AI Project report

Hanyu Hu

May 2018

1 Abstract

For the simple AI, it can accomplish some simple tasks. For my Gaming AI, it is something that can play simple ball game. The main goal for the AI is just collecting the falling ball with the movable bar. My Gaming AI has it's own sensor, brain and Motion handler for itself. They are working cooperate to playing this ball game. It can implement to other field, for example, this AI could be a robot. Overall, the prediction AI(Take a prediction and react, let's call it prediction AI) is not only using in the gaming area, in recent days, AI has been a really popular topics, different kinds of AI have different methods, which Alpha GO will be the most well-known AI. So, I think prediction AI can also be using in some weather condition check, Missile launch, space shuttle launch, planet moving areas, which can predict many things that has not happened but to be happening. The prediction AI can detect the things about happening and make action by its "brain", which can prevent some "worse" cases happen. In the AI report "How We're Predicting AI—or Failing To", in it's introduction mentions that in an database with 257 AI predictions made in a period spanning from the 1950s to the present day, 95 of those cases are considered as AI time-line prediction, since this is research paper from 2012, we are still able to make about 40 percent predictions that are considered valid, but now we have pretty much complex and more accurate Algorithms (like Deep Learning skills) and huge database, which can make AI prediction reliable. Not like the Neural Networks, mine program prediction will only depend on really shallow observe, what it see, just do what it need to do at first sight. in my AI project, there's no learning skills for AI, but it has plasticity to be like that, what I do, actually just ask my to reply on the first layer of neural networks.

2 Introduction

2.1 Game introduce

2.1.1 Overall

The game is based on two objects, a ball and a bar. The player can control the bar by using the keyboard to move left and right. This can pretty much shows on a user interface (see Figure 1), the white cube is which we call it "ball" and the little triangle lays on the bottom is the "bar". The game is simple, we need to control the bar to receive the ball, once the ball touch the ground, you will loose the game, and the terminal will shows your score and terminate your game with saying "you loose the game". Same to AI, if AI loose the game, there will show loosing signal "AI loose the game". After several games, I will collect the data from human's performance and AI's performance, with comparison, we can have a conclusion which only on behalf of the game performance.

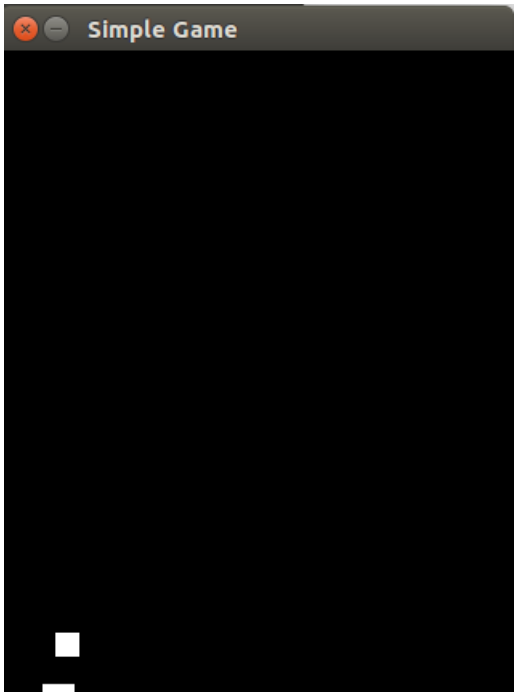


Figure 1: Gaming interface

2.1.2 Getting start

For the AI game, you do not need to take any actions, what you need to do is open terminal in target direction, type "Python3 Project.py" (be careful, only

python3 can compile the game, since python previous vision has no environment for pygame header), then the terminal will say "choose the mode", just type "AI" if you want to get into the AI game, or you can type "self" for playing by your self (see Figure 2).

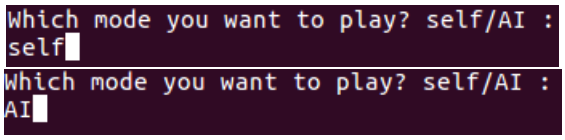


Figure 2: Terminal

Once you get into "AI" mode, any keys you type will not be able to be received. AI will play the game by it's prediction. But you can press "Esc" key to terminate the AI, and stop the game, (Since the last time I test, it reached 500 scores, I will default that AI will not loose the game, so there have to be something can terminate the game). Another way to terminate the game is clicking the cross on the top left window with your mouse. Need attention, once you type "self" mode, and press enter, the game will start immediately, give yourself enough time to prepare the game. Since when I collect data, I will need data accurate, and that will not allowed some accident, if loose the game without prepare, the data will be invalid. In my python code, they also named as "sensor", "brain", "keypress"

2.2 AI introduce

The AI I produced can give reaction based on the environment changed. It can detect the position of the ball, and predict the position where it will land, then moving the bar to the target address waiting for the ball falling down. Compare with some robots, we know robots has the sensor, which can detect the surroundings, and has the moving tools. Also the robot will have a processor, the data collect by sensor will send to the processor, so it will send to the action instruments in order to make reaction. For my AI, it adopt this concept too. My gaming AI has it's "brain", which correspond to the processor we mentioned in the robot. It has "hand", which can type on the key board in order to make bar moving left and right. It has "eyes" which can detect the position of ball and bar, and return the position value to the brain.

3 Background

For prediction AI, there are many uses in many fields. When reading the article "Chemists harness artificial intelligence to predict the future of chemical reactions", it mentioned "In an ideal world, chemists would like to predict which combination of chemicals would deliver the highest yield of product and avoid

unintended by-products or other losses, but predicting the outcome of these multi-dimensional reactions has proven challenging." (paragraph 2.) Which we can see that for prediction AI or some prediction in some industry is really necessary and helpful. the chemical medicine from past hundred year. human can only testing the efficacy on the mouse and then make experiment in the human body. eventually which may cause some kills, and following by many ethic problems. But if some prediction can be applied, and which are accurate and reliable. We still need the experiment, but the experiment risk will go down. In fact, there have been bunch of scientist has working on this, aiming to make the experiment or some projects low-cost, reliable and efficient.

4 Problem description

The main goal is let AI predicate with high accurate, and moving without error. After that, let human being versus my AI, and collecting the data for the futher analysis.

4.1 Early period

Moving Method:

At early period, when I am checking the game, the first thought in my mind was just make the bar has it's sensor, brain, motion handler, and let them help the bar track the ball's moving, so bar's moving will have a principle.

4.2 Mid term period

Prediction implement:

when things go well in the early period, I found the problem that, the moving speed of ball is depending on the speed on the x coordinate and y coordinate speed, and combining with the heading direction, so that we can get the moving motion for the ball. The issue is, when we tracking the ball, the bar is moving left and right, which has nothing to do with prediction, the bar only need to move left and right with speed of x coordinate of the ball. So, the early project looks stupid, only has the same speed move left and right which we are not expect to see. so, next problem I solved is create new entity, which can predicate the moving path of ball.

4.3 Late Program

Improve Accuracy:

After creating the moving predicting method, there are still some little issues, the accuracy, in the AI that in beginning can only gets 50 scores and can never

reach higher, that is because every time predict the moving path of the ball, the predicated landing position has deviation, it not occurred hard each time bar receive the ball, but it will loose the tracking accuracy path after score 50. And there's no boundary for bar moving, in another words, when you keep pressing left or right key, it may move out of the interface, so does AI too, when I did not set the boundary and some moving principles, it can move out of boundary, which will not able to coming back into interface.

5 Approach

5.1 Game and Code introduce

In the original version of the game, it has three main part. 1. bar moving function, 2 ball moving function and 3. Key press event. As we know I as human individual, we have the ability to predicate the game with our own time line. for the bar moving, The function "bar-move-left()" and "bar-move-right()" takes care of it. which simply plus or minus the bar's x coordinate. and key press event, I am using pygame header, which can download online, they provide the key press event function, for press left key, we call the function "bar-move-left()", and so does right side. we are not able to press other keys which I did not implement. For the ball moving, I gave it a random position as initial position, and also gave it an starting direction. Along with the program time line, the ball will moving with a constant speed, which I set x speed is 2 and y speed is 3. Since the boundary set as "SCREEN-SIZE", so when ball reach the end of the screen, I will need to reflect the direction of the ball. To achieve this, just set condition when it is greater than the boundary, if it is left and right boundary, x direction times negative one, if it is top or touch the bar, the y direction times negative one. Pygame has set almost everything for moving method, which I had to say really convenient.

Two ways to terminate the self playing game, you can either click the cross on the top left, otherwise, only when you loose the game, the game will be quit, and shows your loosing message on the terminal.

Each time you catch the ball, the score will be on the terminal, and it will add one.

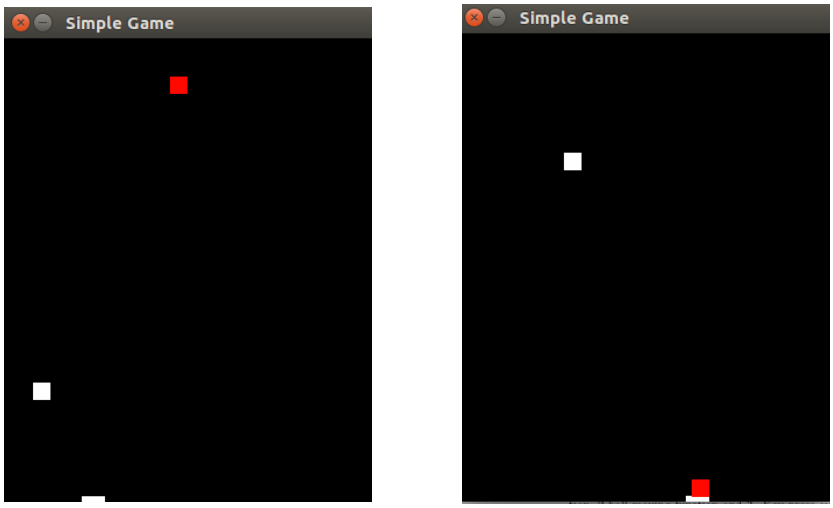
If while you are playing, you feel the game is too hard due to the size of bar size, then terminate the game first, and modified the Global variable "BAR-SIZE", which I initial width 20 units and high 5 units. But you are welcome to re-fix that so, Same as your play interface size, and ball size, they are corresponding to "SCREEN-SIZE" and "BALL-SIZE", if you also want to change the color, you are welcome to do so. I have given three colors, they are "BLACK, WHITE, RED", feel free to add more colors, but you have to implement by your self with RGB table.

For the "AIGame()", the basic moving principles are same as ordinary "Game()", But the differences are "AIGame()" has only one key press event, who uses for terminate the game. Added "left-key-press()" and "right-key-press()". they are pretty much like key press event, instead of pressing the key, when function "left-key-press()" was called, the "bar-move-left()" would be called. same as right side. even this looks odd in this program, but in many cases, when AI needs to accomplish more complex task, it's own motion handler will be necessary.

5.2 Prediction Code introduce

In "AIGame()" the another difference is "AIGame()" has an new entity, which is called "prediction-ball", for this entity, it has the same size as ball, in my approaching, I am just creating new entity as ball, but moving speed is triple times of original ball, so that prediction ball will reach the bottom before the original ball. Add a Digression on some fields, like the rocket launch, in some particular ranges, that we can predict, using prediction by speed up a virtual rocket to predict the target place is useful, so that is where I got my idea. So basically, the prediction ball has same moving principles as ordinary ball, only when ball's x speed is 2 and y speed is 3, prediction ball's x speed is 6 and y speed is 9 (see Figure 3.1a.). Further, one thing that also different as original ball is prediction ball will easily touch the ground, instead of loose the game, we will record that position, the prediction ball stop at where it is, and moving our bar to that place, which we can tell that is the "predicated position"(see Figure 3.1b.), after the ball touch the bar, we get the signal, and start to prepare next prediction, and set the prediction ball's speed back. If you want to visualize the moving of prediction ball, you can add "pygame.draw.rect(self.screen, RED, self.predict-pos)" (Hint: which means the prediction ball is red cube) in the line 170 (in the Project.py), which I had commanded out, since we do not need to see the prediction while we looking the AI play, only AI knows that is enough. So, instead of tracking the ball in the early period, now we are tracking the prediction ball.

But when setting high speed to prediction ball, and let our bar to track the prediction ball. The accuracy will loose really fast, as we mentioned in the problem description, when speed goes fast, once the prediction ball touch the ground, somehow, it may get "into" the ground, but as we know, when ball touch the bar, the bottom of ball is the height of the bar, it will not go deeper than the top of the bar, this is not applied to prediction ball, because the prediction ball will not reflect by bar. Just because the accuracy loosing, that cause the score can not be higher than 50. So, in case of keep prediction accuracy, every time the original ball touch the bar, I reset the prediction ball's position, which means it will launch together with original ball. in the code I'm using "self.predict-pos.center = self.ball-pos.center", to set the position of prediction



(a) 1a (b) 1b

Figure 3: Prediction Method

ball and original ball together.

5.3 AI brain, sensor, motion handler

5.3.1 AI brain

In "Brain", AI can make three decision, if prediction path is on the left part of bar, then move the bar to the left (left-key-press()). If the prediction path is on the right side of the bar, then move the bar to the right (right-key-press). the last condition is little more complex. since if the bar has reach the target address, we would not want to press any keys. Under this case, I create the function call "no-press()", which will move the bar. But as we know, we have accuracy problem, the bar is tracking the prediction path. if just stop at the point where prediction predict, once the prediction path has strong deviation occurred, then we will be not able to receive the ball anymore (this is also one reason that the score in the early game can't be higher than 50). In case of that, I make an accuracy value, which based on the speed of the bar moving speed by one press, if we are looking into the "bar-move-left/right()" function, the line "self.bar-pos-x = self.bar-pos-x - 2" can tell that bar's speed is 2 units/press, so I just set the accuracy number to 2, so that if the prediction ball's x position and bar's x position is within 2 units, call function "no-press()", so that we can make sure the bar can receive the ball, so in the brain, we have to know the prediction position and it's accuracy too.

5.3.2 AI sensor

In "sensor", which actually is very easy, detect the ball's position, and detect the bar's position, return them as a binary set. Then sent the data to the brain (which simply call this function in the brain.). For this project, the sensor is simple, that is because the environment changing is simple, and predictable, but if something in the environment is unpredictable, then if will need some comples sensor to get know with it.

5.3.3 AI key press

In "Key-press()" function, basically just switch on/off the moving status, there are two moving status, move to right, and move to left. if both set to false, then AI can know this is "no move". either on can be switch on, but can not both turned on, otherwise the bar will moving infinity left and right.

6 Testing

6.1 Human Play

I arranged four people play the game, include myself, and make them into a table.
In one minutes:
Hanyu: Score: 12, terminate game with loose.
Fancy: Score: 15, terminate game with no loose.
Xin: Score: 15, terminate game with no loose.
Xicun: Score: 8, terminate game with loose.
In three minutes:
Hanyu: Score: 16, terminate game with loose.
Fancy: Score: 13, terminate game with loose.
Xin: Score: 28, terminate game with loose.
Xicun: Score: 12, terminate game with loose
In five minutes:
Hanyu: Score: 10, terminate game with loose.
Fancy: Score: 10, terminate game with loose.
Xin: Score: 32, terminate game with loose.
Xicun: Score: 14, terminate game with loose.
No time limits:
Hanyu: Score: 20, Game loose.
Facny: Score: 35, Game loose.
Xin: Score: 17, Game loose.
Xicun: Score 5, Game loose.

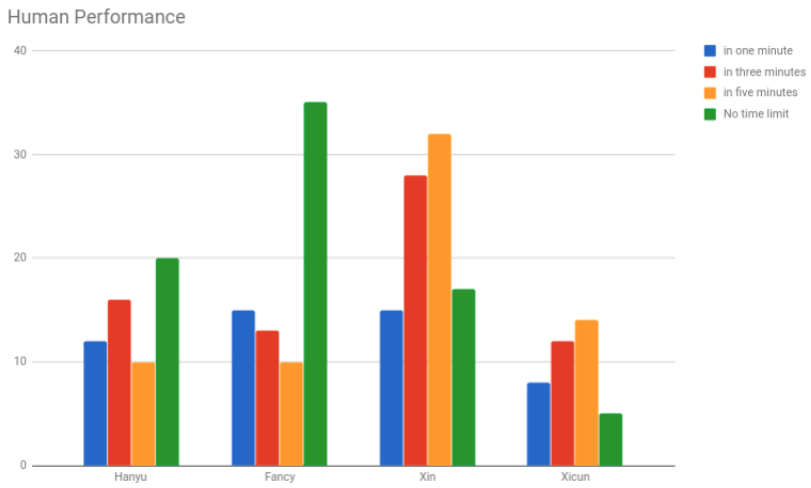


Figure 4: bar graph

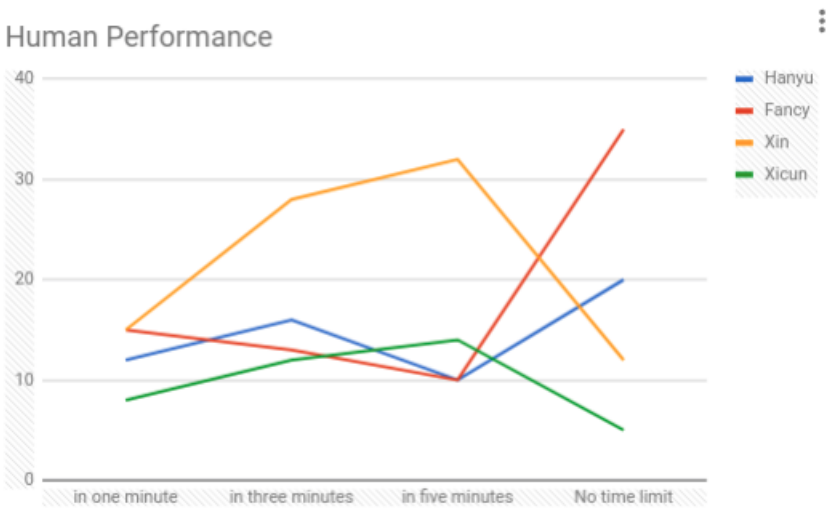


Figure 5: linear graph

6.2 AI Play

In one minutes:
Score: 15, terminate game with no loose.
In three minutes:
Score: 45, terminate game with no loose.
In five minutes:

Score: 75, terminate game with no loose.
No time limits:
Score 289, terminate game with no loose. press "esc"

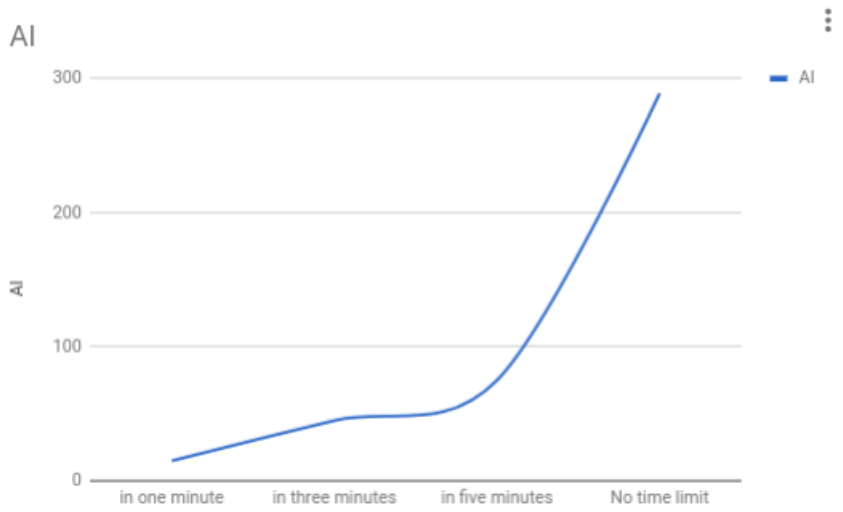


Figure 6: AI's performance in timeline

6.3 Data Analysis

As we see, in the figure7. the performance of AI is keeping increasing, so that I can assume, if there's no time limit, the AI will keep playing and get more scores. the prediction of human is limited as we can see in figure 5, different people have different performance. Like me (Hanyu), you can see my score is stabilise between 10-20 scores, I think that is my limit, since when you playing this game, you will feel tired and some other external factors. but like Fancy, she did better in the long term game, and Xin did worse in the late game. As we looking into AI's game, checking figure6, overall, it is keeping increasing, give it more time, it will get more scours. Express its increasing in function should be F(x) = 15 * time(minutes). since in one minutes the AI can get 15 scours. Comparing with human's data, it is unpredictable and not able to have a formula to calculate next performance, so that makes the data unreliable for the future.

7 Result

Human will affect by many external factors, will tired, will loose focus. But AI not, it just keep playing and playing. So compare with human's prediction, a

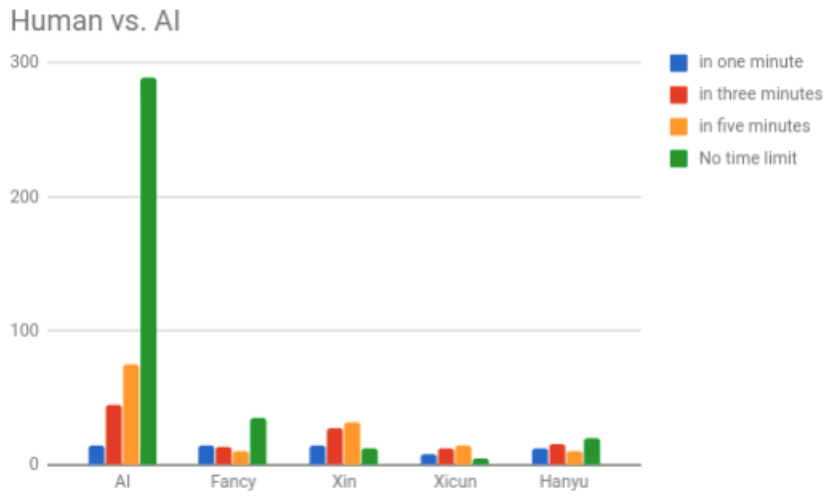


Figure 7: human vs. AI

set of stable system will more reliable. The data that come from stable AI or system also will much more reliable than human's prediction data, since it can be spread out as a formula.

8 Conclusion

Prediction AI has better prediction performance on the ball game than the human's done. prediction AI has more stable and reliable data than human predicted. So, in many industries, people using machine to do some tasks is much better than human. Since machine only need power supply, than it will have more stable performance than human. But in the AI field, now we have different understanding of machine, instead of giving it single unchangeable method, we have machine learning now, instead of do what human told to do, AI should learn from previous experience, which can form another strong relation between AI and it's surroundings.

References

[1] JonD.GiorginiaLance, A.M.BenneraSteven, J.OstroaMichael, C.NolanbMichael and W.Buschc. *Zur Elektrodynamik bewegter Körper.* [*Icarus*]. Elsevier Inc. page:1–19, January 2008.

[2] Stuart Armstrong, Kaj Sotala.
https://intelligence.org/files/PredictingAI.pdf Körper. [*How We're Predict-*

ing AI—or Failing To]. Machine Intelligence research institute. January 2012.

[3] Office of Communications: Chemists harness artificial intelligence to predict the future of chemical reactions, *https://www.princeton.edu/news/2018/02/15/chemists-harness-artificial-intelligence-predict-future-chemical-reactions*

[4] Knuth: Computers and Typesetting, `http://www-cs-faculty.stanford.edu/~uno/abcde.html`

[5] Arun Nair, Praveen Srinivasan, Sam Blackwell, Cagdas Alcicek, Rory Fearon, Alessandro De Maria, Vedavyas Panneershelvam, Mustafa Suleyman, Charles Beattie, Stig Petersen, [*Massively parallel methods for deep reinforcement learning.*]. arXiv preprint arXiv:1507.04296, 2015.

[6] Has¸im Sak, Andrew Senior, Kanishka Rao, Franc¸oise Beaufays, and Johan Schalkwyk. [*Google Voice Search: faster and more accurate,*]. 2015. googleresearch.blogspot.com/2015/09/google-voicesearch-faster-and-more.html.

[7] Anelia Angelova, Alex Krizhevsky, and Vincent Vanhoucke. [*Pedestrian detection with a large-field-of-view deep network. In Robotics and Automation (ICRA),*] [*IEEE International Conference on, pages 704–711*] 2015.

[8] Trishul Chilimbi, Yutaka Suzue, Johnson Apacible, and Karthik Kalyanaraman, Project Adam: Building an efficient and scalable deep learning training system. [*11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14), pages 571–582,*] 2014

[9] P. Moritz, R. Nishihara, I. Stoica, and M. I. Jordan. SparkNet: Training deep networks in Spark. [*In Proceedings of ICLR,*] 2016

[10] C. J. Maddison, A. Huang, I. Sutskever, and D. Silver. Move evaluation in Go using deep convolutional neural networks, 2014