โปรแกรมทำนายผลการเรียนล่วงหน้าด้วยโครงข่ายประสาทเทียม

ACADEMIC RECORD PREDICTION PROGRAM BY USING ARTIFICIAL NEURAL NETWORK

การุณ ฉัตรมณีฤกษ์
นิธิ สกุลพานิช

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

โปรแกรมทำนายผลการเรียนล่วงหน้าด้วยโครงข่ายประสาทเทียม

ACADEMIC RECORD PREDICTION PROGRAM BY USING ARTIFICIAL NEURAL NETWORK

การุณ ฉัตรมณีฤกษ์
นิธิ สกุลพานิช

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

ปริญญานิพนธ์ปีการศึกษา 2557

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมทำนายผลการเรียนล่วงหน้าด้วยโครงข่ายประสาทเทียม
ACADEMIC RECORD PREDICTION PROGRAM BY USING ARTIFICIAL NEURAL NETWORK

ผู้จัดทำ

1. นายการุณ ฉัตรมณีฤกษ์ รหัสนักศึกษา 54010098
2. นายนิธิ สกุลพานิช รหัสนักศึกษา 54010710

.....อาจารย์ที่ปรึกษา

(รองศาสตราจารย์ ดร. เกียรติกูล เจียรนัยธนะกิจ)

โปรแกรมทำนายผลการเรียนล่วงหน้าด้วย โครงข่ายประสาทเทียม

นาย การุณ	ฉัตรมณีฤกษ์	54010098
นาย นิธิ	สกุลพานิช	54010710
รศ.ดร. เกียรติกูล	เจียรนัยธนะกิจ	อาจารย์ที่ปรึกษา

ปีการศึกษา 2557

บทคัดย่อ

ในเทอมแรกของการศึกษาระดับชั้นปริญญาตรีของภาควิชาวิศวกรรมคอมพิวเตอร์ จะมีนักศึกษาบางส่วน ที่ผลการเรียนอยู่ในระดับที่ไม่ดี และในบางครั้งส่งผลให้นักศึกษาบางคนไม่สามารถเรียนอยู่ในภาควิชาวิศวกรรมคอมพิวเตอร์ต่อไปได้ โปรแกรมนี้จะช่วยไม่ให้นักศึกษาเลือกภาควิชาที่ต้องการศึกษาผิด โครงงานนี้จึงได้พัฒนาโปรแกรมทำนายผลการเรียนล่วงหน้าด้วยโครงข่ายประสาทเทียม ที่จะช่วยบอกคะแนน GPA ในเทอมแรกของนักศึกษาใหม่ เพื่อช่วยในการตัดสินใจในการเข้าศึกษาต่อในภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ในส่วนของการประมวลผลนั้น โครงข่ายประสาทเทียมจะทำการประมวลผล จากตัวแปรซึ่งประกอบด้วยคะแนน GAT PAT1 PAT2 PAT3 และ วิธีการเข้าศึกษา แล้วทำนายช่วงของ GPA ออกมาเป็นผลลัพธ์ ผู้ใช้งานสามารถใช้งานผ่านเว็บแอพพลิเคชั่นโดยใช้คะแนนและวิธีการเข้าศึกษาของตนเอง ผู้พัฒนาทำการฝึกฝนโครงข่ายประสาทเทียม โดยใช้ข้อมูลจากนักศึกษาชั้นปีที่ 2 3 และ 4 ของภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง แล้วทำการแปลงข้อมูลที่ได้จากแบบสอบถามให้เป็นชุดข้อมูลที่มีฟอร์แมตสำหรับใช้งานโครงข่ายประสาทเทียม การฝึกฝนนี้เมื่อสำเร็จแล้ว ผู้พัฒนาจะได้ค่าน้ำหนักสำหรับนำไปอัพโหลดเพื่อนำไปใช้งานต่อในการทำนายผลลัพธ์ในลักษณะช่วงของ GPA ในเทอมแรกของการศึกษา

ACADEMIC RECORD PREDICTION PROGRAM BY USING ARTIFICIAL NEURAL NETWORK

Mr. Karun	Chatmaneelerk	54010098
Mr. Nithi	Sakunphanish	54010710
Assoc. Prof. Dr. Kietikul	Jearanaitanakij	Advisor

Academic Year 2014

ABSTRACT

In the first semester of students in Bachelor of Engineering in Computer Engineering program, some students usually get low GPA or retired. This may discourage those students and reflect the mistake of choosing the wrong major. This project develops an academic record prediction program for predicting this problem. This program will predict the first semester GPA to help students deciding whether they are suitable in enrolling the Computer Engineering program at King Mongkut's Institute of Technology Ladkrabang.

In the calculation part, Artificial Neural Network calculates the input attributes composed of GAT, PAT1, PAT2, PAT3 and admission method then predicts the grade range as the output. User can use academic record prediction program in web application by input their GAT, PAT1, PAT2, PAT3 scores and admission method in the input form. Developer trains Artificial Neural Network by using data from the second, third and fourth year students in the Computer Engineering department at King Mongkut's Institute of Technology Ladkrabang. Then transform the data from a survey to the format that is used in Artificial Neural Network. After finishing training part, developer will obtain the set of weights that can upload to web application for predicting the result as the grade range of first semester in the first year's GPA.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ไม่อาจสำเร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และความร่วมมือจากผู้ที่ให้ความรู้ ให้คำปรึกษา ตลอดจนให้ความช่วยเหลือในด้านต่าง ๆ บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้ปริญญานิพนธ์นี้เกิดขึ้นได้ คือ อาจารย์ เกียรติกูล เจียรนัยธนะกิจ อาจารย์ที่ปรึกษาโครงงาน ที่ให้ความช่วยเหลือจนสำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณบิดา มารดา และ ญาติพี่น้องที่ให้การอบรม และสนับสนุน ทั้งในด้านการเรียน ด้านกิจกรรม และในด้านต่าง ๆ

ขอขอบคุณเพื่อนๆ และน้องๆ ที่ให้คำแนะนำ คำปรึกษา แสดงความคิดเห็น รวมไปถึงการให้ความร่วมมือในขั้นตอนการดำเนินงานของโครงงานในส่วนของการเก็บข้อมูลเป็นอย่างดี ถ้าขาดส่วนนี้ไปโครงงานนี้จะไม่อาจสำเร็จลงได้

ขอขอบคุณสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้การฝึกฝนและให้ความรู้ รวมไปถึงสถานที่ ที่เหมาะสมสำหรับการทำโครงงาน ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และ ขอกราบขอบพระคุณมา ณ ที่นี้

นาย การุณ ฉัตรมณีฤกษ์

นาย นิธิ สกุลพานิช

สารบัญ

หน้า

บทคัดย่อ I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VIII

สารบัญรูป IX

บทที่ 1 บทนำ 1

1.1 ความสำคัญและที่มาของโครงงาน 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ขอบเขตของโครงงาน 2

1.4 วิธีการดำเนินการของโครงงาน 2

1.5 ประโยชน์ที่คาดว่าจะได้รับ 4

1.6 ส่วนประกอบของปริญญานิพนธ์ 4

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 5

2.1 ระบบ GAT / PAT 5

2.1.1 ประวัติความเป็นมา 5

2.1.2 GAT/PAT คืออะไร 7

2.1.3 แต่ละคณะใช้คะแนนอะไรบ้าง 8

2.1.4 ใครสอบ GAT/PAT ได้บ้าง 9

2.2 วิธีการเข้าศึกษาในสถาบันฯ 9

2.2.1 แอดมิชชั่น (Admission) 9

2.2.2 สอบตรง 10

2.2.3 โควต้า 11

2.3 โครงข่ายประสาทเทียม (Artificial Neural Network) 11

สารบัญ (ต่อ)

หน้า

2.3.1 โครงข่ายประสาทเทียมคืออะไร 11
2.3.2 นิวรอนเปรียบเสมือนองค์ประกอบในการคำนวณ 14
2.3.3 เพอร์เซปตรอน (Perceptron) 17
2.3.4 Multilayer neural networks 23
2.4 สหสัมพันธ์ (Correlation) 35
2.4.1 การวิเคราะห์สหสัมพันธ์ (Correlation Analysis) 35
2.4.2 สัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน (Pearson product-moment correlation coefficient) 37
2.5 ภาษาจาวา (Java programming language) 39
2.5.1 ประวัติของจาวา 39
2.5.2 คุณลักษณะเด่นของภาษาจาวา 39
2.5.3 จุดมุ่งหมายของการพัฒนาภาษาจาวา 40
2.5.4 องค์ประกอบของภาษาจาวา 40
2.5.5 ขั้นตอนการทำงานของโปรแกรมภาษาจาวา 41
2.5.6 Java Platform 42
2.5.7 เครื่องมือที่ใช้พัฒนาโปรแกรมภาษาจาวาสำหรับซอฟแวร์โครงงานนี้ 43
2.6 เจเอสพีและเซิร์ฟเลต (JSP & Servlet) 44
2.6.1 เจเอสพีและเซิร์ฟเลตคืออะไร 44
2.6.2 ข้อดีของภาษาเจเอสพี 46
2.6.3 เปรียบเทียบเจเอสพีและเอเอสพี 46
2.6.4 เปรียบเทียบเจเอสพีและพีเอชพี 47

บทที่ 3 การออกแบบและการพัฒนาซอฟต์แวร์ 48
3.1 แนวคิดในการพัฒนา 48
3.2 การออกแบบแบบสอบถาม 49
3.3 เครื่องมือที่ใช้ในการพัฒนา 52

สารบัญ (ต่อ)

หน้า

3.3.1 ระบบปฏิบัติการ 52
3.3.2 ภาษาโปรแกรมที่ใช้ 52
3.3.3 เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์ 52
3.4 รายละเอียดส่วนติดต่อผู้ใช้งาน (User Interface) 53
3.4.1 ผังการเชื่อมต่อของส่วนติดต่อผู้ใช้งาน 53
3.4.2 หน้าต่างหลักของเว็บส่วนของผู้ใช้งาน 54
3.4.3 หน้าต่างส่วนเนื้อหาของผู้ใช้งาน 55
3.4.4 หน้าต่างหลักของโปรแกรมส่วนผู้พัฒนา 59
3.5 โครงสร้างของระบบ 63
3.5.1 ส่วนติดต่อผู้ใช้งาน (Graphic user interface) 63
3.5.2 ส่วนของการประมวลผล 63
3.5.3 ส่วนของการตรวจสอบผลลัพธ์ 66
3.6 การออกแบบระบบ 67
3.6.1 System Architecture Diagram 67
3.6.2 แผนภาพแสดงการไหลของข้อมูล (Data Flow Diagram) 67
3.6.3 แผนภาพ UML (Unified Modeling Language Diagram) 73
3.6.4 Artificial Neural Network Diagram 75
3.7 ขอบเขตและข้อจำกัดของซอฟต์แวร์ที่พัฒนา 76

บทที่ 4 การทดลองและผลการทดลอง 77
4.1 การทดลองการทำงานของโครงข่ายประสาทเทียม 77
4.1.1 การทดลองโครงข่ายประสาทเทียมด้วยชุดข้อมูล Iris 77
4.1.2 การทดลองชุดข้อมูลที่ใช้ในโครงงาน 78
4.1.3 การทดลองการตั้งค่าในโครงข่ายประสาทเทียม 81
4.2 การทดลองซอฟต์แวร์เบื้องหลังส่วนของผู้พัฒนา 86
4.2.1 การทดลองใช้ซอฟต์แวร์ส่วนชุดข้อมูล 87

สารบัญ (ต่อ)

หน้า

4.2.2 การทดลองใช้ซอฟต์แวร์ฝึกฝนโครงข่ายประสาทเทียม 88
4.2.3 การทดลองใช้ซอฟต์แวร์ในส่วนของหน้าต่างค่าน้ำหนัก (WEIGHT) 90
4.2.4 การทดลองใช้เว็บแอพพลิเคชั่น 92

บทที่ 5 บทสรุปและข้อเสนอแนะ 97
5.1 บทสรุป 97
5.2 ปัญหา อุปสรรค และ แนวทางแก้ไข 98
5.2.1 ปัญหาและอุปสรรค 98
5.2.2 แนวทางแก้ไขปัญหา 98
5.3 แนวทางในการพัฒนาต่อ 98

บรรณานุกรม 100

สารบัญตาราง

ตารางที่ **หน้า**

1.1 การดำเนินการในภาคเรียนที่ 1 3
1.2 การดำเนินการในภาคเรียนที่ 2 3
2.1 เปรียบเทียบระหว่างโครงข่ายประสาทเทียมกับโครงสร้างทางชีวภาพ 14
2.2 Truth tables for the basic logical operations 20
2.3 Example of perceptron learning: the logical operation AND 21
2.4 ผลลัพธ์สุดท้ายของการฝึกโครงข่ายสามเลเยอร์ ซองตรรกะโอเปอเรชั่น Exclusive-OR 33

สารบัญรูป

รูปที่ **หน้า**

2.1 สถาบันทดสอบทางการศึกษาแห่งชาติ 6
2.2 โครงข่ายประสาททางชีวภาพ 12
2.3 โครงสร้างตัวอย่างของโครงข่ายประสาทเทียม 13
2.4 ไดอะแกรมของนิวรอน 15
2.5 Activation function ประเภทต่างๆ 16
2.6 Single-layer two-input perceptron 17
2.7 เส้นแบ่งของเพอร์เซปตรอน: 18
2.8 Two-dimensional plots of basic logical operations 22
2.9 Multilayer perceptron with two hidden layers 24
2.10 Three-layer back-propagation neural network 26
2.11 โครงข่ายสามเลเยอร์สำหรับแก้ปัญหาโอเปอเรชั่น Exclusive-OR 30
2.12 กราฟการเรียนรู้ของโอเปอเรชั่น Exclusive-OR 33
2.13 การแทนที่โครงข่ายด้วยโมเดลของ McCulloch และ Pitts สำหรับแก้ไขปัญหาโอเปอเรชั่น Exclusive-OR 34
2.14 ขอบเขตการตัดสินใจโดยนิวรอนต่างๆ 35
2.15 องค์ประกอบภาษาจาวา 40
2.16 ขั้นตอนการทำงานของภาษาจาวา 41
2.17 Java Platform 42
2.18 หน้าต่างการทำงานของเน็ตบีนส์ 43
2.19 Java Enterprise Edition Version 6 44
2.20 ไดอะแกรมแสดงหลักการทำงานของเจเอสพีและเซิร์ฟเลต 45
3.1 ตัวอย่างแบบสอบถามที่ใช้ในการเก็บข้อมูลนักศึกษา 51
3.2 แผนผังการเชื่อมต่อของเว็บแอพพลิเคชั่นสำหรับผู้ใช้งาน 53
3.3 แผนผังการเชื่อมต่อของโปรแกรมสำหรับผู้พัฒนา 53
3.4 หน้าใช้งานหลักของเว็บแอพพลิเคชั่น 54
3.5 ส่วนเมนูของเว็บแอพพลิเคชั่น 54

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.6 ส่วนเมนูของเว็บแอพพลิเคชั่นบนอุปกรณ์สมาร์ทโฟน 54
3.7 หน้าแสดงผลลัพธ์ ผลการเรียนของผู้ใช้งาน 55
3.8 หน้าต่างใช้งานส่วน ABOUT 56
3.9 หน้าต่างใช้งานส่วน CE CURRICULUM 56
3.10 ปุ่มแสดงแผนการเรียน ในหน้าต่างใช้งานส่วน CE CURRICULUM 57
3.11 ปุ่ม Download และ ภาพกิจกรรมของภาควิชาวิศวกรรมคอมพิวเตอร์ 57
3.12 หน้าต่างอธิบายการทำงานของโครงข่ายประสาทเทียม 58
3.13 หน้าต่างใช้งานส่วน CONTACT US 58
3.14 หน้าหลักของโปรแกรมสำหรับผู้พัฒนา 59
3.15 ส่วนสำหรับการกำหนดไฟล์ที่ใช้ฝึกฝนและทดสอบ 59
3.16 ส่วนสำหรับตั้งค่าพารามิเตอร์ที่ใช้สำหรับฝึกฝนข้อมูล 60
3.17 กราฟแสดงความสัมพันธ์ระหว่างจำนวนรอบของการฝึกฝนและอัตราความผิดพลาด 61
3.18 หน้าต่างแสดงค่าน้ำหนักที่มีประสิทธิภาพมากที่สุด 61
3.19 การบันทึกค่าน้ำหนัก 62
3.20 การ FTP ไฟล์บันทึกค่าน้ำหนักไปยังเซิร์ฟเวอร์ 63
3.21 ตัวอย่างชุดข้อมูลเพื่อใช้ในการฝึกฝนและทดสอบโครงข่ายประสาทเทียม 64
3.22 โครงสร้างของระบบ 67
3.23 Data Flow Diagram ของโปรแกรมทำนายผลการเรียน 68
3.24 Data Flow Diagram Level 0 ของโปรแกรมทำนายผลการเรียน 68
3.25 Data Flow Diagram ระบบการฝึกฝนข้อมูล 69
3.26 Data Flow Diagram ของการสร้างโครงข่ายประสาทเทียม 69
3.27 Data Flow Diagramของระบบจัดการฝึกฝนข้อมูล 70
3.28 Data Flow Diagram ของการดำเนินงานเก็บข้อมูล 71
3.29 Data Flow Diagram level 0 ของการดำเนินงานเก็บข้อมูล 71
3.30 Data Flow Diagram ของการจัดการแบบสอบถาม 72
3.31 Data Flow Diagram ของการสร้างชุดข้อมูล 72

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.32 แผนภาพ Use-Case Diagram ของโปรแกรมทำนายผลการเรียน 74
3.33 แผนภาพ Class Diagram ของโปรแกรมทำนายผลการเรียน 75
3.34 Neural Network Diagram 75
4.1 แหล่งข้อมูลของชุดข้อมูล Iris 77
4.2 ภาพการฝึกฝนของโครงข่ายประสาทเทียม บนโปรแกรมเนตบีนส์ 78
4.3 การเก็บชุดข้อมูลบนโปรแกรม Excel 79
4.4 ผลการทดลองการตั้งค่าโครงข่ายประสาทเทียม ชุดที่ 1 82
4.5 ผลการทดลองการตั้งค่าโครงข่ายประสาทเทียม ชุดที่ 2 82
4.6 ผลการทดลองการตั้งค่าโครงข่ายประสาทเทียม ชุดที่ 3 83
4.7 ผลการทดลองการตั้งค่าโครงข่ายประสาทเทียม ชุดที่ 4 83
4.8 ผลการทดลองการตั้งค่าโครงข่ายประสาทเทียม ชุดที่ 5 84
4.9 ผลการทดลองการตั้งค่าอัตราการเรียนรู้ ชุดที่ 1 85
4.10 ผลการทดลองการตั้งค่าอัตราการเรียนรู้ ชุดที่ 2 85
4.11 ผลการทดลองการตั้งค่าอัตราการเรียนรู้ ชุดที่ 3 86
4.12 หน้าต่างส่วนติดต่อผู้ใช้งานเพื่อตั้งค่าโครงข่ายประสาทเทียม 87
4.13 ซอฟต์แวร์โครงข่ายประสาทเทียม ส่วนของชุดข้อมูล 87
4.14 การเรียกไฟล์ของส่วนชุดข้อมูลสำหรับฝึกฝน 87
4.15 การเรียกไฟล์ของส่วนชุดข้อมูลสำหรับการทดสอบ 88
4.16 ส่วนของชุดข้อมูลเมื่อเลือกไฟล์แล้ว 88
4.17 การตั้งค่าการทำงานของโครงข่ายประสาทเทียม 89
4.18 ซอฟต์แวร์ระหว่างการฝึกฝนโครงข่ายประสาทเทียม 89
4.19 หน้าต่างผลลัพธ์เมื่อกดปุ่ม WEIGHT 90
4.20 การเลือกเมนูส่วนการบันทึกไฟล์ ในหน้าต่างค่าน้ำหนัก 90
4.21 รูปเปรียบเทียบค่าน้ำหนักที่บันทึก และที่อยู่ในหน้าต่างค่าน้ำหนัก 91
4.22 การกรอก Host IP Username Password 92
4.23 Dialog Box ข้อความ Success เมื่ออัพโหลดสำเร็จ 92

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

4.24 การใช้งานเว็บแอพพลิเคชั่นบนอุปกรณ์ไอแพต 93
4.25 การใช้งานบนอุปกรณ์สมาร์ทโฟน 93
4.26 การใช้งานบนคอมพิวเตอร์โน้ตบุค 94
4.27 การกรอกข้อมูลสำหรับใช้ในการทำนาย 94
4.28 เว็บแอพพลิเคชั่นแสดงผลลัพธ์ 95
4.29 ทดลองแสดงผลบนบราวเซอร์ Chrome 95
4.30 ทดลองแสดงผลบนบราวเซอร์ Firefox 96
4.31 ทดลองแสดงผลบนบราวเซอร์ Internet Explorer 96

บทที่ 1
บทนำ

บทนำจะกล่าวถึงความสำคัญและที่มาของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน วิธีดำเนินการพัฒนา และประโยชน์ที่คาดว่าจะได้รับจากการพัฒนาส่วนติดต่อผู้ใช้งานสำหรับโปรแกรม ทำนายผลการเรียนล่วงหน้าด้วยโครงข่ายประสาทเทียม (Academic record prediction program by using artificial neural network)

1.1 ความสำคัญและที่มาของโครงงาน

ในการศึกษาของนักศึกษาปริญญาตรี ภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังในแต่ละปี จะมีนักศึกษาที่สามารถเรียนได้ดีในภาควิชาวิศวกรรมคอมพิวเตอร์ มีบางส่วนที่เรียนได้บ้างระดับกลาง และบางส่วนเรียนแล้วมีผลการเรียนอยู่ในระดับที่ไม่ค่อยดี

จากปัญหาดังที่กล่าวไป ทำให้เกิดแนวคิดของโปรแกรม ที่สามารถตรวจสอบได้ว่านักเรียนหรือ นักศึกษาที่ต้องการเข้ามาเรียนในภาควิชาวิศวกรรมคอมพิวเตอร์นั้น เมื่อเข้ามาเรียนแล้วจะมีผลการเรียนอยู่ในระดับใด โดยผู้จัดทำโครงงานได้คิดนำเอาการประมวลผลของ Artificial Intelligence ในส่วนของ neural network โดยเก็บข้อมูลมาจากนักศึกษาในภาควิชาวิศวกรรมคอมพิวเตอร์ ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปัจจุบันมาวิเคราะห์ โปรแกรมนี้จะช่วยอำนวยความสะดวก และช่วยในการตัดสินใจ แก่กรรมการคัดเลือกนักเรียน และนักเรียนที่จะเข้าศึกษาต่อในภาควิชาวิศวกรรมคอมพิวเตอร์ได้เป็นอย่างดี

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อศึกษาทฤษฎีโครงข่ายประสาทเทียม (Artificial Neural Network)
2) สร้างโปรแกรมที่ช่วยตรวจสอบว่าผู้ที่ต้องการเข้ามาศึกษาในภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง จะมีผลการเรียนอยู่ในระดับใด

1.3 ขอบเขตของโครงงาน

1) สร้างเว็บแอพพลิเคชั่น (Web Application) สำหรับผู้ใช้งาน และสามารถรับข้อมูลคะแนน GAT, PAT1, PAT2, PAT3 และวิธีการเข้าเช่น แอดมิชชั่น สอบตรง เป็นต้น เพื่อทำนายผลการเรียนที่ได้ ของชั้นปีที่ 1 ภาควิชาวิศวกรรมคอมพิวเตอร์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
2) ใช้โครงข่ายประสาทเทียมในส่วนของการทำนายผลการเรียน
3) ใช้ NetBeans IDE 7.2.1 เป็นเครื่องมือในการพัฒนาซอฟต์แวร์ด้วยภาษาจาวา (Java Programming Language)
4) ซอฟต์แวร์สำหรับผู้พัฒนาสามารถใช้งานผ่านส่วนประสานงานกับผู้พัฒนา (User Interface) ซึ่งใช้ประมวลผลและฝึกโครงข่ายประสาทเทียม และสามารถอัพเดทค่าน้ำหนัก (Weight) ไปยัง Web Applicaion สำหรับผู้ใช้งานได้
5) การทำงานของส่วน Backend อยู่บน PC ของผู้พัฒนา
6) โปรแกรมสามารถทำนายเฉพาะผลการเรียนในชั้นปีที่ 1 ของภาควิชาวิศวกรรมคอมพิวเตอร์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

1.4 วิธีการดำเนินการของโครงงาน

ในการจัดทำโครงงานนี้ ได้วางแผนการทำงาน โดยกำหนดช่วงเวลาดำเนินการตามตารางแผนการดำเนินการดังนี้

ลำดับ	Task Name	สิงหาคม				กันยายน				ตุลาคม				พฤศจิกายน			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	ศึกษาการทำงานของ Artificial neural network																
2	ออกแบบการใช้งานโปรแกรม																
3	ติดตั้งโปรแกรม Netbeans																
4	ทดลองเขียนโปรแกรมประมวลผลด้วย Neural Network																
5	ทดสอบโปรแกรมด้วย Dataset ตัวอย่าง																
6	เก็บข้อมูลสำหรับใช้ฝึกฝน Neural Network																

ลำดับ	Task Name	สิงหาคม				กันยายน				ตุลาคม				พฤศจิกายน			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
7	ทดสอบการฝึกฝนด้วยข้อมูลที่เก็บกับโปรแกรม																
8	นำการทำงานของโปรแกรมออกแบบเป็น GUI																
9	จัดทำวิทยานิพนธ์																
10	ทำไฟล์นำเสนอ , โปสเตอร์เพื่อเตรียมนำเสนอ																

ตารางที่ 1.1 การดำเนินการในภาคเรียนที่ 1

ลำดับ	Task Name	ธันวาคม				มกราคม				กุมภาพันธ์				มีนาคม				เมษายน			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	ทำการกรองข้อมูลเพื่อให้ค่า Error เหลือน้อยที่สุด																				
2	เขียนโปรแกรมส่วนที่จะแสดงผลใน GUI																				
3	ทดสอบการใช้งานของส่วนประกอบต่าง ๆ ของโปรแกรม																				
4	เขียน GUI																				
5	ทดสอบการใช้งานของโปรแกรมโดยรวม																				
6	ทดลองการเปลี่ยนแปลงค่าใน Neural Network กับผลลัพธ์ที่ได้																				
7	หาข้อผิดพลาดและแก้ไข																				
8	จัดทำวิทยานิพนธ์																				
9	ทำไฟล์นำเสนอ , โปสเตอร์เพื่อเตรียมนำเสนอ																				

ตารางที่ 1.2 การดำเนินการในภาคเรียนที่ 2

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) ผู้พัฒนามีความรู้ความเข้าใจในเรื่องของโครงข่ายประสาทเทียม และสามารถนำความรู้ความเข้าใจนี้ ไปประยุกต์ใช้งานต่อได้
2) โปรแกรมสามารถช่วยในการตัดสินใจแก่ผู้ที่สนใจที่จะเข้าเรียนในภาควิชาวิศวกรรมคอมพิวเตอร์
3) ช่วยกรรมการคัดเลือกนักเรียนที่คาดว่าจะเหมาะสมในการศึกษาต่อใน ภาควิชาวิศวกรรมคอมพิวเตอร์

1.6 ส่วนประกอบของปริญญานิพนธ์

เนื้อหาของปริญญานิพนธ์ฉบับนี้ประกอบด้วย 5 บท ได้แก่ บทนำ, ทฤษฎีที่เกี่ยวข้อง, การออกแบบและพัฒนา, การทดลองและผลการทดลอง และบทสรุป โดยมีรายละเอียดดังนี้

บทที่ 1 บทนำ กล่าวถึง ความสำคัญและที่มาของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน วิธีการดำเนินการ และผลที่คาดว่าจะได้รับ

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง กล่าวถึง ทฤษฎีพื้นฐานที่ใช้ในโครงงาน ประกอบด้วยข้อมูลเกี่ยวกับระบบ GAT / PAT วิธีการเข้าศึกษาในสถาบันฯ ทฤษฎีเกี่ยวกับโครงข่ายประสาทเทียม (Artificial Neural Network) สถิติศาสตร์ ภาษาจาวา และ เจเอสพีและเซิร์ฟเลต (JSP & Servlet)

บทที่ 3 การออกแบบและพัฒนา กล่าวถึง รายละเอียดของเครื่องมือที่ใช้ในการพัฒนา ภาพรวมของระบบ และ กระบวนการทำงานของระบบ

บทที่ 4 การทดลองและผลการทดลอง กล่าวถึง รายละเอียดของการทดลอง, การทดสอบการทำงานของโปรแกรม และผลลัพธ์ที่ได้ของโปรแกรม

บทที่ 5 บทสรุป กล่าวถึง บทสรุปของโครงงาน ผลที่ได้จากการทำโครงงาน ปัญหาและอุปสรรคที่พบ แนวทางการแก้ไขปัญหา และแนวทางการพัฒนาต่อ

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

บทนี้จะกล่าวถึงทฤษฎีพื้นฐานที่ใช้ในโครงงาน ประกอบด้วยข้อมูลเกี่ยวกับระบบ GAT / PAT วิธีการเข้าศึกษาในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทฤษฎีเกี่ยวกับโครงข่ายประสาทเทียม (Artificial Neural Network) สถิติศาสตร์ ภาษาจาวา เจเอสพีและเซิร์ฟเลต (JSP & Servlet)

2.1 ระบบ GAT / PAT

2.1.1 ประวัติความเป็นมา

ยุคเอนทรานซ์ 1.0 เกิดขึ้นในช่วงปีก่อน พ.ศ. 2541 คือการยื่นอันดับและสอบครั้งเดียวรู้ผลหลังจากจบ ม.6 เทอม 2 สอบทั้งหมดประมาณสิบวิชา คะแนนเต็มวิชาละ 100 คะแนน รวมคะแนนตามที่คณะที่ต้องเข้ากำหนดไว้ ทำให้ในสมัยนั้นเกิดปัญหาอยู่ทั้งหมด สามเรื่องด้วยกันคือ

1) เด็กทิ้งเกรดในห้องเรียน
2) ข้อสอบส่วนมากเป็นข้อกา ทำให้เด็กที่ไม่ตั้งใจทำ มีโอกาสได้คะแนนเฉลี่ยแล้วเท่ากับ 25%
3) เด็กไม่สนใจเรียนเพื่อให้รู้ แต่จะเรียนกวดวิชาหาสูตรลัดเพื่อเอาไปสอบเท่านั้น เมื่อเข้ามหาวิทยาลัยส่งผลให้ ไม่มีความรู้เพื่อที่จะไปต่อ

จึงมีการคิดแก้ปัญหา คือ

1) มีการนำ GPA (เกรดเฉลี่ย) ในห้องเรียนมาคิดรวมในคะแนนสอบด้วย 10%
2) ออกข้อสอบให้ยากขึ้น และให้ตอบซับซ้อนมากขึ้นกว่าข้อกา และเติมคำตอบ
3) ออกข้อสอบให้แหวกแนวและซับซ้อนมากขึ้น ไม่สามารถใช้สูตรลัดได้

ยุคเอนทรานซ์ 2.0 เกิดขึ้นในช่วง พ.ศ. 2542 – 2548 มีการใช้สอบเอนทรานซ์เช่นเดิม แต่มีการปรับปรุงคือ จากเดินซึ่งสอบได้รอบเดียว เปลี่ยนเป็นสอบสองรอบ ก่อนจะนำคะแนนที่ได้เพื่อไปยื่นอันดับ

ในยุคนี้เกิดปัญหาเรื่อง GPA ที่นำมาคิดคะแนนรวม เพราะมีหลายโรงเรียนให้เกรดเฉลี่ย 4.00 กันเป็นจำนวนมาก แต่บางโรงเรียนก็ยังคงมาตรฐานเดิมอยู่คือให้เกรดเฉลี่ย 2.50 ทำให้เกิดความไม่ยุติธรรมขึ้น ซึ่งจริงๆ ก็มีค่าๆ หนึ่งที่เอาไว้สำหรับแก้ปัญหานี้โดยเฉพาะ นั่นคือสัมประสิทธิ์ของแต่ละโรงเรียน โรงเรียนที่เด็กเก่งค่าก็จะอยู่กลุ่มมาก โรงเรียนที่เด็กอ่อนค่าก็น้อย เอามาคูณกับ

เกรด เป็นคะแนนอีกที แต่ประเด็นนี้ทำให้เกิดการถกเถียงเรื่องความไม่โปร่งใสของค่าสัมประสิทธิ์นี้ ว่าทำไมโรงเรียนนี้ได้เยอะ ได้น้อย สุดท้ายแล้วเพื่อให้ได้มาตรฐานมากขึ้น จึงเกิดการจัดตั้งองค์กรอิสระแห่งใหม่เพิ่มขึ้นมาในปี พ.ศ.2549 นั่นคือสถาบันทดสอบทางการศึกษาแห่งชาติ (สทศ.)

รูปที่ 2.1 สถาบันทดสอบทางการศึกษาแห่งชาติ

ถือเป็นการเปลี่ยนแปลงครั้งใหญ่ในประวัติศาสตร์การศึกษาไทย โดยมีการจัดตั้งหน่วยงานขึ้นมาใหม่เพื่อทำการจัดสอบนักเรียนโดยเฉพาะ ซึ่งสทศ. ได้มีการจัดแจงเปลี่ยนชื่อการสอบเอนทรานซ์ ไปเป็นการสอบ แอดมิชชั่น (Admission) และเปลี่ยนชื่อข้อสอบเอนทรานซ์ไปเป็น A-Net

O-Net ถูกคิดขึ้นในปี พ.ศ.2549 จุดประสงค์ของข้อสอบ O-Net คือเอาไว้วัดค่าสัมประสิทธิ์ของโรงเรียน เพื่อนำไปเฉลี่ยน้ำหนักกับ GPA เพื่อให้ได้คะแนนในส่วนนี้ขึ้นมา ทำให้คะแนนของนักเรียนทุกคนมีการวัดผลอย่างยุติธรรม มีการวัดผลสามครั้งคือ ป.6 ม.3 และ ม.6

ในช่วง พ.ศ. 2549 – 2552 ในตอนนั้นมีข้อสอบอยู่สองชุดด้วยกันคือ O-Net และเอนทรานซ์ ซึ่งชื่อของการสอบทั้ง 2 ชุดนี้ไม่เข้ากัน จึงเกิดการเปลี่ยนชื่อขึ้นมา เพื่อให้ฟังดูเข้าคู่กันมากขึ้น จากเอนทรานซ์ เปลี่ยนไปเป็น A-Net แต่อยู่ได้ไม่กี่ปี A-Net ก็โดนยุบ และเปลี่ยนระบบใหม่เป็น GAT/PAT

ช่วงเริ่มต้นของ GAT/PAT มีการเปลี่ยนแปลงมาก หลักๆ คือบางปีมีการสอบ GAT/PAT ถึงสามครั้งด้วยกัน แต่ในปีหลังเปลี่ยนเหลือเพียงแค่ปีละสองรอบ ด้วยเหตุผลที่ว่าออกข้อสอบไม่พอใช้งาน โดย GAT/PAT นี้เดิมทีเกิดขึ้นมาเพื่อวัตถุประสงค์ในการวัดผลนักเรียนจากความถนัดสองส่วนด้วยกันคือ GAT และ PAT นั่นเอง

2.1.2 GAT/PAT คืออะไร

ความถนัดทั่วไป (GAT: General Aptitude Test) คือ การวัดศักยภาพในการเรียนในมหาวิทยาลัยให้ประสบความสำเร็จ แยกได้ 2 ส่วน ส่วนละ 150 คะแนน คือ

1) ความสามารถในการอ่าน เขียน คิดวิเคราะห์ และแก้โจทย์ปัญหา 50 %
2) ความสามารถในการสื่อสารด้วยภาษาอังกฤษ 50 %

ความถนัดทางวิชาชีพและวิชาการ (PAT: Professional and Academic Aptitude Test) คือ ความรู้ที่เป็นพื้นฐานที่จะเรียนต่อในวิชาชีพนั้น ๆ กับศักยภาพที่จะเรียนในวิชาชีพนั้น ๆให้ประสบความสำเร็จ มี 7 ประเภท คือ

PAT 1 ได้แก่ ความถนัดทางคณิตศาสตร์ ประกอบด้วย 2 ส่วน คือ

1) ความรู้ในวิชาคณิตศาสตร์ พีชคณิต เรขาคณิต แคลคูลัส สถิติ ฯลฯ
2) ความถนัดในการเรียนคณิตศาสตร์ในมหาวิทยาลัยให้ประสบความสำเร็จ เช่น การคิดแบบนักคณิตศาสตร์ การแก้โจทย์ปัญหาคณิตศาสตร์ การอ่านเรื่องทางคณิตศาสตร์แล้วเข้าใจแก้ปัญหาตามกระบวนการคณิตศาสตร์ เป็นต้น

PAT 2 ได้แก่ ความถนัดทางวิทยาศาสตร์ ประกอบด้วย 2 ส่วน คือ

1) ความรู้ในทางวิทยาศาสตร์ที่จะเรียนในคณะวิทยาศาสตร์ และคณะอื่นที่เกี่ยวข้องได้ เช่น ความรู้ในเรื่องเคมี ชีววิทยา ฟิสิกส์ ดาราศาสตร์ เทคโนโลยีสารสนเทศ เป็นต้น
2) ความถนัดในการเรียนวิทยาศาสตร์ในมหาวิทยาลัยประสบผลสำเร็จเช่นการคิดแบบนักวิทยาศาสตร์ การแก้ปัญหาด้วยวิธีการทางวิทยาศาสตร์ ฯลฯ

PAT 3 ได้แก่ ความถนัดทางวิศวกรรมศาสตร์ ประกอบด้วย 2 ส่วน คือ

1) ความรู้พื้นฐานที่จะเรียนต่อในคณะวิศวกรรมศาสตร์สำเร็จ เช่น ความรู้ทางคณิตศาสตร์ วิทยาศาสตร์ เทคโนโลยี เป็นต้น
2) ความถนัดในการเรียนวิศวกรรมในมหาวิทยาลัยประสบความสำเร็จ เช่น การคิดแบบวิศวกร การแก้ปัญหาทางวิศวกรรม เป็นต้น

PAT 4 ได้แก่ ความถนัดทางสถาปัตยกรรมศาสตร์ ประกอบด้วย 2 ส่วน คือ

1) ความรู้พื้นฐานที่จะเรียนต่อในคณะสถาปัตยกรรมศาสตร์สำเร็จ เช่น ความรู้ทางคณิตศาสตร์ วิทยาศาสตร์ ศิลปกรรม ฯลฯ
2) ความถนัดในการเรียนในคณะสถาปัตยกรรมในมหาวิทยาลัยประสบความสำเร็จ เช่น มองเห็นภาพ 3 มิติในใจ การออกแบบ ฯลฯ

PAT 5 ได้แก่ ความถนัดทางครู ประกอบด้วย 2 ส่วน คือ

1. ความรู้พื้นฐานที่จะเรียนต่อในคณะครุศาสตร์/ศึกษาศาสตร์สำเร็จ เช่น ความรู้พื้นฐานทางคณิตศาสตร์ วิทยาศาสตร์ สังคม ภาษาไทย ภาษาอังกฤษ ฯลฯ
2. ความถนัดในการเรียนในคณะครุศาสตร์/ศึกษาศาสตร์สำเร็จ หรือแววในการจะเป็นครู เช่น ความสามารถในการแสวงหาความรู้ ทักษะสื่อสารรู้เรื่อง ฯลฯ

PAT 6 ได้แก่ ความถนัดทางศิลปะ ประกอบด้วย 2 ส่วน คือ

1. ความรู้ในทฤษฎีทัศนศิลป์ นาฏศิลป์ ดนตรี และความรู้อื่นที่เป็นพื้นฐานที่จะเรียนในคณะศิลปกรรม หรือที่เกี่ยวข้องประสบความสำเร็จ
2. ความถนัดในการเรียนศิลปะ เช่น ความคิดสร้างสรรค์ ฯลฯ

PAT 7 ได้แก่ ความถนัดในการเรียนภาษาต่างประเทศ ประกอบด้วย 2 ส่วน คือ

1. ความรู้เรื่องไวยากรณ์ หลักภาษา วรรณกรรม วรรณคดี ฯลฯ
2. ความสามารถในการฟัง พูด อ่าน เขียน สรุป ย่อความ ขยายความ สังเคราะห์ วิเคราะห์ ฯลฯ มี 6 ภาษา คือ ก) ภาษาฝรั่งเศส ข) ภาษาเยอรมัน ค) ภาษาญี่ปุ่น ง) ภาษาจีน จ) ภาษาบาลี ฉ) ภาษาอาหรับ

2.1.3 แต่ละคณะใช้คะแนนอะไรบ้าง

ในระบบแอดมิชชั่นกลางนั้น คณะต่างๆ จะใช้คะแนนสอบ GAT/PAT ในการคำนวณคะแนนที่แตกต่างกัน ซึ่งสรุปตามกลุ่มวิชาคร่าวๆ ดังนี้

PAT 1 ความถนัดทางคณิตศาสตร์ กลุ่มคณะที่ใช้ เช่น กลุ่มคณะวิทยาศาสตร์กายภาพ กลุ่มเกษตร-วนศาสตร์ กลุ่มบริหาร-บัญชี เศรษฐศาสตร์ กลุ่มมนุษยฯ-อักษรฯ-สังคมศาสตร์ (ยื่นคะแนนรูปแบบที่1)

PAT 2 ความถนัดทางวิทยาศาสตร์ กลุ่มคณะที่ใช้ เช่น กลุ่มวิทยาศาสตร์สุขภาพ กลุ่มวิทยาศาสตร์กายภาพ คณะวิศวกรรมศาสตร์

PAT 3 ความถนัดทางวิศวกรรมศาสตร์ มีคณะเดียวที่ใช้ คือ คณะวิศวกรรมศาสตร์

PAT 4 ความถนัดทางสถาปัตยกรรมศาสตร์ มีคณะเดียวที่ใช้ คือ คณะสถาปัตยกรรมศาสตร์

PAT 5 ความถนัดทางวิชาชีพครู มีคณะเดียวที่ใช้ คือ คณะครุศาสตร์/ศึกษาศาสตร์

PAT 6 ความถนัดทางศิลปกรรมศาสตร์ มีคณะเดียวที่ใช้ คือ คณะศิลปกรรมศาสตร์

PAT 7 ความถนัดทางภาษาต่างประเทศ กลุ่มคณะที่ใช้ เช่น กลุ่มการโรงแรมและกลุ่มมนุษยฯ-อักษรฯ-สังคมศาสตร์

2.1.4 ใครสอบ GAT/PAT ได้บ้าง

สำหรับการสอบ GAT/PAT ค่อนข้างจะอิสระพอสมควร เพราะทุกคนมีสิทธิในการสอบหมดทั้ง ม.6 เด็กซิ่ว และ สายอาชีพ และจะสมัครกี่ครั้งก็ได้ (หนึ่งปีมีแค่ 2 ครั้ง) เพราะการสมัครสอบสามารถดำเนินการสมัครได้ด้วยตนเอง ดังนั้นอยู่ที่ความรับผิดชอบของตัวเอง หากสมัครไม่ทัน ผลเสียก็ตกอยู่ที่ตัวผู้สอบเอง อายุของคะแนนสอบ GAT/PAT อยู่ได้ 2 ปี นั่นหมายความว่า หาก ม.6 ปีนี้อยากซิ่วในปีหน้า คะแนนของปีนี้ก็ยังใช้ได้ โดยในการสมัครแอดมิชชั่นกลางจะมีระบุไว้ว่าใช้รอบใดได้บ้าง แต่ในรับตรงบางคณะ จะกำหนดไว้ว่าใช้คะแนนรอบไหนได้บ้าง เพราะฉะนั้นรายละเอียดทุกอย่างของแต่ละคณะ ผู้สอบต้องใช้ความรอบคอบและถี่ถ้วนด้วยตัวเอง

2.2 วิธีการเข้าศึกษาในสถาบันฯ

2.2.1 แอดมิชชั่น (Admission)

แอดมิชชั่น (Admission) มีชื่อเต็มว่า ระบบกลางคัดเลือกบุคคลเข้าศึกษาในสถาบันอุดมศึกษา (Central University Admissions System: CUAS) คือระบบสอบกลางที่นักเรียนระดับชั้นมัธยมศึกษาปีที่ 6 หรือเทียบเท่า ที่มีความต้องการจะศึกษาต่อในระดับมหาวิทยาลัย จึงจำเป็นต้องสอบแล้วนำคะแนนสอบที่ได้มายื่นเลือกคณะที่ต้องการเข้าศึกษาต่อไป

ระบบแอดมิชชั่นนี้ บริหารงานโดย สถาบันทดสอบการศึกษาแห่งชาติ (องค์การมหาชน) หรือที่รู้จักกันดีในชื่อว่า สทศ. โดยที่หน้าที่ของสทศ. คือพัฒนาข้อสอบเพื่อวัดและประเมินมาตรฐานการศึกษา วัดความรู้ความสามารถของผู้เข้าสอบแต่ละคน สทศ.จะรับผิดชอบการประเมินผลด้านการศึกษาให้กับนักเรียนที่เรียนหลักสูตรไทย ในประเทศไทย หลายครั้งด้วยกันคือ ป.3, ป.6, ม.3, และ ม.6 แต่ในการสอบแอดมิชชั่นนั้นจะนับกันเฉพาะ การสอบวัดผลในระดับชั้น ม.6 เท่านั้น

องค์การหนึ่งที่มีผลต่อระบบการศึกษาคือ กระทรวงศึกษาธิการ ซึ่งเป็นผู้วางมาตรฐานหลักสูตรการศึกษาให้กับโรงเรียนต่างๆในประเทศไทย หรือในอีกความหมายคือ กระทรวงศึกษาเป็นผู้กำหนดหลักสูตร โรงเรียนมีหน้าที่สอนนักเรียนตามหลักสูตร และสทศ.มีหน้าที่สอบวัดผลโรงเรียนและนักเรียนแต่ละคนตามหลักสูตร

2.2.1.1 วิชาที่สอบในแอดมิชชั่น

1) O-NET (Ordinary National Education Test) หรือการสอบทางการศึกษาแห่งชาติขั้นพื้นฐาน ในตอนนี้จะพูดถึงการสอบ O-NET ในระดับชั้น ม.6 เพียงอย่างเดียว แนวคิดของ O-NET คือ การวัดผลของโรงเรียนแต่ละโรงเรียนว่า ได้สอนนักเรียนของตัวเองตามหลักสูตรกระทรวงขนาดไหน ข้อสอบ O-NET นี้จะเป็นข้อสอบง่ายๆที่วัดเฉพาะพื้นฐานจริงๆเท่านั้น

2) GAT (General Aptitude Test) หรือมีชื่อเป็นภาษาไทยสั้นๆว่า การสอบความถนัดทั่วไป ซึ่งจะเน้นเนื้อหาทางด้าน การอ่าน เขียน คิดวิเคราะห์การแก้โจทย์ปัญหาทางคณิตศาสตร์ รวมไปถึงการสื่อสารด้วยภาษาอังกฤษ ข้อสอบ GAT นี้จะมีความซับซ้อนมากกว่าความยาก
3) PAT (Professional Aptitude Test) หรือมีชื่อเป็นภาษาไทยสั้นๆว่า การสอบความถนัดเฉพาะด้าน/วิชาการ เป็นข้อสอบที่ยากที่สุดในสามตัวที่พูดถึง วิชาเฉพาะด้านที่มีสอบคือ คณิตศาสตร์ วิทยาศาสตร์ พื้นฐานวิศวกรรม พื้นฐานสถาปัตยกรรม พื้นฐานความเป็นครู และวิชาด้านภาษาอื่นๆนอกเหนือจากภาษาอังกฤษ

2.2.1.2 หลักเกณฑ์การคิดคะแนนในระบบแอดมิชชั่น

การคัดเลือกบุคคลเข้าศึกษาในสถาบันอุดมศึกษาในระบบแอดมิชชั่น ที่ประชุมอธิการบดีแห่งประเทศไทยได้กำหนดให้มีการพิจาณาคัดเลือกผู้สมัครโดยใช้

1) ผลการเรียนเฉลี่ยสะสมตลอดหลักสูตรมัธยมศึกษาตอนปลายหรือเทียบเท่า มีค่าน้ำหนักร้อยละ 20
2) ผลการสอบทางการศึกษาขั้นพื้นฐาน (O NET) ให้ค่าน้ำหนักร้อยละ 30
3) ผลการสอบความถัดทั่วไป (GAT) ให้ค่าน้ำหนักร้อยละ 10 - 50 แล้วแต่ภาควิชาที่ผู้สมัครเลือก
4) ผลการสอบความถนัดทางวิชาชีพและวิชาการ (PAT) ให้ค่าน้ำหนักร้อยละ 0 – 40
5) ผลการสอบสัมภาษณ์และตรวจร่างกาย ใช้เป็นข้อมูลประกอบการพิจารณาความพร้อมและความเหมาะสมก่อนรับการศึกษา ไม่คิดค่าน้ำหนักคะแนน

2.2.2 สอบตรง

ทางมหาลัยจะมีการจัดสอบเอง ซึ่งจะสอบอย่างไรบ้างนั้น ขึ้นอยู่กับภาควิชาจะกำหนด อาจจะมีได้ตั้งแต่วิชาความถนัด แล้วเอามารวมกับเกรดเฉลี่ย หรือผลสอบ O-net หรืออาจจะพิจารณาโดยใช้ผลสอบ O-net เพียงอย่างเดียวอาจจะจัดสอบวิชาเดียวก็ได้ ขึ้นอยู่กับภาควิชานั้นๆหรือขึ้นอยู่กับมหาลัยจะจัดสอบขึ้น

2.2.2.1 ข้อแตกต่างระหว่างการรับตรงและการสอบแบบ Admissions

1) รับตรง เปิดรับสมัครเองโดยมหาวิทยาลัย หรือ คณะของทางมหาวิทยาลัยการรับตรงก็มีด้วยกันหลายชื่อนะครับ เช่น รับตรง, รับตรงวิธีพิเศษ, โควตา เป็นต้น แต่การรับตรงทุกอย่างขึ้นอยู่กับมหาลัยเป็นฝ่ายพิจารณา

2) แอดมิชชั่นกลาง เป็นระบบรับนักศึกษา ที่ที่จัดขึ้นโดยหน่วยงานกลางใช้ระบบนี้ทั้งมหาลัยที่เป็นรัฐบาลและเอกชน คะแนนหลักๆที่ใช้พิจารณาก็คือ GPAX, O-NET, GAT, PAT ซึ่งในแต่ละปีจะมีการเปิดรับเป็นจำนวนมาก

2.2.3 โควต้า

โควตาคือการเข้ามหาวิทยาลัยโดยใช้เกรดเฉลี่ยสะสมตลอดหลักสูตรมัธยมศึกษาตอนปลายหรือเทียบเท่า หรือการเข้ามหาวิทยาลัยโดยใช้เกณฑ์พิเศษ เช่น ความสามารถพิเศษด้านวิชาวิทยาศาสตร์ เป็นต้น หรือภูมิลำเนาในพื้นที่เขตบริการของมหาวิทยาลัยนั้นๆ ซึ่งเกณฑ์หรือวิธีในการรับนั้นขึ้นอยู่กับแต่ละคณะและแต่ละมหาลัยเอง ซึ่งโควตาส่วนมากที่พบกับนักศึกษาในภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง มีดังนี้

1) โควต้าช้างเผือก เป็นโครงการที่ให้โอกาสพิเศษแก่นักเรียน และสอบได้คะแนนเฉลี่ยสะสมในชั้น ม.4, ม.5 เป็นอันดับต้นของกลุ่มนักเรียนสายวิทยาศาสตร์ตามเกณฑ์คุณสมบัติของโรงเรียน และต้องได้คะแนนเฉลี่ยในการสอบวิชาคณิตศาสตร์, เคมี, ฟิสิกส์และอังกฤษไม่ต่ำกว่า 3.5 ในชั้น ม.4, ม.5
2) โควตานักกีฬา คือโควต้าที่เปิดโอกาสพิเศษให้แก่นักเรียนผู้มีความสามารถดีเด่นด้านกีฬา
3) โควตาบุคลากร เป็นโควตาที่ให้โอกาสแก่บุตรของบุคลากรที่ทำหน้าที่อยู่ในมหาวิทยาลัยหรือสถาบัน
4) โควตามูลนิธิส่งเสริมโอลิมปิกวิชาการและพัฒนามาตรฐานวิทยาศาสตร์ศึกษา (สอวน.) เป็นโควต้าสำหรับนักเรียนที่สอบผ่านค่าย สอวน.

2.3 โครงข่ายประสาทเทียม (Artificial Neural Network)

2.3.1 โครงข่ายประสาทเทียมคืออะไร

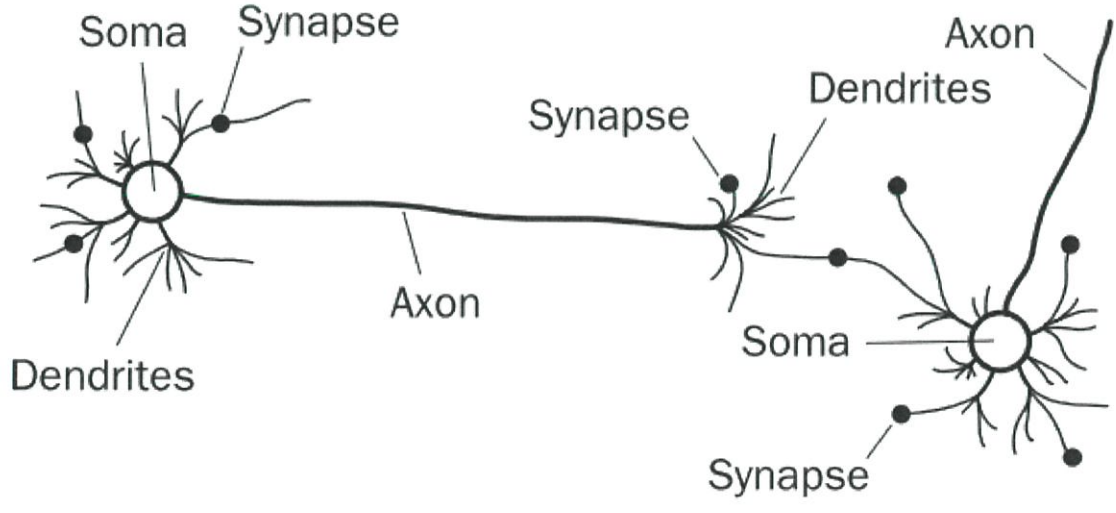
โครงข่ายประสาทเทียมถูกนิยามเป็นโมเดลของการใช้เหตุผลซึ่งมีพื้นฐานเหมือนกับสมองของมนุษย์สมองประกอบด้วยเซลล์ประสาทที่เชื่อมต่อกันอย่างแน่นหนาหรือหน่วยประมวลผลข้อมูลขั้นพื้นฐานที่ถูกเรียกว่า เซลล์ประสาท (Neuron) สมองมนุษย์ประกอบด้วยเซลล์ประสาทเกือบ 10 พันล้านเซลล์และไซแนปซ์ (synapse) อีกกว่า 60 ล้านล้านจุด เมื่อมีการใช้เซลล์ประสาทหลายๆ เซลล์พร้อมกัน สมองสามารถทำหน้าที่ได้อย่างมีประสิทธิภาพและรวดเร็วกว่าคอมพิวเตอร์ที่เร็วที่สุดที่มีอยู่ในปัจจุบันนี้

ถึงแม้ว่าแต่ละเซลล์ประสาทจะมีโครงสร้างที่เรียบง่าย แต่ด้วยองค์ประกอบที่มีจำนวนมหาศาลสามารถก็ให้เกิดการประมวลผลที่มีประสิทธิภาพสูงมาก เซลล์ประสาทประกอบไปด้วย เซลล์บอดี้ (cell body) , เดนไดรซ์ (dendrite) และ แอกซอน (axon) ดังรูปที่ 2.2

สัญญาณจะถูกแพร่กระจายจากเซลล์ประสาทหนึ่งไปยังอีกเซลล์ประสาทหนึ่งด้วยปฏิกิริยาไฟฟ้าเคมีที่ซับซ้อน สารเคมีที่ปล่อยออกจากประสาททำให้เกิดการเปลี่ยนแปลงของศักย์ไฟฟ้าในเซลล์บอดี้ เมื่อศักย์ไฟฟ้าที่เกิดขึ้นเพิ่มถึงเกณฑ์ที่เหมาะสมแล้วหรือค่าเทรสโฮลด์ (threshold) พลังงานไฟฟ้าจะถูกส่งผ่านมายังแอกซอนแล้วกระจายออกไปจนถึงไซแนปซ์ ทำให้ศักย์เหล่านั้นเพิ่มขึ้นหรือลดลง อย่างไรก็ตาม สิ่งที่เราพบและน่าสนใจไปอย่างยิ่งคือการปรับตัวของโครงข่ายประสาท ในรูปแบบของการตอบสนองนั้น เซลล์ประสาทได้แสดงให้เห็นถึงการเปลี่ยนแปลงและการปรับตัวของการเชื่อมต่อระหว่างเซลล์ต่างๆ ให้แข็งแกร่งยิ่งขึ้น นอกจากนี้มันยังสามารถสร้างการเชื่อมต่อใหม่กับเซลล์ประสาทอื่นๆ ได้ ถึงแม้ว่ากลุ่มของเซลล์ประสาทเหล่านั้นจะถูกโยกย้ายไปอยู่ที่อื่นในบางครั้ง แต่กลไกการทำงานเหล่านี้ก็ยังคงเป็นพื้นฐานของการเรียนรู้ของสมองเช่นเดิม

สมองของพวกเราถือได้ว่าเป็นสิ่งที่มีความซับซ้อนเป็นอย่างมาก ข้อมูลจะถูกจัดเก็บและประมวลผลในระบบประสาทพร้อมกันทั่วทั้งโครงข่ายไม่ใช่เพียงเฉพาะเจาะจงแค่ส่วนใดส่วนหนึ่ง หรือเรียกได้ว่าโครงข่ายประสาท

เนื่องจากการปรับตัวที่เกิดขึ้นภายในโครงข่ายประสาท การเชื่อมต่อระหว่างเซลล์ประสาทจะทำให้เกิดการประมวลผลเพื่อให้ได้คำตอบที่ถูกต้องเพิ่มมากขึ้น และได้คำตอบที่ผิดลดลง เป็นผลทำให้โครงข่ายประสาทมีความสามารถในการเรียนรู้จากประสบการณ์ต่างๆ



รูปที่ 2.2 โครงข่ายประสาททางชีวภาพ

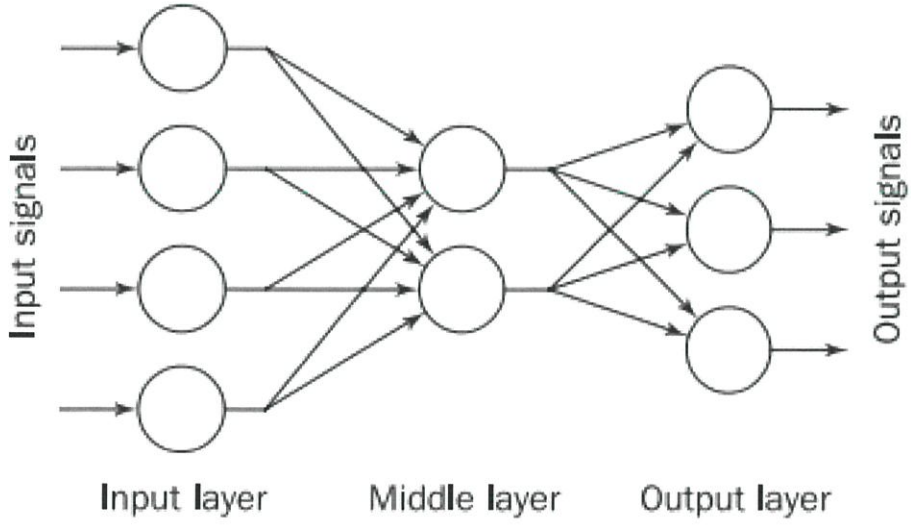
2.3.1.1 โครงข่ายประสาทเทียมทำงานแบบสมองได้อย่างไร

โครงข่ายประสาทเทียมประกอบด้วยจำนวนของหน่วยประมวลผลง่ายๆ ซึ่งเชื่อมต่อระหว่างกัน เราเรียกว่าสิ่งนี้ว่าเซลล์ประสาทหรือนิวรอน (neuron) ซึ่งเซลล์ประสาทนี้มีความคล้ายคลึงกับ

เซลล์ประสาทในทางชีวภาพที่อยู่ในสมองของเรา เซลล์ประสาทจะถูกเชื่อมต่อกันโดยลิงค์ที่มีน้ำหนักกำหนดไว้ (weighted links) เพื่อส่งผ่านสัญญาณจากเซลล์หนึ่งไปยังเซลล์อื่นๆ แต่ละเซลล์จะได้รับสัญญาณอินพุตผ่านการเชื่อมต่อเหล่านั้น อย่างไรก็ตามเซลล์เหล่านั้นจะไม่สามารถสร้างสัญญาณเอาต์พุตได้มากกว่าหนึ่งสัญญาณ เมื่อมีการนำสัญญาณเหล่านี้ไปประมวลผลต่อ จะทำการกระจายสัญญาณนี้ออกไปตามการเชื่อมต่อสาขาต่างๆ โดยที่สัญญาณเอาต์พุตนั้นจะไม่มีการปรับเปลี่ยนหรือลดทอนสัญญาณเพื่อกระจายออกแต่อย่างใด และกลายเป็นอินพุตของเซลล์ถัดไป เมื่อกระประมวลผลสิ้นสุดลง จะได้สัญญาณขาออก ซึ่งเป็นผลลัพธ์ ดังรูปที่ 2.3 แสดงให้เห็นถึงการเชื่อมต่อทั่วไปของโครงข่ายประสาทเทียม และตารางที่ 2.1 เป็นการเปรียบเทียบระหว่างโครงข่ายประสาทเทียมกับโครงสร้างทางชีวภาพ (Medsker and Leibovitz, 1994).

2.3.1.2 โครงข่ายประสาทเทียมทำการเรียนรู้อย่างไร

เซลล์ประสาทจะถูกเชื่อมต่อกันด้วยลิงค์ ซึ่งแต่ละลิงค์นั้นจะมีค่าน้ำหนักเป็นของตัวเอง น้ำหนักเหล่านี้หมายถึงพื้นฐานของหน่วยความจำในระยะยาวในโครงข่ายประสาทเทียม นับเป็นสิ่งที่ขาดไม่ได้และมีความสำคัญเป็นอย่างมากของแต่ละอินพุตเซลล์ โดยโครงข่ายประสาทนั้นจะสามารถเรียนรู้ผ่านการปรับน้ำหนักเหล่านี้ซ้ำไปซ้ำมาจนกระทั่งได้ค่าที่เหมาะสมที่สุด



รูปที่ 2.3 โครงสร้างตัวอย่างของโครงข่ายประสาทเทียม

Biological neural network	Artificial neural network
Soma	Neuron
Dendrite	Input
Axon	Output
Synapse	Weight

ตารางที่ 2.1 เปรียบเทียบระหว่างโครงข่ายประสาทเทียมกับโครงสร้างทางชีวภาพ

2.3.1.3 โครงข่ายประสาทเทียมสามารถปรับน้ำหนักให้เหมาะสมได้อย่างไร

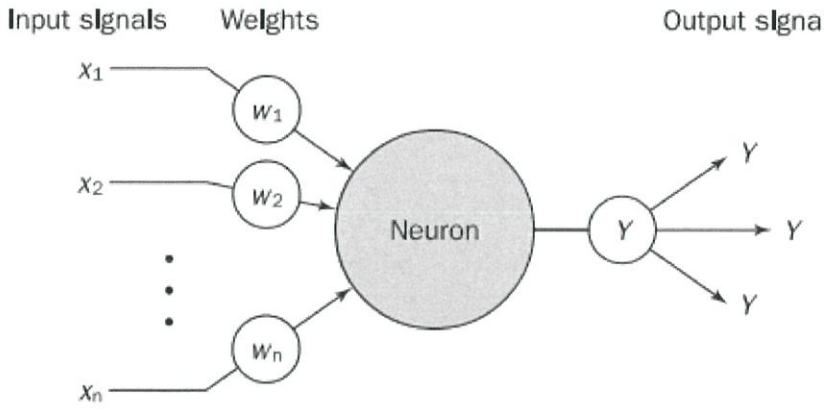
จากรูปที่ 2.3 โดยทั่วไป โครงข่ายประสาทเทียมถูกสร้างขึ้นเป็นเลเยอร์หลายๆชั้น และนิวรอนในโครงข่ายจะนำมาจัดระเบียบอยู่บนชั้นเลเยอร์เหล่านั้น นิวรอนจะเชื่อมต่อกับสภาพแวดล้อมภายนอกในรูปแบบของอินพุตและเอาต์พุตเลเยอร์ น้ำหนักต่างๆ จะถูกแก้ไขและปรับตัวเพื่อให้พฤติกรรมของอินพุตและเอาต์พุตของโครงข่ายเป็นไปในแนวโน้มเดียวกับสภาพแวดล้อม

แต่ละนิวรอนเป็นหน่วยประมวลผลเบื้องต้น ซึ่งสามารถใช้ในการคำนวณ Activation level โดยใช้ข้อมูลจากอินพุตและน้ำหนัก

ในการสร้างโครงข่ายประสาทเทียมนั้น เราต้องตัดสินใจก่อนว่าจะใช้นิวรอนจำนวนเท่าไรและจะนำนิวรอนเหล่านั้นมาเชื่อมต่อเป็นรูปแบบโครงข่ายอย่างไร ความหมายคือ พวกเราต้องเลือกสถาปัตยกรรมโครงข่ายก่อนเป็นลำดับแรก จากนั้นเราก็ตัดสินใจว่าจะใช้อัลกอริทึมสำหรับการเรียนรู้แบบใด และสุดท้ายจะเป็นการฝึกโครงข่ายประสาท โดยจะต้องทำการกำหนดค่าเริ่มต้นของน้ำหนักในโครงข่ายและทำการปรับปรุงน้ำหนักจากชุดข้อมูลตัวอย่างที่เราได้นำมาฝึกให้กับโครงข่ายประสาทเทียม

2.3.2 นิวรอนเปรียบเสมือนองค์ประกอบในการคำนวณ

นิวรอนจะรับสัญญาณจากอินพุตลิงค์ เพื่อคำนวณ Activation level และส่งออกไปเป็นสัญญาณเอาต์พุตผ่านทางเอาต์พุตลิงค์ สัญญาณอินพุตอาจจะเป็นข้อมูลดิบหรือเป็นเอาต์พุตจากนิวรอนอื่นๆ ส่วนสัญญาณเอาต์พุตสามารถเป็นได้ทั้งคำตอบสุดท้ายของปัญหาหรือเป็นอินพุตให้กับนิวรอนอื่นๆ ดังรูปที่ 2.4



รูปที่ 2.4 ไดอะแกรมของนิวรอน

2.3.2.1 นิวรอนสามารถกำหนดเอาต์พุตได้อย่างไร

ในปี 1943 Warren McCulloch และ Walter Pitts ได้เสนอแนวคิดที่เรียบง่ายซึ่งเป็นพื้นฐานของโครงข่ายประสาทเทียมส่วนใหญ่

นิวรอนจะคำนวณผลรวมของน้ำหนักของสัญญาณอินพุตและเปรียบเทียบผลลัพธ์กับค่าเทรสโฮลด์ (Threshold) θ ถ้าผลลัพธ์นั้นมีค่าน้อยกว่าเทรสโฮลด์ เอาต์พุตที่ได้จะเป็น -1 แต่ถ้าผลลัพธ์ที่คำนวณได้มีค่ามากกว่าหรือเท่ากับค่าเทรสโฮลด์ นิวรอนจะถูกเปิดใช้งานและเอาต์พุตที่ได้จะเป็น +1 (McCulloch and Pitts, 1943)

กระบวนการดังกล่าวสามารถนำมาแปลงเป็นฟังก์ชั่นที่นิวรอนใช้ เรียกว่า Activation function

$$X = \sum_{i=1}^{n} x_i w_i \qquad (2.1)$$

$$Y = \begin{cases} +1 \;\; if \;\; X \geq \theta \\ -1 \;\; if \;\; X < \theta \end{cases}$$

เมื่อ X คือผลรวมของผลคูณระหว่างน้ำหนักกับอินพุต

x_i คือค่าของอินพุต i

w_i คือน้ำหนักของอินพุต i

n คือจำนวนของนิวรอนอินพุต

Y คือเอาต์พุตของนิวรอน

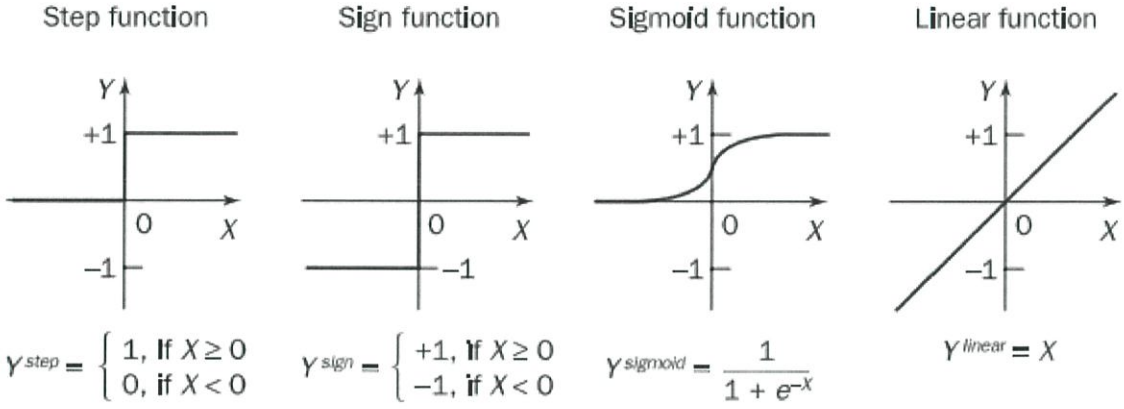
Activation function ประเภทนี้ เรียกว่า “sign function” ดังนั้นเอาต์พุตที่ได้จาก activation function สามารถถูกแสดงเป็น

$$Y = sign[\sum_{i=1}^{n} x_i w_i - \theta] \quad (2.2)$$

2.3.2.2 นิวรอนมีเพียง Sign function ที่ใช้ activation เพียงอย่างเดียวหรือไม่

Activation function จำนวนมากที่ได้ถูกทดสอบนั้น มีเพียงบางฟังก์ชั่นเท่านั้นที่ถูกนำมาใช้ในทางปฏิบัติจริงมี 4 แบบ ได้แก่ step, sign, linear และ sigmoid function ดังรูปที่ 2.5

Step และ sign function ถูกเรียกอีกชื่อหนึ่งว่า hard limit functions ซึ่งถูกมักใช้ในการตัดสินใจสำหรับการจัดหมวดหมู่และรูปแบบ ของงานที่ได้รับ



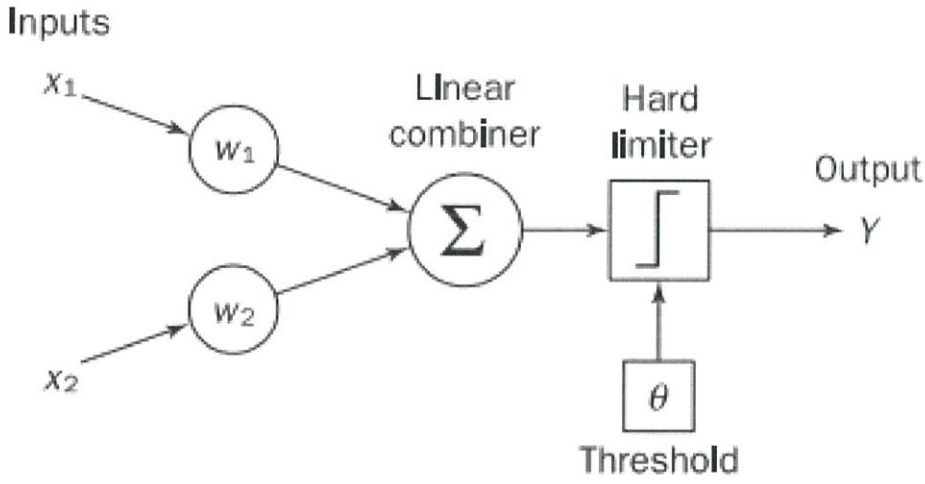
รูปที่ 2.5 Activation function ประเภทต่างๆ

Sigmoid Function จะทำการเปลี่ยนแปลงอินพุตที่จะมีค่าอยู่ระหว่างลบอินฟินิตีและบวกอินฟินิตี เป็น Reasonable value ในช่วงระหว่าง 0 และ 1 โครงข่ายประสาทเทียม ที่มีฟังก์ชันนี้จะถูกใช้ในการทำงานของ โครงข่าย Back Propagation

Linear activation function จะทำให้ค่าเอาต์พุตเท่ากับน้ำหนักของอินพุตของนิวรอน นิวรอนที่เป็น linear function จะถูกใช้สำหรับการประมาณค่าเชิงเส้น

2.3.2.3 ถ้ามีนิวรอนเพียงตัวเดียวจะสามารถทำงานได้หรือไม่

ในปี 1958 Frank Rosenblatt ได้นำเสนอวิธีอัลกอริทึมในการฝึกที่เป็นขั้นตอนแรกสำหรับในการฝึกโครงข่ายประสาทเทียมอย่างง่าย เรียกว่า เพอร์เซปตรอน (Perceptron) สำหรับเพอร์เซปตรอน (Rosenblatt, 1958) เป็นรูปแบบที่ง่ายที่สุดของโครงข่ายประสาท มันประกอบด้วย หนึ่งนิวรอนที่มีค่าน้ำหนักที่ปรับเปลี่ยนได้ และ hard limiter ดังรูปที่ 2.6



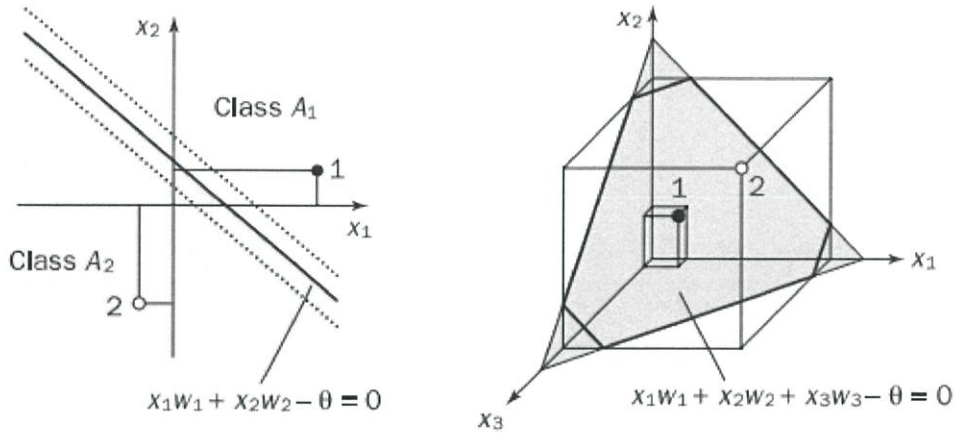
รูปที่ 2.6 Single-layer two-input perceptron

2.3.3 เพอร์เซปตรอน (Perceptron)

วิธีการเพอร์เซปตรอนของ Rosenblatt นั้น มีพื้นฐานมาจากโมเดลนิวรอนของ McCulloch และ Pitts โดยโมเดลนั้นประกอบด้วย ตัวรวมเชิงเส้นตามด้วย Hard limiter ผลรวมน้ำหนักของอินพุตจะถูกปรับใช้กับ Hard limiter ซึ่งเอาต์พุตที่ได้จะเท่ากับ +1 ถ้าผลรวมอินพุตมีค่าเป็นบวก และจะเท่ากับ -1 หากผลรวมอินพุตมีค่าเป็นลบ จุดมุ่งหมายของ เพอร์เซปตรอนนั้น เอาไว้ใช้สำหรับแบ่งกลุ่มของ input หรือในอีกความหมายคือการแบ่งกลุ่มของ $x_1, x_2, \ldots, x_n$ ให้เป็นหนึ่งใน 2 คลาส คือ A_1 และ A_2 ดังนั้น ในวิธีการขั้นต้นของเพอร์เซปตรอนกับมิติต่าง ๆ n มิติ จะถูกแบ่งโดย hyperplane เป็น 2 ส่วน โดย hyperplane จะถูกกำหนด โดย Linear Separable Function

$$\sum_{i=0}^{n} x_i w_i - \theta = 0 \qquad (2.3)$$

ในกรณีที่มีสองอินพุต เป็น x_1 และ x_2 เส้นขอบการตัดสินใจจะใช้รูปแบบของเส้นตรง แสดงเป็นเส้นเข้ม ดังรูปที่ 2.7(a) จุดที่ 1 ที่อยู่เหนือเส้น จะเป็นกลุ่ม A_1และจุดที่ 2 ที่อยู่ใต้เส้น จะเป็นกลุ่ม A_2 ค่าเทรสโฮลด์จะถูกใช้ในการเลื่อนเส้นขอบการตัดสินใจ



(a) (b)

รูปที่ 2.7 เส้นแบ่งของเพอร์เซปตรอน:
(a) two-input perceptron
(b) three-input perceptron

ถ้ามี 3 อินพุต จะเป็นรูปแบบของ hyperplane ในการแบ่งเป็นสามแกน สำหรับ Three-input perceptron ดังรูปที่ 2.7 (b) โดยระนาบแบ่ง จะถูกกำหนด จากสมการ

$$x_i w_i + x_i w_i + x_i w_i - \theta = 0$$

2.3.3.1 เพอร์เซปตรอนเรียนรู้การแบ่งกลุ่มได้อย่างไร

เพอร์เซปตรอนสามารถเรียนรู้การแบ่งกลุ่มได้จากการปรับเปลี่ยนค่าน้ำหนัก เพื่อลดความแตกต่างระหว่างค่าเอาต์พุตปัจจุบัน กับค่าเอาต์พุตที่ต้องการของเพอร์เซปตรอน ค่าน้ำหนักเริ่มต้นจะถูกสุ่มขึ้นมา เป็นค่าระหว่าง -0.5 ถึง 0.5 หลังจากนั้นจึงทำการอัพเดทค่าเอาต์พุตที่ได้ จากตัวอย่างการฝึก สำหรับเพอร์เซปตรอนกระบวนการของการปรับปรุงค่าน้ำหนักสามารถทำได้ง่าย ถ้าค่าการวนซ้ำ p ผลที่เกิดจริงเป็น $Y(p)$ และ ค่าเอาต์พุตที่ต้องการเป็น $Y_d(p)$ ค่าความผิดพลาด จะสามารถคิดได้จาก

$$e(p) = Y_d(p) - Y(p) \qquad \text{เมื่อ } p = 1,2,3,\ldots \qquad (2.4)$$

ค่าการวนซ้ำ p หมายถึง จำนวนรอบที่เราจะทำการฝึกเป็นรอบที่ p ตัวอย่างการฝึก จะถูกนำออกมาใช้โดยเพอร์เซปตรอน

ถ้าค่าความผิดพลาด $e(p)$ มีค่าเป็นบวก เราจำเป็นต้องเพิ่มค่าเอาต์พุตของเพอร์เซปตรอน $Y(p)$ แต่ถ้าค่าที่ได้ออกมาเป็นลบ เราต้องลดค่า $Y(p)$ ในการคำนวณค่าอินพุตของเพอร์เซปตรอนจะ

อยู่ในรูปแบบ $x_i(p) \times w_i(p)$ เพื่อนำมาคำนวณหาผลรวม $X(p)$ จะพบว่า ถ้าค่าอินพุต $x_i(p)$ เป็นบวก การเพิ่มค่าของน้ำหนัก $w_i(p)$ มีแนวโน้มที่จะส่งผลให้ค่าเอาต์พุตของเพอร์เซปตรอน $Y(p)$ เพิ่มขึ้น แต่ถ้าค่าอินพุต $x_i(p)$ เป็นลบ การเพิ่มค่าน้ำหนักของ $w_i(p)$ จะมีแนวโน้มส่งผลให้ค่า $Y(p)$ ลดลง ดังนั้นกฎการเรียนรู้ของเพอร์เซปตรอนจะมาจาก

$$w_i(p+1) = w_i(p) + \alpha \times x_i(p) \times e(p) \qquad (2.5)$$

เมื่อค่า α คืออัตราการเรียนรู้

กฎการเรียนรู้ของเพอร์เซปตรอน ถูกเสนอครั้งแรกโดย Rosenblatt ในปี 1960 ในการใช้กฎนี้ ทำให้เราสามารถใช้อัลกอริทึมในการฝึกเพอร์เซปตรอน สำหรับการแบ่งกลุ่มของข้อมูลได้

ขั้นตอนที่ 1: Initialization

กำหนดค่าเริ่มต้นให้กับน้ำหนัก $w_1, w_2, \dots, w_n$ และค่า threshold θ โดยการสุ่มค่าตัวเลขในช่วงระหว่าง -0.5 ถึง 0.5

ขั้นตอนที่ 2: Activation

ทำการเปิดใช้งานเพอร์เซปตรอน โดยการป้อนอินพุต $x_1(p), x_2(p), \dots, x_n(p)$ และค่าเอาต์พุตในอุดมคติ $Y_d(p)$ จากนั้นทำการคำนวณเพื่อหาค่าเอาต์พุตที่ได้ตามความจริง

$$Y(p) = step[\sum_{i=1}^{n} x_i(p) w_i(p) - \theta] \qquad (2.6)$$

เมื่อ n คือจำนวนของอินพุตเพอร์เซปตรอน และใช้รูปแบบสมการของ step activation function

ขั้นตอนที่ 3: Weight training

ทำการอัพเดทค่าน้ำหนักของเพอร์เซปตรอน

$$w_i(p+1) = w_i(p) + \Delta w_i(p) \qquad (2.7)$$

เมื่อ $\Delta w_i(p)$ คือค่าแก้ไขน้ำหนักที่รอบการทำงานที่ p ซึ่งสามารถคำนวณได้จาก delta rule

$$\Delta w_i(p) = \alpha \times x_i(p) \times e(p) \quad (2.8)$$

ขั้นตอนที่ 4: Iteration

กลับไปทำตามขั้นตอนที่ 2 ซ้ำอีกครั้งโดยเลื่อนรอบการทำงาน p และทำตามขั้นตอนนี้ซ้ำเรื่อยๆ จนกว่าจะถึงค่าลู่เข้า

จากตารางความจริงสำหรับโอเปอเรชั่น AND, OR และ Exclusive-OR ดังตารางที่ 2.2 ตารางนี้แสดงให้เห็นถึงความเป็นไปได้ทั้งหมดที่เกิดจากการทำงานรวมกันระหว่างค่าอินพุต x_1 และ x_2 และผลลัพธ์ของโอเปอเรชั่น เพอร์เซปตรอนจะต้องถูกฝึกโดยใช้รูปแบบของอินพุตดังนี้

Input variables		AND	OR	Exclusive-OR
x_1	x_2	$x_1 \cap x_2$	$x_1 \cup x_2$	$x_1 \oplus x_2$
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

ตารางที่ 2.2 Truth tables for the basic logical operations

ในส่วนนี้จะเป็นการอธิบายตัวอย่างการพิจารณาที่โอเปอเรชั่น AND หลังจากที่เราเสร็จสิ้นขั้นตอน Initialization เพื่อทำการกำหนดค่าเริ่มต้นให้กับน้ำหนัก และ threshold แล้ว เพอร์เซปตรอนจะถูกทำงานโดยการป้อนลำดับของอินพุตทั้ง 4 รูปแบบเข้าไปทีละรอบๆ น้ำหนักของเพอร์เซปตรอนจะถูกอัพเดทหลังผ่านการทำงานในแต่ละครั้ง กระบวนการนี้จะถูกทำซ้ำๆ จนกระทั่งได้ผลลัพธ์ที่ได้มีค่าลู่เข้ากับค่าผลลัพธ์ในอุดมคติ ดังตารางที่ 2.3

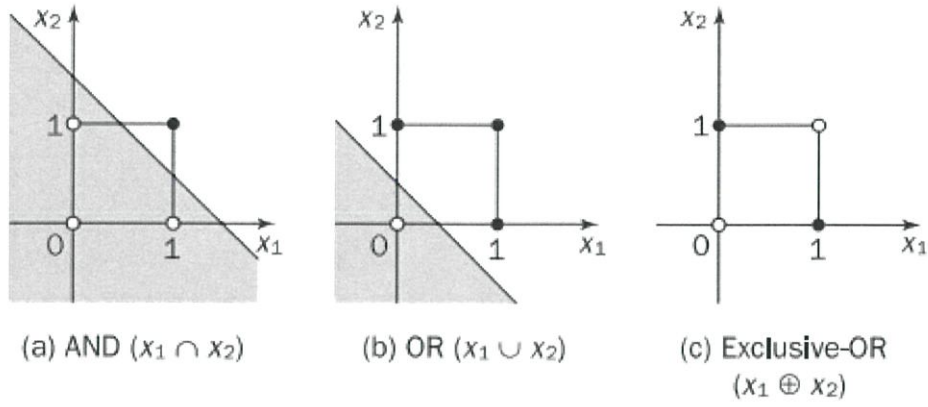
Epoch	Inputs		Desired Output	Initial weights		Actual Output	Error	Final weights	
	x_1	x_2	Y_d	w_1	w	Y	e	w_1	w_2
1	0	0	0	0.3	-0.1	0	0	0.3	-0.1
	0	1	0	0.3	-0.1	0	0	0.3	-0.1
	1	0	0	0.3	-0.1	1	-1	0.2	-0.1
	1	1	1	0.2	-0.1	0	1	0.3	0.0

Epoch	Inputs		Desired Output	Initial weights		Actual Output	Error	Final weights	
	x_1	x_2	Y_d	w_1	w	Y	e	w_1	w_2
2	0	0	0	0.3	0.0	0	0	0.3	0.0
	0	1	0	0.3	0.0	0	0	0.3	0.0
	1	0	0	0.3	0.0	1	-1	0.2	0.0
	1	1	1	0.2	0.0	1	0	0.2	0.0
3	0	0	0	0.2	0.0	0	0	0.2	0.0
	0	1	0	0.2	0.0	0	0	0.2	0.0
	1	0	0	0.2	0.0	1	-1	0.1	0.0
	1	1	1	0.1	0.0	0	1	0.2	0.1
4	0	0	0	0.2	0.1	0	0	0.2	0.1
	0	1	0	0.2	0.1	0	0	0.2	0.1
	1	0	0	0.2	0.1	1	-1	0.1	0.1
	1	1	1	0.1	0.1	1	0	0.1	0.1
5	0	0	0	0.1	0.1	0	0	0.1	0.1
	0	1	0	0.1	0.1	0	0	0.1	0.1
	1	0	0	0.1	0.1	0	0	0.1	0.1
	1	1	1	0.1	0.1	1	0	0.1	0.1

Threshold: θ *= 0.2; learning rate:* α *= 0.1.*

ตารางที่ 2.3 Example of perceptron learning: the logical operation AND

ในทำนองเดียวกัน เพอร์เซปตรอนก็สามารถเรียนรู้โอเปอเรชั่น OR ได้ อย่างไรก็ตาม single-layer perceptron ไม่สามารถถูกฝึกเพื่อให้สามารถดำเนินการโอเปอเรชั่น Exclusive-OR ได้ ตามหลักเรขาคณิตสามารถช่วยให้เราเข้าใจในเรื่องนี้ได้ว่าทำไม ดังรูปที่ 2.8 แสดงให้เห็นฟังก์ชั่นที่พลอตในรูปแบบ 2 มิติของโอเปอเรชั่น AND, OR และ Exclusive-OR โดยการป้อนค่า 2 อินพุต จุดที่อยู่ในพื้นที่ของการป้อนข้อมูลอินพุต และมีค่าเอาต์พุตของฟังก์ชั่นเป็น 1 จะแสดงเป็นจุดสีดำ และจุดที่แสดงค่าเอาต์พุตเป็น 0 คือจุดสีขาว



รูปที่ 2.8 Two-dimensional plots of basic logical operations

จากรูปที่ 2.8 (a) และ 2.8 (b) เราสามารถวาดเส้นเพื่อแยกฝั่งระหว่างจุดสีดำและจุดสีขาวให้อยู่ด้านใดด้านหนึ่งได้ แต่จะเห็นว่าจุดที่แสดงในรูปที่ 2.8 (c) ไม่สามารถถูกแบ่งโดยการวาดเส้นตรงได้ เพอร์เซปตรอนสามารถถูกนำมาใช้ได้เพียงฟังก์ชั่นที่สามารถแบ่งเขตระหว่างจุดดำทั้งหมดและจุดขาวทั้งหมดได้เท่านั้น ซึ่งฟังก์ชั่นเหล่านี้เราเรียกว่า linearly separable ดังนั้นเพอร์เซปตรอนจึงสามารถเรียนรู้โอเปอเรชั่น AND และ OR ได้ แต่ไม่สามารถเรียนรู้ Exclusive-OR

2.3.3.2 เหตุผลที่เพอร์เซปตรอนจึงสามารถเรียนรู้ได้เพียง Linearly separable functions

ในความจริงที่ว่าเพอร์เซปตรอนสามารถเรียนรู้ฟังก์ชั่น linearly separable ได้เท่านั้น จากสมการที่ 2.1 เอาต์พุตของเพอร์เซปตรอน Y จะมีค่าเป็น 1 ถ้าผลรวมของอินพุตน้ำหนัก X มีค่ามากกว่าหรือเท่ากับค่า threshold θ ซึ่งหมายความว่าพื้นที่ของอินพุตจะถูกแบ่งเขต โดย X = θ ตัวอย่างเช่น เส้นแบ่งของโอเปอเรชั่น AND ที่ถูกกำหนดโดยสมการดังนี้

$$x_1 w_1 + x_2 w_2 = \theta$$

ถ้าพวกเราแทนค่าสำหรับน้ำหนัก w_1และ w_2และค่า threshold θ ที่กำหนดดังตารางที่ 2.3 เราจะได้รับสมการของเส้นแบ่งมาคือ

$$0.1x_1 + 0.1x_2 = 0.2$$

หรือ

$$x_1 + x_2 = 2$$

ดังนั้น พื้นที่ที่อยู่ภายใต้เส้นแบ่งนี้ จะมีค่าเอาต์พุตเป็น 0 โดย

$$x_1 + x_2 - 2 < 0$$

และพื้นที่ที่อยู่เหนือเส้นนี้ จะมีค่าเอาต์พุตเป็น 1 โดย

$$x_1 + x_2 - 2 \geq 0$$

ในความจริงที่ว่าเพอร์เซปตรอนนั้นสามารถเรียนรู้ได้เพียงฟังก์ชั่น linear separable นั้นถือเป็นข่าวที่ค่อนข้างเลวร้าย เพราะมีฟังก์ชั่นจำนวนมากที่ไม่ได้เป็นแบบนั้น

เพอร์เซปตรอนหนึ่งเลเยอร์นั้นจะช่วยในการตัดสินใจในทิศทางเดียวเท่านั้น โดยไม่คำนึง Activation function ที่ถูกใช้งานโดยเพอร์เซปตรอน ซึ่งหมายความว่าเพอร์เซปตรอนหนึ่งเลเยอร์นั้นสามารถจำแนกได้เฉพาะรูปแบบที่เป็น linear separable เท่านั้น

ข้อจำกัดของการคำนวณของเพอร์เซปครอนที่ได้รับการวิเคราะห์เชิงคณิตศาสตร์ในหนังสือที่มีชื่อ Perceptrons ซึ่งประพันธ์โดย Minsky และ Papert พวกเขาพิสูจน์ให้เห็นว่าเพอร์เซปตรอนของ Rosenblatt นั้นไม่สามารถครอบคลุมทุกอย่างได้บนพื้นฐานของตัวอย่างที่ได้ศึกษาจากพื้นที่ส่วนหนึ่ง นอกจากนี้พวกเขาได้ให้ข้อสรุปว่าข้อจำกัดของเพอร์เซปตรอนหนึ่งเลเยอร์นั้นยังคงสามารถนำมาประยุกต์ใช้สำหรับโครงข่ายประสาทแบบมัลติเลเยอร์ได้ ข้อสรุปนี้ยังไม่มีความแน่นอนที่จะส่งเสริมให้มีการวิจัยเพิ่มเติมเกี่ยวกับโครงข่ายประสาทเทียม

2.3.3.3 เราจะรับมือกับปัญหาอย่างไร เมื่อเจอกับฟังก์ชั่นที่ไม่ใช่ linearly separable

เพื่อรับมือกับปัญหาดังกล่าวเราจำเป็นต้องใช้โครงข่ายประสาทเทียมแบบมัลติเลเยอร์ จากประวัติศาสตร์ที่ผ่านมาได้พิสูจน์ให้เห็นว่าข้อจำกัดของเพอร์เซปตรอนของ Rosenblatt นั้นสามารถนำมาใช้ได้ในรูปแบบโครงข่ายประสาทเทียมขั้นสูง เช่น มัลติเลเยอร์เพอร์เซปตรอนที่ทำการฝึกโดยใช้อัลกอริทึมส่งผ่านย้อนกลับ (back propagation)

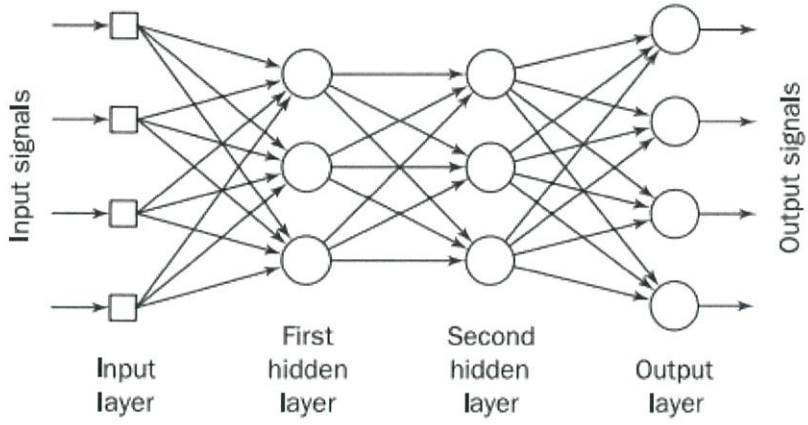
2.3.4 Multilayer neural networks

มัลติเลเยอร์เพอร์เซปตรอนเป็นโครงข่ายประสาทแบบป้อนไปข้างหน้า (feedforward) โดยอาจมีฮิดเดนเลเยอร์ (hidden layer) เพียงหนึ่งเลเยอร์หรือมากกว่านั้น โดยปกติโครงข่ายจะประกอบไปด้วยอินพุตเลเยอร์สำหรับนิวรอนแหล่งกำเนิด ฮิดเดนเลเยอร์หรือมิดเดิลเลเยอร์อย่างน้อยหนึ่งเลเยอร์สำหรับนิวรอนประมวลผล และเอาต์พุตเลเยอร์สำหรับนิวรอนประมวลผลอีกเช่นกัน สัญญาณอินพุตจะถูกแพร่เข้ามาในรูปแบบทิศทางไปข้างหน้าจากเลเยอร์สู่อีกเลเยอร์ รูปที่ 2.9 เป็นรูปที่แสดงถึงมัลติเลเยอร์เพอร์เซปตรอนที่มีฮิดเดนเลเยอร์สองเลเยอร์

2.3.4.1 ทำไมเราจึงจำเป็นต้องมีฮิดเดนเลเยอร์

แต่ละเลเยอร์ในโครงข่ายประสาทจะมีหน้าที่หลักเป็นของตัวเอง อินพุตเลเยอร์จะทำการรับสัญญาณอินพุตจากภายนอกและแพร่กระจายสัญญาณเหล่านั้นไปยังนิวรอนทั้งหมดในฮิดเดนเลเยอร์ ซึ่งหน้าที่ของอินพุตเลเยอร์นั้นจะไม่รวมถึงการประมวลผล ส่วนเอาต์พุตเลเยอร์นั้นจะทำการรับสัญญาณเอาต์พุต หรือในอีกความหมายคือรับสัญญาณจากฮิดเดนเลเยอร์ที่ประมวลผลเสร็จแล้ว จากนั้นเอาต์พุตเลเยอร์จะทำการสร้างรูปแบบของเอาต์พุตเพื่อแสดงผลของโครงข่ายทั้งหมด

นิวรอนในฮิดเดนเลเยอร์จะมีคุณสมบัติคือ น้ำหนักของนิวรอนนั้นจะถูกนำมาในลักษณะที่ซ่อนตัวมากับรูปแบบของอินพุต ซึ่งคุณสมบัตินี้ได้ถูกนำมาใช้โดยเอาต์พุตเลเยอร์เพื่อใช้ในการตัดสินใจในการกำหนดรูปแบบของการแสดงผลเอาต์พุต



รูปที่ 2.9 Multilayer perceptron with two hidden layers

ฮิดเดนเลเยอร์นั้นซ่อนเอาต์พุตที่ต้องการไว้ นิวรอนในฮิดเดนเลเยอร์ไม่สามารถถูกสังเกตผ่านพฤติกรรมของอินพุตและเอาต์พุตของโครงข่าย ไม่มีทางที่จะล่วงรู้ได้ว่าค่าเอาต์พุตที่ได้ของฮิดเดนเลเยอร์นั้นคืออะไร ในอีกความหมายหนึ่งคือ ค่าเอาต์พุตของฮิดเดนเลเยอร์นั้นจะถูกประมวลผลและตัดสินใจโดยเลเยอร์ตัวเอง

โครงข่ายประสาทเทียมในเชิงพาณิชย์นั้น จะใช้เพียงสามเลเยอร์และใช้สี่เป็นบางครั้ง โดยประกอบด้วยฮิดเดนเลเยอร์หนึ่งหรือสองเลเยอร์ แต่ละเลเยอร์สามารถบรรจุนิวรอนได้นับสิบนับพันนิวรอน โครงข่ายประสาทในการทดลองนั้นอาจใช้ถึงห้าหรือหกเลเยอร์ โดยประกอบด้วยฮิดเดนเลเยอร์สามหรือสี่เลเยอร์ และสามารถใช้ประโยชร์จากนิวรอนนับล้านในการประมวล แต่ในทางปฏิบัติจริงส่วนใหญ่จะใช้เพียงสามเลเยอร์เท่านั้น เนื่องจากแต่ละเลเยอร์ที่เพิ่มเข้ามาจะต้องเพิ่มภาระในการประมวลผลอย่างมหาศาล

2.3.4.2 โครงข่ายประสาทแบบมัลติเลเยอร์มีกระบวนการเรียนรู้อย่างไร

มีอัลกอริทึมในการเรียนรู้ที่แตกต่างกันกว่าร้อยวิธีที่สามารถนำมาใช้ได้ แต่วิธีที่นิยมมากที่สุดคืออัลกอริทึมในการแพร่ย้อนกลับ วิธีนี้ถูกนำเสนอครั้งแรกในปี 1969 โดย Bryson และ Ho แต่ว่าวิธีนี้นั้นถูกปฏิเสธการใช้งานไปเรื่องจากมีภาระในส่วนของการคำนวณจำนวนมาก จนกระทั่งในช่วงกลางปี 1980 อัลกอริทึมการเรียนรู้นี้จึงถูกนำกลับมาใช้อีกครั้งหนึ่ง

การเรียนรู้ของโครงข่ายแบบหลายเลเยอร์นั้นมีวิธีการเช่นเดียวกับเพอร์เซปตรอน เซตของอินพุตที่จะใช้ในการฝึกจะถูกส่งเข้าไปยังโครงข่ายเพื่อทำการคำนวณเอาต์พุตออกมา และถ้ามันมีค่าความผิดพลาดเกิดขึ้น หรือในอีกความหมายคือเกิดความแตกต่างระหว่างค่าเอาต์พุตในอุดมคติกับเอาต์พุตที่เกิดขึ้นจริง ค่าน้ำหนักจะถูกเปลี่ยนแปลงเพื่อลดค่าความผิดพลาดนี้

ในเพอร์เซปตรอนนั้น จะมีเพียงหนึ่งน้ำหนักต่อหนึ่งอินพุตและหนึ่งเอาต์พุตเท่านั้น แต่สำหรับมัลติเลเยอร์นั้นจะแตกต่างกัน คือมีหลายน้ำหนัก ซึ่งแต่ละน้ำหนักนี้จะส่งผลกระทบให้เกิดมากกว่าหนึ่งเอาต์พุต

2.3.4.3 เราสามารถประเมินค่าความผิดพลาดและนำไปจัดการกับค่าน้ำหนักได้อย่างไร

โครงข่ายประสาทที่ใช้อัลกอริทึมแบบการแพร่ย้อนกลับนี้ มีอัลกอริทึมในการเรียนรู้แบ่งออกเป็น 2 ช่วงคือ ช่วงแรกจะเป็นการป้อนอินพุตสำหรับฝึกเข้าไปยังอินพุตเลเยอร์ของโครงข่าย โดยอินพุตนั้นจะถูกส่งกระจายต่อไปจากเลเยอร์สู่เลเยอร์ถัดไปจนกระทั่งออกมาเป็นค่าเอาต์พุตที่ถูกสร้างโดยเอาต์พุตเลเยอร์ หากค่าที่ได้นั้นแตกต่างจากค่าเอาต์พุตในอุดมคติ ค่าความผิดพลาดจะถูกคำนวณออกมา และส่งย้อนกลับจากเอาต์พุตเลเยอร์สู่อินพุตเลเยอร์ ค่าน้ำหนักจะถูกแก้ไขเมื่อความผิดพลาดนี้ถูกส่งกลับมา

เช่นเดียวกับโครงข่ายประสาทอื่นๆ การแพร่ย้อนกลับเป็นวิธีการหนึ่งที่จะถูกตัดสินใจโดยการเชื่อมต่อระหว่างนิวรอน และ Activation function ซึ่งถูกใช้โดยนิวรอน และอัลกอริทึมในการเรียนรู้หรือกฎในการเรียนรู้ ขั้นตอนเฉพาะเหล่านี้เป็นสิ่งที่ใช้เพื่อให้มีการปรับปรุงน้ำหนัก

โดยทั่วไปโครงข่ายที่ใช้การแพร่ย้อนกลับจะเป็นโครงข่ายมัลติเลเยอร์ที่มีสามหรือสี่เลเยอร์ ซึ่งมีการเชื่อมต่อกันอย่างสมบูรณ์ นั่นคือทุกนิวรอนในแต่ละเลเยอร์มีการเชื่อมต่อกันกับทุกนิวรอนอื่นๆ ที่อยู่ในเลเยอร์ที่ติดกันถัดไป

นิวรอนจะกำหนดค่าเอาต์พุตด้วยวิธีที่เหมือนกันกับเพอร์เซปตรอนของ Rosenblatt ประการแรกคือการคำนวณเพื่อหาค่าน้ำหนักอินพุตสุทธิก่อน

$$X = \sum_{i=1}^{n} x_i w_i - \theta$$

เมื่อ n คือจำนวนของอินพุต และ θ คือค่า threshold

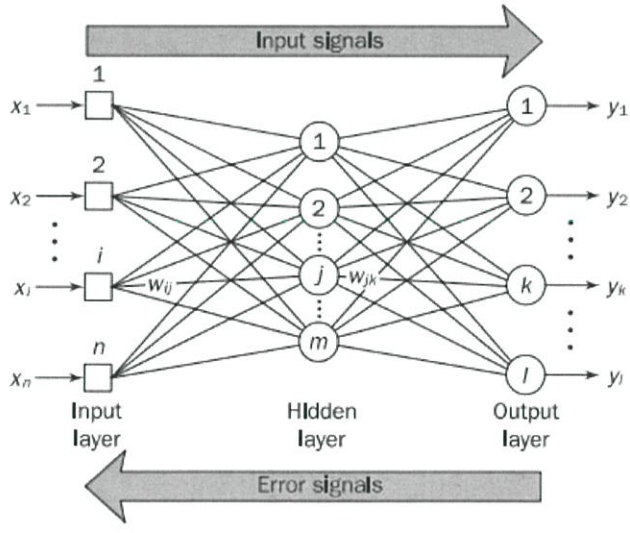
ต่อไป ค่าอินพุตนี้จะถูกส่งเข้าไปทำงานยัง Activation function อย่างไรก็ตาม มันไม่เหมือนกับเพอร์เซปตรอน นิวรอนในโครงข่ายแพร่ย้อนกลับจะใช้ sigmoid activation function แทน

$$Y^{sigmoid} = \frac{1}{1+e^{-X}} \tag{2.9}$$

ผลที่ได้จากฟังก์ชั่นนี้ง่ายต่อการคำนวณ และสามารถรับประกันได้ว่าค่าเอาต์พุตที่ได้จากนิวรอน จะมีค่าอยู่ในขอบเขตระหว่าง 0 และ 1 เสมอ

เพื่อการสร้างอัลกอริทึมในการเรียนรู้แบบแพร่ย้อนกลับนั้น เราได้พิจารณาจากโครงข่ายสามเลเยอร์ ซึ่งแสดงดังรูปที่ 2.10 โดยกำหนดให้ i, j และ k ในที่นี้คือนิวรอนในอินพุตเลเยอร์ ฮิดเดนเลเยอร์ และเอาต์พุตเลเยอร์ตามลำดับ

สัญญาณอินพุต $x_1, x_2, \ldots, x_n$ จะถูกแพร่ผ่านโครงข่ายจากซ้ายไปขวา และสัญญาณความผิดพลาด $e_1, e_2, \ldots, e_i$ จากขวาไปซ้าย สัญลักษณ์ w_{ij} หมายถึงค่าน้ำหนักสำหรับการเชื่อมต่อระหว่างนิวรอน i ในอินพุตเลเยอร์และ j ในฮิดเดนเลเยอร์ และสัญลักษณ์ w_{jk} หมายถึงค่าน้ำหนักระหว่างนิวรอน j ในฮิดเดนเลเยอร์และนิวรอน k ในเอาต์พุตเลเยอร์



รูปที่ 2.10 Three-layer back-propagation neural network

ในการแพร่สัญญาณความผิดพลาดนั้น เราจะเริ่มจากเอาต์พุตเลเยอร์และทำงานย้อนกลับมายังฮิดเดนเลเยอร์ สัญญาณความผิดพลาดที่เอาต์พุตของนิวรอน k ในกระบวนการทำงานรอบที่ p จะถูกกำหนดโดย

$$e_k(p) = y_{d,k}(p) - y_k(p) \quad (2.10)$$

เมื่อ $y_{\mathrm{d,k}}(p)$ คือค่าเอาต์พุตที่คำนวณได้จากนิวรอน k ในกระบวนการทำงานรอบที่ p

นิวรอน k ที่อยู่ในเอาต์พุตเลเยอร์ จะทำการประมวลผลเพื่อหาเอาต์พุตของตัวเอง หลังจากนั้น เราอาจใช้วิธีที่ตรงไปตรงมาเมื่อทำการอัพเดทค่า w_{jk} ซึ่งความจริงแล้วกฎสำหรับการอัพเดทค่าน้ำหนักที่เอาต์พุตเลเยอร์นี้ คล้ายคลึงกับกฎการเรียนรู้ของเพอร์เซปตรอน

$$w_{\mathrm{jk}}(p+1) = w_{\mathrm{jk}}(p) + \Delta w_{\mathrm{jk}}(p) \quad (2.11)$$

เมื่อ $\Delta w_{\mathrm{jk}}(p)$ คือค่าแก้ไขน้ำหนัก

เมื่อเราทำการคำนวณเพื่อหาค่าสำหรับแก้ไขน้ำหนักสำหรับเพอร์เซปตรอน เดิมทีในเพอร์เซปตรอนเราจะใช้สัญญาณอินพุต x_i ในส่วนของการคำนวณ แต่ในโครงข่ายมัลติเลเยอร์นั้น ค่าอินพุตของนิวรอนในเอาต์พุตเลเยอร์จะแตกต่างกับค่าอินพุตของนิวรอนในอินพุตเลเยอร์

เราจะใช้ค่าเอาต์พุตของนิวรอน j ในฮิดเดนเลเยอร์ แทนด้วย y_jแทนที่การใช้อินพุต x_i เพื่อคำนวณหาค่าแก้ไขน้ำหนักของโครงข่ายมัลติเลเยอร์แทน (Fu, 1994)

$$\Delta w_{jk}(p) = \alpha \times y_j(p) \times \delta_k(p) \quad (2.12)$$

เมื่อ $\delta_k(p)$ คือ ค่าเกรเดียนต์ของความผิดพลาดต์ที่นิวรอน k ในเอาต์พุตเลเยอร์ ในรอบการทำงานรอบที่ p

2.3.4.4 ค่าเกรเดียนต์ต์ของความผิดพลาดคืออะไร

ค่าเกรเดียนต์ต์ของความผิดพลาดเป็นค่าที่ได้จากการคูณกัน*ของ* Activation function กับค่าความผิดพลาดที่นิวรอนเอาต์พุต

ดังนั้น ที่นิวรอน k ในเอาต์พุตเลเยอร์ เราจะได้

$$\delta_k(p) = \frac{\partial y_k(p)}{\partial X_k(p)} \times e_k(p) \quad (2.13)$$

เมื่อ $y_k(p)$ คือค่าเอาต์พุตของนิวรอน k ในรอบการทำงานที่ p และ $X_k(p)$ คือค่าผลรวมของน้ำหนักอินพุตที่ไปยังนิวรอน k ในรอบการทำงานเดียวกัน

เมื่อใช้ Sigmoid activation function สมการที่ 1.13 จะเป็นดังนี้

$$\delta_k(p) = \frac{\partial\left\{\frac{1}{1+\exp[-X_k(p)]}\right\}}{\partial X_k(p)} \times e_k(p) = \frac{\exp\left[-X_k(p)\right]}{\{1+\exp[-X_k(p)]\}^2} \times e_k(p)$$

ดังนั้น เราจะได้

$$\delta_k(p) = y_k(p) \times [1 - y_k(p)] \times e_k(p) \quad (2.14)$$

เมื่อ

$$y_k(p) = \frac{1}{1+\exp[-X_k(p)]}$$

2.3.4.5 เราสามารถหาค่าแก้ไขน้ำหนักที่ฮิดเดนเลเยอร์ได้อย่างไร

ในการคำนวณค่าแก้ไขน้ำหนักสำหรับฮิดเดนเลเยอร์นั้น เราจะใช้สมการเดียวกันกับของเอาต์พุตเลเยอร์

$$\Delta w_{ij}(p) = \alpha \times x_i(p) \times \delta_j(p) \quad (2.15)$$

เมื่อ $\delta_j(p)$ แทนค่าเกรเดียนต์ของความผิดพลาดของนิวรอน j ในฮิดเดนเลเยอร์

$$\delta_j(p) = y_j(p) \times [1 - y_j(p)] \times \sum_{k=1}^{l} \delta_k(p) w_{jk}(p)$$

เมื่อ l คือจำนวนของนิวรอนในเอาต์พุตเลเยอร์

$$y_j(p) = \frac{1}{1+e^{-X_j(p)}}$$

$$X_j(p) = \sum_{i=1}^{n} x_i(p) \times w_{ij}(p) - \theta$$

และ n คือจำนวนของนิวรอนในอินพุตเลเยอร์

ในตอนนี้เราสามารถใช้อัลกอริทึมในการฝึกด้วยการแพร่ย้อนกลับได้แล้ว

Step 1: Initialization

กำหนดค่าให้กับน้ำหนักและ Threshold ทั้งหมดในโครงข่ายด้วยการสุ่มค่าตัวเลข ซึ่งมีค่ากระจายอยู่ในช่วงแคบๆ

$$\left(-\frac{2.4}{F_i}, +\frac{2.4}{F_i}\right)$$

เมื่อ F_i คือจำนวนอินพุตทั้งหมดของนิวรอน i ในโครงข่าย การกำหนดค่าเริ่มต้นให้กับน้ำหนักนี้จะถูกใช้กับการเชื่อมต่อระหว่างนิวรอนกับนิวรอนระหว่างเลเยอร์

Step 2: Activation

ดำเนินการใช้อัลกอริทึมแพร่ย้อนกลับในโครงข่ายประสาทโดยป้อนอินพุต $x_1(p), x_2(p), \dots, x_n(p)$ และค่าเอาต์พุตในอุดมคติ $y_{d,1}(p), y_{d,2}(p), \dots, y_{d.n}(p)$

a) คำนวณค่าเอาต์พุตที่ได้จากนิวรอนในฮิดเดนเลเยอร์

$$y_j(p) = sigmoid[\sum_{i=1}^{n} x_i(p) \times w_{ij}(p) - \theta_j]$$

เมื่อ n คือจำนวนของอินพุตในนิวรอน j ในฮิดเดนเลเยอร์ และ $sigmoid$ คือ Sigmoid activation function

b) คำนวณค่าเอาต์พุตที่ได้จากนิวรอนในเอาต์พุตเลเยอร์

$$y_k(p) = sigmoid[\sum_{j=1}^{m} x_{jk}(p) \times w_{jk}(p) - \theta_k]$$

เมื่อ m คือจำนวนของอินพุตของนิวรอน k ในเอาต์พุตเลเยอร์

Step 3: Weight training

อัพเดทค่าน้ำหนักในอัลกอริทึมแพร่ย้อนกลับ โดยเริ่มจากคำนวณค่าความผิดพลาดที่ได้จากเอาต์พุตนิวรอน

a) คำนวณค่าเกรเดียนต์ของความผิดพลาดสำหรับนิวรอนในเอาต์พุตเลเยอร์

$$\delta_k(p) = y_k(p) \times [1 - y_k(p)] \times e_k(p)$$

เมื่อ

$$e_k(p) = y_{d,k}(p) - y_k(p)$$

คำนวณหาค่าแก้ไขน้ำหนัก

$$\Delta w_{jk}(p) = \alpha \times y_j(p) \times \delta_k(p)$$

อัพเดทค่าน้ำหนักที่เอาต์พุตนิวรอน

$$w_{\mathrm{jk}}(p+1) = w_{\mathrm{jk}}(p) + \Delta w_{\mathrm{jk}}(p)$$

b) คำนวณค่าเกรเดียนต์ของความผิดพลาดสำหรับนิวรอนในฮิดเดนเลเยอร์

$$\delta_j(p) = y_j(p) \times [1 - y_j(p)] \times \sum_{k=1}^{l} \delta_k(p) w_{jk}(p)$$

คำนวณหาค่าแก้ไขน้ำหนัก

$$\Delta w_{ij}(p) = \alpha \times x_i(p) \times \delta_j(p)$$

อัพเดทค่าน้ำหนักที่ฮิดเดนนิวรอน

$$\Delta w_{ij}(p) = \alpha \times x_i(p) \times \delta_j(p)$$

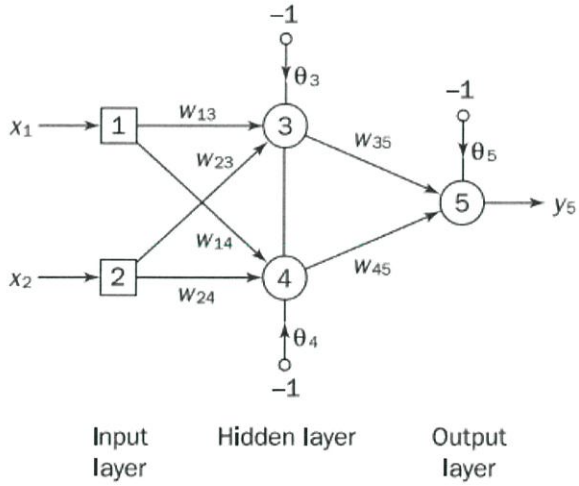
Step 4: Iteration

เข้าสู่การทำงานในรอบถัดไป โดยเพิ่มค่า p อีก 1 แล้วกลับไปยัง Step 2 และทำซ้ำแบบเดิมไปเรื่อยๆ จนกระทั่งครบรอบ หรือได้ผลของความผิดพลาดที่อยู่ในระดับที่น่าพอใจ

จากตัวอย่างเราอาจพิจารณาโครงงข่ายประสาทแบบสามเลเยอร์ ดังรูปที่ 2.11 สมมติว่าหากเราต้องการดำเนินการตรรกะ Exclusive-OR ซึ่งปัญหานี้เพอร์เซปตรอนเลเยอร์เดียวไม่สามารถแก้ไขโอเปอเรชั่นนี้ได้ ในตอนนี้เราจะใช้โครงข่ายสามเลเยอร์แทน

นิวรอน 1 และ 2 ในนอินพุตเลเยอร์จะได้รับอินพุต x_1 และ x_2 ตามลำดับ และแจกจ่ายอินพุตเหล่านี้ไปยังนิวรอนในฮิดเดนเลเยอร์โดยยังไม่ต้องประมวลผลใดๆ

$x_{13} = x_{14} = x_1$ และ $x_{23} = x_{24} = x_2$



รูปที่ 2.11 โครงข่ายสามเลเยอร์สำหรับแก้ปัญหาโอเปอเรชั่น Exclusive-OR

ค่าเทรสโฮลด์ที่ใช้กับนิวรอนในฮิดเดนหรือเอาต์พุตเลเยอร์จะถูกแสดงออกมาในรูปของน้ำหนัก θ เชื่อมต่อกับนิวรอนนั้นๆ และตั้งค่าคงที่ของอินพุตเท่ากับ -1

กำหนดค่าเริ่มต้นให้กับน้ำหนักและค่าเทรสโฮลด์โดยการสุ่มค่า ได้ดังนี้

$w_{13} = 0.5,\ w_{14} = 0.9,\ w_{23} = 0.4,\ w_{24} = 1.0,\ w_{35} = -1.2,\ w_{45} = 1.1,$
$\theta_3 = 0.8, \theta_4 = -0.1$ และ $\theta_5 = 0.3$

พิจารณาจากข้อมูลสำหรับการฝึก ที่อินพุต x_1 และ x_2 เท่ากับ 1 และเอาต์พุตในอุดมคติ $y_{d,5}$ เท่ากับ 0 ค่าเอาต์พุตที่ได้จากการคำนวณของนิวรอน 3 และ 4 ในฮิดเดนเลเยอร์จะถูกคำนวณได้ดังนี้

$$y_3 = sigmoid(x_1 w_{13} + x_2 w_{23} - \theta_3) = 1/\left[1 + e^{-(1\times0.5+1\times0.4-1\times0.8}\right] = 0.5250$$

$$y_4 = sigmoid(x_1 w_{14} + x_2 w_{24} - \theta_4) = 1/\left[1 + e^{-(1\times0.9+1\times1.0+1\times0.1}\right] = 0.8808$$

ต่อไปค่าเอาต์พุตของนิวรอน 5 ในเอาต์พุตเลเยอร์สามารถคำนวณได้ดังนี้

$$y_5 = sigmoid(x_3 w_{35} + x_4 w_{45} - \theta_5) = 1/\left[1 + e^{-(-0.5250\times1.2+0.8808\times1.1-1\times0.3}\right] = 0.5097$$

ดังนั้นค่าความผิดพลาดที่เกิดขึ้นคือ

$$e = y_{d,5} - y_5 = 0 - 0.509 = -0.5097$$

ในขั้นตอนต่อไปคือการปรับค่าน้ำหนักและค่าเทรสโฮลด์ในโครงข่ายของเรา ซึ่งเราจะทำการแพร่กระจายค่าความผิดพลาดที่คำนวณได้นี้ จากเอาต์พุตเลเยอร์กลับไปยังอินพุตเลเยอร์

เริ่มต้น เราจะทำการคำนวณค่าเกรเดียนต์ของความผิดพลาดสำหรับนิวรอน 5 ในเอาต์พุตเลเยอร์

$$\delta_5 = y_5(1 - y_5)e = 0.5097 \times (1 - 0.5097) \times (-0.5097) = -0.1274$$

จากนั้น ทำการคำนวณเพื่อหาค่าแก้ไขน้ำหนัก โดยสมมติว่าค่าอัตราการเรียนรู้ α มีค่าเท่ากับ 0.1

$$\Delta w_{35} = \alpha \times y_3 \times \delta_5 = 0.1 \times 0.5250 \times (-0.1274) = -0.0067$$

$$\Delta w_{45} = \alpha \times y_4 \times \delta_5 = 0.1 \times 0.8808 \times (-0.1274) = -0.0112$$

$$\Delta \theta_5 = \alpha \times (-1) \times \delta_5 = 0.1 \times (-1) \times (-0.1274) = 0.0127$$

ต่อไปเราจะเริ่มทำการคำนวณค่าเกรเดียนต์ของความผิดพลาดสำหรับนิวรอน 3 และ 4 ในฮิดเดนเลเยอร์

$$\delta_3 = y_3(1 - y_3) \times \delta_5 \times w_{35} = 0.5250 \times (1 - 0.5250) \times (-0.1274) \times (-1.2) = -0.1274$$

$$\delta_4 = y_4(1-y_4) \times \delta_5 \times w_{45} = 0.8808 \times (1-0.8808) \times (-0.1274) \times 1.1$$
$$= \quad 0.1274$$

ทำการคำนวณหาค่าแก้ไขน้ำหนัก

$$\Delta w_{13} = \alpha \times x_1 \times \delta_3 = 0.1 \times 1 \times 0.0381 = 0.0038$$

$$\Delta w_{23} = \alpha \times x_2 \times \delta_3 = 0.1 \times 1 \times 0.0381 = 0.0038$$

$$\Delta \theta_3 = \alpha \times (-1) \times \delta_3 = 0.1 \times (-1) \times 0.0381 = -0.0038$$

$$\Delta w_{14} = \alpha \times x_1 \times \delta_4 = 0.1 \times 1 \times (-0.0147) = -0.0015$$

$$\Delta w_{24} = \alpha \times x_2 \times \delta_4 = 0.1 \times 1 \times (-0.0147) = -0.0015$$

$$\Delta \theta_4 = \alpha \times (-1) \times \delta_4 = 0.1 \times (-1) \times (-0.0147) = 0.0015$$

สุดท้าย ทำการอัพเดทค่าน้ำหนักและเทรสโฮลด์ทั้งหมดในโครงข่าย

$$w_{13} = w_{13} + \Delta w_{13} = 0.5 + 0.0038 = 0.5038$$

$$w_{14} = w_{14} + \Delta w_{14} = 0.9 - 0.0015 = 0.8985$$

$$w_{23} = w_{23} + \Delta w_{23} = 0.4 + 0.0038 = 0.4038$$

$$w_{24} = w_{24} + \Delta w_{24} = 1.0 - 0.0015 = 0.9985$$

$$w_{35} = w_{35} + \Delta w_{35} = -1.2 - 0.0067 = -1.2067$$

$$w_{45} = w_{45} + \Delta w_{45} = 1.1 - 0.0112 = 1.0888$$

$$\theta_3 = \theta_3 + \Delta \theta_3 = 0.8 - 0.0038 = 0.7962$$

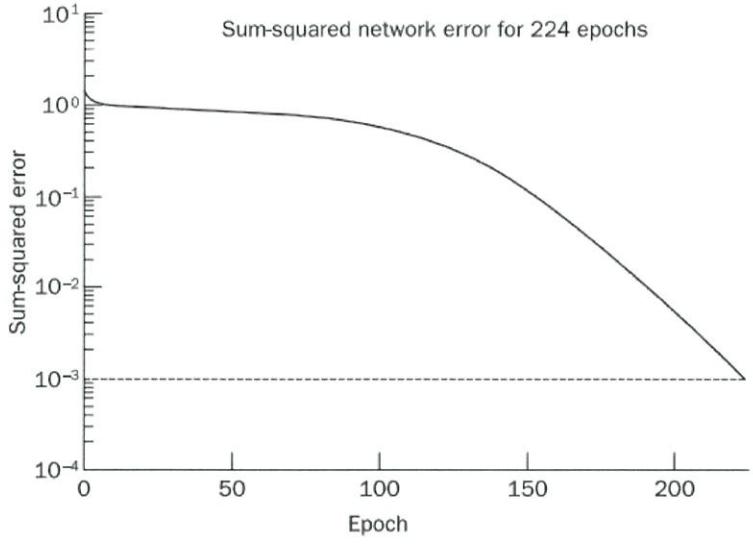
$$\theta_4 = \theta_4 + \Delta \theta_4 = -0.1 + 0.0015 = -0.0985$$

$$\theta_5 = \theta_5 + \Delta \theta_5 = 0.3 + 0.0127 = 0.3127$$

กระบวนการฝึกนี้จะถูกทำซ้ำไปซ้ำมา จนกระทั่งผลรวมกำลังสองของความผิดพลาดมีค่าต่ำกว่า -0.001

2.3.4.6 ทำไมจึงจำเป็นต้องใช้ผลรวมกำลังสองของความผิดพลาด

ค่าผลรวมกำลังสองของความผิดพลาดนั้นถูกใช้เพื่อชี้วัดประสิทธิภาพของโครงข่าย การฝึกโดยใช้อัลกอริทึมแพร่ย้อนกลับนั้นจะทำให้ค่าดังกล่าวนี้ลดลง เมื่อค่าผลรวมกำลังสองของความผิดพลาดครบรอบจำนวนที่เรากำหนดสำหรับการฝึก หรือมีค่าความผิดพลาดน้อยเพียงพอ ดังรูปที่ 2.12 แสดงให้เห็นถึงค่าผลรวมของความผิดพลาดที่มีค่าน้อยกว่า 0.001 และนำมาพลอตกราฟความสัมพันธ์ระหว่างค่าผลรวมกำลังสองของความผิดพลาดกับจำนวนรอบที่ผ่านการฝึก



รูปที่ 2.12 กราฟการเรียนรู้ของโอเปอเรชั่น Exclusive-OR

จะพบว่าทำการฝึกทั้งหมด 224 รอบ หรือรันคำสั่งเพื่อประมวลอินพุต 896 รอบ สำหรับการฝึกโครงข่ายของพวกเราเพื่อให้ค่าความผิดพลาดที่ได้จากการแก้ปัญหาโอเปอเรชั่น Exclusive-OR มีค่าต่ำกว่า 0.001 และผลสุดท้ายของค่าน้ำหนักและเทรสโฮลด์คือ

$w_{13} = 4.7621, w_{14} = 6.3917, w_{23} = 4.7618, w_{24} = 6.3917, w_{35} = -10.3788,$
$w_{45} = 9.7691, \theta_3 = 7.3061, \theta_4 = 2.8441$ และ $\theta_5 = 4.5589$

โครงข่ายนี้สามารถใช้แก้ปัญหาของโอเปอเรชั่นนี้ได้ เราสามารถแสดงผลการทดลองโดยการใช้ข้อมูลรูปแบบของอินพุตและเอาต์พุตของโอเปอเรชั่น Exclusive-OR เพื่อคำนวณค่าเอาต์พุตของโครงข่าย ผลที่ได้ แสดงดังตารางที่ 2.4

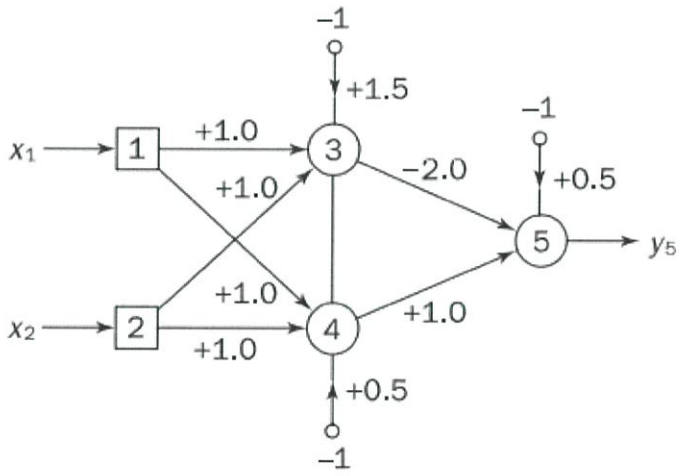
Inputs		Desired Output	Actual Output	Error	Sum of squared errors
x_1	x_2	y_d	y_5	e	
1	1	0	0.0155	-0.0155	0.0010
0	1	1	0.9849	0.0151	
1	0	1	0.9849	0.0151	
0	0	0	0.0175	-0.0175	

ตารางที่ 2.4 ผลลัพธ์สุดท้ายของการฝึกโครงข่ายสามเลเยอร์ ซองตรรกะโอเปอเรชั่น Exclusive-OR

โครงข่ายจะได้รับค่าน้ำหนักและเทรสโฮลด์ที่แตกต่างกันเมื่อเริ่มจากเงื่อนไขของการกำหนดค่าเริ่มต้นที่ต่างกัน แต่สุดท้ายจะสามารถนำไปใช้แก้ไขปัญหาได้เสมอ ถึงแม้ว่าจะใช้จำนวนรอบของการฝึกที่ต่างกันก็ตาม อย่างเช่น เมื่อโครงข่ายถูกนำมาฝึกอีกครั้ง อาจได้รับการแก้ปัญหาเป็นดังนี้

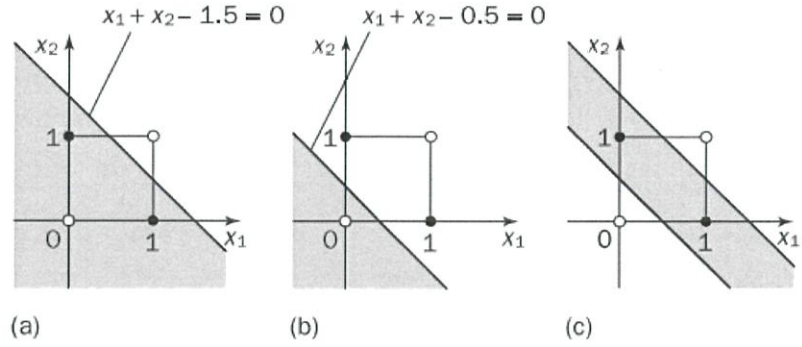
$$w_{13} = -6.3041, w_{14} = -5.7896, w_{23} = 6.2288, w_{24} = 6.0088, w_{35} = 9.6657,$$
$$w_{45} = -9.4242, \theta_3 = 3.3858, \theta_4 = -2.8976 \text{ และ } \theta_5 = -4.4859$$

มันอาจเป็นเรื่องที่ค่อนข้างยากที่จะวาดขอบเขตของการตัดสินใจซึ่งสร้างโดยนิวรอนกับ Sigmoid activation function อย่างไรก็ตาม เราสามารถนำมาแทนที่แต่ละนิวรอนในฮิดเดนและเอาต์พุตเลเยอร์ด้วยโมเดลของ McCulloch และ Pitts ซึ่งใช้ sign function โครงข่ายดังรูปที่ 2.13 จะได้รับการฝึกเพื่อดำเนินการ โอเปอเรชั่น Exclucive-OR



รูปที่ 2.13 การแทนที่โครงข่ายด้วยโมเดลของ McCulloch และ Pitts สำหรับแก้ไขปัญหาโอเปอเรชั่น Exclusive-OR

ตำแหน่งของขอบเขตการตัดสินใจนั้นถูกสร้างขึ้นโดยนิวรอน 3 และ 4 ในฮิดเดนเลเยอร์ซึ่งแสดงดังรูป 2.14(a) และ (b) ตามลำดับ นิวรอน 5 ในเอาต์พุตเลเยอร์จะดำเนินการรวมเส้นขอบเขตของการตัดสินใจทั้งสองเข้าด้วยกัน ซึ่งแสดงดังรูปที่ 2.14(c) โครงข่ายดังรูปที่ 2.13 นั้นจะสามารถแยกจุดสีขาวและดำออกจากกันได้ จึงสามารถแก้ไขปัญหาของโอเปอเรชั่น Exclusive-OR ได้



รูปที่ 2.14 ขอบเขตการตัดสินใจโดยนิวรอนต่างๆ

(a) สร้างโดยนิวรอน 3

(b) สร้างโดย นิวรอน 4

(c) โครงข่ายสามเลเยอร์ที่สมบูรณ์

2.4 สหสัมพันธ์ (Correlation)

ค่าสหสัมพันธ์ (Correlation) เป็นสถิติที่ใช้หาความสัมพันธ์ระหว่างตัวแปร เช่น หาค่าสหสัมพันธ์ระหว่างเจตคติวิชาคณิตศาสตร์ กับผลสัมฤทธิ์ทางการเรียน หาความสัมพันธ์ระหว่างขวัญและกำลังใจในการทำงานกับประสิทธิภาพในการทำงาน เป็นต้น ซึ่งค่าสหสัมพันธ์ที่คำนวณได้เรียกว่า ค่าสัมประสิทธิสหสัมพันธ์ (Correlation coefficient) ซึ่งสถิติสำหรับการคำนวณหาค่าสัมประสิทธิสหสัมพันธ์มีหลายชนิด ซึ่งการเลือกใช้แบบใดนั้นขึ้นอยู่กับเงื่อนไขหลายประการ

2.4.1 การวิเคราะห์สหสัมพันธ์ (Correlation Analysis)

ในการวิเคราะห์ความสัมพันธ์ระหว่างตัวแปรสองตัว (Bivariate Correlation) บางครั้งเราเรียกว่าตัวแปรอิสระว่า ตัวแปรทำนาย (Predictor variable) และเรียกตัวแปรอีกตัวว่าตัวแปรเกณฑ์ (Criterion variable) ซึ่งโดยปกติจะเป็นตัวแปรตาม อย่างไรก็ตามการที่จะทราบว่าตัวแปรทำนายตัวแปรใดเป็นตัวแปรเกณฑ์ ขึ้นอยู่กับงานวิจัยนั้นๆ ในการวิเคราะห์ค่าสหสัมพันธ์ระหว่างตัวแปร ถ้าหากทั้งสองตัวแปรมีระดับการวัดอันตรภาค (Interval scale) หรืออัตราส่วน (Ration scale) จะเรียกว่าการวิเคราะห์โดยใช้พาราเมทริก (Parametric procedure) แต่ถ้ามีระดับการวัดมาตรานามบัญญัติ (Nominal scale) หรือมาตราเรียงอันดับ (Ordinal scale) จะเรียกว่า การวิเคราะห์แบบไม่ใช้พาราเมทริก (Nonparametric procedure)

ในการวัดความสัมพันธ์แต่ละแบบจะต้องมีการทดสอบนัยสำคัญก่อน จึงจะสรุปได้ว่าตัวแปรคู่ใดมีความสัมพันธ์กันจริงหรือไม่ มากน้อยเพียงใด สำหรับการแปลผลจะมองในแง่ของความเกี่ยวพัน ความสอดคล้อง การแปรผันร่วมกัน หรือไปด้วยกัน แต่ไม่ได้หมายความว่าตัวแปรหนึ่งเป็นเหตุและอีก

ตัวแปรเป็นผล (หรือไม่สามารถระบุได้ว่าตัวแปรไหนเป็นตัวแปรต้นหรือตัวแปรตาม) เช่น ศึกษาความสัมพันธ์ระหว่างส่วนสูงกับน้ำหนัก เราไม่สามารถบอกได้ว่าส่วนสูงหรือน้ำหนักตัวใดเป็นเหตุและตัวใดเป็นผล บอกได้เพียงว่ามีความสัมพันธ์กันหรือไม่ และมีขนาดของความสัมพันธ์กันมากน้อยเพียงใด

ค่าสัมประสิทธิ์สหสัมพันธ์ โดยทั่วไปนิยมใช้สัญลักษณ์ r แทนสัมประสิทธิ์สหสัมพันธ์ของกลุ่มตัวอย่าง (บางชนิดจะใช้สัญลักษณ์ C, W หรืออื่นๆ) และ ρ แทนสัมประสิทธิ์สหสัมพันธ์ของประชากร ค่าสัมประสิทธิ์สหสัมพันธ์ ที่ใช้วัดขนาดของความสัมพันธ์กันระหว่างตัวแปร มี 2 ลักษณะ คือ $-1 \leq r \leq 1$ และ $0 \leq r \leq 1$

การบอกระดับหรือขนาดของความสัมพันธ์ จะใช้ตัวเลขของค่าสัมประสิทธิ์สหสัมพันธ์ หากค่าสัมประสิทธิ์สหสัมพันธ์มีค่าเข้าใกล้ -1 หรือ 1 แสดงถึงการมีความสัมพันธ์กันในระดับสูง แต่หากมีค่าเข้าใกล้ 0 แสดงถึงการมีความสัมพันธ์กันในระดับน้อย หรือไม่มีเลย สำหรับการพิจารณาค่าสัมประสิทธ์สหสัมพันธ์ โดยทั่วไปอาจใช้เกณฑ์ดังนี้

ค่า r	ระดับของความสัมพันธ์
.90 - 1.00	มีความสัมพันธ์กันสูงมาก
.70 - .90	มีความสัมพันธ์กันในระดับสูง
.50 - .70	มีความสัมพันธ์กันในระดับปานกลาง
.30 - .50	มีความสัมพันธ์กันในระดับต่ำ
.00 - .30	มีความสัมพันธ์กันในระดับต่ำมาก

เครื่องหมาย +, - หน้าตัวเลขสัมประสิทธิ์สหสัมพันธ์ จะบอกถึงทิศทางของความสัมพันธ์ โดยที่หาก

r มีเครื่องหมาย +	หมายถึง การมีความสัมพันธ์กันไปในทิศทางเดียวกัน (ตัวแปรหนึ่งมีค่าสูงขึ้น อีกตัวหนึ่งจะมีค่าสูงไปด้วย)
r มีเครื่องหมาย -	หมายถึง การมีความสัมพันธ์กันไปในทิศทางตรงกันข้าม (ตัวแปรหนึ่งมีค่าสูงขึ้น ตัวแปรอีกตัวหนึ่งจะมีค่าต่ำลง)
r มีค่าเท่ากับ 0	หมายถึงตัวแปรสองตัวไม่มีความสัมพันธ์ซึ่งกันและกัน

ยกเว้นค่าสัมประสิทธิ์สหสัมพันธ์บางชนิดที่มีลักษณะ $0 \leq r \leq 1$ ซึ่งจะบอกได้เพียงขนาดหรือระดับของความสัมพันธ์เท่านั้น ไม่สามารถบอกทิศทางของความสัมพันธ์ได้

ค่าสัมประสิทธิ์สหสัมพันธ์ ในสถิตินอนพาราเมตริก ใช้ในการหาความสัมพันธ์ระหว่างตัวแปรที่มีมาตรวัดได้ตั้งแต่นามบัญญัติขึ้นไป และไม่เจาะจงชนิดของการแจกแจงความน่าจะเป็นของข้อมูล ได้แก่

1) สถิติไคสแควร์สำหรับการทดสอบความเป็นอิสระต่อกัน
2) สัมประสิทธิ์สหสัมพันธ์ฟาย
3) สัมประสิทธิ์สหสัมพันธ์คอนติงเจนซี
4) สัมประสิทธิ์สหสัมพันธ์แบบสเปียร์แมน
5) สัมประสิทธิ์สหสัมพันธ์แบบพอยท์ไบซีเรียล
6) สัมประสิทธิ์สหสัมพันธ์แบบไบซีเรียล
7) สัมประสิทธิ์สหสัมพันธ์เตตราคอห์ริค

ค่าสัมประสิทธิ์สหสัมพันธ์ในสถิติพาราเมตริก เป็นการหาความสัมพันธ์สำหรับตัวแปรที่มีมาตรวัดอันตรภาค หรืออัตราส่วน และมีการแจกแจกความน่าจะเป็นแบบปกติ ซึ่งได้แก่

1) สัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน
2) สัมประสิทธิ์สหสัมพันธ์แบบแยกส่วน

2.4.2 สัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน (Pearson product-moment correlation coefficient)

สัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน แทนด้วยสัญลักษณ์ r_{xy} เป็นวิธีที่ใช้วัดความสัมพันธ์ระหว่างตัวแปร หรือข้อมูล 2 ชุด โดยที่ตัวแปร หรือข้อมูล 2 ชุดนั้นจะต้องอยู่ในรูปของข้อมูลในมาตราอันตรภาคหรืออัตราส่วน (Interval or Ratio scale) เช่น การหาความสัมพันธ์ระหว่างภาวะสุขภาพกับการดูแลตนเอง การหาความสัมพันธ์ระหว่างน้ำหนักแรกเกิดของทารกกับอายุของมารดา เป็นต้น

สัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน จะใช้ได้อย่างเหมาะสมกับข้อมูลที่มีความสัมพันธ์เชิงเส้นเท่านั้น ดังนั้น ในการคำนวณหากพบว่าค่า r=0 การตีความหมายว่าข้อมูลไม่มีความสัมพันธ์กันอาจไม่ถูกต้อง เนื่องจากอาจเป็นไปได้ว่าข้อมูลมีความสัมพันธ์กันในลักษณะอื่นที่ไม่ใช่เชิงเส้น (เช่น เส้นโค้ง ฯลฯ) ดังนั้น ในการสรุปจึงควรสรุปว่าข้อมูล 2 ชุดไม่มีความสัมพันธ์เชิงเส้น จึงจะถูกต้องชัดเจนกว่า

2.4.2.1 ข้อตกลงเบื้องต้น

1) ตัวแปรหรือข้อมูลทั้ง 2 ชุด อยู่ในมาตราอันตรภาค หรือมาตราอัตราส่วน
2) ข้อมูลทั้ง 2 ชุด มีการแจกแจงแบบปกติ และมีความสัมพันธ์เชิงเส้นตรง
3) ข้อมูลในแต่ละชุดจะต้องมีความเป็นอิสระต่อกัน

2.4.2.2 สูตรคำนวณค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน

$$r_{xy} = \frac{N\sum XY - (\sum X)(\sum Y)}{\sqrt{\left[N\sum X^2 - (\sum X)^2\right]\left[N\sum Y^2 - (\sum Y)^2\right]}}$$

เมื่อ r_{xy} เป็น ค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน

$\sum X$ เป็น ผลรวมของข้อมูลที่วัดได้จากตัวแปรตัวที่ 1 (X)

$\sum Y$ เป็น ผลรวมของข้อมูลที่วัดได้จากตัวแปรตัวที่ 2 (Y)

$\sum XY$ เป็น ผลรวมของผลคูณระหว่างข้อมูลตัวแปรที่ 1 และ 2

$\sum Y^2$ เป็น ผลรวมของกำลังสองของข้อมูลที่วัดได้จากตัวแปรตัวที่ 1

$\sum X^2$ เป็น ผลรวมของกำลังสองของข้อมูลที่วัดได้จากตัวแปรตัวที่ 2

N เป็น ขนาดของกลุ่มตัวอย่าง

2.4.2.3 การทดสอบนัยสำคัญ

1) สมมติฐานของการทดสอบ

H0 : $\rho = 0$ (ตัวแปร X และ Y ไม่มีความสัมพันธ์กัน)

H1 : $\rho \neq 0$ (ตัวแปร X และ Y มีความสัมพันธ์กัน)

2) สถิติทดสอบ เป็นการทดสอบแบบสองทาง

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}}$$, df = n-2

3) อาณาเขตวิกฤตและการสรุปผล

จะปฏิเสธ H_0 เมื่อค่า t ที่คำนวณได้มีค่ามากกว่าหรือเท่ากับค่า $t_{\alpha,n-2}$ ที่เปิดจากตาราง

หรือ t ที่คำนวณได้มีค่าน้อยกว่าหรือเท่ากับค่า $-t_{\alpha,n-2}$

หรือนำค่า r_{xy} ที่ได้ไปเทียบกับค่าวิกฤต r_{xy} จากตารางสำเร็จรูปโดยใช้ df = n-2

ถ้าค่า r_{xy} มีค่ามากกว่าหรือเท่ากับ ค่าวิกฤต r_{xy} จะปฏิเสธ H_0

2.5 ภาษาจาวา (Java programming language)

2.5.1 ประวัติของจาวา

ภาษาจาวา (Java programming language) เป็นภาษาโปรแกรมเชิงวัตถุ (Object Oriented Programming) พัฒนาโดย เจมส์ กอสลิง และวิศวกรคนอื่นๆ ที่ ซัน ไมโครซิสเต็มส์ ภาษาจาวาถูกพัฒนาขึ้นในปี พ.ศ. 2534 (ค.ศ. 1991) โดยเป็นส่วนหนึ่งของ โครงการกรีน (the Green Project) และสำเร็จออกสู่สาธารณะในปี พ.ศ. 2538 (ค.ศ. 1995) ซึ่งภาษานี้มีจุดประสงค์เพื่อใช้แทนภาษาซีพลัสพลัส (C++) โดยรูปแบบที่เพิ่มเติมขึ้น คล้ายกับภาษาอ็อบเจกต์ทีฟซี (Objective-C) แต่เดิมภาษานี้เรียกว่า ภาษาโอ๊ก (Oak) ซึ่งตั้งชื่อตามต้นโอ๊กใกล้ที่ทำงานของ เจมส์ กอสลิง แต่ว่ามีปัญหาทางลิขสิทธิ์ จึงเปลี่ยนไปใช้ชื่อ "จาวา" ซึ่งเป็นชื่อกาแฟแทน

ภาษาจาวาถูกพัฒนาขึ้นโดยบริษัท ซันไมโครซิสเต็มส์ (Sun Microsystems Inc.) เป็นภาษาสำหรับเขียนโปรแกรมภาษาหนึ่ง มีลักษณะสนับสนุนการเขียนโปรแกรมเชิงวัตถุ (OOP : Object-Oriented Programming) ที่ชัดเจน โปรแกรมต่างๆ ถูกสร้างภายใน Class โปรแกรมเหล่านั้นถูกเรียกว่า Method หรือ Behavior โดยปกติจะเรียกแต่ละคลาสว่าวัตถุ โดยแต่ละวัตถุมีพฤติกรรมมากมาย โปรแกรมที่สมบูรณ์จะเกิดจากหลายวัตถุ หรือหลายคลาสมารวมกัน โดยแต่ละคลาสจะมี Method หรือพฤติกรรมแตกต่างกันไป

พัฒนาการในช่วงเวลาต่างๆ ของจาวามีดังนี้

ค.ศ.1991 บริษัท ซันไมโครซิสเต็ม (Sun Microsystems) ได้ทำการวิจัยเพื่อพัฒนาซอฟต์แวร์ที่ใช้ควบคุมอุปกรณ์เล็กทรอนิคส์ขนาดเล็ก ซึ่งได้ผลลัพธ์ที่สำคัญคือ ภาษาโอ๊ค (Oak)

ค.ศ.1993 ภาษาโอ๊คได้ถูกปรับปรุงใหม่เพื่อใช้ในการสร้างเว็บแอพพลิเคชั่น (Web Application) พร้อมกับสร้างเว็บบราวเซอร์ที่รองรับ ชื่อว่าเว็บรันเนอร์ (Web Runner)

ค.ศ.1995 บริษัทซันได้เปิดตัวภาษาจาวา (Java) พร้อมกับเว็บบราวเซอร์ ที่รองรับภาษานี้ชื่อว่า ฮอตจาวา (HotJava) และได้รับการสนับสนุนจากบริษัทใหญ่ทั้งเน็ตสเคบ (Netscape), ไมโครซอฟต์ (Microsoft), และ ไอบีเอ็ม (IBM) และบริษัทซัน ได้เริ่มแจกจ่าย Java development Kit (JDK) ซึ่งเป็นชุดพัฒนาโปรแกรมภาษาจาวาในอินเทอร์เน็ต

2.5.2 คุณลักษณะเด่นของภาษาจาวา

1) ภาษาจาวา เป็นภาษาที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุแบบสมบูรณ์
2) โปรแกรมที่เขียนขึ้นโดยใช้ภาษาจาวา จะมีความสามารถทำงานได้ในระบบปฏิบัติการที่แตกต่างกัน ไม่จำเป็นต้องดัดแปลงแก้ไขโปรแกรม เช่น หากเขียนโปรแกรมบนเครื่อง Sun โปรแกรมนั้นก็สามารถถูก compile และ run บนเครื่องพีซีธรรมดาได้

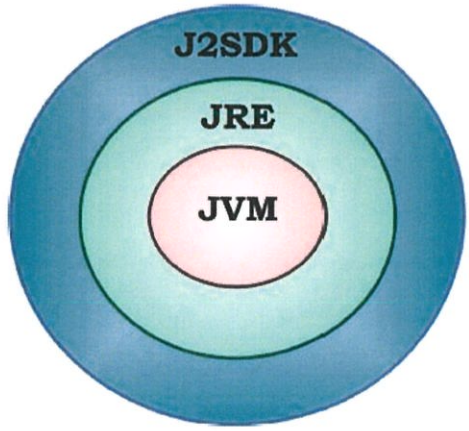
3) เมื่อเปรียบเทียบ code ของโปรแกรมที่เขียนขึ้นโดยภาษา Java กับ C++ พบว่า โปรแกรมที่เขียนโดยภาษาจาวา จะมีจำนวน code น้อยกว่าโปรแกรมที่เขียนโดยภาษา C++ ถึง 4 เท่า และใช้เวลาในการเขียนโปรแกรม น้อยกว่าประมาณ 2 เท่า
4) จาวามี security ทั้ง low level และ high level ได้แก่ electronic signature, public and private key management, access control และ certificatesของภาษาจาวา

2.5.3 จุดมุ่งหมายของการพัฒนาภาษาจาวา

1) ใช้ภาษาโปรแกรมเชิงวัตถุ
2) ไม่ขึ้นกับแพลตฟอร์ม (สถาปัตยกรรม และ ระบบปฏิบัติการ)
3) เหมาะกับการใช้ในระบบเครือข่าย พร้อมมีไลบรารีสนับสนุน
4) เรียกใช้งานจากระยะไกลได้อย่างปลอดภัย

2.5.4 องค์ประกอบของภาษาจาวา

ภาษาจาวามีส่วนประกอบหลัก 3 ส่วน ดังรูปที่ 2.15 ซึ่งแต่ละส่วนมีความสำคัญคือ



รูปที่ 2.15 องค์ประกอบภาษาจาวา

2.5.4.1 Java virtual machine (JVM)

คือ Software program ซึ่งเป็นส่วนประกอบหนึ่งอยู่ใน JRE ซึ่งมีหน้าที่จำลองคอมพิวเตอร์เสมือน (Virtual machine) ขึ้นบนเครื่องคอมพิวเตอร์จริงๆ ของเรา คอมพิวเตอร์เสมือนนี้จะทำหน้าที่ในการแปลไบต์โค้ด (Bytecode) ไปเป็นภาษาเครื่องที่เหมาะสมกับแพลตฟอร์มจริง ของแต่ละเครื่อง หลักการนี้ทำให้ java เป็นภาษาที่สามารถทำงานได้ทุกระบบปฏิบัติการดังแนวคิดเริ่มต้น “Write Once Run Anywhere”

2.5.4.2 Java runtime environment (JRE)

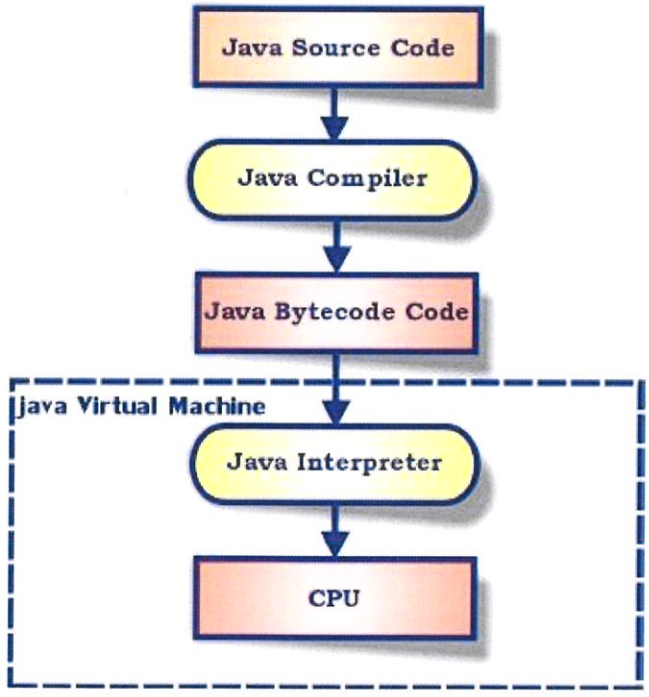
เป็นเทคโนโลยีจาวาที่ใช้ในการรันโปรแกรมภาษาจาวา ที่จะรวบรวม Class และ Interface ต่างๆ ที่จำเป็นต่อการใช้งานของโปรแกรมภาษาจาวา

2.5.4.3 Java 2 software developer kit (J2SDK)

เป็นชุดพัฒนาโปรแกรมประกอบด้วยคอมไพเลอร์ (Compiler) และดีบักเกอร์ (Debugger)

2.5.5 ขั้นตอนการทำงานของโปรแกรมภาษาจาวา

การทำงานของภาษาจาวาเริ่มจากผู้พัฒนาเริ่มเขียนโปรแกรมภาษาจาวา (Java source code *.java) เมื่อเขียนโปรแกรมเรียบร้อยแล้ว จะนำโปรแกรมภาษาจาวา ไปทำการคอมไพล์โดยจาวาคอมไพล์เลอร์ (Java compiler) ผลลัพธ์ออกมาจะเป็นไบต์โค้ด (Java byte code *.class) ซึ่งมีลักษณะเป็นโค้ดที่จะกำหนดรูปแบบการติดต่อกับ จาวาเวอร์ชัวแมชชีน (Java virtual machine) ซึ่งจุดนี้เอง ทำให้จาวาสามารถรันได้ทุกแพลตฟอร์มที่ติดตั้งจาวาเวอร์ชัวแมชชีน เมื่อไบต์โค้ดถูกส่งไปให้เวอร์ชัวแมชชีน จะผ่านเข้าไปที่แปลภาษา (Java interpreter) ซึ่งจะแปลจากไบต์โค้ดให้เป็นภาษาเครื่อง (Assembly) และส่งให้ซีพียูประมวลผลต่อไป ดังรูปที่ 2.16



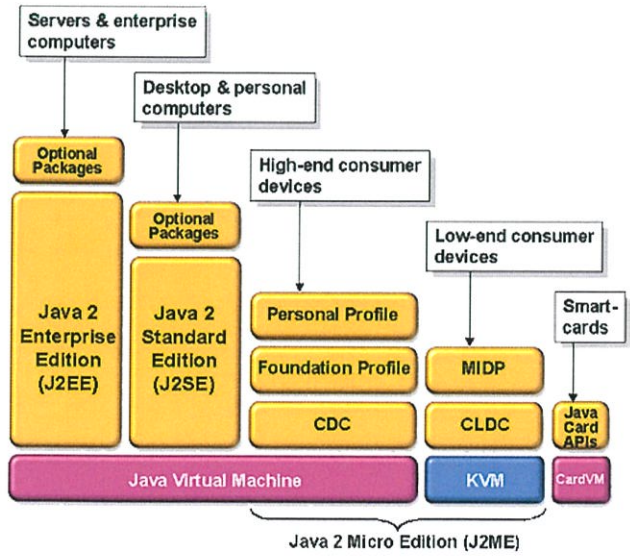
รูปที่ 2.16 ขั้นตอนการทำงานของภาษาจาวา

การทำงานของ Java Interpreter จะใช้หลักการทำงานของตัวแปลภาษา (Interoperation) ซึ่งการทำงานของตัวแปลภาษาตัวนี้ คือ จะอ่านคำสั่งจาก Java Byte Code ทีละบรรทัด แล้วทำการแปลคำสั่งชุดนั้นให้เป็น Executable Code แล้ส่งไปให้ CPU ทำการประมวลผล Executable Code ก็คือ ภาษาเครื่องที่ CPU เข้าใจนั่นเอง เมื่อทำงานคำสั่งชุดนั้นเสร็จก็จะกลับมาอ่านคำสั่งในบรรทัดต่อไป แล้วไปแปลงคำสั่งชุดนั้นให้เป็น Executable Code อีก แล้วก็ส่งไปให้ CPU ประมวลผล และจะทำเช่นนี้ไปเรื่อยๆ จนกว่าจะจบโปรแกรม การทำงานแบบนี้ดูแล้วรู้สึกว่าการทำงานนั้นช้าอยู่ แต่มีข้อดีคือ จะทำให้โปรแกรมมีความยืดหยุ่นในการพัฒนาโปรแกรม เนื่องจากในขณะที่กำลังแปลงอยู่นั้น แล้วเกิดข้อผิดพลาดหรือ Error ที่ไม่ใหญ่หลวงขึ้นมา เครื่องจะอนุญาตให้โปรแกรมแก้ไขโปรแกรมได้ทันที แล้วจึงทำงานในบรรทัดนั้นต่อไปได้

2.5.6 Java Platform

Java Platform ก็คือ Platform หรือสภาพแวดล้อมที่ใช้ในการรันโปรแกรมจาวา โปรแกรมจาวาจะทำงานบน Java Platform เท่านั้น Java platform จะประกอบไปด้วยสองอย่าง คือ Java virtual machine (JVM) และ Runtime library โปรแกรมจาวาที่เราเขียนขึ้นจะทำงานบน Platform ใดก็ได้ที่มี Java Platform ทำงานอยู่ ซึ่งบริษัทซันไมโครซิสเต็มส์ได้กำหนด Platform ของ Java 2 เอาไว้ 3 รูปแบบดังรูปที่ 2.17 ได้แก่

1) Java 2 PlatForm, Standard Edition (J2SE)
2) Java 2 PlatForm, Enterpriise Edition (J2EE)
3) Java 2 PlatForm, Micro Edition (J2ME)

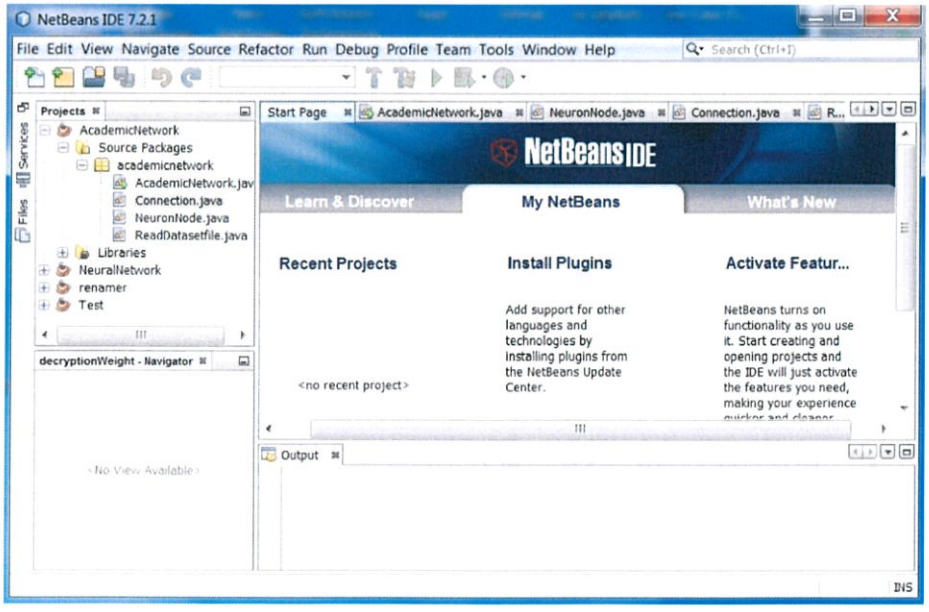


รูปที่ 2.17 Java Platform

2.5.7 เครื่องมือที่ใช้พัฒนาโปรแกรมภาษาจาวาสำหรับซอฟแวร์โครงงานนี้

เน็ตบีนส์ (NetBeans) เป็นเครื่องมือสำหรับนักโปรแกรมเมอร์ที่จะใช้พัฒนาแอพพลิเคชั่นด้วยภาษาจาวา ในปี ค.ศ. 1998 ได้มีกลุ่มนักศึกษา "rock solid software" ได้พัฒนาซอฟต์แวร์ขึ้นมาตัวหนึ่ง ที่จะใช้ในการพัฒนา แอพพลิเคชั่นด้วยภาษาจาวา เป็นโปรเจกต์นักศึกษา โดยตั้งชื่อว่า เน็ตบีนส์ และได้เผยแพร่ให้โปรแกรมเมอร์และบุคคลทั่วไปนำไปใช้งานได้ฟรีในรูปแบบ Opensource software ต่อมาในปี ค.ศ. 2000 บริษัทซัน ไมโครซิสเต็มส์ ผู้พัฒนาภาษาจาวา ได้เข้ามาเป็นผู้สนับสนุนหลักในการพัฒนาเน็ตบีนส์ และได้ทำออกมาในรูปของ Opensource software โดยผู้ใช้งานไม่จำเป็นที่จะต้องเสียเงิน เพื่อซื้อมาใช้งาน และยังได้เปิดเผย Soure code ให้ผู้สนใจและนักพัฒนานำไปดัดแปลง แก้ไข ตามกฎของ Opensource ปัจจุบันมีนักโปรแกรมเมอร์ทั่วโลกต่างช่วยกันพัฒนาเน็ตบีนส์ ให้มีความสามารถสูงยิ่งขึ้น

ปัจจุบัน NetBeans IDE ได้รับความนิยมมากยิ่งขึ้น และได้รับการพัฒนาให้มีความสามารถสูงยิ่งขึ้นเรื่อยๆ นอกจากจะใช้ในการพัฒนา Application ด้วยภาษาจาวาแล้ว ยังสามารถพัฒนาอื่นๆ ได้อีกหลากหลายโดยติดตั้งโปรแกรมเสริม(Add-on)ได้จากเว็บไซต์ หรือผ่านตัวอัปเดตเซนเตอร์ (Update Center) ของเน็ตบีนส์ เช่น ภาษาซี/ซีพลัสพลัส (C/C++), Ruby, UML, SOA, Web Application, Java EE, Mobility(Java ME), Java FX, Java Script, PHP เป็นต้น ในเวอร์ชัน 6.0 เป็นต้นไปมีการรวมโปรแกรมเสริมต่างๆที่สำคัญเข้าในตัวติดตั้งของเน็ตบีนส์ โดยสามารถเลือกติดตั้งได้ภายหลัง ได้จากเว็บไซต์ http://www.netbeans.org ซึ่งรูปที่ 2.18 แสดงให้User Interface ของเน็ตบีนส์



รูปที่ 2.18 หน้าต่างการทำงานของเน็ตบีนส์

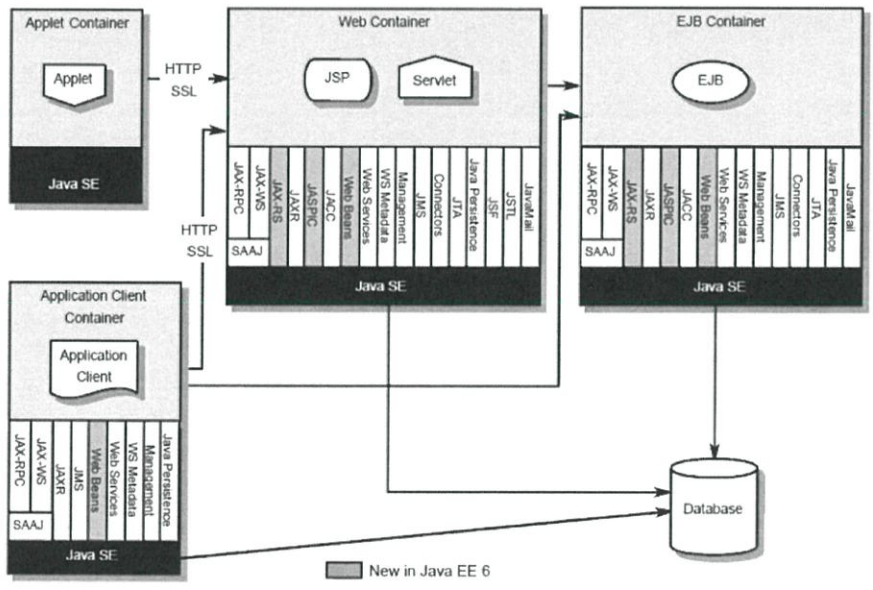
2.5.7.1 จุดเด่นของเน็ตบีนส์

1) เป็นชุดพัฒนาโปรแกรมที่ผู้พัฒนาภาษาจาวามีส่วนสนับสนุน จึงมีโมดูลสำเร็จทุกอย่างในตัวเน็ตบีนส์เอง จึงแทบไม่จำเป็นต้องหา Plugin มาลงเพิ่มอีก
2) ส่วนติดต่อประสานผู้ใช้แบ่งได้เป็นสัดส่วน เข้าใจง่าย ทำให้โปรแกรมเมอร์สามารถสร้างสรรค์งานได้ตรงกับความต้องการได้มากกว่า และงานมีคุณภาพมากยิ่งขึ้น
3) สามารถติดตั้งได้ทั้ง Windows, Linux, Solaris, SUN Spark และ Mac OS X ซึ่งทำให้สามารถพัฒนาโปรแกรมบน Windows และทดสอบโปรแกรมบน Linux ได้

2.6 เจเอสพีและเซิร์ฟเลต (JSP & Servlet)

2.6.1 เจเอสพีและเซิร์ฟเลตคืออะไร

เจเอสพีและเซิร์ฟเลต เป็นเทคโนโลยีด้านเว็บโปรแกรมมิ่ง ซึ่งเป็นทางเลือกหนึ่งสำหรับการพัฒนาเว็บแอพพลิเคชั่นและเว็บไซน์แบบไดนามิค ซึ่งเหมาะสำหรับผู้พัฒนาที่มีพื้นฐานของภาษาจาวา โดยเจเอสพีและเซิร์ฟเลตเป็นส่วนหนึ่งของเทคโนโลยีจาวา ที่เกี่ยวข้องกับเว็บ รูปที่ 2.19 แสดงส่วนประกอบต่างๆ ของ Java EE (Enterprise Edition) v.6



รูปที่ 2.19 Java Enterprise Edition Version 6

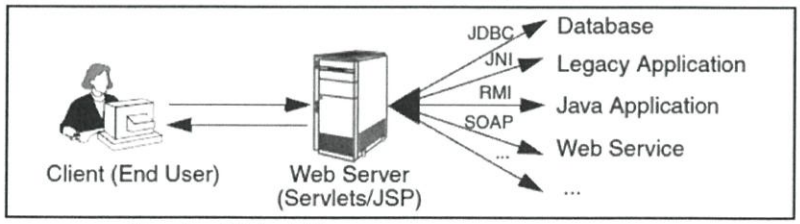
เจเอสพีย่อมาจาก Java Server Pages เป็นเทคโนโลยีที่ทำให้มีการแทรก dynamic content (เนื้อหาที่เปลี่ยนแปลงได้) ลงหน้าเว็บได้ โดยการแทรกส่วนของ tag โปรแกรมพิเศษลงในไฟล์ HTML มาตรฐานทั่วไป

เซิร์ฟเลตเป็นโปรแกรมจาวาที่รันบนเว็บเซิร์ฟเวอร์ ทำหน้าที่สร้างหน้าเว็บ ส่งไปยังผู้ใช้งาน (client) ตามคำร้องขอ (request) ซึ่งหน้าเว็บที่ถูกส่งไปจะมีลักษณะเป็นแบบไดนามิค (เนื้อหาไม่คงที่) ขึ้นอยู่กับข้อมูลที่ได้มีการประมวลผลแล้ว โปรแกรมเซิร์ฟเลต สามารถทำงานได้ดังนี้

1) อ่านข้อมูลที่ส่งมาจากผู้ใช้งานอาจจะผ่านแบบฟอร์มบนหน้าเว็บหรือโปรแกรมทางฝั่งผู้ใช้งาน เช่น Applet
2) อ่านข้อมูลแฝงจากการร้องขอข้อมูลที่ส่งมาจากบราวเซอร์ของผู้ใช้งาน เช่น ข้อมูลจากคุกกี้ ข้อมูลเกี่ยวกับ บราวเซอร์หรืออื่นๆ
3) สร้างผลลัพธ์ หลังจากมีการร้องขอข้อมูล เซิร์ฟเลตจะทำการประมวลผล โดยอาจจะมีการเรียกใช้ข้อมูลจากฐานข้อมูล หรือโปรแกรมอื่นๆ ของฝั่งเซิร์ฟเวอร์แล้วแปลงผลลัพธ์ที่ได้ให้อยู่ในรูปของหน้าเว็บ HTML
4) เซิร์ฟเลตอาจส่งข้อมูลที่ชัดแจ้ง (explicit data) บางอย่างไปยังผู้ใช้งานได้ เช่น ไฟล์เอกสาร รูปภาพ หรือ format อื่นๆ
5) เซิร์ฟเลตส่งข้อมูลแฝง (implicit HTTP response data) ที่ส่งให้กับบราวเซอร์ของผู้ใช้งาน เช่น ข้อมูลประเภทชนิดไฟล์ที่ส่งกลับมา คุกกี้ หรือ พารามิเตอร์อื่นๆ เป็นต้น

ทั้งเอเอสพีและเซิร์ฟเลต ใช้หลักการทำงานแบบ Server-side processing คือฝั่งของเว็บเซิร์ฟเวอร์ เป็นผู้รันโปรแกรมและส่งผลลัพธ์ (หน้าเว็บ) ให้กับผู้ใช้งาน (หรือเว็บบราวเซอร์) ต่างจากหลักงานทำงานแบบ client-side เช่น โปรแกรมประเภท applet ที่รันอยู่บนบราวเซอร์ของฝั่งผู้ใช้งาน เป็นต้น

การทำงานของเจเอสพีและเซิร์ฟเลต เป็นเหมือนตัวกลาง (Middleware) ระหว่างผู้ใช้งานกับข้อมูล หรือ แอพพลิเคชั่นของฝั่งเซิร์ฟเวอร์



The role of Web middleware.

รูปที่ 2.20 ไดอะแกรมแสดงหลักการทำงานของเจเอสพีและเซิร์ฟเลต

เจเอสพีและเซิร์ฟเลตต่างก็สามารถทำงานแยกกันจะใช้งานร่วมกันก็ได้เพื่อสร้างเว็บแอพพลิเคชั่น ที่มีความซับซ้อน ให้ทำงานสมบูรณ์ตามความต้องการได้ เทคโนโลยีเจเอสพี จะมุ่งเน้นทำให้การสร้างและบำรุงรักษา HTML ที่ง่ายดายขึ้น ส่วนด้านของเซิร์ฟเลต เน้นทำงานที่มีความ

ซับซ้อนและการประมวลผล หรือสรุปสั้นๆ คือ เจเอสพีเน้นในส่วนของการพรีเซ้นเป็นหลัก ในขณะที่เซิร์ฟเลตจะเน้นในส่วนของการประมวลผล

2.6.2 ข้อดีของภาษาเจเอสพี

1) ทำงานโดยไม่ยึดติดแพลตฟอร์มใดๆ คือการทำงานโดยไม่ขึ้นกับแพลตฟอร์มหรือระบบปฏิบัติการใด ๆ ไม่ว่าจะเป็นระบบ Windows Linux Mac OS ฯลฯ ดังนั้นเมื่อพัฒนาเว็บด้วยเจเอสพีในแพลตฟอร์มหนึ่งก็สามารถย้ายไปใช้งานกับแพลตฟอร์มอื่นๆ ได้ไม่ยาก
2) ใช้งาน Java API ได้หลากหลาย เจเอสพีสามารถเรียกใช้งาน Java API ได้หลากหลาย ซึ่ง Java API คือกลุ่มของคลาสที่ช่วยอำนวย ความสะดวกในการใช้งานต่างๆ เช่น การจัดการเกี่ยวกับเน็ตเวิร์ก การติดต่อฐานข้อมูลการจัดการทางด้าน การรับส่งอีเมล เป็นต้น
3) นำคอมโพเนนต์กลับใช้ได้อีก ไม่ต้องเสียเวลาสร้างใหม่ เราสามารถนำจาวาบีนมาใช้งานร่วมกับสคริปต์เจเอสพีได้เพราะจาวาบีนเป็นคอมโพเนนต์ที่เขียนขึ้นมาเพื่อใช้สำหรับทำงานหรือทำหน้าที่อย่างใดอย่างหนึ่ง และสามารถนำกลับมาใช้งานได้เสมอ ดังนั้นเราจึงไม่ต้องเสียเวลาเขียนสคริปต์เจเอสพีเพื่อทำงานนั้นทุกครั้ง จึงสามารถพัฒนาเว็บไซต์ให้เสร็จได้เร็วยิ่งขึ้น
4) มีความยืดหยุ่นในการใช้งาน ในการเขียนสคริปต์เจเอสพี เราสามารถกำหนดแท็กใหม่ขึ้นมาใช้งาน ให้เหมาะสมกับความต้องการได้ นอกจากนี้ภาษาที่ใช้เขียนสคริปต์นั้น เจเอสพีไม่ได้จำกัดเฉพาะภาษาจาวาเท่านั้น ตามหลักการแล้วเราสามารถใช้ภาษาอื่นๆ ในการเขียนสคริปต์ได้ รวมทั้งสามารถนำไปใช้งานร่วมกับ XML ได้อย่างมีประสิทธิภาพอีกด้วย
5) มีความปลอดภัยสูง เจเอสพีมีระบบการจัดการข้อผิดพลาดต่างๆ ไม่ว่าจะเป็นข้อผิดพลาดที่เกิดขึ้นในระหว่างการเขียนสคริปต์ นอกจากนี้ยังมีระบบตรวจสอบความปลอดภัย ที่สามารถป้องกันการเขียนสคริปต์ที่ไม่ถูกต้องซึ่งอาจจะส่งผลกระทบต่อการใช้ทรัพยากรต่างๆ ภายในเซิร์ฟเวอร์รวมทั้งการป้องกันการทำงานของสคริปต์ในระหว่างที่ผู้ชมและใช้บริการเว็บไซต์ที่พัฒนาด้วยเจเอสพี

2.6.3 เปรียบเทียบเจเอสพีและเอเอสพี

เจเอสพีและเอเอสพี (ASP: Active Server Pages) เป็นเทคโนโลยีที่ ถูกออกแบบมาให้สามารถสร้างเว็บเพจในลักษณะของ Dynamic Content ได้เหมือนกัน แต่เหตุผลที่ควรเลือกใช้งานเจเอสพี เพราะเจเอสพีสามารถทำงานได้บนระบบปฏิบัติการทุกตัว ในขณะที่เอเอสพีสามารถทำงานได้ เฉพาะบนระบบปฏิบัติการ Microsoft Windows เท่านั้น

2.6.4 เปรียบเทียบเจเอสพีและพีเอชพี

เจเอสพีเป็นภาษาสคริปต์ที่มีพื้นฐานมาจากภาษาจาวา ซึ่งจาวาจะมี API (Application Programming Interface) ต่างๆให้เลือกใช้งานอยู่มากมาย ไม่ว่าจะเป็นเรื่องของการติดต่อกับเครือข่าย การติดต่อกับฐาน ข้อมูล เป็นต้น ในขณะที่พีเอชพี (PHP Hypertext Preprocessor) จะมี API เหล่านี้ในเวอร์ชั่นหลังๆ แต่ก็ไม่ได้มีรูปแบบเป็นการเขียน โปรแกรมเชิงวัตถุ เหมือนอย่างเจเอสพี ทำให้การพัฒนาโปรแกรมขนาดใหญ่และมีความซับซ้อนด้วยภาษาพีเอชพี เป็นไปด้วยความยากลำบากและล่าช้ากว่าการพัฒนาโปรแกรมภาษาเจเอสพี

บทที่ 3
การออกแบบและการพัฒนาซอฟต์แวร์

ในบทนี้จะกล่าวถึงการออกแบบ และพัฒนาซอฟต์แวร์ ซึ่งจะเกี่ยวข้องถึงรายละเอียดของซอฟต์แวร์ และอุปกรณ์ / เครื่องมือที่ใช้พัฒนา และโครงสร้างต่างๆ รวมไปถึงการทำงานของซอฟต์แวร์

3.1 แนวคิดในการพัฒนา

การเข้าศึกษาในภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในช่วงการศึกษาของชั้นปีที่ 1 จะมีนักศึกษาบางส่วนที่ไม่สามารถเรียนได้ทำให้ต้องถูกรีไทร์ ออกไป หรือบางส่วนที่เรียนแล้วได้คะแนนในการเรียนไม่ดี เนื่องจากไม่ถนัดในการเรียนในวิชาของภาควิชาวิศวกรรมคอมพิวเตอร์ จึงเกิดความต้องโปรแกรมที่จะบอกได้ว่า คนที่จะเข้ามาศึกษาในภาควิชาวิศวกรรมคอมพิวเตอร์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังนั้น มีระดับผลการเรียนเท่าใด โดยใช้ข้อมูลที่จะเป็นตัววัดจากคะแนน GAT/PAT เนื่องจากการสอบGAT/PAT เป็นการสอบที่ใช้ข้อสอบกลาง ซึ่ง นักเรียนที่จบจากชั้นมัธยมศึกษาปีที่ 6 ส่วนใหญ่ หรือ เกือบทั้งหมดจะผ่านการสอบนี้ เนื่องจากคะแนน GAT / PAT เป็นผลส่วนหนึ่งของคะแนนที่จะนำไปคิดในการสมัคร สอบรับตรง เข้ามหาวิทยาลัย หรือ ใช้เป็นส่วนการคิดคะแนนในการเข้ามหาวิทยาลัยแบบแอดมิชชั่นหรือ การเข้าศึกษาแบบอื่น ๆ จึงใช้คะแนนGAT / PAT มาเป็น ข้อมูลในการระบุว่าใครจะได้คะแนนเท่าใดเมื่อมาเรียนในภาควิชาวิศวกรรมคอมพิวเตอร์

ในส่วนของตัวประมวลผลว่าคนที่เข้ามาศึกษาในภาควิชาวิศวกรรมคอมพิวเตอร์จะได้คะแนนเท่าใดนั้น ได้เลือกใช้โครงข่ายประสาทเทียมเป็นโปรแกรมในการตัดสินใจ โดยโครงข่ายประสาทเทียมนี้ จะแบ่งออกเป็นสามส่วนคือ ส่วนอินพุตเลเยอร์ ส่วนฮิดเดนเลเยอร์และ ส่วนเอาต์พุตเลเยอร์โดยระหว่างส่วนนี้จะมีข้อมูลน้ำหนักที่ใช้เป็นส่วนคำนวณเพื่อให้ข้อมูลที่ออกมาถูกต้องที่สุด ซึ่งในการปรับน้ำหนักนี้จำเป็นต้องใช้ข้อมูลที่มาจากข้อมูลจริงซึ่งประกอบด้วยอินพุตและเอาต์พุต เพื่อนำมาปรับน้ำหนักของโครงข่ายประสาทเทียมตามค่าข้อมูลที่เก็บมาได้ เรียกว่า การฝึกฝน เมื่อทำการปรับน้ำหนักจนได้ค่าความผิดพลาดออกมาจนเป็นที่ยอมรับได้แล้ว จึงจะนำน้ำหนักที่ได้มาใช้กับโปรแกรมจริง สำหรับการนำมาใช้กับโครงงาน จะทำการเก็บข้อมูลจริงจากนักศึกษาภาควิชาวิศวกรรมคอมพิวเตอร์ ชั้นปีที่ 2, 3 และ 4 โดยเก็บข้อมูล GAT / PAT และข้อมูลเกรดที่ได้ตอนปี 1 นำมาเป็นข้อมูลสำหรับการฝึกฝนโครงข่ายประสาทเทียม โดยข้อมูลผลลัพธ์ ที่ต้องเก็บเป็นเกรดเป็นเกรดรวม

(GPA) ของตอนปี 1 นั้น เนื่องจากโปรแกรมมีจุดประสงค์ในการทำนายว่า ผู้ที่จะเข้าศึกษาในภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง มีผลการเรียน หรือเกรดเฉลี่ยในตอนชั้นปีที่ 1 จึงต้องใช้ผลลัพธ์เดียวกันมาทำการฝึกฝน

สำหรับในส่วนที่ติดต่อกับผู้ใช้งานนั้น จะเป็นลักษณะของเว็บแอพพลิเคชั่นเพื่อให้ผู้ใช้งานสามารถใช้งานระบบได้ ไม่ว่าจะใช้อุปกรณ์ใด เพื่อให้สอดคล้องกับเป้าหมายของโครงงานหนึ่ง คือให้ผู้ที่สัมภาษณ์ ผู้ที่จะเข้าศึกษาในชั้นปีที่ 1 ภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ใช้งาน ดังนั้น ระบบจึงควรใช้งานได้ไม่ว่าจะอยู่บนอุปกรณ์ใด เช่น สมาร์ทโฟน แทปเล็ต หรือ ใช้อุปกรณ์อื่นๆโดยในส่วน Interface จะให้ผู้ใช้งาน ทำการกรอกคะแนน GAT / PAT สำหรับตรวจสอบผลเท่านั้น นอกจากนี้ส่วนของผู้พัฒนาโปรแกรม จะทำส่วนติดต่อผู้ใช้งานอยู่บน ระบบปฏิบัติการณ์ Microsoft Windows สำหรับให้ผู้พัฒนาปรับเปลี่ยนค่าเกี่ยวกับส่วนประมวลผลโครงข่ายประสาทเทียม พร้อมกับเช็คค่าคามผิดพลาดได้

สำหรับโครงงานนี้ นอกจากสร้างโปรแกรมสำหรับให้ผู้ใช้งานได้ตรวจสอบระดับผลการเรียนว่า อยู่ในระดับไหนเมื่อเข้ามาศึกษาในภาควิชา วิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าลาดกระบังแล้วนั้น ยังทำการศึกษาความเปลี่ยนแปลงของส่วนประมวลผลโครงข่ายประสาทเทียม เมื่อเปลี่ยนค่าอินพุตหรือ จำนวนของฮิดเดนโหนด โปรแกรมสำหรับผู้พัฒนาจึงต้องมีความสามารถในการแสดงผล เพื่อช่วยเหลือในการเก็บผลการทดลอง เช่น แสดงผลกราฟ, คำนวณค่าอัตราความผิดพลาดและค่าน้ำหนักในเส้นทางต่างๆ จึงทำการเพิ่มการแสดงผลเหล่านี้ในโปรแกรมสำหรับผู้พัฒนาเข้าไปด้วย

3.2 การออกแบบแบบสอบถาม

ในส่วนประมวลผลของการทำนายผลการเรียนล่วงหน้านั้น โปรแกรมสำหรับผู้พัฒนานั้นได้ใช้แนวคิดของโครงข่ายประสาทเทียมในการประมวลผลข้อมูลจึงจำเป็นต้องใช้ข้อมูลซึ่งเป็นข้อเท็จจริงของนักศึกษาภาควิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกะบังชั้นปีที่ 2, 3, 4 ในการฝึกฝนโครงข่ายประสาทเทียม

เพื่อให้ได้ตัวแปรในการนำมาฝึกฝนนั้น จึงจำเป็นต้องใช้มีการออกแบบ แบบสอบถามเพื่อใช้ในการเก็บข้อมูลในส่วนนี้ จะประกอบไปด้วยคะแนน GAT, PAT1, PAT2 และPAT3 ซึ่งจะแบ่งช่วงคะแนนออกเป็นช่วงละ 30 คะแนน ตั้งแต่ 0 ถึง 300 คะแนน โดยอิงตามผลการประกาศคะแนนในแต่ละปีของ สถาบันทดสอบทางการศึกษาแห่งชาติ และวิธีการเข้าศึกษา ซึ่งได้แก่ แอดมิชชั่น รับตรง โควตาช้างเผือก โควตาบุคลากรและนักกีฬา เป็นตัวแปรต้น และผลการเรียนเฉลี่ยในช่วงชั้นปีที่ 1 โดยทำการแบ่งช่วงของเกรดออกเป็น 4 ช่วงได้แก่ช่วง 1.00-1.99, 2.00-2.49, 2.50-2.99 และ 3.00-

4.00 เนื่องจากตัวเลขดังกล่าวสามารถแบ่งกลุ่มของนักศึกษาที่มีผลการเรียนแตกต่างกันได้เป็นอย่างดี โดยข้อมูลส่วนนี้จะถูกใช้เป็นตัวแปรตามหรือผลลัพธ์ของระบบ

หลังจากนั้นจึงทำการเปลี่ยนแปลงข้อมูลให้อยู่ในรูปแบบของชุดข้อมูล (Data set) ที่ต้องการ โดยผู้พัฒนาจะทำการนำข้อมูลที่ได้รับจากแบบสอบถามนั้น แปลงเป็นข้อมูลจากเชิงคุณภาพให้อยู่ในรูปของเชิงประมาณ ก่อนที่จะนำมาเก็บลงในไฟล์ Text เพื่อใช้เป็นชุดข้อมูลสำหรับฝึกฝนต่อไป

แบบสอบถามที่ใช้ มีรูปแบบดังรูปที่ 3.1

แบบสอบถามเพื่อเก็บข้อมูลในการสร้างโปรแกรมทำนายผลการเรียน
จากคะแนน GAT/PAT โดยใช้โครงข่ายประสาทเทียมของนักศึกษา
สาขาวิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

แบบสอบถามนี้ จัดทำขึ้นเพื่อเก็บข้อมูลผลการเรียน ของนักศึกษา สาขาวิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เพื่อนำข้อมูลไปใช้ในการทำโครงงาน วิชา PROJECT1 (01076301) ของนักศึกษาชั้นปีที่ 4

คำชี้แจง กรุณา ทำเครื่องหมาย ✓ ใน ☐ ข้อที่ตรงกับความเป็นจริงที่สุด

ตอนที่ 1 ข้อมูลเบื้องต้นของผู้ตอบแบบสอบถาม

วิธีการเข้าศึกษา
- ☐ รับตรง
- ☐ แอดมิชชั่น
- ☐ โควต้าช้างเผือก
- ☐ โควต้าบุคลากร / นักกีฬา

ตอนที่ 2 ข้อมูลผลการเรียนของผู้ตอบแบบสอบถาม

GPA ปี 1	☐	1.00 - 1.49	☐	1.50 - 1.99
(เกรดเฉลี่ยรวม เทอม 1	☐	2.00 - 2.49	☐	2.50 - 2.99
และ เทอม 2)	☐	3.00 - 3.49	☐	3.50 - 4.00

คำชี้แจง กรุณา ทำเครื่องหมาย ✓ ในช่องที่ตรงกับความเป็นจริงที่สุด

ลำดับที่	ข้อมูลคะแนน	0.00 - 30.00	30.01 - 60.00	60.01 - 90.00	90.01 - 120.00	120.01 - 150.00	150.01 - 180.00	180.01 - 210.00	210.01 - 240.00	240.01 - 270.00	270.01 - 300.00
1	คะแนน GAT										
2	คะแนน PAT 1										
3	คะแนน PAT 2										
4	คะแนน PAT 3										

ขอขอบคุณที่กรุณาตอบแบบสอบถาม

รูปที่ 3.1 ตัวอย่างแบบสอบถามที่ใช้ในการเก็บข้อมูลนักศึกษา

3.3 เครื่องมือที่ใช้ในการพัฒนา

ในการพัฒนาโปรแกรม จำเป็นต้องติดตั้งเครื่องมือ หรือ สร้างสภาวะ เพื่อให้เหมาะสมต่อระบบ ดังนี้

3.3.1 ระบบปฏิบัติการ

ใช้ระบบปฏิบัติการ Microsoft Windows ในการพัฒนา เนื่องจากเป็นระบบปฏิบัติการที่คุ้นเคย และ ง่ายต่อการใช้งานในด้านต่าง ๆ มากกว่าระบบปฏิบัติการอื่น นอกจากนี้ โปรแกรมที่ใช้ในการพัฒนาสามารถนำมาติดตั้งเพื่อใช้งานบนระบบปฏิบัติการ Microsoft Windows ได้ จึงเลือกระบบปฏิบัติการนี้ในการทำงาน

3.3.2 ภาษาโปรแกรมที่ใช้

ใช้ภาษาจาวา (Java Language) ในการพัฒนาโปรแกรมในส่วนของการฝึกฝนโครงข่ายประสาทเทียมในฝั่งของผู้พัฒนา เนื่องจากเป็นภาษาที่สนับสนุนการเขียนในลักษณะของ Object Oriented Programming ที่มีความนิยมใช้งานแพร่หลาย ทำให้สามารถหาเครื่องมือในการพัฒนาง่าย มีไลบรารี ให้ใช้งานมาก โปรแกรมที่เขียนขึ้นโดยใช้ภาษาจาวา จะมีความสามารถทำงานได้ในระบบปฏิบัติการที่แตกต่างกัน

ในส่วนของเว็บแอพพลิเคชั่นสำหรับผู้ใช้งานนั้น จะได้ภาษาเอชทีเอ็มแอล (HTML) ในหน้าแรกสำหรับผู้ใช้งานกรอกแบบฟอร์มของคะแนนสอบ GAT/PAT และวิธีการเข้า แล้วส่งข้อมูลอินพุตเหล่านี้ไปประมวลผลต่อโดยใช้ Servlet อ่านอินพุตที่ส่งมาจากผู้ใช้งาน เพื่อนำมาเข้าสู่โครงข่ายประสาทเทียม และคำนวณผลลัพธ์ที่ได้เพื่อส่งกลับไปยังผู้ใช้งาน โดยใช้ภาษาเจเอสพี (JSP) ในการรับผลจากการประมวลแล้วนำมาแสดงแก่ผู้ใช้งาน

3.3.3 เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์

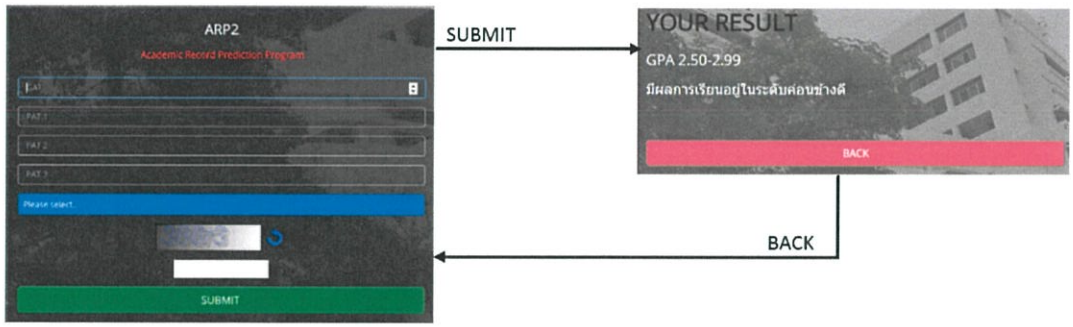
เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์ คือ Netbeans IDE 8.0.2 เนื่องจากนำไปใช้งานง่าย ไม่ต้องศึกษาเพิ่มเติมมาก และมีส่วนติดต่อประสานงานสำหรับผู้ใช้ที่ได้เป็นสัดส่วน เข้าใจง่าย ซึ่งทำให้เกิดความสะดวกต่อการสร้างสรรค์งานได้ตรงกับความต้องการ นอกจากนี้ยังมีโมดูลสำเร็จทุกอย่างในตัวเน็ตบีนส์เอง จึงแทบไม่ต้องหาปลั๊กอินมาลงเพิ่มและใช้ Apache Tomcat เป็น Web server สำหรับรองรับการใช้งานของ JSP/Servlet เนื่องจากเป็น Open source ซึ่งสามารถโหลดมาใช้งานได้ฟรี และค่อนข้างมีความเสถียรต่อการใช้งาน

3.4 รายละเอียดส่วนติดต่อผู้ใช้งาน (User Interface)

ในส่วนของการติดต่อผู้ใช้งาน จะถูกแบ่งออกเป็นสองส่วน ได้แก่ ส่วนของผู้ใช้งาน และ ส่วนของผู้พัฒนาโปรแกรม โดยส่วนของผู้ใช้งาน จะทำงานในลักษณะของเว็บแอพพลิเคชั่นและส่วนของผู้พัฒนาโปรแกรม จะเป็นการทำงานของ Windows Interface

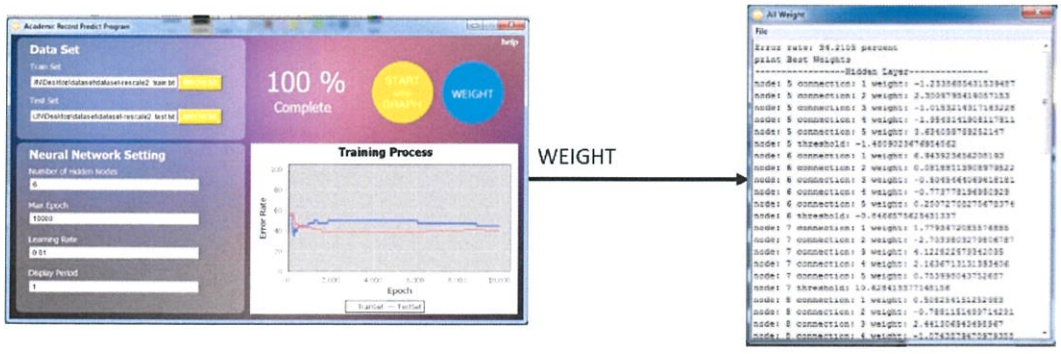
3.4.1 ผังการเชื่อมต่อของส่วนติดต่อผู้ใช้งาน

ในส่วนของผู้ใช้งานจะถูกแบ่งเป็นสองส่วน คือ ส่วนสำหรับใส่ค่าอินพุตของโปรแกรม และส่วนของการแสดงค่าผลลัพธ์จากค่าอินพุตดังรูปที่ 3.2



รูปที่ 3.2 แผนผังการเชื่อมต่อของเว็บแอพพลิเคชั่นสำหรับผู้ใช้งาน

ในส่วนของผู้พัฒนาโปรแกรมจะแบ่งเป็นสองส่วน คือส่วนของการตั้งค่าการฝึกฝนในโปรแกรม และ ส่วนของการติดตามผลลัพธ์ ของโปรแกรม เมื่อค่าของน้ำหนักในโปรแกรมเป็นที่ยอมรับได้แล้ว จะสามารถทำการ อัพโหลดค่าน้ำหนักของโครงข่ายประสาทเทียม ไปใช้งานในส่วนของเว็บแอพพลิเคชั่นผู้ใช้งาน เพื่อใช้งานได้ทันที ดังรูปที่ 3.3



รูปที่ 3.3 แผนผังการเชื่อมต่อของโปรแกรมสำหรับผู้พัฒนา

3.4.2 หน้าต่างหลักของเว็บส่วนของผู้ใช้งาน

เมื่อเข้าใช้งานผ่าน Url ของเว็บ จะแสดงหน้านี้ขึ้นมาเพื่อให้ ผู้ใช้งานกรอกข้อมูลที่จำเป็นในการใช้งานโปรแกรม ได้แก่ ข้อมูลคะแนน GAT, คะแนน PAT1, PAT2, PAT3 และ วิธีการเข้าศึกษา โดยมีปุ่มใช้งานหลักคือ ปุ่ม SUBMIT เพื่อเป็นการยืนยันคะแนนและดูผลลัพธ์ที่ได้ ดังรูปที่ 3.4

รูปที่ 3.4 หน้าใช้งานหลักของเว็บแอพพลิเคชั่น

ในส่วนของเมนูจะมีเมนูให้ผู้ใช้งานได้เลือก คือส่วนของ Home, About, CE Curriculum และ Contact ให้ผู้ใช้งานเข้าไปในส่วนหัวข้อที่สนใจสำหรับการใช้งานเว็บแอพพลิเคชั่นบนอุปกรณ์อื่นนอกเหนือจากคอมพิวเตอร์ เช่น สมาร์ทโฟน ส่วนการใช้งานเมนูจะเปลี่ยนเป็นใช้ปุ่มกด เพื่อเรียกใช้งานส่วนเมนูแทน เพื่อความสะดวกในการใช้งานมากขึ้น ดังรูปที่ 3.5 และ 3.6

รูปที่ 3.5 ส่วนเมนูของเว็บแอพพลิเคชั่น

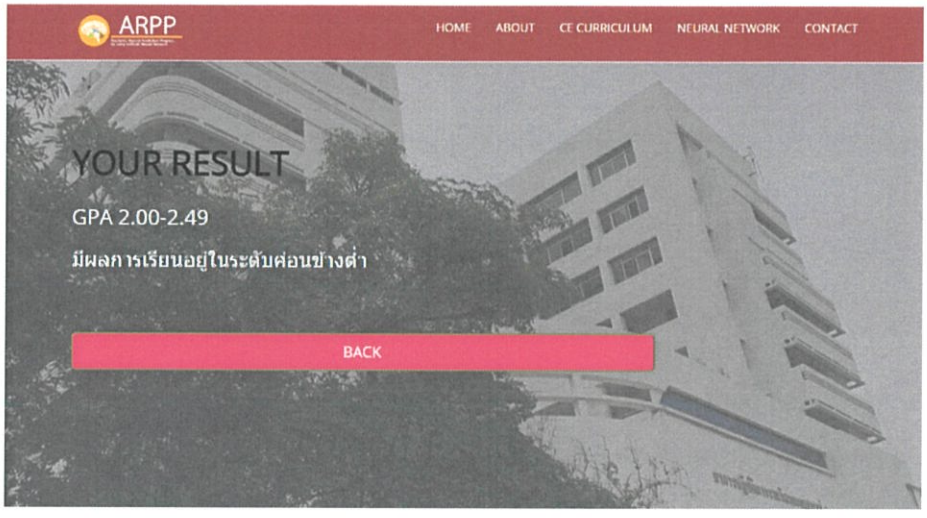
รูปที่ 3.6 ส่วนเมนูของเว็บแอพพลิเคชั่นบนอุปกรณ์สมาร์ทโฟน

3.4.3 หน้าต่างส่วนเนื้อหาของผู้ใช้งาน

ในส่วนนี้จะเป็นส่วนเนื้อหาต่าง ๆ ที่ผู้ใช้งานสามารถเข้าใช้งานได้ ได้แก่

3.4.3.1 หน้าต่างผลลัพธ์ของผู้ใช้งาน

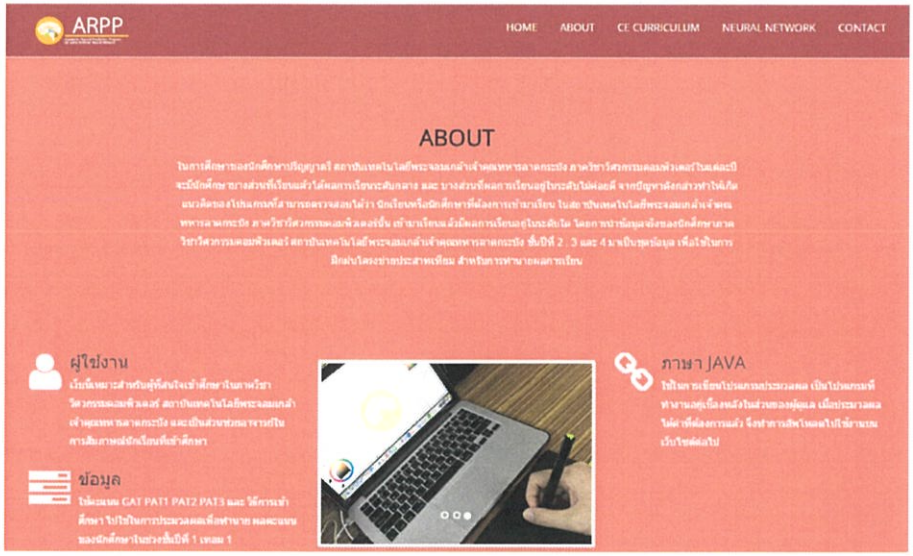
ในส่วนนี้จะแสดงผลลัพธ์จากการกรอกคะแนนของผู้ใช้งาน เมื่อกดปุ่ม SUBMIT จะเข้าสู่หน้าต่างนี้ ซึ่งจะบอกผลลัพธ์ที่บริเวณด้านซ้ายของหน้าต่างแสดงผล โดยผู้ใช้งานสามารถแก้ไขคะแนนเพื่อตรวจสอบผลลัพธ์ใหม่ได้จากการกดปุ่ม BACK จะเปลี่ยนหน้าต่างกลับไปที่ส่วนหน้าต่างหลักของเว็บส่วนของผู้ใช้งานอีกครั้ง ดังรูปที่ 3.7 หรือกดใช้งานส่วนเมนูเพื่อเข้าใช้งานเนื้อหาส่วนอื่น ๆ ของเว็บ



รูปที่ 3.7 หน้าแสดงผลลัพธ์ ผลการเรียนของผู้ใช้งาน

3.4.3.2 หน้าต่างใช้งานส่วน ABOUT

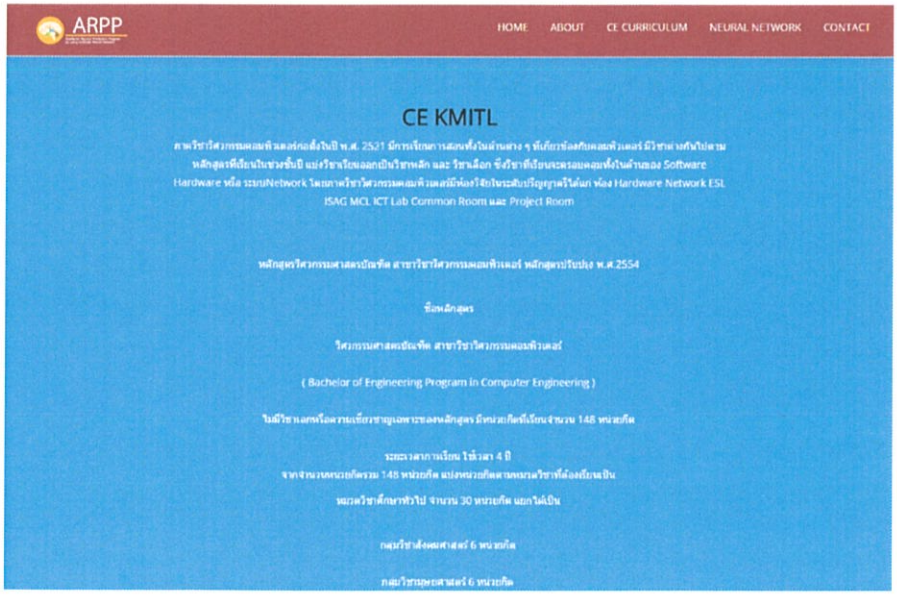
หน้าต่างส่วนนี้จะอธิบายผู้ใช้งานเกี่ยวกับเว็บแอพพลิเคชั่น ว่ามีที่มาอย่างไร มีวัตถุประสงค์การใช้งาน และส่วนของโปรแกรมอย่างไรบ้าง โดยในส่วนนี้จะแสดงรูปที่เกี่ยวข้องกับการสร้างเว็บแอพพลิเคชั่นไว้



รูปที่ 3.8 หน้าต่างใช้งานส่วน ABOUT

3.4.3.3 หน้าต่างใช้งานส่วน CE CURRICULUM

ในส่วนนี้จะมีเนื้อหาเกี่ยวกับประวัติ และข้อมูลทั่วไป ของภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในส่วนของเนื้อหาหลัก จะเป็นเนื้อหาเกี่ยวกับหลักสูตรของภาควิชา โดยหลักสูตรที่แสดงไว้จะเป็น หลักสูตรวิศวกรรมศาสตร์บัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ หลักสูตรปรับปรุง พ.ศ.2554 ซึ่งเป็นหลักสูตรที่ใช้สำหรับนักศึกษาที่จะเข้ามาเป็นเป็นนักศึกษาปี 1 ของภาควิชาวิศวกรรมคอมพิวเตอร์ ดังรูปที่ 3.9

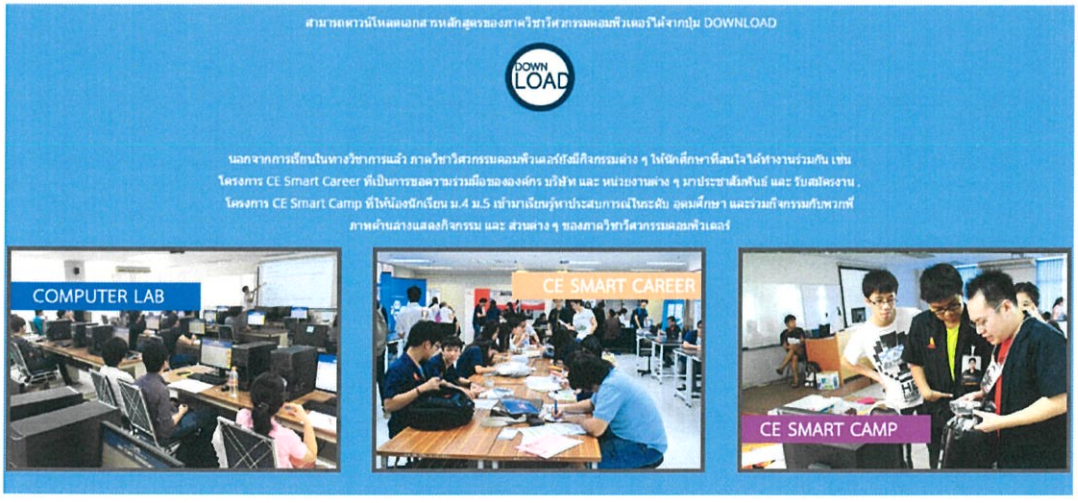


รูปที่ 3.9 หน้าต่างใช้งานส่วน CE CURRICULUM

เมื่อผู้ใช้งานเลื่อนชมเนื้อหา จะพบปุ่มสำหรับดูแผนการเรียนของแต่ละชั้นปี ในหลักสูตรนี้ หลังจากกดปุ่มเพื่อดูรายละเอียด เว็บแอพพลิเคชั่นจะแสดงตารางข้อมูลแผนการเรียนของชั้นปีนั้น

รูปที่ 3.10 ปุ่มแสดงแผนการเรียน ในหน้าต่างใช้งานส่วน CE CURRICULUM

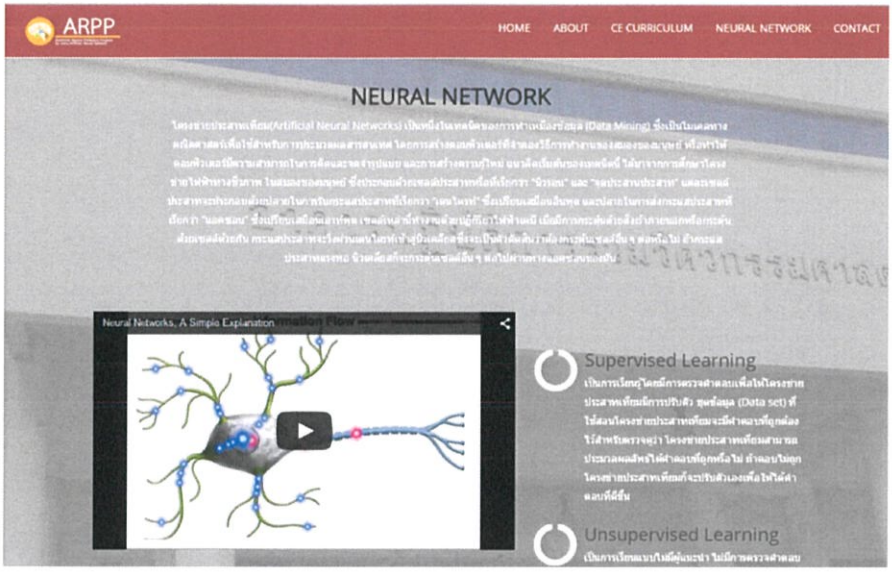
ในส่วนสุดท้ายผู้ใช้งานสามารถทำการดาวน์โหลด หลักสูตรวิศวกรรมศาสตร์บัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ ได้จากปุ่ม DOWNLOAD และดูภาพที่เกี่ยวข้องกับภาควิชาวิศวกรรมคอมพิวเตอร์ได้ เช่น กิจกรรม CE Smart Career หรือ Computer Lab โดยสามารถขยายขนาดรูปได้จากการเลื่อน ลูกศรของเมาส์ไว้บนรูปและกดที่แว่นขยาย (ฟังก์ชันส่วนนี้ไม่สามารถใช้งานบนสมาร์ทโฟนได้)



รูปที่ 3.11 ปุ่ม Download และ ภาพกิจกรรมของภาควิชาวิศวกรรมคอมพิวเตอร์

3.4.3.4 หน้าต่างอธิบายการทำงานของโครงข่ายประสาทเทียม

เป็นส่วนที่อธิบายเนื้อหาที่เกี่ยวข้องกับ โครงข่ายประสาทเทียมโดยจะประกอบด้วยเนื้อหาว่าโครงข่ายประสาทเทียมมีที่มาจากอะไร ประกอบด้วยอะไรบ้าง และมีวิธีการทำงานอย่างไร



รูปที่ 3.12 หน้าต่างอธิบายการทำงานของโครงข่ายประสาทเทียม

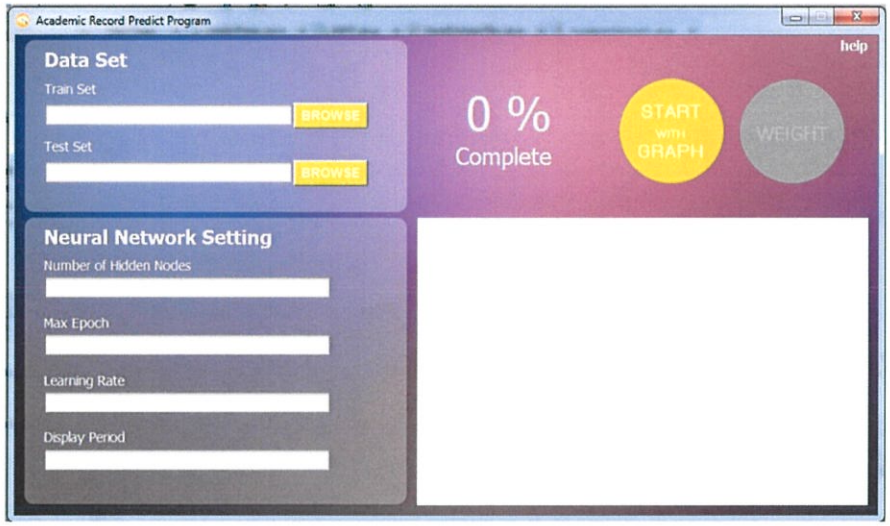
3.4.3.5 หน้าต่างใช้งานส่วน CONTACT US

ในส่วนนี้จะบอกผู้ใช้งานเกี่ยวกับข้อมูลการติดต่อ กับภาควิชาวิศวกรรมคอมพิวเตอร์ ได้แก่ รายละเอียดที่อยู่ และ เบอร์โทรติดต่อ โดยผู้ใช้งานสามารถเข้าชมเว็บไซต์ของภาควิชาได้ จากการกดปุ่ม CE KMITL

รูปที่ 3.13 หน้าต่างใช้งานส่วน CONTACT US

3.4.4 หน้าต่างหลักของโปรแกรมส่วนผู้พัฒนา

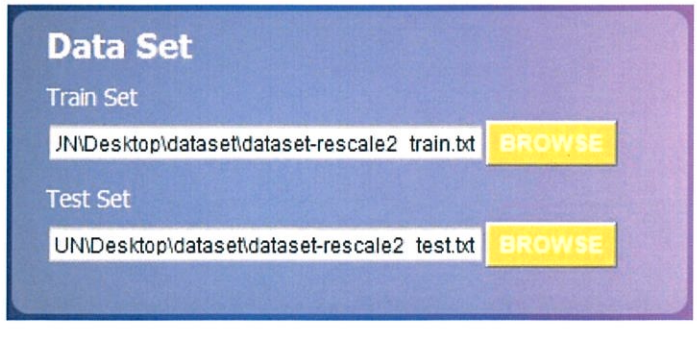
หน้าต่างหลักของโปรแกรมสำหรับผู้พัฒนานี้ จะเป็นส่วนที่ผู้พัฒนาสามารถใช้งานเพื่อทำการฝึกฝนโครงข่ายประสาทเทียมเพื่อหาค่าน้ำหนักที่มีความแม่นยำมากที่สุด เพื่อนำไปใช้ต่อในโปรแกรมทำนายผลการเรียนล่วงหน้าต่อไป ซึ่งจะประกอบด้วยส่วนประกอบต่างๆ ดังรูปที่ 3.14 ในส่วนคู่มือการใช้งานนั้น สามารถกดดูได้ที่ปุ่ม Help มุมบนขวาของโปรแกรม



รูปที่ 3.14 หน้าหลักของโปรแกรมสำหรับผู้พัฒนา

3.4.4.1 ส่วนของการเลือกชุดข้อมูลเพื่อทำการฝึกฝนและทดสอบ

เป็นส่วนหน้าต่างที่ให้ผู้พัฒนา สามารถเข้าไปกำหนดไฟล์ชุดข้อมูล ซึ่งได้แก่ ชุดฝึกฝน (Train Set) เป็นชุดข้อมูลที่ใช้สำหรับฝึกฝนให้กับโครงข่ายประสาทเทียม และชุดทดสอบ (Test Set) เป็นชุดที่ไว้สำหรับใช้ตรวจสอบความแม่นยำหรือความผิดพลาดที่เกิดขึ้นหลังจากที่ได้ทำการฝึกฝนโครงข่ายประสาทเทียมไปแล้วตามที่ได้กำหนดไว้ซึ่งได้แก่ Max Epoch และ Display Period ซึ่งจะกล่าวต่อในส่วนถัดไป ดังรูปที่ 3.15



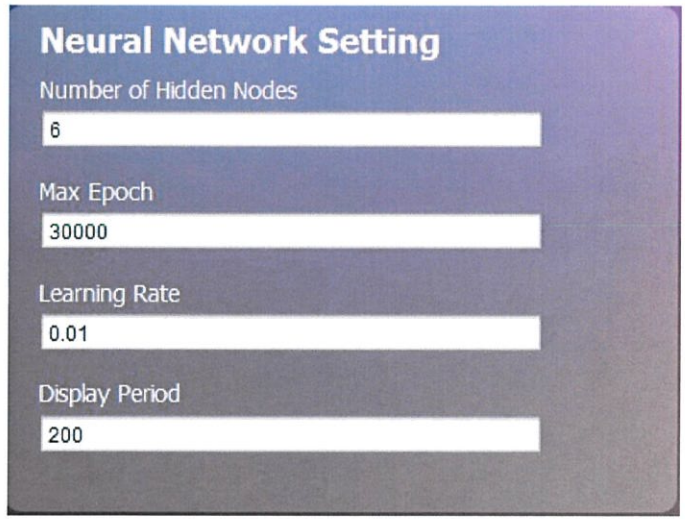
รูปที่ 3.15 ส่วนสำหรับการกำหนดไฟล์ที่ใช้ฝึกฝนและทดสอบ

3.4.4.2 ส่วนของการตั้งค่าฝึกฝนโครงข่ายประสาทเทียม

เป็นส่วนที่ผู้พัฒนา สามารถปรับเปลี่ยนค่าต่าง ๆ ในส่วนการคำนวณของโปรแกรม หรือก็คือส่วนของการเรียนรู้ของโครงข่ายประสาทเทียม โดยส่วนที่สามารถปรับเปลี่ยนได้ ได้แก่

1. จำนวนของฮิดเดนโหนด (Number of Hidden nodes) คือจำนวน node ในส่วนของฮิดเดนเลเยอร์โดยจำนวนฮิดเดนเลเยอร์ที่ใช้ในโปรแกรมนี้ จะตั้งไว้ที่ 1 เลเยอร์
2. จำนวนรอบสูงสุดของการทำงาน (Max Epoch) เป็นจำนวนรอบที่กำหนดให้โครงข่ายประสาทเทียมทำการเรียนรู้ ถ้าจำนวนยิ่งมาก โครงข่ายประสาทเทียมยิ่งเรียนรู้มากขึ้น
3. ค่าอัตราการเรียนรู้ (Learning Rate) เป็นค่าอัตราการเรียนรู้ของโครงข่ายประสาทเทียม ซึ่งตัวแปรนี้ส่งผลต่อการปรับค่าน้ำหนักในแต่ละครั้งที่ทำการฝึกฝน หากกำหนดค่านี้มากเกินไป อาจส่งผลต่อการหาค่าน้ำหนักที่ถูกต้องได้
4. ช่วงของการแสดงผล (Display Period) คือคาบของจำนวนรอบในการทำงานที่โครงข่ายจะทำการคำนวณค่าความผิดพลาด เพื่อนำมาแสดงผลในรูปแบบของกราฟ (Display Period) ยิ่งค่าน้อย กราฟที่ได้ยิ่งมีความละเอียดมากขึ้น

ซึ่งตัวแปรข้างต้นนั้น จำนวนฮิดเดนโหนด และอัตราการเรียนรู้ จะเป็นค่าที่ส่งผลกระทบโดยตรงต่อการเรียนรู้ของโครงข่ายประสาทเทียม ดังรูปที่ 3.16



รูปที่ 3.16 ส่วนสำหรับตั้งค่าพารามิเตอร์ที่ใช้สำหรับฝึกฝนข้อมูล

หลังจากที่ผู้พัฒนาได้ทำการกำหนดชุดข้อมูลที่ใช้สำหรับฝึกฝน ทดสอบ และกำหนดค่าพารามิเตอร์ต่างๆ สำหรับการฝึกฝนแก่โครงข่ายประสาทเทียมครบแล้ว โปรแกรมจะเริ่มทำการฝึกฝนทันทีที่ผู้พัฒนาคลิกปุ่ม Start with Graph

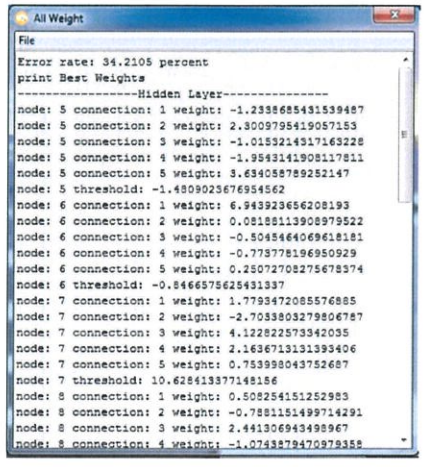
3.4.4.3 ส่วนของการแสดงผลลัพธ์

เมื่อการฝึกฝนเสร็จครบตามรอบที่ผู้พัฒนาได้กำหนดไว้แล้ว หรือสามารถสังเกตได้ที่เปอร์เซ็นต์การทำงานของโปรแกรม เมื่อโปรแกรมหยุดทำงานลง จะเกิดผลลัพธ์เกิดขึ้น ให้เห็น 2 ส่วน ได้แก่

ส่วนของการแสดงผลลัพธ์ของการฝึกฝนในลักษณะของกราฟความสัมพันธ์ระหว่างจำนวนรอบที่ทำการฝึกฝนและค่าอัตราความผิดพลาดที่เกิดขึ้น ดังรูปที่ 3.17 โดยความละเอียดของกราฟนั้นจะขึ้นอยู่กับค่า Display Period ที่ผู้พัฒนาได้กำหนดไว้

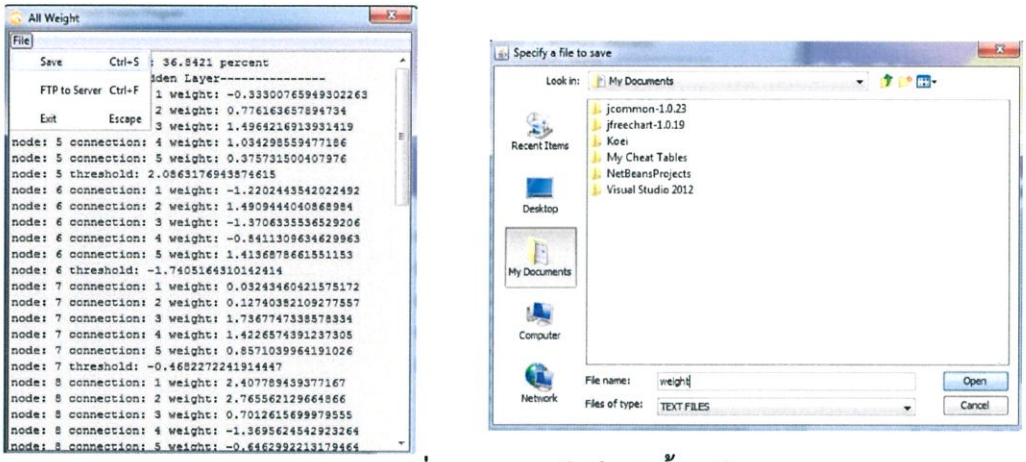
รูปที่ 3.17 กราฟแสดงความสัมพันธ์ระหว่างจำนวนรอบของการฝึกฝนและอัตราความผิดพลาด

ส่วนที่สอง คือส่วนของการแสดงค่าน้ำหนักของโครงข่ายประสาทเทียม ณ ตำแหน่งที่ฝึกฝนแล้วเกิดค่าความผิดพลาดต่ำที่สุด หรืออีกนัยหนึ่งคือเป็นค่าน้ำหนักที่มีประสิทธิภาพมากที่สุดและพร้อมที่จะนำไปใช้งานจริงได้ ดังรูปที่ 3.18 ซึ่งผู้พัฒนาสามารถดูค่าน้ำหนักเหล่านี้ได้จากการกดปุ่ม Weight ซึ่งจะสามารถกดได้หลังจากที่โปรแกรมทำการฝึกฝนเสร็จแล้ว



รูปที่ 3.18 หน้าต่างแสดงค่าน้ำหนักที่มีประสิทธิภาพมากที่สุด

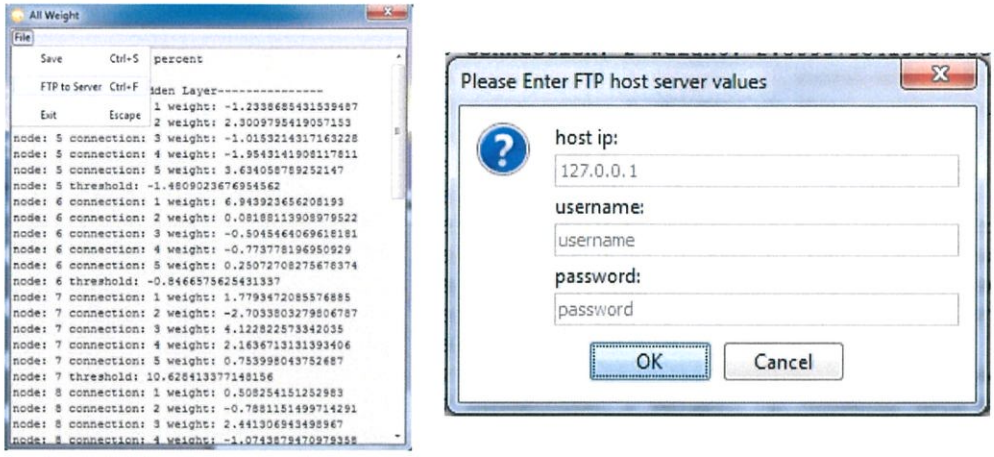
นอกจากนี้ผู้ใช้ยังสามารถบันทึกค่าน้ำหนักที่ได้จากการฝึกฝนซึ่งจะอยู่ในรูปแบบของไฟล์ Text เพื่อเก็บไว้ในเครื่องของตนเอง ดังรูปที่ 3.19 หรือสามารถอัพโหลดไฟล์ไปที่เว็บเซิร์ฟเวอร์เพื่อนำไปใช้งานจริงได้ทันที โดยใช้ FTP (File Transfer Protocol) โดยคลิกที่ปุ่ม File และสามารถเลือกใช้ฟังก์ชั่นที่ต้องการ



รูปที่ 3.19 การบันทึกค่าน้ำหนัก

ในส่วนของการ FTP นั้นผู้พัฒนาจำเป็นต้องติดตั้ง FTP Service ไว้ที่เครื่องเว็บเซิร์ฟเวอร์ด้วย ในที่นี้ผู้พัฒนาได้ใช้ File Zilla Server และทำการกำหนดชื่อผู้ใช้งานและพาสเวิร์ดผู้ใช้งาน และระบุ path ไว้ที่แฟ้ม weight ของโปรแกรมเท่านั้น โดยการกำหนดสิทธิในการใช้งาน FTP ของชื่อผู้ใช้งานดังกล่าว ต้องมีสิทธิในการเขียนไฟล์และลบไฟล์เป็นอย่างน้อย เพื่อที่จะใช้งานฟังก์ชั่นนี้ได้อย่างสมบูรณ์

การใช้งานนั้นผู้พัฒนาจะต้องกดปุ่ม File > FTP to Server และกรอกรายละเอียดต่างๆ ได้แก่ โฮส ชื่อผู้ใช้งาน และพาสเวิร์ด เพื่อใช้งาน FTP ไปยังเครื่องเซิร์ฟเวอร์ ดังรูปที่ 3.20 ซึ่งเมื่อทำกระบวนการดังกล่าวเสร็จสิ้นแล้ว ไฟล์ weight.txt จะถูกอัพโหลดขึ้นไปยังเซิร์ฟเวอร์เพื่อพร้อมใช้งานได้ในทันที



รูปที่ 3.20 การ FTP ไฟล์บันทึกค่าน้ำหนักไปยังเซิร์ฟเวอร์

3.5 โครงสร้างของระบบ

โครงสร้างของระบบ และกระบวนการทำงาน มีดังนี้

3.5.1 ส่วนติดต่อผู้ใช้งาน (Graphic user interface)

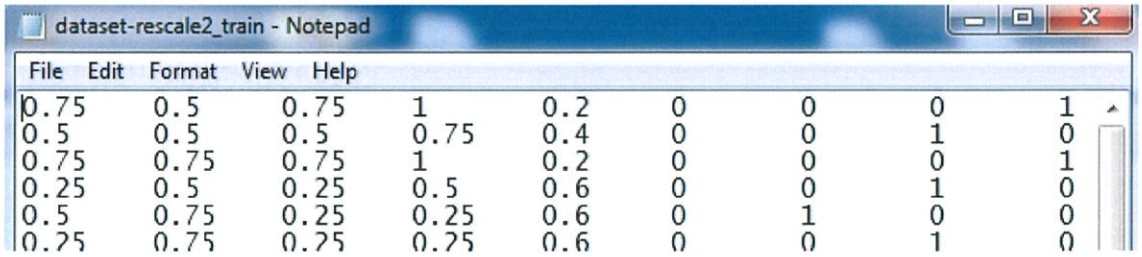
เป็นส่วนที่ทำงานติดต่อกับผู้ใช้งานโดยแบ่งผู้ใช้งานเป็นส่องกลุ่มคือ กลุ่มของผู้ใช้งานโปรแกรม และกลุ่มของผู้พัฒนาโปรแกรม ในส่วนของผู้ใช้งานโปรแกรมจะมีการส่งค่าต่างๆ ที่เกี่ยวข้องไปที่ส่วนของการประมวลผล และแสดงผลลัพธ์ ส่วนในด้านของส่วนผู้พัฒนาโปรแกรมจะมีกำหนดค่าต่างๆ และทำการฝึกฝนให้กับโครงข่ายประสาทเทียม เพื่อหาค่าน้ำหนักใหม่ และส่งไปปรับเปลี่ยนที่ส่วนของการประมวลผลในฝั่งของผู้ใช้งาน

3.5.2 ส่วนของการประมวลผล

ในด้านของผู้พัฒนานั้น โปรแกรมจะประมวลผลโดยรับพารามิเตอร์ต่างๆ ได้แก่ จำนวนรอบสูงสุดที่ใช้ในการฝึกฝน ค่าอัตราการเรียนรู้ จำนวนของโหนดในฮิดเดนเลเยอร์ คาบของการทดสอบโปรแกรมเพื่อพลอตกราฟความสัมพันธ์ระหว่างความผิดพลาดกับจำนวนรอบที่ฝึกฝน และไฟล์ชุดข้อมูลโดยพารามิเตอร์เหล่านี้ จะถูกกำหนดโดยผู้พัฒนาเท่านั้น

สำหรับไฟล์ชุดข้อมูลนั้น ผู้พัฒนาจะต้องทำการแบ่งข้อมูลออกเป็น 2 ส่วนได้แก่ชุดฝึกฝน ซึ่งเป็นข้อมูลที่ใช้สำหรับการฝึกฝน และชุดทดสอบสำหรับใช้ในการตรวจสอบอัตราความผิดพลาด ซึ่งชุดข้อมูลทั้ง 2 ประเภทนี้นั้น จะทำการจัดแบ่งจากข้อมูลที่เก็บจากแบบสอบถามทั้งหมด โดยใช้ 75 เปอร์เซ็นต์ของข้อมูลทั้งหมดเป็นชุดฝึกฝน และ 25 เปอร์เซ็นต์ที่เหลือจะถูกใช้เป็นชุดทดสอบ โดยข้อมูลทั้ง 2 กลุ่มนั้น จะต้องมีอัตราส่วนของเอาต์พุตของผลการเรียนที่เท่าๆ กัน จากนั้นจึงนำข้อมูล

เหล่านี้ มาทำให้อยู่ในรูปแบบไฟล์ Text โดยก่อนหน้านั้น จะต้องทำการกรองข้อมูลทั้งหมดให้อยู่ในเชิงปริมาณ และทำการรีสเกลของอินพุต ให้มีค่าอยู่ในช่วง 0.0 ถึง 1.0 เพื่อทำให้การประมวลผลและการปรับค่าน้ำหนักมีความเสถียร ในส่วนของอินพุตนั้น จะแทนผลลัพธ์ที่เกิดขึ้นจากข้อมูลอินพุตเรคคอร์ดนั้นๆ ด้วย 1 และผลลัพธ์อื่น ที่ไม่ใช่จะกำหนดให้เป็น 0 แล้วจึงนำเรคคอร์ดข้อมูลที่ผ่านการปรับให้เหมาะสมสำหรับการฝึกฝนนี้ ลงในไฟล์ Text ดังรูปที่ 3.21 ซึ่งข้อมูลจะถูกเก็บเป็นเรคคอร์ด ซึ่งแต่ละเรคคอร์ดนั้นจะประกอบไปด้วยค่าตัวเลข 9 ตัว โดยกำหนดให้อินพุตต่างๆ มีความหมาย และรีสเกลด้วยค่าที่กำหนดดังนี้



รูปที่ 3.21 ตัวอย่างชุดข้อมูลเพื่อใช้ในการฝึกฝนและทดสอบโครงข่ายประสาทเทียม

อินพุตตัวที่ 1 หมายถึง คะแนน GAT (General Aptitude Test) เป็นคะแนนจากการสอบวัดความถนัดทั่วไป ซึ่งมีคะแนนเต็ม 300 คะแนน แบ่งเป็นสองส่วนคือ ความสามารถในการ อ่าน เขียน คิดวิเคราะห์ และ แก้โจทย์ปัญหา150 คะแนนและส่วนความสามารถในการสื่อสารด้วยภาษาอังกฤษอีก 150 คะแนน โดยมีการปรับสเกลคะแนนเพื่อที่จะนำมาใช้เป็นชุดข้อมูล ดังนี้

คะแนนในช่วง	ค่าทีรีสเกล
0 – 210	0.25
211 – 240	0.50
241 – 270	0.75
271 – 300	1.00

อินพุตตัวที่ 2 - 4 หมายถึง คะแนน PAT (Professional and Academic Aptitude Test) เป็นคะแนนจากการสอบวัดความถนัดทางวิชาชีพและวิชาการ มีคะแนนเต็ม 300 คะแนน โดยวัดความรู้พื้นฐานที่จะไปเรียนในวิชาชีพนั้น ๆ โดยคะแนนที่ใช้ในโปรแกรม ได้แก่

PAT1 –ความถนัดทางคณิตศาสตร์

คะแนนในช่วง	ค่าทีรีสเกล
0 – 60	0.25
61 – 90	0.50
91 – 120	0.75
121 – 300	1.00

PAT2 – ความถนัดทางวิทยาศาสตร์

คะแนนในช่วง	ค่าทีรีสเกล
0 – 120	0.25
121 – 150	0.50
151 – 180	0.75
181 – 300	1.00

PAT3 – ความถนัดทางวิศวกรรมศาสตร์

คะแนนในช่วง	ค่าทีรีสเกล
0 – 120	0.25
121 – 150	0.50
151 – 180	0.75
181 – 300	1.00

จากอินพุตทั้ง 4 ตัวที่กล่าวมาในข้างต้น ค่าที่นำมารีสเกลของอินพุตแต่ละตัวจะมีความแตกต่างกัน ซึ่งขึ้นอยู่กับจำนวน ค่าเฉลี่ยและการกระจายตัวของข้อมูล ฯลฯ ที่ได้เก็บจากการประเมินจริงของนักศึกษาภาควิชาวิศวกรรมคอมพิวเตอร์ ซึ่งการรีสเกลนี้จะช่วยให้โครงข่ายประสาทเทียมสามารถแยกแยะข้อมูล และสามารถเรียนรู้ได้อย่างมีประสิทธิภาพมากขึ้น

อินพุตตัวที่ 5 หมายถึง วิธีการเข้าศึกษา เป็นวิธีที่นักศึกษาใช้เพื่อเข้ามาศึกษาต่อในภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ได้แก่

รับตรง	มีค่าเป็น	0.2
แอดมิชชั่น	มีค่าเป็น	0.4
โควต้าช้างเผือก	มีค่าเป็น	0.6

โควต้า สอวน. มีค่าเป็น 0.8
โควต้าบุคลากร มีค่าเป็น 1.0

อินพุตตัวที่ 6 - 9หมายถึง ผลลัพธ์ (ผลการเรียน) ที่เกิดขึ้น โดยกำหนดให้ค่าที่เป็น1 แทนผลลัพธ์ที่เกิดขึ้น และ 0 แทนค่าผลลัพธ์ที่เหลือที่ไม่ได้เกิดขึ้น จากข้อมูลเรคคอร์ดนั้นๆ

อินพุต 6 หมายถึง GPA ในช่วง 1.00 – 1.99
อินพุต 7 หมายถึง GPA ในช่วง 2.00 – 2.49
อินพุต 8 หมายถึง GPA ในช่วง 2.50 – 2.99
อินพุต 9 หมายถึง GPA ในช่วง 3.00 – 4.00

หลังจากพารามิเตอร์ทุกอย่างถูกกำหนดพร้อมแล้ว โปรแกรมจะทำการประมวลผลโดยการสร้างโครงข่ายประสาทเทียมโดยอินพุตเลเยอร์จะมี 5 โหนด สำหรับรับข้อมูลอินพุตจากไฟล์ชุดข้อมูล และสร้างฮิดเดนเลเยอร์โดยมีจำนวนโหนดเท่ากับพารามิเตอร์ที่ผู้พัฒนากำหนดไว้ เนื่องจากการปรับเปลี่ยนค่าของจำนวนของฮิดเดนโหนดนั้น ส่งผลต่อการประมวลผลลัพธ์ที่ได้ ที่แตกต่างกัน และสุดท้ายจะทำการสร้างเอาต์พุตเลเยอร์เพื่อประมวลผลในส่วนสุดท้ายสำหรับแสดงผลลัพธ์ที่เกิดขึ้น ซึ่งมี 4โหนด ซึ่งค่าที่ได้จากแต่ละโหนดของเอาต์พุตนั้น คือค่าสัมประสิทธิ์ของผลการเรียน สุดท้ายโปรแกรมจะทำการสร้างการเชื่อมโยงระหว่างเลเยอร์ โดยโหนดๆ หนึ่งนั้นจะทำการเชื่อมโยงไปยังโหนดทุกโหนดในเลเยอร์ถัดไป และการเชื่อมโยงนั้นจะทำการสุ่มค่าตัวเลขค่าหนึ่ง เพื่อใช้เป็นค่าน้ำหนักของอินพุตนั้นๆ

เมื่อโครงสร้างทั้งหมดของการประมวลผลมีครบแล้ว โปรแกรมจะทำการดึงค่าข้อมูลอินพุตจากชุดข้อมูล เพื่อเข้ามาประมวลผลโครงข่ายประสาทเทียมทีละเรคคอร์ดและคำนวณผลลัพธ์ที่ได้มาเทียบกับค่าผลลัพธ์ที่คาดหวังของข้อมูลเรคคอร์ดนั้นๆ จากนั้นจึงนำผลต่างที่ได้หรือค่าความผิดพลาดที่เกิดขึ้นมาคำนวณต่อ ในแนวคิดของการแพร่ย้อนกลับ (Back-propagation) โดยมีค่าอัตราการเรียนรู้ซึ่งกำหนดโดยผู้พัฒนาเป็นส่วนประกอบในการคำนวณ เพื่อหาค่าเพื่อนำไปแก้ไขน้ำหนัก ส่งผลให้ค่าผลต่างของผลลัพธ์ที่คำนวณได้กับผลลัพธ์ที่คาดหวังเข้าใกล้ศูนย์มากที่สุด กระบวนการทำงานนี้จะดึงเรคคอร์ดของข้อมูลมาประมวลผลจนครบ ซึ่งนับเป็น 1 รอบของการฝึกฝน และจะวนซ้ำเช่นนี้เรื่อยๆ จนครบตามจำนวนรอบสูงสุดที่ผู้พัฒนากำหนดไว้

3.5.3 ส่วนของการตรวจสอบผลลัพธ์

ทุกๆ รอบที่ทำการฝึกฝนนั้นจะทำให้ค่าน้ำหนักของอินพุตที่มีการเชื่อมต่อกันระหว่างโหนดมีการเปลี่ยนแปลงไปเรื่อยๆ เพื่อให้ได้ผลลัพธ์ที่คำนวณได้มีค่าเข้าใกล้ผลลัพธ์ที่คาดหวังมากที่สุดต่อ

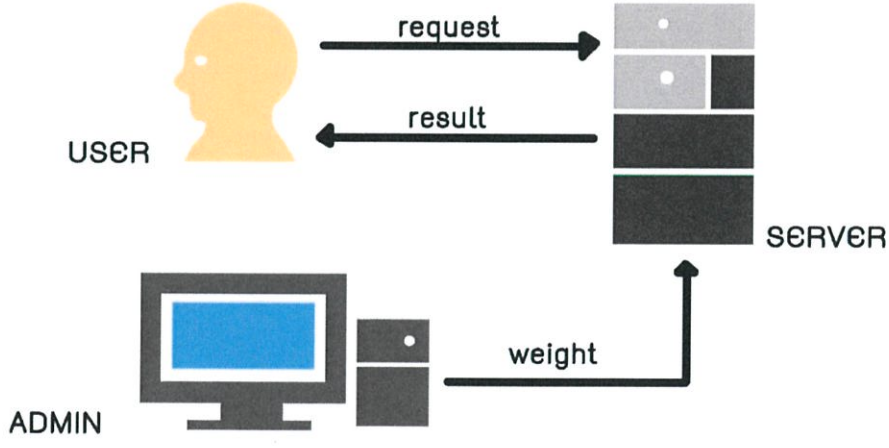
ทุกๆ เรคคอร์ดของข้อมูล ซึ่งในจุดนี้ผู้พัฒนาสามารถกำหนดคาบของการทดสอบโปรแกรม เพื่อนำข้อมูล Test set ซึ่งเป็นชุดข้อมูลใหม่ที่โปรแกรมไม่เคยใช้ฝึกฝนมาก่อน มาทดสอบเพื่อคิดค่าผลลัพธ์ด้วยแนวคิด Winner take all และคำนวณหาค่าอัตราของความผิดพลาด เพื่อนำค่าไปพลอตกราฟความสัมพันธ์ระหว่างอัตราความผิดพลาดกับจำนวนรอบที่ฝึกฝน

ก่อนที่จะนำค่าน้ำหนักที่ได้จากการฝึกฝนโดยโครงข่ายประสาทเทียมไปใช้กับโปรแกรมนั้น ผู้พัฒนาจะต้องทำการตรวจสอบผลลัพธ์ก่อนว่า ค่าน้ำหนักที่ได้มีความเหมาะสมจะนำไปใช้กับโปรแกรมหรือไม่ โดยในส่วนนี้ จะเกิดการตรวจสอบผลลัพธ์ที่เกิดขึ้น โดยใช้ตัวชี้วัดจากพารามิเตอร์ทั้งหมดที่ผู้พัฒนาเป็นคนกำหนด เพื่อทดสอบและคำนวณเพื่อหาจำนวนรอบที่ทำให้เกิดค่าอัตราความผิดพลาดที่ต่ำที่สุดที่เป็นไปได้ ในอีกความหมายหนึ่งคือ ค่าน้ำหนัก ณ จำนวนรอบที่โครงข่ายประสาทเทียมทำการฝึกฝนนั้น เป็นค่าน้ำหนักที่ดีที่สุด และเหมาะสมที่สุดสำหรับการอัพโหลดไปยังเว็บแอพพลิเคชั่น เพื่อให้ผู้ใช้งานได้ใช้งานจริง

3.6 การออกแบบระบบ

3.6.1 System Architecture Diagram

จากรูปที่ 3.22 แสดงให้เห็นถึง โครงสร้างของระบบ ซึ่งประกอบด้วย ผู้ใช้งาน เซิร์ฟเวอร์ และผู้พัฒนา และการส่งข้อมูลของระหว่างกัน



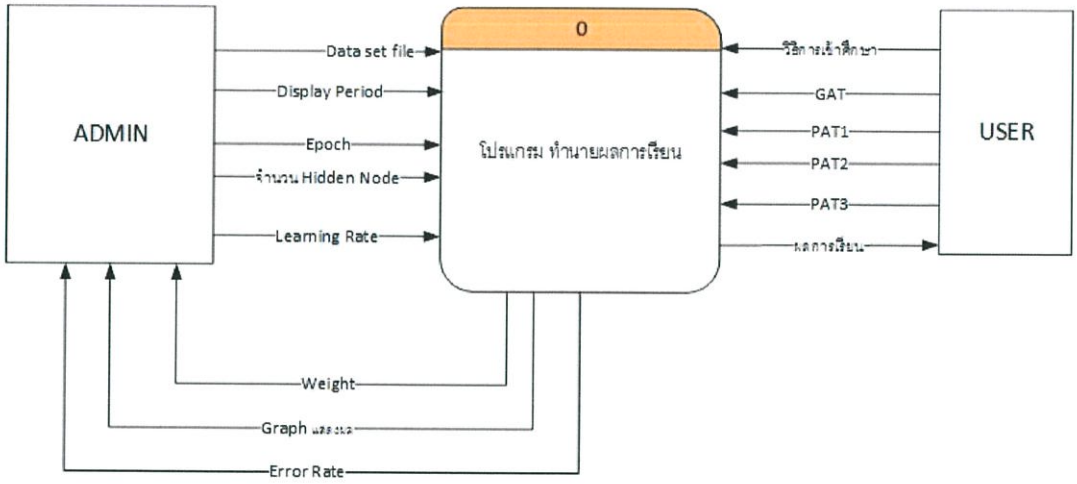
รูปที่ 3.22 โครงสร้างของระบบ

3.6.2 แผนภาพแสดงการไหลของข้อมูล (Data Flow Diagram)

แบ่งแผนภาพแสดงการไหลของข้อมูลเป็นสองส่วนหลักๆ คือ ส่วนของตัวโปรแกรม และส่วนของการเก็บข้อมูลจากนักศึกษา

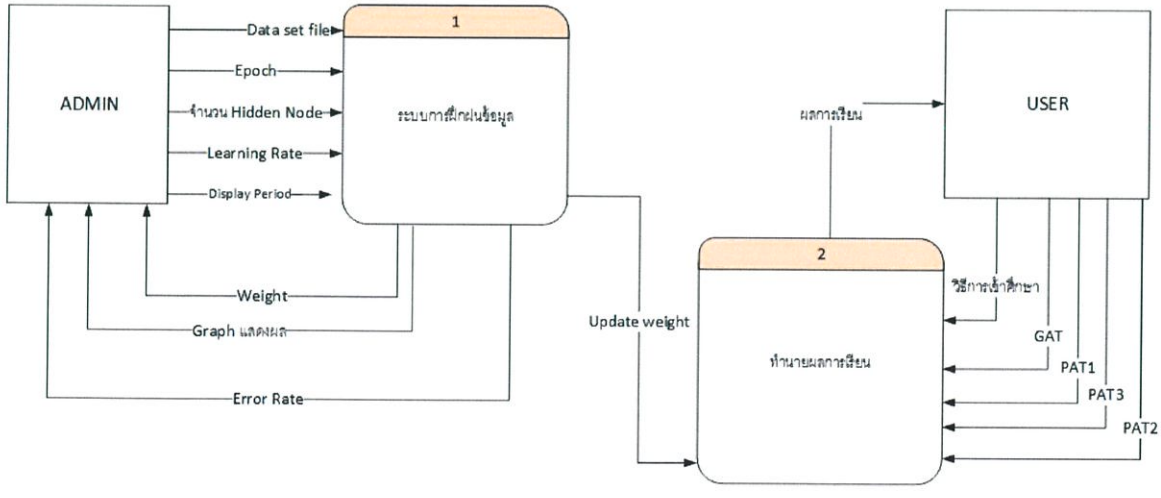
3.6.2.1 Data Flow Diagram ส่วนของโปรแกรม

จากรูปที่ 3.23แสดงให้เห็นถึงไดอะแกรมของการไหลเวียนของแลกเปลี่ยนข้อมูลในภาพรวมระหว่างผู้ใช้งานและผู้พัฒนากับโปรแกรมทำนายผลการเรียน



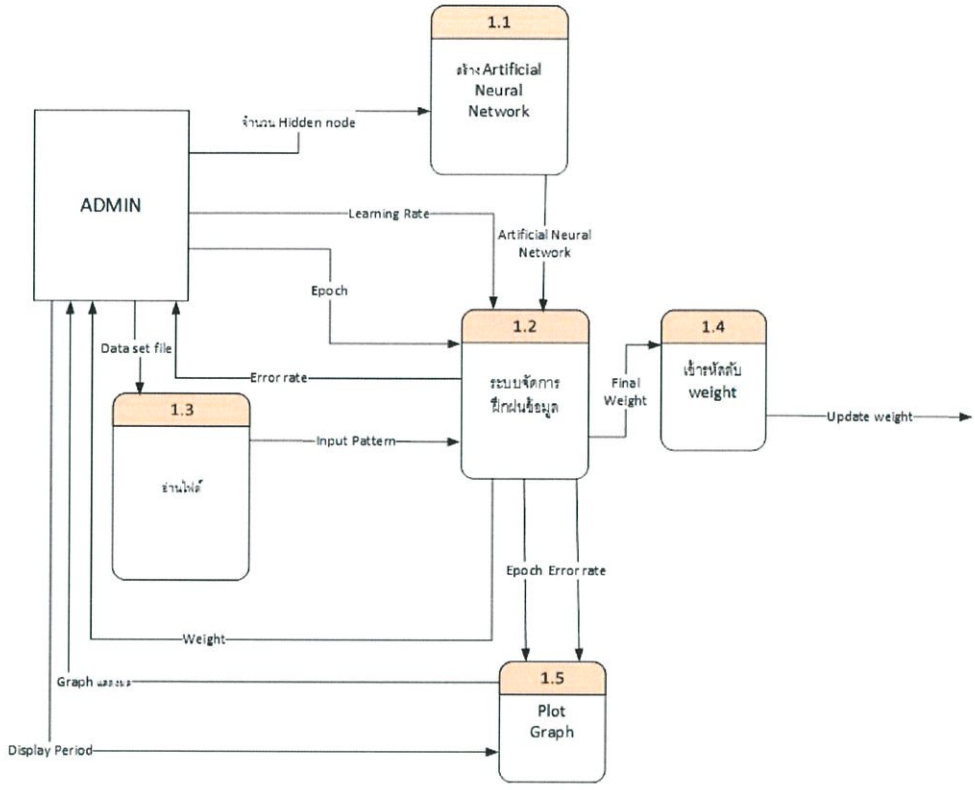
รูปที่ 3.23 Data Flow Diagram ของโปรแกรมทำนายผลการเรียน

จากรูปที่ 3.24 แสดงภาพไดอะแกรมถึงการทำงานโดยรวมของโปรแกรม ซึ่งแบ่งโปรแกรมออกเป็น 2 ส่วนหลักคือ ส่วนของการทำนายผลการเรียน และการฝึกฝนข้อมูล



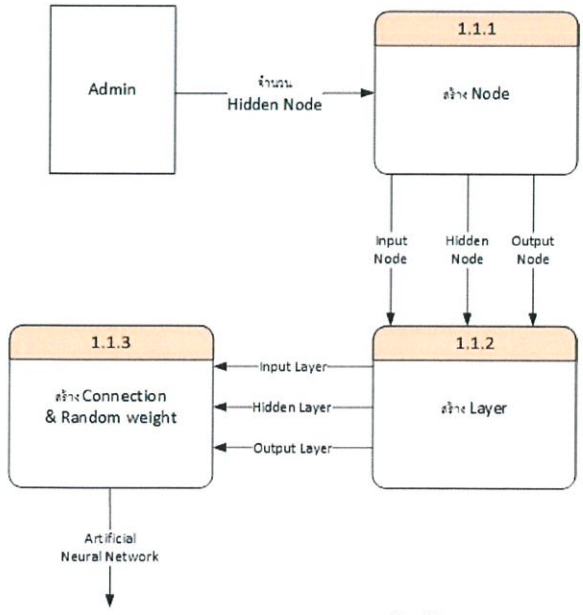
รูปที่ 3.24 Data Flow Diagram Level 0 ของโปรแกรมทำนายผลการเรียน

จากรูปที่ 3.25 เป็นส่วนขยายและอธิบายถึงระบบการฝึกฝนข้อมูลซึ่งเป็นส่วนประมวลผลกลางที่ใช้ในการฝึกฝน เพื่อคำนวณหาค่าน้ำหนักที่เหมาะสมต่อการใช้ทำนายผลการเรียน



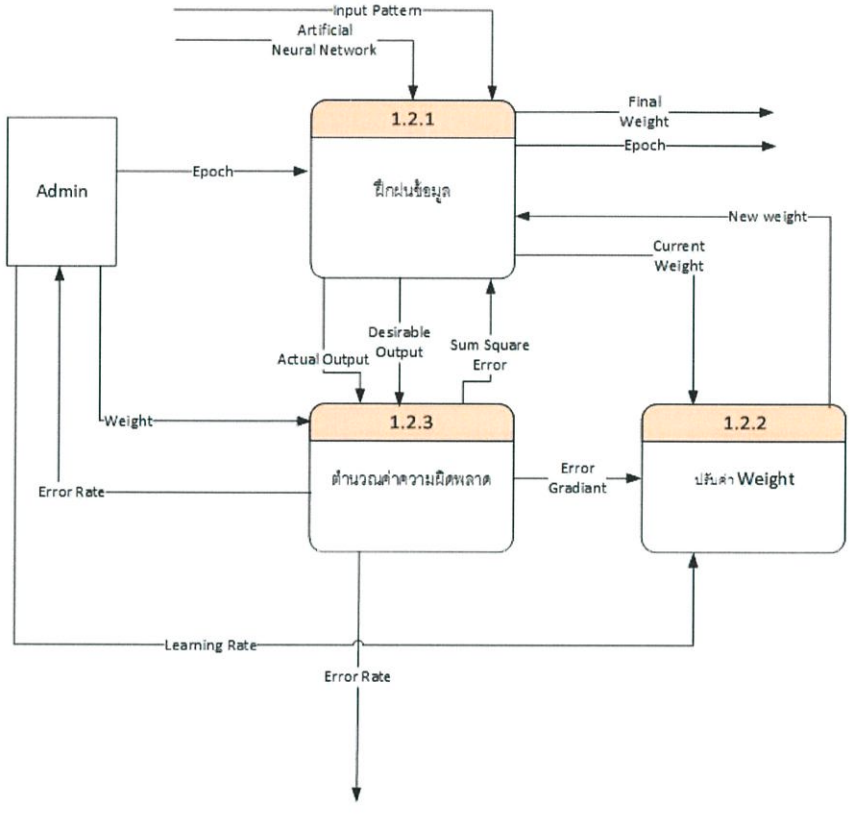
รูปที่ 3.25 Data Flow Diagram ระบบการฝึกฝนข้อมูล

จากรูปที่ 3.26 เป็นส่วนที่อธิบายถึงกระบวนการสร้างโครงข่ายประสาทเทียม โดยมีองค์ประกอบหลักๆ ได้แก่ โหนด เพื่อนำมาประกอบเป็นเลเยอร์ และสร้างการเชื่อมต่อระหว่างกันโดยมีค่าน้ำหนักสำหรับใช้เพื่อประมวลผลทำนายต่อไป



รูปที่ 3.26 Data Flow Diagram ของการสร้างโครงข่ายประสาทเทียม

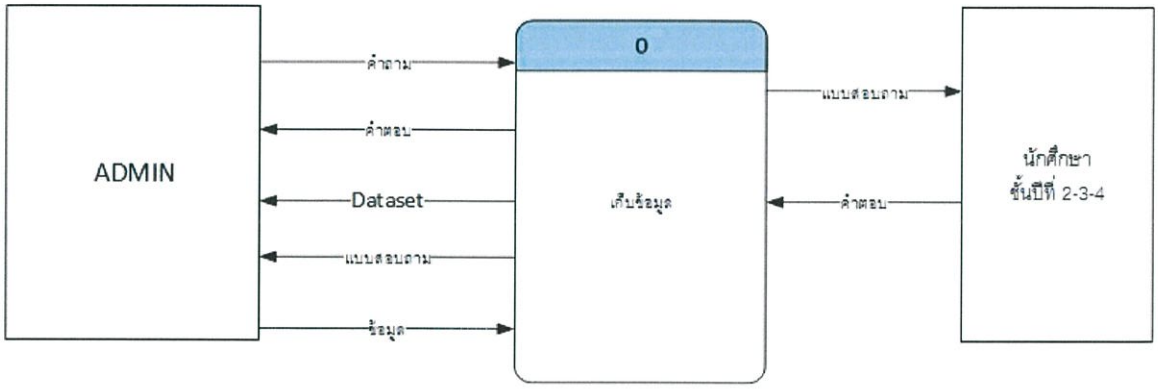
จากรูปที่ 3.27 เป็นส่วนที่อธิบายถึงระบบจัดการฝึกฝนข้อมูล และกระบวนการในการคำนวณค่าความผิดพลาด เพื่อนำข้อมูลไปใช้ในการปรับค่าน้ำหนัก และส่งค่าน้ำหนักไปยังส่วนของการทำนายผลการเรียน



รูปที่ 3.27 Data Flow Diagramของระบบจัดการฝึกฝนข้อมูล

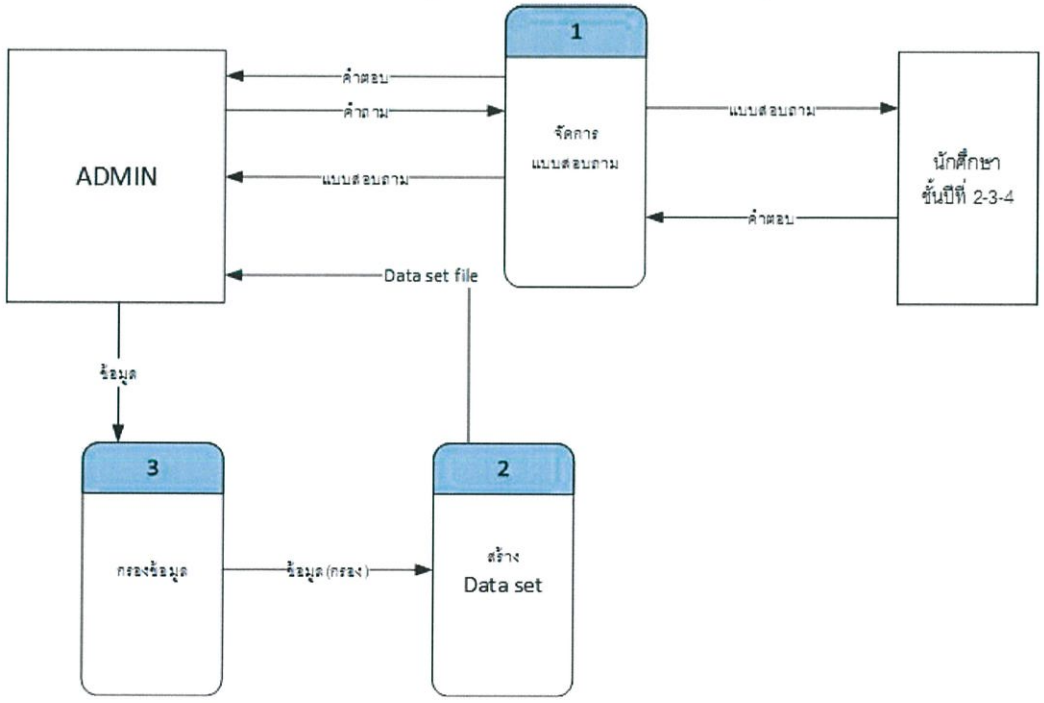
3.6.2.2 Data Flow Diagram ส่วนของการเก็บข้อมูล

จากรูปที่ 3.28 แสดงให้เห็นถึงไดอะแกรมของการไหลเวียนของการแลกเปลี่ยนข้อมูลในภาพรวม ระหว่างผู้พัฒนาและนักศึกษาชั้นปีที่ 2, 3 และ 4 ในส่วนของการเก็บข้อมูลตัวอย่างเพื่อนำไปใช้ฝึกฝนโครงข่ายประสาทเทียม



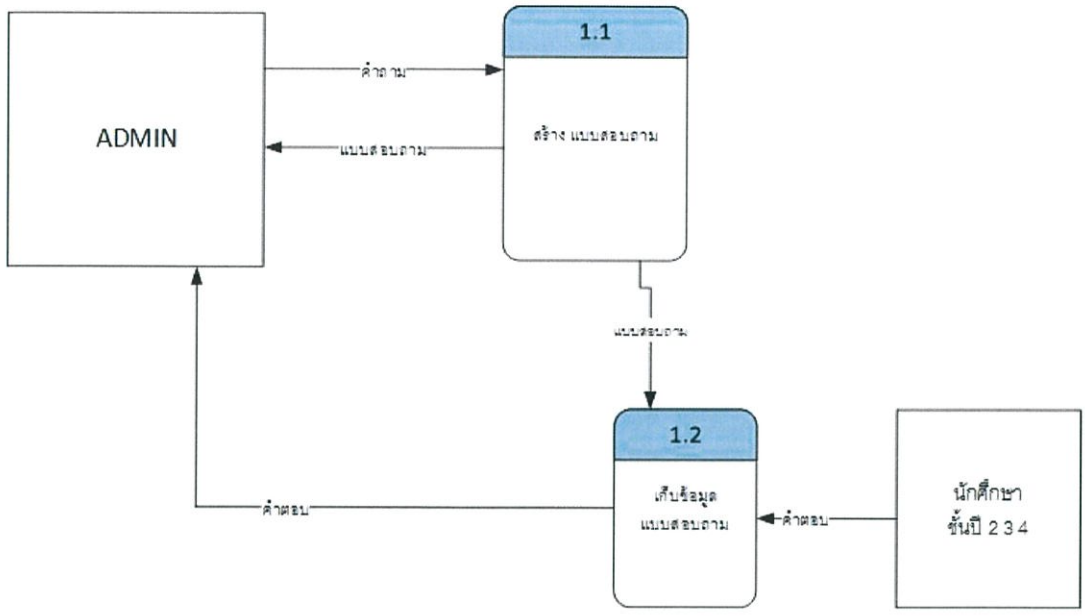
รูปที่ 3.28 Data Flow Diagram ของการดำเนินงานเก็บข้อมูล

จากรูปที่ 3.29 แสดงภาพไดอะแกรมถึงการทำงานโดยรวมของการเก็บข้อมูล โดยมี 3 ส่วนหลักๆ คือ จัดการแบบสอบถาม กรองข้อมูล และสร้างชุดข้อมูล ของข้อมูลที่ผ่านการกรองแล้ว



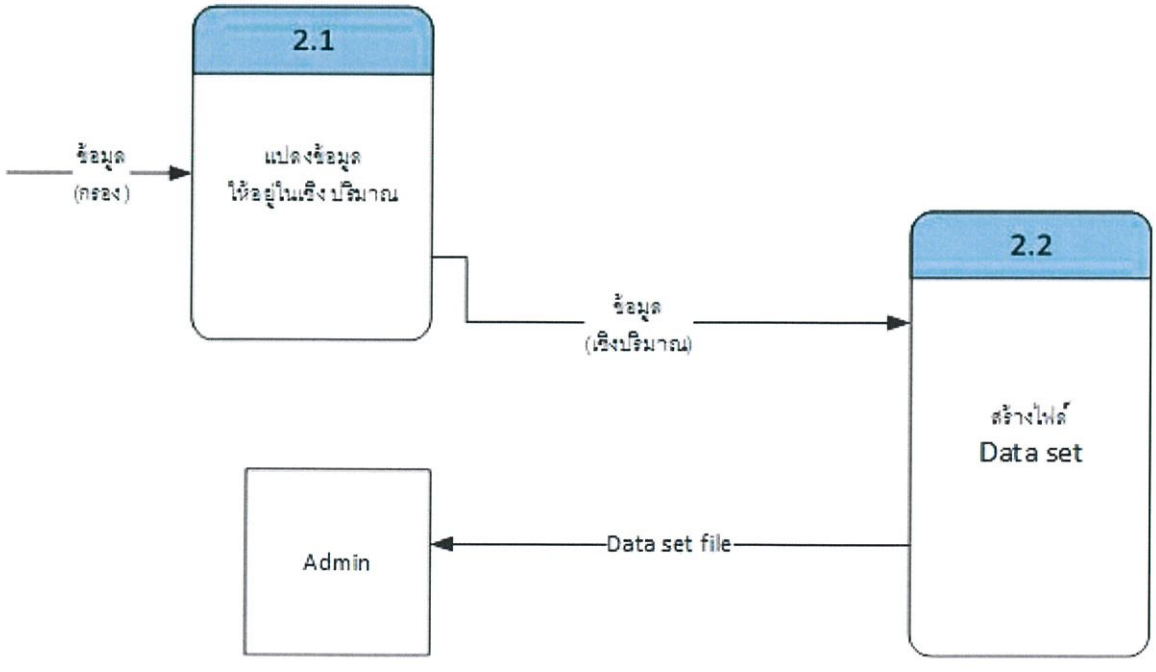
รูปที่ 3.29 Data Flow Diagram level 0 ของการดำเนินงานเก็บข้อมูล

จากรูปที่ 3.30 เป็นส่วนที่แสดงถึงรายละเอียดในส่วนของการจัดการแบบสอบถาม ซึ่งประกอบด้วย 2 ส่วนการทำงานหลักคือ สร้างแบบสอบถาม เพื่อใช้ในการเก็บข้อมูลจากนักศึกษาชั้นปีที่ 2, 3 และ 4



รูปที่ 3.30 Data Flow Diagram ของการจัดการแบบสอบถาม

จากรูปที่ 3.31 เป็นส่วนที่อธิบายถึงกระบวนการหลังจากที่ได้รับข้อมูลจากการเก็บสอบถามมาแล้ว และนำมาผ่านกระบวนการแปลงข้อมูล ให้อยู่ในรูปของเชิงปริมาณ รวมถึงการรีสเกล ก่อนจะนำไปสร้างไฟล์ชุดข้อมูล ซึ่งอยู่ในรูปของไฟล์ Text



รูปที่ 3.31 Data Flow Diagram ของการสร้างชุดข้อมูล

3.6.3 แผนภาพ UML (Unified Modeling Language Diagram)

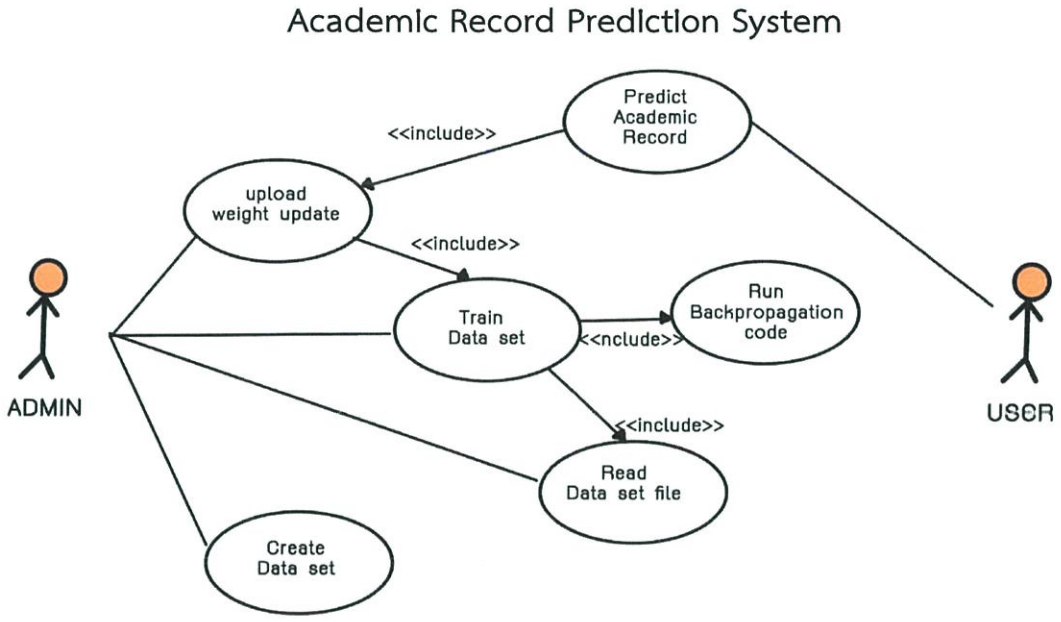
3.6.3.1 Use-Case Diagram

ในส่วนการใช้งานโปรแกรมของผู้พัฒนา ประกอบด้วย 4 ส่วนหลักๆ ได้แก่

1) Create Data set เป็นส่วนที่ผู้พัฒนาต้องนำข้อมูลที่ได้จากการเก็บสอบถามมาจัดรูปแบบและอยู่ในรูปของไฟล์ Text เพื่อใช้ป้อนเข้าโปรแกรมต่อไป
2) Read Data set file เป็นส่วนที่ให้ผู้พัฒนาป้อนไฟล์ชุดข้อมูล เข้าไปยังโปรแกรม เพื่ออ่านไฟล์เข้าไปยังโครงข่ายประสาทเทียมเพื่อใช้ในการฝึกฝนข้อมูล
3) Train Data set เป็นส่วนที่ให้ผู้ใช้กำหนดค่าพารามิเตอร์ต่างๆ สำหรับใช้ในการฝึกฝนของโครงข่ายประสาทเทียม
4) Upload weight update เป็นส่วนที่ผู้พัฒนา สามารถ export ค่าน้ำออกมาเพื่อดูหรือบันทึกเป็นไฟล์เก็บไว้ได้ และสามารถอัพโหลดค่าน้ำหนักดังกล่าวขึ้นไปที่เว็บแอพพลิเคชั่นผ่านโปรแกรมนี้ได้ทันที

ในส่วนของผู้ใช้งานนั้น จะมีเพียงส่วนหลักๆ เพียงส่วนเดียวคือ Predict Academic Record สำหรับให้ผู้ใช้ ป้อนข้อมูลอินพุต ของคะแนน และวิธีการเข้าศึกษา เพื่อทำนายผลการเรียน

นอกจากนี้ระบบที่อยู่นอกเหนือไปจากระบบหลักๆ ที่มีการปฏิสัมพันธ์กับผู้พัฒนาโดยตรงดังกล่าว จะมีการเรียกใช้งาน (include) จากระบบอื่นๆ คือ Run Back propagation code ซึ่งจะถูกเรียกใช้งานเมื่อมีการเรียกใช้ระบบในส่วน Train Data set โดยผู้พัฒนา จะมีการเรียนใช้งาน Read Data set file เพื่ออ่านเซตของเรคคอร์ดทั้งหมดทั้งหมด และ Back propagation code เพื่อสร้างส่วนประมวลผลของโครงข่ายประสาทเทียมสำหรับการฝึกฝน โดยมีพารามิเตอร์จากผู้พัฒนาเป็นองค์ประกอบการฝึกฝนโดยรูปที่ 3.32 แสดงให้เห็นถึงภาพรวมของการใช้งานโปรแกรมทำนายผลการเรียน

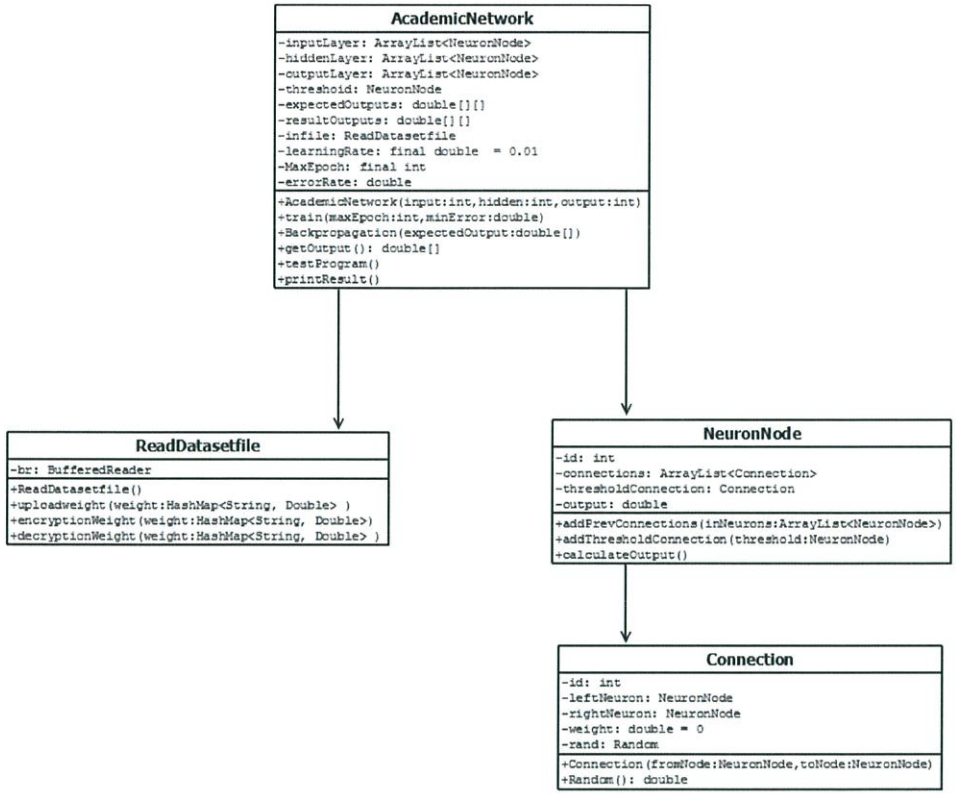


รูปที่ 3.32 แผนภาพ Use-Case Diagram ของโปรแกรมทำนายผลการเรียน

3.6.3.2 Class Diagram

รูปที่ 3.33 อธิบายถึงการทำงานของซอฟต์แวร์สำหรับผู้ใช้งาน ในรูปแบบของแผนภาพคลาสไดอะแกรมโดยมีรายละเอียดต่างๆ ดังนี้

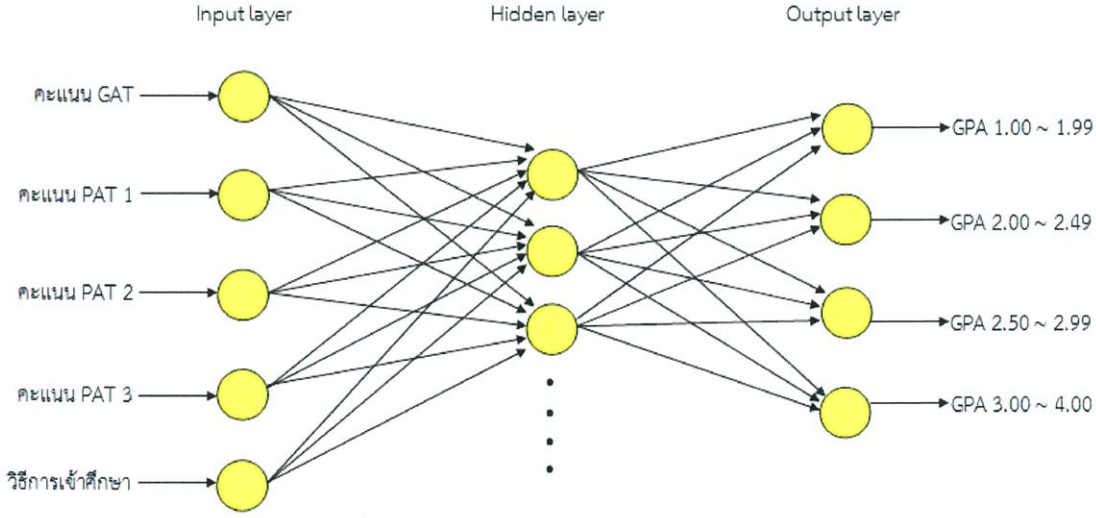
1) Class AcademicNetwork เป็นส่วนทำงานหลักของโปรแกรม โดยมีฟังก์ชั่นทำงานหลักคือฟังก์ชั่น train และ Backpropagation ในการประมวลผลลัพธ์เพื่อปรับค่าน้ำหนัก
2) ClassNeuronNode เป็นส่วนที่ทำงานเพื่อเก็บรายละเอียดของออปเจคนิวรอนซึ่งเก็บเป็นไอดีประจำนิวรอนและการเชื่อมต่อต่างๆ ที่เชื่อมกับนิวรอนในเลเยอร์ก่อนหน้า
3) Class Connection เป็นส่วนของออปเจคของการเชื่อมต่อ จะเก็บข้อมูลของคู่นิวรอนที่เชื่อมต่อกันโดยการเชื่อมต่อนั้นๆ และฟังก์ชั่นในการสุ่มค่าน้ำหนักเมื่อเริ่มมีการสร้างการเชื่อมต่อ
4) Class ReadDatasetfile เป็นส่วนที่ใช้ในการอ่านไฟล์ชุดข้อมูลจากผู้พัฒนาเพื่อนำมาใช้ฝึกฝนข้อมูลต่อไป



รูปที่ 3.33 แผนภาพ Class Diagram ของโปรแกรมทำนายผลการเรียน

3.6.4 Artificial Neural Network Diagram

ในการออกแบบระบบประมวลผล โครงข่ายประสาทเทียม ได้ออกแบบระบบดังนี้



รูปที่ 3.34 Neural Network Diagram

3.7 ขอบเขตและข้อจำกัดของซอฟต์แวร์ที่พัฒนา

1) ซอฟต์แวร์ทำงานในลักษณะของเว็บแอพพลิเคชั่นสำหรับผู้ใช้งานทั่วไป
2) ซอฟต์แวร์สามารถทำนายได้เฉพาะระดับผลการเรียนของการศึกษาในภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังชั้นปีที่ 1 เท่านั้น ไม่สามารถทำนายของชั้นปีอื่นๆได้
3) ซอฟต์แวร์ทำงานเป็นลักษณะของเดสก์ทอปแอพพลิเคชั่นสำหรับผู้พัฒนา ซึ่งเป็นโปรแกรมประยุกต์ เพื่อใช้ในการฝึกฝนจากข้อมูลของโครงงานนี้เท่านั้น
4) การใช้ชุดข้อมูลสำหรับฝึกฝน ต้องเป็นรูปแบบที่ระบบระบุไว้เท่านั้น
5) ซอฟต์แวร์สำหรับผู้พัฒนาจะหยุดทำการฝึกฝนเมื่อครบรอบของการฝึกฝนที่ผู้พัฒนากำหนด

บทที่ 4
การทดลองและผลการทดลอง

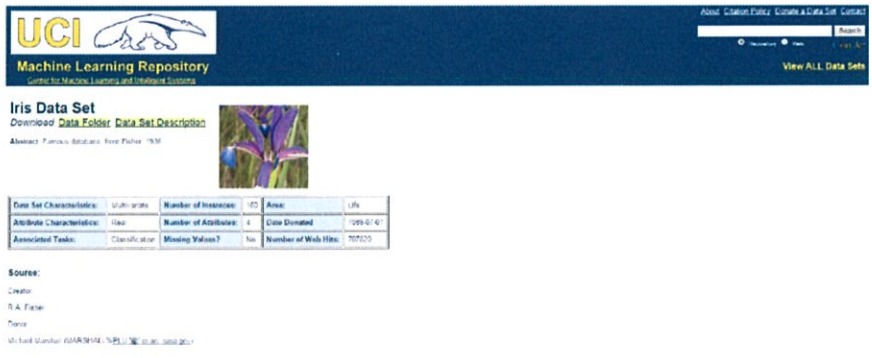
ในบทนี้จะกล่าวถึงการทดสอบซอฟต์แวร์ รายละเอียดการทดสอบและการจำลองการทำงาน

4.1 การทดลองการทำงานของโครงข่ายประสาทเทียม

โครงข่ายประสาทเทียมเป็นส่วนสำคัญของซอฟต์แวร์ เนื่องจากเป็นส่วนที่สำคัญส่วนหนึ่งของโครงงานคือการทำนายผลการเรียน ความถูกต้องของการทำนายผลของโครงข่ายประสาทเทียมจึงจำเป็นต้องทดสอบความถูกต้อง ว่าการทำงานนั้นเป็นไปอย่างถูกต้องหรือไม่

4.1.1 การทดลองโครงข่ายประสาทเทียมด้วยชุดข้อมูล Iris

ชุดข้อมูล Iris นั้นเป็นชุดข้อมูลที่มีการใช้งานอย่างแพร่หลาย จึงสามารถนำมาเป็นชุดข้อมูลสำหรับทดสอบความถูกต้องของซอฟต์แวร์ส่วนโครงข่ายประสาทเทียมได้ เนื่องจากสามารถนำไปอ้างอิงกับการทดลองของผู้อื่นที่เป็นที่ยอมรับแล้วว่า ถูกต้องได้ ในการทดสอบนั้นจะทำการดาวน์โหลดชุดข้อมูลจากแหล่งที่เชื่อถือได้ โดยในการทดสอบครั้งนี้นำชุดข้อมูล Iris จาก https://archive.ics.uci.edu/ml/datasets/Iris ดังรูปที่ 4.1



รูปที่ 4.1 แหล่งข้อมูลของชุดข้อมูล Iris

เมื่อทำการดาวน์โหลดมาแล้วนั้น จากชุดข้อมูลจำนวน 150 เรคคอร์ด จะทำการแบ่งจำนวนเรคคอร์ดสำหรับการฝึกฝนโครงข่ายประสาทเทียม เป็น Training set จำนวน 75 เรคคอร์ด, Validation set จำนวน 37 เรคคอร์ด และ Test set จำนวน 38 เรคคอร์ด เมื่อแบ่งชุดข้อมูลเรียบร้อยแล้วจะทำการเลือก ชุดข้อมูล Training set และ Validation set โดยตั้งค่าส่วนของ อินพุต

โหนด จำนวน 4 โหนด ส่วนเอาท์พุต 3 โหนด โดยทำการวนรอบจำนวน 10,000 รอบ เพื่อหาส่วนที่มีค่าความผิดพลาดน้อยที่สุด ดังรูปที่ 4.2

```
##### EPOCH 9800
ERROR RATE (train set) : 5.3333%
ERROR RATE (validate set) : .000%

##### EPOCH 9900
ERROR RATE (train set) : 5.3333%
ERROR RATE (validate set) : .000%

##### EPOCH 10000
ERROR RATE (train set) : 5.3333%
ERROR RATE (validate set) : .000%
ERROR RATE (test set) : 2.6316%
```

รูปที่ 4.2 ภาพการฝึกฝนของโครงข่ายประสาทเทียม บนโปรแกรมเนตบีนส์

เมื่อทำการฝึกฝนโครงข่ายประสาทเทียมครบตามจำนวนรอบแล้วนั้น จะทำการทดสอบโปรแกรมโดยใช้ Test set ในการทดสอบ เมื่อทดสอบแล้วนั้นพบว่าค่าความผิดพลาดมีค่าเป็น 2.63 % ซึ่งถือว่าเป็นค่าที่ยอมรับได้ เนื่องจากเมื่อเทียบกับโปรแกรมโครงข่ายประสาทเทียมอื่น มีค่าความผิดพลาดใกล้เคียงกัน จึงสรุปได้ว่า ส่วนการประมวลผลโครงข่ายประสาทเทียม ทำงานได้ตามปกติ และ สามารถนำไปใช้งานในโครงงานได้

4.1.2 การทดลองชุดข้อมูลที่ใช้ในโครงงาน

เมื่อทำการเก็บข้อมูลจากแบบสอบถามแล้วนั้น จะทำการทดสอบดูว่าชุดข้อมูลนั้นสามารถนำไปใช้งานได้เลยหรือไม่ โดยนำชุดข้อมูลเข้าฝึกฝนโครงข่ายประสาทเทียม แล้วตรวจสอบค่าความผิดพลาดที่เกิดขึ้น

4.1.2.1 การทดลองชุดข้อมูลเบื้องต้น

ในส่วนนี้จะทำการใส่ชุดข้อมูลที่ยังไม่ผ่านการแปลงข้อมูลเข้าฝึกฝนโครงข่ายประสาทเทียม โดยทำการบันทึกข้อมูลจากแบบสอบถามไว้ในโปรแกรม Excel ดังรูปที่ 4.3 จากนั้นบันทึกข้อมูลอยู่ในลักษณะของ Text file แยกแต่ละคอลัมน์ด้วย ช่องแท็บ (Tab) เมื่อทำการทดสอบกับโครงข่ายประสาทเทียม พบว่ามีค่าเปอร์เซ็นต์ความผิดพลาดอยู่ที่ประมาณ 50 %

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
ลำดับ	รับตรง	อดมิชชั่	ตำช้างเผ	วตำสอ	วตำบุคล		GPA					GAT	PAT1	PAT2	PAT3
							1.00-1.99	2.00 - 2.49	2.50 - 2.99	3.00 - 4.00					
1	1					0.2				1	0.8	0.9	0.3	0.6	0.8
2		1				0.4			1		0.6	0.8	0.3	0.5	0.6
3			1			0.6		1			0.4	0.8	0.4	0.4	0.3
4			1			0.6			1		0.6	0.7	0.4	0.3	0.3
5		1				0.4				1	0.8	0.9	0.5	0.4	0.5
6			1			0.6				1	0.8	0.6	0.3	0.5	0.5
7	1					0.2				1	0.8	0.9	0.4	0.6	0.8
8	1					0.2			1		0.6	0.7	0.3	0.4	0.5
9	1					0.2				1	0.8	0.9	0.5	0.6	0.6
10	1					0.2			1		0.6	0.8	0.3	0.4	0.4
11	1					0.2			1		0.6	0.9	0.2	0.2	0.2
12		1				0.4			1		0.6	0.8	0.2	0.5	0.5
13	1					0.2	1				0.2	0.8	0.4	0.5	0.7
14			1			0.6		1			0.4	0.9	0.3	0.5	0.6
15			1			0.6			1		0.6	0.9	0.2	0.4	0.4
16	1					0.2				1	0.8	0.9	0.2	0.4	0.5
17		1				0.4				1	0.8	0.9	0.5	0.7	0.8
18	1					0.2			1		0.6	0.7	0.4	0.6	0.7

รูปที่ 4.3 การเก็บชุดข้อมูลบนโปรแกรม Excel

4.1.2.2 การทดลองชุดข้อมูลจากการตัดAttribute ของข้อมูล

ในการทดสอบที่ 4.1.2.1 จะพบว่าค่าความผิดพลาดของข้อมูลยังมีมาก ซึ่งอยู่ที่ประมาณ 50% จึงจำเป็นต้องหาวิธีที่ทำให้ค่าความผิดพลาดของข้อมูลลดน้อยลง เนื่องจากส่วนของโครงข่ายประสาทเทียมได้ทดสอบแล้วว่าไม่มีความผิดพลาด จากการทดลองที่ผ่านมา จึงทำการทดลองปรับเปลี่ยนที่ชุดข้อมูลแทน โดยทดสอบการตัด Attribute ของข้อมูลส่วนต่าง ๆ โดยแบ่งเป็น

1.) การทดสอบตัดข้อมูลคะแนน GAT

พบว่ามีค่าเปอร์เซ็นต์ความผิดพลาดอยู่ที่ 48.78 %

2.) การทดสอบตัดข้อมูลคะแนน PAT1

พบว่ามีค่าเปอร์เซ็นต์ความผิดพลาดอยู่ที่ 51.22 %

3.) การทดสอบตัดข้อมูลคะแนน PAT2

พบว่ามีค่าเปอร์เซ็นต์ความผิดพลาดอยู่ที่ 50.00 %

4.) การทดสอบตัดข้อมูลคะแนน PAT3

พบว่ามีค่าเปอร์เซ็นต์ความผิดพลาดอยู่ที่ 53.54 %

จากการทดสอบนี้จะพบว่าค่าเปอร์เซ็นต์ความผิดพลาดยังมีมากอยู่จึงจำเป็นต้องทำการทำลองอื่นต่อไปเพื่อหาชุดข้อมูลที่จะทำให้ได้ค่าเปอร์เซ็นต์ความผิดพลาดที่น้อยที่สุด

4.1.2.3 การทดลองชุดข้อมูลด้วยวิธีการ Rescale ข้อมูล

วิธีการเตรียมชุดข้อมูลแบบหนึ่งคือการ รีสเกล เป็นการปรับข้อมูลให้อยู่ในช่วงที่ทำให้โครงข่ายประสาทเทียมสามารถฝึกฝน และทำการทำนายได้ดีขึ้น โดยช่วงข้อมูลที่ทำการรีสเกล เป็นดังที่แสดงไว้ในบทที่ 3 หัวข้อที่ 3.5.2 คือ

GAT – ความถนัดทั่วไป ประกอบด้วย การอ่าน เขียน คิด วิเคราะห์ และภาษาอังกฤษ

คะแนนในช่วง	ค่าที่รีสเกล
0 – 60	0.25
61 – 90	0.50
91 – 120	0.75
121 – 300	1.00

PAT1 – ความถนัดทางคณิตศาสตร์

คะแนนในช่วง	ค่าที่รีสเกล
0 – 60	0.25
61 – 90	0.50
91 – 120	0.75
121 – 300	1.00

PAT2 – ความถนัดทางวิทยาศาสตร์

คะแนนในช่วง	ค่าที่รีสเกล
0 – 120	0.25
121 – 150	0.50
151 – 180	0.75
181 – 300	1.00

PAT3 – ความถนัดทางวิศวกรรมศาสตร์

คะแนนในช่วง	ค่าทีรีสเกล
0 – 120	0.25
121 – 150	0.50
151 – 180	0.75
181 – 300	1.00

เมื่อทำการรีสเกลแล้ว พบว่าค่าความผิดพลาดอยู่ในระดับที่ยอมรับได้ มีค่าเปอร์เซ็นต์ความผิดพลาดอยู่ที่ 36% สามารถนำชุดข้อมูลนี้ไปใช้ในโครงงานได้ต่อไป

4.1.3 การทดลองการตั้งค่าในโครงข่ายประสาทเทียม

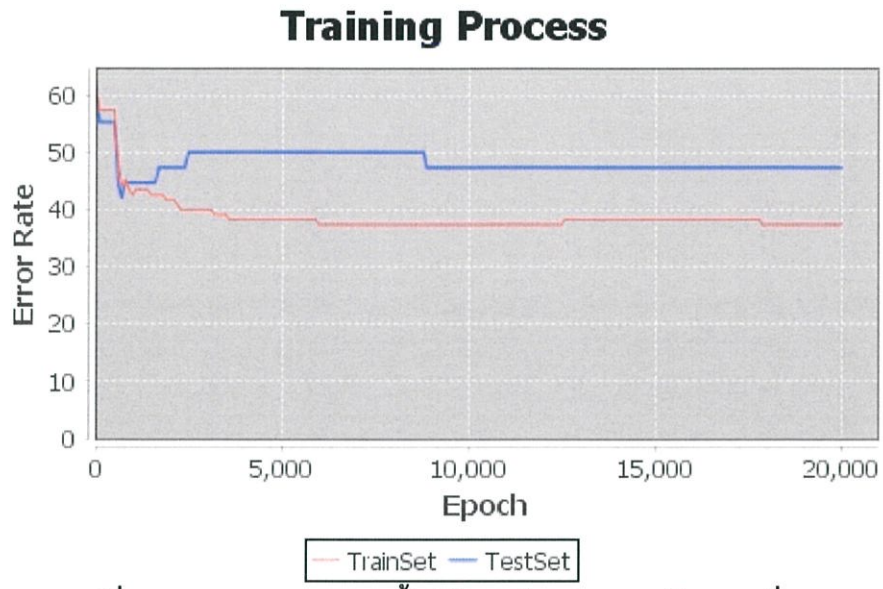
ในซอฟต์แวร์ส่วนของการฝึกฝนโครงข่ายประสาทเทียมนั้น จะมีบางตัวแปรที่เปลี่ยนแปลงแล้วมีผลกระทบต่อการเรียนรู้ของโครงข่ายประสาทเทียม ซึ่งได้แก่ จำนวนฮิดเดนโหนด และค่าอัตราการเรียนรู้ การทดลองนี้จึงเป็นการทดลอง เพื่อหาค่าของตัวแปรทั้ง 2 นี้ที่ทำให้เกิดประสิทธิภาพต่อความแม่นยำของการทำนายมากที่สุด

4.1.3.1 การทดลองโดยการเปลี่ยนแปลงค่าฮิดเดนโหนด

ในการทดลองนี้จะใช้จำนวน ฮิดเดนโหนด ได้แก่จำนวน 2, 4, 6, 8 และ 10 โหนด โดยกำหนดให้มีจำนวนรอบสูงสุดของการฝึกฝนที่ 20000 รอบ อัตราการเรียนรู้ 0.01 และคาบของการแสดงผลเป็น 100 รอบ แล้วทำการดูค่าอัตราความผิดพลาดโดยรวมที่เกิดขึ้น โดยการสังเกตจากกราฟแสดงผลเพื่อดูแนวโน้มของกราฟและจุด ณ ตำแหน่งที่เกิดค่าเปอร์เซ็นต์ความผิดพลาดต่ำที่สุด โดยการตั้งค่าแต่ละครั้ง มีผลการทดลองดังนี้

1) จำนวนฮิดเดนโหนด 2 โหนด

มีเปอร์เซ็นต์ความผิดพลาดอยู่ที่ 42.11 %



รูปที่ 4.4 ผลการทดลองการตั้งค่าโครงข่ายประสาทเทียม ชุดที่ 1

2) จำนวนฮิดเดนโหนด 4 โหนด

มีเปอร์เซ็นต์ความผิดพลาดอยู่ที่ 42.11 %

รูปที่ 4.5 ผลการทดลองการตั้งค่าโครงข่ายประสาทเทียม ชุดที่ 2

3) จำนวนฮิดเดนโหนด 6 โหนด

มีเปอร์เซ็นต์ความผิดพลาดอยู่ที่ 36.84 %

รูปที่ 4.6 ผลการทดลองการตั้งค่าโครงข่ายประสาทเทียม ชุดที่ 3

4) จำนวนฮิดเดนโหนด 8 โหนด

มีเปอร์เซ็นต์ความผิดพลาดอยู่ที่ 36.84 %

รูปที่ 4.7 ผลการทดลองการตั้งค่าโครงข่ายประสาทเทียม ชุดที่ 4

5) จำนวนฮิดเดนโหนด 10 โหนด

มีเปอร์เซ็นต์ความผิดพลาดอยู่ที่ 39.47 %

รูปที่ 4.8 ผลการทดลองการตั้งค่าโครงข่ายประสาทเทียม ชุดที่ 5

จากการทดลองนี้จึงสามารถสรุปค่าที่ใช้ในส่วนของซอฟต์แวร์โครงข่ายประสาทเทียม ในส่วนของตัวแปร จำนวนฮิดเดนโหนดเท่ากับ 6 โหนด ซึ่งจะทำให้เกิดค่าความผิดพลาดที่ต่ำที่สุด

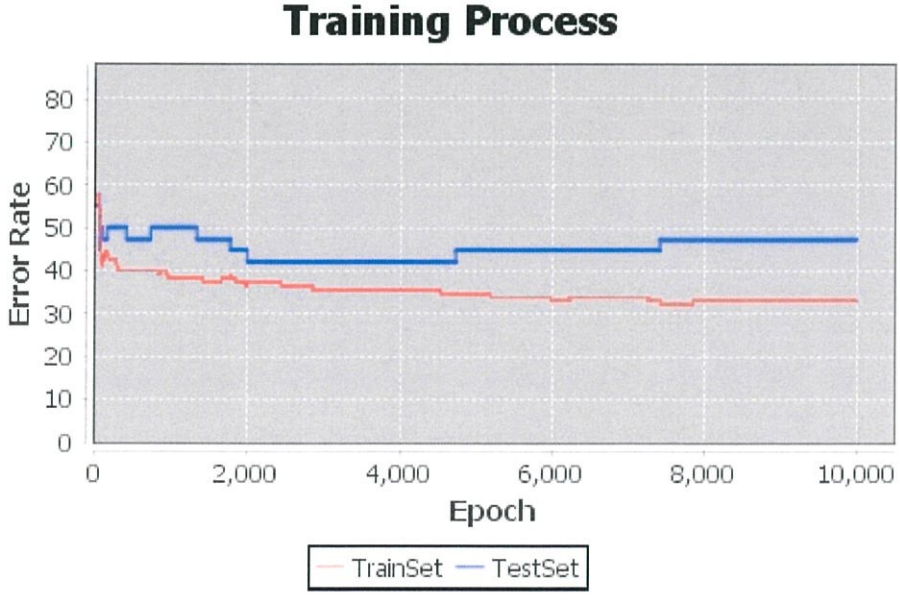
4.1.3.2 การทดลองหาอัตราการเรียนรู้

อัตราการเรียนรู้ (Learning Rate) เป็นหนึ่งปัจจัยที่มีผลต่อการเรียนรู้ของโครงข่ายประสาทเทียม ในการการทดลองนี้ จะทดสอบที่อัตราการเรียนรู้ 3 ค่าได้แก่ 0.1, 0.05 และ 0.01 เพื่อหาค่าที่ดีที่สุดสำหรับการฝึกฝนโครงข่ายประสาทเทียม โดยใช้ฮิดเดนโหนดเท่ากับ 6 โหนด จำนวนรอบสูงสุด 10,000 รอบและคาบของการแสดงผลเท่ากับ 1 รอบ เนื่องจากการตั้งค่าดังนี้ จะทำให้สังเกตความแตกต่างเมื่อทำการเปลี่ยนค่าอัตราการเรียนรู้ได้ง่ายขึ้น

1) ทดลองที่ค่าอัตราการเรียนรู้ 0.1

รูปที่ 4.9 ผลการทดลองการตั้งค่าอัตราการเรียนรู้ ชุดที่ 1

2) ทดลองที่ค่าอัตราการเรียนรู้ 0.05



รูปที่ 4.10 ผลการทดลองการตั้งค่าอัตราการเรียนรู้ ชุดที่ 2

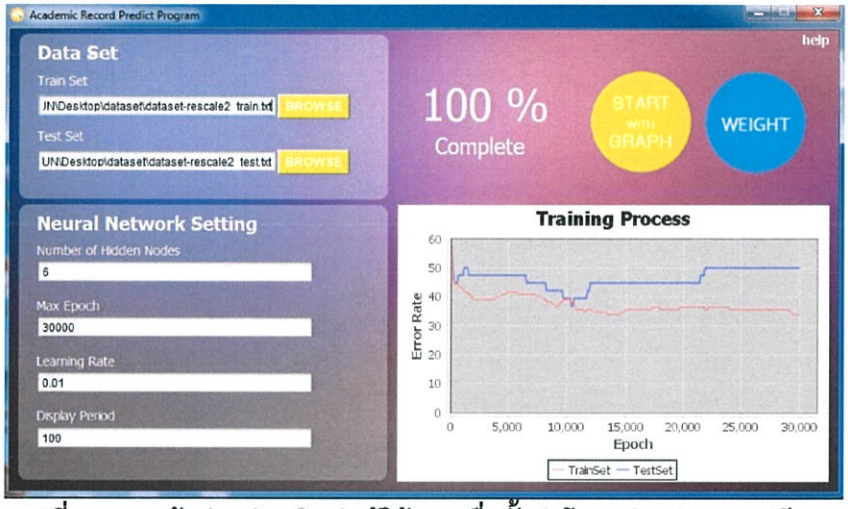
3) ทดลองที่ค่าอัตราการเรียนรู้ 0.01

รูปที่ 4.11 ผลการทดลองการตั้งค่าอัตราการเรียนรู้ ชุดที่ 3

จากการทดลองนี้ทำให้สังเกตเห็นได้ว่า ค่าอัตราการเรียนรู้นั้นส่งผลกระทบโดยตรงต่อการเรียนรู้ของโครงข่ายประสาทเทียม จากการเทียบค่าอัตราการเรียนรู้ทั้ง 3 ค่าข้างต้น จากผลลัพธ์จึงสรุปได้ว่า หากตั้งค่านี้มากเกินไป จะทำให้โครงข่ายประสาทมีการเรียนรู้ที่เร็วเกินไป ซึ่งจะทำให้ค่าน้ำหนักมีการเปลี่ยนแปลงที่รวดเร็ว ส่งผลให้ค่าน้ำหนักที่ปรับปรุงจากการฝึกฝนแต่ละครั้งเกิดความไม่เสถียร ทำให้ประสิทธิภาพการเรียนรู้ของโครงข่ายประสาทเทียมลดลงไป จากข้อสรุปนี้ แสดงให้เห็นว่าค่าอัตราการเรียนรู้ที่มีค่าต่ำจะมีประสิทธิภาพมากกว่า โครงงานนี้จึงได้เลือกใช้ค่าอัตราการเรียนรู้ที่ 0.01 ในการทำงานจริง

4.2 การทดลองซอฟต์แวร์เบื้องหลังส่วนของผู้พัฒนา

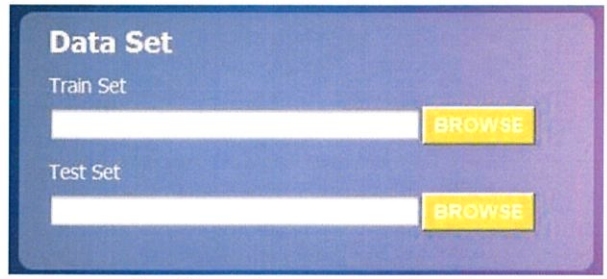
ในส่วนนี้จะทดลองการทำงานของซอฟต์แวร์ที่ใช้ในการตั้งค่าโครงข่ายประสาทเทียม ที่มีส่วนติดต่อกับผู้ใช้งาน (User Interface) ที่ชัดเจน เพื่อความสะดวกสบายในการใช้งานมากขึ้นการทดลองจะทำเพื่อดูว่าซอฟต์แวร์สามารถทำงานได้เป็นปกติหรือไม่ โดยการทดลองใช้ฟังก์ชันต่าง ๆ ของซอฟต์แวร์ ดังรูปที่ 4.12



รูปที่ 4.12 หน้าต่างส่วนติดต่อผู้ใช้งานเพื่อตั้งค่าโครงข่ายประสาทเทียม

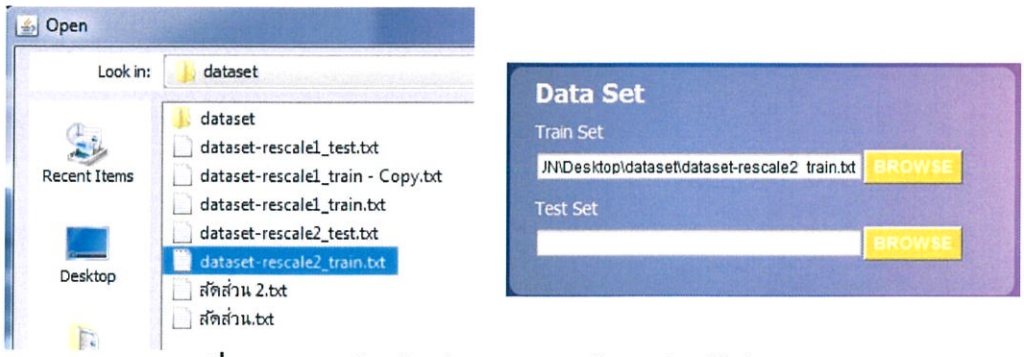
4.2.1 การทดลองใช้ซอฟต์แวร์ส่วนชุดข้อมูล

ในการทดลองส่วนนี้ จะเป็นการทดลองเรียกไฟล์ของซอฟต์แวร์ สำหรับการเรียกไฟล์ชุดข้อมูลสำหรับการฝึกฝนโครงข่ายประสาทเทียม ดังรูปที่ 4.13



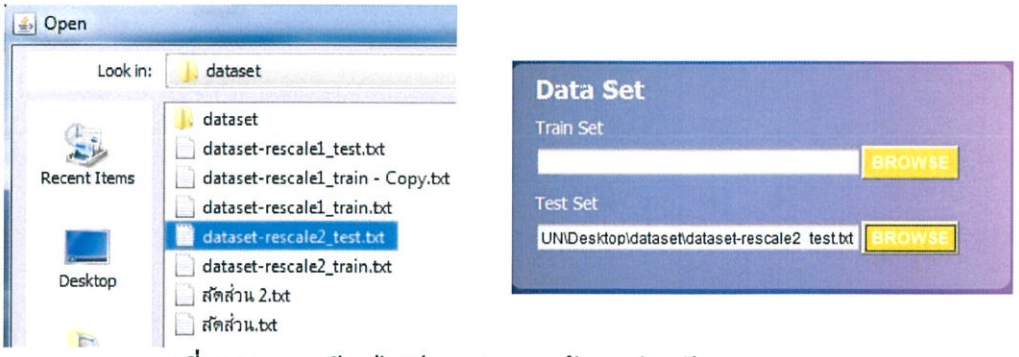
รูปที่ 4.13 ซอฟต์แวร์โครงข่ายประสาทเทียม ส่วนของชุดข้อมูล

1) ทดลองเรียกไฟล์ชุดข้อมูลสำหรับฝึกฝน



รูปที่ 4.14 การเรียกไฟล์ของส่วนชุดข้อมูลสำหรับฝึกฝน

2) ทดลองเรียกไฟล์ชุดข้อมูลสำหรับการทดสอบ



รูปที่ 4.15 การเรียกไฟล์ของส่วนชุดข้อมูลสำหรับการทดสอบ

จากการทดลองนี้ ทุกๆครั้งที่มีการเรียกไฟล์ซึ่งมีข้อมูลหรือมีฟอร์แมตของข้อมูลผิดไปจากที่กำหนดไว้ โปรแกรมจะทำการแจ้งเตือนแก่ผู้พัฒนา เพื่อให้ผู้พัฒนากลับไปแก้ไขหรือตรวจสอบชุดข้อมูลใหม่อีกครั้งเพื่อนำมาใส่ใหม่

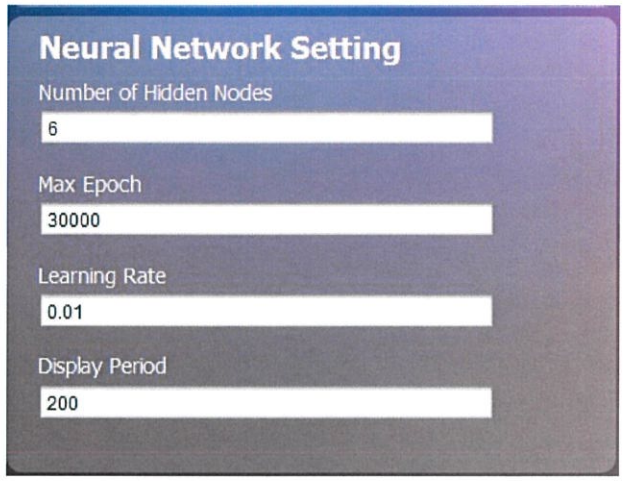
4.2.2 การทดลองใช้ซอฟต์แวร์ฝึกฝนโครงข่ายประสาทเทียม

ในการทดลองเพื่อฝึกฝนโครงข่ายประสาทเทียมนั้นมีขั้นตอนการใช้งานซอฟต์แวร์ดังนี้

1) เลือกไฟล์ชุดข้อมูลสำหรับฝึกฝนและทดสอบโครงข่ายประสาทเทียม

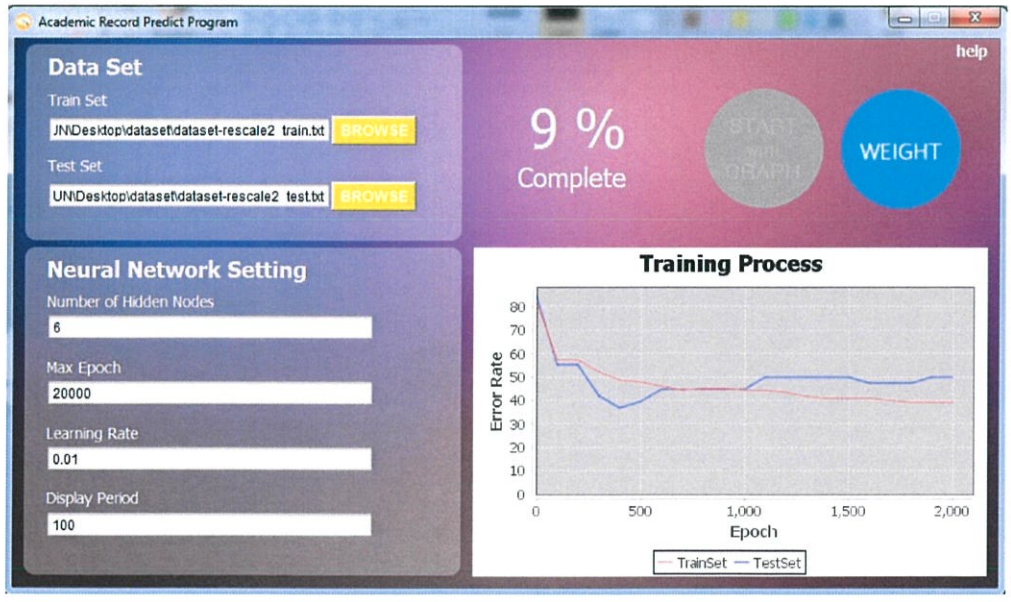
รูปที่ 4.16 ส่วนของชุดข้อมูลเมื่อเลือกไฟล์แล้ว

2) ใส่ค่าที่ใช้เพื่อตั้งค่าการทำงานของโครงข่ายประสาทเทียม ได้แก่ค่า Number of Hidden Nodes, Max Epoch, Learning Rate และ Display Period



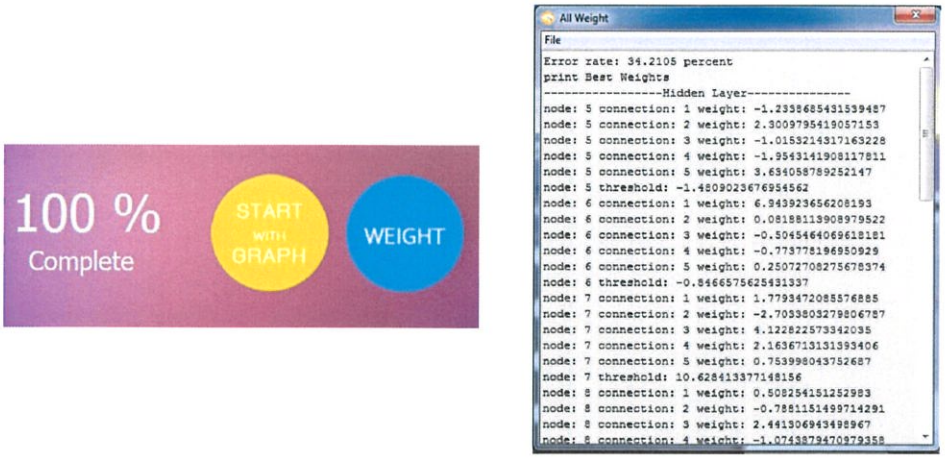
รูปที่ 4.17 การตั้งค่าการทำงานของโครงข่ายประสาทเทียม

3) กดปุ่ม START with GRAPH และดูการทำงานของซอฟต์แวร์จนครบ 100% ในระหว่างนี้จะมีการพล็อตกราฟให้ดูในลักษณะ Real Time



รูปที่ 4.18 ซอฟต์แวร์ระหว่างการฝึกฝนโครงข่ายประสาทเทียม

4) เมื่อซอฟต์แวร์ทำงานเสร็จ กดปุ่ม WEIGHT เพื่อดูค่าผลลัพธ์ได้



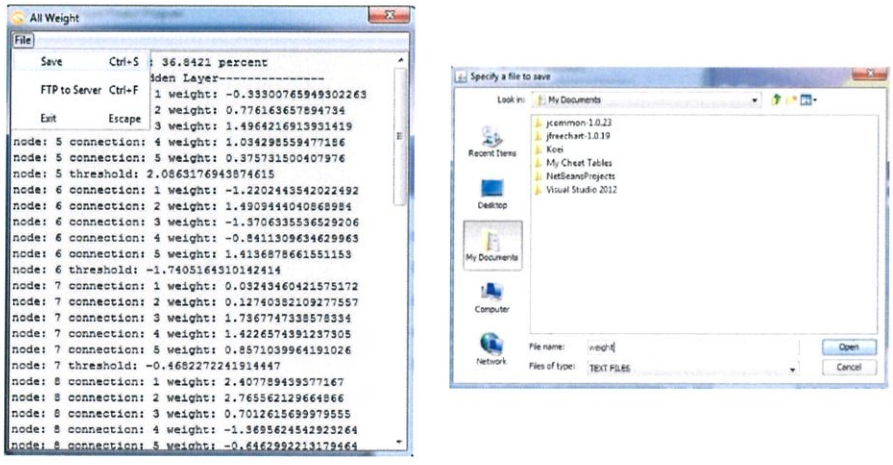
รูปที่ 4.19 หน้าต่างผลลัพธ์เมื่อกดปุ่ม WEIGHT

4.2.3 การทดลองใช้ซอฟต์แวร์ในส่วนของหน้าต่างค่าน้ำหนัก (WEIGHT)

ในส่วนหน้าต่างนี้สามารถเลือกสำหรับทำการ บันทึกผลลัพธ์ไว้ หรือ ทำการอัพโหลด เพื่อนำไปใช้งานบนเว็บแอพพลิเคชั่นได้ โดยแบ่งการทดลองออกเป็นดังนี้

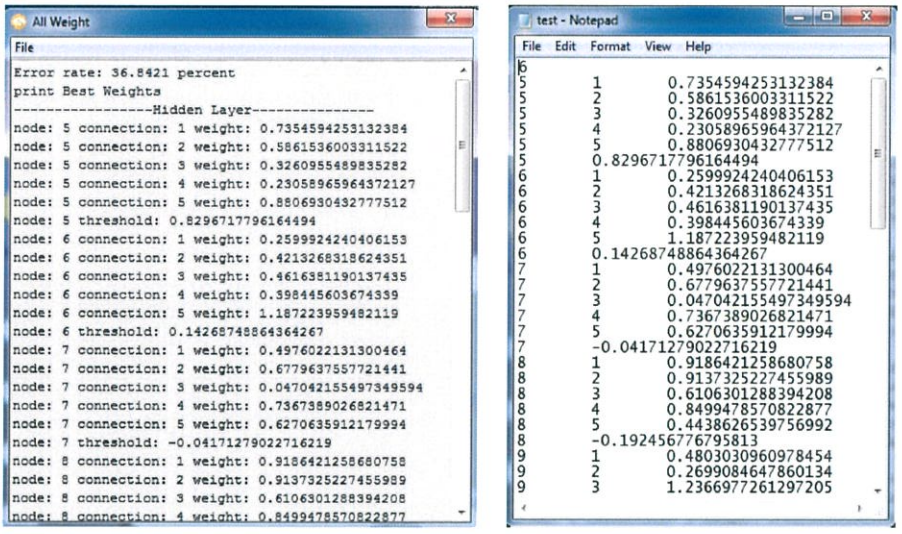
4.2.3.1 การทดลองใช้งานส่วนการบันทึกค่าน้ำหนัก

1) เลือกเมนูส่วนของการบันทึก และเลือกตำแหน่งของโฟลเดอร์ที่ใช้บันทึกค่าน้ำหนัก



รูปที่ 4.20 การเลือกเมนูส่วนการบันทึกไฟล์ ในหน้าต่างค่าน้ำหนัก

2) เข้าสู่โฟลเดอร์ที่บันทึกไฟล์ค่าน้ำหนัก ด้วย Window Explorer และตรวจสอบว่าเป็นค่าน้ำหนักที่บันทึกไปจริง



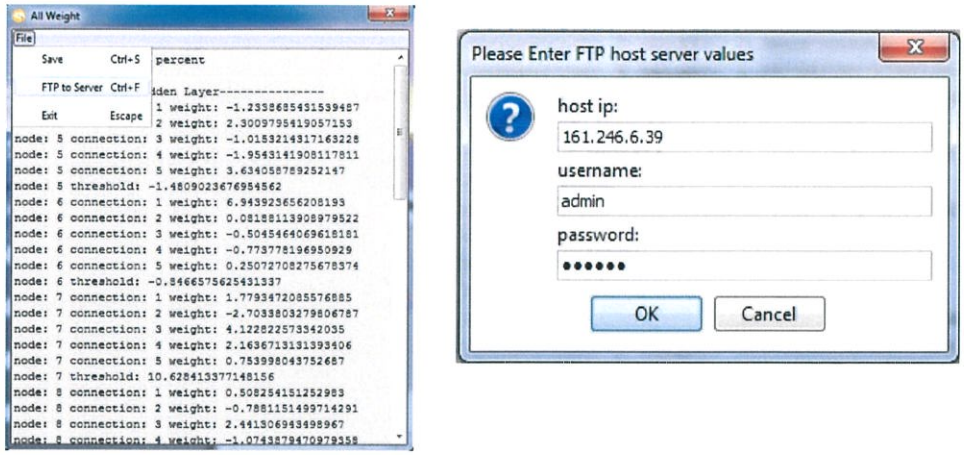
รูปที่ 4.21 รูปเปรียบเทียบค่าน้ำหนักที่บันทึก และที่อยู่ในหน้าต่างค่าน้ำหนัก

จากการทดลองบันทึกค่าน้ำหนักให้อยู่ในรูปของไฟล์ Text จะเห็นได้ว่าคำอธิบายบางอย่างจะถูกตัดออกไป แต่เลขค่าน้ำหนักยังคงเหมือนเดิม โดยในไฟล์ Text ที่บันทึกนั้น จะอยู่ในรูปแบบฟอร์แมตที่บังคับใช้สำหรับการทำนายบนเว็บเซิร์ฟเวอร์ เพื่อให้ผู้พัฒนาสามารถ FTP ไปยังเว็บเซิร์ฟเวอร์ด้วยตนเองได้โดยไม่จำเป็นต้องผ่านโปรแกรม

ฟอร์แมตของไฟล์ที่บันทึกนั้นจะแตกต่างกับที่แสดงผลบนโปรแกรมเพียงเล็กน้อย โดยบรรทัดแรกนั้นจะเป็นเลขของจำนวนฮิดเดนโหนดที่ผู้พัฒนาได้ใช้ฝึกฝน และบรรทัดที่เหลือจะแสดงค่าต่างๆ ซึ่งประกอบด้วย เลขไอดีของโหนด เลขไอดีของการเชื่อมต่อ และค่าน้ำหนัก โดยจะใช้ แท็บ เป็นตัววรรคเสมอ ยกเว้นบางบรรทัดที่ไม่มีเลขไอดีของการเชื่อมต่อ บรรทัดนั้นหมายถึง ค่าน้ำหนักของเทรสโฮลด์ประจำโหนดนั้นๆ ซึ่งจะวรรคด้วยแท็บเช่นเดียวกันกับบรรทัดอื่นๆ

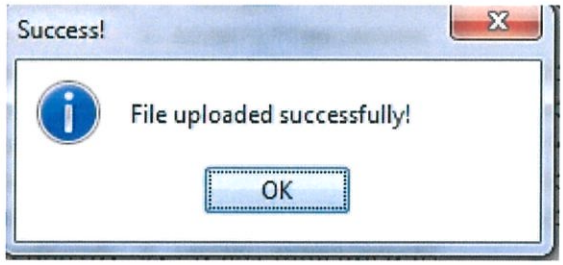
4.2.3.1 ทดลองใช้งานส่วนการอัพโหลดของค่าน้ำหนัก

1) เลือกเมนูส่วนของการอัพโหลดค่าน้ำหนักกรอก Host IP, Username และ Password



รูปที่ 4.22 การกรอก Host IP Username Password

2) เมื่อกรอกข้อมูลครบ จะขึ้น Dialog Box เป็นข้อความ Success เมื่ออัพโหลดข้อมูลสำเร็จ



รูปที่ 4.23 Dialog Box ข้อความ Success เมื่ออัพโหลดสำเร็จ

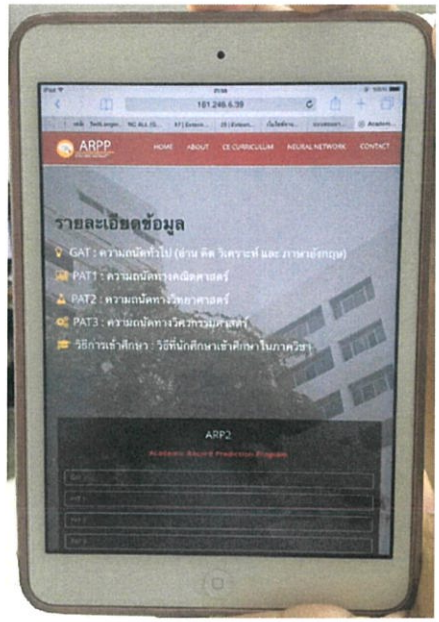
4.2.4 การทดลองใช้เว็บแอพพลิเคชั่น

เว็บแอพพลิเคชั่นเป็นส่วนที่ผู้ใช้งานทั่วไปเข้าใช้งานส่วนการทำนายผลการเรียนล่วงหน้า มีการทดลองใช้งานส่วนต่าง ๆ ดังนี้

4.2.4.1 การทดลองใช้งานเว็บแอพพลิเคชั่นผ่านอุปกรณ์ต่าง ๆ

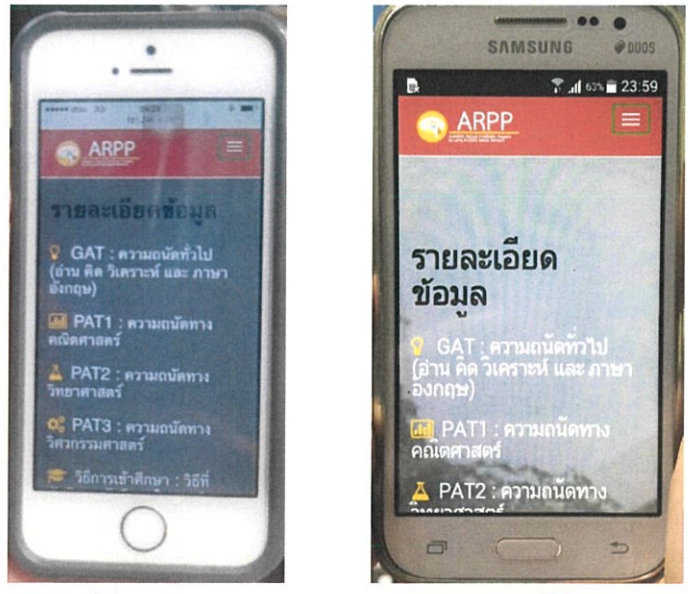
เป็นทดลองเข้าสู่เว็บแอพพลิเคชั่น โดยใช้อุปกรณ์ต่างๆ โดยใช้ชื่อโดเมนคือ 161.246.6.39:8080/arpp/ ซึ่งเป็นโดเมนของแอพพลิเคชั่นนี้

1.) IPad



รูปที่ 4.24 การใช้งานเว็บแอพพลิเคชั่นบนอุปกรณ์ไอแพต

2.) Smartphone



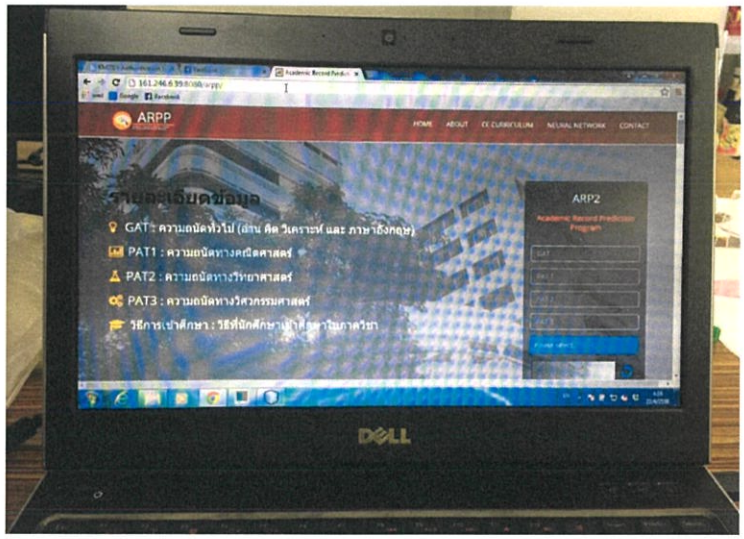
(a) (b)

รูปที่ 4.25 การใช้งานบนอุปกรณ์สมาร์ทโฟน

(a) Iphone

(b) Android

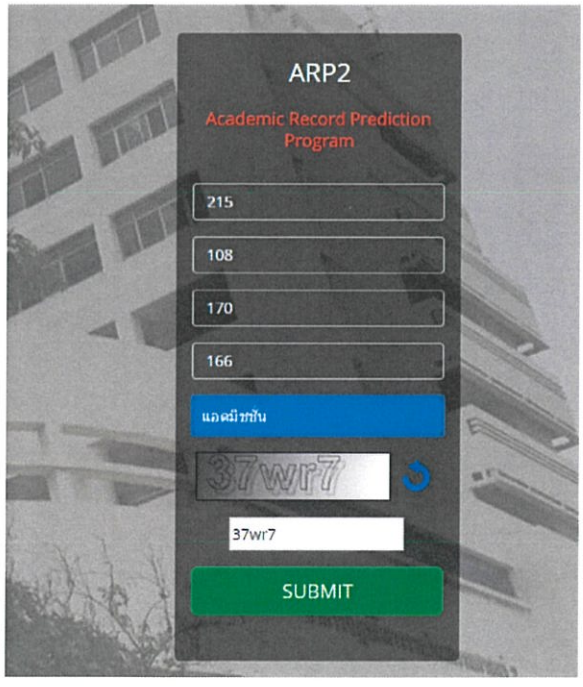
3.) Notebook



รูปที่ 4.26 การใช้งานบนคอมพิวเตอร์โน้ตบุค

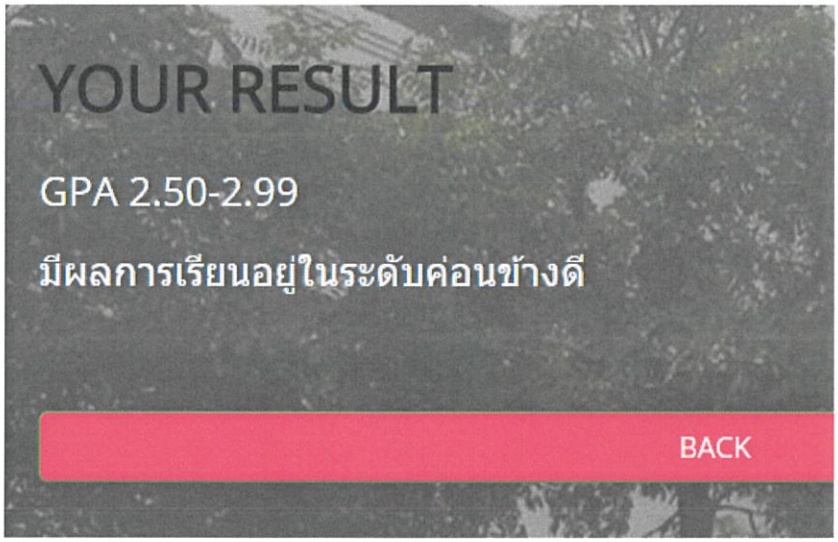
4.2.4.2 การทดสอบใช้งานเว็บแอพพลิเคชั่นทำนายผลการเรียน

1) กรอกข้อมูลในเว็บแอพพลิเคชั่นในส่วนสำหรับทำนายผลการเรียน



รูปที่ 4.27 การกรอกข้อมูลสำหรับใช้ในการทำนาย

2) กด SUBMIT และตรวจสอบว่า เว็บแอพพลิเคชั่นสามารถแสดงผลลัพธ์ได้

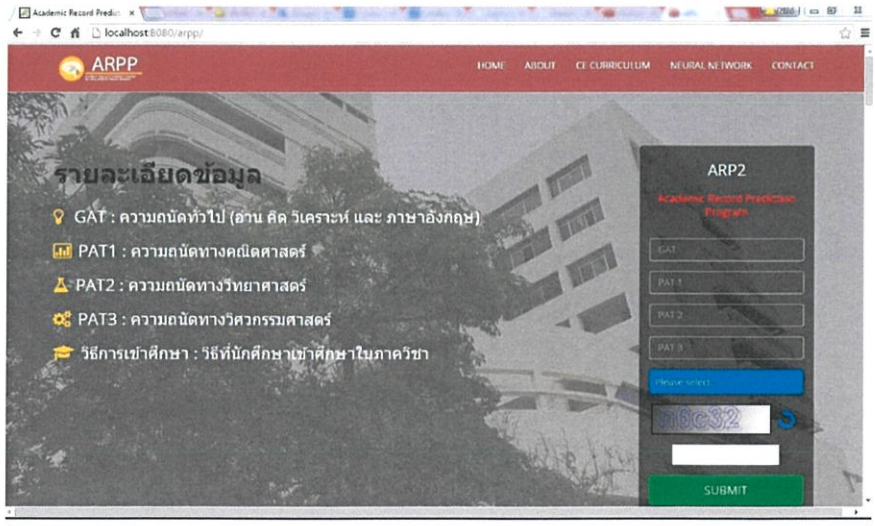


รูปที่ 4.28 เว็บแอพพลิเคชั่นแสดงผลลัพธ์

4.2.4.3 ทดลองการใช้งานเว็บแอพพลิเคชั่นผ่านบราวเซอร์ต่าง ๆ

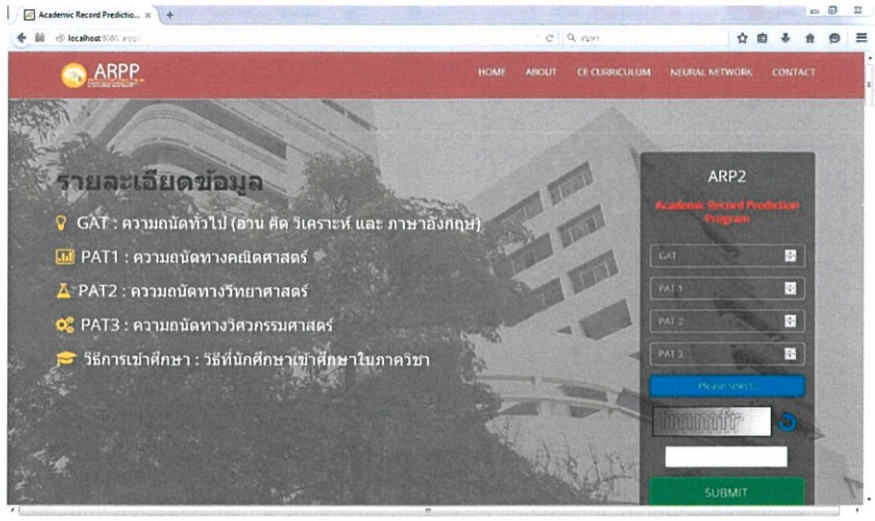
ในการทดลองส่วนนี้จะทำการทดลองผ่าน 3 เว็บบราวเซอร์ต่าง ๆ ที่นิยมใช้ในปัจจุบัน ได้แก่ Chrome, Firefox และ internet explorer

1) ทดลองบนเว็บบราวเซอร์ Chrome



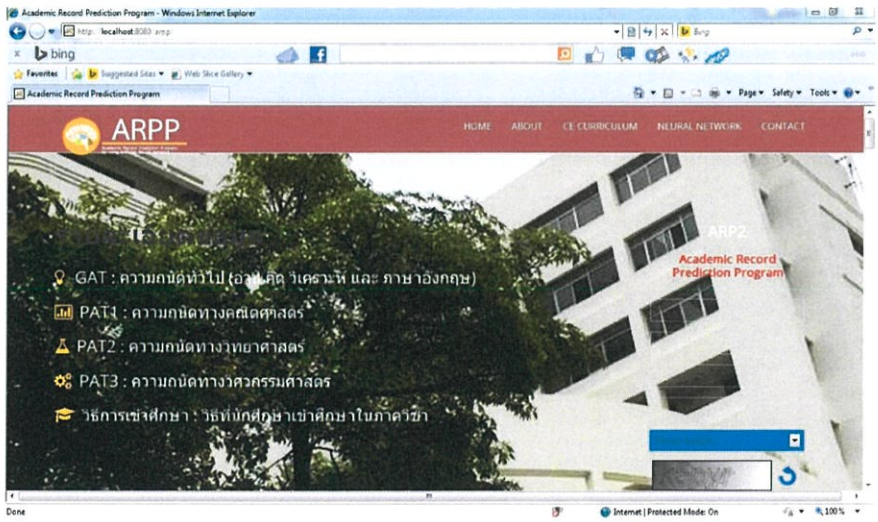
รูปที่ 4.29 ทดลองแสดงผลบนบราวเซอร์ Chrome

2) ทดลองบนเว็บบราวเซอร์ Firefox



รูปที่ 4.30 ทดลองแสดงผลบนบราวเซอร์ Firefox

3) ทดลองบนเว็บบราวเซอร์ Internet Explorer



รูปที่ 4.31 ทดลองแสดงผลบนบราวเซอร์ Internet Explorer

จากการทดลอง พบว่าบราวเซอร์ที่สามารถใช้งานเว็บ แอพพลิเคชั่นได้ดีที่สุดคือ Firefox และ Chrome ส่วนบราวเซอร์ Internet Explorer นั้นเนื่องจากไม่รองรับการทำงานของ Bootstrap ทำให้การแสดงผลบนบราวเซอร์นี้ไม่ดี และ ไม่สามารถใช้งานฟังก์ชันได้ครบ

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

โปรแกรมทำนายผลการเรียนล่วงหน้าด้วยโครงข่ายประสาทเทียม เป็นซอฟต์แวร์ที่จะทำนายผลการเรียนของนักศึกษาชั้นปีที่ 1 เทอม 1 จากคะแนน GAT, PAT1, PAT2, PAT3และ วิธีการเข้าศึกษา เนื่องจากในแต่ละปีนักศึกษาใหม่ของภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังบางส่วน เข้าเรียนแล้วมีผลการเรียนอยู่ในระดับที่ไม่ค่อยดี เช่น คะแนน GPA มีค่าต่ำกว่า หรือใกล้เคียง 2.00 เพื่อให้สามารถตรวจสอบได้ว่า เมื่อเข้ามาเรียนในภาควิชาวิศวกรรมคอมพิวเตอร์แล้วนั้นผลการเรียนจะอยู่ในระดับใด จึงเกิดการพัฒนาโปรแกรมทำนายผลการเรียนล่วงหน้าด้วยโครงข่ายประสาทเทียมขึ้น ในการพัฒนาซอฟต์แวร์นั้น จะพัฒนาส่วนของโครงข่ายประสาทเทียมด้วยภาษา Java ในส่วนของซอฟต์แวร์ถูกแบ่งเป็นสองส่วนคือ ส่วนแรกเป็นส่วนของการตั้งค่าของโครงข่ายประสาทเทียม เป็นหน้าต่างใช้งานบนระบบปฏิบัติการ Windows ให้สำหรับผู้พัฒนาใช้สำหรับการตั้งค่า และอัพโหลดไฟล์ค่าน้ำหนักไปใช้งานบนเว็บแอพพลิเคชั่น สำหรับส่วนที่สองนั้นเป็นเว็บแอพพลิเคชั่น ถูกพัฒนาขึ้นด้วยภาษา HTML

ส่วนของการใช้งานระบบ ในขั้นแรก ผู้พัฒนาจะทำการฝึกฝนโครงข่ายประสาทเทียม เพื่อให้ได้ค่าน้ำหนัก เพื่อนำไปใช้ในการทำนายผลการเรียน ในส่วนของผู้ใช้งานนั้น สามารถใช้งานระบบจากการใช้งานเว็บแอพพลิเคชั่น เมื่อกรอกข้อมูลที่จำเป็นสำหรับการทำนาย ได้แก่ คะแนน GAT, PAT1, PAT2, PAT3 และ วิธีการเข้าศึกษาแล้วนั้น ค่าผลลัพธ์จะออกมาเป็นช่วงคะแนน ได้แก่ 0.00-1.99, 2.00-2.49, 2.50-2.99 และ 3.00-4.00

จากการทดลองสามารถสรุปได้ว่าการตั้งค่าโครงข่ายประสาทเทียมเพื่อให้ได้ผลลัพธ์ที่ดีที่สุด ซึ่งถูกเทียบโดยหาค่าเปอร์เซ็นต์ความผิดพลาดที่น้อยที่สุด จากการเปลี่ยนแปลงจำนวนฮิดเดนโหนด จำนวนรอบการฝึกฝน และ อัตราการเรียนรู้ โครงข่ายประสาทเทียมจะให้ผลการทำนายดีที่สุดจากการตั้งค่าจำนวนฮิดเดนโหนด อยู่ที่ 6 โหนดมีรอบการฝึกฝนเพื่อให้ได้ค่าเปอร์เซ็นต์ความผิดพลาดที่น้อยที่สุดอยู่ที่ 10,000 รอบ และตั้งค่าอัตราการเรียนรู้ของโครงข่ายประสาทเทียมเป็น 0.01 ถึงแม้ว่าการตั้งค่าของโครงข่ายประสาทเทียมจะทำให้ได้ค่าเปอร์เซ็นต์ความผิดพลาดที่น้อยที่สุด แต่ค่าที่ได้นั้นยังถือว่าอยู่ในระดับที่สูง จากการทดลองเกี่ยวกับตัวแปรที่ใช้ในการทำนายผลการเรียนล่วงหน้า สามารถสรุปได้ว่า ค่าคะแนน GAT, PAT1, PAT2 และ PAT3 มีความสัมพันธ์ต่อคะแนน GPA น้อย เมื่อนำมาเป็นตัวแปรสำหรับการทำนายแล้ว ผลการทำนายจึงมีค่าความผิดพลาดสูง

5.2 ปัญหา อุปสรรค และ แนวทางแก้ไข

5.2.1 ปัญหาและอุปสรรค

1) การกรอกข้อมูลของแบบสอบถามไม่ครบ
2) จำนวนเรคคอร์ดของข้อมูลมีน้อย
3) ค่าความผิดพลาดของการทำนายยังมีมาก
4) บาง Browser ไม่สามารถใช้งานเว็บแอพพลิเคชั่นได้สมบูรณ์

5.2.2 แนวทางแก้ไขปัญหา

1) ทำการตัดข้อมูลจากแบบสอบถามที่กรอกไม่ครบทิ้ง
2) ทำการเก็บข้อมูลเพิ่ม แต่เนื่องจากจำนวนของนักศึกษาที่พบ และส่งแบบสอบถามให้ทำนั้นมีน้อย ในขั้นสุดท้ายจำนวนข้อมูลที่นำมาใช้ยังมีจำนวนน้อยเกินไปการตัดข้อมูลที่เป็นสิ่งรบกวนจึงทำได้ยาก
3) ในส่วนของชุดข้อมูลจะทำการกรองข้อมูล ตัดข้อมูลที่เป็นตัวรบกวน (Noise) และทำการรีสเกลข้อมูล ส่วนในด้านของซอฟต์แวร์โครงข่ายประสาทเทียมนั้น จะทำการตรวจสอบอัลกอริทึม และตรวจหาข้อผิดพลาดของโปรแกรม
4) เว็บบราวเซอร์ที่พบว่าใช้งานได้ไม่สมบูรณ์นั้น คือ Internet Explorer ที่ไม่สามารถใช้งานได้สมบูรณ์เนื่องจากการไม่รองรับการใช้งาน Bootstrap ซึ่งเป็นส่วนหลักในการพัฒนาเว็บแอพพลิเคชั่น แต่เนื่องจากบราวเซอร์นี้ไม่เป็นที่นิยมในการใช้งานในปัจจุบัน และ มีทางเลือกตัวอื่นให้ใช้ เช่น Chrome หรือ Firefox ปัญหาที่เกิดขึ้นบน Internet Explorer จึงไม่มีความจำเป็นในการแก้ไข

5.3 แนวทางในการพัฒนาต่อ

แนวทางการพัฒนาต่อ อาจทำการทดลองเพื่อหาวิธีการที่ทำให้ค่าเปอร์เซ็นต์ความผิดพลาดลดน้อยลงเพื่อให้การทำนายผลมีความแม่นยำมากขึ้น หรือทำการขยายการทำนายจากปัจจุบันใช้การทำนายของนักศึกษาปีที่ 1 เทอม 1 ให้สามารถทำนายผลการเรียนของชั้นปีอื่นจากการเก็บข้อมูลและพัฒนาซอฟต์แวร์เพิ่มเติม

ในการพัฒนาซอฟต์แวร์ต่อไปนั้นมีแนวทางการพัฒนาจากการเพิ่มชุดข้อมูลที่ใช้ในการฝึกฝนโครงข่ายประสาทเทียม หรือ เพิ่มจำนวนเรคคอร์ดที่อยู่ในชุดข้อมูลสำหรับการฝึกฝนจากการเก็บข้อมูลจากนักศึกษาให้มากขึ้น นอกจากนี้การเพิ่มความแม่นยำในการประมวลผลของโครงข่ายประสาทเทียม สามารถเพิ่มได้จากการเพิ่มจำนวนตัวแปรอินพุต จากเดิมใช้คะแนน GAT PAT1 PAT2

PAT3 และวิธีการเข้าศึกษา อาจเพิ่มตัวแปรที่เกี่ยวข้องอื่นๆ เช่น เกรดที่จบการศึกษาช่วงมัธยม หรือวิชาที่ถนัดในช่วงมัธยม ในการเพิ่มประสิทธิภาพการทำนาย อาจใช้อัลกอริทึมหรือเครื่องมือที่ใช้ในการทำนายเพิ่มเติม นอกเหนือจากโครงข่ายประสาทเทียม เช่น เปรียบเทียบผลการทำนายของต้นไม้การตัดสินใจกับโครงข่ายประสาทเทียม เป็นต้น

บรรณานุกรม

[1] Michael Negnevitsky. **Artificial Intelligence A Guide to Intelligent Systems.** Second Edition. Pearson Education Limited. 2005.

[2] **GAT/PAT คือ.** [Online]. Available : http://www.niets.or.th/index.php/exam_information/view_se1/9/7

[3] ติวฟรี ดอทคอม(นามแฝง). **GAT/PAT คืออะไร? ทำความรู้จักอย่างหมดเปลือก.** [Online]. Available : http://www.tewfree.com/gat-pat-คืออะไร/#gat-pat

[4] วิกิพีเดีย สารานุกรมเสรี. 2554.**ภาษาจาวา.**[Online]. Available : http://th.wikipedia.org/wiki/ภาษาจาวา

[5] SukdaSetrin. **ประวัติความเป็นมาของภาษาจาวา.**[Online]. Available : http://java-sample-thai.blogspot.com/2013/06/blog-post_24.html

[6] SettawutNamkam. **การเขียนโปรแกรมด้วยภาษา Java.** [Online]. Available : http://settawut123456.blogspot.com/2013/05/java.html

[7] Suchada(นามแฝง). 2552. **ภาษา JSP**. [Online]. Available : http://suchada51122470136.blogspot.com/

[8] ธีรภัทร มนตรีศาสตร์. **Apache เว็บเซิร์ฟเวอร์สารพัดประโยชน์**. [Online]. Available : http://www.itdestination.com/articles/idc-apache/