MineSweeper report

Yuchen Zhou, Wei Li

December 2020

1 Introduction

We trained a Minesweeper agent in a supervised learning fashion with a deep neural network instead of reinforcement learning. Our results show that the model achieved over 50% winning rate in a 6x6 Minesweeper grid with 6 mines within 10,000 iteration gradient updates. More importantly, loss decreases and accuracy for each action increases stably during training.

2 Related work

Determining a mine configuration satisfying a board constraint has been proven to be a NP-complete problem [1]. To fully estimate the mine probability distribution of mines, all possible configurations need to be found. This has been proven to be a class of #P-complete problems [2].

Among machine learning supervised learning of local classification, global probability regression, simplified Q-Learning, Constraint Satisfaction Problem (CSP) [3], Q-learning achieves the highest winning rate of all with 70% on a 4x4 board. As the board size increases, the performance of the three algorithms decreases, with <10% on 5x5 boards and <5% on 6x6 boards. However, The CSP method achieves 80% winning rate on 8x8 boards and decreases slightly when board size is increased to 32x32 with approximately 70% winning rate. To our acknowledge, [4] has achieved the state of art winning rate of 90.2% on 6x6 board with six mines by Deep Q-Learning.

3 Methods

3.1 Supervised Training

We decided to use supervised training based on a single observation: there is no bad local optimum decision in early game. Comparing to reinforcement learning, instead of estimating the values of each potential states, supervised learning only needs to make a local optimum decision. We designed a Deep Neural Network (DNN) with a lifting function from one-hot encoding, a residual Convolutional Neural Network (CNN) layer with batch normalization and ReLU. In addition,

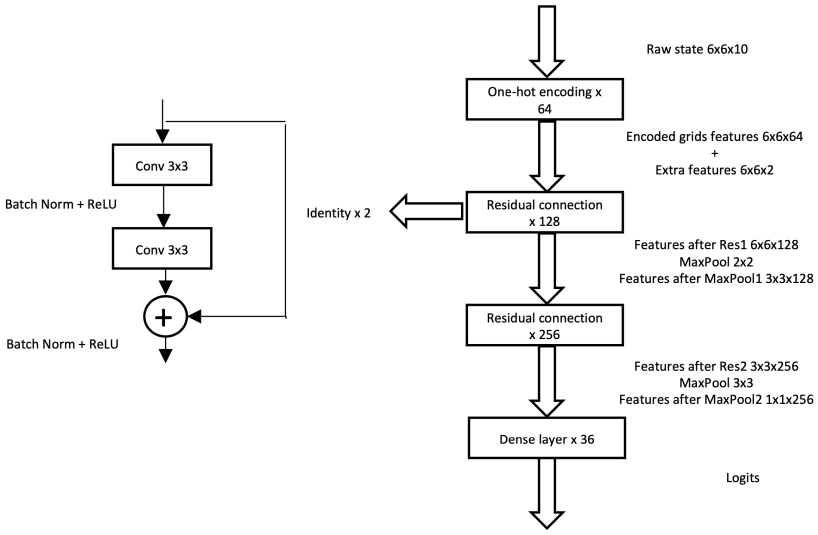

Figure 3.1: Supervised training Minesweeper prediction model architecture

supervised learning has a more stable loss. Figure 3.1 shows the architecture of the model. The model estimates a probability distribution of the most likely successful next move.

3.2 Loss

A cross entropy target function is used as the loss for the model. A uniform distribution of all correct next step actions is provided as the label for the cross entropy. The accuracy per action is used to evaluate the validation of the loss.

4 Experiments

4.1 Benchmarks, Data Generation and Feature Encoding

A 6x6 grids Minesweeper with six mines is chosen to be the configuration of the test and winning rate is being estimated to evaluate the model.

In order to supervise train the model, 1 million different Minesweeper game states with corresponding correct next move labels has been generated by approximately 330,000 different game initializations.

A global feature of ratio of the number of mines to grid size and local feature of scaler of quantity of mines are encoded as a two-dimensional vector of each grid, the features are added to the one-hot encoded feature.

4.2 Results

To maximize local optimum decisions for every single action, cross entropy with uniformed correct next move probably distribution is used to gradient descend the parameters. Figure 4.1 shows the validity of the model.

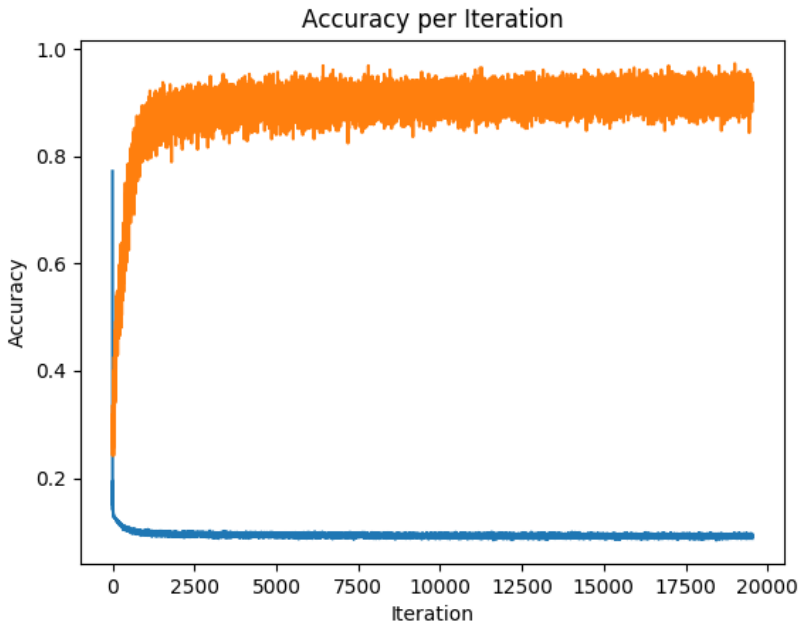

Figure 4.1: Loss and accuracy versus gradient updates iterations

The model is finally being tested by playing the Minesweeper game with 6x6 grid size and 6 mines. We report a 62% winning rate with approximately 20,000 update iterations over a batch size of 256 and a 58% winning rate with approximately 12,000 update iterations over the same batch size.

5 Future work

The winning rate at 20,000 update iterations and 12,000 update iterations, and loss figure shows that this model has a potential to achieve a better winning rate. We believe that decreasing the learning rate after certain epoch or update iteration also helps to achieve a better winning rate.

We used a uniform distribution of the correct next step action to estimate the whole population. We further used this distribution as the label in cross entropy loss. This means we used a sample of size one to estimate the whole population. The model can be improved by a better estimation of the population

distribution.

Furthermore, other experiments setups can be tried on the model. For example, we can adjust the board size and number of mines. We also want to test whether the feature encoding helps to increase the winning rates.

6 Conclusion

Due to the property of Minesweeper: an early game optimum decision would not lead to bad game state. This means a supervised learning, which predicts a grid with the lowest probability containing a mine, could fit to this game. Additionally, a supervised learning method could lead to a more stable loss descend comparing to Reinforcement learning.

References

[1] Kaye, R. (2000). Minesweeper is NP-complete. The Mathematical Intelligencer, 22, 9-15

[2] Nakov, P., & Wei, Z. (2003). MINESWEEPER, #MINESWEEPER

[3] Gardea, L., Koontz, G., & Silva, R. (2015). Training a Minesweeper Solver

[4] Hansen, J., Havtorn, J., Johnsen, M., & Kristensen, A. (2017). EVOLUTION STRATEGIES AND REINFORCEMENT LEARNING FOR A MINESWEEPER AGENT