Attila Egri-Nagy

Igo Math – Natural and Artificial Intelligence and the Game of Go

v2021.07.15

These preliminary notes are being written for the MAT230 course at Akita International University, and for the Technical Communication course at Akita University in Japan. Some parts are finished, others just barely started. Comments are welcome! When reporting errors please specify the version number. The latest version can be downloaded from `https://egri-nagy.github.io/igomath/`

Contents

Preface

What is this book about? What can we gain from reading it?

This book is about *the game of Go*. It describes the rules and teaches some elementary tactics and strategy. However, it does not contain a training program and it does not introduce the culture of the game. Therefore, it cannot compete with decent introductory books.

The book is also about *artificial intelligence*. It describes its core ideas and fundamental algorithms. It explains how inanimate 'mechanisms' can imitate and surpass the abilities of our thought processes. However, this book concentrates on a single application only. There are more detailed and more comprehensive textbooks on this technical field.

The book is also about *mathematics*. Once we talk about something technical, a precise language is needed. By definition, that is mathematics, so it is inevitable. Obviously, one can build up Go strength without studying mathematics (as almost all professional players in Asia do). But, if we want to talk about that expertise, or want to explain to a computer how to play the game, logical exactness and statistical analysis are needed. Therefore, we will introduce concepts from combinatorics, graph theory and probability theory. We will restrict the mathematical content to those concepts only that are needed for discussing the game. Again, there are better textbooks for these mathematical topics.

Why this book then? What is the problem we are trying to solve here? Consider the following scenario. A student's meeting with her academic advisor.

STUDENT: I worked hard, but my results are not good. I have to improve.
ADVISOR: What is the problem? Do you have a plan to fix it?
STUDENT: I don't know, but I will work harder.

What is wrong here? The student has an evidently failing method for studying. With good intentions, the idea of doubling the effort comes naturally. However, a good advisor should say something like this.

ADVISOR: Maybe, you need to work 'easier'.

Mental effort is tiring, so the wrong method will make more trouble, failure guaranteed. The student needs to find a different way for studying, a more efficient method. For that purpose, one has to

perform self-reflection. Asking questions like *How do I study?*, *Can I do it more efficiently, maybe in shorter time?*, *What did I do before? Was it successful? Did it fail?*, etc.. This is not easy. Self-reflection is a learned skill. It would be helpful if we could learn it in a simpler situation. Board games offer such a context.

The main purpose of this book is to facilitate the reader to learn more about his or her thinking skills. Go, as any other abstract strategy board game is a clean laboratory for experimenting with our minds. *What is my strategy? Is there a better one? What did I do in won/lost games?* These questions are similar to the questions about the learning method above, but easier to answer. Go is lot simpler than life itself. Yes, it's deep and beautiful, complex and mysterious. However, we can always play another game, in which we can use our knowledge gained from previous games. In life, we have to come up with a good move in each new situation, and often there is no way to repeat the situation, or we can do it only at a great price (e.g. choosing partner, a profession, schools, etc.). Real world problems are way more complicated than board games, but some meta-level general wisdom gained on the board could be transferred to life. This is the main assumption and the promise of this book.

Knowledge transfer is not automatic. Popular belief is that playing strategy board games will make you more clever. There is no evidence for this. This fallacy is a typical example of ignoring the 'correlation does not imply causation' rule of Statistics. The only guaranteed effect of studying the game is becoming a better player. However, paying attention to the process of improving, being conscious about the learning process could make learning the next skill more efficient. This is the main idea.

AIs are often modelled after our own thinking processes. Thus, learning about them is a way to learn about our cognition. AIs can serve as mirrors, in which we can see ourselves. By studying AI algorithms, we can get a new appreciation of the capabilities of, and an understanding of the limitations of the human brain.

This idea is as old as it could be. Alan Turing said: "The whole thinking process is still rather mysterious to us, but I believe that the attempt to make a thinking machine will help us greatly in finding out how we think ourselves. " In: Can digital computers think?, 1951.

We can summarize these points in an 'equation':

$\frac{1}{3}$ Go + $\frac{1}{3}$ AI + $\frac{1}{3}$ Mathematics = transferable metacognition skills.

We don't know what skills future jobs will require. It is a reasonable guess that the one will need to learn new skills often, which is demanding. Therefore, training emotional intelligence and for mental resilience can be a good investment.

Part I

Go

A metaphorical introduction to the game of Go

The goal of the game of Go is to surround more territory than the opponent, building a wall around some areas by placing stones on an initially empty grid. This enclosing task would be trivial if it was done in the way of city planning. In that case, the land for a new district development is parcelled out first, and then people can buy the allotted plots. Officers in high-viz suits appear on site equipped with theodolites. They take measurements and put pegs in the ground to designate the corners of the plots. After that, the owners can build the walls and erect fences to enclose their estates, undisturbed by the construction on the neighboring plots. No need to hurry since the map in the district office contains the exact dimensions of the estate as a permanent legal record. You can enclose a smaller area inside your plot, but you will still be the owner of the whole parcel. However, you cannot fence a larger area off, as there would be legal consequences. The land size cannot be changed by what you build.

Now imagine that there is no central authority deciding land ownership and keeping legal records. You can have any area, as long as you can build a complete circle of walls around it. Initially, it is a free grab. The boundaries are not known, determined, or dictated in any way before construction work starts. On the contrary, the borders are the result of the construction work itself. But you are not the only one having this opportunity. Wall-building becomes competitive, and thus, grabbing a sizeable share of the land is challenging. You may start erecting a fence, but others will do the same, crossing your planned borderline. You realize that you do not have time for continuously building a wall. You need to place a brick here and another over there, always keeping an eye on where the others put theirs. This situation would be chaotic and dirty in real life, leading to harm and an ugly cityscape. However, this is what happens on the Go board: **Area ownership is decided by what the players build on it.** The art of skillful play is in the trained imagination. One should be able to foresee the final boundaries from a few scattered stones on the board. Recognizing the future potential of the shape of a few stones close to each other is essential, both for a finished wall and for disrupting the construction work of the nearby opponent stones. A short list of guidelines cannot summarize this skill. It can only be earned by seeing many examples, i.e., playing many games, both for humans and computers.

The metaphor, of course, has a limited reach. In Go, it is possible to capture enemy stones. What would that be in the construction work land grabbing scenario? Stealing bricks, kidnapping masons? Also, in Go there is only one opponent. Variants with more than two players are different games, since the off-board group dynamics can be decisive.

The rules of Go for human beings

How to play Go? What are the rules?

Go is often advertised as a game with very simple rules, which is therefore easy to learn. True, compared to other games, we need less words to define the rules. Unlike Chess, for Go we do not need to describe the different pieces and their movements. there is only one type of stones, and they do not move at all. However, it is not entirely true that the game is easier to learn. Simply knowing the rules does not guarantee meaningful games. For instance, beginners often have no idea when to finish the game. One has to know some basic tactical and strategic knowledge, the ability to decide the status of groups. And that requires longer explanations. In any case, we have to start with stating the rules of the game.

The game of Go is an ancient, abstract, two player, turn taking, pure skill, strategy board game. It is based on the concept of *surrounding*, enclosing on all sides.

The goal of the game is to surround more territory than the opponent.

This can be achieved by

1. building walls with your stones,

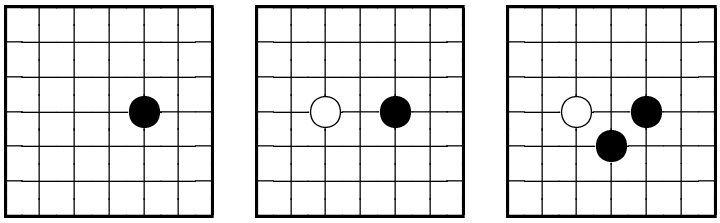
2. capturing your opponent's stones.

Capturing itself is not the primary purpose of the game, but it is an efficient way of encircling more territory.

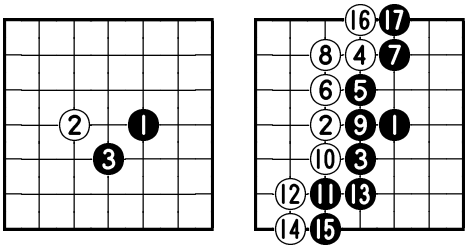
Wall building

The game played on a square grid. The standard size is 19×19. Smaller ones, 13×13 and 9×9, are used for shorter games and recommended for beginners.

The two players are represented by black and white. Starting with black, the players take turns in placing their *stones* on empty intersections, one at a time. Here are the first 3 moves of a 7×7 game.

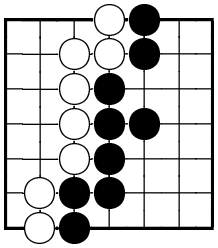


By numbering them, these three moves can be summarized in a single diagram. Since the stones do not move, we can record a whole game this way.



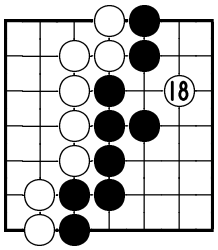
The game ends when both players pass, when they see no benefit of making more moves. Scoring this game is easy, since territory ownership is clear. More precisely, we can pick an empty intersection on the right side, and by hopping from an intersection to another, following the lines in any direction we can reach only black stones, or the edge of the board. Therefore, the right side is black's territory. Similar thinking shows that the left side belongs to white. At the end, black surrounded 19 territory points and white 13, therefore black wins this game.

In the beginning, deciding whether a game is finished or not, is not at all straightforward. As a quick rule of thumb, if the territories are not sealed off, then the game is not finished yet.



This example looks like a peaceful wall building game. However, each move is a territorial threat or counter-threat. A game has the tension of a serious fight even if there is no capture.

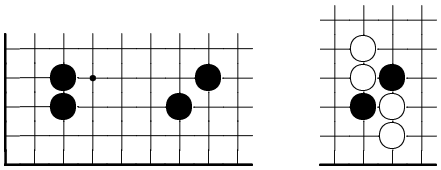
But what happens, when white makes another move?



This is invasion, ⑱ is behind enemy lines. Is the right side still black's territory then? This question often confuses beginners. The answer is simple: they have to fight it out. If black can capture invading white stones, then it remains black's territory. If black fails to surround the invading stones, then white can gain more territory while reducing black's at the same time.

Capturing

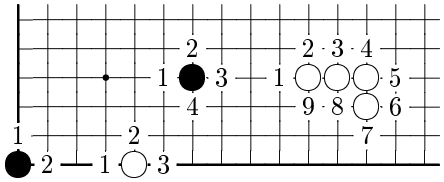
Stones are *connected* if they occupy neighbouring intersections, they 'touch' each other either horizontally or vertically. Here are two connected stones (left), and two disconnected stones (right). There is no diagonal connection, just as there are no diagonal lines on the board. The diagonal black stones can be separated by white stones.



Connected stones of the same color form *chains*. In terms of surrounding, a chain is a single thing: either the whole chain remains on the board, or it gets captured. In other words, connected stones share the same fate. We can think of a single stone as a chain by itself.

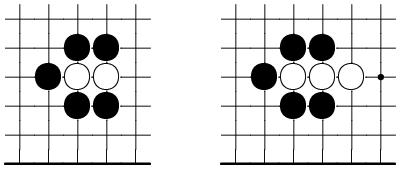
The word 'chain' is used metaphorically. It's about the connectedness of stones of the same color, not about having a long, signle-threaded shape.

The chains need "breathing space", i.e. empty intersections in direct contact with stones in the chain. These are called *liberties*. The number of liberties is an important property of a chain, and counting liberties is crucial in tactical fights. It is a measure of how easily the chain can be surrounded. Here are four chains with their liberties counted. A single stone in a corner is very weak, better on an edge, and strongest with 4 liberties.

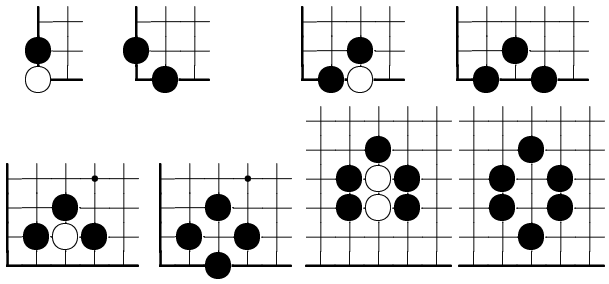


Surrounding a chain is the process of reducing its liberties. A special situation is when a chain has only one liberty left, and we say it is in *atari*. White needs to extend the chain to avoid capture.

Is it possible to be in atari without contacting an enemy stone?



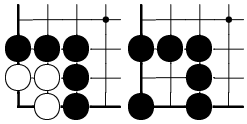
Capturing a chain is filling its last remaining liberty, when all stones in that chain are removed from the board and they are kept separately as prisoners. Here are four examples of capturing.



Self-capturing is not allowed. Black cannot make a move into the corner as that intersection is surrounded by white.

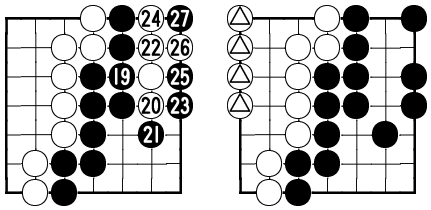
Prohibiting self-capture is not necessary. Some rulesets allow it. While it may look useless, it can make a difference in ko fights (see later).

However, one can fill his/her own last liberty in order to capture an enemy chain at the same time. It creates new liberties, so the move is not a self-capture anyway.

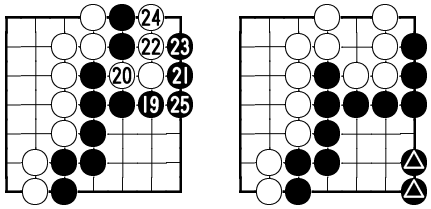


In the scoring phase, prisoners are put back on the board into their own territories to reduce score. This way, capturing an enemy stone is worth two points. One for the surrounded point and one for reducing the enemy's territory.

Going back to the example game, here is one way how black can deal with the invasion. ㉗ captures the invading white stones. Now black has 14 points of territory and white's territory is reduced to 9, since the captured stones △ are placed inside white's territory.

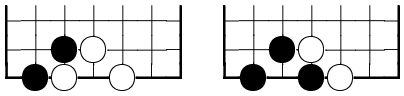


Another scenario might be that black does not connect his two chains. ⑳ cuts away the upper black chain. ㉑ is atari, but white can capture the two black stones. Successful invasion, white turned the game. At the end, black has 8 points of territory, white has 15. The upper right corner does not belong to anyone, it is a *dame* point.



No eternal games

To guarantee that a Go game will be finished in a finite number of steps, **it is not allowed to repeat a previous board position**. Here is a simple situation where the potential for infinite play occurs.

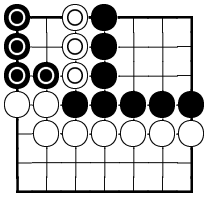


Black captures the white stone, but the capturing stone ends up in atari. However, white cannot recapture, since that would restore the previous situation. Consequently, white is obliged to play somewhere else first, thus changing the overall board position. Then, black can choose between answering the remote threat, or resolving the local situation. This is known as the *ko rule*. This sounds like a technical rule, but it leads to an interesting game dynamics by connecting remote parts of the board in the same fight. Ko fights are very much part of the strategy of the game.

More on scoring

Statistical investigation shows that black has a real advantage by moving first. Therefore, white gets compensation points, the *komi*. The actual value is unclear since we only have empirical evidence. Typical values are 5.5, 6.5 or 7.5. The half a point is used as a tie-breaker.

The following game is already in the scoring phase. Whose territory is the upper left corner? Whenever, there is uncertainty, players can decide it by continuing to play. However, for the circled chains, none of the players wants to make a move. Putting the enemy chain into atari is self-atari as well. This is mutual life, *seki*. Both chains stay on the board, but no territorial points can be made in that corner.



Black has 9 in the upper right, and white 15 in the bottom half.

Advice for beginners

These rules describe how to play valid Go games. However, they give no instructions on what are the good moves. Mastering the game is a long and gradual process. For beginners, the best advice is: **PLAY!** In the first few games it is important to simply observe with an open mind what happens on the board, without trying too hard to win. Have fun!

"...if you see an enemy stone, try to capture it, or cut it off. If you see a friendly stone, try to save it from capture, try to connect it." [1]

[1] T. Kageyama. *Lessons in the Fundamentals of Go*. Beginner and Elementary go Books Series. Kiseido Publishing Company, 1998

Building up terminology

"The limits of my language are the limits of my mind. All I know is what I have words for. " Ludwig Wittgenstein

How can we say it in simple words, what is happening on the board?

Whenever we want to talk about something, we need to find words for describing the object. We can create new words, or assign new shades of meaning to existing words, or used them metaphorically. Mathematics is a way of building a language precisely, often not by words, but simply by symbols. First, we just need a descriptive and intuitive language for talking about what happens on the board.

People often rely on intuitive understanding, communicating based on the given context. Therefore, few books get strict about Go terminology, see for instance EZ-GO [2] – written by a software engineer.

[2] B. Wilcox and S. Wilcox. *EZ-go: Oriental Strategy in a Nutshell*. Ki Press, 1996

Describing board positions

Stones connected or close to each other are often referred as 'groups'. Here we refine this concept by identifying the 'building blocks' of board positions.

Stones are the smallest units, the 'atoms' for building a game.

Chains are formed by stones that are in direct contact vertically or horizontally.

Links are close range, but not direct connections.

Groups are links of chains.

Board position An arrangement of black and white groups.

Properties of chains

The rules of Go state that stones fully surrounded cannot stay on board. Can we quantify this concept? Can we measure how far is a chain from capturing?

Liberty is an empty intersection neighbouring/adjacent to/directly next to a chain. We count liberties for chains, not for stones.

Atari means the situation when a chain has one liberty left, i.e. being one move away from being captured.

Status of groups

Dead stones: stones that cannot form two eyes or connect to living friendly stones.

Eye is some empty space surrounded by stones of one color.

False eye is an empty space fully surrounded, but with the possibility of a surrounding stone being captured.

Eyespace is the room available for a group to make eyes.

Alive groups have at least two eyes, i.e. the territory they surround is divided into at least two parts. A group with single eye can also be classified as alive if the eyespace is big enough to make two eyes even if the opponent interferes with the eye building process.

Dead stones have no chance of making two eyes.

Unsettled stones' fate depends on whose turn is it.

Rules say nothing about shapes with two eyes being unconditionally alive. It is an *emergent* property.

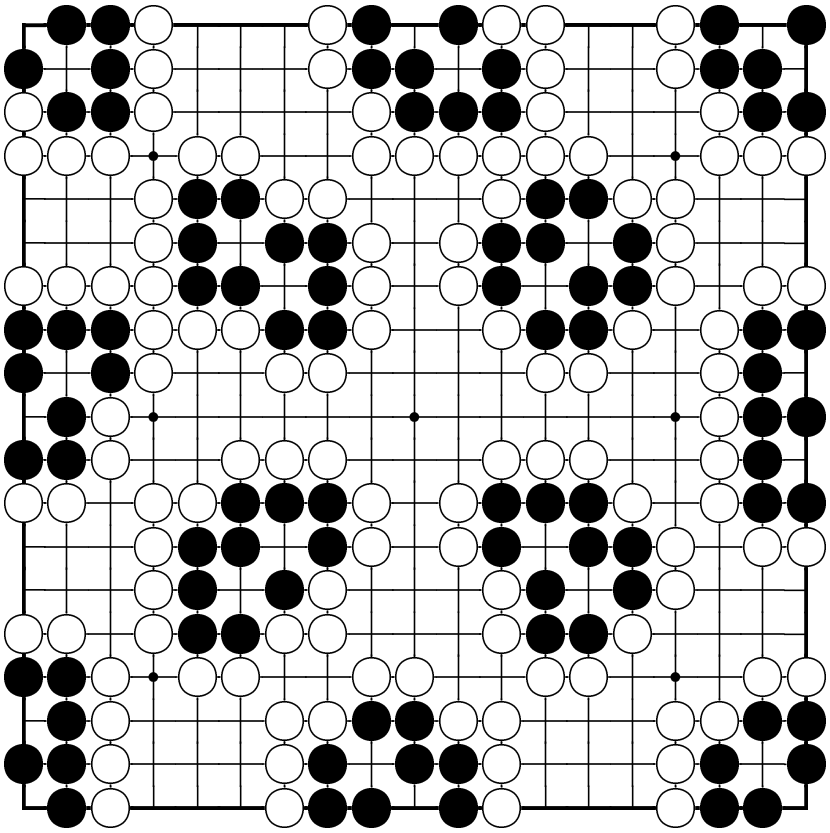


Figure 1: A collection of minimal living shapes.

Part II

Game management

Time control

How much time we have for a game? How to make time usage fair?

The ko rule ensures that a Go game will surely end, sooner or later. But how long can a game be? We need to limit the amount of time for two reasons. First, for practical purposes. Tournaments last a day or two; classroom activities have their allocated time slots. Therefore, we need to able to predict the length of a game, not just the fact that it will end. Secondly, time control changes the nature of the game. Good moves need time consuming consideration, therefore time limits make the game harder to play well. Faster games more rely on intuition than calculation skills.

The basic idea of time control is to penalize using too much time. One can even loose a game due to the lack of time. There are numerous ways to control time. The different methods can be classified by the following questions.

1. *Do we limit the total, or the time for making a move, or both?*
2. *Is the available time fixed, or changes dynamically?*
3. *How to manage overtime? What happens when a player has no time left? Is it immediate loss, or is the player allowed to play under more severe restrictions?*

Here are some frequently used time control methods. The most basic ones are those where going overtime means loosing the game, regardless of the board position.

Absolute A fixed amount of time is given for the whole game. It's up to the player how much time to spend on each move. This requires a strategy: Is it better to spend more time in the opening, aiming for a better position, or to keep time for tactical fights and endgame problems, making a comeback after an inferior opening?

Simple A fixed amount of time is given to each move. The length of a game can be estimated by the average number moves in games. Of course, not all moves are equal. Forced moves can be played immediately, while strategic decisions need lot of thinking. Therefore, simple time control can be distortive.

In more sophisticated methods, going overtime leads to some 'grace period'. The player can continue but on different terms for time. Thus depleting main time is not an immediate loss, but one is forced to play faster.

Byo-yomi (countdown) After depleting the fixed main time, the player has a fixed number of time periods. If a move is made within a period, then it restarts for the next move. Otherwise, it expires, reducing the number of available time periods.

Canadian After the main time, the player has to make a certain number of moves within a time period in order to get another time period – sort byo-yomi for several moves.

Then there are schemes where the available time changes dynamically, rewarding moves.

Fischer (bonus) Initial time is given to the player, then each move earns bonus time. The bonus times can be accumulated to a certain limit.

Rankings and ratings

What does it mean to be good at playing Go? How can we measure playing strength? Why do we need to measure strength?

For a simple game, like Tic-tac-toe, we have only two levels of strength: knowing or not knowing the best strategy. However, if a game is complex enough, measuring strength is not straightforward. If the game is not solved, and playing it well requires numerous mini-skills, then even the pairwise comparison of players can be difficult. It is a known phenomenon that player A beats reliably B, B plays better than C, but C can beat A. This circular relationship violates the idea of a strict order. Therefore, a purely logical approach is not suitable. Whenever we deal with something that is complex enough that we cannot understand and describe it succinctly, we need a different approach. Science figured this out long time ago: the world is messy enough, so we can best describe it in a probabilistic and statistical language. Playing skills are also complex enough, therefore we need to estimate strength of players by the result of many games against several different players.

It also makes 'Rock paper scissors' to be an interesting game.

A warning about terminology needs to be made here. *Ranking* is about comparing players. Telling whether one player is ranked higher or lower than the other. *Rating* is putting a player on a common scale, no comparison is involved. In everyday usage, these terms are not distinguished strictly.

Traditional Go rankings

Student ranks are *kyu*s. Beginners start at 30kyu and by playing a couple of games quickly advance to around 20kyu. Casual players (19-10kyu) are often referred as DDKs (double digit kyus). Similarly, intermediate amateurs are called SDKs (single digit kyus). After 1kyu one can reach 1dan – the equivalent of the black belt. After that levels can go up to 7dan.

The idea of these ranks is that difference between the ranks determines how many handicap stones have to be given to the weaker player in order to have an even game. For instance, a 3kyu player would give 2 stones to a 5kyu player, and a 3-dan would give a 2 to a 1-dan. This system becomes unusable beyond 9 handicap stones, so normally that's the largest handicap size on the full board.

The handicap stone calculation is valid on the 19×19 board. Playing black with no komi or 0.5 is one handicap stone.

How can a player be promoted? One of the simplest algorithms is that 3 consecutive wins against stronger players will allow the player

to advance one rank. This is of course a very informal approach and not so reliable.

Professional players have a different scale from 1dan to 9dan. The levels are closer to each other than a full handicap stone. These are issued according to strict rules of professional organizations.

Élő rating system

When players of different strengths start a game, what can we say about the expected result? Can we guarantee that the better player will win? No, not really. She has a higher chance of winning for sure, but probabilities are not about certainties. Human minds are complex enough, and the circumstances can be different, therefore a game is a probability event. The idea is that player's rating should predict the probability of winning a game. More precisely, a player's strength can be modelled as a normal distribution, depicted as a Bell curve. The center of the curve is the rating of the player, representing the average performance. Using two Bell curves, we can compute the probability of the outcome of the game. The actual game result can be different form the expectation, thus we need to adjust the ratings. There is a possibility for an upset win, when the weaker player wins the game, in that case ratings should change significantly. If the stronger player wins, then ratings should change little, so there is not much point in beating weaker players.

The Élő rating system, originally devised for chess, is a mathematical implementation of these ideas. Here is the notation used.

players	A, B
player ratings	R_A, R_B
expected scores	E_A, E_B
game result	S_A, S_B

The sanctioned lowest rating value is 100, a beginner starts at 800, an average player is around 1500, and master level starts at 2200.

Expected results

A game result can be 0 (loss), 0.5 (draw), and 1 (win). Why these numbers? Because they can be interpreted as probability values. Indeed, the expected scores are the probabilities of winning, therefore $E_A + E_B = 1$. Here are the formulas for calculating the winning probabilities for both players.

These come from the way we can calculate the probability of the difference for random samples from the two normal distributions.

$$E_A = \frac{1}{1 + 10^{\frac{R_B - R_A}{400}}}$$

$$E_B = \frac{1}{1 + 10^{\frac{R_A - R_B}{400}}}$$

Calculating E_B with a formula is of course not needed, since $E_B = 1 - E_A$. The heart of this formula is the signed difference between

the players' ratings: $R_A - R_B = -(R_B - R_A)$. With this, it is an easy exercise in algebra to show that $\frac{1}{1+10^x} + \frac{1}{1+10^{-x}} = 1$. But what is this mysterious number, 400, doing in the formula?

Let's check a concrete example. Let $R_A = 1400$ and $R_B = 1000$, so A is stronger than B with a difference exactly by 400 points. How does this translate into winning probabilities?

$$E_A = \frac{1}{1+10^{\frac{1000-1400}{400}}} = \frac{1}{1+10^{\frac{-400}{400}}} = \frac{1}{1+10^{-1}} = \frac{1}{1+\frac{1}{10}} = \frac{1}{\frac{11}{10}} = \frac{10}{11}$$

Consequently, $E_B = \frac{1}{11}$. Therefore player A is ten times more likely to win the game than player B. That's the meaning of a 400 point difference. Simple reverse calculation shows that about 120 points difference corresponds to a player being twice as good (winning twice as many games).

Why 400? For historical reasons only. Mathematically any other number would do.

The computation can be simplified by letting $Q_A = 10^{\frac{R_A}{400}}$, and $Q_B = 10^{\frac{R_B}{400}}$. Then

$$E_A = \frac{Q_A}{Q_A + Q_B}, \; E_B = \frac{Q_B}{Q_A + Q_B}.$$

How does this simplification work? In the exponent we can write the fractions as two separate fractions.

$$E_A = \frac{1}{1+10^{\frac{R_B}{400} - \frac{R_A}{400}}}$$

Then using the law of negative exponents,

$$E_A = \frac{1}{1+\frac{10^{\frac{R_B}{400}}}{10^{\frac{R_A}{400}}}},$$

so

$$E_A = \frac{1}{1+\frac{Q_B}{Q_A}}.$$

Updating ratings

After a game result S_A, S_B (note that also $S_A + S_B = 1$) the updated rating can be calculated by

$$R'_A = R_A + K(S_A - E_A),$$

where K is the so called K-factor regulating the speed of rating changes. It is bigger for lower rated players, for instance $K = 32$ and smaller for master players, like $K = 16$, regulating the speed at which ratings can change.

Properties

Élő rating system is self-correcting. An upset win will take points from a stronger player and give it to the winning weaker player, since we have an evidence for the weaker player's improvement. Also, an expected win for a strong player yields little increase, thus it is no point in beating weak players.

Improvements: Glicko and Glicko-2

`http://www.glicko.net/glicko.html`

Ratings reliability. *RD*, ratings deviation (1 standard deviation), measures the accuracy of a player's rating.

Rating volatility, the expected rate of fluctuations in one's rating. The measure of how consistent is someone's performance.

Tournaments

How to decide who is the best player in a group? In case we are not content to choose the best player by their ratings, i.e. based on the previous performance, we need to organize a competition. A *tournament* is a single competitive event with a large number of players. The problem is logistical. *How much time is needed for a tournament?* There is a relationship between the length and the quality of the tournament. The longer ones tend to be regarded as higher quality, in terms of the quality of the individual games and overall fairness and enjoyability of the competition. *How to minimize the role of luck in a tournament? How to maximize the number of meaningful games in a given time period?* There are of course several ways to answer the above questions.

Round-robin, all-play-all tournaments

In a round-robin tournament, everyone plays everyone else. Therefore, for n players, we need $n-1$ rounds, and there will be $n*(n-1) = n^2 - n$ games played in total. This way, we remove the luck element. A single bad game has a limited influence on the outcome. We can prove mathematically that a round-robin tournament (with no draws) guarantees a complete ranking of the players, not just deciding the champion. Mathematically speaking, there exists a Hamiltonian path in the directed graph of the tournament. This path, however, may form a cycle, the so-called circle of death: A defeats B, B defeats C, and C defeats A, in a rock-paper-scissors-like situation. The probability of this problem decreases as the tournament size grows.

Disadvantages are the length of the tournament and weaker players will play meaningless games against strong players.

Elimination tournaments

If we want to shorten the length of a tournament, then we need to eliminate players who lose games as the competion proceeds. *How many lost games results in elimination?*

Single-elimination

This is the strictest possible elimination tournament: a single loss means leaving the tournament. Every round of games half the number of contestants. Consequently, length is $\log_2 n$ rounds. Luck is decisive. Whoever faces the best player in the first round, will have no chance to win games against other players. Single-elimination is only good for choosing the champion and the second player. We need to play one more game to decide the 3rd and 4th places.

Half of the players play all only one game. This could be a major disadvantage if traveling to a tournament has some costs.

Double-elimination

We can improve the elimination tournament by allowing the players to have one loss and still stay in the tournament. After first loss, a player falls down to the lower bracket, but retains the possibility of being the champion. The final is between the winner in upper bracket and the winner in the lower bracket.

Swiss and MacMahon tournaments

These combine features of round-robin and elimination tournaments. The aim is to maximize the number of games between players of equal strength. In the Swiss system, after some random choices in the beginning, the subsequent pairing is done by the number of points accumulated so far. This way winners play winners and loosers are pitted against other loosers.

In the MacMahon system (developed specially for Go) the are initial groups (bands) based on the known ratings of the players.

Ladder tournaments

Ladder tournaments are ongoing, challenge based competitions. A player can challenge another player above in the ladder order. When loosing, nothing happens. When winning, the challenger jumps one step above the challenged player. In the long run, assuming random challenges, the order will reflect the true ratings of the players. In practice, this does not guaranteed since players can be very strategic in choosing their challenges.

Part III

Go AIs

The logical rules of Go – for computers

How to 'explain' Go to a computer? Can we describe the rules with mathematical precision?

One thing is to roughly explain the rules for human beings, so they can start playing right away. If something is not fully clear, they can pause, discuss the matter and then continue the game with some mutual agreement. Humans are good at dealing with unclear situations cooperatively.

It is a different level of precision if we want computer to play the game. The rules have to be strictly watertight and unambiguous. The computer cannot stop and negotiate. A Go playing program is a mechanism after all. At each step it has to be clear what to do next.

As such, they are not really suitable for human consumption. Giving only mathematical rules to beginners would probably reduce the number of players worldwide.

Here are the rules that are more clear logically (based on the Tromp-Taylor rules [3]), and can be used for implementing the rules in a computer program. The comments can shed light on the meaning of the terse mathematical description and show the connections with the intuitive rules.

[3] J. Tromp. Tromp–Taylor rules. `http://tromp.github.io/go.html`, 1995

1. Go is played on a $m \times n$ grid of points, by two players called Black and White.

   Traditionally square grids, but nothing in the rules rely on m being the same as n. In fact, the game can be generalized to be played on any set of points with an adjacency relationship (for each point we can find its immediate neighbours, mathematically speaking this structure is called a *graph*).

2. Each intersection point on the grid may be colored black, white or empty.

   The term 'coloring' naturally describes the act of putting down a stone (while removing a stone may be understood as coloring the intersection empty). The coloring idea also comes from *graph theory*, the mathematical study of relations between objects.

3. A point P, not colored C, is said to *reach* C, if there is a path of (vertically or horizontally) adjacent points of P's color from P to a point of color C.

   This idea of reaching, or seeing, or touching is the crucial one for defining a simple rule set.

   This compresses a lot of meaning into a single rule. It implicitly defines the chains (points of the same color connected by path(s) along the grid lines). Then, for a chain, we check what *other* color(s) does it have contact with. Interestingly, this is done from the perspective

of a single point. The chain emerges due to the fact that all points in a chain reach the same set of colors. A point does not reach its own color.

4. *Clearing* a color is the process of emptying all points of that color that don't reach empty.

   This process ensures that there will be no chains left on the board with no liberties. The empty color has the role of deciding what stones can stay on the board.

5. Starting with an empty grid, the players alternate turns, starting with Black.

   Alternating turns are common ot many strategy board games. Starting with black is traditional.

6. A *turn* is either a pass; or a move that doesn't repeat an earlier grid coloring.

   Turn is a word for covering two different possible actions. Passing is doing nothing, the position on the board does not change. It happens when a player thinks the game is over. A move changes the position, but cannot go back to a previous position (positional superko).

7. A move consists of coloring an empty point one's own color; then clearing the opponent color, and then clearing one's own color.

   A move is a three-stage process. 1. placing a stone, 2. capturing enemy stones, 3. capturing friendly stones. The order is important and there are some logical relationships between stages 2 and 3: at most one of them can happen in a move. If 2 happens, 3 will not take place: capturing enemy stones will guarantee at least one liberty for the capturing stone. If 2 does not happen, 3 might happen: the case of self-capture, suicide move.

8. The game ends after two consecutive passes.

   Dead stones are removed from the board and scoring begins. If there is a disagreement about the status of some chains, playing can resume.

9. A player's score is the number of points of her color, plus the number of empty points that reach only her color.

   This is are scoring. It allows to fill up one's own territory, since it doesn't matter whether we count a point as a surrounded territory or as a friendly stone. Thus proving that enemy stones in one's territory can be captured has no cost.

10. The player with the higher score at the end of the game is the winner. Equal scores result in a tie.

    Komi (points added to white) can be used for adjusting the scores to offset black's advantage due to starting first. Komi including half a point breaks ties. The exact value of komi is disputed, depends on statistical evidence only.

Game Trees

How to represent a Go game mathematically?

A *game tree* is a theoretical construct that describes all possibilities of a game. It is a data structure that contains all information we need to play the game perfectly. A game is *solved* if we have quick access to its full game tree, i.e. we know the best play in each position. Complete solutions are only available for simple games, or for small board sizes. The 9×9 board has 1039191487912938343189830904387987934 69 legal positions. For the 19×19 board this number has 171 decimal digits [4]. That's why the game tree is a theoretical construct. For real games we can never have the full tree.

[4] John Tromp and Gunnar Farnebäck. Combinatorics of go. In *Computers and Games*, pages 84–99. Springer, 2007

What kind of information does a game tree contain? It has all game states, i.e. all situations that can appear in the game. In other words, these are the snapshots of a game. For instance, all legal arrangements of stones on a Go board.

The key idea of the game tree is how this collection of game states is organized. We do not just list them. Game states have a natural relationship: one board position can be a result of a legal move made in another board position. Like cities are connected by roads, a game tree is a network of board positions connected by moves. Another metaphor for this structure is a family tree, in which people are connected by parental relationship. We often say 'child node' even in an abstract tree.

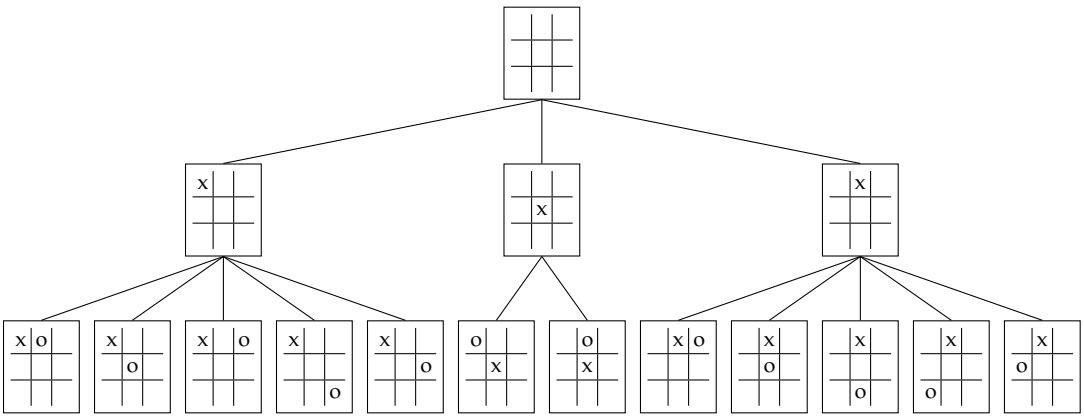


Figure 2: Part of the game tree for Tic-tac-toe. The first two moves are displayed. We simplified the game tree by considering the rotational and reflectional symmetries of the board.

An actual game is a *path* in this game tree. The path starting from the *root node* (empty board) to a *terminal node* (a finished game). A

game is simple sequence of board positions. However, when we play, we need to consider other moves as well, not just the played ones. This is exactly the decision process we need to execute in our minds to play well. We compare future possibilities, alternative moves. This requires exploring the tree, so a question naturally arises. *How many possibilities do we have on average at each position?* That number is the *branching factor* of the tree. Even if the branching factor is only 2, the growth of the tree is exponential. The number of possibilities n moves in the future is 2^n. This is an exponential function, which is a fast growing function. Exponential growth is bad news for playing games. It means that exploring the tree is a resource intensive (time and memory) operation.

For small games it could happen that several branches of the tree end up in the same node. This simplifies the drawing, but it is not a tree any more, but a more general *game graph*. For bigger games this is highly improbable. In the case of Go, it is simply not allowed by the rules. Computationally, it is better to work with the game tree, since checking every new positions against existing nodes is time consuming.

Minimax

Assuming that we don't need to worry about the exponential growth of the game tree, there is an algorithm to solve the game. If we can get to the terminal nodes from any node, so we can see into the future far enough to see finished games, we can find the best result we can achieve, no matter what the opponent does.

The algorithm is based on a common sense idea: we want to maximize our gain and try to minimize the opponent's progress. This simple idea is the *minimax* algorithm. It is about "minimizing the expected maximum loss", hence the name. It is important that we assume perfect play from the opponent. We are not thinking about an ideal sequence of move, where the opponent cooperates. If the worst case scenario is still favourable, then we are winning. This good thinking in real life playing as well.

To solve the game, we need to work backwards from the terminal nodes. We know the scores for all finished games. For calculating the best value for a non-terminal node, we need to know the best achievable scores of all of its children. Then we just need to find the maximum or minimum based on whose turn is it. This way the best achievable values eventually propagate back to the root node.

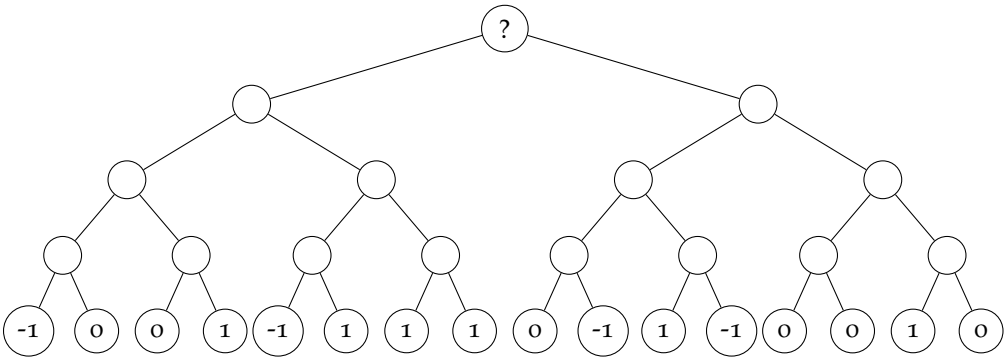


Figure 3: For this abstract game tree, we know the game scores for terminal nodes. Once the game is finished it is clear who is the winner. The question is, if A moves first, what result can be guaranteed?

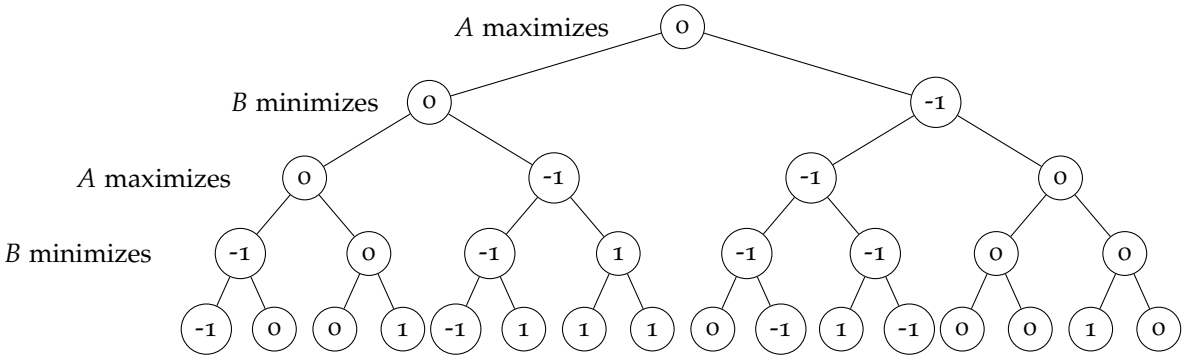


Figure 4: Going bottom-up, the minimax algorithm shows that the best guaranteed result is a draw for A. A can only win if B makes a mistake.

Monte Carlo Tree Search

Can we use randomness to produce intelligent behaviour?

In Go, the classical tree search algorithms are useless. The branching factor of the game tree is high. On average, there are simply too many legal moves. And unless the game is finished, it is difficult to evaluate a position. Who is winning? Who's got the advantage? This depends on the status of the groups, and solving life and death problems itself is already challenging enough. Therefore, rational calculation is not enough for solving the evaluation problem. We need a new idea.

Randomness is defined by the lack of a regular pattern, the absence of predictability. In a sense, it is the opposite of intelligibility. It is then surprising that randomness can be used for realizing intelligent behaviour. Indeed, the next step in computer Go was the application of Monte Carlo methods.

We will explain these in three steps, solving successively more complex problems:

- evaluating a single board position,
- choosing the next move in a given position,
- searching for the best sequence of moves from a given position.

Evaluating a single position

What does it mean to evaluate a board position? The value of the position for a player is defined by the chance of winning. The ultimate answer is to give a set of paths in the game tree that leads to the best achievable result, no matter what the opponent does. The guaranteed result for perfect play defines the exact value of the position. However, for complex enough games, this is not an option. The next best thing is to estimate the probability p of Black winning the game from that position. In order to calculate this probability, we can do something that looks silly: playing a random game – just putting stones on the board without thinking. For computers, it does not require much effort to play random moves, since they can generate random (enough) numbers efficiently and use them to choose from the set of available legal moves. When a random game finishes, it is easy to decide who is the winner by simple counting. No need to decide the life and death status of the groups. Since, what could be captured,

The probability of White winning is simply $1 - p$.

For humans it is quite challenging to produce truly random behaviour.

had been captured. Of course, a single random game is not useful at all, the result is due to chance. The trick is to do it many times. We can sample the future possibilities by playing random games from the given board position. Then, for instance, how can we explain the observation of Black winning more often then White? The playing strength is exactly the same for both players, so the only remaining explanation is that in the given position Black has the advantage.

We call these random games *playouts* or *rollouts*. We can calculate the probability of winning by the ratio of wins in the simulations by the number simulations.

$$P(\text{Black winning}) = \frac{\text{number of playouts that Black won}}{\text{total number of playouts}}$$

To establish notation, we simply write

$$p = \frac{w}{n}.$$

As we do more and more rollouts, this ratio becomes a more reliable approximation of the value of board position. Thus we can conclude that in order to do something clever, it is enough to do something dumb, but for many many times. Later, researchers tried to put more knowledge about Go itself into the random games. For instance, not filling up second eyes, or using a collection of patterns for good local moves. These are called 'heavy' rollouts. Of course, one has to trade in the number of simulations, since with more things to check, rollouts become slower. It is again surprising that heavy rollouts do not help much. In a way, it is really possible to play the game well without understanding it.

Exploration or exploitation

Let's say we can make k interesting moves from a given board position. We want to figure out which move is the best one by random sampling. Random sampling works if we can do many playouts. The more, the better it performs. On the other hand, we have a time limit, so we can do only a fixed number of rollouts. Therefore, a question arises. How do we distribute the available simulations between the moves? Or more directly, which move shall we simulate next?

This problem is known as the multi-armed bandit problem.

The simplest idea is to do the same number of rollouts for each candidate move. However, it is not efficient enough. We don't want to waste simulations on obviously bad moves, and we want to be sure about the promising ones.

This is the problem of exploration vs. exploitation. Shall we do more rollouts on promising moves, or shall we try unexplored moves?

A better approach is the ϵ-greedy algorithm. With probability *epsilon* we choose randomly (exploration), and with probability $1 - \epsilon$ we choose the one best performing so far. It is easy that a fixed ϵ value may not be good in all situations. We need an algorithm that can dynamically find the best exploration/exploitation ratio.

The technical name of a good solution is UCT, Upper Confidence Bound 1 applied to trees. It is described by a simple formula. In order to decide where to spend the next rollout, we calculate a score value for each child node.

This is a simplified view of MCTS, restricting the search for one level only. In a real implementation we would explore the tree locally by choosing a path with this method.

$$s_i = \frac{w_i}{n_i} + c\sqrt{\frac{\ln N}{n_i}},$$

where w_i is the number of wins for the ith node, n_i is the number of simulations so far, c is the exploration parameter (its theoretical value is $\sqrt{2}$, in practice it can be changed), $N = \sum_i n_i = n_1 + n_2 + \ldots + n_k$, the total number of simulations so far.

Having the score values, we can simply choose the node with the highest score for the next rollout. Or, we can use a more sophisticated method. We can normalize the score values, i.e. for each s_i we calculate $\frac{s_i}{S}$, where S is the sum of score values. This gives probabilities for each node , then we just roll the dice and choose randomly according to these probabilities.

Not to be mistaken with the winning probabilities, these are the probabilities for choosing a node for the next simulation.

After spending all available rollouts, we simply choose the next move by finding the highest winning probability so far.

MCTS

The above idea of evaluating a single board position can be combined with the tree search. Using the same decision about exploration versus exploitation, we can traverse the existing tree and decide where to extend it. This way we can concentrate on the promising lines of play and discard the bad moves.

Deep Learning

We recognize human faces without conscious effort. Similarly, professional Go players can evaluate board positions faster than the time required by a logical analysis. They look at the board, and they can tell who is ahead. Therefore, we can try to model Go playing after our vision. How does our vision work?

Neural networks

Neural networks can process information in a distributed way, where each processing unit do a very simple computation. There is no central control and the solution for a problem emerges from the activity of the whole network.

Biological neurons

Biological *neurons* are cells that carry electrical impulses. They are connected to each other through *synapses*. The human brain has billions of neurons and trillions of synapses. How is it possible that a network of these very simple neurons can give rise to consciousness and cognitive processes? We don't know (yet). It could be that not even possible, the neural network model misses an important ingredient. However, these networks are certainly capable of recognizing images of, let's say, cats and dogs. They can also learn how to evaluate board positions.

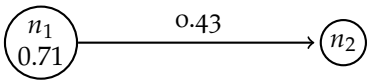
Artificial Neurons

Loosely based on biological neurons, an artificial neurons are little information processing units. At a given moment of time it is just a piece of information, its current level of *activation*. This could be represented as a real number between 0 (inactive) and 1 (active). Here is a neuron n_1 with activation level 0.71.

Or we can use the interval $[-1, 1]$ for activation values, depending on the application we use the network for.

One neuron itself is not worth much, but in a network they can do some complex information processing. How does the network process information?

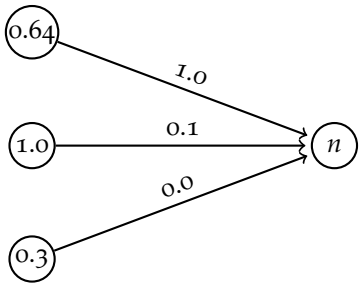
A neuron can send its activation level as a signal to other neurons through a synapse.



the arrow indicates the direction of the information travelling. The connection has a certain strength. We representing this by putting a *weight* value on the connection. If the connection is stronger, then more of the activation is received by the second neuron. If it is a weaker connection, then the receiver will get a smaller value even if the activation is high. In the above example, the second neuron will receive the value $0.71 \times 0.43 = 0.3053$.

Negative weights can model inhibitory connections.

A neuron receives signals from several other neurons. The total amount of input it receives is the weighted sum of all the inputs.

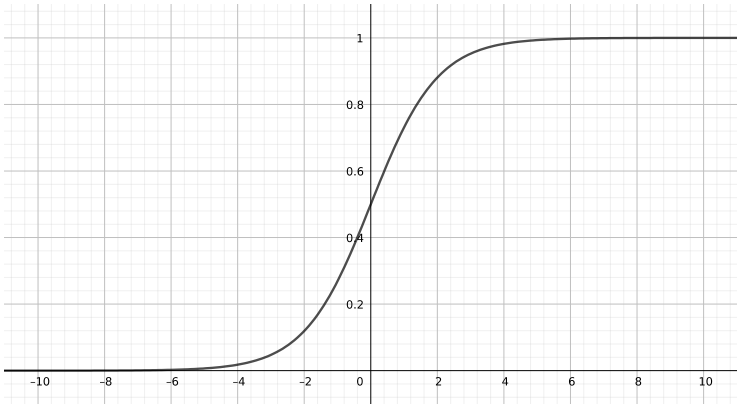


The total input that neuron n receives is $0.64 \times 1.0 + 1.0 \times 0.1 + 0.3 \times 0.0 = 0.64 + 0.1 + 0 = 0.74$.

The operation for computing the weighted sum for several neurons is the multiplication of a vector (of activations) by a matrix (of weights). Therefore, in essence, neural networks are just some linear algebra.

The value of the weighted sum of inputs is passed to the neuron's *activation function*, which is a non-linear function, to calculate the activation level. A classical choice is the sigmoid function.

$$f(x) = \frac{1}{1 + e^{-x}}$$

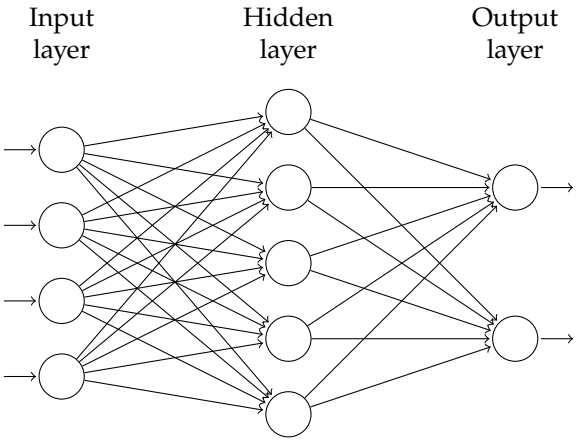


With this we can compute the new activation level, the output: $f(0.74) = 0.68$. Thus, a neuron is a *function*, in the technical sense, of many variables. It is just not given as a short algebraic formula.

Calculus is a useful subject for understanding neural networks.

For image recognition, a neural network has a feed-forward structure. The pixels of the image are connected to the input neurons. The

output neurons give the answer, for example for recognizing cats and dogs.



The network is *deep* if it has many hidden layers.

If a single neuron is a function, then its output feeding into other neurons, functions is *function composition*.

Training

Part IV

Getting better

Thinking in Strategic Board Games: Calculation and Intuition

What is thinking? This is a very big and general question. In order to say something useful about this cognitive process, we need to limit the discussion. We can consider *problem solving*, which is reaching some goal in a situation by choosing suitable actions from a range of possibilities. To make it even more concrete, we restrict our attention to traditional board games like Chess and Go. In these, the goal is clearly defined by the conditions of winning. Similarly, the possible actions, the legal moves, are clearly defined.

Chess and Go are two player games. They are *abstract*: they do not directly model reality like other themed board games. Even Chess pieces are quite removed from how a real army in historical times looked like. They are *strategic*: the outcome is determined by the decisions of the players. There are no secrets, no hidden information (like in Poker). The position on the board is all that's needed for making decisions. Chance is not involved, only the decisions matter. Therefore, these games are like personal laboratories for the decision making process [5], with all the possibly confusing factors removed.

What is the problem to be solved? To win a game. *What are the possible choices?* All legal moves in a position, allowed by the rules of the game. Winning depends on making good decisions when choosing moves. This is by no means easy. There are basically two different ways to achieve successful move selection: *calculation* and *intuition*. These are also two basic modes of thinking in general [6].

What is calculation? It is working out a desired result by following some rules of computation. It is a step-by-step process. At each stage we know exactly where we are, and how we got there. Therefore, the steps can be verified. The prime example is arithmetic calculation, but it is not the only one. In board games, it is natural to consider not just the next move, but the opponent's possible next moves. And the next moves after the opponent's move, and so on. Trying to maximize the gain for our moves, and minimize the opponent's benefit. This is also a standard adversarial search technique in classical artificial intelligence , called minimax [7].

There are limits to this method. The number of combinations of moves and countermoves gets huge quickly. Therefore, it is most useful for local tactical fights in Go, or in endgame situations in Chess, where the number of pieces to move is small. Then the question arises, how do we make strategic decisions involving the whole posi-

[5] G. Kasparov. *How Life Imitates Chess – Insights into life as a game of strategy*. Arrow Books, 2008

[6] D. Kahneman. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 2011

[7] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, 3rd edition, 2009

tion on the board? We rely more on our intuition.

What is intuition? As we keep playing these games, insights about the next move start to happen. We suddenly have the right idea, we just "see" the solution, but cannot explain why. We feel that a move is a good one. But, we do not have a calculation proving that it is indeed the right choice. Is this some magical talent? Not really. It is just that our brain has very efficient information retrieval capabilities from previous memories.

How to improve thinking in games? To play well we need a mix of calculation and intuition. Both can be improved by practice.

The challenge in calculation is that we have to use our short term memory excessively. We cannot put the variation on the board, thus we have to play it out in our head. We have to imagine the sequence of moves. This is a very good memory exercise. No one doubts that calculation skills can be learned. By doing similar calculation exercises repeatedly, the process gets automated. Some patterns become familiar, so we do not need to calculate them. Consequently, we can do longer calculations.

On the other hand, intuition is often thought as a gift. A talent that people simply have or they do not. The opposite is true. Expert intuition can be trained. It is the natural consequence of calculation practice. Just by playing games, exposing the brain to the patterns on the board we can improve intuition. The brain does the clever pattern storage, but we have to constantly feed it with data. Either by playing, or replaying famous games. Intuition originates from a network of internalized and interconnected knowledge, and its creativity comes from the swift navigation of this network [8]. Roughly speaking, it is a very efficient and inventive database search.

[8] J. Waitzkin. *The Art of Learning: An Inner Journey to Optimal Performance*. Free Press, 2008

In reality, the distinction between calculation and intuition is not that clear cut. Repeated calculation becomes second nature, turning into intuition; while we try to turn intuition into a more calculable method, or at least more verifiable. Getting stronger is not a linear process. It is a messy interplay between the two basic modes of thinking. The development has the following stages (again, not so well separated).

1. applying simple "rules of thumbs"; they often work in beginner games

2. accumulating expertise by making observations on how the simple rules succeed or fail; this could happen without conscious thinking, simply by exposure,

3. extracting 'deep', statistical rules from intuition, often in the form of proverbs

There seem to be no shortcuts in the learning process. We may be tempted to ask an advanced player, to give us a summary of the essence of her expertise, a list of deep rules. The trouble is that these rules are statistical. Simple tactical advice comes in sharp 'if condition then action' form. But for statistical rules, the condition is difficult to recognize, thus expertise is not directly transferable.

Are there deep laws of precision at all? Is the perfect strategy expressible in a compact way, or does it have to be an exploding list of positions? Fully solved games are probably just vast databases of positions and the corresponding winning moves.

How do computers think? Artificial intelligence follows the same development. Deep Blue mastered calculation [9], and AlphaGo intuition [10]. Now, AIs are stuck at the same problem of extracting deep rules. It is not clear how to understand intuition in a step by step calculative manner, how to interpret the knowledge distributed in the connection weights of an artificial neural network. Through self-playing, a powerful computer can recreate and surpass all human Go wisdom in three days [11]. Still, that knowledge is not directly transferable. We can play the engine, just like learning from a human master player.

To end with a philosophical note, we observe that strategic board games are lot easier than life itself. In human life we simply don't have time to build up expertise, as the situation keeps changing. Literature can offer some shortcuts. We can experience other people lives by reading novels, training our intuition for the challenges of human existence.

[9] G. Kasparov. *Deep Thinking: Where Machine Intelligence Ends and Human Creativity Begins*. Millennium series. Hodder & Stoughton, 2017

[10] David Silver et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016

[11] David Silver et al. Mastering the game of go without human knowledge. *Nature*, 550:354–359, October 2017

Playing strength – Seeing the future

What is the difference between stronger and weaker players? Anyone aspiring to be a better player asks this question at some point. Ratings quantify strength by measuring the probability of winning. Predicting the winner of a game is a good practical tool, but it doesn't say much about what makes up the real difference. The question is still there: What does a dan player know, that a kyu player does not?

First we can think about what would be satisfying answer. Certainly a list of skills would be helpful. One could use it as a to do list, acquiring skills one by one. We have no such list. We could try to assemble one. Maybe listing tesujis, finding some representative life-and-death problems. The list might be very long, containing numerous miniskills. Go books try to do that in a way. But there is no universally agreed selection of books. Moreover, the skills may not be as separated as a list would suggest. The connections between the miniskills might be different for different people. Therefore, comparing what books they read and internalized may not be a reliable way to understand the difference. What we can do is to give a framework, a narrative, in which we can understand the differences. The framework here is defined by how we can relate to future events.

"The acquisition of expertise in complex tasks such as high-level chess, professional basketball, or firefighting is intricate and slow because expertise in a domain is not a single skill but rather a large collection of miniskills."

D. Kahneman. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 2011

Planning is reasoning in a "what-if" style. It involves objects and events that are not present in the time of reasoning. Therefore it is also called off-line thinking [12]. It is argued that mathematics is simply this activity performed on a higher level of abstraction [13]. The thesis here that strength come from the ability of 'reasoning' about future events on the board.

In the very beginning, after learning the rules, players consider only what is on the board. If a group is in atari, they capture it or save it. If there is a cutting point, then they do the cut or connect. The time horizon of their thinking is limited to the present, bound to the current board position.

As the understanding of the game progresses, future possibilities come into play. Moves that are not only important now, but will be decisive later on. The time horizon expands. A game becomes the clash of thoughts about future possibilities. It is a richer scene compared to what is available now, the present arrangement of stones. There is only one configuration on the board, but there are many follow-up positions. Well, not just many. The right term is 'combinatorial explosion'.

Thinking about the future also explains why beginners find the games of stronger players incomprehensible. The reason for a move

[12] Derek Bickerton. *Language and Human Behavior*. Jessie and John Danz lectures. University of Washington Press, 1995

[13] K. Devlin. *The Math Gene: How Mathematical Thinking Evolved And Why Numbers Are Like Gossip*. Basic Books, 2001

is not on the board yet. They may answer a potential threat, or eliminate the potential. The real reason may never appear on the board. In a way playing strength can be understood as the ability to "see" future possibilities.

In a sense, calling Go a complete information game is slightly misleading. Surely, there is no hidden information, the game state is totally defined by the board position. But that is just information about the present. In order to win, we need information about the future. We have to imagine where the game goes. *What are the ways for imagining the future?*

1. In a local, tactical fight we can systematically go through the available moves and counter moves. Then moves and counter moves again, and so on. This is **brute-force search**, and even computer are limited in performing it. Seeing everything in the future is not an option. The number of possibilities grow way too fast. We use the term *combinatorial explosion*, instead of growth.

2. Standard sequences (josekis) are time travel devices. We can safely fast forward to a future position, since deviating from the sequence is a disadvantage (by the definition of a standard sequence). It's one big macro-move. We can evaluate that future position, and maybe other josekis as well. This is **chunking**, treating a great deal of information (moves in the sequence) as a single unit.

3. Good shape is not the easiest concept, but it can be understood through a simple definition: a group of stones has a good shape if it has good potential for making territory, i.e. few moves can achieve the goal of making eyes or attacking. An aesthetic sense develops when one plays games attentively. This is **pattern recognition**.

At the end of the game it becomes clear what each stone had achieved in contributing to the final score. Estimating this score count beforehand is the ultimate way of using the time horizon. Each move is an investment and we are guessing its future return. On small boards it is easier to see how each move changes the estimated final score.

Sente is about reducing the number of possibilities available to the opponent.

This also explains why practising tsumegos with a computer may not be efficient. We are tempted to click through the variations to see which one is correct. Though systematic thinking, elimination of the possibilities is useful for finding a solution, but it trains only the skill of recognizing what's on the board now. Just trying out the variations does not train the skill of imagining future. It does not require expanding the working memory. Surely, in a real game we are not allowed to try the variations out on the board.

Strength in Go is common sense, practical judgement about what to do given a board position. Attack when needed, defended when needed. However, common sense is something to acquire. It takes

several years to learn it. Kids certainly don't have it, e.g. if you want to wake up early then you need to go to bed early.

Activity: Liberty Analysis

Counting liberties during the game is important, but beginners often neglect to do so. They recognize atari, but that might be too late. Here is an exercise that may help to see the dynamics of liberty counts for chains on the board.

Take a 9×9 game record and replay the game. For each move,

- for isolated stones, record its coordinates and number of liberties – a new chain is created
- when attaching to an existing chain, update the liberties of the existing chain by simply writing down the new number (so the history of liberty changes can be seen); if it is the same number, then it may be omitted
- when two (or more) chains get connected, keep the most recent one with recomputed liberties and cross out the previous one(s);
- in any case, update the liberties of opponent chains

This list is similar to data structures that classical AI Go playing programs use to keep track of the chains.

Learning from the AIs

Who can we learn from? What are the necessary conditions of learning? Without trying to sketch a full theory, we can list a couple of common-sense ideas.

During learning, we acquire some knowledge we did not have before. Where do the new insights come from? Either we create them, or we get them from someone else. Here we think about the second option. Therefore the first condition is that we need someone who has the expertise we desire to have. There can be exceptions, though. In Go, a beginner can create board positions that the stronger player's skills do not cover. This case is not about transmitting information. The novice player is not telling what to do but creates a situation where learning can happen for the more experienced. While emphasizing the cognitive process's collaborative aspect is essential, we restrict the investigation to the straightforward way of learning: a teacher imparts wisdom, and the student absorbs.

The second condition is the possibility of communication. The teacher should be able to explain, and at the same time, the student should understand the reasoning. What can go wrong? The student may not understand the language of the teacher. This communication failure can happen both literally or metaphorically. The teacher might speak an unknown foreign language. Or, the language can be comprehensible, but missing background knowledge could hamper understanding.

Let's summarize. For learning to happen in a teacher-student relationship,

1. the teacher should know,
2. he/she should be able to communicate that knowledge, and
3. the student should be able to comprehend that imparted knowledge.

We can check these conditions in a human player's learning from a deep learning Go AI engine.

1. The AI knows how to play well. It exhibits consistent superhuman playing strength. Thus the first condition is fulfilled completely.
2. The AI has a severely limited way of communicating ideas. It only tells the winning probabilities and score estimates of promising moves. Using the AI for analysis, we can see the strong choices,

but there is no explanation. Why do they work? Why are they better than the alternatives? These questions can be answered weakly by showing principal variations, the projected optimal lines of play. These may or may not give a reasoning for the chosen moves. Therefore, the communication condition fails.

3. The lack of explanations puts an extra burden on the student, the user. Since the teacher does not actively help, the student has to do the work. It is the task of a scientist: creating explanations and criticize them. In Go, explanations are plans, strategies. In human games, they determine what happens on the board. In deep neural networks, high-level plans are not present, or at least not in an understandable way. Thus the third condition does indeed depend on the student.

What is the difference between a Go book and a Go AI? A book never contains the exact board position of my game. If it is a typical shape, it may have a similar arrangement. But we know that in Go, similar could be wildly different. Just a stone being a bit closer or further away could change the power relations. Whether the difference is brutal or subtle, it does not matter. The point is that distinct board positions have different evaluations; thus, no book answers me.

We can conclude that the best way to learn is by having an active conversation with a better human player. AIs can be used as a substitute if one is willing to work hard with the mindset of a scientist.

Bibliography

[1] Derek Bickerton. *Language and Human Behavior*. Jessie and John Danz lectures. University of Washington Press, 1995.

[2] K. Devlin. *The Math Gene: How Mathematical Thinking Evolved And Why Numbers Are Like Gossip*. Basic Books, 2001.

[3] T. Kageyama. *Lessons in the Fundamentals of Go*. Beginner and Elementary go Books Series. Kiseido Publishing Company, 1998.

[4] D. Kahneman. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 2011.

[5] G. Kasparov. *How Life Imitates Chess – Insights into life as a game of strategy*. Arrow Books, 2008.

[6] G. Kasparov. *Deep Thinking: Where Machine Intelligence Ends and Human Creativity Begins*. Millennium series. Hodder & Stoughton, 2017.

[7] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, 3rd edition, 2009.

[8] David Silver et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[9] David Silver et al. Mastering the game of go without human knowledge. *Nature*, 550:354–359, October 2017.

[10] J. Tromp. Tromp–Taylor rules. `http://tromp.github.io/go.html`, 1995.

[11] John Tromp and Gunnar Farnebäck. Combinatorics of go. In *Computers and Games*, pages 84–99. Springer, 2007.

[12] J. Waitzkin. *The Art of Learning: An Inner Journey to Optimal Performance*. Free Press, 2008.

[13] B. Wilcox and S. Wilcox. *EZ-go: Oriental Strategy in a Nutshell*. Ki Press, 1996.