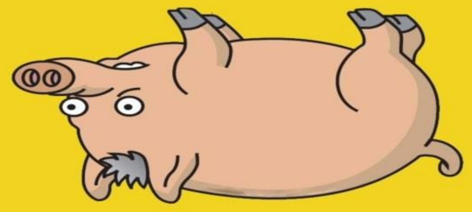
Dipartimento di Fisica
Sapienza
Università di Roma

Implementation of a neural network for poker behavior classification

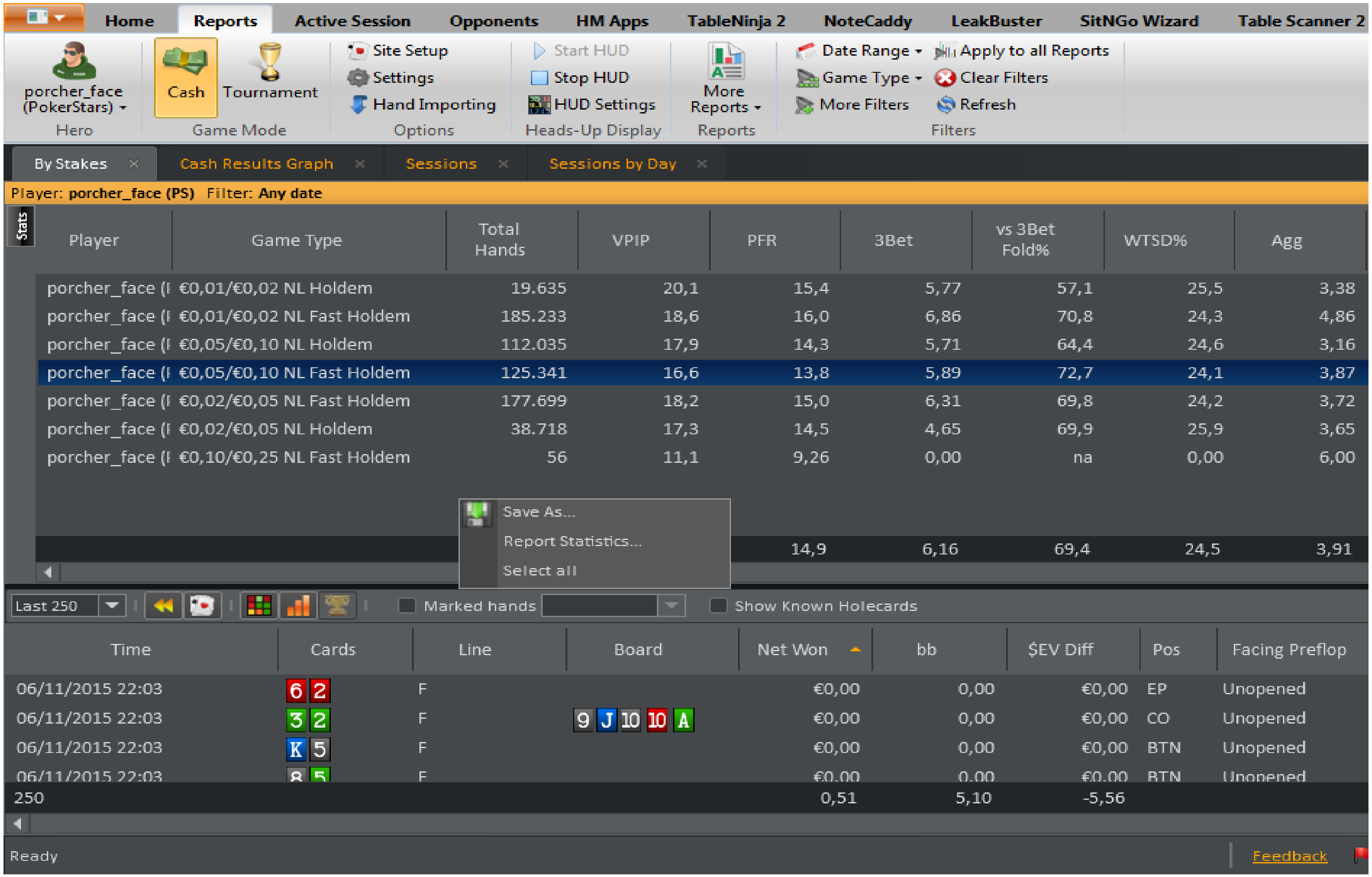


Andrea Mazzei

Fast, introductory frame.

Aim of this work is to generate a pattern recognition tool to distinguish winning regular poker players from losing ones.

Data entry from HM



Passing data to PatRec tool

All data is gathered from my personal poker account and is recorded by the third party software Hold'em Manager 2 (HM). HM is able to save players' reports in .csv files. Each report must be saved manually.

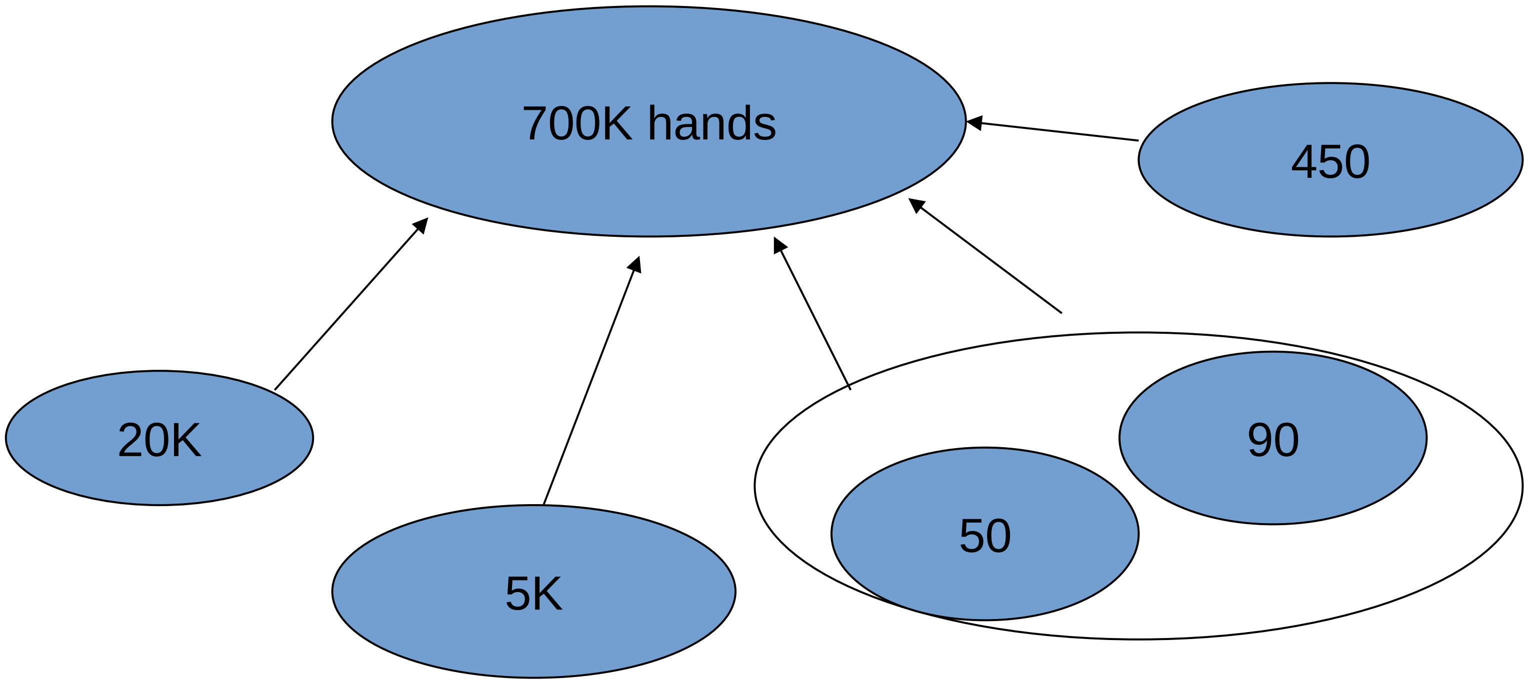
In order to adapt to the input structure requested by the pattern recognition tools provided by matlab, a bit of data manipulation has been implemented

```
"Player","Game Type","Total Hands","VPIP","PFR","3Bet","vs 3Bet Fold%","WTSD%","Agg","Flop CBet%","Turn CBet%","Flop Fold vs Cbet","bb/100"
"foky07 (PS)","€0,02/€0,05 NL Fast Holdem","1559","24,13","19,00","10,24","44,12","20,76","3,73","78,18","69,44","65,57","-7,30"
"foky07 (PS)","€0,05/€0,10 NL Fast H[illegible]em","4550","21,94","16,83","8,43","39,60","25,00","2,53","63,16","53,52","52,86","-38,96"
"foky07 (PS)","€0,05/€0,10 NL Holdem[illegible]14216"[illegible]20,51","16,69","8,85","48,78","26,81","2,53","59,76","52,99","57,41","-9,29"
"foky07 (PS)","€0,02/€0,05 NL Holdem[illegible]2135",[illegible]9,46",[illegible],71","11,17","41,94","26,70","2,39","78,91","35,00","70,97","37,50"
```

DATA <11x130 double>

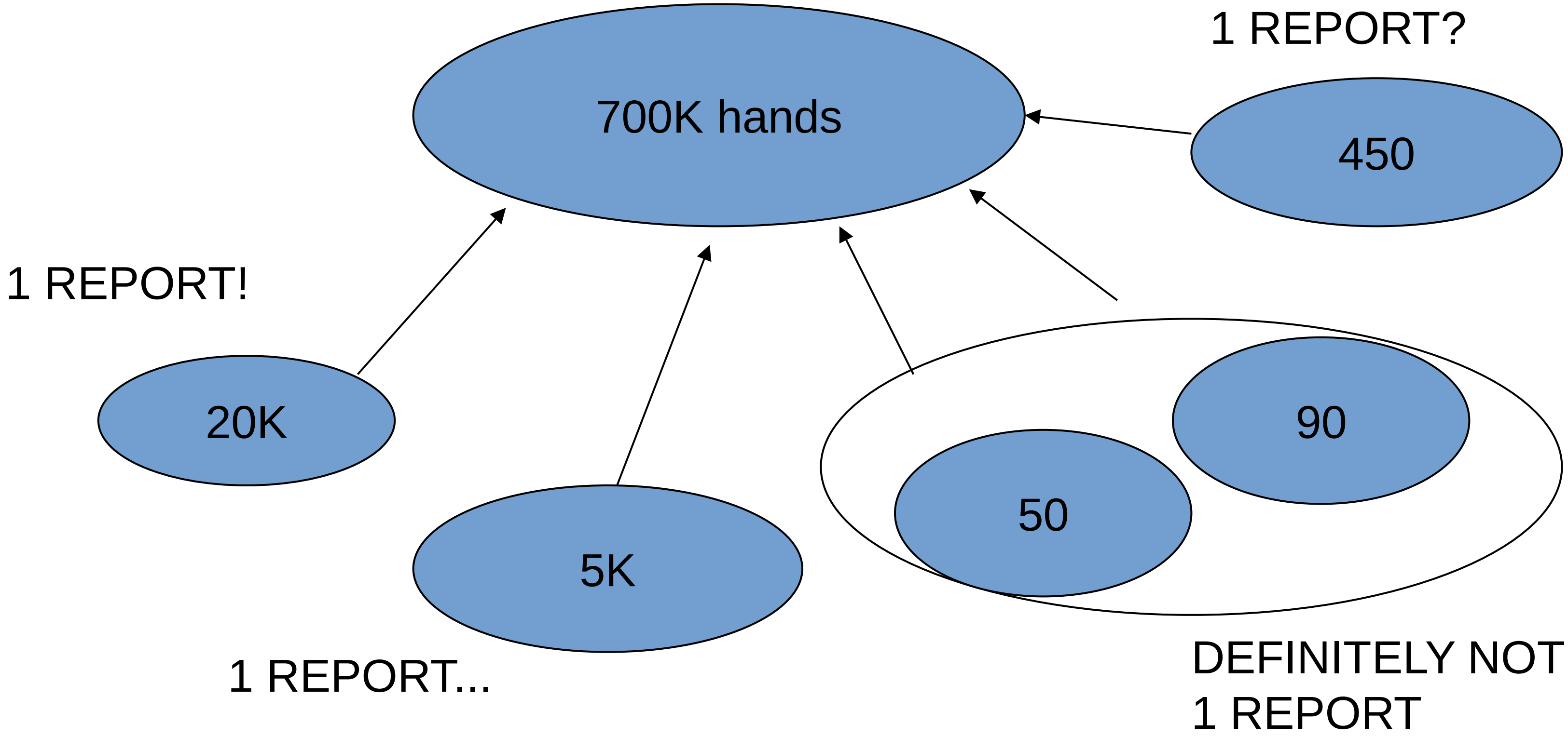
	1	2	3	4	5	6	7	8
1	15.5900	45.5000	142.1600	21.3500	66.9500	42.4100	56.1400	44.7100
2	0.2413	0.2194	0.2051	0.1946	0.1746	0.2006	0.1785	0.1737
3	0.1900	0.1683	0.1669	0.1771	0.1444	0.1629	0.1444	0.1228
4	0.1024	0.0843	0.0885	0.1117	0.0621	0.0621	0.0525	0.0235
5	0.4412	0.3960	0.4878	0.4194	0.5000	0.4024	0.5738	0.5287
6	0.2076	0.2500	0.2681	0.2670	0.2659	0.3404	0.2559	0.2769
7	0.0373	0.0253	0.0253	0.0239	0.0321	0.0220	0.0341	0.0199
8	0.7818	0.6316	0.5976	0.7891	0.7867	0.6742	0.4645	0.5381
9	0.6944	0.5352	0.5299	0.3500	0.6772	0.6176	0.4559	0.3958
10	0.6557	0.5286	0.5741	0.7097	0.4836	0.5109	0.6699	0.6435
11	-0.0730	-0.3896	-0.0929	0.3750	-0.1273	-0.0367	0.0535	-0.1293
12								

BIG data?



Fishy opponents play less! A possibility would be to gather all of em into an alias

BIG data?

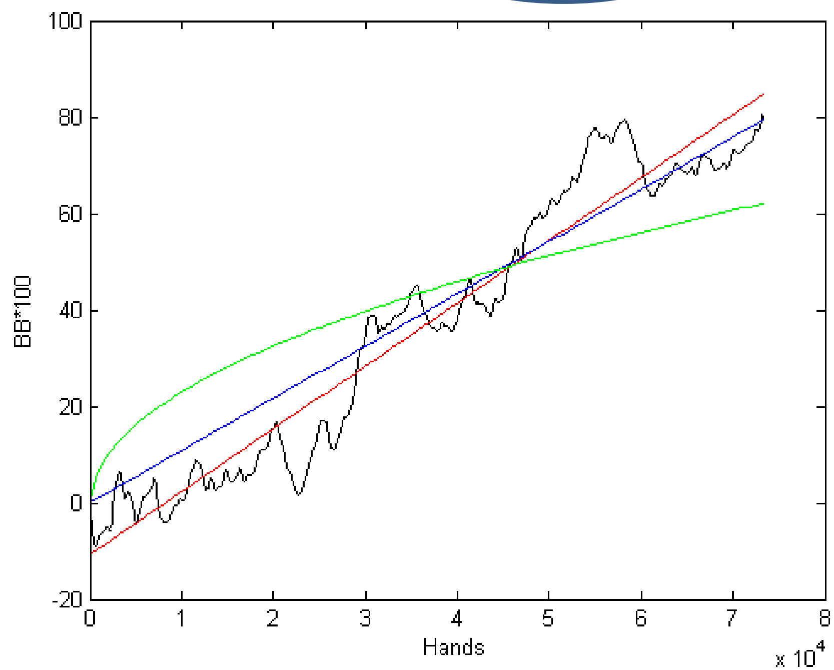


Fishy opponents play less! A possibility would be to gather all of em into an alias

Correct definition of “BIG”

Online poker is essentially aleatory. By examining the stochastic process associated to hero's personal hands, we can infer information on the variance of the process. Our aim is now to define what is a “good” sample.

2K hands?



The neural network toolbox

Input data (“features”): The list of sensible stats available for the players. Those range from very common stats to very improbable one (as example “Missed cbet turn after check raising flop on 3bet pot”, completely useless unless we got a good sample. It should range around 100K hands on that opponent).

Some stats are easy to mine (VPIP and PFR take 1 hand per sample), while some others are gathered much more slowly (some actions are “rare” events and also depend on the possibility of committing that action).

Input data (“target”): The oracle for the network, seems pretty legit to consider their winrate ($/hand, more properly expressed in bb/100hands) as a good oracle... however some precautions must be taken

The features vector

(HANDS,VPIP,PFR,3BET,CCL,CBET,FCB,WTSD,STL,AGGR) 10 elements

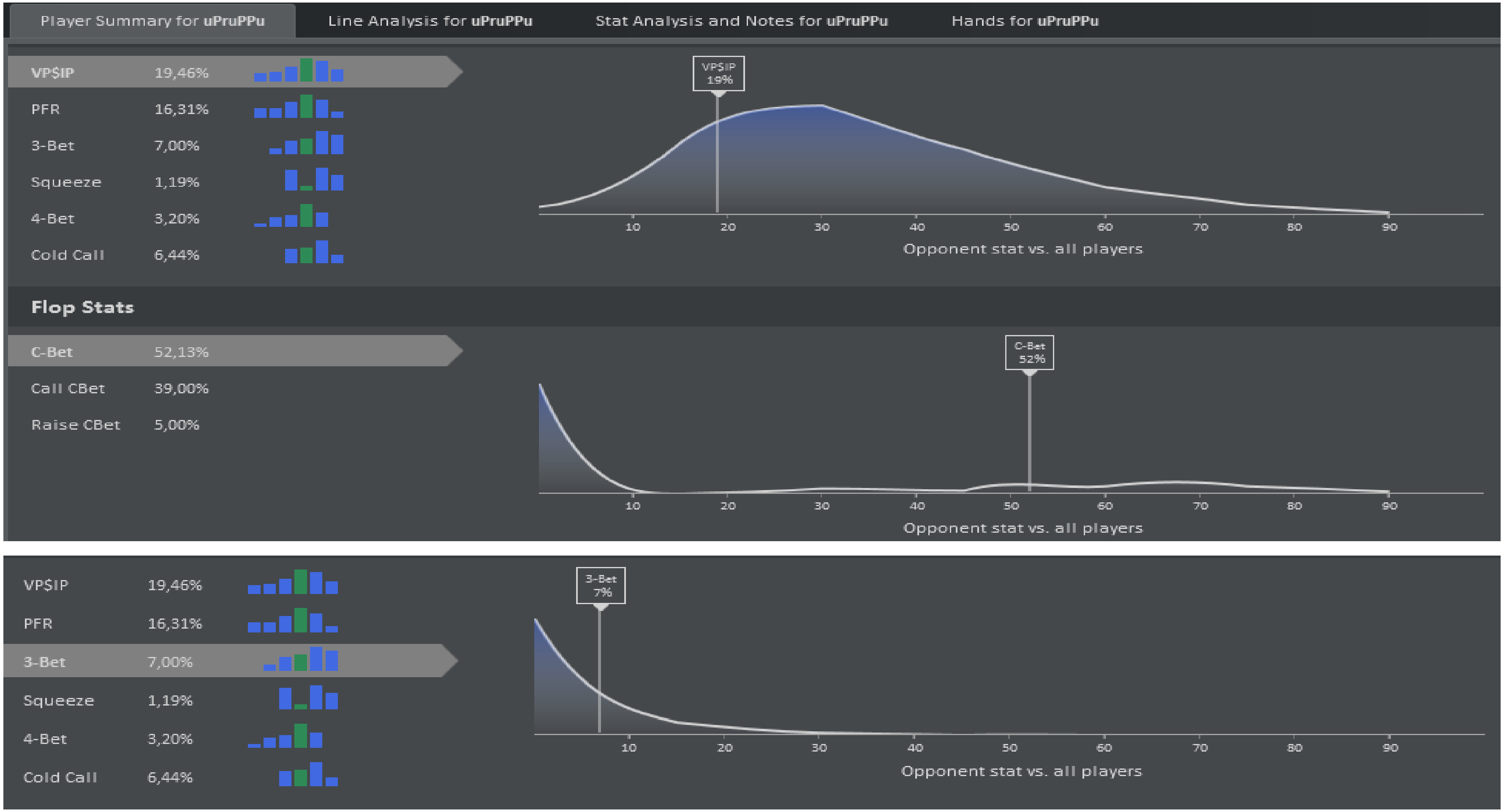
The features vector

(HANDS,VPIP,PFR,3BET,CCL,CBET,FCB,WTSD,STL,AGGR) 10 elements

(HANDS,VPIP,PFR,3BET,F3B,CBET,FCB,~~WTSD~~,~~STL~~,~~AGGR~~) 5-6 elements

Poker player experience comes handy here.

Discriminatory power (qualitatively)



The oracle?

Intuition suggest that the correct question for determining a good player is:

Is he winning money? (or, better, is his bb/100 positive?)

Since the response suffers of indetermination, a good player might has been recorded as non-winning. A way to overcome is to consider the so-called all-inEV adjusted winrate.

-330 bb at the end of the game session? Bad player or bad classification?

All-in EV winnings

VS

Net winnings

Improving the oracle.

Improving the oracle.

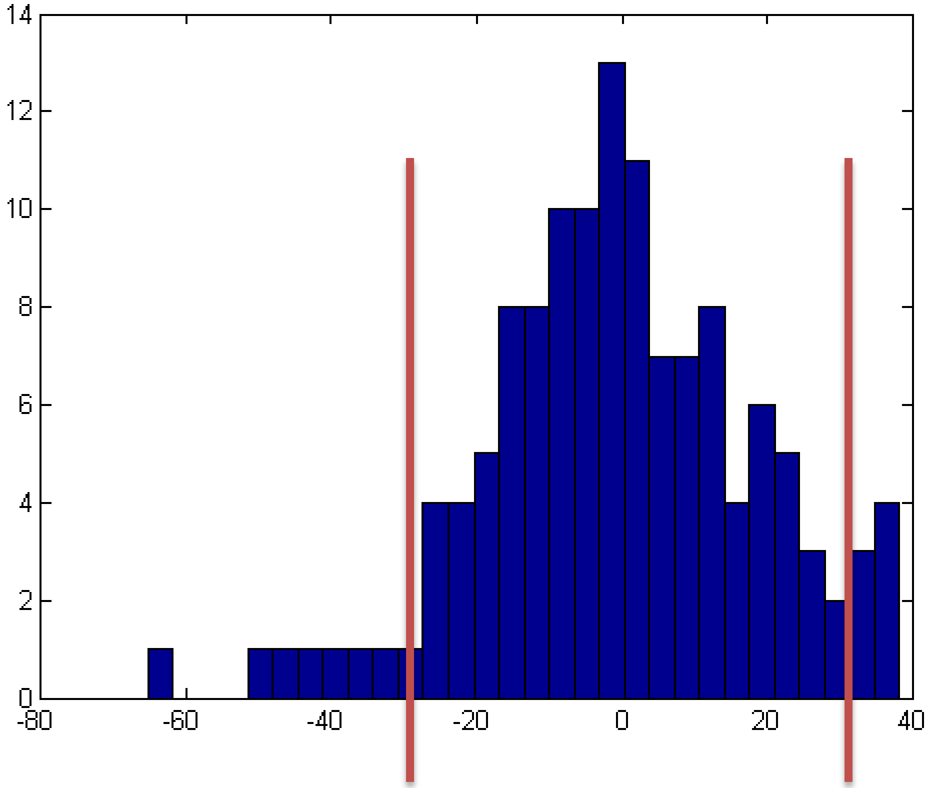
Net Won is -116bb

Adj. Won is -116bb*0.17 + 109bb*0.83 = 71bb

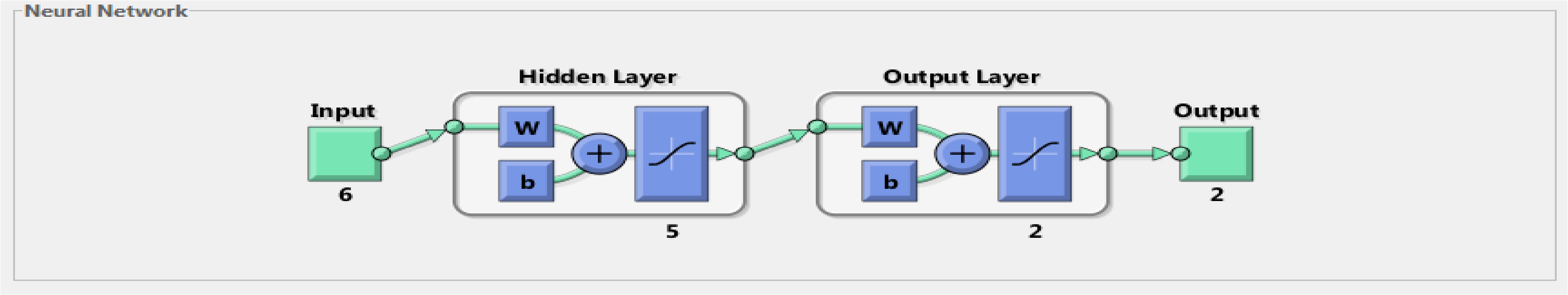
In this way we are able to remove, at least partially, the aleatoric character of the target set.

Improving the oracle.

Moreover, player with small game samples may have been recorded in a surprisingly lucky or unlucky session (variance rush). In order to balance this behaviour, players at the tails of the bb100 distribution have been cut out.



The neural network toolbox



Properties of the network:

- #layers: 1 + 1 hidden.
- Acceptance function: sigmoid.
- Output: binary (two classes).
- Method: gradient descent.

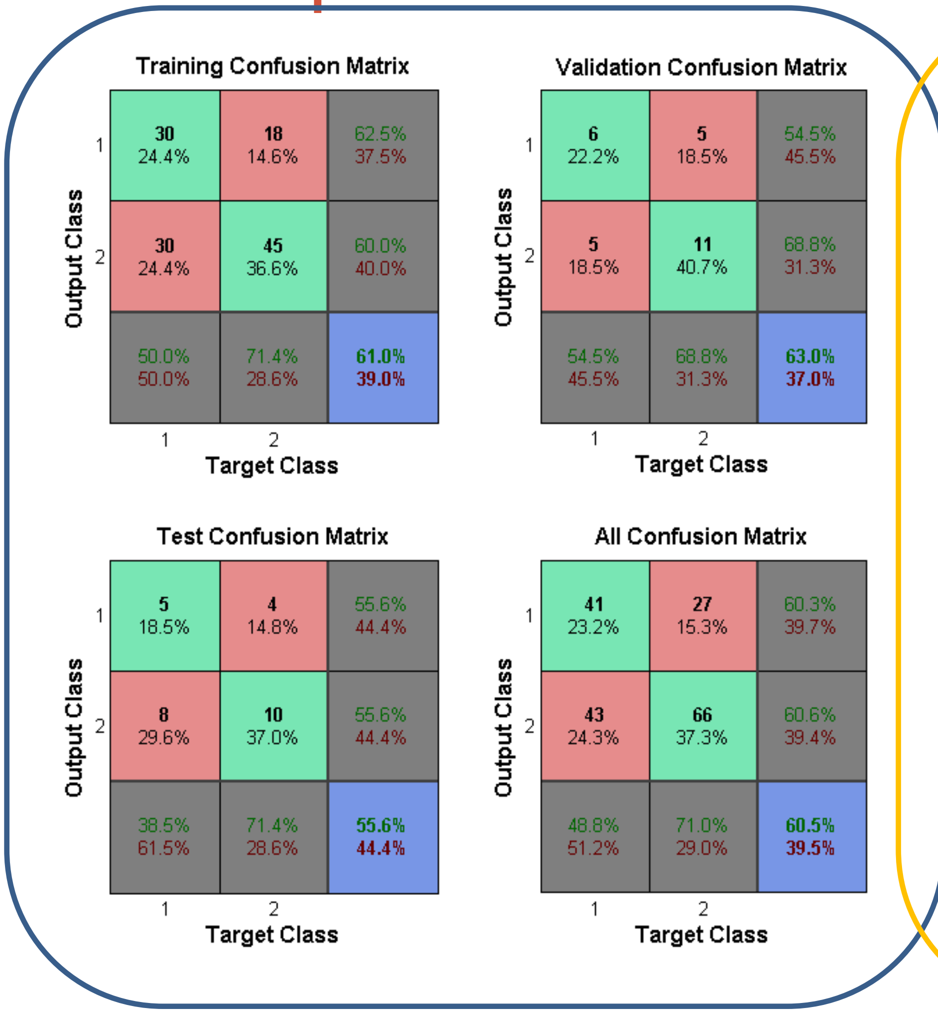
Training of the network

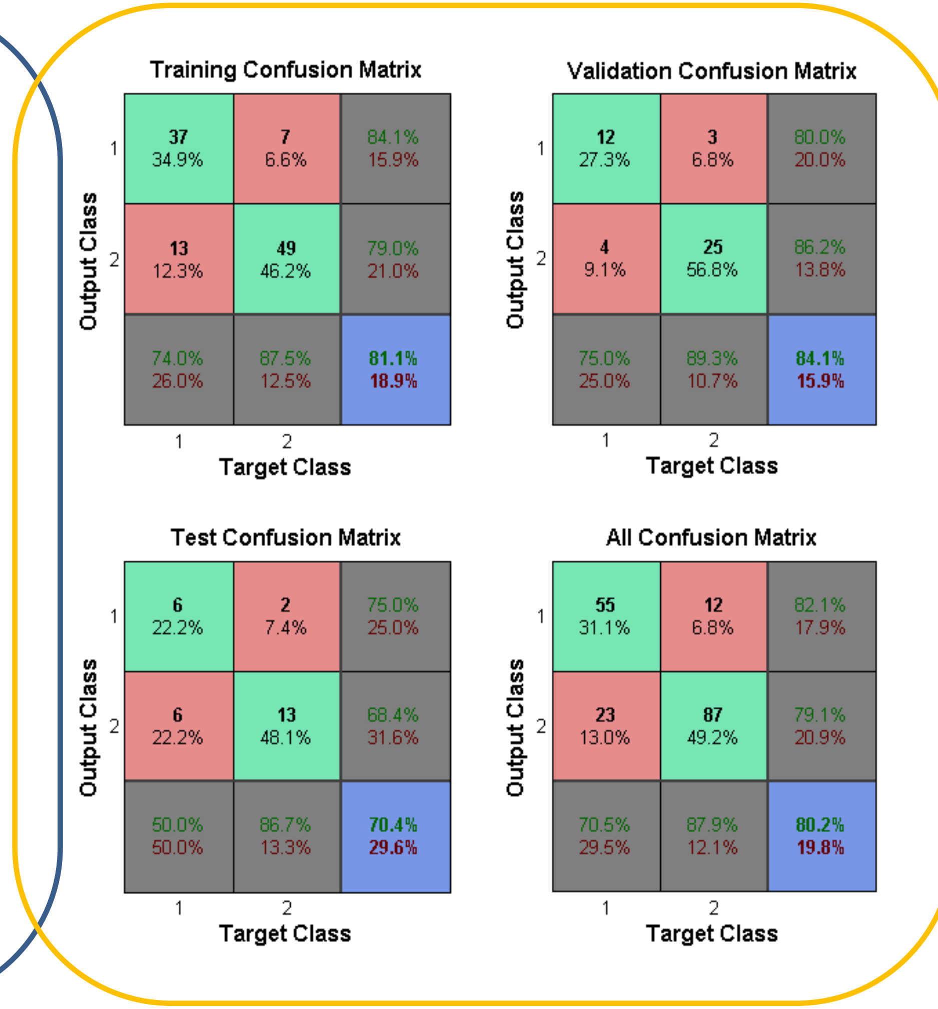
Data is composed of 200 different reports, with a cutoff at 1000 hands. If the number of hands in the report is lesser than the cutoff, the report is ignored.
If we lower the hand-cutoff we get more data for the training, but we increment the error associated to the oracle.

The toolbox suggests a partitioning for the training, validation and testing elements of 70%, 15%, 15%. Unfortunately, these values restrict our count to only a few element for the latter two phases. I prefer a slightly higher sample for the validation partition, around 25%.

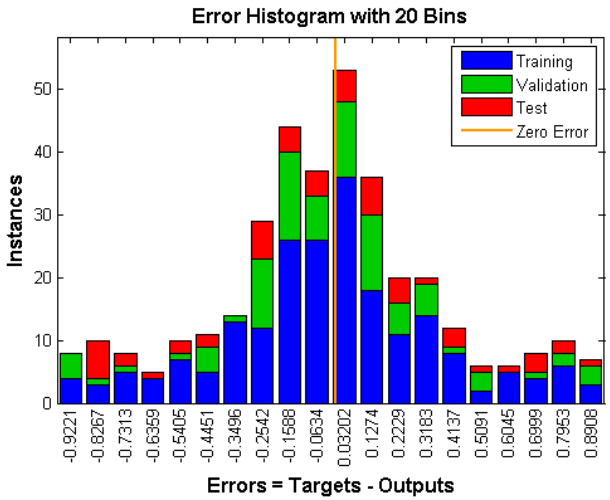
The number of neurons has been varied manually in a range from two to ten.
The majority of the runs have been made with a value of four and five neurons.

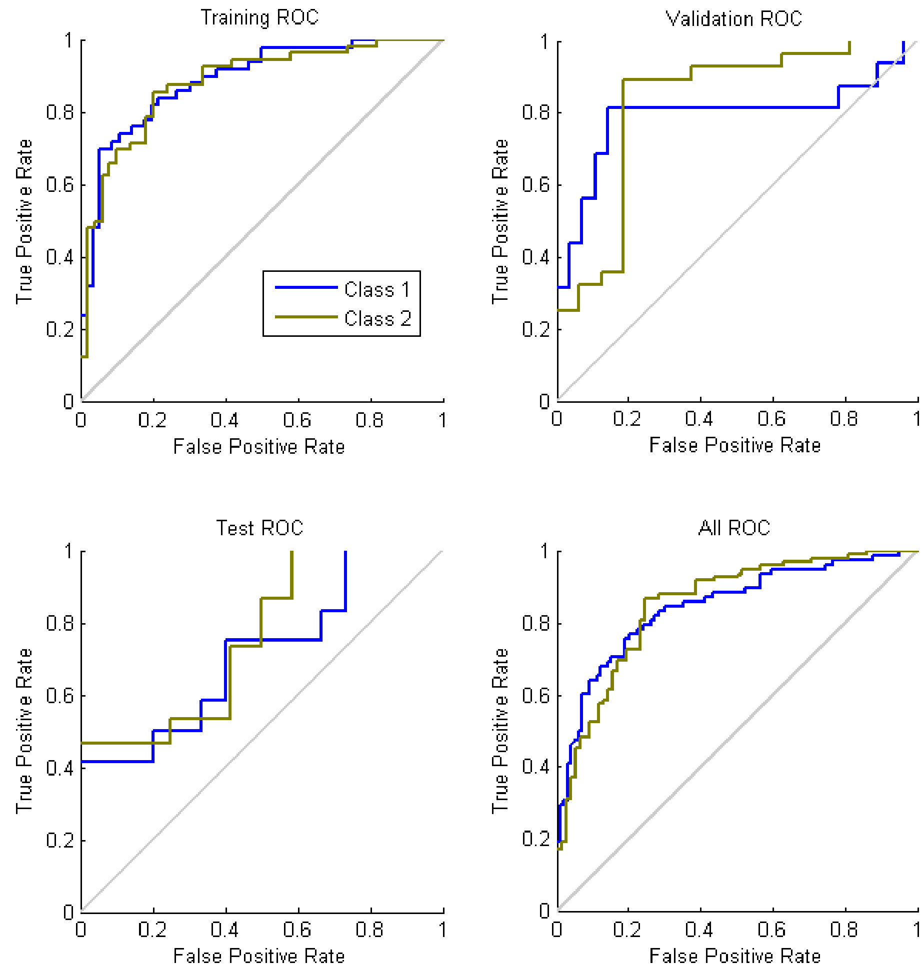
Output of the network.





Output of the network.



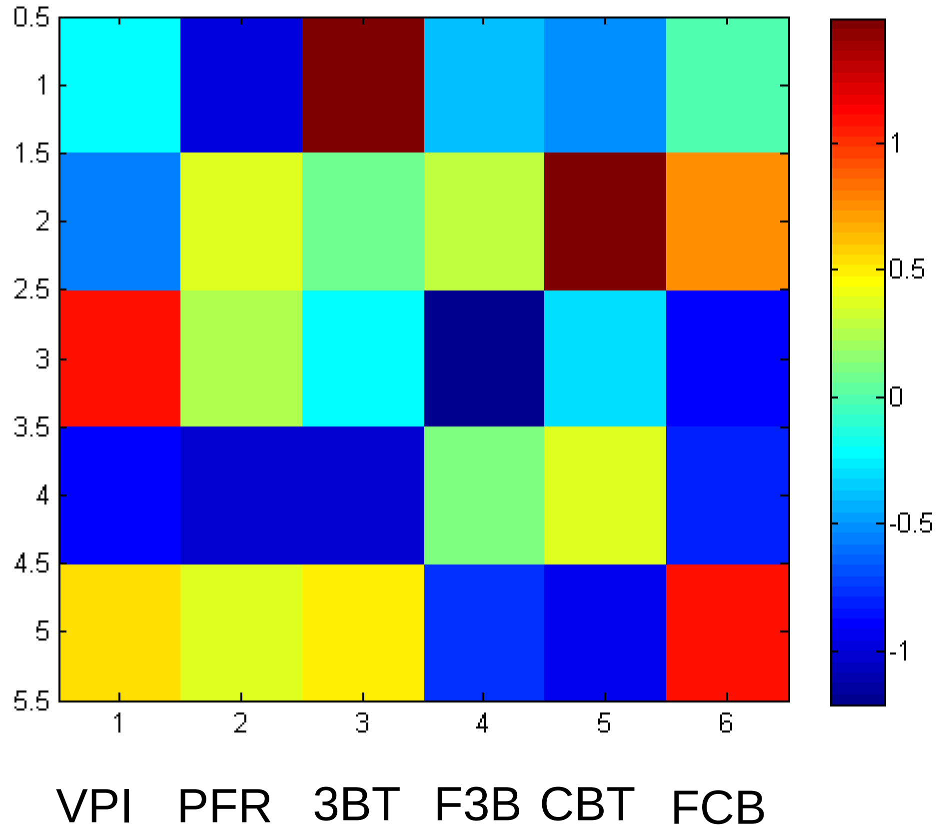


Error distribution is still large

Interpreting the output.

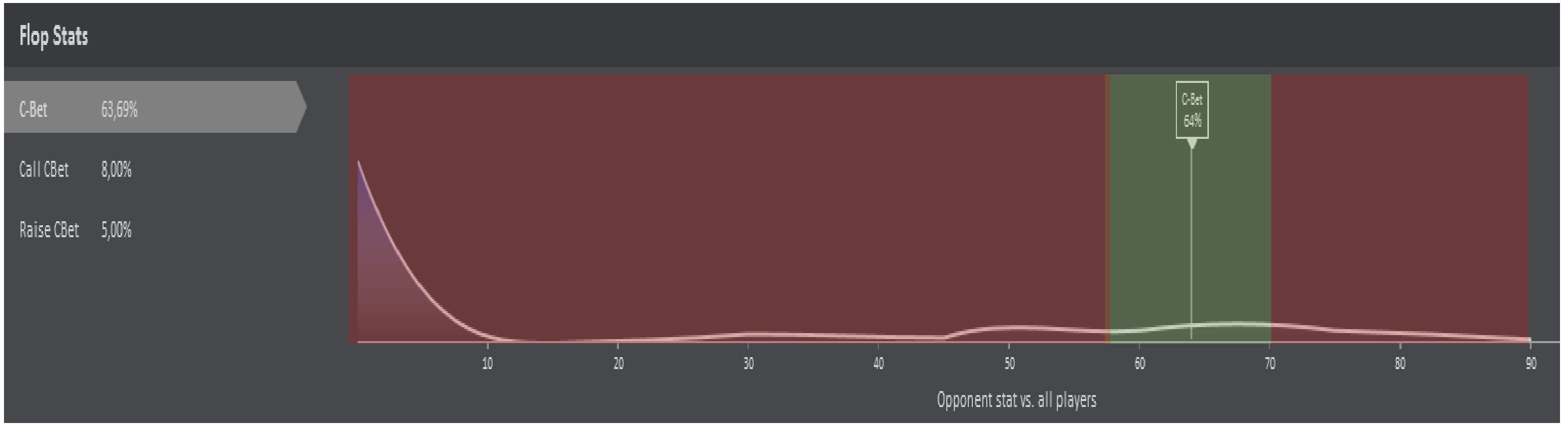
Colormap for the hidden layer weights. It is possible to read some information here.

As expected, FCB and PFR value are strong indicators of the presence of Category 2 player, whilst the 3BET stat and CBET stat are strong indicators of the opposite.



Non linear-separability?

The Continuation bet example:
Poker theory suggests to cbet as frequently as possible, combined to the evaluation on how much our opponent is likely to fold. On low stakes cash games, due to the vastity of fishes in the field with a low fold to cbet value, cbet optimal ranges have a net lower and upper bound.



A SVM classification

Support vector machines (SVM) are an example of supervised learning algorithm capable of performing non linear classification.

I LOOOOOOOOOVE SCIENCE!

Considerations and conclusions.

I like science

Thanks for reading,

Email:
andreamazzei88@gmail.com
porcherface@gmail.com