Utilizing Machine Learning to Predict the Malignancy of a Breast Tumor

Megan Hu*

March 15th, 2024

Abstract

Machine learning is creating computational models from data that predict outcomes and facilitate decisions. Using past data, machine learning can help predict the label of whether a current patient will more or less likely have breast cancer given only the features. This paper describes the process of creating a model to test if a patient has breast cancer using a dataset with data on the radius, texture, compactness, etc., of 569 breast tumors. One hot encoding, binary classification, validation and training sets, neural networks, binary cross-entropy error, accuracy, and the testing of epochs and different combinations of nodes, hidden layers, and activation functions will be included in this paper.

1 Introduction

Machine learning is a subfield of artificial intelligence where machines imitate human intelligence learning from data and experience to create programs that predict. In computer systems, experience exists in form of data, and the main task of machine learning is to develop learning algorithms and build models from existing data [Zho87]. There, they will identify patterns to foresee the future. Today, machine learning is used in various applications, including verbal recognition, image recognition, and healthcare. Machine learning can develop a computational model that can later be used to predict the label of new examples where we only know the values of the features. Regarding breast cancer, this indicates that the model would predict whether the tumor is malignant from its features.

1.1 Background

Breast lumps can be caused by infection, trauma, fibroadenoma, cyst, fat necrosis, or fibrocystic breasts. Most are benign, non-cancerous cysts or tumors but have the potential to be cancerous. Breast cancer occurs when some breast cells

*Advised by Guillermo Goldsztein

begin to grow abnormally and these cells begin to divide more rapidly than healthy cells and continue to accumulate forming a mass or a lump. It is the second most common cancer in women in the United States as it is about 30% of all new female cancers each year and has a 2.6% chance of death [Sta]. As breast tumors are commonly found in women, it would be beneficial if machines could help doctors decide whether to treat the tumor and begin surgery or therapy before it worsens.

The dataset that will be used in this paper was published by the University of California Irvine and was obtained from kaggle.com [Brs], a data science and machine learning community with hundreds of thousands of datasets. The data is an example of supervised learning which is when the data has a set of examples and each example has features and a label, the target variable. It is a binary classification problem: the label is categorical with 2 possible categories. There are 569 examples with 357 benign and 212 malignant making it a relatively balanced dataset.

diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	texture_worst	perimeter_worst
M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...	17.33	184.60
M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...	23.41	158.80
M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...	25.53	152.50
M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...	26.50	98.87
M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...	16.67	152.20

Figure 1: The first 5 patients and 11 features and the label of the dataset used in this research paper

There are 30 features. There are two parts to each feature's name: the first part being either radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension and the second part being either mean, se, or worst. Appendix contains more details.

The following of this paper includes the steps towards the development of the model for predicting the malignancy of a breast tumor including data preprocessing, training and validation set, neural networks, and testing implemented with the software library Keras and dataset with 596 patients.

2 Methodology

2.1 Data Preprocessing

Sometimes in datasets, there are irrelevant features that we have to remove in order for an accurate model. For example, in a dataset used to predict stroke, there is a feature about the person's marital status, which is unrelated and hence should be removed. However, in this dataset we are using, all the features are related to the tumor and so none were removed except one. This feature included the data of the patient's id. The number of the id does not provide any information as to details of the tumor that would help predict if the tumor is malignant or not, thus was dropped from the dataset.

Another step in data preprocessing is converting words to numbers. In one hot encoding, categorical data variables such as letters and words are converted into numbers for the computer to understand. Using df.replace, in the column 'diagnosis,' B which stands for benign is replaced with 0, and M which stands for malignant is replaced with 1.

Before	After
B	0
M	1
M	1
B	0
M	1
B	0
B	0

Table 1: Example of before and after of one hot encoding

2.2 Binary Classification Problem

There are three different classifications of data: binary classification, multiclassification, and non-classification problems. Binary classification is where the label has two categorical - categorical meaning it is related to letters or words- possibilities, multiclassification is where the label has multiple categorical possibilities, and non-classification is where the label is numerical - numerical meaning it is related to numbers. Depending on the classification, different ways are used to create the model.

The model we are developing in this paper predicts the likelihood of a cancerous breast tumor based on the features. Each patient in this dataset either has a malignant tumor or a benign one making there only be two possible values of the label which makes it a binary classification problem. As in common practice, y is denoted as the prediction and output from the model and the number is always between 0 and 1. The example belongs to the category 1 or malignant if $\hat{y} < 0.5$ and the category 0 or benign if $\hat{y} < 0.5$.

2.3 Training and Validation Sets

The dataset is split into 2 sets randomly: the training set and the validation set. The training set accounts for 75% of the original dataset and the validation set is the remaining 25%. The training set helps develop the model and the validation set validates it which is how we can see the accuracy of the model. The model that was created with data from the training set is compared to the validation set and tells us the accuracy that we can use to find the best model.

2.4 Neural Networks

Neural networks are the collection of nodes that mimic the neurons in the human brain. It works for both classification and non-classification problems to model complex relationships between the output and input. There are three parts to a neural network: the input layer, the hidden layer(s), and the output layer. There can be multiple hidden layers depending on the dataset and every node in one layer is connected to every node in the next layer.

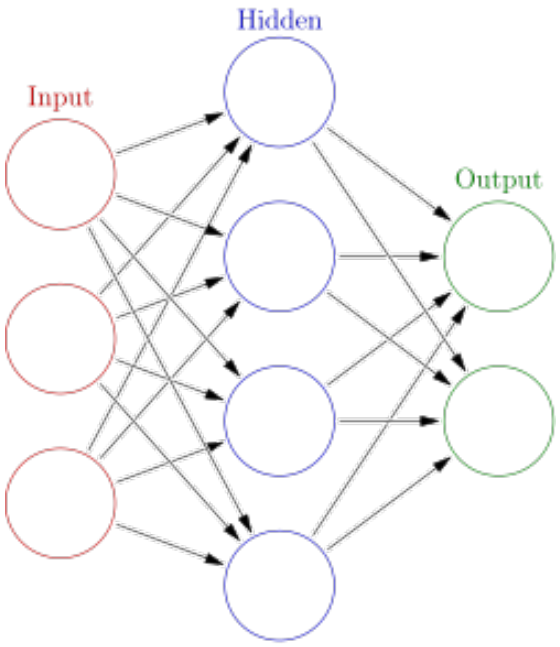

Figure 2: A simple example of a neural network

In a binary classification problem, there is only one node in the output layer and it must be a sigmoid activation function. The number of nodes in the input layer is equal to the number of features. The hidden layer(s) is the only thing you can change in a neural network. Changes include adding more hidden layers. more nodes to the hidden layers, and choosing different activation functions for the hidden layers. More nodes and layers are to fit more complicated functions such as one that fluctuates and change directions more. If the model fluctuates or changes directions more than the data needs, it is called overfitting which can produce a less accurate model. The opposite can happen as well with underfitting when the model does not fluctuate or change direction enough which can also produce a less accurate model. An example of overfitting and underfitting is shown below.

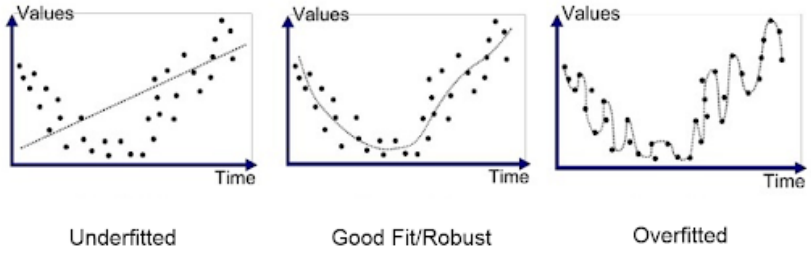

Figure 3: graphs of overfitted and underfitted model

2.5 Error and Accuracy

Binary cross entropy error is the error for binary classification problems and it compares each of the predictions ($\hat{y}_i$) to the actual label ($\hat{y}$) which can be either 0 or 1 and calculates the difference. We denote this error by J. Let y_1, y_2, y_3, ..., y_{32} be the labels, each being equal to 0 or 1. Let $\hat{y}_1$, $\hat{y}_2$, $\hat{y}_3$, ..., $\hat{y}_{32}$ be the predictions by the model of the labels.

$$J = -\frac{1}{n}\sum_{i=1}^{n} y_i \log(y_i) + (1 - y_i)\log(1 - y_i)$$

As stated previously, the validation set accuracy is the number that will determine the accuracy of the model, and the Binary cross entropy error will be used to calculate the error which is also known as loss. The smaller the binary cross entropy error, the better the predictions.

The performance of a machine learning model is evaluated by its accuracy which is calculated by dividing the number of correct predictions by the number of total predictions.

The accuracy is between 0 and 1 where when the accuracy is closer to 1, the more accurate or the more the predictions match up with the actual data. The opposite is true when the accuracy is closer to 0. The accuracy closest to 1 is the goal for a machine learning model.

2.6 Testing

Now, we test multiple different combinations of nodes and layers to find the model with the smallest error. Since our dataset is relatively small with only 569 examples when compared to datasets with tens of thousands of examples, many nodes and layers are not necessary.

First, we looked at a simple logistic regression which is a machine-learning technique used to create models for binary classification problems. This is where there are 0 hidden layers, only an input layer, and an output layer. The validation set accuracy ends up being 0.9580. This is already very good but more tests should be conducted to validate that this is the most accurate.

After several tests with different combinations of nodes, activation functions, and layers, the most accurate, at least in our tests, is a neural network with one hidden layer which holds 2 nodes with the sigmoid activation function.

```
model = 0
model = Sequential()
model.add(Dense(2, activation='sigmoid'))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', metrics = ['accuracy'])
model.fit(X_train_scaled,y_train,epochs=500,verbose=1, validation_data = (X_val_scaled,y_val))
```

Figure 4: code for the model

We then tested the model with one hidden layer and 2 nodes with the sigmoid activation function with multiple different epochs of 20, 100, and 250. Epochs are the number of iterations of model tuning.

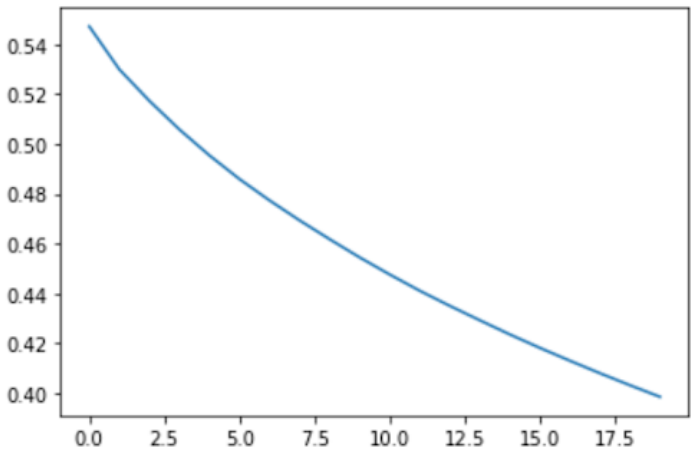

Figure 5: graph of the error of the model with epochs = 20

Above is the graph where epochs = 20. As shown, the graph does not flatten out which indicates that there are not enough epochs.

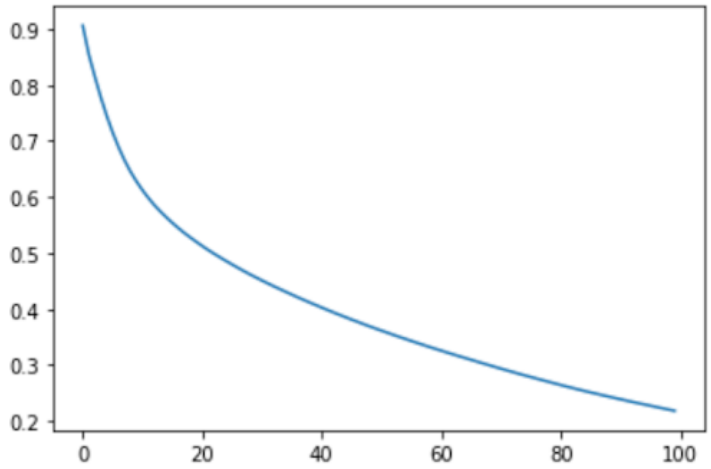

Figure 6: Graph of the error of the model with epochs = 100

Again, we tested the model with 100 epochs, and the graph clearly flattens a little more compared to the graph with epochs = 20.

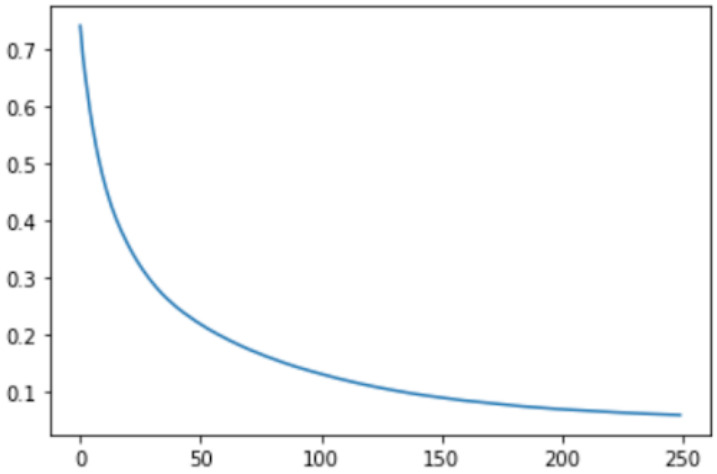

Figure 7: graph of the error of the model with epochs = 250

Finally, with epochs = 250, the graph flattens out and the accuracy is 0.9720 and the loss is 0.0855. To test the epochs a little more, epochs = 300 is tested to see if the graph needs to flatten out more. This only resulted in a lower accuracy of 0.9580 and a greater loss of 0.1005. This is an example of overfitting. The best-performing model had one hidden layer with 2 nodes and sigmoid activation

function. Testing with epochs showed that 250 epochs resulted in the highest accuracy.

3 Conclusion

Utilizing machine learning, we have created a model with a 0.9720 accuracy that can help decide the likelihood of cancer in a new breast tumor. We have trained a model so that it can detect breast cancer in other patients, not in the dataset we trained it with. Data preprocessing, dividing data into training and validation sets, and testing multiple models to find the optimal one is the process of creating machine learning models. However, it must be noted that the dataset of this model is too small to ultimately decide whether or not to start treatment and take action that can affect the patient but is a good practice. With more data and research, machine learning techniques can be applied to the medical field to aid doctors with diagnosing certain cancers and diseases which can provide a more accurate and definite diagnosis, thus saving more patients. Machine learning is a powerful tool where we can learn from the past to help the future.

References

[Brs] Breast cancer wisconsin data. `https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data`.

[Sta] Key statistics for breast cancer. `www.cancer.org/cancer/breast-cancer/about/how-common-is-breast-cancer.html`. No date.

[Zho87] Zhi-hua Zhou. *Machine Learning*. Springer Verlag, Singapore, 1987.

4 Appendix

	mean	se	worst
radius	radius_mean: mean of distances from center to points on the perimeter	radius_se: standard error for the mean of distances from center to points on the perimeter	radius_worst: "worst" or largest mean value for mean of distances from center to points on the perimeter
texture	texture_mean: standard deviation of gray-scale values	texture_se: standard error for standard deviation of gray-scale values	texture_worst: "worst" or largest mean value for standard deviation of gray-scale values
perimeter	perimeter_mean: mean size of the core tumor	perimeter_se	perimeter_worst
area	area_mean	area_se	area_worst
smoothness	smoothness_mean: mean of local variation in radius lengths	smoothness_se: standard error for local variation in radius lengths	smoothness_worst: "worst" or largest mean value for local variation in radius lengths
compactness	compactness_mean: mean of perimeterˆ2 / area - 1.0	compactness_se: standard error for perimeterˆ2 / area - 1.0	compactness_worst: "worst" or largest mean value for perimeterˆ2 / area - 1.0
concavity	concavity_mean: mean of severity of concave portions of the contour	concavity_se: standard error for severity of concave portions of the contour	concavity_worst: "worst" or largest mean value for severity of concave portions of the contour
concavity_points	concave_points_mean: mean for number of concave portions of the contour	concave_points_se: standard error for number of concave portions of the contour	concave_points_worst: "worst" or largest mean value for number of concave portions of the contour
symmetry	symmetry_mean	symmetry_se	symmetry_worst
fractal dimension	fractal_dimension_mean: mean for "coastline approximation" - 1	fractal_dimension_se: standard error for "coastline approximation" - 1	fractal_dimension_worst: "worst" or largest mean value for "coastline approximation" - 1