CS230

Breast Cancer Fine-Needle Classification

Lea-Tereza Tenekedjieva
Stanford University
lten@stanford.edu

Abstract

The purpose of this project is to design an algorithm that improves breast mass diagnosis based on features derived from a digital image of fine-needle aspirates. Fine-needle aspiration is a minimally invasive diagnostic procedure for examining lumps and the benefit of such an algorithm would be allow for a reliable diagnosis of breast cancer without the need for an invasive surgical biopsy. Three models were implemented and compared. The baseline model performed exceptionally well and we aimed to determine whether performance can be significantly improved by employing a more complex model. The result was that the more rudimentary models (baseline - logistic regression and two layer network with tanh activation) achieved better performance.

1 Introduction

Breast cancer is the most common type of cancer worldwide, contributing to 12.3% of total number of new cancer cases in 2018, according to American Institute for cancer research [1]. Around 12 % of US women will be diagnosed with breast cancer in their lifetime, and approximately 11% of those diagnosed will not survive past 5 years of the diagnosis. A breast cancer patient's chance of recovery is highly dependent on early detection and accurate diagnosis of the disease.[2] The goal of the current project is to develop an effective diagnostic tool for breast cancer that extracts feature from digital images images with the goal of discriminating between benign or malignant tumors.

Currently there are 3 ways to diagnose breast cancer: mammography, with reported sensitivity between 68% and 79% [3], Fine-Needle Aspiration (FNA) with visual interpretation from the doctor with 65 to 98% sensitivity [4], and surgical biopsy with almost 100% sensitivity. Those statistics demonstrate a problem with the current treatment options: mammography is not sensitive enough, FNA's sensitivity varies widely, and surgical biopsy, while accurate, is very invasive procedure [5]. The current project focuses on designing an algorithm that improves breast mass diagnosis based on features derived from a digital image of fine-needle aspirates. Fine-needle aspiration is a minimally invasive diagnostic procedure for examining lumps and masses where a hollow needle is inserted directly into the mass with the goal of sampling cells [6]. The benefit of an algorithm which can classify malignant or benign cancers with high sensitivity would allow for a reliable diagnosis of breast cancer without the need for an invasive surgical biopsy.

The input of our algorithm is 30 extracted features from fine needle aspirate (FNA) images of a breast mass. Three different models were used (logistic regression, two-layer neural network with tanh activation function and two-layer neural network with RELU activation) to output a predicted cancer type (benign or malignant).

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

2 Related work

There are a few papers which include the same or similar breast cancer dataset, and who follow the general model of comparing the performance of several neural networks. Salama et al found that using fusion of MLP, J48, SMO and IBK is superior to the other classifiers [7]. George et al found that probabilistic neural networks were optimal over multilayer perceptron (MLP) using back-propagation algorithm, learning vector quantization (LVQ) and support vector machine (SVM)[8]. A Back Propagation Neural Network (BPPN) and radial basis neural networks (RBFN)was employed by Kaymak et al., although they only achieved 59.0% and 70.4% accuracies respectively [9].

3 Dataset and Features

The projects uses *Breast Cancer Wisconsin (Diagnostic) Data Set* obtained from UCI Marchine Learning Repository [10]. Dataset consists of features extracted from a digitized image of fine needle aspirate (FNA) of breast lumps. The first column is an ID number, second represents the diagnosis (malignant or benign) followed by ten real-valued features for each cell nucleus [10]:

- radius (mean of distances from center to points on the perimeter)
- texture (standard deviation of gray-scale values)
- perimeter
- area
- smoothness (local variation in radius lengths)
- compactness (perimeter2 / area - 1.0)
- concavity (severity of concave portions of the contour)
- concave points (number of concave portions of the contour)
- symmetry and fractal dimension ("coastline approximation" - 1).

For each feature, the mean, standard error and largest of these features are computed, resulting in 30 features total. For example, feature 3, feature 13 and feature 23 are the mean, standard error and worst radius respectively [10]. The class distribution is 357 benign and 212 malignant tumors. Because of the small size of the data, we split the dataset into 60% training examples (343), 20% validation examples(113), 20% testing examples (113). We performed basic preprocessing of the data, and carried out normalization on the input features.

4 Methods

The extracted features were inputs for three different models, whose performance was compared.

4.1 Model 1: Logistic Regression

As a baseline, logistic regression was implemented. The model has input layer of size 30 (because we have 30 input features for each example), to which we first apply a linear function, and then the activation function sigmoid, as shown mathematically below [12]. Note that z_i denotes the input of layer i and a_i is the output of layer i after applying the activation function, in our case the sigmoid function. W_1, b_1, W_2, b_2 are the parameters that are model is attempting to learn in the network [11]. The architecture of the network is visually depicted in *Figure 1* and can summarized as
Input → Linear → Sigmoid→ Output.

$$z^{(i)} = w^T x^{(i)} + b$$

$$y^{(i)} = a^{(i)} = \sigma(z^{(i)})$$

$$J = -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}log(a^{(i))}) + (1-y^{(i)})log(1-a^{(i)})\right)$$

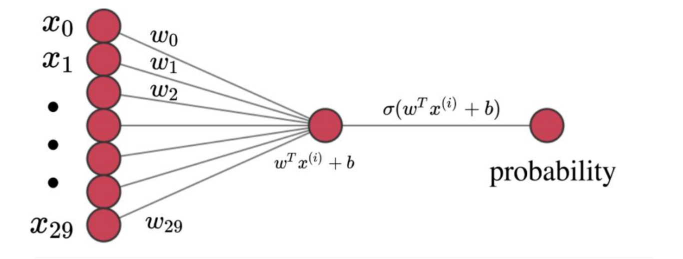

Figure 1. *Architecture of Logistic Regression*

4.2 Model 2: Two-Layer Neural Network with tanh Activation Function

The model is a two-layer network with input layer of size 30, one hidden layer of size 4. The input features undergo a linear function, then a tanh activation function, then another linear function and finally a sigmoid activation function to obtain the output layer, as shown mathematically below [12]. The architecture of the network is visually displayed in *Figure2* and can be summarized as ***Input→Linear → tanh→Linear→Sigmoid→ Output***.

$$z^{[1](i)} = w^{[1]}x^{(i)} + b^{[i]}$$
$$a^{[1](i)} = tanh(z^{[1](i)})$$
$$z^{[2](i)} = w^{[2]}a^{[i](i)} + b^{[2]}$$
$$y^{(i)} = a^{[2](i)} = \sigma(z^{[2](i)})$$
$$J = -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}log(a^{[2](i)}) + (1-y^{(i)})log(1-a^{[2](i)})\right)$$

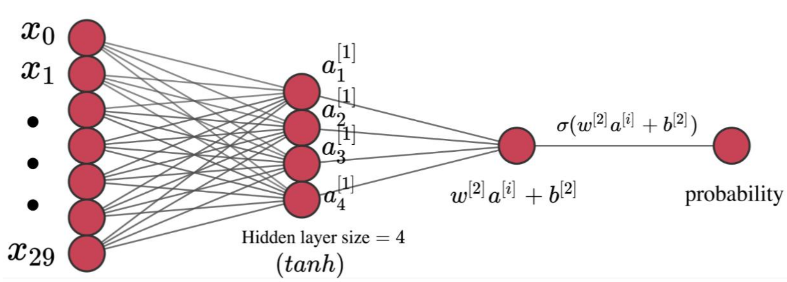

Figure 2. *Architecture of two-layer neural network with tanh activation function.*

4.3 Model 3: Two-layer Neural Network with RELU Activation Function

The model is a two-layer network with one hidden layer of size 7. The input features undergo a linear function, then a RELU activation function, then another linear function and finally a sigmoid activation function to obtain the output layer, as depicted mathematically[12]. The architecture of the network is visually displayed in *Figure3* and can be summarized as ***Input→Linear → tanh→Linear→Sigmoid→ Output***.

$$z^{[1](i)} = w^{[1]}x^{(i)} + b^{[i]}$$
$$a^{[1](i)} = \mathrm{RELU}(z^{[1](i)})$$
$$z^{[2](i)} = w^{[2]}a^{[i](i)} + b^{[2]}$$
$$y^{(i)} = a^{[2](i)} = \sigma(z^{[2](i)})$$
$$J = -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}log(a^{[2](i)}) + (1-y^{(i)})log(1-a^{[2](i)})\right)$$

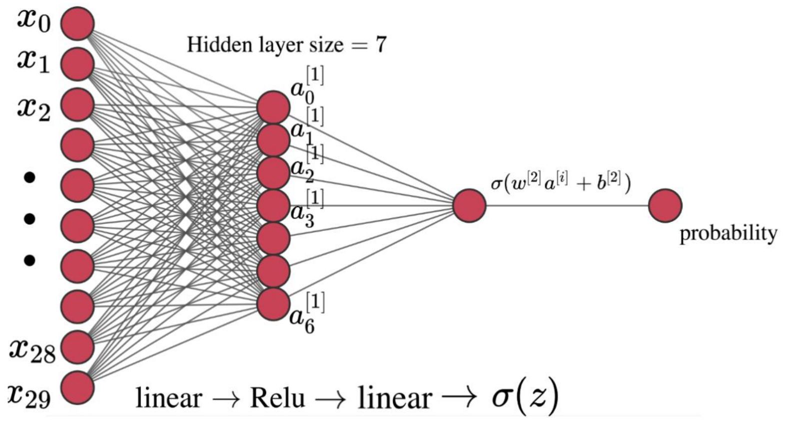

Figure 3. *Architecture of two-layer neural network with RELU activation function.*

5 Results and Discussion

We compared the three models on accuracy, sensitivity (recall) and specificity. Sensitivity, also known as recall, is defined as ... Specificity is ....

A summary of the results is depicted in Figure 4. Logistic Regression had a high performance, with 95% accuracy, 100% sensitivity and close to 93% specificity. Given those results, we wanted to determine if a more complex model would be able to improve the performance by increasing the specificity. Our second model, the two-layer NN with tanh activation, did perform better in sensitivity, achieving 98% but significantly decreased to 76% in sensitivity. That result is particularly detrimental in medical settings because it means we increase the cases of disease missed. Our third and most complex model with RELU activation fun and 7 hidden nodes achieves perfect sensitivity of 100% but exceptionally bad accuracy and specificity - barely 19%. The Reciever Operator Curves for Model 2 and Model 3 can be are shown on Figure 5.

	LogReg		**tanh**	**RELU**	
	Train	Test	Test	Train	Dev
Accuracy	0.8260	0.9534	0.936813	0.9478	0.3804
Sensitivity	0.9611	1.0	0.7619	0.8618	1.0
Specificity	0.7459	0.9259	0.9859	0.9917	0.19718

Figure 4. *Summary of results in the primary metrics used to compare the three models.*

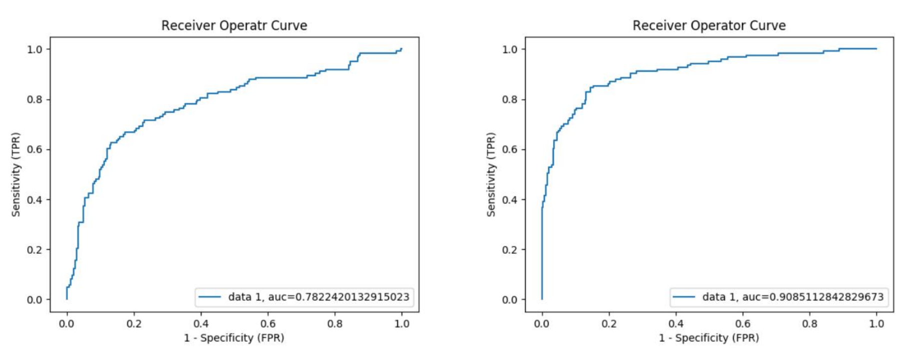

Figure 5.*Receiver operating characteristic curves for Model 2 - NN with tanh activation function (left) and Model 3 - NN with RELU activation function (right).*

Further, we explored the importance of different features by comparing their assigned weights in our first two models. As, you can see on *Figure 6*, both logistic regression and network with tanh activation are learning the same features. We were able to identify that the most significant features are perimeter, area and radius. Interestingly, we can see that almost no learning at all occurs on the rest of the features. This finding could potentially have significant diagnostic benefit, as it provides evidence for breast mass features that do not have significant influence on the type of cancer.

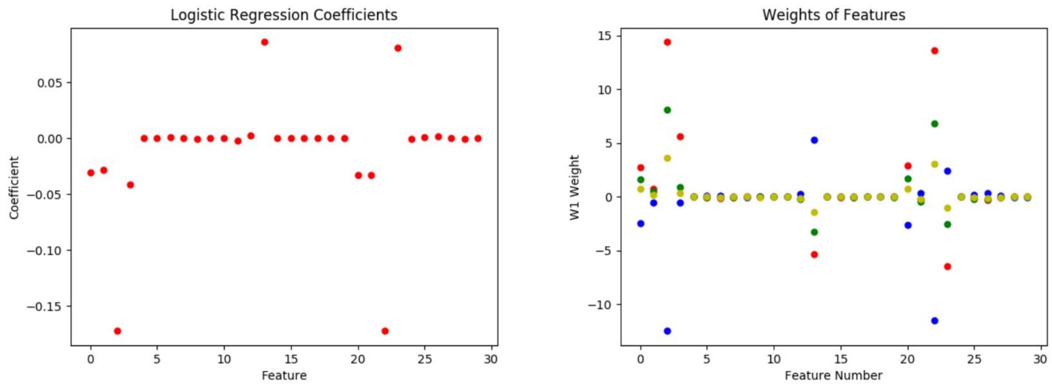

Figure 6. *Weights of features in: Logistic regression model (left) and Model 2 - NN with tanh activation (right).*

We found that our third and most complex model had the poorest performance in terms of specificity and accuracy. We suspect that it is because the model does not generalize well to new data and is likely overfitting. *Figure 7* shows the difficulty that was presented with minimizing the cost function. To tackle the issue, we attempted various techniques like l2 regularization, tuning lambda and alpha, learning rate decay, number of iterations and early stopping. However, we were not able to significantly improve performance and further work is necessary to determine the right balance of hyperparameters.

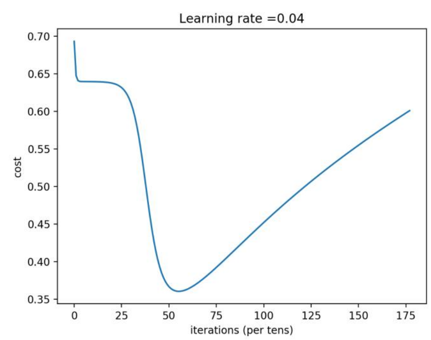

Figure 7. *Cost function of Model 3 – NN with RELU activation. Difficulties minimizing the cost function.*

6 Conclusion/Future Work

The most important take away from our results is that simpler, easier to implement models like logistic regression and two-layer neural network performed significantly better. This finding is particularly useful in medical setting because it shows that more rudimentary models can get a very good diagnostic prediction, while also being accessible to implement by physicians.

In addition to optimizing Model 3 further, further work would include measuring and comparing the performance of a Recurrent Neural Network. It would be very useful to explore whether an RNN would have a high performance on the somewhat-recurring nature of the dataset.

References

[1] Bray F, Ferlay J, Soerjomataram I, Siegel RL, Torre LA, Jemal A. Global Cancer Statistics 2018: *GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries.* CA Cancer J Clin, in press. The online GLOBOCAN 2018 database is accessible at http://gco.iarc.fr/, as part of IARC's Global Cancer Observatory.

[2] George, Yasmeen, Elbagoury, Bassant Zayed, Hala Roushdy, Mohamed. (2012). *Breast Fine Needle Tumor Classification using Neural Networks*. IJCSI International Journal of Computer Science. 9. 247-256.

[3] S.W. Fletcher, W.Black, R.Harris, B.K. Rimer, S.Shapiro. *Report of the international workshop of screening for breast cancer.* Journal of the National cancer Institute, 85:1644-1656,1993

[4] R.W.M. Giard and J.Hermans. *The value of aspiration cytologic examination of the breast.* A statistical review of the medical literature. Cancer, 69:2104-2110, 1992

[5] O.L. Mangasarian, W.Nick Street, W.H.Wolberg. Breast Cancer Diagnosis and Prognosis via Linear Programming. Mathematical programming technical report 94-10. 1994

[6] Fine needle aspiration (FNA) biopsy information. (2017, August 17). Retrieved from https://www.myvmc.com/investigations/fine-needle-aspiration-biopsy-fna/

[7] Salama, G.I. Abdelhalim, mohamed b Zeid, Magdy. (2012). *Breast Cancer Diagnosis on Three Different Datasets using Multi-classifiers.* International Journal of Computer and Information Technology. 1. 36-43.

[8] George, Yasmeen & Elbagoury, Bassant Zayed, Hala Roushdy, Mohamed. (2012). *Breast Fine Needle Tumor Classification using Neural Networks.* IJCSI International Journal of Computer Science. 9. 247-256.

[9] Kaymak, Sertan & Helwan, Abdulkader Uzun Ozsahin, Dilber. (2017). *Breast cancer image classification using artificial neural networks.* Procedia Computer Science. 120. 126-131. 10.1016/j.procs.2017.11.219.

[10] UCI Machine Learning Repository. *Diagnostic Wisconsin Breast Cancer Database.* Retrieved from: `https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)` and from `https://www.kaggle.com/uciml/breast-cancer-wisconsin-data`

[11] http://www.wildml.com/2015/09/implementing-a-neural-network-from-scratch/

[12] DeepLearning.ia. *Course: Neural Networks and Deep Learning*, as part of CS230: Deep Learning course materials.

Resources used for writing code:

[13] DeepLearning.ia.*Programming Assignment: Logistic Regression with a Neural Network mindset*Retrieved from: https://www.coursera.org/learn/neural-networks-deep-learning/programming/XaIWT/logistic-regression-with-a-neural-network-mindset/submission

[14] DeepLearning.ia. *Programming Assignment:Planar Data Classification with a Hidden Layer*Retrieved from: https://www.coursera.org/learn/neural-networks-deep-learning/programming/wRuwL/planar-data-classification-with-a-hidden-layer

[15] DeepLearning.ia. *Programming Assignment: Building your Deep Neural Network: Step by Step*Retrievef from: https://www.coursera.org/learn/neural-networks-deep-learning/notebook/lSYZM/building-your-deep-neural-network-step-by-step

[16] DeepLearning.ia. *Programming Assignment: Deep Neural Network - Application* Retrieved from: Retrievef from: https://www.coursera.org/learn/neural-networks-deep-learning/notebook/24mxX/deep-neural-network-application

Tool for generating diagrams of neural network architectures:

[17] Alex Lenail: NN-SVG Retrieved from: http://alexlenail.me/NN-SVG/LeNet.html

Other Resources

Personal communication with Tynan Challenor. Special thanks to my friend who tirelessly supported and advised me throughout the process.