Deep Learning in Go - Overview

Josef Moudřík

sui @ 13.4.2017

j.moudrik@gmail.com

Overview

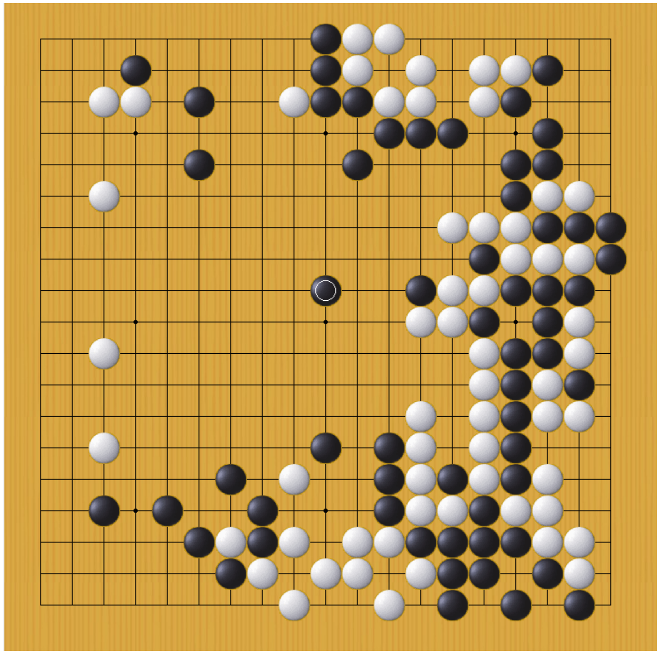
- Go
- AI v Go
- DL v AI v Go
- JM v DL v AI v Go

Alpha Go 2016

Go

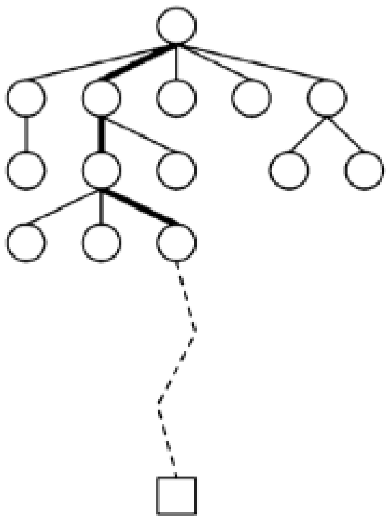
- ~ Nejstarší hra na světě

  => hodně záznamů her
- Hrací deska 19 × 19
- černé a bílé kameny
- jednoduchá pravidla
- kameny se nehýbou
- komplikované pozice
- tahy mají dalekosáhlé globální důsledky



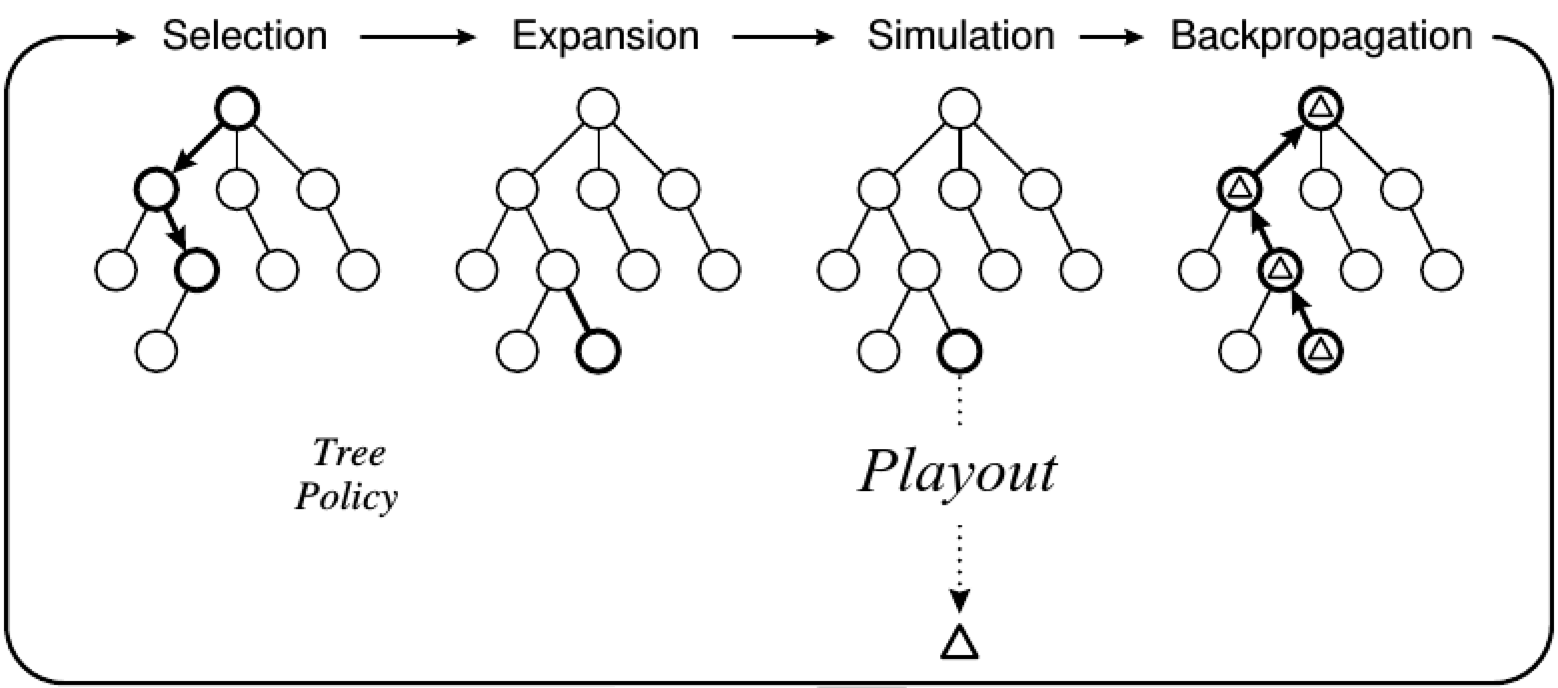
AI v Go

- velký větvící faktor (#dalších tahů ~250)
- hluboký strom (|hra| ~ 150 tahů)
- není jasná heuristika evaluace pozic (vs. šachy)
- 3 období:
  - gofai - rule-based, domain knowledge ručně (~10kyu)
  - MCTS - tree-search + playouts (~5dan)
  - DL + MCTS - (~ ???)



MCTS

- Heuristické stromové prohledávání

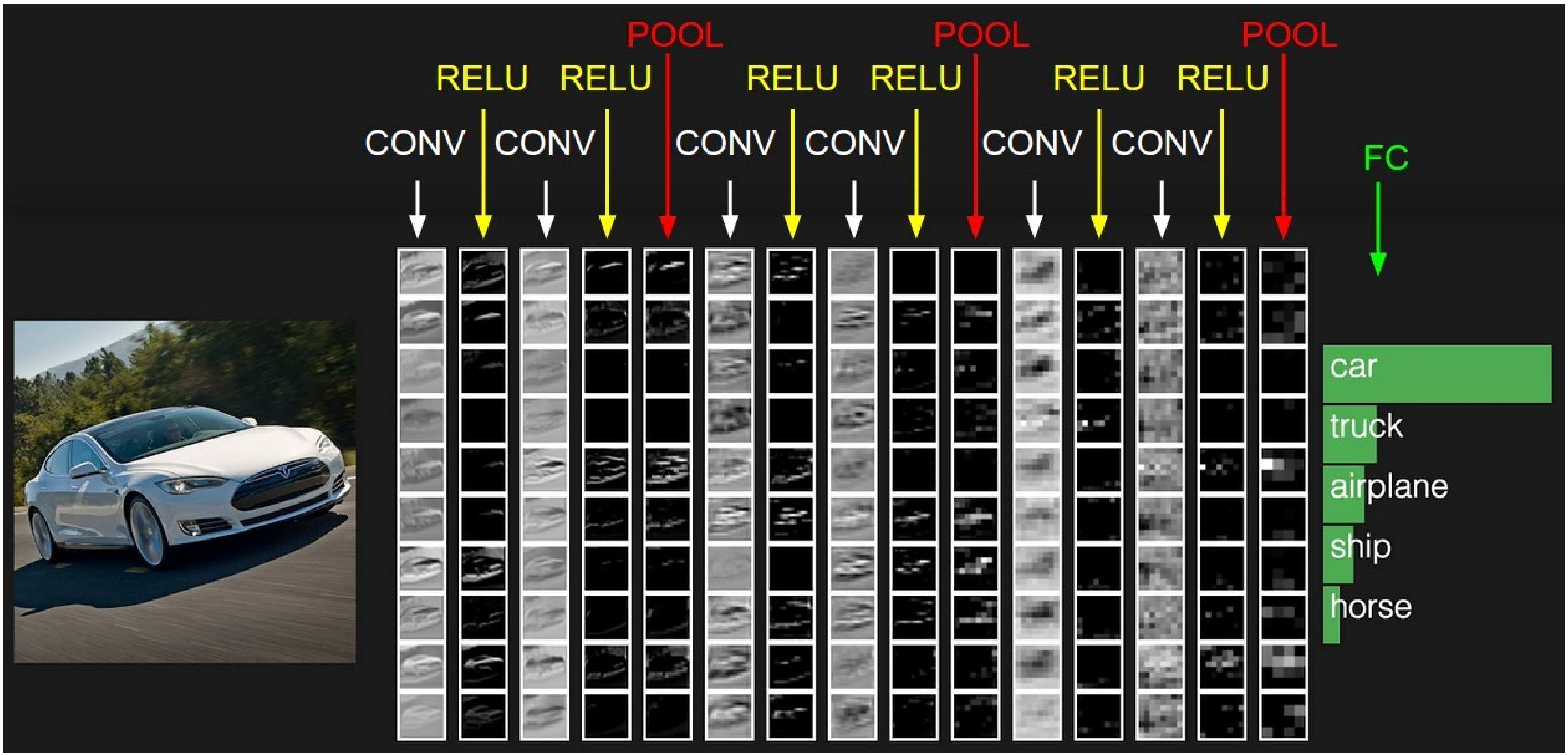


- *Tree Policy* $\mathrm{UCT} = \frac{w_i}{n_i} + c\sqrt{\frac{\ln t}{n_i}}$
- *Playout*
- v praxi těžké: domain knowledge, optimalizace parametrů,...

Deep Learning

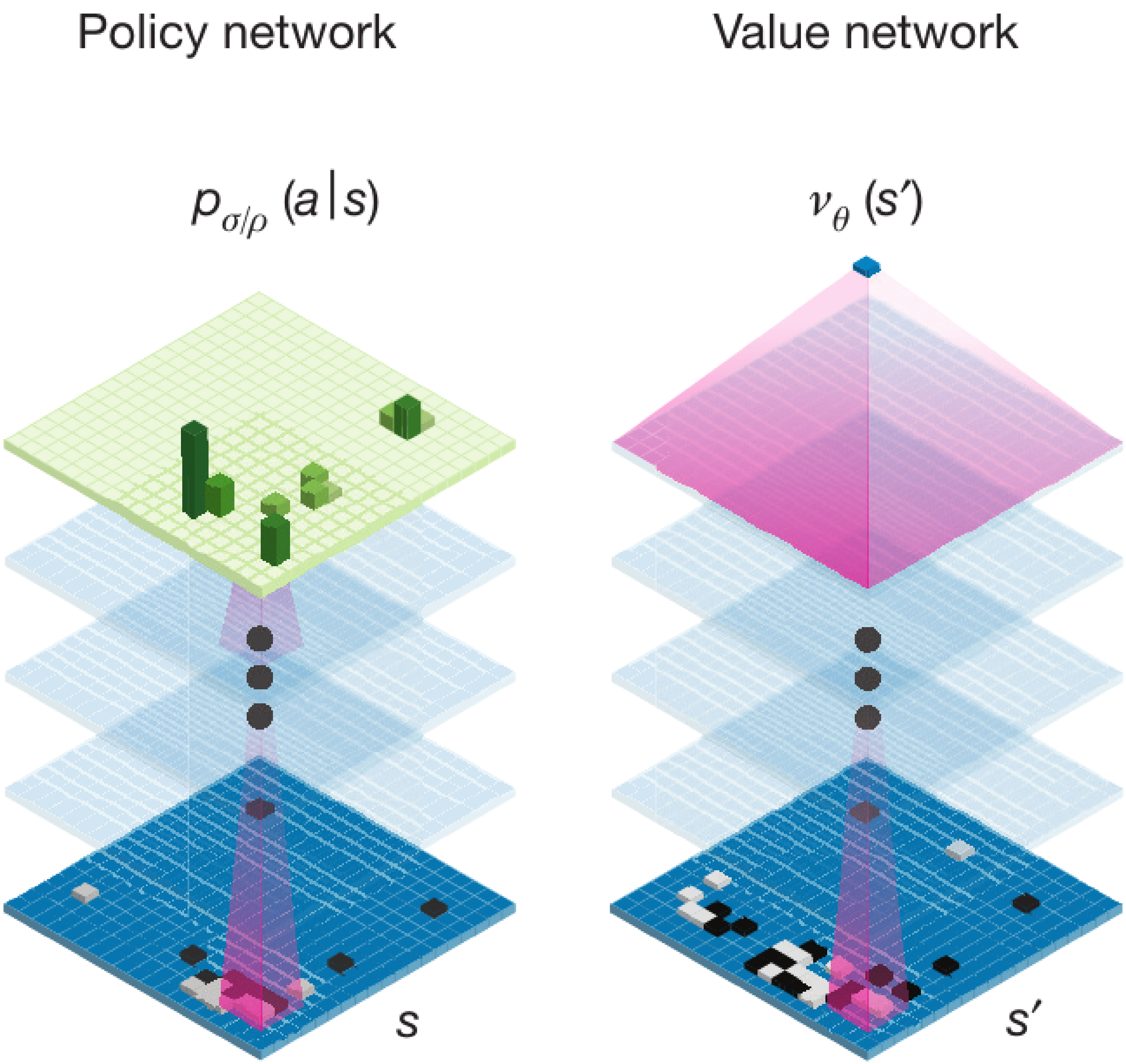
- Deep Learning $\overset{\text{IMHO}}{==}$ učení reprezentací
- **Goal**:
  - modely, které mají dobré (sémantické) reprezentace
- **Means**:
  - hluboké modely s mnoha stupni volnosti
  - hodně dat
  - chytré učící algoritmy
  - GPU / TPU

Konvoluční sítě



DL + MCTS + scale == Alpha Go

- Policy net:
  - G: příští tah
  - 13 vrstev
  - 57% acc (~3d)!!
- Value net:
  - G: win / loss



DL + MCTS + scale == Alpha Go

1) SL Policy net (logloss)

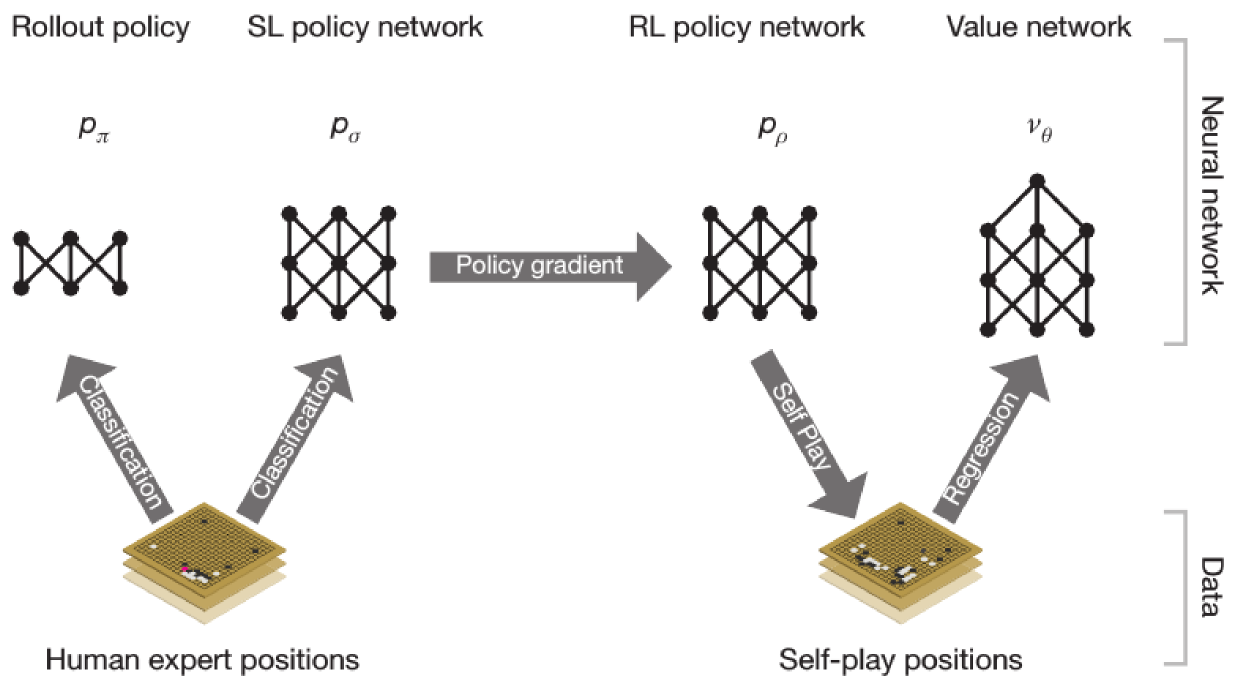
2) RL Policy net, self-play (logloss)

3) SL Value net, self-play (logloss)

$$\Delta\sigma \propto \frac{\partial \log p_\sigma(a|s)}{\partial \sigma}$$

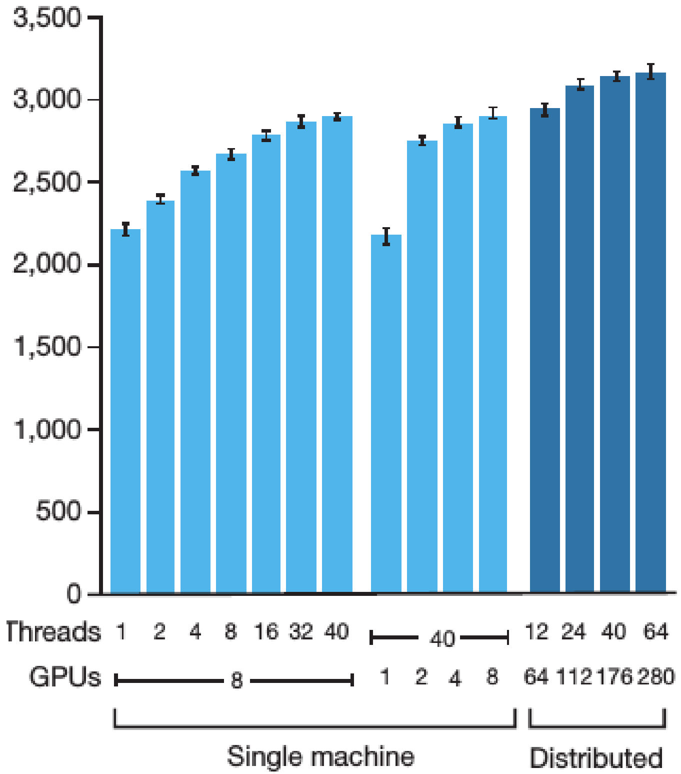
$$\Delta\rho \propto \frac{\partial \log p_\rho(a_t|s_t)}{\partial \rho} z_t$$

$$\Delta\theta \propto \frac{\partial v_\theta(s)}{\partial \theta}(z - v_\theta(s))$$



DL + MCTS + scale == Alpha Go

- Dohromady:
  - Policy net – šířka stromu
  - Value net (+ playouts) – hloubka stromu
  - VELKÝ cluster pro učení
    - RL ~ 30 000 000 self-play her
  - turnaj – 1202 CPU, 176 GPU



Determining Player Skill in the Game of Go with Deep Neural Networks

Josef Moudřík[1] Roman Neruda[2]

[1]Charles University in Prague
Faculty of Mathematics and Physics
J.Moudrik@gmail.com

[2]Institute of Computer Science
Academy of Sciences of the Czech Republic
roman@cs.cas.cz

TPNC 2016
DOI: 10.1007/978-3-319-49001-4_15

Presentation Outline

- Introduction: Go, Computer Go, Deep Learning
- Motivation
- Dataset
- Augmentation & Downsampling
- Model Architecture
- Experiments
- Conclusions

- One of the oldest games.
- 2 players, perfect information, deterministic rules.
- Board size of 19×19 intersections.
- **Goal**: control the board

  — enclose territory, capture enemy.

Introduction: Computer Go

- **Go AI is hard:**
- high branching factor,
- no clear evaluation function.

- **Recently solved by Google AlphaGo,**
- a combination of Monte Carlo Tree Search with **deep learning**. [Silver et al., 2016]

Introduction: Deep Learning

- Differentiable neural network models,
- large number of parameters,
- deep — error is back-propagated through many steps.

- **Convolutional Neural Networks:**
- hierarchical model based on learning convolutional kernels,
- great for data with spatial structure — e.g. images, sound spectrograms, Go boards.
- Learns increasingly abstract hierarchical representations.

Introduction: Motivation

- Strength of Go players is measured by rating:
  - a numerical quantity — rating — is assigned to each player,
  - updated after each game, using win/loss information.
  - Rating is used to e.g. pair opponents with similar strength.
- Rating converges slowly for new players, causing problems such as badly matched opponents and rating deflation.
- Can we use more information (than the win/loss bit) from each game?

Introduction: Motivation

- Strength of Go players is measured by rating:
  - a numerical quantity — rating — is assigned to each player,
  - updated after each game, using win/loss information.
  - Rating is used to e.g. pair opponents with similar strength.
- Rating converges slowly for new players, causing problems such as badly matched opponents and rating deflation.
- Can we use more information (than the win/loss bit) from each game?
- **Maybe the game record itself?!**

Introduction: Motivation

- Strength of Go players is measured by rating:
  - a numerical quantity — rating — is assigned to each player,
  - updated after each game, using win/loss information.
  - Rating is used to e.g. pair opponents with similar strength.
- Rating converges slowly for new players, causing problems such as badly matched opponents and rating deflation.
- Can we use more information (than the win/loss bit) from each game?
- **Maybe the game record itself?!**
- **Our Work:** Use Deep Learning to predict player's strength from a board position, aiming to improve convergence of rating systems.

Dataset

- 188,700 Games from Online Go Server (OGS).
- this makes for 3,426,489 pairs (X, y), where
  - y is one of 3 classes based on strength, $y \in \{\text{strong}, \text{intermediate}, \text{beginner}\}$
  - X is encoding of position and last 4 moves, represented as a volume of size $13 \times 19 \times 19$:
    - 4 planes of liberties of current player,
    - 4 planes of liberties of opponent,
    - 1 plane for empty intersections,
    - 4 planes marking the last 4 moves.

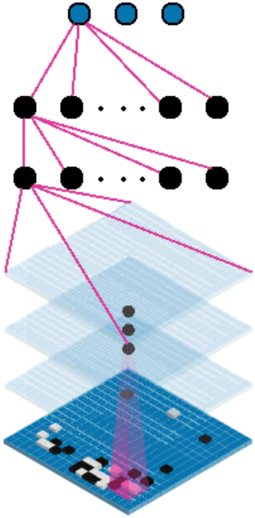
Augmentation & Downsampling

- Techniques to reduce over-fitting and improve generalization.
- **Sub-sampling:** on average, take every 5th position from each game (uniformly randomly).
- **Augmentation:** each sample is randomly transformed into 1 of its 8 symmetries during training.
- **Equalization:** y classes are equally represented in the training set (throwaway superfluous examples).

Model Architecture

- Input layer,
- 1 Convolutional layer of 512 filters of size 5×5,
- 3 Convolutional layer of 128 filters of size 3×3,
- 2 fully connected layers of 128 neurons,
- Output layer, 3-way Softmax.

- All layers (except for the final one) have ReLU activation.
- Trained with mini-batched SGD with Nesterov momentum.



Img. adapted from [Silver et al., 2016].

Experiments and Results
Single Position

- Baseline case, accuracy 71.5%

True Label \ Predicted Label	Strong	Intermediate	Weak
Strong	0.79	0.18	0.02
Intermediate	0.21	0.63	0.15
Weak	0.03	0.19	0.78

Confusion Matrix

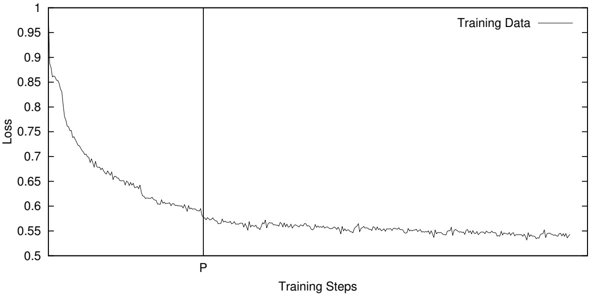


Figure: Training Loss Evolution

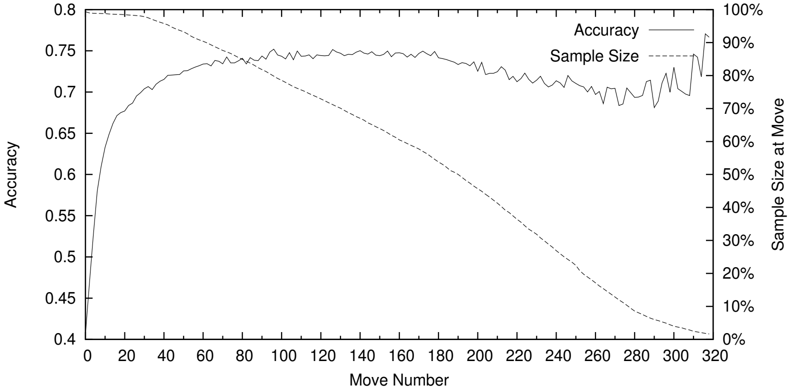


Figure: Dependency of accuracy and sample size on move number.

Table: Summary of results. **A**ugmentation (ensemble of 8 symmetries), **C**ropped (skip first 30 moves), **W**eighted (proportionaly to avg. Acc. for given move).

Model	Acc.	Acc. (Top-2)
Single Position	71.5 %	94.6%
Single Position (**A**)	72.5 %	94.9%
Aggregated per Game, mode (**A**)	76.8 %	N/A
Aggregated per Game, sum (**A**)	77.1 %	96.4%
Aggregated per Game, sum (**A**, **C**)	77.7 %	96.7%
Aggregated per Game, sum (**A**, **W**)	77.9 %	96.8%

Conclusions

- We have used Deep Learning to predict player's strength from a single game position (= little information).
- The method is applicable to whole games by aggregating individual predictions.
- Works nicely for 3 target classes, more data would be good to move towards accurate regression.
- Will be experimentally deployed on Online Go Server (hopefuly) soon.

References I

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489.