การหาค่าเหมาะที่สุดของทรานส์ฟอร์เมอร์สสำหรับการแปลภาษาด้วยเครื่องประสาทที่ใช้ทรัพยากรต่ำด้วยการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค

โดย
นายพุฒิสรรค์ ชาติชาญชัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ แผน ก แบบ ก 2
ภาควิชาวิศวกรรมไฟฟ้า
มหาวิทยาลัยศิลปากร
ปีการศึกษา 2567
ลิขสิทธิ์ของมหาวิทยาลัยศิลปากร

การหาค่าเหมาะที่สุดของทรานส์ฟอร์เมอร์สสำหรับการแปลภาษาด้วยเครื่องประสาทที่ใช้ทรัพยากรต่ำด้วยการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค

โดย
นายพุฒิสรรค์ ชาติชาญชัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ แผน ก แบบ ก 2
ภาควิชาวิศวกรรมไฟฟ้า
มหาวิทยาลัยศิลปากร
ปีการศึกษา 2567
ลิขสิทธิ์ของมหาวิทยาลัยศิลปากร

OPTIMIZING HYPERPARAMETERS OF TRANSFORMERS USING PARTICLE SWARM OPTIMIZATION FOR LOW-RESOURCE NEURAL MACHINE TRANSLATION

By

MR. Puttisan CHARTCHARNCHAI

A Thesis Submitted in Partial Fulfillment of the Requirements

for Master of Engineering ELECTRICAL AND COMPUTER ENGINEERING

Department of ELECTRICAL ENGINEERING

Academic Year 2024

Copyright of Silpakorn University

หัวข้อ	การหาค่าเหมาะที่สุดของทรานส์ฟอร์เมอร์สสำหรับการแปลภาษาด้วยเครื่องประสาทที่ใช้ทรัพยากรต่ำด้วยการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค
โดย	นายพุฒิสรรค์ ชาติชาญชัย
สาขาวิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์ แผน ก แบบ ก 2
อาจารย์ที่ปรึกษาหลัก	ผู้ช่วยศาสตราจารย์ ดร. ยุทธนา เจวจินดา

คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยศิลปากร ได้รับพิจารณาอนุมัติให้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์และ
(ผู้ช่วยศาสตราจารย์ ดร. อรุณศรี ลีจีรจำเนียร) เทคโนโลยีอุตสาหกรรม

พิจารณาเห็นชอบโดย

.....ประธานกรรมการ
(รองศาสตราจารย์ ดร. ชูเกียรติ สอดศรี)

.....อาจารย์ที่ปรึกษาหลัก
(ผู้ช่วยศาสตราจารย์ ดร. ยุทธนา เจวจินดา)

.....ผู้ทรงคุณวุฒิภายใน
(ผู้ช่วยศาสตราจารย์ ดร. จิรัฏฐ์ เหมือนชู)

.....ผู้ทรงคุณวุฒิภายนอก
(รองศาสตราจารย์ ดร. ยรรยงค์ พันธ์สวัสดิ์)

660920010 : วิศวกรรมไฟฟ้าและคอมพิวเตอร์ แผน ก แบบ ก 2
คำสำคัญ : การแปลภาษาด้วยเครื่องประสาท, แบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส, อัลกอริทึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาค, การค้นหาค่าไฮเปอร์พารามิเตอร์ที่เหมาะสม

นาย พุฒิสรรค์ ชาติชาญชัย: การหาค่าเหมาะที่สุดของทรานส์ฟอร์เมอร์สสำหรับการแปลภาษาด้วยเครื่องประสาทที่ใช้ทรัพยากรต่ำด้วยการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก : ผู้ช่วยศาสตราจารย์ ดร. ยุทธนา เจวจินดา

วิทยานิพนธ์นี้มีวัตถุประสงค์เพื่อปรับปรุงประสิทธิภาพของแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส (Transformers) สำหรับการแปลภาษาด้วยเครื่องประสาทในสภาวะที่มีข้อมูลจำกัด (Low-Resource Neural Machine Translation) โดยใช้วิธีการหาค่าเหมาะสมที่สุดของค่าไฮเปอร์พารามิเตอร์ด้วยอัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization) โดยได้เสนอแนวทางในการพัฒนาอัลกอริทึมเดิมที่ใช้กับตัวปัญหาเลขจำนวนจริงเพื่อให้สามารถจัดการกับค่าไฮเปอร์พารามิเตอร์ที่มีลักษณะเป็นตัวแปรเชิงกลุ่ม (Categorical Variables) ผ่านการใช้เวกเตอร์ความน่าจะเป็น (Probability Vector)

การดำเนินการวิจัยใช้ชุดข้อมูลคู่ขนานภาษาเยอรมัน-อังกฤษ (IWSLT14) และภาษาไทย-อังกฤษ (SCB) เพื่อประเมินประสิทธิภาพของแบบจำลอง พบว่าการใช้อัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคที่พัฒนาขึ้นสามารถค้นหาค่าไฮเปอร์พารามิเตอร์ที่ทำให้แบบจำลองมีประสิทธิภาพสูงขึ้นได้เมื่อเปรียบเทียบกับค่าไฮเปอร์พารามิเตอร์พื้นฐาน โดยในชุดข้อมูลภาษาเยอรมัน-อังกฤษ คะแนน BLEU เพิ่มขึ้นจาก 28.5 เป็น 31.5 และในชุดข้อมูลภาษาไทย-อังกฤษ คะแนน BLEU เพิ่มขึ้นจาก 10.1 เป็น 12.1 นอกจากนี้ แบบจำลองที่ได้ยังถูกนำไปทดสอบบนระบบสมองกลฝังตัวราสเบอรี่พายซึ่งสามารถทำงานได้อย่างมีประสิทธิภาพ เพื่อแสดงว่าแบบจำลองมีขนาดเล็กเพียงพอ ผลการวิจัยชี้ให้เห็นว่าการค้นหาค่าไฮเปอร์พารามิเตอร์ด้วยอัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคที่พัฒนาขึ้น สามารถเพิ่มประสิทธิภาพของแบบจำลองทรานส์ฟอร์เมอร์สในงานแปลภาษาด้วยเครื่องประสาทในสภาวะที่มีทรัพยากรจำกัดได้อย่างมีประสิทธิภาพ

660920010 : Major ELECTRICAL AND COMPUTER ENGINEERING
Keyword : Neural Machine Translation, Transformers, Particle Swarm Optimization (PSO), Hyperparameter Optimization

MR. Puttisan CHARTCHARNCHAI : Optimizing Hyperparameters of Transformers Using Particle Swarm Optimization for Low-Resource Neural Machine Translation Thesis advisor : Assistant Professor Yutana Jewajinda, Ph.D.

This thesis presents an approach to improve the performance of Transformers models for low-resource neural machine translation using the particle swarm optimization algorithm. Using probability vectors, we modify the original algorithm for real number problems to handle categorical variables. The research uses the German-English (IWSLT14) and Thai-English (SCB) parallel datasets to evaluate the performance of the models. The developed particle swarm optimization algorithm can search for near-optimum hyperparameter values that make the models more efficient than the baseline hyperparameter values. In the German-English dataset, the BLEU score increases from 28.5 to 31.5 in the Thai-English dataset, the BLEU score increases from 10.1 to 12.1. In addition, the obtained model is validated on a Raspberry Pi embedded system, where it performs well, to demonstrate that the model is sufficiently compact. The results indicate that the developed particle swarm optimization algorithm for hyperparameter search can effectively improve the performance of the transformer model in neural machine translation tasks under resource constraints.

กิตติกรรมประกาศ

ข้าพเจ้าขอขอบคุณ ผู้ช่วยศาสตราจารย์ ดร. ยุทธนา เจวจินดา อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ได้กรุณาให้คำแนะนำ แนวคิด และคำปรึกษาอย่างต่อเนื่องตลอดระยะเวลาการทำวิจัย จนทำให้ข้าพเจ้าสามารถดำเนินงานวิจัยจนสำเร็จลุล่วงไปได้ด้วยดี

ขอขอบพระคุณ รองศาสตราจารย์ ดร. ชูเกียรติ สอดศรี ประธานกรรมการสอบวิทยานิพนธ์, รองศาสตราจารย์ ดร. ยรรยงค์ พันธ์สวัสดิ์ ผู้ทรงคุณวุฒิภายนอก และ ผู้ช่วยศาสตราจารย์ ดร. จิรัฏฐ์ เหมือนชู ผู้ทรงคุณวุฒิภายใน ที่ได้กรุณาให้ข้อเสนอแนะอันเป็นประโยชน์อย่างยิ่ง ซึ่งช่วยให้วิทยานิพนธ์ฉบับนี้มีความสมบูรณ์มากยิ่งขึ้น

ข้าพเจ้าขอขอบพระคุณ คณาจารย์ประจำภาควิชาวิศวกรรมไฟฟ้า ทุกท่าน ที่ได้ถ่ายทอดความรู้ รวมทั้งให้การสนับสนุนในด้านต่าง ๆ ไม่ว่าจะเป็นทุนการศึกษา อุปกรณ์สำหรับการวิจัย และประสบการณ์อันมีค่า ซึ่งช่วยให้ข้าพเจ้าสามารถทำวิจัยได้อย่างมีประสิทธิภาพ

ท้ายที่สุดนี้ ข้าพเจ้าขอขอบพระคุณมารดาและคุณกชพร พิทักษ์บูรพา ผู้เปรียบเสมือนมารดา ตลอดจนสมาชิกในครอบครัวและเพื่อน ๆ ทุกคน ที่ได้ให้กำลังใจและให้ความช่วยเหลือในด้านต่าง ๆ จนข้าพเจ้าสามารถดำเนินการจัดทำวิทยานิพนธ์ฉบับนี้จนสำเร็จลงได้ด้วยดี

พุฒิสรรค์ ชาติชาญชัย

มหาวิทยาลัยศิลปากร

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ฌ

สารบัญภาพ ..... ญ

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์งานวิจัย 2

1.3 สมมติฐานของงานวิจัย 3

1.4 ขอบเขตของงานวิจัย 3

1.5 ความจำกัดของงานวิจัย 3

1.6 ประโยชน์ที่คาดว่าจะได้รับ 4

บทที่ 2 วรรณกรรมที่เกี่ยวข้อง 5

2.1 ความรู้พื้นฐานและวรรณกรรมที่เกี่ยวข้องกับการเรียนรู้เชิงลึก 5

2.2 แบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส 6

2.3 ทฤษฎีการแปลภาษาด้วยเครื่องประสาท (Neural Machine Translation หรือ NMT) 13

2.4 การหาค่าเหมาะที่สุดของค่าไฮเปอร์พารามิเตอร์ 14

2.5 ชุดข้อมูลภาษาไทย - ภาษาอังกฤษ 28

2.6 การหาค่าเหมาะที่สุดของแบบจำลองการเรียนรู้เชิงลึกที่ใช้ทรัพยากรต่ำ 29

2.7 เกณฑ์การวัดคะแนน BLEU (Bilingual Evaluation Understudy) 29

2.8 วรรณกรรมที่เกี่ยวกับการประยุกต์ใช้อัลกอริทึมเชิงพันธุกรรมในแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส 31
บทที่ 3 วิธีดำเนินการวิจัย 36
3.1 รวบรวมข้อมูลและกำหนดขอบเขตของงาน 36
3.2 ศึกษาทฤษฎีที่เกี่ยวข้องกับการประยุกต์ใช้อัลกอริทึมเชิงพันธุกรรมกับแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส 37
3.3 ออกแบบอัลกอริทึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาค 39
3.4 ออกแบบการทดลอง 44
3.5 การดำเนินการทดลอง 45
3.6 รวบรวม วิเคราะห์ผลการทดลองและจัดทำรายงานวิทยานิพนธ์ 46
บทที่ 4 ผลการดำเนินงาน 47
4.1 รูปแบบการทดลอง 47
4.2 ขั้นตอนการทดลอง 51
4.3 ผลการทดลอง 51
4.4 การประยุกต์ใช้แบบจำลองกับระบบสมองกลฝังตัว 57
บทที่ 5 สรุป อภิปรายผล และข้อเสนอแนะ 60
5.1 บทสรุปของงานวิจัย 60
5.2 ข้อเสนอแนะ 61
รายการอ้างอิง 62
ประวัติผู้เขียน 66

สารบัญตาราง

หน้า

ตารางที่ 1 ค่าไฮเปอร์พารามิเตอร์ของแบบจำลองทรานส์ฟอร์เมอร์ส 31

ตารางที่ 2 ค่าไฮเปอร์พารามิเตอร์ที่ใช้ในอัลกอริทึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาค 50

ตารางที่ 3 ค่าไฮเปอร์พารามิเตอร์ที่ได้จากอัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคที่พัฒนาขึ้นของการแปลภาษาเยอรมันเป็นภาษาอังกฤษ 52

ตารางที่ 4 ผลลัพธ์ของคะแนน BLEU ที่ Fine-tuning ของชุดข้อมูลภาษาเยอรมัน-อังกฤษ..... 52

ตารางที่ 5 ค่าไฮเปอร์พารามิเตอร์ที่ได้จากอัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคที่พัฒนาขึ้นของการแปลภาษาไทยเป็นภาษาอังกฤษ 54

ตารางที่ 6 ผลลัพธ์ของคะแนน BLEU ที่ Fine-tuning ของชุดข้อมูลภาษาไทย-อังกฤษ 55

มหาวิทยาลัยศิลปากร

สารบัญภาพ

หน้า

ภาพที่ 1 ภาพรวมการสร้างข้อความ 5
ภาพที่ 2 แบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส 6
ภาพที่ 3 ภาพรวมการปรับแต่งค่าไฮเปอร์พารามิเตอร์ของอัลกอริทึมการเรียนรู้เครื่อง 15
ภาพที่ 4 การค้นหาแบบตาราง และ การค้นหาแบบสุ่ม 18
ภาพที่ 5 กระบวนการทำงานของ Evolution Strategy 25
ภาพที่ 6 วิธีการที่เกี่ยวข้องกับเครื่องแปลภาษาประสาท 33
ภาพที่ 7 กระบวนการของอัลกอริทึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาค 40
ภาพที่ 8 ผลลัพธ์คะแนน BLEU จากกระบวนการค้นหาของภาษาเยอรมันเป็นภาษาอังกฤษ 53
ภาพที่ 9 ผลลัพธ์คะแนน BLEU เทียบกับขนาดของชุดข้อมูลในการแปลภาษาเยอรมันเป็นภาษาอังกฤษ 53
ภาพที่ 10 ผลลัพธ์คะแนน BLEU จากกระบวนการค้นหาของภาษาไทยเป็นภาษาอังกฤษ 56
ภาพที่ 11 ผลลัพธ์คะแนน BLEU เทียบกับขนาดของชุดข้อมูลในการแปลภาษาไทยเป็นภาษาอังกฤษ 56
ภาพที่ 12 การใช้งานจริงต่อบอร์ดราสเบอรรี่พายกับหน้าจอ 59
ภาพที่ 13 GUI ในระบบสมองกลฝังตัวราสเบอร์รี่พาย 59

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การแปลภาษาด้วยเครื่องประสาท (Neural Machine Translation) เป็นแนวทางหลักในปัจจุบันในการแปลภาษาด้วยคอมพิวเตอร์ โดยที่แนวทางนี้เน้นใช้แบบจำลองโครงข่ายประสาทเทียมเชิงลึก [1] การแปลภาษาด้วยเครื่องประสาทมีการพัฒนาอย่างรวดเร็วในช่วงไม่กี่ปีที่ผ่านมาและมีการนำไปประยุกต์ใช้งานที่หลากหลาย การแปลภาษานี้เริ่มจากการใช้แบบจำลองเครือข่ายประสาทเทียมเชิงลึกแบบดั้งเดิมโดยเฉพาะโครงข่ายประสาทเทียมแบบวนซ้ำ (Recurrent Neural Network) แบบต่าง ๆ เช่น Long Short-Term Memory (LSTM) และ Gated Recurrent Unit (GRU) จากนั้นมีการเพิ่มการจับความสนใจ (Attention) เข้าไปในโครงสร้างของโครงข่ายประสาทเทียมแบบวนซ้ำ จนกระทั่งพัฒนามาสู่สถาปัตยกรรมทรานส์ฟอร์เมอร์ส (Transformer) ในปัจจุบัน

แบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สเป็นแบบจำลองตามลำดับ ซึ่งประกอบด้วยตัวเข้ารหัสและตัวถอดรหัส [2] นำแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สมาประยุกต์ใช้กับการแปลภาษาด้วยเครื่อง (Machine Translation) จำเป็นต้องใช้ชุดข้อมูลขนาดใหญ่มากกว่าล้านตัวอย่าง ซึ่งเหมาะกับภาษาที่มีการใช้งานอย่างแพร่หลาย เช่น ภาษาอังกฤษ ภาษาจีน ภาษาเยอรมัน และภาษาฝรั่งเศส แต่จะมีข้อจำกัดในกรณีภาษาที่ไม่สามารถเตรียมชุดข้อมูลขนาดใหญ่ได้เพียงพอ หรือชุดข้อมูลที่สอนมีขนาดเล็กในระดับแสนตัวอย่างหรือน้อยกว่า การสอนแบบจำลองชนิดทรานส์ฟอร์เมอร์สด้วยชุดข้อมูลที่มีขนาดเล็กจะต้องให้ความสำคัญกับการปรับแต่งค่าไฮเปอร์พารามิเตอร์ที่เหมาะสมซึ่งเป็นปัญหาการหาค่าเหมาะที่สุด (Optimization) [3], [4]

อัลกอริทึมเชิงวิวัฒนาการถือเป็นแนวทางหลักแนวทางหนึ่งที่ใช้แก้ปัญหาการหาค่าเหมาะที่สุดของค่าไฮเปอร์พารามิเตอร์สำหรับแบบจำลองการเรียนรู้ของเครื่อง และได้มีการนำมาใช้งานกับโครงข่ายประสาทเทียมเชิงลึกซึ่งเป็นการค้นหาค่าไฮเปอร์พารามิเตอร์ที่ทำให้แบบจำลองมีประสิทธิภาพดีขึ้นเพื่อให้เหมาะสมกับขนาดของชุดข้อมูล [5], [6]

การหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization) เป็นหนึ่งในอัลกอริทึมเชิงวิวัฒนาการที่สำคัญ ซึ่งอาศัยแนวคิดการค้นหาแบบรวมกลุ่มของสิ่งมีชีวิต โดยแต่ละอนุภาคจะเคลื่อนที่เพื่อค้นหาคำตอบที่เหมาะสม พร้อมทั้งมีการสื่อสารและแลกเปลี่ยนข้อมูลระหว่าง

กัน ในกระบวนการนี้ หากอนุภาคตัวใดเข้าใกล้คำตอบที่ดี อนุภาคอื่นจะถูกดึงดูดให้เคลื่อนที่เข้าหาจุดนั้น ส่งผลให้เกิดการสะสมความรู้และการปรับปรุงตำแหน่งอย่างต่อเนื่อง ซึ่งช่วยให้กระบวนการค้นหาคำตอบมีความรวดเร็วและมีประสิทธิภาพมากขึ้น [4], [5]

การหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค ได้รับการประยุกต์ใช้ในการค้นหาค่าไฮเปอร์พารามิเตอร์ของแบบจำลองการเรียนรู้ของเครื่องมาก่อน [4], [7] และต่อมาได้ถูกนำมาใช้งานกับโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Networks) [8] อย่างไรก็ตามยังไม่มีการนำมาใช้งานกับแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สโดยเฉพาะกับปัญหาการแปลภาษาการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคดั้งเดิม ใช้กับปัญหาจำนวนจริงแต่ค่าไฮเปอร์พารามิเตอร์ของแบบจำลองการเรียนรู้เชิงลึกชนิดทรานฟอร์เมอร์สเป็นเลขจำนวนเต็มเป็นส่วนใหญ่และมีการผสมกับเลขจำนวนจริงจึงทำให้มีงานวิจัยนำเสนอแนวทางการปรับปรุงให้ใช้ได้กับเลขจำนวนเต็ม [7], [9], [10]

งานวิจัยนี้มีจุดมุ่งหมายเพื่อออกแบบอัลกอริทึมใหม่ตามแนวทางของการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคสำหรับค้นหาค่าไฮเปอร์พารามิเตอร์ของแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สในงานแปลภาษาไทย-อังกฤษ ภายใต้ข้อจำกัดของชุดข้อมูลฝึกที่มีขนาดเล็กไม่เกินหนึ่งล้านคู่ประโยค โดยในงานวิจัยนี้ได้กำหนดขอบเขตของชุดข้อมูลไว้ที่ไม่เกินสองแสนคู่ประโยค นอกจากนี้ยังมีการนำค่าไฮเปอร์พารามิเตอร์ที่ค้นหาได้ไปฝึกแบบจำลองที่เหมาะสมกับการประยุกต์ใช้ในระบบสมองกลฝังตัวหรือเครื่องคอมพิวเตอร์ที่มีทรัพยากรฮาร์ดแวร์ประสิทธิภาพระดับปานกลาง

1.2 วัตถุประสงค์งานวิจัย

จุดมุ่งหมายหลักของงานวิจัยคือ

1. ศึกษาปัญหาการแปลภาษาไทยเป็นภาษาอังกฤษ โดยใช้แบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส ซึ่งเป็นแบบจำลองการเรียนรู้เชิงลึกสำหรับการแปลภาษาที่มีประสิทธิภาพสูง
2. ออกแบบอัลกอริธึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคที่เหมาะสมกับปัญหาการค้นหาค่าไฮเปอร์พารามิเตอร์ของสถาปัตยกรรมของแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สโดยใช้ตัวอย่างสอนขนาดเล็ก

3. นำแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส ที่เหมาะสมสำหรับการแปลภาษาไทยเป็นภาษาอังกฤษที่ได้จากข้อสองนำมาทดสอบเพื่อใช้งานในระบบสมองกลฝังตัว

1.3 สมมติฐานของงานวิจัย

1. แบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส ปรับค่าไฮเปอร์พารามิเตอร์มีประสิทธิภาพมากกว่าแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สพื้นฐาน
2. การใช้อัลกอริธึมเชิงวิวัฒนาการทำให้ค่าไฮเปอร์พารามิเตอร์ที่ดีและทำให้แบบจำลองมีประสิทธิภาพมากขึ้น

1.4 ขอบเขตของงานวิจัย

ขอบเขตของการวิจัยการหาค่าเหมาะที่สุดของแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส โดยการใช้อัลกอริทึมการหาค่าเหมาะที่สุด มีดังนี้

1. พัฒนาอัลกอริทึมโดยใช้ภาษาไพทอน
2. ทำการจัดเตรียมชุดข้อมูลขนาดเล็กไม่เกินหนึ่งล้านคู่ประโยค สำหรับการแปลภาษาไทยเป็นภาษาอังกฤษ และภาษาอังกฤษเป็นภาษาไทย
3. พัฒนาซอฟต์แวร์สำหรับค้นหาสถาปัตยกรรมของแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส โดยใช้กรอบการเรียนรู้เชิงลึก
4. ทำการทดลองและเก็บข้อมูลการค้นหาสถาปัตยกรรมของแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส โดยใช้งานกรอบเรียนรู้เชิงลึก
5. นำแบบจำลองที่ได้มาทดลองในระบบสมองกลฝังตัวราสเบอร์รี่พาย (Raspberry Pi)

1.5 ความจำกัดของงานวิจัย

1. วิธีการที่ได้จากการวิจัยมีขนาดชุดข้อมูลขนาดเล็ก เนื่องจากต้องการวิจัยในการหาค่าที่เหมาะสมในการปรับค่าไฮเปอร์พารามิเตอร์ด้วยอัลกอริทึมวิธีการหาค่าเหมาะที่สุด
2. งานวิจัยนี้ไม่ได้มุ่งเน้นการเปรียบเทียบกับแบบจำลองขนาดใหญ่ เนื่องจากวัตถุประสงค์หลักคือการปรับแต่งค่าไฮเปอร์พารามิเตอร์บนชุดข้อมูลขนาดเล็ก ซึ่งในงานวิจัยนี้กำหนดให้มีขนาดไม่เกิน 200,000 คู่ประโยค

3. ทำการสอนแบบจำลองโดยใช้คอมพิวเตอร์ทั่วไป เวลาในการสอนแบบจำลองมากกว่าซุปเปอร์คอมพิวเตอร์

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. วิธีการที่นำเสนอทำให้การหาค่าไฮเปอร์พารามิเตอร์ที่เหมาะสมที่สุด ทำได้อย่างสะดวกและมีประสิทธิภาพมากขึ้น
2. แบบจำลองที่สร้างขึ้นด้วยค่าไฮเปอร์พารามิเตอร์ ที่หาได้จากอัลกอริทึมนี้สามารถทำงานได้อย่างมีประสิทธิภาพ
3. เป็นแนวทางให้แก่ผู้ที่สนใจศึกษาเกี่ยวกับแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส

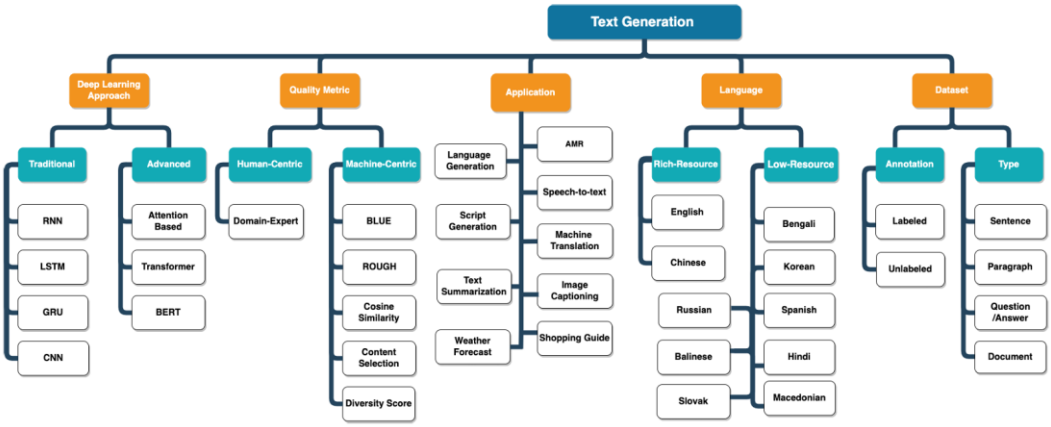
มหาวิทยาลัยศิลปากร

บทที่ 2
วรรณกรรมที่เกี่ยวข้อง

ในส่วนของการทบทวนวรรณกรรมจะทำการนำเสนอความรู้พื้นฐานและงานวิจัยที่เกี่ยวข้องกับการเรียนรู้เชิงลึกโดยเนื้อหาในส่วนนี้ประกอบด้วย ความรู้พื้นฐานและวรรณกรรมที่เกี่ยวข้องกับการเรียนรู้เชิงลึก, แบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส, ทฤษฎีการแปลภาษาด้วยเครื่องประสาท, อัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค และวรรณกรรมที่เกี่ยวข้องกับการประยุกต์ใช้อัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคในการหาค่าไฮเปอร์พารามิเตอร์ให้กับแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส

2.1 ความรู้พื้นฐานและวรรณกรรมที่เกี่ยวข้องกับการเรียนรู้เชิงลึก

จากบทความสำรวจการเรียนรู้เชิงลึก [1] ได้มีการทบทวนวรรณกรรมเกี่ยวกับการสร้างข้อความโดยใช้แบบจำลองเครือข่ายประสาทลึก โครงสร้างการเรียนรู้เชิงลึกแบบดั้งเดิมมี 4 แบบจำลอง 1.Convolutional Neural Network (CNN) 2.Recurrent Neural Network (RNN) 3.Long Short-Term Memory (LSTM) 4.Gated Recurrent Unit (GRU) โครงสร้างการเรียนรู้เชิงลึกแบบขั้นสูงมี 3 แบบจำลอง 1.Attention Based 2.Transformer 3.BERT



ภาพที่ 1 ภาพรวมการสร้างข้อความ

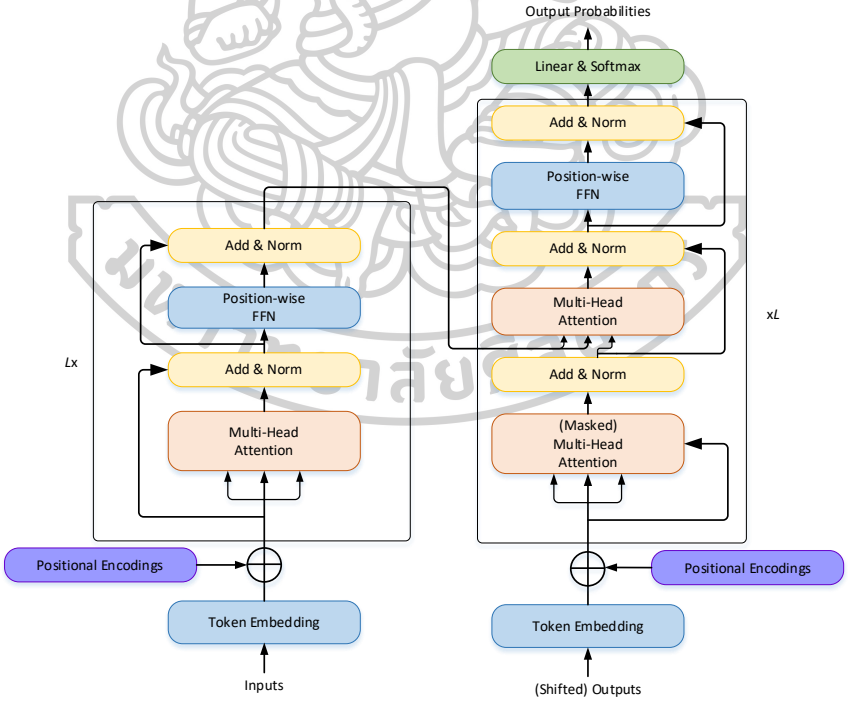
ที่มา: บทความ [1]

บทความ [1] ให้มุมมองที่ครอบคลุมเกี่ยวกับการพัฒนาของเทคโนโลยีและระบุช่องว่างในการวิจัยที่สำคัญสำหรับแนวทางการวิจัยในอนาคตการสร้างข้อความเป็นด้านสร้างสรรค์ของ

ปัญญาประดิษฐ์ เป็นเวลาหลายทศวรรษแล้วที่นักคอมพิวเตอร์มีความพยายามในการพัฒนาปัญญาประดิษฐ์ให้ใกล้เคียงกับมนุษย์ การสร้างข้อความได้รับความนิยมอย่างกว้างขวางเนื่องจากมีการใช้งานแอปพลิเคชันหลากหลายและข้อความออนไลน์ก็มีจำนวนมากด้วยสังคมออนไลน์ สำนักข่าว และแหล่งข้อมูลอื่น ๆ ที่ใช้ข้อความจำนวนมหาศาล แอปพลิเคชันบางตัวที่ได้รับประโยชน์จากการสร้างข้อความ ได้แก่ การสร้างและการคาดเดาคำหรือประโยค การพัฒนาแชทบอท การเขียนบทละคร บทกวีและอื่น ๆ นอกจากนี้การสร้างข้อความยังดึงดูดความสนใจของนักวิจัยในด้านการประยุกต์ใช้ในการศึกษา อุตสาหกรรม และเครือข่ายทางสังคม

ในบริบทนี้ การทบทวนวรรณกรรมอย่างเป็นระบบได้วิเคราะห์รายงานที่เกี่ยวข้อง 90 ฉบับ โดยเน้นห้าแง่มุมที่สำคัญ ได้แก่ แนวทางการสร้างข้อความ ตัวชี้วัดคุณภาพ ชุดข้อมูล ภาษา และแอปพลิเคชันในบริบทของการเรียนรู้เชิงลึก ผลการทบทวนช่วยระบุแนวโน้มที่สำคัญ ตลอดจนความท้าทายที่ต้องเผชิญในการพัฒนาเทคโนโลยีนี้ให้มีประสิทธิภาพยิ่งขึ้น

2.2 แบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส



ภาพที่ 2 แบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส

ที่มา: บทความ [2]

แบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สได้รับการพัฒนาอย่างก้าวกระโดดและถือว่าเป็นแบบจำลองหลักในงานวิจัยที่เกี่ยวข้องกับการประมวลผลภาษาธรรมชาติ โดยบทความ [2] ได้นำเสนอโครงสร้างแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สแบบดั้งเดิม ซึ่งประกอบด้วยสองส่วนสำคัญคือตัวเข้ารหัส (Encoder) ตัวถอดรหัส (Decoder) ในแนวทางนี้จะมีการใช้สถาปัตยกรรมของแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สแบบเต็ม ซึ่งรวมถึงทั้งตัวเข้ารหัสและตัวถอดรหัสแนวทางนี้ใช้ในงานสร้างแบบจำลองลำดับต่อลำดับเพื่อการแปลภาษาของเครื่อง ตัวเข้ารหัสใช้ลำดับอินพุตและสร้างการแสดงความยาวคงที่ซึ่งจะถูกใช้โดยตัวถอดรหัสเพื่อสร้างลำดับเอาต์พุต

2.2.1 ส่วนประกอบของแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สมีดังนี้

1. **Token embedding** ทำหน้าที่แปลงคำหรือตัวอักษรในประโยคเป็นเวกเตอร์เชิงตัวเลข ซึ่งแต่ละเวกเตอร์จะบรรจุข้อมูลเชิงความหมายและไวยากรณ์ของโทเค็นนั้น ๆ
2. **Positional Encodings** ทำหน้าที่เพิ่มข้อมูลเกี่ยวกับตำแหน่งของคำในประโยคเข้าไปในเวกเตอร์ที่ได้จาก Token Embedding เพื่อให้แบบจำลองสามารถเข้าใจลำดับของคำในประโยคได้ โดยใช้การเข้ารหัสเชิงคณิตศาสตร์ เช่น Sine และ Cosine
3. **Layer Normalization** ทำหน้าที่ปรับค่าเฉลี่ยและค่าความแปรปรวนของพารามิเตอร์ในแต่ละชั้นให้เหมาะสม ช่วยให้การฝึกแบบจำลองมีเสถียรภาพและลดปัญหาการเรียนรู้ที่ช้า
4. **SoftMax** เป็นฟังก์ชันที่แปลงเวกเตอร์ของค่าจริงให้เป็นเวกเตอร์ของความน่าจะเป็น โดยแต่ละค่าจะอยู่ในช่วงระหว่าง 0 ถึง 1 และผลรวมของความน่าจะเป็นทั้งหมดในเวกเตอร์จะเท่ากับ 1
5. **Multi-Head Attention** ทำหน้าที่ในการคำนวณความสำคัญของคำในประโยค โดยให้ค่าน้ำหนักกับคำแต่ละคำตามบริบท ช่วยให้แบบจำลองสามารถโฟกัสกับคำสำคัญหลายคำพร้อมกันได้
6. **Position-wise Feed-Forward Network** ทำหน้าที่ประมวลผลข้อมูลที่ได้จาก Attention โดยใช้ฟังก์ชันไม่เชิงเส้น เพื่อเพิ่มความสามารถในการเรียนรู้ลักษณะที่ซับซ้อนของข้อมูล

2.2.2 สมการของแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส

$$Attention(Q,K,V) = softmax\left(\frac{QK^{\top}}{\sqrt{D_k}}\right)V = AV \quad (2.1)$$

จากสมการที่ (2.1) อธิบายองค์ประกอบในสมการ:

- Q (Query Matrix) เมทริกซ์ที่แสดงถึงคำถามหรือบริบทที่ต้องการค้นหา
- K (Key Matrix) เมทริกซ์ที่แสดงถึงคำหลักหรือค่าที่ใช้เปรียบเทียบ
- V (Value Matrix) เมทริกซ์ที่แสดงถึงคำตอบหรือค่าที่จะใช้ในผลลัพธ์
- D_k (Dimension of Key Vectors) ขนาดของเวกเตอร์ Key

ขั้นตอนการทำงาน:

1. คำนวณ dot product ระหว่าง Q และ K^T เพื่อหาความคล้ายคลึงกันระหว่างแต่ละคำในประโยค
2. หารด้วย$\sqrt{D_k}$เพื่อทำให้ค่ามีขนาดที่เหมาะสมและป้องกันการเกิด vanishing gradient หรือ exploding gradient
3. ใช้ฟังก์ชัน softmax เพื่อแปลงค่าให้อยู่ในช่วง [0,1] และให้ผลรวมของค่ามีค่าเท่ากับ 1
4. นำผลลัพธ์จาก softmax ไปคูณกับ V เพื่อรวมบริบทของคำที่สำคัญ

$$\mathrm{MultiHeadAttn(Q,K,V)} = \mathrm{Concat(head_1,\ldots,head_H)W^0} \quad (2.2)$$

จากสมการที่ (2.2) องค์ประกอบในสมการ

- $\boldsymbol{Concat(\ldots)}$ คือการนำผลลัพธ์จากแต่ละ Head มาต่อกันในแนวแกนความกว้าง (axis = 1)
- $\boldsymbol{W^O}$ เป็นเมทริกซ์ Weight ที่ใช้ปรับผลรวมของทุก Head ให้ได้ข้อมูลที่มีมิติเหมาะสมกับการส่งต่อไปยังชั้นถัดไป

$$where\ head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (2.3)$$

จากสมการที่ (2.3) องค์ประกอบในสมการ

- QW_i^Q, KW_i^K, VW_i^V เป็นเมทริกซ์ Weight ที่แปลงข้อมูลจาก Q, K, V ให้กลายเป็นข้อมูลเฉพาะของแต่ละ Head การแปลงนี้ช่วยให้แต่ละ Head สามารถเรียนรู้รูปแบบความสัมพันธ์ที่แตกต่างกันในข้อมูลได้

สมการ Position-wise Feed-Forward Network

$$\mathrm{FFN}(\mathrm{H}') = \mathrm{ReLU}(\mathrm{H}'W^1 + b^1)W^2 + b^2 \quad (2.4)$$

จากสมการที่ (2.4) องค์ประกอบในสมการ

- $\boldsymbol{H'}$ = ข้อมูลอินพุตที่ได้จากการทำ Self-Attention และผ่าน Layer Normalization
- $\boldsymbol{W^1, W^2}$ = เมทริกซ์ Weight ที่เรียนรู้ได้ ซึ่งใช้สำหรับแปลงข้อมูลในชั้น FFN
- $\boldsymbol{b^1, b^2}$ = ค่าไบแอส (Bias) ในแต่ละชั้น
- $\boldsymbol{ReLU}$ = ฟังก์ชัน Activation แบบไม่เชิงเส้น ทำหน้าที่นำค่าเชิงลบให้เป็นศูนย์ เพื่อเพิ่มความสามารถในการจับรูปแบบที่ซับซ้อนมากขึ้น

$$\mathrm{H}' = \mathrm{LayerNorm}(\mathrm{SelfAttention}(\mathrm{X}) + \mathrm{X}) \quad (2.5)$$

จากสมการที่ (2.5) องค์ประกอบในสมการ:

- $SelfAttention(X)$: ผลลัพธ์จากการทำ Self-Attention
- X: ข้อมูลอินพุตตั้งต้นที่เข้ามาใน Layer
- $LayerNorm$: ทำหน้าที่ทำให้ค่าต่าง ๆ ในเวกเตอร์อยู่ในช่วงที่มีค่าเฉลี่ยเป็นศูนย์และส่วนเบี่ยงเบนมาตรฐานเป็นหนึ่ง

การทำงานของสมการ

1. คำนวณผลลัพธ์จาก Self-Attention
2. มีการใช้ Residual Connection โดยนำผลลัพธ์ของ Self-Attention มาบวกกับอินพุตดั้งเดิม (เพื่อเก็บบริบทดั้งเดิมไว้)
3. ทำการ Layer Normalization เพื่อปรับค่าให้อยู่ในช่วงที่สมดุลกัน ช่วยให้แบบจำลองเรียนรู้ได้เร็วขึ้นและลดปัญหา vanishing gradient หรือ exploding gradient

$$\mathrm{H} = \mathrm{LayerNorm}(\mathrm{FFN}(\mathrm{H}') + \mathrm{H}') \quad (2.6)$$

จากสมการที่ (2.6) องค์ประกอบในสมการ

- $\boldsymbol{FFN(H')}$ = ผลลัพธ์ที่ได้จากการทำ Position-wise Feed-Forward Network

- $\boldsymbol{H'}$ = ข้อมูลที่ผ่าน Self-Attention และ Layer Normalization
- $\boldsymbol{LayerNorm}$ = ใช้เพื่อปรับค่าให้อยู่ในช่วงสมดุลกัน

การทำงานของสมการ

1. นำผลลัพธ์จาก FFN มาบวกกับค่าของ H' (Residual Connection)
2. ทำการ Layer Normalization เพื่อรักษาสมดุลของข้อมูลและช่วยให้แบบจำลองมีความเสถียรมากขึ้น
3. ผลลัพธ์ที่ได้จะถูกนำไปใช้ในขั้นตอนถัดไปของแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส

สรุปกระบวนการของแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส

1. **Self-Attention:**
   - คำนวณความสัมพันธ์ระหว่างคำในประโยค
   - ใช้ **Q, K, V** ในการคำนวณ Attention Score
   - ใช้ Softmax เพื่อสร้างค่าน้ำหนัก
2. **Residual Connection + Layer Normalization:**
   - บวกผลลัพธ์จาก Self-Attention กับข้อมูลต้นฉบับ
   - ทำ Layer Normalization เพื่อรักษาสมดุลของข้อมูล
3. **Position-wise Feed-Forward Network:**
   - ผ่าน Layer แบบ Fully Connected ด้วยฟังก์ชัน ReLU
   - เพิ่มความสามารถในการจับรูปแบบที่ซับซ้อนของข้อมูล
4. **Residual Connection + Layer Normalization:**
   - บวกผลลัพธ์จาก FFN กับผลลัพธ์จาก Self-Attention
   - ทำ Layer Normalization เพื่อรักษาสมดุลของข้อมูล

2.2.3 หลักการทำงานของแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สมีดังนี้

2.2.3.1 การทำงานฝังอินพุตข้อมูล (Input Embedding)

เป็นขั้นตอนสำคัญในกระบวนการประมวลผลภาษาธรรมชาติ ที่แปลงข้อมูลโทเค็น (เช่น คำหรือสัญลักษณ์) ให้อยู่ในรูปแบบเวกเตอร์ทางคณิตศาสตร์ที่สามารถนำไปประมวลผลในโมเดลการเรียนรู้เชิงลึกได้ [11]

- การแปลงโทเค็นเป็นเวกเตอร์: ข้อมูลขาเข้าจะถูกแปลงเป็นโทเค็น เช่น หากข้อมูลขาเข้าเป็นประโยค โทเค็นอาจเป็นคำหรืออักขระ จากนั้นโทเค็นเหล่านี้จะถูกแปลงเป็นเวกเตอร์เชิงตัวเลข เวกเตอร์เหล่านี้มีลักษณะหลายมิติและประกอบด้วยข้อมูลเชิงความหมายและไวยากรณ์
- ตัวอย่าง: ให้คำว่า "banana" และ "mango" มีเวกเตอร์สมมุติที่แสดงความสัมพันธ์ในมิติทางคณิตศาสตร์ เช่น banana= [0.12,0.75,0.34] และmango=[0.15,0.78,0.32] การคำนวณระหว่างเวกเตอร์สามารถแสดงความสัมพันธ์ระหว่างคำ เช่น การวัดความคล้ายคลึงทางความหมาย
- ความสำคัญ: การฝังเวกเตอร์ช่วยให้แบบจำลองเข้าใจความสัมพันธ์เชิงคณิตศาสตร์ระหว่างคำในประโยคได้ เช่น การใช้เวกเตอร์ในพื้นที่มิติสูงที่มีข้อมูลเกี่ยวกับไวยากรณ์ ความหมาย และการใช้งานในบริบท

ด้วยกระบวนการนี้ ข้อมูลโทเค็นที่ไม่ต่อเนื่องสามารถถูกแปลงให้อยู่ในรูปแบบต่อเนื่อง ทำให้แบบจำลองสามารถเรียนรู้และวิเคราะห์ได้อย่างมีประสิทธิภาพมากขึ้น

2.2.3.2 การเข้ารหัสตามตำแหน่ง (Positional Encoding)

เป็นส่วนสำคัญในสถาปัตยกรรมของแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์เนื่องจากแบบจำลองไม่ได้ประมวลผลข้อมูลตามลำดับโดยธรรมชาติ เช่นเดียวกับโครงสร้างแบบลำดับแบบดั้งเดิม ดังนั้น เพื่อให้แบบจำลองสามารถพิจารณาลำดับของโทเค็นในข้อมูลขาเข้า การเข้ารหัสตามตำแหน่งจึงเพิ่มข้อมูลเกี่ยวกับตำแหน่งของแต่ละโทเค็นเข้าไปในเวกเตอร์ฝัง (Token Embedding) การเข้ารหัสตามตำแหน่งดำเนินการโดยการเพิ่มสัญญาณตำแหน่งที่ไม่ซ้ำกันให้กับการฝังของโทเค็น สัญญาณนี้สามารถสร้างได้

โดยใช้ฟังก์ชันทางคณิตศาสตร์ เช่น Sine และ Cosine โดยค่าที่ได้จะเพิ่มเข้าไปในเวกเตอร์ของโทเค็น เพื่อให้แบบจำลองเข้าใจตำแหน่งและความสัมพันธ์ในลำดับได้ [11]

การเข้ารหัสตามตำแหน่งทำให้แบบจำลองสามารถรักษาลำดับของข้อมูลและเข้าใจบริบทในลำดับได้อย่างมีประสิทธิภาพ ซึ่งเป็นปัจจัยสำคัญที่ทำให้แบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์เหมาะสมสำหรับงานที่ต้องการการวิเคราะห์ลำดับ เช่น การแปลภาษา การสรุปข้อความ และการสร้างข้อความ

2.2.3.3 ชั้นแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส (Transformer Blocks)

แบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์มีโครงสร้างที่ประกอบด้วยบล็อกหลายชั้นเรียงต่อกัน โดยแต่ละบล็อกประกอบด้วยส่วนประกอบสำคัญสองรายการ ได้แก่ Self-Attention แบบหลายหัว และ Position-Wise Feed-Forward Network [11]

1. Self-Attention แบบหลายหัว (Multi-Head Self-Attention): กลไกนี้ช่วยให้แบบจำลองสามารถประเมินความสำคัญของโทเค็นในลำดับ โดยพิจารณาความสัมพันธ์ระหว่างคำต่าง ๆ ตัวอย่างเช่น ในประโยค "Speak no lies" และ "He lies down" คำว่า "lies" มีความหมายที่แตกต่างกันตามคำบริบทที่อยู่รอบ ๆ Self-Attention ช่วยให้แบบจำลองสามารถเชื่อมโยงคำเหล่านี้เข้ากับบริบทที่ถูกต้องได้ โดยใช้ Query, Key, และ Value เพื่อคำนวณคะแนนความสำคัญ (Attention Scores)
2. Position-Wise Feed-Forward Network: เครือข่ายนี้ช่วยปรับปรุงความสามารถในการเรียนรู้ของแบบจำลอง โดยมีหน้าที่ดังนี้:
    - การเชื่อมต่อแบบทางลัด (Residual Connections): ช่วยให้ข้อมูลไหลผ่านเครือข่ายได้อย่างราบรื่น โดยข้ามการดำเนินการบางอย่าง
    - การทำบรรทัดฐาน (Layer Normalization): ปรับค่าพารามิเตอร์ให้อยู่ในช่วงที่เหมาะสม เพื่อเพิ่มเสถียรภาพระหว่างการฝึก
    - การแปลงเชิงเส้น (Linear Transformation): เพิ่มความสามารถของแบบจำลองในการเรียนรู้งานเฉพาะ เช่น การแปลภาษา หรือการสรุปข้อความ

การทำงานร่วมกันระหว่าง Self-Attention และ Feed-Forward Network ในบล็อกแต่ละชั้น ช่วยให้แบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สามารถจัดการกับข้อมูลขนาดใหญ่และเข้าใจบริบทของข้อมูลได้อย่างแม่นยำ

2.2.3.4 ชั้นเชิงเส้นและฟังก์ชันซอฟต์แม็กซ์ (Linear & SoftMax)

แบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์จำเป็นต้องแปลงข้อมูลที่ซับซ้อนเป็นการคาดการณ์ที่เข้าใจได้ ตัวอย่างเช่น การเลือกคำถัดไปในลำดับ ขั้นตอนนี้ดำเนินการผ่านสองชั้นสำคัญ [11] ได้แก่

1. บล็อกเชิงเส้น: เป็นชั้นเชื่อมต่อแบบเต็ม (Fully Connected Layer หรือ Dense Layer) ที่ทำหน้าที่แปลงการแสดงผลลัพธ์ในพื้นที่เวกเตอร์เชิงลึกให้อยู่ในรูปแบบคะแนน (logits) สำหรับแต่ละโทเค็นที่เป็นไปได้ ตัวอย่างเช่น หากแบบจำลองกำลังพิจารณาคำถัดไปในลำดับ "The cat sat on the ...", บล็อกเชิงเส้นจะกำหนดคะแนน logit ให้กับคำที่เป็นไปได้ เช่น "mat," "floor," หรือ "sofa"
2. ฟังก์ชันซอฟต์แม็กซ์: เป็นฟังก์ชันทางคณิตศาสตร์ที่ทำหน้าที่แปลงคะแนน logits ให้เป็นการกระจายความน่าจะเป็น ซึ่งค่าความน่าจะเป็นแต่ละค่าจะแสดงถึงระดับความเชื่อมั่นของแบบจำลองในแต่ละโทเค็น ยกตัวอย่าง คะแนน logits: [2.5, 0.3, -1.2] หลังผ่าน ฟังก์ชันซอฟต์แม็กซ์: [0.85, 0.12, 0.03] ฟังก์ชันซอฟต์แมกซ์ช่วยให้แบบจำลองสามารถคาดการณ์คำถัดไปที่เป็นไปได้สูงสุด เช่น ในกรณีนี้คำว่า "mat" มีความน่าจะเป็นสูงสุด (0.85) ซึ่งจะถูกเลือกเป็นคำถัดไปในลำดับ

2.3 ทฤษฎีการแปลภาษาด้วยเครื่องประสาท (Neural Machine Translation หรือ NMT)

การแปลภาษาด้วยเครื่องประสาท เป็นกระบวนการแปลภาษาที่อาศัยแบบจำลองเครือข่ายประสาทเทียม ในการเรียนรู้รูปแบบและโครงสร้างของภาษาต้นทางและภาษาปลายทาง โดยการแปลภาษาด้วยเครื่องประสาทได้รับการพัฒนาขึ้นเพื่อตอบสนองข้อจำกัดของวิธีการแปลภาษาแบบเดิม เช่น การแปลด้วยกฎ (Rule-Based Machine Translation: RBMT) และการแปลด้วยสถิติ (Statistical Machine Translation: SMT) ซึ่งมีปัญหาเรื่องการพึ่งพากฎที่ซับซ้อนและความแม่นยำของคำศัพท์ จุดเด่นของการแปลภาษาด้วยเครื่องประสาทอยู่ที่การเรียนรู้เชิงลึก โดยเฉพาะในส่วนของลำดับคำ ซึ่งช่วยให้แบบจำลองสามารถเข้าใจบริบทของข้อความและความสัมพันธ์ระหว่างคำในประโยคได้อย่างมีประสิทธิภาพ นอกจากนี้การแปลภาษาด้วยเครื่องประสาท ยังมีความสามารถใน

การลดข้อผิดพลาดจากการแปลที่เกิดจากการแบ่งส่วนข้อมูลเป็นชิ้นย่อย โดยใช้วิธีการแปลงข้อความทั้งหมดในครั้งเดียว แทนที่จะประมวลผลทีละส่วนแบบการแปลด้วยสถิติ [3], [11]

โครงสร้างพื้นฐานของการแปลภาษาด้วยเครื่องประสาทมีองค์ประกอบสำคัญคือ Encoder-Decoder Framework ซึ่งประกอบด้วยสองส่วนหลัก:

- Encoder: ทำหน้าที่เข้ารหัสข้อความต้นทางให้เป็นเวกเตอร์เชิงความหมาย หรือ การเป็นตัวแทนที่สะท้อนความหมายของข้อความทั้งหมด
- Decoder: ทำหน้าที่ถอดรหัสจากเวกเตอร์เชิงความหมายที่ได้จาก Encoder ออกมาเป็นข้อความในภาษาปลายทาง

ในกระบวนการนี้ Attention Mechanism มีบทบาทสำคัญ เนื่องจากช่วยให้แบบจำลองสามารถโฟกัสไปที่คำหรือส่วนที่เกี่ยวข้องในประโยคต้นทางขณะทำการแปล ช่วยแก้ปัญหาข้อจำกัดของ Encoder-Decoder แบบดั้งเดิมที่ใช้เพียงเวกเตอร์เดียว ซึ่งอาจทำให้ข้อมูลบางส่วนสูญหาย โดย Attention Mechanism ช่วยให้น้ำหนักกับคำหรือวลีที่มีความสำคัญต่อการแปล ส่งผลให้ประสิทธิภาพของการแปลเพิ่มขึ้น เทคนิคเหล่านี้ได้สร้างรากฐานสำคัญให้กับแบบจำลองการเรียนรู้เชิงลึกทรานส์ฟอร์เมอร์ส

2.4 การหาค่าเหมาะที่สุดของค่าไฮเปอร์พารามิเตอร์

ค่าไฮเปอร์พารามิเตอร์คือค่าตัวแปรที่ใช้ปรับแต่งแบบจำลองการเรียนรู้ของเครื่อง ซึ่งเป็นองค์ประกอบสำคัญที่ช่วยเพิ่มประสิทธิภาพของแบบจำลอง ในแบบจำลองการเรียนรู้เชิงลึก การปรับแต่งค่าไฮเปอร์พารามิเตอร์มีความท้าทายมากขึ้น เนื่องจากโครงสร้างที่ซับซ้อนและจำนวนค่าที่ต้องกำหนดมีมากกว่าแบบจำลองดั้งเดิม ประเด็นสำคัญที่ต้องพิจารณาในการปรับแต่งค่าไฮเปอร์พารามิเตอร์ [4], [5] ได้แก่

2.4.1 การกำหนดปัญหาการปรับค่าไฮเปอร์พารามิเตอร์ (Hyper-parameter optimization problem definition)

การปรับแต่งค่าไฮเปอร์พารามิเตอร์เป็นกระบวนการสำคัญที่มีผลโดยตรงต่อประสิทธิภาพของอัลกอริทึมการเรียนรู้ของเครื่อง ค่าไฮเปอร์พารามิเตอร์ เช่น อัตราการเรียนรู้ (Learning Rate) และขนาดแบตช์ (Batch Size) [12] ส่งผลต่อการทำงานของแบบจำลองในรูปแบบที่ซับซ้อนและไม่สามารถกำหนดด้วยสูตรคงที่

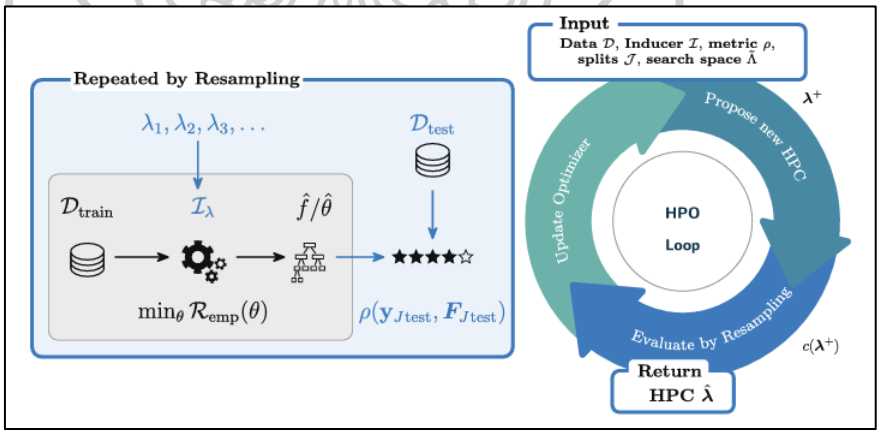
องค์ประกอบสำคัญของปัญหา คือ

1.พื้นที่การค้นหา ประกอบด้วยค่าไฮเปอร์พารามิเตอร์ที่กำลังพิจารณา ซึ่งอาจมีลักษณะเป็น ช่วงค่าต่อเนื่อง เช่น [0.001, 0.01] ของอัตราการเรียนรู้ ค่าที่ไม่ต่อเนื่อง เช่น [16, 32, 64] ของขนาดแบตช์ หมวดหมู่ เช่น RMSprop, Adam, SGD ของอัลกอริทึมการปรับน้ำหนัก บางครั้งค่าไฮเปอร์พารามิเตอร์มีความพึ่งพาซึ่งกันและกัน เช่น การปรับขนาดแบตช์อาจส่งผลต่อการกำหนดค่าอัตราการเรียนรู้ที่เหมาะสม

2.เป้าหมายของการปรับแต่ง ลดข้อผิดพลาดและเพิ่มประสิทธิภาพของอัลกอริทึมค้นหาค่าที่เหมาะสมที่สุดเพื่อให้ได้ผลลัพธ์ที่แม่นยำ เช่น ความแม่นยำในงานจำแนกรูปภาพหรือคะแนน BLEU ในงานแปลภาษา

3.ข้อจำกัด กระบวนการปรับแต่งต้องใช้ทรัพยากรในการประมวลผลสูงคือเวลาและกำลังคำนวณ โดยเฉพาะอย่างยิ่งเมื่อแบบจำลองมีขนาดใหญ่

ตัวอย่าง ในงานจำแนกรูปภาพ การปรับอัตราการเรียนรู้ให้เหมาะสมอาจช่วยให้แบบจำลองเรียนรู้ได้รวดเร็วขึ้นโดยไม่ติดปัญหาการเรียนรู้ที่ช้าหรือค่าความผิดพลาดที่เพิ่มขึ้น การปรับแต่งค่าไฮเปอร์พารามิเตอร์เป็นกระบวนการที่ท้าทาย แต่มีความสำคัญอย่างยิ่งในงานวิจัยและการพัฒนาแบบจำลองการเรียนรู้ของเครื่อง เพื่อให้ได้ผลลัพธ์ที่แม่นยำและมีประสิทธิภาพ



ภาพที่ 3 ภาพรวมการปรับแต่งค่าไฮเปอร์พารามิเตอร์ของอัลกอริทึมการเรียนรู้เครื่อง

ที่มา: บทความ [4]

2.4.2 อัลกอริทึมการปรับค่าไฮเปอร์พารามิเตอร์ที่ได้รับการยอมรับอย่างกว้างขวาง (Well-established HPO algorithms)

อัลกอริทึมการปรับค่าไฮเปอร์พารามิเตอร์ (Hyper-parameter Optimization - HPO) เป็นกระบวนการสำคัญที่ช่วยปรับแต่งค่าพารามิเตอร์เพื่อเพิ่มประสิทธิภาพของแบบจำลอง กระบวนการนี้ทำงานแบบวนซ้ำ โดยแต่ละรอบจะเสนอค่าพารามิเตอร์ใหม่ ประเมินผลการทำงาน และจัดเก็บข้อมูลใน "แฟ้มเก็บข้อมูล" เพื่อใช้อ้างอิงและปรับปรุงในรอบถัดไป [4]

2.4.2.1 การค้นหาแบบตาราง (Grid Search)

เป็นหนึ่งในวิธีการปรับค่าไฮเปอร์พารามิเตอร์ที่ง่ายและมีโครงสร้าง โดยกำหนดช่วงค่าของไฮเปอร์พารามิเตอร์แต่ละตัวและทดสอบทุกการรวมค่าที่เป็นไปได้ วิธีนี้เหมาะสำหรับพื้นที่การค้นหาขนาดเล็กที่ต้องการสำรวจอย่างครอบคลุม คุณสมบัติของการค้นหาแบบตารางมีดังนี้ [4], [5]

1. การเลือกค่าไฮเปอร์พารามิเตอร์: ค่าไฮเปอร์พารามิเตอร์ที่ต้องการปรับจะถูกกำหนดเป็นช่วงหรือรายการ ตัวอย่างเช่น อัตราการเรียนรู้: [0.01, 0.1, 1.0] ขนาดแบตช์: [16, 32, 64] สำหรับค่าที่เป็นตัวเลข จะถูกจัดวางอย่างเท่าเทียมกันในช่วงที่กำหนด เรียกว่า "ความละเอียดของตาราง" ในขณะที่ค่าที่เป็นหมวดหมู่ เช่น ฟังก์ชันเปิดปิด สามารถเลือกได้จาก ReLU, Sigmoid, Tanh

2. การประเมินผล: แบบจำลองจะถูกฝึกและประเมินผลด้วยการรวมค่าที่เป็นไปได้ของค่าไฮเปอร์พารามิเตอร์ โดยทั่วไปใช้การตรวจสอบแบบข้าม (Cross-Validation) เพื่อหลีกเลี่ยงปัญหา Overfitting และให้ผลการประเมินที่แม่นยำ

3. ความละเอียดของตาราง: หมายถึงจำนวนค่าที่แตกต่างกันที่จะทดสอบต่อค่าไฮเปอร์พารามิเตอร์หนึ่งตัว ความละเอียดที่สูงขึ้นช่วยครอบคลุมพื้นที่การค้นหาได้ดีขึ้น แต่ก็เพิ่มเวลาและทรัพยากรที่ใช้ เช่น ความละเอียดต่ำ Learning Rate = [0.01, 0.1] ความละเอียดสูง Learning Rate = [0.01, 0.05, 0.1, 0.5]

4. การปรับแต่งตามหมวดหมู่ สำหรับค่าไฮเปอร์พารามิเตอร์ที่เป็นหมวดหมู่ เช่น ประเภทของแบบจำลอง (Model Type) หรือฟังก์ชันเปิดปิด (Activation Function) สามารถพิจารณาทดสอบทุกค่าหรือเลือกเฉพาะค่าที่สนใจ

ตัวอย่าง สมมติว่าต้องการปรับอัตราการเรียนรู้ [0.01,0.1] และขนาดของแบตช์ [32,64] การค้นหาแบบตารางจะประเมินผลค่าได้เป็นไปได้ทั้งหมด คือ [(0.01,32),(0.01,64),(0.1,32),(0.1,64)]

2.4.2.2 การค้นหาแบบสุ่ม (Random Search)

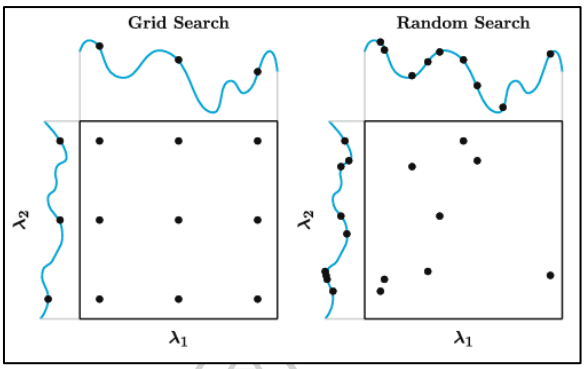
เป็นวิธีการปรับค่าไฮเปอร์พารามิเตอร์โดยการสุ่มเลือกค่าจากการแจกแจงที่กำหนดไว้ล่วงหน้า เช่น การแจกแจงแบบเท่ากัน (Uniform Distribution) วิธีนี้สามารถใช้ได้กับพารามิเตอร์ที่เป็นตัวเลข, จำนวนเต็ม หรือหมวดหมู่ และเหมาะสำหรับกรณีที่มีพื้นที่การค้นหาขนาดใหญ่ [4], [5]

คุณสมบัติของการค้นหาแบบสุ่มมีดังนี้

1. ความเรียบง่ายและความเร็ว: การค้นหาแบบสุ่มไม่จำเป็นต้องทดลองทุกการรวมค่าพารามิเตอร์ที่เป็นไปได้เหมือนในการค้นหาแบบตารางทำให้ประหยัดเวลาและทรัพยากร โดยเฉพาะในกรณีที่พื้นที่การค้นหามีขนาดใหญ่หรือมีหลายมิติ
2. การจัดการกับมิติคำสั่ง: การค้นหาแบบสุ่มสามารถจัดการกับพื้นที่การค้นหาที่มีจำนวนพารามิเตอร์มากได้ดีกว่าการค้นหาแบบตาราง เนื่องจากมีโอกาสสุ่มค่าไปยังบริเวณที่อาจเป็นค่าที่เหมาะสมที่สุด (Optimal Values)
3. การตั้งค่าการแจกแจง: การสุ่มค่าพารามิเตอร์ส่วนใหญ่ใช้การแจกแจงแบบเท่ากัน ซึ่งทุกค่ามีโอกาสถูกเลือกเท่ากัน อย่างไรก็ตาม สามารถเลือกการแจกแจงแบบอื่นได้ตามความเหมาะสม เช่น การแจกแจงแบบปกติ (Normal Distribution) เหมาะสำหรับค่าที่อยู่ใกล้ศูนย์กลาง การแจกแจงแบบลำดับกำลัง (Logarithmic Distribution) เหมาะสำหรับค่าที่มีช่วงกว้าง เช่น อัตราการเรียนรู้
4. การใช้กับหมวดหมู่: การค้นหาแบบสุ่มยังสามารถใช้กับพารามิเตอร์ที่เป็นหมวดหมู่ เช่น การสุ่มเลือกฟังก์ชันเปิดปิด จาก ReLU, Sigmoid, Tanh หรือประเภทของอัลกอริทึมจาก SGD, Adam, RMSprop

ตัวอย่าง สมมติว่าต้องการปรับอัตราการเรียนรู้ [0.001,0.01,0.1] และขนาดของแบตช์ [16,32,64] การค้นหาสุ่มจะสุ่มค่าต่างๆ เช่น [(0.01,32),(0.1,16),(0.001,64)] ซึ่งลด

จำนวนการประเมินลงเมื่อเทียบกับการทดสอบทุกการรวมค่าที่เป็นไปได้ในการค้นหาแบบตาราง



ภาพที่ 4 การค้นหาแบบตาราง และ การค้นหาแบบสุ่ม

ที่มา: บทความ [4]

2.4.2.3 การปรับแต่งโดยใช้เกรเดียนต์ (Gradient-based optimization)

การปรับแต่งโดยใช้เทคนิคลดความชัน (Gradient Descent) เป็นวิธีดั้งเดิมที่คำนวณค่าความชัน เพื่อหาทิศทางที่ลดค่าฟังก์ชันวัตถุประสงค์ได้ดีที่สุด เทคนิคนี้เคลื่อนที่ในทิศทางตรงข้ามกับความชันเพื่อไปยังจุดที่เหมาะสมที่สุด ตัวอย่างเช่น การฝึกแบบจำลองเครือข่ายประสาทเทียมใช้เทคนิคลดความชัน เพื่อปรับน้ำหนัก (Weights) และอคติ (Biases) ของเครือข่าย [4]

กระบวนการมีดังนี้

1. เริ่มต้นด้วยค่าพารามิเตอร์สุ่ม
2. คำนวณค่าความชันของฟังก์ชันวัตถุประสงค์
3. ปรับค่าพารามิเตอร์ในทิศทางตรงข้ามกับความชัน
4. ทำซ้ำจนกว่าค่าฟังก์ชันวัตถุประสงค์จะลู่เข้าสู่ค่าที่เหมาะสมที่สุด

2.4.2.4 การปรับแต่งพารามิเตอร์แบบเบย์เซียน (Bayesian Optimization หรือ BO)

เป็นเทคนิคการปรับแต่งค่าไฮเปอร์พารามิเตอร์ที่เหมาะสำหรับฟังก์ชันกล่องดำ โดยเฉพาะในกรณีที่การประเมินผลฟังก์ชันมีค่าใช้จ่ายสูง เช่น การฝึกแบบจำลองการเรียนรู้

เชิงลึก เทคนิคนี้ใช้ แบบจำลองทดแทน (Surrogate Model) เพื่อประมาณค่าความสัมพันธ์ระหว่างพารามิเตอร์และประสิทธิภาพของแบบจำลอง พร้อมกับฟังก์ชันการได้มา (Acquisition Function) เพื่อเลือกค่าพารามิเตอร์ที่ควรทดลองถัดไป [4]

กระบวนการทำงานมีดังนี้

1. สุ่มตัวอย่างเริ่มต้น: เริ่มต้นด้วยการสุ่มค่าไฮเปอร์พารามิเตอร์จากพื้นที่การค้นหาเพื่อสร้างฐานข้อมูลเริ่มต้น (Initial Dataset)
2. สร้างแบบจำลองทดแทน: ใช้กระบวนการเกาส์เซียน (Gaussian Process) หรือการสุ่มป่าไม้ (Random Forests) เพื่อสร้างแบบจำลองทดแทนที่สามารถให้ค่าประมาณการประสิทธิภาพและค่าความไม่แน่นอนของการทำนาย
3. ฟังก์ชันการได้มา: ใช้ฟังก์ชันการได้มา เช่น Expected Improvement (EI) หรือ Upper Confidence Bound (UCB) เพื่อเลือกค่าพารามิเตอร์ถัดไป โดยฟังก์ชันนี้สมดุลระหว่าง การใช้ข้อมูลที่มีอยู่ และ การสำรวจข้อมูลใหม่
4. ประเมินค่าจริง: ทดสอบค่าพารามิเตอร์ที่เลือกและเพิ่มผลลัพธ์เข้าไปในฐานข้อมูล จากนั้นปรับปรุงแบบจำลองทดแทนให้แม่นยำยิ่งขึ้น
5. วนซ้ำ: กระบวนการนี้ดำเนินต่อไปจนกว่าจะครบจำนวนรอบที่กำหนดหรือถึงเกณฑ์การสิ้นสุด เช่น เมื่อค่าประสิทธิภาพไม่เปลี่ยนแปลงอย่างมีนัยสำคัญ

2.4.2.5 แนวคิดของการปรับแต่งค่าไฮเปอร์พารามิเตอร์แบบมัลติฟิเดลิตี้ (Multifidelity-MF)

เป็นเทคนิคที่ใช้ในงานที่ต้องการประมาณผลลัพธ์จากฟังก์ชันกล่องดำ เช่น การฝึกแบบจำลองการเรียนรู้ของเครื่องขนาดใหญ่หรือการจำลองเชิงวิศวกรรม เทคนิคนี้ช่วยลดต้นทุนการประเมินผลโดยปรับระดับความแม่นยำหรือคุณภาพของการประมาณการตามขั้นตอนของกระบวนการ [4]

กลไกของการปรับแต่งแบบมัลติฟิเดลิตี้มีดังนี้

1. พารามิเตอร์ฟิเดลิตี้ (λ_{fid}): พารามิเตอร์นี้กำหนดระดับความแม่นยำหรือคุณภาพของการประมาณการ
2. การใช้งานในกระบวนการปรับแต่ง:

- การสำรวจ: ในระยะแรก ใช้ค่า λ_{fid} ต่ำเพื่อสำรวจพื้นที่การค้นหาอย่างกว้างขวาง โดยใช้ทรัพยากรต่ำ เช่น การฝึกแบบจำลองด้วยข้อมูลย่อยหรือจำกัดจำนวน epochs
- การลงลึก: หลังจากระบุพื้นที่ที่มีแนวโน้มดีแล้ว ค่า λ_{fid} จะถูกเพิ่มเพื่อใช้แบบจำลองที่แม่นยำขึ้นสำหรับการค้นหาค่าที่เหมาะสมที่สุด

3. การป้องกัน Overfitting และการจัดการต้นทุน: การใช้ค่า λ_{fid} ที่สูงตลอดเวลาอาจทำให้เกิด Overfitting หรือใช้ทรัพยากรมากเกินไป กลยุทธ์มัลติฟิเดลิตี้ช่วยลดต้นทุนโดยการประเมินผลแบบลำดับขั้น

ตัวอย่าง สมมติต้องการปรับค่าไฮเปอร์พารามิเตอร์ 2 ตัว ได้แก่อัตราการเรียนรู้ และขนาดของแบตช์

ขั้นตอนที่ 1: ใช้ค่า λ_{fid} ต่ำโดยการสุ่มค่าพารามิเตอร์และฝึกแบบจำลองด้วยข้อมูลย่อยและจำกัดจำนวน epochs (เช่น 10 epochs)

ขั้นตอนที่ 2: เลือกค่าพารามิเตอร์ที่มีแนวโน้มดีจากขั้นแรก และเพิ่มค่า λ_{fid} โดยใช้ข้อมูลเต็มและเพิ่มจำนวน epochs (เช่น 50 epochs) เพื่อปรับแต่งแบบละเอียด

2.4.2.6 Hyperband: การปรับแต่งไฮเปอร์พารามิเตอร์อย่างมีประสิทธิภาพ

เป็นวิธีการปรับแต่งค่าไฮเปอร์พารามิเตอร์ที่พัฒนาขึ้นจากแนวคิดของ Successive Halving (SH) โดยเน้นการจัดสรรงบประมาณฟิเดลิตี้ (λ_{fid}) อย่างมีประสิทธิภาพ Hyperband เหมาะสำหรับการค้นหาไฮเปอร์พารามิเตอร์ในแบบจำลองที่มีพื้นที่ค้นหาขนาดใหญ่และทรัพยากรจำกัด [4], [5]

หลักการทำงานมีดังนี้

1. การเริ่มต้นกระบวนการ: เริ่มต้นด้วยการสุ่มจำนวนผู้สมัคร (แบบจำลองตั้งต้น) จากพื้นที่ค้นหา ตัวอย่างเช่น หากต้องการปรับอัตราการเรียนรู้และขนาดแบตช์อาจเริ่มต้นด้วย 27 แบบจำลอง (n = 27)
2. กระบวนการ Successive Halving (SH):
   - การประเมินผลในรอบแรก: ทุกแบบจำลองจะถูกประเมินด้วยงบประมาณฟิเดลิตี้ที่ต่ำ (λ_{fid} เริ่มต้น เช่น 10 epochs)

- การคัดเลือก: เพียง 1/ηHB ของแบบจำลองที่ทำผลงานดีที่สุดจะถูกเลือกเข้ารอบถัดไป โดยที่ ηHB (มักเป็น 2 หรือ 3) คือตัวคูณที่ควบคุมการลดจำนวนแบบจำลอง
- การเพิ่มงบประมาณฟิเดลิตี้: แบบจำลองที่ผ่านเข้ารอบจะได้รับการประเมินด้วยงบประมาณฟิเดลิตี้ที่เพิ่มขึ้น (ηHB×λfid) เช่น 20 epochs ในรอบถัดไป

3. การวนซ้ำจนเหลือผู้สมัครคนสุดท้าย: กระบวนการ Successive Halving จะดำเนินต่อไปจนกว่าจะเหลือแบบจำลองเพียงตัวเดียวที่มีประสิทธิภาพดีที่สุด

ตัวอย่าง สมมติว่าต้องการปรับค่าไฮเปอร์พารามิเตอร์ 3 ตัว ได้แก่ Learning Rate, Batch Size, และ Dropout จำนวนแบบจำลองตั้งต้น n=27 งบประมาณฟิเดลิตี้เริ่มต้น λfid = 10 (epochs) ตัวคูณ ηHB= 3

- **รอบแรก:**
  - ทดสอบแบบจำลองทั้ง 27 ตัวด้วย 10 epochs
  - คัดเลือก 27/3= 9 แบบจำลองที่ดีที่สุด
- **รอบสอง:**
  - ทดสอบแบบจำลอง 9 ตัวที่เหลือด้วย 30 epochs
  - คัดเลือก 9/3= 3 แบบจำลองที่ดีที่สุด
- **รอบสุดท้าย:**
  - ทดสอบแบบจำลอง 3 ตัวด้วย 90 epochs
  - เลือกแบบจำลองที่ดีที่สุดจากการประเมิน

2.4.2.7 Multifidelity Bayesian Optimization (MFBO): การปรับแต่งแบบมีหลายระดับด้วย Bayesian Optimization

เป็นการผสมผสานระหว่างแนวคิดของ Hyperband และการปรับแต่งพารามิเตอร์แบบเบย์เซียน เพื่อเพิ่มประสิทธิภาพในการค้นหาค่าที่เหมาะสมที่สุดในงานปรับแต่งแบบมีหลายระดับ Hyperband เน้นการกำจัดพารามิเตอร์ที่มีประสิทธิภาพต่ำตั้งแต่เริ่มต้น ขณะที่การปรับแต่งพารามิเตอร์แบบเบย์เซียน ใช้แบบจำลองทดแทนเพื่อเสนอค่าพารามิเตอร์อย่างชาญฉลาด โดย MFBO รวมข้อดีของทั้งสองวิธีเพื่อปรับปรุงกระบวนการค้นหา กระบวนการทำงานมีดังนี้ [4], [5]

1. การกำหนดระดับฟิเดลิตี้ (λfid): ใช้ควบคุมความแม่นยำและต้นทุนของการประเมินผล ตัวอย่างเช่น
   - λfid ต่ำ: ใช้ข้อมูลย่อยหรือจำกัดจำนวนรอบการฝึกเพื่อประหยัดทรัพยากร
   - λfid สูง: ใช้ข้อมูลเต็มและเพิ่มจำนวนรอบการฝึกเพื่อความแม่นยำ
2. การกำจัดพารามิเตอร์ที่มีประสิทธิภาพต่ำ: Hyperband ใช้ตัวควบคุม ηHB เพื่อลดจำนวนพารามิเตอร์ในแต่ละรอบ เช่น กำจัด 1/ηHB ของพารามิเตอร์ที่มีประสิทธิภาพต่ำ
3. การใช้แบบจำลองทดแทนของการปรับแต่งพารามิเตอร์แบบเบย์เซียน: ใช้กระบวนการเกาส์เซียนหรือการสุ่มป่าไม้ เพื่อสร้างแบบจำลองทดแทนที่สามารถประเมินประสิทธิภาพของพารามิเตอร์และตัดสินใจเกี่ยวกับ ระดับฟิเดลิตี้ (λfid) ที่จะใช้ในรอบถัดไปและพารามิเตอร์ที่ควรทิ้งหรือสำรวจเพิ่มเติม
4. การตัดสินใจอย่างชาญฉลาด: ฟังก์ชันการได้มาเช่น Expected Improvement (EI) ถูกใช้เพื่อสมดุลระหว่างการใช้ข้อมูลเดิมและการสำรวจข้อมูลใหม่

ตัวอย่าง สมมติว่าต้องการปรับค่าไฮเปอร์พารามิเตอร์ 2 ตัว ได้แก่อัตราการเรียนรู้และขนาดของแบตช์

1. ใช้ λfid ต่ำ (ข้อมูลย่อยและ 10 Epochs) กับพารามิเตอร์ 20 ชุด
2. การปรับแต่งพารามิเตอร์แบบเบย์เซียนใช้แบบจำลองทดแทนเพื่อเลือกพารามิเตอร์ 5 ชุดที่ดีที่สุด พร้อมเพิ่ม λfid เป็นข้อมูลเต็มและ 50 Epochs
3. ประเมินพารามิเตอร์ที่เหลือและเลือกค่าที่เหมาะสมที่สุด

2.4.2.8 กระบวนการ Iterated Racing (IR): การปรับแต่งไฮเปอร์พารามิเตอร์ด้วยแนวคิดการแข่งขัน

เป็นวิธีการปรับแต่งพารามิเตอร์ที่เน้นการประเมินและกำจัดค่าพารามิเตอร์ที่ไม่มีประสิทธิภาพอย่างรวดเร็วในระยะเริ่มต้น โดยใช้แนวคิด "การแข่งขัน" เพื่อโฟกัสไปยังพื้นที่การค้นหาที่น่าสนใจมากขึ้น กระบวนการนี้ไม่จำกัดเฉพาะอัลกอริทึมการเรียนรู้ของเครื่องแต่ยังเหมาะสำหรับการค้นหาค่าที่เหมาะสมที่สุดในปัญหาอื่น ๆ ด้วย [4], [5]

กระบวนการทำงานมีดังนี้

1. การเริ่มต้นการแข่งขัน: การประเมินพารามิเตอร์การกำหนดค่า (HPCs) ถูกสุ่มเลือกจากพื้นที่การค้นหา แต่ละ HPC ถูกประเมินด้วยการสุ่มตัวอย่างซ้ำ เช่น การตรวจสอบแบบข้าม
2. การตัดสินใจทิ้ง HPCs: ใช้การทดสอบทางสถิติเพื่อเปรียบเทียบค่าประสิทธิภาพ HPCs ที่ไม่มีแนวโน้มจะปรับปรุงประสิทธิภาพถูกกำจัดออก เพื่อประหยัดทรัพยากร
3. การแข่งขันหลายรอบ: กลุ่ม HPCs ที่รอดจากการแข่งขันรอบแรกจะถูกนำไปใช้ในการแข่งขันรอบถัดไป HPC ใหม่ถูกสร้างขึ้นโดยใช้กลไกการกลายพันธุ์ เช่น การปรับค่าพารามิเตอร์ตัวเลขผ่านการแจกแจงปกติที่ตัดทอนแล้ว
4. การจัดการพื้นที่การค้นหา: IR สามารถจัดการกับพื้นที่การค้นหาที่มีการพึ่งพากันได้ ใช้หลักการของอัลกอริทึมการประมาณการการแจกแจง เพื่อจัดการความไม่แน่นอน

ตัวอย่าง สมมติว่าต้องการปรับค่าไฮเปอร์พารามิเตอร์ 2 ตัวในแบบจำลองการเรียนรู้ของเครื่อง ได้แก่อัตราการเรียนรู้และขนาดของแบตช์

- รอบแรก: สุ่ม HPCs 20 ค่า และประเมินผลโดยใช้การตรวจสอบแบบข้าม
- การคัดเลือก: ใช้การทดสอบสถิติ เลือก HPCs 10 ค่าแรกที่มีประสิทธิภาพดีที่สุด
- รอบสอง: สร้าง HPC ใหม่จากกลุ่มที่รอดผ่านกระบวนการกลายพันธุ์ เช่น การสุ่มค่าอัตราการเรียนรู้ใกล้กับค่าที่ดีที่สุดในรอบแรก

2.4.2.9 กลยุทธ์วิวัฒนาการ (Evolution Strategies หรือ ES): การปรับแต่งพารามิเตอร์ด้วยแรงบันดาลใจจากชีววิทยา

เป็นเทคนิคการปรับแต่งค่าไฮเปอร์พารามิเตอร์ที่ได้รับแรงบันดาลใจจากกระบวนการวิวัฒนาการทางชีววิทยา กลยุทธ์วิวัฒนาการไม่ต้องการข้อมูลไล่ระดับ (Gradient-Free) ทำให้เหมาะสำหรับการปรับแต่งในสภาวะกล่องดำ เช่น การปรับค่าพารามิเตอร์ในฟังก์ชันที่ซับซ้อนหรือไม่มีสูตรอนุพันธ์ชัดเจน [4], [5]

- ตัวแทน (Individual): การกำหนดค่าพารามิเตอร์หนึ่งชุด
- ประชากร (Population): ชุดของตัวแทนที่ได้รับการประเมิน

- ความเหมาะสม (Fitness): ค่าผกผันของข้อผิดพลาด เช่น ค่าความแม่นยำหรือค่าข้อผิดพลาดทั่วไป

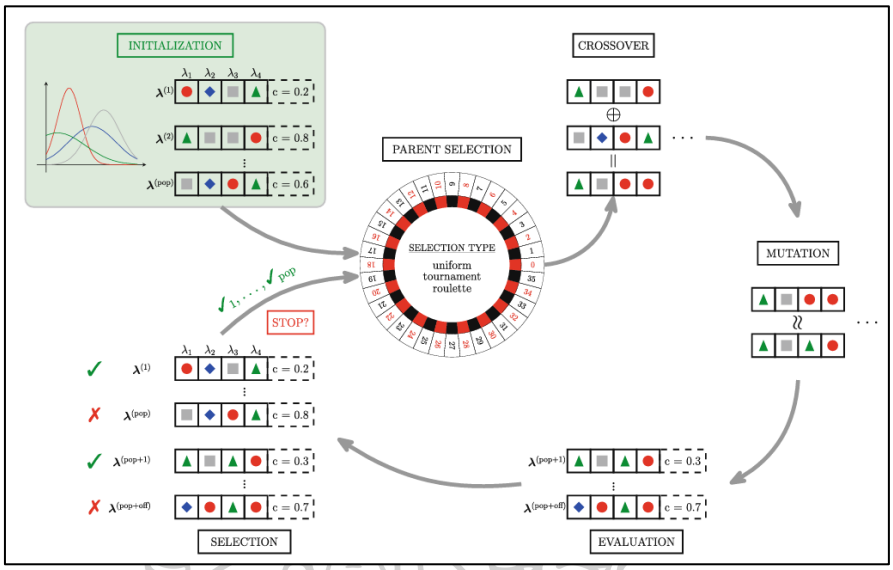
กระบวนการทำงานมีดังนี้

1. สุ่มตัวอย่างประชากรเริ่มต้น: สุ่มค่าพารามิเตอร์ตั้งต้นสำหรับประชากรเริ่มต้น เช่น 20 ตัวแทน
2. ประเมินความเหมาะสม: ประเมินตัวแทนแต่ละตัวในประชากรโดยคำนวณค่าความเหมาะสม เช่น ค่าผลลัพธ์ของฟังก์ชันเป้าหมาย
3. เลือกพ่อแม่พันธุ์: เลือกตัวแทนที่มีค่าความเหมาะสมสูงสุดเป็นพ่อแม่พันธุ์ เช่น เลือก 10 ตัวจากประชากร 20 ตัว
4. เพิ่มประชากรโดยการกลายพันธุ์และผสมพันธุ์:
    - การกลายพันธุ์: เปลี่ยนแปลงค่าพารามิเตอร์บางตัวแบบสุ่มโดยใช้การแจกแจง เช่น การแจกแจงปกติ
    - การผสมพันธุ์: รวมค่าพารามิเตอร์จากพ่อแม่พันธุ์สองตัวเพื่อสร้างลูกหลานใหม่
5. ประเมินลูกหลาน: ประเมินค่าความเหมาะสมของลูกหลานที่เกิดจากการกลายพันธุ์และผสมพันธุ์
6. เลือกตัวแทนที่เหมาะสมที่สุด: รวมพ่อแม่พันธุ์และลูกหลาน เลือกตัวแทนที่มีค่าความเหมาะสมสูงสุดสำหรับประชากรรอบถัดไป
7. ทำซ้ำจนถึงเงื่อนไขสิ้นสุด: กระบวนการทำงานนี้จะวนซ้ำจนกว่าจะถึงเกณฑ์สิ้นสุด เช่น จำนวนรอบที่กำหนดหรือความแม่นยำที่ต้องการ

ตัวอย่าง สมมติว่าต้องการปรับค่าพารามิเตอร์ 2 ตัว ได้แก่ อัตราการเรียนรู้และ Dropout

- ประชากรเริ่มต้น สุ่มอัตราการเรียนรู้จากช่วง [0.001,0.1] สุ่ม Dropout จากช่วง [0.2,0.5]
- การกลายพันธุ์ เปลี่ยนอัตราการเรียนรู้จาก 0.05 เป็น 0.06 ลด Dropout จาก 0.3 เป็น 0.25

- การผสมพันธุ์ รวมอัตราการเรียนรู้ จากพ่อแม่ 2 ตัวเพื่อสร้างค่ากลาง เช่น (0.04+0.06)/2 = 0.05



ภาพที่ 5 กระบวนการทำงานของ Evolution Strategy

ที่มา: บทความ [4]

2.4.2.10 อัลกอริธึมพันธุกรรม (Genetic Algorithm หรือ GA): การปรับแต่งไฮเปอร์พารามิเตอร์ด้วยแนวคิดวิวัฒนาการ

เป็นเทคนิคที่ได้รับแรงบันดาลใจจากทฤษฎีวิวัฒนาการ โดยอาศัยกระบวนการทางชีววิทยา เช่น การเลือก การข้าม และการกลายพันธุ์ เพื่อค้นหาค่าที่เหมาะสมที่สุดในพื้นที่การค้นหา [4], [5]

- โครโมโซม (Chromosome): แทนค่าพารามิเตอร์ไฮเปอร์หนึ่งชุด
- ยีน (Gene): แทนค่าพารามิเตอร์ย่อย เช่น ค่าทศนิยมของพารามิเตอร์
- ประชากร (Population): กลุ่มของโครโมโซมที่ได้รับการประเมิน
- ฟังก์ชันความเหมาะสม (Fitness Function): ชี้วัดความเหมาะสมของโครโมโซมในบริบทของฟังก์ชันวัตถุประสงค์

กระบวนการทำงานมีดังนี้

1. การเริ่มต้น (Initialization): สุ่มสร้างประชากรเริ่มต้น โดยกำหนดค่าของโครโมโซมและยีน เช่น อัตราการเรียนรู้ และ ขนาดของแบตช์
2. การประเมินความเหมาะสม (Fitness Evaluation): คำนวณฟังก์ชันความเหมาะสมของแต่ละโครโมโซม เช่น ค่าความแม่นยำ หรือค่าความสูญเสียของแบบจำลอง
3. การเลือก (Selection): เลือกโครโมโซมที่มีค่าความเหมาะสมสูงสุดเป็นพ่อแม่พันธุ์ โดยใช้วิธีเช่นการสุ่มแบบวงล้อรูเล็ต หรือ การคัดเลือกแบบจัดการแข่งขัน
4. การข้ามและการกลายพันธุ์ (Crossover and Mutation):
    - การข้าม: สร้างลูกหลานใหม่โดยผสมค่าจากพ่อแม่ เช่น ค่าอัตราการเรียนรู้จากพ่อแม่คนที่ 1 และขนาดของแบตช์จากพ่อแม่คนที่ 2
    - การกลายพันธุ์: เปลี่ยนแปลงค่าของยีนบางตัวแบบสุ่ม เช่น เปลี่ยน Dropout จาก 0.3 เป็น 0.4
5. การสร้างรุ่นถัดไป (Next Generation): รวมพ่อแม่พันธุ์และลูกหลาน เลือกโครโมโซมที่มีค่าความเหมาะสมสูงสุดเพื่อสร้างประชากรใหม่
6. การทำซ้ำ (Iteration): ทำซ้ำขั้นตอนการเลือก การข้าม และการกลายพันธุ์ จนกว่าจะถึงเงื่อนไขการสิ้นสุด เช่น จำนวนรอบที่กำหนดหรือค่าความแม่นยำที่ต้องการ

ตัวอย่าง สมมติว่าต้องการปรับค่าไฮเปอร์พารามิเตอร์ 3 ตัว ได้แก่ อัตราการเรียนรู้,ขนาดของแบตช์ และ Dropout

1. สุ่มค่าอัตราการเรียนรู้จากช่วง [0.01, 0.1], ขนาดของแบตช์จาก [16, 64], และ Dropout จาก [0.2, 0.5]
2. ประเมินฟังก์ชันความเหมาะสม เช่น ความแม่นยำของแบบจำลองที่ได้จากค่าพารามิเตอร์นั้น
3. สร้างลูกหลานโดยใช้การข้าม เช่น ค่าอัตราการเรียนรู้ = 0.05 จากพ่อแม่คนที่ 1 และขนาดของแบตช์ = 32 จากพ่อแม่คนที่ 2
4. ทำซ้ำกระบวนการจนกว่าจะพบค่าพารามิเตอร์ที่เหมาะสมที่สุด

2.4.2.11 อัลกอริธึมการปรับแต่งแบบกลุ่มอนุภาค (Particle Swarm Optimization หรือ PSO)

เป็นอัลกอริธึมวิวัฒนาการที่ได้รับแรงบันดาลใจจากพฤติกรรมของกลุ่มสิ่งมีชีวิต เช่น ฝูงนกหรือฝูงปลา อัลกอริธึมนี้ทำงานโดยใช้กลุ่มของอนุภาคที่เคลื่อนที่ในพื้นที่ค้นหา โดยแต่ละอนุภาคจะเรียนรู้และปรับปรุงตำแหน่งของตนผ่านข้อมูลที่แบ่งปันระหว่างกัน [4], [5]

- อนุภาค (Particle): แทนค่าพารามิเตอร์หนึ่งชุดในพื้นที่ค้นหา
- ตำแหน่ง (Position): ค่าพารามิเตอร์ปัจจุบันของอนุภาค
- ความเร็ว (Velocity): เวกเตอร์ที่กำหนดทิศทางการเคลื่อนที่ของอนุภาค
- ตำแหน่งที่ดีที่สุดส่วนตัว (Personal Best): ตำแหน่งที่ดีที่สุดที่อนุภาคเคยพบ
- ตำแหน่งที่ดีที่สุดของกลุ่ม (Global Best): ตำแหน่งที่ดีที่สุดที่พบในกลุ่มทั้งหมด

กระบวนการทำงานมีดังนี้

1. การเริ่มต้น: สุ่มตำแหน่งและความเร็วของอนุภาคแต่ละตัวในพื้นที่ค้นหา
2. การประเมินความเหมาะสม: ใช้ฟังก์ชันความเหมาะสม เพื่อประเมินตำแหน่งของแต่ละอนุภาค เช่น ค่าความแม่นยำหรือค่าความสูญเสีย
3. การอัปเดตตำแหน่งที่ดีที่สุด: อัปเดตตำแหน่งที่ดีที่สุดส่วนตัวสำหรับแต่ละอนุภาคและตำแหน่งที่ดีที่สุดของกลุ่ม สำหรับกลุ่มทั้งหมด
4. การอัปเดตความเร็วและตำแหน่ง: คำนวณความเร็วใหม่ และ อัปเดตตำแหน่ง
5. การวนซ้ำ: ทำซ้ำกระบวนการจนกว่าจะถึงเงื่อนไขการสิ้นสุด เช่น จำนวนรอบที่กำหนดหรือค่าความเหมาะสมที่ต้องการ

ตัวอย่าง สมมติว่าต้องการปรับค่าพารามิเตอร์ อัตราการเรียนรู้และขนาดของแบตช์

1. ประชากรเริ่มต้น: สุ่มค่าอัตราการเรียนรู้จากช่วง [0.01, 0.1] สุ่มค่าขนาดของแบตช์จากช่วง [16, 128]
2. การอัปเดตตำแหน่ง: อัตราการเรียนรู้อัปเดตจาก 0.05 เป็น 0.06 ตามความเร็วและทิศทางที่เหมาะสม ขนาดของแบตช์เปลี่ยนจาก 32 เป็น 64
3. การประเมิน: ใช้ค่าความแม่นยำของแบบจำลองเป็นตัวชี้วัดความเหมาะสม

2.4.3 การปรับแต่งค่าไฮเปอร์พารามิเตอร์จากมุมมองการอนุมานแบบสองระดับ (Bilevel inference perspective)

เป็นกระบวนการปรับแต่งค่าไฮเปอร์พารามิเตอร์ (HPO) ที่เน้นการลดความเสี่ยงของแบบจำลอง โดยคำนวณข้อผิดพลาดทั่วไป เมื่อทดสอบบนข้อมูลการตรวจสอบความถูกต้อง [4]

กระบวนการนี้แบ่งออกเป็นสองระดับ

1. ระดับแรก (Inner Level): ปรับพารามิเตอร์แบบจำลอง ($b\theta$) เพื่อหาค่าที่เหมาะสมที่สุดสำหรับการแจกแจงข้อมูลที่กำหนด
2. ระดับที่สอง (Outer Level): ปรับค่าไฮเปอร์พารามิเตอร์ ($b\lambda$) เช่น อัตราการเรียนรู้หรือ Regularization Parameter เพื่อให้ค่าพารามิเตอร์ ($b\theta$) ในระดับแรกเหมาะสมที่สุด

2.5 ชุดข้อมูลภาษาไทย - ภาษาอังกฤษ

จากงานบทความ “scb-mt-en-th-2020: A large english-thai parallel corpus” [13] ชุดข้อมูลคู่ขนานภาษาอังกฤษ-ภาษาไทยที่ได้รับการรวบรวมจากแหล่งข้อมูลหลากหลาย เช่น ข่าว บทความ วิกิพีเดีย บทสนทนาและเอกสารของรัฐบาล โดยมีทั้งหมด 1,001,752 คู่ประโยค ซึ่งเป็นคู่ประโยคในภาษาอังกฤษและภาษาไทยที่ได้รับการจัดสอดคล้องเพื่อให้สามารถแปลได้ระหว่างทั้งสองภาษา โดยเราได้มุ่งเน้นเลือกใช้ในส่วนของชุดข้อมูลสาธารณะกลุ่มภาษาอังกฤษ ต่อไปนี้สำหรับงานการประมวลผลภาษาธรรมชาติ และงานความเข้าใจภาษาธรรมชาติ เป็นกลุ่มแหล่งที่มาชุดข้อมูลเหล่านี้ได้รับการแปลเป็นภาษาไทยโดยนักแปลมืออาชีพและจากมวลชน ซึ่งมีขนาด 222,733 คู่ประโยค มีรายละเอียดดังนี้

1. Taskmaster-1 เป็นชุดข้อมูลที่มีบทสนทนาตามงาน 13,215 รายการใน 6 โดเมนที่แตกต่างกันโดเมน ได้แก่ การสั่งซื้อพิซซ่า นัดหมายซ่อมรถยนต์ กำหนดการนั่ง การสั่งซื้อตั๋วภาพยนตร์ สั่งเครื่องดื่มกาแฟ และการจองร้านอาหาร
2. The National University of Singapore เป็นข้อความ SMS 67,093 ข้อความที่เขียนโดยชาวสิงคโปร์
3. Mozilla Common Voice เป็นการบันทึกเสียง 61,584 รายการในภาษาต่าง ๆ ประกอบด้วยทั้งส่วนภาษาอังกฤษที่เขียนและพูด
4. Microsoft Research Paraphrase Identification Corpus ประกอบด้วยคู่กลุ่มภาษาอังกฤษ 5,801 คู่จากแหล่งข่าว

2.6 การหาค่าเหมาะที่สุดของแบบจำลองการเรียนรู้เชิงลึกที่ใช้ทรัพยากรต่ำ

ในการแปลภาษามีประเด็นที่สำคัญในงานวิจัยปัจจุบันสองประเด็นที่น่าสนใจ คือการใช้งานตัวแปลภาษาเชิงประสาทกับภาษาที่มีชุดข้อมูลสอนจำนวนจำกัดและการสอนแบบจำลองเพื่อให้แบบจำลองมีขนาดเล็กและใช้ทรัพยากรต่ำ [3]

สำหรับประเด็นที่เกี่ยวข้องกับตัวแปลภาษาเชิงประสาทกับภาษาที่มีชุดข้อมูลสอนจำนวนจำกัดนั้นกำลังเป็นที่สนใจ เนื่องจากในการสอนแบบจำลองการเรียนรู้เชิงลึกต้องใช้ข้อมูลจำนวนมาก แต่ในการสร้างตัวแปลภาษาเชิงประสาทจำเป็นต้องมีการสร้างคู่ของประโยคในภาษาต้นทางและปลายทาง ซึ่งเราเรียกว่าตัวอย่าง ในภาษาหลักของโลกเช่นภาษาจีน ภาษาอังกฤษ ภาษาเยอรมัน ภาษาฝรั่งเศส ภาษาสเปน เนื่องจากมีจำนวนผู้ใช้งานภาษาเหล่านี้มากในโลก ทำให้สามารถสร้างตัวอย่างเป็นจำนวนหลายร้อยล้านตัวอย่าง ทำให้มีทรัพยากรที่ใช้สอนจำนวนมาก แต่ในบางภาษาที่มีผู้ใช้งานจำกัด การสร้างตัวอย่างหลายสิบล้านตัวอย่างทำได้ยาก ดังนั้นการที่จะทำให้แบบจำลองเรียนรู้ได้ดีกับตัวอย่างจำนวนจำกัดในระดับหลักแสนตัวอย่างหรือน้อยกว่า จึงเป็นประเด็นที่ท้าทาย อีกทั้งเมื่อพิจารณาการเรียนรู้ การพูดภาษาของมนุษย์ เด็กจะมีการเรียนรู้ภาษาที่จำกัดกว่าผู้ใหญ่ ทั้งในแง่จำนวนคำศัพท์และรูปแบบประโยค แต่เด็กก็ยังสามารถสื่อสารโดยใช้ภาษากับผู้ใหญ่ได้ ดังนั้นเมื่อพิจารณาตัวอย่างที่สอนเพื่อให้ตัวแปลภาษามีความสามารถเทียบกับเด็ก ก็ยังสามารถสื่อสารได้ซึ่งในกรณีนี้จำนวนตัวอย่างที่ใช้สอนจะมีอย่างจำกัด ดังนั้นการปรับแต่งแบบจำลองในกระบวนการสอนจึงมีความสำคัญมาก [3]

ในแง่ของข้อจำกัดทางทรัพยากรคอมพิวเตอร์ซึ่งหมายถึงคอมพิวเตอร์ที่มีขนาดเล็กและราคาถูกลง จำเป็นต้องสอนแบบจำลองเพื่อให้แบบจำลองที่ได้มีขนาดเล็กลง ซึ่งในประเด็นนี้จะสัมพันธ์เกี่ยวข้องกับการปรับแต่งค่าไฮเปอร์พารามิเตอร์ของแบบจำลอง รวมทั้งสถาปัตยกรรมของแบบจำลองด้วย ยิ่งถ้าพิจารณาการใช้งานแบบจำลองนี้กับระบบสมองกลฝังตัวที่มีทรัพยากรฮาร์ดแวร์จำกัด จึงนับเป็นประเด็นที่น่าสนใจ

2.7 เกณฑ์การวัดคะแนน BLEU (Bilingual Evaluation Understudy)

BLEU เป็นวิธีการวัดประสิทธิภาพของการแปลอัตโนมัติที่โดยอิงจากบทความ [14] โดยมีวัตถุประสงค์เพื่อให้การประเมินผลการแปลทำได้อย่างรวดเร็วและมีความสอดคล้องกับการประเมินของมนุษย์ การคำนวณคะแนน BLEU มีองค์ประกอบสำคัญดังนี้:

2.7.1 การจับคู่ n-gram ที่ถูกแก้ไข (Modified n-gram Precision)

BLEU จะคำนวณความแม่นยำของ n-gram ในประโยคที่แปลโดยระบบเทียบกับประโยคอ้างอิงที่มนุษย์แปล โดยมีการปรับแก้ไขเพื่อป้องกันไม่ให้เกิดการนับคำซ้ำเกินความจำเป็น เช่น หากคำใดคำหนึ่งปรากฏในประโยคแปลมากกว่าที่ปรากฏในประโยคอ้างอิง การนับจะถูกจำกัดไว้ที่จำนวนสูงสุดที่ปรากฏในประโยคอ้างอิงเท่านั้น

2.7.2 การคำนวณความแม่นยำของ n-gram หลายระดับ

BLEU จะคำนวณความแม่นยำของ n-gram ในหลายระดับ เช่น unigram, bigram, trigram และ 4-gram จากนั้นนำผลลัพธ์มารวมกันในรูปของค่าเฉลี่ยเรขาคณิต (Geometric Mean) เพื่อสะท้อนถึงความถูกต้องทั้งในแง่ของความหมายและความต่อเนื่องของภาษา

2.7.3 โทษจากความยาว (Brevity Penalty)

BLEU จะมีบทลงโทษสำหรับการแปลที่มีความยาวสั้นหรือยาวเกินไป เมื่อความยาวของประโยคที่แปลน้อยกว่าความยาวของประโยคอ้างอิง BLEU จะคูณคะแนนด้วยค่าโทษจากความยาว (brevity penalty)

2.7.4 การรวมคะแนน BLEU

BLEU จะคำนวณคะแนนสุดท้ายโดยการคูณค่าเฉลี่ยเรขาคณิตของ n-gram precision กับค่าโทษจากความยาว ดังนี้:

$$\boldsymbol{BLEU = BP \cdot exp(\sum_{n=1}^{N} W_N \, log p_n)} \qquad (2.7)$$

โดยที่ $\boldsymbol{W_n}$ คือค่าน้ำหนักของแต่ละ n-gram (ซึ่งมักตั้งค่าเป็น $\frac{1}{N}$ เท่า ๆ กัน) และ p_n คือค่าความแม่นยำของ n-gram ระดับที่ n

2.7.5 ผลลัพธ์ของ BLEU

คะแนน BLEU มีค่าอยู่ในช่วงระหว่าง 0 ถึง 1 โดยที่คะแนนสูงใกล้ 1 หมายถึงการแปลมีความใกล้เคียงกับการแปลของมนุษย์มาก ในทางปฏิบัติ คะแนน BLEU ที่สูงกว่า 0.3 ถึง 0.4 มักสะท้อนถึงการแปลที่มีคุณภาพดี

2.8 วรรณกรรมที่เกี่ยวกับการประยุกต์ใช้อัลกอริทึมเชิงพันธุกรรมในแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส

2.8.1 บทความ: “Optimizing Transformer for Low-Resource Neural Machine Translation” [15]

จากบทความนี้ศึกษาประสิทธิภาพของระบบการแปลภาษาด้วยเครื่องประสาทกับภาษาที่มีทรัพยากรจำกัดโดยแสดงให้เห็นว่า การแปลภาษาด้วยเครื่องประสาทมีประสิทธิภาพต่างกันระหว่างภาษาที่มีชุดข้อมูลขนาดใหญ่กับขนาดเล็ก การปรับปรุงการแปลภาษาด้วยเครื่องประสาทในสถานการณ์ชุดข้อมูลขนาดเล็ก มุ่งไปที่การใช้ข้อมูลเพิ่มเติม ซึ่งระบบที่ปรับแต่งอย่างเหมาะสมสามารถทำงานได้ดีในสภาพชุดข้อมูลขนาดเล็ก โดยพบว่าการปรับค่าไฮเปอร์พารามิเตอร์ที่เหมาะสมมีความสำคัญต่อประสิทธิภาพของการแปลภาษาด้วยเครื่องประสาทโดยเฉพาะกับขนาดข้อมูลที่แตกต่างกัน ตั้งแต่ 5,000 ถึง 165,000 คู่ประโยค แสดงให้เห็นถึงความสำคัญของการเลือกค่าไฮเปอร์พารามิเตอร์อย่างเหมาะสม จากตารางค่าที่เป็นโครงสร้างส่วนสำคัญคือ 1,2,3 และ 5 ซึ่งจะนำไปเป็นส่วนการปรับค่าไฮเปอร์พารามิเตอร์ในส่วนของวิธีการหาค่าเหมาะที่สุดของแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส

ตารางที่ 1 ค่าไฮเปอร์พารามิเตอร์ของแบบจำลองทรานส์ฟอร์เมอร์ส

Step	Hyper-parameter	Values
1	feed-forward dimension	128, 256, 512, 1024, 2048, 4096
2	embedding dimension	256, 512, 1024
3	attention heads	1, 2, 4, 8, 16
4	dropout	0.1, 0.2, 0.3, 0.4, 0.5
5	number of layers	1, 2, 3, 4, 5, 6, 7
6	label smoothing	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8
7	enc/dec layer dropout	0.1, 0.2, 0.3, 0.4
8	src/tgt word dropout	0.1, 0.2, 0.3
9	attention dropout	0.1, 0.2, 0.3
10	activation dropout	0.1, 0.2, 0.3, 0.4, 0.5
11	embedding layer normalization	yes, no
12	batch size	512, 1024, 2048, 4096, 8192, 12288
13	learning rate scheduler	Transformer standard, inverse square root
14	warm-up steps	2000, 4000, 5000, 6000, 8000, 10000
15	learning rate	0.01, 0.001, 0.0001, 0.00001

ที่มา: บทความ [15]

2.8.2 บทความ: “Neural Machine Translation for Low-resource Languages: A Survey” [3]

บทความนี้แสดงให้เห็นภาพรวมอย่างละเอียดของความก้าวหน้าในเทคโนโลยีการแปลภาษาด้วยเครื่องแบบประสาทสำหรับภาษาที่มีทรัพยากรต่ำ ซึ่งมักจะได้รับการแทนที่อย่างมากในข้อมูลที่มีให้สำหรับระบบการแปลภาษา แม้จะมีการพัฒนาเทคโนโลยีการแปลภาษาด้วยเครื่องประสาทอย่างรวดเร็ว ประสิทธิภาพสำหรับภาษาที่มีทรัพยากรข้อมูลจำกัดยังคงไม่เพียงพอเนื่องจากการขาดแคลนคลังข้อมูลคู่ขนาดใหญ่

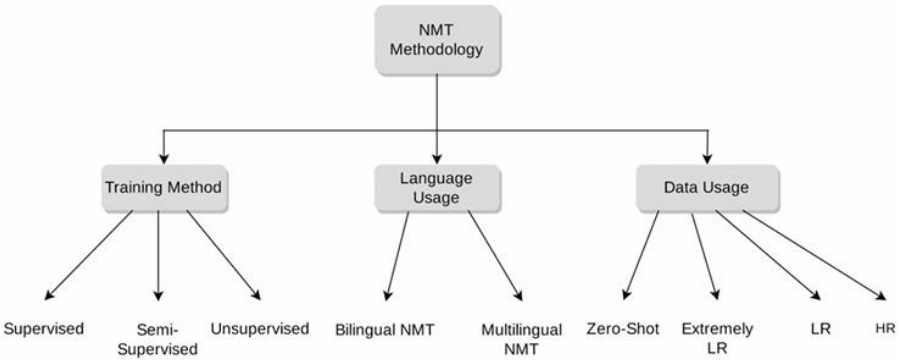
- การแปลภาษาด้วยเครื่องประสาทแบบมีผู้สอน: วิธีการนี้เป็นสถาปัตยกรรมพื้นฐานที่ต้องพึ่งพาชุดข้อมูลคู่ขนานขนาดใหญ่ สถาปัตยกรรมเครือข่ายประสาทที่ใช้งานทั่วไปได้แก่ เครือข่ายประสาทแบบวนซ้ำที่มีกลไกการให้ความสนใจและสถาปัตยกรรม แบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สที่เพิ่งถูกนำมาใช้ เนื่องจากความจำเป็นในการใช้ชุดข้อมูลคู่ขนานขนาดใหญ่ ทำให้การแปลภาษาด้วยเครื่องประสาทแบบมีผู้สอนมีความท้าทายสำหรับภาษาที่มีทรัพยากรต่ำ

- การแปลภาษาด้วยเครื่องประสาทแบบกึ่งมีผู้สอน: เมื่อมีชุดข้อมูลคู่ขนานขนาดเล็ก สามารถผสมผสานข้อมูลคู่ขนานเหล่านี้กับข้อมูลภาษาทางเดียวในลักษณะกึ่งมีผู้สอน วิธีการนี้ช่วยปรับปรุงคุณภาพการแปลแม้ว่าจะมีชุดข้อมูลคู่ขนานที่จำกัด

- การแปลภาษาด้วยเครื่องประสาทแบบไม่มีผู้สอน: ในกรณีที่ไม่มีชุดข้อมูลคู่ขนานเลย สามารถใช้เทคนิคการแปลภาษาด้วยเครื่องประสาทแบบไม่มีผู้สอนได้ วิธีการเหล่านี้ไม่ต้องพึ่งพาชุดข้อมูลคู่ขนาน ทำให้เหมาะสมสำหรับการแปลภาษาที่มีทรัพยากรต่ำ

- แบบจำลองการแปลหลายภาษาด้วยเครื่องประสาท: การสร้างแบบจำลองการแปลภาษาด้วยเครื่องประสาทแบบสองภาษาสำหรับแต่ละคู่ภาษานั้นไม่สามารถทำได้ในทางปฏิบัติ โดยเฉพาะสำหรับการแปลภาษาที่มีทรัพยากรต่ำ ซึ่งสามารถแปลระหว่างคู่ภาษาหลายคู่โดยใช้แบบจำลองเดียว เป็นวิธีแก้ปัญหาที่ดี ส่วนใหญ่ใช้การแปลภาษาด้วยเครื่องประสาทแบบมีผู้สอน แต่มีการวิจัยเกี่ยวกับการประยุกต์ใช้ในบริบทกึ่งมีผู้สอนและไม่มีผู้สอนอย่างต่อเนื่องแบบจำลองเหล่านี้มีความน่าสนใจอย่างยิ่งในการแปลคู่ภาษาที่มีทรัพยากรต่ำ

- การเพิ่มข้อมูล: เพื่อแก้ปัญหาการขาดแคลนชุดข้อมูลคู่ขนานสำหรับที่มีทรัพยากรต่ำ สามารถใช้เทคนิคการเพิ่มข้อมูล ซึ่งเกี่ยวข้องกับการสร้างข้อมูลสังเคราะห์เพื่อเพิ่มขนาดชุดข้อมูลการฝึก เทคนิคเหล่านี้สามารถใช้งานได้ไม่ว่าจะใช้สถาปัตยกรรมการแปลภาษาด้วยเครื่องประสาทใดก็ตาม การเพิ่มข้อมูลสามารถใช้เพื่อสร้างคู่ประโยคคู่ขนานสังเคราะห์เพื่อฝึกแบบจำลองแปลภาษาที่มุ่งเน้นข้อมูล เทคนิคการเพิ่มข้อมูลโดยทั่วไปจะไม่เปลี่ยนแปลงสถาปัตยกรรมการแปลภาษาด้วยเครื่องประสาท แต่จะสร้างข้อมูลเพื่อฝึกสถาปัตยกรรมเหล่านี้ เทคนิคการเพิ่มข้อมูลสำหรับการแปลภาษาด้วยเครื่องประสาท สามารถแบ่งออกเป็นสามประเภท ได้แก่ 1.การเพิ่มข้อมูลโดยการแทนที่คำหรือวลี 2.การเพิ่มข้อมูลโดยการแปลย้อนกลับ 3.การขุดค้นคอร์ปัสคู่ขนาน



ภาพที่ 6 วิธีการที่เกี่ยวข้องกับเครื่องแปลภาษาประสาท
ที่มา: บทความ [3]

2.8.3 บทความ: “Set-Based Particle Swarm Optimization: A Review” [10]

บทความนี้นำเสนอการทบทวนสถานะปัจจุบันของการวิจัยเกี่ยวกับอัลกอริทึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาค โดยมุ่งเน้นไปที่การปรับแต่งแบบ Set-Based Particle Swarm Optimization (SBPSO) วัตถุประสงค์หลักของบทความคือการรวมและทบทวนวรรณกรรมเกี่ยวกับอัลกอริทึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคที่ใช้แนวคิดแบบเซตและการให้ภาพรวมที่ครอบคลุมของ SBPSO ซึ่งถือว่าเป็นอัลกอริทึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคที่มีพื้นฐานแน่นในทฤษฎีเซตและมีการใช้งานหลากหลาย

บทความนี้ได้ทบทวนและอธิบายการทำงานของ 8 อัลกอริทึม ดังนี้

1. การเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคสำหรับปัญหาไม่ต่อเนื่อง
   (Discrete Particle Swarm Optimization หรือ DPSO)
2. การเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคแบบเซต
   (Set Particle Swarm Optimization หรือ SetPSO)
3. การเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคเซต
   (Set Swarm Optimization หรือ SSO)
4. การเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคแบบเซตพื้นฐาน
   (Set-based Particle Swarm Optimization หรือ S-PSO)
5. การเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคแบบคลุมเครือ
   (Fuzzy Particle Swarm Optimization หรือ FPSO)
6. การเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคแบบคลุมเครือวิวัฒนาการ
   (Fuzzy Evolutionary Particle Swarm Optimization หรือ FEPSO)
7. การเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคแบบจำนวนเต็มและจำแนกประเภท
   (Integer and Categorical Particle Swarm Optimization หรือ ICPSO)
8. การเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคแบบเซตหยาบ
   (Rough set-based Particle Swarm Optimization หรือ RoughPSO)

นอกจากนี้ยังมีการทบทวนอัลกอริทึม SBPSO ซึ่งพิสูจน์ว่าเป็นอัลกอริทึมแบบเซตที่มีพื้นฐานมั่นคงในทฤษฎีเซตและมีการประยุกต์ใช้ที่หลากหลาย มีการทบทวนการประยุกต์ใช้ SBPSO เจ็ดประการได้แก่ การแก้ปัญหากระเป๋าหลายมิติ, การเลือกคุณลักษณะ, การเพิ่มประสิทธิภาพพอร์ตโฟลิโอ, การประมาณพหุนาม, การฝึกซัพพอร์ตเวกเตอร์แมชชีน, การจัดกลุ่มและการหากฎ สรุปว่าการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคแบบเซ็ตเป็นการปรับเปลี่ยนที่สำคัญของรูปแบบการปรับเพิ่มประสิทธิภาพแบบกลุ่มอนุภาค ทำให้สามารถหาทางแก้ไขปัญหาแบบไม่ต่อเนื่องและการคำนวณแบบผสมผสานได้อย่างมีประสิทธิภาพ อย่างไรก็ตาม ยังต้องมีการพัฒนาและการปรับปรุงต่อเนื่องเพื่อให้สามารถใช้ประโยชน์จากศักยภาพได้อย่างเต็มที่ในหลากหลายสาขาการใช้งาน

2.8.4 บทความ: “Particle Swarm Optimization for Hyper-Parameter Selection in Deep Neural Networks” [16]

บทความนี้กล่าวถึงการใช้อัลกอริทึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคเพื่อเลือกปรับแต่งค่าไฮเปอร์พารามิเตอร์อัตโนมัติในเครือข่ายประสาทเทียมลึก วิธีการนี้โดดเด่นในความสามารถในการสำรวจพื้นที่คำตอบ ซึ่งทำให้สามารถใช้การกำหนดค่าเครือข่ายประสาทเทียมลึกที่น้อยที่สุดเพื่อทำให้มีประสิทธิภาพการแข่งขันสูงในชุดข้อมูลภาพที่ซับซ้อน MNIST และ CIFAR-10

การปรับแต่งพารามิเตอร์ปรับแต่งในเครือข่ายประสาทเทียมลึกมีความสำคัญสำหรับประสิทธิภาพ แต่มักต้องการความรู้ของผู้เชี่ยวชาญและการคำนวณที่กว้างขวาง การเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคเสนอทางเลือกที่ได้แรงบันดาลใจจากชีววิทยาและมีประสิทธิภาพในการคำนวณ วิธีการและการทดลองเกี่ยวกับแนวทางของพวกเขาโดยใช้การเพิ่มประสิทธิภาพแบบกลุ่มอนุภาค เป็นวิธีห่อหุ้มกระบวนการฝึกอบรมปรับแต่งพารามิเตอร์เพื่อลดข้อผิดพลาดในการจำแนกประเภท พวกเขาดำเนินการทดลองอย่างกว้างขวางที่แสดงให้เห็นว่าการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคสามารถปรับปรุงประสิทธิภาพของสถาปัตยกรรม เครือข่ายประสาทเทียมลึกที่มีอยู่ได้ และมีประสิทธิภาพโดยเฉพาะเมื่อใช้กับโทโพโลยีเครือข่ายขนาดเล็ก การทดลองแสดงให้เห็นว่าเครือข่ายประสาทเทียมลึกที่ได้รับการปรับให้เหมาะสมด้วย การเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคสามารถบรรลุความแม่นยำในการจำแนกประเภทที่น่าสนใจบนชุดข้อมูลภาพ MNIST และ CIFAR-10 โดยมักจะเอาชนะรุ่นที่ซับซ้อนกว่าที่ไม่ได้รับการปรับให้เหมาะสมด้วยการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาค

สรุปว่าการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคเป็นเครื่องมือที่มีประสิทธิภาพสำหรับการปรับแต่งพารามิเตอร์ปรับแต่งในเครือข่ายประสาทเทียม โดยให้วิธีการที่ทรงสมดุลระหว่างการสำรวจและการใช้ประโยชน์จากพื้นที่พารามิเตอร์อย่างมีประสิทธิภาพ วิธีนี้สามารถขยายได้และสามารถปรับใช้กับสถาปัตยกรรมเครือข่ายประสาทเทียมลึกและชุดข้อมูลที่หลากหลาย

บทที่ 3
วิธีดำเนินการวิจัย

ในส่วนของวิธีในการดำเนินงานวิจัยนี้ จะกล่าวถึงขั้นตอนการดำเนินงานของงานวิจัยฉบับนี้ ซึ่งสามารถแบ่งขั้นตอนการดำเนินงานได้เป็น 6 ขั้นตอน คือ รวบรวมข้อมูลและกำหนดขอบเขตของงาน ศึกษาทฤษฎีที่เกี่ยวข้องกับการประยุกต์ใช้อัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคกับแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส ออกแบบอัลกอริทึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาค ออกแบบการทดลอง ดำเนินการทดลอง และสรุปผลการดำเนินงานวิจัยและจัดทำรายงานวิทยานิพนธ์ ซึ่งมีรายละเอียดของแต่ละขั้นตอนการดำเนินงานดังนี้

3.1 รวบรวมข้อมูลและกำหนดขอบเขตของงาน

ในขั้นตอนนี้จะเป็นการศึกษา รวบรวมข้อมูลที่เกี่ยวข้องกับงาน ผลงานที่เกี่ยวข้อง แล้วนำข้อมูลที่ได้มาทำการวิเคราะห์ และกำหนดถึงขอบเขตงานวิจัย ซึ่งมีรายละเอียดดังนี้

3.1.1 กำหนดโปรแกรมที่ใช้ในการสร้างแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สสำหรับการวิจัย

จากการทบทวนงานวิจัยที่เกี่ยวข้อง พบว่าปัจจุบันมีโปรแกรมหลายตัวที่ใช้เขียนโปรแกรมและพัฒนาแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ในภาษาไพทรอน (Python) อาทิเช่น Google Colab, PyCharm, และ Anaconda ด้วยเหตุนี้การวิจัยนี้จึงเลือกใช้โปรแกรม Anaconda ในส่วนของ Jupyter lab เพราะมีความสะดวกในการใช้ชุดคำสั่งและรองรับการใช้ Cuda สำหรับการประมวลผลด้วยการ์ดจอ ทำให้ผลการทดสอบแบบจำลองมีประสิทธิภาพและความแม่นยำ รวมทั้งช่วยลดเวลาในการทดสอบได้อย่างมีประสิทธิภาพ

3.1.2 กำหนดชุดข้อมูลที่ใช้ในการทดสอบและประเมินความสามารถของแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส

เพื่อประเมินประสิทธิภาพของโมเดลการเรียนรู้เชิงลึกที่พัฒนาขึ้นโดยการปรับค่าไฮเปอร์พารามิเตอร์ตามอัลกอริทึมที่ได้เลือกใช้ การวิจัยนี้จึงใช้ชุดข้อมูล IWSLT14 เป็นชุดข้อมูลคู่ประโยคภาษาเยอรมัน-ภาษาอังกฤษซึ่งได้อ้างอิงตาม [15] ได้นำมาใช้ในการทดลอง 165,000 คู่ประโยค และชุดข้อมูล SCB ซึ่งเป็นคู่ประโยคภาษาไทย-อังกฤษ [13] ได้

เลือกใช้ในส่วนของ Taskmaster-1 ตามที่อธิบายไปในบทก่อนหน้า นำมาใช้ 200,000 คู่ประโยค ซึ่งศึกษาเครื่องมือทางด้านซอฟต์แวร์เกี่ยวกับภาษาไทย [17], [18] และเครื่องมือวัดผลการประเมินแบบสองภาษา (Bilingual Evaluation Understudy-BLEU) [14] เป็นหนึ่งในเมทริกที่ใช้กันอย่างแพร่หลายในการประเมินคุณภาพของข้อความที่แปลโดยระบบการแปลภาษาอัตโนมัติ โดยเฉพาะในด้านการแปลภาษา เพื่อทดสอบและประเมินความสามารถของระบบการเรียนรู้เชิงลึก ทำให้สามารถเปรียบเทียบผลลัพธ์ที่ได้กับงานวิจัยอื่น ๆ ที่เกี่ยวข้องได้อย่างมีมาตรฐานและเชื่อถือได้

3.2 ศึกษาทฤษฎีที่เกี่ยวข้องกับการประยุกต์ใช้อัลกอริทึมเชิงพันธุกรรมกับแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส

ในขั้นตอนนี้ เป็นการศึกษาทฤษฎีและแนวคิดที่เกี่ยวข้องกับการประยุกต์ใช้อัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค กับแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส ซึ่งมีความสำคัญต่อการออกแบบและพัฒนาระบบที่ใช้ในการทดลอง รวมถึงการปรับแต่งไฮเปอร์พารามิเตอร์ให้มีประสิทธิภาพสูงสุด โดยหัวข้อที่สำคัญในส่วนนี้ประกอบด้วย:

1. ความรู้พื้นฐานเกี่ยวกับการเรียนรู้เชิงลึก

การเรียนรู้เชิงลึกเป็นสาขาหนึ่งของการเรียนรู้ของเครื่องที่เน้นการใช้โครงข่ายประสาทเทียมที่มีโครงสร้างหลายชั้น ในการเรียนรู้ลักษณะเด่น จากข้อมูลเชิงซ้อน เช่น ข้อความ รูปภาพ และเสียง โดยโครงสร้างที่ซับซ้อนนี้ช่วยให้โมเดลสามารถจับความสัมพันธ์ที่ลึกซึ้งและซับซ้อนของข้อมูลได้ ซึ่งเป็นรากฐานสำคัญสำหรับการพัฒนาระบบการแปลภาษาโดยอาศัยแบบจำลองทรานส์ฟอร์เมอร์ส [1]

2. แบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส

ทรานส์ฟอร์เมอร์สเป็นแบบจำลองการเรียนรู้เชิงลึกที่ได้รับความนิยมสูงในงานด้านการประมวลผลภาษาธรรมชาติ โดยมีองค์ประกอบสำคัญคือ Self-Attention Mechanism ที่ช่วยให้แบบจำลองสามารถโฟกัสกับบริบทของคำในประโยคได้อย่างมีประสิทธิภาพ แบบจำลองการเรียนรู้เชิงลึดชนิดทรานส์ฟอร์เมอร์สยังมีความสามารถในการปรับแต่งไฮเปอร์พารามิเตอร์ เช่น ขนาดของเวกเตอร์คำ จำนวนชั้นของ Encoder และ Decoder รวมถึง

ค่าที่เกี่ยวข้องกับ Attention Mechanism ซึ่งทั้งหมดนี้สามารถปรับปรุงได้โดยใช้อัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค [2]

3. ทฤษฎีการแปลภาษาด้วยเครื่องประสาท

ทฤษฎีการแปลภาษาด้วยเครื่องประสาทเป็นพื้นฐานสำคัญสำหรับการใช้งานแบบจำลองการเรียนรู้เชิงลึดชนิดทรานส์ฟอร์เมอร์สในงานแปลภาษา โดยเฉพาะการเรียนรู้แบบลำดับคำ และการใช้ Attention Mechanism ที่ช่วยเพิ่มประสิทธิภาพการแปล ในงานวิจัยนี้ การศึกษาวรรณกรรมเกี่ยวกับการแปลภาษาโดยอาศัยการแปลภาษาด้วยเครื่องประสาทจะช่วยสนับสนุนการประยุกต์ใช้อัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคในการปรับแต่งไฮเปอร์พารามิเตอร์ [3]

4. อัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค

เป็นอัลกอริทึมเชิงวิวัฒนาการที่ได้รับแรงบันดาลใจจากพฤติกรรมการเคลื่อนไหวของฝูงสัตว์ เช่น นกและปลา โดยอัลกอริทึมนี้มีข้อดีในด้านความเร็วในการค้นหาและความสามารถในการหลีกเลี่ยงการติดอยู่ในค่าต่ำสุดท้องถิ่น ซึ่งเหมาะสำหรับการหาค่าที่เหมาะสมที่สุดของไฮเปอร์พารามิเตอร์ในแบบจำลองที่มีโครงสร้างซับซ้อนอย่างแบบจำลองการเรียนรู้เชิงลึดชนิดทรานส์ฟอร์เมอร์ส [4]

5. วรรณกรรมที่เกี่ยวข้องกับการประยุกต์ใช้อัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคกับแบบจำลองการเรียนรู้เชิงลึดชนิดทรานส์ฟอร์เมอร์ส

การศึกษาวรรณกรรมเกี่ยวกับการใช้อัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคในการปรับแต่งไฮเปอร์พารามิเตอร์ของแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส จะช่วยให้เข้าใจถึงแนวทางปฏิบัติที่ดีที่สุด และผลลัพธ์ที่ได้จากงานวิจัยที่ผ่านมา ตัวอย่างการประยุกต์อัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคที่ประสบความสำเร็จ เช่น การลดเวลาในการฝึกแบบจำลอง การเพิ่มความแม่นยำในการพยากรณ์ หรือการปรับปรุงประสิทธิภาพการแปลในบริบทที่มีข้อมูลจำนวนน้อย

การศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องทั้งหมดนี้ จะถูกนำไปใช้เป็นแนวทางสำคัญสำหรับการออกแบบและพัฒนาระบบในงานวิจัยนี้ เพื่อให้ได้ผลลัพธ์ที่มีความแม่นยำและประสิทธิภาพสูงสุด

3.3 ออกแบบอัลกอริทึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาค

การเพิ่มประสิทธิภาพแบบกลุ่มอนุภาค (Particle Swarm Optimization: PSO) เป็นอัลกอริทึมเชิงเมตาฮิวริสติกที่ได้แรงบันดาลใจจากพฤติกรรมของฝูงปลาและนกในการค้นหาอาหารในธรรมชาติ แนวคิดพื้นฐานคือการจำลองพฤติกรรมการค้นหาดังกล่าวผ่านกลุ่มที่เรียกว่า "ฝูง" (Swarm) ซึ่งประกอบด้วยหน่วยย่อยที่เรียกว่า "อนุภาค" (Particles) อนุภาคเหล่านี้จะทำการเคลื่อนที่ในพื้นที่คำตอบ (Search Space) เพื่อค้นหาตำแหน่งที่ให้ค่าความเหมาะสมสูงที่สุด [5]

สถานะของอนุภาค แต่ละอนุภาคในฝูงจะมีสถานะสำคัญ 2 อย่างที่ต้องบริหารจัดการ:

1. ตำแหน่ง (Position): ระบุตำแหน่งของอนุภาคในพื้นที่ค้นหาที่กำหนด
2. ความเร็ว (Velocity): กำหนดทิศทางและขนาดของการเคลื่อนที่ของอนุภาคไปยังตำแหน่งใหม่ในแต่ละรอบของการอัปเดต

กระบวนการอัปเดตตำแหน่งและความเร็ว การปรับปรุงตำแหน่งและความเร็วของอนุภาคเกิดขึ้นโดยการรวมข้อมูลจากแหล่งสำคัญสองประการ:

- ตำแหน่งที่ดีที่สุดส่วนบุคคล (Personal Best): ค่าที่ดีที่สุดที่อนุภาคนั้นเคยค้นพบ
- ตำแหน่งที่ดีที่สุดของฝูง (Global Best): ค่าที่ดีที่สุดที่อนุภาคทั้งหมดในฝูงเคยค้นพบ

กระบวนการนี้ช่วยให้อนุภาคสามารถปรับตัวเข้าหาตำแหน่งที่มีค่าความเหมาะสมสูงสุดได้ โดยอาศัยการผสมผสานระหว่างการสำรวจพื้นที่ใหม่และการใช้ข้อมูลจากประสบการณ์ที่ผ่านมาเพื่อเพิ่มประสิทธิภาพในการค้นหา

Algorithm 1 Proposed Particle Swarm Optimization

initialize: hyperparameter sets $\boldsymbol{H}_i$ in Eq. (2) in Table I,
number of paticle N, number of iteration t,
total number of hypeparamerter values N_s,
$t = 0$
for i = 1 : N **do**
  initialize D_i^k in Eq. (4) for all set $\boldsymbol{H}_i$ in Eq. (2)
  initialize X_i in Eq. (3) from D_i^k in $\mathbb{R}^{N_s}$
  initialize fitness of the position best to zero $f(P_{b,i}) = \mathbf{0}$
  find solution S_i in Eq. (1) by using Eq. (8) (9) (10)
  find fitness $f(S_i)$ by training limited num. epochs
end for
//set fitness of the global best to zero
$f(G_{b,}) = \mathbf{0}$
while not stopping criteria **do**
  for i = 1 : N **do**
    // update position best $P_{b,i} = [p_i^1, p_i^2, p_i^3, \dots, p_i^D]$
    if $f(S_i) > f(P_{b,i})$ **then**
      $P_{b,i} = S_i$
    end if
    // update global best $G_b = [g^1, g^2, g^3, \dots, g^D]$
    if $f(P_{b,i}) > f(G_{b,})$ **then**
      $G_b = P_{b,i}$
    end if
    update velocity using Eq. (6)
    update position using Eq. (7)
    find solution S_i in Eq. (1) by using Eq. (8) (9) (10)
    find fitness $f(S_i)$ by training limited num. epochs
    $t = t + 1$
  end for
end while

ภาพที่ 7 กระบวนการของอัลกอริทึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาค

3.3.1 การปรับปรุงอัลกอริธึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาค

ในงานวิจัยนี้ อัลกอริทึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคถูกดัดแปลงเพื่อให้สามารถใช้งานได้กับค่าไฮเปอร์พารามิเตอร์ที่มีลักษณะเป็นตัวแปรแบบหมวดหมู่ (Categorical) ซึ่งไม่สามารถจัดการได้โดยตรงด้วยอัลกอริทึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคดั้งเดิมที่ออกแบบมาสำหรับตัวแปรแบบ ต่อเนื่องแนวทางที่ใช้ในงานวิจัยนี้มีรายละเอียดดังนี้

3.3.1.1 การแทนค่าพารามิเตอร์ด้วยเวกเตอร์ความน่าจะเป็น (Probability Vector)

ในการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาค ค่าไฮเปอร์พารามิเตอร์ที่อยู่ในรูปแบบหมวดหมู่ เช่น จำนวน Attention Heads [2, 4, 8, 16] หรือค่า Feed-forward Dimension [256, 512, 1024, 2048] จะถูกแทนด้วยชุดของค่าที่เป็นไปได้ เพื่อใช้ในการค้นหาและอัปเดตค่าที่เหมาะสมที่สุด การแทนค่าด้วยเวกเตอร์ความน่าจะเป็นอนุภาคแต่ละตัวในฝูง จะมี

ตำแหน่งที่แสดงในรูปของเวกเตอร์ความน่าจะเป็น เวกเตอร์นี้ประกอบด้วยค่าความน่าจะเป็นแต่ละตัว ซึ่งแสดงถึงโอกาสที่ค่าไฮเปอร์พารามิเตอร์ในชุดที่กำหนดจะถูกเลือก

ตัวอย่าง สมมติค่าไฮเปอร์พารามิเตอร์คือ Attention Heads และเวกเตอร์ความน่าจะเป็นมีค่าเป็น [0.1, 0.3, 0.5, 0.1]:

· โอกาสเลือกค่าพารามิเตอร์ 2 = 0.1 (10%)

· โอกาสเลือกค่าพารามิเตอร์ 4 = 0.3 (30%)

· โอกาสเลือกค่าพารามิเตอร์ 8 = 0.5 (50%)

· โอกาสเลือกค่าพารามิเตอร์ 16 = 0.1 (10%)

เวกเตอร์ความน่าจะเป็นนี้ช่วยในการแสดงการกระจายตัวของค่าพารามิเตอร์ในพื้นที่การค้นหา และยังสามารถอัปเดตค่าความน่าจะเป็นในเวกเตอร์ได้ตามกระบวนการเรียนรู้ของการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคเพื่อมุ่งเน้นไปยังค่าที่มีความเหมาะสมสูงสุด

3.3.1.2 การอัปเดตตำแหน่งและความเร็วของเวกเตอร์

อัลกอริทึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคดั้งเดิมมีสมการอัปเดตความเร็ว ($\boldsymbol{v}$) และตำแหน่ง ($\boldsymbol{x}$) ที่รองรับค่าต่อเนื่อง งานวิจัยนี้ปรับให้เหมาะสมกับค่าพารามิเตอร์แบบหมวดหมู่โดยใช้เวกเตอร์ความน่าจะเป็น แทนตำแหน่ง ($\boldsymbol{x}$) และความเร็ว ($\boldsymbol{v}$) ถูกอัปเดตตามสมการการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคดั้งเดิม โดยผลลัพธ์ที่ได้จะถูกใช้ปรับค่าในเวกเตอร์ความน่าจะเป็น สมการพื้นฐานในการอัปเดตความเร็วและตำแหน่งของอนุภาคคือ:

- การปรับความเร็ว:

$$V_i = \omega V_i + c_1 r_1 (P_{best} - X_i) + c_2 r_2 (G_{best} - X_i) \quad (3.1)$$

- การปรับตำแหน่ง:

$$X_i = X_i + V_i \quad (3.2)$$

โดยที่

- ω คือน้ำหนักความเฉื่อย (Inertia Weight)

- c_1 และ c_2 คือค่าสัมประสิทธิ์การเรียนรู้ (Learning Coefficients)
- r_1 และ r_2 ค่าที่สุ่มขึ้นในช่วง [0,1]
- X_i คือตำแหน่งของอนุภาคที่ i
- P_{best} และ G_{best} คือตำแหน่งที่ดีที่สุดของอนุภาคและของฝูงตามลำดับ

3.3.1.3 การปรับค่าความน่าจะเป็นด้วยฟังก์ชันซอฟต์แม็กซ์

หลังจากการอัปเดตความเร็ว (v) และตำแหน่ง (x) ของอนุภาคแล้วเวกเตอร์ความน่าจะเป็นที่ได้อาจมีค่าที่อยู่นอกช่วง [0,1] และผลรวมของค่าความน่าจะเป็นในเวกเตอร์อาจไม่เท่ากับ 1 เพื่อแก้ปัญหานี้ ฟังก์ชันซอฟต์แม็กซ์ถูกนำมาใช้ในการปรับค่าความน่าจะเป็น มีคุณสมบัติการทำให้อยู่ในช่วง [0, 1] ฟังก์ชันซอฟต์แม็กซ์จะปรับให้ค่าทั้งหมดในเวกเตอร์ความน่าจะเป็นอยู่ในช่วง [0,1] เมื่อผลรวมเท่ากับ 1 ผลรวมของค่าความน่าจะเป็นทั้งหมดในเวกเตอร์ที่ได้จะเท่ากับ 1 ซึ่งช่วยให้ เป็นไปตามนิยามของเวกเตอร์ความน่าจะเป็น

$$\boldsymbol{P_i} = \frac{e^{v_i}}{\sum_{j=1}^{n} e^{v_j}} \tag{3.3}$$

โดยที่

- P_i คือค่าความน่าจะเป็นหลังการปรับสำหรับตัวเลือก i
- v_i คือค่าความเร็วของตัวเลือก i หลังจากการอัปเดต
- n คือจำนวนตัวเลือกทั้งหมดในเวกเตอร์

ตัวอย่างค่า (v) ของตัวเลือกในเวกเตอร์ความน่าจะเป็นคือ [1.2,0.8,-0.5,0.3] นำมาคำนวณ e^{v_i} จะได้ค่า [3.32,2.23,0.61,1.35] คำนวณผลรวมได้ 7.51 หลังจากนั้นจะนำมา คำนวณหาค่า P_i ตามสมการข้างต้นจะได้ [0.44,0.3,0.08,0.18]

3.3.1.4 การเลือกค่าไฮเปอร์พารามิเตอร์

วิธีการเลือกค่าพารามิเตอร์ คือ ตรวจสอบค่าเวกเตอร์ความน่าจะเป็นซึ่งระบุความน่าจะเป็นของแต่ละตัวเลือกและเลือกค่าไฮเปอร์พารามิเตอร์ที่มีค่าความน่าจะเป็นสูงสุดจากเวกเตอร์ความน่าจะเป็นให้เป็นค่าไฮเปอร์พารามิเตอร์สำหรับอนุภาคในรอบการอัปเดตปัจจุบัน

ตัวอย่างเช่น ค่าเวกเตอร์ความน่าจะเป็นคือ [0.2,0.5,0.2,0.1] ของค่าไฮเปอร์พารามิเตอร์ [2,4,8,16] ค่า 4 ซึ่งมีความน่าจะเป็นสูงสุด 0.5 จึงถูกเลือก

3.3.1.5 การคำนวณค่าความเหมาะสม

การประเมินค่าความเหมาะสมของอนุภาคในแต่ละรอบการอัปเดต ถูกดำเนินการผ่านการทดลองใช้งานแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส โดยใช้ค่าไฮเปอร์พารามิเตอร์ที่เลือกมาจากเวกเตอร์ความน่าจะเป็นและวัดประสิทธิภาพของแบบจำลองด้วยคะแนน BLEU ซึ่งเป็นตัวชี้วัดที่ใช้ประเมินคุณภาพของผลลัพธ์การแปลภาษา

กระบวนการประเมินค่าความเหมาะสม

1. การฝึกแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส: ใช้ค่าไฮเปอร์พารามิเตอร์ที่ได้จากเวกเตอร์ความน่าจะเป็นของอนุภาคในรอบการอัปเดตและทำการฝึกแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สกับชุดข้อมูลที่จัดเตรียมไว้คือ ภาษาอังกฤษ-ภาษาเยอรมัน และ ภาษาไทย-ภาษาอังกฤษ
2. การวัดผลลัพธ์ด้วยคะแนน BLEU: คำนวณคะแนน BLEU เพื่อวัดคุณภาพของการแปลภาษาซึ่งคะแนน BLEU ยิ่งสูงหมายถึงการแปลที่มีความใกล้เคียงกับข้อความต้นฉบับมากขึ้น

การอัปเดตตำแหน่งที่ดีที่สุด หลังจากที่ค่าความเหมาะสมถูกคำนวณแล้ว:

1. ตำแหน่งที่ดีที่สุดส่วนบุคคล: หากคะแนน BLEU ที่ได้ในรอบนี้สูงกว่าค่าที่ดีที่สุดของอนุภาคนั้นในรอบก่อนหน้า จะทำการอัปเดตตำแหน่งที่ดีที่สุดส่วนบุคคลให้เป็นค่าปัจจุบัน
2. ตำแหน่งที่ดีที่สุดของฝูง: หากคะแนน BLEU ที่ได้ในรอบนี้สูงกว่าค่าที่ดีที่สุดของอนุภาคทั้งหมดในฝูง จะทำการอัปเดตตำแหน่งที่ดีที่สุดของฝูงให้เป็นค่าปัจจุบัน

3.3.1.6 กระบวนการวนซ้ำจนกว่าจะถึงเกณฑ์หยุด

อัลกอริทึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคจะทำงานในรูปแบบการวนซ้ำ (Iterative Process) โดยในแต่ละรอบจะมีการอัปเดตความเร็ว ($\boldsymbol{v}$) และตำแหน่ง ($\boldsymbol{x}$) และค่าความน่าจะเป็นในเวกเตอร์ความน่าจะเป็น จนกว่าจะถึงเกณฑ์หยุดที่กำหนดไว้

กระบวนการวนซ้ำ ในแต่ละรอบของการทำงาน PSO จะดำเนินการดังนี้:

1. อัปเดตความเร็ว ($\boldsymbol{v}$): คำนวณความเร็วใหม่ตามสมการของการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาค
2. อัปเดตตำแหน่ง ($\boldsymbol{x}$): ปรับค่าความน่าจะเป็นในเวกเตอร์ความน่าจะเป็นโดยใช้ความเร็วที่อัปเดต
3. ปรับค่าความน่าจะเป็นด้วยฟังก์ชันซอฟต์แม็กซ์: เพื่อทำให้ค่าความน่าจะเป็นอยู่ในช่วง [0,1] และผลรวมเท่ากับ 1
4. เลือกค่าพารามิเตอร์: เลือกค่าไฮเปอร์พารามิเตอร์ที่มีความน่าจะเป็นสูงสุดจากเวกเตอร์ความน่าจะเป็น
5. ประเมินค่าความเหมาะสม: ฝึกแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สด้วยค่าไฮเปอร์พารามิเตอร์ที่เลือก และวัดผลด้วยคะแนน BLEU
6. อัปเดตตำแหน่งที่ดีที่สุด: อัปเดตตำแหน่งที่ดีที่สุดส่วนบุคคลและตำแหน่งที่ดีที่สุดของฝูงหากค่าความเหมาะสมในรอบปัจจุบันดีกว่า

เกณฑ์หยุดการทำงาน กระบวนการดังกล่าวจะดำเนินการวนซ้ำจนกว่าจะถึงเกณฑ์หยุดที่กำหนด ซึ่งในกรณีนี้คือการทำงานของการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคจะหยุดเมื่อรอบการวนซ้ำถึงค่าที่กำหนดไว้ล่วงหน้าซึ่งเราได้กำหนดไว้ที่ 50 รุ่น(รอบ)

3.4 ออกแบบการทดลอง

ขั้นตอนนี้เป็นขั้นตอนในการออกแบบและพัฒนาแบบจำลองจากแนวความคิดใหม่ที่ได้จากการวิเคราะห์และสังเคราะห์ความรู้เกี่ยวกับการประยุกต์ใช้อัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคกับการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส ที่ได้จากการทบทวนวรรณกรรมและการรวบรวมข้อมูล แล้วนำมาสร้างและพัฒนาเป็นระบบและอัลกอริทึมใหม่ โดยระบบที่สร้างขึ้นจะถูกพัฒนาปรับปรุง และทดสอบ ด้วยโปรแกรม Anaconda

3.4.1 การทดสอบแบบจำลองที่ได้

1. การทดสอบในคอมพิวเตอร์เพื่อประเมินประสิทธิภาพการแปลภาษาของแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สที่ปรับค่าไฮเปอร์พารามิเตอร์ด้วยอัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค ใช้เครื่องมือวัดผลการประเมินแบบสองภาษาเพื่อทดสอบและวัดผลการแปลภาษาของแบบจำลอง ค่าที่ได้จากการทดสอบนี้จะถูกนำไปเปรียบเทียบกับงานวิจัยอื่น ๆ ที่เกี่ยวข้องเพื่อหาความแม่นยำและประสิทธิภาพในการแปลภาษาของแบบจำลอง

2. การทดสอบแบบจำลองในระบบสมองกลฝังตัว เริ่มต้นจากการนำค่าไฮเปอร์พารามิเตอร์จากอัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคที่พัฒนาขึ้น ไปฝึกแบบจำลองที่เหมาะสมกับระบบสมองกลฝังตัว ในสภาพแวดล้อมของระบบสมองกลฝังตัว แบบจำลองจะถูกตรวจสอบความสามารถในการทำงานภายใต้ข้อจำกัดของฮาร์ดแวร์ในระบบสมองกลฝังตัว เช่น การจำกัดพลังงาน และทรัพยากรคำนวณที่น้อยกว่าเมื่อเปรียบเทียบกับคอมพิวเตอร์ การทดสอบนี้จะช่วยให้เห็นภาพชัดเจนถึงประสิทธิภาพของแบบจำลองในการใช้งานจริงและการปรับแต่ง เพื่อให้เหมาะสมกับการใช้งานในระบบฝังตัว

3.5 การดำเนินการทดลอง

การดำเนินการทดลองประกอบด้วย 2 ขั้นตอนหลัก ได้แก่:

1. การฝึกแบบจำลอง (Training) ในคอมพิวเตอร์
    - ใช้ชุดข้อมูลคู่ขนาน ในการฝึกแบบจำลองบนแพลตฟอร์ม Anaconda
    - ปรับค่าไฮเปอร์พารามิเตอร์ของแบบจำลองโดยใช้อัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค เพื่อเพิ่มประสิทธิภาพในการเรียนรู้
    - ประเมินประสิทธิภาพของแบบจำลองและปรับปรุงเพื่อเพิ่มความแม่นยำในการแปลภาษา
2. นำแบบจำลองไปใช้งานใน Raspberry Pi 4

- ทำการติดตั้งแบบจำลองที่ผ่านการฝึกและทดสอบแล้วในระบบสมองกลฝังตัว Raspberry Pi 4
- ประเมินประสิทธิภาพในการทำงานในสภาพแวดล้อมที่มีข้อจำกัดด้านทรัพยากร เช่น พลังงานและหน่วยความจำ
- ปรับแต่งและปรับปรุงแบบจำลองให้มีขนาดเล็กลงและเหมาะสมกับการทำงานในระบบฝังตัว เพื่อให้ได้ประสิทธิภาพสูงสุดภายใต้ข้อจำกัดของฮาร์ดแวร์

3.6 รวบรวม วิเคราะห์ผลการทดลองและจัดทำรายงานวิทยานิพนธ์

การสรุปผลของการดำเนินงานวิจัยและจัดทำรายวิทยานิพนธ์ถือเป็นขั้นตอนสุดท้ายของการวิจัยนี้ โดยจะนำผลลัพธ์ที่ได้จากการทดลองทั้งหมด มาทำการวิเคราะห์และประเมินผลว่าตรงตามจุดประสงค์ที่กำหนดไว้หรือไม่ จากการวิจัยได้ผลการทดลองเป็นที่น่าพอใจ และเป็นไปตามวัตถุประสงค์ที่ตั้งไว้ทุกประการ งานวิจัยนี้ [19] ได้ถูกเผยแพร่ในรูปของบทความที่งานประชุมวิชาการระดับนานาชาติ ICSEC 2024 ที่จัดขึ้นในวันที่ 6-8 พฤศจิกายน 2567 ณ โรงแรมพูลแมนขอนแก่น ราชา ออคิด จังหวัดขอนแก่น นอกจากนี้บทความที่ได้รับการตอบรับให้เข้าร่วมจะได้รับการบรรจุเข้าไปในฐานข้อมูล IEEE

บทที่ 4
ผลการดำเนินงาน

ตามที่กล่าวไว้ในบทก่อนหน้า งานวิจัยนี้เป็นการวิจัยเพื่อการหาค่าเหมาะสมที่สุดของค่าไฮเปอร์พารามิเตอร์ให้แก่แบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สโดยใช้การเพิ่มประสิทธิภาพแบบกลุ่มอนุภาค ในบทนี้จะนำเสนอการรูปแบบการทดลอง ขั้นตอนการทดลอง ผลการทดลองและการประยุกต์ใช้แบบจำลองกับระบบสมองกลฝังตัว

4.1 รูปแบบการทดลอง

4.1.1 ชุดข้อมูลสำหรับประมวลผลในเครื่องคอมพิวเตอร์

การทดลองในงานวิจัยนี้ใช้ชุดข้อมูลสองชุดข้อมูล ได้แก่ ภาษาอังกฤษ-เยอรมัน (IWSLT14) และ ภาษาอังกฤษ-ไทย (SCB) เพื่อวัดประสิทธิภาพของการปรับค่าไฮเปอร์พารามิเตอร์แบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สผ่านอัลกอริทึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคที่พัฒนาขึ้น

โดยชุดข้อมูลถูกแบ่งออกเป็นหลายขนาดตามบทความ [15] เพื่อศึกษาผลกระทบของขนาดข้อมูลต่อประสิทธิภาพของแบบจำลอง ขนาดข้อมูลที่ใช้ในการทดลองได้แก่ 5,000, 10,000, 20,000, 40,000, 80,000 และ 165,000 ตัวอย่าง ซึ่งเป็นจำนวนข้อมูลสำหรับชุดฝึกสอน (train set) นอกจากนี้ ยังได้จัดสรรข้อมูลสำหรับชุดตรวจสอบ (validation set) และชุดทดสอบ (test set) อย่างละ 10% ของแต่ละชุดข้อมูล เพื่อประเมินความแม่นยำของแบบจำลองโดยใช้คะแนน BLEU จากนั้นจึงนำผลคะแนนที่ได้ไปเปรียบเทียบกับผลลัพธ์ของแบบจำลองที่ใช้ค่าไฮเปอร์พารามิเตอร์พื้นฐานเพื่อวัดประสิทธิภาพในการปรับค่าไฮเปอร์พารามิเตอร์ที่พัฒนาเพิ่มเติม

4.1.2 ทรัพยากรฮาร์ดแวร์

ในการดำเนินการทดลองและประมวลผลในงานวิจัยนี้ ได้ใช้ทรัพยากรฮาร์ดแวร์ที่มีประสิทธิภาพสูง เพื่อรองรับการฝึกแบบจำลองและการปรับค่าไฮเปอร์พารามิเตอร์ที่มีความซับซ้อนสูง โดยรายละเอียดของฮาร์ดแวร์ที่ใช้มีดังนี้:

4.1.2.1 เครื่องคอมพิวเตอร์หลักสำหรับประมวลผลแบบจำลอง

เครื่องคอมพิวเตอร์หลักที่ใช้ในการฝึกแบบจำลองและปรับแต่งค่าไฮเปอร์พารามิเตอร์มีรายละเอียดดังนี้:

- **หน่วยประมวลผลกลาง (CPU):**
    - รุ่น: AMD Ryzen 7 5700X
    - สถาปัตยกรรม: Zen 3 (7nm)
    - จำนวนคอร์ (Core): 8 คอร์
    - จำนวนเธรด (Thread): 16 เธรด
    - ความเร็วสัญญาณนาฬิกา: ฐานที่ 3.4 GHz และเพิ่มได้สูงสุด 4.6 GHz
    - แคช (Cache): L2 ขนาด 4 MB และ L3 ขนาด 32 MB
    - TDP: 65 W
- **หน่วยประมวลผลกราฟิก (GPU):**
    - รุ่น: NVIDIA GeForce RTX 4090
    - สถาปัตยกรรม: Ada Lovelace (4nm)
    - หน่วยความจำ (VRAM): 24 GB GDDR6X
    - CUDA Cores: 16,384 คอร์
    - Tensor Cores: 512 คอร์ สำหรับการประมวลผลเชิงลึก
    - ความเร็วสัญญาณนาฬิกา: ฐานที่ 2.23 GHz และเพิ่มได้สูงสุด 2.52 GHz
    - Memory Bandwidth: 1,008 GB/s

- กำลังไฟฟ้า (TDP): 450 W

- **หน่วยความจำหลัก (RAM):**
  - ขนาด: 32 GB
  - ประเภท: DDR4
  - ความเร็ว: 3200 MHz
  - โหมดการทำงาน: Dual Channel

4.1.2.2 อุปกรณ์เสริมสำหรับการทดลองในสภาพแวดล้อมจำกัด

นอกจากนี้ ยังได้ใช้ Raspberry Pi 4 เพื่อจำลองการทำงานในสภาพแวดล้อมที่มีข้อจำกัดด้านทรัพยากร โดยมีรายละเอียดดังนี้:

- **หน่วยประมวลผลกลาง (CPU):**
  - รุ่น: Broadcom BCM2711
  - สถาปัตยกรรม: ARM Cortex-A72 (64-bit)
  - จำนวนคอร์ (Core): 4 คอร์
  - ความเร็วสัญญาณนาฬิกา: ฐานที่ 1.5 GHz
- **หน่วยความจำหลัก (RAM):**
  - ขนาด: 8 GB
  - ประเภท: LPDDR4-3200
- **หน่วยจัดเก็บข้อมูล (Storage):**
  - ประเภท: microSD
  - ขนาด: 64 GB
  - ความเร็วการอ่าน/เขียน: สูงสุด 100 MB/s

ฮาร์ดแวร์ที่เลือกใช้ในงานวิจัยนี้ถูกออกแบบมาเพื่อรองรับการประมวลผลเชิงลึกที่ต้องการพลังประมวลผลสูง โดยเฉพาะในการฝึกแบบจำลอง Transformer ที่มีความซับซ้อนสูง ซึ่งต้องอาศัยหน่วยประมวลผลกราฟิก (GPU) ที่มีประสิทธิภาพสูงเพื่อเร่งการประมวลผล รวมถึงหน่วยความจำที่เพียงพอสำหรับการจัดเก็บและประมวลผลข้อมูลขนาดใหญ่ ในขณะเดียวกัน Raspberry Pi 4 ถูกใช้เพื่อทดสอบและประเมินประสิทธิภาพของแบบจำลองในสภาพแวดล้อมที่มีทรัพยากรจำกัด เพื่อให้สามารถวิเคราะห์ความสามารถในการทำงานของแบบจำลองในเงื่อนไขที่แตกต่างกันได้อย่างครอบคลุม

4.1.3 การตั้งค่าไฮเปอร์พารามิเตอร์ของแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ส

ในงานวิจัยนี้ แบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สถูกปรับแต่งผ่านการตั้งค่าไฮเปอร์พารามิเตอร์ที่เหมาะสม [15], [19], [20] โดยใช้วิธีการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาค เพื่อค้นหาค่าที่ดีที่สุดสำหรับแบบจำลองการแปลภาษา มี 5 องค์ประกอบหลักที่จะปรับดังนี้

ตารางที่ 2 ค่าไฮเปอร์พารามิเตอร์ที่ใช้ในอัลกอริทึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาค

ลำดับ	ค่าไฮเปอร์พารามิเตอร์	ค่าพื้นฐาน
1	Feed-forward dimension	256,512,1024,2048
2	Embedding dimension	128,256,512,1024
3	Attention heads	2,4,8,16
4	Number of encoder layers	4,5,6,7
5	Number of decoder layers	4,5,6,7

รายละเอียดการตั้งค่า

1. Feed-forward dimension: ค่าพื้นฐานที่ใช้คือ 256, 512, 1024 และ 2048 มีผลต่อขนาดของชั้นแบบ Feed-forward ซึ่งช่วยเพิ่มความสามารถในการจับลักษณะของข้อมูล
2. Embedding dimension: ค่าพื้นฐานที่ใช้คือ 128, 256, 512 และ 1024 กำหนดขนาดของเวกเตอร์การฝัง ซึ่งใช้แทนคำในข้อมูลข้อความ

3. Attention heads: ค่าพื้นฐานที่ใช้คือ 2, 4, 8 และ 16 กำหนดจำนวนหัวความสนใจที่ใช้ในกลไก Multi-head Attention เพื่อเรียนรู้ความสัมพันธ์ระหว่างคำ
4. Number of encoder layers: ค่าพื้นฐานที่ใช้คือ 4, 5, 6 และ 7 กำหนดจำนวนเลเยอร์ในส่วนของตัวเข้ารหัสซึ่งมีหน้าที่ประมวลผลข้อมูลขาเข้า
5. Number of decoder layers: ค่าพื้นฐานที่ใช้คือ 4, 5, 6 และ 7 กำหนดจำนวนเลเยอร์ในส่วนของตัวถอดรหัสซึ่งใช้สร้างผลลัพธ์การแปลภาษา

4.2 ขั้นตอนการทดลอง

1. การตั้งค่าเริ่มต้น
   - ขนาดของฝูง (Swarm Size): 8 อนุภาค แต่ละอนุภาคแทนค่าชุดพารามิเตอร์ที่แตกต่างกัน
   - จำนวนรุ่น (Generations): 50 รุ่น กระบวนการค้นหาค่าที่เหมาะสมที่สุดจะทำงานวนซ้ำ 50 รอบเพื่ออัปเดตความเร็ว ตำแหน่ง และค่าความน่าจะเป็น
2. การฝึกแบบจำลอง แบ่งออกเป็น 2 ขั้นตอนหลัก:
   1. ค้นหาค่าไฮเปอร์พารามิเตอร์เบื้องต้น: ทำการฝึกแบบจำลองในแต่ละชุดค่าไฮเปอร์พารามิเตอร์ที่ค้นหาได้จากการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาค จำนวน 30 Epochs เพื่อประเมินคะแนน BLEU จะถูกใช้ในการอัปเดตตำแหน่งที่ดีที่สุดส่วนบุคคลและตำแหน่งที่ดีที่สุดของฝูง
   2. Fine-Tuning แบบจำลองที่ดีที่สุด: เมื่อเลือกค่าชุดไฮเปอร์พารามิเตอร์ที่ดีที่สุดจากขั้นตอนการค้นหาแล้ว จะทำการ Fine-Tuning แบบจำลองเพิ่มเติมด้วยจำนวน 300 Epochs เพื่อปรับปรุงความแม่นยำและประสิทธิภาพให้ดียิ่งขึ้น ใช้ชุดข้อมูลที่กำหนดสำหรับการฝึกและการประเมินผล

4.3 ผลการทดลอง

ในการวิจัยครั้งนี้ ได้ดำเนินการทดลองสองชุด ได้แก่ การแปลภาษาเยอรมันเป็นภาษาอังกฤษ และการแปลภาษาไทยเป็นภาษาอังกฤษ ผลการทดลองแสดงให้เห็นว่าอัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค สามารถค้นหาค่าไฮเปอร์พารามิเตอร์ที่เหมาะสมที่สุดสำหรับแบบจำลองการ

เรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สในบริบทของการแปลภาษาที่มีขนาดชุดข้อมูลแตกต่างกัน ผลการทดลองและการเปรียบเทียบมีรายละเอียดดังนี้

4.3.1 การแปลภาษาเยอรมันเป็นภาษาอังกฤษ

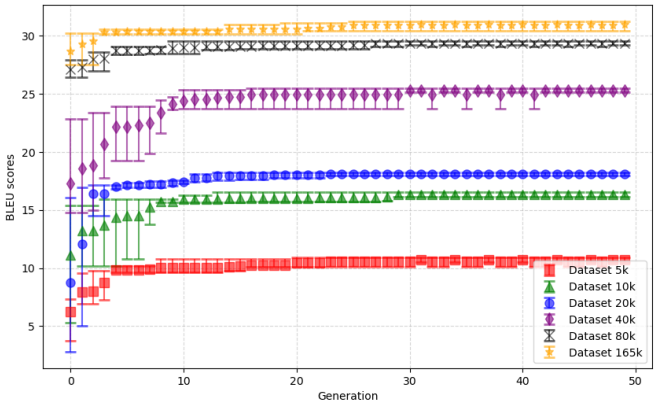
ผลการทดลองสำหรับชุดข้อมูลภาษาเยอรมัน-ภาษาอังกฤษแสดงใน ตารางที่ 3 ซึ่งแสดงค่าไฮเปอร์พารามิเตอร์ที่ถูกค้นหาโดยอัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค สำหรับชุดข้อมูลขนาดดังนี้ 5,000, 10,000, 20,000, 40,000, 80,000 และ 165,000 ประโยค จากการทดลองพบว่าการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคสามารถปรับปรุงค่าไฮเปอร์พารามิเตอร์ให้เหมาะสมจนทำให้คะแนน BLEU เพิ่มขึ้นตามขนาดของชุดข้อมูล โดยคะแนน BLEU สูงสุดอยู่ที่ 31.2 สำหรับชุดข้อมูลขนาด 165,000 ประโยค

ตารางที่ 3 ค่าไฮเปอร์พารามิเตอร์ที่ได้จากอัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคที่พัฒนาขึ้นของการแปลภาษาเยอรมันเป็นภาษาอังกฤษ

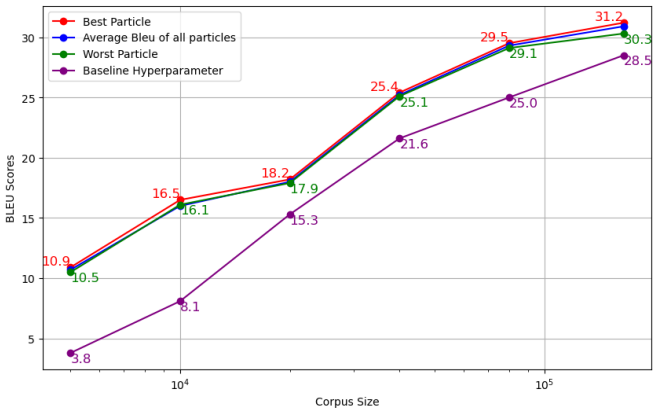
		Hyperparameter Values					
ID	Configuration	5k	10k	20k	40k	80k	165k
1	Feed-forward dimension	512	2048	256	2048	1024	2048
2	Embedding dimension	1024	1024	512	512	512	256
3	Attention heads	2	2	4	2	2	8
4	Number of encoder layers	6	4	5	5	5	5
5	Number of decoder layers	7	5	6	7	7	7
6	Dropout	0.1	0.1	0.1	0.1	0.1	0.1
7	Label smoothing	0.1	0.1	0.1	0.1	0.1	0.1
8	Batch size	128	128	128	128	128	128
Training Times (Hour) 32 gen		11	16	26	44	74	113
BLEU		10.9	16.5	18.2	25.4	29.5	31.2

ตารางที่ 4 ผลลัพธ์ของคะแนน BLEU ที่ Fine-tuning ของชุดข้อมูลภาษาเยอรมัน-อังกฤษ

Sentences DE → EN	T-base	T-Opt
5k	3.8	11.4
10k	8.1	16.7
20k	15.3	19.5
40k	21.6	25.6
80k	25.0	29.7
165k	28.5	31.5



ภาพที่ 8 ผลลัพธ์คะแนน BLEU จากกระบวนการค้นหาของภาษาเยอรมันเป็นภาษาอังกฤษ



ภาพที่ 9 ผลลัพธ์คะแนน BLEU เทียบกับขนาดของชุดข้อมูลในการแปลภาษาเยอรมันเป็นภาษาอังกฤษ

ตารางที่ 3 แสดงค่าไฮเปอร์พารามิเตอร์และคะแนน BLEU ที่ได้ในแต่ละขนาดของชุดข้อมูล สำหรับการแปลภาษาเยอรมันเป็นภาษาอังกฤษ การเปรียบเทียบประสิทธิภาพแสดงใน ตารางที่ 4 โดยแบบจำลองที่ปรับแต่งด้วยการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค (T-opt) ที่ทำการฝึกอบรมละเอียดมากขึ้น 300 epochให้คะแนน BLEU ที่สูงกว่าแบบจำลองเริ่มต้น (T-base) ในทุกขนาดของชุดข้อมูล

ภาพที่ 8 แสดงการเพิ่มขึ้นของคะแนน BLEU ในแต่ละรุ่นของการฝึก โดยอัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคที่พัฒนาขึ้นช่วยให้แบบจำลองเข้าถึงค่าที่เหมาะสมที่สุดได้อย่างรวดเร็ว คะแนน BLEU เพิ่มขึ้นอย่างมีนัยสำคัญหลังจากรุ่นแรก ๆ และเข้าสู่ค่าสูงสุดในรุ่นที่ 32 สำหรับแต่ละขนาดของชุดข้อมูล

ภาพที่ 9 แสดงกราฟคะแนน BLEU ตามขนาดชุดข้อมูล โดยแต่ละจุดแสดงคะแนน BLEU ที่ดีที่สุด (best), ค่าเฉลี่ย (average) และค่าต่ำสุด (worst) จากกลุ่มตัวอย่างในรุ่นที่ 32 รวมถึงค่าไฮเปอร์พารามิเตอร์เริ่มต้น เปรียบเทียบกับค่าไฮเปอร์พารามิเตอร์พื้นฐาน สำหรับชุดข้อมูลแต่ละขนาดสามารถสังเกตได้ว่าคะแนน BLEU เพิ่มขึ้นอย่างรวดเร็วหลังจากที่ค่าไฮเปอร์พารามิเตอร์เข้าใกล้ค่าที่เหมาะสมที่สุด

สรุปผลการทดลองแปลภาษาเยอรมันเป็นภาษาอังกฤษ แสดงให้เห็นว่าอัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคที่พัฒนาขึ้น มีประสิทธิภาพในการปรับแต่งค่าไฮเปอร์พารามิเตอร์ของแบบจำลองทรานส์ฟอร์เมอร์ส ส่งผลให้คะแนน BLEU เพิ่มขึ้นอย่างต่อเนื่องในทุกขนาดของชุดข้อมูลเมื่อเปรียบเทียบกับค่าไฮเปอร์พารามิเตอร์พื้นฐาน ซึ่งดูได้จากภาพ 9 แบบจำลองที่ได้รับการปรับแต่ง (T-opt) สามารถทำงานได้ดีกว่าแบบจำลองพื้นฐาน (T-base) ในทุกกรณี โดยคะแนน BLEU สูงสุดอยู่ที่ 31.5 สำหรับชุดข้อมูลขนาด 165,000 ประโยค

4.3.2 การแปลภาษาไทยเป็นภาษาอังกฤษ

ตารางที่ 5 ค่าไฮเปอร์พารามิเตอร์ที่ได้จากอัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคที่พัฒนาขึ้นของการแปลภาษาไทยเป็นภาษาอังกฤษ

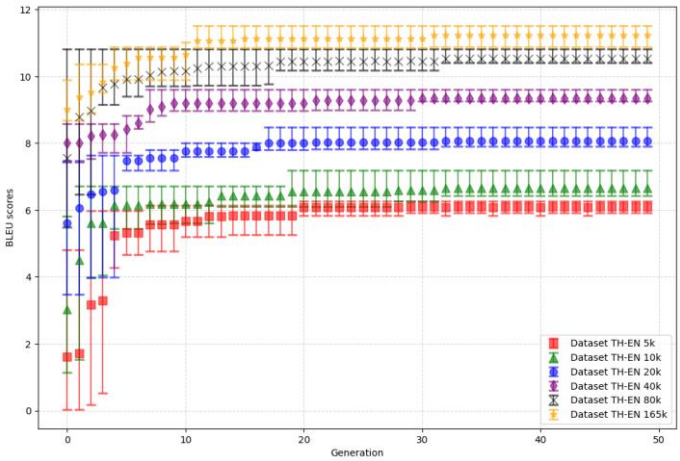
		Hyperparameter Values					
ID	Configuration	5k	10k	20k	40k	80k	165k
1	Feed-forward dimension	1024	512	256	1024	2048	2048
2	Embedding dimension	512	512	512	256	128	256
3	Attention heads	8	2	8	2	4	2
4	Number of encoder layers	4	6	4	5	6	5
5	Number of decoder layers	7	6	6	7	5	6
6	Dropout	0.1	0.1	0.1	0.1	0.1	0.1
7	Label smoothing	0.1	0.1	0.1	0.1	0.1	0.1
8	Batch size	128	128	128	128	128	128
Training Times (Hour) 32 gen		14	20	34	62	120	192
BLEU		6.2	7.1	8.4	9.6	10.8	11.7

ตารางที่ 6 ผลลัพธ์ของคะแนน BLEU ที่ Fine-tuning ของชุดข้อมูลภาษาไทย-อังกฤษ

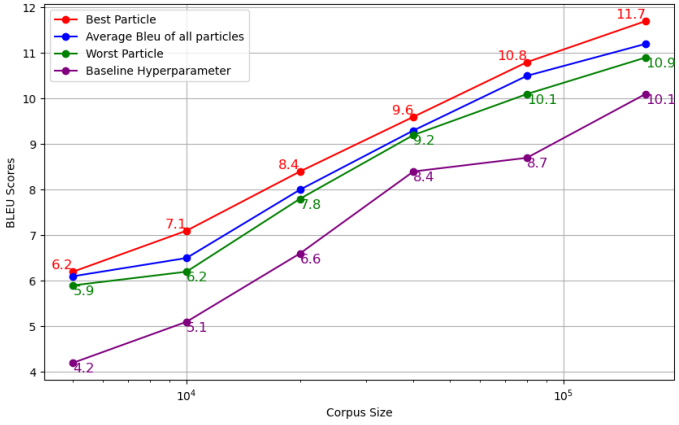
Sentences TH→ EN	T-base	T-Opt
5k	4.2	6.7
10k	5.1	7.6
20k	6.6	8.9
40k	8.4	10.1
80k	8.7	11.3
165k	10.1	12.1

การแปลภาษาไทยเป็นอังกฤษแสดงในตารางที่ 5 ซึ่งแสดงค่าไฮเปอร์พารามิเตอร์ที่ถูกปรับแต่งให้เหมาะสมสำหรับชุดข้อมูลขนาด 5,000, 10,000, 20,000, 40,000, 80,000 และ 165,000 ประโยค คะแนน BLEU สูงสุดที่ได้คือ 12.1 สำหรับชุดข้อมูลขนาด 165,000 ประโยค ผลการทดลองพบว่าคะแนน BLEU เพิ่มขึ้นอย่างช้า ๆ เมื่อฝึกแบบจำลองด้วยชุดข้อมูลขนาดใหญ่ขึ้น แต่ยังคงได้คะแนน BLEU ที่ค่อนข้างต่ำ ซึ่งผลลัพธ์นี้สอดคล้องกับการแปลภาษาไทย-อังกฤษในงานวิจัย [13] ที่ได้คะแนน BLEU สูงสุดไม่เกิน 18 เมื่อใช้ชุดข้อมูล SCB ที่มี 1 ล้านคู่ประโยค อย่างไรก็ตาม แม้จะใช้ชุดข้อมูลที่มีขนาดแตกต่างกัน แต่อัลกอริทึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาค ที่เสนอสามารถค้นหาค่าพารามิเตอร์ที่ให้คะแนน BLEU ใกล้เคียงกันได้ ซึ่งสะท้อนถึงประสิทธิภาพของอัลกอริทึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคที่นำเสนอ

สำหรับการเปรียบเทียบ เราได้กำหนดค่าไฮเปอร์พารามิเตอร์ของแบบจำลองเริ่มต้น (baseline model) ตามที่แสดงในตารางที่ 6 แสดงคะแนน BLEU ของแบบจำลองที่ได้รับการปรับแต่งให้เหมาะสมและแบบจำลองเริ่มต้น สำหรับการแปลภาษาไทยเป็นอังกฤษ แบบจำลองที่ได้รับการปรับแต่งแสดงประสิทธิภาพการแปลที่ดีกว่าแบบจำลองเริ่มต้นอย่างมีนัยสำคัญ



ภาพที่ 10 ผลลัพธ์คะแนน BLEU จากกระบวนการค้นหาของภาษาไทยเป็นภาษาอังกฤษ



ภาพที่ 11 ผลลัพธ์คะแนน BLEU เทียบกับขนาดของชุดข้อมูลในการแปลภาษาไทยเป็นภาษาอังกฤษ

การปรับแต่งแบบจำลองสำหรับการแปลภาษาไทย-อังกฤษ แสดงประสิทธิภาพการค้นหาที่เพิ่มคะแนน BLEU ต่อรุ่นตามที่แสดงในภาพที่ 10 ผลการค้นหาแสดงการบรรลุค่าเปอร์พารามิเตอร์เฉพาะที่เหมาะสมที่สุด เมื่อขนาดชุดข้อมูลฝึกเพิ่มขึ้น แบบจำลองที่ปรับแต่งสามารถให้คะแนน BLEU ที่ดีขึ้น แต่มีช่วงการปรับปรุงที่ลดลง สำหรับการฝึกฝนโดยใช้ 30 epochs ต่อรุ่น คะแนน BLEU จะถึงค่าสูงสุดที่ประมาณรุ่นที่ 35

ภาพที่ 11 แสดงผลของขนาดชุดข้อมูลต่อคะแนน BLEU ซึ่งแสดงให้เห็นว่าขนาดชุดข้อมูลที่ใหญ่ขึ้นจะให้คะแนน BLEU ที่ดีกว่า อย่างไรก็ตามสำหรับแต่ละขนาดชุดข้อมูล

อัลกอริทึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคที่นำเสนอสามารถค้นหาค่าไฮเปอร์พารามิเตอร์ที่ช่วยเพิ่มประสิทธิภาพของแบบจำลองได้ดีกว่าแบบจำลองค่าไฮเปอร์พารามิเตอร์เริ่มต้น

สรุปผลการทดลองแปลภาษาไทยเป็นภาษาอังกฤษ แสดงให้เห็นว่าอัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคที่พัฒนาขึ้น สามารถปรับแต่งค่าไฮเปอร์พารามิเตอร์ของแบบจำลองทรานส์ฟอร์เมอร์ส สำหรับการแปลภาษาไทยเป็นภาษาอังกฤษได้อย่างมีประสิทธิภาพเมื่อเปรียบเทียบกับค่าไฮเปอร์พารามิเตอร์พื้นฐาน ซึ่งดูได้จากภาพที่ 11 แม้ว่าคะแนน BLEU จะเพิ่มขึ้นอย่างช้า ๆ ตามขนาดของชุดข้อมูล แต่แบบจำลองที่ได้รับการปรับแต่ง (T-opt) ก็ให้คะแนน BLEU สูงกว่าแบบจำลองเริ่มต้น (T-base) ในทุกขนาดของชุดข้อมูล โดยคะแนน BLEU สูงสุดที่ได้คือ 12.1 สำหรับชุดข้อมูลขนาด 165,000 ประโยค

4.3.3 สรุปผลการทดลอง

ผลการทดลองทั้งสองชุดแสดงให้เห็นว่าอัลกอริทึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคสามารถค้นหาค่าไฮเปอร์พารามิเตอร์ที่เพิ่มประสิทธิภาพของแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สได้อย่างมีประสิทธิภาพ โดยเฉพาะในงานแปลภาษาเยอรมัน-ภาษาอังกฤษ ซึ่งมีคะแนน BLEU สูงสุดที่ 31.5 อย่างไรก็ตาม ในงานแปลภาษาไทย-อังกฤษ คะแนน BLEU สูงสุดที่ 12.1 แสดงให้เห็นถึงความท้าทายในการประยุกต์ใช้ทรานส์ฟอร์เมอร์สกับภาษาที่มีโครงสร้างซับซ้อนและข้อมูลที่จำกัด

4.4 การประยุกต์ใช้แบบจำลองกับระบบสมองกลฝังตัว

ในการทดลองนี้ เราได้เลือกระบบสมองกลฝังตัวราสเบอร์รี่พาย 4 ซึ่งมีหน่วยความจำขนาด RAM 8 GB เนื่องจากราสเบอร์รี่พาย 4 มีประสิทธิภาพเพียงพอสำหรับการประมวลผลแบบจำลองภาษาขนาดกลางถึงขนาดใหญ่ เราได้ทำการศึกษารายละเอียดเชิงเทคนิคของบอร์ด ซึ่งมีคุณสมบัติที่สำคัญดังนี้:

- หน่วยประมวลผล (CPU): Quad-core Cortex-A72 (ARM v8) ความเร็ว 1.5 GHz
- หน่วยความจำ (RAM): ขนาด 8 GB

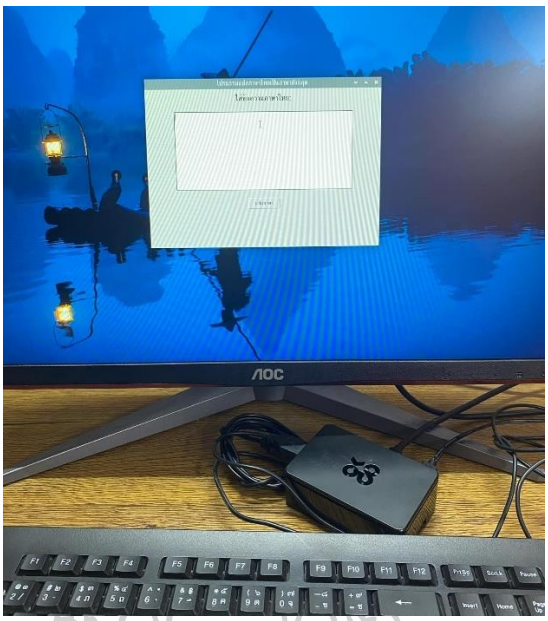
จากการศึกษานี้ เราได้ทำการทบทวนงานวิจัยและบทความที่เกี่ยวข้อง เพื่อค้นหาแบบจำลองการแปลภาษาที่มีประสิทธิภาพและสามารถทำงานได้บนแพลตฟอร์มที่มีทรัพยากรจำกัด โดยแบบจำลอง

ที่เลือกใช้คือ MarianMT [21], [22] ซึ่งเป็นแบบจำลองที่พัฒนาบนพื้นฐานของแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์ และได้นำค่าไฮเปอร์พารามิเตอร์ที่ได้จากการค้นหาด้วยอัลกอริธึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคที่พัฒนาขึ้นมาใช้ในการทดลอง การปรับแต่งขนาดของพารามิเตอร์ในแบบจำลองที่นำมาใช้งานบนราสเบอร์รี่พาย เพื่อให้สามารถทำงานได้โดยไม่เกิดปัญหาด้านหน่วยความจำหรือการประมวลผลที่ล่าช้า ในขั้นตอนนี้ เราได้เลือกใช้แบบจำลองที่มีขนาดของพารามิเตอร์ประมาณ 120 ล้านพารามิเตอร์ ซึ่งเหมาะสมกับหน่วยประมวลผลและหน่วยความจำของราสเบอร์รี่พาย 4 ทั้งนี้ เพื่อให้ระบบสามารถทำงานได้อย่างมีประสิทธิภาพและตอบสนองได้อย่างรวดเร็วในสภาวะแวดล้อมที่มีทรัพยากรจำกัด

4.4.1 การนำแบบจำลองไปใช้งานในระบบสมองกลฝังตัวประกอบด้วยขั้นตอนดังนี้

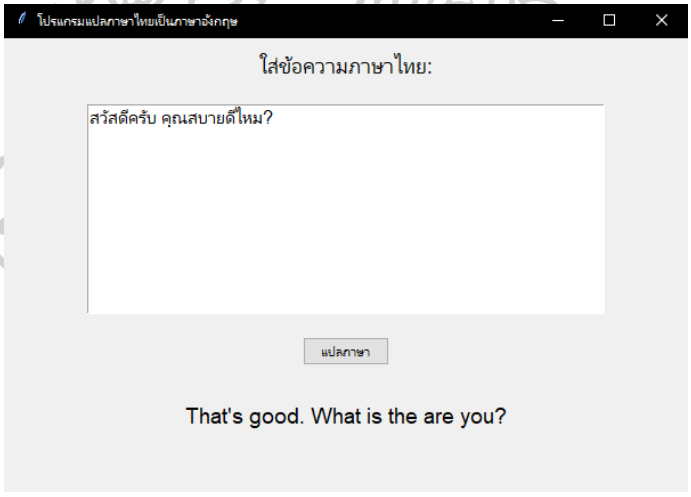
1. การเตรียมแบบจำลอง แบบจำลอง MarianMT ที่ผ่านการปรับแต่งไฮเปอร์พารามิเตอร์ถูกแปลงเป็นไฟล์ที่เหมาะสมสำหรับการใช้งานในสภาพแวดล้อมที่มีทรัพยากรจำกัด ลดขนาดและปรับแต่งแบบจำลองเพื่อให้สามารถประมวลผลได้บนระบบสมองกลฝังตัวราสเบอรี่พาย
2. การพัฒนา GUI ระบบถูกพัฒนาร่วมกับส่วนต่อประสานกราฟิกกับผู้ใช้ (Graphical User Interface: GUI) เพื่อให้ผู้ใช้งานสามารถใช้งานการแปลภาษาได้สะดวก
3. การทดสอบการใช้งาน ทดสอบระบบด้วยชุดข้อมูลตัวอย่างเพื่อวัดความถูกต้องของการแปลภาษา ประเมินประสิทธิภาพด้านเวลาในการประมวลผลและการตอบสนองของระบบ

ผลการทดลอง จากการทดลองใช้งาน พบว่าแบบจำลองที่พัฒนาขึ้นสามารถทำงานบนระบบสมองกลฝังตัวราสเบอร์รี่พาย ได้อย่างมีประสิทธิภาพ การแปลภาษาเป็นไปตามที่คาดหวัง โดยมีการตอบสนองที่เหมาะสมสำหรับการใช้งานในระดับเบื้องต้นใช้วเลาในการประมวลผล 20 วินาที การประยุกต์ใช้แบบจำลอง MarianMT [21] บนระบบสมองกลฝังตัวราสเบอร์รี่พาย 4 ช่วยให้สามารถทำงานด้านการแปลภาษาบนฮาร์ดแวร์ขนาดเล็กได้อย่างมีประสิทธิภาพ การปรับแต่งค่าไฮเปอร์พารามิเตอร์ด้วยอัลกอริธึม PSO ช่วยเพิ่มประสิทธิภาพในการแปล และลดภาระการประมวลผล ทำให้ระบบสามารถตอบสนองได้อย่างรวดเร็วแม้ในสภาพแวดล้อมที่มีทรัพยากรจำกัด ซึ่งนับเป็นแนวทางที่มีประโยชน์สำหรับการนำไปพัฒนาต่อในงานด้านการแปลภาษาบนระบบสมองกลฝังตัวในอนาคต



ภาพที่ 12 การใช้งานจริงต่อบอร์ดราสเบอรรี่พายกับหน้าจอ

4.4.2 ส่วนของหน้าโปรแกรมในระบบสมองกลฝังตัวราสเบอร์รี่พาย



ภาพที่ 13 GUI ในระบบสมองกลฝังตัวราสเบอร์รี่พาย

จากภาพที่ 12 จะเป็นหน้าต่างของโปรแกรมที่พัฒนาขึ้น สามารถพิมพ์ภาษาไทยที่จะทำการแปลภาษาลงในช่องว่าง หลังจากนั้นกดปุ่มแปลภาษาและรอผลลัพธ์ เราจะได้ค่าผลของการแปลภาษาไทยเป็นภาษาอังกฤษ จากตัวอย่าง “สวัสดีครับ คุณสบายดีไหม?” แปลได้เป็น “That’s good. What is the are you?”

บทที่ 5
สรุป อภิปรายผล และข้อเสนอแนะ

5.1 บทสรุปของงานวิจัย

วิทยานิพนธ์นี้ออกแบบและพัฒนาอัลกอริทึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคแบบเชิงกลุ่มสำหรับการปรับปรุงประสิทธิภาพค่าไฮเปอร์พารามิเตอร์ของแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สสำหรับการแปลภาษา โดยการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคที่นำเสนอในงานวิจัยนี้ได้พิจารณาค่าไฮเปอร์พารามิเตอร์เป็นตัวแปรเชิงกลุ่ม (Categorical Variables) ซึ่งค่าของตัวแปรเหล่านี้เป็นสมาชิกในชุดค่าที่กำหนดไว้ (Hyperparameter Set)

การเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคที่นำเสนอได้รับการปรับปรุงจากอัลกอริทึมดั้งเดิม โดยใช้ความน่าจะเป็นในการเลือกสมาชิกจากชุดค่าไฮเปอร์พารามิเตอร์ ด้วยการกำหนดค่าความน่าจะเป็นให้กับสมาชิกในชุดดังกล่าว การเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคจึงสามารถนำเทคนิคเวกเตอร์ไลเซชันมาใช้และอัปเดตเวกเตอร์ตำแหน่งโดยอิงค่าความน่าจะเป็นเหล่านี้

การทดลองในงานวิจัยนี้ดำเนินการกับชุดข้อมูลคู่ภาษาเยอรมัน-อังกฤษ (IWSLT14) และคู่ไทย-อังกฤษ (SCB) [13] เพื่อประเมินประสิทธิภาพในการค้นหาค่าที่เหมาะสมที่สุด (Value Discovery) ผลลัพธ์จากกราฟความเหมาะสมแสดงให้เห็นว่า การเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคที่นำเสนอสามารถหาคำตอบที่เหมาะสมได้อย่างรวดเร็ว นอกจากนี้ ผลลัพธ์การทดลองในรูปแบบคะแนน BLEU ชี้ให้เห็นว่าอัลกอริทึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคที่พัฒนาขึ้นสามารถปรับค่าไฮเปอร์พารามิเตอร์ของแบบจำลองการเรียนรู้เชิงลึกชนิดทรานส์ฟอร์เมอร์สได้อย่างมีประสิทธิภาพกับชุดข้อมูลที่มีทรัพยากรต่ำ

อย่างไรก็ตาม ประสิทธิภาพในการแปลภาษาไทย-อังกฤษให้คะแนน BLEU ที่ต่ำกว่าการแปลภาษาเยอรมัน-อังกฤษ เนื่องจากมีจำนวนคู่ข้อมูลฝึกสอนที่จำกัด ในส่วนของบทสรุป อัลกอริทึมการเพิ่มประสิทธิภาพแบบกลุ่มอนุภาคที่นำเสนอสามารถปรับปรุงเพิ่มเติมได้ และเหมาะสมสำหรับการทดลองกับชุดข้อมูลอื่น ๆ ที่มีทรัพยากรต่ำในอนาคต และสามารถนำแบบจำลองที่ได้ไปใช้งานกับระบบสมองกลฝังตัวหรือเครื่องคอมพิวเตอร์ที่สมถรรนะไม่สูงมากนัก

5.2 ข้อเสนอแนะ

แม้ว่าการทดลองจะประสบผลสำเร็จเป็นที่น่าพอใจแต่การทดลองที่สร้างได้ในงานวิจัยนี้ยังมีข้อจำกัดอยู่คือ ขนาดหน่วยเก็บความจำของเครื่องคอมพิวเตอร์ที่ใช้ในการทดลองหาค่าเหมาะที่สุด ซึ่งถ้ามีขนาดหน่วยเก็บความจำที่มากกว่านี้สามารถทำการทดลองเพื่อไปเปรียบเทียบกับงานวิจัยอื่น ๆ ได้อีก และความซับซ้อนของการค้นหาค่าไฮเปอร์พารามิเตอร์ สามารถยกระดับและเพิ่มตัวแปรในการค้นหาที่มากขึ้น เพื่อผลลัพธ์ที่ดีต่องานวิจัยในอนาคต การปรับปรุงค่าไฮเปอร์พารามิเตอร์ที่ทำการค้นหาด้วยอัลกิริทึมที่พัฒนาขึ้นให้มีช่วงค่าที่กว้างขึ้นและมีค่าในการเลือกมากขึ้น

รายการอ้างอิง

[1] N. Fatima, A. S. Imran, Z. Kastrati, S. M. Daudpota, and A. Soomro, "A systematic literature review on text generation using deep neural network models," *IEEE Access,* vol. 10, pp. 53490-53503, 2022.

[2] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open,* 2022.

[3] S. Ranathunga, E.-S. A. Lee, M. Prifti Skenduli, R. Shekhar, M. Alam, and R. Kaur, "Neural machine translation for low-resource languages: A survey," *ACM Computing Surveys,* vol. 55, no. 11, pp. 1-37, 2023.

[4] B. Bischl *et al.*, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery,* vol. 13, no. 2, p. e1484, 2023.

[5] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing,* vol. 415, pp. 295-316, 2020.

[6] M. Kaveh and M. S. Mesgari, "Application of meta-heuristic algorithms for training neural networks and deep learning architectures: A comprehensive review," *Neural Processing Letters,* vol. 55, no. 4, pp. 4519-4622, 2023.

[7] Y. Q. Wang, J. Y. Li, C. H. Chen, J. Zhang, and Z. H. Zhan, "Scale adaptive fitness evaluation-based particle swarm optimisation for hyperparameter and architecture optimisation in neural networks and deep learning," *CAAI Transactions on Intelligence Technology,* vol. 8, no. 3, pp. 849-862, 2023.

[8] Y. Wang, H. Zhang, and G. Zhang, "cPSO-CNN: An efficient PSO-based algorithm for fine-tuning hyper-parameters of convolutional neural networks," *Swarm and Evolutionary Computation,* vol. 49, pp. 114-123, 2019.

[9] S. Strasser, R. Goodman, J. Sheppard, and S. Butcher, "A new discrete particle swarm optimization algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference* 2016, 2016, pp. 53-60.

[10] J.-P. van Zyl and A. P. Engelbrecht, "Set-based particle swarm optimisation: a Review," *Mathematics,* vol. 11, no. 13, p. 2980, 2023.

[11] D. Jurafsky and J. H. Martin, "Speech and Language Processing: An Introduction

to Natural Language Processing, Computational Linguistics, and Speech Recognition," ed.

[12] À. R. Atrio and A. Popescu-Belis, "Small batch sizes improve training of low-resource neural mt," 2022.

[13] L. Lowphansirikul, C. Polpanumas, A. T. Rutherford, and S. Nutanong, "SCB-MT-EN-TH-2020: A large english-thai parallel corpus," 2020.

[14] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the* 40*th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311-318.

[15] A. Araabi and C. Monz, "Optimizing transformer for low-resource neural machine translation," 2020.

[16] P. R. Lorenzo, J. Nalepa, M. Kawulok, L. S. Ramos, and J. R. Pastor, "Particle swarm optimization for hyper-parameter selection in deep neural networks," in *Proceedings of the genetic and evolutionary computation conference*, 2017, pp. 481-488.

[17] R. Arreerard, S. Mander, and S. Piao, "Survey on Thai NLP language resources and tools," 2022.

[18] E. Kochmar, *Getting started with natural language processing*. Simon and Schuster, 2022.

[19] P. Chartcharnchai, Y. Jewajinda, and K. Praditwong, "A Categorical Particle Swarm Optimization for Hyperparameter Optimization in Low-Resource Transformer-based Machine Translation," in 2024 28*th International Computer Science and Engineering Conference (ICSEC)*, 2024: IEEE, pp. 1-6.

[20] L. Verma and K. N. Kolhatkar, "Optimizing transformer-based machine translation model for single GPU training: a hyperparameter ablation study," 2023.

[21] M. Junczys-Dowmunt *et al.*, "Marian: Fast neural machine translation in C++," 2018.

[22] L. J. Laki and Z. G. Yang, "Neural machine translation for Hungarian," *Acta Linguistica Academica,* vol. 69, no. 4, pp. 501-520, 2022.

มหาวิทยาลัยศิลปากร

มหาวิทยาลัยศิลปากร

ประวัติผู้เขียน

ชื่อ-สกุล	พุฒิสรรค์ ชาติชาญชัย
วุฒิการศึกษา	วิศวกรรมศาสตรบัณฑิต (วศ.บ.) วิศวกรรมอิเล็กทรอนิกส์และระบบคอมพิวเตอร์ มหาวิทยาลัยศิลปากร วิศวกรรมศาสตรมหาบัณฑิต (วศ.ม.) วิศวกรรมไฟฟ้าและคอมพิวเตอร์ มหาวิทยาลัยศิลปากร