SMALL CELL LUNG DIAGNOSIS THROUGH CHEST X-RAY IMAGES USING ARTIFICIAL NEURAL NETWORKS

Usman Mahmood Khan

Zain Kabir

Fahad Fareed

Signals & Systems Semester Project

Table of Contents

SMALL CELL LUNG DIAGNOSIS THROUGH CHEST X-RAY IMAGES USING ARTIFICIAL NEURAL NETWORKS 1

Abstract 3

Introduction: 3

Background: 3

Related Work: 3

Project Model: 4

Data Collection: 4

Image Sampling: 4

Neural Network Structure: 4

Loading of Weights: 4

Feed-forward Mechanism: 5

Error Back-Propagation: 5

Regularization: 6

Results and Analysis: 6

Future Work: 7

1.0: Abstract

The project diagnoses small cell lung cancer in patients using their chest X-ray images. Currently X-rays are analyzed manually by relatively experienced doctors who identify high and low-risk patients. This process is, however, tedious and consumes considerable resources. The current project seeks to automate this process by automatically developing patterns in the images and learning about them through artificial neural networks.

2.0: Introduction:

2.1: Background:

Digital radiography is a recent X-ray imaging technique, where digital X-ray sensors are used instead of traditional photographic films. DX (Digital Radiography) is a superior technique in several ways: 1) It is time efficient and provides relatively quick results, 2) It bypasses chemical processing and uses less radiation than a traditional X-ray machine requires, and 3) It allows the images to be digitally transferred and enhanced. Digital radiography, therefore, is a promising technology for several machine learning algorithms.

The idea behind the project is to develop a model that inputs digital X-ray images from patients and indicates to them if they have malignant, benign or no tumor. This process is currently carried out manually and has several shortcomings, 1) Hospitals and clinics have to spend considerable resources to manually analyze all the X-ray images, 2) A considerable experience is required to be able to accurately assess those images, and 3) The results are not always accurate and sometimes, only a top surgeon or specialist is able to correctly analyze the X-ray image. The proposed model reduces such dependencies and, therefore, offers a great alternative to current practice.

2.2: Related Work:

In the past two decades, the advent of machine learning techniques has opened new gates for computer aided diagnosis. Several complex models have been built or are being built which predict certain conditions based on a set of patient's parameters. These works have either been done on expensive scanning technologies such as CT scans, ECG, EEG or have low accuracy. Therefore, the aims of the this project are to, 1) develop an efficient model for Digital Radiography images that are easily available to the patients, and 2) Enhance the accuracy of similar models developed previously.

3.0: Project Model:

3.1: Data Collection:

The data was collected from Cancer Image Archive from the following link: https://public.cancerimagingarchive.net

The data contain chest X-ray images of around 200 patients and a separate spreadsheet that contains information of their tumors. The tumors are classified into four categories:

0: Unknown

1: Benign or non-malignant disease

2: Malignant, primary lung cancer

3: Malignant Metastatic

For the current project, 14 out of 200 images were chosen that identified small cell lung cancer in the patients. Out of 14, 12 images were used for training the model and 2 for testing it.

3.2: Image Sampling:

A sample image (obtained from cancer image archive) has 1951*2000 samples. To reduce the complexity of training for the current model, sampling was performed to take a limited number of samples only and reduce the time of operation.

400 pixels were uniformly taken using the sampling techniques.

3.3: Neural Network Structure:

The neural network structure consisted of 1 input layer, 1 hidden layer and 1 output layer. The hidden layer contained 25 neuron units. It should, however, be noted that increasing the number of hidden layers would increase the accuracy of obtained results and models with increased number of hidden layers can be implemented in future.

3.4: Loading of Weights:

The next step involves loading weights in the model. Initially, the weights are randomly initialized. Since, there are 400 neurons in inputs layer and 25 neurons in hidden layer, the size of *weight matrix* from input layer to hidden layer is 25 * 401 (1 additional matrix element is for input bias unit).

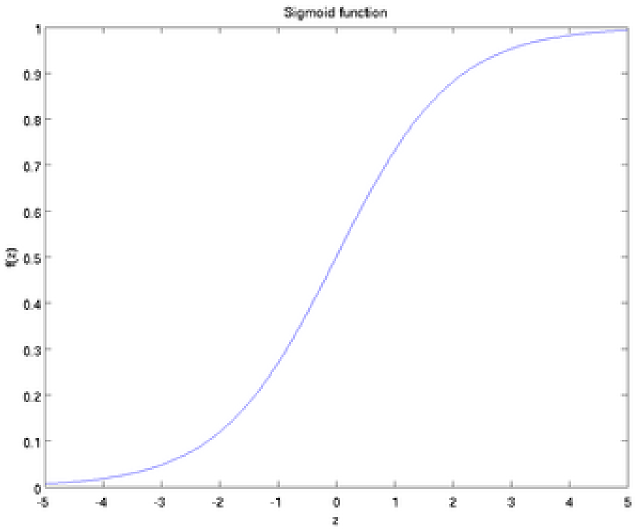
Similarly, the size of *weight matrix* from hidden layer to output layer is 3 * 25 (3 being the number of neurons in output layer).

3.5: Feed-forward Mechanism:

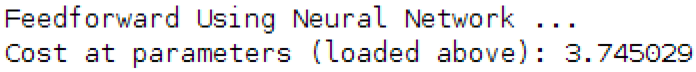
After the weights have been loaded, the output of the network is calculated through a feed forward mechanism using the random weights specified. The output through feed-forward mechanism is then compared against the actual output.

At each neuron level, a sigmoid function is applied to get an output between 0 & 1. The sigmoid function is defined as follows:

$$H = 1/1 + e^{-\theta}$$



Using feed forward the cost calculated was as follows:



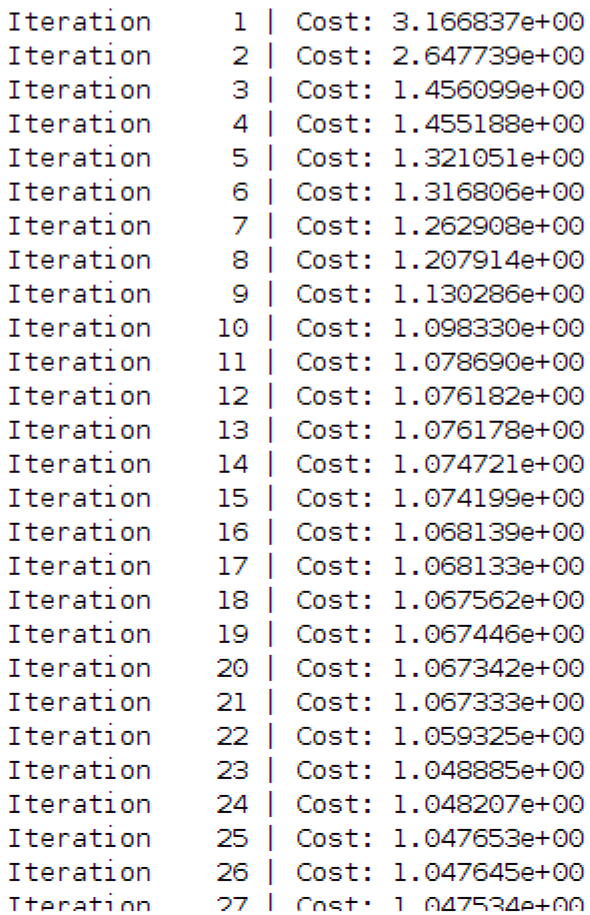
```
Feedforward Using Neural Network ...
Cost at parameters (loaded above): 3.745029
```

3.6: Error Back-Propagation:

Once the error has been calculated, it is back propagated to evaluate which neuron calculated most to the error. Weights are then adjusted using the gradient descent method in such a manner that the errors are minimized. This is a self correcting phenomenon.

This is done through gradient descent which identifies the path that should be adopted to reach the minimum point. For the current project, the program was restricted to 50 iterations. However, the more the iterations, the better the results in terms of minimization of cost.

The figure below shows the cost after each iteration:



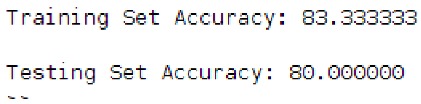
```
Iteration     1 | Cost: 3.166837e+00
Iteration     2 | Cost: 2.647739e+00
Iteration     3 | Cost: 1.456099e+00
Iteration     4 | Cost: 1.455188e+00
Iteration     5 | Cost: 1.321051e+00
Iteration     6 | Cost: 1.316806e+00
Iteration     7 | Cost: 1.262908e+00
Iteration     8 | Cost: 1.207914e+00
Iteration     9 | Cost: 1.130286e+00
Iteration    10 | Cost: 1.098330e+00
Iteration    11 | Cost: 1.078690e+00
Iteration    12 | Cost: 1.076182e+00
Iteration    13 | Cost: 1.076178e+00
Iteration    14 | Cost: 1.074721e+00
Iteration    15 | Cost: 1.074199e+00
Iteration    16 | Cost: 1.068139e+00
Iteration    17 | Cost: 1.068133e+00
Iteration    18 | Cost: 1.067562e+00
Iteration    19 | Cost: 1.067446e+00
Iteration    20 | Cost: 1.067342e+00
Iteration    21 | Cost: 1.067333e+00
Iteration    22 | Cost: 1.059325e+00
Iteration    23 | Cost: 1.048885e+00
Iteration    24 | Cost: 1.048207e+00
Iteration    25 | Cost: 1.047653e+00
Iteration    26 | Cost: 1.047645e+00
Iteration    27 | Cost: 1.047534e+00
```

3.7: Regularization:

To avoid over fitting and ensure that the algorithm not only does well on the training data but also on the test data, regularization was performed.

4.0: Results and Analysis:

After the calculation of optimal weights, the model was run on the training data to test the accuracy. The following were the results from training and testing data:



```
Training Set Accuracy: 83.333333

Testing Set Accuracy: 80.000000
```

The testing set accuracy is not a very good parameter since the test data contained only 2 images. The model can only be validated by providing a great variety of data.

5.0: Future Work:

The future work should focus on:

1- Training the model with a great variety of images
2- Increasing the sampling from just 400 pixels (in the current project) to well over 10,000 pixels
3- Testing the results with different thresholds
4- Using more advanced and optimal methods for minimization of cost