**COMPUTER HOMEWORK 3**

**Training Multilayer Perceptron using Genetic Algorithm**

**จัดทำโดย**

**นายศุภเดช บริสุทธนารักษ์**

**รหัสนักศึกษา 600610783**

**เสนอ**

**รศ.ดร.ศันสนีย์ เอื้อพันธ์วิริยะกุล**

**รายงานนี้เป็นส่วนหนึ่งของวิชา 261456 Introduction to Computational Intelligences**

**ภาคเรียนที่ 1 ปีการศึกษา 2563**

**ภาควิชาวิศวกรรมคอมพิวเตอร์**

**คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่**

**สารบัญ**

**การทำงานของโปรแกรม**

1. **แหล่งข้อมูลสำหรับการทำระบบ**

ชุดข้อมูลที่ถูกนำมาใช้ในการฝึกมาจาก Wisconsin Diagnostic Breast Cancer (WDBC) จาก UCI Machine learning Repository จำนวน 569 ตัวอย่าง แต่ละตัวอย่างประกอบไปด้วยฟีเจอร์ของข้อมูลทั้งหมด 30 features และมี 1 feature สำหรับแบ่งคลาสจำนวนทั้งหมด 2 คลาส ได้แก่ M = malignant และ B = benign

ก่อนที่จะนำข้อมูลไปฝึก ผู้จัดทำได้ทำการ normalize ข้อมูลให้อยู่ในช่วง [0, 1] ด้วยวิธีการ min-max normalization เพื่อป้องกันไม่ให้ข้อมูลในบางฟีเจอร์มีค่ามากเกินไป หรือน้อยเกินไปจนกระทบกับ feature อื่นๆ ด้วย

1. **วิธีการทำงานของระบบ**
2. สร้างประชากร P ตามจำนวน chromosomes ที่ฟังก์ชันนี้รับค่าเข้ามา โดยที่ chromosome นี้จะประกอบด้วย genes ซึ่งเป็นค่า weight ที่ใช้ใน Multilayer Perceptron และค่า weight แต่ละตัวจะถูกสุ่มค่าขึ้นมา
3. คำนวณค่าความเหมาะสม (fitness) จากความสัมพันธ์

โดยที่ accuracy คือจำนวนที่แต่ละ sample ทายถูก และ N คือจำนวน sample ทั้งหมด ดังนั้น fitness function จะมีค่าได้ตั้งแต่ [0, 1]

1. จดจำ chromosome ที่ให้ค่า fitness ดีที่สุด (สำหรับโปรแกรมนี้จะเก็บ chromosome ที่ให้ค่า fitness มากที่สุด)
2. การคัดเลือกโครโมโซม (Selection) จะใช้วิธีการ Stochastic Tournament

* สุ่มเลือกโครโมโซมจากประชากร 2 ตัวขึ้นมาแข่งขัน
* คำนวณหาค่า fitness จากสมการในข้อ 2 ของแต่ละโครโมโซมที่เลือกมา
* สุ่มค่า R ตั้งแต่ 0 ถึง 1 และสร้างตัวแปร p เป็นค่าความน่าจะเป็นที่ต้องกำหนดขึ้นมา
* ถ้า R < p แล้วเลือกโครโมโซมที่มีค่า fitness น้อยกว่า copy ให้ไปอยู่ใน list ของบ่อผสมพันธุ์ (Mating Pool) มิฉะนั้นให้เลือกโครโมโซมที่มีค่า fitness มากกว่า
* สุ่ม 2 โครโมโซมประชันกันเรื่อยๆ จนจำนวนโครโมโซมในบ่อผสมพันธุ์ครบ N ตัว
* ปรับลดค่า p เมื่อเวลาผ่านไป t generation ตามสมการ โดยที่ c เป็นค่าคงที่บวก

1. การทำครอสโอเวอร์ (Crossover) จะใช้วิธีการ 2-point crossover

* ทำการจับคู่ Crossover ให้กับทุกโครโมโซมใน Mating Pool ทุกตัว หากจำนวนโครโมโซมเป็นจำนวนคี่ให้เติมโครโมโซมเข้ามาใหม่ 1 ตัว แล้วค่อยสุ่มจับคู่
* สุ่มค่า k โดยที่ (L คือความยาวของโครโมโซม) 2 ครั้ง จะได้ค่าเป็น k1 และ k2
* สลับคู่โครโมโซมที่ตำแหน่งที่ k1 และสลับอีกครั้งที่ตำแหน่ง k2
* แทนที่โครโมโซมใหม่ ทับกับโครโมโซมพ่อและแม่ของตัวเอง กลายเป็นประชากรรุ่นใหม่

1. นำประชากรรุ่นใหม่ที่ได้จากการทำครอสโอเวอร์ มาเข้าสู่กระบวนการกลายพันธุ์ (Mutation) ซึ่งจะใช้แบบ Strong mutation

* สุ่มค่าความน่าจะเป็นของการเกิดการกลายพันธุ์ (pm) อยู่ในช่วง [0.001, 0.01]
* สำหรับทุกโครโมโซมและทุกตำแหน่งของโครโมโซม
* สุ่มค่า q จาก Uniform Distribution U(0,1)
* ถ้า q < pm ให้สุ่ม weight ที่ตำแหน่งนั้นใหม่ แต่ถ้า q >= pm ค่าในตำแหน่งนั้นจะไม่มีการเปลื่ยนแปลง
* ผลลัพธ์ที่ได้จากการทำ Mutation จะกลายเป็นประชากรใน generation ถัดไป

**การทดลอง**

**ภาคผนวก**

Source Code ที่ใช้ในการทำการบ้านชุดนี้

1. ไฟล์หลัก “hw3.py”
2. “GAMLP.py”
3. “MLP.py”
4. “load\_text.py”