บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทวิจัยนี้ผู้วิจัยได้นำเสนอเนื้อหาที่เน้นถึงทฤษฎีและงานวิจัยที่เกี่ยวข้อง รวมถึงเอกสารและงานเขียนอื่น ๆ ที่เกี่ยวข้องกับการวิจัยโดยในบทนี้จะแบ่งเนื้อหาหลัก ๆ ออกเป็น 9 หัวข้อ ประกอบด้วย

2.1 ภาษามือ (Sign Language)

2.2 การเรียนรู้เชิงลึก (Deep Learning)

2.3 โครงข่ายประสาทเทียม (Artificial Neural Networks: ANN)

2.4 โครงข่ายประสาทเทียมแบบวนกลับ (Recurrent Neural Networks: RNN)

2.5 หน่วยความจำระยะสั้นยาว (Long Short-Term Memory: LSTM)

2.6 หน่วยเกทแบบวนกลับ (Gated Recurrent Unit)

2.7 หน่วยความจำระยะสั้นยาวแบบสองทิศทาง (Bidirectional Long Short-Term Memory: BiLSTM)

2.8 ภาษาและเครื่องมือที่ใช้

2.9 งานวิจัยที่เกี่ยวข้อง

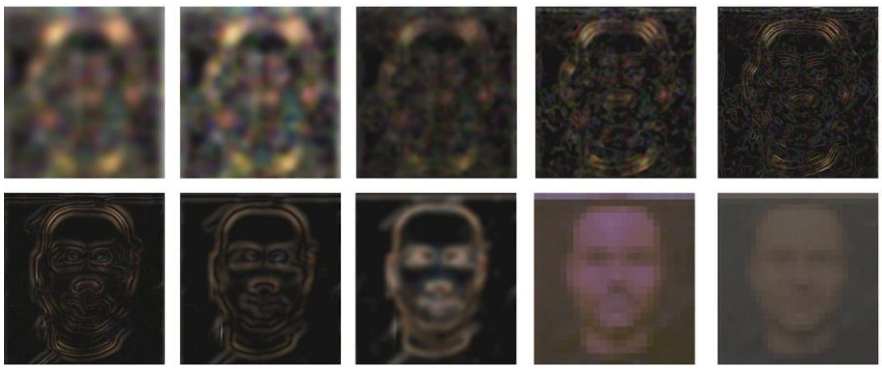
2.1 ภาษามือ (Sign Language)

นักการศึกษาทางด้านการศึกษาของเด็กที่มีความบกพร่องทางการได้ยินตกลงและยอมรับว่าภาษามือเป็นภาษาหนึ่งสำหรับการติดต่อสื่อความหมาย และกรมสามัญศึกษาได้ให้ความหมายของภาษามือไว้ดังนี้

ภาษามือ คือ ภาษาสำหรับคนหูหนวก โดยใช้มือ สีหน้าและกิริยาท่าทางในการประกอบในการสื่อความหมาย และถ่ายทอดอารมณ์แทนการพูด ภาษามือของแต่ละชาติมีความหมายแต่ต่างกันเช่นเดียวกับภาษาพูด ซึ่งแตกต่างกันตามขนบธรรมเนียม ประเพณี วัฒนธรรมและลักษณะภูมิศาสตร์ เช่น ภาษามือจีน ภาษามืออเมริกัน และภาษามือไทย เป็นต้น ภาษามือเป็นภาษาที่นักการศึกษาทางด้านการศึกษาคนหูหนวกตกลงและยอมรับกันแล้วว่าเป็นภาษาหนึ่งสำหรับการติดต่อสื่อความหมายระหว่างคนหูหนวกกับคนหูหนวกด้วยกัน และระหว่างคนปกติกับคนหูหนวก (bkkthon, 2563: Online)

2.2 การเรียนรู้เชิงลึก (Deep Learning)

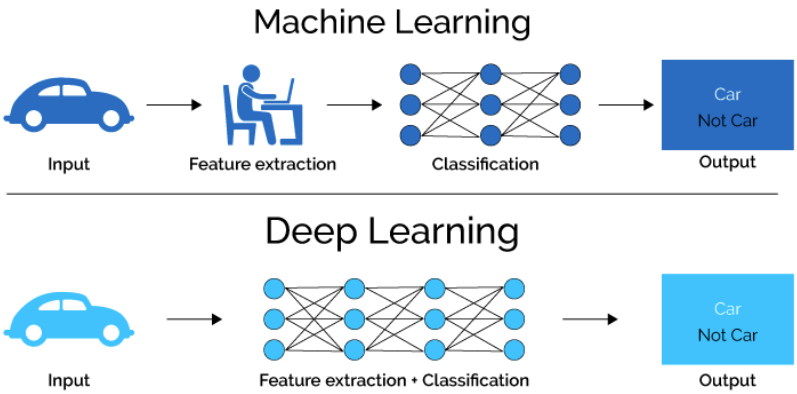
Deep Learning คือวิธีการเรียนรู้แบบอัตโนมัติด้วยการเลียนแบบการทำงานของโครงข่ายประสาทของมนุษย์ (Neuronss) โดยนำระบบโครงข่ายประสาท (Neural Network) มาซ้อนกันหลายชั้น (Layer) และทำการเรียนรู้ข้อมูลตัวอย่าง ซึ่งข้อมูลดังกล่าวจะถูกนำไปใช้ในการตรวจจับรูปแบบ (Pattern) หรือจัด หมวดหมู่ข้อมูล (Classify the Data)



ภาพที่ 2.1 ข้อมูลภาพที่ซ้อนกันหลายชั้นโครงข่าย
ที่มา : Divva Sheel (2565: ออนไลน์)

ตัวอย่างเช่น ภาพที่ 2.1 รูปภาพจากแต่ละชั้นของโครงข่าย ที่จะทำให้เกิดความสามารถ ในการจดจำ เช่น ใบหน้า ซึ่งจะต้องใช้ชั้นของโครงข่าย (Layer) จำนวนมากมายซ้อนกัน จะมีการเรียนรู้ชั้นของข้อมูลตัวอย่างโดยระบบโครงข่าย ประสาท จัดเป็นการเรียนรู้ของเครื่องจักร (Machine Learning) ประเภทหนึ่ง โดยทั่วไประบบโครงข่ายประสาทจะเรียนรู้ได้ เพียงไม่กี่ชั้น เนื่องจากยังไม่มี

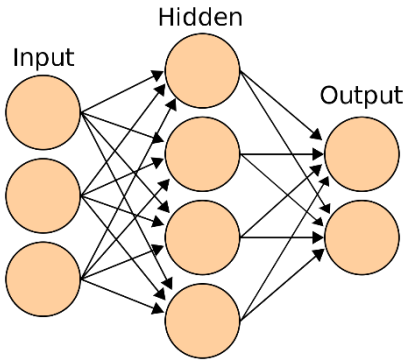
ข้อมูลสอน (Training Data) หรือ ความสามารถด้านคอมพิวเตอร์ยังไม่สูงพอ อย่างไรก็ดี ช่วงหลายปีมานี้ เทคโนโลยีได้มีการพัฒนามากขึ้น จึงทำให้มีข้อมูลชั้นของ โครงข่ายได้ง่ายขึ้นและมากขึ้น ยิ่งมีซ้อนกันหลายชั้น โครงข่ายก็ยิ่ง มีความซับซ้อนและลึกขึ้น จึงเป็นที่มาของคำว่า Deep Learning ตามรูปแบบของ Machine Learning โดยทั่วไป เมื่อมีข้อมูลดิบ เข้ามา จะไม่มีการประมวลโดยอัตโนมัติ แต่จะต้องอาศัยความรู้ เฉพาะทาง (Domain Knowledge) สำหรับคุณลักษณะในการ จัดหมวดหมู่ข้อมูลบางประเภท (Hand-Craft Features) (Divya Sheel, 2565: Online)



ภาพที่ 2.2 ความแตกต่างระหว่าง Machine Learning กับ Deep Learning
ที่มา : Vithan Minaphinant (2565: ออนไลน์)

แต่ถ้าเป็น Deep Learning จะรับข้อมูลดิบเข้าทันที และทำการ ประมวลอัตโนมัติเพื่อหาข้อมูลตัวอย่างที่จำเป็นในการตรวจจับ รูปแบบหรือจัดหมวดหมู่ข้อมูล ความสามารถในการเรียนรู้คุณลักษณะอัตโนมัติทำให้ Deep Learning เป็นประโยชน์อย่างยิ่ง สำหรับการใช้งานในสถานการณ์ต่าง ๆ สิ่งท้าทายที่ยังต้องเผชิญ คือการหาโครงข่ายระบบประสาท ที่เหมาะสมและการค้นหาตัวแปรที่มีผลต่อสมรรถนะในการสอน (Training Performance) ของโครงข่าย ยังคงเป็นเรื่องยากที่จะ รู้ได้ว่า Deep Learning สามารถเรียนรู้คุณลักษณะใดบ้าง นอกจากนี้ Deep Learning ยังมีลักษณะไม่ต่างจาก Machine Learning นั่นคือ ยังไม่สามารถจัดการข้อมูลรับเข้าที่มีความละเอียดเฉพาะทาง (Carefully Crafted Input) จึงอาจทำให้โมเดล เกิดการอนุมานผิดพลาด (Wrong Inferences) ซึ่งประเด็นเหล่านี้ เป็นสิ่งที่นักวิจัยสาขาที่เกี่ยวข้องให้ความสนใจอยู่ เมื่อเร็วๆ นี้ Deep Learning ประสบความสำเร็จอย่างมาก ในด้านการจดจำใบหน้าและคำพูด (Divya Sheel, 2565: Online)

2.3 โครงข่ายประสาทเทียม (Artificial Neural Networks: ANN)



ภาพที่ 2.3 ภาพโครงสร้างโครงข่ายประสาทเทียม
ที่มา : Wikipedia (2022: Online)

โครงข่ายประสาทเทียม (Artificial Neural Networks) หรือที่มักจะเรียกสั้น ๆ ว่า โครงข่ายประสาท (Neural Networks หรือ Neural Net) เป็นหนึ่งในเทคนิคของการทำเหมืองข้อมูล (Data Mining) คือโมเดลทางคณิตศาสตร์ สำหรับประมวลผลสารสนเทศด้วยการคำนวณแบบคอนเนคชันนิสต์ (Connectionist) เพื่อจำลองการทำงานของเครือข่ายประสาทในสมองมนุษย์ ด้วยวัตถุประสงค์ที่จะสร้างเครื่องมือซึ่งมีความสามารถในการเรียนรู้การจดจำรูปแบบ (Pattern Recognition) และการสร้างความรู้ใหม่ (Knowledge Extraction) เช่นเดียวกับความสามารถที่มีในสมองมนุษย์ แนวคิดเริ่มต้นของเทคนิคนี้ได้มาจากการศึกษาโครงข่ายไฟฟ้าชีวภาพ (Bioelectric Network) ในสมอง ซึ่งประกอบด้วย เซลล์ประสาท หรือ "นิวรอน" (Neuronss) และ "จุดประสานประสาท" (Synapses) แต่ละเซลล์ประสาทประกอบด้วยปลายในการรับกระแสประสาท เรียกว่า "เดนไดรท์" (Dendrite) ซึ่งเป็น Input และปลายในการส่งกระแสประสาทเรียกว่า "แอคซอน" (Axon) ซึ่งเป็นเหมือน Output ของเซลล์ เซลล์เหล่านี้ทำงานด้วยปฏิกิริยาไฟฟ้าเคมี เมื่อมีการกระตุ้นด้วยสิ่งเร้าภายนอกหรือกระตุ้นด้วยเซลล์ด้วยกัน กระแสประสาทจะวิ่งผ่านเดนไดรท์เข้าสู่นิวเคลียสซึ่งจะเป็นตัวตัดสินว่าต้องกระตุ้นเซลล์อื่น ๆ ต่อหรือไม่ ถ้ากระแสประสาทแรงพอ นิวเคลียสก็จะกระตุ้นเซลล์อื่น ๆ ต่อไปผ่านทางแอคซอนของมัน นักวิจัยส่วนใหญ่ในปัจจุบันเห็นตรงกันว่าโครงข่ายประสาทเทียมมีโครงสร้างแตกต่างจากโครงข่ายในสมอง แต่ก็ยังเหมือนสมอง ในแง่ที่ว่าโครงข่ายประสาทเทียม คือการรวมกลุ่มแบบขนานของหน่วยประมวลผลย่อย ๆ และการเชื่อมต่อนี้เป็นส่วนสำคัญที่ทำให้เกิดสติปัญญาของโครงข่าย เมื่อพิจารณาขนาดแล้วสมองมีขนาดใหญ่กว่าโครงข่ายประสาทเทียมอย่างมาก รวมทั้งเซลล์ประสาทยังมีความซับซ้อนกว่าหน่วยย่อยของโครงข่าย อย่างไรก็ดีหน้าที่สำคัญของสมอง เช่น การเรียนรู้ยังคงสามารถถูกจำลองขึ้นอย่างง่ายด้วยโครงข่ายประสาทนี้ สำหรับในคอมพิวเตอร์ Neuronss ประกอบด้วย Input และ Output เหมือนกัน โดยจำลองให้ Input แต่ละอันมี Weight เป็น

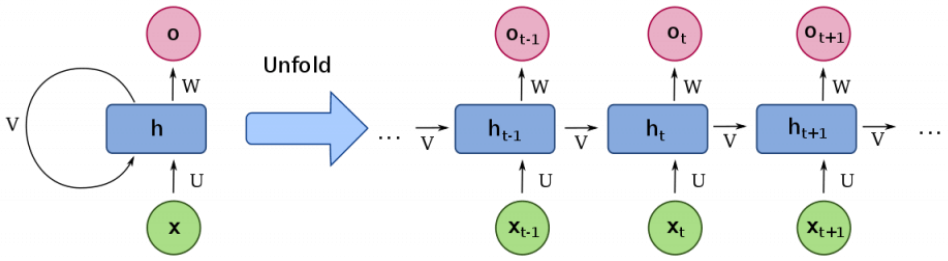
ตัวกำหนดน้ำหนักของ Input โดย Neurons แต่ละหน่วยจะมีค่า Threshold เป็นตัวกำหนดว่าน้ำหนักรวมของ Input ต้องมากขนาดไหนจึงจะสามารถส่ง Output ไปยัง Neuronss ตัวอื่นได้ เมื่อนำ Neurons แต่ละหน่วยมาต่อกันให้ทำงานร่วมกันการทำงานนี้ในทางตรรกแล้วก็จะเหมือนกับปฏิกิริยาเคมีที่เกิดในสมอง เพียงแต่ในคอมพิวเตอร์ทุกอย่างเป็นตัวเลขเท่านั้นเอง การทำงานของ Neural Networks คือเมื่อมี Input เข้ามายัง Network ก็เอา Input มาคูณกับ weight ของแต่ละขา ผลที่ได้จาก Input ทุก ๆ ขาของ Neurons จะเอามารวมกันแล้วก็เอามาเทียบกับ threshold ที่กำหนดไว้ ถ้าผลรวมมีค่ามากกว่า threshold แล้ว Neurons ก็จะส่ง Output ออกไป Output นี้ก็จะถูกส่งไปยัง Input ของ Neurons อื่น ๆ ที่เชื่อมกันใน Network ถ้าค่าน้อยกว่า Threshold ก็จะไม่เกิด Output สิ่งสำคัญคือต้องทราบค่า Weight และ Threshold สำหรับสิ่งที่ต้องการเพื่อให้คอมพิวเตอร์รู้จำ ซึ่งเป็นค่าที่ไม่แน่นอน แต่สามารถกำหนดให้คอมพิวเตอร์ปรับค่าเหล่านั้นได้โดยการสอนให้มันรู้จัก Pattern ของสิ่งที่ต้องการให้มันรู้จำ เรียกว่า "Back Propagation" ซึ่งเป็นกระบวนการย้อนกลับของการรู้จำ ในการฝึก Feed-Forward Neural Networks จะมีการใช้อัลกอริทึมแบบ Back-Propagation เพื่อใช้ในการปรับปรุงน้ำหนักคะแนนของเครือข่าย (Network Weight) หลังจากใส่รูปแบบข้อมูลสำหรับฝึกให้แก่เครือข่ายในแต่ละครั้งแล้ว ค่าที่ได้รับ (Output) จากเครือข่ายจะถูกนำไปเปรียบเทียบกับผลที่คาดหวัง แล้วทำการคำนวณหาค่าความผิดพลาด ซึ่งค่าความผิดพลาดนี้จะถูกส่งกลับเข้าสู่เครือข่ายเพื่อใช้แก้ไขค่าน้ำหนักคะแนนต่อไป การเรียนรู้สำหรับ Neural Networks มีอยู่ 2 ประเภทได้แก่

1) Supervised Learning การเรียนแบบมีการสอน เป็นการเรียนแบบที่มีการตรวจคำตอบเพื่อให้โครงข่ายประสาทเทียมปรับตัว ชุดข้อมูลที่ใช้สอนโครงข่ายประสาทเทียมจะมีคำตอบไว้คอยตรวจดูว่าโครงข่ายประสาทเทียมให้คำตอบที่ถูกหรือไม่ ถ้าตอบไม่ถูกโครงข่ายประสาทเทียมก็จะปรับตัวเองเพื่อให้ได้คำตอบที่ดีขึ้น (เปรียบเทียบกับคน เหมือนกับการสอนนักเรียนโดยมีครูผู้สอนคอยแนะนำ)

2) Unsupervised Learning การเรียนแบบไม่มีการสอน เป็นการเรียนแบบไม่มีผู้แนะนำ ไม่มีการตรวจคำตอบว่าถูกหรือผิด โครงข่ายประสาทเทียมจะจัดเรียงโครงสร้างด้วยตัวเองตามลักษณะของข้อมูล ผลลัพธ์ที่ได้ โครงข่ายประสาทเทียมจะสามารถจัดหมวดหมู่ของข้อมูลได้ (เปรียบเทียบกับคน เช่น การที่สามารถแยกแยะพันธุ์พืช พันธุ์สัตว์ตามลักษณะรูปร่างของมันได้เองโดยไม่มีใครสอน) (วิทยา พรพัชรพงศ์, 2565: ออนไลน์)

2.4 โครงข่ายประสาทเทียมแบบวนกลับ (Recurrent Neural Networks: RNN)

โครงข่ายประสาทเทียมแบบวนกลับ (Recurrent Neural Networks: RNN) เป็นวิธีการที่ถูกนำมาใช้ในการวิจับเกี่ยวกับการรู้จำเสียง (Speech Recognition) และการประมวลผลภาษาธรรมชาติ (Natural Language Processing) การทำงานของ RNN คือการนำผลลัพธ์ที่ได้จากการคำนวณย้อนกลับมาใช้เป็นข้อมูลขาเข้าอีกครั้ง ซึ่งมีประโยชน์อย่างมากในข้อมูลที่มีความต่อเนื่อง เช่น ข้อมูลเสียง ข้อความ หรือแม้แต่รูปภาพเองก็ตาม



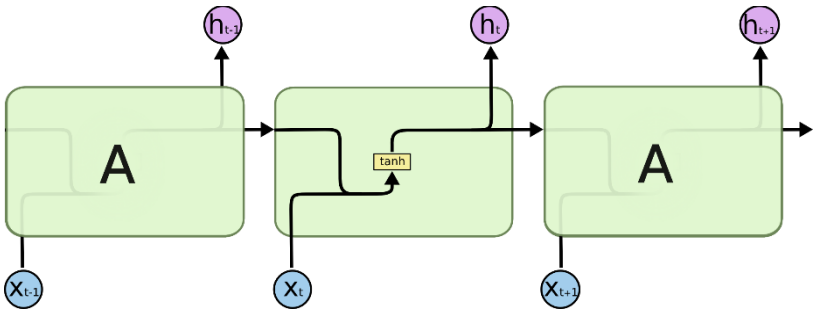
ภาพที่ 2.4 การทำงานของ RNN
ที่มา : bualabs (2565: ออนไลน์)

RNN ถูกออกแบบมาเพื่อแก้ปัญหาสำหรับงานที่มีข้อมูลที่มีลำดับ โดยใช้หลักการนำสถานะภายในของโมเดล กลับมาเป็นข้อมูลเข้าใหม่คู่กับข้อมูลเข้าแบบปกติ เรียกว่า สถานะซ่อน (Hidden State) หรือสถานะภายใน (Internal State) ช่วยให้โมเดลรู้จำรูปแบบ ของลำดับข้อมูลนำเข้า (Input Sequence) ได้แสดงดังรูปที่ 2.4

ในแต่ละโหนดของ RNN จะมีข้อมูลเข้าสองอย่าง ได้แก่ 1) ข้อมูลเข้า ณ โหนดนั้น ๆ และ 2) ผลลัพธ์ที่ได้จากการคำนวณในโหนดก่อนหน้า ซึ่งทั้งสองข้อมูลจะถูกนำมารวมเข้าด้วยกันและออกผลลัพธ์มาเป็นสองทางคือ 1) ผลลัพธ์ที่ออกมา ณ โหนดนั้น ๆ และออกเพื่อไปเป็นข้อมูลขาเข้าในโหนดถัดไป ข้อดีของ RNN คือ มีการใช้ข้อมูลก่อนหน้าในการทำนายสิ่งที่อาจจะเกิดขึ้นในอนาคต ซึ่งหมายถึงอะไรที่เคยเกิดขึ้นในอดีตย่อมส่งผลต่อเหตุการณ์ที่จะเกิดขึ้นในอนาคตด้วย แม้ RNN จะมีข้อดีในการทำงานของข้อมูลที่มีความต่อเนื่อง แต่ข้อเสียของ RNN คือ สามารถดูย้อนกลับได้แค่เพียงในช่วงระยะเวลาสั้น ๆ เท่านั้น ซึ่งปัญหาหลัก ๆ ของ RNN เกิดมาจากเกรเดียนท์ที่เริ่มน้อยลงในข้อมูลที่มีความยาวขึ้น ปัญหาการสูญเสียของเกรเดียนท์ (Vanishing Gradient Problem: VGP) ซึ่งปัญหานี้ถูกแก้ไขโดยใช้เกทแบบวนกลับ (Gated Recurrent Unit: GRU) และหน่วยความจำระยะสั้นยาว (Long Short-Term Memory: LSTM) (csit, 2565: Online)

2.5 หน่วยความจำระยะสั้นยาว (Long Short-Term Memory: LSTM)

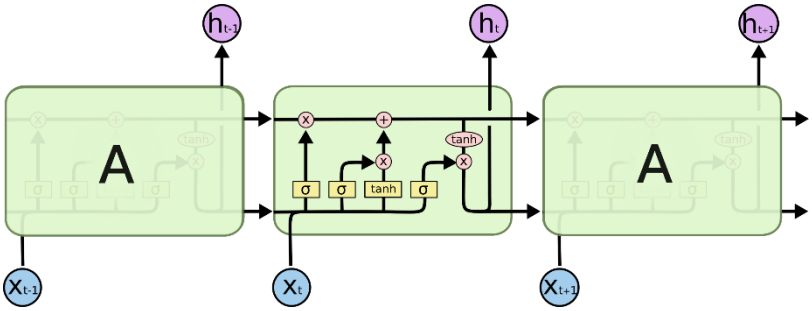
Long Short-Term Memory Model (LSTM) เป็นเทคนิคหนึ่งที่ถูกพัฒนาจาก Recurrent Neural Network (RNN) ซึ่ง RNN นั้นมีหลักการทำงาน คือการนำ Output (ผลลัพธ์) ที่ได้จากการคำนวณจากโหนดก่อนหน้านี้กลับมาใช้เป็นข้อมูล Input ที่ผ่านการคำนวณจากโหนดก่อนหน้า โดยข้อมูลทั้ง 2 ชุดที่เข้ามาในโหนดจะถูกรวมเข้าด้วยกันก่อนจะถูกแยกผลลัพธ์ออกเป็น 2 ส่วนคือ ผลลัพธ์ที่ได้จากโหนดนั้น ๆ และผลลัพธ์ที่จะถูกนำไปเป็นข้อมูล Input ของโหนดถัดไป เทคนิค RNNs นั้นเหมาะนำมาใช้งานกับข้อมูลที่มีลักษณะเป็นลำดับ (Sequence) หรือข้อมูลที่มีความต่อเนื่อง เช่น ข้อมูลอนุกรมเวลา (Time Series) ข้อมูลเสียง, ข้อมูลประเภทข้อความ, ข้อมูลประเภทรูปภาพและวิดิทัศน์ เป็นต้น



ภาพที่ 2.5 โครงสร้าง RNN

ที่มา : Christopher Olah (2022: Online)

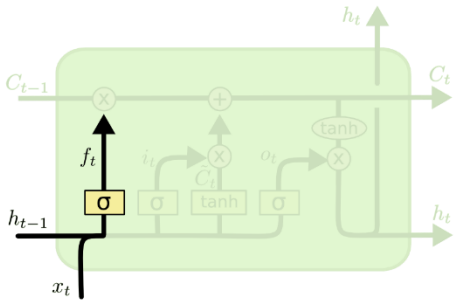
ข้อดีของ RNN คือสามารถนำข้อมูลก่อนหน้า (ในอดีต) มาใช้ในการทำนายสิ่งที่อาจจะเกิดขึ้นในอนาคตได้ ส่วนข้อเสียของ RNN คือ จะสามารถดูข้อมูลย้อนหลังได้เพียงแค่ระยะสั้น ๆ เท่านั้น ซึ่งทำให้เกิดปัญหาในการทำ Backpropagation หรือการคำนวณหาความผิดพลาดย้อนหลังของแต่ละโหนดเมื่อสิ้นสุดการทำงาน เพราะการ Backpropagation นั้นจะต้องทำย้อนกลับไปหลายขั้นตอนและหลายโหนด จึงทำให้เกิดปัญหา Vanishing Gradient Problem ดังนั้นเพื่อแก้ปัญหาดังกล่าวจึงทำให้เกิดเทคนิค LSTM ขึ้น



ภาพที่ 2.6 โครงสร้าง LSTM

ที่มา : Christopher Olah (2022: Online)

Long Short-Term Memory (LSTM) เป็นโครงค่ายประสาทเทียมประเภท RNNs รูปแบบหนึ่งที่ถูกพัฒนาขึ้นมาให้มีความเสถียรและมีประสิทธิภาพมากขึ้น LSTM เริ่มเป็นที่รู้จักในปี ค.ศ. 1997 โดย Hochreiter และ Schmidhuber (Hochreiter & Schmidhuber. 1997) โดยมีหลักการทำงานคือ สามารถเก็บ ‘สถานะ’ หรือข้อมูลของแต่ละโหนดเอาไว้เพื่อที่เวลาย้อนกลับมาดูจะได้ทราบถึงที่ของข้อมูลดังกล่าวว่าเดิมเป็นค่าอะไรและจุดเด่นของเทคนิค LSTM คือฟังก์ชันพิเศษที่มีหน้าทีเหมือน ‘ประตู (Gate)’ ที่คอยควบคุมข้อมูลที่จะเข้ามาในแต่ละโหนด ซึ่งประกอบไปด้วย Forget Gate Layer, Input Gate และ Output Gate Layer



ภาพที่ 2.7 ภาพโครงสร้าง Forget Gate Layer

ที่มา : Christopher Olah (2022: Online)

เป็น Gate ที่มีหน้าที่ในการกำหนดว่าข้อมูลที่เข้ามาใน Cell State นั้นควรจะถูกเก็บไว้หรือควรที่จะทิ้งไป ซึ่งข้อมูลที่ถูกตัดสินว่าควรเก็บไว้นั้นจะถูกประเมินจากข้อมูล Input ที่เข้ามาในโหนดนั้น ๆ รวมกับผลลัพธ์ที่ได้จากการคำนวณของโหนดก่อนหน้าผ่านฟังก์ชัน Sigmoid ดังสมการต่อไปนี้

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

จากสมการที่กล่าวมา

f_t คือ Forget Gate

σ คือ ฟังก์ชัน Sigmoid

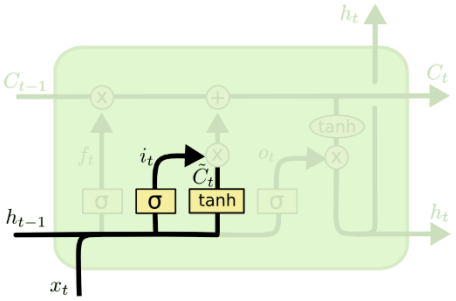
w_f คือ ค่าน้ำหนักของ Matrices

h_{t-1} คือ ค่า Output ของ Cell State ก่อนหน้า (ที่ timestamp t-1)

x_t คือ ค่า Input ที่เข้ามาใน Cell State ณ เวลา t

b_f คือ ค่า Bias

ผลลัพธ์ที่ได้จาก Forget Gate Layer จะอยู่ระหว่างค่า 0 และ 1 ซึ่งถ้าได้ค่าเป็น 0 นั้นหมายถึงให้ลบค่า Cell State เดิมออก แต่ถ้าได้ค่าเป็น 1 นั้นหมายถึงให้เก็บค่า Cell State นี้ต่อไป



ภาพที่ 2.8 ภาพโครงสร้าง Input Gate

ที่มา : Christopher Olah (2022: Online)

เป็น Gate ที่มีหน้าที่รับข้อมูล Input เข้ามาใหม่แล้วจึงทำการบันทึกหรือ ‘เขียน (write)’ ข้อมูลลงไปในแต่ละโหนด โดยมีการทำงานแบ่งออกเป็น 2 ส่วน โดยส่วนแรกคือถ้าต้องการ Update Cell State เมื่อทำการรับข้อมูล Input เข้ามาแล้วฟังก์ชัน Sigmoid ที่เป็นตัวควบคุมจะเรียกใช้ Input Gate เพื่อเลือกว่าจะให้ Update Cell State ฟังก์ชัน Tanh ก็จะทำการสร้าง Candidate Values ($\tilde{c}_t$) ขึ้นมาใน State ดังสมการ $i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i)$ และ สมการ $\tilde{c}_t = tanh(w_C \cdot [h_{t-1}, x_t] + b_C)$

จากสมการที่กล่าวมา

i_t คือ Input Gate

σ คือ ฟังก์ชัน Sigmoid

$\tilde{c}_t$ คือ ค่า Candidate ของ Cell State ที่เวลา t

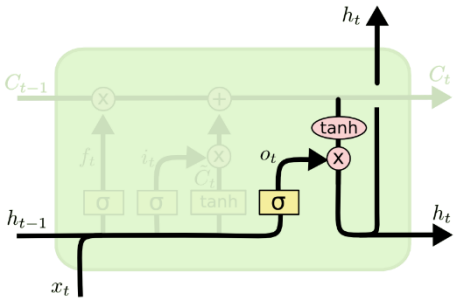
$tanh$ คือ ฟังก์ชัน tanh

w_i, w_c คือ ค่าน้ำหนักของ Matrices

h_{t-1} คือ ค่า Output ของ Cell State ก่อนหน้า (ที่ timestamp t-1)

x_t คือ ค่า Input ที่เข้ามาใน Cell State ณ เวลา t

b_i, b_c คือ ค่า Bias



ภาพที่ 2.9 ภาพโครงสร้าง Output Gate Layer

ที่มา : Christopher Olah (2022: Online)

เป็น Gate ที่มีหน้าที่เตรียมทำการส่งข้อมูล (Output Data) โดยข้อมูลที่จะทำหการ Output นั้นจะดูจาก Cell State ที่ผ่านกระบวนการคำนวณต่าง ๆ แล้วโดยฟังก์ชัน Sigmoid จะเป็นตัวเลือกว่าข้อมูลส่วนไหจนใน Cell State ที่จะถูก Output จากนั้นจะนำค่า Cell State เข้าฟังก์ชัน tanh (เพื่อหาว่าค่าจะได้ออกมาเป็น 1 หรือ -1) แล้วนำค่าที่ได้จากฟังก์ชัน tanh มาทำการคำนวณกับค่า Output ที่ได้จาก Sigmoid Gate จากนั้นก็จะได้ค่า Output ที่ต้องการดังสมการต่อไปนี้

$$o_t = \sigma(w_0 \cdot [h_{t-1}, x_t] + b_0)$$ และสมการ $$h_t = o_t * \tanh(C_t)$$

จากสมการที่กล่าวมา

o_t คือ Output Gate

σ คือ ฟังก์ชัน Sigmoid

W_o คือ ค่าน้ำหนักของ Matrices

h_{t-1} คือ ค่า Output ของ Cell State ก่อนหน้า (ที่ timestamp t-1)

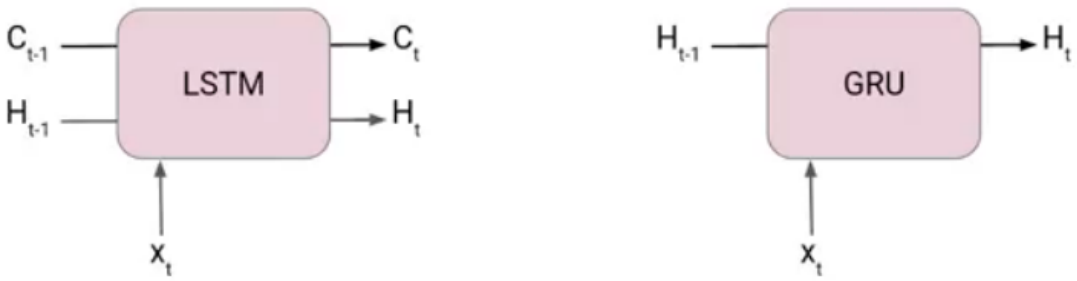
x_t คือ ค่า Input ที่เข้ามาใน Cell State ณ เวลา t

b_o คือ ค่า Bias

ซึ่งค่า Output ที่ได้ออกมานั้นจะถูกแบ่งออกเป็น 2 ส่วน คือ ค่า Output ที่ได้จากโหนดนั้นกับค่า Output ที่จะถูกส่งไปเป็นข้อมูล Input ของโหนดถัดไป (กานต์กมล ทวีผล, 2022)

2.6 หน่วยเกทแบบวนกลับ (Gated Recurrent Unit: GRU)

หน่วยเกทแบบวนกลับนั้นมีความคล้ายคลึงกับ Long Short-Term Memory (LSTM), GRU จะใช้เกทเพื่อควบคุมการไหลของข้อมูล ซึ่งเป็นอะไรที่แปลกเมื่อเทียบกับ LSTM และเป็นเหตุผลที่เสนอการปรับปรุงบางอย่างที่เหนือ LSTM และมีสถาปัตยกรรมที่เรียบง่ายกว่า



ภาพที่ 2.10 ความแตกต่างระหว่าง LSTM และ GRU
ที่มา : analyticsvidhya (2023: Online)

สิ่งที่น่าสนใจอีกอย่างเกี่ยวกับ GRU คือไม่มีสถานะของเซลล์ (Ct) ซึ่งแตกต่างจาก LSTM จะมีเพียง Hidden State (Ht) เนื่องจากสถาปัตยกรรมที่เรียบง่าย GRU จึงเทรนได้ง่ายกว่า LSTM ในแต่ละ Timestamp t จะรับ Input Xt และ Hidden State Ht-1 จาก Timestamp ก่อนหน้า t-1 หลังจากนั้นจะแสดง Hidden State Ht ใหม่ ซึ่งจะส่งต่อไปยัง Timestamp อีกครั้ง ขณะนี้สองเกทหลักใน GRU แทนที่จะเป็นสามเกทในเซลล์ LSTM เกทแรกคือประตูรีเซ็ตและอีกประตูคือประตูอัพเดท

เกทรีเซ็ต (Reset Gate Short Term memory) รีเซ็ตเกทจะรับผิดชอบหน่วยความจำระยะสั้นของเครือข่าย เช่น Hidden State (Ht) ซึ่งสมการของรีเซ็ตเกทคือ

$$r_t = \sigma(X_t * U_r + H_{t-1} * W_r)$$

ซึ่งจะมีความคล้ายกับสมการของ LSTM เกท ค่าของ rt จะอยู่ในช่วงตั้งแต่ 0 ถึง 1 เนื่องจากฟังก์ชัน Sigmoid, Ur และ Wr เป็นเมทริกซ์น้ำหนักสำหรับประตูรีเซ็ต

เกทอัพเดท (Update Gate Long Short Term Memory) ก็จะคล้ายกับสมการของ เกทรีเซ็ต แต่จะมีข้อแตกต่างคือการวัดน้ำหนัก เช่น Uu และ Wu ดังสมการต่อไปนี้

$$U_t = \sigma(X_t * U_u + H_{t-1} * W_u)$$

การทำงานของเกท หากต้องการหา Hidden State ใน GRU จำเป็นจะต้องมี Candidate Hidden State ดังสมการต่อไปนี้

$$H\hat{}_t = tanh(x_t * U_g + (r_t \cdot H_{t-1}) * W_g)$$

ซึ่งจะเป็นการรับ Input และ Hidden State จาก Timestamp t-1 x Output เกทรีเซ็ต rt หลังจากนั้นจะส่งข้อมูลทั้งหมดไปยังฟังก์ชัน Tanh ค่าผลลัพธ์คือ Candidate Hidden State ส่วนที่สำคัญที่สุดของสมการนี้คือวิธีที่ใช้หาค่าของเกทรีเซ็ตเพื่อควบคุมสถานะว่า Hidden State ก่อนหน้านี้จะมีผลต่อ Candidate Hidden State มากน้อยเพียงใด หากค่าของ rt เท่ากับ 1 หมายความว่าข้อมูลทั้งหมดจาก Hidden State Ht-1 ก่อนหน้านี้กำลังถูกพิจารณา ในขณะเดียวกันถ้าค่าของ rt เป็น 0 หมายความว่าข้อมูลจาก Hidden State จะถูกปัดทิ้งทันที

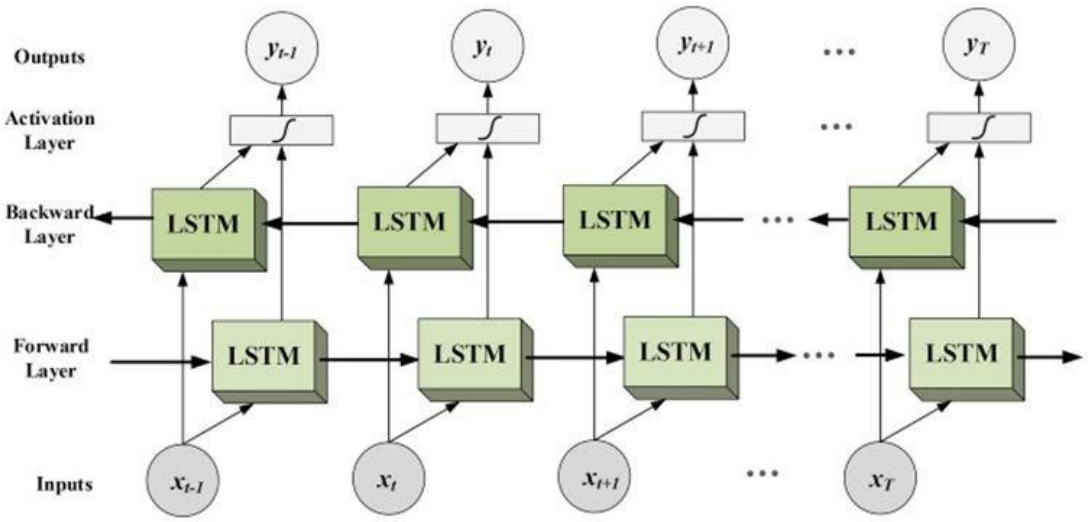
Hidden State เมื่อมี Candidate Hidden State ใช้เพื่อสร้าง Hidden State ปัจจุบันเป็นที่ที่เกทอัพเดทดังสมการ

$$H_t = u_t \cdot H_{t-1} + (1 - u_t) \cdot H\hat{}_t$$

GRU ใช้เกทอัพเดทเดียวเพื่อควบคุมทั้งข้อมูลประวัติซึ่งเป็น Ht-1 ตลอดจนข้อมูลใหม่ที่มาจาก Candidate State สมมติให้ค่าของ ut อยู่ที่ 0 จากนั้นเทอมแรกในสมการจะหายไป ซึ่งหมายความว่า Hidden State ใหม่จะมีไม่มีข้อมูลมากจาก Hidden State ก่อนหน้านี้ ในทางกลับกันส่วนที่สองแทบจะกลายเป็นส่วนเดียว ซึ่งโดยหลักแล้วหมายถึงหมายความว่า Hidden State ที่ Timestamp ปัจจุบันจะมีแค่ข้อมูลจาก Candidate Hidden State เท่านั้น แต่หากค่า ut อยู่ในเทอมที่สองจะกลายเป็น 0 ทั้งหมดและ Hidden State ปัจจุบันจะขึ้นอยู่กับเทอมแรกทั้งหมด นั่นคือข้อมูลจาก Hidden State ที่ Timestamp t-1 ก่อนหน้า ดังนั้นจึงสามารถสรุปได้ว่าค่าของ ut มีความสำคัญอย่างยิ่งในสมการนี้ และมีค่าตั้งแต่ 0 ถึง 1

2.7 หน่วยความจำระยะสั้นยาวแบบสองทิศทาง (Bidirectional Long Short-Term Memory: BiLSTM)

หน่วยความจำระยะสั้นยาวแบบสองทิศทาง เป็นกระบวนการสร้างเครือข่ายประสาทที่มีข้อมูลลำดับทั้งสองทิศทางย้อนหลัง (จากอนาคตไปยังอดีต) หรือไปข้างหน้า (จากอดีตไปยังอนาคต) ในแบบสองทิศทาง อินพุตจะไหลในสองทิศทาง ทำให้ BiLSTM แตกต่างจาก LSTM ปกติ เนื่องจาก LSTM แบบปกติจะทำให้อินพุตไหลไปในทิศทางเดียว ไม่ว่าจะย้อนกลับหรือไปข้างหน้า อย่างไรก็ตาม ในแบบสองทิศทางจะทำให้อินพุตไหลได้ทั้งสองทิศทางเพื่อรักษาข้อมูลในอนาคตและข้อมูลในอดีต ยกตัวอย่างเช่นในประเช่น “Boys go to .....” จะไม่สามารถเติมช่องว่างได้ แม้กระนั้น หากมีประโยคในอนาคตว่า “Boy come out of school” ทำให้สามารถทำนายพื้นที่ว่างในอดีตได้อย่างง่ายดาย ซึ่งสิ่งที่คล้ายกันที่ต้องดำเนินการโดยแบบจำลองแบบ BiLSTM แบบสองทิศทางช่วยทำให้โครงข่ายประสาทเทียมสามารถดำเนินการนี้ได้



ภาพที่ 2.11 โครงสร้าง BiLSTM

ที่มา : analyticsindiamag (2023: Online)

2.8 ภาษาและเครื่องมือที่ใช้

2.8.1 TensorFlow

ภาพที่ 2.12 Tensorflow

ที่มา : Tensorflow (2022: Online)

Tensorflow ก็คือ deep learning library ของ Google ที่กำลังเป็นดาวเด่นอยู่ในตอนนี้, โดยทาง Google ก็ได้ใช้ machine learning เพิ่มประสิทธิภาพกับผลิตภัณฑ์มากมาย ไม่ว่าจะเป็น เครื่องมือค้นหา (Search Engine), การแปลภาษา (Translation), คำบรรยายภาพ (Image Captioning) และ เครื่องมือช่วยการเสนอแนะ (Recommendations) เพื่อช่วยให้เห็นภาพมากขึ้น Google นำ AI มาช่วยให้พัฒนาประสบการณ์ของผู้ใช้ ทั้งในแง่ความเร็วของผลลัพธ์ และ ในแง่ผลลัพธ์ที่ถูกต้องแม่นยำมากขึ้น อย่างเช่น ถ้าลองพิมพ์คำอะไรลงไปในช่องค้นหาล่ะก็ Google สามารถแนะนำคำต่อไป หรือคำที่สมบูรณ์ให้ได้ทันทีเลย Google ต้องการใช้ประโยชน์จาก Machine Learning กับชุดข้อมูลขนานใหญ่ เพื่อให้ผู้ใช้มีประสบการณ์การใช้งานที่ดีที่สุด โดยมีกลุ่มผู้ใช้เทคโนโลยีตัวนี้ราว ๆ 3 กลุ่มด้วยกันโปรแกรมเมอร์, นักวิจัยและนักวิทยาศาสตร์ข้อมูลโดยที่กลุ่มคนทั้งสามกลุ่มสามารถใช้เครื่องชุดเดียวกัน มาพัฒนาต่อหรือปรับปรุงประสิทธิภาพได้ตามต้องการ Tensorflow สร้างมาเพื่อใช้งานได้บนหลากหลายอุปกรณ์ Tensorflow เป็นหนึ่งในผลงานพัฒนาจาก Google Brain Team ทีมที่ถูกตั้งขึ้นมาเพื่อพัฒนา Machine Learning และ Deep Learning โดยเฉพาะ (thaiprogrammer.org, 2022: Online)

2.8.2 OpenCV

ภาพที่ 2.13 OpenCV

ที่มา : Wikipedia (2022: Online)

OpenCV (Open source Computer Vision) เป็นไลบรารีฟังก์ชันการเขียนโปรแกรม (Library of Programming Functions) โดยส่วนใหญ่จะมุ่งเป้าไปที่การแสดงผลด้วยคอมพิวเตอร์แบบเรียลไทม์ (Real-Time Computer Vision) เดิมทีแล้วถูกพัฒนาโดย Intel แต่ภายหลังได้รับการสนับสนุนโดย Willow Garage ตามมาด้วย Itseez (ซึ่งต่อมาถูกเข้าซื้อโดย Intel) OpenCV เป็นไลบรารีแบบข้ามแพลตฟอร์ม (Cross-Platform) และใช้งานได้ฟรีภายใต้ลิขสิทธิ์ของ BSD แบบโอเพ่นซอร์ส (Open-Source BSD License) OpenCV ยังสนับสนุน Frame Work การเรียนรู้เชิงลึก (Deep Learning Frameworks) ได้แก่ TensorFlow, Torch/PyTorch และ Caffe โดย OpenCV ถูกเขียนขึ้นด้วยภาษา C++ มีการรองรับ Python, Java และ MATLAB/OCTAVE API สำหรับ Interface เหล่านี้สามารถพบได้ในเอกสารออนไลน์ ซึ่งมีการรวมไว้หลากหลายภาษา เช่น C#, Perl, Ch, Haskell และ Ruby ได้รับการพัฒนาเพื่อส่งเสริมการนำมาใช้งานโดยผู้ใช้ที่เพิ่มขึ้น (Nuttakan Chuntra, 2565: ออนไลน์)

2.8.3 MediaPipe

ภาพที่ 2.14 MediaPipe

ที่มา : Priyanshu Kumar (2022: Online)

MediaPipe Holistic คือโทโปโลยีล้ำสมัยที่สามารถตรวจจับท่าทาง มือ และใบหน้าของมนุษย์ในเวลาเดียวกัน และรองรับการใช้งานในแบบที่ไม่เคยมีแพลตฟอร์มไหนทำได้มาก่อน โซลูชันนี้จะใช้ Pipeline แบบใหม่ที่ประกอบด้วยการตรวจจับท่าทาง หน้า และมือที่ปรับแต่งให้ดีที่สุดเพื่อให้ทำงานได้เรียลไทม์ โดยใช้การโอนถ่ายหน่วยความจำระหว่าง Interference Backend ซึ่ง Pipeline จะรวมรูปแบบการปฏิบัติการและการประมวลผลที่แตกต่างกันตามการตรวจจับภาพแต่ละส่วนเข้าด้วยกัน และจะได้เป็นโซลูชันแบบครบวงจรที่ใช้งานได้แบบเรียลไทม์และสม่ำเสมอ ซึ่งใช้การทำงานแลกเปลี่ยนกันระหว่างการตรวจจับทั้งสามจุด โดยประสิทธิภาพของการทำงานจะขึ้นอยู่กับความรวดเร็วและคุณภาพของการแลกเปลี่ยนข้อมูล เมื่อรวมการตรวจจับทั้งสามเข้าด้วยกัน จะได้เป็นโทโปโลยีที่ทำงานร่วมกันเป็นหนึ่งเดียว โดยสามารถจับ Key Points ของภาพเคลื่อนไหวได้ถึง 540+ จุด (ส่วนของท่าทาง 33 จุด มือข้างละ 21 จุด และส่วนใบหน้า 468 จุด) ซึ่งเป็นระดับที่ไม่เคยทำได้มาก่อน และสามารถประมวลผลได้เกือบจะเรียลไทม์ในการแสดงผลทางโทรศัพท์มือถือ โดยรองรับการใช้งานทั้งในโทรศัพท์มือถือ (ทั้งระบบ Android และ iOS) และบนคอมพิวเตอร์ นอกจากนี้ Google ยังเปิดให้ใช้ MediaPipe APIs แบบพร้อมใช้งาน สำหรับการใช้งานกับ Python และ JavaScript เพื่อทำให้เทคโนโลยีนี้เข้าถึงได้ง่ายมากขึ้น (Sertis, 2565: ออนไลน์)

2.8.4 Keras

ภาพที่ 2.15 Keras
ที่มา : Keras (2022: Online)

Keras เป็นไลบรารีโอเพนซอร์ซของภาษาไพทอนสำหรับการพัฒนาโครงข่ายประสาทเทียม สามารถทำงานบน TensorFlow, Microsoft Cognitive Toolkit, R, Theano, หรือ PlaidML ได้ เคราสถูกออกแบบมาให้ผู้ใช้สามารถพัฒนาโปรแกรมด้วยการเรียนรู้เชิงลึกได้อย่างรวดเร็ว จึงใช้งานง่าย มีฟังก์ชันให้เลือกหลากหลาย ทำงานเป็นสัดเป็นส่วน ซึ่งถูกพัฒนาขึ้นโดยฟรองซัวส์ ชอลเลต์ วิศวกรของกูเกิล โดยในปี ค.ศ. 2017 ทีมพัฒนา TensorFlow ของกูเกิลเริ่มนำไลบรารีหลักไปสนับสนุนเคราส ชอลเลต์อธิบายว่าเคราสเป็นเหมือนส่วนต่อประสานมากกว่าเป็นเฟรมเวิร์กเดี่ยวๆสำหรับการเรียนรู้ของเครื่อง เคราสมีฟังก์ชันระดับสูงที่เข้าใจง่าย ทำให้การพัฒนาโมเดลด้วยการเรียนรู้เชิงลึกทำได้ง่าย (Wikipedia, 2565: ออนไลน์)

2.8.5 ภาษา Python

ภาพที่ 2.16 Python

ที่มา : Wikipedia (2022: Online)

Python เป็นภาษาการเขียนโปรแกรมที่ใช้อย่างแพร่หลายในเว็บแอปพลิเคชัน การพัฒนาซอฟต์แวร์ วิทยาศาสตร์ข้อมูล และแมชชีนเลิร์นนิง (ML) นักพัฒนาใช้ Python เนื่องจากมีประสิทธิภาพ เรียนรู้ง่าย และสามารถทำงานบนแพลตฟอร์มต่าง ๆ ได้มากมาย ทั้งนี้ซอฟต์แวร์ Python สามารถดาวน์โหลดได้ฟรี ผสานการทำงานร่วมกับระบบทุกประเภท และเพิ่มความเร็วในการพัฒนา ข้อดีต่างๆ ของ Python เช่น นักพัฒนาสามารถอ่านและทำความเข้าใจโปรแกรม Python ได้อย่างง่ายดาย เนื่องจากมีไวยากรณ์พื้นฐานเหมือนภาษาอังกฤษ Python ทำให้นักพัฒนาทำงานได้อย่างมีประสิทธิภาพมากขึ้น เนื่องจากพวกเขาสามารถเขียนโปรแกรม Python ได้โดยใช้โค้ดน้อยลงเมื่อเปรียบเทียบกับภาษาอื่นๆ อีกมากมาย Python มีไลบรารีมาตรฐานขนาดใหญ่ที่มีโค้ดที่ใช้ซ้ำได้สำหรับเกือบทุกงาน ด้วยเหตุนี้ นักพัฒนาจึงไม่ต้องเขียนโค้ดขึ้นใหม่ทั้งหมด (Aws, 2565: ออนไลน์)

2.8.6 โปรแกรม Anaconda

ภพที่ 2.17 Anaconda

ที่มา : Wikipedia (2022: Online)

Anaconda ถือว่ามีความโดดเด่นมาก ไม่เพียงแต่ Data Science และ Machine Learning เท่านั้น แต่สำหรับวัตถุประสงค์อื่นๆ เกี่ยวกับ Python Development ด้วย โดย Anaconda ช่วยให้คุณเข้าถึง Package เกี่ยวกับ Data Science ที่ถูกใช้งานบ่อยๆ เช่น NumPy, Pandas, Matplotlib และอื่นๆ อีกมากมาย โดยสามารถใช้ผ่านการ Custom Package Management System ที่เรียกว่า Conda ซึ่งใน Conda-installed Packages ยังรวมไปถึง Binary Dependencies ที่ไม่สามารถจัดการผ่าน Pip ของ Python ได้ (แต่คุณยังคงสามารถใช้ Pip ได้หากว่าต้องการ) แต่ละ Package จะถูก update อยู่เสมอโดย Anaconda และจะถูก Compile ด้วย Intel MKL extensions เพื่อความรวดเร็ว (techstarthailand, 2565: ออนไลน์)

2.9 งานวิจัยที่เกี่ยวข้อง

Gerges H. Samaan, Abanoub R. Widie, Abanoub K. Attia, Abanoub M. Asaad, Andrew E. Kamel, Salwa O. Slim, Mohamed S. Abdallah and Young-Im Cho (2022) ในงานวิจัยนี้ได้ใช้ MediaPipe ในการเชื่อมเข้ากับ RNN โมเดล เพื่อแก้ปัญหาการรู้จำภาษามือแบบไดนามิก MediaPipe ถูกใช้เพื่อสร้าง Landmarks บนร่างกายแล้วสกัด Keypoints ของมือ ตัวและหน้าส่วน RNN โมเดล เช่น GRU, LSTM และ BiLSTM ถูกใช้เพื่อการรู้จำภาษามือ เนื่องจากไม่มีชุดข้อมูลภาษามือ จึงได้สร้าง DSL 10 Dataset ซึ่งมีคำศัพท์ 10 คำที่ซ้ำกัน 75 ครั้งโดยผู้ลงนาม 5 คนซึ่งให้คำแนะนำขั้นตอนในการสร้างคำศัพท์ดังกล่าว มีการทดลองสองครั้งในชุดข้อมูล DSL 10 Dataset โดยใช้แบบจำลอง RNN เพื่อเปรียบเทียบความแม่นยำของการรู้จำภาษามือแบบไดนามิกที่มีและไม่มี Keypoint ผลการทดลองคือโมเดลมีค่าความแม่นยำมากกว่า 90%

ทวีศักดิ์ เอี่ยมสวัสดิ์ (2559) โดยเป้าหมายของวิทยานิพนธ์นี้คือการประยุกต์ใช้หน่วยความจำระยะสั้นแบบยาว ซึ่งเป็นวิธีไม่แบ่งส่วนในการรู้จำตัวอักษรภาษาไทย นอกจากนี้วิทยานิพนธ์นำเสนอวิธีการเลื่อนองค์ประกอบแนวตั้ง ในการแก้ไขปัญหารูปแบบการรวมกันของอักษรที่เกิดขึ้นแนวตั้ง ในการแก้ไขปัญหารูปแบบการรวมกันของตัวอักษรที่เกิดขึ้นแนวตั้งจำนวนมากบนโครงสร้าวตัวอักษรสี่ระดับภาษาไทย และยากต่อการนำมาใช้กับโครงข่ายหน่วยความจำระยะสั้นแบบยาวมาตรฐาน ผลการทดลองแสดงความแม่นยำเปรียบเทียบวิธีนำเสนอบนโครงข่ายหน่วยความจำระยะสั้นแบบยาวกับซอฟต์แวร์เชิงพาณิชย์ในการรู้จำตัวอักษรภาษาไทย

A. Chaikaew, K Somkuan and T. Yuyen (2564) วัตถุประสงค์ของงานวิจัยนี้คือเพื่อพัฒนาแอปพลิเคชันสำหรับการรู้จำภาษามือที่เป็นภาษาไทยแบบเรียลไทม์โดยการใช้ MidiaPipe Framework มาช่วยในการสกัดแลนด์มาร์กจากวิดีโอท่าทางภาษามือและใช้แลนด์มาร์กเพื่อสร้างโมเดลสำหรับการรู้จำท่าท่างภาษามือด้วย Recurrent Neural Network (RNN) ผลที่ได้จากการวิจัยคือ โมเดลที่สร้างโดย LSTM, BiLSTM และ GRU มีค่าความถูกต้องมากกว่า 90% วิธีนี้สามารถสร้างความแม่นยำได้ใกล้เคียงกับวิธีการแบบดั้งเดิม

กานต์กมล ทวีผล (2562) ได้ศึกษาการทำนายหาปริมาณความหนาแน่นของฝุ่นละออง PM2.5 โดยในการวิจัยนี้ได้ใช้แบบจำลองโครงข่ายประสาทเทียมเชิงลึกแบบ Long Short-Term Memory (LSTM) และแบบจำลองอนุกรมเวลา Seasonal Autoregressive Integrated Moving Averages with Exogenous Regressors (SARIMAX) โดยใช้ข้อมูลฝุ่นละออง ข้อมูลสารก่อมลพิษทางอากาศ งานวิจัยมุ่งหวังในการแสดงสมรรถนะของแบบจำลอง LSTM เปรียบเทียบแบบจำลอง SARIMAX ในการทำนายความหนาแน่นของฝุ่นละออง PM2.5 ในอีก 24 ชั่วโมงข้างหน้า และจากการทดลองพบว่าแบบจำลอง LSTM นั้นให้ค่า RMSE และ MAE แต่ละช่วงเวลาในการทำนายออกมาดีกว่าแบบจำลอง SARIMAX ซึ่งการทำนายในอีก 1 ชั่วโมงข้างหน้านั้นแบบจำลอง LSTM ได้ค่าเฉลี่ยน RMSE = 3.11 ไมโครกรัมต่อลูกบาศก์เมตร และ MAE = 2.36 ไมโครกรัมต่อลูกบาศก์เมตร ในขณะที่ค่าความผิดพลาด (Error) ของแบบจำลอง SARIMAX นั้นมีค่าสูงกว่าเป็นเท่าตัว จากการทดลองจะ

สังเกตได้ว่ายิ่งจำนวนชั่วโมงในการทำนายเพิ่มมากขึ้น ค่าความผิดพลาดที่ได้จากการทำนายของทั้งสองแบบจำลองก็จะยิ่งสูงขึ้น

เอกนรินทร์ ดิษฐ์สันเทียะ (2561) ในงานวิจัยนี้ ผู้วิจัยได้นำเสนอวิธีการในการเรียนรู้เพื่อเพิ่มประสิทธิภาพการตรวจจับพฤติกรรมความรุนแรงในวิดีโอ ซึ่งในวิธีการที่นำเสนอนั้นประกอบด้วยส่วนดังนี้ ในส่วนแรกคือ การสกัดคุณลักษณะของภาพวิดีโอโดยใช้เทคนิคโครงข่ายประสาทเทียมแบบคอนโวลูชัน เพื่ออธิบายข้อมูลเชิงพื้นที่ในแต่ละเฟรมของวิดีโอนอกจากนี้ในงานวิจัยยังได้นำเสนอคุณลักษณะของรูปภาพชนิดใหม่คือ Multiscale Convolution ซึ่งใช้ในการตรวจจับการเปลี่ยนแปลงขนาดเล็กน้อยในวิดีโอ สำหรับในส่วนที่สอง ใช้เทคนิค Long Short-Term Memory (LSTM) ในการจำแนกระดับวิดีโอจากวิดีโอทั้งที่มีเนื้อหาความรุนแรงและไม่มีความรุนแรง จากการทดสอบโดยใช้ข้อมูล 3 ชุด ได้แก่ Hockey Movie และ Real-Violent พบว่าเทคนิคที่นำเสนอให้ค่าความแม่นยำสูงเมื่อเปรียบเทียบกับวิธีอื่น

ผุสดี บุญรอด (2559) โครงการวิจัยนี้มีวัตถุประสงค์การวิจัยเพื่อพัฒนาแบบจำลองการพยากรณ์ราคากองทุนรวมตราสารหนี้และ ตราสารทุนโดยใช้ภาษาไพทอน (Python) และไลบรารีสำหรับการพัฒนาแบบจำลองการพยากรณ์ ได้แก่ Keras และ TensorFlow ซึ่งงานวิจัยประยุกต์ใช้วิธีการแบบผสมผสาน ได้แก่ วิธีการโครงข่าย ประสาทเทียมแบบหน่วยความจ าระยะสั้นแบบยาว (Long Short-Term Memory: LSTM) และ วิธีการล าดับถึงล าดับ (Sequence to Sequence: Seq2Seq) ที่น าไปเปรียบเทียบกับวิธีการ โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network: CNN) จากผลการวิจัย พบว่า วิธีการ LSTM ให้ค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสอง (Mean Square Error: MSE) น้อยที่สุดทั้งการพยากรณ์ราคากองทุนรวมตราสารหนี้และตราสารทุน จากนั้นจึงได้น าแบบจำลอง การพยากรณ์ราคากองทุนรวมที่มีประสิทธิภาพไปใช้ทดลองต่อยอดเพื่อพัฒนาระบบจำลองสำหรับ พยากรณ์ราคากองทุนรวมในรูปแบบเว็บแอปพลิเคชัน (Web Application) ที่ประยุกต์ใช้เครื่องมือ Tableau Desktop ในการพัฒนาระบบสารสนเทศ ดังนั้นผลการวิจัยสามารถสรุปได้ว่า แบบจำลอง การพยากรณ์ราคากองทุนรวมตราสารหนี้และตราสารทุนที่พัฒนาขึ้นมีประสิทธิภาพ ตรงตาม วัตถุประสงค์การวิจัย อีกทั้งสามารถนำไปใช้ต่อยอด หรือเป็นแนวทางสำหรับการพัฒนาระบบ พยากรณ์หรือระบบแนะนำกองทุนรวมตราสารหนี้และตราสารทุนได้ในอนาคต