MACHINE LEARNING
NEURAL NETWORKS
BREAST-CANCER-WISCONSIN

Andres Bobadilla – Kevin Julio
William Caicedo

ABSTRACT
en el siguiente documento se encuentra la implementaciòn de una red neuronal multicapa para la clasificación de tejidos mamarios entre malignos y benignos con los datos dados. teniendo en cuenta que nuestro clasificador la clase benigno toma el valor de 0 y la clase maligno toma el valor de 1. se hace el entrenamiento con la mitad de los datos, la validación con un un cuarto de ellos al igual que los datos que se usaron para la prueba final, el objetivo fue predecir si el tejido es maligno o no y a partir de esto tomar las decisiones correspondientes.

En el desarrollo de nuestro algoritmo, podemos afirmar con un 97% de veracidad que nuestro modelo predice y clasifica correctamente las muestras tomadas biopsias del tejido mamario.

Keywords: Machine Learning, redes neuronales, Perceptron

INTRODUCCIÒN
El objetivo de este trabajo, es mostrar el uso de Redes Neuronales, para predecir, mediante un grupo de variables (32 en total, de donde solo se trabajan con 31, puesto que la primera columna de datos es de identificación y no es relevante para el proceso de entrenamiento), si las muestras provenientes de biopsias de tejido mamario son malignas o benignas Para esto, se entrena la máquina con un conjunto de datos de distintos tejidos con el fin que la máquina al usar los algoritmos pueda predecir, si este tejido es maligno o benigno.
Este entrenamiento es de tipo supervisado, ya que se conocen los resultados de cada tejido mamario dado, además este problema es de clasificación,ya que la respuesta de estos son uno para los tejidos malignos y cero para los benignos.

1) DATOS QUE EL MODELO
Los datos con los que se trabajan en este clasificador, fueron los proveidos por el profesor que se encuentran en el siguiente enlace:
https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.data

donde tenemos las variables

Variables

1. ID number
2. Diagnóstico (M = maligno, B = benigna)
3. Diez características con valores reales se calculan para cada núcleo celular:
    a. radio (media de las distancias de centro a puntos en el perímetro)
    b. textura (desviación estándar de los valores de escala de grises)
    c. perímetro
    d. Área
    e. la suavidad (variación local en longitudes de radio)
    f. compacidad (perímetro ^ 2 / área - 1.0)
    g. concavidad (severidad de las porciones cóncavas del contorno) puntos
    h. cóncavas (número de porciones cóncavas del contorno)
    i. la simetría
    j. la dimensión fractal ("aproximación costa" - 1)

2) DATOS

Para llevar a cabo el entrenamiento del algoritmo, se deben seguir ciertos pasos para evitar tener errores, como cargar los datos del archivo de excel, y dividir los datos en tres grupos, un grupo de entrenamiento, un grupo de validación y un grupo de prueba.

Se hace uso de la librería "xldr" para leer directamente del archivo de excel que contiene los datos, luego se procede a la separación de nuestros datos en este caso para el entrenamiento usaremos la mitad de los datos (50%), en la validación usaremos el (25%) y por ultimo en la prueba usaremos el (25%) restante. Esto nos permitirá una carga limpia de los datos entendible a la hora de codificar y achica el margen de error de cometer un error de capa 1000.

3) ALGORITMO

De la librería pybrain2 implementamos algoritmo para crear la red neuronal, luego dividimos nuestros datos en 3 datasets: dataset de entrenamiento el cual tuvo la mitad del total de los datos (50%), el 50% restante, se dividió entre dataset de validación y dataset prueba, con 25% cada uno. Posteriormente se entrenó la **red** con el algoritmo con backpropagation y para tener más eficiencia en el entrenamiento de decidió entrenar el algoritmo con 100 iteraciones llamadas **Epoch.**

4) INTERPRETACIÓN DE RESULTADOS

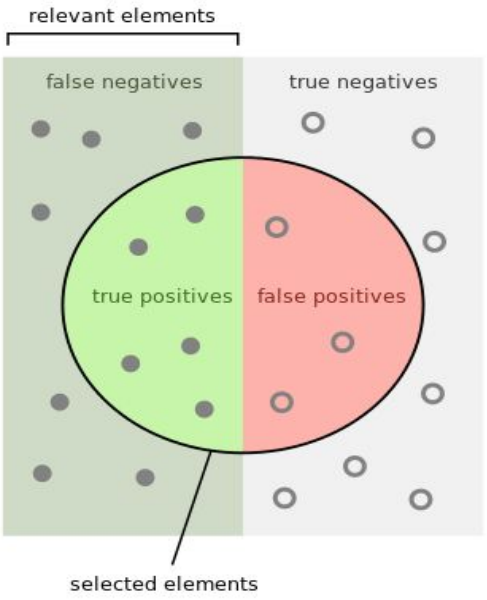
Se deben tener en cuenta los siguientes términos a la hora de realizar las pruebas.

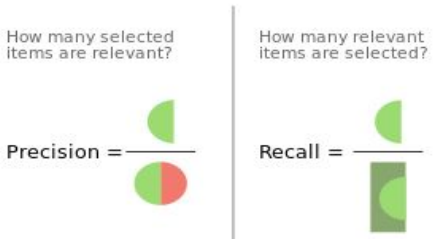
Los elementos están clasificados en dos grupos, los positivos y los negativos, pero cada uno de ellos tiene una parte falsa, es decir, un falso positivo y un falso negativo.

Se evalúan elementos de los resultados de las predicciones y se lleva a cabo la clasificación de estos.

La precisión está dada por los elementos positivos sobre el número total de elementos de la predicción.

El recall, está dato por el número de elementos positivos, sobre el número total de elementos relevantes





$$\text{Precision} = \frac{tp}{tp + fp}$$

Donde:

tp: es el número de elementos positivos verdaderos

fp: es el número de elementos que son falsos positivos

$$\text{Recall} = \frac{tp}{tp + fn}$$

Donde:

fn: es el número de elementos falsos negativos

y el accuracy (exactitud) está dada por:

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

4.1) AJUSTES Y RESULTADOS :

Luego de realizar cambios e iterar un número de veces suficientes, obtuvimos resultados mucho mejores. para mejor comportamiento de la red modificamos parámetros, pero no fue suficiente con solo modificar parámetros, fue necesario normalizar los datos para estandarizar las columnas y fuera mucho más fácil para nuestro modelo encontrar el mínimo global.

Obtuvimos un resultado 97% de precisión en la clasificación de las muestras, para esto ajustamos la red con los siguientes parámetros:

- Neuronas de la capa oculta : 400
- Momentum : 0.1
- weight decay : 0.0001
- Learningrate : 0.1

los resultados fueron:

MATRIZ DE CONFUSIÓN	
34	1

2	105

1. Falsos positivos : 1
2. Falsos Negativos
3. Número de aciertos en predicción : 139
4. Número de fallos en prediccion : 3

	precisión	recall	f1- score	support
0.0	0.99	0.98	0.99	107
1.0	0.94	0.97	0.96	35
avg/ total	0.98	0.98	0.98	142

5) MUESTRA DE LA EJECUCIÓN:

```
Matriz de Confusion Conjunto de Prueba

[[ 34   1]
 [  2 105]]


falsos positivo 1
falsos Negativos 2
Numero de aciertos en prediccion:  139
Numero de fallos en prediccion:  3
porcentaje de prediccion acertada(Accuracy):  0.978873239437

             precision    recall  f1-score   support

        0.0       0.99      0.98      0.99       107
        1.0       0.94      0.97      0.96        35

avg / total       0.98      0.98      0.98       142
```

6) MUESTRA DE LA EJECUCIÓN DE LA RED NEURONAL

```
epoch:   71   error entrenamiento:  1.05%   error prueba: 13.38% Total error: 0.0104010651372
epoch:   72   error entrenamiento:  1.75%   error prueba: 14.08% Total error: 0.00635870532651
epoch:   73   error entrenamiento:  2.81%   error prueba:  4.93% Total error: 0.0118581293062
epoch:   74   error entrenamiento:  1.75%   error prueba: 10.56% Total error: 0.0136139049727
epoch:   75   error entrenamiento:  2.81%   error prueba:  5.63% Total error: 0.0129874630084
epoch:   76   error entrenamiento:  2.11%   error prueba:  7.75% Total error: 0.0181328916026
epoch:   77   error entrenamiento:  2.81%   error prueba:  7.04% Total error: 0.0146967194304
epoch:   78   error entrenamiento:  1.75%   error prueba: 19.01% Total error: 0.0131855903296
epoch:   79   error entrenamiento:  2.11%   error prueba:  9.15% Total error: 0.0151033959758
epoch:   80   error entrenamiento:  3.16%   error prueba: 24.65% Total error: 0.00572299242013
epoch:   81   error entrenamiento:  2.11%   error prueba: 14.79% Total error: 0.0100119069986
epoch:   82   error entrenamiento:  1.40%   error prueba:  9.15% Total error: 0.00858734035788
epoch:   83   error entrenamiento:  1.05%   error prueba:  9.86% Total error: 0.00728326482596
epoch:   84   error entrenamiento:  5.26%   error prueba:  2.11% Total error: 0.00966858166684
epoch:   85   error entrenamiento:  2.11%   error prueba: 16.90% Total error: 0.0125958194558
epoch:   86   error entrenamiento:  1.40%   error prueba:  4.93% Total error: 0.0122300980362
epoch:   87   error entrenamiento:  1.05%   error prueba:  9.15% Total error: 0.010780981739
epoch:   88   error entrenamiento:  1.40%   error prueba:  7.04% Total error: 0.00958786679435
epoch:   89   error entrenamiento:  4.56%   error prueba: 23.24% Total error: 0.0118129749664
epoch:   90   error entrenamiento:  2.81%   error prueba:  2.11% Total error: 0.00996158529336
epoch:   91   error entrenamiento:  2.46%   error prueba:  2.11% Total error: 0.00921491753328
epoch:   92   error entrenamiento:  5.96%   error prueba: 28.17% Total error: 0.0117289214677
epoch:   93   error entrenamiento:  1.40%   error prueba:  7.04% Total error: 0.0141986728264
epoch:   94   error entrenamiento:  1.40%   error prueba:  5.63% Total error: 0.0110977643601
epoch:   95   error entrenamiento:  3.16%   error prueba:  2.11% Total error: 0.00935385041816
epoch:   96   error entrenamiento:  2.11%   error prueba:  4.23% Total error: 0.010262130779
epoch:   97   error entrenamiento:  1.40%   error prueba:  7.75% Total error: 0.0095974182384
epoch:   98   error entrenamiento:  3.86%   error prueba:  2.11% Total error: 0.0104781264026
epoch:   99   error entrenamiento:  1.05%   error prueba:  7.75% Total error: 0.0128476472697
epoch:  100   error entrenamiento:  3.51%   error prueba:  2.11% Salidas Entrenamiento : [1, 1,
```

7) CONCLUSIÓN

Después de implementar, cargar los datos, entrenar el algoritmo, realizar la validación y posteriormente prueba en el algoritmo de Machine Learning y evaluar sus predicciones tenemos una predicción acertada del 97% , lo que indica que se llevó a cabo de la mejor manera la construcción del clasificador con la red neuronal y con esto podemos decir, que nuestro clasificador es la mejor opción para determinar si las muestras tomadas de las biopsias del tejido mamario para establecer la probabilidad de que el paciente tenga cáncer, son benignas o malignas

REFERENCES

Gracias a las indicaciones del profesor y a sus consejos se implementò el algoritmo de redes neuronales multicapa de una manera correcta.