Using CNN to solve Two player zero-sum games

Dawen Wu

Sep 2021

Introduction

Background

- A two-person zero-sum game can be represented by a matrix, e.g.,
$$A = \begin{bmatrix} 1 & 7 & 5 \\ 2 & 4 & 3 \\ 3 & 10 & 6 \end{bmatrix}.$$
- One of the classical problems is to solve $v = \min_y \max_x x^T Ay$, where v denotes the saddle point value and (x^*, y^*) denotes the saddle point strategy.
- This problem can be solved by linear programming.
- The famous minmax theorem guarantees the existence of a saddle point.

Problem formulation

We consider the matrix A to satisfy the following,

- The shape of A is given and fixed.
- The elements of A come from a given probability distribution.

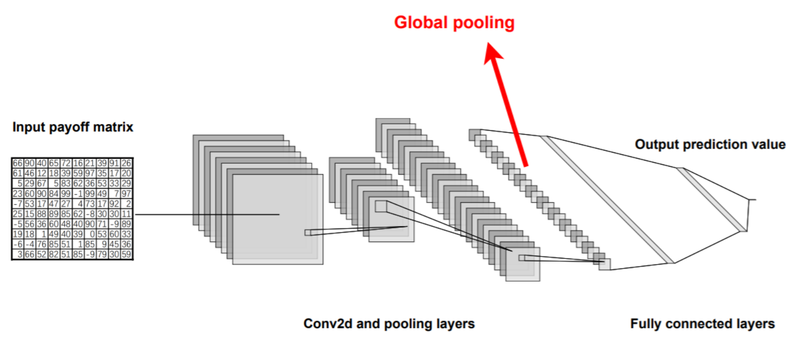
We use a convolutional neural network (CNN) approach to find an estimate $\hat{v}$ and use a linear system to derive $(\hat{x}, \hat{y})$ in the corresponding $\hat{v} = \hat{x}^T A\hat{y}$.

We evaluate the CNN method results by 1) Accuracy: The predicted $\hat{v}$ is expected to be as close as possible to the real v. 2) Efficiency: The CNN method is expected to be faster than the classical linear programming method.

Method

CNN method to predict saddle point value

1. Generate a certain number of matrices A according to the given shape and probability distribution. Use linear programming to find v. Multiple (A, v) together form the training set.
2. Construct a CNN model in which the number of input channels is 1, like the following,



3. Go training.

Linear system to obtain saddle point strategy

After the CNN predicts $\hat{v}$, the remaining problem is how to get (x, y) in $\hat{v} = \hat{x}^T A\hat{y}$

Solving the $\hat{v} = \hat{x}^T A\hat{y}$ problem can be converted to solving down the following linear system problems, where n, m denote the size of x and y.

$$\begin{aligned} \mathbf{A}^T\mathbf{x} &\geq \hat{v}\mathbf{e_m} \\ \mathbf{x}^T\mathbf{e_n} &= 1, \mathbf{x} \geq 0, \end{aligned} \tag{1}$$

$$\begin{aligned} \mathbf{A}\mathbf{y} &\leq \hat{v}\mathbf{e_n} \\ \mathbf{y}^T\mathbf{e_m} &= 1, \mathbf{y} \geq 0, \end{aligned} \tag{2}$$

- when $\hat{v} = v$, both (1) and (2) are feasible.
- When $\hat{v} < v$, only (1) is feasible, and the corresponding $\hat{y}$ is the best response of $-A^T\hat{x}$.
- When $\hat{v} > v$, only (2) is feasible, and the corresponding $\hat{x}$ is the best response of $A\hat{y}$.

Experimental results

We consider the game size to be $1000 * 1000$ and a uniform(-10, 100) probability distribution. Compare the CNN method to linear programming(LP), fictitious play(FP) and EXP3 methods.

	1000*1000		
Algorithms	CPU Time	Value	Gap (%)
LP	1.3035	45.0293	-
CNN	0.00086	45.6481	1.3553%
FP	259.3845	47.6020	5.4043%
Exp3	0.7100	45.0431	0.0304%

Table 1: Comparison between LP, CNN and two learning algorithms

References

[1] Dawen Wu, Abdel Lisser. Using CNN for solving two-player zero-sum games. 2021. ⟨hal-03341813⟩

[2] G. B. Dantzig, Linear programming and extensions, Vol. 48, Princeton university press, 1998.

[3] P. Fischer, A. Dosovitskiy, T. Brox, Image orientation estimation with convolutional networks, in: German Conference on Pattern Recognition, Springer, 2015, pp. 368–378