Overview

Machine learning has become an integral part of every field which relies on past experience and a series of activities whose result needs to be improved in the upcoming product or service. For this purpose large chunks of data needs to be processed and useful trends/ patterns need to be extracted from it.

Machine learning has been implemented in all major software languages (C++, Java, Python, MATLAB), with each having their individual set of libraries to support and improve existing numerical analysis and computation.

This proposal aims to implement existing* machine learning algorithms in Scilab. All the above mentioned features will be made available to the user through a machine learning toolbox or through command line interface.

Project subdivided into 2 phases:
[1] all major machine learning algorithms implementation through Scilab code

Models working till now:
1. Linear regression
2. Logistic regression

Before 15 June:
3. Neural networks (model ready, datasets being selected)
4. K-means Clustering
5. Support Vector Machine
6. Naïve Bayes Classifier
7. Nearest Neighbors
8. Decision Trees

*more algorithms to be implemented based on mentor suggestions and discussion on mailing list during community bonding period
Important machine learning features (functions) added:
- Gradient Descent
- Backpropagation algorithm
- Dimensionality reduction (MATLAB implementation done)

[2] Integration approach will be followed to adapt popular ml library Tensorflow for SCILAB.

ml libraries to be adapted from python to scilab-
1. Scikit-learn
2. tensorflow

Deep learning algorithms through Scilab (July):
1. Convolutional nn
2. Deep belief networks
3. Recursive nn
4. Deep Boltzmann machines

Goals
1. Implement all machine learning algorithms through SCILAB code
2. Make user accessible toolbox for using the created machine learning features
3. Create documentation and demos for usage of machine learning toolbox
4. Integrate Tensorflow or scikit-learn implementation of machine learning algorithm through SCILAB
5. Make added features available to user through toolbox and command line interface
6. Create documentation and demos for future integration of other machine learning libraries into SCILAB

Timeline and objectives

Before April 3:
- To completely familiarize myself with MATLAB to SCILAB change in coding syntax and formats.
- Implement linear regression and logistic regression in Scilab as a sample code
April 3 - May 3:
- Study all the machine learning algorithms to be implemented along with their mathematical explanations
- To familiarize myself with tensorflow and or scikit-learn’s(or any other ml toolbox) implementation techniques

May 4 – May 30 (Before the official coding period):
- During this period I will be in constant touch with my mentor and the SCILAB community. I will remain active on IRC and mailing lists to discuss and finalize the modifications (if any) that need to be done on existing neural network toolbox
- To do self-coding with SCILAB to for my understanding and ease of use with its toolboxes and internal coding
- With the help of my mentor I will become absolutely clear about my project goals - the final toolbox implementations that need to be done as well as the integration approach for other machine learning libraries

June 1 – June 15 (Official coding period starts):
- SCILAB implementation of all discussed machine learning algorithms and features
  - Linear Regression
  - Logistic regression
  - Neural networks
  - Support vector machines
  - K-means algorithms
  - Any other features discussed during community bonding period
- Test implemented features on various datasets (iris, MNIST etc)

June 16 – June 26:
- Write necessary documentation for the features created in the first 2 weeks
- Merge the code written in first 2 weeks with scilab master branch and do bug fixing wherever necessary
- Compare and document errors generated through SCILAB implementation of machine learning features versus its Python implementation

JUNE 26th - JUNE 30th  
FIRST EVALUATIONS

July 1 – July 15:
- Converting the code implementation into a machine learning toolbox for users
- Create necessary user interface and functionality for the same
- Refine existing code written before First evaluation based on Mentor’s recommendations
July 15 – July 24:
- To be in constant touch with the SCILAB mentors and sync up our progress with them
- Write necessary documentation for using the machine learning toolbox and its usage scheme
- Most of the time will be devoted to rigorous testing and bug fixes.

JULY 24th - JULY 28th

SECOND EVALUATIONS

July 28 – August 7:
- Discuss possible integration approach for using existing machine learning libraries (tensorflow or scikit-learn) for SCILAB implementation
- Implement ml features in SCILAB through the decided set of libraries
- Document necessary instructions for external library usage
- If time permits, convert it into an external ml toolbox

August 7 - August 21:
Setting a buffer of 2 weeks for any further extension in project goals or refining existing code written by August 7

AUGUST 21st- AUGUST 29th

CODE SUBMISSION AND EVALUATIONS
About me:
Hello, I am Mandar Deshpande student of Visvesvaraya National Institute of Technology Nagpur, currently enrolled in 4th year of Electrical and Electronics engineering. I have always been an active fan and developer of machine learning and deep learning based solutions to Electrical engineering problems (through MATLAB). And have been tinkering with newer algorithms and faster techniques to solve the already existing engineering problems.

I have made myself familiar with Scilab’s interface and style of coding in the last 3 months, and have been working on rewriting my MATLAB scripts into Scilab scripts for practice. So I am very comfortable working with Scilab and am confident that I will be able to work efficiently during GSoC’s timeline.

Email Address: mandar061095@gmail.com
Nickname: mandroid6
Real Address:
208, Shivalee Apartment Trimurti Nagar, Nagpur, Maharashtra, India 440022
Phone Number:
+918055215955
IT Languages:
MATLAB, C, Java – 3 years’ experience  C++, python – 2 years’ experience
Skills:
Machine learning (mathematics and programing experience), Deep learning (neural networks)
Familiarity with:
GIT - 7/10  SVN - 3/10
SCILAB usage:
Yes I have used SCILAB before as a free alternative to MATLAB for my University’s digital signal processing projects (image processing and segmentation).

Present Contributions to SCILAB: Already started adding machine learning features to scilab and corresponding documentation. Source code has been added to my github account:

https://github.com/mandroid6/scilab_ml
These machine learning features have been sent on the mailing list.

Coverity Bug fix submitted CID: 1371810

Contribute to SCILAB after GSOC:

Yes, I have been using SCILAB since 3rd year of my engineering life and I am very glad that it's made freely available for students to use and test their project idea instead of using the costly licenced MATLAB software package. Though there is a lot of improvement to be made in the SCILAB software package, it can only move forward if students/developers like myself take initiative to take it forward.

Undergraduate Research Project:

Meta-heuristic Algorithms and their implementation to Electrical and Electronics (July 2016 - Present)
- Researched Genetic Algorithm, Crow search Algorithm, Artificial Bee Colony, Neural Networks, Simulated Annealing, Firefly Algorithm, Swarm Optimization and related applications using MATLAB and python
- Meta-heuristic algorithms for estimation of fault location on transmission lines
- Simulink based model which runs meta-heuristic MATLAB script to optimize duty ratio of solar PV module

Internships:

Summer Internship (May – July 2016)
Citicorp Services India Limited, Pune
- Worked as a Software Analyst in Cash Management under the Treasury and Trade Solutions umbrella
- Actively worked alongside the Global Concentration Engine (GCE) team to develop Report Configuration Utility (using python, Java, Javascript)
- Also assisted in the development of the Global Concentration Engine’s functionality

Chief App Developer Rentarctica, VNIT Nagpur (May – November 2015)
- Worked as the Chief App Developer for the company
- Developed the UI (User interface) and functionality for its Android application using Android Studio and Eclipse (Java)
- Ranked 3rd at the Google Startup Weekend 2015. Currently mentored by VNIT Alumnus and founder of Nagpur Angels, Mr. Shashikant Chaudhari
Open-source Experience:

1. **Android Application for Rentarctica.com** (online renting portal based out of Nagpur)
2. **Implementation of handwritten digit recognition using TensorFlow library**
   https://github.com/mandroid6/tensorFlow_learn
3. **Chatbot using Telegram API and python backend**
   https://github.com/mandroid6/Mandroid6bot

Online Courses Completed:

1. Machine Learning by Stanford University (MATLAB) on Coursera
2. Programming for Everybody (Python) by University of Michigan on Coursera
3. Digital Image And Video Processing (MATLAB) by Northwestern University on Coursera

Quick links:


Commitment to the Project:

I have my summer vacations from 3rd Week of May till 30th July, which fits perfectly with the GSoC timeline, so I will be actively engaged in the project work for Google Summer of Code. After that the workload in mid-weeks of August will be light and I will have my full focus on this project.

I’m flexible with timings and am willing to adjust to my mentor’s schedule if required.

Time Zone - UTC+0530

Therefore, I Mandar Deshpande am making a formal commitment that I will be involved full time on the GSoC. And agree to work for 40 hours at minimum per week during the GSoC official coding period.
Sample code and documentation written for this project:

Machine learning Demo for **Linear Regression** using **Scilab** (complete code and data-set available on my github profile link above)

**Steps to be followed:**

1. **Create random data**
   - having two features, in a .csv file named “random_linear.csv”
   - This data will be used for training of model for linear regression.

2. **Import the data**
   - The data present in random.csv files has total of 97 samples divided into input ‘x’ and output ‘y’ on the basis of the parameters stored in 1\textsuperscript{st} and 2\textsuperscript{nd} column of the .csv file.
   ```scilab
t = csvRead(“random_linear.csv”);
```

3. **Split the data into t1 and t2 for plots**
   ```scilab
t1 = t(:,1);
t2 = t(:,2);
```

4. **Plot or represent the random data generated**
   ```scilab
clf(0); scf(0);
plot(t1,t2,’bx’);
```
5. Building a classification model

Our model should figure out how to fit the best straight line to our data

Separate data into features and results
\[ x = t_1; \quad y = t_2; \]
\[ [m, n] = \text{size}(x); \]

Add an extra column to feature vector \( x \) to accommodate the intercept term
\( x = [\text{ones}(m, 1) \times] \)

Hypothesis function for logistic regression is defined as
\[ h_\theta(x) = \theta^T x = \theta_0 + \theta_1 x_1 \]

It’s magnitude is the probability that the data with the features \( x \) lies on the line \( h_\theta(x) \)

The Cost Function in logistic regression is
\[
J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2
\]

Gradient Descent

Update rule for \( \theta \) using gradient descent algorithm is

\[
\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \quad \text{(simultaneously update } \theta_j \text{ for all } j).\]

Code:

```plaintext
//initialize fitting parameters
theta = zeros(n + 1, 1);

// Learning rate 'a' and number of iterations 'n_iter'
a = 0.01;
n_iter = 10000;

for iter = 1:n_iter do
  theta = theta - a * (x' * (x*theta-y)) / m;
  J(iter) = 1/(2*m) * sum((x*theta - y).^2)
end
```
6. Visualize the output

```matlab
// Display the result
disp(theta)
clf(1);scf(1);
plot(t1,t2,'rx');
plot(x(:,2), x*theta, '-')
```

7. Visualize the cost function for convergence of the model

```matlab
clf(2);scf(2);
plot(1:n_iter, J);
xtitle('Convergence','Iterations','Cost')
```
References:

http://www.holehouse.org/mlclass/04_Linear_Regression_with_multiple_variables.html