**บทที่ 2**

# ทฤษฏีการนำโครงข่ายระบบประสาทเชิงลึก มาใช้ในการทำงานของไฟร์วอลล์

## 2.1. เทคโนโลยีของไฟร์วอลล์และโครงข่ายระบบประสาทเชิงลึก

2.1.1. **Firewall**

Firewall เป็นระบบควบคุมและรักษาความปลอดภัยของระบบเครือข่าย คัดกรองข้อมูลเข้าออกในช่องทางอินเตอร์เน็ต เปรียบเสมือนยามเฝ้าประตู คอยป้องกันการโจมตี สแปม ผู้บุกรุกต่างๆที่ไม่หวังดีต่อระบบ และยังสามารถใช้ควบคุมการใช้งานของโปรแกรมที่ต้องการ ในปัจจุบันมีการใช้งานได้ทั้งระบบ Hardware และ Software ขึ้นอยู่กับความเหมาะสม ผลลัพธ์ที่ออกมาจาก Firewall จะพิจารณาการกระทำของ Packet ออกมาเป็น Allow หรือ Deny

**2.1.1.1. Packet Filtering**

ระบบการทำงานของ Firewall ทำงานในระบบ Internet Layer และ Transport Layerตรวจสอบและคั่นกรอง Packet ที่เข้ามาในเครือข่าย โดยพิจารณาจาก Packet Header ตัดสินใจว่าจะทำการ Allow หรือ Deny โดยใช้กฎของ Firewall ในการอ้างอิง ซึ่ง Firewall แบ่งประเภทตามลักษณะการทำงาน ได้แก่

Diagram

Description automatically generated

**รูปที่ 2.1** กระบวนการทำงานของกลไก Packet Filtering Firewall

2.1.1.2. **Stateful Filtering**

Stateful Filtering จะมีเก็บสถานะ Packet ใดที่เคยถูกปล่อยผ่านและเก็บบันทึกไว้ใน State Table ทำให้การทำงานของ Firewall นี้จะถูกตรวจสอบเริ่มจากที่ State Table ก่อน ถ้าหาก Packet ที่กำลังถูกตรวจสอบอยู่ยังไม่เคยถูกปล่อยผ่านยังไม่มีการเก็บสถานะเอาไว้ถึงจะไปพิจารณากฎของไฟรวอลล์เป็นอันดับถัดไป กลไกนี้จะช่วยไฟร์วอลล์ทำงานได้เร็วขึ้น เพราะช่วยลดระยะเวลาในการทำงานไม่ต้องเสียเวลาพิจารณาทุก Packet Header ในกลไก Packet Filtering

2.1.1.3. **Application Firewall**

มีชื่อเรียกได้อีกอย่างหนึ่งว่า “Application-level Firewall” หรือ “Application Gateway” เป็น Firewall ชนิดที่ติดตั้งบนเครื่องคอมพิวเตอร์แยกต่างหาก ทำให้คอมพิวเตอร์เครื่องดังกล่าวทำหน้าที่เป็น Firewall โดยเฉพาะ อย่างไรก็ตามApplication Firewall สามารถกรอง Packet ที่จะผ่านเข้ามาในเครือข่ายอีกทั้งยังตรวจสอบเนี้อหาใน Packet ได้เช่นเดียวกับ Stateful Filtering Firewall นอกจากนี้ Application Firewall ยังทำหน้าที่คล้ายกับ Proxy Server ในการให้บริการคำร้องขอของผู้ใช้ได้อีกด้วย โดยความสามารถของ Application Firewall สามารถแบ่งทำได้ดังนี้

**Security**

การยืนยันตัวตนด้วย AAA คือ Authentication, Authorization และ Audit โดยการสร้าง Token ไปให้ทั้งผู้รับ และผู้ส่ง มีการกำหนด Policy เพื่อการเข้าถึงข้อมูล และยังทำการเก็บข้อมูลการเข้าออกของ Policyนั้นๆ อีกทั้งยังมีการป้องกันด้วยการตรวจสอบข้อมูลที่ได้รับก่อนว่าถูกต้องตามโครงสร้างที่ได้กำหนดไว้หรือไม่

**Integration**

การสร้างการเชื่อมต่อเข้ากับระบบต่างๆให้สามารถทำงานร่วมกันได้ เช่น ถ้าหากระบบที่ใช้มีโปรโตคอลที่แตกต่างกัน จะทำการแปลงโครงสร้างข้อความโดยการจับคู่

**Control and Managing**

การควบคุมปริมาณของข้อความที่จะวิ่งเข้าไปหา Server โดยการกำหนด Policy แยกตามประเภทของ API และประเภทของข้อมูล สำหรับการควบคุมปริมาณข้อความนี้จะเป็นการป้องกันการถูกผู้ไม่หวังดีโจมตีจากช่องโหว่ของระบบได้ เช่น เรามี API ที่เปิดให้ลูกค้าหรือบุคคลอื่นๆเข้ามาใช้งานได้ ถ้าหากไม่มีการกำหนดปริมาณการเรียกใช้ API หรือเส้นทางของข้อมูล ก็จะเกิดช่องโหว่ของระบบที่ผู้ไม่หวังดีสามารถทำการ DOS ได้

**Optimizing**  
   การลดภาระการทำงาน ของ Server โดยการทำ SSL และนำภาระงานจากการถอดรหัสที่ Server ไปให้ไฟร์วอลล์ทำงานแทน จะทำให้ Server มีทรัพยากรเหลือพอที่จะรองรับการทำงานมากขึ้น

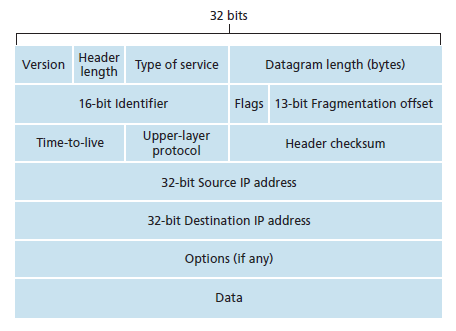
**A picture containing application

Description automatically generated**

**รูปที่ 2.2** กระบวนการทำงานของ Application Firewall

**2.1.2. Packet Header**

Packet Header เป็นโปรโตคอลอินเตอร์เน็ต มาตรฐานที่ทำให้อินเตอร์เน็ตสามารถเชื่อมต่อเข้าหากัน ติดต่อสื่อสารข้อมูลได้ด้วยการกำหนดวิธีการติดต่อสื่อสารร่วมกัน ในส่วนของ Packet Header จะเป็นลำดับชั้นโครงสร้างประกอบไปด้วย Field ข้อมูลที่บ่งบอกถึงวัตถุประสงค์และลักษณะการทำงานของ Packet โดยองค์ประกอบของ Packet Header มีดังนี้



**รูปที่ 2.3** ส่วนประกอบที่สำคัญของ Packet Header Datagram

* Version ส่วนที่ระบุเวอร์ชั่นโปรโตคอลของ Datagram
* Header length ส่วนที่ระบุขนาดของ Datagram Header
* Type of service ส่วนที่ระบุประเภทของ Datagram
* Datagram length ส่วนที่ระบุขนาดของ Datagram ทั้งหมดรวมถึง Datagram Header
* Identifier ส่วนที่มีไว้เพื่อยืนยันตัว หากมีการทำ Fragmention
* Flags ส่วนที่ระบุว่า Datagram นี้จะทำการ Fragmention หรือไม่
* Fragmentation offset ส่วนที่แสดงให้เห็นถึงจำนวนของข้อมูลก่อนทำการ Fragmention
* Time-to-live ส่วนที่กำหนดวงจรชีวิตของ Datagram เพื่อป้องกันไม่เกิด Loop ในเครื่อข่าย
* Protocol ส่วนที่ระบุโปรโตคอลที่ใช้ใน Datagram นี้
* Header checksum ส่วนที่ใช้สำหรับตรวจสอบความถูกต้อง Datagram Header
* Source and destination IP addresses ส่วนที่ระบุที่อยู่ของ IP ต้นทางกับ IP ปลายทาง
* Options ส่วนเพิ่มเติมที่คอยเก็บข้อมูลเช่น เส้นทางที่ใช้โดยเก็บไว้เพื่อตรวจสอบการทำงาน

**2.1.3. Artificial Intelligent**

Artificial Intelligence คือ เครื่องจักรอัจฉริยะที่มีความสามารถในการทำความเข้าใจ เรียนรู้องค์ความรู้ต่างๆ เช่น การรับรู้ การให้เหตุผล ในการแก้ไขปัญหาต่างๆเพื่อปฎิบัติงานตามความต้องการของมนุษย์ เครื่องจักรที่มีความสามารถนี้ถูกเรียกอีกชื่อหนึ่งว่า “ปัญญาประดิษฐ์”

A picture containing diagram

Description automatically generated

**รูปที่ 2.4** ขั้นตอนกระบวนการฝึกฝนปัญญาประดิษฐ์

AI ถูกจำแนกเป็น 3 ระดับตามความสามารถดังนี้

**Narrow Artificial Intelligent** ปัญญาประดิษฐ์เชิงแคบ คือ AI ที่มีความสามารถเฉพาะทางได้ดีกว่ามนุษย์ เช่น เครื่องจักรที่ใช้ในการผ่าตัด

**General Artificial Intelligent** ปัญญาประดิษฐ์ทั่วไป คือ AI ที่มีความสามารถระดับเดียวกับมนุษย์สามารถทำทุกอย่างในประประสิทธิภาพที่ใกล้เคียงกับมนุษย์

**Strong Artificial Intelligent** ปัญญาประดิษฐ์แบบเข้ม คือ AI ที่มีความสามารถมากกว่ามนุษย์ในหลายๆด้าน

และจากการนำปัญญาประดิษฐ์มาประยุกต์ใช้ในการแก้ไขปัญหา มุมมองต่อ AI ที่แต่ละคนมีอาจไม่เหมือนกัน ขึ้นอยู่กับว่า เราต้องการความฉลาดโดย คำนึงถึงพฤติกรรมที่มีต่อสิ่งแวดล้อมหรือคำนึงการคิดได้ของผลผลิต AI ดังนั้นจึงมีคำนิยาม AI ตามความสามารถที่มนุษย์ต้องการ ให้มันแบ่งได้ 4 กลุ่ม ดังนี้

**Thinking humanly** (การคิดคล้ายมนุษย์)

natural language processing สื่อสารกับ มนุษย์ได้ด้วยภาษาที่มนุษย์ใช้ เช่น ภาษาอังกฤษ เป็นการประมวลผลภาษาธรรมชาติ

computer vision มีประสาทรับสัมผัสคล้ายมนุษย์ เช่นคอมพิวเตอร์วิทัศน์ รับภาพได้โดยใช้อุปกรณ์รับสัญญาณภาพ

machine learning เพื่อปรับให้เข้ากับสถานการณ์ใหม่และ ตรวจจับและคาดการณ์รูปแบบ

**Thinking rationally (คิดอย่างมีเหตุผล)**

คิดอย่างมี เหตุผล หรือคิดถูกต้อง โดยใช้หลักตรรกศาสตร์ในการคิดหาคำตอบอย่างมีเหตุผล เช่น ระบบผู้เชี่ยวชาญ

**Acting humanly (การกระทำคล้ายมนุษย์)**

การคิดคล้าย มนุษย์ ก่อนที่จะทำให้เครื่องคิดอย่างมนุษย์ได้ ต้องรู้ก่อนว่ามนุษย์มีกระบวนการคิดอย่างไร ซึ่งการวิเคราะห์ลักษณะการคิดของมนุษย์เป็นศาสตร์ด้าน cognitive science เช่น ศึกษาโครงสร้างสามมิติของเซลล์สมอง การแลกเปลี่ยนประจุไฟฟ้าระหว่างเซลล์สมอง วิเคราะห์การเปลี่ยนแปลงทางเคมีไฟฟ้าในร่างกายระหว่างการคิด ซึ่งจนถึงปัจจุบันเราก็ยังไม่รู้แน่ชัดว่า มนุษย์เรา คิดได้อย่างไร

**Acting rationally (การกระทำอย่างมีเหตุผล)**

กระทำอย่างมีเหตุผล เช่น agent (agent เป็นโปรแกรมที่มีความสามารถในการกระทำ หรือเป็นตัวแทนในระบบอัตโนมัติต่าง ๆ ) สามารถกระทำอย่างมีเหตุผลคือ agent ที่กระทำการเพื่อบรรลุเป้าหมายที่ได้ตั้งไว้ เช่น agent ใน ระบบขับรถอัตโนมัติที่มีเป้าหมายว่าต้องไปถึงเป้าหมายในระยะทางที่สั้นที่ สุด ต้องเลือกเส้นทางที่ไปยังเป้าหมายที่สั้นที่สุดที่เป็นไปได้จึงจะเรียกได้ ว่า agent กระทำอย่างมีเหตุผล อีกตัวอย่างเช่น agent ใน เกมหมากรุกมีเป้าหมายว่าต้องเอาชนะคู่ต่อสู้ ต้องเลือกเดินหมากที่จะทำให้คู่ต่อสู้แพ้ให้ได้ เป็นต้น

2.1.4. **Machine Learning**

Machine Learning คือ ส่วนการเรียนรู้ของเครื่อง ถูกใช้งานเสมือนเป็นสมองของปัญญาประดิษฐ์ในการสร้างความฉลาด มักจะใช้เรียกโมเดลที่เกิดจากการเรียนรู้ของปัญญาประดิษฐ์ โดยมนุษย์มีหน้าที่เขียนโปรแกรมให้เรียนรู้จากชุดข้อมูลฝึกสอนหรือ Training set และอาศัยกลไกที่เป็นโปรแกรม หรือเรียกว่า Algorithm ที่มีหลากหลายแบบ โดยมี Data Scientist เป็นผู้ออกแบบ หนึ่งใน Algorithm ที่ได้รับความนิยมสูง คือ Deep Learning ซึ่งถูกออกแบบมาให้ใช้งานได้ง่าย และประยุกต์ใช้ได้หลายลักษณะงาน อย่างไรก็ตาม ในการทำงานจริง Data Scientist จำเป็นต้องออกแบบตัวแปรต่างๆ ทั้งในตัวของ Deep Learning เอง และต้องหา Algorithm อื่นๆ มาเป็นคู่เปรียบเทียบ เพื่อมองหา Algorithm ที่เหมาะสมที่สุดในการใช้งานจริง โดยตามหลักแล้วจะแบ่งประเภทของ Machine Learning ได้ดังนี้

2.1.4.1. **Supervised**

การทำให้เครื่องคอมพิวเตอร์สามารถเรียนรู้ได้จากชุดข้อมูลฝึกสอนหรือ Training set ก่อนที่จะประมวลผล โดยมนุษย์จะเป็นผู้กำหนดคุณลักษณะความสัมพันธ์เฉพาะของข้อมูลที่ต้องการให้เครื่องคอมพิวเตอร์เรียนรู้ หรือที่เรียกว่า Label และเมื่อโมเดลผ่านการเรียนรู้แล้ว จะสามารถแยกแยะประเภทมีวิธีการคิดที่เริ่มมีเหตุผล เมื่อข้อมูลที่ต้องการวิเคราะห์มีจำนวนที่มากขึ้นจำเป็นต้องมีข้อมูลที่เป็น Training set มากขึ้นเช่นเดียวกัน โดยการเรียนรู้แบบ Supervised Learning นี้จะประกอบไปด้วยดังนี้

**Classification**

คือการสอนโมเดลให้สามารถแบ่งหรือแยกประเภทกลุ่มข้อมูลได้ โดยอ้างอิงจากความสัมพันธ์และน้ำหนักของข้อมูลแต่ละ Label ตัวอย่างเช่น การแยกกลุ่มผู้ป่วยว่าเป็นเนื้องอกในสมอง ซึ่งจะมีปัจจัยต่างๆมากมายไม่ว่าจะเป็น ขนาด, รูปร่าง, ตำแหน่ง หรือ สีผิว ซึ่งถ้าหากมีข้อมูลเพียงแค่ Label เดียว ไม่สามารถพิสูจน์หรือแบ่งกลุ่มได้

**Regression**

การสอนโมเดลโดยอิงจากผลลัพธ์ที่ผ่านมา โดยผลลัพธ์จะเป็นการประมาณค่าความเป็นไปได้ที่จะเกิดขึ้นต่อ ทำให้เหมาะแก่การวิเคราะห์ความสัมพันธ์ของตัวแปรที่อยู่ในรูปกราฟ เช่น การหาความสัมพันธ์ระหว่างขนาดบ้านและราคา การประเมินราคาหุ้น

**2.1.4.2.** **Unsupervised**

รูปแบบการเรียนรู้ที่ไม่จำเป็นต้องใช้ชุดข้อมูลฝึกสอน แต่เป็นการป้อนข้อมูล Test set ไปประมวลผลเพียงอย่างเดียว ทำให้ผลลัพธ์ที่ออกมาไม่รู้ผลลัพธ์แน่ชัด ซึ่งอัลกอรึทึ่มจะวิเคราะห์และหาโครงสร้างของข้อมูลเอง

**Clustering**

เป็นการกำหนดให้เครื่องคอมพิวเคอร์หาวิธีแบ่งกลุ่มหรือจัดกลุ่มข้อมูลเอง เปรียบเสมือนการลด Label ของข้อมูลที่มีปริมาณมาก จัดกลุ่มหาข้อมูลที่มีความสัมพันธ์ใกล้เคียงกัน ผลลัพธ์ที่ได้ออกมาจะมีปริมาณ Label ที่น้อยลงเป็นอย่างมาก

**Dimensionality Reduction**

เป็นการกลไกการบีบอัดและลดมิติข้อมูลจำนวนมากให้มีจำนวนลดลงโดยที่ข้อมูลยังครบถ้วน และยังสามารถนำไปจำแนกข้อมูลได้เหมือนเดิม

Diagram

Description automatically generated

**รูปที่ 2.5** ขั้นตอนการแยกหมวดหมู่และรูปแบบโมเดลที่จะศึกษา

2.1.4.3. **Reinforcement Learning**

เป็นการเรียนรู้ด้านหนึ่งของ Machine Learning มักใช้พัฒนาหุ่นยนต์หรือการเรียนรู้ที่อยู่ภายในเกมคอมพิวเตอร์ เช่น การลองผิดลองถูกไปเรื่อยเพื่อหาผลลัพธ์ที่ดีที่สุดประเมินออกมาเป็นคะแนน โดยชุดข้อมูลทดสอบจะเป็นสภาพแวดล้อมโดยรอบขึ้นอยู่กับความต้องการของผู้พัฒนา

2.1.5. **Deep Learning**

Deep learning คือ อัลกอริทึมการเรียนรู้เชิงลึกโดยใช้หลักการ Artificial Neural Networks ที่มีรูปแบบการทำงานคล้ายคลึงกับเซลล์ประสาทที่เชื่อมต่อกันเป็นโครงข่ายประสาทในสมองมนุษย์ เหมาะกับการวิเคราะห์ข้อมูลขนาดใหญ่ที่มีความซับซ้อน เช่น การจำแนกรูปภาพ การจำแนกใบหน้า ประกอบไปด้วย โครงสร้างของหน่วยประมวลผลจำนวนมากคือเซลล์ประสาท หรือ Neuron โดยอัลกอริทึ่มนี้จะประกอบไปด้วยชั้นต่างๆ ดังนี้

* Input Layer มีหน้าที่รับข้อมูลเข้ามาประมวลผลและส่งต่อให้ Hidden Layer
* Hidden Layer มีหน้าที่คำนวณและประมวลผลข้อมูลโดยสามารถมีได้หลายชั้น หลายขนาดขึ้นอยู่กับความซับซ้อนของข้อมูล
* Output Layer มีหน้าที่ส่งผลลัพธ์ข้อมูลที่ผ่านการประมวลผลแล้วออกมา

เมื่อเริ่มการฝึกฝนจะเริ่มจากการสุ่มค่าถ่วงน้ำหนัก (Weight) และจะเริ่มปรับผลลัพธ์เอามาคูณกับค่าถ่วงน้ำหนักแล้วบวกด้วยค่าความเอนเอียงของข้อมูล (Bias) หลังจากนั้นจะนำผลลัพธ์ที่ได้มาในแต่ละขาของ Neural Network มารวมกันแล้วมาผ่านฟังก์ชั่นส่งต่อให้ลำดับชั้นถัดไปประมวลผลมีการใช้วิธีการประมวลผลทางคณิตศาสตร์ (Activation Function) โดยทุกวันนี้มีการะประยุกต์ใช้อย่างแพร่หลาย แบ่งชนิดโครงข่ายประสาทออกเป็นดังนี้

Chart, scatter chart

Description automatically generated

**รูปที่ 2.6** ความแตกต่างระหว่าง Machine Learning และ Deep Learning

2.1.5.1. โครงข่ายประสาทแบบป้อนไปข้างหน้า **(Forward Propagation)**

Feed-forward neural networks ถือเป็นโมเดลที่มีโครงสร้างที่เรียบง่ายที่สุด เพราะว่า การดำเนินการของข้อมูลจะเป็นไปในทิศทางเดียว ก็คือ รับข้อมูลจาก input layer แล้วส่งไปต่อไปยัง hidden layer เลื่อยๆ จนกระทั้งถึง output layer ก็จะหยุด สังเกตุได้ว่าจะไม่มีวงวน หรือ loop เกิดขึ้นเลย

2.1.5.2. โครงข่ายแบบวนซ้ำ **(Recurrent neural networks : RNN)**

Recurrent neural networks คือ neural networks หลายเลเยอร์ที่สามารถเก็บข้อมูล informationไว้ที่ node จึงทำให้มันสามารถรับข้อมูลเป็นเเบบลำดับ (data sequences) และให้ผลลัพธ์ออกเป็นลำดับของข้อมูลได้ อธิบายอย่างง่ายๆ RNN ก็คือ neural network เชื่อมต่อกันหลายๆอันเเละยังสามารถต่อกันเป็นวงวนหรือ loop ได้นั่นเอง เพราะฉะนั้น RNN จึงเหมาะสมในการประมวลผลข้อมูลที่เป็นลำดับอย่างมาก

## 2.2. ทบทวนวรรณกรรม

2.2.1. การนำเอาความสามารถของ **GPU** มาใช้ในการคำนวณ

การที่เราเลือกใช้ GPU ในการทำ Machine Learning เนื่องจากตัว GPU นั้นมีหน่วยความจำที่ให้ค่าแบนด์วิดธ์ที่สูง และตัว GPU เองยังออกแบบให้สามารถแก้สมการทางคณิตศาสตร์ได้อย่างรวดเร็ว นอกจากนี้ยังมีจำนวนหน่วยประมวลผลที่มีมากกว่า CPU หลายเท่าตัว จึงทำให้มีอัตราการประมวลผลที่สูงกว่า CPU และยังมีแพลตฟอร์มของ Nvidia ที่รองรับอย่าง CUDA ซึ่งเป็น Parallel Computing แพลตฟอร์มเพื่อช่วยให้นักพัฒนาสร้าง Tools ในการเรียกใช้การประมวลผลของ GPU และยังมี library อย่าง NVDIA cuDNN ซึ่งรองรับการทำ Deep Neural Network โดยตัว cuDNN ได้อำนวยการปรับแต่งขั้นสูงสำหรับการทำงานของ DNN เช่น forward และ backward convolution pooling normalization activation layers เป็นต้น

2.2.2. การกำหนดรูปแบบของโมเดลที่จะนำมาใช้ทำ **Packet Filtering AI Firewall**

โครงสร้างของโมเดลจะถูกกำหนดขึ้นอยู่กับรูปแบบความสัมพันธ์ของข้อมูล ชนิดและรูปแบบของคำตอบที่ต้องการ ซึ่งค่าเหล่านี้จะเป็นการกำหนดวิธีการและรูปแบบอัลกอริทึ่มเชิงลึกคำนวณค่าน้ำหนักด้วยไลบราลีของเครื่องมือที่เลือกใช้ โดยการพัฒนา Packet Filtering AI Firewall มีหลักการที่ต้องเลือกใช้และมีหัวข้อที่พิจารณา ดังต่อไปนี้

**Binary Logistic Regression**

การวิเคราะห์สมการถดถอยโลจิสติกเป็นเทคนิคการวิเคราะห์สถิติเชิงคุณภาพ แตกต่างไปจากการวิเคราะห์เชิงปริมาณ ซึ่งตัวแปรเชิงคุณภาพเรียกอีกอย่างหนึ่งว่าเชิงกลุ่ม การทำงานของ Logistic Regression จะมีความคล้ายคลึงกับ Linear Regression ต่างกันที่ฟังก์ชั่น Logistic จะตัดสินเพียงแค่ใช่หรือไม่ใช้เท่านั้นต่างกับฟังก์ชั่น Linear ที่เป็นการทำนายแบบต่อเนื่อง เช่น การทำนายขนาด เป็นต้น โดยเทคนิคการวิเคราะห์ Logistic Regression นี้ใช้กับตัวแปรที่ถูกแบ่งออกเป็น 2 กลุ่มย่อย (dichotomous variable) คือ 0 และ 1 ซึ่งในงานวิจัยของเราถูกแบ่ง Output ที่วิเคราะห์ออกมาเป็น 2 กลุ่มเช่นเดียวกันคือ Allow และ Deny

Chart, line chart, scatter chart

Description automatically generated  
การทำนายข้อมูลของฟังก์ชั่น Binary Logistic Regression

**ฟังก์ชั่นประมวลผล Sigmoid**

ฟังก์ชั่นประมวลผล คือ ฟังก์ชั่นที่รับผลการประมวลผลทั้งหมดจากทุก Input ภายใน 1 นิวรอนแล้วพิจารณาว่าจะส่งต่อ Output ออกมาเป็นอย่างไร โดยฟังก์ชั่นประมวลผล Sigmoid จะถูกวางไว้ใน Layer สุดท้าย มักเป็นที่นิยมใช้ใน Deep Learning เพราะมีการถูกใช้ใน Artificial Neural Network ที่มีการต้องหาค่าความน่าจะเป็น ซึ่งจะเป็นค่า Output ได้แค่ 0 ถึง 1 เท่านั้น หนึ่งในฟังก์ชั่น Sigmoid ที่พบบ่อยที่สุดคือฟังก์ชั่น Sigmoid Logistic สิ่งนี้มักถูกใช้กับการเรียนรู้โครงข่ายประสาทเทียม โดยฟังก์ชั่นจะถูกกำหนดไว้ ดังนี้

Chart

Description automatically generated  
กราฟของฟังก์ชั่นประมวลผล Sigmoid

**จำนวน Hidden Layer**

จำนวน Hidden Layer ขึ้นอยู่กับความซับซ้อนของโมเดล ซึ่งในงานวิจัยของเราคือการสร้างโมเดลโครงข่ายประสาทเทียมเชิงลึกหรือ Deep Neural Network ซึ่งต้องการมากกว่า 2 Hidden Layer ขึ้นไป

Table, timeline

Description automatically generated  
ตารางอ้างอิงในการเลือกใช้จำนวน Hidden Layer ของโมเดล

**จำนวน Node หรือนิวรอนในแต่ละ Layer**

การกำหนดจำนวนนิวรอนในแต่ละชั้น Layer เป็นส่วนสำคัญในสถาปัตยกรรมโครงสร้างโครงข่ายประสาทเทียม แม้ว่าจำนวนชั้น Layer จะไม่ค่อยส่งผลกระทบมาก แต่จำนวนนิวรอนมีอิทธิผลอย่างมากต่อผลลัพธ์สุดท้าย ดังนั้นการเลือกใช้จำนวนนิวรอนจะต้องมีการพิจารณาอย่างรอบคอบ

การใช้เซลล์ประสาทน้อยเกินไปใน Hidden Layer จะทำให้เกิดสิ่งที่เรียกว่า Underfitting ปํญหาที่จำนวนนิวรอนมีจำนวนน้อยเกินไปจนทำให้ระบบไม่สามารถหาความสัมพันธ์ได้อย่างเพียงพอต่อข้อมูลที่มีมากเกิน หรือถ้าหากตั้งจำนวนนิวรอนมีมากเกินไปก็ส่งผลไปหลายประการเช่นกัน ประการแรกคือระบบมีความสามารถในการประมวลผลมากเกินไปทำให้ไม่เพียงพอต่อข้อมูลที่มีอยู่เพียงจำกัด ประการที่สองคือ ระยะเวลาในการฝึกสอนโมเดลสามารถเพิ่มขึ้นจนถึงจุดที่โมเดลไม่สามารถฝึกได้อย่างเพียงพอ

จำนวน Node ในแต่ละ Hidden Layer มีสูตรคำนวณ ดังนี้

A picture containing diagram

Description automatically generated

โดยความหมายของตัวแปรแต่ละตัว ให้เป็น

Ni = จำนวนนิวรอนของ Input Layer  
No = จำนวนนิวรอนของ Output Layer.  
Ns = จำนวนของชุดฝึกสอนโมเดลที่ใช้ในโมเดล  
α = ค่าตัวแปรสำหรับปรับขนาด มักใช้เลขระหว่าง 2-10

**ขนาดของ Batch Size**

Batch คือขนาดของรายการที่จะให้ประมวลผลและคำนวณใน 1 ครั้ง การกำหนดขนาดของ Batch จะมีผลกับความเร็วในการคำนวณ ยิ่ง Batch ใหญ่จะยิ่งคำนวณเร็ว แต่ Batch ขนาดที่ใหญ่เกินไปอาจจะส่งผลให้มีข้อมูลขนาดใหญ่เกินกว่า Memory เราจะรองรับได้

**Underfitting** คือ การที่โมเดลของเราไม่สามารถทำงานได้ จากการที่ไม่สามารถจัดแนวโน้มของข้อมูลได้ อันเนื่องมากจากโมเดลเราไม่เหมาะสมหรือข้อมูลมีจำนวนน้อยไป กรณีนี้โมเดลมีค่าความเอนเอียงสูง (high bias) ยกตัวอย่างเช่น หากเรานำช้อมูลที่ Train มาลองแล้วได้ความแม่นยำต่ำ เมื่อนำชุดข้อมูลทดสอบมาลองก็จะได้ความแม่นยำต่ำเช่นกัน

**Overfitting** คือ การที่โมเดลตอบสนองต่อการรบกวน (noise) จำนวนมาก จนเริ่มเรียนจากการรบกวนและรายละเอียดของข้อมูลที่ไม่ถูกต้อง แล้วโมเดลของเราจะไม่เหมาะสมสำหรับการสามารถทำนายข้อมูล เช่น ทำนายข้อมูลที่ไม่เคยมีอย่างผิดพลาดกว่าที่คาดจะเป็นมาก (ล้มเหลวที่จะทำนายข้อมูลได้ถูกต้อง) เพราะมีรายละเอียดและการรบกวนมากเกินไป กรณีนี้โมเดลมีค่าค่าความแปรปรวนของข้อมูลสูง (high variance) ยกตัวอย่างเช่น โมเดลที่พัฒนาขึ้นมีความแม่นยำจากชุดข้อมูลทดสอบมากถึง 99% แต่เมื่อนำชุดข้อมูลทดสอบซึ่งไม่เคยปรากฏเลยในชุดข้อมูลฝึกสอนมาทดสอบ ทำให้ความแม่นยำเหลืออยู่เพียง 40% ปัญหานี้คือ Overfitting

Chart, scatter chart

Description automatically generated  
ปัญหา error จาก overfitting และ underfitting

**จำนวนรอบการเรียนรู้ซ้ำ (Epoch)**

Epoch คือจำนวนรอบของการเทรนหรือเรียนรู้ของโมเดลในข้อมูลฝึกสอนชุดเดิม ซึ่งการเรียนรู้ซ้ำแต่ละครั้งจะทำให้ค่า Loss หรืออัตราการสูญเสียของข้อมูลลดลง ในขณะเดียวกันค่าแม่นยำ (Accuracy) จะเพิ่มขึ้น ซึ่งเป็นผลดีกับโมเดล การเพิ่มจำนวนรอบการเรียนรู้แม้ความแม่นยำจะเพิ่มขึ้น แต่ทุกๆรอบการวนใหม่นั้นจะทำให้ learning rate ลดลง เนื่องจากใช้ข้อมูลเดิม อัตราการเรียนรู้จะสูงขึ้นในช่วงแรกและลดลงจนกระทั่งไม่เห็นผลความแตกต่าง ความแม่นยำจะขยับเข้าค่าคงที่ถึงจุดๆหนึ่ง ดังนั้นการควบคุมตัวแปรจำนวน epoch สามารถช่วยย่นระยะเวลาในการทดลองได้ เพราะแม้เพิ่ม epoch ไปก็จะไม่ส่งผลดีกับโมเดลมากนัก

Chart, line chart

Description automatically generated  
 ความสัมพันธ์อัตราการเรียนรู้กับจำนวนรอบการเรียนรู้