

ปัญหาและการวิเคราะห์ปัญหา

ปัจจุบัน ปัญญาประดิษฐ์ (AI) เริ่มมีบทบาทในชีวิตประจำวันมากขึ้น ซึ่ง AI สามารถทำงานแทนมนุษย์ได้ และในประเทศไทยยังไม่ค่อยมีการใช้งาน AI มากนัก เราจึงเริ่มต้นด้วยการพัฒนาการโปรแกรมที่รับข้อมูลเป็นรูปของตัวเลขไทย 1 หลักที่ผู้ใช้งานเป็นคนวาด (Input) แล้วทำการวิเคราะห์ และเรียนรู้รูปร่างนั้นผ่านโครงข่ายประสาทเทียม(Neural network)ซึ่งเป็นพื้นฐานของการเรียนรู้ของเครื่อง(Machine Learning) หรือเป็นขั้นตอนเริ่มต้นของการสร้างปัญญาประดิษฐ์ ให้รู้จักสิ่งที่วาด (Process) แล้วให้โปรแกรมทายผลที่ได้ว่าเลขไทยที่ผู้ใช้งานวาดมานั้นเป็นเลขอะไร (Output)

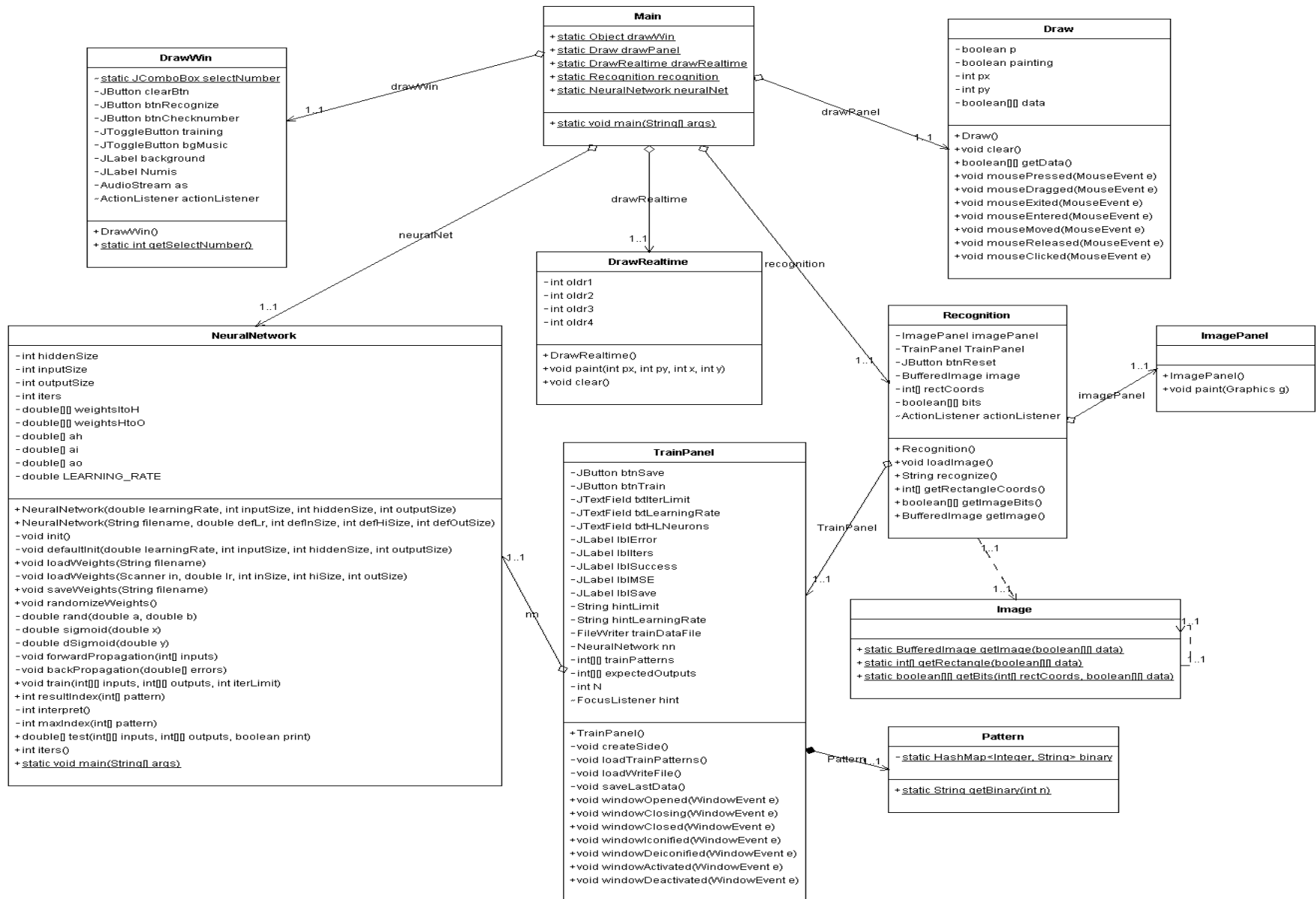
วัตถุประสงค์

- ฝึกให้ AI เรียนรู้รูปแบบการวาด(Pattern)ที่ผู้ใช้งานมา 500 แบบ
- เพื่อเอาเนื้อหาต่างๆในกระบวนวิชา และหลักการของ Object Oriented Programming (OOP) มาประยุกต์ใช้ให้เกิดประโยชน์
- เพื่อเป็นโปรแกรมพื้นฐานในการต่อยอดและพัฒนาด้าน Machine Learning ของประเทศไทยต่อไป

แนวทางการแก้ปัญหา

- วางแผนและวิเคราะห์เกี่ยวกับโปรแกรมที่จะพัฒนา
- ศึกษาหลักการการทำงานอย่างง่ายของ Machine Learning
- ออกแบบอัลกอริทึมโดยอาศัยฟังก์ชัน Sigmoid
- ทดสอบการทำงานของอัลกอริทึม
- ออกแบบและสร้าง User interface (UI)
- วาดตัวเลขไทยระหว่าง 0 ถึง 9 และเซฟเก็บไว้ แล้วให้โปรแกรมเรียนรู้
- ทดสอบการทำงานของโปรแกรม
- สรุปและนำเสนอ

การออกแบบคลาส (Class Design)



Class Main

เป็น main() หลักของตัวโปรแกรมสำหรับเปิดโปรแกรมขึ้นมา

Class DrawWin extends JFrame

เป็นคลาสสำหรับทำหน้าต่างผู้ใช้ ซึ่งสืบทอดจากคลาส JFrame

Class Draw extends JPanel implements MouseMotionListener, MouseListener

เป็นคลาสสำหรับพื้นที่วาดตัวเลขไทย สืบทอดจากคลาส JPanel เพื่อนำไปใส่ในหน้าต่างผู้ใช้ (JFrame) และมีการใช้ Interface class ของ MouseMotionListener และ MouseListener ในการจัดการเคลื่อนไหวและการกระทำต่าง ๆ ของ mouse และเก็บตำแหน่งของพื้นที่เมื่อกดวาดเป็น array of Boolean 2 มิติ (true คือตำแหน่งที่ถูกวาด false คือ ตำแหน่งที่ไม่ถูกวาด)

Class DrawRealtime extends JPanel

เป็นคลาสสำหรับวิเคราะห์หาสิ่งที่วาดลงไปในพื้นที่วาด มีการสืบทอดจากคลาส JPanel เพื่อนำไปใส่ในหน้าต่างผู้ใช้ (JFrame)

Class Recognition extends JFrame

เป็นคลาสสำหรับทำหน้าต่างที่มีการสืบทอดจากคลาส JFrame เป็นหน้าต่างเพื่อให้ผู้ใช้สอนโปรแกรมให้เรียนรู้ตัวเลขไทย

Class TrainPanel extends JComponent implements WindowListener

เป็นคลาสเครื่องมือใช้สำหรับให้โปรแกรมเรียนรู้และวิเคราะห์ ซึ่งจะเป็นคลาสเชื่อมต่อระหว่างหน้าบ้านและหลังบ้าน โดยมีการสืบทอดจากคลาส JComponent และมีการใช้คลาส Interface ของ WindowListener เพื่อกำหนดการกระทำของหน้าต่าง

Class ImagePanel extends JPanel

เป็นคลาสสำหรับเก็บข้อมูลรูปภาพเพื่อหา pixel และมีกรอบโฟกัสกรอบสิ่งที่วาด โดยมีการสืบทอดจากคลาส JPanel เพื่อนำไปใส่ในหน้าต่างผู้ใช้ (JFrame)

Class Image

เป็นคลาสสำหรับแปลงค่าข้อมูลตำแหน่งที่ถูกวาด (Boolean) เป็นข้อมูลรูปภาพบิตแมป, แปลงค่าข้อมูลตำแหน่งที่ถูกวาดเป็นข้อมูล bit 0 หรือ 1 และเก็บค่าตำแหน่งต่ำสุด สูงสุด ทั้งแกน x และแกน y เพื่อนำไปใช้โฟกัสสิ่งที่มีการวาด

Class NeuralNetwork

เป็นคลาสสำคัญที่ทำให้โปรแกรมมีการเรียนรู้ผ่านอัลกอริทึมต่าง ๆ รวมทั้งฟังก์ชัน sigmoid จากการที่ผู้ใช้ป้อนการสอนเข้ามาและมีการทดสอบความแม่นยำจากค่า weight (ค่าที่ได้จากการเรียนรู้หรือค่าที่ให้นำหนักความแม่นยำของ Neuron)

Class Pattern

เป็นคลาสบอกรูปแบบ Output ของตัวเลขไทย ตั้งแต่ 0 ถึง 1 เพื่อใช้สำหรับตรวจสอบข้อมูลเข้ามา (Input)

โปรแกรมหลัก (Main Program)

โปรแกรม Guess จะแบ่งเป็น 2 ส่วน คือ ส่วนทดสอบและส่วนการสอนให้โปรแกรมเรียนรู้

1. ส่วนทดสอบ

รับ input จากผู้ใช้ที่เขียนเข้ามา ค่า input ที่ใช้จะเป็นค่าตำแหน่งบนหน้าต่างในแนวแกน x และแกน y โดยแบ่งเป็นแกนละ 280 หน่วย (0 ถึง 280) โดยข้อมูลนำเข้ามาจะเก็บเป็นค่าความจริง (Boolean) และนำค่าที่ได้ให้โปรแกรมคำนวณหา Output ออกมาเทียบกับรูปแบบที่บันทึกไว้ในไฟล์ train.txt และแสดงผลออกมา



หมายเลข 1 ช่องสำหรับเขียนตัวเลขไทย

หมายเลข 2 ปุ่มตรวจสอบคำตอบ

หมายเลข 3 ปุ่มล้างหน้าต่างในหมายเลข 1

หมายเลข 4 ตำแหน่งแสดงผลคำตอบ

หมายเลข 5 ปุ่มเปลี่ยนโหมด (สถานะสีแดง = โหมดทดสอบ)

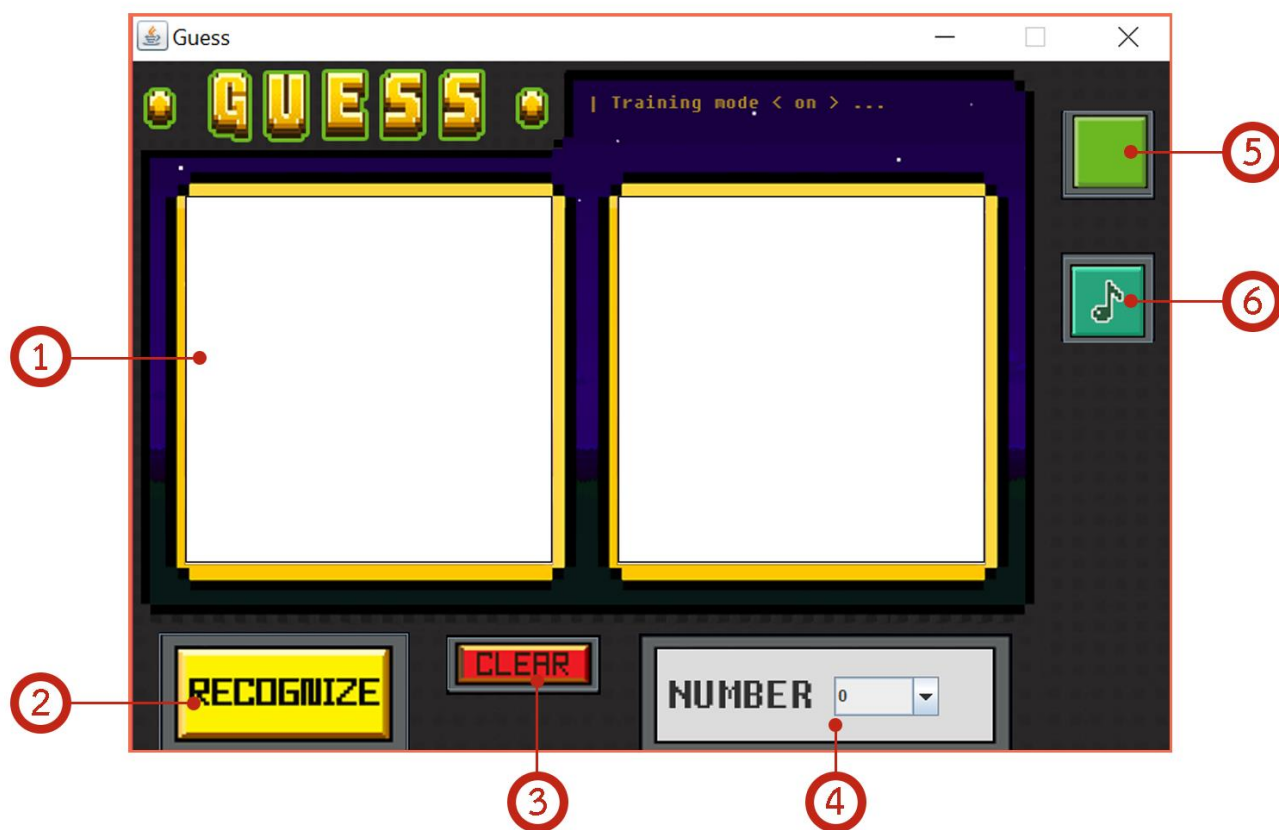
หมายเลข 6 ปุ่มปิด-เปิดเสียง

2. ส่วนการสอนให้โปรแกรมเรียนรู้

รับ input จากผู้ใช้ที่เขียนเข้ามาและบอกตัวโปรแกรมว่าเลขที่เขียนเป็นเลขอะไร เพื่อนำไปเปรียบเทียบกับรูปแบบที่กำหนดไว้ (Pattern) แล้วให้มันพัฒนาตัวเองและเรียนรู้ จากนั้นเมื่อเราให้ Input ใหม่เข้าไป ตัว Machine Learning ก็จะทำให้คำตอบออกมาโดยอ้างอิงจากสิ่งที่มันได้เรียนรู้ไปแล้ว โดยส่วนที่เล็กที่สุดของ Neural Network ก็คือ Neuron ซึ่งทำหน้าที่คำนวณ input ที่เข้ามาเพื่อให้ได้ผลลัพธ์ออกไป โดยมีส่วนประกอบสำคัญดังนี้

- Weight เป็นการให้น้ำหนักของค่าที่ส่งเข้ามา (input) โดยมีค่าระหว่าง 0-1 เมื่อเริ่มต้นจะเป็นการ Random ขึ้นมา จากนั้นตัว Neuron ทำการเรียนรู้เรื่อยๆ ก็จะเป็นการปรับ weight ให้มันได้คำตอบ (output) ที่ใกล้เคียงที่สุด
- Bias คือค่าที่จะช่วยเข้ามาทำให้ค่าที่เข้ามาอยู่ในระหว่าง 0 - 1 ได้ โดยจะเป็นเลข random และปรับเปลี่ยนไปเรื่อยๆ ทุกครั้งที่เรียนรู้
- Output คือผลลัพธ์
- Back Propagation คือการที่ Neuron นำค่า Error ของ Output ที่ได้ กับ Output ที่กำหนดไว้ นำไปปรับ Weight และ Bias ให้เกิดผลลัพธ์ที่ถูกต้องตามที่ได้เรียนรู้มา

ฟังก์ชันของโปรแกรมที่พัฒนานี้ จะใช้เลือกใช้ฟังก์ชันที่ง่ายและเป็นพื้นฐานที่สุดของ Neural Network คือ ฟังก์ชัน Sigmoid



หมายเลข 1 ช่องสำหรับเขียนตัวเลขไทย

หมายเลข 2 ปุ่มเพื่อไปยังหน้า Recognition

หมายเลข 3 ปุ่มล้างหน้าต่างในหมายเลข 1

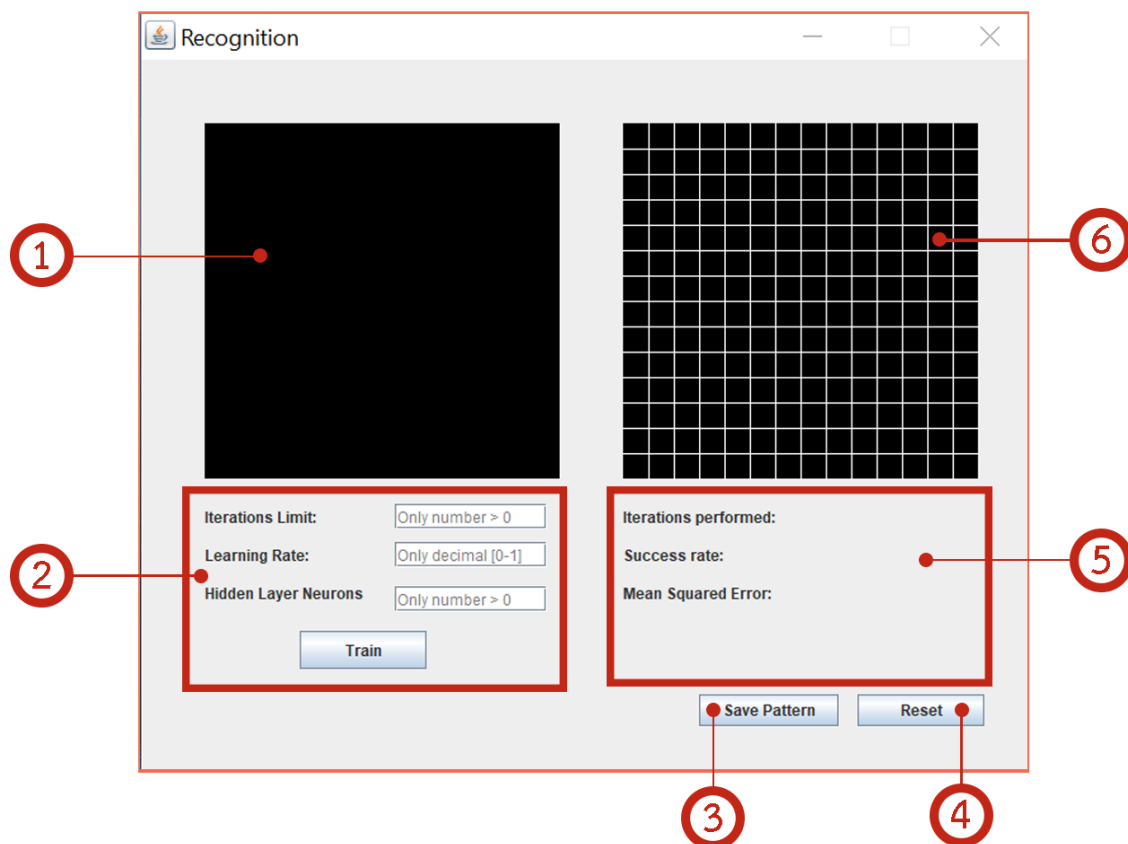
หมายเลข 4 ปุ่มเลือกตัวเลข

หมายเลข 5 ปุ่มเปลี่ยนโหมด (สถานะสีเขียว = โหมดเรียนรู้)

หมายเลข 6 ปุ่มปิด-เปิดเสียง

การสอนในแต่ละครั้ง จะมีการกำหนดจำนวนรอบที่จะให้โปรแกรมเรียนรู้ ดังนั้นการสอนตัว Neuron นั้นจำเป็นต้องสอนหลาย ๆ รอบเพื่อให้เกิด Error น้อยที่สุด อย่างที่สองคือความเร็วในการเรียนรู้ในแต่ละรอบ หรือ Learning rate เป็นตัวบ่งบอกว่าค่าที่เรียนรู้จะเปลี่ยนแปลงไปมากหรือน้อย อย่างสุดท้ายที่ผู้ใช้จะให้โปรแกรมทำคือ ต้องการให้ตัว Neural กี่ตัว ซึ่งจะเปรียบได้กับการทำงานคนเดียว ก็จะได้งานได้แค่ระดับหนึ่ง แต่ถ้าเป็นงานใหญ่มากหรือเป็นข้อมูลที่รับเข้ามา (input) มาก จะต้องใช้ Neuron หลายตัวในการช่วยกันเรียนรู้

เบื้องหลังการทำงานขณะ Train จะเป็นกระบวนการอัลกอริทึมที่เรียกว่า Feed-Forward Neural Network โดยจะแบ่ง Neuron ออกเป็นกลุ่มๆ โดยแต่ละกลุ่มจะเรียกเป็น Layer โดยข้อมูลที่เข้ามาจะไหลไปในทิศทางเดียว ไม่ไหลย้อนกลับ จาก Layer หนึ่งสู่อีก Layer หนึ่ง



หมายเลข 1 หน้าต่างแสดง Pixel จากพื้นที่ที่วาด (ขนาด 280 x 280 pixel)

หมายเลข 2 ช่องกรอกค่าต่างๆ เพื่อสอนให้โปรแกรมเรียนรู้

หมายเลข 3 ปุ่มบันทึกค่าที่เรียนรู้ได้และรูปแบบที่ผู้ใช้วาดเข้ามา

หมายเลข 4 ปุ่มยกเลิกการ Train กลับไปยังหน้าแรก

หมายเลข 5 แสดงค่าที่โปรแกรมได้เรียนรู้จากข้อมูลการสอนของหมายเลข 2

หมายเลข 6 ขยาย Pixel ของรูปที่โปรแกรมโฟกัสอยู่

Capture Output



โหมดทดสอบ (สถานะสีแดง)

1. หมายเลข 1 ให้ผู้ใช้งานเขียนเลขไทยหนึ่งหลัก
2. กดปุ่ม CHECK หมายเลข 2 เพื่อเช็คหมายเลข
3. โปรแกรมจะแสดงผลพร้อมออกมาที่ตำแหน่งหมายเลข 3 ดังรูป

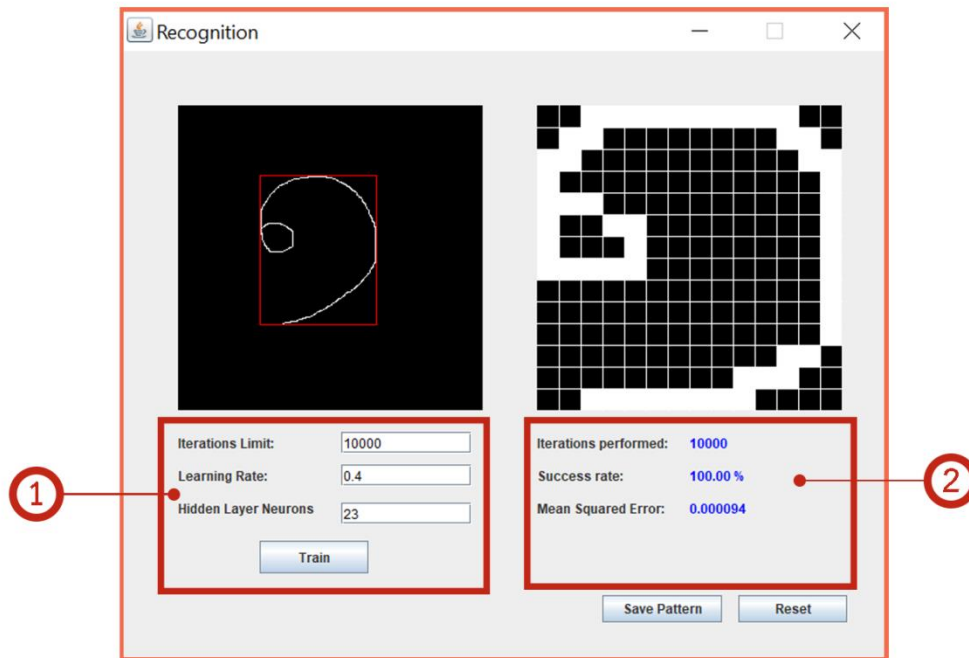


4. ผู้ใช้สามารถกดปุ่ม CLEAR สำหรับเคลียร์รูปเพื่อเขียนเลขใหม่ได้



โหมดเรียนรู้ (สถานะสีเขียว)

1. หมายเลข 2 ให้ผู้ใช้กดเลือกหมายเลขที่ต้องการ Train
2. หมายเลข 1 ให้ผู้ใช้งานเขียนเลขไทยหนึ่งหลักตามที่ได้เลือกไว้ในช่องหมายเลข 2
3. กดปุ่ม RECOGNIZE หมายเลข 3 เพื่อไปยังหน้า Recognition สำหรับการ Train



หน้า Recognition

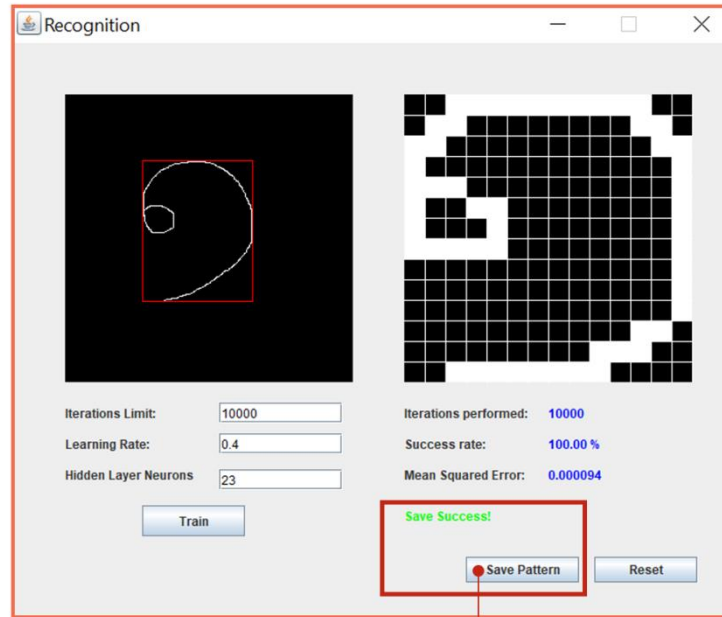
1. กรอกค่าในหมายเลข 1

- ช่องแรกกรอกจำนวนรอบสำหรับการ Train (กรอกค่ามากกว่า 0)
- ช่องที่ 2 กรอกค่าเรทการเรียนรู้ (กรอกค่าทศนิยมระหว่าง 0 ถึง 1)
- ช่องที่ 3 กรอกจำนวน hidden layer ของ neuron (กรอกค่ามากกว่า 0)

2. กดปุ่ม Train เพื่อให้โปรแกรมวิเคราะห์ และเรียนรู้

3. หมายเลข 2 จะแสดงผลเมื่อทำตามข้อ 1 และ 2 เสร็จแล้ว จะแสดงค่าจำนวนรอบที่โปรแกรมได้ Train ไป, ค่าเปอร์เซ็นต์ความสำเร็จที่เรียนรู้ได้ และค่า Error เฉลี่ยที่โปรแกรมเรียนฝึกพลาด

4. หมายเลข 3 กด Save Pattern เพื่อบันทึกข้อมูลที่โปรแกรมเรียนรู้ได้ เมื่อบันทึกผลเสร็จเรียบร้อยแล้วจะแสดงดังรูป



5. กดปุ่ม Reset เพื่อไปยังโหมดทดสอบ

แหล่งอ้างอิง

- <http://python3.wannaphong.com/2016/04/neural-networks-python.html?m=0>
- <https://coladev.com/machine-learning/neural-network/2017/02/22/neural-network-basic>
- <https://medium.com/@sanparithmarukatat/deep-learning-67d82eaffc0d>
- <http://incompetech.com/music/royalty-free/index.html?isrc=USUAN1100764>