วิธีการเรียนรู้เชิงลึกสำหรับระบบแนะนำ

Deep Learning Method for Recommender System

ศุภวิชญ์ วงศ์อดุลวิทย์
สหรัฐ หริรักษ์
อัษฎาวุธ ทองธรรมชาติ

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

Deep Learning Method for Recommender System

Suphawit Wongadoonwit
Saharat Harirak
Assadavud Thongthammachad

A SPECIAL PROBLEM SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT FOR
THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)
DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2017

หัวข้อปัญหาพิเศษ	วิธีการเรียนรู้เชิงลึกสำหรับระบบแนะนำ Deep Learning Method for Recommender System
ชื่อนักศึกษา	นายศุภวิชญ์ วงศ์อดุลวิทย์ รหัสนักศึกษา 57050335 นายสหรัฐ หริรักษ์ รหัสนักศึกษา 57050337 นายอัษฎาวุธ ทองธรรมชาติ รหัสนักศึกษา 57050364
ปริญญา	วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2560
อาจารย์ที่ปรึกษา	ดร.อัคเดช อุดมชัยพร

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) อนุมัติให้ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์) ประจำปีการศึกษา 2560

คณะกรรมการสอบ	**ลายมือชื่อ**
ผศ.ดร.ศรัณย์ อินทโกสุม ประธานกรรมการ	
ผศ.ดร.นวลสวาท หิรัญสกลวงศ์ กรรมการ	
ดร.อัคเดช อุดมชัยพร กรรมการและอาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อปัญหาพิเศษ วิธีการเรียนรู้เชิงลึกสำหรับระบบแนะนำ
ชื่อนักศึกษา นายศุภวิชญ์ วงศ์อดุลวิทย์ รหัสนักศึกษา 57050335
นายสหรัฐ หิริรักษ์ รหัสนักศึกษา 57050337
นายอัษฎาวุธ ทองธรรมชาติ รหัสนักศึกษา 57050364
ปริญญา วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา วิทยาการคอมพิวเตอร์
คณะ วิทยาศาสตร์
มหาวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา 2560
อาจารย์ที่ปรึกษา ดร.อัคเดช อุดมชัยพร

บทคัดย่อ

ปัญหาพิเศษนี้นำเสนอวิธีการเรียนรู้เชิงลึกสำหรับระบบแนะนำโดยใช้เทคนิคโครงข่ายประสาทเทียมแบบวนซ้ำ เพื่อแนะนำข้อมูลต่าง ๆ ให้กับผู้ใช้งาน การทดลองในงานวิจัยนี้ใช้ชุดข้อมูลทดสอบจากมูฟวีเลนส์ (MovieLens) และชุดข้อมูลทดสอบจากอเมซอน (Amazon) เพื่อทำการทดสอบประสิทธิภาพและประสิทธิผลของวิธีการที่ได้นำเสนอ และเปรียบเทียบกับวิธีการแบบดั้งเดิม ผลการทดลองบ่งชี้ว่าวิธีการเรียนรู้เชิงลึกให้ผลลัพธ์ที่ดีกว่าวิธีการแบบดั้งเดิมในแง่ของค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย สำหรับการทดลองเปรียบเทียบเทคนิควิธีการเรียนรู้เชิงลึกโดยใช้เทคนิคโครงข่ายประสาทเทียมแบบวนซ้ำแบบ LSTM และ GRU ผลปรากฏว่าวิธีการแบบ GRU และ LSTM ให้ผลลัพธ์ไม่แตกต่างกันอย่างมีนัยสำคัญในแง่ของความแม่นยำและค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย แต่วิธีการแบบ GRU ให้ผลลัพธ์ที่ดีกว่าในแง่ของเวลาที่ใช้ในการประมวลผล

คำสำคัญ : ระบบแนะนำ การเรียนรู้เชิงลึก โครงข่ายประสาทเทียมแบบวนซ้ำ แอลเอสทีเอ็ม จีอาร์ยู

Title	Deep Learning Method for Recommender System			
Students	Mr.Suphawit	Wongadoonwit	Student ID	57050335
	Mr.Saharat	Harirak	Student ID	57050337
	Mr.Assadavud	Thongthammachad	Student ID	57050364
Degree	Bachelor of Science (Computer Science)			
Department	Computer Science			
Faculty	Science			
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)			
Academic Year	2017			
Advisor	Dr.Akadej Udomchaiporn			

Abstract

This special problem presents deep learning method for recommender system. The proposed method applied Recurrent Neural Network (RNN) to recommender system using MovieLens dataset and Amazon dataset. The experimental results indicated that the proposed approach produced the better results, comparing to traditional approaches such as Collaborative Filtering and Latent Semantic Indexing, in terms of Mean Absolute Error (MAE). The experimental results also indicated that Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) produced the similar results in terms of effectiveness but GRU produced the better result in terms of efficiency.

Keywords : Recommender System, Deep Learning, Recurrent Neural Network, Long-Short Term Memory, Gated Recurrent Unit.

กิตติกรรมประกาศ

การจัดทำปัญหาพิเศษเรื่องวิธีการเรียนรู้เชิงลึกสำหรับระบบแนะนำนี้สำเร็จลุล่วงไปได้ด้วยดี คณะผู้จัดทำต้องขอขอบพระคุณอาจารย์ที่ปรึกษา ดร. อัคเดช อุดมชัยพร ที่ช่วยให้คำปรึกษาและคำแนะนำที่ดีแก่คณะผู้จัดทำในการปรับปรุงปัญหาพิเศษนี้

ขอขอบพระคุณอาจารย์ผู้ควบคุมการสอบปัญหาพิเศษ ผศ.ดร.ศรัณย์ อินทโกสุม และ ผศ.ดร.นวลสวาท หิรัญสกลวงศ์ ที่มีส่วนช่วยในการตรวจสอบ และให้คำแนะนำ ทำให้ปัญหาพิเศษนี้มีความสมบูรณ์มากยิ่งขึ้น

ขอขอบพระคุณบุคคลต่าง ๆ ที่เกี่ยวข้องที่ได้ให้การช่วยเหลือในการทำปัญหาพิเศษนี้ และสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่มอบโอกาสให้ได้เข้าศึกษาในสถาบันแห่งนี้

ศุภวิชญ์ วงศ์อดุลวิทย์
สหรัฐ หริรักษ์
อัษฎาวุธ ทองธรรมชาติ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ช
สารบัญภาพ	ฌ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตการศึกษา	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	3
1.5 ขั้นตอนการดำเนินงาน	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
2.1 ระบบแนะนำ	4
2.1.1 วิธีการแนะนำแบบขึ้นอยู่กับเนื้อหา	5
2.1.2 วิธีการแนะนำโดยใช้การกรองด้วยความร่วมมือ	5
2.2 โครงข่ายประสาทเทียมแบบวนซ้ำ	7
2.2.1 Long Short Term Memory	8
2.2.2 Gated Recurrent Units	12
2.3 Keras Framework	15
2.3.1 การติดตั้ง Keras Framework	16
2.4 Surprise Library	16
2.5 Confusion Matrix	17
2.6 Mean Absolute Error : MAE	19
2.7 งานวิจัยที่เกี่ยวข้อง	20
2.7.1 งานวิจัยเรื่องวิธีการกรองด้วยความร่วมมือแบบแอลเอสไอ สำหรับระบบแนะนำ	20
2.7.2 งานวิจัยเรื่องการศึกษาทดลองตัววัดความเหมือนสำหรับการกรองด้วยความร่วมมือ	21

สารบัญ (ต่อ)

หน้า

2.7.3 งานวิจัยเรื่อง A Recurrent Neural Network Based Recommendation System 24

บทที่ 3 วิธีดำเนินงานวิจัย 26

3.1 ระเบียบวิธีวิจัย 26

3.1.1 ศึกษาวิธีการต่าง ๆ ของวิธีการเรียนรู้เชิงลึกสำหรับระบบแนะนำ 27

3.1.2 ทดลองการเรียนรู้เชิงลึกสำหรับระบบแนะนำโดยใช้วิธีการต่าง ๆ 27

3.1.3 ทดลองปรับค่าพารามิเตอร์ที่ใช้ในวิธีการต่าง ๆ 27

3.1.4 ประเมินผลการทดลอง 27

3.1.5 ทดสอบทางสถิติ 28

3.1.6 วิเคราะห์และสรุปผลการวิจัย 28

3.2 ขั้นตอนวิธีการทดลองระบบแนะนำโดยใช้วิธีการเรียนรู้เชิงลึก 28

3.2.1 ขั้นตอนการใช้โครงข่ายประสาทเทียมแบบวนซ้ำกับระบบแนะนำ 28

3.2.1.1 การทดสอบชุดข้อมูล MovieLens โดยใช้โครงข่ายประสาทเทียมแบบวนซ้ำด้วย Keras Framework 33

3.2.1.2 การทดสอบชุดข้อมูล Amazon โดยใช้โครงข่ายประสาทเทียมแบบวนซ้ำด้วย Keras Framework 35

3.2.1.3 ทดสอบชุดข้อมูล MovieLens และ Amazon โดยใช้ Surprise Library 37

บทที่ 4 ผลการดำเนินงานและการอภิปรายผล 39

4.1 ผลการดำเนินงาน 39

4.1.1 ใช้โครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูลจาก MovieLens 39

4.1.2 ใช้โครงข่ายประสาทเทียมแบบวนซ้ำ ทดสอบกับชุดข้อมูลจาก Amazon 42

4.1.3 ใช้วิธีการกรองด้วยความร่วมมือทดสอบกับชุดข้อมูลจาก MovieLens 48

4.1.4 ใช้วิธีการกรองด้วยความร่วมมือทดสอบกับชุดข้อมูลจาก Amazon 49

สารบัญ (ต่อ)

	หน้า
4.2 การอภิปรายผลการดำเนินงาน	51
4.2.1 การเปรียบเทียบค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย	52
4.2.2 การเปรียบเทียบค่าความแม่นยำ	54
4.2.3 การเปรียบเทียบเวลาที่ใช้ในการสร้าง และทดสอบโมเดล	58
4.3 การทดสอบสถิติ	63
4.3.1 การวิเคราะความแปรปรวน	63
4.3.1.1 การวิเคราะห์ความแปรปรวนของค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย	66
4.3.1.2 การวิเคราะห์ความแปรปรวนของค่าความแม่นยำ	68
4.3.1.3 การวิเคราะห์ความแปรปรวนของเวลาที่ใช้ในการสร้างและทดสอบโมเดล	69
4.4 ปัญหาที่พบในการดำเนินงาน	71
4.4.1 ชุดข้อมูลที่นำมาทดลองคนละประเภท	71
4.4.2 ข้อจำทางทรัพยากร	71
4.4.3 การประมวลผลข้อมูลประเภทข้อความ	71
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ	72
5.1 สรุปผลการดำเนินงาน	72
5.2 ข้อเสนอแนะ	73
เอกสารอ้างอิง	74

สารบัญตาราง

ตารางที่		หน้า
2.1	เมตริกซ์ที่ผู้ใช้งานให้คะแนนกับข้อมูลแต่ละชิ้น	6
2.2	Confusion Matrix	17
2.3	ตัวอย่าง Confusion Matrix ที่มีค่าเปรียบเทียบระหว่างคลาสคำตอบและการพยากรณ์ของโมเดล	18
2.4	ผลลัพธ์จากการทดลองเปรียบเทียบประสิทธิภาพของ SVM, GRU และ LSTM	25
3.1	ข้อมูลของผู้ใช้งานที่ให้คะแนนไว้กับภาพยนตร์ในชุดข้อมูล MovieLens	29
3.2	ข้อมูลของผู้ใช้งานในชุดข้อมูล MovieLens	31
4.1	ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของวิธีโครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูล MovieLens	41
4.2	ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของวิธีโครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูล Amazon	43
4.3	ค่าความแม่นยำของของวิธีโครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูล Amazon	44
4.4	เวลาที่ใช้ของวิธีโครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูล Amazon	45
4.5	ผลการทดลองโดยใช้วิธีการกรองด้วยความร่วมมือทดสอบกับชุดข้อมูลจาก MovieLens	48
4.6	ผลการทดลองโดยใช้วิธีการกรองด้วยความร่วมมือทดสอบกับชุดข้อมูลจาก Amazon	50
4.7	การเปรียบเทียบค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของวิธีการต่าง ๆ โดยทดสอบกับชุดข้อมูล MovieLens	52
4.8	การเปรียบเทียบค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของวิธีการต่าง ๆ โดยใช้ชุดข้อมูล Amazon	53
4.9	การเปรียบเทียบค่าความแม่นยำของวิธีการต่าง ๆ ทดสอบกับชุดข้อมูล MovieLens	55
4.10	การเปรียบเทียบค่าความแม่นยำของวิธีการต่าง ๆ ทดสอบกับชุดข้อมูล Amazon	57
4.11	การเปรียบเทียบเวลาที่ใช้ของวิธีการต่าง ๆ ทดสอบกับชุดข้อมูล MovieLens	59
4.12	การเปรียบเทียบเวลาที่ใช้ของวิธีการต่าง ๆ ทดสอบกับชุดข้อมูล Amazon	61

สารบัญตาราง (ต่อ)

ตารางที่		หน้า
4.13	การวิเคราะห์ความแปรปรวน	65
4.14	ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยที่ดีที่สุดของแต่ละวิธีเมื่อทดสอบกับชุดข้อมูล MovieLensและ Amazon	66
4.15	ผลลัพธ์จากการวิเคราะห์ความแปรปรวนของค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยด้วยโปรแกรม Microsoft Excel	67
4.16	ค่าความแม่นยำที่ดีที่สุดของแต่ละวิธีเมื่อทดสอบกับชุดข้อมูล MovieLens และ Amazon	68
4.17	ผลลัพธ์จากการวิเคราะห์ความแปรปรวนของค่าความแม่นยำด้วยโปรแกรม Microsoft Excel	69
4.18	เวลาที่ใช้ที่ดีที่สุดของแต่ละวิธีเมื่อทดสอบกับชุดข้อมูล MovieLens และ Amazon	70
4.19	ผลลัพธ์จากการวิเคราะห์ความแปรปรวนของเวลาที่ใช้ด้วยโปรแกรม Microsoft Excel	70

สารบัญภาพ

ภาพที่		หน้า
2.1	ขั้นตอนวิธีของการแนะนำโดยใช้การกรองด้วยความร่วมมือ	6
2.2	แผนภาพกระบวนการทำงานของโครงข่ายประสาทเทียมแบบวนซ้ำ	7
2.3	กระบวนการทำงานของ LSTM	8
2.4	การคำนวณของ Forget Gate	9
2.5	การคำนวณของ Input Gat และ Input Modulation Gate	10
2.6	การปรับปรุงค่าใน Cell State	11
2.7	การคำนวณผลลัพธ์ของ Output Vector	12
2.8	การทำงานภายใน GRU	13
2.9	การคำนวณ Read Gate และ Forget Gate	14
2.10	การคำนวณ State Candidate	14
2.11	การคำนวณผลลัพธ์	15
2.12	คำสั่งติดตั้ง Keras Framework	16
2.13	คำสั่งติดตั้ง Surprise library ด้วย Command Prompt	17
2.14	กราฟเปรียบเทียบค่า MAE ระหว่างวิธีการกรองด้วยความร่วมมือแบบแอลเอสไอกับแบบดั้งเดิม	21
2.15	กราฟแสดงผลลัพธ์ของวิธีเพียร์สันด้วย MAE	22
2.16	กราฟแสดงผลลัพธ์ของวิธีโคไซน์ด้วย MAE	22
2.17	กราฟแสดงผลลัพธ์ของวิธีไดซ์ด้วย MAE	23
2.18	กราฟแสดงผลลัพธ์ของวิธีสเปียร์แมนด้วย MAE	23
2.19	กราฟแสดงค่าความผิดพลาดสัมบูรณ์เฉลี่ยทั้ง 4 วิธี	24
3.1	ระเบียบวิธีวิจัยของวิธีการเรียนรู้เชิงลึกสำหรับระบบแนะนำ	26
3.2	การทำงานภายในโครงข่ายประสาทเทียมแบบวนซ้ำ	28
3.3	แสดงการประมวลผลเพื่อปรับค่า Weights และ Biases	32
3.4	คำสั่งเรียกใช้งาน Keras Framework	33
3.5	คำสั่งดึงข้อมูลจากไฟล์และแบ่งข้อมูล	34
3.6	คำสั่งแบ่งข้อมูลออกเป็นชุดข้อมูลฝึกสอน และชุดข้อมูลทดสอบ	34
3.7	คำสั่งสร้างโครงข่ายประสาทเทียมแบบวนซ้ำสำหรับชุดข้อมูล MovieLens	34
3.8	ผลลัพธ์ของโครงข่ายประสาทเทียมแบบวนซ้ำเมื่อทดสอบกับชุดข้อมูล MovieLens	35
3.9	คำสั่งสร้างฟังก์ชันในการทำความสะอาดข้อมูล	36

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
3.10	คำสั่งแปลงข้อความเป็นชุดตัวเลข	36
3.11	คำสั่งสร้างโครงข่ายประสาทเทียมแบบวนซ้ำสำหรับชุดข้อมูล Amazon	36
3.12	ผลลัพธ์ของโครงข่ายประสาทเทียมแบบวนซ้ำเมื่อทดสอบกับชุดข้อมูล Amazon	37
3.13	คำสั่งฝึกสอนและทดสองโมเดลของวิธีการกรองด้วยความร่วมมือ	38
3.14	ผลลัพธ์ของวิธีการกรองด้วยความร่วมมือ	38
4.1	ภาพตัวอย่างของข้อมูลนำเข้า	40
4.2	แผนภาพค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของวิธีโครงข่ายประสาทเทียม แบบวนซ้ำทดสอบกับชุดข้อมูล MovieLens	42
4.3	แผนภาพค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยโดยใช้วิธีโครงข่าย ประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูล Amazon	46
4.4	แผนภาพค่าความแม่นยำที่ได้จากการทดลองของวิธีโครงข่าย ประสาทเทียมทดสอบกับชุดข้อมูล Amazon	46
4.5	แผนภาพเวลาที่ใช้ในการสร้าง และทดสอบแบบจำลองของวิธีโครงข่าย ประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูล Amazon	47
4.6	แผนภาพค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของวิธีโครงข่ายประสาทเทียม แบบวนซ้ำทดสอบกับชุดข้อมูล Amazon	54
4.7	แผนภาพการเปรียบเทียบค่าความแม่นยำของวิธีโครงข่ายประสาทเทียม แบบวนซ้ำทดสอบกับชุดข้อมูล MovieLens	56
4.8	แผนภาพค่าความแม่นยำของวิธีโครงข่ายประสาทเทียม แบบวนซ้ำทดสอบกับชุดข้อมูล Amazon	58
4.9	แผนภาพเวลาที่ใช้ของวิธีโครงข่ายประสาทเทียมแบบวนซ้ำทดสอบ กับชุดข้อมูล MovieLens	60
4.10	แผนภาพเวลาที่ใช้ของวิธีการต่าง ๆ ทดสอบกับชุดข้อมูล Amazon	62

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ความต้องการของธุรกิจส่วนใหญ่มุ่งเน้นไปที่การเพิ่มผลกำไรให้กับองค์กร เพื่อที่จะตอบโจทย์ดังกล่าว หลายองค์กรจึงค้นหาวิธีการต่าง ๆ เพื่อตอบสนองความต้องการของธุรกิจ อาทิ การวางแผนการใช้ทรัพยากรภายในองค์กร การจัดรูปแบบการส่งเสริมการขาย การสร้างความประทับใจ ในการบริการให้กับลูกค้า หรือการใช้ระบบแนะนำเพื่อชักจูงให้ลูกค้าซื้อสินค้าเพิ่มขึ้น ในปัจจุบันองค์กรต่าง ๆ ได้มีการเก็บรวบรวมข้อมูลพฤติกรรมของลูกค้าเพื่อนำมาศึกษา วิเคราะห์หาความน่าจะเป็นที่จะเกิดขึ้น เพื่อวางแผนในการดึงดูดลูกค้าให้สนใจกับผลิตภัณฑ์หรือบริการของธุรกิจ วิธีการดังกล่าวเรียกว่าระบบแนะนำ (Recommender System: RS)

ระบบแนะนำเป็นเทคนิคหรือเครื่องมือในการให้คำแนะนำแก่ลูกค้า อาทิ จะซื้อสินค้าอะไร จะพักโรงแรมไหน เป็นต้น ซึ่งระบบแนะนำมีอยู่หลายเทคนิควิธี เช่น Content-Based, Collaborative Filtering, Demographic, Knowledge-Based, Community-Based และ Hybrid Recommender System หรือวิธีการแนะนำโดยการวิเคราะห์ และทำนายข้อมูลโดยใช้ความ น่าจะเป็น เช่น Decision Tree, Bayesian Network, K-means Clustering และ Hierarchical ซึ่งวิธีการต่าง ๆ มีขั้นตอนการทำงานที่แตกต่างกัน ความแม่นยำที่ได้จากวิธีการต่าง ๆ ก็แตกต่างกัน ออกไป โดยในปัญหาพิเศษนี้จะใช้วิธีการเรียนรู้เชิงลึก (Deep Learning) โดยมุ่งเน้นไปที่ขั้นตอนวิธีแบบโครงข่ายประสาทเทียมแบบวนซ้ำ (Recurrent Neural Network : RNN) สำหรับระบบแนะนำ โดยมุ่งหวังว่าระบบแนะนำจะมีความแม่นยำตรงตามความต้องการของผู้ใช้งานมากขึ้น เมื่อเปรียบเทียบกับวิธีการพัฒนาระบบแนะนำแบบดั้งเดิม

1.2 วัตถุประสงค์ของการวิจัย

1) เพื่อศึกษาและทดลองวิธีการต่าง ๆ ของระบบแนะนำรวมถึงวิธีการเรียนรู้เชิงลึกแบบวนซ้ำ

2) เพื่อเปรียบเทียบผลลัพธ์ที่ได้จากการพัฒนาระบบแนะนำ โดยใช้วิธีการเรียนรู้เชิงลึกแบบวนซ้ำและวิธีการแบบดั้งเดิมอื่น ๆ

3) เพื่อนำผลลัพธ์ที่ได้มาศึกษาและวิเคราะห์หาองค์ความรู้ต่อไป

1.3 ขอบเขตการศึกษา

1) ปัญหาพิเศษนี้จะมุ่งเน้นไปที่การทดลองระบบแนะนำ โดยใช้วิธีโครงข่ายประสาทเทียมแบบวนซ้ำสองแบบ คือ Long Short-Term Memory (LSTM) และ Gated Recurrent Units (GRU) รวมทั้งวิธีการแบบดั่งเดิม คือ Traditional Collaborative Filtering (TCF) โดยจะนำผลลัพธ์ที่ได้ไปเปรียบเทียบกับผลลัพธ์ของงานวิจัยที่เกี่ยวข้องที่ใช้วิธีการ TCF (แต่ใช้ชุดพารามิเตอร์ที่ต่างจากในปัญหาพิเศษนี้) และ Latent Semantic Indexing (LSI)

2) ในการทดลอง และเปรียบเทียบประสิทธิภาพ และประสิทธิผลของระบบแนะนำจะทดสอบกับ

   ก) ชุดข้อมูลมาตรฐานจาก MovieLens โดยที่มีการให้คะแนนความพึงพอใจจำนวน 100,004 ครั้ง ผู้ใช้งานจำนวน 671 คน และภาพยนตร์จำนวน 9,125 เรื่อง โดยมีช่วงคะแนนความพึงพอใจระหว่างหนึ่งถึงห้า และ

   ข) ชุดข้อมูลการขายหนังสือจาก Amazon ที่มียอดรีวิวจำนวน 8,898,041 รีวิว แต่เนื่องจากมีข้อจำกัดด้านเวลาและทรัพยากรจึงสุ่มใช้เพียง 500,000 รีวิวจากข้อมูลทั้งหมด

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้ทราบถึงประสิทธิภาพและประสิทธิผลของระบบแนะนำที่ถูกพัฒนาจากวิธีการต่าง ๆ

2) ได้ทราบวิธีที่เหมาะสมสำหรับการแนะนำสินค้าให้กับผู้ใช้งาน

3) ได้ทราบองค์ความรู้ที่ได้มาจากการศึกษาและวิเคราะห์ผลลัพธ์ที่ได้

1.5 ขั้นตอนการดำเนินงาน

1) ศึกษาวิธีการทำงานของระบบแนะนำที่ถูกพัฒนาโดยวิธีการต่าง ๆ และศึกษาการทำงานของวิธีการเรียนรู้เชิงลึก

2) กำหนดและรวบรวมชุดข้อมูลทดลองมาตรฐานเพื่อใช้ทดลองระบบแนะนำ

3) พัฒนาระบบแนะนำ โดยใช้วิธีการเรียนรู้เชิงลึกและทดลองกับชุดข้อมูลมาตรฐาน

4) ประเมินผลและปรับปรุงผลการทดลอง

5) วิเคราะห์และสรุปผลการทดลอง

6) จัดทำเอกสารประกอบปัญหาพิเศษ

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้อง ซึ่งประกอบไปด้วย ระบบแนะนำ วิธีการแนะนำแบบขึ้นอยู่กับเนื้อหา วิธีการแนะนำโดยใช้การกรองด้วยความร่วมมือ วิธีการแบบโครงข่ายประสาทเทียมแบบวนซ้ำ และงานวิจัยที่เกี่ยวข้องกับระบบแนะนำ โดยจะแสดงรายละเอียดในหัวข้อ 2.1 ถึง 2.7

2.1 ระบบแนะนำ (Recommender System)

ระบบแนะนำ เป็นระบบที่เข้ามาช่วย E-Business ในการแนะนำข้อมูลหรือผลิตภัณฑ์ให้กับลูกค้า โดยจะใช้ข้อมูลที่รวบรวมจากการกระทำของลูกค้าเช่น ซื้อ เรียกดู หรือให้คะแนนกับสิ่งของมาทำการทำนายความสนใจของลูกค้า ตัวอย่างเช่น Amazon.com ได้เก็บรวบรวมข้อมูลการซื้อของผู้ใช้แต่ละคนเพื่อนำไปทำนายว่าผู้ใช้จะมีแนวโน้มสนใจสินค้าประเภทใด หลังจากนั้น จะนำผลลัพธ์ที่ทำนายได้มาแนะนำสินค้าให้กับผู้ใช้ โดยผู้ที่ให้คำแนะนำต่อระบบจะถูกเรียกว่า User และสิ่งของที่ถูกให้คำแนะนำจะเรียกว่า Item

สำหรับวิธีที่เป็นที่นิยมในการใช้ในระบบแนะนำคือ Collaborative Filtering และ Content-Based โดยวิธีแนะนำแบบ Collaborative Filtering จะพิจารณาจากการให้คะแนน ความพึงพอใจของผู้ใช้และผู้ใช้คนอื่น ๆ ในระบบ ซึ่งวิธีนี้จะแนะนำ Item ที่มีค่าความพึงพอใจคล้ายกันมากที่สุด ส่วนวิธี Content-Based จะพิจารณาจากความชอบหรือความคุ้นเคยในอดีต ของผู้ใช้ และจะแนะนำ Item ที่มีความคล้ายกันกับความชอบในอดีตของผู้ใช้มากที่สุด

2.1.1 วิธีการแนะนำแบบขึ้นอยู่กับเนื้อหา (Content-Based Recommender System)

วิธีการแนะนำแบบขึ้นอยู่กับเนื้อหา เป็นวิธีการที่จะแนะนำสิ่งของที่ต้องการ ให้กับผู้ใช้ โดยพิจารณาจากคุณลักษณะของสิ่งของ เมื่อผู้ใช้ให้คะแนนความพอใจกับสิ่งของหรือสิ่งที่ผู้ใช้เคยชอบ ระบบก็จะแนะนำสิ่งของที่มีคุณลักษณะใกล้เคียงกันกับที่ผู้ใช้ ให้คะแนน โดยสิ่งที่พิจารณาจะเป็นชื่อ หรือ คุณลักษณะที่โดดเด่นของสิ่งของ เช่น การซื้อหนังสือ เมื่อผู้ใช้สนใจซื้อหนังสือ ระบบก็อาจจะแนะนำหนังสือที่มีผู้แต่งเดียวกันให้กับผู้ใช้ หรือหนังสือที่มีคุณลักษณะอื่น ๆ ที่ใกล้เคียงกัน

วิธีการแนะนำแบบขึ้นอยู่กับเนื้อหานี้ ไม่เหมาะสมกับข้อมูลที่ไม่ได้อยู่ในรูปแบบข้อความ เช่นเสียง หรือรูปภาพ และสำหรับการให้คะแนนความพอใจ หากมีคะแนนที่ขาดหายไปจะทำให้การทำงานไม่สามารถทำได้อย่างมีประสิทธิภาพ

2.1.2 วิธีการแนะนำโดยใช้การกรองด้วยความร่วมมือ (Collaborative Filtering Recommender System)

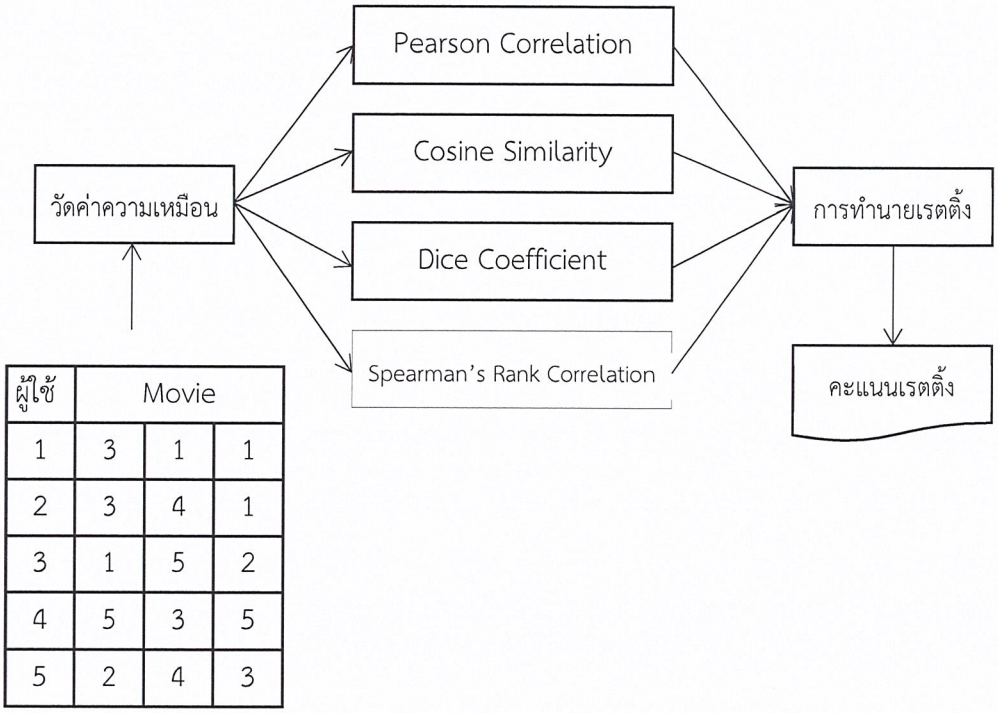
เป็นวิธีการแนะนำที่มักจะใช้ในฐานข้อมูลที่มีขนาดใหญ่ โดยจะต้องทราบถึงเป้าหมายของผู้ใช้ก่อนว่าชอบหรือต้องการข้อมูลประเภทใด จึงจะสามารถค้นหาข้อมูลให้ตรงตามความต้องการของผู้ใช้ได้ กระบวนการทำงานของวิธีการแนะนำโดยใช้การกรองด้วยความร่วมมือมีสองขั้นตอน คือ 1) การจัดเก็บข้อมูล 2) ขั้นตอนวิธีการแนะนำ

ขั้นตอนที่หนึ่ง การจัดเก็บข้อมูลเป็นการนำข้อมูลมาเก็บในรูปของเมตริกซ์ของผู้ใช้กับข้อมูลเรตติ้ง (User-Item Matrix) แสดงดังตารางที่ 2.1 โดยที่ User# คือ ลำดับข้อมูลของผู้ใช้ Item# คือ ลำดับข้อมูลของสิ่งของ

ตารางที่ 2.1 เมตริกซ์ที่ผู้ใช้งานให้คะแนนกับข้อมูลแต่ละชิ้น

	Item #1	Item #2	Item #3	Item #4	Item #5	Item #6
User #1	1	4	?	3	3	2
User #2	3	?	2	5	?	4
User #3	?	5	3	4	3	5
User #4	2	3	4	?	3	4
User #5	3	2	4	3	3	4

ขั้นตอนที่สอง ขั้นตอนวิธีการแนะนำประกอบด้วยสองขั้นตอนย่อย คือ 1) การคำนวณด้วยมาตรวัดความเหมือน 2) การทำนายข้อมูลเรตติ้งออกมาเป็นตัวเลข ที่คาดว่าผู้ใช้เป้าหมายจะให้ไว้กับข้อมูลชิ้นนั้น ๆ ขั้นตอนวิธีของวิธีการแนะนำโดยใช้การกรองด้วยความร่วมมือแสดงดังภาพที่ 2.1

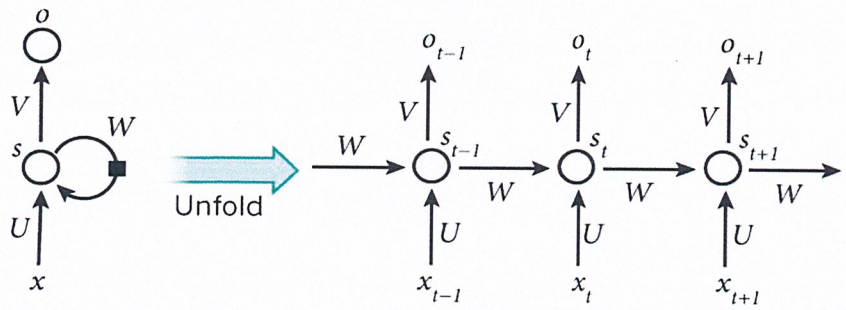


ภาพที่ 2.1 ขั้นตอนวิธีของการแนะนำโดยใช้การกรองด้วยความร่วมมือ

2.2 โครงข่ายประสาทเทียมแบบวนซ้ำ

ปกติแล้ว Input หรือ Output ที่ได้จาก Traditional Neural Network นั้น สันนิษฐานว่าไม่มีความเกี่ยวข้องซึ่งกันและกัน แต่ในหลาย ๆ กรณี การที่จะทำนายให้เกิดประสิทธิภาพมากขึ้นนั้น บางครั้งจำเป็นต้องทราบถึง Output ล่าสุดเพื่อนำไปเป็นตัวแปรนำเข้าในการทำนายผลครั้งถัดไป

โครงข่ายประสาทเทียมแบบวนซ้ำเป็น Neural Network ที่ใช้กับข้อมูลที่เป็นลำดับ (Sequence) ในการทำนายข้อมูลต่าง ๆ เช่น ทำการคำนวนโดยใช้ Output ที่ได้จากการคำนวณครั้งก่อนมาพิจารณานาเพื่อทำนายผล โดยภายในโครงข่ายประสาทเทียมแบบวนซ้ำนั้นมีหน่วยความจำที่รวบรวมผลที่ทำการคำนวณไปแล้ว แสดงดังภาพที่ 2.2



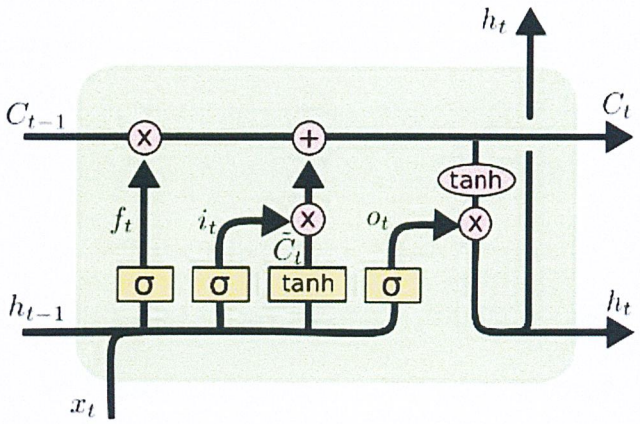
ภาพที่ 2.2 กระบวนการทำงานของโครงข่ายประสาทเทียมแบบวนซ้ำ

จากภาพที่ 2.2 แสดงถึงแนวคิดของโครงข่ายประสาทเทียมแบบวนซ้ำเห็นได้ว่าภายในขั้นตอนวิธีมีการคำนวณโดยใช้ Output ครั้งก่อนมาคำนวณในครั้งถัดไปด้วย

โดยปกติ RNN จะนำข้อมูล Output ก่อนหน้ามาคำนวณเพื่อทำนาย Input ถัดไป ปัญหาอย่างหนึ่งของ RNN คือการไม่สามารถเรียนรู้ข้อมูลที่เป็นชุดใหญ่ ๆ ในการทำนายข้อมูลที่อยู่ท้าย ๆ ซึ่งมีความเกี่ยวข้องกันกับข้อมูลที่อยู่แรก ๆ ตัวอย่างเช่น เมื่ออ่านข้อความ “ฉันเกิดที่ประเทศไทย ... ฉันพูดภาษา ไทย” โดยช่องว่างคือข้อมูลจำนวนมาก เมื่ออ่านจะสามารถคาดเดาได้ว่าคำตอบคือภาษาไทยซึ่งเป็นประเทศที่เกิดโดยพิจารณากับความสัมพันธ์ที่อยู่ช่วงแรก แต่โครงข่ายประสาทเทียมแบบวนซ้ำไม่สามารถที่จะจับใจความและทำนายได้ จึงเรียกปัญหานี้ว่า Long-Term Dependencies

2.2.1 Long Short-Term Memory

Long Short-Term Memory หรือ LSTM เป็นโครงข่ายประสาทเทียมแบบหนึ่งที่ถูกออกแบบมาสำหรับการประมวลผลลำดับ ซึ่งถูกใช้ในการแก้ปัญหาของ โครงข่ายประสาทเทียมแบบวนซ้ำที่มีต่อการประมวลผลลำดับของข้อมูลที่มีความยาว กระบวนการทำงานของ LSTM แสดงดังภาพที่ 2.3



ภาพที่ 2.3 กระบวนการทำงานของ LSTM

เมื่อ x_t คือ Input Vector

C_t คือ Cell State Vector

h_t คือ Output Vector

f_i คือ Forget Gate

i_t คือ Input Gate

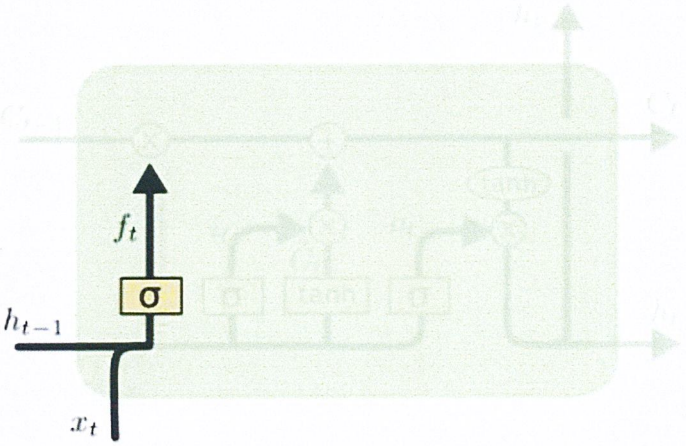
$\tilde{C}_t$ คือ Input Modulation Gate

o_t คือ Output Gate

หลักการทำงานของ LSTM

LSTM จะพิจารณาจากข้อมูลนำเข้าว่า จะเก็บข้อมูลตัวก่อนหน้าไว้หรือลบข้อมูลตัวก่อนหน้า ในการคำนวณจะใช้ Sigmoid Function ซึ่งจะมีค่าระหว่างศูนย์ถึงหนึ่ง ส่วนนี้จะเรียกว่า Forget Gate โดยจะพิจารณาจากข้อมูลนำเข้าคือ h_{t-1} และ x_t หากมีค่าศูนย์ก็จะลบข้อมูล หากมีค่าหนึ่งก็จะเก็บข้อมูลไว้ การคำนวณของ Forget Gate แสดงดังภาพที่ 2.4

$$f_t = \sigma\left(W_f * [h_t - 1, x_t] + b_t\right)$$



ภาพที่ 2.4 การคำนวณของ Forget Gate

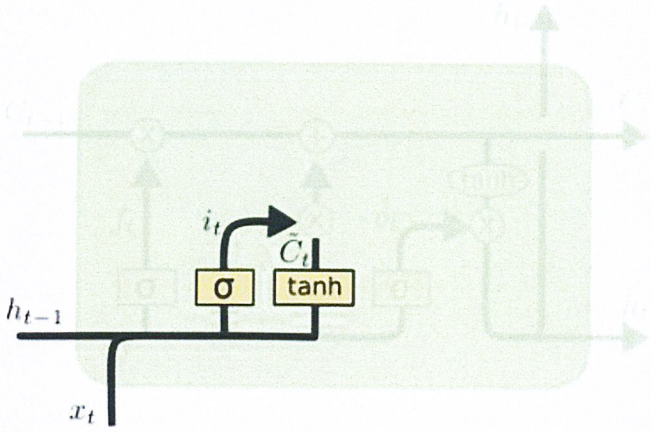
ลำดับต่อไปเป็นการพิจารณาว่าจะนำข้อมูลไปเขียนใน Cell State หรือไม่ โดยภาพที่ 2.5 แสดงถึงการคำนวณ Input Gate และ Input Modulation Gate ส่วนของ Input Gate ใช้ข้อมูล h_{t-1} และ x_t เป็นข้อมูลนำเข้าโดยใช้ Sigmoid Function พิจารณาว่าจะนำข้อมูลนำเข้าใหม่ไปเขียนหรือไม่

$$i_t = \sigma\ (W_i * [h_t - 1, x_t]\ +\ b_i)$$

ส่วนที่สองเรียกว่า Input Modulation Gate หรือเรียกว่า Candidate Values จะเป็นการพิจารณาข้อมูลนำเข้าว่าจะเขียนด้วยค่าอะไร โดยจะใช้ฟังก์ชัน Hyperbolic Tangent

$$\widetilde{C_t} = \tanh\ (W_c * [h_t - 1, x_t]\ +\ b_c)$$

หลังจากได้ค่าทั้งสองแล้ว จะนำไปรวมกันและนำไปปรับปรุงข้อมูลใน Cell State ในขั้นตอนต่อไป

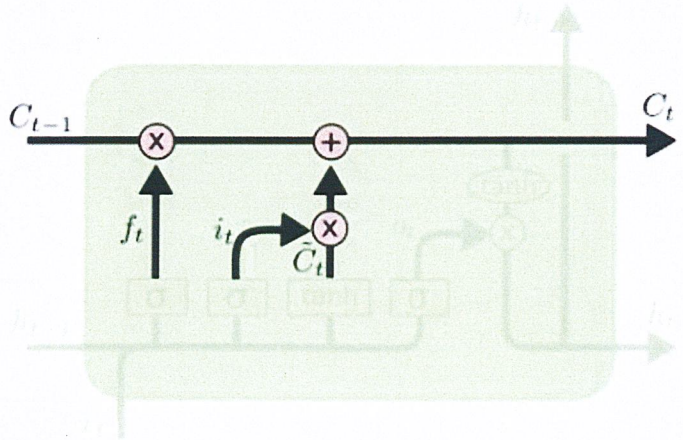


ภาพที่ 2.5 การคำนวณของ Input Gate และ Input Modulation Gate

ในการปรับปรุงค่าใน Cell State จะเริ่มปรับปรุงจาก C_{t-1} ไปยัง C_t โดยในการปรับปรุงจะใช้ข้อมูลที่ได้จากขั้นตอนก่อนหน้า คือข้อมูลจาก Forget Gate, Input Gate และ Input Modulation Gate มาปรับปรุง

โดยเริ่มจาก f_t คูณกับ C_{t-1} ในส่วนนี้จะเป็นการเลือกว่าจะนำข้อมูลจาก Cell State ก่อนหน้ามาใช้หรือลืมข้อมูลนั้นไป จากนั้นนำไปบวกรวมกับส่วนของ Input คือ $i_t * \tilde{C}_t$ โดยค่า Candidate Values ใหม่ที่ได้นั้นจะมากหรือน้อย ก็ขึ้นอยู่กับ Input Gate ด้วย กระบวนการปรับปรุงค่าใน Cell State แสดงดังภาพที่ 2.6

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t$$

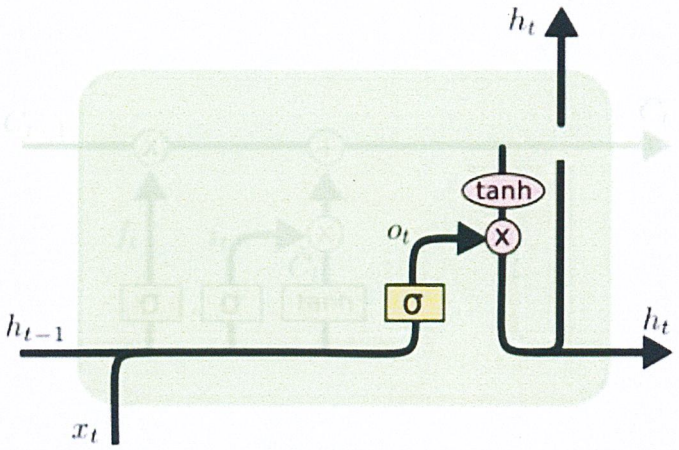


ภาพที่ 2.6 การปรับปรุงค่าใน Cell State

ขั้นตอนการอ่านข้อมูลนั้นจะพิจารณาว่าจะอนุญาตให้อ่านหรือไม่ โดยใช้ Output Gate และข้อมูลนำเข้า h_{t-1} และ x_t จากนั้นนำข้อมูลจาก Cell State มาผ่านฟังก์ชัน Hyperbolic Tangent คูณเข้ากับ Output Gate และส่งผลลัพธ์ออกไปยังการคำนวณรอบต่อไป การคำนวณผลลัพท์แสดงดังภาพที่ 2.7

$$o_t = \sigma\left(W_o * [h_t - 1, x_t] + b_o\right)$$

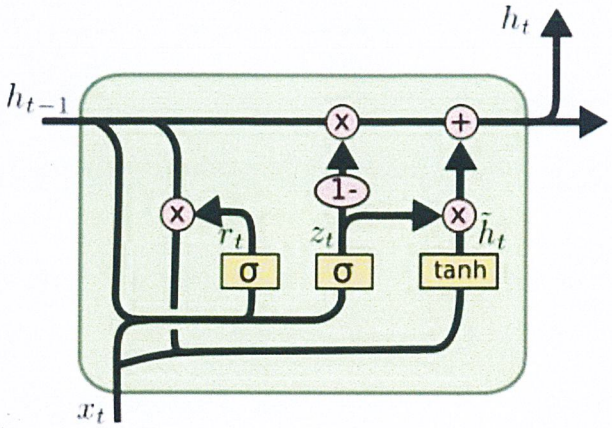
$$h_t = o_t * \tanh(C_t)$$



ภาพที่ 2.7 การคำนวณผลลัพธ์ของ Output Vector

2.2.2 Gated Recurrent Units

Gated Recurrent Units (GRU) เป็นโครงข่ายประสาทเทียมประเภทหนึ่ง ที่มีการทำงานคล้าย LSTM แต่ต่างกันตรงที่ GRU นั้นไม่มี Output Gate ในการคำนวณด้านใน โดยความสามารถของ GRU สามารถคำนวณข้อมูลประเภทลำดับ (Sequential Data) และข้อมูลเชิงเวลา (Temporal Data) ได้ เช่นข้อมูล Speech Recognition, Natural Language Processing และ Machine Translation กระบวนการทำงานภายใน GRU แสดงดังภาพที่ 2.8



ภาพที่ 2.8 การทำงานภายใน GRU

เมื่อ h_t คือ Output Vector

x_t คือ Input Vector

r_t คือ Read Gate

z_t คือ Forget Gate

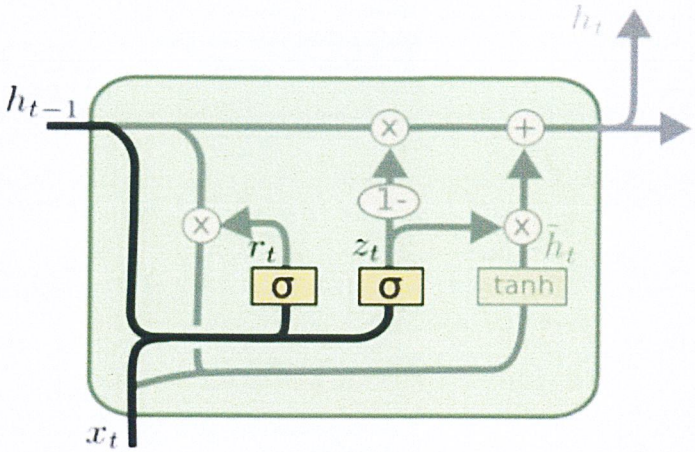
$\tilde{h}_t$ คือ State Candidate

หลักการทำงานของ GRU

อันดับแรก GRU จะอ่านค่าเพื่อปรับปรุงค่า Read Gate และ Forget Gate ในการคำนวณจะใช้ข้อมูลนำเข้า h_{t-1} และ x_t โดยผ่าน Read Gate และ Forget Gat ขั้นตอนการคำนวณ Read Gate และ Forget Gate แสดงดังภาพที่ 2.9

$$z_t = \sigma(W_z * [h_{t-1}, x_t])$$

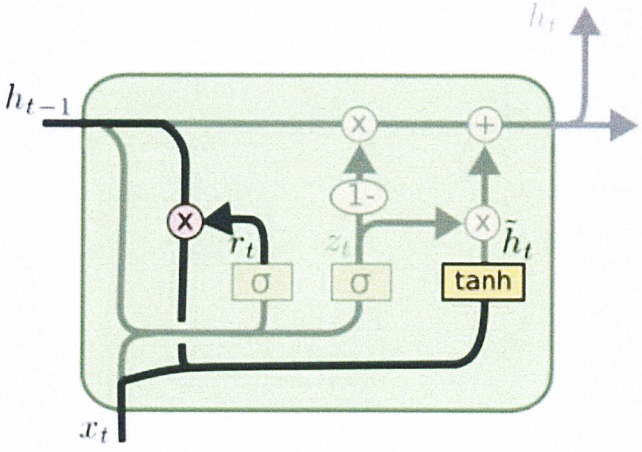
$$r_t = \sigma(W_r * [h_{t-1}, x_t])$$



ภาพที่ 2.9 การคำนวณ Read Gate และ Forget Gate

เมื่อได้ข้อมูลจาก Read Gate จะนำไปคำนวณ State Candidate โดยการนำข้อมูลเข้า Read Gate และ x_t ผ่านฟังก์ชัน Hyperbolic Tangent ขั้นตอนการคำนวณ State Candidate แสดงดังภาพที่ 2.10

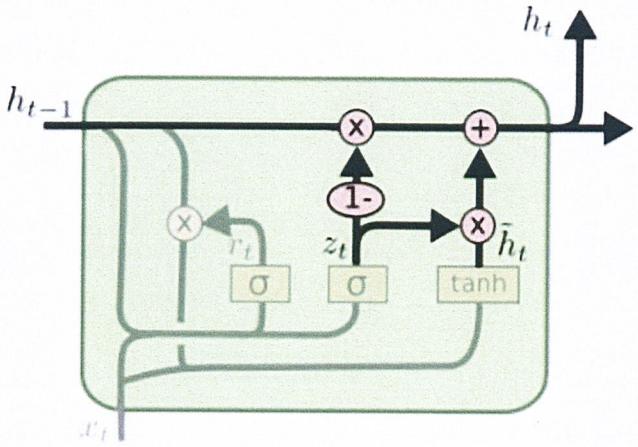
$$\widetilde{h_t} = \tanh\left(W * [r_t * h_{t-1}, x_t]\right)$$



ภาพที่ 2.10 การคำนวณ State Candidate

จากนั้นจะเป็นการคำนวณผลลัพธ์ (h_t) โดยการนำ *1 - z_t* (Write Gate) ซึ่งเป็นตัวควบคุมว่าจะเขียนอะไรลงไป และคูณกับข้อมูลรอบที่แล้ว (h_{t-1}) จากนั้นรวมเข้าด้วยกันกับ $z_t * \tilde{h}_t$ ซึ่ง z_t จะเป็นตัวควบคุมที่จะกำหนดว่าว่าจะลืมค่าหรือไม่ เมื่อคำนวณเสร็จได้เป็นผลลัพธ์ (h_t) ส่งไปยังรอบต่อไป การคำนวณผลลัพธ์แสดงดังภาพที่ 2.11

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h_t}$$



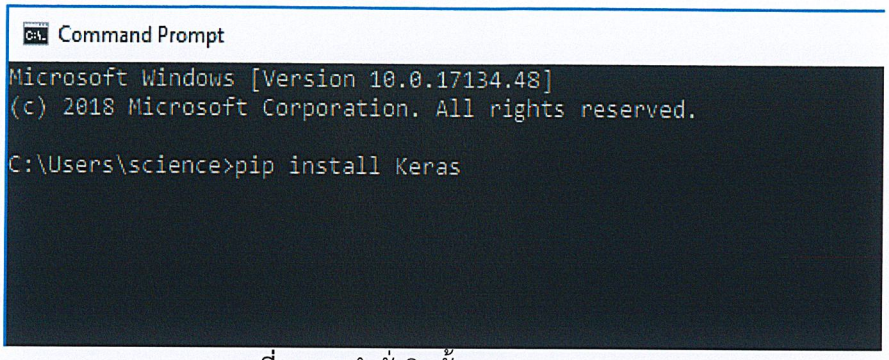
ภาพที่ 2.11 การคำนวณผลลัพธ์

2.3 Keras Framework

Keras เป็น Neural Network Framework แบบ Open Source ที่พัฒนาด้วยภาษา Python และเป็น high-level API ที่มีประสิทธิภาพในการสร้างแบบจำลองโครงข่ายประสาทเทียมและทำการเรียนรู้เชิงลึก (Deep Learning) ซึ่งง่ายต่อการใช้งานและสามารถเลือกใช้งานร่วมกับ Library อื่นได้เช่น Tensorflow และ Theano

2.3.1 การติดตั้ง Keras Framework

ในหัวข้อนี้จะกล่าวถึงการติดตั้ง Keras Framework ซึ่งมีข้อกำหนด คือ ต้องมีการติดตั้ง Python มาก่อน จึงจะสามารถติดตั้ง Keras Framework ได้แสดงดังภาพที่ 2.12

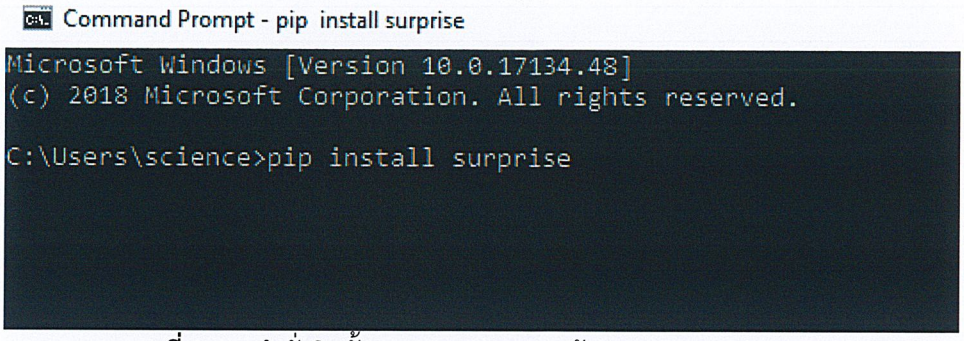


ภาพที่ 2.12 คำสั่งติดตั้ง Keras Framework

ภาพที่ 2.12 แสดงการติดตั้ง Keras Framework โดยใช้ Command Prompt ซึ่งหลังจากที่ติดตั้ง Python และตั้งค่า Environment เรียบร้อยแล้ว เราสามารถติดตั้ง Keras Framework ได้โดยพิมพ์คำสั่ง pip install Keras ใน Command Prompt หลังจากนั้นคอมพิวเตอร์จะทำการดาวน์โหลดและติดตั้ง Keras Framework ให้ โดยอัตโนมัติ

2.4 Surprise Library

Surprise library เป็นหนึ่งใน Library ที่ถูกพัฒนาด้วยภาษา Python เพื่อใช้สร้างแบบจำลองวิธีการกรองด้วยความร่วมมือ (Collaborative Filtering) โดย Surprise จะวัดประสิทธิภาพด้วยการแบ่งข้อมูลเป็นชุดก่อน (K-Fold cross validation) แล้วจึงนำข้อมูลไปประมวลผล ในการติดตั้งสามารถพิมพ์คำสั่ง pip install surprise ได้ใน Command Prompt ดังตัวอย่างในภาพที่ 2.13



ภาพที่ 2.13 คำสั่งติดตั้ง Surprise library ด้วย Command Prompt

จากภาพที่ 2.13 เมื่อพิมพ์คำสั่งเสร็จคอมพิวเตอร์จะทำการดาวน์โหลดและติดตั้ง Library ให้โดยอัตโนมัติ

2.5 Confusion Matrix

ก่อนนำโมเดลไปใช้จริงเราจำเป็นต้องวัดประสิทธิภาพโมเดลเพื่อดูว่าโมเดลที่สร้างมีประสิทธิภาพหรือไม่ โดย Confusion Matrix เป็นเทคนิคการวัดประสิทธิภาพการพยากรณ์ของโมเดลก่อนที่จะนำไปใช้งานจริง ซึ่งจะเอาผลการพยากรณ์ของโมเดลเปรียบเทียบกับคลาสคำตอบของชุดข้อมูลดังตารางที่ 2.2

ตารางที่ 2.2 Confusion Matrix

		Actual Class	
		Yes	No
Predicted Class	Yes	True Positive (TP)	False Positive (FP)
	No	False Negative (FN)	True Negative (TN)

จากตารางที่ 2.2 ภายในตาราง Confusion Matrix เป็นจำนวนของการเปรียบเทียบโดยที่

- True Positive (TP) : คลาสคำตอบคือ “Yes” โมเดลพยากรณ์ว่า “Yes”
- True Negative (TN) : คลาสคำตอบคือ “No” โมเดลพยากรณ์ว่า “No”
- False Positive (FP) : คลาสคำตอบคือ “No” โมเดลพยากรณ์ว่า “Yes”
- False Negative (FN) : คลาสคำตอบคือ “Yes” โมเดลพยากรณ์ว่า “No”

ในงานวิจัยนี้จะใช้ Confusion Matrix เพื่อวัดความแม่นยำในการพยากรณ์ของโมเดล (Accuracy) หาได้จากสูตร $\frac{TP+TN}{TP+TN+FP+FN}$

ในการคำนวนค่าความแม่นยำด้วย Confusion Matrix เราจะยกตัวอย่างค่าการเปรียบเทียบระหว่างคลาสคำตอบและการพยากรณ์ของโมเดลแสดงดังตารางที่ 2.3

ตารางที่ 2.3 ตัวอย่าง Confusion Matrix ที่มีค่าเปรียบเทียบระหว่างคลาสคำตอบและการพยากรณ์ของโมเดล

		Actual Class	
		Yes	No
Predicted Class	Yes	9	3
	No	9	9

จากตารางที่ 2.3 ค่าเปรียบเทียบระหว่างคลาสคำตอบและการพยากรณ์ของโมเดลประกอบไปด้วย

- True Positive (TP) : คลาสคำตอบคือ “Yes” โมเดลพยากรณ์ว่า “Yes” มี 9 ครั้ง
- True Negative (TN) : คลาสคำตอบคือ “No” โมเดลพยากรณ์ว่า “No” มี 9 ครั้ง
- False Positive (FP) : คลาสคำตอบคือ “No” โมเดลพยากรณ์ว่า “Yes” มี 3 ครั้ง
- False Negative (FN) : คลาสคำตอบคือ “Yes” โมเดลพยากรณ์ว่า “No” มี 9 ครั้ง

ดังนั้นค่าความแม่นยำจากสูตร $\frac{TP+TN}{TP+TN+FP+FN}$ จะนำค่าภายใน Confusion Matrix มาใช้โดยจะหาได้ดังนี้

$$Accuracy = \frac{9+9}{9+9+3+9} = \frac{18}{30} = 60\%$$

2.6 Mean Absolute Error : MAE

ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Error : MAE) ถูกใช้บ่อยครั้งเพื่อจะประเมินระบบแนะนำ เป็นวิธีการหาค่าเฉลี่ยของความแตกต่างสัมบูรณ์ระหว่างค่าพยากรณ์และคะแนนที่ผู้ใช้ได้ให้ไว้กับภาพยนต์ หากค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยมีค่าน้อย แสดงว่าค่าพยากรณ์สามารถประมาณค่าได้ใกล้เคียงกับค่าคะแนนที่ผู้ใช้ได้ให้ไว้โดยสามารถหาค่าได้จาก

$$Mean\ Absolute\ Error = \frac{1}{N}\sum_{i=1}^{N}|p_i - r_i|$$

เมื่อ p_i คือ คะแนนที่ผู้ใช้ได้ให้ไว้กับภาพยนต์เรื่องที่ i, r_i เป็นค่าคะแนนที่โมเดลพยากรณ์ให้กับเรื่อง i และ N คือจำนวนคู่ของคะแนนทั้งหมดที่นำมาเปรียบเทียบกัน

2.7 งานวิจัยที่เกี่ยวข้อง

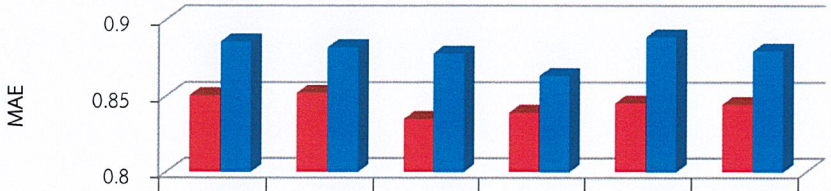
หัวข้อนี้จะกล่าวถึงงานวิจัยที่เกี่ยวข้องซึ่งประกอบไปด้วยงานวิจัยเรื่องวิธีการกรองด้วยความร่วมมือแบบแอลเอสไอสำหรับระบบแนะนำ เขียนโดย เกียรติชัย ปรีชาอนุสรณ์ และ ณัฐวัฒน์ โนนสังข์ (ปัญหาพิเศษของภาควิชาวิทยาการคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปี 2551) การศึกษาทดลองตัววัดความเหมือนสำหรับการกรองด้วยความร่วมมือ เขียนโดย ชลัท ศรีวิเศษสม ธีรเกียรติ พ่วงตามพงษ์ และ วิธวินท์ เทพศุภรังษิกุล (ปัญหาพิเศษของภาควิชาวิทยาการคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปี 2551) และ A Recurrent Neural Network Based Recommendation System เขียนโดย David Zhan Liu และ Gurbir Singh (Report จาก Department of Computer Science, Stanford University)

2.7.1 งานวิจัยเรื่องวิธีการกรองด้วยความร่วมมือแบบแอลเอสไอสำหรับระบบแนะนำ

งานวิจัยนี้ใช้ข้อมูลที่กลุ่มงานวิจัย Grouplens ได้เก็บสะสมเอาไว้ โดยข้อมูลที่มีผู้ใช้ในระบบ 943 คนให้คะแนนไว้ต่อภาพยนตร์ทั้งหมด 1,682 เรื่อง ช่วงของการให้คะแนนอยู่ในระหว่างหนึ่งถึงห้า โดยค่าคะแนนห้าแสดงถึงผู้ใช้ชอบภาพยนตร์เรื่องนั้นมากส่วนค่าคะแนนหนึ่งหมายถึงผู้ใช้นั้นมีความชอบต่อภาพยนตร์เรื่องนั้นน้อย และค่าคะแนนจะมีค่าเป็นศูนย์หมายถึงผู้ใช้ไม่ได้ให้คะแนนต่อภาพยนตร์เรื่องนั้นไว้ ซึ่งผู้ใช้ที่อยู่ในระบบแต่ละคนจะต้องให้คะแนนต่อภาพยนตร์อย่างน้อยที่สุด 20 เรื่อง โดยจำนวนค่าคะแนนที่มีทั้งหมดในชุดข้อมูลนี้คือ 100,004 ค่าคะแนน

ในการทดลองจากคะแนนทั้งหมด 100,004 ค่าคะแนนจะถูกแบ่งออกเป็นสองส่วน คือ ส่วนที่ใช้ในการสร้างระบบประมาณ 80,000 ค่าคะแนน (ร้อยละ 80) และส่วนที่ใช้ในการทดสอบผลประมาณ 20,000 ค่าคะแนน (ร้อยละ 20) โดยข้อมูลมีห้าชุด และข้อมูลแต่ละชุดไม่ซ้ำกันแสดงดังภาพที่ 2.12

กราฟเปรียบเทียบค่า MAE ระหว่างกรรมวิธีการกรองด้วย
ความร่วมมือแบบแอลเอสไอกับแบบดั้งเดิม



	Set 1	Set 2	Set 3	Set 4	Set 5	Average
LSI Collaborative (K=11)	0.850278	0.851837	0.834659	0.838842	0.844982	0.84412
Traditional Collaborative	0.885584	0.881864	0.877569	0.863273	0.88859	0.879376

ภาพที่ 2.14 กราฟเปรียบเทียบค่า MAE
ระหว่างวิธีการกรองด้วยความร่วมมือแบบแอลเอสไอกับแบบดั้งเดิม

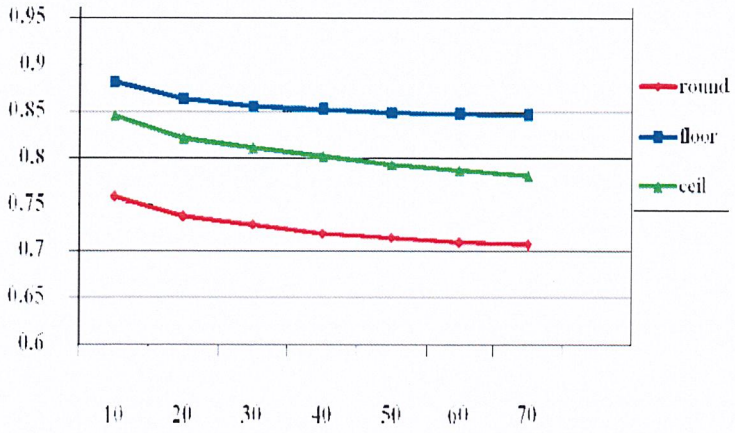
จากภาพที่ 2.14 แสดงถึงกราฟความร่วมมือแบบแอลเอสไอกับแบบดั้งเดิม จากผลการทดลอง จะเห็นได้ว่าวิธีการกรองด้วยความร่วมมือแบบแอลเอสไอแนะนำผลิตภัณฑ์ได้ดีกว่าวิธีการกรองด้วยความร่วมมือแบบดั้งเดิม โดยค่าเฉลี่ยความคลาดเคลื่อนสัมบูรณ์ของข้อมูลห้าชุดข้อมูลของวิธีการกรองด้วยความร่วมมือแบบแอลเอสไอนั้นมีค่าอยู่ที่ 0.84412 ส่วนวิธีการกรองด้วยความร่วมมือแบบดั้งเดิมมีค่าอยู่ที่ 0.879376

2.7.2 งานวิจัยเรื่องการศึกษาทดลองตัววัดความเหมือนสำหรับการกรองด้วยความร่วมมือ

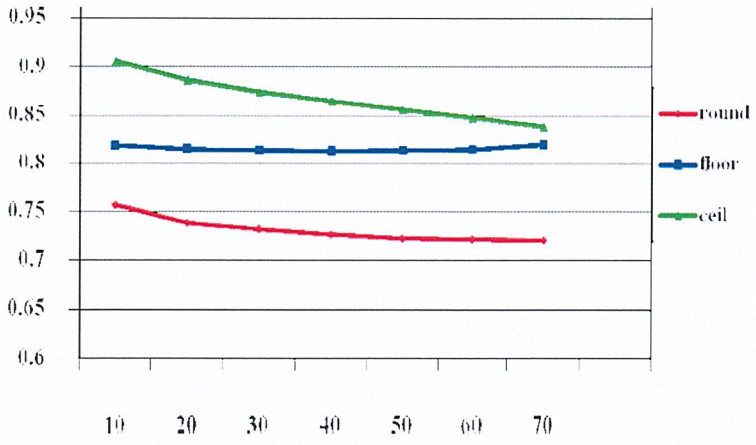
งานวิจัยนี้มีจุดประสงค์เพื่อจะหาว่าวิธีการแนะนำแบบการกรองด้วยความร่วมมือโดยใช้มาตรวัดความเหมือนสี่วิธี ได้แก่ ค่าสัมประสิทธ์สหสัมพันธ์ของเพียร์สัน ค่าความคล้ายคลึงของโคไซน์ ค่าสัมประสิทธิ์ของไดซ์ และค่าสหสัมพันธ์โดยการจัดลำดับของสเปียร์แมน วิธีการใดที่มีประสิทธิภาพมากที่สุดสำหรับระบบแนะนำ โดยใช้ชุดข้อมูลจาก Grouplens โดยข้อมูลมีผู้ใช้ในระบบ 943 คนให้คะแนนไว้ต่อภาพยนตร์ทั้งหมด 1,682 เรื่อง โดยค่าคะแนนจะอยู่ในช่วงหนึ่งถึงห้าขึ้นกับความชอบของแต่ละผู้ใช้ที่มีต่อภาพยนตร์ ซึ่งค่าคะแนนห้าแสดงว่าผู้ใช้นั้นมีความชอบต่อภาพยนตร์เรื่องนั้นมากที่สุด โดยผู้ใช้ต้องให้

คะแนนอย่างน้อยที่สุด 20 เรื่อง และจำนวนค่าคะแนนที่มีทั้งหมดในชุดข้อมูลนี้คือ 100,004 ค่าคะแนน

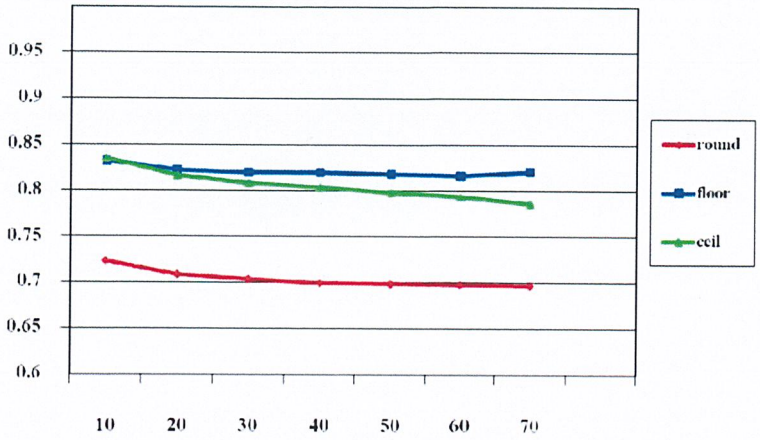
การประมวลผล จะได้ค่าของมาตราวัดความเหมือนที่เป็นเลขจำนวนจริง ซึ่งไม่สามารถนำมาใช้เป็นคำตอบในการอ้างอิงได้ทันที จะต้องมีการปรับค่าตัวเลขก่อน โดยแบ่งอออกเป็นสามวิธี คือ Floor, Round และ Ceil ซึ่งวิธีปรับค่าที่ได้ค่าผิดพลาดสัมบูรณ์เฉลี่ยน้อยที่สุดคือ การปรับค่า Round แสดงดังภาพที่ 2.15-2.18



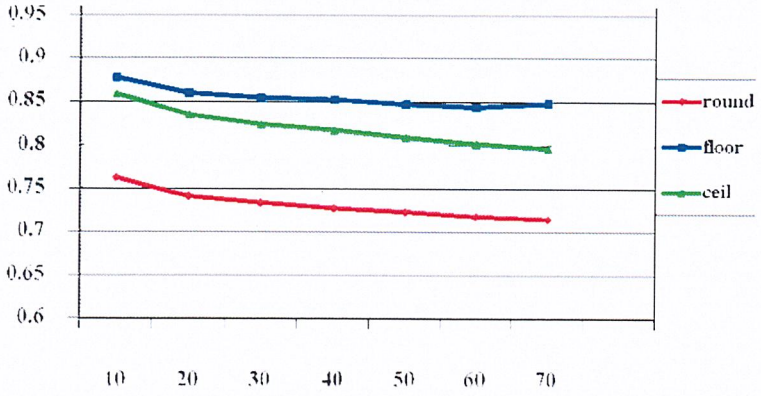
ภาพที่ 2.15 กราฟแสดงผลลัพธ์ของวิธีเพียร์สันด้วย MAE



ภาพที่ 2.16 กราฟแสดงผลลัพธ์ของวิธีโคไซน์ด้วย MAE

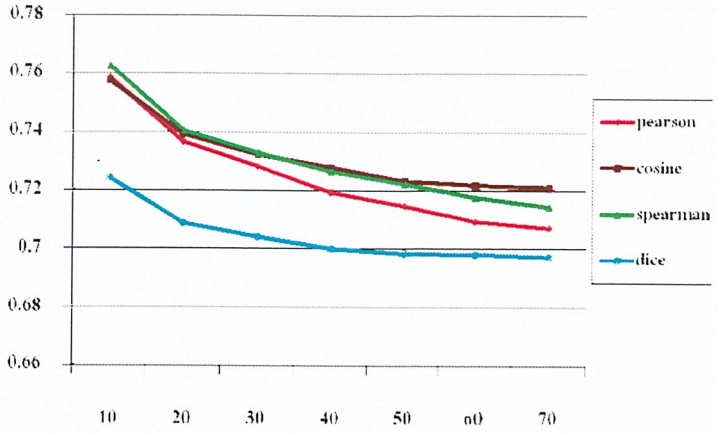


ภาพที่ 2.17 กราฟแสดงผลลัพธ์ของวิธีไดซ์ด้วย MAE



ภาพที่ 2.18 กราฟแสดงผลลัพธ์ของวิธีสเปียร์แมนด้วย MAE

เมื่อนำผลลัพธ์การคำนวณทั้ง 4 วิธีมาเปรียบเทียบ พบว่าผลลัพธ์ที่แม่นยำที่สุดคือวิธีหาค่าสัมประสิทธิ์ของไดซ์ ที่ให้ค่าความผิดพลาดสัมบูรณ์เฉลี่ยต่ำที่สุดแสดงดังภาพที่ 2.17



ภาพที่ 2.19 กราฟแสดงค่าความผิดพลาดสัมบูรณ์เฉลี่ยทั้ง 4 วิธี

2.7.3 งานวิจัยเรื่อง A Recurrent Neural Network Based Recommendation System

งานวิจัยนี้นำเสนอการเปรียบเทียบประสิทธิภาพของโครงข่ายประสาทเทียมสองประเภท ได้แก่ Gate Recurrent Unit (GRU) และ Long Short-Term Memory (LSTM) และใช้ Support Vector Machine (SVM) มาเป็นวิธีการพื้นฐานในการเปรียบเทียบผลลัพธ์ ชุดข้อมูลทดสอบของงานวิจัยนี้นำมาจากเว็บไซต์ Yelp Data Challenge ซึ่งเป็นข้อมูลเกี่ยวกับการให้คะแนนความพึงพอใจในการใช้บริการร้านอาหาร ชุดข้อมูลประกอบไปด้วยการให้คะแนน 1,231,275 ครั้ง จาก 27,882 ร้านอาหาร จากผลการทดลองเปรียบเทียบประสิทธิภาพของ SVM, GRU และ LSTM ได้ผลลัพธ์ออกมา ดังตารางที่ 2.2

ตารางที่ 2.4 ผลลัพธ์จากการทดลองเปรียบเทียบประสิทธิภาพของ SVM, GRU และ LSTM

	Train Accuracy	Validation Accuracy	Test Accuracy
SVM	87.25	82.00	76.25
GRU	99.60	93.25	82.75
LSTM	93.70	87.00	81.74

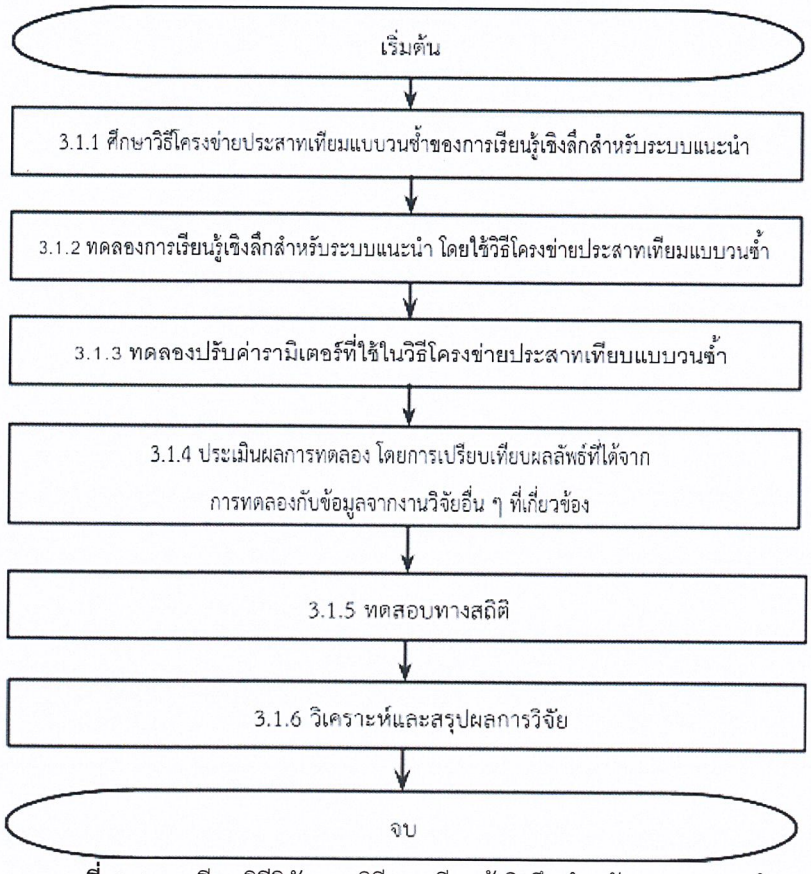
จากตารางที่ 2.4 สรุปได้ว่าวิธีการแบบ GRU ให้ประสิทธิภาพที่แม่นยำที่สุดเมื่อเทียบกับวิธีการแบบ LSTM และ SVM

บทที่ 3
วิธีการดำเนินงานวิจัย

ในบทนี้จะกล่าวถึงการดำเนินการวิจัยเรื่องการเรียนรู้เชิงลึกสำหรับระบบแนะนำ โดยเนื้อหาประกอบด้วยระเบียบวิธีวิจัย และ ขั้นตอนวิธีการทดลองระบบแนะนำโดยใช้วิธีการเรียนรู้เชิงลึก รายละเอียดแสดงในหัวข้อ 3.1 และ 3.2

3.1 ระเบียบวิธีวิจัย

การศึกษาวิธีการต่าง ๆ ของการเรียนรู้เชิงลึกสำหรับระบบแนะนำ ซึ่งมีลำดับการดำเนินงานดังแสดงในตารางภาพที่ 3.1 โดยที่รายละเอียดของแต่ละขั้นตอน ได้ถูกอธิบายในหัวข้อที่ 3.1.1 ถึง 3.1.6



ภาพที่ 3.1 ระเบียบวิธีวิจัยของวิธีการเรียนรู้เชิงลึกสำหรับระบบแนะนำ

3.1.1 ศึกษาวิธีการต่าง ๆ ของวิธีการเรียนรู้เชิงลึกสำหรับระบบแนะนำ

ศึกษาวิธีการต่าง ๆ ที่เกี่ยวข้อง ได้แก่ วิธี (Recurrent Neural Network : RNN) แบบ Long Short-Term Memory และ Gated Recurrent Units

3.1.2 ทดลองการเรียนรู้เชิงลึกสำหรับระบบแนะนำโดยใช้วิธีการต่าง ๆ ที่ได้ศึกษามา

ทดลองวิธีการที่ได้ศึกษามากับชุดข้อมูลทดสอบจำนวน 2 ชุด ได้แก่

1. ชุดข้อมูล MovieLens ประกอบด้วยการให้คะแนนความพึงพอใจของผู้ใช้งานจำนวน 100,004 ครั้ง ที่มีต่อภาพยนตร์จำนวน 1,682 เรื่อง และจากผู้ใช้งานจำนวน 943 คน (ข้อมูลเผยแพร่เมื่อเดือนเมษายน ปี ค.ศ. 1998)
2. ชุด Amazon Books ประกอบด้วยการให้คะแนนความพึงพอใจของหนังสือที่ขายบนเว็บไซต์ Amazon.com จำนวน 8,898,041 รีวิว (ข้อมูลตั้งแต่เดือนพฤษภาคม ค.ศ. 1996 จนถึงเดือนกรกฎาคม ค.ศ. 2014) ซึ่งการทดลองนี้จะสุ่มดึงข้อมูลมาใช้ทดสอบ 500,000 รีวิว จากข้อมูลรีวิวทั้งหมด

3.1.3 ทดลองปรับค่าพารามิเตอร์ที่ใช้ในวิธีการต่าง ๆ

ทดลองปรับค่าพารามิเตอร์ของวิธีการต่าง ๆ ที่ได้ศึกษามา เพื่อทดสอบดูว่าชุดของค่าพารามิเตอร์ชุดไหนที่ให้ผลลัพธ์ที่ดีที่สุด สำหรับการทดลอง โดยใช้วิธีการต่าง ๆ

3.1.4 ประเมินผลการทดลอง

ประเมินผลการทดลอง โดยการเปรียบเทียบกับผลลัพธ์ที่ได้กับผลลัพธ์จากงานวิจัยอื่น ๆ คืองานวิจัยเรื่องวิธีการกรองด้วยความร่วมมือแบบแอลเอสไอสำหรับระบบแนะนำ เขียนโดย เกียรติชัย ปรีชาอนุสรณ์ และ ณัฐวัฒน์ โนนสังข์ (ปัญหาพิเศษของภาควิชาวิทยาการคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปี 2551) และการศึกษาทดลองตัววัดความเหมือนสำหรับการกรองด้วยความร่วมมือ เขียนโดย ชลัท ศรีวิเศษสม ธีรเกียรติ พ่วงตามพงษ์ และ วิธวินท์ เทพศุภรังษิกุล (ปัญหาพิเศษของภาควิชาวิทยาการคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปี

2551) โดยจะเน้นที่การวัดความแม่นยำ (Accuracy) และค่าเฉลี่ยความคลาดเคลื่อนสัมบูรณ์ (MAE) ที่ได้จากการทดลองที่ใช้ชุดข้อมูลทดสอบชุดเดียวกัน

3.1.5 ทดสอบทางสถิติ

ทดสอบทางสถิติ เพื่อพิจารณาว่าผลลัพธ์ที่ได้จากการทดลองและเปรียบเทียบกับงานวิจัยอื่น ๆ มีความแตกต่างกันอย่างมีนัยสำคัญหรือไม่

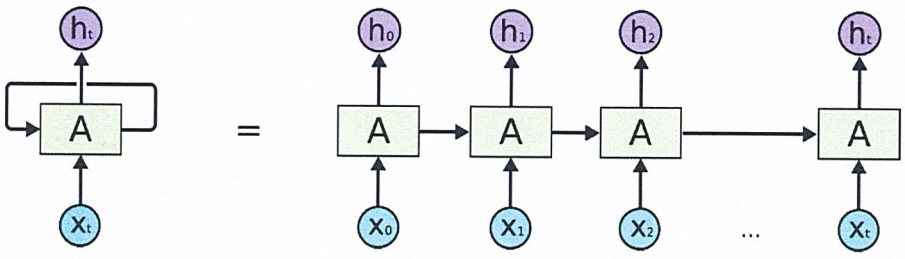
3.1.6 วิเคราะห์และสรุปผลการวิจัย

ในขั้นตอนนี้จะนำผลลัพธ์ที่ได้จากการทดลองมาวิเคราะห์ เพื่อหาองค์ความรู้ที่ได้รับจากการทดลองและเปรียบเทียบกับงานวิจัยอื่น ๆ ว่ามีปัจจัยอะไรบ้างที่ส่งผลต่อผลลัพธ์ของการทดลอง

3.2 ขั้นตอนวิธีการทดลองระบบแนะนำโดยใช้วิธีการเรียนรู้เชิงลึก

ในหัวข้อนี้จะกล่าวถึงการดำเนินงานทดลองกระบวนการของโครงข่ายประสาทเทียมแบบวนซ้ำที่นำมาใช้กับระบบแนะนำ

3.2.1 ขั้นตอนการใช้โครงข่ายประสารทเทียมแบบวนซ้ำกับระบบแนะนำ



ภาพที่ 3.2 การทำงานภายในโครงข่ายประสาทเทียมแบบวนซ้ำ

ภาพที่ 3.2 แสดงถึงการทำงานของโครงข่ายประสาทเทียมแบบวนซ้ำ การทำงานประกอบไปด้วยส่วนของข้อมูลนำเข้า ส่วนของการประมวลผล และส่วนของผลลัพธ์การคำนวณ

ส่วนของข้อมูลนำเข้า

การนำข้อมูลเข้าจะใช้ชุดข้อมูลจาก MovieLens และ Amazon Books ดังที่กล่าวไปแล้วในหัวข้อ 3.1.2 ซึ่งตัวอย่างข้อมูลที่นำมาใช้งานมีดังนี้

ข้อมูลของชุดข้อมูล MovieLens ประกอบไปด้วยไฟล์ u.data, u.item และ u.user ซึ่งแต่ละไฟล์มีรายละเอียดดังต่อไปนี้

ไฟล์ u.data มีข้อมูลของผู้ใช้งานที่ได้ให้คะแนนความพึงพอใจไว้กับภาพยนตร์ซึ่งประกอบไปด้วยคุณลักษณะของข้อมูล 4 คุณลักษณะ แสดงดังตารางที 3.1 โดยที่แต่ละคุณลักษณะมีรายละเอียดต่อไปนี้

user id คือ รหัสของผู้ใช้งาน

item id คือ รหัสของสิ่งของ

rating คือ คะแนนความพึงพอใจ

timestamp คือ เวลาที่ผู้ใช้งานให้คะแนนความพึงพอใจ

ตารางที่ 3.1 ข้อมูลของผู้ใช้งานที่ให้คะแนนไว้กับภาพยนตร์ในชุดข้อมูล MovieLens

user id	item id	rating	timestamp
196	242	3	881250949
186	302	3	891717742
22	377	1	878887116
244	51	2	880606923

ไฟล์ u.item มีข้อมูลของภาพยนตร์ ประกอบไปด้วยคุณลักษณะ 23 คุณลักษณะ โดยที่ 4 คุณลักษณะแรกมีรายละเอียดดังนี้

movie id	คือ รหัสของภาพยนตร์
release date	คือ วันที่ภาพยนตร์เริ่มฉาย
video release date	คือ วันที่ภาพยนตร์เริ่มมีการจำหน่ายวีดีทัศน์
IMDb URL	คือ เว็บไซต์ของภาพยนตร์ในฐานข้อมูลภาพยนตร์ออนไลน์ (The Internet Movie Database)

สำหรับ 19 คุณลักษณะสุดท้ายจะบ่งบอกถึงประเภทของภาพยนตร์ที่เกี่ยวข้องซึ่งในภาพยนตร์หนึ่งเรื่องสามารถมีได้หลายประเภท ประเภทของภาพยนตร์ประกอบไปด้วย Unknown, Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War และWestern โดยที่ข้อมูลในไฟล์ u.item จะเก็บค่า 1 ถ้าภาพยนตร์นั้น ๆ กับประเภทของภาพยนตร์นั้น ๆ มีความเกี่ยวข้องกัน และจะเก็บค่า 0 ถ้าไม่มีความเกี่ยวข้องกัน ข้อมูลในไฟล์ u.item มีตัวอย่างดังนี้

1|Toy Story (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Toy%20Story%20(1995)|0|0|0|1|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0

2|GoldenEye (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?GoldenEye%20(1995)|0|1|1|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0

3|Four Rooms (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Four%20Rooms%20(1995)|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0

4|Get Shorty (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Get%20Shorty%20(1995)|0|1|0|0|0|1|0|0|1|0|0|0|0|0|0|0|0|0|0

ไฟล์ u.user มีข้อมูลของผู้ใช้งาน ประกอบไปด้วยคุณลักษณะ 5 คุณลักษณะ แสดงดังตารางที่ 3.2 โดยที่แต่ละคุณลักษณะมีรายละเอียดต่อไปนี้ดังนี้

user id คือ รหัสของผู้ใช้งาน
age คือ อายุ
gender คือ เพศ
occupation คือ อาชีพ
zip code คือ รหัสไปรษณีย์

ตารางที่ 3.2 ข้อมูลของผู้ใช้งานในชุดข้อมูล MovieLens

user id	age	gender	occupation	zip code
1	24	M	technician	85711
2	53	F	other	94043
3	23	M	writer	32067
4	24	M	technician	43537

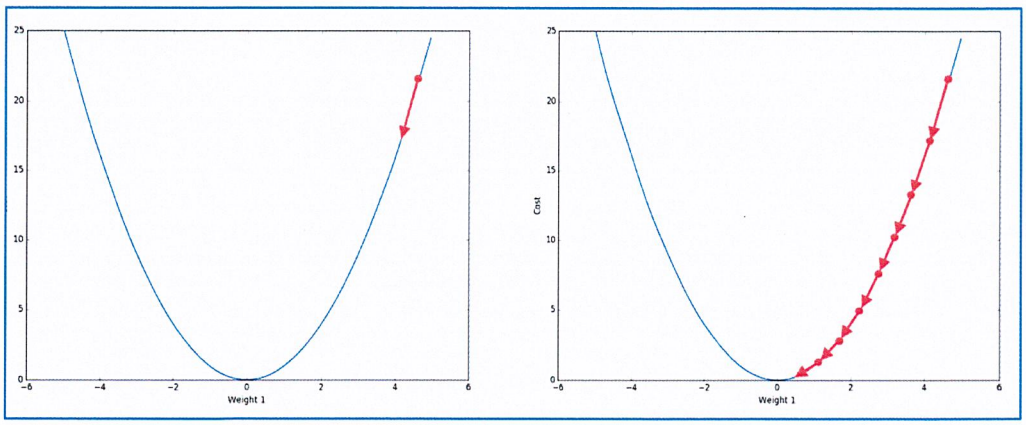
ข้อมูลของชุดข้อมูล Amazon Book มีตัวอย่างดังต่อไปนี้

{
"reviewerID": "A2SUAM1J3GNN3B",
"asin": "0000013714",
"reviewerName": "J. McDonald",
"helpful": [2, 3],
"reviewText": "I bought this for my husband who plays the piano. He is having a wonderful time playing these old hymns. The music is at times hard to read because we think the book was published for singing from more than playing from. Great purchase though!",
"overall": 5.0,
"summary": "Heavenly Highway Hymns",
"unixReviewTime": 1252800000,
"reviewTime": "09 13, 2009"
}

โดยที่ reviewerID คือ รหัสของการให้คะแนนความพึงพอใจ
asin คือ รหัสของหนังสือ
reviewerName คือ ชื่อของผู้ให้คะแนนความพึงพอใจ
helpful คือ คะแนนคุณภาพของการให้คะแนนความพึงพอใจครั้งนั้น
reviewText คือ ข้อความที่ผู้ใช้งานเขียนถึงความพึงพอใจ
overall คือ คะแนนความพึงพอใจต่อหนังสือเล่มนั้น ๆ
summary คือ ข้อความสรุปของการให้คะแนนความพึงพอใจ
unixReviewTime คือ เวลาที่ให้คะแนนความพึงพอใจ (เก็บในรูปของ Unix time)
reviewTime คือ เวลาที่ให้คะแนนความพึงพอใจ

ส่วนของการประมวลผล

ในการประมวลผลจะปรับค่า Weights และ Biases เพื่อให้ค่า Loss Function มีค่าน้อยที่สุด แสดงดังภาพที่ 3.3



ภาพที่ 3.3 แสดงการประมวลผลเพื่อปรับค่า Weights และ Biases

ส่วนของผลลัพธ์การคำนวณ

ผลลัพธ์ที่ได้จะนำไปเปรียบเทียบกับวิธีการต่าง ๆ ของระบบแนะนำจากงานวิจัยที่กล่าวถึงในบทที่ 2 เพื่อได้ทราบถึงวิธีการที่เหมาะสมสำหรับระบบแนะนำ โดยในการเขียนคำสั่งเพื่อคำนวนผลลัพธ์จะใช้ Keras Framework และ Surprise Library ที่ได้ไว้ในหัวข้อที่

2.3 และ 2.4 ทดสอบกับชุดข้อมูล MovieLens และ Amazon อธิบายในหัวข้อที่ 3.2.1.1 ถึง 3.2.1.3

3.2.1.1 การทดสอบชุดข้อมูล MovieLens โดยใช้โครงข่ายประสาทเทียมแบบวนซ้ำด้วย Keras Framework

ในหัวข้อนี้จะกล่าวถึงการใช้ Keras Framework ที่กล่าวไว้ในหัวข้อที่ 2.3 ทดสอบกับชุดข้อมูล MovieLens

การเรียกใช้ Keras Framework สามารถพิมพ์คำสั่ง import keras ได้ในโปรแกรมโดยจะหมายถึงการเรียกใช้ทุกฟังก์ชันทั้งหมดของ Keras แต่ในปัญหาพิเศษนี้จะเลือกใช้ฟังก์ชันบางส่วนตามภาพที่ 3.4

```
from keras.layers import Dense
from keras.layers import GRU, LSTM
from keras.layers.embeddings import Embedding
from keras.models import Sequential
```

ภาพที่ 3.4 คำสั่งเรียกใช้งาน Keras Framework

จากภาพที่ 3.4 เป็นตัวอย่างฟังก์ชันบางส่วนของ Keras ที่จะนำไปใช้สร้างโครงข่ายประสาทเทียมแบบวนซ้ำประกอบไปด้วย

- from keras.layers import Dense หมายถึง การเรียกใช้ตัวกำหนดความหนาแน่นของข้อมูล
- from keras.layers import GRU, LSTM หมายถึง การเรียกใช้โมเดล GRU และ LSTM
- from keras.layers.embeddings import Embedding หมายถึง การเรียกใช้ตัวกำหนดจำนวนข้อมูลนำเข้าที่เข้าสู่โหนดภายใน
- from keras.models import Sequential หมายถึง การเรียกใช้โมเดล Sequential

ข้อมูลที่จะใช้ฝึกสอนและทดสอบโมเดลจะถูกแบ่งออกเป็นสองประเภท ได้แก่ ส่วนข้อมูลนำเข้าและคลาสคำตอบโดยคำสั่งที่ใช้แสดงดังภาพที่ 3.5

```
dataset = numpy.loadtxt("movlen_nor4_5.csv", skiprows=1, delimiter=",")
X = dataset[:,0:24]
Y = dataset[:,24:25]


X = numpy.array(X).tolist()
Y =  numpy.array(Y).tolist()
```

ภาพที่ 3.5 คำสั่งดึงข้อมูลจากไฟล์และแบ่งข้อมูล

จากภาพที่ 3.5 ตัวแปร X จะเก็บข้อมูลนำเข้าส่วนตัวแปร Y เก็บคลาสคำตอบที่จะใช้เพื่อฝึกสอนและทดสอบโมเดล

ขั้นตอนต่อไปคือการแบ่งข้อมูลจากตัวแปร X และ Y ให้เป็นชุดข้อมูลฝึกสอน 80% และชุดข้อมูลทดสอบ 20% และสุดท้าย คือ การสร้างโครงข่ายประสาทเทียมแบบวนซ้ำซึ่งคำสั่งที่ใช้แสดงได้ดังภาพที่ 3.6 และ 3.7

```
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=seed)
X_train = sequence.pad_sequences(X_train, maxlen=max_review_length)
X_test = sequence.pad_sequences(X_test, maxlen=max_review_length)
```

ภาพที่ 3.6 คำสั่งแบ่งข้อมูลออกเป็นชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบ

```
model = Sequential()
model.add(Embedding(top_words, embedding_vecor_length, input_length=max_review_length))
model.add(LSTM(256, dropout = 0.2))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='mae', optimizer='sgd', metrics=['acc'])
print(model.summary())
model.fit(X_train, y_train, nb_epoch=3, batch_size=32)
# Final evaluation of the model
scores = model.evaluate(X_test, y_test, verbose=0)
```

ภาพที่ 3.7 คำสั่งสร้างโครงข่ายประสาทเทียมแบบวนซ้ำสำหรับชุดข้อมูล MovieLens

จากภาพที่ 3.6 ตัวแปร X และ Y จะถูกแบ่งออกเป็นชุดข้อมูลฝึกสอน 80% แล้วเก็บไว้ที่ตัวแปร X_train กับ y_train ส่วนชุดข้อมูลทดสอบ 20% เก็บไว้ในตัวแปร X_test และ y_test

จากภาพที่ 3.7 เป็นตัวอย่างคำสั่งสร้างโครงข่ายประสาทเทียมแบบวนซ้ำโดยกำหนดให้เป็นแบบ LSTM โหนดภายเท่ากับ 256 และให้มีการคำนวนค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยและค่าความแม่นยำ โดยผลลัพธ์ที่ได้แสดงดังภาพที่ 3.8

```
Epoch 1/3
80000/80000 [==============================] - 64s 800us/step - loss: 0.4782
Epoch 2/3
80000/80000 [==============================] - 72s 901us/step - loss: 0.4557
Epoch 3/3
80000/80000 [==============================] - 68s 848us/step - loss: 0.4505
Accuracy: 55.27%
Runtime : 0:03:42.078291
```

ภาพที่ 3.8 ผลลัพธ์ของโครงข่ายประสาทเทียมแบบวนซ้ำเมื่อทดสอบกับชุดข้อมูล MovieLens

จากภาพที่ 3.8 แสดงตัวอย่างผลลัพธ์ของโครงข่ายประสาทเทียมแบบวนซ้ำเมื่อทดสอบกับชุดข้อมูล MovieLens ประกอบไปด้วย

- "loss" หมายถึง ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย
- "Accuracy" หมายถึง ความแม่นยำ
- "Runtime" หมายถึง เวลาที่ใช้ในการสร้างและทดสอบโมเดล

3.2.1.2 การทดสอบชุดข้อมูล Amazon โดยใช้โครงข่ายประสาทเทียมแบบวนซ้ำด้วย Keras Framework

ในหัวข้อนี้จะกล่าวถึงการนำ Keras Framework ที่กล่าวในหัวข้อ 2.3 ทดสอบกับชุดข้อมูล Amazon

เนื่องจากชุดข้อมูล Amazon เป็นข้อมูลประเภทข้อความ ซึ่งการวิเคราะห์ข้อความนั้นเราไม่สามารถใช้ข้อมูลที่เป็นตัวอักษรกับโครงข่ายประสาทเทียมแบบวนซ้ำได้โดยตรง เนื่องจากการทำงานของโครงข่ายประสาทเทียม โดยทั่วไปต้องใช้ข้อมูลนำเข้าประเภทตัวเลขเท่านั้น จึงต้องทำการแปลงข้อมูลนำเข้าที่เป็นตัวอักษรให้เป็นตัวเลขเสียก่อนโดยจะใช้วิธีการที่เรียกว่า Bag of Word (BoW) ผลลัพธ์ของกระบวนการนี้คือชุดตัวเลขที่แทนคำที่ปรากฏอยู่ในแต่ละทรานแซกชั่น ซึ่งคำสั่งที่ใช้แสดงดังภาพที่ 3.9 และ 3.10

```
def review_to_word(raw):
    letters_only = re.sub("[^a-zA-Z]", " ", str(raw))
    words = letters_only.lower().split()
    stops = set(stopwords.words('english'))
    success_words = [w for w in words if not w in stops]
    return (" ".join(success_words))

start_time = datetime.now()
num_reviews = len(review)
clean_all_review = list()
for each_review in range(0, num_reviews):
    if each_review == 10:
        break
    if each_review % 100000 == 0:
        print('INDEX : {}'.format(each_review))
    clean_all_review.append(review_to_word(review[each_review]))
print("Clean review's time : {}".format(datetime.now() - start_time))
```

ภาพที่ 3.9 คำสั่งสร้างฟังก์ชันในการทำความสะอาดข้อมูล

จากภาพที่ 3.9 เป็นตัวอย่างคำสั่งการทำความสะอาดข้อมูลโดยสร้าง และใช้ฟังก์ชันเพื่อนำข้อความมากำจัดคำที่ไม่สื่อความหมาย (Stop Word) และทำให้เป็นตัวอักษรตัวเล็กทั้งหมดแล้วนำมาจัดเรียงใหม่เนื่องจากคำ หรือขนาดของตัวอักษรไม่มีผลต่อการคำนวน หรืออาจทำให้การคำนวนของโครงข่ายประสาทเทียมแบบวนซ้ำผิดพลาดได้

```
vectorizer = CountVectorizer(analyzer="word", tokenizer=None,
                             preprocessor=None, stop_words=None, max_features=2000)
train_data_features = vectorizer.fit_transform(clean_all_review)
train_data_features = train_data_features.toarray()
```

ภาพที่ 3.10 คำสั่งแปลงข้อความเป็นชุดตัวเลข

จากภาพที่ 3.10 เป็นตัวอย่างคำสั่งการแปลงข้อความที่ทำความสะอาดข้อมูลเสร็จแล้วให้เป็นชุดตัวเลข ซึ่งเมื่อทำการแปลงข้อความให้เป็นชุดตัวเลขแล้วขั้นตอนต่อไปคือการสร้างโครงข่ายประสาทเทียมแบบวนซ้ำ โดยใช้คำสั่งแสดงตามภาพที่ 3.11

```
embedding_vecor_length = 32
model = Sequential()
model.add(Embedding(800, embedding_vecor_length, input_length=max_review_length))
model.add(GRU(1, dropout=0.2))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='mae', optimizer='sgd', metrics=['accuracy'])
print(model.summary())
model.fit(X_train, y_train, nb_epoch=3, batch_size=64)
scores = model.evaluate(X_test, y_test, verbose=0)
```

ภาพที่ 3.11 คำสั่งสร้างโครงข่ายประสาทเทียมแบบวนซ้ำสำหรับชุดข้อมูล Amazon

จากภาพที่ 3.11 เป็นตัวอย่างคำสั่งสร้างโครงข่ายประสาทเทียมแบบวนซ้ำชนิด GRU โดยกำหนดโหนดภายในเท่ากับ 1 ซึ่งกำหนดให้ตัวเลขนำเข้าของแต่ละชุดตัวเลขเท่ากับ 800 และให้มีการคำนวนค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยและค่าความแม่นยำ โดยผลลัพธ์ที่ได้จากคำสั่งแสดงดังรูปภาพที่ 3.12

```
Epoch 1/3
80000/80000 [==============================] - 252s 3ms/step - loss: 0.2469
Epoch 2/3
80000/80000 [==============================] - 245s 3ms/step - loss: 0.2123
Epoch 3/3
80000/80000 [==============================] - 261s 3ms/step - loss: 0.2120
Accuracy: 78.70%
Runtime : 0:13:01.146088
```

ภาพที่ 3.12 ผลลัพธ์ของโครงข่ายประสาทเทียมแบบวนซ้ำเมื่อทดสอบกับชุดข้อมูล Amazon

จากภาพที่ 3.12 แสดงตัวอย่างผลลัพธ์ของโครงข่ายประสาทเทียมแบบวนซ้ำเมื่อทดสอบกับชุดข้อมูล Amazon ประกอบไปด้วย

- “loss” หมายถึง ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย
- “Accuracy” หมายถึง ความแม่นยำ
- “Runtime” หมายถึง เวลาที่ใช้ในการสร้างและทดสอบโมเดล

3.2.1.3 ทดสอบชุดข้อมูล MovieLens และ Amazon โดยใช้ Surprise Library

ในหัวข้อนี้จะกล่าวถึงการนำ Surprise Library ที่กล่าวไว้ในหัวข้อที่ 2.4 ทดสอบกับชุดข้อมูล MovieLens และ Amazon

ในการเรียกใช้งาน Surprise library นั้นสามารถพิมพ์คำสั่ง import surprise ในโปรแกรมได้เลย ซึ่งคำสั่งนี้หมายถึงเรียกใช้ทุกฟังก์ชันใน Library นี้ โดยคำสั่งในการฝึกสอนและทดสอบโมเดลของวิธีการกรองด้วยความร่วมมือ (ระบบแนะนำแบบดั้งเดิม) แสดงได้ดังภาพที่ 3.13

```
from surprise import Reader
from surprise import SVD, evaluate, accuracy
from datetime import datetime

reader = Reader(line_format='user item rating', sep=' ')
data = Dataset.load_from_file('review.txt', reader=reader)

start = datetime.now()
data.split(n_folds=5)
algo = SVD()
evaluate(algo, data, measures=['MAE'])
accuracy.mae(predictions, verbose=True)
print('Runtime : {}'.format(datetime.now()-start))
```

ภาพที่ 3.13 คำสั่งฝึกสอนและทดสองโมเดลของวิธีการกรองด้วยความร่วมมือ

จากภาพที่ 3.13 ขั้นตอนวิธีที่ใช้ในการหาความคล้ายคลึงกัน (Similarity) ใน Matrix Factorization ได้แก่ Singular Value Decomposition (SVD) และให้ทำการแบ่งชุดข้อมูลด้วยวิธี 10-fold cross validation ผลลัพธ์ที่ได้จากคำสั่งนี้แสดงได้ดังภาพที่ 3.14

```
Mean MAE : 0.7323
------------
------------
MAE:  0.5328
Runtime : 0:02:53.834990

Process finished with exit code 0
```

ภาพที่ 3.14 ผลลัพธ์ของวิธีการกรองด้วยความร่วมมือ

จากภาพที่ 3.14 แสดงผลลัพธ์ของวิธีการกรองด้วยความร่วมมือประกอบไปด้วย

- “Mean MAE” หมายถึง ค่าเฉลี่ยของค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยที่ถูกแบ่งวิธี 10-fold cross validation
- “MAE” หมายถึง ค่าความแม่นยำในการทำนายผล
- “Runtime” หมายถึง เวลาที่ใช้ในการสร้างและทดสอบโมเดล

บทที่ 4

ผลการดำเนินงานและการอภิปรายผล

ในบทนี้จะกล่าวถึง ผลการดำเนินงาน การอภิปรายผลการดำเนินงาน การทดสอบทางสถิติ และปัญหาที่พบในการดำเนินงาน

4.1 ผลการดำเนินงาน

การดำเนินงานวิจัยนี้ได้ทำการแบ่งการทดลองออกเป็นสามการทดลองประกอบไปด้วย

1) ใช้โครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูลจาก MovieLens

2) ใช้โครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูลจาก Amazon

3) ใช้วิธีการกรองด้วยความร่วมมือ (วิธีการแบบดั้งเดิม) ทดสอบกับชุดข้อมูลจาก MovieLens

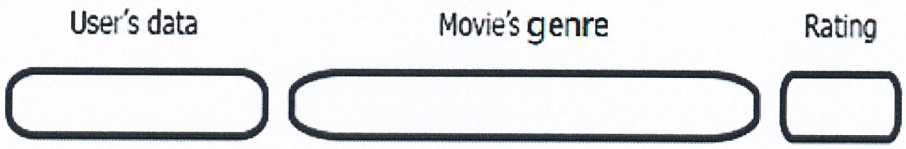
4) ใช้วิธีการกรองด้วยความร่วมมือ (วิธีการแบบดั้งเดิม) ทดสอบกับชุดข้อมูลจาก Amazon

4.1.1 ใช้โครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูลจาก MovieLens

ในการทดลองนี้จะนำโครงข่ายประสาทเทียบแบบวนซ้ำทั้งแบบ LSTM และ GRU ทดสอบกับชุดข้อมูลจาก MovieLens เพื่อหาค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย โดยได้ออกแบบการทดลองไว้ดังนี้

- กำหนดชุดคำตอบให้ “ผลลัพธ์ = แนะนำ” เมื่อการให้คะแนนความพึงพอใจ = 5 และ “ผลลัพธ์ = ไม่แนะนำ” เมื่อการให้คะแนนความพึงพอใจ = 1, 2, 3 และ 4
- กำหนดชุดคำตอบให้ “ผลลัพธ์ = แนะนำ” เมื่อการให้คะแนนความพึงพอใจ = 4 และ 5 และ“ผลลัพธ์ = ไม่แนะนำ” เมื่อการให้คะแนนความพึงพอใจ = 1, 2 และ 3
- ตัวแปรที่สำคัญที่ใช้ในการทดลองคือจำนวนรอบของการคำนวณเพื่อปรับค่าน้ำหนักที่เหมาะสม หรือจำนวนของโหนดภายใน (Hidden Nodes) ซึ่งในการทดลองได้ใช้จำนวนของโหนดภายในเท่ากับ 1, 2, 4, 8 ,16, 32, 64, 128, 256 และ 512 เพื่อทดสอบว่าจำนวนของโหนดภายในจะส่งผลต่อประสิทธิภาพและประสิทธิผลของระบบแนะนำอย่างไร

- สำหรับข้อมูลนำเข้าที่ใช้ในการสร้างโมเดลเพื่อทำนายความพึงพอใจให้กับระบบประกอบด้วย
    - ข้อมูลของผู้ใช้งาน ประกอบด้วย รหัสผู้ใช้งาน อายุ เพศ และอาชีพ
    - ข้อมูลของภาพยนตร์ ประกอบด้วย รหัสภาพยนตร์ และประเภทของภาพยนตร์
    - ข้อมูลการให้คะแนนความพึงพอใจของผู้ใช้ ที่มีค่าตั้งแต่ 1 ถึง 5



ภาพที่ 4.1 ภาพตัวอย่างของข้อมูลนำเข้า

ก่อนนำข้อมูลเข้าสู่โครงข่ายประสาทเทียมแบบวนซ้ำ ข้อมูลนำเข้าจะถูกแบ่งออกเป็นสองส่วน (Split validation) ส่วนหนึ่งเพื่อใช้สร้างแบบจำลอง (Model Training) จำนวน 80% และอีกส่วนเพื่อใช้ทดสอบแบบจำลอง (Model Testing) ที่ถูกสร้างขึ้น จำนวน 20%

ตารางสรุปผลการทดลองของ LSTM และ GRU เมื่อทดสอบกับชุดข้อมูลจาก MovieLens แสดงดังตารางที่ 4.1 ซึ่งจากการทดลอง ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยที่ได้มีแนวโน้มลดลงตามจำนวนโหนดภายในโดยค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยที่ดีที่สุดของการทดลองนี้คือ 0.2111 ซึ่งมาจากวิธีการแบบ GRU เมื่อกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 5 ตัวอักษรย่อต่าง ๆ ในตารางที่ 4.1, 4.3 และ 4.4 มีความหมายดังต่อไปนี้

- LSTM 4-5 คือ วิธีการแบบ LSTM และกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 4 และ 5

- LSTM 5 คือ วิธีการแบบ LSTM และกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 5

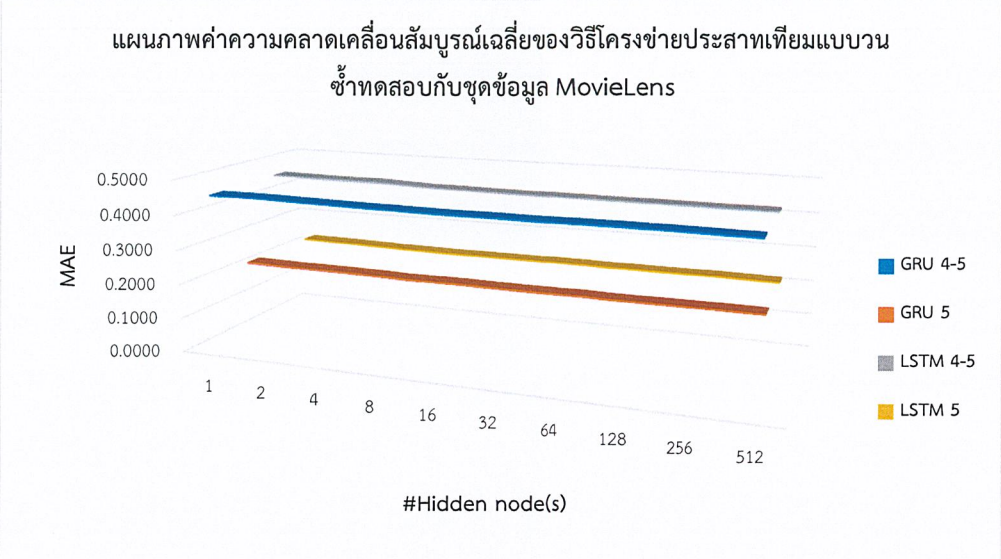
- GRU 4-5 คือ วิธีการแบบ GRU และกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 4 และ 5

- GRU 5 คือ วิธีการแบบ GRU และกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 5

ตารางที่ 4.1 ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของวิธีโครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูล MovieLens

Node(s)	Mean Absolute Error (MAE)			
	LSTM 4-5	LSTM 5	GRU 4-5	GRU 5
1	0.4587	0.2214	0.4533	0.2197
2	0.4571	0.2193	0.4536	0.2170
4	0.4572	0.2165	0.4505	0.2168
8	0.4516	0.2146	0.4492	0.2139
16	0.4498	0.2145	0.4476	0.2145
32	0.4485	0.2142	0.4480	0.2125
64	0.4474	0.2134	0.4473	0.2122
128	0.4472	0.2129	0.4487	**0.2111**
256	0.4466	0.2134	0.4482	0.2118
512	0.4466	0.2116	0.4472	0.2115

ตารางที่ 4.1 ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยมีแน้วโน้มลดลงตามจำนวนโหนดภายใน และสังเกตได้ว่าค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของวิธีโครงข่ายประสาทเทียมแบบวนซ้ำเมื่อกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 5 ดีกว่าวิธีโครงข่ายประสาทเทียมแบบวนซ้ำเมื่อกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 4 และ 5



ภาพที่ 4.2 แผนภาพค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของวิธีโครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูล MovieLens

4.1.2 ใช้โครงข่ายประสาทเทียมแบบวนซ้ำ ทดสอบกับชุดข้อมูลจาก Amazon

ชุดข้อมูลจาก Amazon มีจำนวน 8,898,041 รีวิว แต่เนื่องจากข้อจำกัดด้านทรัพยากรในการประมวลผล ดังนั้นในการทดลองนี้จะสุ่มใช้ชุดข้อมูลจาก Amazon จำนวน 500,000 รีวิวจากทั้งหมดมาใช้ในการประมวลผล โดยได้ออกแบบการทดลองดังต่อไปนี้

- ข้อมูลนำเข้าที่ใช้ประกอบด้วย reviewText และ overall ดังตัวอย่างต่อไปนี้

  {

  "reviewText": "I bought this for my husband who plays the piano. He is having a wonderful time playing these old hymns. The music is at times hard to read because we think the book was published for singing from more than playing from. Great purchase though!",

  "overall": 5.0,

  }
- reviewText เป็นข้อความรีวิวที่ผู้ใช้งานได้ให้ไว้กับภาพยนตร์เรื่องนั้น ๆ
- overall เป็นคะแนนที่ผู้ใช้งานได้ให้ไว้กับภาพยนตร์เรื่องนั้นโดยการทดลองจะแบ่งออกเป็นสองแบบได้แก่

- แบบที่ 1: “ผลลัพธ์ = แนะนำ” เมื่อการให้คะแนนความพึงพอใจ = 5 และ “ผลลัพธ์ = ไม่แนะนำ” เมื่อการให้คะแนนความพึงพอใจ = 1, 2, 3 และ 4
- แบบที่ 2: “ผลลัพธ์ = แนะนำ” เมื่อการให้คะแนนความพึงพอใจ = 4 และ 5 และ “ผลลัพธ์ = ไม่แนะนำ” เมื่อการให้คะแนนความพึงพอใจ = 1, 2 และ 3

- ปรับค่าโหนดภายในเป็น 1, 2, 4, 8, 16, 32, 64, 128, 256 และ 512 ตามลำดับ
- เปลี่ยนชุดข้อมูลประเภทข้อความ (reviewText) ให้เป็นตัวเลขด้วยกระบวนการ Bag of Word (BoW) ดังที่กล่าวไปในบทที่ 3 (หัวข้อที่ 3.2.1.2)

ตารางที่ 4.2 ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของวิธีโครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูล Amazon

Hidden node(s)	MAE			
	LSTM 4-5	GRU 4-5	LSTM 5	GRU 5
1	0.4613	0.4554	0.2252	0.2208
2	0.4561	0.4536	0.2198	0.2176
4	0.4570	0.4515	0.2161	0.2159
8	0.4517	0.4491	0.2160	0.2156
16	0.4518	0.4500	0.2164	0.2146
32	0.4492	0.4476	0.2134	0.2142
64	0.4492	0.4487	0.2126	0.2123
128	0.4485	0.4483	0.2122	**0.2117**
256	0.4472	0.4479	0.2130	0.2120
512	0.4466	0.4477	0.2135	0.2119

จากตารางที่ 4.2 ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยที่ดีที่สุดได้แก่ 0.2117 จากวิธีแบบ GRU โดย “ผลลัพธ์ = แนะนำ” เมื่อการให้คะแนนความพึงพอใจ = 5 เมื่อกำหนดให้โหนดภายในเท่ากับ 128

ตารางที่ 4.3 ค่าความแม่นยำของของวิธีโครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูล Amazon

Hidden node(s)	Accuracy			
	LSTM 4-5	GRU 4-5	LSTM 5	GRU 5
1	55.41	55.27	79.28	78.66
2	55.64	55.31	78.61	78.46
4	55.31	55.83	78.83	78.61
8	55.27	55.62	78.97	78.66
16	55.86	55.64	**79.32**	79.23
32	55.51	55.27	78.57	**79.32**
64	55.86	55.55	78.64	78.66
128	55.86	55.46	78.66	78.46
256	55.51	55.44	79.08	78.66
512	55.33	55.27	**79.32**	78.61

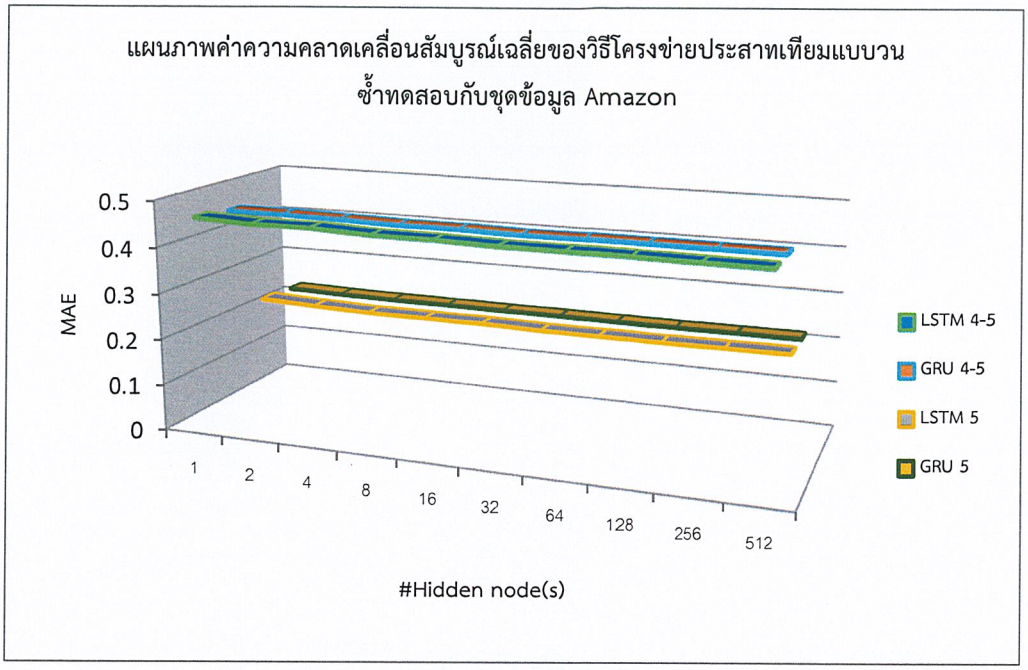
จากตารางที่ 4.3 ค่าความแม่นยำที่ดีที่สุดได้แก่ 79.32% จากวิธีแบบ GRU โดย “ผลลัพธ์ = แนะนำ” เมื่อการให้คะแนนความพึงพอใจ = 5 เมื่อกำหนดให้โหนดภายในเท่ากับ 32 และวิธีแบบ LSTM โดย “ผลลัพธ์ = แนะนำ” เมื่อการให้คะแนนความพึงพอใจ = 5 เมื่อกำหนดให้โหนดภายในเท่ากับ 16 และ 512

ตารางที่ 4.4 เวลาที่ใช้ของวิธีโครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูล Amazon

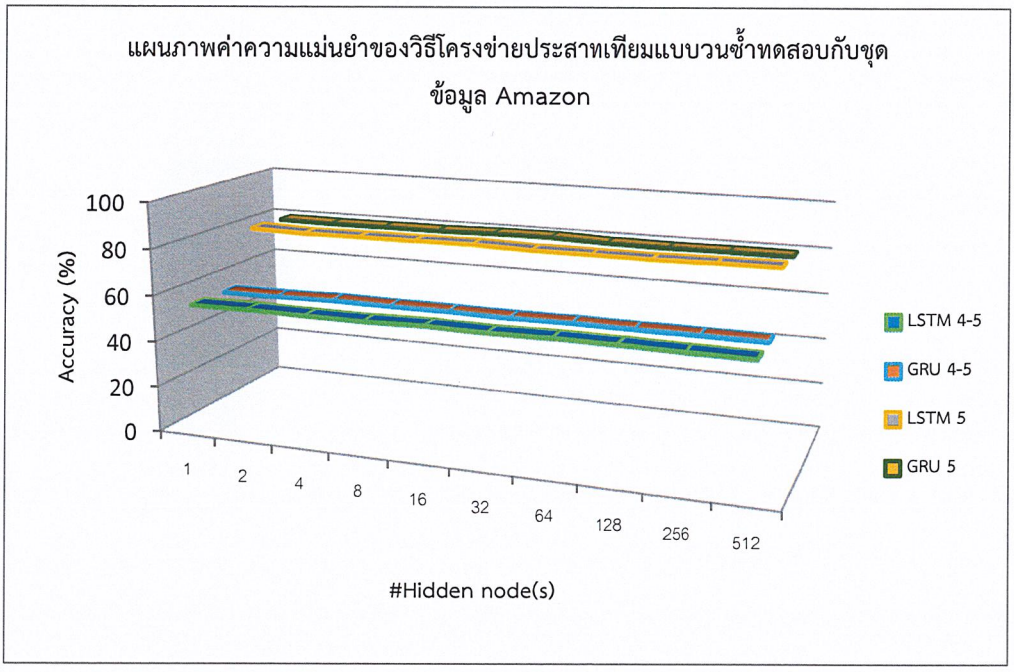
Hidden node(s)	Runtime			
	LSTM 4-5	GRU 4-5	LSTM 5	GRU 5
1	0:03:24	0:02:48	0:03:46	0:02:51
2	0:03:23	0:02:39	0:05:13	0:03:02
4	0:03:28	0:02:54	0:03:25	0:02:36
8	0:04:36	0:03:07	0:03:11	0:03:25
16	0:03:11	**0:02:14**	0:02:58	0:02:21
32	0:03:46	0:02:41	0:03:34	0:02:47
64	0:03:58	0:02:49	0:04:08	0:02:38
128	0:05:50	0:04:21	0:05:35	0:04:33
256	0:12:38	0:10:31	0:12:51	0:10:49
512	1:03:48	0:44:26	1:03:56	0:44:56

จากตารางที่ 4.4 เวลาในการสร้างและทดสอบโมเดลที่น้อยที่สุดได้แก่วิธีแบบ GRU โดย “ผลลัพธ์ = แนะนำ” เมื่อการให้คะแนนความพึงพอใจ = 4 และ 5 เมื่อกำหนดให้โหนดภายในเท่ากับ 16

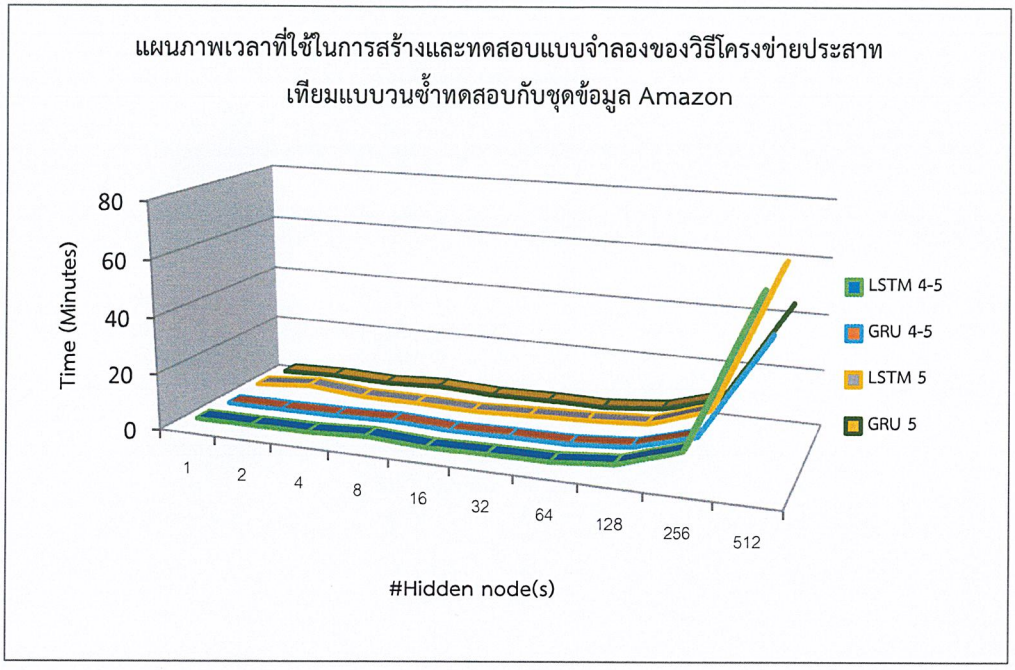
ในการทดลอง ข้อมูลทั้งหมดถูกแบ่งออกเป็นสองส่วน (Split validation) ส่วนหนึ่งเพื่อใช้สร้างแบบจำลอง (Model Training) จำนวน 80% และอีกส่วนเพื่อใช้ทดสอบแบบจำลอง (Model Testing) ที่ถูกสร้างขึ้น จำนวน 20% และใช้มาตรวัด 3 อย่างในการวัดผลประสิทธิภาพและประสิทธิผลของระบบได้แก่ ค่าความแม่นยำ (Accuracy) ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (MAE) และเวลาที่ใช้ในการสร้างและทดสอบโมเดล (Runtime)



ภาพที่ 4.3 แผนภาพค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยโดยใช้วิธีโครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูล Amazon



ภาพที่ 4.4 แผนภาพค่าความแม่นยำที่ได้จากการทดลองของวิธีโครงข่ายประสาทเทียมทดสอบกับชุดข้อมูล Amazon



ภาพที่ 4.5 แผนภาพเวลาที่ใช้ในการสร้างและทดสอบแบบจำลองของวิธีโครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูล Amazon

ผลการทดลองที่ได้โดยใช้มาตรวัดทั้งสามแบบแสดงได้ดังภาพที่ 4.3, 4.4 และ 4.5 โดยภาพที่ 4.3 แสดงแผนภาพค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของวิธีโครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูล MovieLens ภาพที่ 4.4 แสดงแผนภาพค่าความแม่นยำของวิธีโครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูล Amazon และภาพที่ 4.5 แสดงแผนภาพเวลาที่ใช้ในการสร้างและทดสอบแบบจำลองของวิธีโครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูล Amazon

จากภาพที่ 4.3 แสดงให้เห็นได้ว่าค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยที่ได้จากทั้งวิธีการ LSTM และ GRU มีแนวโน้มที่ลดลงหรือมีประสิทธิผลดีขึ้นเมื่อจำนวนโหนดภายในเพิ่มขึ้น โดยค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยที่ดีที่สุดที่ได้จากการทดลองคือ 0.21111 ซึ่งมาจากวิธีการแบบ GRU เมื่อกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 5 และจำนวนโหนดภายในเท่ากับ 128

สำหรับภาพที่ 4.4 แสดงให้เห็นได้ว่าค่าความแม่นยำที่ได้จากการแนะนำของ LSTM และ GRU มีความคล้ายกันมากจนแทบจะไม่แตกต่างกัน เช่นเดียวกันกับจำนวนของโหนดภายในแทบจะไม่ให้ผลลัพธ์ที่แตกต่างกันในการทดลองนี้ แต่ที่สังเกตได้ชัดเจนก็คือ ค่าความแม่นยำเมื่อกำหนดให้ "ผลลัพธ์ = แนะนำ" เมื่อคะแนนความพึงพอใจ = 5 มากกว่าค่าความแม่นยำเมื่อกำหนดให้ "ผลลัพธ์ = แนะนำ" เมื่อคะแนนความพึงพอใจ = 4 และ 5 อย่างชัดเจน โดยค่าความแม่นยำที่สูงที่สุดคือ 79.32%

และจากภาพที่ 4.5 แสดงให้เห็นได้ว่าเวลาที่ใช้ในการสร้างและทดสอบแบบจำลองมีแนวโน้มที่สูงขึ้นเรื่อย ๆ เมื่อจำนวนโหนดภายในเพิ่มขึ้น และเวลาที่ใช้สร้างและทดสอบแบบจำลองของวิธีการแบบ GRU น้อยกว่าเวลาที่ใช้สร้างและทดสอบแบบจำลองของ LSTM อย่างชัดเจน

4.1.3 ใช้วิธีการกรองด้วยความร่วมมือ (วิธีการแบบดั้งเดิม) ทดสอบกับชุดข้อมูลจาก MovieLens

ในการทดลองนี้จะกล่าวถึงการนำวิธีการกรองด้วยความร่วมมือทดสอบกับชุดข้อมูล MovieLens โดยใช้ Surprise Library (หัวข้อที่ 2.4)

ในการทดลองนี้ได้ใช้มาตรวัด 3 อย่างในการวัดผลประสิทธิภาพและประสิทธิผลของระบบได้แก่ ค่าความแม่นยำ ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย และเวลาที่ใช้ในการสร้างและทดสอบโมเดล โดยผลการทดลองแสดงในตารางที่ ผลลัพธ์ที่ได้แสดงในตารางที่ 4.5

ตารางที่ 4.5 ผลการทดลองโดยใช้วิธีการกรองด้วยความร่วมมือ (วิธีการแบบดั้งเดิม) ทดสอบกับชุดข้อมูลจาก MovieLens

MAE	Accuracy (%)	Runtime (h/m/s)
0.7323	53.28	0:02:53

จากตารางที่ 4.5 ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยคือ 0.7323 ค่าความแม่นยำ 53.28% และเวลาที่ใช้ในการสร้างและทดสอบโมเดล 2 นาที 53 วินาที

4.1.4 ใช้วิธีการกรองด้วยความร่วมมือ (วิธีการแบบดั้งเดิม) ทดสอบกับชุดข้อมูลจาก Amazon

ในการทดลองนี้จะนำข้อมูลชุดเดียวกับการทดลองในหัวข้อ 4.1.2 ทดสอบกับวิธีกรองด้วยความร่วมมือ โดยได้ออกแบบการทดลองดังนี้

- เนื่องจากการกรองด้วยความร่วมมือนั้นจัดเก็บข้อมูลนำเข้าอยู่ในรูปของเมตริกซ์ประกอบไปด้วย User, Item และ Rating (ดังที่กล่าวไปในหัวข้อที่ 2.1.2) ซึ่งในการทดลองนี้จะใช้ reviewerID, asin และ overall จากชุดข้อมูลของ Amazon โดย reviewerID แทน User, asin แทน Item และ overall แทน Rating

<u>ตัวอย่าง</u>

{

"reviewerID": "A2SUAM1J3GNN3B",

"asin": "000100039X ",

"overall": 5.0,

}

- ใช้ Surprise Library ในการดำเนินงาน (กล่าวในหัวข้อที่ 2.2.4)
- แบ่งข้อมูลนำเข้าด้วยวิธี 10-fold cross validation

ในการทดลองนี้ได้ใช้มาตรวัด 3 อย่างในการวัดผลประสิทธิภาพและประสิทธิผลของระบบได้แก่ ค่าความแม่นยำ ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย และเวลาที่ใช้ในการสร้างและทดสอบโมเดล โดยผลการทดลองแสดงในตารางที่ 4.6

ตารางที่ 4.6 ผลการทดลองโดยใช้วิธีการกรองด้วยความร่วมมือ (วิธีการแบบดั้งเดิม) ทดสอบกับชุดข้อมูลจาก Amazon

MAE	Accuracy (%)	Runtime (h/m/s)
0.7935	54.27	0.13.43

จากตารางที่ 4.6 ค่าเฉลี่ยของความคลาดเคลื่อนสัมบูรณ์เฉลี่ยอยู่ที่ 0.7935 ค่าความแม่นยำของการทำนายผล 54.27% และใช้เวลาในการสร้างและทดสอบโมเดล 13 นาที 43 วินาที

4.2 การอภิปรายผลการดำเนินงาน

จากการทดลองในหัวข้อที่ 4.1.1 ถึง 4.1.3 เราจะนำผลลัพธ์ที่ได้มาเปรียบเทียบและอภิปรายในรายละเอียด ซึ่งวิธีที่นำมาเปรียบเทียมประกอบไปด้วย

- Traditional Collaborative Filtering (TCF I) จากงานวิจัยในหัวข้อที่ 2.7.1
- Traditional Collaborative Filtering (TCF II) จากวิธีการกรองด้วยความร่วมมือโดยใช้วัดค่าความเหมือน (Similarity) ใน Matrix Factorization ด้วยวิธี Singular Value Decomposition (SVD)
- Latent Semantic Indexing (LSI) จากงานวิจัยในหัวข้อที่ 2.7.2
- Long Short-Term Memory (LSTM 4-5) เมื่อกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 4 และ 5
- Long Short-Term Memory (LSTM 5) เมื่อกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 5
- Gated Recurrent Units (GRU 4-5) เมื่อกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 4 และ 5
- Gated Recurrent Units (GRU 5) เมื่อกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 5

การเปรียบเทียบผลการทดลองจะถูกแบ่งออกเป็นสามส่วนได้แก่

1.) การเปรียบเทียบค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Error : MAE)
2.) การเปรียบเทียบค่าความแม่นยำ (Accuracy)
3.) การเปรียบเทียบเวลาที่ใช้ในการสร้าง และทดสอบโมเดล (Runtime)

โดยทั้งสามส่วนจะกล่าวถึงในหัวข้อที่ 4.2.1 ถึง 4.2.3

4.2.1 การเปรียบเทียบค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (MAE)

จากการทดลองในหัวข้อที่ 4.1.1 และ 4.1.3 (ชุดข้อมูล MovieLens) ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยที่ดีที่สุดคือ 0.2111 ซึ่งมาจากวิธีการแบบ GRU เมื่อกำหนดให้ "ผลลัพธ์ = แนะนำ" เมื่อคะแนนความพึงพอใจ = 5 และจำนวนโหนดภายในเท่ากับ 128 และ เมื่อนำมาเปรียบเทียบค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยที่ดีที่สุดของวิธีต่าง ๆ จะได้ผลลัพธ์ตามตารางที่ 4.7

ตารางที่ 4.7 การเปรียบเทียบค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของวิธีการต่าง ๆ โดยทดสอบกับชุดข้อมูล MovieLens

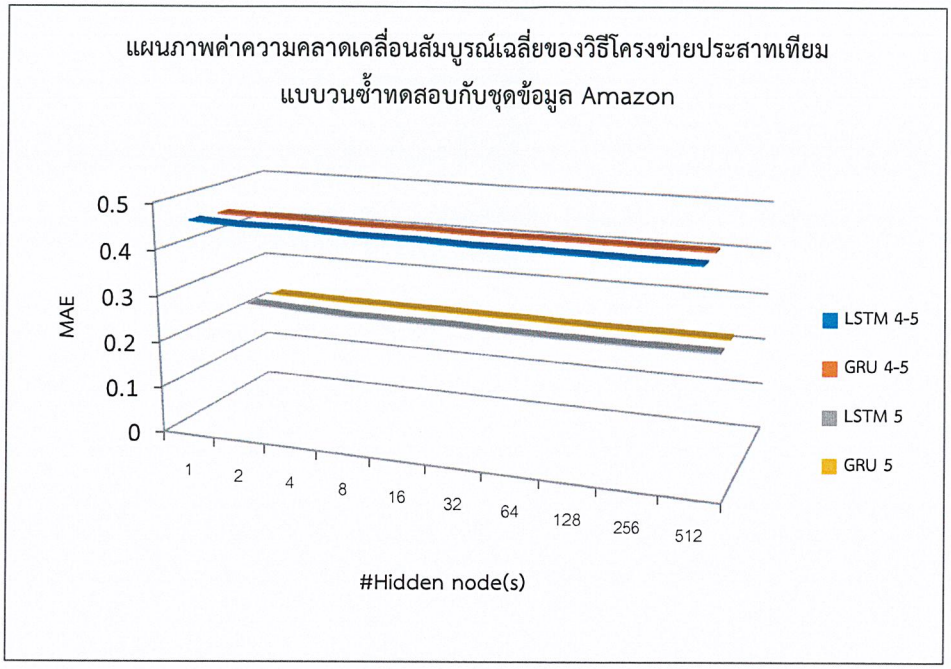
Mean Absolute Error : MAE						
TCF I	TCF II	LSI	LSTM 4-5	GRU 4-5	LSTM 5	GRU 5
0.6971	0.7323	0.8441	0.4466	0.4472	0.2122	**0.2111**

จากการทดลองในหัวข้อที่ 4.1.2 และ 4.1.4 (ชุดข้อมูล Amazon) ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยได้ถูกเปรียบเทียบดังแสดงในตารางที่ 4.5 โดยค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยที่ดีที่สุดคือ 0.2117 ซึ่งมาจากวิธีแบบ GRU เมื่อกำหนดให้ "ผลลัพธ์ = แนะนำ" เมื่อคะแนนความพึงพอใจ = 5 และจำนวนโหนดภายในเท่ากับ 128 และเมื่อนำมาเปรียบเทียบค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยที่ดีที่สุดของวิธีต่างๆ จะได้ผลลัพธ์ดังตารางที่ 4.8

ตารางที่ 4.8 การเปรียบเทียบค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของวิธีการต่าง ๆ โดยใช้ชุดข้อมูล Amazon

Hidden node(s)	MAE				
	TCF II	LSTM 4-5	GRU 4-5	LSTM 5	GRU 5
1	0.7935	0.4613	0.4554	0.2252	0.2208
2		0.4561	0.4536	0.2198	0.2176
4		0.4570	0.4515	0.2161	0.2159
8		0.4517	0.4491	0.2160	0.2156
16		0.4518	0.4500	0.2164	0.2146
32		0.4492	0.4476	0.2134	0.2142
64		0.4492	0.4487	0.2126	0.2123
128		0.4485	0.4483	0.2122	**0.2117**
256		0.4472	0.4479	0.2130	0.2120
512		0.4466	0.4477	0.2135	0.2119

จากตารางที่ 4.7 และ 4.8 พบว่าโครงข่ายประสาทเทียมแบบวนซ้ำทั้งวิธี LSTM และ GRU มีค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยที่ดีกว่าวิธี Traditional Collaborative Filtering (TCF I และ TCF II), Latent Semantic Indexing (LSI) และค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของโครงข่ายประสาทเทียมแบบวน เมื่อกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 5 ให้ประสิทธิผลที่ดีกว่า “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 4 และ 5



ภาพที่ 4.6 แผนภาพค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของวิธีโครงข่ายประสาทเทียมแบบ-วนซ้ำทดสอบกับชุดข้อมูล Amazon

ภาพที่ 4.6 แสดงแผนภาพแผนภาพค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของวิธีโครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูล Amazon

จากภาพที่ 4.6 เห็นได้ชัดว่าค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของโครงข่ายประสาทเทียมแบบวนซ้ำ เมื่อกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 5 ให้ประสิทธิผลที่ดีกว่า “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 4 และ 5

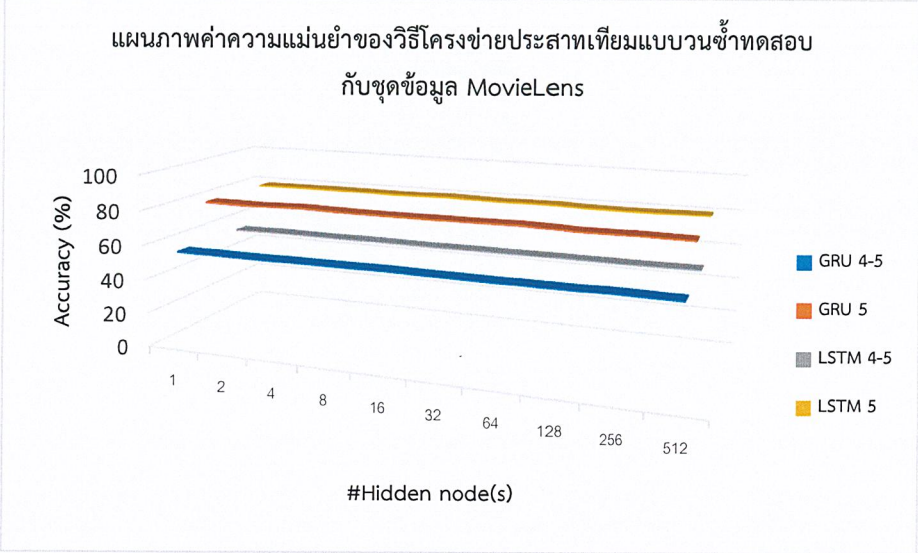
4.2.2 การเปรียบเทียบค่าความแม่นยำ (Accuracy)

เนื่องจากงานวิจัยที่เกี่ยวข้อง (หัวข้อที่ 2.7.1 และ 2.7.2) ไม่มีการบันทึกค่าความแม่นยำไว้ การเปรียบเทียบค่าความแม่นยำของวิธีต่าง ๆ ทดสอบกับชุดข้อมูล MovieLens จึงเปรียบเทียบได้เพียง Traditional Collaborative Filtering (TCF II) กับวิธีแบบ LSTM และวิธีแบบ GRU ซึ่งแสดงได้ดังตารางที่ 4.9

ตารางที่ 4.9 การเปรียบเทียบค่าความแม่นยำของวิธีการต่าง ๆ ทดสอบกับชุดข้อมูล MovieLens

Hidden node(s)	Accuracy (%)				
	TCF II	LSTM 4-5	GRU 4-5	LSTM 5	GRU 5
1	53.28	55.46	55.64	78.27	78.79
2		55.64	55.31	78.66	78.46
4		55.41	55.37	78.66	**79.32**
8		55.46	55.55	78.59	78.59
16		55.22	55.67	**79.32**	78.79
32		55.37	55.37	78.79	78.92
64		55.33	55.27	79.08	**79.32**
128		55.41	55.27	78.57	78.46
256		55.22	55.67	78.70	78.72
512		55.55	55.26	79.23	78.46

จากตารางที่ 4.9 สังเกตได้ว่าค่าความแม่นยำของการกรองด้วยความร่วมมือมีค่าใกล้เคียงกับวิธีโครงข่ายประสาทเทียมแบบวนซ้ำที่กำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 4 และ 5 และวิธีโครงข่ายประสาทเทียมแบบวนซ้ำเมื่อกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 5 มีค่าความแม่นยำสูงที่สุด



ภาพที่ 4.7 แผนภาพการเปรียบเทียบค่าความแม่นยำของวิธีโครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูล MovieLens

ภาพที่ 4.7 แสดงแผนภาพการเปรียบเทียบค่าความแม่นยำของวิธีโครงข่ายประสาทเทียมแบบวน-ซ้ำ ทดสอบกับชุดข้อมูล MovieLens

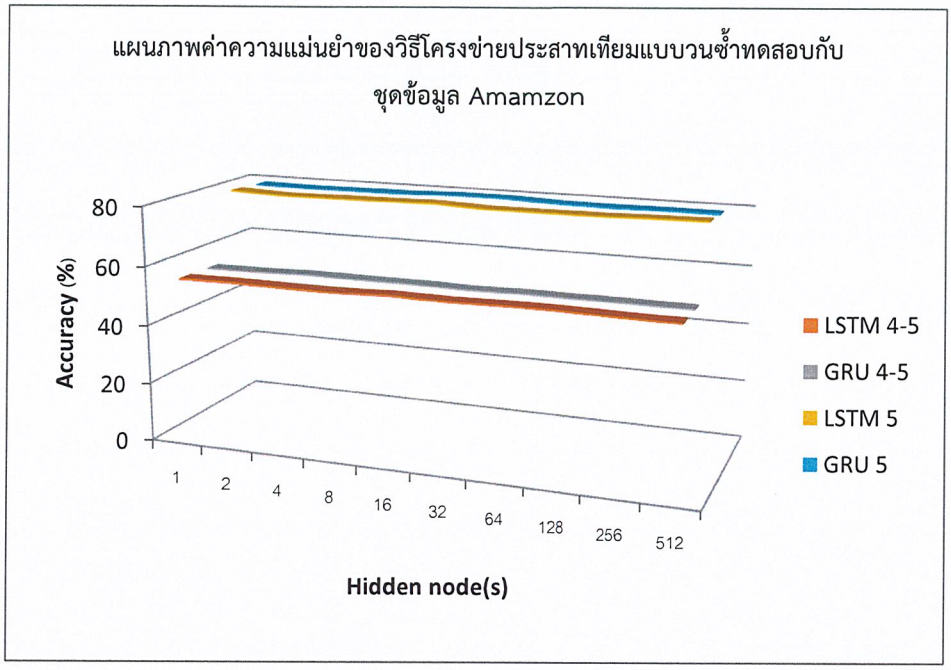
จากภาพที่ 4.7 ค่าความแม่นยำจากการใช้โครงข่ายประสาทเทียมแบบวนซ้ำทั้งแบบ LSTM และ GRU เมื่อกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 5 มีค่ามากกว่า “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 4 และ 5

จากการทดลองในหัวข้อที่ 4.1.2 และ 4.1.4 ชุดข้อมูล Amazon ค่าความแม่นยำที่ดีที่สุดคือ 79.32 ซึ่งมาจากวิธีแบบ LSTM เมื่อกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 5 และจำนวนโหนดภายในเท่ากับ 16 และ 512 และวิธีแบบ GRU เมื่อกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 5 และจำนวนโหนดภายในเท่ากับ 32 การเปรียบเทียบค่าความแม่นยำของวิธีการต่าง ๆ เมื่อทดสอบกับชุดข้อมูล Amazon แสดงได้ดังตารางที่ 4.10

ตารางที่ 4.10 การเปรียบเทียบค่าความแม่นยำของวิธีการต่าง ๆ ทดสอบกับชุดข้อมูล Amazon

Hidden node(s)	Accuracy (%)				
	TCF II	LSTM 4-5	GRU 4-5	LSTM 5	GRU 5
1	54.27	55.41	55.27	79.28	78.66
2		55.64	55.31	78.61	78.46
4		55.31	55.83	78.83	78.61
8		55.27	55.62	78.97	78.66
16		55.86	55.64	**79.32**	79.23
32		55.51	55.27	78.57	**79.32**
64		55.86	55.55	78.64	78.66
128		55.86	55.46	78.66	78.46
256		55.51	55.44	79.08	78.66
512		55.33	55.27	**79.32**	78.61

จากตารางที่ 4.10 จะสังเกตได้ว่าค่าความแม่นยำของวิธีโครงข่ายประสาทเทียมแบบวนซ้ำ เมื่อกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 4 และ 5 มีสูงกว่าวิธีการกรองด้วยความร่วมมือเล็กน้อย และวิธีโครงข่ายประสาทเทียมแบบวนซ้ำเมื่อกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 5 มีค่าสูงที่สุด



ภาพที่ 4.8 แผนภาพค่าความแม่นยำของวิธีโครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูล Amazon

ภาพที่ 4.8 แสดงแผนภาพค่าความแม่นยำของวิธีโครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูล Amazon

จากภาพที่ 4.8 ค่าความแม่นยำของวิธีโครงข่ายประสาทเทียมแบบวนซ้ำเมื่อกำหนดให้ “ผลลัพธ์ = แนะนำ”เมื่อคะแนนความพึงพอใจ = 5 สูงกว่าวิธีโครงข่ายประสาทเทียมแบบวนซ้ำเมื่อกำหนดให้ “ผลลัพธ์ = แนะนำ”เมื่อคะแนนความพึงพอใจ = 4 และ 5

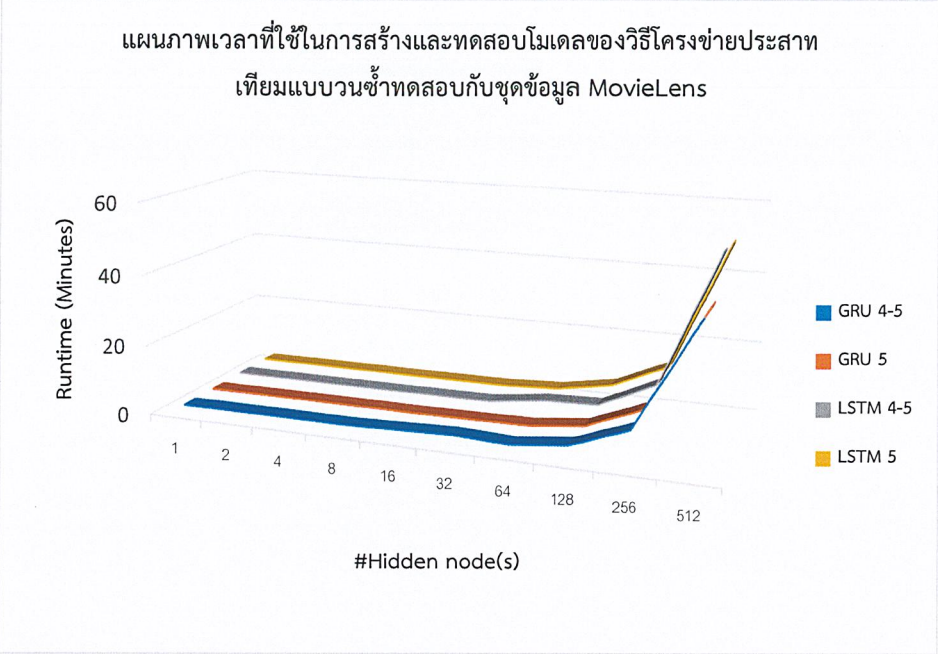
4.2.3 การเปรียบเทียบเวลาที่ใช้ในการสร้าง และทดสอบโมเดล (Runtime)

เนื่องจากงานวิจัยที่เกี่ยวข้อง (หัวข้อที่ 2.7.1 และ 2.7.2) ไม่มีการบันทึกเวลาที่ใช้การเปรียบเทียบเวลาที่ใช้ของชุดข้อมูล MovieLens จึงเปรียบเทียบระหว่างวิธี Traditional Collaborative Filtering (TCF II) กับวิธีแบบ LSTM และ GRU เวลาที่ใช้ของวิธีการต่าง ๆ เมื่อทดสอบกับชุดข้อมูล MovieLens แสดงได้ดังตารางที่ 4.11

ตารางที่ 4.11 การเปรียบเทียบเวลาที่ใช้ของวิธีการต่าง ๆ ทดสอบกับชุดข้อมูล MovieLens

Hidden node(s)	Runtime (H/M/S)				
	TCF II	LSTM 4-5	GRU 4-5	LSTM 5	GRU 5
1	0:02:53	0:02:45	0:02:19	0:02:40	0:02:17
2		0:02:36	0:02:18	0:02:41	0:02:15
4		0:02:40	0:02:20	0:02:40	0:02:16
8		0:02:40	0:02:16	0:02:39	0:02:17
16		0:02:45	0:02:57	0:02:39	**0:02:14**
32		0:02:49	0:03:10	0:02:44	0:02:24
64		0:04:51	0:02:40	0:03:21	0:02:41
128		0:05:03	0:04:18	0:06:00	0:04:09
256		0:12:47	0:10:18	0:13:01	0:10:50
512		0:50:34	0:39:05	0:49:43	0:39:41

จากตารางที่ 4.11 จะเห็นได้ว่าเวลาที่ใช้ในการสร้างและทดสอบโมเดลของแต่ละวิธีมีความใกล้เคียงกัน แต่โครงข่ายประสาทเทียมแบบวนซ้ำ เวลาที่ใช้มีแนวโน้มเพิ่มขึ้นตามค่าโหนดภายในและวิธีแบบ GRU ใช้เวลาน้อยกว่าวิธี LSTM



ภาพที่ 4.9 แผนภาพเวลาที่ใช้ของวิธีโครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูล MovieLens

ซึ่งภาพที่ 4.9 แสดงแผนภาพเวลาที่ใช้ของวิธีโครงข่ายประสาทเทียมแบบวนซ้ำทดสอบกับชุดข้อมูล MovieLens

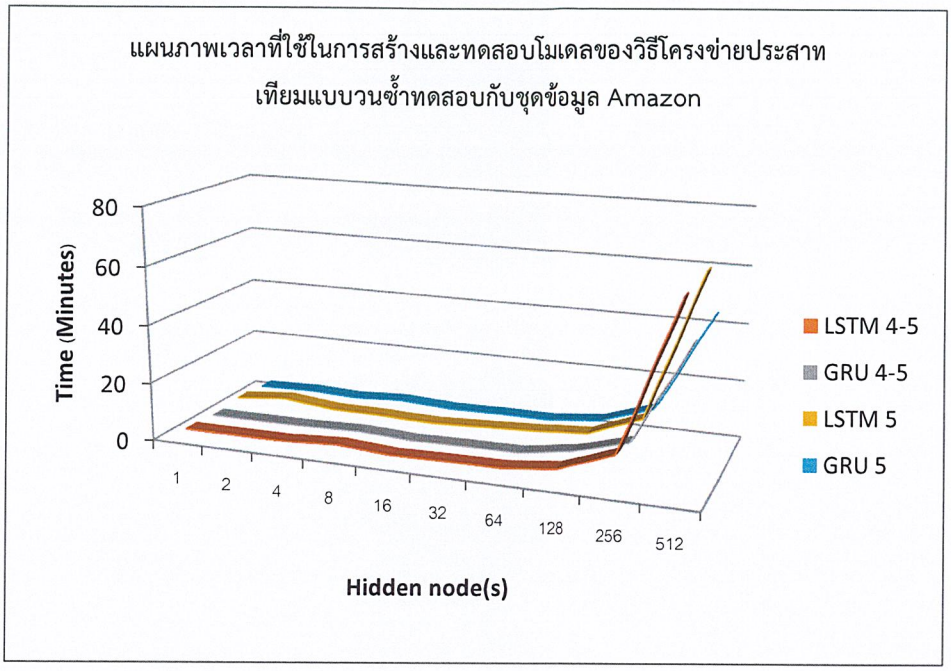
จากภาพที่ 4.9 เห็นได้ชัดว่าเวลาในการใช้สร้างและทดสอบโมเดลของโครงข่ายประสาทเทียมแบบวนซ้ำสูงขึ้นตามจำนวนโหนดภายในและเมื่อโหนดภายในมากขึ้นวิธีแบบ GRU จะใช้เวลาน้อยกว่าวิธีแบบ LSTM

จากการทดลองในหัวข้อที่ 4.1.2 และ 4.1.4 ชุดข้อมูล Amazon เวลาที่ใช้ในการสร้างและทดสอบโมเดลของวิธีการต่าง ๆ แสดงได้ดังตารางที่ 4.12

ตารางที่ 4.12 การเปรียบเทียบเวลาที่ใช้ของวิธีการต่าง ๆ ทดสอบกับชุดข้อมูล Amazon

Hidden node(s)	Runtime (H/M/S)				
	TCF II	LSTM 4-5	GRU 4-5	LSTM 5	GRU 5
1	0:13:43	0:03:24	0:02:48	0:03:46	0:02:51
2		0:03:23	0:02:39	0:05:13	0:03:02
4		0:03:28	0:02:54	0:03:25	0:02:36
8		0:04:36	0:03:07	0:03:11	0:03:25
16		0:03:11	**0:02:14**	0:02:58	0:02:21
32		0:03:46	0:02:41	0:03:34	0:02:47
64		0:03:58	0:02:49	0:04:08	0:02:38
128		0:05:50	0:04:21	0:05:35	0:04:33
256		0:12:38	0:10:31	0:12:51	0:10:49
512		1:03:48	0:44:26	1:03:56	0:44:56

จากตารางที่ 4.12 จะสังเกตได้ว่าเวลาที่ใช้ในการสร้างและทดสอบโมเดลของโครงข่ายประสาทเทียมแบบวนซ้ำน้อยกว่าวิธีการกรองด้วยความร่วมมือ แต่โครงข่ายประสาทเทียมแบบวนซ้ำมีแน้วโน้มที่จะใช้เวลามากขึ้นตามจำนวนโหนดภายใน



ภาพที่ 4.10 แผนภาพเวลาที่ใช้ของวิธีการต่าง ๆ ทดสอบกับชุดข้อมูล Amazon

จากภาพที่ 4.10 เวลาที่ใช้ในการสร้างและทดสอบโมเดลของโครงข่ายประสาทเทียมแบบวนซ้ำมีแนวโน้มสูงขึ้นตามค่าโหนดภายใน โดยเมื่อโหนดภายในมากขึ้นวิธีแบบ GRU จะใช้เวลาน้อยกว่าวิธีแบบ LSTM

4.3 การทดสอบทางสถิติ

ในหัวข้อนี้จะกล่าวถึงการนำผลลัพธ์ที่ได้จากการทดลองในหัวข้อที่ 4.1 มาทำการทดสอบทางสถิติเพื่อวิเคราะห์ว่าผลลัพธ์ที่ได้มีความแตกต่างกันอย่างมีนัยสำคัญหรือไม่ โดยการวิเคราะห์ทางสถิติในปัญหาพิเศษนี้จะใช้การวิเคราะห์ความแปรปรวน (ANOVA) ซึ่งจะอธิบายในหัวข้อที่ 4.3.1

4.3.1 การวิเคราะความแปรปรวน (Analysis of Variance : ANOVA)

ในการทดสอบสมมติฐานของปัญหาพิเศษนี้จะใช้วิธี Analysis of Variance หรือ ANOVA ซึ่งเป็นการวิเคราะห์ข้อมูลที่ใช้ในการทดสอบสมมติฐานเกี่ยวกับความแตกต่างของค่าเฉลี่ยประชากรมากกว่าสองกลุ่ม โดยทำการทดสอบเพียงครั้งเดียว โดยสมมติฐานของการทดสอบมีดังนี้

$H_0 : \mu_1 = \mu_2 = \ldots = \mu_k$

$H_1 : \mu_1 \neq \mu_2 \neq \ldots \neq \mu_k$ หรือมี μ_i อย่างน้อย 1 คู่ที่ต่างกัน

หรือ

H_0 : ค่าเฉลี่ยของประชากร k กลุ่มไม่แตกต่างกัน

H_1 : ค่าเฉลี่ยของประชากร k กลุ่มแตกต่างกันอย่างน้อย 1 คู่

ถ้าผลการทดสอบสมมติฐานปฏิเสธ H_0 หมายความว่ามีค่าเฉลี่ยของประชาการอย่างน้อย 1 คู่ที่มีค่าต่างกัน ซึ่งหลักเกณฑ์สำคัญในการวิเคราะห์ความแปรปรวนคือ การแบ่งความแปรปรวนของข้อมูลทั้งหมดออกตามสาเหตุที่ทำให้ข้อมูลแตกต่างกันได้แก่ ความแปรปรวนภายในกลุ่ม (within group) และความแปรปรวนระหว่างกลุ่ม (between group)

ความแปรปรวนระหว่างกลุ่ม (Between Group Sum F Square) เขียนแทนด้วยสัญลักษณ์ SSB เป็นการพิจารณาความแปรปรวนที่เกิดจากการที่ค่าเฉลี่ยของตัวอย่างในแต่ละกลุ่มแตกต่างจากค่าเฉลี่ยรวม โดยที่

$$SSB = \sum_{i=1}^{k} n_i(\bar{X}_i - \bar{X})^2$$

ความแปรปรวนภายในกลุ่ม (Within Group Sum f Square) เขียนแทนด้วยสัญลักษณ์ SSE เป็นการพิจารณาความแปรปรวนที่เกิดขึ้นภายในกลุ่มแต่ละกลุ่มซึ่งไม่ทราบสาเหตุว่าเป็นความแปรปรวนที่เกิดจากสาเหตุใด ในบางครั้งจึงเรียกว่าความคลาดเคลื่อน (Error Sum of Square) โดยที่

$$SSE = \sum_{i=1}^{k}\sum_{j=1}^{n_i}(X_{ij} - \bar{X})^2$$

ความแปรปรวนรวม (Total Sum of Square) เขียนแทนด้วยสัญลักษณ์ SST เป็นการพิจารณาความแปรปรวนที่เกิดจากค่าสังเกตแต่ละค่าแตกต่างจากค่าเฉลี่ยรวม โดยที่

$$SST = \sum_{i=1}^{k}\sum_{j=1}^{n_i}(X_{ij} - \bar{X})^2 \text{ และ } SST = SSB + SSE$$

การวิเคราะห์ความแปรปรวนสามารถคำนวนได้จากตารางการวิเคราะห์ความแปรปรวน (Analysis of Variance Table)

ตารางที่ 4.13 การวิเคราะห์ความแปรปรวน

Source of variation	Degree of freedom (df)	Sum of Square (SS)	Mean of Square ($MS = \frac{ss}{df}$)	F
ระหว่างกลุ่ม	k-1	SSB	$MSB = \frac{SSB}{k-1}$	$F = \frac{MSB}{MSE}$
ภายในกลุ่ม	n-k	SSE	$MSE = \frac{SSE}{n-k}$	
รวม	n-1	SST		

จากตารางที่ 4.13 ตัวสถิติในการทดสอบคือ $F = \frac{MSB}{MSE}$ และสมมติฐาน H_0 จะถูกปฏิเสธก็ต่อเมื่อค่าสถิติทดสอบ F มีค่ามากกว่าค่าวิกฤต $f_{1-\alpha}, k-1, n-k$

ในปัญหาพิเศษนี้จะใช้โปรแกรม Microsoft Excel ในการทำวิเคราะห์ความแปรปรวนของแต่ละวิธีมาทดสอบประกอบไปด้วย

- Traditional Collaborative Filtering (TCF II) จากการกรองด้วยความร่วมมือโดยวัดค่าความเหมือน (Similarity) ใน Matrix Factorization ด้วย Singular Value Decomposition (SVD)
- Long Short-Term Memory (LSTM 4-5) เมื่อกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 4 และ 5
- Long Short-Term Memory (LSTM 5) เมื่อกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 5
- Gated Recurrent Units (GRU 4-5) เมื่อกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 4 และ 5
- Gated Recurrent Units (GRU 5) เมื่อกำหนดให้ “ผลลัพธ์ = แนะนำ” เมื่อคะแนนความพึงพอใจ = 5

ในการทดสอบความแปรปรวนประกอบไปด้วยสามการทดสอบได้แก่การวิเคราะห์ความแปรปรวนของค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย การวิเคราะห์ความแปรปรวนของค่าความแม่นยำและการวิเคราะห์ความแปรปรวนของเวลาที่ใช้ในการสร้างและทดสอบโมเดลตามหัวข้อที่ 4.3.1.1 ถึง 4.3.1.3

4.3.1.1 การวิเคราะห์ความแปรปรวนของค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย

หัวข้อนี้จะกล่าวถึงการใช้วิเคราะห์ค่าความแปรปรวนกับค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยที่ดีที่สุดของแต่ละวิธีเมื่อทดสอบกับชุดข้อมูล MovieLens และ Amazon โดยค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยที่ดีที่สุดของแต่ละวิธีเพื่อทดสอบกับชุดข้อมูล MovieLens และ Amazon แสดงดังตารางที่ 4.14

ตารางที่ 4.14 ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยที่ดีที่สุดของแต่ละวิธีเมื่อทดสอบกับชุดข้อมูล MovieLens และ Amazon

Dataset	MAE				
	TCF II	LSTM 4-5	LSTM 5	GRU 4-5	GRU 5
MovieLens	0.6971	0.4466	0.2122	0.4472	0.2111
Amazon	0.7935	0.4466	0.2122	0.4476	0.2117

จากตารางที่ 4.14 กำหนดให้แถวคือชุดข้อมูลและคอลัมน์คือวิธีต่าง ๆ ที่ใช้ทดสอบ โดยแถวแรกหมายถึงค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยที่ดีที่สุดเมื่อทดสอบกับชุดข้อมูล MovieLens และแถวที่สองหมายถึงค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยที่ดีที่สุดเมื่อทดสอบกับชุดข้อมูล Amazon โดยกำหนดสมมติฐานของการทดสอบทางสถิติดังนี้

H_0 : ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของแต่ละวิธีการไม่มีความแตกต่างกันอย่างมีนัยสำคัญ

H_1 : ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของแต่ละวิธีการแตกต่างกันอย่างมีนัยสำคัญ

หลังจากที่ใช้เครื่องมือการวิเคราะห์ความแปรปรวนในโปรแกรม Microsoft Excel แล้ว ผลลัพธ์คือตารางการวิเคราะห์ความแปรปรวน (Analysis pf Variance Table) ดังแสดงได้ในตารางที่ 4.15

ตารางที่ 4.15 ผลลัพธ์จากการวิเคราะห์ความแปรปรวนของค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยด้วยโปรแกรม Microsoft Excel

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	0.387396	4	0.096849	104.2117	5.31E-05	5.192168
Within Groups	0.004647	5	0.000929			
Total	0.392043	9				

จากตารางที่ 4.15 ค่าวิกฤต $F\ crit$ ที่ได้คือ 5.192168 และค่า F ที่ได้คือ 104.2117 เนื่องจากค่าวิกฤต $f_{1-\alpha}, k-1, n-k$ จะปฏิเสธ H_0 เมื่อค่าสถิติทดสอบ F มากกว่าค่าวิกฤต ดังนั้นการทดสอบนี้จะปฏิเสธสมมติฐาน H_0 ซึ่งหมายความว่าค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยของแต่ละวิธีมีความแตกต่างกันอย่างมีนัยสำคัญ

4.3.1.2 การวิเคราะห์ความแปรปรวนของค่าความแม่นยำ

หัวข้อนี้จะกล่าวถึงการวิเคราะห์ความแปรปรวนกับค่าความแม่นยำที่ดีที่สุดของแต่ละวิธีเมื่อทดสอบกับชุดข้อมูล MovieLens และ Amazon โดยแสดงถึงตารางที่ 4.16 ค่าความแม่นยำที่ดีที่สุดของแต่ละวิธีเพื่อทดสอบกับชุดข้อมูล MovieLens และ Amazon Amazon โดยกำหนดสมมติฐานของการทดสอบทางสถิติดังนี้

H_0 : ค่าความแม่นยำของแต่ละวิธีการไม่มีความแตกต่างกันอย่างมีนัยสำคัญ

H_1 : ค่าความแม่นยำของแต่ละวิธีการแตกต่างกันอย่างมีนัยสำคัญ

ตารางที่ 4.16 ค่าความแม่นยำที่ดีที่สุดของแต่ละวิธีเมื่อทดสอบกับชุดข้อมูล MovieLens และ Amazon

Dataset	Accuracy				
	TCF II	LSTM 4-5	LSTM 5	GRU 4-5	GRU 5
MovieLens	53.28	55.64	79.32	55.67	79.32
Amazon	54.27	55.86	79.32	55.83	79.32

จากตารางที่ 4.16 กำหนดให้แถวแรกหมายถึงค่าความแม่นยำที่ดีที่สุดของแต่ละวิธีเมื่อทดสอบกับชุดข้อมูล MovieLens และแถวที่สองหมายถึงค่าความแม่นยำที่ดีที่สุดของแต่ละวิธีเมื่อทดสอบกับชุดข้อมูล โดยตารางการวิเคราะห์ความแปรปรวนที่ได้จากการทดสอบแสดงได้ดังตารางที่ 4.17

ตารางที่ 4.17 ผลลัพธ์จากการวิเคราะห์ความแปรปรวนของค่าความแม่นยำด้วยโปรแกรม Microsoft Excel

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	1414.03	4	353.5075	3353.643	9.376E-09	5.192168
Within Groups	0.52705	5	0.10541			
Total	1414.557	9				

จากตารางที่ 4.17 เนื่องจากค่า F ที่ได้คือ 3,353.643 ซึ่งมากกว่าค่าวิกฤต $F\ crit$ ที่มีค่า 5.192168 ดังนั้นสมมติฐาน H_0 จึงถูกปฏิเสธ หมายความว่าค่าความแม่นยำของแต่ละวิธีมีความแตกต่างกันอย่างมีนัยสำคัญ .

4.3.1.3 การวิเคราะห์ความแปรปรวนของเวลาที่ใช้ในการสร้างและทดสอบโมเดล

หัวข้อนี้จะกล่าวถึงการวิเคราะห์ความแปรปรวนกับเวลาที่ดีที่สุดที่ใช้ในการสร้างและทดสอบโมเดลของแต่ละวิธีเมื่อทดสอบกับชุดข้อมูล MovieLens และ Amazon โดยตารางที่ 4.18 แสดงถึงเวลาที่ดีที่สุดที่ใช้ในการสร้างและทดสอบโมเดลของวิธีการต่าง ๆ ทดสอบกับชุดข้อมูล MovieLens และ Amazon โดยกำหนดสมมติฐานของการทดสอบทางสถิติดังนี้

H_0 : เวลาที่ดีที่สุดที่ใช้ในการสร้างและทดสอบโมเดลของแต่ละวิธีการไม่มีความแตกต่างกันอย่างมีนัยสำคัญ

H_1 : เวลาที่ดีที่สุดที่ใช้ในการสร้างและทดสอบโมเดลของแต่ละวิธีการแตกต่างกันอย่างมีนัยสำคัญ

ตารางที่ 4.18 เวลาที่ใช้ที่ดีที่สุดที่ใช้ในการสร้างและทดสอบโมเดลของแต่ละวิธีเมื่อทดสอบกับชุดข้อมูล MovieLens และ Amazon

Dataset	Runtime (Minutes)				
	TCF II	LSTM 4-5	LSTM 5	GRU 4-5	GRU 5
MovieLens	2:53	2:36	2:39	2:16	2:14
Amazon	13:43	3:23	2:58	2:14	2:21

จากตารางที่ 4.18 กำหนดให้แถวแรกหมายถึงเวลาที่ดีที่สุดที่ใช้ในการสร้างและทดสอบโมเดลของแต่ละวิธีเมื่อทดสอบกับชุดข้อมูล MovieLens และแถวที่สองหมายถึงเวลาที่ดีที่สุดที่ใช้ในการสร้างและทดสอบโมเดลของแต่ละวิธีเมื่อทดสอบกับชุดข้อมูล Amazon โดยตารางการวิเคราะห์ความแปรปรวนที่ได้จากการทดสอบแสดงได้ดังตารางที่ 4.19

ตารางที่ 4.19 ผลลัพธ์จากการวิเคราะห์ความแปรปรวนของเวลาที่ใช้ในการสร้างและทดสอบโมเดลด้วยโปรแกรม Microsoft Excel

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	50.34866	4	12.58717	1.052366	0.465373	5.192168
Within Groups	59.80415	5	11.96083			
Total	110.1528	9				

จากตารางที่ 4.19 จะเห็นได้ว่าค่า F ที่ได้คือ 1.052366 ซึ่งน้อยกว่าค่าวิกฤต $F\ crit$ ที่มีค่า 5.192168 ดังนั้นสมมติฐาน H_0 จึงถูกยอมรับ หมายความว่าเวลาที่ใช้ในการสร้างและทดสอบโมเดลของแต่ละวิธีไม่มีความแตกต่างกันอย่างมีนัยสำคัญ

4.4 ปัญหาที่พบในการดำเนินงาน

4.4.1 ข้อจำกัดทางทรัพยากร

เนื่องจากชุดข้อมูล Amazon ที่นำมาใช้ประกอบไปด้วยข้อมูลของยอดรีวิวจำนวน 8,898,041 รีวิว เป็นข้อมูลประเภทข้อความ และตัวอักษร ซึ่งส่งผลต่อเวลา และหน่วยความจำในการประมวลผล เนื่องด้วยทรัพยากรทีมีจำกัด จึงใช้วิธีการลดจำนวนข้อมูลที่นำมาทดสอบ โดยทำการสุ่มข้อมูลของผู้ใช้ขนาด 500,000 รีวิว มาจากข้อมูลทั้งหมดก่อนที่จะนำชุดข้อมูลมาทดลองในขั้นตอนต่อไป

4.4.2 ชุดข้อมูลที่นำมาทดลองคนละประเภท

ชุดข้อมูลที่จะนำมาใช้กับโครงข่ายประสาทเทียมแบบวนซ้ำในการทดลองต้องเป็นข้อมูลประเภทเดียวกัน เนื่องด้วยชุดข้อมูลที่นำมาทดสอบมีข้อมูลทั้งประเภทข้อความ และตัวอักษร จึงต้องแปลงชนิดข้อมูลก่อนที่จะนำชุดข้อมูลมาดำเนินงานในขั้นตอนต่อไป

4.4.3 การประมวลผลข้อมูลประเภทข้อความ

การประมวลผลข้อมูลประเภทข้อความ นั้นเราไม่สามารถใช้ข้อมูลที่เป็นตัวอักษรกับโครงข่ายประสาทเทียมแบบวนซ้ำได้โดยตรง เนื่องจากการทำงานของโครงข่ายประสาทเทียมทั่วไปต้องใช้ข้อมูลนำเข้าประเภทตัวเลขเท่านั้น จึงต้องทำการแปลงข้อมูลนำเข้าที่เป็นตัวอักษรให้เป็นตัวเลขเสียก่อน โดยจะใช้วิธีที่เรียกว่า Bag of Word (BoW) ซึ่งเป็นการสร้างเวกเตอร์ข้อมูลที่เป็นตัวเลขจากกลุ่มคำที่ปรากฏอยู่ในชุดข้อมูล ก่อนที่จะนำชุดข้อมูลไปประมวลผลต่อไป

บทที่ 5

สรุปผลการดำเนินงานและข้อเสนอแนะ

ในบทนี้จะกล่าวถึงการสรุปผลการดำเนินงานและข้อเสนอแนะ ซึ่งการสรุปผลการดำเนินงานจะถูกนำเสนอในหัวข้อ 5.1 และข้อเสนอแนะจะถูกนำเสนอในหัวข้อ 5.2

5.1 สรุปผลการดำเนินงาน

ปัญหาพิเศษนี้นำเสนอวิธีการเรียนรู้เชิงลึกสำหรับระบบแนะนำ โดยใช้เทคนิคโครงข่ายประสาทเทียมแบบวนซ้ำแบบ Long Short-Term Memory (LSTM) และ Gated Recurrent Unit (GRU) ซึ่งช่วยแก้ปัญหาการจัดการข้อมูลที่เบาบางและกระจาย (Sparsity) ที่เกิดขึ้นจากการใช้วิธีการแบบดั้งเดิมสำหรับระบบแนะนำได้ จากการทดลองโดยใช้ชุดข้อมูลมาตรฐานและการทดสอบทางสถิติ พบว่าระบบแนะนำโดยใช้วิธีการเรียนรู้เชิงลึกดีกว่าระบบแนะนำที่ใช้วิธีการแบบดั้งเดิมในแง่ของประสิทธิผล แต่การทดสอบทางสถิติบ่งชี้ว่าประสิทธิภาพของวิธีการเรียนรู้เชิงลึกไม่แตกต่างกับวิธีการแบบดั้งเดิมอย่างมีนัยสำคัญ ทั้งนี้อาจเป็นเพราะว่าขนาดชุดข้อมูลทดสอบไม่มีขนาดใหญ่เพียงพอ สังเกตได้จากชุดข้อมูล MovieLens เวลาที่ใช้สำหรับทั้งสองวิธีไม่แตกต่างกันมาก แต่ชุดข้อมูลที่มีขนาดใหญ่อย่าง Amazon ใช้เวลาที่แตกต่างกันมาก ซึ่งการทดสอบทางสถิติในปัญหาพิเศษนี้มีการทดสอบแค่สองชุดข้อมูลจึงได้ผลลัพธ์ดังกล่าว แต่ก็สังเกตได้ว่าเวลาที่ใช้ในการประมวลผลของวิธีการเรียนรู้เชิงลึกเร็วกว่าวิธีการแบบดั้งเดิมมากเมื่อใช้ชุดข้อมูลที่มีขนาดใหญ่ จึงอาจกล่าวได้ว่าวิธีการเรียนรู้เชิงลึกให้ประสิทธิภาพที่ดีกว่าวิธีการแบบดั้งเดิมเมื่อชุดข้อมูลมีแนวโน้มที่ใหญ่มากขึ้น สำหรับประสิทธิผลของการทดลองของระบบแนะนำโดยใช้วิธีการเรียนรู้เชิงลึกประสิทธิผลของระบบแนะนำได้แปรผันตามจำนวนของโหนดภายในแต่ประสิทธิภาพของระบบแนะนำแปรผกผันกับจำนวนโหนดภายใน โดยจำนวนโหนดภายในมากที่สุดที่ใช้ในการทดลองคือ 512 แต่เนื่องด้วยข้อจำกัดทางด้านทรัพยากรจึงไม่สามารถทดลองใช้จำนวนโหนดภายในได้มากขึ้น ข้อสังเกตที่น่าสนใจอีกประเด็นที่ได้จากการทดลอง คือ ประสิทธิผลของระบบแนะนำเมื่อใช้เทคนิค LSTM และ GRU มีความคล้ายกันมาก แต่เวลาที่ใช้ในการประมวลผลเมื่อใช้เทคนิค GRU เร็วกว่าเทคนิค LSTM อย่างเห็นได้ชัด และเมื่อมีการเปรียบเทียบประสิทธิผลของระบบแนะนำโดยใช้วิธีการเรียนรู้เชิงลึกที่ได้นำเสนอกับงานวิจัยแบบดั้งเดิมอื่น ๆ ที่ได้ใช้ชุดข้อมูลทดสอบเดียวกัน ผลลัพธ์บ่งชี้ว่าวิธีการที่ได้นำเสนอให้ประสิทธิผลที่ดีกว่าวิธีการแบบดั้งเดิม

5.2 ข้อเสนอแนะ

จากการดำเนินงานของปัญหาพิเศษนี้ทำให้ทราบทิศทางงานวิจัยที่สามารถต่อยอดได้ในอนาคต ขณะนี้ทางทีมวิจัยกำลังทดลองวิธีการที่ได้นำเสนอกับชุดข้อมูลที่มีจำนวนมากและมีคุณลักษณะ (Attributes) ที่มีความหลากหลายมากยิ่งขึ้น รวมทั้งกำลังพิจารณาถึงการใช้วิธีการเรียนรู้เชิงลึกแบบอื่น ๆ เช่น Convolutional Neural Network (CNN) หรือการผสมผสานกันระหว่างเทคนิควิธีการเรียนรู้เชิงลึกแบบต่าง ๆ เป็นต้น

เอกสารอ้างอิง

เกียรติชัย ปรีชาอนุสรณ์ และณัฐวัฒน์ โนนสังข์. 2551. "วิธีการกรองด้วยความร่วมมือแบบแอลเอสไอสำหรับระบบแนะนำ."ปริญญานิพนธ์ วิทยาศาสตร์บัณฑิต สาขาวิทยาการคอมพิวเตอร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

ชลัท ศรีวิเศษสม, ธีรเกียรติ พ่วงตามพงษ์ และวิธวินท์ เทพศุภรังษิกุล. 2551. "การศึกษาทดลองตัววัดความเหมือนสำหรับการกรองด้วยความร่วมมือ." วิทยาศาสตร์บัณฑิต สาขาวิทยาการคอมพิวเตอร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

Gemmis, M.D. Lops, P. and Semeraro, G. "Content-based Recommender Systems: State of The Arts and Trends", *Recommender Systems Handbook*, Pages 73-105, Springer, Boston, 2011.

Schafer, J. Frankowski, D. Herlocker, J. and Sen, S. "Collaborative Filtering Recommender Systems", *Lecture Notes in Computer Science*, Volume 4321, Pages 291-324, Springer, Berlin, 2007.

Papadimitriou, C. H. Raghavan, P. Tamaki, H. and Vempala, S. "Latent Semantic Indexing: A Probabilistic Analysis", *Journal of Computer System Sciences*, Volume 61, Issue 2, Pages 217-235, October 2000.

Burget, L. Cernocky, J. Karafiat, M. Khudanpur, S. and Mikolov, T. "Recurrent Neural Network Based Language Model", *11th Annual Conference of the International Speech Communication Association*, Chiba, Japan, September 2010.

Hochreiter, S. and Schmidhuber, J. "Long Short-Term Memory", *Neural Computation*, Volume 9, Issue 8, November 1997.

Bengio, Y. Cho, K. Chung, J. Gulcehre, C. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modelling", *arXiv preprint*: 1412.3555, 2014.

Matsuura, K. and Willmott, C. "Advantages of the Mean Absolute Error (MAE) over the Root Mean Square Error (RMSE) in Assessing Average Model Performance", *Climate Research Journal*, Volume 30, No. 1, Pages 79-82, December 2005.

Liu, D. Z. and Singh, G. "A Recurrent Neural Network Based Recommender System", *Internal Report*, Department of Computer Science, Stanford University, 2016.

Baeza-Yates, R. and Ribeiro-Neto, B. "*Modern Information Retrieval*", Addison Wesley, ACM Press, New York, 1999.

Suykens, J. and Vandewalle, J. "Least Squares Support Vector Machine Classifiers", *Neural Processing Letters*, Volume 9, Issue 3, Pages 293-300, Springer, June 1999.

"Yelp Dataset Challenge." [Online]. https://www.yelp.com/dataset/challenge/. (5 April 2018).

"Recurrent Neural Networks." [Online].http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/. (5 April 2018).

" Understanding LSTM Networks. " [Online]. http://colah.github.io/posts/2015-08-Understanding-LSTMs/. (5 April 2018).

" MovieLens Dataset. " [Online]. https://grouplens.org/datasets/movielens/. (5 April 2018).

งานทะเบียนคณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

คำรับรองเล่มโครงงานพิเศษ/ปัญหาพิเศษ/สหกิจศึกษา

วันที่ 13 เดือน มิถุนายน พ.ศ 2561

ข้าพเจ้า นาย/นาง/นางสาว..... รหัสประจำตัว.....

นาย/นาง/นางสาว สันติ หริรักษ์ รหัสประจำตัว 57050837

นาย/นาง/นางสาว..... รหัสประจำตัว.....

นักศึกษาหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชา วิทยาการคอมพิวเตอร์ ภาควิชา วิทยาการคอมพิวเตอร์

ขอรับรองว่าโครงงานพิเศษ/ปัญหาพิเศษ/สหกิจศึกษา เรื่อง

ชื่อภาษาไทย ดิอัลกอริทึมการเรียนรู้เชิงลึกสำหรับระบบแนะนำ

.....

ชื่อภาษาอังกฤษ Deep Learning Method for Recommender System

.....

ปีการศึกษา 2560

เป็นผลงานวิจัยที่มิได้คัดลอกหรือละเมิดลิขสิทธิ์ของผู้อื่นและได้ผ่านการตรวจสอบความซ้ำซ้อนเรียบร้อยแล้ว และได้แนบเอกสารการตรวจสอบการลอกเลียนงานวรรณกรรมที่ตรวจสอบจากเล่มโครงงานพิเศษ/ปัญหาพิเศษ/สหกิจศึกษาฉบับสมบูรณ์แล้ว

โปรแกรมอักขราวิสุทธิ์ 0 % หรือโปรแกรม Turnitin.....%

ลงชื่อ.....	ลงชื่อ.....	ลงชื่อ.....
(นาย ศุภวิช วงศ์อดุลวิทย์)	(นายสันติ หริรักษ์)	(นายอัษฎาวุธ หลวงสมบัติ)
นักศึกษา	นักศึกษา	นักศึกษา

ข้าพเจ้า ศ. / รศ. / ผศ. / ดร. / อ..... อาจารย์ที่ปรึกษาโครงงานพิเศษ/ปัญหาพิเศษ/สหกิจศึกษา ได้ตรวจสอบโครงงานพิเศษ/ปัญหาพิเศษ/สหกิจศึกษาของนักศึกษาข้างต้น แล้ว ขอรับรองว่าเป็นผลงานวิจัยของนักศึกษาจริงและมีเนื้อหาสมบูรณ์ จึงลงชื่อไว้เป็นหลักฐาน

ลงชื่อ.....	ลงชื่อ.....	ลงชื่อ.....
อาจารย์ที่ปรึกษา	อาจารย์ที่ปรึกษาร่วม	อาจารย์ที่ปรึกษาร่วม