A MACHINE LEARNING APPROACH TO PROTEIN FOLD RECOGNITION

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Engineering

by

ASHWIN FERNANDES	**ROLL NUMBER 21**
KAVYA KOTIAN	**ROLL NUMBER 32**
CHIRAAG LIMAYE	**ROLL NUMBER 36**
KARAN MANGHI	**ROLL NUMBER 39**

Supervisor:

UDAY NAYAK

Department of Information Technology

Don Bosco Institute of Technology

2016-2017

AFFILIATED TO

UNIVERSITY OF MUMBAI

A PROJECT REPORT
ON
"A MACHINE LEARNING APPROACH TO PROTEIN FOLD RECOGNITION"

Submitted to
UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN INFORMATION TECHNOLOGY

BY

ASHWIN FERNANDES	**ROLL NUMBER 21**
KAVYA KOTIAN	**ROLL NUMBER 32**
CHIRAAG LIMAYE	**ROLL NUMBER 36**
KARAN MANGHI	**ROLL NUMBER 39**

UNDER THE GUIDANCE OF
PROF. UDAY NAYAK

DEPARTMENT OF INFORMATION TECHNOLOGY
DON BOSCO INSTITUTE OF TECHNOLOGY
VIDYAVIHAR STATION ROAD, MUMBAI - 400 070

2016-2017

AFFILIATED TO

UNIVERSITY OF MUMBAI

DON BOSCO INSTITUTE OF TECHNOLOGY

Vidyavihar Station Road, Mumbai - 400070

Department of Information Technology

CERTIFICATE

This is to certify that the project entitled **"A MACHINE LEARNING APPROACH TO PROTEIN FOLD RECOGNITION"** is a bonafide work of

ASHWIN FERNANDES	**ROLL NUMBER 21**
KAVYA KOTIAN	**ROLL NUMBER 32**
CHIRAAG LIMAYE	**ROLL NUMBER 36**
KARAN MANGHI	**ROLL NUMBER 39**

submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **Undergraduate** in **Bachelor of Information Technology**

Date: / /

(Prof. UDAY NAYAK)
Supervisor

(Prof. Janhavi Baikerikar)
HOD, IT Department

(Dr. Prasanna Nambiar)
Principal

DON BOSCO INSTITUTE OF TECHNOLOGY

Vidyavihar Station Road, Mumbai - 400070

Department of Information Technology

Project Report Approval for B.E.

This project report entitled **"A MACHINE LEARNING APPROACH TO PROTEIN FOLD RECOGNITION"** by **Ashwin Fernandes ,Kavya Kotian ,Chiraag Limaye ,Karan Manghi** is approved for the degree of **Bachelor of Engineering in Information Technology**

(Examiner's Name and Signature)

1. _____

2. _____

_____

(Supervisor : Uday Nayak)

Date:

Place:

DON BOSCO INSTITUTE OF TECHNOLOGY
Vidyavihar Station Road, Mumbai - 400070

Department of Information Technology

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea / data / fact / source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(_____)
(Ashwin Fernandes 21)

(_____)
(Kavya Kotian 32)

(_____)
(Chiraag Limaye 36)

(_____)
(Karan Manghi 39)

Date:

ABSTRACT

Protein identification and its subsequent classification into families is a important issue in biology and genetics. This can be used for disease identification and early prediction thus saving tremendous number of lives from deadly diseases like cancer. We train our system to identify the point of similarity of proteins and thus classify them into their respective classes, all using deep learning. We attempt to increase the accuracy and speed of our classification as compared to existing algorithms.

Keywords: Deep Learning , Protein-Family

Contents

Chapter 1

Introduction

1.1 Problem Statement

To use deep learning to classify the given protein into categories based on folds.

1.2 Scope of the Project

- Initially we give two similar proteins as input to our system. We ask the system to find out the similarity using machine/deep learning.
- Once the similarity is found, the system will remember what it has to look for when it is comparing two proteins in the future.
- We define a set of 8-10 classes and feed it in the system. All these classes are different from each other.
- Now when the user gives a new protein as input to the system, our system will find out which class does that query protein belong to based on the similarity of the protein with the class.

1.3 Current Scenario

The current scenario in context to protein fold recognition can be explained by an application called FOLDPro . It Directly compares query proteins with the template Based on family info, alignment, structural features . It uses SVM as it's algorithm .

1.4 Need for the Proposed System

- Disease Prediction, one of our future deliverables, requires classification of proteins. We need to find out similarities between proteins .
- If Protein A and Protein B are similar and Protein A causes a disease X, we can say that Protein B may also cause the same disease X.
- After folding, a protein performs a unique function. This means each fold has a unique function. Once we classify proteins, we come to know what functions they would perform later on.
- For instance, if Protein A and Protein B are similar and protein A is responsible for hair color, we can say that protein B and other proteins in the same class as protein A have something to do with hair color.

1.5 Task completed

- Literature survey
- Feasibility study

Chapter 2

Review of Literature

2.1 Summary of the investigation in the published papers

- Deep learning approach to ab initio protein secondary structure prediction:
  - Traditional approaches used for prediction could not achieve accuracy higher than 65
  - Therefore a machine learning approach i.e support vector machines and neural networks were used.
  - Combining machine learning with proteins increased the accuracy from 65
  - The accuracy was increased by combining the Q3 score and the SOV score.
  - Datasets used were: PDB,CASP9 and CASP10.
  - Tools used were:
    - DSSP- Tool that utilizes the dictionary of protein secondary structure.
    - PSI-BLAST- Used to calculate a position- specific scoring matrix for each of the training and testing proteins.
- Protein family classification with Neural networks:
  - Protein families are defined to group together proteins that share similar structure.
  - It focusses on training vector representations for protein sequences and investigate various neural network models for predicting a protein's family.

* Dataset used: Universal Protein Resource (UniProt).

* Used Global Vector for amino acid sequence representation.

* Support Vector Machine was used to classify the proteins. It took 7 hours to train the model.

* Neural Network models:

  · Gated Recurrent Neural Networks (GRU)

  · Long Short-Term Memory (LSTM)

  · Bidirectional LSTM (biLSTM)

  · Convolutional Neural Network (CNS)

* Protein families were accurately predicted from amino acid sequences. All the neural network models performed better than SVM.

– Improving Protein Fold Recognition By Deep Learning Networks :

* More number of hidden layers

* The additional hidden layers allow the network to capture more correlations and patterns present in the data and this often leads to improved performance

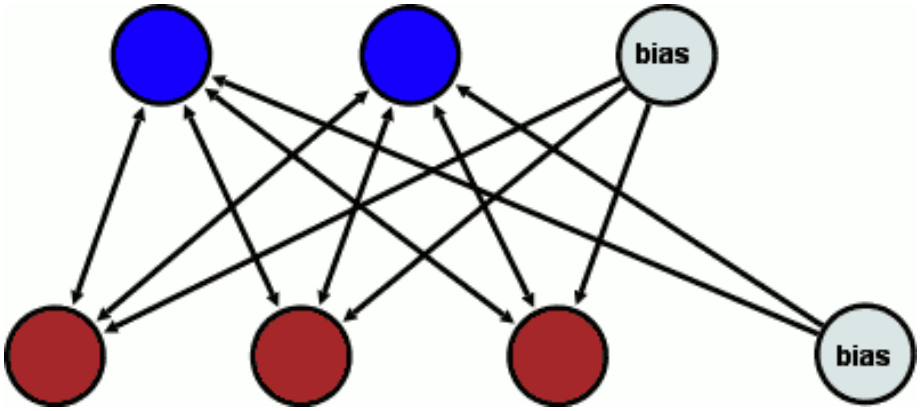
* This leads to an unsupervised learning process that first attempts to learn patterns in the data and then maps these learned patterns to the proper labels

2.2 Comparison between the algorithms

Features	SVM(Support Vector Machines)	Logistic Regression	Random Forests	Restricted Boltzmann Machine(RBM)
Average Predictive Accuracy	Lower	Lower	Higher	Higher
Training speed	Moderate	Fast	Slow	Slow
Prediction speed	Moderate	Fast	Moderate	Fast
Performs well with small no of observations?	Yes	Yes	No	No
Automatically Learns Feature Interactions?	No	No	Yes	Yes

2.3 Algorithm(s) with example

WHAT IS A RESTRICTED BOLTZMANN MACHINE:
Restricted Boltzmann Machine is a stochastic neural network (that is a network of neurons where each neuron have some random behavior when activated). It consist of one layer of visible units (neurons) and one layer of hidden units. Units in each layer have no connections between them and are connected to all other units in other layer (fig.1). Connections between neurons are bidirectional and symmetric . This means that information flows in both directions during the training and during the usage of the network and that weights are the same in both directions.



RBM Network works in the following way:
First the network is trained by using some data set and setting the neurons on visible layer to match data points in this data set. After the network is trained we can use it on new unknown data to make classification of the data (this is known as unsupervised learning)

Training algorithm :
Restricted Boltzmann machines are trained to maximize the product of probabilities assigned to some training set V (a matrix, each row of which is treated as a visible vector v),

$$\arg\max_W \prod_{v \in V} P(v)$$

or equivalently, to maximize the expected log probability of a training sample v selected randomly from V:

$$\arg\max_W E\left[\log P(v)\right] \arg\max_W E\left[\log P(v)\right]$$

The algorithm most often used to train RBMs, that is, to optimize the weight vector W, is the contrastive divergence (CD) algorithm due to Hinton, originally developed to train PoE (product of experts) models. The algorithm performs Gibbs sampling and is used inside a gradient descent procedure (similar to the way backpropagation is used inside such a procedure when training feedforward neural nets) to compute weight update.

The basic, single-step contrastive divergence (CD-1) procedure for a single sample can be summarized as follows:

– Take a training sample v, compute the probabilities of the hidden units

and sample a hidden activation vector h from this probability distribution.

- Compute the outer product of v and h and call this the positive gradient.
- From h, sample a reconstruction v' of the visible units, then resample the hidden activations h' from this. (Gibbs sampling step)
- Compute the outer product of v' and h' and call this the negative gradient.
- Let the update to the weight matrix W be the positive gradient minus the negative gradient, times some learning rate:

$$\Delta W = \varepsilon(vh^{\mathsf{T}} - v'h'^{\mathsf{T}})\Delta W = \varepsilon(vh^{\mathsf{T}} - v'h'^{\mathsf{T}}).$$

- Update the biases a and b analogously:

$$\Delta a = \varepsilon(v - v')\Delta a = \varepsilon(v - v'), \Delta b = \varepsilon(h - h')\Delta b = \varepsilon(h - h').$$

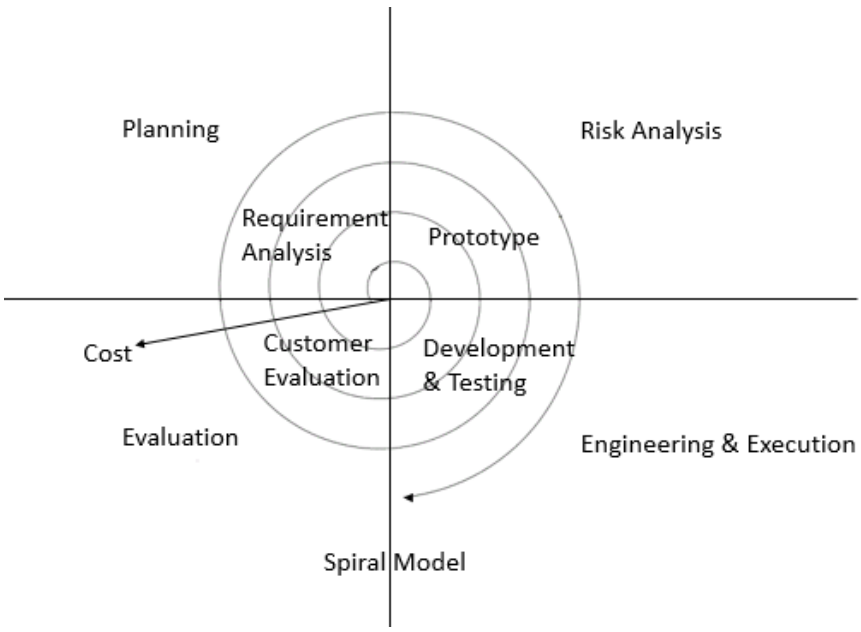
Chapter 3

Analysis and Design

3.1 Methodology / Procedure adopted

Spiral Model :

We will be implementing the spiral model for our proposed project . It is a combination of a waterfall model and iterative model. Each phase in spiral model begins with a design goal and ends with the client reviewing the progress. The development team in Spiral-SDLC model starts with a small set of requirement and goes through each development phase for those set of requirements. The development team adds functionality for the additional requirement in every-increasing spirals until the application is ready for the production phase.



Spiral Model spans into 4 phases :
Planning :
It includes estimating the cost, schedule and resources for the iteration. It also involves understanding the system requirements .Feasibility Study

Risk Analysis :
Requirements are studied and brain storming sessions are done to identify the potential risks.Once the risks are identified , risk mitigation strategy is planned and finalized

Engineering :
Actual development and testing if the software takes place in this phase

Evaluation :
Customers evaluate the software and provide their feedback and approval

For measuring and monitoring the progress of the project we can use the following milestones provided by the spiral model :

- Life Cycle Objectives (LCO)
- Life Cycle Architecture (LCA)
- Initial Operational Capability (IOC)

These milestones serve as intermediate checkpoints to keep the project moving full steam ahead in the right direction. The LCO milestone checks to see if the technical approach to a project is well-defined enough to proceed, and that stakeholder conditions are met. If “Yes”, continue. If no, abandon ship or commit to another lifecycle and try again. The LCA milestone checks that an optimal approach has been defined and that all major risks are accounted for and planned for. If “Yes”, continue. If no, abandon ship or commit to another lifecycle and try again. The ICO milestone checks that adequate preparations have been made to satisfy stakeholders prior to launch. This includes the software, site, users, operators, and maintainers. If “Yes”, its time for launch. If no, abandon ship or commit to another lifecycle and try again.

3.2 Analysis

- Requirements Gathered:
  - The requirement gathering process started with the collection of information about proteins.
  - We studied the atoms and molecules that formed a protein, different structures of proteins, their significance, etc.
  - While studying about the structures of proteins, we came across protein folding. Protein folding refers to the change in structure of a protein in a specific way which is called as a fold. This structure is formed so that it can perform a specific function.
  - We found out that almost all the discovered proteins consisted of two components namely Alpha Helices and Beta Sheets.
  - After protein fold, we came across protein family. Every protein is similar to a few other proteins which belong to the same family. This similarity is based on the structure of protein, its function, etc.
  - To implement a project that is a combination of Computer Science and Biology, we needed a method or an algorithm. The machine learning algorithms that we came across are support vector machines, regression, deep learning, etc.
- Feasibility study:
  - To develop a system that will predict the protein fold or the tertiary structure of a protein required a lot of data about all the discovered proteins, the folds that they undergo, the cause of protein folding, the environment required for protein folding, etc.
  - This study helped us understand that protein fold recognition cannot be achieved on an undergraduate level.
  - To compare proteins and find the similarity between them and then classifying them into classes was possible using deep learning.

3.3 Proposed System

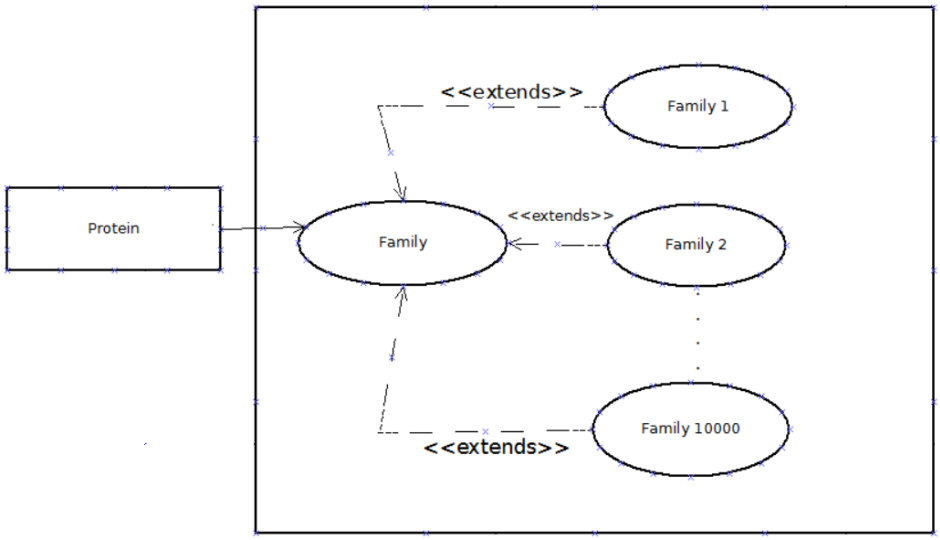
3.3.1 Hardware / Software requirements

- TensorFlow

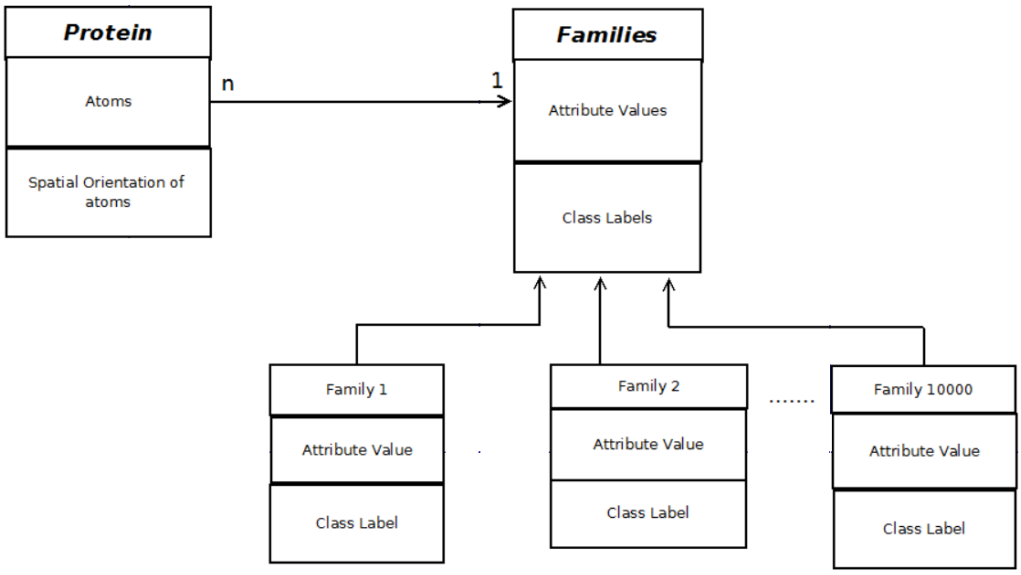
- Anaconda framework for Python
- iPython notebook
- Jupyter Editor for Python

3.3.2 Design Details

Different UML diagrams are shown below :
USE CASE DIAGRAM :



CLASS DIAGRAM :


Protein
Atoms
Spatial Orientation of atoms
n
1
Families
Attribute Values
Class Labels
Family 1
Attribute Value
Class Label
Family 2
Attribute Value
Class Label
.....
Family 10000
Attribute Value
Class Label

Chapter 4

Results and Discussion

Summarizing the working of the project in the following points:

- The system will be provided with 100 pairs of proteins and it will be told that those proteins are similar
- Then the system will run a sequence identifier machine learning program to check the reasons(points) at which the protein similarity is determined and will remember these points
- Now the system knows where and what to look for in a protein sequence to classify it into a particular class
- We will define 10000 classes(families) into which the input proteins can be classified
- When a protein in the form of a text file is fed into the system, the system will use the 2nd point(of the summary provided above) to classify the protein into one of the families

Tasks completed:

- Literature Survey
- Clarity of the concept of the project
- Determining the input
- Determining the algorithm to be used
- Defining the output

Contribution of team members: As this was a team effort, we all gained knowledge on everything needed for the project and worked on all the things together.

Chapter 5

Conclusion

In conclusion, our project will be using a spiral model for it's development. The main algorithm that our system will be using is the Restricted Boltzmann Machine(RBM) along with a sequence identification program to detect the similarity between pairs of protein initially. The only actor in our project will be the protein input in the form of a text file and the use cases will be the 10000 protein families. This system will benefit the society in the way that it could be used for protein classification and then associating the class with a particular disease and hence detecting the disease with the help of probable proteins causing that disease. It could help save people suffering from deadly diseases like cancer by detecting these diseases earlier than usual.

References

[1] Timothy K. Lee , Tuan Nguyen ; Protein Family Classification with Neural Networks

[2] Restricted Boltzman Machine `https://en.wikipedia.org/wiki/Restricted_Boltzmann_machine` , last modified on 27 October 2016.

[3] Restricted Boltzman Machine `http://imonad.com/rbm/restricted-boltzmann-machine/` .

[4] Matt Spencer, Jesse Eickholt, and Jianlin Cheng; *A Deep Learning Network Approach to ab initio*; IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, VOL. 12, NO. 1, JANUARY/FEBRUARY 2015

[5] Ken A. Dill, S. Banu Ozkan, M. Scott Shell, and Thomas R. Weikl; *The Protein Folding Problem*

[6] Taeho Jo, Jie Hou, Jesse Eickholt Jianlin Cheng; *Improving Protein Fold Recognition by Deep Learning Networks*; Scientific Reports 5, Article number: 17573 (2015)

Acknowledgements

We as a group have taken sincere efforts in this project. And since we used a Merge and Conquer strategy, all the team members worked on all the aspects of the project equally and dedicatedly. However it wouldn't have been possible without the kind support of our project guide, Mr Uday Nayak. We would like to extend our sincere thanks to him.

We are also highly indebted to our college, Don Bosco Institute of Technology, for the guidance that we received from the various faculty members and the resources in the form of computers and space that it provided to us.

(_____)
(Ashwin Fernandes 21)

(_____)
(Kavya Kotian 32)

(_____)
(Chiraag Limaye 36)

(_____)
(Karan Manghi 39)

Date: