

RED NEURAL PARA RECONOCER MINERALES EN UNA IMAGEN DIGITALIZADA

Mauricio Solar* (msolar@toqui.usach.cl)

Francisco Watkins** (watkins@lauca.usach.cl)

Patricio Perez*** (pperez@lauca.usach.cl)

Universidad de Santiago de Chile

*Departamento de Ingeniería Informática, Facultad de Ingeniería

**Departamento de Ingeniería Eléctrica, Facultad de Ingeniería

***Departamento de Física, Facultad de Ciencias

Avda. Ecuador 3659

Fono: 56 2 778-1208

Fax: 56 2 776-3511

Resumen

Este trabajo presenta el diseño e implementación de una Red Neuronal para reconocer imágenes digitalizadas a partir de fotografías de minerales. A la entrada de la red se presenta un histograma comprimido de una sección de la imagen seleccionada por el usuario que se desea reconocer, la cual es procesada por la red diseñada, identificando uno de los seis minerales enseñados.

1. Introducción.

El procedimiento actual para el reconocimiento de minerales se realiza obteniendo información de los distintos minerales que componen una roca. Para realizar esto se preparan muestras pulidas de un cuarto de pulgada de grosor. La pulimentación de la muestra permite la obtención de superficies lisas para su análisis al microscopio. La muestra pulida se denomina *briqueta*. La briqueta se coloca sobre una cuadrícula, como una manera de discretizar la imagen al microscopio. De esta manera es posible cuantificar la cantidad de puntos que abarca u ocupa cada *grano* de mineral existentes en la muestra. Un grano es una sección de la muestra que contiene un determinado mineral, y es identificado por el experto recordando las características visuales de cada mineral.

El proceso es extremadamente lento y proclive al error, dada la excesiva dependencia de la atención del experto a cargo de la detección y contabilización de puntos.

El problema planteado se divide en dos áreas relevantes: a) Reconocimiento de mineral; y b) Contabilización de puntos en el área asociada a cada grano, con el objetivo de determinar su presencia porcentual en la muestra.

Para resolver el problema de reconocimiento de minerales se procedió a digitalizar fotografías tomadas directamente desde el laboratorio donde se realiza el análisis de las muestras. A través de un *scanner*, se generó un conjunto de imágenes digitalizadas en formato TIFF, a las cuales se les realizó un pre-procesamiento para generar un archivo con patrones de entrada para el entrenamiento de una Red Neuronal (RN) diseñada para este propósito.

Para el diseño de la arquitectura de RN [1] que identifique los patrones asociados a cada mineral, se utilizó el software *Professional NeuralWorks* bajo ambiente *SunOS*. Para el proceso de

aprendizaje se utilizó el algoritmo de *backpropagation* [3] para entrenar la RN con los patrones de entrada digitalizados previamente. Luego en la fase de operación, para el Reconocimiento de Minerales, se construyó una interfaz amistosa bajo ambiente *Windows* que utiliza la RN previamente entrenada con los patrones disponibles, permitiendo el reconocimiento automático.

El producto final es un sistema capaz de reconocer los minerales a través de la inspección sobre un monitor de computador de la imagen digitalizada de una briqueta. La entrada consiste en la selección de un área de la briqueta por parte del usuario y la salida es la identificación del mineral.

Para resolver el segundo problema de obtener la presencia porcentual de cada grano de mineral en la muestra, se implementó una opción, que permite inspeccionar exhaustivamente toda la imagen para determinar la cantidad total del mineral dentro de la imagen (en tanto por ciento).

2. Diseño de la Red Neuronal.

Para entrenar la RN se dispone de 36 fotografías a color, cada una de las cuales cubría una fracción de diferentes partes de una briqueta. En todas las fotografías se observaron regiones delimitadas que correspondían a granos de diferentes minerales (muchos de ellos componentes del cobre). Basados en la información dada por el experto, se obtuvo patrones de los siguientes seis minerales utilizados:

1	Calcopirita
2	Calcosina
3	Covelina
4	Bornita
5	Pirita
6	Energita

Se procedió a digitalizar las 36 fotografías usando un scanner a color que permite almacenarlas en formato TIFF. Se desplazó una ventana de un tamaño de 10×10 pixels sobre las imágenes, seleccionando aquellas regiones que corresponden a cualquiera de los 6 minerales antes mencionados. En condiciones constantes de captura de datos, es posible afirmar que cada uno de los minerales estudiados en una muestra digitalizada poseen una textura similar para distintas muestras. La arquitectura de RN debe ser capaz de aprender estas texturas, mediante la clasificación de sus histogramas característicos. Para esto, se descompuso esta información en 3 filtros (rojo, verde y azul) y se descubrió que comprimiendo el histograma en 23 intensidades es suficiente para discriminar los minerales.

Como entrada a la RN se tiene entonces, un histograma, que es una función de frecuencia [2]. Es decir, el histograma de una imagen digitalizada es la contabilización del número de pixel clasificados por el nivel de color existente en ella.

En la construcción de la RN feedforward, se definió como entrada una capa de 23 neuronas y una capa de salida de 8 neuronas. La neurona activada en la capa de salida identifican los minerales. Se encontró necesario incorporar 2 neuronas adicionales en la capa de salida, dado que se descubrió que existen 2 patrones diferentes para identificar la covelina, y la otra neurona para activarla cuando se está en presencia del fondo de la briqueta (no hay presencia de mineral). El hecho de identificar el fondo es importante para determinar la presencia relativa de cada mineral en la muestra, cuando esta se recorre completamente.

Los datos del histograma se mapearon al intervalo $(0,1)$, el cual permite trabajar en la región de mayor sensibilidad de la función de transferencia $f(z)$, llamada también función logística, en la cual el estado de la neurona de salida j en la capa s está dado por:

$$x_j^{[s]} = f\left(\sum_i (w_{ji}^{[s]} x_i^{[s-1]})\right) \quad (1)$$

donde

$$f(z) = (1 + e^{-z})^{-1} \quad (2)$$

$w_{ji}^{[s]}$ representa el peso de la conexión desde la neurona i de la capa $s-1$ con la neurona j en la capa s . En la fase de entrenamiento, estos pesos se inicializaron con valores aleatorios, y posteriormente se ajustaron en forma iterativa de acuerdo a:

$$\Delta w_{ji}^{[s]} = \varepsilon^{[s]} \varepsilon_j^{[s]} x_i^{[s-1]} + \alpha^{[s]} \Delta w_{ji}^{[s]}(t-1) \quad (3)$$

donde $\varepsilon_j^{[s]}$ es una medida del error local en la neurona j de la capa s , $\varepsilon^{[s]}$ es el coeficiente de aprendizaje de la capa s y $\alpha^{[s]}$ es conocido como el coeficiente de momentum.

Al realizar las investigaciones sobre el efecto de variar el número de neuronas en la capa escondida, se consideró que los mejores resultados se obtuvieron con 13 neuronas en esa capa, con los siguientes parámetros:

- $\varepsilon^{[1]} = 0.4$
- $\varepsilon^{[2]} = 0.4$
- $\alpha^{[1]} = 0.8$
- $\alpha^{[2]} = 0.8$
- Número de ciclos de entrenamiento: 60.000
- Se decrementó el coeficiente de aprendizaje en un 10% cada 10.000 ciclos.

La RN resultante y entrenada generó sus pesos y umbrales por capas en diferentes archivos que contienen: los pesos de la capa de entrada hacia la capa oculta; los pesos de la capa oculta hacia la

capa de salida; los umbrales de la capa de entrada hacia la capa oculta; los umbrales de la capa oculta hacia la capa de salida; información relativa a normalización del histograma hacia la capa de entrada de la red; e información relativa a normalización del histograma hacia la capa de entrada de la red. Esta información define totalmente la RN entrenada, y se transportó para su uso bajo ambiente *Windows* en la fase de reconocimiento.

2.1. Reconocimiento de Mineral por Rectángulo

En el proceso de reconocimiento se implementó la modalidad de selección de área. El usuario define un rectángulo de tamaño variable de la imagen con el *mouse*, y luego comienza el siguiente procedimiento (cuyos algoritmos se muestran en Anexo A):

- a. Genera histograma de área seleccionada almacenándola con 256 valores;
- b. Transforma *paleta* de 256 valores a un intervalo de 22 y colocando el valor 23 como 0, con el fin de generar el vector de entrada a la RN;
- c. Normaliza tamaño de área seleccionada a tamaño estándar con que se entrenó la RN (10x10);
- d. El vector debe normalizarse nuevamente debido a que debe estar en un rango (0,1);
- e. El vector de entrada a la RN se procesa por intermedio del algoritmo de propagación;
- f. Siete (7) neuronas de salida representan un mineral (covelina usa 2 neuronas). Si hay alguna neurona excitada sobre el valor 0.6 (determinado por el experto), entonces es altamente factible que el área seleccionada sea dicho mineral. Si ninguna neurona alcanza el umbral, la muestra no es suficientemente clara, y se excita la octava neurona que identifica el Fondo. La relación es:

oi[0]	Calcopirita
oi[1]	Calcosina
oi[2]	Covelina
oi[3]	
oi[4]	Bornita
oi[5]	Pirita
oi[6]	Energita
oi[7]	Fondo

2.2. Presencia Porcentual de cada Mineral en la Muestra

Este proceso realiza un recorrido exhaustivo de la imagen, barriéndola en su totalidad por medio de dos ciclos en el cual contempla el tamaño del área seleccionada por el experto (para este caso fue 5x5). El resultado es la cantidad de veces que reconoció el mineral *i*-ésimo. Luego los resultados en relación a estos valores determinan la relación porcentual de cada mineral en la muestra.

3. Interfaz Desarrollada

La figura 1 muestra la interfaz desarrollada, en donde existe la opción de operar de dos maneras con una imagen cargada: Reconocimiento y Porcentajes.

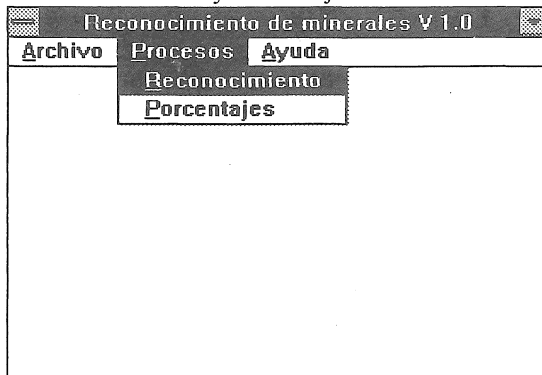


Fig. 1: Interfaz del sistema.

3.1. Modalidad Reconocimiento

Para ejecutar esta opción se debe marcar previamente un cuadrado en el área de trabajo. Una vez elegida un área en la imagen del mineral que se desea reconocer, se elige ésta opción para que el sistema indique de que mineral se trata (Fig. 2). La respuesta se indica en una ventana de diálogo, en la que aparece el nombre del mineral identificado, junto con el grado de certeza del reconocimiento (Fig. 3). Puede darse el caso que se entregue como resultado más de un mineral, caso en el que se agrega la información del grado de confiabilidad de cada uno de ellos. También

puede darse el caso la muestra entregada no sea ninguno de los minerales que es capaz de reconocer esta aplicación, caso en el cual entregará un mensaje de mineral no reconocido.

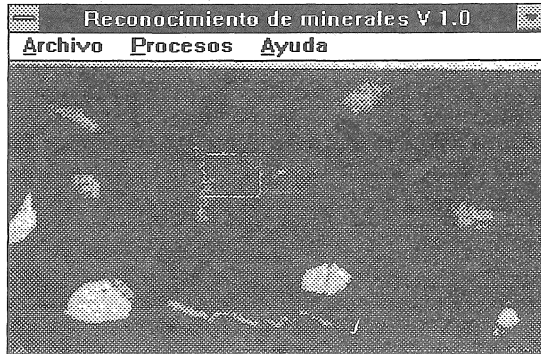


Fig. 2: Selección de un área a examinar.

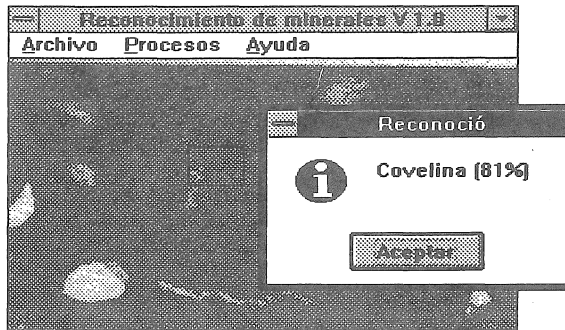


Fig 3: Reconocimiento de un mineral.

3.2 Modalidad Porcentajes

Realiza un proceso de reconocimiento en toda la imagen, entregando como resultado el porcentaje total que ocupa cada mineral en la muestra. (Fig. 4).

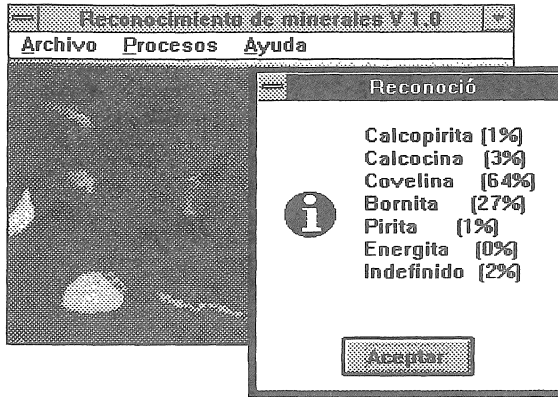


Fig. 4: Cálculo porcentual de cada mineral en la muestra.

4. Conclusiones y Consideraciones

Dado que la aplicación opera con imágenes digitalizadas, se debe atender dos puntos importantes:

- Las imágenes con las que se entrenó la RN son de formato TIFF paletizado, por lo que cualquier otro tipo de imagen será incomprensible para la aplicación.
- Dado que las imágenes están en 256 colores, se recomienda utilizar un equipo que soporte *Windows* con esta cantidad de colores o mas. A pesar que la RN opera correctamente al trabajar con 16 colores, se degrada la imagen, haciéndola menos visible, pudiendo ésto acarrear problemas al seleccionar las áreas de reconocimiento.

Los resultados obtenidos con el uso de RN en el reconocimiento de minerales son:

- En la fase de entrenamiento se obtuvo un reconocimiento correcto de 97% identificado.
- En la fase de prueba con imágenes nuevas la identificación correcta fue de 91%.
- Se usaron 160 patrones de entrenamiento y se probaron 120 patrones.

Actualmente, se está en la fase de incorporar una nueva capa escondida, lo que permitirá mejorar eficiente y significativamente los resultados.

Se observó que varios de los problemas de mala clasificación se debe al hecho que algunas regiones de imágenes presentan superposición de 2 o más minerales, lo que hace que los histogramas sean poco "claros". Una ventana de menor tamaño ayuda a resolver este problema.

Otra fuente de error corresponde al proceso de revelado, película utilizada, papel de la fotografía, y proceso de digitalización, entre otros. Estas condiciones de procesamiento tienen gran relevancia en los resultados. Para eliminar las posibles fuentes de error, se está trabajando sobre datos obtenidos directamente desde una cámara de video instalada en el microscopio, desde donde se obtuvieron las fotografías de las muestras de las briquetas de los minerales.

De los resultados obtenidos, y de acuerdo a la opinión de usuarios en el Centro de Investigaciones Minero Metalúrgico (CIMM), consideramos que el tipo de RN descrita en este trabajo, permite automatizar satisfactoriamente el proceso de reconocimiento de minerales.

Reconocimientos

Agradecemos la colaboración C. Aguirre, G. Salini y P. Verdugo financiados por la Comisión Nacional de Ciencia y Tecnología (CONICYT) y el Centro de Investigaciones Minero Metalúrgico (CIMM) a través del proyecto FONDEF IN-03.

Referencias

- [1] Gallant, S., *Neural Networks Learning and Expert Systems*, MIT Press, Cambridge, Mass., 93.
- [2] Gonzalez, R. and Woods, R., *Digital Image Processing*, Addison-Wesley Publ., 1992.
- [3] Rumelhart, D.; McClelland J. and PDP Research Group, *Parallel and Distributed Processing*; Vol.1, MIT Press, Cambridge, Mass., 1986.

ANEXO

Algoritmo a:

```

memset(&r_histo[0],0, 22*sizeof(int));
memset(&g_histo[0],0, 22*sizeof(int));
memset(&b_histo[0],0, 22*sizeof(int));
memset(&rgb_histo[0],0,67*sizeof(int));
for(x=0;x<256;x++) paleta[x]=0;
for(i=PosYs;i<PosYi;i++)
    for(j=PosXs;j<PosXi;j++) {
        paleta[im[i][j]]++;
        r_histo[aprox(trgb[a][0]/3)]++;
        g_histo[aprox(trgb[a][1]/3)]++;
        b_histo[aprox(trgb[a][2]/3)]++;
    }

```

Algoritmo b:

```

/* Forma patrón de 66 componentes (3x22), uniendo histogramas RGB */
j=0;
for(k=0;k<=21;k++) rgb_histo[j++]=r_histo[k];
for(k=0;k<=21;k++) rgb_histo[j++]=g_histo[k];
for(k=0;k<=21;k++) rgb_histo[j++]=b_histo[k];
/* Generación de Intervalos */
intervalo=0;
sum=0;
j=0;
for(k=0 ;k<=66 ;k++)
{ sum=sum+rgb_histo[k];
  j++;
  if(k==66) j=A_INTER;
  if(j==A_INTER)
  { j=0;
    integ[intervalo++]=sum;
    sum=0;
  }
}
integ[22]=0;

```

Algoritmo c:

```

std = 15*15;
largox=PosXs-PosXi;
largoy=PosYs-PosYi;
muestra = largox*largoy;
muestra=(muestra>0)?muestra:-muestra;
coef=muestra/std;
for (x=0;x<23;x++) {

```

```

    resto=integ[x]/coef-(int) integ[x]/coef;
    ti[x]=(resto>0.5)?(integ[x]/coef)+1:(integ[x]/coef);
}

```

Algoritmo d:

```

for(x=0;x<23;x++)
if (max[x] > 0) tic[x]=((rri-ri)*ti[x]+max[x]* ri- min[x]*rri)/(max[x]-min[x]);
else tic[x]=ti[x];

```

Algoritmo e:

```

for(x=0;x<11;x++) { /* Propagación Input to Hidden */
    zi[x]= 0;
    for(y=0;y<23;y++)    zi[x] += pih[x][y] * tic[y];
    zi[x] += uih[x];
    hi[x] = 1 / (1 + exp(-zi[x]));
}
for(x=0;x<6;x++) { /* Propagación Hidden to Output */
    zo[x]= 0;
    for(y=0;y<11;y++)    zo[x] += pho[x][y]*hi[y];
    zo[x] += uho[x];
    oi[x] = 1/(1+exp(-zo[x]));
}

```

Algoritmo Porcentaje:

```

for(i=0;i<7;i++) porcentaje[i]=0;
for(PosXs=0;PosXs<320;PosXs+=INCRE) {
    PosXi=PosXs+INCRE;
    if (PosXi>=320) PosXi=319;
    for(PosYs=0;PosYs<200;PosYs+=INCRE)
    {
        PosYi=PosYs+INCRE;
        if (PosYi>=200) PosYi=199;
        calcula_mineral();
        mayor=oi[0];
        mayorind=0;
        for(i=1;i<6;i++)    if (oi[i]>mayor) {
                                mayor=oi[i];
                                mayorind=i;
                            }
    }
    if (mayor<.6) mayorind=6;
    porcentaje[mayorind]++;
    total++;
}

```