COMPUTATIONAL INTELLIGENCE
- labs & project

Patryk Gałczyński

Labs

Iris, MLP, SOM, AGDS..

Summary no.1

Personal neural net:

- Written in Python3
- Follows OO concept

- Loads data from csv
- Exports net to yaml and cypher

- Pluginable bias nodes
- Any activation function

The right way to do that

```yaml
!!python/object:core.network.Network
constants:
  activation: sigmoid
  beta: 0.7
  learning_factor: 0.3
  normalised: true
  weight_from: -2
  weight_to: 2
layers:
- !!python/object:core.layer.InputLayer
  network_name: xor
  neurons:
  - &id005 !!python/object:core.neuron.InputNeuron
    delta: -0.06134907502497106
    deltas: !!set {}
    inputs: null
    learning_factor: 0.3
    name: input1
    network_name: xor
    outputs:
    - &id001 !!python/object:core.neuron.Connection
      another: &id002 !!python/object:core.neuron.Neuron
        delta: 0.19809358937246652
        deltas: !!set {}
        inputs:
        - *id001
        - &id003 !!python/object:core.neuron.Connection
          another: *id002
          one: &id014 !!python/object:core.neuron.InputNeuron
```

Protip #0

Use YAML, its awesome

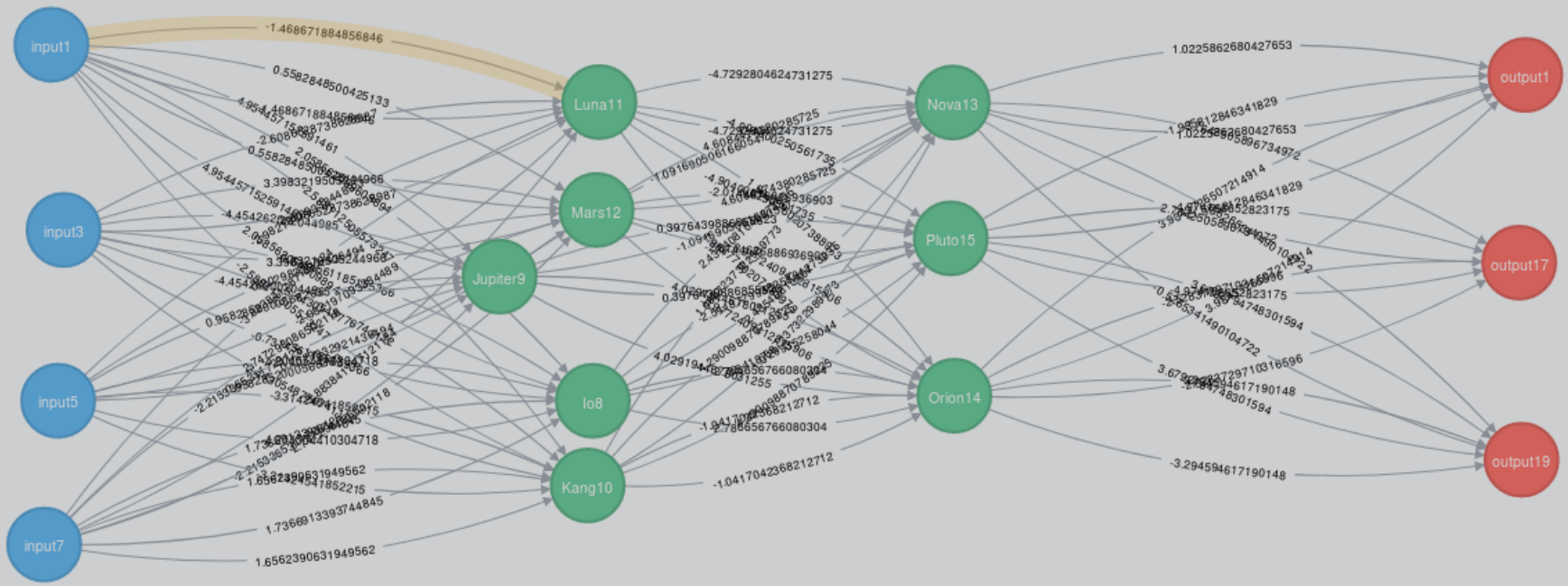
1

Basic iris

We all love iris dataset

Iris dataset

1000 iterations in 4 folds



Iris dataset

1000 iterations in 4 folds

folds results: [99.6, 99.96842105263158, 94.68947368421053, 91.9361111111111]

In0	In1	In2	In3	Iris-virginica	Iris-versicolor	Iris-setosa
6.900	3.100	5.100	2.300	1.000	0.001	0.000
5	2	3.500	1	0.000	0.966	0.007
5.400	3.700	1.500	0.200	0.000	0.008	0.994

Took obout 40s

Learning rate: 0.3

Beta: 0.7

Random weights: from -2 to 2

Activation function: sigmoid

Iris dataset

1000 iterations in 4 folds

folds results: [99.6, 99.96842105263158, 94.68947368421053, 91.9361111111111]

In0	In1	In2	In3	Iris-virginica	Iris-versicolor	Iris-setosa
6.900	3.100	5.100	2.300	1.000	0.001	0.000
5	2	3.500	1	0.000	0.966	0.007
5.400	3.700	1.500	0.200	0.000	0.008	0.994

Took obout 40s

Learning rate: 0.3

Beta: 0.7

Random weights: from -2 to 2

Activation function: sigmoid

For Tensor Flow it was 0.3s..

Notice #1

Python is slow.

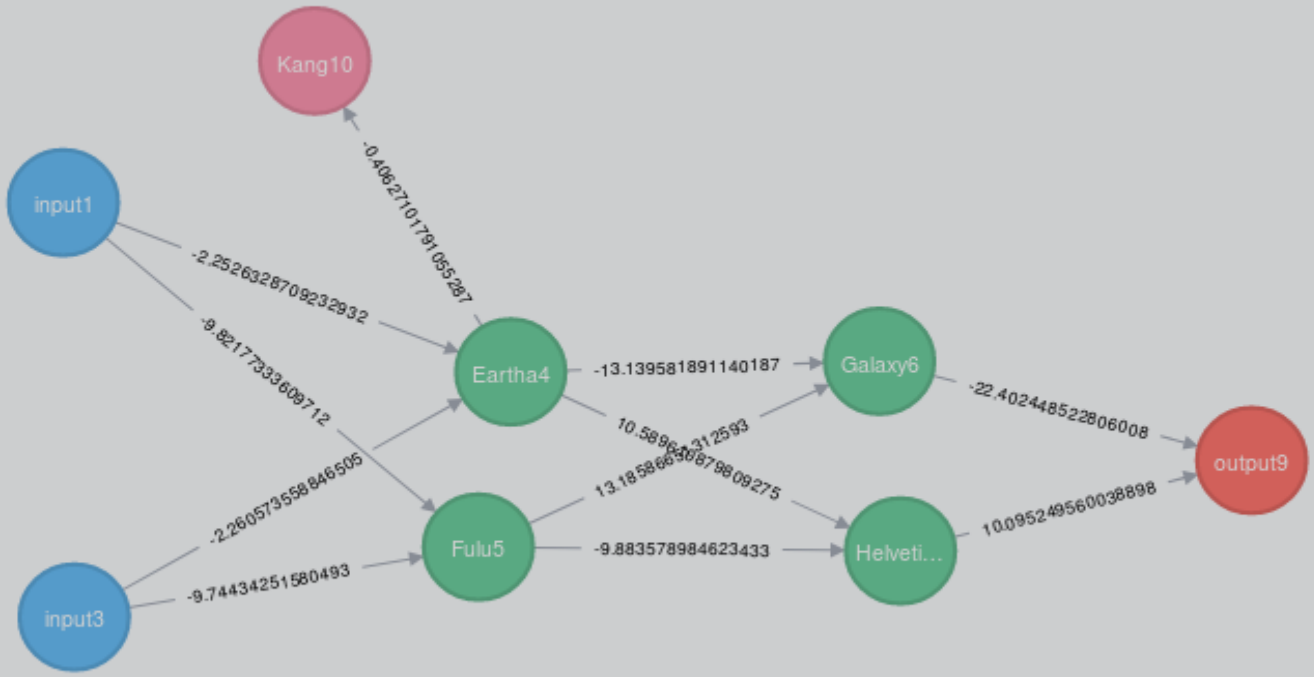
or I can’t write fast code

1

xor

xor?

Another dataset



Another dataset

```
------- test -------
+-----+-----+-------+
| In0 | In1 | Out0  |
+-----+-----+-------+
| 0   | 0   | 0.022 |
+-----+-----+-------+
| 0   | 1   | 0.934 |
+-----+-----+-------+
| 1   | 1   | 0.081 |
+-----+-----+-------+
| 1   | 0   | 0.937 |
+-----+-----+-------+
```

Learning rate: 0.1

Beta: 0.98

Weights: from -5 do 5

10k iterations

Took about 2s

Random bias added

Activation: ReLu

Pro tip #2

Try different activation function!

1

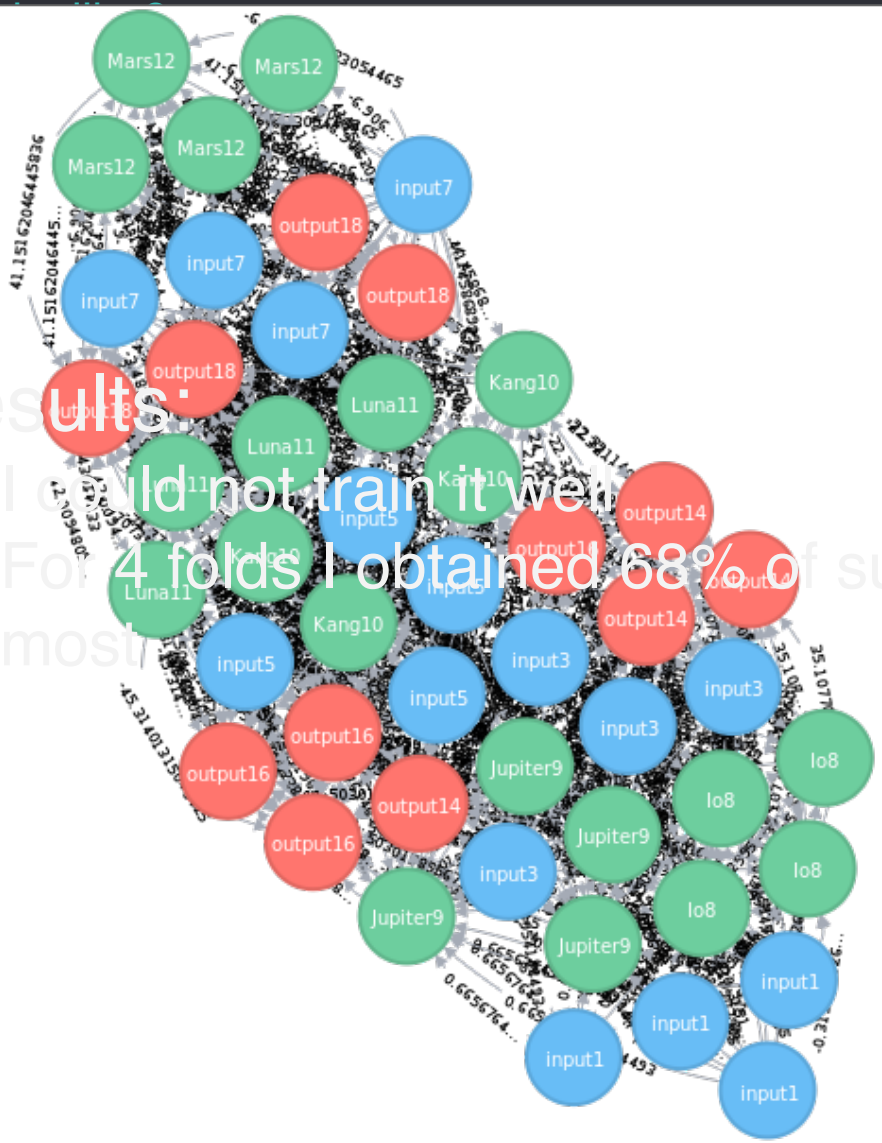
Deep MLP

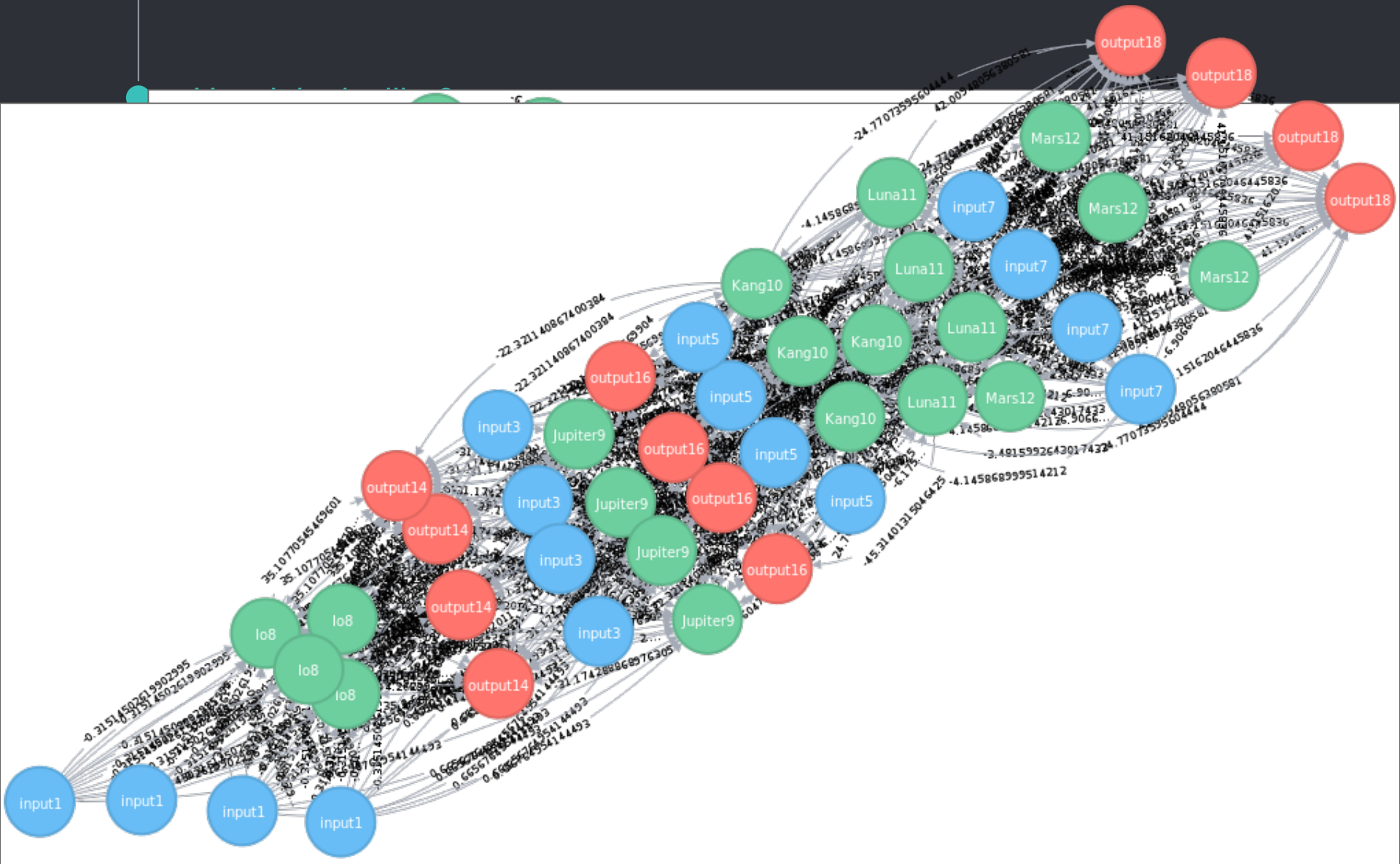
Multi layer perceptron

How it looks like?

- Results:
  - I could not train it well
  - For 4 folds I obtained 68% of success at most

- Results:
- I could not train it well
- For 4 folds I obtained 68% of success at most




output18
output18
output18
output18
Mars12
Mars12
Mars12
Mars12
Luna11
Luna11
Luna11
Luna11
input7
input7
input7
input7
Kang10
Kang10
Kang10
Kang10
input5
input5
input5
input5
output16
output16
output16
output16
input3
input3
input3
input3
Jupiter9
Jupiter9
Jupiter9
Jupiter9
output14
output14
output14
output14
Io8
Io8
Io8
Io8
input1
input1
input1
input1

2

Som

Self organizing maps

Som

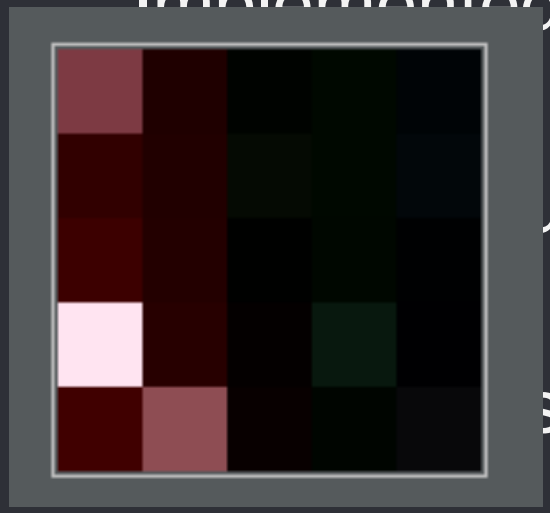
Implemented with 5x5 matrix (numpy),

Trained unsupervised

Plugged as second input to mlp

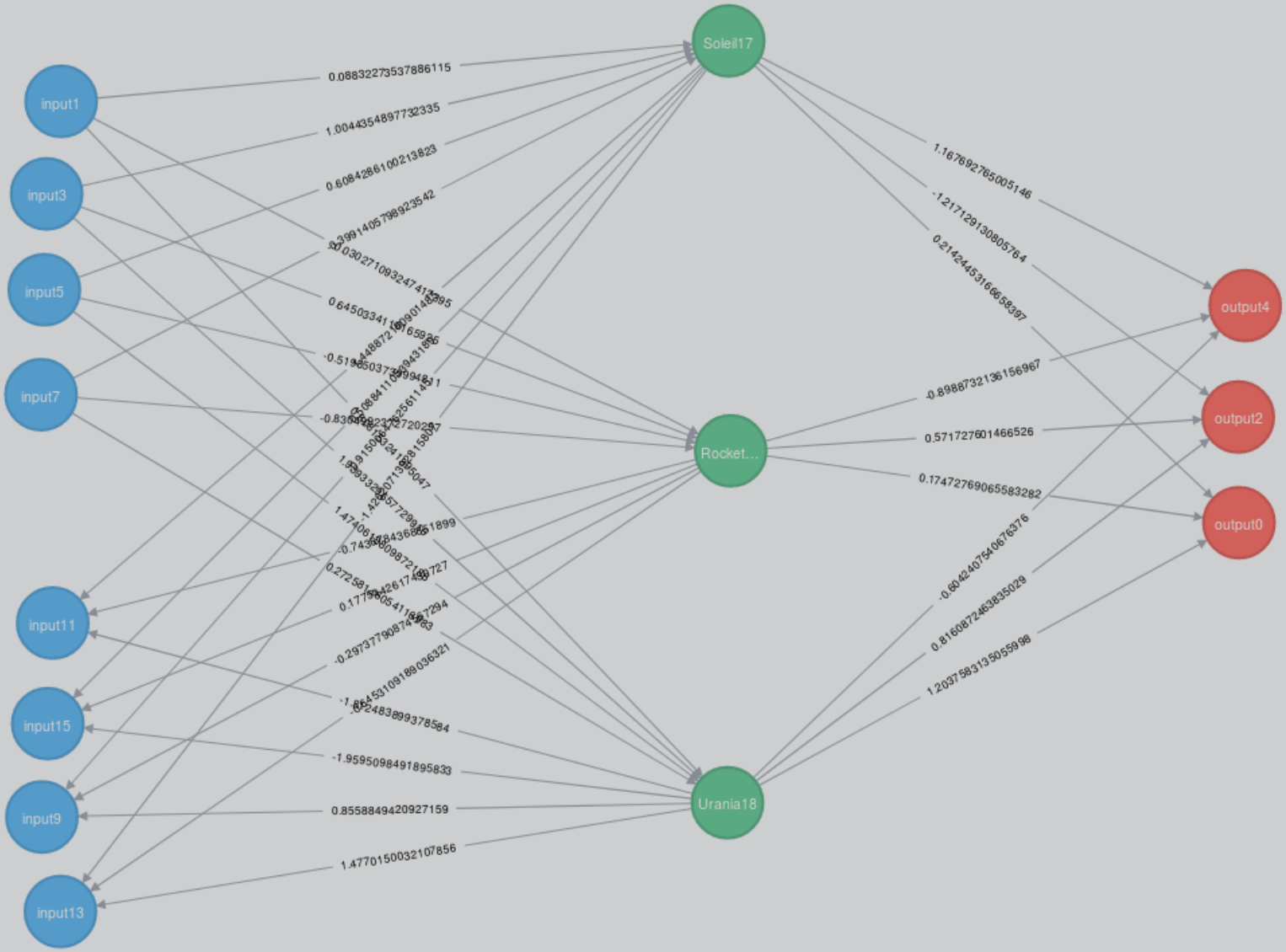
Som

Implemented with 5x5 matrix (numpy)



upervised

second inp


input1
input3
input5
input7
input11
input15
input9
input13
Soleil17
Rocket...
Urania18
output4
output2
output0
0.08832273537886115
1.0044354897732335
0.608428610021382 3
0.6450334110901485395
-0.8359372720297
1.4740614843668 6
-0.2972779087 4263
-1.9595098491895833
0.8558849420927159
1.4770150032107856
1.1676927650051 46
-1.2171291308057 84
0.2142445316658397
-0.8988732136156967
0.571727601466526
0.1747276906558 3282
-0.6042407540676376
0.8160872463835029
1.2037583135055998

Gains:

- 95.75 % without SOM

- 96.50 % with SOM

3 AGDS

Similarity percentage:

[((id: 140403750792328, class: Iris-versicolor, 4.0-1.2-5.8-2.6), 100.0)
((id: 140403750816568, class: Iris-virginica, 5.6-1.4-6.1-2.6), 81.24),
((id: 140403750808208, class: Iris-virginica, 5.0-1.5-6.0-2.2), 79.67),
((id: 140403750818872, class: Iris-virginica, 5.1-1.9-5.8-2.7), 79.6),
((id: 140403750799008, class: Iris-virginica, 5.1-1.9-5.8-2.7), 79.6),
((id: 140403750816288, class: Iris-virginica, 5.1-1.5-6.3-2.8), 79.33),
((id: 140403750789528, class: Iris-versicolor, 3.9-1.2-5.8-2.7), 78.83),
((id: 140403750806528, class: Iris-virginica, 5.0-2.0-5.7-2.5), 78.55),
((id: 140403750814328, class: Iris-virginica, 4.8-1.8-6.2-2.8), 78.4),
((id: 140403750809328, class: Iris-virginica, 4.9-1.8-6.3-2.7), 78.34),
((id: 140403750800408, class: Iris-virginica, 4.5-1.7-4.9-2.5), 78.31),
((id: 140403750817688, class: Iris-virginica, 4.8-1.8-6.0-3.0), 77.84),
((id: 140403750808768, class: Iris-virginica, 4.9-2.0-5.6-2.8), 77.5),
((id: 140403750820832, class: Iris-virginica, 5.1-1.8-5.9-3.0), 77.38),
((id: 140403750819992, class: Iris-virginica, 5.0-1.9-6.3-2.5), 77.17),
((id: 140403750785264, class: Iris-versicolor, 4.1-1.0-5.8-2.7), 77.16),
((id: 140403750791768, class: Iris-versicolor, 4.4-1.2-5.5-2.6), 76.98),
((id: 140403750814608, class: Iris-virginica, 4.9-1.8-6.1-3.0), 76.95),
((id: 140403750785824, class: Iris-versicolor, 3.9-1.1-5.6-2.5), 76.88),
...

Notice #3

Precalculated AGDS is blazing-fast

Final project

Artificial player (neural net) for Snake

1

Design Features

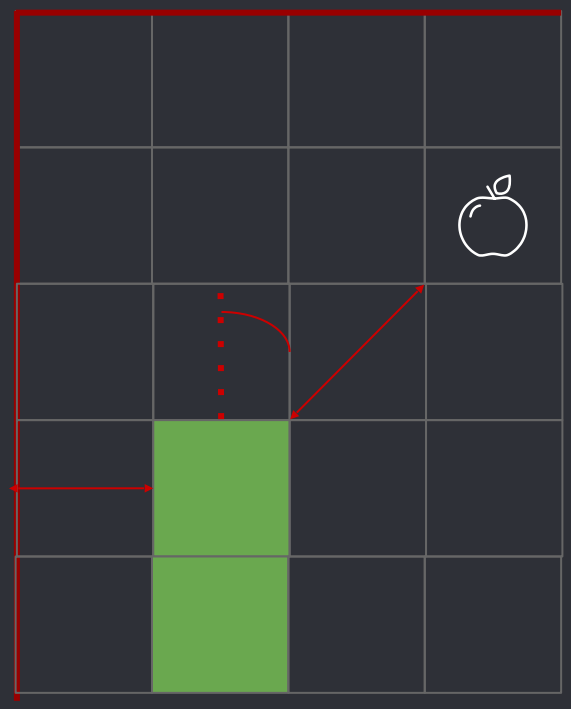
Which parameters could be useful?

Features

Obstacles?
- In front
- Left side
- Right side

Distance to apple?

Angle to apple?



2

Gather data

Lots of it

How to gather data?

Generate your own

Since snake is fairly simple, we can generate random moves and evaluate them

Get some friends

Track moves from real games

Learning data gathering

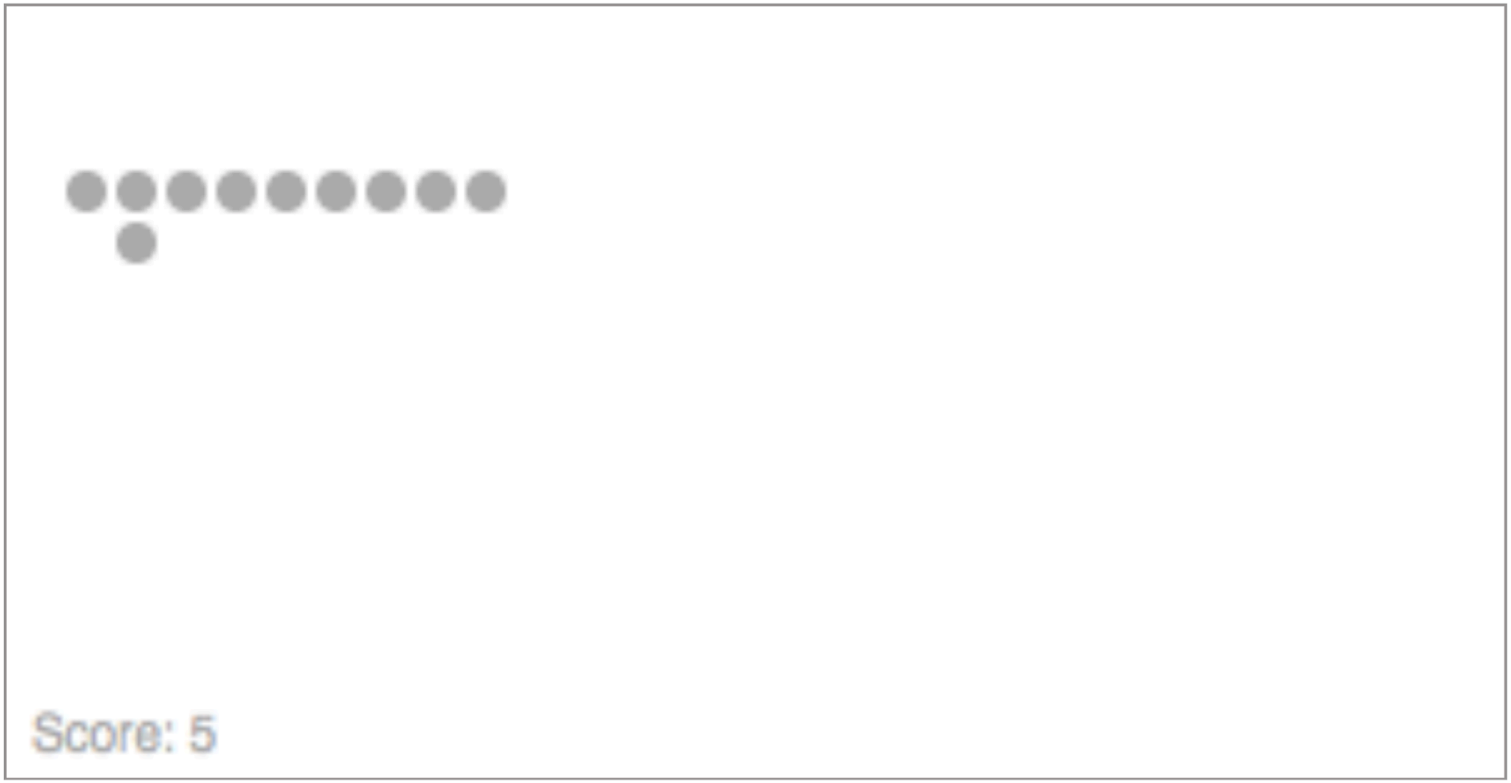
https://storage.googleapis.com/snek-front/main.html

Safari File Edit View History Bookmarks Develop Window Help 74% Tue 00:52

storage.googleapis.com

Olivia · Slid | Eleanor · Slid.. | Adrian Horzy... | Adrian Horzy... | home.agh.ed... | home.agh.ed... | Google Keep | Neural Netw... | Datastore Q... | Bucket detail... | https://stora...

Snek Snek

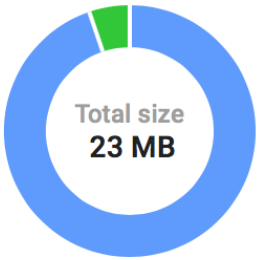


Kind

senk

Summary

Last updated 15 hours ago — June 18, 2018 at 9:43:41 AM UTC+2



Resource	Count	Size
Entities	118	21.82 MB
Built-in indexes	14,808	1.2 MB
Composite indexes	0	0 B

Kind "senk" is 100% of total Datastore storage
23.02 MB / 23.02 MB

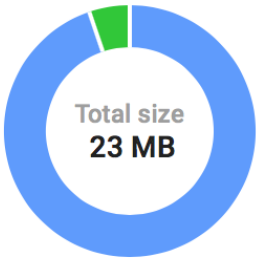
Score: 5

Kind

senk

Summary

Last updated 15 hours ago — June 18, 2018 at 9:43:41 AM UTC+2



Resource	
Entities	
Built-in indexes	
Composite indexes	

Kind "senk" is 100% of total Datastore storage
23.02 MB / 23.02 MB

Score: 5

Entity details

Key: senk id:4795020674596864 Key literal: Key(senk, 4795020674596864) URL-safe key: ahpnfm15LXByb2plY3QtMTQ3NzczMDQ2MDYwMHIRCxIEc2Vuax

Properties

date
1525194393519

game
[{"score":"0","new_dir":"down","snek":[{"y":"0","x":"4"},{"y":"1","x":"4"},{"y":"2","x":"4"},{"y":"2","x":"5"},{"y":"2","x":"6"},{"y":"2","x":"7"},{"y":"2","x":"8"},{"y":"2","x":"9"},{"y":"2","x":"10"},{"y":"2","x":"11"}],"old_dir":"right","food":{"x":"13","y":"5"}},{"new_dir":"right","snek":[{"x":"4","y":"0"},{"x":"4","y":"1"},{"y":"2","x":"4"},{"y":"2","x":"5"},{"x":"6","y":"2"},{"y":"2","x":"7"},{"y":"2","x":"8"},{"y":"2","x":"9"},{"y":"2","x":"10"},{"y":"2","x":"11"}],"old_dir":"down","food" ...

Pro tip #4

Gather more, filter later

2

Learning time

This thing is so lazy..

Baby steps

```
Score : 0                    SNAKE

        #
        ##          *

{'r': 0, 'f': 0, 'a': -1, 'l': 1}
```

```
Score : 0                    SNAKE
                    ###

                    *

                    1

{'l': 0, 'f': 0, 'a': 0, 'r': 1}
```

Baby steps

```
Score : 0                    SNAKE

        #
        ##          *

{'r': 0, 'f': 0, 'a': -1, 'l': 1}
```

```
Score : 0                    SNAKE
                    ###

                    *

                    1

{'l': 0, 'f': 0, 'a': 0, 'r': 1}
```

3

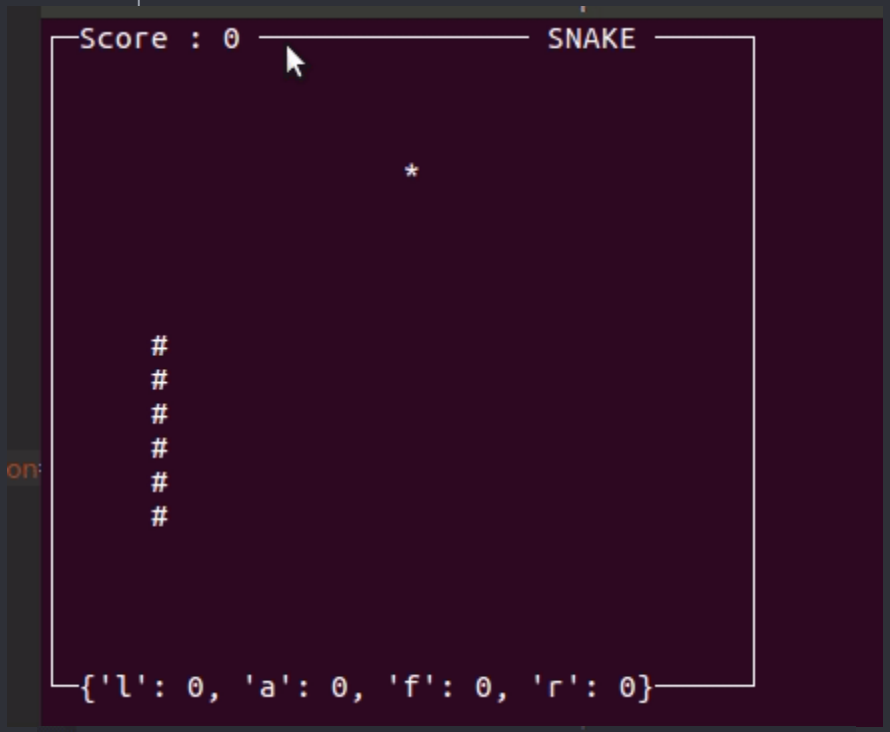
We can do better

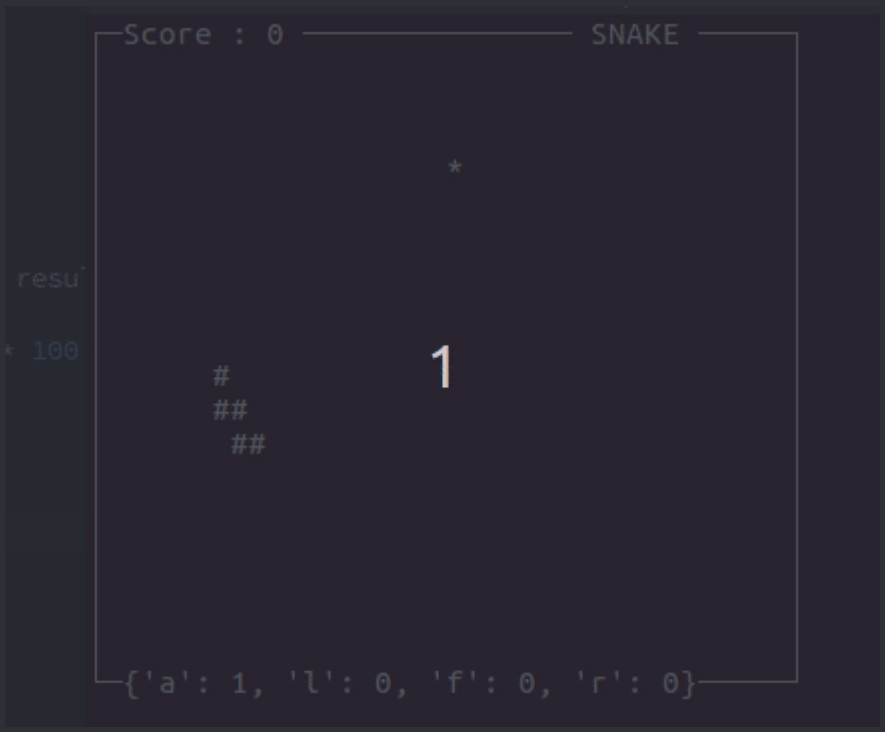
Uh oh.

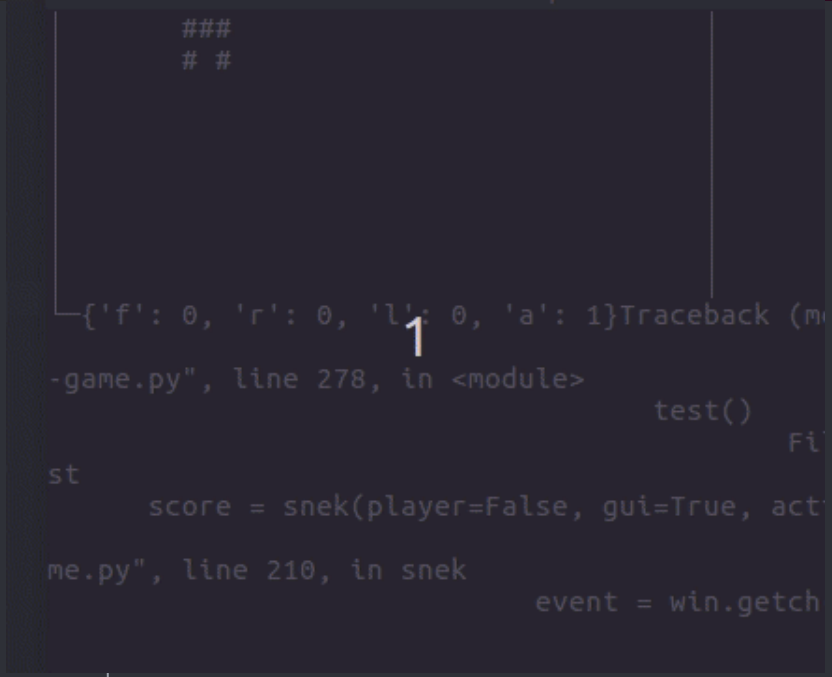
After tinkering for a while

Things done to improve:

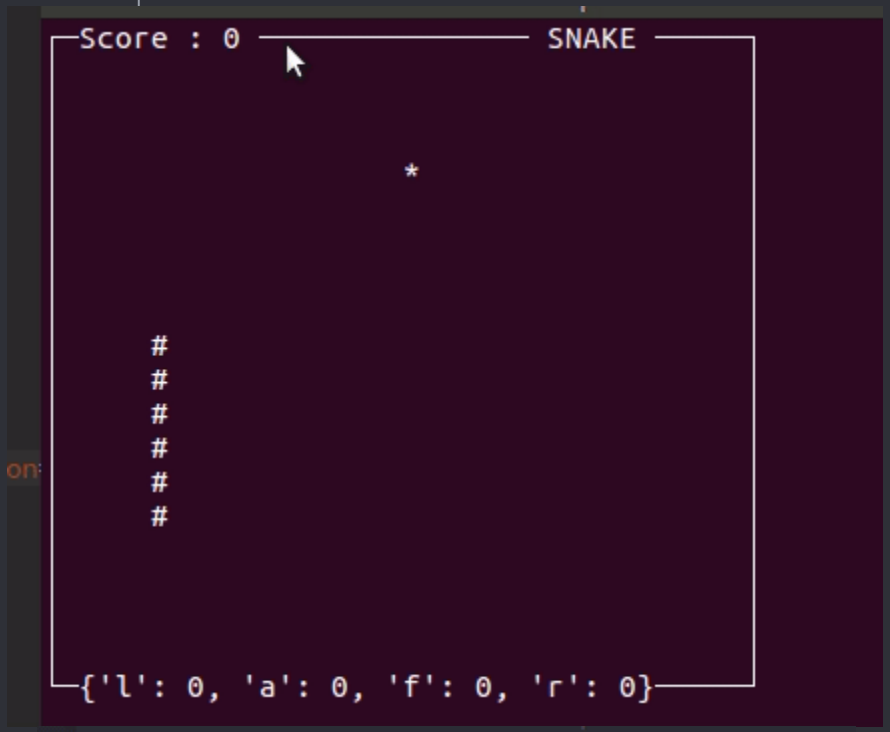
- Change number of layers/neurons/ connections
- Add some bias
- Change activation function to ReLu
- Tinker around parameters (learning rate and so on..)

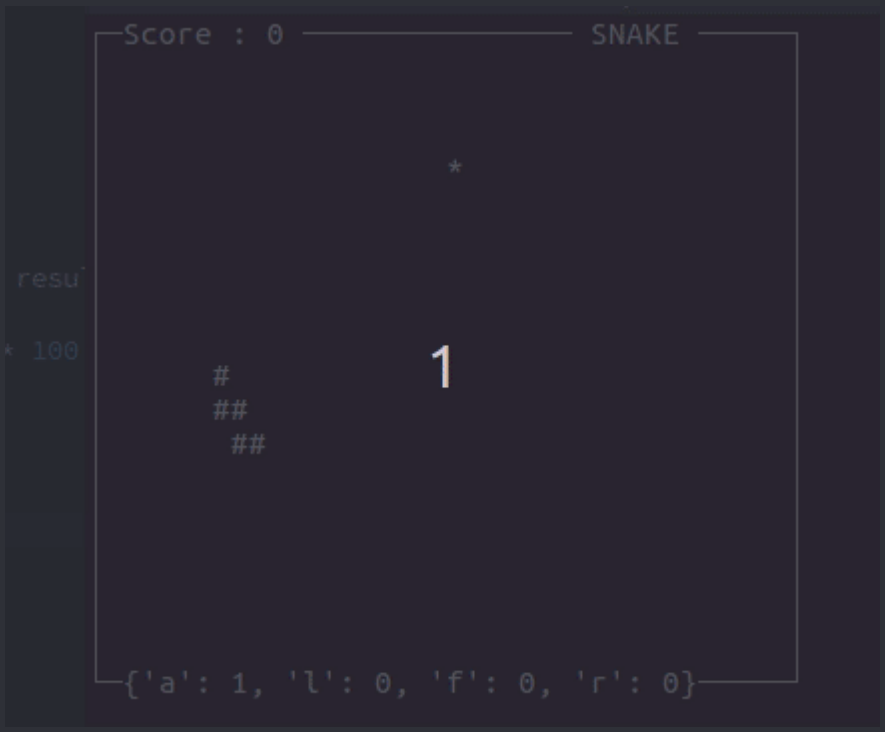

Score : 0
SNAKE
*
#
#
#
#
#
#
{'l': 0, 'a': 0, 'f': 0, 'r': 0}

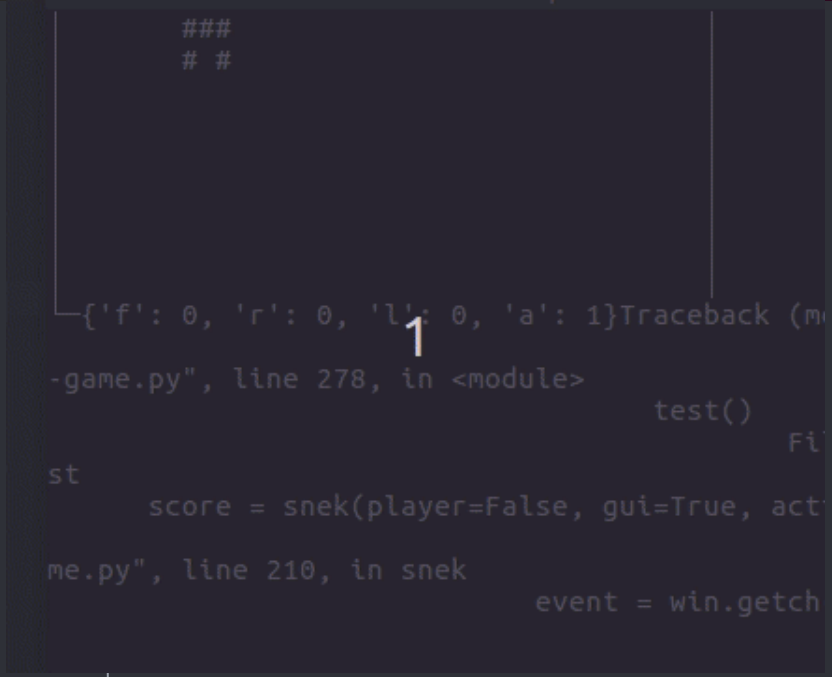

Score : 0
SNAKE
*
1
#
##
##
{'a': 1, 'l': 0, 'f': 0, 'r': 0}


###
# #
1
{'f': 0, 'r': 0, 'l': 0, 'a': 1}Traceback (m
-game.py", line 278, in <module>
test()
Fi
st
score = snek(player=False, gui=True, act
me.py", line 210, in snek
event = win.getch

meh.


Score : 0
SNAKE
*
#
#
#
#
#
#
{'l': 0, 'a': 0, 'f': 0, 'r': 0}


Score : 0
SNAKE
*
1
#
##
##
{'a': 1, 'l': 0, 'f': 0, 'r': 0}


###
# #
1
{'f': 0, 'r': 0, 'l': 0, 'a': 1}Traceback (m
-game.py", line 278, in <module>
test()
Fi
st
score = snek(player=False, gui=True, act
me.py", line 210, in snek
event = win.getch

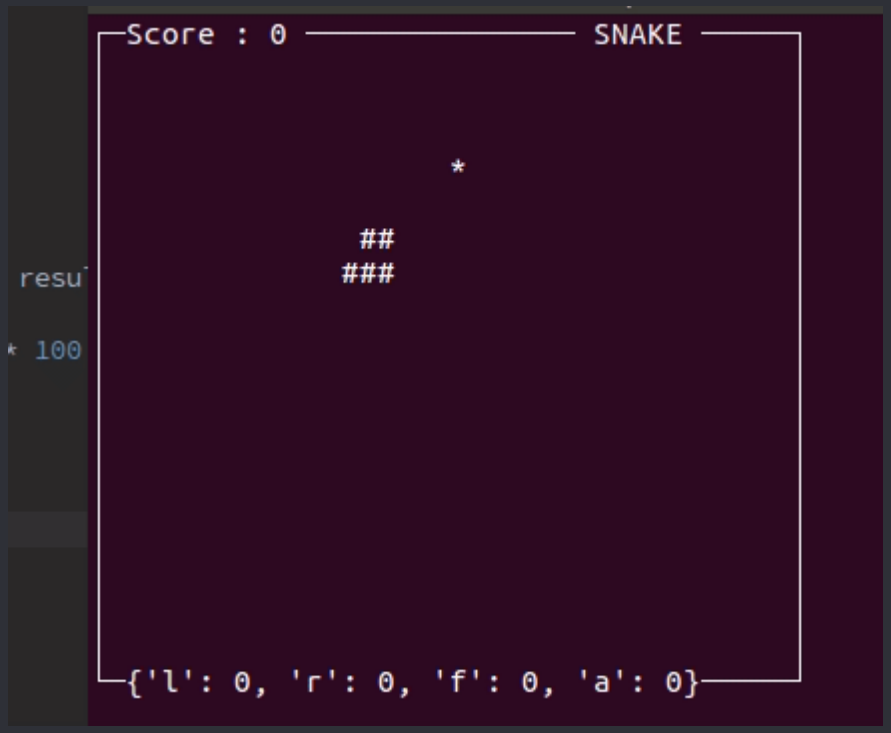
meh.

And then something clicked

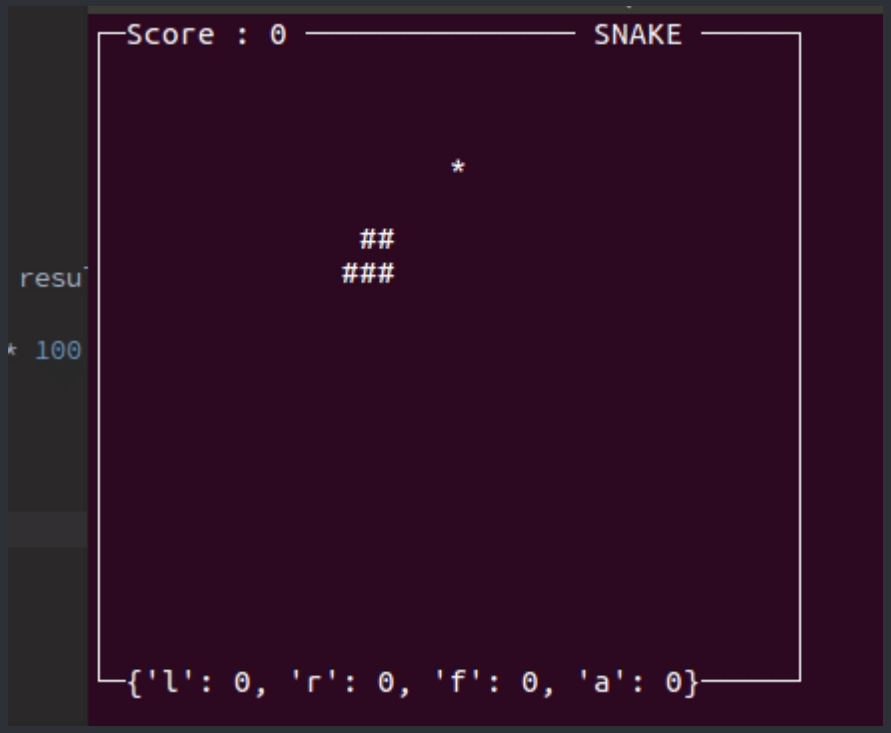
```
if violated_boundary_conditions(snake):
    result = -1
elif not violated_boundary_conditions(snake) and (new_dist < prev_dist) or score > prev_score:
    result = 1
else:
    result = 0
```

```
if violated_boundary_conditions(snake):
    result = -1
elif violated_boundary_conditions(snake) and (new_dist < prev_dist or score > prev_score):
    result = 0.5
elif not violated_boundary_conditions(snake) and (new_dist < prev_dist) or score > prev_score:
    result = 1
else:
    result = -0.5
```

Its first points gathered



Its first points gathered



Pro tip #5

Be patient

Obtained results:

With:

- Over 100 real games
- About 2000 data points extracted
- Over 10k points generated

After:

- 20 minutes of learning

Got:

- ~ 86% from 4 folds

Summary

```
Average steps: 188.566
Counter({112: 11, 75: 10, 76: 10, 82: 10, 97: 1
0: 7, 124: 7, 149: 7, 159: 7, 182: 7, 60: 6, 64
: 5, 96: 5, 99: 5, 100: 5, 103: 5, 119: 5, 135:
5, 44: 4, 50: 4, 52: 4, 53: 4, 54: 4, 55: 4, 58
: 4, 141: 4, 146: 4, 150: 4, 162: 4, 165: 4, 17
3, 56: 3, 71: 3, 80: 3, 90: 3, 91: 3, 94: 3, 95
 3, 143: 3, 153: 3, 156: 3, 158: 3, 163: 3, 167
 3, 285: 3, 299: 3, 305: 3, 318: 3, 325: 3, 327
 2, 104: 2, 107: 2, 118: 2, 122: 2, 125: 2, 152
 2, 194: 2, 195: 2, 197: 2, 198: 2, 202: 2, 206
 2, 251: 2, 253: 2, 261: 2, 266: 2, 268: 2, 277
 2, 329: 2, 333: 2, 335: 2, 343: 2, 352: 2, 355
 2, 443: 2, 450: 2, 502: 2, 526: 2, 532: 2, 587
, 129: 1, 140: 1, 166: 1, 181: 1, 200: 1, 207:
, 248: 1, 250: 1, 252: 1, 254: 1, 256: 1, 257:
, 297: 1, 298: 1, 300: 1, 302: 1, 303: 1, 308:
, 357: 1, 362: 1, 363: 1, 367: 1, 369: 1, 372:
, 417: 1, 418: 1, 419: 1, 425: 1, 428: 1, 429:
, 487: 1, 490: 1, 494: 1, 517: 1, 548: 1, 553:
, 647: 1, 682: 1, 793: 1, 809: 1, 917: 1})
Average score: 9.386
```

1 network

5 layers

18 neurons

Summary

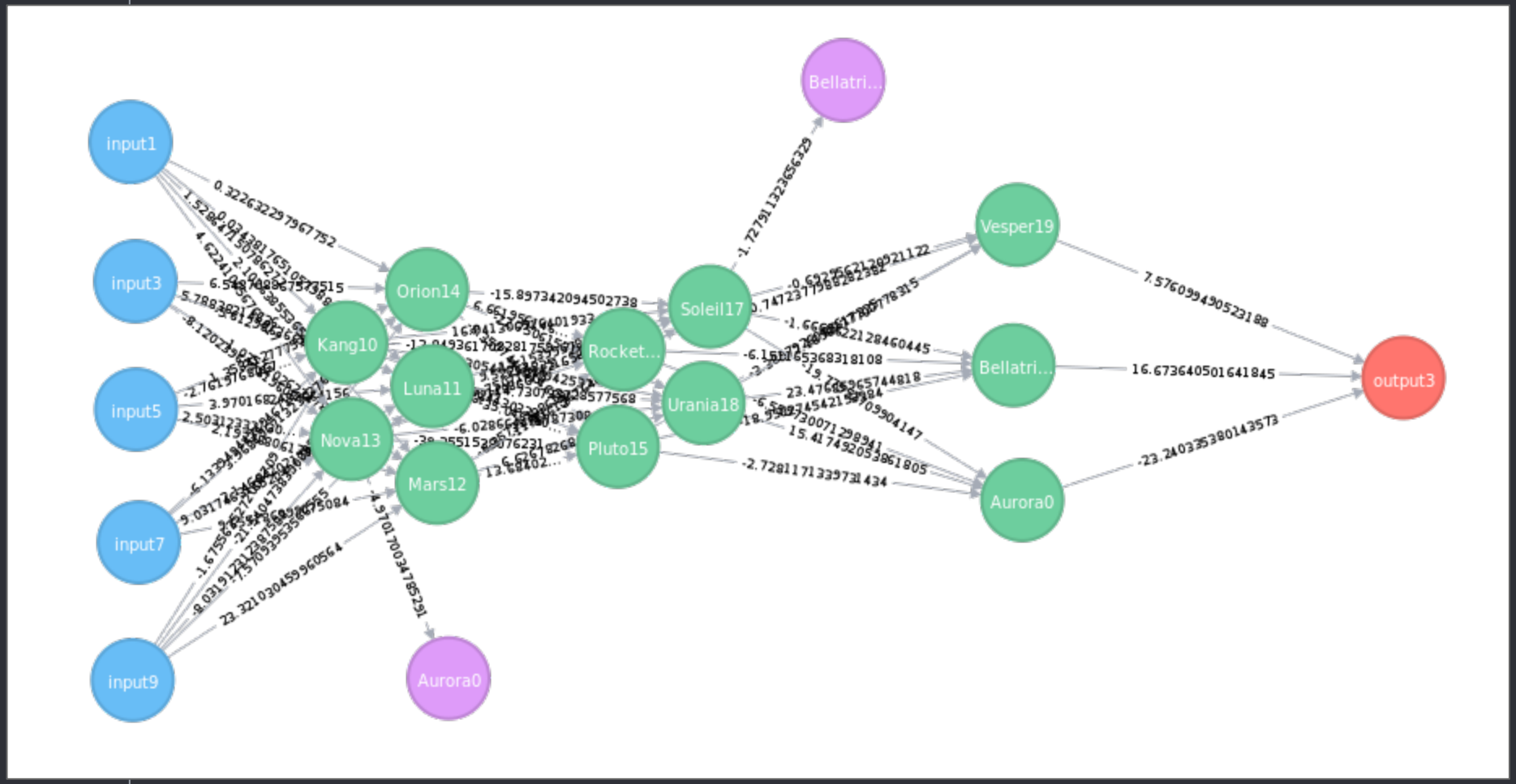
```
Average steps: 188.566
Counter({112: 11, 75: 10, 76: 10, 82: 10, 97: 1
0: 7, 124: 7, 149: 7, 159: 7, 182: 7, 60: 6, 64
: 5, 96: 5, 99: 5, 100: 5, 103: 5, 119: 5, 135:
5, 44: 4, 50: 4, 52: 4, 53: 4, 54: 4, 55: 4, 58
: 4, 141: 4, 146: 4, 150: 4, 162: 4, 165: 4, 17
3, 56: 3, 71: 3, 80: 3, 90: 3, 91: 3, 94: 3, 95
 3, 143: 3, 153: 3, 156: 3, 158: 3, 163: 3, 167
 3, 285: 3, 299: 3, 305: 3, 318: 3, 325: 3, 327
 2, 104: 2, 107: 2, 118: 2, 122: 2, 125: 2, 152
 2, 194: 2, 195: 2, 197: 2, 198: 2, 202: 2, 206
 2, 251: 2, 253: 2, 261: 2, 266: 2, 268: 2, 277
 2, 329: 2, 333: 2, 335: 2, 343: 2, 352: 2, 355
 2, 443: 2, 450: 2, 502: 2, 526: 2, 532: 2, 587
, 129: 1, 140: 1, 166: 1, 181: 1, 200: 1, 207:
, 248: 1, 250: 1, 252: 1, 254: 1, 256: 1, 257:
, 297: 1, 298: 1, 300: 1, 302: 1, 303: 1, 308:
, 357: 1, 362: 1, 363: 1, 367: 1, 369: 1, 372:
, 417: 1, 418: 1, 419: 1, 425: 1, 428: 1, 429:
, 487: 1, 490: 1, 494: 1, 517: 1, 548: 1, 553:
, 647: 1, 682: 1, 793: 1, 809: 1, 917: 1})
Average score: 9.386
```

1 network

5 layers

18 neurons

Summary


input1
input3
input5
input7
input9
Orion14
Kang10
Luna11
Nova13
Mars12
Aurora0
Rocket...
Pluto15
Soleil17
Urania18
Bellatri...
Vesper19
Bellatri...
Aurora0
output3
0.32263229796775 2
-15.897342094502738
-1.7279113236563 29
7.576099430523188
16.673640501641845
-23.240353801435 73
-2.7281171339731434
4.9701700347852 91
23.321030459960564

Thanks!

ANY QUESTIONS?

Thanks!

ANY QUESTIONS?