ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

Deep Neural Network Analysis of Go Games: Which Stones Motivate a Move?

Thomas Burwick and Luke Ewig

Frankfurt Institute for Advanced Studies, Goethe University Frankfurt
Ruth-Moufang-Str. 1, 60438 Frankfurt am Main, Germany

Abstract. Recently, deep learning was used to construct deep convolution network models for move prediction in Go. Here, we develop methods to analyze the inner workings of the resulting deep architectures. Our example network is learned and tested using a database of over 83,000 expert games with over 17 million moves. We present ways of visualizing the learned features ("shapes") and a method to derive aspects of the motivation behind the expert's moves. The discussed methods are inspired by recent progress made in constructing saliency maps for image classification.

1 Introduction

The game of Go is of ancient Asian origin and is a board game of unparalleled complexity. Due to this complexity it serves as a classical challenge for studying artificial intelligence. An obvious goal in this context is the generation of a computer Go system that uses methods which are not brute force but akin to methods used by human players. A related goal is to construct a system that learns from a database of expert games to predict the next move in such games. Recently, remarkable progress has been achieved in this direction by using deep neural network learning [1, 2]. Here, we use such learning methods to obtain a deep architecture from a database of over 83,000 expert games with over 17 million moves (section 2) similar to [1]. We then analyze the inner workings of the resulting deep architecture by adapting methods which have been developed for deep convolutional image classification [3, 4]. We demonstrate how to visualize and understand learned features of Go positions (section 3) and develop a method to determine the relevance of stones in a current position, thereby revealing aspects of the expert's motivations for her next move (section 4).

2 Move predictor from Deep Learning

The complexity of the game is related to the vast number of its possible courses. Its underlying rules, however, are simple. Two players alternatively place black and white stones on the board, starting with black. The stones are placed on the intersection of 19×19 lines; see fig. 1 for examples of resulting intermediate positions. The stones are not moved, they either remain at their position until the end of the game or are removed if they turn "dead". A set of neighboring stones (where neighboring refers to left or right, above or below) forms a "chain" that turns dead if none of the neighboring positions is empty; see fig. 2, panels A to C, for examples. The goal of the game is to occupy and surround more

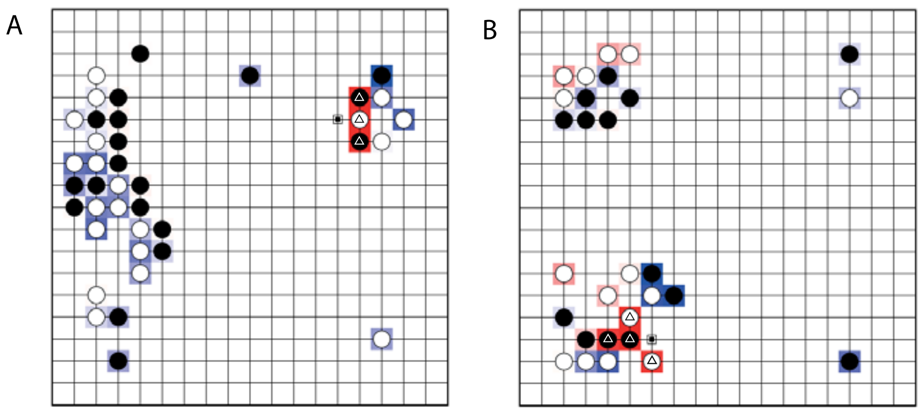

Fig. 1: Visualization of the relevances ρ_{pred} of stones for a predicted move. (A) and (B) show the 19 x 19 Go board position examples that are discussed in the text. The predicted moves (indicated with the filled squares) agree with the actual moves (empty squares). As defined in section 4, the predicition-relevances $\rho_{\text{pred}}(m)$ of the stones at positions m for the respective moves are highlighted red ($\rho_{\text{pred}}(m) > 0$) and blue ($\rho_{\text{pred}}(m) < 0$). A stronger intensity of theses colors indicates a larger magnitude of the relevance. (For non-colored prints: the darkest red regions with (A) three stones and (B) four stones are marked by the triangles, the other backgrounds are blue or light red.)

territory than the opponent and the game ends if both players "pass", that is, when they refrain from setting further stones.

For the network architecture we follow the recent work as described in [1], using an input layer, four hidden layers and one output layer, where the output units give as prediction the probability of the next move by a softmax function. For the input layer, we use $F_0 = 3$ feature maps with $(19 + P_0) \times (19 + P_0)$ units, the first for the white (black) stones if the next move is white (black), the second for the black (white) stones (with values 1 whenever a stone is at the corresponding position and 0 otherwise). The board is extended with $P_0/2 = 3$ lines on each side, realizing "padding" so that the $7 \times 7 \times 3$ filters between the input layer and layer 1 produce a 19×19 output in layer 1. For the black and white input planes, the additional components are set to zero. The third input plane describes the boundary with 0s everywhere, except for 1s at the $4 \cdot (19 + 1) = 80$ positions that surround the 19×19 board positions.

For the convolutional hidden layers $\ell = 1, ..., 4$, we use F_ℓ feature maps with $F_1 = 48$ and $F_2 = F_3 = F_4 = 32$. These have the filter sizes $5 \times 5 \times F_{\ell-1}$. Correspondingly, the zero-padded feature maps for $\ell = 1, 2, 3$ have size $(19 + P_\ell) \times (19 + P_\ell)$ with $P_\ell = 4$, assuring that the feature maps in the next layer (before padding) have size 19×19. The connections from the final hidden layer ($\ell = 4$) to the output layer ($\ell = L = 5$) are not convolutional but fully connected.

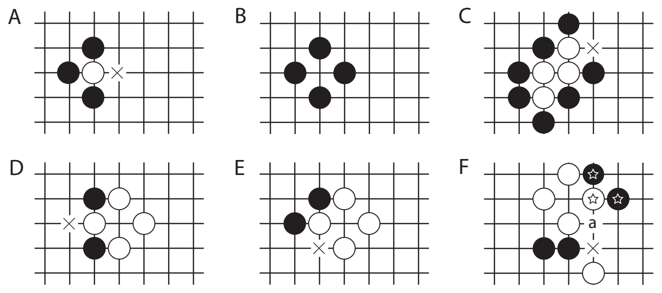

Fig. 2: (A) The white stone has only the one liberty at the marked position. Therefore, Black may capture the white stone by setting at this position and leaving no liberty of the white stone; the resulting configuration in (B) is denoted as "diamond" shape and is generally considered to be of great tactical value. (C) Adjacent stones (see section 2) form a chain that can only survive or be captured as a whole. Here, the chain of white stones has only the marked liberty and if Black places a stone there, Black may take the four white stones. (D and E) The four white stones form the diamond shape. Due to the value of this shape, a frequent next step of the black player is to attack this configuration by placing a stone at the marked position, implementing the threat to destroy the white diamond shape through capturing one of the white stones with a move as in (A) and (B). (F) This configuration corresponds to the position in the lower right corner of fig. 3 and the full-board position shown in fig. 1B. The move that would be a diamond-attacking move like the ones in (D) end (E) would be to place the black stone at position a. Here, however, the additional stones imply another motivation that lets the network's prediction (and the actual move) place the next black stone at the marked position as discussed in section 4.

We trained the network with 83,581 games, obtained from the GoGoD database[1], resulting in 17,297,161 moves. We applied the backpropagation algorithm where 88% of the moves were used for training to predict the next moves, 4% for validation, and 8% for testing the resulting network; see [1] for details on how a similar learning process is organized. Our resulting network predicts the expert moves correctly in 35% of the test cases. The expert move is among the first three of the predicted moves (that is, the ones with the highest probability) in 55% of the cases. In 64% of the cases it is among the first five.

3 Visualizing and understanding the learned features

The learning process generates filters that represent certain features. In this section, we introduce a method that allows to visualize and understand the meaning of these feature by relating them to characteristic positional "shapes".

[1]Website: gogodonline.co.uk

Let the values of feature map f_ℓ in layer ℓ, $\ell = 1, .., 4$, be given by

$$y^n_{\ell f_\ell} = g\left(x^n_{\ell f_\ell}\right) \quad \text{with} \quad x^n_{\ell f_\ell} = b_{\ell f_\ell} + \sum_{f_{\ell-1}} \sum_m \{W^{(\ell-1)\to\ell}\}^{f_{\ell-1}n}_{f_\ell m}\, y^m_{(\ell-1)f_{\ell-1}} \,, \qquad (1)$$

where $g(x) = x$ if $x \geq 0$ and $g(x) = 0$ if $x < 0$. The index $f_\ell = 1, ..., F_\ell$ indicates the feature maps and n is a two-dimensional index that describes the 19×19 board positions. For the positions m we assume a wider range that includes the additional positions resulting from padding as described in section 2. The layer $\ell - 1 = 0$ is identified with the input. Each feature map f_ℓ in layer ℓ is generated with a filter that processes the values of the foregoing layer through the W components. The convolutional character of the filters may be expressed through:

$$\{W^{(\ell-1)\to\ell}\}^{f_{\ell-1}n}_{f_\ell m} =: \{W'^{(\ell-1)\to\ell}\}^{f_{\ell-1}}_{f_\ell(n-m)} \,,$$

where "$n - m$" stands for a two-dimensional difference. The b in eq. 1 are biases. For image classification, ways of visualizing the learned features have been proposed in [3, 4]. We now use a combination of these methods, adapted to the requirements of the Go problem, to demonstrate how features represented by a filter W' may be visualized and thereby related to characteristic shapes.

As an example, we consider a feature f_ℓ of layer $\ell = 2$ that may be understood with respect to the situation illustrated in fig. 2, panels D and E. Following [3], we display the top-9 inputs that generate the strongest responses for the selected filter; see fig. 3. The representation of the feature by showing the inputs that give the top-9 largest responses is, however, not enough to understand what feature (or shape) is actually represented by the filter. Therefore, in the following we also introduce a procedure that visualizes the relevances of the stones. This procedure is inspired by the construction of saliency maps in the context of image classification [4].

The feature map $f_0 = 1$ ($f_0 = 2$) of the input, $\ell = 0$, describes the positions m of black (white) stones through values $y^m_{01} = 1$ ($y^m_{02} = 1$), while the other components vanish. Define the feature-relevance ρ_{feat} of the stone at position m for the feature map value $y^n_{\ell f_\ell}$ through removing this stone by changing the corresponding value to $y^m_{01} = 0$ ($y^m_{02} = 0$) and computing the resulting change $(\rho_{\text{feat}})^n_{\ell f_\ell}(m) = \Delta y^n_{\ell f_\ell}(m) = y^n_{\ell f_\ell}(y^m_{01} = 1) - y^n_{\ell f_\ell}(y^m_{01} = 0)$ for a black stone and analogously for a white stone. These values of $\rho_{\text{feat}}(m)$ are then illustrated trough red ($\rho_{\text{feat}}(m) > 0$) and blue ($\rho_{\text{feat}}(m) < 0$) colors at the positions m (see the caption of fig. 3), where intensity corresponds to the magnitude of ρ_{feat}.

Applying this procedure to compute the relevances of the stones for the feature under consideration in fig. 3 reveals that this feature encodes a "white diamond" configuration, formed by the four white stones of high relevance for this feature.

4 Which stones are relevant for the next move?

Applying the procedure that we described in the foregoing section to the output layer, we may determine the relevance of stones for the predicted move as follows.

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

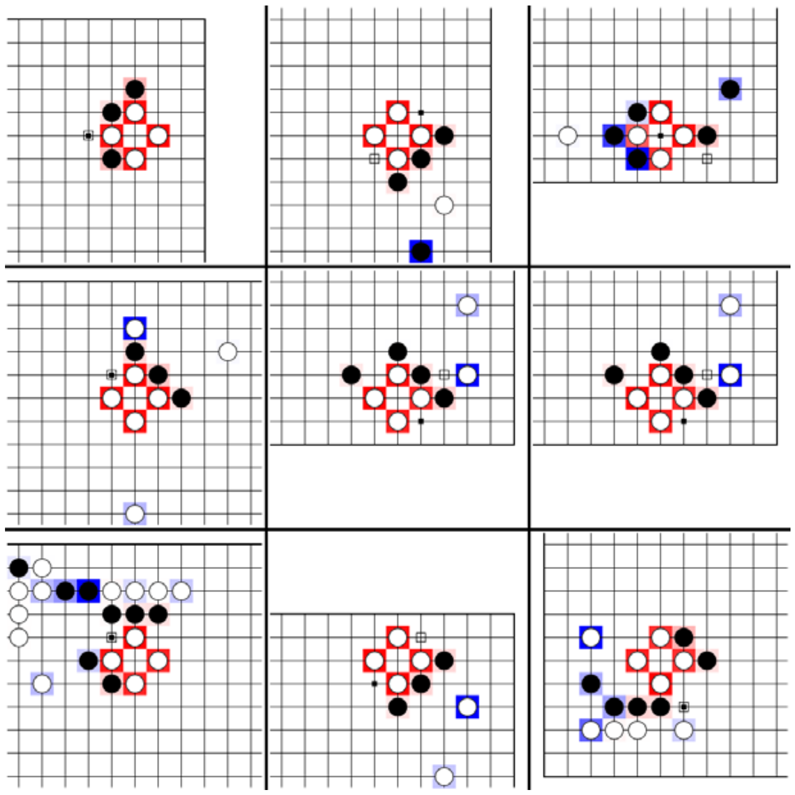

Fig. 3: As an example, we consider a feature given by a filter from layer 2 that is particularly easy to understand with respect to a black attack on the white diamond shape as illustrated in fig. 2, panels D and E. Following [3], the figure shows the top-9 inputs from the test set that generate the strongest filter responses. Notice that the size of the filters (see section 2) implies that each unit of layer 2 receives input from 11×11 board positions, eventually reaching outside the board's boundary if the receptive field includes a part of the padding frame. The colors, constructed with the method presented in section 3, allow to visualize the essence of the considered feature through displaying the feature-relevances ρ_{feat} defined in section 3: red (blue) indicates that the filter response decreases (increases) if the stone is removed. This relates the feature to the white diamond shape. (Hint for non-colored prints: only the four stones of the white diamond shape show dark red, other backgrounds are blue or light red.)

Let us denote the output values as z_n where n is the two-dimensional index that indicates the 19×19 board positions. The probabilities for the next black move at one of the empty board positions n is given by the softmax function:

$$p_n = \frac{z_n}{Z} \quad \text{with} \quad Z = \sum_{n'} z_{n'} \quad \text{and} \quad z_n = \exp\left(x^n_{Lf_L}\right) \ , \tag{2}$$

where the definition of the $x^n_{Lf_L}$ is obtained from eq. 1 with $\ell = L$.

We now define the prediction-relevance ρ_{pred} in analogy to the feature-relevance ρ_{feat} of the foregoing section as the effect of removing a stone on the probabilities p_n. The resulting changes $\rho_{\text{pred}} = \Delta p_n$ may then be color coded as illustrated for the examples in fig. 1 with red (blue) for $\rho_{\text{pred}} > 0$ ($\rho_{\text{pred}} < 0$) and color intensity indicating the magnitude of ρ_{pred}.

To understand the relation of the feature- and the prediction-relevance, consider again the top-9 filter responses in fig. 3 and concentrate on the upper left (case A) and the lower right (case B) positions. The corresponding board positions are shown in fig. 1A and 1B. For both cases, as indicated by fig. 3, the white diamond shape is “recognized”, expressed through a large filter response, but only in case A does it imply a diamond-shape attacking black move (as for most of the other situations in fig. 3). What could motivate the different move in case B? (Of course, the notion of “motivation” has several connotations. Here, we want to restrict its meaning to the described relevance measure.)

The prediction-relevance ρ_{pred} shown in fig. 1A confirms that indeed the white diamond shape is relevant for the attacking move. In contrast, a different motivation shows up in case B; see fig. 1B. There, it is not the white diamond that is relevant for the predicted (and actual) move but instead a shape that is shown in fig. 2F. The black move at the marked position serves to “cut” a possible connection between the lower white stone and the diamond.

Note also that the negative relevances ρ_{pred} of the white stones on the left board side in fig. 1A indicate that there are tactical necessities in this region that compete in relevance with the diamond-attacking move. Therefore, the prediction of the latter increases when removing white stones on the left side as this lowers the necessity to act there. The analog remark applies to the star-marked stones in fig. 2F that have a negative relevance in fig. 1B.

References

[1] Christopher Clark and Amos Storkey. Training deep convolutional neural networks to play go. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1766–1774, 2015.

[2] Chris J. Maddison, Aja Huang, Ilya Sutskever, and David Silver. Move Evaluation in Go Using Deep Convolutional Neural Networks. In *International Conference on Learning Representations.* 2015.

[3] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer, 2014.

[4] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.