

ระบบสนับสนุนการตัดสินใจในการซื้อขายหุ้นด้วยปัญญาประดิษฐ์
STOCK TRADING DECISION SUPPORT WITH AI



โดย
นางสาว วิลาสินี รวมเกียรติขจร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

ระบบสนับสนุนการตัดสินใจในการซื้อขายหุ้นด้วยปัญญาประดิษฐ์
STOCK TRADING DECISION SUPPORT WITH AI

โดย

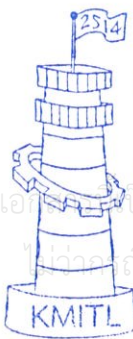
นางสาว วิลาสิณี รวมเกียรติขจร

58011164

อาจารย์ที่ปรึกษา

ดร. สมปอง วิเศษพานิชกิจ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561



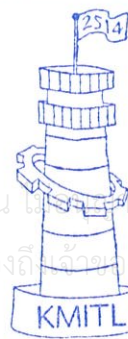
ผ่านการตรวจรูปเล่มแล้ว

(*[Signature]*)

อาจารย์ที่ปรึกษา

23/5/2562

วิศวกรรมโทรคมนาคม
Telecommunications Engineering



ผ่านการตรวจชิ้นงานแล้ว

(*[Signature]*)

กรรมการผู้ตรวจชิ้นงาน

23/5/62

วิศวกรรมโทรคมนาคม
Telecommunications Engineering

เอกสารนี้เป็นเอกสารลิขสิทธิ์ส่วนตัว สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่สามารถเผยแพร่หรือใช้ซ้ำโดยไม่ได้รับอนุญาต
ถ้าคุณมีข้อสงสัยใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2561

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง


เรื่อง ระบบสนับสนุนการตัดสินใจในการซื้อขายหุ้นด้วยปัญญาประดิษฐ์

STOCK TRADING DECISION SUPPORT WITH AI

ผู้จัดทำ

1. นางสาว วิลาสิณี รวมเกียรติขจร

58011164


..... อาจารย์ที่ปรึกษา
(ดร. สมปอง วิเศษพานิชกิจ)



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการดำเนินการจัดทำปริญญาานิพนธ์ “ระบบสนับสนุนการตัดสินใจในการซื้อขายหุ้นด้วยปัญญาประดิษฐ์” จะไม่สามารถสำเร็จลุล่วงไปได้ด้วยดี หากไม่ได้รับความกรุณาอย่างสูงจาก ดร. สมปอง วิเศษพานิชกิจ อาจารย์ที่ปรึกษาปริญญาานิพนธ์, ผศ.ดร. นภัทร สระเยี่ยม และ ผศ.ดร. ธเนศ พัฒนธาดาพงษ์ ที่กรุณาให้คำแนะนำ ปรึกษาตลอดจนปรับปรุงแก้ไขข้อบกพร่องต่างๆ ที่เป็นประโยชน์ต่อการศึกษา คำนึงว่าให้ปริญญาานิพนธ์นี้สำเร็จและสมบูรณ์ยิ่งขึ้นด้วยความเอาใจใส่อย่างดียิ่ง ผู้จัดทำได้ตระหนักถึงความตั้งใจจริงและความทุ่มเทของอาจารย์ และขอขอบพระคุณอาจารย์ทุกท่านเป็นอย่างสูง

ขอขอบคุณคณาจารย์และเจ้าหน้าที่ประจำภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ได้อบรมสั่งสอนให้ความรู้ และประสบการณ์แก่ผู้จัดทำ

ขอบคุณเพื่อนทุกคนที่ให้ความช่วยเหลือ และคำแนะนำที่เป็นประโยชน์ในการจัดทำปริญญาานิพนธ์ ตลอดจนขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของผู้จัดทำที่เกี่ยวข้องที่คอยให้คำปรึกษา และให้ความสนับสนุนด้านการเรียน อีกทั้งยังเป็นกำลังใจสำคัญในการจัดทำปริญญาานิพนธ์ฉบับนี้

ขอขอบคุณผู้เขียนบทความ รวมถึงหนังสือต่างๆ ที่ให้ความรู้อันเป็นประโยชน์แก่ผู้จัดทำ ตลอดจนผู้มีส่วนเกี่ยวข้องที่ไม่ได้กล่าวถึงทุกท่านที่คอยให้การสนับสนุน ช่วยเหลือ ให้ความรู้เพิ่มเติมจนสามารถทำให้ปริญญาานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี

นางสาว วิลาสินี รวมเกียรติขจร
ผู้จัดทำ

ระบบสนับสนุนการตัดสินใจในการซื้อขายหุ้นด้วย
ปัญญาประดิษฐ์

STOCK TRADING DECISION SUPPORT WITH AI

โดย นางสาววิลาสิณี รวมเกียรติขจร 58011164

อาจารย์ที่ปรึกษา ดร. สมปอง วิเศษพานิชกิจ

บทคัดย่อ

ปัญญานิพนธ์ฉบับนี้นำเสนอระบบสนับสนุนการตัดสินใจในการซื้อขายหุ้น โดยใช้ความสามารถของปัญญาประดิษฐ์มาประยุกต์ใช้ ซึ่งจะทำการวิเคราะห์ข้อมูลราคาหุ้นที่มีอยู่เดิมด้วยวิธีการของ Machine Learning หรือ Deep Learning แบบ LSTM (Long Short Term Memory Network) เพื่อให้ได้รูปแบบการคาดการณ์ผลราคาหุ้นในอนาคต และใช้ภาษา Python ในการเขียนโปรแกรมเพื่อสร้างระบบที่ใช้แสดงผลสำหรับผู้ใช้ โดยระบบนี้สามารถใช้เป็นตัวช่วยในการตัดสินใจในการลงทุนซื้อขายหุ้น เพื่อลดความเสี่ยงที่อาจเกิดขึ้นได้

ABSTRACT

This thesis presents a decision support system for stock trading. By using the ability of artificial intelligence that will analyze the existing stock price data with the method of Machine Learning or Deep Learning, LSTM (Long Short-Term Memory Network) to obtain a forecast model for future stock price results. And use Python programming language to create a system for users that can be used as a tool for making investment decisions to reduce potential risks.

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	III
สารบัญรูป	VI
สารบัญตาราง	VIII
บทที่ 1	
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของปริญญานิพนธ์	1
บทที่ 2	
ทฤษฎีและหลักการที่เกี่ยวข้อง	2
2.1 STOCK	2
2.1.1 ประเภทของหุ้น	2
2.1.2 ตลาดหลักทรัพย์ หรือตลาดหุ้น	3
2.1.3 ตลาดหลักทรัพย์แห่งประเทศไทย (STOCK EXCHANGE OF THAILAND: SET)	3
2.1.4 หลักการวิเคราะห์หลักทรัพย์	3
2.2 ARTIFICIAL INTELLIGENCE	6
2.2.1 การกระทำคล้ายมนุษย์ (ACTING HUMANLY)	7
2.2.2 การคิดคล้ายมนุษย์ (THINKING HUMANLY)	7
2.2.3 คิดอย่างมีเหตุผล (THINKING RATIONALLY)	7
2.2.4 กระทำอย่างมีเหตุผล (ACTING RATIONALLY)	7
2.3 MACHINE LEARNING	8
2.4 DEEP LEARNING	8
2.5 NEURAL NETWORK	9
2.5.1 รูปแบบการเรียนรู้ของโครงข่ายประสาทเทียม	10

สารบัญ (ต่อ)

	หน้า
2.5.2 RECURRENT NEURAL NETWORK	11
2.5.3 LONG SHORT-TERM MEMORY NETWORK	13
2.5.4 ACTIVATE FUNCTION	14
2.5.5 OPTIMIZER	18
2.6 การวัดความแม่นยำของการคาดการณ์ผล	19
2.7 TENSORFLOW	19
2.8 KERAS	20
2.9 ANACONDA	21
2.9.1 JUPYTER NOTEBOOK	21
2.9.2 VISUAL STUDIO CODE	22
2.10 PYTHON	23
2.10.1 PACKAGE PYTHON ที่ใช้งาน	23
บทที่ 3 การออกแบบและการจัดทำปริญญานิพนธ์	24
3.1 การออกแบบ	24
3.1.1 การออกแบบการทำงานของระบบ	24
3.1.2 การดำเนินงานในส่วนของการเตรียมข้อมูล	24
3.1.3 การเขียนโปรแกรมในการสร้าง MODEL	28
3.1.4 การออกแบบ GUI ในส่วนของผู้ใช้งาน	29
3.2 เครื่องมือที่ใช้ในการทดลอง	30
3.2.1 คอมพิวเตอร์	30
3.2.2 โปรแกรม ANACONDA	30
3.2.3 โปรแกรม JUPYTER NOTEBOOK	31
3.2.4 โปรแกรม VISUAL STUDIO CODE	31
3.3 การจัดเก็บผลการทดลอง	31
3.3.1 การดาวน์โหลดข้อมูลด้วยตนเอง	31
3.3.2 การใช้ภาษา PYTHON เพื่อจัดเรียงข้อมูล	31

สารบัญ (ต่อ)

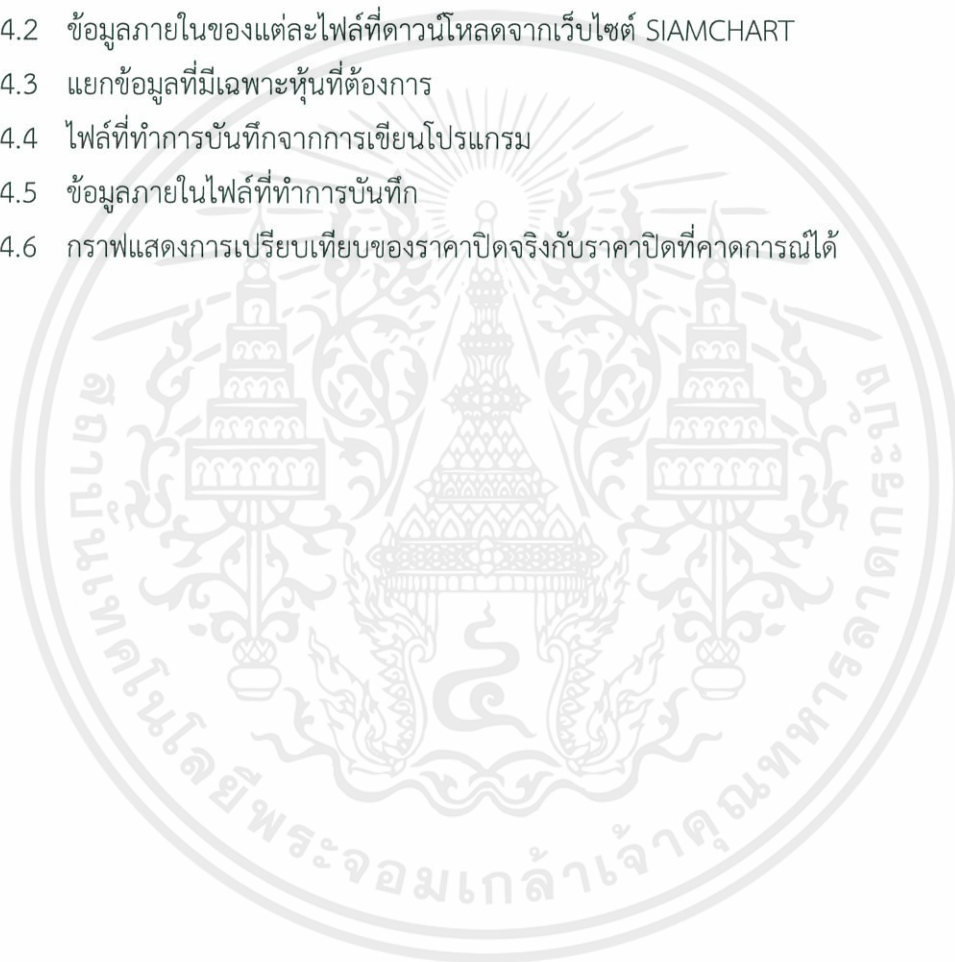
	หน้า
3.3.3 การปรับเปลี่ยนพารามิเตอร์เพื่อสร้าง MODEL	31
3.3.4 การทดลองปรับเปลี่ยนอัตราส่วนการแบ่งชุดข้อมูล	31
3.3.5 การทดลองการคาดการณ์ผลราคาหุ้นหลายวัน	31
บทที่ 4 ผลการทดลอง	32
4.1 ผลการดาวน์โหลดไฟล์ข้อมูลราคาหุ้นจากเว็บไซต์ SIAMCHART	32
4.2 ผลการใช้ภาษา PYTHON ในการจัดเตรียมข้อมูล	33
4.3 ผลการปรับเปลี่ยนพารามิเตอร์เพื่อสร้าง MODEL	35
4.4 ผลการทดลองปรับเปลี่ยนอัตราส่วนการแบ่งชุดข้อมูล	38
4.5 ผลการทดลองการคาดการณ์ผลราคาหุ้นหลายวัน	39
บทที่ 5 สรุปผลและข้อเสนอแนะ	41
5.1 สรุปผล	41
5.2 ข้อเสนอแนะ	42
5.3 ปัญหาและอุปสรรค	42
บรรณานุกรม	43

สารบัญรูป

รูปที่	หน้า
2.1	4
2.2	8
2.3	8
2.4	9
2.5	11
2.6	13
2.7	15
2.8	15
2.9	16
2.10	17
2.11	17
2.12	18
2.13	19
2.14	20
2.15	21
2.16	21
2.17	22
3.1	24
3.2	24
3.3	25
3.4	25
3.5	26
3.6	27
3.7	28
3.8	29
3.9	30

สารบัญรูป (ต่อ)

รูปที่		หน้า
3.10	หน้าต่าง GUI สำหรับผู้ใช้งานในส่วนของการใช้ MODEL คาดการณ์ผล ราคาหุ้น	30
4.1	ไฟล์ข้อมูลที่ทำกรดาวน์โหลดด้วยตนเองจากเว็บไซต์ SIAMCHART	32
4.2	ข้อมูลภายในของแต่ละไฟล์ที่ดาวน์โหลดจากเว็บไซต์ SIAMCHART	33
4.3	แยกข้อมูลที่มีเฉพาะหุ้นที่ต้องการ	34
4.4	ไฟล์ที่ทำกรบันทึกจากการเขียนโปรแกรม	34
4.5	ข้อมูลภายในไฟล์ที่ทำกรบันทึก	34
4.6	กราฟแสดงการเปรียบเทียบของราคาปิดจริงกับราคาปิดที่คาดการณ์ได้	40



สารบัญตาราง

ตารางที่	หน้า
4.1 ความสัมพันธ์ของค่า RMSE กับการปรับเปลี่ยนค่าพารามิเตอร์ต่างๆ ใน การสร้าง MODEL การคาดการณ์ผล เมื่อกำหนดให้รอบของการ ประมวลผลมีค่าเท่ากับ 50	35
4.2 ความสัมพันธ์ของค่า RMSE กับการปรับเปลี่ยนค่าพารามิเตอร์ต่างๆ ใน การสร้าง MODEL การคาดการณ์ผล เมื่อกำหนดให้รอบของการ ประมวลผลมีค่าเท่ากับ 100	36
4.3 ความสัมพันธ์ของค่า RMSE กับการปรับเปลี่ยนค่าพารามิเตอร์ต่างๆ ใน การสร้าง MODEL การคาดการณ์ผล เมื่อกำหนดให้รอบของการ ประมวลผลมีค่าเท่ากับ 200	37
4.4 ความสัมพันธ์ของค่า RMSE กับการปรับเปลี่ยนอัตราการแข่งขันข้อมูล เมื่อกำหนดให้รอบของการประมวลผลมีค่าเท่ากับ 50	38
4.5 ความสัมพันธ์ของค่า RMSE กับการปรับเปลี่ยนอัตราการแข่งขันข้อมูล เมื่อกำหนดให้รอบของการประมวลผลมีค่าเท่ากับ 100	38
4.6 ความสัมพันธ์ของค่า RMSE กับการปรับเปลี่ยนอัตราการแข่งขันข้อมูล เมื่อกำหนดให้รอบของการประมวลผลมีค่าเท่ากับ 200	39
4.7 ตารางผลการทดลองการคาดการณ์ผลราคาหุ้นหลายวัน	39

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ระบบสนับสนุนการตัดสินใจในการซื้อขายหุ้นด้วยปัญญาประดิษฐ์ มีวัตถุประสงค์เพื่อศึกษาการประมวลผลแบบ Machine Learning และ Deep Learning โดยใช้ TensorFlow ทำงานร่วมกับ Keras และจัดทำขึ้นเพื่อสร้างระบบที่สามารถใช้เป็นเครื่องมือที่ช่วยในการตัดสินใจเกี่ยวกับการลงทุน ซึ่งนำเทคนิคการประมวลผลดังกล่าวมาประยุกต์ใช้กับข้อมูลราคาหุ้นที่มีอยู่เพื่อคาดการณ์ผลราคาหุ้นในอนาคต

1.2 วัตถุประสงค์

- 1) มีความรู้ความเข้าใจในการประมวลผลประเภท Machine Learning และ Deep Learning
- 2) เพื่อประยุกต์ใช้ความรู้ทางด้านวิศวกรรมกับการลงทุนหุ้น
- 3) เพื่อศึกษากระบวนการใช้ TensorFlow และ Keras
- 4) เพื่อศึกษาการเขียนโปรแกรมในการคำนวณค่าความเสี่ยงในการลงทุนหุ้น

1.3 ขอบเขตของปริญยานิพนธ์

- 1) สามารถเขียนโปรแกรมเพื่อใช้เป็นเครื่องมือเพื่อช่วยในการตัดสินใจในการลงทุนซื้อขายหุ้น
- 2) สามารถนำปัญญาประดิษฐ์ (AI) มาประยุกต์ใช้ในการลงทุนซื้อขายหุ้น
- 3) สามารถใช้ภาษา Python ร่วมกับ TensorFlow และ Keras ในการเขียนโปรแกรมเพื่อใช้เป็นเครื่องมือในการตัดสินใจในการซื้อขายหุ้น
- 4) สามารถเขียนโปรแกรมเพื่อคาดการณ์ราคาหุ้นในอนาคตได้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

ปริญญานิพนธ์เรื่อง “ระบบสนับสนุนการตัดสินใจในการซื้อขายหุ้นด้วยปัญญาประดิษฐ์” สามารถแบ่งการทำงานหลักๆ ได้เป็น 2 ส่วน คือ ส่วนของการเตรียมข้อมูล และ ส่วนของการฝึกอบรมข้อมูลเพื่อสร้างโมเดลหรือรูปแบบการคาดการณ์ ดังนั้นปริญญานิพนธ์นี้จึงมีหลักการที่เกี่ยวข้องดังต่อไปนี้

2.1 Stock

หุ้น (Stock) เป็นรูปแบบหนึ่งของการลงทุน ซึ่งเป็นตราสารที่กิจการออกให้แก่ผู้ถือ (Holder) เพื่อระดมเงินทุนไปใช้ในกิจการ โดยผู้ถือตราสารทุนจะมีฐานะเป็น “เจ้าของกิจการ” ซึ่งจะมีส่วนได้เสีย หรือมีสิทธิในทรัพย์สิน และรายได้ของกิจการ รวมทั้งมีโอกาสได้รับผลตอบแทนเป็นเงินปันผล (Dividend) โดยจะขึ้นอยู่กับผลกำไร และข้อตกลงของกิจการนั้นๆ

2.1.1 ประเภทของหุ้น

1) หุ้นสามัญ (Common Stock)

เป็นตราสารที่ออกโดยบริษัทมหาชนจำกัด (บมจ.) ที่ต้องการระดมเงินทุนจากประชาชน โดยผู้ถือหุ้นสามัญจะมีสิทธิร่วมเป็นเจ้าของบริษัท มีสิทธิในการออกเสียงลงมติในที่ประชุมผู้ถือหุ้นตามสัดส่วนของหุ้นที่ถือครองอยู่

กล่าวคือผู้ถือหุ้นสามารถร่วมเป็นผู้ตัดสินใจในปัญหาสำคัญในที่ประชุมผู้ถือหุ้น เช่น การเพิ่มทุน การจ่ายเงินปันผล การควบรวมกิจการ ฯลฯ นอกจากนี้ ผู้ถือหุ้นสามัญยังมีสิทธิที่จะได้รับเงินปันผลเมื่อบริษัทมีผลกำไร และมีโอกาสได้รับกำไรจากส่วนต่างของราคาเมื่อราคาหลักทรัพย์ปรับตัวสูงขึ้นตามศักยภาพของบริษัท รวมถึงมีโอกาสได้รับสิทธิในการจองซื้อหุ้นออกใหม่เมื่อบริษัทเพิ่มทุนหรือจัดสรรใบสำคัญแสดงสิทธิต่างๆ ให้แก่ผู้ถือหุ้น

2) หุ้นบุริมสิทธิ (Preferred Stock)

เป็นตราสารที่ผู้ถือมีส่วนร่วมเป็นเจ้าของกิจการเช่นเดียวกับหุ้นสามัญ แต่จะไม่มีสิทธิในการออกเสียงลงมติในที่ประชุมผู้ถือหุ้น และเมื่อกิจการมีกำไรที่เกิดจากการดำเนินงาน ผู้ถือหุ้นบุริมสิทธิจะได้รับเงินปันผลในอัตราคงที่ ซึ่งอาจจะมากหรือน้อยกว่าผู้ถือหุ้นสามัญก็ได้ ขณะเดียวกัน

หากกิจการนั้นต้องเลิกกิจการและมีการชำระบัญชีโดยการขายทรัพย์สิน ผู้ถือหุ้นบุริมสิทธิก็จะได้รับเงินคืนทุนก่อนผู้ถือหุ้นสามัญ

2.1.2 ตลาดหลักทรัพย์ หรือตลาดหุ้น

เป็นสถานที่สำหรับซื้อขายแลกเปลี่ยนหลักทรัพย์ระยะยาวของบริษัทมหาชนจำกัด (บมจ.) ซึ่งถือว่าเป็นตลาดรอง (Secondary Market) ทั้งนี้เนื่องจากในตลาดหุ้นจะทำการซื้อขายเฉพาะหลักทรัพย์ที่ได้ออกจำหน่ายให้แก่ประชาชนทั่วไป โดยหลักทรัพย์ระยะยาว จะประกอบไปด้วยตราสารหนี้ และตราสารทุน ได้แก่ หุ้นสามัญ หุ้นบุริมสิทธิ ใบสำคัญแสดงสิทธิแบบต่างๆ ใบสำคัญแสดงสิทธิอนุพันธ์ หุ้นกู้ และหน่วยลงทุน เป็นต้น โดยเรียกว่าเป็นประเภทของตราสารเพื่อการลงทุนในตลาดหลักทรัพย์มีอยู่แทบทุกประเทศทั่วโลก

2.1.3 ตลาดหลักทรัพย์แห่งประเทศไทย (Stock Exchange of Thailand: SET)

เป็นตลาดหลักทรัพย์แห่งประเทศไทย จัดตั้งขึ้นโดยพระราชบัญญัติตลาดหลักทรัพย์แห่งประเทศไทย พ.ศ. 2517 อยู่ภายใต้การกำกับดูแลของสำนักงานคณะกรรมการกำกับหลักทรัพย์และตลาดหลักทรัพย์ (ก.ล.ต.) เปิดทำการซื้อขายขึ้นอย่างเป็นทางการครั้งแรกในวันที่ 30 เมษายน พ.ศ. 2518 ทำหน้าที่เป็นตลาดรองเพื่อแลกเปลี่ยนซื้อขายตราสารทุนของบริษัทต่างๆ ที่ขึ้นทะเบียนไว้ และเพื่อให้สามารถระดมเงินทุนเพิ่มเติมจากสาธารณะได้โดยสะดวก ปัจจุบันการดำเนินงานของตลาดหลักทรัพย์แห่งประเทศไทย อยู่ภายใต้พระราชบัญญัติหลักทรัพย์และตลาดหลักทรัพย์ พ.ศ. 2535

2.1.4 หลักการวิเคราะห์หลักทรัพย์

หลักการวิเคราะห์หุ้นอย่างมีประสิทธิภาพแบ่งออกเป็นศาสตร์ใหญ่ๆ อยู่ 2 ประเภท คือ Fundamental Analysis เป็นการวิเคราะห์หลักทรัพย์โดยใช้ปัจจัยพื้นฐาน และ Technical Analysis เป็นการวิเคราะห์หลักทรัพย์โดยใช้ปัจจัยทางเทคนิค

1) การวิเคราะห์หลักทรัพย์โดยใช้ปัจจัยพื้นฐาน

เป็นการวิเคราะห์หลักทรัพย์แบบหนึ่งซึ่งจะมุ่งประเมินมูลค่าของหลักทรัพย์ในปัจจุบัน โดยพิจารณาปัจจัยที่มีผลต่อมูลค่าที่แท้จริง (Intrinsic Value) ของหลักทรัพย์ และทำการประเมินมูลค่าที่แท้จริงของหลักทรัพย์ออกมา ผลหรือข้อมูลจากการวิเคราะห์ด้วยวิธีการดังกล่าวนี้ สามารถใช้เป็นเกณฑ์ประกอบการพิจารณาตัดสินใจการลงทุนได้ อาทิเช่น จะซื้อหลักทรัพย์ต่อเมื่อราคาตลาดของหลักทรัพย์นั้นต่ำกว่ามูลค่าตามปัจจัยพื้นฐานที่คำนวณได้ และจะขายหลักทรัพย์เมื่อพบว่าราคาตลาดของหลักทรัพย์นั้นสูงกว่ามูลค่าตามราคาพื้นฐาน มีขั้นตอนการวิเคราะห์ 3 ขั้นตอน คือ

การวิเคราะห์เศรษฐกิจโดยทั่วไป (Economic Analysis) การวิเคราะห์อุตสาหกรรม (Industry Analysis) และการวิเคราะห์บริษัท (Company Analysis)

2) การวิเคราะห์หลักทรัพย์โดยใช้ปัจจัยทางเทคนิค

เป็นการวิเคราะห์การเคลื่อนไหวของราคาหลักทรัพย์ในอดีต ที่ใช้หลักสถิติ เพื่อคาดการณ์ราคาที่น่าจะเป็นในอนาคต โดยเชื่อราคาปัจจุบันที่สะท้อนข่าวสารในด้านต่างๆ ซึ่งจะมีลักษณะเป็นแนวโน้ม และมีการเคลื่อนไหวของราคาที่เกี่ยวข้องกับในอดีต ซึ่งการวิเคราะห์ปัจจัยทางเทคนิคนั้นจะมีเทคนิคต่าง ๆ ได้แก่ การดูลักษณะของเส้นแนวโน้ม (Trend Line) และการใช้ดัชนีบ่งชี้ (Indicators)

2.1) เส้นแนวโน้ม (Trend Line)



รูปที่ 2.1 เส้นแนวโน้มลักษณะต่าง ๆ [1]

เส้นแนวโน้มเป็นเส้นที่แสดงถึงทิศทางหุ้นที่เคลื่อนไหวไปในแนวทางหนึ่งตามแนวโน้มของราคาหุ้นที่จะเกิดขึ้นในอนาคต ซึ่งเป็นเส้นที่ลากผ่านระหว่างจุดที่มีนัยสำคัญตั้งแต่ 2 จุดขึ้นไป มีลักษณะดังนี้

- แนวโน้มขาขึ้น (Uptrend) จะเป็นเส้นตรงที่ลากผ่านจุดต่ำอย่างน้อย 2 จุดในแนวขึ้น มีรูปแบบ คือ จุดยอดของราคาที่ขึ้นไปในแต่ละครั้งอาจจะสูงกว่าจุดสูงสุดเก่า และราคาต่ำสุดในครั้งใหม่จะมีค่าสูงกว่าครั้งก่อน หากว่าราคาหุ้นตกลงทะลุผ่านเส้นแนวโน้มนี้ เป็นการบอกถึงแนวโน้มหุ้นจะเปลี่ยนแปลง

- แนวโน้มขาลง (Downtrend) จะเป็นเส้นตรงที่ลากผ่านจุดสูงสุดอย่างน้อย 2 จุดในแนวลง มีรูปแบบ คือ จุดยอดของราคาจะต่ำกว่าจุดสูงสุดเก่า ถ้าหากว่าราคาหุ้นทะลุผ่านเส้นแนวโน้มขึ้นไป จะเป็นการบอกถึงแนวโน้มหุ้นจะเปลี่ยนแปลง

- แนวโน้มไม่เปลี่ยนแปลง หรือแนวโน้มราบ (Sideway) จะเป็นเส้นที่มีลักษณะการปรับตัวของราคาหุ้นในช่วงแคบๆ มีรูปแบบ คือ ราคาหุ้นที่สูงขึ้นครั้งใหม่จะเท่ากับราคาที่เคยสูงขึ้นไปครั้งก่อน หรือราคาหุ้นต่ำลงครั้งใหม่จะเท่ากับหรือใกล้เคียงกับราคาต่ำลงในครั้งก่อน

2.2) การใช้ดัชนีบ่งชี้ (Indicators)

ดัชนีบ่งชี้ (Indicators) เกิดจากการนำค่าของราคาและปริมาณหลักทรัพย์ที่ซื้อขายในอดีตมาคำนวณ โดยใช้หลักทางคณิตศาสตร์และสถิติ แบ่งได้หลายแบบ ดังนี้

- Trend Following เช่น ค่าเฉลี่ยเคลื่อนที่ (Moving Average) ดัชนีบ่งชี้นี้เป็นตัวบอกถึงการเปลี่ยนแปลงแนวโน้มตัวหนึ่ง ซึ่งจะบอกถึงการเปลี่ยนแปลงอย่างมีนัยสำคัญ เช่น Exponential Moving Average (EMA) ดัชนีบ่งชี้เป็นตัวชี้ถึงการเปลี่ยนแปลงแนวโน้มตัวหนึ่ง ซึ่งจะบอกถึงการเปลี่ยนแปลงอย่างมีนัยสำคัญ ซึ่งสามารถใช้หาแนวโน้มของราคาหุ้นได้ และการตัดกันของเส้นค่าเฉลี่ยเคลื่อนที่ที่สามารถเป็นตัวบอกถึงสัญญาณการเปลี่ยนแปลงแนวโน้มต่างๆ ได้ ซึ่งมีความหมาย คือ

$$EMA_t = EMA_{t-1} + a(P_t - EMA_{t-1}) \quad (2.1)$$

โดยที่

EMA	คือ	ค่าของ Exponential Moving Average ณ เวลาปัจจุบัน
EMA_{t-1}	คือ	ค่าของ Exponential Moving Average ณ คาบเวลาก่อนหน้า
P_t	คือ	ราคาปัจจุบัน
a	คือ	$SmoothingFactor = \frac{2}{n+1}$
n	คือ	จำนวนวัน

ในการกำหนดเส้นค่าเฉลี่ย จะมีการกำหนดจำนวนวันที่แตกต่างกันขึ้นอยู่กับระยะเวลาในการลงทุน เส้นค่าเฉลี่ยที่ใช้กันทั่วไปมีดังนี้

5	วัน (1 สัปดาห์)	ใช้สำหรับการลงทุนระยะสั้น
10	วัน (2 สัปดาห์)	ใช้สำหรับการลงทุนระยะสั้น
25	วัน (ประมาณ 1 เดือน)	ใช้สำหรับการลงทุนระยะค่อนข้างปานกลาง
75	วัน (ประมาณ 1 ไตรมาส)	ใช้สำหรับการลงทุนระยะกลาง
200	วัน (ประมาณ 1 ปี)	ใช้สำหรับการลงทุนระยะยาว

- Oscillators ดัชนีบ่งชี้ใช้ประโยชน์เมื่อราคาหุ้นไม่ได้เคลื่อนไหวเป็นตามแนวโน้มอย่างเด่นชัด เช่น ราคาหุ้นที่เคลื่อนไหวไปด้านข้าง และยังสามารถบอกถึงขอบเขตแนวที่มีการซื้อมากเกินไป (OverBought) และขอบเขตแนวที่มีการขายมากเกินไป (OverSold) เช่น Relative Strength Index (RSI) และ Moving Average Convergence Divergence (MACD)

โดยที่ RSI สามารถเป็นตัววัดความเร็วในการเคลื่อนไหว และทิศทางของราคาหุ้น ซึ่งจะมีค่าอยู่ระหว่าง 0 ถึง 100 ค่า RSI จะบอกถึงช่วงการซื้อขายหุ้น โดยถ้า RSI มากกว่า 70 จะเป็นการเข้าเขต Overbought ซึ่งมีความหมายว่าเป็นช่วงที่กำลังเข้าสู่แนวโน้มขาลง และหาก RSI มีค่าน้อยกว่า 30 จะเป็นการเข้าสู่เขต Oversold ซึ่งมีความหมายว่าเป็นช่วงที่กำลังเข้าสู่แนวโน้มขาขึ้น ในการคำนวณค่า RSI นั้นจะต้องทำการหาค่า RS ก่อน โดยมีการคำนวณดังนี้

$$RS = \frac{\text{Average Gain}}{\text{Average Loss}} \quad (2.2)$$

โดย Average Gain คือ ค่าเฉลี่ยของการเปลี่ยนแปลงราคาหุ้นที่เป็นบวก

Average Loss คือ ค่าเฉลี่ยของการเปลี่ยนแปลงราคาหุ้นที่เป็นลบ

จากนั้นนำค่า RS มาคำนวณหาค่า RSI

$$RSI = 100 - \frac{100}{(1 + RS)} \quad (2.3)$$

และ Moving Average Convergence Divergence (MACD) เป็นดัชนีบ่งชี้ชนิดหนึ่งที่สามารถบอกได้ทั้งสามารถทิศทาง และแรงส่งของราคาหุ้น ซึ่งเป็นการนำเอาลักษณะของกราฟเส้นค่าเฉลี่ยเคลื่อนที่ที่วาดขึ้นพร้อมๆ กัน 2 เส้นมาวิเคราะห์ลักษณะ โดยค่า MACD ถูกออกแบบมาเพื่อวิเคราะห์ความแตกต่างของค่าเฉลี่ยทั้งสองเส้น เป็นการนำเส้นค่าเฉลี่ยระยะสั้นเป็นตัวตั้งและลบด้วยเส้นค่าเฉลี่ยระยะยาว ระยะเวลาของเส้นค่าเฉลี่ยของแต่ละเส้นสามารถปรับได้แล้วแต่กลยุทธ์วิธีการของแต่ละบุคคล แต่โดยส่วนใหญ่นักลงทุนทั่วไปจะใช้ค่า EMA12 และ EMA26 เป็นหลัก

นัยสำคัญของค่า MACD มีดังนี้ หากค่า MACD เป็นบวกมีความหมายว่าแนวโน้มของราคาหุ้นมีโอกาสสูงที่จะมีทิศทางเป็นแนวโน้มขาขึ้น โดยที่เส้น EMA ที่เป็นเส้นระยะสั้นมีค่ามากกว่า และมีเส้นกราฟอยู่เหนือเส้น EMA ระยะยาว และค่า MACD มีค่าเป็นลบมีความหมายว่าแนวโน้มของราคาหุ้นมีโอกาสสูงที่จะมีทิศทางเป็นแนวโน้มขาลง โดยเส้น EMA ที่เป็นเส้นระยะสั้นมีค่าน้อยกว่า และเส้นกราฟจะอยู่ใต้เส้น EMA ที่เป็นเส้นระยะยาว มีสมการการคำนวณดังนี้

$$MACD = EMA_{(short)} - EMA_{(long)} \quad (2.4)$$

2.2 Artificial Intelligence

ปัญญาประดิษฐ์ (Artificial Intelligence: AI) คือ โปรแกรมที่ถูกเขียนและพัฒนาให้มีความฉลาด มีความสามารถคิด วิเคราะห์ วางแผน และตัดสินใจได้ โดยการประมวลผลจากฐานข้อมูลขนาดใหญ่ มากไปกว่านั้นยังสามารถดัดแปลงการประมวลผล หรือประยุกต์ให้เป็นไปตามสถานการณ์ต่างๆ เช่น Amazon Alexa และ Siri ดังนั้นงานประเภทใดก็ตามที่มีการทำงานเป็นรูปแบบ ก็สามารถถูกปัญญาประดิษฐ์แทนที่ได้ทั้งหมด ไม่ว่าจะเป็นการขับรถ งานบัญชี หรือการวิเคราะห์การเงิน สินเชื่อและการลงทุน หรือแม้แต่งานที่ซับซ้อน ต้องใช้การคิดวิเคราะห์ก็สามารถ

ถูกแทนที่ได้เช่นกัน คำนิยามของปัญญาประดิษฐ์ตามความสามารถที่มนุษย์ต้องการ แบ่งเป็น 4 กลุ่ม ดังนี้

2.2.1 การกระทำคล้ายมนุษย์ (Acting Humanly)

การสร้างเครื่องจักรที่ทำงานในสิ่งซึ่งอาศัยปัญญาเมื่อกระทำโดยมนุษย์ การสื่อสารด้วยภาษาที่มนุษย์ใช้ เช่น ภาษาไทย ภาษาอังกฤษ เช่น การแปลงข้อความเป็นคำพูด และการแปลงคำพูดเป็นข้อความ การมีประสาทสัมผัสคล้ายมนุษย์ เช่น คอมพิวเตอร์รับภาพได้โดยอุปกรณ์รับสัมผัส จากนั้นนำภาพไปประมวลผล รวมไปถึงการเคลื่อนไหวได้คล้ายมนุษย์ เช่น หุ่นยนต์ช่วยงานต่างๆ อย่างการดูดฝุ่น หรือเคลื่อนย้ายสิ่งของ

2.2.2 การคิดคล้ายมนุษย์ (Thinking Humanly)

กลไกของกิจกรรมที่เกี่ยวข้องกับความคิดมนุษย์ เช่น การตัดสินใจ การแก้ปัญหา และการเรียนรู้ นั้นต้องทำการศึกษาก่อนว่ามนุษย์มีกระบวนการคิดอย่างไร โดยการวิเคราะห์ลักษณะการคิดของมนุษย์นั้นเป็นศาสตร์ด้าน Cognitive Science เช่น ศึกษาโครงสร้างสามมิติของเซลล์สมอง การแลกเปลี่ยนประจุไฟฟ้าระหว่างเซลล์สมอง วิเคราะห์การเปลี่ยนแปลงทางเคมีไฟฟ้าในร่างกายระหว่างการคิด

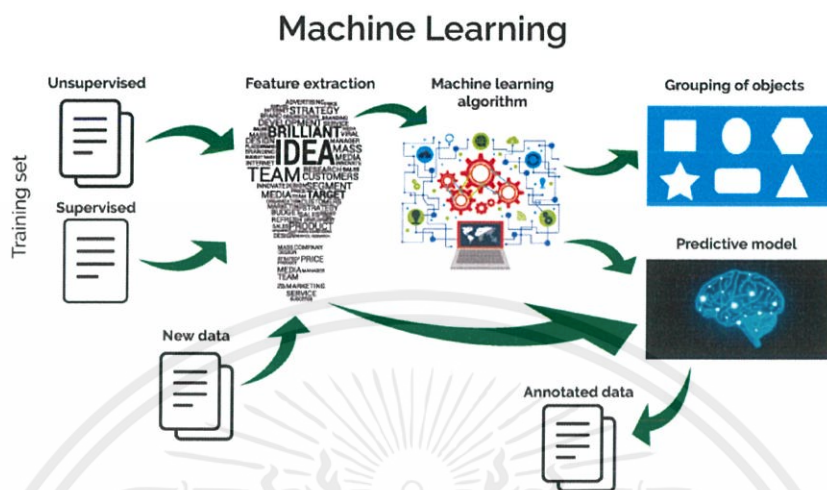
2.2.3 การคิดอย่างมีเหตุผล (Thinking Rationally)

การศึกษาความสามารถในด้านสติปัญญาโดยการใช้โมเดลการคำนวณ การใช้หลักตรรกศาสตร์ในการคิดหาคำตอบอย่างมีเหตุผล เช่น ระบบผู้เชี่ยวชาญ

2.2.4 การกระทำอย่างมีเหตุผล (Acting Rationally)

การกระทำเพื่อบรรลุเป้าหมายที่ได้ตั้งไว้ เช่น โปรแกรมเล่นเกมหมากรุกที่มีเป้าหมายว่าต้องเอาชนะคู่ต่อสู้ และระบบขับรถอัตโนมัติที่มีเป้าหมายว่าต้องไปถึงเป้าหมายในระยะทางที่สั้นที่สุด รวมไปถึงการเลือกเส้นทางที่ไปยังเป้าหมายที่สั้นที่สุด

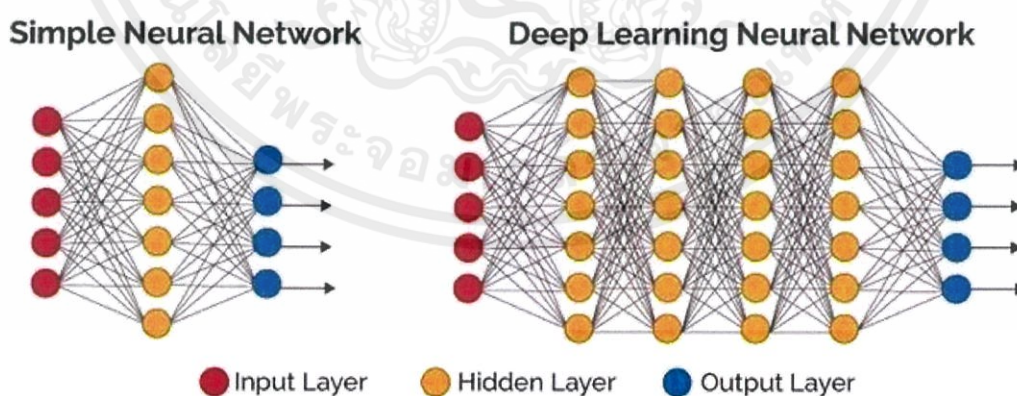
2.3 Machine Learning



รูปที่ 2.2 รูปแบบการประมวลผล Machine Learning [5]

Machine Learning เป็นการใช้อัลกอริทึม (Algorithm) ในการแยกแยะและวิเคราะห์ข้อมูล เรียนรู้จากข้อมูลนั้น เพื่อสร้างโมเดลในการตัดสินใจหรือคาดการณ์เกี่ยวกับบางสิ่งบางอย่าง ซึ่ง Machine Learning นั้นถูกใช้งานเสมือนเป็นสมองของปัญญาประดิษฐ์ อาจกล่าวได้ว่าปัญญาประดิษฐ์ใช้ Machine Learning ในการสร้างความฉลาด ซึ่งการสร้างโมเดลหรือรูปแบบการเรียนรู้จะมาจากข้อมูลที่มีอยู่โดยการเรียนรู้ด้วยตนเอง

2.4 Deep Learning



รูปที่ 2.3 การประมวลผล Deep Learning [6]

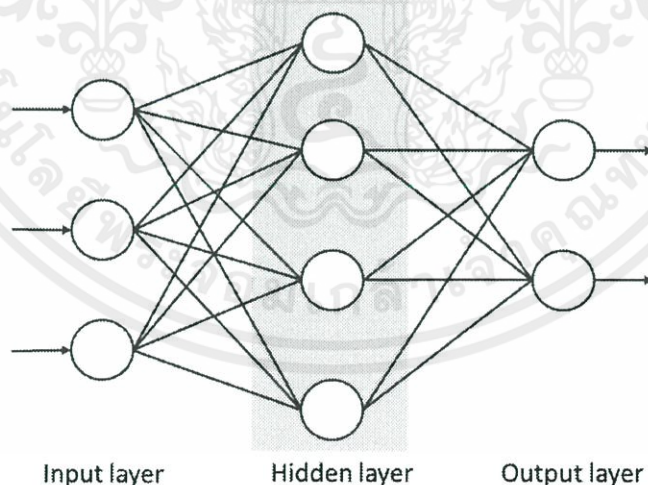
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Deep Learning หรือการเรียนรู้เชิงลึก คือชุดคำสั่ง (Algorithm) ที่ถูกสร้างขึ้นมาเพื่อการเรียนรู้ของเครื่องจักร (Machine Learning) หรือทำให้ตัวเครื่องจักรสามารถประมวลผลข้อมูลจำนวนมาก ด้วยการจำลองโครงข่ายประสาทเทียม (Artificial Neuron Network: ANN) ซึ่งเป็นเทคนิคที่เลียนแบบการทำงานของสมองมนุษย์ ที่ประกอบด้วยเซลล์ประสาท (Neuron) และแต่ละเซลล์จะถูกเชื่อมโยงกันเป็นโครงข่าย

โดยในซอฟต์แวร์แต่ละเซลล์ประสาทจะถูกเรียกว่า โหนด (Node) ซึ่งโครงข่ายประสาทเทียม ประกอบด้วย 3 ส่วน คือ Input Layer เป็นข้อมูล input จำนวนของโหนดขึ้นอยู่กับจำนวนของข้อมูล Hidden Layer เป็นชั้นที่อยู่กึ่งกลางระหว่าง Input Layer และ Output Layer จะเป็นส่วนการเรียนรู้ข้อมูลเชิงลึก ซึ่งการเพิ่มชั้นของ Hidden Layer และจำนวนโหนด จะส่งผลต่อการทำงาน และประสิทธิภาพในการเรียนรู้ของโมเดล และส่วนสุดท้าย Output Layer เป็นชั้นที่นำเอาข้อมูลจากการคำนวณไปใช้ จำนวนของโหนดในชั้นนี้ ขึ้นอยู่กับรูปแบบของ Output ที่ต้องการ

หรืออีกความหมายหนึ่ง Deep Learning คือ การจำลองโครงข่ายประสาทเทียมที่มีโหนดหลายๆ ชั้น และใช้การประมวลผลแบบขนาน (Parallel Processing) ทำให้สามารถประมวลผลได้ครั้งละจำนวนมาก ช่วยให้การเรียนรู้ของเครื่องสามารถให้ผลลัพธ์ในการตัดสินใจและคาดการณ์ได้ดีมากยิ่งขึ้น

2.5 Neural Network



รูปที่ 2.4 ลักษณะโครงข่ายประสาทเทียม [7]

โครงข่ายประสาทเทียม (Artificial Neural Network: ANN) หรือเรียกสั้นๆว่า ข่ายงานประสาท (Neural Network) เป็นระบบการประมวลผลที่สร้างเลียนแบบการทำงานของระบบสมองมนุษย์ เพื่อใช้ประโยชน์ในการคาดคะเนเหตุการณ์จากข้อมูลที่มีอยู่ เช่น การพยากรณ์อากาศ การพยากรณ์หุ้นในตลาดหลักทรัพย์ และนอกจากนี้ได้มีการพัฒนานำเอาโครงข่ายประสาทเทียมไปใช้ประโยชน์ในสาขาอื่นๆ มากมาย เช่น Image Processing และ Pattern Recognition โดยรูปแบบหลักของโครงข่ายประสาทเทียมแบ่งออกเป็นสามส่วนตามที่กล่าวไปข้างต้น ได้แก่ Input Layer, Hidden Layer และ Output Layer

Hidden layer จะปฏิบัติตนราวกับ Black Box คือ จะไม่รู้การทำงานว่าเป็นอย่างไร รู้แค่เพียงว่ามันจะทำการสุ่มตัวเลขมาบนแต่ละ Node และสร้างน้ำหนักให้กับแต่ละเส้นที่โยงถึงกัน ผลลัพธ์ที่ได้ในแต่ละ Node จะมาจากการคูณกันของ น้ำหนักในเส้นกับค่าในแต่ละ Node

$$hidden\ node = \sigma(wight * node\ value) + bias \quad (2.5)$$

2.5.1 รูปแบบการเรียนรู้ของโครงข่ายประสาทเทียม

ความสามารถในการเรียนรู้ของโครงข่ายประสาทเทียมขึ้นอยู่กับโครงการสร้างของโครงข่ายที่ออกแบบ และวิธีการในการฝึกอบรม (Training) โดยรูปแบบการเรียนรู้สามารถแบ่งวิธีการเรียนรู้ออกได้เป็น 3 แบบ คือ

1) การเรียนรู้แบบมีผู้สอน (Supervised Learning)

เป็นการเรียนรู้โดยอาศัยข้อมูลและ Label จากข้อมูลสำหรับการเรียนรู้ ได้ผลลัพธ์ออกมาเป็นการแบ่งประเภทข้อมูล (Classification) และการวิเคราะห์การถดถอย (Regression)

2) การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning)

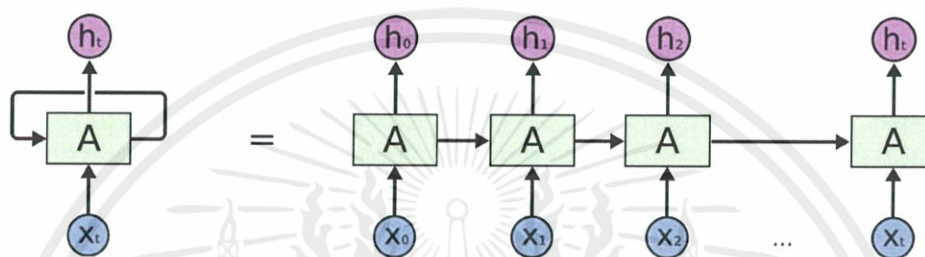
เป็นการเรียนรู้ซึ่งกำหนดการกระทำของระบบจากสิ่งที่สังเกตได้ หรือเรียนรู้จากสิ่งแวดล้อมรอบตัว โดยอัลกอริทึมของการเรียนรู้พยายามจะหารูปแบบหรือสิ่งที่เชื่อมโยงกันในการกระทำในสถานะนั้นๆ ซึ่งเป็นการเรียนรู้ที่มีความแตกต่างไปจากการเรียนรู้แบบมีผู้สอนตรงที่ว่า คอมพิวเตอร์จะไม่ใช่รู้ว่าอะไรถูกอะไรผิด กล่าวคือไม่มีการระบุอย่างชัดเจนว่าการกระทำใดดีและการกระทำใดไม่ดี

3) การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning)

การเรียนรู้ไม่ได้มี Label หรือสัญลักษณ์ที่บอกว่าข้อมูลนั้นคืออะไร แต่การเรียนรู้จัดข้อมูลนำเข้า (input) จัดเป็นกลุ่ม (Cluster) บนพื้นฐานของความเหมือน (Similarities) และความแตกต่าง (Differences) ระหว่างรูปแบบของข้อมูลนำเข้า ตัวอย่างเช่น การหาโครงสร้างที่ซ่อนอยู่ในข้อมูล เป็นต้น

2.5.2 Recurrent Neural Network

Recurrent Neural Network (RNN) โครงข่ายประสาทเทียมแบบย้อนกลับ คำว่า Recurrent จะหมายถึง การหวนย้อนคืนกลับมาอีก การทำงานของ RNN เป็นการนำผลลัพธ์ที่ได้จากการคำนวณย้อนกลับมาใช้เป็นข้อมูลเข้าอีกครั้ง ซึ่งนิยมใช้กับข้อมูลที่มีความต่อเนื่อง หรือใช้สำหรับการประมวลผลลำดับ (sequence) เช่น Time Series ข้อมูลเสียง ข้อความ หรือแม้แต่รูปภาพ



รูปที่ 2.5 รูปแบบการประมวลผลแบบโครงข่าย RNN [9]

Node ของ RNN จะมีข้อมูลขาเข้าสองอย่างได้แก่ Input ณ Node นั้นๆ และผลลัพธ์ที่ได้จากการคำนวณใน Node ก่อนหน้า ซึ่งทั้งสองข้อมูลจะถูกนำมารวมเข้าด้วยกันและออกผลลัพธ์มาเป็นสองทางคือ ผลลัพธ์ที่ออกจาก Node นั้นๆ และออกเพื่อไปเข้าเป็นข้อมูลขาเข้าใน Node ถัดไป

ข้อดีของ RNN คือ มีการใช้ข้อมูลก่อนหน้าในการทำนายสิ่งที่จะเกิดขึ้นในอนาคต ซึ่งหมายถึง อะไรที่เคยเกิดขึ้นในอดีตย่อมส่งผลต่อเหตุการณ์ที่จะเกิดขึ้นในอนาคตด้วย

ข้อเสีย คือ สามารถดูย้อนกลับได้แค่เพียงในช่วงระยะเวลาสั้นๆ เท่านั้น ซึ่งปัญหาหลักๆ ของ RNN นั้นเกิดมาจากค่าความผิดพลาดโดยรวม (Gradient) ที่เริ่มน้อยลงในข้อมูลที่มีความยาวมากขึ้น จนแทบจะไม่สามารถเห็นความเปลี่ยนแปลงของ Gradient ได้เลย ซึ่งปัญหานี้ถูกเรียกว่า Vanishing Gradient Problem (VGP)

Gradient คือ ค่าความผิดพลาดโดยรวม ที่เป็นตัวบอกว่าการเปลี่ยนแปลงของค่าเริ่มต้นส่งผลต่อผลลัพธ์ที่เกิดขึ้น และ Vanishing Gradient Problem คือค่า Gradient ที่ส่งผ่านจากชั้น output กลับเข้ามาน้อยลงเรื่อยๆ ทำให้การฝึกชั้นแรกๆ ใช้เวลานาน ส่วนหนึ่งมีผลมาจากค่าอนุพันธ์ของฟังก์ชันช่วย (Activation Function) ซึ่งมันมีผลต่อการเรียนรู้ของทั้งระบบ

ซึ่งกระบวนการคำนวณย้อนหลัง (Back-Propagation) เป็นการคำนวณเพื่อทำการปรับค่าน้ำหนักในแต่ละเส้นให้มีความสมดุลมากขึ้น ซึ่งช่วยให้โครงข่ายประสาทเทียมสามารถเรียนรู้ข้อมูลได้อย่างถูกต้องและแม่นยำมากขึ้น

ก่อนที่จะตรวจสอบน้ำหนักย้อนกลับโดยใช้ Back-Propagation สิ่งแรกที่ต้องทำก็คือ ต้องคำนวณไปจนเจอผลลัพธ์ (Forward Pass) จากนั้นค่อยนำผลลัพธ์ของความผิดพลาดที่ได้มาทำการเปลี่ยนแปลงค่าน้ำหนักใหม่ ซึ่งการทำย้อนกลับ (Backward Pass)

ในการคำนวณไปข้างหน้านั้น จะสนใจเพียงสองค่าหลักๆ คือ Total Net Input ซึ่งเป็นการคำนวณน้ำหนักคูณถ่วงกับค่าของแต่ละ Node และ Output ที่เป็นการคำนวณค่าตอบของ Node ที่ต้องการโดยใช้ Activation Function เข้ามาช่วย

$$net = \sigma(wight \times node\ value) + bias \quad (2.6)$$

$$out = activation_function(net) \quad (2.7)$$

เมื่อการคำนวณได้ Node ของผลลัพธ์ จะสามารถคำนวณค่าความผิดพลาดโดยรวมทั้งหมดของทุกผลลัพธ์ที่ออกมาได้โดยใช้สูตร Square Error มาช่วยในการคิดค่าความผิดพลาดโดยรวม (Gradient)

$$E_{total} = \sigma\left(\left(1/2\right) \times (target - output)^2\right) \quad (2.8)$$

สมการ Back-Propagation ของ RNN เป็นการใช้หลักการ Chain Rule ธรรมดาๆ เพื่อหาค่าตอบที่ต้องการในด้านซ้าย

$$\frac{\partial E}{\partial w} = \sum_{k=1}^t \frac{\partial E_t}{\partial o_t} \frac{\partial o_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial w} \quad (2.9)$$

โดยที่ E คือ ค่าความผิดพลาดโดยรวม

o คือ ผลลัพธ์ที่ได้ในแต่ละ Node

h คือ Total Net Input

w คือ น้ำหนัก

และมีหลักการ ดังนี้

$\frac{\partial E_t}{\partial o_t}$ คือ การคำนวณว่าค่าความผิดพลาดที่เกิดขึ้นมีผลต่อค่าของผลลัพธ์ที่ได้มาก

น้อยเพียงใด

$\frac{\partial o_t}{\partial h_t}$ คือ การคำนวณความเปลี่ยนแปลงของ Total Net Input ในผลลัพธ์ที่ออกไปใน

แต่ละ Node

$\frac{\partial h_t}{\partial h_k}$ คือ การคำนวณการเปลี่ยนแปลงของ Total Net Input ใน Node หนึ่งๆ ซึ่ง

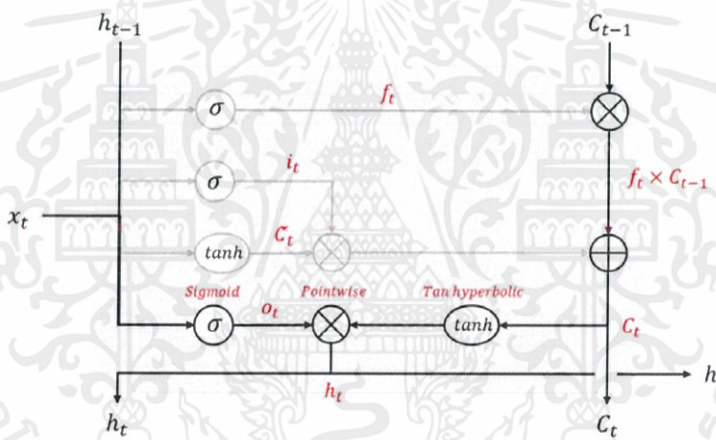
อาจจะส่งผลต่อ Total Net Input ในอีก Node หนึ่ง

$\frac{\partial h_k}{\partial w}$ คือ การคำนวณค่า Net Input ที่อาจจะส่งผลต่อการเปลี่ยนแปลงในน้ำหนัก

ท้ายที่สุดแล้ว Gradient นั้นเมื่อถูกคำนวณไปนานๆ ค่าจะยิ่งลดน้อยลงจนแทบจะไม่เห็นการเปลี่ยนแปลงใดๆ แม้ข้อมูลที่เข้ามาจะมีการเปลี่ยนแปลงไปมากแค่ไหนก็ตาม เหตุผลหลักที่ทำให้ Gradient ถดถอยลงไปจนแทบไม่เห็นความเปลี่ยนแปลงนั้นก็มาจากตัว Activation Function

2.5.3 Long Short-Term Memory Network

Long Short-Term Memory Network (LSTM) เป็นโครงข่ายประสาทที่พัฒนามาจากโครงข่าย RNN ซึ่งสืบเนื่องจากปัญหาที่เกิดขึ้นใน RNN เกี่ยวกับค่า Gradient ที่มีค่าน้อยลงจากการทำงานของ Back-Propagation ซึ่งจะใช้หลักการคล้ายๆ เดิม แต่เปลี่ยนตัวฟังก์ชันด้านในให้มีความเสถียรและมีประสิทธิภาพมากขึ้น เรียกได้ว่า LSTM คือ RNN ที่มี Memory Cell State หรือ เซลล์จดจำ



รูปที่ 2.6 รูปแบบการประมวลผลโครงข่าย LSTM [7]

โครงข่าย LSTM นั้นจะแตกต่างจากโครงข่าย RNN คือ จะมีการใช้ฟังก์ชันพิเศษเพื่อคัดแยกข้อมูลหรือมีการพิจารณาข้อมูลก่อนที่จะนำไปประมวลผลในขั้นต่อไป ฟังก์ชันเหล่านี้จะเป็นสิ่งที่คัดเลือกข้อมูลที่จะนำเข้าในการประมวลผล ซึ่งตัวแปรต่างๆ มีความหมายดังนี้

Forget Gate (f_t) คือ เกตที่จะกำหนดว่าข้อมูลที่ได้รับมานั้นควรจะ ผ่านออกไปได้หรือไม่ ซึ่งข้อมูลที่ใช้ในการตัดสินใจนั้นมาจากอินพุตของ Node นั้นๆ ร่วมกับผลลัพธ์ ที่ได้จาก Node ก่อนหน้าและผ่าน Sigmoid Function และคูณด้วยสถานะของ Node ก่อนหน้า ผลลัพธ์ได้ของเกตนี้จะอยู่ที่ $[0,1]$ ซึ่งค่า 0 หมายถึงไม่มีข้อมูลใดที่จะสามารถไหลผ่านไปได้เลย ในขณะที่ 1 หมายถึงปล่อยให้ข้อมูลที่เข้ามาไหลผ่านไปได้ทั้งหมด ซึ่งมีสมการ คือ

$$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.10}$$

Input Gate (i_t) คือ เกตต์ที่เปิดรับข้อมูลอินพุตที่เข้ามาใหม่จากนั้นจะบันทึกลงในแต่ละ Node

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.11)$$

Cell State (C_t) คือ ข้อมูลอินพุตที่ถูกคูณด้วยฟังก์ชัน tan Hyperbolic จากนั้นจะถูกลำดับกับข้อมูล Input Gate และต่อด้วยรวมกับข้อมูลแรกจาก Forget Gate ค่าสุดท้ายที่ได้จะออกมาเป็นค่าสถานะใหม่ของ Node นั้นๆ ซึ่งผ่านการอัปเดตสถานะจาก Node ก่อนหน้า

$$C_t = \tanh(w_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.12)$$

Output Gate (o_t) คือ เกตต์ที่ใช้บอกว่าข้อมูลเตรียมพร้อมที่จะเป็นข้อมูลขาออกแล้วหรือไม่ ซึ่งข้อมูลขาออกของแต่ละ Node ใน LSTM มีทั้งหมด 3 จุด แต่มีค่าตัวเลขเพียง 2 ค่า ได้แก่ ค่าที่เกิดจากการคำนวณโดยการอัปเดตสถานะแล้ว (จากขั้นตอนการอัปเดต Cell State) และอีกค่าหนึ่งคือ ค่าของข้อมูลขาเข้าที่ผ่านการคำนวณใน Output Gate ซึ่งค่านี้จะถูกส่งต่อเป็นข้อมูลอินพุตของ Node ถัดไป และเป็นผลลัพธ์ของ Node นั้นๆ

$$o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.13)$$

Hidden state (h_t) คือ ค่าของ o_t ที่จะถูกนำเข้าไปคูณกับฟังก์ชัน tan Hyperbolic ผลลัพธ์ที่ได้จะถูกใช้เป็น 2 ข้อมูล คือ ผลลัพธ์ของ Node และเป็นข้อมูลที่จะเป็นอินพุตของ Node ถัดไป

$$h_t = o_t \cdot \tanh(C_t) \quad (2.14)$$

2.5.4 Activate Function

Activation Function ทำหน้าที่เป็นฟังก์ชันช่วย ซึ่งจะทำให้ผลลัพธ์ที่เป็นข้อมูลขาออกที่มาจากแต่ละ Node มีประสิทธิภาพมากขึ้น โดยฟังก์ชันช่วยจะช่วยให้การจำกัดขอบเขตของค่าที่เป็นไปได้ให้น้อยลงจนสามารถเห็นขอบเขตได้ เช่น หากค่าผลลัพธ์ออกมาเป็นอะไรก็ได้ตั้งแต่ $(-\infty, \infty)$ จะหมายถึงการที่ข้อมูลไร้ซึ่งขอบเขต และยากต่อการคำนวณใน Node ต่อๆ ไป Activation Function ที่ใช้ในการสร้าง Model ในปริณญาณิพนธ์นี้ได้แก่

1) Linear Function

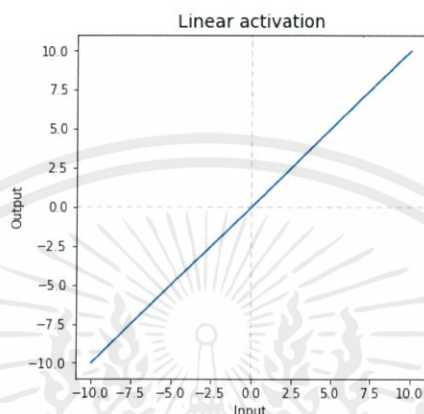
สมการของการทำ Linear

$$f(x) = ax + b \quad (2.15)$$

ในการทำ Back-Propagation จำเป็นต้องทำการคำนวณย้อนหลังซึ่งต้องใช้หลักการอนุพันธ์ (Derivative) ในการหาค่าตอบด้วย ซึ่งค่าอนุพันธ์ของสมการ Linear นั้นเป็นค่าคงที่

$$f'(x) = a \quad (2.16)$$

นั่นหมายความว่า ไม่ว่าผลลัพธ์ที่ทำนายมาจะผิดหรือจะถูกก็ตาม หากคำนวณย้อนหลังกลับไป ค่าที่ได้ออกมาก็จะคงที่ โดยไม่ได้สนใจว่าค่าจะเปลี่ยนมากหรือเปลี่ยนน้อยแค่ไหน ซึ่งไม่ส่งผลดีต่อผลลัพธ์ที่ต้องการในอนาคต

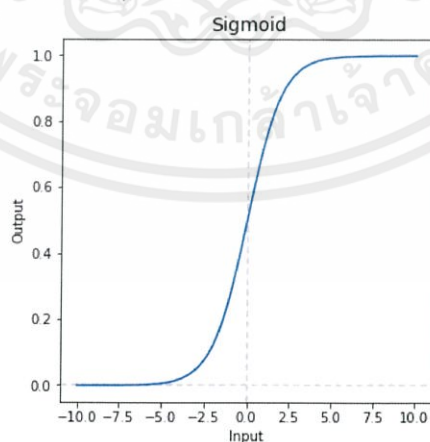


รูปที่ 2.7 ช่วงผลลัพธ์ของ Linear Activation [11]

2) Logistic Activation Function (Sigmoid: σ)

เมื่อ Linear ไม่ได้เป็นผลลัพธ์ที่ดีเสมอไป และช่วงคำตอบของการใช้ Linear Function นั้นเป็นไปได้ตั้งแต่ $(-\infty, \infty)$ ซึ่งกว้างมาก การใช้ Sigmoid Function จึงเหมือนตัวช่วยที่ช่วยร่นระยะของข้อมูลให้อยู่ในช่วง $[0, 1]$ เท่านั้น และที่สำคัญ คือ ช่วยให้ค่า Gradient มีความเรียบมากขึ้น (Smooth Gradient) ไม่แข็งเหมือน Step Function และใน Linear Function

$$\sigma(x) = 1 / (1 + \exp(-x)) \quad (2.17)$$

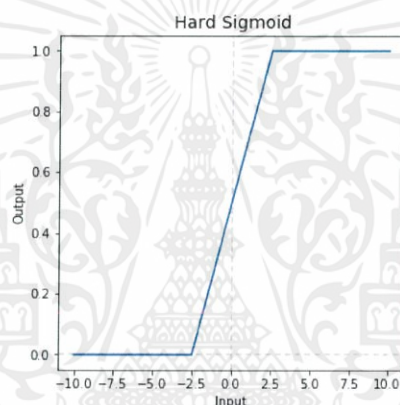


รูปที่ 2.8 ช่วงผลลัพธ์ของ Sigmoid Activation [8]

แม้ Sigmoid Function จะได้รับความนิยมในการคำนวณของโครงข่ายประสาทเทียม แต่ก็มีข้อเสียอยู่เหมือนกัน และตัวของ Sigmoid Function นี้เป็นต้นเหตุของปัญหา VGP ซึ่งเกิดจากการที่ไม่เห็นความเปลี่ยนแปลงของ Gradient

ซึ่งในการคำนวณย้อนหลังหลายๆ รอบของ Sigmoid Function นั้นยิ่งทำให้ค่า Gradient ลดลงไปตามระยะทางที่ต้องผ่าน ซึ่งหมายความว่า แม้ค่าเริ่มต้นจะเปลี่ยนแปลงไปมากหรือน้อยแค่ไหนก็ตาม ค่า Gradient เองก็ไม่ได้มีความเปลี่ยนแปลงไปจากเดิมมากสักเท่าไร ซึ่งก็จะทำให้โครงข่ายประสาทเทียมไม่สามารถเรียนรู้ได้อย่างมีประสิทธิภาพเท่าที่ควร

Hard Sigmoid มีความคล้ายคลึงกับ Sigmoid Function มีจุดประสงค์ คือ เพื่อให้ค่าความน่าจะเป็น (ซึ่งจะเป็นการจำกัดให้ค่าอยู่ระหว่าง 0 ถึง 1) สำหรับใช้ในการสุ่มค่าพารามิเตอร์ของโครงข่ายประสาทเทียม ใช้ในการคำนวณที่ต้องการความรวดเร็วมากกว่าความแม่นยำ



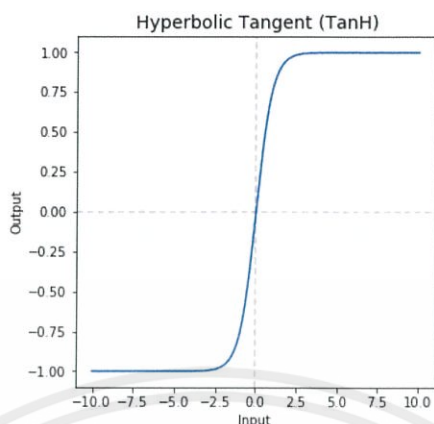
รูปที่ 2.9 ช่วงผลลัพธ์ของ Hard Sigmoid Activation [8]

3) Tan Hyperbolic Function (\tanh)

ปัญหาของ Sigmoid อาจอยู่ช่วงของมันมีค่าเล็กจนเกินไปจนทำให้เห็นความเปลี่ยนแปลงใน Gradient ได้น้อย \tanh จึงได้ทำการแก้ไขปัญหานั้นด้วยการปรับระยะใหม่ของ Sigmoid ให้มีช่วงกว้างมากขึ้นจาก $[0, 1]$ ใน Sigmoid ให้กลายเป็น $[-1, 1]$

$$f(x) = (\exp(z) - \exp(-z)) / (\exp(z) + \exp(-z)) \quad (2.18)$$

ระยะที่กว้างขึ้นใน \tanh ทำให้ Gradient ใน \tanh สามารถสังเกตได้ง่ายและเห็นความเปลี่ยนแปลงได้ดีขึ้นกว่าใน Sigmoid Function

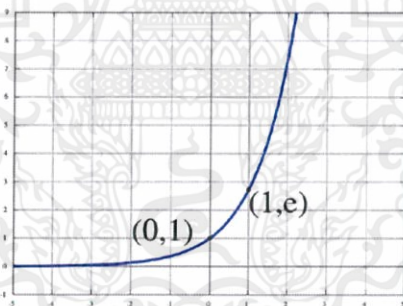


รูปที่ 2.10 ช่วงผลลัพธ์ของ Tan Hyperbolic Activation [8]

4) Exponential Activation

Exponential Function หรือสมการเลขชี้กำลัง จะเป็นการประมวลผลโดยนำค่าที่รับเข้ามาไปเป็นเลขชี้กำลังของ e โดยที่ e มีค่าประมาณเท่ากับ 2.71828 โดยมีรูปสมการดังนี้

$$f(x) = \exp(x) \quad (2.19)$$

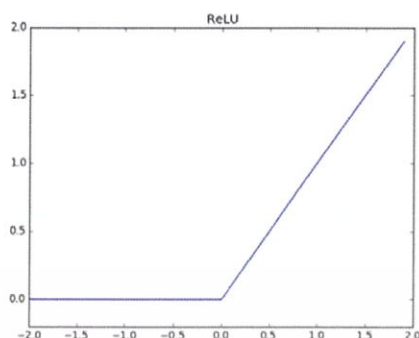


รูปที่ 2.11 ช่วงผลลัพธ์ของ Exponential Activation [10]

5) Rectified Linear Unit (ReLU)

จะเป็นฟังก์ชันที่ประมวลผลแบบมีเงื่อนไข คือ หากค่าที่รับเข้ามามีค่าน้อยกว่า 0 ผลลัพธ์ที่ออกมาจะเป็น 0 แต่ถ้าหากค่าที่รับเข้ามามีค่ามากกว่า 0 ผลลัพธ์ที่ออกมาจะเป็นค่าตัวมันเอง ReLU Function มีรูปสมการดังนี้

$$f(x) = \max(0, x) \quad (2.20)$$



รูปที่ 2.12 ช่วงผลลัพธ์ของ Rectified Linear Unit Activation [8]

2.4.5 Optimizer

Optimization Algorithms เป็นอัลกอริทึมที่ใช้ในการปรับค่าพารามิเตอร์ที่มีผลกับ อัตราการเรียนรู้ที่อยู่ภายใน Model ให้เหมาะสม

1) Adagrad (Adaptive Gradient Algorithm)

เป็นอัลกอริทึมที่จะมีการปรับอัตราการเรียนรู้สำหรับแต่ละพารามิเตอร์แยกกัน ซึ่งจะปรับให้สัมพันธ์กับความถี่ที่พารามิเตอร์ได้รับการอัปเดตค่าในระหว่างการฝึกอบรม (training) โดยยิ่งพารามิเตอร์ได้รับการอัปเดตมากเท่าไรอัตราการเรียนรู้ก็จะยิ่งน้อยลงเท่านั้น

2) AdaDelta

พัฒนาจาก Adagrad ที่มีข้อเสียคือหากเรียนรู้มากขึ้นเรื่อยๆ อัตราการเรียนรู้จะยิ่งลดลง โดยใช้ moving window ในการอัปเดตค่า Gradient แทนที่จะสะสม Gradient ไว้ทั้งหมด

3) Adam (Adaptive Moment Estimation)

เป็นอัลกอริทึมที่นำหลักการของโมเมนตัมมาใช้ในการปรับอัปเดตค่าการถ่วงน้ำหนักของโครงข่ายในการฝึกอบรม (training)

2.6 การวัดความแม่นยำของการคาดการณ์ผล

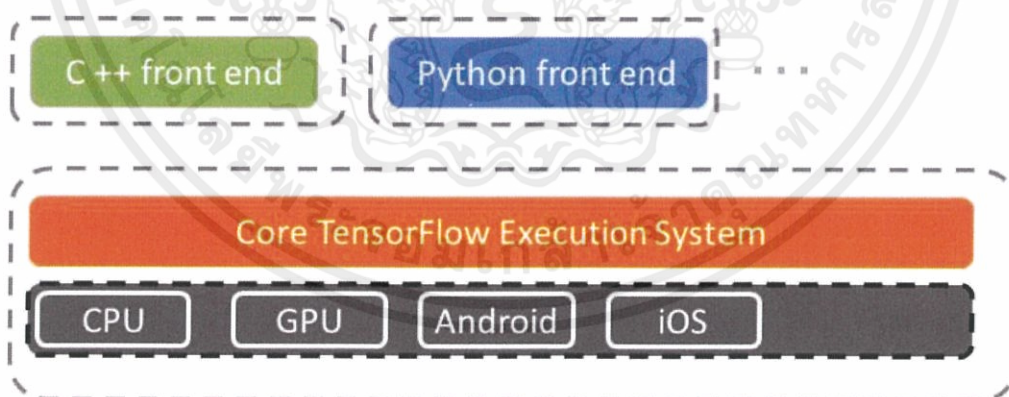
ค่าเฉลี่ยของรากที่สองของกำลังสองของความคลาดเคลื่อน (Root Mean Square Error: RMSE) สามารถใช้เป็นตัววัดความแม่นยำของการคาดการณ์หรือการทำนายผลได้ โดยถ้าค่า RMSE ที่ได้จากการประมวลผลมีค่าน้อยหมายถึง Model มีความแม่นยำ มีการคำนวณดังนี้

$$RMSE = \sqrt{\frac{\sum (A_t - P_t)^2}{n}} \quad (2.22)$$

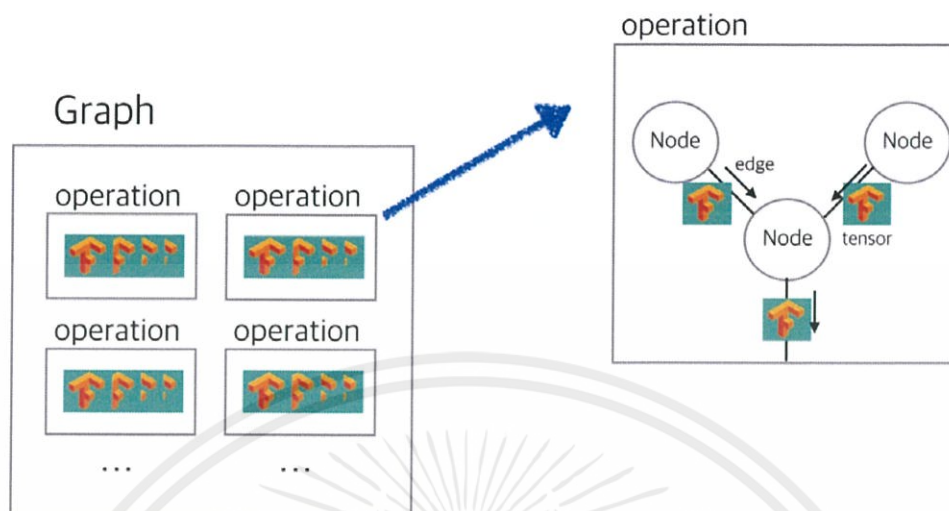
โดยที่ A_t คือ ค่าข้อมูลจริง
 P_t คือ ค่าที่ได้จากการคาดการณ์
 n คือ จำนวนข้อมูลทั้งหมด

2.7 TensorFlow

TensorFlow เป็นไลบรารีที่ใช้ในการพัฒนา Machine Learning ที่พัฒนาโดยบริษัท Google และได้ทำการเปิดตัวเมื่อวันที่ 11 กุมภาพันธ์ 2017 ซึ่ง TensorFlow นั้นจะเป็น Open Source ที่ใช้ภาษา Python ในการเขียนโปรแกรม ซึ่งรองรับทั้ง Python2 และ Python3 โดย TensorFlow สามารถทำงานบน CPU และ GPUs อีกทั้งยังรองรับระบบปฏิบัติการต่างๆ ได้แก่ Linux, macOS, Windows และ Android



รูปที่ 2.13 TensorFlow Architecture [17]



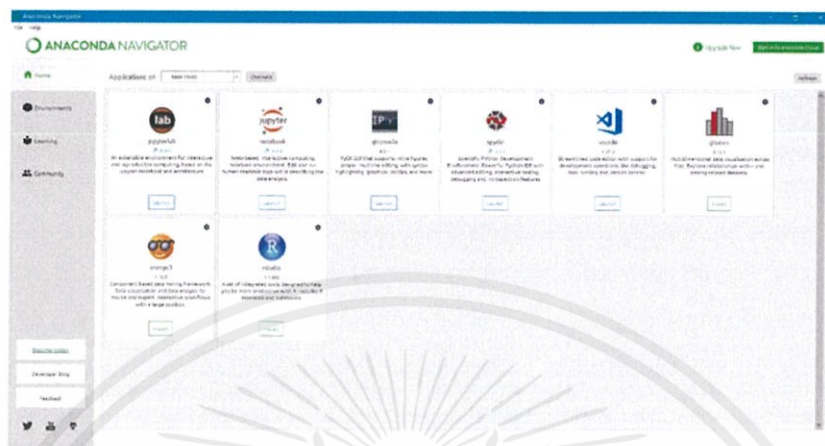
รูปที่ 2.14 แบบจำลองการประมวลผลของ TensorFlow [17]

นอกจากนี้ TensorFlow ยังรองรับ Deep Learning และมีระบบที่ใช้สำหรับคำนวณตัวเลข จากการใช้ Data Flow Graph ซึ่งแต่ละ Node จะแทนด้วยการคำนวณทางคณิตศาสตร์ ในขณะที่ edge จะแทนข้อมูลอาร์เรย์แบบหลายมิติ (Matrix) ซึ่งทั้งสองอย่างนี้จะทำงานเชื่อมระหว่างกัน ซึ่งอัลกอริทึมสำหรับการทำ Machine Learning แบบ Gradient-Based Algorithm นั้นก็ได้รับประโยชน์จากความสามารถ Auto-Differentiation และระบบ Optimizer ของ TensorFlow

2.8 Keras

Keras คือ Library ที่ใช้ในกระบวนการ Deep Learning Python และยังเป็น API (Application Programming Interface) ชั้นสูง โดยจะทำให้การเข้าถึงข้อมูล และการสร้าง Model ทำได้ง่ายขึ้น และ Keras สามารถรันบน CPU และเพิ่มประสิทธิภาพโดยการรันบน GPU ได้ การทำงานของ Keras จะทำงานบน TensorFlow หรือ Theano

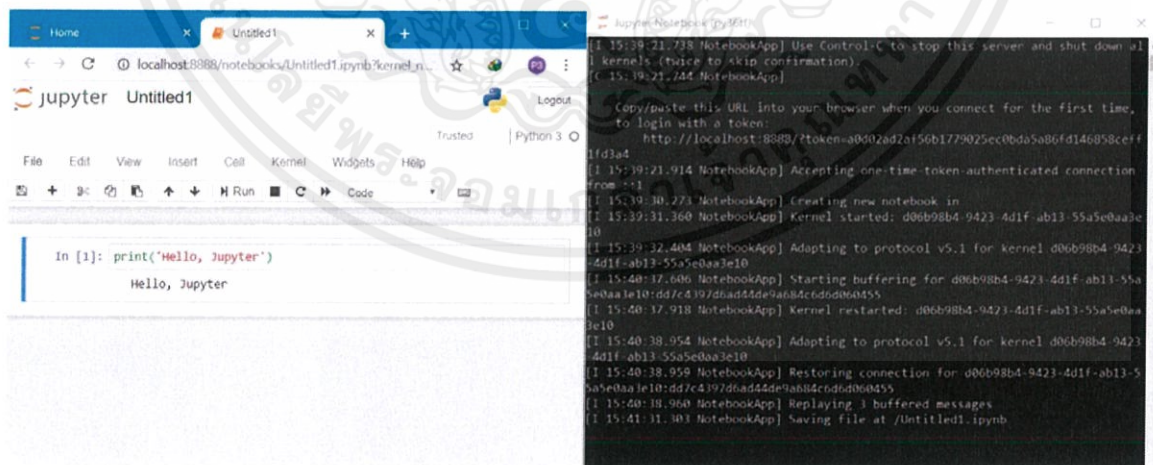
2.9 Anaconda



รูปที่ 2.15 Anaconda Navigator

Anaconda โปรแกรมที่ช่วยในการจัดการ Python ซึ่งจากเดิมที่เป็นตัวโปรแกรมพื้นฐานของ Python นั้นจะต้องทำการติดตั้ง Library ที่ต้องการใช้งานด้วยตนเอง โปรแกรม Anaconda นั้นเป็นโปรแกรมที่ทำให้การติดตั้ง Library โมดูลหรือเครื่องมือต่างๆ สะดวกและง่ายขึ้น และยังใช้งานได้ทั้ง Python2 Python3 ทั้งยังรองรับระบบปฏิบัติการ Windows, Mac OS และ Linux

2.9.1 Jupyter Notebook

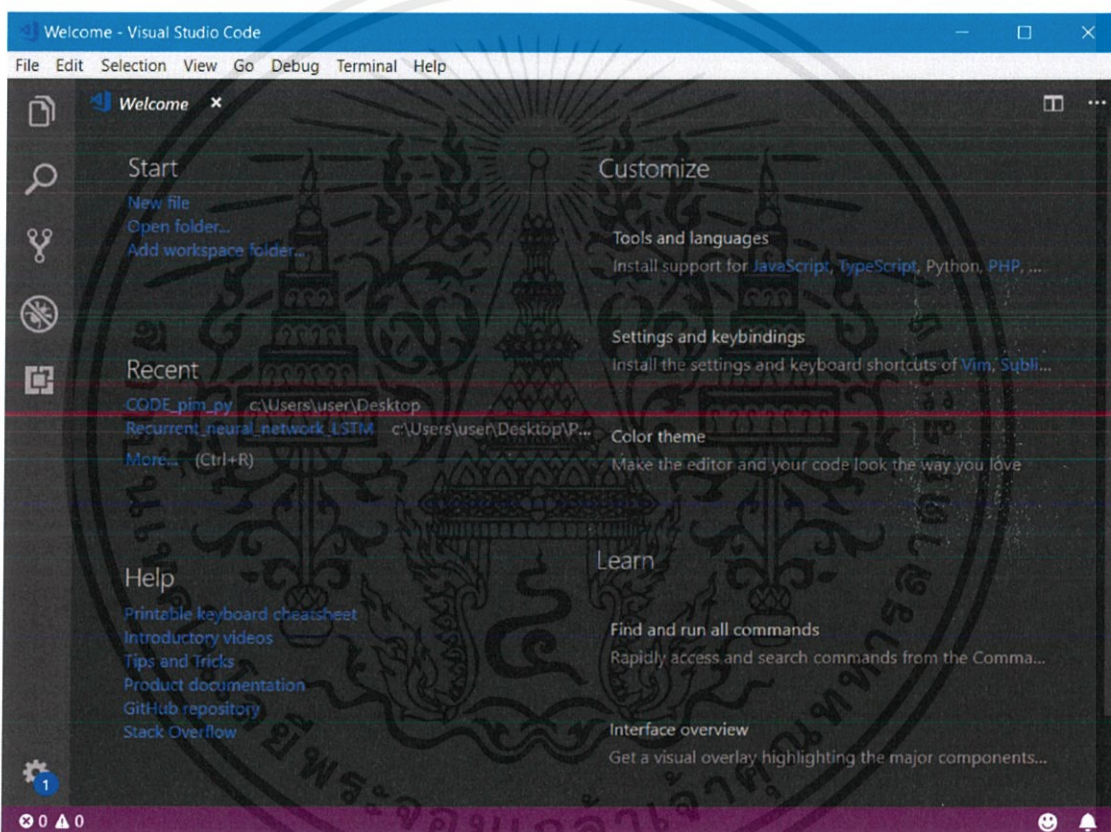


รูปที่ 2.16 โปรแกรม Jupyter Notebook และหน้าต่าง Terminal แสดงการประมวลผลของโปรแกรม Jupyter Notebook

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Jupyter Notebook ชื่อเดิมคือ IPython เป็นโปรแกรมที่ใช้เขียนภาษา Python ซึ่งถูกพัฒนาขึ้นโดยการนำเอาข้อดีของโหมดโต้ตอบ (Interactive Mode) และโหมดสคริปต์ (Script Mode) รวมเข้าไว้ด้วยกัน โดยโปรแกรม Jupyter Notebook นั้นจะถูกติดตั้งพร้อมกับโปรแกรม Anaconda และ Jupyter Notebook ยังสามารถทำ Markdown ที่ใช้เขียนคำอธิบายโค้ดหรือคำบรรยายต่างๆ ได้โดยที่ไม่ส่งข้อความไปให้ Python ประมวลผล ในปฏิญญาพันธันนี้ใช้ Jupyter Notebook ในการจัดเรียงข้อมูล เนื่องจากสามารถบอกรายละเอียดของข้อมูลได้ดีกว่า

2.9.2 Visual Studio Code



รูปที่ 2.17 โปรแกรม Visual Studio Code

Visual Studio Code หรือ VSCode เป็นโปรแกรมใช้เขียนภาษา Python ที่ถูกติดตั้งพร้อมกับโปรแกรม Anaconda เช่นเดียวกับกับ Jupyter Notebook ซึ่ง VSCode เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขและปรับแต่งโปรแกรมต่างๆ ได้ในปฏิญญาพันธันนี้จะใช้ VSCode ในส่วนแสดงผลของ GUI ซึ่งทำได้ดีกว่า Jupyter Notebook

2.10 Python

Python เป็นภาษาโปรแกรมมิ่งภาษาหนึ่งที่เป็นโอเพนซอร์ซ (Open Source) สามารถนำมาพัฒนาโปรแกรมได้หลากหลายลักษณะโดยที่ไม่ต้องเสียค่าใช้จ่าย ใช้งานได้ทั้งบนระบบยูนิกซ์ ลินุกซ์ (Unix Linux) และวินโดวส์ (Windows) มีรูปแบบการประมวลผลไปทีละบรรทัด และปฏิบัติตามคำสั่งที่ได้รับ Python เวอร์ชันปัจจุบันคือ 3.7 ปรินูญาพนธ์นี้ใช้เวอร์ชัน 3.66 เนื่องจากข้อจำกัดของ TensorFlow ยังไม่รองรับเวอร์ชัน 3.7

2.10.1 Package Python ที่ใช้งาน

- 1) Glob เป็นหนึ่งในไลบรารีมาตรฐานของ Python ซึ่งจะทำให้สามารถค้นหาไฟล์ตามรูปแบบที่กำหนดได้
- 2) Os เป็นไลบรารีที่มีไว้สำหรับจัดการกับสิ่งที่เกี่ยวข้องกับระบบปฏิบัติการ ซึ่งเป็นจุดรวมฟังก์ชันให้เรียกใช้งานที่เกี่ยวกับการติดต่อขอใช้บริการผ่านการควบคุมของ Operation System (OS)
- 3) NumPy เป็นนิยามออบเจกต์ชนิดแถวอาเรย์ที่สามารถคำนวณเมทริกซ์ใช้ในการคำนวณได้อย่างมีประสิทธิภาพ
- 4) Matplotlib เป็นไลบรารีสำหรับวาดกราฟที่ใช้งานได้หลากหลายยืดหยุ่น
- 5) Pandas เป็นไลบรารีที่มี Data Structure Function สำหรับการจัดการข้อมูลได้ต่างๆ
- 6) Math เป็นไลบรารีที่เก็บฟังก์ชันที่เกี่ยวกับการคำนวณทางคณิตศาสตร์
- 7) TensorFlow เป็นไลบรารีที่ใช้สำหรับพัฒนา Machine Learning
- 8) Keras เป็นไลบรารีที่ใช้ในการประมวลผล Deep Learning หากติดตั้งทั้ง TensorFlow และ Keras แล้วเมื่อเรียกใช้ Keras โปรแกรมจะให้ TensorFlow ทำงานเป็น Backend โดยอัตโนมัติ
- 9) Scikit-Learn เป็นไลบรารีสำหรับทำ Machine Learning
- 10) Datetime เป็นไลบรารีที่ใช้ในการเรียกข้อมูลวันและเวลาจากคอมพิวเตอร์
- 11) Tkinter เป็นแพ็คเกจ GUI (Graphical User Interface) มาตรฐานของ Python ซึ่งจะเป็นส่วนที่โต้ตอบกับผู้ใช้งาน

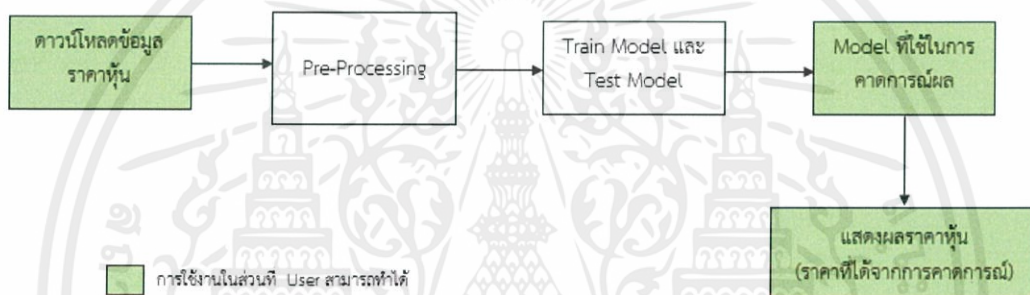
บทที่ 3

การออกแบบและการจัดทำปริญญานิพนธ์

3.1 การออกแบบ

3.1.1 การออกแบบการทำงานของระบบ

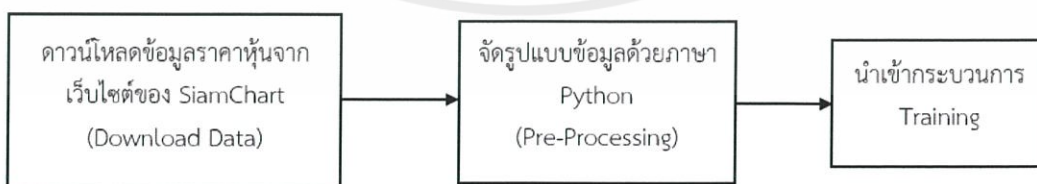
ระบบสนับสนุนการตัดสินใจในการซื้อขายหุ้นด้วยปัญญาประดิษฐ์ สามารถแบ่งการทำงานหลักๆ ได้เป็น 2 ส่วน คือ ขั้นตอนการเตรียมข้อมูล และส่วนของการฝึกอบรมข้อมูล (training) ข้อมูลเพื่อสร้างโมเดล (Model) ที่ใช้ในการคาดการณ์ ซึ่งการทำงานทั้ง 2 ส่วนนี้จะใช้ภาษา Python ในการเขียนโปรแกรม โดยมีบล็อกไดอะแกรมของระบบแสดงดังรูปที่ 3.1



รูปที่ 3.1 บล็อกไดอะแกรมของระบบสนับสนุนการตัดสินใจในการซื้อขายหุ้นด้วยปัญญาประดิษฐ์

3.1.2 การออกแบบในส่วนของการเตรียมข้อมูล

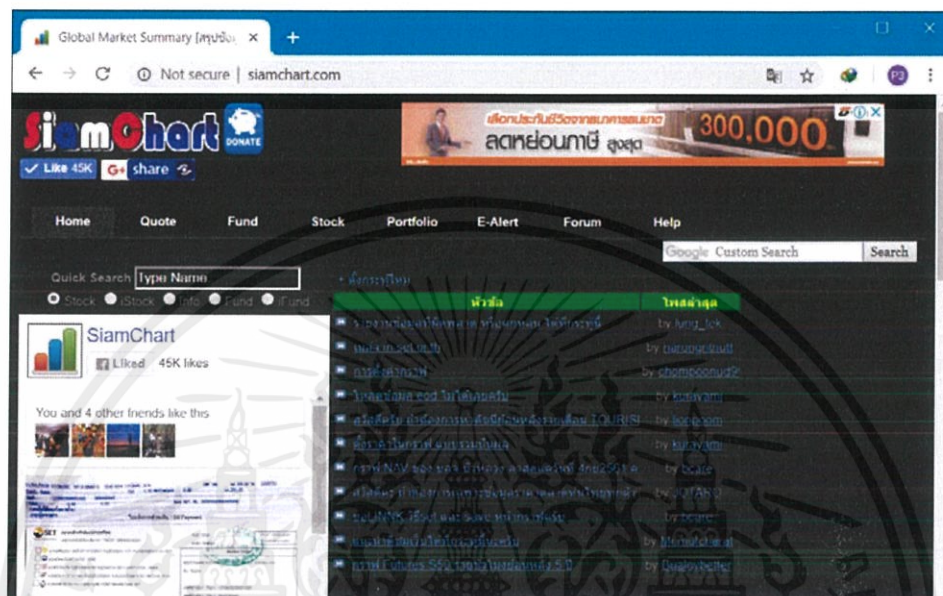
ขั้นตอนก่อนกระบวนการ (Pre-Processing) จะเป็นขั้นตอนในการจัดการกับข้อมูลเพื่อเตรียมสำหรับการฝึกอบรม (Training) มีขั้นตอนย่อย คือ ดาวน์โหลดข้อมูล จัดรูปแบบข้อมูล และการใช้เทคนิคเพิ่มเติมซึ่งเป็นเทคนิคที่ใช้ในการวิเคราะห์ราคาหุ้นมาไว้เพื่อช่วยในการคาดการณ์ราคาหุ้น โดยใช้ภาษา Python แสดงดังแสดงในรูปที่ 3.2



รูปที่ 3.2 ภาพรวมของขั้นตอนการเตรียมข้อมูล

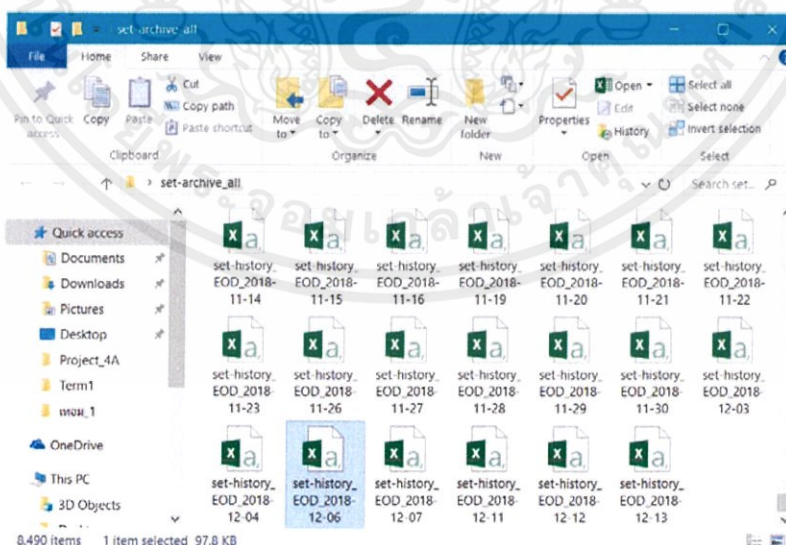
3.1.2.1 โหลดข้อมูลด้วยตนเอง

รวบรวมข้อมูลราคาหุ้นต่างๆ ย้อนหลัง ซึ่งจะได้โดยดาวน์โหลดข้อมูลจาก www.siamchart.com



รูปที่ 3.3 เว็บไซต์ SiamChart (www.siamchart.com) [12]

เมื่อเข้าไปยังเว็บไซต์ SiamChart ให้ไปที่ Stock List เพื่อเข้าสู่หน้าต่างดาวน์โหลดไฟล์ข้อมูล จากนั้นให้เลือกไฟล์ข้อมูลที่จะทำการดาวน์โหลด



รูปที่ 3.4 ไฟล์ข้อมูลที่ดาวน์โหลดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.2 ใช้ภาษา Python ในการจัดเรียงข้อมูล

เมื่อทำการโหลดข้อมูลเสร็จเรียบร้อยแล้ว จะทำการเลือกข้อมูลเฉพาะ ข้อมูลราคาหุ้นที่ต้องการ โดยใช้ฟังก์ชันที่เขียนขึ้นโดยภาษา Python จากนั้นทำการบันทึกเป็นไฟล์ .csv แยกออกมา

	A	B	C	D	E	F	G	H
13	TH100	20181206	4.82	4.82	4.82	4.82	3000	
14	2S	20181206	2.74	2.76	2.72	2.76	56000	
15	7UP	20181206	0.51	0.51	0.49	0.5	4578300	
16	A	20181206	6.65	6.65	6.65	6.65	20000	
17	AAV	20181206	4.26	4.3	4.24	4.28	4152100	
18	ABICO	20181206	4.6	4.62	4.6	4.62	24500	
19	ABM	20181206	0.97	1.04	0.97	0.99	480300	
20	ABPIF	20181206	7.9	7.9	7.7	7.8	610700	
21	ACAP	20181206	3.98	3.98	3.94	3.94	380100	
22	ACC	20181206	0.51	0.51	0.5	0.5	98500	
23	ADB	20181206	0.98	0.98	0.95	0.96	398000	

รูปที่ 3.5 ข้อมูลในแต่ละไฟล์ที่ดาวน์โหลดได้

	A	B	C	D	E	F
1	Date	Open	High	Low	Close	
2	6/12/01	38	38.25	35.5	35.75	
3	7/12/01	36	36.25	35.5	35.5	
4	11/12/01	35.5	35.5	34	34	
5	12/12/01	34.25	35.25	34	35	
6	13/12/01	35	35.25	35	35	
7	14/12/01	35	35.25	34.5	34.5	
8	17/12/01	34.5	34.75	34.5	34.5	
9	18/12/01	34.5	34.75	34.5	34.5	

รูปที่ 3.6 ข้อมูลไฟล์ .csv ที่ทำการจัดเรียงแล้ว

เมื่อทำการบันทึกไฟล์ที่มีเพียงเฉพาะข้อมูลราคาหุ้นตัวที่ต้องการแล้ว ผู้จัดทำได้ทำการเลือกค่าราคาปิดเป็นข้อมูลที่จะใช้ทำการคาดการณ์ผล ซึ่งต้องมีการจัดรูปแบบก่อนที่จะนำเข้ากระบวนการฝึกอบรม

ทำการแบ่งชุดข้อมูลออกเป็น 2 ส่วน ได้แก่ ชุด train และ test จากนั้นทำการจัดกลุ่มข้อมูลภายในชุด train และ test โดยการเขียนโปรแกรม ดังนี้

- `def new_dataset(dataset, step_size):` เป็นคำสั่งการเขียนฟังก์ชัน เพื่อแบ่งชุดข้อมูลซึ่งแบ่งออกเป็น `data_x` และ `data_y` โดยจะแบ่งจากค่า `step_size`
- `np.reshape` เป็นคำสั่งจัดเรียงรูปร่างของข้อมูลในอาร์เรย์ ให้มีค่าตามตัวแปรที่กำหนด
- `MinMaxScaler()` จะเป็นการคำนวณหรือปรับลดข้อมูลซึ่งเป็นการ `normalize` ข้อมูล ให้มีค่าอยู่ในช่วง 0 และ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- `fit_transform()` เปรียบเสมือนคำสั่งย่อยของคำสั่ง `MinMaxScaler()` คือการนำเข้าข้อมูลเพื่อใช้ในกระบวนการการปรับลดข้อมูล โดยใช้คำสั่ง `fit` และ `transform` จะเป็นคำสั่งในการแปลงค่าข้อมูลให้เป็นค่าที่ปรับแล้ว

- `len()` เป็นคำสั่งหาค่าความยาวหรือจำนวนของข้อมูล

```
In [3]: 1 def new_dataset(dataset, step_size):
2         data_X, data_Y = [], []
3         for i in range(len(dataset)-step_size-1):
4             a = dataset[i:(i+step_size), 0]
5             data_X.append(a)
6             data_Y.append(dataset[i + step_size, 0])
7         return np.array(data_X), np.array(data_Y)
8
9         close_val = dataset[['Close']]
10
11         Close_val_avg = np.reshape(close_val.values, (len(close_val),1))
12         scaler = MinMaxScaler(feature_range=(0, 1))
13         Close_avg = scaler.fit_transform(Close_val_avg)
14
15         # Train and Test splitting
16         train_Close = int(len(Close_avg) * 0.75)
17         test_Close = len(Close_avg) - train_Close
18         train_Close, test_Close = Close_avg[0:train_Close,:], Close_avg[train_Close:len(Close_avg),:]
19
20         # Time-series dataset (for at time = t, values for time = t + 1)
21         trainX, trainY = new_dataset(train_Close, 1)
22         testX, testY = new_dataset(test_Close, 1)
23
24         # Reshaping train and test data
25         trainX = np.reshape(trainX, (trainX.shape[0], 1, trainX.shape[1]))
26         testX = np.reshape(testX, (testX.shape[0], 1, testX.shape[1]))
27
```

รูปที่ 3.7 คำสั่งการจัดกลุ่มข้อมูลในชุด train และ test

3.1.3 การเขียนโปรแกรมในการสร้าง Model

ขั้นตอนนี้จะเป็นการกำหนดค่าตัวแปรต่างๆ เพื่อใช้ในสร้าง Model การคาดการณ์ โดยใช้คำสั่งต่างๆ ดังนี้

- `model = Sequential()` เป็นคำสั่งที่ให้ model เป็นข้อมูลแบบตามลำดับ

- `model.add(LSTM(32, input_shape = (1, step_size), return_sequences = True))` เป็นคำสั่งระบุรายละเอียดของ Model โดยคำสั่ง `add()` ได้แก่ ให้ประมวลผลเป็นโครงข่าย LSTM 32 hidden layer ซึ่งมี input มีลักษณะเป็น 3 มิติ

- `model.add(Dense(1, activation=Activate_fnc))` เป็นคำสั่งระบุรายละเอียดของ Model เพิ่มเติม โดยคำสั่ง `add()` ได้แก่ 1 โครงข่ายประสาทเทียม activation เป็นการกำหนดให้ชนิดของ activation หรือฟังก์ชันที่ช่วยในการคำนวณ

- `model.compile(loss='mean_square_error' optimizer=Optimizer)` เป็นคำสั่งที่ระบุวิธีที่ใช้ในการคำนวณค่าความผิดพลาด และประสิทธิภาพในการประมวลผล

- `model.fit(trainX, trainY, epochs=Epochs, batch_size= Batch_size, verbose=2)` เป็นคำสั่งนำเข้าข้อมูล input ได้แก่ trainX และ trainY epochs เป็นการกำหนดรอบการประมวลผลเท่ากับ 100 รอบ batch_size จำนวนข้อมูลที่ย้อนกลับ (output ของการประมวลผลลำดับก่อนหน้า คือ input ของการประมวลผลถัดไป) verbose เป็นการกำหนดรูปแบบการแสดงความคืบหน้าของการประมวลผล

```

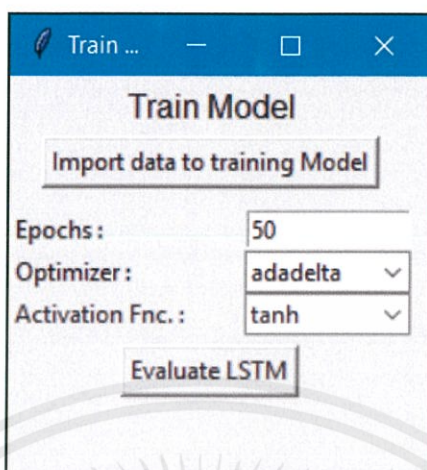
1 #Parameter
2 Epochs = 100
3 Batch_size = 1
4 Activate_fnc = 'tanh' # 'sigmoid', 'softmax', 'tanh', 'hard_sigmoid', 'exponential', 'linear'
5 step_size = 1
6
7 Optimizer = 'adadelta' # 'adagrad', 'adam'
8
9
10 model = Sequential()
11
12 #Build LSTM model
13 model.add(LSTM(32, input_shape = (1, step_size), return_sequences = True))
14 model.add(Dense(1, activation=Activate_fnc))
15
16 model.compile(loss = 'mean_squared_error', optimizer = Optimizer)
17 model.fit(trainX, trainY, epochs = Epochs, batch_size = Batch_size, verbose = 2)

```

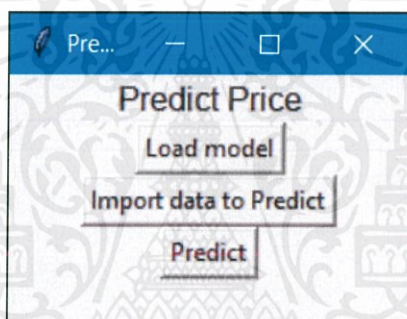
รูปที่ 3.8 การเขียนโปรแกรมกำหนดค่าพารามิเตอร์และการสร้าง Model การคาดการณ์ผล

3.1.4 การออกแบบ GUI ในส่วนผู้ใช้งาน

การออกแบบ GUI (Graphical User Interface) จะเป็นส่วนของการแสดงผลซึ่งจะเป็นส่วนโต้ตอบกับผู้ใช้งาน ซึ่งจะออกแบบให้ผู้ใช้งานสามารถ ปรับค่าพารามิเตอร์ในการเทรน Model ได้ เช่น จำนวนรอบของการประมวลผล (Epochs) ชนิดของฟังก์ชันช่วย (Activation Function) ปุ่มสำหรับนำเข้าข้อมูล ปุ่มสำหรับนำเข้าตัวโมเดลที่เทรนเรียบร้อยแล้ว และปุ่มในการประมวลผล



รูปที่ 3.9 หน้าต่าง GUI สำหรับผู้ใช้งานในส่วนของการ Train Model



รูปที่ 3.10 หน้าต่าง GUI สำหรับผู้ใช้งานในส่วนของการใช้ Model คาดการณ์ผลราคาหุ้น

3.2 เครื่องมือที่ใช้ในการทดลอง

ในปฏิญญาพันธนี้ มีอุปกรณ์และเครื่องมือที่ใช้ในการทดลอง ดังนี้

3.2.1 คอมพิวเตอร์

คอมพิวเตอร์หรือโน้ตบุ๊กที่ติดตั้งหน่วยประมวลผลด้านกราฟฟิก (Graphics Processing Unit: GPU) TensorFlow สามารถทำงานได้ทั้งบน CPU และ GPU แต่การที่ติดตั้ง TensorFlow ให้ประมวลผลบน GPU นั้นช่วยให้ประมวลผลได้เร็วมากกว่า

3.2.2 โปรแกรม Anaconda

โปรแกรม Anaconda เป็นโปรแกรมที่ทำให้การติดตั้ง Library โมดูลหรือเครื่องมือที่ใช้ในการเขียนโปรแกรมสะดวกและง่ายขึ้น

3.2.3 โปรแกรม Jupyter Notebook

Jupyter Notebook เป็นโปรแกรมที่ใช้เขียนภาษา Python ที่ถูกติดตั้งพร้อมกับโปรแกรม Anaconda ที่สามารถทำ Markdown ที่ใช้เขียนคำอธิบายโค้ดหรือคำบรรยายต่างๆ ได้ โดยที่ไม่ส่งข้อความไปให้ Python ประมวลผล ในปฏิญานพนธ์นี้ใช้ Jupyter Notebook ในการจัดเรียงข้อมูล เนื่องจากสามารถบอกรายละเอียดของข้อมูลได้ดีกว่า

3.2.4 โปรแกรม Visual Studio Code

Visual Studio Code หรือ VSCode เป็นโปรแกรมใช้เขียนภาษา Python ที่ถูกติดตั้งพร้อมกับโปรแกรม Anaconda เช่นเดียวกับกับ Jupyter Notebook ซึ่ง VSCode เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขและปรับแต่งโปรแกรมต่างๆ ได้ในปฏิญานพนธ์นี้จะใช้ VSCode ในส่วนแสดงผลของ GUI ซึ่งทำได้ดีกว่า Jupyter Notebook

3.3 การจัดเก็บผลการทดลอง

การจัดเก็บผลการทดลองแบ่งออกเป็น 3 ส่วน

3.3.1 การดาวน์โหลดข้อมูลด้วยตนเอง

จัดเก็บเป็นไฟล์ .csv ที่มีข้อมูลราคาหุ้นทั้งตลาดหลักทรัพย์

3.3.2 การใช้ภาษา Python เพื่อจัดเรียงข้อมูล

จัดเก็บเป็นไฟล์ .csv ที่มีข้อมูลราคาหุ้นเฉพาะตัวที่เลือก และเป็นไฟล์ที่จัดเรียงข้อมูลเรียบร้อยแล้ว

3.3.3 การปรับเปลี่ยนพารามิเตอร์เพื่อสร้าง Model

จัดทำตารางเก็บผลค่า RMSE เมื่อปรับเปลี่ยนพารามิเตอร์ต่างๆ ซึ่ง RMSE เป็นค่าที่สามารถบอกรายละเอียดความแม่นยำของ Model

3.3.4 การทดลองปรับเปลี่ยนอัตราส่วนการแบ่งชุดข้อมูล

จัดทำตารางเก็บผลค่า RMSE เมื่อปรับอัตราส่วนการแบ่งชุดข้อมูล

3.3.5 การทดลองการคาดการณ์ผลราคาหุ้นหลายวัน

จัดทำตารางเก็บผลการคาดการณ์ราคาหุ้นหลายวัน

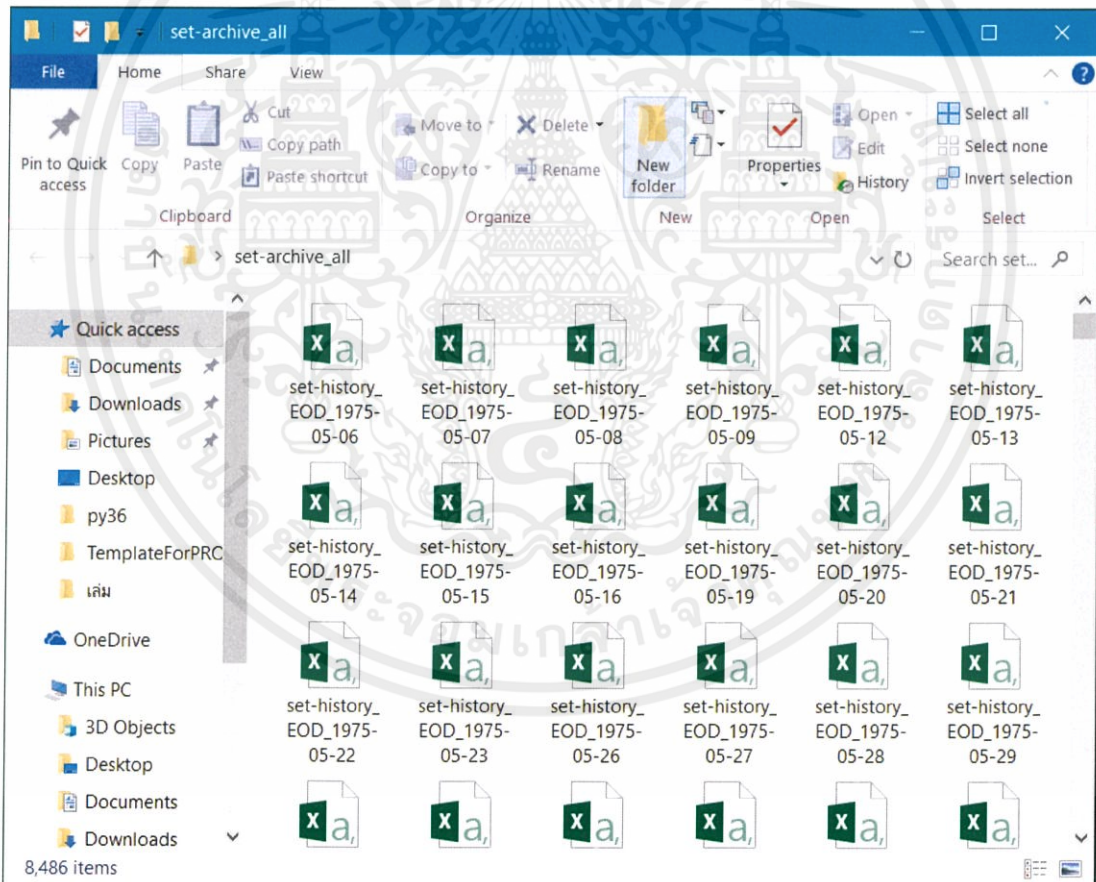
บทที่ 4

ผลการทดลอง

การจัดเก็บผลการทำงานของระบบ สามารถแบ่งการทดลองและจัดเก็บผลการทดสอบออกเป็น 3 ส่วนดังนี้

4.1 ผลการดาวน์โหลดไฟล์ข้อมูลราคาหุ้นด้วยตนเอง

ขั้นตอนนี้จะเป็นการดาวน์โหลดข้อมูลด้วยตนเอง โดยทำการดาวน์โหลดจากเว็บไซต์ SiamChart (www.siamchart.com) ซึ่งไฟล์ข้อมูลที่ได้จะเป็นไฟล์ .csv แต่ละไฟล์คือข้อมูลตลาดหลักทรัพย์ทั้งหมดในวันนั้นๆ ภายในไฟล์จะประกอบด้วยรายชื่อหุ้นทั้งหมดในตลาดหลักทรัพย์และข้อมูลราคาของหุ้นนั้นๆ เช่น วันที่ ราคาปิด ราคาเปิด ราคาสูงสุด ราคาต่ำสุด และระดับราคา



รูปที่ 4.1 ไฟล์ข้อมูลที่ทำกรดาวน์โหลดด้วยตนเองจากเว็บไซต์ SiamChart

	A	B	C	D	E	F	G	H	I	J	K
1	<TICKER>	<DTYYYY>	<OPEN>	<HIGH>	<LOW>	<CLOSE>	<VOL>				
2	1DIV	20180103	12.5	12.55	12.4	12.55	6400				
3	BMSCITH	20180103	13.2	13.39	13.2	13.39	564000				
4	BSET100	20180103	10.99	11.15	10.98	11.15	590900				
5	CHINA	20180103	6.51	6.56	6.48	6.49	94800				
6	ECOMM	20180103	9.11	9.11	9.11	9.11	2000				
7	EICT	20180103	4.98	5.82	4.98	5.13	3800				
8	ENY	20180103	7.22	7.22	7.22	7.22	1000				
9	GLD	20180103	1.81	1.82	1.81	1.82	197700				
10	TDEX	20180103	11.25	11.44	11.25	11.42	562700				
11	TGOLDETF	20180103	3.99	3.99	3.98	3.98	566500				
12	2S	20180103	5	5	4.94	4.94	258100				
13	A	20180103	6.55	6.6	6.55	6.6	100500				
14	AAV	20180103	6.1	6.15	6	6	22567500				
15	ABICO	20180103	9.45	9.55	9.4	9.45	137300				
16	ABPIF	20180103	8.55	8.6	8.55	8.6	114600				
17	ACAP	20180103	14.4	14.4	13.7	13.7	1788600				
18	ACC	20180103	0.52	0.52	0.51	0.52	356200				
19	ADB	20180103	1.33	1.34	1.3	1.31	2239300				
20	ADVANC	20180103	191.5	197.5	191	196.5	15405400				
21	AEC	20180103	0.59	0.6	0.58	0.58	805000				
22	AEONTS	20180103	104	104	103	103.5	76800				
23	AF	20180103	0.92	0.93	0.88	0.91	3283300				
24	AFC	20180103	9	9	9	9	1000				
25	AGE	20180103	1.9	1.91	1.88	1.88	1996400				
26	AH	20180103	34.75	34.75	34	34.5	354000				
27	AHC	20180103	22.3	22.6	22.2	22.6	3200				
28	AIRA	20180103	2.28	2.28	2.26	2.26	76300				
29	AIT	20180103	30.75	32	30.5	32	3520700				
30	AJ	20180103	12	12.2	12	12.1	55000				
31	AJA	20180103	0.55	0.55	0.53	0.53	2732000				

รูปที่ 4.2 ข้อมูลภายในของแต่ละไฟล์ที่ดาวน์โหลดจากเว็บไซต์ SiamChart

4.2 ผลการใช้ภาษา Python ในการจัดเรียงข้อมูล

เขียนโปรแกรมด้วยภาษา Python โดยการใช้โปรแกรม Jupyter Notebook เริ่มจากการดึงข้อมูลหุ้นตัวที่ต้องการทำการฝึกอบรม (Training) แยกออกมาจากข้อมูลไฟล์ทั้งหมด

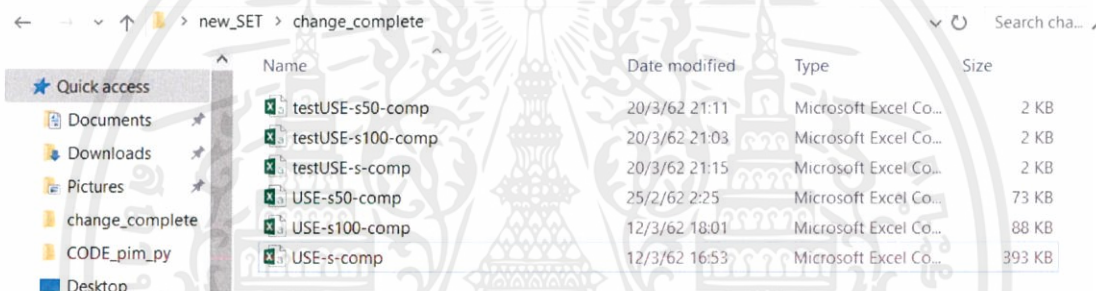
ข้อมูลที่ได้ในขั้นตอนนี้จะเป็นข้อมูลตารางที่มีเฉพาะข้อมูลราคาหุ้นตัวดังกล่าว และมีการเรียงลำดับข้อมูลตามวัน

Out[3]:

	Open	High	Low	Close
2001-12-06	38.00	38.25	35.5	35.75
2001-12-07	36.00	36.25	35.5	35.50
2001-12-11	35.50	35.50	34.0	34.00
2001-12-12	34.25	35.25	34.0	35.00
2001-12-13	35.00	35.25	35.0	35.00

รูปที่ 4.3 แยกข้อมูลที่มีเฉพาะหุ้นที่ต้องการ

จากนั้นทำการบันทึกข้อมูลดังกล่าว ให้เป็นไฟล์ .csv ไฟล์ใหม่ที่ประกอบด้วยข้อมูลราคาหุ้นเฉพาะตัวที่เลือก ภายในไฟล์ที่บันทึกผลจะประกอบด้วยข้อมูลราคาของหุ้นที่เลือก ได้แก่ ราคาเปิด ราคาสูงสุด ราคาต่ำสุด และราคาปิด



รูปที่ 4.4 ไฟล์ที่ทำการบันทึกจากการเขียนโปรแกรม

	Date	Close	Open	High	Low
2	16/8/11	748.83	759.69	759.93	748.83
3	17/8/11	760.83	755.34	762.55	754.04
4	18/8/11	758.83	765.44	768.51	757.87
5	19/8/11	742.86	746.31	751.14	740.64
6	22/8/11	741.98	742.08	742.53	729.07
7	23/8/11	733.83	741.96	742.98	732.56

รูปที่ 4.5 ข้อมูลภายในไฟล์ที่ทำการบันทึก

4.3 ผลการปรับเปลี่ยนพารามิเตอร์เพื่อสร้าง Model

จัดทำตารางเก็บผล RMSE เมื่อลองปรับเปลี่ยนค่าพารามิเตอร์ต่างๆ ภายในโมเดล ซึ่งค่า RMSE สามารถบอกความแม่นยำของโมเดลได้

เมื่อข้อมูลหุ้นที่ใช้ คือ SET50 ช่วงเวลาที่นำมาใช้คือข้อมูลราคาหุ้นย้อนหลังตั้งแต่วันที่ 16/08/2011 ถึงวันที่ 28/12/2018 ซึ่งเป็นการใช้ Epoch = 50 หรือกำหนดการประมวลผลภายในหน่วยโครงข่ายประสาทเทียม เท่ากับ 50 รอบ มีผลการทดลองดังต่อไปนี้

ตารางที่ 4.1 ความสัมพันธ์ของค่า RMSE กับการเปลี่ยนค่าพารามิเตอร์ต่างๆ ในการสร้าง Model การคาดการณ์ผล เมื่อกำหนดให้รอบของการประมวลผลมีค่าเท่ากับ 50

Model	Epoch	Optimizer	Activation Function	RSME		Price Predict	
				train	test	Last Day	Next Day
1	50	adagrad	tanh	10.343	29.462	1036.200	921.843
2	50	adagrad	sigmoid	12.117	42.634	1034.465	872.294
3	50	adagrad	hard_sigmoid	10.250	24.250	1037.493	941.005
4	50	adagrad	exponential	13.630	10.930	1046.540	997.574
5	50	adagrad	linear	10.250	8.710	1042.382	1053.320
6	50	adagrad	relu	10.219	8.683	1042.331	1053.248
7	50	adam	tanh	10.768	24.901	1040.837	937.368
8	50	adam	sigmoid	10.727	16.455	1046.013	966.844
9	50	adam	hard_sigmoid	10.426	11.999	1042.514	999.076
10	50	adam	exponential	11.260	26.340	1045.858	925.465
11	50	adam	linear	10.315	11.551	1041.101	1002.821
12	50	adam	relu	10.194	16.230	1035.918	982.555
13	50	adadelata	tanh	10.834	19.949	1037.040	920.080
14	50	adadelata	sigmoid	12.290	42.359	1034.234	875.221
15	50	adadelata	hard_sigmoid	10.607	28.098	1038.064	945.659
16	50	adadelata	exponential	10.599	17.939	1038.626	973.466
17	50	adadelata	linear	10.260	11.590	1039.651	1003.847
18	50	adadelata	relu	10.256	11.674	1039.696	1003.286

เมื่อข้อมูลหุ่นที่ใช้ คือ SET50 ช่วงเวลาที่นำมาใช้คือข้อมูลราคาหุ้นย้อนหลังตั้งแต่วันที่ 16/08/2011 ถึงวันที่ 28/12/2018 ซึ่งเป็นการใช้ Epoch = 100 หรือกำหนดการประมวลผลภายในหน่วยโครงข่ายประสาทเทียม เท่ากับ 100 รอบ มีผลการทดลองดังต่อไปนี้

ตารางที่ 4.2 ความสัมพันธ์ของค่า RMSE กับการเปลี่ยนค่าพารามิเตอร์ต่างๆ ในการสร้าง Model การคาดการณ์ผล เมื่อกำหนดให้รอบของการประมวลผลมีค่าเท่ากับ 100

Model	Epoch	Optimizer	Activation Function	RSME		Price Predict	
				train	test	Last Day	Next Day
1	100	adagrad	tanh	10.273	27.761	1036.601	927.708
2	100	adagrad	sigmoid	12.077	42.437	1034.459	872.964
3	100	adagrad	hard_sigmoid	10.183	22.831	1037.906	948.480
4	100	adagrad	exponential	11.700	25.944	1040.549	929.473
5	100	adagrad	linear	10.007	7.776	1040.946	1045.171
6	100	adagrad	relu	9.955	7.919	1040.481	1031.602
7	100	adam	tanh	11.091	23.269	1036.256	941.320
8	100	adam	sigmoid	10.193	26.863	1039.122	924.706
9	100	adam	hard_sigmoid	10.433	13.055	1038.690	994.357
10	100	adam	exponential	10.235	23.883	1038.164	938.682
11	100	adam	linear	11.715	20.508	1034.040	959.563
12	100	adam	relu	11.874	18.132	1033.506	973.179
13	100	adadelat	tanh	10.426	29.577	1037.243	920.566
14	100	adadelat	sigmoid	11.941	41.320	1033.499	878.587
15	100	adadelat	hard_sigmoid	10.369	24.534	1037.611	943.017
16	100	adadelat	exponential	10.259	7.915	1042.541	1031.223
17	100	adadelat	linear	10.105	10.832	1041.365	1006.567
18	100	adadelat	relu	10.102	10.889	1041.365	1006.138

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อข้อมูลหุ่นที่ใช้ คือ SET50 ช่วงเวลาที่นำมาใช้คือข้อมูลราคาหุ้นย้อนหลังตั้งแต่วันที่ 16/08/2011 ถึงวันที่ 28/12/2018 ซึ่งเป็นการใช้ Epoch = 200 หรือกำหนดการประมวลผลภายในหน่วยโครงข่ายประสาทเทียม เท่ากับ 200 รอบ มีผลการทดลองดังต่อไปนี้

ตารางที่ 4.3 ความสัมพันธ์ของค่า RMSE กับการเปลี่ยนค่าพารามิเตอร์ต่างๆ ในการสร้าง Model การคาดการณ์ผล เมื่อกำหนดให้รอบของการประมวลผลมีค่าเท่ากับ 200

Model	Epoch	Optimizer	Activation Function	RSME		Price Predict	
				train	test	Last Day	Next Day
1	200	adagrad	tanh	10.286	27.739	1036.930	928.684
2	200	adagrad	sigmoid	11.784	40.531	1034.853	880.328
3	200	adagrad	hard_sigmoid	10.112	20.979	1038.072	956.487
4	200	adagrad	exponential	11.215	34.391	1037.174	899.246
5	200	adagrad	linear	9.933	9.033	1038.535	1019.766
6	200	adagrad	relu	9.923	9.275	1039.282	1018.337
7	200	adam	tanh	10.412	23.893	1041.793	938.210
8	200	adam	sigmoid	10.040	24.292	1040.323	932.265
9	200	adam	hard_sigmoid	10.030	13.004	1040.223	989.925
10	200	adam	exponential	10.292	31.278	1041.661	900.825
11	200	adam	linear	10.948	11.250	1044.052	1001.016
12	200	adam	relu	10.825	22.076	1043.615	948.439
13	200	adadelat	tanh	11.634	31.416	1037.447	914.691
14	200	adadelat	sigmoid	11.884	40.832	1034.764	879.301
15	200	adadelat	hard_sigmoid	10.944	25.513	1039.238	938.781
16	200	adadelat	exponential	10.501	8.129	1042.747	1027.459
17	200	adadelat	linear	10.543	14.514	1040.392	987.854
18	200	adadelat	relu	10.517	13.871	1040.468	990.992

4.4 ผลการทดลองปรับเปลี่ยนอัตราการส่วนการแบ่งชุดข้อมูล

จัดทำตารางเก็บผล RMSE เมื่อลองปรับเปลี่ยนอัตราส่วนการแบ่งชุดข้อมูล โดยมีอัตราส่วนดังนี้ 50:50 60:40 75:25 และ 85:15 ซึ่งจะนำ Model ที่ดีที่สุดจากการปรับค่าพารามิเตอร์ของแต่ละ Epoch มาใช้ในทำการทดลอง

เมื่อข้อมูลหุนที่ใช้ คือ SET50 ช่วงเวลาที่นำมาใช้คือข้อมูลราคาหุ้นย้อนหลังตั้งแต่วันที่ 16/08/2011 ถึงวันที่ 28/12/2018 กำหนดให้ Epoch มีค่าเท่ากับ 50 รอบ โดย Model ที่นำมาใช้ประกอบด้วยพารามิเตอร์ ดังนี้ Optimizer คือ AdaGrad และ Activation Function คือ ReLU มีผลการทดลองดังต่อไปนี้

ตารางที่ 4.4 ความสัมพันธ์ของค่า RMSE กับการปรับเปลี่ยนอัตราการแบ่งชุดข้อมูล เมื่อกำหนดให้รอบของการประมวลผลมีค่าเท่ากับ 50

Model	อัตราการส่วนการแบ่งชุดข้อมูล	Optimizer	Activation Function	RMSE		Price Predict	
				train	test	last day	next day
1	50:50	adagrad	relu	10.661	8.826	1042.320	1050.657
2	60:40	adagrad	relu	10.718	9.413	1043.950	1060.402
3	75:25	adagrad	relu	10.219	8.683	1042.331	1053.248
4	85:15	adagrad	relu	9.909	11.354	1043.076	1075.905

เมื่อข้อมูลหุนที่ใช้ คือ SET50 ช่วงเวลาที่นำมาใช้คือข้อมูลราคาหุ้นย้อนหลังตั้งแต่วันที่ 16/08/2011 ถึงวันที่ 28/12/2018 กำหนดให้ Epoch มีค่าเท่ากับ 100 รอบ โดย Model ที่นำมาใช้ประกอบด้วยพารามิเตอร์ ดังนี้ Optimizer คือ AdaGrad และ Activation Function คือ Linear มีผลการทดลองดังต่อไปนี้

ตารางที่ 4.5 ความสัมพันธ์ของค่า RMSE กับการปรับเปลี่ยนอัตราการแบ่งชุดข้อมูล เมื่อกำหนดให้รอบของการประมวลผลมีค่าเท่ากับ 100

Model	อัตราการส่วนการแบ่งชุดข้อมูล	Optimizer	Activation Function	RMSE		Price Predict	
				train	test	last day	next day
1	50:50	adagrad	linear	10.487	8.476	1041.692	1040.147
2	60:40	adagrad	linear	10.311	8.066	1041.538	1037.042
3	75:25	adagrad	linear	10.007	7.776	1040.946	1045.171
4	85:15	adagrad	linear	9.627	9.217	1042.570	1038.147

เมื่อข้อมูลหุ้นที่ใช้ คือ SET50 ช่วงเวลาที่นำมาใช้คือข้อมูลราคาหุ้นย้อนหลังตั้งแต่วันที่ 16/08/2011 ถึงวันที่ 28/12/2018 กำหนดให้ Epoch มีค่าเท่ากับ 200 รอบ โดย Model ที่นำมาใช้ประกอบด้วยพารามิเตอร์ ดังนี้ Optimizer คือ AdaDelta และ Activation Function คือ Exponential มีผลการทดลองดังต่อไปนี้

ตารางที่ 4.6 ความสัมพันธ์ของค่า RMSE กับการปรับเปลี่ยนอัตราการแบ่งชุดข้อมูล เมื่อกำหนดให้รอบของการประมวลผลมีค่าเท่ากับ 200

Model	อัตราการส่วนการแบ่งชุดข้อมูล	Optimizer	Activation Function	RMSE		Price Predict	
				train	test	last day	next day
1	50:50	adadelta	exponential	10.858	9.035	1043.223	1037.485
2	60:40	adadelta	exponential	10.266	11.914	1038.867	989.454
3	75:25	adadelta	exponential	10.501	8.129	1042.747	1027.459
4	85:15	adadelta	exponential	9.759	9.338	1043.612	1034.337

4.5 ผลการทดลองการคาดการณ์ผลราคาหุ้นหลายวัน

จัดทำตารางเก็บผลการคาดการณ์ราคาหุ้นหลายวัน โดยใช้ราคาปิด 5 วัน ในการคาดการณ์ราคาผล 1 วัน โดยใช้ราคาปิดจริง 5 วัน เป็นชุดข้อมูลเริ่มต้น และจะนำผลการคาดการณ์ที่ได้ใช้เป็นข้อมูลอินพุตร่วมในการคาดการณ์ผลในงวดถัดไป

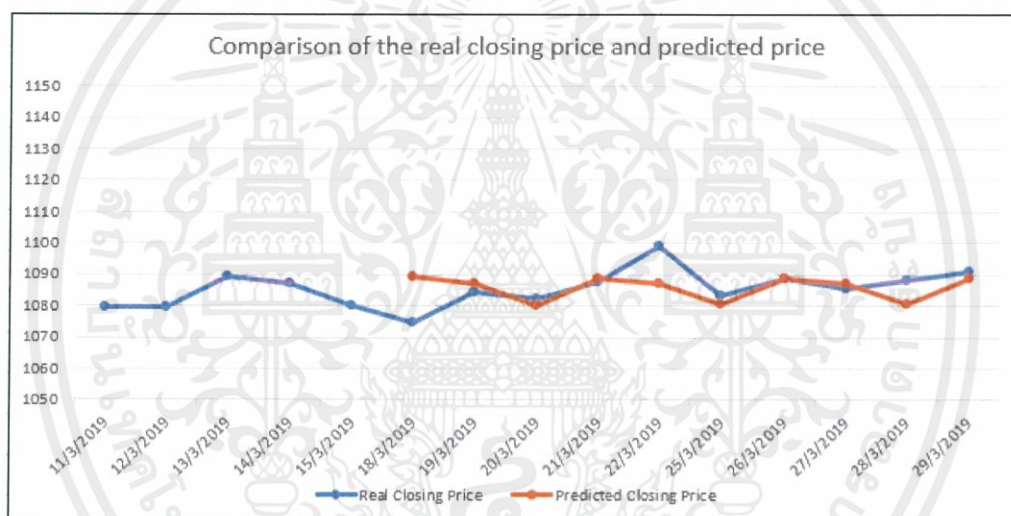
ตารางที่ 4.7 ตารางผลการทดลองการคาดการณ์ผลราคาหุ้นหลายวัน

Date	Real Price	Predicted Price	NOTE
11/3/19	1079.36	-	-
12/3/19	1079.46	-	-
13/3/19	1089.1	-	-
14/3/19	1087.12	-	-
15/3/19	1080.01	-	-
18/3/19	1074.63	1089.0408	ใช้ราคาปิดจริงของวันที่ 11-15 มี.ค.
19/3/19	1084.3	1087.1205	ใช้ราคาปิดจริงของวันที่ 12-15 มี.ค. และผลที่คาดการณ์ได้ของวันที่ 18
20/3/19	1081.97	1080.2155	ใช้ราคาปิดจริงของวันที่ 13-15 มี.ค. และผลที่คาดการณ์ได้ของวันที่ 18-19
21/3/19	1087.41	1088.9858	ใช้ราคาปิดจริงของวันที่ 14-15 มี.ค. และผลที่คาดการณ์ได้ของวันที่ 18-20

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.7 ตารางผลการทดลองการคาดการณ์ผลราคาหุ้นหลายวัน (ต่อ)

Date	Real Price	Predicted Price	NOTE
22/3/19	1098.85	1087.1216	ใช้ราคาปิดจริงของวันที่ 15 มี.ค. และผลที่คาดการณ์ได้ของวันที่ 18-21
25/3/19	1083.05	1080.415	ใช้ราคาปิดจริงของวันที่ 18-22 มี.ค.
26/3/19	1088.47	1088.9325	ใช้ราคาปิดจริงของวันที่ 19-22 มี.ค. และผลที่คาดการณ์ได้ของวันที่ 25
27/3/19	1085.46	1087.1226	ใช้ราคาปิดจริงของวันที่ 20-22 มี.ค. และผลที่คาดการณ์ได้ของวันที่ 25-26
28/3/19	1088.36	1080.6088	ใช้ราคาปิดจริงของวันที่ 21-22 มี.ค. และผลที่คาดการณ์ได้วันที่ 25-27
29/3/19	1090.94	1088.8807	ใช้ราคาปิดจริงของวันที่ 22 มี.ค. และผลที่คาดการณ์ได้ของวันที่ 25-28



รูปที่ 4.6 กราฟแสดงการเปรียบเทียบของราคาปิดจริงกับราคาปิดที่คาดการณ์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ปริญญาานิพนธ์นี้มีวัตถุประสงค์เพื่อใช้เป็นเครื่องมือช่วยสนับสนุนในการตัดสินใจในการซื้อขายหุ้น ได้ทำการศึกษาการออกแบบในส่วนของการจัดเตรียมข้อมูล และการฝึกอบรมข้อมูล เพื่อสร้างโมเดลหรือรูปแบบการคาดการณ์ผลราคาหุ้น ซึ่งในปริญญาานิพนธ์นี้ใช้ TensorFlow ร่วมกับ Keras ที่ใช้เทคนิค Machine Learning Deep Learning และโครงข่ายประสาทเทียมชนิด Long Short-Term Memory Networks (LSTMs) ซึ่งเป็นการประมวลผลรูปแบบหนึ่งในการสร้างโมเดลการคาดการณ์ โดยเขียนโปรแกรมด้วยภาษา Python ผ่านโปรแกรม Jupyter Notebook และ Visual Studio Code จากนั้นทำการออกแบบในส่วนของการแสดงผลที่เป็นส่วนโต้ตอบกับผู้ใช้งานหรือ GUI (Graphical User Interface) โดยมีฟังก์ชันการใช้งาน คือ ผู้ใช้งานสามารถเลือกใช้ข้อมูลที่จะนำมาประมวลผล จำนวนของ Epoch ชนิดของ Activation Function และสามารถเรียกใช้งาน Model ที่ฝึกอบรมเรียบร้อยแล้วหรือ สามารถใส่ค่าพารามิเตอร์ต่างๆ เพื่อฝึกอบรมข้อมูลเพื่อสร้าง Model ในการคาดการณ์ผลด้วยตนเองได้

จากการทดลองปรับเปลี่ยนค่าพารามิเตอร์ต่างๆ ภายใน Model ได้แก่ จำนวนรอบในการประมวลผลภายในโครงข่ายประสาทเทียม (Epoch) ฟังก์ชันที่ใช้ช่วยในการคำนวณ (Activation Function) เครื่องมือที่ช่วยปรับปรุงประสิทธิภาพ (Optimizer) โดยใช้ข้อมูลหุ้นคือ SET50 ในการคาดการณ์ผลในวันถัดไปของการทดลอง หรือคาดการณ์ผลราคาหุ้นในวันที่ 2/1/2019 ซึ่งมีราคาปิดจริง คือ 1,046.71 บาท

เมื่อ Epoch เท่ากับ 50 Model ที่สามารถคาดการณ์ราคาหุ้นได้ใกล้เคียงที่สุดคือ Model ที่ใช้ Optimizer คือ AdaGrad (Adaptive Gradient) และใช้ Activation Function คือ ReLU Function (Rectified Linear Unit) ซึ่งคาดการณ์ราคาได้ 1,053.248 บาท

เมื่อ Epoch เท่ากับ 100 Model ที่สามารถคาดการณ์ราคาหุ้นได้ใกล้เคียงที่สุดคือ Model ที่ใช้ Optimizer คือ AdaGrad (Adaptive Gradient) และใช้ Activation Function คือ Linear Function ซึ่งคาดการณ์ราคาได้ 1,045.171 บาท

เมื่อ Epoch เท่ากับ 200 Model ที่สามารถคาดการณ์ราคาหุ้นได้ใกล้เคียงที่สุดคือ Model ที่ใช้ Optimizer คือ AdaDelta และใช้ Activation Function คือ Exponential Function ซึ่งคาดการณ์ราคาได้ 1,027.459 บาท

จากการทดลองการปรับเปลี่ยนอัตราส่วนการแบ่งข้อมูล โดยมีอัตราส่วนเท่ากับ 50:50 60:40 75:25 และ 85:15 โดยใช้ Model ที่ดีที่สุดจากการทดลองการปรับเปลี่ยนพารามิเตอร์ของแต่ละ Epoch มีผลกาทดลองดังนี้

เมื่อใช้ Model ที่ให้ค่า RMSE ที่ดีที่สุดของการปรับเปลี่ยนค่าพารามิเตอร์ของแต่ละ Epoch ให้ผลการทดลองที่ตรงกัน คือ อัตราการส่วนการแบ่งชุดข้อมูลที่ทำให้ Model มีการคาดการณ์แม่นยำที่สุด คือ อัตราส่วน 75:25

5.2 ข้อเสนอแนะ

ระบบสนับสนุนการตัดสินใจในการซื้อขายหุ้นด้วยปัญญาประดิษฐ์นั้นจัดทำขึ้นเพื่อเป็นเครื่องมือที่ช่วยสนับสนุนในการตัดสินใจในการซื้อขายหุ้น ซึ่งในการคาดการณ์ผลราคาหุ้นนั้นต้องทำการสร้างโมเดลหรือรูปแบบการคาดการณ์ ซึ่งต้องอาศัยการปรับค่าพารามิเตอร์ต่างๆ เพื่อให้ได้ผลการคาดการณ์ที่มีความแม่นยำ และข้อมูลหุ้นที่จะนำมาใช้ในการคาดการณ์ผลต้องเป็นหุ้นตัวเดียวกันกับที่นำมาเข้ากระบวนการเพื่อสร้าง Model เนื่องจากหุ้นแต่ละตัวนั้นมีพฤติกรรมเปลี่ยนแปลงราคาที่แตกต่างกัน

5.3 ปัญหาและอุปสรรค

เนื่องจากการเปลี่ยนแปลงของราคาหุ้นนั้นเกิดจากปัจจัยและตัวแปรที่หลากหลาย เช่น สภาวะเศรษฐกิจของแต่ละอุตสาหกรรม สภาวะทางการเงิน ผลประกอบการของบริษัท รวมไปถึงเหตุการณ์ต่างๆ ที่เกิดขึ้นในประเทศไทยและต่างประเทศ ดังนั้นการคาดการณ์ราคาหุ้นอาจจะมี การคาดเคลื่อนซึ่งอาจนำมาซึ่งความเสียหายที่ส่งผลกระทบต่อเป็นอย่างมากได้

บรรณานุกรม

- [1] Forex official. “แนวโน้มคืออะไร? ทำความรู้จัก Uptrend, Downtrend และ Sideway.” <https://www.forextoforex.com/%E0%B9%81%E0%B8%99%E0%B8%A7%E0%B9%82%E0%B8%99%E0%B9%89%E0%B8%A1%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/>.
- [2] Athiwat. “Artificial Intelligence, Machine Learning, และ Deep Learning แตกต่างกันอย่างไรร?” <https://medium.com/athivvat/artificial-intelligence-machine-learning-%E0%B9%81%E0%B8%A5%E0%B8%B0-deep-learning-%E0%B9%81%E0%B8%95%E0%B8%81%E0%B8%95%E0%B9%88%E0%B8%B2%E0%B8%87%E0%B8%81%E0%B8%B1%E0%B8%99%E0%B8%AD%E0%B8%A2%E0%B9%88%E0%B8%B2%E0%B8%87%E0%B9%84%E0%B8%A3-68a8aa6f56f0>.
- [3] wanisa866. “AI In Library.” <https://wanisa866.wordpress.com/2018/06/15/ai-in-library/>.
- [4] Bonaccorso, Giuseppe. *Machine Learning Algorithms*. Packt Publishing Ltd. 2017.
- [5] “Difference Between Artificial Intelligence-MachineLearning and DeepLearning?.” <https://socialprachar.com/difference-between-artificial-intelligence-machine-learning-and-deep-learning/>.
- [6] dullboy. "How Machine Learning is changing Software Development." <https://steemit.com/datascience/@dullboy/how-machine-learning-is-changing-software-development>.
- [7] Doratong24. “[ML] LSTMs แบบ (เกือบ) ละเอียดยิบ ๆ.” <https://medium.com/@tongkornkitt/ml-lstms-%E0%B9%81%E0%B8%9A%E0%B8%9A-%E0%B9%80%E0%B8%81%E0%B8%B7%E0%B8%AD%E0%B8%9A-%E0%B8%A5%E0%B8%B0%E0%B9%80%E0%B8%AD%E0%B8%B5%E0%B8%A2%E0%B8%94%E0%B8%A2%E0%B8%B4%E0%B8%9A-%E0%B8%A3-68a8aa6f56f0>.

0%B9%86-a3a55cd37883.

- [8] vibhor nigam. “Understanding Neural Networks. From neuron to RNN, CNN, and Deep Learning.” <https://towardsdatascience.com/understanding-neural-networks-from-neuron-to-rnn-cnn-and-deep-learning-cd88e90e0a90>.
- [9] Niklas Donges. “Recurrent Neural Networks and LSTM.”
<https://towardsdatascience.com/recurrent-neural-networks-and-lstm-4b601dd822a5>.
- [10] “ฟังก์ชันเลขชี้กำลัง.”
<https://th.wikipedia.org/wiki/%E0%B8%9F%E0%B8%B1%E0%B8%87%E0%B8%81%E0%B9%8C%E0%B8%8A%E0%B8%B1%E0%B8%99%E0%B9%80%E0%B8%A5%E0%B8%82%E0%B8%8A%E0%B8%B5%E0%B9%89%E0%B8%81%E0%B8%B3%E0%B8%A5%E0%B8%B1%E0%B8%87>.
- [11] SAGAR SHARMA. “Activation Functions in Neural Networks.”
<https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>.
- [12] Kurayami Corp. “SiamChart: Global Market Summary.”
<http://siamchart.com/>.
- [13] QUANTITATIVE TRADING. “วิธีโหลดข้อมูลหุ้นจาก Siamchart มาใช้งานในPython.”
<https://qtmlresearch.com/2017/04/25/%E0%B8%A7%E0%B8%B4%E0%B8%98%E0%B8%B5%E0%B9%82%E0%B8%AB%E0%B8%A5%E0%B8%94%E0%B8%82%E0%B9%89%E0%B8%AD%E0%B8%A1%E0%B8%B9%E0%B8%A5%E0%B8%AB%E0%B8%B8%E0%B9%89%E0%B8%99%E0%B8%88%E0%B8%B2%E0%B8%81-siamchart/>.
- [14] นางสาวน้ำทิพย์ รัตนาวงษ์ไชยยา. “ปัญญาประดิษฐ์.”
https://il.mahidol.ac.th/e-media/computer/evolution/AI_what.html.

- [15] Abhishek Sharma. “Data Preprocessing for Machine learning in Python.”
<https://www.geeksforgeeks.org/data-preprocessing-machine-learning-python/>.
- [16] Witchapong Daroontham. “มาลอง forecast ราคาหุ้นแบบง่าย ด้วย Deep Learning— LSTM (Python).” <https://medium.com/@witchapoNgdaroontham/%E0%B8%A1%E0%B8%B2%E0%B8%A5%E0%B8%AD%E0%B8%87forecast%E0%B8%A3%E0%B8%B2%E0%B8%84%E0%B8%B2%E0%B8%AB%E0%B8%B8%E0%B9%89%E0%B8%99%E0%B9%81%E0%B8%9A%E0%B8%9A%E0%B8%87%E0%B9%88%E0%B8%B2%E0%B8%A2%E0%B9%86%E0%B8%94%E0%B9%89%E0%B8%A7%E0%B8%A2-deep-learning-lstm-python-305c480db223>.
- [17] “TensorFlow, Theano, Keras, Torch, Caffe.”
<https://project.inria.fr/deeplearning/files/2016/05/DLFrameworks.pdf>.
- [18] Hearty, John. *Advanced Machine Learning with Python*. Packt Publishing Ltd. 2016.
- [19] Giancarlo Zaccone, Md. Rezaul Karim, Ahmed Menshawy. *Deep Learning with TensorFlow*. Packt Publishing Ltd. 2017.
- [20] Zaccone, Giancarlo. *Getting Started with TensorFlow*. Packt Publishing Ltd. 2016.