GomokuNet: A Novel UNet-style Network for Gomoku Zero Learning via Exploiting Positional Information and Multiscale Features

Yifan Gao
Northeastern University
Shenyang, China
yifangao@stumail.neu.edu.cn

Lezhou Wu
Northeastern University
Shenyang, China
20184005@stu.neu.edu.cn

Haoyue Li
Northeastern University
Shenyang, China
20185541@stu.neu.edu.cn

Abstract—Since the tremendous success of the AlphaGo family (AlphaGo, AlphaGo Zero, and Alpha Zero), zero learning has become the baseline method for many board games, such as Gomoku and Chess. Recent works on zero learning have demonstrated that improving the performance of neural networks used in zero learning programs remains nontrivial and challenging. Considering both positional information and multiscale features, this paper presents a novel positional attention-based UNet-style model (GomokuNet) for Gomoku AI. An encoder-decoder architecture is adopted as the backbone network to guarantee the fusion of multiscale features. Positional information modules are incorporated into our model in order to further capture the location information of the board. Quantitative results obtained by ablation analysis indicate that our GomokuNet outperforms previous state-of-the-art zero learning networks on the RenjuNet dataset. Our method shows the potential to improve zero learning efficiency and AI engine performance.

Index Terms—Convolution Neural Network, Zero Learning, Gomoku Game, UNet, Attention

I. INTRODUCTION

Gomoku is also called five in a row. Originating from ancient Japan, Gomoku has been popular all around the world because of its relatively simple rules and moderate difficulty. Gomoku is played with black and white stones on a board with a size of 15. The black plays first, and two players take turns placing one stone of their own color on an empty place. The player who gets a row of five stones horizontally, vertically, or diagonally first can win the game.

Zero learning, mainly based on reinforcement learning, has become a benchmarking method of many board games and been widely used in training other board game AI like ELF OpenGo [1], Polygames [2], and KataGo [3], since the birth of the AlphaGo family (AlphaGo [4], AlphaGo Zero [5], and Alpha Zero [6]). Moreover, convolution neural network (CNN), as one of the most important structures of deep reinforcement learning, is turned out to have a significant impact on the final performance of the AI engine.

There are few pieces of research on Gomoku AI based on zero learning. Shao’s work [7] first put forward an effective way to predict the move of Gomoku with CNN. Other works [8]–[10] look into the neural network structures of Gomoku trained by the zero learning method in the absence of expert knowledge. However, the neural network structures in the former researches are relatively simple and cannot be used to train a powerful Gomoku AI model.

Many improvements on CNN have been employed in the game of Go. Residual networks [11] used by AlphaGo contributed the first significant performance improvement. Compared with the typical CNN, AlphaGo Zero’s performance increased by 600 Elo with the help of the residual networks. Nevertheless, further research [2] shows that in board games, the lack of location information dramatically reduces the final performance of neural networks because the fully connected layer used in AlphaGo Zero breaks the location relations. As an improvement of AlphaGo Zero, Polygames uses fully convolutional networks (FCN) to maintain the location information and achieves optimal performance in many kinds of board games. The authors of Polygames claims that learning strategies in board games tend to be similar to pixel-intensive image segmentation instead of classical image classification because moves are naturally reflected on the board. KataGo, as an open-source implementation of AlphaGo Zero, uses many innovative approaches to improve the performance of the CNN, like pre-activated residual networks, global pooling structure, and fixup technology. In particular, the global pooling structure makes the convolution layer base on the global state and further strengthens the model’s ability to capture global information.

Although former models enhance the information capture process, it is still challenging for them to fit pixel-level intensive prediction because single-scale CNNs only have limited convolutional perception radius, leading to the lack of global strategies. Previous studies on board games proved this point, for example, the “ladder mistake” in the Go AI that led to the loss of the game [1]. Therefore, we are inspired to explore CNN structures based on multiscale feature fusion. As a classical design in computer vision, multiscale feature fusion extracts low-level and high-level features through different receptive fields. It has been widely used in most computer
Fig. 1. An illustration of our proposed GomokuNet.

vision tasks and achieved advanced performance.

In this paper, we propose a novel CNN structure for the zero learning process of Gomoku game AI. Different from the previous structures based on classification models, our method is based on U-Net [12]. U-Net is a powerful CNN based on the encoder-decoder model and widely used in medical image segmentation. U-Net includes several times down-sampling and up-sampling. Thus, it can integrate low-level features and high-level features with different scales. Notably, to further capture the location information, we add the positional attention module (PAM) to the network according to the victory conditions of Gomoku. Compared with the widespread attention mechanisms, our PAM explicitly models the location information along with both horizontal and vertical directions. To be specific, the PAM integrates features along the horizontal direction and vertical direction, respectively, and encodes the feature map into the attention map sensitive to the location to improve the network’s ability to capture the information of the location.

We carry out the experiment on a human Gomoku dataset called RenjuNet to verify the effectiveness of the proposed network. The result shows that our method achieves the optimal performance compared with the previous neural network models used in zero learning.

Our contributions are concluded as follows:

- We design a novel CNN structure based on U-Net. To the best of our knowledge, it is the first time that the multiscale fusion structure has been used in zero learning.
- According to the rules of Gomoku, we put forward the PAM, which will integrate features along with horizontal and vertical directions. The location attention explicitly models the positional information of Gomoku stones and improves the performance of the model.

II. Method

A. Model Overview

In this paper, we propose a new network called GomokuNet to improve the learning efficiency of Gomoku AI. As shown in Figure 1, the basic structure of our model is a UNet-style network, including three down-sampling and up-sampling. Since the board size of Gomoku is 15, we need to use nearest-neighbor interpolation to sample it to 16x16 at the very beginning of the input and sample it back to 15x15 at the output of the policy head of the network.

B. UNet-style Architecture

The GomokuNet architecture is a UNet-style architecture that uses the strength of the residual networks and U-Net. The proposed GomokuNet takes advantage of the pre-activated residual block and the PAM. The residual block propagates information on the layer, allowing the establishment of a deeper neural network, which can solve the degradation problem of each encoder. This improves inter-channel dependency while reducing the computational cost. The proposed GomokuNet architecture contains three encoder blocks and three decoder blocks. Figure 1 shows a schematic diagram of the GomokuNet. The schematic diagram shows that each encoder block combines a residual block, a positional attention residual block, and a down-sampling layer. Each residual block
A. Dataset

In this article, the dataset used to train the deep learning model comes from RenjuNet\(^1\). RenjuNet is an online database of Gomoku containing as many as 96,000 games. All game records and related information are stored in XML format. RenjuNet includes 25 game rules, including different opening modes. Therefore, we eliminated the first six moves of each game and finally got a dataset containing 2301460 samples. We divide the dataset into a training set and a test set. The test set includes 15000 games randomly sampled and contains 355087 samples.

B. Baseline Model

In this work, we compare GomokuNet with several state-of-the-art previous works, including ConvNet, MobileNet, AlphaNet, PolyNet, KataNet.

- ConvNet: [7] designs a CNN to predict the moves of Gomoku, and it has achieved good performance in the RenjuNet database. This article uses a classic CNN, so we call it ConvNet.
- AlphaNet: We call the CNN architecture used in AlphaGo Zero as AlphaNet.
- PolyNet: We call the neural network used in Polygames as PolyNet. In the experiment, the only difference between PolyNet and AlphaNet is using a fully convolutional network.
- KataNet: We call the neural network used in KataGo as KataNet. Our implementation of KataNet does not include domain-specific improvements (auxiliary ownership and scoring goals) because such information does not exist in Gomoku.

To ensure the fairness of the experiment, we set the different networks to roughly the same depth and width, that is, 20 blocks and 256 channels.

\(^1\)http://www.renju.net/downloads/downloads.php
C. Training

All models are trained for ten epochs using the SGD optimizer with a momentum of 0.9, and the learning rate is 0.005. Our experiments were carried out on the RTX 2080Ti graphic card. The implementation of the deep learning model uses PyTorch. During the training process, the data enhancement we used included eight positions of reflection and rotation.

The evaluation criteria of each model are the Top-1, Top-5, and Top-10 accuracy of the policy network and the MSE loss of the value network. The definition of MSE loss is as follows:

$$\text{Loss} = \frac{1}{N} \sum_{n=1}^{N} (\text{predict}_n - \text{label}_n)^2 \quad (1)$$

D. Results

The comparison results are shown in Table I. It is observed that the model of capturing positional information demonstrates clear superiority. According to the experimental results, our model performs as the best on the RenjuNet dataset, outperforming the second performer (KataNet) with 1.07%, 0.94%, 0.62%, 0.044, respectively, and the model complexity is much lower (2.7 vs. 4.9). The lower complexity means that the zero learning AI engine can execute more Monte Carlo tree search (MCTS) per unit time.

E. Ablation Analysis

In this section, we performed ablation analysis on crucial modules, i.e., residual blocks, PAM, and multiscale feature aggregation method, to validate the effectiveness of our proposed model. The quantitative results of the ablation study on validation of the proposed GomokuNet are listed in Table II. It can be observed from the table that by gradually introducing residual blocks, PAM, and multiscale feature aggregation into the model, the performance of the model can be improved. In particular, multiscale feature aggregation mainly reduces the MSE loss of the value network.

IV. Conclusion and Future Work

In this paper, we proposed a novel CNN architecture for the Gomoku AI, which we refer to as GomokuNet. It introduces a UNet-style encoder-decoder network and the PAM to improve the ability of neural networks to extract features. Our network explicitly models the position information and builds the global strategy of the model through multiscale feature fusion, which is different from the existing zero learning networks. Under general settings, it achieved state-of-the-art results on the RenjuNet dataset. Ablation experiments also validate the effectiveness of the proposed method.

Although these preliminary results are encouraging, there is still much work to be done. Our network is still not trained under a complete zero learning process. The future work is to continue optimizing the network and applying it to the zero learning of Gomoku.

REFERENCES


