.mx

Resumen. En la palinología se han aplicado con éxito las redes neuronales para la clasificación de granos de polen. Con este fin se seleccionó RPROP como algoritmo de entrenamiento de la red neuronal para la clasificación de un dataset previamente reportado, se consideraron las variantes con y sin retroceso de peso. Se utilizó un algoritmo genético para optimizar la topología de la red neuronal (número de capas ocultas y número de neuronas por capa), razones de entrenamiento y prueba, pesos iniciales y número de iteraciones. Las topologías encontradas y reportadas en este documento exhiben resultados de clasificación superiores a los previamente reportados. Aunque no es el objetivo final, se pretende que este desarrollo sea útil en la identificación de los tipos de polen en la miel producida en la región de Misantla.

Palabras clave: palinología, dataset, red neuronal, RPROP.

Neural Network Configuration for Pollen Analysis

Abstract. In palynology, neural networks have been successfully applied for the classification of pollen grains. For this purpose, RPROP was selected as training algorithm of a neural network for the classification of a dataset previously reported. The RPROP variants with and without weight backtracking were considered. A genetic algorithm was used to optimize the ANN topology (number of hidden layers and number of neurons per layer), training and test ratios, initial weights and number of iterations. The topologies found and reported in this document shows better classification results than those previously reported. Although it is not the final goal, this development is intended to be useful in identifying the pollen types in honey produced in the Misantla region.

Keywords: palynology, dataset, neural network, RPROP.

1. Introducción

La palinología es una disciplina de la botánica dedicada al estudio del polen y las esporas [1], se centra fundamentalmente en el análisis de la morfología externa que presenta patrones estructurales diferentes al de las variaciones en la exina, que es la pared externa de los granos de polen. El estudio y análisis microscópico de su simetría,

apertura en las paredes, contorno, forma, tamaño, etc., tiene un valor taxonómico y permite distinguir diferentes taxones a distintos niveles: familia, géneros, especies.

El estudio de los granos de polen resulta ser una tarea difícil, en sus diferentes fases, a partir de pequeñas muestras microscópicas. El análisis de estos son una fuente importante de información para muchas aplicaciones científicas e industriales, lo que hace de la palinología una herramienta valiosa para diversas áreas del conocimiento [2].

Desde hace tiempo se han propuesto métodos de sistemas semiautomáticos o automáticos para el análisis de polen basados en técnicas de análisis de imágenes e inteligencia artificial, en particular redes neuronales y máquinas de soporte vectorial [3-6]. En términos generales, las metodologías propuestas segmentan la imagen y extraen características de textura tanto en imágenes de intensidad, como en algún espacio de color. Santiago Tello Mijares y Francisco Flores [6] plantean una metodología para segmentar granos de polen superpuestos. El método separa el polen en función del proceso de cambio medio y la erosión por mínimos basados en la serie de Fibonacci.

De la imagen segmentada extraen características de forma, y del espacio de color CIELab obtienen características de textura de primer y segundo orden. Para el reconocimiento de los granos de polen consideran tres clasificadores: bosque de árboles al azar, perceptrón multicapa y red de Bayes. Encontrando resultados de clasificación del 96.2% y 96.1% en sensibilidad y exactitud con un perceptrón multicapa.

En este documento, se analiza el desempeño de tres topologías de red neuronal entrenadas con el algoritmo RPROP con y sin retroceso de peso para la clasificación de un dataset de granos de polen, el cual consta de 12 clases palinológicas de polen con 618 instancias [6]. Las topologías mencionadas se obtuvieron utilizando un algoritmo genético, el cual determina la arquitectura adecuada que maximiza la exactitud. En la sección 2 se describe RPROP y las topologías utilizadas. El dataset se aborda en la sección 3 y se muestran los resultados obtenidos en la sección 4.

2. Marco teórico

2.1. Redes neuronales

Una red neuronal artificial es un sistema de procesamiento de información que funciona de manera similar a las redes neuronales biológicas [7]. La base de estos modelos son los siguientes supuestos:

- El procesamiento de la información se produce en un gran número de elementos simples llamadas neuronas.
- Las señales se transmiten entre las neuronas a lo largo de los enlaces de conexión.
- A cada enlace de conexión se le asigna un peso que multiplica la señal transmitida.
- Cada neurona aplica una función de activación en su entrada de red (que es la suma de las señales de entrada ponderadas) para obtener su señal de salida.

Existen dos estructuras en el desarrollo de las redes neuronales, redes de una sola capa y redes multicapa. En la Fig. 1 se muestra un ejemplo de una red neuronal, en la cual se indican las interconexiones de la red, así como los pesos entre estas.

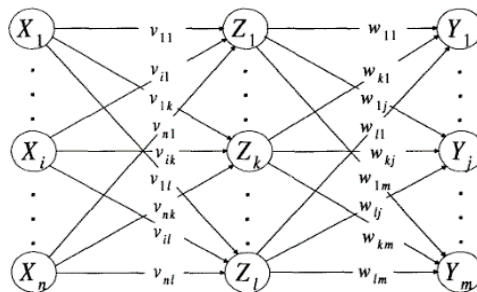


Fig. 1. Estructura de una red neuronal multicapa.

2.2. RPROP

Riedmiller y Braun [8] introdujeron RPROP como una extensión del algoritmo clásico de entrenamiento de propagación hacia atrás (*backpropagation*). RPROP reemplaza la tasa de aprendizaje constante ρ por tamaños de pesos adaptativo Δ_{ij} que se controlan durante el aprendizaje de los pesos w_{ij} . El paso del tamaño Δ_{ij} de los pesos se actualiza como:

$$\Delta_{ij}^{(t-1)} = \begin{cases} \Delta_{ij}^{(t-1)} \cdot \eta^+ & \text{if } \frac{\partial E^{(t)}}{\partial w_{ij}} \cdot \frac{\partial E^{(t-1)}}{\partial w_{ij}} > 0 \\ \Delta_{ij}^{(t-1)} \cdot \eta^- & \text{if } \frac{\partial E^{(t)}}{\partial w_{ij}} \cdot \frac{\partial E^{(t-1)}}{\partial w_{ij}} < 0 \\ \Delta_{ij}^{(t-1)} & \text{else} \end{cases} \quad (1)$$

con parámetros $0 < \eta^- < 1 < \eta^+$, que especifica la magnitud de adaptación del tamaño de paso, si el signo del gradiente $\frac{\partial E^{(t-1)}}{\partial w_{ij}}$ cambia. Donde E indica el error a la salida de la red. Los pesos se actualizan con

$$\Delta w_{ij}^{(t)} = -\text{sin} \left(\frac{\partial E^{(t)}}{\partial w_{ij}} \right) \cdot \Delta_{ij}^{(t)} \quad (2)$$

cada vez que la derivada parcial de la función de error con respecto al peso correspondiente cambia su signo, indica que el valor de la última actualización fue demasiado grande y el algoritmo ha saltado por encima de un mínimo local. El valor de actualización Δ_{ij} se reduce en un factor η^- . Si la derivada conserva su signo, el valor de actualización aumenta ligeramente con el factor η^+ para acelerar la convergencia en regiones poco profundas.

Una de las ventajas del algoritmo RPROP, en comparación con la propagación hacia atrás, es que la magnitud de la derivada parcial no afecta la actualización de peso y depende sólo de los signos de la derivada parcial. Por lo tanto, permite una convergencia más rápida de lo que puede hacer la propagación hacia atrás. RPROP

realiza una adaptación directa de la etapa de ponderación basada en la información del gradiente local. En la Tabla 1, se muestra una comparación entre el algoritmo de entrenamiento *Backpropagation* y su variante RPROP [9].

RPROP considera propagación hacia atrás con y sin retroceso de peso, definida por los algoritmos RPROP⁺ y RPROP⁻, respectivamente [8].

Tabla 1. Comparación de algoritmos.

<i>Backpropagation</i>	<i>RPROP</i>
<ul style="list-style-type: none"> • Emplea un ciclo propagación-adaptación de dos fases. • Es iterativo. • Su convergencia es tardada. • No requiere la normalización de los vectores de entrada. • Es capaz de generalizar, clasificando correctamente patrones ruidosos o incompletos. 	<ul style="list-style-type: none"> • Su escala de requisitos de tiempo y memoria es lineal con el número de parámetros a optimizar. • Proporciona un entrenamiento y una tasa de convergencia más rápido. • Es robusto. • Es considerado como uno de los mejores métodos de aprendizaje entre los algoritmos. • Es complicado de programar.

2.3. Topología

Un aspecto importante de las aplicaciones de redes neuronales es la determinación de una topología adecuada, es decir, el número de capas ocultas y el número de neuronas por capa oculta, razones de entrenamiento y prueba, pesos iniciales y número de iteraciones (épocas). Si bien existen algunas reglas generales, no se puede determinar una solución óptima a este problema en un tiempo razonable. Generalmente, una topología de red se selecciona por prueba y error, optando por la mejor.

Resulta complejo el hecho de descubrir cuántas capas ocultas deben utilizarse, cuántas neuronas debe tener cada capa oculta, qué tipo de función de activación utilizar, qué algoritmo y tasa de aprendizaje debe emplearse para el entrenamiento de la red neuronal. Los algoritmos genéticos (GA) junto con las redes neuronales se han utilizado en diversas aplicaciones [10-13]. Estos se han empleado para encontrar la topología de las capas ocultas, el número de conectividad y parámetros de entrenamiento óptimos para analizar datos biomédicos relacionados con padecimientos cardíacos [13]. Al emplear GA se reemplazan los métodos de prueba y error para determinar la topología óptima de la red neuronal y aumentar la velocidad, precisión y eficiencia de la fase de entrenamiento para así obtener un mayor grado de exactitud en la clasificación. Los GA se utilizan para la optimización de múltiples variables y tienen la capacidad de resolver una amplia gama de problemas de optimización, con el uso de tres operaciones simples: selección, cruce y mutación [13]. Consecuentemente, al combinar algoritmos genéticos y redes neuronales se obtienen arquitecturas óptimas de redes en menos tiempo. Con este enfoque, además de reducir el tiempo de ejecución al ejecutar el algoritmo logrando puntos óptimos. Para este trabajo, dentro del espacio de búsqueda del algoritmo genético se consideró maximizar la exactitud de las configuraciones de la red neuronal. En la Tabla 2 se muestran las 3 configuraciones las

cuales arrojan la mayor exactitud con distintas razones de entrenamiento y prueba. Cabe mencionar que cada una de estas configuraciones tienen 33 neuronas de entrada y 12 neuronas de salida.

Tabla 2. Configuraciones seleccionadas.

	Capa de entrada	Capas ocultas			Épocas	Razón de entrenamiento y prueba
		Primera	Segunda	Tercera		
Configuración 1	33	42	28	18	26	70-30
Configuración 2	33	11	21	45	31	90-10
Configuración 3	33	31	44	19	37	80-20

Como se puede observar en la Tabla 2, el número de neuronas de las capas ocultas para cada una de las configuraciones consideradas son demasiadas, por lo cual no mostramos figura alguna de las topologías analizadas ya que no se alcanzan a distinguir las capas ocultas con sus respectivas neuronas, sus interconexiones y pesos w_{ij} .

3. Dataset

El dataset empleado en este documento consta de 618 instancias divididas en 12 clases diferentes de polen [6]. En la Figura 2 se muestra una imagen representativa de cada una de las clases de polen que se analizan. Las clases que tiene el dataset, con su respectivo nombre palinológico y el número de observaciones por clase se muestra en la Tabla 3.



Fig. 2. Granos de polen por clases del 1- 12 y se representan así: 1-Huizache, 2-Alfalfa,3-Ortiga, 4-Pasto, 5-Hierba, 6-Morera, 7-Pacana, 8-Oliva, 9-Mesquite, 10-Sauce, 11-Pimienta, 12-Sorgo.

Las características reportadas en el dataset se extrajeron de imágenes segmentadas en el espacio de color CIELab. CIELab es un modelo de color perceptual uniforme en el que la componente L representa la luminosidad de un objeto desde el negro (0) al blanco (100), a representa la variación del verde (-) al rojo (-), y b representa la variación del azul (-) al amarillo (+). Una variación numérica de estos valores corresponde aproximadamente a la misma cantidad de cambio percibido visualmente. De las imágenes segmentadas se obtuvieron las características de forma, y del espacio de color CIELab las características de textura de primero y segundo orden, descritas

por matrices de coocurrencia de niveles de gris. Dichos datos se muestran en la Tabla 4. Como se mencionó son un total de 33 características, 6 de forma, 15 de textura de primer orden y 12 de textura de segundo orden.

Tabla 2. Clasificación palinológica del conjunto de datos de imágenes de especie de polen.

Clase	Nombre polen	Observaciones
1	Huisache	46
2	Alfalfa	46
3	Ortiga	56
4	Pasto	37
5	Hierba	42
6	Morera	56
7	Pacana	32
8	Oliva	52
9	Mesquite	80
10	Sauce	64
11	Pimienta	73
12	Sorgo de Alepo	34

Tabla 3. Características del dataset.

Características		
Forma	Textura (CIE-L*a*b*)	
	Primer orden	Segundo orden
Ancho Alto Perímetro Área Circularidad Compactancia	Promedio Media Varianza Desviación estándar Entropía	Contraste Correlación Energía Homogeneidad

4. Resultados

Para probar cada una de las topologías descritas en la Tabla 1 se utilizó el software estadístico R y su interfaz gráfica de usuario RStudio, en una laptop con procesador i3, memoria RAM de 8GB, un sistema operativo de 64 bits.

En la Tabla 5 se muestra un ejemplo de la matriz de confusión obtenida en la configuración 1 utilizando RPROP+, con la proporción de muestreo 70-30, o sea, 70% del dataset para entrenar la red neuronal y 30% para realizar pruebas. Dicha configuración obtuvo una exactitud del 99.35%. En la matriz de confusión el color rojo indica que existió un error en la clasificación, por ejemplo, la alfalfa se confundió tanto con el pasto y con el mesquite.

En la tabla 6 se muestran los resultados obtenidos de cada una de las configuraciones mencionadas previamente, cabe mencionar únicamente se muestran los resultados de las proporciones que obtuvieron la mejor exactitud. Como se mencionó los algoritmos de entrenamiento de las topologías fueron Rprop+ y Rprop-.

Tabla 4. Matriz de confusión 70-30 con el algoritmo Rprop+.

		Referencia												Total	
		70%	1	2	3	4	5	6	7	8	9	10	11		12
Clasificación	1	46	0	0	0	0	0	0	0	0	0	0	0	0	46
	2	0	46	0	1	0	0	0	0	1	0	0	0	0	48
	3	0	0	56	0	0	0	0	0	0	0	0	0	0	56
	4	0	0	0	36	0	0	0	0	0	0	1	0	0	37
	5	0	0	0	0	42	0	0	0	0	0	0	0	0	42
	6	0	0	0	0	0	55	0	0	0	0	0	0	0	55
	7	0	0	0	0	0	0	32	0	0	0	0	0	0	32
	8	0	0	0	0	0	0	0	52	0	0	0	0	0	52
	9	0	0	0	0	0	0	0	0	79	0	0	0	0	79
	10	0	0	0	0	0	1	0	0	0	64	0	0	0	65
	11	0	0	0	0	0	0	0	0	0	0	72	0	0	72
	12	0	0	0	0	0	0	0	0	0	0	0	34	0	34
Total		46	46	56	37	42	56	32	52	80	64	73	34	618	

En las configuraciones 1 y 2, el algoritmo Rprop con retroceso de peso (Rprop+) obtuvo el mejor resultado de exactitud alcanzando valores del 99.35% al 99.83%, mientras que en la configuración 3 el algoritmo Rprop sin retroceso de peso (Rprop-) alcanzó una exactitud del 99.35%, siendo este mayor que el resultado de exactitud de Rprop+ para esta configuración.

La validez de los resultados se comprobó a través de validación cruzada, obteniendo resultados de exactitud entre el 94 y 97%, tal como se muestra. La validación cruzada es un método estadístico utilizado para evaluar, comparar y evitar el sobreentrenamiento de algoritmos de aprendizaje, mediante el uso del subconjunto de datos etiquetados de validación.

En la validación cruzada la información se divide en dos subconjuntos: uno se utiliza para el entrenamiento del modelo, mientras que el otro se utiliza para validarlo, su forma más básica es conocida como k-folds donde los subconjuntos de entrenamiento y validación se intercambian a través de las iteraciones de tal manera que cada subconjunto de datos pueda ser validado.

Tabla 5. Resultados.

	Razón de entrenamiento y prueba	Rprop +	Rprop -	Validación Cruzada
Configuración 1	70-30	99.35%	98.70%	95.40
Configuración 2	90-10	99.83	99.19	94.03
Configuración 3	80-20	98.70	99.35	96.61

5. Conclusión

En este trabajo, se consideraron tres topologías de redes neuronales entrenadas con el algoritmo RPROP para realizar la clasificación polen. Los resultados obtenidos con éstas son superiores a los obtenidos previamente [6], pues en ambos casos se utiliza el mismo dataset. Nuestras topologías arrojan resultados de exactitud por encima del 98%, mientras que en los resultados reportados la exactitud es del 96%, utilizando también

redes neuronales artificiales, pero se desconoce la topología que ellos emplearon para hacer sus respectivos análisis.

Referencias

1. Polen: Propiedades, utilización y beneficios, <https://mejorconsalud.com/polen-propiedades-utilizacion-beneficios/>. Último acceso: 2019/20/03
2. Fundación UNAM Ecología, <http://www.fundacionunam.org.mx/ecologia/la-ciencia-del-polen/>. Último acceso: 2019/03/20
3. Sánchez Mesa, J.A., Galán, C., Martínez Heras, J.A., Hervás Martínez, C.: The use of a neural network to forecast daily grass pollen concentration in a Mediterranean region southern part of the Iberian Peninsula. *Clinical & Experimental Allergy*, vol. 32, no. 11, pp. 1606–1612 (2002)
4. Li, P., Treloar, W.J., Flenley, J.R., Empson L.: vf: vf Towards automation of palynology 2: the use of texture measures and neural network analysis for automated identification of optical images of pollen grains. *Journal of Quaternary Science: Published for the Quaternary Research Association*, vol. 19, no. 8, pp. 755–762 (2004)
5. Li, P., Flenley, J.R.: Pollen texture identification using neural networks. *Grana*, vol. 38, no. 1, pp. 59–64 (1999)
6. Tello-Mijares, S., Flores, F.: A Novel Method for the Separation of Overlapping Pollen Species for Automated Detection and Classification. *Computational and Mathematical Methods in Medicine*, Vol. 2016, Article ID 5689346, 12 pages, (2016)
7. Friedman, M., Kandel, A.: Introduction to Pattern Recognition Statistical, Structural, Neuronal and Fuzzy Logic Approaches. World Scientific Pub Co Inc., March 2, (1999)
8. Riedmiller, M., Braun, H.: Direct adaptive method for faster backpropagation learning: the RPROP algorithm. In: *Proceedings of the IEEE International Conference on Neural Networks*, pp. 586–591 (1993)
9. Reddy, C.S., KVSVN, R.: An optimal neural network model for software effort estimation. *Int. J. of Software Engineering, IJSE*, vol. 3, no. 1, pp. 63–78. (2010)
10. Venkatesan, D., Kannan K., Seravanan, R.: A genetic algorithm-based artificial neural network model for the optimization of machining processes. *Neural Computing and Applications*, vol. 18, no. 2, pp. 135–140 (2009)
11. Karimi, H., Yousefi, F.: Application of artificial neural network-genetic algorithm (ANN-GA) to correlation of density in nanofluids. *Fluid Phase Equilibria* 336, pp. 79–83 (2012)
12. Tonnizam Mohamad, E., Shirani Faradonbeh, R., Jahed Armaghani, D., Monjezi, M., Abd Majid, M.Z.: An optimized ANN model based on genetic algorithm for predicting ripping production. *Neural Computing and Applications* 28, Supplement 1, pp. 393–406 (2017)
13. Rubio Escamilla, U., Arguijo Hernández, S.P.: Configuración de arquitectura de una RNA mediante algoritmos genéticos. *Academia Journals*, vol.10, no. 3, pp. 3058–3064 (2018)