

Efficiently Updatable Neural-Network-based Evaluation Functions for Computer Shogi¹

Yu Nasu
Ziosoft Computer Shogi Club
April 28, 2018

Abstract

Die „Sankoma-Kankei“ (lit. „Drei-Figuren-Beziehung“), die in heutigen Computer-Shogi Softwareprogrammen als Bewertungsfunktion benutzt wird, kann zwar sehr schnell berechnet werden, aber keine nicht-linearen Zusammenhänge erfassen.

Dieses Paper beschreibt eine auf einer CPU schnell berechenbare, neuronale-netzwerkbasierende Bewertungsfunktion „Efficiently Updatable Neural-Network-based evaluation functions“.

Die NNUE-Bewertungsfunktion ist durch die Optimierung für den CPU-Betrieb und durch Beschleunigung mittels Differenzberechnung gleicher Geschwindigkeit läuft wie die Sankoma-Kankei-Bewertungsfunktion.

„the end of genesis T.N.K.evolution turbo type D“ ist das erste Shogi-Programm, dass diese Bewertungsfunktion implementiert, und wird an der 28. Shogi Computer-Weltmeisterschaft teilnehmen.

Vorwort

Thema dieses Paper ist das Softwareprogramm „the end of genesis T.N.K.evolution turbo type D“, dass an der 28. Shogi Computerweltmeisterschaft teilnehmen wird. Beschrieben

¹ Übersetzt aus dem Japanischen von Dominik Klein (Vorname . Nachname at outlook . com)

wird die verwendete Bewertungsfunktion, experimentelle Auswertungen sind jedoch nicht enthalten. Die Fähigkeiten oder grundsätzliche Stärke des Programms ist ebenfalls nicht Gegenstand dieses Papers; hier wird auf die Ergebnisse der Weltmeisterschaft verwiesen.

Einführung

Die Auswahl eines Zugs bei Computer-Shogi Software ist hauptsächlich bestimmt zum einen durch eine Bewertungsfunktion die die Stellung bewertet, sowie einer Suchfunktion. Um eine hohe Spielstärke zu erreichen, ist es notwendig eine Bewertungsfunktion zu verwenden, die sowohl genau ist als auch schnell berechnet werden kann. Je genauer die Bewertung ist, desto desto stärker spielt das Programm. Wenn die Bewertungsfunktion genau ist, kann ein genaues Urteil über die Spielsituation abgegeben werden.

Wenn die Genauigkeit des Bewertungswertes hoch ist, kann die Form des Spiels genau beurteilt werden, und wenn die Berechnung schnell ist, ist es möglich, tiefer zu suchen. Der Grund dafür ist, dass die Beurteilung der Form umso genauer ausfällt, je höher die Genauigkeit des Bewertungswertes ist, und je schneller die Berechnung ist.

Mit Stand vom März 2018 basiert die typische Bewertungsfunktion, die in Computer-Shogi-Software verwendet wird, auf einem lineares Modell, das als "Sankoma-Kankei" (etwa: „Drei-Figuren-Beziehung“) bezeichnet wird. Die Idee wurde erstmals 2003 vorgestellt [2], und seit der Veröffentlichung des Quellcodes von "Bonanza" [3], der nur für Kombinationen mit Kugeln implementiert wurde, im Jahr 2009, ist sie weit verbreitet. Nach einigen Erweiterungen, wie z. B. der 2014 in "NineDayFever" [4] eingeführten Zugauswertung, wird sie immer noch in den meisten der hochrangigen Shogi-Software verwendet.

Der Vorteil der Sangoma-Kankei Bewertungsfunktion ist, dass die Funktion durch eine große Anzahl von Parametern bestimmt ist, der Bewertungswert (d.h. die numerische Stellungsbewertung) aber schnell berechnet werden kann. Bei der Berechnung des Bewertungswertes einer Stellung, die einen Zug von einer anderen Stellung entfernt ist, und der Bewertungswert für die andere Stellung schon berechnet ist, kann das gleiche Ergebnis wie bei der Berechnung der gesamten Bewertungsfunktion viel effizienter erzielt werden. Nämlich indem die Differenz zwischen den Bewertungswerten der ursprünglichen Stellung und dem Bewertungswert der neuen Stellung durch Addition oder Subtraktion der Gewichte der bewegten Figuren berechnet wird. Die Einfachheit der Differenzberechnung ist ein großer Vorteil der Verwendung eines solchen linearen Modells für die Bewertungsfunktion.

Die Tatsache, dass es sich um ein lineares Modell handelt, erschwert jedoch auch die Ausdrucksfähigkeit der Bewertungsfunktion. Nichtlineare Beziehungen, wie z. B. die Positionsbeziehungen bestimmter Figuren, die in manchen Situationen vorteilhaft, in anderen aber nutzlos sind, können nicht direkt durch die Sangoma-Kankei Bewertungsfunktion ausgedrückt werden. Um die Ausdrucksfähigkeit bei Beibehaltung eines linearen Modells zu verbessern, ist es notwendig, die Bewertungspunkte zu ändern, aber mit Stand März 2018 ist kein besseres lineares Modell als die Sangoma-Kankei Bewertungsfunktion bekannt.

Die Verwendung nichtlinearer Modelle, insbesondere neuronaler Netze, wurde als Ansatz zur Realisierung von Bewertungsfunktionen mit höherer Ausdrucksfähigkeit versucht. Der Pionier dieses Ansatzes ist "Naruso" [5], eine Shogi-Software mit einer nichtlinearen Bewertungsfunktion, die den Wert einer Figur nach der Sicherheit des eigenen und gegnerischen Königs gewichtet. In den letzten Jahren war Software, die eine groß angelegte CNN-Auswertungsfunktion (Convolutional Neural Networks) verwendet, im Computer-Go

erfolgreich [6], und es wurden Versuche unternommen, CNN im Computer-Shogi einzusetzen. Die meisten Shogi-Programme, die CNNs verwenden, nutzen GPUs, die gut im parallelen Rechnen sind, und verarbeiten Batches von Spielpositionen, um mehrere Positionen auf einmal auszuwerten, um die Geschwindigkeit zu erhöhen. Dennoch gibt es bis März 2018 mit Ausnahme von "AlphaZero" [7], das spezielle, für neuronale Netze optimierte Hardware verwendet, kein berichtetes Beispiel für eine Softwareimplementierung, die eine mit erstklassiger Shogi-Software vergleichbare Spielleistung erreicht.

In diesem Aufsatz schlagen wir eine Bewertungsfunktion auf Basis neuronaler Netze vor, die auf einer CPU mit hoher Geschwindigkeit ausgeführt werden kann, da das für CPU-Operationen optimierte Design und die Beschleunigungstechniken wie die Differenzialberechnung es ermöglichen, die Bewertungsfunktion auf Basis neuronaler Netze mit mehreren hidden Layers mit der gleichen Ausführungsgeschwindigkeit wie die Sankoma-Kankei Bewertungsfunktion auszuführen. Wir nennen diese Bewertungsfunktion "NNUE-Evaluation Function" (Efficiently Updatable Neural-Network-based evaluation functions).

2. NNUE Bewertungsfunktion

Die NNUE-Bewertungsfunktion ist eine neuronale Netzwerkbewertungsfunktion, die jeweils ein Spielposition auf der CPU bewertet, ohne einen Grafikprozessor oder Batch-Verarbeitung zu verwenden. Ähnlich wie bei der Sankoma-Kankei Bewertungsfunktion ist die Berechnungszeit für die Auswertung einer Spielposition gering, und sie lässt sich leicht mit Minimax-Suche und Branch Pruning kombinieren. Abbildung 1 zeigt ein schematisches Diagramm der NNUE-Auswertungsfunktion in "the end of genesis T.N.K.evolution turbo type D". Diese Struktur ist für Hochgeschwindigkeitsberechnungen auf einer CPU ausgelegt und unterscheidet sich stark von der in "AlphaZero" verwendeten neuronalen Netzwerkstruktur. Im Folgenden werden der Entwurf der NNUE-Auswertungsfunktion und das Beschleunigungsverfahren beschrieben.

2.1 Fully-Connected Neural Network

Benutzt wird ein vollständig verbundenes (fully connected) neuronales Netzwerk anstelle eines CNNs, das die zweidimensionale Struktur eines Shogi-Bretts verwendet. Die beiden Hauptvorteile sind, dass Speicherzugriffsmuster einfach und leicht zu beschleunigen sind, und dass die unten beschriebene Differenzberechnung leicht zu realisieren ist.

Die folgende Gleichung zeigt die Berechnung des Bewertungswertes durch das vollverbundene (fully connected) neuronale Netz. Unter der Annahme, dass der Input-Vektor x ist, die Anzahl der hidden Layers L ist, die Gewichtsmatrix der Schicht l mit W_l bezeichnet wird, der Biasvektor der Schicht l b_l bezeichnet wird, und die Aktivierungsfunktion σ ist, erhält man den Bewertungswert y als

$$[y]_{1 \times 1} = z_{L+1} \quad (1)$$

$$z_l = b_l + W_l a_{l-1} \quad (2)$$

$$a_l = \begin{cases} \sigma(z_l) & (\text{if } l > 0) \\ x & (\text{if } l = 0) \end{cases} \quad (3)$$

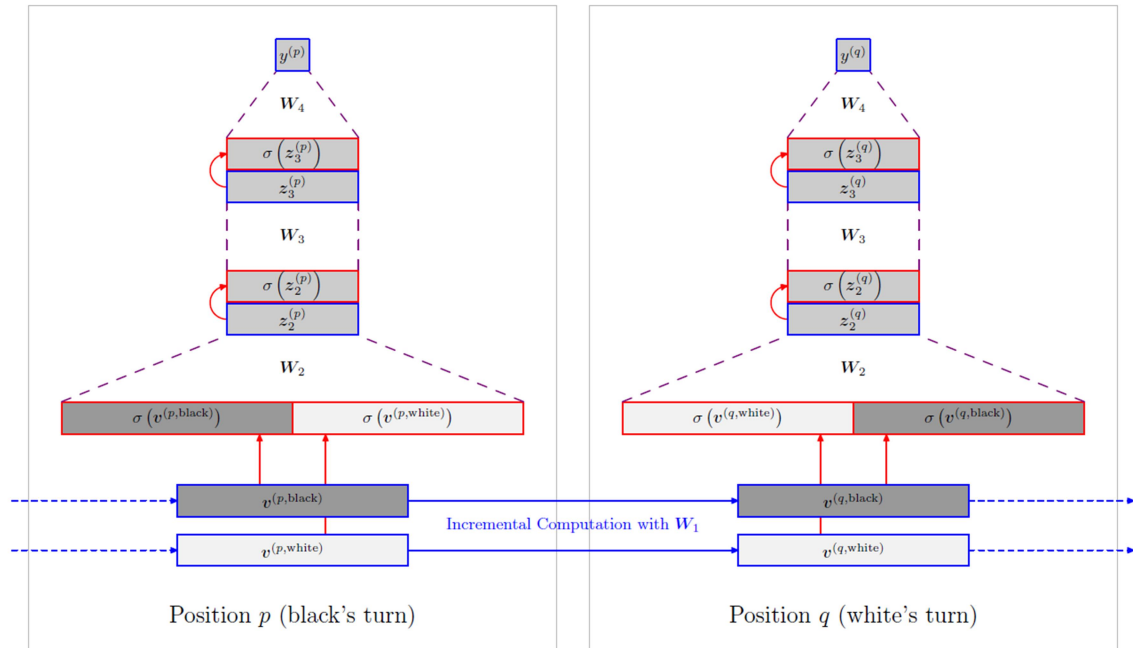


Abbildung 1: Bewertungsfunktion von "the end of genesis T.N.K.evolution turbo type D".

Der Bewertungswert wird für die Spielposition q berechnet, die einen Zug von der Spielposition p entfernt ist. Für einen Teil der Berechnung wird Differenzberechnung verwendet. Die Zugnummer wird bei der Auswertung berücksichtigt.

2.2 Aktivierungsfunktion

Für die Aktivierungsfunktion sei die Anzahl der Einheiten in Layer l bezeichnet mit N_l , und es wird die geclippte ReLU verwendet, die ausgedrückt wird als

$$\sigma(z_l) = \sigma \left(\begin{bmatrix} z_{l,1} \\ z_{l,2} \\ \vdots \\ z_{l,N_l} \end{bmatrix}_{N_l \times 1} \right) \quad (4)$$

$$= \begin{bmatrix} \sigma(z_{l,1}) \\ \sigma(z_{l,2}) \\ \vdots \\ \sigma(z_{l,N_l}) \end{bmatrix}_{N_l \times 1}$$

$$\sigma(z_{l,i}) = \begin{cases} 0 & (\text{if } z_{l,i} \leq 0) \\ z_{l,i} & (\text{if } 0 < z_{l,i} < 1) \\ 1 & (\text{if } z_{l,i} \geq 1) \end{cases} \quad (5)$$

Der Vorteil ist hier, dass der Wertebereich endlich ist, also mit Ganzzahlberechnungen realisiert werden kann, und durch die unten beschriebenen SIMD-Operationen effizient berechnet werden kann.

2.3 Feature-Vektor

Grundsätzlich wird für den Featurevektor ein binärer Vektor verwendet, bei dem jeder Eintrag entweder den Wert 0 oder 1 annimmt. Diese Einschränkung dient auch der Vereinfachung der Differenzberechnung. Um die Differenzberechnung mit hoher Geschwindigkeit durchführen zu können, ist es wünschenswert, dass die Anzahl der Einträge, deren Werte sich in den Spielpositionen vor und nach einem Zug ändern, klein ist.

Ein Beispiel für einen Featurevektor, das diese Eigenschaften erfüllt, ist die KP (King-Piece)-Kodierung, die die Positionsbeziehung zwischen dem eigenen König oder dem gegnerischen König und einer anderen Figur als dem König darstellt. Die tatsächlich verwendeten Merkmale in "the end of genesisT.N.K. evolution turbo type D" basieren auf KP und sind modifiziert. Die Details werden später beschrieben.

2.4 Affine Transformationen und Speicherlayout

Um eine affine Transformation wie die in Gleichung (2) durchzuführen, ist es normalerweise effizient, das innere Produkt der Matrixzeile und des Vektors zu berechnen; die i -te Zeile von \mathbf{W}_l wird als $\mathbf{W}_l(i, :)$ berechnet.

$$\mathbf{z}_l = \begin{bmatrix} z_{l,1} \\ z_{l,2} \\ \vdots \\ z_{l,N_l} \end{bmatrix}_{N_l \times 1} \quad (6)$$

$$z_{l,i} = b_{l,i} + \mathbf{W}_l(i, :)\mathbf{a}_{l-1} \quad (7)$$

Da sich das Speicherzugriffsmuster fortlaufend auf die Elemente der Matrixzeilen bezieht, wird das Speicherlayout als Row-Major angenommen.

Wenn der Vektor \mathbf{a}_{l-1} jedoch dünn besetzt ist (die meisten Einträge sind Null), ist es schneller, ihn nur mit einigen Spalten der Matrix wie folgt zu berechnen: Berechnen $\mathbf{W}_l(:, j)$ für die j -te Spalte von \mathbf{W}_l .

$$\mathbf{z}_l = \mathbf{b}_l + \sum_{j \in \{j | a_{l-1,j} \neq 0\}} a_{l-1,j} \mathbf{W}_l(:, j) \quad (8)$$

In diesem Fall bezieht sich das Speicherzugriffsmuster nacheinander auf die Elemente der Matrixspalten, und es ist effizienter, ein Column-Major Speicherlayout zu verwenden.

In der NNUE-Auswertungsfunktion wird die affine Transformation des Merkmalsvektors \mathbf{x} nach der Methode in Gleichung (8) berechnet.

$$\mathbf{z}_1 = \mathbf{v} \quad (9)$$

$$\mathbf{v} = \mathbf{b}_1 + \mathbf{W}_1 \mathbf{x} \quad (10)$$

Wie bereits erwähnt, ist \mathbf{x} ein binärer Vektor, und wenn \mathbf{x} dünn besetzt ist, kann die affine Transformation schnell durch Gleichung (8) berechnet werden².*1 Außerdem können wir unter Verwendung der Tatsache, dass \mathbf{x} ein binärer Vektor ist, Gleichung (8) vereinfachen durch $a_{l-1}, j = x_j \in \{0,1\}$ zu:

$$\mathbf{v} = \mathbf{b}_1 + \sum_{j \in \{j | x_j = 1\}} \mathbf{W}_1(:, j) \quad (11)$$

2.5 Differenzberechnung

In der NNUE-Auswertungsfunktion wird der Vektor, der sich aus der affinen Transformation des Merkmalsvektors $\mathbf{x}^{(q)}$ in Spielposition q als

$$\mathbf{v}^{(q)} = \mathbf{b}_1 + \mathbf{W}_1 \mathbf{x}^{(q)} \quad (12)$$

ergibt, mit $\mathbf{v}^{(p)}$ berechnet, das bereits für die vorherige Position p berechnet wurde. Konkret wird $\mathbf{v}^{(q)}$ nach der folgenden Gleichung berechnet:

$$\begin{aligned} \mathbf{v}^{(q)} = \mathbf{v}^{(p)} - & \sum_{j \in \{k | x_k^{(p)} = 1 \wedge x_k^{(q)} = 0\}} \mathbf{W}_1(:, j) \\ & + \sum_{j \in \{k | x_k^{(p)} = 0 \wedge x_k^{(q)} = 1\}} \mathbf{W}_1(:, j) \end{aligned} \quad (13)$$

Dies kann als ein Ausdruck interpretiert werden, der die Differenzberechnung des linearen Modells auf einen Vektor erweitert.

² Es ist wichtig, dass die Änderung der Feature-Vektors (Hamming-Distanz) vor und nach einem Zug klein ist, da die Differenzberechnung tatsächlich durchgeführt wird und man nicht annehmen kann, dass auch hier auf dünnbesetzten Vektoren gerechnet wird.

Die Differenzberechnung wird nur für die affine Transformation von Feature-Vektoren verwendet, nicht für andere Teile.

2.6 Gegnerpaar-basierte Berechnung

Beim Shogi hängt die Position der Figuren, auch wenn die Anordnung der Figuren gleich ist, davon ab, welcher Spieler am Zug ist. In der Sankoma-Kankei Bewertungsfunktion wird der Bewertungswert unter Ausnutzung der Linearität in zwei Terme zerlegt und als Summe aus einem von dem am Zug seienden Spieler unabhängigen und einem von dem am Zug seienden Spieler abhängigen Term ausgedrückt [4].

In der NNUE-Auswertungsfunktion wird die Zugauswertung realisiert, indem die Spielposition immer aus der Sicht des Spielers am Zug ausgewertet wird. Zunächst erklären wir eine einfache Methode. Sei $\text{active}(p)$ der Spieler am Zug in Position p und $\mathbf{x}^{(p,c)}$ der Feature-Vektor von Position p aus der Sicht des Spielers c . Dann kann die Bewertung der Position

$$z_1^{(p)} = b_1 + \mathbf{W}_1 \mathbf{x}^{(p, \text{active}(p))} \quad (14)$$

aus der Sicht des Spielers am Zug realisiert werden.

Da sich der Feature-Vektor ändert, je nachdem, ob der Spieler am Zug ist, muss die Differenzberechnung sowohl aus Sicht des ersten (schwarzen) als auch des zweiten (weißen) Spielers durchgeführt werden.

Für jeden Zug mit $c \in \{\text{Schwarz}, \text{Weiß}\}$ wird die Differenz für jeden von ihnen berechnet.

$$\mathbf{v}^{(p,c)} = b_1 + \mathbf{W}_1 \mathbf{x}^{(p,c)} \quad (15)$$

Für die Berechnung des Bewertungswertes wird der Wert des Spielers, der gerade am Zug ist, verwendet:

$$z_1^{(p)} = v^{(p, \text{active}(p))} \quad (16)$$

Als nächstes wird erklärt, wie diese Methode erweitert wird, um den Vektor des nicht am Zug seienden Spielers bei der Berechnung des Bewertungswertes zu verwenden. Sei $\text{opponent}(p)$ die nicht spielende Seite in Position p , kann die ie folgende Gleichung anstelle von Gleichung (16) benutzt werden:

$$\begin{aligned} z_1^{(p)} &= \begin{bmatrix} v^{(p, \text{active}(p))} \\ v^{(p, \text{opponent}(p))} \end{bmatrix}_{2N_1 \times 1} \\ &= \begin{cases} \begin{bmatrix} v^{(p, \text{black})} \\ v^{(p, \text{white})} \end{bmatrix}_{2N_1 \times 1} & (\text{if } \text{active}(p) = \text{black}) \\ \begin{bmatrix} v^{(p, \text{white})} \\ v^{(p, \text{black})} \end{bmatrix}_{2N_1 \times 1} & (\text{if } \text{active}(p) = \text{white}) \end{cases} \end{aligned} \quad (17)$$

Das heißt, die Feature-Vektoren des Spielers am Zug und des Spielers nicht am Zug werden jeweils affin transformiert, und der kombinierte Vektor ist \mathbf{z}^p_1 . Die Parameter \mathbf{b}_1 und \mathbf{W}_1 der affinen Transformation sind gleich, so dass diese Berechnung als eine Faltung in Spieler-Richtung interpretiert werden kann.

Die in Abb. 1 dargestellte Bewertungsfunktion von "the end of genesis T.N.K. evolution turbo type D" basiert auf Gleichung (17).

2.7 Half-KP Encoding

Wie bereits erwähnt, verwendet "the end of genesis T.N.K.evolution turbo type D" eine modifizierte Version von KP, eine Art Zwei-Figuren-Beziehung, wobei KP ein Encoding ist, das "die Positionsbeziehung zwischen dem eigenen König oder dem gegnerischen König und einer anderen Figur als der König" darstellt, während hier die Positionsbeziehung bzgl. des gegnerischen Königs nicht verwendet wird. Im Folgenden wird dieser letztere Ansatz benutzt und wir nennen die Funktion "HalfKP".

HalfKP ist ein Encoding, das mit der im vorherigen Abschnitt beschriebenen Methode kombiniert werden kann, die sowohl die Am-Zug als auch die Nicht-Am-Zug Featurevektoren verwendet.

Wenn HalfKP verwendet wird, enthält $v(p, \text{active}(p))$ in Gleichung (17) die Information, die "KP des Königs am Zug aus der Sicht des Spielers am Zug" entspricht, und $v(p, \text{opponent}(p))$ hat die Information, die dem "KP des nicht-am Zug seienden Königs aus Sicht des nicht-am-Zug seienden Spielers" entspricht.

Obwohl die Merkmale selbst keine Informationen über die Position der gegnerischen Figuren enthalten, kann daher die Information über die gesamte Position ausgedrückt werden, indem sowohl die Merkmale, die von der am-Zug-Seite aus gesehen werden, als auch die Merkmale, die von der nicht-am-Zug Seite aus gesehen werden, verwendet werden.

Der Vorteil dieser Methode ist, dass der Rechenaufwand für die Darstellung der gleichen Informationen geringer ist als bei der Verwendung des gewöhnlichen KP-Encoding.

2.8 Ganzzahl SIMD-Operationen

SIMD-Operationen (Single Instruction Multiple Data) sind Operationen, die dieselbe Operation auf mehrere Daten gleichzeitig anwenden. Die NNUE-Auswertungsfunktion ist für schnelle Berechnungen ausgelegt, indem sie ganzzahlige SIMD-Operationen in jedem Teil

verwendet. Die affine Transformation der Featurevektoren wird durch Addition und Subtraktion von Vektoren mit und ohne Differenzoperation berechnet. Die Gewichtsmatrix W_1 wird in 16-Bit-Ganzzahlen konvertiert und gehalten und dann mit dem Befehl zum Addieren/Subtrahieren von 16-Bit-Ganzzahlvektoren mit Vorzeichen (VPADDW, VPSUBW in AVX2) berechnet.

Für affine Transformationen, die keine Featurevektoren sind, werden die Aktivierung a_{l-1} der vorherigen Schicht und die Gewichtsmatrix W_l in 8-Bit dargestellt, und das innere Produkt wird mit Befehlen wie VPMADDUBSW berechnet, der die Summe der Produkte zweier 8-Bit-Ganzzahl-Vektoren berechnet.

Die Aktivierungsfunktion wird durch Befehle berechnet, die die Bitbreite eines Vektors von Ganzzahlen umwandeln (VPACKSSDW, VPACKSSWB), sowie durch Befehle, die den Maximalwert jedes Elements von zwei 8-Bit-Ganzzahl-Vektoren nehmen (VPMAXSB).

	Anzahl Spalten	Anzahl Zeilen
W_1	125388	256
W_2	512	32
W_3	32	32
W_4	32	1

Tabelle 1: Größe der Gewichtsmatrix, die in der Bewertungsfunktion von "Ende der Genese T.N.K.evolution turbo type D" verwendet wird.

2.9 Netzwerkgröße

Tabelle 1 zeigt die Größen der Gewichtsmatrizen, die in den Auswertefunktionen von „end of Genesis T.N.K.evolution turbo Typ D“ verwendet werden. Die Anzahl der Spalten und Zeilen

entspricht der Anzahl der Dimensionen des Eingangs bzw. des Ausgangs der affinen Transformation. Die Größe des Netzwerks wird so eingestellt, dass die Anzahl der Positionen, die pro Zeiteinheit gelesen werden können, dem Fall entspricht, wenn die Auswertefunktion die Sankoma-Kankei Bewertungsfunktion verwendet.

Den größten Teil der Gesamtzahl der Parameter nimmt die Gewichtsmatrix W_1 ein, die für die affine Transformation der Merkmalsvektoren verwendet wird. Obwohl die affinen Transformationen von W_2 , W_3 , W_4 nicht differenziell berechnet werden können, wird eine schnell Berechnung ermöglicht, da die Anzahl der Parameter relativ klein gehalten wird und die 8-Bit-Integer-SIMD-Operation benutzt wird.

3. Fazit

In dieser Arbeit haben wir eine Bewertungsfunktion für Shogi unter Verwendung neuronaler Netze vorgeschlagen und das Design und die Beschleunigungstechniken für den Hochgeschwindigkeitsbetrieb auf CPUs beschrieben. „the end of genesis T.N.K.evolution turbo type D“, der ersten Shogi-Software mit der auf dieser Technologie basierenden Auswertungsfunktion, wird in der 28. Weltmeisterschaft im Computer-Shogi [1] teilnehmen.

Wir planen auch, den Quellcode zur Implementierung dieser Technologie später zu veröffentlichen. Es basiert auf der Open-Source-Shogi-Software "Yaneura-Oh" [8] und ist in C++ implementiert.

Zusätzlich zu den in diesem Dokument beschriebenen Inhalten verfügt es über die folgenden Merkmale:

- Anpassbarkeit von Funktionen und Netzwerkstruktur durch Verwendung von Klassen-Templates

- Schnelle Implementierung durch kompilierte Berechnung und Optimierung durch den Compiler
- Lernen kombiniert mit stationärer Suche, ähnlich der Methode zum Lernen der Sankoma-Kankei
- Parameter-Sharing beim Lernen durch Featurezerlegung

Die in "the end of genesis T.N.K.evolution turbo type D" übernommenen Eigenschaften und die Netzwerkstruktur sind nicht unbedingt die besten. Wir hoffen, dass andere Entwickler den Quellcode erweitern und verwenden werden. Wir hoffen, dass diese Technologie zur weiteren Entwicklungen von Computer-Shogi führen wird.

Literatur

- [1] 第28回世界コンピュータ将棋選手権. <http://www2.computer-shogi.org/wcsc28/>
- [2] 金子知適, 田中哲朗, 山口和紀, 川合慧. 駒の関係を利用した将棋の評価関数. ゲームプログラミングワークショップ2003 論文集, pp. 14-21, 2003.
- [3] Kunihiro Hoki and Tomoyuki Kaneko. Large-scale optimization for evaluation functions with mini-max search. Journal of Artificial Intelligence Research, Vol. 49, No. 1, pp. 527-568, 2014.
- [4] 金澤裕治. NineDayFever アピール文書. <http://www.computer-shogi.org/wcsc24/appeal/NineDayFever/NDF.txt>, 2014.
- [5] 竹内章. コンピュータ将棋における大局観の実現を目指して. 人工知能学会誌, Vol. 27, No. 4, pp. 443-448, 2012.
- [6] David Silver et al.: Mastering the game of Go with deep neural networks and tree search. Nature, Vol. 529, No. 7587, pp. 484-489, 2016.
- [7] David Silver et al.: Mastering chess and shogi by self-play with a general reinforcement learning algorithm. [arXiv:1712.01815v1](https://arxiv.org/abs/1712.01815v1) [cs.AI], 2017.
- [8] 磯崎元洋. やねうら王オープンソースプロジェクト. http://yaneuraou.yaneu.com/yaneuraou_mini/