
Neuronales Lernen in Go

Kai Brodmann



TU Darmstadt
FB Informatik – FG Knowledge Engineering

Seminar: Knowledge Engineering und Lernen in Spielen

Prof. J. Fürnkranz



Neuronales Lernen in Go

- **Hohe Komplexität von Go**
- **Trotz gesteigerter Bemühungen spielen Computer Go nur auf relativ niedrigem Level**
- **Gründe**
 - Hoher Verzweigungsgrad des Spielbaumes
 - Musterorientierung nötig
 - Viele interagierende Ziele



Neuronales Lernen in Go

- **Eröffnungsspiel**
- **Mittelspiel**
- **Endspiel**



Neuronales Lernen in Go

- **Hauptproblem des Computers bisher – betrachtet Spielsituation als Knoten des Spielbaumes, benutzt keine Mustererkennung**
- **Neuronale Netze sehr geeignet für Mustererkennung**
- **Problem: erwünschte Ausgabe einzelner Züge, nur das Endergebnis des Spieles ist bekannt**
- **Neuro-Evolutionsmethoden**
 - Anstatt einzelne Züge zu bewerten werden Netze bewertet, ausgesucht und rekombiniert auf Grund Ihrer Gesamt-Performanz beim Spielen



Neuronales Lernen in Go

- **SANE (Symbiotic Adaptive Neuro-Evolution)**
- **Ansatz: nicht jedes Individuum in der Population stellt ein Neuronales Netz dar, sondern es existieren zwei Populationen**
 - Population von Neuronen
 - Population von Netzplänen
- **Suchraum wird aufgespalten, sobald geeignete Entscheidungsneuronen gefunden sind, wird nur noch über deren Art des Zusammenbaus entschieden**
- **Netze können zufällig aus der Neuronen Population aufgebaut werden**



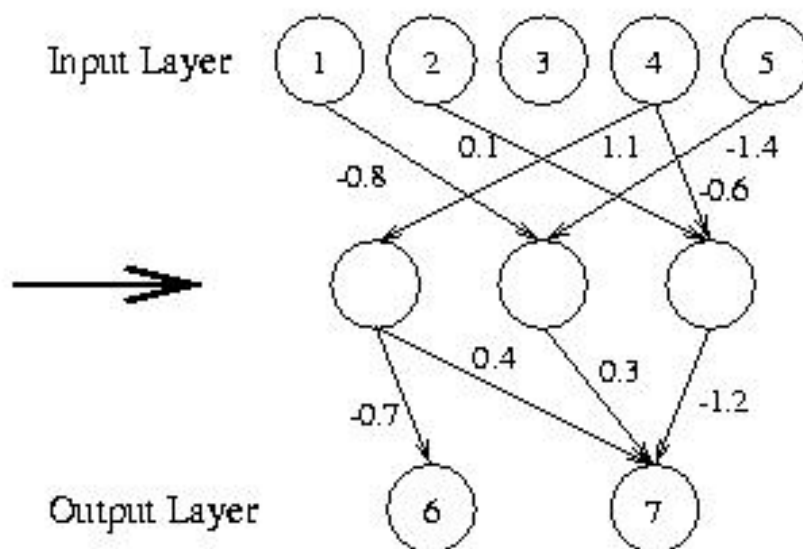
Neuronales Lernen in Go

- **Effektiver ist Netzpläne in einer zweiten evolutionären Entwicklung zu produzieren**
- **Selektion, Rekombination (1 Punkt Crossover), Eliteparameter**
- **Mutation auf den Neuronen sehr gering (1 %)**
- **Mutation auf den Netzplänen**
 - 1 % der Zeiger auf Neuronen werden zufällig vertauscht
 - Die Zeiger auf Neuronen mit Nachkommen werden mit 50% Wahrscheinlichkeit auf deren Nachkommen umgesetzt



Neuronales Lernen in Go

Label Weight					
4	0.1	6	-0.7	7	0.4
1	-0.8	5	-1.4	7	0.3
2	0.1	4	-0.6	7	-1.2

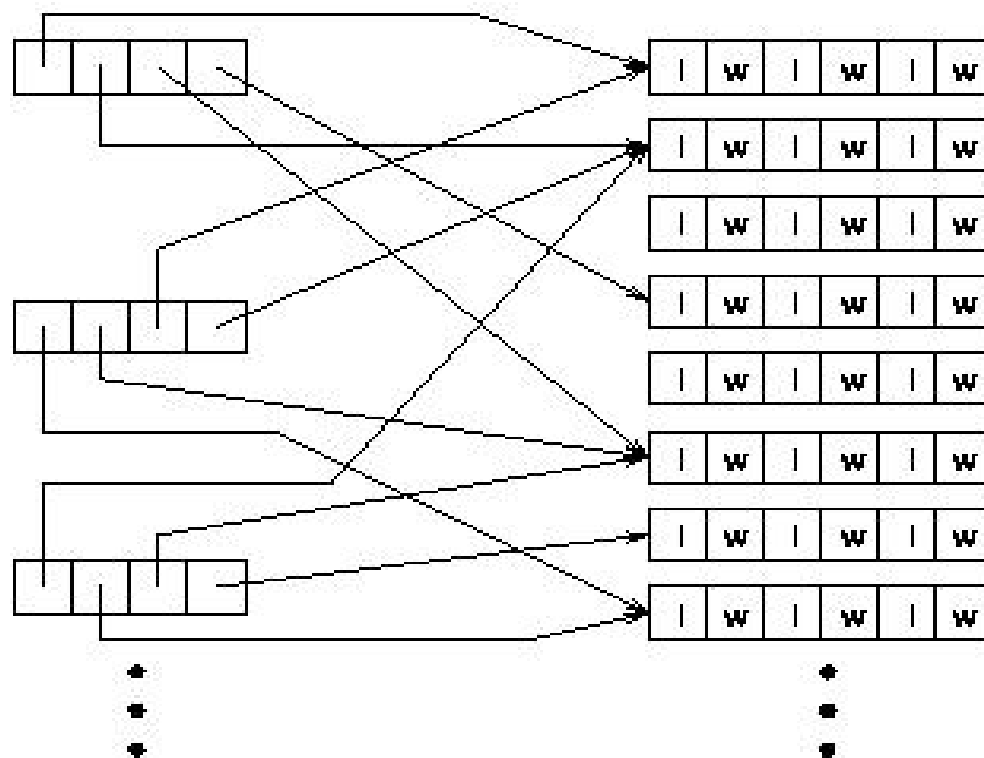




Neuronales Lernen in Go

Network Blueprint Population

Neuron Population





Neuronales Lernen in Go

```
for each neuron  $n$  in population  $P_n$ 
   $n.fitness \leftarrow 0$ 
   $n.participation \leftarrow 0$ 
for each blueprint  $b$  in population  $P_b$ 
   $neuralnet \leftarrow \mathbf{decode}(b)$ 
   $b.fitness \leftarrow \mathbf{task}(neuralnet)$ 
  for each neuron  $n$  in  $b$ 
     $n.fitness \leftarrow n.fitness + b.fitness$ 
     $n.participation \leftarrow n.participation + 1$ 
for each neuron  $n$  in population  $P_n$ 

   $n.fitness \leftarrow n.fitness / n.participation$ 
```



Neuronales Lernen in Go

- **SANE und Go**
 - Netzwerkparameter
 - Evolutionsparameter
 - Bewertungsfunktion
- **Dreischichtige Feed Forward Netze**
- **Feste Netzwerkarchitektur – entwickelt werden Gewichte und Verbindungen**
- **2 Eingabeneuronen und 1 Ausgabeneuron**



Neuronales Lernen in Go

- **Eingabeneuronen – weisser/schwarzer Stein vorhanden**
- **Ausgabeneuron – Fließkommazahl mit Vorzeichen**
- **Positiver Wert bedeutet guter Zug, negativer Wert schlechter Zug**
- **Neuronen 312 bits lang – repräsentieren 12 Gewichte von der Eingangsschicht zur verdeckten Schicht oder von der verdeckten Schicht zur Ausgabeschicht**
- **In jeder Generation werden 200 Netzwerke gebaut**



Neuronales Lernen in Go

- **Mutationsrate 0.1 %**
- **Rekombinationsoperator :**
1-Punkt-Crossover zwischen
Neuronen oder Netzplänen
- **25-30 % der Population dürfen**
Nachkommen erzeugen

Board size	Neuron Population	Blueprint Population	Number of neurons per network
5 × 5	2000	200	100
7 × 7	3000	200	300
9 × 9	4000	200	500



Neuronales Lernen in Go

- **Evaluationsfunktion**
- **Problem des Spielendes, es ist kein separates Ausgangsneuron für das Passen vorhanden, deshalb passt das Netzwerk nur wenn keine positiven Zugbewertungen einem legalen Zug entsprechen**
- **Obergrenze der erlaubten Züge, nur einfache KO Regel wird abgefragt**



Neuronales Lernen in Go

Ergebnisse:

- **Gegner war Wally, ein frei verfügbares Go Programm**
 - Im Quellcode verfügbar, also lässt sich der Gegner feinjustieren
 - Qualität von Wally's Spiel etwa richtig, nicht zu schwer
- **Effizienz der Evolutionsstrategie**
 - SANE baute Netzwerk welches Wally auf kleinen Spielbrettern schlug
 - Auf 5x5 brauchte SANE nur 20 Generationen
 - 7x7 50 Generationen
 - 9x9 260 Generationen
 - 13 x 13 wahrscheinlich tausende von Generationen
 - 19 x 19 zehntausende von Generationen ?



Neuronales Lernen in Go

- **Effekt des Nichtdeterminismus**
- **Deterministischer Gegner wird besiegt, doch dieses Netz generalisiert sehr schlecht auf andere Gegner, es lernt nicht wie man Go Spiel, sondern eben nur, genau die Schwächen dieses Gegners auszuloten**
- **In Wally wurde 10 % Nichtdeterminismus eingebaut**
 - Legaler Zufallszug in 10 % der Fälle



Neuronales Lernen in Go

- **Lernen ohne Vorwissen – welche Strategien entwickeln sich?**
- **Problem – Evolution passt sich genau auf den Gegner an – eine Art „Overfitting“**
- **Netzwerk erfährt nie explizit, was eine lebende Gruppe ist.**
- **Wenn ein Gegner eine Gruppe nicht töten kann, wird das Netzwerk diese Aufstellung präferieren, auch wenn ein anderer Gegner leicht damit fertig wird**
- **Es sollte gegen möglichst unterschiedliche Gegner gespielt werden**



Neuronales Lernen in Go

- **Evolutionsverlauf**
 - Anfangs nur Zufall
 - Später werden einfache lebende Gruppen gebildet
 - Bedeutung des Spielfeldrandes wird verstanden
 - Entwicklung komplexer lebender Gruppen
 - Wenn die Spielstärke des Gegners übertroffen ist, „endet“ die Evolution
- **Bekannte Go Theorien von Menschen werden auch entwickelt**
 - Hoffnung das verbesserte Neuro-Evolution tatsächlich den gewünschten Erfolg bringt



Neuronales Lernen in Go

- **Zukünftige Arbeiten**
 - Grössere Spielbretter
 - Modell entwickeln was unabhängig von der Spielbrettgrösse ist ?
 - Andere Netzwerk Architekturen
 - „preprogrammed feature detectors“
 - Hierarchischer Aufbau von Netzen (vergleiche Neuro-Netzplan)
 - Wie weit macht solch ein Aufbau Sinn ?
 - Stärkere Gegner
 - Die Methode der „Handicaps“ allein reicht nicht aus



Neuronales Lernen in Go

Evaluation in Go by a neural network using soft segmentation

- **Architektur eines Neuronalen Netzes welches Soft Segmentation einer Spielbrettposition realisiert**
- **Zwei Verbindungskarten, für jede Farbe eine**
 - Weist allen Punkten des Spielfeldes paarweise einen Wert 0-1 zu



Neuronales Lernen in Go

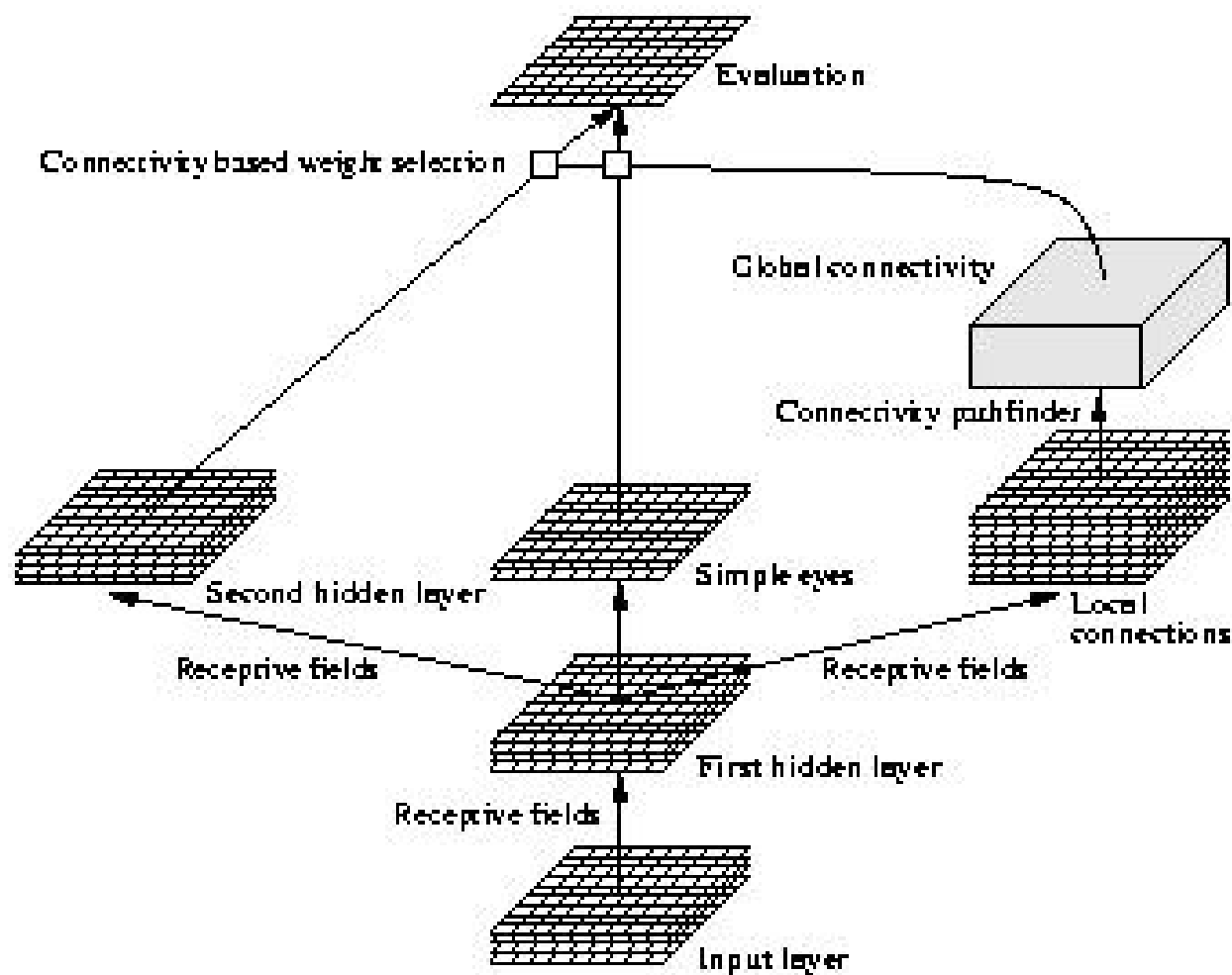


Figure 1. Network architecture.



Neuronales Lernen in Go

- **Bestärkungssignal**
 - Gruppen mit einem Auge
 - Zwei Punkte sind verbunden wenn ein Pfad zwischen ihnen existiert der nur aus Steinen dieser Farbe oder Augen besteht.
 - Ein Punkt wird als lebend bezeichnet wenn er Verbindung zu zwei Augen hat



Neuronales Lernen in Go

Die einzelnen Neuronen-Schichten in der Architektur

- **Eingabeschicht:** mindestens ein Neuron pro Spielbrettposition abhängig von der Anzahl der binären Eingabefeatures
- **Erste verdeckte Schicht:** mindestens ein Neuron pro Spielbrettposition
- **Zweite verdeckte Schicht:** wie erste verdeckte Schicht
- **Augenschicht:** 2 Neuronen, für beide Farben. Beschreibt die Wahrscheinlichkeit, das eine Farbe ein Auge an dieser Stelle aufbauen kann, Bestärkungssignal ist dann der Aufbau



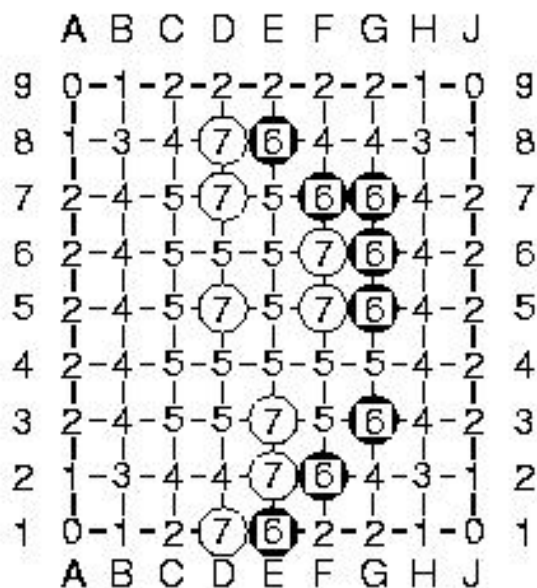
Neuronales Lernen in Go

- **Lokale Verbindungsschicht: 18 Neuronen pro Spielbrettposition, 9 für jede Farbe. Wahrscheinlichkeit von einem Punkt zu dem lokalen 3x3 Fenster um den Punkt eine Verbindung zu schaffen**
- **Globale Verbindungsschicht: Enthält $2 \cdot n^2$ Neuronen pro Spielbrettposition. Wahrscheinlichkeit dieser Position zu jeder anderen Position auf dem Spielfeld eine Verbindung zu erstellen. Baut auf dem „Pfadfinder“ der lokalen Verbindungsschicht auf**
- **Bewertungsschicht: Ein Neuron pro Spielbrettposition Aktivierung 0-1, Prognose ob Stein lebend sein wird für s/w**



Neuronales Lernen in Go

- Typen von Spielbrettpositionen, dementsprechend werden die Gewichtssets ausgesucht



$type(p)$

0	Empty corner point
1	Empty edge point next to corner
2	Other empty edge point
3	Empty point diagonal from corner
4	Other empty point on second line
5	Empty point on line 3 or higher
6	Black stone
7	White stone

Figure 2. Point types: definition and example.



Neuronales Lernen in Go

- Typen von Spielbrettpositionen
- „Receptive Fields“
- Pfadfinder der lokalen Verbindungsschicht
- Verbindungsbasierte Gewichtsselektion

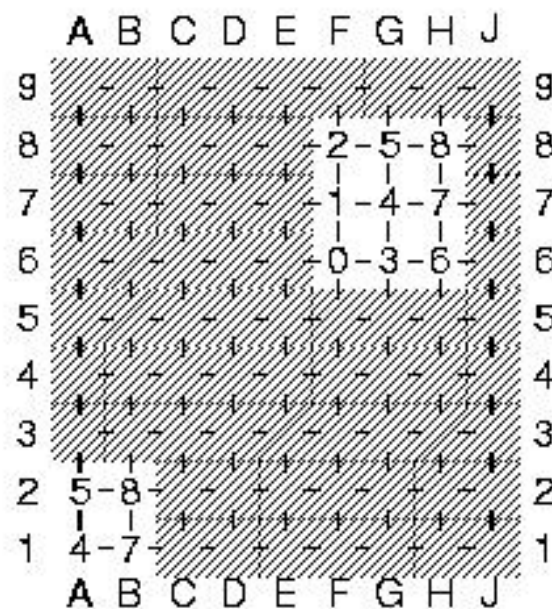


Figure 3. Receptive fields. Field indices for 2 receptive fields centred at A1 and G7.



Neuronales Lernen in Go

- **Lernen: Spiele mit sich selbst**
 - Bewertung dabei die Summe aller Ausgaben (für alle Felder) der Bewertungsschicht vor und nach dem Zug
 - In 15 % der Züge nicht den besten Zug sondern Heuristik (Gibbs sampling).
 - $P(s) = \exp(s/T)$



Neuronales Lernen in Go

- Beispiel zur Evaluierung und zur globalen Verbindungsschicht

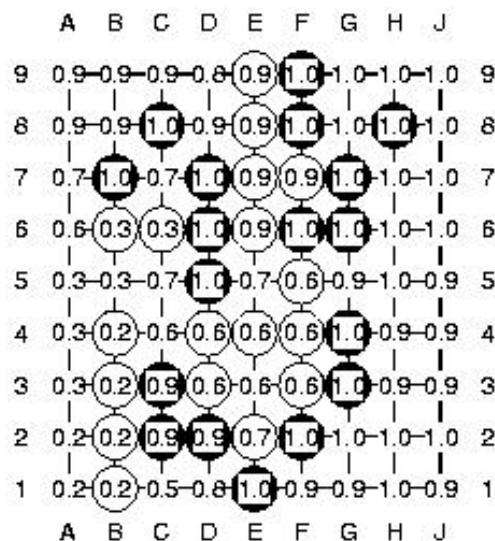


Figure 5. Example position (last move White E2). The numbers show the evaluation output of the network using preprocessed input.

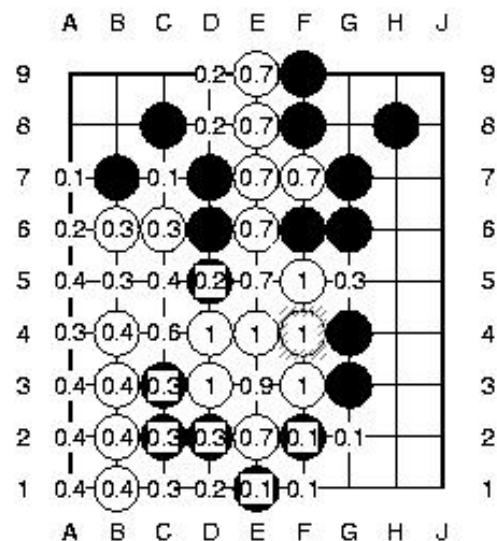


Figure 6. Example position (last move White E2). The numbers show the connectivity map of the network using preprocessed input for White from the point F4.



Neuronales Lernen in Go

- **Allgemeines zur Bewertung der Spielstärke von Programmen**
- **Spielausgänge sind im Allgemeinen nicht transitiv**
- **Schach tendiert dazu transitiv in diesem Sinne zu sein, deshalb ist hier eine Bewertung noch sinnvoll möglich**
- **Bei Go ist die Spielstärke mehrdimensionaler**
- **Möglichkeit: Jeder gegen Jeden, Punkte summieren**
- **Problem beim Menschen – spielt er sein „normales Spiel,, oder passt er sich genau den Schwächen des Computers an ?**



Neuronales Lernen in Go

- **Go gehört in Klasse "P-Space complete"**
Eigenschaft von P-space complete:
Alle bekannten Algorithmen haben exponentielle Laufzeit
- **Go Programm Honte**
- **Ansatz: Benutze Neuronale Netze an so vielen Stellen wie möglich, für den Rest werden klassische Methoden angewendet**
- **Scheinbar ist es sinnvoll die Spielprozesse bei denen der Mensch „Augenmass“ einsetzt, durch Neuronale Netze zu modellieren**

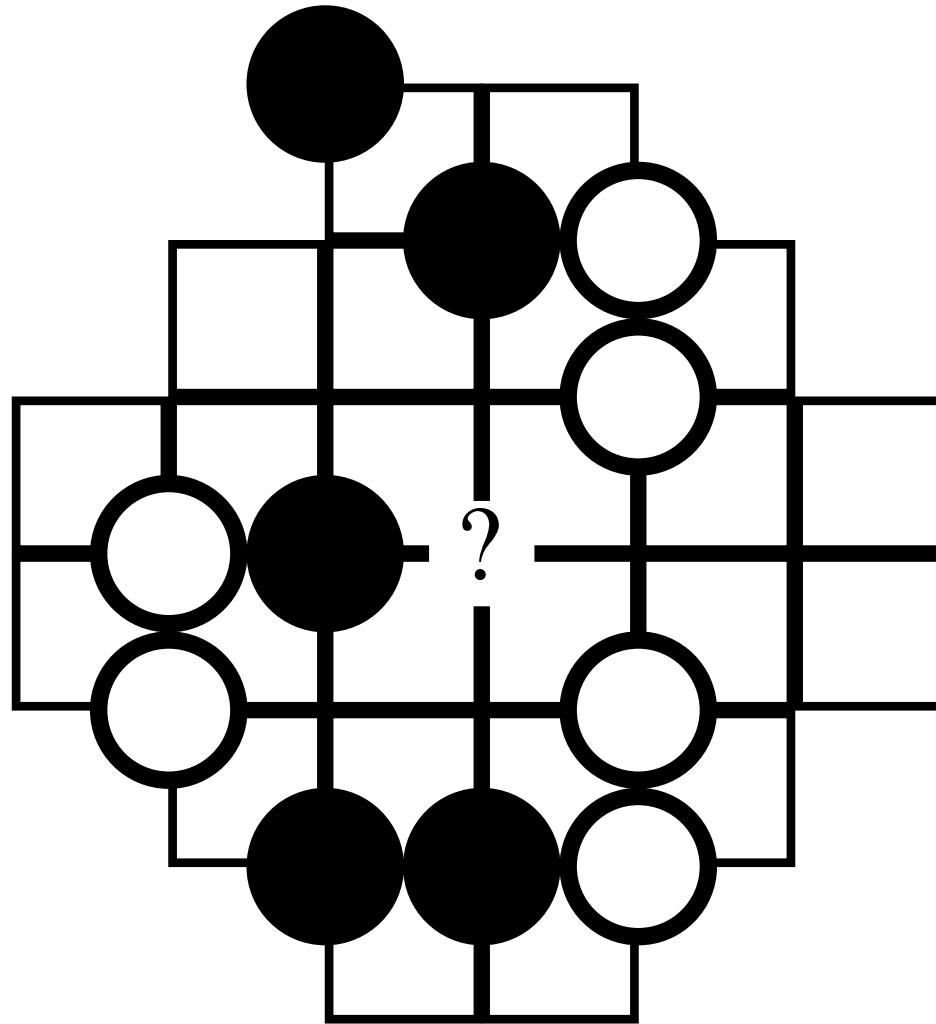


Neuronales Lernen in Go

- **Joseki Bibliothek**
- **Neuronales Netz zur Mustererkennung in Honte**
- **Das „Receptive Field“ mit Durchmesser 7 und dem Referenzpunkt als Mittelpunkt, gedreht das die am nächsten gelegene Ecke unten ist und die zweitnächste links – Auswahl eines Repräsentanten der 8-elementigen Symmetriegruppe**



Neuronales Lernen in Go





Neuronales Lernen in Go

- **Zwei Eingangsneuronen für zwei Farben, außerdem der Abstand zu den nächstgelegenen 2 Ecken als Eingänge**
- **Training**
 - 400 Expertenspiele
 - Zufällige Situation im Spiel
 - Der reale Spielzug als positive Bestärkung
 - Ein zufälliger erlaubter Spielzug als negative Bestärkung
- **Konzept des Einflusses**
 - Die meisten Go Programme verwenden eine Einflussfunktion
 - Hier: Zufallslauf vom Referenzpunkt aus mit $\frac{1}{4}$ Wahrscheinlichkeit in jede Richtung zu verzweigen. Durch Aufsummieren dieser Zufallsläufe erhält man die Stärke des Einflusses auf diesen Referenzpunkt
 - Nach 20 Feldern wird der Zufallslauf gestoppt. Damit erkennt man ob sich der Stein in einem dicht besiedelten Gebiet aufhält oder nicht



Neuronales Lernen in Go

- **Neuronales Netz für Sicherheit und Territorium**
- **Sicherheit von Gruppen wird beim Menschen durch Augenmass festgestellt**
- **Bestimmung der benutzten Features schwieriger als beim Netz zur Mustererkennung**
- **Eingangsdaten**
 - Anzahl der Freiheiten aller Gruppen in einem Gebiet
 - Anzahl der Augen
 - Fremdeinfluss über die Freiheiten



Neuronales Lernen in Go

Quellen

Evolving Neural Networks to play Go

Norman Richards, David E. Moriarty, Risto Miikkulainen

The University of Texas at Austin

Evaluation in Go by a Neural Network using soft segmentation

M. Enzenberger

University of Alberta, Edmonton, Alberta, Canada

Honte, a Go-Playing Program using Neural Nets

Frederik A. Dahl

Norwegian Defence Research Establishment