## DRL Bridge bidding Inventor

An attempt to learn and use bidding systems.

As an example we used 1 Spade(5+,12-21) – 1 NT (0-2 spades, 8-11).

Learning rates 1e-3, 1e-4, 1e-5, 1e-6, 1e-7 was tried; not much difference but 1e-6 sometimes better.

## Experiences;

1. Gamma = 0.95 reduces the future values by this factor and encourages short biddings?

Well, because we compare with par-result and losing imps the results are negative resulting in strange learning.

Solution: Add offset +20 imps to results to keep them positive (+20 corresponds to par-result), subtract 20 when printing/showing to the user.

2. Adding more bids actually got worse performance than just a single rebid (-4 imps/deal); what to do if partner just bid random 3 C or 4D after my rebid? – better to pass and play in 1 NT.

Solution: Don't allow Pass as rebid..later: create new examples with excluded hands like (semi)bal 5 sp and 11-13 h.c.p assuming to be passed.

3. Couldn't solve DA2C-algorithm on GPU.. but worked good on multi-core CPUs and actually faster. Now a run with 2 million trial bids took 15 mins on a 5 yr old i5-8700K with 12 hyperthreaded cores, using 10 and saving 2 for GUI etc.

Examples (Learning rate = 1e-6, two bids, no pass-rebid)

```
1♠-Bidding Contract Score Impdiff (Parscore) N[shape,hcp] S[shape,hpc] N-cards E-cards S-cards W-cards
1N 2 2N -100 -5 ( 110 ) [6, 2, 3, 2, 15] [1, 3, 3, 6, 9] n AKJ987.A6.K64.T8 e 642.873.AQJ93.32 s T.KQ5.T82.KJ7654 w Q53.JT942.75.AQ9
1N 3 3 170 -6 ( 420 ) [5, 1, 5, 2, 18] [2, 4, 2, 5, 8] n AQJ84.A.AKT52.63 e 7.862.Q986.KQT54 s K6.K975.J3.J9872 w T9532.QJT43.74.Ā
1N 2 2N 180 -6 ( 430 ) [5, 3, 3, 2, 15] [2, 2, 5, 4, 8] n AQJ74.KJT.JT8.K3 e 52.74.9764.A8764 s T9.65.KQ532.QJ92 w K863.AQ9832.A.T5
1N 24 2N 180 -7 (450) [6, 1, 1, 5, 13] [2, 4, 3, 4, 11] n AJ9853.K.4.KQT96 e 764.Q62.KJ972.A5 s T2.A3P55.AQT.J842 w KQ_JT843.8653.73
1N 34 44 190 -15 (1440) [6, 2, 3, 2, 15] [1, 3, 3, 6, 11] n AQ9632.AQ.K96.T9 e T874.KJT9854.5.K s 5.732.A32.AQJ864 w KJ.6.QJT874.7532
1N 2 130 -8 (450) [5, 3, 4, 1, 15] [1, 5, 3, 4, 9] n AT872.A63.K974.A e KQ43.T98.63.Q953 s 5.KQJ75.T85.KT74 w J96.42.AQJ2.J862
1N 2N 3♣ 170 -13 ( 920 ) [5, 2, 5, 1, 19] [1, 1, 4, 7, 9] n AQT53.QJ.AKQJ9.T e KJ874.A852.T6.J6 s 2.9.8743.AKQ8754 w 96.KT7643.52.932
1N 2 2N 150 -7 ( 430 ) [5, 1, 4, 3, 15] [2, 4, 3, 4, 10] n AKJ86.9.AK95.654 e T972.J853.74.AJ3 s Q3.KQ76.T62.K987 w 54.AT42.QJ83.QT2
1N 2 ◆ 2 ★ 230 -13 ( 990 ) [5, 3, 1, 4, 20] [2, 4, 5, 2, 9] n AKT32.KT2.Q.AKJ9 e 965.Q85.A762.875 s J4.AJ97.KT983.32 w Q87.643.J54.QT64
1N 2 ★ 3 ★ -250 -11 (300) [5, 5, 2, 1, 12] [2, 5, 2, 4, 11] n AQT96.KQ954.J9.8 e J543.A7.AT7642.K s K7.JT832.KQ.Q653 w 82.6.853.AJT9742 1N 2 ★ 3 ★ 110 -7 (420) [6, 1, 1, 5, 16] [2, 5, 4, 2, 9] n AKQ984.K.5.KJ852 e J2.T87.QJ84.QT94 s T6.Q9542.AK97.63 w 753.AJ63.T632.A7
1N 2 2N 150 -6 ( 400 ) [5, 3, 3, 2, 15] [1, 3, 4, 5, 11] n A8432.K83.A54.KJ e T.A752.762.98765 s Q.Q64.KQT8.QT432 w KJ9765.JT9.J93.A
1N 2♥ 2N 150 -7 ( 420 ) [5, 3, 4, 1, 16] [1, 5, 4, 3, 11] n AKT87.K72.KQJ2.7 e J54.QJT4.3.AQ984 s 2.A8653.A864.KT5 w Q963.9.T975.J632 1N 2N 3♣ -50 -14 ( 920 ) [5, 2, 4, 2, 17] [1, 5, 4, 3, 11] n KQT76.A6.K432.KQ e 53.KJ5.QT76.T942 s J.T9743.AJ95.AJ8 w A9842.Q82.8.7653
1N 2N 3♥ -250 -8 ( 100 ) [6, 3, 2, 2, 16] [2, 2, 6, 3, 8] n AKT984.K32.K3.K4 e J652.AJ87.9.AQ93 s 73.65.AQJ765.JT5 w Q.QT94.T842.8762
1N 2 2N -100 -5 (110) [6, 2, 2, 3, 13] [2, 5, 3, 3, 9] n K96532.A5.AQ.T42 e J4.KJT8.KJ952.A7 s A7.Q9764.T74.QJ6 w QT8.32.863.K9853 1N 3 2 170 -6 (430) [5, 3, 1, 4, 17] [2, 4, 4, 3, 9] n KQT76.K92.8.AKQ5 e AJ2.Q74.AJ732.97 s 53.A865.KQT4.T86 w 984.JT3.965.J432
1N 3♣ 3♥ -100 -5 ( 110 ) [5, 4, 2, 2, 12] [2, 3, 5, 3, 10] n AQ632.Q873.A2.76 e 84.42.KQT85.J854 s T9.K65.J9743.AQ2 w KJ75.AJT9.6.KT93
1N 3 4 4 -100 -6 ( 120 ) [5, 1, 4, 3, 12] [1, 5, 3, 4, 10] n K9864.Q.AK42.985 e 2.AJ963.QT9.AJ73 s Q.KT754.875.KQT4 w AJT753.82.J63.62
1N 2 € 2 € 140 0 (140 ) [5, 2, 4, 2, 12] [2, 4, 4, 3, 10] n KQT82.Q2.KQ84.43 e J6.K763.972.T762 s 94.AJ54.JT53.A95 w A753.T98.A6.KQJ8
1N 3 4 4 -100 -11 ( 420 ) [5, 1, 3, 4, 14] [2, 6, 4, 1, 11] n AKQT9.7.Q83.QJT6 e J632.A6.65.AK983 s 74.KJT842.AK42.7 w 85.Q953.JT97.542
1N 3 3N -50 -5 (140) [5, 2, 4, 2, 14] [1, 5, 4, 3, 9] n A9754.96.AKQ4.J8 e KQJ6.K5.T9.A7653 s 2.A8732.8652.KQ9 w T83.QJT4.J73.T42 1N 2 2N 210 -6 (460) [6, 2, 1, 4, 14] [2, 2, 5, 4, 11] n AK8732.AQ.T.JT64 e T.JT86542.K8753 s Q4.97.QJ542.AQ92 w J965.K3.AK98763.
1N 2 2 2 110 0 ( 110 ) [5, 4, 3, 1, 12] [2, 1, 4, 6, 10] n AQT54.J863.AJ4.6 e J9.AKT97.T6.J854 s K3.5.Q752.KQ9732 w 8762.Q42.K983.AT
1N 3★ 3★ -50 -10 ( 400 ) [5, 1, 3, 4, 17] [2, 5, 3, 3, 10] n QJT74.J.AT6.AKQT e 3.Q7653.Q42.J962 s K5.AKT98.975.874 w A9862.42.KJ83.53 1N 2N 3★ -150 7 ( -450 ) [5, 2, 4, 2, 14] [2, 2, 5, 4, 11] n AK853.Q9.KQ32.82 e QT7642.A6542.T.9 s J9.KJ.A9754.QT54 w .T873.J86.AKJ763
```

Not very useful but can be improved by removing weak openers and with more training.

## Running with 3 possible bids 2 miljon times;

Ended with opener always bidding 2 C (instead of always pass). Learning rate 1e-6 was used, maybe another run can be better.

A try with bigger Neural net (from 75 to 100 neurons) and 11 possible bids instead of 6 resulted in 2C – pass on all deals, -7.7 imps/deal, worse..

A try with smaller net with 50 neurons and reducing layers (fully connected+ReLU) from 4 to 3: 2C-pass on all as well..

## Suggestions/Ideas/Improvement

- maybe there are some bugs in the code
- trying to invent bidding backwards, reducing search space no use to try 7NT with 13+9 hcp.
- try other algs like PPO A3C, Curiosity-based, Multi-Agent RL, evolutionary methods