Breast cancer Prediction Using ANN, SVM, SGD

***Abstract*—Breast Cancer is the second cause of death among women. Earlier detection followed by the treatment can reduce the risk. There will be some cases where even the medical professionals can make mistakes in identifying the disease. So, with the help of the technology in Machine Learning and Artificial Intelligence can substantially improve the diagnosis accuracy. Artificial Neural Networks (ANN) has been widely in healthcare domain so as in detecting the cancer in earlier stage. Here, I am proposing a Gradient based Back Propagation Artificial Neural Networks (GBP-ANN). Along with ANN, I am also using Stochastic Gradient Descent Classifier (SGD) and SVM with Linear and Gaussian Kernel and I will compare the performances and trade-off's between models. The development of this technique is promising as intelligent component in medical decision support systems.**

***Keywords*—Breast cancer, Artificial Neural Networks, Support Vector Machine, Stochastic Gradient Descent, Cancer prediction.**

I. INTRODUCTION

Artificial Neural Networks (ANN) is one of the best Artificial Intelligence techniques for common Data Mining tasks, such as classification and regression problems (Supervised Learning). A lot of research already showed that ANN delivered good accuracy in breast cancer diagnosis. In this method, first it involves in building the network for the model, parameters to be tuned in the beginning of the training process such as number of input nodes, hidden nodes, output nodes and initial weights, learning rates, and activation function.

We know that it takes long time for training process due to complex architecture and parameter update process in each iteration and that has more computation cost. So, to overcome this I am using the Gradient based Back Propagation techniques and to speed up the computation I am leveraging the **Deep Learning platform** (**Cloud GPU**) called **FloydHub.** In this project, I implement the Artificial Neural Network from scratch for efficiently detecting the Breast Cancer. Here to calculate the performance of the model I am using various performance metrics like Accuracy, Specificity, Sensitivity, Recall, Precision and visualizing it by using Confusion matrix.

II. DESCRIPTION

The main goal of this project is to detect or predict whether a tumor found in breast cancer is a Malignant or Benign.

Dataset:

The Breast Cancer Wisconsin (Diagnostic) dataset was taken from a Kaggle competition and can be found at UCI Machine Learning Repository. The dataset has a dimension of 569 x 32, it has 30 real attributes and one numeric attribute (id field), and one categorical attribute, which is a class label. Since this is a two-class classification problem which is also called as Binary Classification, so this dataset has two class values for diagnosis they are M (Malignant) and B (Benign).

Ten real-valued features are computed for each cell nucleus:

a) radius (mean of distances from center to points on the perimeter) b) texture (standard deviation of gray-scale values) c) perimeter d) area e) smoothness (local variation in radius lengths)f) compactness (perimeter^2 / area - 1.0) g) concavity (severity of concave portions of the contour) h) concave points (number of concave portions of the contour) i) symmetry j) fractal dimension ("coastline approximation" - 1)

The mean, standard error and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius.

All feature values are recorded with four significant digits.

This dataset doesn't contain any missing attribute values. The class distribution is 357 benign and 212 malignant.

Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.

III. METHODOLOGIES

There are mainly two steps involved in this project

1. Data Preparation
2. Building the Model

1. Data Preparation:

This is an important step in a cleaning and processing a data that is unstructured with inconsistencies. The step is crucial for the model building, it is worth noting that good data is more important that a good model.

The dataset contains 569 samples with 33 features. The last column unnamed: 32 is an empty column and it is of no use, so I remove it.

Next, separated feature data and label data (target). The feature data is that going to be trained and learn from the model and the target data is a diagnosis, which says whether the tumor is malignant or benign.

Since, the true labels are categorical variables (M, B) so, I decoded it to 1, 0.

Next, I had partitioned the data into two sets i.e., training and testing sets. Here I have randomly sampled from the data and split it into 70:30 ratio, the reason why I randomly selecting the data from the whole data is that sometimes, selected data that will be highly correlated that might affect the results.

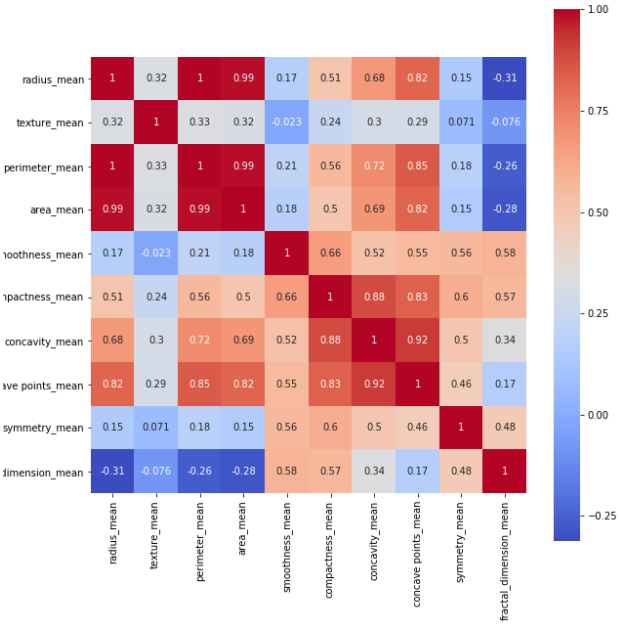
The reason behind splitting the data into training and test sets is that we can't train the model on the entire data because it will lead to overfitting. That is, model will perform well on the training data and fails on the new unseen data (i.e., training error is low and testing error is high or high variance and low bias).

Train-Test Split

The final task in data preparation phase is feature scaling, here I convert data so that all values are scaled to be within the same range. It makes easier for our model to predict and to perform faster.

The StandardScaler function will transform each variable of the data to have a mean =0 and standard deviation =1.

I used scatterplot to see the correlation between the features for feature selection.



Model:

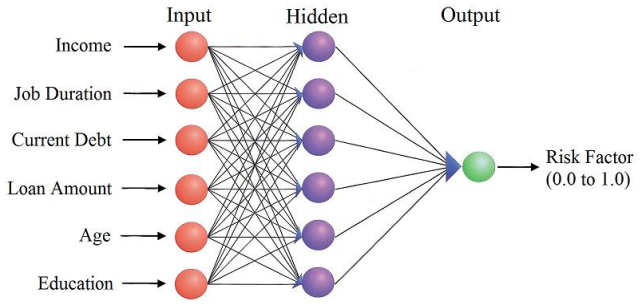


Fig: ANN

Artificial Neural Networks (ANN) is a part of Artificial Intelligence that uses various optimization techniques to train and learn from the data and predict for the new unseen data(test).

Steps:

1. Network Architecture
2. Forward Propagation
3. Backward Propagation
4. Updating Network Weights
5. Testing, Performance calculation

Network Architecture involves defining parameters for Network (Input, Output, Hidden, learning rate, number of epochs).

Forward Propagation is first step in training the network, that multiplies the input node with weights and it is given the activation function to generate the value between [0, 1]. The values will be input for next layer and then it gets multiplied with the next layer weights.

$$h_j = \sum_{k=1}^{p} w_{jk} x_k$$

$$v_j = \frac{1}{1 + e^{-h_j}}$$

$$g_i = \sum_{j=1}^{m} w_{jk} v_j$$

Activation Function:

As an activation function, I had used sigmoid function or logistic function. Since, it is a binary classification problem, it will return value either 0 or 1.

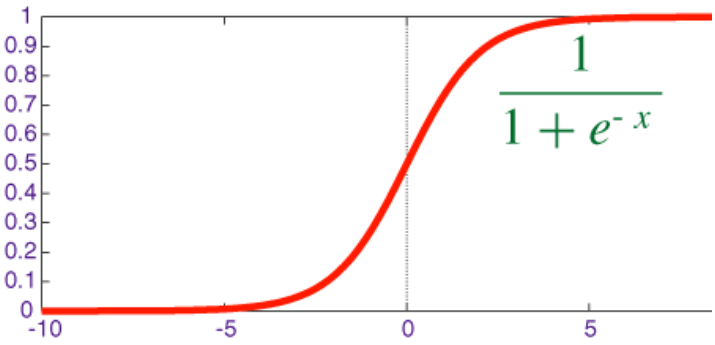


fig: sigmoid function returns values [0, 1]

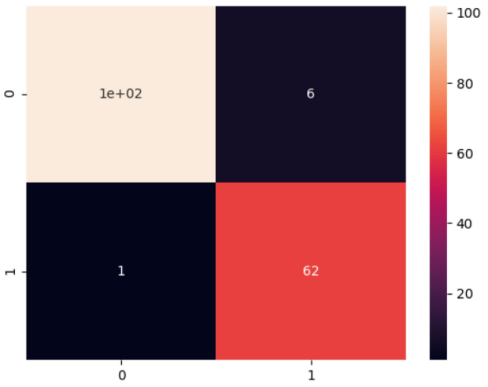
Backward Propagation is where the network actual learns from the data. I used gradient descent to train the model. In this phase, the network first checks error in the network after the forward propagation, error = True- Predicted. Then checks the error in each layer by multiplying the error in previous layer, derivative of current layer.

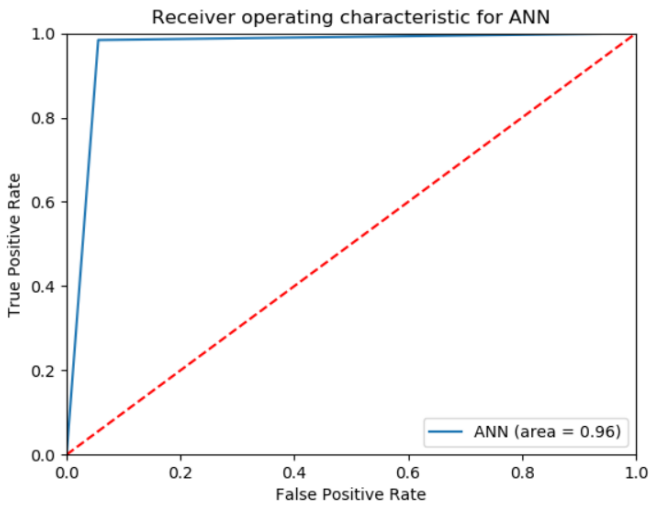
$$\delta_i = y.(1-y).(y^d - y)$$

Then, updating the weights to make predictions that are closely matches with the true value.

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_i h_j$$

After training the network with epoch =1000 or 5000, the network will learn the data and adjusts weights according to predict the correct value. Here, next testing data is given to predict and evaluate how network performs on the unseen data. The evaluation of network is calculated by various performance indicator like Precision, Recall, Accuracy, FPR, TPR, ROC, AUC score.





Stochastic Gradient Descent Classifier (SGD):

Stochastic Gradient Descent (SGD) is a simple and efficient optimization technique to discriminate learning of linear classifiers under convex loss functions e.g., SVM, Logistic Regression. It is widely used in large scale learning to find the solution efficiently. It finds the minima of a loss function. Unlike the gradient descent, SGD will consider only one random point while changing the weight whereas, gradient descent will consider the entire training data.

SGD can scale to large dataset with 105 x 105 dimensional.

SGD is sensitive to feature scaling.

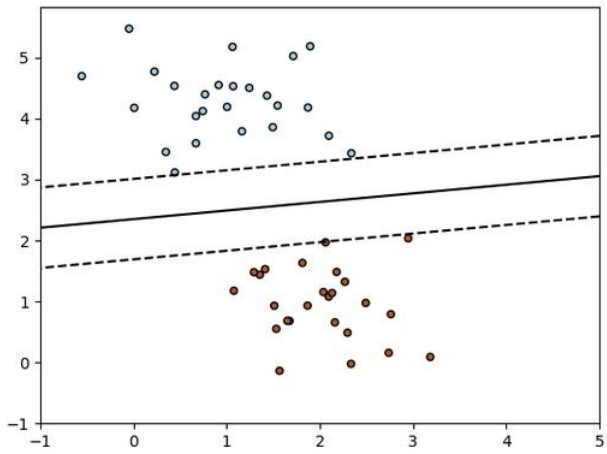
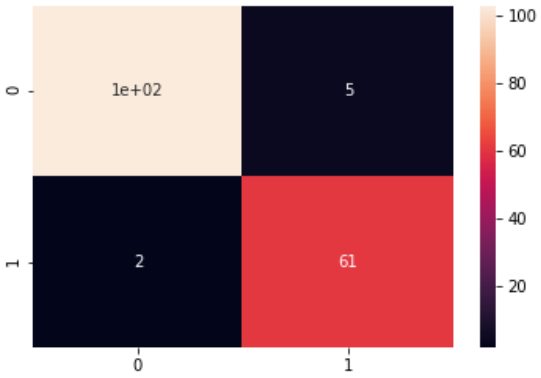
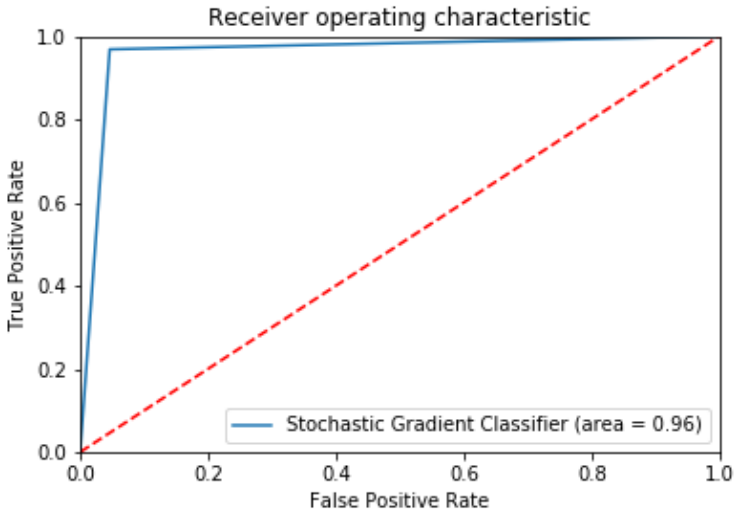


Fig: SGD with 'hinge loss'



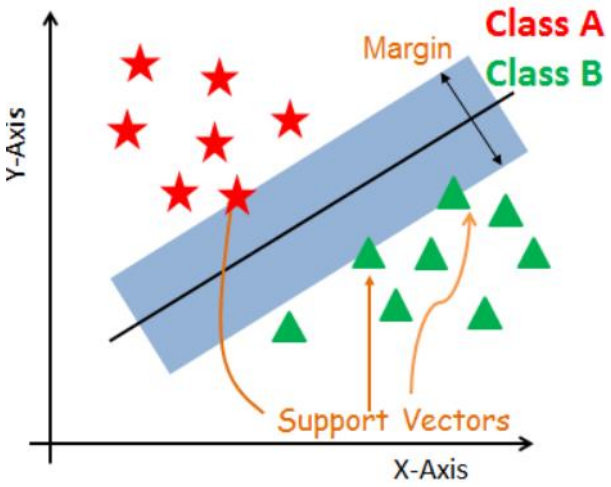


Support Vector Machines (SVM):

SVM is a Supervised Machine Learning classification algorithm. In case of linearly separable 2D data al typical Machine Learning algorithm will try to find the boundary that divides the data in such a way that misclassification error is minimized.

SVM works differently from other classification algorithms in the way that it chooses decision boundary that maximizes the distance from the nearest neighbor data points of classes. It not only finds the decision boundary; it finds the optimal decision boundary. SVM constructs the hyperplane in multidimensional space o separate different classes. The main ideas behind SVM is to find the Maximum Marginal Hyperplane (MMH) that can best separate the dataset into classes. So, it iteratively performs

to find the optimal hyperplane that can minimize the misclassification error.



Non-linearly separable planes: SVM Kernels

If the data is not linearly separable then linear hyperplane doesn't work. So, SVM uses the Kernel technique to transform the input space to higher dimensional space.

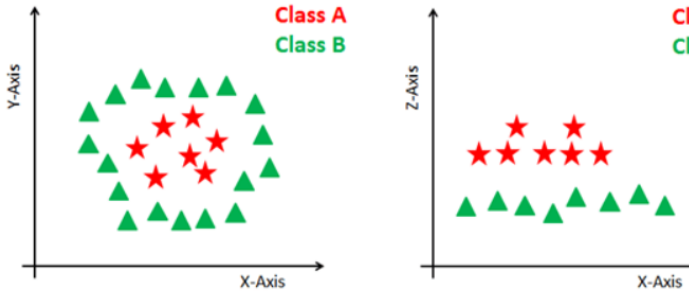
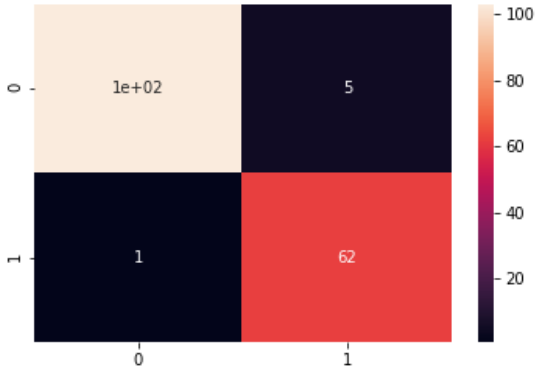
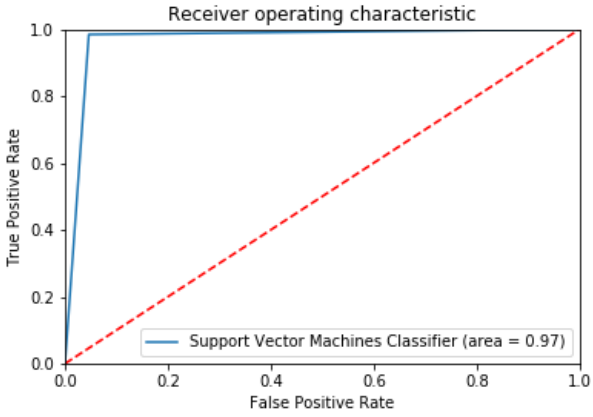


Fig on the left is non-linearly separable data is then given to SVM with kernel. Now the figure on the right is linearly separable by hyperplane. In other words, SVM kernel non-linearly separable problems to linearly separable by adding more dimensions to it. It builds more accurate classifier.

Linear Kernel

It can be used as normal dot product between two given observations. K(x, xi) = sum(x * xi)



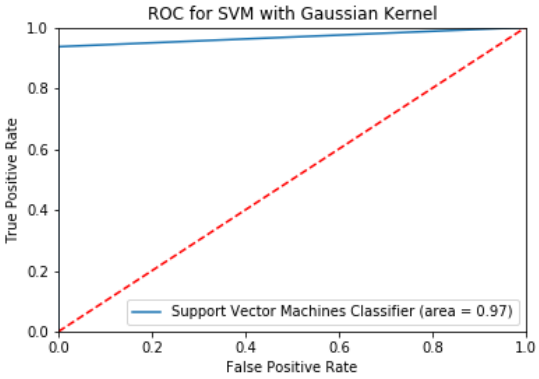


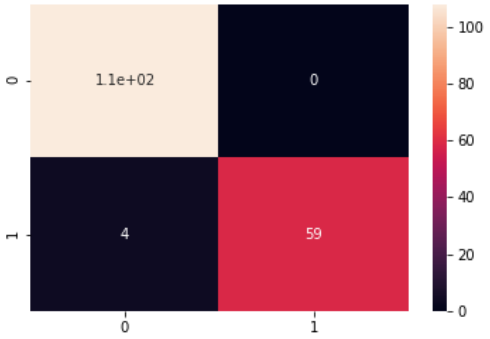
Gaussian Kernel

Radial Basis function will map the input space to infinite dimensional space.

$$K(x,xi) = exp(-gamma * sum((x - xi^2))$$

The gamma is a tuning parameter which fit the training data, higher the value more chance to overfit on training data.





IV. RESULTS:

Model	Accuracy (%)	Precision	Recall	F1-Score	AUC	Execution Time (sec)
ANN	**95.90**	**0.9117**	**0.9841**	**0.9465**	**0.9649**	**30.7725**
SGD	**94.74**	**0.9145**	**0.9523**	**0.9302**	**0.9517**	**0.015621**
SVM - Linear	**96.49**	**0.9334**	**0.9841**	**0.9538**	**0.9689**	**0.015629**
SVM - Gaussian	**97.66**	**1.00**	**0.9365**	**0.9672**	**0.9682**	**0.015619**

V. CONCLUSION

Classifiers works differently on different nature of data. There is trade-off between the accuracy of prediction on test data, and higher computation time and power. Simpler model will have low variance, high bias whereas complex models can perform well in prediction and can have high accuracy and sometimes it tries to generalize every single data point in the data which can lead to overfitting.

ANN performs well on the data, but it takes high computation power and time to train it. So, optimization should be used in order to find the predictions in less time and iterations, so I have used to gradient descent optimization while training. For 5000 epochs for training and testing ANN took 30.772 sec.

The results show that this model works well in predicting the cancer. It can be used as intelligent component in medical decision making.

REFERENCES

[1]. https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

[2]. https://www.kaggle.com/uciml/breast-cancer-wisconsin-data/home

[3]. http://ftp.cs.wisc.edu/math-prog/cpo-dataset/machine-learn/cancer/WDBC/

[4]. M. H. Khan, "Automated breast cancer diagnosis using artificial neural network (ANN)," *2017 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS)*, Shahrood, 2017, pp. 54-58.

[5]. Nastac, P. Jalava, M. Collan, Y. Collan, T. Kuopio and B. Back, "Breast cancer prediction using a neural network model," *Proceedings World Automation Congress, 2004.*, Seville, 2004, pp. 423-428