Redes neuronales y su aplicación en la clasificación de patrones

Edgar Alamilla Jiménez*
Addy Margarita Bolívar Cimé
Edilberto Nájera Rangel

Universidad Juárez Autónoma de Tabasco
linking_1990@hotmail.com*

1. Introducción

Las redes neuronales tienen sus orígenes en encontrar representaciones matemáticas del procesamiento de información en sistemas biológicos tales como el cerebro. Rosenblatt (1958) propuso el primer modelo precursor de redes neuronales, el perceptrón. Sin embargo, en 1969 éste tenía capacidades muy limitadas, lo que trajo como consecuencia que en la década de 1970 esta área de investigación fuera casi abandonada; no fue sino hasta la década de 1980, con el uso de hardware computacional, que se dio un auge en la investigación de redes neuronales, el cual persiste hasta el día de hoy. En este cartel se dará una explicación teórica de redes neuronales y se mostrará una aplicación de interés.

2. Objetivos

1. Explicar brevemente qué son las redes neuronales y presentar la implementación teórica para su aplicación a datos reales.
2. Mostrar un ejemplo de clasificación de patrones para exhibir una aplicación de redes neuronales.

3. Redes neuronales

Perceptrón multicapa

A continuación describimos brevemente lo que son las redes neuronales. Para ello describiremos el **discriminante lineal** o **perceptrón**, que tiene la regla de decisión

$$\phi(x) = \begin{cases} 0, \text{ si } \psi(x) \leq 1/2; \\ 1, \text{ en otro caso,} \end{cases}$$

basado en una combinación lineal $\psi(x)$,

$$\psi(x) = c_0 + \sum_{i=1}^{d} c_i x^{(i)} = c_0 + c^T x, \tag{1}$$

donde las c_i's son los pesos, $x = (x^{(1)}, \ldots, x^{(d)})^T$ y $c = (c_1, \ldots, c_d)^T$. Esto es llamado una **red neuronal sin capas ocultas**, ver Figura 1.

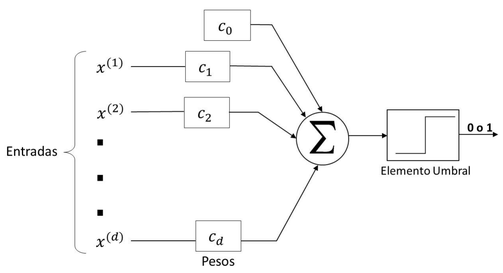

Figura 1: *El perceptrón de Rosenblatt. La decisión se basa en una combinación lineal de los componentes del vector de entrada.*

- En una **red neuronal** (hacia adelante), o **perceptrón multicapa**, con una capa oculta se tiene

$$\psi(x) = c_0 + \sum_{i=1}^{k} c_i \sigma(\psi_i(x)),$$

donde las c_i's son como antes y cada ψ_i es de la forma dada en (1): $\psi_i(x) = b_i + \sum_{j=1}^{d} a_{ij} x^{(j)}$ para algunas constantes b_i y a_{ij}; ver Figura 2.

- La función σ es llamada **sigmoide**, la cual se define como una función no decreciente con $\sigma \to -1$ cuando $x \downarrow -\infty$ y $\sigma \to 1$ cuando $x \uparrow \infty$. Algunos ejemplos de sigmoides son: el sigmoide umbral, el estándar o logístico, el arctangente y el gaussiano.

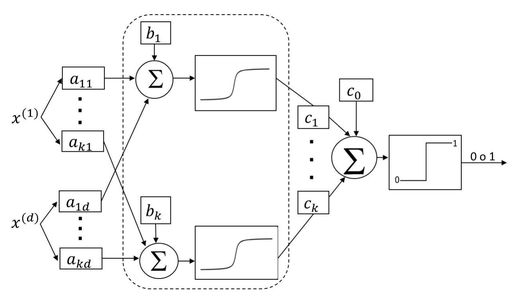

Figura 2: *Red neuronal con una capa oculta. Las neuronas ocultas se encuentran dentro del marco.*

4. Gradiente descendente

En discriminación los parámetros son de los discriminantes y están optimizados para minimizar el error de clasificación. Cuando $w = (w^{(1)}, \ldots, w^{(d)})^T$ denota el vector de pesos o el conjunto de parámetros y $\mathbf{E}(w|X)$ es el error con parámetros w para el conjunto de entrenamiento X dado, buscamos

$$w^* = \arg\min_w \mathbf{E}(w|X).$$

En muchos casos no existe una solución analítica y debemos recurrir a métodos de optimización iterativos. El método más empleado es el **gradiente descendente**. Cuando $\mathbf{E}(w)$ es una función diferenciable de un vector de variables, tenemos el **vector gradiente** compuesto de las derivadas parciales

$$\nabla_w \mathbf{E} = \left[\frac{\partial \mathbf{E}}{\partial w_1}, \frac{\partial \mathbf{E}}{\partial w_2}, \cdots, \frac{\partial \mathbf{E}}{\partial w_d}\right]^T;$$

el procedimiento del gradiente descendente para minimizar $\mathbf{E}$ comienza a partir de un w aleatorio y en cada paso actualiza w en la dirección opuesta del gradiente

$$\Delta w_i = -\eta \frac{\partial \mathbf{E}}{\partial w_i}, \forall i, \tag{2}$$

por lo que

$$w_i = w_i + \Delta w_i. \tag{3}$$

A η se le llama el **tamaño de paso** o **factor de aprendizaje** y determina cuanto moverse en esa dirección. Por ende el uso de un buen valor para η también es crítico, si es demasiado pequeño la convergencia puede ser demasiado lenta, pero un valor grande puede causar oscilaciones e incluso divergencia.

5. Algoritmo de Backpropagation

Entrenar a un perceptrón multicapa es lo mismo que entrenar a un perceptrón, la única diferencia es que ahora la salida es una función no lineal de las entradas debido a la función de base no lineal en las neuronas ocultas.

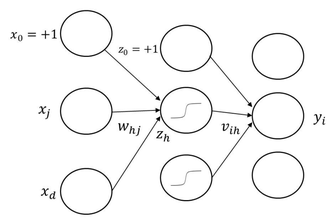

Figura 3: *Estructura de un perceptrón multicapa en la cual las x_j, $j = 0, \ldots, d$, son las entradas y las z_h, $h = 1, \ldots, H$, son las neuronas ocultas, donde H es la dimensión de este espacio oculto, z_0 es el intercepto de la capa oculta, las y_i, $i = 1, \ldots, k$, son las unidades de salida, las w_{hj} son los pesos de la primera capa, y las v_{ih} son los pesos de la segunda capa.*

Considerando las neuronas ocultas como entradas, la segunda capa es un perceptrón, y ya se ha visto cómo obtener los parámetros, v_{ij}, en este caso, dadas las entradas z_h. Para los pesos de la primera capa, w_{hj}, utilizamos la regla de la cadena para calcular la derivada parcial, ver Figura 3,

$$\frac{\partial \mathbf{E}}{\partial w_{hj}} = \frac{\partial \mathbf{E}}{\partial y_i} \frac{\partial y_i}{\partial z_h} \frac{\partial z_h}{\partial w_{hj}}.$$

Es como si el error se propagara desde la salida y de regreso a las entradas, y por lo tanto se le denominó **Backpropagation** (ver Alpaydin, 2014). Con la fórmula anterior se calculan las entradas del vector gradiente y se lleva a cabo el método del gradiente descendente.

6. Ejemplo de clasificación de patrones

- Se cuenta con datos de imágenes de rostros de varias personas en varias poses, los cuales son proporcionados por Mitchell, 1997.
- La base de datos consiste de imágenes de 20 personas diferentes, con aproximadamente 32 imágenes por persona variando su estado de ánimo (feliz, triste, enojado, neutral), la dirección en la que miraban (izquierda, derecha, de frente, arriba) y si portan o no gafas de sol, tal como puede verse en la Figura 4.
- En total se recopilaron 624 imágenes en escala de grises, cada una con una resolución de 64×60, con cada pixel de imagen descrito por un valor de la intensidad en la escala de grises entre 0 (negro) y 255 (blanco). Cada imagen fue transformada en un vector de entradas reales de dimensión $3840 = 64 \times 60$, mediante la concatenación de columnas.
- En cada análisis que se presenta se procedió a dividir de forma aleatoria en dos partes a la base de datos, el $70\,\%$ de los puntos conformaron los datos de entrenamiento (436 puntos) y el $30\,\%$ restante (188 puntos) conformaron los datos de prueba.

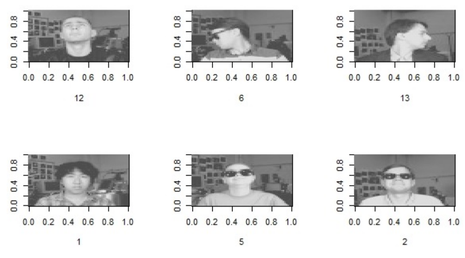

Figura 4: *Imágenes de rostros de varias personas de 64×60 pixeles en diferentes poses, diferentes estados de ánimos y si portan o no gafas de sol.*

- La motivación de este ejemplo es mostrar que la clasificación de sujetos puede ser automatizada mediante redes neuronales, y que el proceso de clasificar grandes cantidades de caras se puede efectuar en lapsos de tiempo relativamente cortos. El procesamiento es realizado por un equipo de computo y el trabajo de catalogar las caras se realiza de manera objetiva.
- De igual forma, la misma base de datos puede ser utilizada para clasificar a los sujetos de acuerdo a si portan o no gafas de sol en la imagen, para clasificar la dirección en la que miran, así como para clasificar sus estados de ánimo.

Clasificación de sujetos

- Para la clasificación de las caras de los sujetos, se procedió a reconocer a un sujeto en particular de entre los demás, para ello se etiquetó al sujeto de interés, en este caso al sujeto an2i con la etiqueta 1, y a los demás sujetos que conforman la base de datos con la etiqueta 2.
- Se ajustaron redes neuronales utilizando la función sigmoide logística y la función sigmoide arcotangente, en cada caso se consideró una capa oculta con una neurona oculta. Observamos que las dos tablas de confusión son iguales y que la proporción de error con ambas funciones de activación es 0.

Clasificación del uso de gafas de sol

- Para la clasificación de uso de gafas de sol de los sujetos, se procedió a etiquetarlos de la siguiente manera: "sin gafas de sol"=1, "con gafas de sol"=2.
- Se ajustaron redes neuronales utilizando la función sigmoide logística y la función sigmoide arcotangente, en cada caso se consideró una capa oculta con una neurona oculta. Observamos que las dos tablas de confusión son iguales. Una vez obtenidas las tablas de confusión, las cuales muestran la clasificación hecha de los sujetos de acuerdo si usan o no gafas de sol, se observó que la proporción de error con ambas funciones de activación fue 0.0106.

Clasificación de dirección de miradas

- Para el caso de clasificar la dirección en la que miran los sujetos, se etiquetó dicha dirección de la siguiente forma: izquierda=1, derecha=2, de frente=3 y arriba=4.
- Se ajustaron redes neuronales usando la función sigmoide logística y la función sigmoide arcotangente, en ambos casos se consideró una capa oculta y se varió de 1 a 6 el número de neuronas ocultas. Una vez obtenidas las tablas de confusión respectivas, las cuales muestran la clasificación de la dirección en la que miran los sujetos, se obtuvieron las proporciones de error de clasificación, las cuales se muestran en la Tabla 1.

Neuronas ocultas \ Función de activación	Logística	Arcotangente
1	0.2340	0.1649
2	0.0213	0.0213
3	0.0372	0.0372
4	0.0372	0.0372
5	0.0213	0.0213
6	0.0372	0.0372

Tabla 1. *Proporciones de error de clasificación considerando una capa oculta y variando de 1 a 6 neuronas ocultas.*

Clasificación de emociones

- Finalmente para clasificar las emociones de los sujetos, se etiquetaron las emociones de la siguiente manera: enojado=1, feliz=2, neutral=3 y triste=4.
- Se ajustaron redes neuronales usando la función sigmoide logística y la función sigmoide arcotangente. En cada caso se consideró una capa oculta y se varió el número de neuronas ocultas de 1 a 6. Una vez obtenidas las tablas de confusión respectivas, las cuales muestran la clasificación hecha de las emociones de los sujetos, se obtuvieron las proporciones de error de clasificación, las cuales se proporcionan en la Tabla 2

Neuronas ocultas \ Función de activación	Logística	Arcotangente
1	0.2819	0.2766
2	0.0638	0.1543
3	0.0319	0.0745
4	0.0266	0.0319
5	0.0213	0.0372
6	0.0266	0.0213

Tabla 2. *Proporciones de error de clasificación considerando una capa oculta y variando de 1 a 6 neuronas ocultas.*

7. Conclusiones

1. En el ejemplo de los datos de las imágenes de rostros de varias personas en varias poses, se observó que basta con aplicar una red neuronal de una capa oculta e ir variando el número de neuronas ocultas para todos los escenarios propuestos.
2. Cuando se consideró la clasificación de un sujeto de interés y la clasificación del uso de gafas de sol, bastó con aplicar una red neuronal de una capa oculta con una neurona oculta con cualquiera de las dos funciones de activación consideradas, para alcanzar una proporción de error de 0 y 0.0106, respectivamente.
3. Para el caso de clasificar de la dirección de miradas de las caras al considerar una neurona oculta, las proporciones de error con ambas funciones de activación fueron significativamente elevadas. No obstante, a partir de 2 neuronas ocultas se obtuvieron proporciones de error muy pequeñas y similares al aplicarlas con ambas funciones de activación.
4. Finalmente, para el caso de clasificar las emociones, al considerar una neurona oculta las proporciones de error con ambas funciones de activación son significativamente elevadas. Por otro lado a partir de 2 neuronas ocultas, es recomendable utilizar la función de activación logística debido a que su proporción de error es pequeña comparada con la función de activación arcotangente. Sin embargo, a partir de 3 neuronas ocultas con la función de activación arcotangente se puede observar una proporción de error pequeña.

Referencias

[1] L. Devroye, L. Györfi, and G. Lugosi. A Probabilistic Theory of Pattern Recognition. Stochastic Modelling and Applied Probability. Springer, New York, 2013.

[2] E. Alpaydin. Introduction to Machine Learning. Adaptive Computation and Machine Learning series. MIT Press, 2014.

[3] T. Mitchell. Machine Learning. McGraw-Hill international editions computer science series. McGraw-Hill Education, 1997.