OmegaGo

A neural network trained 9x9 Go Playing Robot

Why brute force doesn't work

Possible **positions**:

9x9 board: 3^81 (4.4 x 10^38)

19x19 board: 3^361 (1.7 x 10^172)

Legal positions calculated by John Tromp

9×9 board: ~1.039 × 10^38

19×19 board: ~2.082 × 10^170

Atoms in the universe: 10^80

AlphaGo Techniques

- Neural net to predict top few moves

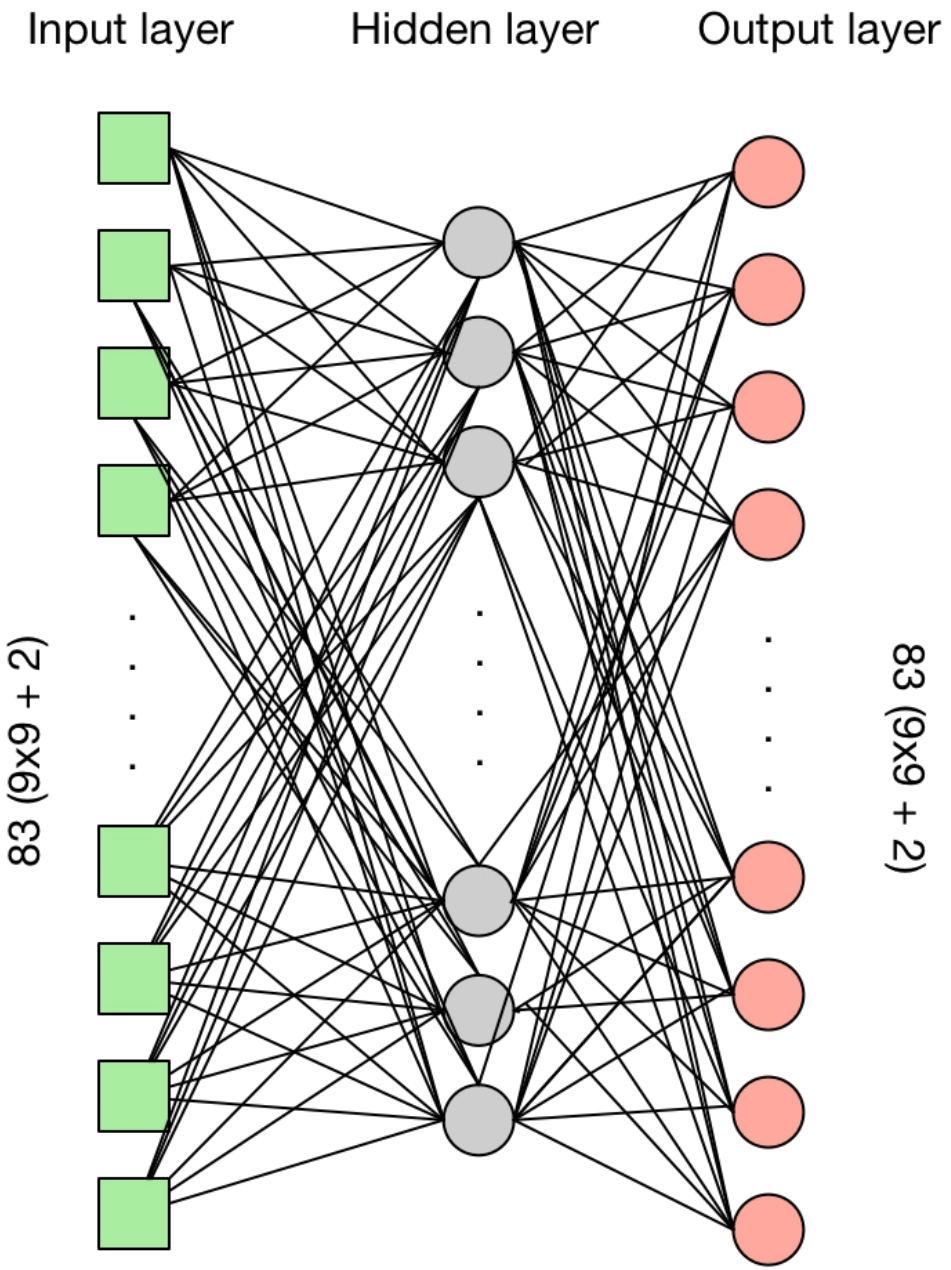
  (policy network)

- Does a Monte Carlo tree search on these top moves to determine the best one (with assistance of value network to prune the tree)

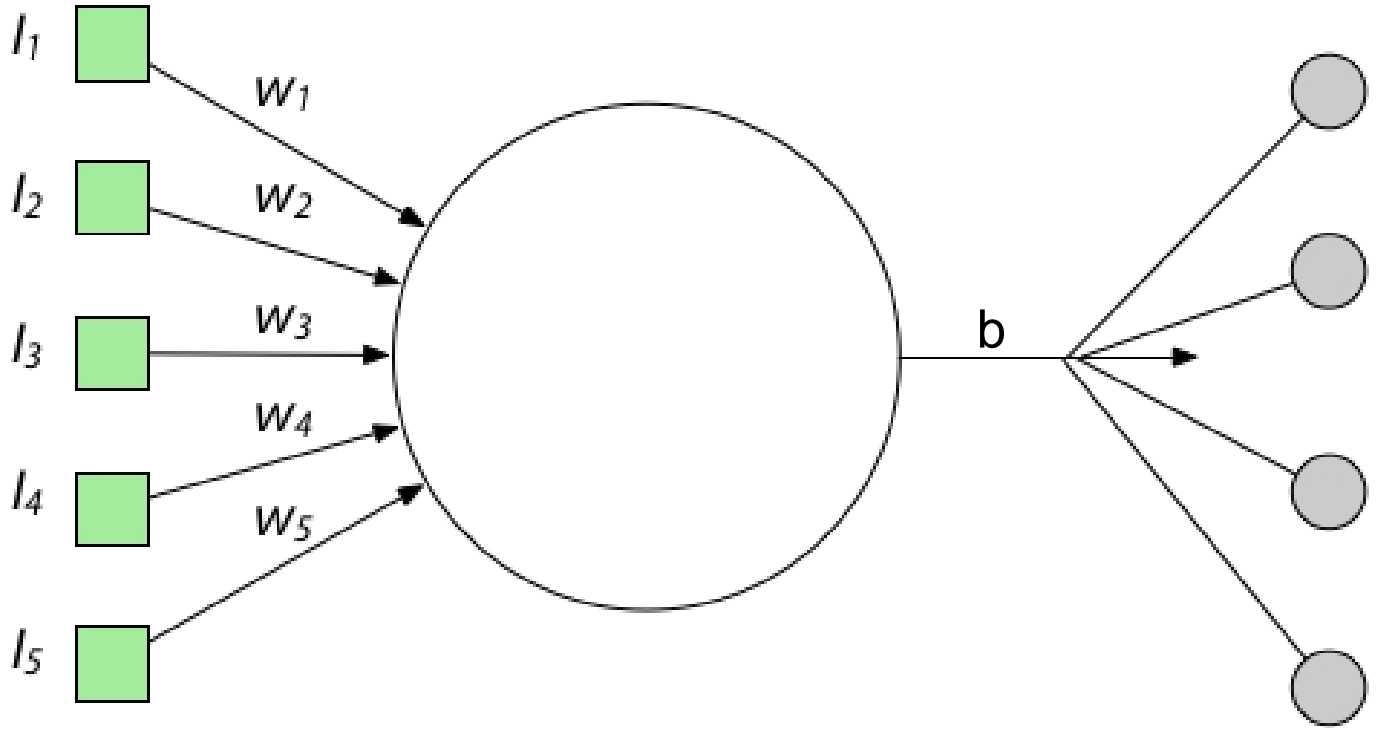
OmegaGo

- Strictly a neural network – no Monte Carlo tree search.
- OmegaGo has no knowledge of Go rules. If its first move choice is illegal it moves on to its 2nd choice.
- Trained on tens of thousands of 9x9 game records downloaded from the internet broken up into over 10 million moves.
- Games were played by players rated 5 kyu to pro

What is a neural network?

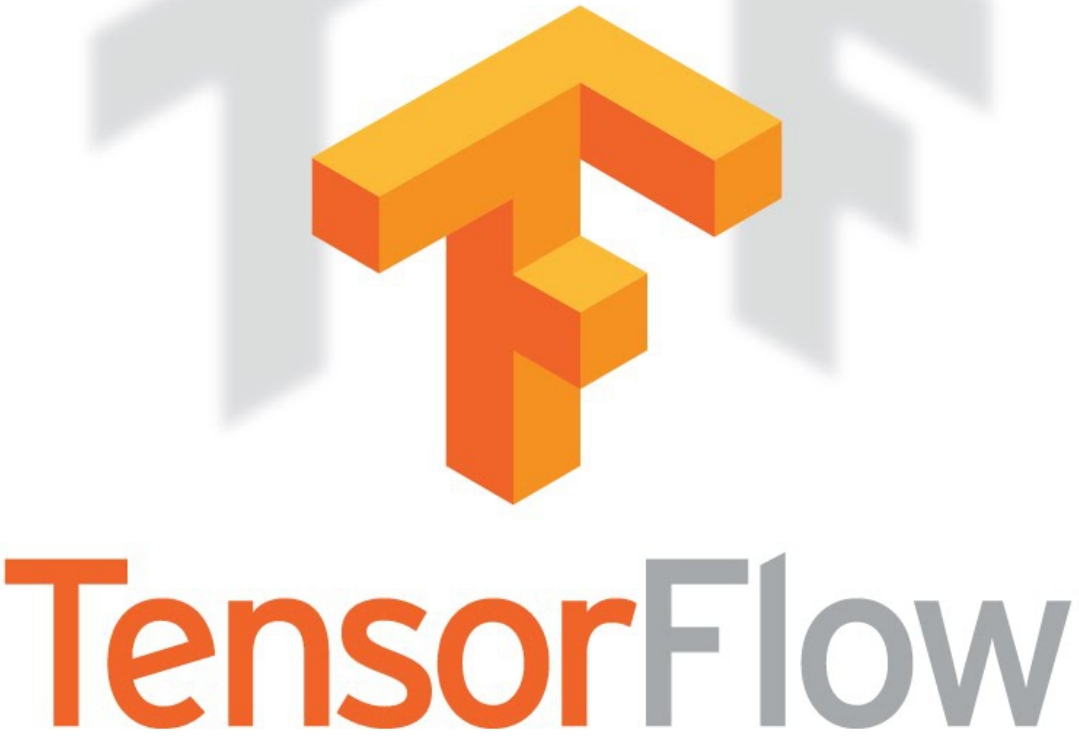


Zoomed in on one neuron



$$O = R(\sum_{j=1}^{n} w_j I_j + b)$$

TensorFlow



Feature
(Board position)

```
[[ 0  0  0  0  0  0  0  0  0]
 [ 0 -1  1  1 -1  0  0  0  0]
 [ 0  0  0  1  0 -1  0  0  0]
 [ 0  0 -1  0  1 -1  0 -1  0]
 [ 0  0  0  0  0  0  0  1 -1]
 [ 0  0  1  1  1  1  1 -1  0]
 [ 0  0  1 -1  0 -1 -1  0 -1]
 [ 0  0  1 -1  0  0  0 -1  0]
 [ 0  0  0  0  0  0  0  0  0]]
```

Label
(Next Move)

```
[[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]]
```

Feature

(Board position)

```
[[ 0  0  0  0  0  0  0  0  0]
 [ 0 -1  1  1 -1  0  0  0  0]
 [ 0  0  0  1  0 -1  0  0  0]
 [ 0  0 -1  0  1 -1  0 -1  0]
 [ 0  0  0  0  0  0  0  1 -1]
 [ 0  0  1  1  1  1  1 -1  0]
 [ 0  0  1 -1  0 -1 -1  0 -1]
 [ 0  0  1 -1  0  0  0 -1  0]
 [ 0  0  0  0  0  0  0  0  0]]
```

Label

(Next Move)

```
[[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]]
```

Prediction

```
[[ 0.    0.25  0.    0.    0.    0.    0.01  0.    0.  ]
 [ 0.    0.01  0.05  0.    0.    0.    0.01  0.    0.  ]
 [ 0.02  0.01  0.    0.02  0.    0.    0.01  0.    0.  ]
 [ 0.    0.    0.    0.01  0.    0.    0.    0.    0.  ]
 [ 0.    0.    0.01  0.02  0.01  0.01  0.2   0.01  0.  ]
 [ 0.    0.    0.    0.04  0.    0.01  0.    0.01  0.  ]
 [ 0.    0.    0.02  0.04  0.    0.    0.02  0.    0.  ]
 [ 0.01  0.03  0.    0.03  0.01  0.    0.    0.    0.07]
 [ 0.    0.    0.    0.    0.    0.    0.01  0.    0.  ]]
```

Data Sources

Data Source	Number of Games	Ratings of Players
Top 50 players from app Go Quest	6 735	ELO rating above 2000 (equals about 1 Kyu)
Amateur games from app Go Quest	16 764	ELO ratings above 2000
Pro games from Mini-Go broadcasts by Yomiuri TV	413	Pro players
No Name Go Server (NNGS nngs.cosmic.org)	1 705	5 Kyu or better
Online-go.com	2 672	5 Kyu or better
TOTAL	28 289	
TOTAL # of moves	1 398 844	
TOTAL dataset after 8 transformations (flip, rotate board)	11 190 752	

SGF files: standard go format

```
(;
GM[1]US[Brought to you by No Name Go Server]
CoPyright[
  This game was played on the No Name Go Server
  Permission to reproduce this game is given.]
GN[AdamSmith-vida(B) NNGS]
EV[None]
RE[W+2.5]
PW[AdamSmith]WR[3k*]
PB[vida]BR[5k*]
PC[NNGS New York USA: ra.york.cuny.edu 9696]
DT[1996-06-30]
SZ[9]TM[300]KM[0.5]

;B[fc];W[cg];B[cd];W[ff];B[gg];W[fg];B[gf];W[ge];B[h
e];W[gd];B[hd];W[gc];B[fd];W[gh];B[hh];W[fe];B[fh];W
[eh];B[gi];W[dc];B[dd];W[ec];B[ed];W[fb];B[cc];W[cb]
;B[db];W[eb];B[bb];W[da];B[bf];W[ba];B[ac];W[bg];B[c
f];W[af];B[ae];W[ag];B[bd];W[hc];B[dh];W[dg];B[ei];W
[eg];B[di];W[df];B[de];W[id];B[if];W[ch];B[ci];W[bi]
;B[fi];W[ee];B[ie];W[ic];B[ab];W[ca];B[tt];W[ig];B[h
f];W[tt];B[tt]
;)
```

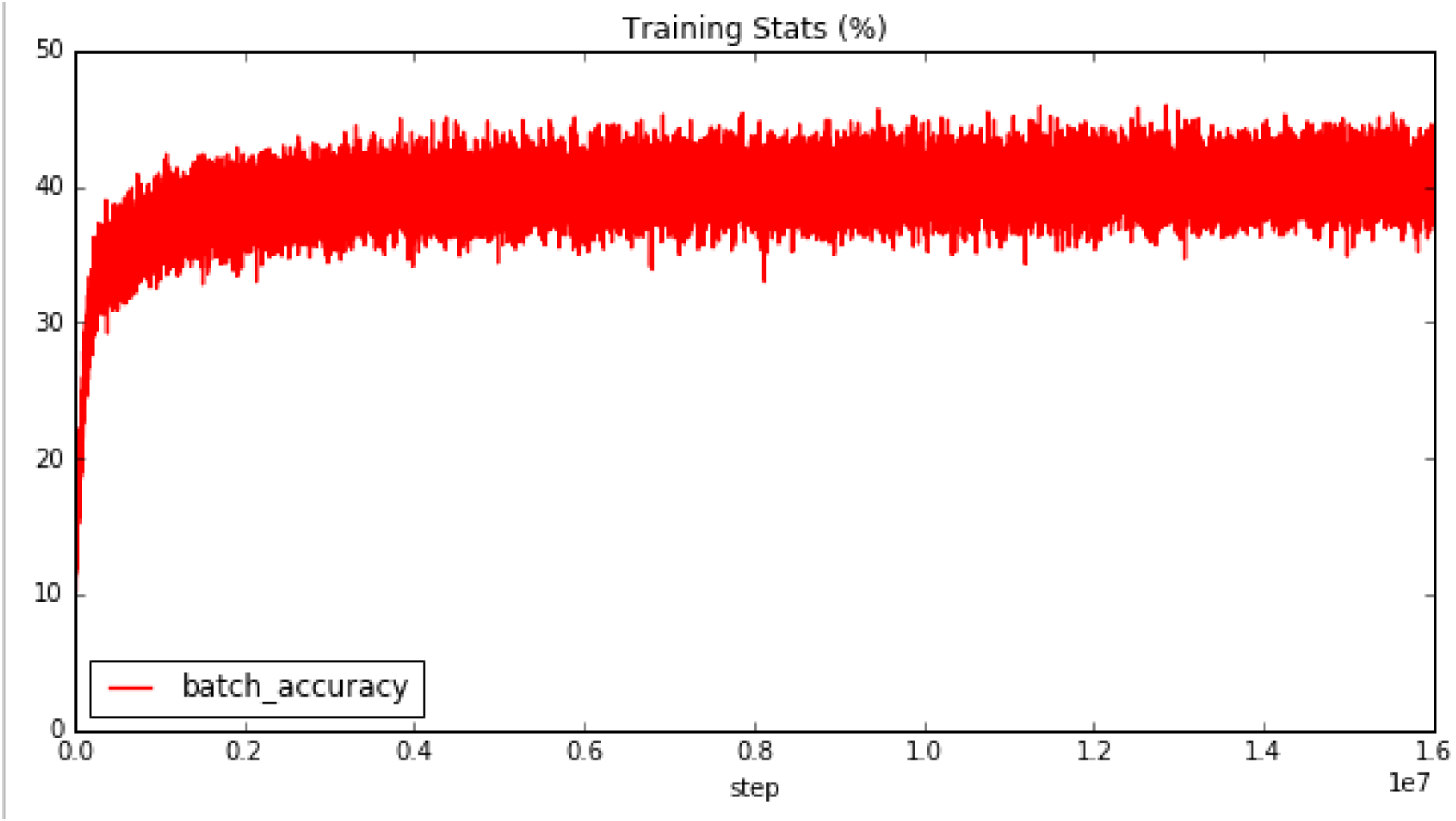
Training Results

Test accuracy: 37%

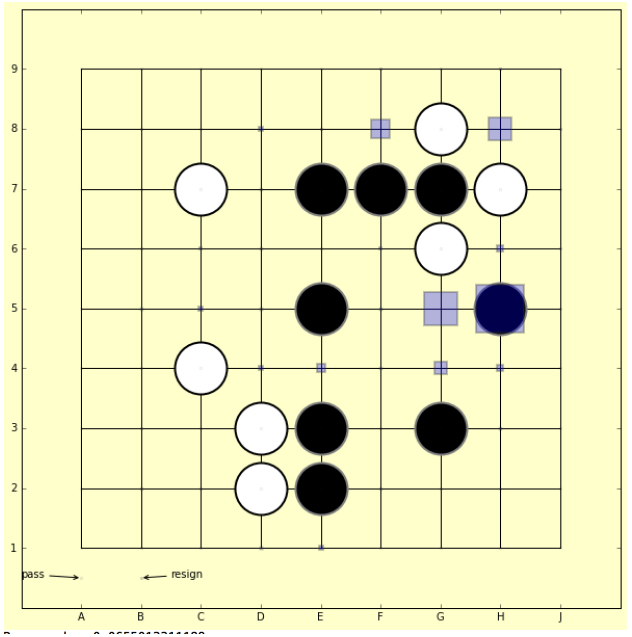
AlphaGo's accuracy:
In the 50-60% range

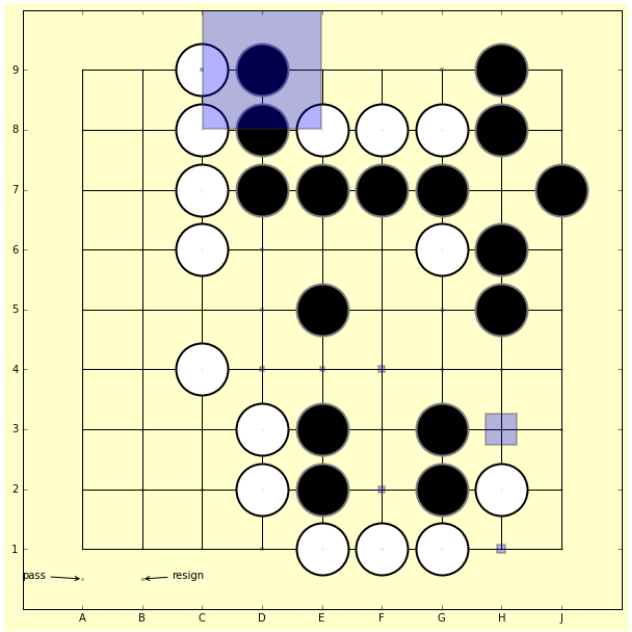
Training Results 500,000 runs

Training Results 16 million runs



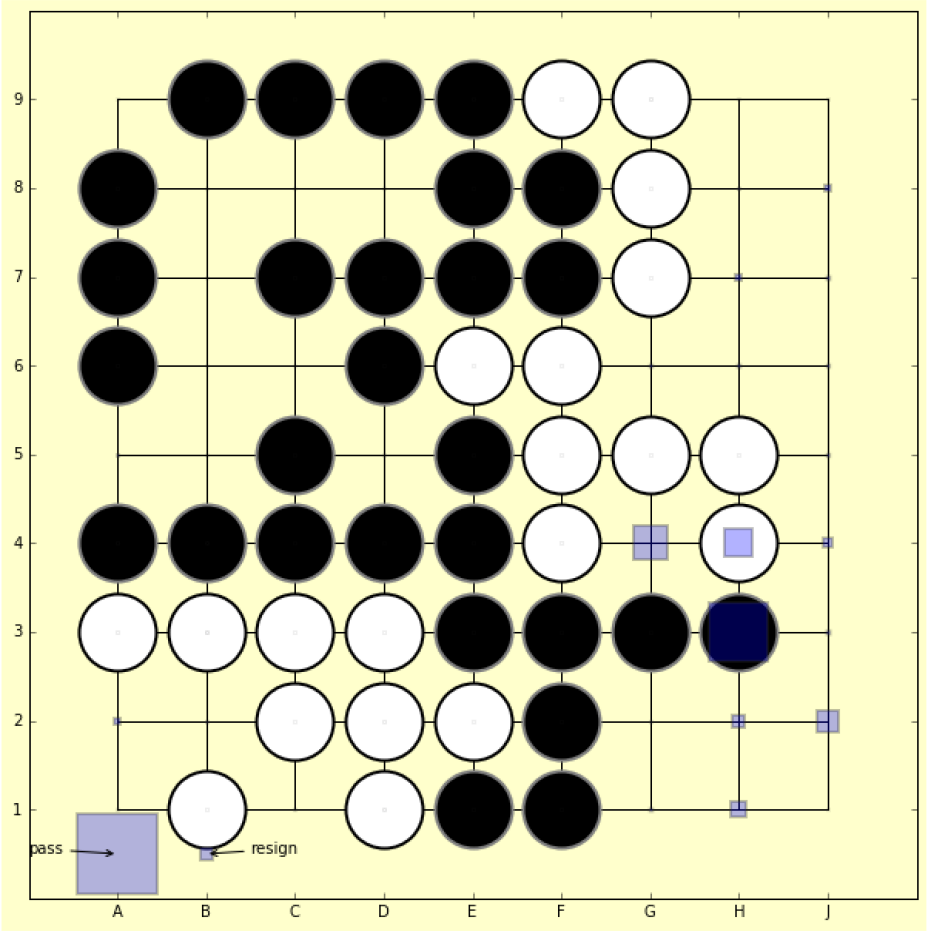
Prediction Array = Heat Map






```
for index in range(len(predictionArray)):

        x = index % 9
        y = index / 9

        heatEdgeColor = (.2,.2, .2)
        heatFaceColor = 'b'
        prediction = 200 * predictionArray[index]

        # draw heat map
        s2, = ax.plot(x,y,'s',markersize=prediction,
                      markeredgecolor=heatEdgeColor,
                      markerfacecolor=heatFaceColor,
                      markeredgewidth=2, alpha=0.3)
```

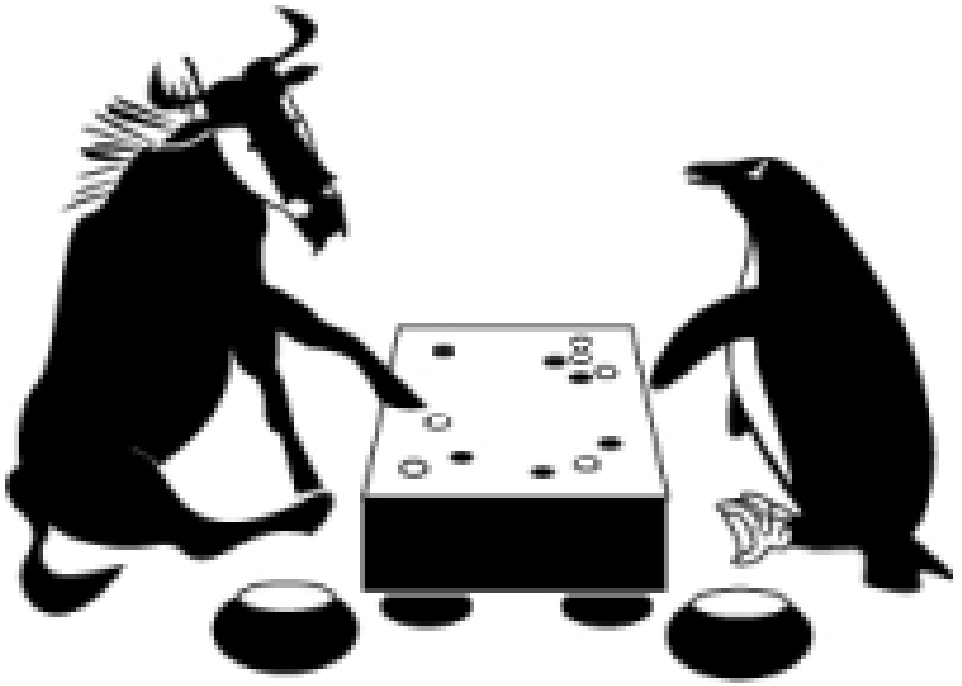
Handling Pass and Resign




```
if c.atEnd:
    # Add in the last board position
    # - the final pass or resign move.
    # The robot needs to train
    #when to pass and when to resign

    if (gameEndedByTime == True):
        break
        #No passing move needed

    label = np.zeros(83)

    if (gameEndedInResignation == True):
        # Board has just been flipped
        label[82]=1
    else:
        label[81]=1
```

Playing against GnuGo

A free software program by the Free Software Foundation that plays Go using the Go Text Protocol (GTP)



```
1 boardsize 9
=1
2 play black D5
=2
3 genmove white
=3 C3
4 play black C3
?4 illegal move
```

Demo of OmegaGo Versus GnuGo

GnuGo versus OmegaGo Stats

Even Game

(OmegaGo plays black.

Komi is 6.5)

48% win

One stone handicap

(OmegaGo plays black.

Komi is 0.5)

61% win

Even Game

(OmegaGo plays white.

Komi is 6.5)

53% win

Two stone handicap

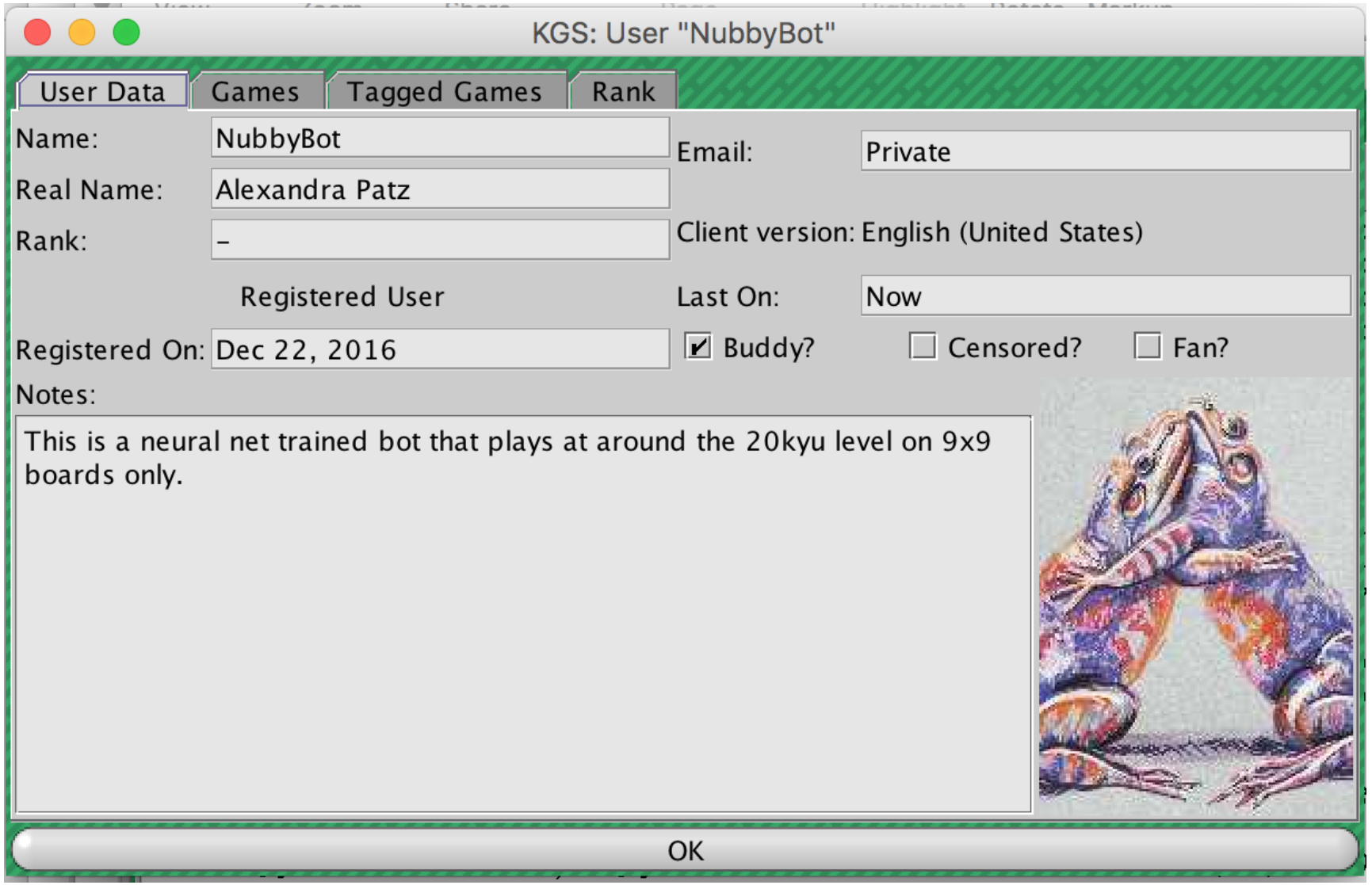
(OmegaGo plays black.

Komi is 0.5)

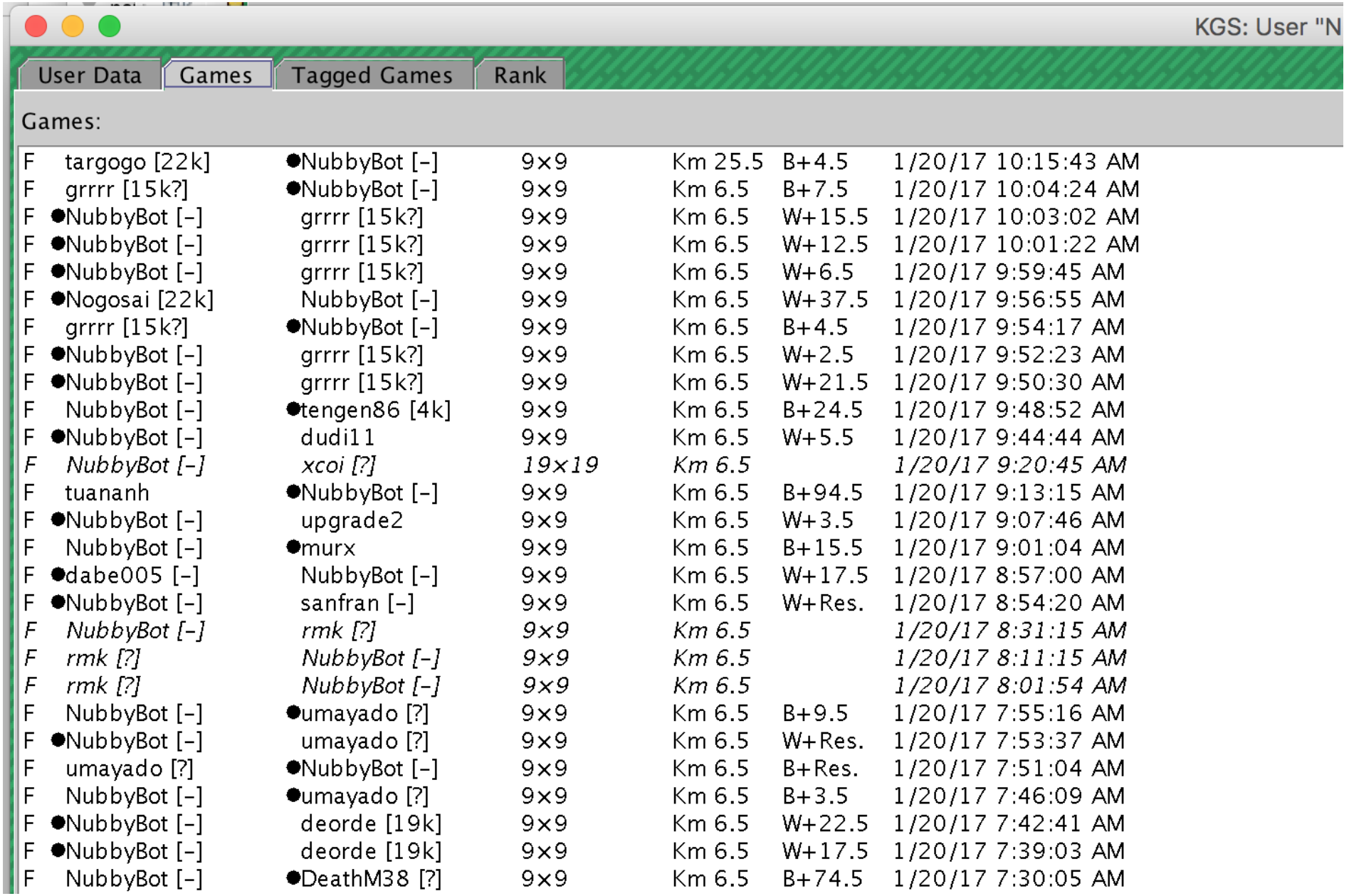
85% win

Playing on KGS (NubbyBot)

Use kgsGtp.jar to connect



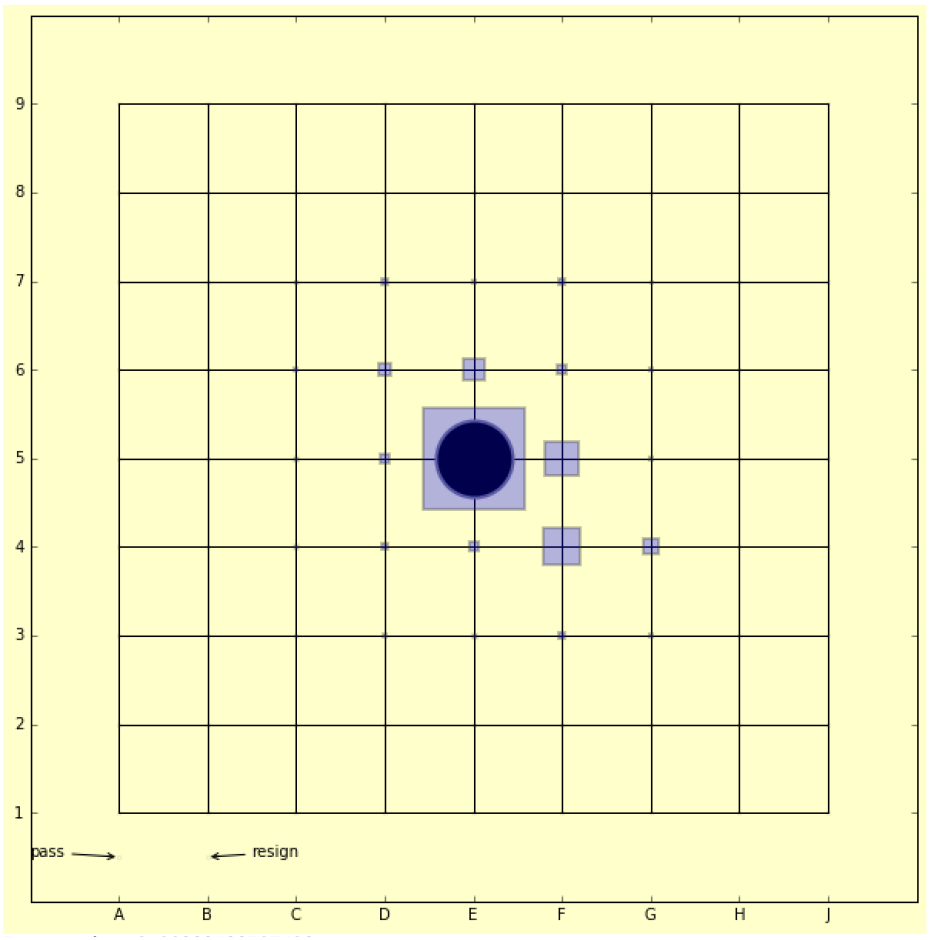
KGS Results



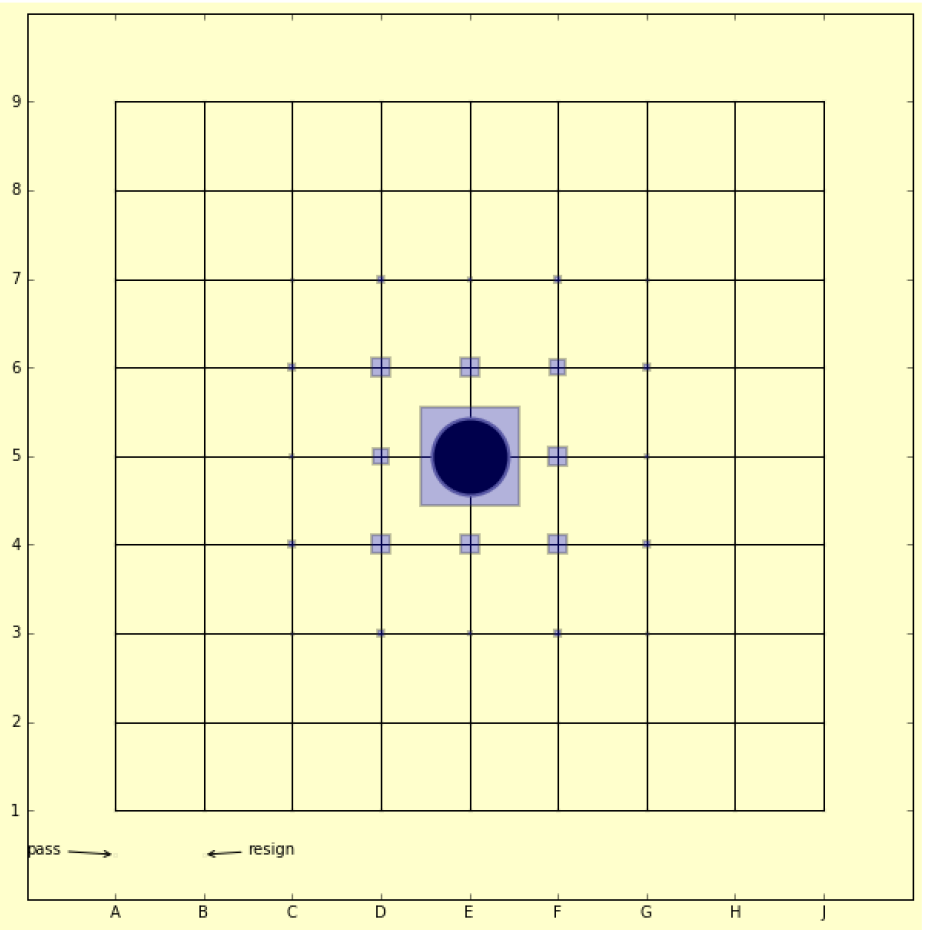
F	targogo [22k]	●NubbyBot [-]	9×9	Km 25.5	B+4.5	1/20/17 10:15:43 AM
F	grrrr [15k?]	●NubbyBot [-]	9×9	Km 6.5	B+7.5	1/20/17 10:04:24 AM
F	●NubbyBot [-]	grrrr [15k?]	9×9	Km 6.5	W+15.5	1/20/17 10:03:02 AM
F	●NubbyBot [-]	grrrr [15k?]	9×9	Km 6.5	W+12.5	1/20/17 10:01:22 AM
F	●NubbyBot [-]	grrrr [15k?]	9×9	Km 6.5	W+6.5	1/20/17 9:59:45 AM
F	●Nogosai [22k]	NubbyBot [-]	9×9	Km 6.5	W+37.5	1/20/17 9:56:55 AM
F	grrrr [15k?]	●NubbyBot [-]	9×9	Km 6.5	B+4.5	1/20/17 9:54:17 AM
F	●NubbyBot [-]	grrrr [15k?]	9×9	Km 6.5	W+2.5	1/20/17 9:52:23 AM
F	●NubbyBot [-]	grrrr [15k?]	9×9	Km 6.5	W+21.5	1/20/17 9:50:30 AM
F	NubbyBot [-]	●tengen86 [4k]	9×9	Km 6.5	B+24.5	1/20/17 9:48:52 AM
F	●NubbyBot [-]	dudi11	9×9	Km 6.5	W+5.5	1/20/17 9:44:44 AM
F	*NubbyBot [-]*	*xcoi [?]*	*19×19*	*Km 6.5*		*1/20/17 9:20:45 AM*
F	tuananh	●NubbyBot [-]	9×9	Km 6.5	B+94.5	1/20/17 9:13:15 AM
F	●NubbyBot [-]	upgrade2	9×9	Km 6.5	W+3.5	1/20/17 9:07:46 AM
F	NubbyBot [-]	●murx	9×9	Km 6.5	B+15.5	1/20/17 9:01:04 AM
F	●dabe005 [-]	NubbyBot [-]	9×9	Km 6.5	W+17.5	1/20/17 8:57:00 AM
F	●NubbyBot [-]	sanfran [-]	9×9	Km 6.5	W+Res.	1/20/17 8:54:20 AM
F	*NubbyBot [-]*	*rmk [?]*	*9×9*	*Km 6.5*		*1/20/17 8:31:15 AM*
F	*rmk [?]*	*NubbyBot [-]*	*9×9*	*Km 6.5*		*1/20/17 8:11:15 AM*
F	*rmk [?]*	*NubbyBot [-]*	*9×9*	*Km 6.5*		*1/20/17 8:01:54 AM*
F	NubbyBot [-]	●umayado [?]	9×9	Km 6.5	B+9.5	1/20/17 7:55:16 AM
F	●NubbyBot [-]	umayado [?]	9×9	Km 6.5	W+Res.	1/20/17 7:53:37 AM
F	umayado [?]	●NubbyBot [-]	9×9	Km 6.5	B+Res.	1/20/17 7:51:04 AM
F	NubbyBot [-]	●umayado [?]	9×9	Km 6.5	B+3.5	1/20/17 7:46:09 AM
F	●NubbyBot [-]	deorde [19k]	9×9	Km 6.5	W+22.5	1/20/17 7:42:41 AM
F	●NubbyBot [-]	deorde [19k]	9×9	Km 6.5	W+17.5	1/20/17 7:39:03 AM
F	NubbyBot [-]	●DeathM38 [?]	9×9	Km 6.5	B+74.5	1/20/17 7:30:05 AM

Human Bias

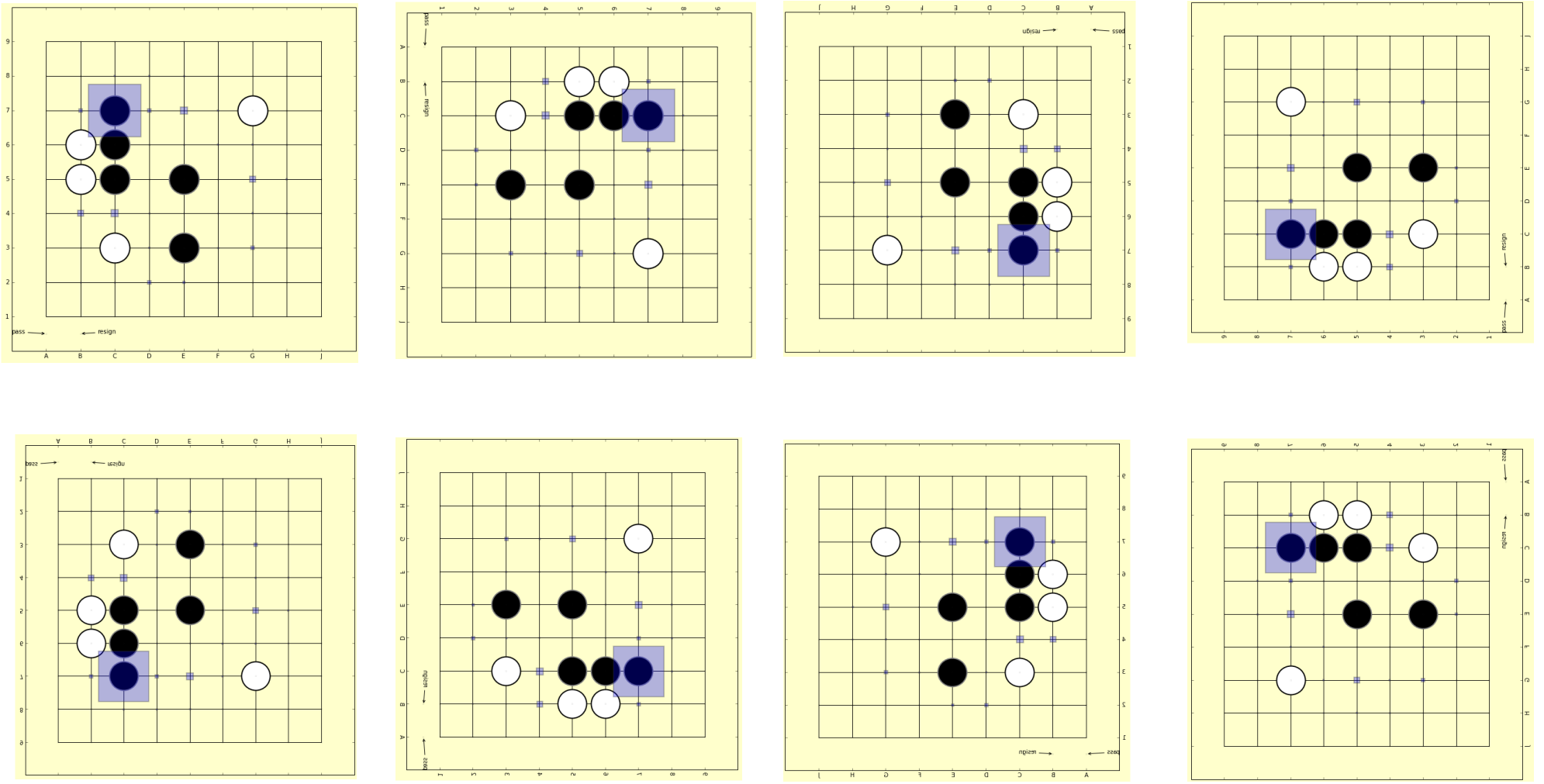
No data transformation: Note human bias



Eight transformations performed



Transforming data to increase data size and overcome human bias



Using numpy.fliplr(), numpy.flipud() and numpy.rot90()

Neural Network Design

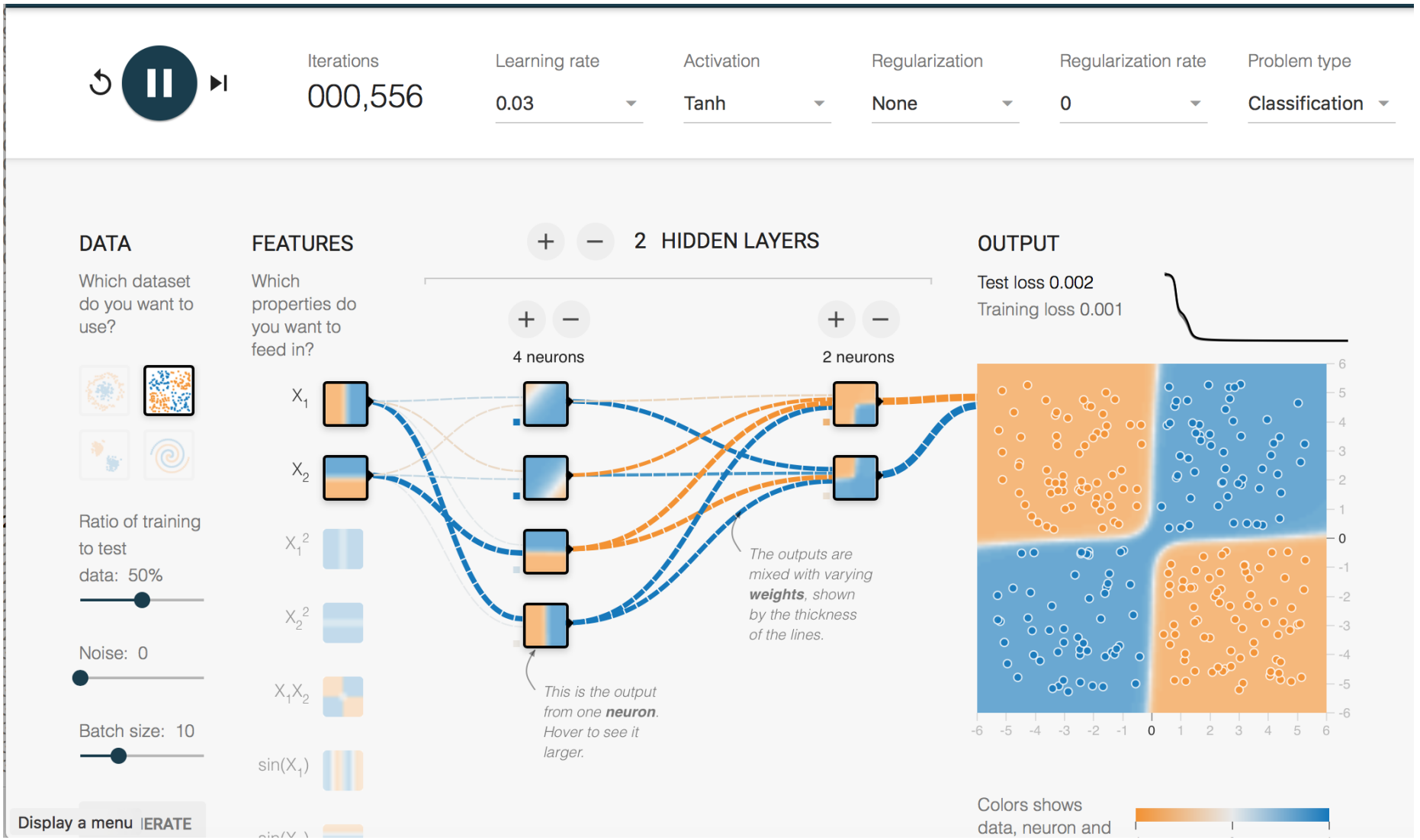
Dataset: 11 million features from strong games
Optimizer: GradientDescentOptimizer
Learning rate: 0.05
Layers: 2 (one hidden layer)
Activation function: Relu
Hidden layer number of nodes: 1024
Batch size: 1000
Runs: 20 million
Shuffling after each full pass through the dataset

Hyper parameter variations tried:

- Three layer network (2 hidden layers)
- More hidden nodes / less hidden nodes
- Different optimizer algorithm (AdagradOptimizer)
- Different activation function (tanh instead of relu)
- Decaying learning rate

TensorFlow Playground

playground.tensorflow.org



Other variations tried

- Added mixed level games to the dataset

  Black: >16 kyu

  White: < 6 kyu

  Improved win rates against gnuGo by 5%!

- Tried adding another feature plane

Adding another feature plane

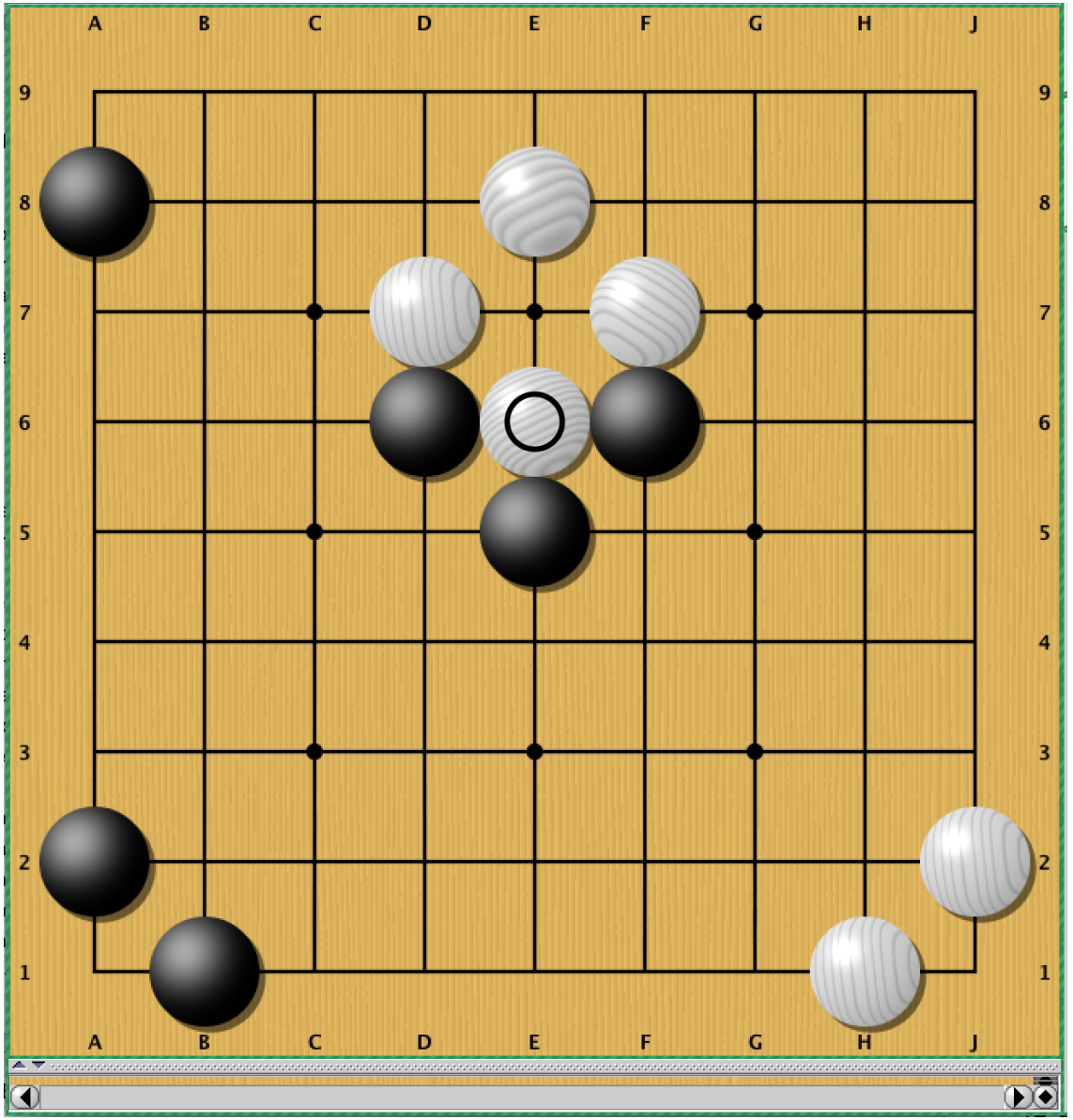
RESEARCH ARTICLE

Extended Data Table 2 | Input features for neural networks

Feature	# of planes	Description
Stone colour	3	Player stone / opponent stone / empty
Ones	1	A constant plane filled with 1
Turns since	8	How many turns since a move was played
Liberties	8	Number of liberties (empty adjacent points)
Capture size	8	How many opponent stones would be captured
Self-atari size	8	How many of own stones would be captured
Liberties after move	8	Number of liberties after this move is played
Ladder capture	1	Whether a move at this point is a successful ladder capture
Ladder escape	1	Whether a move at this point is a successful ladder escape
Sensibleness	1	Whether a move is legal and does not fill its own eyes
Zeros	1	A constant plane filled with 0
Player color	1	Whether current player is black

Feature planes used by the policy network (all but last feature) and value network (all features).

More Feature Planes: Legal and Illegal Moves



One-hot-encoded feature planes

```
[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. 1. 0. 1. 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.
 1. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.
```

Black

```
0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 1. 0. 0. 0. 0.
0. 0. 0. 1. 0. 1. 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. 1.
0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0.
```

White

```
1. 1. 1. 1. 1. 1. 1. 1. 1.
0. 1. 1. 1. 0. 1. 1. 1. 1.
1. 1. 1. 0. 0. 0. 1. 1. 1.
1. 1. 1. 0. 0. 0. 1. 1. 1.
1. 1. 1. 1. 0. 1. 1. 1. 1.
1. 1. 1. 1. 1. 1. 1. 1. 1.
1. 1. 1. 1. 1. 1. 1. 1. 1.
0. 1. 1. 1. 1. 1. 1. 1. 0.
1. 0. 1. 1. 1. 1. 1. 0. 0. 0. 0.
```

Empty Legal

```
0. 0. 0. 0. 0. 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. 1. 0. 0.]
```

Empty Illegal

Code on GitHub:

https://github.com/nubbyp/omega_go