การพัฒนาตัวแบบสำหรับการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม

Model Development for Sales Forecasting by Artificial Neural Network

ชาญชัย วงศ์โกศลจิต

สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (สถิติประยุกต์)
ภาควิชาสถิติ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Model Development for Sales Forecasting by Artificial Neural Network

Charnchai Wonkkosonjit

CO-OPERATIVE EDUCATION SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR

THE DEGREE OF BACHELOR OF SCIENCE (APPLIED STATISTICS)

DEPARTMENT OF STATISTICS, FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2018

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา	การพัฒนาตัวแบบสำหรับการพยากรณ์ยอดขายสินค้า โดยวิธีโครงข่ายประสาทเทียม Model Development for Sales Forecasting by Artificial Neural Network
ชื่อนักศึกษา	นายชาญชัย วงศ์โกศลจิต รหัสนักศึกษา 58051211
ปริญญา	วิทยาศาสตรบัณฑิต (สถิติประยุกต์)
ภาควิชา	สถิติ
ปีการศึกษา	2561
อาจารย์ที่ปรึกษา	ผศ.วราพร เหลือสินทรัพย์

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) อนุมัติให้สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา วิทยาศาสตรบัณฑิต (สถิติประยุกต์) ประจำปีการศึกษา 2561

คณะกรรมการสอบ	ลายมือชื่อ
ผศ.วราพร เหลือสินทรัพย์ ประธานกรรมการ	
ดร.สกุณา ศรีอโนมัย กรรมการ	
คุณชญาณ์นันท์ นวพรอนันต์ กรรมการ	
คุณวิศิษฐ์ กิจชัยนุกูล กรรมการ	

ลิขสิทธิ์ของคณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา	การพัฒนาตัวแบบสำหรับการพยากรณ์ยอดขายสินค้า โดยวิธีโครงข่ายประสาทเทียม Model Development for Sales Forecasting by Artificial Neural Network
ชื่อนักศึกษา	นายชาญชัย วงศ์โกศลจิต รหัสนักศึกษา 58051211
ปริญญา	วิทยาศาสตรบัณฑิต (สถิติประยุกต์)
ภาควิชา	สถิติ
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา	2561
อาจารย์ที่ปรึกษา	ผศ.วราพร เหลือสินทรัพย์

บทคัดย่อ

สหกิจศึกษาครั้งนี้มีวัตถุประสงค์เพื่อพัฒนาตัวแบบพยากรณ์ยอดขายเครื่องดื่มของแต่ละพื้นที่การขาย โดยวิธีโครงข่ายประสาทเทียม ด้วยกระบวนการ CRISP-DM และใช้ภาษา Python ข้อมูลที่ศึกษาในครั้งนี้คือผลิตภัณฑ์เครื่องดื่ม ซึ่งมีปริมาณยอดขายมากกว่าร้อยละ 80 ของผลิตภัณฑ์สินค้าทั้งหมด และพิจารณาข้อมูลออกเป็น 2 ส่วน ส่วนที่ 1 สำหรับการฝึกฝน (Training) และ ส่วนที่ 2 สำหรับทดสอบ (Testing) ตัวแบบ

ผลการทดลองได้ตัวแบบโครงข่าย 87-44-1 มีค่า MAPE เท่ากับ 28.31 ซึ่งน้อยกว่าตัวแบบพยากรณ์ของบริษัทที่มีค่า MAPE เท่ากับ 52.4 สรุปได้ว่าตัวแบบโครงข่ายประสาทเทียมที่สร้างขึ้นมีประสิทธิภาพมากกว่าของบริษัท ดังนั้นสามารถนำตัวแบบที่พัฒนานี้ไปแทนที่ตัวแบบของบริษัทเพื่อพยากรณ์ยอดขายสินค้าของแต่ละพื้นที่การขาย

คำสำคัญ : การพยากรณ์, โครงข่ายประสาทเทียม, กระบวนการ CRISP-DM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Model Development for Sales Forecasting by Artificial Neural Network
Students	Mr. Charnchai Wonkkosonjit 58051211
Degree	Bachelor of Science (Applied Statistics)
Department	Statistics
Faculty	Science
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)
Academic Year	2018
Advisor	Asst. Prof.Waraporn Lursinsap

Abstract

The objective of this cooperative study is to develop sales forecasting model for beverages in each sales area by artificial neural network with the CRISP-DM process and using Python language. Data in the study are beverages that have sales volume more than 80 percent of all products. The data are divided into 2 parts, for training model and testing model.

The result indicates that the model neural network 87-4 4-1 has MAPE of 28.31 which is less than the forecasting model of the company with MAPE of 52.4. In conclusion, the neural network model created is more efficient than the company's forecasting model. Therefore, this model can be used to replace the company's forecasting model in order to sales forecasting in each sales area.

Keywords : Forecasting, Neural Network, CRISP-DM Process

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การทำเล่มสหกิจศึกษาฉบับนี้สำเร็จได้ เนื่องจากได้รับความเมตตาจากอาจารย์ที่ปรึกษาสหกิจศึกษา ผศ.วราพร เหลือสินทรัพย์ ที่เมตตาให้คำปรึกษา และสละเวลาตรวจทานแก้ไขข้อบกพร่องต่างๆ ตลอดการทำเล่มสหกิจศึกษาฉบับนี้ ผู้จัดทำซาบซึ้งในความเมตตาจากท่านอาจารย์และกราบขอบพระคุณเป็นอย่างสูง ณ โอกาสนี้

รวมถึงกราบขอบพระคุณ ดร.สกุณา ศรีอโนมัย ที่ให้เกียรติเป็นคณะกรรมการสอบสหกิจ ที่เมตตาให้คำปรึกษา และสละเวลาตรวจทานแก้ไขข้อบกพร่องต่างๆ ตลอดการทำเล่มสหกิจศึกษาฉบับนี้ และกราบขอบพระคุณ คุณชญาณ์นันท์ นวพรอนันต์ และ คุณวิศิษฐ์ กิจชัยนุกูล ที่ให้ความเมตตาในการตรวจสอบ รวมทั้งให้คำปรึกษาเพื่อแก้ไขเครื่องมือที่ใช้ในการทำสหกิจให้มีคุณภาพ และขอกราบขอบพระคุณคณาจารย์ภาควิชาสถิติประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกๆท่านที่ได้มอบวิชาความรู้และประสบการณ์การเรียนการสอนกับผู้จัดทำ

ขอขอบพระคุณบิดามารดาที่ให้กำลังใจและให้การสนับสนุน รวมถึงขอขอบคุณเพื่อนนักศึกษาและพี่ร่วมงาน ที่ให้คำปรึกษา ให้กำลังใจในการทำสหกิจศึกษาครั้งนี้

สุดท้ายนี้ขอขอบคุณบริษัทกรณีศึกษาที่ได้อนุญาตให้นำข้อมูลมาใช้งานในการทำเล่มสหกิจศึกษาฉบับนี้

ชาญชัย วงศ์โกศลจิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ฉ

สารบัญรูป ..... ฌ

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ของงานวิจัย 2

1.3 ขอบเขตของงานวิจัย 2

1.4 ขั้นตอนการดำเนินงานวิจัย 3

1.5 ประโยชน์ที่คาดว่าจะได้รับ 3

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 4

2.1 การพยากรณ์ 4

2.2 กระบวนการ CRISP-DM 4

2.3 วิธีโครงข่ายประสาทเทียม (Artificial Neural Networks) 5

2.3.1 ประวัติความเป็นมาของโครงข่ายประสาทเทียม 5

2.3.2 สถาปัตยกรรมของโครงข่ายประสาทเทียม 6

2.3.3 โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ 11

2.3.4 เพอร์เซ็ปตรอนแบบหลายชั้น (Multi-Layer Perceptron) 11

2.3.5 ประเภทการเรียนรู้ของโครงข่ายประสาทเทียม 12

2.3.6 วิธีการหาค่าเหมาะสมที่สุด (Optimizer) 12

2.3.7 Adam 13

2.3.8 SGD 14

2.3.9 ฟังก์ชันสูญเสีย (Loss Function) 15

2.3.10 Dropout 15

2.4 การประเมินความแม่นยำจากการพยากรณ์ 16

2.5 งานวิจัยที่เกี่ยวข้อง 17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 วิธีการดำเนินงานวิจัย 20
3.1 การทำความเข้าใจธุรกิจ (Business Understanding) 21
3.2 การทำความเข้าใจข้อมูล (Data Understanding) 22
3.2.1 การเก็บข้อมูลขั้นต้น (Collect Initial Data) 22
3.2.2 การอธิบายข้อมูล (Describe Data) 23
3.2.3 การตรวจสอบคุณภาพของข้อมูล (Verify Data Quality) 24
3.3 การเตรียมข้อมูล (Data Preparation) 25
3.3.1 การคัดเลือกข้อมูล (Select Data) 25
3.3.2 การทำความสะอาดข้อมูล (Data Cleaning) 27
3.3.3 การสร้างข้อมูลใหม่ (Construct Data) 29
3.3.4 การแบ่งข้อมูล (Dividing Data) 30
3.3.4 การแปลงข้อมูล (Data Transformations) 31
3.4 การสร้างตัวแบบ (Modeling) 34
3.4 การประเมินผล (Evaluation) 38
3.4 การนำไปใช้ (Deployment) 39
บทที่ 4 ผลการวิจัยและการอภิปรายผล 40
4.1 ตัวแบบที่เหมาะสมสำหรับการพยากรณ์ 40
4.2 การเปรียบเทียบประสิทธิภาพของตัวแบบพยากรณ์ 64
บทที่ 5 สรุปผลการวิจัย การอภิปรายผล และข้อเสนอแนะ 67
5.1 สรุปผลการวิจัย 67
5.2 การอภิปรายผล 67
5.3 ข้อเสนอแนะ 68
บรรณานุกรม 69
ภาคผนวก 72
ภาคผนวก ก 73
ภาคผนวก ข 74

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

3.1 ข้อมูลยอดขายปี 2559-2561 21

3.2 แสดงร้อยละแต่ละสินค้า โดยแยกสินค้า A ขนาด AA กับสินค้า A ออกจากกัน 22

3.3 แสดงข้อมูลที่ได้จากฐานข้อมูลของบริษัท และตัวแปรของข้อมูล..... 23

3.4 แสดงตัวแปรของข้อมูลที่นำมาใช้ในการสร้างตัวแบบ 26

3.5 แสดง Python Code สำหรับการคัดเลือกข้อมูล (Select Data) 27

3.6 แสดง Python Code สำหรับการทำความสะอาดข้อมูล (Data Cleaning)..... 28

3.7 แสดง Python Code สำหรับการสร้างข้อมูลใหม่ (Construct Data)..... 30

3.8 แสดง Python Code สำหรับการแบ่งข้อมูล (Dividing Data)..... 31

3.9 แสดง Python Code สำหรับการแปลงข้อมูลคุณภาพ (Data Tranformations)..... 32

3.10 แสดง Python Code สำหรับการแปลงข้อมูลปริมาณ (Data Tranformations)..... 33

3.11 แสดงการกำหนดค่าไฮเปอร์พารามิเตอร์หรือกระบวนการในการสร้างตัวแบบในแต่ละข้อกำหนด..... 35

3.12 แสดง Python Code สำหรับตรวจสอบรอบในการเรียนรู้ (Epoch) ที่เหมาะสม 37

3.13 แสดง Python Code สำหรับการสร้างตัวแบบ (Modeling)..... 38

4.1 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ Adam และ Activation Function คือ Sigmoid (Hidden Layer), Sigmoid (Output Layer)..... 40

4.2 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ Adam และ Activation Function คือ Relu (Hidden Layer) , Relu (Output Layer)..... 42

4.3 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ Adam และ Activation Function คือ Relu (Hidden Layer) , Sigmoid (Output Layer)..... 43

4.4 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ Adam และ Activation Function คือ Sigmoid (Hidden Layer) , Relu (Output Layer)..... 44

4.5 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ SGD และ Activation Function คือ Sigmoid (Hidden Layer) , Sigmoid (Output Layer) และโมเมนตัม 0.5..... 45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ SGD และ Activation Function คือ Sigmoid (Hidden Layer) , Sigmoid (Output Layer) และโมเมนตัม 0.7 46
4.7 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ SGD และ Activation Function คือ Sigmoid (Hidden Layer) , Sigmoid (Output Layer) และโมเมนตัม 0.9 48
4.8 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ SGD และ Activation Function คือ Relu (Hidden Layer) , Relu (Output Layer) และโมเมนตัม 0.5 49
4.9 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ SGD และ Activation Function คือ Relu (Hidden Layer) , Relu (Output Layer) และโมเมนตัม 0.7 50
4.10 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ SGD และ Activation Function คือ Relu (Hidden Layer) , Relu (Output Layer) และโมเมนตัม 0.9 51
4.11 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ SGD และ Activation Function คือ Relu (Hidden Layer) , Sigmoid (Output Layer) และโมเมนตัม 0.5 52
4.12 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ SGD และ Activation Function คือ Relu (Hidden Layer) , Sigmoid (Output Layer) และโมเมนตัม 0.7 54
4.13 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ SGD และ Activation Function คือ Relu (Hidden Layer) , Sigmoid (Output Layer) และโมเมนตัม 0.9 55
4.14 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ SGD และ Activation Function คือ Sigmoid (Hidden Layer) , Relu (Output Layer) และโมเมนตัม 0.5 56
4.15 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ SGD และ Activation Function คือ Sigmoid (Hidden Layer) , Relu (Output Layer) และโมเมนตัม 0.7 57
4.16 แสดงโครงข่ายประสาทเทียมที่ดีที่สุดแต่ละ Optimizer และ Activation Function โดยสนใจที่ชุดข้อมูลฝึกฝน (Training Set) และ MAPE 58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.17 แสดงโครงข่ายประสาทเทียมที่ดีที่สุดแต่ละ Optimizer และ Activation Function โดยสนใจที่ชุดข้อมูลฝึกฝน (Training Set) และ MAD 59
4.18 แสดงโครงข่ายประสาทเทียมที่ดีที่สุดแต่ละ Optimizer และ Activation Function โดยสนใจที่ชุดข้อมูลทดสอบ (Test Set) และ MAPE 60
4.19 แสดงโครงข่ายประสาทเทียมที่ดีที่สุดแต่ละ Optimizer และ Activation Function โดยสนใจที่ชุดข้อมูลทดสอบ (Test Set) และ MAD 61
4.20 แสดงโครงข่ายประสาทเทียมที่ดีที่สุดแต่ละ Optimizer และ Activation Function โดยสนใจที่ชุดข้อมูลทวนสอบ (Validation Set) และ MAPE 62
4.21 แสดงโครงข่ายประสาทเทียมที่ดีที่สุดแต่ละ Optimizer และ Activation Function โดยสนใจที่ชุดข้อมูลทวนสอบ (Validation Set) และ MAD 63
4.22 แสดงรายละเอียดของตัวแบบที่เหมาะสมที่สุดที่ตัวแทนของบริษัทเลือก โดยพิจารณาจากค่า MAPE ของชุดข้อมูลทดสอบ (Test Set) 64
4.23 เปรียบเทียบประสิทธิภาพของการพยากรณ์โดยใช้ข้อมูลชุดทดสอบ 65
ก.1 ข้อมูลยอดขาย เดือนมกราคม 2559 – เดือนกุมภาพันธ์ 2562 73
ข.1 คำสั่งที่ใช้ในการเตรียมข้อมูลเพื่อสร้างตัวแบบพยากรณ์ยอดขายสินค้า 74
ข.2 คำสั่งที่ใช้ในการสร้างตัวแบบ บันทึกโครงสร้างตัวแบบ และค่าของน้ำหนัก 81
ข.3 คำสั่งที่ใช้ในการนำตัวแบบที่บันทึก และค่าของน้ำหนักไว้ มาทำการหาค่าไฮเปอร์พารามิเตอร์ที่เหมาะสมที่สุด 85

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 โครงข่ายประสาทมนุษย์	6
2.2 แสดงถึงโครงข่ายประสาทเทียมที่มีจำนวนชั้นทั้งหมด 3 ชั้น	7
2.3 องค์ประกอบของนิวรอน (Neuron)	7
2.4 ฟังก์ชันซิกมอยด์	9
2.5 ฟังก์ชันไฮเปอร์โบลิกแทนเจนต์	10
2.6 ฟังก์ชันลี่ลู	11
2.7 (a) ตัวแบบโครงข่ายประสาทเทียม (b) ตัวแบบโครงข่ายประสาทเทียมหลัง Dropout	16
3.1 ขั้นตอนCRISP-DM	20
3.2 กราฟเปรียบเทียบยอดขายจริง (สีน้ำเงิน) กับการประมาณยอดขายสินค้าของแต่ละพื้นที่การขาย (สีเขียว) มกราคม พ.ศ. 2559 ถึง ธันวาคม พ.ศ. 2561	21
3.3 แผนภูมิแสดงยอดขายแต่ละสินค้า	22
3.4 แสดงยอดขายจริง (Actual) ตั้งแต่ปี 2559 ถึงปี2561	24
3.5 แสดงราคาสินค้าต่อขวด (Price Per Unit) ในช่วงระยะเวลาปี 2559 ถึงปี 2561	24
3.6 แสดงข้อมูลการประมาณยอดขายที่ปรับ Budget (LE) ทางฝั่งการกระจายสินค้าสมัยใหม่ (Modern Trade)	25
3.7 แสดงข้อมูลยอดขายจริง (Actual) ในแต่ละปี	29
3.8 แสดงจำนวนรอบในการเรียนรู้ที่เหมาะสม (Epoch)	37
4.1 กราฟเปรียบเทียบยอดขายจริง (สีน้ำเงิน) ตัวแบบการพยากรณ์ยอดขายสินค้า โดยวิธีโครงข่ายประสาทเทียม (สีแดง) และการพยากรณ์ยอดขายสินค้าของแต่ละพื้นที่การขาย (Sale Forecast) (สีเขียว)	65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันเศรษฐกิจมีการแข่งขันที่สูงมาก เนื่องจากสินค้ามีความหลากหลายมากขึ้น มีการแข่งขันด้านราคา และการบริการที่สูงขึ้นมากกว่าอดีตที่ผ่านมา ดังนั้นบริษัทต่างๆ จึงหาวิธีมาใช้เพื่อปรับปรุงและพัฒนาสินค้าอยู่ตลอดเวลา เพื่อให้การสนับสนุนความต้องการของลูกค้าได้อย่างทันท่วงที เพราะถ้าหากให้การสนับสนุนที่ไม่ดีพอ อาจนำมาซึ่งการสูญเสียลูกค้าให้กับบริษัทอื่น ซึ่งถือว่าเป็นการสูญเสียที่รุนแรงสำหรับบริษัทเป็นอย่างมาก หนึ่งในวิธีที่ช่วยในการสนับสนุนความต้องการของลูกค้าได้อย่างทันท่วงที คือการพยากรณ์ความต้องการสินค้าที่คาดว่าจะเกิดขึ้นในช่วงเวลาหนึ่งในอนาคต เพื่อจะได้วางแผนในการเตรียมวัตถุดิบในการผลิตเพื่อตอบสนองความต้องการของลูกค้า รวมถึงโปรโมชั่นเพื่อกระตุ้นยอดขายในบางเดือน

ในการทำสหกิจศึกษาครั้งนี้ พบว่าบริษัทเป็นธุรกิจประเภทเครื่องดื่ม มีการกระจายสินค้าให้กับลูกค้า 3 ประเภทใหญ่ๆ คือ การขายสินค้าแบบดั้งเดิม (Traditional Trade) การขายสินค้าแบบสมัยใหม่ (Modern Trade) และการขายให้กับรถกระจายสินค้าหรือร้านค้า (Cash Van / On Trade) ทางบริษัทจึงได้ทำการพยากรณ์ยอดขายแต่ละการกระจายสินค้าให้กับลูกค้า เพื่อที่จะได้สนับสนุนการขายของลูกค้าได้อย่างครอบคลุม ซึ่งบริษัทได้ทำการพยากรณ์ยอดขายสินค้ามา 3 ระดับ คือ การประมาณยอดขายของทางผู้บริหาร (Budget) การประมาณยอดขายที่ปรับ Budget (LE) และการพยากรณ์ยอดขายสินค้าของแต่ละพื้นที่ (Sale Forecast) โดยที่ 3 ระดับนี้มีการพยากรณ์ที่ยังคลาดเคลื่อนกับยอดขายจริงอยู่มากในบางเดือน ทางบริษัทจึงต้องการตัวแบบใหม่ที่มีความแม่นยำกว่าตัวแบบเดิม เพื่อที่จะสนับสนุนเพื่อลดความคลาดเคลื่อนในบางเดือน ทั้งนี้การหาตัวแบบใหม่มาทดแทนนั้น จะมาทดแทนได้เพียงตัวแบบการพยากรณ์ยอดขายสินค้าของแต่ละพื้นที่ (Sale Forecast) เท่านั้น เนื่องด้วยการประมาณยอดขายของทางผู้บริหาร (Budget) และการประมาณยอดขายที่ปรับ Budget (LE) เป็นการประมาณที่อยู่ในระดับผู้บริหารไม่สามารถแทนที่ได้

ดังที่กล่าวมาข้างต้น สหกิจศึกษาครั้งนี้จึงทำการพยากรณ์ยอดขายสินค้า โดยทางบริษัทได้มอบหมายให้สร้างตัวแบบที่พยากรณ์ยอดขายสินค้าให้กับการกระจายสินค้า 2 ประเภท คือ การขายสินค้าแบบดั้งเดิม (Traditional Trade) และการขายสินค้าแบบสมัยใหม่ (Modern Trade) โดยผู้จัดทำได้เลือกเอาข้อมูลจากฐานข้อมูลของบริษัท มาทำการศึกษาเพื่อสร้างตัวแบบที่จะใช้สำหรับการพยากรณ์ยอดขายสินค้า โดยใช้วิธีโครงข่ายประสาทเทียม สำหรับใช้ทดแทนการพยากรณ์ยอดขายสินค้าของแต่ละพื้นที่ (Sale Forecast) และผลลัพธ์ที่ได้เพื่อเป็นแนวทางในการวางแผนการขาย การผลิต การกำหนดสินค้าคงคลังที่เหมาะสม การขนส่ง ที่จะเกิดขึ้นในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของงานวิจัย

1) เพื่อสร้างตัวแบบที่เหมาะสมสำหรับการพยากรณ์สินค้า โดยใช้วิธีโครงข่ายประสาทเทียม

2) นำตัวแบบที่เหมาะสมมาแทนที่การพยากรณ์ยอดขายสินค้าของแต่ละพื้นที่ (Sale Forecast) เพื่อเพิ่มความแม่นยำในการพยากรณ์ยอดขายสินค้าให้กับบริษัท

1.3 ขอบเขตของงานวิจัย

1) ข้อมูลที่ได้นำมาศึกษาในการวิจัยครั้งนี้เป็นข้อมูลทุติยภูมิ โดยทำการเก็บรวบรวมมาจากฐานข้อมูลของบริษัท ซึ่งเป็นข้อมูลรายเดือนตั้งแต่เดือนมกราคม พ.ศ.2558 ถึงเดือนกุมภาพันธ์ พ.ศ. 2562 ซึ่งประกอบด้วย

1.1) ปริมาณยอดขาย (ลิตร)

1.2) การประมาณยอดขายที่ปรับ Budget (ลิตร)

1.3) พื้นที่การขาย

1.4) ปีที่ทำการขาย

1.5) เดือนที่ทำการขาย

1.6) ราคาขายรวม (บาท)

1.7) ภาษีรวม (บาท)

1.8) ปริมาณยอดขายรวม (SKU)

1.9) ปริมาณยอดขายรวม (ขวด)

1.10) ราคาสินค้าต่อขวด (บาท)

1.11) วันที่ทำการขาย

1.12) ภูมิภาคการขาย

1.13) ช่องทางการกระจายสินค้า

2) ตัวแบบที่ใช้ทำการพยากรณ์ยอดขายสินค้า A ขนาด AA ใช้วิธีโครงข่ายประสาทเทียม และใช้โครงข่ายแบบเพอร์เซ็ปตรอนหลายชั้น

3) ตัวแบบใช้ได้กับสินค้า A ขนาด AA และใช้กับการกระจายสินค้าแบบดั้งเดิม (Traditional Trade) เท่านั้น

4) สร้างตัวแบบโดยใช้ภาษา Python โดยใช้ไลบรารี Keras ประมวลผลบน Jupyter Notebook

5) เกณฑ์ที่ใช้ในการเปรียบเทียบความแม่นยำในการพยากรณ์คือ MAPE และ MAD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนการดำเนินงานวิจัย

งานวิจัยนี้ได้ดำเนินการตามขั้นตอนของ Cross-Industry Standard Process for Data Mining (CRISP-DM) โดยมีขั้นตอนดังนี้ (Eakasit Pacharawongsakda, Ph.D, 2558)

1) การทำความเข้าใจธุรกิจ (Business Understanding)

2) การทำความเข้าใจข้อมูล (Data Understanding)

3) การเตรียมข้อมูล (Data Preparation)

4) การสร้างตัวแบบ (Modeling)

5) การประเมินผล (Evaluation)

6) การนำไปใช้ (Deployment)

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้ตัวแบบที่มีประสิทธิภาพในการพยากรณ์ยอดขายสินค้า โดยวิธีโครงข่ายประสาทเทียม

2) ได้ตัวแบบที่มีประสิทธิภาพมาแทนที่การพยากรณ์ยอดขายสินค้าของแต่ละพื้นที่ (Sale Forecast) เพื่อเพิ่มความแม่นยำในการพยากรณ์ให้กับทางบริษัท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 การพยากรณ์ (Forecasting)

อัจฉรา จันทร์ฉาย (2544) ได้ให้คำนิยาม การพยากรณ์ คือ การคาดการณ์ถึงสิ่งใดสิ่งหนึ่งที่จะเกิดขึ้นในช่วงเวลาในอนาคตและนำค่าพยากรณ์ที่ได้นั้นมาใช้ประโยชน์เพื่อการตัดสินใจใด ๆ โดยทั่วไปแล้ว พยากรณ์จะถูกจัดแบ่งตามหน้าที่หลักๆ ที่เกี่ยวข้อง

กัลยา วานิชย์บัณชา (2544) ได้ให้คำนิยาม การพยากรณ์ คือ การประมาณหรือการคาดคะเนว่าอะไรจะเกิดขึ้นในอนาคต การพยากรณ์แต่ละปัญหามีความแตกต่างกัน ในการพยากรณ์แต่ละปัญหาจึงควร ต้องมีการพิจารณาถึงปัจจัยที่สำคัญบางประการ เช่น ระยะเวลาในอนาคตที่ต้องการพยากรณ์เพื่อนำผลที่ได้มาช่วยสนับสนุนการตัดสินใจของผู้บริหาร

นิภา นิรุตติกกุล (2549) ได้ให้คำนิยาม การพยากรณ์ คือ การคาดคะเนหรือทำนายการเกิดเหตุการณ์หรือสภาพการณ์ต่างๆ ในอนาคต โดยการพยากรณ์ จะทำการศึกษาแนวโน้มและรูปแบบการเกิดเหตุการณ์จากข้อมูลในอดีตและ/หรือใช้ความรู้ ความสามารถ ประสบการณ์ และดุลยพินิจของผู้พยากรณ์

2.2 กระบวนการ CRISP-DM (Eakasit Pacharawongsakda, Ph.D, 2558)

กระบวนการ CRISP-DM ประกอบด้วย 6 ขั้นตอน ได้แก่

1) การทำความเข้าใจธุรกิจ (Business Understanding)

เป็นขั้นตอนแรกของกระบวนการ CRISP-DM เป็นการทำความเข้าใจปัญหาที่เกิดขึ้นในธุรกิจหรือโอกาสเชิงธุรกิจ ระบุเป้าหมาย (Target) ที่ต้องการได้จากการวิเคราะห์ข้อมูลด้วยการทำเหมืองข้อมูล (Data Mining) และนำมาวางแผน

2) การทำความเข้าใจข้อมูล (Data Understanding)

ข้อมูลเป็นตัวแปรที่สำคัญที่ขาดไม่ได้ในการทำเหมืองข้อมูล (Data Mining) เป็นขั้นตอนที่ทำการเก็บรวบรวมข้อมูลที่เกี่ยวข้องเบื้องต้น ข้อมูลจะต้องมีความน่าเชื่อถือและถูกต้อง ข้อมูลที่ได้ให้มีปริมาณมากพอต่อการนำไปใช้

3) การเตรียมข้อมูล (Data Preparation)

เป็นขั้นตอนที่ใช้เวลานานที่สุดของกระบวนการ CRISP-DM เนื่องด้วยตัวแบบที่ได้จากการทำเหมืองข้อมูล (Data Mining) ความถูกต้องที่ได้จากผลลัพธ์จะขึ้นอยู่กับคุณภาพของข้อมูลที่จะนำไปวิเคราะห์ โดยการเตรียมข้อมูลนั้น สามารถแบ่งออกได้เป็น 5 ขั้นตอนย่อยคือ การคัดเลือกข้อมูล (Data Selection) การทำความสะอาดข้อมูล (Data Cleaning) การสร้างข้อมูลใหม่ (Construct Data) การแบ่งข้อมูล (Dividing Data) และการแปลงข้อมูล (Data Transformation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) การสร้างตัวแบบ (Modeling)

เป็นขั้นตอนการวิเคราะห์ข้อมูลด้วยเทคนิคการทำเหมืองข้อมูล (Data Mining) เช่น เทคนิคความสัมพันธ์ (Association) เทคนิคการจัดกลุ่ม (Clustering) เทคนิคการจำแนกกลุ่ม (Classification) และเทคนิคการทำนาย (Prediction)

5) การประเมินผล (Evaluation)

ในขั้นตอนนี้เป็นการประเมินประสิทธิภาพของผลลัพธ์ที่เกิดจากตัวแบบ ว่าครอบคลุมและสามารถตอบโจทย์ทางธุรกิจที่ตั้งไว้ในขั้นตอนแรกหรือไม่ ซึ่งอาจจะย้อนกลับไปยังขั้นตอนก่อนหน้าเพื่อเปลี่ยนแปลงแก้ไขเพื่อให้ได้ผลลัพธ์ตามที่ต้องการได้

6) การนำไปใช้ (Deployment)

การนำผลลัพธ์ที่ได้จากการวิเคราะห์ข้อมูลด้วยเทคนิคการทำเหมืองข้อมูล (Data Mining) นำไปใช้งาน ตัวอย่างเช่น การพยากรณ์ยอดขายสินค้า การจัดกลุ่มลูกค้าเพื่อจัดโปรโมชั่น เป็นต้น

2.3 วิธีโครงข่ายประสาทเทียม (Artificial Neural Networks) (พยุง มีสัจ, 2555)

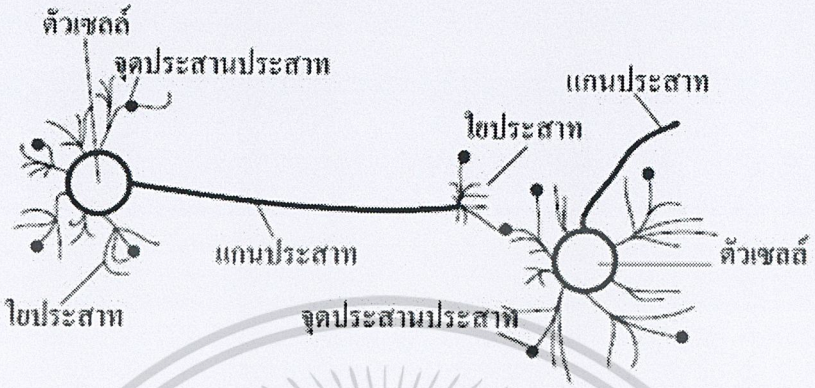
โครงข่ายประสาทเทียมเป็นการจำลองการทำงานของสมองมนุษย์ด้วยโปรแกรมคอมพิวเตอร์ เป็นแนวความคิด ที่ต้องการให้คอมพิวเตอร์มีความชาญฉลาดในการเรียนรู้เหมือนที่มนุษย์มีการเรียนรู้ สามารถฝึกฝนได้ และสามารถนำความรู้และทักษะไปแก้ปัญหาต่างๆ มีนักวิจัยจำนวนมากได้คิดค้นรูปแบบโครงข่ายประสาทเทียมแบบต่างๆ เพื่อนำมาประยุกต์ใช้อย่างกว้างขวาง การประยุกต์ใช้งานโครงข่ายประสาทเทียมมีตั้งแต่การใช้เพื่อตัดสินใจง่ายไปจนถึงงานที่มีความยุ่งยากซับซ้อน

2.3.1 ประวัติความเป็นมาของโครงข่ายประสาทเทียม (พยุง มีสัจ, 2555)

ประวัติความเป็นมาของโครงข่ายประสาทเทียมมีประวัติความเป็นมาย้อนหลังไปประมาณ 60 กว่าปีก่อน ในปี ค.ศ. 1943 McCuloch และ Pitts แห่งมหาวิทยาลัยชิคาโก ประเทศสหรัฐอเมริกา ได้นำเสนอบทความวิชาการ “Boolean Brain” ซึ่งได้กลายเป็นจุดกำเนิดของการจัดรูปแบบคณิตศาสตร์ของประสาทเทียม ต่อมาได้มีนักวิจัยได้คิดค้นรูปแบบโครงข่ายประสาทเทียมแบบต่างๆ มากมาย และทุกรูปแบบวิธีจะประกอบกับวิธีการสอนโครงข่ายด้วย ซึ่งวิธีการต่างๆ จะมีความซับซ้อนแตกต่างกันไปโครงข่ายประสาทเทียม (Artificial Neural Network : ANN) เป็นการจำลองการทำงานโครงข่ายประสาทของมนุษย์ (Biological Neurons) ซึ่งประกอบด้วยส่วนของการประมวลผลที่เรียกว่านิวรอล (Neurons) ซึ่งอยู่ในส่วนสมองที่ประกอบด้วยประสาทจำนวนมหาศาล และมีจุดต่อจำนวนโครงข่ายประสาทประกอบขึ้นด้วยส่วนสำคัญ 3 ส่วน คือ ใยประสาท (Nerve Fiber หรือ Dendrites) จุดกลางตัวเซลล์ (Cell Body หรือ Soma) และแกนประสาทนำออก (Axon) ในแต่ละโครงข่ายประสาทเทียมจะเชื่อมต่อกันที่จุดประสานประสาท (Synapse) ซึ่งสามารถเปลี่ยนค่าความต้านทานได้ตามสัญญาณที่ส่งระหว่างกันของเซลล์ประสาท การส่งสัญญาณระหว่างเซลล์ประสาทเป็นการถ่ายเทสารประกอบโซเดียมและโพแทสเซียม ภาพร่างของประสาทแสดงในรูปที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 จากภาพเส้นใยประสาททำหน้าที่รับสัญญาณไฟฟ้าจากเซลล์อื่นๆ เข้าตัวเซลล์หรือส่วนกลางเซลล์ ทำหน้าที่รวมสัญญาณที่เข้ามาแล้วสร้างสัญญาณออกไปยังแกนประสาทนำออก แอกซอนหรือแกนประสาทนำออกจะทำหน้าที่ส่งสัญญาณไปยังใยประสาทของเซลล์อื่นๆ ต่อไป



รูปที่ 2.1 โครงข่ายประสาทมนุษย์
(http://www.mut.ac.th/research-detail-92)

2.3.2 สถาปัตยกรรมของโครงข่ายประสาทเทียม (Fausett, L., 1994)

โครงสร้างของโครงข่ายประสาทเทียม ประกอบด้วย 3 องค์ประกอบ ดังนี้

1) ชั้นอินพุต (Input Layer) เป็นชั้นแรกในโครงสร้างของตัวแบบโครงข่ายประสาทเทียม จะประกอบด้วยนิวรอนรับข้อมูล (Input Neuron) จำนวนนิวรอนรับข้อมูลจะเท่ากับจำนวนของข้อมูลอินพุต โดยทำหน้าที่รับข้อมูลเข้าสู่ตัวแบบโครงข่ายประสาทเทียม

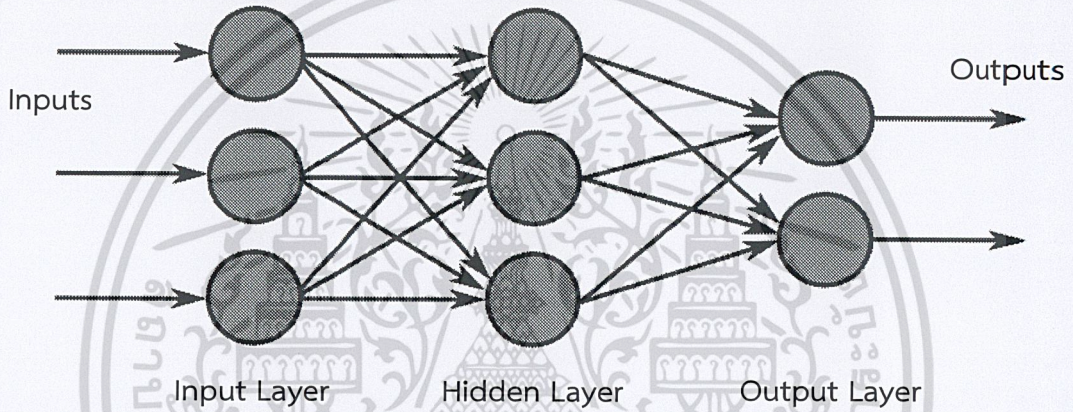
2) ชั้นเอาต์พุต (Output Layer) เป็นชั้นสุดท้ายในโครงสร้างของตัวแบบโครงข่ายประสาทเทียม จะประกอบด้วยนิวรอนแสดงผลข้อมูล (Output Neuron) จำนวนนิวรอนของข้อมูลจะเท่ากับจำนวนของข้อมูลเอาต์พุตของตัวแปร ทำหน้าที่แสดงผลลัพธ์ (Output) ของตัวแบบโครงข่ายประสาทเทียม

3) ชั้นซ่อน (Hidden Layer) จะอยู่ระหว่างชั้นอินพุตและชั้นเอาต์พุตประกอบด้วยนิวรอนในชั้นซ่อน (Hidden Neuron) ทำหน้าที่ในการรวมอินพุตที่คูณด้วยน้ำหนัก (Weight) ที่แปรค่าเพื่อแปลงให้เป็นผลลัพธ์ (Output) และส่งไปยังชั้นเอาต์พุต โดยไม่มีกฎตายตัวในการกำหนดจำนวนชั้นซ่อนและนิวรอนที่เหมาะสม นักวิจัยหลายท่าน (Masters, T., 1994 ; H.R. Maier, and G.C. Dandy, 2000) ได้แนะนำจำนวนชั้นซ่อนและนิวรอนที่เหมาะสมสามารถหาได้โดยกระบวนการลองผิดลองถูก (Trial and Error) ในแต่ละกรณีของปัญหา

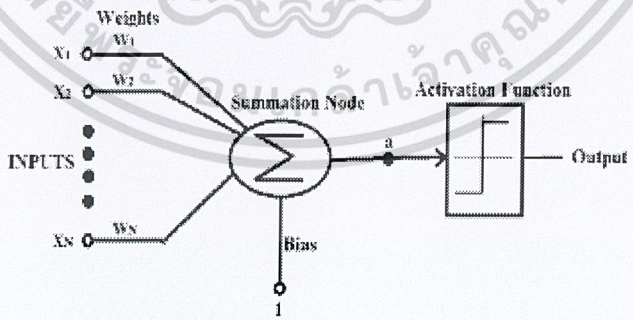
ตัวแบบโครงข่ายประสาทเทียม ซึ่งประกอบด้วยส่วนของการประมวลผลที่เรียกว่านิวรอน (Neuron) ทุกๆ นิวรอนสามารถมีอินพุตได้หลายอินพุตแต่มีเอาต์พุตเพียงเอาต์พุตเดียว และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทุกๆ เอาต์พุตจะแยกไปยังอินพุตของนิวรอนอื่นๆ ภายในโครงข่าย การติดต่อกันภายในระหว่างนิวรอนไม่ใช่ลักษณะการต่อแบบธรรมดา ทุกๆ อินพุตจะมีการกำหนดน้ำหนัก (Weight) ของการเชื่อมต่อภายใน เพื่อช่วยในการตัดสินใจการทำงานของนิวรอนในบางโครงข่ายจะถูกกำหนดไว้ตายตัว แต่บางโครงข่ายสามารถที่จะปรับแต่งได้ ซึ่งอาจจะเป็นการปรับแต่งจากภายนอกโครงข่าย หรือนิวรอนสามารถปรับได้ด้วยตัวเอง ในจุดนี้แสดงถึงความสามารถในการเรียนรู้และจดจำของโครงข่ายประสาทเทียม (พยุง มีสัจ, 2555) ซึ่งภาพโครงข่ายประสาทเทียมแสดงได้ดังรูปที่ 2.2 โดยองค์ประกอบที่ถูกบรรจุอยู่ในนิวรอนแต่ละตัวประกอบด้วย 2 ฟังก์ชันย่อย คือฟังก์ชันผลรวม (Summation Function) และ ฟังก์ชันกระตุ้น (Activation Function) และมีรายละเอียดการทำงานของนิวรอน แสดงดังรูปที่ 2.3



รูปที่ 2.2 แสดงถึงโครงข่ายประสาทเทียมที่มีจำนวนชั้นทั้งหมด 3 ชั้น
(https://python3.wannaphong.com/2016/04/neural-networks-python.html)



รูปที่ 2.3 องค์ประกอบของนิวรอน (Neuron)
(S.Arun Balaji and K.Baskaran. 2013.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) ฟังก์ชันผลรวม (Summation Function) (Hagan. Et al., 1996)

ฟังก์ชันผลรวม ทำหน้าที่ในการคำนวณผลรวมของข้อมูลที่ได้จากชั้นอินพุต (Input Layer) ซึ่งคำนวณได้ดังสมการที่ 2.1

$$S = \sum_{i=1}^{n} x_i w_i \tag{2.1}$$

กำหนดให้ S หมายถึง ผลลัพธ์ของฟังก์ชันผลรวม

n หมายถึง จำนวนนิวรอนของข้อมูลเข้า

x หมายถึง ค่าข้อมูลอินพุตตัวที่ i

w หมายถึง ค่าน้ำหนักของข้อมูลอินพุตตัวที่ i

2) ฟังก์ชันกระตุ้น (Activation Function) (ธนาวุฒิ ประกอบผล, 2552)

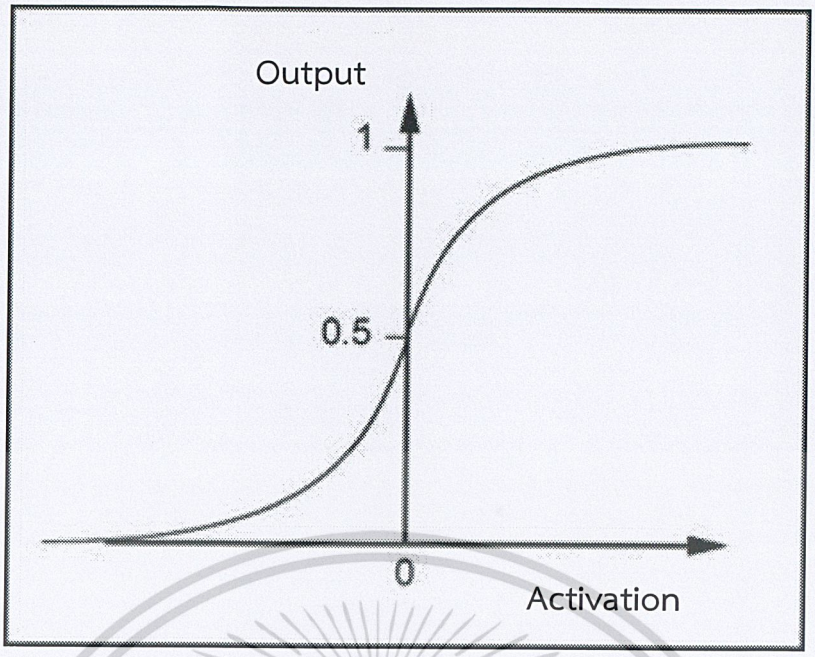
ทำหน้าที่ปรับค่าของข้อมูลที่ได้จากฟังก์ชันผลรวมให้อยู่ในช่วงที่ต้องการ ได้แก่ ฟังก์ชันซิกมอยด์ (Sigmoid Function) ฟังก์ชันไฮเปอร์โบลิกแทนเจนต์ (Hyperbolic Tangent Function) ฟังก์ชันลี่ลู (Rectified Linear Unit Function) เป็นต้น

2.1) ฟังก์ชันซิกมอยด์ (Sigmoid Function)

เป็นฟังก์ชันที่เปลี่ยนค่าทั้งหมดบนเส้นจำนวนให้กลายเป็นช่วง (0, 1) แสดงได้ดังรูปที่ 2.4 สามารถเขียนเป็นสมการทางคณิตศาสตร์ดังสมการที่ 2.2

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.2}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



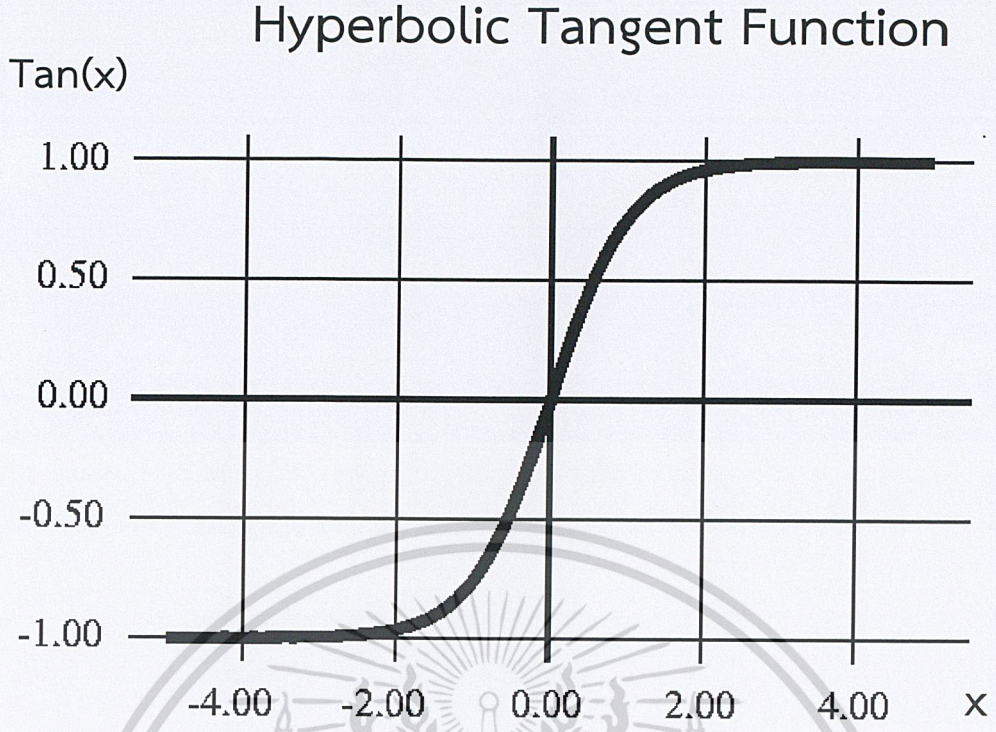
รูปที่ 2.4 ฟังก์ชันซิกมอยด์
(https://www.analyticsindiamag.com/most-common-activation-functions-in-neural-networks-and-rationale-behind-it/)

2.2) ฟังก์ชันไฮเปอร์โบลิกแทนเจนต์ (Hyperbolic Tangent Function)

ทำงานคล้าย Sigmoid แต่เปลี่ยนช่วงของ Output เป็น (-1, 1) แสดงได้ดังรูปที่ 2.5 ซึ่งสามารถเขียนเป็นสมการทางคณิตศาสตร์ดังสมการที่ 2.3

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.3}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



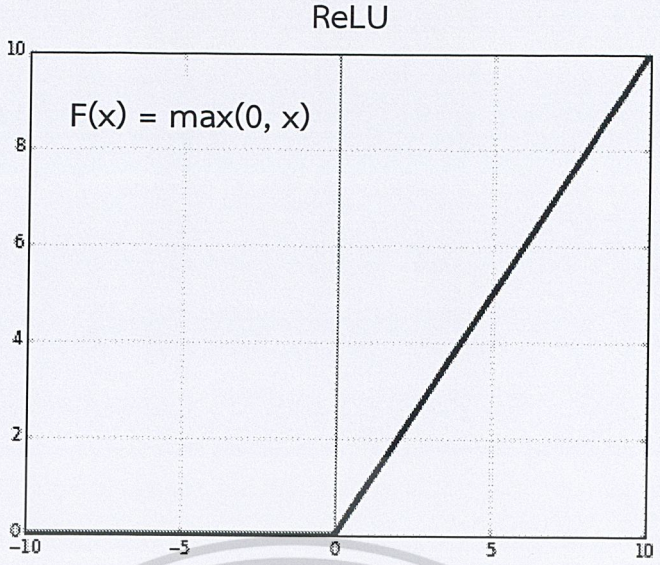
รูปที่ 2.5 ฟังก์ชันไฮเปอร์โบลิกแทนเจนต์
(https://www.researchgate.net/figure/Tanh-function_fig4_220204237)

2.3) ฟังก์ชันลี่ลู (Rectified Linear Unit Function)

เป็นฟังก์ชันที่นิยมที่สุดในปัจจุบัน เพราะความได้เปรียบในเชิงแคลคูลัส ในตอนที่ตัวแบบเรียนรู้เพื่อปรับค่า Weight จะทำได้เร็วกว่าแบบอื่นๆ ช่วงของ Output เป็น $[0, \infty)$ และแสดงได้ดังรูปที่ 2.6 ซึ่งสามารถเขียนเป็นสมการทางคณิตศาสตร์ดังสมการที่ 2.4

$$f(x) = \max(0, x) \tag{2.4}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 ฟังก์ชันลี่ลู
(https://medium.com/@kanchansarkar/relu-not-a-differentiable-function-why-used-in-gradient-based-optimization-7fef3a4cecec)

2.3.3 โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ

โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ เป็นตัวแบบโครงข่ายที่ได้รับความนิยม เนื่องจากสามารถแก้ปัญหาในลักษณะเชิงเส้น (Linear) และไม่เป็นเชิงเส้น (Nonlinear) ได้ ซึ่งมีลักษณะสำคัญ คือ จำนวนชั้นซ่อน (Hidden Layer) สามารถมีกี่ชั้นก็ได้ มีการเชื่อมโยงระหว่างโหนดในแต่ละชั้นเป็นการเชื่อมถึงกันหมด โดยจะมีการส่งค่าความคลาดเคลื่อนจากชั้นเอาต์พุต (Output Layer) กลับมาชั้นก่อนหน้า เพื่อปรับน้ำหนักระหว่างโหนด

2.3.4 เพอร์เซ็ปตรอนแบบหลายชั้น (Multi-Layer Perceptron) (เกียรติชัย เสนา, 2551)

โครงข่ายประสาทเทียมเพอร์เซ็ปตรอนแบบหลายชั้น (Multi-Layer Perceptron) เป็นโครงข่ายประสาทเทียมแบบป้อนไปข้างหน้า (Feed-Forward) ประกอบด้วยปมประสาท (Neural) ที่เรียกว่า เพอร์เซ็ปตรอน (Perceptron) โดยที่เพอร์เซ็ปตรอนประกอบด้วยปมประสาทเพียงหนึ่งยูนิต ที่มีตัวถ่วงน้ำหนักการเชื่อมโยง (Weight) และค่าไบแอส (Bias) โครงข่ายประสาทเทียมเพอร์เซ็ปตรอนหลายชั้นสามารถสร้างได้จากการนำเพอร์เซ็ปตรอนมาต่อกันเป็นโครงข่าย การเรียนรู้ของโครงข่ายก็คือ การหาค่าถ่วงน้ำหนักที่เหมาะสมโครงข่ายนั้นๆ การนำโครงข่ายประสาทเทียมมาใช้งานต้องมี 2 ขั้นตอน คือ ขั้นตอนการฝึกฝน (Training) และขั้นตอนการทดสอบ (Testing) ซึ่งในการฝึกฝนนั้นมี 2 ลักษณะ คือ การคำนวณไปข้างหน้า (Forward Propagation) และการคำนวณแพร่กลับ (Backward Propagation) ส่วนแรกการคำนวณไปข้างหน้าเป็นการป้อนข้อมูลขาเข้าให้กับโครงข่าย จากนั้นโครงข่ายจะทำการคำนวณจากขั้นขาเข้าไปยังชั้นขาออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.5 ประเภทการเรียนรู้ของโครงข่ายประสาทเทียม (สุพรรษา เหลืองอำนวยศิริ, 2551)

1) การเรียนรู้แบบมีผู้สอน (Supervised Learning) รูปแบบการเรียนรู้แบบมีผู้สอน เริ่มด้วยการส่งสิ่งเร้าที่ใช้ในการสอนเข้าไปเป็นอินพุต (Input) ซึ่งผลลัพธ์จะเป็นอย่างไรขึ้นอยู่กับสภาวะในตอนที่เริ่มเรียนรู้ของโครงข่ายประสาทเทียม โดยผลลัพธ์จะเป็นอย่างไรขึ้นอยู่กับสภาวะในตอนที่เริ่มเรียนรู้ของโครงข่ายประสาทเทียม โดยผลลัพธ์ที่ได้นั้นจะถูกนำมาเปรียบเทียบกับผลลัพธ์เป้าหมายซึ่งผู้สอนจะเป็นผู้สร้างขึ้น หากมีความแตกต่างกันนั่นคือมีความคลาดเคลื่อนเกิดขึ้น ความคลาดเคลื่อนดังกล่าวจะถูกนำไปคำนวณการปรับแต่งค่าน้ำหนักต่างๆ ในโครงข่ายประสาทเทียมเพื่อลดความคลาดเคลื่อนลงให้เหลือน้อยที่สุด

2) การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) การเรียนรู้แบบไม่มีผู้สอน ไม่จำเป็นต้องมีค่าเป้าหมายของแต่ละข้อมูลตัวอย่าง จะสอนโครงข่ายโดยการปรับข้อมูลอินพุตอย่างต่อเนื่องเพียงอย่างเดียว การปรับน้ำหนักจะปรับตามกลุ่มข้อมูลที่นำมาสอนโดยปรับเข้าตามกลุ่มที่มีรูปแบบที่คล้ายๆ กัน

2.3.6 วิธีการหาค่าเหมาะสมที่สุด (Optimizer) (S.V.G.Reddy. et al, 2018)

การเคลื่อนลงตามความชัน (The Gradient Descent) เป็นหนึ่งในเทคนิคที่นิยมมากที่สุดในการดำเนินการเพิ่มประสิทธิภาพสำหรับโครงข่ายประสาทเทียม เพื่อให้เข้าใจ ลองจินตนาการว่ามีคนยืนอยู่บนยอดเขา และถ้าเขาต้องการที่จะไปจุดต่ำสุดของภูเขาด้วยมีเงื่อนไขว่าเขาตาบอดปราศจากการมองเห็น วิธีที่ดีที่สุดคือการเลือกพื้นผิวซึ่งมีแนวโน้มที่จะลงมาเป็นขั้นตอนแรก โดยทำตามกลยุทธ์ที่คล้ายกันนี้ เขาใช้ขั้นตอนต่างๆ ในการไปถึงจุดต่ำสุด ทีนี้เราจำเป็นต้องหาพารามิเตอร์ที่ดีที่สุดสำหรับอัลกอริทึม และค่าใช้จ่าย (Cost) ถึงจะไม่มีค่าใช้จ่าย แต่ประสิทธิภาพของอัลกอริทึมจะแตกต่างกันตามค่าของพารามิเตอร์การเคลื่อนลงตามความชัน (The Gradient Descent) มีหลายรูปแบบ เช่น Batch, Stochastic และ Mini Batch

1) The Batch Gradient Descent

จะใช้กับข้อมูลฝึกฝนทั้งหมด และคำนวณการเคลื่อนลงตามฟังก์ชันค่าใช้จ่าย (Gradient of the Cost Function) โดยพิจารณาจากค่าพารามิเตอร์

2) The Stochastic Gradient Descent

เป็นสิ่งที่ดำเนินการปรับปรุงพารามิเตอร์ สำหรับทุกตัวอย่างฝึกฝน (Training Example)

3) The Mini Batch

ใช้กลยุทธ์ที่ดีที่สุดจากสองกลยุทธ์ข้างต้น โดยทำการปรับปรุงพารามิเตอร์สำหรับตัวอย่างฝึกฝนทุกชุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในขั้นต้น การเรียนรู้เชิงลึก (Deep Learning) ได้ใช้อัลกอริทึม Gradient Descent พื้นฐาน ซึ่งตัวแบบที่ใช้ในการรันมีประสิทธิภาพน้อยมาก ในงานวิจัยนี้ ได้นำเสนอ อัลกอริทึมเพิ่มประสิทธิภาพ Few Gradient Descent เช่น Adam สำหรับโครงข่ายประสาทเทียม

2.3.7 Adaptive Moment Estimation (Adam) (S.V.G.Reddy. et al, 2018)

Adam เป็นหนึ่งในอัลกอริทึม Optimizer ที่ดีที่สุด ที่คำนวณอัตราการเรียนรู้ (Learning Rate) ไฮเปอร์พารามิเตอร์ที่จำเป็น คือ อัตราการเรียนรู้ทั้งหมด (α) อัตราสลายตัวสำหรับการประมาณช่วงเวลา (β_1) และ (β_2) ค่าน้ำหนักเริ่มต้น (θ_t) และค่าคงที่ (δ) ปกตินิยมใช้ (10^{-8}) โดยมีขั้นตอนการปรับปรุงค่าพารามิเตอร์ดังนี้

1) เริ่มต้นตัวแปรช่วงเวลาที่ 1 และ 2 $r = 0$ และ $s = 0$

2) คำนวณค่าการไล่ระดับสี (Gradients) $g_t = \nabla_\theta f_t(\theta_t)$

3) อัพเดตช่วงเวลาแรกที่มีอคติ $s = \beta_1 s + (1 - \beta_1) g_t$

4) อัพเดตช่วงเวลาที่สองที่มีอคติ $r = \beta_2 r + (1 - \beta_2) g_t^{\,2}$

5) คำนวณค่าประมาณช่วงเวลาแรกที่ได้รับการแก้ไขโดยอคติ

$$\tilde{s} = \frac{s}{1 - \beta_1^{\,t}}$$

6) คำนวณค่าประมาณช่วงเวลาที่สองที่ได้รับการแก้ไขโดยอคติ

$$\tilde{r} = \frac{r}{1 - \beta_2^{\,t}}$$

7) คำนวณการอัพเดต $\Delta\theta_{(t+1)} = \alpha \cdot \frac{\tilde{s}}{\delta + \sqrt{\tilde{r}}}$

8) ใช้การอัพเดต $\theta_{(t+1)} = \theta_t - \Delta\theta_{(t+1)}$

จะทำการคำนวณอย่างนี้ไปจนกว่าจะหมดรอบการเรียนรู้

Adam พิจารณาค่าเฉลี่ยการสลายตัวชี้กำลังของการไล่ระดับสี (Gradients) (เช่น โมเมนตัม (Momentum)) และการไล่ระดับสีกำลังสอง (Squared Gradients) และพวกเขาเรียกว่าช่วงเวลาแรก (First Moment) และช่วงเวลาที่ 2 (Second Moment) ตามลำดับ และด้วยเหตุนี้จึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อว่าช่วงเวลาการปรับตัว (Adaptive Moment : Adam) ที่ผ่านมาคือการคำนวณ การไล่ระดับสี (Gradients) และการไล่ระดับสีกำลังสอง (Squared Gradients) และการเอนเอียง (Bias) ส่วนใหญ่ใกล้จะเป็น 0 และหลังจากนั้นจะคำนวณ เพื่อแก้ไขการเอนเอียงช่วงเวลาแรก (First Moment) และช่วงเวลาที่ 2 (Second Moment) และท้ายที่สุดค่าน้ำหนักจะถูกปรับปรุง

2.3.8 Stochastic Gradient Descent (SGD) (มนัสกานต์ เสน่หา, 2559)

การเรียนรู้ด้วย SGD มีการปรับค่าของน้ำหนัก โดยนำค่าน้ำหนักก่อนหน้ามาคำนวณ ดังสมการที่ 2.5

$$w_t = w_{t-1} - \alpha \frac{\partial MSE_t}{\partial w} \tag{2.5}$$

กำหนดให้ w หมายถึง พารามิเตอร์ซึ่งเป็นน้ำหนักที่ต้องการจะปรับค่า

α หมายถึง อัตราการเรียนรู้ (Learning Rate)

$\frac{\partial MSE}{\partial w}$ หมายถึง การไล่ระดับสี (Gradients) ของฟังก์ชันสูญเสีย เทียบกับ w

การที่จะทำให้การเรียนรู้มีการลู่เข้าดีขึ้นและหลีกเลี่ยงการติดอยู่ที่โลคอลออพติมา (Local Optima) ซึ่งเป็นค่าที่เหมาะสมที่สุดในช่วงๆ หนึ่งแต่ไม่ใช่ทั้งหมด สามารถใช้โมเมนตัม (Momentum) แก้ปัญหาดังกล่าว ดังสมการที่ 2.6 และ 2.7

$$v_t = \gamma v_{t-1} + \alpha \frac{\partial MSE_t}{\partial w} \tag{2.6}$$

$$w_t = w_{t-1} - v_t \tag{2.7}$$

กำหนดให้ v หมายถึง ค่าความเร็วซึ่งมีการปรับค่าพร้อมกับ w

γ หมายถึง ค่าสัมประสิทธิ์ของโมเมนตัม (Momentum Coefficient)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.9 ฟังก์ชันสูญเสีย (Loss Function) (S.V.G.Reddy. et al, 2018)

ฟังก์ชันสูญเสียที่ได้รับความนิยมคือ ค่าเฉลี่ยกำลังสองความคลาดเคลื่อน (Mean Squared Error : MSE) และไบนารีครอส-เอนโทรปี (Binary Cross Entropy) รวมไปถึงอัลกอริทึมเพิ่มประสิทธิภาพที่อยู่ด้านบน การคำนวณค่าเฉลี่ยกำลังสองความคลาดเคลื่อน จะทำในลักษณะดังนี้

1) สำหรับชั้นอินพุต, อินพุตหรือตัวแปรของข้อมูลชุดฝึกฝนคูณด้วยค่าน้ำหนัก แล้วส่งไปยังชั้นซ่อน

2) สำหรับชั้นซ่อน รับเอาท์พุตจากชั้นอินพุตคูณด้วยค่าน้ำหนัก โดยนำมาคำนวณด้วยฟังก์ชันกระตุ้น แล้วส่งไปชั้นเอาท์พุต

3) สำหรับชั้นเอาท์พุต รับเอาท์พุตจากชั้นซ่อนคูณด้วยค่าน้ำหนัก โดยนำมาคำนวณด้วยฟังก์ชันกระตุ้น

4) เอาท์พุตสุดท้ายจะถูกคำนวณออกมา ความแตกต่างระหว่างค่าจริงและค่าเอาท์พุตสุดท้าย เรียกว่าความคลาดเคลื่อน ขั้นตอนนี้จะคำนวณความคลาดเคลื่อนทั้งหมดในตัวอย่างฝึกฝน (Training Example) , นำความคลาดเคลื่อนมายกกำลังสอง และคำนวณหาค่าเฉลี่ยของความคลาดเคลื่อนทั้งหมด

สิ่งนี้คือ ค่าเฉลี่ยกำลังสองความคลาดเคลื่อน (Mean Squared Error : MSE) ที่มีการอัปเดตค่าน้ำหนักเริ่มต้น และวนซ้ำโดยเริ่มจากชั้นอินพุตจนถึงความคลาดเคลื่อนลู่เข้าสู่เกณฑ์ขั้นต่ำ ค่าเฉลี่ยกำลังสองความคลาดเคลื่อน (Mean Squared Error : MSE) มีสมการดังที่ 2.8

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 \qquad (2.8)$$

กำหนดให้ $\hat{x}_i$ หมายถึง ค่าเอาท์พุตสุดท้าย ณ ข้อมูลที่ i

n หมายถึง จำนวนข้อมูลในตัวอย่างฝึกฝน

x_i หมายถึง ค่าจริง ณ ข้อมูลที่ i

2.3.10 Dropout (S.V.G.Reddy. et al, 2018)

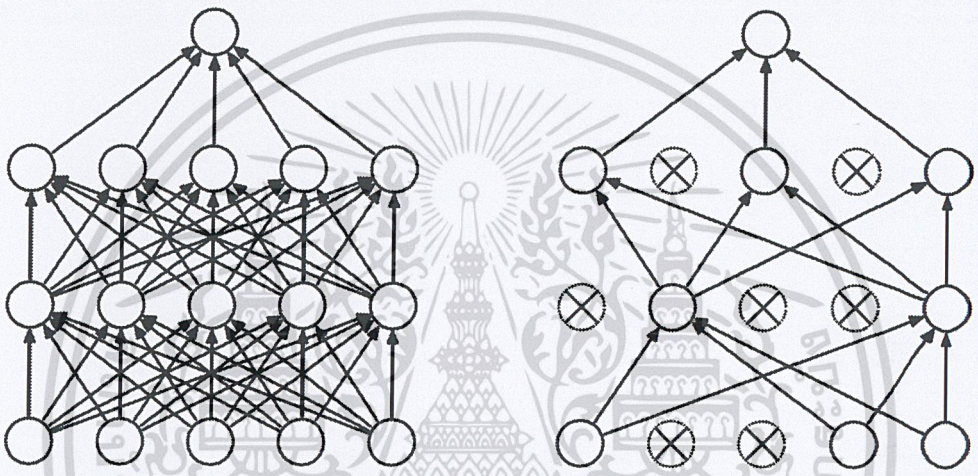
โดยทั่วไปปัญหาที่พบเจอได้บ่อยที่สุดของวิธีโครงข่ายประสาทเทียม คือ Overfitting

Overfitting หมายถึง รูปแบบมีประสิทธิภาพมากเกินไปกับชุดข้อมูลฝึกฝน แต่ไม่สามารถทำงานได้ดีกับชุดข้อมูลทดสอบ เพื่อลดปัญหา Overfitting คือการนำกระบวนการทำให้เป็น Regularization ไปใช้กับตัวแบบ Regularization เป็นเทคนิคการดัดแปลงเล็กน้อยกับตัวแบบที่มีอยู่เดิม และอัลกอริทึมการเรียนรู้ เพื่อให้ทำงานได้ดีทั้งในการฝึกฝนและการทดสอบ มีเทคนิคการทำให้เป็น Regularization หลากหลายรูปแบบในการเรียนรู้ของเครื่อง (Machine Learning)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น L1&L2Regularization, Dropout, Data Augmentation, Earlystopping ฯลฯ หนึ่งในแนวคิดที่น่าสนใจเร็วๆ นี้ เพื่อการเรียนรู้เชิงลึก (Deep Learning) คือ Dropout

Dropout จะดำเนินการทั้งในขั้นตอนการฝึกฝนและทดสอบ ในขั้นตอนการฝึกฝนเศษส่วนแบบสุ่ม 'p' ของโหนด และละเว้นการใช้งานในแต่ละชั้นซ่อน แต่ละตัวอย่างฝึกฝน และแต่ละรอบการเรียนรู้ ในขั้นตอนการทดสอบ จะถูกพิจารณาให้เปิดใช้งานทั้งหมด แต่ลดค่าลงด้วยปัจจัย 'p' (factor 'p') เพื่อพิจารณาการเปิดใช้งานที่ขาดหายไปในขั้นตอนการฝึกฝน ดังนั้นเพื่อให้เข้าใจ รูปที่ 2.7 (a) แสดงถึงตัวแบบโครงข่ายประสาทเทียม และรูปที่ 2.7 (b) แสดงถึงตัวแบบโครงข่ายประสาทเทียมหลัง Dropout โดยโหนดที่ถูกทิ้งแทนที่ด้วยสัญลักษณ์ ' X '



(a) Standard Neural Net (b) After Applying Dropout

รูปที่ 2.7 (a) ตัวแบบโครงข่ายประสาทเทียม (b) ตัวแบบโครงข่ายประสาทเทียมหลัง Dropout (https://medium.com/@maksutov.rn/deep-study-of-a-not-very-deep-neural-network-part-5-dropout-and-noise-29d980ece933)

2.4 การประเมินความแม่นยำจากการพยากรณ์

1) ค่าเบี่ยงเบนสัมบูรณ์เฉลี่ย (Mean Absolute Deviation: MAD) โดยที่

$$MAD = \frac{1}{n}\sum_{t=1}^{n}\left|e_t\right|$$

เป็นการวัดความแม่นยำที่วัดจากค่าความคลาดเคลื่อนของการพยากรณ์ โดยไม่ได้คำนึงถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทิศทางของความคลาดเคลื่อน หน่วยของ MAD จะเหมือนกับหน่วยของข้อมูลจริง ค่า MAD ยิ่งน้อย การพยากรณ์ยิ่งแม่นยำ

2) ร้อยละความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Percentage Error: MAPE) โดยที่

$$MAPE = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{e_t}{Y_t}\right|$$

เป็นการวัดความแม่นยำจากค่าความคลาดเคลื่อนของการพยากรณ์เทียบกับค่าจริงของข้อมูลโดยไม่พิจารณาเครื่องหมาย ค่า MAPE เป็นค่าวัดความแม่นยำที่ไม่มีหน่วย จึงมีความเหมาะสมที่ใช้ในการเปรียบเทียบอนุกรมเวลาหลายชุด เมื่อใช้วิธีการพยากรณ์เดียวกันหรือเปรียบเทียบวิธีการพยากรณ์หลายวิธีเมื่อใช้ข้อมูลอนุกรมเวลาชุดเดียวกัน สำหรับเกณฑ์ในการพิจารณาตัวแบบที่พยากรณ์ได้แม่นยำก็จะพิจารณาจากค่า MAPE ที่ต่ำที่สุด ค่า MAPE ยิ่งน้อย การพยากรณ์ยิ่งแม่นยำ

2.5 งานวิจัยที่เกี่ยวข้อง

ณัฐพล รองศรีแย้ม (2542) ได้ทำการศึกษาการพยากรณ์โหลดหม้อแปลงโดยใช้เครือข่ายประสาทเทียม โดยใช้ปริมาณการใช้ไฟฟ้าของผู้ใช้ไฟที่ต่ออยู่กับหม้อแปลงเป็นข้อมูลด้านเข้าป้อนให้แก่เครือข่ายประสาท และได้ผลลัพธ์เป็นค่าพารามิเตอร์แสดงลักษณะโหลดของหม้อแปลงที่สำคัญ 2 ค่าคือ ค่าตัวประกอบโหลด และค่าตัวประกอบกำลังไฟฟ้าขณะหม้อแปลงจ่ายโหลดสูงสุด ซึ่งนำไปใช้ในการคำนวณหาค่าโหลดสูงสุด Utilization factor และค่าโหลดโดยเฉลี่ยของหม้อแปลงต่อไป เครือข่ายประสาทเทียมที่ใช้เป็นเครือข่ายแบบป้อนไปสู่ข้างหน้าที่มีสองชั้น (Two Layer Feed-Forward Neural Network) และใช้การปรับสอนแบบ Modified back-propagation (MBP) ผลจากการทดสอบความแม่นยำในการพยากรณ์พบว่า สามารถนำไปประยุกต์ใช้ในทางปฏิบัติได้จริง

อาทิตย์ อินทวี (2551) ได้ทำการศึกษาการพยากรณ์สึนามิโดยวิธีโครงข่ายประสาทเทียม โดยใช้ข้อมูลจากสถานีวัดน้ำในการพยากรณ์ตำแหน่งของรอยเลื่อน และประมาณค่าการเคลื่อนตัวของรอยเลื่อนด้วยหลักการซ้อนทับ ผลการศึกษาพบว่าความคลาดเคลื่อนในช่วง ±5 นาที ยังไม่มีผลต่อการทำนายตำแหน่งของรอยเลื่อน สำหรับการประมาณค่าการเคลื่อนตัวของรอยเลื่อน ได้ใช้ค่ารากของผลรวมผลต่างกำลังสอง เป็นตัวบ่งชี้ที่ใช้ในการหาระยะเคลื่อนตัว โดยทดสอบกับกรณีศึกษาสำหรับแผ่นดินไหวขนาด 8.3 และ 8.6 ให้ค่าความผิดพลาดระหว่าง 2.8 % ถึง 14.3 % นอกจากนั้นยังพบว่าควรใช้ระยะเวลาเดินทางถึงของคลื่นอย่างน้อย 30 นาที หลังจากคลื่นเดินทางถึง ณ สถานีสุดท้าย ซึ่งจะไม่มีผลต่อการพยากรณ์สึนามิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพชรนรินทร์ แก้วหล้า (2553) ได้ทำการศึกษาระบบพยากรณ์จำนวนผู้ป่วยโรคหัวใจโดยใช้วิธีโครงข่ายประสาทเทียม (Artificial Neural Network) แบบแพร่กระจายกลับ (Back Propagation) ผลการศึกษาพบว่า ระบบที่พัฒนาขึ้นมีประสิทธิภาพอยู่ในระดับดี สามารถนำไปประยุกต์ใช้งานได้อย่างมีประสิทธิภาพ

คงฤทธิ์ โกมาสถิตย์ (2555) ได้ทำการศึกษาการพยากรณ์อุปสงค์พลังงานไฟฟ้าของประเทศไทยในระยะยาวด้วยวิธีโครงข่ายประสาทเทียม โดยได้นำโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ (Back-Propagation Neural Network: BPNN) มาใช้ในการพยากรณ์พลังงานไฟฟ้าของประเทศไทย จากการประเมินประสิทธิภาพโดยร้อยละความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Percentage Error: MAPE) ผลการศึกษาพบว่าสามารถพยากรณ์ได้แม่นยำกว่าวิธีของคณะอนุกรรมการพยากรณ์แห่งประเทศไทย (Thailand Load Forecast Sub-Committee: TLFS)

วชิราภรณ์ แก้วมาตย์ และ สุรชัย จันทร์จรัส (2556) ได้ทำการศึกษาเพื่อหาตัวแบบโครงข่ายประสาทเทียม (Artificial Neural Network) ของดัชนีราคาหลักทรัพย์ของ 9 ประเทศหลัก ได้แก่ ไทย สิงคโปร์ มาเลเซีย อินโดนีเซีย ฟิลิปปินส์ สหรัฐอเมริกา อังกฤษ ญี่ปุ่น และฮ่องกง โดยใช้ข้อมูลราคาปิดรายวันของดัชนีราคาหลักทรัพย์ จากการประเมินประสิทธิภาพโดยร้อยละความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Percentage Error: MAPE) ผลการศึกษาพบว่า วิธีโครงข่ายประสาทเทียม (Artificial Neural Network) แบบแพร่กระจายกลับ (Back Propagation) ให้ผลการพยากรณ์ที่แม่นยำ

ไววิทย์ พานิชอัศดร และ มหศักดิ์ เกตุฉ่ำ (2558) ได้ทำการพยากรณ์ยอดขายสถานีบริการแก๊ส LPG โดยวิธีโครงข่ายประสาทเทียม (Artificial Neural Network) และวิธีการวิเคราะห์อนุกรมเวลา (Time Series) จากการประเมินประสิทธิภาพโดยรากที่สองความคลาดเคลื่อนกำลังสองเฉลี่ย (Root Mean Square Error : RMSE) โดยผลการศึกษาพบว่าวิธีโครงข่ายประสาทเทียมมีประสิทธิภาพดีที่สุด

ผุสดี บุญรอด และ กรวัฒน์ พลเยี่ยม (2558) ได้ทำการศึกษาตัวแบบการพยากรณ์ราคามันสำปะหลัง โดยใช้ข้อมูลจากสมาคมแป้งมันสำปะหลังและสำนักงานเศรษฐกิจการเกษตร ซึ่งประยุกต์ใช้วิธีโครงข่ายประสาทเทียมแบบหลายชั้น (Multi-layer Artificial Neural Network) โดยนำไปเปรียบเทียบกับเคเนียเรสเนเบอร์ (k-Nearest Neighbor, k-NN) และการวิเคราะห์การถดถอยเชิงเส้น (Linear Regression Analysis) จากการประเมินประสิทธิภาพโดยร้อยละความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Percentage Error: MAPE) ผลการศึกษาพบว่า ตัวแบบการพยากรณ์โดยใช้วิธีโครงข่ายประสาทเทียมแบบหลายชั้นมีประสิทธิภาพดีที่สุด

พรรณิภา คุ้มสิน (2560) ได้ทำการศึกษาหาตัวแบบสำหรับการพยากรณ์ปริมาณยอดขายเครื่องปรับอากาศ รุ่น A-1 A-2 และ A-3 โดยใช้วิธีปรับให้เรียบเอ็กซ์โพเนนเชียลแบบโฮลท์และวินเทอร์ (Holt and Winters Exponential Smoothing Method) วิธีบ็อกซ์-เจนกินส์ (Box-Jenkins Method) และวิธีโครงข่ายประสาทเทียม (Artificial Neural Network) จากการประเมิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประสิทธิภาพของตัวแบบพยากรณ์รากที่สองความคลาดเคลื่อนกำลังสองเฉลี่ย (Root Mean Square Error : RMSE) ในการคัดเลือกตัวแบบพยากรณ์ ส่วนร้อยละความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Percentage Error : MAPE) ใช้เพื่อแสดงร้อยละความคลาดเคลื่อนจากค่าจริง ผลการศึกษาพบว่าตัวแบบที่ได้จากวิธีบ็อกซ์-เจนกินส์มีประสิทธิภาพดีที่สุดของการพยากรณ์ยอดขายเครื่องปรับอากาศรุ่น A-1 และ A-2 มากที่สุด สำหรับการพยากรณ์ยอดขายเครื่องปรับอากาศรุ่น A-3 พบว่าตัวแบบการพยากรณ์ที่ได้จากวิธีปรับให้เรียบเอ็กซ์โพเนนเชียลแบบโฮลท์และวินเทอร์มีประสิทธิภาพดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีการดำเนินงานวิจัย

ตามแนวทางของ CRISP-DM ซึ่งขั้นตอนการดำเนินการวิจัย ดังรูปที่ 3.1 ดังนี้

1) การทำความเข้าใจธุรกิจ (Business Understanding)

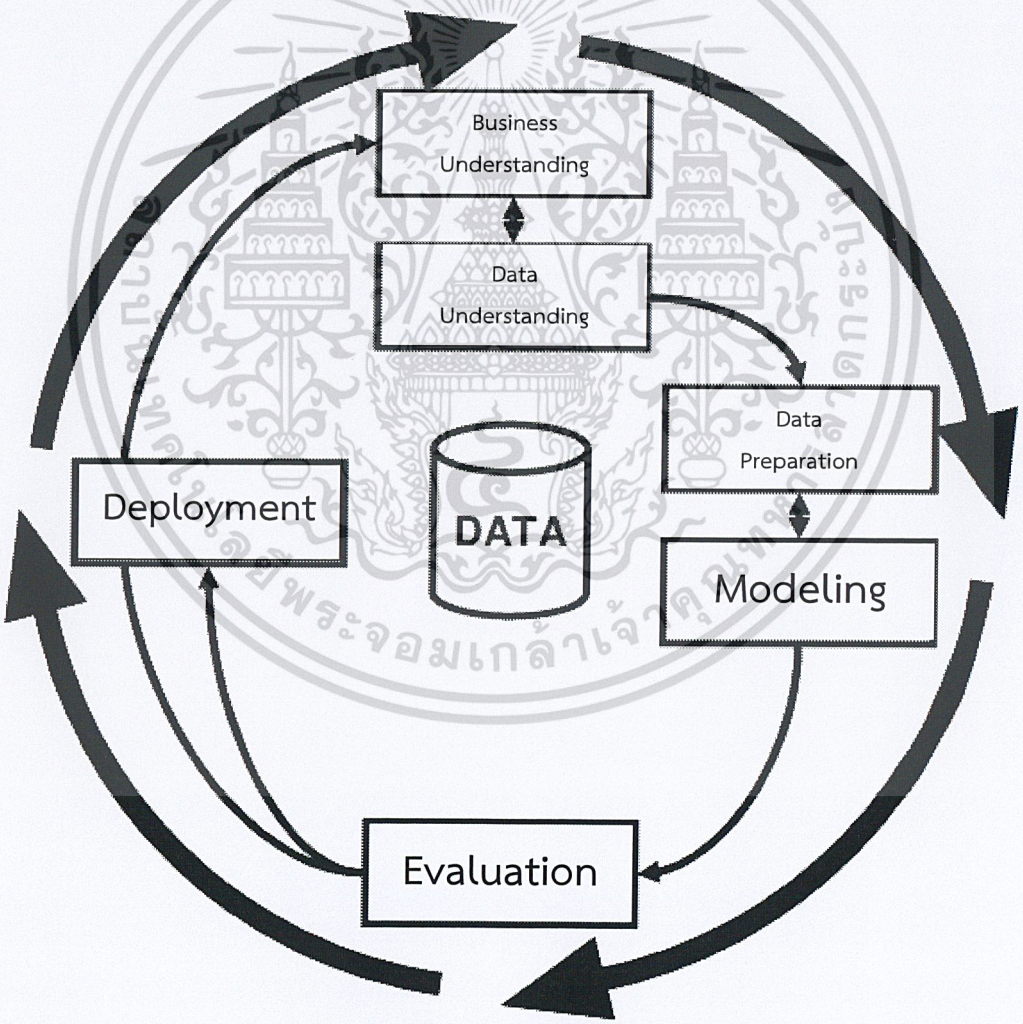
2) การทำความเข้าใจข้อมูล (Data Understanding)

3) การเตรียมข้อมูล (Data Preparation)

4) การสร้างตัวแบบ (Modeling)

5) การประเมินผล (Evaluation)

6) การนำไปใช้ (Deployment)



รูปที่ 3.1 ขั้นตอนCRISP-DM
(https://www.researchgate.net/figure/CRISP-DM-methodology_fig1_258835132)

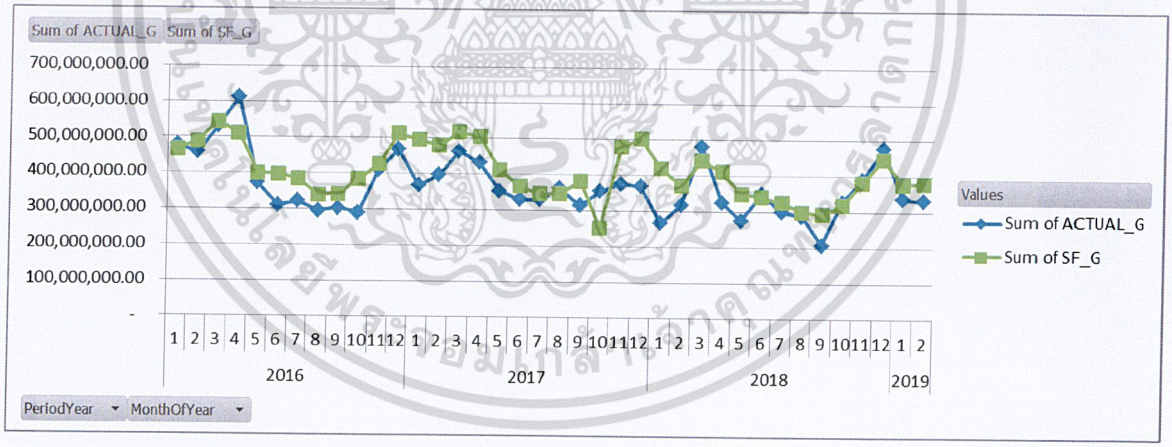
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 การทำความเข้าใจธุรกิจ (Business Understanding)

ปัจจุบันการพยากรณ์ยอดขายสินค้าของบริษัท เกิดจากการประมาณยอดขายในแต่ละเดือนของแต่ละพื้นที่การขาย (Sale Forecast) และแต่ละพื้นที่การขายมีความต้องการสินค้าที่แตกต่างกันออกไป จากตารางที่ 3.1 พบว่าในปี 2559 – 2561 ค่าพยากรณ์ยอดขายสินค้าของบริษัทสูงกว่ายอดขายจริงทุกๆ ปี และจากรูปที่ 3.2 แกน X คือช่วงเวลา ณ ปี 2559 – 2561 แกน Y คือ ปริมาณยอดขาย (Actual) ความคลาดเคลื่อนในบางเดือนที่ค่อนข้างสูงมาก ทำให้ผลิตสินค้าเกินความต้องการจริง ส่งผลให้บริษัทต้องแบกรับสินค้าที่เหลือ อาจจะต้องขายขาดทุนเพื่อให้ขายสินค้าได้หมด ผู้จัดทำจึงต้องการนำโครงข่ายประสาทเทียมมาแก้ปัญหาการพยากรณ์ที่เกิดจากการคาดการณ์ของมนุษย์

ตารางที่ 3.1 ข้อมูลยอดขายปี 2559-2561

ปี	ค่าพยากรณ์ยอดขายสินค้า		คลาดเคลื่อน
	ยอดขายจริง	ค่าพยากรณ์ของทางบริษัท	
2559	5,839,700,100.22	6,139,540,058.82	299,839,958.60
2560	5,413,290,381.57	6,040,494,267.49	627,203,885.92
2561	4,873,786,117.65	5,193,871,680.30	320,085,562.66

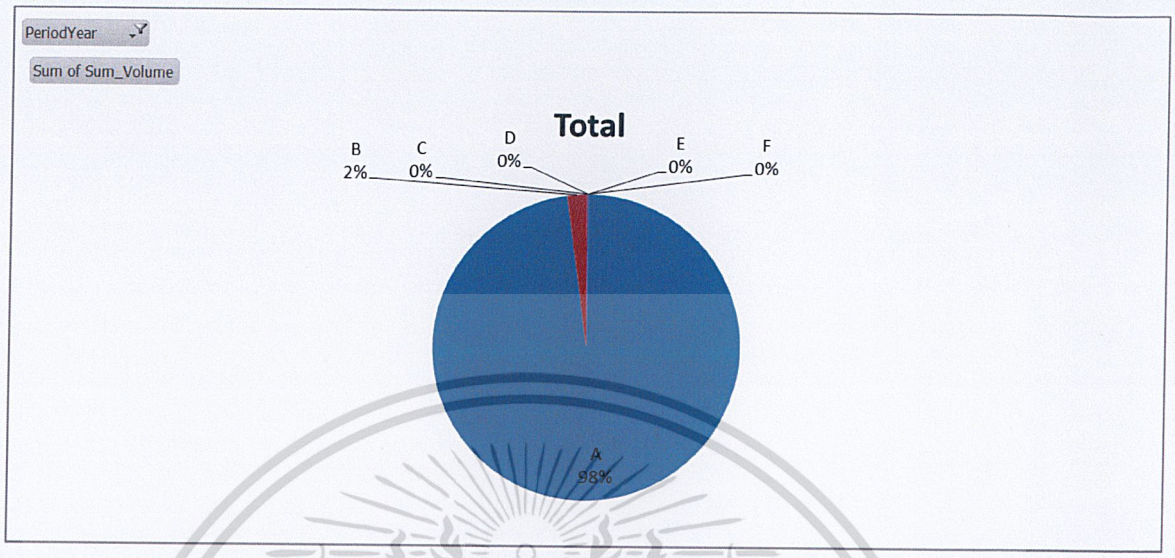


รูปที่ 3.2 กราฟเปรียบเทียบยอดขายจริง (สีน้ำเงิน) กับการประมาณยอดขายสินค้าของแต่ละพื้นที่การขาย (สีเขียว) มกราคม พ.ศ. 2559 ถึง ธันวาคม พ.ศ. 2561

นอกจากนี้บริษัทมีการขายสินค้าหลากหลายประเภท ผู้จัดทำจึงเลือกเฉพาะสินค้าที่มียอดขายมากกว่าร้อยละ 80 จากยอดขายทั้งหมด จากรูปที่ 3.3 พบว่าสินค้า A มียอดขายอยู่ที่ร้อยละ 98 จากยอดขายทั้งหมด และจากตารางที่ 3.2 พบว่าสินค้า A ขนาด AA มียอดขายรวมอยู่ที่ร้อยละ 83.2052

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากยอดขายรวมทั้งหมด ดังนั้นผู้จัดทำจึงเลือกสินค้า A ขนาด AA เพียงอย่างเดียวมาพยากรณ์สินค้าโดยวิธีโครงข่ายประสาทเทียม



รูปที่ 3.3 แผนภูมิแสดงยอดขายแต่ละสินค้า

ตารางที่ 3.2 แสดงร้อยละแต่ละสินค้า โดยแยกสินค้า A ขนาด AA กับสินค้า A ออกจากกัน

Product	Format	Volume	Percent
A	AA	13,495,629,893.13	83.2052
A	Except AA	2,363,976,849.19	14.5747
B	All	328,353,387.13	2.0244
C	All	232,013.99	0.0014
D	All	29,927,151.56	0.1845
E	All	223,943.46	0.0014
F	All	1,344,548.35	0.0083

3.2 การทำความเข้าใจข้อมูล (Data Understanding)

3.2.1 การเก็บข้อมูลขั้นต้น (Collect Initial Data)

เป็นขั้นตอนการคัดลอกข้อมูลจากฐานข้อมูลของบริษัท มาเก็บไว้ยังฐานข้อมูลจำลองเพื่อทำการศึกษาในการพัฒนาตัวแบบ โดยข้อมูลที่คัดลอกจากฐานข้อมูลของบริษัทที่สำคัญ และรายละเอียดของข้อมูล แสดงดังตารางที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

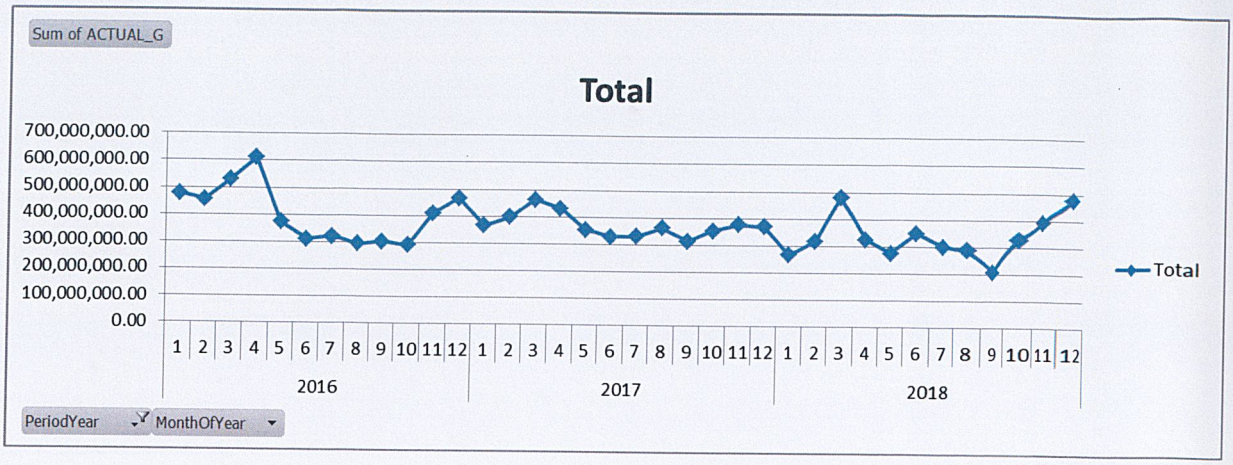
ตารางที่ 3.3 แสดงข้อมูลที่ได้จากฐานข้อมูลของบริษัท และตัวแปรของข้อมูล

ข้อมูล	ตัวแปร
Actual	ปริมาณยอดขาย (ลิตร)
LE	การประมาณยอดขายที่ปรับ Budget (ลิตร)
Branch Code	พื้นที่การขาย
Period Year	ปีที่ทำการขาย
Month of Year	เดือนที่ทำการขาย
Product Brand	ชื่อสินค้า
FG Package Format	ขนาดสินค้า
Amount	ราคาขายรวม (บาท)
Tax Amount	ภาษีรวม (บาท)
Sale Qty	ปริมาณยอดขายรวม (SKU)
Bast Qty	ปริมาณยอดขายรวม (ขวด)
Price Per Unit	ราคาสินค้าต่อขวด (บาท)
Doc Date	วันที่ทำการขาย
Region	ภูมิภาคการขาย
Distribution Channel Group	ช่องทางการกระจายสินค้า

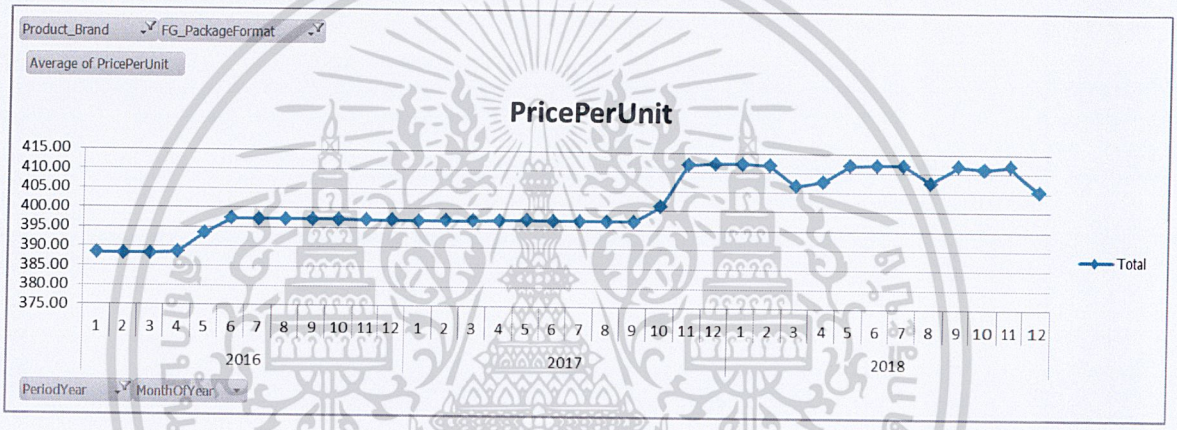
3.2.2 การอธิบายข้อมูล (Describe Data)

เป็นการทำความเข้าใจข้อมูลที่เก็บรวบรวมตามขั้นตอนก่อนหน้า ว่ามีรายละเอียดของข้อมูลเป็นอย่างไร ชนิดของข้อมูลเป็นอย่างไร และมีความสัมพันธ์กับยอดขายจริงหรือไม่ ดังตัวอย่างความสัมพันธ์ระหว่างราคาสินค้าต่อขวดกับยอดขายจริง หากเปรียบเทียบรูปที่ 3.4 กับรูปที่ 3.5 โดยแกน Y ของรูปที่ 3.4 แทนปริมาณยอดขายจริง (Actual), แกน Y ของรูปที่ 3.5 แทนราคาสินค้าต่อขวด (Price Per Unit) และแกน X ของทั้ง 2 รูปคือช่วงเวลา ณ พ.ศ. 2559 ถึง พ.ศ. 2562 จะพบว่ายอดขายจริงไม่มีความสัมพันธ์กับราคาสินค้าต่อขวด เพราะในขณะที่ยอดขายจริงมีกราฟเส้นที่มีลักษณะขึ้นๆ ลงๆ แต่กราฟราคาสินค้าต่อขวด ส่วนใหญ่กลับเป็นเส้นตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงยอดขายจริง (Actual) ตั้งแต่ปี 2559 ถึงปี2561

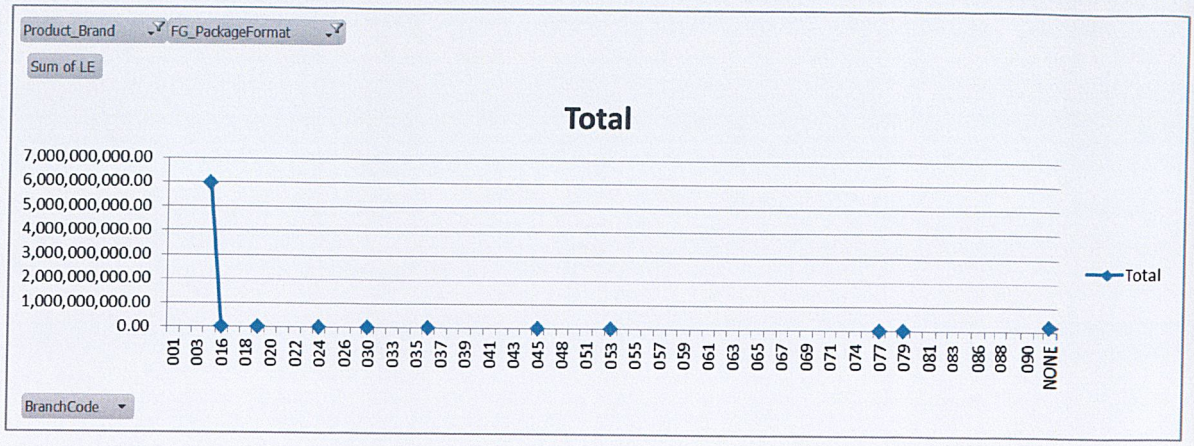


รูปที่ 3.5 แสดงราคาสินค้าต่อขวด (Price Per Unit) ในช่วงระยะเวลาปี 2559 ถึงปี 2561

3.2.3 การตรวจสอบคุณภาพของข้อมูล (Verify Data Quality)

เป็นการตรวจสอบความสมบูรณ์ของข้อมูล ซึ่งพบว่าข้อมูลการประมาณยอดขายที่ปรับ Budget (LE) ทางฝั่งการกระจายสินค้าสมัยใหม่ (Modern Trade) มีข้อมูลสูญหาย (Missing Value) จำนวนมากในหลายพื้นที่การขาย (Branch Code) ดังรูปที่ 3.6 ทางผู้จัดทำจึงทำการปรึกษากับตัวแทนจากทางบริษัท จึงได้ข้อสรุปว่าจะทำตัวแบบการพยากรณ์ยอดขายสินค้าเฉพาะการกระจายสินค้าแบบดั้งเดิม (Traditional Trade)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดงข้อมูลการประมาณยอดขายที่ปรับ Budget (LE) ทางฝั่งการกระจายสินค้าสมัยใหม่ (Modern Trade)

จากรูปที่ 3.6 แกน X คือ พื้นที่การขาย (Branch Code) แกน Y คือ ปริมาณยอดขายจริง (Actual) จะพบว่าข้อมูลการประมาณยอดขายที่ปรับ Budget (LE) ทางฝั่งการกระจายสินค้าสมัยใหม่ (Modern Trade) มีข้อมูลแค่ 11 พื้นที่การขาย (Branch Code) จาก 83 พื้นที่การขาย (Branch Code) เท่านั้น ทางผู้จัดทำจึงตัดสินใจว่าฝั่งการกระจายสินค้าสมัยใหม่ (Modern Trade) ไม่มีคุณภาพมากพอที่จะนำไปใช้ในการวิเคราะห์ต่อได้

3.3 การเตรียมข้อมูล (Data Preparation)

เป็นขั้นตอนการนำข้อมูลที่ผ่านขั้นตอนการทำความเข้าใจข้อมูล (Data Understanding) มาทำการเตรียมข้อมูล (Data Preparation) ใน Jupyter Notebook โดยภาษา Python เพื่อให้พร้อมสำหรับการสร้างตัวแบบ (Modeling) โดยมีขั้นตอนดังนี้

3.3.1 การคัดเลือกข้อมูล (Select Data)

เป็นการคัดเลือกข้อมูลที่มีความสัมพันธ์กับข้อมูลยอดขายจริง โดยผ่านกระบวนการทำความเข้าใจข้อมูล (Data Understanding) มาแล้ว ซึ่งในขั้นตอนนี้ได้ทำการเลือกข้อมูลดังนี้

1) ปริมาณยอดขายจริง (Actual) เนื่องจากเป็นตัวแปรตาม

2) การประมาณยอดขายที่ปรับ Budget (LE) เนื่องจากมีรูปแบบที่คล้ายคลึงกันมาก กับปริมาณยอดขายจริง (Actual)

3) พื้นที่ขาย (Branch Code) เป็นตัวแปรที่ทางบริษัทต้องการให้พยากรณ์ในระดับพื้นที่ขาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ปีที่ทำการขาย (Period Year) เป็นตัวแปรที่จำเป็นในการพยากรณ์ ณ ช่วงเวลาใดเวลาหนึ่ง

5) เดือนที่ทำการขาย (Month of Year) เป็นตัวแปรที่จำเป็นในการพยากรณ์ ณ ช่วงเวลาใดเวลาหนึ่ง

รายละเอียดของข้อมูลที่นำมาใช้ในการสร้างตัวแบบหลังคัดเลือกข้อมูล ดังตารางที่ 3.4

ตารางที่ 3.4 แสดงตัวแปรของข้อมูลที่นำมาใช้ในการสร้างตัวแบบ

ข้อมูล	ตัวแปร
Actual	ปริมาณยอดขาย (ลิตร)
LE	การปริมาณยอดขายที่ปรับ Budget (ลิตร)
Branch Code	พื้นที่การขาย
Period Year	ปีที่ทำการขาย
Mounth of Year	เดือนที่ทำการขาย

โดยข้อมูลที่ไม่ได้คัดเลือกนั้นเป็นเพราะทางผู้จัดทำมีความคิดเห็นว่าไม่มีความสัมพันธ์กับปริมาณยอดขายจริง (Actual) หรือไม่มีความจำเป็นที่ต้องนำมาใช้ในการสร้างตัวแบบการพยากรณ์ยอดขายสินค้า โดยมีข้อมูลที่ไม่ได้คัดเลือกดังนี้

1) ชื่อสินค้า (Product Brand) โดยค่าของข้อมูลมีเพียงค่าเดียวคือ สินค้า A ซึ่งได้อธิบายเหตุผลไปแล้วในหัวข้อการทำความเข้าใจธุรกิจ (Business Understanding)

2) ขนาดสินค้า (FG Package Format) โดยค่าของข้อมูลมีเพียงค่าเดียวคือ ขนาด AA ซึ่งได้อธิบายเหตุผลไปแล้วในหัวข้อการทำความเข้าใจธุรกิจ (Business Understanding)

3) ภูมิการขาย (Region) เนื่องจากต้องการพยากรณ์ยอดขายสินค้าในระดับพื้นที่การขาย (Branch Code)

4) ราคาขายรวม (Amount) เนื่องจากเป็นข้อมูลประเภทตัวแปรตามเช่นเดียวกันกับปริมาณยอดขายจริง (Actual)

5) ภาษีรวม (Tax Amount) เนื่องจากเป็นข้อมูลประเภทตัวแปรตามเช่นเดียวกันกับปริมาณยอดขายจริง (Actual)

6) ปริมาณยอดขายรวม (SKU) (Sale Qty) เนื่องจากเป็นข้อมูลประเภทตัวแปรตามเช่นเดียวกันกับปริมาณยอดขายจริง (Actual)

7) ปริมาณยอดขายรวม (ขวด) (Base Qty) เนื่องจากเป็นข้อมูลประเภทตัวแปรตามเช่นเดียวกันกับปริมาณยอดขายจริง (Actual)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8) ราคาสินค้าต่อขวด (Price Per Unit) ซึ่งได้อธิบายเหตุผลที่ไม่เลือกไปแล้วในหัวข้อการอธิบายข้อมูล (Describe Data)

9) วันที่ขาย (Doc Date) เนื่องจากต้องการพยากรณ์ยอดขายสินค้าในระดับรายเดือน จึงไม่จำเป็นต้องใช้ข้อมูลยอดขายรายวัน

10) ช่องทางการกระจายสินค้า (Distribution Channel Group) โดยค่าของข้อมูลมีเพียงค่าเดียวคือ การกระจายสินค้าแบบดั้งเดิม (Traditional Trade) ซึ่งได้อธิบายเหตุผลไปแล้วในหัวข้อการตรวจสอบคุณภาพของข้อมูล (Verify Data Quality)

โดยมีคำสั่งดังตารางที่ 3.5

ตารางที่ 3.5 แสดง Python Code สำหรับการคัดเลือกข้อมูล (Select Data)

```
df_Product=df_Product[~((df_Product.PeriodYear==2019)&
(df_Product.MonthOfYear >=3))]
df_Product=df_Product[(df_Product['Product_Brand']=='A')&
(df_Product['FG_PackageFormat']=='AA')]
df_Product=df_Product.drop(columns=['Product_Brand','FG_PackageFormat',
'Region','Amount','TaxAmount','SaleQty','BaseQty','PricePerUnit','DocDate',
DistributionChannel_Group])
```

จากตารางที่ 3.5 เป็นการแสดง Python Code สำหรับการคัดเลือกข้อมูล (Select Data) โดยการคัดเลือกเฉพาะข้อมูลปี พ.ศ. 2562 เดือนมกราคมและกุมภาพันธ์เท่านั้น นอกจากนั้นข้อมูลปี พ.ศ. 2562 เดือนอื่นๆ จะทำการคัดออกเนื่องจากบริษัทยังมีข้อมูลยอดขายจริง (Actual) ไม่เพียงพอที่จะนำมาใช้ในการวิเคราะห์ได้ โดยเลือกที่จะพยากรณ์เฉพาะสินค้า A ขนาด AA เท่านั้น และทำการเอาข้อมูลที่ไม่จำเป็นสำหรับการวิเคราะห์ออก

3.3.2 การทำความสะอาดข้อมูล (Data Cleaning)

หลังจากทำการคัดเลือกข้อมูลเสร็จแล้ว ต่อไปจะเป็นขั้นตอนจัดการกับพื้นที่การขาย (Branch Code) ที่มีข้อมูลค่าสูญหาย (Missing Value) เนื่องจากข้อมูลสูญหาย (Missing Value) หากนำเข้าไปวิเคราะห์ในการสร้างตัวแบบแล้วนั้น จะนำมาซึ่งผลลัพธ์ที่ผิดพลาด ซึ่งขั้นตอนการทำความสะอาดข้อมูล (Data Cleaning) ตามขั้นตอนดังนี้

1) กรองให้เหลือเฉพาะพื้นที่การขาย (Branch Code) ที่มีข้อมูลมากกว่าร้อยละ 80 ของข้อมูลทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) แทนที่ค่าสูญหาย (Missing Value) ด้วยค่าเฉลี่ย (Mean)

โดยมีคำสั่งดังตารางที่ 3.6

ตารางที่ 3.6 แสดง Python Code สำหรับการทำความสะอาดข้อมูล (Data Cleaning)

```
df_ProductAC=df_Product.dropna(subset=['ACTUAL'])
df_ProductAC=df_ProductAC.pivot_table(index=['PeriodYear','MonthOfYear']
,columns=['BranchCode'],values='ACTUAL',aggfunc=np.sum)
df_ProductAC=df_ProductAC[df_ProductAC.columns[df_ProductAC.isin(['NaN'])
.sum()/len(df_ProductAC)*100<20]]
df_ProductAC=df_ProductAC.fillna(df_ProductAC.mean()[:])
df_ProductAC=df_ProductAC.reset_index()
df_ProductAC=pd.melt(df_ProductAC,id_vars= ['PeriodYear','MonthOfYear'])
df_ProductAC.rename(columns={'value':'ACTUAL'},inplace=True)

df_ProductLE=df_Product.dropna(subset=['LE'])
df_ProductLE=df_ProductLE.pivot_table(index=['PeriodYear','MonthOfYear']
,columns=['BranchCode'],values='LE',aggfunc=np.sum)
df_ProductLE=df_ProductLE.fillna(df_ProductLE.mean()[:])
df_ProductLE=df_ProductLE.reset_index()
df_ProductLE=pd.melt(df_ProductLE,id_vars=['PeriodYear','MonthOfYear'])
df_ProductLE.rename(columns={'value':'LE'},inplace=True)
```

จากตารางที่ 3.6 เป็นการแสดง Python Code สำหรับการทำความสะอาดข้อมูล (Data Cleaning) โดยขั้นตอนนี้จะเป็นการทำความสะอาดข้อมูล (Data Cleaning) ยอดขายจริง (Actual) โดยเอาพื้นที่การขาย (Branch Code) ที่มีค่าสูญหาย (Missing Value) เกินร้อยละ 20 ออก โดยพื้นที่การขาย (Branch Code) ที่ไม่ได้เอาออก จะแทนที่ค่าสูญหาย (Missing Value) ด้วยค่าเฉลี่ย (Mean) ส่วนข้อมูลการประมาณยอดขายสินค้าโดยปรับ Budget (LE) ก็จะทำการแทนที่ค่าสูญหาย (Missing Value) ด้วยค่าเฉลี่ย (Mean) เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 การสร้างข้อมูลใหม่ (Construct Data)

หลังจากจัดการกับค่าสูญหาย (Missing Value) เสร็จแล้ว ขั้นตอนนี้จะเป็นการสร้างข้อมูลขึ้นมาใหม่ เนื่องจากรูปที่ 3.7 จะพบว่าในแต่ละปีมีรูปแบบที่คล้ายคลึงกัน ผู้จัดทำจึงเพิ่มข้อมูลยอดขายย้อนหลัง 1 ปี (Last Year) เพื่อเพิ่มประสิทธิภาพในการวิเคราะห์ของตัวแบบให้ดียิ่งขึ้น โดยมีคำสั่งดังตารางที่ 3.7

รูปที่ 3.7 แสดงข้อมูลยอดขายจริง (Actual) ในแต่ละปี

จากรูปที่ 3.7 แกน X คือ เดือนที่ทำการขาย (Month of Year) แกน Y คือ ปริมาณยอดขายจริง (Actual) จะพบว่าในปี 2559 -2561 ในแต่ละปีจะมียอดขายจริง (Actual) เพิ่มสูงขึ้นในเดือนที่ 3 และเดือนที่ 12 เนื่องด้วยวันที่ 13 – 15 เมษายน เป็นวันสงกรานต์ และวันที่ 1 มกราคม เป็นวันปีใหม่ จึงมีการซื้อสินค้ากักเก็บไว้ เพื่อให้เพียงพอในช่วงเทศกาล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.7 แสดง Python Code สำหรับการสร้างข้อมูลใหม่ (Construct Data)

```
df_Time=df_ProductAC.copy()
df_Time['NextYear']=df_Time.PeriodYear.replace([2015,2016,2017,2018,2019],
[2016,2017,2018,2019,2020])
df_Time.drop(['PeriodYear'],axis=1,inplace=True)
df_Time=df_Time.rename(columns={'ACTUAL':'RENAME'})
df=pd.merge(df,df_Time,how='left',left_on=['PeriodYear','MonthOfYear',
'BranchCode'],right_on = ['NextYear', 'MonthOfYear','BranchCode'])
df=df[['PeriodYear','MonthOfYear','BranchCode','ACTUAL','LE','SF_Volumn',
'RENAME']]
df=df.rename(columns={'RENAME':'last year'})
df=df[~(df['PeriodYear']==2015)].reset_index(drop=True)
```

จากตารางที่ 3.7 เป็นการแสดง Python Code สำหรับการสร้างข้อมูลใหม่ (Construct Data) เป็นขั้นตอนการสร้างข้อมูลย้อนหลัง 1 ปี (Last Year) โดยอ้างอิงจากข้อมูลยอดขายจริง (Actual) เช่น ข้อมูลยอดขายเดือนมกราคม ปี 2561 ยอดขายย้อนหลัง 1 ปี (Last Year) ก็จะเป็นข้อมูลยอดขายเดือนมกราคม ปี 2560

3.3.4 การแบ่งข้อมูล (Dividing Data)

ได้แบ่งข้อมูลออกเป็น 3 ส่วน ดังนี้

1) ชุดข้อมูลฝึกฝน (Training Set) ร้อยละ 70 ของข้อมูลเดือน มกราคม พ.ศ. 2559 ถึง ธันวาคม พ.ศ. 2561 สำหรับสร้างตัวแบบ

2) ชุดข้อมูลทดสอบ (Test Set) ร้อยละ 30 ของข้อมูลเดือน มกราคม พ.ศ. 2559 ถึง ธันวาคม พ.ศ. 2561 สำหรับประเมินตัวแบบ

3) ชุดข้อมูลทวนสอบ (Validation Set) ของข้อมูลเดือน มกราคม พ.ศ. 2562 ถึง กุมภาพันธ์ สำหรับเพิ่มความเชื่อมั่นในการนำไปใช้

โดยมีคำสั่งดังตารางที่ 3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.8 แสดง Python Code สำหรับการแบ่งข้อมูล (Dividing Data)

```
df_2019=df[(df.PeriodYear==2019)]
df=df[~(df.PeriodYear==2019)]
df_train=df.sample(frac=0.7)
df_test=df[~df.index.isin(df_train.index)]
```

จากตารางที่ 3.8 เป็นการแสดง Python Code สำหรับการแบ่งข้อมูล (Dividing Data) โดยทำการแบ่งปี พ.ศ. 2562 ออกมาก่อน เพื่อทำชุดข้อมูลทวนสอบ แล้วจึงทำการแบ่งข้อมูลที่เหลือออกเป็นชุดข้อมูลฝึกฝนร้อยละ 70 และชุดข้อมูลทดสอบร้อยละ 30

3.3.5 การแปลงข้อมูล (Data Transformations)

หลังจากทำการเตรียมข้อมูลตามขั้นตอนก่อนหน้าเสร็จหมดแล้ว ขั้นตอนนี้จะเป็นขั้นตอนการแปลงข้อมูลต่างๆ เพื่อให้เหมาะสมกับอัลกอริทึม (Algorithm) ที่จะนำไปใช้สร้างตัวแบบ โดยทางผู้จัดทำได้แบ่งข้อมูลออกเป็น 2 ประเภท เพื่อให้เหมาะสมกับการแปลงในแต่ละประเภทดังนี้

1) ข้อมูลเชิงคุณภาพ (Qualitative Data) คือข้อมูลที่วัดออกมาเป็นคุณลักษณะของข้อมูลนั้นไม่สามารถวัดออกมาเป็นตัวเลขได้ โดยจะทำการแปลงข้อมูลเหล่านี้ให้เป็นตัวแปรหุ่น (Dummy Variable) ที่มีค่าแค่ 0 กับ 1 เช่น พื้นที่ขาย (Branch Code) โดยมีคำสั่งดังตารางที่ 3.9

2) ข้อมูลเชิงปริมาณ (Quantitative Data) คือข้อมูลที่วัดออกมาเป็นตัวเลขได้ โดยจะทำการแปลงข้อมูลเหล่านี้ให้อยู่ในช่วง 0 ถึง 1 ด้วยวิธี Min-Max Normalization สามารถคำนวณได้ตามสมการที่ 3.1 โดยมีคำสั่งดังตารางที่ 3.10 และตารางที่ 3.11

$$X(\text{Norm}) = \frac{X - X(\text{Min})}{X(\text{Max}) - X(\text{Min})} \qquad (3.1)$$

โดยที่ $X(\text{Norm})$ คือ ค่าหลังจากที่ทำการแปลงข้อมูลแล้ว

$X(\text{Min})$ คือ ค่าที่ต่ำที่สุดในข้อมูลชุดฝึกฝน

$X(\text{Max})$ คือ ค่าที่สูงที่สุดในข้อมูลชุดฝึกฝน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.9 แสดง Python Code สำหรับการแปลงข้อมูลคุณภาพ (Data Tranformations)

```
dummiesBranchCode=pd.get_dummies(df['BranchCode'])
df=pd.concat([df,dummiesBranchCode],axis='columns')
X_train.drop(['BranchCode'],axis=1,inplace=True)
X_test.drop(['BranchCode'],axis=1,inplace=True)
X_2019.drop(['BranchCode'],axis=1,inplace=True)
```

จากตารางที่ 3.9 เป็นการแสดง Python Code สำหรับการแปลงข้อมูลคุณภาพ (Data Tranformations) ด้วยวิธีการแปลงให้เป็นตัวแปรหุ่น (Dummy Variable) โดยทำการแปลงข้อมูลพื้นที่การขาย (Branch Code) และทำการเอาข้อมูลที่ไม่ได้แปลงออกเพื่อลดความซ้ำซ้อนของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.10 แสดง Python Code สำหรับการแปลงข้อมูลปริมาณ (Data Tranformations)

```
Y_train['ACTUAL_NORM']=(Y_train.ACTUAL-
Y_train.ACTUAL.min())/(Y_train.ACTUAL.max()-Y_train.ACTUAL.min())
Y_test['ACTUAL_NORM']=(Y_test.ACTUAL-
Y_train.ACTUAL.min())/(Y_train.ACTUAL.max()-Y_train.ACTUAL.min())
Y_train.drop(['ACTUAL'],axis=1,inplace=True)
Y_test.drop(['ACTUAL'],axis=1,inplace=True)

X_train['PeriodYear_NORM']=(X_train.PeriodYear–
X_train.PeriodYear.min())/(X_train.PeriodYear.max()-X_train.PeriodYear.min())
X_test['PeriodYear_NORM']=(X_test.PeriodYear–
X_train.PeriodYear.min())/(X_train.PeriodYear.max()-X_train.PeriodYear.min())
X_2019['PeriodYear_NORM']=(X_2019.PeriodYear–
X_train.PeriodYear.min())/(X_train.PeriodYear.max()-X_train.PeriodYear.min())
X_train.drop(['PeriodYear'],axis=1,inplace=True)
X_test.drop(['PeriodYear'],axis=1,inplace=True)
X_2019.drop(['PeriodYear'],axis=1,inplace=True)

X_train['MonthOfYear_NORM']=(X_train.MonthOfYear–
X_train.MonthOfYear.min())/(X_train.MonthOfYear.max()-X_train.MonthOfYear.min())
X_test['MonthOfYear_NORM']=(X_test.MonthOfYear–
X_train.MonthOfYear.min())/(X_train.MonthOfYear.max()-X_train.MonthOfYear.min())
X_2019['MonthOfYear_NORM']=(X_2019.MonthOfYear–
X_train.MonthOfYear.min())/(X_train.MonthOfYear.max()-X_train.MonthOfYear.min())
X_train.drop(['MonthOfYear'],axis=1,inplace=True)
X_test.drop(['MonthOfYear'],axis=1,inplace=True)
X_2019.drop(['MonthOfYear'],axis=1,inplace=True)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.10 แสดง Python Code สำหรับการแปลงข้อมูลปริมาณ (Data Tranformations) (ต่อ)

```
X_train['LE_NORM']=(X_train.LE–X_train.LE.min())/(X_train.LE.max()-
X_train.LE.min())
X_test['LE_NORM']=(X_test.LE–X_train.LE.min())/(X_train.LE.max()-X_train.LE.min())
X_2019['LE_NORM']=(X_2019.LE–X_train.LE.min())/(X_train.LE.max()-
X_train.LE.min())
X_train.drop(['LE'],axis=1,inplace=True)
X_test.drop(['LE'],axis=1,inplace=True)
X_2019.drop(['LE'],axis=1,inplace=True)

X_train['last year_NORM']=(X_train.LastYear–
X_train.LastYear.min())/(X_train.LastYear.max()-X_train.LastYear.min())
X_test['last year_NORM' ]=(X_test.LastYear–
X_train.LastYear.min())/(X_train.LastYear.max()-X_train.LastYear.min())
X_2019['last year_NORM' ]=(X_2019.LastYear –
X_train.LastYear.min())/(X_train.LastYear.max()-X_train.LastYear.min())
X_train.drop(['last year'],axis=1,inplace=True)
X_test.drop(['last year'],axis=1,inplace=True)
X_2019.drop(['last year'],axis=1,inplace=True)
```

จากตารางที่ 3.10 และตารางที่ 3.11 เป็นการแสดง Python Code สำหรับการแปลงข้อมูลปริมาณ (Data Tranformations) ด้วยวิธี Min-Max Normalization โดยทำการแปลงข้อมูลยอดขาย (Actual), ปีที่ทำการขาย (Period Year), เดือนที่ทำการขาย (Month of Year), การประมาณยอดขายโดยปรับ Budget (LE) และยอดขายย้อนหลัง 1 ปี (Last Year) และทำการเอาข้อมูลที่ไม่ได้แปลงออกเพื่อลดความซ้ำซ้อนของข้อมูล

3.4 การสร้างตัวแบบ (Modeling)

หลังจากการเตรียมข้อมูล (Data Preparation) จะเป็นขั้นตอนการนำข้อมูลมาสร้างตัวแบบ (Modeling) เพื่อนำมาพยากรณ์ยอดขายสินค้า โดยใช้ไลบรารี Keras ซึ่ง Keras เป็นหนึ่งในไลบรารีที่นิยมในการทำการเรียนรู้เชิงลึก (Deep Learning) โดยผู้จัดทำได้กำหนดค่าไฮเปอร์พารามิเตอร์หรือกระบวนการในการสร้างตัวแบบดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.11 แสดงการกำหนดค่าไฮเปอร์พารามิเตอร์หรือกระบวนการในการสร้างตัวแบบในแต่ละข้อกำหนด

ข้อกำหนดต่างๆในการสร้างตัวแบบ	ค่าไฮเปอร์พารามิเตอร์
อัลกอริทึม (Algorithm)	Neural Network
จำนวนโหนดในชั้นอินพุต (Input Node)	87
จำนวนชั้นซ่อน (Hidden Layer)	1
จำนวนโหนดในชั้นซ่อน (Hidden Node)	22, 44, 87
จำนวนโหนดในชั้นเอาท์พุต (Output Node)	1
Drop Out	0.1
ฟังก์ชันกระตุ้น (Activation Function)	Sigmoid Function, Relu Function
ฟังก์ชันสูญเสีย (Loss Function)	Mean Square Error (MSE)
Optimizer	SGD, Adam
โมเมนตัม (Momentum)	0.5, 0.7, 0.9
อัตราการเรียนรู้ (Learning Rate)	0.001 ,0.002, 0.004 ,0.006, 0.008
β_1	0.9
β_2	0.999
จำนวนรอบในการเรียนรู้ (Epoch)	200

จากตารางที่ 3.12 แสดงการกำหนดค่าไฮเปอร์พารามิเตอร์สำหรับการสร้างตัวแบบ โดยผู้จัดทำได้กำหนดอัลกอริทึม (Algorithm) ที่ใช้ในการสร้างตัวแบบการพยากรณ์ยอดขายสินค้า คือ โครงข่ายประสาทเทียม (Neural Network) เนื่องจากโครงข่ายประสาทเทียม (Neural Network) สามารถเรียนรู้ชุดข้อมูลที่ซับซ้อนมากๆ ได้ ซึ่งอาจแสดงความสัมพันธ์บางอย่างของอินพุต (Input) ที่คนมองข้าม โดยได้กำหนดเงื่อนไขต่างๆ ดังนี้

จำนวนโหนดในชั้นอินพุต (Input Node) กำหนดตามข้อมูลที่นำเข้า คือการประมาณยอดขายที่ปรับ Budget (LE), ปีที่ทำการขาย (Period Year), เดือนที่ทำการขาย (Month of Year), ยอดขายย้อนหลัง 1 ปี (Last Year) และพื้นที่การขาย (Branch Code) ทั้งหมด 83 พื้นที่ หลังจากการแปลงข้อมูล (Format Data) จะมีข้อมูลนำเข้าทั้งหมด 87 ตัวแปร

จำนวนชั้นซ่อน (Hidden Layer) เนื่องจากผู้จัดทำไม่ต้องการให้การเรียนรู้ของเครื่องจักร (Machine Learning) เรียนรู้ข้อมูลมากเกินไปจนทำให้เกิดการ Overfitting จึงกำหนดจำนวนชั้นซ่อน (Hidden Layer) เพียงชั้นเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนโหนดในชั้นซ่อน (Hidden Node) จากที่ผู้จัดทำได้ทำการศึกษางานวิจัยของผู้อื่น ผู้จัดทำพบว่าไม่มีข้อกำหนดที่แน่นอนในการกำหนดจำนวนโหนดในชั้นซ่อน (Hidden Node) ต้องลองผิดลองถูกถึงจะทราบจำนวนโหนดในชั้นซ่อน (Hidden Layer) ที่เหมาะสมที่สุดกับข้อมูล

จำนวนโหนดในชั้นเอาท์พุต (Output Node) กำหนดจากผลลัพธ์ที่ได้โดยในการศึกษานี้ คือ ข้อมูลยอดขายจริง (Actual)

Dropout ผู้จัดทำได้กำหนดเป็น 0.1 เนื่องจากไม่ต้องการให้เกิดการถูกทิ้งมากเกินไป จนทำให้ตัวแบบเรียนรู้ข้อมูลในแต่ละรอบน้อยเกินไป

ฟังก์ชันกระตุ้น (Activation Function) ผู้จัดทำได้กำหนดเป็น Sigmoid Function กับ Relu Function ซึ่งพบว่ามีรายงานวิจัยส่วนใหญ่ใช้ฟังก์ชันกระตุ้น (Activation Funtion) นี้

ฟังก์ชันสูญเสีย (Loss Function) ผู้จัดทำได้กำหนดเป็น MSE (Mean Square Error) ซึ่งพบว่ามีรายงานวิจัยส่วนใหญ่ใช้ฟังก์ชันสูญเสีย (Loss Funtion) นี้

Optimizer ผู้จัดทำได้กำหนดเป็น SGD กับ Adam ซึ่งพบว่ามีรายงานวิจัยส่วนใหญ่ใช้ Optimizer นี้ โดย Adam มีไฮเปอร์พารามิเตอร์ที่ต้องกำหนดดังนี้ อัตราการเรียนรู้ (Learning Rate), β_1, β_2, SGD มีไฮเปอร์พารามิเตอร์ที่ต้องกำหนดดังนี้ อัตราการเรียนรู้ (Learning Rate), โมเมนตัม (Momentum) และจำนวนรอบในการเรียนรู้ (Epoch) ดังตัวอย่างเช่น ตารางที่ 3.13 ได้กำหนด Optimizer คือ Adam, อัตราการเรียนรู้ (Learning Rate) เท่ากับ 0.006, β_1 เท่ากับ 0.9 และ β_2 เท่ากับ 0.999

โมเมนตัม (Momentum) จากที่ผู้จัดทำได้ทำการศึกษางานวิจัย พบว่าไม่มีข้อกำหนดที่แน่นอนในการกำหนดค่าโมเมนตัม (Momentum) ต้องลองผิดลองถูกถึงจะทราบค่าโมเมนตัม (Momentum) ที่เหมาะสมที่สุดกับข้อมูล ซึ่งในงานวิจัยนี้พบว่าได้ค่าโมเมนตัม (Momentum) เท่ากับ 0.9

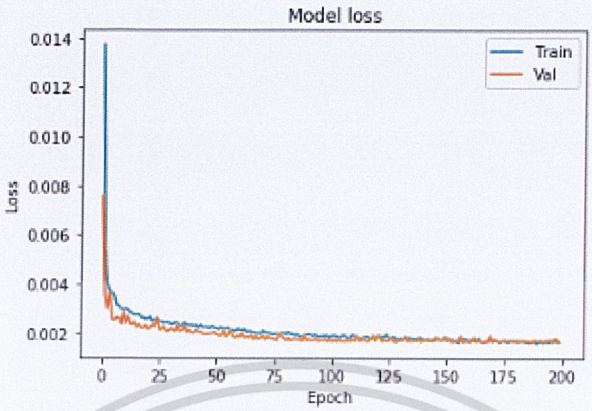
อัตราการเรียนรู้ (Learning Rate) จากที่ผู้จัดทำได้ทำการศึกษางานวิจัย พบว่าไม่มีข้อกำหนดที่แน่นอนในการกำหนดค่าอัตราการเรียนรู้ (Learning Rate) ต้องลองผิดลองถูกถึงจะทราบค่าอัตราการเรียนรู้ (Learning Rate) ที่เหมาะสมที่สุดกับข้อมูล ซึ่งในงานวิจัยนี้พบว่าได้ค่าอัตราการเรียนรู้ (Learning Rate) เท่ากับ 0.006

β_1 และ β_2 ผู้จัดทำได้กำหนดเป็น 0.9 และ 0.999 ตามลำดับ เนื่องจากค่า β_1 และ β_2 ควรมีค่าใกล้ 1 ทางผู้จัดทำจึงไม่ได้ปรับค่าใดๆ

และจำนวนรอบในการเรียนรู้ (Epoch) จากรูปที่ 3.8 เป็นตัวอย่างการกำหนดจำนวนรอบในการเรียนรู้ (Epoch) ที่เหมาะสม สามารถประมาณได้จากเส้นสีฟ้า (ชุดฝึกฝน) และสีเหลือง (ชุดทดสอบ) ซึ่ง 2 เส้นนี้ควรจะทับกัน และควรมีค่าของฟังก์ชันสูญเสีย (Loss Function) น้อยที่สุด โดยผู้จัดทำได้ทำการทดลองหลายครั้ง และได้พบว่าจำนวนรอบในการเรียนรู้ (Epoch) ที่เหมาะสมส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใหญ่ในระหว่างทดลองคือ 200 โดยคำสั่งที่เอาไว้ตรวจสอบรอบในการเรียนรู้ (Epoch) ที่เหมาะสม แสดงได้ดังตารางที่ 3.13



รูปที่ 3.8 แสดงจำนวนรอบในการเรียนรู้ที่เหมาะสม (Epoch)

รูปที่ 3.8 แกน X คือ จำนวนรอบในการเรียนรู้ (Epoch) แกน Y คือ เป็นค่าของฟังก์ชันสูญเสีย (Loss Function) แสดงให้เห็นว่าตัวแบบเรียนรู้ชุดข้อมูลฝึกฝน (สีฟ้า) และชุดข้อมูลทดสอบ (สีเหลือง) มีเส้นที่แสดงค่าของฟังก์ชันสูญเสีย (Loss Function) ที่ใกล้เคียงกันจนทับกันในช่วง 150 - 200

ตารางที่ 3.12 แสดง Python Code สำหรับตรวจสอบรอบในการเรียนรู้ (Epoch) ที่เหมาะสม

```
plt.plot(hist.history['loss'])
plt.plot(hist.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='upper right')
plt.show()
```

ตารางที่ 3.13 Python Code สำหรับตรวจสอบรอบในการเรียนรู้ (Epoch) ที่เหมาะสม โดยระบุชื่อกราฟเป็น 'Model loss', แกน Y เป็นค่าของฟังก์ชันสูญเสีย (Loss Function), แกน X เป็นจำนวนรอบในการเรียนรู้ (Epoch), ชื่อค่าของฟังก์ชันสูญเสีย (Loss Function) ของชุดข้อมูลฝึกฝน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น ‘Train’, ชื่อค่าของฟังก์ชันสูญเสีย (Loss Function) ของชุดข้อมูลทดสอบเป็น ‘Val’ โดยให้ชื่อระบุอยู่บนมุมขวาบนของกราฟ

โดยสามารถเขียนคำสั่งในการสร้างตัวแบบ (Modeling) ได้ดังตารางที่ 3.14

ตารางที่ 3.13 แสดง Python Code สำหรับการสร้างตัวแบบ (Modeling)

```
model=Sequential()
n_cols=X_train.shape[1]
model.add(Dense(44, activation='sigmoid', input_shape=(n_cols,)))
model.add(Dropout (0.1))
model.add(Dense(1, activation='sigmoid'))
Adam=optimizers.Adam(lr=0.006, beta_1=0.9, beta_2=0.999)
model.compile(optimizer=Adam, loss='mean_squared_error')
hist=model.fit(X_train, Y_train, epochs= 200,validation_data=(X_test,Y_test))
```

จากตารางที่ 3.14 Python Code สำหรับการสร้างตัวแบบ (Modeling) โดยกำหนดโครงสร้าง 87-44-1 ซึ่งเป็นโครงข่ายที่มีโหนดในชั้นอินพุต (Input Node) เท่ากับ 87 โหนด, ชั้นซ่อน (Hidden Node) เท่ากับ 44 โหนด, ชั้นเอาท์พุต (Output Node) เท่ากับ 1 โหนด และกำหนดค่าไฮเปอร์พารามิเตอร์ Dropout เท่ากับ 0.1, ฟังก์ชันกระตุ้น (Activation Function) เป็น Sigmoid สำหรับชั้นซ่อน (Hidden Layer) และชั้นเอาท์พุต (Output Layer), Optimizer เป็น Adam, อัตราการเรียนรู้ (Learning Rate) เท่ากับ 0.006, β_1 เท่ากับ 0.9, β_2 เท่ากับ 0.999, ฟังก์ชันสูญเสีย (Loss Function) เป็น Mean Square Error (MSE) และจำนวนรอบในการเรียนรู้ (Epoch) เท่ากับ 200 โดยนำข้อมูลชุดฝึกฝนมาสร้างตัวแบบ และนำชุดข้อมูลทดสอบมาตรวจสอบ Overfitting

3.5 การประเมินผล (Evaluation)

หลังจากการสร้างตัวแบบ (Modeling) จะเป็นการนำผลลัพธ์ที่ได้จากตัวแบบมาเปรียบเทียบกันระหว่างตัวแบบ เพื่อเลือกตัวแบบที่มีประสิทธิภาพที่สุด โดยพิจารณาจากค่า MAPE สำหรับการนำมาแทนที่การพยากรณ์ยอดขายสินค้าของแต่ละพื้นที่การขาย (Sale Forecast) ซึ่งจะแสดงผลลัพธ์ในบทที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 การนำไปใช้ (Deployment)

เป็นขั้นตอนที่จะเกิดขึ้นในอนาคต เป็นการนำตัวแบบที่จะนำมาแทนที่การพยากรณ์ยอดขายสินค้าของแต่ละพื้นที่การขาย (Sale Forecast) โดยต้องผ่านการประเมินผลทุกๆ เดือนในอนาคต จนกว่าทางบริษัทจะเชื่อมั่นที่จะนำตัวแบบไปใช้งานจริง ถ้าไม่ผ่านการประเมินผลเดือนใดเดือนหนึ่งจะต้องย้อนกลับไปทำขั้นตอนของกระบวนการ CRISP-DM เพื่อสร้างตัวแบบใหม่มาแทนที่การพยากรณ์ยอดขายสินค้าของแต่ละพื้นที่การขาย (Sale Forecast)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการวิจัย

4.1 ตัวแบบที่เหมาะสมสำหรับการพยากรณ์

การพยากรณ์ด้วยวิธีโครงข่ายประสาทเทียม เมื่อปรับค่าไฮเปอร์พารามิเตอร์เพื่อหาตัวแบบพยากรณ์ที่เหมาะสมได้ผลดังตารางที่ 4.23 ซึ่งผลลัพธ์ของการทดลองปรับค่าไฮเปอร์พารามิเตอร์สามารถดูได้ในตารางที่ 4.1 – 4.15

ตารางที่ 4.1 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ Adam และ Activation Function คือ Sigmoid (Hidden Layer), Sigmoid (Output Layer)

โครงข่าย	อัตราการเรียนรู้	Training Set		Test Set		Validation Set	
		MAPE	MAD	MAPE	MAD	MAPE	MAD
87-22-1	0.001	27.77	89,115.12	33.78	92,730.50	22.76	83,419.63
	0.002	26.72	84,595.60	34.10	88,595.43	23.50	91,139.39
	0.004	25.54	78,891.58	33.02	86,083.72	28.58	124,766.90
	0.006	24.07	77,586.14	29.40	86,326.12	39.33	165,290.59
	0.008	25.97	78,260.03	32.10	87,879.18	39.62	175,132.63
87-44-1	0.001	28.19	88,348.92	35.07	91,582.09	**22.27**	**79,502.95**
	0.002	25.71	82,915.80	33.35	87,714.31	24.37	96,772.82
	0.004	24.17	78,488.98	30.46	85,928.93	31.74	141,741.60
	0.006	**23.83**	77,797.22	**28.31**	87,665.06	42.78	180,895.96
	0.008	24.15	**75,239.75**	29.95	**84,641.90**	41.50	180,882.54
87-87-1	0.001	27.91	88,639.45	35.00	92,074.91	23.13	84,786.16
	0.002	25.26	82,169.49	31.93	86,606.25	27.47	117,546.55
	0.004	24.40	78,132.90	30.46	86,692.02	32.08	145,944.13
	0.006	25.86	77,165.57	31.57	85,568.87	30.56	135,714.91
	0.008	26.69	75,308.49	33.39	86,501.23	36.24	169,831.63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 4.1 เมื่อกำหนด Optimizer คือ Adam และ Activation Function คือ Sigmoid (Hidden Layer), Sigmoid (Output Layer) พบว่าในชุดข้อมูลฝึกฝน (Training Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 23.83) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.006 และค่า MAD ต่ำที่สุด (MAD = 75,239.75) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และพบว่าในชุดข้อมูลทดสอบ (Test Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 28.31) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.006 และค่า MAD ต่ำที่สุด (MAD = 84,641.90) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และพบว่าในชุดข้อมูลทวนสอบ (Validation Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 22.27) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.001 และค่า MAD ต่ำที่สุด (MAD = 79,502.95) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ Adam และ Activation Function คือ Relu (Hidden Layer) , Relu (Output Layer)

โครงข่าย	อัตราการเรียนรู้	Training Set		Test Set		Validation Set	
		MAPE	MAD	MAPE	MAD	MAPE	MAD
87-22-1	0.001	30.97	81,727.19	39.93	95,576.32	57.09	180,419.99
	0.002	26.40	74,235.62	40.88	92,392.91	**35.94**	**108,682.46**
	0.004	27.08	76,632.03	39.54	90,331.97	66.30	233,464.02
	0.006	27.37	80,211.28	39.20	98,761.98	44.83	137,430.99
	0.008	31.96	86,092.78	42.07	99,711.49	70.25	277,657.11
87-44-1	0.001	23.82	74,067.94	34.97	95,523.53	56.59	189,004.55
	0.002	24.62	71,718.50	**30.10**	95,445.50	72.75	275,858.14
	0.004	26.47	75,891.48	32.23	99,706.93	82.89	346,872.53
	0.006	33.00	81,220.31	37.96	103,436.56	87.16	365,125.43
	0.008	27.30	78,181.32	39.18	95,781.12	38.12	113,163.14
87-87-1	0.001	**22.69**	**64,230.64**	42.09	**89,550.30**	44.34	146,297.43
	0.002	30.69	78,941.61	42.40	104,839.70	56.39	185,494.81
	0.004	30.86	74,713.13	36.98	102,218.09	80.72	332,379.54
	0.006	29.16	75,319.66	34.90	97,639.10	74.54	289,219.24
	0.008	33.15	86,220.50	37.43	104,907.12	96.70	412,149.53

จากตารางที่ 4.2 เมื่อกำหนด Optimizer คือ Adam และ Activation Function คือ Relu (Hidden Layer), Relu (Output Layer) พบว่าในชุดข้อมูลฝึกฝน (Training Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 22.69) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.001 และค่า MAD ต่ำที่สุด (MAD = 64,230.64) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.001 และพบว่าในชุดข้อมูลทดสอบ (Test Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 30.10) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.002 และค่า MAD ต่ำที่สุด (MAD = 89,550.30) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.001 และพบว่าในชุดข้อมูลทวนสอบ (Validation Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 35.94) คือโครงข่าย 87-22-1 โดยมีอัตราการเรียนรู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เท่ากับ 0.002 และค่า MAD ต่ำที่สุด (MAD = 108,682.46) คือโครงข่าย 87-22-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.002

ตารางที่ 4.3 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ Adam และ Activation Function คือ Relu (Hidden Layer) , Sigmoid (Output Layer)

โครงข่าย	อัตราการเรียนรู้	Training Set		Test Set		Validation Set	
		MAPE	MAD	MAPE	MAD	MAPE	MAD
87-22-1	0.001	24.40	76,915.59	32.81	83,777.79	26.27	112,054.85
	0.002	23.50	73,154.04	32.74	84,761.77	26.61	117,789.60
	0.004	23.76	71,303.09	33.25	85,963.36	25.66	112,449.92
	0.006	23.70	73,424.24	30.46	85,485.96	41.31	183,726.54
	0.008	25.06	72,039.29	36.54	89,570.01	36.08	151,937.27
87-44-1	0.001	23.32	71,752.76	32.45	83,885.95	26.96	121,531.98
	0.002	25.10	70,073.36	34.25	83,875.32	**24.15**	**103,634.06**
	0.004	22.59	67,699.72	31.25	88,810.40	41.32	184,889.19
	0.006	25.39	73,634.24	34.37	88,853.88	33.75	155,048.21
	0.008	22.87	71,326.71	**30.15**	90,760.97	42.60	189,899.65
87-87-1	0.001	23.31	67,941.18	32.57	**83,486.90**	33.91	159,642.15
	0.002	**20.69**	63,495.48	32.41	87,444.84	29.39	142,773.82
	0.004	23.02	**62,429.84**	35.25	88,455.73	29.46	126,824.35
	0.006	22.46	63,002.42	33.26	87,954.99	38.70	184,698.95
	0.008	25.04	67,485.66	33.45	92,261.42	53.76	244,327.49

จากตารางที่ 4.3 เมื่อกำหนด Optimizer คือ Adam และ Activation Function คือ Relu (Hidden Layer), Sigmoid (Output Layer) พบว่าในชุดข้อมูลฝึกฝน (Training Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 20.69) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.002 และค่า MAD ต่ำที่สุด (MAD = 62,429.84) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.004 และพบว่าในชุดข้อมูลทดสอบ (Test Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 30.15) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และค่า MAD ต่ำที่สุด (MAD = 83,486.90)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.001 และพบว่าในชุดข้อมูลทวนสอบ (Validation Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 24.15) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.002 และค่า MAD ต่ำที่สุด (MAD = 103,634.06) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.002

ตารางที่ 4.4 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ Adam และ Activation Function คือ Sigmoid (Hidden Layer), Relu (Output Layer)

โครงข่าย	อัตราการเรียนรู้	Training Set		Test Set		Validation Set	
		MAPE	MAD	MAPE	MAD	MAPE	MAD
87-22-1	0.001	31.21	92,213.91	41.48	95,190.78	26.71	78,825.96
	0.002	29.86	89,261.30	39.64	93,295.17	26.18	85,181.44
	0.004	29.36	85,446.81	35.69	91,484.38	51.30	137,438.91
	0.006	27.47	81,188.04	**34.22**	**90,011.15**	50.40	141,106.22
	0.008	**26.96**	**79,932.76**	34.78	90,751.42	46.32	127,811.35
87-44-1	0.001	28.66	91,900.35	37.74	97,494.45	26.30	90,584.13
	0.002	99.61	472,805.35	99.65	500,855.01	99.66	431,395.40
	0.004	27.44	80,898.70	34.59	90,728.83	45.70	134,327.37
	0.006	99.61	472,805.35	99.65	500,855.01	99.66	431,395.40
	0.008	99.61	472,805.35	99.65	500,855.01	99.66	431,395.40
87-87-1	0.001	29.62	90,882.76	38.98	93,475.82	**24.67**	**78,808.07**
	0.002	29.90	85,406.17	39.91	90,974.30	25.37	87,208.31
	0.004	27.91	80,987.31	34.79	88,791.50	45.23	132,448.09
	0.006	99.61	472,805.35	99.65	500,855.01	99.66	431,395.40
	0.008	99.61	472,805.35	99.65	500,855.01	99.66	431,395.40

จากตารางที่ 4.4 เมื่อกำหนด Optimizer คือ Adam และ Activation Function คือ Sigmoid (Hidden Layer), Relu (Output Layer) พบว่าในชุดข้อมูลฝึกฝน (Training Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 26.96) คือโครงข่าย 87-22-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และค่า MAD ต่ำที่สุด (MAD = 79,932.76) คือโครงข่าย 87-22-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พบว่าในชุดข้อมูลทดสอบ (Test Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 34.22) คือโครงข่าย 87-22-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.006 และค่า MAD ต่ำที่สุด (MAD = 90,011.15) คือ โครงข่าย 87-22-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.006 และพบว่าในชุดข้อมูลทวนสอบ (Validation Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 24.67) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.001 และค่า MAD ต่ำที่สุด (MAD = 78,808.07) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.001

ตารางที่ 4.5 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ SGD, Activation Function คือ Sigmoid (Hidden Layer) , Sigmoid (Output Layer) และโมเมนตัม 0.5

โครงข่าย	อัตราการเรียนรู้	Training Set		Test Set		Validation Set	
		MAPE	MAD	MAPE	MAD	MAPE	MAD
87-22-1	0.001	211.39	306,646.73	185.22	322,230.91	165.85	263,945.50
	0.002	217.24	307,867.56	188.40	323,896.99	177.88	272,369.89
	0.004	**207.88**	304,250.33	184.92	319,539.91	163.50	262,342.06
	0.006	220.07	307,825.26	192.19	323,034.44	182.58	274,657.89
	0.008	211.49	**300,424.52**	183.58	**314,664.97**	175.83	266,538.61
87-44-1	0.001	214.92	308,577.66	187.03	324,847.15	186.54	278,778.45
	0.002	211.89	305,524.74	185.64	321,677.65	171.08	268,032.62
	0.004	215.69	307,446.31	188.36	322,034.31	190.20	279,171.71
	0.006	210.02	306,994.21	185.63	322,431.91	172.39	269,525.34
	0.008	211.73	301,347.09	187.76	316,140.68	176.75	265,559.56
87-87-1	0.001	215.87	308,382.19	186.80	324,200.46	180.60	274,329.32
	0.002	210.42	306,386.99	**182.87**	321,201.47	165.16	265,240.98
	0.004	215.32	304,960.21	188.59	319,827.95	179.41	272,462.84
	0.006	215.47	304,921.79	186.84	319,890.10	176.02	269,288.36
	0.008	215.55	301,543.58	185.43	316,827.38	**159.66**	**256,026.36**

จากตารางที่ 4.5 เมื่อกำหนด Optimizer คือ SGD, Activation Function คือ Sigmoid (Hidden Layer), Sigmoid (Output Layer) และโมเมนตัมเท่ากับ 0.5 พบว่าในชุดข้อมูลฝึกฝน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Training Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 207.88) คือโครงข่าย 87-22-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.004 และค่า MAD ต่ำที่สุด (MAD = 300,424.52) คือโครงข่าย 87-22-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และพบว่าในชุดข้อมูลทดสอบ (Test Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 182.87) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.002 และค่า MAD ต่ำที่สุด (MAD = 314,664.97) คือโครงข่าย 87-22-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และพบว่าในชุดข้อมูลทวนสอบ (Validation Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 159.66) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และค่า MAD ต่ำที่สุด (MAD = 256,026.36) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008

ตารางที่ 4.6 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ SGD, Activation Function คือ Sigmoid (Hidden Layer) , Sigmoid (Output Layer) และโมเมนตัม 0.7

โครงข่าย	อัตราการเรียนรู้	Training Set		Test Set		Validation Set	
		MAPE	MAD	MAPE	MAD	MAPE	MAD
87-22-1	0.001	211.87	302,399.52	**183.01**	317,220.74	167.90	262,694.76
	0.002	212.81	307,271.03	186.61	322,926.67	**162.62**	263,339.02
	0.004	213.97	302,461.00	184.90	317,716.95	169.35	262,102.86
	0.006	216.24	304,622.27	187.19	320,600.50	174.59	268,935.41
	0.008	212.51	298,929.28	184.35	314,187.01	168.17	259,736.05
87-44-1	0.001	216.73	305,915.58	186.49	322,082.78	168.53	266,188.43
	0.002	215.80	306,223.34	186.06	321,031.30	189.46	279,948.23
	0.004	214.31	302,859.52	186.14	317,901.87	183.15	271,265.44
	0.006	210.56	299,934.98	184.61	314,592.45	168.74	262,209.40
	0.008	**210.03**	**296,413.03**	187.68	**310,638.37**	166.13	**255,707.82**
87-87-1	0.001	212.56	305,442.10	187.63	320,217.89	183.81	274,247.64
	0.002	214.32	306,088.28	186.09	320,899.00	187.46	278,575.36
	0.004	214.29	302,846.33	185.57	317,180.80	171.98	264,344.80
	0.006	210.04	297,638.02	183.79	312,592.97	165.64	257,582.59
	0.008	213.34	297,076.12	183.43	310,863.33	163.71	255,996.32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 4.6 เมื่อกำหนด Optimizer คือ SGD, Activation Function คือ Sigmoid (Hidden Layer), Sigmoid (Output Layer) และโมเมนตัมเท่ากับ 0.7 พบว่าในชุดข้อมูลฝึกฝน (Training Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 210.03) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และค่า MAD ต่ำที่สุด (MAD = 296,413.03) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และพบว่าในชุดข้อมูลทดสอบ (Test Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 183.01) คือโครงข่าย 87-22-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.001 และค่า MAD ต่ำที่สุด (MAD = 310,638.37) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และพบว่าในชุดข้อมูลทวนสอบ (Validation Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 162.62) คือโครงข่าย 87-22-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.002 และค่า MAD ต่ำที่สุด (MAD = 255,707.82) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.7 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ SGD, Activation Function คือ Sigmoid (Hidden Layer) , Sigmoid (Output Layer) และโมเมนตัม 0.9

โครงข่าย	อัตราการเรียนรู้	Training Set		Test Set		Validation Set	
		MAPE	MAD	MAPE	MAD	MAPE	MAD
87-22-1	0.001	219.13	306,236.09	187.10	321,759.53	173.83	267,591.94
	0.002	217.80	307,778.62	190.40	323,099.73	173.39	267,967.92
	0.004	195.66	268,105.02	169.73	280,629.32	153.83	231,797.74
	0.006	198.38	275,650.45	175.15	288,802.28	148.86	234,770.44
	0.008	177.21	242,637.56	157.62	255,680.63	136.74	207,097.46
87-44-1	0.001	214.60	305,848.28	187.69	321,489.39	175.50	269,758.40
	0.002	210.81	298,014.81	184.60	312,963.65	170.00	261,214.83
	0.004	213.48	294,982.05	185.33	309,666.57	167.87	256,091.96
	0.006	162.12	223,598.29	146.26	237,239.01	115.87	188,940.72
	0.008	**154.07**	217,802.35	**138.34**	229,084.99	**109.22**	185,885.52
87-87-1	0.001	213.05	299,729.51	187.11	315,037.32	174.83	263,597.87
	0.002	211.70	296,743.33	185.62	311,921.55	165.02	254,698.97
	0.004	205.22	282,546.06	177.12	295,958.76	151.00	239,135.15
	0.006	177.01	246,088.82	157.69	259,597.34	137.51	212,039.15
	0.008	157.63	**215,341.01**	139.72	**227,522.43**	113.04	**182,571.39**

จากตารางที่ 4.7 เมื่อกำหนด Optimizer คือ SGD, Activation Function คือ Sigmoid (Hidden Layer), Sigmoid (Output Layer) และโมเมนตัมเท่ากับ 0.9 พบว่าในชุดข้อมูลฝึกฝน (Training Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 154.07) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และค่า MAD ต่ำที่สุด (MAD = 215,341.01) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และพบว่าในชุดข้อมูลทดสอบ (Test Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 138.34) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และค่า MAD ต่ำที่สุด (MAD = 227,522.43) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และพบว่าในชุดข้อมูลทวนสอบ (Validation Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE =

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

109.22) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และค่า MAD ต่ำที่สุด (MAD = 182,571.39) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008

ตารางที่ 4.8 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ SGD, Activation Function คือ Relu (Hidden Layer), Relu (Output Layer) และโมเมนตัม 0.5

โครงข่าย	อัตราการเรียนรู้	Training Set		Test Set		Validation Set	
		MAPE	MAD	MAPE	MAD	MAPE	MAD
87-22-1	0.001	134.28	204,774.25	98.94	205,063.53	93.98	198,742.31
	0.002	80.51	143,471.77	65.16	145,850.64	62.70	150,528.31
	0.004	79.73	135,066.45	86.33	146,726.40	58.46	169,897.79
	0.006	49.53	115,493.22	58.12	123,756.24	57.45	145,007.68
	0.008	45.77	108,558.78	45.69	118,779.36	45.81	145,906.97
87-44-1	0.001	89.23	184,719.74	90.89	190,547.31	87.03	207,526.60
	0.002	63.24	146,408.13	70.61	159,954.66	72.10	184,452.13
	0.004	48.60	123,500.21	46.73	130,175.55	56.07	173,270.93
	0.006	41.95	114,568.85	42.92	121,047.96	54.95	172,921.14
	0.008	37.97	103,171.24	44.52	108,184.63	**44.43**	134,740.97
87-87-1	0.001	58.55	150,518.80	65.12	159,273.90	62.29	163,318.08
	0.002	84.43	169,828.95	74.68	174,414.84	65.90	216,193.30
	0.004	40.42	109,388.95	46.62	117,391.39	50.03	159,909.27
	0.006	40.57	101,245.20	**39.08**	**107,195.81**	53.14	156,711.58
	0.008	**37.91**	**99,719.88**	47.54	107,998.83	45.24	**131,744.00**

จากตารางที่ 4.8 เมื่อกำหนด Optimizer คือ SGD, Activation Function คือ Relu (Hidden Layer), Relu (Output Layer) และโมเมนตัมเท่ากับ 0.5 พบว่าในชุดข้อมูลฝึกฝน (Training Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 37.91) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และค่า MAD ต่ำที่สุด (MAD = 99,719.88) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และพบว่าในชุดข้อมูลทดสอบ (Test Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 39.08) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.006 และค่า MAD ต่ำที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(MAD = 107,195.81) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.006 และพบว่าในชุดข้อมูลทวนสอบ (Validation Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 44.43) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และค่า MAD ต่ำที่สุด (MAD = 131,744.00) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008

ตารางที่ 4.9 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ SGD ฟังก์ชันกระตุ้น (Activation Function) คือ Relu (Hidden Layer), Relu (Output Layer) และโมเมนตัม 0.7

โครงข่าย	อัตราการเรียนรู้	Training Set		Test Set		Validation Set	
		MAPE	MAD	MAPE	MAD	MAPE	MAD
87-22-1	0.001	136.09	201,792.58	124.57	212,782.01	131.39	208,422.48
	0.002	62.23	139,310.48	67.21	149,389.90	55.94	155,504.62
	0.004	67.22	118,490.94	70.31	128,196.35	53.23	152,687.79
	0.006	47.83	104,767.60	53.06	112,297.94	46.26	125,668.37
	0.008	35.22	100,115.15	48.72	106,709.67	48.24	142,701.38
87-44-1	0.001	80.83	152,558.93	72.91	166,106.06	75.57	185,557.64
	0.002	52.64	121,924.30	44.15	130,991.48	51.52	153,921.55
	0.004	40.10	107,620.12	50.29	114,280.32	**39.49**	136,742.51
	0.006	37.92	100,003.02	50.32	111,364.69	60.31	169,179.26
	0.008	34.50	98,011.64	43.22	103,584.33	45.94	144,988.11
87-87-1	0.001	60.66	138,647.56	60.20	146,606.15	69.16	220,176.59
	0.002	42.93	113,368.85	54.32	123,545.73	52.30	174,566.28
	0.004	41.50	104,345.13	48.45	114,821.53	40.97	**116,001.78**
	0.006	34.19	95,414.72	41.99	102,537.93	41.89	126,345.78
	0.008	**31.96**	**91,869.96**	**37.72**	**99,979.25**	49.00	157,375.37

จากตารางที่ 4.9 เมื่อกำหนด Optimizer คือ SGD, Activation Function คือ Relu (Hidden Layer), Relu (Output Layer) และโมเมนตัมเท่ากับ 0.7 พบว่าในชุดข้อมูลฝึกฝน (Training Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 31.96) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และค่า MAD ต่ำที่สุด (MAD = 91,869.96) คือโครงข่าย 87-87-1 โดยมีอัตราการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรียนรู้เท่ากับ 0.008 และพบว่าในชุดข้อมูลทดสอบ (Test Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 37.72) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และค่า MAD ต่ำที่สุด (MAD = 99,979.25) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และพบว่าในชุดข้อมูลทวนสอบ (Validation Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 39.49) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.004 และค่า MAD ต่ำที่สุด (MAD = 116,001.78) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.004

ตารางที่ 4.10 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ SGD ฟังก์ชันกระตุ้น (Activation Function) คือ Relu (Hidden Layer), Relu (Output Layer) และโมเมนตัม 0.9

โครงข่าย	อัตราการเรียนรู้	Training Set		Test Set		Validation Set	
		MAPE	MAD	MAPE	MAD	MAPE	MAD
87-22-1	0.001	59.88	125,364.83	62.52	135391.91	49.09	147,611.52
	0.002	58.89	105,898.77	51.54	112,920.12	60.79	124,317.51
	0.004	36.21	94,158.20	42.92	103,767.40	36.27	**100,801.83**
	0.006	38.45	95,917.31	44.25	101,942.21	36.51	110,653.14
	0.008	35.10	92,663.22	45.79	101,219.70	45.55	136,706.92
87-44-1	0.001	58.45	114,852.84	59.45	121,667.36	59.44	182,570.51
	0.002	37.88	105,742.51	**37.43**	112,191.47	55.43	177,339.83
	0.004	40.57	95,795.63	45.47	104,398.90	42.08	125,271.42
	0.006	32.56	89,689.06	48.39	100,647.03	34.08	107,417.14
	0.008	32.71	88,937.97	45.51	95,318.16	34.18	113,061.57
87-87-1	0.001	41.56	110,681.82	52.01	122,202.74	51.74	167,920.50
	0.002	33.69	96,532.64	39.02	105,570.20	54.09	177,046.98
	0.004	29.51	87,535.98	41.65	97,412.78	40.13	119,001.27
	0.006	32.04	88,003.54	41.10	96,092.58	**34.03**	116,951.63
	0.008	**28.56**	**84,353.72**	38.49	**92,598.68**	36.64	124,792.74

จากตารางที่ 4.10 เมื่อกำหนด Optimizer คือ SGD, Activation Function คือ Relu (Hidden Layer), Relu (Output Layer) และโมเมนตัมเท่ากับ 0.9 พบว่าในชุดข้อมูลฝึกฝน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Training Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 28.56) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และค่า MAD ต่ำที่สุด (MAD = 84,353.72) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และพบว่าในชุดข้อมูลทดสอบ (Test Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 37.43) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.002 และค่า MAD ต่ำที่สุด (MAD = 92,598.68) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และพบว่าในชุดข้อมูลทวนสอบ (Validation Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 34.03) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.006 และค่า MAD ต่ำที่สุด (MAD = 100,801.83) คือโครงข่าย 87-22-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.004

ตารางที่ 4.11 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ SGD ฟังก์ชันกระตุ้น (Activation Function) คือ Relu (Hidden Layer), Sigmoid (Output Layer) และโมเมนตัม 0.5

โครงข่าย	อัตราการเรียนรู้	Training Set		Test Set		Validation Set	
		MAPE	MAD	MAPE	MAD	MAPE	MAD
87-22-1	0.001	234.25	314,570.91	198.88	322,808.73	178.09	272,698.07
	0.002	190.02	279,099.76	172.77	288,651.47	123.56	229,984.08
	0.004	183.81	258,995.48	163.86	273,429.17	131.01	217,934.23
	0.006	178.21	243,777.66	154.42	252,547.43	119.79	201,379.48
	0.008	125.87	178,584.37	125.48	188,935.26	83.04	157,882.15
87-44-1	0.001	216.94	317,112.36	192.55	331,372.32	143.39	255,039.22
	0.002	204.11	302,791.56	182.91	316,659.34	156.48	252,005.38
	0.004	183.50	247,300.58	165.25	255,376.48	142.86	212,754.09
	0.006	145.88	197,307.25	124.71	208,956.07	85.53	169,376.22
	0.008	112.02	160,881.68	106.92	168,133.37	71.47	149,824.41
87-87-1	0.001	220.86	303,343.50	192.45	316,152.25	153.02	253,833.91
	0.002	197.29	274,761.10	177.25	284,876.13	134.84	228,361.34
	0.004	173.84	233,523.96	150.64	241,680.08	127.84	197,127.89
	0.006	154.60	196,248.32	135.52	207,334.94	95.46	174,586.47
	0.008	108.85	159,252.46	107.22	171,034.85	69.50	143,986.38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 4.11 เมื่อกำหนด Optimizer คือ SGD, Activation Function คือ Relu (Hidden Layer), Sigmoid (Output Layer) และโมเมนตัมเท่ากับ 0.5 พบว่าในชุดข้อมูลฝึกฝน (Training Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 108.85) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และค่า MAD ต่ำที่สุด (MAD = 159,252.46) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และพบว่าในชุดข้อมูลทดสอบ (Test Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 106.92) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และค่า MAD ต่ำที่สุด (MAD = 168,133.37) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และพบว่าในชุดข้อมูลทวนสอบ (Validation Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 69.50) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และค่า MAD ต่ำที่สุด (MAD = 143,986.38) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.12 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ SGD ฟังก์ชันกระตุ้น (Activation Function) คือ Relu (Hidden Layer), Sigmoid (Output Layer) และโมเมนตัม 0.7

โครงข่าย	อัตราการเรียนรู้	Training Set		Test Set		Validation Set	
		MAPE	MAD	MAPE	MAD	MAPE	MAD
87-22-1	0.001	209.41	295,550.70	176.54	304,077.74	122.66	240,812.84
	0.002	171.09	259,786.68	163.10	269,598.15	115.94	215,948.34
	0.004	138.36	205,323.60	130.95	213,822.42	96.95	173,411.58
	0.006	107.96	153,313.78	91.91	161,401.60	68.31	140,831.32
	0.008	86.95	136,208.35	**78.80**	147,068.68	**54.10**	**137,437.65**
87-44-1	0.001	207.36	305,603.99	181.68	320,083.46	128.73	248,109.50
	0.002	192.72	263,512.26	169.97	276,768.71	123.63	219,870.53
	0.004	134.19	186,689.95	122.27	192,592.92	94.09	158,951.58
	0.006	108.34	165,215.60	96.35	172,254.31	67.71	149,330.17
	0.008	94.62	140,517.67	88.14	144,840.90	63.70	147,802.86
87-87-1	0.001	218.34	304,009.35	190.11	316,234.54	142.96	246,539.12
	0.002	194.44	262,748.52	172.15	272,107.45	123.83	211,757.75
	0.004	122.89	162,977.37	112.11	174,741.78	78.27	154,356.56
	0.006	106.48	146979.56	96.22	152,837.44	67.54	140,479.77
	0.008	**86.64**	**132,495.75**	79.01	**137,156.57**	55.29	141,620.57

จากตารางที่ 4.12 เมื่อกำหนด Optimizer คือ SGD, Activation Function คือ Relu (Hidden Layer), Sigmoid (Output Layer) และโมเมนตัมเท่ากับ 0.7 พบว่าในชุดข้อมูลฝึกฝน (Training Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 86.64) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และค่า MAD ต่ำที่สุด (MAD = 132,495.75) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และพบว่าในชุดข้อมูลทดสอบ (Test Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 78.80) คือโครงข่าย 87-22-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และค่า MAD ต่ำที่สุด (MAD = 137,156.57) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และพบว่าในชุดข้อมูลทวนสอบ (Validation Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 54.10) คือโครงข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

87-22-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และค่า MAD ต่ำที่สุด (MAD = 137,437.65) คือ โครงข่าย 87-22-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008

ตารางที่ 4.13 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ SGD ฟังก์ชันกระตุ้น (Activation Function) คือ Relu (Hidden Layer), Sigmoid (Output Layer) และโมเมนตัม 0.9

โครงข่าย	อัตราการเรียนรู้	Training Set		Test Set		Validation Set	
		MAPE	MAD	MAPE	MAD	MAPE	MAD
87-22-1	0.001	164.01	242,704.44	157.03	255,824.46	127.09	209,801.87
	0.002	98.19	155,261.47	101.20	161,039.67	72.76	141,798.80
	0.004	78.91	125,054.92	72.28	130,695.40	53.15	133,570.16
	0.006	76.98	118,760.57	70.92	122,054.77	52.92	124,913.52
	0.008	65.56	111,004.39	61.98	113,964.22	50.00	**115,538.25**
87-44-1	0.001	166.12	228,083.87	148.31	237,679.25	108.35	187,397.30
	0.002	99.42	148,825.82	91.38	155,232.83	62.42	150,589.95
	0.004	80.32	124,559.91	70.09	128,858.31	54.57	130,040.02
	0.006	66.33	112,519.67	60.64	117,981.35	47.00	139,048.91
	0.008	63.64	108,106.50	61.45	111,985.82	48.57	129,983.19
87-87-1	0.001	171.20	230,183.13	151.66	239,469.11	119.71	199,913.98
	0.002	99.55	139,526.81	89.31	146,966.98	59.55	142,119.95
	0.004	71.32	118,292.13	69.34	121,789.38	46.78	133,445.84
	0.006	71.16	112,658.11	66.50	117,217.77	50.90	140,667.34
	0.008	**58.85**	**103,948.56**	**57.71**	**108,401.45**	**45.86**	145,894.05

จากตารางที่ 4.13 เมื่อกำหนด Optimizer คือ SGD, Activation Function คือ Relu (Hidden Layer), Sigmoid (Output Layer) และโมเมนตัมเท่ากับ 0.9 พบว่าในชุดข้อมูลฝึกฝน (Training Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 58.85) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และค่า MAD ต่ำที่สุด (MAD = 103,948.56) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และพบว่าในชุดข้อมูลทดสอบ (Test Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 57.71) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และค่า MAD ต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่สุด (MAD = 108,401.45) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และพบว่าในชุดข้อมูลทวนสอบ (Validation Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 45.86) คือโครงข่าย 87-87-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และค่า MAD ต่ำที่สุด (MAD = 115,538.25) คือโครงข่าย 87-22-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008

ตารางที่ 4.14 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ SGD ฟังก์ชันกระตุ้น (Activation Function) คือ Sigmoid (Hidden Layer) Relu (Output Layer) และโมเมนตัม 0.5

โครงข่าย	อัตราการเรียนรู้	Training Set		Test Set		Validation Set	
		MAPE	MAD	MAPE	MAD	MAPE	MAD
87-22-1	0.001	99.27	467,868.75	99.27	494,812.69	99.28	426,867.50
	0.002	138.02	273,913.22	122.89	293,624.15	117.93	233,228.44
	0.004	82.33	220,235.30	82.71	**240,775.10**	**58.78**	206,795.31
	0.006	182.83	264,283.94	165.04	277,698.76	137.72	221,530.24
	0.008	**57.15**	**217,906.85**	**61.56**	241,417.95	64.45	249,956.43
87-44-1	0.001	99.61	472,805.35	99.65	500,855.01	99.66	431,395.40
	0.002	129.17	243,903.10	106.29	262,808.71	81.26	219,140.75
	0.004	180.19	266,343.20	157.40	281,922.60	128.64	223,892.20
	0.006	199.84	279,605.82	181.24	293,161.59	145.17	230,439.47
	0.008	147.16	242,092.91	130.07	257,851.64	98.94	**205,514.92**
87-87-1	0.001	70.63	343,629.92	73.22	367,539.42	96.78	415,738.50
	0.002	110.18	250,154.60	92.12	269,436.14	75.65	248,352.44
	0.004	208.87	272,906.06	186.58	286,007.26	164.50	235,745.65
	0.006	176.60	258,734.87	159.60	271,186.13	135.50	217,595.04
	0.008	175.20	253,633.58	152.62	266,346.99	126.33	212,065.42

จากตารางที่ 4.14 เมื่อกำหนด Optimizer คือ SGD, Activation Function คือ Sigmoid (Hidden Layer), Relu (Output Layer) และโมเมนตัมเท่ากับ 0.5 พบว่าในชุดข้อมูลฝึกฝน (Training Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 57.15) คือโครงข่าย 87-22-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และค่า MAD ต่ำที่สุด (MAD = 217,906.85) คือโครงข่าย 87-22-1 โดยมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตราการเรียนรู้เท่ากับ 0.008 และพบว่าในชุดข้อมูลทดสอบ (Test Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 61.56) คือโครงข่าย 87-22-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008 และค่า MAD ต่ำที่สุด (MAD = 240,775.10) คือโครงข่าย 87-22-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.004 และพบว่าในชุดข้อมูลทวนสอบ (Validation Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 58.78) คือโครงข่าย 87-22-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.004 และค่า MAD ต่ำที่สุด (MAD = 205,514.92) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008

ตารางที่ 4.15 ค่า MAPE และ MAD ของการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม โดยใช้ Optimizer คือ SGD ฟังก์ชันกระตุ้น (Activation Function) คือ Sigmoid (Hidden Layer), Relu (Output Layer) และโมเมนตัม 0.7

โครงข่าย	อัตราการเรียนรู้	Training Set		Test Set		Validation Set	
		MAPE	MAD	MAPE	MAD	MAPE	MAD
87-22-1	0.001	127.71	269,089.77	108.12	291,662.33	94.82	236,602.51
	0.002	99.61	472,805.35	99.65	500,855.01	99.66	431,395.40
	0.004	114.85	238,017.02	100.75	256,482.59	84.31	206,673.86
	0.006	62.68	205,313.93	71.47	221,463.88	55.75	196,948.00
	0.008	189.53	266,345.07	164.80	279,041.85	137.18	218,483.36
87-44-1	0.001	99.61	472,805.35	99.65	500,855.01	99.66	431,395.40
	0.002	88.63	300,508.85	66.71	322,518.34	84.95	343,424.44
	0.004	99.37	471,052.23	99.49	499,049.36	95.36	415,363.89
	0.006	189.71	263,994.44	166.05	277,210.35	143.21	222,961.35
	0.008	131.95	224,550.11	122.87	239,955.45	90.51	189,523.20
87-87-1	0.001	120.05	250,219.47	118.69	272,904.29	113.43	216,160.90
	0.002	194.73	266,793.87	166.63	277,574.64	146.54	225,082.16
	0.004	99.61	472,805.35	99.65	500,855.01	99.66	431,395.40
	0.006	99.61	472,805.35	99.65	500,855.01	99.66	431,395.40
	0.008	186.01	245,715.56	162.61	256,404.47	128.80	198,385.29

จากตารางที่ 4.15 เมื่อกำหนด Optimizer คือ SGD, Activation Function คือ Sigmoid (Hidden Layer), Relu (Output Layer) และโมเมนตัมเท่ากับ 0.7 พบว่าในชุดข้อมูลฝึกฝน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Training Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 62.68) คือโครงข่าย 87-22-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.006 และค่า MAD ต่ำที่สุด (MAD = 205,313.93) คือโครงข่าย 87-22-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.006 และพบว่าในชุดข้อมูลทดสอบ (Test Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 66.71) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.002 และค่า MAD ต่ำที่สุด (MAD = 221,463.88) คือโครงข่าย 87-22-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.006 และพบว่าในชุดข้อมูลทวนสอบ (Validation Set) โครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 55.75) คือโครงข่าย 87-22-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.006 และค่า MAD ต่ำที่สุด (MAD = 189,523.20) คือโครงข่าย 87-44-1 โดยมีอัตราการเรียนรู้เท่ากับ 0.008

โดยการเลือกตัวแบบที่เหมาะสม สามารถเลือกชุดข้อมูลฝึกฝน (Training Set), ชุดข้อมูลทดสอบ (Test Set) หรือชุดข้อมูลทวนสอบ (Validation Set) และ MAPE หรือ MAD ก็ได้แล้วแต่ดุลพินิจของบริษัทที่จะนำไปใช้ ซึ่งทางผู้จัดทำได้ทำการสรุปต่างๆ ดังตารางที่ 4.16 – 4.21

ตารางที่ 4.16 แสดงโครงข่ายประสาทเทียมที่ดีที่สุดแต่ละ Optimizer และ Activation Function โดยสนใจที่ชุดข้อมูลฝึกฝน (Training Set) และ MAPE

โครงข่าย	Optimizer	อัตราการเรียนรู้	โมเมนตัม	Activation Function		MAPE
				ชั้นซ่อน	ชั้นเอาท์พุต	
87-44-1	Adam	0.006	-	Sigmoid	Sigmoid	23.83
87-87-1	Adam	0.001	-	Relu	Relu	22.69
87-87-1	Adam	0.002	-	Relu	Sigmoid	**20.69**
87-22-1	Adam	0.008	-	Sigmoid	Relu	26.96
87-22-1	SGD	0.004	0.5	Sigmoid	Sigmoid	207.88
87-44-1	SGD	0.008	0.7	Sigmoid	Sigmoid	210.03
87-44-1	SGD	0.008	0.9	Sigmoid	Sigmoid	154.07
87-87-1	SGD	0.008	0.5	Relu	Relu	37.91
87-87-1	SGD	0.008	0.7	Relu	Relu	31.96
87-87-1	SGD	0.008	0.9	Relu	Relu	28.56
87-87-1	SGD	0.008	0.5	Relu	Sigmoid	108.85
87-87-1	SGD	0.008	0.7	Relu	Sigmoid	86.64
87-87-1	SGD	0.008	0.9	Relu	Sigmoid	58.85
87-22-1	SGD	0.008	0.5	Sigmoid	Relu	57.15
87-22-1	SGD	0.006	0.7	Sigmoid	Relu	62.68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 4.16 เมื่อสนใจจะเลือกตัวแบบโดยพิจารณาจากชุดข้อมูลฝึกฝน (Training Set) และค่า MAPE พบว่าโครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 20.69) คือ 87-87-1 โดยใช้ Optimizer คือ Adam อัตราการเรียนรู้ (Learning Rate) เท่ากับ 0.002 และมีฟังก์ชันกระตุ้น (Activation Function) คือ Relu (Hidden Layer), Sigmoid (Output Layer)

ตารางที่ 4.17 แสดงโครงข่ายประสาทเทียมที่ดีที่สุดแต่ละ Optimizer และ Activation Function โดยสนใจที่ชุดข้อมูลฝึกฝน (Training Set) และ MAD

โครงข่าย	Optimizer	อัตราการเรียนรู้	โมเมนตัม	Activation Function		MAD
				ชั้นซ่อน	ชั้นเอาท์พุต	
87-44-1	Adam	0.008	-	Sigmoid	Sigmoid	75,239.75
87-87-1	Adam	0.001	-	Relu	Relu	64,230.64
87-87-1	Adam	0.004	-	Relu	Sigmoid	**62,429.84**
87-22-1	Adam	0.008	-	Sigmoid	Relu	79,932.76
87-22-1	SGD	0.008	0.5	Sigmoid	Sigmoid	300,424.52
87-44-1	SGD	0.008	0.7	Sigmoid	Sigmoid	296,413.03
87-87-1	SGD	0.008	0.9	Sigmoid	Sigmoid	215,341.01
87-87-1	SGD	0.008	0.5	Relu	Relu	99,719.88
87-87-1	SGD	0.008	0.7	Relu	Relu	91,869.96
87-87-1	SGD	0.008	0.9	Relu	Relu	84,353.72
87-87-1	SGD	0.008	0.5	Relu	Sigmoid	159,252.46
87-87-1	SGD	0.008	0.7	Relu	Sigmoid	132,495.75
87-87-1	SGD	0.008	0.9	Relu	Sigmoid	103,948.56
87-22-1	SGD	0.008	0.5	Sigmoid	Relu	217,906.85
87-22-1	SGD	0.006	0.7	Sigmoid	Relu	205,313.93

จากตารางที่ 4.17 เมื่อสนใจจะเลือกตัวแบบโดยพิจารณาจากชุดข้อมูลฝึกฝน (Training Set) และค่า MAD พบว่าโครงข่ายที่ให้ค่า MAD ต่ำที่สุด (MAD = 62,429.84) คือ 87-87-1 โดยใช้ Optimizer คือ Adam อัตราการเรียนรู้ (Learning Rate) เท่ากับ 0.004 และมีฟังก์ชันกระตุ้น (Activation Function) คือ Relu (Hidden Layer), Sigmoid (Output Layer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.18 แสดงโครงข่ายประสาทเทียมที่ดีที่สุดแต่ละ Optimizer และ Activation Function โดยสนใจที่ชุดข้อมูลทดสอบ (Test Set) และ MAPE

โครงข่าย	Optimizer	อัตราการเรียนรู้	โมเมนตัม	Activation Function		MAPE
				ชั้นซ่อน	ชั้นเอาท์พุต	
87-44-1	Adam	0.006	-	Sigmoid	Sigmoid	28.31
87-44-1	Adam	0.002	-	Relu	Relu	30.10
87-44-1	Adam	0.008	-	Relu	Sigmoid	30.15
87-22-1	Adam	0.006	-	Sigmoid	Relu	34.22
87-87-1	SGD	0.002	0.5	Sigmoid	Sigmoid	182.87
87-22-1	SGD	0.001	0.7	Sigmoid	Sigmoid	183.01
87-44-1	SGD	0.008	0.9	Sigmoid	Sigmoid	138.34
87-87-1	SGD	0.006	0.5	Relu	Relu	39.08
87-87-1	SGD	0.008	0.7	Relu	Relu	37.72
87-44-1	SGD	0.002	0.9	Relu	Relu	37.43
87-44-1	SGD	0.008	0.5	Relu	Sigmoid	106.92
87-22-1	SGD	0.008	0.7	Relu	Sigmoid	78.80
87-87-1	SGD	0.008	0.9	Relu	Sigmoid	57.71
87-22-1	SGD	0.008	0.5	Sigmoid	Relu	61.56
87-44-1	SGD	0.002	0.7	Sigmoid	Relu	66.71

จากตารางที่ 4.18 เมื่อสนใจจะเลือกตัวแบบโดยพิจารณาจากชุดข้อมูลทดสอบ (Test Set) และค่า MAPE พบว่าโครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 28.31) คือ 87-44-1 โดยใช้ Optimizer คือ Adam อัตราการเรียนรู้ (Learning Rate) เท่ากับ 0.006 และมีฟังก์ชันกระตุ้น (Activation Function) คือ Sigmoid (Hidden Layer), Sigmoid (Output Layer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.19 แสดงโครงข่ายประสาทเทียมที่ดีที่สุดแต่ละ Optimizer และ Activation Function โดยสนใจที่ชุดข้อมูลทดสอบ (Test Set) และ MAD

โครงข่าย	Optimizer	อัตราการเรียนรู้	โมเมนตัม	Activation Function		MAD
				ชั้นซ่อน	ชั้นเอาท์พุต	
87-44-1	Adam	0.008	-	Sigmoid	Sigmoid	84,641.90
87-87-1	Adam	0.001	-	Relu	Relu	89,550.30
87-87-1	Adam	0.001	-	Relu	Sigmoid	**83,486.90**
87-22-1	Adam	0.006	-	Sigmoid	Relu	90,011.15
87-22-1	SGD	0.008	0.5	Sigmoid	Sigmoid	314,664.97
87-44-1	SGD	0.008	0.7	Sigmoid	Sigmoid	310,638.37
87-87-1	SGD	0.008	0.9	Sigmoid	Sigmoid	227,522.43
87-87-1	SGD	0.006	0.5	Relu	Relu	107,195.81
87-87-1	SGD	0.008	0.7	Relu	Relu	99,979.25
87-87-1	SGD	0.008	0.9	Relu	Relu	92,598.68
87-44-1	SGD	0.008	0.5	Relu	Sigmoid	168,133.37
87-87-1	SGD	0.008	0.7	Relu	Sigmoid	137,156.57
87-87-1	SGD	0.008	0.9	Relu	Sigmoid	108,401.45
87-22-1	SGD	0.004	0.5	Sigmoid	Relu	240,775.10
87-22-1	SGD	0.006	0.7	Sigmoid	Relu	221,463.88

จากตารางที่ 4.19 เมื่อสนใจจะเลือกตัวแบบโดยพิจารณาจากชุดข้อมูลทดสอบ (Test Set) และค่า MAD พบว่าโครงข่ายที่ให้ค่า MAD ต่ำที่สุด (MAD = 83,486.90) คือ 87-87-1 โดยใช้ Optimizer คือ Adam อัตราการเรียนรู้ (Learning Rate) เท่ากับ 0.001 และมีฟังก์ชันกระตุ้น (Activation Function) คือ Relu (Hidden Layer), Sigmoid (Output Layer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.20 แสดงโครงข่ายประสาทเทียมที่ดีที่สุดแต่ละ Optimizer และ Activation Function โดยสนใจที่ชุดข้อมูลทวนสอบ (Validation Set) และ MAPE

โครงข่าย	Optimizer	อัตราการเรียนรู้	โมเมนตัม	Activation Function		MAPE
				ชั้นซ่อน	ชั้นเอาท์พุต	
87-44-1	Adam	0.001	-	Sigmoid	Sigmoid	22.27
87-22-1	Adam	0.002	-	Relu	Relu	35.94
87-44-1	Adam	0.002	-	Relu	Sigmoid	24.15
87-87-1	Adam	0.001	-	Sigmoid	Relu	24.67
87-87-1	SGD	0.008	0.5	Sigmoid	Sigmoid	159.66
87-22-1	SGD	0.002	0.7	Sigmoid	Sigmoid	162.62
87-44-1	SGD	0.008	0.9	Sigmoid	Sigmoid	109.22
87-44-1	SGD	0.008	0.5	Relu	Relu	44.43
87-44-1	SGD	0.004	0.7	Relu	Relu	39.49
87-87-1	SGD	0.006	0.9	Relu	Relu	34.03
87-87-1	SGD	0.008	0.5	Relu	Sigmoid	69.50
87-22-1	SGD	0.008	0.7	Relu	Sigmoid	54.10
87-87-1	SGD	0.008	0.9	Relu	Sigmoid	45.86
87-22-1	SGD	0.004	0.5	Sigmoid	Relu	58.78
87-22-1	SGD	0.006	0.7	Sigmoid	Relu	55.75

จากตารางที่ 4.20 เมื่อสนใจจะเลือกตัวแบบโดยพิจารณาจากชุดข้อมูลทวนสอบ (Validation Set) และค่า MAPE พบว่าโครงข่ายที่ให้ค่า MAPE ต่ำที่สุด (MAPE = 22.27) คือ 87-44-1 โดยใช้ Optimizer คือ Adam อัตราการเรียนรู้ (Learning Rate) เท่ากับ 0.001 และมีฟังก์ชันกระตุ้น (Activation Function) คือ Sigmoid (Hidden Layer), Sigmoid (Output Layer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.21 แสดงโครงข่ายประสาทเทียมที่ดีที่สุดแต่ละ Optimizer และ Activation Function โดยสนใจที่ชุดข้อมูลทวนสอบ (Validation Set) และ MAD

โครงข่าย	Optimizer	อัตราการเรียนรู้	โมเมนตัม	Activation Function		MAD
				ชั้นซ่อน	ชั้นเอาท์พุต	
87-44-1	Adam	0.001	-	Sigmoid	Sigmoid	79,502.95
87-22-1	Adam	0.002	-	Relu	Relu	108,682.46
87-44-1	Adam	0.002	-	Relu	Sigmoid	103,634.06
87-87-1	Adam	0.001	-	Sigmoid	Relu	78,808.07
87-87-1	SGD	0.008	0.5	Sigmoid	Sigmoid	256,026.36
87-22-1	SGD	0.002	0.7	Sigmoid	Sigmoid	255,707.82
87-87-1	SGD	0.008	0.9	Sigmoid	Sigmoid	182,571.39
87-44-1	SGD	0.008	0.5	Relu	Relu	131,744.00
87-44-1	SGD	0.004	0.7	Relu	Relu	116,001.37
87-22-1	SGD	0.004	0.9	Relu	Relu	100,801.83
87-87-1	SGD	0.008	0.5	Relu	Sigmoid	143,986.38
87-22-1	SGD	0.008	0.7	Relu	Sigmoid	137,437.65
87-22-1	SGD	0.008	0.9	Relu	Sigmoid	115,538.25
87-44-1	SGD	0.008	0.5	Sigmoid	Relu	205,514.92
87-44-1	SGD	0.008	0.7	Sigmoid	Relu	189,523.20

จากตารางที่ 4.21 เมื่อสนใจจะเลือกตัวแบบโดยพิจารณาจากชุดข้อมูลทวนสอบ (Validation Set) และค่า MAD พบว่าโครงข่ายที่ให้ค่า MAD ต่ำที่สุด (MAD = 78,808.07) คือ 87-87-1 โดยใช้ Optimizer คือ Adam อัตราการเรียนรู้ (Learning Rate) เท่ากับ 0.001 และมีฟังก์ชันกระตุ้น (Activation Function) คือ Sigmoid (Hidden Layer), Relu (Output Layer)

โดยผู้จัดทำได้ปรึกษากับทางตัวแทนของบริษัท ได้ข้อสรุปว่าจะพิจารณาตัวแบบจากค่า MAPE ของชุดข้อมูลทดสอบ (Test Set) ซึ่งตัวแบบพิจารณาจากตารางที่ 4.18 จะพบว่าตัวแบบที่เหมาะสมที่สุดได้กำหนดค่าไฮเปอร์พารามิเตอร์ดังตารางที่ 4.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.22 แสดงรายละเอียดของตัวแบบที่เหมาะสมที่สุดที่ตัวแทนของบริษัทเลือก โดยพิจารณาจากค่า MAPE ของชุดข้อมูลทดสอบ (Test Set)

โครงสร้างของตัวแบบพยากรณ์		87-44-1
Optimizer		Adam
Dropout		0.1
ฟังก์ชันสูญเสีย (Loss Function)		MSE
อัตราการเรียนรู้ (Learning Rate)		0.006
β_1		0.9
β_2		0.99
ฟังก์ชันกระตุ้น (Activation Function)	ชั้นซ่อน (Hidden Layer)	Sigmoid
	ชั้นเอาท์พุต (Output Layer)	Sigmoid
จำนวนรอบการเรียนรู้ (Epoch)		200
MAPE	ข้อมูลชุดฝึกฝน (Training Set)	23.83
	ข้อมูลชุดทดสอบ (Test Set)	28.31
	ข้อมูลชุดทวนสอบ (Validation Set)	42.78

จากตารางที่ 4.22 จะพบว่าตัวแบบที่เหมาะสมที่สุดจากการพิจารณาค่า MAPE ของชุดข้อมูลทดสอบ (Test Set) คือโครงข่ายประสาทเทียม 87-44-1 โดยมีค่าไฮเปอร์พารามิเตอร์ดังนี้ Optimizer คือ Adam, Dropout เท่ากับ 0.1, ฟังก์ชันสูญเสีย (Loss Function) คือ Mean Square Error (MSE), อัตราการเรียนรู้ (Learning Rate) เท่ากับ 0.006, β_1 เท่ากับ 0.9, β_2 เท่ากับ 0.99, ฟังก์ชันกระตุ้น (Activation Function) คือ Sigmoid (ชั้นซ่อน), Sigmoid (ชั้นเอาท์พุต) และจำนวนรอบการเรียนรู้ (Epoch) เท่ากับ 200 ซึ่งตัวแบบที่ที่เหมาะสมนี้จะนำไปพิจารณาต่อในการเปรียบเทียบ เพื่อแทนที่การพยาพยากรณ์ยอดขายสินค้าในแต่ละพื้นที่การขาย (Sale Forecast)

4.2 การเปรียบเทียบประสิทธิภาพของตัวแบบพยากรณ์

นำตัวแบบการพยากรณ์ยอดขายสินค้า โดยวิธีโครงข่ายประสาทเทียมที่ดีที่สุด ดังตารางที่ 4.22 มาเปรียบเทียบกับการพยากรณ์ยอดขายสินค้าของแต่ละพื้นที่การขาย (Sale Forecast) โดยการพยากรณ์สินค้าทั้งสองมี MAPE และ MAD ดังตารางที่ 4.23

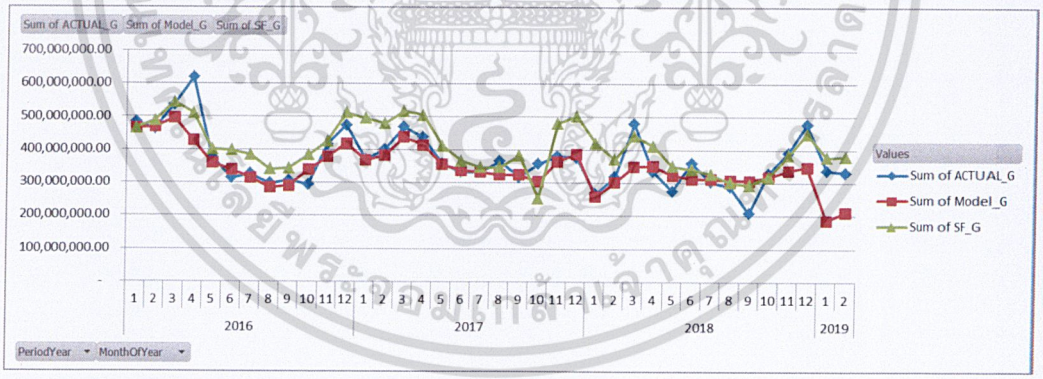
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.23 เปรียบเทียบประสิทธิภาพของการพยากรณ์โดยใช้ข้อมูลชุดทดสอบ

วิธีการพยากรณ์	ค่าวัดความถูกต้องของการพยากรณ์	
	MAPE	MAD
วิธีพยากรณ์ยอดขายของแต่ละพื้นที่การขาย	52.4	150,018.63
วิธีโครงข่ายประสาทเทียม	28.31	87,665.06

จากตารางที่ 4.23 พบว่าการพยากรณ์ยอดขายสินค้าของแต่ละพื้นที่การขาย (Sale Forecast) มีค่า MAPE เท่ากับ 52.4 และค่า MAD เท่ากับ 150,018.63 และตัวแบบการพยากรณ์ยอดขายสินค้าโดยวิธีโครงข่ายประสาทเทียม มีค่า MAPE เท่ากับ 28.31 และค่า MAD เท่ากับ 87,665.06

จากการอธิบายตารางที่ 4.23 จะพบว่าค่า MAPE และ MAD ที่ได้จากวิธีพยากรณ์ยอดขายของแต่ละพื้นที่การขาย (Sale Forecast) มีค่าเท่ากับ 52.4 และ 150018.63 ตามลำดับ และตัวแบบการพยากรณ์ยอดขาย โดยวิธีโครงข่ายประสาทเทียมให้ค่า MAPE และ MAD เท่ากับ 28.31 และ 87665.06 ตามลำดับ จากผลที่ได้พบว่าตัวแบบการพยากรณ์ยอดขายสินค้า โดยวิธีโครงข่ายประสาทเทียมให้ค่า MAPE และ MAD ต่ำกว่าวิธีพยากรณ์ของทางบริษัท ผลการเปรียบเทียบยอดขายจริง การพยากรณ์ยอดขายสินค้า โดยวิธีโครงข่ายประสาทเทียม และการพยากรณ์ยอดขายสินค้าของแต่ละพื้นที่การขาย (Sale Forecast) แสดงดังรูปที่ 4.1



รูปที่ 4.1 กราฟเปรียบเทียบยอดขายจริง (สีน้ำเงิน) ตัวแบบการพยากรณ์ยอดขายสินค้า โดยวิธีโครงข่ายประสาทเทียม (สีแดง) และการพยากรณ์ยอดขายสินค้าของแต่ละพื้นที่การขาย (Sale Forecast) (สีเขียว)

จากรูปที่ 4.1 แกน X คือ ช่วงเวลา ณ มกราคม พ.ศ. 2559 ถึง กุมภาพันธ์ พ.ศ. 2562 แกน Y คือ ปริมาณยอดขายจริง (Actual) จะพบว่าการพยากรณ์ยอดขายสินค้า โดยวิธีโครงข่ายประสาทเทียม และการพยากรณ์ยอดขายสินค้าของแต่ละพื้นที่การขาย (Sale Forcast) มีค่าที่ใกล้เคียงกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยอดขายจริง จะมีเพียงแค่บางเดือนที่แตกต่างกันอย่างเห็นได้ชัด แต่ในภาพรวมแล้วนั้นการพยากรณ์ยอดขายสินค้า โดยวิธีโครงข่ายประสาทเทียมใกล้เคียงกับยอดขายจริงมากกว่าการพยากรณ์ยอดขายสินค้าของแต่ละพื้นที่การขาย (Sale Forcast)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการวิจัย การอภิปรายผล และข้อเสนอแนะ

5.1 สรุปผลการวิจัย

ในการทำสหกิจศึกษาครั้งนี้ได้ทำการพัฒนาตัวแบบการพยากรณ์ยอดขายสินค้า โดยวิธีโครงข่ายประสาทเทียม เพื่อนำมาแทนที่การพยากรณ์ยอดขายสินค้าของแต่ละพื้นที่การขาย (Sale Forecast) โดยข้อมูลที่ได้ทำการศึกษาอยู่ในช่วง มกราคม พ.ศ. 2558 - กุมภาพันธ์ พ.ศ. 2562 และได้แบ่งข้อมูลเป็น 3 ส่วนโดยการสุ่ม ส่วนที่ 1 คือชุดข้อมูลฝึกฝน (Training Set) พ.ศ. 2559 – พ.ศ. 2561 ร้อยละ 70, ส่วนที่ 2 คือชุดข้อมูลทดสอบ (Test Set) พ.ศ. 2559 - พ.ศ. 2561 ร้อยละ 30 และส่วนที่ 3 คือชุดข้อมูลทวนสอบ (Validation Set) มกราคม พ.ศ. 2562 ถึงกุมภาพันธ์ พ.ศ. 2562 นอกจากนี้บริษัทได้ตัดสินใจเลือกตัวแบบจากชุดข้อมูลทดสอบ (Test Set) ที่มีค่า MAPE น้อยที่สุด

โครงข่ายประสาทเทียมที่เหมาะสมที่สุด คือโครงข่าย 87-44-1 โดยเป็นโครงข่ายแบบเพอร์เซ็ปตรอนหลายชั้น (Multi-Layer Perceptron) มี Optimizer คือ Adam, Dropout เท่ากับ 0.1, ฟังก์ชันสูญเสีย (Loss Function) คือ Mean Square Error (MSE), อัตราการเรียนรู้ (Learning Rate) เท่ากับ 0.006, β_1 เท่ากับ 0.9, β_2 เท่ากับ 0.99, ฟังก์ชันกระตุ้น (Activation Function) คือ Sigmoid (ชั้นซ่อน), Sigmoid (ชั้นเอาท์พุต) และจำนวนรอบการเรียนรู้ (Epoch) เท่ากับ 200 โดยชุดข้อมูลทดสอบ (Test Set) มีค่า MAPE เท่ากับ 28.31 และค่า MAD เท่ากับ 87,665.06

ผลการศึกษาพบว่าตัวแบบโครงข่ายประสาทเทียมมีค่า MAPE เท่ากับ 28.31 ซึ่งน้อยกว่าตัวแบบของบริษัทที่พยากรณ์ที่มีค่า MAPE เท่ากับ 52.4 โดยสามารถสรุปได้ว่าตัวแบบโครงข่ายประสาทเทียมเป็นตัวแบบที่มีประสิทธิภาพมากกว่าตัวแบบพยากรณ์ของบริษัท

5.2 การอภิปรายผล

ในการวิจัยครั้งนี้ได้กำหนดวัตถุประสงค์ไว้ 2 ข้อ ดังนี้

1) เพื่อสร้างตัวแบบที่เหมาะสมสำหรับการพยากรณ์สินค้า โดยใช้วิธีโครงข่ายประสาทเทียม

2) นำตัวแบบที่เหมาะสมมาแทนที่การพยากรณ์ยอดขายสินค้าของแต่ละพื้นที่ (Sale Forecast) เพื่อเพิ่มความแม่นยำในการพยากรณ์ยอดขายสินค้าให้กับบริษัท

ในกรณีวัตถุประสงค์ข้อที่ 1 ได้สร้างตัวแบบจากอัลกอริทึม (Algorithm) โครงข่ายประสาทเทียม (Neural Network) แบบเพอร์เซ็ปตรอนหลายชั้น (Multi-Layer Perceptron) ซึ่งเป็นหนึ่งในอัลกอริทึมที่มีประสิทธิภาพในการวิเคราะห์ข้อมูลที่ซับซ้อน โดยเลือก Optimizer และฟังก์ชันกระตุ้น (Activation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Function) ซึ่งเป็นวิธีที่ได้รับการยอมรับในปัจจุบัน (Mr.P L, 2019) มาใช้ในการทดลองหาค่าไฮเปอร์พารามิเตอร์ที่ทำให้ผลพยากรณ์ออกมาแม่นยำที่สุด และในการวิจัยในครั้งนี้ได้ตัวแบบที่เหมาะสมที่สุดให้ค่า MAPE เท่ากับ 28.31

ในกรณีวัตถุประสงค์ข้อที่ 2 ได้ทำการนำตัวแบบโครงข่ายประสาทเทียมที่เหมาะสมที่สุดไปเปรียบเทียบกับการพยากรณ์ยอดขายสินค้าของแต่ละพื้นที่การขาย (Branch Code) ได้ผลว่าตัวแบบโครงข่ายประสาทเทียมให้ผลการพยากรณ์ที่แม่นยำกว่า ทั้งนี้การนำไปใช้นั้น ทางบริษัทจะทำการประเมินตัวแบบอีก 2-3 เดือน เพื่อเพิ่มความเชื่อมั่นในการนำไปใช้แทนการพยากรณ์ยอดขายสินค้าของแต่ละพื้นที่การขาย (Branch Code)

5.3 ข้อเสนอแนะ

ตัวแบบการพยากรณ์ยอดขายสินค้า โดยวิธีโครงข่ายประสาทเทียม อาจสามารถเพิ่มประสิทธิภาพให้ตัวแบบได้อีก โดยผู้วิจัยขอเสนอแนะดังนี้

1) หากสามารถเพิ่มข้อมูลตัวแปรที่มีผลต่อปริมาณยอดขายจริง (Actual) เช่น จำนวนสินค้าในคลังสินค้าของลูกค้า (Stock) เพราะว่าการที่ลูกค้ามีจำนวนสินค้าอยู่ในคลังสินค้าจำนวนมากส่งผลให้ลูกค้าสั่งสินค้าน้อยลง

2) การเปลี่ยนวิธีการแทนค่าสูญหาย (Missing Value) เช่น การแทนค่าสูญหาย (Missing Value) ด้วยวิธีสมการถดถอยเชิงเส้น (ทัตดา หิรัญพต, 2554)

3) การเพิ่มจำนวนข้อมูลที่นำมาวิเคราะห์ และเป็นข้อมูลที่มีคุณภาพ เช่น การเพิ่มปี 2015 เข้าไปในการวิเคราะห์สร้างตัวแบบโดยเป็นข้อมูลที่มีคุณภาพ หมายถึงข้อมูลที่ปราศจากค่าสูญหาย (Missing Value) และค่าผิดปกติ (Outlier)

4) การเปลี่ยนวิธีการแปลงข้อมูล (Data Tranformations) เช่น การแปลงข้อมูล (Data Tranformation) ด้วยวิธี Z-Score มาใช้ในการเปรียบเทียบเพื่อหาวิธีการแปลงข้อมูล (Data Tranformation) ที่เหมาะสมที่สุด

5) การเปลี่ยนอัลกอริทึม (Algorithm) เช่น การนำ Long/Short Term Memory (LSTM) มาใช้ในการเปรียบเทียบเพื่อหาอัลกอริทึม (Algorithm) ที่เหมาะสมกับข้อมูลมากที่สุดในการวิจัยครั้งต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

เกียรติชัย เสนา. 2551. **"ดัชนีสมรรถนะระบบและการวิเคราะห์คอนตินเจนซีโดยใช้โครงข่ายประสาทเทียมเพอร์เซ็ปตรอนหลายชั้น"** วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย, จุฬาลงกรณ์มหาวิทยาลัย.

เพชรนรินทร์ แก้วหล้า. 2553. **"ระบบการพยากรณ์จำนวนผู้ป่วยโรคหัวใจโดยใช้โครงข่ายประสาทเทียม"** วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ.

ไววิทย์ พานิชอัศดร และ มหศักดิ์ เกตุฉ่ำ. 2558. "การพยากรณ์ยอดขายปลีกแก๊สรถยนต์ ด้วยวิธีการวิเคราะห์อนุกรมเวลาและโครงข่ายประสาทเทียม." **JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY.** 7(1) : 42-49.

กัลยา วานิชย์บัญชา. 2544. **หลักสถิติ.** กรุงเทพมหานคร : โรงพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย.

คงฤทธิ์ โกมาสถิตย์. 2555. **"การพยากรณ์อุปสงค์พลังงานไฟฟ้าของประเทศไทยในระยะยาวด้วยวิธีโครงข่ายประสาทเทียม"** วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมอุตสาหการ บัณฑิตวิทยาลัย, จุฬาลงกรณ์มหาวิทยาลัย.

ณัฐพล รองศรีแย้ม. 2542. **"การพยากรณ์โหลดหม้อแปลงโดยใช้เครือข่ายประสาทเทียม"** วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย, จุฬาลงกรณ์มหาวิทยาลัย.

ทัตดา หิรัญพต. 2554. **"การเปรียบเทียบประสิทธิภาพระหว่างวิธีการประมาณข้อมูลสูญหายด้วยค่าถดถอยและวิธีการประมาณข้อมูลสูญหายด้วยค่าถดถอยแบบสโทแคสติก"** ปัญหาพิเศษวิทยาศาสตร์บัณฑิต สาขาวิชาสถิติ บัณฑิตวิทยาลัย, มหาวิทยาลัยบูรพา.

ธนาวุฒิ ประกอบผล. 2552. "โครงข่ายประสาทเทียม." **วารสาร มฉก.วิชาการ.** 12(24) : 73-87.

นิภา นิรุตติกุล. 2549. **การพยากรณ์การขาย.** กรุงเทพมหานคร : สำนักพิมพ์มหาวิทยาลัยเกษตรศาสตร์.

ผุสดี บุญรอด และ กรวัฒน์ พลเยี่ยม. 2558. "แบบจำลองการพยากรณ์ราคามันสำปะหลัง โดยใช้โครงข่ายประสาทเทียมแบบหลายชั้น." **วารสารวิทยาศาสตร์และเทคโนโลยี.** 25(3) : 533-543.

พยุง มีสัจ. 2555. **"ระบบฟัซซีและโครงข่ายประสาทเทียม."** [E-book]. กรุงเทพมหานคร : คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยพระจอมเกล้าพระนครเหนือ.

พรรณิภา คุ้มสิน. 2560. **"การเปรียบเทียบตัวแบบการพยากรณ์ยอดขายเครื่องปรับอากาศ โดยวิธีปรับให้เรียบเอ็กซ์โพเนนเชียลแบบโฮลท์และวินเทอร์ วิธีบ็อกซ์-เจนกินส์ และวิธีโครงข่ายประสาทเทียม"** วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิชาสถิติประยุกต์ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มนัสกานต์ เสน่หา. 2559. **"การทำนายอายุการใช้งานคงเหลือของเครื่องจักรด้วยเน็ตเวิร์กคอนโวลูชันเชิงลึกที่เพิ่มประสิทธิภาพ"** วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ บัณฑิตวิทยาลัย, จุฬาลงกรณ์มหาวิทยาลัย.

วชิราภรณ์ แก้วมาตย์ และ สุรชัย จันทร์จรัส. 2556. "การใช้โครงข่ายประสาทเทียมในการพยากรณ์ดัชนีราคาหลักทรัพย์." **วารสารวิจัย มข. มส. (บศ.).** 1(1) : 108-122.

วรรณพงษ์ ภัททิยไพบูลย์. 2559. **เริ่มต้น Neural Networks กับ Python.** [Online]. Available : https://python3.wannaphong.com/2016/04/neural-networks-python.html (เข้าถึงเมื่อวันที่ 20 เมษายน 2562).

สุพรรษา เหลืองอำนวยศิริ. 2551. **"การศึกษาเปรียบเทียบตัวแบบการพยากรณ์ปริมาณการส่งออกแป้งมันสำปะหลัง โดยวิธีการวิเคราะห์การถดถอยเชิงเส้นแบบพหุ และวิธีโครงข่ายประสาทเทียม"** วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาสถิติประยุกต์ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

อัจฉรา จันทร์ฉาย. 2544. **การพยากรณ์เพื่อตัดสินใจทางธุรกิจ.** กรุงเทพมหานคร : สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย.

อาทิตย์ อินทวี. 2551. **"การพยากรณ์สึนามิโดยวิธีโครงข่ายประสาทเทียม"** วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมโยธา บัณฑิตวิทยาลัย, จุฬาลงกรณ์มหาวิทยาลัย.

อำภา สาระศิริ. 2559. **เทคนิคการเรียนรู้พื้นฐานโครงข่ายประสาทเทียม.** [Online]. Available : http://www.mut.ac.th/research-detail-92 (เข้าถึงเมื่อวันที่ 20 เมษายน 2562).

Eakasit Pacharawongsakda, Ph.D. 2558. **Prectical Data Mining with RapidMiner Studio 6.** พิมพ์ครั้งที่ 34. กรุงเทพมหานคร : สำนักพิมพ์เอเชีย ดิจิตอลการพิมพ์ จำกัด.

Fausett, L. 1994. **Fundamentals of Neural Networks.** New jersey : Prentice Hall, Inc.

H.R. Maier, and G.C. Dandy. 2000. "Neural Networks for the Prediction and Forecasting of Water Resources Variables: a Review of Modeling Issues and Applications." **Environmental Modeling & Software.** 15: 101-124.

Hagan, M., Demuth, H., and Beale, M. 1996. **Neural Network Design.** University of Colorado Bookstore : United States of America.

Jean David Caprace. 2007. **A Data Mining Analysis Applied to a Straightening Process Database.** [Online]. Available : https://www.researchgate.net/figure/CRISP-DM-methodology_fig1_258835132 (เข้าถึงเมื่อวันที่ 25 เมษายน 2562).

Kanchan Sarkar. 2018. **ReLU : Not a Differentiable Function : Why used in Gradient Based**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Optimization? and Other Generalizations of ReLU. [Online]. Available : https://medium.com/@kanchansarkar/relu-not-a-differentiable-function-why-used-in-gradient-based-optimization-7fef3a4cecec (เข้าถึงเมื่อวันที่ 25 เมษายน 2562).

Kishan Maladkar. 2018. **Types Of Activation Functions In Neural Networks And Retionale Behind It.** [Online]. Available : https://www.analyticsindiamag.com/most-common-activation-functions-in-neural-networks-and-rationale-behind-it/ (เข้าถึงเมื่อวันที่ 25 เมษายน 2562).

Masters, T. 1993. **Practical Neural Network Recipes in C++.** California : Academic Press, Inc.

Mr.P L. 2019. **Deep Learning แบบฉบับสามัญชน EP 2 Optimization & Activation Function เรียนกันสบายๆสไตล์ชิลๆ.** [Online]. Available : https://bit.ly/2WpyO6X. (เข้าถึงเมื่อวันที่ 27 เมษายน 2562)

Praveen Ranjan Srivastava. 2010. **Test Effort Estimation Using Neural Network.** [Online]. Available : https://www.researchgate.net/figure/Tanh-function_fig4_220204237 (เข้าถึงเมื่อวันที่ 28 เมษายน 2562).

Rinat Maksutov. 2018. **Deep study of a not very deep neural network. Part 5 : Dropout and Noise.** [Online]. Available : https://medium.com/@maksutov.rn/deep-study-of-a-not-very-deep-neural-network-part-5-dropout-and-noise-29d980ece933 (เข้าถึงเมื่อวันที่ 28 เมษายน 2562).

S.Arun Balaji and K.Baskaran. 2013. "DESIGN AND DEVELOPMENT OF ARTIFICIAL NEURAL NETWORKING (ANN) SYSTEM USING SIGMOID ACTIVATION FUNCTION TO PREDICT ANNUAL RICE PRODUCTION IN TAMILNADU." International Journal of Computer Science, Engineering and Information Technology (IJCSEIT). 3(1) : 13-31.

S.V.G. Reddy, K. Thammi Reddy and V. ValliKumari. 2018. "Optimization of Deep Learning using various Optimizers, Loss functions and Dropout." International Journal of Recent Technology and Engineering (IJRTE). 7(4S2) : 448-455.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

ข้อมูลที่ใช้ในการพยากรณ์ยอดขายสินค้า

ตารางที่ ก.1 ข้อมูลยอดขาย เดือนมกราคม 2559 – เดือนกุมภาพันธ์ 2562

เดือน	พ.ศ.			
	2559	2560	2561	2562
ม.ค.	566,084,060.55	443,403,170.32	344,520,435.59	413,107,295.69
ก.พ.	562,605,597.90	489,486,471.07	406,050,770.82	409,629,917.06
มี.ค.	669,502,146.21	565,737,846.74	548,321,287.98	93,555,310.69
เม.ย.	669,350,940.50	518,707,066.91	422,080,838.12	-
พ.ค.	436,233,548.43	423,517,774.40	375,378,627.78	-
มิ.ย.	380,720,008.12	401,441,053.87	393,439,859.52	-
ก.ค.	394,361,977.72	400,586,590.92	342,570,875.03	-
ส.ค.	359,009,007.41	439,622,622.66	344,259,824.98	-
ก.ย.	371,349,193.15	394,601,234.98	255,402,584.47	-
ต.ค.	356,610,400.37	402,729,308.23	401,308,242.32	-
พ.ย.	502,851,438.77	466,494,645.36	470,216,809.50	-
ธ.ค.	571,021,781.09	466,962,596.10	570,235,961.52	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

คำสั่งที่ใช้ในการสร้างตัวแบบพยากรณ์ยอดขายสินค้า

ตารางที่ ข.1 คำสั่งที่ใช้ในการเตรียมข้อมูลเพื่อสร้างตัวแบบพยากรณ์ยอดขายสินค้า

```
#บังคับให้ประมวลผลบน CPU เครื่องคอมพิวเตอร์
import os
os.environ["CUDA_DEVICE_ORDER"]="PCI_BUS_ID"
os.environ["CUDA_VISIBLE_DEVICES"]=""

#นำเข้าไลบรารีที่จำเป็นในการเตรียมข้อมูล
import pandas as pd
import numpy  as np

#นำเข้าข้อมูลที่เตรียมไว้สำหรับพัฒนาตัวแบบพยากรณ์ยอดขายสินค้า
df_Product=pd.read_csv('File/SF_Branch_Month.csv')

#ขั้นตอนการคัดเลือกข้อมูล (Select Data)
#กรองให้เหลือเฉพาะข้อมูลที่จำเป็นต่อการพัฒนาตัวแบบพยากรณ์ยอดขายสินค้า
df_Product=df_Product[~((df_Product.PeriodYear==2019)&(df_Product.MonthOfYear
>=3))]
df_Product=df_Product[(df_Product['Product_Brand']=='A')&
(df_Product['FG_PackageFormat']=='AA')]
df_Product=df_Product.drop(columns=['Product_Brand','FG_PackageFormat',
'Region'])
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ข.1 คำสั่งที่ใช้ในการเตรียมข้อมูลเพื่อสร้างตัวแบบพยากรณ์ยอดขายสินค้า (ต่อ)

```
#สร้างตารางตัวแบบการพยากรณ์เดิมของบริษัท (Sale Forecast) เพื่อนำไปเปรียบเทียบกับตัวแบบโครงข่ายประสาทเทียมในภายหลัง
df_ProductSF=df_Product.pivot_table(index=['PeriodYear','MonthOfYear'],columns=['BranchCode'],values='SF_Volumn',aggfunc=np.sum)
df_ProductSF=df_ProductSF.reset_index()
df_ProductSF=pd.melt(df_ProductSF,id_vars=['PeriodYear','MonthOfYear'])
df_ProductSF.rename(columns={'value':'SF_Volumn'},inplace=True)

#เนื่องจากตัวข้อมูลที่ได้รับมามีค่าสูญหาย (Missing Value) จึงแยกทำความสะอาดข้อมูล (Data Cleaning) ที่มีค่าสูญหาย (Missing Value)
#ทำความสะอาดข้อมูลยอดขายจริง (Data Actual Cleaning)
#จัดการให้เหลือเฉพาะพื้นที่การขาย (Branch Code) ที่มีค่าสูญหาย (Missing Value) น้อยกว่า 20%
#หลังจากนั้นแทนที่ค่าสูญหายด้วยค่าเฉลี่ยของแต่ละพื้นที่การขาย (Branch Code)
df_ProductAC=df_Product.dropna(subset=['ACTUAL'])
df_ProductAC=df_ProductAC.pivot_table(index=['PeriodYear','MonthOfYear'],columns=['BranchCode'],values='ACTUAL',aggfunc=np.sum)
df_ProductAC=df_ProductAC[df_ProductAC.columns[df_ProductAC.isin(['NaN']).sum()/len(df_ProductAC)*100<20]]
df_ProductAC=df_ProductAC.fillna(df_ProductAC.mean()[:])
df_ProductAC=df_ProductAC.reset_index()
df_ProductAC=pd.melt(df_ProductAC,id_vars= ['PeriodYear','MonthOfYear'])
df_ProductAC.rename(columns={'value':'ACTUAL'},inplace=True)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ข.1 คำสั่งที่ใช้ในการเตรียมข้อมูลเพื่อสร้างตัวแบบพยากรณ์ยอดขายสินค้า (ต่อ)

```
#ทำความสะอาดข้อมูลการประมาณยอดขายที่ปรับ Budget (LE) (Data LE Cleaning)
#จัดการให้เหลือเฉพาะพื้นที่การขาย (Branch Code) ที่มีค่าสูญหาย (Missing Value) น้อยกว่า 20%
#หลังจากนั้นแทนที่ค่าสูญหายด้วยค่าเฉลี่ยของแต่ละพื้นที่การขาย (Branch Code)
df_ProductLE=df_Product.dropna(subset=['LE'])
df_ProductLE=df_ProductLE.pivot_table(index=['PeriodYear','MonthOfYear'],columns=[
'BranchCode'],values='LE',aggfunc=np.sum)
df_ProductLE=df_ProductLE.fillna(df_ProductLE.mean()[:])
df_ProductLE=df_ProductLE.reset_index()
df_ProductLE=pd.melt(df_ProductLE,id_vars=['PeriodYear','MonthOfYear'])
df_ProductLE.rename(columns={'value':'LE'},inplace=True)

#รวมตารางที่แยกทำความสะอาดข้อมูล (Data Cleaning) เพื่อนำไปวิเคราะห์ต่อ
df=pd.merge(df_ProductAC,df_ProductLE,how='left',on=['BranchCode','PeriodYear',
'MonthOfYear'])
df=pd.merge(df,df_ProductSF,how='left',on=['BranchCode','PeriodYear',
'MonthOfYear'])
df=df.dropna(axis=0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ข.1 คำสั่งที่ใช้ในการเตรียมข้อมูลเพื่อสร้างตัวแบบพยากรณ์ยอดขายสินค้า (ต่อ)

```
#ผู้จัดทำพิจารณาว่าข้อมูลย้อนหลัง 1 ปี (Last Year) น่าจะมีผลต่อการวิเคราะห์ของตัวแบบ เนื่องจาก
ยอดขายในแต่ละปีมีรูปแบบที่คล้ายคลึงกัน
#สร้างข้อมูลยอดขายย้อนหลัง 1 ปี (Last Year) โดยอ้างอิงจากข้อมูลยอดขายจริงที่ผ่านการทำความ
สะอาดข้อมูล (Data Cleaning) แล้ว
df_Time=df_ProductAC.copy()
df_Time['NextYear']=df_Time.PeriodYear.replace([2015,2016,2017,2018,2019],
[2016,2017,2018,2019,2020])
df_Time.drop(['PeriodYear'],axis=1,inplace=True)
df_Time=df_Time.rename(columns={'ACTUAL':'RENAME'})
df=pd.merge(df,df_Time,how='left',left_on=['PeriodYear','MonthOfYear',
'BranchCode'],right_on = ['NextYear', 'MonthOfYear','BranchCode'])
df=df[['PeriodYear','MonthOfYear','BranchCode','ACTUAL','LE','SF_Volumn',
'RENAME']]
df=df.rename(columns={'RENAME':'last year'})
df=df[~(df['PeriodYear']==2015)].reset_index(drop=True)

#สร้างข้อมูล MAPE และ MAD ของตัวแบบการพยากรณ์ยอดขายสินค้าแต่ละพื้นที่ (Sale Forcast) เพื่อ
นำไปเปรียบเทียบกับ MAPE และ MAD ของตัวแบบโครงข่ายประสาทเทียมในภายหลัง
df['MAPE_SF']=abs((df.ACTUAL-df.SF_Volumn)/df.ACTUAL)*100
df['MAD_SF']=abs(df.ACTUAL-df.SF_Volumn)

#แปลงข้อมูลพื้นที่การขาย (Branch Code) ให้เป็นตัวแปรดัมมี่ (Dummy Variable) คือมีค่าแค่ 0 กับ 1
เนื่องจากเป็นข้อมูลคุณภาพ เพื่อทำให้อัลกอริทึม (Algorithm) โครงข่ายประสาทเทียมสามารถนำไป
วิเคราะห์ต่อได้
dummiesBranchCode=pd.get_dummies(df['BranchCode'])
df=pd.concat([df,dummiesBranchCode],axis='columns')

#ทำการแบ่งข้อมูลออกเป็น 2 ส่วน โดยแยกปี 2019 ออกมาเพื่อทำชุดทวนสอบ
df_2019=df[(df.PeriodYear==2019)]
df=df[~(df.PeriodYear==2019)]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ข.1 คำสั่งที่ใช้ในการเตรียมข้อมูลเพื่อสร้างตัวแบบพยากรณ์ยอดขายสินค้า (ต่อ)

```
#ส่วนปี 2016 - 2018 ทำการแบ่งส่วนเป็นชุดฝึกฝน 70% และชุดทดสอบ 30% โดยการสุ่ม
df_train=df.sample(frac=0.7)
df_test=df[~df.index.isin(df_train.index)]

#ทำการเอาข้อมูลที่ไม่จำเป็นออกก่อนจะนำไปสร้างตัวแบบ ทดสอบตัวแบบ และทวนสอบตัวแบบ
#แบ่งชุดข้อมูลออกเป็น 2 ส่วน คือ ชุดข้อมูลสำหรับให้ตัวแบบฝึกฝน (ขึ้นต้นด้วย X) และชุดข้อมูลสำหรับ
เฉลยตัวแบบ (ขึ้นต้นด้วย Y)
X_train=df_train.drop(['ACTUAL','SF_Volumn','MAPE_SF','MAD_SF'],axis=1)
.reset_index(drop=True)
X_test=df_test.drop(['ACTUAL','SF_Volumn','MAPE_SF','MAD_SF'],axis=1).reset_index
(drop=True)
Y_train=df_train[['ACTUAL']].reset_index(drop=True)
Y_test=df_test[['ACTUAL']].reset_index(drop=True)
Y_testCheck=df_test.reset_index(drop=True)
Y_trainCheck=df_train.reset_index(drop=True)
SF_2019=df_2019.copy()
X_2019=df_2019.drop(['ACTUAL','SF_Volumn','MAPE_SF','MAD_SF'],axis=1)
.reset_index(drop=True)
Y_2019=df_2019.reset_index(drop=True)

#ทำการหาค่ายอดขายต่ำสุด และยอดขายสูงสุด เพื่อนำไปใช้ในวิธี Min-Max Normalization
maxAC=Y_train.ACTUAL.max()
minAC=Y_train.ACTUAL.min()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ข.1 คำสั่งที่ใช้ในการเตรียมข้อมูลเพื่อสร้างตัวแบบพยากรณ์ยอดขายสินค้า (ต่อ)

```
#ทำการแปลงข้อมูลปริมาณ โดยวิธี Min-Max Normalization
#เอาข้อมูลที่ไม่ได้แปลงออกเพื่อลดความซ้ำซ้อนของข้อมูล
Y_train['ACTUAL_NORM']=(Y_train.ACTUAL-Y_train.ACTUAL.min())/(Y_train.ACTUAL.max()-
Y_train.ACTUAL.min())
Y_test['ACTUAL_NORM']=(Y_test.ACTUAL-Y_train.ACTUAL.min())/(Y_train.ACTUAL.max()-
Y_train.ACTUAL.min())
Y_train.drop(['ACTUAL'],axis=1,inplace=True)
Y_test.drop(['ACTUAL'],axis=1,inplace=True)

X_train.drop(['BranchCode'],axis=1,inplace=True)
X_test.drop(['BranchCode'],axis=1,inplace=True)
X_2019.drop(['BranchCode'],axis=1,inplace=True)

X_train['PeriodYear_NORM']=(X_train.PeriodYear–
X_train.PeriodYear.min())/(X_train.PeriodYear.max()-X_train.PeriodYear.min())
X_test['PeriodYear_NORM']=(X_test.PeriodYear–
X_train.PeriodYear.min())/(X_train.PeriodYear.max()-X_train.PeriodYear.min())
X_2019['PeriodYear_NORM']=(X_2019.PeriodYear–
X_train.PeriodYear.min())/(X_train.PeriodYear.max()-X_train.PeriodYear.min())
X_train.drop(['PeriodYear'],axis=1,inplace=True)
X_test.drop(['PeriodYear'],axis=1,inplace=True)
X_2019.drop(['PeriodYear'],axis=1,inplace=True)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ข.1 คำสั่งที่ใช้ในการเตรียมข้อมูลเพื่อสร้างตัวแบบพยากรณ์ยอดขายสินค้า (ต่อ)

```
X_train['MonthOfYear_NORM']=(X_train.MonthOfYear–
X_train.MonthOfYear.min())/(X_train.MonthOfYear.max()-X_train.MonthOfYear.min())
X_test['MonthOfYear_NORM']=(X_test.MonthOfYear–
X_train.MonthOfYear.min())/(X_train.MonthOfYear.max()-X_train.MonthOfYear.min())
X_2019['MonthOfYear_NORM']=(X_2019.MonthOfYear–
X_train.MonthOfYear.min())/(X_train.MonthOfYear.max()-X_train.MonthOfYear.min())
X_train.drop(['MonthOfYear'],axis=1,inplace=True)
X_test.drop(['MonthOfYear'],axis=1,inplace=True)
X_2019.drop(['MonthOfYear'],axis=1,inplace=True)

X_train['LE_NORM']=(X_train.LE–X_train.LE.min())/(X_train.LE.max()-X_train.LE.min())
X_test['LE_NORM']=(X_test.LE–X_train.LE.min())/(X_train.LE.max()-X_train.LE.min())
X_2019['LE_NORM']=(X_2019.LE–X_train.LE.min())/(X_train.LE.max()-X_train.LE.min())
X_train.drop(['LE'],axis=1,inplace=True)
X_test.drop(['LE'],axis=1,inplace=True)
X_2019.drop(['LE'],axis=1,inplace=True)

X_train['last year_NORM']=(X_train.LastYear–
X_train.LastYear.min())/(X_train.LastYear.max()-X_train.LastYear.min())
X_test['last year_NORM' ]=(X_test.LastYear–
X_train.LastYear.min())/(X_train.LastYear.max()-X_train.LastYear.min())
X_2019['last year_NORM' ]=(X_2019.LastYear –
X_train.LastYear.min())/(X_train.LastYear.max()-X_train.LastYear.min())
X_train.drop(['last year'],axis=1,inplace=True)
X_test.drop(['last year'],axis=1,inplace=True)
X_2019.drop(['last year'],axis=1,inplace=True)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ข.2 คำสั่งที่ใช้ในการสร้างตัวแบบ บันทึกโครงสร้างตัวแบบ และค่าของน้ำหนัก

```
#บังคับให้ประมวลผลบน CPU เครื่องคอมพิวเตอร์
import os
os.environ["CUDA_DEVICE_ORDER"]="PCI_BUS_ID"
os.environ["CUDA_VISIBLE_DEVICES"]=""

#นำเข้าไลบรารีที่จำเป็นในการสร้างตัวสร้าง
import pandas as pd
import numpy  as np
import matplotlib.pyplot as plt
from keras.models import Sequential
from keras.layers.core import Dense, Activation, Dropout
from keras import optimizers
%matplotlib inline
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ข.2 คำสั่งที่ใช้ในการสร้างตัวแบบ บันทึกโครงสร้างตัวแบบ และค่าของน้ำหนัก (ต่อ)

```
#นำเข้าข้อมูลทั้ง 3 ชุดที่ผ่านการเตรียมข้อมูลแล้ว สำหรับฝึกฝนตัวแบบ ทดสอบตัวแบบ และทวนสอบตัวแบบ
X_train=pd.read_csv('File/X_train.csv')
Y_train=pd.read_csv('File/Y_train.csv')
X_train.drop(['Unnamed: 0'],axis=1,inplace=True)
Y_train.drop(['Unnamed: 0'],axis=1,inplace=True)
X_test=pd.read_csv('File/X_test.csv')
Y_test=pd.read_csv('File/Y_test.csv')
X_test.drop(['Unnamed: 0'],axis=1,inplace=True)
Y_test.drop(['Unnamed: 0'],axis=1,inplace=True)
X_2019=pd.read_csv('File/X_2019.csv')
Y_2019=pd.read_csv('File/Y_2019.csv')
X_2019.drop(['Unnamed: 0'],axis=1,inplace=True)
Y_2019.drop(['Unnamed: 0'],axis=1,inplace=True)
df_testCheck=pd.read_csv('File/Y_testCheck.csv')
df_trainCheck=pd.read_csv('File/Y_trainCheck.csv')
df_testCheck.drop(['Unnamed: 0'],axis=1,inplace=True)
df_trainCheck.drop(['Unnamed: 0'],axis=1,inplace=True)

#นำเข้ายอดขายสูงสุด และยอดขายต่ำสุด ของชุดข้อมูลฝึกฝน
maxAC=3204910.2
minAC=606.98
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ข.2 คำสั่งที่ใช้ในการสร้างตัวแบบ บันทึกโครงสร้างตัวแบบ และค่าของน้ำหนัก (ต่อ)

```
#กำหนดอัลกอริทึม (Algorithm) และไฮเปอร์พารามิเตอร์ สำหรับการสร้างตัวแบบ
model=Sequential()
n_cols=X_train.shape[1]
model.add(Dense(44, activation='sigmoid', input_shape=(n_cols,)))
model.add(Dropout (0.1))
model.add(Dense(1, activation='sigmoid'))
Adam=optimizers.Adam(lr=0.006, beta_1=0.9, beta_2=0.999)
model.compile(optimizer=Adam, loss='mean_squared_error')
hist=model.fit(X_train, Y_train, epochs= 200,validation_data=(X_test,Y_test))

#พล็อกราฟฟังก์ชันสูญเสีย (Loss Function) ของชุดข้อมูลฝึกฝน และชุดข้อมูลทดสอบ เพื่อตรวจสอบตัว
แบบว่าเกิดการ Overfitting หรือไม่
plt.plot(hist.history['loss'])
plt.plot(hist.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='upper right')
plt.show()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ข.2 คำสั่งที่ใช้ในการสร้างตัวแบบ บันทึกโครงสร้างตัวแบบ และค่าของน้ำหนัก (ต่อ)

```
#นำผลลัพธ์ของข้อมูลทั้ง 3 ชุด มาสร้างตาราง เพื่อนำมาวิเคราะห์ในภายหลัง
yhatTest=model.predict(X_test)
df_predictTest=pd.DataFrame(yhatTest,columns=['yhatTest'])
df_predictTest=pd.concat([df_predictTest,df_testCheck],axis='columns')
df_predictTest['yhatTestTR']=df_predictTest['yhatTest']*(maxAC-minAC)+minAC
df_predictTest['MAPE']=abs((df_predictTest['ACTUAL']-
df_predictTest['yhatTestTR'])/df_predictTest['ACTUAL'])*100
df_predictTest['MAD']=abs((df_predictTest['ACTUAL']-df_predictTest['yhatTestTR']))
yhatTrain=model.predict(X_train)
df_predictTrain=pd.DataFrame(yhatTrain,columns=['yhatTrain'])
df_predictTrain=pd.concat([df_predictTrain,df_trainCheck],axis='columns')
df_predictTrain['yhatTrainTR']=df_predictTrain['yhatTrain']*(maxAC-minAC)+minAC
df_predictTrain['MAPE']=abs((df_predictTrain['ACTUAL']-
df_predictTrain['yhatTrainTR'])/df_predictTrain['ACTUAL'])*100
df_predictTrain['MAD']=abs((df_predictTrain['ACTUAL']-df_predictTrain['yhatTrainTR']))
yhat2019=model.predict(X_2019)
df_2019=pd.DataFrame(yhat2019,columns=['yhat2019'])
df_2019=pd.concat([df_2019,Y_2019],axis='columns')
df_2019['yhat2019TR']=df_2019['yhat2019']*(maxAC-minAC)+minAC
df_2019['MAPE']=abs((df_2019['ACTUAL']-df_2019['yhat2019TR'])
/df_2019['ACTUAL'])*100
df_2019['MAD']=abs((df_2019['ACTUAL']-df_2019['yhat2019TR']))

#นำ MAPE และ MAD ระหว่างตัวแบบโครงข่ายประสาทเทียม และการพยากรณ์สินค้าของแต่ละพื้นที่
การขาย (Sale Forecast) มาเปรียบเทียบกัน
df_predictTrain[['MAPE','MAD','MAPE_SF','MAD_SF']].mean()
df_predictTest[['MAPE','MAD','MAPE_SF','MAD_SF']].mean()
df_2019[['MAPE','MAD','MAPE_SF','MAD_SF']].mean()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ข.2 คำสั่งที่ใช้ในการสร้างตัวแบบ บันทึกโครงสร้างตัวแบบ และค่าของน้ำหนัก (ต่อ)

```
#นำ MAPE และ MAD ระหว่างตัวแบบโครงข่ายประสาทเทียม และการพยากรณ์สินค้าของแต่ละพื้นที่
การขาย (Sale Forecast) มาเปรียบเทียบกัน ในแต่ละพื้นที่การขาย (Branch Code)
(df_predictTrain.groupby(['BranchCode'])[['MAPE','MAD','MAPE_SF','MAD_SF']].mean())
.sort_values('MAPE')
(df_predictTest.groupby(['BranchCode'])[['MAPE','MAD','MAPE_SF','MAD_SF']].mean())
.sort_values('MAPE')
(df_2019.groupby(['BranchCode'])[['MAPE','MAD','MAPE_SF','MAD_SF']].mean())
.sort_values('MAPE')

#บันทึกโครงสร้างตัวแบบ และค่าของน้ำหนัก เพื่อทำการเปรียบเทียบค่าไฮเปอร์พารามิเตอร์
#หากไม่บันทึก การทดสอบแต่ละครั้งจะไม่เหมือนเดิม แม้ว่าค่าไฮเปอร์พารามิเตอร์เหมือนเดิม เนื่องจาก
ค่าของน้ำหนักจะเปลี่ยนแปลงไปในแต่ละการฝึกฝน
model_yaml=model.to_yaml()
with open("MODEL/48_30_1/Learning_0.35/Momentum_0.9/model.yaml", "w") as
yaml_file : yaml_file.write(model_yaml)
model.save_weights("MODEL/48_30_1/Learning_0.35/Momentum_0.9/model.h5")
```

ตารางที่ ข.3 คำสั่งที่ใช้ในการนำตัวแบบที่บันทึก และค่าของน้ำหนักไว้ มาทำการหาค่าไฮเปอร์พารามิเตอร์ที่เหมาะสมที่สุด

```
#บังคับให้ประมวลผลบน CPU เครื่องคอมพิวเตอร์
import os
os.environ["CUDA_DEVICE_ORDER"] = "PCI_BUS_ID"
os.environ["CUDA_VISIBLE_DEVICES"] = ""

#นำเข้าไลบรารีที่จำเป็นในการหาค่าไฮเปอร์พารามิเตอร์ที่เหมาะสมที่สุด
import pandas as pd
import numpy  as np
from keras.models import model_from_yaml
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ข.3 คำสั่งที่ใช้ในการนำตัวแบบที่บันทึก และค่าของน้ำหนักไว้ มาทำการหาค่าไฮเปอร์พารามิเตอร์ที่เหมาะสมที่สุด (ต่อ)

```
#นำเข้าข้อมูลทั้ง 3 ชุดที่ผ่านการเตรียมข้อมูลแล้ว สำหรับฝึกฝนตัวแบบ ทดสอบตัวแบบ และทวนสอบตัวแบบ
X_train=pd.read_csv('File/X_train.csv')
Y_train=pd.read_csv('File/Y_train.csv')
X_train.drop(['Unnamed: 0'],axis=1,inplace=True)
Y_train.drop(['Unnamed: 0'],axis=1,inplace=True)
X_test=pd.read_csv('File/X_test.csv')
Y_test=pd.read_csv('File/Y_test.csv')
X_test.drop(['Unnamed: 0'],axis=1,inplace=True)
Y_test.drop(['Unnamed: 0'],axis=1,inplace=True)
X_2019=pd.read_csv('File/X_2019.csv')
Y_2019=pd.read_csv('File/Y_2019.csv')
X_2019.drop(['Unnamed: 0'],axis=1,inplace=True)
Y_2019.drop(['Unnamed: 0'],axis=1,inplace=True)
df_testCheck=pd.read_csv('File/Y_testCheck.csv')
df_trainCheck=pd.read_csv('File/Y_trainCheck.csv')
df_testCheck.drop(['Unnamed: 0'],axis=1,inplace=True)
df_trainCheck.drop(['Unnamed: 0'],axis=1,inplace=True)

#นำเข้ายอดขายสูงสุด และยอดขายต่ำสุด ของชุดข้อมูลฝึกฝน
maxAC=3204910.2
minAC=606.98

#เป็นคำสั่งที่ใช้ดึงโครงสร้างตัวแบบที่ และค่าของน้ำหนักของตัวแบบที่ถูกบันทึกไว้
yaml_file=open("Model/model.yaml", 'r')
loaded_model_yaml=yaml_file.read()
yaml_file.close()
model=model_from_yaml(loaded_model_yaml)
model.load_weights("Model/model.h5")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ข.3 คำสั่งที่ใช้ในการนำตัวแบบที่บันทึก และค่าของน้ำหนักไว้ มาทำการหาค่าไฮเปอร์พารามิเตอร์ที่เหมาะสมที่สุด (ต่อ)

```
#นำผลลัพธ์ของข้อมูลทั้ง 3 ชุด มาสร้างตาราง เพื่อนำมาวิเคราะห์ในภายหลัง
yhatTest=model.predict(X_test)
df_predictTest=pd.DataFrame(yhatTest,columns=['yhatTest'])
df_predictTest=pd.concat([df_predictTest,df_testCheck],axis='columns')
df_predictTest['yhatTestTR']=df_predictTest['yhatTest']*(maxAC-minAC)+minAC
df_predictTest['MAPE']=abs((df_predictTest['ACTUAL']-
df_predictTest['yhatTestTR'])/df_predictTest['ACTUAL'])*100
df_predictTest['MAD']=abs((df_predictTest['ACTUAL']-df_predictTest['yhatTestTR']))
yhatTrain=model.predict(X_train)
df_predictTrain=pd.DataFrame(yhatTrain,columns=['yhatTrain'])
df_predictTrain=pd.concat([df_predictTrain,df_trainCheck],axis='columns')
df_predictTrain['yhatTrainTR']=df_predictTrain['yhatTrain']*(maxAC-minAC)+minAC
df_predictTrain['MAPE']=abs((df_predictTrain['ACTUAL']-
df_predictTrain['yhatTrainTR'])/df_predictTrain['ACTUAL'])*100
df_predictTrain['MAD']=abs((df_predictTrain['ACTUAL']-df_predictTrain['yhatTrainTR']))
yhat2019=model.predict(X_2019)
df_2019=pd.DataFrame(yhat2019,columns=['yhat2019'])
df_2019=pd.concat([df_2019,Y_2019],axis='columns')
df_2019['yhat2019TR']=df_2019['yhat2019']*(maxAC-minAC)+minAC
df_2019['MAPE']=abs((df_2019['ACTUAL']-
df_2019['yhat2019TR'])/df_2019['ACTUAL'])*100
df_2019['MAD']=abs((df_2019['ACTUAL']-df_2019['yhat2019TR']))

#นำ MAPE และ MAD ระหว่างตัวแบบโครงข่ายประสาทเทียม และการพยากรณ์สินค้าของแต่ละพื้นที่การขาย (Sale Forecast) มาเปรียบเทียบกัน
df_predictTrain[['MAPE','MAD','MAPE_SF','MAD_SF']].mean()
df_predictTest[['MAPE','MAD','MAPE_SF','MAD_SF']].mean()
df_2019[['MAPE','MAD','MAPE_SF','MAD_SF']].mean()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ข.3 คำสั่งที่ใช้ในการนำตัวแบบที่บันทึก และค่าของน้ำหนักไว้ มาทำการหาค่าไฮเปอร์พารามิเตอร์ที่เหมาะสมที่สุด (ต่อ)

```
#นำ MAPE และ MAD ระหว่างตัวแบบโครงข่ายประสาทเทียม และการพยากรณ์สินค้าของแต่ละพื้นที่การขาย (Sale Forecast) มาเปรียบเทียบกัน ในแต่ละพื้นที่การขาย (Branch Code)
(df_predictTrain.groupby(['BranchCode'])[['MAPE','MAD','MAPE_SF','MAD_SF']].mean())
.sort_values('MAPE')
(df_predictTest.groupby(['BranchCode'])[['MAPE','MAD','MAPE_SF','MAD_SF']].mean())
.sort_values('MAPE')
(df_2019.groupby(['BranchCode'])[['MAPE','MAD','MAPE_SF','MAD_SF']].mean())
.sort_ values('MAPE')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้