Prediction of Human Development Index using Artificial Neural Networks

Prepared at

ISO 9001:2008
ISO 27001:20013

CMMI LEVEL-5

Bhaskaracharya Institute for Space Applications & Geo-informatics. Science & Technology Department, Govt. of Gujarat.

Gandhinagar

Prepared By

Rishab Parakh
2014A7PS114G

Ashutosh Saboo
2014B4A7427G

Guided By

Parth Trivedi
BISAG Gandhinagar
Project Coordinator

Submitted To

Birla Institute of Technology and Science

DEPARTMENT OF COMPUTER SCIENCE
BITS,PILANI – 333 031

Bhaskaracharya Institute for Space Applications and Geo-informatics

Department of Science & Technology

Government of Gujarat

Phone: 079 - 23213081 Fax: 079-23213091

E-mail: info@bisag.gujarat.gov.in, website: www.bisag.gujarat.gov.in

ISO 9001:2008
ISO 27001:20013
CMMI LEVEL-5

CERTIFICATE

This is to certify that the project report compiled by ***Mr. Rishab Parakh and Ashutosh Saboo****students of 4th Semester* ***B.E.(Hons.) from Department Of Computer Science, Birla Institute of Technology and Science,*** *have completed their Practice School-1 project satisfactorily. To the best of our knowledge this is an original and bonafide work done by them. They have worked on a software application for* ***"Prediction of Human Development Index using Artificial Neural Networks"****, starting from May 23rd, 2016 to July 16th, 2016.*

During their tenure at this Institute, they were found to be sincere and meticulous in their work. We appreciate their enthusiasm & dedication towards the work assigned to them.

We wish them every success.

Parth Trivedi

Project Scientist

BISAG, Gandhinagar

T. P. Singh

Director

BISAG, Gandhinagar

Organization Profile

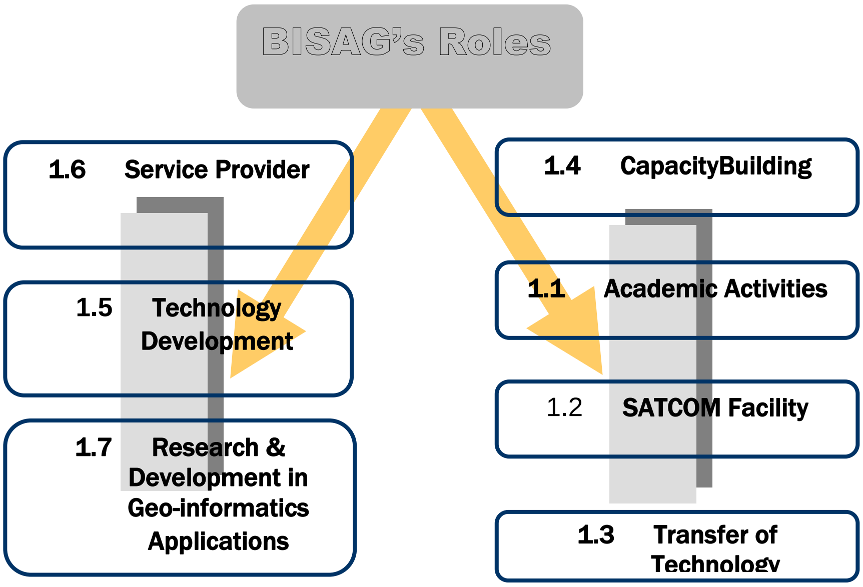
1. BACKGROUND

The applications of space technologies and geo-informatics contribute significantly towards socio-economic development of the society. Recognizing the importance and need of Space technology and geo-informatics for developmental planning purposes, the Government of Gujarat established the Bhaskaracharya Institute for Space Applications and Geo-informatics (BISAG) in the year 1997, as the State nodal agency to utilize space technology and geo-informatics for various developmental activities of the State.

Since its foundation, the Institute has experienced extensive growth in the spheres of space technology and geo-informatics. The objective with which BISAG was established is manifested in the extent of services its renders to almost all departments of the State. Year after year the institute has been endeavoring to increase its outreach to disseminate the use of geo-informatics up to grassroots level. In this span of eleven years, BISAG has assumed multi-dimensional roles and achieved several milestones to become an integral part of the development process of the GujaratState.

2. PROFILE

BISAG's has strengthened its role as a facility provider, a technology developer and as a facilitator for transferring technology to the grass root level.


BISAG's Roles
1.6 Service Provider
1.5 Technology Development
1.7 Research & Development in Geo-informatics Applications
1.4 CapacityBuilding
1.1 Academic Activities
1.2 SATCOM Facility
1.3 Transfer of Technology

Further reinforcing its functions, BISAG has achieved ISO 9001:2008 and ISO 27001:2005 certifications for quality management and security management services respectively. This has led to an organized and systematic development of its services and outputs.

3. ACTIVITIES OF BISAG

BISAG's activities are multi-fold and have expanded in a big way and focused on the following:

- **Satellite Communicatio**

  Promoting and facilitating the use of satellite broadcasting networks for distant interactive training, education and extensions

- **Remote Sensing**

  Inventory mapping, developmental planning and monitoring of natural and man-made resources

- **Geo-informatics Systen**

  Conceptualizing, creating and organizing multi-purpose common geo-spatial database for sectoral and thematicapplications for various users

- **Photogrammetry**

  Creation of Digital Elevation Model, Terrain characteristics, Resource planning,etc.

- **Global Navigation Satellite System**

  Location based services, geo-referencing, engineering applications and research

- **Software Developmen**

  For providing low-cost Decision Support Systems, desktop as well as web-based geo-informatics applications to users for wider usage.

- **Disaster Managemen**

  For preparing geo-spatial information to provide necessary inputs to the Government to assess and mitigate extent of damage in the event of a disaster

- **Education, Research and Training**

  For providing education, research and training facilities to promote number of end users through the Academy for Geo-informatics.

- **Value Added Services**

  For providing services which can be customized as per the needs of the users.

- **Technology Transfer**

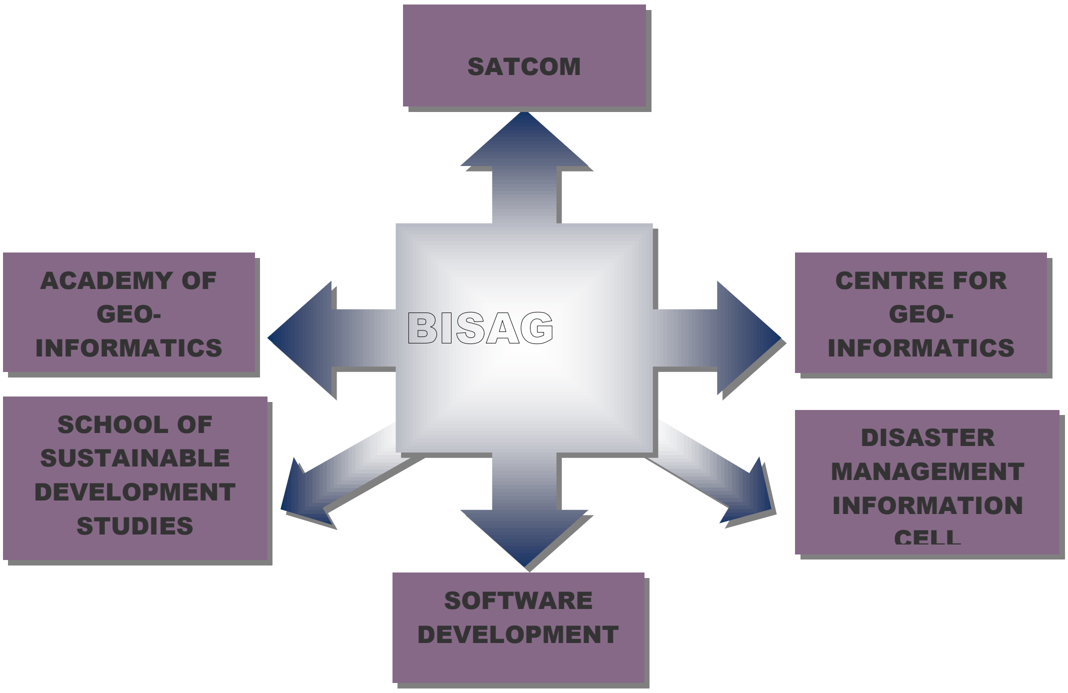
  Transferring technology to a large number of end users.

4. UNITS OF BISAG

BISAG initially set up to carry out Space Technology applications, has evolved into ***an Academic Institute, a Centre for Research and Technology Innovations, a Facility Provider, a Technology Developer and a Facilitato***r for transferring technology to the grass root level. BISAG is the first such State Centre having such multifarious activities with ISO certification. BISAG has gradually progressed over the years and has grown into several units. Each unit focuses on specific functions and objectives to ensure efficiency in over all activities of the institute.

- **Gujarat Satellite Communication Network (GUJSAT):**SATCOM facilitates the promotion and facilitation of the use of broadcast and teleconferencing networks for distant interactive training, education and extension.
- **Centre for Geo-Informatics Applications:** The Centre for Geo-informatics provides services for the developmental and planning activities pertaining to Agriculture, Land and Water Resources Management, Wasteland/ Watershed development, Forestry, Disaster Management, and Infrastructure etc.
- **Software Development:** For wider usage of geo-spatial applications, customised software are developed by the Software Development Team. The institute has provided many indigenous software solutions in the field of Geographic Information Systems, Decision Support Systems and Image Processing.
- **Academy of Geo-informatics:** The Academy for Geo-informatics carries out Education, Research and Training activities.
- **Disaster Management Information cell:** BISAG works closely with the Gujarat State Disaster Management Authority (GSDMA), for assessment of existing situation through integrated analysis and for planning appropriate preventive

and preparatory measures, providing necessary support through data generation and analysis.


SATCOM
BISAG
ACADEMY OF GEO-INFORMATICS
CENTRE FOR GEO-INFORMATICS
SCHOOL OF SUSTAINABLE DEVELOPMENT STUDIES
DISASTER MANAGEMENT INFORMATION CELL
SOFTWARE DEVELOPMENT

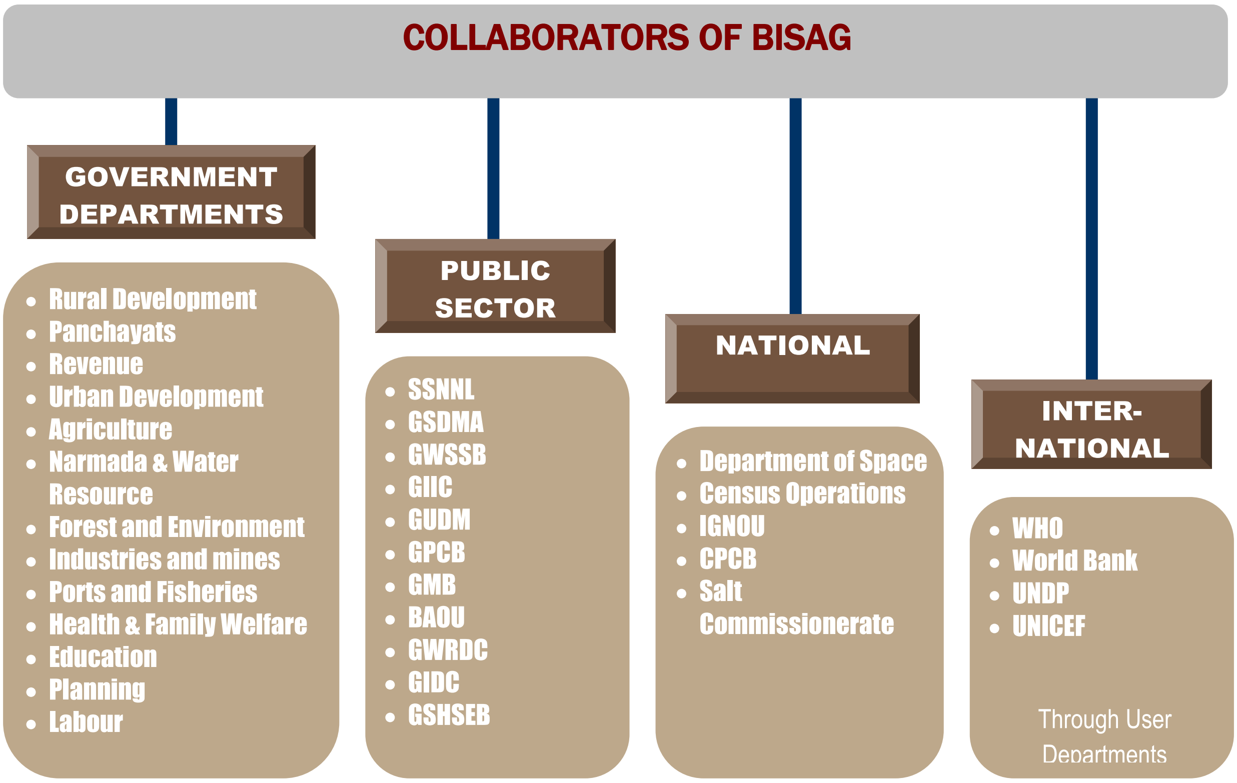
5. INFRASTRUCTURE DEVELOPMENT

The growth and progress of any institute is gauged by the infrastructure it develops and possesses. BISAG has a sound infrastructure setup that has developed in tandem with the growth of the institute. Having started with one building, there are now dedicated facilities for different units.

1.1 The laboratories are equipped with state-of the art technology with latest Hardware and Software required for executing its activities. BISAG also has a rich satellite data archive, which includes Satellite data of different spatial, spectral and temporal resolutions

6. COLLABORATIONS OF BISAG.....Creating A Sense Of Ownership

BISAG works with almost all Government Departments and Organizations. Each of these Departments/Organization contributes in preparation of the respective projects. With strong Government support and proactive efforts on part of the staff of BISAG, the list of Collaborators is expanding and increasing.



Partnerships and Collaborations by BISAG

7. INSTITUTIONAL STRENGTHENING

BISAG has achieved institutional strengthening through:

- **Reinforcement of Decision Support Systems**

Developing customized solutions as per user requirements through partnerships and collaborations, which are affordable and easy to use. Areas of natural and manmade resources, socio-economic parameters, are being effectively addressed with the help of Geo-informatics.

- **Establishing Linkage between Government and People through GUJSAT**

GUJSAT facility is being constantly employed for the promotion and facilitation of the use of teleconferencing networks for distant interactive training, education and extension. Experts, leaders, specialists and professionals can conduct their programs from a central location reaching out to remote areas through two-way audio-video channel making them interactive and meaningful.

- **Developing Innovative Education Programmes**

Innovative educational programmes are conducted regularly through GUJSAT, allowing people residing in remote areas to have an access to good quality educational and awareness programmes.

- **Solving real life problems through Human Resource Development**

The institute has a young multi-disciplinary team of professionals and a continuing induction programme. Multi-nationals and IT agencies pick up the trained staff that in turn is replaced by new people. This results in availability of more and more trained manpower in the realm of space applications. Every year BISAG provides training to about 300 students in the field of Geo-informatics.

- **Creation of the multipurpose sectoral comprehensive databases for the entire state of Gujarat**

The institute has made efforts towards conceptualization, creation and organization of multi-purpose common digital database for sectoral / integrated decision support systems. This has provided impetus to planning and developmental activities at grass root level as well as monitoring and management potential in various disciplines like water resources, land resources, disaster management, infrastructure, urban management.

CANDIDATE'S DECLARATION

We declare that Practice School-1 report entitled "PREDICTION OF HUMAN DEVELOPMENT INDEX USING ARTIFICIAL NEURAL NETWORKS" is our own work conducted under the supervision of our guide Mr. Parth Trevidi from BISAG (Bhaskaracharya Institute for Space Applications & Geo-informatics).We further declares that to the best of our knowledge the report for B.E. (Hons.) Practice School-1 does not contain part of the work which has been submitted for the award of Bachelor's Degree either in this or any other university without proper citation.

Candidate 1's Signature
Rishab Parakh
Student ID: 2014A7PS114G

Candidate 2's Signature
Ashutosh Saboo
Student ID: 2014B4A7427G

Submitted To:
Department Of Computer Science
BITS Pilani, Goa Campus

ACKNOWLEDGMENT

We are grateful to **Mr. T.P. Singh**, **Director (BISAG)** for giving us this opportunity to work the guidance of renowned people of the field of GIS Based Portal and also providing us with the required resources in the institution.

We would like to express our endless thanks to our external guide **Mr. Parth Trevidi**, Project Scientist at Bhaskaracharya Institute of Space Applications and Geo-informatics for their sincere and dedicated guidance throughout the project development.

At this juncture I feel deeply honored in expressing my sincere thanks to **Mr.Viraj Choksi** of Bhaskaracharya Institute of Space Applications and Geo-informatics for making the resources available at right time and providing valuable insights leading to the successful completion of my project.

We would also like to express our hearty gratitude to our Head of Department **Mr. Bharat Deshpande** and our PS faculty **Mr. Gavax Joshi** for giving us encouragement and technical support on the project.

Rishab Parakh

Student ID: 2014A7PS114G

Ashutosh Saboo

Student ID: 2014B4A7427G

PROJECT INDEX

Sr No.	Topics	Page Number
1	Introduction	12
1.1	Human Development Index	12
1.3	Artificial Neural Network	13
2	System Requirement Study	14
2.1	Octave	14
2.2	Python	15
3	Methodology	16
3.1	Setting up Dependencies	16
3.2	Prediction Algorithm Used	19
3.3	Training Process of ANN	21
3.4	Our ANN Model for predicting HDI	24
3.5	Images illustrating our Artificial Neural Network at work	25
4	Conclusion	27
5	Bibliography	28

Chapter 1: Introduction

Human Development Index

Human development is defined as the process of enlarging people's freedoms and opportunities and improving their well-being. Countries in today's world are developing faster than they ever have in the past. The concept of human development was devised by Pakistani economist Mahbub ul Haq in 1990. In this rat race for development there comes a need to measure and compare the development that nations try to achieve. Human Development Index is a measure of the economic and social development of a country. The Human Development Index takes into account three important criteria of development to classify countries into four different groups:

1. Health: Life expectancy at birth is used to measure the health and access to health care facilities of the people living in the country. It uses a naive assumption that healthier people live longer.
2. Education: Literacy rate is used to measure how educated the people of a nation are. Two parameters are used to measure literacy: Mean Years of Schooling and Expected Years of Schooling,
3. Standard of living: This is measured by calculating the country's gross domestic product with the total population to account for the difference in population of various nations.

By including Health and Education to the index Haq shifted the notion of development from traditionally being totally economic to social. He argues that the country's economy does not reflect the well-being of the people of the country.

The first Human Development Index was presented in the Human Development Report by the United Nations Development Program in 1990. Now Human Development Index is presented every year ranking almost every country in the world.

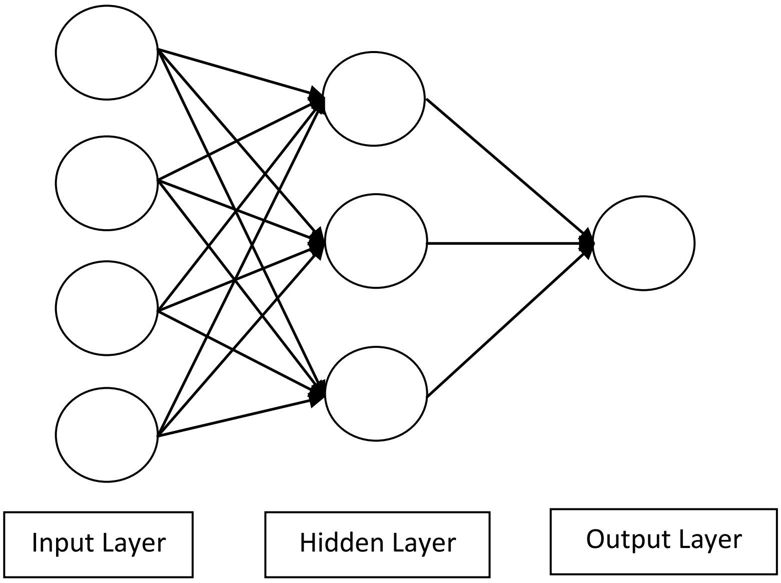
Artificial Neural Networks

Artificial Neural Networks, also known as Neural Networks or Neural Nets are effective Machine Learning tools that are designed to process information in such a way that it is similar to the way the human brain processes Information. It is inspired by the studies of the ability of the human brain to learn from observation and generalization from abstraction. A Human brain is a massive collection of about 10^{11} neurons each firing signals as we think, read, write or even breathe. Similar to this, in an Artificial Neural Network, artificial neurons or nodes are structured and connection in hierarchical manner.

All Artificial Neural Networks have a distinct input and a distinct output layer connected by one or many "hidden" layer(s). The strength of each connection between two neurons is represented by a numeric value called weight. These weights correspond to the decision boundary obtained by the Artificial Neural Network classifier.

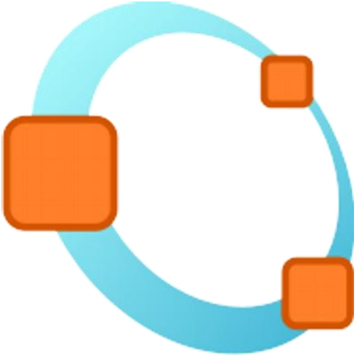
When an Artificial Neural Network is given a set of input-output values, it trains, that is it sets the weights between the nodes to match the predicted value of the Artificial Neural Network with the actual output. It does so by using various learning algorithms. Once the estimated values of the weights are stabilized after validation, trained Artificial Neural Network is tested against a test set to evaluate its predictive power. The trained Artificial Neural Network can then be used to make predictions on the input values.

On the right is a representation of an Artificial Neural Network which has four input units, one hidden layer which has three units and one final output layer. All the connections have some weight which is calculated during the training phase.



Chapter 2: System Requirement Study

2.1 Octave



GNU Octave is software featuring a high-level programming language, primarily intended for numerical computations. Octave helps in solving linear and nonlinear problems numerically, and for performing other numerical experiments using a language that is mostly compatible with MATLAB. It may also be used as a batch-oriented language.

Octave is one of the major free alternatives to MATLAB, another being Scilab. Scilab however puts less emphasis on (bidirectional) syntactic compatibility with MATLAB than Octave does.

The Octave language is an interpreted programming language. It is a structured programming language (similar to C) and supports many common C standard library functions, and also certain UNIX system calls and functions.

2.2 Python

Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java. The language provides constructs intended to enable clear programs on both a small and large scale.

Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.

Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems. Using third-party tools, such as Py2exe or Py-installer, Python code can be packaged into stand-alone executable programs for some of the most popular operating systems, so Python-based software can be distributed to, and used on, those environments with no need to install a Python interpreter.

Chapter 3: Methodology

3.1 Setting up Dependencies

- **NumPy 1.10:** NumPy (pronounced "Numb Pie" or sometimes "Numb pee") is an extension to the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open source and has many contributors.

- **Tkinter**: Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit and is Python's de facto standard GUI, and is included with the standard Microsoft Windows and Mac OS X install of Python. The name Tkinter comes from Tk interface. Tkinter was written by Fredrik Lundh. As with most other modern Tk bindings, Tkinter is implemented as a Python wrapper around a complete Tcl interpreter embedded in the Python interpreter. Tkinter calls are translated into Tcl commands which are fed to this embedded interpreter, thus making it possible to mix Python and Tcl in a single application.

- **Beautiful Soup** : Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favourite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work. Beautiful Soup is a Python library designed for quick turnaround projects like screen-scraping. Three features make it powerful:

- Beautiful Soup provides a few simple methods and *Pythonic* idioms for navigating, searching, and modifying a parse tree: a toolkit for dissecting a document and extracting what you need. It doesn't take much code to write an application
- Beautiful Soup automatically converts incoming documents to Unicode and outgoing documents to UTF-8. You don't have to think about encodings, unless the document doesn't specify an encoding and Beautiful Soup can't detect one. Then you just have to specify the original encoding.
- Beautiful Soup sits on top of popular Python parsers like lxml and html5lib, allowing you to try out different parsing strategies or trade speed for flexibility.

- **Mechanize**: A very useful python module for navigating through web forms is Mechanize. Mechanize Browser implements the urllib2.OpenerDirector interface. Browser objects have state, including navigation history, HTML form state, cookies, etc. The set of features and URL schemes handled by Browser objects is configurable. The library also provides an API that is mostly compatible with urllib2: your urllib2 program will likely still work if you replace "urllib2" with "mechanize" everywhere. Features include: ftp:, http: and file: URL schemes, browser history, hyperlink and HTML form support, HTTP cookies, HTTP-EQUIV and Refresh, Referrer [sic] header, robots.txt, redirections, proxies, and Basic and Digest HTTP authentication.

- **Selenium**: Selenium Python bindings provide a simple API to write functional/acceptance tests using Selenium WebDriver. Through Selenium Python API you can access all functionalities of Selenium WebDriver in an intuitive way. Selenium Python bindings provide a convenient API to access Selenium WebDrivers like Firefox, Ie, Chrome, Remote etc. The current supported Python versions are 2.7, 3.2 and above.

- **Urllib2**: urllib2 is a Python module for fetching URLs (Uniform Resource Locators). It offers a very simple interface, in the form of the urlopen function. This is capable of fetching URLs using a variety of different protocols. It also offers a slightly more complex interface for handling common situations - like basic authentication, cookies, proxies and so on. These are provided by objects called handlers and openers.
  urllib2 supports fetching URLs for many "URL schemes" (identified by the string before the ":" in URL - for example "ftp" is the URL scheme of "ftp://python.org/") using their associated network protocols (e.g. FTP, HTTP). This tutorial focuses on the most common case, HTTP.
  For straightforward situations urlopen is very easy to use. But as soon as you encounter errors or non-trivial cases when opening HTTP URLs, you will need some understanding of the HyperText Transfer Protocol. The most comprehensive and authoritative reference to HTTP is RFC 2616.

- **Requests**: The Requests module is a an elegant and simple HTTP library for Python. Requests allow you to send HTTP/1.1 requests. You can add headers, form data, multipart files, and parameters with simple Python dictionaries, and access the response data in the same way.

- **OpenPyXL**: Openpyxl is a Python library for reading and writing Excel 2010 xlsx/xlsm/xltx/xltm files. It was born from lack of existing library to read/write natively from Python the Office Open XML format.

3.2 Prediction Algorithm Used

Artificial Neural Networks (ANNs) are processing devices (algorithms or actual hardware) that are loosely modeled after the neuronal structure of the mammalian cerebral cortex but on much smaller scales. A large ANN might have hundreds or thousands of processor units, whereas a mammalian brain has billions of neurons with a corresponding increase in magnitude of their overall interaction and emergent behavior. Although ANN researchers are generally not concerned with whether their networks accurately resemble biological systems, some have. For example, researchers have accurately simulated the function of the retina and modeled the eye rather well. Although the mathematics involved with neural networking is not a trivial matter, a user can rather easily gain at least an operational understanding of their structure and function. Neural networks are typically organized in layers. Layers are made up of a number of interconnected 'nodes' which contain an 'activation function'. Patterns are presented to the network via the 'input layer', which communicates to one or more 'hidden layers' where the actual processing is done via a system of weighted 'connections'. The hidden layers then link to an 'output layer' where the answer is output. Most ANNs contain some form of 'learning rule' which modifies the weights of the connections according to the input patterns that it is presented with. In a sense, ANNs learn by example as do their biological counterparts; a child learns to recognize dogs from examples of dogs.

Although there are many different kinds of learning rules used by neural networks, this demonstration is concerned only with one; the delta rule. The delta rule is often utilized by the most common class of ANNs called 'backpropagational neural networks' (BPNNs). Backpropagation is an abbreviation for the backwards propagation of error.

With the delta rule, as with other types of backpropagation, 'learning' is a supervised process that occurs with each cycle or 'epoch' (i.e. each time the network is presented with a new input pattern) through a forward activation flow of outputs, and the backwards error propagation of weight adjustments. More simply, when a neural network is initially presented with a pattern it makes a random 'guess' as to what it might be. It then sees how far its answer was from the actual one and makes an appropriate adjustment to its connection weights. Note also, that within each hidden layer node is a sigmoidal activation function which polarizes network activity and helps it to stabilize. Backpropagation performs a gradient descent within the solution's vector space towards a 'global minimum' along the steepest vector of the error surface. The global minimum is that theoretical solution with the lowest possible error. The error surface itself is a hyper paraboloid but is seldom 'smooth' as is depicted in the graphic below. Indeed, in most problems, the solution space is quite irregular with numerous 'pits' and 'hills' which may cause the network to settle down in a 'local minima' which is not the best overall solution.

Since the nature of the error space can not be known a prior, neural network analysis often requires a large number of individual runs to determine the best solution. Most learning rules have built-in mathematical terms to assist in this process which control the 'speed' (Beta-coefficient) and the 'momentum' of the learning. The speed of learning is actually the rate of convergence between the current solution and the global minimum. Momentum helps the network to overcome obstacles (local minima) in the error surface and settle down at or near the global minimum.

Once a neural network is 'trained' to a satisfactory level it may be used as an analytical tool on other data. To do this, the user no longer specifies any training runs and instead allows the network to work in forward propagation mode only. New inputs are presented to the input pattern where they filter into and are processed by the middle layers as though training were taking place, however, at this point the output is retained and no backpropagation occurs. The output of a forward propagation run is the predicted model for the data which can then be used for further analysis and interpretation. It is also possible to over-train a neural network, which means that the network has been trained exactly to respond to only one type of input; which is much like rote memorization. If this should happen then learning can no longer occur and the network is referred to as having been "grand mothered" in neural network jargon. In real-world applications this situation is not very useful since one would need a separate grand mothered network for each new kind of input.

3.3 Training Process of ANN

Learning in a neural network is called training. Like training in athletics, training in a neural network requires a coach, someone that describes to the neural network what it should have produced as a response. From the difference between the desired response and the actual response, the error is determined and a portion of it is propagated backward through the network. At each neuron in the network the error is used to adjust the weights and threshold values of the neuron, so that the next time, the error in the network response will be less for the same inputs.

This corrective procedure is called backpropagation (hence the name of the neural network) and it is applied continuously and repetitively for each set of inputs and corresponding set of outputs produced in response to the inputs. This procedure continues so long as the individual or total errors in the responses exceed a specified level or until there are no measurable errors. At this point, the neural network has learned the training material and you can stop the training process and use the neural network to produce responses to new input data.

Backpropagation starts at the output layer with the following process:

For the ith input of the jth neuron in the output layer, the weight w_{ij} is adjusted by adding to the previous weight value, w'_{ij}, a term determined by the product of a learning rate, LR, an error term, e_j, and the value of the ith input, X_i. The error term, e_j, for the jth neuron is determined by the product of the actual output, Y_j, its complement, $1 - Y_j$, and the difference between the desired output, d_j, and the actual output.

Once the error terms are computed and weights are adjusted for the output layer, the values are recorded and the next layer back is adjusted. The same weight adjustment process is followed, but the error term is generated by a slightly modified version of Back Propagation.

In this version, the difference between the desired output and the actual output is replaced by the sum of the error terms for each neuron, k, in the layer immediately succeeding the layer being processed (remember, we are going backwards through the layers so these terms have already been computed) times the respective pre-adjustment weights.

The learning rate, LR, applies a greater or lesser portion of the respective adjustment to the old weight. If the factor is set to a large value, then the neural network may learn more quickly, but if there is a large variability in the input set then the network may not learn very well or at all. In real terms, setting the learning rate to a large value is analogous to giving a child a spanking, but that is inappropriate and counter-productive to learning if the offense is so simple as forgetting to tie their shoelaces. Usually, it is better to set the factor to a small value and edge it upward if the learning rate seems slow.

This is similar to Equation 3, with a momentum factor, M, the previous weight, w'ij, and the next to previous weight, w''_{ij}, included in the last term. This extra term allows for momentum in weight adjustment. Momentum basically allows a change to the weights to persist for a number of adjustment cycles. The magnitude of the persistence is controlled by the momentum factor. If the momentum factor is set to 0, then the equation reduces to that of Equation 3. If the momentum factor is increased from 0, then increasingly greater persistence of previous adjustments is allowed in modifying the current adjustment. This can improve the learning rate in some situations, by helping to smooth out unusual conditions in the training set.

As you train the network, the total error, that is the sum of the errors over all the training sets, will become smaller and smaller. Once the network reduces the total error to the limit set, training may stop. You may then apply the network, using the weights and thresholds as trained.

It is a good idea to set aside some subset of all the inputs available and reserve them for testing the trained network. By comparing the output of a trained network on these test sets to the outputs you know to be correct, you can gain greater confidence in the validity of the training. If you are satisfied at this point, then the neural network is ready for running.

Usually, no backpropagation takes place in this running mode as was done in the training mode. This is because there is often no way to be immediately certain of the desired response. If there were, there would be no need for the processing capabilities of the neural network! Instead, as the validity of the neural network outputs or predictions are verified or contradicted over time, you will either be satisfied with the existing performance or determine a need for new training. In this case, the additional input sets collected since the last training session may be used to extend and improve the training data.

3.4 Our ANN Model for predicting HDI

Artificial Neural Networks (ANN) offers a very intelligent and efficient way to perform tasks similar to the human brain. ANNs comprise of the input layer, the hidden layers, and the output layer which can be configured in various ways to generate a higher accuracy.

In our ANN model for predicting HDI, as part of the input layer, we generate all possible cube root, square root, single, and squared powers of <those 4>. In addition to this, we also add the logarithm of these terms to the input layer as well.

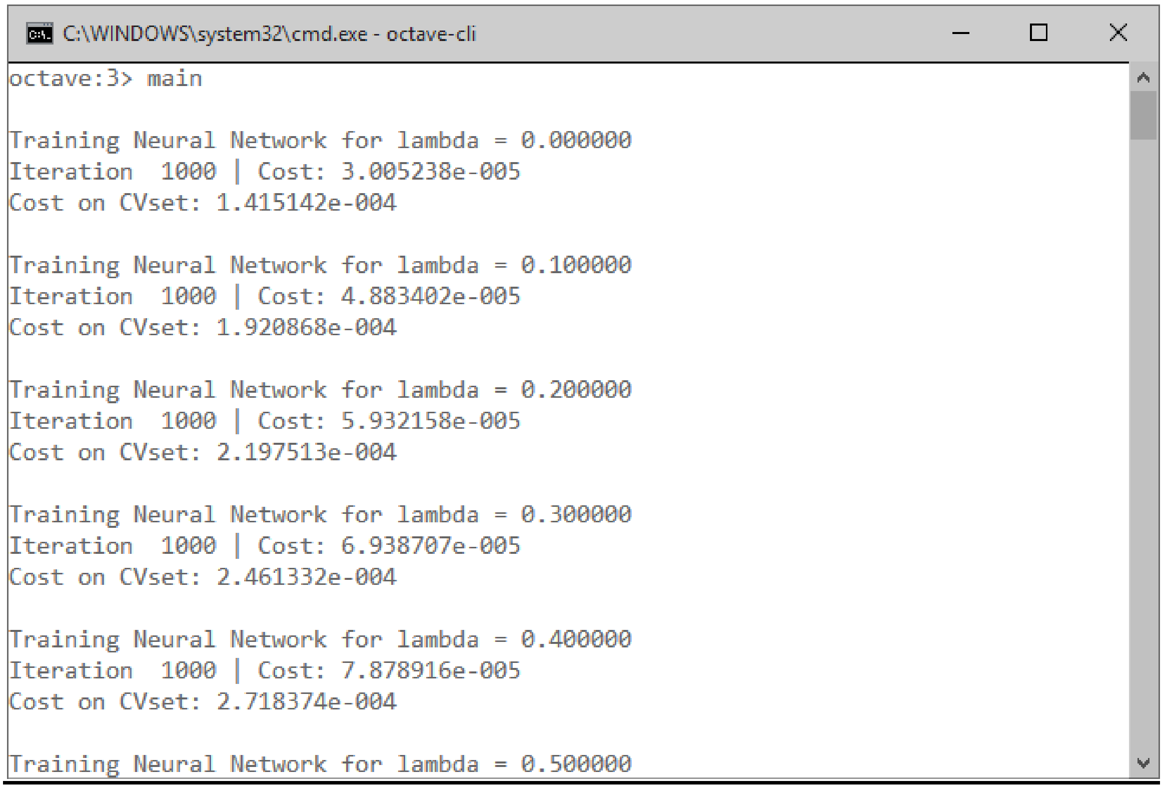
For the hidden layer, we set a fixed number of 100 hidden neurons as part of our hidden layer. These neurons in the hidden layer perform the necessary computation which help us in predicting the HDI values. As part of the ANN model, weights are assigned to the hidden layer, which in turn affect the values computed by the hidden layer neurons as well. Regularization is also implemented in the ANN, to prevent over-fitting of graphs on the training set, so as to prevent a very high error on the test set.

ANNs are provided with basically 3 sets of data - Training set, cross validation set, and the test set. The training set is used for the ANN to predict the hidden layer weights, based on the input data provided. The Cross Validation set is used to validate and set the regularization parameter, which gives the best results. Finally, the Test set is used to test the ANN with the initialized weights, and the regularization parameter to determine the final accuracy of the ANN.

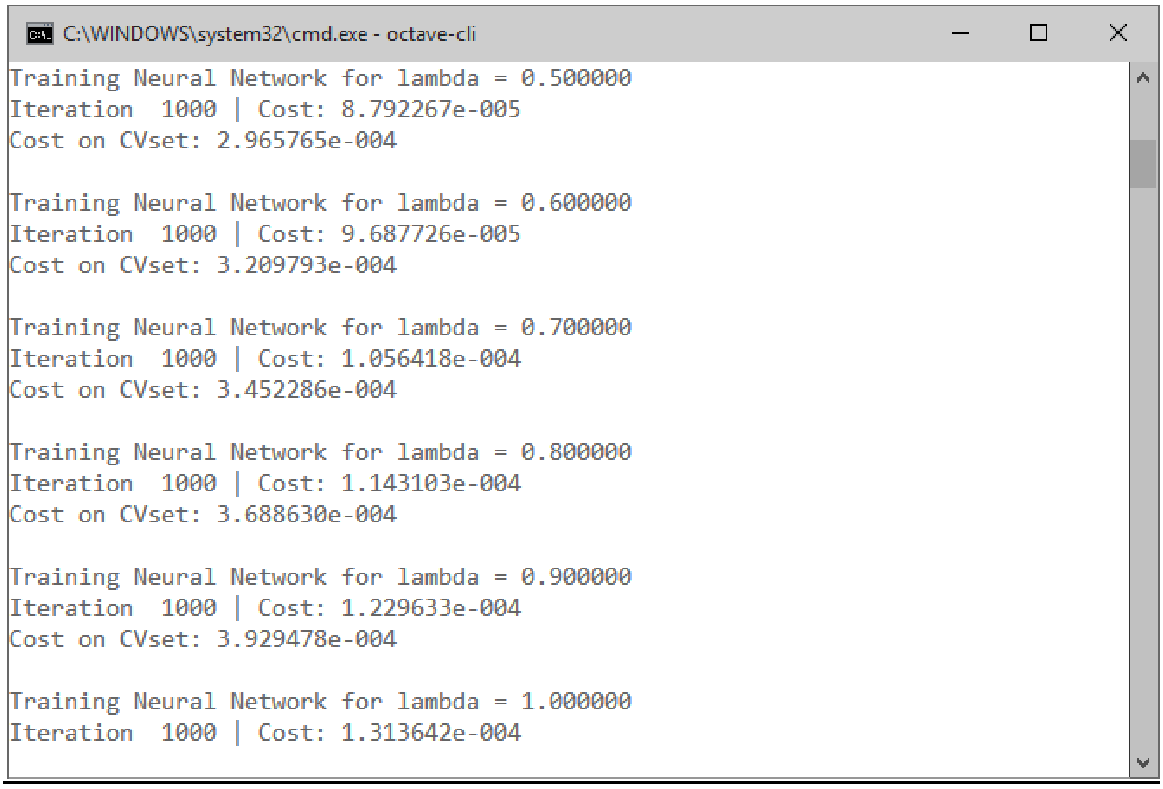
The hidden layer weights are initialized randomly, and subsequently get trained automatically by the feed-forward and back-propagation methods of the ANN, based on the input training data provided to the ANN.

As part of the output layer of our ANN model, we have a single neuron, whose value ranges from 0 to 1, and predicts the HDI values of countries, The difference between the actual HDI value and the predicted HDI value is the error in each test set provided to the ANN, and the accuracy of the ANN model is determined by the percentage of the test set errors with the range of -0.03 to 0.03.

3.5 Images illustrating our Artificial Neural Network at work

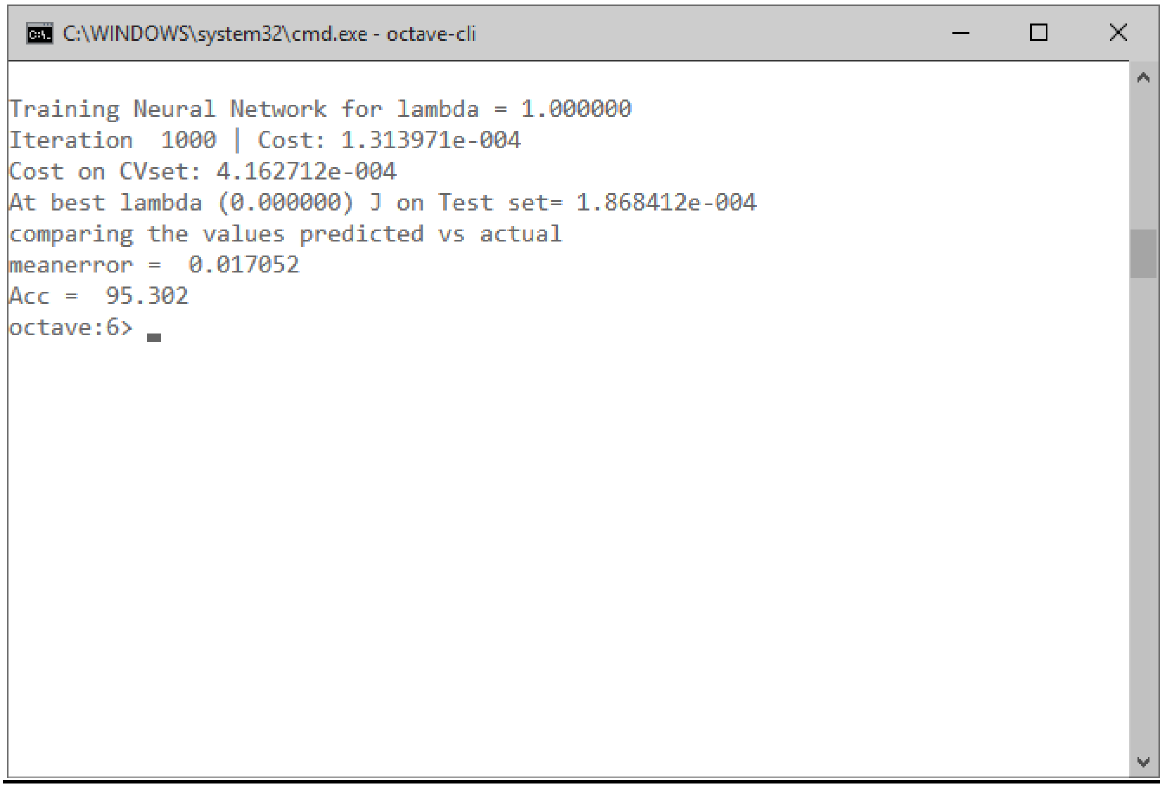


```
octave:3> main

Training Neural Network for lambda = 0.000000
Iteration  1000 | Cost: 3.005238e-005
Cost on CVset: 1.415142e-004

Training Neural Network for lambda = 0.100000
Iteration  1000 | Cost: 4.883402e-005
Cost on CVset: 1.920868e-004

Training Neural Network for lambda = 0.200000
Iteration  1000 | Cost: 5.932158e-005
Cost on CVset: 2.197513e-004

Training Neural Network for lambda = 0.300000
Iteration  1000 | Cost: 6.938707e-005
Cost on CVset: 2.461332e-004

Training Neural Network for lambda = 0.400000
Iteration  1000 | Cost: 7.878916e-005
Cost on CVset: 2.718374e-004

Training Neural Network for lambda = 0.500000
```

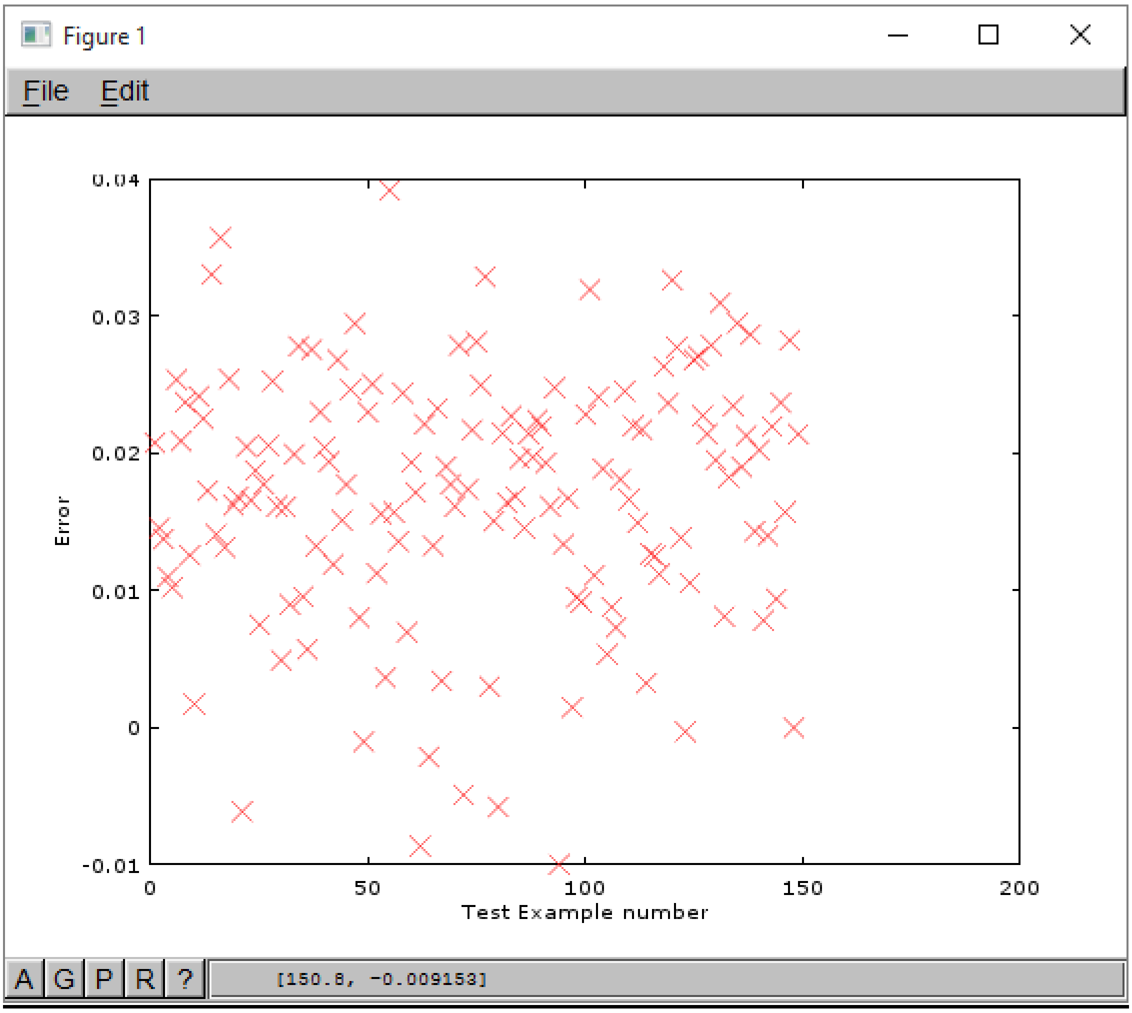


```
Training Neural Network for lambda = 0.500000
Iteration  1000 | Cost: 8.792267e-005
Cost on CVset: 2.965765e-004

Training Neural Network for lambda = 0.600000
Iteration  1000 | Cost: 9.687726e-005
Cost on CVset: 3.209793e-004

Training Neural Network for lambda = 0.700000
Iteration  1000 | Cost: 1.056418e-004
Cost on CVset: 3.452286e-004

Training Neural Network for lambda = 0.800000
Iteration  1000 | Cost: 1.143103e-004
Cost on CVset: 3.688630e-004

Training Neural Network for lambda = 0.900000
Iteration  1000 | Cost: 1.229633e-004
Cost on CVset: 3.929478e-004

Training Neural Network for lambda = 1.000000
Iteration  1000 | Cost: 1.313642e-004
```

-



```
Training Neural Network for lambda = 1.000000
Iteration  1000 | Cost: 1.313971e-004
Cost on CVset: 4.162712e-004
At best lambda (0.000000) J on Test set= 1.868412e-004
comparing the values predicted vs actual
meanerror =  0.017052
Acc =  95.302
octave:6> _
```



The final errors on out test set.

Chapter 4: Conclusion

We successfully modeled our Artificial Neural Network to predict the Human Development Index of any given country in the upcoming years, once the required input features are provided correctly to our program. Our program achieves a best-case accuracy of 95.3%.

The Artificial Neural Networks Algorithm that we implemented, can further be modified a bit, to predict the change in static parameters (which generally change only after a period of 10 years) as well, which are used by the United Nations, to calculate the Human Development Index.

All in all, our Neural Networks representation of predicting the HDI, helped us to predict the same, in a very accurate way, giving us error ranges majorly in the range of -0.03 to 0.03 only.

Chapter 5: Bibliography

- https://www.coursera.org/learn/machine-learning
- http://www.andrewng.org/
- www.cs.cmu.edu/~tom/10701_sp11/lectures.shtml
- https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html
- http://www.journals.elsevier.com/neural-networks
- http://neuralnetworksanddeeplearning.com/
- http://natureofcode.com/book/chapter-10-neural-networks/
- http://www.sciencedirect.com/science/journal/08936080
- https://github.com/encog
- http://www.heatonresearch.com/encog/
- http://scikit-learn.org/
- https://github.com/scikit-learn/scikit-learn
- https://arxiv.org/abs/1506.04776
- http://machinelearningmastery.com/a-gentle-introduction-to-scikit-learn-a-python-machine-learning-library/
- http://www.dm.unibo.it/~lenci/teaching/14/maa/octavetut.pdf
- http://ocw.mit.edu/courses/nuclear-engineering/22-15-essential-numerical-methods-fall-2014/tutorial-videos/octave-matlabae-for-beginners-part-1-starting-from-scratch/
- http://www.dm.unibo.it/~lenci/teaching/14/maa/octavetut.pdf
- http://www.opengardensblog.futuretext.com/wp-content/uploads/2014/03/OctaveTutorialAndrewNg.pdf
- https://www.gnu.org/software/octave/
- http://www.i-programmer.info/programming/other-languages/4779-a-programmers-guide-to-octave.html