Analysis of AlphaGo Artificial Intelligence Program and Implementation.

<u>**AIM**</u> **:** *To analyze the working of Google's Deep Learning powerful AI i.e. AlphaGo and discovering its implementations. Also, study various latest implementations and derive a hybrid model named as* ***'MinAlphaGo'*** *with a better and faster user interface along with the analysis of game state at any instant.*

Mr. Devender Kumar
(Project Supervisor)
Department of Information Technology
Netaji Subhas Institute Of Technology, Delhi.

By:-

Aditi Kumar (706/IT/14)
Arushi Bhatt (718/IT/14)
Diksha Sharma (731/IT/14)

OBJECTIVE

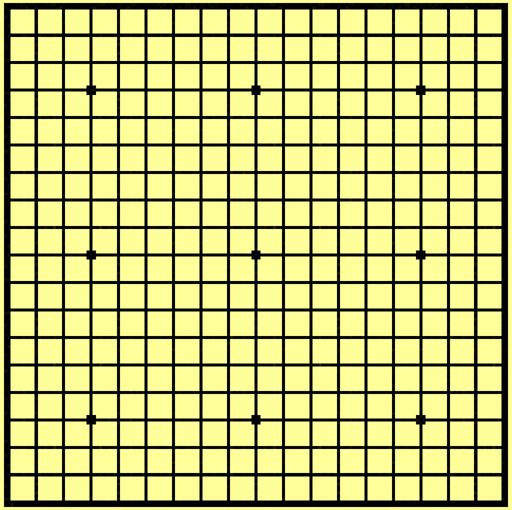
- Studying AlphaGo (Google's Deep Learning powerful AI).
- Study of Google's research paper titled as : 1.Mastering the game of Go with deep neural networks and tree search : https://storage.googleapis.com/deepmind-media/alphago/AlphaGoNaturePaper.pdf
- Combining plus points of BetaGo and Leela Game Engine implementations and coming out with hybrid version named as 'MinAlphaGo'.
- Comparing the interface of Leela and MinAlphaGo.

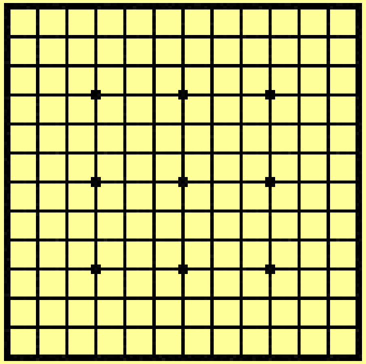
WHAT IS GO GAME?

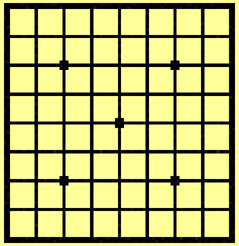
-Go is an ancient(2500 yrs) Chinese two player abstract strategy board game(oldest board game).
-Go has simpler rules than chess.

More about go GAME :

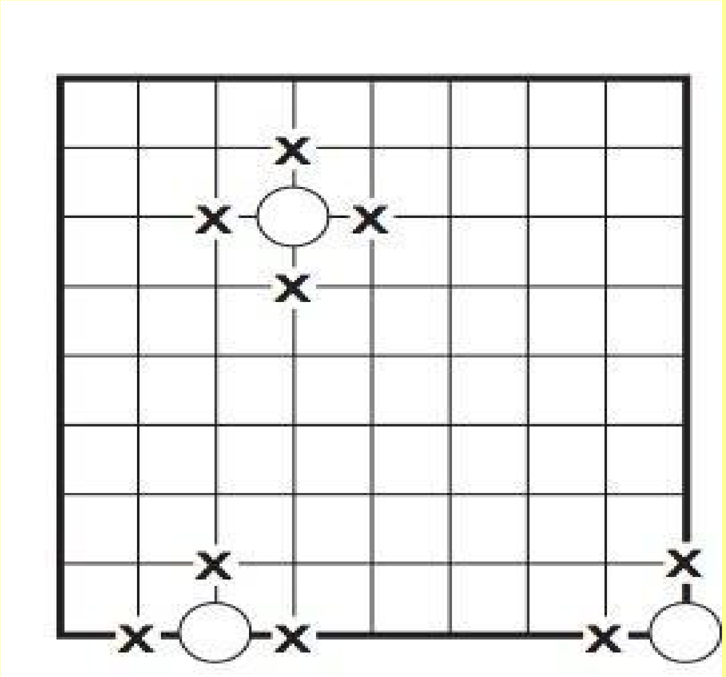
- Although the official size of a Go board is 19 x 19 ,but it can be played on a 13 x 13 board and 9 x 9 board.
- **Objective** is to capture more territories than the   opponent and other players stone.

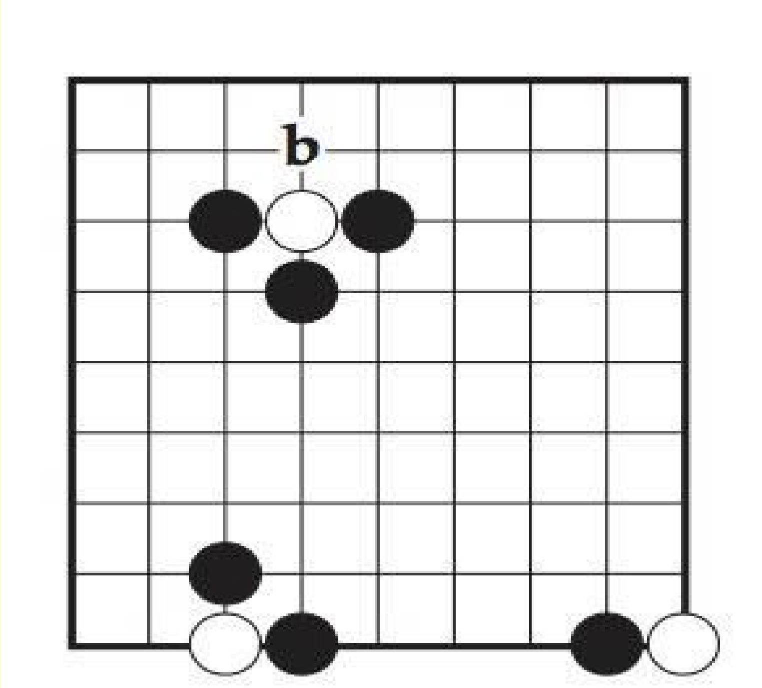




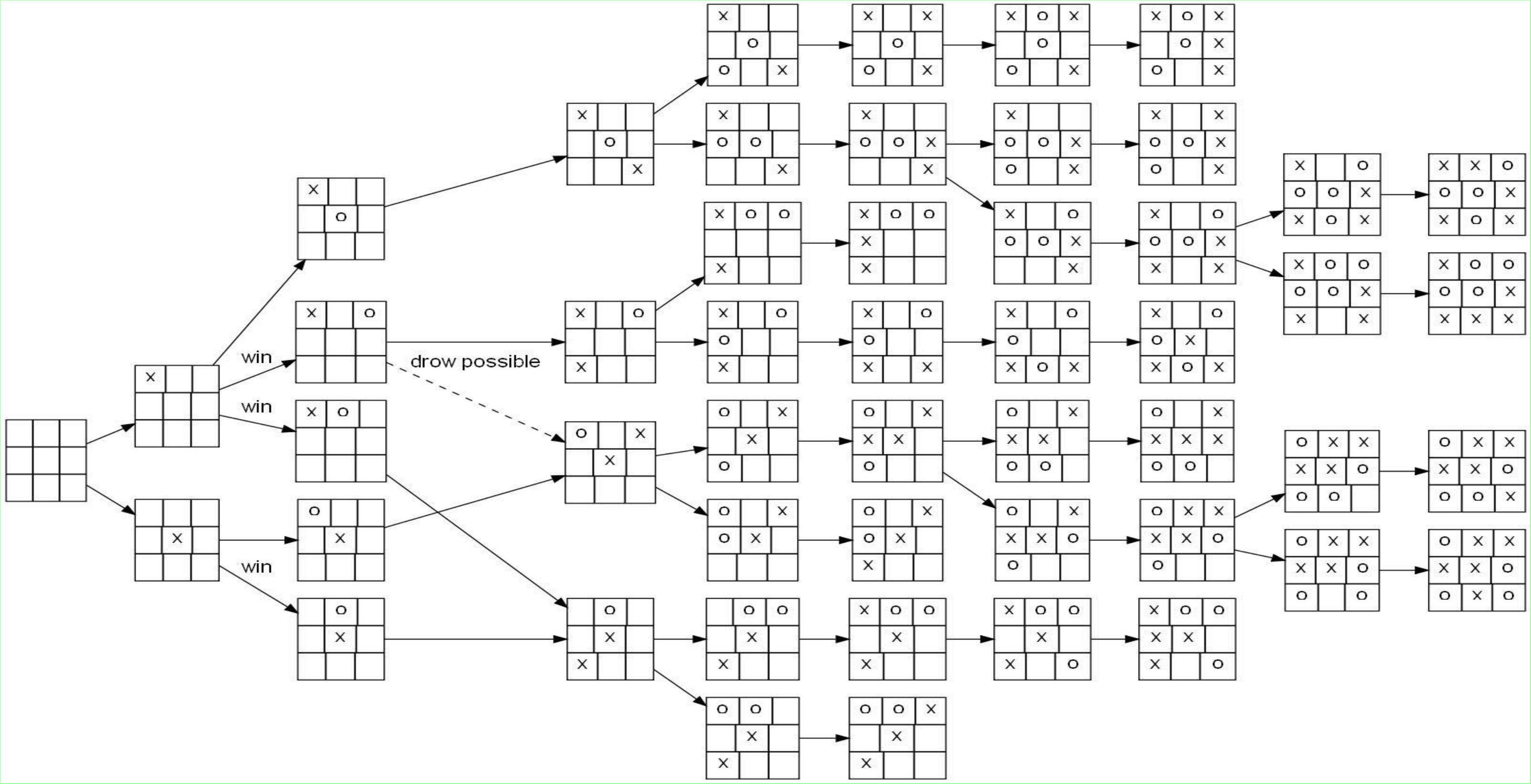


- A basic principle of Go is that a group of stones must have at least one liberty(open point bordering the group) to remain on the board.
- An enclosed liberty (or liberties) is called an eye, and a group of stones with two or more eyes is said to be unconditionally alive.
- open "point" (intersection) bordering the group
- Such groups cannot be captured, even if surrounded.
- A group with one eye or no eyes is "dead" and cannot resist eventual capture.





- To understand how AIs are capable of playing games such as chess and Go, we have to understand what a game tree is.
- A **GAME TREE** represents game states (positions) as nodes in the tree, and possible actions as edges.
- The root of the tree represents the state at the beginning of the game. The next level represents the possible states after the first move, etc... For simple games such as tic-tac-toe, it is possible to represent all possible game states (the complete game tree) visually:



GO VS CHESS:

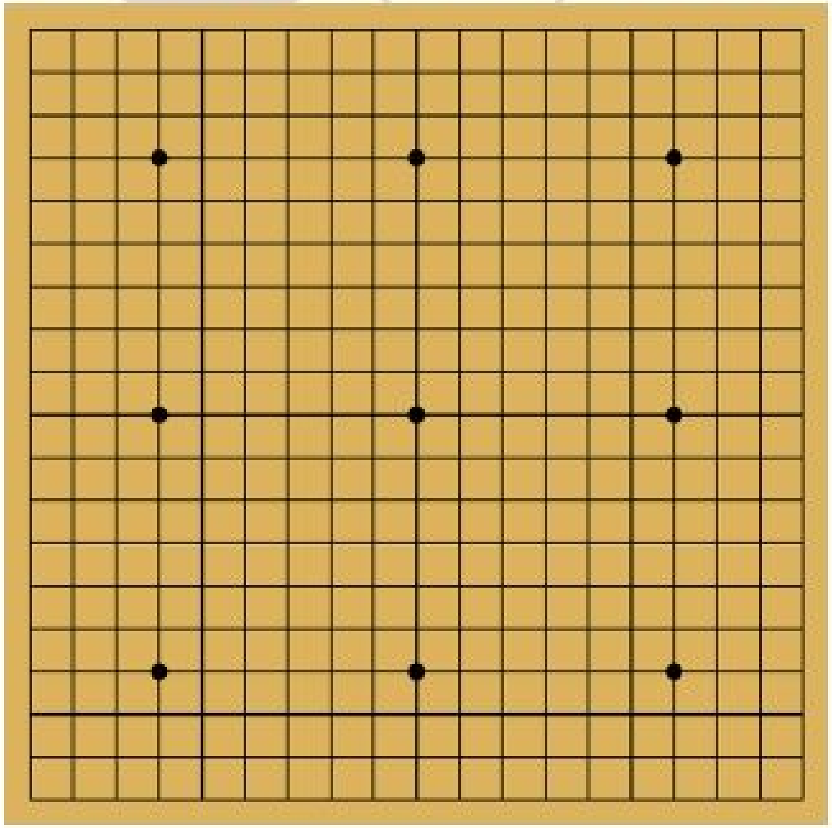
- Similar in some respects: both are **played by two players taking turns**, and there is **no random element involved** (no dice rolling, like in backgammon).
- In 1997, Garry Kasparov was defeated by **Deep Blue**, a computer program written by IBM, running on a supercomputer, Superficially, AlphaGo's win against Lee Sedol can be compared to Deep Blue's win against Gary Kasparov.

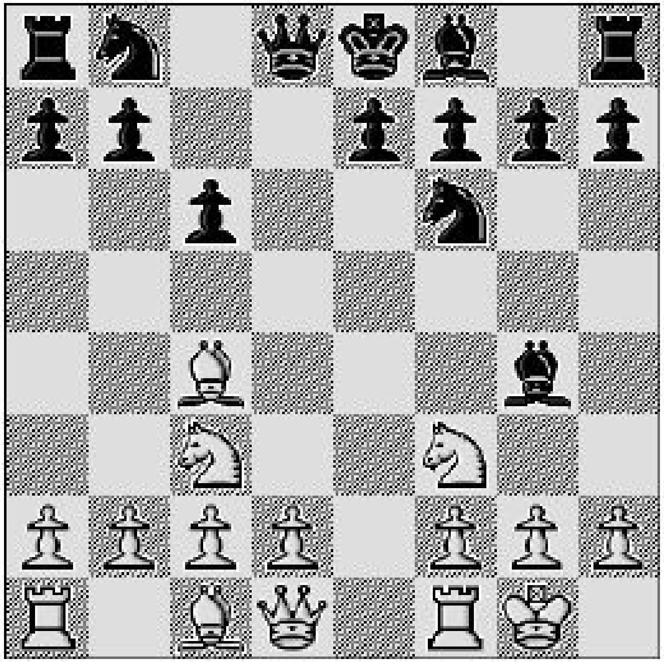
- In chess,
  - each player begins with 16 pieces of six different types.
    - Each piece type moves differently.
    - The goal of the game is to capture the opponent's king.
- Go starts with an empty board.
  - At each turn, a player places a stone (the equivalent of a piece in chess) on the board.
  - Stones all obey the same rules.
  - The goal of the game is to capture as much territory as possible. It can therefore be argued that **Go has *simpler* rules than chess.**
- In spite of the fact that the rules of Go might appear simpler than the rules of chess, **the *complexity* of Go is higher. At each game state, a player is faced with a choice of a greater number of possible moves compared to chess (about 250 in Go vs. 35 in chess).**

Because of this, the total number of possible games of Go has been estimated at 10^{761}, compared to 10^{120} for chess.

GO VS CHESS:

Game Board





19X19 lines vs. 64 squares (8 rows and 8 columns)

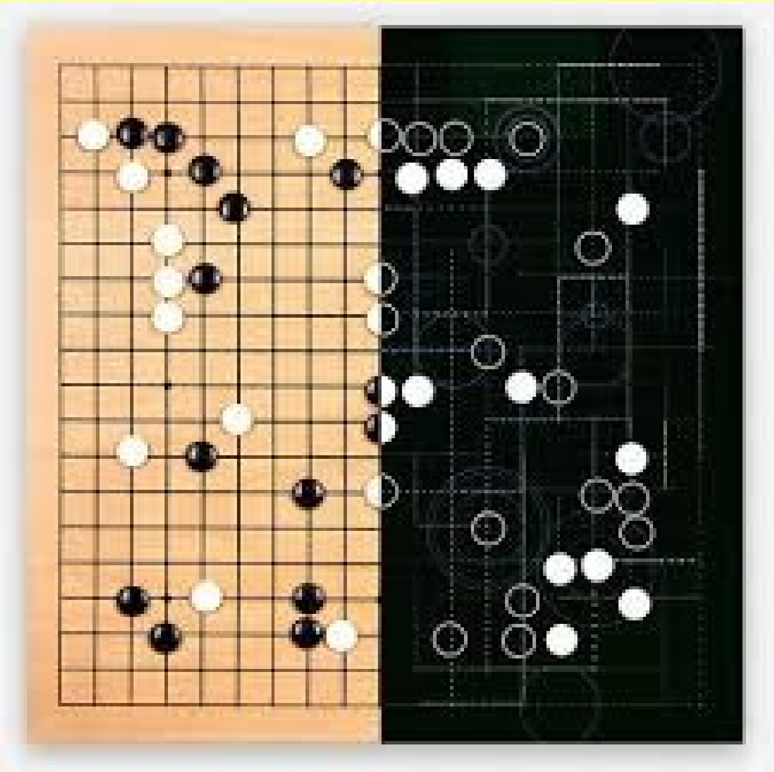
What's interesting about Go?

-**Has been viewed as the most challenging of classic games for Artificial Intelligence** owing to its enormous search space and the difficulty of evaluating board positions and moves.
-**Despite its relatively simple rules, Go is very complex**, even more so than chess, and possesses more possibilities than the total number of atoms in the visible universe.

-Number of atoms=10^80;
-Number of possible games of chess=10^120;
-Number of possible games of Go game= 10^761;

AlphaGo

Google DeepMind



What Is ALPHAGO?

- **AlphaGo**, an **AI computer program** which was developed by **Alphabet Inc.'s Google DeepMind** in London in October 2015,**became the first Computer Go program** to beat a human professional Go player on a official-sized 19×19 board.

- **Instead of using brute force, alphago used reinforcement learning and Neural Network** to mimic the leaning process of human mind.

- By combining deep learning and reinforcement learning in a series of artificial neural networks, **AlphaGo first learned human expert-level play in Go from 30 million moves from human games**.

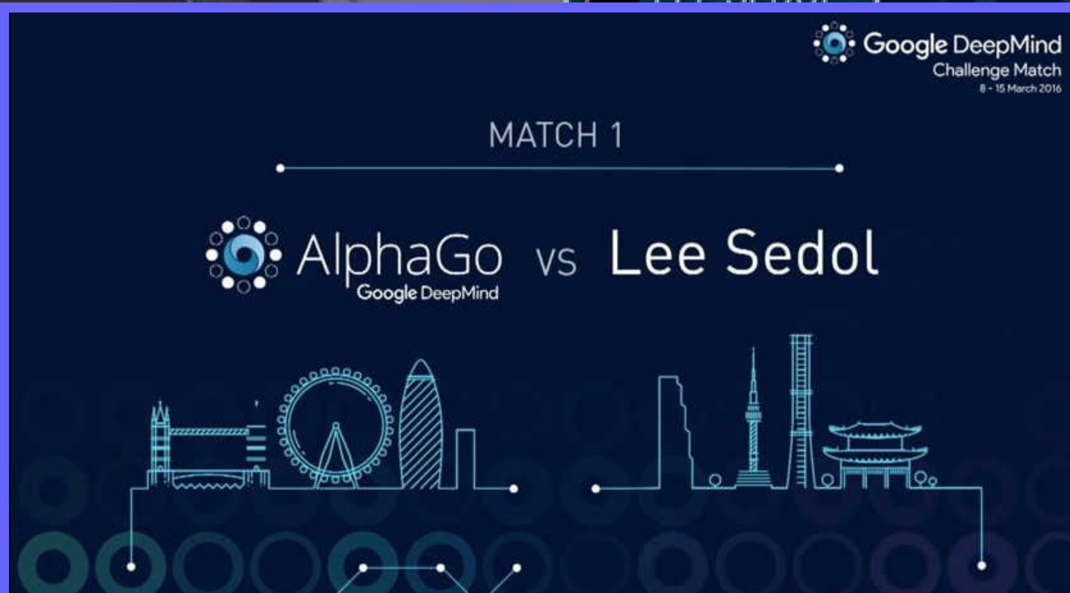
- But then it started playing against itself, using the outcome of each game to relentlessly refine its decisions about the best move in each board position.

- **Two networks are used**,
  - A **value network** learned which is used to predict the likely outcome given any position,
  - while a **policy network** learned the best action to take in each situation.
- Also much of AlphaGo’s power is based on a technique called back-propagation learning that helps it correct errors

- **Alphago is made up of number of relatively standard techniques:-**

  - Convolution Neural Network
  - Supervised Learning
  - Reinforcement learning
  - Monte Carlo Tree Search
  - Value Network
  - Policy Network and Fast Policy Network


ALPHAGO
00:20:49
Google DeepMind
Challenge Match
LEE SEDOL


Google DeepMind
Challenge Match
8 - 15 March 2016
MATCH 1
AlphaGo
Google DeepMind
vs
Lee Sedol

Study of the Google's Official Paper...

1.Mastering the game of Go with deep neural networks and tree searchhttps://storage.googleapis.com/deepmind-media/alphago/AlphaGoNaturePaper.pdf

2. Google Research Blog :
https://research.googleblog.com/2016/01/alphago-mastering-ancient-game-of-go.html

3. And also various articles and Blogs.

4. Relevant Courses on :

- Machine Learning Specialization by University of Washington.
- Deep Learning and Neural Networks by Co-founder of Coursera.
- Game Theory by David Silver.

Deep Mind’s AlphaGo Official Site

Nature Paper published on 28 Jan 2016 :
https://storage.googleapis.com/deepmind-media/alphago/AlphaGoNaturePaper.pdf

Summary of the Research paper :

- New approach: uses **‘Value Networks’** to evaluate board positions and **‘Policy Networks’** to select moves.
- Combination of **Supervised Learning and Reinforcement Learning**
- New search algo: combines Monte Carlo simulation with value and policy networks (lookahead search)
- Using this search algo, **AlphaGo achieved 99.8% winning rate.**

- AlphaGo uses a SL policy to initialize the learning of an RL policy that gets perfected with self play, which they then estimate a value function from, which then plugs into MCTS that uses the SL policy to sample rollouts.

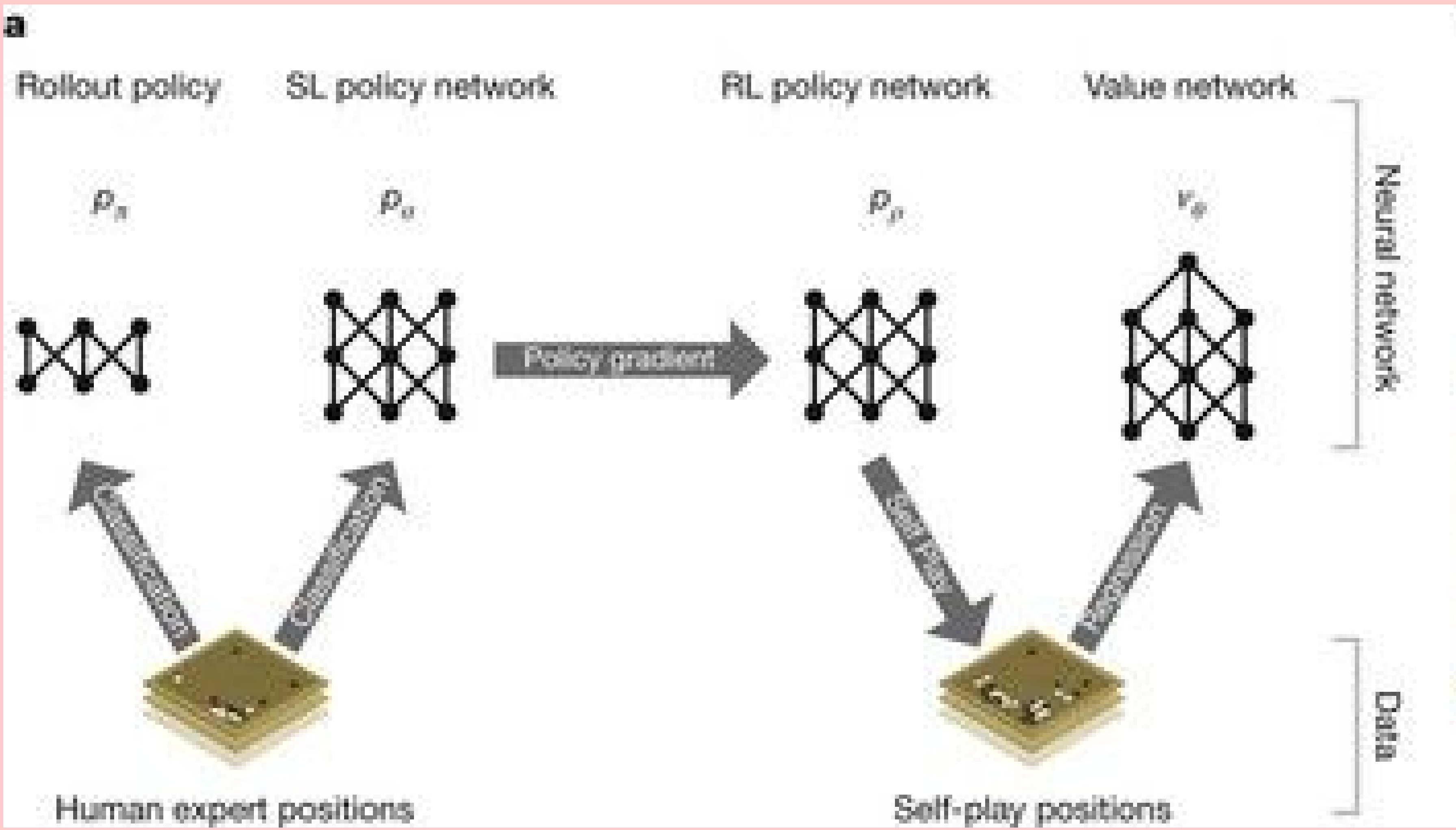
- Optimal Function : determines the outcomes of the game from every board position or state s.

- This optimal value function can be solved by traversing a search tree containing approx. b^d sequence.
  Where b-game's breadth (no. of legal moves per position)
  d-depth (game length)

- AlphaGo combines Monte Carlo simulations with value and policy networks called "Asynchronous policy and value MCTS" (APV-MCTS).

Training Pipeline

- Supervised Learning of policy network
- Reinforcement Learning of policy network
- Reinforcement Learning of value network



- SL policy performed better in APV-MCTS than the stronger RL policy because SL policy better reflect the diverse beam of promising moves humans select, whereas RL optimizes for a single best move.
- In case of value function, the value function derived from the stronger RL policy network perform better than the value function derived from SL policy network.

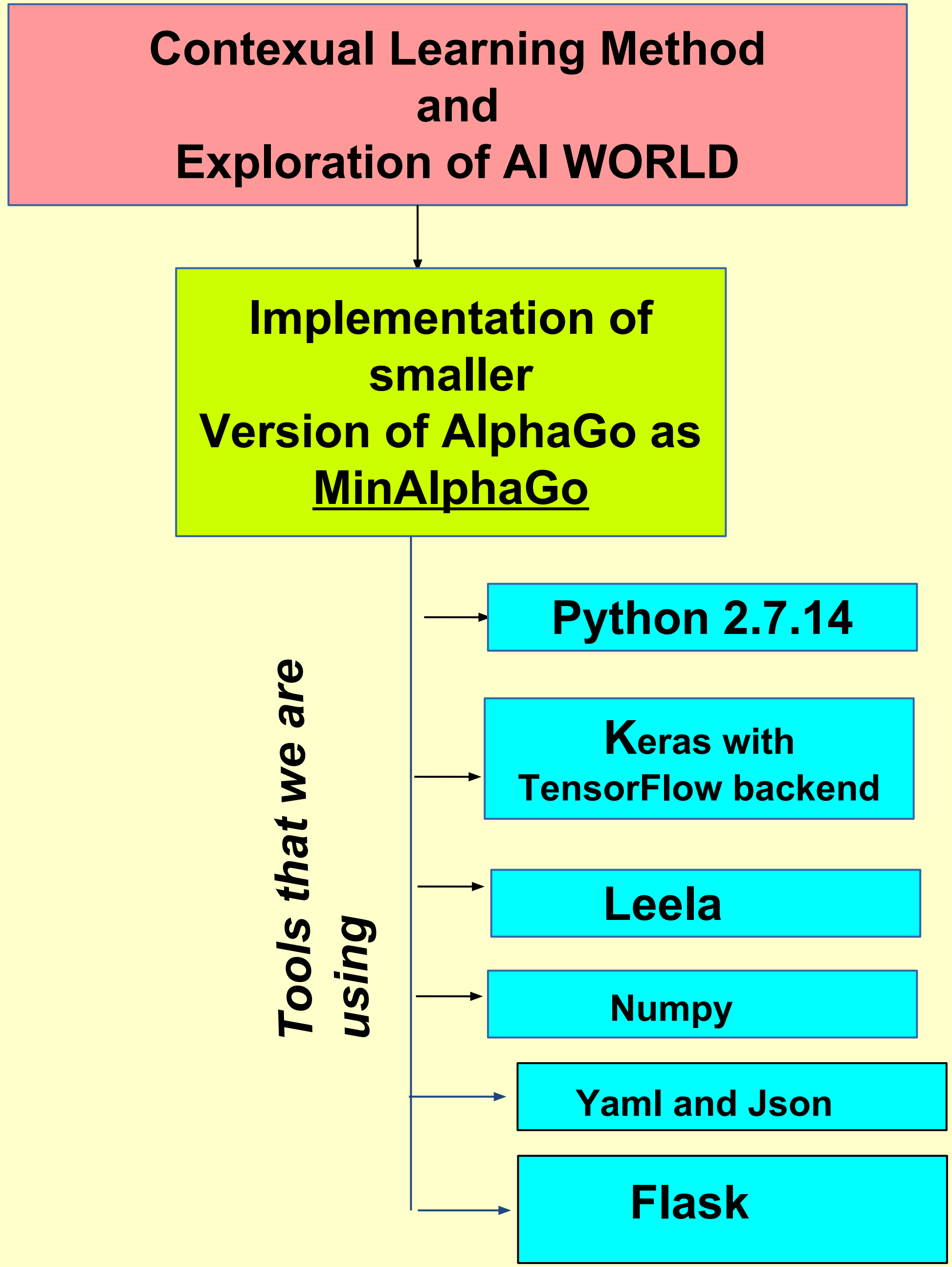
Our Contribution…

Latest Works :

- BetaGO
- Leela Game Engine
- Rochester AlphaGo

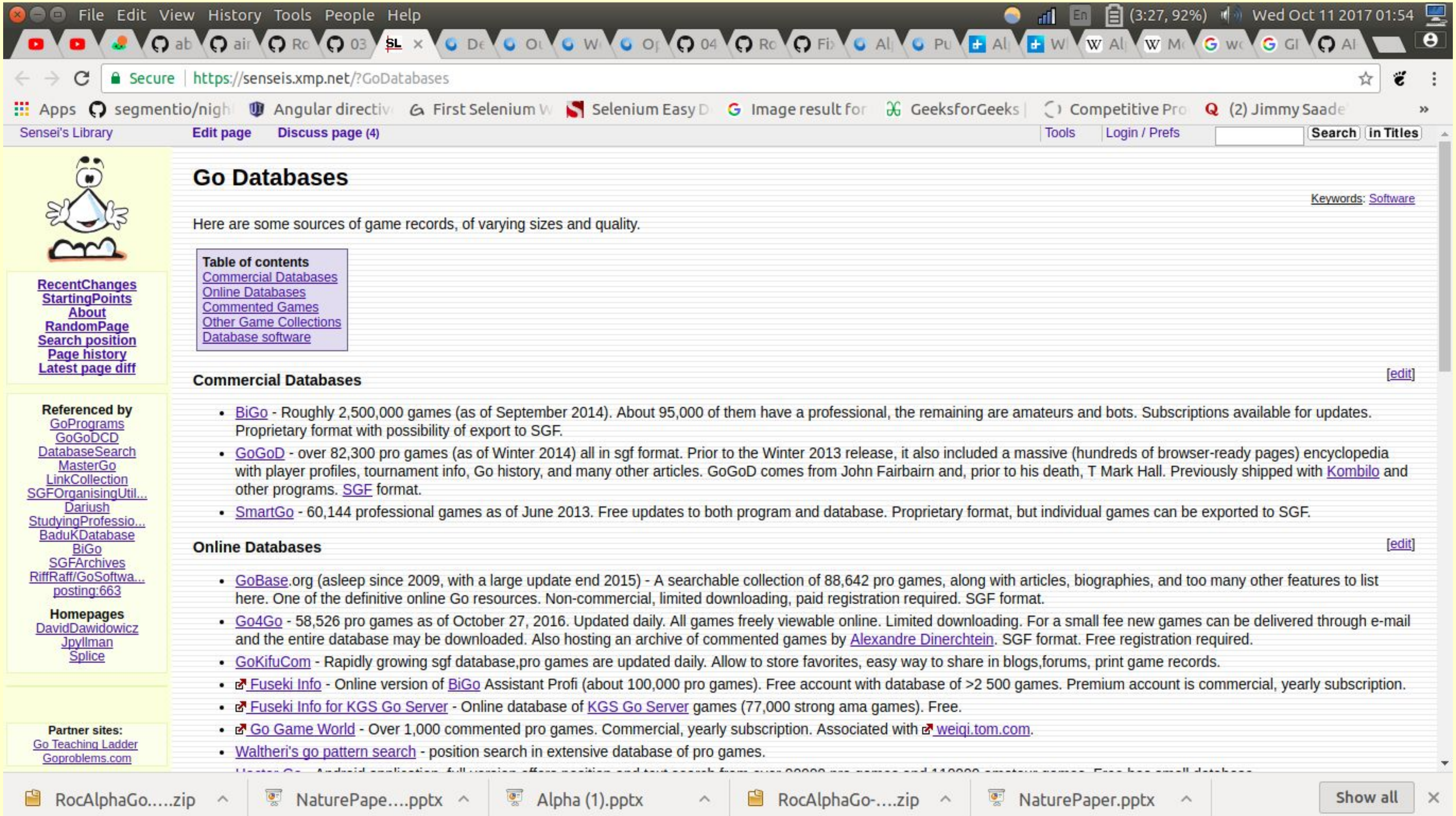
We took ‘BetaGo’ AND ‘Leela Game Engine’ into consideration and have joined the plus points of the implementation and have come out with our newer version named as ‘MinAlphaGo’

PHASE 1

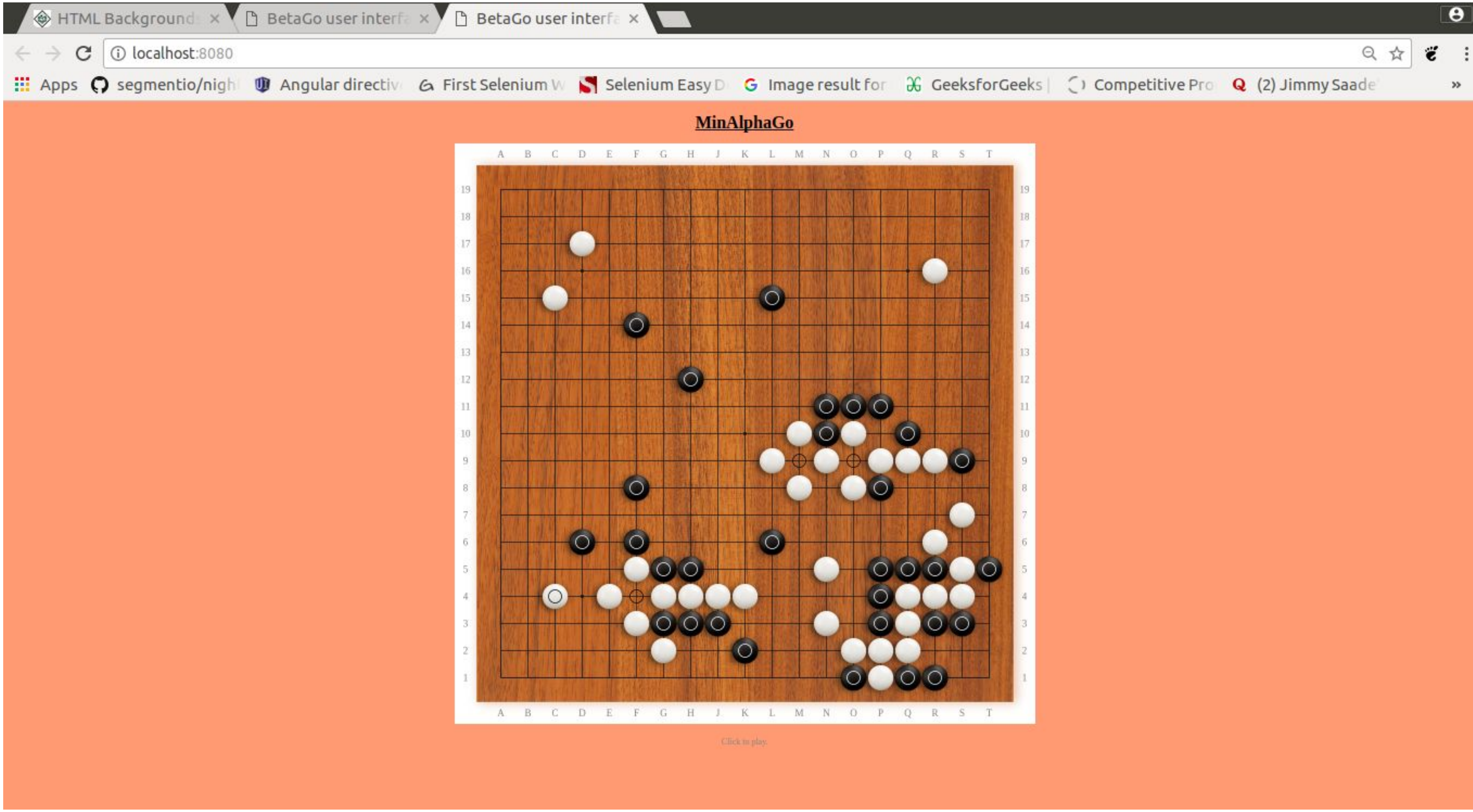

Contexual Learning Method
and
Exploration of AI WORLD
Implementation of smaller
Version of AlphaGo as
MinAlphaGo
Tools that we are using
Python 2.7.14
Keras with
TensorFlow backend
Leela
Numpy
Yaml and Json
Flask

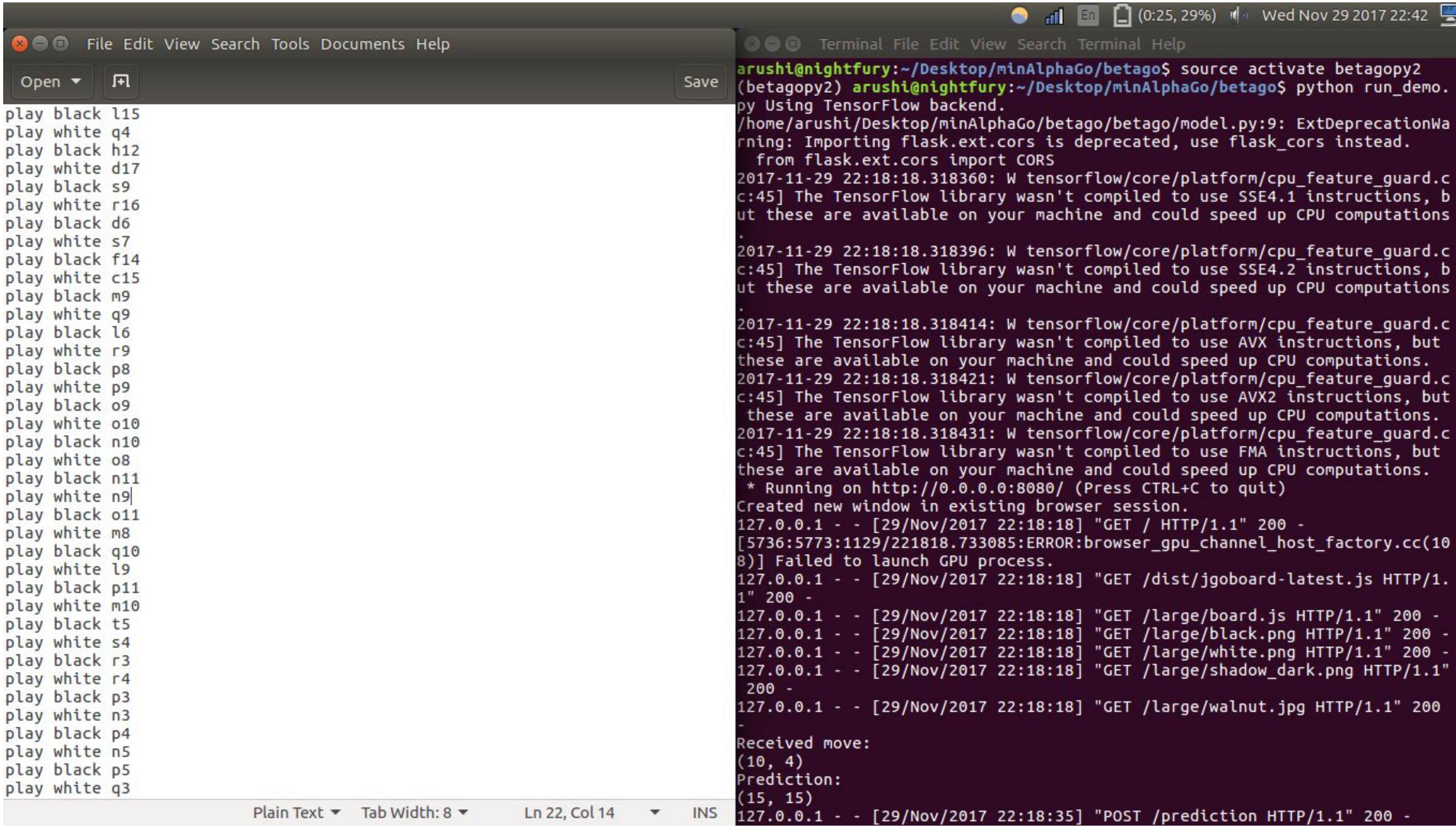
Data Source

https://senseis.xmp.net/?GoDatabases

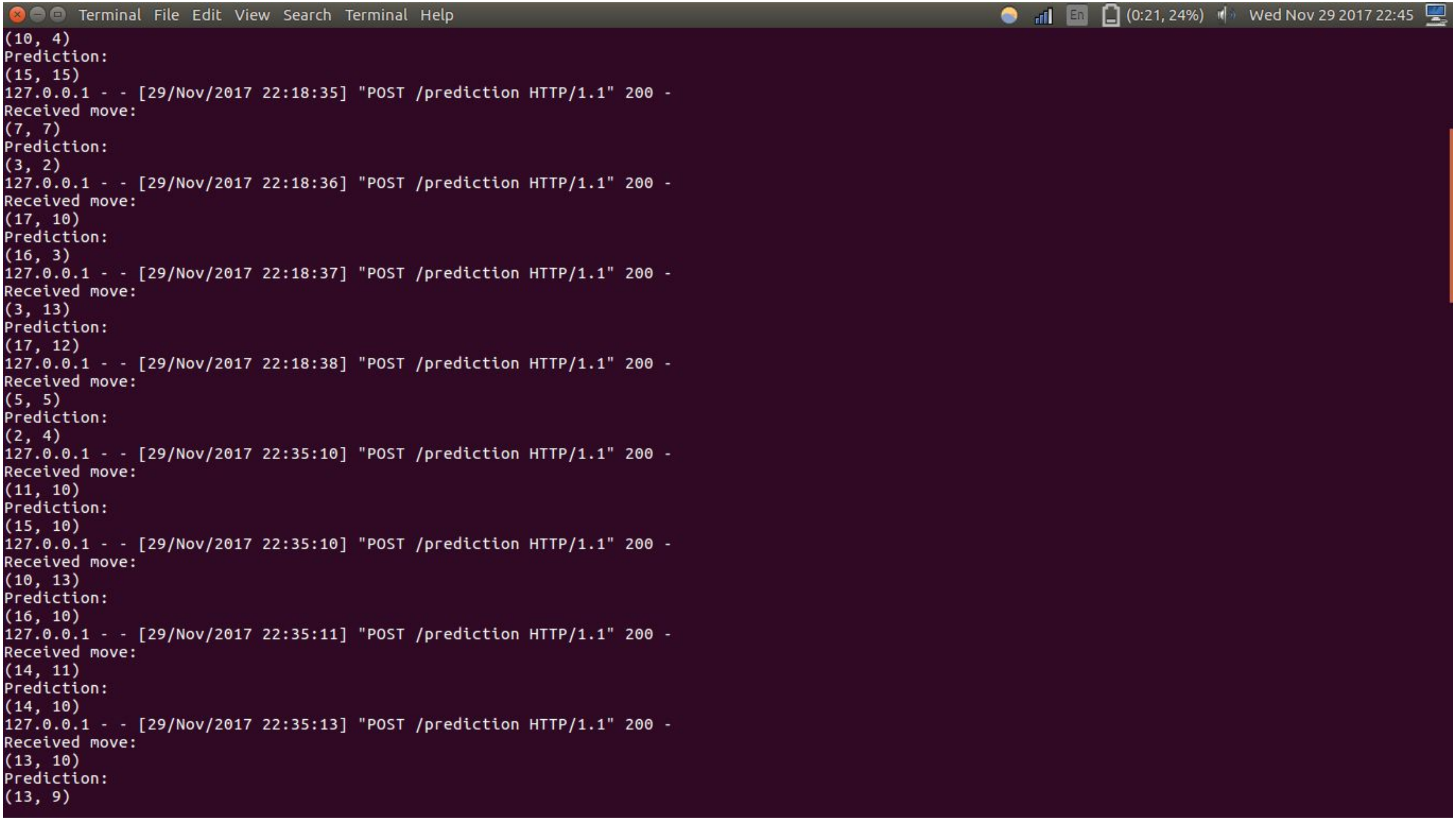


ANALYSIS AND RESULTS

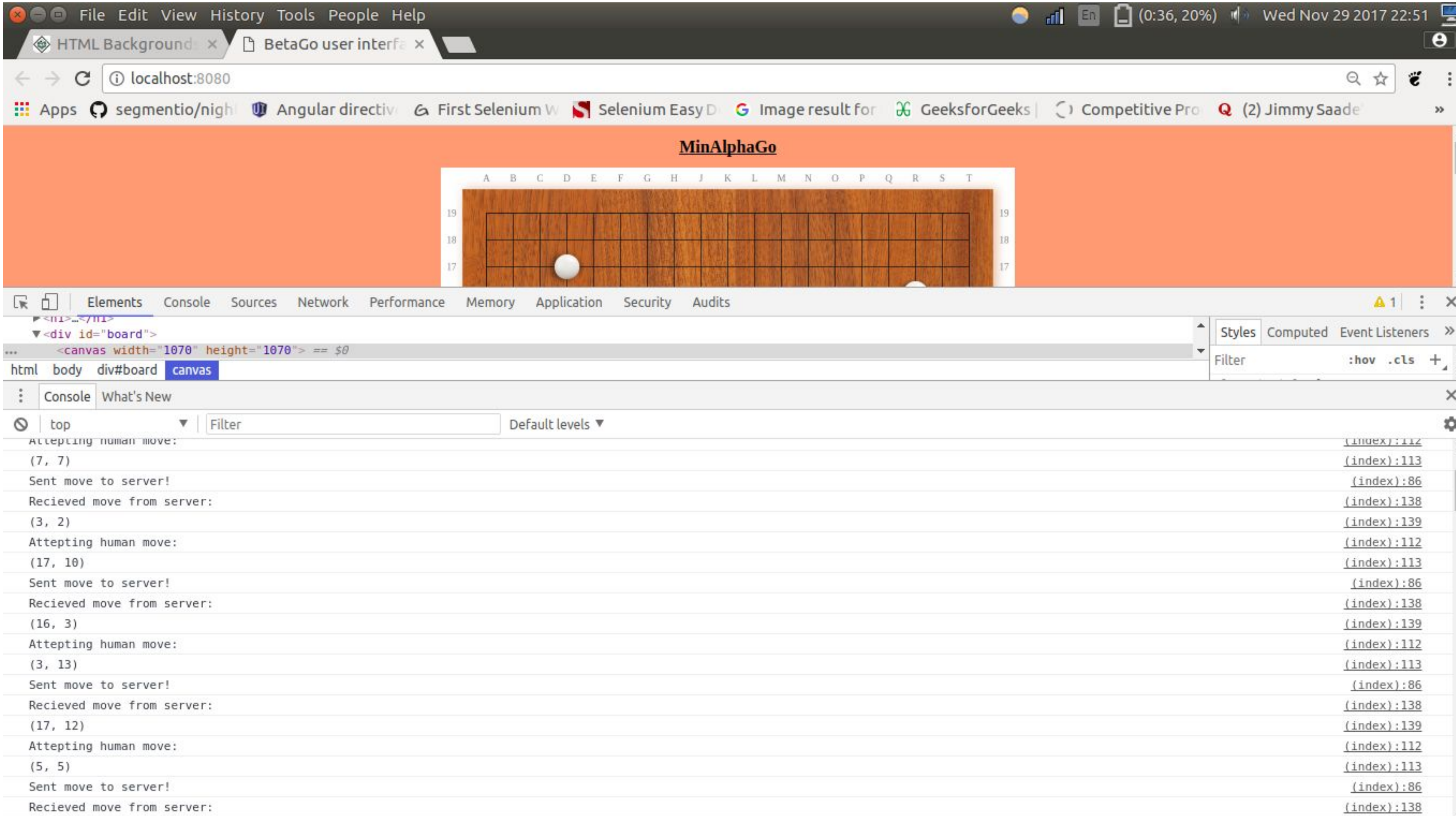

HTML Background
BetaGo user interf
BetaGo user interf
localhost:8080
Apps
segmentio/nigh
Angular directiv
First Selenium W
Selenium Easy D
Image result for
GeeksforGeeks
Competitive Pro
(2) Jimmy Saade
MinAlphaGo
A B C D E F G H J K L M N O P Q R S T
19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
Click to play.



(0:25, 29%) Wed Nov 29 2017 22:42
File Edit View Search Tools Documents Help
Open
Save
play black l15
play white q4
play black h12
play white d17
play black s9
play white r16
play black d6
play white s7
play black f14
play white c15
play black m9
play white q9
play black l6
play white r9
play black p8
play white p9
play black o9
play white o10
play black n10
play white o8
play black n11
play white n9
play black o11
play white m8
play black q10
play white l9
play black p11
play white m10
play black t5
play white s4
play black r3
play white r4
play black p3
play white n3
play black p4
play white n5
play black p5
play white q3
Plain Text
Tab Width: 8
Ln 22, Col 14
INS
Terminal File Edit View Search Terminal Help
arushi@nightfury:~/Desktop/minAlphaGo/betago$ source activate betagopy2
(betagopy2) arushi@nightfury:~/Desktop/minAlphaGo/betago$ python run_demo.py Using TensorFlow backend.
/home/arushi/Desktop/minAlphaGo/betago/betago/model.py:9: ExtDeprecationWarning: Importing flask.ext.cors is deprecated, use flask_cors instead.
from flask.ext.cors import CORS
2017-11-29 22:18:18.318360: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.1 instructions, but these are available on your machine and could speed up CPU computations.
2017-11-29 22:18:18.318396: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.2 instructions, but these are available on your machine and could speed up CPU computations.
2017-11-29 22:18:18.318414: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX instructions, but these are available on your machine and could speed up CPU computations.
2017-11-29 22:18:18.318421: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX2 instructions, but these are available on your machine and could speed up CPU computations.
2017-11-29 22:18:18.318431: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use FMA instructions, but these are available on your machine and could speed up CPU computations.
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
Created new window in existing browser session.
127.0.0.1 - - [29/Nov/2017 22:18:18] "GET / HTTP/1.1" 200 -
[5736:5773:1129/221818.733085:ERROR:browser_gpu_channel_host_factory.cc(108)] Failed to launch GPU process.
127.0.0.1 - - [29/Nov/2017 22:18:18] "GET /dist/jgoboard-latest.js HTTP/1.1" 200 -
127.0.0.1 - - [29/Nov/2017 22:18:18] "GET /large/board.js HTTP/1.1" 200 -
127.0.0.1 - - [29/Nov/2017 22:18:18] "GET /large/black.png HTTP/1.1" 200 -
127.0.0.1 - - [29/Nov/2017 22:18:18] "GET /large/white.png HTTP/1.1" 200 -
127.0.0.1 - - [29/Nov/2017 22:18:18] "GET /large/shadow_dark.png HTTP/1.1" 200 -
127.0.0.1 - - [29/Nov/2017 22:18:18] "GET /large/walnut.jpg HTTP/1.1" 200 -
Received move:
(10, 4)
Prediction:
(15, 15)
127.0.0.1 - - [29/Nov/2017 22:18:35] "POST /prediction HTTP/1.1" 200 -


Terminal File Edit View Search Terminal Help
En (0:21, 24%) Wed Nov 29 2017 22:45
(10, 4)
Prediction:
(15, 15)
127.0.0.1 - - [29/Nov/2017 22:18:35] "POST /prediction HTTP/1.1" 200 -
Received move:
(7, 7)
Prediction:
(3, 2)
127.0.0.1 - - [29/Nov/2017 22:18:36] "POST /prediction HTTP/1.1" 200 -
Received move:
(17, 10)
Prediction:
(16, 3)
127.0.0.1 - - [29/Nov/2017 22:18:37] "POST /prediction HTTP/1.1" 200 -
Received move:
(3, 13)
Prediction:
(17, 12)
127.0.0.1 - - [29/Nov/2017 22:18:38] "POST /prediction HTTP/1.1" 200 -
Received move:
(5, 5)
Prediction:
(2, 4)
127.0.0.1 - - [29/Nov/2017 22:35:10] "POST /prediction HTTP/1.1" 200 -
Received move:
(11, 10)
Prediction:
(15, 10)
127.0.0.1 - - [29/Nov/2017 22:35:10] "POST /prediction HTTP/1.1" 200 -
Received move:
(10, 13)
Prediction:
(16, 10)
127.0.0.1 - - [29/Nov/2017 22:35:11] "POST /prediction HTTP/1.1" 200 -
Received move:
(14, 11)
Prediction:
(14, 10)
127.0.0.1 - - [29/Nov/2017 22:35:13] "POST /prediction HTTP/1.1" 200 -
Received move:
(13, 10)
Prediction:
(13, 9)

```
Terminal File Edit View Search Terminal Help        (0:40, 22%)  Wed Nov 29 2017 22:46
 1 . . . . . . . . . . . . . X O X X . .  1
    a b c d e f g h j k l m n o p q r s t

Hash: D8A0D2862DAE1754 Ko-Hash: 9F0CC4C773E9D20D

Black time: 00:30:00
White time: 00:30:00

Leela:
=


Passes: 0            Black (X) Prisoners: 0
Black (X) to move    White (O) Prisoners: 3

   a b c d e f g h j k l m n o p q r s t
19 . . . . . . . . . . . . . . . . . . . 19
18 . . . . . . . . . . . . . . . . . . . 18
17 . . . O . . . . . . . . . . . . . . . 17
16 . . . + . . . . . + . . . . . + O . . 16
15 . . O . . . . . . . X . . . . . . . . 15
14 . . . . . X . . . . . . . . . . . . . 14
13 . . . . . . . . . . . . . . . . . . . 13
12 . . . . . . . X . . . . . . . . . . . 12
11 . . . . . . . . . . . . X X X . . . . 11
10 . . . + . . . . . + . O X O . X . . . 10
 9 . . . . . . . . . . O . O . O O O X .  9
 8 . . . . . X . . . . . O . O X . . . .  8
 7 . . . . . . . . . . . . . . . . . O .  7
 6 . . . X . X . . . . X . . . . . O . .  6
 5 . . . . . O X X . . . . O . X X X O X  5
 4 . .(O)+ O . O O O O . . . . X O O O .  4
 3 . . . . . O X X X . . . O . X O X X .  3
 2 . . . . . . O . . X . . . O O O . . .  2
 1 . . . . . . . . . . . . . X O X X . .  1
   a b c d e f g h j k l m n o p q r s t

Hash: 3E414760141E406C Ko-Hash: D220FAECE1942EF8

Black time: 00:30:00
White time: 00:30:00

Leela:
```


File Edit View History Tools People Help
(0:36, 20%)
Wed Nov 29 2017 22:51
HTML Background
BetaGo user interf
localhost:8080
Apps
segmentio/nigh
Angular directiv
First Selenium W
Selenium Easy D
Image result for
GeeksforGeeks
Competitive Pro
(2) Jimmy Saade
MinAlphaGo
A B C D E F G H J K L M N O P Q R S T
19
18
17
Elements Console Sources Network Performance Memory Application Security Audits
<div id="board">
<canvas width="1070" height="1070"> == $0
html body div#board canvas
Styles Computed Event Listeners
Filter :hov .cls
Console What's New
top
Filter
Default levels
(7, 7) (index):113
Sent move to server! (index):86
Recieved move from server: (index):138
(3, 2) (index):139
Attepting human move: (index):112
(17, 10) (index):113
Sent move to server! (index):86
Recieved move from server: (index):138
(16, 3) (index):139
Attepting human move: (index):112
(3, 13) (index):113
Sent move to server! (index):86
Recieved move from server: (index):138
(17, 12) (index):139
Attepting human move: (index):112
(5, 5) (index):113
Sent move to server! (index):86
Recieved move from server: (index):138

```
Terminal  File  Edit  View  Search  Terminal  Help                    En  (0:12, 14%)  Wed Nov 29 2017 22:56
 9 . . . . . . . . . . O . O . O O O X .  9
 8 . . . . . X . . . . . O . O X . . . .  8
 7 . . . . . . . . . . . . . . . . . O .  7
 6 . . . X . X . . . . X . . . . . O . .  6
 5 . . . . . O X X . . . . O . X X X O X  5
 4 . .(O)+ O . O O O O . . . . X O O O .  4
 3 . . . . . O X X X . . . O . X O X X .  3
 2 . . . . . . O . . X . . . O O O . . .  2
 1 . . . . . . . . . . . . . X O X X . .  1
   a b c d e f g h j k l m n o p q r s t

Hash: 3E414760141E406C Ko-Hash: D220FAECE1942EF8

Black time: 00:30:00
White time: 00:30:00

Leela: genmove black
MC winrate=0.000000, NN eval=0.000044, score=W+85.4
Nodes: 2961, Win:  1.26% (MC: 3.20%/VN: 0.00%), PV: H2 F2 L3 L4
Nodes: 6936, Win:  1.37% (MC: 3.46%/VN: 0.00%), PV: H2 F2 L3 L4 M2 J1
Nodes: 10679, Win:  1.40% (MC: 3.55%/VN: 0.00%), PV: H2 F2 L3 L4 M2 J1 J2
Allowing early exit: score: 1.509197%

  C5 ->    3190 (W:  1.51%) (U:  3.81%) (V:  0.00%:     60) (N: 23.6%) PV: C5 B4 C13 G17 P17 Q17 P16
  H2 ->    2149 (W:  1.51%) (U:  3.81%) (V:  0.00%:     44) (N: 16.0%) PV: H2 F2 L3 L4 M2 J1 J2
 C13 ->    2009 (W:  1.43%) (U:  3.61%) (V:  0.00%:     38) (N: 16.6%) PV: C13 B6 F17 E15 F15 D13 C12
  E5 ->    1423 (W:  1.42%) (U:  3.59%) (V:  0.00%:     31) (N: 11.8%) PV: E5 F4 C13 G17 C5
 P17 ->    1337 (W:  1.45%) (U:  3.67%) (V:  0.00%:     23) (N: 10.7%) PV: P17 P16 Q16 Q15 Q17
  J6 ->     582 (W:  1.30%) (U:  3.28%) (V:  0.00%:     12) (N:  5.6%) PV: J6 P17 C13
 F17 ->     501 (W:  1.79%) (U:  4.52%) (V:  0.00%:     10) (N:  3.1%) PV: F17 P17 R14
 P16 ->     210 (W:  1.26%) (U:  3.18%) (V:  0.00%:      3) (N:  2.3%) PV: P16 P17 O17
  L4 ->     176 (W:  1.36%) (U:  3.42%) (V:  0.00%:      3) (N:  1.9%) PV: L4 K3
  D3 ->     112 (W:  2.77%) (U:  7.00%) (V:  0.00%:      3) (N:  0.5%) PV: D3 C3
  M4 ->      94 (W: -0.47%) (U: -1.18%) (V:  0.00%:      2) (N:  2.4%) PV: M4 M3
  F2 ->      63 (W:  0.81%) (U:  2.05%) (V:  0.00%:      1) (N:  1.1%) PV: F2
====================================
3190 visits, score  1.51% (from  1.46%) PV: C5 B4 C13 G17 P17 Q17 P16

11907 visits, 3537 nodes, 11907 playouts, 1378 p/s

= C5
```

Terminal File Edit View Search Terminal Help (0:12, 14%) Wed Nov 29 2017 22:57

```
 3 . . . . . O X X X . . . O . X O X X .  3
 2 . . . . . . O . . X . . . O O O . . .  2
 1 . . . . . . . . . . . . . X O X X . .  1
   a b c d e f g h j k l m n o p q r s t

Hash: BACA27BB75F70531 Ko-Hash: FD6631FA2BB0C068

Black time: 00:29:51
White time: 00:30:00

Leela: =

   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   3   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0  38   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   3   0   5   0   0   0   0   0   0   0   0   0   0   0   0
   0  34   0  14   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0 888   0   4   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0

Passes: 0            Black (X) Prisoners: 0
White (O) to move    White (O) Prisoners: 3

   a b c d e f g h j k l m n o p q r s t
19 . . . . . . . . . . . . . . . . . . . 19
18 . . . . . . . . . . . . . . . . . . . 18
17 . . . O . . . . . . . . . . . . . . . 17
16 . . . + . . . . . + . . . . . + O . . 16
15 . . O . . . . . . . X . . . . . . . . 15
14 . . . . . X . . . . . . . . . . . . . 14
13 . . . . . . . . . . . . . . . . . . . 13
```

```
   a b c d e f g h j k l m n o p q r s t

Hash: BACA27BB75F70531 Ko-Hash: FD6631FA2BB0C068

Black time: 00:29:51
White time: 00:30:00

Leela: play white c5
=


Passes: 0             Black (X) Prisoners: 0
Black (X) to move     White (O) Prisoners: 3

   a b c d e f g h j k l m n o p q r s t
19 . . . . . . . . . . . . . . . . . . . 19
18 . . . . . . . . . . . . . . . . . . . 18
17 . . . O . . . . . . . . . . . . . . . 17
16 . . . + . . . . . + . . . . . + O . . 16
15 . . O . . . . . . . X . . . . . . . . 15
14 . . . . . X . . . . . . . . . . . . . 14
13 . . . . . . . . . . . . . . . . . . . 13
12 . . . . . . . X . . . . . . . . . . . 12
11 . . . . . . . . . . . . X X X . . . . 11
10 . . . + . . . . . + . O X O . X . . . 10
 9 . . . . . . . . . . O . O . O O O X .  9
 8 . . . . . X . . . . . O . O X . . . .  8
 7 . . . . . . . . . . . . . . . . . O .  7
 6 . . . X . X . . . . X . . . . . O . .  6
 5 . .(O). . O X X . . . . O . X X X O X  5
 4 . . O + O . O O O O . . . . X O O O .  4
 3 . . . . . O X X X . . . O . X O X X .  3
 2 . . . . . . O . . X . . . O O O . . .  2
 1 . . . . . . . . . . . . . X O X X . .  1
   a b c d e f g h j k l m n o p q r s t

Hash: C80501E44C5BFEDC Ko-Hash: 2464BC68B9D19048

Black time: 00:29:51
White time: 00:30:00

Leela: mc_score
= W+12.5
```

Move	Effort%	Simulations	Win%	MC Win%	Net Win%	Net Prob%	PV
S7	37.18	12414	6.31	16.92	0.10	1.26	S7 M
P5	22.44	7491	5.36	14.36	0.10	27.15	P5 M
M2	6.59	2201	5.42	14.50	0.11	7.66	M2 C
S3	6.26	2090	5.22	13.94	0.12	9.25	S3 P
B4	5.92	1978	5.25	14.08	0.08	8.45	B4 C
Q1	4.31	1438	5.03	13.51	0.08	7.72	Q1 M
N4	3.69	1232	5.58	14.89	0.13	3.60	N4 N
O1	3.65	1220	3.69	9.92	0.05	14.63	O1 Q
M1	1.62	540	5.56	14.88	0.12	1.85	M1 C
C4	1.53	511	5.67	15.18	0.12	2.05	C4 C
C10	1.27	424	5.66	15.13	0.12	1.38	C10
R8	1.23	410	5.03	13.44	0.12	2.39	R8 M
S4	0.77	257	4.67	12.52	0.08	2.00	S4 S
P13	0.63	209	6.20	16.58	0.13	0.52	P13
R7	0.44	146	3.85	10.19	0.14	1.66	R7 P
H3	0.37	125	**6.60**	17.74	0.09	0.24	H3 R
D18	0.31	103	5.12	13.75	0.07	0.73	D18
C6	0.26	86	4.94	13.27	0.08	0.31	C6 D
L2	0.25	83	4.70	12.63	0.07	0.72	L2 N
B5	0.22	75	4.21	11.25	0.09	0.84	B5 B
D17	0.21	71	5.01	13.30	0.16	0.58	D17
D9	0.19	64	**6.78**	18.15	0.14	0.22	D9 C
D13	0.12	41	**7.05**	18.91	0.11	0.17	D13
R17	0.10	35	6.21	16.63	0.11	0.23	R17
L16	0.09	30	2.26	5.92	0.12	0.26	L16
H17	0.07	23	1.42	3.62	0.13	0.59	H17
F17	0.07	22	-0.21	-0.75	0.10	0.73	F17
R15	0.06	19	3.74	9.87	0.15	0.32	R15
E18	0.05	18	1.73	4.54	0.08	0.26	E18
O2	0.04	14	5.12	13.77	0.06	0.19	O2

RESULTS

MinAlphaGO:

- We were able to perform the analysis part successfully.
- It predicts move faster than leela.
- Improved User Interface.

BetaGo:

- Model trained using Supervised Learning.
- UI was not implemented properly.
- No analysis part.
- Faster than Leela.

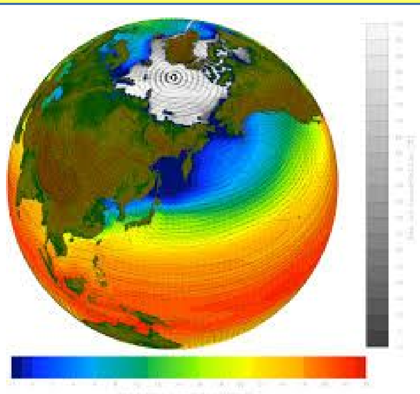
Leela:

- Has a good User Interface.
- It is comparatively slower.

SCOPE OF PROJECT

The learning done so far
will be implemented
In one of these areas ...

Climatic Modelling for Monitoring Environmental issues



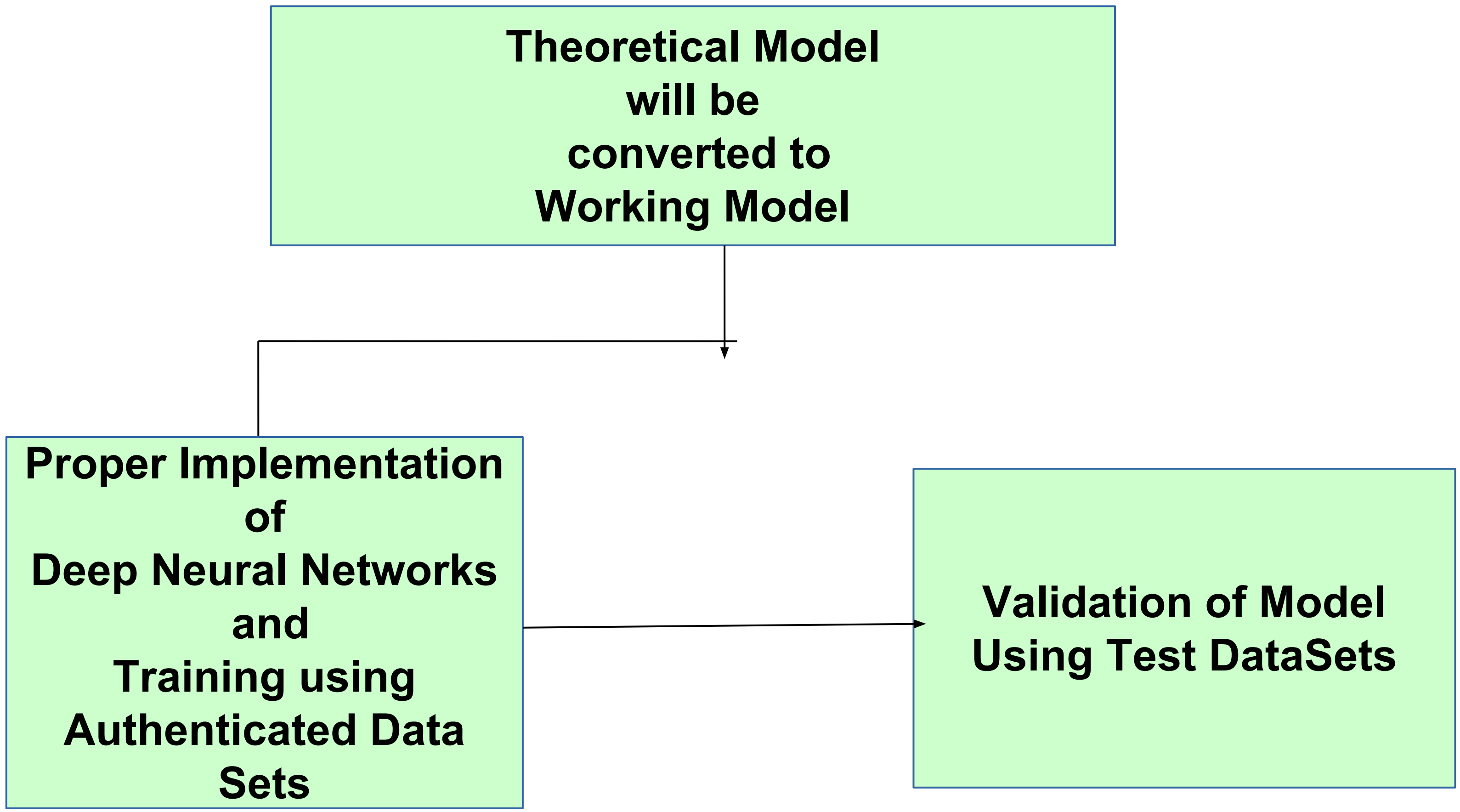
Space Explorations

Complex Disease Analysis
{Zika Virus}

FUTURE SCOPE

- MCST can also be used to refine the training process.
- Reinforcement learning along with policy and value networks can be used to train the model more efficiently.
- Different bots can be implemented.

PHASE 3


Theoretical Model
will be
converted to
Working Model
Proper Implementation
of
Deep Neural Networks
and
Training using
Authenticated Data
Sets
Validation of Model
Using Test DataSets

Improvements and Suggestions are most welcomed ...

:)