Hanabi : Learning by Reinforcement

Bruno Bouzy

LIPADE
Paris Descartes University

Ongoing research

October, 2017

Outline

- The game of Hanabi, Previous work
- Learning by Reinforcement (ongoing research)
  - Shallow learning with « Deep » ideas
  - Experiments and Results
  - Hanabi Challenges
    - How to learn a convention ?
- Conclusions and future work

Hanabi Game Set

Hanabi features

- Card game
- Cooperative game with N players
- Hidden information : the deck and my cards
- I see the cards of my partners
- Explicit information moves

Previous work

- (Osawa 2015) : Partner models, NP=2, NCPP=5, <score> ~= 15
- (Baffier & al 2015) : Standard and open Hanabi : NP complete
- (Kosters & al 2016) : Miscellan., NP=3, NCPP=5, <score> ~= 15
- (Franz 2016) : MCTS, NP=4, NCPP=5, <score> ~= 17
- (Walton-Rivers & al 2016) : Several approaches, <score> ~= 15
- (Piers & al 2016) : Cooperative games with Partial Observability
- (Cox 2015) : Hat principle, NP=5, NCPP=4, <score> = 24.5
- (Bouzy 2017) : Depth-one search + Hat, NP in {2, 3, 4, 5} NCPP in {3, 4, 5}

Learning by Reinforcement

- Deep Learning is the current trend
  - Facial recognition (2014, 2015)
  - Alfago (2016, 2017)
- Deep RL for Hanabi ?
- Let us start with shallow RL
  - (Sutton & Barto 1998)
- Approximate Q or V with a neural network.
  - QN approach

Relaxing the rules or not

- Always :
  - I can see the cards of my partners
  - I cannot see the deck
- Open Hanabi
  - I can see my cards (seer of previous part)
- Standard Hanabi
  - I cannot see my cards

Neural network for Function Approximation

- One neural network shared by each player
- Inputs
  - Open Hanabi (81 boolean values for NP=3 and NCPJ=3)
  - Standard Hanabi (133 boolean values for NP=3 and NCPJ=3)
- One hidden layer and NUPL units
  - (NUPL=10, 20, 40, 80, 160)
  - Two layers or three-layers were tried, but unsuccessfully
- Sigmoid for hidden units
- No sigmoid for the output
- Output used to approximate
  - V value
  - Q value

Inputs

- Always
  - 5 firework values, 25 dispensable values
  - Deck size, current score,
  - # red tokens, # remaining turns
- Open Hanabi
  - For each card in my hand,
    - Card value, dispensable, dead, playable
- Standard Hanabi
  - # blue tokens,
  - For each card in my hand,
    - Information about color, information about rank
  - For each partner,
    - For each card,
      - Card value, dispensable, dead, playable
      - Information about color, information about rank

Inputs

- Open Hanabi

NP \ NCPP	3	4	5
any	81	89	97

- Standard Hanabi

NP \ NCPP	3	4	5
2	106	121	136
3	133	157	181
4	160	193	226
5	187	229	271

Learning and testing

- Test :
  - Fixed set of 100 card distributions (CD) (seeds from 1 up to 100)
  - Average score obtained on this fixed set
  - Performed every 10^5 iterations
  - TDL : policy = TDL + depth-one search with 100 simulations (slow)
  - QL : policy = greedy on Q values (fast)
- Learn :
  - Set of 10^7 card distributions
  - Average score of the CD played so far
- 1 iteration == 1 CD == 1 game == 1 T
  - #iteration = 10^5, 10^6 or 10^7
- Interpretation
  - QL : Learning average score < Testing average score
  - TDL : Learning average score << Testing average score

Q learning versus TD Learning

- Context : Function Approximation
- Goal : Learn Q or learn V
  - TD Gammon (Tesauro & Sejnowski 1989) DQN (Mnih 2015)
  - Theoretical studies : (Tsitsiklis & Van Roy 2000), (Maei & al 2010)
- Number of states < Number of action states
  - Choose TD for an rough convergence and Q for an accurate one.
- Control policy
  - QLearning : the policy is implicit : (epsilon) greedy on the action values
  - TDLearning : the policy is a depth-one search with NCD card distributions after each action state. (NCD=1, 10, 100) : computationally heavy
- Q learning architecture
  - One network with |A| outputs. One output per action value.
    - What is the target of unused actions ?
    - All the Q values are computed in parallel. Learning is hard because done in parallel.
  - |A| networks with one output. One network per action.
    - This study

Which values, which target ?

- Our definition of V values and Q values :
  - $V_{our} = V_{usual}$ + current score
  - $Q_{our} = Q_{usual}$ + current score
  - Our study : value = expectation on the endgame score
  - Equivalent.
- Target = actual endgame score

Replay memory

- (Lin 1992) (Mnih & al 2013, 2015)
- Idea:
  - Shuffle the chronological order used at timeplay and learn on shuffled examples
  - The chronological order is bad at learning time
  - Two subsequent transitions (examples) share similarities
- After each action :
  - Store the transition into a replay memory (transition = state or action state + target)
- After each game :
  - 100 transitions are drawn at random in the replay memory
  - For each drawn transition perform one backprop step
- Replay memory size == 10k
  - (our « best » value versus 1k, 100k, 1M)

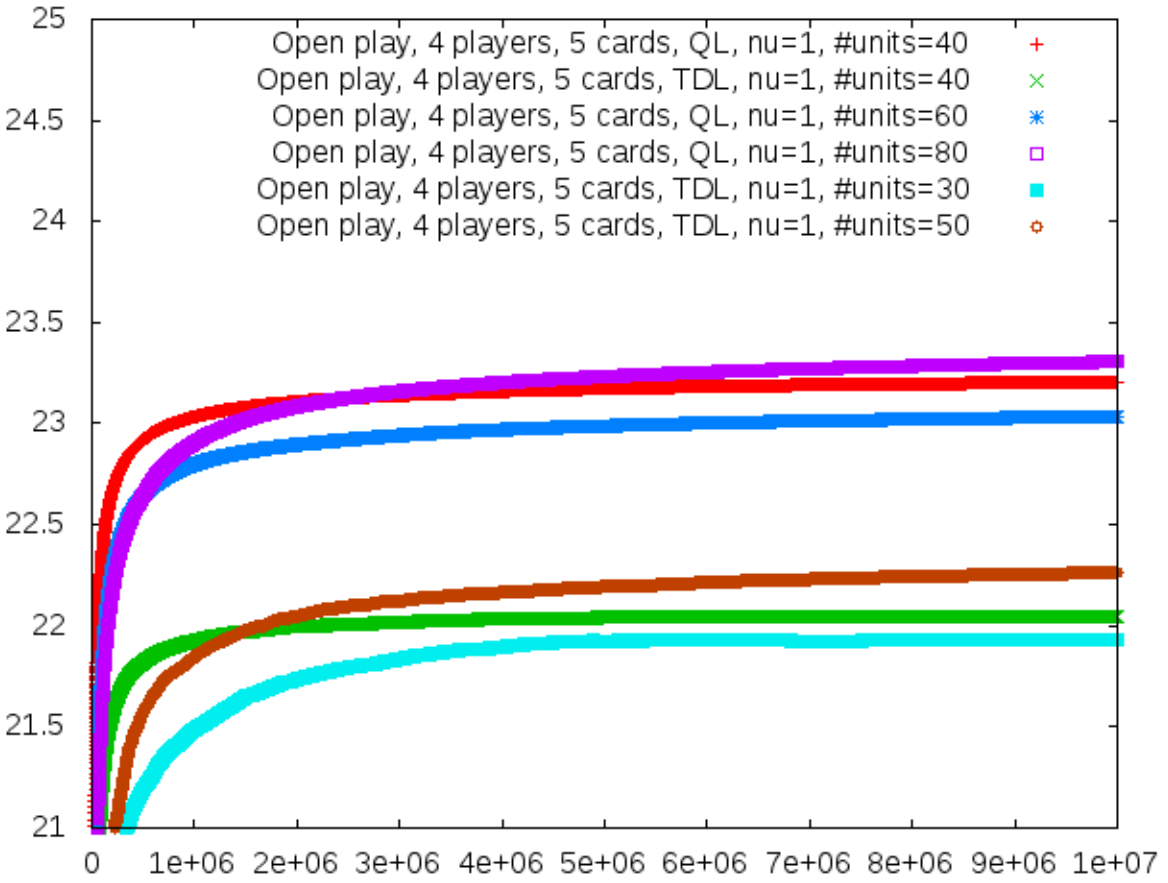
Stochastic Gradient Descent

- Many publications
  - (Bishop 1995), (Bottou 2015),
  - RL with function approximation : Non stationarity and Instability
- Tuning the learning step.
  - NU = constant value ?
  - NU = Nu_0 / sqrt(T)
    - Experimentally proved by our study
    - Better than [constant NU] or than [NU = Nu_0 / T] or than [NU = Nu_0 / (log(1+T)]
  - Many techniques :
    - momentum
    - bold driver
    - ADAM (Kingma & Ba 2014), No more pesky learning rates (Schaul 2013), Lecun's recipe (1993)
    - conjugate gradients (heavy method)
  - This study :
    - Simple momentum with parameter = 0.125 works well for TD and normal Hanabi (NP=3, NCPP=3)
    - ADAM tested but the results were inferior to our best settings.
    - Minibatches : no

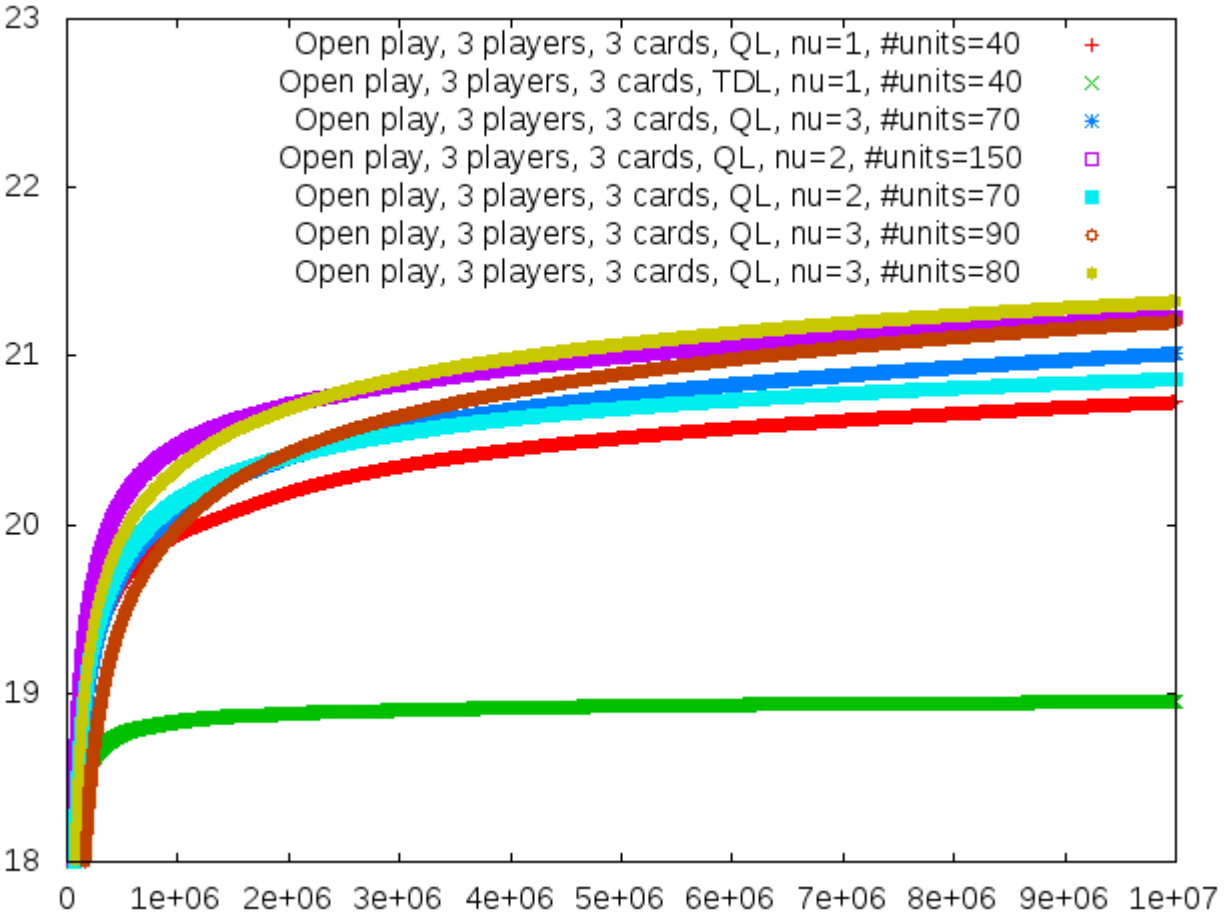
Quantitative results

- Open Hanabi (seer learners)
  - NP players (NP=2, 3, 4, 5)
  - NCPP cards per players (NCPP=3, 4, 5)
- Standard Hanabi
  - Starting with NP=2 and NCPP=3
  - One more card ? (NP=2 and NCPP=4)
  - One more player ? (NP=3 and NCPP=3)
  - The current limit (N=4 and NCPP=3)

Results Open Hanabi (4, 5)



Results Open Hanabi (3, 3)

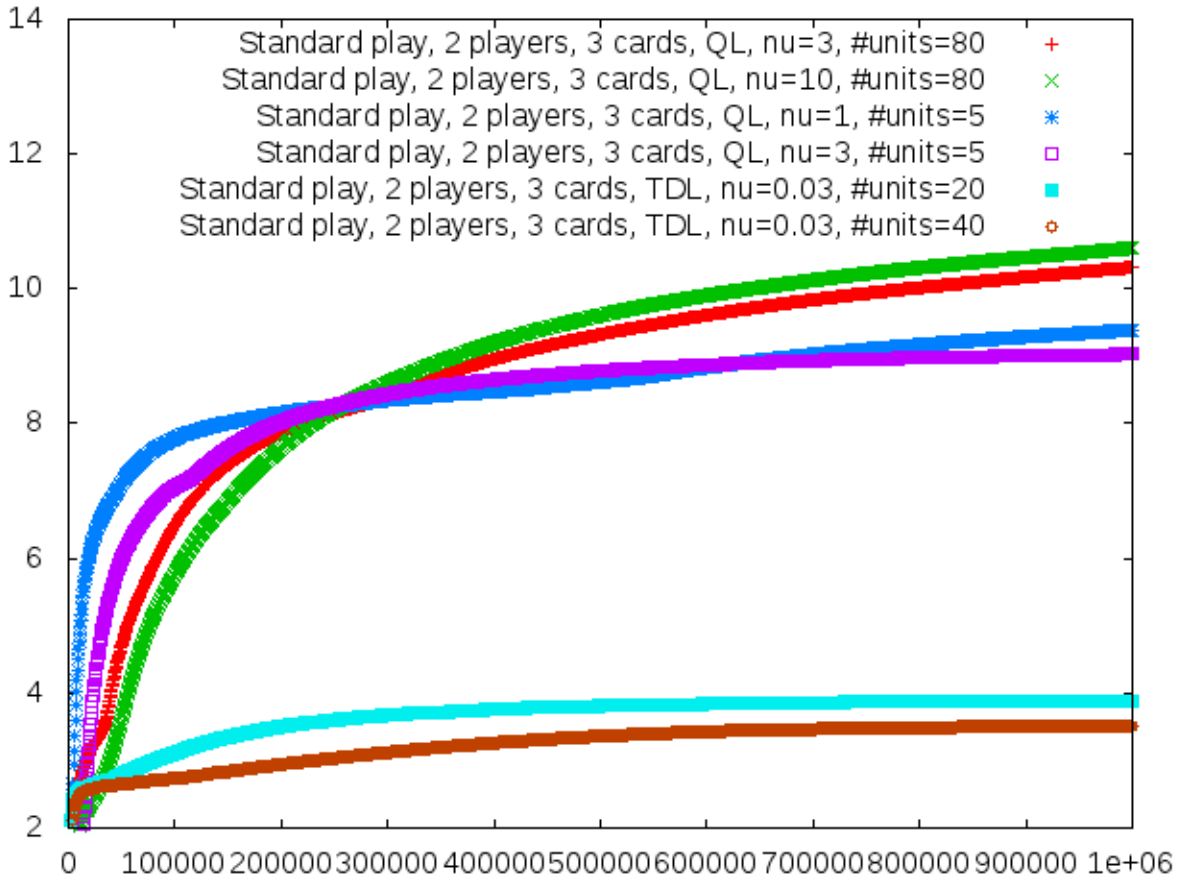


Results on Open Hanabi

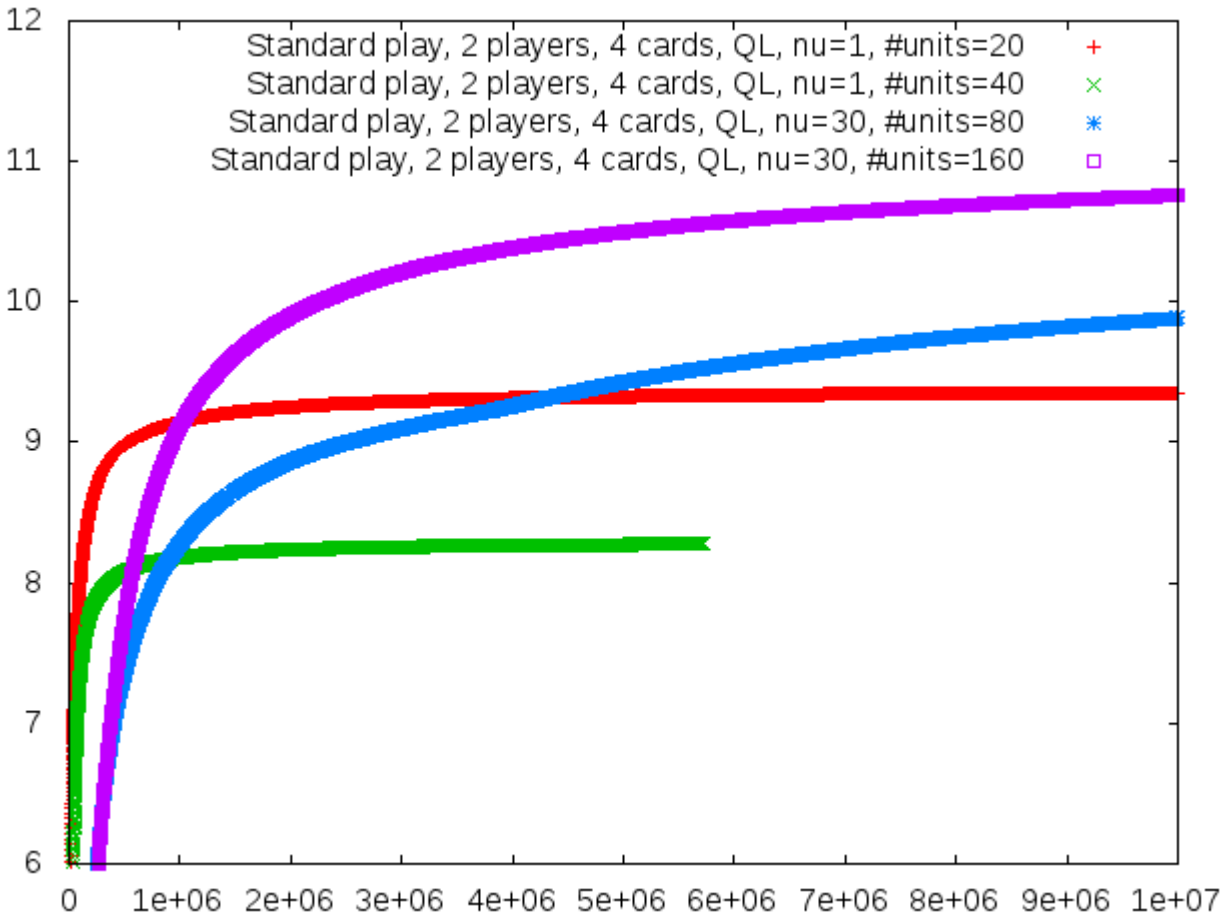
- NP in {2, 3, 4, 5} and NCPP in {3, 4, 5}
- Neural network (average scores +- 0.1 in [19, 24])
- Simple knowl.-based player (av. scores +-0.01 in [20.4, 24.4])

NP \ NCPP	3	4	5
2	19.3 20.49	21.0 22.91	22.3 24.12
3	21.1 22.08	22.1 23.82	23.4 24.36
4	21.2 22.75	22.8 23.85	23.3 24.03
5	21.6 22.82	22.9 23.42	23.1 * 22.92

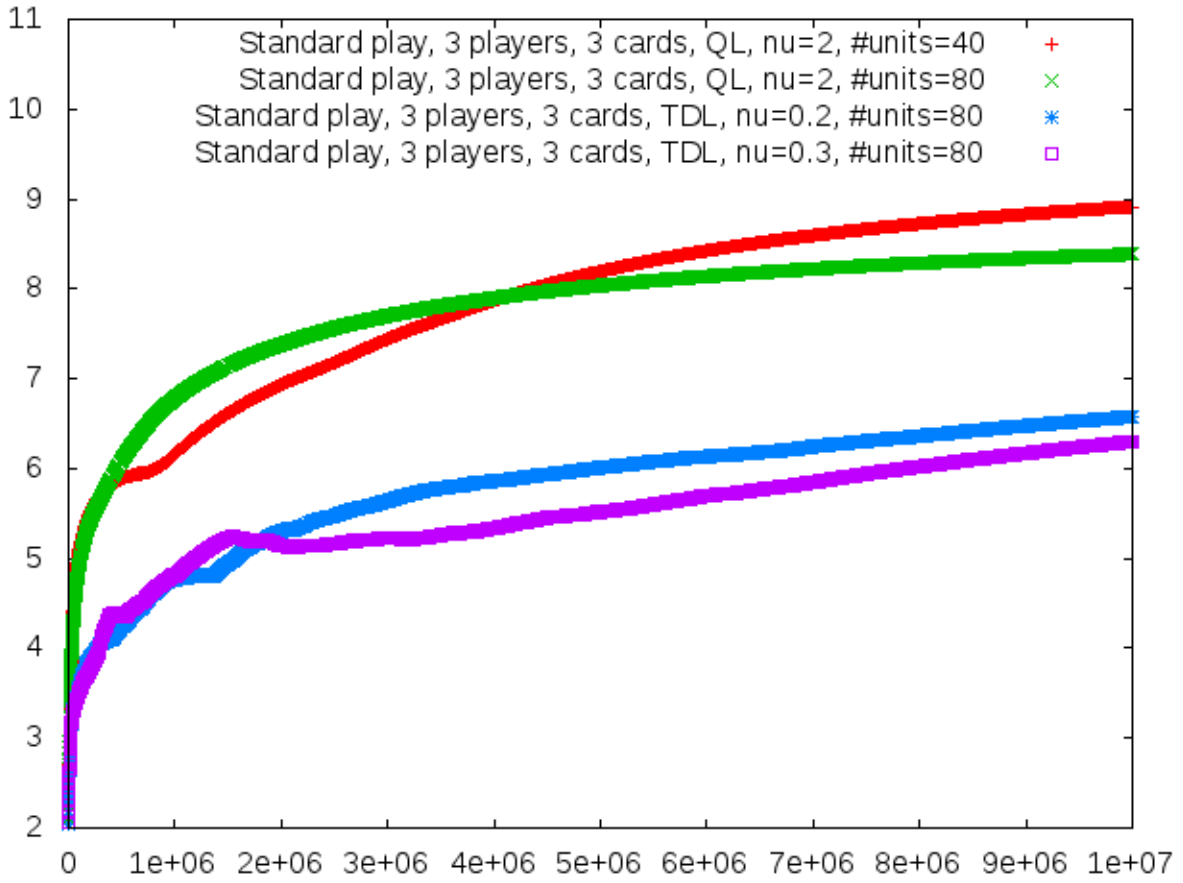
Results Standard Hanabi (2, 3)



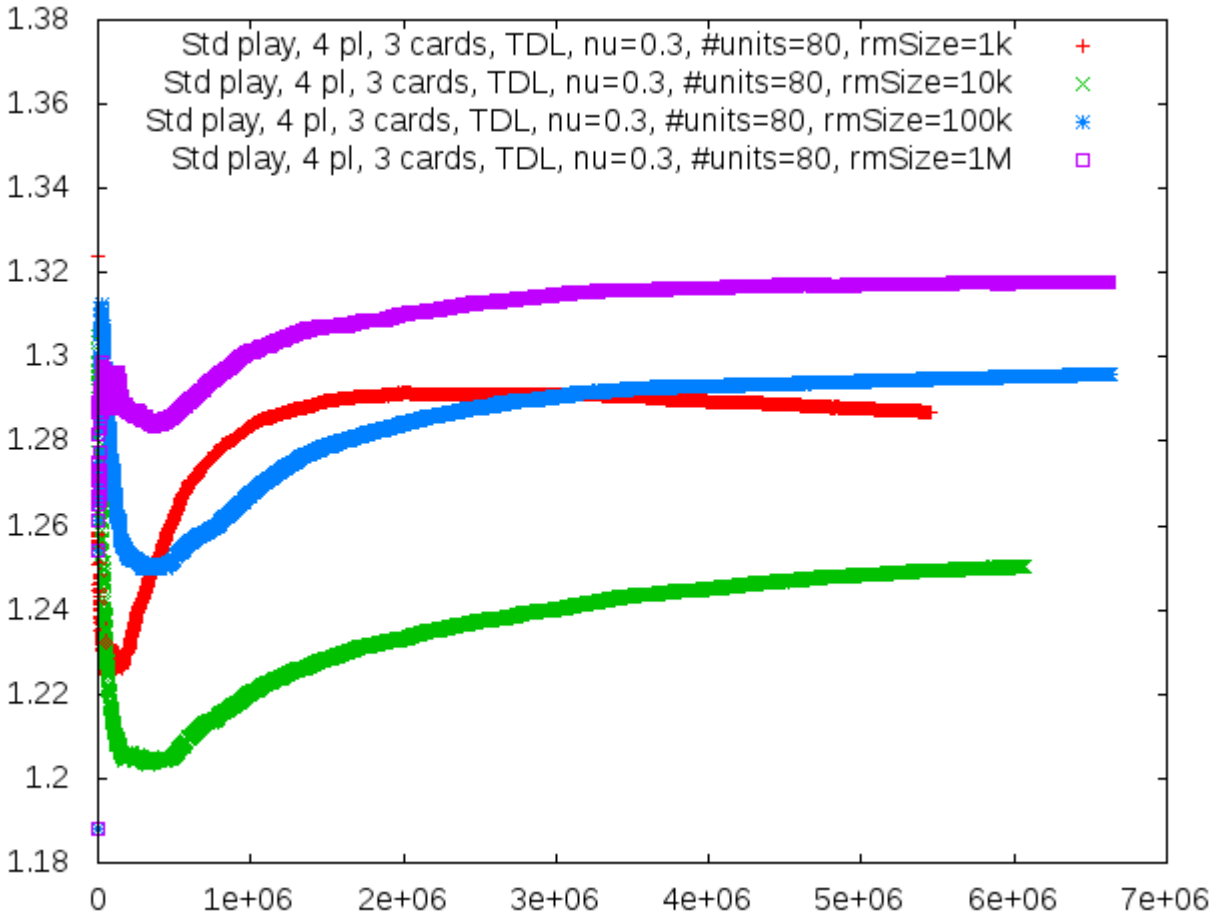
Results Standard Hanabi (2, 4)



Results Standard Hanabi (3, 3)



Results Standard Hanabi (4, 3)



Results on Standard Hanabi

- NP in {2, 3, 4} and NCPP in {3, 4}
- Average scores obtained by our neural network
  - Average score (QL or TDL ?, NUPL, NU)
- The range [9, 13] corresponds to the certainty player scores

NP \ NCPP	3	4
2	Learn : 12.3 (QL, 80, 10) Test : 13.2 (QL, 80, 10)	Learn : 10.8 (QL, 160, 30) Test : 11.9 (QL, 160, 30)
3	Learn : 8.90 (QL, 40 3) Test : 12.6 (TDL, 80 0.3)	
4	Learn : 1.5 Test : __	

Qualitative Analysis

- Open Hanabi
  - Quite easy : the average score is « good » (near 23 or 24)
  - Not perfect : inferior to the hat score.
- Standard Hanabi
  - Playing level similar to the certainty player level
  - Various stages of learning :

    1° Learn that a « playing move » is a good move (score += 1)
    - Average score up to 3 :

    2° Learn the negative effect of red tokens and delay « playing moves » (!?).
    - Average score up to S (S=6, 7 up to 12 or 13)

    3° Learn some tactics
    - Average score greater than 15 or 20 : not observed in our study

    4° Learn a convention
    - Average score approaching 25 : out of the scope of our study

The challenge

- How to learn a given convention (with a teacher) ?
  - Imitation of the confidence player ?
  - Imitation of the hat player ?
- How to uncover a convention (in self-play) ?
  - the confidence convention
  - the hat convention
  - a novel convention

Learning a convention

- Why is it hard ?
  - The convention defines the transition probability function from state-action to next state.
  - Within the MDP formalism, this function is given by the environment
  - Here, it has to be learnt ==> Go beyond MDP ?
- TDL or QL ?
  - TDL + explicit depth-one policy that could use the convention
    - 2 networks : value network + convention network
  - QL the convention should be learnt implicitly with the action values
    - 1 action value network
- Multi-agent RL problem
  - One network per player

Next : (Deep) learning ?

- (Deep) Learning techniques to learn better
  - Rectifier Linear Unit (ReLU) rather than a sigmoid
    - ReLU : f(x) = log(1 + exp(x)). (Nair & Hinton 2010)
  - Residual learning
    - Connect the previous layer of the previous layer to the current layer (He & al 2017).
  - Batch Normalization
    - (Ioffe & Szegedy 2015)
  - Asynchronous Methods
    - (Minh & al 2016)
  - Double Q learning
    - (Van Hasselt 2010)
  - Prioritized Experience Replay
    - (Wang & al 2016)
  - Rainbow
    - (Hessel & al 2018)
- Deep Learning + Novel architecture
  - To learn a Hanabi convention
  - To be found :-)


1
5
4
2
3
3

Conclusions and future work

- Conclusions
  - Learning Hanabi in self-play : hard task !
    - Testing the shallow RL approach
    - Preliminary Results for NP=2 or 3 and NCPP=3 and 4
    - Current limit : NP=4
  - Hanabi is a partially observed domain, incomplete information game

- Future work :
  - Deep RL approach :
    - Extend the current results to greater values of NP and NCPP
    - Learn a given convention
  - Deep RL + novel idea
    - Learn a novel convention in self-play
  - Surpass the hat derived players
  - Focus on incomplete information games
  - Solve Bridge and Poker !

Thank you for your attention!

Questions ?

bruno.bouzy@parisdescartes.fr