

รายงานวิชา 261456

เรื่อง

Multilayer Perceptron (MLP) using Genetic Algorithms

จัดทำโดย

นายธนัช ธนัญชัย 650610838

เสนอ

รศ.ดร.คันสนีย์ เอื้อพันธ์วิริยะกุล

รายงานนี้เป็นส่วนหนึ่งของวิชา 261456

ภาคการศึกษาที่ 1 ปีการศึกษา 2567

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของวิชา 261456 โดยมีวัตถุประสงค์เพื่อพัฒนาและฝึกสอนโครงข่ายประสาทเทียมแบบหลายชั้น (MLP) สำหรับการจำแนกประเภทข้อมูลสองคลาส โดยใช้วิธีการทางพันธุกรรม (Genetic Algorithm : GA) แทนการใช้วิธีการปรับค่าน้ำหนักแบบดั้งเดิมโดยวิธีการของ GA จะมุ่งเน้นการปรับปรุงกลุ่มประชากรผ่านกระบวนการเลือกสรร (selection), การกลายพันธุ์ (mutation) และการผสมข้ามสายพันธุ์ (crossover) เพื่อค้นหาโมเดลที่มีความแม่นยำสูงสุด โครงการนี้มีการทดลองซ้ำหลายรอบต่อโครงสร้างชั้นซ่อนเพื่อให้ได้ผลลัพธ์ที่น่าเชื่อถือและลดความแปรปรวนและเพื่อตรวจสอบประสิทธิภาพของการหาค่าน้ำหนักแบบใหม่ให้มีหลายทางเลือกนอกจากการวิธี Gradient Descent

ผู้จัดทำ

27/10/2567

ผลการทดลอง

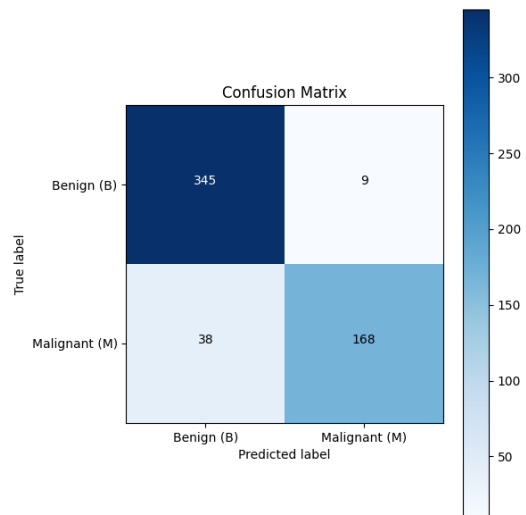
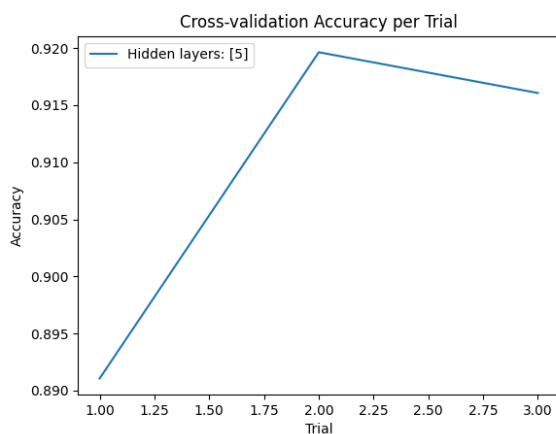
จากการทดลองในการเขียนโปรแกรมสำหรับ Train Multilayer Perceptron โดยใช้ Genetic Algorithms โดยได้ทำการทดลองกับ wdbc.data (Wisconsin Diagnostic Breast Cancer (WDBC) จาก UCI Machine learning Repository) โดยที่ data set นี้ มี 2 classes และ 30 features ซึ่งในแต่ละ sample จะมีทั้งหมด 32 ค่าโดย

ประกอบไปด้วย 1) ID number , 2) Diagnosis (M = malignant, B = benign) 3-32) เป็นค่า features ทั้ง 30 ตัวและ ให้ทำการทดลองโดยใช้ 10% cross validation เพื่อทดสอบ validity ของ network ที่ได้ และให้ทำการเปลี่ยนแปลงจำนวน hidden layer และ nodes ได้ข้อสรุปดังนี้

ทดลองโดยมีการจัดค่าพารามิเตอร์ทั้งหมดดังนี้

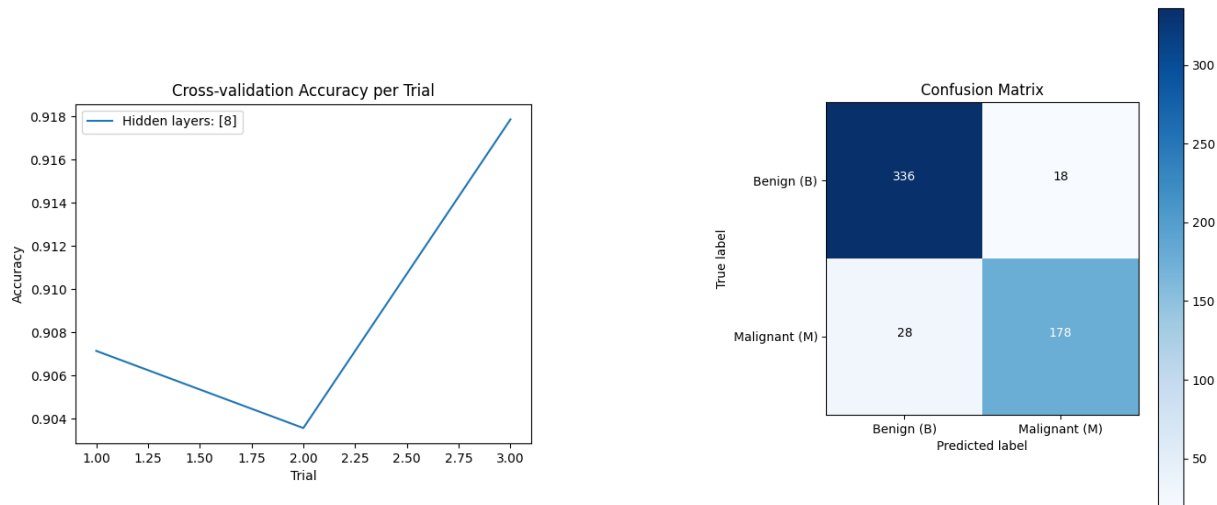
กรณีเปลี่ยน hidden layers

กำหนดให้ population size = 350 , num generations = 150 , mutation rate = 0.1 , crossover rate = 0.8 และให้ hidden layers = 5 โดยทดลองทั้งหมด 3 รอบแล้วหาค่าเฉลี่ยพบว่าได้ข้อสรุปดังนี้



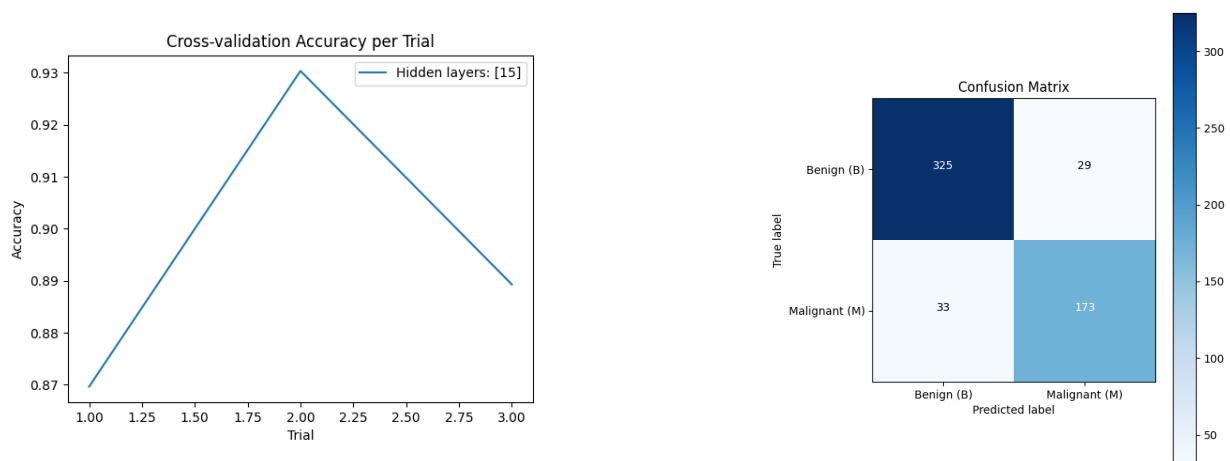
ได้ผลลัพธ์เป็น Average cross-validation accuracy over 3 trials: 0.91

กำหนดให้ population size = 350 , num generations = 150 , mutation rate = 0.1 , crossover rate = 0.8 และให้ hidden layers = 8 โดยทดลองทั้งหมด 3 รอบแล้วหาค่าเฉลี่ยพบว่าได้ข้อสรุปดังนี้



ได้ผลลัพธ์เป็น Average cross-validation accuracy over 3 trials: 0.91

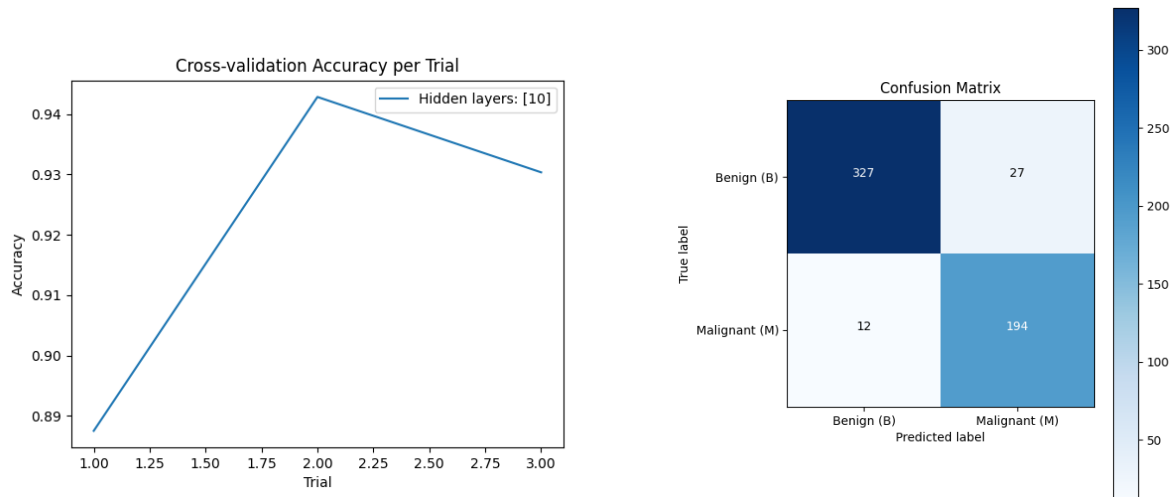
กำหนดให้ population size = 350 , num generations = 150 , mutation rate = 0.1 , crossover rate = 0.8 และให้ hidden layers = 15 โดยทดลองทั้งหมด 3 รอบแล้วหาค่าเฉลี่ยพบว่าได้ข้อสรุปดังนี้



ได้ผลลัพธ์เป็น Average cross-validation accuracy over 3 trials: 0.90

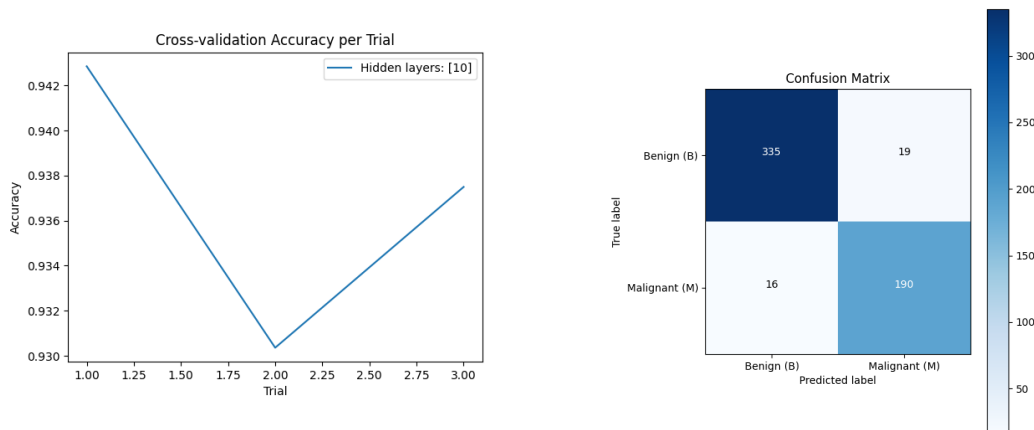
กรณีเปลี่ยน population sizes

กำหนดให้ population size = 500 , num generations = 150 , mutation rate = 0.1 , crossover rate = 0.8 และให้ hidden layers = 10 โดยทดลองทั้งหมด 3 รอบแล้วหาค่าเฉลี่ยพบว่าได้ข้อสรุปดังนี้



ได้ผลลัพธ์เป็น Average cross-validation accuracy over 3 trials: 0.92

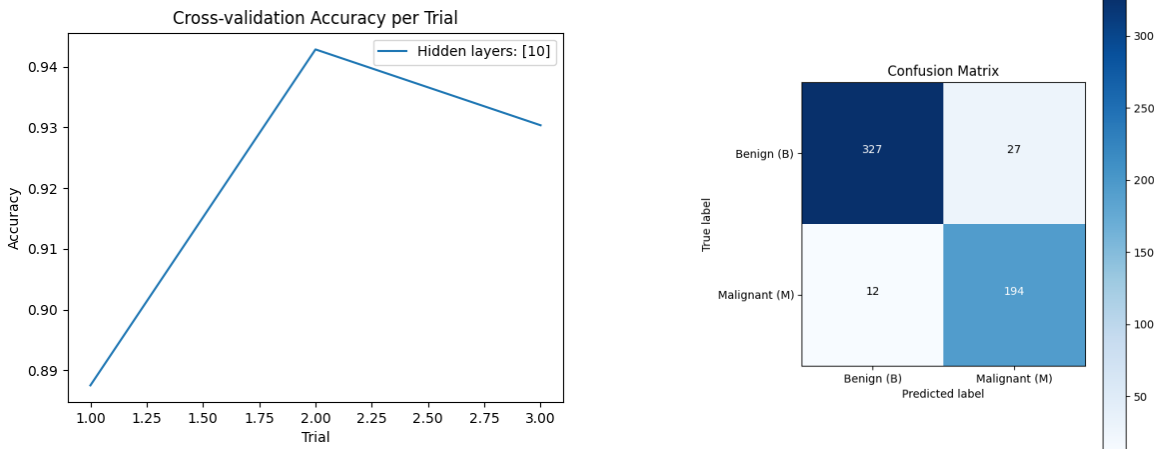
กำหนดให้ population size = 1000 , num generations = 150 , mutation rate = 0.1 , crossover rate = 0.8 และให้ hidden layers = 10 โดยทดลองทั้งหมด 3 รอบแล้วหาค่าเฉลี่ยพบว่าได้ข้อสรุปดังนี้



ได้ผลลัพธ์เป็น Average cross-validation accuracy over 3 trials: 0.94

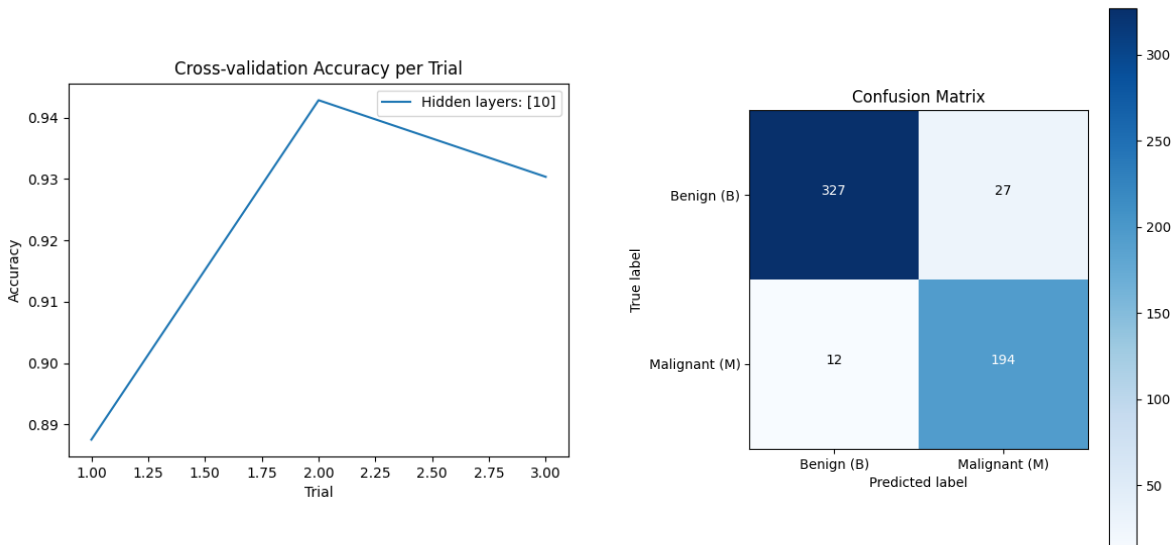
กรณีเปลี่ยน num generations

กำหนดให้ population size = 500 , num generations = 350 , mutation rate = 0.1 , crossover rate = 0.8 และให้ hidden layers = 10 โดยทดลองทั้งหมด 3 รอบแล้วหาค่าเฉลี่ยพบว่าได้ข้อสรุปดังนี้



ได้ผลลัพธ์เป็น Average cross-validation accuracy over 3 trials: 0.92

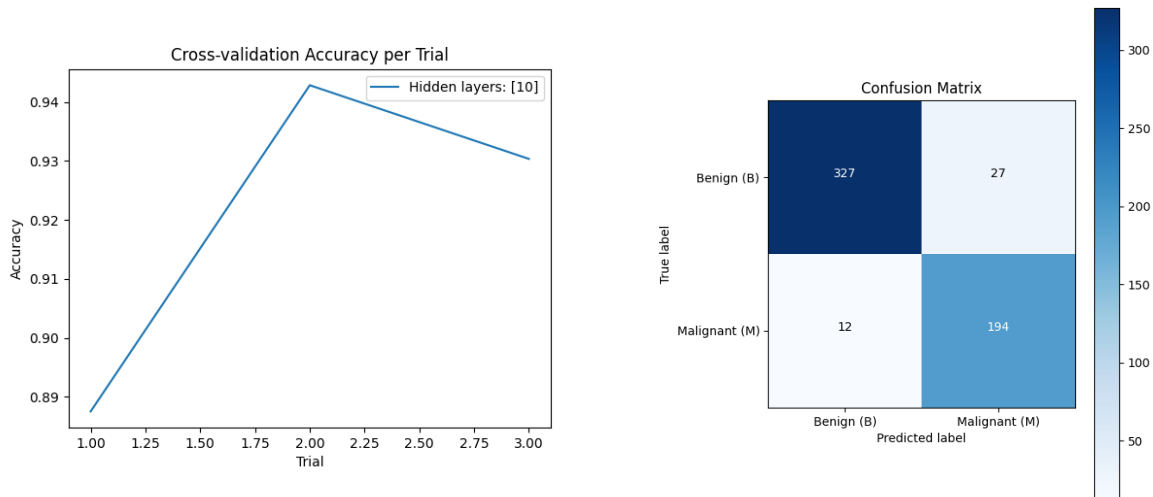
กำหนดให้ population size = 500 , num generations = 750 , mutation rate = 0.1 , crossover rate = 0.8 และให้ hidden layers = 10 โดยทดลองทั้งหมด 3 รอบแล้วหาค่าเฉลี่ยพบว่าได้ข้อสรุปดังนี้



ได้ผลลัพธ์เป็น Average cross-validation accuracy over 3 trials: 0.92

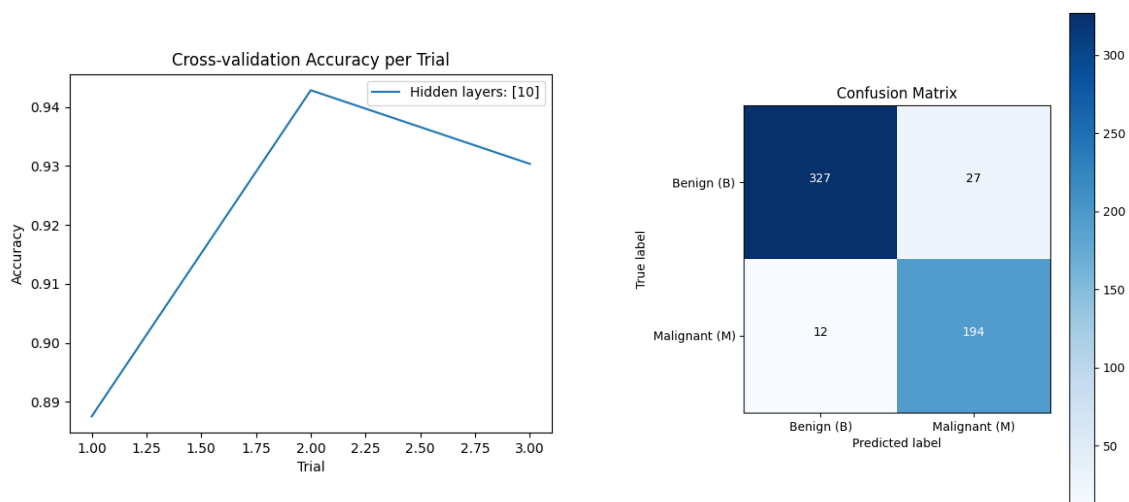
กรณีเปลี่ยน mutation rate

กำหนดให้ population size = 500 , num generations = 350 , mutation rate = 0.01 , crossover rate = 0.8 และให้ hidden layers = 10 โดยทดลองทั้งหมด 3 รอบแล้วหาค่าเฉลี่ยพบว่าได้ข้อสรุปดังนี้



ได้ผลลัพธ์เป็น Average cross-validation accuracy over 3 trials: 0.92

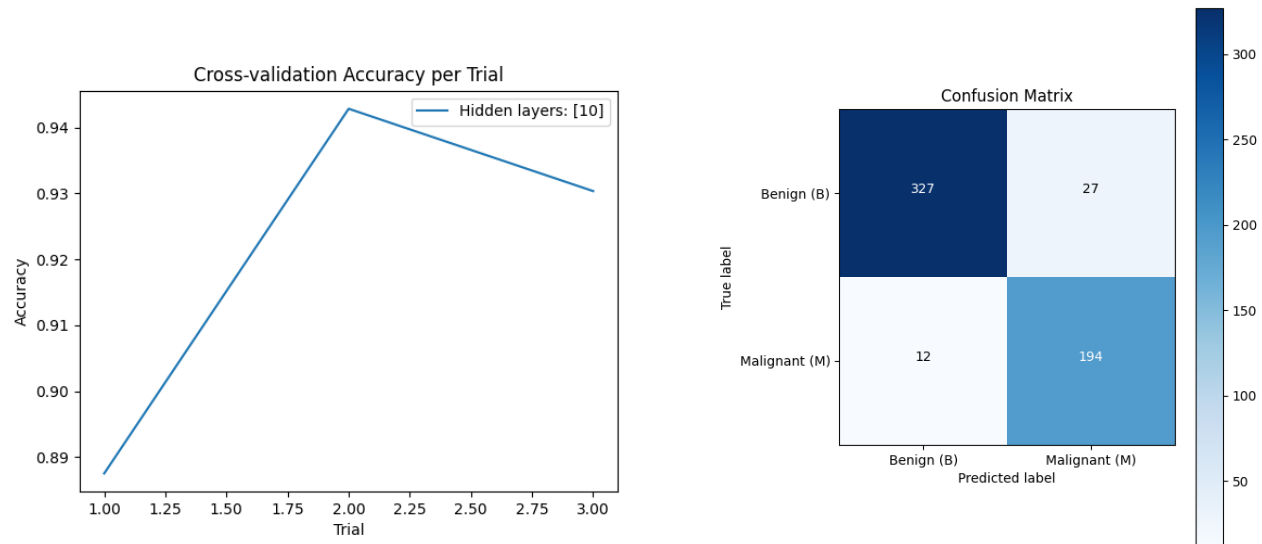
กำหนดให้ population size = 500 , num generations = 350 , mutation rate = 0.1 , crossover rate = 0.8 และให้ hidden layers = 10 โดยทดลองทั้งหมด 3 รอบแล้วหาค่าเฉลี่ยพบว่าได้ข้อสรุปดังนี้



ได้ผลลัพธ์เป็น Average cross-validation accuracy over 3 trials: 0.92

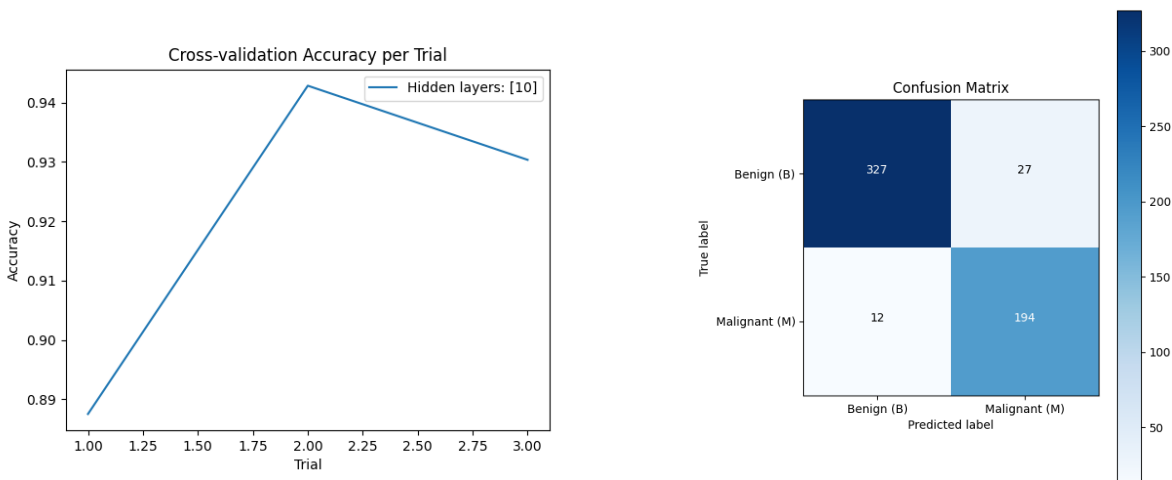
กรณีเปลี่ยน crossover rate

กำหนดให้ population size = 500 , num generations = 350 , mutation rate = 0.1 , crossover rate = 0.1 และให้ hidden layers = 10 โดยทดลองทั้งหมด 3 รอบแล้วหาค่าเฉลี่ยพบว่าได้ข้อสรุปดังนี้



ได้ผลลัพธ์เป็น Average cross-validation accuracy over 3 trials: 0.92

กำหนดให้ population size = 500 , num generations = 350 , mutation rate = 0.8 , crossover rate = 0.1 และให้ hidden layers = 10 โดยทดลองทั้งหมด 3 รอบแล้วหาค่าเฉลี่ยพบว่าได้ข้อสรุปดังนี้



ได้ผลลัพธ์เป็น Average cross-validation accuracy over 3 trials: 0.92

วิเคราะห์ผลการทดลอง

ในการวิเคราะห์ผลการทดลองนี้ สามารถพิจารณาได้จากค่าเฉลี่ยของความแม่นยำ (Average Accuracy) และการกระจายตัวของผลการทดลองแต่ละครั้ง เพื่อประเมินประสิทธิภาพของแบบจำลองว่ามีความเสถียรเพียงใดและจากการพิจารณา Confusion Matrix ยังช่วยแสดงให้เห็นถึงการทำนายที่ถูกต้องและผิดพลาดของแต่ละกลุ่มข้อมูล ซึ่งช่วยให้เข้าใจถึงความสามารถของแบบจำลองในการจำแนกข้อมูลระหว่างสองคลาสได้ดีขึ้น ดังรายละเอียดต่อไปนี้

1. ความแม่นยำเฉลี่ย (Average Accuracy)

ค่าเฉลี่ยของความแม่นยำ (Average cross-validation accuracy) เป็นตัวบ่งชี้ถึงความสามารถของแบบจำลองในการจำแนกข้อมูล หากค่าเฉลี่ยสูงกว่า 80% ถือว่าประสิทธิภาพดี แต่หากต่ำกว่า 70% ควรพิจารณาปรับพารามิเตอร์หรือเพิ่มความซับซ้อนให้กับแบบจำลอง เช่น เพิ่มชั้นซ่อน (Hidden Layers) หรือขนาดประชากรในอัลกอริทึมเชิงพันธุกรรม

2. ความผันแปรของความแม่นยำระหว่างการทดลอง (Trial Accuracy Variation)

การกระจายตัวของค่า Accuracy ในแต่ละ Trial บ่งบอกถึงเสถียรภาพของแบบจำลอง หากค่า Accuracy ผันแปรมากระหว่าง Trial อาจแสดงถึงความไม่เสถียรของแบบจำลองหรือขนาดชุดข้อมูลที่ใช้ใน Cross-Validation ไม่เพียงพอ แต่หากค่ามีความใกล้เคียงกัน จะบ่งบอกถึงเสถียรภาพที่สูงขึ้น และสะท้อนประสิทธิภาพที่แท้จริงของแบบจำลองได้ดียิ่งขึ้น

3. Confusion Matrix

Confusion Matrix แสดงถึงการทำนายที่ถูกต้องและผิดพลาดของแบบจำลองในแต่ละคลาส โดยค่าที่อยู่บนเส้นทแยงมุมหลัก (มุมบนซ้ายและมุมล่างขวา) จะแสดงจำนวนการทำนายที่ถูกต้อง ในขณะที่ค่าที่อยู่นอกเส้นทแยงมุมหลักแสดงถึงการทำนายที่ผิดพลาด

สรุปผลการทดลอง

การปรับเปลี่ยนพารามิเตอร์ในโมเดลนี้มีผลต่อประสิทธิภาพการทำนายและเสถียรภาพของโมเดล ดังนี้:

1. ขนาดประชากร (Population Size)

การเพิ่มขนาดประชากร: ช่วยเพิ่มความหลากหลายให้กับแต่ละรุ่น (Generation) ในกระบวนการของอัลกอริทึมเชิงพันธุกรรม ซึ่งอาจทำให้โมเดลมีโอกาสค้นพบชุดค่าพารามิเตอร์ที่มีประสิทธิภาพสูงขึ้น โดยเฉพาะอย่างยิ่งในข้อมูลที่มีความซับซ้อน อย่างไรก็ตาม การเพิ่มขนาดประชากรมากเกินไปจะทำให้การคำนวณใช้เวลามากขึ้น

การลดขนาดประชากร: ลดเวลาที่ใช้ในการประมวลผลแต่ละรุ่น อย่างไรก็ตาม อาจทำให้โมเดลไม่สามารถค้นพบวิธีการปรับปรุงที่เหมาะสมได้ เนื่องจากความหลากหลายในการทดลองลดลง

2. จำนวนชั่วอายุ (Generations)

การเพิ่มจำนวนชั่วอายุ: ช่วยให้กระบวนการปรับปรุงพารามิเตอร์สามารถดำเนินไปได้หลายรอบ ซึ่งอาจทำให้โมเดลมีโอกาสปรับค่าพารามิเตอร์ให้เหมาะสมยิ่งขึ้น โดยเฉพาะอย่างยิ่งหากต้องการความแม่นยำสูงสุด อย่างไรก็ตาม การเพิ่มจำนวนชั่วอายุจะทำให้เวลาที่ใช้ในการฝึกโมเดลเพิ่มขึ้น

การลดจำนวนชั่วอายุ: ช่วยให้การประมวลผลเสร็จเร็วขึ้น แต่หากจำนวนชั่วอายุไม่เพียงพอ โมเดลอาจไม่สามารถปรับปรุงให้ได้ผลลัพธ์ที่ดี ทำให้ประสิทธิภาพลดลง

3. อัตราการกลายพันธุ์ (Mutation Rate)

การเพิ่มอัตราการกลายพันธุ์: ช่วยเพิ่มความหลากหลายของลักษณะพันธุกรรมในแต่ละชั่วอายุ โดยการสุ่มเปลี่ยนแปลงบางค่าในตัวบุคคล (Individual) อัตราการกลายพันธุ์ที่สูงสามารถช่วยให้โมเดลหลีกเลี่ยงการติดในค่าที่แย่ได้ (Local Optimum) แต่หากสูงเกินไป อาจทำให้โมเดลมีความไม่เสถียรและเกิดการสุ่มมากเกินไป

การลดอัตราการกลายพันธุ์: ช่วยรักษาลักษณะพันธุกรรมเดิมของประชากรให้คงอยู่ ทำให้โมเดลไม่ผันแปรมากเกินไป แต่อาจทำให้โมเดลมีโอกาสสูงในการติดอยู่ในค่าที่แย่ได้ (Local Optimum) ซึ่งอาจลดประสิทธิภาพในระยะยาว

4. อัตราการผสมข้ามพันธุกรรม (Crossover Rate)

การเพิ่มอัตราการผสมข้ามพันธุกรรม: ทำให้ประชากรในแต่ละรุ่นมีการผสมพันธุกรรมกันบ่อยขึ้น ส่งผลให้การค้นหาวิธีแก้ปัญหาที่ดีขึ้นทำได้เร็วขึ้น แต่หากอัตราการผสมข้ามสูงเกินไป ประชากรอาจขาดเสถียรภาพและทำให้แบบจำลองมีการเปลี่ยนแปลงมากเกินไปในแต่ละชั่วอายุ

การลดอัตราการผสมข้ามพันธุกรรม: ช่วยให้ลักษณะทางพันธุกรรมของแต่ละรุ่นมีการเปลี่ยนแปลงน้อยลง ซึ่งอาจช่วยให้แบบจำลองมีความเสถียรในระยะยาว แต่ในบางกรณี อัตราการผสมข้ามที่ต่ำเกินไป อาจทำให้แบบจำลองปรับปรุงได้ช้าและลดโอกาสในการค้นหาค่าที่ดีขึ้นในรุ่นถัดไป

5. โครงสร้างของชั้นซ่อนในโครงข่ายประสาทเทียม (Hidden Layers Structure)

การเพิ่มจำนวนชั้นซ่อน: ทำให้แบบจำลองสามารถเรียนรู้ความซับซ้อนของข้อมูลได้มากขึ้น โดยเฉพาะกับข้อมูลที่มีความซับซ้อนสูง แต่การเพิ่มชั้นซ่อนมากเกินไปอาจทำให้โมเดลมีความเสี่ยงที่จะเกิดการฟิตมากเกินไป (Overfitting) และทำให้การฝึกอบรมช้าลง

การลดจำนวนชั้นซ่อน: ลดความซับซ้อนของแบบจำลอง ช่วยให้การฝึกอบรมเร็วขึ้น แต่หากจำนวนชั้นซ่อนน้อยเกินไป โมเดลอาจไม่สามารถเรียนรู้ความซับซ้อนของข้อมูลได้ดี ส่งผลให้ประสิทธิภาพการทำงานย่ำแย่ลง

สรุปผลจากการทดลองปรับพารามิเตอร์

การปรับค่าพารามิเตอร์ต่าง ๆ ในการทดลองนี้ควรทำด้วยความระมัดระวังเพื่อหาจุดสมดุลระหว่างความแม่นยำและเสถียรภาพ โดยการทดสอบในหลาย ๆ ค่าและปรับตามความเหมาะสมจะช่วยให้แบบจำลองทำงานได้มีประสิทธิภาพมากขึ้นตามที่ต้องการ