

Deep Convolution Q learning for Thai stock trading

Wisarut Duangmorakot

King mongkut University of Technology Thonburi, FRA500 Term paper

Abstract

Deep Q learning และ Deep Convolution neural network เป็นที่นิยมมากในการนำเอาไปพัฒนา AI ในการเล่นเกมยากๆ อย่างเช่น Atari หรือเกมที่ยากกว่านั้น คือ GO ที่เรียกว่า Alphago เนื่องจากมี performance ที่สูงสามารถใช้แก้ปัญหาต่างๆ ได้ หุ่นก็อาจจะเรียกได้ว่าเป็นเกมที่ยากได้เช่นกัน ใน paper นี้เราจะข้อมูลหุ้นมาสร้างและพล็อตให้อยู่ในรูปแบบรูปภาพที่เป็นกราฟเพื่อสามารถทำ Image Recognition โดยใช้ Convolution Neural network ผสมผสานกับ Deep Q network เพื่อสามารถสร้าง model ที่สามารถเทรดหุ้นได้อย่างมีประสิทธิภาพ ซึ่งเราสร้าง environment ที่ให้ model สามารถเทรดหุ้นได้ โดยใช้ข้อมูลในอดีต ซึ่งเราทดสอบกับหุ้น 3 ตัว ได้แก่ และได้ผลตอบแทนที่ดีคือ ซึ่งมากกว่าการ Buy and hold ที่ทำได้เพียง

Keywords: Stock, DeepQ, ConvNets

1. Introduction

Reinforcement learning เป็นวิธีการเรียนรู้ที่มีความนิยมมากในการแก้ปัญหาที่ยากๆ ซึ่งเราไม่สามารถ Label ข้อมูลชุดนั้นได้โดยตรง ไม่มีการ Label ผลลัพธ์แน่นอน เน้นการ Trial and error เช่นการ Control robot ที่มีข้อต่อเยอะๆ ให้สามารถเดินได้เพื่อให้สามารถเดินได้อย่างถูกต้องและเร็วที่สุด ซึ่งยากที่จะโปรแกรมมันไปตรงๆ Convolution neural network (CNN) ก็เป็นอีกหนึ่ง network ที่ได้รับความนิยมเป็นอย่างสูงในเวลานี้ เพราะสามารถนำไปใช้ในงาน Image recognition ได้เช่น Face recognition (Donghyun Kim & Matthias Hernandez & Jongmoo Choi & Gerard Medioni, 2017) Self-driving car (Brilian Tafjira Nugraha & Shun-Feng Su & Fahmizal, 2017) เนื่องจากเป็น network ที่มีประสิทธิภาพสูง แต่ก็ต้องใช้ข้อมูลจำนวนมาก Reinforcement learning และ Convolution Neural network มักจะถูกนำมาใช้ด้วยกันในทางด้านการทำ AI ในการแก้ปัญหาเรื่องเกม เช่น ให้ AI สามารถ เล่นเกม Atari (Volodymyr Mnih & Koray Kavukcuoglu & David Silver & Alex Graves & Ioannis Antonoglou & Daan Wierstra & Martin Riedmiller, 2013) ได้ อย่างเก่งกาจ การเทรดหุ้นก็เหมือนกับการเล่นเกม ชนิดหนึ่ง ที่มีแพ้ มีชนะ มีรางวัล มีโทษ ซึ่งใน paper นี้จะกล่าวถึงการนำ Reinforcement learning และ Convolution Neural network มาใช้ในการสร้าง strategy สำหรับการเทรดหุ้น เริ่มต้นจากการใช้ข้อมูลหุ้นจากที่เป็นค่าตัวเลขแล้วเปลี่ยนเป็นรูปภาพ การสร้าง environment สำหรับการเทรดหุ้น ซึ่งจะประกอบไปด้วย observation, reward, action หลังจากนี้จะพูดถึงการใช้ Convolution Neural network และ Deep Q Learning network เพื่อสร้าง strategy ในการเทรดหุ้น และทดสอบผลลัพธ์

2. Theory

ในส่วนนี้จะกล่าวถึงการ scale ค่าให้อยู่ในช่วงที่กำหนด เพื่อนำไปใช้กับการสร้างรูปภาพของกราฟหุ่น และการ

2.1 การ scale ค่าให้อยู่ในช่วง min, max

$$x_{std} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

$$x_{scaled} = x_{std} * (max_{scale} - min_{scale}) + min_{scale}$$

สูตรการ scale ค่าให้อยู่ในช่วงที่กำหนด

ซึ่งจะได้ว่า x_{scaled} จะมีค่า $max_{scale} \leq x_{scaled} \leq min_{scale}$

2.2 การ Train ด้วย Reinforcement learning

เป้าหมายในการใช้ Reinforcement learning ก็คือการทำให้โมเดลได้รับ reward ที่มากที่สุด จาก environment ซึ่ง reward จะขึ้นอยู่กับ action ที่เกิดขึ้นของโมเดล และจะมีการกำหนดให้ Agent มี memory สำหรับจำตาเดินต่าง ๆ ไว้ ซึ่งเมื่อจบแต่ละ episode เราจะทำการสุ่มตาเดินเหล่านี้มาทำการปรับ gradient policy จะทำการปรับ model เท่ากับจำนวน batch size โดยใช้ Loss function ดังนี้

$$Loss = [r + \gamma Q(s', a') - Q(s, a)]^2$$

3. Experimental Design

ในส่วนนี้จะกล่าวถึงข้อมูลหุ้น, การสร้างกราฟหุ้นจากข้อมูลหุ้นที่เป็นตัวเลข, การสร้าง Environment เพื่อ เทรน Reinforcement learning และ architecture ของโมเดล

3.1 Data

เราได้ใช้ข้อมูลหุ้น 3 นาที ตั้งแต่ MAY 2017 - MAY 2018 หลังจากนั้นเรานำข้อมูลดังกล่าวมาพล็อตให้อยู่ในรูปแบบของรูปของกราฟหุ้น

สร้างรูปกราฟหุ้นจากข้อมูลหุ้นที่เป็นตัวเลข

นำข้อมูลหุ้นย้อนหลัง 20 วันซึ่งได้แก่ Close หลังจากนั้นนำ SMA (Standard moving average) ของ Close มาพล็อตลงบน Array ซึ่งขั้นตอนการพล็อตเราจะนำ Data ทั้งหมดมา scale ให้อยู่ระหว่าง 0 ถึง 9 โดยใช้สูตรซึ่งกำหนดให้ $\max_{\text{scale}} = 9$ และ $\min_{\text{scale}} = 0$ จะได้ค่าที่อยู่ระหว่าง 0 ถึง 9 หลังจากจึงพล็อตแต่ละ pixel ลงไปบน array รูปภาพ ขนาด 20 x 10 นอกจากนี้เรายังใช้ค่าผลต่างระหว่าง Open กับ Close ที่ผ่านการ scale แล้ว พล็อตลงบน array รูปภาพ 20 x 10 อีกอันหนึ่ง หลังจากนั้นจึงนำเอา array รูปภาพทั้งสองมาต่อกันในแนวนอนจะได้รูปที่มีขนาด 20 x 20 Pixels ซึ่งเราจะได้รูปทั้งหมด 1 รูป แสดงถึงกราฟหุ้นย้อนหลัง 20 period หลังจากนั้นเราทำต่อไปเรื่อยๆจนครบรอบของข้อมูลหุ้น

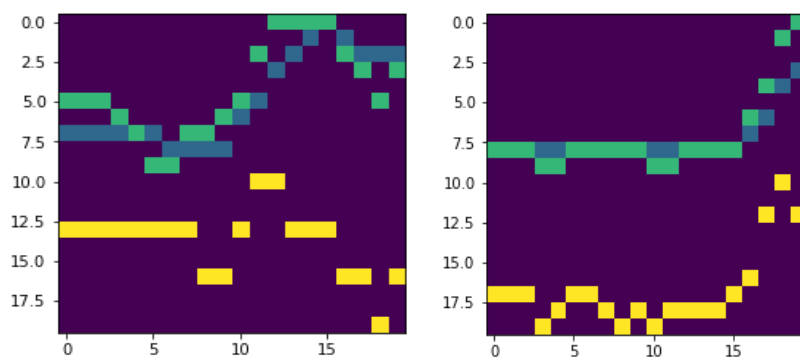


Figure 1: รูปภาพกราฟหุ้นที่ได้จากการ plot ราคาหุ้นที่ผ่านการ scale

3.2 Environment

สร้าง Environment ที่ใช้เชื่อมต่อกับโมเดลโดยมีส่วนประกอบดังนี้

Observation รูปภาพของกราฟหุ้น ที่มีขนาด 20 x 20 Pixels ในแต่ละ time-step

Action มีทั้งสิ้น 2 Action คือ

1. ถือหุ้น เมื่อ Action ที่ t-1 เป็น 0 หมายความว่าให้วันนี้ทำการถือหุ้นที่ราคา Close ของ t ทั้งนี้ แต่ถ้า t-1 เป็น 1 หมายความว่าให้ทำการถือหุ้นที่ราคา t-1 ต่อไป
2. ไม่ถือหุ้น เมื่อ Action ที่ t-1 เป็น 1 หมายความว่าให้วันนี้ทำการขายหุ้นที่ราคา Close ของ t ทั้งนี้ แต่ถ้า t-1 เป็น 0 หมายความว่าไม่ต้องถือหุ้นต่อไป

Reward

Environment จะ return reward กลับเมื่อ

1. เมื่อ $action_t = 0$, $holdPrice \neq 0$ จะได้ $reward = (stockClosePrice_t / holdPrice) - 1.0$
2. เมื่อ $action = 0$, $holdPrice = 0$ จะได้ $reward = -0.2$
3. เมื่อ $action = 1$, $holdPrice \neq 0$ จะได้ $reward = (stockClosePrice_t / holdPrice) - 1.0) / 4$

Done

เป็น true เมื่อวันนี้เป็นวันสุดท้ายในการเล่นหุ้น นอกจากนั้น return ค่ากลับเป็น false

3.3 Architecture ของ Deep Convolution Q learning for Thai stock trading

โมเดลจะทำการรับข้อมูลที่เป็นรูปภาพ Feed เข้าไปใน network โดยมีขนาด 20*20 pixels และจะผ่านการ Image Recognition รูปภาพกราฟและ output ออกมาเป็น action หลังจากเสร็จการทำงานหนึ่งรอบจะมีการ update weight จาก reward ที่ได้รับ

Layer (type)	Output Shape	Parameters
conv2d_360 (Conv2D)	(None, 16, 16, 64)	1664
conv2d_361 (Conv2D)	(None, 14, 14, 32)	18464
conv2d_362 (Conv2D)	(None, 13, 13, 10)	1290
max_pooling2d_144 (MaxPoolin	(None, 6, 6, 10)	0
dropout_120 (Dropout)	(None, 6, 6, 10)	0
flatten_120 (Flatten)	(None, 360)	0
dense_324 (Dense)	(None, 64)	23104
dense_325 (Dense)	(None, 32)	2080
dense_326 (Dense)	(None, 2)	66

Figure 2: Layer แต่ละชั้นของโมเดล, ขนาด และจำนวน parameters

ประกอบไปด้วยทั้งหมด 9 layer

Layer 1-3 Convolution Neural network ทั้งหมด 3 ชั้น ซึ่งประกอบไปขนาดของฟิลเตอร์เป็น 5x5, 3x3, 2x2 ตามลำดับ ใช้สำหรับ Extract Feature เพื่อทำ Image recognition บนรูปภาพของราคาหุ้น

Layer 4 เป็น Max_pooling layer เพื่อลดขนาดของ output สำหรับส่งไป layer ชั้นถัดไป

Layer 5 Dropout ใช้เพื่อลดการ Overfit ของ model

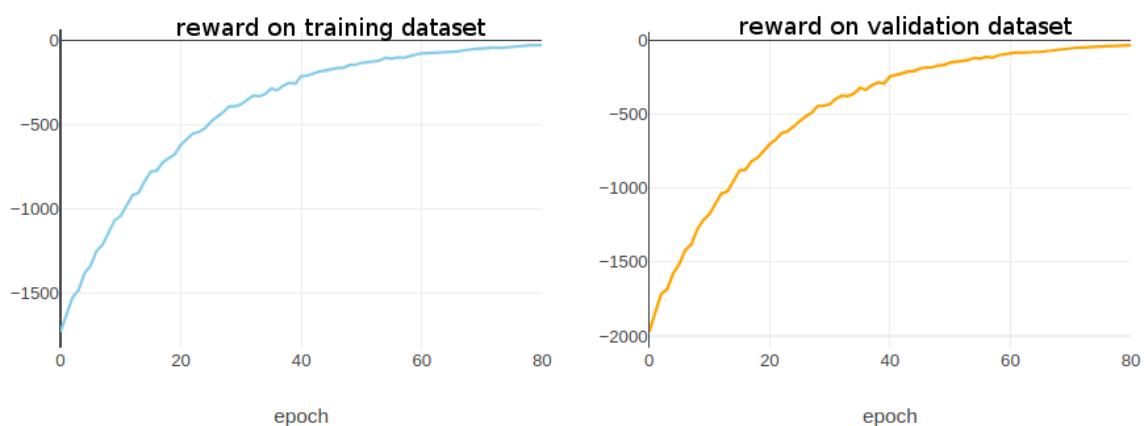
Layer 6 Flatten ใช้เพื่อ ทำ Dimensional reduction ให้สามารถส่ง output ไปยังชั้น Feed forward network

Layer 7-8 Feed forward network เป็น layer ที่สำหรับใช้เรียนรู้เรื่องและเก็บ Policy ทั้งหมดไว้

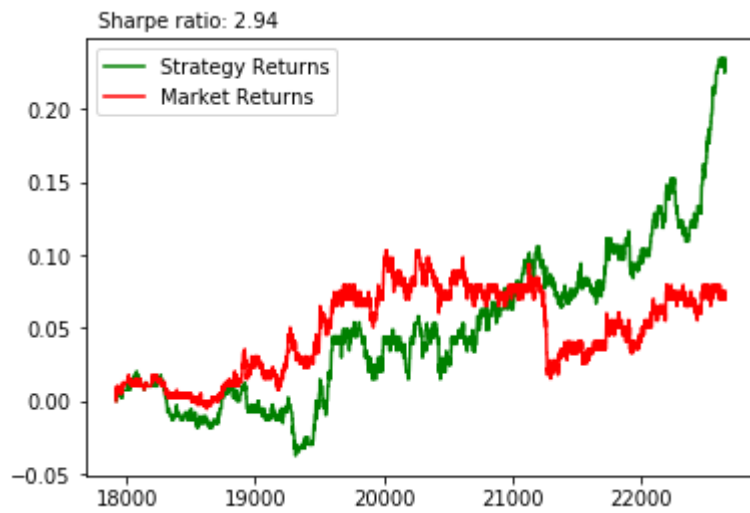
Layer 9 เป็น Output Node ขนาด 1 Node เพื่อส่งออก output ที่เป็น action ซึ่งมีค่าที่เป็นไปได้คือ 0 กับ 1

4. Experimental Results and Evaluation

เราได้ทำการทดสอบกับหุ้นทั้งหมด 3 ตัว ซึ่งได้แก่ เหตุผลที่ทดสอบกับหุ้นทั้งหมดสามตัวหุ้นไทยบน SET โดยใช้ ข้อมูลหุ้นที่มี interval ราย 3 นาที ซึ่งมี period ระหว่างปี MAY 2017 - MAY 2018 ซึ่งวิธีการ Evaluation เรา จะใช้ accumulative return ที่ได้จาก Strategy ที่ได้จาก Model ทำการวัดเพื่อเทียบกับการถือไว้เฉยๆแล้วขาย (Buy and Hold) ของหุ้นทั้ง 3 ตัว โดยเราได้แบ่งข้อมูลตั้งแต่ MAY 2017 - MARCH 2018 เป็น Train dataset และ MARCH 2018 - MAY 2018 เป็น Validation dataset หลังจากนั้นได้ทำการคัดเลือกโมเดลที่สามารถทำผลตอบแทนได้ดีที่สุดกับ train dataset เพื่อนำมาโมเดลมาทดสอบกับ validation dataset

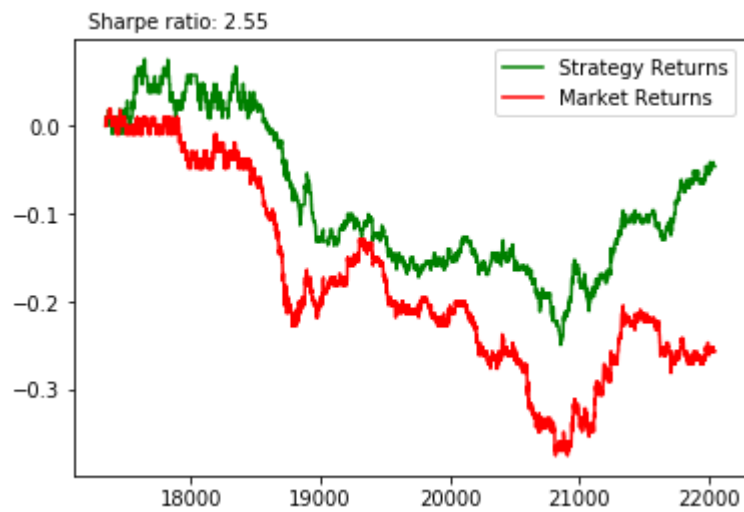


Reward ของ Train dataset และ Validation dataset เมื่อเทรนโมเดลกับหุ้น ADVANC.BK

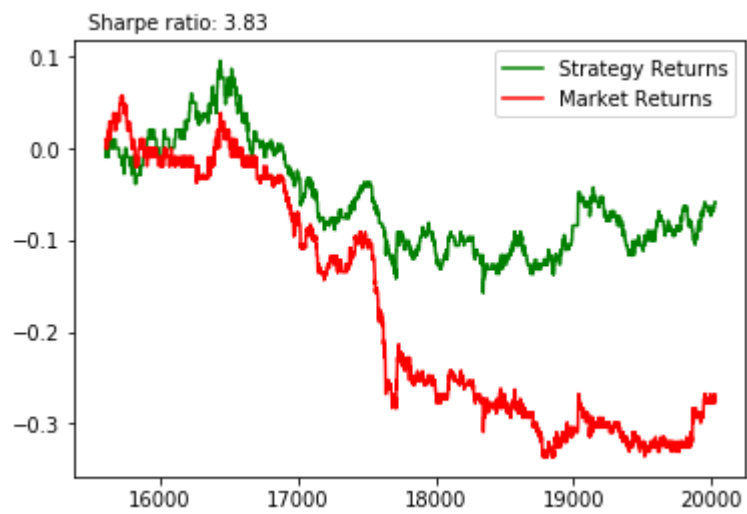


กราฟแสดง

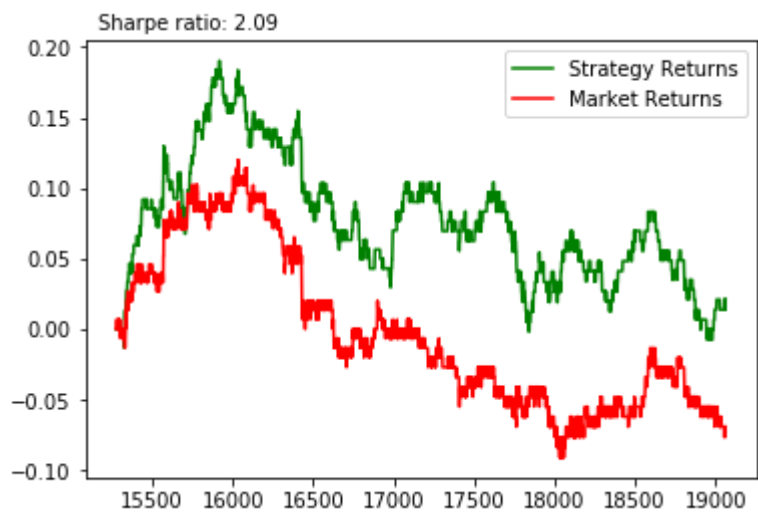
acumulative dialy return ของ model strategy กับ buy and hold บม ADVANC.BK



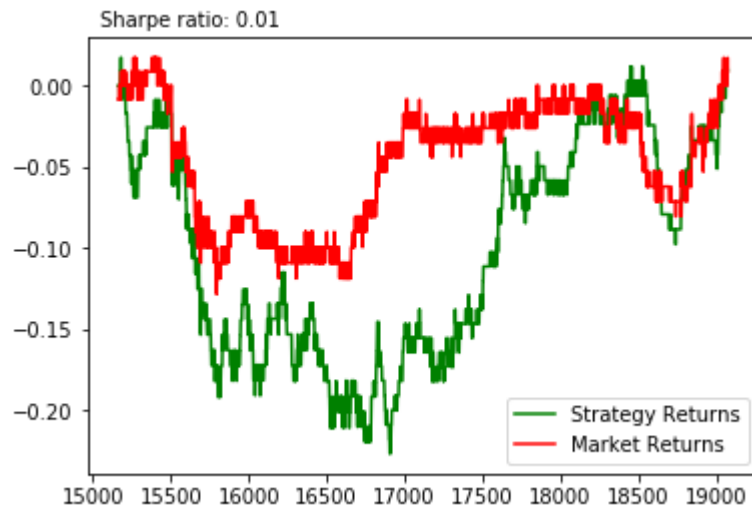
กราฟแสดง acumulative dialy return ของ model strategy กับ buy and hold บม AMATA.BK



กราฟแสดง acumulative dialy return ของ model strategy กับ buy and hold บม ANAN.BK



กราฟแสดง acumulative dialy return ของ model strategy กับ buy and hold บม GGC.BK



กราฟแสดง acumulative daly return ของ model strategy กับ buy and hold บน MAJOR.BK

Stock	Strategy Return	Buy and hold return
ADVANC.BK	21.47%	7.64 %
AMATA.BK	-4.60 %	-25.67 %
ANAN.BK	-5.86 %	-27.29 %
GGC.BK	2.08 %	-6.95 %
MAJOR.BK	0.94 %	0.85%

ผลตอบแทนที่โมเดลทำได้ในการทดสอบใน Validation dataset กับหุ้นทั้ง 3 ตัวในปี 2016-2018

จากผลการทดสอบสามารถเห็นได้ว่าโมเดลที่เราสร้างขึ้นสามารถทำผลตอบแทนได้ค่อนข้างดี และดีกว่าการ Buy and Hold ซึ่งอาจจะไม่สามารถบอกได้ว่าโมเดลนี้ได้ผลที่ดีเพราะว่าทดสอบด้วยหุ้นจำนวน 3 ตัว เนื่องด้วยมีข้อจำกัดทางด้านทรัพยากรการคำนวณ

5. Conclusion

เราสามารถนำ Convolution Neural network ร่วมกับการใช้ Deep reinforcement learning network ได้ เช่นเดียวกับการนำเอาไปใช้บนเกม เช่น AI เล่นเกม Atari, Pong โดยการนำเอาราคาหุ้นไปพล็อตเป็นกราฟหุ้นก่อน หลังจากนั้นจึงใช้ CNN และ DQN ในการเรียนรู้ และจากผลการทดสอบสามารถเห็นได้ว่าโมเดลที่เราสร้างขึ้นสามารถทำผลตอบแทนได้ดีกว่าการ Buy and Hold ซึ่งอาจจะไม่สามารถบอกได้ว่าโมเดลนี้ได้ผลที่ดีมากเพราะว่าทดสอบด้วยหุ้นจำนวน 5 ตัว รวมไปถึงการใช้จำนวนรอบในการเทรนน้อย ใช้เพียง 100 epoch เท่านั้น เนื่องด้วยมีข้อจำกัดทางด้านทรัพยากรในการคำนวณ

6. References

Brilian Tafjira Nugraha, Shun-Feng Su, Fahmizal, Towards self-driving car using convolutional neural network and road lane detector, DOI: 10.1109/ICACOMIT.2017.8253388 ,2017

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller, Playing Atari with Deep Reinforcement Learning, arXiv:1312.5602, 2013

Donghyun Kim, Matthias Hernandez, Jongmoo Choi, Gerard Medioni, Deep 3D Face Identification, arXiv:1703.10714, 2017