Neural Networks in Go

Markus Enzenberger

Go Seminar, University of Alberta

September 2002

`~emarkus/web_docs/nngo/nngo-2up.ps`

Motivation

- Evaluation in Go is more difficult than in other games
- The evaluation has many components that are hard to define, like shape or aji (latent threats or possibilities)
- Humans are good at intuitive pattern recognition

Neural networks for subtasks

- Local move prediction (van der Werf, 2002; Dahl, 1999)
- Abstract input
- Simple network architectures with few weights
- Offline training from expert games

Neural network as the central part

- Network does the board evaluation
- Large number of weights
- Reinforcement learning by self-play and Temporal Differences

Network architecture

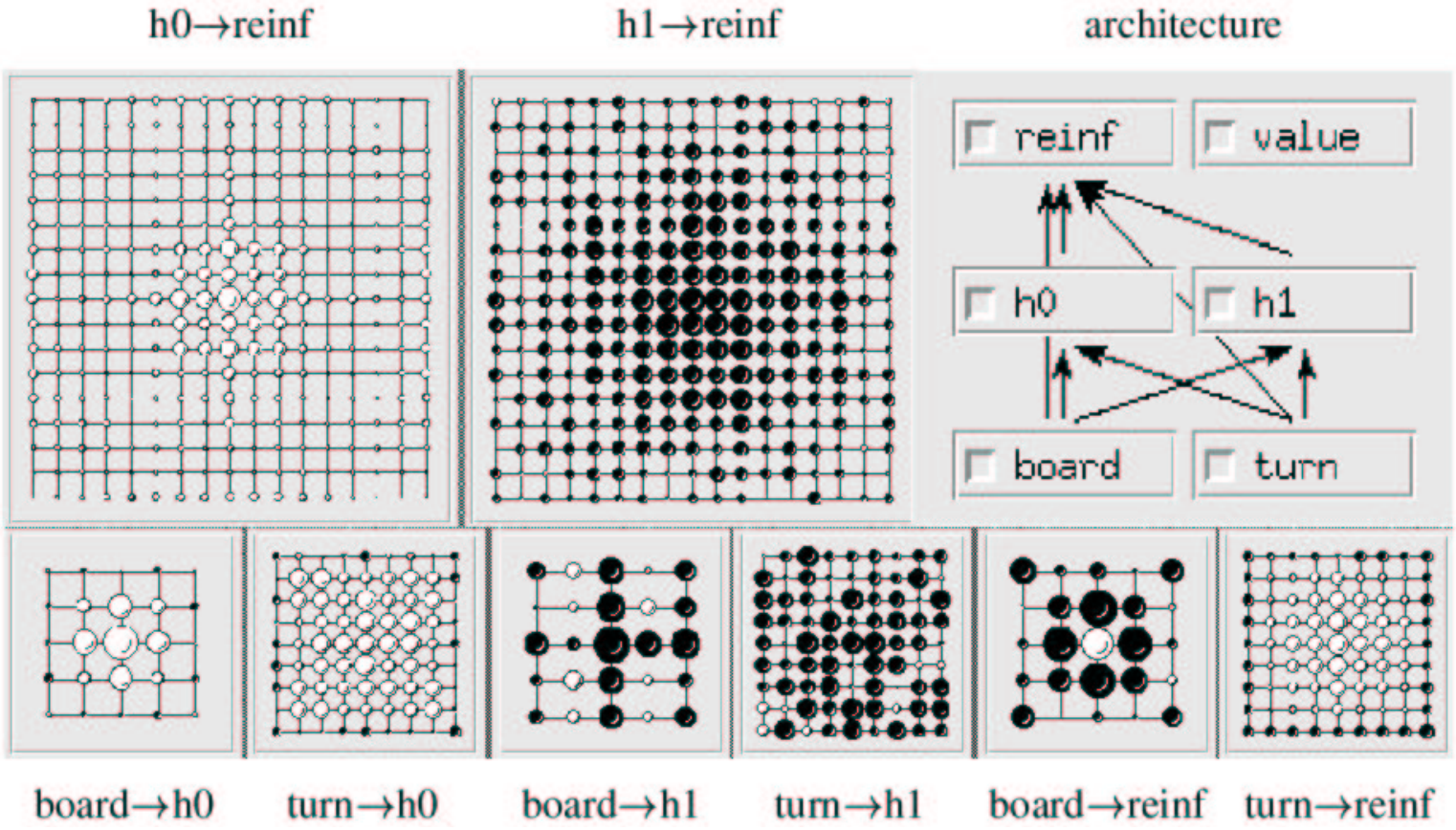
How to make training of large scale neural networks feasible?

- Highly structured neural networks
- Sparse connectivity
- Reflecting invariances by weight sharing
- Enriched (local) reinforcement signal
- Preprocessing position to get local features

Schraudolph, Dayan, Sejnowski 1994

- Raw board position as input
- 5x5 receptive fields for projection from input to hidden layer
- Full connectivity from hidden layer to output

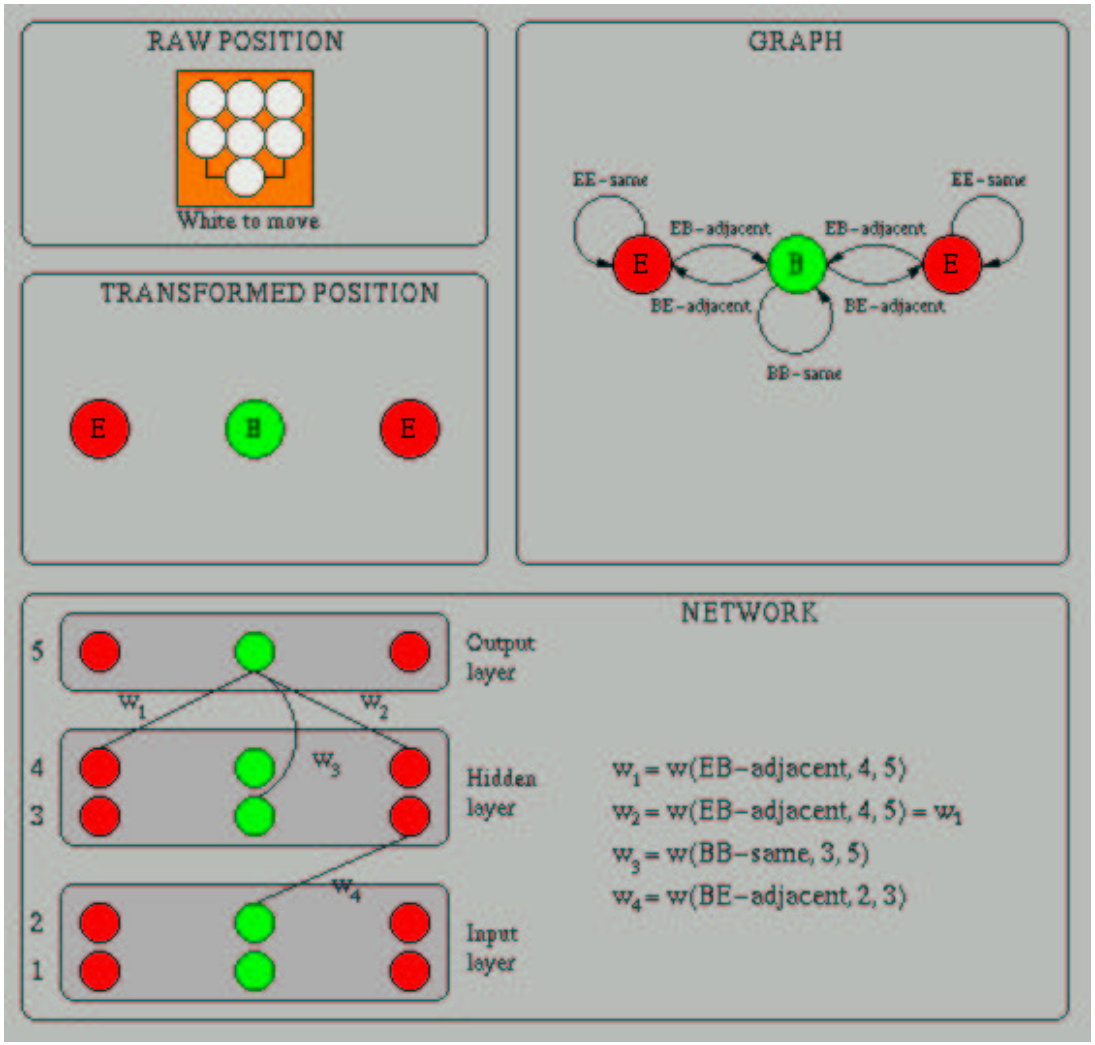
Schraudolph, Dayan, Sejnowski 1994 (cont'd)



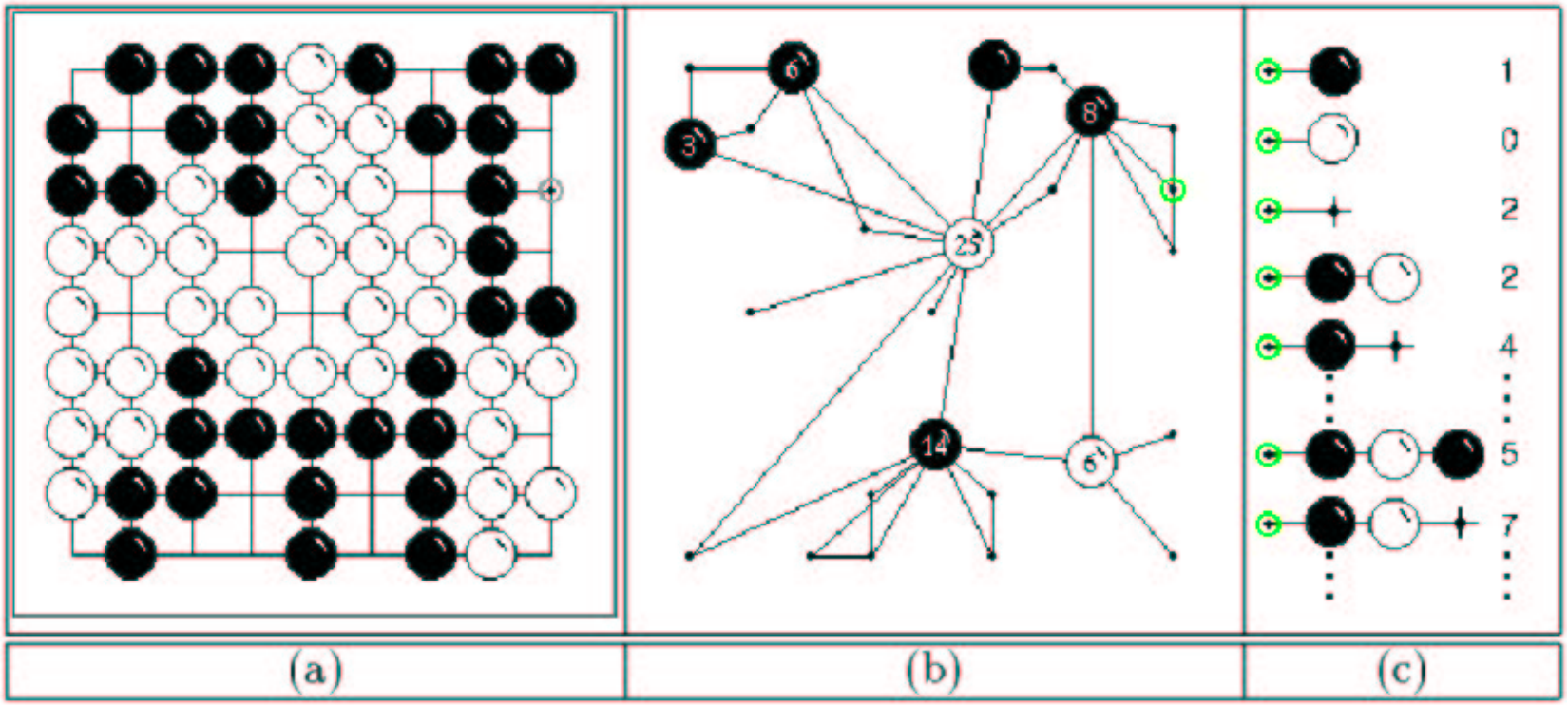
NeuroGo 2

- Translates position into graph of blocks and empty points
- Edges correspond to features that are given as additional input to the network
- Dynamic neural network architecture based on this graph
- Maps a large class of invariances to equal representations, but some exceptions (mapping is not isomorphic) must be compensated with additional input features
- External experts for local outcome prediction can override network output and are used as early reinforcment signal

NeuroGo 2


RAW POSITION
White to move
TRANSFORMED POSITION
E
B
E
GRAPH
EE - same
EE - same
EB - adjacent
EB - adjacent
E
B
E
BE - adjacent
BE - adjacent
BB - same
NETWORK
5
Output layer
w_1
w_2
4
3
w_3
Hidden layer
w_4
2
1
Input layer
$w_1 = w(EB-adjacent, 4, 5)$
$w_2 = w(EB-adjacent, 4, 5) = w_1$
$w_3 = w(BB-same, 3, 5)$
$w_4 = w(BE-adjacent, 2, 3)$

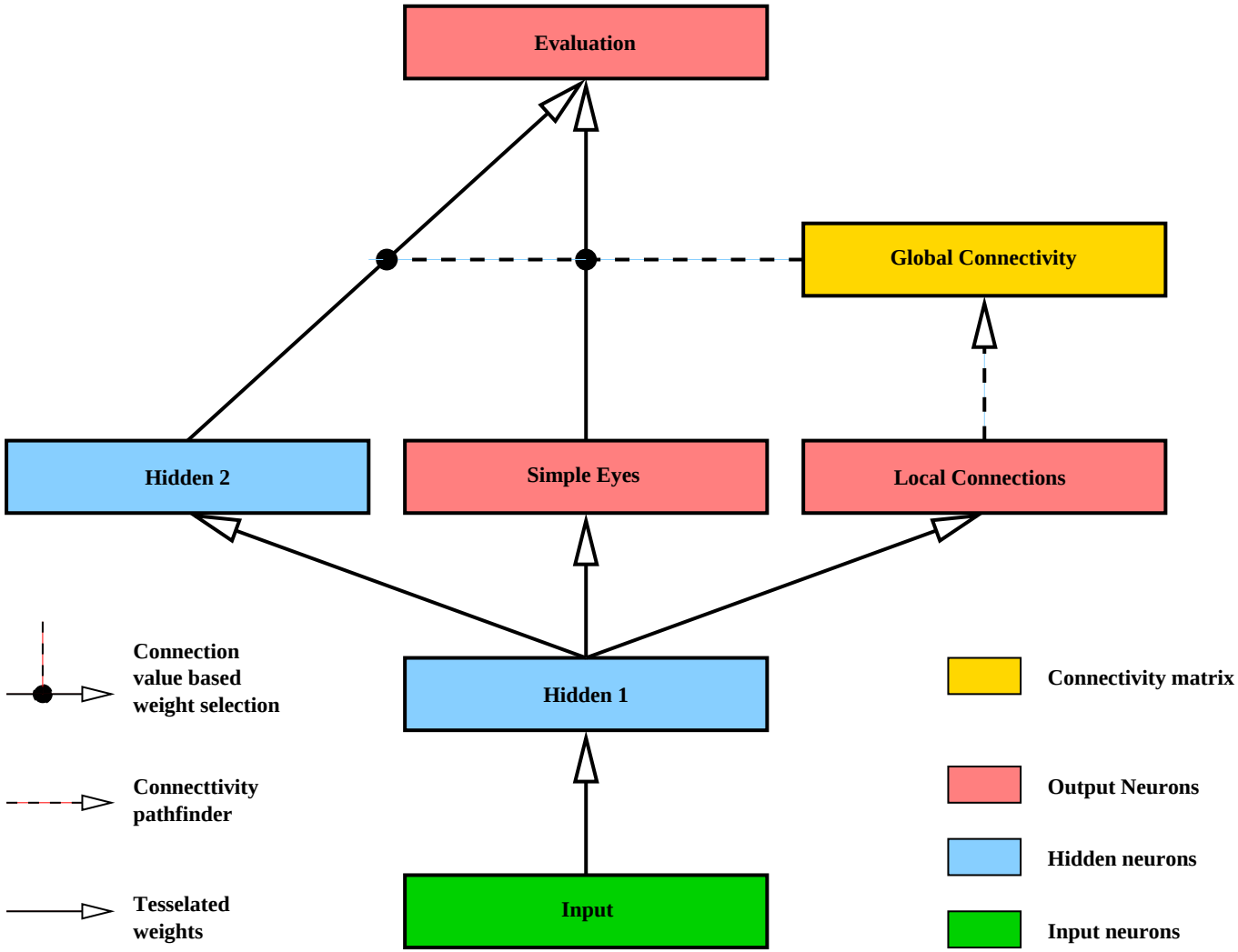
Graepel, Goutrie, Krüger 2000



Graepel, Goutrie, Krüger 2000 (cont'd)

- Preprocessing based on graph representation similar to NeuroGo 2
- Subgraph features as input
- No results published yet on using this network for board evaluation (only move prediction)

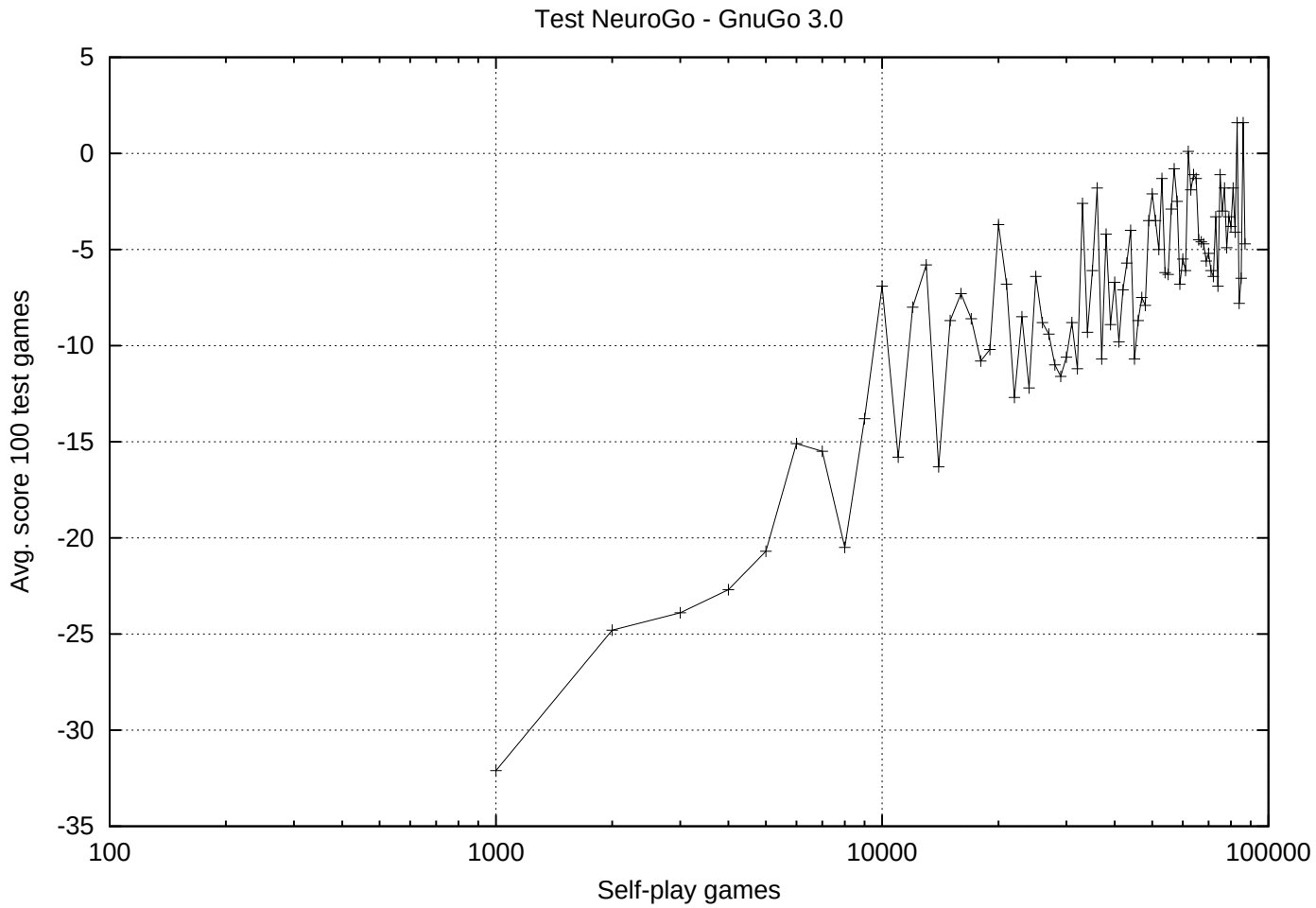
NeuroGo - The Next Generation



NeuroGo - The Next Generation (cont'd)

- No longer uses graph / dynamic network architecture
- Local connections and one-point eyes as additional output and reinforcement signal
- Pathfinding algorithm calculates global soft segmentation of the board
- Segmentation is used in projection to evaluation layer
- More high level features in input compared with NeuroGo 2 (Possible Omission Number estimation, some life and death shapes)

Training on 9x9



Resources

Schraudolph, Dayan, Sejnowski: Learning to Evaluate Go Positions via Temporal Difference Methods.
`http://www.inf.ethz.ch/~schraudo/pubs/`

Enzenberger: The integration of a priori knowledge into a Go playing neural network.
`http://www.markus-enzenberger.de/neurogo.html`

Dahl: Honte, a Go-playing program using neural nets.
`http://www.ai.univie.ac.at/icml-99-ws-games/`

Graepel et al.: Learning on Graphs in the Game of Go.
`http://stat.cs.tu-berlin.de/~guru/publications.html`

van der Werf et al.: Local move prediction in Go.
`http://www.cs.rulimburg.nl/~vanderwerf/publications.html`