BREAST CANCER CLASSIFICATION

Assignment 2

ABSTRACT

This work uses artificial neural networks to classify breast cancer cases according to whether their tumors are benign or malignant.

22CSCI33H

Table of Contents

Breast Cancer

Breast cancer is one of the most severe diseases that affects women across the globe, regardless of their age or race. According to the National Center for Biotechnology Information (NCBI), breast cancer is the second most frequent cancer-related mortality among women worldwide[1]. This work aims to classify whether the breast tumor is benign or malignant.

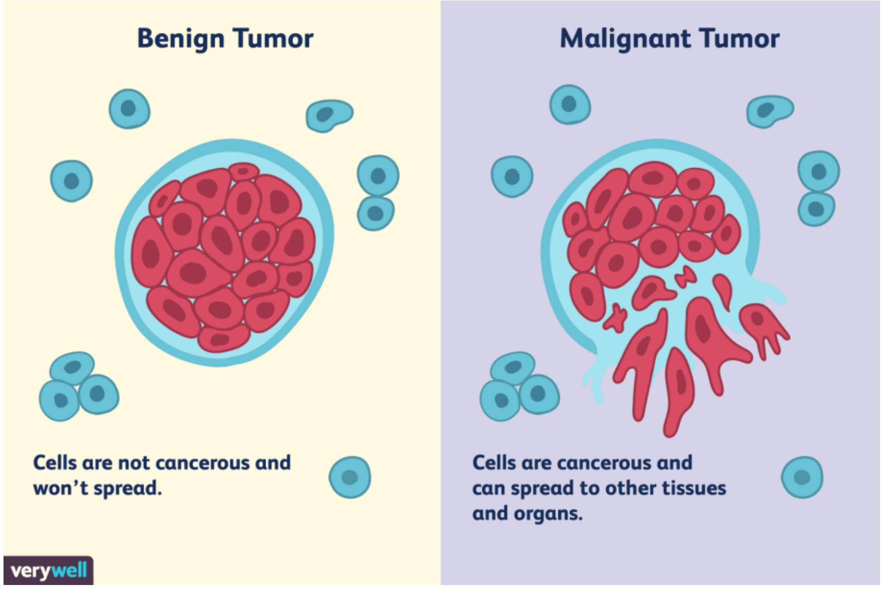

Benign Tumor
Malignant Tumor
Cells are not cancerous and won't spread.
Cells are cancerous and can spread to other tissues and organs.
verywell

Figure 1 : Benign and malignant tumors [2]

Dataset

Data Description

The breast cancer database was obtained from the University of Wisconsin Hospitals in Madison by Dr William H. Wolberg.

- **Data source :** UCI Machine Learning Repository: Breast Cancer Wisconsin (Original) Data Set
- **Dataset size :** 699 instances, 9 features, 1 target
- **Missing values :** Yes
- **Columns description :**

Name	Description	Data type	Values range
ID	Sample code number for each case.	Integer	--
Clump Thickness	Malignant cells are multi layered while benign cells are layered in a single band.	Integer	1-10
Uniformity of Cell Size	Cancer cells tend to vary in size. That is why this feature is valuable in saying whether the cells are cancerous or not.	Integer	1-10
Uniformity of Cell Shape	Cancer cells tend to vary in shape. That is why this feature helps to know if the cell is cancerous or not.	Integer	1-10

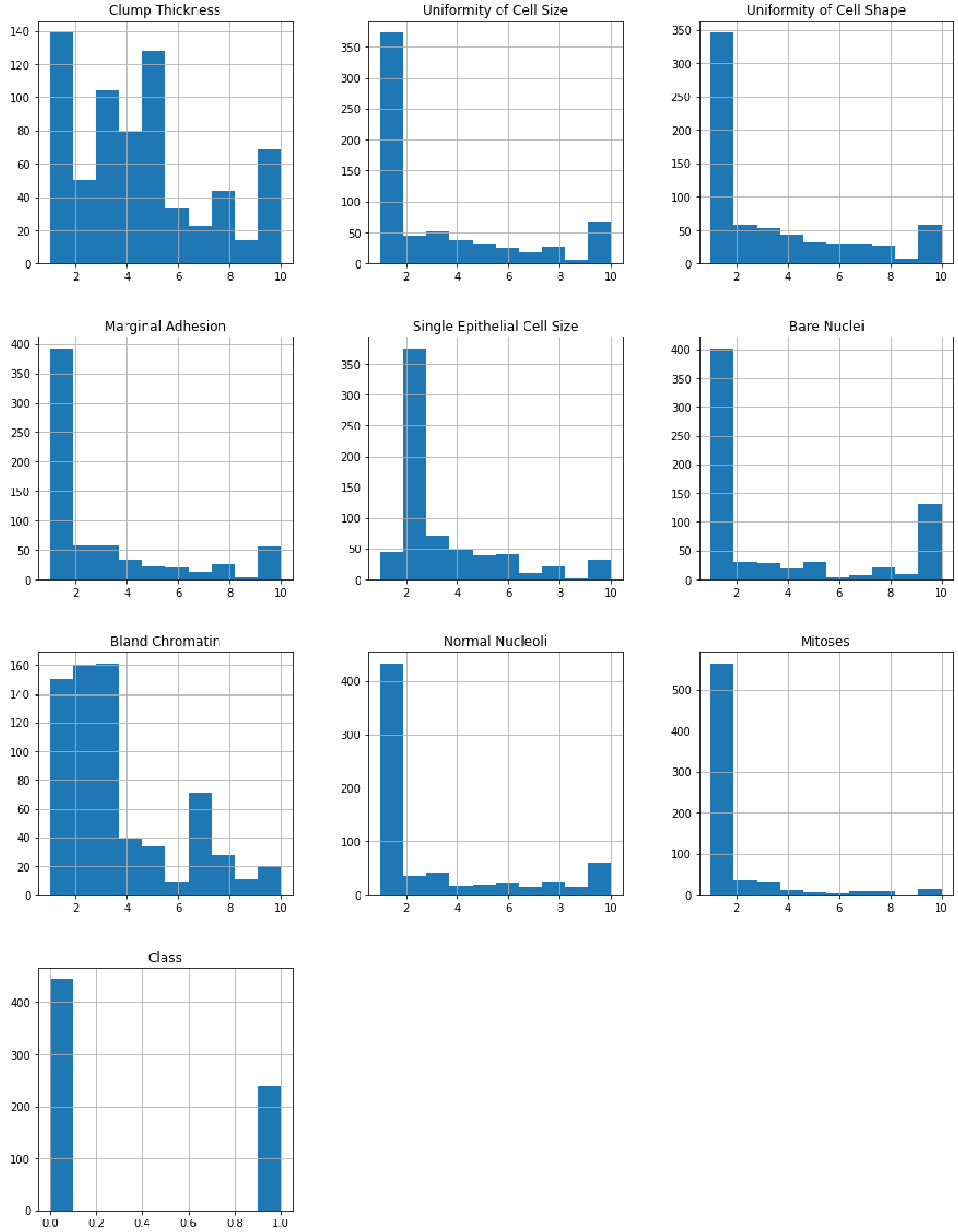
Marginal Adhesion	Normal blood cells adhere to each other. In cancer cell this ability diminishes. So, loss of adhesion is a sign of malignancy.	Integer	1-10
Single Epithelial Cell Size	Epithelial cells that are enlarged significantly t out to be malignant.	Integer	1-10
Bare Nuclei	Term for nuclei not surrounded by cytoplasm. They are typically seen in harmless tumors.	Integer	1-10
Bland Chromatin	In malignant cells the chromatin is found to be rough in texture, while the nucleuses of noncancerous cells have an unvarying texture.	Integer	1-10
Normal Nucleoli	In malignant cells the nucleoli are largely visible and are sometimes present in a greater quantity. In contrast, benign cells have microscopic nucleoli.	Integer	1-10
Mitoses	Mitosis is a process that is inherently managed by the genes inside every cell. If this control fails, a single cell can multiply to make new Cells that can lose control and are cancerous.	Integer	1-10
Class	The tumor is benign or malignant.	Integer	2 or 4

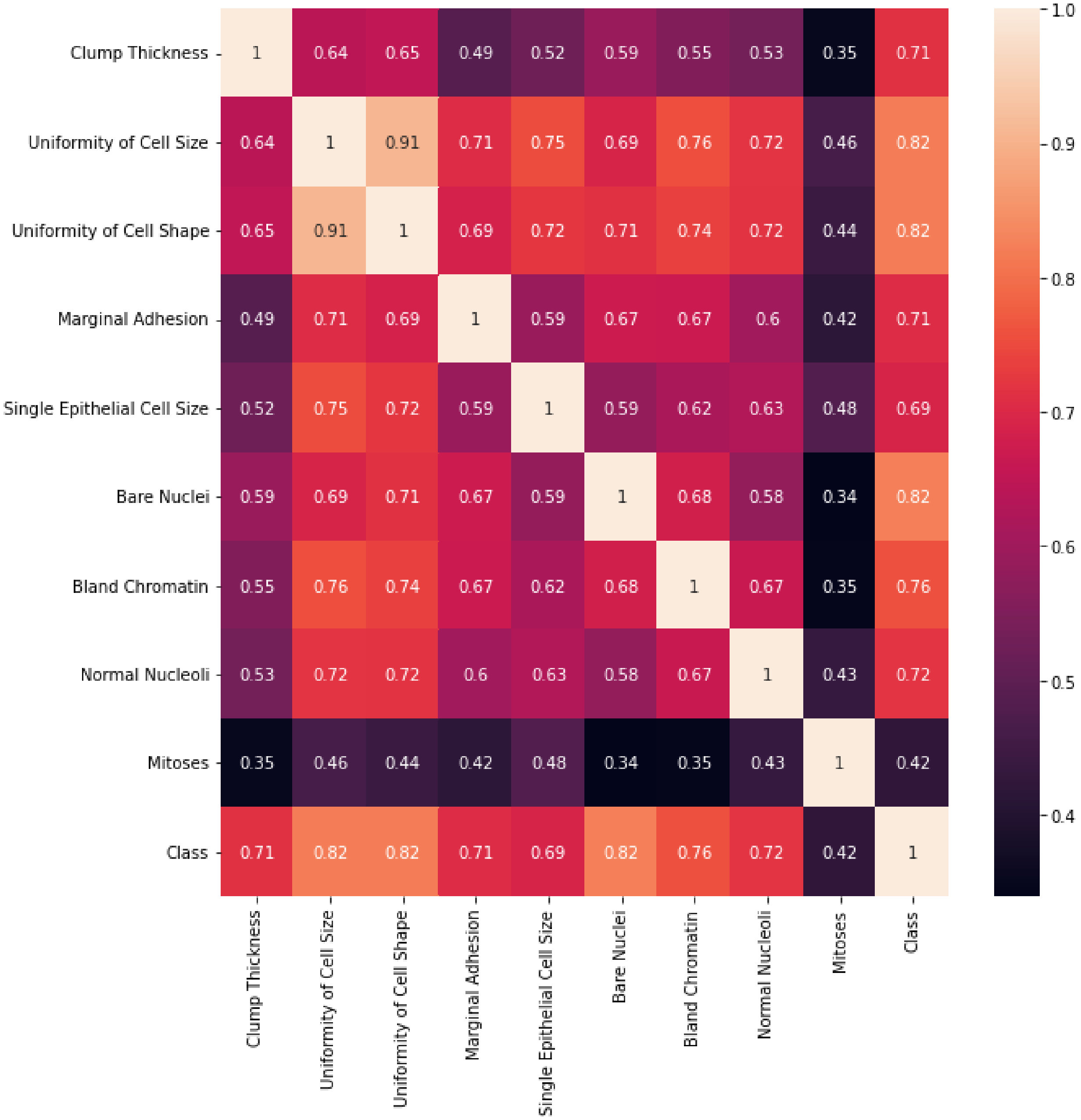
- **Number of classes :** 2

Data Pre-Processing

The dataset contains some missing values, so we dropped them because it is a medical record, so we need to have accurate values. Also, we dropped the ID column as it does not affect our data. Moreover, since ‘Uniformity of Cell Size’ is highly correlated with another feature, it has been removed.

Data Visualization





Multi-Layer Perceptron (MLP)

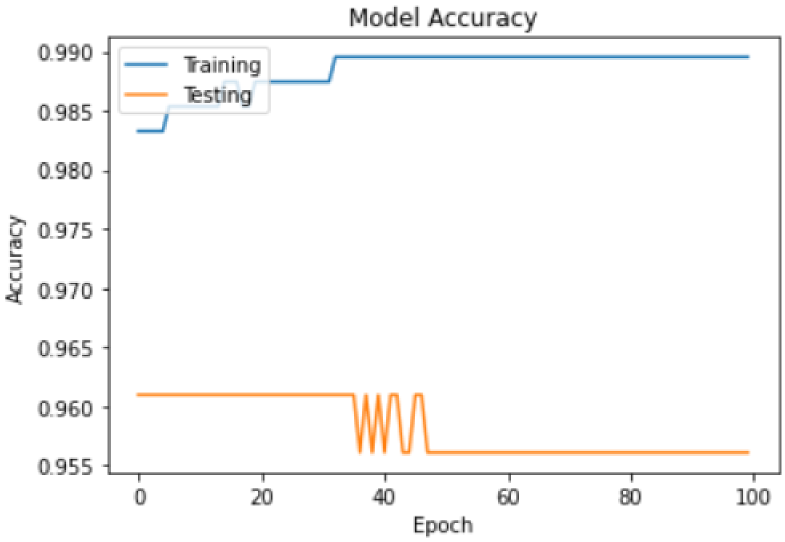
The Multi-Layer Perceptron (MPL) is a collection of perceptrons that are layered in numerous layers and link to one another. Furthermore, an MLP is a fully densely connected neural network in which every node on each layer is linked to every other node on the next layer. Nodes inside a single layer, on the other hand, do not share any connections.

Code

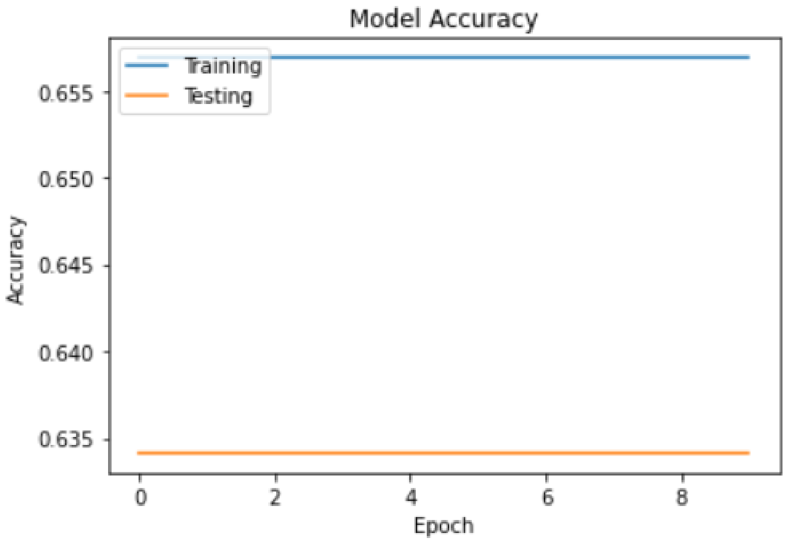
The code implements MLP network and tuned the hyperparameters to choose the best model.

Name	Model 1	Model 2
Layers	▪ 1 input layer attached to one hidden layer ▪ 1 hidden layer ▪ 1 output layer	▪ 1 input layer attached to one hidden layer ▪ 2 hidden layers ▪ 1 output layer
Perceptrons	▪ Input layer : 10 ▪ Hidden layer : 10 ▪ Output layer : 1	▪ Input layer : 15 ▪ Hidden layer 1 : 15 ▪ Hidden layer 2 : 10 ▪ Output layer : 1
Activation functions	▪ Input layer : ReLU ▪ Hidden layer : ReLU ▪ Output layer : Sigmod	▪ Input layer : SoftMax ▪ Hidden layer 1 : SoftMax ▪ Hidden layer 2 : SoftMax ▪ Output layer : ReLU
Optimizer	adam	RMSprop with learning rate 0.01
Loss	binary_crossentropy	mean_squared_error
Metrics	accuracy	accuracy
Batch size	32	5
Epochs	100	10
Threshold	50%	50%
Accuracy	95%	63%

Learning curves

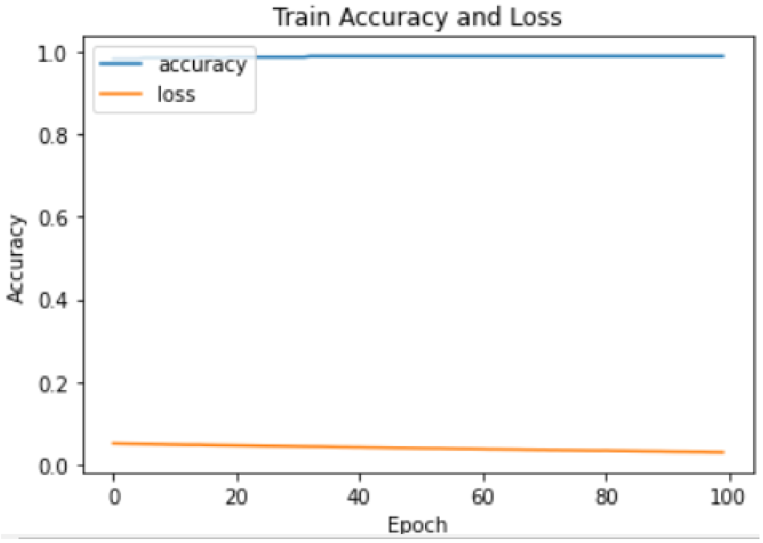


Model 1

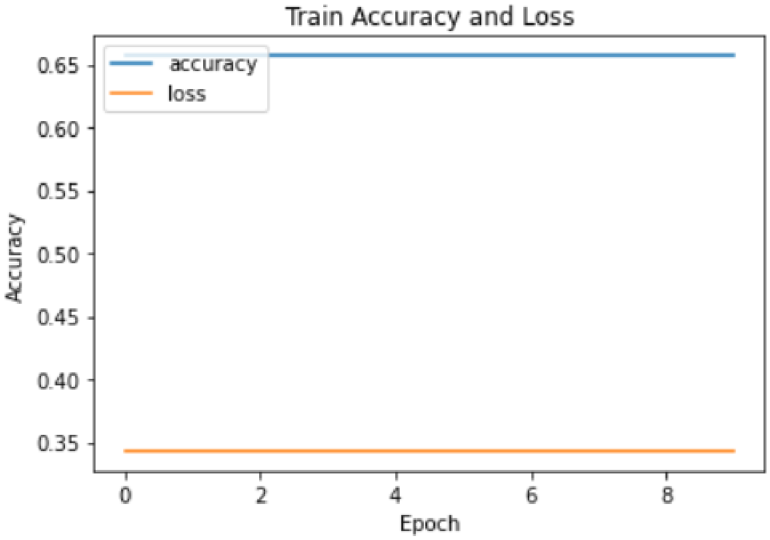


Model 2

Evaluation Metrics



Model 1



Model 2

Results

Model 1 achieved accuracy higher than model 2.

Radial Basis Function Network (RBFN)

Radial Basis Function networks are a type feedforward neural network, comprising three layers: input, hidden, and output. Each input is fed forwardly into the neurons of the hidden layer, of which each consists of a radial basis function (for instance Gaussian). The outputs of these are then weighted and summed to produce the output of the network. The network learns through adjusting parameters, namely the weights, the centers of each neuron, and radii r (unit width). Ultimately, the Euclidean distance, from the point being evaluated, to the centers of each neuron, is evaluated. The kernel functions are then applied to understand the influence of each neuron. The further away the neuron is, the lesser influence it has on the point. The network can then group similar points together via clustering, and these clusters can be used for classification.

Code

The outline architecture for a radial basis function networks was first defined, and an initial model was built. After examining the performance of this model, a second, hyper-tuned model was developed with better hyper-parameters. Both models were trained over 1000 epochs, with a batch size of 32.

First Model:

- Activation Function: Linear
- Loss: MSE
- Optimizer: Root Mean Squared Propagation

Second Model:

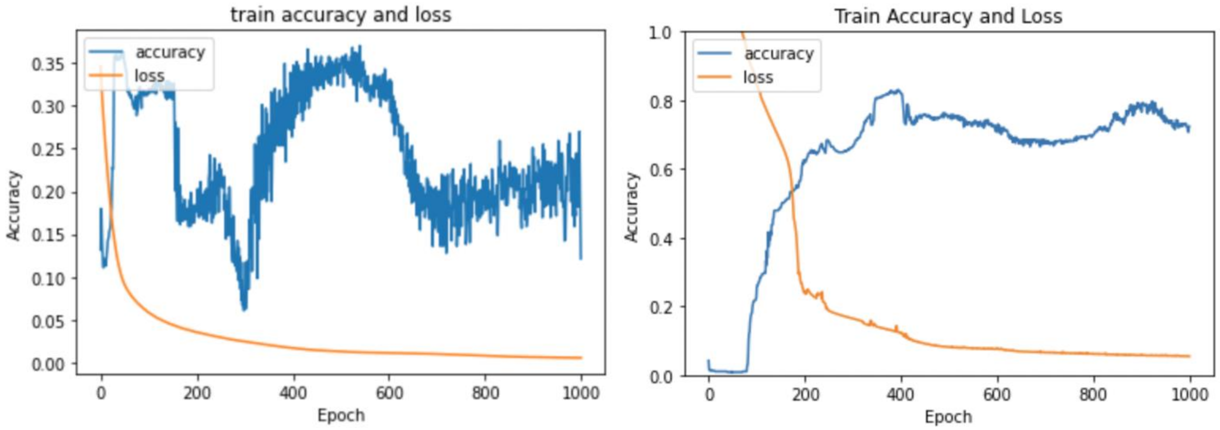
- Activation Function: ReLU
- Loss: Binary Cross Entropy
- Optimizer: Adam

Data structures

Classes were used to implement the hidden layer of RBF units, as well as to initialize the centers using kMeans clustering.

Evaluation Metrics and Results

Both loss and accuracy were plotted for the training performance of both models:



Left: Model 1, Right: Model 2

The testing loss and accuracy were also computed, and were as such:

Model 1:

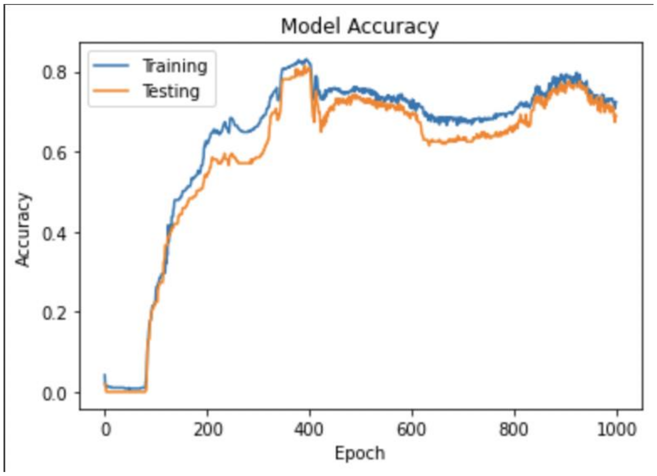
- Test loss: 0.0547
- Test accuracy: 12.20%

Model 2:

- Test loss: 0.4490
- Test accuracy: 78.54%

The second model greatly outperformed the first and is able to make more accurate tumor classification. However, given the medical context of the classification problem at hand, and the great risks accompanied by misclassifications, this accuracy is still too poor for this model to be applied in the area of breast cancer classification. It is fundamental to note that perhaps further tuning and refining of the model, both structurally and via its hyperparameters, may lead to a RBFN with sufficient accuracy, or perhaps the RBFN model is simply not successful in this application (on this data).

Learning Curve



The above graph depicts the learning curve for the optimized model (model 2). It can clearly be seen that in a low number of iterations (below 200 epochs) the model performs very poorly at classifying benign vs malignant tumors. However, as the number of iterations progresses, both training and testing accuracies increase, with the testing accuracy naturally being lower. The accuracies roughly plateau and stabilize after approximately 400 epochs, suggesting that the large epoch size used could've been significantly reduced to save computational effort and resources.

Deep Belief Networks (DBNs)

It's a generative model that addresses classical neural networks. It is used to construct unbiased values that could be stored in the leaf node. As for its architecture, the DBN uses a deep architecture of stacked RBMs that each model transforms nonlineary on its input vectors and produces output vectors to be the input of the next RBM.

Code

The network was built using tensorflow's SupervisedDBNClassification and two models of it were generated and trained afterwards to help in testing and comparing the accuracies. As for the first model, the hidden layers were set to 50, learning rate=0.05, activation function=relu. While for the second model, the hidden layers were set to 20, learning rate=0.08, activation function=sigmoid

Limitations

There was no output generated for the models, for there was an error in dbn.tensorflow. SupervisedDBNClassification and nolearn.dbn which are the most helpful and used imports in DBN classification, and the same error runs across all our laptops. However, as the error we got is that it requires higher GPUs, it could run on more advanced laptops

Comparison

Among the three implemented networks, MLP is the best one to classify this dataset.

References

[1] DeSantis, C. et al. International variation in female breast cancer incidence and mortality rates. Cancer epidemiology, biomarkers & prevention : a publication of the American Association for Cancer Research, cosponsored by the American Society of Preventive Oncology Available at: https://pubmed.ncbi.nlm.nih.gov/26359465/.

[2] B. Splane, "Differences between a malignant and benign tumor," *Verywell Health*. [Online]. Available: https://www.verywellhealth.com/what-does-malignant-and-benign-mean-514240.