

Follow this and additional works at: <https://digitalcommons.georgiasouthern.edu/etd>

 Part of the [Artificial Intelligence and Robotics Commons](#)

JACK N. AVERITT

## Recommended Citation

- [1] Oludele Awodele and Olawale Jegede, Neural networks and its application in engineering, Proceedings of Informing Science & IT Education Conference (InSITE), 2009.
- [2] Thanatip Chankong, Nipon Theera-Umpon, and Sansanee Auephanwiriyaikul, Automatic cervical cell segmentation and classification in pap smears, *Computer Methods and Programs in Biomedicine* 113 (2014), no. 2, 539 - 556. [3] M. Anousouya Devi, S. Ravi, J. Vaishnavi, and S. Punitha, "classification of cervical cancer using artificial neural networks", *Procedia Computer Science* 89 (2016), 465 - 472, Twelfth International Communication Networks, ICCN 2016, August 19-21, 2016, Bangalore, India Twelfth International Conference on Data Mining and Warehousing, ICDMW 2016, August 19-21, 2016, Bangalore, India Twelfth International Conference on Image and Signal Processing, ICISP 2016, August 19-21, 2016, Bangalore, India. [4] Dua Dheeru and Karra Tanisikidou, UCI machine learning repository, April 2017. [5] Kelwin Fernandes, Jaime S. Cardoso, and Jessica Fernandes, Transfer Learning with Partial Observability Applied to Cervical Cancer Screening, Proceedings of Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA), 2017. [6] Eugenio Fusco, Francesco Padula, Emanuela Mancini, Alessandro Cavaliere, and Goran Grubisic, History of colposcopy: a brief biography of Hinselmann, *J Prenat Med.* 2(2) (2008), no. 4, 19-23. [7] Frauke G"unther and Stefan Fritsch, neuralnet: Training of Neural Networks, *The R Journal* 2 (2010), no. 1, 30-38. [8] Orna Intrator and Nathan Intrator, Interpreting neural-network results: A simulation study, *Comput. Stat. Data Anal.* 37 (2001), no. 3, 373-393. [9] Paulo J. Lisboa and Azzam F.G. Taktak, The use of artificial neural networks in decision support in cancer: A systematic review, *Neural Networks* 19 (2006), no. 4, 408 - 415. [10] Laurie J. Mango, Computer-assisted cervical cancer screening using neural networks, *Cancer Letters* 77 (1994), no. 2, 155 - 162, Computer applications for early detection and staging of cancer. [11] Warren S. McCulloch and Walter Pitts, A logical calculus of the ideas immanent in nervous activity, *The bulletin of mathematical biophysics* 5 (1943), no. 4, 115-133. [12] Mitchell MF, Schottenfeld D, Tortolero-Luna G, Cantor SB, and Richards-Kortum R, Colposcopy for the diagnosis of squamous intraepithelial lesions: a meta-analysis, *Obstet Gynecol.* 91 (1998), no. 4, 626-631. [13] Steven Miller, Mind: How to Build a Neural Network (Part One), <https://stevenmiller888.github.io/mind-howto-build-a-neural-network/>, March 2018. [14] Joe Pater, Did Frank Rosenblatt invent deep learning in 1962?, <https://blogs.umass.edu/comphon/2017/06/15/did-frank-rosenblatt-inventdeep-learning-in-1962/>, March 2018. [15] Xiaoping Qiu, Ning Tao, Yun Tan, and Xinxing Wu, Constructing of the risk classification model of cervical cancer by artificial neural network, *Expert Systems with Applications* 32 (2007), no. 4, 1094 - 1099. [16] Alejandro Lopez Rincon, Alberto Tonda, Mohamed Elati, Olivier Schwander, Benjamin Piwowarski, and Patrick Gallinari, Evolutionary optimization of convolutional neural networks for cancer mirna biomarkers classification, *Applied Soft Computing* (2018). [17] F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain, *Psychological Review* (1958), 65-386. [18] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams, Learning representations by back-propagating errors, *Nature* 323 (1986), 533-536. [19] Abid Sarwar, Vinod Sharma, and Rajeev Gupta, Hybrid ensemble learning technique for screening of cervical cancer using papanicolaou smear image analysis, *Personalized Medicine Universe* 4 (2015), 54 - 62. [20] Jrgen Schmidhuber, Deep

MULTICLASS CLASSIFICATION OF RISK FACTORS FOR CERVICAL CANCER  
USING ARTIFICIAL NEURAL NETWORKS

by

ABDULLAH AL MAMUN

(Under the Direction of Ionut E. Iacob)

ABSTRACT

World Health Organization statistics show that cervical cancer is the fourth most frequent cancer in women with an estimated 530,000 new cases in 2012. Cervical cancer diagnosis typically involves liquid based cytology (LBC) followed by a pathologist review. The accuracy of decision is therefore highly influenced by the expert's skills and experience, resulting in relatively high false positive and/or false negative rates. Moreover, given the fact that the data being analyzed is highly dimensional, same reviewer's decision is inherently affected by inconsistencies in interpreting the data.

In this study, we use an Artificial Neural Network based model that aims to considerably reduce experts' inconsistencies in predicting cervical cancer. We rely on standard machine learning techniques to train the neural network using six experts' predictions for cervical cancer (based on analysis of more than sixty parameters/risk factors) and we produce a model where the unanimous decision is predicted with very good accuracy.

INDEX WORDS: Neural networks, Classification, Pattern recognition

2009 Mathematics Subject Classification: 92B20, 68T10

MULTICLASS CLASSIFICATION OF RISK FACTORS FOR CERVICAL CANCER  
USING ARTIFICIAL NEURAL NETWORKS

by

ABDULLAH AL MAMUN

B.S., Jagannath University, Bangladesh, 2013

A Thesis Submitted to the Graduate Faculty of Georgia Southern University in Partial  
Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

STATESBORO, GEORGIA

©2018

ABDULLAH AL MAMUN

All Rights Reserved

MULTICLASS CLASSIFICATION OF RISK FACTORS FOR CERVICAL CANCER  
USING ARTIFICIAL NEURAL NETWORKS

by

ABDULLAH AL MAMUN

Major Professor: Ionut E. Iacob  
Committee: Stephen Carden  
Divine Wanduku

Electronic Version Approved:  
July 2018

## DEDICATION

I dedicate this to my father, mother, siblings, and friends whose support and guidance made all this possible.

## ACKNOWLEDGMENTS

I wish to acknowledge Dr. Jacob for being a decent friend and mentor and also the awesome faculty and staff at Georgia Southern's Mathematical Sciences Department.

## TABLE OF CONTENTS

	Page
ACKNOWLEDGMENTS . . . . .	3
LIST OF TABLES . . . . .	6
LIST OF FIGURES . . . . .	7
<b>CHAPTER</b>	
1 Introduction . . . . .	9
1.1 What is Cervical Cancer? . . . . .	10
1.2 The Experimental Data . . . . .	12
1.3 The Model . . . . .	15
2 Artificial Neural Networks . . . . .	18
2.1 History . . . . .	18
2.2 What are Neural Networks? . . . . .	19
2.3 Why Use Neural Networks? . . . . .	19
2.4 Types of Classifications . . . . .	20
3 Cervical Cancer Risk Prediction Using The Artificial Neural Network Model	25
3.1 The Classification Problem . . . . .	25
3.2 The Artificial Neural Network Models For Classification . . . . .	27
3.3 The Generalized Weights . . . . .	32
4 Implementation And Experimental Results . . . . .	34
4.1 Data Analysis Experiments . . . . .	35
4.2 Model Predictions Experiments . . . . .	36



4.2.1 Risk Predictions With Single ANN Models With One Output, One Model For Each Expert And Consensus . . . . .	39
4.2.2 Risk Predictions With ANN Model With 7 Outputs . . . . .	39
4.2.3 Risk Prediction With One ANN Model With One Output, For Consensus Split Decisions . . . . .	39
4.3 Model Sensitivity Analysis . . . . .	48
5 Conclusions And Future Work . . . . .	52
REFERENCES . . . . .	53
A Data Description . . . . .	56
B R code . . . . .	61
B.1 Experiment 1 . . . . .	61
B.2 Experiment 2 . . . . .	66
B.3 Experiment 2 . . . . .	71

## LIST OF TABLES

Table	Page
2.1 XOR Table . . . . .	22

## LIST OF FIGURES

Figure	Page
1.1 Cervical cancer image . . . . .	11
1.2 Image of a histologic specimen of a patient cervix using Hinselmann's test (left: original; right: processed) . . . . .	13
1.3 Image of a histologic specimen of a patient cervix using Schiller's test (left: original; right: processed) . . . . .	14
1.4 Image of a histologic specimen of a patient cervix using Green's test (left: original; right: processed) . . . . .	14
1.5 Experimental data excerpt (98 samples, 69 variables) . . . . .	15
1.6 An Artificial Neural Network with one hidden layer and a single output . . . . .	16
2.1 A Neural Network (NN) with one hidden layer. . . . .	21
2.2 Neural Networks with weights and bias. . . . .	23
2.3 Sigmoid function. . . . .	24
3.1 Artificial Neural Network with single output . . . . .	28
3.2 Artificial Neural Network with multiple outputs . . . . .	29
4.1 Experts accuracies relative to majority opinion . . . . .	36
4.2 Experts positive cases identification (relative to the majority opinion) . . . . .	37
4.3 Experts diagnostics agreements . . . . .	38
4.4 Experts predictions accuracy (Green) . . . . .	40
4.5 Experts predictions accuracy (Hinselmann) . . . . .	41
4.6 Experts predictions accuracy (Schiller) . . . . .	42

4.7	Consensus predictions accuracy . . . . .	43
4.8	Experts predictions accuracy using a single model (one layer, 8 neurons) for all experts (Green) . . . . .	44
4.9	Experts predictions accuracy using a single model (one layer, 8 neurons) for all experts (Hinselmann) . . . . .	45
4.10	Experts predictions accuracy using a single model (one layer, 8 neurons) for all experts (Schiller) . . . . .	46
4.11	Consensus predictions accuracy using a single model (one layer, 8 neurons) .	47
4.12	Consensus predictions for split decisions . . . . .	48
4.13	Positive influence . . . . .	49
4.14	Negative influence . . . . .	50
4.15	Positive and negative influence . . . . .	50

## CHAPTER 1

### INTRODUCTION

Worldwide, according to the World Health Organization fact sheet [26], cervical cancer is the fourth most frequent cancer in women with an estimated 530,000 new cases in 2012 representing 7.5% of all female cancer deaths. For women living in less developed regions, cervical cancer is the second most common cancer, with an estimated 445,000 new cases in 2012 (84% of the new cases worldwide). In 2012, approximately 270,000 women died from cervical cancer; more than 85% of these deaths occurring in low- and middle-income countries. Cervical cancer diagnosis typically involves liquid based cytology (LBC) followed by a pathologist review. The accuracy of the decision is therefore highly influenced by the expert's skills and experience, resulting in relatively high false positive ( $\sim 10\%$ ) or false negative rates ( $\sim 20\%$ ) [23]. Moreover, given the fact that the data being analyzed is high dimensional, the same reviewer's decision is inherently affected by inconsistencies in interpreting the data.

Typically, a few pathologists review a patient's screening data and present their diagnosis. The predictions can be consequently affected by the experts' ability to process large amount of information and their own subjectivity. In this study, we create an Artificial Neural Network (ANN) based model that aims to considerably reduce experts' inconsistencies in predicting cervical cancer. We rely on standard machine learning techniques to train the neural network using six experts' predictions for cervical cancer (based on analysis of more than sixty parameters/risk factors) and we produce a model where the unanimous decision is predicted with very good accuracy. The advantages of using such a model are twofold: (i) consistent predictions based on data collected during patients screening and (ii) reduced diagnostics costs by identifying the most important screening parameters and reducing the number of experts involved in the cancer risk prediction process.

## 1.1 WHAT IS CERVICAL CANCER?

Cancer begins when cells in the body start to become out of control. Cells in about any piece of the body can progress toward becoming cancer and can spread to different zones of the body.

Cervical growth begins with the cells coating the cervix - the lower portion of the uterus (womb). This is once in a while called the uterine cervix. The fetus develops in the body of the uterus (the upper part). The cervix interfaces the body of the uterus to the vagina (birth channel).

The cervix has two distinct parts and is secured with two unique sorts of cells [1]:

1. The piece of the cervix nearest to the body of the uterus is known as the endocervix and is secured with glandular cells.
2. The part beside the vagina is the exocervix (or ectocervix) and is canvassed in squamous cells.

These two cell types meet at a place called the transformation zone. The correct area of the transformation zone changes as you get more established and on the off chance that you conceive an offspring (Figure 1.1).

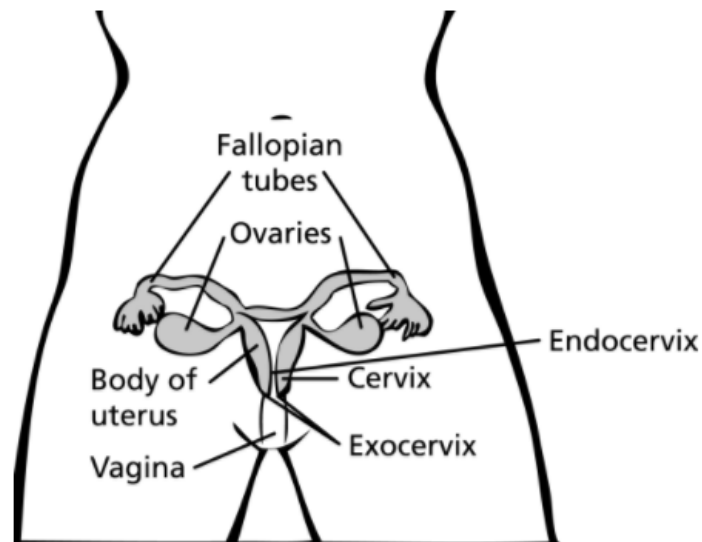


Figure 1.1: Cervical cancer image

Most cervical cancers start in the cells in the transformation zone. These cells do not abruptly change into cancer. Rather, the ordinary cells of the cervix first progressively create pre-cancerous changes that transform into cancer. Specialists utilize a few terms to depict these pre-cancerous changes, including cervical intraepithelial neoplasia (CIN), squamous intraepithelial lesion (SIL), and dysplasia. These progressions can be recognized by the Pap test and treated to keep growth from creating.

Even though cervical cancers begin from cells with pre-cancerous changes (pre-cancers), just a portion of the women with pre-cancerous cells in the cervix will create cancer. It generally takes quite a long while for cervical pre-cancer to change to cervical cancer, yet it likewise can occur in under a year. For most women, pre-cancerous cells will leave with no treatment. All things considered, in some women pre-cancers transform into genuine (intrusive) cancers. Treating all cervical pre-cancers can anticipate every cervical growth. Pre-cancerous changes and particular sorts of treatment for pre-cancers are examined in Cervical Cancer Prevention and Early Detection.

Cervical cancers and cervical pre-cancers are characterized by what they look like under a microscope. The fundamental kinds of cervical cancers are squamous cell carcinoma and adenocarcinoma.

- Most (up to 9 out of 10) cervical cancers are squamous cell carcinomas. These cancers originate from cells in the exocervix and the cancer cells have highlights of squamous cells under the magnifying lens. Squamous cell carcinomas frequently start in the transformation zone (where the exocervix joins the endocervix).
- The vast majority of the other cervical cancers are adenocarcinomas. Adenocarcinomas are cancers that originate from gland cells. Cervical adenocarcinoma originate from the mucus-producing gland cells of the endocervix. Cervical adenocarcinomas appear to have turned out to be more typical in the previous 20 to 30 years.
- Less usually, cervical cancers have highlights of both squamous cell carcinomas and adenocarcinomas. These are called adenosquamous carcinomas or blended carcinomas.

Albeit almost cervical cancers are either squamous cell carcinomas or adenocarcinomas, different kinds of cancer likewise can originate in the cervix. These different kinds (for example; melanoma, sarcoma, and lymphoma) happen normally in different parts of the body[1].

## 1.2 THE EXPERIMENTAL DATA

We give a brief description for the experimental data [5] we used for our model. The complete experimental data set description is given in Appendix A. Our model is created based on numeric measurements performed on images of histologic specimens from 98 patient's cervixes. The images were retrieved using three different standard tests: Hinselmann, Schiller, and Green. Each original image was subsequently converted into a black



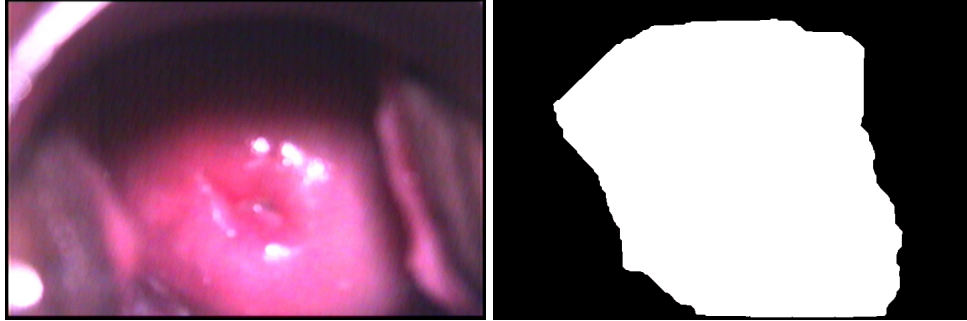


Figure 1.2: Image of a histologic specimen of a patient cervix using Hinselmann's test (left: original; right: processed)

and white image of the cervix shape and various numerical measurements were performed on both images (original and processed): cervix area, walls thicknesses, color intensities, etc.

Hinselmann's test is based on the first colposcopy experiments performed by Hinselmann as early as 1924 [7]. The method has since evolved and perfected with the contribution of many researchers. Figure 1.2 shows an image of a histologic specimen of a patient cervix using Hinselmann's test.

Schiller's test or Schiller's Iodine test [27] is a medical test in which iodine solution is applied to the cervix in order to diagnose cervical cancer. Schiller's iodine solution is applied to the cervix under direct vision. Normal cervical mucosa contains glycogen and stains brown, whereas abnormal areas, such as early cervical cancer, do not take up the stain. The abnormal areas can then be examined histologically (or biopsied). In Figure 1.3 we show an image of a histologic specimen retrieved using Schiller's test.

The Green test consists of the use of simple agents such as acetic acid and iodine (like in Schiller's test), together with the use of a green illumination filter, which can highlight suspicious regions [13]. An image of a histologic specimen of a patient cervix using Green's test is shown in Figure 1.4.

The experimental data collected consists of 98 image samples for each method. For



Figure 1.3: Image of a histologic specimen of a patient cervix using Schiller's test (left: original; right: processed)

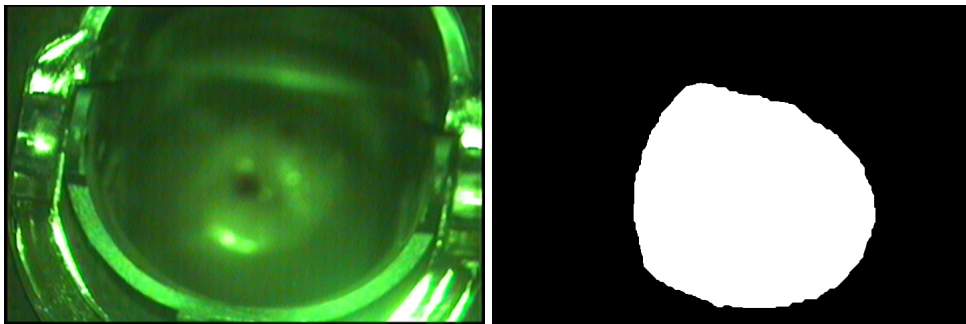


Figure 1.4: Image of a histologic specimen of a patient cervix using Green's test (left: original; right: processed)

	cervix_area	rgb_cervix_g_mean	rgb_total_g_mean_plus_std	experts..0	experts..1	experts..2	experts..3	experts..4	experts..5
1	0.02611469044	0.16511367021	0.23039489717	1	1	1	1	1	1
2	0.02629536501	0.15084490841	0.24335247935	1	1	1	0	1	1
3	0.02637156125	0.13960933996	0.17515757012	1	1	1	1	1	1
4	0.02619799369	0.16407599948	0.23079950725	1	1	1	1	1	1
5	0.02658614850	0.15636970645	0.19650784866	1	1	1	1	1	1
6	0.02652363284	0.17627865561	0.20897250215	1	0	1	1	1	0
7	0.02650263709	0.12453075207	0.17920415086	1	0	1	1	1	1
8	0.02645158688	0.15097733973	0.26013212650	1	1	1	0	1	1
9	0.02643263148	0.15747469288	0.21559374377	1	1	1	1	1	1
10	0.02606646103	0.16644452627	0.24308739811	1	0	1	0	0	0
11	0.02647429485	0.14761734411	0.22833090301	1	0	0	0	0	1
12	0.02588870384	0.28677970872	0.24153814642	1	0	0	0	1	1

Showing 1 to 12 of 97 entries

Figure 1.5: Experimental data excerpt (98 samples, 69 variables)

each sample, 62 measurements were performed and the results were stored in three different files (one file per each method). In addition to the 62 measurements, the opinions (0 = no risk, 1 = risk of cancer) of six experts were recorded for each data sample. Consequently, each data set/file contains  $98 \text{ samples} \times 69 \text{ variables}$ . A data set excerpt (12 samples and 9 variables) is shown in Figure 1.5 (In Hinselmann it has 97 samples).

### 1.3 THE MODEL

The main objective of this work is to create a mathematical model for predicting cervical cancer, based on the collected experimental data (including the 6 experts opinions for each sample). The main assumption we make is that there exists a non-linear relationship between the 62 variables representing image measurements and the opinions (0 or 1) of experts:

$$F : \mathbb{R}^{62} \rightarrow \{0, 1\}^6 \quad (1.1)$$

We create an Artificial Neural Network based model and rely on standard machine learning techniques to train our model using six experts' predictions for cervical cancer, based on the analysis of 62 parameters/risk factors. Figure 1.6 shows a general Artificial Neural Network (ANN) model with one hidden layer and a single output. For simplicity,

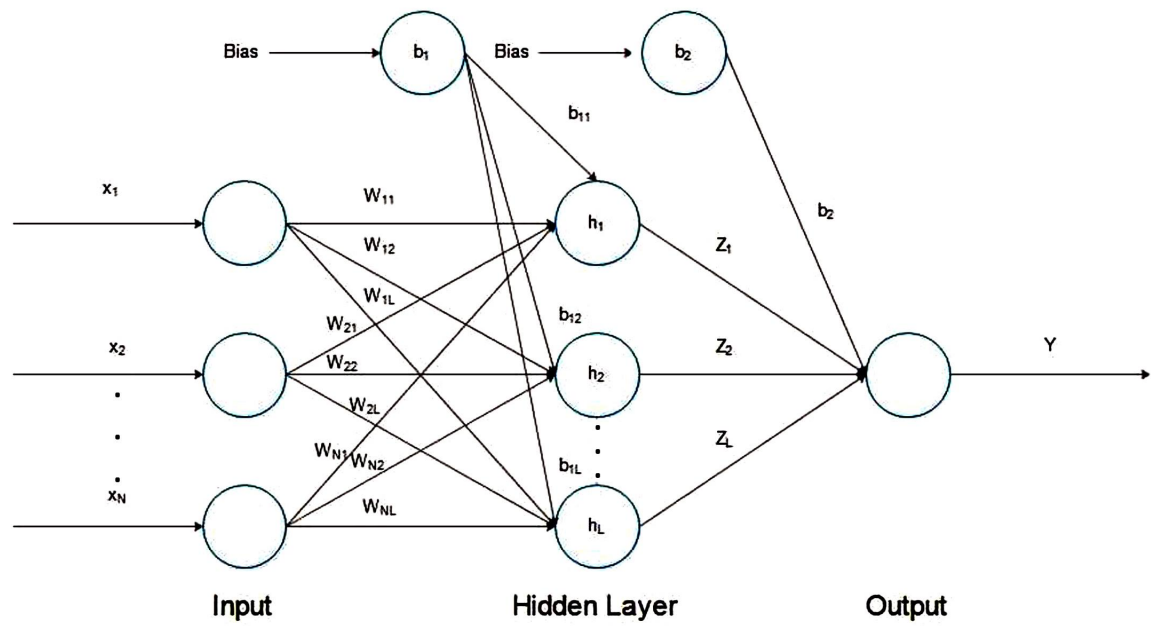


Figure 1.6: An Artificial Neural Network with one hidden layer and a single output

we show here a model with a single output, but the extension to six outputs (predictions) follows naturally from this model and it will be presented in detail in Chapter 3. For an ANN model as in Figure 1.6 with  $N$  inputs, one hidden layer with  $L$  neurons, and a single output  $y$ , the model is described by:

$$y : \mathbb{R}^{62} \rightarrow (0, 1)$$

$$y(x_1, \dots, x_N; w_{11}, \dots, w_{NL}, z_1, \dots, z_L, b_{11}, \dots, b_{1L}, b_2) = \sigma \left( \sum_{j=1}^L z_j \sigma \left( \sum_{i=1}^N w_{ij} x_i + b_{1j} \right) + b_2 \right) \quad (1.2)$$

where  $N = 62$ ,  $L$  will be determined experimentally, and  $\sigma(\cdot)$  is called the *activation function*, which in our model is the *sigmoid function*:

$$\sigma : \mathbb{R} \rightarrow \mathbb{R}, \quad \sigma(x) = \frac{1}{1 + e^{-x}}$$

Using a subset of  $S$  samples  $\{(x_{i1}, \dots, x_{i,62}, e_i) \mid i = 1 \dots S\}$  of the experimental data (where  $x_{i1}, \dots, x_{i,62}$  represent the sample's measurement values and  $e_i$  is the experts' predictions consensus for the sample), the parameters  $w_{11}, \dots, w_{NL}, z_1, \dots, z_L, b_{11}, \dots, b_{1L}, b_2$  in (1.2) are determined as the optimal solution of the unconstrained optimization problem:

$$\min \sum_{i=1}^S (-e_i \log y_i - (1 - e_i) \log(1 - y_i)) \quad (1.3)$$

(where  $y_i = y(x_{i1}, \dots, x_{i,62}; w_{11}, \dots, w_{NL}, z_1, \dots, z_L, b_{11}, \dots, b_{1L}, b_2)$ )

Finally, we define our ANN prediction model as follows.

**Definition 1.** For the given experimental data, an ANN model as in (1.2) with parameters computed using (1.3), we define the cancer risk prediction model:

$$\hat{F} : \mathbb{R}^{62} \rightarrow \{0, 1\}$$

$$\hat{F}(x_1, \dots, x_N) = \begin{cases} 1, & \text{if } y(x_1, \dots, x_N) \geq T \\ 0, & \text{if } y(x_1, \dots, x_N) < T \end{cases} \quad (1.4)$$

where  $T$  denotes the decision threshold (typically,  $T = 0.5$ ).

## CHAPTER 2

### ARTIFICIAL NEURAL NETWORKS

#### 2.1 HISTORY

In 1943, neurophysiologist Warren McCulloch and mathematician Walter Pitts wrote a paper [12] on how neurons might work. In order to describe how neurons in the brain might work, they modeled a simple neural network using electrical circuits.

In 1949, Donald Hebb wrote *The Organization of Behavior*, a work which pointed out the fact that neural pathways are strengthened each time they are used, a concept fundamentally essential to the ways in which humans learn. If two nerves fire at the same time, he argued, the connection between them is enhanced [25]. Deep learning involves training neural networks with hidden layers, sometimes many levels deep. Frank Rosenblatt (1928-1971) is widely acknowledged as a pioneer in the training of neural networks, especially for his development of the perceptron update rule, a provably convergent procedure for training single layer feedforward networks. He is less widely acknowledged for his pioneering work with other network architectures, including multi-layer perceptions and models with connections backwards through the layers, as in recurrent neural nets [15]. In 1986, with multiple layered neural networks in the news, the problem was how to extend the Widrow-Hoff rule to multiple layers. Three independent groups of researchers, one of which included David Rumelhart, a former member of Stanford's psychology department, came up with similar ideas which are now called backpropagation networks because it distributes pattern recognition errors throughout the network [19]. Hybrid networks used just two layers, these back-propagation networks use many. The result is that back-propagation networks are "slow learners," needing possibly thousands of iterations to learn. Now, neural networks are used in several applications. The fundamental idea behind the nature of neural networks is that if it works in nature, it must be able to work in computers. The

future of neural networks, though, lies in the development of hardware. Much like the advanced chess-playing machines like Deep Blue, fast, efficient neural networks depend on hardware being specified for its eventual use [25, 21].

## 2.2 WHAT ARE NEURAL NETWORKS?

An artificial neural network (ANN), often just called a “neural network” (NN), is a mathematical model or computational model based on biological neural networks, in other words, is an emulation of biological neural system. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. In most cases, an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase [22]. An ANN is arranged for a particular application, for example, pattern recognition or data classification.

## 2.3 WHY USE NEURAL NETWORKS?

Neural systems, with their striking capacity to get significance from complicated or uncertain data can be utilized to extricate designs and identify patterns that are too complex to possibly to be seen by either humans or other computer techniques[2]. A trained neural network can be thought of as a “specialist” in the classification of data, it has been given to analyze. This expert would then be able to be utilized to give projections given new circumstances and reply to “what if” questions. Other advantages include: adaptive learning, self-Organization, real-Time operation and fault tolerance via redundant information coding. Neural networks process data likewise the human brain does. The network is made from countless interconnected processing components (neurons) working in parallel to solve of a specific issue. Neural networks learn by example. They can not be modified to play out a specific task. The examples must be chosen carefully otherwise time is squan-

dered or much more dreadful the network may work inaccurately. The disadvantage is that because the network finds out how to solve the issue without anyone else's input, its task can be unusual.

Neural networks and regular algorithmic computers are not in rivalry but rather supplement each other [2].

## 2.4 TYPES OF CLASSIFICATIONS

There are two different kinds of classification:

1. Binary classification
2. Multiclass classification

In our research, we will use both classifications. ANNs incorporate the two fundamental components of biological neural nets:

1. Neurons (nodes)
2. Synapses (weights)

Neuron comprises three basic components:

1. Weights
2. Threshold
3. Activation function

A neural network is a collection of neurons with synapses associating them. This collection is composed of three main parts: the input layer, the hidden layer, and the output layer[14].

We can have “n” number hidden layers which are called multiple hidden layers. Hidden layers are essential when the neural network needs to comprehend something extremely convoluted, relevant, or non-self-evident, like image recognition. The term “deep learning” came from many hidden layers. These layers are known as “hidden” since they are not visible as a neural network.

The circles represent neurons and lines represent synapses. Where synapses take the input



and multiply it by a weight. After getting the results from all synapses then we will add them with the bias terms. At that point, we will apply an activation function.

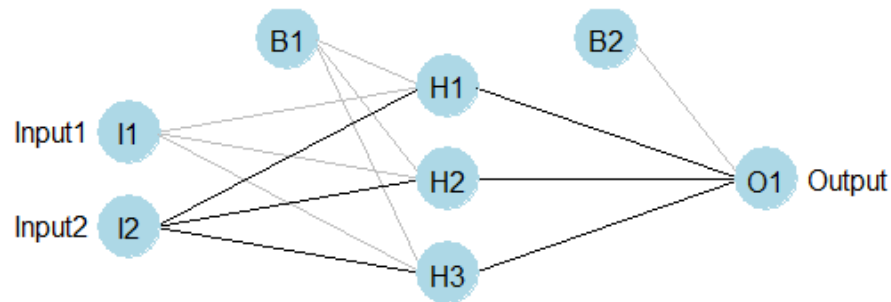


Figure 2.1: A Neural Network (NN) with one hidden layer.

Basically, training a neural network proceeds by adjusting the weights by repeating two key steps:

1. Forward propagation
2. Backpropagation

In forward propagation, we calculate the output by applying a set of weights to the input data. The set of weights is selected randomly for the first forward propagation. Now we will make a simple example of training a neural network to function as an “XOR (Exclusive or) operation, explaining each step in the training process.

We will be using as training data the inputs and outputs of the XOR function represented in the table below.

Table 2.1: XOR Table

Input	Output
0,0	1
0,1	1
1,0	1
1,1	0

Now, we allocate weights to all the synapses. Remember that these weights are chosen randomly (based on Gaussian distribution). Here, we will see how forward propagation works. To get the first values for the hidden layers, we add the product of the inputs with their corresponding set of weights and, we add bias term. Suppose two values of input are 1.

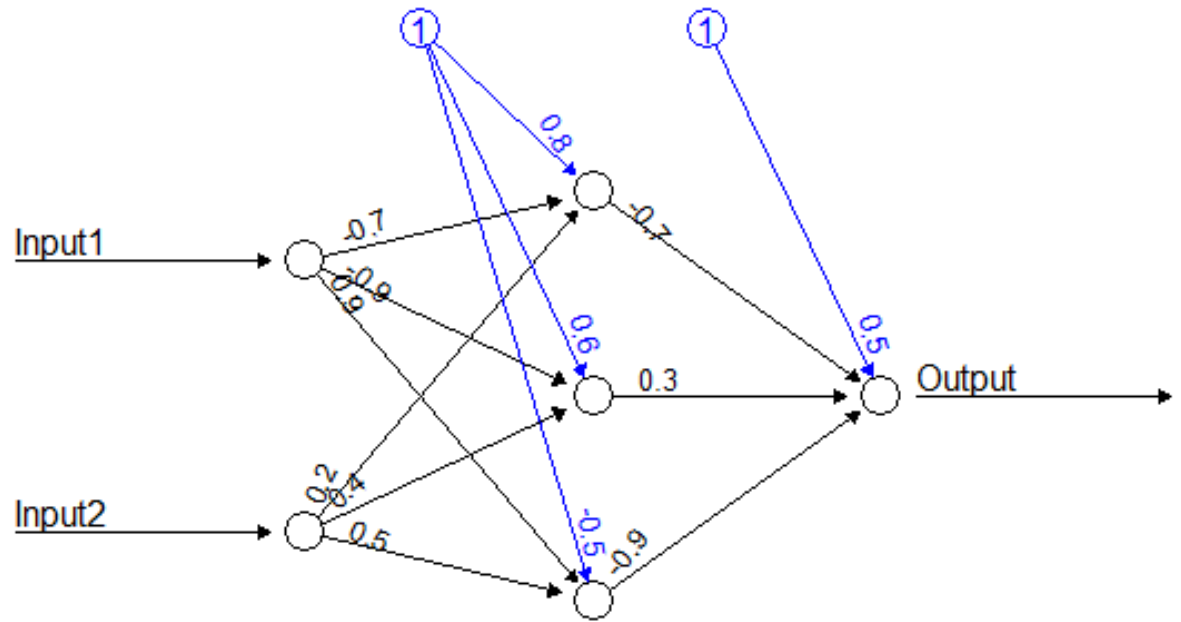


Figure 2.2: Neural Networks with weights and bias.

$$1 * (-0.7) + 1 * 0.2 + 0.8 = 0.3$$

$$1 * (-0.9) + 1 * 0.4 + 0.6 = 0.1$$

$$1 * 0.9 + 1 * 0.5 + (-0.5) = 0.9$$

Then we apply the activation function with the result to get the eventual answer. We will apply this process to the three nodes of the hidden layer. The goal of the activation function is to transform the input signal into an output signal which is necessary for neural networks to model complex non-linear patterns that simpler models may miss.

There are numerous sorts of activation functions such as linear, sigmoid, hyperbolic tangent:

- Linear:  $y = x$

- Sigmoid function (logistic):  $y = 1 / (1 + e^{-x})$
- Hyperbolic Tangent:  $y = (1 - e^{-2x}) / (1 + e^{2x})$

We will use the sigmoid function here. The sigmoid function graph is shown in Figure 2.3.

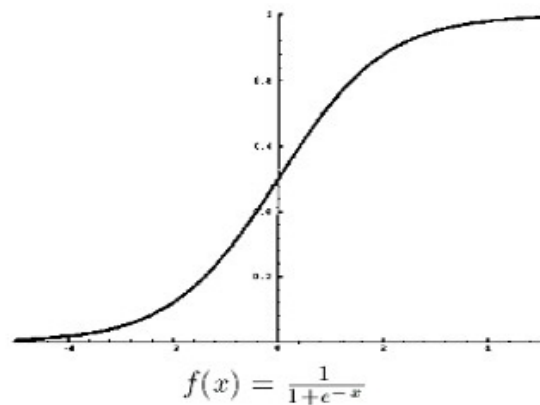


Figure 2.3: Sigmoid function.

$$S(0.3) = 0.57$$

$$S(0.1) = 0.52$$

$$S(0.9) = 0.71$$

At that point, again we add the product of the hidden layer results with the second set of weights to get the output and we add this result to second bias term. Then we apply the activation function with the result to acquire our final output.

$$0.57 * (-0.7) + 0.52 * 0.3 + 0.71 * (-0.9) + 0.5 = -0.382$$

$$S(-0.382) = 0.40564461209$$

So, our final output is 0.41.

This is the entire process of a Neural Network [14].

## CHAPTER 3

### CERVICAL CANCER RISK PREDICTION USING THE ARTIFICIAL NEURAL NETWORK MODEL

We have informally introduced the problem of cervical cancer risk prediction, described the data set, and gave a general overview of Artificial Neural Networks. In this chapter we will formally describe how we plan to use Artificial Neural Networks and the cervical cancer data to create models for cervical cancer risk predictions.

The chapter is organized as follows. We give an overview of the Statistical Learning/-Machine Learning general problem of classification in Section 3.1. The formal description for performing cervical cancer risk prediction using an ANN is given in Section 3.2. In Section 3.3 we present the concept of *generalized weights* model for computing the model inputs contributions to the model prediction error.

#### 3.1 THE CLASSIFICATION PROBLEM

*Machine Learning Classification* (Supervised learning) is the problem of identifying the unknown category or class (out of a list of categories or classes) of an observation, given the known categories (classes) of a set of observations (training dataset). When the decision is between two classes, we call the problem binary classification. When there are more than two classes, it is called multiclass classification.

A *classifier* is a mathematical model (often a function) that takes as input a new observation and produces as output the predicted class of the observation.

As briefly described in Section 1.3, our goal is to produce a classifier that takes as input 62 real-valued variables and produces 0 (no risk of cervical cancer) or 1 (risk of cervical cancer). We are therefore aiming to produce a binary classifier. However, while the final decision for predicting risk or no risk corresponds to the *consensus* output variable, our model will compute a binary decision (0 or 1) for each of the six output variables

corresponding to the six experts opinions in addition to the consensus result.

Let us first introduce a formal definition for the binary classification problem for the case of the input variables from the Euclidian space.

**Definition 2.** [Binary Classification in Euclidian Space] Let  $S, T \subseteq \mathbb{R}^n$  be finite sets. The binary classification problem consists in finding a function  $f : \mathbb{R}^n \rightarrow \{-1, 1\}$  such that

$$f(x) = \begin{cases} 1, & \text{if } x \in S \\ -1, & \text{if } x \in T \end{cases}$$

We now formally introduce the problem of cervical cancer risk prediction as a version of a binary classification problem. Let us start by introducing some notations. We let  $\mathbb{D} = [\mathbb{X} \ \mathbb{E}]$  denote the cervical cancer data set, which is vertically partitioned in the sets of covariates  $\mathbb{X} \subset \mathbb{R}^{62}$  and experts' (plus consensus') diagnostics  $\mathbb{E} \in \{0, 1\}^7$ . A seventh predicted variable, *consensus*, is computed as a majority voting of the six experts diagnostics (with positive bias, that is, three 0 votes to three 1 votes meaning 1 = positive). That is, each element  $d \in \mathbb{D}$  is of form  $\mathbf{d} = [\mathbf{x} \ \mathbf{e}]$ , where  $\mathbf{x} \in \mathbb{X}$  and  $\mathbf{e} \in \mathbb{E}$ . Recall also that diagnostic 0 means no risk of cancer and diagnostic 1 means risk of cancer. For each  $\mathbf{e} \in \mathbb{E}$ , each of the components  $\mathbf{e} = (e_1, e_2, \dots, e_7)$  corresponds, respectively, to an expert diagnostic:  $e_1 \in \{0, 1\}$  is the diagnostic of expert 1,  $e_2 \in \{0, 1\}$  is the diagnostic of expert 2, etc., with  $e_7 \in \{0, 1\}$  being the consensus.

In our work we aim to perform individual binary prediction of each expert diagnostic (including consensus), taken separately from the others, as well as a global prediction for all experts at once (including consensus). That is, we will tackle the problem by producing 7 individual models (one for each of the six experts diagnostics plus consensus) and then one global model that will predict 7 diagnostics at once.

Let us begin by formally defining the individual prediction models.

**Definition 3.** [Cervical cancer risk prediction with individual models for each expert and

consensus] The cervical cancer risk prediction with individual models for each expert consists in finding a set of models  $\{M_i \mid M_i : \mathbb{X} \rightarrow \{0, 1\}, i = 1 \dots 7\}$  such that

$$M_i(\mathbf{x}) = e_i, \forall [\mathbf{x} \mathbf{e}] \in \mathbb{D}, \mathbf{e} = (e_1, \dots, e_7), i = 1 \dots 7$$

The definition above represents an ideal model scenario: a model that is accurately mapping all covariates into the correct diagnostics. This is very difficult, if not impossible, to achieve in practice. The real model is inherently affected by errors, which we would like to estimate. For practical reasons we would also want to consider a global model capable of predicting all diagnostics (including the consensus) at once. Such a model is defined as follows.

**Definition 4.** [Cervical cancer risk prediction model for all diagnostics] The cervical cancer risk prediction global model is a mapping:

$$M : \mathbb{X} \rightarrow \{0, 1\}^7$$

where

$$M(\mathbf{x}) = (m_1(\mathbf{x}), \dots, m_7(\mathbf{x}))$$

and

$$m_i(\mathbf{x}) = e_i, \forall [\mathbf{x} \mathbf{e}] \in \mathbb{D}, \mathbf{e} = (e_1, \dots, e_7), i = 1 \dots 7$$

Having in mind these prediction models we want to achieve, we proceed next to describing how we plan to implement them using Artificial Neural Networks and how to estimate accuracy of the real models.

### 3.2 THE ARTIFICIAL NEURAL NETWORK MODELS FOR CLASSIFICATION

We implement the models described in Definitions 3 and 4 using Artificial Neural Networks with single output (as in Figure 3.1) and with multiple (seven) outputs (as sketched

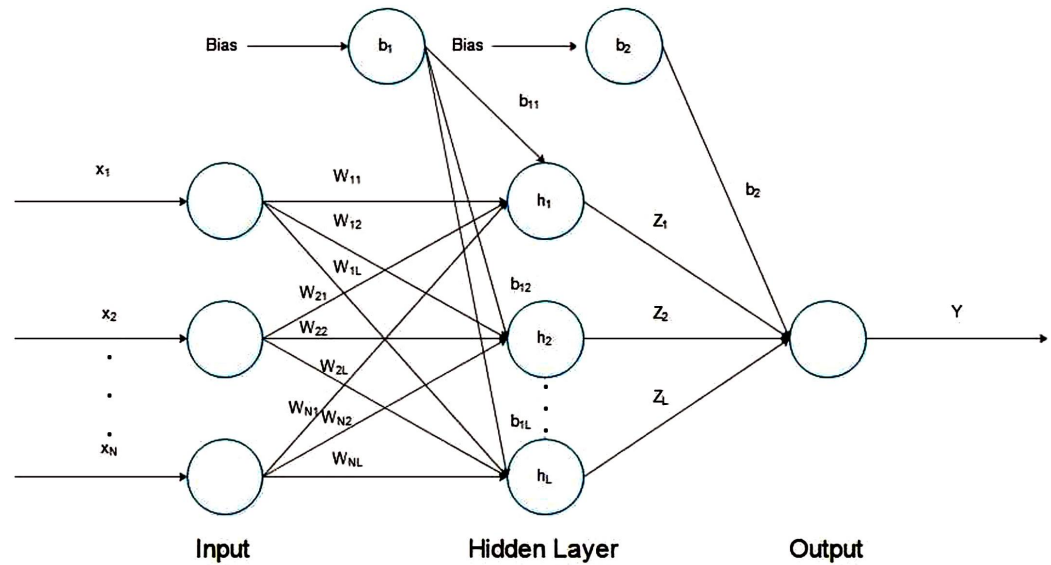


Figure 3.1: Artificial Neural Network with single output

in Figure 3.2, but with 7 outputs), respectively. Each such ANN functions as successive nonlinear transformations that map the input space  $\mathbb{D} \subset \mathbb{R}^{62}$  into the output space  $(0, 1)$  and  $(0, 1)^7$ , respectively. We notice already a difference from the ideal model: ANNs produce real values in  $(0, 1)$  rather than discrete values in  $\{0, 1\}$ .

**Definition 5.** [ANN forward step nonlinear transformation] An Artificial Neural Network forward step nonlinear transformation, from a hidden layer  $i$  with  $H_i$  neurons (can be the



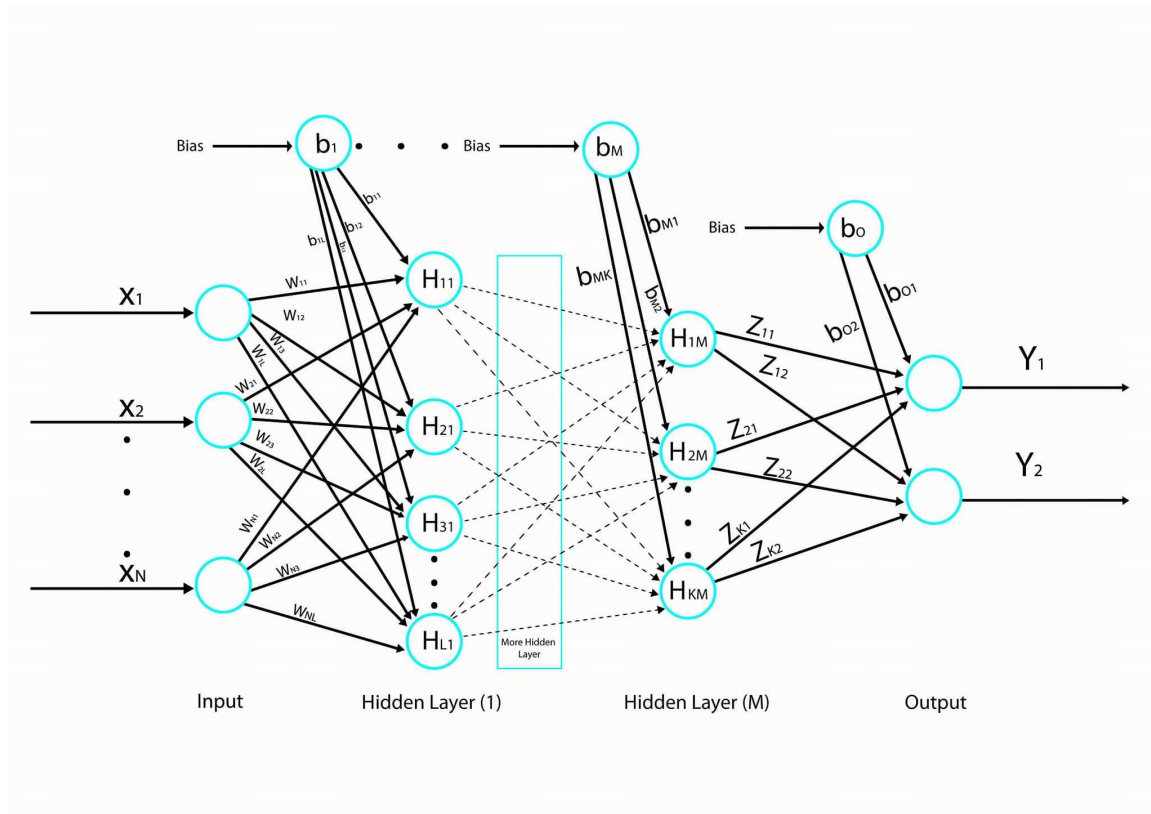


Figure 3.2: Artificial Neural Network with multiple outputs

input layer) to a hidden layer  $j$  with  $H_j$  neurons (can be the output layer), is a mapping:

$$h_{ij} : \mathbb{R}^{H_i} \rightarrow \mathbb{R}^{H_j}$$

$$h_{ij}(\mathbf{x}) = \sigma_{H_i}(W_{ij}^T \cdot \bar{\mathbf{x}})$$

where:

$$\bar{\mathbf{x}} = \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix}, \quad \mathbf{x} \in \mathbb{R}^{H_i} \text{ is the augmented input variable,}$$

$$\sigma_{H_i} : \mathbb{R}^{H_i} \rightarrow \mathbb{R}^{H_i}, \quad \sigma_{H_i}(\mathbf{x}) = (\sigma(x_1), \dots, \sigma(x_{H_i})),$$

$$\sigma : \mathbb{R} \rightarrow \mathbb{R}, \quad \sigma(x) = \frac{1}{1 + e^{-x}} \text{ is the sigmoid function, and}$$

$$W_{ij} \in \mathbb{R}_{(H_i+1) \times H_j}, \quad W_{ij} = \begin{bmatrix} b_1 & b_2 & \cdots & b_j \\ w_{11} & w_{12} & \cdots & w_{1H_j} \\ \cdots & \cdots & \cdots & \cdots \\ w_{H_i 1} & w_{H_i 2} & \cdots & w_{H_i H_j} \end{bmatrix} \text{ is the list of ANN parameters}$$

from layer  $i$  to layer  $j$  (including the bias into layer  $j$ ).

Now we introduce formally the ANN models, corresponding to the ideal scenarios introduced in the previous section, as successive ANN forward step nonlinear transformations.

**Definition 6.** [ANN models for individual diagnostics predictions] The cervical cancer risk prediction ANN individual models  $N_i : \mathbb{R}^{62} \rightarrow (0, 1)$ ,  $i = 1 \dots 7$ , are mappings:

$$\mathbb{R}^{62} \xrightarrow{h_{0H_1}^i} \mathbb{R}^{H_1} \xrightarrow{h_{H_1H_2}^i} \dots \xrightarrow{h_{H_{N_h}H_{N_h+1}}^i} (0, 1)$$

where  $N_h \geq 1$  is the number of hidden layers,  $H_l$ ,  $l = 1 \dots N_h$ , are the number of neurons per each hidden layer, and  $h_{ft}^i : \mathbb{R}^{H_f} \rightarrow \mathbb{R}^{H_t}$  is the ANN forward step nonlinear transformation from the hidden layer  $f$  to the hidden layer  $t$  (where  $f = 0$  means the input layer and  $t = N_h + 1$  means the output layer). The  $h_{ft}^i$  transformations are specific to each  $N_i$ , as they are created by solving seven different optimization problems as in (1.3).

In a similar manner, the ANN model for predicting all diagnostics is defined as follows.

**Definition 7.** [ANN model for all diagnostics predictions] The cervical cancer risk ANN prediction model for all diagnostics is a mapping  $N : \mathbb{R}^{62} \rightarrow (0, 1)^7$ :

$$\mathbb{R}^{62} \xrightarrow{h_{0H_1}} \mathbb{R}^{H_1} \xrightarrow{h_{H_1H_2}} \dots \xrightarrow{h_{H_{N_h}H_{N_h+1}}} (0, 1)^7$$

where  $N_h \geq 1$  is the number of hidden layers,  $H_l, l = 1 \dots N_h$ , are the number of neurons per each hidden layer, and  $h_{ft} : \mathbb{R}^{H_f} \rightarrow \mathbb{R}^{H_t}$  is the ANN forward step nonlinear transformation from the hidden layer  $f$  to the hidden layer  $t$  (where  $f = 0$  means the input layer and  $t = N_h + 1$  means the output layer). The  $h_{ft}$  transformations are created by solving an optimization problem as in (1.3).

Since the ANN model we create yields outputs in the interval  $(0, 1)$ , we would like to convert these output into discrete values 0 or 1 for the diagnostic decision. This is typically done by choosing a threshold  $T$  and all outputs less than  $T$  are converted to 0, and all outputs greater than or equal to  $T$  are converted to 1. We call this the ANN decision model  $N_T$  of the model  $N$ . In all our experiments we use  $T = 0.5$ , but one can use different values and introduce a corresponding bias in the final diagnostic prediction.

With these formal definition we can introduce now a practical way of measuring the accuracy of the model.

**Definition 8.** [Accuracy of an ANN model  $N$ ] Let  $N$  be an ANN model model as in Definitions 6 or 7,  $N_T$  its decision model for a given threshold  $T$ , and  $M$  be the corresponding ideal model as in Definitions 3 or 4. Let  $\mathbb{T} \subset \mathbb{D}$ , the *test set*, be a sample of the cervical cancer data. The accuracy of model  $N$  with threshold  $T$  is computed as:

$$accuracy_T(N) = \frac{|\{(\mathbf{x}, \mathbf{e}) \in \mathbb{T} : N(\mathbf{x}) = M(\mathbf{x})\}|}{|\mathbb{T}|}$$

In other words, the accuracy of an ANN model  $N$  for a given test set  $\mathbb{T} \subset \mathbb{D}$  is the number of times the ANN model  $N$  agrees with the corresponding ideal model  $M$  over the total number of samples in  $\mathbb{T}$ . In Chapter 4 we present numerical values of the model accuracy in various contexts. We used 10-fold cross-validation to sample  $\mathbb{D}$  and select the training set for computing the parameters of each model, and the testing set for computing the accuracy of the model.

### 3.3 THE GENERALIZED WEIGHTS

As part of our numerical experiments we computed the *generalized weights* [9], as measures of the contributions of input (covariates) to log-odds of the output  $y$ :

$$gw_i = \frac{\partial \left[ \log \left( \frac{y(\mathbf{x})}{1 - y(\mathbf{x})} \right) \right]}{\partial x_i}$$

This equation represents the effect of the input component (covariate)  $x_i$  to the log-odds of the output, over the whole training set used to determine the parameters of the model. In practice, the distribution of the generalized weights suggests that the covariate has no or little effect on the output when all generalized weights are zero or very small, or an important effect if the generalized weights values are large.

For instance, for an ANN with one hidden layer and  $L$  neurons with output as in (1.2):

$$y(\mathbf{x}) = \sigma \left( \sum_{j=1}^L z_j \sigma \left( \sum_{i=1}^N w_{ij} x_i + b_{1j} \right) + b_2 \right)$$

and

$$\sigma(s) = \frac{1}{1 + e^{-s}}$$

we solve for  $e^s$  in the formula for  $\sigma(\cdot)$ :

$$\sigma(s) = \frac{1}{1 + e^{-s}} = y(\mathbf{x})$$

to obtain:

$$e^s = \frac{y(\mathbf{x})}{1 - y(\mathbf{x})}$$

We take logarithms of both sides:

$$s = \log \left( \frac{y(\mathbf{x})}{1 - y(\mathbf{x})} \right)$$

and hence:

$$\log \left( \frac{y(\mathbf{x})}{1 - y(\mathbf{x})} \right) = \sum_{j=1}^L z_j \sigma \left( \sum_{i=1}^N w_{ij} x_i + b_{1j} \right) + b_2$$

Then the generalized weight  $gw_i$  of covariate  $x_i$  can be computed as:

$$gw_i = \frac{\partial \left[ \log \left( \frac{y(\mathbf{x})}{1 - y(\mathbf{x})} \right) \right]}{\partial x_i} = \sum_{j=1}^L z_j \sigma' \left( \sum_{i=1}^N w_{ij} x_i + b_{1j} \right) w_{ij}$$

Finally, by noting that:

$$\sigma'(s) = \sigma(s) (1 - \sigma(s))$$

we obtain:

$$gw_i = \sum_{j=1}^L z_j \sigma \left( \sum_{i=1}^N w_{ij} x_i + b_{1j} \right) \left[ 1 - \sigma \left( \sum_{i=1}^N w_{ij} x_i + b_{1j} \right) \right] w_{ij} \quad (3.1)$$

## CHAPTER 4

## IMPLEMENTATION AND EXPERIMENTAL RESULTS

We performed experiments on three different data files [5]: `green.csv`, `hinselmann.csv`, and `schiller.csv`. Each file consists of 98 samples of numerical data (62 variables) harvested from images of histologic specimens from 98 patients' cervixes. Each data file corresponds to a colposcopy method: Green, Hinselmann, and Schiller, respectively. In our subsequent experimental results description the indicated experiment method will implicitly identify the data file used for the respective experiment. In addition to the 62 numerical variable (*predictors*), each file contains 6 independent diagnostics from 6 experts (pathologists). These are binary (*predicted*) variables with 0 meaning negative (no cancer) and 1 meaning positive (cancer). A seventh predicted variable, *consensus*, is computed as a majority voting of the six experts diagnostics (with positive bias, that is, three 0 votes to three 1 votes meaning 1 = positive). The voting bias for positive diagnostic has a clear practical explanation: a patient would be identified as positive as a preventative measure (and subsequently follow up with further investigations, including biopsy), rather than being identified as negative and possibly let the disease evolve undetected.

The model was implemented in R v3.3.3, running on Windows 10, 64-bit Intel Core i7 CPU @3.40GHz, 16GB RAM. As our main objective was obtaining the best accuracy rather than comparing the model performance using different libraries, we only test the model with the *neuralnet* package [8].

Our experiments were organized in three major categories:

1. Data analysis. We analyzed the original data (for each method) and identified experts precision (relative to the majority diagnostic), their positive diagnostics (as a measure of the positive diagnostic bias), and their agreement with each-other and majority.
2. Model predictions. We implemented the model described in Chapter 3 and performed fine tuning in order to identify model parameters for best predictions.

3. Model sensitivity analysis. We performed a generalized weights analysis for the model parameters, in order to determine the importance of each model parameter for the model prediction.

These experimental results are reported in the subsequent subsections.

#### 4.1 DATA ANALYSIS EXPERIMENTS

The original data analysis was conducted in three directions: (i) experts diagnostics accuracies, (ii) experts positive diagnostics, and (iii) experts diagnostics agreements with each-other and consensus. We report the findings below.

- I. The accuracy percentages for each expert:

$$accuracy(E_i) = \frac{\# \text{ cases expert } E_i \text{ agrees with consensus}}{\# \text{ of all cases}}, \quad i = 1, \dots, 6$$

are reported in Figure 4.1.

- II. Figure 4.2 reports the positive cases identification of each expert, relative to the majority diagnostic:

$$positive(E_i) = \frac{\# \text{ of expert } E_i \text{ positive cases in agreement with consensus}}{\# \text{ of consensus positive cases}}, \quad i = 1, \dots, 6$$

- III. The experts diagnostics agreements with each-other and consensus are illustrated in the heatmaps in Figure 4.3.

In each figure, a square color intensity shows mutual agreement/disagreement percentages between all pairs of experts  $E_i$  and  $E_j$ , where  $(i, j)$  are the coordinates of the centers of squares:

$$agreement(E_i, E_j) = \frac{\# \text{ of cases } E_i \text{ agrees with } E_j}{\# \text{ of all cases}}, \quad i, j = 1, \dots, 6$$

Clearly, the diagonal indicates 100% agreement, as it always represents the total agreement of an experts with himself/herself ( $agreement(E_1, E_1)$ ,  $agreement(E_2, E_2)$ , etc.).

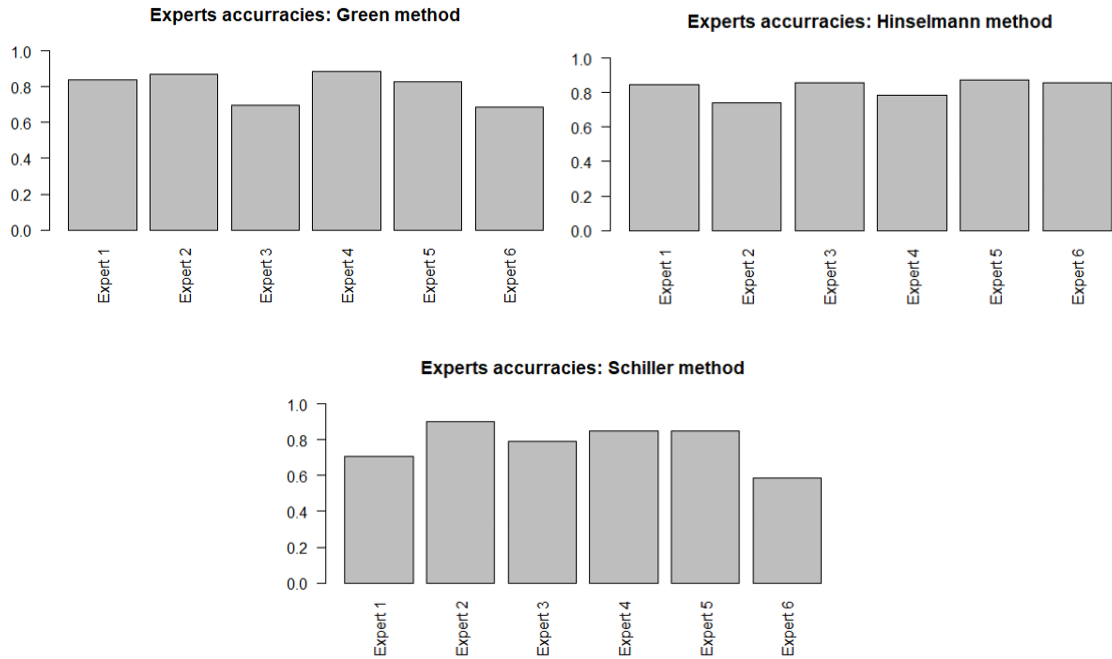


Figure 4.1: Experts accuracies relative to majority opinion

## 4.2 MODEL PREDICTIONS EXPERIMENTS

We implemented ANN prediction models with single and multiple outputs and performed experiments in order to identify the best prediction models. We organized our experiments in three groups:

- I. ANN models with one output, one model for each expert, respectively, and one model for the consensus prediction.
- II. ANN models with 7 outputs (for predicting 6 experts diagnostics and one consensus diagnostic).
- III. ANN models trained from samples where consensus is decided by majority and predicting consensus for split decisions (3 zeros and 3 ones).

In each case we identified the best model and run 10 trials, averaging the accuracies in each



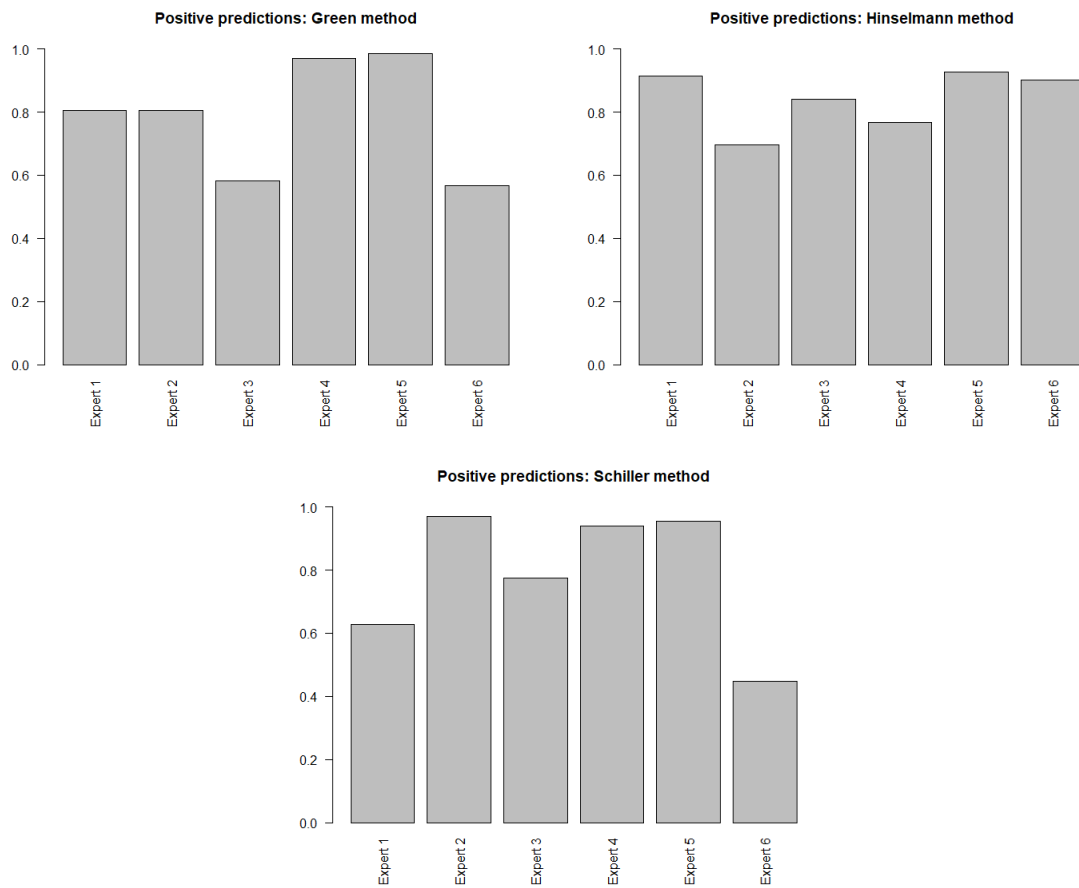


Figure 4.2: Experts positive cases identification (relative to the majority opinion)

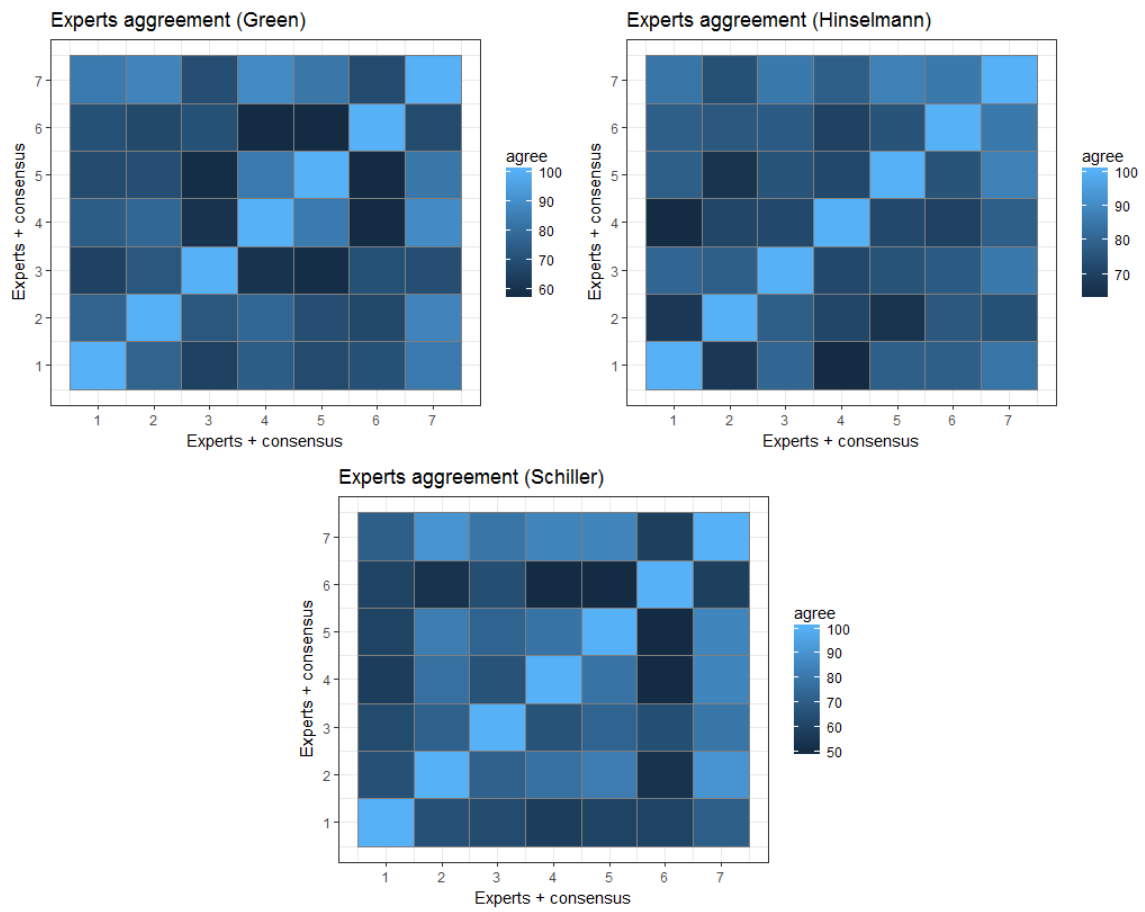


Figure 4.3: Experts diagnostics agreements

case. The best experimental results are reported below and the complete R code for these experiments is listed in Annex B.

#### 4.2.1 RISK PREDICTIONS WITH SINGLE ANN MODELS WITH ONE OUTPUT, ONE MODEL FOR EACH EXPERT AND CONSENSUS

We run the code in Annex B.1 for each of the data file 7 times, once per each of the six experts plus consensus. From multiple trials we identified the best model architecture (number of hidden layers and neurons for each layer).

The experimental results are presented for each expert per each data file in Figures 4.4, 4.5, and 4.6 and consensus for all data files in Figure 4.7.

An interesting remark is that the predicted accuracies match, in general, the respective expert's accuracy relative to the majority opinion in Figure 4.1.

#### 4.2.2 RISK PREDICTIONS WITH ANN MODEL WITH 7 OUTPUTS

For this experiment we trained one model with a single hidden layer and 8 neurons. We computed accuracy predictions for all experts and consensus and the results are shown in Figures 4.8, 4.9, 4.10, and 4.11. We found that using a single model is much more difficult to train and less accurate in practice.

The R code for this experiment is listed in Annex B.2.

#### 4.2.3 RISK PREDICTION WITH ONE ANN MODEL WITH ONE OUTPUT, FOR CONSENSUS SPLIT DECISIONS

For this experiment we trained one model which learns from consensus when there is a majority decision (more zeros or more ones) and predicts consensus for experts' split decision (3 ones, 3 zeros). Interestingly, this experiment easily produced accuracy of over

<pre>[1] "Expert 1" [1] "Layers: 12, 64"       Predicted Original 0 1       0 7 4       1 0 9 [1] "Accuracy: 0.8"</pre>	<pre>[1] "Expert 2" [1] "Layers: 40, 24"       Predicted Original 0 1       0 5 4       1 1 10 [1] "Accuracy: 0.75"</pre>
<pre>[1] "Expert 3" [1] "Layers: 40, 24, 12"       Predicted Original 0 1       0 10 1       1 3 6 [1] "Accuracy: 0.8"</pre>	<pre>[1] "Expert 4" [1] "Layers: 40, 24, 12"       Predicted Original 0 1       0 3 1       1 3 13 [1] "Accuracy: 0.8"</pre>
<pre>[1] "Expert 5" [1] "Layers: 8, 16, 4"       Predicted Original 0 1       0 1 0       1 1 18 [1] "Accuracy: 0.95"</pre>	<pre>[1] "Expert 6" [1] "Layers: 8, 32, 16"       Predicted Original 0 1       0 11 0       1 7 2 [1] "Accuracy: 0.65"</pre>

Figure 4.4: Experts predictions accuracy (Green)

<pre>[1] "Expert 1" [1] "Layers: 24, 64"       Predicted Original  0  1           0  3  2           1  0 15 [1] "Accuracy: 0.9"</pre>	<pre>[1] "Expert 2" [1] "Layers: 64, 32, 10"       Predicted Original  0  1           0  7  2           1  5  6 [1] "Accuracy: 0.65"</pre>
<pre>[1] "Expert 3" [1] "Layers: 40, 32, 12"       Predicted Original  0  1           0  5  2           1  0 13 [1] "Accuracy: 0.9"</pre>	<pre>[1] "Expert 4" [1] "Layers: 64, 64, 24"       Predicted Original  0  1           0  4  0           1  4 12 [1] "Accuracy: 0.8"</pre>
<pre>[1] "Expert 5" [1] "Layers: 64, 128, 16, 8"       Predicted Original  1           0  3           1 17 [1] "Accuracy: 0.85"</pre>	<pre>[1] "Expert 6" [1] "Layers: 64, 40, 4"       Predicted Original  0  1           0  2  3           1  2 13 [1] "Accuracy: 0.75"</pre>

Figure 4.5: Experts predictions accuracy (Hinselmann)

<pre>[1] "Expert 1" [1] "Layers: 256, 64, 32"       Predicted Original 0 1       0 7 2       1 2 8 [1] "Accuracy: 0.7894"</pre>	<pre>[1] "Expert 2" [1] "Layers: 256, 64, 32"       Predicted Original 0 1       0 1 3       1 1 14 [1] "Accuracy: 0.7894"</pre>
<pre>[1] "Expert 3" [1] "Layers: 256, 4"       Predicted Original 0 1       0 3 2       1 4 10 [1] "Accuracy: 0.6842"</pre>	<pre>[1] "Expert 4" [1] "Layers: 64, 8"       Predicted Original 0 1       0 2 1       1 1 15 [1] "Accuracy: 0.8947"</pre>
<pre>[1] "Expert 5" [1] "Layers: 256, 64, 8"       Predicted Original 0 1       0 0 1       1 1 17 [1] "Accuracy: 0.8947"</pre>	<pre>[1] "Expert 6" [1] "Layers: 64, 32, 4"       Predicted Original 0 1       0 9 0       1 7 3 [1] "Accuracy: 0.6315"</pre>

Figure 4.6: Experts predictions accuracy (Schiller)

<pre>[1] "Dataset: green.csv" [1] "Expert 7" [1] "Layers: 32, 40"        Predicted Original  0  1           0  6  0           1  2 12 [1] "Accuracy: 0.9"</pre>	<pre>[1] "Dataset: hinselmann.csv" [1] "Expert 7" [1] "Layers: 64, 32, 14"        Predicted Original  1           0  3           1 17 [1] "Accuracy: 0.85"</pre>
---	--

<pre>[1] "Dataset: schiller.csv" [1] "Expert 7" [1] "Layers: 40, 12"        Predicted Original  0  1           0  2  2           1  2 13 [1] "Accuracy: 0.7894"</pre>
---

Figure 4.7: Consensus predictions accuracy

<pre>[1] "Expert 1"       Predicted Original 0 1       0 1 10       1 0 9 [1] "Accuracy: 0.5"</pre>	<pre>[1] "Expert 2"       Predicted Original 0 1       0 3 6       1 0 11 [1] "Accuracy: 0.7"</pre>
<pre>[1] "Expert 3"       Predicted Original 0 1       0 8 3       1 9 0 [1] "Accuracy: 0.4"</pre>	<pre>[1] "Expert 4"       Predicted Original 0 1       0 2 2       1 0 16 [1] "Accuracy: 0.9"</pre>
<pre>[1] "Expert 5"       Predicted Original 0 1       0 0 1       1 1 18 [1] "Accuracy: 0.9"</pre>	<pre>[1] "Expert 6"       Predicted Original 0 1       0 8 3       1 9 0 [1] "Accuracy: 0.4"</pre>

Figure 4.8: Experts predictions accuracy using a single model (one layer, 8 neurons) for all experts (Green)



<pre>[1] "Expert 1"       Predicted Original  1          0  5          1 15 [1] "Accuracy: 0.75"</pre>	<pre>[1] "Expert 2"       Predicted Original 0 1          0 0 9          1 2 9 [1] "Accuracy: 0.45"</pre>
<pre>[1] "Expert 3"       Predicted Original  1          0  7          1 13 [1] "Accuracy: 0.65"</pre>	<pre>[1] "Expert 4"       Predicted Original  0  1          0  0  4          1  1 15 [1] "Accuracy: 0.75"</pre>
<pre>[1] "Expert 5"       Predicted Original  1          0  3          1 17 [1] "Accuracy: 0.85"</pre>	<pre>[1] "Expert 6"       Predicted Original  1          0  5          1 15 [1] "Accuracy: 0.75"</pre>

Figure 4.9: Experts predictions accuracy using a single model (one layer, 8 neurons) for all experts (Hinselmann)

<pre>[1] "Expert 1"       Predicted Original 0 1       0 2 7       1 3 7 [1] "Accuracy: 0.4736"</pre>	<pre>[1] "Expert 2"       Predicted Original 0 1       0 0 4       1 1 14 [1] "Accuracy: 0.7368"</pre>
<pre>[1] "Expert 3"       Predicted Original 0 1       0 1 4       1 3 11 [1] "Accuracy: 0.6315"</pre>	<pre>[1] "Expert 4"       Predicted Original 1       0 3       1 16 [1] "Accuracy: 0.8421"</pre>
<pre>[1] "Expert 5"       Predicted Original 0 1       0 0 1       1 1 17 [1] "Accuracy: 0.8947"</pre>	<pre>[1] "Expert 6"       Predicted Original 0       0 9       1 10 [1] "Accuracy: 0.4736"</pre>

Figure 4.10: Experts predictions accuracy using a single model (one layer, 8 neurons) for all experts (Schiller)

<pre>[1] "Dataset: green.csv" [1] "Expert 7"       Predicted Original  0  1       0  2  4       1  0 14 [1] "Accuracy: 0.8"</pre>	<pre>[1] "Dataset: hinselmann.csv" [1] "Expert 7"       Predicted Original  1       0  3       1 17 [1] "Accuracy: 0.85"</pre>
<pre>[1] "Dataset: schiller.csv" [1] "Expert 7"       Predicted Original  0  1       0  0  4       1  2 13 [1] "Accuracy: 0.6842"</pre>	

Figure 4.11: Consensus predictions accuracy using a single model (one layer, 8 neurons)

<pre>[1] "Dataset: green.csv" [1] "Expert 7" [1] "Layers: 64, 24"       Predicted Original  0  1       1  1 10 [1] "Accuracy: 0.9090"</pre>	<pre>[1] "Dataset: hinselmann.csv" [1] "Expert 7" [1] "Layers: 64, 24"       Predicted Original  1       1 12 [1] "Accuracy: 1"</pre>
<pre>[1] "Dataset: schiller.csv" [1] "Expert 7" [1] "Layers: 64, 32, 4"       Predicted Original  0  1       1  1  9 [1] "Accuracy: 0.9"</pre>	

Figure 4.12: Consensus predictions for split decisions

90% for all data files. The experiment falls into the category of “transfer of learning” [6], in the sense that the model is trained for data where consensus is achieved by majority then used to predict consensus for split decisions. The results of the experiment are presented in Figure 4.12.

The R code for this experiment is listed in Annex B.3.

### 4.3 MODEL SENSITIVITY ANALYSIS

As indicated in (1.3), our ANN model is trained by minimizing the *cross-entropy* output error. We consequently perform an analysis of the model effectiveness by computing the generalized weights [9] for a few features (covariates) and the “consensus” output. Such

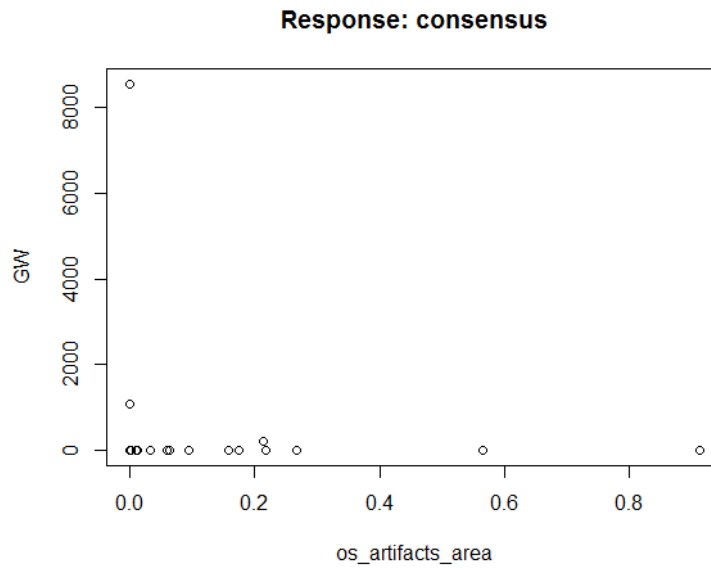


Figure 4.13: Positive influence

generalized weight  $w_i$  expresses the contribution of the  $i$ th corresponding feature  $x_i$  to the log-odds of the 7th output (“consensus”):

$$gw_i = \frac{\partial \left[ \log \left( \frac{y_7(\mathbf{x})}{1 - y_7(\mathbf{x})} \right) \right]}{\partial x_i}$$

In some sense, the generalized weight has an interpretation analogous to the corresponding regression parameter in regression models. The difference is that the generalized weight depends on all covariates.

We checked the generalized weights for all 62 variables in the case of an ANN model for all 7 outputs. We used one hidden layer with 8 neurons and computed the generalized weights for the “consensus” output. Sample plots corresponding to three of the input variables () are displayed in Figures 4.13, 4.14, and 4.15. In our experiments we did not find any variable that produced zero variation of the corresponding generalized weight. However, the examples we show illustrate three different case scenarios we found: (i) Figure 4.13 shows a very strong, positive influence of the “os\_artifacts\_area” covariate concentrated in

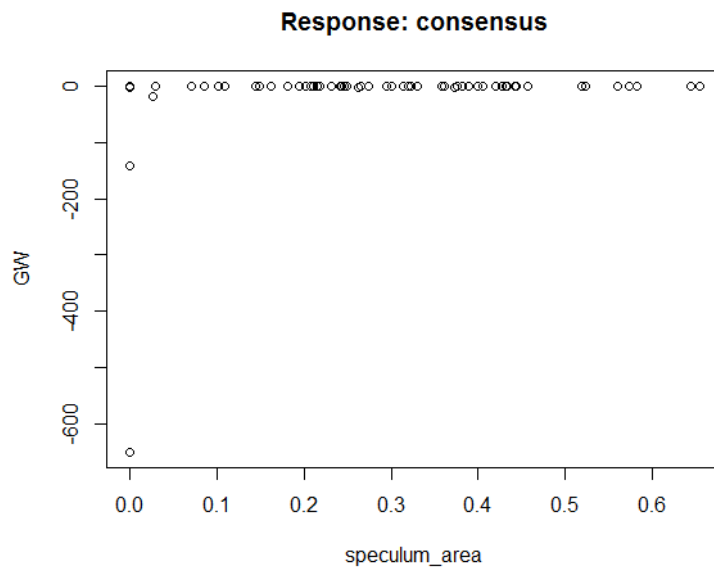


Figure 4.14: Negative influence

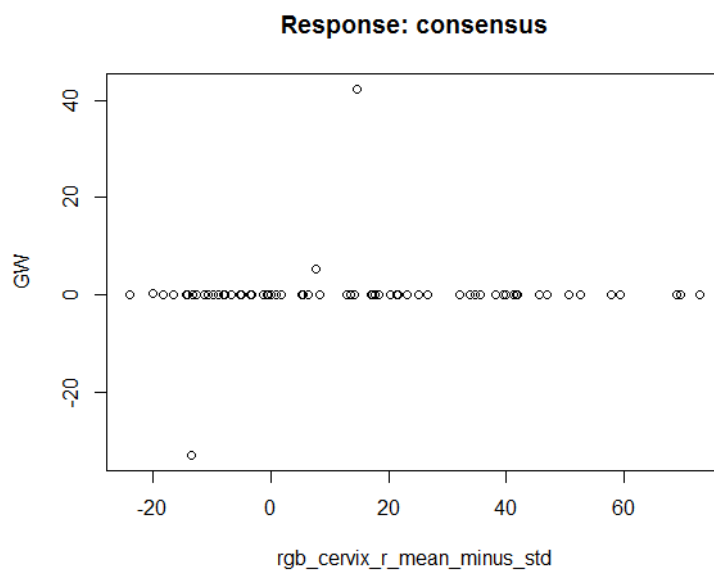


Figure 4.15: Positive and negative influence

the neighborhood of zero ; (ii) Figure 4.14 shows a very strong, negative influence of the “speculum\_area” covariate concentrated in the neighborhood of zero; and (iii) Figure 4.15 shows both a positive and negative effect of the covariate “rgb\_cervix\_r\_mean\_minus\_std”.

## CHAPTER 5

### CONCLUSIONS AND FUTURE WORK

Summary of contributions:

- Cervical cancer is one of the most deadly cancers for women worldwide. It may be expensive to fully diagnose. We present a model that can assist experts to predict the disease in (i) a consistent and (ii) less expensive manner.
- We use Neural Networks to perform diagnostic predictions on cervical cancer data
- We implemented the model in R and presented experimental results.
- We performed predictions for (i) each expert (out of 6 opinions) and (ii) consensus.
- We used a cross-entropy minimization error method.
- We performed sensitivity measures of each input parameter on the model prediction error.



## REFERENCES

- [1] American Cancer Society, *What is cervical cancer?*, [https:// www.cancer.org/cancer/cervical- cancer/ about/what- is- cervical- cancer.html](https://www.cancer.org/cancer/cervical-cancer/about/what-is-cervical-cancer.html), March 2018.
- [2] Oludele Awodele and Olawale Jegede, *Neural networks and its application in engineering*, Proceedings of Informing Science & IT Education Conference (InSITE), 2009.
- [3] Thanatip Chankong, Nipon Theera-Umpon, and Sansanee Auephanwiriyakul, *Automatic cervical cell segmentation and classification in pap smears*, Computer Methods and Programs in Biomedicine **113** (2014), no. 2, 539 – 556.
- [4] M. Anousouya Devi, S. Ravi, J. Vaishnavi, and S. Punitha, "*classification of cervical cancer using artificial neural networks*", Procedia Computer Science **89** (2016), 465 – 472, Twelfth International Conference on Communication Networks, ICCN 2016, August 19-21, 2016, Bangalore, India Twelfth International Conference on Data Mining and Warehousing, ICDMW 2016, August 19-21, 2016, Bangalore, India Twelfth International Conference on Image and Signal Processing, ICISP 2016, August 19-21, 2016, Bangalore, India.
- [5] Dua Dheeru and Efi Karra Taniskidou, *UCI machine learning repository*, April 2017.
- [6] Kelwin Fernandes, Jaime S. Cardoso, and Jessica Fernandes, *Transfer Learning with Partial Observability Applied to Cervical Cancer Screening*, Proceedings of Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA), 2017.
- [7] Eugenio Fusco, Francesco Padula, Emanuela Mancini, Alessandro Cavaliere, and Goran Grubisic, *History of colposcopy: a brief biography of Hinselmann*, J Prenat Med. **2(2)** (2008), no. 4, 19–23.
- [8] Frauke Günther and Stefan Fritsch, *neuralnet: Training of Neural Networks*, The R Journal **2** (2010), no. 1, 30–38.
- [9] Orna Intrator and Nathan Intrator, *Interpreting neural-network results: A simulation study*, Comput. Stat. Data Anal. **37** (2001), no. 3, 373–393.
- [10] Paulo J. Lisboa and Azzam F.G. Taktak, *The use of artificial neural networks in decision support in cancer: A systematic review*, Neural Networks **19** (2006), no. 4, 408 – 415.

- [11] Laurie J. Mango, *Computer-assisted cervical cancer screening using neural networks*, *Cancer Letters* **77** (1994), no. 2, 155 – 162, Computer applications for early detection and staging of cancer.
- [12] Warren S. McCulloch and Walter Pitts, *A logical calculus of the ideas immanent in nervous activity*, *The bulletin of mathematical biophysics* **5** (1943), no. 4, 115–133.
- [13] Mitchell MF, Schottenfeld D, Tortolero-Luna G, Cantor SB, and Richards-Kortum R, *Colposcopy for the diagnosis of squamous intraepithelial lesions: a meta-analysis*, *Obstet Gynecol.* **91** (1998), no. 4, 626–631.
- [14] Steven Miller, *Mind: How to Build a Neural Network (Part One)*, [https:// stevenmiller888.github.io/ mind - how - to - build - a - neural - network/](https://stevenmiller888.github.io/mind-how-to-build-a-neural-network/), March 2018.
- [15] Joe Pater, *Did frank rosenblatt invent deep learning in 1962?*, [https:// blogs.umass.edu/ comphon/ 2017/ 06/15/ did - frank - rosenblatt - invent - deep - learning - in - 1962/](https://blogs.umass.edu/comphon/2017/06/15/did-frank-rosenblatt-invent-deep-learning-in-1962/), March 2018.
- [16] Xiaoping Qiu, Ning Tao, Yun Tan, and Xinxing Wu, *Constructing of the risk classification model of cervical cancer by artificial neural network*, *Expert Systems with Applications* **32** (2007), no. 4, 1094 – 1099.
- [17] Alejandro Lopez Rincon, Alberto Tonda, Mohamed Elati, Olivier Schwander, Benjamin Piwowarski, and Patrick Gallinari, *Evolutionary optimization of convolutional neural networks for cancer mirna biomarkers classification*, *Applied Soft Computing* (2018).
- [18] F. Rosenblatt, *The perceptron: A probabilistic model for information storage and organization in the brain*, *Psychological Review* (1958), 65–386.
- [19] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams, *Learning representations by back-propagating errors*, *Nature* **323** (1986), 533–536.
- [20] Abid Sarwar, Vinod Sharma, and Rajeev Gupta, *Hybrid ensemble learning technique for screening of cervical cancer using papanicolaou smear image analysis*, *Personalized Medicine Universe* **4** (2015), 54 – 62.
- [21] Jrgen Schmidhuber, *Deep learning in neural networks: An overview*, *Neural Networks* **61** (2015), 85 – 117.

- [22] Yashpal Singh and Alok Singh Chauhan, *Neural Networks in Data Mining*, Journal of Theoretical and Applied Information Technology **5** (2009), no. 1, 37–42.
- [23] Mark H. Stoler, *Advances in cervical screening technology*, Modern Pathology **13** (2000), 275–284.
- [24] Siti Noraini Sulaiman, Nor Ashidi Mat-Isa, Nor Hayati Othman, and Fadzil Ahmad, *Improvement of features extraction process and classification of cervical cancer for the neuralpap system*, Procedia Computer Science **60** (2015), 750 – 759, Knowledge-Based and Intelligent Information & Engineering Systems 19th Annual Conference, KES-2015, Singapore, September 2015 Proceedings.
- [25] Web, *Neural networks*, [https:// cs.stanford.edu/ people/ eroberts/ courses/ soco/ projects/ neural - networks/ History/ history1.html](https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history1.html), March 2018.
- [26] WHO, *Human papillomavirus (hpv) and cervical cancer fact sheet*, <http://www.who.int/mediacentre/factsheets/fs380/en/>, June 2016.
- [27] Wikipedia, *Schiller's test*, <https://en.wikipedia.org/wiki/Schiller>

## Appendix A

### DATA DESCRIPTION

#### I. Experimental Data Set Source:

<https://archive.ics.uci.edu/ml/datasets/Quality+Assessment+of+Digital+Colposcopies>

#### II. Experimental Data Set Information:

- The dataset was acquired and annotated by professional physicians at 'Hospital Universitario de Caracas'.
- The subjective judgments (target variables) were originally done in an ordinal manner (poor, fair, good, excellent) and was discretized in two classes (bad, good).
- Images were randomly sampled from the original colposcopic sequences (videos).
- The original images and the manual segmentations are included in the 'images' directory.
- The dataset has three modalities (i.e. Hinselmann, Green, Schiller).
- Number of Attributes: 69 (62 predictive attributes, 7 target variables)
- The target variables are expert::X (X in 0,...,5) and consensus.

#### III. Attribute Information:

1. cervix\_area: image area with cervix.
2. os\_area: image area with external os.
3. walls\_area: image area with vaginal walls.
4. speculum\_area: image area with the speculum.
5. artifacts\_area: image area with artifacts.
6. cervix\_artifacts\_area: cervix area with the artifacts.

7. os\_artifacts\_area: external os area with the artifacts.
8. walls\_artifacts\_area: vaginal walls with the artifacts.
9. speculum\_artifacts\_area: speculum area with the artifacts.
10. cervix\_specularities\_area: cervix area with the specular reflections.
11. os\_specularities\_area: external os area with the specular reflections.
12. walls\_specularities\_area: vaginal walls area with the specular reflections.
13. speculum\_specularities\_area: speculum area with the specular reflections.
14. specularities\_area: total area with specular reflections.
15. area\_h\_max\_diff: maximum area differences between the four cervix quadrants.
16. rgb\_cervix\_r\_mean: average color information in the cervix (R channel).
17. rgb\_cervix\_r\_std: stddev color information in the cervix (R channel).
18. rgb\_cervix\_r\_mean\_minus\_std: (avg - stddev) color information in the cervix (R channel).
19. rgb\_cervix\_r\_mean\_plus\_std: (avg + stddev) information in the cervix (R channel).
20. rgb\_cervix\_g\_mean: average color information in the cervix (G channel).
21. rgb\_cervix\_g\_std: stddev color information in the cervix (G channel).
22. rgb\_cervix\_g\_mean\_minus\_std: (avg - stddev) color information in the cervix (G channel).
23. rgb\_cervix\_g\_mean\_plus\_std: (avg + stddev) color information in the cervix (G channel).
24. rgb\_cervix\_b\_mean: average color information in the cervix (B channel).
25. rgb\_cervix\_b\_std: stddev color information in the cervix (B channel).

26. `rgb_cervix_b_mean_minus_std`: (avg - stddev) color information in the cervix (B channel).
27. `rgb_cervix_b_mean_plus_std`: (avg + stddev) color information in the cervix (B channel).
28. `rgb_total_r_mean`: average color information in the image (B channel).
29. `rgb_total_r_std`: stddev color information in the image (R channel).
30. `rgb_total_r_mean_minus_std`: (avg - stddev) color information in the image (R channel).
31. `rgb_total_r_mean_plus_std`: (avg + stddev) color information in the image (R channel).
32. `rgb_total_g_mean`: average color information in the image (G channel).
33. `rgb_total_g_std`: stddev color information in the image (G channel).
34. `rgb_total_g_mean_minus_std`: (avg - stddev) color information in the image (G channel).
35. `rgb_total_g_mean_plus_std`: (avg + stddev) color information in the image (G channel).
36. `rgb_total_b_mean`: average color information in the image (B channel).
37. `rgb_total_b_std`: stddev color information in the image (B channel).
38. `rgb_total_b_mean_minus_std`: (avg - stddev) color information in the image (B channel).
39. `rgb_total_b_mean_plus_std`: (avg + stddev) color information in the image (B channel).
40. `hsv_cervix_h_mean`: average color information in the cervix (H channel).
41. `hsv_cervix_h_std`: stddev color information in the cervix (H channel).

42. hsv\_cervix\_s\_mean: average color information in the cervix (S channel).
43. hsv\_cervix\_s\_std: stddev color information in the cervix (S channel).
44. hsv\_cervix\_v\_mean: average color information in the cervix (V channel).
45. hsv\_cervix\_v\_std: stddev color information in the cervix (V channel).
46. hsv\_total\_h\_mean: average color information in the image (H channel).
47. hsv\_total\_h\_std: stddev color information in the image (H channel).
48. hsv\_total\_s\_mean: average color information in the image (S channel).
49. hsv\_total\_s\_std: stddev color information in the image (S channel).
50. hsv\_total\_v\_mean: average color information in the image (V channel).
51. hsv\_total\_v\_std: stddev color information in the image (V channel).
52. fit\_cervix\_hull\_rate: Coverage of the cervix convex hull by the cervix.
53. fit\_cervix\_hull\_total: Image coverage of the cervix convex hull.
54. fit\_cervix\_bbox\_rate: Coverage of the cervix bounding box by the cervix.
55. fit\_cervix\_bbox\_total: Image coverage of the cervix bounding box.
56. fit\_circle\_rate: Coverage of the cervix circle by the cervix.
57. fit\_circle\_total: Image coverage of the cervix circle.
58. fit\_ellipse\_rate: Coverage of the cervix ellipse by the cervix.
59. fit\_ellipse\_total: Image coverage of the cervix ellipse.
60. fit\_ellipse\_goodness: Goodness of the ellipse fitting.
61. dist\_to\_center\_cervix: Distance between the cervix center and the image center.
62. dist\_to\_center\_os: Distance between the cervical os center and the image center.
63. experts::0: subjective assessment of the Expert 0 (target variable).

64. experts::1: subjective assessment of the Expert 1 (target variable).
65. experts::2: subjective assessment of the Expert 2 (target variable).
66. experts::3: subjective assessment of the Expert 3 (target variable).
67. experts::4: subjective assessment of the Expert 4 (target variable).
68. experts::5: subjective assessment of the Expert 5 (target variable).
69. consensus: subjective assessment of the consensus (target variable).



## Appendix B

## R CODE

## B.1 EXPERIMENT 1

```
#####  
# Artificial Neural Networks prediction  
# Experiment 1: Quality Assessment of Digital Colposcopies  
# Data Set  
# This dataset explores the subjective quality assessment of  
# digital colposcopies.  
# The judgements of 6 experts and their consensus  
# (majority judgement) are included.  
# Experiment 1 predicts the consensus of individual  
# judgements (one at the time).  
# The consensus prediction is performed in two ways:  
# (i) on the majority binary value 0 or 1;  
# (ii) on a continuous value [0,1] computed as the  
# mean of the six experts judgements.  
#  
# Data Set Information:  
# * The dataset was acquired and annotated by professional  
# physicians at 'Hospital Universitario de Caracas'.  
# * The subjective judgments (target variables) were  
# originally done in an ordinal manner (poor, fair, good,  
# excellent) and was discretized in two classes (bad, good).  
# * Images were randomly sampled from the original colposcopic
```

```

#   sequences (videos).
#   * The original images and the manual segmentations are
#   included in the
#   'images' directory.
#   * The dataset has three modalities (i.e. Hinselmann,
#   Green, Schiller).
#   * The target variables are expert:: $X$  ( $X$  in  $0, \dots, 5$ )
#   and consensus.
#
#####
#set memory limit
memory.limit(6410241024*1024)

SOURCE <- 'data/DColposcopy/'
FILE1 <- 'green.csv'
FILE2 <- 'hinselmann.csv'
FILE3 <- 'schiller.csv'

file <- FILE3      #the data file currently used
data.source <- read.csv(paste(SOURCE, file, sep = ''),
header = TRUE)
# add a consensus mean value: a continuous value in [0,1]
# representing
# the mean of the judgement of all experts.
data.source$consensus <- rowMeans(data.source[,63:68])

```

```

#str(data.source)

#normalization
# Min-Max Normalization
normidx <- 1:62 #the predictors to be normalized
#data.source[,normidx]<- (data.source[,normidx]-min
#(data.source[,normidx]))/(max(data.source[,normidx])
#-min(data.source[,normidx]))

#####
#
# Create aa ANN and test prediction(s)
#
#####

#Data-partition
seeds <- c(123, 4721, 3097, 5326, 9271, 2870, 6901,
7751, 8292, 5028)

tf <- 0.8 #the training fraction (typically 80%
training, 20% testing)
trials <- 1:10
for (trial in trials) {

```

```

set.seed(seeds[trial])
ind <- sample(nrow(data.source), tf * nrow(data.source))
data.training <- data.source[ind,]
data.testing <- data.source[-ind,]

#Create and train an Artificial Neural Network with h
# hidden layers
library(neuralnet)
set.seed(123)
#ann parameters
h <- c(64,128,4) #hidden layers
data.pred <- 1:62 #the predictors
data.tar <- 68 #the target variable(s)
pflhst <- colnames(data.source)[data.tar]
#the prediction formula left-hand-side terms
pfrhst <- colnames(data.source)[data.pred]
#the prediction formula right-hand-side terms
pfl <- paste(pflhst, collapse = '+')
pfr <- paste(pfrhst, collapse = '+')
sigmoid = function(x) {
  1 / (1 + exp(-x/20))
}
#ann model
ann.formula <- as.formula(paste(pfl, '~', pfr))
ann.model <- neuralnet(ann.formula,
  data = data.training,

```

```

        hidden = h,
        err.fct = "ce",
        #threshold = 0.01, #default 0.01
        #stepmax = 1e+05, #default 1e+05
        #rep = 1,          #default 1
        #act.fct = sigmoid, #default "logistic"
        linear.output = F)
#plot(ann.model)

# Use the ANN model for Prediction
ann.output<- compute(ann.model, data.testing[,data.pred])

# Display the results
print(paste('Dataset: ', file, sep = ' '))
#print('Original data:')
#print(data.testing[1,data.tar])
#print('Predicted data:')
#print(ann.output$net.result[1])

# Compute and Show the confusion matrix

#the target index
t <- 1
res <- data.frame(orig = ifelse(data.testing[,data.tar[t]]
>= 0.5, 1, 0), pred = ifelse(ann.output$net.result >= 0.5, 1, 0))
#print(paste0(' ', pflhst[t]))

```

```

print(paste0('Expert ',(data.tar -62)))
print(paste0('Layers: ', paste(h, collapse = ', ')))
print(table(res$orig , res$pred , dnn = c("Original", "Predicted")))

print(paste0("Accuracy: ", (sum(res$orig == res$pred) / nrow(res))))
}
#gwplot(ann.model, selected.covariate = 4, selected.response = 1)

```

## B.2 EXPERIMENT 2

```

#####
# Artificial Neural Networks prediction
# Experiment 2: Quality Assessment of Digital Colposcopies
# Data Set
# This dataset explores the subjective quality assessment
# of digital colposcopies.
# The judgements of 6 experts and their consensus
# (majority judgement) are included.
# Experiment 2 predicts the judgements for all experts at
# the same time (using one model).
#
# Data Set Information:
# * The dataset was acquired and annotated by professiona
# physicians at 'Hospital Universitario de Caracas'.
# * The subjective judgments (target variables) were originally
# done in an ordinal manner (poor, fair, good, excellent) and
# was discretized in two classes (bad, good).

```

```

# * Images were randomly sampled from the original colposcopic
# sequences (videos).
# * The original images and the manual segmentations are included
# in the 'images' directory.
# * The dataset has three modalities (i.e. Hinselmann, Green,
# Schiller).
# * The target variables are expert::X (X in 0,...,5) and
# consensus.
#
#####
#set memory limit
memory.limit(6410241024*1024)

SOURCE <- 'data/DColposcopy/'
FILE1 <- 'green.csv'
FILE2 <- 'hinselmann.csv'
FILE3 <- 'schiller.csv'

METHOD1 <- 'Hinselmann'
METHOD2 <- 'Schiller'
METHOD3 <- 'Green'

METHOD <- METHOD2

file <- paste0(tolower(METHOD), '.csv')
#the data file currently used

```

```

data.source<- read.csv(paste(SOURCE, file , sep = ''),
header = TRUE)
#str(data.source)

NCOL <- ncol(data.source)
NPRED <- 6 + 1

#####
#
# Create aa ANN and test prediction(s)
#
#####

#Data-partition
seeds <- c(123, 4721, 3097, 5326, 9271, 2870, 6901, 7751,
8292, 5028)

tf <- 0.8 #the training fraction (typically 80%
training , 20% testing)
trials <- 1:10
for (trial in trials) {
  set.seed(seeds[trial])
  ind <- sample(nrow(data.source), tf * nrow(data.source))
  data.training <- data.source[ind,]

```





```

        linear.output = F)
#plot(ann.model)

# Use the ANN model for Prediction
ann.output<- compute(ann.model, data.testing[,data.pred])

# Display the results
print(paste('Dataset: ', file, sep = ' '))
#print('Original data:')
#print(data.testing[1,data.tar])
#print('Predicted data:')
#print(ann.output$net.result[1,])

# Compute and Show the confusion matrix for each target variable
for (t in 1:length(data.tar)) {
  res <- data.frame(orig = data.testing[,data.tar[t]],
                    pred = ifelse(ann.output$net.result[,t]
                                   >= 0.5, 1, 0))
  print(paste0('Expert ',t))
  #print(paste0('Confusion Table for ', pflhst[t]))
  print(table(res$orig, res$pred, dnn = c("Original",
    "Predicted")))

  print(paste0("Accuracy:~", (sum(res$orig == res$pred)
    / nrow(res))))
}

```

}

### B.3 EXPERIMENT 2

```
#####
# Artificial Neural Networks prediction
# Experiment 2: Quality Assessment of Digital Colposcopies
# Data Set
# This dataset explores the subjective quality assessment
# of digital colposcopies.
# The judgements of 6 experts and their consensus
# (majority judgement) are included.
# Experiment 2 predicts the judgements for all experts at
# the same time (using one model).
#
# Data Set Information:
# * The dataset was acquired and annotated by professiona
# physicians at 'Hospital Universitario de Caracas'.
# * The subjective judgments (target variables) were originally
# done in an ordinal manner (poor, fair, good, excellent) and
# was discretized in two classes (bad, good).
# * Images were randomly sampled from the original colposcopic
# sequences (videos).
# * The original images and the manual segmentations are included
# in the 'images' directory.
# * The dataset has three modalities (i.e. Hinselmann, Green,
```

```

# Schiller ).
# * The target variables are expert::X (X in 0,...,5) and
# consensus.
#####

#set memory limit
memory.limit(6410241024*1024)

SOURCE <- 'data/DColposcopy/'
FILE1 <- 'green.csv'
FILE2 <- 'hinselmann.csv'
FILE3 <- 'schiller.csv'

file <- FILE3      #the data file currently used
data.source<- read.csv(paste(SOURCE, file , sep = ''),header = TRUE)
#str(data.source)

#normalization
# Min-Max Normalization
normidx <- 1:62 #the predictors to be normalized
#data.source[,normidx]<- (data.source[,normidx]-min(data.source
#[,normidx]))/(max(data.source[,normidx])-min(data.source[,normidx]))
#####

```

```

#
# Create an ANN and test prediction(s)
#
#####

#Data-partition
set.seed(123)

votes <- c()
for (r in 1:nrow(data.source)) {
  votes[r] <- sum(data.source[r,63:68])
}

ind <- which(votes == 3)
data.training <- data.source[-ind,]
data.testing <- data.source[ind,]

#Create and train an Artificial Neural Network with h hidden layers
library(neuralnet)
set.seed(123)
#ann parameters
h <- c(64,32,4) #hidden layers
data.pred <- 1:62 #the predictors

```

```

data.tar <- 69 #the target variable(s)
pflhst <- colnames(data.source)[data.tar]
#the prediction formula left-hand-side terms
pfrhst <- colnames(data.source)[data.pred]
#the prediction formula right-hand-side terms
pfl <- paste(pflhst, collapse = '+')
pfr <- paste(pfrhst, collapse = '+')
sigmoid = function(x) {
  1 / (1 + exp(-x/20))
}
#ann model
ann.formula <- as.formula(paste(pfl, '~', pfr))
ann.model <- neuralnet(ann.formula,
  data = data.training,
  hidden = h,
  err.fct = "ce",
  threshold = 0.01, #default 0.01
  stepmax = 1e+05, #default 1e+05
  rep = 2, #default 1
  #act.fct = sigmoid, #default "logistic"
  linear.output = F)
#plot(ann.model)

# Use the ANN model for Prediction
ann.output <- compute(ann.model, data.testing[,data.pred])

```

```

# Display the results
print(paste('Dataset: ', file , sep = ' '))
#print('Original data:')
#print(data.testing[1, data.tar])
#print('Predicted data:')
#print(ann.output$net.result[1])

# Compute and Show the confusion matrix

#the target index
t <- 1
res <- data.frame(orig = data.testing[, data.tar[t]],
                  pred = ifelse(ann.output$net.result
                               >= 0.5, 1, 0))
#print(paste0('Confusion Table for ', pflhst[t]))
print("Expert_7")
print(paste0('Layers: ', paste(h, collapse = ', ')))
print(table(res$orig, res$pred, dnn = c("Original",
"Predicted")))

print(paste0("Accuracy: ", (sum(res$orig == res$pred)
/ nrow(res))))

#gwplot(ann.model, selected.covariate = 4,
#selected.response = 1)

```