Tumor Classification

Brianna Grissom

March 2025

1 Motivation & Problem Statement

1.1 Motivation

Breast cancer, where abnormal breast cells grow and form tumors, is a prevalent type of cancer affecting many women around the world. In fact, 670,000 people have died from breast cancer in 2022. Thus, it is important to know whether tumors are benign or malignant to determine if an individual should receive treatment.

As a result, the aim of this project is to predict whether tumors are benign or malignant using a dataset of 569 observations, where 357 tumors are benign and 212 tumors are malignant. To do this, I will create a machine learning model to classify tumors as benign or malignant using 31 continuous-valued feature variables. I hope to discover whether benign and malignant tumors can be sufficiently classified using all 31 feature variables, which set of three feature variables is best able to classify tumors, and how quickly the training algorithm is able to converge. The binary classification model will be a single layer perceptron, which generates a hyperplane to linearly separate the data points as benign or malignant in $p-$dimensional space. The single layer perceptron is the simplest form of a neural network and assumes that the data are linearly separable.

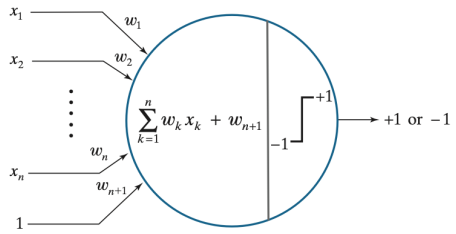


Figure 1: The single-layer perceptron.

1.2 Problem Statement

Let $\mathbf{X} \in \mathbb{R}^{n \times (p+1)}$ denote the predictor dataset containing n observations and p predictor variables. Denote the corresponding classification labels as $\mathbf{y} \in \mathbb{R}^n$. Write

$$\mathbf{X} = \begin{bmatrix} \mathbf{x_1}^T \\ \mathbf{x_2}^T \\ \vdots \\ \mathbf{x_n}^T \end{bmatrix},$$

where each row vector $\mathbf{x_i}^T = [x_{i1}, ..., x_{ip}, 1]$ is an observation with a 1 appended to the end. Write

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix},$$

where $y_i \in \{-1, 1\}\ \forall i = 1, ..., n$. Label -1 indicates benign tumors and label 1 indicates malignant tumors. The goal is to find the hyperplane

$$w_1 x_1 + w_2 x_2 + ... + w_p x_p + b = \mathbf{w}^T \mathbf{x} = 0,$$

where b is the bias term, in $\mathbb{R}^p$ that maximizes the classification precision by best linearly separating the data points as benign or malignant. Define the classifier (activation) function as

$$C_{\mathbf{w}}(\mathbf{x}) = \begin{cases} -1, & \text{if } \mathbf{w}^T \mathbf{x} < 0 \\ 1, & \text{if } \mathbf{w}^T \mathbf{x} \geq 0 \end{cases}.$$

$C_{\mathbf{w}}(\mathbf{x})$ classifies observations as benign or malignant. Thus, benign observations will lie below the hyperplane and malignant observations will lie on or above the hyperplane. To generate the ideal hyperplane, we seek $\mathbf{w}^\star \in \mathbb{R}^p$ such that

$$\mathbf{w}^\star = arg\ max_{\mathbf{w}} \frac{\sum_{i=1}^{n} I(C_{\mathbf{w}}(\mathbf{x_i}) = y_i)}{n},$$

where I is the indicator function. This is the $\mathbf{w}^\star$ that has the highest classification accuracy. This result is important because $\mathbf{w}^\star$ will be the coefficient vector that generates the hyperplane $(\mathbf{w}^\star)^T \mathbf{x} = 0$ that best separates data points as benign and malignant.

2 Methods

The optimal hyperplane will be generated by performing the gradient descent algorithm on the mean squared error (MSE) loss function, denoted $f(\mathbf{w})$, to minimize the MSE between $\mathbf{w}^T \mathbf{x}$ and the observed label of -1 or 1 for observation $\mathbf{x}$.

The overall MSE loss function will be an average of the loss function

$$f_i(\mathbf{w}) = (\mathbf{w}^T\mathbf{x}_i - y_i)^2,$$

for each observation $(\mathbf{x_i}^T, y_i)$. Thus, the loss function that will be minimzed is

$$f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{w}^T\mathbf{x}_i - y_i)^2.$$

Prior to training the model, the data matrix will be normalized to promote faster convergence of the algorithm. It has been shown that for minimizing a smooth objective function, normalizing the data will reduce the condition number of the Hessian, which will speed up the algorithm. To start, the iterative gradient descent algorithm picks an initial guess at the optimal weight value of $\mathbf{w}^{(0)}$. Then, it sets the next weight value, $\mathbf{w}^{(k+1)}$, to be

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \alpha_k \nabla f(\mathbf{w}^{(k)}).$$

The direction vector is $-\nabla f(\mathbf{w}^{(k)})$ because by using the Cauchy-Schwarz inequality, it can be shown that the direction of maximum decrease for a function f at point $\mathbf{w}^{(k)}$ is $-\nabla f(\mathbf{w}^{(k)})$.
Next, we need to select the learning rate parameter α_k. The steepest descent method of

$$\alpha_k = arg\, min_{\alpha \geq 0} f(\mathbf{w}^{(k)} - \alpha \nabla f(\mathbf{w}^{(k)}))$$

will to used to descend the loss function in as few amount of steps as possible. Notice that $f(\mathbf{w})$ can be written in the quadratic form of $f(\mathbf{w}) = \mathbf{w}^T\mathbf{Q}\mathbf{w} - \mathbf{b}^T\mathbf{w} + c$, where $\mathbf{Q} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T$ is a symmetric matrix. Using the first order necessary condition for minimizers, it can be shown that

$$\alpha_k = \frac{\nabla f(\mathbf{w}^{(k)})^T \nabla f(\mathbf{w}^{(k)})}{\nabla f(\mathbf{w}^{(k)})^T \mathbf{Q} \nabla f(\mathbf{w}^{(k)})}$$

provides that $f(\mathbf{w}^{(k)})$ is a quadratic function.
With the above specifications, the algorithm will descend on $f(\mathbf{w})$ in a way that $f(\mathbf{w}^{(k)})$ is reduced each iteration, hence always reducing loss. The algorithm is repeated for a total of K iterations.

In summary, I will

1. Apply normalization to the input data, choose initial weight $\mathbf{w}^{(0)}$.
2. For iteration k, $k = 1, 2, ..., K$:
   (a) For each observation $\mathbf{x}_i$:
      i. Compute $(\mathbf{w^{(k)}})^T\mathbf{x}_i$.
      ii. If $(\mathbf{w}^{(k)})^T\mathbf{x}_i < 0$, set $C_{\mathbf{w}^{(k)}}(\mathbf{x}_i) = -1$, else set $C_{\mathbf{w}^{(k)}}(\mathbf{x}_i) = 1$.
      iii. Compute $f_i(\mathbf{w}) = (\mathbf{w}^T\mathbf{x}_i - y_i)^2$

iv. Compute $\nabla f_i(\mathbf{w}^{(k)}) = -((\mathbf{w}^{(k)})^T - y_i)\mathbf{x}_i$

(b) Compute accuracy $= \frac{\sum_{i=1}^{n} I(C_{\mathbf{w}}(\mathbf{x_i})=y_i)}{n}$

(c) Compute gradient vector $\nabla f(\mathbf{w}^{(k)}) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\mathbf{w}^{(k)})$

(d) Compute $\alpha_k = \frac{\nabla f(\mathbf{w}^{(k)})^T \nabla f(\mathbf{w}^{(k)})}{\nabla f(\mathbf{w}^{(k)})^T \mathbf{Q} \nabla f(\mathbf{w}^{(k)})}$

(e) Set $\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \alpha_k \nabla f(\mathbf{w}^{(k)})$.

3. Pick $\mathbf{w}^\star = arg_{\mathbf{w}}\, max \frac{\sum_{i=1}^{n} I(C_{\mathbf{w}}(\mathbf{x_i})=y_i)}{n}$

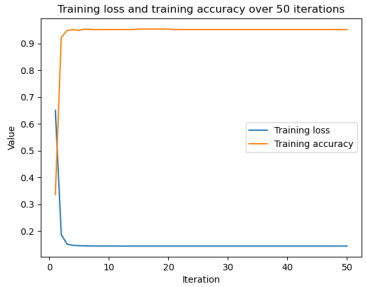
3 Results

Using the 31 predictor variables, the algorithm performed sufficiently well. The $\mathbf{w}^\star$ chosen using the above criteria achieved a testing and training accuracy of 94.7% and 95.4%, respectively.

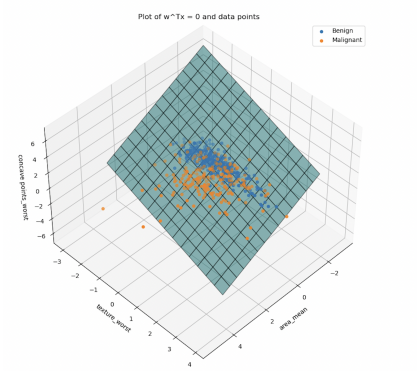
The combination of three predictor variables best able to classify tumors were "area_mean", "texture_worst", and "concave points_worst", with a testing and training accuracy of 96.5% and 92.3%, respectively. This result was surprising, as using significantly fewer predictor variables paradoxically increased the testing accuracy. Additionally, the algorithm converged very quickly; the loss and accuracy values flatlined after about 3 iterations. Further, the plot on the right indicates that the data points are linearly seperable when these 3 predictor variables are used.

Also, since the training loss decreased after every iteration, albeit by a small amount, this shows that the results agree with the theory of steepest descent in that the image of the loss function is reduced after iteration.

In conclusion, the results answered my questions well: tumor data are linear separable and hence able to be classified with high accuracy as benign or malignant via a hyperplane in p-dimensional space. Additionally, the classifications can be performed with few feature variables and the algorithm is able to converge quickly.



(a) Loss and accuracy



(b) Plot of hyperplane and datapoints