Indian Institute of Technology, Kanpur

Using Collaborative System of Neural Networks for creating Intelligent Tutoring System

by

Adarsh Srivastava

Dr.Amey Karkare
Department of Computer Science and Engineering

July 2017

Declaration of Authorship

I, AUTHOR NAME, declare that this thesis titled, 'THESIS TITLE' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

Indian Institute of Technology, Kanpur

Abstract

Dr.Amey Karkare
Department of Computer Science and Engineering

Doctor of Philosophy

by Adarsh Srivastava

ITS, is the new teaching technology of the future where not only programming questions could be checked by online judges but also various maths problem could be verified so that the teachers can give more time to quality teaching rather than wasting time on checking 50-60 copies, this system could be really helpful in Developing Countries where the Teacher/Student Ratio is very low. Students can learn the connection between Co-ordinates and simple Geometric Entities, this will help them build a strong foundation. The ITS system uses Neural Net Classification Mechanism to break down complex English Sentences into proper geometric constructs. Which can be easily mapped and drawn.

By adding more and more components we can scale this system to not only include Geometry but also many other maths problems such as Trigonometry and Calculus.

So now a student need not just get information from the internet, but also have an interactive learning experience.

This tool is equally beneficial to the students in Developed Countries because of it's cloud based nature, the student can pick at home from the point he left in class. The easy to use interface of both Web and Mobile App makes it really easy for students to get hold of it.

Acknowledgements

I would like to Thank, Dr.Amey Karkare for his vision and support, without which this would have never been possible. There were moments where I thought that solving a particular problem was impossible, but then sir came up with a very in-genius solution and the problem was solved in just matter of seconds.
He also helped me build a deployment based system from a small project type of a system.

Contents

Abbreviations

ITS **I**ntelligent **T**utoring **S**ystem

Chapter 1

Introduction

Teaching computers is the thing of today, with the advent of machine learning, a whole new dimension has been exposed. Earlier the computer needed set of specific instructions or Algorithms to work, in many complex cases it was almost impossible to design an efficient one. But after machine learning, almost all problems related to real world can now be solved in a logically broken down steps.

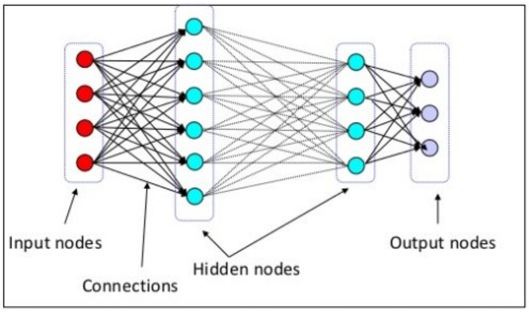
1.1 Intelligent Tutoring System - ITS

Application of computer technology to the field of education is very important, as with this fast growing world the needs of students are constantly evolving, and these system provide them with a tool to solve some of their academics related problems. In return we get a chance of doing user study and finding out how students approach a given problem, which in turn will help us create better systems for their needs, a system that adapts to a user.This feedback mechanism is what is needed, to determine how different people think, based on which we can target each student to a specific stream.

1.2 Deep Learning

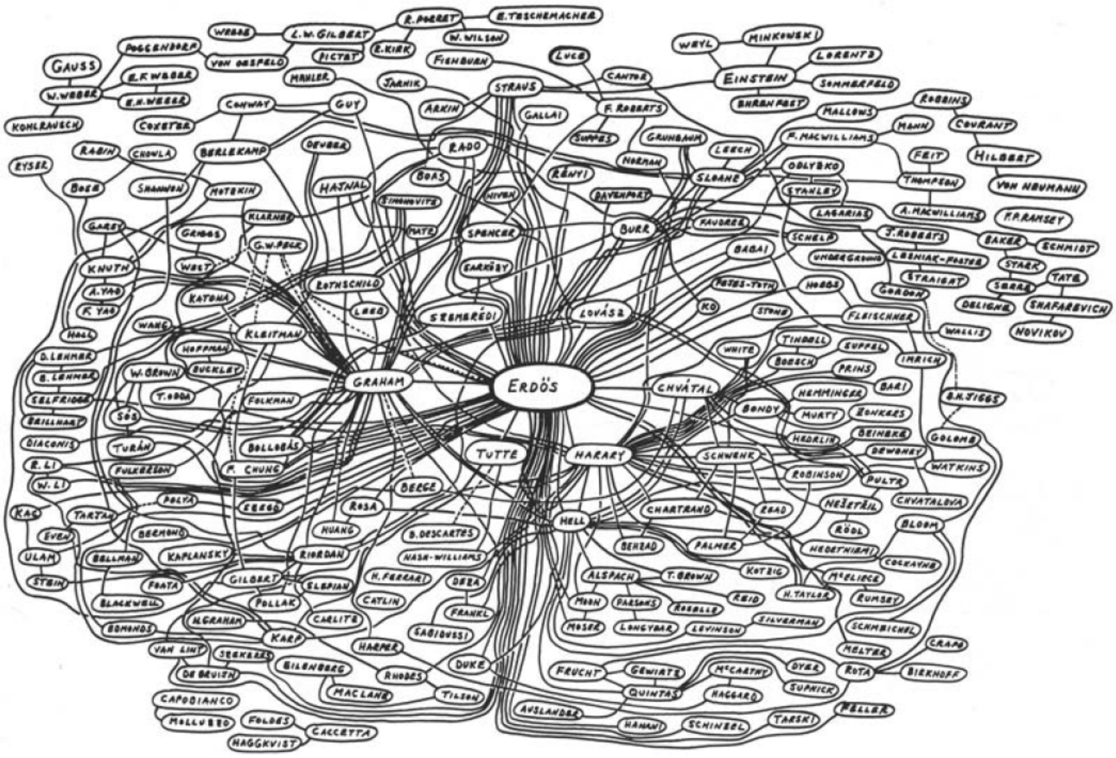
The concept of **Intelligent Tutoring System** can be approached using various ways, some of us that we experimented were **N-Gram**, **Giza++**, **Regular Expressions** but these approach somewhere failed to capture the essence of Geometry, using a **Deep**

Learning Classifier helps us create a system that understands what geometrical sentences mean.



1.3 Erdos Numbers

You might be wondering what a mathematical concept is doing here, but yes that's the key! There are 4 servers namely - line, circle, quadrilateral, triangle

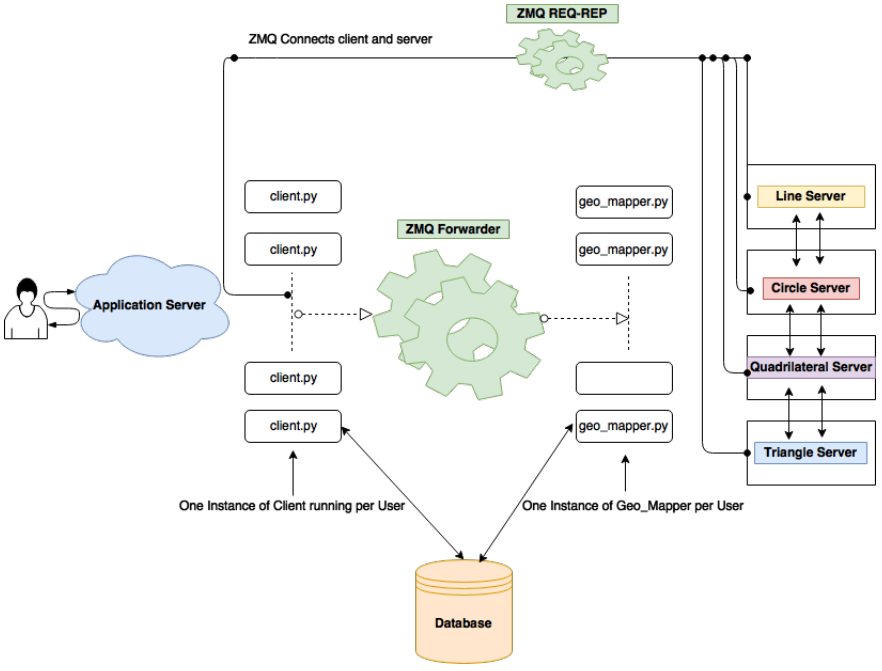


Chapter 2

Outline of the System

2.1 Top Level Abstraction

If seen on a top level the application takes input from the user and draw it on the screen of the user's browser. The process is best described in the following diagram - add diagram here The web server directs the request by creating a new instance of client, once the client is started it starts an instance of Geo Mapper. Geo Mapper calls the required function and then outputs to a JS file that stores the list of drawing commands. Which is then picked up by the web server and the commands in the JS are executed to produce the diagram.



2.1.1 Server

There are only 4 instances of server running for the whole setup, all user share this resources. The communication is asynchronous so there is no conflict, and the

$(n)^{th}$ user does not have to wait for the command of $(n-1)^{th}$ user to be processed.

Both can run independently. The server takes the raw statement that that the user inputs and then classify it based on Deep Learning Processes. It outputs to the client.

2.1.2 Client

A client gets the input string and the current active user from the web server and then sends the input to all the servers running predicting modules that make the use of NLP and Deep Learning. And it gets the output from these serves and then calculates the maximum probability, using which it decides whether the command is to draw a line or circle or other shapes. Once decided it send the info to the Geo Mapper which extracts other information from the string and then draws the shape.

2.1.3 Geo Mapper

Geo mapper accepts input from the Client and outputs to a JS file. It's main function is to process the data sent by the client and then process it using many functions. Some of which includes line mapper, circle mapper etc.

Chapter 3

Details of Components

3.1 Front End

The web clients sits at top of every components and receiving the input from the user and showing the corresponding output. It has all the features necessary for a cloud based system. It has a User Authentication Portal, users can sign up and then use the app from any device. Listing out the currents and well as the components that will be used to achieve this goal.
The main aim is to reuse the components once written.

3.1.1 Web Application Frontend

This would be made in Angular 2 using REACT.JS this will ensure a smooth and fast interface, this will also reduce the lag and loading time as most of the request response mechanism will run in back ground.

3.1.2 Desktop Application

The Desktop application would be written using Electron this will ensure reusibility of the code written in Angular 2. And one code could be ported to all the three operating system i.e Windows, Mac and Linux.

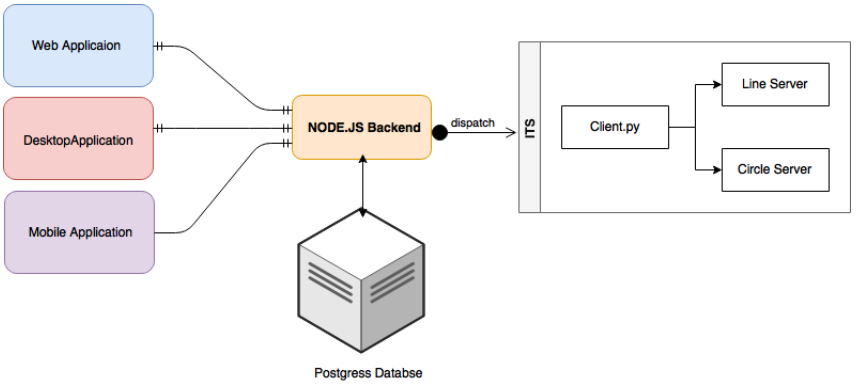
3.1.3 Mobile Application

The mobile app would be made using Ionic.JS this will ensure that the frontend code is reused and we can target all the possible operating system such as Android, iOS, Symbian and Java.

3.2 Backend

The Backend has been written in Node.JS, this backend acts as a communication between the Fronend and the ITS Application. The diagram below will explain the flow of information.

FIGURE 3.1: Backend takes the input from the USER and then passes it on the specific client.



3.3 Internals of ITS

ITS consists of various parts that are grouped together. These components together form a complex network of information flow, using the power of socket communications. **The ITS has been designed to be completely Modular, any part of the application can be used Independently**

3.3.1 Informers

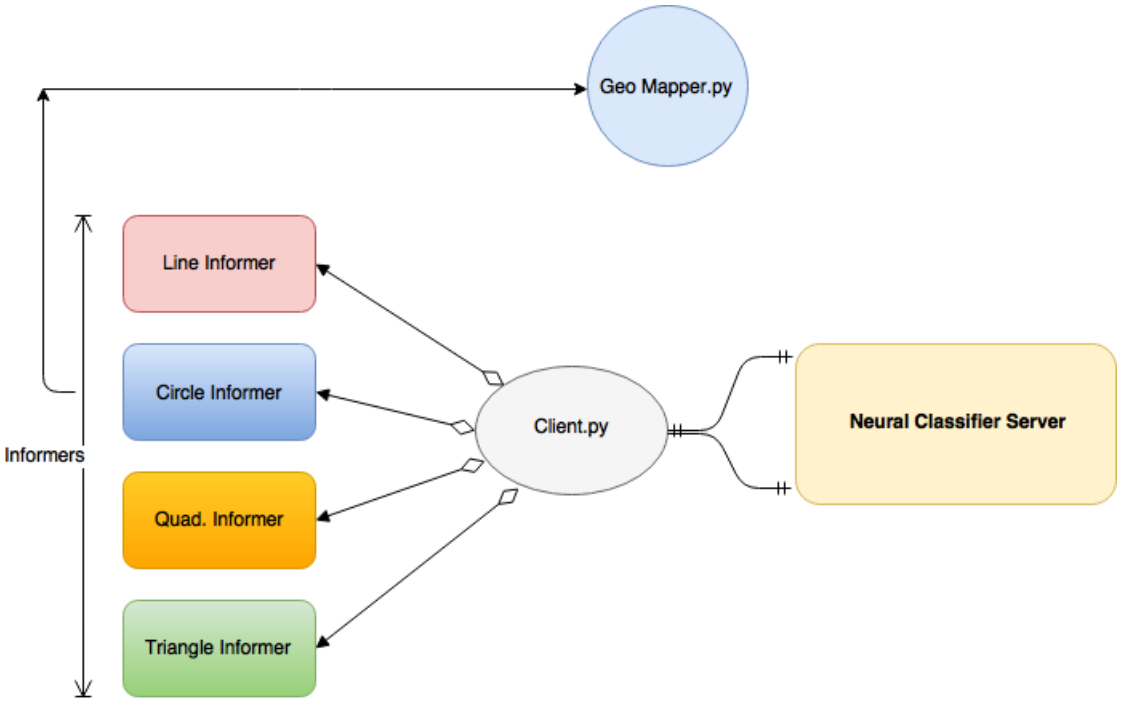
Informers extract the given information from the raw input.The information extracted varies from the context of sentence which is supplied from the neural net system. If the neural net assigns a class of "Line", then the Line Informer is invoked and it then extracts the information from the sentence and passes it to the Geo Mapper.

```
def line_info(input_sentence,max_output_line):
    length = re.findall("\d+\.\d+", input_sentence)
    points = re.findall("[A-Z][A-Z]", input_sentence)
    raw_string = max_output_line["sentence"]
    processed_dict = {
        "command":command,
        "end_points":points,
```

```
        "length":length,
    }
    return str(processed_dict)
```

LISTING 3.1: Sample Code for Line Inforrmer

FIGURE 3.2: Informers send data directly to Geo Mapper using PUB/SUB TCP Model



3.3.2 Geo Mapper

Geo Mapper takes the input from the Informers and then Calls the required Mappers to draw the corresponding figure. It also keeps the track of clusters of line, circles and points. Clearing them when the user logouts, then the details from the MongoDB goes into the Postgress DB, which can be loaded as and when required.

3.3.3 Mappers

These are dedicated sub routines for mapping information to geometric figures, there are mappers for each and every geometric construct. These are the current mappers in the program, new mappers will be added as need arises.

- Line Mapper

- Circle Mapper
- Bisector Mapper
- Perpendicular Mapper
- Join Mapper
- Mark Mapper

These mappers then output to the corresponding drawing routines written in Java Script that eventually draws on the screen.

```
get_point_1 = coll_coordinates.find_one({"point_name" : p1,"user":user})
p1_x = get_point_1.get('point_x')
p1_y = get_point_1.get('point_y')
get_point_2 = coll_coordinates.find_one({"point_name" : p2,"user":user})
p2_x = get_point_2.get('point_x')
p2_y = get_point_2.get('point_y')
pb_x = float(p1_x+p2_x)/2
pb_y = float(p1_y+p2_y)/2
bisector_point = p1.lower()
coordinates_map = {
        "user" : user,
        "point_name" : "",
        "time_added" : datetime.now(),
    }
coordinates_map["point_name"] = bisector_point
coordinates_map["point_x"] = pb_x
coordinates_map["point_y"] = pb_y
insert_point = coll_coordinates.insert_one(coordinates_map)
```

Listing 3.2: Sample Code for Perpendicular Mapper

3.3.4 Drawing Module

The drawing module is written in Java Script and contains Basic HTML5 Canvas commands to draw simple shapes like Line, Circle, Quadrilateral and Triangle.

```
var canvas = document.getElementById('canvas');
var ctx = canvas.getContext('2d');

ctx.fillStyle = 'green';
ctx.fillRect(10, 10, 100, 100);
```

Listing 3.3: A sample code to draw a rectangle in HTML5 Canvas

3.3.5 Training Data

The training data is a set of classifier based data set, that can be directly modified by the USER if he/she needs to add extra support, or need to enhance the corpus. The Data Set will be open to contributions.

```
Line, line: $point_pair, length: $length cm
#Draw a line $point_pair of length $length cm
Line, line: $point_pair, length: $length cm
#Construct a line $point_pair of length $length cm
Line, line: $point_pair, length: $length cm
#Draw a line segment $point_pair of length $length cm
```

Listing 3.4: Few Entries in Line Data Set

3.3.6 Neural Classifier

The main heart of our ITS Geometry Application. It accepts raw input from the user, by client.py file. And then returns the information about the sentence classifying it under "Line", "Circle", "Triangle", "Join", "Rectangle", "Perpendicular" etc.

```
def classify(sentence, show_details=False):
    results = think(sentence, show_details)

    results = [[i,r] for i,r in enumerate(results) if r>ERROR_THRESHOLD ]
    results.sort(key=lambda x: x[1], reverse=True)
    return_results =[[classes[r[0]],r[1]] for r in results]
    length = len(return_results)
    max = 0.0
    max_index = 0
    if length > 1:
        max = return_results[0][1]
        for i in range(0,length):
            if (return_results[i][1] > max):
                max = return_results[i][1]
                max_index = i
    results = []
    results.append(return_results[max_index][0])
    results.append(return_results[max_index][1])
    return results
```

Listing 3.5: Sample Code of our Classifier

3.3.7 Connecting Devices

All the flow of infirmations occurs via TCP Sockets, they pass information from cliet.py to geo mapper.py and from the classifier servers to client.py We are using ZMQ for all socket based communications

FIGURE 3.3: Zero MQ

FIGURE 3.4: Internal Architecture of ZMQ

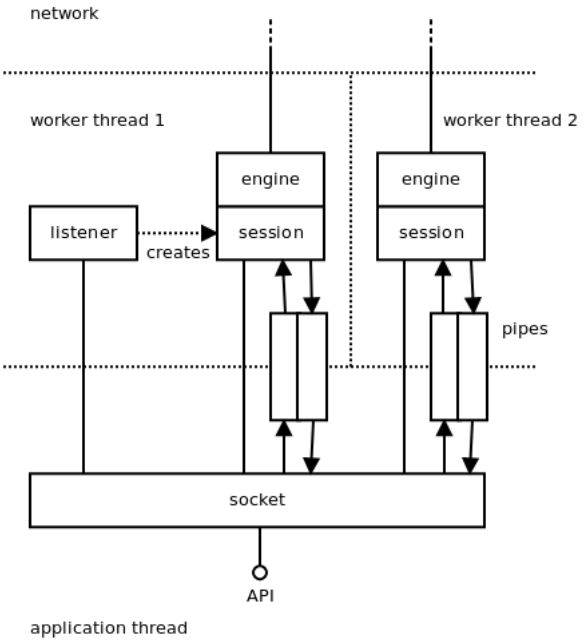
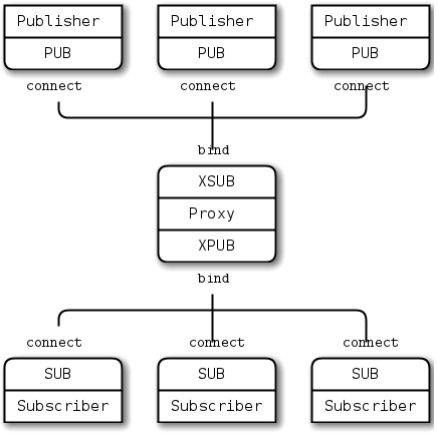


FIGURE 3.5: Publisher/Subscriber Model of ZMQ



Chapter 4

Application of ITS

4.1 Application to Teaching

This ITS can be used by Teachers as well as Students. The simple interface makes it easy for students of junior classes. It can also be used like a tool to teach students how to give instructions to computer in raw English. The Student can draw a specific figure and get deeper insights into it by the additional information provided by our application. **The student can learn about the connection of practical geometry and Co-ordinate Geometry**, this will help build a stronger foundation for students.
ITS will soon have a tool to check the correctness of a solution, using which this can also be used as a judging platform, **where a teacher can give some questions to the students and the solution given by the students would be self verified by the system**. Sparing their time from checking 50 geometrical drawings and making sure each one of them is correct by making measurements each time.

4.2 Application to User Study

ITS can be used to do study on how different students approach different questions. The Postgress database that was connected with the bakcend stores the raw English commands inputted by the user, and the corresponding mapping command generated. Now by using regression analysis on these two fields with the raw English commands on

X axis and the mapping commands on the Y axis we can find an approximate relation between what level of difficulty of a geometrical construct a student is comfortable with.