Online Match Prediction in Shogi using Deep Convolutional Neural Networks

Jim O'Connor[1], Melanie Fernández[1]
[1]*Department of Computer Science, Connecticut College, New London, USA*
{joconno2, mfernande}@conncoll.edu

Keywords: shogi, deep learning, classification, evaluation function, dynamic match prediction

Abstract: This paper presents a novel approach to online evaluation of shogi games using Deep Convolutional Neural Networks (DCNNs). Shogi, a complex deterministic abstract strategy game, poses unique challenges due to its extensive game tree and the dynamic nature of piece movement, including the ability to play captured pieces. Traditional methods of game evaluation for shogi rely on either expert knowledge and handcrafted heuristics, or prohibitively high computational costs and limited scalability. Our method promotes a unique dataset of shogi game records and SFEN (Forsyth Edward Notation) strings to convert board positions into binary representations, which are then fed into a DCNN. The DCNN architecture, tailored for shogi board analysis, consists of convolutional and fully connected layers culminating in a binary classification output indicating a winning or losing position. Training the DCNN on approximately one million board states resulted in an 82.7% classification accuracy on a validation set. Our approach allows for online single board evaluation, while offering a computationally efficient alternative to traditional methods, paving the way for the development of additional shogi evaluation methods without the need for extensive expert knowledge or computational resources

1 INTRODUCTION

The development of intelligent agents to play deterministic abstract strategy games, such as shogi, chess, and Go, has long been a cornerstone of artificial intelligence (AI) research. These games offer an invaluable platform for probing the limits of computational strategies and the development of advanced AI techniques. Among these games, shogi, with its rich strategic depth and complex game mechanics, stands out as a particularly challenging domain. This complexity, coupled with the unique elements of piece promotion and the ability to drop captured pieces, underscores the need for innovative approaches in AI game analysis and strategy development. Historically, the quest to master strategic games through computation has evolved dramatically, from early heuristic-based engines to sophisticated algorithms capable of achieving superhuman performance. Beginning with pioneering work such as Arthur Samuel's 'Samuel Checkers' (Samuel, 1959) and Tesauros 'TD-Gammon' (Tesauro, 1995), AI agents were initially developed to lean heavily on heuristics and simple learning techniques. These initial forays into the domain of game agents to conquer deterministic abstract strategy games were extremely successful, but relied in large part on the simplicity of the particular game at hand. Consequently, researchers were slow to expand this domain to more and more complex games. Some notable examples of this expansion were the conquering of Western chess through the development of DeepBlue in 1997 (Campbell et al., 2002), and the more recent success of AlphaGo in the game of Go (Silver et al., 2016). Although DeepBlue and other contemporary strategies relied heavily on hand-crafted heuristics by expert players for evaluation functions, the paradigm shift in game choice to the more complex domains of Go and subsequently shogi have been represented by a move into more complex computational approaches. This is clear in the current state of the art programs like AlphaZero (Silver et al., 2018) and MuZero (Schrittwieser et al., 2020), which employ deep reinforcement learning to master games of chess, Go, and shogi. These systems demonstrate the potential of AI to not only match but exceed the strategic capabilities of the world's best human players; however, this efficacy comes at a significant cost of extensive computational resources and training time.

2 SHOGI

Shogi is a deterministic abstract strategy game, sometimes referred to as Japanese chess. In shogi, two players compete on a 9x9 board (Figure 1) using a number of pieces that can be easily compared to chess and chess-variant counterparts; such as knights, pawns, and kings. Some amount of strategic depth is added through pieces such as the gold and silver generals, which can move in directions that are unique from existing chess pieces. The most meaningful rules difference in shogi for most contexts, is the 'drop' rule. When a piece is captured by an opposing player, that piece is then held 'in hand' by the player who captured it. On any player's turn, they may 'drop' a piece that they hold in hand instead of moving a piece. This recycling of the total pieces on the board dramatically expands the tree of possible game moves compared to other similar games, such as chess. This significantly larger search space is one of the primary motivations for research using shogi as a platform for AI agents.

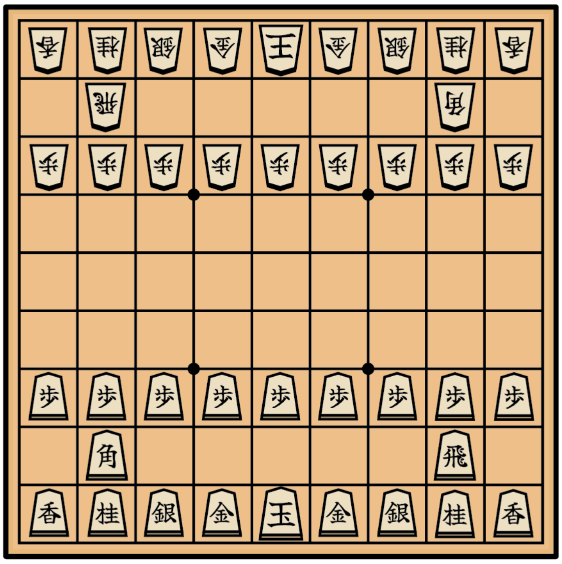

Figure 1: A game of shogi in the starting position. Shogi is played on a 9x9 board where each player starts with 20 pieces. Enemy pieces that are captured can be played again by the capturing player on their turn.

The game of shogi itself has had a long and complex history as a goal and domain for artificial intelligence research. Although the field of computer shogi has existed for decades, it wasn't until landmark research and development of Bonanza in 2007 (Takizawa et al., 2015) that a program was able to play shogi at a competitive professional level. Further advancements in computer shogi continued with the development of various additional heuristic-based methods such as those developed by Wan (Wan and Kaneko, 2018) and Grimbergen (Grimbergen, 1997), which chipped away at the abilities of the top professional players throughout the next decade. Although there was a significant amount of success and positive development during this time, these heuristic-based methods required extreme amounts of hand-tuning by domain area experts; only a handful of skilled developers in the world were able to contribute competitive and consistent results in their engines. To the contrary, later groundbreaking results in the development of a computer shogi agent came through OpenAI's innovative work on AlphaZero and MuZero, two agents that learn through self play and Deep Reinforcement Learning. These agents are able to achieve absolutely superhuman levels of performance without relying on handcrafted heuristics or expert level involvement. However, the computational and temporal costs associated with this learning technique are prohibitive. Just as the expertise-based costs of the previous approaches left a competitive agent out of reach for all but a few researchers and programmers, the new costs in compute and running time in the state of the art algorithms by OpenAI have once again left the development of a competitive agent out of reach for all but a select few. Our work focuses on a tractable and necessary part of the game playing ecosystem via single-board match prediction, and showcases a simple and effective approach with very low costs utilizing a novel dataset and the classifying power of Deep Convolutional Neural Networks.

2.1 *Match Prediction in Computer Shogi*

Within the field of AI for computer game playing, prediction of match outcomes is generally handled in gestalt via an evaluation function. An evaluation function is generally designed for each specific game or task to act as a heuristic, offering an estimation of a state's 'quality' or 'worth' in a given problem domain. In games such as shogi, AI agents begin by mapping out a game tree with feasible moves through the use of established algorithms like minimax or Monte Carlo tree search. These algorithms then leverage the evaluation function to gauge the potential success of each move within the context of the created game tree. The effectiveness and precision of an evaluation function are crucial for the performance of AI algorithms, especially in making decisions.

In the game of shogi, the role of the evaluation function in AI algorithm development is critical due to the game's intricate strategies, the expansive 9x9 board, and the distinctive mechanism of reusing captured pieces. Analyzing a board state in shogi involves considering various complex attributes like material balance, the mobility of pieces, the safety of the King, control over crucial squares, and the ability to reintroduce captured pieces onto the board. These elements together inform the score that the evaluation function assigns to a board setup. Crafting an evaluation function demands thorough knowledge of the specific problem area and often entails a trade-off between simplicity for computational speed and accuracy for optimal estimations. Although a traditional evaluation function crafted in this way can be effective for game-playing agents, the significant overhead inherent to the evaluation being tightly coupled to the agent as well as the expert knowledge necessary to craft a competent evaluation function preclude their efficacy as a simple method of match prediction.

The latest advancements in shogi AI, exemplified by systems like AlphaZero and MuZero from DeepMind, mark a departure from traditional, manually crafted evaluation functions towards those generated through self-play and reinforcement learning. These systems learn exclusively from self-play, devoid of human input, enabling the AI to discover a wide range of strategies and tactics, some of which might be unconventional yet highly effective. The evaluation functions in these models are the result of deep neural networks trained across countless games of self-play. This strategy has been extraordinarily successful, propelling these AI to surpass human capabilities not just in shogi, but also in other complex games like chess and Go. However, the computational demands of these innovative approaches have not only significantly raised the barrier to entry in the forefront of computer shogi, but also present a meaningful and unnecessary overhead to online match prediction.

3 METHODOLOGY

In the past few years, DCNN's (Deep Convolutional Neural Networks) have been commonly utilized for tasks such as image classification, object detection, and semantic segmentation in matrices. DCNN's leverage multiple layers of non-linear processing units for both low-level and high-level information processing. This Deep Learning model is a feedforward network that can be broken down into two stages; feature learning and classification. The feature learning stage of the network consists of convolutional and pooling layers that are grouped into modules and repeated according to the chosen architecture. The convolutional layers extract features from the input layer with weighted kernels and non-linear activation functions that send the outputs to the next layer. The objective of the pooling layers is to reduce spatial resolution and therefore to achieve spatial invariance. The classification stage of the network is made up of a number of fully connected layers, ending in a final softmax or equivalent function providing a classification value.

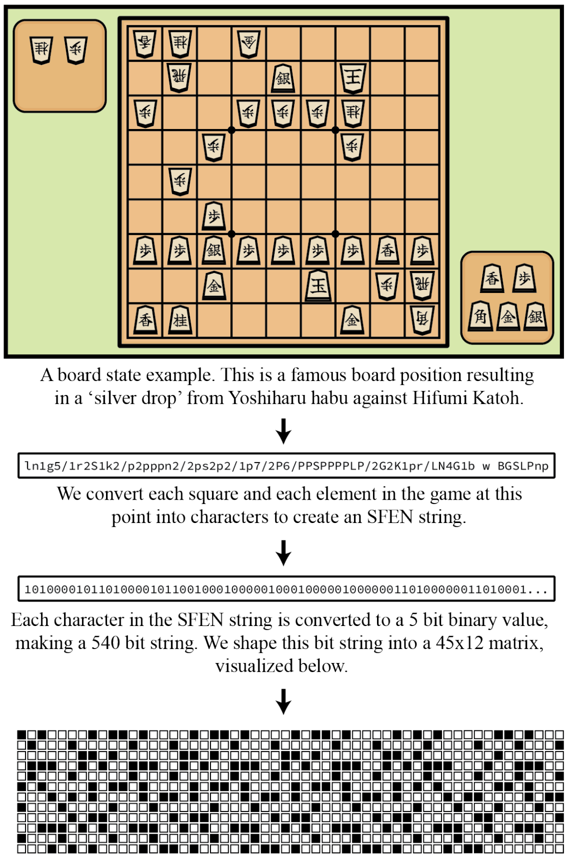

Figure 2: An example of how a shogi board state is converted to our binary matrix representation. A board state from a professional game is pulled randomly from our novel dataset. This board is then converted to SFEN, which is a standard shogi board notation. This SFEN string is then converted to a binary representation by assigning each letter to a 5 bit binary string. The binary string is then shaped into a 45x12 binary matrix, which is shown in this figure using a black and white grid.

A large number of significant advancements have been made in the field of AI through variations of this architecture. Some notable examples, such as ResNet (Targ et al., 2016) and Omnivec (Srivastava and Sharma, 2023), can rival or even surpass human level performance in image recognition tasks. Although the game of shogi does not directly present itself as an image classification problem, we present a novel approach to leveraging the power of DCNNs

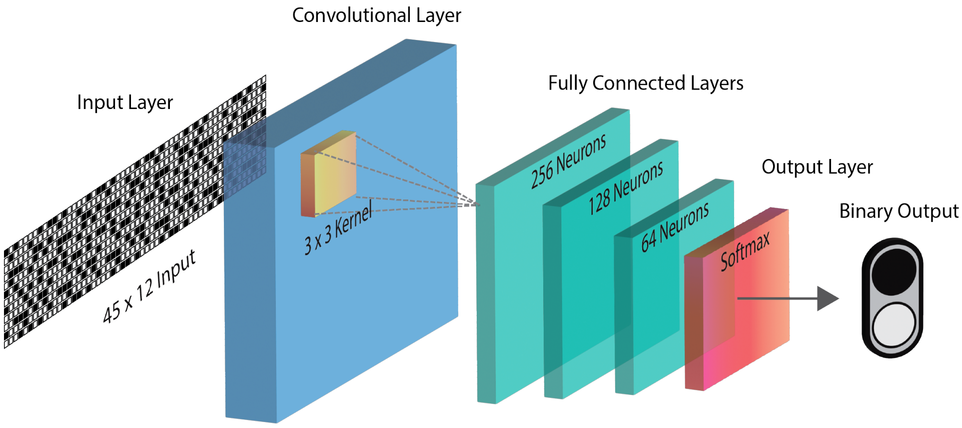

Figure 3: Our Deep Convolutional Neural Network architecture, which consists of a 45x12 binary input layer, a convolutional layer with a 3x3 kernel and 128 output channels, and 4 fully connected layers, the last of which is a single output neuron. This network classifies our binary board state representation into a single 'black will win' or 'white will win' binary classification.

to classify image as a method of evaluating a winning vs losing board position in a game of shogi. This approach hinges on both our novel dataset of shogi games and our shogi board representation that is usable by a DCNN.

The adaptation of a DCNN for shogi evaluation entails the conversion of textual representations of board positions, denoted in SFEN (Forsyth Edward Notation) strings, into binary bit strings. SFEN strings serve as a textual representation of board positions, consisting of three fields separated by spaces. The first field describes the pieces on the board; each piece is represented by a letter, and each stretch of spaces laterally is represented by an integer. So the subsection "ln1g5" would represent a row of the board filled by a Lance, a Knight, one empty space, a Gold General, and five more empty spaces. A forward slash then indicates the beginning of the next row, and the process repeats. The second field is either "w" or "b" indicating the player who is next to move. The third and final field represents the pieces in hand, with black's pieces being represented by capital letters and white's by lowercase. Our approach begins with a novel dataset composed of approximately one million SFEN strings gathered from records of professional shogi games. We convert each SFEN string to a binary representation, with each square of the board represented by five bits, and each additional metadata character of the SFEN string being represented by a further binary number (Figure 2). In total all of the information included in the SFEN string is converted into a 540 digit binary string. To convert the SFEN to binary, each piece character is assigned a corresponding 5 bit value, from 00001 to 11110. Empty spaces are represented by a zeroed five bit string, and the black and white players are represented by either 11111 or 00000, respectively. This binary string, as a tensor, is utilized as one input to the DCNN.

3.1 *Deep Convolutional Neural Network Architecture*

The Deep Convolutional Neural Network (DCNN) architecture employed in this research (Figure 3) is tailored for the analysis of shogi board states, treating them as 45x12 binary matrices akin to black and white images. As shown in Figure 3, the network begins with an input layer, where the existing 540 bit board state representation is converted into a tensor and fed to the convolutional layer. The convolutional layer has a kernel size of 3x3 and 1 pixel of padding to ensure the output feature map has the same spatial dimensions as the input. 128 distinct filters are created by the convolutional layer, which are then flattened into another tensor and fed to the four continuous fully connected layers. These layers progressively reduce the dimensionality of the feature space, eventually ending in a layer made up of a single neuron with 64 inputs, which uses a sigmoid squashing function to output our single binary classification. Rectified Linear Unit (ReLU) activation functions follow all but the final layer, introducing necessary non-linearity to the network.

The loss function we leverage is the standard Binary Cross-Entropy Loss for one output, given as:

$$-(y\log(p)+(1-y)\log(1-p)) \tag{1}$$

Where y is our labeled winner or loser, and p is the predicted probability of the label being 1. For our optimizer we leverage Adaptive Moment Estimation, or Adam, with a learning rate of 0.001. Adam, developed by Kingma and Ba (Kingma and Ba, 2014), is a first-order gradient-based method for efficient stochastic optimization. We've found that this optimizer is particularly applicable to our problem due to its effective handling of sparse data representations like ours.

4 RESULTS

Our Deep Convolutional Neural Network was trained for 20 epochs, a greater than 80% successful classification rate in all runs of the network, and a 82.7% success rate in the best case. The training data was a random 80% of 200,000 board states pulled randomly from our novel data set, and the validation data was made up of the remaining 20% of that 200,000. Our greater than 80% success rate is notable in a number of ways. Notably, this level of accuracy allows for far more successful dynamic match prediction based on a single board than in previous works. In the more tractable domain of chess dynamic match prediction, Masud et al. (Masud et al., 2015) achieved a success rate of nearly 66% under similar conditions. Related attempts at binary classification have confronted simpler problems, particularly when attempting to tackle the difficult domain of shogi. Grimbergen was able to achieve a success rate greater than 80% but on the far more manageable problem of whether or not the king was in danger, rather than determining the predicted winner of the entire game from a single board. Our promising results indicate a significant step forward in deterministic board game classification and represent a number of new opportunities for game playing agents that can be created without the prohibitive cost of standard evaluation functions seen in other state of the art programs.

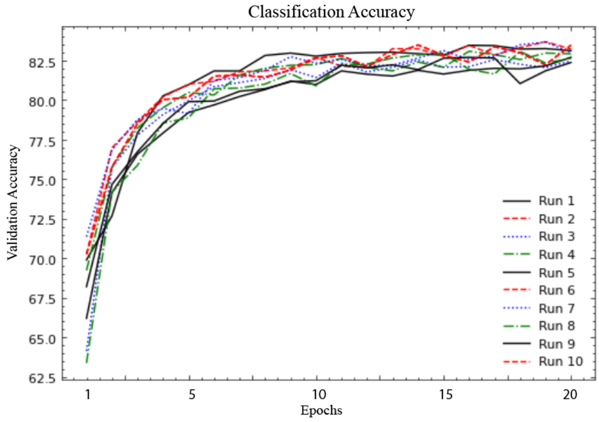

Figure 4: The resultant classification accuracy of ten runs of our DCNN classifier. These ten runs were executed sequentially with consistent parameters and random subsections of our one million board state dataset, as determined by our 80/20 training schedule.

5 CONCLUSIONS

Our results display several meaningful steps forward in the domain of classifying shogi board states and evaluating the position of a player in more efficient ways than has been shown previously. Our strategy of using DCNN-based classification allows us to give an accurate estimate towards the winner of a game of shogi without any input from a subject matter expert, instead using an online match predictor. This classification method can also be implemented and used by developers with little to no experience in the domain, due to the algorithm being agnostic of any game rules or heuristics. Additionally, we are able to make these predictions with a small fraction of the computational resources and temporal resources of other large state of the art algorithms. With our results achieving over 80% of accuracy in predicting online match outcomes, this contribution presents itself as a reasonable alternative to classification of single board states in comparison to the high computation time that other shogi engines propose. This efficiency allows us to train this classifier in a matter of minutes to hours on a standard desktop computer or laptop. Consequently, this sophisticated shogi analysis can be accessible to a broader audience that vary within skill levels. Potential research plans will explore further optimizations and applications as well as extending these techniques to other complex strategy games and enhancing their educational and competitive use.

REFERENCES

Campbell, M., Hoane, A., and Hsu, F.-h. (2002). Deep blue. *Artificial Intelligence*, 134(1–2):57–83.

Grimbergen, R. (1997). Pattern recognition for candidate generation in the game of shogi. *CiteSeerX*. Accessed: 2024-02-28.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization.

Masud, M. M., Al-Shehhi, A., Al-Shamsi, E., Al-Hassani, S., Al-Hamoudi, A., and Khan, L. (2015). *Online Prediction of Chess Match Result*, page 525–537. Springer International Publishing.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., Lillicrap, T., and Silver, D. (2020). Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., and Hassabis, D. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144.

Srivastava, S. and Sharma, G. (2023). Omnivec: Learning robust representations with cross modal sharing.

Takizawa, T., Ito, T., Hiraoka, T., and Hoki, K. (2015). *Contemporary Computer Shogi*, page 1–10. Springer International Publishing.

Targ, S., Almeida, D., and Lyman, K. (2016). Resnet in resnet: Generalizing residual architectures.

Tesauro, G. (1995). Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68.

Wan, S. and Kaneko, T. (2018). *Heterogeneous Multitask Learning of Evaluation Functions for Chess and Shogi*, page 347–358. Springer International Publishing.