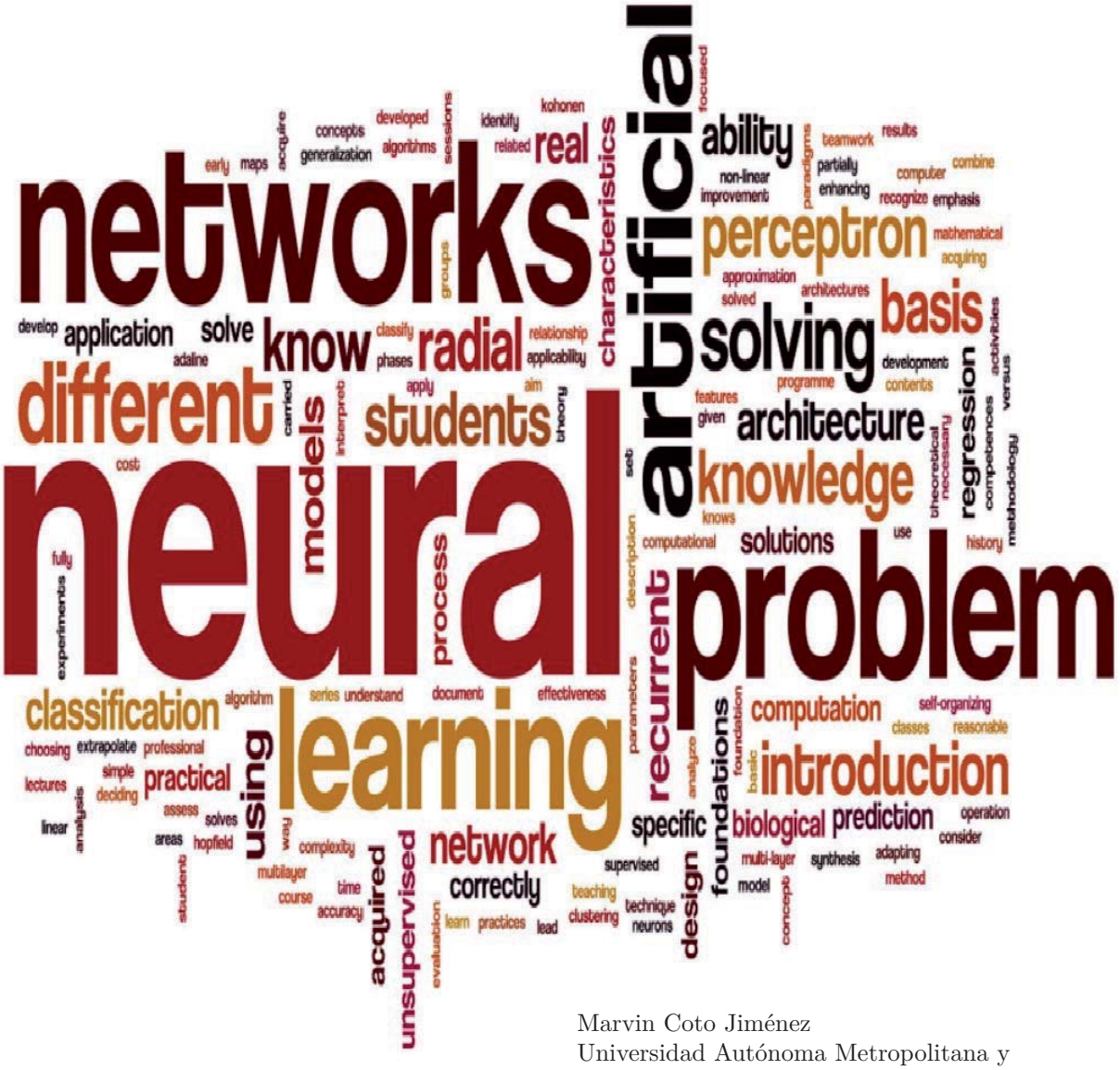
Ejemplo de generación y clasificación de bases de datos de imágenes simples



Marvin Coto Jiménez
Universidad Autónoma Metropolitana y
Universidad de Costa Rica
marvin.coto@ucr.ac.cr

Resumen—*Se documenta la creación de una base de datos para clasificación, que consiste de patrones de imágenes simples. Se presentan resultados de una clasificación de estas imágenes utilizando redes neuronales tipo perceptrón multicapa, así como las perspectivas que puedan tener su uso en entornos educativos.*

Palabras clave—*reconocimiento de patrones; inteligencia artificial; redes neuronales, perceptrón multicapa.*

I. INTRODUCCIÓN

El reconocimiento automático, la descripción, clasificación y agrupamiento de patrones son problemas importantes en diversas áreas de la ciencia y la ingeniería. Un patrón puede ser una imagen de huella digital, una letra escrita a mano o una señal de voz[1].

Dado un patrón, su reconocimiento o clasificación puede consistir en una clasificación supervisada, en la cual el patrón de entrada se define como miembro de una clase ya establecida, o bien de forma no supervisada, en que no se cuenta con esta clasificación previa. El proceso general que se debe seguir consta de tres partes:

1. Adquisición y procesamiento de los datos.
2. Representación o codificación de los datos.
3. Herramienta de decisión.

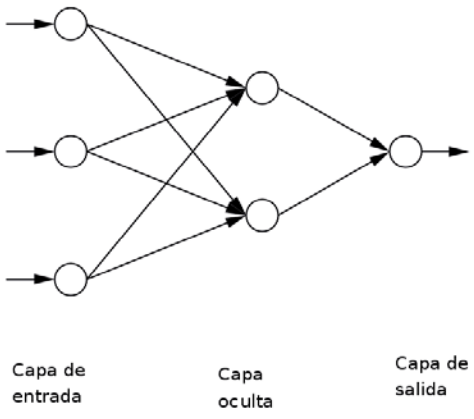
Las redes neuronales son herramientas de uso extendido en clasificación, por lo que son introducidas generalmente como una de las primeras técnicas de inteligencia artificial en cursos de educación superior. En las siguientes secciones se documenta una experiencia de creación de base de datos de imágenes simples y un ejemplo de clasificación por redes neuronales tipo perceptrón. Se plantean las experiencias semejantes que pueden ser de provecho en entornos de educación pues permiten apreciar algunas dificultades subyacentes a la creación de bases de datos, así como apreciar el potencial de herramientas de inteligencia artificial como redes neuronales en aplicaciones de clasificación. La simplicidad de los datos propuestos es tal, que puede desarrollarse en el contexto de un curso trimestral.

A este respecto, *Venayagamoorthy* [4], indica que incluso a nivel de pregrado, el reto principal de la enseñanza de redes neuronales es encontrar espacio dentro del currículo, pues si bien es difícil ofrecer un curso competo de este tema, con las herramientas adecuadas podría hacerse una introducción en un espacio de dos semanas. Esto aplica para muchos cursos de las carreras de ingeniería.

II. EL PERCEPTRÓN MULTICAPA

El perceptrón multicapa (PMC) es un modelo matemático inspirado en el funcionamiento de las neuronas biológicas, las cuales funcionan de forma paralela y no lineal. Se puede representar como un conjunto de capas ordenadas [2], empezando con una capa de entrada, una o más capas intermedias llamados ocultas, y una capa de salida. Cada capa tiene una cantidad fija de nodos (neuronas), que están conectados con los nodos de la siguiente capa. En la figura 1 se muestra un ejemplo de un PMC con 3 nodos en la capa de entrada, 2 nodos en la capa ocula y un nodo en la salida.

Figura 1: Ejemplo de PMC



Cada conexión tiene un valor numérico asociado, llamado peso. Cada peso está representado por wij, donde i es el nodo de salida y j el nodo de llegada. Cuando se recibe un vector de entradas, sus valores se propagan a través del PCM, por medio del cálculo de las salidas de cada nodo, el cual está dado por

$$y_j = f\left(\sum_i w_{ij} y_i + b\right)$$

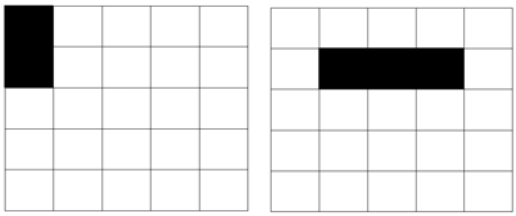
en la que y_i es la salida del nodo i de la capa anterior, y b es un coeficiente llamado umbral, usualmente inicializado como 1. f es llamada función de activación. En el caso de la implementación presentada en este trabajo, se utilizó la función sigmoide:

$$f : \mathbb{R} \to [0, 1], f(x) = \frac{1}{1 + e^{-x}}$$

El uso PMC en clasificación de datos, parte comúnmente de un conjunto de entrenamiento y otro de prueba. De ambos se conoce la clase a la que pertenece cada vector de valores. El primero se utiliza para ajustar los pesos, de manera que se obtenga la salida deseada (la pertenencia a la clase) para nuevos vectores. Para el ajuste de pesos existen varios algoritmos, en el presente trabajo se ha utilizado retropropagación. El conjunto de prueba es para verificar la eficacia del PMC ante un nuevo conjunto de datos.

III. GENERACIÓN DE LOS DATOS

Los datos fueron generados en su totalidad en una hoja de cálculo, y consisten en la codificación de imágenes de 5x5 píxeles, las cuales se suponen son representaciones de los resultados de la detección de un ente que puede ocupar 2 ó 3 píxeles consecutivos en la imagen. Los datos se clasifican entonces en resultados positivos de detección, como los que se esquematizan con píxeles cuadrados de la Figura 1, o bien negativos, como los de la Figura 2.



(A) (B)

Figura 2: Ejemplos de imágenes simples clasificadas como positivas.

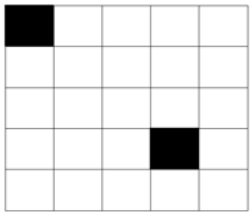


FIGURA 3: EJEMPLO DE IMAGEN CLASIFICADA COMO NEGATIVA.

La Figura 3, a pesar de que tiene cuadros que sí contienen detección de información, son considerados como ruido en el proceso, pues es necesario contar con un conjunto suficientemente grande de datos contrastantes, y no solamente una celda vacía. Para los datos, no se consideraron como imágenes sin detección positiva del patrón celdas inmediatamente diagonales, para tener suficiente contraste con las de detección positiva.

La codificación se realizó de la siguiente manera: Dada una imagen, como la de la Figura 1 A, se codifica con un 0 aquellas celdas donde no hay información, y con un 1 donde sí hay información. La Figura 3 muestra la numeración de las celdas propuesta:

x_1	x_2	x_3	x_4	x_5
x_6	x_7	x_8	x_9	x_{10}
x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
x_{16}	x_{17}	x_{18}	x_{19}	x_{20}
x_{21}	x_{22}	x_{23}	x_{24}	x_{25}

Figura 4: Numeración de celdas para todas las imágenes.

La siguiente matriz muestra la codificación de la Figura 1 A:

$$X_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Con la información de la matriz, los datos se cons-

truyen haciendo un vector de 25 entradas, a partir de las filas de ésta escritas de forma adyacente, y de forma consecutiva siguiendo la secuencia planteada de la Figura 4. En total cada tupla consistiría de un vector de 26 entradas, donde la última entrada representaría si hay detección positiva del patrón (con un 1) o negativa (con un 0).
El resumen de las características de los datos generados de forma manual se da a continuación:

- Número de tuplas: 408, 204 corresponden a detecciones positivas y 204 a detecciones negativas.
- Número de atributos: 25 (cada uno representa un píxel)
- Valores faltantes: Ninguno.
- Clases: 2.
- Descripción de las clases:

• Clase 0: No hay presencia de elemento en la imagen.
• Clase 1: Sí hay presencia de elemento en la imagen.

III. EJEMPLO DE USO

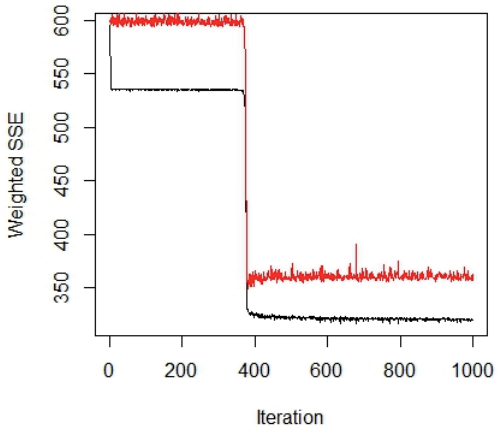
El problema que se plantea, es el entrenamiento y prueba de una red neuronal tipo PMC para la clasificación de los 408 patrones creados. Se utilizó un 70% de los datos (286) para el entrenamiento, y el restante 30% para las pruebas.

La implementación se realizó con el software R, y la adición del paquete RSNNS, el cual tiene implementado este tipo de red como una función, con las siguientes entradas básicas:

- Número de nodos.
- Número de iteraciones.
- Porcentaje de los datos que son de entrenamiento y de verificación.

Las salidas que se pueden obtener de RSNNS son:

- Matriz de confusión: Consiste en una matriz de tamaño pxp, donde p es el número de clases, y en la que la entrada (i, j) corresponde a la cantidad de predicciones de la salida i que fueron clasificadas como j. De manera que se tiene una predicción correcta si i = j, e incorrecta si i ≠ j.
- Gráfico de error iterativo: Consiste en el gráfico del error del conjunto de tuplas de entrenamiento y de prueba, como el que se muestra en la Figura 5



- Gráfico de error de regresión: Muestra la predicción como una línea recta, en comparación con la predicción ideal de la recta identidad.

Para extender las características básicas de RSNNS, se plantearon y desarrollaron las siguientes características:

- Porcentaje de aciertos de la configuración en prueba. Esto se logró contando los elementos de la diagonal de la matriz de predicción y el total de entradas de esta matriz. Es importante destacar que esta matriz no necesariamente es cuadrada en todos los casos, pues puede tenerse un conjunto de datos sobre el cual no se realice ninguna predicción de alguna de las clases, al no tenerse ninguna tupla de este subconjunto en el conjunto de verificación.

- Búsqueda extensiva de configuraciones de PMC. Aprovechando que R tiene un lenguaje flexible a la programación de bucles y condicionales, fue posible realizar pruebas de configuraciones para determinar el PMC óptimo dentro de las siguientes restricciones:
  1. Para una capa, entre 1 y 10 neuronas.
  2. Para dos capas, entre 1 y 10 neuronas por capa.
  3. Para tres capas, entre 1 y 5 neuronas por capa.

En esta experiencia, todas las pruebas se real-

izaron con 1000 iteraciones, y un bias de 0.
El siguiente pseudocódigo sintetiza la búsqueda realizada, para el caso de tres capas ocultas:

```
for (i = 1 to 5)
{
  for (i = 1 to 5)
   {
     for(i=1 to 5)
      {
       model(i, j, k)     % Calcula el modelo con
                          % i, j, k neronas en
                          % cada capa.
      predictionmatrix % Genera matriz.
      }
   }
}
```

Como se muestra, se prueban entonces de una a cinco neuronas en las tres capas ocultas, resultando un total de 125 arquitecturas de perceptrón en prueba.

- Tabla y gráfica de aciertos: Una vez implementado el algoritmo de conteo de las entradas de la matriz de predicción y obtenido el porcentaje de aciertos, se procede a tabular el resultado con el número de nodos en cada capa y el porcentaje de aciertos para cada configuración. La figura 6 muestra el porcentaje de aciertos obtenido en todo el conjunto de configuraciones para el caso de tres capas ocultas.

Si bien las pruebas realizadas consistieron solamente en un parámetro analizado, la combinación de R y RSNNS permite extender éstas hacia otras pruebas, lo cual podría agregar una dimensión adicional a la gráfica de la figura 5. Por ejemplo, se podría graficar prueba, porcentaje de aciertos y número de iteraciones, agregando un bucle *for* adicional al código. O bien en lugar de este último modificaciones al sesgo.

La finalidad es poder ilustrar la variación en la efectividad de la clasificación con los diferentes parámetros que se pueden manipular en los perceptrones, a partir de un conjunto de datos creados con este fin, y sobre los que se puede ejercer un control directo. En entornos de educación, pueden plantearse preguntas adicionales, no solamente relacionadas con las características del perceptrón, sino de la codificación de los datos, por ejemplo: Observar variación en los resultados si los píxeles se codifican como -1 y 1 en lugar de 0 y 1, u otros valores como 0 y 100.

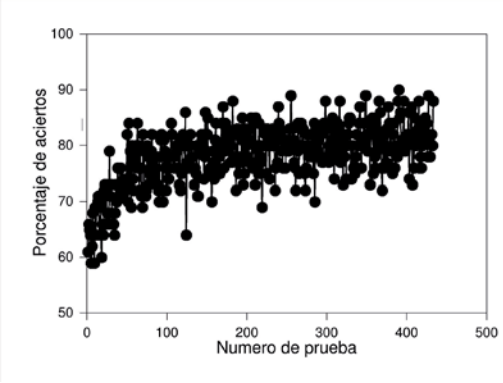


Figura 6: Porcentaje de aciertos para las pruebas realizadas.

El mejor resultado de clasificación, un 90.4% de aciertos, se obtuve con la configuración de dos capas ocultas y 36 nodos en la primera y 37 nodos en la segunda.

IV. PERSPECTIVAS DE DESARROLLO Y APLICACIÓN FUTURA

La metodología de creación de datos y análisis propuesto en R, puede extenderse para su planteamiento y aprovechamiento en el contexto de cursos de educación superior, con extensiones como construir una base de datos y análisis semejante para casos como:

- Clasificación de figuras abiertas y cerradas (figura 7).

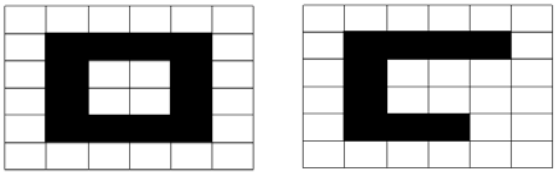


Figura 7: Muestra de imágenes simples de 6X6 píxeles con figura cerrada y abierta.

- Reconocimiento de patrones de color, como celdas conjuntas del mismo color. Esto podría realizarse con códigos distintos para cada color presente, o utilizar, por ejemplo, codificación RGB.
- Extensión a imágenes con mayor cantidad de píxeles, y clasificación de patrones más complejos, como figuras con formas más reconocibles.

Esto, además de las mencionadas posibilidades de

exploración con R sobre todos los parámetros que están presentes en el PMC: cantidad de capas ocultas, cantidad de nodos por capa, bias, iteraciones, función de transferencia y activación.
Esto podría extender propuestas como las realizadas por Gershenson[3], y acercar a temas más avanzados como el reconocimiento de texturas, planteado por Venayagamoorthy[4].

CONCLUSIONES

Se ha mostrado una experiencia de construcción de base de datos y prueba de clasificación con perceptrones multicapa, lo cual ejemplifica las posibilidades que planteamientos semejantes puedan tener en contextos educativos.
Las ventajas de este tipo de experiencias pueden ser:

- La experimentación con la generación de un conjunto de datos y su codificación, con las respectivas observaciones sobre las variaciones en codificaciones y dificultades que se presentan.
- La referencia visual del conjunto de datos.
- La introducción a conceptos de reconocimiento de patrones.
- La exploración exhaustiva de redes neuronales tipo perceptrón.
- La observación de tendencias y la mayor o menor influencia de la cantidad de capas, cantidad de neuronas u otros parámetros en el porcentaje de aciertos.

Dada la simplicidad de los datos y su tratamiento, es factible plantear las extensiones propuestas en cursos con duración trimestral o semestral.

REFERENCIAS

[1] *International Journal of Software Engineering and Its Applications 2010*. Vol. 4, No. 2, April 2010 Use of Artificial Neural Network in Pattern Recognition.

[2] De los Cobos, S., Goddard, J., Gutiérrez, M. y Martínez, A. *Búsqueda y exploración estocástica*. Universidad Autónoma Metropolitana Unidad Iztapalapa. Primera edición 2010.

[3] Venayagamoorthy , G. "Teaching neural networks concepts and their learning techniques." *Proceedings of the 2004 American Society for Engineering Education Midwest Section Conference* .

[4] (Marzo 2012) Gershenson , C. "Artificial Neural Networks for Beginners" . ArXiv, 2003. [Online]. Disponible en http://turing.iimas.unam.mx/~cgg/cogs/doc/FCS-ANN-tutorial.htm

[5] Musoko, V.; Kol, M. y Prochzka, A. "Image classification using competitive neural networks". 12th Scientific Conference MATLAB2004, Humusoft, 2004.