Comparison of SVM and Neural Networks (with and without Backpropagation) on performance over Classification of Breast Cancer

Aman Arora
18121004
aarora@ee.iitr.ac.in

Utkarsh Mathur
18121024
umathur@ch.iitr.ac.in

Abstract

The project aims to develop classifier for Classification of Breast Tumor into Malignant (cancer tumor) and Benign (non-cancer tumor) using features obtained from several cell images. The dataset we used for this purpose was Breast Cancer Wisconsin dataset from Scikit Learn. The classification process was carried out by three models, Support Vector Machine (SVM), Neural Network (with Particle Swarm Optimizer), and Neural Network (with Gradient Descent). The main objective is to compare these three models and find the most suitable model.

Main Objectives

- Comparison of performance of various models on classification of dataset with small number of training examples.

Status and Other Details

- Fully completed
- Percentage contribution of members:
  Aman Arora – 50%
  Utkarsh Mahut – 50%
- Total time spent on the project: 3 weeks

1. Introduction

Breast cancer is the most common cancer amongst women in the world. It accounts for 25% of all cancer cases, and affected over 2.1 Million people in 2015 alone. It starts when cells in the breast begin to grow out of control. These cells usually form tumors that can be seen via X-ray or felt as lumps in the breast area.

Early diagnosis significantly increases the chances of survival. The key challenge against its detection is how to classify tumors into malignant (cancerous) or benign (non-cancerous). A tumor is considered malignant if the cells can grow into surrounding tissues or spread to distant areas of the body. A benign tumor does not invade nearby tissue nor spread to other parts of the body the way cancerous tumors can. But benign tumors can be serious if they press on vital structures such as blood vessels or nerves.

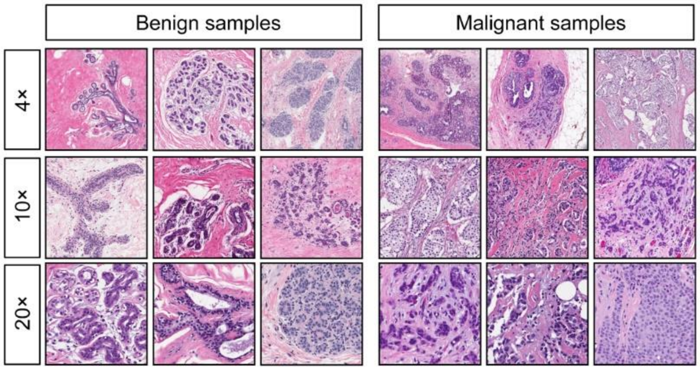
Machine Learning technique can dramatically improve the level of diagnosis in breast cancer. Research shows that experienced physicians can detect cancer by 79% accuracy, while a 91% (sometimes up to 97%) accuracy can be achieved using Machine Learning techniques.

2. Dataset

In this study, our task is to classify tumors into malignant (cancerous) or benign (non-cancerous) using features obtained from several cell images. The dataset was originally developed by Dr. William H. Wolberg, W. Nick Street, and Olvi L. Mangasarian from University of Wisconsin and is famously know as “Breast Cancer Wisconsin Dataset”. 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.

The dataset has 569 instances, 357 for benign and 212 for malignant. There are 30 features in the dataset each of which is described below: -

1. mean radius = mean of distances from center to points on the perimeter
2. mean texture = standard deviation of gray-scale values
3. mean perimeter = mean size of the core tumor
4. mean area
5. mean smoothness = mean of local variation in radius lengths
6. mean compactness = mean of perimeter^2 / area - 1.0
7. mean concavity = mean of severity of concave portions of the contour
8. mean concave points = mean for number of concave portions of the contour
9. mean symmetry
10. mean fractal dimension = mean for "coastline approximation" – 1
11. radius error = standard error for the mean of distances from center to points on the perimeter
12. texture error = standard error for standard deviation of gray-scale values
13. perimeter error
14. area error
15. smoothness error = standard error for local variation in radius lengths
16. compactness error = standard error for perimeter^2 / area - 1.0
17. concavity error = standard error for severity of concave portions of the contour
18. concave points error = standard error for number of concave portions of the contour
19. symmetry error
20. fractal dimension error = standard error for "coastline approximation" – 1
21. worst radius = "worst" or largest mean value for mean of distances from center to points on the perimeter
22. worst texture = "worst" or largest mean value for standard deviation of gray-scale values
23. worst perimeter
24. worst smoothness = "worst" or largest mean value for local variation in radius lengths
25. worst compactness = "worst" or largest mean value for perimeter^2 / area - 1.0
26. worst concavity = "worst" or largest mean value for severity of concave portions of the contour
27. worst concave points = "worst" or largest mean value for number of concave portions of the contour
28. worst fractal dimension = "worst" or largest mean value for "coastline approximation" – 1

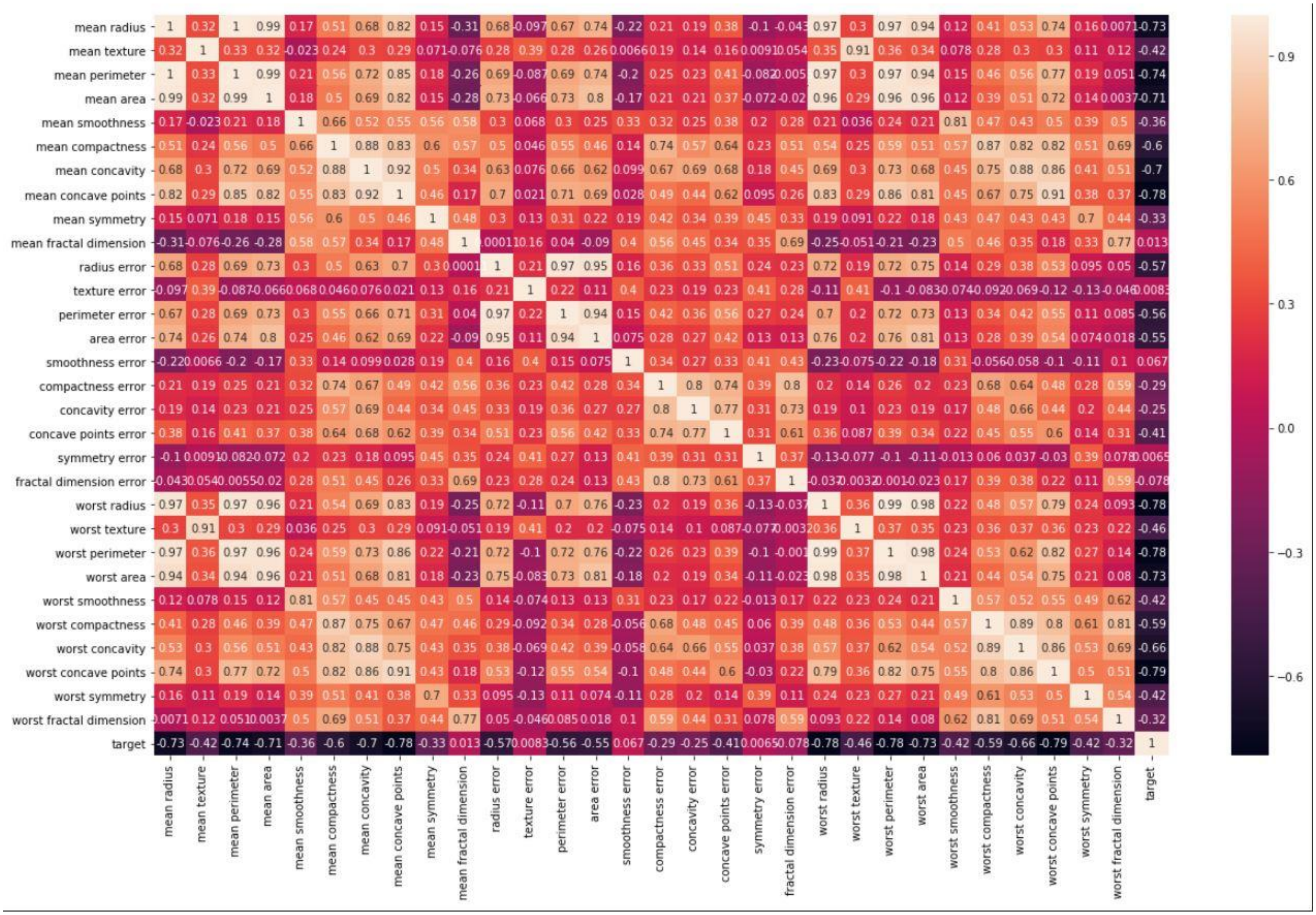


(Figure 1 – Sample images used for classification)

As shown in the figure 2, the heat map clearly shows that the dependencies between features is very less, hence there is no need to apply dimension reduction techniques like Principal Component Analysis (PCA).

In all the three model (described in the next sections), the data is scaled using Min-Max scaling. The transformation is best described as:

X_std = (X - X.min) / (X.max - X.min)
X_scaled = X_std * (max - min) + min



(Figure 2 – Covariance Matrix of all features)

3. Models

The main aim of the project is to compare the performances of Support Vector Machine model and Neural Networks model (with and without backpropagation) on a classification job where the number of instances is relatively few (<1000).

All these models are trained on 512 training examples and tested on 57 validation examples which are prepared by a 90 - 10 split of annotated data.

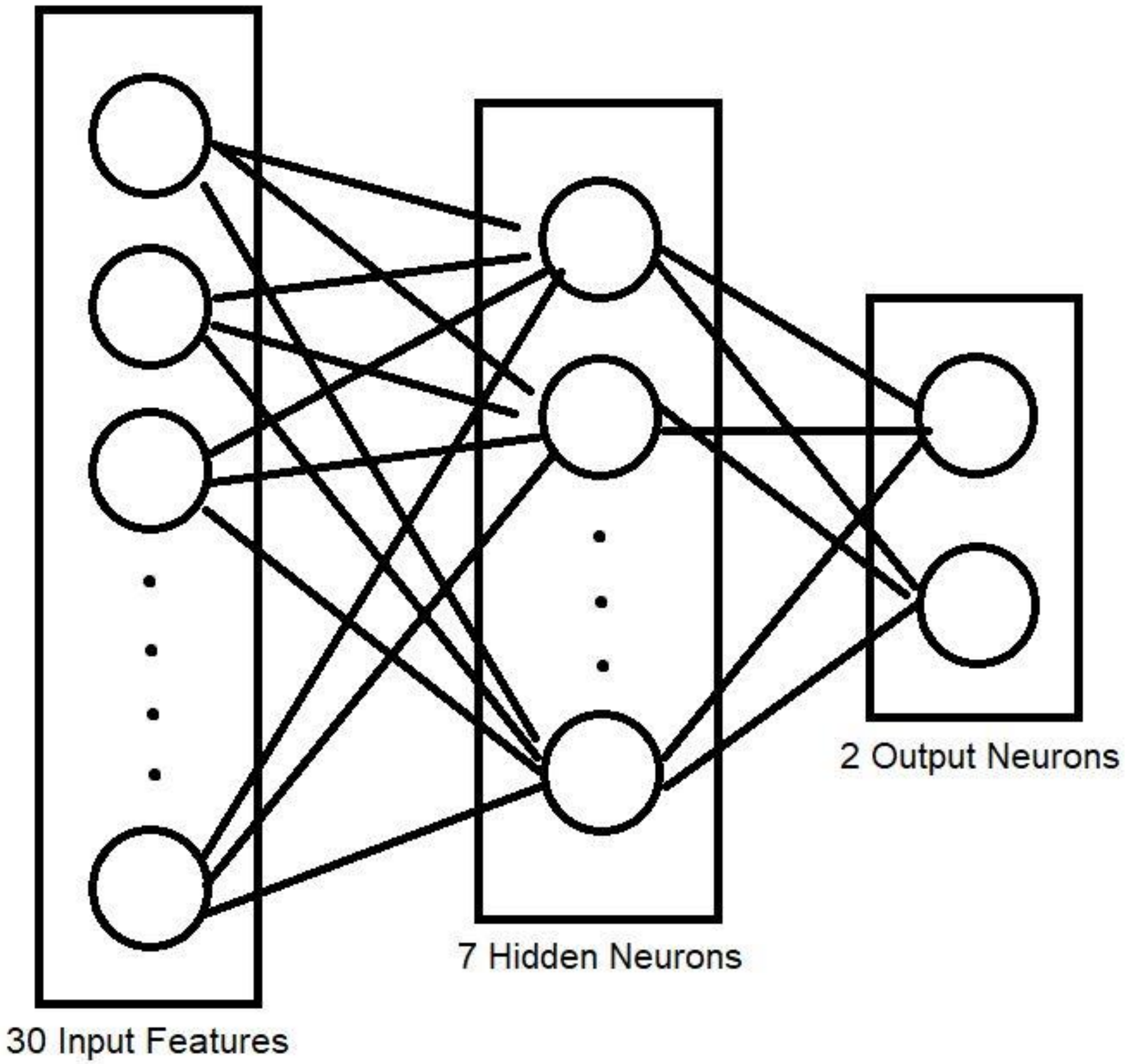
As described above the data is scaled using Min-Max Scaling. The metric used to measure the performance of the models is accuracy because the targets are fairly uniformly distributed.

A. Support Vector Machine (SVM)

A Support Vector Machine (SVM) is a binary linear classification whose decision boundary is explicitly constructed to minimize generalization error. It is a very powerful and versatile Machine Learning model, capable of performing linear or nonlinear classification, regression and even outlier detection.

SVM is well suited for classification of complex but small or medium sized datasets.

The model uses RBF kernel with degree 3 and the regularization parameter is 1. The result from this model is that accuracy obtained is 99.12% on validation set.



(Figure 3 – Architecture of Neural Network model)

B. Neural Networks (with Backpropagation)

The second model used to train the dataset is Neural Networks. The model is composed of Multi-Layer Perceptron with 1 hidden layer, architecture of which is shown in the figure 3.

The model uses Adam optimizer over Categorical Cross-entropy loss function to train the weights. The model is trained for 100 epochs and the batch-size is 16 examples per batch. After training, the accuracy obtained on validation set is 87.71%.

C. Neural Networks (without Backpropagation)

The last model uses Particle Swarm Optimization (PSO) technique to optimize the weights of the neural network and increase the accuracy of the model. This approach rules out the use of backpropagation to optimize weights of the neural network.

A basic variant of the PSO algorithm works by having a population (called a swarm) of candidate solutions (called particles). All of particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. The particles fly through the problem space by following the current optimum particles. A number of particles are initialized randomly within the search space. Each particle has a very simple 'memory' of its personal best solution so far, called 'pbest'. This is the best solution (fitness) it has achieved so far. The global best solution for each iteration is also found and labelled 'gbest'. It is the best value, obtained so far by any particle in the population. On each iteration, every particle is moved a certain distance from its current location, influenced a random amount by the pbest and gbest values.

$$v_i(t+1) = w.v_i(t) + c_1.r_1(pbest_i - x_i(t)) + c_2.r_2(gbest - x_i(t))$$

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

here, i є 1 …. P (P = number of particles)

r_1, r_2 are random numbers uniformly distributed between [0,1]

c_1, c_2 are acceleration constants values of which are taken 2

In training this model, the swarm consists of 12 particles of 233 dimensions (all the weights of the neural networks). Now we use this swarm intelligence to optimize the weights of the neural network by optimizing their fitness value over fitness function. The fitness function that's used here is Categorical Cross-entropy loss function of the neural network. This way, by optimizing the fitness function it automatically reduces the loss function which in turn increases the accuracy of classifier.

The architecture used for this model is the same as the one in figure 3. The model was trained for 100 epochs with batch size of 16 examples per batch. The accuracy obtained over the validation set was 92.98%.

4. Results

The summary of the results from all the three models are formulated in the table below.

Model	Accuracy	Time taken
Model A	99.12%	3.14 seconds
Model B	87.71%	6.56 seconds
Model C	92.98%	19.34 seconds

(Table 1 – Summary of results of different models)
Model A – Support Vector Machine
Model B – Neural Networks (with backpropagation)
Model C – Neural Networks (without backpropagation)

5. Conclusions

The following conclusions can be drawn from the results of this project: -

1. Support Vector Machine (SVM) models performs a better classification task than Neural Networks models when the number of training examples are small (in our case only 512). In our case, the SVM model took half the time to get around 12% more accuracy than Neural Network (with backpropagation) and took around 1/6th time to get around 5% more accuracy than Neural Networks (without backpropagation).

2. When training on a smaller dataset, Neural Networks perform better with PSO than

Backpropagation, although it takes more time.

6. References

1. **Repository for codes of all the models** - https://github.com/007mathur/BreastCancerClassification
2. https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html#sklearn.datasets.load_breast_cancer
3. https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)
4. http://scikit-learn.org/stable/modules/svm.html
5. http://www.robots.ox.ac.uk/~az/lectures/ml/lect2.pdf
6. http://pyml.sourceforge.net/doc/howto.pdf
7. https://www.bcrf.org/breast-cancer-statistics
8. https://www.cancer.org/cancer/breast-cancer/about/what-is-breast-cancer.html
9. https://towardsdatascience.com/breast-cancer-classification-using-support-vector-machine-svm-a510907d4878
10. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC
11. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
12. https://en.wikipedia.org/wiki/Particle_swarm_optimization
13. https://github.com/piyush2896/PSO-for-Neural-Nets
14. https://ieeexplore.ieee.org/abstract/document/494215.?casa_token=m9uzu0BNarkAAAAA:BNuHp8AbZzycfCWZ87hPqJEQJ1pk2YMUH77q7YFj78NRytt5JrDvhKmq6R8DJPXCYDcb1AKjJQ