Application for breast cancer classification using Computational Intelligence techniques

Manoel Eric N. Oliveira[1], **Felipe Barros Muniz**[2], **Ruann C. Farrapo**[3]

[1]Department of Electrical Engineering – Federal University of Ceará
Sobral,CE.

[2,3]Department of Computer Engineering – Federal University of Ceará
Sobral,CE.

{manoeleric59,felipemuniz,ruann.campos_01}@alu.ufc.br,

Abstract. *In this work, a comparative study was carried out between two classification methods: The Multi layer Perceptron Artificial Neural Network (MLP ANN) and the method of classification of the Nearest Neighbors, used in the classification of the diagnosis of breast cancer. The data used in this work were taken from the UCI Machine Learning Repository and contains numerical data extracted from mammography images.In addition, the results were evaluated based on the cross-validation strategy.*

1. Introduction

A form of early detection of breast cancer is conventional mammography, which consists of the analysis of images by radiologists and able to identify mammography signs. Previous studies show that exhaustive analysis of mammography images in the same period of work can be passive of errors. This fact implies that the observer may end up making mistakes for showing interest in certain areas, making other areas go unnoticed [Dellani and Borges]. The Brazilian mortality peripheral has undergone an intense change, changing from infectious-parasitic diseases to chronic-degenerative diseases, such as cancer.[Haddad and SILVA 2001]

Thus, considering that breast cancer is the cancer that kills the most among women, diagnostic aid tools have been developed to assist radiologists in detecting suspected microcalcifications and nodular masses [Abdou et al. 2020]. Thus, the aid to computer diagnosis is a tool for health professionals evidence probabilistic estimates the occurrence of breast cancer in certain cases.

Thus, this work proposes to make a comparative analysis between two classification methods that use Computational Intelligence techniques. Using the UCI dataset for breast cancer diagnosis, an MLP (Multilayer Perceptron) neural network and a KNN (K-Nearest Neighbors) classifier were implemented, which were compared with each other in order to provide one more computing tool applied to health.

2. Methodology

This work deals with experimental research, which aims to compare two methods for classifying breast cancer as malignant or benign.The method used consists of providing the data for a neural network of the MLP type and a KNN classifier, changing the settings

such as number of neurons and neighbors and analyzing the results. The methodology of this work follows the flowchart shown in figure 2 below, where from a database, pre-processing and algorithms, it obtains a classification.

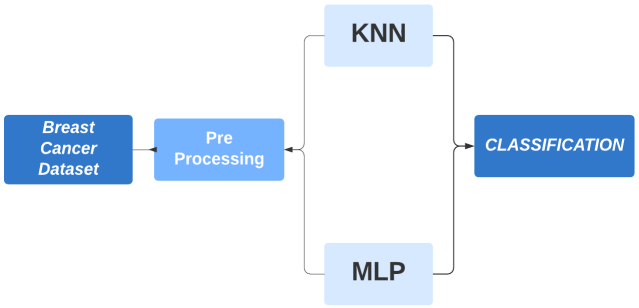

Figure 1. Methodology steps

2.1. Data Base

A database available for free at the UCI Machine Learning Repository [Dua and Graff 2017] was used, which contains resources calculated from a scanned image of a mammogram. Such data describe characteristics of the cell nuclei present in the image how to: id number, diagnosis, radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry and fractal dimension. The database has 569 instances and 32 attributes. The attributes are all numeric, except for the diagnostic output.The figure below shows the main characteristics of the database used in this work.

Data Set Characteristics:	Multivariate	Number of Instances:	569	Area:	Life
Attribute Characteristics:	Real	Number of Attributes:	32	Date Donated	1995-11-01
Associated Tasks:	Classification	Missing Values?	No	Number of Web Hits:	1301326

Figure 2. Breast Cancer Wisconsin (Diagnostic) Data Set (adapting of [Dua and Graff 2017])

Before training, strategies had to be made so that problems such as non-converging and missing data would not cause an error in the process. For that, the data had to be normalized. Data normalization and data validation will be explained in the validation section later.

2.2. MLP training

Multilayer perceptron ANNs (MLP) are neurons connected by connections synaptic cells that are divided into input neurons, which receive stimuli from the middle into internal neurons, responsible for making the neurons of the layers of entrance and exit; and in output neurons, which communicate with the outside [Haykin 2001]. An input signal xi at the input of a neuron I is multiplied by the synaptic weight wij and, after calculation, the value is sent to the input of neuron J. Each neuron J performs the sum of all signals applied to its input, according to equation (1), and apply to an activation function.

$$u = \sum w_{ij} x_i \tag{1}$$

The activation function used in this algorithm was the relu function, which has the following representation in equation (2):

$$f(x) = max(0, x) \tag{2}$$

The output y_j is equal to the value of the activation function given by equation (3):

$$y_j = f(u) \tag{3}$$

The algorithm used for learning in MLP is called descent from stochastic gradient. The stochastic gradient drop (SGD) updates the parameters using the gradient negative of the loss function in relation to a parameter that needs to be changed. Thus, as the gradient points to where the function is increasing, one seeks to walk in the opposite direction to maximize the solution [Zeybek et al. 2006]. The following equation (4) demonstrates how the SGD does to adjust the weights, minimizing the error.

$$E = \sum E^p \Leftrightarrow \sum_P (d_r - d_p)^2 \tag{4}$$

Where, E^P represents the error, d_r is the desired output and d_p is the output obtained. The partial derivative of the error is calculated. Subsequently, the descending gradient method is used to update the weights according to equation (5):

$$w_{ij}(t+1) = w_{ij}(t) + \frac{dE}{dw_{ij}} \tag{5}$$

In addition to the conventional error propagation method shown above, the momentum insertion technique was used. The term momentum is a device that aims to consider how much the synaptic weights have been changed between two consecutive interactions. such proposal aims that the algorithm is not stuck in local minimums thereby improving network efficiency. Equation (6) can be obtained by modifying equation (5) with the addition of the α variable, with a value between 0 and 1.

$$w_{ij}(t+1) = w_{ij}(t) + \alpha[w_{ij}(t) - w_{ij}(t-1)] + \frac{dE}{dw_{ij}} \tag{6}$$

In addition to including the term momentum, the algorithm used in this work was implemented with the strategy that if the error is not minimized between 3 consecutive iterations in 0.0001, the algorithm stop.

2.3. KNN training

The K-nearest neighbor (KNN) classification method has been an algorithm widely used in classification problems. This fact is noticeable, for example, when we look at the works of [Athitsos and Sclaroff 2005] and [Athitsos et al. 2005]. The algorithm proceeds as follows: Given a query vector x_o and a set of N labeled instances $\{x_i, y_i\}_1^N$, he classifier's function is to predict the class label of $x0$ in the predefined P classes. The K-nearest neighbor (KNN) method tries to find the nearest neighbor to $x0$ and uses a kind of majority vote to determine the class label of $x0$.The most common and used form in KNN is to apply Euclidean distances as the distance metric as shown in the following equation:

$$D_{pq} = \sqrt{\sum_{1}^{n}(p_i - q_i)^2} \tag{7}$$

As in the case of the Breast Cancer database, the data can be distributed in a Linear way and the database does not have missing and scattered data, it was observed in practice that the Euclidean distance presented a good way to solve the problem described.

Summing up, KNN is a non-parametric algorithm where the structure of the model will be determined by the database. The algorithm basically works in 3 steps: find the distance, find the nearest neighbors and vote for the markers.

3. Validation

For the validation of the algorithms, a cross-validation strategy was used. However, before showing how the validation was done, we need to talk about how the data was normalized. Data normalization occurs when in a large database we have many attributes of numerical values with many significant figures. Such occurrences can end up affecting the processing of the algorithms because they are repetition structures that will work in various periods of training, validation and testing.

The standardization of a data set is a common requirement for many machine learning estimators: they can behave badly if individual resources do not look more or less like normally distributed standard data. So, to make it easier, we can use the standardization strategy using StandardScaler. This method removes the average and the dimensioning of the unit, this results in data with an average equal to 0 and deviation equal to 1. The formulas for this process, follow below where they show the formula for standardization, mean and standard deviation:

$$z = \frac{x - \mu}{\sigma} \tag{8}$$

$$\mu = \frac{1}{N}\sum_{i=1}^{N}(x_i) \tag{9}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{10}$$

The main difficulty in using classification algorithms is how to identify the best stopping point for training, as the training error tends to decrease according to the number of times of the algorithms used. [Haykin 2001]. For this, seeking a better generalization of the classification algorithms, we use the cross validation strategy, where we partition the original database in sub intervals so that the algorithm is submitted to different data from the one previously trained, thus improving its generalization capacity.[Guimarães et al. 2008] A methodology for operating cross-validation can be shown in the following figure:

The cross validation used divided the database into 5 splits, 75% being test and 25% for validation. In addition, the cross-validation was within a repetition structure

Figure 3. Cross Validation architecture. [Silva et al. 2010]

that was repeated 10 times for each MLP topology and KNN classifier. Thus, at each compilation of the algorithm, we had a list with 50 results for network accuracy. Accuracy can be calculated using equation (11) below, where VP corresponds to the positive truths, VN the true negatives, FP the false positives and FN the false negatives.

$$\frac{VP + VN}{VP + VN + FP + FN} \tag{11}$$

4. RESULTS AND DISCUSSION

4.1. MLP Results

The training of the Neural MLP Network was divided into 3 stages: dividing the pre-processed database, training the network with the training data and validating the network with the validation data. Subsequently, these steps are repeated with the change of the data selected for training and validation as stated in the cross-validation strategy.

Empirically, it was realized that the best organization would be the one with only an intermediate layer of neurons. Thus, successive tests were made for the amounts of neurons, where we varied from 1 to 100. Thus, we realized that the best results were the organization that used an intermediate layer and 36,38,40 and 50 neurons. The following is a table with the evaluation metrics and a graph with the validation results obtained for 40 neurons, respectively.

Table 1. Data of MLP training

Number of Neurons	Mean Accuracy	Bigger Accuracy	Less Accuracy	Stan-dard Detour
36	0.9727	1.0	0.9385	0.0163
38	0.9722	1.0	0.9385	0.0158
40	0.9717	1.0	0.9298	0.0157
50	0.9724	1.0	0.9210	0.0160

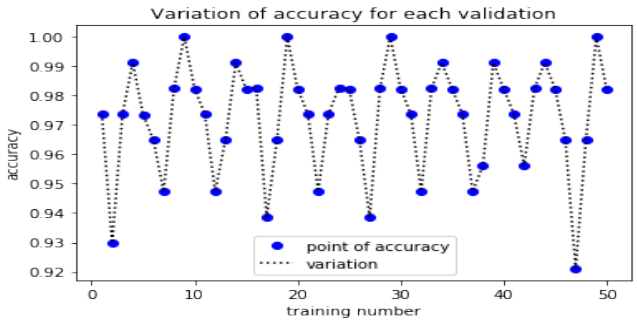

Figure 4. Graphic of MLP validation for 40 neurons

The figure below shows a general histogram of the accuracy of each validation for 40 neurons.

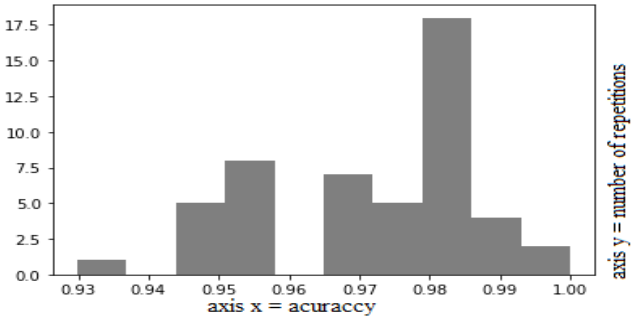

Figure 5. Histogram for 40 neurons

4.2. KNN Results

The training carried out by the KNN classifier follows the same methodology previously used in MLP. However, as evaluation metrics they are adjusted with variation of the parameter k, number of neighbors..For the KNN algorithm, a large variation in thenumber of neighbors was tested, following the form thatk= 2n+ 1.Thus, in an empirical way k between 1 and 100 were tested. Finally, it was noticedthat the best results were obtained forkcorresponding to 3,13,15,17. The following tableshows how the evaluation metrics were distributed.

Table 2. Data of KNN training

Number of Neighbors	Mean Accuracy	Bigger Accuracy	Less Accuracy	Standard Detour
3	0.9366	0.9623	0.9373	0.0163
13	0.9596	0.9734	0.9473	0.0089
15	0.9578	0.9734	0.9385	0.0116
17	0.9543	0.9734	0.9298	0.0151

Next, we have a graph and histogram that illustrates the behavior of the KNN classifier for k = 13.

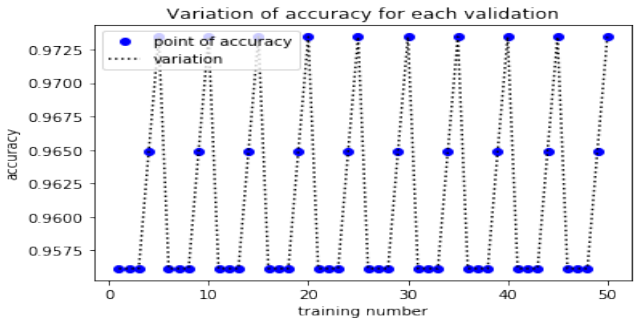

Figure 6. Graphic of KNN validation for 13 neighbors

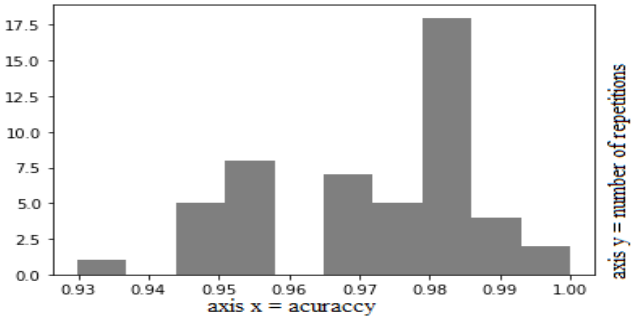

Figure 7. Histogram for 13 neighbors

4.3. Discussion

To better discuss the data, use the following graph, which compares the best associations for each algorithm: MLP and KNN.

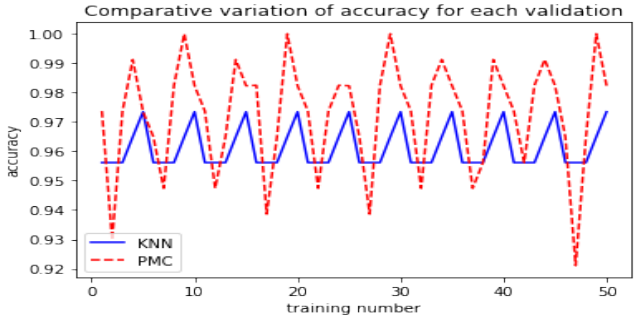

Figure 8. Comparative performance between MLP e KNN classification

Note that the two algorithms behave well after classification, however, note that the MLP network provided the best results. Although the KNN classifier presents low deviations from the standards, an average of them does not register an improvement with variation in the parameter of the number of neighbors.In addition, we can observe that the KNN classifier graphically shows a generalization difficulty even in its best structure. It is observed that the algorithm falls to respective lows and takes some time to converge to a higher value, and the path is shown to be unreliable due to its minimal variation. On the other hand, the MLP neural network shows graphically that although it starts with a

very low accuracy, it manages to evolve gradually to converge in higher values, and the decrease in the same way.

5. Conclusions

In summary, we concluded in this work that the MLP neural network method is more reliable to be used in the classification of breast cancer, considering that its results were superior in several aspects to the KNN classifier. However, each database has its peculiarity, in this case presented the MLP network was better than the KNN. However, in another database the result could be different. Therefore, this work can influence future research that aims to use and compare MLP and KNN in the diagnosis of other pathology's.

References

Abdou, Y., Attwood, K., Cheng, T.-Y. D., Yao, S., Bandera, E. V., Zirpoli, G. R., Ondracek, R. P., Stein, L., Bshara, W., Khoury, T., et al. (2020). Racial differences in cd8+ t cell infiltration in breast tumors from black and white women. *Breast Cancer Research*, 22(1):1–10.

Athitsos, V., Alon, J., and Sclaroff, S. (2005). Efficient nearest neighbor classification using a cascade of approximate similarity measures. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 486–493. IEEE.

Athitsos, V. and Sclaroff, S. (2005). Boosting nearest neighbor classifiers for multiclass recognition. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops*, pages 45–45. IEEE.

Dellani, P. R. and Borges, P. S. Implementação de um método para a classificação de microcalcificações pleomórficas invariante a posição, escala e orientação com rede neural de kohonen em mamografia convencional.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Guimarães, A. M., Mathias, I. M., Dias, A. H., Ferrari, J. W., and Carlos, R. d. O. (2008). Módulo de validação cruzada para treinamento de redes neurais artificiais com algoritmos backpropagation e resilient propagation. *Publicatio UEPG: Ciências Exatas e da Terra, Agrárias e Engenharias*, 14(01).

Haddad, N. and SILVA, M. D. (2001). Mortalidade por neoplasmas em mulheres em idade reprodutiva-15 a 49 anos-no estado de são paulo, brasil, de 1991 a 1995. *Revista da Associação Médica Brasileira*, 47(3):221–230.

Haykin, S. (2001). Redes neurais: princípios e prática, 2ª edição, tradução: Paulo martins engel. *Editora: Bookman, Porto Alegre, Cap*, 1(2):3.

Silva, I. d., Spatti, D. H., and Flauzino, R. A. (2010). Redes neurais artificiais para engenharia e ciências aplicadas. *São Paulo: Artliber*, 23(5):33–111.

Zeybek, Z., Çetinkaya, S., Hapoglu, H., and Alpbaz, M. (2006). Generalized delta rule (gdr) algorithm with generalized predictive control (gpc) for optimum temperature tracking of batch polymerization. *Chemical engineering science*, 61(20):6691–6700.